Computers **2014**, *3*, 130-173; doi:10.3390/computers3040130

OPEN ACCESS

computers

ISSN 2073-431X

www.mdpi.com/journal/computers

Review

A Survey on M2M Service Networks

Juhani Latvakoski [1,*], Antti Iivari [1], Paul Vitic [2], Bashar Jubeh [3], Mahdi Ben Alaya [4], Thierry Monteil [4], Yoann Lopez [5], Guillermo Talavera [6], Javier Gonzalez [6], Niclas Granqvist [7], Monir Kellil [8], Herve Ganem [9] and Teemu Väisänen [1]

[1] VTT Technical Research Centre of Finland Kaitoväylä 1, Oulu 90571, Finland; E-Mails: antti.iivari@vtt.fi (A.I.); teemu.vaisanen@vtt.fi (T.V.)

[2] Vektor Telematics Necatibey Caddesi, Akçe Sokak, No: 6 Karaköy Istanbul, 34420 Turkey; E-Mail: paul@vektortelematics.com

[3] Bull, 207, Cours du Medoc, Bordeaux 33000, France; E-Mail: bashar.jubeh@bull.net

[4] CNRS-LAAS, 7 Avenue du Colonel Roche BP 54200 31031 Toulouse cedex 4, France; E-Mails: ben.alaya@laas.fr (M.B.A.); monteil@laas.fr (T.M.)

[5] Thales Communications S.A. 160, Boulevard de Valmy—BP 82. 92704 Colomber, France; E-Mail: yoann.lopez@thalesgroup.com

[6] UAB, Celphis Etse Campus, 08193 Bellaterra (Cerdanvola del Valles), Barcelona, Spain; E-Mails: guillermo.talavera@gmail.com (G.T.); javipapi@gmail.com (J.G.)

[7] Polar Elektro Oy. Professorintie 5, Kempele 90440, Finland; E-Mail: Niclas.Granqvist@polar.com

[8] CEA. Saclay DRT/LIST/DIASI/LSC Bat 528, Boite courrier 94. 91191 Gif-sur-Yvette, Cedex, France; E-Mail: mounir.kellil@cea.fr

[9] Gemalto. 6, rue de la Verrerie, Meudon 92197, France; E-Mail: Herve.Ganem@gemalto.com

***** Author to whom correspondence should be addressed; E-Mail: Juhani.Latvakoski@vtt.fi; Tel.: +358-40-5200-149; Fax: +358-20-722-2320.

External Editor: Aaron Quigley

Received: 8 January 2014; in revised form: 17 October 2014 / Accepted: 20 October 2014 / Published: 20 November 2014

Abstract: The number of industrial applications relying on the Machine to Machine (M2M) services exposed from physical world has been increasing in recent years. Such M2M services enable communication of devices with the core processes of companies. However, there is a big challenge related to complexity and to application-specific M2M systems called “vertical silos”. This paper focuses on reviewing the technologies of M2M service networks and discussing approaches from the perspectives of M2M information

and services, M2M communication and M2M security. Finally, a discussion on technologies and approaches potentially enabling future autonomic M2M service networks are provided. According to our conclusions, it is seen that clear definition of the architectural principles is needed to solve the "vertical silo" problem and then, proceeding towards enabling autonomic capabilities for solving complexity problem appears feasible. Several areas of future research have been identified, e.g., autonomic information based services, optimization of communications with limited capability devices, real-time messaging, creation of trust and end to end security, adaptability, reliability, performance, interoperability, and maintenance.

Keywords: machine to machine communication; Internet of things; cyber-physical systems

1. Introduction

The number of embedded devices has continuously increased in recent years. Traditionally, such devices have worked locally in an independent way and provided services for human users. Advances in radio communication technologies have enabled even mobile connectivity for the referred devices over the Internet. These trends are now visible as the increasing number of application cases which rely on the services exposed from physical equipment, such as sensors, actuators, RFID tags, machines, vehicles and industrial embedded devices. Such service systems are described here as Machine to Machine (M2M) service networks, and they can also be called Internet of Things (IoT), Web of Things, Wireless Sensor Networks and Actuator Networks (WSANs) or Cyber Physical Systems [1,2].

Usually, such M2M service networks include capabilities for remote measurements and remote control of embedded devices. Remote measurements consist of sensing physical phenomenon, storing, sending, receiving and processing of measured information. Remote control of devices includes access control, mutual exclusion, sending, receiving and processing of control commands. The basic enabler for such functionality is M2M connectivity, which links various kinds of embedded devices into the Internet based M2M services. The added value is created by the enabled M2M services based on the use of the measured information in a smart way, reasoning, and execution of smart remote control actions with the M2M asset devices. The real benefits of such smart M2M services will be realized when connecting them with the core processes of the companies. This can enable real-time situation awareness in the company processes, and create opportunities for novel services for customers, increase service quality and enable cost savings, e.g., in maintenance processes. Furthermore, this is also an enabler for transition from product centric to service centric businesses.

The main problems in these M2M service networks arise from the complexity and the vertical fragmentation of M2M markets. This complexity is due to the number of embedded devices, connectivity means, and service platforms and especially to their heterogeneity. Establishing and maintaining an interactive system capable of interoperating with the human user is here expected to go beyond human capabilities soon. M2M market is fragmented to multiple vertical industries, and the resulting systems are usually domain or vendor specific closed systems, also called "vertical silos". In addition, the natural needs of businesses to protect themselves seem to lead such systems that require

special access rights for each specific system, resulting in vendor specific closed systems. This has caused problems, for example in residential home environments, and prevented the emergence of home automation to a large extent. Smart grid solutions cannot interoperate with infrastructure and buildings/homes, even if it would be strongly required to reach higher level energy efficiency. Therefore, it is observed here that the technological complexity and vertical M2M silos are the cause of a *grand research challenge* for the development of a modern ecosystem.

It is assumed here that the referred challenge is so fundamental that it requires novel approaches for the system architectures and application of novel paradigms. It is seen that Internet "IP everywhere" cannot alone solve the problem with industrial M2M systems. Something more is needed, such as horizontal approach and autonomic computing, which may have potential to create basis for solving this sizable challenge. Most of the existing vertical M2M solutions have difficulty in scaling, and therefore enabling the *horizontal model* is important for realizing embedded M2M [3]. Autonomic computing is a concept inspired from biological systems that aims to develop systems capable of automatic management for solving the complexity problem [4,5].

There are multiple attempts to build such autonomic systems, e.g., [6,7], and different design approaches such as externalization and internalization [8]. According to IBM, the evolutionary path to autonomic computing is represented by five levels [9], starting from basic, through managed, predictive, adaptive and finally to autonomic. Such approaches as software agents [10], active networks [11] and policy-based systems [12] have been developed to automate the management tasks to achieve higher response times with less management cost. Attempts have been made to solve the communication complexity problem by using be solved by built-in mechanisms to let systems manage their own communications [13]. The self-management capability aims to solve the rapidly growing complexity of computing system management and enable dynamic future growth of the system. There are four properties that enable autonomic capabilities in a system: self-configuration, self-optimization, self-healing and self-protection [14]. Self-Configuration is the ability of the system to perform configurations according to pre-defined high level policies and seamlessly adapt to change caused by automatic configurations. Self-Optimization is the ability of the system to continuously monitor and control resources to improve performance and efficiency. Self-Healing is the ability of the system to automatically detect, diagnose and repair faults. Self-Protection is the ability of the system to pro-actively identify and protect itself from malicious attacks or cascading failures that are not corrected by self-healing measures. An autonomic system should satisfy these properties through a reactive or a proactive behavior. A reactive autonomic system tries to detect problems or meaningful events and then finds an appropriate action or solution after the problem has been already detected. A proactive autonomic system [15] uses preventive measures to maintain, improve or optimize the system performance. Those measures are based on the analysis of the current state, anticipated events and the predicted system reaction to them.

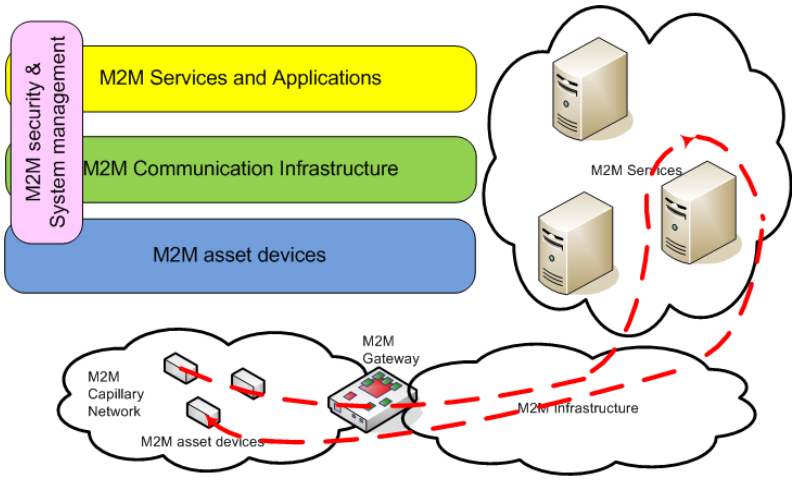
Thus, we review here the technologies to evaluate their capabilities to solve the grand challenge in the context of M2M service networks. The approaches for M2M service networks are first discussed in Chapter 2 and the taxonomy for the review is defined. Then, the related M2M standards and technologies for each category are reviewed as follows: Chapter 3 M2M information and services, Chapter 4 M2M communication and Chapter 5 M2M security. After the review, a discussion is provided in Chapter 6, and finally concluding remarks are provided in Chapter 7.

2. M2M Service Networks

2.1. M2M System

A M2M system usually consists of a set of M2M asset devices attached in a M2M capillary network, a kind of M2M gateway, M2M communication infrastructure and a set of M2M services and applications as shown on Figure 1. The red line represents a typical scenario related to the remote monitoring and control process in M2M service networks. Such a scenario requires operation of the complete M2M system including the functionalities of M2M information and service, M2M communication and M2M security, which are therefore included in a sensible level into this survey.

Figure 1. A view of a Machine to Machine (M2M) system.



M2M services and applications can be divided further to multiple levels such as, e.g., information, service platform and applications. The applications are usually domain specific business logic, high level management of the M2M system, devices, and domain specific information. The *information level* usually contains information management services and exchange transactions between the stakeholders of the system. A standardized common information model can enable smooth information exchange and business interactions between stakeholders of the domain. An example of this kind of common information model is CIM standardized for energy grid [16]. *M2M Service Platform* includes service solutions and frameworks, which may be applied in multiple domains. The service solutions can contain generic service elements such as e.g., event notification, environment monitoring, service discovery and delivery, generic profiling, access control, generic storage and device management [17]. ETSI M2M has specified a set of service capabilities related to application, communication, reachability, addressing and repository, remote management, security, history and data containers [18]. A standardized M2M service platform could enable smooth application development and interaction between service platforms of different vendors, and services of M2M asset devices to be applied in multiple cases.

M2M communication infrastructure contains heterogeneous networks including local M2M asset network (M2M capillary network) such as personal/body area network, vehicular network and wireless sensor network (WSN), and Internet including various overlay networks. The overlay network can logically connect the M2M asset devices, M2M gateways, infrastructure servers and user with each

other to hide the heterogeneity of physical networks and solve such problems as mobility, device power saving features, security including firewalls and NAT restricted networks. Thus the overlay network is a logical network operating on top of physical networks like, e.g., the Internet. The local asset network can contain M2M gateway(s), which may be needed to connect the local area network to the wide area network. If the local M2M asset devices are able to handle communication themselves such a gateway device is not necessarily needed. Several standardized short range radio technologies such as Bluetooth, ZigBee, RFID, *etc.*, can be utilized locally, but also as vendor specific and optimized radio technologies for WSNs. The wide area and Internet connectivity can be provided by, e.g., Telecom operator and/or ISP using, e.g., Ethernet, DSL, GSM, 3GPP, WiFi, WiMax, *etc.* There are a number of standardization bodies such as IETF, 3GPP, Bluetooth, *etc.*, whose works are related to specific areas of connectivity.

M2M asset devices consist of a huge number of heterogeneous embedded devices, which can be general purpose sensors, actuators, tags or M2M gateways in addition to domain, or even business case specific, heterogeneous devices. These devices, ranging from miniaturized sensors to large industrial machinery, operate with different technologies, operational characteristics and environments. It is obvious that all these devices cannot have the same type of interface because the purpose and capabilities can be very different. However, there are usually dependencies with the domain specific application, information, M2M service platform and M2M connectivity levels in the M2M asset devices. This may lead to a need to make changes into various levels of the M2M system, if, e.g., an M2M asset device manufacturer is changed. Therefore, application of standard based technologies in M2M service networks is very important. In addition, the system needs to scale, be interoperable, flexible and extensible during the complete life-cycle [3,19].

Last but not the least is security and management of the complete M2M system. They are seen to be cross issues, which mean that these features shall be built-in to the levels of the system applying widely approved standards. In addition, the needs of ensuring services provided by industrial companies needs to be taken into account.

2.2. Related M2M Standardization Approaches

An analysis of some M2M standardization approaches is provided in Table 1. At least a tiny IP stack is required in the end-to-end Internet based approach to enable connectivity for small devices. If such a stack is available, then this approach may be possible. However, the challenge is that also the embedded devices which are not necessarily Internet-capable are required to be connected to the Internet. The M2M gateway based approach may enable also their connectivity; however, the challenge may be dynamic behavior of the wireless systems and need to adapt with different kinds of service back-end systems.

A number of industry associations have been established to cover some specific category of M2M asset devices, such as electronic product codes, generic identification numbering, RFID tags, video devices, electricity, gas, water and heat meters and different kind of sensors (e.g., Geospatial) and devices for specific applications (e.g., medical devices).

The capability of user interface to adapt to the dynamic behavior of wireless devices has been a challenge. UIML type of approach seems to be too heavy to be applied with the UIs of embedded

M2M asset devices. It is also possible to apply the natural need of the M2M asset device manufacturers to make drivers and UIs for their products, and connect these SW components into the dynamic configuration and discovery process [20].

Table 1. M2M Standardization approaches.

Approach/Focus	Forums	Contributions
End-to-End Internet based approach	- IPSO Alliance—Enabling the Internet of Things [21] - The Internet Engineering Task Force (IETF) [22]	- Network related standards, e.g., IPv6 over Low Power WPAN (6LoWPAN) - Constrained RESTful Environments (CoRE) - Routing Over Low power and Lossy Networks (ROLL) - RPL for tiny battery operated devices
M2M gateway based approach	- ETSI M2M/Smart M2M Technical Committee [23] - One M2M forum [24]	- Generic horizontal service capability layer with standard interfaces - M2M concept and architecture
Electronic product codes	EPC Global [25]	Usage of electronic product codes with RFID technology
Generic identification numbering	uID center [26]	Identification of objects and places uniquely, and association of information with them
RFID coordination	Coordination And Support Action For Global RFID-related Activities and Standardization (CASAGARAS)	- Global Coding System (GCS) in relation to RFID systems - Ontologies for identification - IoT architectural components
Video devices	ONVIF [27]	ONVIF defines a common protocol for the exchange of information between network video devices including automatic device discovery, video streaming and metadata
Electricity, gas, water and heat meters	Openmeter [28]	Development of open standards for advanced meter interface (AMI)
Sensors	OGC Sensor Web Enablement (SWE) [29]	Sensor web enablement standards to enable developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and usable via Web
User interfaces	UIML [30]	User Interface Mark-up Language (UIML), which allows dynamic change of the UI content by enabling UI generation
Devices and their services	Universal Plug and Play (UPnP) [31]	Configuration and discovery of the devices and their services, UPnP
(Building) automation and control networks	For BACnet: ASHRAE Standing Standard Project Committee (SSPC)	- A data communication protocol for building automation and control networks. - BACnet, Modbus, DNP3, CAN, *etc.*
Radio access protocols for IoT and M2M	IEEE, NFC Forum, ZigBee Alliance, Bluetooth SIG	ZigBee, NFC, Bluetooth, WiFi, *etc.*
Semantic access to IoT and M2M data	W3C	- Ontology Web Language (OWL) - Darpa Agent Mark-up Language (DAML) - Resource Description Framework (RDF)
Medical devices	- Center for Integration of Medicine and Innovative Technology (CIMIT) - Continua Health Alliance	Standards and/or profiles for health related or medical devices

Dynamic configuration and discovery methods have been applied in large scale within peer to peer (P2P) content delivery systems, e.g., Napster, KaZaA, Gnutella, Morpheus and BitTorrent [32]. However, content delivery differs essentially from the discovery of services exposed from M2M asset devices. There are some service discovery systems targeted to local scale and embedded devices such as UPnP and Bluetooth SDP. However, there is lack of solutions for large-scale wireless and hybrid networks [33].

Overlay networking is based on the virtual communication layer, which is built on top of other transport media and/or physical network [34]. Overlay networking has been applied to improve the robustness and availability of Internet paths between hosts (e.g., MIT RON), enable smooth transition to the improved technology (e.g., 6Bone) or to reduce network load by peer assisted data delivery (e.g., BitTorrent). Overlays have been used to route control messages and connect different entities (e.g., SIP and XMPP), and also to implement data forwarding and dissemination (e.g., Chord, Tapestry, and Pastry). Another motivation for overlay networking arises from security challenges. For example, Virtual Private Ad Hoc Networking (VPAN) has been developed to create virtual overlay networks between trusted IP capable devices [35].

In an M2M system, M2M asset devices, network and users can be mobile. In these conditions, information exchange with the M2M asset device need to solve such challenges as unreliable communication channels, temporal presence, limited power and computing capabilities. Solutions for wireless sensors networks in some specific domain and application have been developed [36]. Usually, these solutions have vendor dependent optimized solutions for computing platform, radio technology, communication and services. However, there is at least one potential emerging standard for low power communication between wireless sensors called as Bluetooth 4.0 [37], Bluetooth Smart, which may be applicable for multiple domains. However, there is still a need for generic standard based communication and service protocols working with limited capability embedded M2M assets and mobile gateways.

2.3. Discussion

The M2M standardization approaches demonstrate the heterogeneity of the M2M service networks area. The area is very large; our aim is to review the available technological solutions at least from the following perspectives: M2M Information and services, M2M communication and M2M security. In this review we categorize the available technologies into these perspectives. Within each category, the key contribution of each technology is analyzed. After the review, the potential contributions of the technologies to support horizontal and autonomic M2M are discussed.

3. M2M Information and Services

M2M Information and Service technologies are reviewed in this chapter. After the review, two potential technologies: sensor web enablement (SWE) and ETSI M2M service capability layer are shortly overviewed. Finally, autonomic features and related technologies are discussed.

3.1. M2M Information and Service Technologies

The selected set of M2M information and service level technologies are reviewed in Table 2, and some of them are shortly analyzed in the following.

Table 2. Review of M2M Information and Service Technologies.

Technology	Forum(s), References	Main Contribution
ETSI M2M	ETSI M2M service capability layer [18,23,38]	Specification of a Horizontal M2M service capability layer. RESTful approach to represent information by resources which are structured as a tree. The content of the information is transparent and out of scope.
OMA DM data model	Open Mobile Alliance [39,40]	Information model for management and dynamic service provisioning of OMA DM enabled devices.
BBF TR069 data model	Broadband forum [41,42]	Information model for Auto-configuration and dynamic service provisioning of TR069 enabled customer premise equipment.
UPNP data model	Universal Plug and Play [31,43]	Information model based on XML to describe device and its provided services for UPNP enabled devices.
DPWS data model	Devices Profile for Web Services (DPWS) [44]	Information model defined in the WSDL file based on the XML Schema specification for DPWS enabled devices.
oBix data model	Open building information Xchange [45]	Restful approach based on an object oriented data model to represent data for the mechanical and electrical systems in commercial buildings.
OPC UA data model	OPC Foundation [46,47]	Information model based on object-oriented technics to represent data for Object Linking and Embedding for Process Control (OPC).
OWL-S ontology	Semantic Markup for Web Services [48,49]	Semantic web services ontology that enables users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints.
SSN ontology	W3C Semantic Sensor Networks [50,51]	Ontology that defines the capabilities of sensors and sensor networks, and to develop semantic annotations of a key language used by services based sensor networks.
FIPA-device ontology	FIPA Device Ontology Specification [52,53]	Semantic device profile and service ontology that can be used by agents when communicating.
Sensor Web Enablement	Sensor Web enablement [54–56]	Overlay architecture for integrating sensor networks and applications on the Web.
OSGi	OSGi Alliance [57]	Modular system and service platform for Java.
Service Delivery Framework (SDF)	TM Forum [58]	Providing a reference in the industry on management of "next generation services".

ETSI M2M service capability layer has been defined to be horizontal in the sense that the service capabilities have been specified to be applied with multiple application domains. It uses a resources tree structure to represent the service capabilities layer of each M2M device [18,38]. This tree describes information such as registered SCLs, registered applications, access rights, subscription, groups, data containers, *etc.* ETSI M2M describes only data containers without specifying any information content, which means that information content is transparent and out of scope from ETSI M2M. In addition, device descriptions are not properly included, but interfaces with OMA DM and BBF TR069 device management are specified. Therefore, it is necessary to decide the containers, information content and device management of the devices and applications beforehand.

The devices offering services can be managed by Device Management Servers in OMA DM [39]. OMA DM devices are identified by a model number, and represent services as management objects in the OMA DM data model (Device Management Tree). These service objects are just service representations which contain nodes. These nodes contain properties for the actual service which can be managed by the ACS. The calls to configure the management objects are done over SyncML. Devices offering services can be managed by Auto Configuration Servers in TR-069 [41]. TR-069 devices are identified by a serial number and a type, and represent services as service objects in the TR-069 data model. These service objects are just service representations which contain parameters for the actual service which can be managed by the ACS. Service objects can be classified by profiles which are specified by the DSLForum.

UPnP devices [43] are represented by XML device descriptions, which contain a device ID, the device type and the list of provided services. Each entry in this service list points to an XML service description which contains a service ID, the service type and the actions and states which specify the functionalities of the service. Service description is handled with WSDL file in DPWS [59]. A service's metadata includes information about its relations with other services, such as the hosting service's relationship to the hosted service and vice versa. The application payload data model is defined in the WSDL by means provided by the XML Schema specification, which is an XML based language that can be used to define the properties of other XML based languages.

OBix [45] proposes a restful data model based on a highly extensible full-blown object oriented model. It includes classes and even method signatures and objects composed of other objects. It offers a standard library providing the base object types and special purpose classes such as watch, history, batch operations, *etc*. In classic OPC, only pure data was provided, such as raw sensor values, with only limited semantic information, such as the tag name and the engineering unit. OPC UA [46,47] offers more powerful capability for semantic modeling of data. It uses object-oriented techniques, including type hierarchies and inheritance, to model information. The OPC UA Node model allows for information to be connected in various ways, by allowing for hierarchical and non-hierarchical reference type. This facilitates exposing the same information in many ways, depending on the use case.

OWL-S [48,49] is an OWL-based Web service ontology, which supplies Web service providers with a core set of markup language, constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. OWL-S markup of Web services facilitates the automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation. The SSN ontology [50] describes sensors, the accuracy and capabilities of such sensors, observations and methods used for sensing. Also, concepts for operating and survival ranges are included, as these are often part of a given specification for a sensor, along with its performance within those ranges. Finally, a structure for field deployments is included to describe deployment lifetime and sensing purpose of the deployed macro instrument. Fipa-Device ontology [52,53] can be used as reference to express the capabilities of different devices in a ubiquitous computing system. Some concepts of FIPA are: Device, Hardware Description, Software Description and Connection Description. FIPA can be used by agents when communicating about devices. Agents pass profiles of devices to each other and validate them against the Fipa-Device ontology.

Data transmitted in M2M networks need both, the semantics and the level of abstraction, that could make it possible to provide them as a pool of common data available in a given environment and to

share them between different applications, without needing to know beforehand the specifics of these data. The physical entities that are sensed and acted need to be modeled with a level of abstraction allowing the M2M system to treat them as generic entities, intrinsic to the environment and not tied to a specific M2M application.

Open Geospatial Consortium (OGC) Sensor Web Enablement (OGC SWE) initiative is focused on developing an infrastructure which enables an interoperable usage of sensor resources by enabling their discovery, access, tasking, as well as events and alerting on the Web from the application level. It hides the underlying layers, the network communication details, and heterogeneous sensor hardware, from the applications built on top of it, but does not provide sensor network management functionality [54–56].

3.2. Sensor Web Enablement

The term *Sensor Web* refers to a wireless network of sensor nodes which autonomously share the data they collect and adjust their behavior based on this data [60,61]. Today, it refers in practice into WWW overlay for integrating sensor networks and applications [62–64]. The view of *Sensor Web* has been largely influenced by the architecture that is being developed by OGC SWE work group [56]. The SWE architecture represents a *logical view-point* of the *Sensor Web*, as an infrastructure which enables an interoperable usage of sensor resources by enabling their discovery, access, tasking, as well as events and alerting, in a standardized way [63]. The SWE architecture consists of two main blocks: Interface model and Information model.

The information model, formalized as XML schema documents, consists of the conceptual data models. The interface model (services model), is the specification of services in the form of Web service interface specifications (WSDL). Figures 2 and 3 show the new generation SWE standards (Version 2.0) which were published in the second half of 2011. Green boxes are specifications approved as standards (or in standardization process), grey boxes are discussion papers and dashed arrows point to dependencies.

SWE Common specifies low level data models for exchanging sensor related data between SWE framework nodes. It defines the representation, nature, structure and encoding of sensor related data [65]. Observations & Measurements (O&M) defines a conceptual schema for observations, and for features involved in sampling when making observations. These provide models for the exchange of information describing observation acts and their results, both within and between different scientific and technical communities [66]. SWE services use SensorML as a format for describing their associated sensors [67]. The model provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing of configurable sensor properties. A sensor is defined as a process which is capable of observing a phenomenon and returning an observed value. For example, a process can be a measurement procedure conducted by a sensor or the post processing of previously gathered data. Event Pattern Markup Language (EML) defines subscription filters to SWE framework nodes that provide publish/subscribe type services. Currently, the Sensor Event Service (SES) is the first prototype within OGC SWE that supports EML. Both the O&M and SensorML models extend the OGC's Geographic Markup Language (GML) which is a standard XML encoding for expressing geographical features. This provides the functionality to integrate sensors into Spatial Data Infrastructures (SDI).

Figure 2. Sensor web enablement (SWE) common.

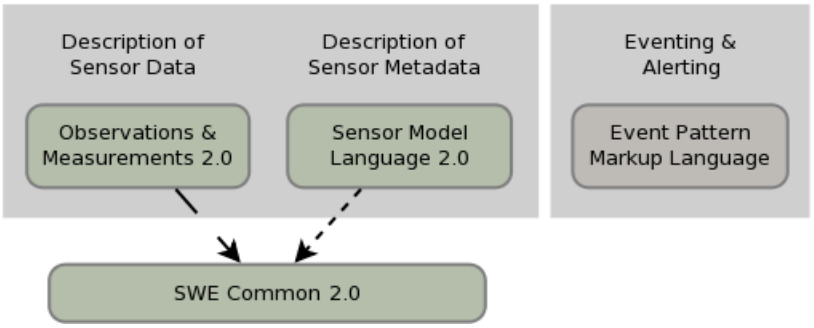
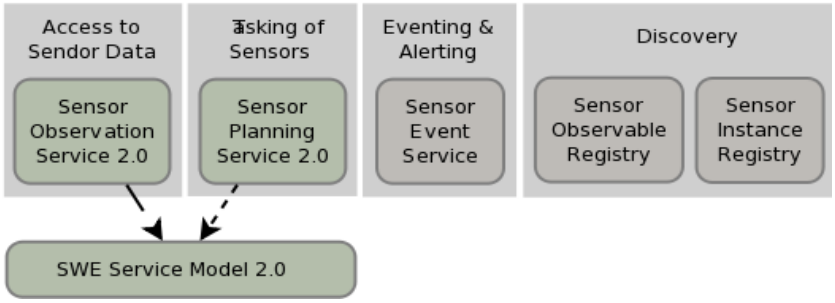


Figure 3. SWE Service Model.



SWE Service Model defines packages with data types and operation request and response types for common use across SWE services [68]. Sensor Observation Service (SOS) provides standardized access to sensor observations and sensor metadata [69]. The service acts as a mediator between a client and a sensor data archive or a real-time sensor system. The heterogeneous communication protocols and data formats of the associated sensors are hidden by the standardized interface of SOS. SOS defines O&M as its mandatory and default response format for sensor data and recommends the usage of SensorML for sensor metadata exchange. Sensor Planning Service (SPS) is a web service interface that allows clients to submit tasks to sensors for operations such as the configuration of the sampling rate or the steering of a movable sensor platform [70]. SPS operations derive from the abstract request type defined by SWE Service Model and aggregates operations covering the complete process of controlling and planning sensor tasks.

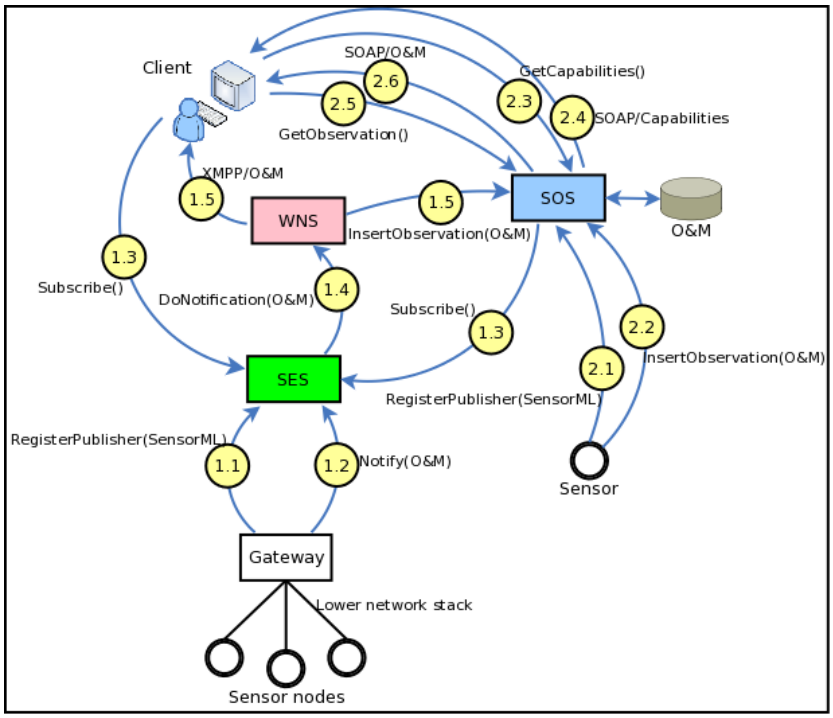
Sensor Event Service (SES) is a SWE service specification at discussion phase. It provides a push-based access to sensor data as well as functionality for sensor data mining and fusion inside a spatial data structure. SES uses the OASIS WS-Notification (WS-N) standard for the definition of the service operations needed for a publish/subscribe communication. This suite of standards defines operations for subscription handling and notifications (WS-BaseNotification), for the brokering of notifications (WS-BrokeredNotification) and for the use of event channels (WS-Topics). Event channels allow a grouping of notifications with respect to a specific topic, for instance weather forecasts. Instead of defining the filter for forecasts in each consumer's subscription, a consumer can simply subscribe for all notifications on the weather channel. Consumers may also use filter criteria while subscribing to sensor data. Depending on the filter type used these criteria operate on single observations or on observation streams, potentially aggregating observations into higher-level information (which itself can be regarded as observation data).

Sensor Instance Registry (SIR) provides functionality to collect, manages, transform and transfer sensor metadata through SensorML as well as extended discovery functionality and sensor status information. It is intended to close the gap between the SensorML based SWE world and conventional Spatial Data Infrastructures (SDIs). In the future it is expected that the functionality of the SIR interface will partly be covered by other existing SWE services (e.g., SOS for retrieving sensor status information and the SES for filtering sensor status updates).

Sensor Observable Registry (SOR) is designed in order to support users when dealing with identifiers pointing to phenomenon observed by a sensor (observable) which is very important when searching for sensors. Usually such parameters are expressed within SensorML documents through some kind of identifier (*i.e.*, URIs). Figure 4 depicts two scenarios where the SWE architecture is in action: sensors without embedded IP stack, and sensors with embedded IP stack. The scenarios have been simplified in order to clarify the basic idea. More details can be found in the standards specifications and the SWE white papers.

SWE is a standard M2M architecture that focuses primarily on the requirements of the scientific community dealing with remote sensor and sensor data management. Certain features of the SWE architecture pose a barrier for its widespread adoption for this purpose. For example, the complexity of the SOAP based SWE service interface specification and other extension specifications (e.g., OASIS WS-Notification, *etc.*) are one of these barriers. Future versions add a lightweight HTTP GET binding for certain operations with the aim of improving ease of adoption. Another issue is the tight coupling of the information model with GML which binds spatial data infrastructure across all data ubiquitously. While this feature makes it easy to relate sensor data with other spatial-temporal resources (e.g., maps, raster as well as vector data) at the application level, this is not a requirement in all M2M scenarios. However, adaptation of the SWE information model may provide useful solutions within the context of M2M service networks.

Figure 4. SWE Architecture in action.



Scenario 1—Sensors without Embedded IP Stack

1.1 Multiple sensor nodes connect to a SWE Gateway. Sensor networks that utilize various network topologies and low level communication protocols may be connected to the SWE architecture in this manner. The Gateway layers the sensor network data with Internet and Web protocols and relays it to a SES server. Initially the gateway calls the RegisterPublisher operation to add sensors to the SES. The request includes a SensorML which describes the sensor and its capabilities.

1.2 Sensor data is published to the SES via the Gateway. Notify operation, which contains sensor data in the O&M standard, is used.

1.3 A SOS server and a user application subscribe to sensor data using the Subscribe operation. The requests include the topic and the filtering specifications, as well as the information about the endpoint to which the filtered sensor data should be delivered. The step would probably be preceded by GetCapabilities and DescribeSensor operations which allow the consumers to collect information about the available sensor data that is being brokered by the SES. Note that while the SOS acts as a consumer in this interaction it is in fact a broker of sensor data. In SWE any service, application or process may act as a consumer as long as it provides an endpoint where notifications can be delivered to. Thus, even chaining of information brokers is possible.

1.4 SES forwards the sensor data that conforms to the subscription filters; to the WNS by invoking the DoNotification operation. The request includes the id of the delivery endpoint account.

1.5 WNS delivers the sensor data to the intended consumers. It must be noted that to receive notifications consumers have to be registered to the WNS beforehand. In this scenario the SOS consumer receives the data via HTTP POST while the user application receives it via XMPP.

Scenario 2—Sensors with Embedded IP Stack

2.1 A sensor registers itself to an SOS server by invoking the RegisterSensor operation. The request includes a SensorML document that describes the sensor capabilities. In this scenario the sensor system includes an embedded IP stack that allows it to communicate directly with the SOS server over the Internet.

2.2 The sensor sends streaming observation data to the SOS by invoking its InsertObservation operation. The data is encoded in O&M format and it is persisted on the SOS backend.

2.3 A SWE client invokes the GetCapabilities operation of the SOS. This operation allows clients to retrieve service metadata about a specific service instance.

2.4 SOS responds with the response with an XML encoded document. This document provides clients with service Filter Capabilities which indicate what types of query parameters are supported by the service and Contents Section in the form of Observation Offerings. These are sets of sensor observation groupings that can be queried using the GetObservation operation. Note that the response will not only include metadata that describe observations collected from the sensor mentioned in this scenario but also the observations received from the sensor nodes in the previous scenario.

2.5 Based on the information gained in the previous steps the client requests sensor observations by invoking the GetObservation operation. The request parameters are based on the Filter Capabilities information received in the previous GetCapabilities request. This may be a one-off operation or the client may poll the SOS periodically.

2.6 The response to a GetObservation request is an O&M Observation, an element in the Observation substitution group, or an ObservationCollection. These are historical sensor observations retrieved from the SOS backend.

The SWE Common and O&M which are data models for exchanging heterogeneous sensor data at low and higher levels respectively allow applications and/or servers to structure, encode and transmit data in a self-describing and semantically enabled way. O&M, which is based on Martin Fowler's Observation & Measurements analysis pattern [71], is a dynamic model that can adapt smoothly to new requirements without necessarily needing to change implementation or the way objects interact. It is therefore, a particularly interesting solution for the description of device data across different business domains.

One of the key requirements of M2M systems is the ability to fuse, interpret or transform device data into higher level information, required by various M2M applications. A mechanism that enables such on-demand processing, using generic software that utilizes standard process descriptors, simplifies M2M application development and facilitates interoperability. The view-point of the sensor as a process or a chain of processes in the SensorML model with detailed process input, output, parameter, and method descriptions makes it a possible candidate for such a generic process descriptor.

The ongoing work on SensorML 2.0 is planning additional functionalities such as the Sensor Interface Descriptor (SID) extension. SID enables the precise description of a sensor's protocol and interface. This work may be used as a basis to develop generic device drivers carrying protocol definitions of legacy devices which in turn would allow on-the-fly, plug-and-play integration of such non-compliant devices into the standard M2M service network without the need for manual ad-hoc development for each device type [3].

3.3. ETSI M2M Service Capability Layer

ETSI provides an M2M network architecture specification with a generic set of service capabilities. The idea of the architecture is to provide horizontal service capability layer that can be applied to several vertical M2M application domains with standardized reference points. The ETSI M2M Network is composed of two domains: (1) The Device and Gateway domain with M2M devices and Gateways and (2) the Network domain with core network access, M2M service capabilities, M2M applications and management functions, see Figures 5 and 6.

Three different Service Capability Layers are specified: a Device Service Capability Layer, a Gateway Service Capability Layer and a Network Service Capability Layer. Each SCL exposes the services required by the M2M applications residing in the underlying M2M Network. The ETSI adopted a RESTful [72] architecture style for the M2M Applications and/or M2M SCL information exchange [18].

Each SCL is comprised of several services groups: Application Enablement Capability (AEC) for providing M2M point of contact for using the M2M applications of the corresponding SCL, Generic Communication Capability (GCC) for interfacing between the different SCL available on a given M2M Network, Reachability, Addressing and Repository Capability (RARC) for managing events relative subscriptions and notifications as well as for handling applications registration data and information storage, Communication Selection Capability (CSC) for network selection and alternative

communication service selection after a communication failure, Remote Entity Management Capability (REM) for remote provisioning, Security Capability (SECC), History and Data Retention for archiving data (HDR), and Interworking Proxy (IP) for integrating non ETSI compliant systems.

Figure 5. M2M system with ETSI M2M standard reference points.

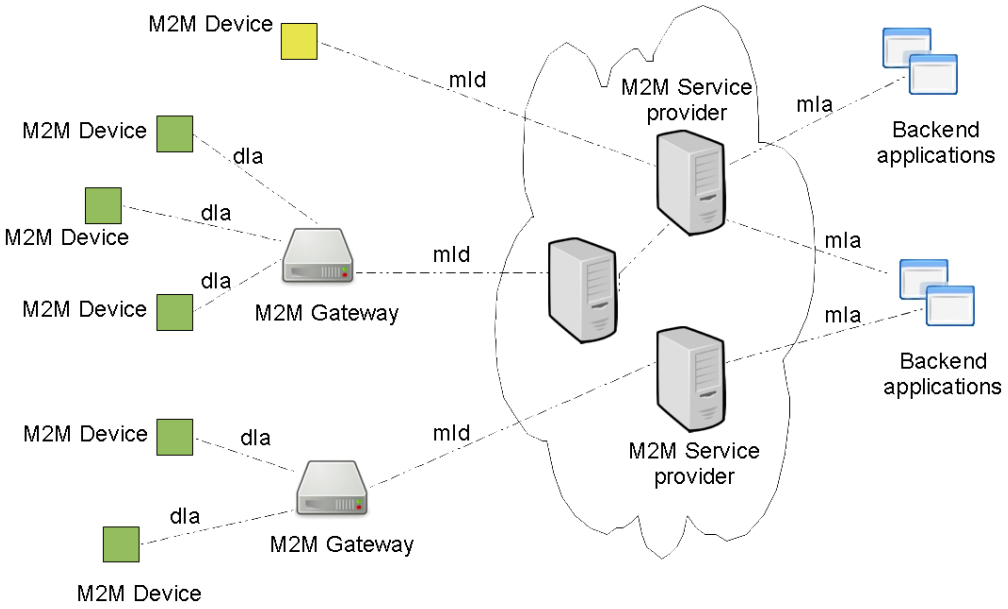
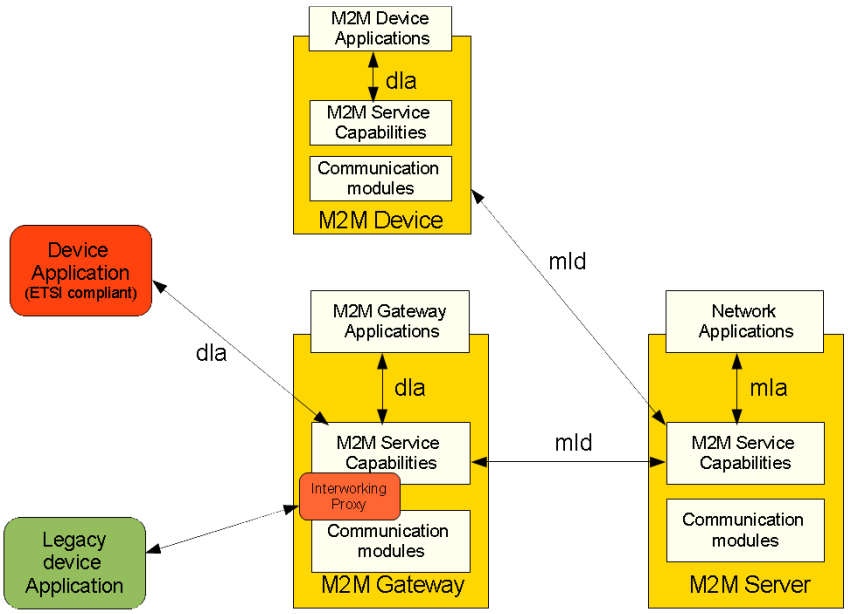


Figure 6. Interfaces of ETSI M2M system.



The presented service capabilities are logical groups of functions identified by ETSI, but all of them are not mandatory [18]. Only the external reference points, dIa, mId and mIa, between the M2M applications and SCLs are mandatory. Where the dIa interfaces devices applications and Gateway or Device Service Capability Layer, the mId interfaces the M2M Gateway or Device Service Capability Layer and the M2M Network Service Capability Layer and the mIa interfaces backend M2M Applications and the M2M Network Service Capability Layer. These interfaces aim to be applicable to a wide range of network technology and application and access independent [35].

The ETSI M2M architecture indicates that it does not explicitly specify means for integrating legacy devices or other already existing non-ETSI compliant systems, but it presents integration points (Interworking Proxy) on the Gateway and Network Service Capability Layers [15]. Figure 7 presents the different M2M Service Capability Layers that comprise the ETSI M2M Network as well as the related reference points.

According to the ETSI M2M architecture, an M2M Gateway can act as a proxy for M2M devices available in the same local area network. Once M2M devices applications are registered on a given GSCL, they become available to the registered SCLs and M2M applications if they acquire the appropriate access rights. Ex: Network applications can subscribe to information produced by a sensor (Device application) registered on a reachable GSCL. Figure 8 explains how this can be done.

We presume that the GSCL is registered to the NSCL and that the Device Application is registered to the GSCL. First the Network application registers to the NSCL (001). Then Network Application Enablement capability checks if the issuer is authorized to be registered and treats the request (002). The registration information is then stored by the Network Reachability, Addressing and Repository Capability. After it Network Application receives a positive answer (003). The Network Application subscribes to the data produced by the desired sensor (Device application) (004). This data can also follow a certain criteria specified by the issue. If the issuer is authorized to subscribe to the given Device Application, the Gateway Reachability, Addressing and Reachability capability treat the request (005). Network application receives a positive response (006). Device application sends information to the Gateway (007). Then Gateway Reachability, Addressing and Repository Capability identify an event that needs to be reported to a subscriber (008). Finally, The GRAR Capability notifies the subscribers (009).

Figure 7. Architecture of ETSI M2M Service Capabilities.

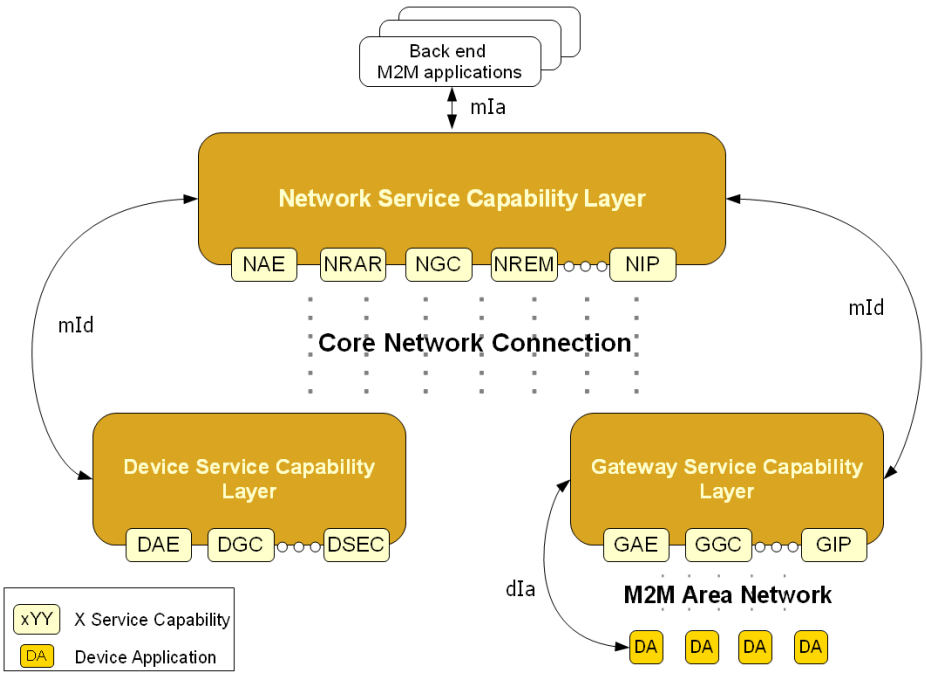
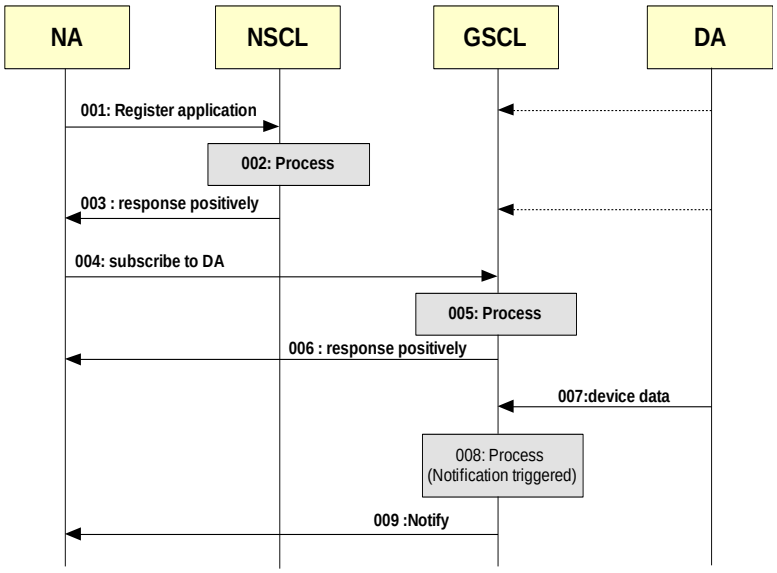


Figure 8. Sequence diagram for a network application subscription to device data.



3.4. Autonomic M2M Services

Autonomic computing has been inspired from biological systems including both centralized and decentralized approaches, which have been proposed to enable autonomic behavior in computer systems. The centralized approach focuses on the role of human nervous system as a controller to regulate and maintain the other systems in body [73]. Decentralized approaches take inspiration from local and global behavior of biological cell and ant colony networks [74]. Another area focuses on developing self-managed large scale distributed systems and databases. One such project is Oceano [75], a project of IBM providing a pilot prototype of a scalable, manageable infrastructure for a large scale utility power plant. OceanStore [76], a project of University of California Berkeley describes a global persistent data store for scaling billion of users. IBM's SMART [77] offers a solution to reduce complexity and improve quality of service through the advancement of self-managing capabilities. Similarly, Microsoft's AutoAdmin [78] makes database system self-tuning and self-administering by enabling them to track the usage of their system and gracefully adapt to application requirements. Agent architecture has been frequently used to develop infrastructures supporting autonomic behavior based on a decentralized approach. Unity [79] makes use of this approach to achieve goal-driven self-assembly, self-healing and real-time self-optimization. Similarly, Autonomia [80] is a University of Arizona project providing the application developers with all the tools required to specify the appropriate control and management schemes to maintain any quality of service requirements.

Component based frameworks have been suggested for enabling autonomic behavior in systems. The Accord framework [81] facilitates development and management of autonomic applications in grid computing environments. It provides self-configuration by separating component behavior from component interaction. Selfware [82] models the managed environment as a component based software architecture which provides means to configure and reconfigure the environment and implements control loops which link probes to reconfiguration services and implement autonomic behaviors. A number of frameworks make use of techniques such as artificial intelligence (AI) and

control theory. In [83] an AI planning framework based on the predicate model has been presented. It achieves user defined abstract goals by taking into account current context and security policies. Another area focuses on modifying control theory technique [84] and presents a framework to address resource management problem. A mathematical model is used for forecasting over a limited time horizon. Service Oriented Architecture has been also used to design architecture with autonomic behavior. In [85], autonomic web services implement the monitoring, analysis, planning and execution life cycle. The autonomic web service uses log file provided by a defective functional service to diagnose the problem and consults the policy database to apply appropriate recovery action. Infrastructures have been proposed to inject autonomic behavior in legacy and non-autonomic system, where design and code information is unavailable [86]. Some of the identified frameworks make use of layered architecture [87], case based reasoning (CBR) [88], and a rule driven approach [89] to enable automaticity in existing systems.

Different techniques can be used to achieve the capabilities described in each of the four major self-* properties. There are more system specific techniques to achieve self-configuration such as hot swapping [90] and data clustering [91], and techniques that make use of more generic approaches such as machine learning, ABLE [92] and case-based reasoning [93]. Many applications that require continuous performance improvement and resource management make use of control theory approach demonstrated in the Lotus Notes application [94], learning based self-optimization techniques such as LEO [95], and active learning based approach [96] based on building statistical predictive models. Self-healing is the property that involves problem detection, diagnosis and repair. CBR [93] performs problem detection in the analysis phase, finds a solution in planning phase and performs problem repair in execution phase. Hybrid approach [97] to create a survivability model provides more robust survivability services. Active probing and Bayesian network [98] is a problem diagnosis technique that allows probes to be selected and sent on demand. The heartbeat failure detection algorithm [99] enables detecting problem when the monitor experiences a delay in receiving of message. Another technique to detect problems in distributed systems is to use tools like Pip [100] that expose structural errors and performance problems by comparing actual system behavior and expected system behavior. Performance Query Systems (PQS) [101] can be used to enable user space monitoring of servers. Component level rebooting is a technique that recovers from defects, without disturbing the rest of the application as shown in Microreboot [102]. Another safe technique to quickly recover programs from deterministic and non-deterministic bugs is demonstrated in Rx [103]. It rolls back the program to a recent checkpoint upon a software failure, and then re-executes the program in a modified environment. Some notable techniques that enable systems to enable self-protection make use of component models such as Jade [104] and self-certifying alerts as demonstrated in Vigilante [105].

4. M2M Communication

M2M Communication technologies are reviewed in this chapter. Afterwards, a set of potential technologies is shortly overviewed: Constrained Application Protocol (CoAP), EXtensible Messaging and Presence Protocol (XMPP), 6LowPAN and RPL, and Bluetooth Smart. Finally, autonomic features and communication technologies are discussed.

4.1. Overview on M2M Communication Technologies

A set of M2M communication technologies is reviewed in Table 3. M2M applications impose several new constraints on communication solutions. They generate a type of traffic which is most likely to be comparatively small but with a larger number of involved devices. Even if the messages are usually quite short, in most cases there are strong requirements on the reliability and delays. In addition, the embedded asset devices may have computing and power constraints, which challenge the traditional communication protocols.

Thus reliable and delay aware transfer of messages over the network is very important for M2M applications. The hypertext transfer protocol (HTTP) is usually applied to transfer messages in client-server manner in World Wide Web. There are protocols such as SMTP, POP and IMAP for electronic mail systems. In addition, protocols like XMPP and SIP/SIMPLE have enabled capabilities such as instant messaging and presence to be applied in more real-time communication. XMPP uses decentralized client-server architecture to keep clients simple, and pushes most of the complexity into the servers [106]. The architecture is different than WWW in the sense that it supports inter-domain connections called federations. In addition, email network uses multiple hops between servers to deliver messages but the XMPP architecture uses direct connections. When taking the communication requirements set by M2M applications into concern, it is seen that XMPP type of architecture has succeeded to enable real-time messaging with support for simple clients.

Table 3. Review of the M2M Communication technologies.

Technology	**Forum(s), References**	**Main Contribution**
Sensor-Over-XMPP	XMPP Extension (protoXEP) [107]	a payload format for communicating sensor and actuation information
Data Distribution Service for Real-Time Systems (DDS)	Object Management Group (OMG)	Scalable, real-time
Advanced Message Queuing Protocol (AMQP)	OASIS AMQP standard	Broker-based messaging, publish-subscribe
STOMP	STOMP	Simple broker-based text-based protocol
MQ Telemetry Transport (MQTT)	MQTT Eclipse M2M Industry Working Group	Lightweight publish/subscribe binary messaging protocol
ZeroMQ	ZeroMQ protocol [108]	An openly published simple and lightweight publish-subscribe type of messaging protocol designed for constrained devices and low-bandwidth, high-latency or unreliable networks.
COAP usage for REsource LOcation And Discovery (RELOAD)	Proposed as IETF Internet-draft	peer-to-peer federation of geographically distributed WSN islands
Content-Centric Network	CCNx	Content-based networking. Security by design.
Websocket protocol	IETF RFC 6455	two-way communications with Low-overhead transport (single TCP connection)
CoAP	IETF's Constrained RESTful environments (CoRE) working group	An application layer protocol designed for constrained devices allowing them to communicate over the Internet.

Table 3. *Cont.*

		Wired	Radio	Full IP
Cellular	3GPP	No	Yes	Yes
Bluetooth	Bluetooth Special Interest Group	No	Yes	No
Bluetooth LE	Bluetooth Special Interest Group	No	Yes	No
Ant/Ant+				
ZigBee	ZigBee Alliance (and IEEE)	No	Yes	No
ZigBee IP	ZigBee Alliance (and IEEE and IETF)	No	Yes	Yes
6LowPan	IETF	No	Yes	Yes
Z-Wave	Z-Wave alliance	No	Yes	No
WiFi low power	IEEE	No	Yes	Yes
Modbus	Modicon (society)	Yes	No	No
BacNet	ASHRAE	Yes	No	No
LonWorks	ANSI	Yes	No	No
KNX	ISO/IEC	Yes	Yes	No

The websocket protocol (a part of HTML5 initiative) which relies on HTTP for handshake and negotiation, is message-based and has been designed to allow bidirectional communications. The related message queues protocols can be broker-based (e.g., DDS, AMQP, and STOMP) or broker less (e.g., ZeroMQ) and allow asynchronous communications and operate at the same level as HTTP. MQTT is a message queue designed with M2M applications in mind to enable lightweight publish/subscribe messaging transport.

A challenge is related to application of HTTP within constrained local M2M asset network. To solve this problem, IETF CoRE (Constrained RESTful Environments) working group has specified Constrained Application Protocol (CoAP) standard with the goal of supporting REST-like applications in constrained environments. The second challenge is related to the sizes of IP packets and headers. To solve this problem, IETF has created the 6lowpan (IPv6 Low Power wireless Area Networks), which describes an adaptation layer between IPv6 and a layer 2 protocol, such as (but not limited to) IEEE 802.15.4, to handle MTU sizes and compress IPv6 headers from 60 bytes to 7 bytes. There are also other challenges arising from heterogeneity of M2M devices and local M2M asset networks, and coding and integration of M2M application content.

4.2. Low Power and Lossy Networks

A Low Power and Lossy Network (LLN) is a network between constrained embedded devices, in which the communication links may change frequently, or even disappear. In order to face the challenges in those networks, the IETF has set working groups to elaborate standards which guarantee that such an infrastructure is scalable, secure and reliable in terms of communications between each device. One of the most important aspects of these standardization activities is the use of the IP layer as the reference layer. Using this reference, new protocols have been developed for routing, transport, and applications in LLNs. IPv6 brings some outstanding benefits such as an addressing scheme which

allows identifying billions of devices and supporting point-to-point communications between a device and a PC connected to Internet. However, the IPv6 protocol is inadequate for LLNs in terms of network overhead. As a result, the IETF 6LowPAN WG [109] proposes adaptations of the IPv6 protocol when the underlying network is constrained. For example, standards have been proposed for the transmission of compressed IPv6 packets over IEEE 802.15.4 networks [110]. IPv6 and 6LowPAN network stacks are natively available on common operating systems for embedded devices (e.g., Contiki and TinyOS), therefore making them able to communicate with both Internet and LLNs devices.

Another aspect or LLNs is the strong constraints on routing protocols, which must be different from those used in traditional IP networks. First of all, link conditions may change frequently during time; therefore a routing protocol must react quickly to these changes. Second, the nodes have really strong storage constraints; therefore, a routing protocol should work even if a node has not stored all the routes to each of the other nodes in the network. Third, since the nodes have severe energy constraints, the exchange of control messages should be kept as low as possible.

One solution to the above-mentioned LLNs limitations is provided by the RPL routing protocol. It has been developed to have really limited control traffic, to fit harsh and constrained environments, with limited data rate and potentially elevated error rate. RPL is a distance-vector protocol based on the creation of a routing tree, referred to as Destination Oriented Acyclic Directed Graph (DODAG), where the cost of each path is evaluated according to the metrics defined in an objective function. The goal of this protocol is the creation of a collection tree protocol, as well as a point-to-multipoint network from the root of the network to the devices inside the LLN.

In order to keep the status of the network updated, the root of the RPL tree sends periodical messages, referred to as DODAG Information Object (DIO). The receiving nodes may relay this message or just consume it, if configured as leaves of the tree. The RPL protocol also introduces a *trickle* mechanism which allows reducing the transmission frequency of DIO messages according to the stability of the network. In addition, RPL offers several advanced functionalities, the detection of loops in the routes, and the management of local faults (via local or global repair).

4.3. Constrained Application Protocol (CoAP)

The IETF CoRE (Constrained RESTful Environments) working group [111] has defined the Constrained Application Protocol (CoAP) standard with the goal of supporting REST-like applications inside constrained environments like those identified by the RoLL [112] and 6lowPAN working groups. Application domains include LLNs and more generally M2M communications, and span over a large range of business use cases such as smart energy or building automation. The specification defines a binary message structure between CoAP endpoints as well as the interaction protocol. By following REST architectural principles [71], CoAP exposes a representation of the information available on a constrained device as a set of identifiable resources. This way, any CoAP endpoint may interact with it remotely using the interaction methods used by the HTTP protocol: GET, POST, PUT, and DELETE.

In order to make the resources discoverable, the CoAP protocol standard advises to expose CoAP endpoint's resource metadata using the CoRE Link Format [113] at a specific URI. CoAP messages rely on the UDP transport protocol between endpoints. This is to accommodate the potentially

unreliable and lossy wireless environments that render the TCP protocol inefficient in terms of network resource usage. In order to meet eventual QoS requirements, since UDP is natively unreliable, CoAP has introduced the use of confirmation messages, which correspond to an acknowledgement that a CoAP message has been received.

Collection of data from a CoAP-enabled device is achieved by sending a CoAP request message (GET method) to the CoAP server hosted on the device: as soon as the CoAP server receives such a request, it replies with a CoAP response with data requested by the CoAP client or notifies that the response will be sent in a separate response. Another interaction scheme supported by the CoAP protocol is the publish/subscribe paradigm. Instead of sending periodical requests to a CoAP server to be kept updated on the status of a resource, the CoAP client may subscribe, through specific exposed end-points, to a CoAP server, which will be in charge of periodical updating all the subscribed clients of the status of a given resource.

RESTful architectures make caching of the data possible within the network. Caching is supported by CoAP and makes it possible to optimize the data delivery over potentially constrained wireless links. For each CoAP observed value a lifetime is defined; if two consecutive requests are received by a CoAP server or proxy in a period of time smaller than that defined by the lifetime parameter, the former request will be sent querying the resource, whereas the latter will be served using the cached value. Using caching, some optimizations can be easily foreseeable for M2M communications. By serving fresh information from a cache instead of querying the endpoint itself, one could experience a shorter delay or a better QoS on a particular request. Also, caching may help reduce the overall consumption of an energy-constrained network by reducing the number of wireless transmissions required for collecting data.

A low-power version of CoAP has been implemented for Contiki [114]. The implementation leverages the ContikiMAC low-power duty cycling mechanism to provide power efficiency. Based on the results of the CoAP request/response, cycles are most energy-efficient when each message fits into a single 802.15.4 frame. When messages are bigger than frames, the interoperation of information models, data encoding/decoding, and segmentation/reassembly with constrained M2M capillary networks and M2M asset devices need to be carefully considered. HC (HTTP/CoAP) proxies provide the interworking functionalities for application spanning across LLNs (potentially running CoAP/UDP/IPv6/IEEE 802.15.4 protocol stack) and the Internet (HTTP/TCP/IPv6).

CoAP base specifications identify DTLS [115] and IPsec [116] as mechanisms to offer data origin authentication, integrity and replay protection, and encryption for the CoAP messages. In addition to these, an alternative [117] to IPsec and DTLS has been presented.

4.4. Extensible Messaging and Presence Protocol (XMPP)

XMPP has been developed to enable message oriented communication services applicable in the Internet context on top of TCP/IP. The XMPP communication architecture is based on distributed client-server model; however, also client-to-client (peer to peer) communication is enabled. The core services of XMPP includes support for presence information, secure messaging (TLS), overlay communication over IP, near real-time messaging, authentication, contact list management, and service discovery. Each XMPP client has an account hosted by a XMPP server, and the client can be addresses

by unique Jabber ID (JID). XMPP JID contains three parts: user, domain and resource as shown in Table 4, RFC 6122/RFC3921 [118]. In XMPP, network domain-part of JID must be a fully qualified domain name or IP address. Each domain presents one logical groups with one user account database. Each domain may present own user account policies. The device owner is usually considered to be also a user of the M2M domain and an owner of at least one XMPP user id. All devices share same user id (local-part and domain-part) with their owner. Separation of devices is done by examining the resource part of JID.

Table 4. An example of Extensible Messaging and Presence Protocol (XMPP) Jabber ID.

JID: Matt@home.com/heating-regulator		
Local-part	Domain-part	Resource
Matt	home.com	heating-regulator

A client connects to the server to send and receive messages, the servers routes messages to the others enabling client-to-client communications over multiple servers, Figure 9. The procedure for M2M client establishing connection with the XMPP server is shown in Figure 10. Discovery can be done directly using a domain part as a server address or discovering server address with SRV lookup from DNS. All clients connect to only their own server specified by JID. Connections are persistent XML streams over TCP and optionally encrypted by Transport Layer Security (TLS) layer. Encryption of TCP stream is strongly recommended, but not required. An administrator of a domain may specify that encryption is mandatory and it is up to administrator or designer to choose whether TLS certificates should be checked. Most client libraries accepts self-signed certificates, this should be taken into account when considering security aspects of client-to-server connections. The availability of each client can be detected with the aid of presence messages. Presence information is shared only with XMPP users that are in the roster/address book of the client sending the presence information.

Server to server connection is an XML stream over a TCP connection, similar to client to server connections. The most important difference is that server to server connections are not authenticated because they happen in between different domains that do not share a common user database. XMPP servers may use XMPP Dial back (XEP-0220), to verify the domain of the connecting server.

The domain administrator may require stronger identification verification by using TLS certificates and Simple Authentication and Security Layer (SASL). When M2M clients located in different domains would like to exchange messages, routing of messages will be done by the domain specific servers, see Figure 11. Then, a communication link between servers of the domains is negotiated to enable messaging between the referred M2M clients.

The capability for handing presence information has been developed for XMPP. The availability of each client can be detected with the aid of presence messages. XMPP offers mechanisms to select how, when and to whom presence information is shared to, as described in requirements listed in [119]. For example, presence can be shared only with contacts that are in roster/address book of client sending the presence information. The related presence information (JIDs) is critical information in the system and it is stored in the roster database. Another useful feature is the publish-subscribe model (XEP0060), which can be used by XMPP entities to subscribe information of the presence of other entities, and receive notifications accordingly.

Figure 9. XMPP architecture.

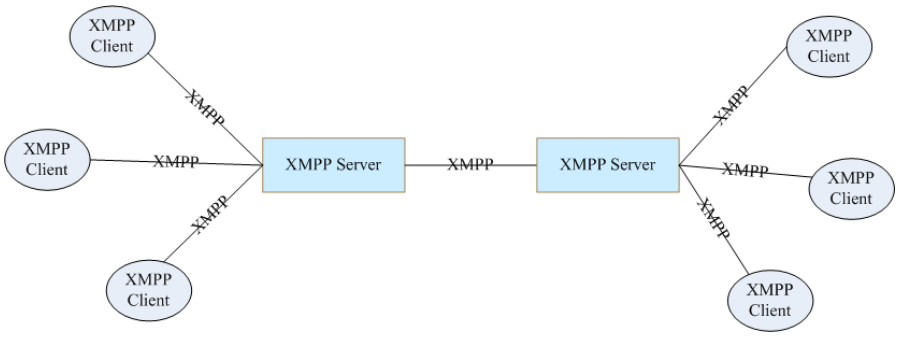


Figure 10. M2M client connecting to XMPP Server.

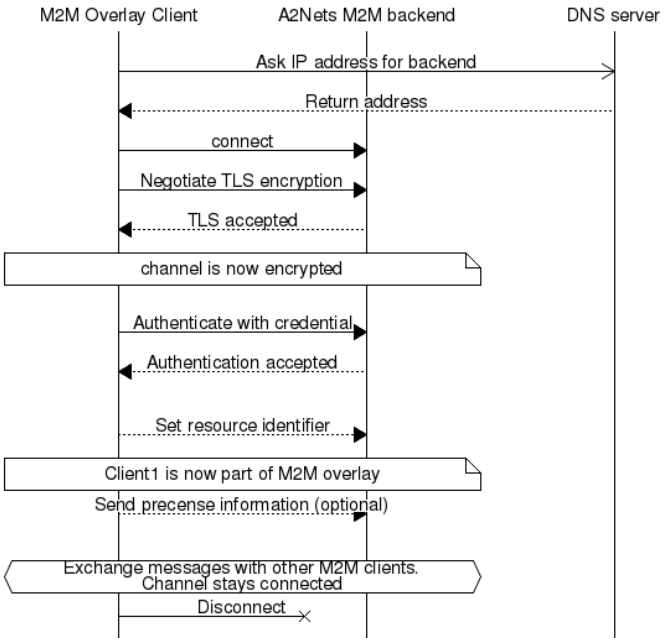
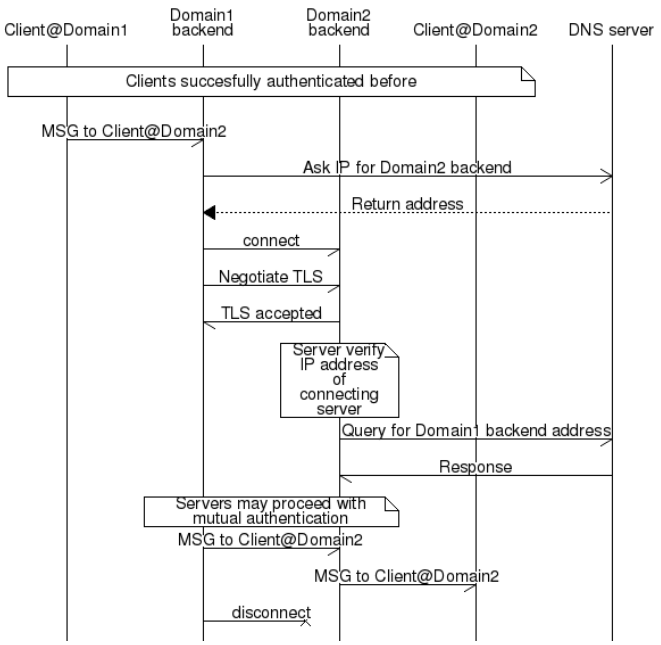


Figure 11. Server to server interdomain communication.



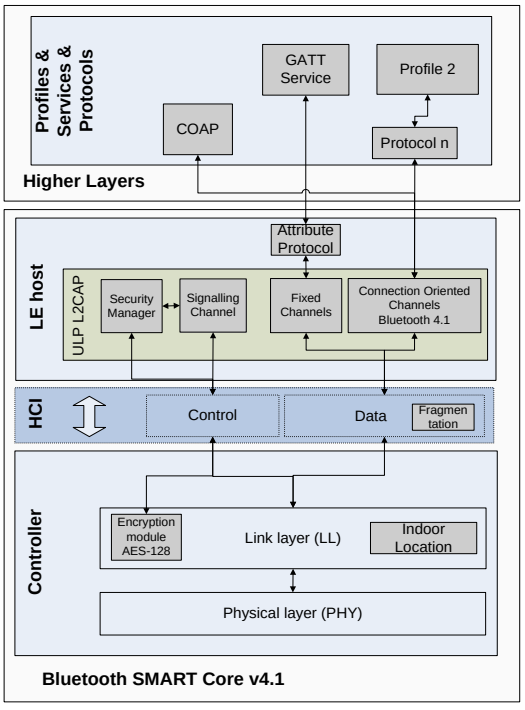
4.5. Bluetooth Low Energy

Bluetooth SMART v. 4.0 [120] is a PAN technology that was introduced to the market by the Bluetooth Special Interest Group in 2010. Bluetooth SMART consists of two distinctly different transports called Basic and Enhanced data rate (BR/EDR) and Low energy (LE). The following focuses briefly on the LE technology and use for web applications in battery operated sensor applications.

LE supports two modes of connection: Non-connected, unidirectional advertisements (broadcast, unicast and scan support), and Connected, bidirectional and reliable (maximum application throughput is 270 kbit/s). The network topology for connection is star/bus where the collector device is typically a master and sensors are slaves. LE uses three channels for advertisements and 37 channels for data. One reason for having the master role assigned to the collector is that collocation of multiple radios in a handset requires operation between all radios. The Bluetooth co-existence controller is specified in Core specification addendum 4 (CSA4) [121] and it is particularly important in combination with 4G networks as LTE TDD. LE also improves co-existence and robustness with nearby networks by using adaptive frequency hopping (AFH).

The main layers of Bluetooth are shown in the following Figure 12 and consist of the following components: Radio and link layer (LL) with an AES-128 bit encryption unit, Multiplexer—Logical link control and adaption layer protocol (L2CAP) providing fixed and connection oriented channels together with fragmentation and reassembly (FAR), Security Manager (SM), Host Controller Interface (HCI)—Connects application processor and Bluetooth controller, The General Access profile (GAP)—Contains a collection of standard procedures, and Generic attribute profile (GATT), which provides an interoperable framework with service discovery and operation. Bluetooth SIG defined service data is characterized by 16 bit universally unique identifiers (uuids) while proprietary extensions use randomized 128 uuids.

Figure 12. The Bluetooth SMART core 4.0.



All packets in LE have the same layout and the maximum length of a packet is 47 Octets. Depending on mode of connection the payload may be max 31 Bytes, for an advertisement, implying an efficiency of 31 Bytes/47 Bytes * 100% = 66%. A connection can have max 27 Bytes per packet implying an efficiency of 27 Bytes/47 Bytes * 100% = 57% for a raw L2CAP channel without encryption. The efficiency drops to 20 Bytes/41 Bytes * 100 = 49% when using GATT with notification and encryption.

4.6. Autonomic M2M Communication

Several projects have targeted autonomic network and communications. Some examples are DASADA [122], Autonomia [80], ACCORD [123], ANA [124] and the IoT-A [125]. The University of Bologna is building a framework [126] to support the design, implementation and evaluation of peer-to-peer Internet applications using Swarm Intelligence; the University College London is working on different projects inspired on bio-approaches for autonomous configuration of distributed systems. All those projects and research groups treat the Internet as an ecosystem and study the relations between the different network elements from a cross-disciplinary perspective. The main goals of every approach are to achieve efficiency in a self-CHOP manner in a large-scale heterogeneous communication infrastructure. The autonomic communication final objective is to provide an evolving network platform for communication between devices (including decision making and reacting) with a high degree of autonomy to decrease to the minimum the human intervention providing a high level of efficiency for both users and telecommunication industry.

5. M2M Security

The emergence of the M2M ecosystem has led to the revision of the conventional network security paradigm [127]. The M2M communication systems, by essence, interconnect heterogeneous network segments, where the heterogeneity is expressed in terms of functional capabilities and capacities. This encompasses, for instance, the interconnection of a WAN with low-power network segments like WSNs in multihop communication schemes. The security of communications involving several hops is usually addressed using hop by hop security where each segment of the communication path from source to destination is secured using distinct credentials possibly managed by distinct parties involving rekeying operations at each transmission node. Credential management with this model may be complex. Furthermore, in order to provide reasonable security, the various parties involved in credentials distribution and management needs to trust each other. This requirement is often difficult to achieve in the case of M2M communications and it is generally accepted that end to end security involving the use of a single set of credential from source to destination is a better model. However the deployment of end to end security is a challenging task as it requires solving the problem of credential distribution at a global level.

A typical network security service (e.g., authentication, access control, data confidentiality, *etc.*) is made up of two phases: bootstrapping and enforcement. Providing an end-to-end security service in M2M requires thus exploring methods for both phases, where the said methods should take into account both the interoperability and trust issues.

Security service bootstrapping comprises authentication and key agreement methods needed for the security enforcement phase. Therefore, given the interoperability and trust challenges in M2M systems, a flexible and adaptive bootstrapping phase is needed in order to guarantee the enforcement of an efficient end-to-end security service.

M2M communications are commonly classified into three main communication domains: M2M device, network and application domains. Those three domains are associated to three distinct security domains involving generally distinct business actors to perform credential management.

Thus, the device domain corresponds to M2M capillary communications occurring in a LAN or PAN proximity network. Generally the owner of this network has the responsibility to manage the credentials used to secure those communications occurring between the devices and the gateway.

In some cases there is a need to secure the connection between the Gateway/device and the point of entry to the Internet (WAN). This is the case when using 3GPP communications to protect the radio transmission. In this case, the SIM card holding credentials used to secure the access to the wide area network is managed by the mobile network operator to achieve "network access security".

Table 5 lists the major security technologies used in the LAN (device) domains to offer, e.g., authentication, integrity and confidentiality. Table 6 lists the major network technologies, or security technologies used to secure the network access. These technologies provide, e.g., interoperability for secure communication in WANs but in some cases also secure and interoperable communication between devices in WANs and devices in LANs. Table 7 lists security technologies that have been or can be used at application layer, e.g., in M2M services, and also in M2M architectures.

Table 5. Review of security technologies used in the LAN device domains to offer, e.g., authentication, integrity and confidentiality.

Technology, What to Solve?	Forum(s), References	Main Contribution
- Security technologies of Bluetooth - To enable authentication and pairing of Bluetooth devices, to encrypt the transmitted data, and to check integrity of transmitted data packets in Bluetooth network.	Bluetooth Special Interest Group (SIG)	- Frequency-hopping spread spectrum - Master clock is shared to slaves - Binding of devices is done through selected pairing mechanisms - Different security modes - Link keys to establish authenticated and/or encrypted ACL links - SAFER+ block cipher (not for encryption!) - E0 stream cipher - Secure Simple Pairing (SSP) with Elliptic Curve Diffie-Hellman (ECDH).
- Security technologies of Bluetooth low energy - Similar problems to solve as in BT.	Bluetooth Special Interest Group (SIG)	- AES-CCM block cipher - Data can be signed with Connection Signature Resolving Key (CSRK), MAC, counter - Privacy feature to change private address on a frequent basis

Table 5. *Cont.*

- Security technologies of ZigBee - Authentication, encryption, integrity checking, countermeasures against replay-attacks.	ZigBee Alliance	- 16 channel hopping - Key establishment, key transport, frame protection, ad device management implemented mostly at the network (NWK) and application support sub-layer (APS). - AES-CCM to offer confidentiality and authentication, AES-CBC-MAC to offer authentication or AES-CTR to offer confidentiality. 32-, 64-, or 128-bit MAC and 128-bit key. - Key sequence counter
Security technologies of ZigBee RF4CE (Even lower power and simplified version of ZigBee)	ZigBee Alliance	- 3 channel hopping - Simpler pairing mechanism than in ZigBee - AES-CCM with 128-bit key.
- Security technologies of Wi-Fi - Authentication, encryption, integrity, replay protection		- WPA2 that uses Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) based on 128-bit AES-CCM block cipher - Cryptographic hash function - Key management - Replay protection - WPA2 can be used in two modes: - Shared key: relying upon the use of a key shared by all Wi-Fi clients and access point. - Enterprise, relying upon the use of an external AAA server
- Default security technologies of EPCglobal UHF Class 1 Generation 2(v1.2.0) RFID tags - To prevent unauthorized writing and disabling and counterfeiting of RFID tags.	EPCglobal	- 32-bit KILL and ACCESS passwords - (perma)locking of memory - Unique unprogrammable TID numbers

Table 6. Network related security technologies used to enable secure network access.

Technology, What to Solve?	Forum(s), References	Main Contribution
IPsec	IETF's IP Security Protocol (IPsec) concluded working group	IPsec WG developed a security protocol in the network layer to provide cryptographic security services to support combinations of authentication, integrity, access control, and confidentiality. IPsec has been implemented, e.g., over 6LoWPAN [128].
- EAP based authentication - EAP/TLS - EAP/TTLS - EAP/SIM - EAP/AKA - EAP/PEAP	IETF network working group	EAP is an authentication protocol commonly used to secure access to wireless networks and point to point connections.

Table 6. *Cont.*

- TLS/DTLS - To choose cipher suites for providing (mutual) authentication of end-points, encryption and integrity of transmitted data.	IETF's Transport Layer Security (TLS) working group	Specifying new TLS and DTLS protocols and extensions to them. They run on top of transport layer protocols and provide, e.g., confidentiality and data integrity between two communicating applications. Both has been implemented, e.g., over 6LoWPAN.
- WTLS - To offer data integrity, confidentiality and authentication of end-points.	Open Mobile Alliance (OMA)	Similar to TLS but originally meant to be used devices without TCP/IP. Works over UDP or WAP Datagram Protocol (WDP)
3GPP	3rd Generation Partnership Project's (3GPP) Service and System Aspects (SA) working group 3	- IMS security - Security of multimedia broadcast and multicast service (MBMS) - Generic Bootstrapping Architecture (GBA) - Key establishment mechanisms - Ongoing: security for system improvement for machine-type communications - Lawful interception

The security of group communications is an important problem that has been addressed for more than a decade. Most of the research efforts focused on conventional Internet (*i.e.*, Internet Multicasting) both in the wired and wireless environments (e.g., [129–131]). Protocol scalability and the security-performance tradeoff are the major concerns in the group communications security topic. Furthermore, a number of standards have emerged in the IETF [132], and some of them have even been adopted by a number of technology standardization bodies (e.g., 3GPP's MBMS [133] and ETSI's SES [134]).

Group communication security is particularly important in the WSN context when implementing end to end security given the critical resources of WSN networks (CPU, memory, battery, and bandwidth) (e.g., [135–138]). If the data has to be secured from the source, then it is quite important to send a single protected data stream from the device. Unfortunately, asymmetric cryptographic techniques usually involve ciphering the data stream with the public key of the receiver, creating the need for redundant data transmission, which in the case of WSN creates an unbearable load upon the device. Group communication security techniques must therefore be used, involving the distribution of a group key to all parties involved in the communication scheme.

The other aspect that needs to be considered in M2M is the interoperability problem due to network heterogeneity in terms of capacity (e.g., WAN *vs.* LAN) and supported protocols. This interoperability problem requires defining dynamic configuration mechanisms and adaptive protocols to ensure end-to-end security for M2M group communications.

Finally, most of the M2M devices are involved in applications producing or consuming data which need to be secured from the source to the destination, either using hop by hop or end-to-end security. This is addressing "application domain security", and the application providers or the M2M service providers are typically responsible to manage the associated credentials.

Table 7. Security technologies applicable for applications layer.

Technology, What to Solve?	Forum(s), References	Main Contribution
- JSON security - To provide authentication, integrity and confidentiality, access control and resource control	- IETF's JavaScript Object Signing and Encryption (JOSE) working group - Used, e.g., in/by SAML 2.0, WS-Federation, OpenID, OAuth 2.0, XMPP, ALTO and to provide integrity of exigent (alarms)	Standardizing integrity protection and encryption security services, in order to increase interoperability of security features between protocols that use JSON.
- XML security - To provide authentication, integrity and confidentiality.	W3C's XML security working group	XML Signature provides integrity, message authentication, and/or signer authentication services for data. XML Encryption specifies a process for encrypting data and representing the result in XML. XKMS specifies protocols for distributing and registering public keys.
HTTP Strict Transport Security (HSTS)	IETF's Web Security (websec) working group	HSTS is designed to allow web sites or http servers to tell users' browsers or http clients that they want to communicate only over an encrypted connection.
- DNSSEC - To provide secure DNS.	IETF's Domain Name System Security (dnssec) working group	Enhancements to the secure DNS protocol.
ETSI M2M security architecture	ETSI	- Pre-provisioned device/gateway credential types, e.g., SIM/AKA or X.509v3 certificates. - Defining M2M bootstrap procedures based on GBA, TLS or EAP/PANA. - Securing M2M service connection by GBA, TLS or EAP/PANA. - Securing mId by TLS or DTLS, XML security or relying access network security.
- SASL - To provide authentication.	IETF's Simple Authentication and Security Layer (SASL) concluded working group	A framework for authentication and data security in Internet protocols. Application protocol that uses SASL can in theory use any authentication mechanisms supported by SASL.
- HTTPS - Authentication of end-points, protection against MitM attacks, encryption of communication	IETF	Usage of TLS to secure HTTP connections
- AAA protocols - To authenticate entity's identification - To authorize the entity - To account aka track a network resource consumption	- IETF's Authentication, Authorization and Accounting (AAA) concluded working group, RADIUS Extensions (radext) working group, Diameter maintenance and Extensions (dime) working group - 3GPP	- Diameter - Radius - 3GPP EPS AAA
- Trusted Key Distribution Centers (KDC) - Authentication and key exchange, sometimes also access control	- MIT (Kerberos) - OpenID Foundation	- Kerberos 5 - OpenID - Single Sign-On (SSO)

6. Discussion

M2M Service networks are inherently multiple stakeholder systems, which usually consist of parts evolving in different timescales. For example, the generations of cellular radio systems may evolve 10 years, while novel M2M applications may be born even every month. However, the M2M service system lifecycles are required to be even longer than 20 years. If a part of the system is dependent on a single provider, then it is a strong risk for system being operational for such a long lifecycle. Therefore, the system should be based on open standards, and horizontal layering shall be kept clear. If autonomic M2M solutions are developed for the systems where horizontal layers are mixed, the challenge with such solutions is that their application is likely to be limited to the special case only.

In this survey we categorize the available technologies into M2M Information and services, M2M communication and M2M security. According to the review, this categorization could act as a starting point for horizontal architectures of the M2M service networks. However, it is seen that principles and guidelines for the architectures are needed to be defined in order to establish a solid basis for multiple stakeholder system. In addition, it is seen that both end-to-end Internet approach, and M2M gateway based approach are needed to enable horizontally capable M2M service networks. These are needed for solving the heterogeneity of technologies to enable communication between objects and applications, which are not initially been designed to communicate together. It is also seen that the horizontal architecture of the system should be defined first, and then proceeding into the solving of complexity would be feasible. Otherwise, the solutions for solving the complexity may prevent the horizontal approach and thus the basis for a multiple stakeholder system.

6.1. M2M Information and Services

When looking at the reviewed M2M information and service technologies, several subcategories can be found. ETSI M2M defines horizontal M2M service capabilities, which are agnostic for the information content. There are several technologies defining data models (e.g., oBix), ontologies (e.g., OWL) and device management (e.g., OMA DM, UPnP). Open service frameworks (e.g., OSGi) and specification of overlay architectures for integrating sensor networks and applications on the Web (e.g., OGC SWE). The review shows that this area contains a huge amount of technologies which are not necessarily interoperable, and mix at least in the subcategory level. In addition, there are multiple projects working in the area and have defined their own proposal for the architecture and technologies. For example, Hydra [139], Runes [140], IoT-A [141], iCore [142], Sofia [143] and Fi-Ware [144]. It is seen here that there are some essential starting points, such as e.g., having horizontal M2M service capabilities which could be applied with multiple domain M2M applications. However, more detailed analysis and synthesis are needed to establish solid basis for subcategories to enable multiple stakeholder system within M2M information and service level. It is clear that technologies in this area are not ready for this, because for example, ETSI M2M assumes that M2M applications know all details of the device installation and data interpretation, which is challenging for application developers. It is also important to enable means for information level interaction with multiple stakeholders, and enabling autonomic decision making with applications which have not initially been designed for such a purpose. This kind of system is expected to be able to monitor the system in

information level, analyze the situation, plan the required actions, and execute the control events towards the system automatically or at least semi-automatically. In addition, interacting systems need to understand the meaning of the received events, and be able to create actions in such a way that the other player understands it accordingly. This area is still mostly open for future research, and solutions are needed to make *autonomic* M2M service networks a reality. There is a need to develop self-active decision engines enabling autonomic monitoring, analyzing, planning and executing responses to M2M systems. This includes need for smart and adaptive technologies for reasoning based on the complex events and state changes happening in the system. In addition, the autonomic decision engines need to be connected with generic service capabilities in a horizontal way to enable basis for interoperation in a services level.

6.2. M2M Communication

When looking at technologies for M2M communications, subcategories related to M2M communication overlays (e.g., MQTT, HTTP/Coap, XMPP), Internet technologies (e.g., IPv6, 6LowPan), and radio access technologies for WAN (e.g., cellular 3G), LAN (e.g., WiFi) and sensor networks (e.g., Bluetooth Smart) can be found. M2M applications are usually based on messaging with M2M devices. Traditionally, such messaging is done with short message service (SMS) or Email systems. It is seen that real-time messaging, capabilities to handle not always on mobile devices and capabilities for more dynamic topologies are needed. Therefore, technologies such as, e.g., XMPP to enable real-time M2M messaging, presence management and dynamic topologies are potential for M2M systems. In addition, there is need for virtualization of communications, and having a kind of M2M communication overlay hiding the heterogeneity of underlying networks and ensuring security. This is because of, e.g., the need to connect M2M asset devices with limited power and computing capabilities into the system. In addition, the referred M2M asset devices ownership set limits for the communication. When connecting such resources into system, also lower levels of communication need optimization. Examples of potential areas of research may be related to novel Bluetooth profiles, IPv6 over Bluetooth low energy, energy efficient IPv6 multicast, Coap, 6LowPAN and RPL including optimized M2M gateway and M2M messaging solutions.

6.3. M2M Security

There are huge amount of technologies, which could be applied for enabling M2M security as shown by the review. There is very strong need for defining guidelines for the use of M2M security technologies also because M2M applications are usually very sensitive on misuse there are high requirements for reliability. When both ends of the communication are mobile, there is need e.g., for advanced credential management, creation of trust, bridging mobile asset network and WAN security, adaptive M2M secure elements and horizontal end to end security solutions. For example, creation of trust is traditionally established in hop by hop manner between M2M asset devices and M2M applications. This kind of model can be challenging in M2M systems, because of M2M information content may be business critical and it may contain high privacy requirements. Therefore, these areas are also open for future research and novel solutions.

6.4. Interoperability

According to the review, it can be seen that there are technologies available for M2M service communications between remote physical objects over the network infrastructures with virtual world services. However, there are still challenges related to adaptability, reliability, performance, security, smartness, interoperability, and cost of development, manual operation and maintenance during the life-cycle of M2M products. Interoperability with existing systems and resource constrained embedded devices are usually approached via gateways of some sort. For example, the functional architecture designed by ETSI M2M [145], Cisco [146], AnyBridge [147], Systech [148], Alcatel Lucent [149] and IOT-A [141] relies on the use of a kind of M2M gateway mainly because of challenges related to communication with constrained devices. Such a M2M gateway can handle, e.g., the issues related to communicating with a system based on an incompatible communication protocol, low-power devices which are unable to communicate with the rest of the system directly due to limited resources or capabilities, or communication with a domain in which the access is otherwise restricted by some service provider. Such a gateway can take care of mapping of protocols to be more applicable for embedded capillary networks and devices, and enable interoperability between various proprietary networks. For example, the M2M gateway (which may also be called a border router) can translate HTTP to CoAP, IPv6 to 6LowPan, XMPP to Bluetooth Smart and 6LowPan to Bluetooth Smart messages. In addition, the gateway can act as a translating and security element, which can interconnect two systems having different protocols and data formats and perhaps belonging to different security domains. Such gateway component may not be optimal from communication point of view, but it is required in some cases because of interoperability and security.

A gateway may also prevent message flooding from devices to the backbone network, enable management of M2M asset devices in groups, make maintenance and configuration smooth, enable usage of unlicensed frequency bands and/or optimized radio technologies for specific M2M asset devices. Typically, a gateway is then connected to a back-end server which is taking care of data storages, management, centralized control and enforcement of security policies. The use of back-end servers have a crucial role in combatting the various scalability and reliability issues found in pure peer-to-peer and ad-hoc -type systems [150–152].

There seems to be multiple communication options for realization of M2M gateways: M2M gateway as an Internet protocol (IP) router, M2M gateway as a service gateway and direct connection to M2M devices without any M2M gateway. When M2M gateway is acting as an IP router, it makes possible to establish end-to-end IP connectivity if M2M asset devices are supporting IP. In that case, the local radio access technology needs to have mapping to IP communication. If M2M asset devices are not supporting IP, then there is a need to have an M2M service gateway, which is able to act as a bridge/protocol translator between M2M asset network and M2M infrastructure. There are also multiple options for making protocol translation from WAN connectivity to LAN connectivity in M2M gateways. For example, the service capable M2M gateway is able to make protocol adaptation between proprietary protocol stack, and ETSI M2M SCL. Communication with constrained M2M devices can apply several options, however, there are several practical challenges which require optimization within the local M2M asset network such as, e.g., application of web services within constrained local M2M asset network, sizes of IP packets and headers, power consumption of radio access protocols,

heterogeneity and mobility of M2M devices and local M2M asset networks, and coding and integration of M2M application content. Some solutions for these challenges (e.g., Coap, 6LowPAN, Bluetooth LE) are already available; however, there are still several open issues for future research in these areas.

6.5. Horizontal and Autonomic M2M

Based on the review, it is seen that principles for the architecture need to be defined in order to establish a solid basis for multiple stakeholder system for M2M service networks. In addition, architecture principles for solving the heterogeneity of technologies are needed to enable communication between objects and applications, which are not initially been designed to communicate together. An approach may be horizontal approach for the system architecture and application of autonomic computing principles [153]. Such horizontal architecture is based on open standards to enable connectivity and interoperation with multiple M2M domains and heterogeneous devices and to enable reaching wide acceptance as a basis of multiple stakeholder M2M systems. In such a way, several different M2M domains could become capable of deploying the architecture, enabling interoperability, lowering the development cost and boosting the arising M2M markets by contributing towards transferring from vertical towards more horizontal M2M markets.

7. Concluding Remarks

This survey provides an overview of the potential technologies enabling future autonomic M2M service networks. However, a clear definition of the architectural principles for horizontal M2M system is needed first, which must then proceed towards enabling autonomic capabilities in a feasible way. Otherwise, the solutions for solving the complexity may prevent the horizontal approach and thus the basis for multiple stakeholder system and enablers for autonomic M2M. In addition, it is important to enable interoperation with multiple application domains.

Solving the complexity in future M2M systems opens several areas for future research, such as, e.g., autonomic decision engines, reasoning, generic service capabilities for autonomic computing, virtualization of M2M communications, optimization of communications with limited capability devices, M2M messaging, and last but not the least M2M security for advanced credential management, creation of trust, bridging mobile asset network and WAN security, adaptive M2M secure elements and horizontal end to end solutions.

According to the review, there are technologies available for M2M service communications between remote physical objects over the network infrastructures and virtual world services. However, there are still challenges related to adaptability, reliability, performance, security, smartness, interoperability, and cost of development, manual operation and maintenance during the life-cycle of M2M products.

Acknowledgments

This review is related to deliverables of ITEA2 A2Nets project dealing with autonomic M2M service networks. The authors wish to thank all contributors of A2Nets project, and especially the

A2Nets ITEA2 reviewers and public research funding organizations in Spain, France, Turkey and Finland for making this work possible.

Authors Contributions

Paul Vitic has contributed into Section 3.2; Bashar Jubeh into Section 3.3; Mahdi Ben Alaya and Thierry Monteil into Section 3.4; Yoann Lopez into Sections 4.2 and 4.3; Guillermo Talavera and Javier Gonzalez into Sections 4.1 and 4.6; Niclas Granqvist into Section 4.5; Monir Kellil into group communication security issues in Chapter 5; Herve Ganem and Teemu Väisänen into Chapter 5; Antti Iivari into interoperability issues in Chapter 6. Juhani Latvakoski has contributed into all sections.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Miorandi, D.; Sicari, S.; de Pellegrini, F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw*. **2012**, *10*, 1497–1516.
2. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw*. **2002**, *38*, 393–422.
3. Wu, G.; Talwar, S.; Johnsson, K.; Himayat, N.; Johnson, K.D. M2M: From mobile to embedded internet. *IEEE Commun. Mag.* **2011**, *49*, 36–43.
4. Kephart, J.O.; Chess, D.M. The vision of autonomic computing. *Computer* **2003**, *36*, 41–50.
5. Horn, P. Autonomic Computing: IBM's Perspective on the State of Information Technology. Available online: http://people.scs.carleton.ca/~soma/biosec/readings/autonomic_computing.pdf (accessed on 11 November 2014).
6. Salehie, M.; Tahvildari, L. Autonomic computing: Emerging trends and open problems. *ACM SIGSOFT Softw. Eng. Notes* **2005**, *30*, 1–7.
7. Parashar, M.; Hariri, S. Autonomic computing: An overview. In *Unconventional Programming Paradigms*, Springer: Berlin/Heidelberg, Germany; 2005; pp. 257–269.
8. Garlan, D.; Cheng, S.-W.; Huang, A.-C.; Schmerl, B.; Steenkiste, P. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer* **2004**, *37*, 46–54.
9. Nami, M.R.; Bertels, K. A survey of autonomic computing systems. In Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems (ICAS'07), Athens, Greece, 11–13 February 2007, pp. 26–30.
10. Karmouch, A. Mobile software agents for telecommunications. *IEEE Commun. Mag.* **1998**, *36*, 24–25.
11. Boutaba, R.; Polyrakis, A. Projecting advanced enterprise network and service management to active networks. *IEEE Netw*. **2002**, *16*, 28–33.
12. Calo, S.; Sloman, M. Guest Editorial: Policy-Based Management of Networks and Services. *J. Netw. Syst. Manag.* **2003**, *11*, 249–252.
13. Murch, R. *Autonomic Computing*; Prentice Hall: Upper Saddle River, NJ, USA, 2004.

14. IBM Corporation. An Architectural Blueprint for Autonomic Computing. White Paper, 4th ed. October 2004. Available online: http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf (accessed on 30 October 2014)
15. Want, R.; Pering, T.; Tennenhouse, D. Comparing autonomic and proactive computing. *IBM Syst. J.* **2003**, *42*, 129–135.
16. IEC 61970. Common Information Model (CIM)/Energy Management. Available online: http://www.iec.ch/smartgrid/standards/ (accessed on 30 October 2014).
17. Pakkala, D.; Latvakoski, J. Distributed service platform for adaptive mobile service. *Int. J. Pervasive Comput. Commun.* **2006**, *2*, 135–147.
18. ETSI Technical Specification 102 690 Machine to Machine communications (M2M) Functional Architecture. V2.1.1. Available online: http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v020101p.pdf (accessed on 30 October 2014).
19. Chang, K.; Soong, A.; Tseng, M.; Xiang, Z. Global Wireless Machine-to-Machine Standardization. *IEEE Internet Comput.* **2011**, *15*, 64–69.
20. Latvakoski, J.; Pakkala, D.; Paakkonen, P. A communication architecture for spontaneous systems. *IEEE Wirel. Commun.* **2004**, *11*, 36–42.
21. IPSO Alliance. Available online: http://www.ipso-alliance.org/ (accessed on 29 October 2014).
22. The Internet Engineering Task Force (IETF). Available online: http://www.ietf.org/ (accessed on 29 October 2014).
23. ETSI M2M/Smart M2M. Available online: http://www.etsi.org/ (accessed on 29 October 2014).
24. One M2M forum. Available online: http://www.onem2m.org/ (accessed on 29 October 2014).
25. Electronic Product Codes (EPCglobal). Available online: http://www.gs1.org/epcglobal/ (accessed on 29 October 2014).
26. uID center. Available online: http://www.uidcenter.org/ (accessed on 29 October 2014).
27. ONVIF. Available online: http://www.onvif.org/ (accessed on 29 October 2014).
28. Openmeter. Available online: http://www.openmeter.com/ (accessed on 29 October 2014).
29. OGC Sensor Web Enablement (SWE). Available online: http://www.opengeospatial.org/ogc/markets-technologies/swe (accessed on 29 October 2014).
30. OASIS User Interface Markup Language (UIML). Available online: http://www.uiml.org (accessed on 29 October 2014)
31. Universal Plung and Play (UPnP). Available online: http://www.upnp.org/ (accessed on 29 October 2014).
32. Li, J. On peer-to-peer (P2P) content delivery. *Peer-to-Peer Netw. Appl.* **2008**, *1*, 363–381.
33. Meshkova, E.; Riihijärvi, J.; Petrova, M.; Mähönen, P. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Comput. Netw.* **2008**, *52*, 2097–2128.
34. Tarkoma, S. *Overlay Networks—Towards Information Networking*; Auerbach Publications, Taylor & Francis Group: Boca Raton, FL, USA, 2010; p. 245.
35. Hoebeke, J.; Moerman, I.; Dhoedt, B.; Demeester, P. An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *J. Commun. Netw.* **2004**, *3*, 60–66.
36. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330.

37. Bluetooth 4.0. Available online: https://www.bluetooth.org/apps/content/ (accessed on 29 October 2014).
38. ETSI. ETSI Technical Specification 102 921: "Machine to Machine Communications (M2M); mIa, dIa and mId interfaces". Available online: http://www.etsi.org/deliver/etsi_ts/102900_102999/102921/02.01.01_60/ts_102921v020101p (accessed on 30 October 2014).
39. Open Mobile Alliance (OMA). OMA Device Management Tree and Description Serialization Specification, Version 1.2.1 Open Mobile Alliance. Available online: http://technical.openmobilealliance.org/Technical/release_program/docs/DM/V1_2_1-20080617-A/OMA-TS-DM_TND-V1_2_1-20080617-A.pdf (accessed on 30 October 2014).
40. Open Mobile Alliance. Available online: http://openmobilealliance.org/ (accessed on 29 October 2014).
41. Broadband Forum. TR-181 Device Data Model for TR-069, Issue 2. Available online: http://www.broadband-forum.org/technical/download/TR-181_Issue-2.pdf (accessed on 30 October 2014).
42. Broadband Forum. Available online: http://www.broadband-forum.org/ (accessed on 29 October 2014).
43. Jeronimo, M.; Weast, J. *UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play*; Intel Press: Santa Clara, CA, USA, 2003.
44. OASIS Devices Profile for Web Services (DPWS) Available online: http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01 (accessed on 29 October 2014).
45. Open building information Xchange (oBIX). Available online: http://www.obix.org/ (accessed on 29 October 2014).
46. Hannelius, T.; Salmenpera, M.; Kuikka, S. Roadmap to adopting OPC UA. In Proceedings of the 6th IEEE International Conference on Industrial Informatics, Daejeon, Korea, 13–16 July 2008; pp.756–761.
47. OPC Foundation. Available online: http://www.opcfoundation.org/ (accessed on 29 October 2014).
48. Martin, D.; Burstein, M.; Mcdermott, D.; Mcilraith, S.; Paolucci, M.; Sycara, K.; Mcguinness, D.L.; Sirin, E.; Srinivasan, N. Bringing Semantics to Web Services with OWL-S. 2007. Available online: http://link.springer.com/article/10.1007%2Fs11280-007-0033-x (accessed on 30 October 2014).
49. Semantic Markup for Web Services. Available online: http://www.w3.org/Submission/OWL-S/ (accessed on 29 October 2014).
50. Compton, M.; Barnaghi, P.; Bermudez, L.; Garc á-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; *et al*. The SSN ontology of the W3C semantic sensor network incubator group, *Web Semant. Sci. Serv. Agents World Wide Web* **2012**, *17*, 25–32.
51. W3C Semantic Sensor Networks. Available online: http://www.w3.org/2005/Incubator/ssn/ (accessed on 29 October 2014).
52. Bandara, A.; Payne, T.; de Roure, D.; Clemo, G. An Ontological Framework for Semantic Description of Devices. In Proceedings of the International Semantic Web Conference (ISWC), Hiroshima, Japan, 7–11 November 200.

53. FIPA Device Ontology Specification. Available online: http://www.fipa.org/specs/fipa00091/PC00091A.html (accessed on 29 October 2014).
54. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New Generation Sensor Web Enablement. Open Access-Sensors ISSN 1424–8220. Available online: http://www.mdpi.com/1424-8220/11/3/2652 (accessed on 30 October 2014).
55. Botts, M. *OpenGeo Sensor Web Enablement (SWE) Suite*; Botts Innovative Research Inc.: June 2011. Available online: http://boundlessgeo.com/whitepaper/opengeo-sensor-web-enablement-swe-suite/ (accessed on 30 October 2014).
56. Sensor Web Enablement (SWE). Available online: http://www.ogcnetwork.net/swe (accessed on 30 October 2014).
57. OSGi Alliance. Available online: http://www.osgi.org/ (accessed on 29 October 2014).
58. TMForum. Available online: http://www.tmforum.org/ (accessed on 29 October 2014).
59. Jammes, F.; Mensch, A.; Smit, H. Service-oriented device communications using the devices profile for web services. In Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, Grenoble, France, 28 November–2 December 2005.
60. Delin, K.; Jackson, S.; Some, R. Sensor Webs. *NASA Tech. Briefs* **1999**, *23*, 90.
61. Gibbons, P.; Karp, B.; Ke, Y.; Nath, S.; Seshan, S. Irisnet: An Architecture for a Worldwide Sensor Web. *IEEE Pervasive Comput.* **2003**, *2*, 22–33.
62. Shneidman, J.; Pietzuch, P.; Ledlie, J.; Roussopoulos, M.; Seltzer, M.; Welsh, M. *Hourglass: An Infrastructure for Connecting Sensor Networks and Applications*; Technical Report; Harvard University: Columbia, MA, USA, 2004.
63. Moodley, D.; Simonis, I. A New Architecture for the Sensor Web: The SWAP Framework. In Proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, GA, USA, 5–9 November 2006; Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., Eds.; Lecture Notes in Computer Science; Volume 4273.
64. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. OGC Sensor Web Enablement: Overview and High Level Architecture. In Proceedings of the 2nd International Conference on GeoSensor Networks, GSN 2006, Boston, MA, USA, 1–3 October 2006; Nittel, S., Labrinidis, A., Stefanidis, A., Eds.; Lecture Notes In Computer Science; Springer: Berlin, Germany, 2008; Volume 4540, pp. 175–190.
65. Open Geospatial Consortium Inc. OGC SWE Common Data Model Encoding Standard (OGC SWE Common). 2011. Available online: http://www.opengis.net/doc/IS/SWE/2.0 (accessed on 30 October 2014).
66. Open Geospatial Consortium. Observations and Measurements—XML Implementation (OGC O&M). 2011. Available online: http://www.opengis.net/doc/IS/OMXML/2.0 (accessed on 30 October 2014).
67. Open Geospatial Consortium. (*Sensor Model Language (SensorML). OpenGIS Implementation Specification* (OGC SensorML); Version 1.0.0; 2007. Available online: http://portal.opengeospatial.org/files/?artifact_id=21273 (accessed 30 October 2014)
68. Open Geospatial Consortium. OpenGIS SWE Service Model Implementation Standard. 2011. Available online: http://www.opengis.net/doc/IS/SWES/2.0 (accessed on 30 October 2014).

69. Open Geospatial Consortium. Sensor Observation Service Implementation Standard, SOS, Version 1.0, 2007. Available online: http://www.opengeospatial.org/ (accessed on 30 October 2014).
70. Open Geospatial Consortium. OGC Sensor Planning Service Implementation Standard SPS, 2011. Available online: http://www.opengis.net/doc/IS/SPS/2.0 (accessed on 30 October 2014).
71. Fowler M. *Analysis Patterns: Reusable Object Models*; Addison-Wesley: Indianapolis, IN, USA, 1997.
72. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. Thesis, University of California, Irvine, CA, USA, 2000.
73. Hariri, S.; Khargharia, B.; Chen, H.; Yang, J.; Zhang, Y.; Parashar, M.; Liu, H. The autonomic computing paradigm. *Clust. Comput.* **2006**, *9*, 5–17.
74. Chakravarti, A.J.; Baumgartner, G. The organic grid: Self-organizing computation on a peer-to-peer network. In Proceedings of the 1st International Conference on Autonomic Computing (ICAC), New York, NY, USA. 17–18 May 2004.
75. Appleby, K.; Fakhouri, S.; Fong, L.; Goldszmidt, G.; Kalantar, M.; Pazel, D.; Pershing, J.; Rochwerger, B. Oceano—SLA based management of a computing utility. In Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management, Seattle, WA, USA, 14–18 May 2001; pp. 855–868.
76. Rhea, S.; Wells, C.; Eaton, P.; Geels, D.; Zhao, B.; Weatherspoon, H.; Kubiatowicz, J. Maintenance-free global data storage. *IEEE Internet Comput.* **2001**, *5*, 40–49.
77. Lohman, G.M.; Lightstone, S.S. Smart: Making DB2 (more) autonomic. In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, 20–23 August 2002; pp. 877–879.
78. Chaudhuri, S.; Narasayya, V. Self-tuning database systems: A decade of progress. In Proceedings of the 33rd International Conference on Very large data bases, Vienna, Austria. 23–27 September 2007; pp. 3–14.
79. Chess, D.; Segal, A.; Whalley, I.; White, S. Unity: Experiences with a prototype autonomic computing system. In Proceedings of the International Conference on Autonomic Computing, TeX90 CD ’04, ICAC, New York, NY, USA, 17–18 May 2004; pp. 140–147.
80. Dong, X.; Hariri, S.; Xue, L.; Chen, H.; Zhang, M.; Pavuluri, S.; Rao, S. Autonomia: An autonomic computing environment. In Proceedings of the Performance, Computing, and Communications Conference, Phoenix, AZ, USA, 9–11 April 2003; pp. 61–68.
81. Liu, H.; Parashar, M. A component based programming framework for autonomic applications. In Proceedings of the International Conference on Automatic Computing, New York, NY, USA. 17–19 May 2004; pp. 10–17.
82. Sicard, S.; Boyer, F.; de Palma, N. Using components for architecture-based management: The self-repair case. In Proceedings of the 30th International Conference on Software Engineering, Leipzig, Germany,10–18 May 2008.
83. Ranganathan, A.; Campbell, R.H. Autonomic pervasive computing based on planning. In Proceedings of the International Conference on Autonomic Computing, New York, NY, USA. 17–19 May 2004; pp. 80–87.

84. Abdelwahed, S.; Kandasamy, N.; Neema, S. Online control for self-management in computing systems. In Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004), Toronto, ON, Canada, 25–28 May 2004; pp. 368–375.
85. Gurguis, S.A.; Zeid, A. Towards autonomic web services: Achieving self-healing using web services. *SIGSOFT Softw. Eng. Notes* **2005**, *30*, 1–5.
86. Kaiser, G.; Gross, P.; Kc, G.; Parekh, J.; Valetto, G. An approach to autonomizing legacy systems. In Proceedings of the Workshop on Self-Healing, Adaptive and Self-Managed Systems, New York, NY, USA, 23 June 2002.
87. Kaiser, G.; Parekh, J.; Gross, P.; Valetto, G. Kinesthetics extreme: An external infrastructure for monitoring distributed legacy systems. In Proceedings of the Autonomic Computing Workshop Fifth Annual International Workshop on Active Middleware Services (AMS'03), Seattle, WA, USA, 25 June 2003; pp. 22–30.
88. Anglano, C.; Montani, S. Achieving self-healing in autonomic software systems: A case-based reasoning approach. In Proceedings of the International Conference on Intelligent Computing, Qingdao, China, 21–24 August 2007; pp. 1–19.
89. Stanfel, Z.; Hocenski, Z.; Martinovic, G. A self-manageable rule driven enterprise application. In Proceedings of the 29th International Conference on Information Technology Interfaces (ITI 2007), Cavtat/Dubrovnik, Croatia, 25–28 June 2007; pp. 717–722.
90. Appavoo, J.; Hui, K.; Soules, C.; Wisniewski, R.; Silva, D.; Krieger, O.; Marc, D.; Auslander, A.; Gamsa, B.; Ganger, G.; *et al*. Enabling autonomic behavior in systems software with hot-swapping. *IBM Syst. J.* **2003**, *42*, 60–76.
91. Kcman, E.; Wang, Y.-M. Discovering correctness constraints for self-management of system configuration. In Proceedings of the International Conference on Autonomic Computing, New York, NY, USA, 17–19 May 2004; pp. 28–35.
92. Mesnier, M.; Thereska, E.; Ganger, G.R.; Ellarda, D.; Seltzer, M. File classification in self-* storage systems. In Proceedings of the 1st International Conference on Autonomic Computing (ICAC-04), New York, NY, USA, 17–19 May 2004; pp. 44–51.
93. Khan, M.J.; Awais, M.M.; Shamail, S. Achieving self-configuration capability in autonomic systems using case-based reasoning with a new similarity measure. In Proceedings of the 3rd International Conference on Intelligent Computing, Qingdao, China, 21–24 August 2007; pp. 97–106.
94. Hellerstein, J.L.; Gandhi, N.; Parekh, S.S. Managing the Performance of Lotus Notes: A Control Theoretic Approach. In Proceedings of the 27th International Computer Measurement Group Conference, Anaheim, CA, USA, 2–7 December 2001; pp. 397–408.
95. Markl, V.; Lohman, G.M.; Raman, V. LEO: An autonomic query optimizer for DB2. *IBM Syst. J.* **2003**, *42*, 98–106.
96. Shivam, P.; Babu, S.; Chase, J.S. Learning application models for utility resource planning. In Proceedings of the International Conference on Automatic Computing, Dublin, Ireland, 13–16 Jun 2006.
97. Park, J.; Chandramohan, P. Static *vs.* Dynamic recovery models for survivable distributed systems. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences; Big Island, HI, USA, 5–8 January 2004; pp. 1–9.

98. Rish, I.; Brodie, M.; Odintsova, N.; Ma, S.; Grabarnik, G. Realtime problem determination in distributed systems using active probing. In Proceedings of the Network Operations Management Symposium (NOMS 2004), Seoul, Korea, 23 April 2004; pp.133–146.
99. Mills, K.; Rose, S.; Quirolgico, S.; Britton, M.; Tan, C. An autonomic failure-detection algorithm. *ACM SIGSOFT Softw. Eng. Notes* **2004**, *29*, 79–83.
100. Reynolds, P.; Killian, C.; Wiener, J.L.; Mogul, J.C.; Shah, M.A.; Vahdat, A. Pip: Detecting the unexpected in distributed systems. In Proceedings of the conference on Symposium on Networked Systems Design and Implementation, San Jose, CA, USA, May 2006; pp. 1–14.
101. Roblee, C.; Berk, V.; Cybenko, G. Implementing large-scale autonomic server monitoring using process query systems. In Proceedings of the Second International Conference on Automatic Computing, Seattle, WA, USA, 13–16 June 2005; pp. 123–133.
102. Candea, G.; Kawamoto, S.; Fujiki, Y.; Friedman, G.; Fox, A. Microreboot—A technique for cheap recovery. In Proceedings of the 6th Symposium on Opearting Systems Design and Implementation, San Francisco, CA, USA, 5 December 2004; pp. 31–44.
103. Qin, F.; Tucek, J.; Sundaresan, J.; Zhou, Y. Rx: Treating bugs as allergies- A safe method to survive software failures. *ACM Trans. Comput. Syst.* **2007**, *25*, doi:10.1145/1275517.1275519.
104. Claudel, B.; Palma1, N.D.; Lachaize, R.; Hagimont, D. Self protection for distributed component-based applications. In Proceedings of the 8th Symposium on Stabilization, Safety, and Security of Distributed Systems, Dallas, TX, USA, 17–19 November 2006; pp. 184–198.
105. Costa, M.; Crowcroft, J.; Castro, M.; Rowstron, A.; Zhou, L.; Zhang, L.; Barham, P. Vigilante: End-to-end containment of internet worms. In Proceedings of the ACM symposium on Operating systems principles, Brighton, UK, 23–26 October 2005; pp. 133–147.
106. Saint-Andre, P.; Smith, K.; Tronçon, R. *XMPP: The Definitive Guide*; O' Reilly Media Inc.: Sebastopol, CA, USA, 2009; p. 306.
107. Sensor over XMPP. Available online: http://xmpp.org/extensions/inbox/sensors.html (accessed on 30 October 2014).
108. MQTT protocol. Available online: http://mqtt.org (accessed on 30 October 2014).
109. IETF IPv6 over Low Power WPAN (6LowPAN) WG. Available online: http://tools.ietf.org/wg/6lowpan (accessed on 30 October 2014).
110. Hui, J. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. IETF RFC 6282, September 2011. Available online: http://tools.ietf.org/html/rfc6282 (accessed on 30 October 2014)
111. Constrained RESTful Environments (CoRE) WG. Available online: http://tools.ietf.org/wg/core/ (accessed on 30 October 2014).
112. IETF Routing Over Low power and Lossy networks (ROLL) WG. Available online: http://tools.ietf.org/wg/roll/ (accessed on 30 October 2014).
113. Constrained RESTful Environments (CoRE) Link Format, IETF RFC 6690. Available online: http://tools.ietf.org/html/rfc6690 (accessed on 30 October 2014).
114. Kovatsch, M.; Duquennoy, S.; Dunkels, A. A Low-Power CoAP for Contiki. In Proceedings of the Workshop on Internet of Things Technology and Architectures (IEEE IoTech 2011), Valencia, Spain, 17 October 2011.

115. Rescorla, E.; Modadugu, N. "Datagram Transport Layer Security" IETF RFC 4347, April 2006. Available online: http://tools.ietf.org/html/rfc4347 (accessed on 30 October 2014).
116. Kent, S. "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005. Available online: http://tools.ietf.org/html/rfc4303 (accessed on 30 October 2014).
117. Yegin, A.; Shelby, Z. CoAP Security Options. Internet-Draft, October 14, 2011, Expired April 16, 2012. Available online: http://www.ietf.org/archive/id/draft-yegin-coap-security-options-00.txt (accessed on 30 October 2014).
118. Saint-Andre, P. Extensible Messaging and presence protocol (XMPP): IETF RFC 3921. Available online: https://tools.ietf.org/html/rfc3921 (accessed on 30 October 2014).
119. Instant Messaging/Presence Protocol Requirements. RFC 2779, February 2000. Available online: https://tools.ietf.org/html/rfc2779 (accessed on 30 October 2014).
120. Bluetooth SIG. The Bluetooth Core specification v4.0. 2010. Available online: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737 (accessed on 11 November 2014)
121. Bluetooth SIG. Core specification addendum 3, 2012. Available online: https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=269452 (accessed on 11 November 2014)
122. Parekh, J.; Kaiser, G.; Gross, P.; Valetto, G. Retrofitting Autonomic Capabilities onto Legacy Systems. *J. Cluster Comput.* **2006**, *9*, 141–159.
123. Liu, H.; Parashar, M. Accord: A programming framework for autonomic applications. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2006**, *36*, 341–352.
124. ANA: Autonomic Network Architecture. Available online: http://www.uio.no/studier/emner/matnat/ifi/INF5090/v08/undervisningsmateriale/D.1.4-5-6.Blueprint.pdf (accessed 30 October 2014).
125. Initial IoT Protocol Suite Definition. Available online: http://www.iot-a.eu/public/public-documents/documents-1/1/1/d3.3/at_download/file (accessed on 30 October 2014).
126. The Anthill Project. Available online: http://www.cs.unibo.it/projects/anthill (accessed on 30 October 2014).
127. Inhyok, C.; Shah, Y.; Schmidt, A.U.; Leicher, A.; Meyerstein, M. Security and trust for M2M Communications. *IEEE Veh. Tech. Mag.* **2009**, 69–75.
128. Raza, S.; Chung, T.; Duquennoy, S.; Yazar, D.; Voigt, T.; Roedig, U. Securing Internet of Things with Lightweigth IPsec. Available online: http://soda.swedish-ict.se/4052/ (accessed on 30 October 2014).
129. Judge, P.; Ammar, M. Security issues and solutions in multicast content distribution: A survey. *IEEE Netw*. **2003**, *17*, 30–36.
130. Challal, Y.; Bettahar, H.; Bouabdallah, A. A Taxonomy of Multicast Data Origin Authentication: Issues and Solutions. *IEEE Commun. Surv.* **2004**, *3*, 34–57.
131. Kellil, M.; Romdhani, I.; Lach, H-Y. Multicast Receiver and Sender Access Control and its Applicability to Mobile IP Environments: A Survey. *IEEE Commun. Surv.* **2005**, *7*, 46–70.
132. Hardjono, T.; Weiss, B. The Multicast Group Security Architecture. Available online: https://tools.ietf.org/html/rfc3740 (accessed on 30 October 2014)

133. ETSI. Security of Multimedia Broadcast/Multicast Service (MBMS). Available online: http://www.etsi.org/deliver/etsi_ts/133200_133299/133246/10.01.00_60/ts_133246v100100p.pdf (accessed 30 October 2014)
134. ETSI. Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM). Available online: http://www.etsi.org/deliver/etsi_ts/102400_102499/102466/01.01.01_60/ts_102466v010101p.pdf (accessed on 30 October 2014)
135. Ben Jaballah, W.; Meddeb, A.; Youssef, H. An efficient source authentication scheme in wireless sensor networks. In Proceedings of the Proceedings of IEEE/ACS AICCSA Conference, Hammamet, Tunisia, 16–19 May 2010; pp. 1–7.
136. Lim, S.Y.; Lim, M.H. Energy-Efficient and Scalable Group Key Management for Hierarchical Sensor Network. *J. Ubiquitous Syst. Pervasive Netw.* **2011**, *2*, 39–47.
137. Alexander, R.; Tsao, T. Adapted Multimedia Internet KEYing (AMIKEY): An extension of Multimedia Internet KEYing (MIKEY) Methods for Generic LLN Environments. Available online: http://tools.ietf.org/html/draft-alexander-roll-mikey-lln-key-mgmt-04 (accessed on 30 October 2014)
138. Keoh, S.; Kumar, S.; Garcia-Morchon, O.; Dijk, E.; Rahman, A. DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs). Available online: http://tools.ietf.org/html/draft-keoh-dice-multicast-security-04 (accessed 30 October 2014)
139. Hydra Project. Available online: http://www.hydramiddleware.eu (accessed on 17 April 2014).
140. Runes Project. Available online: http://www.ist-runes.org (accessed on 17 April 2014).
141. IoT-A Project. Available online: http://www.iot-a.eu (accessed on 17 April 2014).
142. ICore Project. Available online: http://www.iot-icore.eu (accessed on 17 April 2014).
143. Sofia Project. Available online: http://www.artemis-ia.eu/project/index/view?project=4 (accessed on 17 April 2014).
144. Fi-Ware Project Architecture. Available online: http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Architecture (accessed on 17 April 2014).
145. ETSI Machine to Machine Communications. Available online: http://www.etsi.org/website/technologies/m2m.aspx (accessed on 17 April 2014).
146. Cisco. Available online: http://www.fiercebroadbandwireless.com/story/cisco-introduces-small-m2m-gateway-businesses/2011-08-25 (accessed on 17 April 2014).
147. AnyBridge. Available online: http://www.anybridge-m2m.nl/home (accessed on 17 April 2014).
148. Systech. Available online: http://www.systech.com/ (accessed on 17 April 2014).
149. Alcatel-Lucent. Available online: http://www2.alcatel-lucent.com/blogs/techzine/2011/getting-ready-for-m2m-traffic-growth/ (accessed on 17 April 2014).
150. Androutsellis-Theotokis, S.; Spinellis, D. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.* **2004**, *36*, 335–371.
151. Zheng, H.; Yan, M. Research and Analysis of the Optimization of the Unstructured P2P Overlay Networks. In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'09), Beijing, China, 24–26 September 2009; IEEE Press: Piscataway, NJ, USA. 2009; pp. 4376–4379.
152. Lua, E.K.; Crowcroft, J.; Pias, M.; Sharma, R.; Lim, S. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Commun. Surv. Tutor.* **2005**, *7*, 72–93.

153. Latvakoski, J.; Alaya, M.B.; Ganem, H.; Jubeh, B.; Iivari, A.; Leguay, J.; Bosch, J.M.; Granqvist, N. Towards Horizontal Architecture for Autonomic M2M Service Networks. *Futur. Internet* **2014**, *6*, 261–301.