A Survey of Technologies for the Internet of Things

Vangelis Gazis*, Manuel Görtz*, Marco Huber*, Alessandro Leonardi*, Kostas Mathioudakis*
Alexander Wiesmaier*, Florian Zeiger*, Emmanouil Vasilomanolakis*†
*AGT Group (R&D) GmbH, Hilpertstrasse 35, 64295 Darmstadt, Germany
{vgazis, mgoertz, mhuber, aleonardi, kmathioudakis, awiesmaier}@agtinternational.com, fzeiger@ieee.org
†TU Darmstadt / CASED, Mornewegstr. 32, 64293 Darmstadt, Germany
manolis@cased.de

Abstract—**The number of smart things is growing exponentially. By 2020, tens of billions of things will be deployed worldwide, collecting a wealth of diverse data. Traditional computing models collect in-field data and then transmit it to a central data center where analytics are applied to it, but this is no longer a sustainable model. New approaches and new technologies are required to transform enormous amounts of collected data into meaningful information. Technology also will enable the interconnection around things in the IoT ecosystem but further research is required in the development, convergence and interoperability of the different IoT elements. In this paper, we provide a picture of the main technological components needed to enable the interconnection among things in order to realize IoT concepts and applications.**

I. Introduction

Just as the majority of technological innovations created by the human being through the ages, also the emerging IoT technology is driven by applications that are mainly aimed at improving people's quality of life while saving operational costs for companies or public authorities. Applications related to IoT can be found in several and different domains: energy, health, transportation, environment, etc. Thousands of applications can be identified in each domain and new ones appear everyday, requiring a strong interconnection among things [1].

Interconnection of things is not only a mere technological issue but it concerns also other aspects such as privacy, standardization, legal issues, etc. This inevitably brings new challenges inside the IoT ecosystem that have been keeping industrial and academics researchers busy over the last decade [2]. Surely, the technological interconnection among the devices (i.e., things) is fundamental to enable the IoT ecosystem. For this reason, an important role will be played by some technological components and techniques that have to evolve and adapt in order to guarantee interconnection among heterogenous devices having in most of the cases very low resources in terms of both computation and energy capacity.

This paper provides a survey of the main technological components needed to enable the interconnection among things as extracted by the current literature, research projects and standardization bodies.

Accordingly, the rest of the paper is organized as follows: Section II identifies the technologies related to IoT gateways and device management techniques. Section III provides an overview of the most important wireless technologies related to the IoT ecosystem and Section IV presents a survey of IoT protocols and a comparison among their features. Finally, Section V closes the paper with conclusion and outlook.

978-1-4799-5344-8/15/$31.00 © 2015 IEEE

II. IoT Gateways and Device Management

In this Section we provide a description of the role of the IoT gateway and the work done on device management inside some standardisation bodies. These two can be considered main components in order to enable the inter-operability in the IoT domain.

A. IoT Gateways

IoT gateways primarily act as the bridge to connect sensor networks with traditional communication networks, having capabilities such as protocol conversion and device management. Figure 1 gives an overview of IoT gateway functionalities resulting from the current technological trends, and we can see that gateways play an essential role to enable technological interoperability. Nevertheless, recent approaches allocate much more functionalities to IoT gateways so that all layers of the OSI model are involved and also services or applications will have a place there. IoT gateways are devices near the edge of future IoT environments, they have processing power and thus are able to process immediately the incoming data reducing significantly both the bandwidth required to send the raw data to the control center and the delay to create a response to a detected event. Hence, they will enable the implementation of paradigms like: *putting intelligence to the edge of the network*, *distributed data acquisition and control*, and *in-network processing*.

Additionally, full-featured operating systems running on powerful processors in the IoT gateways itself can manage communications, perform signal processing, and execute sophisticated control applications locally, rather than requiring instructions from a central service. Thus, IoT gateways will also contribute in enabling semantic inter-operability.

Existing technologies and solutions proceed towards the above described direction:

1) Gateway Features in the Home Gateway Initiative (HGI): The HGI develops specifications and standards for home gateway equipments for the residential market segment and has published a set of technical requirements in the form of a residential profile [3]. These address, amongst others, functional and management concerns, such as connectivity,

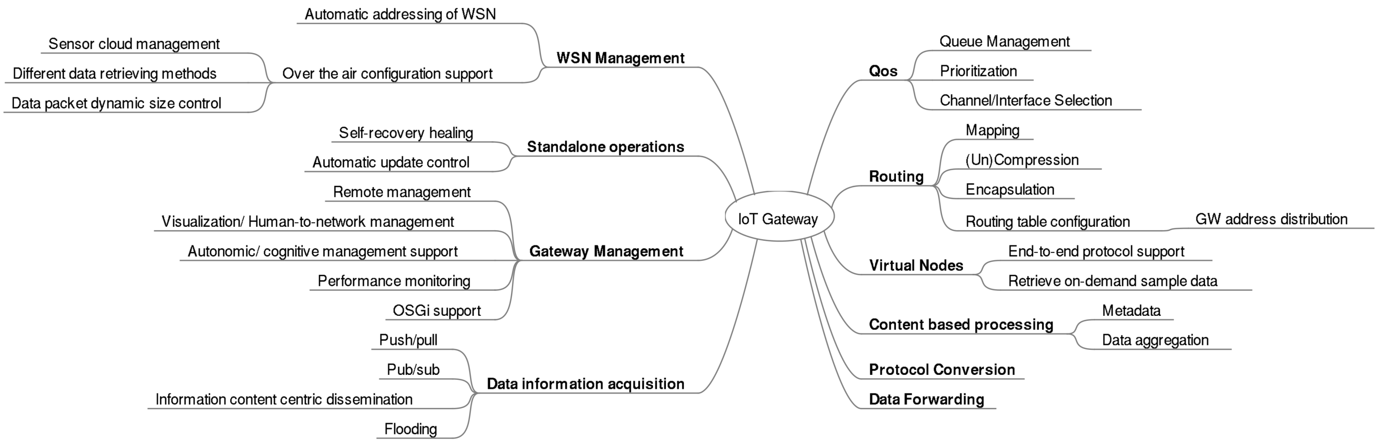

Fig. 1. IoT Gateway functionalities.

security, interworking to external service networks (i.e., 3GPP IMS), policies on exchanged content (e.g., parental control), guest services (e.g, WLAN hotspot), and remote device access. In particular, the HGI specification addresses the provision of third-party applications on top of the home gateway by defining the required capacities of a software execution environment. HGI work has considered the Open Services Gateway Initiative (OSGi) specification as an externally referenced and reused part of the HGI architecture. In real-life equipment, an OSGi instrumentation can provide parts of the specified HGI feature set.

2) Gateway Features in the Open Services Gateway Initiative (OSGi): The OSGi Alliance is a global consortium of industry stakeholders that develops open specifications to enable and promote the modular assembly of applications built with Java platform technologies. Therefore, the OSGi Alliance has defined a software framework to support modular applications for the Java platform. The OSGi framework architecture defines schemas for essential application metadata and specifies the details of dynamic deployment and life cycle management aspects on the basis of distinct application modules termed bundles. An OSGi application is dynamically assembled during runtime by binding (i.e., interconnecting through appropriate interfaces) together appropriate bundles registered in the OSGi service registry [4]. In its latest version (i.e., Release 5), OSGi specifications provide the *OSGi/Minimum-1.2*, the *Java Micro Edition (ME) Connected Device Configuration (CDC) 1.1/Foundation-1.1* execution environments, as well as standard OSGi mechanisms for on demand deployment.

B. Device Management

Given that IoT will be characterised by a significantly large number of devices, effort has been invested in identifying suitable device management solutions for IoT. These efforts are coordinated by standardisation work at the regional (e.g., ETSI) and global (e.g., ITU) levels. As a result, the device management frameworks developed by the Open Mobile Alliance (OMA) and the BroadBand Forum (BBF) have been accredited as the standard technology solutions for device management in IoT.

1) Device Management in OMA: The OMA has published its Device Management (DM) specifications designed for the management of mobile devices and supporting the following use cases:

- *Device Provisioning*, addressing the configuration of a mobile device and enabling/disabling selected features.
- *Device Configuration*, supporting changes to the operational settings of a mobile device.
- *Software Upgrade*, supporting the remote (i.e., over-the-air) installation of (application and/or system) software on a mobile device.
- *Fault Management*, supporting the reporting of error conditions detected at the mobile device and remotely querying its status.

In addition, OMA is developing enabler specifications like *Remote Entity Management Service Capability*, *Gateway Management Object* [5], *OMA Lightweight M2M*, *M2M Device Classification*, *Device API Connection Manager*, and *Device API DM Client-Side API Framework*.

2) Device Management in BBF: The BBF has defined the TR-069 specification for an application layer protocol termed CPE WAN Management Protocol (CWMP) to manage Customer Premises Equipment (CPE) over a (wireline) Wide Area Network. TR-069 defines the interactions between a CPE and an Auto-Configuration Server (ACS) entity to support the secure auto-configuration of a CPE and establish the framework for the following CPE management use cases:

- *Auto-Configuration and Dynamic Service Provisioning*, supporting a runtime renegotiation of the current provisioning.
- *Software/Firmware Image Management*, supporting the management software/firmware of the CPE.
- *Software Module Management*, enabling an ACS execute life cycle management operations (i.e., install, update, uninstall).
- *Status and Performance Monitoring*, enabling a CPE to

make available information that the ACS may use to monitor the status of the CPE.

- *Diagnostics*, enabling a CPE to make available information that the ACS may use to diagnose and resolve connectivity or service issues.

TR-069 is being reused as protocol solution for remote management within the architecture work undertaken in other bodies, including the Home Gateway Initiative (HGI), the Digital Video Broadcasting (DVB) Forum and the WiMAX Forum.

III. Wireless Communication

Technological interoperability, resilience and reliability are existing challenges which have to be overcome in order to enable IoT. This Section gives a short overview of important wireless communication technologies that already have the potential to play an essential role in connecting smart things to the IoT. On the other hand, this Section does not describe communication technologies that are currently under development or in draft status and might have their place in this environment in the future.

A. WiMAX and IEEE 802.16

The WiMAX (Worldwide Interoperability for Microwave Access) Forum is an industry-led, non-profit organization that certifies and promotes the compatibility and interoperability of broadband wireless products based on the IEEE Standard 802.16. It is designed as wireless broadband access technology and can support up to 1 Gbit/s. Current work is done on IEEE 802.16p which is an enhancement to support machine-to-machine applications. It introduces enhancements to Medium Access Control (MAC) which, along with appropriate modifications to the Orthogonal Frequency Division Multiple Access (OFDMA), enable lower power consumption at the communication device. Typical application areas for IEEE 802.16p will be applications for observation and control purposes (e.g., as in the domain of industrial automation).

B. IEEE 802.15.4

IEEE 802.15.4 was defined as a standard for the physical layer and the media access control layer in wireless personal area networks. The physical layer is typically responsible for channel selection and signal management. IEEE 802.15.4 also specifies the assigned frequencies for communication. On top of the physical layer, IEEE 802.15.4 also defines the MAC layer properties like time slot coordination and node associations. Other prominent standards like ZigBee, WirelessHART, and 6LoWPAN are based on IEEE 802.15.4. IEEE 802.15.4 supports peer-to-peer as well as star topologies.

C. IETF 6LoWPAN

6LoWPAN is a protocol standard for IP based data transmission in low power wireless personal area networks. The most important component is a 6LoWPAN adaption layer located on the network layer of the OSI layer model. It is responsible for typical tasks for network layer protocols, e.g., header compression, fragmentation and de-fragmentation of packets as well as routing. Routing capabilities of 6LoWPAN supports routing in mesh networks supporting also mobility of nodes. 6LoWPAN is often used to establish a simplified IPv6 protocol based communication of resource constrained devices over IEEE 802.15.4.

D. ZigBee

The ZigBee standard was developed for short range communication (i.e., in the order of 10 to 100 meters) of wireless networks based on IEEE 802.15.4. The ZigBee stack spans from the network layer up to the application layer of the OSI model and includes a basic security model. ZigBee is optimized for low power applications and supports different application profiles (e.g., ZigBee Home Automation, ZigBee Health Care, ZigBee Building Automation, etc.), with each profile engineered according to the requirements of a particular application domain.

In Table I we provide a comparison of the wireless technologies. As can be seen, each proposed technology has unique characteristics, consequently the choice and the combination of the most appropriate ones should be based on a good understanding of the architecture and requirements for each target system.

IV. Protocols

In this Section we provide a description of the most important protocols that are being proposed for IoT and M2M communications. Specifically, we categorize the protocols into the following groups: application protocols, payload container protocols, messaging protocols and legacy protocols. Nevertheless, to have a comprehensive view, first we describe approaches to address the challenge of semantic inter-operability, as well as the two architectural styles of reference for the M2M domain: REST and WS-*.

A. Metadata and Semantics

Given the infrastructure for disseminating IoT data, overcoming the challenge of semantic inter-operability will largely depend on the existence of common information models and the level of support for metadata management and semantic annotation. These concerns have been addressed by the W3C Incubator Group on Semantic Sensor Networks (SSN) and their work includes recommendations on methods to semantically enable applications developed according to existing standards such as its own developed ontology for describing sensors and networks thereof [6] or the Open Geospatial Consortium's (OGC) standards for Sensor Web Enablement (SWE) described subsequently. OGC has defined a service-oriented architecture that enables applications to discover available sensors and to consume data acquired through them in an interoperable manner [7]. The OGC SWE is based on web technologies and enables the following features:

- The discovery of sensors and their associated observations, through the Observations and Measurements (O&M) [8], [9] standard.

	WiMAX/IEEE 802.16	**IEEE 802.15.4**	**IETF 6LoWPAN**	**ZigBee**
Layer	PHY/MAC	PHY/MAC	Network	Network and upper layers
Range	30Km	1Km	-	(see 8021.5.4)
Application	Wide Area/Industrial Automation	(see ZigBee)	Automation/Factory Environment	Monitor/Control
Power Consumption	Medium/High	Low	Low	Low

TABLE I
COMPARISON OF WIRELESS TECHNOLOGIES FOR IoT.

- The exchange of observation data acquired through sensors, through the Sensor Observation Service (SOS) [10] standard.
- The processing of sensor observations, through the Sensor Observation Service (SOS) standard taking into account geometric, dynamic, and observational properties of individual sensors as well as complex sensor systems defined by the OGC Sensor Model Language (SensorML) [11].
- The tasking of sensors and sensor systems, through the Sensor Planning Service (SPS) [12] standard.

B. REST and Web Services

1) REST: REpresentational State Transfer was developed by W3C Technical Architecture Group and follows the principal that every physical object and/or logical entity is a *resource* that has a particular *state* that can be “manipulated”. A resource that is accessible via HTTP URI gives access to its data via GET and accepts inputs via PUT. REST aims on minimizing latency and network communication, while at the same time maximizing the independence and scalability of component implementations [13]. The effort needed to develop applications, especially in the IoT domain, can be greatly reduced since REST adopts a much lighter tool chain than other service oriented architectures [14].

2) WS:* Web Services is a set of protocols and specifications including several standards, such as: message specifications (SOAP, WS-Addressing, WS-Enumeration), metadata specifications (WSDL, WS-Discovery, WS-Policy), security specifications (WS-Federation, WS-Security, WS-Trust), reliable messaging specifications (WS-ReliableMessaging), XML specifications (XML), management specifications (WS-Management). WS-* declare their functionality and interfaces in a WSDL file. Client requests and service response objects are encapsulated using the SOAP protocol and are transmitted over the network using the HTTP protocol.

Recent studies [15] [16] show that in the context of IoT, RESTful Web Services have many advantages over Web Services (i.e., SOAP), such as less overhead, less parsing complexity, statelessness, and tighter integration with HTTP. But, in case of strong security and quality of service requirements WS-* offers a competitive advantage.

C. Application protocols

Application protocols rely on the underlying protocols and are used to establish device-to-device data exchange. A well suited candidate for an application protocol in IoT environments is RTPS (Real-Time Publish-Subscribe). It was specifically developed to support the unique requirements of data-distributions in industrial automation and was in approved by IEC as part of the Real-Time Industrial Ethernet Suite (IEC-PAS-62030). RTPS is designed to run over an unreliable transport such as UDP and it provides a publish-subscribe protocol. A close synergy exists between the OMG Data-Distribution Service (DDS) and the RTPS, both in terms of the underlying architecture and features. The DDS for real-time systems is the only open standard for messaging that supports the unique Quality of Service requirements of both enterprise and real-time systems and it often provides the only standards-compliant alternative to proprietary or custom integration approaches [17].

D. Payload container protocols

Two well known candidates for payload container protocols are the Simple Object Access Protocol (SOAP) and Constrained Application Protocol (CoAP). They define some basic message types and then encapsulate the payload in the message body relying on the underlying protocols for the message length and so on.

1) SOAP: The SOAP specification [18] is currently maintained by the XML Protocol Working Group of the World Wide Web Consortium and specifies how to exchange structured information in the context of web services by using XML for its message format. Usually it relies on HTTP and SMTP for message negotiation and transmission.

2) CoAP: CoAP [19] is an application layer protocol standardized by the Internet Engineering Task Force (IETF) Constrained RESTful environments (CoRE) Working Group. It is designed to emulate the REST features of HTTP but in a way that is more constrained environment friendly. With CoAP over UDP it is possible to disable the reliability features and not store any state about connections. Moreover, CoAP can easily translate to HTTP for simplified integration with the web, while also meeting specialized requirements such as multicast support, very low overhead and simplicity.

In Table II we provide a comparison of SOAP and CoAP features.

E. Messaging Protocols

Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT), eXtensible Messaging and Presence Protocol (XMPP), and Java Message Service (JMS) are well known representatives for messaging protocols and we give a brief overview of them.

	SOAP	**CoAP**
Architecture Style	WS-*	REST
QoS	Addressed by W3C WG	No
Real-time	No	Near real time
Protocol	HTTP, SMTP, TCP, JMS	HTTP/UDP
Messaging format	XML	not-defined
Standards	W3C recommendation	IETF (CoRE) WG upcoming standard
Mobile Support	Yes	Yes
Security	Addressed by WS-Security	DTLS/IPSEC

TABLE II
COMPARISON OF SOAP AND COAP.

1) AMQP: AMQP is an open standard application layer protocol for message-oriented middleware [20], that provides flow controlled communication with message-delivery guarantees and topic-based publish-and-subscribe messaging. In addition, authentication and/or encryption based on SASL and/or TLS is supported. AMQP requires an underlying reliable transport layer protocol such as TCP.

2) MQTT: Originally invented in 1999, MQTT currently is part of the OASIS standardization process to make MQTT an open, simple and lightweight standard protocol for M2M telemetry data communication [17]. Using the TCP/IP layer for basic network connectivity, MQTT enables a publish/subscribe messaging model in a lightweight way and can also be used for mobile applications. It is designed for small code footprints (e.g., 8-bit, 256KB ram controllers), low power, low bandwidth, high-cost connections, high latency, variable availability, and negotiated delivery guarantees. MQTT is estimated to be running on thousand of devices (e.g., Healthcare Industry Segment, Energy Industry Segment, Facebooks Messenger application). MQTT can be implemented in devices with less than 64kb of RAM. In comparison to HTTPS, MQTT shows a faster throughput, it requires less power and less network overhead [21]. On the other hand, it does not support transactions, connection security, discovery of clients or servers, and fragmentation of messages. Furthermore, it is not extensible, i.e., it requires a new protocol revision to evolve its capabilities.

3) XMPP: XMPP has been formalized by IETF in RFC3920 (now updated in RFC6120 [22]). It is an open XML-based protocol for near real-time messaging, presence and request-response services using TCP as transport [17]. XMPP is currently used in e.g., Jabber.org, Cisco/WebEx and Google Talk. Microsoft provides an XMPP interface to its Microsoft Messenger Service and has XMPP gateways integrated in their messaging systems; Facebook presents an XMPP interface to its clients for enabling some features. XMPP is easily extensible and can directly interact with other objects running XMPP. Moreover, it can store contents if the receiving entity is in sleep mode or offline using its *store and push* mechanism. The use of TCP might lead to some overhead if compared with other messaging protocols.

4) JMS: As part of the Java Platform Enterprise Edition, Java Message Service (JMS), specified in JSR914, is one of the most widely used messaging technologies. It is a message-centric API and allows application components to create, send, receive, and read messages. Messages between two or more clients can be exchanged loosely coupled, reliably, and asynchronously via point-to-point and publish-and-subscribe. The main limitation of JMS is that it is a Java API standard only and does not define a wire protocol. Therefore, JMS implementations from different vendors may not interoperate.

In Table III we provide a comparison of the messaging protocols according to some criteria identified with the help of the work in [21].

F. Legacy Protocols

Legacy protocols are available in other domains like building automation (e.g., BACnet [23], KNX European [24]–[26] and ISO [27]–[29]) or home networking with service discovery technologies (e.g., Jini [30] and UPnP [31]).

V. CONCLUSIONS

Taking into consideration the current growing trend, it can be deduced that IoT will emerge more and more in the near future. Machine-to-machine communication will facilitate the massive increase of data in both local and wide area networks. This will enable the evolution of Big Data systems but also the need to secure the data.

Shared infrastructures with common standards will be needed and the any time - any place communication will be extended by a third dimension - the any thing communication.

Concluding, as more functionalities are pushed out from central servers to devices close to where the data are generated and consumed, we see the need to investigate further the analytics at the edge and the in-network processing concepts. These concepts, among others, will allow the pre-processing of data and thus will lead to significant bandwidth reduction, better support of privacy, improved scalability, as well as minimize the response time for a required action.

To this purpose, in this paper we have presented and compared the most discussed and promoted technologies on M2M and IoT produced by standards regulation, industry and research projects that will enable the inter-operability among things being a significant factor to the commercial success of IoT products, services and applications.

The choice and the combination of the most appropriate ones should be based on a good understanding of the architecture and requirements for each target system.

	AMQP	**MQTT**	**XMPP**	**JMS**
Abstraction	Pub/Sub	Pub/Sub	Pub/Sub	Pub/Sub
Architecture Style	P2P/Brokered	Brokered	P2P/Brokered	Brokered
QoS	Yes	Yes	Yes	Yes
Interoperability	Yes	Partial	Yes	No
Real-time	No	No	Near real-time	No
Transport	TCP	TCP	TCP	Not specified, typically TCP
Standard	OASIS AMQP	Proposed OASIS standard	IETF	JCP JMS standard
Licensing	Open Source and Commercially Licensed	Open Source and Commercially Licensed	Open Source and Commercially Licensed	Open Source and Commercially Licensed
Mobile Support	Yes	Yes	Yes	Dependent of the Java capabilities of the OS
Security	SASL authentication, TLS for data encryption	Simple Username/Password Authentication, SSL for data encryption	SASL authentication, TLS for data encryption	Vendor specific but typically based on SSL or TLS. Commonly used with JAAS API

TABLE III
COMPARISON OF MESSAGING PROTOCOLS.

REFERENCES

[1] *Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*, 2012. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6424332

[2] I. G. Smith, O. Vermesan, P. Friess, and A. Furness, *The Internet of Things 2012 New Horizons*, I. G. Smith, Ed., 2012. [Online]. Available: www.internet-of-things-research.eu/pdf/IERC_Cluster_Book_2012_WEB.pdf

[3] H. G. Initiative, *Home Gateway Technical Requirements: Residential Profile*, Home Gateway Initiative (HGI) TR TRRP, 2008.

[4] OSGi, *OSGi Release 5 Specifications*, OSGi Alliance Std., 2012. [Online]. Available: http://www.osgi.org/Specifications/HomePage

[5] OMA, *OMA Gateway Management Object (GwMO) Architecture*, Open Mobile Alliance Std., 2013. [Online]. Available: http://technical.openmobilealliance.org/Technical/release_program/docs/CopyrightClick.aspx?pck=GwMO&file=V1_1-20130625-C/OMA-AD-GwMO-V1_1-20130625-C.pdf

[6] M. Compton, P. Barnaghi, L. Bermudez, R. Garca-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The {SSN} ontology of the {W3C} semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, no. 0, pp. 25 – 32, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570826812000571

[7] OGC, *Sensor Web Enablement Architecture*, OGC Std. [Online]. Available: http://portal.opengeospatial.org/files/?artifact_id=29405

[8] ——, *Observations and Measurements*, OGC Std. [Online]. Available: http://www.opengeospatial.org/standards/as

[9] ——, *Observations and Measurements - XML Implementation*, OGC Std. [Online]. Available: http://www.opengeospatial.org/standards/om

[10] ——, *Sensor Observation Service*, OGC Std. [Online]. Available: http://www.opengeospatial.org/standards/sos

[11] ——, *Sensor Model Language (SensorML)*, OGC Std. [Online]. Available: http://www.opengeospatial.org/standards/sensorml

[12] ——, *Sensor Planning Service*, OGC Std. [Online]. Available: http://www.opengeospatial.org/standards/sps

[13] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Trans. Internet Technol.*, vol. 2, no. 2, pp. 115–150, May 2002. [Online]. Available: http://doi.acm.org/10.1145/514183.514185

[14] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M Communications: A Systems Approach*. Wiley, 2012.

[15] D. Guinard, I. Ion, and S. Mayer, "In search of an internet of things service architecture: Rest or ws-*? a developers' perspective," in *MobiQuitous*, 2011, pp. 326–337.

[16] Z. Shelby, "Embedded web services," *Wireless Commun.*, vol. 17, no. 6, pp. 52–57, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1109/MWC.2010.5675778

[17] oneM2M, "oneM2M TR-0009 v0.4.0 - Technical Report - Protocol Analysis ," ftp://ftp.onem2m.org/.../oneM2M-TR-0009-Protocol_Analysis-V0_4_0.doc, 2014, [Online; accessed 30-January-2014].

[18] W3C, "SOAP Version 1.2," http://www.w3.org/TR/soap12/, 2007, [Online; accessed 30-January-2014].

[19] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP)," Working Draft, IETF Secretariat, Fremont, CA, USA, Tech. Rep. draft-ietf-core-coap-07.txt, Jul. 2011. [Online]. Available: http://www.rfc-editor.org/internet-drafts/draft-ietf-core-coap-07.txt

[20] OASIS, "OASIS Advanced Message Queuing Protocol (AMQP)," https://www.oasis-open.org/committees/amqp, 2014, [Online; accessed 30-January-2014].

[21] A. Foster, "Messaging Technologies - A Comparison Between DDS, AMQP, MQTT, JMS and REST ," 2013.

[22] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," http://www.rfc-editor.org/info/rfc6120, 2011, [Online; accessed 30-January-2014].

[23] ANSI/ASHRAE, *ANSI/ASHRAE Standard 135*, ANSI/ASHRAE Std.

[24] CEN, *EN 13321-1:2012; Open Data Communication in Building Automation, Controls and Building Management - Home and Building Electronic Systems - Part 1: Product and system requirements*, CEN Std. [Online]. Available: http://www.itu.int/en/ITU-T/focusgroups/m2m/Pages/default.aspx

[25] ——, *EN 13321-2:2012; Open Data Communication in Building Automation, Controls and Building Management - Home and Building Electronic Systems - Part 2: KNXnet/IP Communication*, CEN Std.

[26] CENELEC, *EN 50090-1:2011; Home and Building Electronic Systems (HBES) - Part 1: Standardization structure*, CENELEC Std.

[27] ISO, *International Standard ISO/IEC14543-3-1; Information technology – Home Electronic Systems (HES) Architecture – Part 3-1: Communication layers – Application layer for network based control of HES Class 1*, International Standards Organization (ISO) Std.

[28] ——, *International Standard ISO/IEC14543-3-3; Information technology – Home electronic system (HES) architecture – Part 3-3: User process for network based control of HES Class 1*, International Standards Organization (ISO) Std.

[29] ——, *International Standard ISO/IEC14543-5-7; Information technology – Home electronic system (HES) architecture – Part 5-7: Intelligent Grouping and 3 Resource Sharing – Remote Access System Architecture*, International Standards Organization (ISO) Std.

[30] S. Microsystems, *Jini Architecture Specification*, Sun Microsystems Std. [Online]. Available: http://river.apache.org/doc/specs/html/jini-spec.html

[31] U. Forum, *Device Architecture, Version 1.1*, UPnP$^{\text{TM}}$ Forum Std., 2008. [Online]. Available: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf