On interworking between rapidly evolving Internet of Things and Open Web Platform

Markus Isomäki
Nokia, Finland
Email: markus.isomaki@nokia.com

Teemu Savolainen
Nokia, Finland
Email: teemu.savolainen@nokia.com

Art Barstow
Nokia, USA
Email: art.barstow@nokia.com

***Abstract*—In the next few years, a large number of new types of communicating objects will emerge. They will come in the form of sensors, actuators and tags that interact with the physical world, comprising the "Internet of Things" (IoT). These *smart objects* will be used in both industrial and consumer oriented applications. Most of the objects will use the Internet as the backbone for their communication, but especially for the most constrained objects, mainstream Internet access technologies and application protocols are not always feasible, e.g. due to power consumption. For that reason, a lot of new low power radio technologies and application protocols have been developed. Some of these protocols allow direct Internet connection, some assume a gateway between the Internet and smart objects. There are many interesting use cases regarding how browsers and Web applications should be able to interact with such objects. However, to make this feasible, new protocols and APIs may be required in the browsers. This paper introduces some of the use cases, discusses the different browser to smart object communication options and provides a survey of the emerging IoT protocols and frameworks. Finally, a few concrete proposals for future work in W3C and elsewhere are made.**

I. Introduction

The next few years will witness the emergence of billions of new connectable devices and objects. They will come in various forms and usages: Sensors, actuators, beacons, tags, home appliances and wearable gadgets, to name just a few. Collectively they are sometimes called "Smart Objects". Another term, the "Internet of Things" (IoT), refers to the virtual Internet-type structure these uniquely identifiable objects form, as well as to them actually being physically connected to the Internet. Some of the new devices can indeed perfectly well use the current Internet in the same way as smartphones and tablets use it today. For instance, most home appliances can simply and conveniently be connected to home's pre-existing Wi-Fi network.

However, the traditional Internet access technologies and protocols do not suit all device types and use cases. First of all, many of the devices are extremely constrained in terms of power and computation capacity including CPU power and memory. A good classification of different types of constrained devices and their capabilities is presented in [1]. A sensor powered by a coin cell battery or a small solar panel can only use an extremely low-power radio technology such as Bluetooth Low Energy. For the same reasons it may not be practical for it to run mainstream protocols such as HTTP and TLS either, but use something more lightweight. Secondly, not all communication needs to be network based at all, but may happen directly between devices. For instance, a smartphone may receive an advertisement from a beacon on a wall, and connect directly to it to get further information.

The Open Web Platform (OWP) has become an almost ubiquitous standard for Internet application development, and its capabilities are constantly evolving. The Web will interface with smart objects and Internet of Things in many ways. The term "Web of Things" (WoT) has been coined to refer to the possibilities of the Web in the Internet of Things revolution. Some of this has already happened based on the Web's traditional capabilities. For instance, web servers have been embedded in devices such as coffee machines already for decades. Similarly, web browsers have been introduced in more and more environments, and no doubt eventually even a small wearable device such as a watch may have one. Clearly the Web has demonstrated its low-end scalability and the progress will continue in a linear manner.

There are, however, also scenarios that may require entirely new type of development to become feasible. In this paper we will look at some of the potential scenarios. The focus will be on how a device such as a smartphone or tablet, equipped with a fully capable web browser, should be able to interact with smart objects with minimal trouble from the users. For the billions of Smart Objects the "plug and play" paradigm is no longer enough - what is needed is simply "play".

We attempt to identify what types of new capabilities the browser or the smart objects would require. Rather than looking at this just from the theoretical perspective, we also give an introduction to actual IoT related technologies and standards under development and try to see how they fit with the Web's technologies and standards.

Section II provides a survey of IoT technologies and standards that could be relevant in the Web context. Section III analyses how Web application clients can communicate with smart objects. The paper is concluded in Section IV with discussion on possible next steps.

II. Internet of Things technology survey

The requirements stemming from many constraints have triggered lots of research and standardization activities in the areas of low power radios and more optimized communication protocols. Some of these activities are complementing, but some are competing. In here we show a small survey of ongoing activities in some organizations in no particular order.

A. Low power radio technologies and considerations

There are many radio technologies aimed at IoT. Their main characteristic is low power consumption, compared to e.g. the mainstream Wi-Fi or cellular technologies. Consequently they are often limited in bitrate or physical range. These technologies include at least IEEE 802.15.4 [2] (used e.g. in ZigBee [3]), Bluetooth Low Energy [4], Z-Wave and 802.11ah low-power variant of Wi-Fi under development [5]. They each have their strongholds, for instance Bluetooth Low Energy is popular in devices such as smartphones, tablets and their peripherals, but so far much less widely deployed in actual networking situations such as home networks.

From networking and application perspective the key question on these technologies is whether they support IP based communication or not. If they do, devices using them can be connected directly to the Internet, and can potentially use any Internet protocols end-to-end. If not, a specific protocol translation gateway needs to be deployed in between. Currently the situation is mixed. IPv6 over IEEE 802.15.4 [6] is a mature standard, but it is just one alternative, and other Zigbee protocol profiles are often used in practice. In Bluetooth Low-Energy the IPv6 support is still work in progress [7], and all today's products are based on non-IP based alternatives.

Another important consideration is performance, especially related to power consumption. For instance, if a Bluetooth Low Energy connected sensor reports its reading periodically, the overall packet sizes used in the reporting transactions may contribute significantly to the power consumption within each period. A few dozens of extra bytes may drop the operating time of the sensor considerably. This means even HTTP with its TCP connection setup may be too much.

For these reasons, a lot of specialized application protocols have been developed for IoT or constrained device purposes.

B. Application (and transport) protocols and frameworks

Active standardization work for protocols for IoT is ongoing in multiple forums.

Internet Engineering Task Force (IETF) has been developing UDP and DTLS using Constrained Application Protocol (CoAP) [8], which is a lightweight binary encoded protocol providing very similar RESTful properties as HTTP. Alternative transports are being defined, such as CoAP over TCP, WebSockets, and SMS. IETF is also working in access control and authentication and network management topics, and not to forget HTTP 2.0 [9], which will bring optimizations to the HTTP 1.1.

Open Mobile Alliance (OMA) has previously defined protocols for M2M communications using HTTP, and has also adopted IETF's CoAP as protocol for Lightweight M2M [10]. The OMA LWM2M can use both UDP and SMS as transport.

Newly founded *AllSeen Alliance* is promoting Qualcomm-originated AllJoyn software framework [11] for "Internet of Everything". The AllJoyn implements a remote procedure call framework that works over distributed AllJoyn bus using UDP, HTTP, Bluetooth and possibly more. The AllJoyn is mostly for local ad-hoc communications, but could provide long distance communications by utilizing Rendezvous Servers.

Organization for the Advancement of Structured Information Standards (OASIS) has been recently promoting IBM-originated Message Queuing Telemetry Transport (MQTT) [12] as a protocol for machine-to-machine type of communications. The MQTT uses publish-subscribe paradigm for allowing nodes to share information via MQTT server. MQTT uses reliable stream protocols as transports, namely TCP or WebSockets. OASIS has also worked on protocols such as Advanced Message Queue Protocol (AMQP).

Object Management Group (OMG) is defining Data-Distribution Service for Real-Time Systems (DDS) [13], which is a solution providing publish-subscribe communications using DDS data bus. Communications occur over UDP, but can also be done at least with TCP and via shared memories. Work is also ongoing for Web-Enabled DDS.

OneM2M is working to build common M2M Service Layer in collaboration of over two hundred participants, including ARIB, ATIS, CCSA, ETSI, TIA, TTA and TTC. The work includes at least bindings for CoAP, HTTP, and MQTT.

Bluetooth SIG and ZigBee Alliance are providing profile-based solutions for talking to Smart Objects, and then there are also other organizations and companies providing their solutions for the IoT.

The bottom line is that there are currently a large number of different frameworks and protocols either in use or under development. It is too early to say which will succeed on the market. From the Web perspective IP-based protocols and especially CoAP, due to its HTTP-like RESTful design, would be the easiest fit.

III. Web of Things use cases and communication patterns

A. Use cases

Web based clients and smart objects may interact in multitude of ways, for example:

- Setup and configuration: Most smart objects do not have a proper user interface to set them up. Browsers are often used for that purpose. Examples include home appliances and wearable devices.
- Providing a user interface: In addition to setup and configuration tasks, a browser can provide an actual user interface for communicating with the smart objects, for instance displaying sensor values or controls for actuators. An example could be a home automation system.
- Contextual proximity sensing: When a mobile device moves around, it can receive information from beacons in its proximity, or even connect to smart objects in its proximity to gain more contextual information. This information could be utilized by Web based applications. An example would be a beacon advertising an identifier, based on which relevant venue or product information could be shown by the browser.

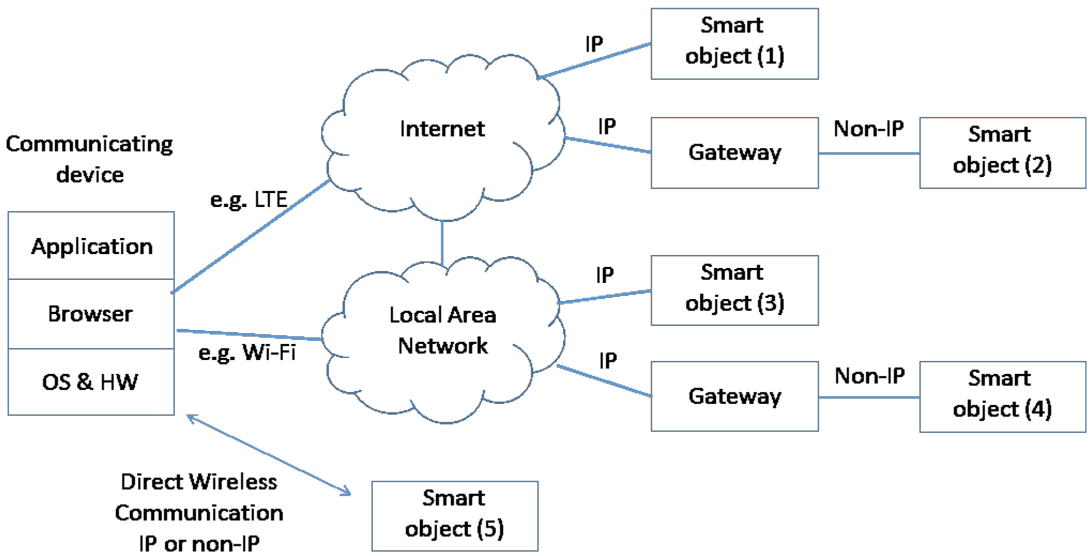

Figure 1: Web application communicating with smart objects

In some cases the communication would be local, but in others remote access should be an option too. For instance if a Web application/browser is used to interact with the home automation system it could happen within home (e.g. adjusting lights while sitting in the living room) or remotely from anywhere (e.g. adjusting heating when traveling). Another aspect is authorization: In some cases only authorized users should be able to access the smart object while in others at least "read" access could be provided to anyone. In lot of the use cases discovery of the smart objects and their capabilities is a crucial step before the actual communication happens.

For any of these use cases to be possible, there needs to be a way how a browser or a Web application can communicate with the resources/services provided by the Smart Objects.

B. Communication patterns

Figure 1. depicts the general framework on how web based client applications can communicate with smart objects. In this case we are specifically interested in objects that are constrained, i.e. they cannot trivially run a full Web protocol stack including HTTP, WebSockets, TLS and TCP.

The web client application runs within the Communicating device on the left. It is connected to the Internet either via a Wi-Fi based local area network connection or via a cellular radio network. The device also supports direct device-to-device radio connections using e.g. Bluetooth Low Energy.

Smart objects (1) and (2) are reachable only over a wide area Internet connection, whereas objects (3) and (4) are located in the same local area network the communicating device is using. This means that the communicating device may be able to discover these objects and their capabilities by using some local area discovery mechanism such as UPnP, multicast DNS Service Discovery, CoAP or AllJoyn. The discovery capability can be offered to the Web application via an API provided by the browser. Object (5) is not Internet connected but the communicating device can discover and communicate with it using its direct radio connection capability (Bluetooth Low Energy). In order for the web application to be able to use that capability, another browser API is needed.

Objects (1) and (3) support direct Internet end-to-end communication, while objects (2) and (4) are only reachable via a protocol translation gateway.

Based on this framework we can summarize the three main ways how the Web application can communicate with the objects:

1) Using its normal Internet connection via a protocol translation gateway: This is applicable to objects (2) and (4). No new protocol or API support may be required from the browser. The gateway can translate HTTP and/or WebSockets to IP (e.g. CoAP) or non-IP (e.g. Bluetooth Low Energy Generic Attribute Profile) based protocol. The downside of this approach is that a gateway is always needed. Whether this is acceptable depends on the application. If the gateway is only used for translating protocols, direct communication might be preferable.

2) Using its normal Internet connection with an end-to-end protocol: This is applicable to objects (1) and (3). To enable this the browser would have to support new protocols specifically designed for Smart Object communications, such as CoAP. The aim in this approach is to get rid of pure protocol-translating gateways and simplify deployment.

3) Using a direct device-to-device radio connection: This is applicable to object (5). If these connections support IP, the same protocols used as used over the Internet should be sufficient. However, at least currently many of them are not IP-based. To enable them the browser should support APIs that allow applications to use device-to-device communication technologies, such as Bluetooth Low-Energy. In some cases it would be enough just to be able to receive the information sent by the constrained device, for instance a Bluetooth Low-Energy tag. In others actual two-way communication would be needed.

IV. Bridging the gap

With the multitude of organizations specifying protocols for IoT it is clear that there will not be a single protocol for IoT anytime soon. All of the protocols being developed have their best suited scenarios and usage patterns, and hence are not trivially replaced by generic solutions. Interoperability between different protocols can be provided by cross-protocol proxies, as we saw in Section III.

The increased trend to use IP even in constrained devices and low-bitrate radio links will help to enable browser to Smart Object communication. However, for direct communication over the Internet or device-to-device, use of lightweight application and transport protocols would still be beneficial if not often even mandatory, and this would be a clear area to extend the current Open Web Platform. Support for non-IP based device-to-device communication by offering APIs, or an architecture that would allow addition and use of IoT APIs for non-IP protocols, would be valuable especially in the short term. Whatever protocols are used for interacting with Smart Objects, the user and developer experience has to be simple and coherent. A generic architecture for IoT APIs could be helpful on that.

Perhaps, at this point, the main philosophical question is whether the IoT technologies should become more web-like by ubiquitously adopting IP and HTTP, and where not possible utilize translation proxies, or whether browsers and Open Web Platform in general should start accommodating Smart Objects' special needs by supporting some set of CoAP, AllJoyn, MQTT, Bluetooth Low-Energy profiles and other protocols designed on purpose for Smart Object communications.

A. Concrete proposals

The authors believe that at least the following concrete activities would be useful for allowing Web applications to better interact with Smart Objects. Some of the activities are ongoing, while some would be new:

In OWP and Browsers:

- Service discovery API including IoT oriented protocols such as CoAP. This work could be done as an extension to the W3C Network Service Discovery API [14].
- APIs for key IoT communication technologies on mobile devices, especially Bluetooth Low Energy. Some work on a Bluetooth API has been planned by the W3C System Applications WG [15], but it is not clear if the Low Energy part of Bluetooth is covered.
- API for CoAP allowing Javascipt applications to use it in the same way as HTTP. There is no work on this currently in W3C.

In Smart Objects / IoT:

- HTTP and/or CoAP based RESTful interfaces to resources and services offered by the Smart Objects, including initial setup. There are some standards efforts in the IETF and IP for Smart Objects Alliance (IPSO), but most of this would be implementation rather than standardization.

References

[1] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained Node Networks", draft-ietf-lwig-terminology-07, IETF, Feb 2014, available at http://tools.ietf.org/id/draft-ietf-lwig-terminology-07.txt

[2] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE Std 802.15.4™-2011, 5 Sep 2011, available at http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf

[3] ZigBee Alliance, "ZigBee Specification", ZigBee Document 053474r17, 17 Jan 2008, available at https://www.zigbee.org/Specifications/ZigBee/download.aspx

[4] BLUETOOTH Special Interest Group, "BLUETOOTH Core Specification Version 4.1", 3 Dec 2013, available at https://www.bluetooth.org/en-us/specification/adopted-specifications

[5] IEEE, "P802.11ah - Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment- Sub 1 GHz License-Exempt Operation", under development as of Apr 2014, see status at https://standards.ieee.org/develop/project/802.11ah.html

[6] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks", IETF, Sep 2007, available at http://www.ietf.org/rfc/rfc4944.txt

[7] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH Low Energy", draft-ietf-6lo-btle-00, IETF, Nov 2013, available at http://tools.ietf.org/id/draft-ietf-6lo-btle-00.txt

[8] Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18, IETF, Jun 2013, available at http://tools.ietf.org/id/draft-ietf-core-coap-18.txt

[9] M. Belshe, R. Peon, and M. Thomson, Ed., "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-11, IETF, Apr 2014, available at http://tools.ietf.org/id/draft-ietf-httpbis-http2-11.txt

[10] Open Mobile Alliance, "Lightweight Machine to Machine Technical Specification", OMA-TS-LightWeightM2M-V1_0-20131210-C, Candidate version 1.0, 10 Dec 2013, available at http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma

[11] AllSeen Alliance, "Introduction to the AllJoyn™Framework", 10 Dec 2013, available at https://allseenalliance.org/developer-resources/alljoyn/docsdownloads

[12] OASIS, "MQTT Version 3.1.1 Committee Specification Draft 01/Public Review Draft 01", 12 Dec 2013, available at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd01/mqtt-v3.1.1-csprd01.pdf

[13] Object Management Group, "Data Distribution Service for Real-time Systems Version 1.2", OMG Available Specification formal/07-01-01, Jan 2007, available at http://www.omg.org/spec/DDS/1.2/

[14] W3C, "Network Service Discovery API", W3C Working Draft, 20 Feb 2014, available at http://www.w3.org/TR/discovery-api/

[15] W3C, "System Applications WG Roadmap, phase 2", 15 Apr 2014, available at http://www.w3.org/2012/sysapps/#future