Introduction to the Web of Things

Dave Raggett, W3C

Progressive improvements in electronics is enabling widespread deployment of connected sensors and actuators (the Internet of Things) with predictions of 50 billion smart objects by 2020 (Cisco, 2011). This raises huge challenges for security, privacy and data handling, along with huge opportunities across many application domains, e.g. homes and buildings, retail, healthcare, electrical grids, transport, logistics, manufacturing, and environmental monitoring (IERC 2014).

The Internet of Things started with work on radio frequency identity tags (RFID) and expanded to connected sensors and actuators, along with many communication technologies designed for different purposes. IPv6 has greatly expanded the address space compared to IPv4 and makes it feasible to give each device its own IP address. Considerable effort has been made on supporting IP all the way to constrained devices, e.g. the 6LoWPAN protocol for wireless access, CoAP for lightweight message exchange and easy bridges to HTTP, and MQTT as a lightweight pub-sub protocol. Related work has enabled self organising mesh networks of devices (sensors networks).

Today, product silos are prevalent for the Internet of Things, and a sign of market immaturity. This is where open standards and open platforms can play a major role in enabling the growth of rich open ecosystems that can realise the huge potential benefits. Can we repeat the run away success of the World Wide Web and build a Web of Things? Turning that around, what can we learn from the Web in relation to encouraging an open market of services?

Much of the work on the Internet of Things (IoT) has focused on the technologies needed for constrained devices, long battery life, efficient use of wireless spectrum, sensor networks and so forth. Relatively little work has focused on applications and services. One of the challenges is the large number of IoT technologies and the way that these are continuing to evolve at a rapid pace. Examples include: ZigBee, NFC, Bluetooth, ANT, DASH7, EnOcean, Infrared, USB, Wireless USB, IEEE 1394 (Firewire), WiFi (ISO 802.11), M2M, LTN and Weightless. Some devices may support direct access to HTTP.

This motivates looking for ways to reduce the need for developers to have to deal directly with the IoT communication technologies through the use of appropriate abstractions that hide details best left to platform developers. Such abstractions can also help with the inevitable heterogeneous mix of vendors and device versions, enabling today's services to work with yesterday's devices and tomorrow's devices.

Another challenge is to provide end to end security. IoT devices are often very constrained, and this limits the options for handling security. A further problem is that many such devices may not be software upgradable, something noted by Internet pioneer David Clark in his keynote talk at IoT2014. Software upgrades are essential for addressing security flaws, as well as for updating devices to match changes in standards. This motivates the use of gateways that virtualise the IoT devices, bridging IoT and Web protocols, and offering higher levels of security, including the means to manage automatic updates.

With the increasing number of sensors, we all need to be confident that our privacy is being safeguarded. This implies end to end encryption against eavesdroppers, strong mutual authentication and support for access control and data handling according to the data owner's policies. With the ability to combine data from different sources, it becomes necessary to track provenance so that the originating data owner's policies can be applied to derived data. This in turn motivates work on techniques for static analysis of service logic and dynamic enforcement of policies.

With the very large numbers of sensors and actuators expected, it will be inevitable that some will fail, either through hardware faults, electrical noise or even botched upgrades. Services need to be designed to be resilient in the face of such failures. This will need to happen at multiple levels of abstraction. Resilience is also important for handling rapid changes in demand without overloading the platforms the services are running on. Resilience is also key to handling cyber attacks. One approach to counter this is defence in depth with successive security zones and automatic trip wires for detecting intrusion and raising the alarm.   Continuous monitoring can be combined with machine learning techniques for spotting unusual signs of behaviour.

Raw data often has limited value and only becomes valuable when it has been processed through multiple levels of interpretation that combines multiple sources of information, and provides results that are pertinent to a given context. This is where we can learn from nature by examining and mimicking the progressive processes involved in animal perception and recognition. The same is true for actuation, where high level intents can be progressively transformed into lower level control over different subsystems. What is needed to coordinate and synchronise distributed systems? As human beings, when we speak, our brains are able to coordinate the movements of many components each of which have widely varying response times. The jaw bone is massive and needs to be set in motion well before the tip of our tongue, which can move much more quickly.

The Web places a strong emphasis on scripting, and the same will apply for the Web of Things. Scripts could be used to define service logic for scalable cloud based platforms, for small scale platforms, e.g. as a new breed of home hubs, and for device gateways that bridge the IoT and the Web. JavaScript and Node.JS are being used to explore the potential. However, further work is needed to determine what APIs are needed to support a broad range of use cases. For instance, the scalable cloud-based COMPOSE platform addresses use cases involving event streams. New work is needed to support continuously changing properties as opposed to discrete events, and to address the rich requirements for complex cyber-physical systems. These are likely to involve different requirements at different levels of abstraction, e.g. tight requirements on timing at a low level, and perhaps transactional robustness at a high level.

To enable open markets of services, we need a standard way to access the service descriptions, so that a search engine can cover services hosted in different clouds operated by different vendors. We then need a standard framework for representing descriptions along with standard vocabularies/ontologies. This needs to cover the purpose of a service, the interfaces it exposes or depends upon, its security and privacy related properties, and so forth. Interoperability depends upon having compatible semantics and data representations. What is needed to motivate the re-use of vocabularies? When existing vocabularies aren’t a good fit to particular needs, what is needed to encourage the registration of a new vocabulary along with the assumptions that motivate it?

Where services have mismatching semantics or data formats, there is an opportunity for intermediaries to bridge the gaps. Search services can formulate plans for assembling services to fulfil the designated requirements. Such assemblies can be static or dynamic, e.g. all cars in this section of a road. Plans can also be applied to managing tasks, e.g. sequential tasks, concurrent tasks, and hierarchical arrangements of tasks, where tasks are associated with preconditions and postconditions. For the Web of Things, this corresponds to finding services that can perform these tasks, e.g. transforming data, identifying events that are implied by a combination of lower level events, or carrying out actions with a combination of actuators.

The “things” in the Web of Things are virtual objects. They can represent real world objects such as sensors and actuators, people and locations, or even abstract ideas like periods of time (e.g. the 70’s) or various kinds of events (e.g. a football match, concert or play). The “web” in the Web of Things refers to the idea that “things” are accessible via Web technologies, e.g. HTTP at the protocol layer, or scripting APIs at the services layer. Where “things” represent real-world objects such as people, the things can be considered as avatars that know about the person they represent, and can perform actions in fulfilment of the goals of that person. This is related to the concept of the Social Web of Things in which things have relationships to you, to your “friends” and to other things. The social network can provide a basis for routing notifications and for sharing services.

Google’s “Physical Web” (Google, 2014) is about beacons that broadcast web addresses to devices in their neighbourhood. This can be compared to walking down a crowded market street with all the store holders shouting out their wares and special offers. This calls for personal agents or avatars that are aware of your current interests and are able to recognise which beacons are relevant and which can be safely ignored. The agent could notify you directly or could perform some tasks on your behalf.

Avatars are also related to the concept of personal zones as explored in the EU FP7 webinos project. Your personal zone is an abstraction that groups all of your personal devices and services. It provides an overlay model for secure messaging between your devices as well as offering zone wide services to trusted applications running on your devices. Your personal zone also acts as an avatar on your behalf offering services to your friends based upon the access control policies you set. For the Web of Things, personal zones offers a unified means for people to manage the data they own.

Another consideration is the lifecycle of services, their provisioning, and packaging as products for sale to consumers. As an example, consider a security camera purchased for use at home. The camera may be bundled with the software and services, or this could be purchased separately from a Web of Things marketplace. Either way, the user needs a really simple approach to installing the hardware and setting up the associated services. How is the camera “discovered” and “paired” with a service? How does the user provide additional metadata, e.g. giving it a name, describing its location, and setting its access control policy? The package could include an application that the user runs to set this up, and to install any required device drivers. The package could include other applications that enable the user to manage the device and service, and as a user interface for the service when it is up and running. In the context of a smart city, there may be a need to install and set up large numbers of sensors. This too should be as simple and painless as possible. The same should be true for managing software upgrades and for taking devices and services out of service as needed.

Suppliers and consumers of services need to reach agreements, and this can be formalised as contracts that cover payments, data handling policies and so forth. For open markets of services such contracts should be legally binding on the parties involved. Whilst data may be provided free, in other cases, some form of payment will be required, for instance, one off payments, per usage payments and subscription based payments. To enable open markets to operate across borders, there is a need for international standards around payments. Even if services are provided free of charge, they may be restricted to consumers in certain groups. Access control may be based on rules, e.g. as with the XACML XML access control language, or based upon the possession of tokens as with capability based systems.

Access control is related to identity management. Mutual authentication is needed to ensure that both suppliers and consumers can be sure of the other party’s identity. Identity verification is about linking identities to real world properties, e.g. the physical location of a sensor, or the full name and postal address of a human. There is a need for trusted agents that provide identity verification services. Trust is also important to decisions about whether to use services: are they safe, do they come from a bone fide source, will they safeguard my privacy and so forth. This provides opportunities for agents that perform security and privacy audits of services. This can be complemented by crowd sourced reputations and reviews. Recommendation systems can further provide suggestions based upon what other people have looked at in similar contexts.

To realise the potential, W3C has launched the Web of Things Interest Group. This will survey use cases across application domains, conduct a technology landscape and identity requirements for web technology standards that W3C can then follow through in Working Groups.

References

W3C Web of Things Interest Group:
http://www.w3.org/WoT/IG/

Cisco, 2011: The Internet of Things — How the Next Evolution of the Internet Is Changing Everything,
http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

IERC 2104: European Research Cluster on the Internet of Things, 2014: Internet of Things From Research and Innovation to Market Deployment,
http://www.internet-of-things-research.eu/pdf/IoT-From%20Research%20and%20Innovation%20to%20Market%20Deployment_IERC_Cluster_eBook_978-87-93102-95-8_P.pdf

IoT2014: David Clark — Not making the same mistakes again,
http://www.iot-conference.org/iot2014/keynote-speakers/

Google, 2014: The Physical Web
https://google.github.io/physical-web/

COMPOSE: EU FP7 project 2012-2015,
http://www.compose-project.eu

Webinos: EU FP7 project 2010-2013,
http://webinos.org

6LoWPAN: IPv6 over low power wireless personal area networks,
Wikipedia article,
IETF 6LoWPAN RFCs

CoAP: IETF Constrained application protocol,
Wikipedia article,
RFC 7252,
RFC 7228,
coap.technology

MQTT: OASIS Message Queuing Telemetry Transport,
Wikipedia article,
MQTT Committee,
MQTT.org,