The Web of Things

Dave Raggett
The situation today ...

• Lots of research projects with similar but different approaches
  – IoT focus on devices and comms technologies
    • Rapid evolution and many standards
    • But not much work on how to create services
    • Inconsistent architectures and terminologies
  – Risk of fragmentation frustrating the potential

• Agreement of importance of HTTP
  – The Web of Things as a beacon of hope
The Web of Things

● “Using web technologies to harness the potential of the IoT together with the Semantic Web to enable an open ecosystem of services”

● Standards are needed to realize the economic and human potential, and to avoid the risk of fragmentation caused by a plethora of non-interoperable proprietary solutions.

● Key benefits for doing this work at W3C include the W3C emphasis on ensuring that W3C standards can be implemented royalty free, thereby encouraging innovation, and the availability of the large community of web developers.
Many use cases, many challenges!

- Smart cities – transport, power, water, governance, …
- Smart Homes – home automation, warranties, security, entertainment, …
- Home healthcare – improved outcomes and lower costs
- Retail and Richer Shopping experiences
- Smart factories & distribution

- Scalability
- Security
- Interoperability
- Coping with continuing change
  - Comms technologies and business models
- Privacy, personal control and threats to individual freedoms
  - Avoiding dystopia!
The Web as an Abstraction layer

- HTTP server as gateway to sensors and actuators
  - The IoT devices in many cases won't be using HTTP due to the need to conserve battery life …
    - WiFi, Weightless, ZigBee, CoAP & 6LoWPAN, NFC, BlueTooth & BLE, ANT, Infrared, USB, IEEE 1394, DASH7, KNX for buildings, EnOcean, GPRS/3G/4G, WiMAX, …
    - Server can host local services e.g. for use in homes
      - ABB, Bosch, Cisco & LG: Open Standard for Smart Homes
- HTTP2, REST and authentication
- JavaScript API for objects as proxies for services
  - Simplifying services by hiding details of communication
- How to declare services?
A Web of Value-Added Services

- Scripts & Metadata
  - Executes in sandbox on distributed platform

Data (JSON)
- Feeds from other services
- Feeds to other services
A Web of Services coupled to the IoT
Personal Zones

My devices, apps and services, including my cloud-based services

Generalization of concept from EU Webinos project
Personal Zones

My devices, apps and services, including my cloud-based services

Mary's Personal Zone

Inter zone peer to peer social apps and services

Internet

John's Personal Zone
Personal Zones

My devices, apps and services, including my cloud-based services

Mary's Personal Zone

Mary's zone has a TV, phone and tablet, a home gateway server and IoT devices, plus a couple of cloud based personal services

John's Personal Zone

Inter zone peer to peer social apps and services
A Market of Services

- Some people design and sell services
  - Analogous to selling native apps with Apple and Google
  - Example: John sells a service for home security webcams
- Others purchase services and apply them to their personal or business needs
  - Example: Mary purchases the service and applies it to her webcam in her garden in north London
    - As a service running in her home*,
    - As a service running in the cloud**
  - Mary has to bind the service to her webcam
    - Involves some kind of discovery/peering mechanism
  - Mary uses an app to access the service
    - This could be sold packaged with the service

* installed on her home gateway
** installed as part of her personal zone
API vs Data Feed

• Mary's webcam supports
  – Taking video clips triggered by motion detector
  – Live streaming of video when needed
  – Remote pan/zoom control and other settings
• Video clips uploaded to cloud for storage
• Mary can use app to view history of clips
• Mary can request direct live video feed and take dynamical control of the webcam
• Webcam as sensor with an API
  – Not just a simple data feed!
Functional Range of Services?

- Computationally trivial services
  - changing temperature units: Celsius to Fahrenheit
- Aggregating lots of data sources (Data Fusion)
  - traffic density maps from locations of thousands of cars
- Processing of media streams
  - Speech recognition from audio stream
- Complex image processing
  - Face recognition and tracking people via street or store cams
- Massively parallel computation
  - Big Data services involving Big Table and Map Reduce (Hadoop)
- Biological metaphors for pipelined perception and actuation
  - Extracting meaning from progressive interpretation of ambiguous or noisy data
  - Mapping high level intent to coordinated & synchronized control of actuators
  - Intelligent agents acting on their user's behalf
What standards are needed?

- Metadata standards
  - Basic metadata (name, developer, owner, description, ...)
  - Developer and platform fees (one off, subscription, per-use)
  - Access control policies
    - Sticky policies limiting who can use data provided by the service and for what purposes
    - Provenance
  - Service as software interface
- Identity and authentication
  - Mutual authentication based upon public key cryptography
- Payment protocols
- Platform APIs for use by scripts
  - Apps, Services, Developer tools, Search, ...
Service as API

- Scripts declare their sources and sinks
  - **Source**: an interface the service imports from another service
  - **Sink**: an interface this service exports to other services
  - Scripts are event driven and most platform APIs are asynchronous
    - APIs based upon `promises` `method().then().catch()`
      - `method()` returns a promise object, `then()` sets handler to be called when promise is fulfilled

- Platform does the heavy lifting
  - Directly analogous to package management on Linux
  - Decouples service scripts from details of how data is transferred and where service is executed
  - Platform manages distributed storage of data and eventing
  - Platform enforces security and access control policies
  - Allows for services to distributed across servers for a scalable platform
A Federation of Clouds

Standards are needed to connect services in different clouds
Some potentially relevant standards

- HTTP2 and Web Sockets
- OAuth and work on identity and authentication
  - Second factors, secure elements, ...
- HTML5 JavaScript APIs for browsers
  - Reuse existing standards where practical, e.g. web crypto
- JSON, JSON-LD, JSON-Pointer, JSON Schema, JSON-RPC, WebIDL, JSON.IQ, ...
- SPARQL, OWL, RIF and Semantic Web stack
  - Linked Data Platform – REST for R/W linked data
- XACML for XML based access control rules
  - Separation of decision and control points
Web of Things Community Group

http://www.w3.org/community/wot/

• Mission: to accelerate the adoption of Web technologies as a basis for enabling services for the Internet of Things
• Anyone can join, no fees
• Currently just a wiki and 64 participants
• Plan to actively drive work from January 2014
  – Collect use cases/stories, requirements, implementation experience, identify missing standards, help with standards workshop, ...
W3C Web of Things Workshop

- W3C workshops provide a venue for discussions on timeliness and scope for potential work on standards
- Plan for mid-2014, somewhere in Europe
- Position papers, presentations, panels, breakouts, and perhaps a hackathon?
- Leading to chartering new working groups?

* W3C is also planning a web payments workshop in March 2014