

The Web of Things

Breakout session on service descriptions

Dave Raggett, W3C

The Web of Things

A huge variety of potential application domains including ...



Perils of Fragmentation

- Advances in electronics enabling low cost sensors and actuators
- Many potential application domains
- But confusion arising from the plethora of communication technologies
- And lack of a shared approach to services
- This fragmentation is holding back the true potential
- We need to work together to define open standards and marketplaces!

The Web of Things

- This workshop is an opportunity to discuss
 - Combining the Web of Devices with the Web of Data to create the Web of Things
 - Service platforms at the network edge (e.g. home hubs) or in the cloud
 - The role of open standards as a basis for open markets of services
 - What common requirements emerge from the various application domains?
 - What kinds of standards should W3C focus on?

Why Focus on Services?

- ***Services, Not Sensors:*** Gartner expects Internet of Things vendors to top \$309 billion in direct revenue by 2020, with most of that money deriving from services
- ***The trick will be whether hardware companies will push hard enough for standardization so they can capitalize on services revenue. Companies that see themselves as pure hardware manufacturers are likely doomed, but those that see beyond the "things" to instead focus on the services built on the "Internet," the future is very bright.*** Matt Asay, MongoDB
- ***Eventually, something like HTML, the language of the web, will be required to make the internet of things realize its potential. "Interoperability is critical,"*** says Mike Bell, head of wearables at Intel.
- VisionMobile: IoT: Breaking Free From Internet And Things
 - ***Platforms are valuable because of their developer communities***
 - More developers leads to more innovative services, boosting the value of the platform to end users
 - Attracting developers by making it easy for users to find apps and services, and for developers to monetize their work

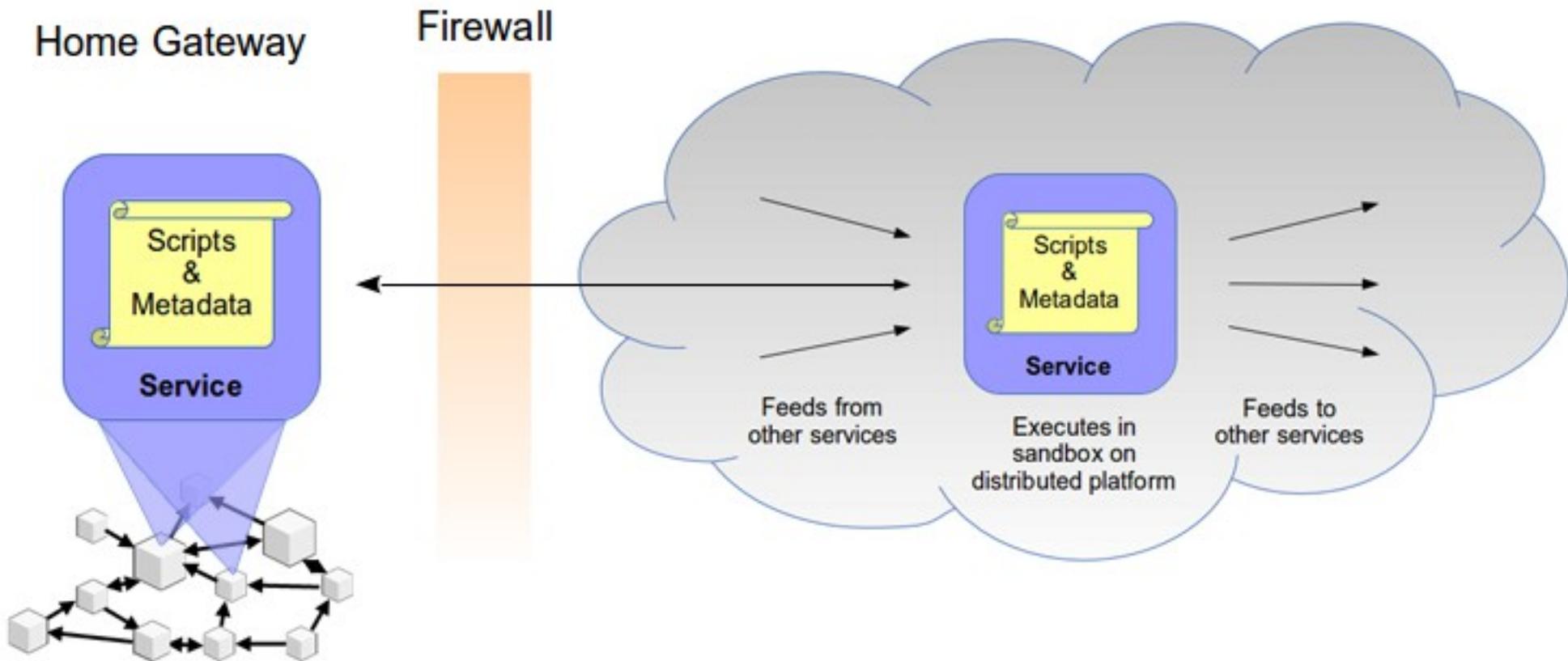
Basic Requirements

- Services vary according to the application domain and use case, e.g. latency requirements
 - Water levels in reservoirs
 - Current traffic conditions
 - Free parking slots
 - Remote control of robots and UAV's
- The number of sensors and amount of data
- Whether it is okay to drop data
- Graceful handling of faults

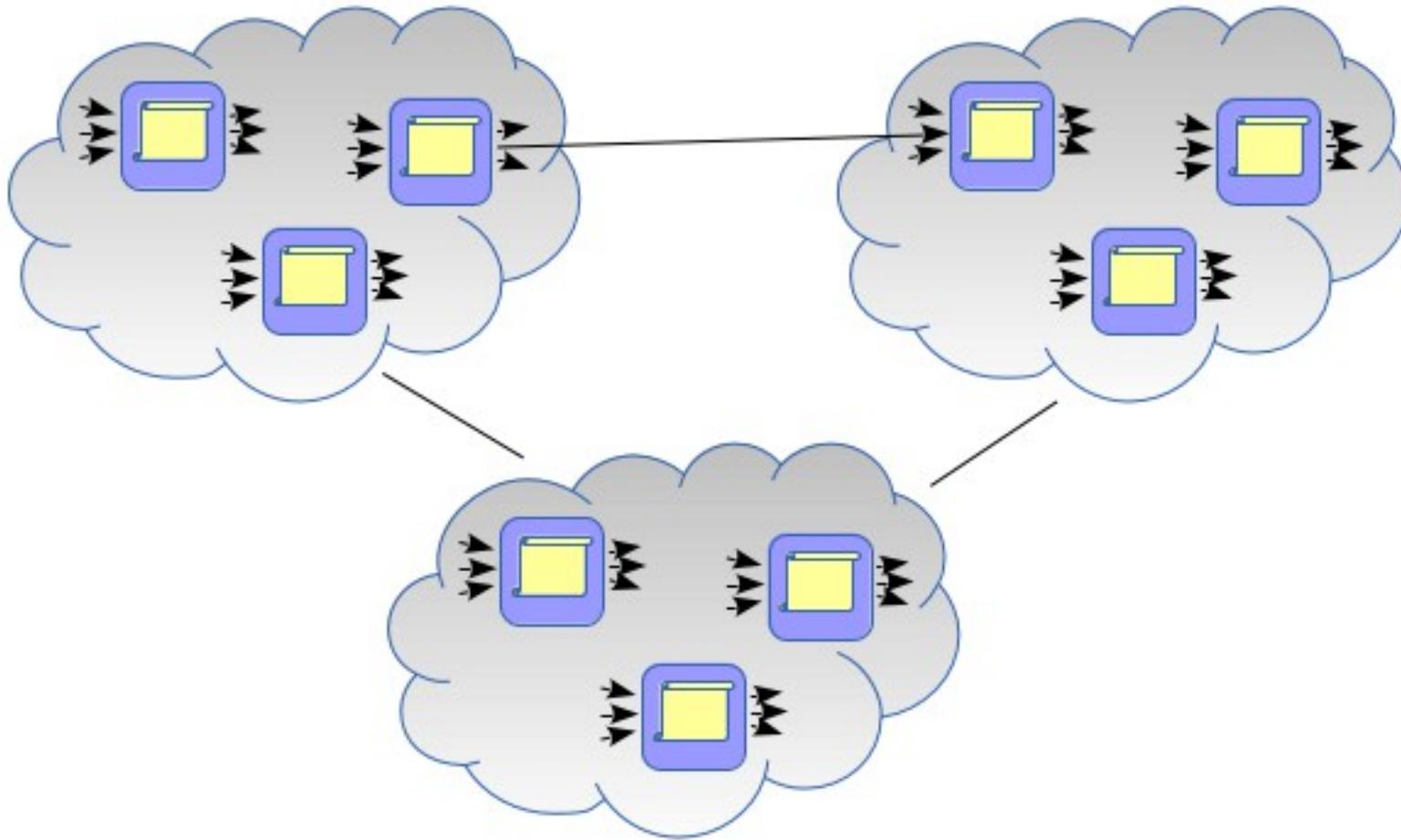
JavaScript based Services

- JavaScript is widely used for web pages
- Also available for services, e.g. node.js
- And on wide variety of devices, even microcontrollers!
- Platforms that allow developers to use JavaScript, JSON, and related technologies would enable a huge number of web developers to create services
- Shared open standards are essential to maximising the potential

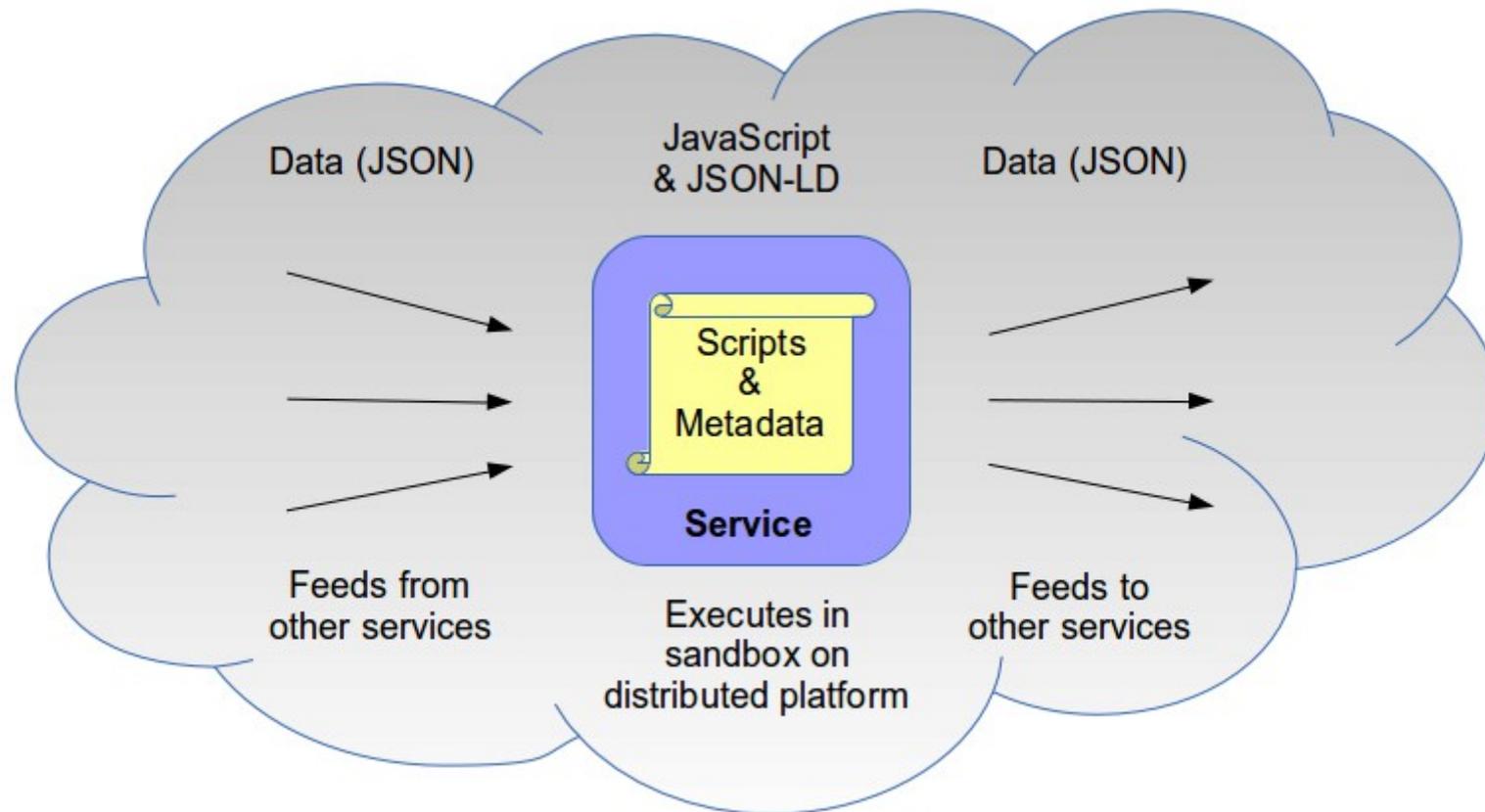
Services at the network edge



Federation of Service Clouds



JSON & JSON-LD



JavaScript Object Notation

```
{  
  "type": "depth-gauge",  
  "site": "Upton-Deveril",  
  "depth": "9.3m",  
  "date": "2014-06-14T06:30"  
}
```

- Hierarchical text-based data format with strings, numbers, arrays and named fields.
- Easy to parse or stringify

JSON-LD for descriptions

```
{  
  "@context": "http://json-ld.org/contexts/person.jsonld",  
  "@id": "http://dbpedia.org/resource/John_Lennon",  
  "name": "John Lennon",  
  "born": "1940-10-09",  
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"  
}
```

- Machine interpretable descriptions
- Linked data format expressed in JSON
 - <http://www.w3.org/TR/json-ld/>
- Relevant to Web of Things service metadata
- Relation to Linked Data Platform
 - <http://www.w3.org/TR/ldp-primer/>

What do we need to describe?

- What APIs a given service depends upon
- What APIs a given service exports
- Who owns the service
- Who developed the service
- Human and machine interpretable descriptions
- Reputation, reviews and audits
- What payments may be required
- Access control policies
 - Relationship to identity, authentication, privacy and provenance

Terminology

- We need to work on a common terminology to ensure we are talking about the same things
- Is a service bound to specific sensors, or is it a class, prototype or template that you purchase and then bind to your sensors as a service instance?
- This could be handled via a Primer published as a W3C NOTE

APIs vs Protocols

- IoT commonly involves detailed discussions of the protocols used to access devices
- Web developers would prefer simple APIs and leave the protocols to the platform to handle
- Mapping from APIs to RESTful HTTP
- APIs for pub-sub protocols like MQTT
- Device drivers to hide communication with devices
 - Generally used at network edge
 - How to identify which driver is needed?
 - How much trust to give to device descriptions?

Analogy to Linux Packages

- Redhat .rpm and Debian .deb packages
 - Packaged apps and libraries
 - Include metadata on versions and dependencies
 - Allows package manager to ensure that dependencies are met when installing software
 - Knowing what can be safely deleted when uninstalling packages
 - A big advantage over platforms without package management standards (e.g. Windows and OSX)
- We need something analogous that works across clouds from different vendors

Value-added Services

- Services that combine other services in some useful way
- Services that combine services with rich descriptions and other sources of information
- Services that transform service data into the same, higher or lower level of abstraction

Abstraction Layers

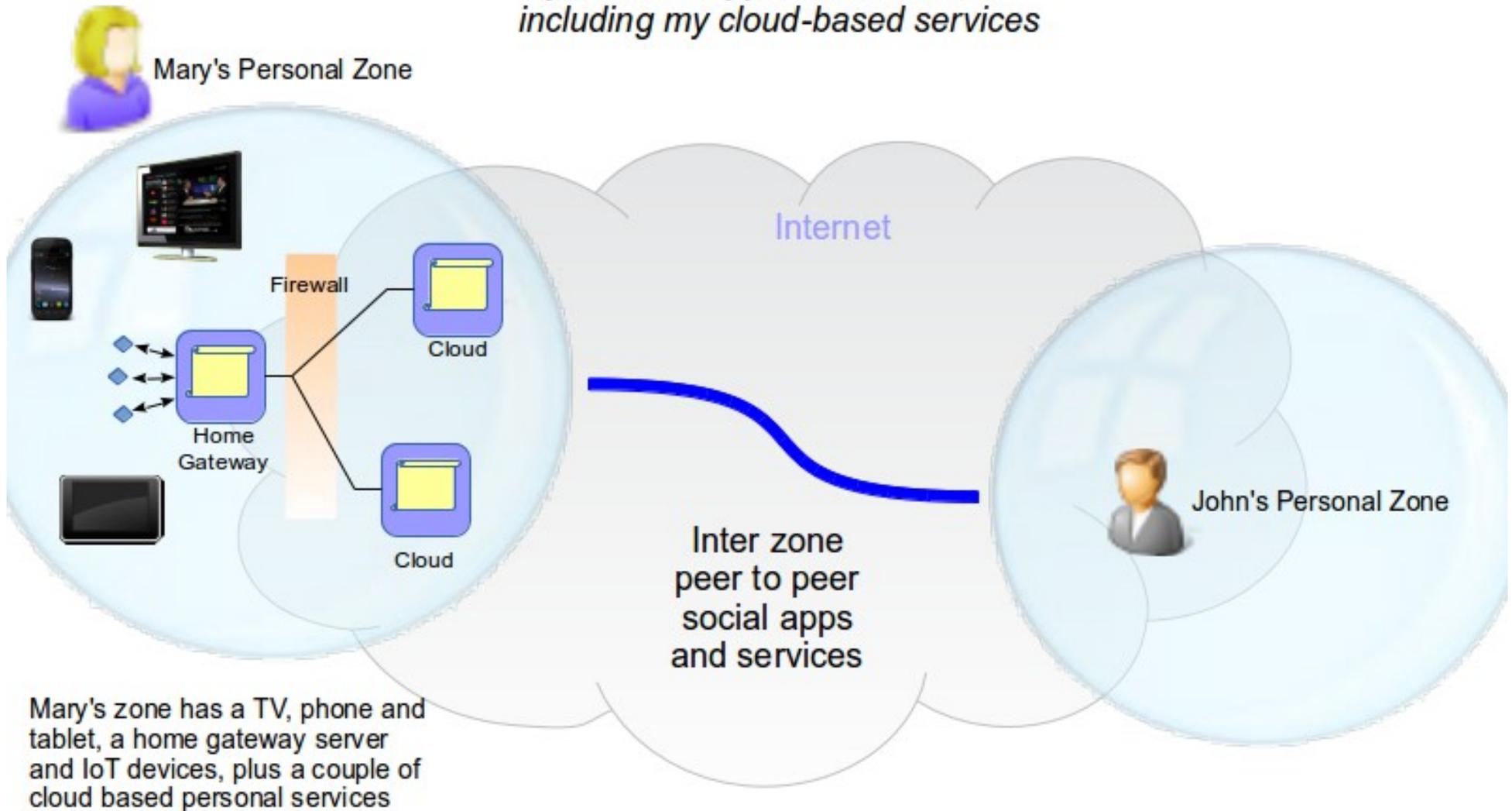
- Perception – progressive interpretation of sensor data in combination with contextual data
 - Transforming from low level data to high level descriptions
 - Addressing noisy data and faults
 - Identifying events that are implicit in the data
- Actuation – mapping high level intent into progressively lower level commands
 - Synchronized control of multiple actuators
 - Adapting to faults

Socially Aware Services

- Services that involve models of devices, people and their inter-relationships
 - Structured data about sensors, locations, people, events and processes
 - Relevance to sharing and access control policies
- Webinos project and Personal Spaces
 - Your personal devices, apps and services
 - A basis for exposing services to people you know

Personal Zones

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



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

APIs and IDLs

- WebIDL as example of a syntax for describing APIs
 - Commonly used for W3C specifications
 - Could inspire a WoT IDL syntax
- For interoperable services we need
 - Machine interpretable representations of data types
 - e.g. for temperatures expressed in Celcius
 - Access control policies
 - Integrity constraints to simplify error handling

An example

- Remotely controlled video camera
- Pan, tilt and zoom
- Motion sensing
- Video as RTSP stream
- Infrared lights for night time
- Bidirectional audio



Functional Descriptions

- Describes the behavior of a service
 - e.g. a camera with various features
 - Not a complete specification
- Used to support search for services
- Human readable descriptions in one or more languages
- Machine interpretable semantic descriptions
- Challenge of keeping these consistent and accurate

Simplified API for camera

```
enum Quality { "high", "medium", "low" };  
  
interface WebCam {  
  attribute float azimuth;  
  attribute float elevation;  
  attribute Quality resolution;  
  readonly attribute DOMString videoURL;  
  callback motionDetectedCallback = void ();  
};
```

- Using WebIDL like syntax
- Further details needed to express
 - Limits, e.g. on azimuth and elevation
 - RTSP video format, and relation to quality settings

Mapping to RESTful HTTP

POST settings HTTP/1.1

Host: example.com

User-Agent: Mozilla/5.0 (X11; Linux x86_64)

Keep-Alive: 300

Connection: keep-alive

Content-Type: application/json

Content-Length: 49

```
{"azimuth":180,"elevation":0,"resolution":"high"}
```

- Data payload as JSON

Simplified Service Description

```
{  
  "Developer" : "Shenzhen EasyN Technology Company",  
  "Description" : "description.html",  
  "Version" : "1.0.0",  
  "Owner" : "Mary Smith",  
  "API" : "webcam.idl",  
  "Behavior" : "behavior.js"  
}
```

- Functional description in description.html
- Service logic implemented in behavior.js

Additional Discussion Topics

- Security
 - Identities and authentication for people, devices and services
 - Access control policies
 - Updates
 - Proactive and retroactive security
- Payments and service monetization
- What platform APIs are needed to support developers and users?
- Services used for intent based search