An introduction to the

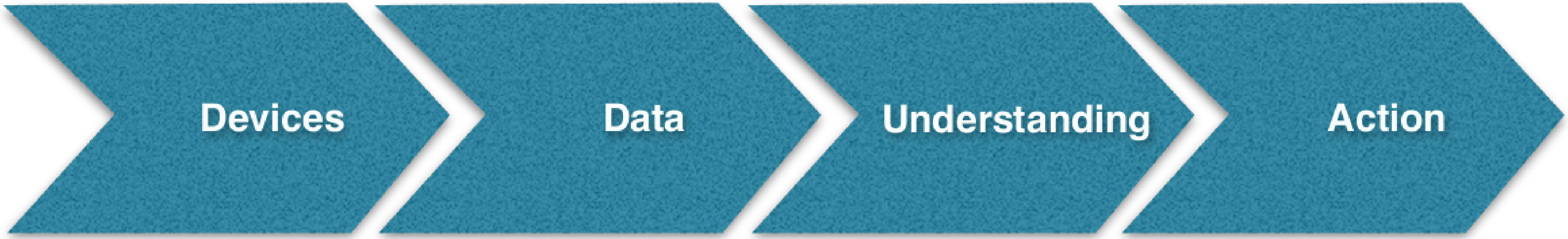
Web of Things Framework

May 2015

Dave Raggett, W3C

This work is supported by the European Union's 7th Research Framework Programme (FP7/ 2013-2015) under grant agreement nº317862 - Compose

The Promise of the IoT



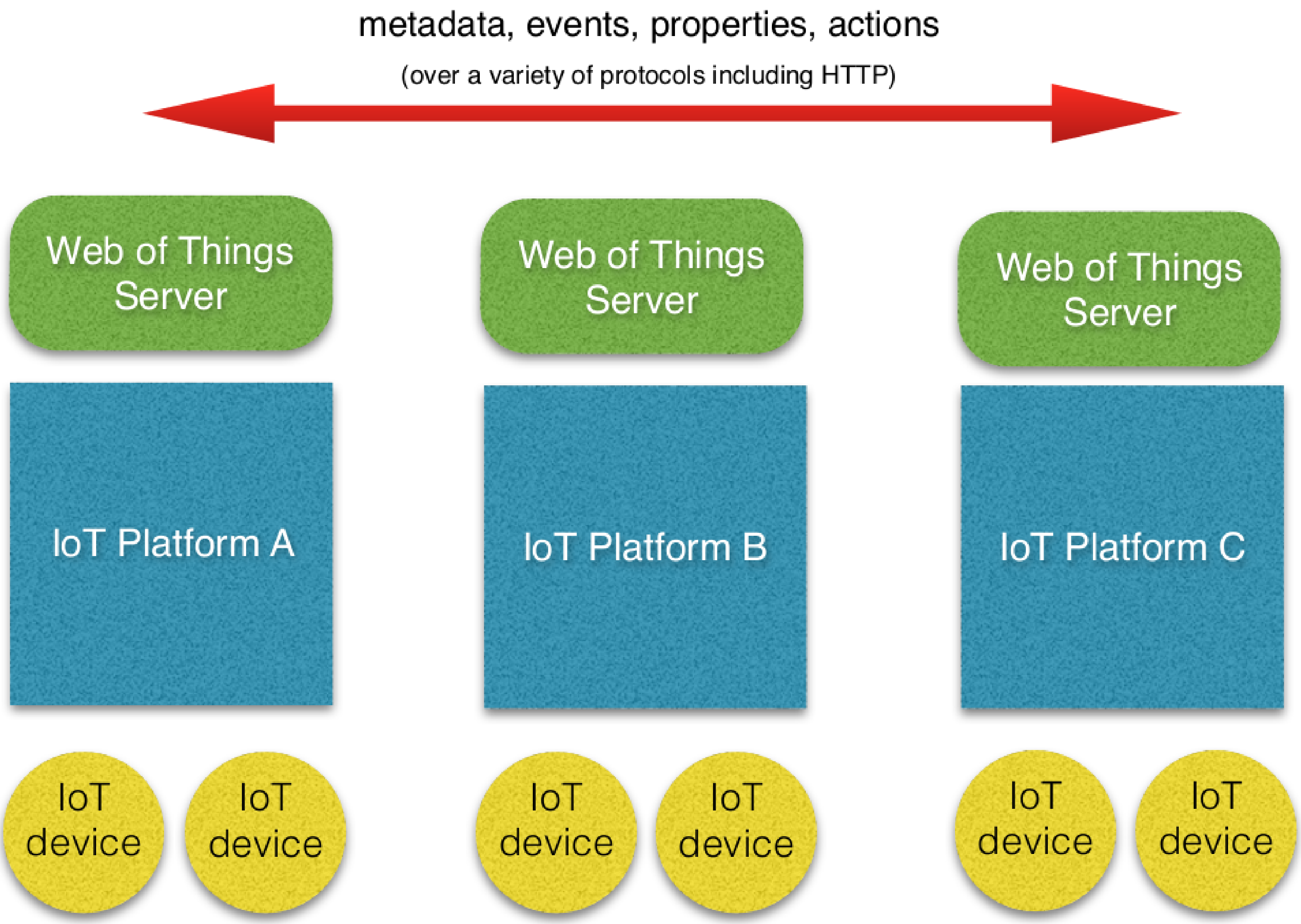
With thanks to Vision Mobile

The Challenge

- **We expect one hundred billion IoT devices to be deployed within ten years**
- **But, the Internet of Things is currently beset with problems**
  - Product **silos** that **don't interoperate** with each other
  - Plethora of approaches & **incompatible platforms**
  - **This is blocking the benefits of the network effect**
- **This is painful for developers**
  - Hard to keep track of who is doing what
  - Expensive to learn and port to different platforms
  - Challenging to create services that span domains and platforms
- **Platform developers seeking to unlock the commercial potential**
  - To reduce development costs for IoT applications and services
  - To fulfil customer demand for services requiring integration with other platforms
  - To grow the size of the overall markets
    - A small share of a huge market is better than a big share of a small market

The Web as the Solution

“Things” as virtual objects acting as proxies for physical and abstract entities



metadata, events, properties, actions

(over a variety of protocols including HTTP)

Web of Things Server

Web of Things Server

Web of Things Server

IoT Platform A

IoT Platform B

IoT Platform C

IoT device

IoT device

IoT device

IoT device

IoT device

IoT device

Many Business Opportunities

- Homes & Offices
  - Security, Energy, Entertainment, Pets, Lighting, Heating (HVAC), White Goods, ...
- Life and Healthcare
  - Fitness, monitors, medication
- Cities
  - Transportation, Utilities, Planning, Security
    - Cars as an integral spart of the Web of Things
- Energy
  - Reducing peak demand through smart energy appliances
  - Dealing with local generation of power, and rise of electric cars
- Retail & Catering
  - Logistics, Beacons, Richer information on products, Loyalty schemes
- Industry
  - Logistics, Design, Manufacturing
    - Reduction of 1% of operational costs can save billions
    - Speeding design to delivery of bespoke products
- Environment
  - Droughts, Floods, Fires, Emergency management
- Big Data
  - Creating value by analysing combinations of multiple sources of data

From Pages to Things

- The web of pages is founded upon
  - IRIs for addressing
  - HTTP for access
  - HTML for pages and for discovery
    - Search engines following the links in pages
- Web of Things by analogy with web of pages
  - IRIs for addressing
  - HTTP and other protocols for access
    - No one protocol can satisfy all needs
  - Thing Description Language (TDL)
    - Semantics and data formats as basis for interoperability
    - Relationships to other things as basis for discovery

** IRI is an abbreviation for Internationalised Resource Identifiers (aka URIs or web addresses)

Web of Things Framework

- Expose IoT platforms and devices through the World Wide Web for a Web of Things
  - Device abstraction layer to bridge IoT to the Web
- “Things” as proxies for physical and abstract entities
- Modelled in terms of events, properties and actions
  - What events does this thing generate?
    - *Someone has just rung the door bell*
    - *Someone has just inserted a door key*
  - What properties does this thing have?
    - *Door is open or closed*
  - What actions can we invoke on this thing?
    - *Unlock the door*
  - Thing with on/off property as proxy for a light switch
- With bindings to scripting APIs and protocols
  - Service logic decoupled from underlying communication details

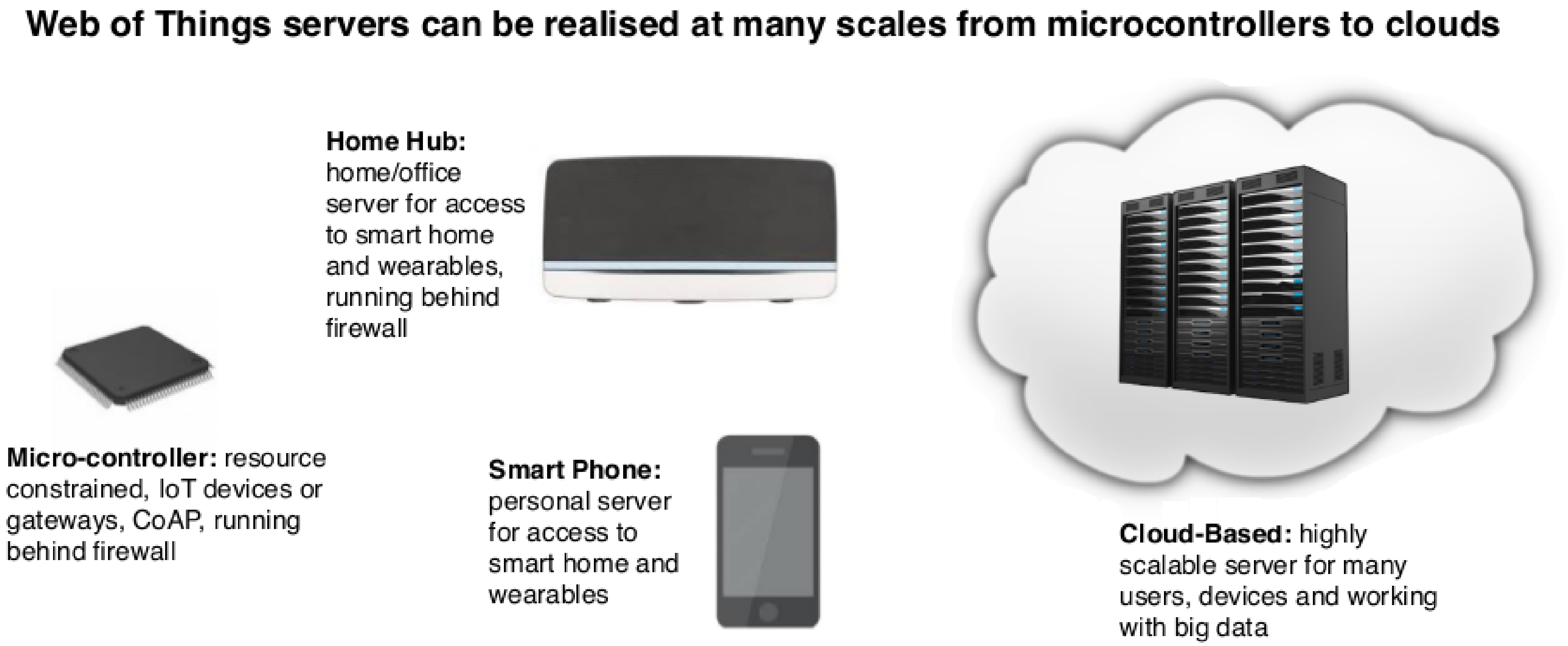
Web of Things Framework

- Standard way to retrieve “thing” descriptions
- Standard format for “thing” descriptions (e.g. JSON-LD)
- Owner, purpose, version, access control, terms & conditions, relationships to other things, security best practices, . . .
  - Giving data owners control over who can access their data and for what purposes – contract between consumer & supplier
- Semantics and data formats for events, properties & actions
- Properties have discrete values, or smoothly changing values that are interpolated between data points, e.g. for robotics
  - Delegating control to where it makes the most sense
  - Clock sync across controllers: 1-10 mS with NTP, and microseconds with IEEE protocols
- Communication patterns
  - Push, pull, pub-sub, and peer to peer
- Bindings to a range of protocols
  - HTTP, Web Sockets, CoAP, MQTT, STOMP, XMPP, WebRTC

Interacting with a “Thing”

- Representational State Transfer (REST)
  - HTTP GET to retrieve a thing's description
  - HTTP GET to retrieve all properties of a thing
  - HTTP PUT to update all properties of a thing
  - HTTP PATCH to apply changes to some properties
  - HTTP POST to invoke actions on a thing
  - HTTP POST is also used to notify events
    - To proxies or dependent things
- REST can be used with other protocols
  - To send actions to thing within a firewall
  - To distribute updates via pub-sub model

Servers at many scales



Web of Things servers can be realised at many scales from microcontrollers to clouds

Home Hub: home/office server for access to smart home and wearables, running behind firewall

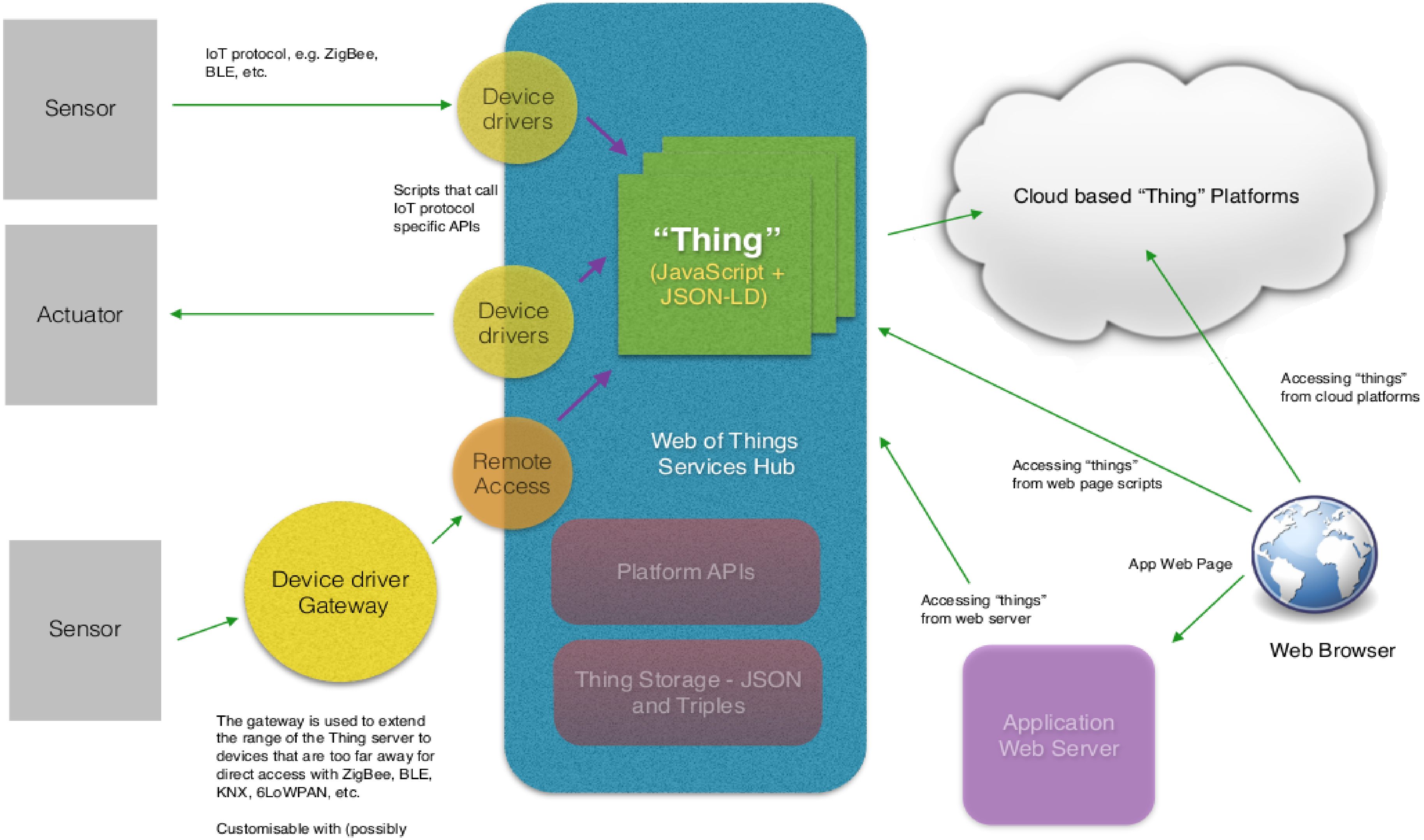
Micro-controller: resource constrained, IoT devices or gateways, CoAP, running behind firewall

Smart Phone: personal server for access to smart home and wearables

Cloud-Based: highly scalable server for many users, devices and working with big data

Servers are free to choose which scripting languages they support
Could precompile service behaviour for constrained devices

Example of a Home Hub



Sensor

IoT protocol, e.g. ZigBee, BLE, etc.

Device drivers

Scripts that call IoT protocol specific APIs

Actuator

Device drivers

“Thing” (JavaScript + JSON-LD)

Remote Access

Web of Things Services Hub

Platform APIs

Thing Storage - JSON and Triples

Sensor

Device driver Gateway

Cloud based “Thing” Platforms

Accessing “things” from cloud platforms

Accessing “things” from web page scripts

App Web Page

Accessing “things” from web server

Web Browser

Application Web Server

The gateway is used to extend the range of the Thing server to devices that are too far away for direct access with ZigBee, BLE, KNX, 6LoWPAN, etc.

Customisable with (possibly precompiled) scripts that call IoT protocol specific APIs

Relationships between Things

- “Thing” description includes the relationships to the things that this thing depends upon
    - Server uses this to retrieve descriptions of related things as basis for deciding how to connect to them and expose them to scripts that define this thing's behaviour
        - With support for circular dependencies
    - Enables search engines to index the web of things
    - Supports richer search queries based upon relationships
    - Enables dependency management
        - Perhaps analogous with Linux package management
    - Decouples service behaviour from data protocols
    - Simpler expression of service behaviour via local names for things
    - Choice between early and late bound types
        - Strong vs weak typing and implications for extensibility

End-User Service Creation

- Event-condition-action rules
  - Trigger action upon event if condition is true
  - High level events defined in terms of lower level events
  - Higher level actions defined in terms of lower level actions
    - Ordered and unordered sequences of actions
    - Pre- and Post-conditions
- Simple to use graphical editing tools
- Vocal commands (as with Apple's Siri)
  - *“turn the heating down when I leave home”*

Appeal of JSON-LD

- What makes JSON-LD attractive as basis for the thing description language?
- W3C Recommendation from 16 Jan 2014
  - http://www.w3.org/TR/json-ld/
- Combines simplicity of JSON with the power of the Linked Data and the Semantic Web
  - Out of band profiles and binary JSON formats for short packet protocols
- We would define a core profile for a vocabulary common to all “thing” descriptions
- Implementers would be encouraged to re-use vocabularies for specific application domains
  - These could be defined by industry specific groups
  - Need for better schema/vocabulary languages?
    - Dynamic data validation for greater robustness

Thing Descriptions

Server uses IRI for a thing to download its description

- Door

```
{
   “@events” : {
      “bell”: null,
      “key”: {
         “valid” : “boolean”
      }
   },
   “@properties” : {
      “is_open” : “boolean”
   },
   “@actions” : {
      “unlock” : null
  }
}
```

- Light switch

```
{
   “@properties” : {
      “on” : {
          “type” : “boolean”,
          “writable” : true
      }
   },
}
```

TDL's default JSON-LD context defines bindings of core vocabulary to IRIs
Data models may be defined explicitly or by reference to an external definition

Thing as Agent

- Thing description

```
{
   @context : {
       @base=”http://….
    },
    “@dependencies” : {
        “door” : “door12”,
        “light” : “switch12”
    }
}
```

- It's behaviour

```
// invoked when service starts

function start () {
    door.observe(“key”, unlock);
}

function unlock(key) {
    If (key.valid) {
        door.unlock();
        light.on = true;
    }
}
```

This “thing” is an agent that is bound to a specific door and light switch.

It unlocks the door and turns on the light when a valid key is presented.

n.b. @base defines a base IRI for resolving relative IRIs

Miscellany

- For validation and specification of vocabularies
  - JSON-Schema
  - RDF-Schema
  - OWL
- For efficient transfer of structured data
  - JSON *(defined by RFC7159, ECMA 404)*
    - MessagePack, Universal Binary JSON, etc.
  - Google's Protocol Buffers
  - XML with EXI
- Bindings to protocols need to cover encodings
  - **/.well-known/protocols** for retrieving server's protocol support?
- Actions on things are asynchronous and may return results

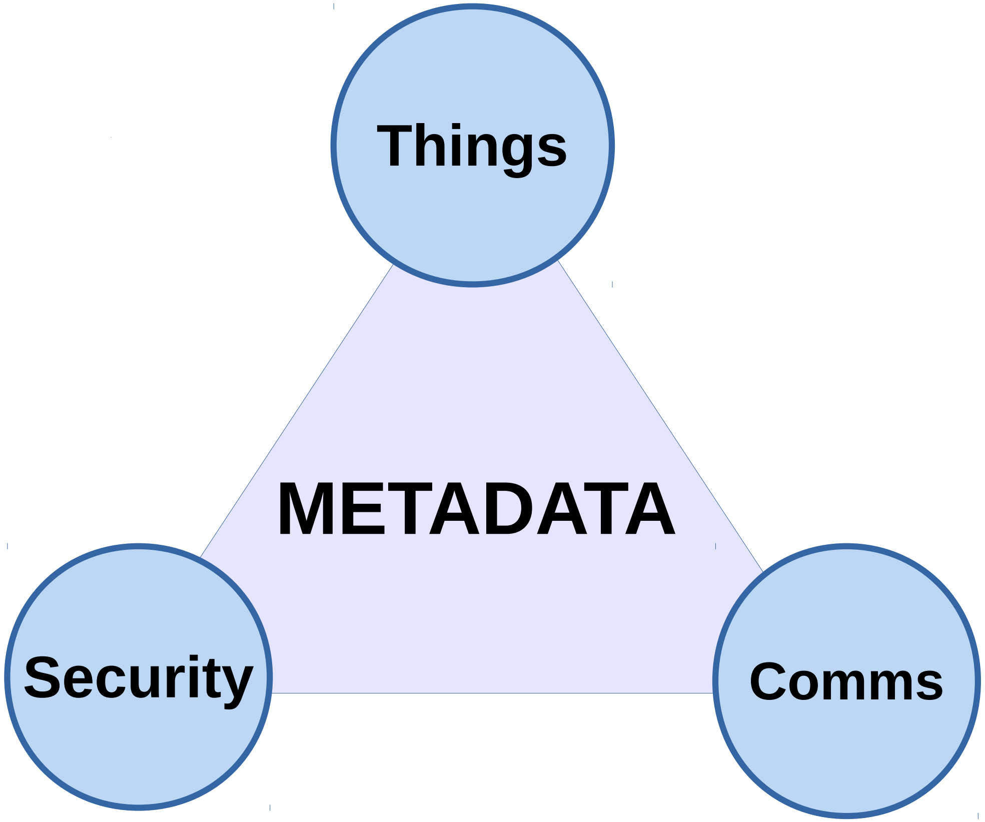
Thingsonomies

- The purpose of a “thing” can be defined formally in respect to an ontology
- The purpose can be defined informally using free text, e.g. as one or more tags chosen by the maintainer
- Co-occurrence of tags across many “things” performs an informal expression of semantics
  - In same way as folksonomies for images or blog posts
- Statistical treatment of natural language and cognitive models make this increasingly attractive, e.g.
  - Apple Siri
  - Google Now
  - IBM Watson

Thing Descriptions

- Thing descriptions may be static and shared by many “things”
  - These things can define their description by reference
    - Relationship to statically typed programming languages
- Some kinds of things may involve descriptions that change over time, e.g. a new owner, or a new physical location for a sensor
  - Events signalling changes to metadata?
  - Thing memories that record changes over a thing's lifetime
- Bindings to protocols may involve self tagged data
  - Analogous to “unions” in programming languages
  - Allows for extensibility – ignore new fields you don't know about
- The properties of a “thing” may include data blobs that have a meaning and a content-type
  - Photo of someone and encoded as image/jpeg

Focus of W3C Contribution



- **Thing descriptions**
  - Data models & relationships between things
  - Dependencies and version management
  - Discovery and provisioning
  - Bindings to APIs and protocols
- **Security related metadata**
  - Security practices
  - Mutual authentication
  - Access control
  - Terms & conditions
    - Relationship to “Liability”
  - Payments
  - Trust and Identity Verification
  - Privacy and Provenance
  - Resilience
- **Communication related metadata**
  - Protocols and ports
  - Data formats & encodings
  - Multiplexing and buffering of data
  - Efficient use of protocols
  - Devices which sleep most of the time

Semantics for Smart Appliances

- Semantic Sensor Network Ontology
  - W3C SSN Incubator Group report
  - SSN Ontology
- Sensor Model Language (SensorML)
  - Developed by Open Geospatial Consortium
- Sensor Markup Language
  - JSON & XML/EXI – IETF draft-jennings-core-senml
- TNO's smart appliance ontology for ETSI M2M
  - Developed on behalf of European Commission

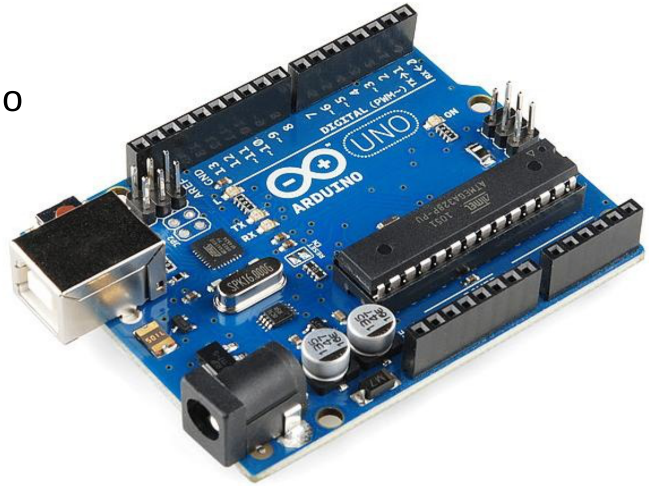
IETF CoRE WG

- CoRE WG with focus on resource oriented applications for constrained IP networks, and responsible for CoAP protocol
  - See tracker page and CoAP website
  - CoAP is based on REST and similar to HTTP
    - GET, PUT, POST, DELETE, OBSERVE methods
  - CoAP is a good fit for the Web of Things
- Resource discovery
  - Unicast or multicast queries
    - Link format (RFC6690) analogous to HTTP Link header
      - Which itself is modelled on HTML's LINK element
      - JSON link format under consideration
    - GET /.well-known/core returns list of resources
- Notifications with push and pub-sub
  - Interested parties register with GET
  - Notifications are sent with OBSERVE method

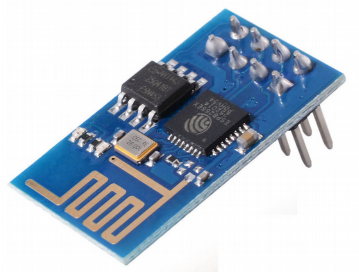
The Maker Community

- Open hardware and open source software are a huge opportunity for a bottom up approach to growing the Web of Things
  - Let's have lots of fun!

Arduino



ESP8266: 80MHz
32 bit MCU + WiFi
Available for 2 USD



CoAP: REST for IoT devices
MicroCoAP: 16 KB including the Ethernet library, for more see:
https://github.com/1248/microcoap

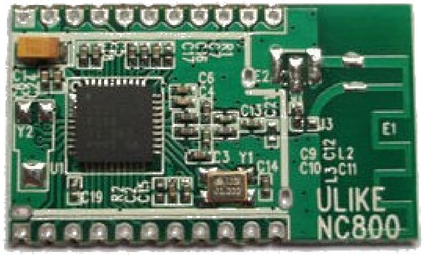
MQTT as a lightweight binary
Pub-sub protocol with brokers, see:
https://github.com/knolleary/pubsubclient

NodeJS based Web of Things server
with many libraries available for IoT
(run on Raspberry Pi as Home Hub)

Sensors

C & Arduino IDE
Lua & NodeMCU
MicroPython
RIOT OS

CC2530: 8051 MCU + IEEE 802.15.4
Up to 256 KB flash + 8 KB RAM
Available at 0.5 USD in quantity



Web of Things Interest Group

- Launched in January 2015
  - First face to face in Munich, April 20-22
- Focus on pre-standardisation work
  - Survey of use cases and requirements across app domains and business sectors
  - Architecture and technology landscape
  - Security, privacy and resilience
  - Discovery and provisioning
- Liaison with external IoT groups
    - See list on IG wiki page
- Preparation of proposed WG charter for launch late 2015
  - Standardise Web of Things Description Language
  - Standardise Bindings to APIs and Protocols
    - In collaboration with other groups
      - IETF protocols such as HTTP, WebSockets, CoAP, and XMPP
      - Security best practices

Join the Web of Things IG

http://www.w3.org/WoT/IG/

- Help us to prepare the ground for standardizing the Web of Things Framework by joining the Interest Group
- We've recently set up the following task forces
    - Use cases and requirements
        - We're looking for help with broadening this across a wide range of business sectors
        - You can help us to set up task forces for specific business sectors
            - How much money could your company benefit from the IoT?
    - Web of Things Framework
        - Thing descriptions
        - Discovery and provisioning
        - Bindings to APIs and protocols
    - Security and Privacy
- We plan to review a draft Working Group charter at our July F2F in North America

Summary

- The IoT is growing very fast but needs web scale standards to realise its potential
  - Web technologies to enable web scale markets and reduce the cost for developing IoT apps and services
- The Web of Things is based upon Linked Data to model virtual objects as proxies for physical and abstract entities
  - JSON-LD combines simplicity of JSON with the power of Linked Data
- A new class of web servers that communicate with each other at an abstract level in terms of metadata, events, properties and actions
- Bound to a variety of protocols to support common communication patterns
  - Push, pull, pub-sub and peer to peer
  - Protocols like HTTP, WebSockets, CoAP, MQTT, XMPP and WebRTC
- Implementable on a wide range of scales
  - Microcontrollers with CoAP
  - Smart phones and home/office hubs
  - Cloud-based server farms
- Supports early and late bound data types
  - Important for extensibility and resilience to changes
- W3C WoT IG on pre-standardisation activities, WG to follow late 2015
- Help us to enable the IoT to realise its full potential!