The Web of Things

as presented to the

Web Science Institute
University of Southampton

Dave Raggett <dsr@w3.org>

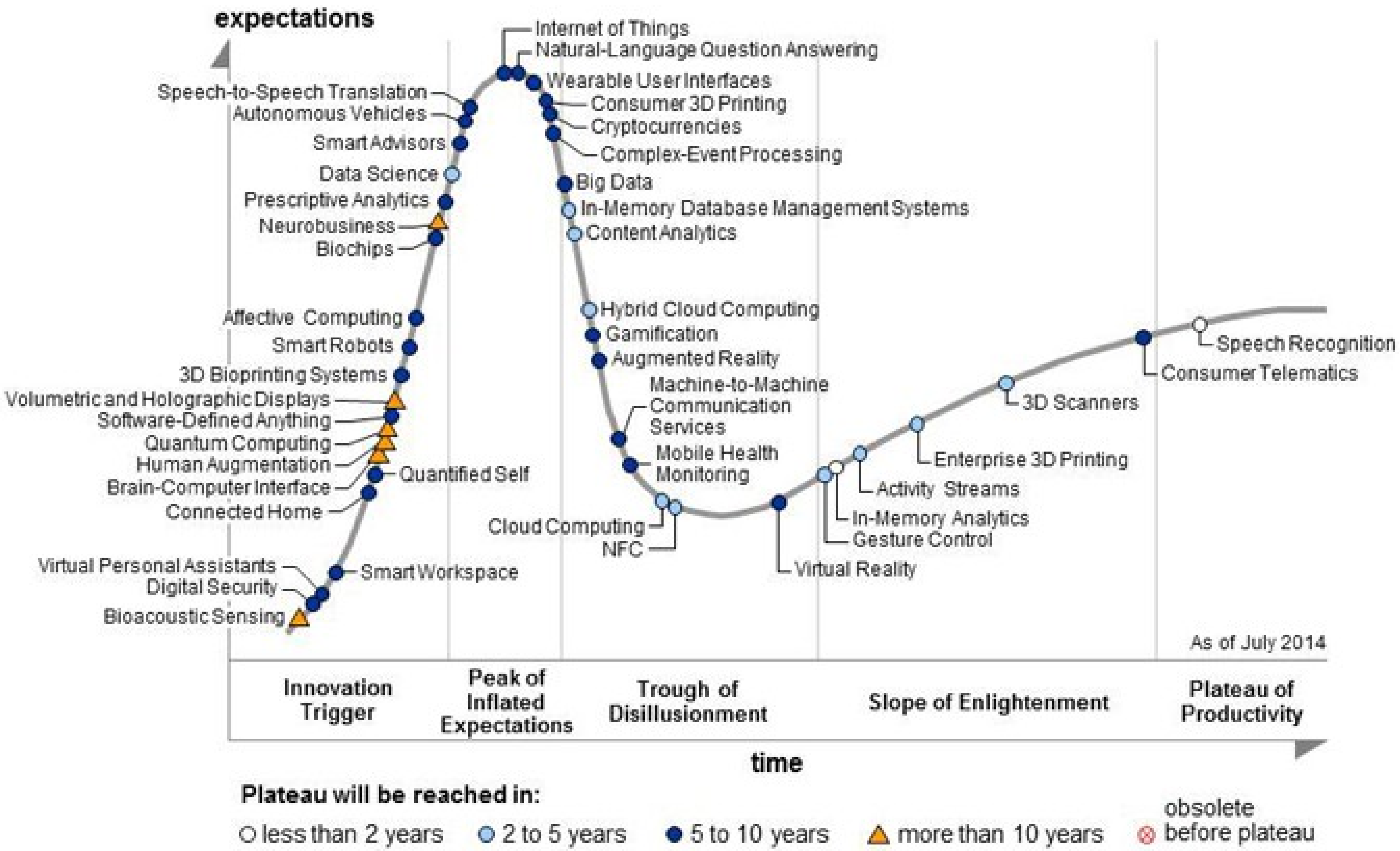
15th July 2015

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

The Internet of Things

- Services that connect into the physical world around us
- There are many application domains
  - Smart homes and buildings
  - Life and healthcare
  - Retail, beacons, and improved logistics
  - Transportation, Utilities, City planning
  - Smart grids, electric cars and local power generation
  - Smart industry and evolution of manufacturing
  - Environmental monitoring and handling of emergencies
- Legitimate concerns over security and privacy
  - Fears over abuse of big data and pervasive monitoring

IoT at the top of the hype cycle*



*From Gartner's hype cycle for emerging technologies – August 2014

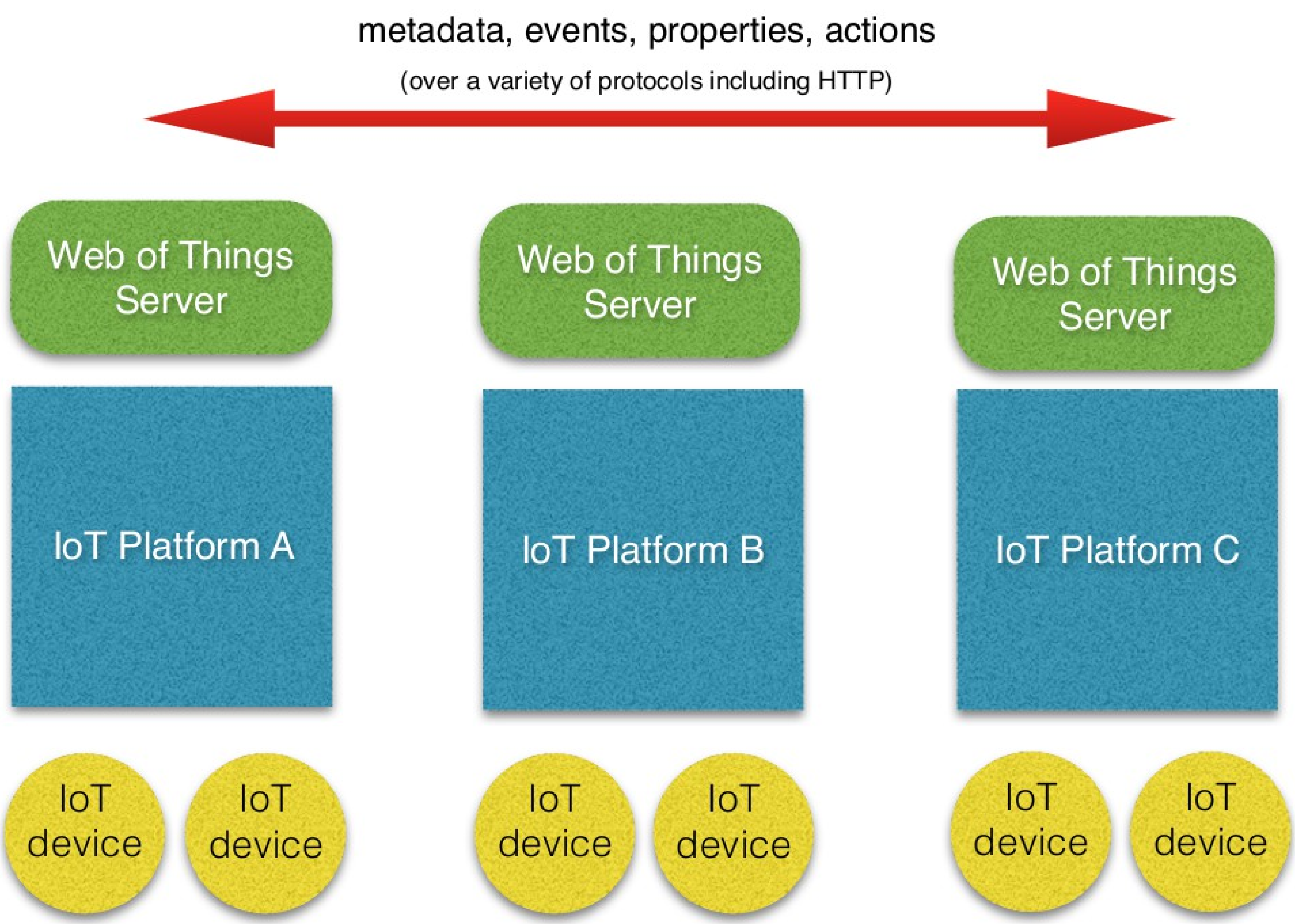
The Internet of Things

- Internet of Things is at top of hype cycle, and it will take some years yet to become mature
- Over optimistic product expectations
  - Disappointment as sales fail to perform as expected
  - More realistic: Trying ideas out, seeing what works in the marketplace, imitating the market leaders
- Lack of interoperability and lots of product silos
  - But most of the commercial benefits will accrue higher up the value chain though progressive layers of interpretation and combination with other services
- Silos block the benefits of the network effect
  - Value of network proportional to number of participants squared

The Web to the Rescue . . .

- We can use the Web to connect up services across different IoT platforms
- Extending the Web from a Web of Pages into a Web of *Things*
- *Things* as representations of physical or abstract entities
  - Virtual objects that reside on Web servers
  - Modelled in terms of events, properties and actions
  - Formal basis in terms of Linked Data
- Web architecture at its core is about *addresses, protocols* and *declarative formats*
  - Declarative formats as basis for describing behaviour & discovery
    - From HTML for pages to a ***Thing Description Language*** for things

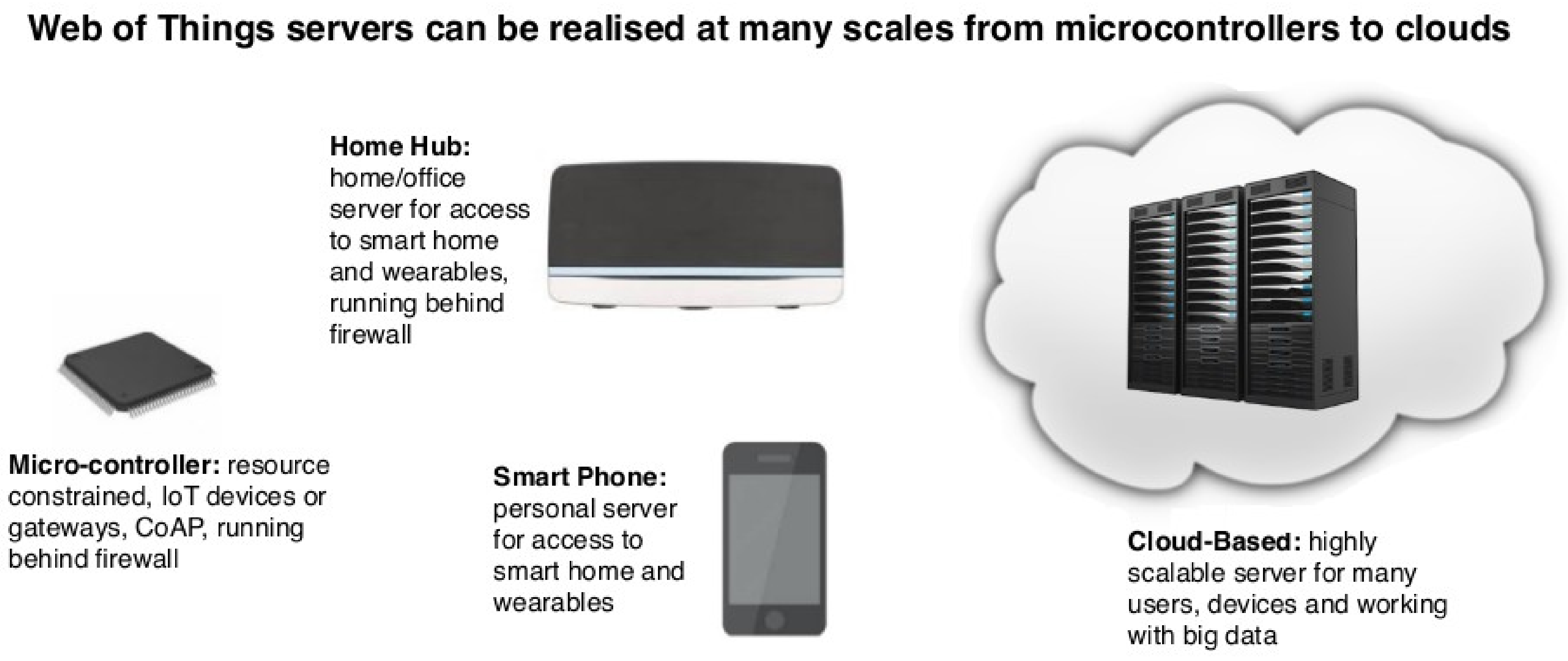
The Web as the Global Data Bus



The Web of Things

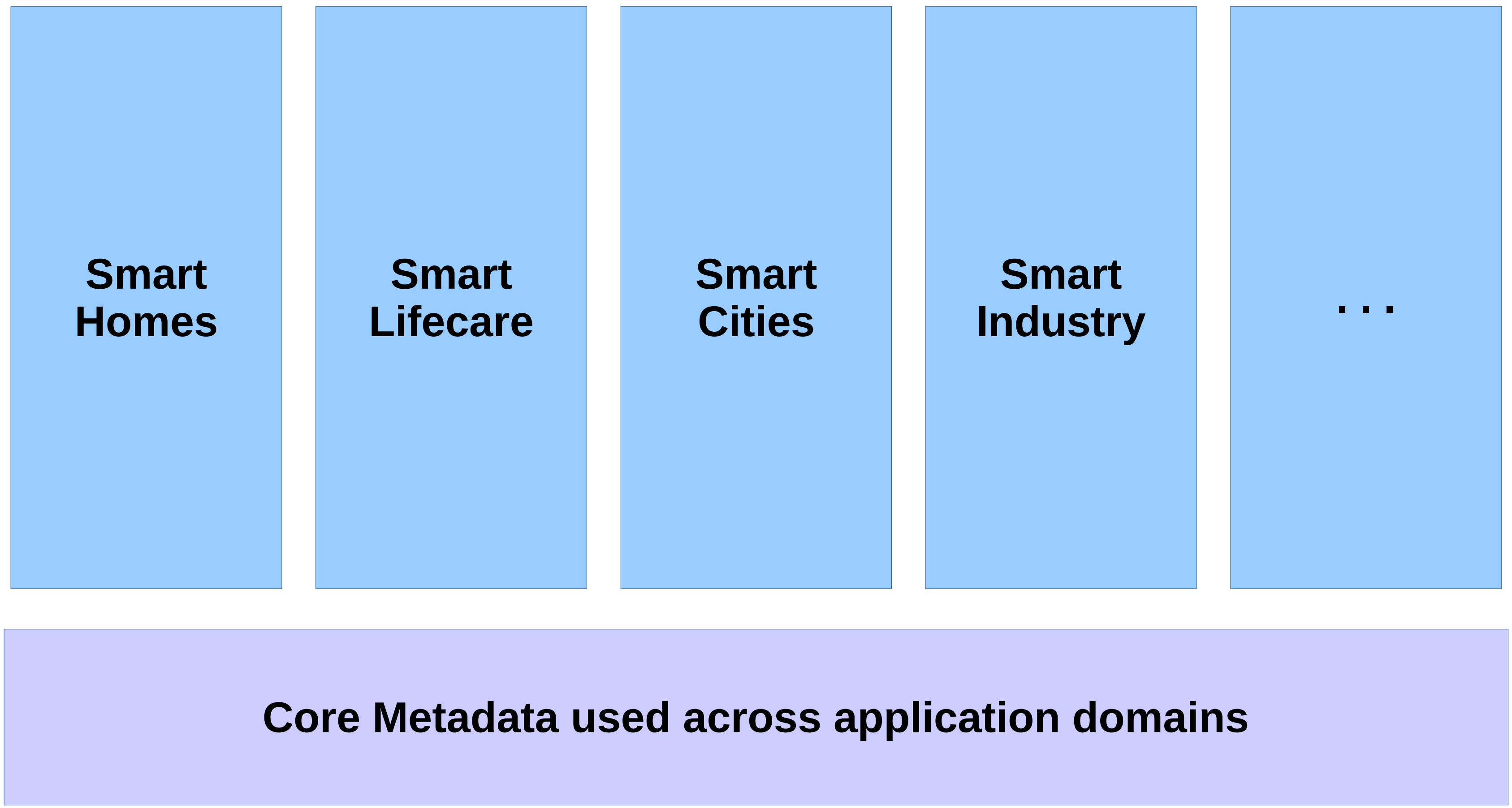
- W3C is one of the few organisations capable of establishing open standards that will enable discovery and interoperability world wide
- We want to connect IoT platforms via the Web
- Abstraction layer sitting on top of transport protocols
  - HTTP as good as it is, isn't always the answer
    - Web Sockets, CoAP, MQTT, XMPP, …
  - Interoperability based upon shared semantics, protocols, data formats and encodings
    - Building upon W3C's solid foundations for describing metadata
- Simplifying scripting for web developers
  - Decouple messaging protocols and discovery mechanisms
  - Things as virtual objects for physical and abstract entities
    - Thing metadata, events, properties and actions
    - Reducing the cost for developing and maintaining IoT services

Web Servers at Many Scales



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

Horizontal & Vertical Metadata



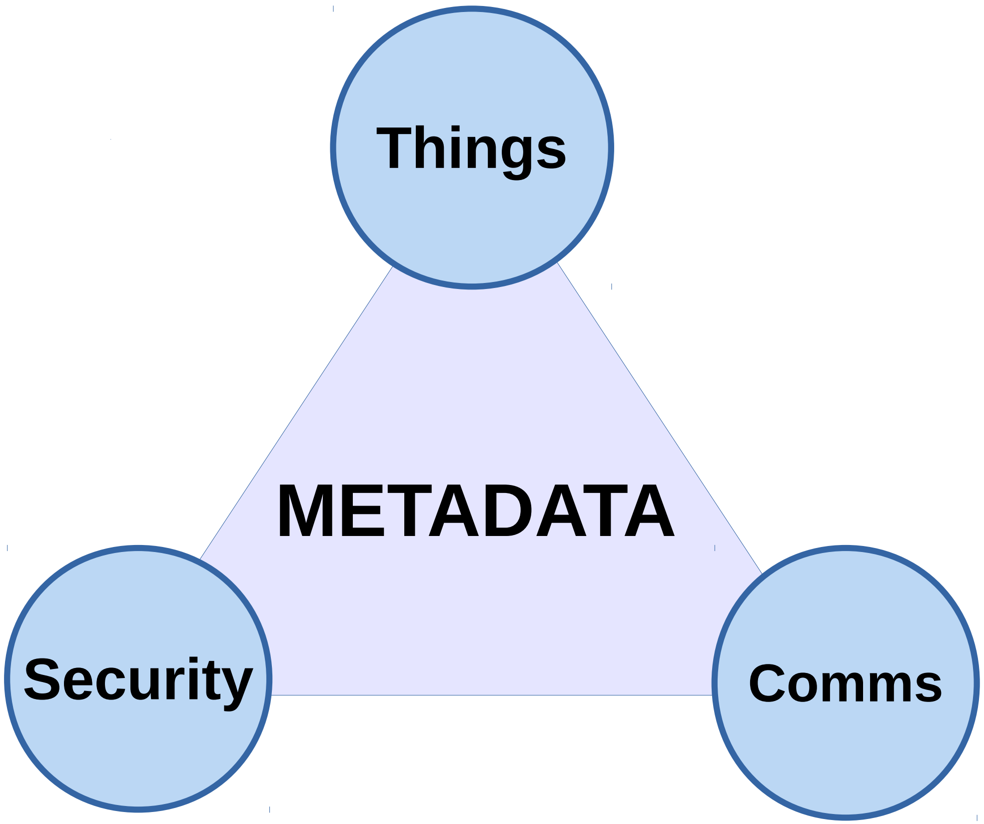
Industry specific groups are in best position to define metadata for each vertical

Shared Vocabularies

- Efficient handling of very large amounts of data
- The value of data is increased when it uses shared vocabularies
- Also critical for interoperability of services
  - Otherwise costs go up due to need for intermediaries who can bridge the gaps
- What can we all do to incentivise use of shared vocabularies?
- Taking into account different attitudes in different communities
  - We all see the world through the prism of our experience

Focus of W3C Contribution

Core metadata applicable across application domains



- **Thing descriptions**
  - Links to thing semantics
  - 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

Example

- Let's consider a example for a hotel room
  - Door has a card reader and a bell
  - Room has a light
- We want to unlock the door and turn on the room's light when the correct card is presented
- Describe things using JSON-LD
  - Serialisation of RDF in JSON
  - W3C Recommendation Jan 2014
    - http://www.w3.org/TR/json-ld/

Thing Descriptions

Server uses URI for a thing to download its description and create a local proxy object for use by scripts

- 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 URIs
Data models may be defined explicitly or by reference to an external definition

Thing as Agent

- Thing description

```
{
	“properties” : {
		“door” : {
			“type” : “thing”,
			“uri” : “door12”,
		},
		“light” : {
			“type” : “thing”,
			“uri” : “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.

W3C and The Web of Things

- Berlin workshop in mid 2014
- Web of Things Interest Group
  - Launched end of 2014
  - Use cases, requirements, shared vision
- Web of Things Working Group
  - Planned for late 2015
  - Metadata and bindings to protocols
    - Thing Description Language (JSON-LD)
- Open source projects for Web Servers
  - NodeJS, GO, Arduino & ESP8266
  - Gaining experience & involving the Maker community

IoT Connectivity

- There are many technologies and these are continuing to evolve rapidly
- IP based protocols
  - HTTP & Web Sockets (more powerful devices)
  - CoAP
  - MQTT & MQTT-SN
  - 6LowPAN (IPv6 over 802.15.4)
- Wireless
  - Cellular
  - Bluetooth Smart (formerly BLE)
  - ZigBee, IEEE 802.15.4, WiFI IEEE 802.11*
  - ETSI LTN, Weightless, LoRaWAN, SIGFOX UNB, . . .
  - KNX, EnOcean, DASH7
  - NFC
- Others
  - Bar codes, Infrared, Audio Chirps

CoAP*

- UDP analog of HTTP for constrained devices
  - RFC7252 from the IETF CoRE Working Group
  - HTTP & TCP are too memory hungry!
- Designed for RESTful services
  - Roy Fielding's representational state transfer
    - PUT & GET transfer complete state
- GET, PUT, POST, DELETE and Observe
  - Support for breaking up and reassembling resources that don't fit into a single short packet
  - No support for HTTP's PATCH method
  - Clean HTTP-CoAP mapping for gateways
- Pub-Sub mechanism
  - Interested parties register with GET and observe header
  - Notifications are sent asynchronously with Observe header
    - See draft-ietf-core-observe
- Resource discovery
  - Unicast and multicast queries (RFC7390)
  - Link format (RFC6690) analogous to HTTP Link header
    - With well defined mapping to RDF
  - GET /.well-known/core returns list of resources

*“**CoAP is aimed at tiny resource constrained devices, e.g. IoT system on a chip, where TCP and HTTP are not a good fit**”*

Matthias Kovatsch, ETH Zurich

Security is based on DTLS

- Elliptic Curve Cryptography
- Pre-shared secrets, certs or raw public keys
- IETF currently working on authentication and authorisation (ACE), and
- DTLS profiles (DICE)

IETF Class devices 1 and above
- 10 KB RAM and 100 KB Flash

In use by
- OMA Lightweight M2M
- IPSO Alliance
- ETSI M2M & OneM2M

* *Hands on with CoAP*

MQTT*

pub-sub for the masses

- Pub-sub messaging protocol over TCP/IP
  - Topic based message routing via brokers & gateways
  - MQTT-SN runs over UDP for smaller devices
- Designed for constrained devices
  - Connect, publish, (un)subscribe, disconnect
  - Message body treated as byte array
  - Smallest possible packet size is 2 bytes
- Features
  - 3 quality of service levels
    - 0: at most once delivery
    - 1: at least once delivery
    - 2: exactly once delivery
  - Retained messages (last known good value)
  - Topic wildcards
  - Last will & testament for broker to publish if client goes offline
  - Persistent sessions
  - Heartbeats

OASIS MQTT v3.1.1

- 1st byte contains
- Message type (4 bits)
- DUP flag (1 bit)
- QoS level (2 bits)
- Retain flag (1 bit)
- 2nd byte contains length in bytes
- Top bit set implies length continues in next byte (max of 4 bytes for length)
- Followed by *length* bytes as sequence of length prefixed fields
- Variable header, e.g. client Id, topic name, and packet Id
- Message payload

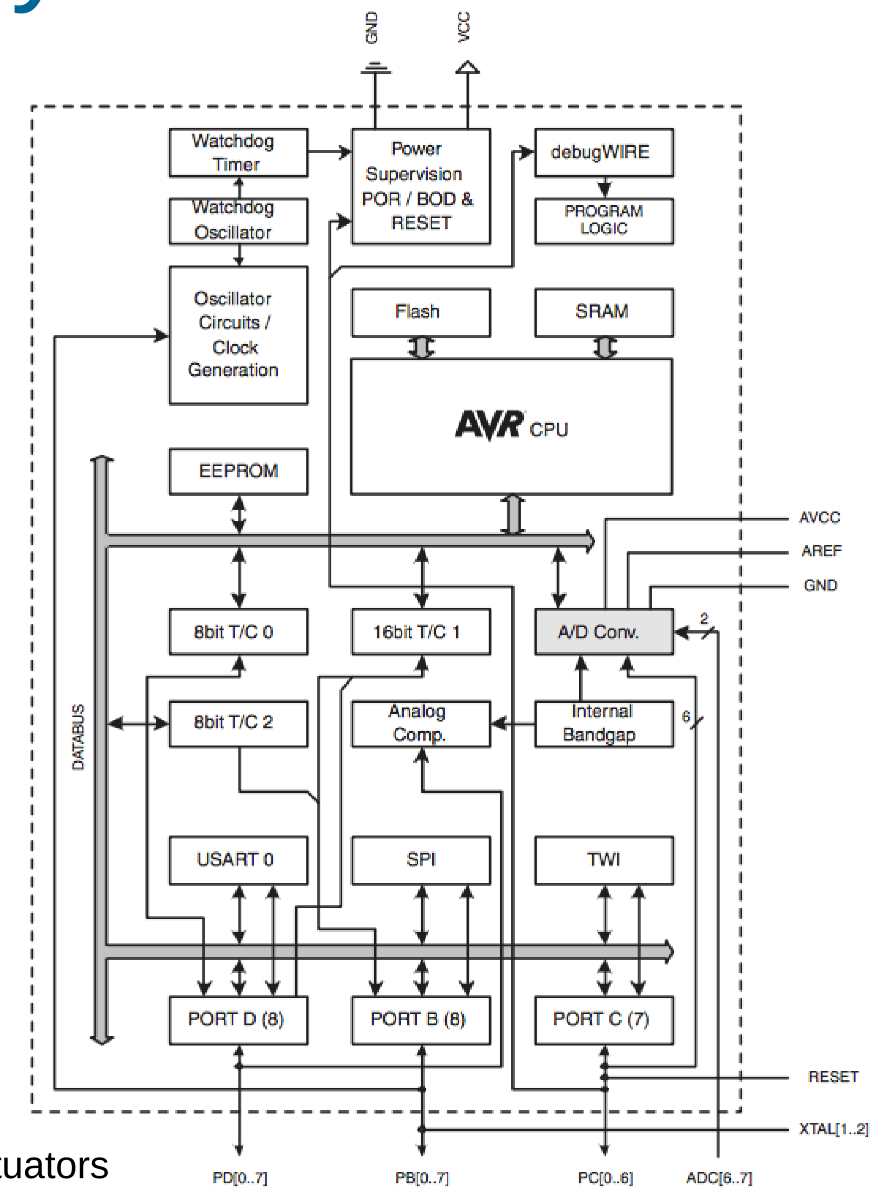
MQTT-SN over UDP

- Multicast socket based discovery of message gateway

* Originally named “message queuing message transport”

Embedded Systems

- IoT devices are typically embedded systems
  - Microcontroller plus sensors and actuators
  - Often battery operated and designed to work for months or years
    - Variety of power saving modes
      - If RAM state is not preserved, need fast-reboot
- Resource constrained
  - RAM with Kilo bytes not Giga bytes!
    - Arduino Uno uses ATmega328 which has 2 Kbytes RAM
  - Flash for program code and static data
  - EEPROM for configuration data
    - Limited number of update cycles
- Harvard vs Von Neumann CPU architecture
  - Harvard architecture has separate paths for data and code
- Interrupts, Timers and Real-Time Clocks
- Data bus between chips
  - I2C, SPI and CAN
    - Access to Flash, EEPROM, and other microcontrollers (e.g. in a car)
    - Access to sensors, e.g. MPL3115A2 barometric pressure & temperature
  - USART for serial connection to host computer
- GPIO, ADC, PWM pins for low level interfacing to sensors/actuators
  - Turn on a LED, control a servo, read analog value for ECG



Building Momentum through 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 hands on fun!*

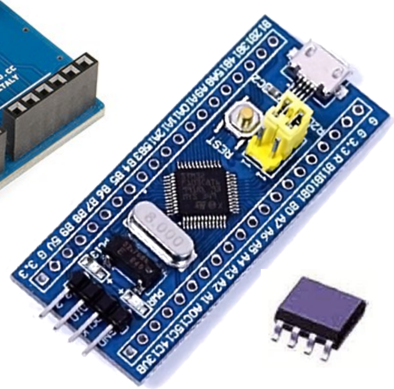
CoAP: REST for IoT devices
MicroCoAP: 16 KB including the Ethernet library, for more see:
https://github.com/1248/microcoap

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)

Arduino Uno
ATmega328
2 KB RAM
2.59 GBP

ARM STM32
20 KB RAM
64KB Flash
3.03 GBP

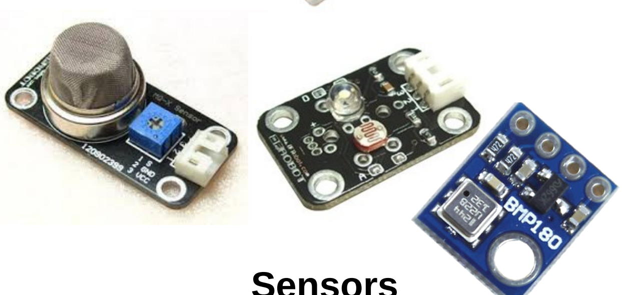


ESP8266 96 KB RAM, 512KB Flash
32 bit MCU + WiFi
1.5 GBP

ATECC108A ECC Crypto
ATAES103A AES Crypto

nRF24L01 2.4 GHz
Sensor Network
1.34 GBP

CC2530: 8051 MCU + IEEE 802.15.4
Up to 256 KB flash + 8 KB RAM
Available for 6 USD



Sensors

C++ & Arduino IDE
Lua & NodeMCU
MicroPython
RIOT OS

Open Source Servers

https://github.com/w3c/wot-arduino

- A work in progress — goal is to enable WoT hackathons in 2016
- Stretch challenge: can we create a Web of Things server that will work with the 2 Kbytes RAM in an Arduino Uno?
  - Statically allocate memory pool for JSON nodes
    - true, false, strings, numbers, objects, arrays and null
    - 6 bytes per node on ATmega328 *and* on 32 bit MCU's
      - Nodes can be formed into linked lists if needed with no extra memory
        - Assuming pool of up to 4095 nodes and a extra list node for strings
  - AVL trees for representing objects and arrays
    - Approximately balanced binary tree with 6 bytes per node
      - Assumes limit of 255 properties per object and items per array, and pool of 65535 nodes
        - Or perhaps 1023 object properties/array items and pool of 16383 nodes
    - Shares node pool with JSON
- Map names to numeric symbols when parsing a thing's data model
  - Saves memory and enables compact messages
    - Single byte for JSON tags and 200 different symbols
- Statically typed versus dynamically typed languages
  - More cumbersome to work with, but not too bad
    - C++ not nearly as nice as Lua or JavaScript

Arduino* Sketch

- C/C++ environment for Microcontrollers
- Extensive suite of libraries
- Your app is written as a “sketch”
- Compiled and copied to MCU's Flash
- USB serial connection for debug messages

```
// the setup function runs once when you press reset or power the board

#define LED 13

void setup() {
   pinMode(LED, OUTPUT);  // initialize digital pin 13 as an output
}

// the loop function runs over and over again forever

void loop() {
   digitalWrite(LED, HIGH);    // turn the LED on (HIGH is the voltage level)
   delay(1000);                // wait for a second
   digitalWrite(LED, LOW);     // turn the LED off by making the voltage LOW
   delay(1000);                // wait for a second
}
```

* Named after the Italian king “Arduin” who reined from 1002 – 1014

Agent using C++

- The agent's model declares the door and light as properties
- The server downloads the models for the door and light, and creates proxies for them
- It then calls the agent's initialisation code
- The dictionary of names to symbols is then discarded
- The sketch uses global variables to keep track of things and symbols
- Door and Light use similar code along with hardware interrupts and GPIO pins to interface to the sensors and actuators
- Server supports single threading model to avoid complications with code execution in interrupt handlers

```
Thing *agent, *door, *light;
Symbol unlock_sym, on_sym;

void setup() {
  RegisterThing(agent_model, init_agent);
}

void init_agent(Thing *thing, Names *names) {
  agent = thing;
  door = agent->get_thing(names, F("door"));
  light = agent->get_thing(names, F("light"));
  unlock_sym = names->symbol(F("unlock"));
  on_sym = names->symbol(F("on"));
  agent->observe(names, F("key"), unlock);
}

void unlock(JSON *valid) {
  if (JSON_BOOLEAN(valid)) {
    door->invoke(unlock_sym);
    light->set_property(on_sym, JSON_TRUE);
  }
}

void loop() {
  DispatchEvents();
}
```

Note: PROGREM and F() macro direct compiler to save strings in program memory

Authentication

- W3C is seeking to move the Web away from user name and password
  - Increasing emphasis on public key cryptography
  - Learning lessons from experience with PKI
- New Web Authentication WG planned with support from the FIDO Alliance and other groups
    - Multi-factor authentication as appropriate to context
    - Focus on assuring that this is the same device+user as when the user account with the website was originally set up
    - Does ***not*** address binding of Web Identity to Real-World Identity
- W3C hardware based Web Security WG
  - Leveraging secure elements of various kinds including SIMs
    - Secure tamper-proof storage and computation
      - Provisioning opportunities and management of updates

Credentials

- Attestations by trusted 3rd party about the attributes of an identity
  - Needed to tie web identity to real-world identity
  - Applicable to people, IoT devices, services, ...
- Increasingly important for an online world
- Ephemeral vs Long Lived credentials
  - Reduced risks through short lived credentials issued against a session ID
    - Potential role for secure elements
- W3C is collecting use cases and requirements with a view to a Credentials Working Group

Privacy and Contracts

- The IoT makes attention to security and privacy particularly important given the amount of personal or confidential information that can be collected by sensors
- Privacy laws vary considerably across jurisdictions
- Contract law by contrast is much more uniform
- Terms & conditions as basis for binding agreement between service providers and service consumers
  - Including the liability taken on by the service provider
- Used in conjunction with access control
  - Dependency on identity management and authentication
- Precedent of Creative Commons 3 level agreements
  - Icons
  - Human readable
  - Legal details for lawyers

Simplifying Discovery

- Many different ways to discover things
  - mDNS, UPnP and other local area network techniques
  - Bluetooth (beacons), ZigBee
  - NFC, barcodes, IR and audio chirps
  - By following dependencies in Thing descriptions
  - Devices can register themselves in hubs or cloud
  - Social relationships between people and things
    - Personal and organisational zones
    - Spatial (geographic) zones, temporal zones
      - Events and processes as abstract entities
- Simplify discovery via agent API
  - Context based discovery queries
    - Aided by semantic descriptions
  - Agents can collaborate but should respect privacy

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

Network Efficiency

It is all in the metadata!

- Smart meters vs Security Cameras vs ...
  - Small amounts of data that isn't time critical
  - Large amounts of data that is needed in real-time
  - Privacy sensitive data e.g. health sensors
- Multiplexing data from sensor networks
- Pushing Interpretation to the Network Edge
  - Upload scripts to Web of Things server (hubs)
  - Reduces amount of data to be sent over network
- Pushing control to the Network Edge
  - Clock synchronisation across group of controllers
    - Coordinated control of actuators, e.g. traffic lights, factory floor
- The need to collect representative use cases

Varying kinds of data

- Different kinds of sensors and actuators have very different kinds of data requirements
  - This needs to be reflected in the metadata
- Simple sensors where you just need the latest value, e.g. a temperature and humidity sensor
- Sensor streams where you need a log of readings over time
  - Ability to query data for a specific time or time window
  - Composite data values for each reading
  - Interpolation between readings for smoothly varying properties
    - Programming path of robot hand via smooth control of its joints
- Real-time sensor streams
  - ECG as example of healthcare sensor stream
  - Remotely controllable Security Camera
    - Higher bandwidth and need for low latency
    - Role of events to draw attention to a given sensor

Provisioning and Resilience

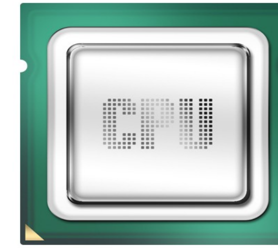
- Reducing provisioning and operating costs
  - Avoid $50 provisioning cost on a $1 device
  - Reduced costs enables new business opportunities
- Bringing a new device into service and decommissioning old devices and services
  - Discovery, authentication, binding to real-world identity
- Software and security updates
  - Best security practices plus patches for security flaws
- Monitoring for faults and security attacks
  - Graceful degradation with dynamic adaptation
- Managing dependencies across a distributed system
  - Weakly coupled systems scale better
    - Late binding, knowing what can be safely ignored and what can't
  - Lessons from Linux package/library management

Unlocking the Silos

Let's work together!

- The Internet of Things is still very immature
  - There is a lot of work to be done to realise the full potential
  - The role of the Web for reducing costs and Web scale integration
  - Be part of the solution for security, privacy and resilience
    - Support W3C work on authentication and secure hardware
  - Open architectures that can support a wide range of contexts
- Importance of relevant use cases for driving standardisation
  - Opportunities to help with understanding use cases for given domains
  - Examples that are driven by Big Data, and associated scaling challenges
    - Incentives for sharing vocabularies and unlocking data silos for value added services
    - Requirements for provisioning and managing large scale distributed systems
  - Share your experience with designing and deploying systems
    - Experimental work and practical experience are key to defining effective standards
- SDO's need to collaborate on converging IoT related standards if we are to realise the benefits of the network effect
  - European companies need to identify and drive this convergence

Beyond the Web of Things



http://www.w3.org/2014/10/29-dsr-wot.pdf

- We're now extending the Web from a Web of Pages to the Web of Things
  - But this is only the first step . . .
- It will soon be time to extend the Web from the Web of Things to the **Web of Thought**
- Web based assistants that understand everyday things and can communicate with us at our level
  - People and personal relationships, space, time, causality and naïve physics, tools, the natural world, the urban world, story telling, humour, emotions, empathy, personality traits
  - Avatars that **forget** like we do, something crucial to how we think
  - and much much more ….
- We need interdisciplinary discourse
  - Today's computer science is still in its infancy
  - To move forward we need to combine ideas from Computer Science, AI and Cognitive Science and the likes of John R. Anderson and Marvin Minsky
- Learning like we do from instruction and assessment
  - Lesson plans for cognitive AI's based upon taxonomies of common sense

Questions?