Web of Things as abstraction layer across platforms
- Our aim is to enable open markets of services on a Web scale
- Multiple platforms & protocols are inevitable
- Enabling discovery and composition of services

Things stand for physical or abstract entities
- As software objects exposed to applications
  - With data models for properties, actions and events
- As named entities with rich descriptions
  - Based upon W3C’s work on Web Architecture & Linked Data
  - Used for semantic interoperability across platforms
  - Horizontal and vertical metadata vocabularies (see next slide)

Data models
- Core types, e.g. boolean, string, number, array, ...
- Things and streams as first class data types
- Early and late binding
- Integrity constraints for robustness
- Multiple serializations, e.g. JSON, XML, ...
- Need to be usable on resource constrained devices
Core metadata vocabularies used across application domains

**Thing metadata**
- Links to thing semantics
- Data models and relationships between things
- Dependencies and version management
- Discovery and provisioning
- Bindings to APIs and protocols

**Security & market related metadata**
- Security practices
- Mutual authentication
- Access control
- Terms and conditions – relationship to “Liability”
- Payments
- Trust and Identity Verification
- Privacy and Provenance
- Safety, Compliance and Resilience

**Communication-related metadata**
- Protocols and ports
- Data formats and encodings
- Multiplexing and buffering of data
- Efficient use of protocols
- Devices that are asleep most of the time

Horizontal and Vertical Metadata
## Communications Stack

<table>
<thead>
<tr>
<th>Application</th>
<th>Scripts that define thing behaviour in terms of their properties, actions and events, using APIs for control of sensor and actuator hardware</th>
</tr>
</thead>
</table>
| Things      | Software objects that hold their state  
Abstract thing to thing messages  
Semantics and Metadata, Data models and Data |
| Transfer    | Bindings of abstract messages to mechanisms provided by each protocol, including choice of communication pattern, e.g. pull, push, pub-sub, peer to peer, etc. |
| Transport   | REST based protocols, e.g. HTTP, CoAP  
Pub-Sub protocols, e.g. MQTT, XMPP  
Others, including non IP transports, e.g. Bluetooth |
| Network     | Underlying communication technology with support for exchange of simple messages (packets)  
Many technologies designed for different requirements |
Distributed Web of Things

Software objects for things standing for physical or abstract entities

- Thing descriptions can be used to create proxies for a thing, allowing scripts to interact with a local proxy for a remote entity.
- Scripts can run on servers or as part of Web pages in a Web browser for human machine interface.
- Thing topologies
  - Peer to Peer, Peer to Peer via Cloud, Star, Device to Cloud, Star to Cloud

Servers on devices from microcontrollers to cloud-based server farms, with a wide variety platforms, protocols and standards.
Points raised in the submissions

- Need for informational and interaction models
- Interaction model with properties, methods, signals (aka actions and events)
- Question around need to be able to pass through blobs that others will understand
- Annotations in respect to reliability and trust
- What we can learn from human languages cognitive science and progress in AI
- Best practices for ontologies
- Some existing ontologies as a starting point
- Upper versus Lower ontologies (what are these?)
- Modularity and orthogonality
- Scalability challenges including evolution over time and communities
- Discoverability – search engines for IoT services
- Importance of security and privacy
- Need for security metadata
- Need for abstracting away from protocols and implementation details
- Must be machine interpretable
- Need for convergence on information models
- Potential for bottom-up, open and crowd-sourced approach
- Interoperability has to happen at all layers
- Independence from IoT network topology
- Syntactic noise - need for easier to understand representations
- A common underpinning for terms (RDF)
- What do we mean: resources vs things?
- Late binding and the challenges it brings
Encouraging Re-use and Coping with the Inevitability of Change

• Data models, information models, and code
  • Impact of variations in requirements
  • Formal vs informal approaches to semantics
  • Best practices for modular vocabularies
  • Convergence across IoT alliances & SDO’s
  • Easy access to existing vocabularies
  • Healthy open source ecosystems – role of “maker” communities

• Sharing of data across organizations
  • Easy access to open market of services & intent based search
  • B2B, access control, terms & conditions, privacy policies
  • Role of intermediaries for bridging the gaps
  • Federation vs centralization

• Coping with evolutionary change in systems of systems
  • Inevitable drift in weakly coupled communities
  • Named fields that you can ignore only gets you so far
  • What can we learn from Linux package management?