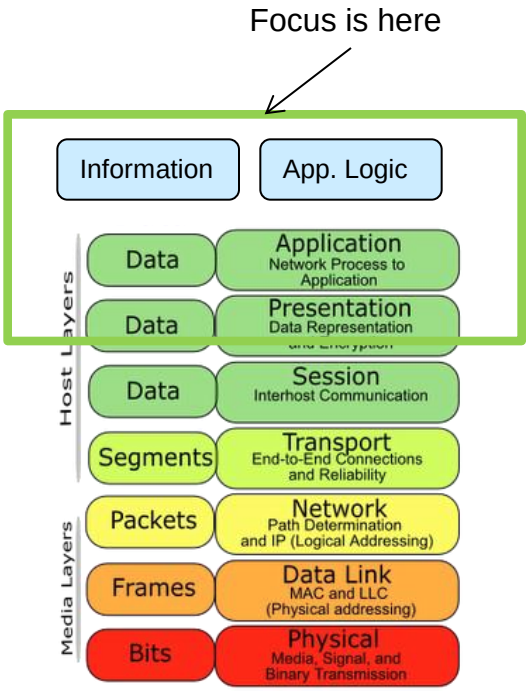
# Interoperability on the Internet of Things

**Artem Katasonov**

**@ University of Jyväskylä, 06.02.2014**
**for TIES410 “Future Internet”**

# Outline

- Some definitions
- IoT interoperability at present
  (what companies show at CES)
- A vision of the future of IoT
  (interoperable and supporting innovation)
- Web of Things
  (reliance on the Web architecture as one enabler of interoperability on IoT)
- Semantic technology
  (for interoperability and openness on the information-level)



# Eeh.. what?

- **Interoperability** is the ability of making systems and organizations to work together (inter-operate). While the term was initially defined for information technology or systems engineering services to allow for *information exchange*, a more broad definition takes into account social, political, and organizational factors that impact system-to-system performance.

- The **Internet of Things** (or **IoT** for short) refers to *uniquely identifiable objects* and their *virtual representations* in an Internet-like structure.

from Wikipedia aka. “the source of all knowledge”

# IoT technology worlds


- *Attached devices*: Identifiers such as RFID tags or barcodes are attached to things to enable their automatic identification and tracking. Based on a thing's identifier, the information about the thing is then retrieved from a database or from the Web.
- *Sensing and Actuating devices*: These devices are placed in the close vicinity of 'things' and provide a "second-hand" access (from outside) to their properties or functions. Examples are temperature and other sensors, cameras for vehicle register plate recognition, and actuators like remotely-controlled lights, locks or window blind controls.
- *Embedded devices*: Some 'things', such as industrial machinery, home electronics, smart phones, and wearable devices, have embedded processors, data storages, sensors and actuators, enabling "first hand" access (from inside) to them.

# Why IoT?

- IoT will revolutionize the *telecommunications sector* by enabling global connectivity between physical objects (rather than connecting places or people).
- IoT will revolutionize the *World Wide Web* by bringing real-time machine-published information to the Web.
- IoT will revolutionize *cyber-physical systems* (CPS) by making the processing power of the Cloud available to local ubiquitous applications.

# Interoperability at present

## (in IoT-related systems)

# CES: “Screen-to-screen” media sharing is everywhere


- Some solutions use video streaming, some original content sharing

# CES: Phones are connected to absolutely everything



*Smartphone as a "remote control for life"*

# New in CES 2014: LG Home Chat


- Uses Natural Language Processing algorithms
- Base on LINE instant messaging

# CES: “Smart” fridges



- Can check your Facebook
- Can send a picture from smartphone
- Fridge can push to a smartphone alarms about expiring goods

# CES: “Smart” washing machines


- Can check on laundry progress from TV

# CES: Fridge communicating with an oven

- Fridge can have food recipes stored
- A recipe can be sent to the oven
- Oven will set the cooking temperature and other settings according to the recipe

# CES: Home automation is always there, but little news



- Light bulbs with integrated Z-Wave

# Wireless radio technologies for IoT











- Device -> phone -> Internet
- Typical in fitness, healthcare devices

- Device -> gateway -> Internet
- Typical in home automation

- Device -> cellular infrastructure -> Internet

# New in CES 2014: Revolv hub

# Wi-Fi for IoT





- Energy consumption is high, so cannot be battery-operated
- Do not need a special gateway, just a Wi-Fi access point -> quite popular as a result
- Work on Wi-Fi low power is ongoing -> if successful, may be the killer of all other short and medium range approaches
- IP-based from first link, backed by e.g. Cisco

# GPRS/3G for IoT

- Right now, e.g. on-street pay-and-display machines have SIM cards in them and use GPRS to send data to corporate back-ends.
- Cellular networking equipment manufacturers (Ericsson, Nokia) and network operators prefer a future where most of IoT devices would be connected to Internet via a cellular connection.
- Advantages: range, no own gateway needed.
- Inserting a SIM card into each sensor may be a hustle, but software SIMs are likely to be developed.

# So, interoperability?

- Well, some devices are shown interoperating. As such, the future looks “cooler” every year.
- But, right now it always one of:
  - Buy all the devices from one vendor,
  - Connect “smart” devices (phones, TVs) from different vendors through installing a particular software client (from one vendor) on each of them (limited list of supported platforms),
  - Use a particular gateway box, then can connect devices from different vendors (from a limited list of supported by the gateway).
- In all three cases, a single vendor is responsible for all of the “interoperability”.

# A vision of the future of IoT

## (interoperable and supporting innovation)

# Current issues

- There is currently a rather large number of various and non-interoperable IoT platforms, each with a rather small market penetration.
- Some platforms do not even offer any application programming interfaces (APIs) and thus end users have access only to software developed by the hardware vendor itself (e.g. Revolv).
- In cases, where an API is available (e.g. There, SmartThings):
  - Lack of interoperability leads to a need to develop software separately for each particular IoT platform.
  - Small market penetration of each platform is discouraging to developers.
  - In result: it is still mostly the software from the hardware vendor that is available to end users.

# Vision

**Can 3rd-party software application development for IoT environments like smart homes become as easy and as popular as the development of applications for smartphones nowadays? Is "IoT App store" possible?**

We would like IoT to provide:

- Ability to interconnect devices from different vendors.
- Ability to have gradually growing IoT environments, in contrast to a need to install and interconnect all IoT devices and software at once.
- Ability of 3rd parties to develop software applications for IoT environments, in contrast to applications coming only from the devices' vendors.

Plus, unless the number of IoT platforms will converge to 2-3 dominant ones:

- Ability to develop applications that are generic in the sense of being able to run on various IoT platforms (different vendors, same purpose), in contrast to developing applications for a very particular platform.

# A simple (yet difficult) scenario

1. Mary has at her home a remotely-controlled heater, an indoor temperature sensor in the living room, and a control software that adjusts the heater's power based on the temperature and Mary's preference settings. All devices and software are from some Vendor A.

2. Mary buys another temperature sensor, from a Vendor B, and it is installed in the bedroom. Let's assume that:
   - Control software is able to work with more than one sensor (e.g. to keep the temperature in both rooms close to Mary's preferences).
   - There is no common standard on how temperature measurements are encoded and transmitted. Vendor A and Vendor B use different approaches.

Problem: How to enable "Plug-n-Play", i.e. that software of Vendor A is able to utilize the sensor of Vendor B (mostly) automatically?

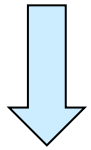
# A simple (yet difficult) scenario cntd.

3. Mary learns that there exists a software application from some Vendor C that is capable of energy-saving predictive heating control that takes into account not only the indoor temperature but also the changes in the outdoor temperature:
   - Mary wants to use this application instead of the control software she has in place. She downloads it from a Web-based store, or runs a service instance in the Cloud (the latter is probably easier to realize).
   - The application discovers the heater and the two indoor temperature sensors Mary has at home.
   - It then fails to find an outdoor temperature sensor in Mary's home network and opts to use an appropriate Web-service (current data or a forecast) for that.

Problem: How such a decoupling of hardware (and other data providers) from software applications can be possible, in particular, that generic applications are able to configure themselves to work with various sensors / devices / services?

# IoT supporting innovation

Variety of companies and individuals able to develop and publish IoT software applications (as they develop smartphone applications at present)



A plethora of *innovative* applications and ideas for future evolution of hardware

*(not stuff like checking Facebook on a fridge)*

Safety and security become even more important issues then…
(outside the scope here)

# Two approaches

- *Standardize everything*

*(difficult to reach agreements, difficult to cover everything in an evolving world)*

- *Some level of standardization + some intelligence*
  - *E.g. Web protocols and formats (W3C recommendations)*
  - *Plus intelligence through semantic technology*
    - *Requiring formats/languages for encoding metadata and ontologies (W3C recommendations)*

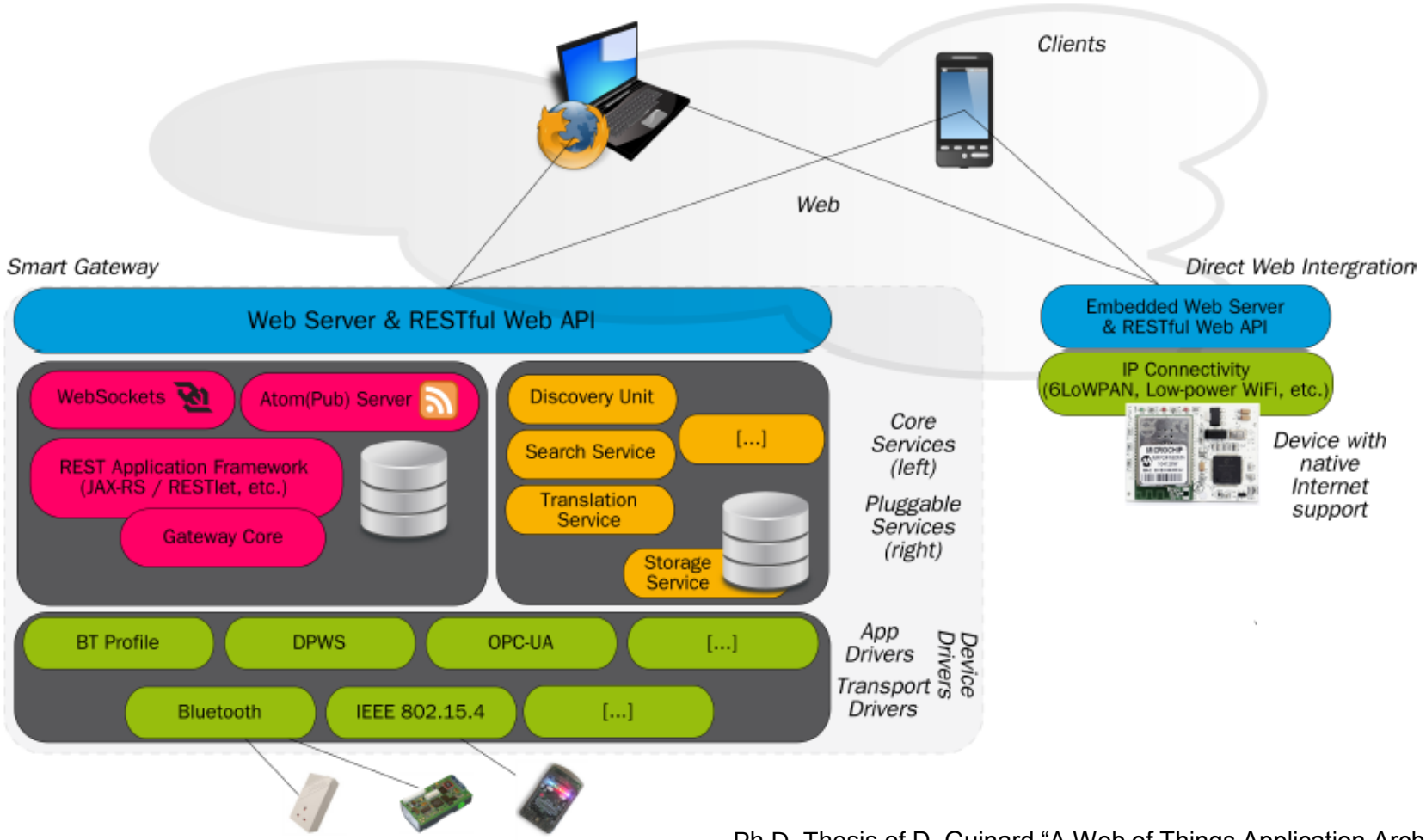
# Web of Things

# Web of Things

- A trend in IoT area is to attempt to integrate ‘things’ seamlessly with the *existing Web infrastructure* and to expose connected ‘things’ uniformly as Web resources, resulting in what is called the Web of Things (WoT).

- The aim is to reuse *the architectural principles of the Web* and apply them to the connection with the real world, i.e. with (smart) entities e.g. smart fridges (with embedded computers), smart packages (with RFIDs), smart rooms (with sensors and actuators), thereby making them first-class citizens of the Web.

# Integrating the 'things' into the Web



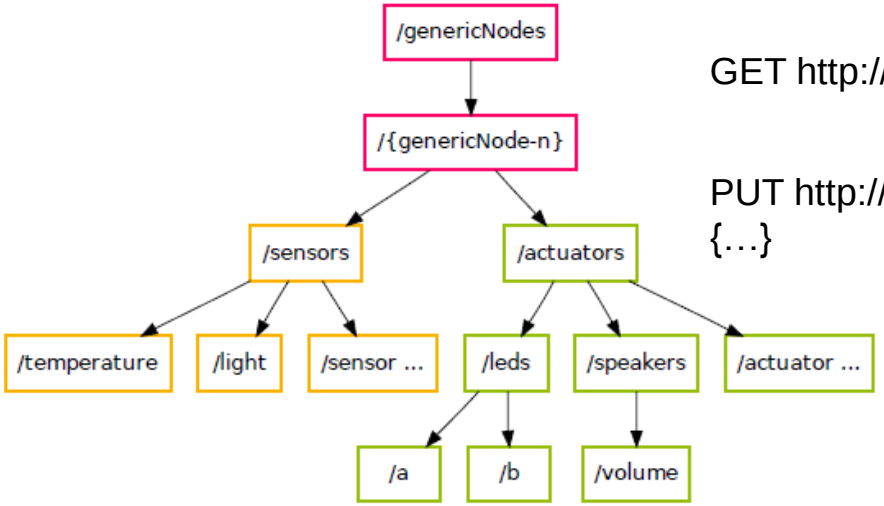

Ph.D. Thesis of D. Guinard "A Web of Things Application Architecture - Integrating the Real-World into the Web"

# REST requests

*REST: "*Representational state transfer"
*RESTful*: stateless, the server does not have to keep the history of past requests to process and answer a request.

Use of HTTP operations and hierarchical URLs

- GET is used to retrieve the representation of a resource.
- POST creates a new record about resource (includes payload).
- PUT is used to update the state of an existing resource or to create a new resource record (includes payload).
- DELETE is used to remove a record about a resource.




GET http://<domain>:<port>/generic-nodes/1/sensors/temperature

PUT http://<domain>:<port>/generic-nodes/1/sensors/temperature/value
{…}

Ph.D. Thesis of D. Guinard

# REST responses and payloads

- JSON (easiest to process from JavaScript, i.e. in client Web apps)

or

- XML
- CSV
- RDF

```
{"resource":
        {"name": "Temperature",
         "content":
                {name":"Current Temperature",
                "description":"Ambient Temperature",
                "value": 24.0,
                "unit": "celsius"}
        }
}
```

Ph.D. Thesis of D. Guinard

# Example of a REST API: ThereGate


GET http://<domain>:<port>/api/services/22/get


```
{
   "tobj": "",
   "Availability": 0,
   "HistoryUpdated": [
      "na",
      0,
      0
   ],
   "Temperature": 23.5,
   "Name": "Temperature sensor 1"
}
```


GET http://<domain>:<port>/api/services/22/get?Temperature


```
{
   "Temperature": 23.5,
   "tobj": ""
}
```


GET http://<domain>:<port>/api/services/25/set?SwitchState=1

Result: light on

# Semantic Interoperability

# Is WoT enough?

- Applying Web architecture to Internet of Things is a great facilitator of interoperability.
- WoT, however, is mostly about the protocols and formats.
- WoT as such will not enable realization of our “simple (yet difficult) scenario” (from earlier slides):
  - E.g. two temperature sensors both delivering measurements over HTTP GET as JSON, but of different structure and with different object/property names.
  - E.g. two heater devices accepting commands over HTTP PUT as JSON, but of different structure and with different object/property names.
- For true interoperability, we need also the semantic interoperability, the ability of the devices to unambiguously convey the meaning of data they communicate over Web protocols.

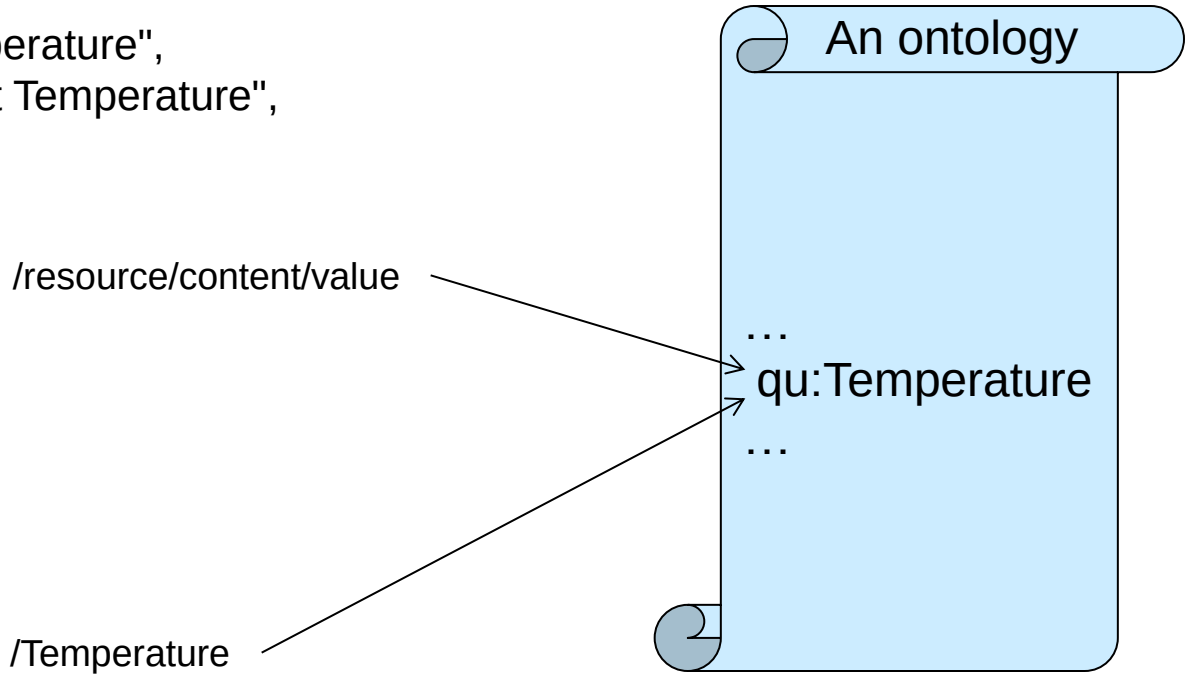
# One slide summary: What is semantic technology?

- Question: How to make computers act in an intelligent way?
  - Approach 1: To make computers ***so clever*** that they will be able to process the information about the world in its ***full complexity***, e.g. understand human language => traditional Artificial Intelligence (AI)
  - Approach 2: To ***simplify*** the description of the world to a level that even ***stupid*** computers will be able to act “intelligently” based on it => Semantic Technology (aka. Linked Data).
  - Approach 3: To record data about **so many real cases**, so that most decision problems can be solved via “nearest neighbor” search => “Google” way
- The goals of semantic technology:
  - to make the meaning of data is as **explicit** as possible: as **unambiguous** as possible and as **context-independent** as possible.
  - to **link** data sources globally: more meaning with same data (small messages, fetch the rest from Web).
- Achieved through “three pillars” of the semantic technology: *Semantic Network data model*, *URI*, and *Ontologies* (formal controlled vocabularies).

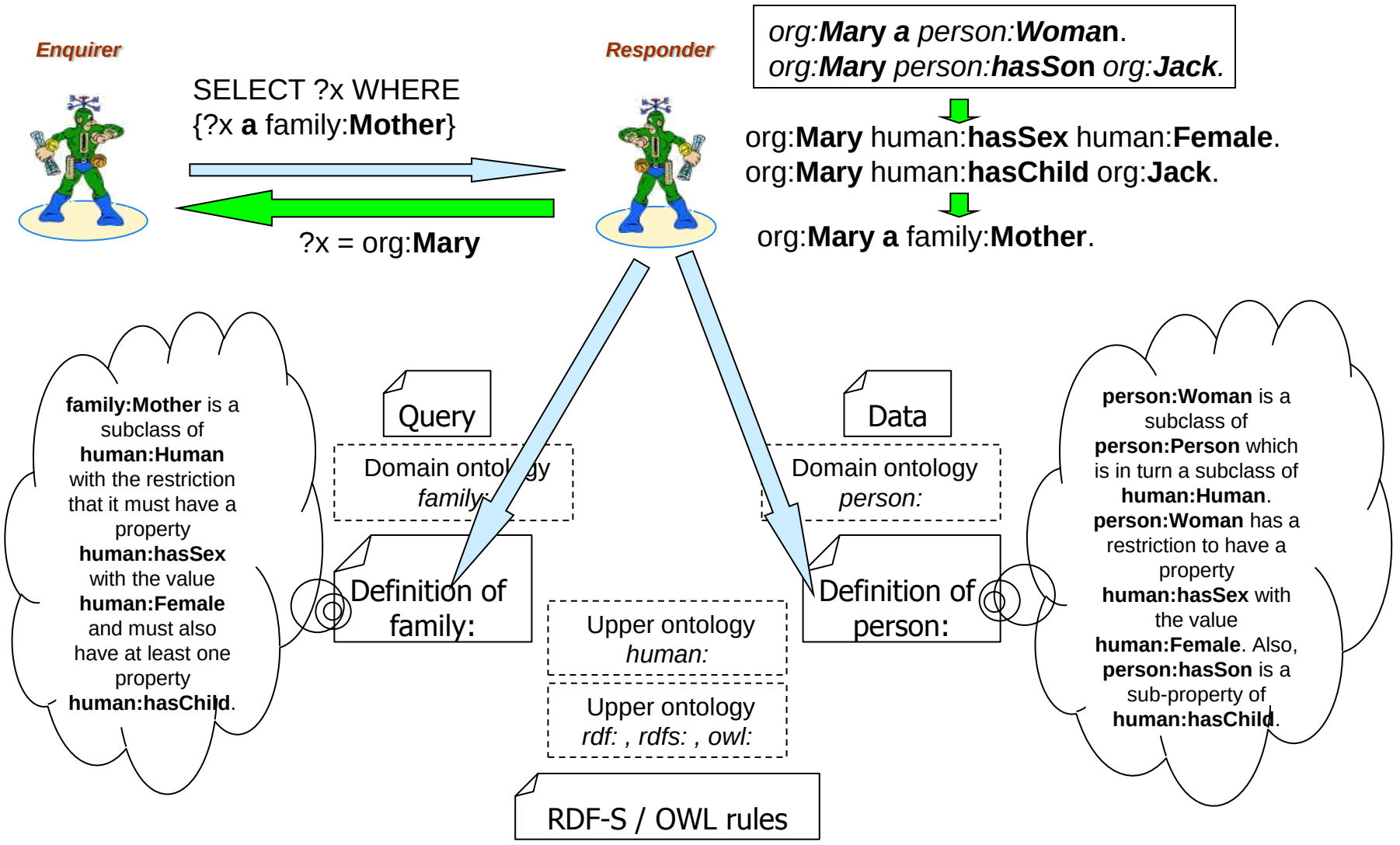
# Semantic annotation

```
{"resource":
        {"name": "Temperature",
         "content":
                {name":"Current Temperature",
                "description":"Ambient Temperature",
                "value": 24.0,
                "unit": "celsius"}
        }
}
```

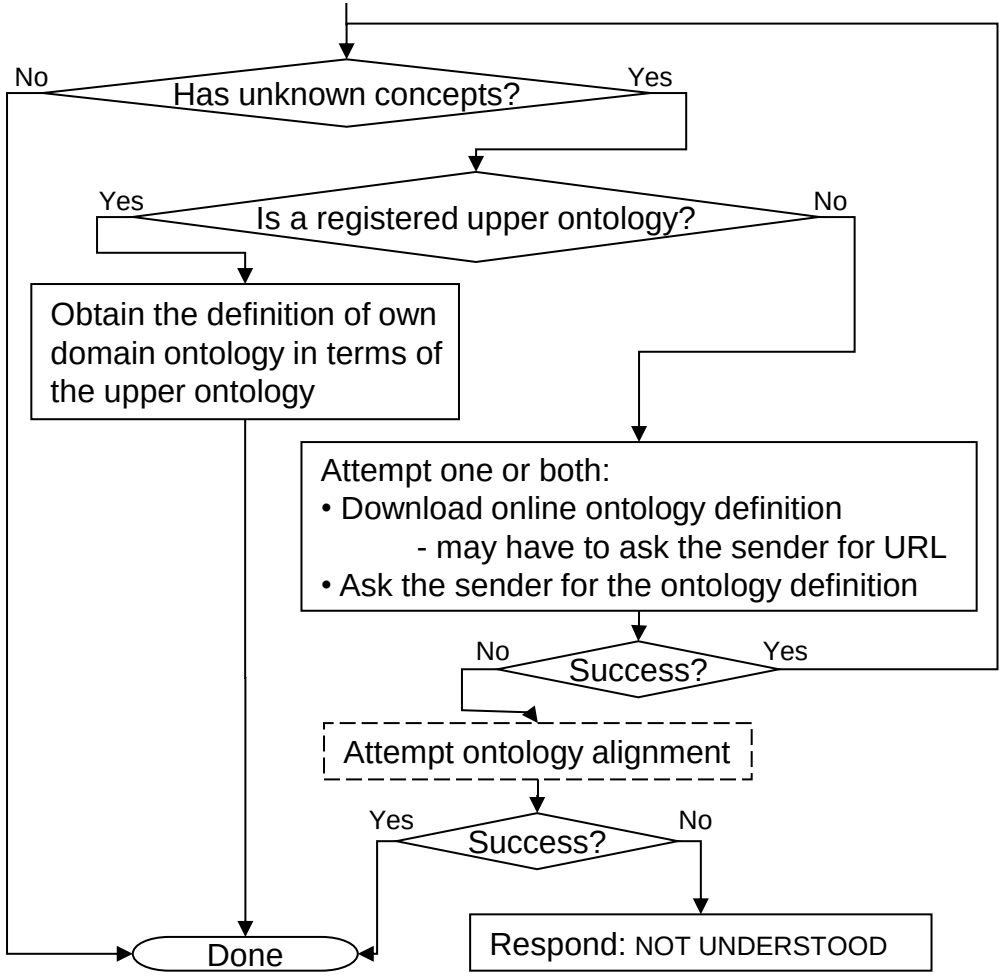
```
{
   "tobj": "",
   "Availability": 0,
   "HistoryUpdated": [
      "na",
      0,
      0
   ],
   "Temperature": 23.5,
   "Name": "Temperature sensor 1"
}
```



# Semantic reasoning



# Ontology linking process



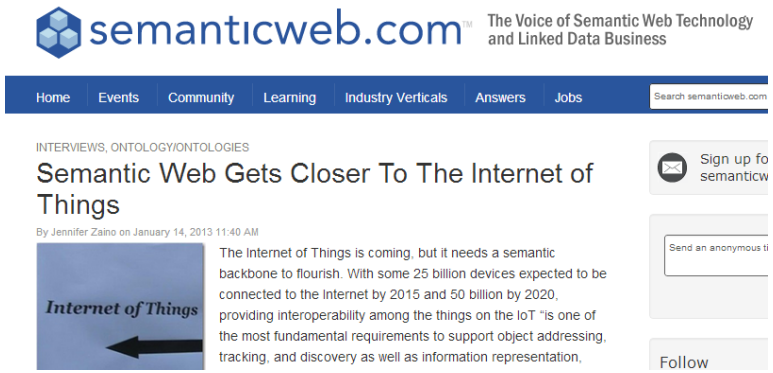

Katasonov A. and Terziyan V. (2009) *Semantic Approach to Dynamic Coordination in Autonomous Systems*, In: Proc. 5th International Conference on Autonomic and Autonomous Systems (ICAS'09), April 20-25, 2009, Valencia, Spain, IEEE, pp.321-329

# Ontology Alignment

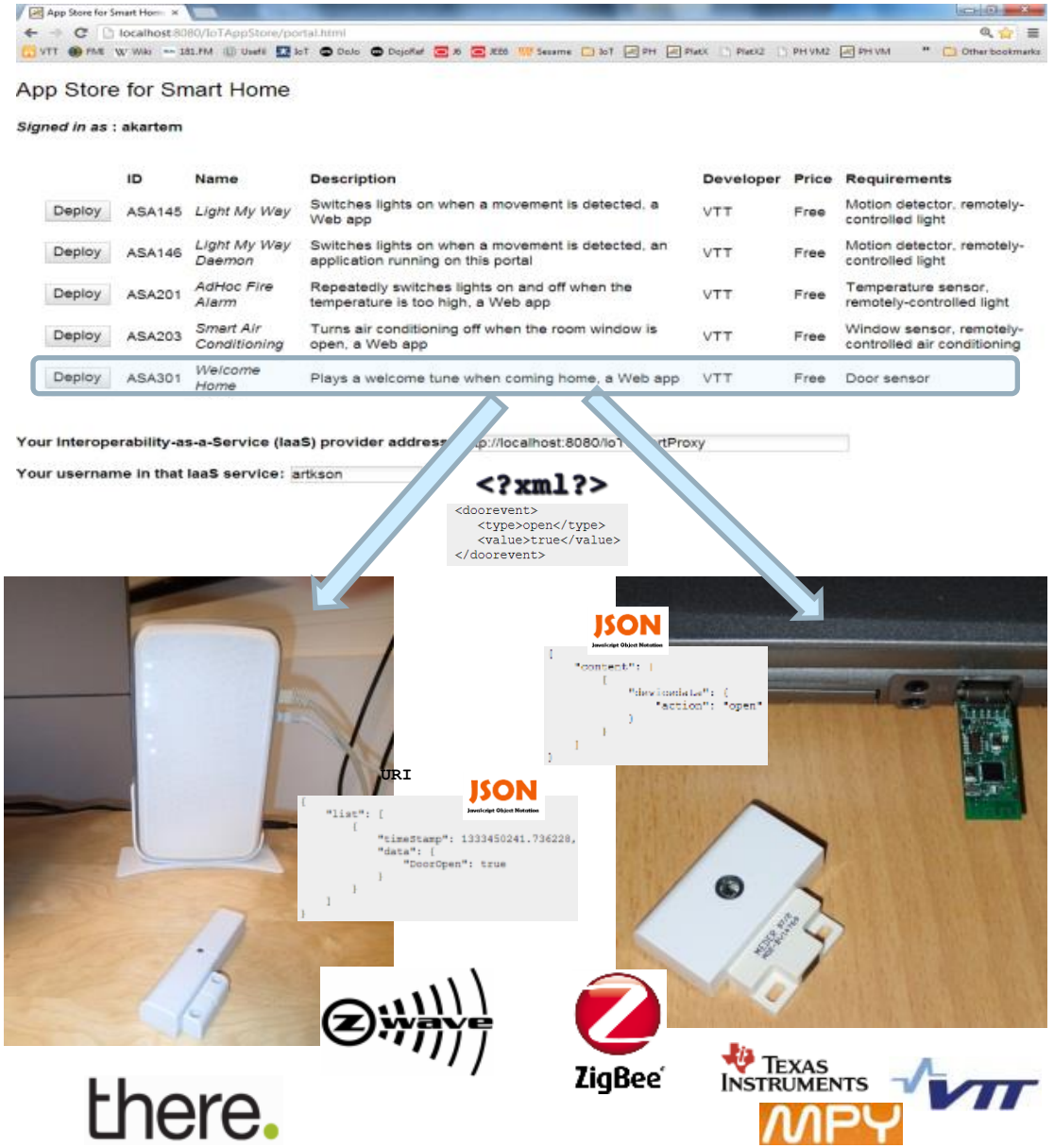
- *Ontology alignment*, or *ontology matching*, is the process of determining correspondences between concepts.

- Methods vary greatly and have varying precision (wrt. false positives) and recall (wrt. false negatives). These methods do some or all of:
  - Performing simple lexical analysis -> can map e.g. “Temperature” and “Temp”
  - Consulting existing data sources such as WordNet® -> can map e.g. “Temperature” and “Heat”
  - Apply translation services like Google Translate -> can map e.g. “Temperature” and “Lämpötila”
  - etc.

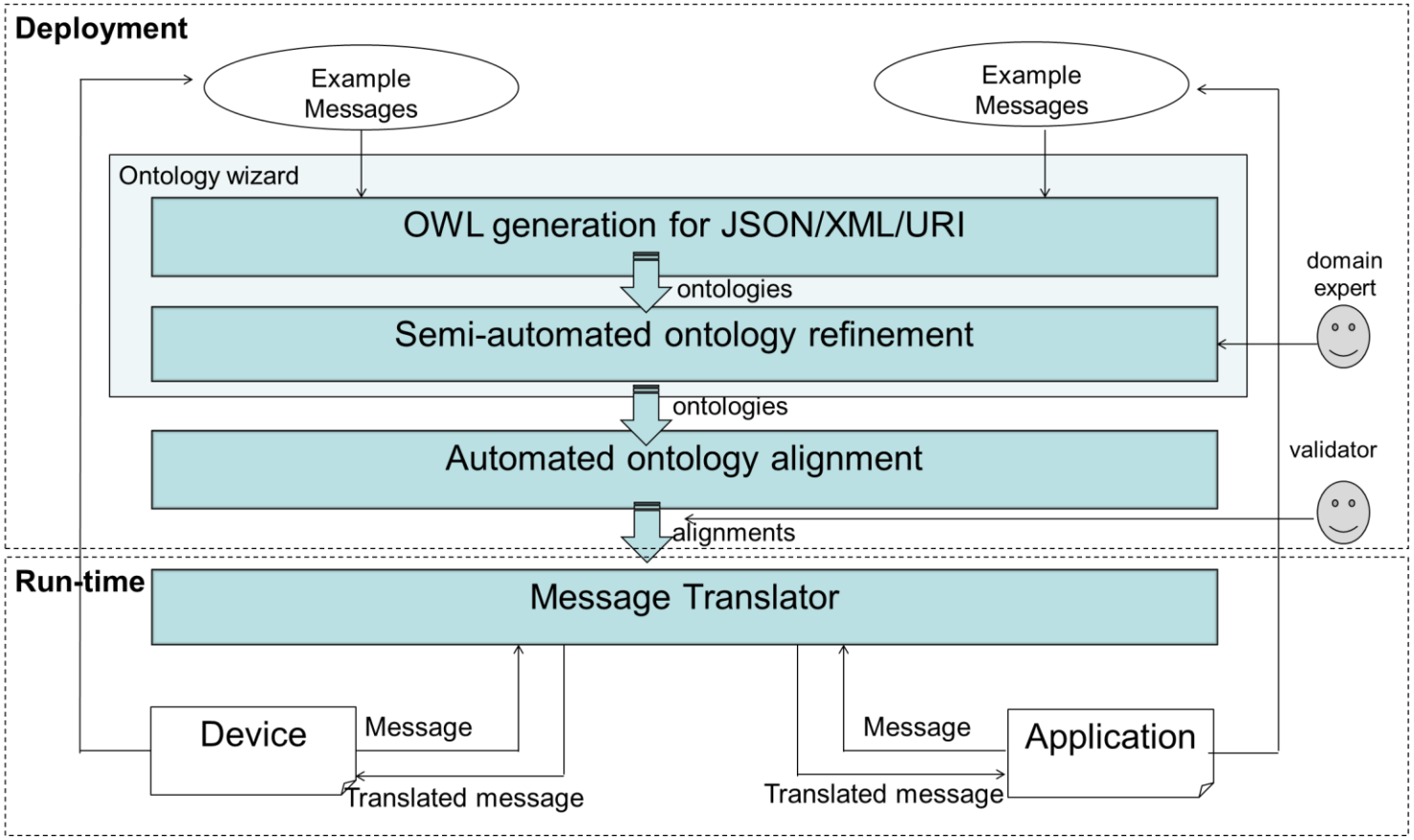
# Semantic Smart Gateway Framework (SSGF)




http://www.youtube.com/watch?v=R15Xnc2-Ovs



# Ontology Alignment in a smart gateway




Kotis K. and Katasonov A. (2013) *Semantic Interoperability on the Internet of Things: The Semantic Smart Gateway Framework*, International Journal of Distributed Systems and Technologies 4(3): 47-69

# Summary

- Our vision of interoperability on IoT is:
  - Ability to interconnect devices from different vendors.
  - Ability to have gradually growing IoT environments.
  - Ability of 3rd parties to develop software applications for IoT environments.
  - Ability to develop applications that are generic in the sense of running on various IoT device sets (different vendors, same purpose).

- We believe that:
  - Web of Things approach (RESTful HTTP) is the first enabler.
  - Semantic technologies is the second one.