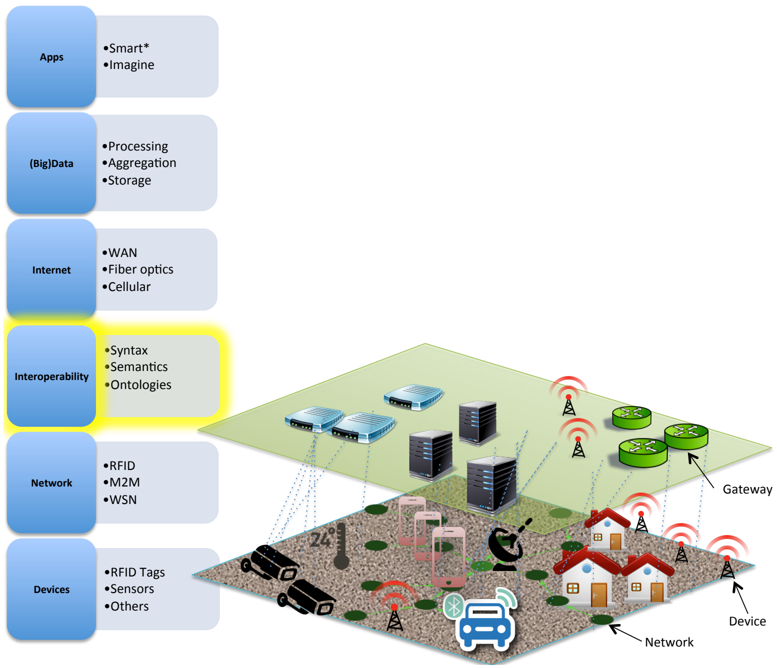
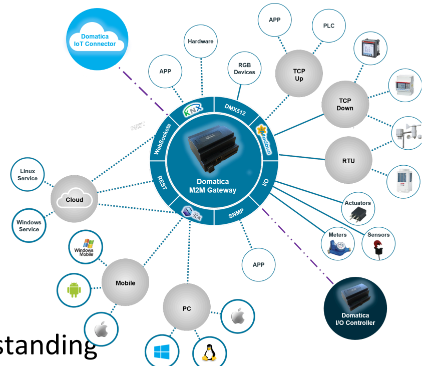
On the IoT, From A to Z

Part 2



Interoperability



- Why matters?
  - More systems → better understanding
  - Better understanding → Better decision making
  - Build smarter systems
- Towards Web of Things
  - Global access (availability)
  - Myriad of devices (scalability)
  - Heterogeneity
  - Sensed data is often redundant (reliability&Quality)

Interoperability: Syntax and Semantics

Fala inglês ou francês?

My rescue question.

Interoperability: Syntax and Semantics

- Devices speak different languages
  - Device-Device
- Humans also speak different language
  - Human-Device

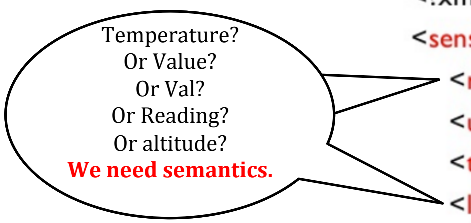
- Approach:
  - Syntax: data representation
  - Semantics: what the syntax means?

Interoperability: Syntax


```
<value>15</value>
<unit>C</unit>
<time>08:15</time>
<longitude>51.243057</longitude>
<latitude>-0.58944</latitude>
```

Interoperability: Syntax

- XML maybe ok for humans, what about devices?
- XML (or SensorML) and DTD specify the structure of a doc not the meaning.




```
<?xml version="1.0"?>
<sensor_data>
  <reading>15</reading>
  <u>C</u>
  <timestamp>08:15</timestamp>
  <long>51.243057</long>
  <lat>-0.58944</lat>
</sensor_data>
```

Interoperability: SSW

- SSW: Semantic Sensor Web

- Device and data representation

- Better autonomous interaction between devices and/or humans via meaningful relations between objects.

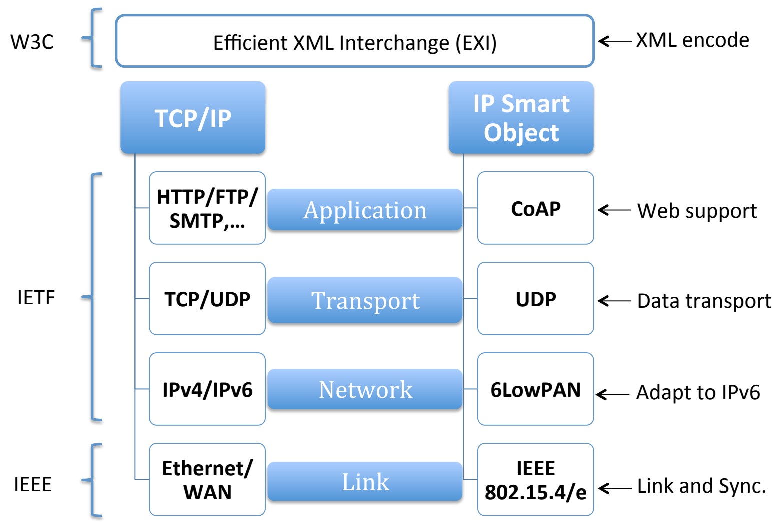
Interoperability: SSW (RDF)

- RDF: Resource Description Framework (W3C)
- Defines relations between documents via annotation
- Triplets to describe data and metadata as
  - Object -> Attribute-> Value
    - <“Sensor”, hasType, “Temperature”>
  - Object -> Property -> Subject
    - <“Sensor1”, hasValue, ”15”>
- Supports XML and Graph representations
- SPARQL is a query language for RDF

Interoperability: SSW (Ontologies)

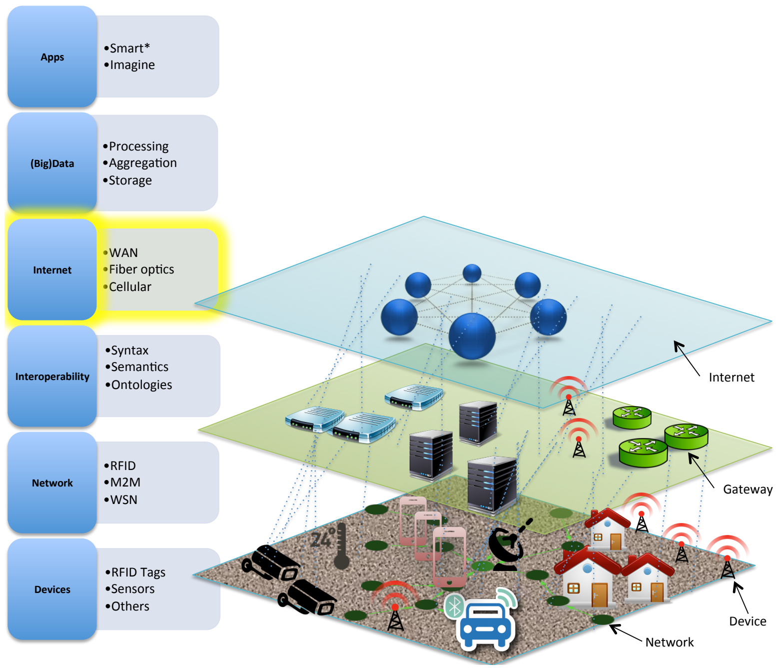
- Again
  - Temperature=Temp=Tmp=Val=Value?
- Ontologies: finite list of terms and their relations to describe a set of important concepts in a domain.
- OWL: Web Ontology Language
  - Similarity and difference in terms
  - Build classes not only names
- Annotate dynamically

Interoperability: Standards



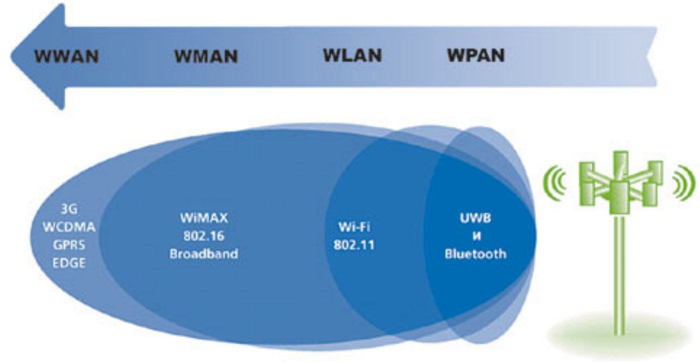
Interoperability: Middlewares

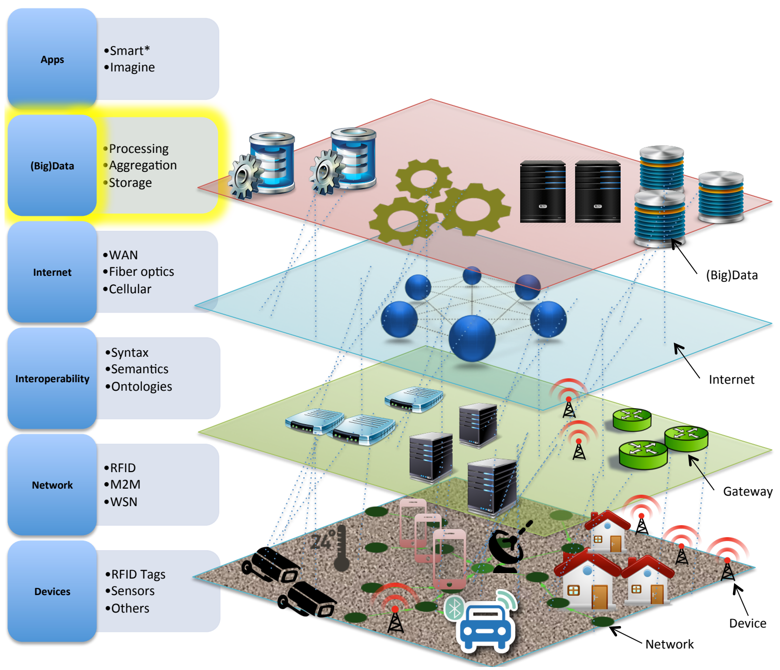
- Why?
  - Bring Plug&Play functionality
  - Support different components
  - Facilitate the interaction between the ‘Internet’ and the ‘Things’
- Some middlewares
  - WISeMid: Energy-aware
  - TinyDB: Transparent to devices
  - Hydra: Loosely coupled devices (SOAP-based API).
  - AURA: Easy configuration



Internet

- You know it..
  - ADSL? Fiber optics? 2G? 3G? 4G? 5G?
  - WAN? **W**ireless **A**rea **N**etwork





Data: Processing

Data is all what you want, right?

Not really,

we need **actionable-Information**

Data: Processing (Learning and Mining)

- To understand correlations and draw conclusions → help in decision making.
- All about knowledge discovery and prediction
- Approaches
  - Data mining, text mining, pattern recognition
  - Machine learning, multi-view learning, deep learning.

Data: Processing (Batch)

- Huge volumes of Data → Batch processing
  - Use large files to **reduce** costly disk **access**
    - Google File System (GFS)
    - Hadoop Distributed File System (HDFS)
  - Map&Reduce to **improve processing** speed
    - MAP: split data into thousands many parallel processes
    - Reduce: aggreagate the results
- Problem: significant delay (minutes, hours..)

Data: Processing (Stream)

- (Near) Real-time processing
- Complex Event Processing (CEP) OR Event Stream Processing (ESP)
- Approach:
  - Sliding window slices of stream
  - Process slices in parallel (MAP&Reduce)
  - E.g., Twitter Storm, Apache S4
- Not intended for analyzing full big data set

Data: Processing (Fast)

- Even faster processing?
- Parallelize everything
  - Coordination, query planning, optimization, scheduling, and execution
  - Google Dremel, Apache Drill
- Lambda Architecture
  - Stream processing and
  - Incremental analytics and
  - Indexing

Data: Aggregation

- Lots of data, do we need it?
  - Sensors in same area might give same info.

- Solution?
  - **Process** data **closer to the devices** (on gateways)
  - **Store** data **closer to devices** (RDF Triplestores)

- Aggregation approaches:
  - Data approximation (average, max, min..)
  - Other abstraction-based solutions......

Data: Aggregation (Challenges)

- Accuracy: lossy, lossless?
  - Spikes are sometimes important
  - Maybe use SAX (Symbolic Aggregate Approximation)
    - Transform time-series data into string representations
    - Better for machine learning tools
- Completeness: cover all sensors?
- Latency (computation, complexity)
- Dissemination (Pub/Sub)
  - At most once (OK for ambient sensing)
  - At least once (often)
  - Exactly once (occasionally)

Data: Storage

- Need more storage
  - Data centers and warehouses
- Need availability
  - Geo-replicated storage
    - Multiple (smaller) data centers
    - Clusters
    - Sharding
    - Bring data closer to clients
    - Allow stale reads
- But what about consistency?
  - CAP (Consistency/Availability/Partition-tolerance)
  - Consistency-Availability tradeoff

Data: Storage

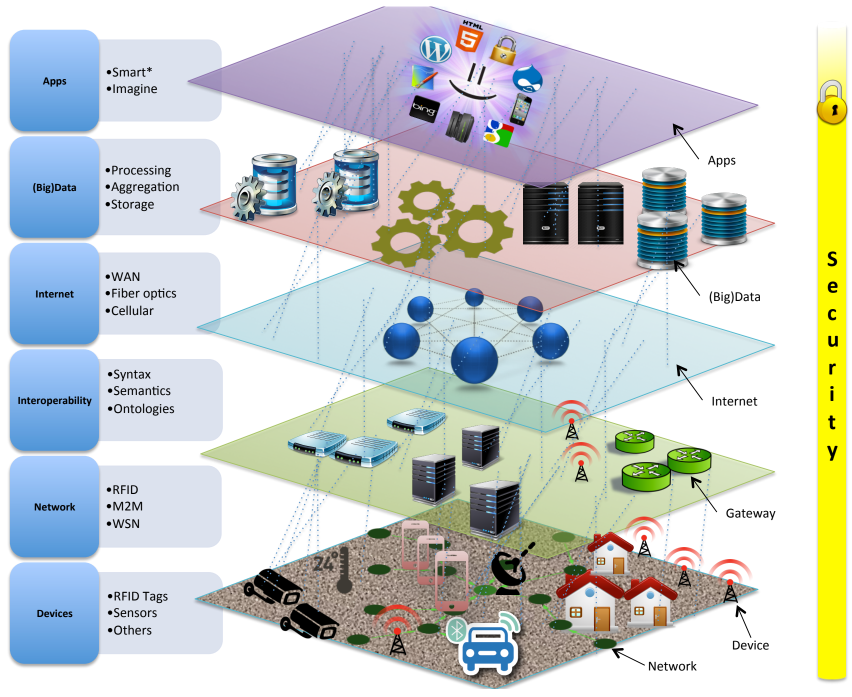
- Strong consistency (you know ACID, SQL?)
  - Not highly available
  - Not very elastic
  - Scales for few clusters only
  - Useful for sensitive services (banking..)

- Relaxed (eventual) consistency (you know BASE, NoSQL?)
  - Priority for availability over consistency
  - Allow stale Reads
  - Highly available and elastic
  - Needs conflict resolution (you know CRDTs?)

- Others
  - Column-based, Document-based, etc.

Data: Challenges

- IoT data is:
  - Massive
  - Dynamic
  - Multi-modal
  - Distributed
  - Discovery
  - Heterogeneous
  - Noisy
  - Cleansing
  - Incomplete
  - Context-Time-location dependent
  - Unreliable (crowdsourcing)
  - Requires real-time analysis
  - …



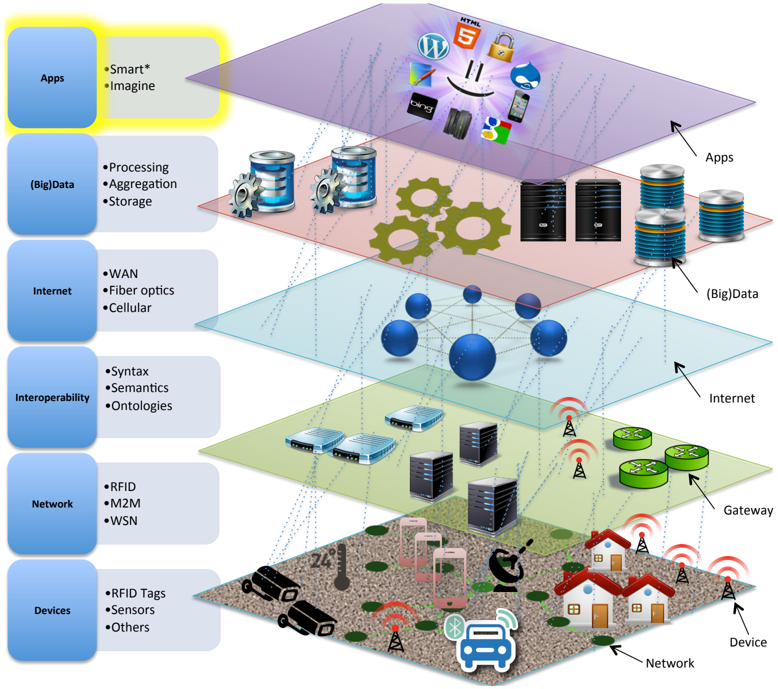
Security

- What to secure?
  - Devices?
  - Information?
    - Confidentiality
    - Integrity
    - Availability

Security (Challenges)

- "Things" inspect your life (Smart TV, Smartphone, cameras, consumption)
- Larger attack surface (points of vulnerability)
- More devices (50B objects by 2020)→ more data→ more attacks
- Cost: IoT is wildly developing and open
  - afford paying to secure everything?
- Complexity (lots of different devices and contexts).

- Solutions?
  - Security by design (OWASP recommendations).
  - And lots of more research…

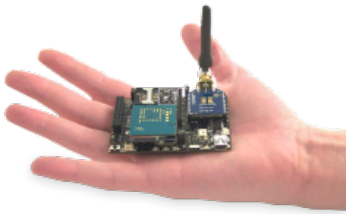


Applications

- Unlimited, depends on imagination.

- Smart[building, city, transport etc etc]
- Agriculture
- Space
- Military
- Disasters
- Environmental monitoring
- Structural Health Monitoring
- ...

Play with IoT at home



- Libelium Waspmote
  - IDE with code examples (C/C++)
  - Boards with USB and WIFI (~200€)
  - Many sensors (~ 20€)

- Viper (www.viperize.it)
  - IDE + VMachine + Mobile Apps
  - Code examples (Python)

- Arduino/Genuino
  - Another well-known open-source platform
  - See more www.arduino.cc

Interesting Research

- Security on all levels
- Energy on all levels
- Reliability of routing algorithms in WSN
- Data aggregation and edge computing
- Real-time data processing
- Semantics and ontologies.
  - How to find the best representation and ontology across heterogeneous (diff. standard) sensor networks and annotate dynamically.

Concluding Remarks

- Use:
  - Sensors are becoming cheap
  - Platforms are easy to use
- Pros
  - IoT makes life easier
  - IoT improves business
  - IoT makes science fiction reality
- Cons
  - IoT is wild
  - IoT is scary

Tons of research questions and challenges