Capturing the Structure of IoT Systems with Graph Databases

for open bidirectional multiscale data mediation

Gilles Privat
Dana Popovici

Orange Labs

Grenoble, France

orange™

Outline

What IoT is about

Data models for the Internet of Things

Role of IoT platforms for data abstraction and mediation

Capturing an IoT System as a graph

Using a graph database

Crawling a REST interface

Opening up IoT systems with RDF graphs & linked data

Hype-Style IoT : Connected devices

with SIM products

Usual players

Samsung, Sony, Nokia

New players

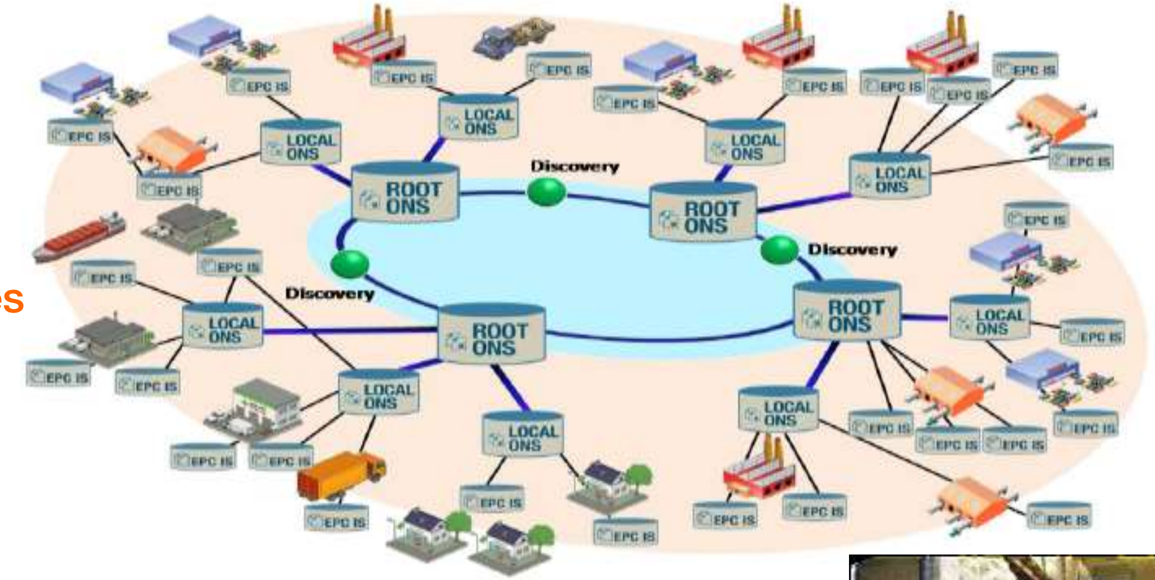
Filip, Linkoo, Tagg

with screen & apps ecosystem

Home products

Sensors products

Tag-style IoT



- Supply chain and inventory management as canonical applications
- RFID, but also optical codes
- Universal identification schemes

Telco-style IoT: M2M in lieu of H2H

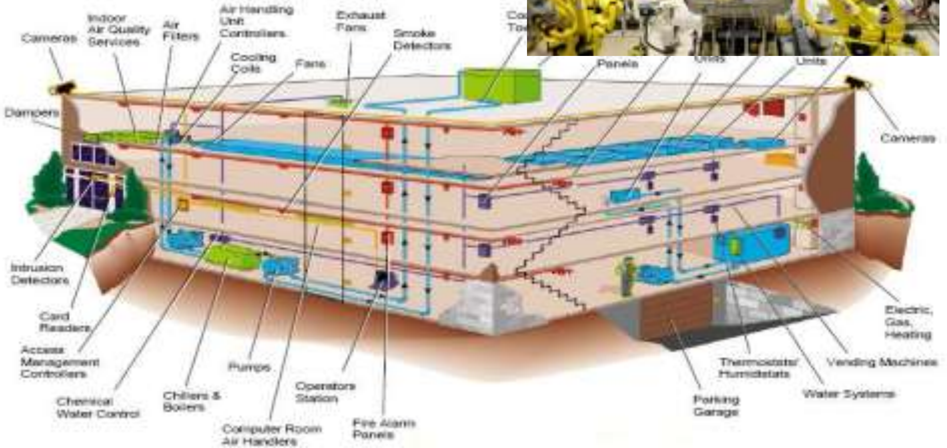
Devices with SIM cards

- forecast : >200 million active cellular M2M connections by 2014
- high-end sensors/actuators
- concentrators with “capillary” network links to low-end sensors

Up to 3G, cellular networks fit M2M requirements poorly

- energy constraints for battery-powered devices
- latency

Blue collar IoT



➔Domain-specific networks

- BACnet
- LonWorks
- X10
- CEBus
- CANbus
- emWare
- ECHONET
- CCN
- I2C
- Fieldbus

Data models for the IoT

Are not generic IT data models!   → they have to account for :

- the physical nature of things being described
- the use of low-level domain-specific protocols (e.g. CANbus or zwave)
    - which may enforce their own (often implicit) data models
- strict temporal constraints in the case of reactive systems :
    - determinacy
    - latency boundedness
    - reliability
    - concurrency

Yet have to draw upon generic IT data models in order to :

- use ascending levels of abstraction
- incorporate explicit domain knowlege
- model context data and integrate it with primary data
- get integrated into general purpose platforms
- interoperate through application-level « narrow waists »

Devices vs. Things

basic ICT devices
sensors & actuators

subsets of space
non-ICT physical entities
complex appliances
















The current IoT data morass

Data locked in silos

- most applications are vertically integrated
  - unimodal sensors dedicated to unimodal applications…
- many legacy systems (e.g. security) are non-connected or closed
- most IoT platforms are just message brokers!
  - no exploitation of message payload by the IoT platform itself (only by application)
  - no storage of permanent features of the environment →no Data Base
- new consumer-oriented « connected objects » each add their own silo!

Lack of metadata or rich data models

- no explicit type or structure

No shared environment models for applications that share same environment

- examples : smart homes, smart buildings, smart cities

-no exploitation of leveraging invariants from one environment instance to another

The neglected treasure of IoT data

Exploitation of sensor data confined within each silo for one application

– mostly one sensor modality used by each application

– low-level data (no high-level information) exploited, if at all

No cross-silo exploitation of data

– no high-level interpretation

Examples of cross-cutting exploitation of home data

– security sensors used for activity and presence detection

  → contextual adaptation of multimedia services

  → energy efficiency

Cloud-based post hoc analytics will not suffice to uncover this treasure

– sheer volume of raw unstructured data does not make up for lost structure in data sources

– has to be close to data sources (edge of cloud !) for real-time applications (involving control)

IoT data abstraction

Beyond device and protocol abstraction!

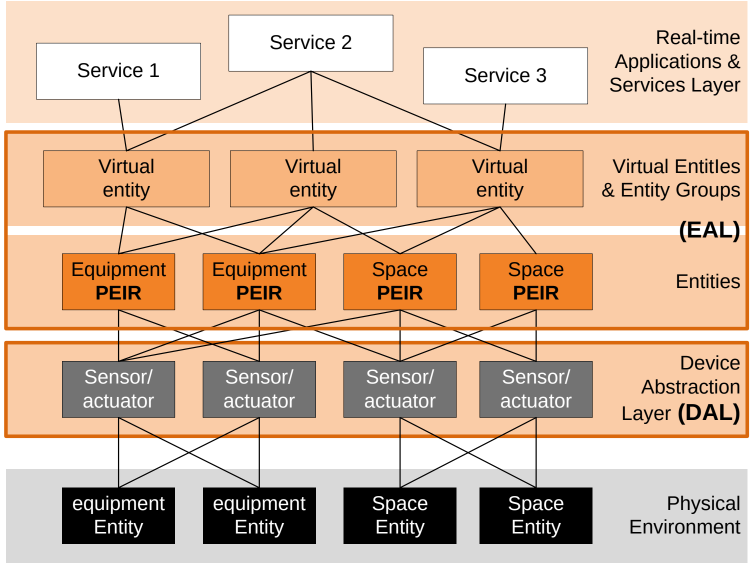
Capturing the invariants in home environment instances

Abstracting all relevant physical entities in the environment

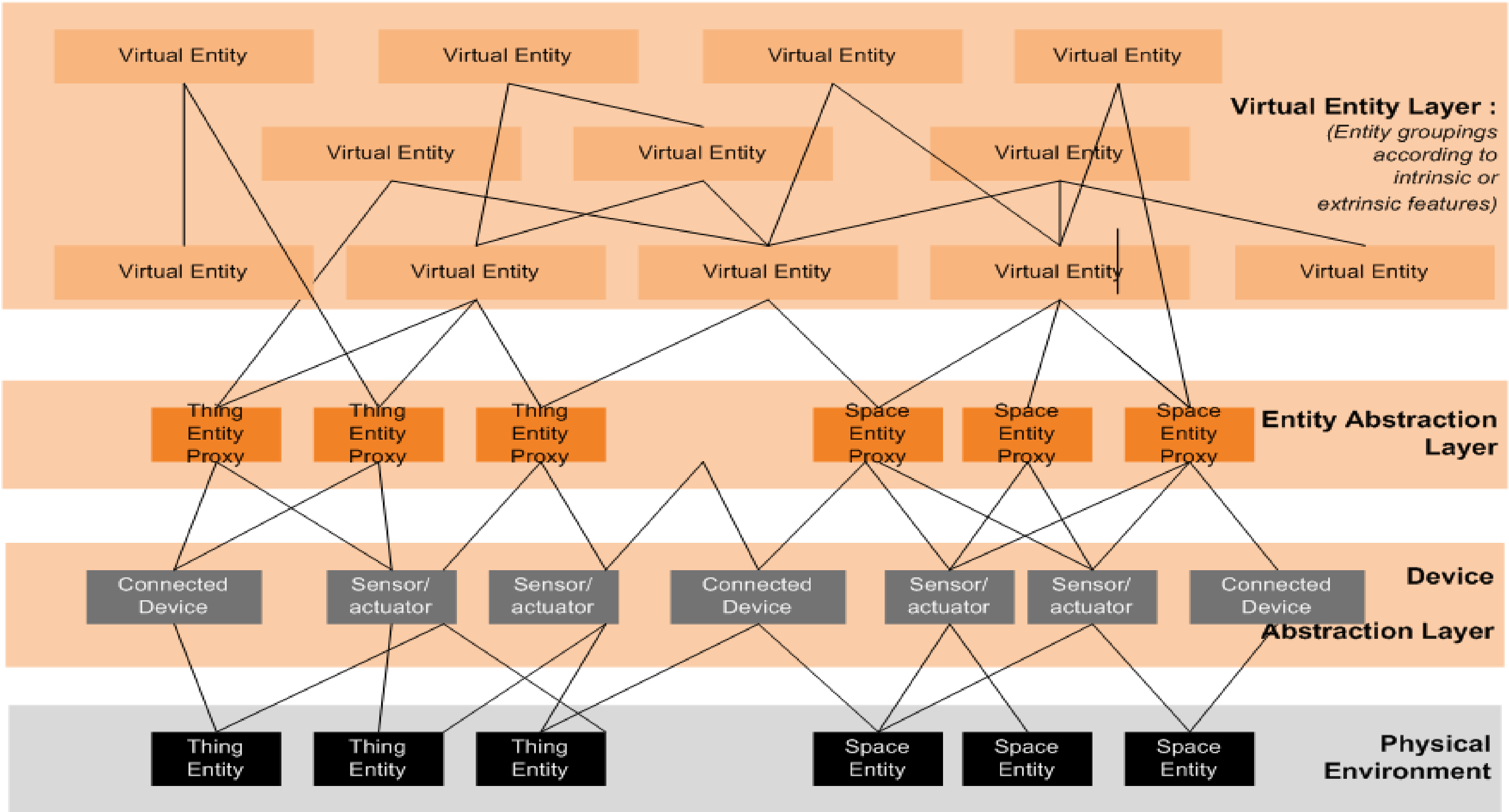
- rooms, places (→ akin to context entities in context middleware)
- non-connected appliances and legacy systems
- passive items

Providing higher layers of abstraction

- virtual entities based on properties and categories (intrinsic)
- entity & device instance groups (extrinsic and ad hoc)



IoT platform : data abstraction layers



Capturing an IoT System with a graph data base

Capturing invariants & relevant complexity of environments shared by # IoT applications

- e.g. smart home, smart building, smart city

Relationships graph are the key!

- Focus on domain-specific entities rather than devices
- Entity models (nodes of the graph) capture real-time behavior
- Directed links capture invariant (or slowly evolving) structure of target environment

Entity to entity & entity to device relationships

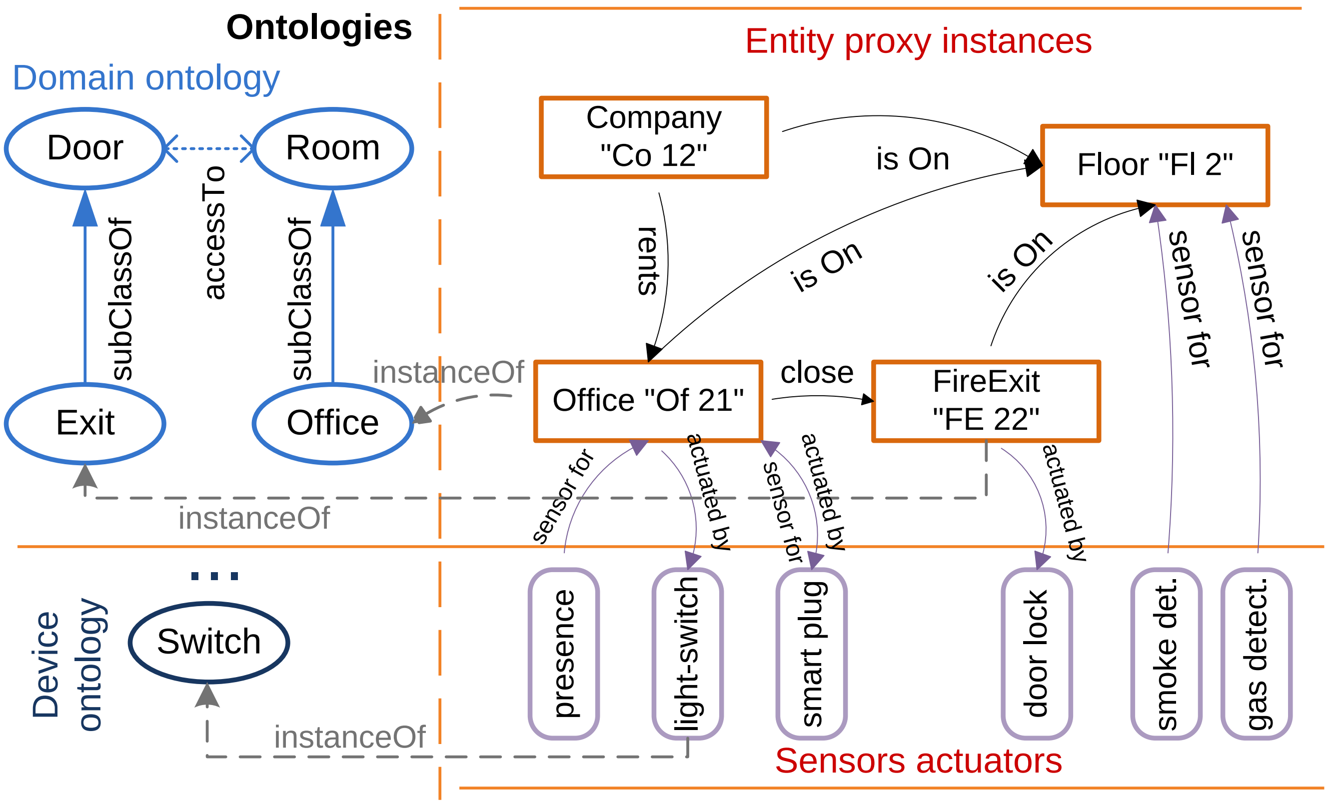
  - device used as primary or secondary sensor for an entity
  - device used as actuator for an entity
  - device acting upon an entity as a side effect
  - entity containing another, entity adjacent to another
  - device connected to another through the network

Entity to entity group relationships

Entity to category relationships

Capturing IoT data as a graph

example
smart building graph



Database solutions for IoT system representation

Object-oriented graph data base

Benefits

– Performance

– Scalability

– Tight coupling with IoT infrastructure

Limitations

– Centralization

– Limited openness

– Specific APIs and query languages

– No native reasoning tools

– No native integration of semantic modeling

RDF triplestore

Benefits

– Openness and integration with linked data

– Native standard semantic model (RDF, OWL)

– Reasoning tools

– Standard query language (SPARQL)

Limitations

– Partial centralization of triplestore

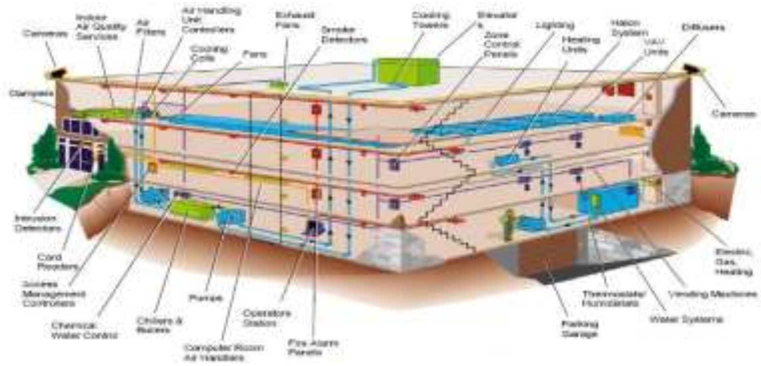
– Limited performance for real-time & reactive applications

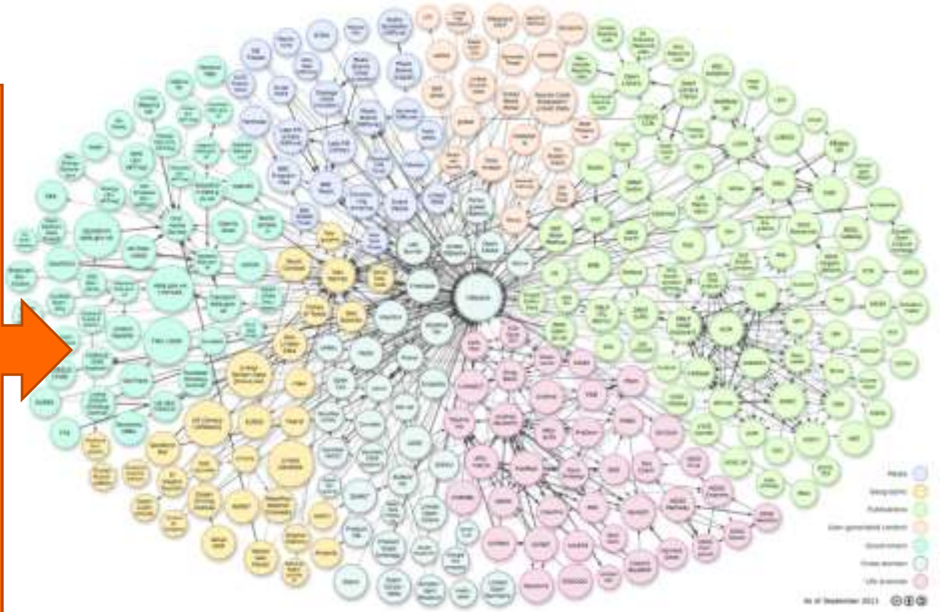
– Not tested for mission-critical and large-scale applications

Opening up the IoT to linked data

IoT systems no longer locked in silos, or isolated islands

They become part of the larger linked data archipelago





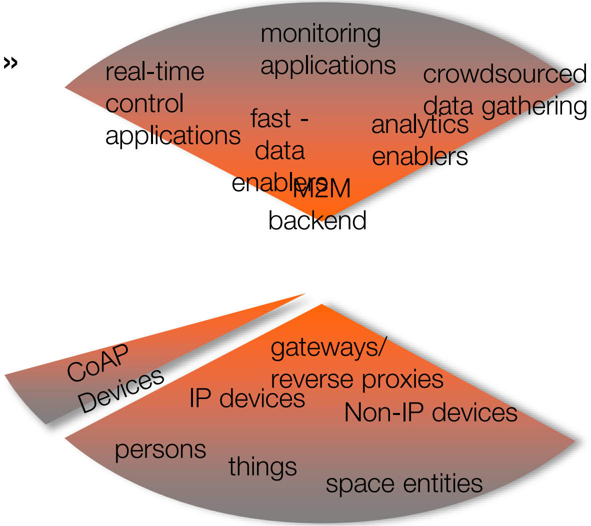
Linked data from the « web of things »

Narrow waist =REST identifiers shared by different infrastructures and abstraction layers

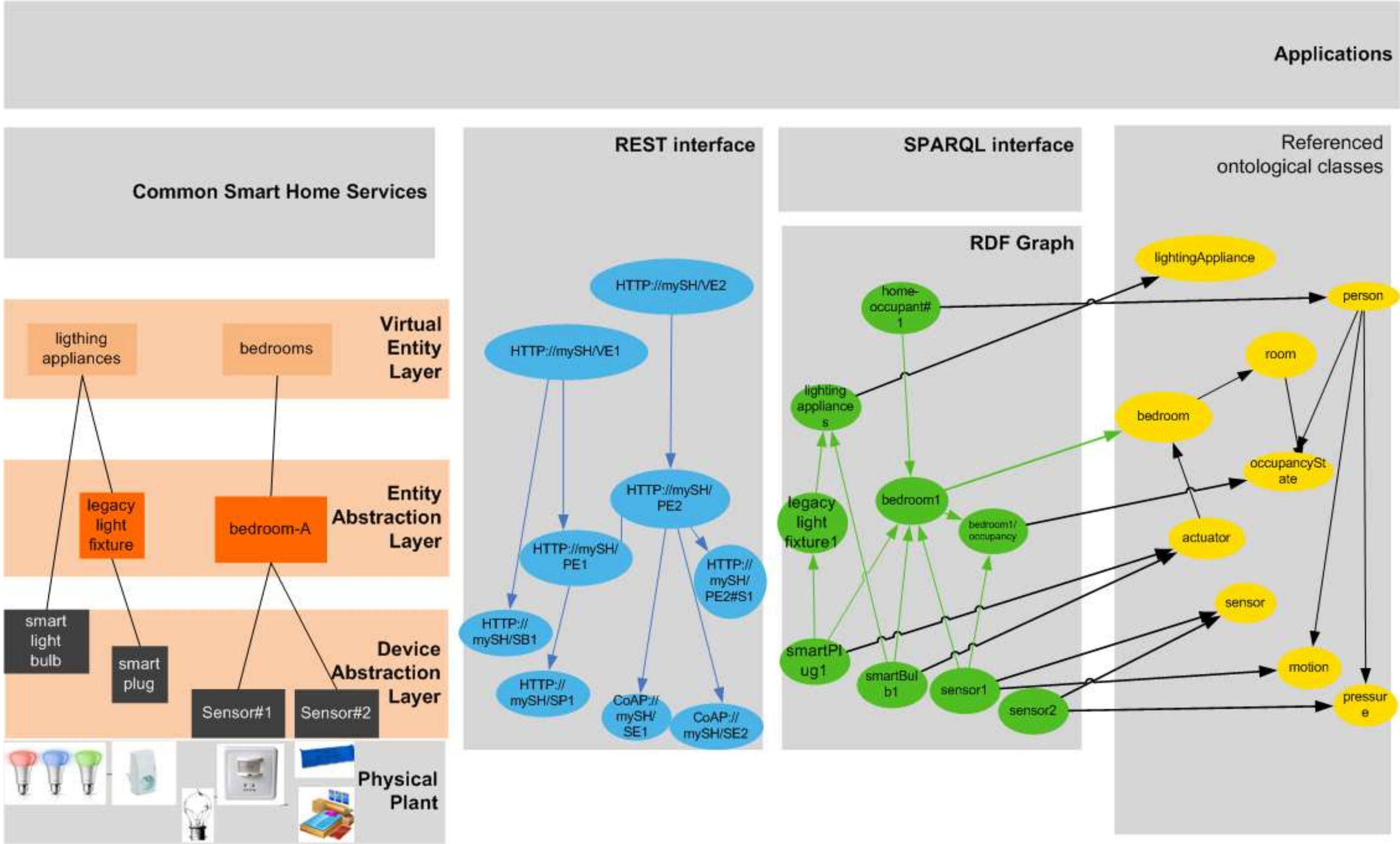
- entities are resources, states are subresources, instant values are representations
- devices are resources, reading from sensors and actuator controls are representations
- HTTP or CoAP URIs for all resources and subresources
- no hidden or implicit semantics (opaque URIs!)
- exclusively use hyperlinks for resource description → « follow your nose »
- no declarative descriptions à la WSDL!

IoT platform as presented

before is but one underlying ROA solution

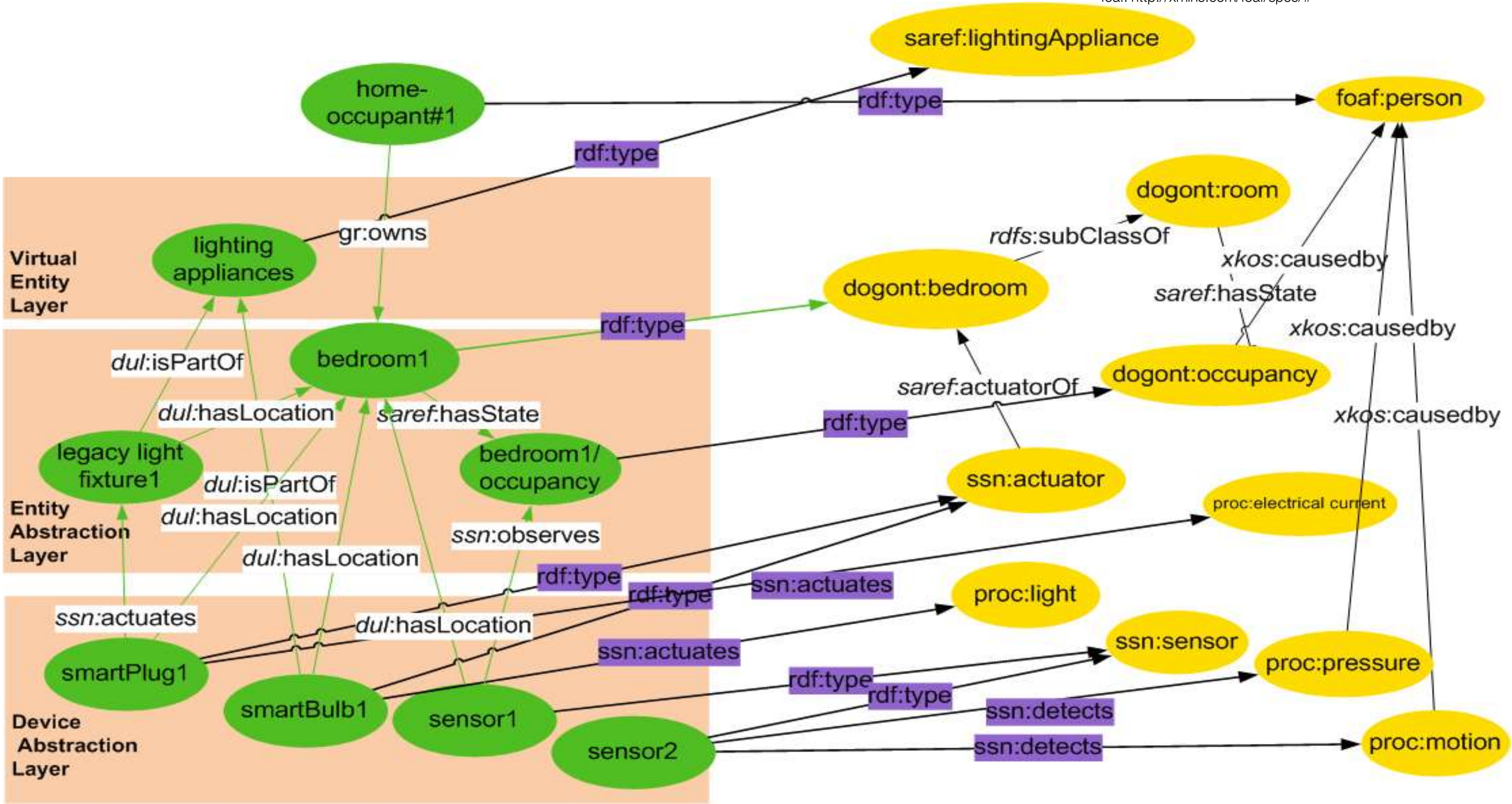


Example Smart home IoT infrastructure, linked up



Linking up IoT infrastructure : RDF graph as keystone

dogont: http://elite.polito.it/ontologies/dogont#
saref: http://ontology.tno.nl/saref#
ssn: http://purl.oclc.org/NET/ssnx/ssn#
dul: http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
gr: http://purl.org/goodrelations/v1#
sensor: http://mmisw.org/ont/univmemphis/sensor
xkos: http://rdf-vocabulary.ddialliance.org/xkos#
proc: http://sweet.jpl.nasa.gov/2.3/procPhysical.owl#
foaf: http://xmlns.com/foaf/spec/#



Quest for the IoT data grail…

Overcome the walled garden/fortress/silo mindset

Store permanent environment data in standards-based & open graph database

Be mindful of the pitfalls :

– preserve rights of legitimate stakeholders
– safeguard privacy
– ensure security (not from obscurity)!

Reap the many benefits of linked open data!