UNIK4750 - Measurable Security for the Internet of Things

L8 – Security Semantics

György Kálmán,
Mnemonic/CCIS/UNIK
gyorgy@unik.no

Josef Noll
UiO/UNIK
josef@unik.no

http://cwi.unik.no/wiki/UNIK4750, #IoTSec, #IoTSecNO

Overview

- Learning outcomes L8
- Recap: technology mapping
- Service requirements
  - ➡ Functional Requirements
  - ➡ Non-functional requirements
  - ➡ Security requirements
- Semantic technologies
  - ➡ why Semantics
  - ➡ elements of semantics
  - ➡ examples
- Security Ontologies
  - ➡ traditional view
  - ➡ Application-oriented view
- Map Security, Privacy, Dependability
- Conclusions

Expected Learning outcomes

Having followed the lecture, you can

- explain components of the Smart Grid (AMS) System of Systems
- can explain the difference between functional, non-functional and security components
- provide examples of security challenges in IoT

- explain the difference between the web, the semantic web, web services and semantic web services
- explain the core elements of the Semantic Web

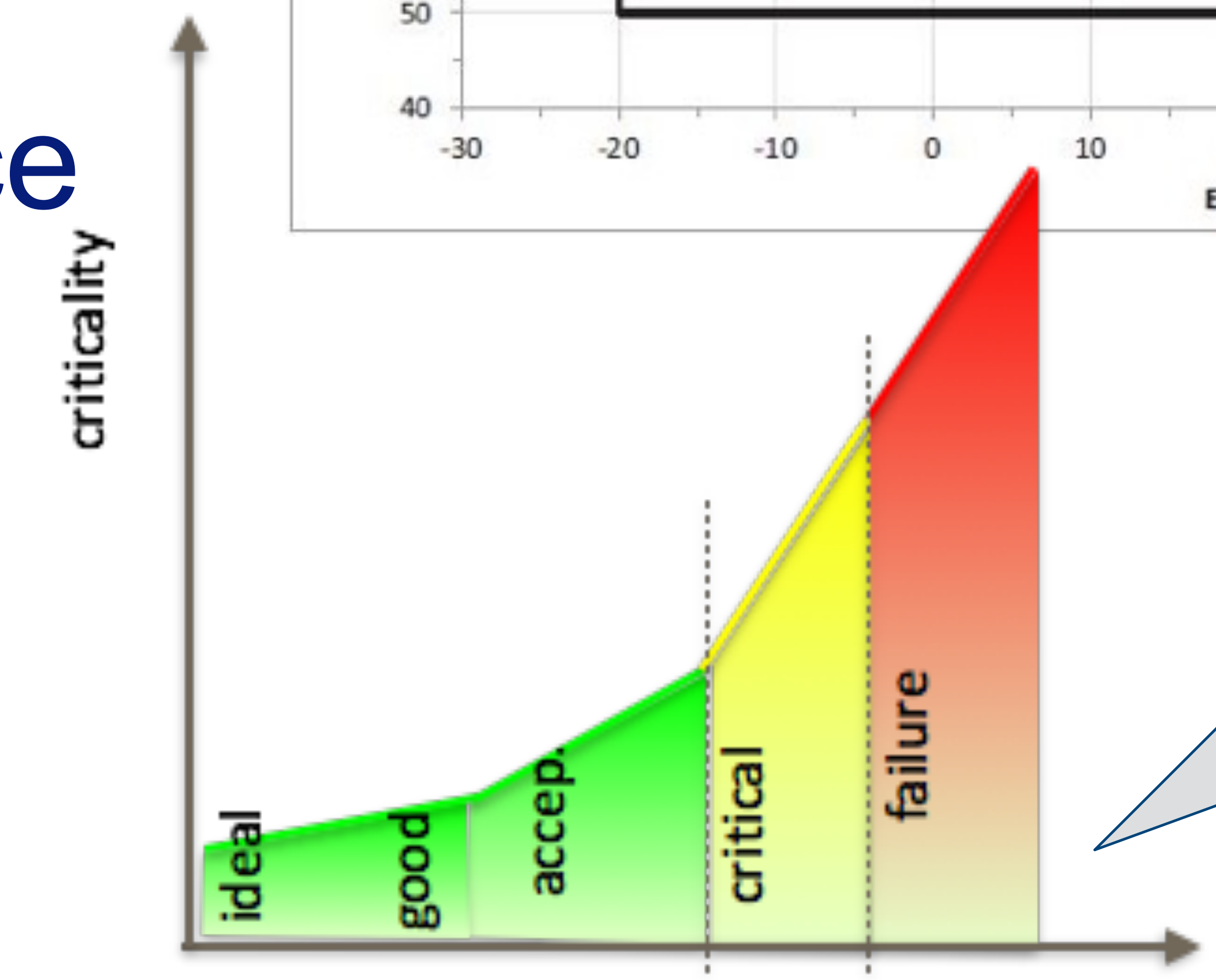
- apply semantics to IoT systems
- provide an example of attribute based access control

- discuss the shortcomings of the traditional threat-based approach
- list the main elements of the semantic descriptions of s,p,d functionalities
- perform a semantic mapping of s,p,d attributes

Service Requirements
- Functional Requirements
- Non-functional requirements
- Security requirements

Recap:
Conversion and operating envelope

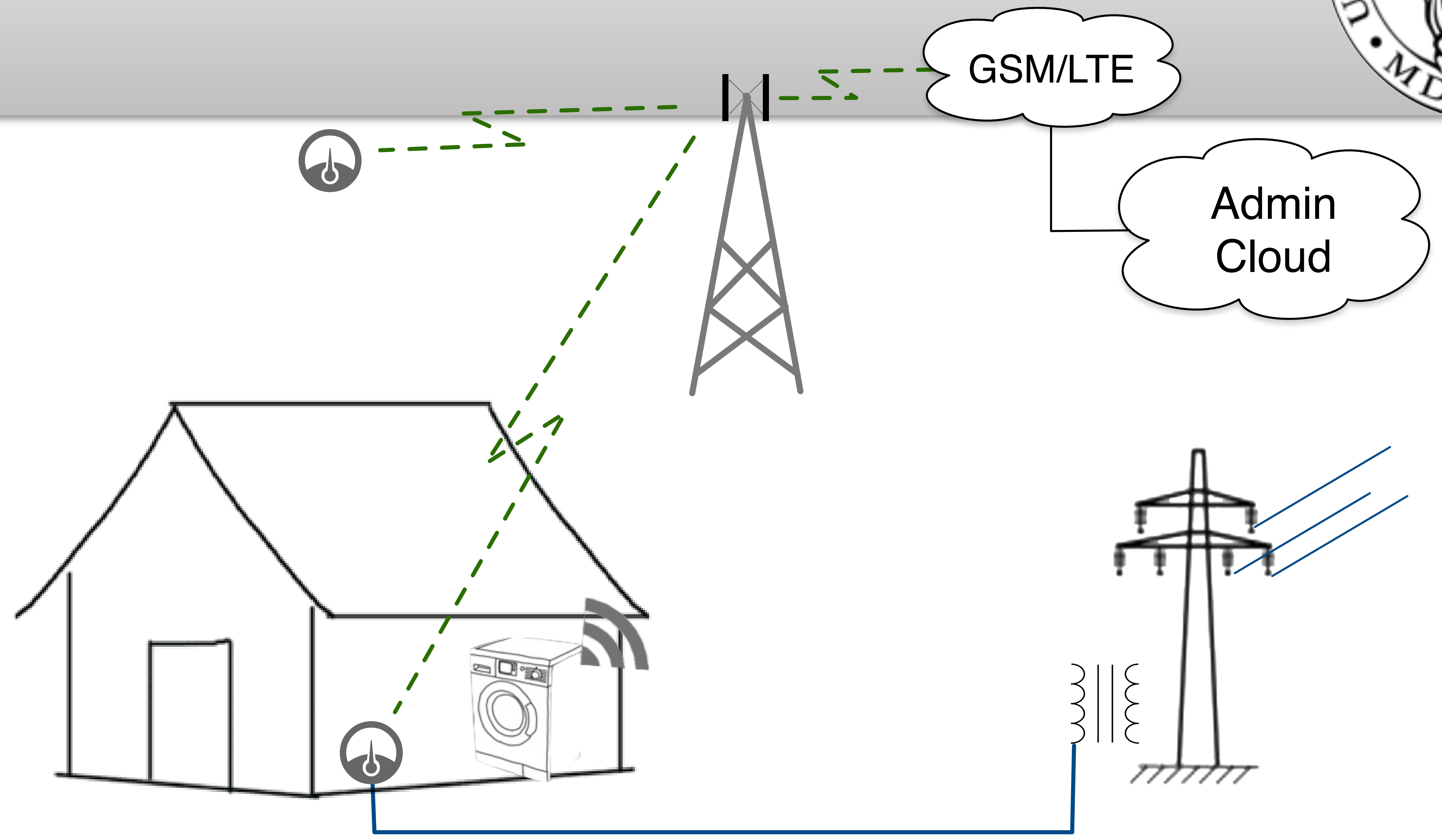
- Operating envelope: the operational parameters where our network can work “well”, depends on the technology and on the task
- For traffic estimation we need it in ”communication” QoS
  - ➡ Bandwidth, delay, jitter, (redundancy)
- Often can be done with simple arithmetic with a certain confidence level

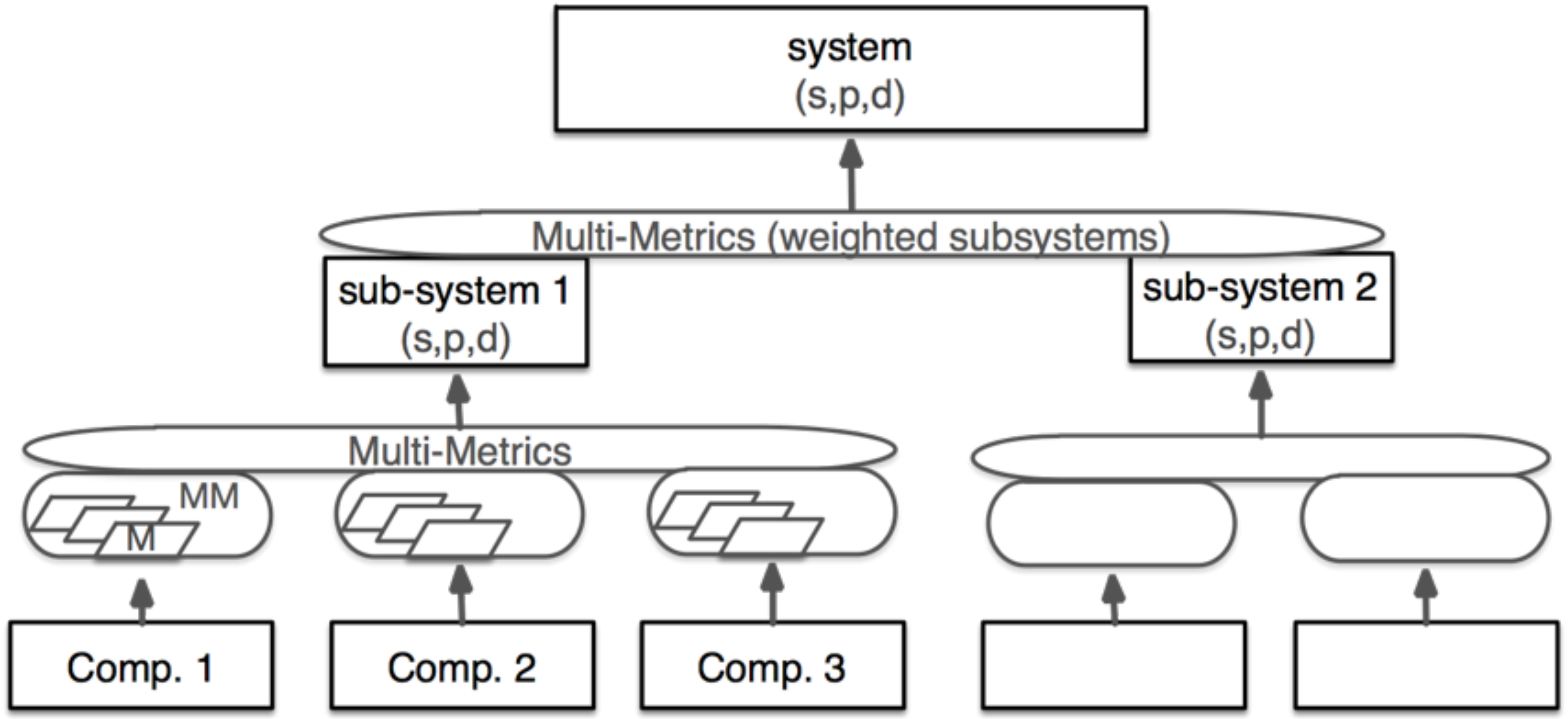


1) How does he Operating Envelope look like applying criticality?
2) How can the criticality be applied for SPD?

Example:
System of Systems

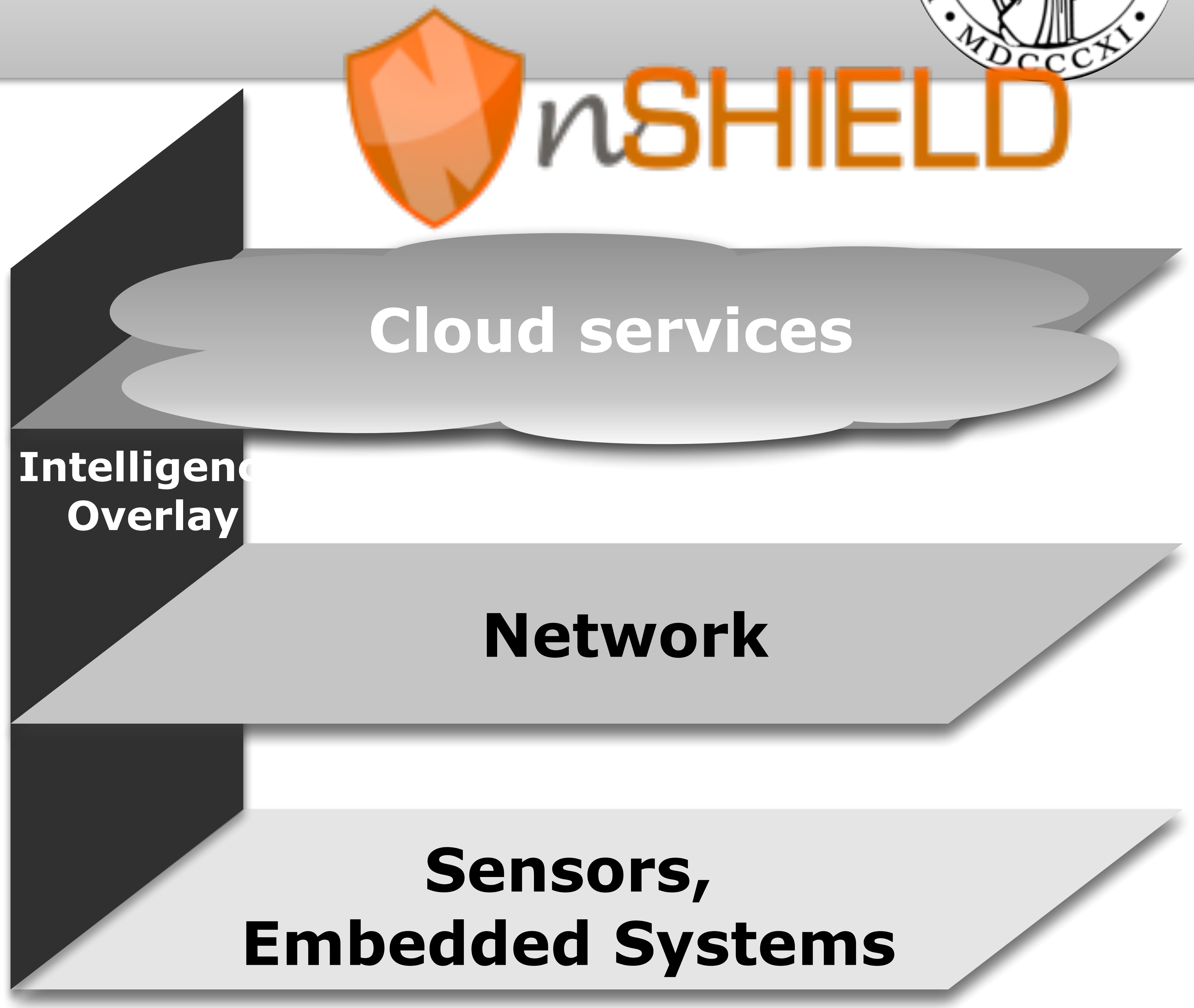
- A system consists of sub-systems
  - ➡ Example: Automatic Meter System (AMS) consists of reader (AMR), aggregator, communications, storage, user access
- A sub-system consists of sub-…-system
  - ➡ Example: AMR consists of power monitor, processing unit, communication unit
- A sub-....-system consists of components
  - ➡ Ex: AMR communication contains of a baseband processing, antenna, wireless link
- Components have parameters
  - ➡ Wireless link component: f=868 MHz, output power=?, Encryption=?

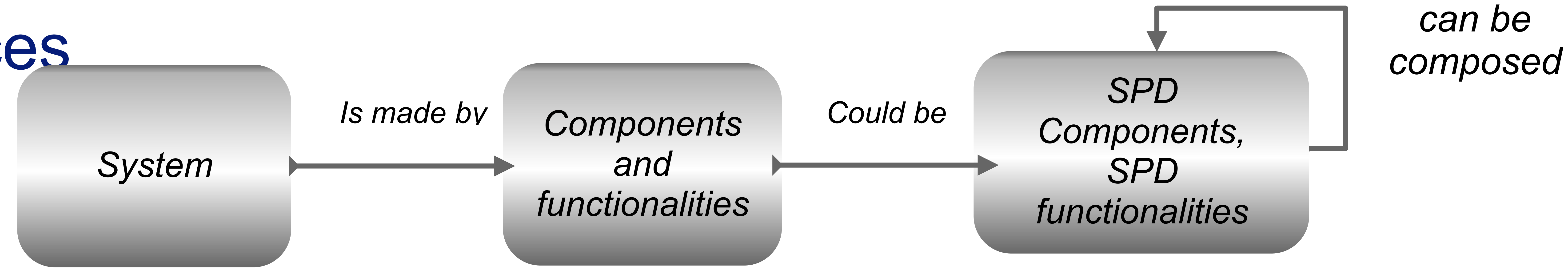




newSHIELD.eu approach

- Security approach by JU Artemis
  - ➡ Industry, National and EU supported (JU) activities
  - ➡ special focus on sensor systems
- Security, here
  - ➡ security (S)
  - ➡ privacy (P)
  - ➡ dependability (D)
- across the value chain
  - ➡ from sensors to services
- measurable security





Examples of
Security challenges in the IoT

- **System**: Intrusion awareness, fault-tolerance, data redundancy and diversity
- **Platform**: Auto start up on power failure, Auto reconfigurable on software failure, Auto synchronization on software failure, End-to-end secure communication, Mal-user detection, Access control for accessing sensor data
- **Middleware**: SPD Audit, Cryptographic Support, Identification and Authentication, Protection of the SPD functionalities, Security Management
- **Hardware**: SPD metrics, Self−recovery from hardware transient faults (through fault−injection), Auto−reconfiguration, Data encryption, Provision of security and privacy services, data encryption/decryption
- **Radio**: Threats tolerant transmission

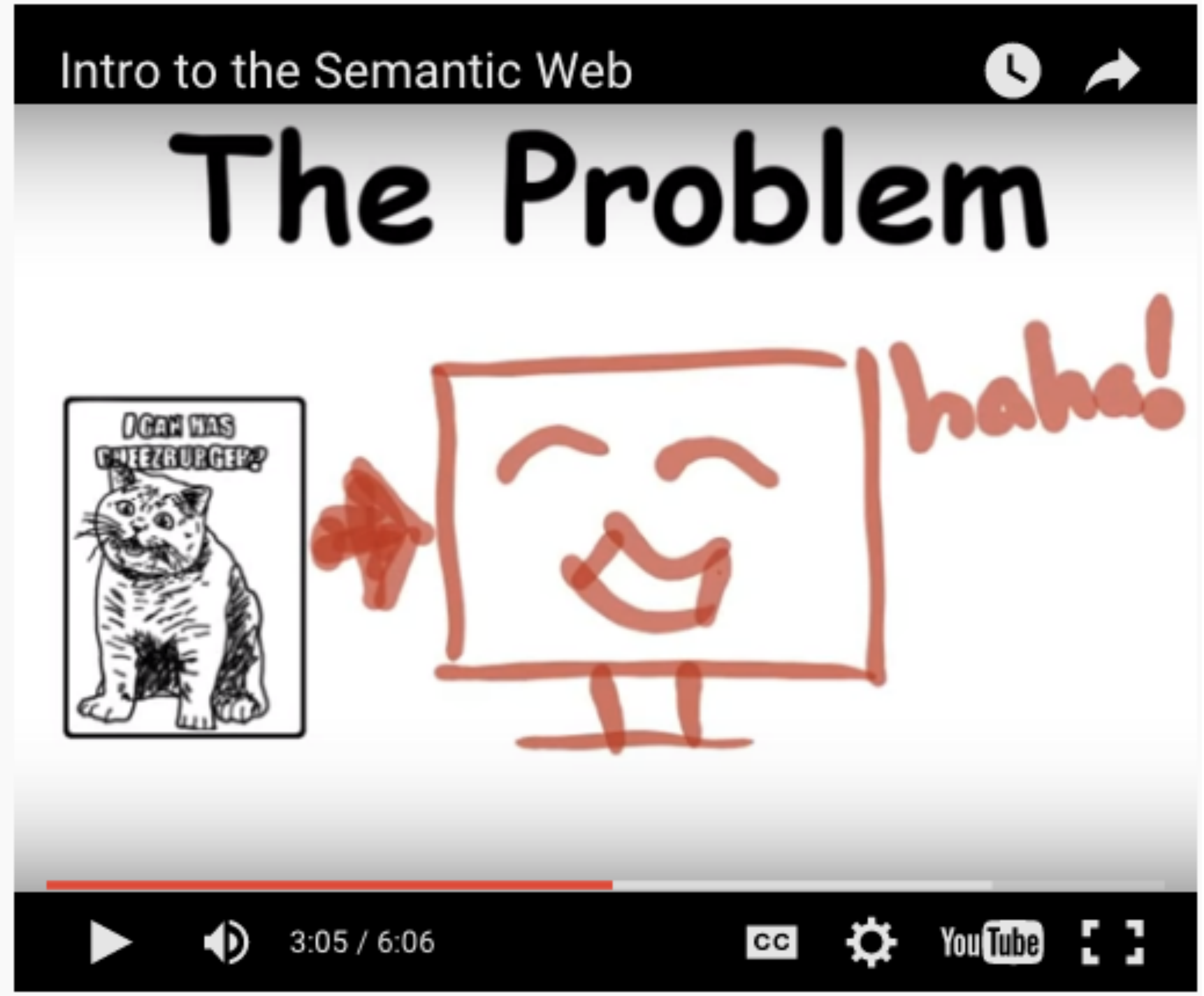
System components
classified after objective

- Functional components
  - ➡ input component (sensors, keyboard, mouse,..)
  - ➡ output component (alarm, screen, actuator,..)
  - ➡ processing component
  - ➡ Storing component (data base, files,)
  - ➡ Connection (wireless connection, wired connection)

- Security, Privacy, Dependability (SPD) components:
  - ➡ Encryption: Encryption algorithm, keys,..
  - ➡ Protocols
  - ➡ Authentication(mechanism (fingerprint, password, password complexity,.….) .
  - ➡ Authorization (privileges, ..)

- Management components (OS, Web server, data server)
- Human component (admin, user, ..).
- Physical component, car being a component in a car factory. (if treated as “sub-system)

Semantic technologies
- why Semantics
- elements of semantics
- examples



The Semantic Dimension of
the Internet of Things (IoT)

Source: L. Atzori et al., The Internet of Things: A survey, Comput. Netw. (2010), doi: 10.1016/ j.comnet.2010.05.010

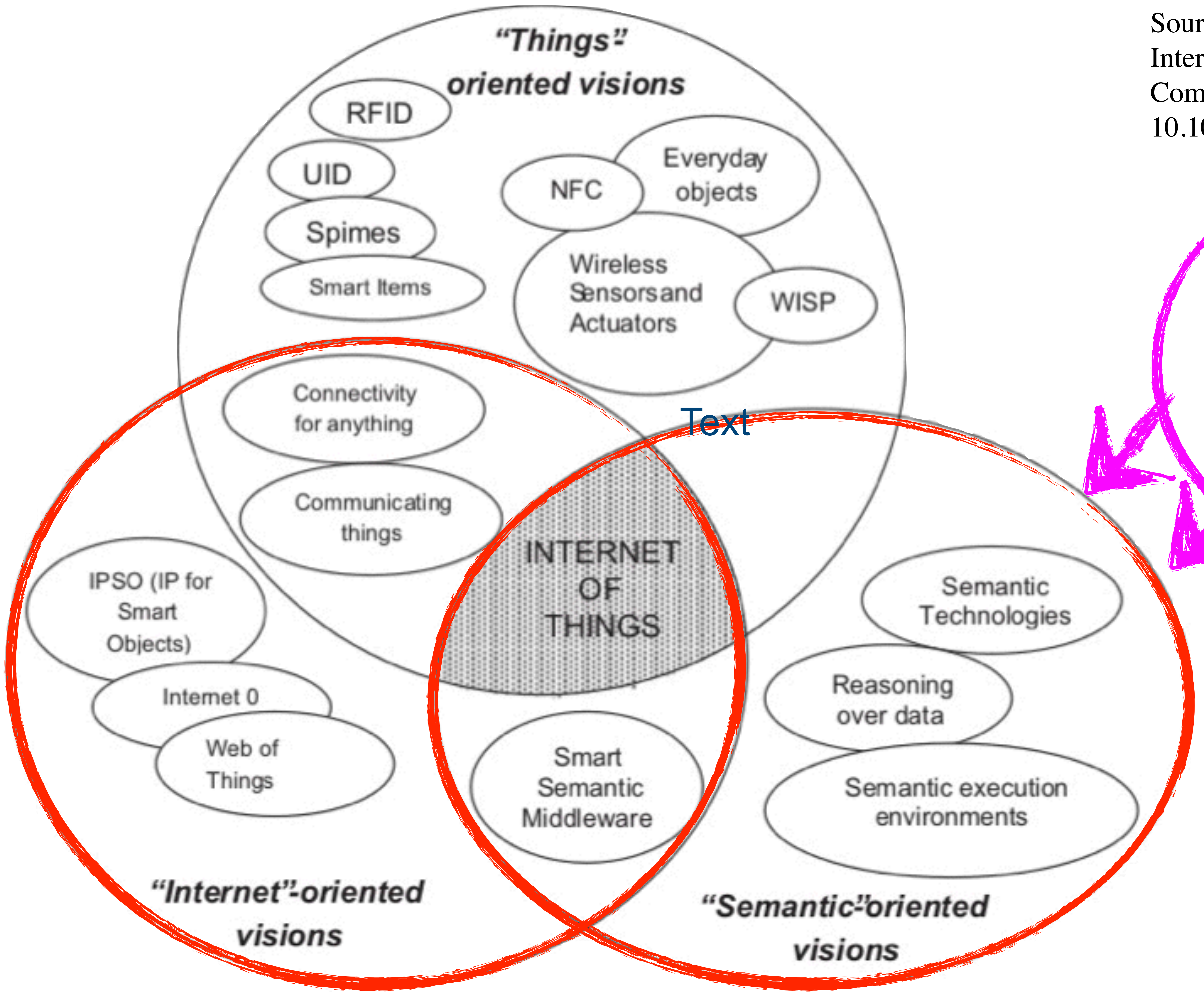
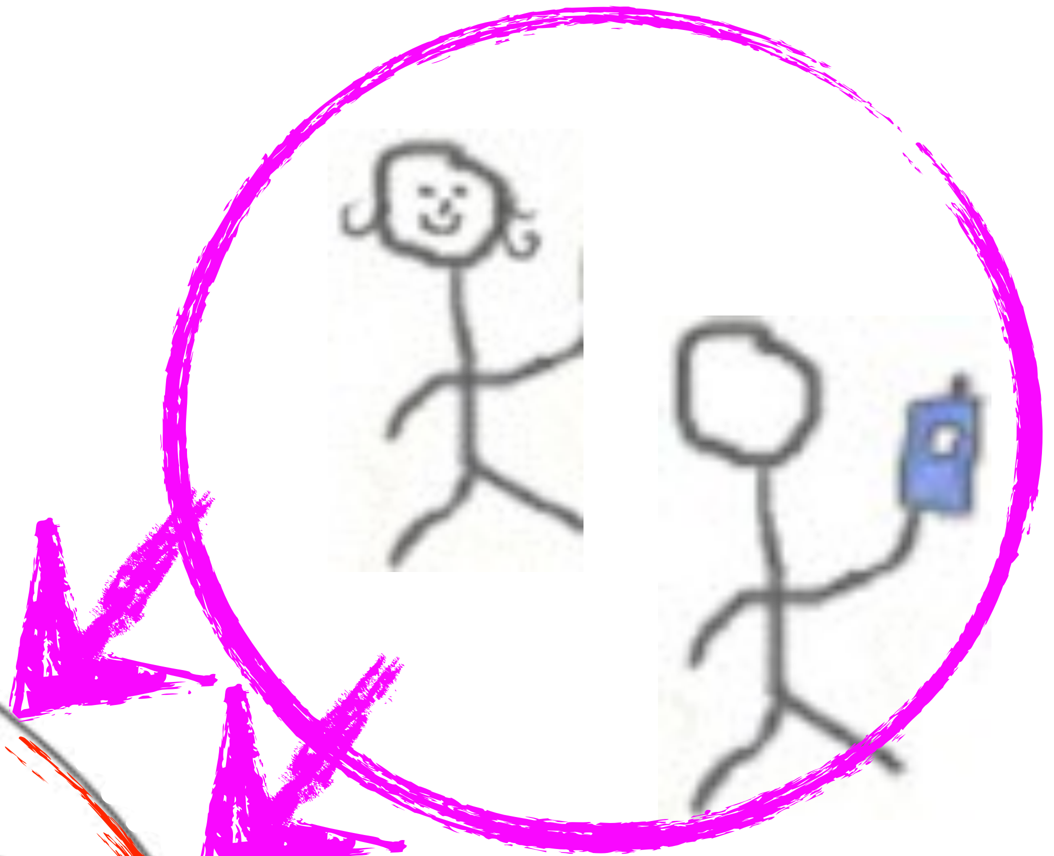


Fig. 1. "Internet of Things" paradigm as a result of the convergence of different visions.

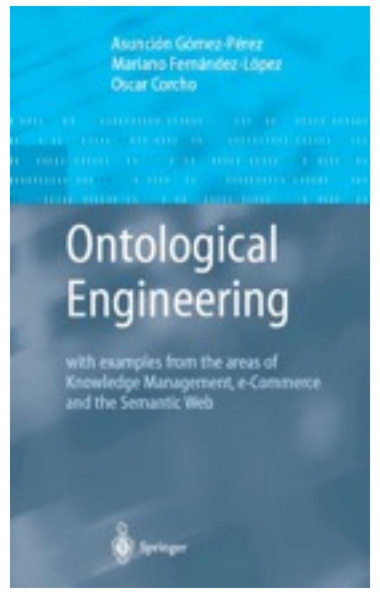


* security
* privacy
* dependability
  - context
  - content
* personalised

Why Semantics?

- Syntax vs. Semantics

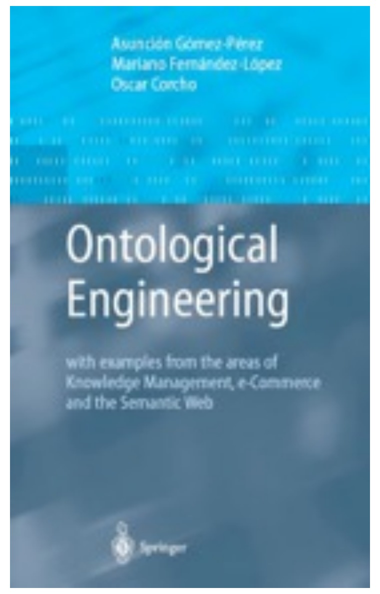
Arab



الاسم: الهندسة فىعلم التطو ر
المؤلّفون: آسنسيون غومزبرز
السّعر: $74.95
المنتج: الكتاب

```
<الاسم>الهندسة فىعلم التطو ر</الاسم>
<المؤلّفون>آسنسيون غومزبرز</المؤلّفون>
<السّعر>74.95$</السّعر>
<المنتج>الكتاب</المنتج>
```

English



Title: Ontological Engineering
Authors: Asunción Gómez-Pérez...
Price: $74.95
Product: Book

```
<Title>Ontological Engineering</Title>
<Author>Asunción Gómez-Pérez...</Author>
<Price>$74.95</Price>
<Product>Book</Product>
```

What do the tags mean for the machine?

Source: Juan Miguel Gomez, University Carlos III de Madrid

Why Semantics?

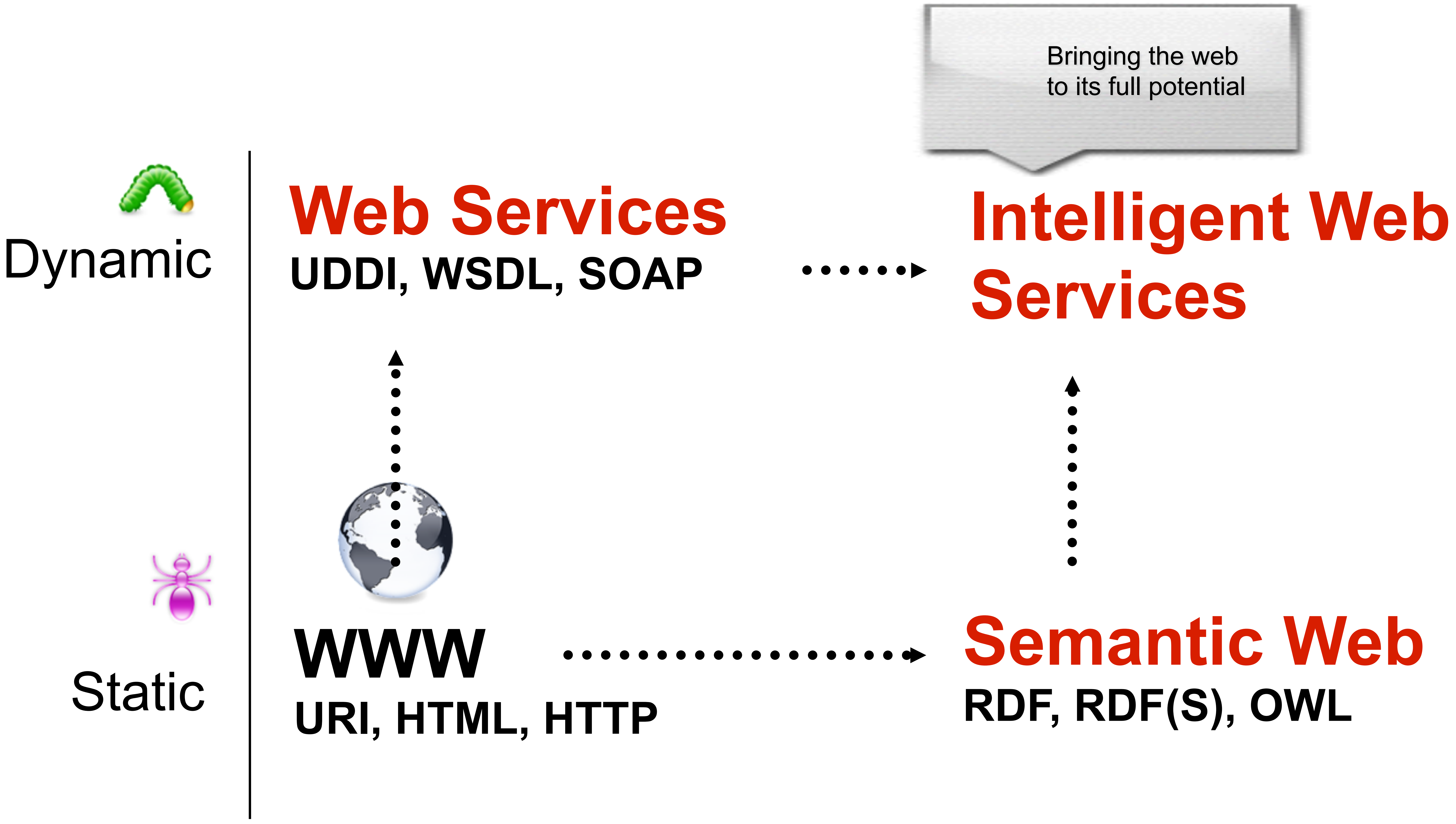
- Conceptual Level

lunch (.no)

lunch (.es)

Source: Juan Miguel Gomez, University Carlos III de Madrid

Semantic Web Services



Source: Juan Miguel Gomez, University Carlos III de Madrid

Requirements for Service Evolution

Web services

- Fixed service set, Static service composition, Low degree of automation
- Poor reliability
- Fixed Service Level Agreement

Semantic Web Services

- Flexible services, easy new services
- Alternative service provision
- Global, dynamic services

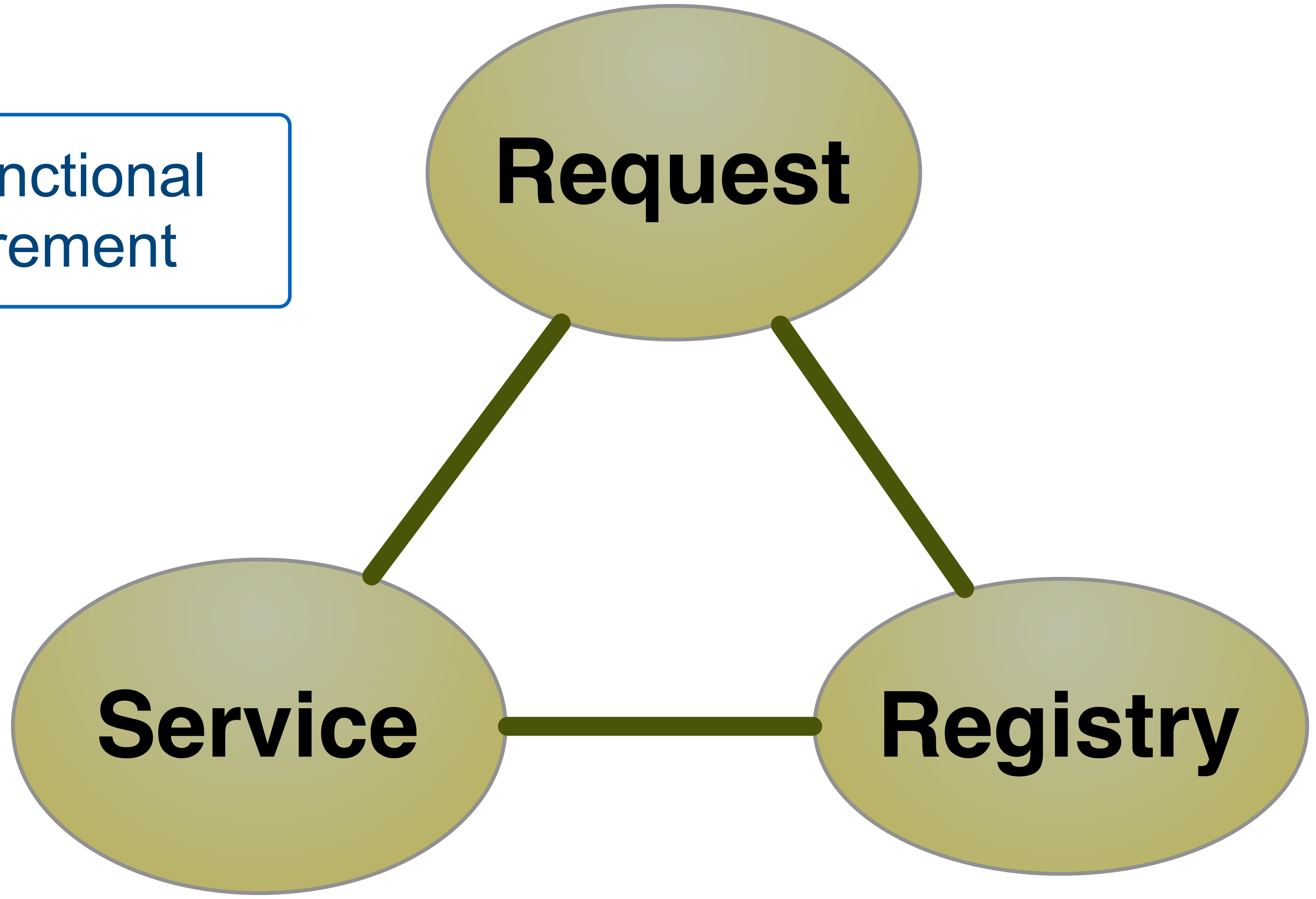
Elements of Web Services

- Service Request
  - want to come to Barcelona University
- Services
  - buy a flight ticket (cheap, direct, …)
  - buy a metro/bus ticket
- Service registry
  - link to ticket ordering at norwegian.no
- (Security) - Privacy attribute
  - only use company which does not sell my data

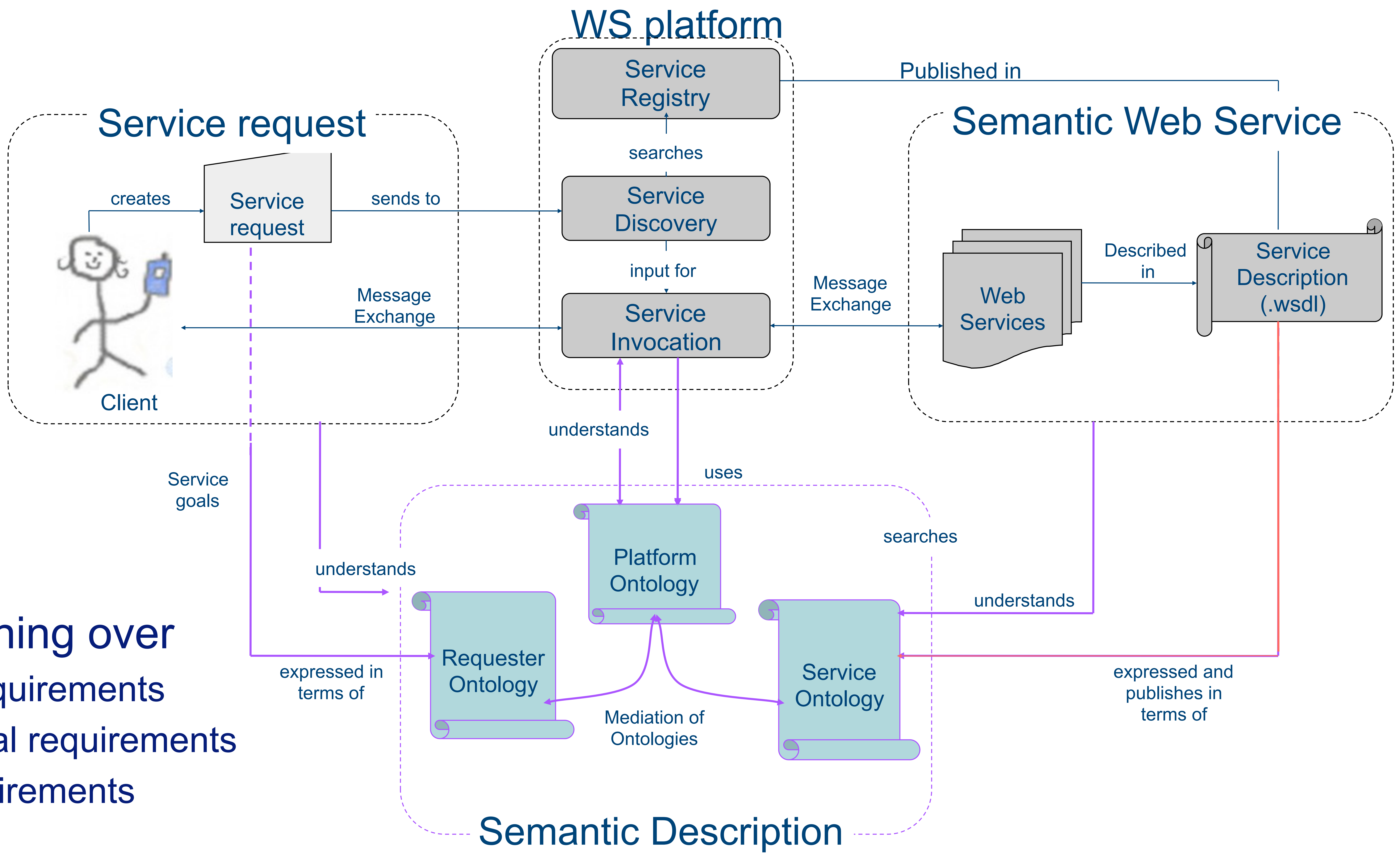
functional requirement

non-functional requirement

s,p,d requirement



Semantic Web Services Architecture



- Goal: Reasoning over
  - ➡ functional requirements
  - ➡ non-functional requirements
  - ➡ security requirements

Elements in Semantic Technologies

[Source: Wikipedia]

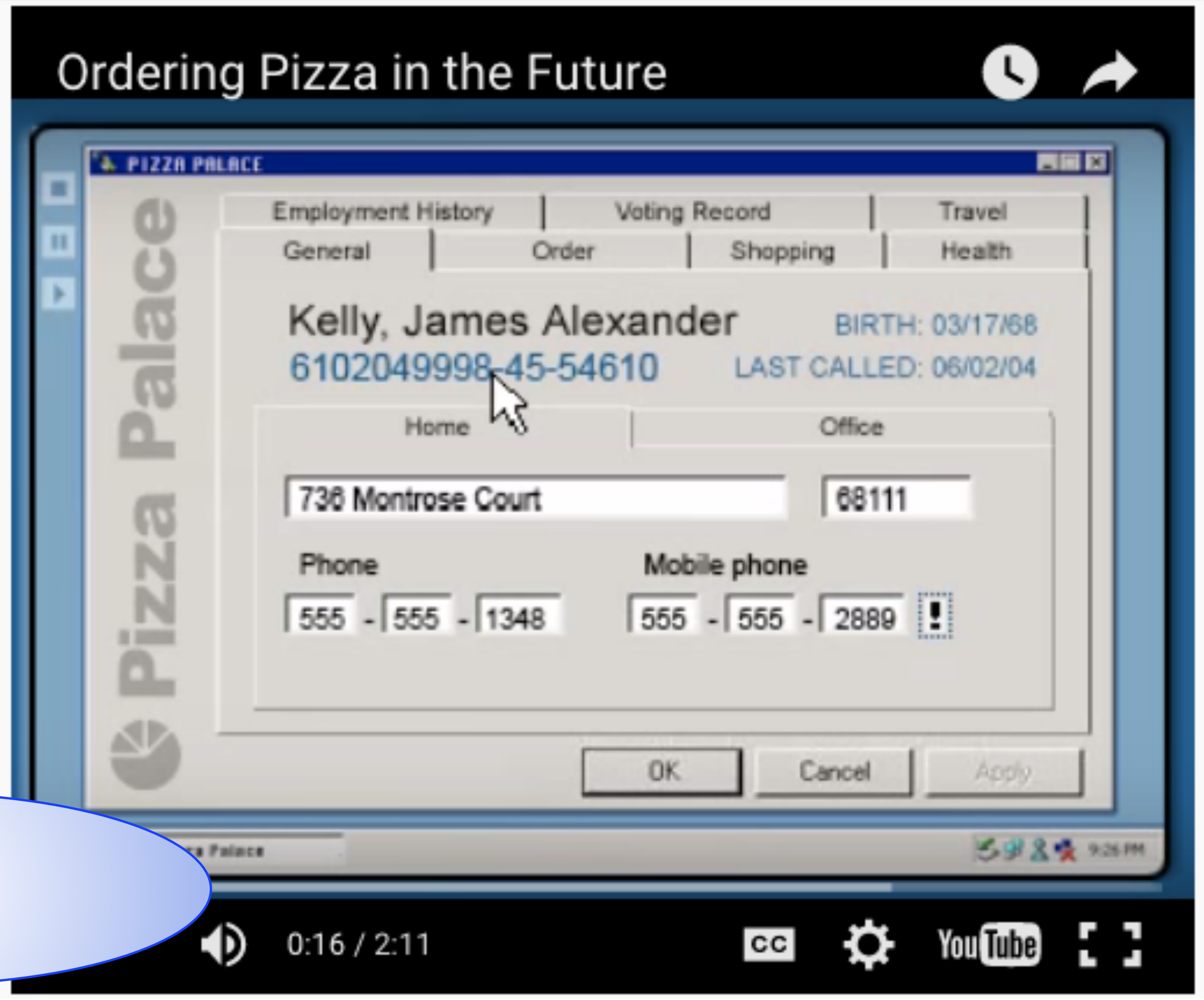
- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.

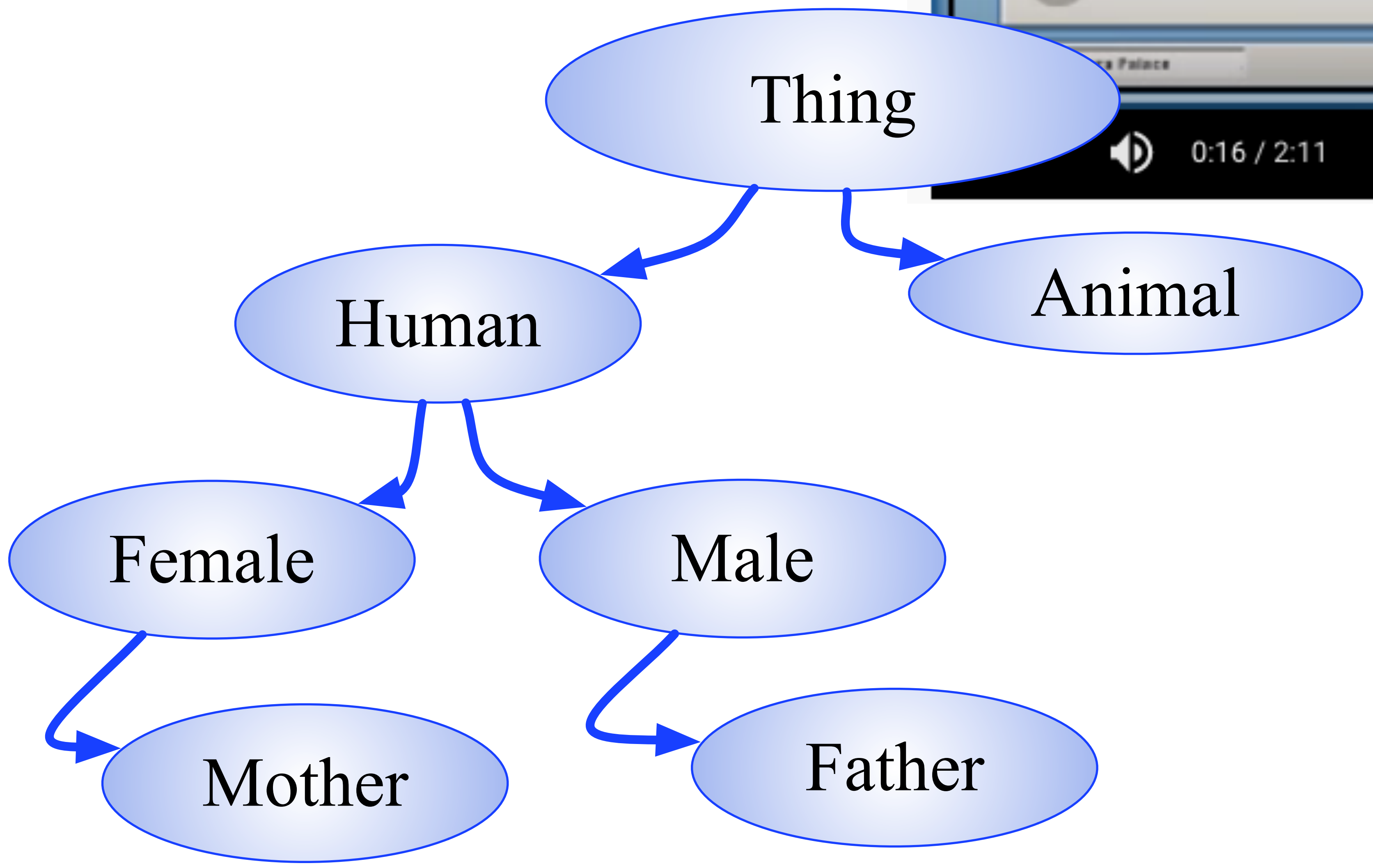
- RDF - Formal semantics is built upon a W3C XML standard for objects called the Resource Description Framework (RDF)

- OWL - The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies.

- A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms.

- Classes (concepts) are abstract groups, sets, or collection of objects (example: human, woman)
- Individuals (instances) are the specific objects, e.g. Josef is a Father
- Attributes (properties) describing objects (individual and classes) in the ontology. Example: Human hasName, Josef has name Josef Noll

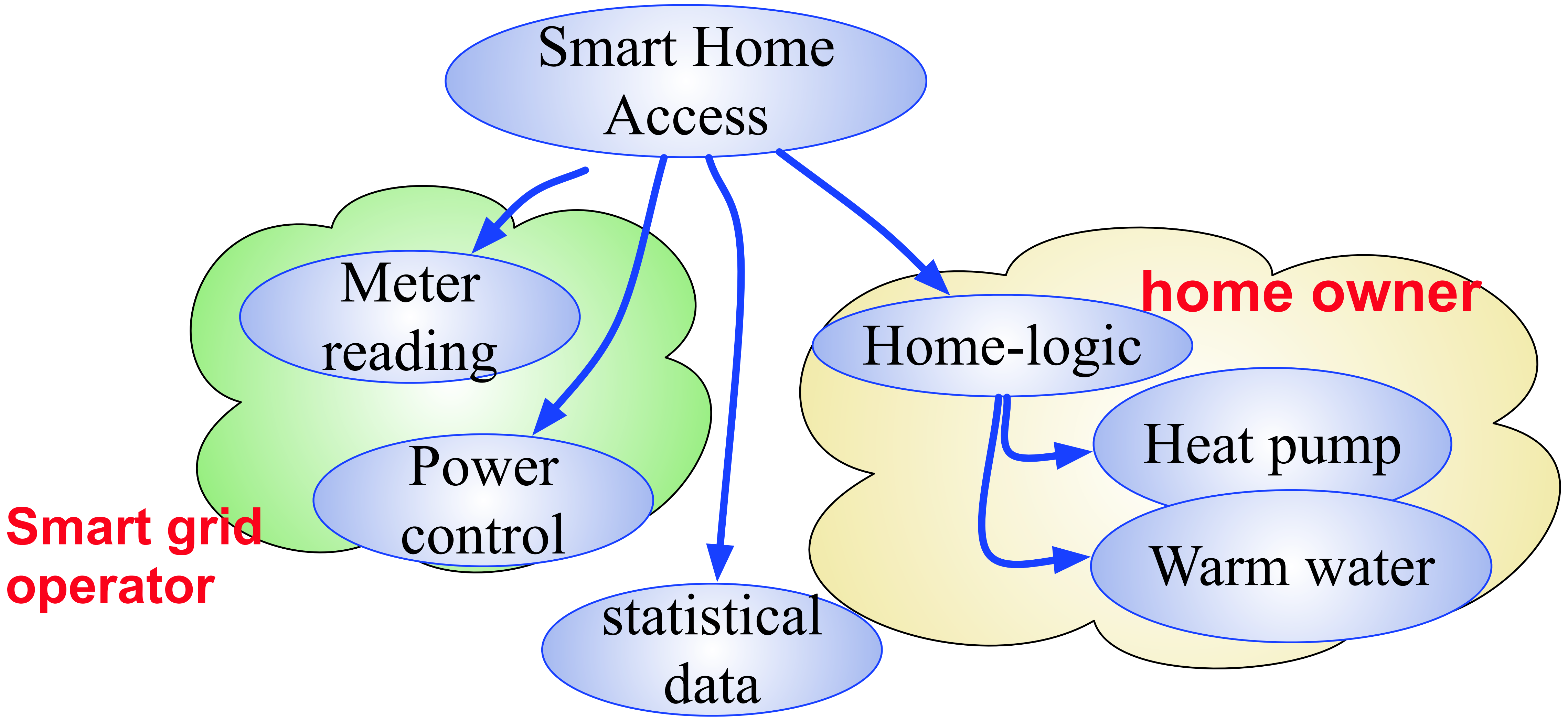




further reading:
http://www.slideshare.net/SergeLinckels/semantic-web-ontologies

Semantic attribute based access control (S-ABAC)

- Access to information
  - ➡who (sensor, person, service)
  - ➡what kind of information
  - ➡from where
- Attribute-based access
  - ➡role (in organisation, home)
  - ➡device, network
  - ➡security tokens
- OWL & SWRL implementation
- Rules inferring security tokens

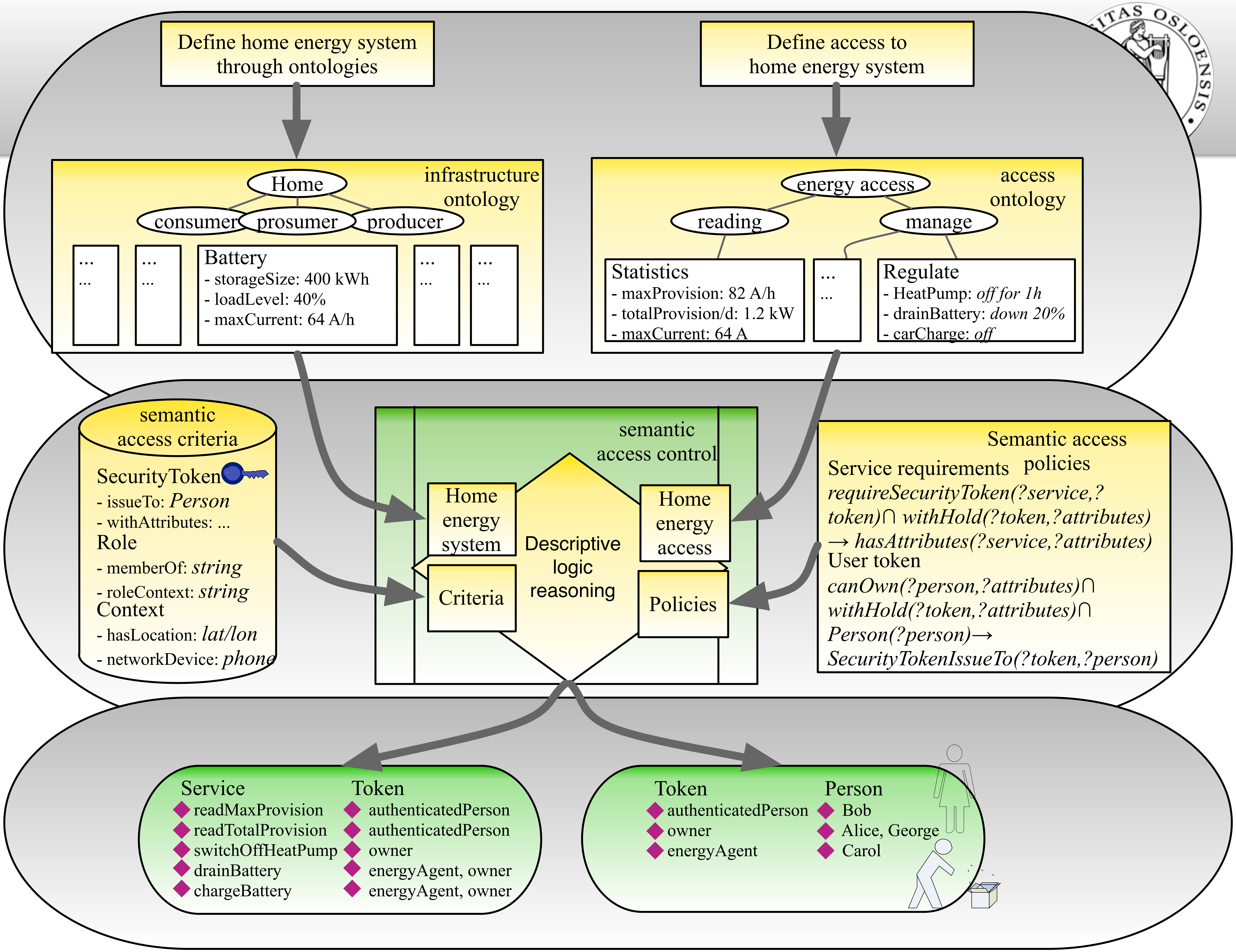


Attributes: roles, access, device, reputation, behaviour, ...

canOwn(?person,?attributes) ∩ withHold(?token,?attributes) ∩ (Person(?person) -> SecurityTokenIssueTo(?token, ?person)

[token]	principal
◆ BasicToken_1	◆ Carol
◆ BasicToken_2	◆ Alice

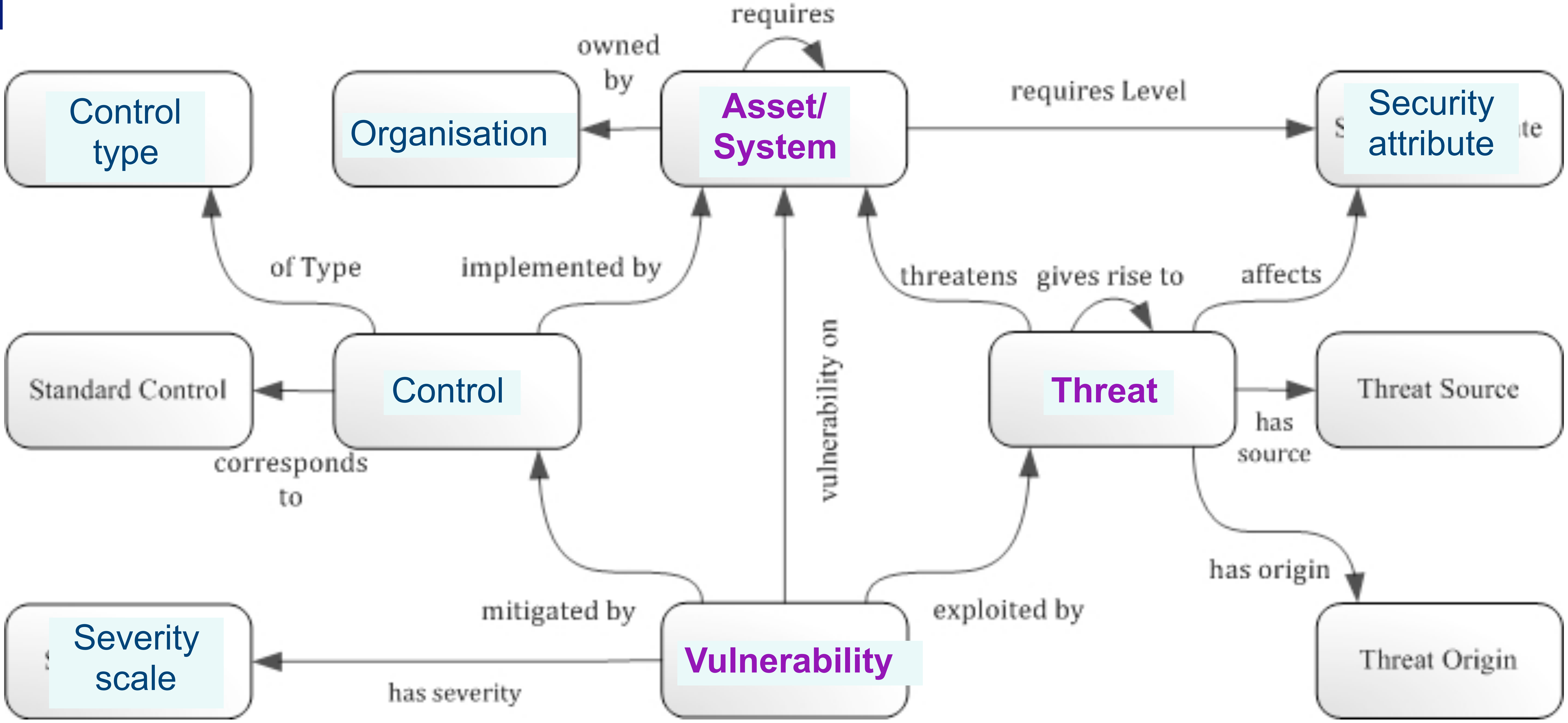
Smart Home: Complex access control



Security Ontologies
- traditional view
- Application-oriented view

Traditional approach

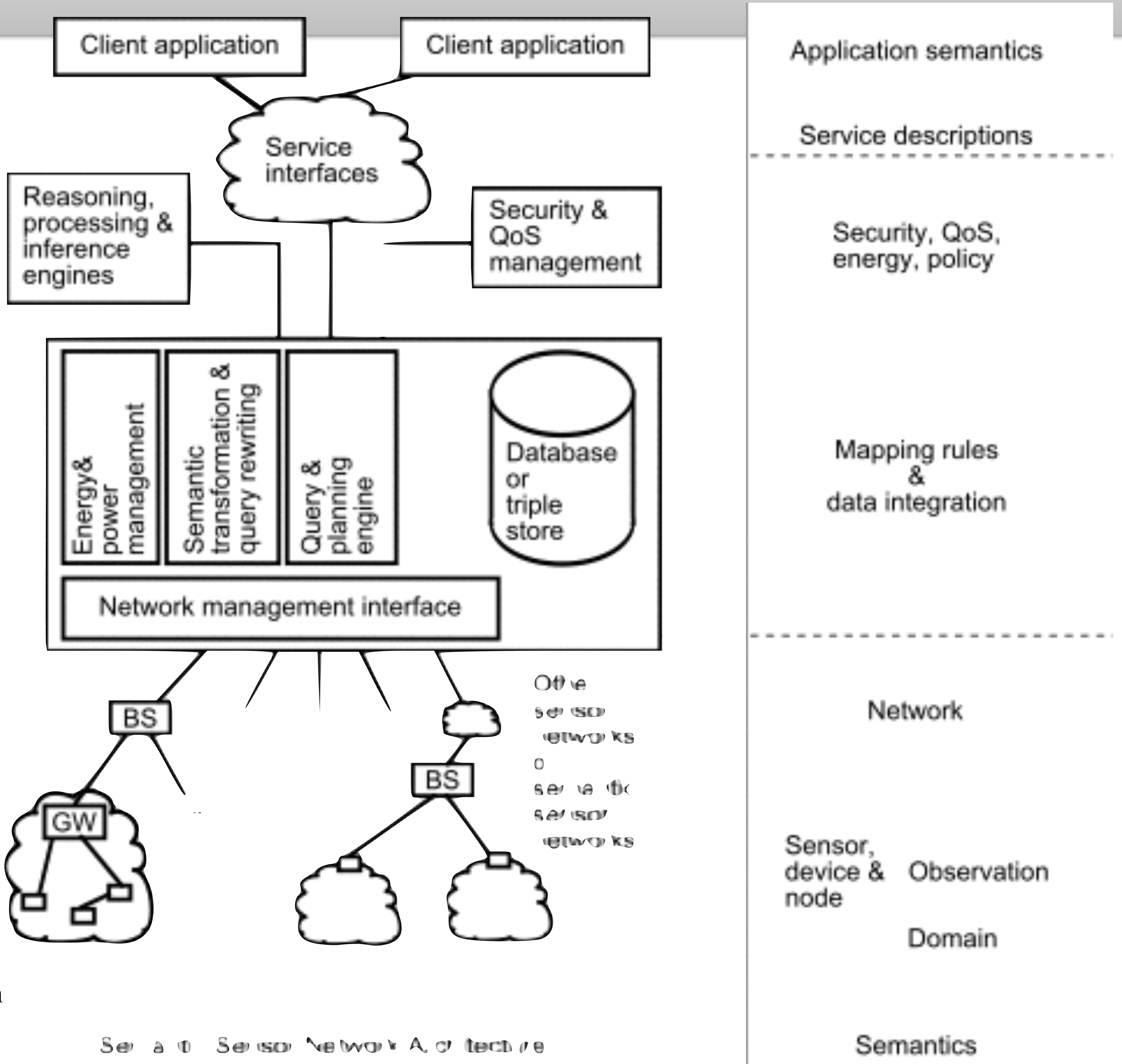
- Combined approach, addressing threat, vulnerability, system impact and control



[source: http://securityontology.sba-research.org/]

Sensor Network Architecture

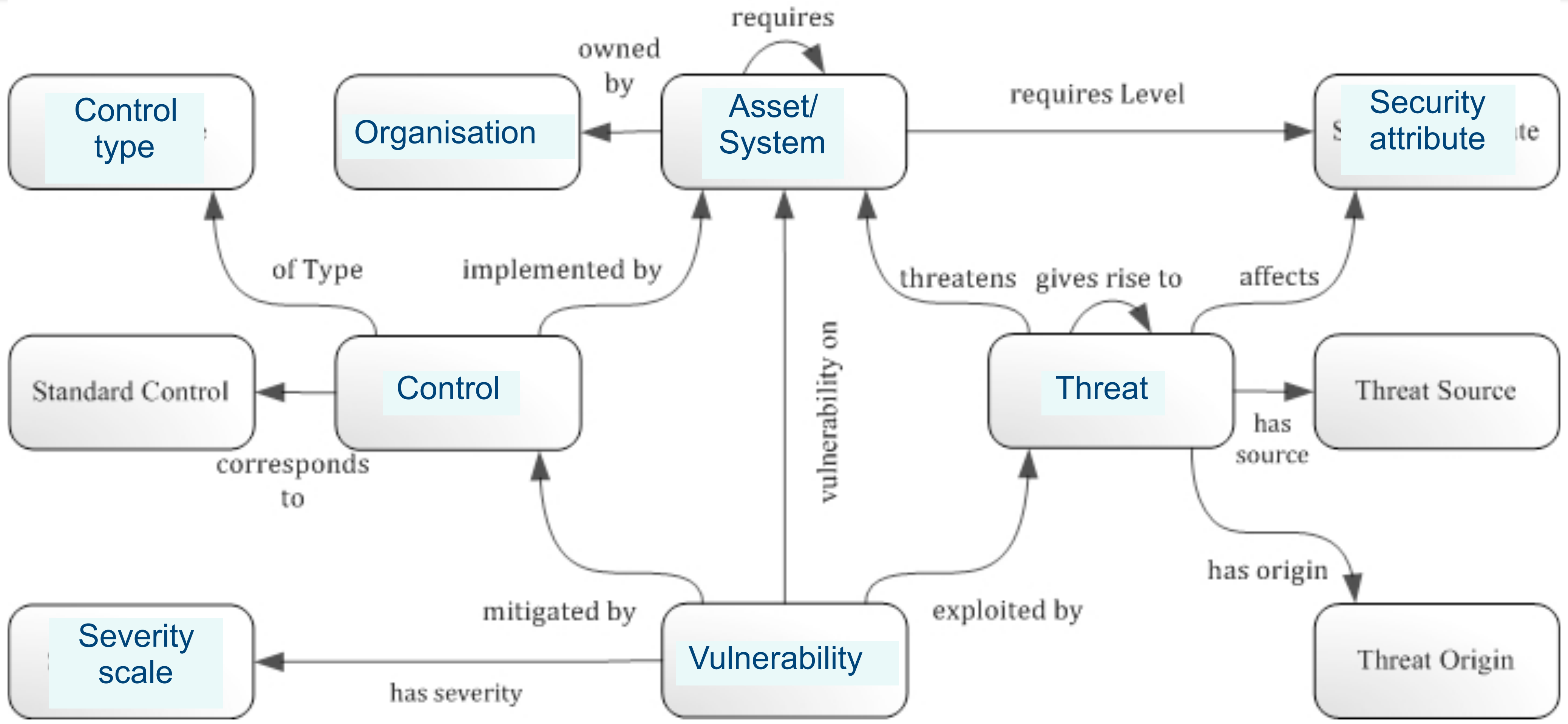
- Semantic dimension
  - ➡ Application
  - ➡ Services
  - ➡ Security, QoS,
  - ➡ Policies
  - ➡ mapping
- System
  - ➡ sensor networks
  - ➡ gateway
  - ➡ base station



Source: Compton et al., A survey of semantic specification of sensors, 2009

Limitations of the traditional approach

- Scalability
  - ➡ Threats
  - ➡ System
  - ➡ Vulnerability
- System of Systems
  - ➡ sensors
  - ➡ gateway
  - ➡ middleware
  - ➡ business processes

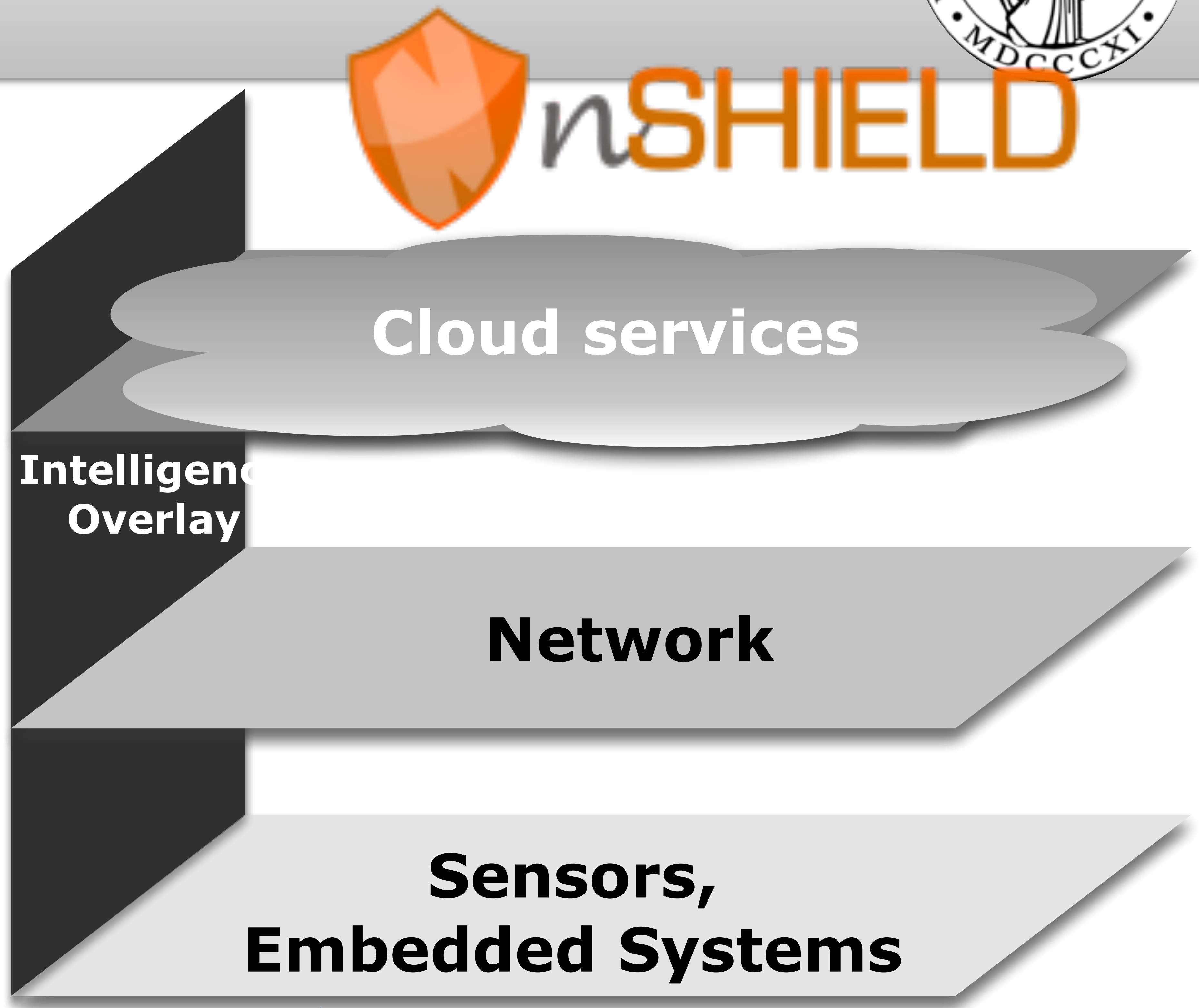


Recommendation:

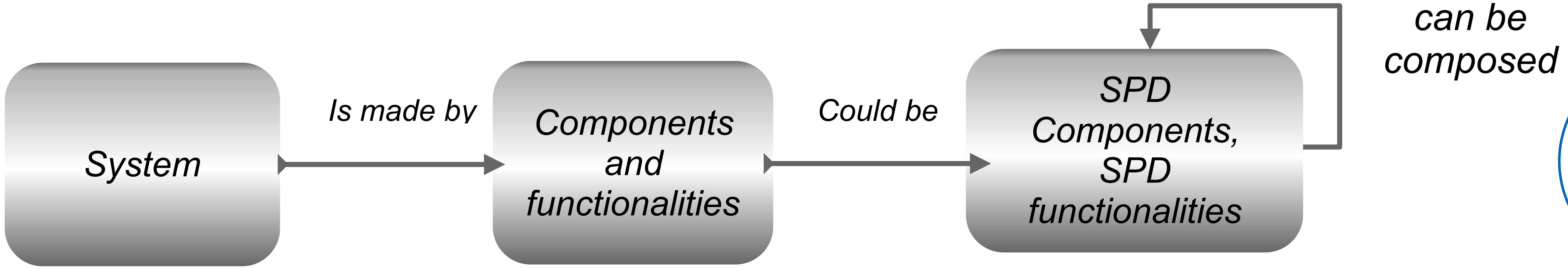
One ontology per aspect:
- security
- system
- threats
...

Applied security

- Security, here
  - ➡ security (S)
  - ➡ privacy (P)
  - ➡ dependability (D)
- across the value chain
  - ➡ from sensors to services
- measurable security

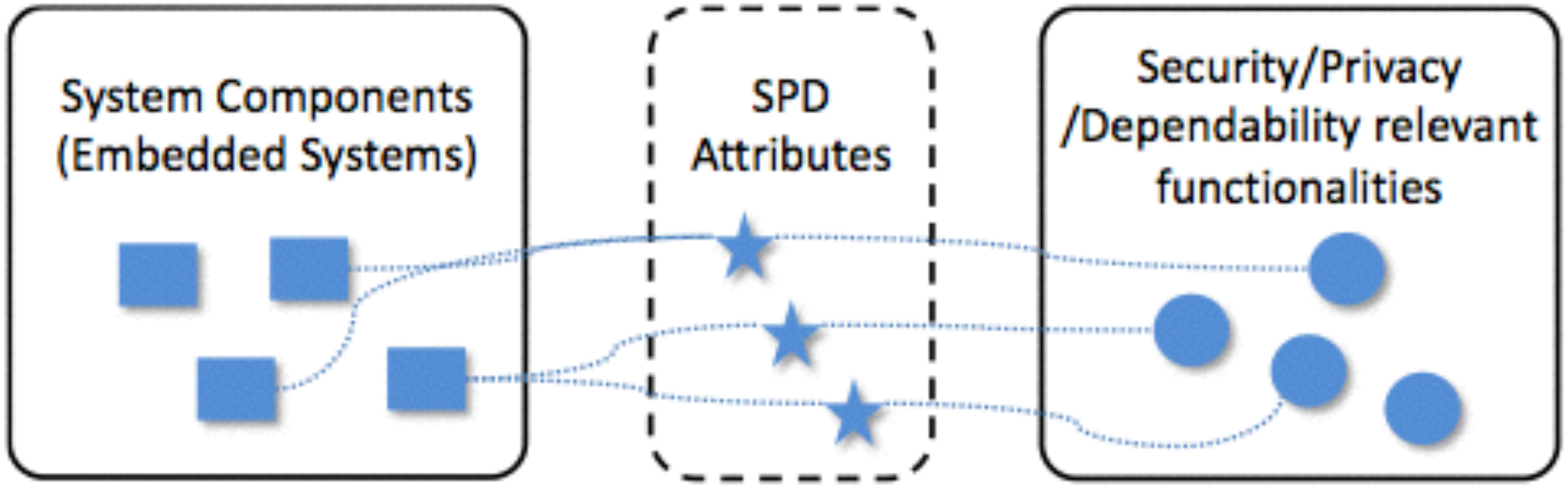


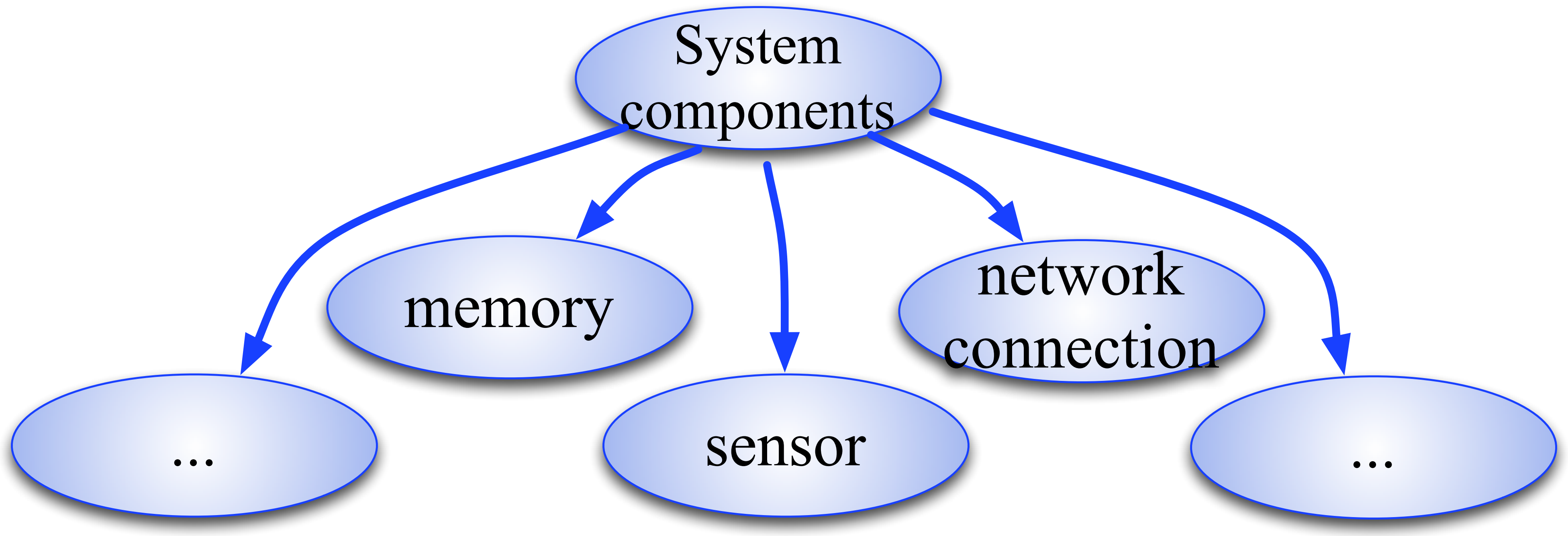
What ontologies are needed?

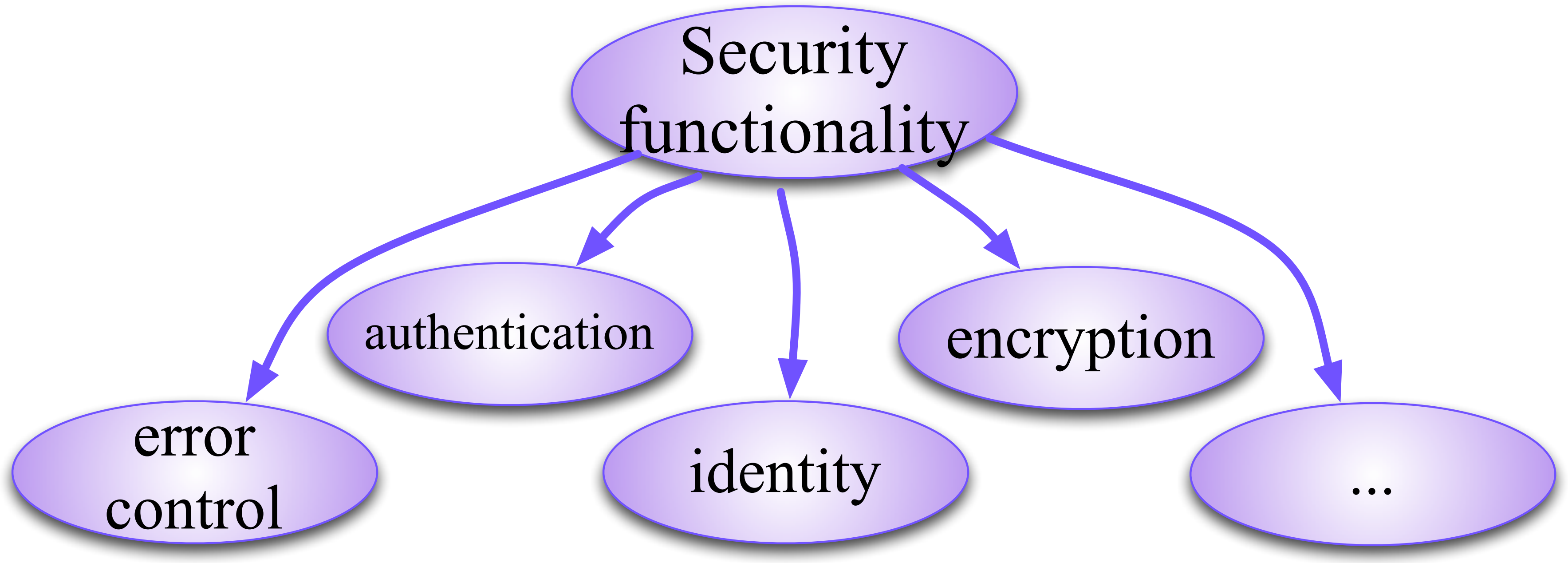


Security description

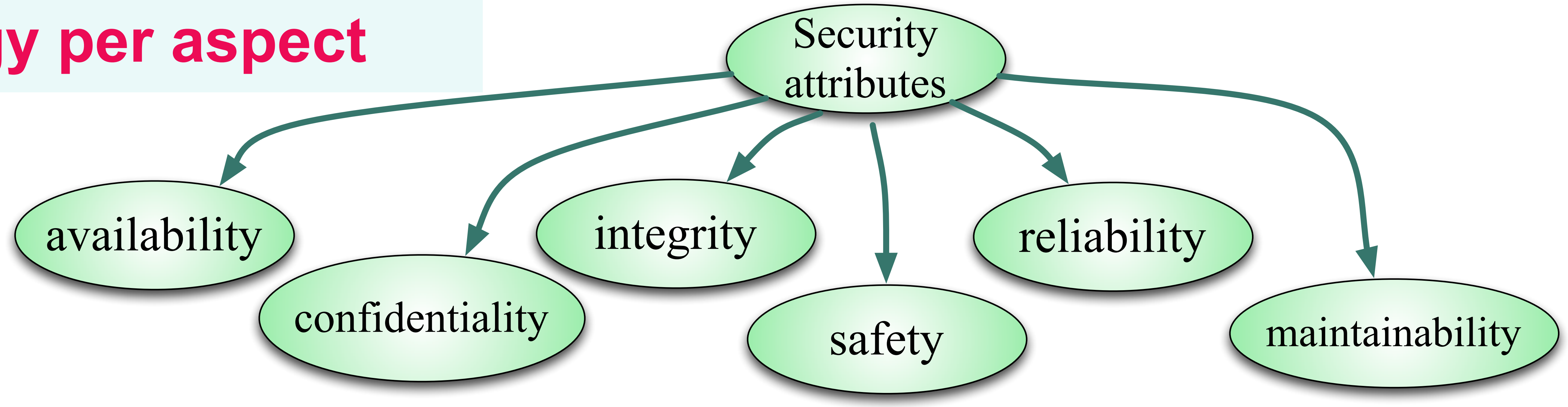
- Ontologies for system, security attributes, security functionality







Recommendation: One ontology per aspect



SPD Attributes

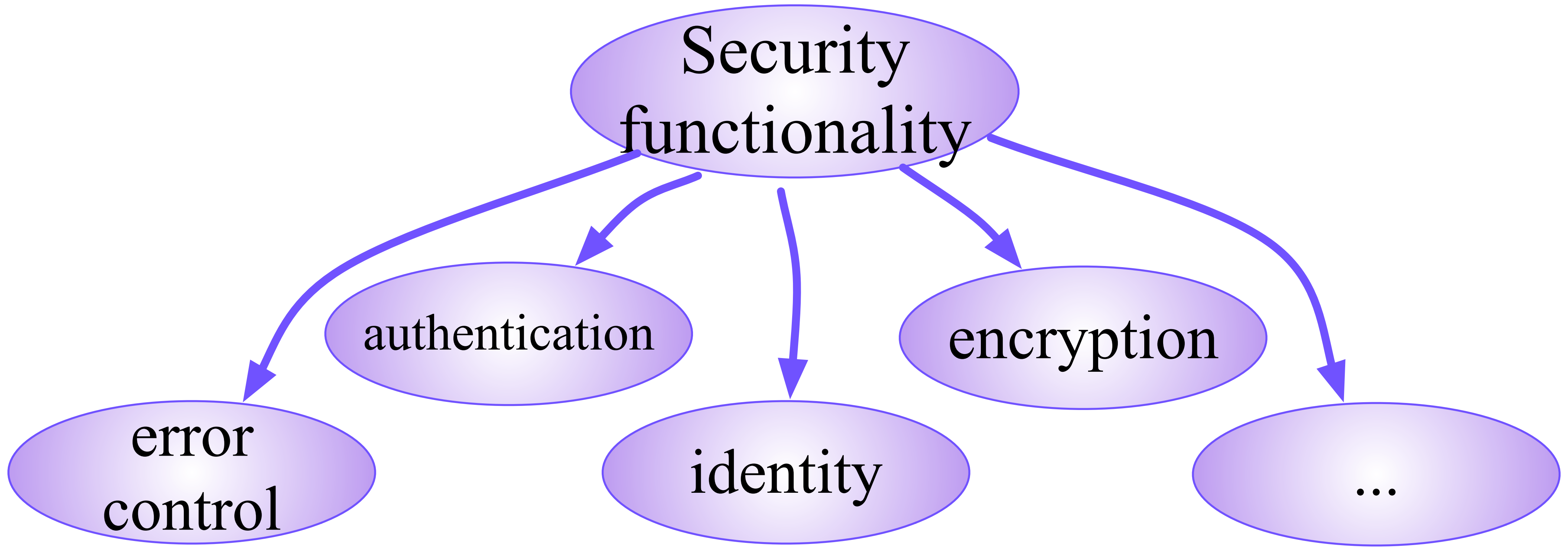
SPD Attributes	Description
Avaliability	
Integrity	
Safety (D)	
Reliability	
Maintainability(D)	feedback
Non-repudiation	
Authorization	
Accountability	
adaptability	Based on available resources
Accessibility (Attack surface)	Interaction points with system (Interfaces from which system could be attacked)
(p) Data controllability	System user could control his/her data
Peer authenticity	
Message authenticity	
User authenticity	
(p)Location Anonymity	
Entities Anonymity	
Visualization (D)	

SPD Attributes	Description
Auditing	Logging
Invulnerability (number and weight of Vulnerability)	
Resilience	
Alarm	
Continuity	
Detection accuracy	
Predict accuracy	
Immunity (inviolability)	
Recoverability	
Preventability	Ability to prevent occurrence or reduce consequences
Component Compatibility	
Physical security	
Usability	Simplicity, learnability
Trace-ability (forensic-ability)	
Configurability	- Ability to change configuration for different use case. - Remote configuration possibility.
Scalability (D)	Ability to include more node.

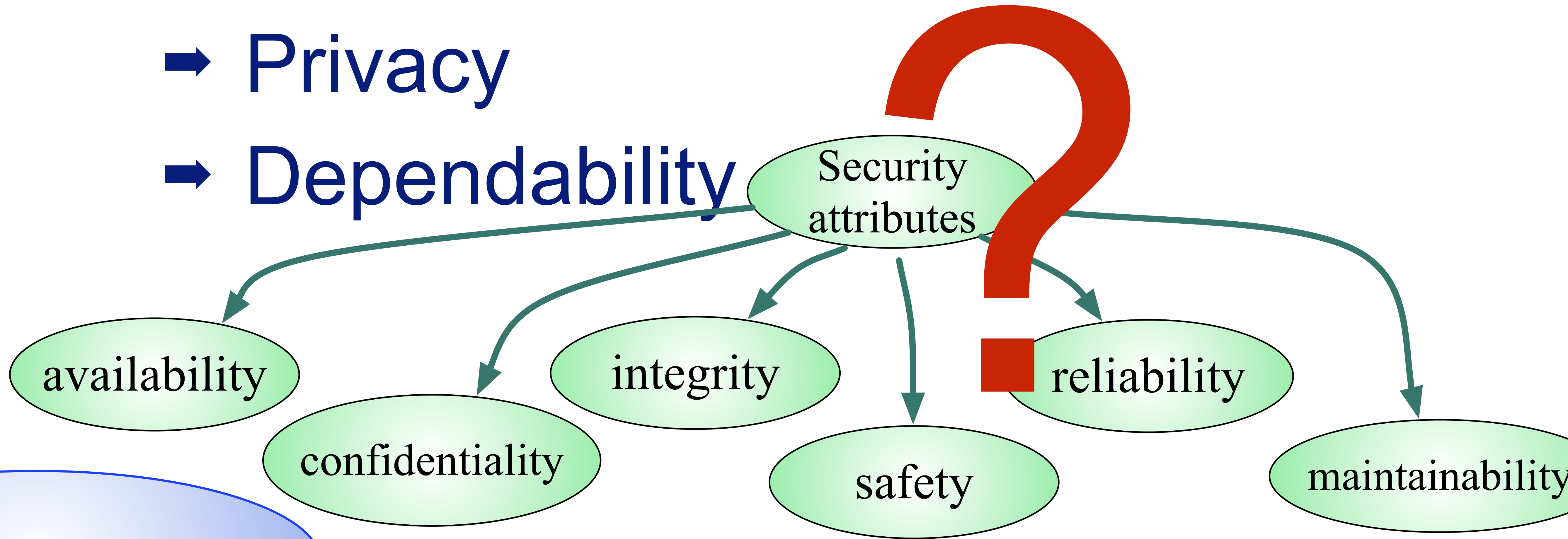
SPD Attributes	Description
Indemnification	
Transmitted data Confidentiality	
Traffic flow confidentiality	
stored data confidentiality	
Virtualization	
Update-ability	
Human awareness.	
Virtualization	

Task: Mapping SPD attributes into areas

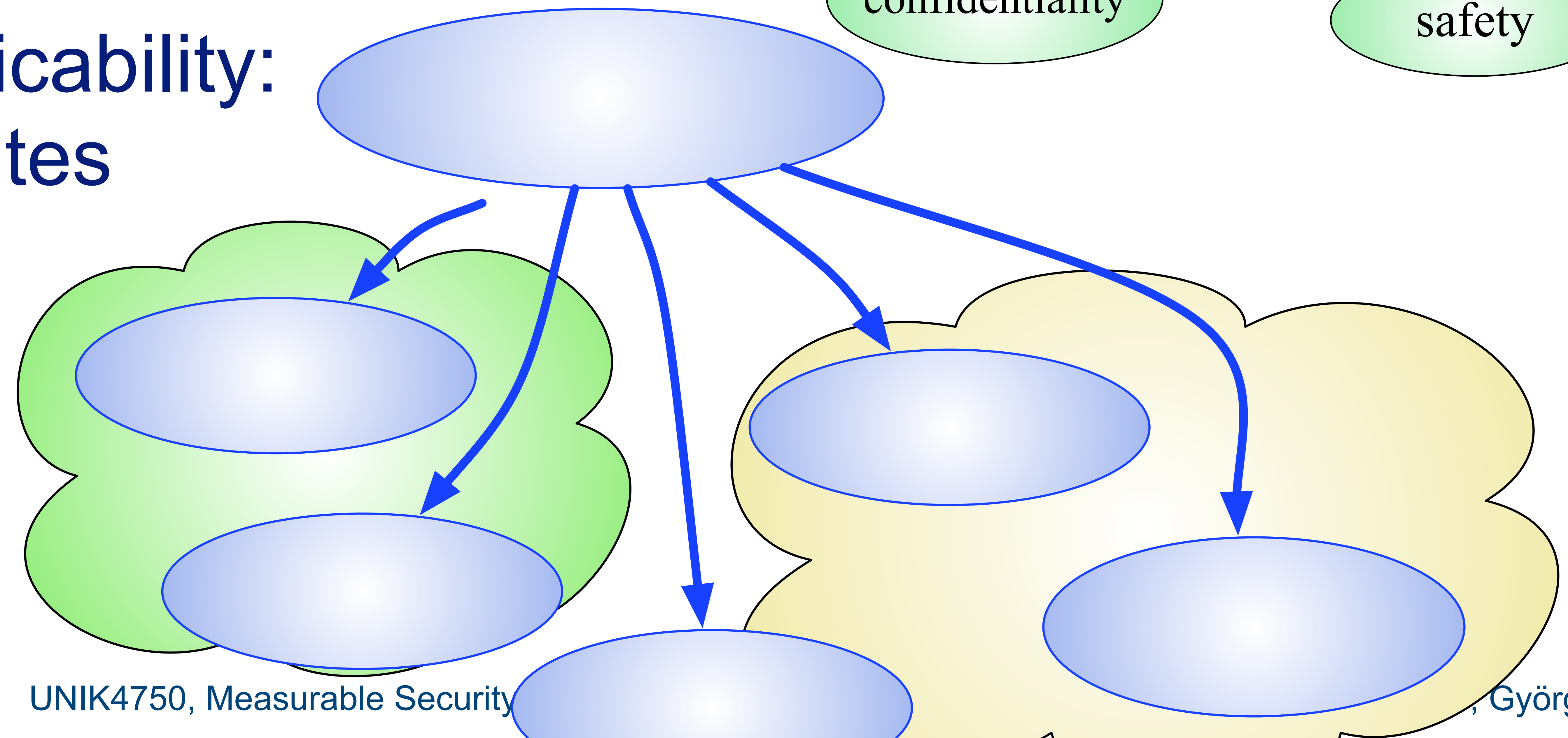
- From Security Functionality



- To Security attribute mapping
  - ➡ Security
  - ➡ Privacy
  - ➡ Dependability



Task for System applicability:
- map relevant attributes



L8 - Learning outcomes

Having followed the lecture, you can

- explain components of the Smart Grid (AMS) System of Systems
- can explain the difference between functional, non-functional and security components
- provide examples of security challenges in IoT
- explain the difference between the web, the semantic web, web services and semantic web services
- explain the core elements of the Semantic Web
- apply semantics to IoT systems
- provide an example of attribute based access control
- discuss the shortcomings of the traditional threat-based approach
- list the main elements of the semantic descriptions of s,p,d functionalities
- perform a semantic mapping of s,p,d attributes *(future work L13, L14)*
- **Further readings**
- https://plus.google.com/u/0/+MarcelEggum/posts/9kbGFHA972J (about the Semantic Web)
- http://www.slideshare.net/SergeLinckels/semantic-web-ontologies (on Ontologies)