

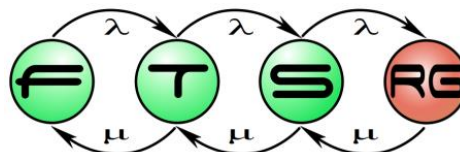


Cyber-Physical Systems

Prof. András Pataricza

Budapest University of Technology and Economics

pataric@mit.bme.hu



Cyber-Physical Systems definition

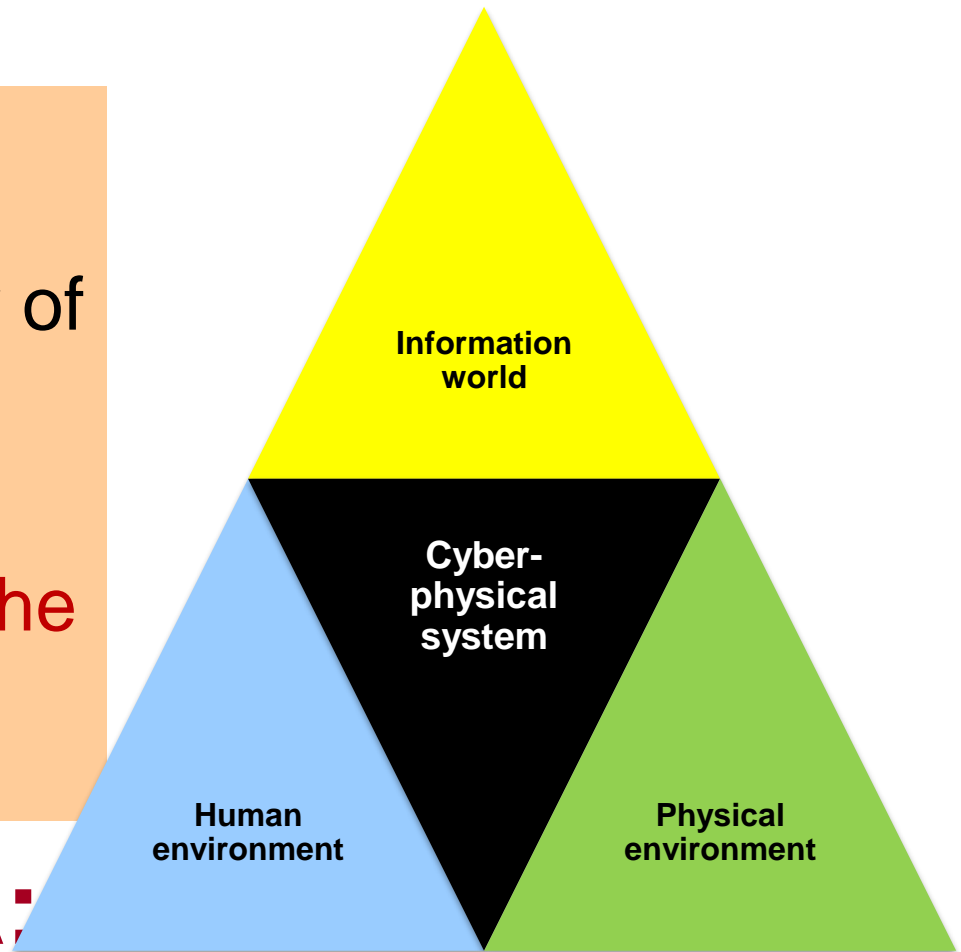
- *“Cyber-Physical Systems or “smart” systems are co-engineered interacting networks of physical and computational components. These systems will provide the foundation of our critical infrastructure, form the basis of emerging and future smart services, and improve our quality of life in many areas.”*



**National Institute of
Standards and Technology**
U.S. Department of Commerce

Let's reach an unlimited intelligence by the synergy of

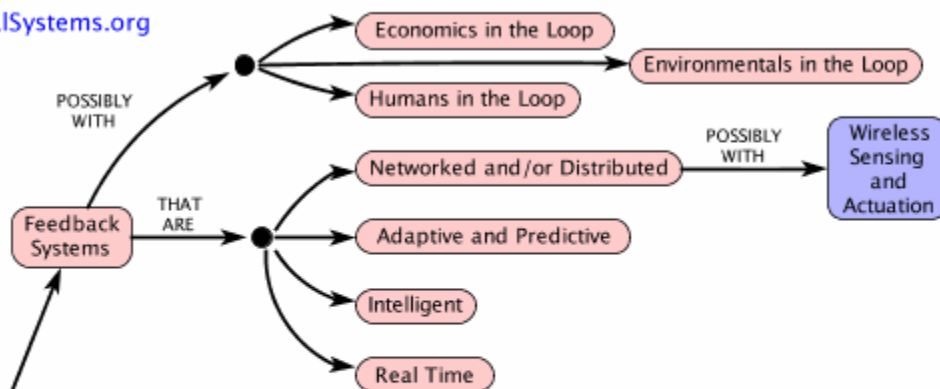
- intelligence in the cyber space and
- ES interfacing them to the physical world



THE NEW ERA: INTERNET OF THINGS AKA CYBER-PHYSICAL SYSTEMS

[See authors and contributors.](#)

POSSIBLY
WITH



/ARE

REQUIRE.

HAVE
APPLICATIONS
IN

Cyber Security

→ Resilience

Privacy

Malicious Attacks

Intrusion Detection

Improved Design Tools

THAT
ENABLE

Design Methodology

THAT

Specification, Modeling, and Analysis

OF,

Hybrid and Heterogeneous Models

Models of Computation

Continuous and Discrete

Networking

Interoperability

Time Synchronization

Scalability and Complexity Management

THROUGH

Modularity and Composability

Synthesis

Interfacing with Legacy Systems

Validation and Verification

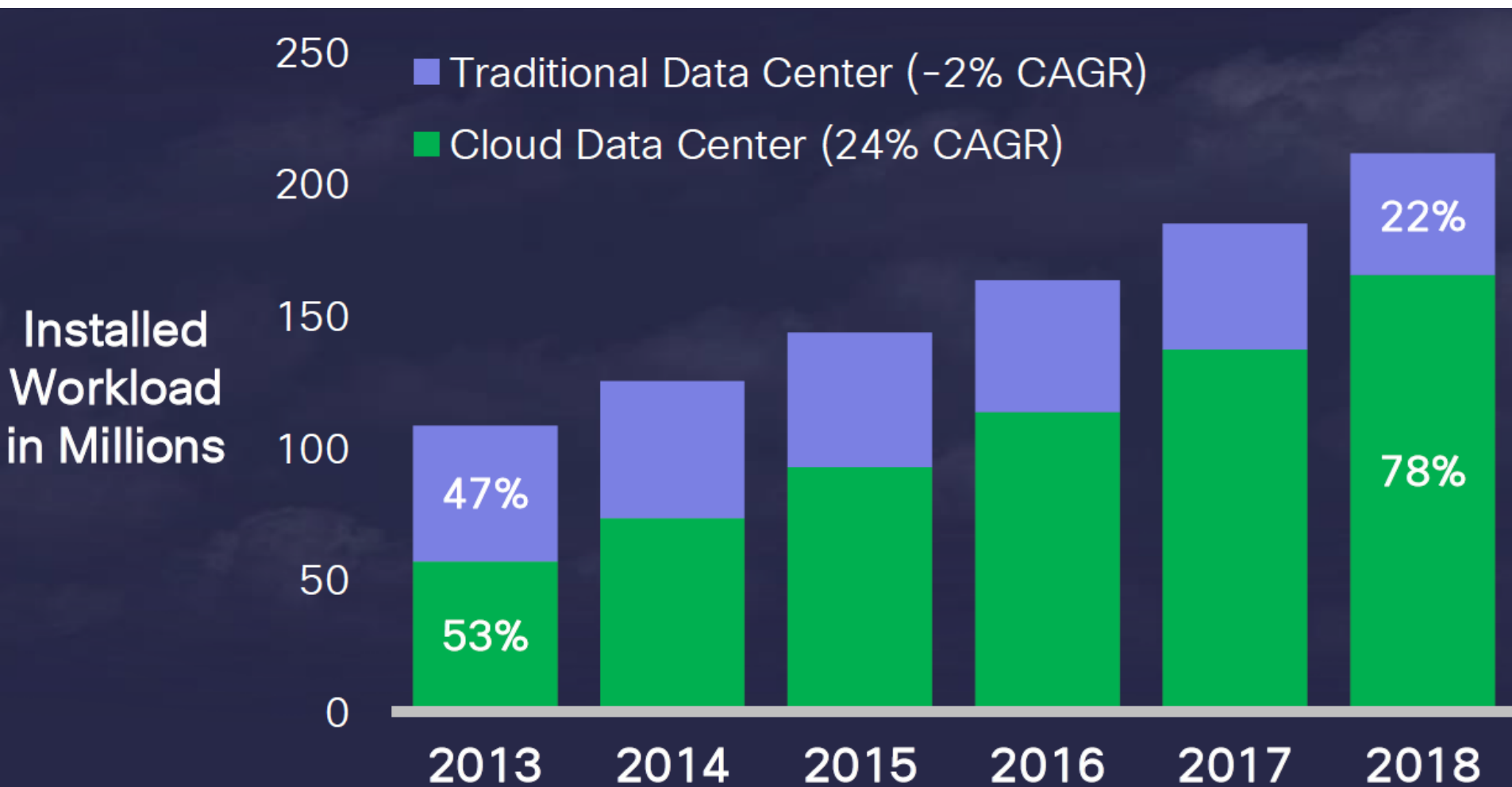
Assurance

Certification

Simulation

Stochastic Models

Cloud computing around the globe

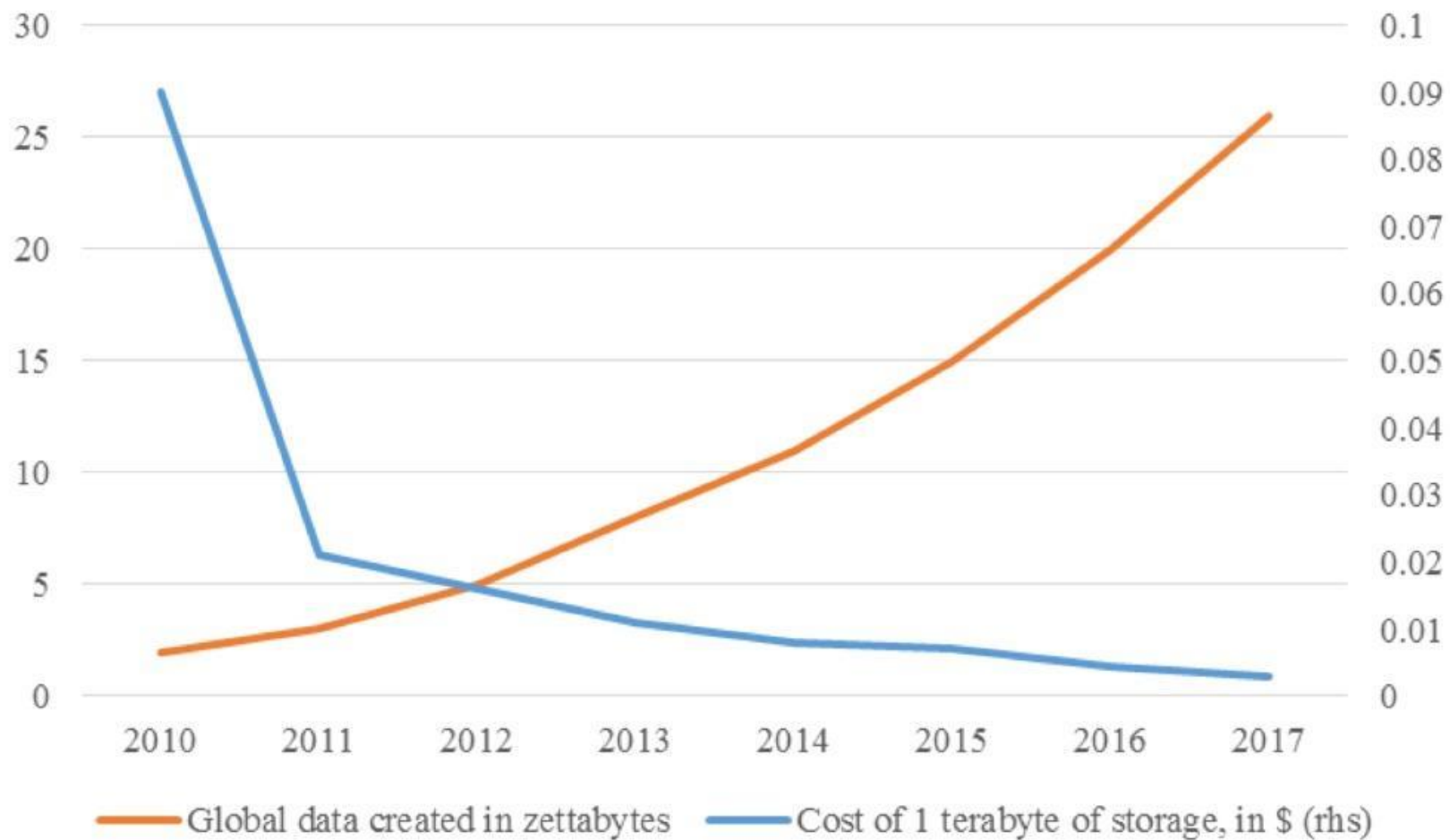


Source: [Cisco Global Cloud Index: Forecast and Methodology 2013–](#)

Mission critical cloud computing

- August 31, 2015
 - Federal aviation administration
 - 108\$ million now, \$1 billion in 10 years
 - Source: [CSC news](#)
- Network Functions Virtualization
 - The „telco” cloud
 - Source: [NFV](#)

Figure 3: Costs of storage and global data availability, 2009-2017



Source: Reinsel, Gantz and Rydning (2017); Klein (2017). One zettabyte is equal to one billion terabytes.

Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)



The Statistics Portal <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>

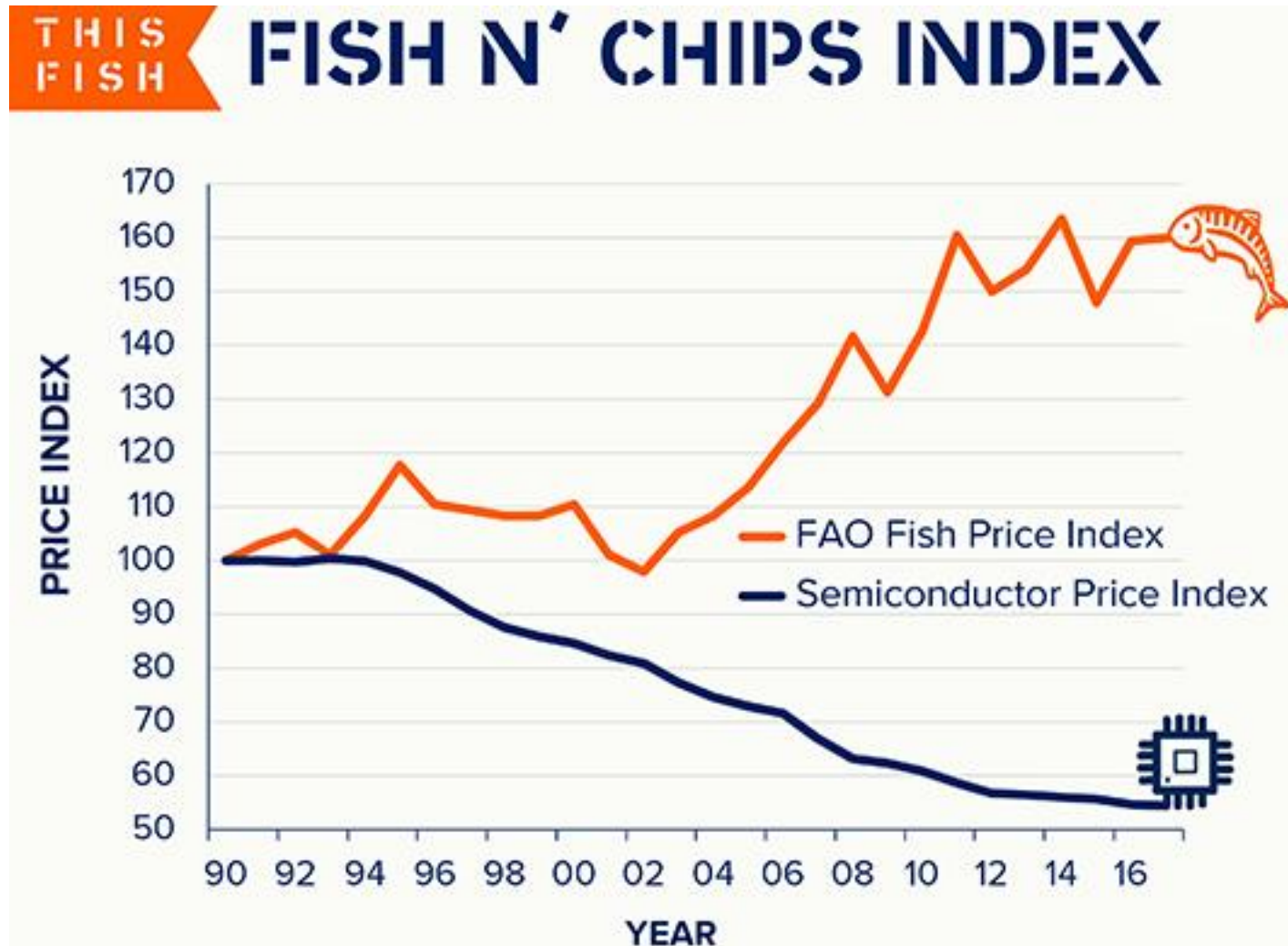
Fish and chips



First-ever Fish n' Chips Index shows meta-trends driving seafood innovation

<https://thisfish.info/generic/article/fish-n-chips-index-2017/h>

<https://thisfish.info/generic/article/fish-n-chips-index-2017/>



SOURCE | FAO Fish Price Index & U.S. Bureau of Labor Statistics Producer Price Index

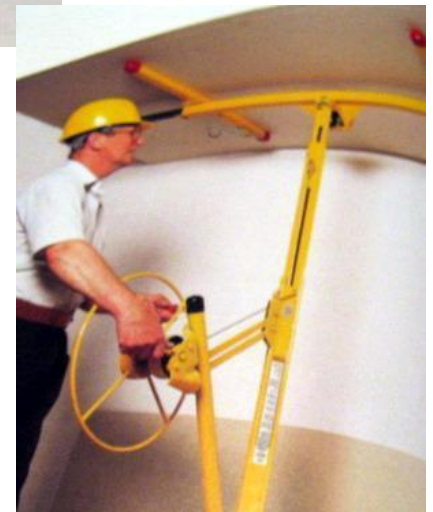
ES paradigm shift

Traditional

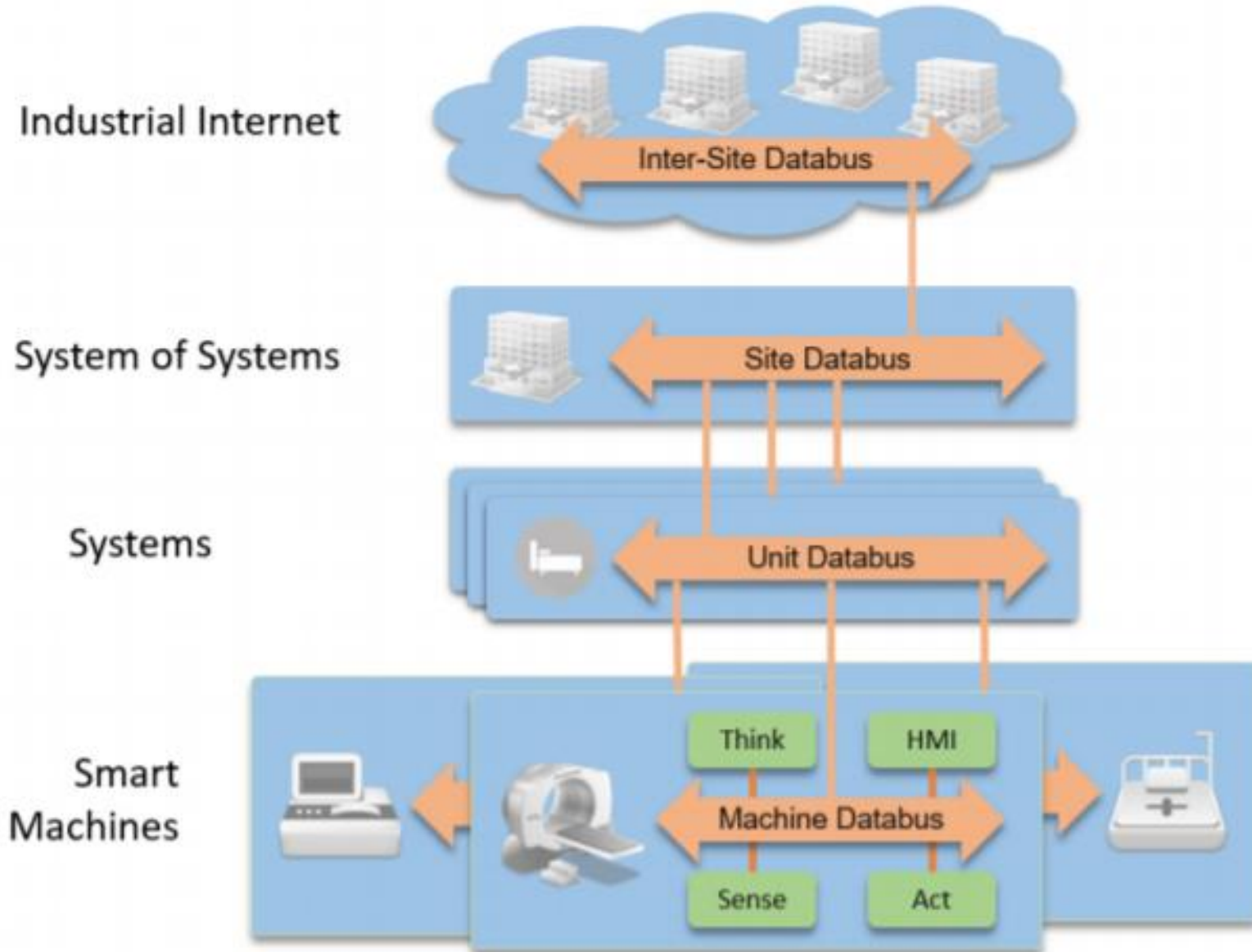


Industrialized

- Best component technologies
- Standardized components
- Automated system design



Hierarchy-Industrial Internet





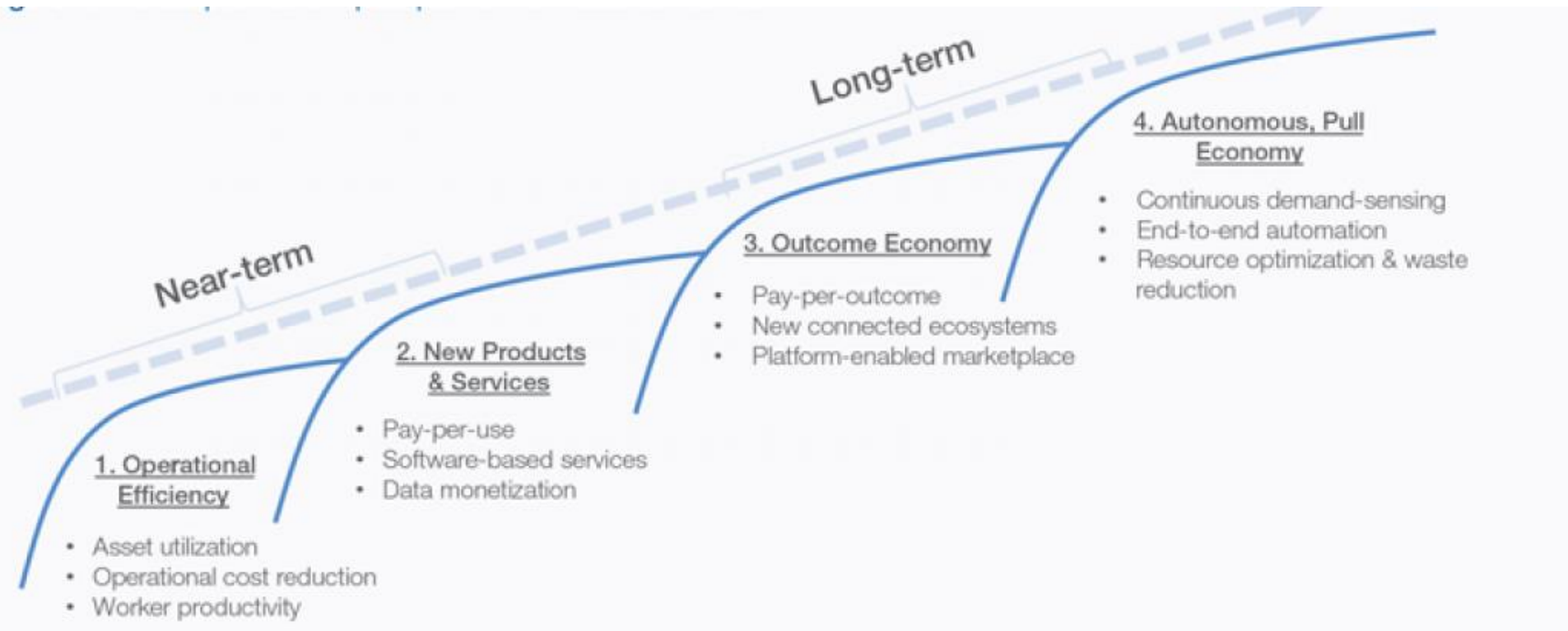
Industrial Internet of Things:

Unleashing the Potential of Connected Products and Services

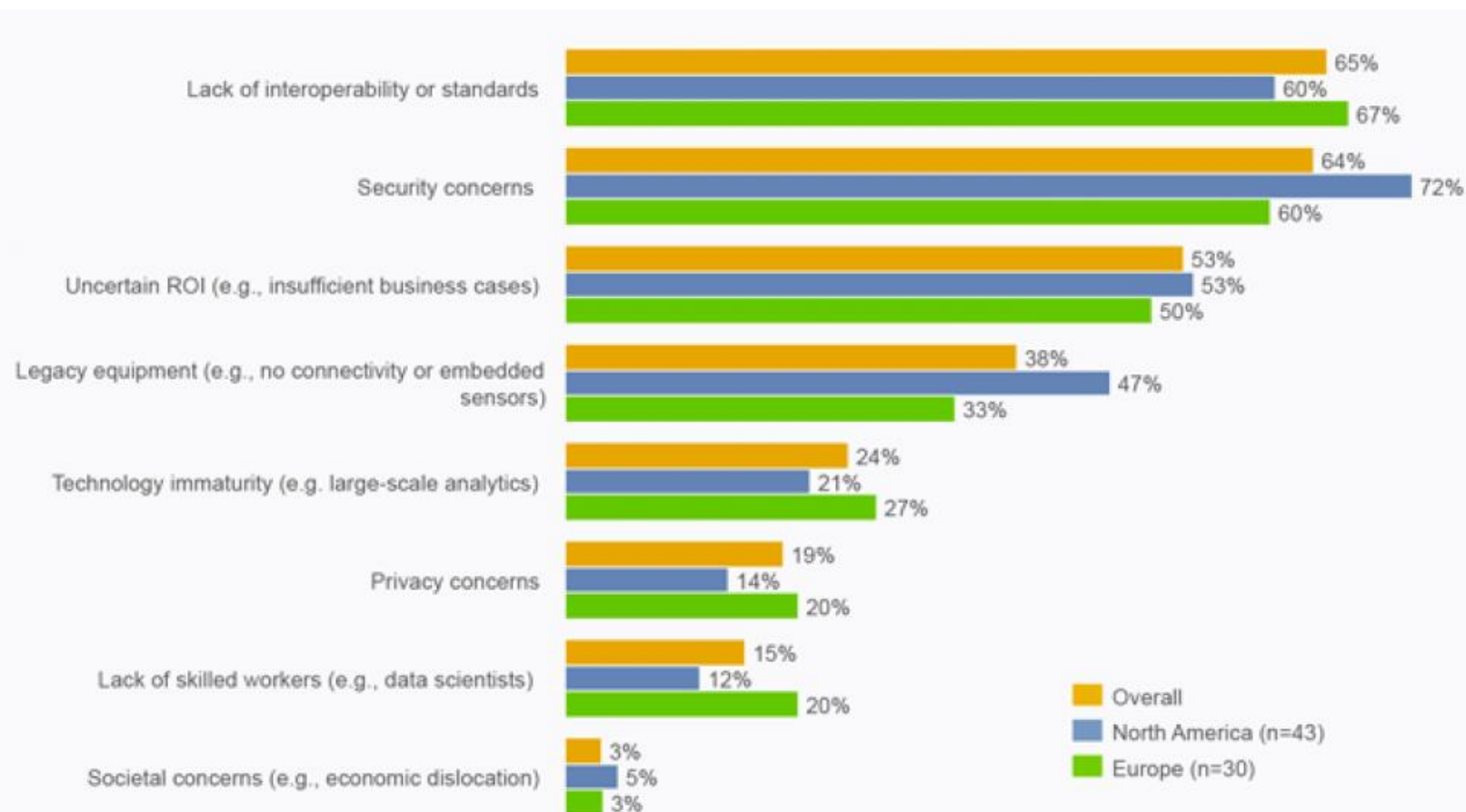
http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf

Impact of CPS (IIoT) on the economy

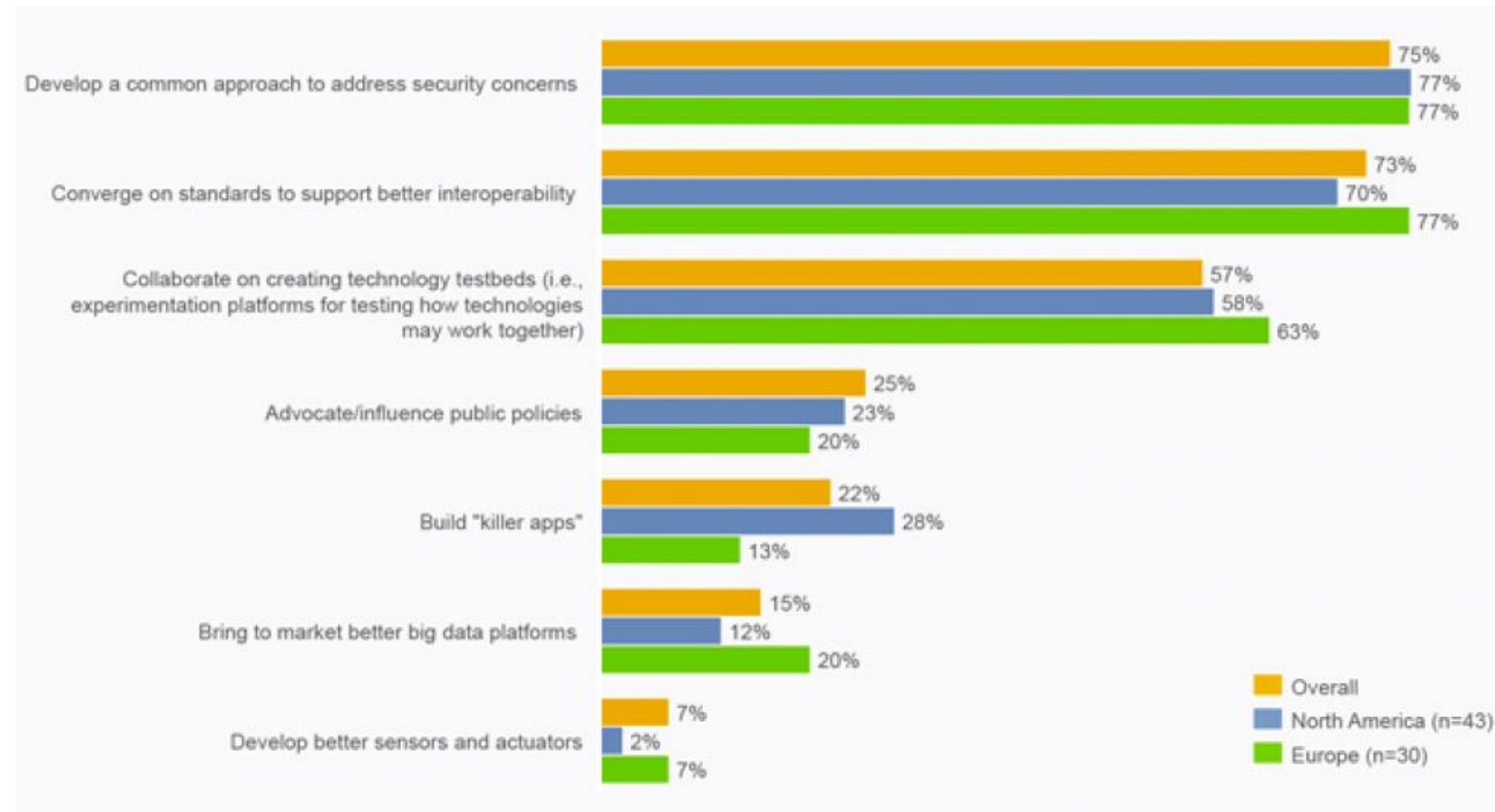
Industrial Internet evolution



Barriers



IT technology



Local vs. remote data




Now 8:00 am CEST

Weekend

Extended

Month

Satellite




Sunny

11°

RealFeel® 9°

Hourly Forecast

Today Sep 8



Times of clouds and sun

Hi 19°

RealFeel 20°

more

Tonight Sep 8




Mainly clear

Lo 8°

RealFeel 6°

more

Tomorrow Sep 9



Partly sunny


Hi 21°

RealFeel 23°

more

Video Weather Forecast

Tuesday's Forecast



Europe Weather Forecast

Kiskunlacháza-airport Satellite



Data integration

Official euro exchange rates

314.24

07 September 2015

[Other exchange rates](#)

Central bank base rate

1.35 %

22 July 2015

[Base rate history](#)

Inflation

Medium term target

3%

(±1 p.p. tolerance band)

July 2015, KSH:

0.4%

mpl Service

To use it to call the service, You can do this using the svcutil.exe tool from the command line with the following syntax:

[amok.asmx?wsdl](#)

single file:

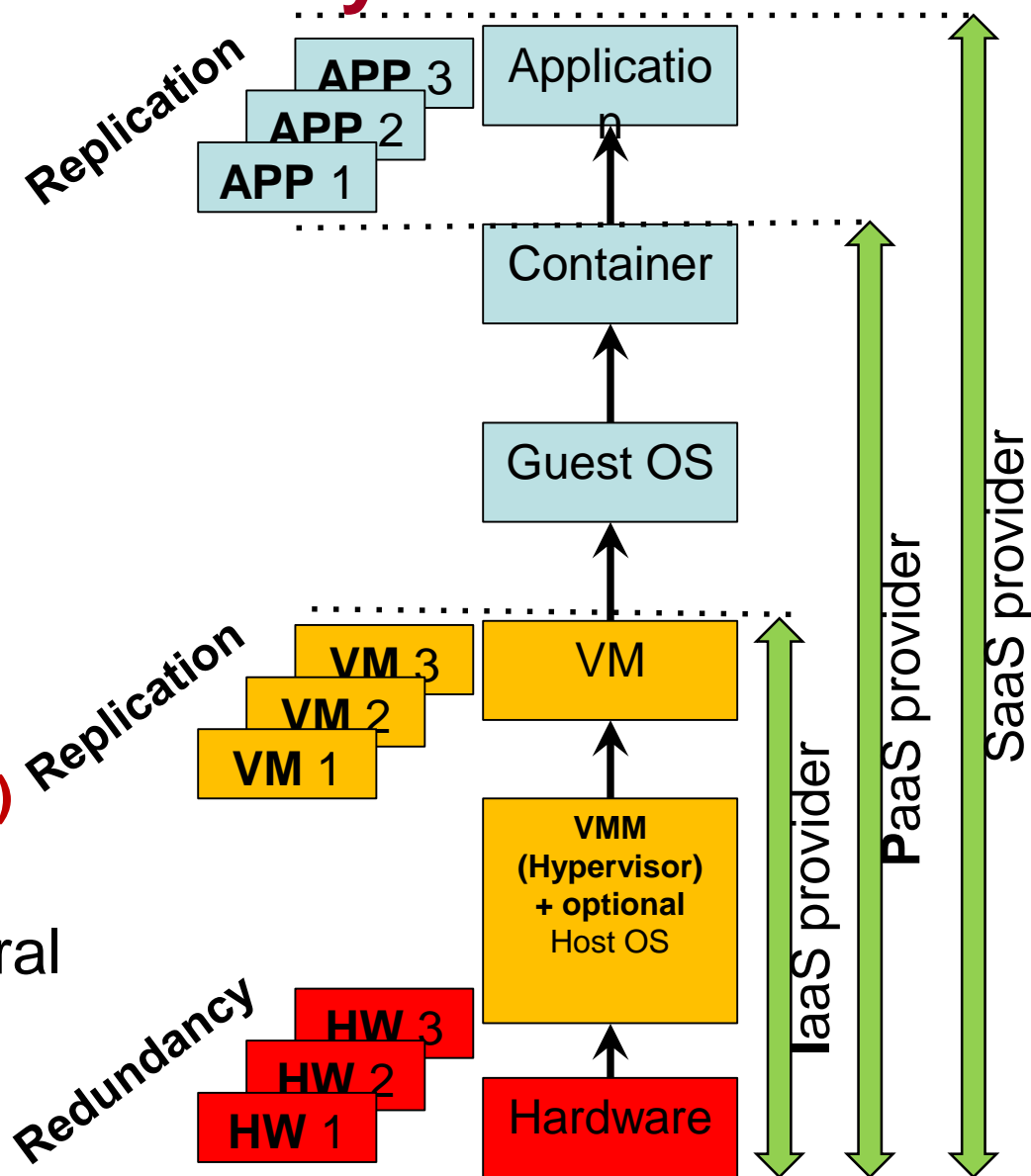
[singleWsdl](#)

that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

```
client = new MNBArfolyamServiceSoapClient();
to call operations on the service.
```

Critical services over ordinary clouds?

- Environment
 - HW/SW stack
 - Cloud service models
- Research objective
 - Scope
 - **Carrier grade IaaS**
 - SDN
 - Objective
 - **Availability (downtime)**
 - **Cost**
 - Redundancy architectural pattern
 - HW, VM, App, else

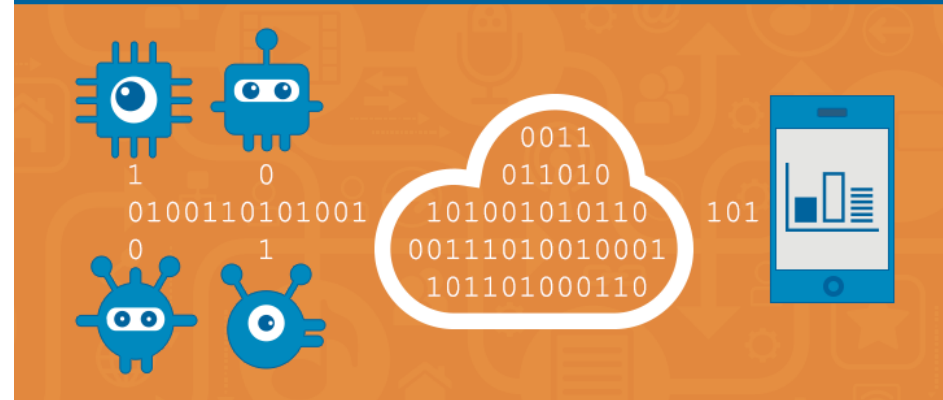


Appearance of cloud-based semantic services

- “Drag-and-drop” application prototyping
- Uniformization
 - Meta-algorithms
 - Data
 - COMPUTED
 - SENSED

IBM Internet of Things (IoT) Foundation

Cloud-connect your Things in minutes
Write apps that use the data from real physical devices



Wolfram Connected Devices Project



Example

- Cameras on riverside
- Different applications concurrently using the same primary information
- Tasks can change according to time/season/requirements
 - Identification of ships
 - Monitoring the break-up of ice
 - Monitoring the water level
 - Monitoring the speed of flood
 - Pollution check
 - Supervision of hostile entrance to the ship



On-demand Cyber Physical Systems



Problem

Service

Co

ment



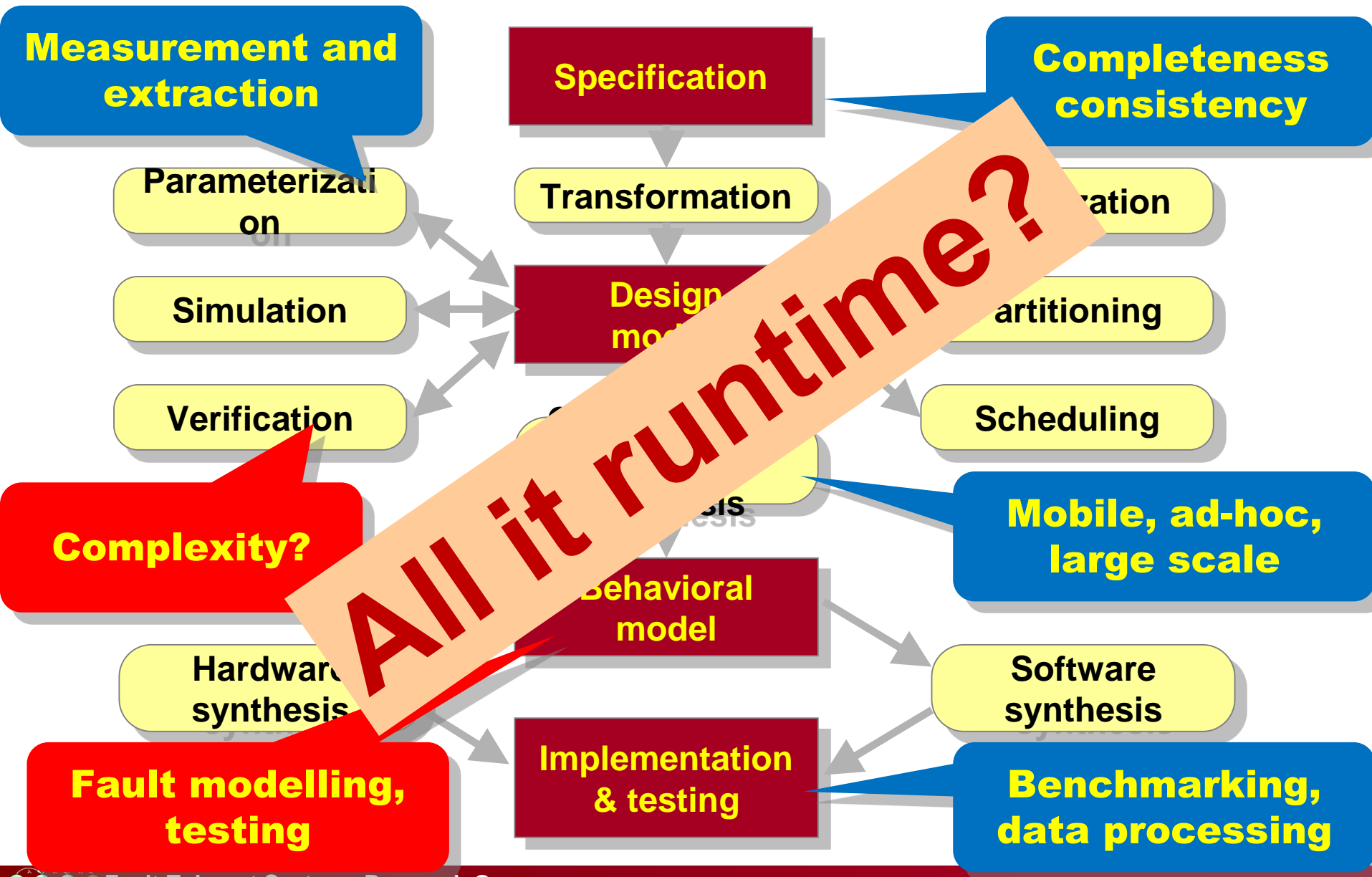
**Cyber
world**

**Distributed
over multiple
domains of control**



**Physical
world**

Critical CPS design and challenges



End of lecture 1

Service Oriented Approach

- Embedded systems provide services
 - Information of sensors
 - information of Internet
 - high level information derived
 - actuation possibility (limited)
- Services in a database
- Upon a new task: solution derived based on design patterns and available resources
- new solution deployed with no interference with the already running ones

Case study: supervising a server room

■ Observations

- temperature
- humidity
- state of doors/windows
- monitoring the power consumption weather (temp./humidity)
- temperature of outflow air of air conditioning
- state of server computers/switches (video based)



Sensor platform

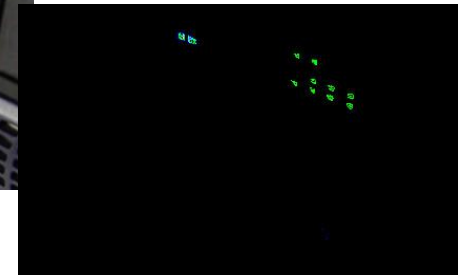
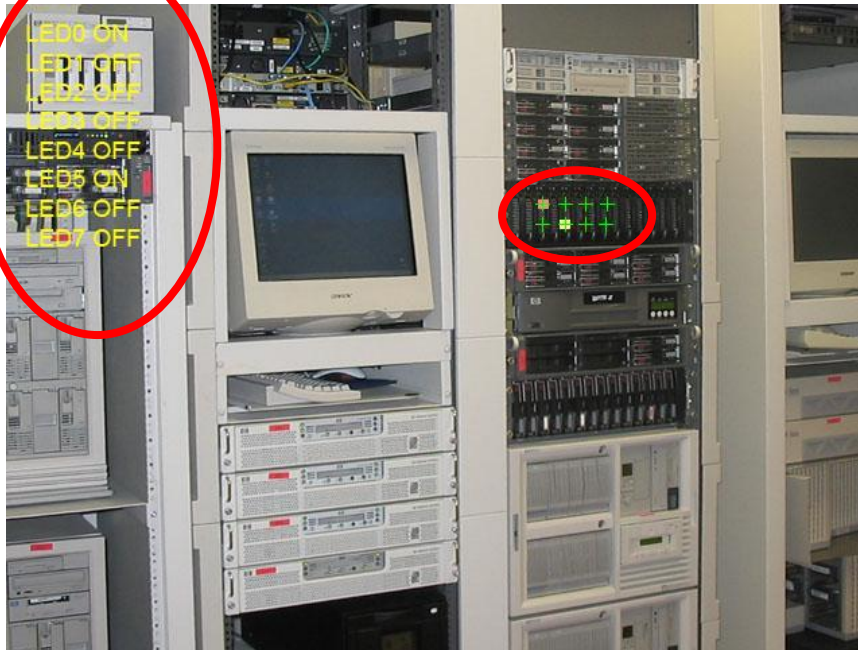
- Beagleboard-XM embedded SBC
- Sensors
 - temperature
 - humidity
 - web camera
 - power meters
 - microswitches to windows/doors
- Information from the web
 - weather status
 - weather forecast



3.25"×
3.25"

Processing the camera pictures in the Cloud

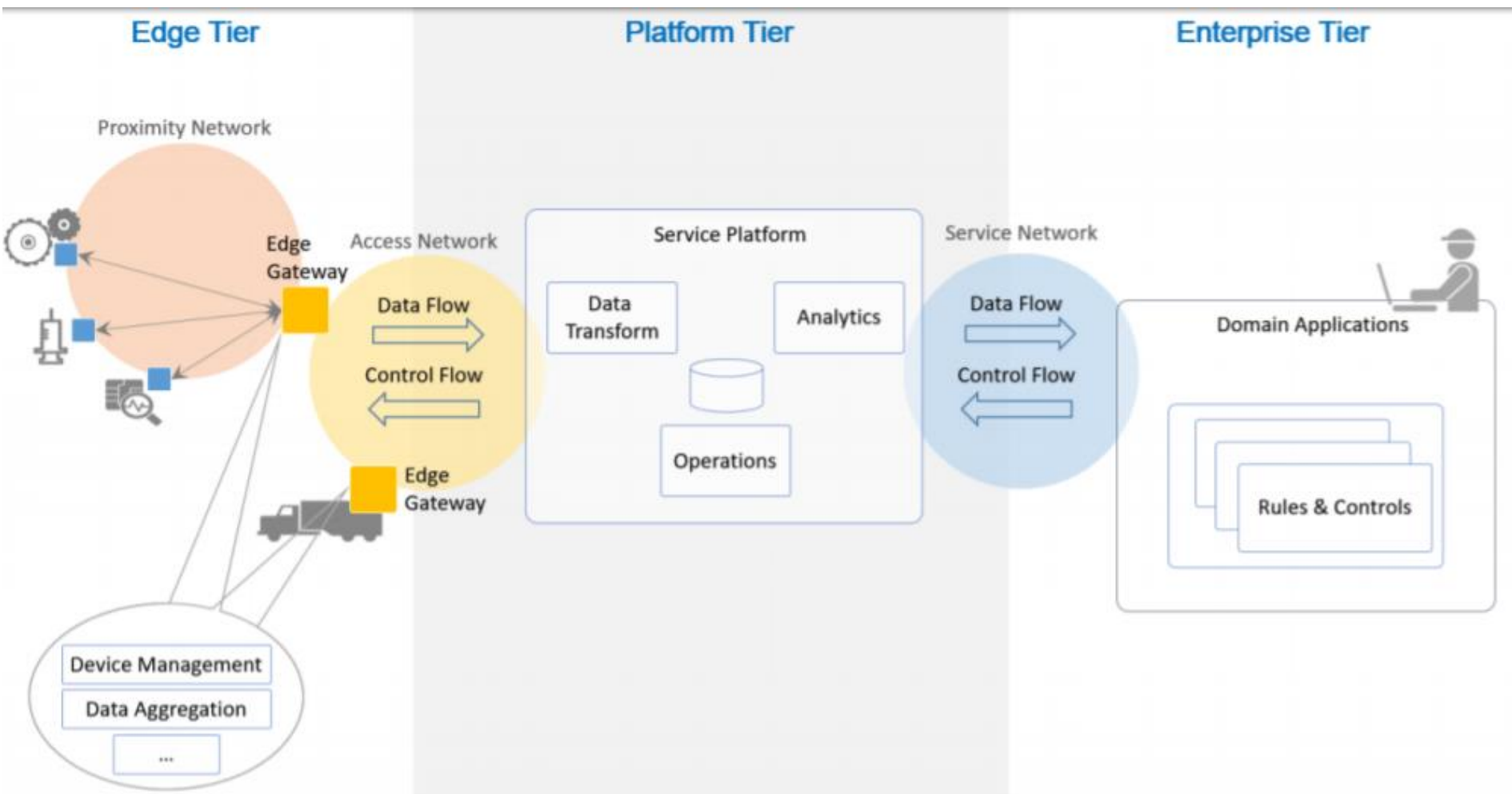
- Motion JPEG stream
 - available on the Internet
- Threshold



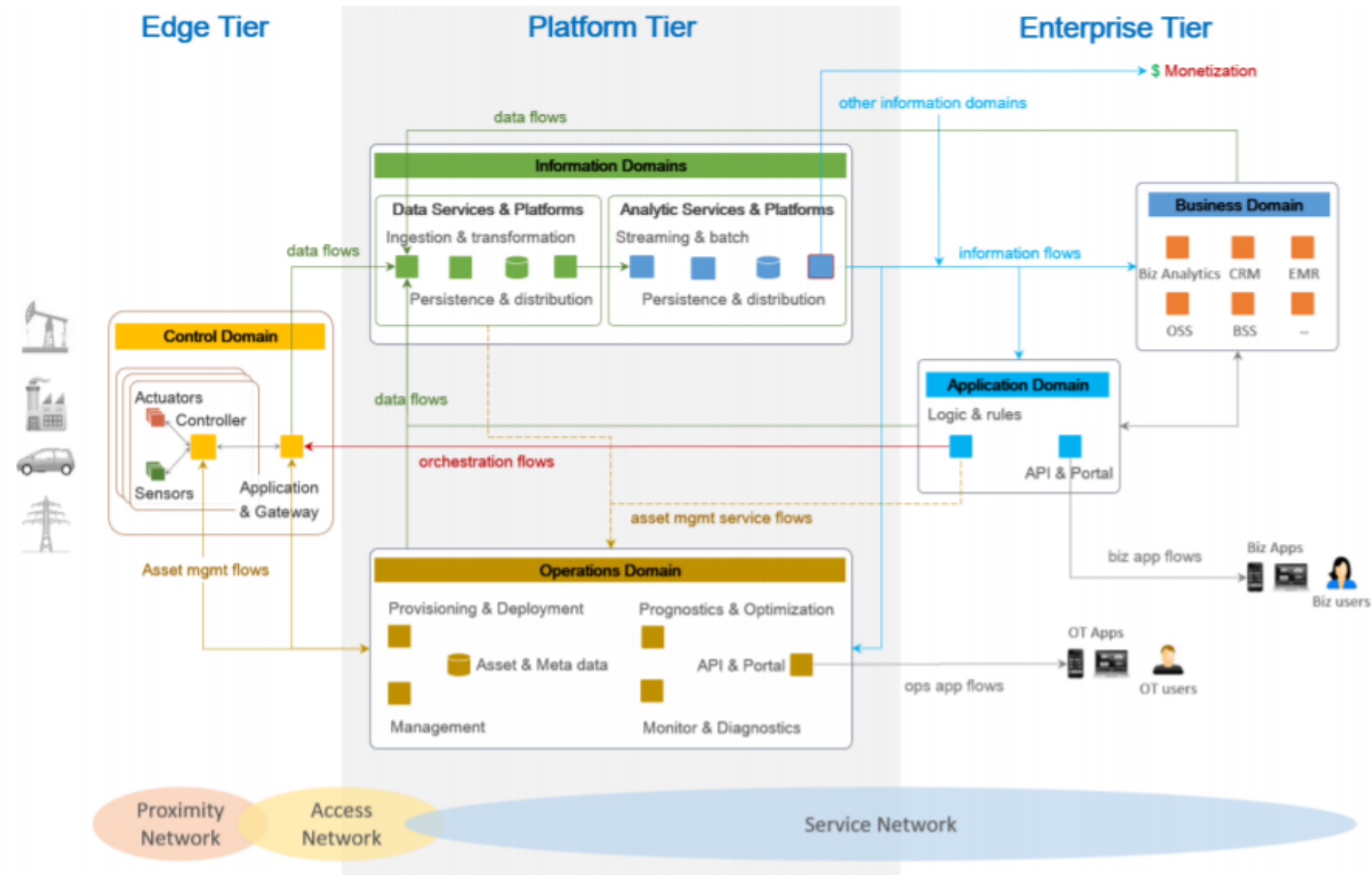
- Virtualization for sensor drivers

Systematic Approach: architecture frameworks

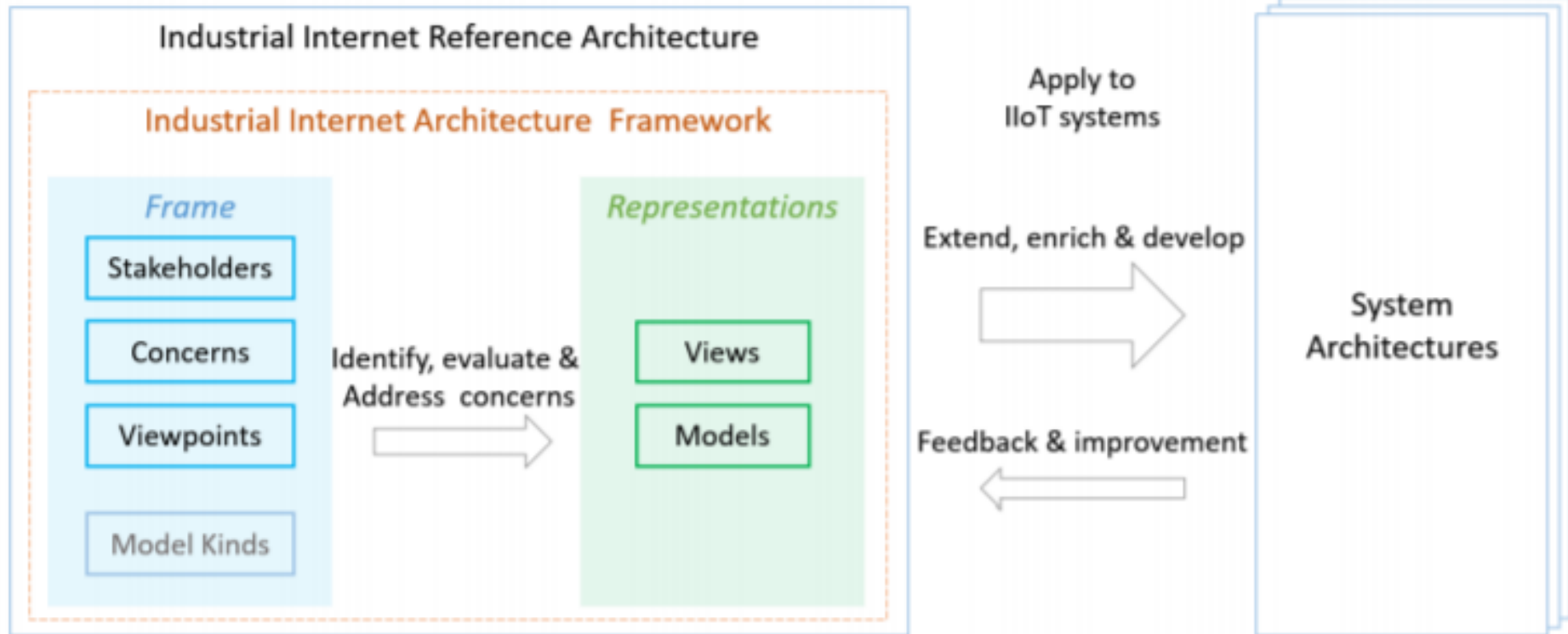
Tiers in Industrial Internet -example



Industrial Internet Consortium 3-tier

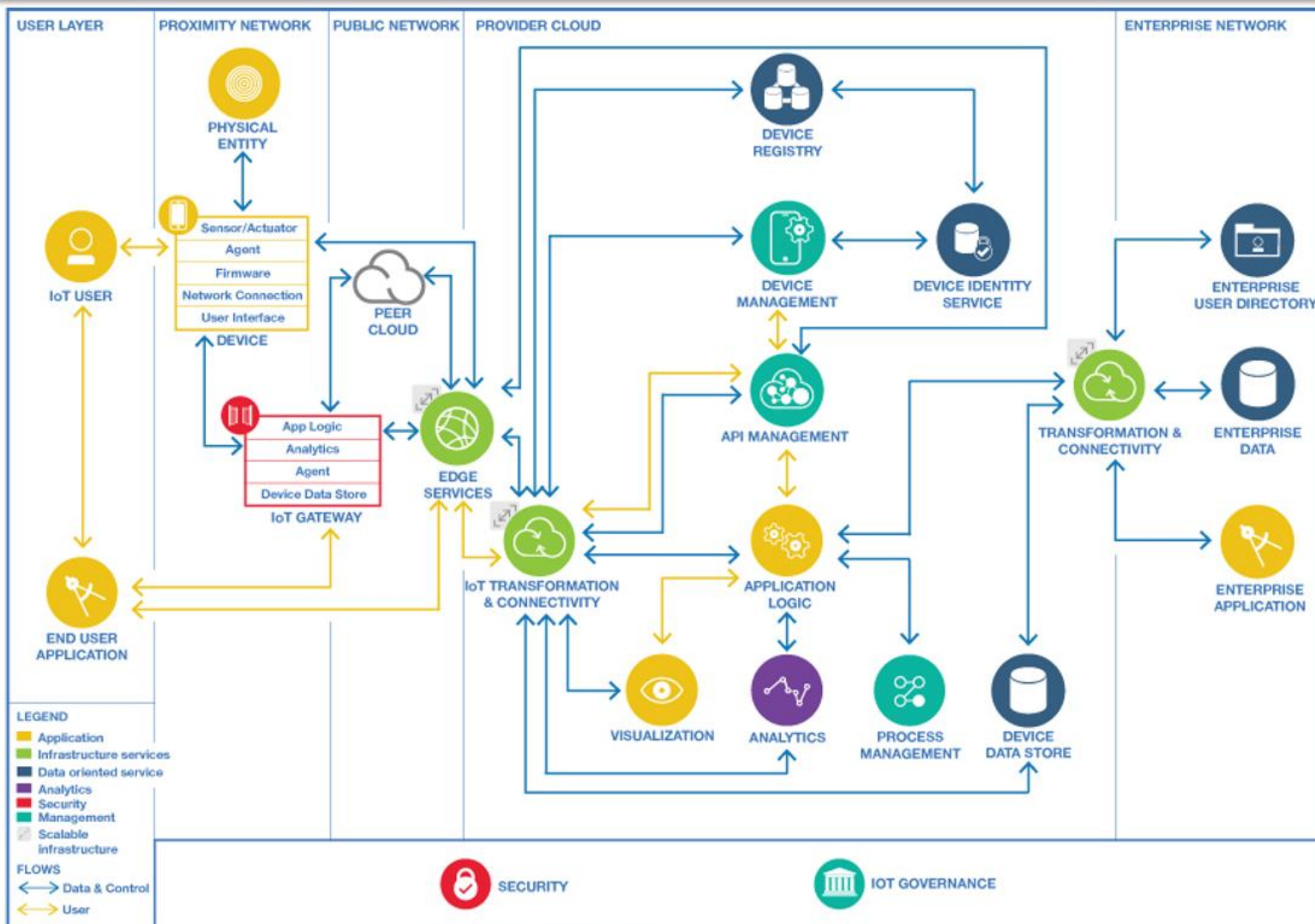


Reference Architecture and application



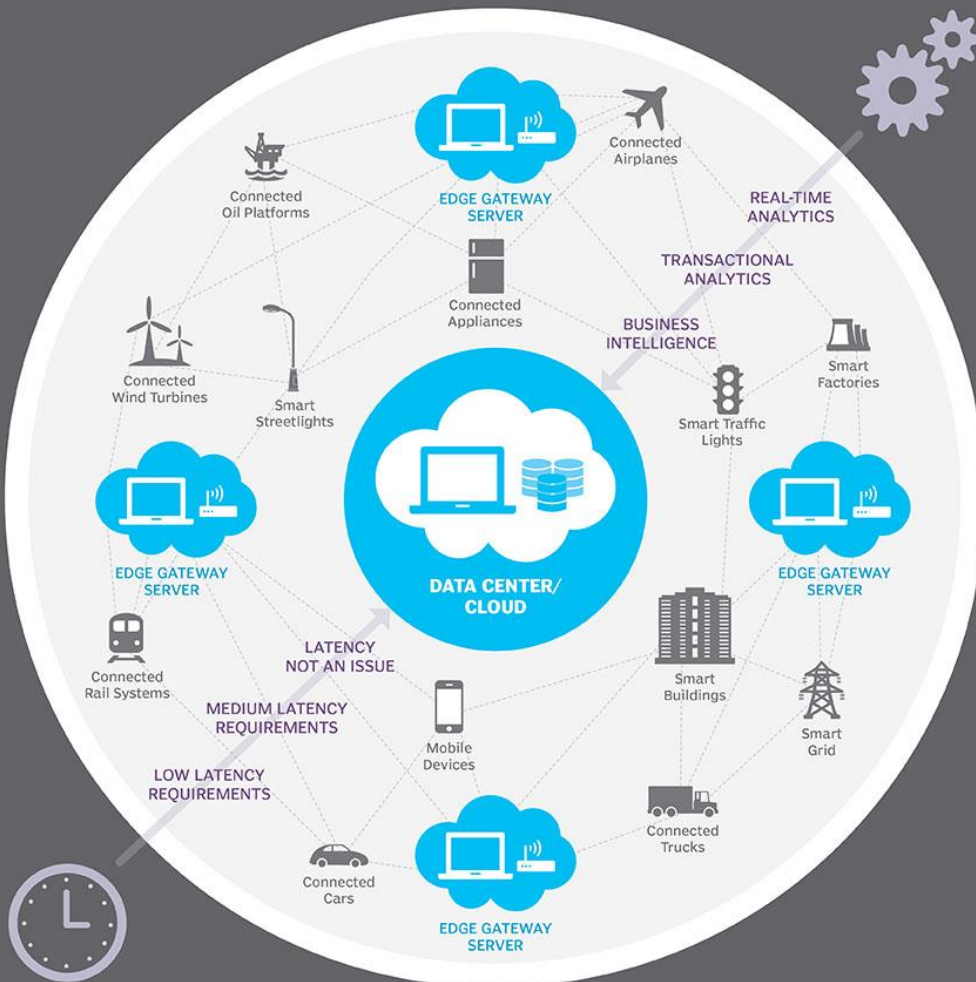
Cloud Customer Architecture for IoT

- <http://www.cloud-council.org/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf>



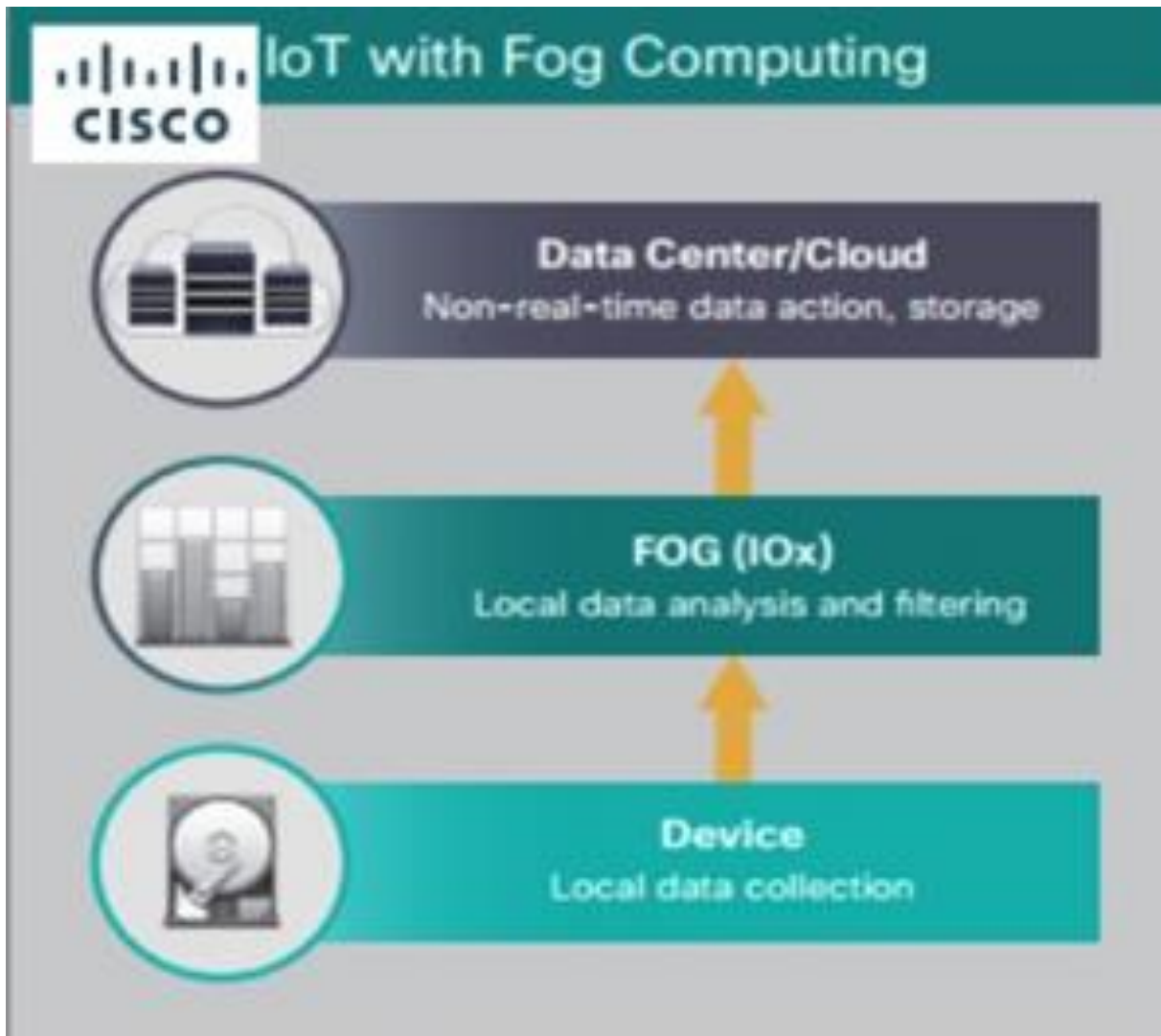
Edge computing

Edge Computing

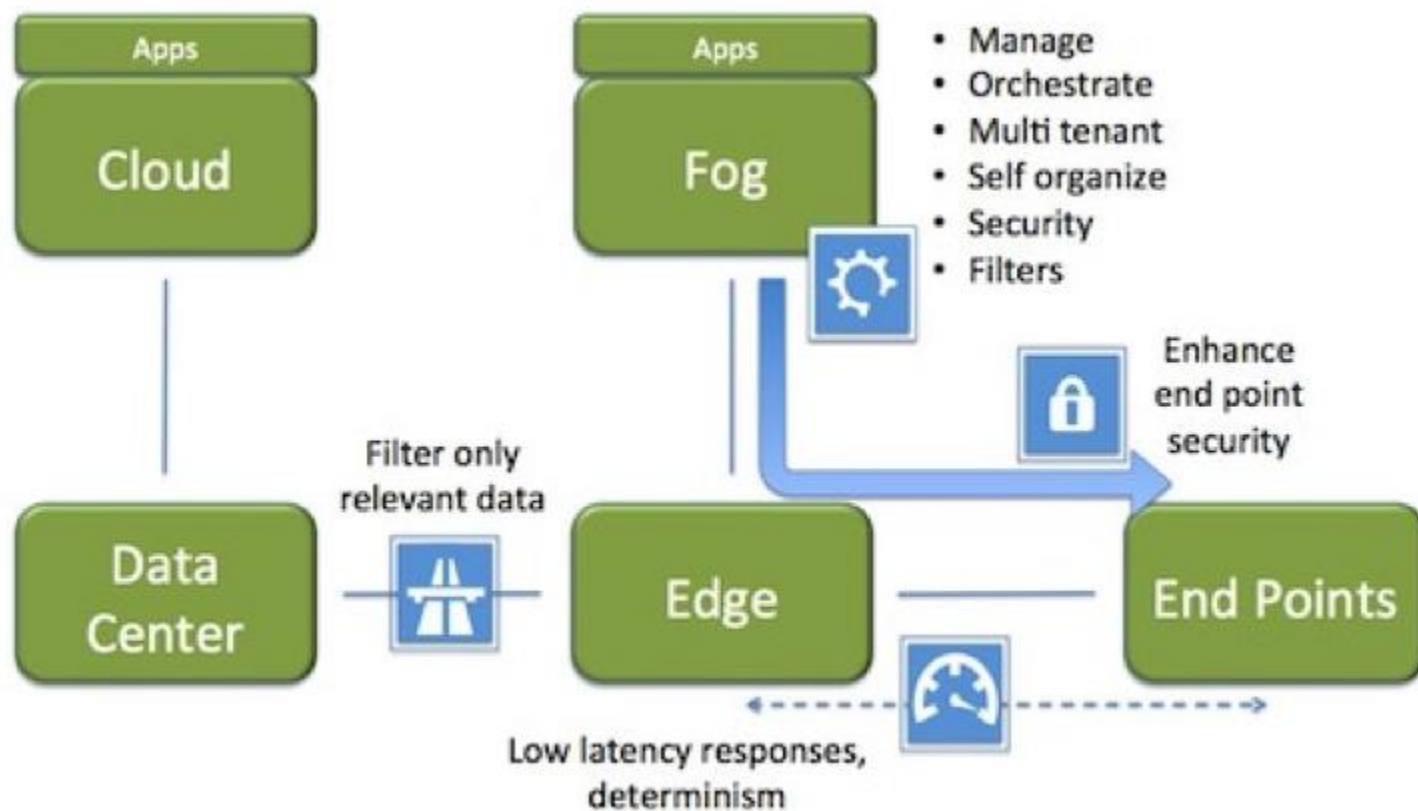


- Techtarget:
Edge computing
definition

Fog computing



Edge and fog computing



End of lecture 2

Frameworks

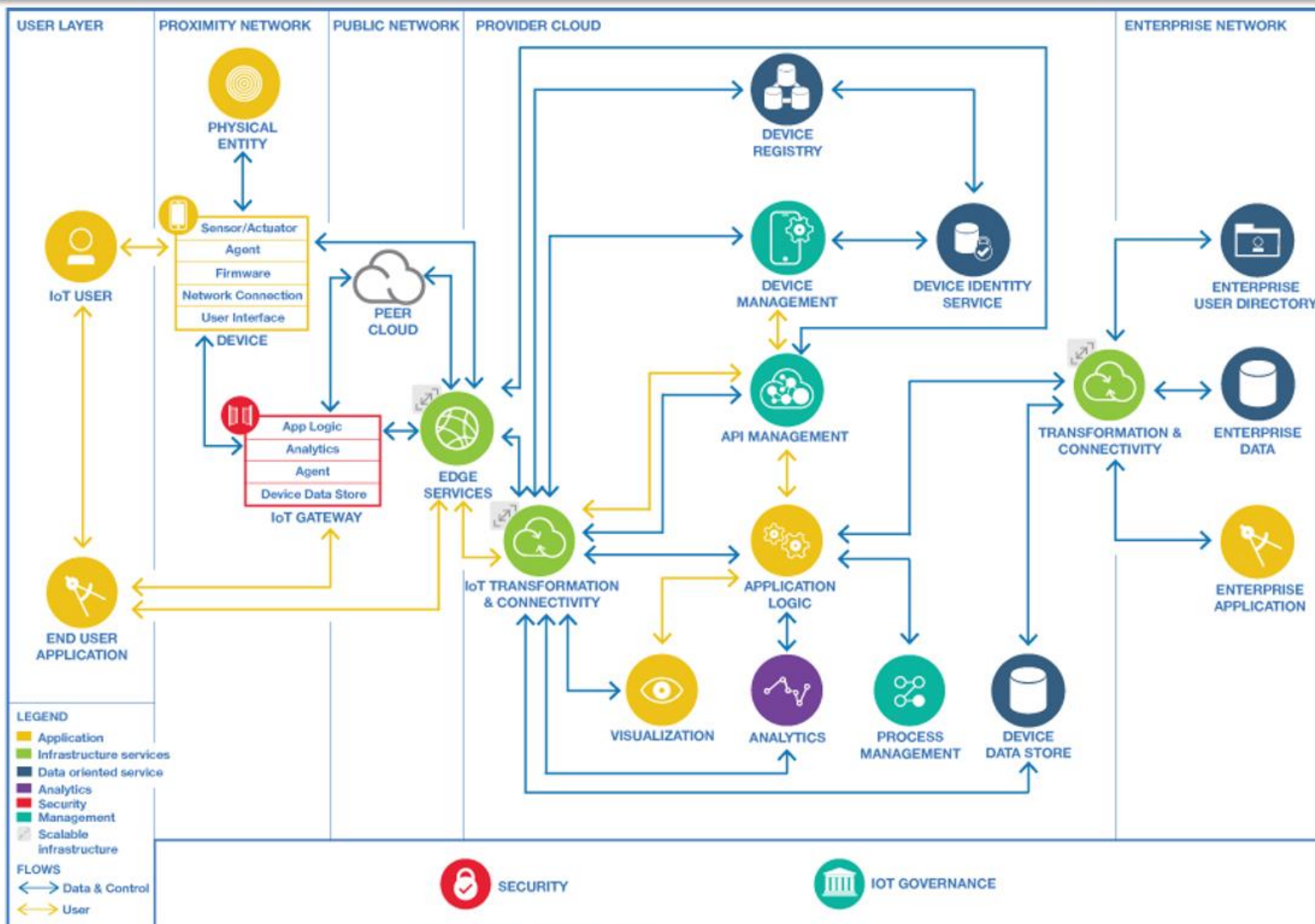
Cost impact estimation

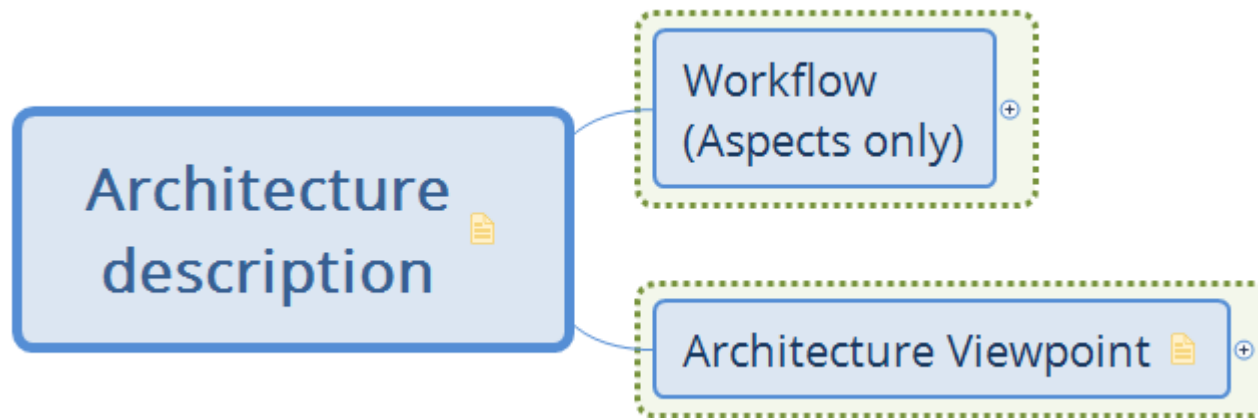
# of System Requirements	Easy	Nom.	Diff.
# New	0,5	1,0	5,0
# Design For Reuse	0,7	1,4	6,9
# Modified	0,3	0,7	3,3
# Deleted	0,3	0,5	2,6
# Adopted	0,2	0,4	2,2
# Managed	0,1	0,2	0,8

1. Quality and stability
Modification: ~ **70% !**
2. Requirement set
complexity reduction
 - 5 similar problems
 - Separate solution:
 $5 \times \text{New} = \mathbf{500\%}$
 - Global solution:
 $1 \times \text{Reuse} + 4 \times \text{Adopt}$
 $= 140\% + 4 * 70\%$
 $= \mathbf{420\%}$

Cloud Customer Architecture for IoT

- <http://www.cloud-council.org/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf>

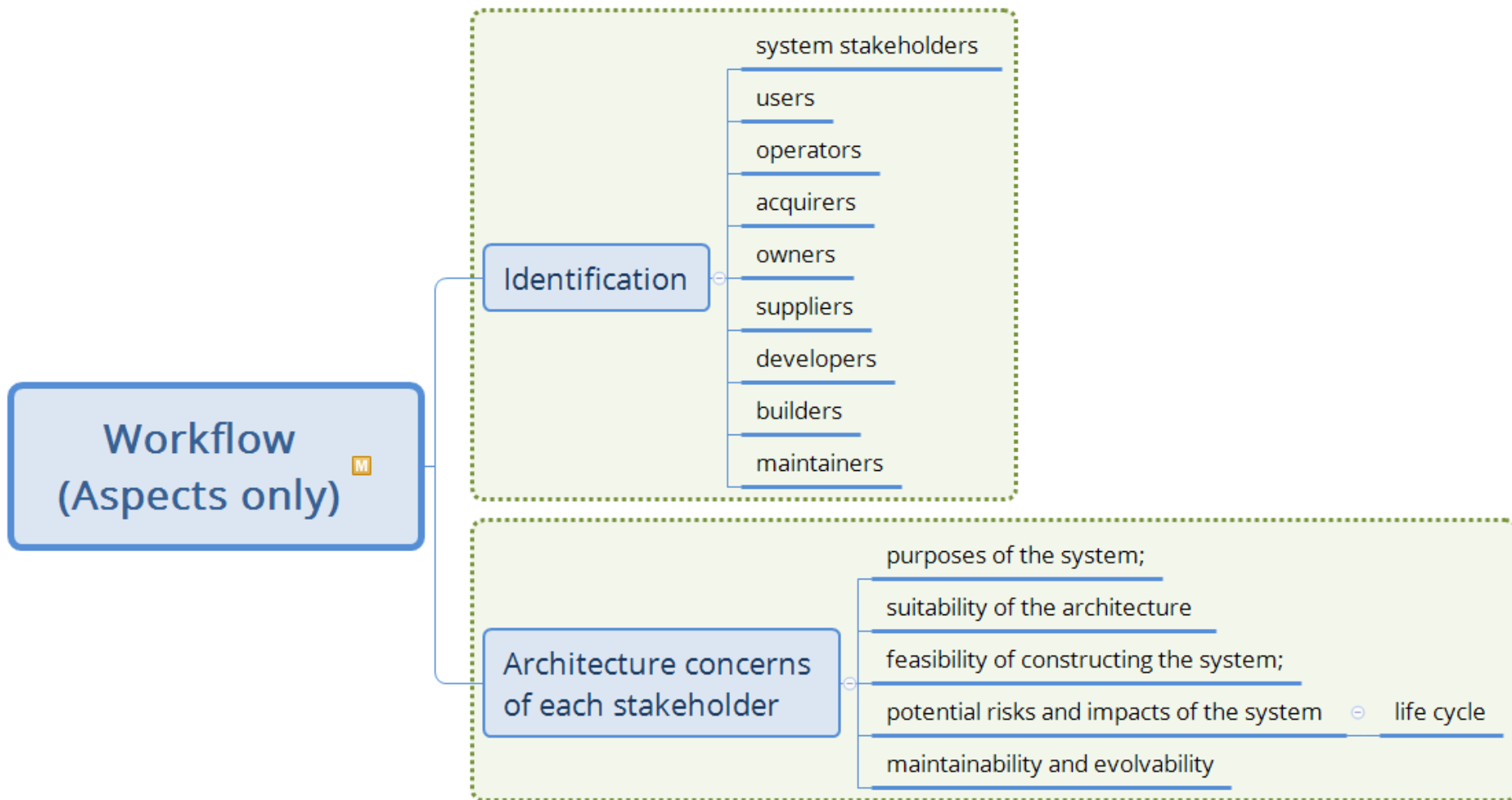




ISO/IEC/IEEE 42010:2011,
Systems and software engineering — Architecture description,
<http://www.iso-architecture.org/ieee-1471/>

Architecture description

Towards architecture design



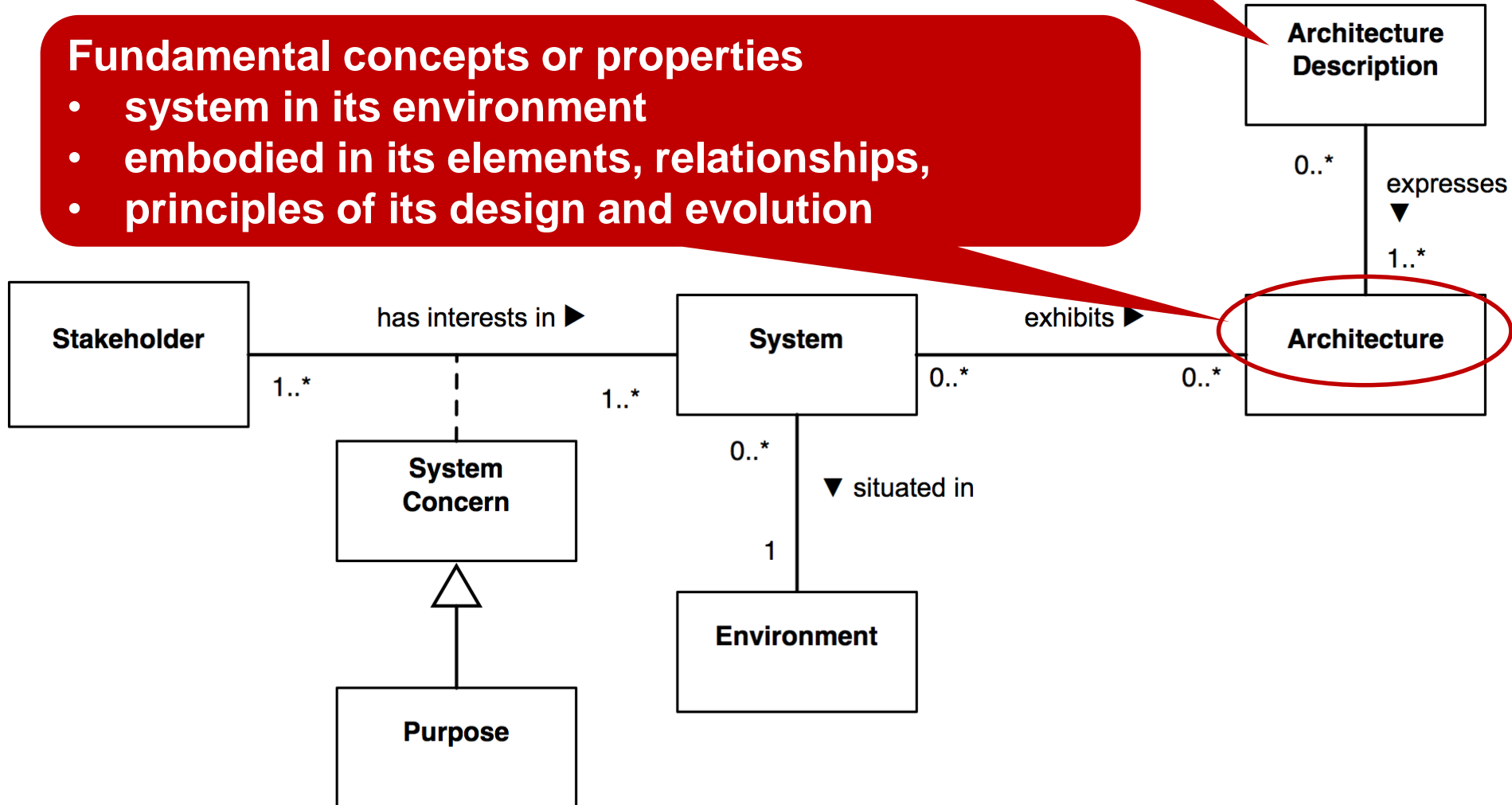
Conceptual model (=metamodel)

A Conceptual Model of AD

- Understand
- Analyze
- Compare
- “blueprints”

Fundamental concepts or properties

- system in its environment
- embodied in its elements, relationships,
- principles of its design and evolution

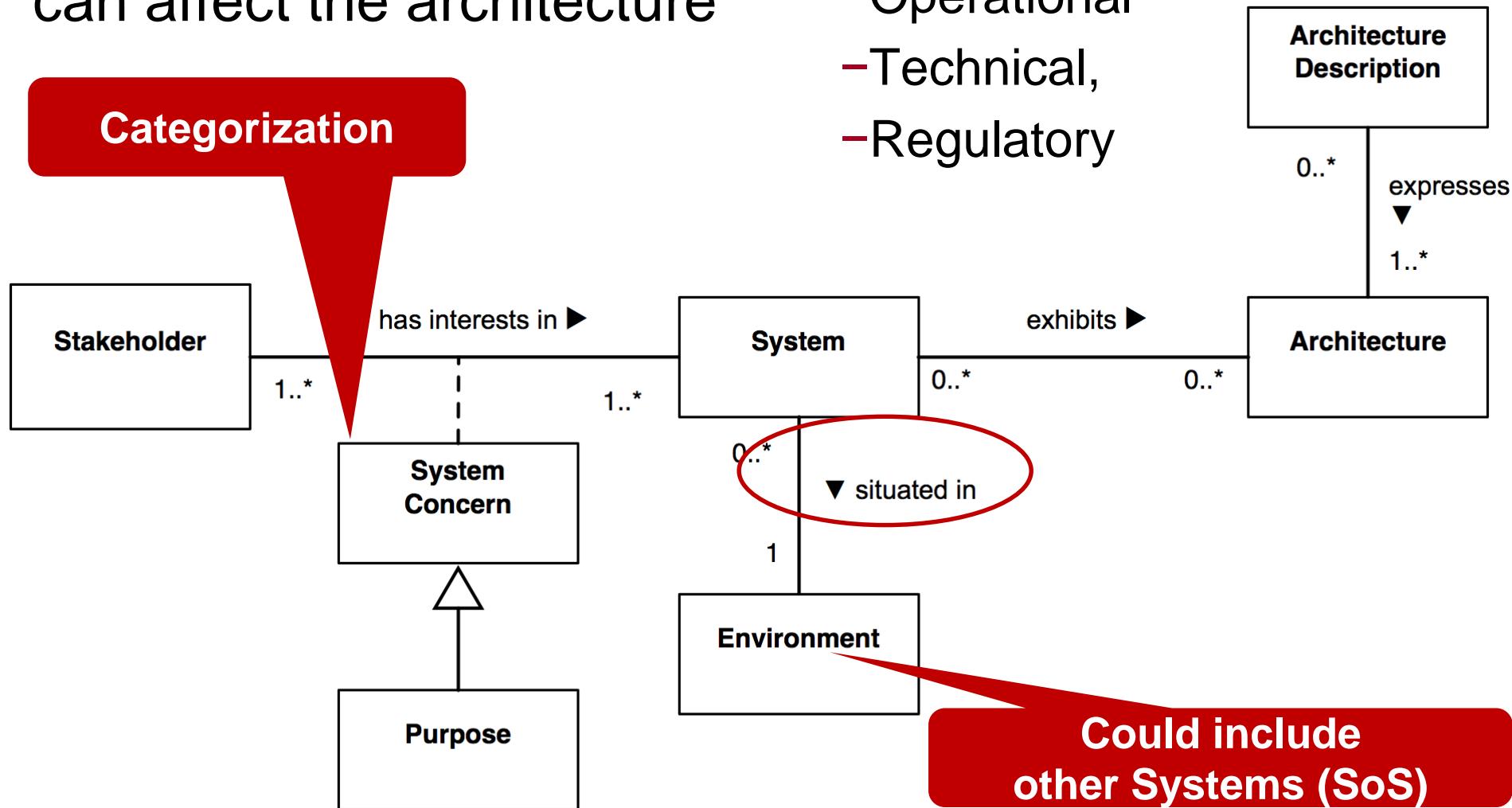


System-and environment

Influencing factors which can affect the architecture

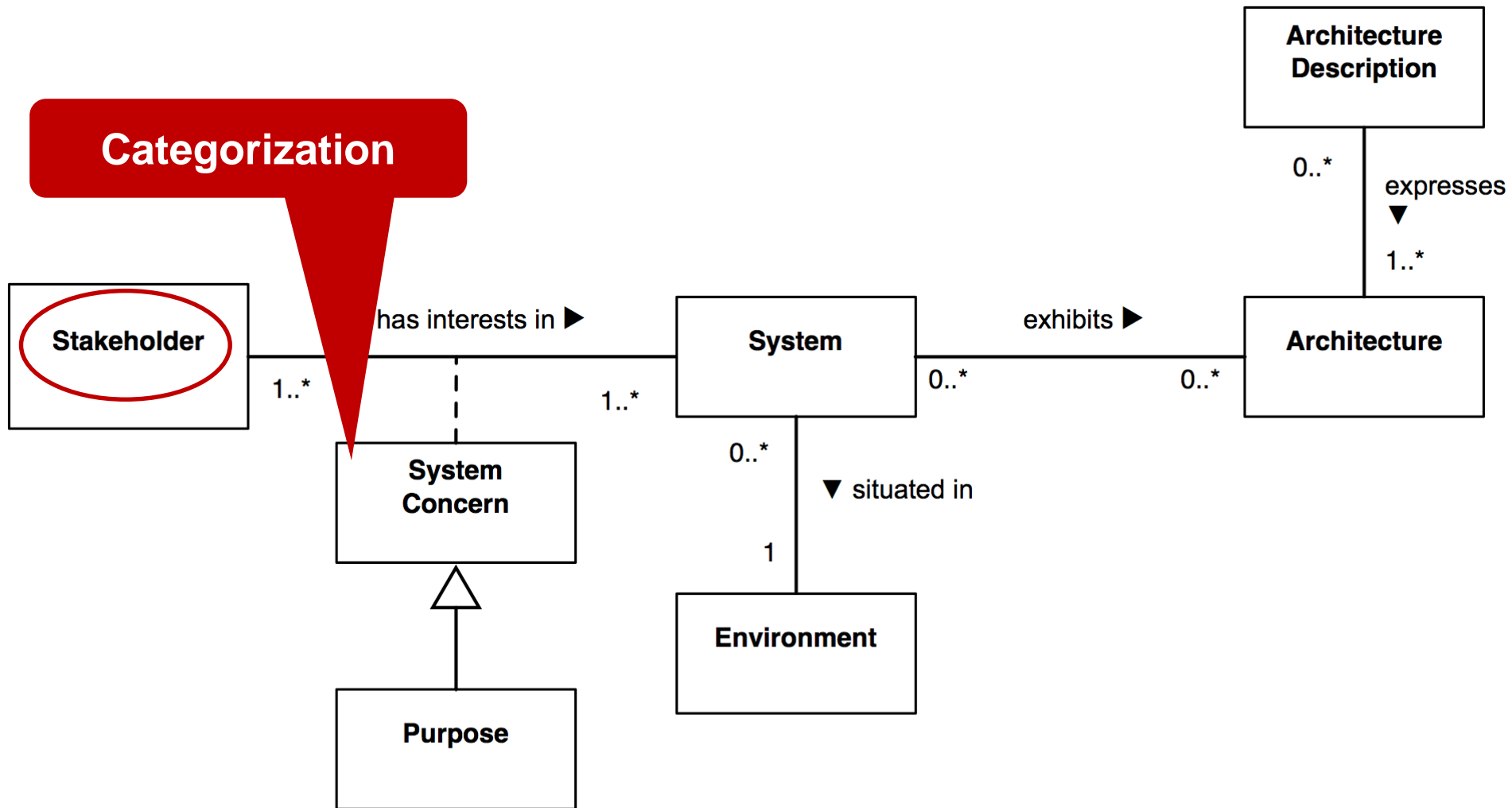
- Developmental
- Operational
- Technical,
- Regulatory

Categorization

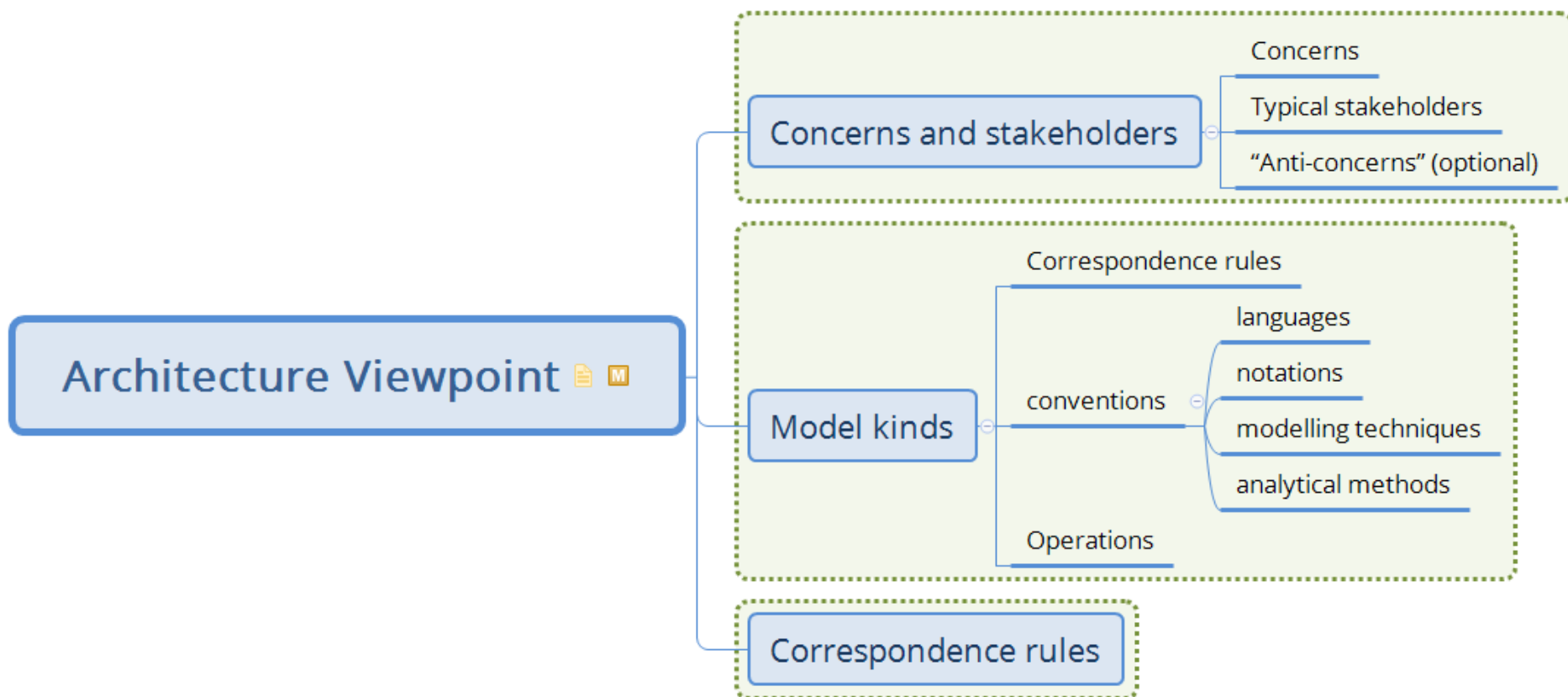


Context

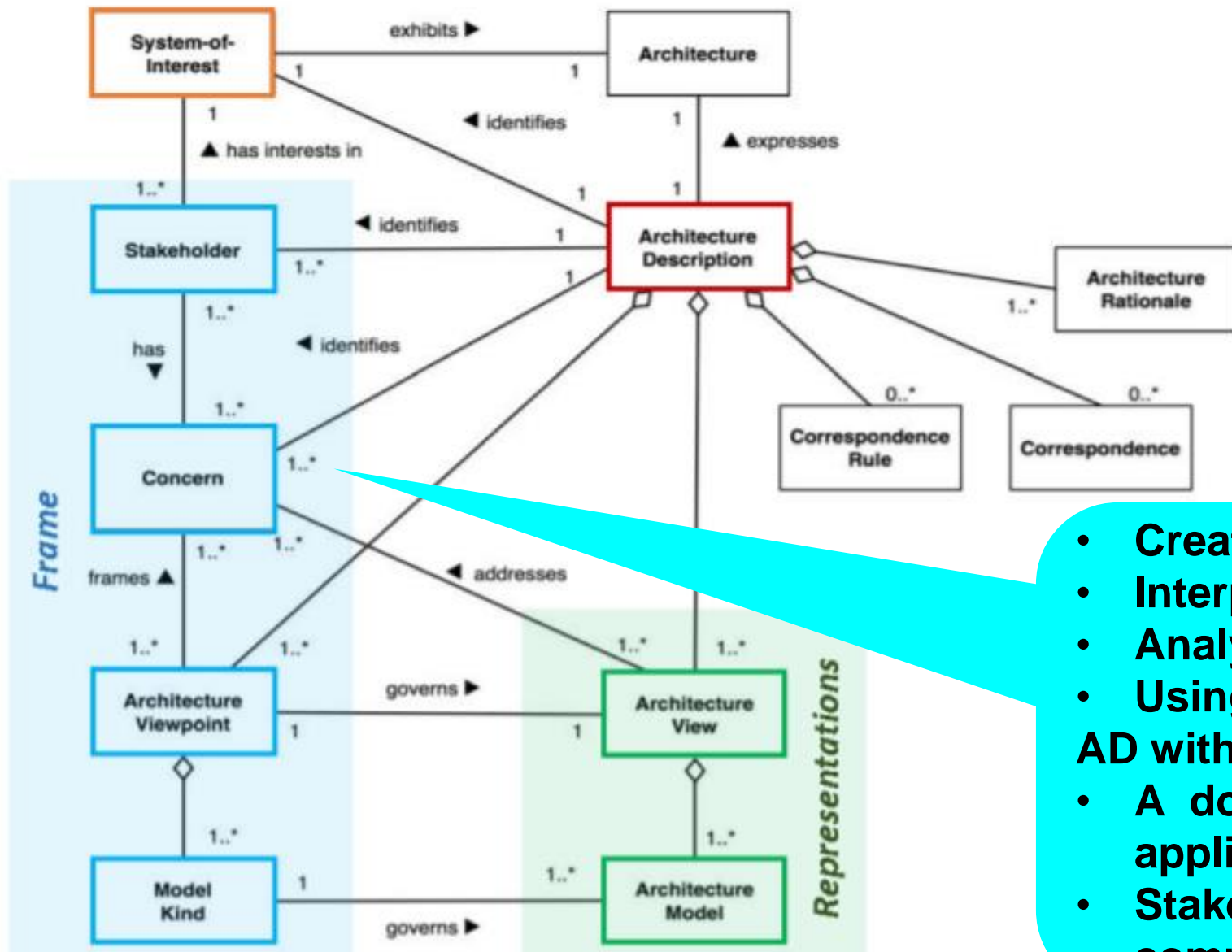
Categorization



Viewpoint



Core of Architecture Description

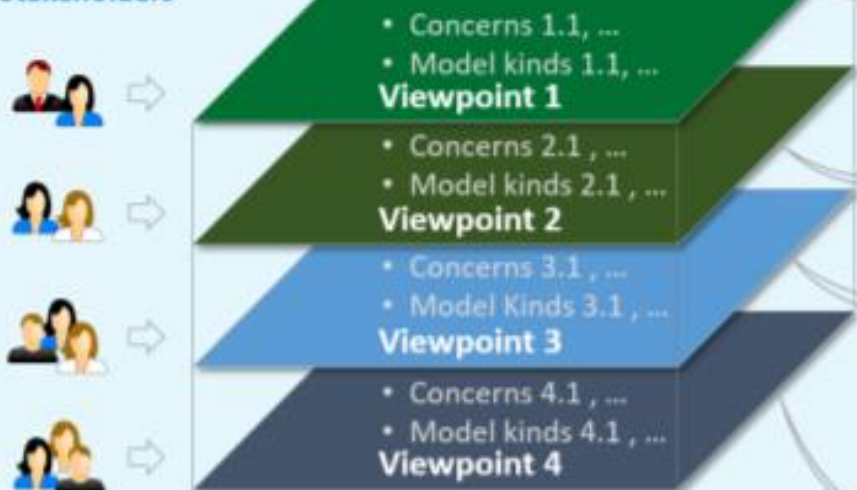


- Creating
- Interpreting
- Analyzing
- Using AD within
- A domain of application
- Stakeholder community

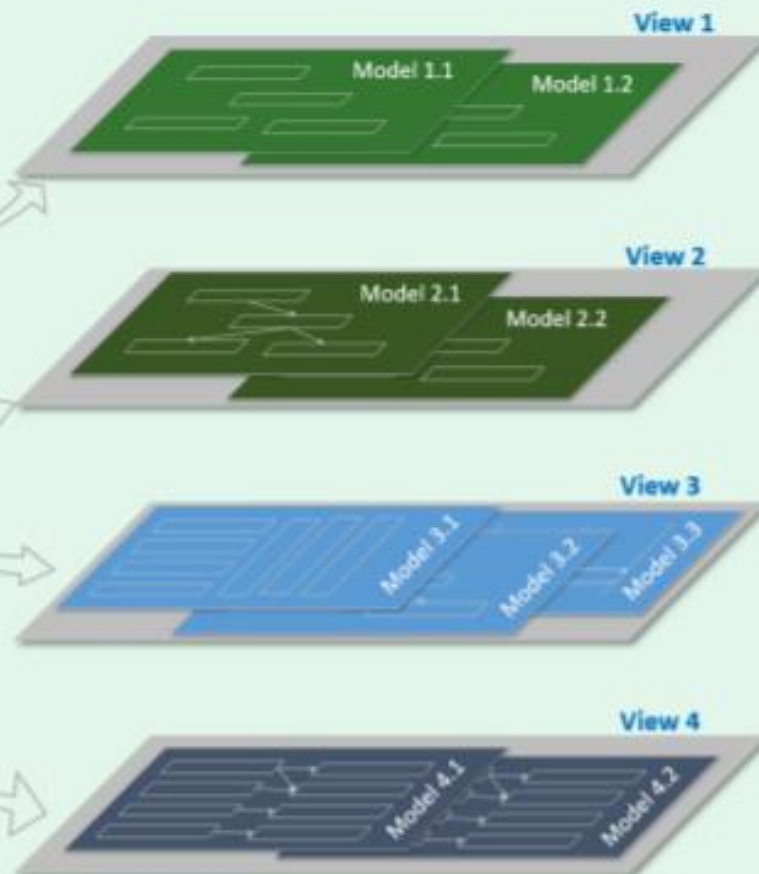
Frame and representation

Architecture Frame

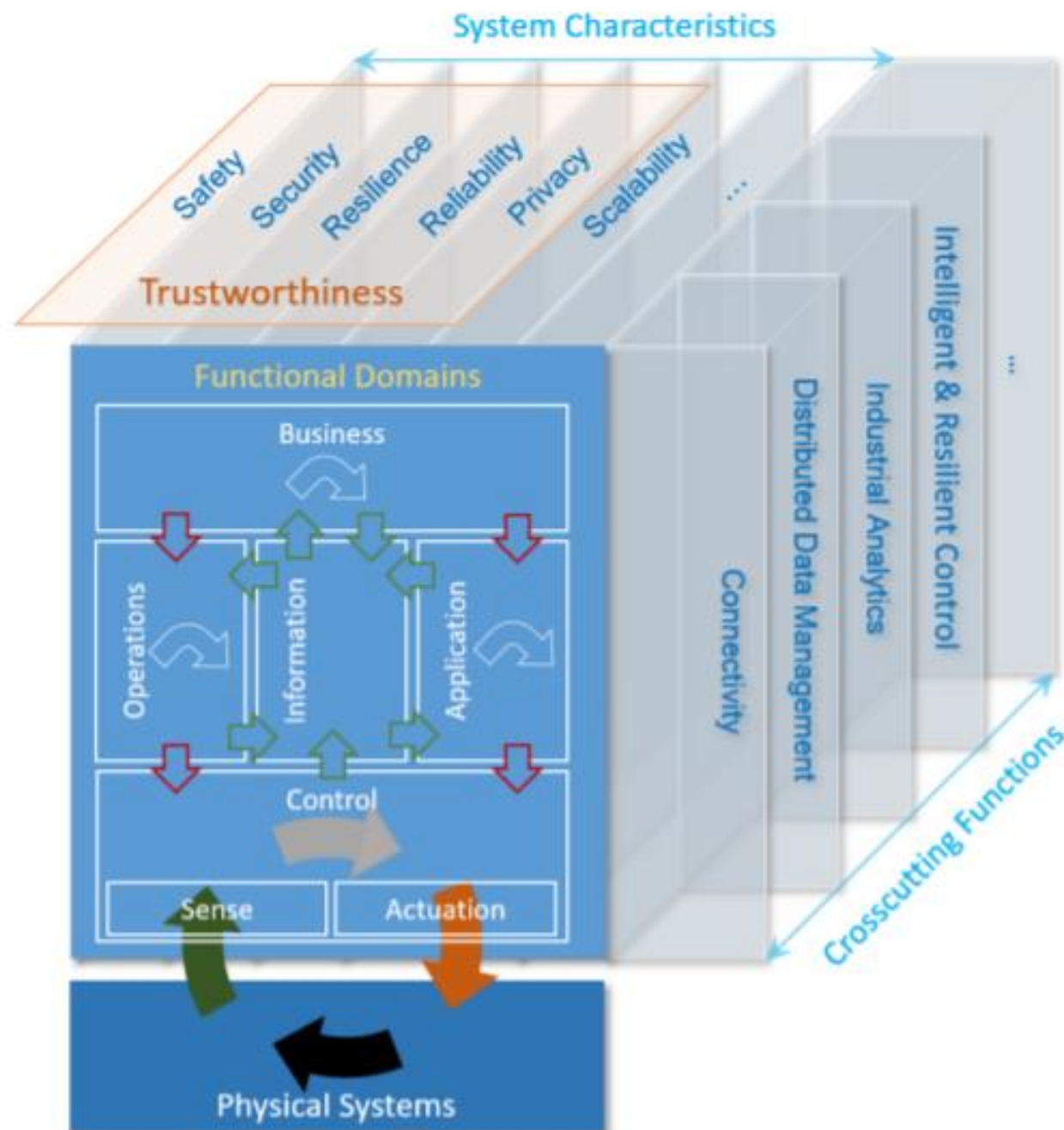
Stakeholders



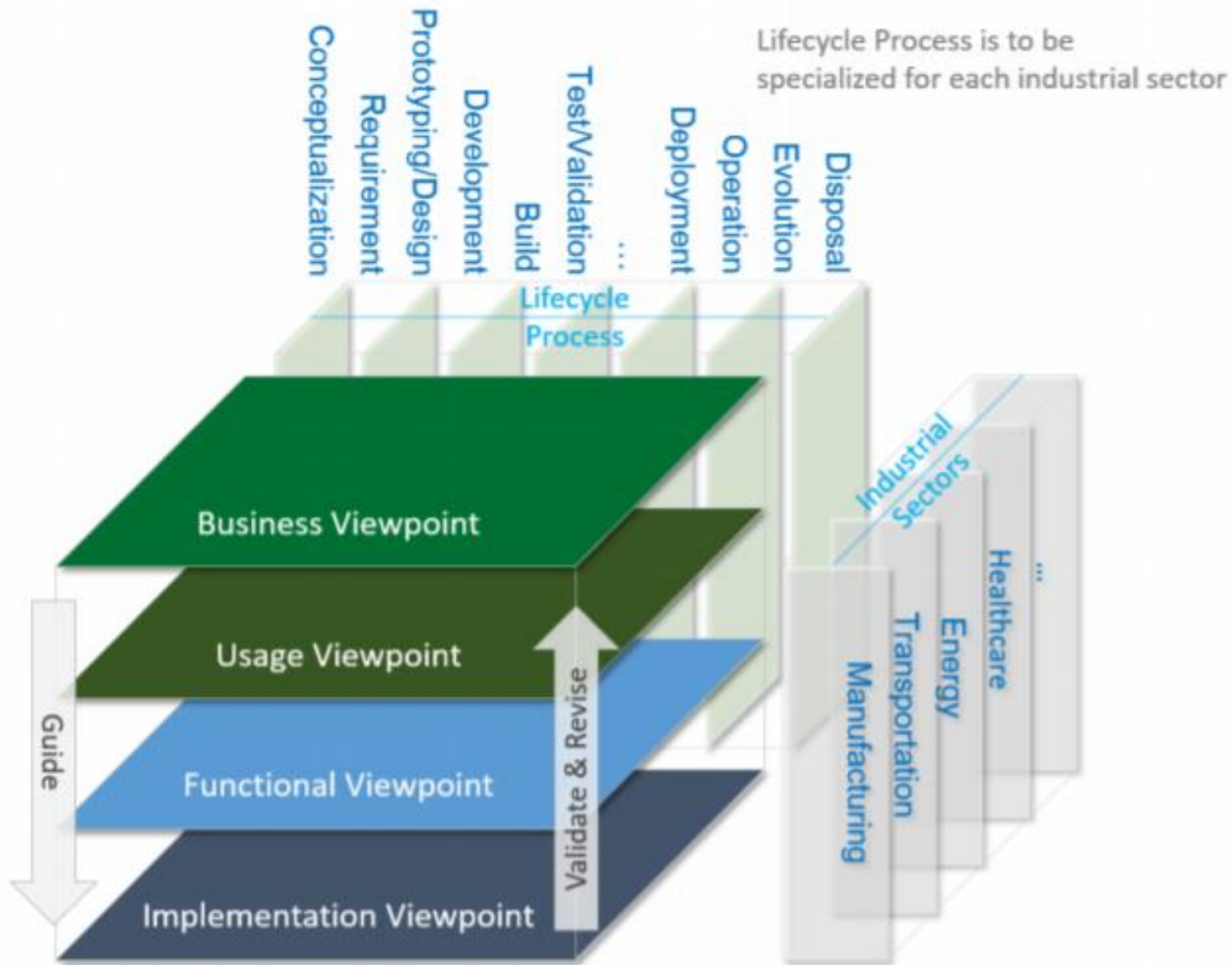
Architecture Representations



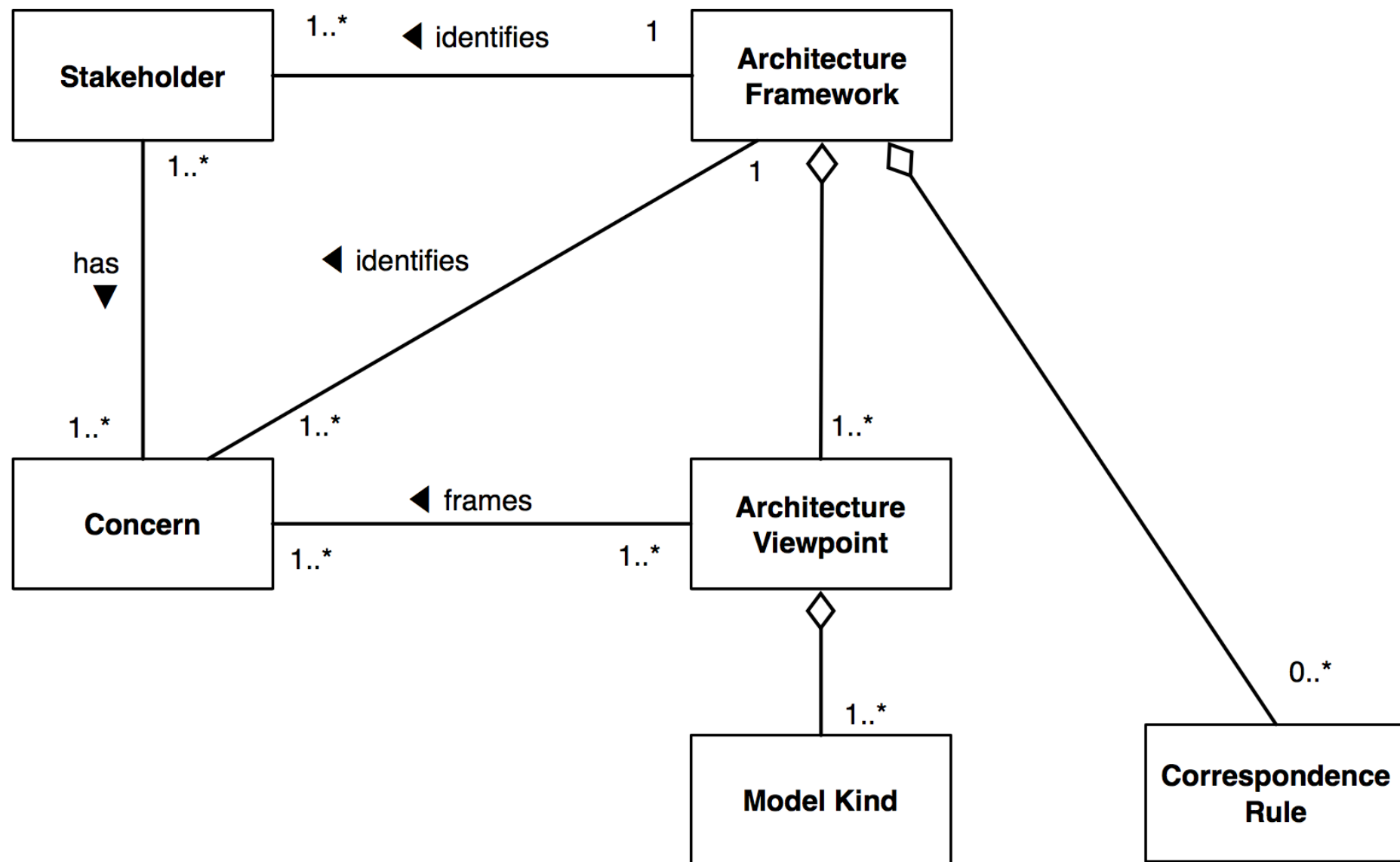
Viewpoints



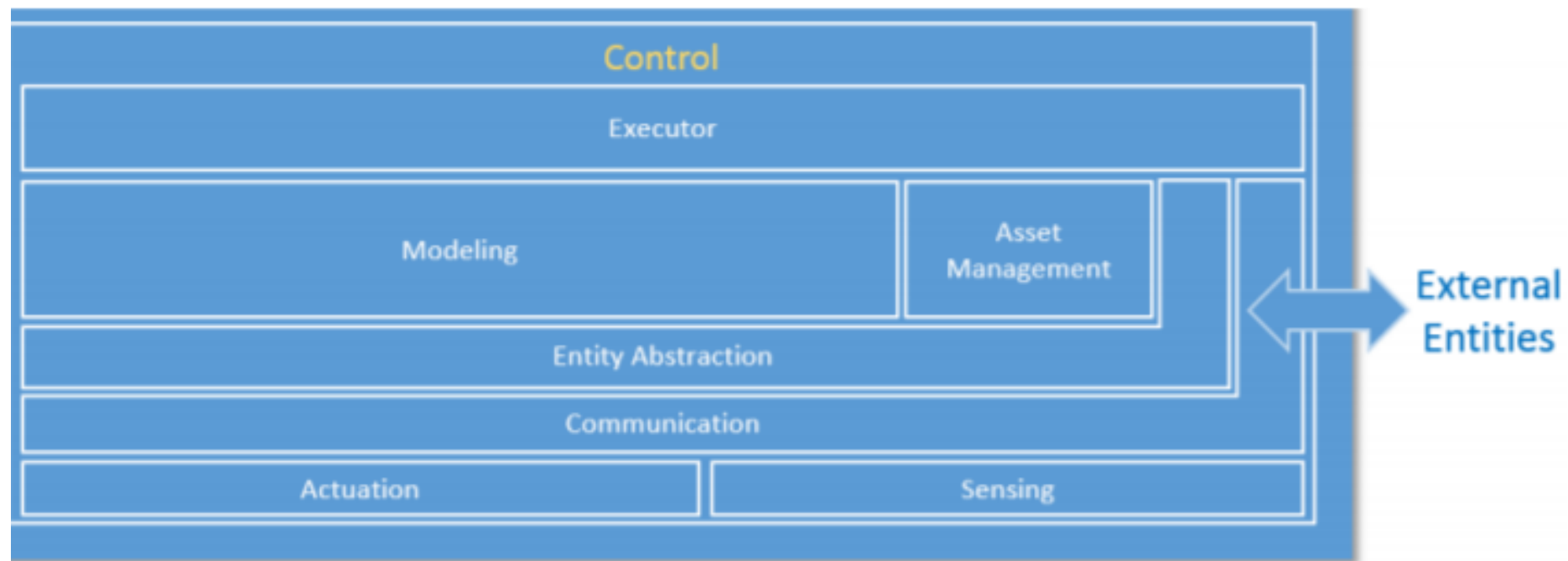
IIC viewpoints



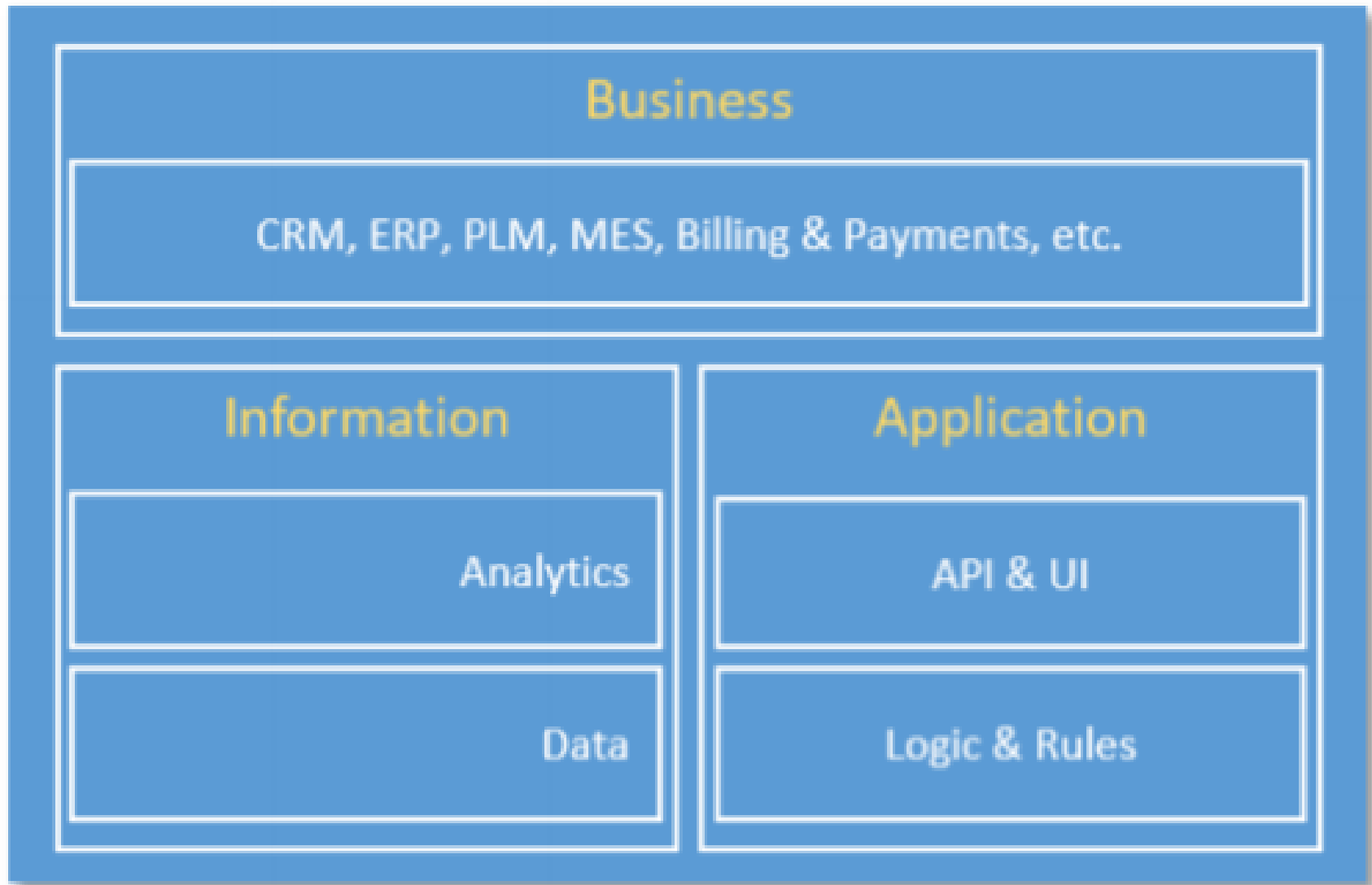
Architecture Frameworks and ADL



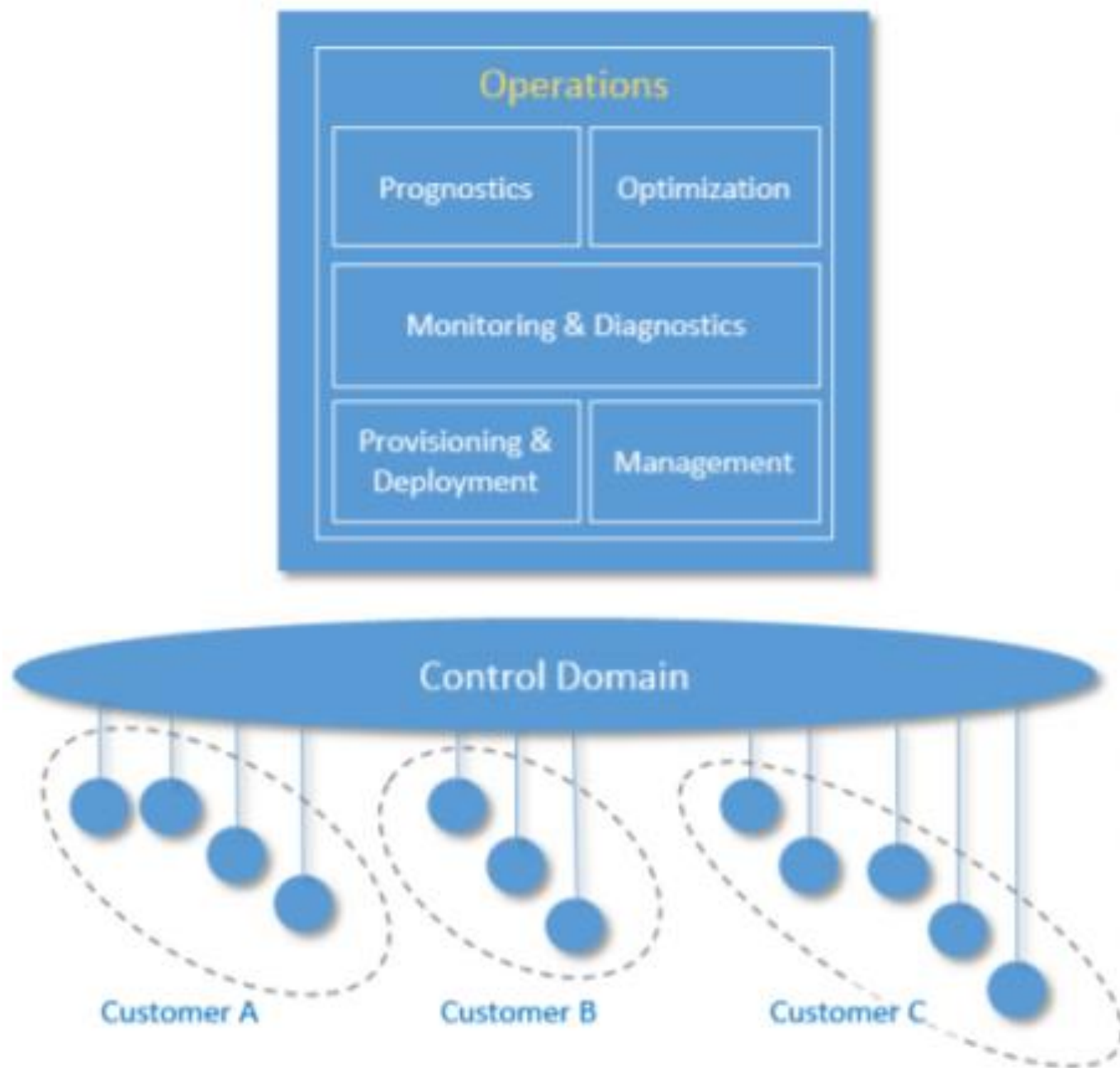
Control



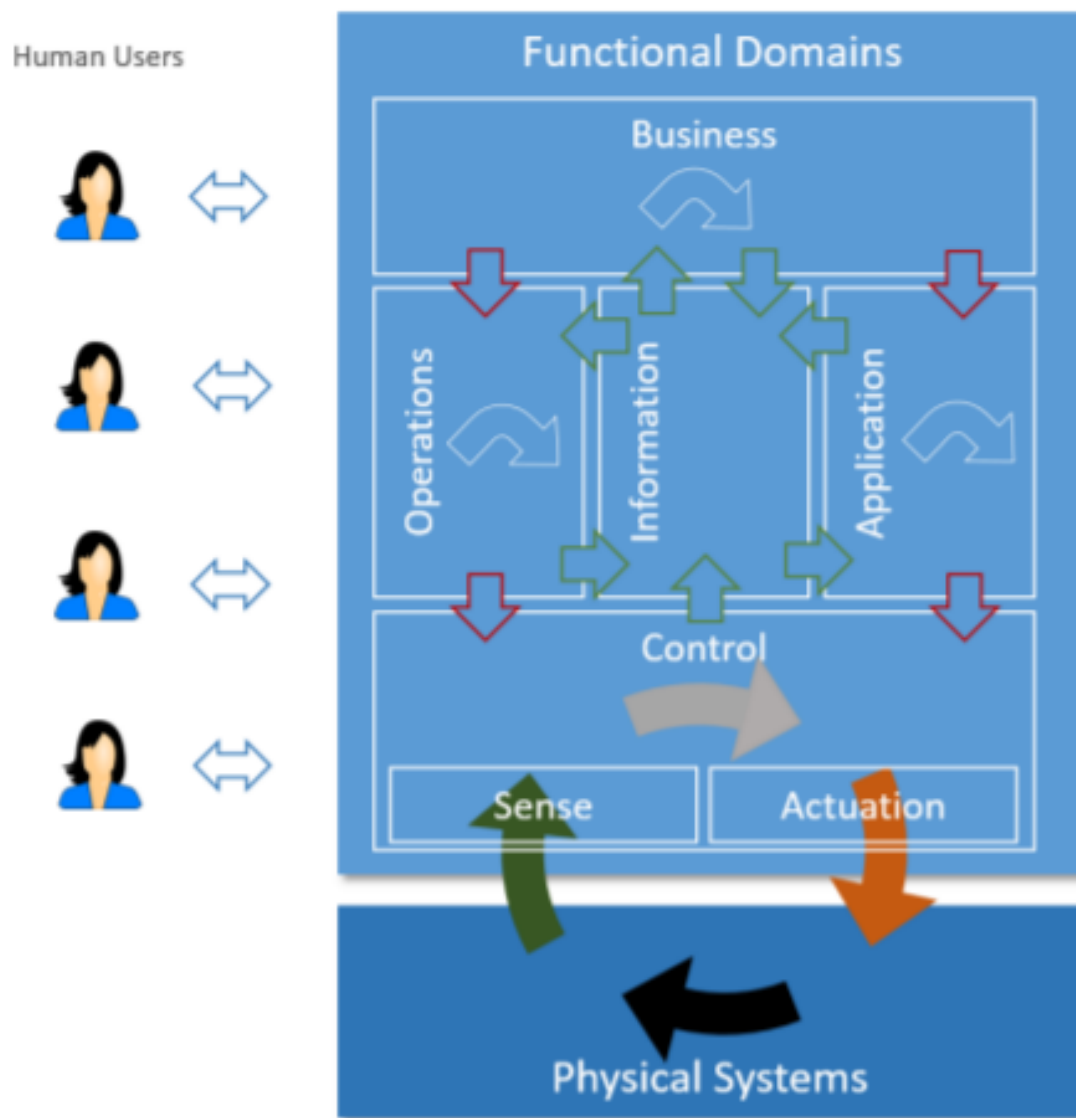
Business



Operation



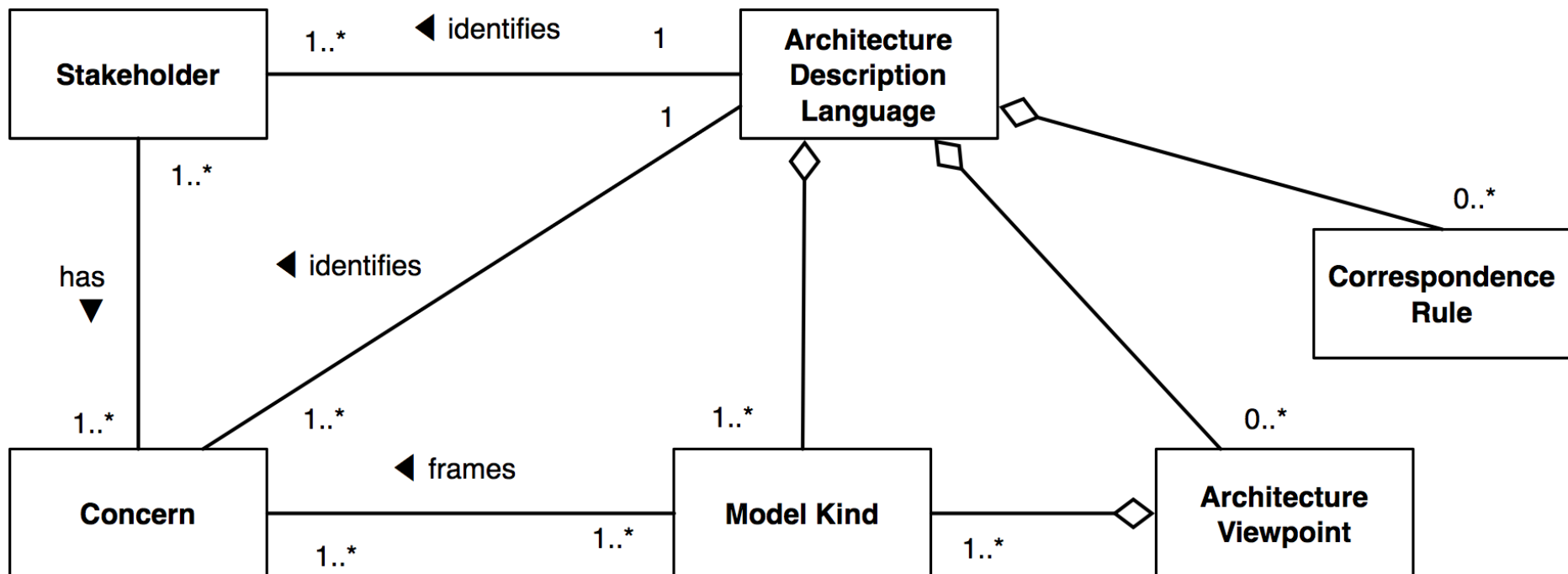
Domains



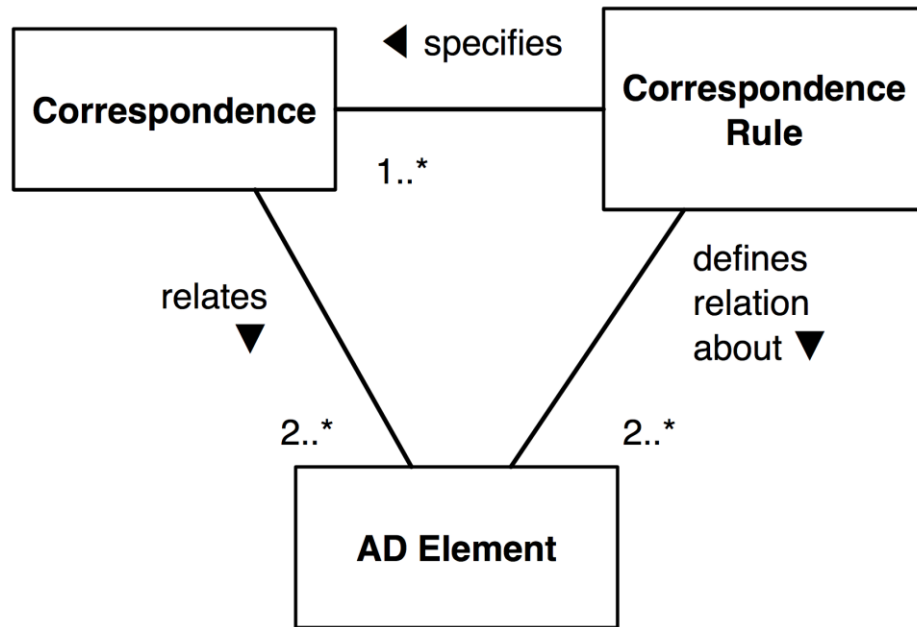
Architecture Description Language

- *Any form of expression* for use in AD
 - single Model Kind,
 - single viewpoint or
 - multiple viewpoints.

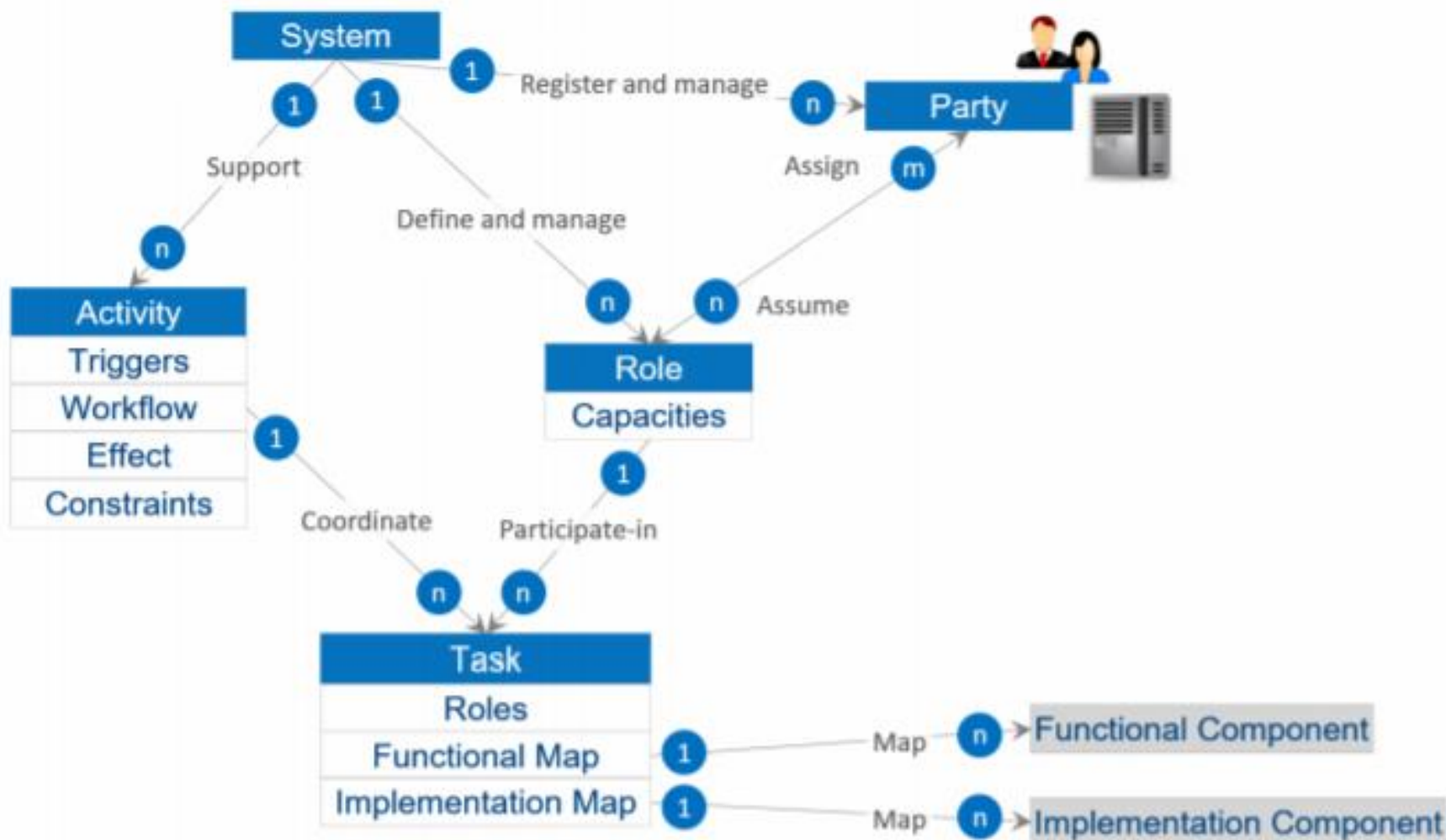
- Examples of ADLs:
 - **SysML**,
 - ArchiMate,
 - xADL.

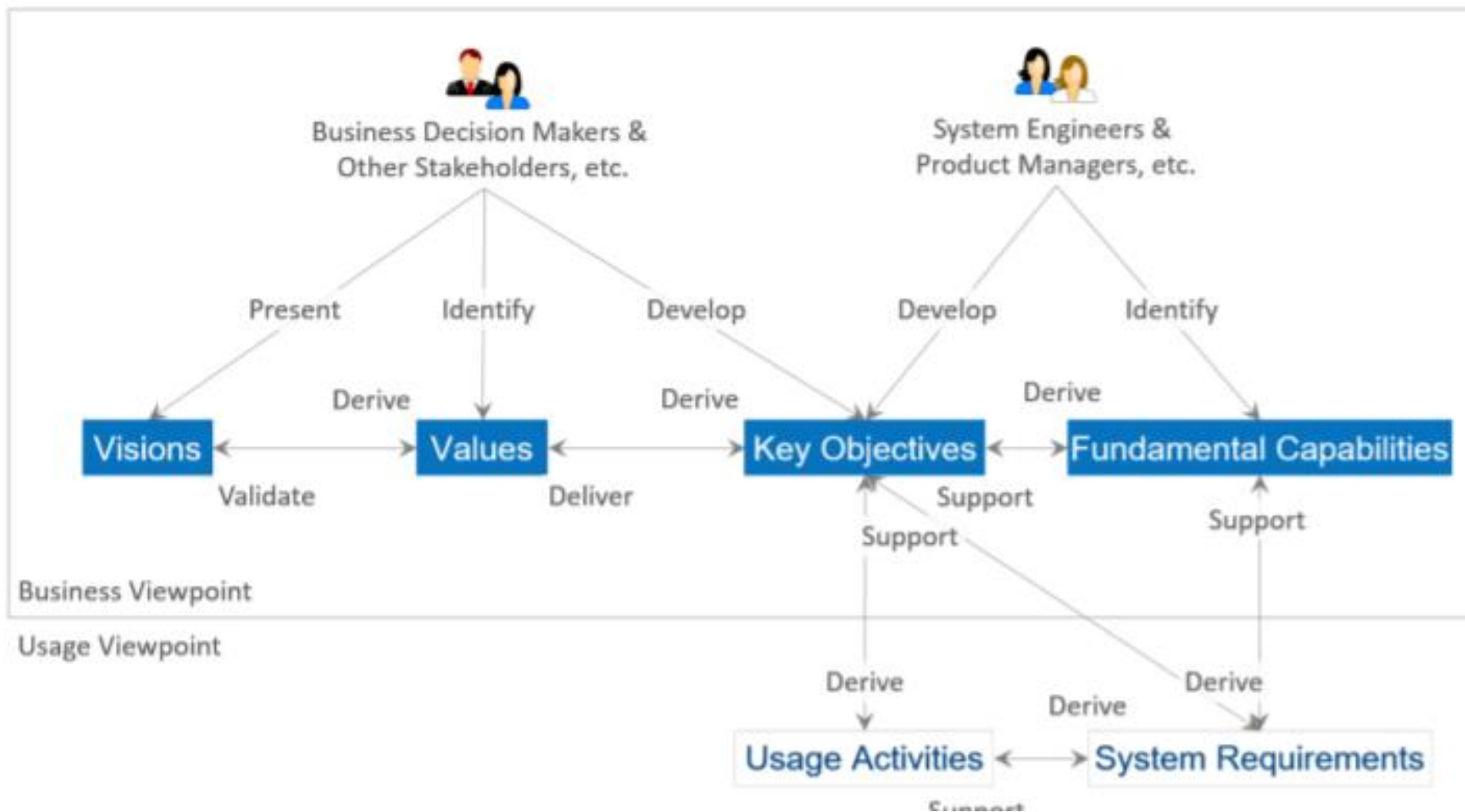


AD Elements and Correspondences



- Relationships between AD Elements.
- Express and enforce architecture relations: within or between ADs.
 - Composition
 - Refinement
 - Consistency
 - Traceability
 - Dependency
 - Constraint
 - Obligation

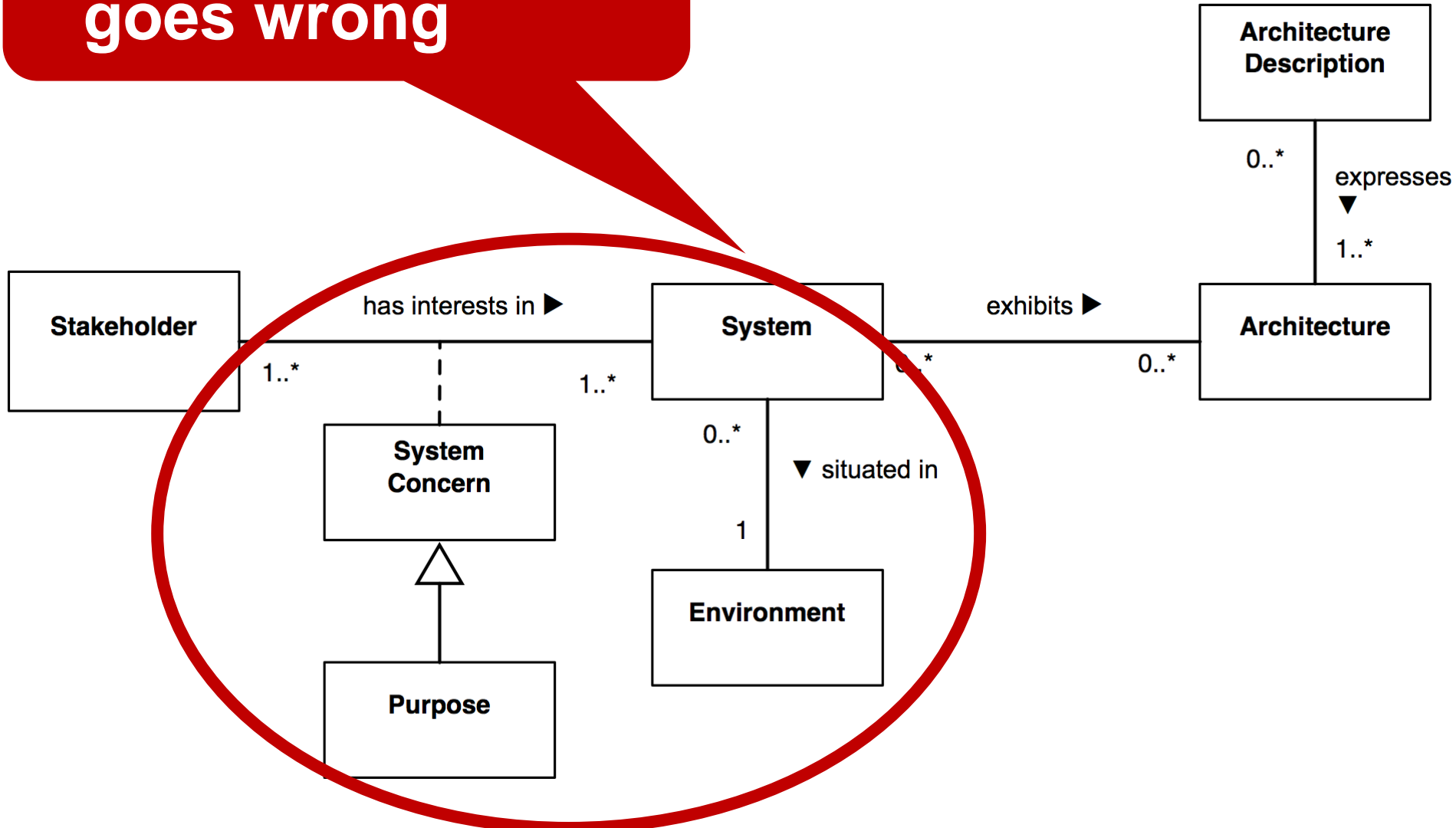




Extrafunctional properties

A Conceptual Model of AD

- What if something goes wrong



Risk definition and expression

- IEC 61508– Combination of the probability of a damage and of its severity
- MIL-STD-882D– An expression of the impact and possibility of a mishap in terms of potential mishap severity and probability of occurrence

		Mishap severity			
		Negligible	Marginal	Critical	Catastrophic
Probability of occurrence	Frequent			High	
	Probable				
	Occasional	Low	Medium		Serious
	Remote				
	Improbable				

Probabilities of occurrence and mishap severity ↔ Application domains (transportation, energy production, telecommunications, banking, etc.)

Basic Concepts of Dependability

Jean-Claude Laprie



DeSIRE and DeFINE Workshop — Pisa, 25-27 November 2002



Dependability : ability to deliver service that can justifiably be trusted

Service delivered by a system: its behavior as it is perceived by its user(s)

User: another system that interacts with the former

Function of a system: what the system is intended to do

(Functional) **Specification**: description of the system function

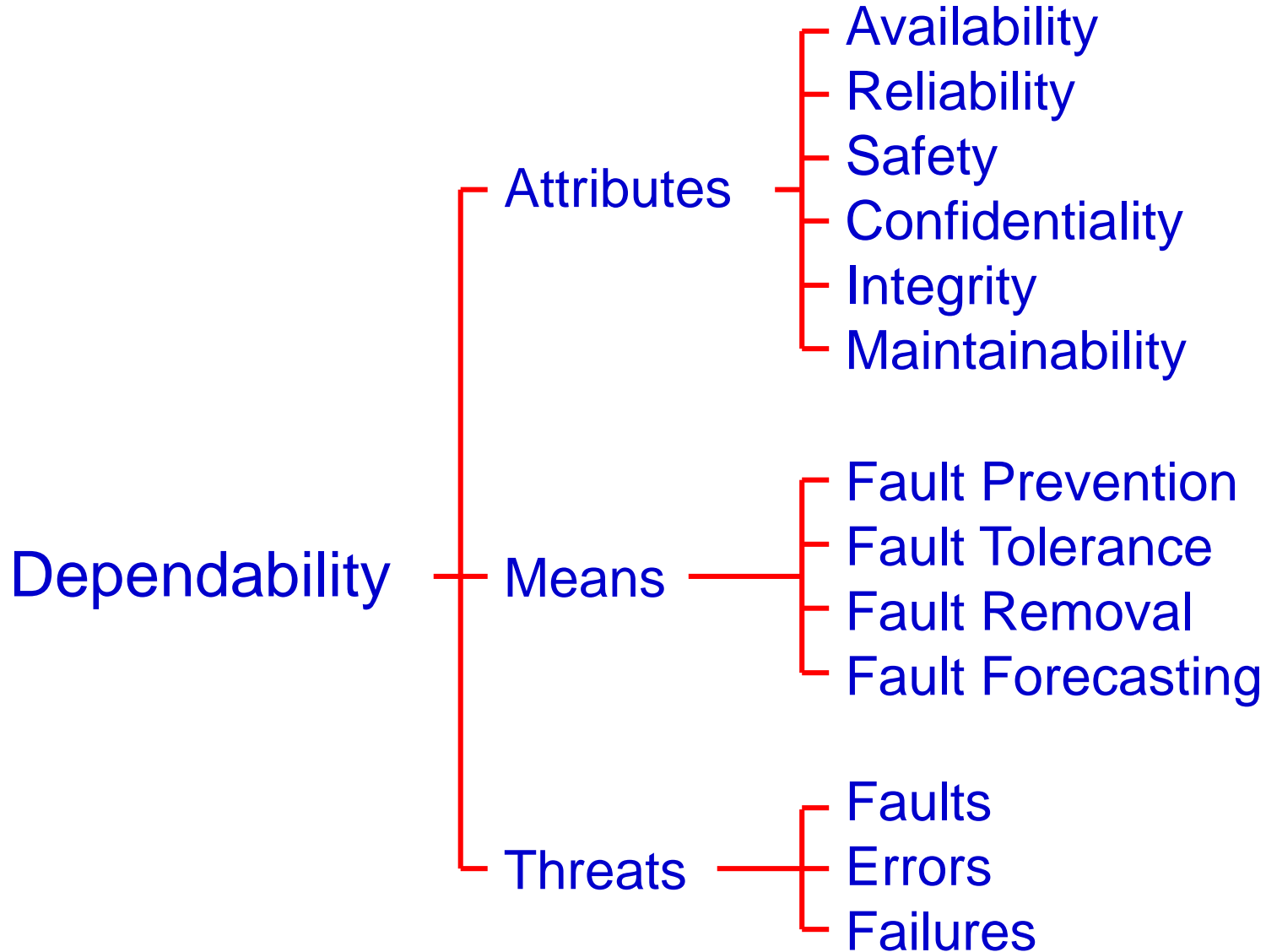
Correct service: when the delivered service implements the system function

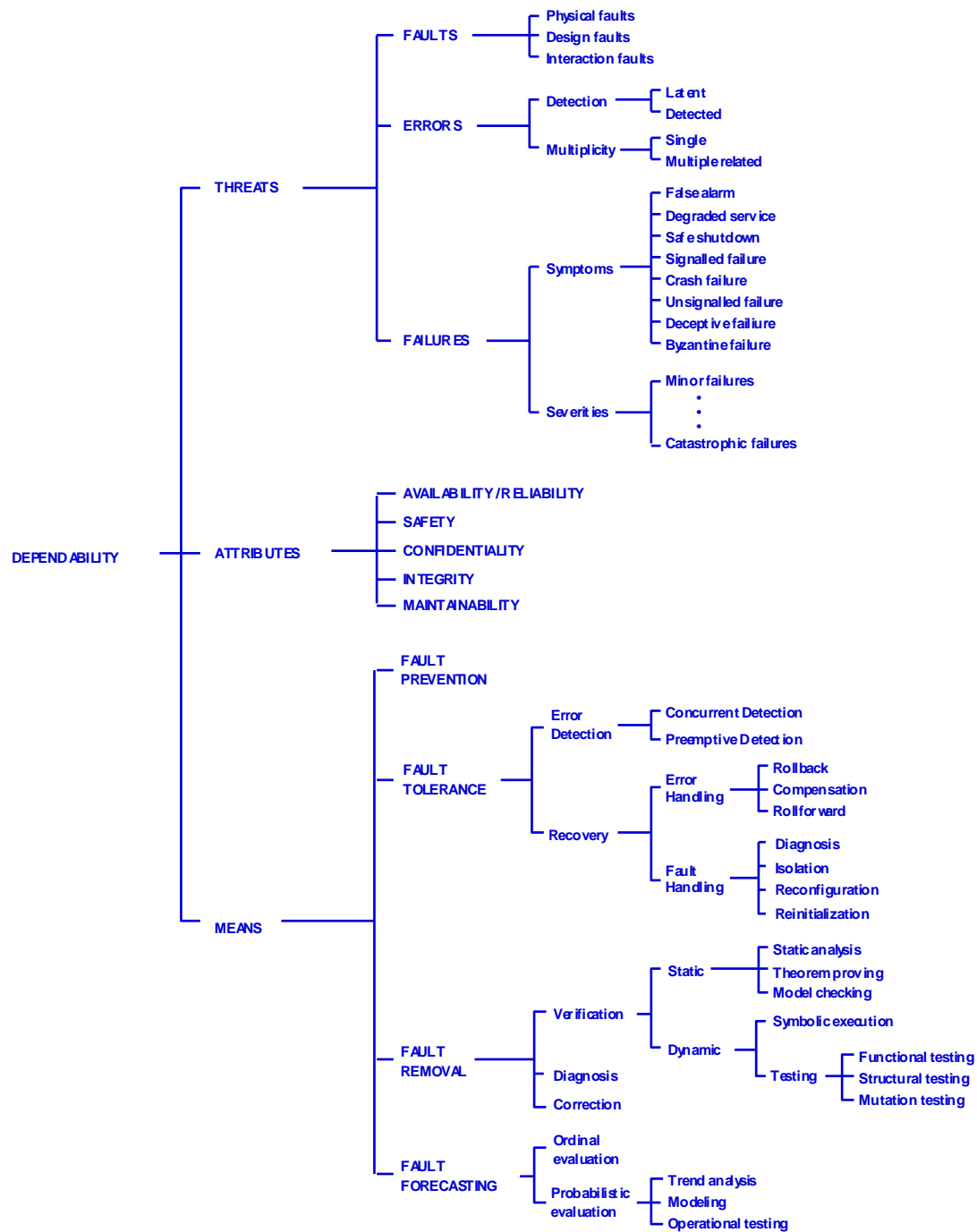
System failure: event that occurs when the delivered service deviates from correct service, either because the system does not comply with the specification, or because the specification did not adequately describe its function

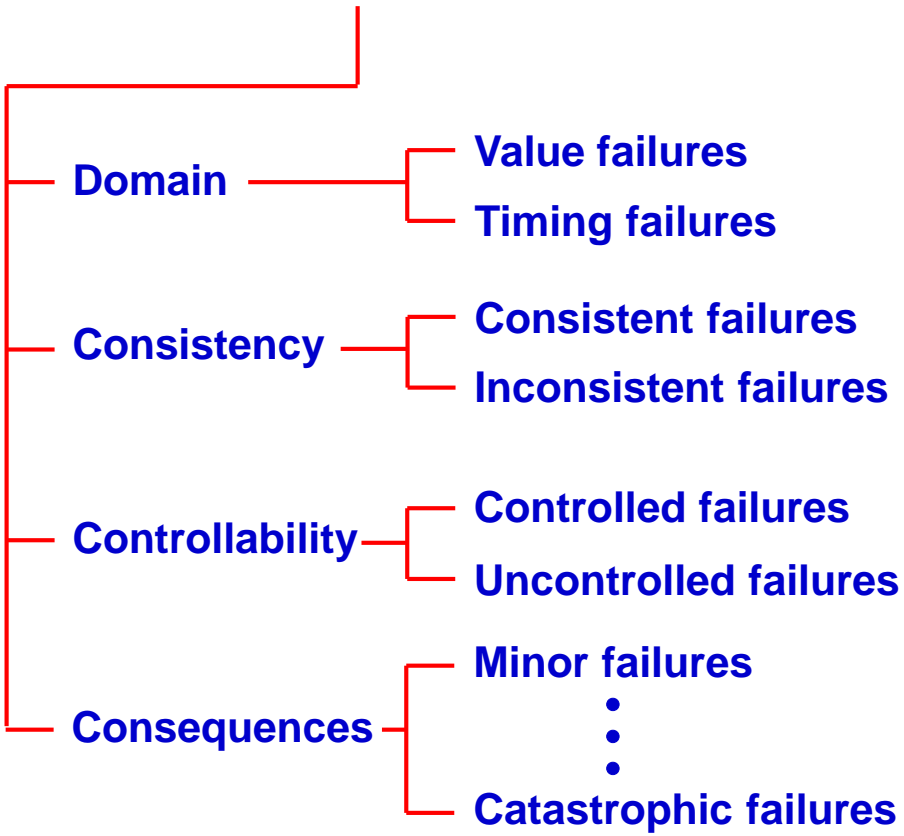
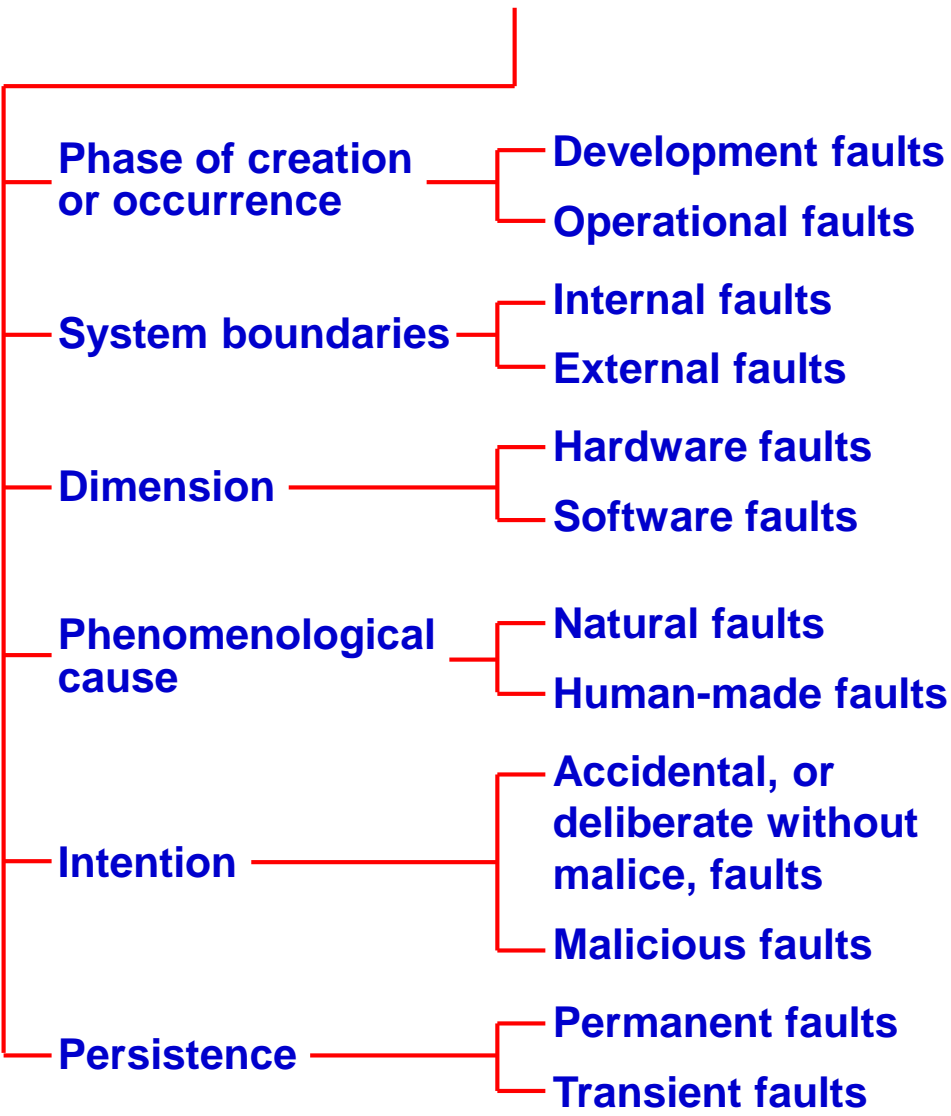
Failure modes: the ways in which a system can fail, ranked according to failure severities

Dependability: ability to avoid failures that are more frequent or more severe than is acceptable to the user(s)

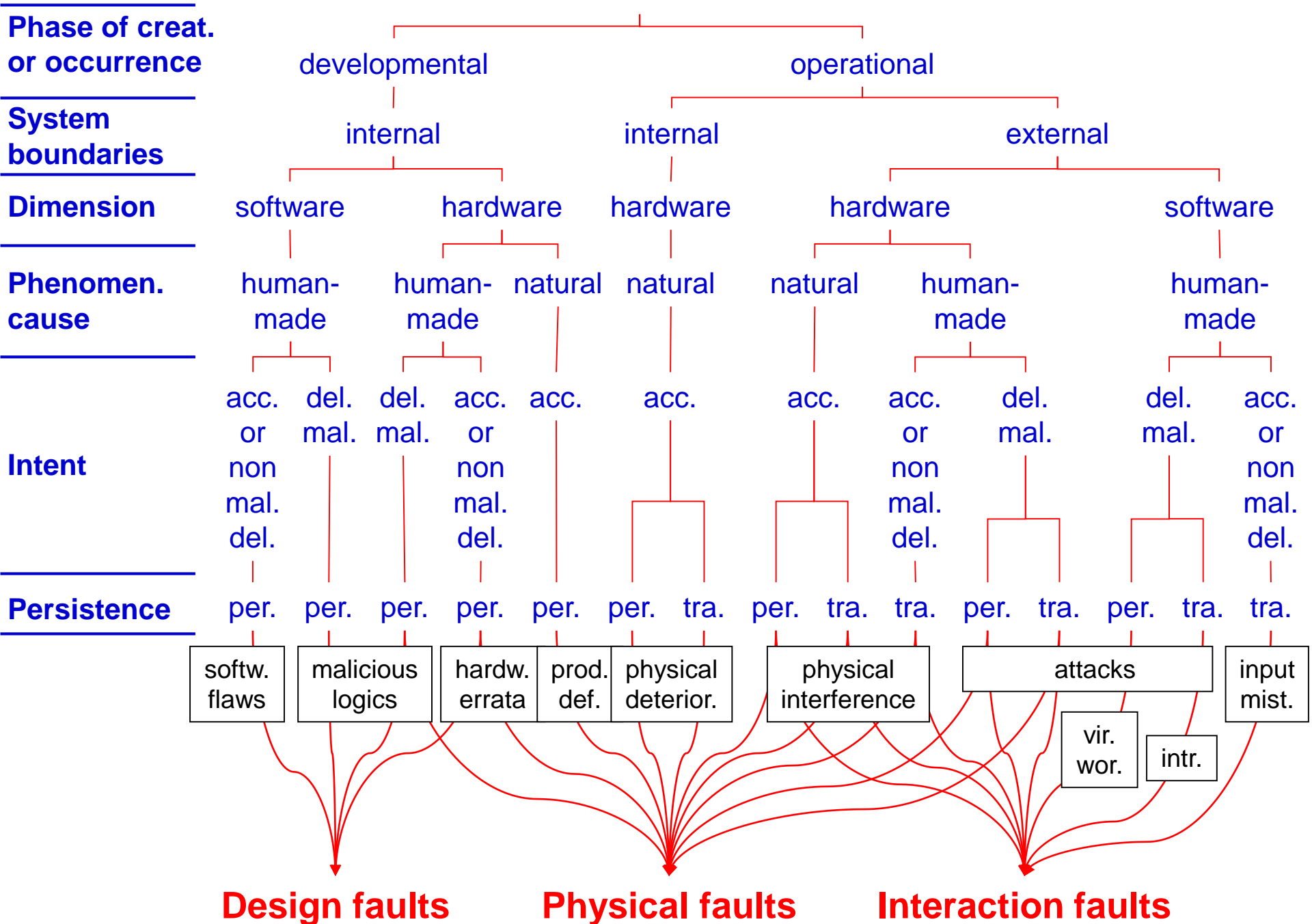
When failures are more frequent or more severe than acceptable:
dependability failure







Faults



Importance of concept formulation

- ❖ Agreed terminology for people exchanges and interactions
- ❖ Shared understanding

Update and evolution

- ❖ Relationship dependability - security
- ❖ Dependability specification
- ❖ Dependability scales and classes (partial ordering, distributions)
- ❖ Socio-technical systems
 - Risk (losses and gains)
 - Human faults, including malicious ones
 - Operation, incl. organizational drifts
 - Development → process failures

Concept	Dependability	Survivability	Trustworthiness
Goal	1) ability of a system to deliver service that can justifiably be trusted 2) ability of a system to avoid failures that are more frequent or more severe than is acceptable to the user(s)	capability of a system to fulfill its mission in a timely manner	assurance that a system will perform as expected
Threats present	1) design faults (e.g., software flaws, hardware errata, malicious logics) 2) physical faults (e.g., production defects, physical deterioration) 3) interaction faults (e.g., physical interference, input mistakes, attacks, including viruses, worms, intrusions)	1) attacks (e.g., intrusions, probes, denials of service) 2) failures (internally generated events due to, e.g., software design errors, hardware degradation, human errors, corrupted data) 3) accidents (externally generated events such as natural disasters)	1) hostile attacks (from hackers or insiders) 2) environmental disruptions (accidental disruptions, either human-made or natural) 3) human and operator errors (e.g., software flaws, mistakes by human operators)
Reference	«Fundamental concepts of dependability» ¹	«Survivable network systems» ²	«Trust in cyberspace» ³

1 A. Avizienis, J.C. Laprie, B. Randell, "Fundamental concepts of dependability", March 2001.

2 R.J. Ellison, D.A. Fischer, R.C. Linger, H.F. Lipson, T. Longstaff, N.R. Mead, "Survivable network systems: an emerging discipline", Technical Report CMU/SEI-97-TR-013, November 1997, revised May 1999.

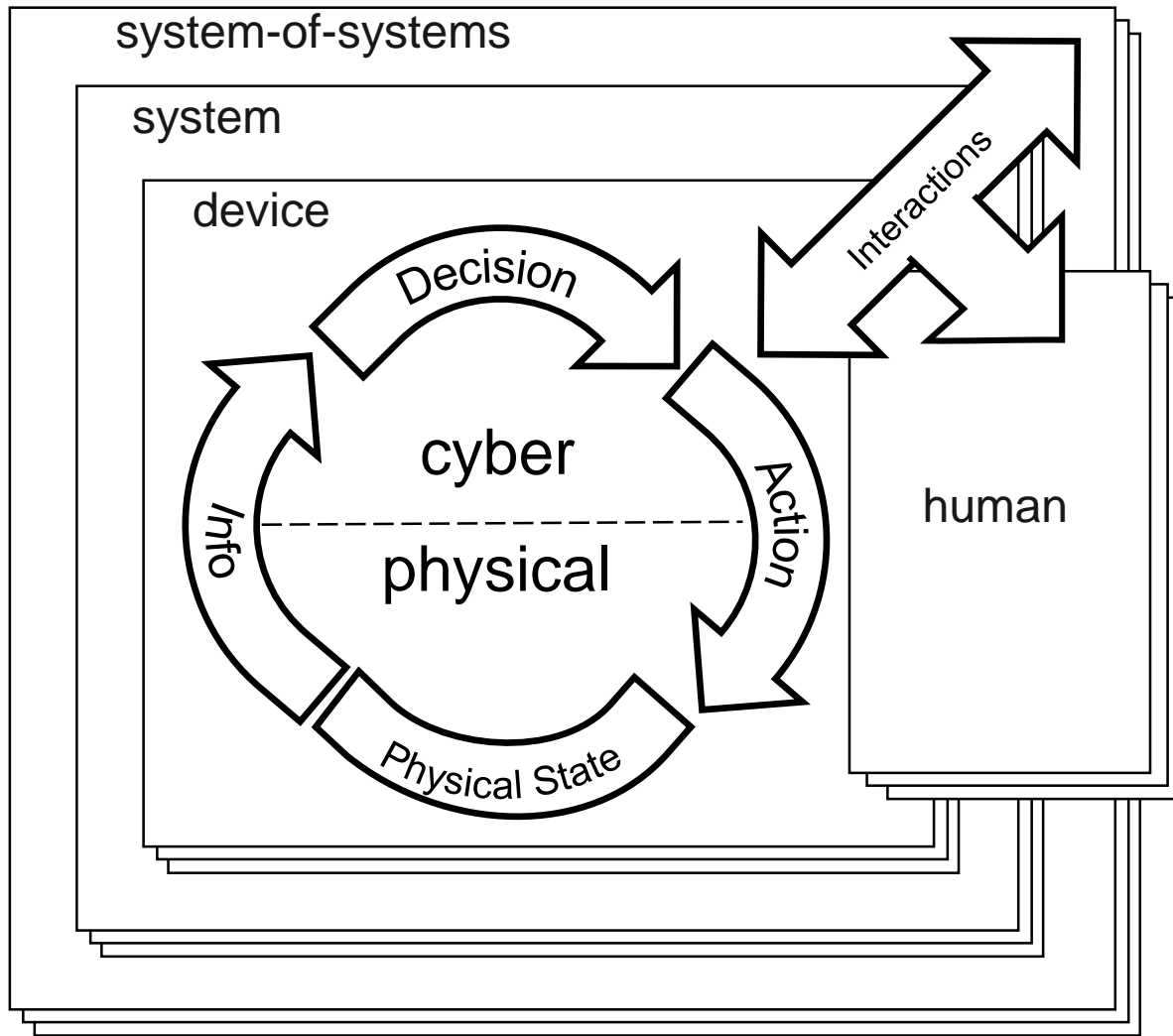
3 F. Schneider, ed., *Trust in Cyberspace*, National Academy Press, 1999.

<https://pages.nist.gov/cpspwg/>

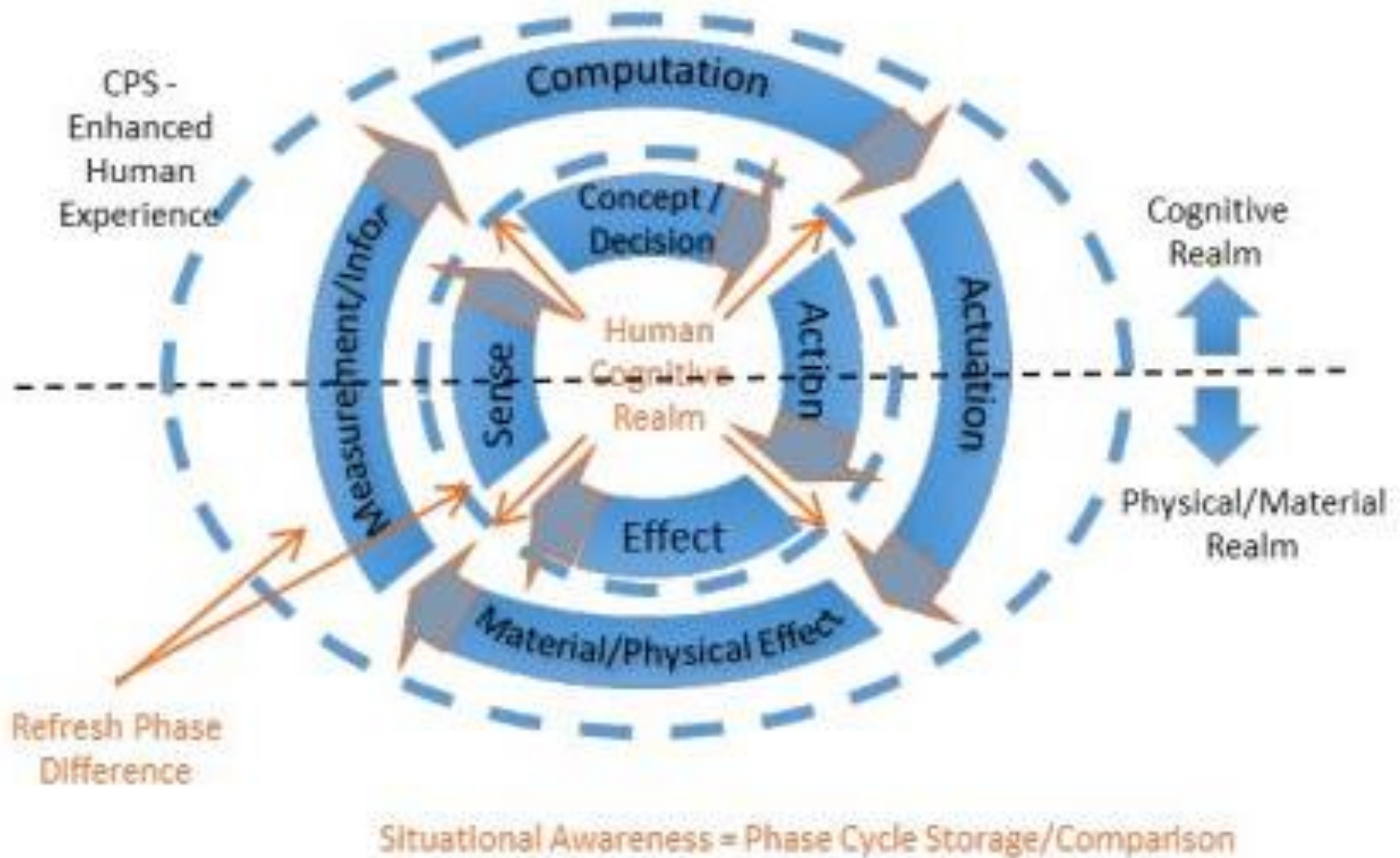
https://s3.amazonaws.com/nist-sgcps/smartcityframework/ies-city_framework/TechnicalArtifacts/ApplicationFramework_BreadthAssessmentTool.xlsm

Use cases for a CPS framework

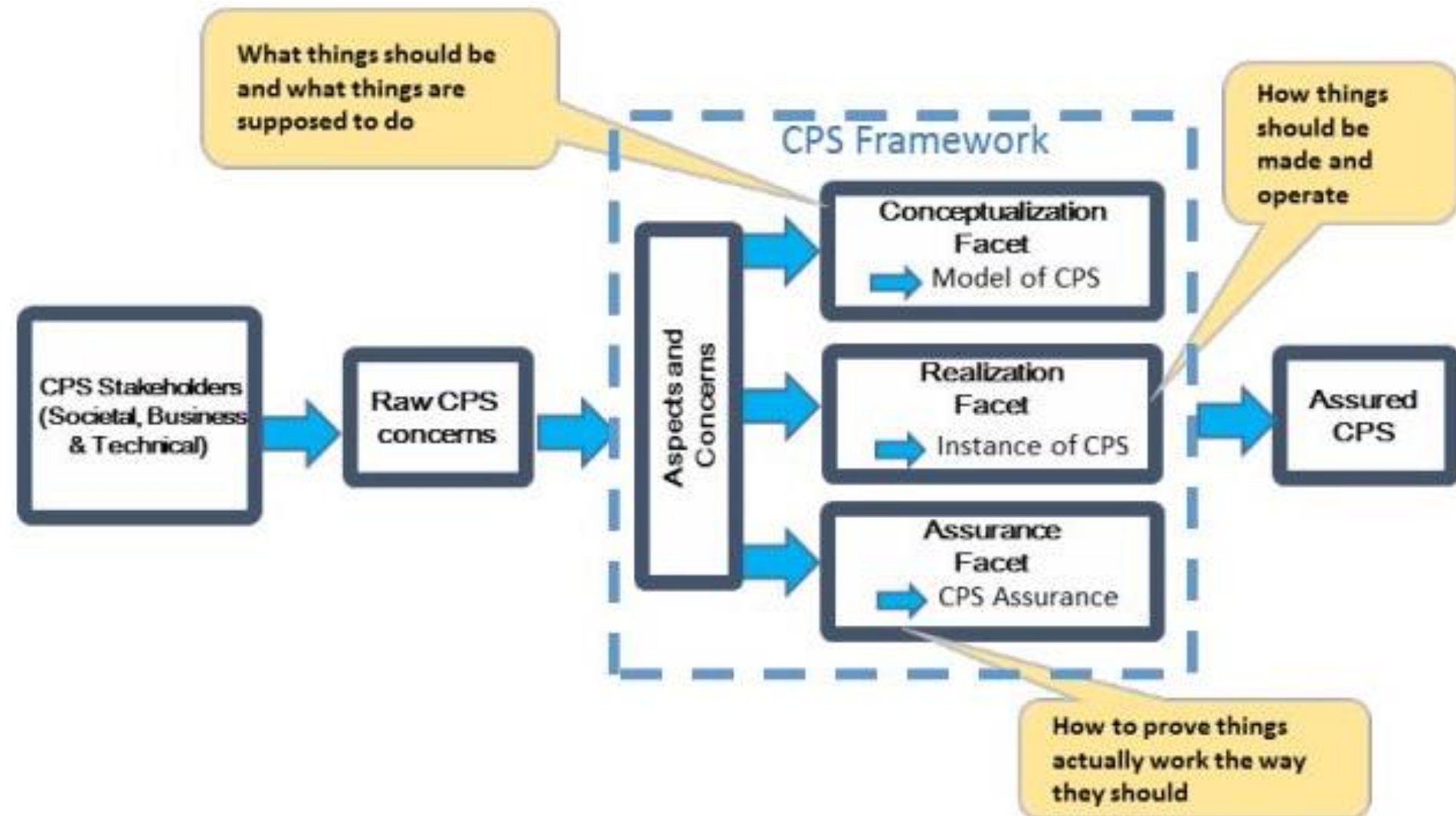
NIST CPS Framework



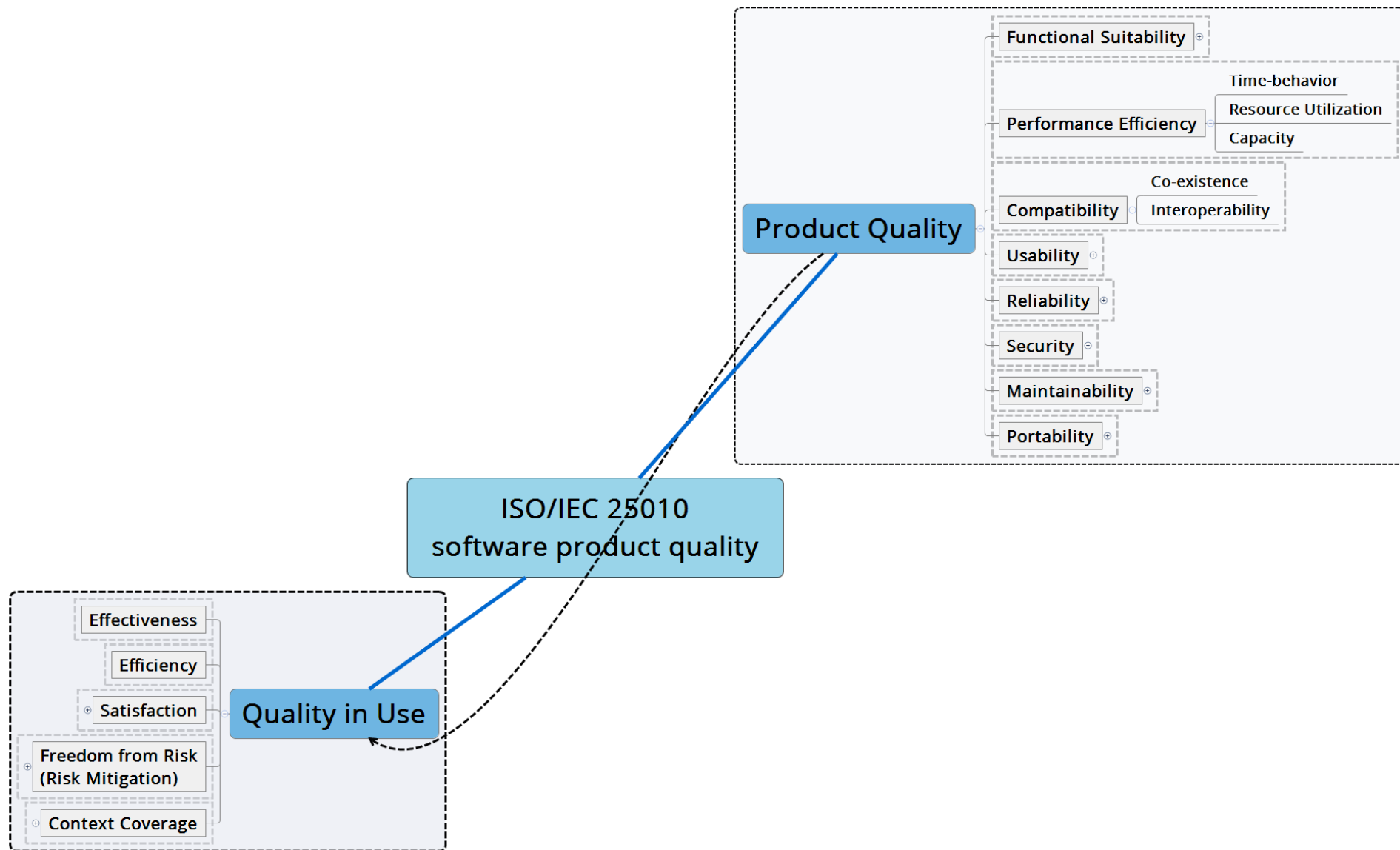
NIST CPS Cognitive Cycle



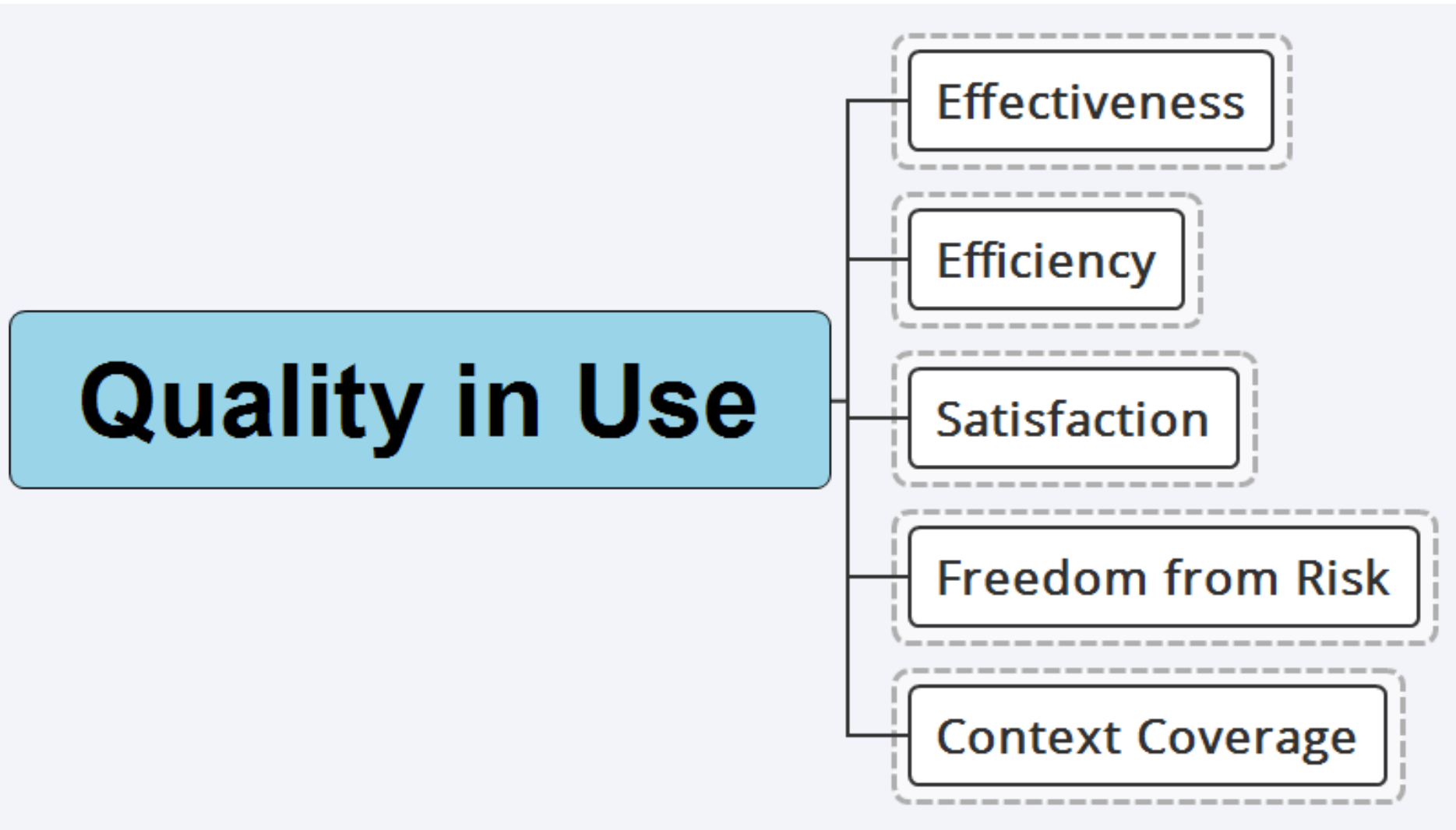
NIST CPS Framework for System Design



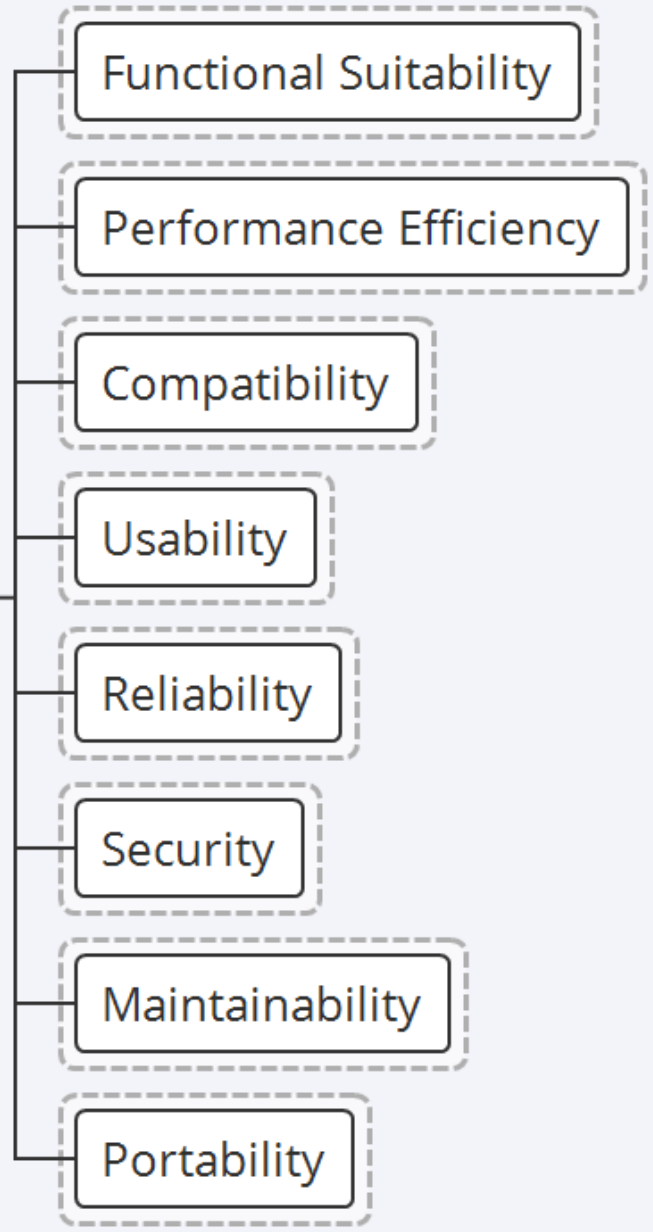
ISO/IEC 25010 hhsoftware product quality



Quality in Use overview



Product Quality

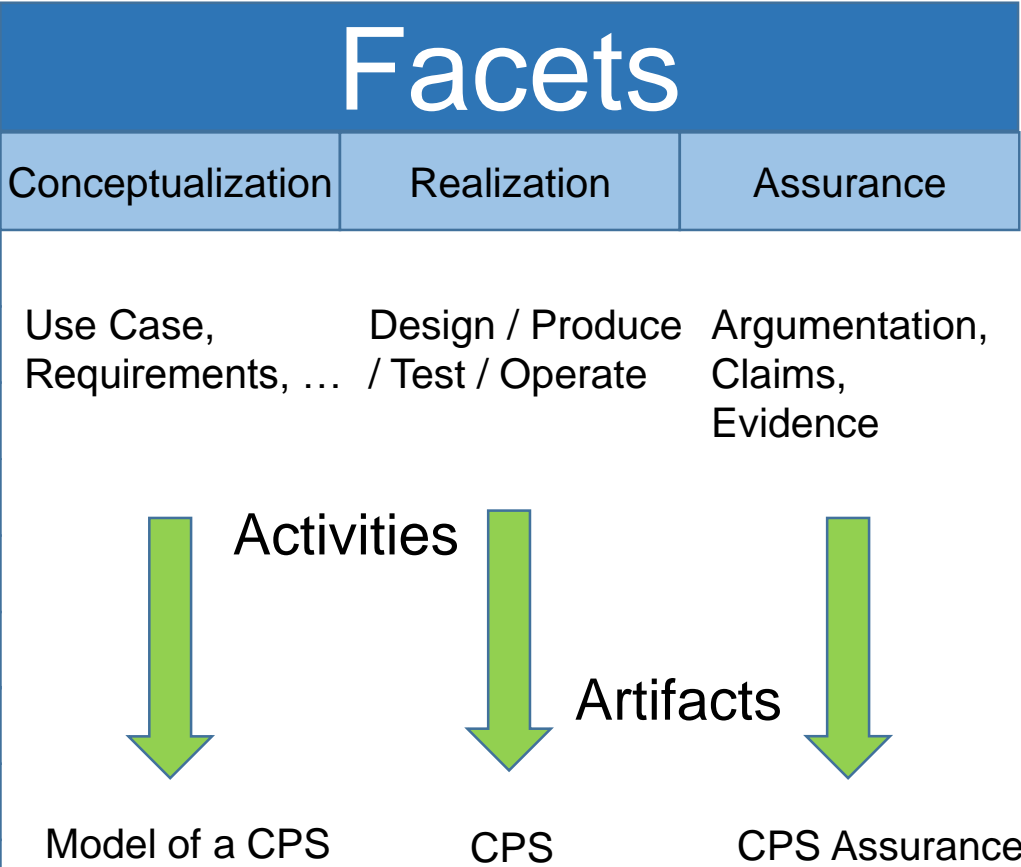
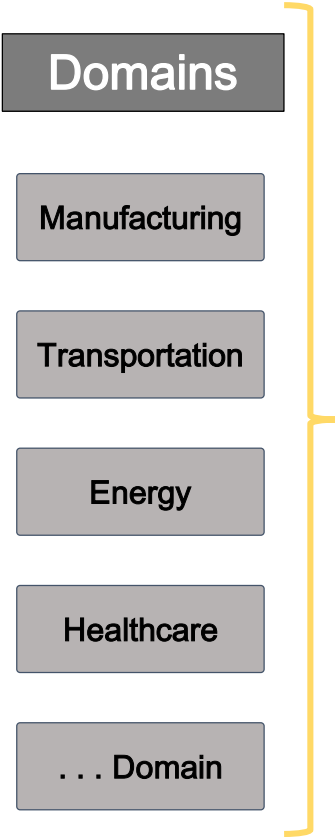




Quality in Use



CPS Framework



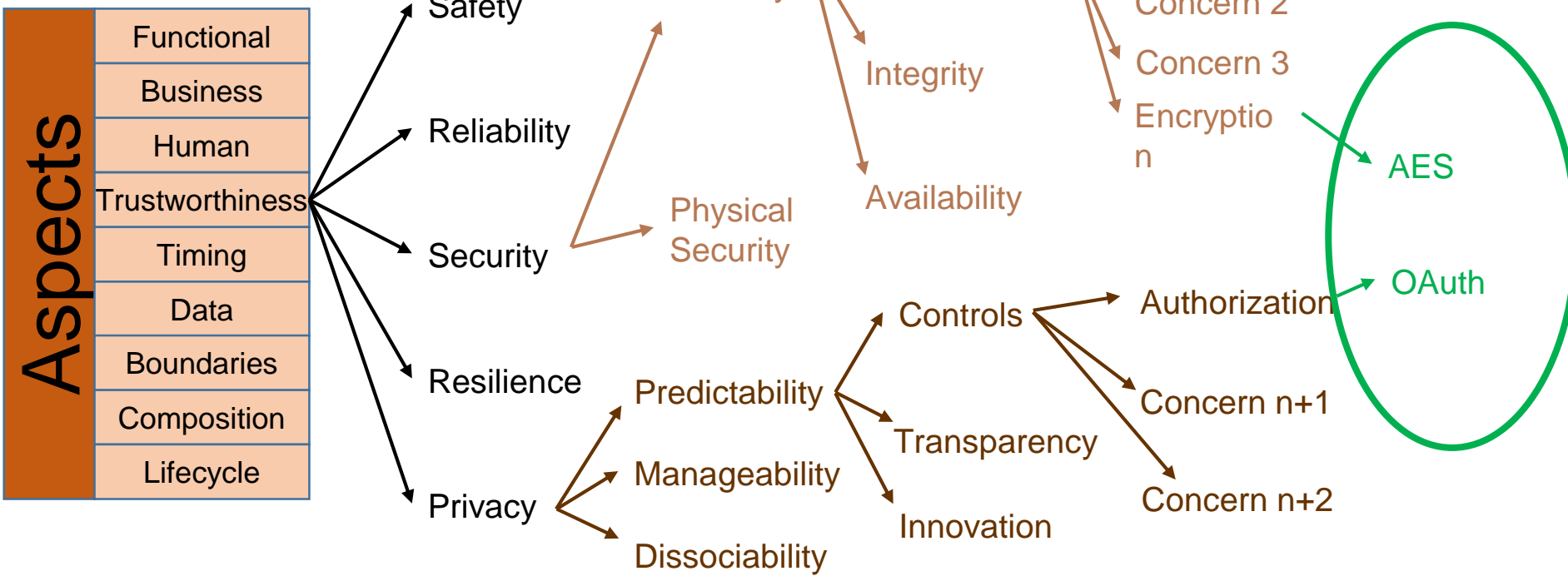
Aspects

Aspect	Description
Functional	Concerns about function including sensing, actuation, control, communications, physicality, etc.
Business	Concerns about enterprise, time to market, environment, regulation, cost, etc.
Human	Concerns about human interaction with and as part of a CPS.
Trustworthiness	Concerns about trustworthiness of CPS including security, privacy, safety, reliability, and resilience.
Timing	Concerns about time and frequency in CPS, including the generation and transport of time and frequency signals, timestamping, managing latency, timing composability, etc.
Data	Concerns about data interoperability including fusion, metadata, type, identity, etc.
Boundaries	Concerns related to demarcations of topological, functional, organizational, or other forms of interactions.
Composition	Concerns related to the ability to compute selected properties of a component assembly from the properties of its components. Compositionality requires components that are composable: they do not change their properties in an assembly. Timing composability is particularly difficult.
Lifecycle	Concerns about the lifecycle of CPS including its components.

Concerns

Aspect	Concern	Description
		the ability to modify a CPS or its function, if necessary.
Functional	functionality	Concerns related to the function that a CPS provides.
Functional	manageability	Concerns related to the management of CPS function. For example, Managing Timing in complex CPS or SoS is a new issue with CPS that did not exist before. It is being developed with new standards
Functional	measurability	Concerns related to the ability to measure the characteristics of the CPS.
Functional	monitorability	Concerns related to the ease and reliability with which authorized entities can gain and maintain awareness of the state of a CPS and its operations. Includes logging and audit functionality.
Functional	performance	Concerns related to the ability of a CPS to meet required operational targets.
Functional	physical	Concerns about purely physical properties of CPS including seals, locks, safety, and EMI.
Functional	physical context	Concerns relating to the need to understand a specific observation or a desired action relative to its physical position (and uncertainty.) While this information is often implied and not explicit in traditional physical systems, the distributed, mobile nature of CPS makes this a critical concern.
Functional	sensing	Concerns related to the ability of a CPS to develop the situational awareness required to perform its function.

CPS Property Tree



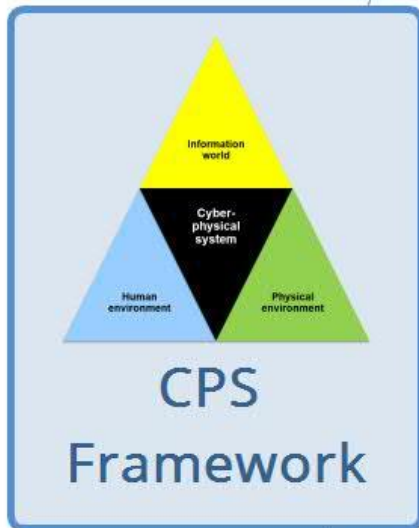
A secure, privacy protected message exchange might consist of the simultaneous (set of) properties:
 {Trustworthiness.Security.Cybersecurity.Confidentiality.Encryption.AES, Trustworthiness.Privacy.Predictability.Controls.Authorization.OAuth}

Specialization of frameworks: domain specific modeling

DOMAIN

CATEGORIES

SUBCATEGORIES



Built environment 

Smart Home

Smart Building

Land use and management

Water and wastewater

Waste

Energy

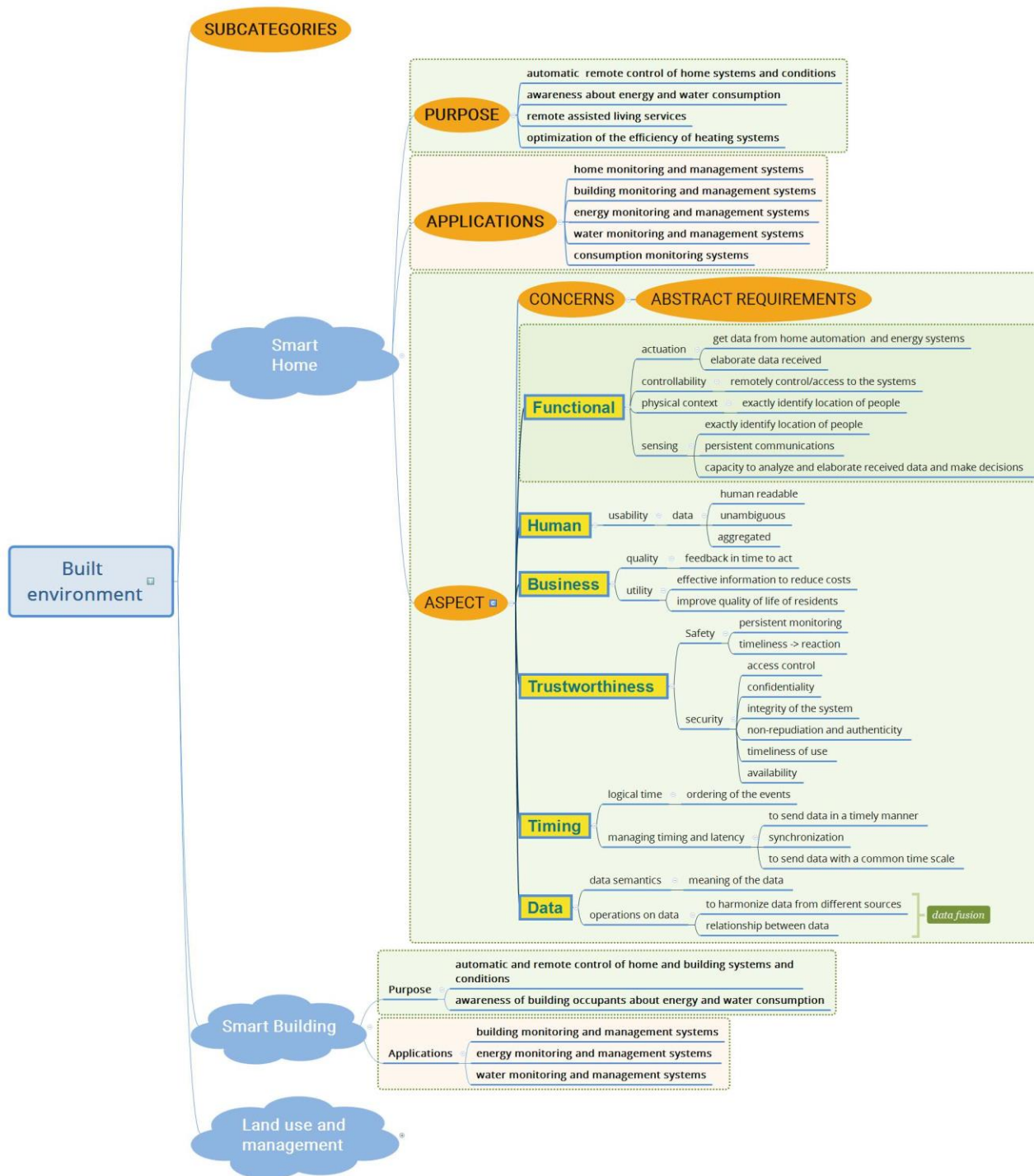
Transportation

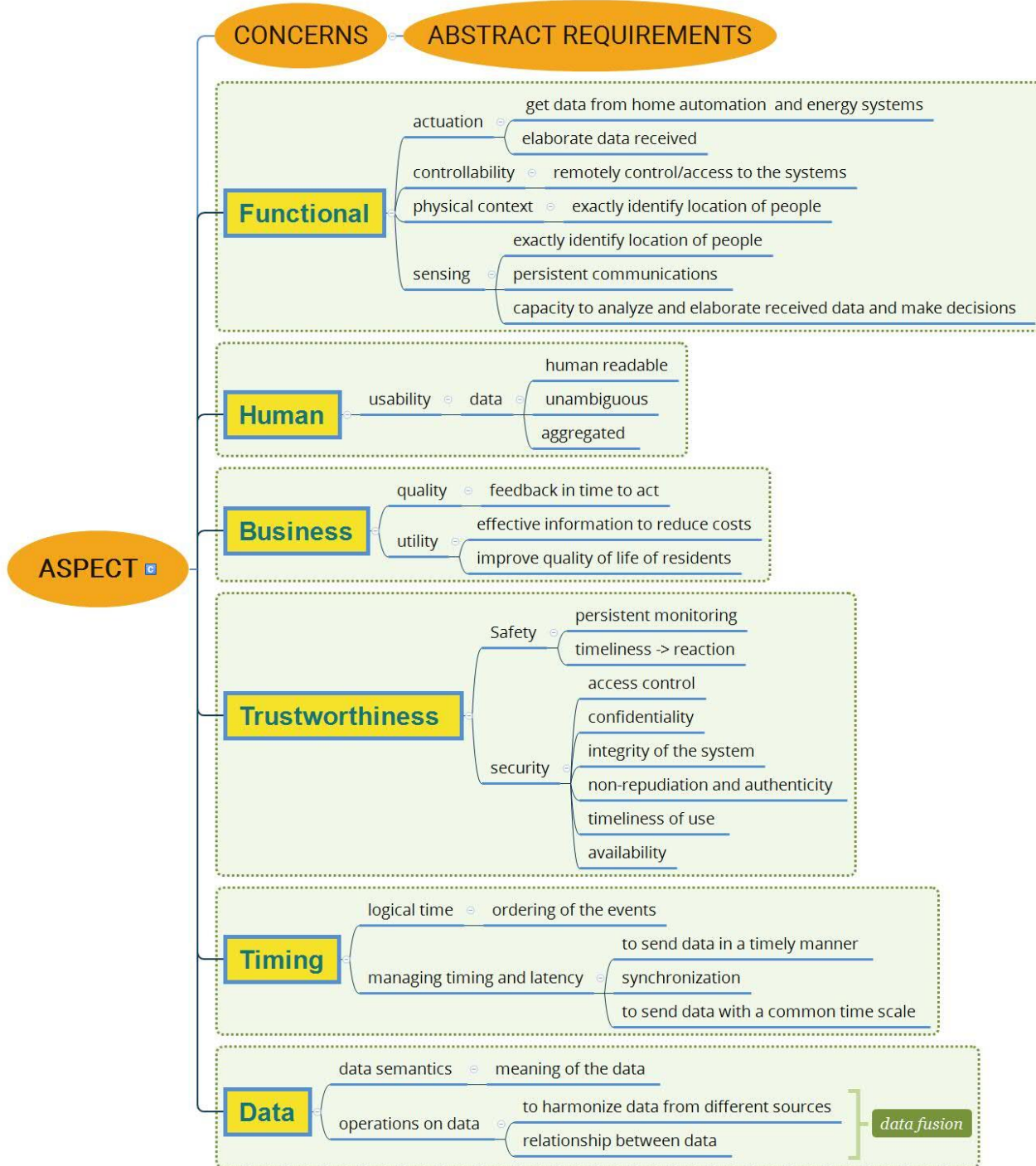
Education 

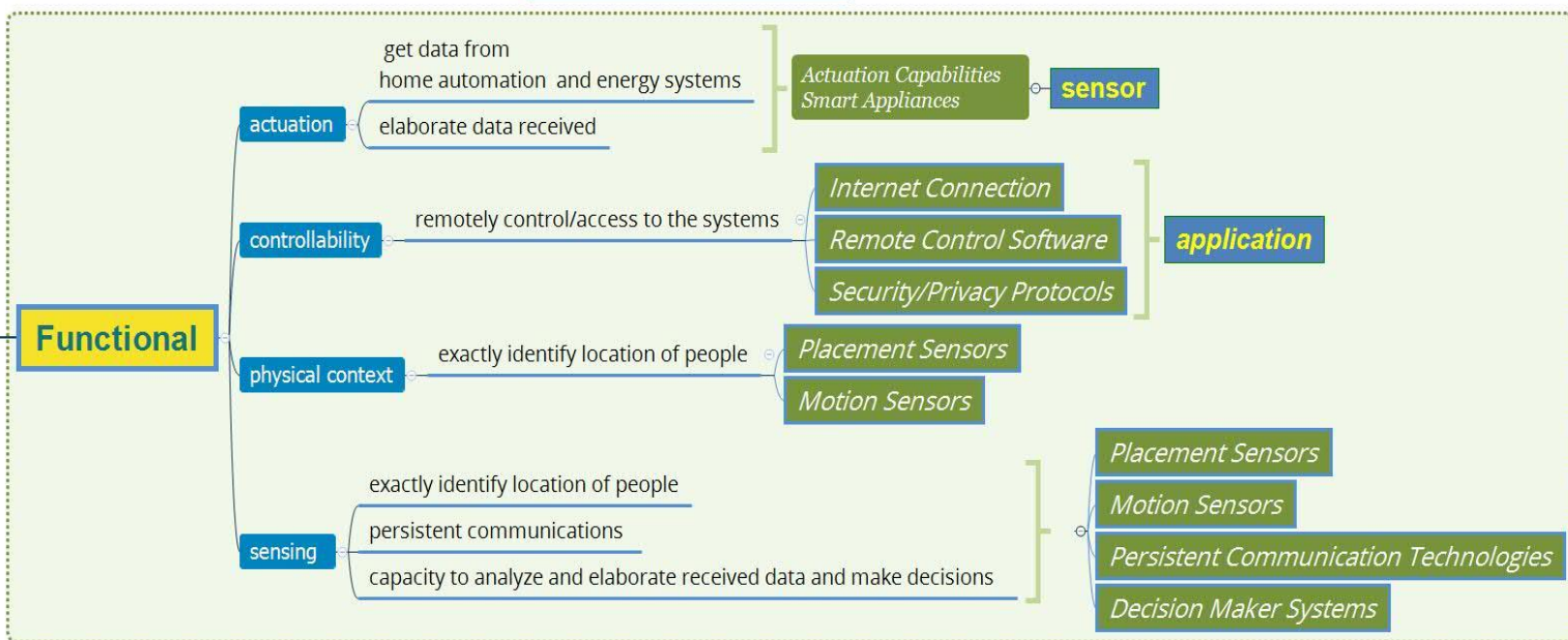
Health

Socio-economic development

Public safety, policing and emergency response







ASPECT

Human

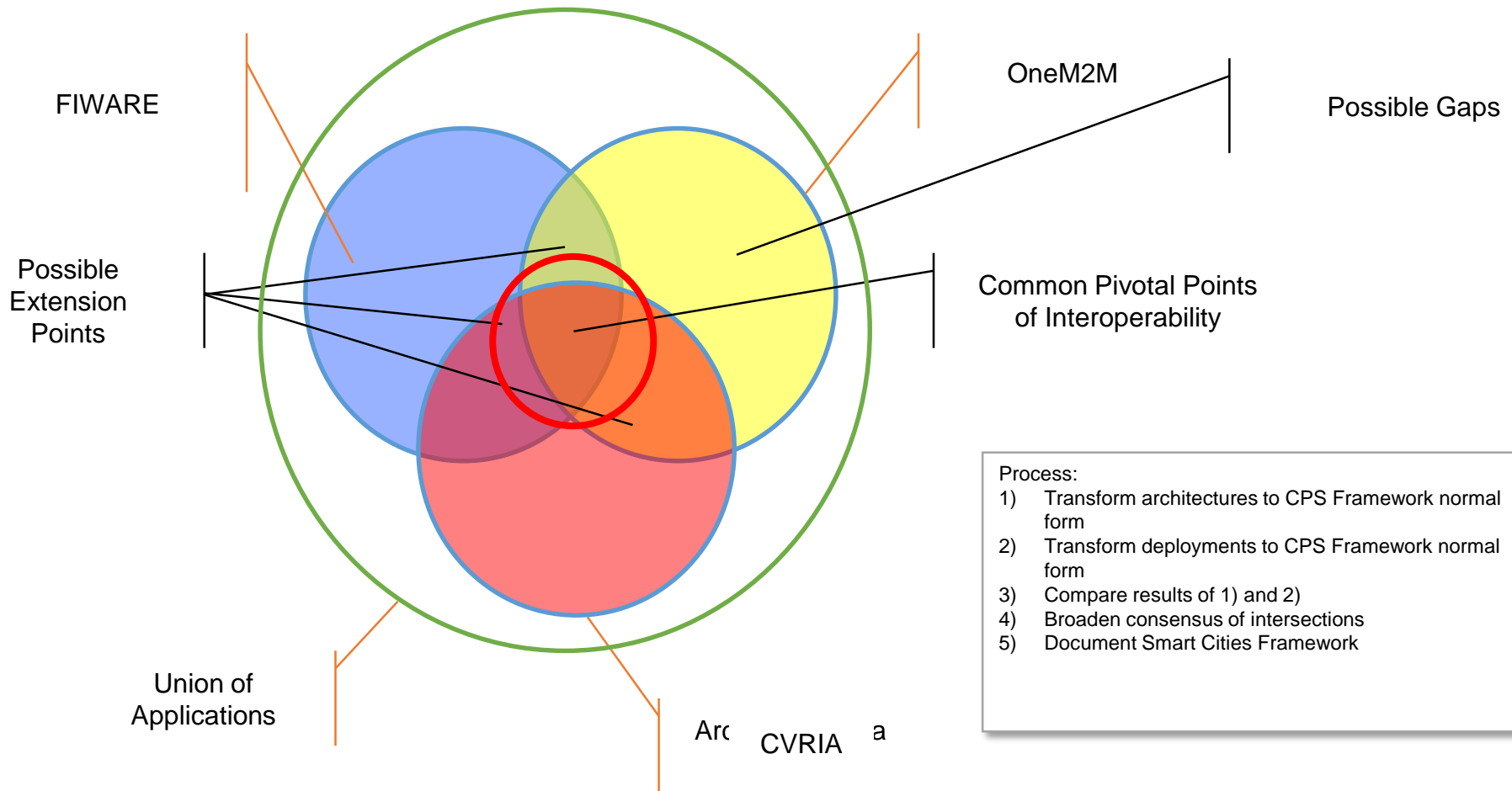
Business

Trustworthiness

Timing

Data

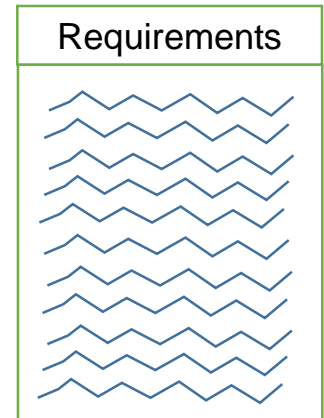
How to Discover Consensus



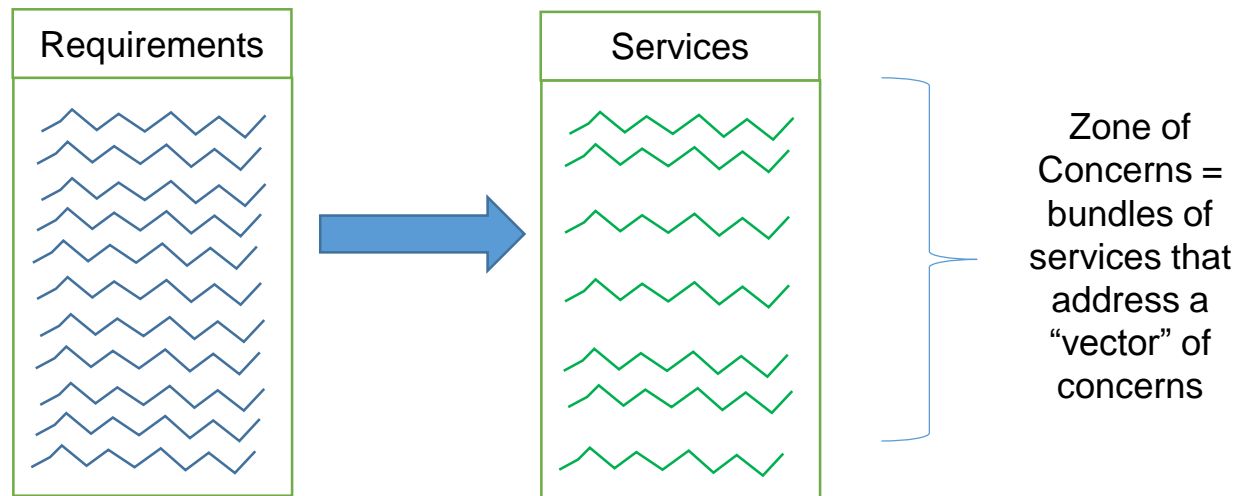
Specs to Pivotal Points of Interoperability



Technology level (Device, System, System of Systems) Technology scope description (text) Zone of Concerns (text -- Enterprise, Field, Mobile, Premises)			
Aspect/Concern	Is a Solution Provided?		
	CVRIA	FIWARE	
Functional	yes		
physical actuation	yes	yes	no
communication	yes	yes	yes
Syntactic Interoperability	see nest 3		yes
OSI-Application	yes		yes
OSI-Presentation	yes		
Network Interoperability	see nest 3		
OSI-Session	yes	yes	
OSI-Transport	yes		no
OSI-Network	yes		no
Basic Connectivity	see nest 3	no	no
OSI-Data Link	yes		no
OSI-Physical	yes		no

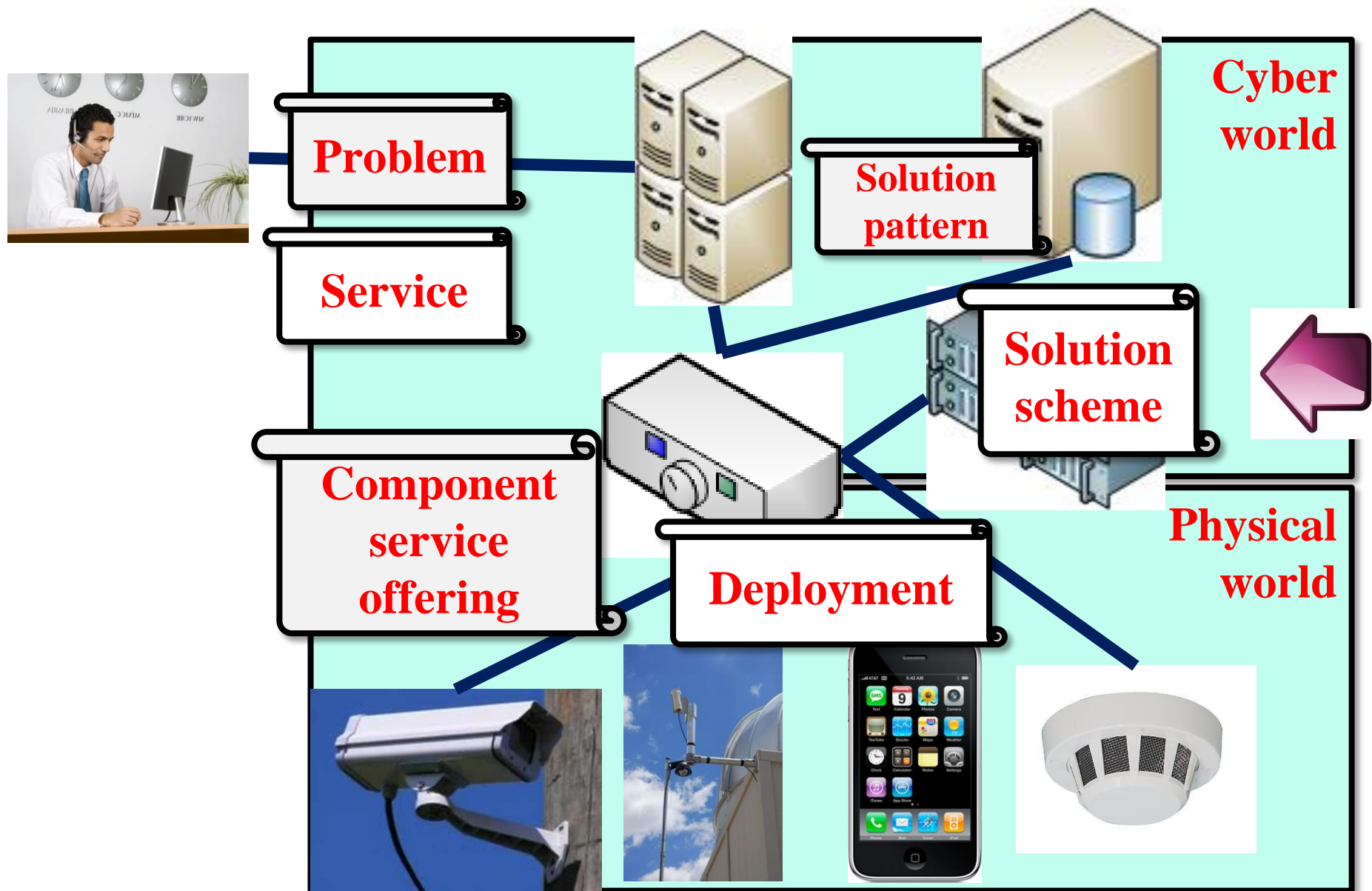


Zones of Concern: From Requirements To Services

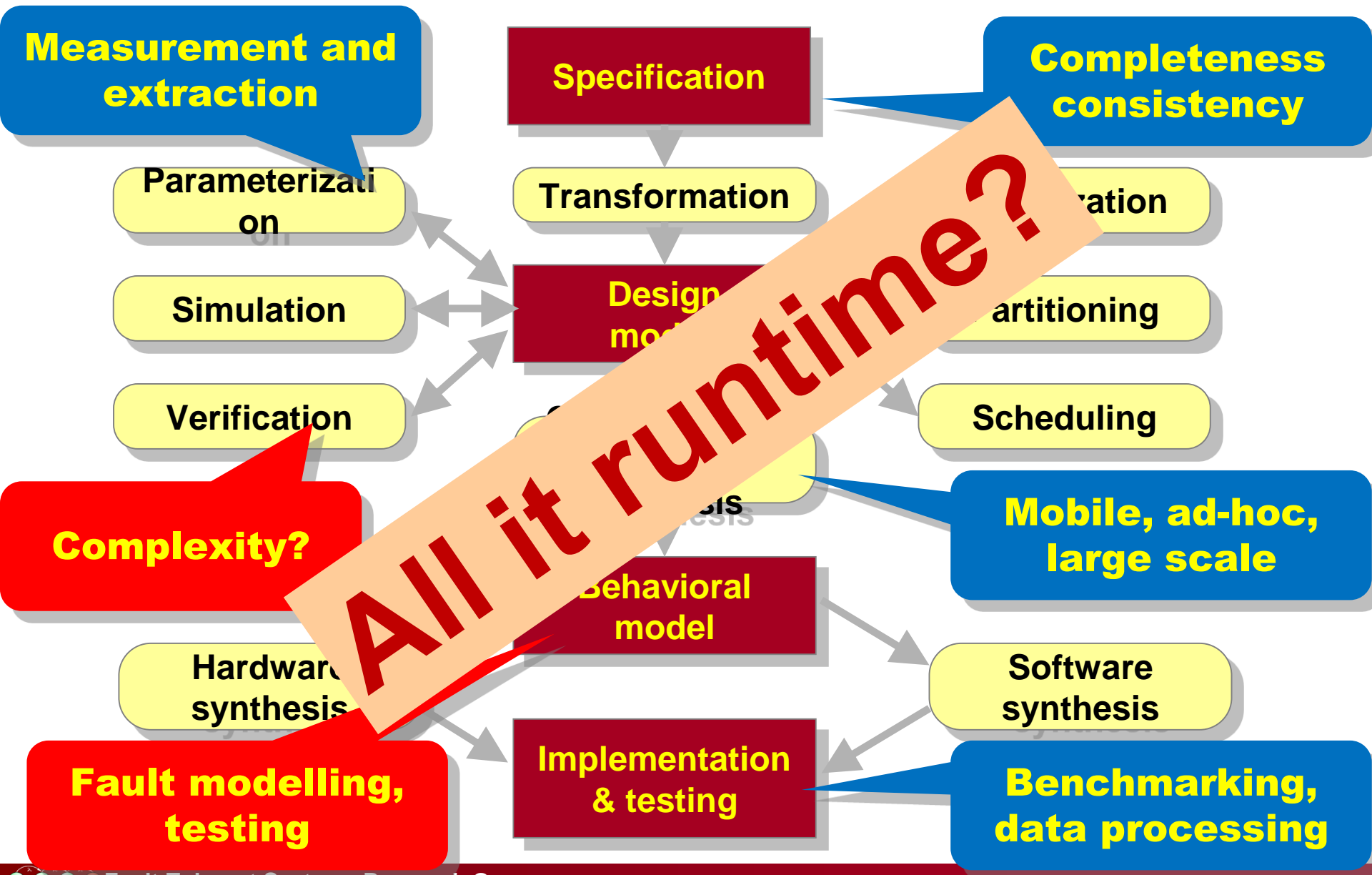


E.g. CyberSecurity Zone of Concerns: Authorization service + Confidentiality Service

COMPOSITION OF CYBER-PHYSICAL SYSTEMS



Critical CPS design and challenges



Development Process for Critical Systems

Unique Development Process (Traditional V-Model)



Critical Systems Design

- requires a **certification process**
- to develop **justified evidence**
- that the **system is free of flaws**

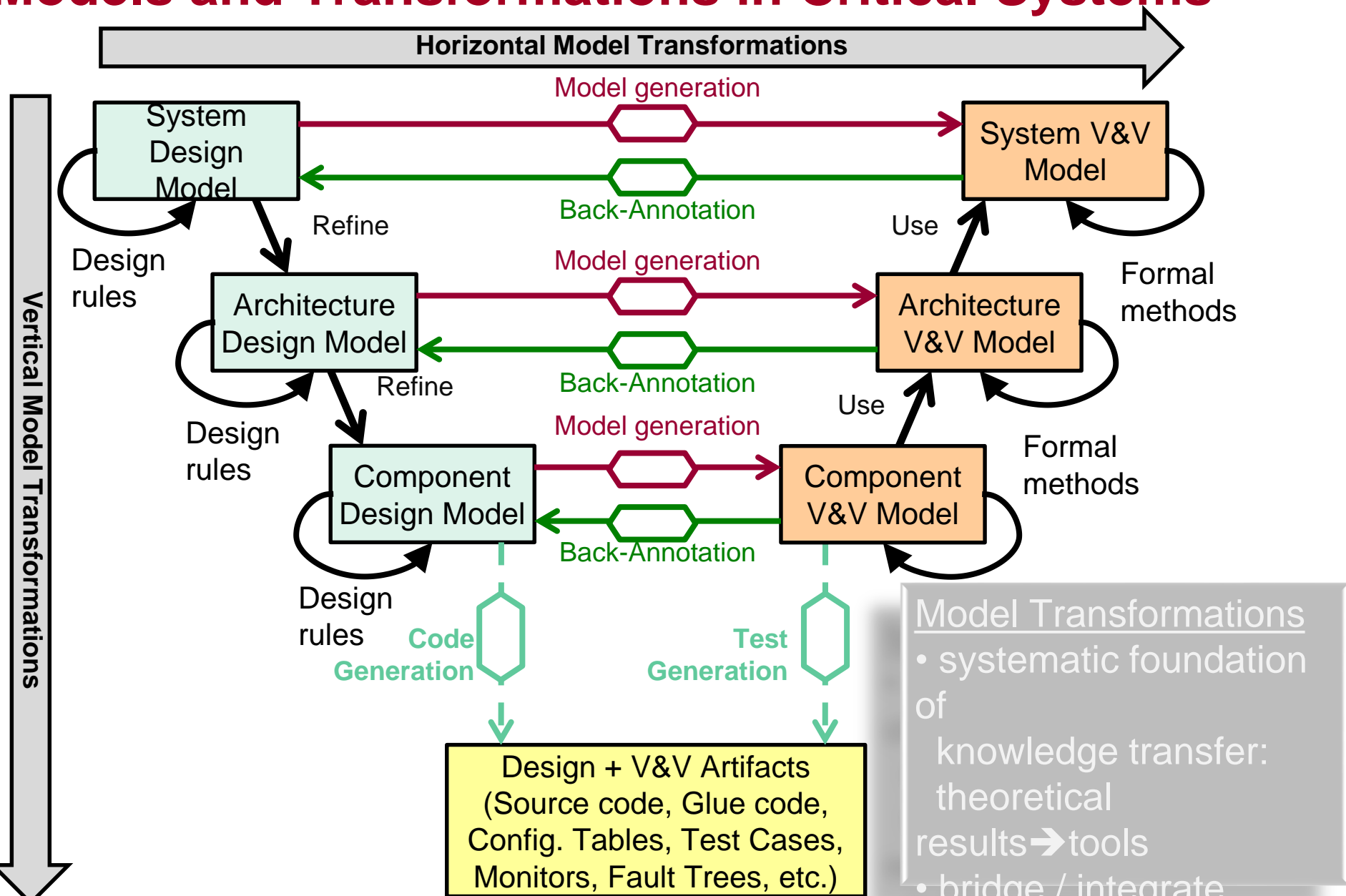
Software Tool Qualification

- obtain **certification credit**
- for a **software tool**
- used **in critical system design**

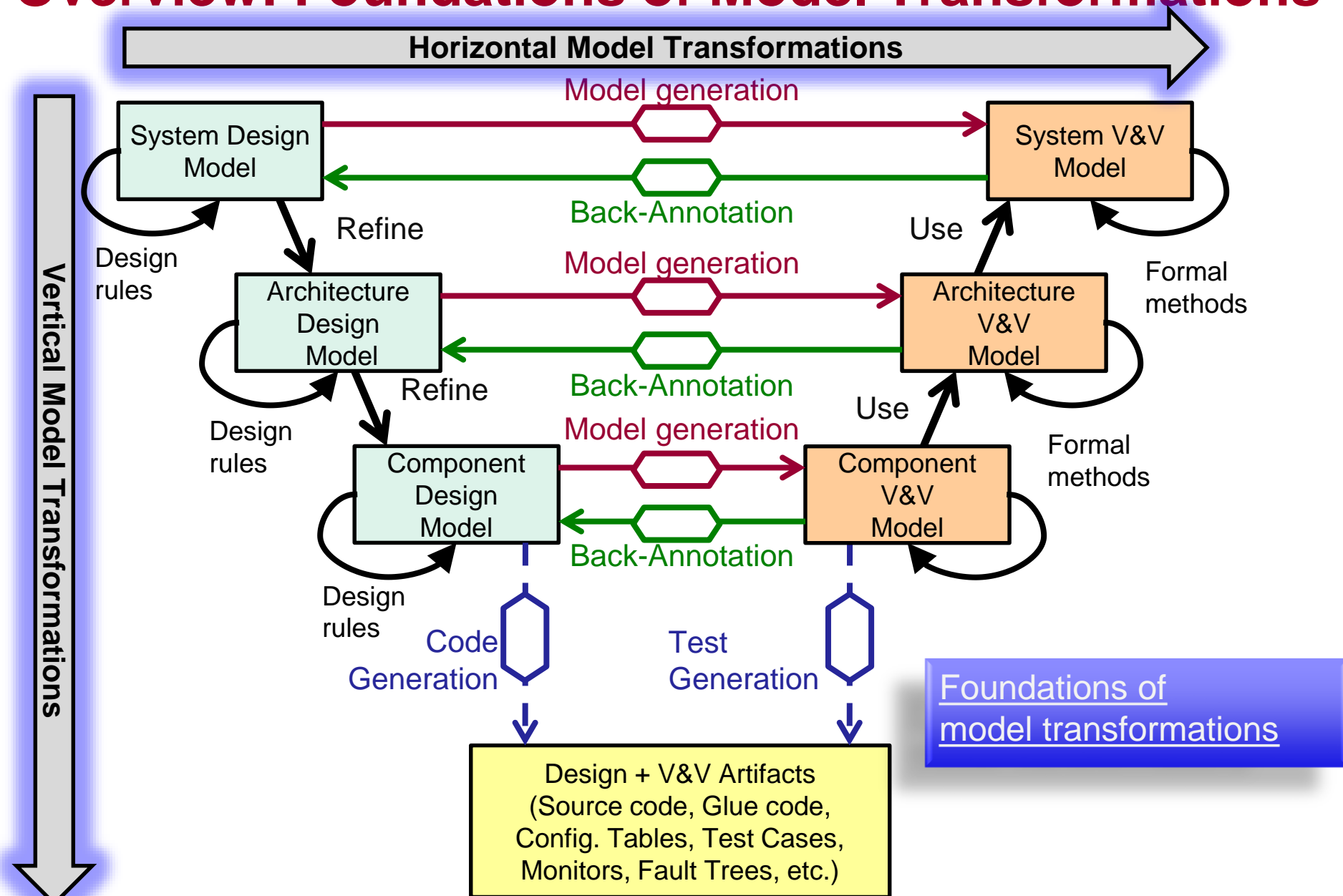
Innovative Tool → Better System

Qualified Tool → Certified Output

Models and Transformations in Critical Systems



Overview: Foundations of Model Transformations

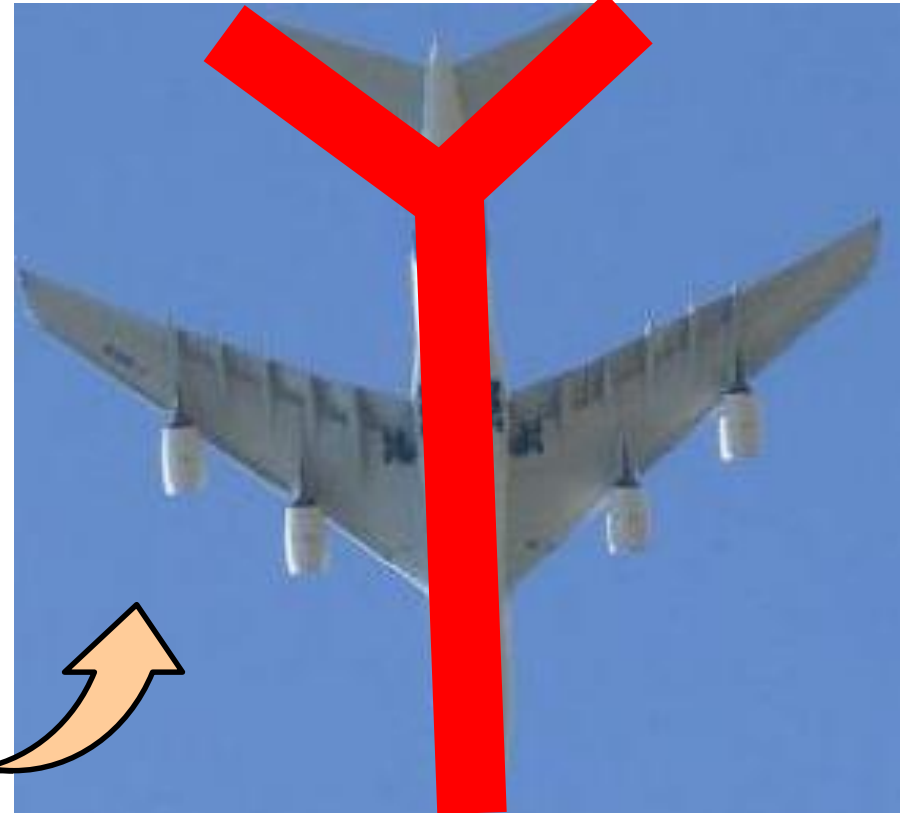


Model-Driven Engineering of Critical Systems

Traditional V-Model



Model-Driven Engineering



- DO-178B/C: Software Considerations in Airborne Systems and Equipment Certification (RTCA, EUROCAE)
- Steven P. Miller: Certification Issues in Model Based Development (Rockwell Collins)

Main ideas of MDE

- early validation of system models
- automatic source code generation
- ➔ quality++ tools ++ development cost--

Design schemes

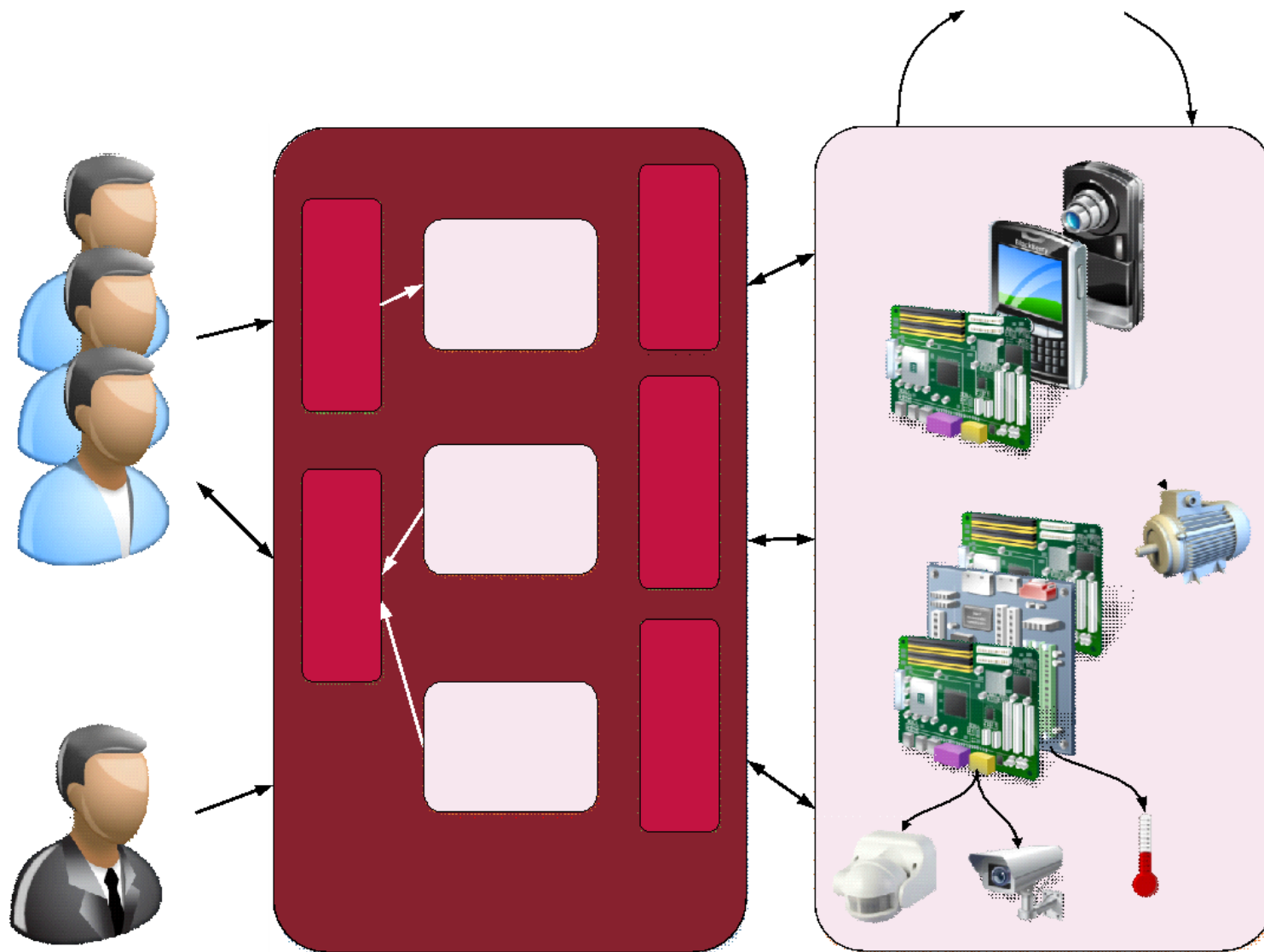
- Detecting changes in system state
- Detecting changes in environment
- Reconfiguration
 - Resource allocation
 - generation of new application/middleware
 - replacement of sensors
 - new information fusion etc.
 - Design space exploration
 - Qualitative
 - Quantitative

Composability

- System design principle:
 - recombinant components
 - can be assembled in various combinations
- Meaningful fusion of self-contained services
- Provide interoperability of devices
 - Bridging the gap between different
 - physical,
 - computational and
 - communication capabilities



Dynamic composition of cyber-physical systems



Requirements of composability

- User interface for describing domain specific constraints
- Abstract interfaces between cooperating nodes
 - Embedded systems connected to sensors and actuators
 - Mobile devices
 - Conventional computing devices,
 - cloud resources
- Automated system maintenance,
- Fault tolerance, redundancy

Composability through abstraction

- Finding a conceptual domain where devices are homogeneous
 - Possibly the lowest level of such domains
- Abstraction of computing capabilities
 - Virtualization (QEMU, Java, Python)
- Abstraction of physical capabilities
 - Sensor virtualization (SOS),
 - Feature discovery
- Abstraction of communication capabilities
 - Self-describing communication interface (SOS)

Sensor Observation Service (SOS)

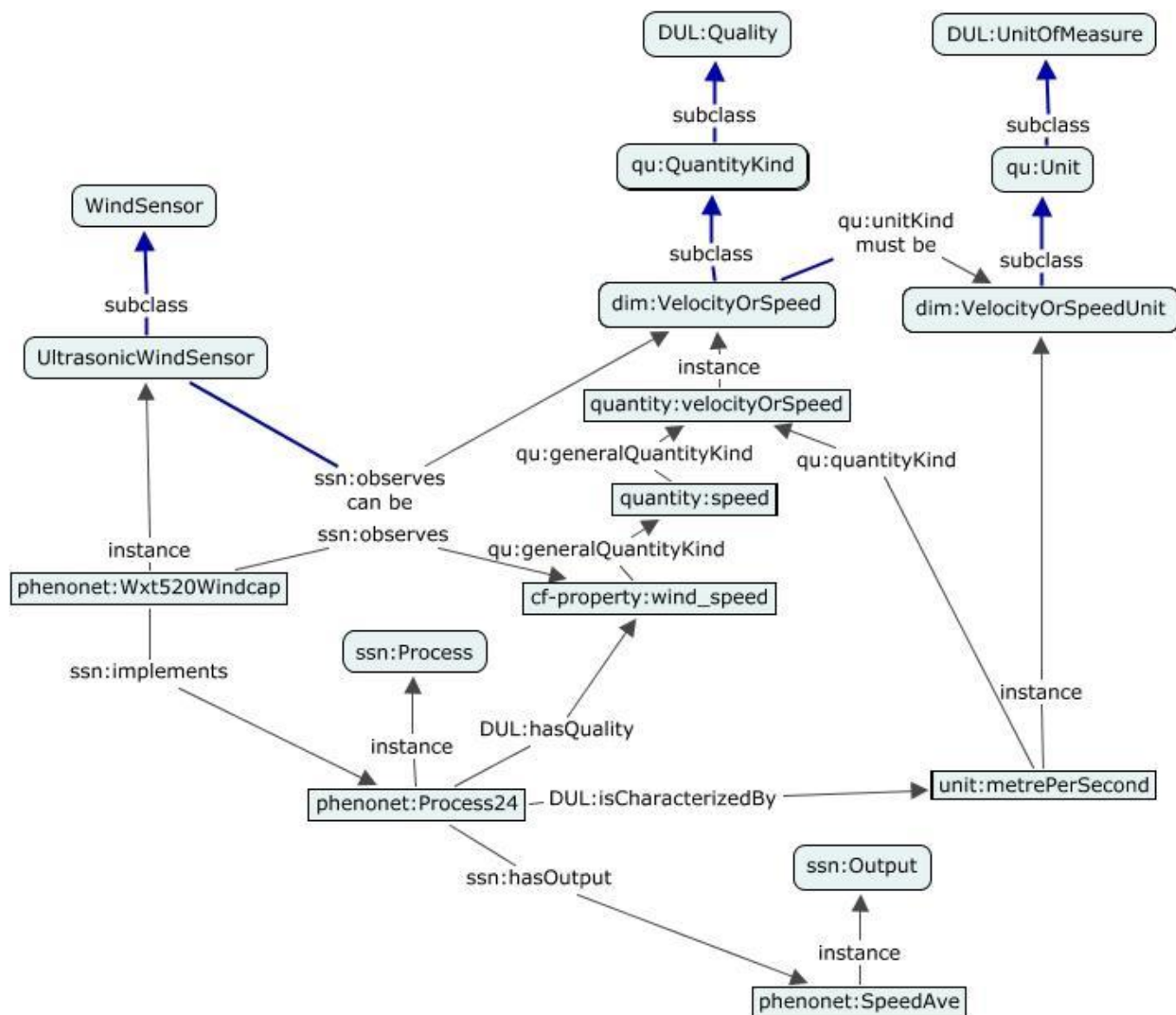
- Abstracts sensor data and communication
 - Self-describing sensor information database
 - Stores sensor data with geographic relevance
 - Efficient data queries
 - temporal or spatial filters
- Members of the CPS
 - direct communication with the SOS



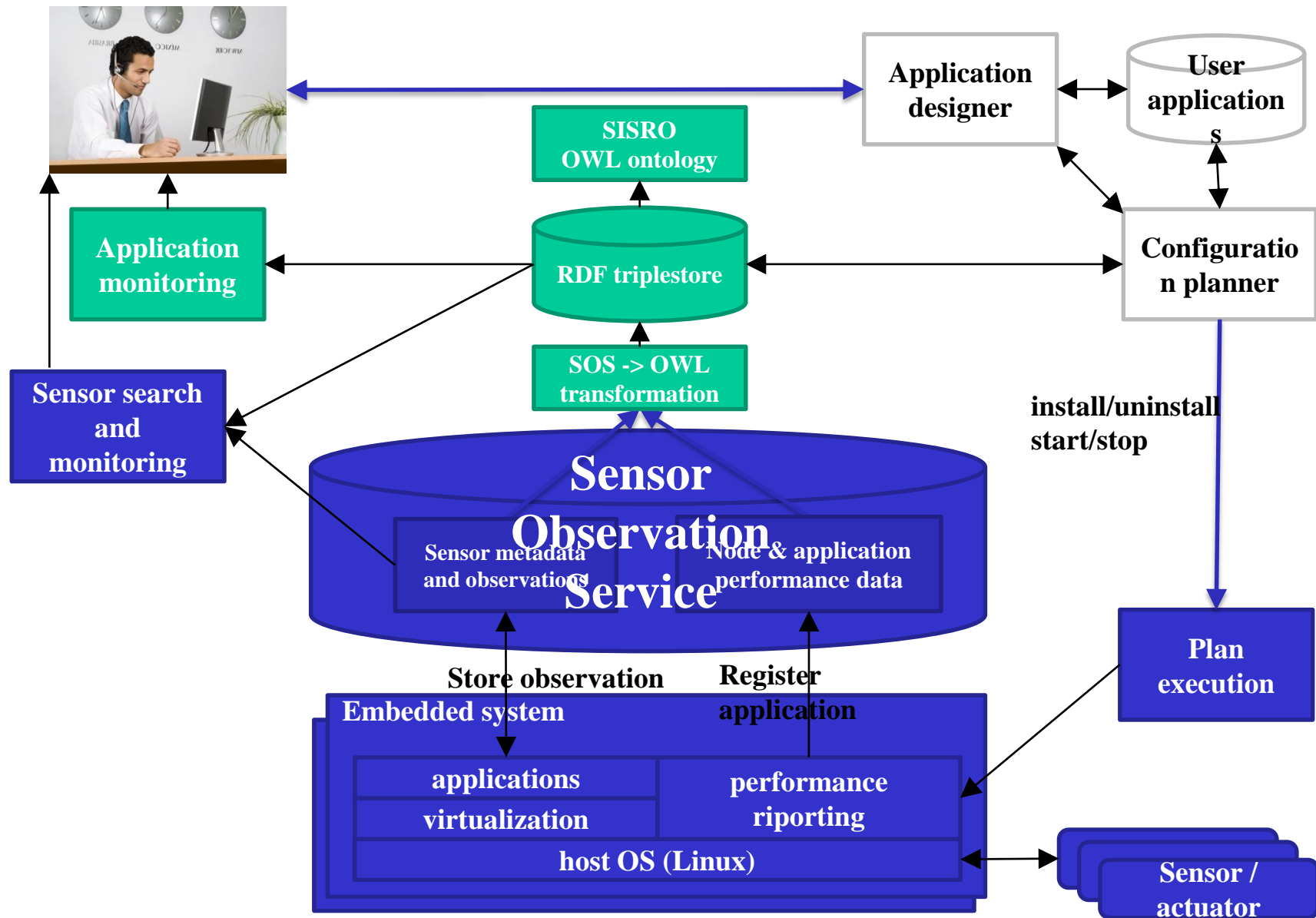
Semantic Sensor Network (SSN) ontology

- W3C Incubator Group (2009-2011)
- Capabilities of sensors and sensor networks
 - Formal ontology
- Covers:
 - system, deployment, sensing device, process
 - observed phenomenon (e.g. wind)
 - sensor type (e.g. ultrasonic wind sensor)
 - property (e.g. wind direction)
 - meaning (e.g. blows from direction)
 - unit of measure (e.g. radian)
 - operating range (e.g. temperature, humidity, ...)

SSN example: wind sensor



Architecture

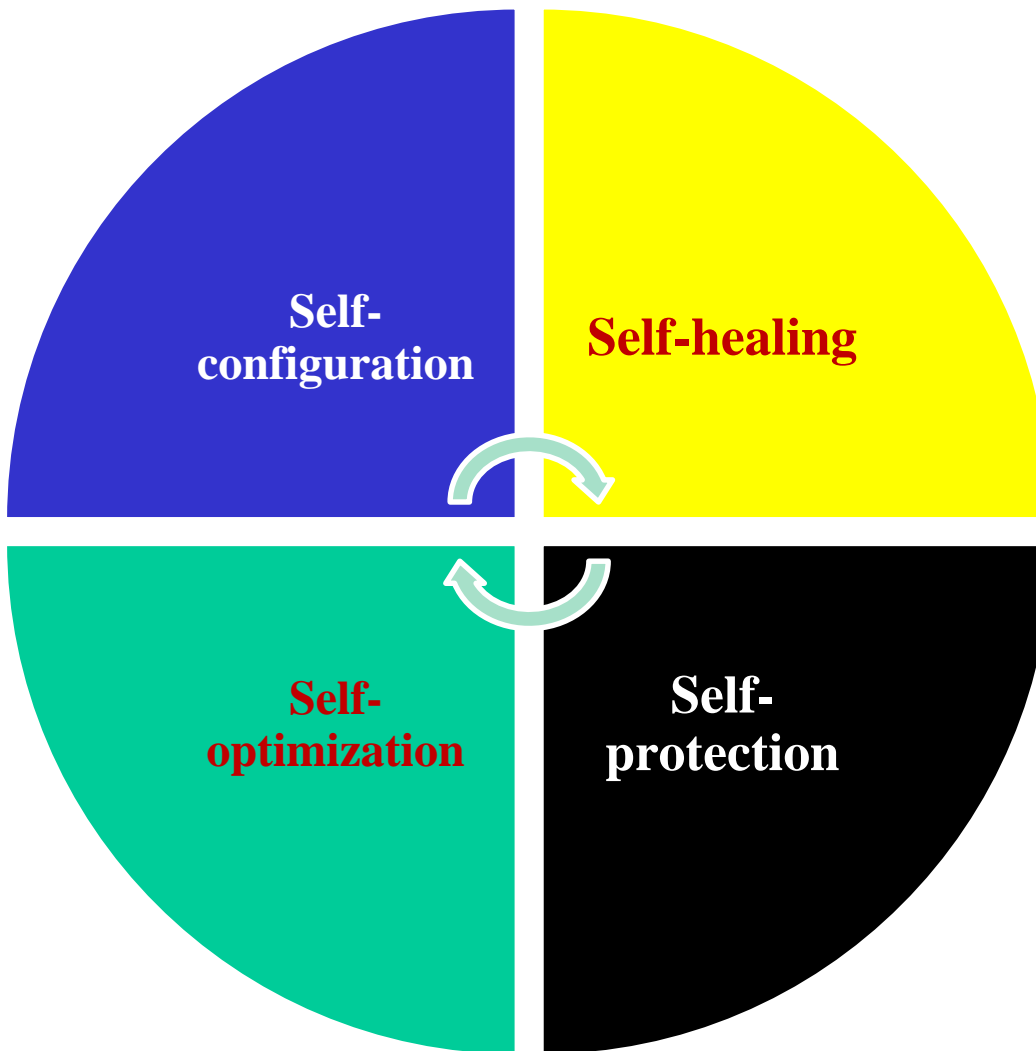


END OF LECTURE #2

Opportunities and threats in the cps paradigm

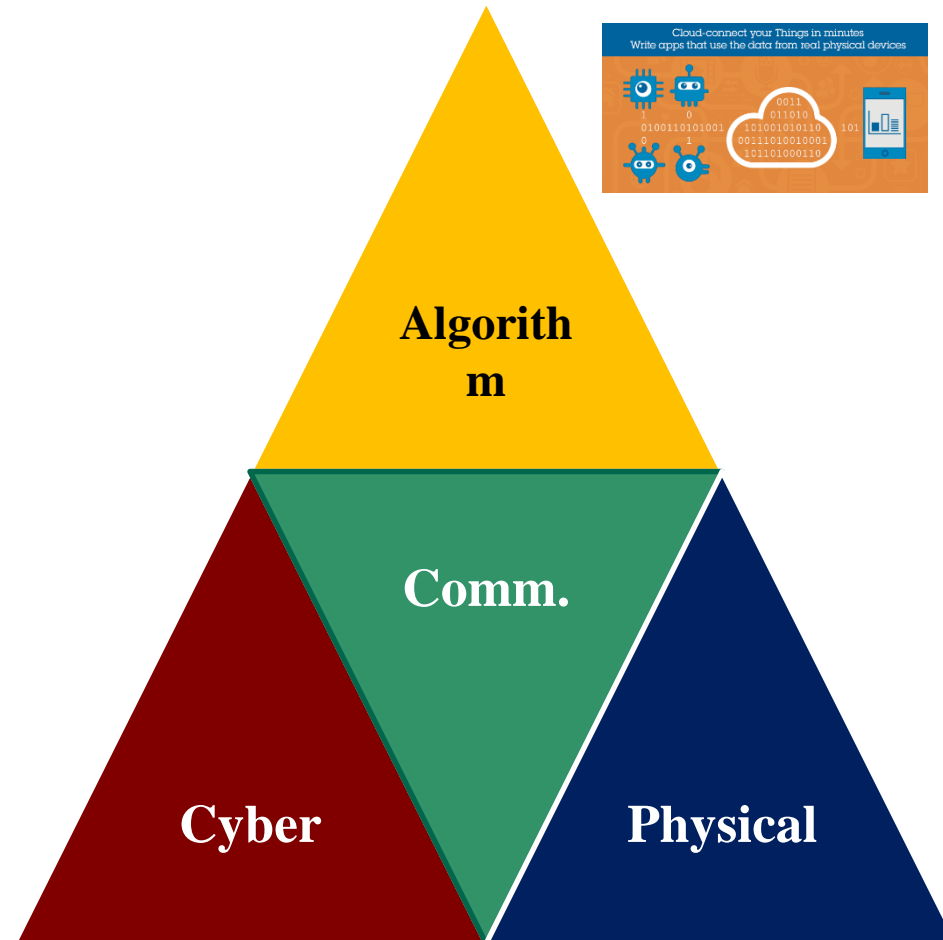
RESILIENCE

Self-* properties – dynamic challenges and solutions



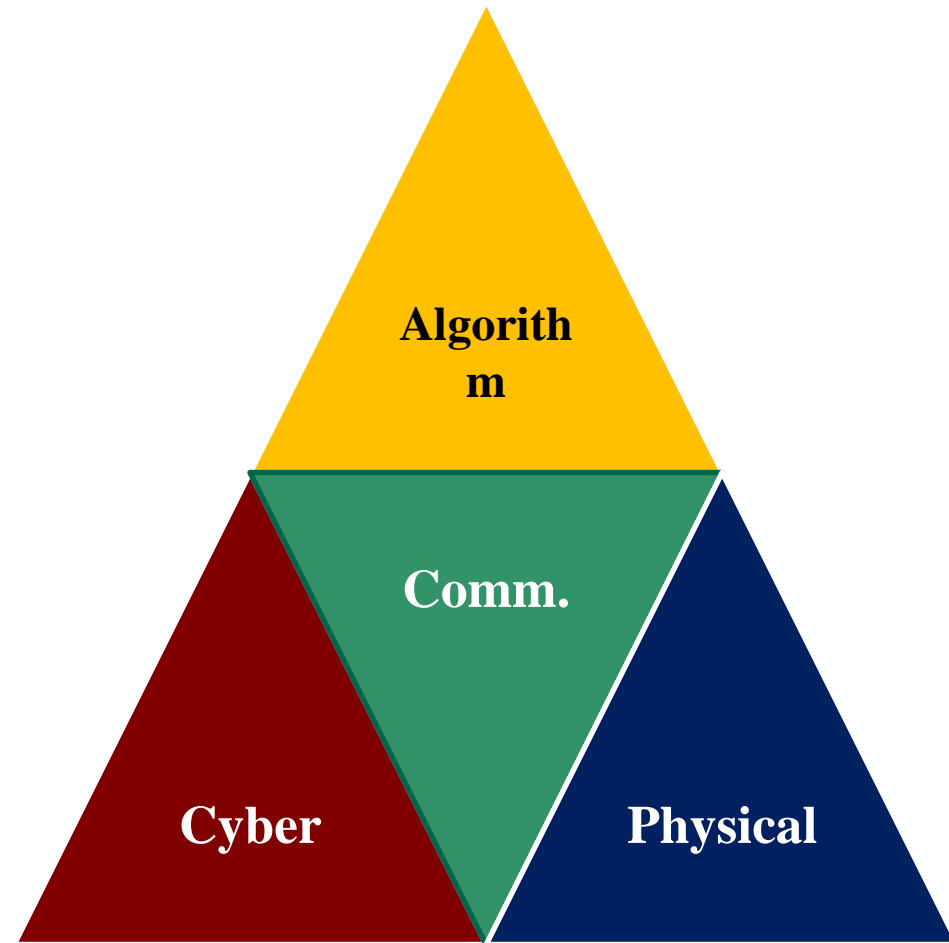
Opportunities-algorithmic diversity

- “Meta-algorithms”
- Different principles
 - Speed control in Italy:
 - Radar
 - Laser
 - TUTOR
 - Resource requirements
- External providers
 - AaaS – algorithm as a service
 - External validator

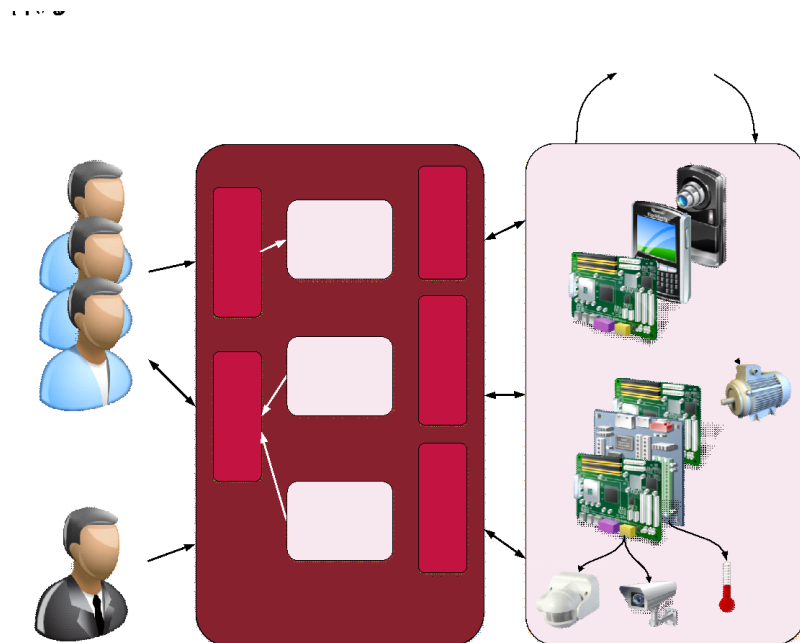
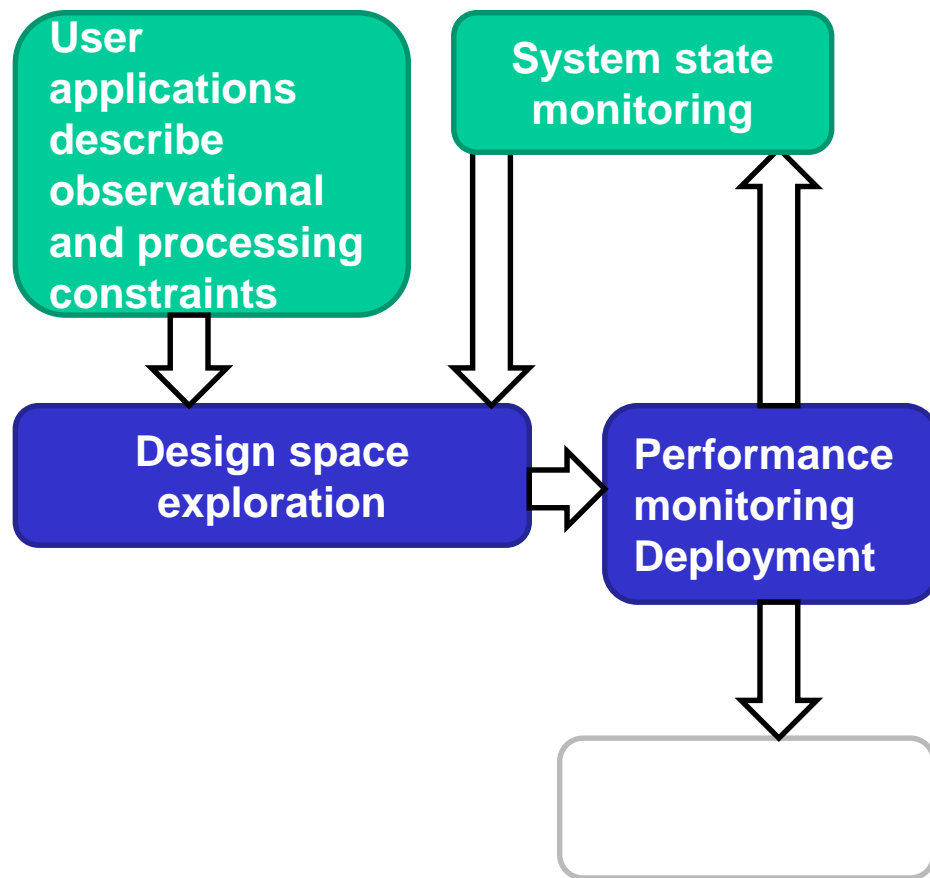


Opportunities- resource redundancy

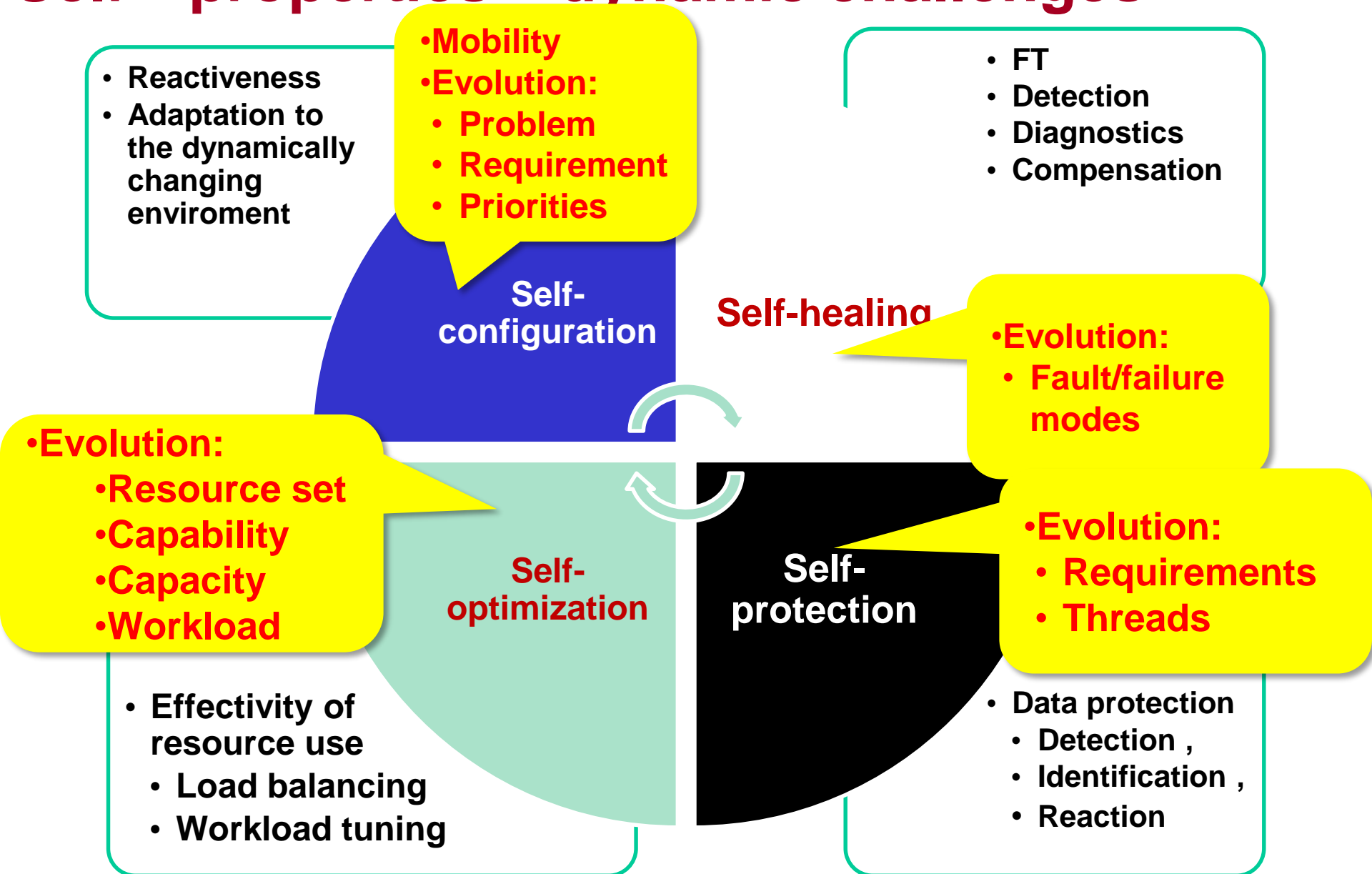
- Cheap computational redundancy, but
 - Depends on the reservation policy
- Virtualized network (SDN)
 - Fast failover
- Cheap sensors
 - Multitude of sensors



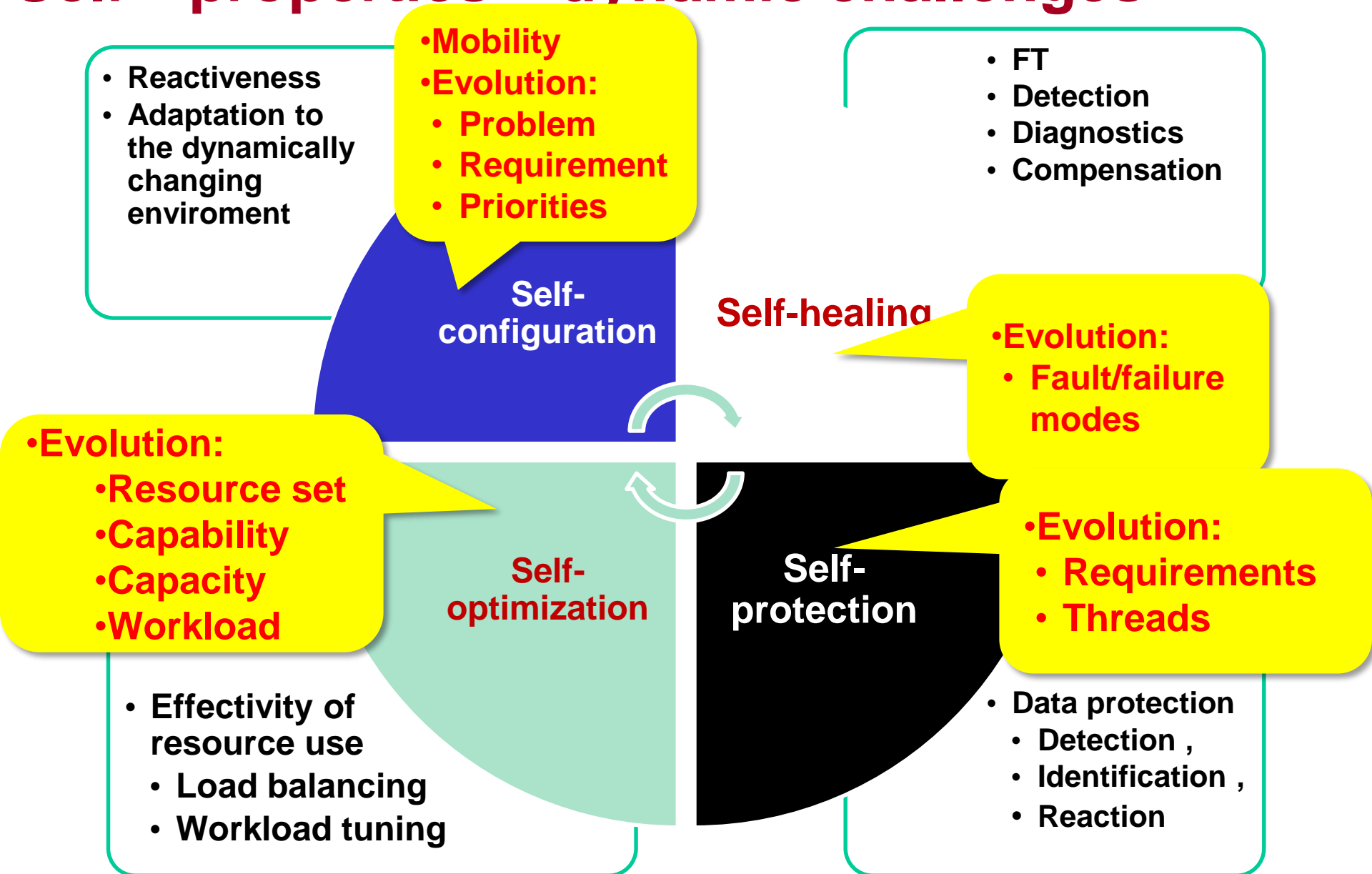
Dynamic reconfiguration of resources



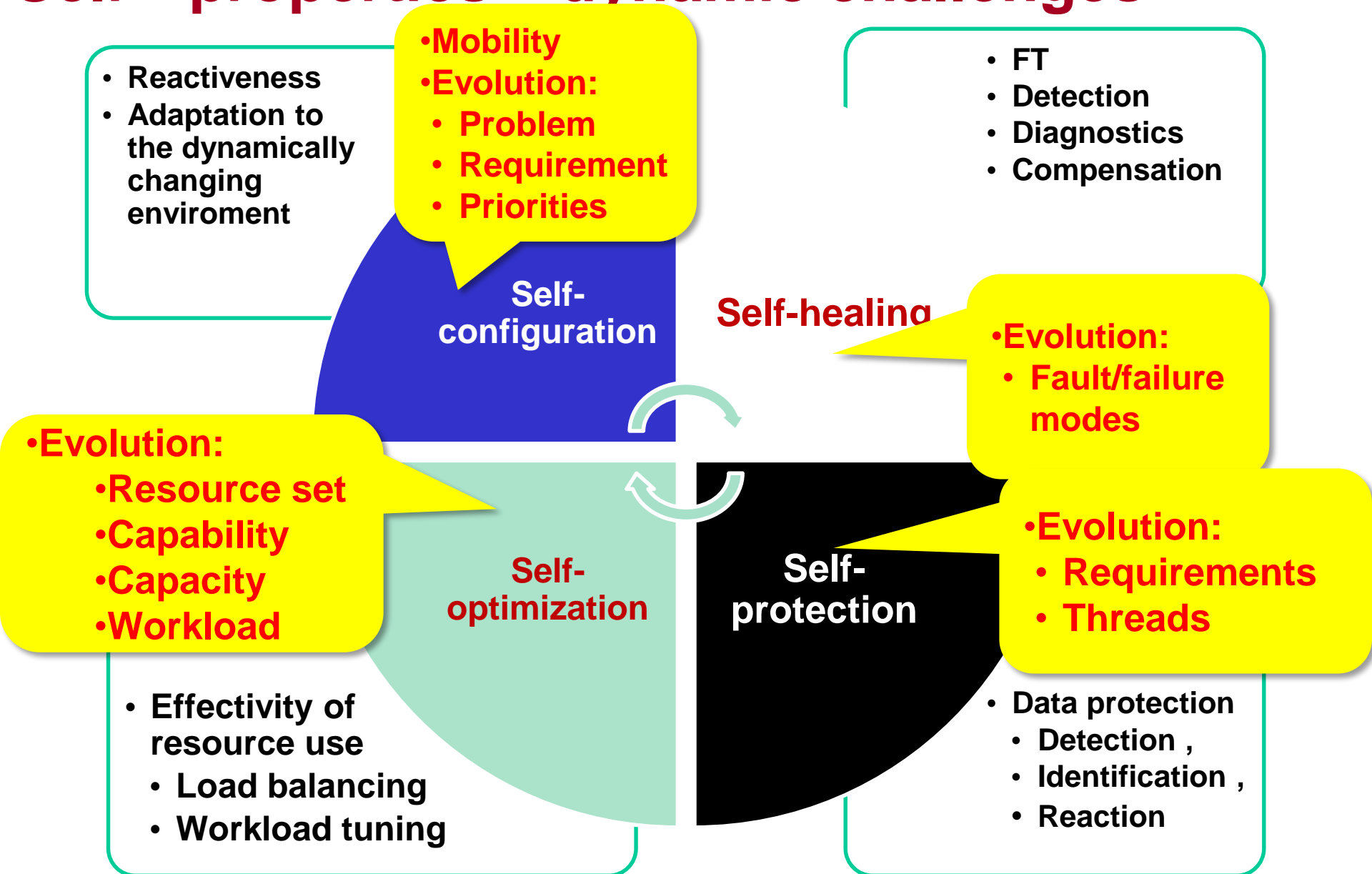
Self-* properties – dynamic challenges



Self-* properties – dynamic challenges



Self-* properties – dynamic challenges



Safety-critical systems are

- protected against **worst-case technical faults**, but
- unprotected against **malicious attacks**

**THE HORRIBLE MOTIVATION
- NEW DANGERS ARE HERE...**

Polish teen derails tram after hacking train network

A 14-year-old Polish boy turned the tram system in the city of Lodz into his “train set”.

- a modified TV remote control to change track points, and derailed four vehicles.
- Twelve people injured.

Past ES products in service without the full spectrum of extrafunctional properties as design aspects

The Telegraph



Malware implicated in fatal Spanair plane crash

Authorities investigating the 2008 crash of Spanair flight 5022 have discovered a central computer system used to monitor technical problems in the aircraft was infected with malware.

An internal report issued by the airline revealed the infected computer failed to detect three technical problems with the aircraft, which if detected, may have prevented the crash.

ES: long life span

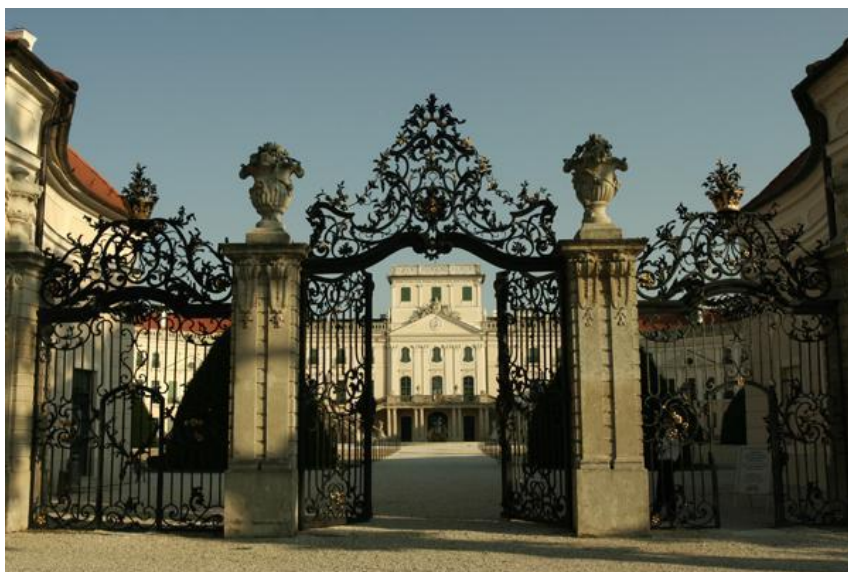
Security: evolving threats

Flight 5022 crashed after takeoff from Madrid-Barajas International Airport two years ago today, killing 154 and leaving only 18 survivors.

Safety contra security?

Safe, but not secure

- People may escape danger from inside



Secure, but not safe

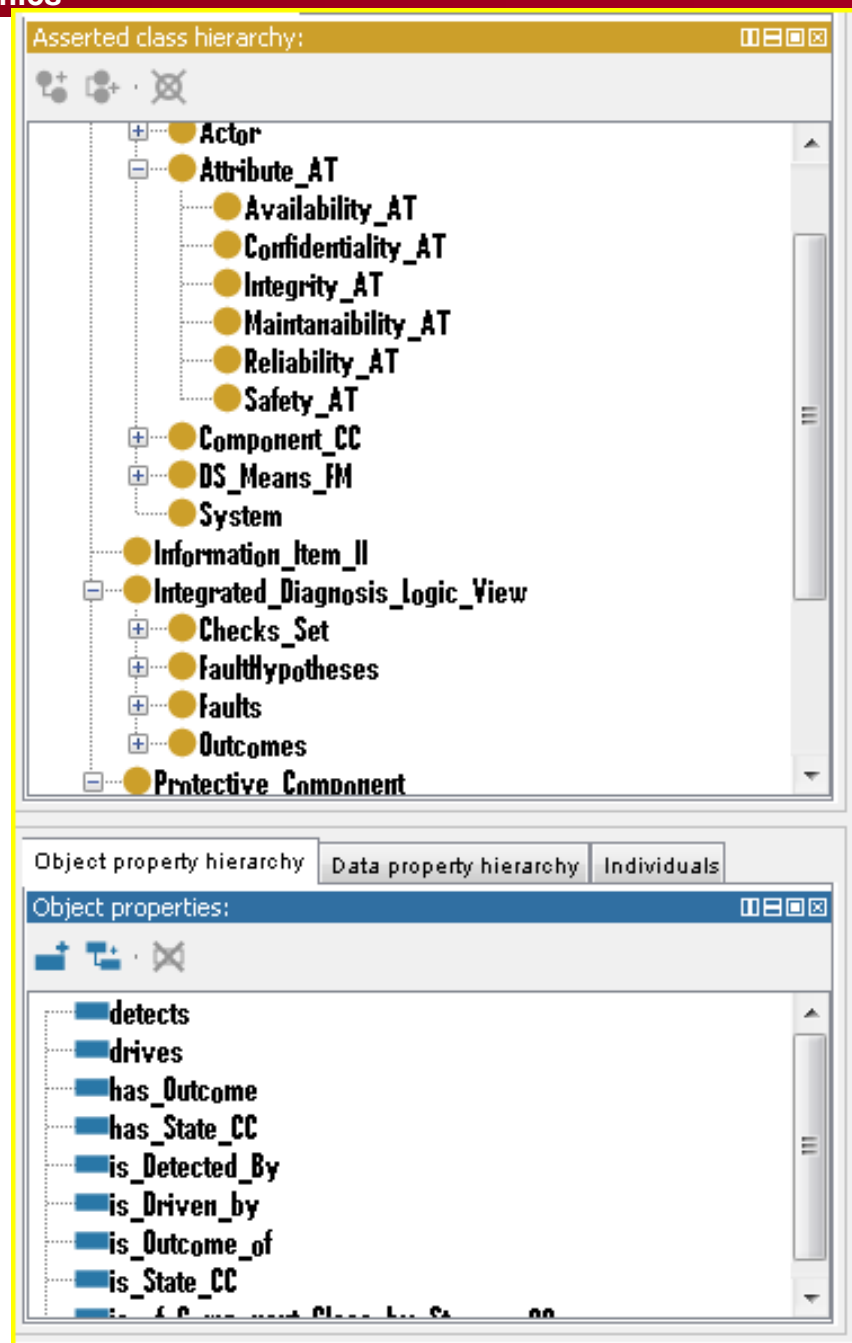
- No intruder can enter the gate



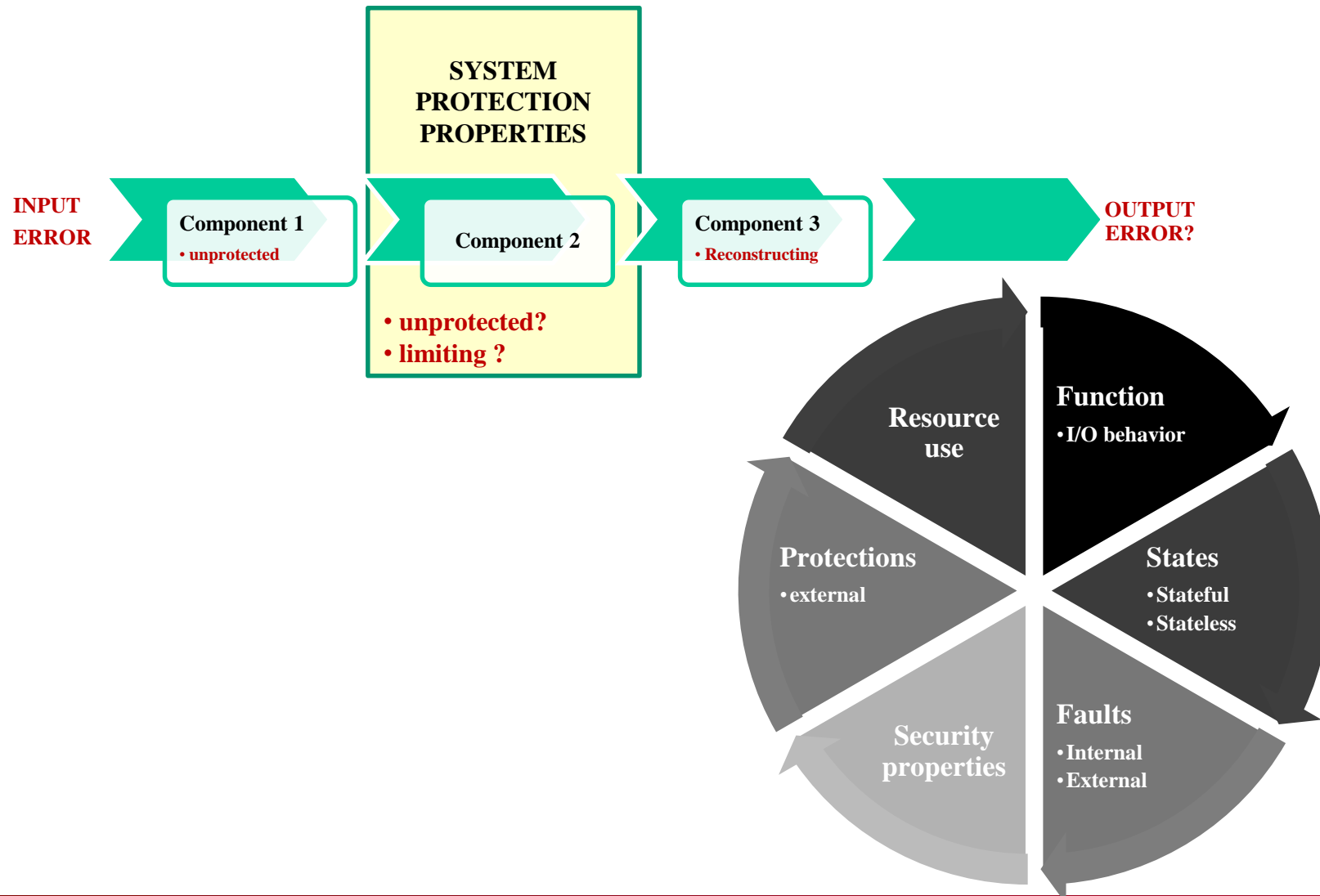
Specialization: error propagation/ protection

Introduces:

- Security aspects
- Protection profiles
- Error propagation attributes



System level fault impact analysis



Safety vs. security analysis

	Safety	Security
Fault	HW/SW Unintentional defects LIMITED FAULTS	Intrusion
Error	Distorted values/states	
Failure	Critical failure	
Propagation model	Functional/ architectural	Functional/ architectural + attack surface

Dependability/security problems and analysis

