Basics on Cyber-Physical SoSs

Dipartimento di Matematica e Informatica, Universita' di Firenze

bondavalli@unifi.it, andrea.ceccarelli@unifi.it

The SoSs Era

Computing evolution

- Mainframe computing (60's-70's)
  - Large computers to execute big data processing applications

- Desktop computing & Internet (80's-90's)
  - One computer at every desk to do business/personal activities
- Ubiquitous computing (00's)
  - Numerous computing devices in every place/person
  - "Invisible" part of the environment
  - Millions for desktops vs. billions for embedded processors

- Cyber Physical Systems (10's)

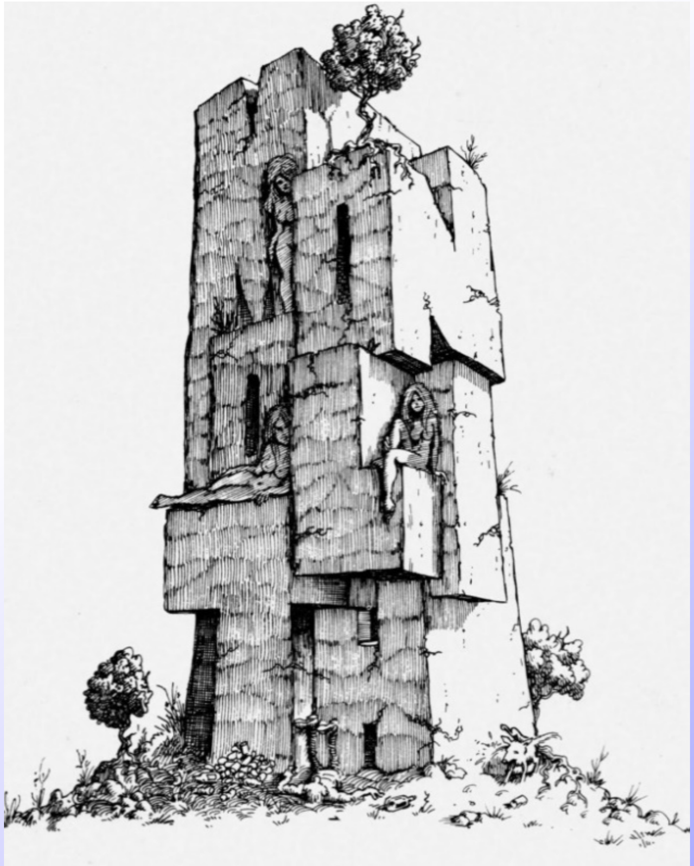
What are Cyber-Physical Systems?

- Cyber – computation, communication, and control that are discrete, logical, and switched
- Physical – natural and human-made systems governed by the laws of physics and operating in continuous time
- Cyber-Physical Systems – a system consisting of a computer system (the cyber system), a controlled object (a physical system) and possibly of interacting humans.

- "*CPS will transform how we interact with the physical world just like the Internet transformed how we interact with one another.*" [Fei Hu. Cyber-Physical Systems. CRC press. 2013]

From Monolithic Systems …

- Starting from **MainFrames**, computers were usually characterized by distinguishable services that are not clearly separated in the implementation but are interwoven,
- for example
  - data input and output,
  - data processing,
  - error handling, and
  - the user interface,
- rather than containing separate components



- Such "monolithic" architecture defines the **monolithic software systems**

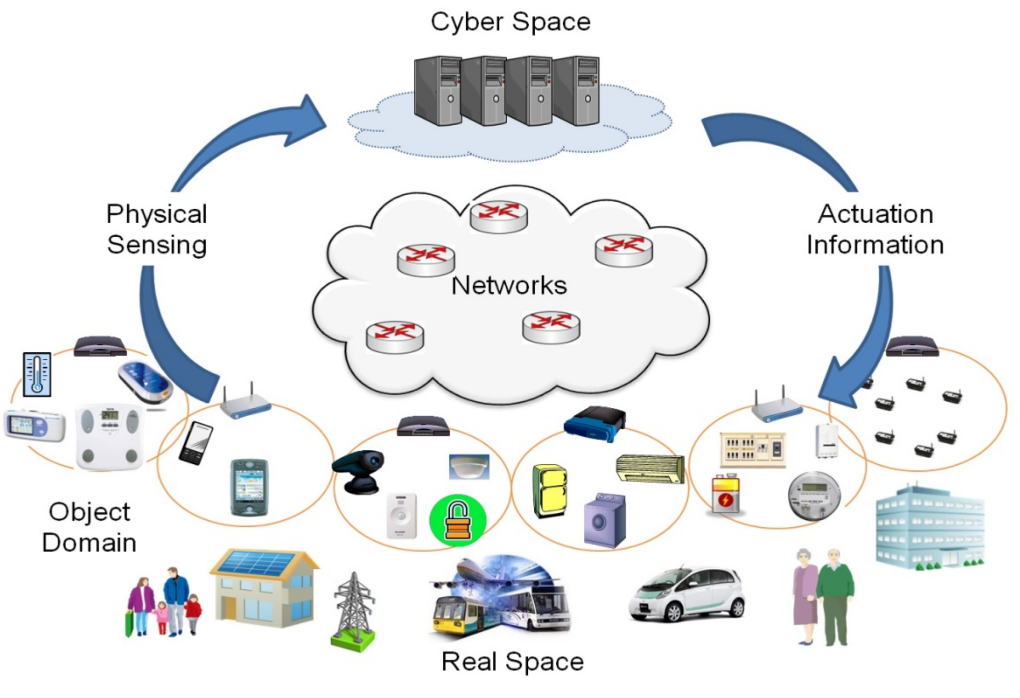
… to Systems of Systems (SoSs)

- However, many of the established assumptions in classical system design, such as
  - the scope of the system is known,
  - the design phase of a system is terminated by an acceptance test or
  - faults are exceptional events,
- are not always justified in modern cyber-physical software systems.

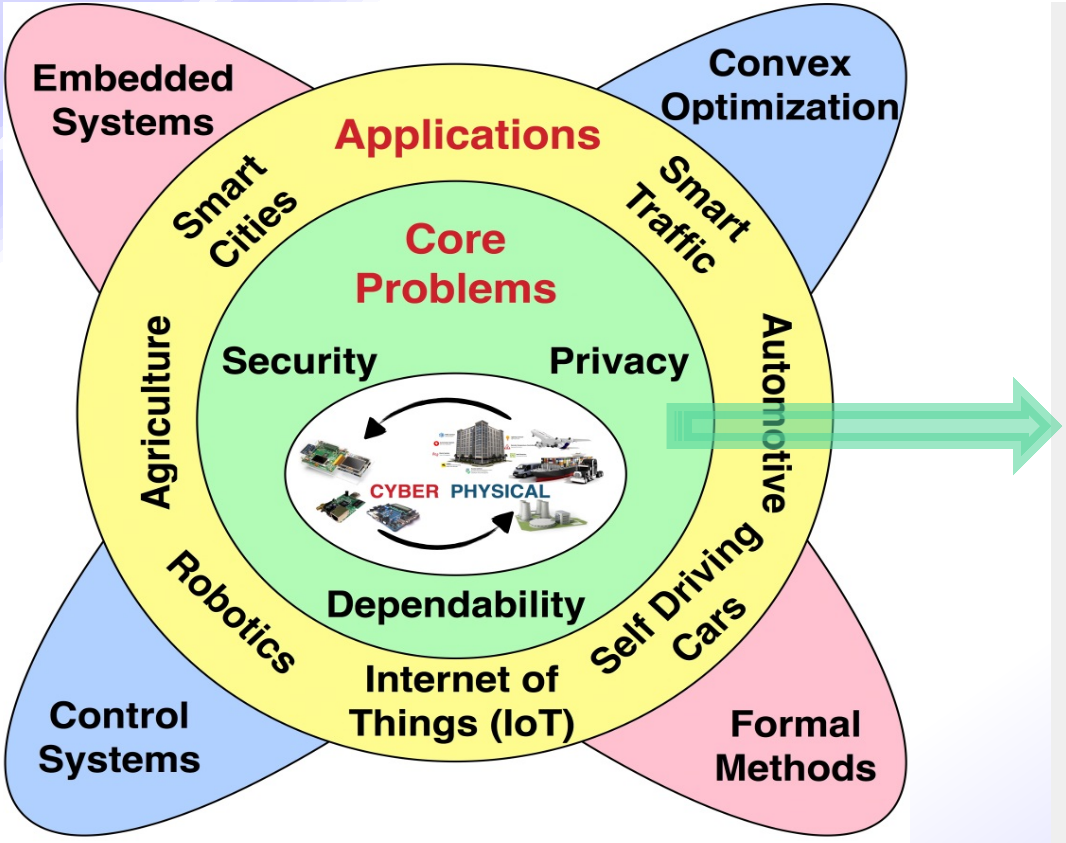
A **System of System (SoS)** stems from the integration of existing systems (legacy systems), normally operated by different organizations, and new systems that have been designed to take advantage of this integration.

CPS view

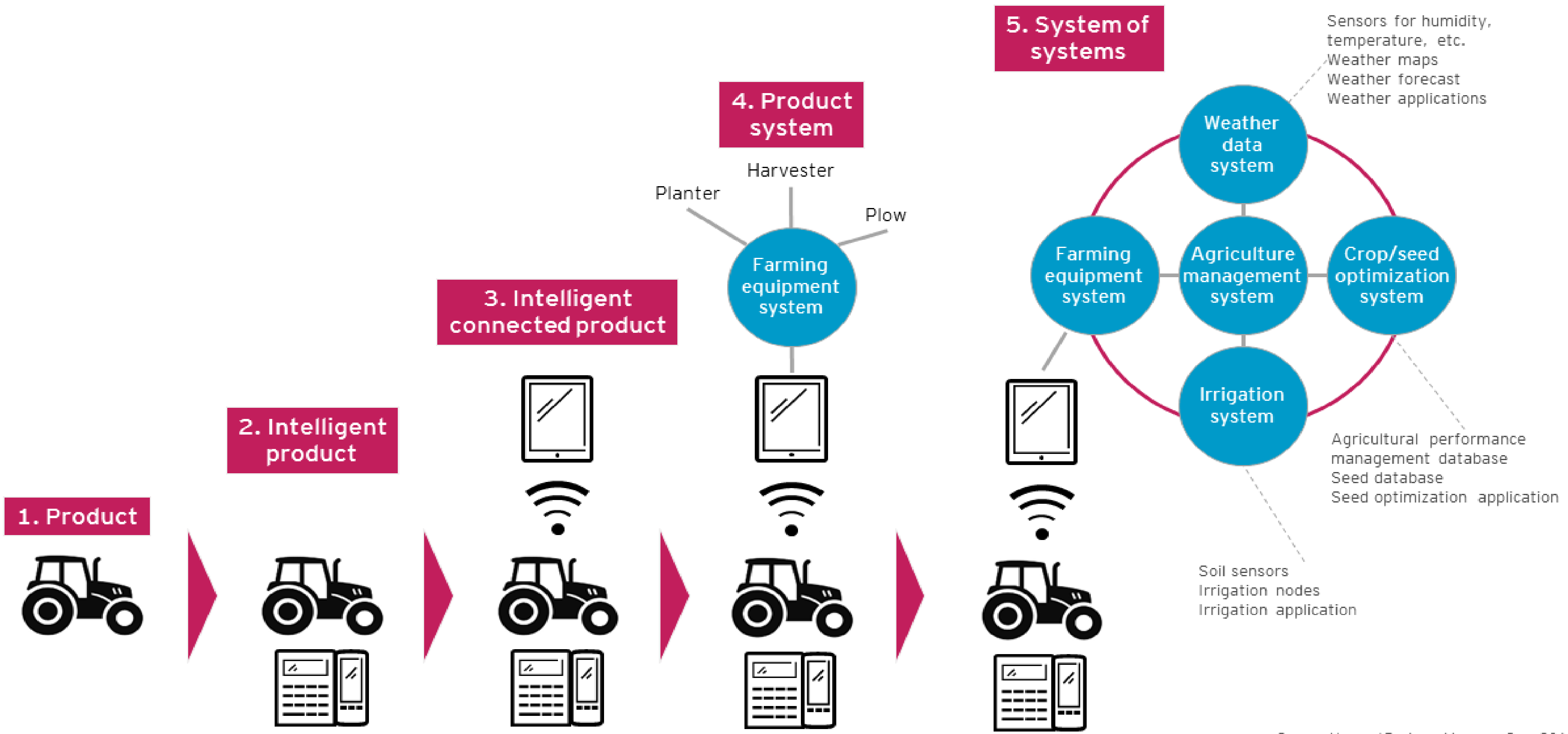
- CPSs are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core.



Core problem: how to make such systems *Secure*, *Resilient*, ...

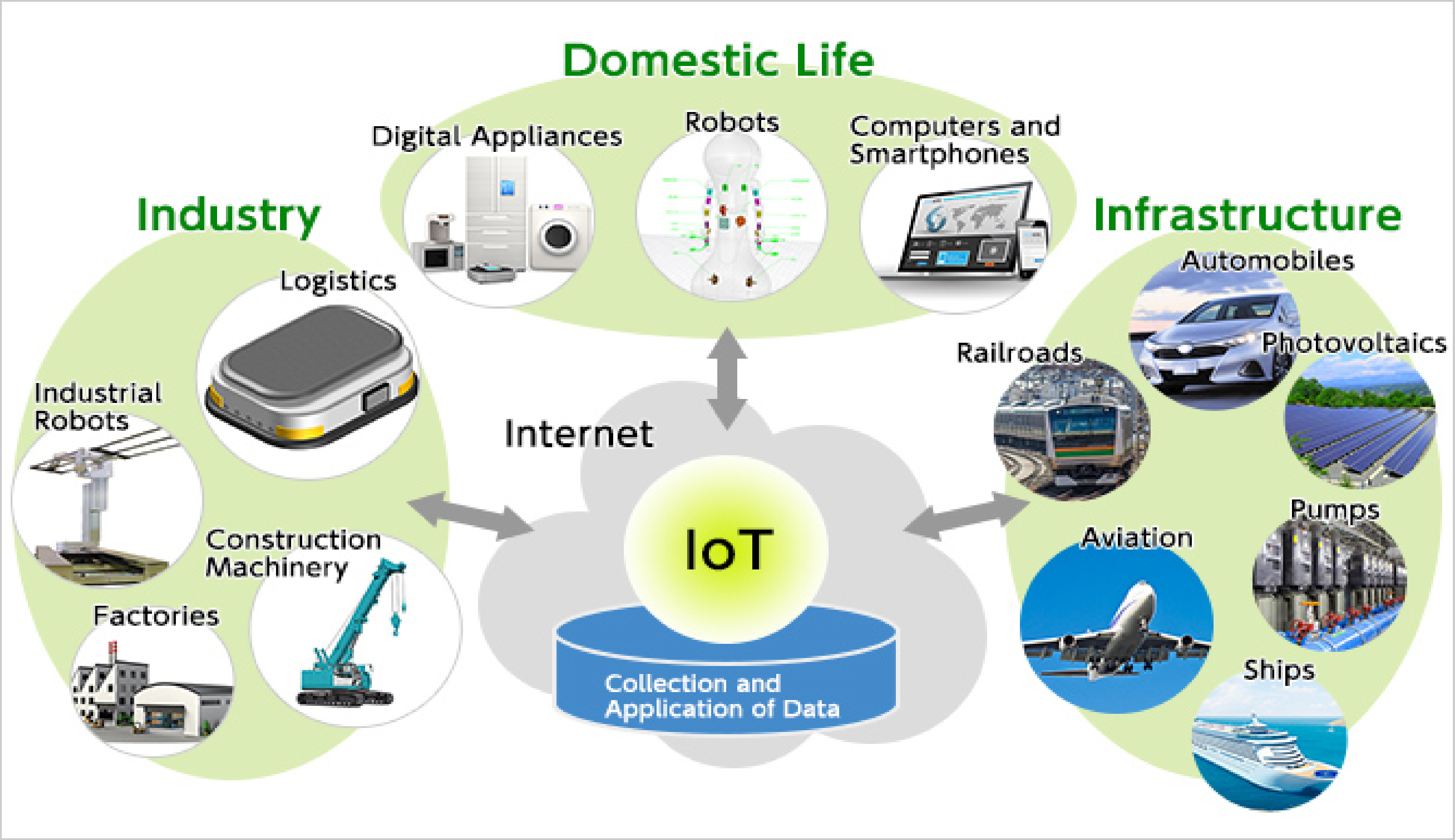


CPS... and related terms: *(Cyber-Physical) Systems of Systems*



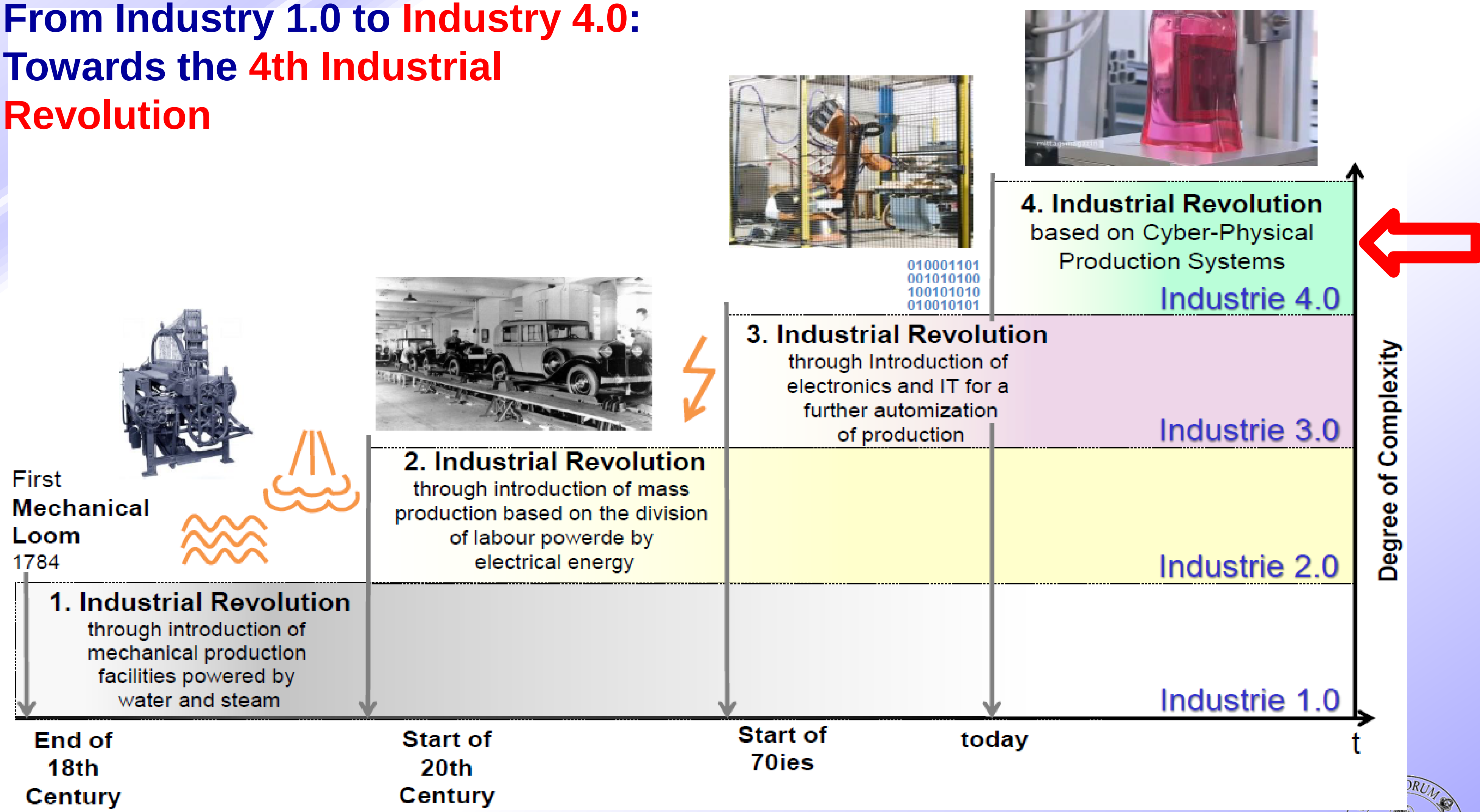
Source: Harvard Business Manager, Dec. 2014

CPS... and related terms: *Internet of Things*



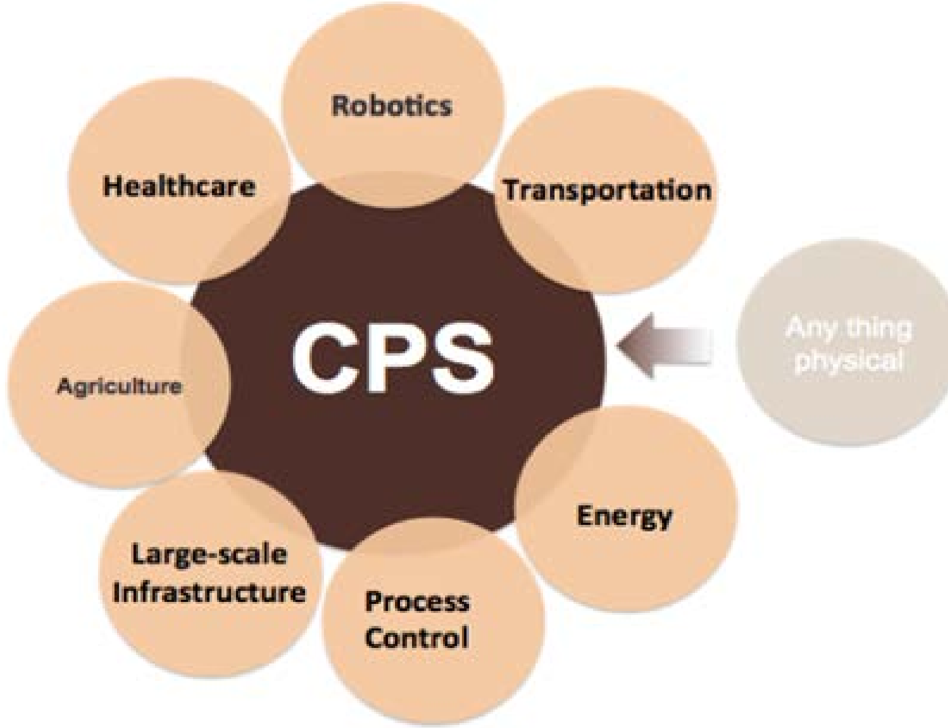
CPS & Industry 4.0

From Industry 1.0 to Industry 4.0: Towards the 4th Industrial Revolution



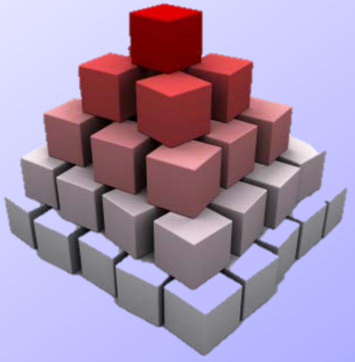
Application Domains of Cyber-Physical Systems

- Healthcare
  - Medical devices
  - Health management networks
- Transportation
  - Automotive electronics
  - Vehicular networks and smart highways
  - Aviation and airspace management
  - Avionics
  - Railroad systems
- Process control
- Large-scale Infrastructure
  - Physical infrastructure monitoring and control
  - Electricity generation and distribution
  - Building and environmental controls
- Defense systems
- Tele-physical operations
  - Telemedicine
  - Tele-manipulation



Multi-Level Hierarchy

- SoSs are characterized by a **multi-level** hierarchy, or rather a recursive structure where
  - a system, the whole at the level of interest (the macro-level), can be taken apart into
  - a set of subsystems, the parts, that interact statically or dynamically at the level below (the micro-level).
- Each one of these subsystems can be viewed as a system of their own.
- Recursion ends when the internals of a subsystem is of no further interest.
  - We call such a subsystem at the lowest level of interest - the base of the hierarchy - an elementary part or a **component**



Main Differences

The main differences between the two approaches can be summarized as follows

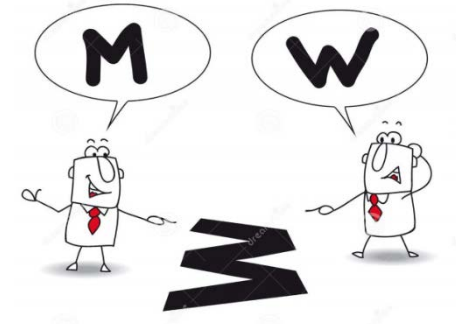
Characteristic	Monolithic	System-of-system
Scope of the System	Fixed (Known)	Unknown
Clock Synchronization	Internal	External e.g., GPS
Structure	Hierarchical	Networked
Requirements and Spec.	Fixed	Changing
Evolution	Version Control	Uncoordinated
Testing	Test Phases	Continuous
Implementation	Technology Given and Fixed	Unknown
Faults (Physical, Design)	Exceptional	Normal
Control	Central	Autonomous
Emergence	Insignificant	Important
System Development	Process Model	???

Viewpoints (I)

To reduce the *cognitive effort* needed to comprehend the behaviour of an SoS, its main characteristics can be summarized as viewpoints, or rather simplified dimensions of analysis:

- Fundamental System Concepts.
  - The definition of an SoS and its related parts
- Time.
  - The progression of time and its role in an SoS.
- Data and state.
  - the data and information exchanged between the parts of an SoS.
- Actions and Behaviour.
  - the dynamics of an SoS, either event-based view or a state-based view.

Viewpoints (II)



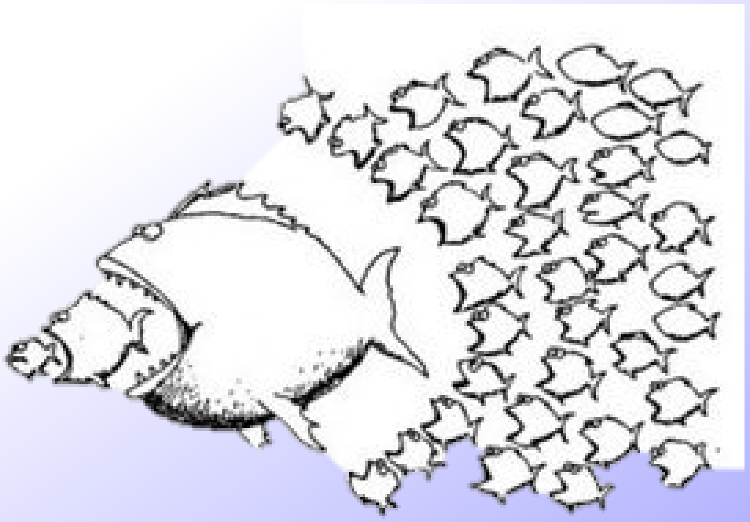
- Communications.
  - the role of a communication system in an SoS.
- Interfaces.
  - the interaction of components with each other and with the environment
- Evolution and Dynamicity.
  - SoS dynamicity, intended as short term changes, and evolution (long term changes).
- System design and tool.
  - The concepts to define design methodologies to engineer SoSs.
- Dependability and Security.
  - Dependability and security concepts, in compliance with existing taxonomies.
- **Emergence.**

Emergence

- We defined a SoS as an integration of existing (either cyber of physical) subsystems. However, the SoS is not just the sum of its components.

Emergence: a phenomenon of a whole at the macro-level
is emergent if and only if it is of a new kind with respect to the non-relational phenomena of any of its proper parts at the micro level".

- Emergent phenomena can be
  - either beneficial or detrimental, and
  - either expected or unexpected.
- Managing emergence is
  - Essential to avoid undesired, possibly unexpected situations
  - Usually the higher goal of an SoS



Basic Concepts

Fundamentals

Domain: The Domain comprises the set of entities and the relations among the entities that are of interest when modeling the selected view

- To structure the domain, we must identify objects that have a distinct and self-contained existence.
- **Entity**: Something that exists as a distinct and self-contained unit.
  - **Thing**: A physical entity that has an identifiable existence in the physical world.
  - **Construct**: A non-physical entity, a product of the human mind, such as an idea.

System

System: An entity that is capable of interacting with its environment and may be sensitive to the progression of time (*from EU Project DSOS*).

- Note that the system may react differently, to the same pattern of input activity, depending on the environment e.g., a time-controlled heating system.

Environment of a System: The entities and their actions in the domain that are not part of a system but have the capability to interact with the system.

- System and Environment are separated by a **System Boundary**, a dividing line between two systems or between a system and its environment.

System Architecture

System Architecture: The blueprint of a design that establishes the overall structure, the major building blocks and the interactions among these major building blocks and the environment.

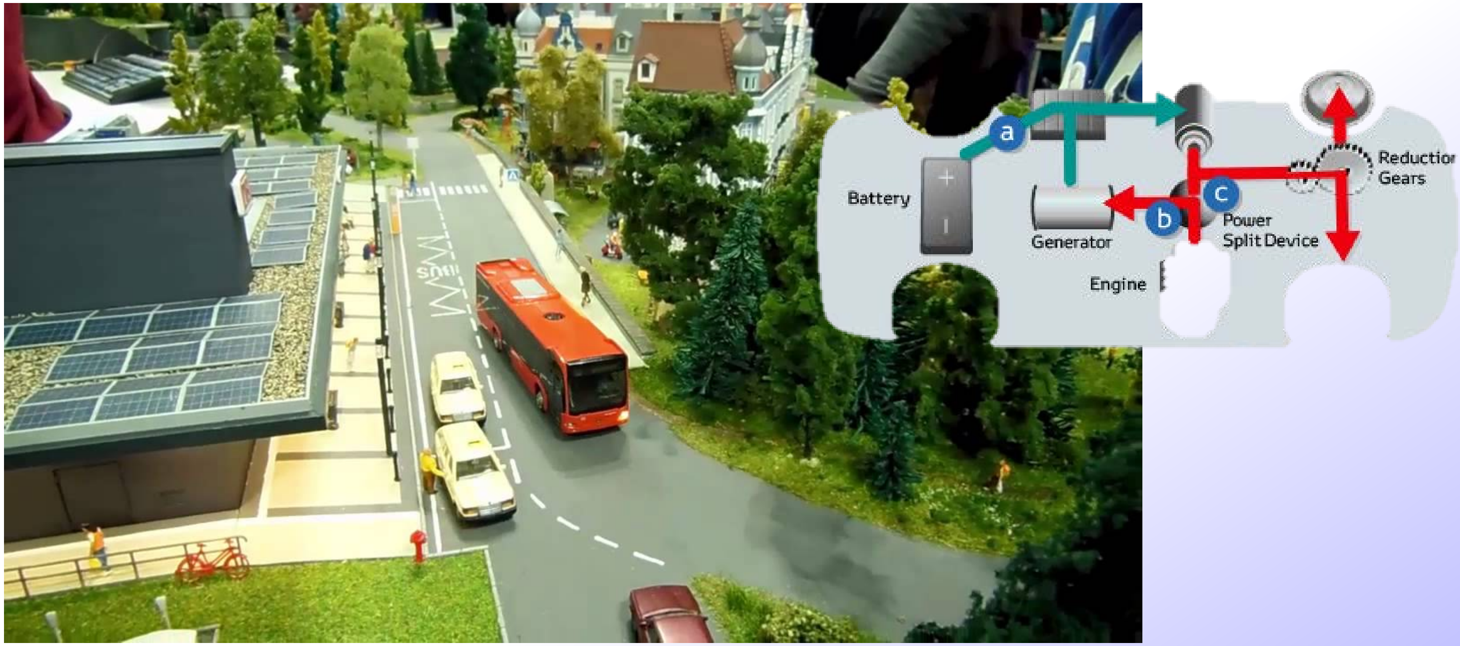
- When designing the system, every organization that develops a system follows a set of explicit or implicit rules and conventions, such as
  - naming conventions,
  - representation of data (e.g., endianness of data),
  - protocols

- These explicit or implicit rules and conventions are called the **architectural style**.

Boundaries in SoSs

In SoS Engineering, such an approach can be problematic, because in many SoS the system boundary may change frequently.

- Consider a car-to-car SoS that consists of a plurality of cars cruising in an area.

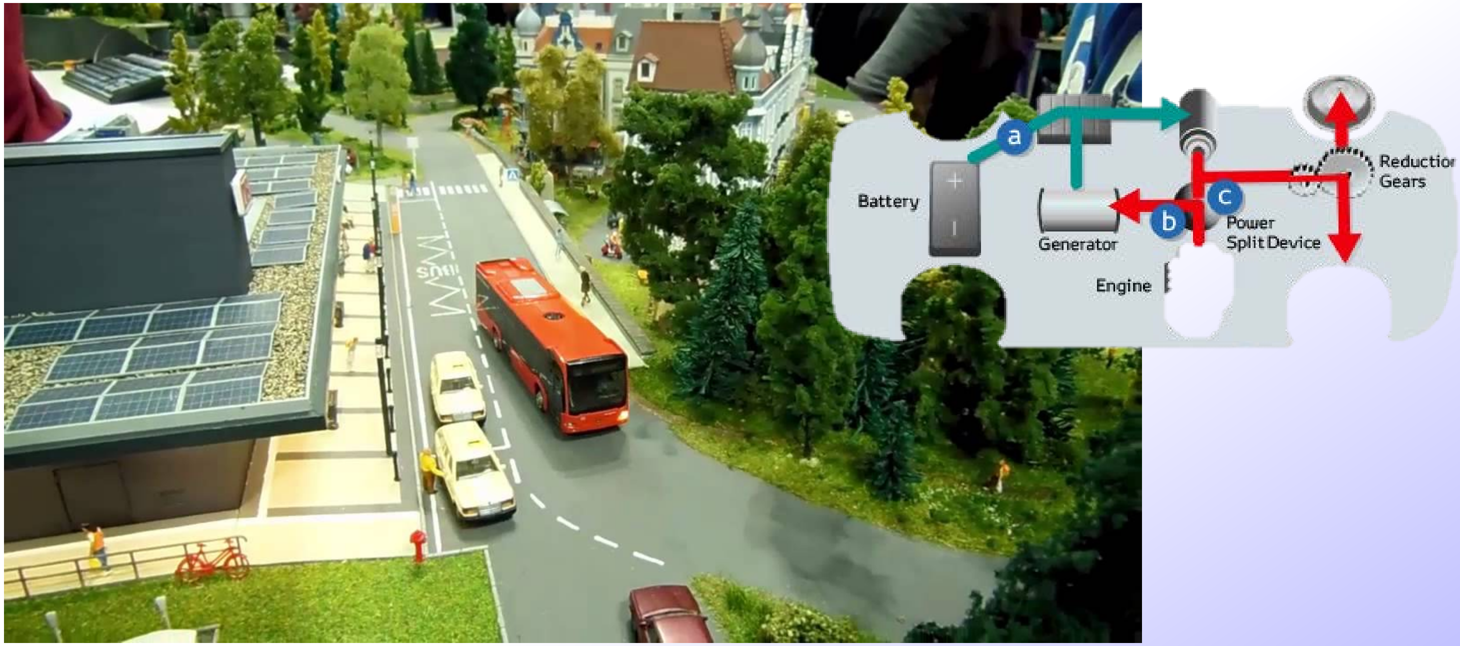


Where is the boundary of such an SoS?

Boundaries in SoSs

In SoS Engineering, such an approach can be problematic, because in many SoS the system boundary may change frequently.

- Consider a car-to-car SoS that consists of a plurality of cars cruising in an area.



Where is the boundary of such an SoS?

Answer: it is hardly possible to define a stable boundary of an SoS

Autonomous System

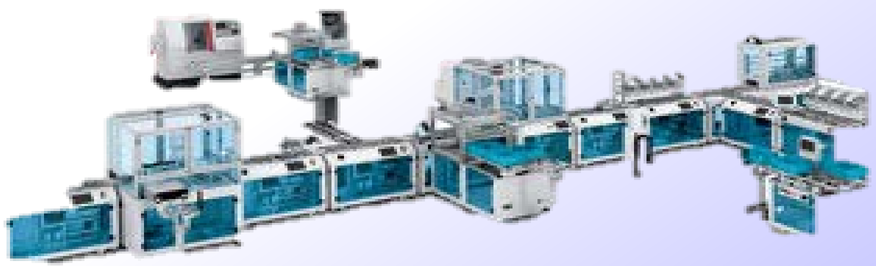
- In the above example of a car-to-car SoS each individual car in the system
  - consisting of the mechanics of the car, the control system, and the driver
- Can be considered as an autonomous system that tries to achieve its given objective without any control by another system.

Autonomous System: A system that can provide its services without guidance by another system.

Cyber-Physical System

- Many systems are composed of (autonomous) interrelated parts, each of them hierarchic in structure until some lowest level of elementary subsystem, a subordinate system that is a part of an encompassing system.

Constituent System (CS): An autonomous subsystem of an SoS, consisting of computer systems and possibly of controlled objects and/or human role players that interact to provide a given service.



Cyber-Physical System (CPS): A system consisting of a computer system (the cyber system), a controlled object (a physical system) and possibly of interacting humans

Systems of Systems

System-of-Systems (SoS): An SoS is an integration of a finite number of constituent systems (CS) which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal (*Jamshidi*).

Note: boundaries are defined for a period of time, than they may change

- **Directed SoS**: An SoS with a central managed purpose and central ownership of all CSs. An example would be the set of control systems in an unmanned rocket.
- **Acknowledged SoS**: Independent ownership of the CSs, but cooperative agreements among the owners to an aligned purpose.
- **Collaborative SoS**: Voluntary interactions of independent CSs to achieve a goal that is beneficial to the individual CS.
- **Virtual SoS**: Lack of central purpose and central alignment.

Managing Time

Notion of Time

- In the (Cyber-Physical) SoS paradigm we start being concerned with change, that depends on the progression of time.
- In an SoS a global notion of time is required in order to
  - Enable the interpretation of timestamps in the different CSs.
  - Limit the validity of real-time control data.
  - Synchronize input and output actions across nodes.
  - Provide conflict-free resource allocation.
  - Perform prompt error detection.
  - Strengthen security protocols.

Time Cycle

Time: A continuous measurable physical quantity in which events occur in a sequence proceeding from the past to the present to the future.

- Timeline: A dense line denoting the independent progression of time from the past to the future.
  - Instant: A cut of the timeline.
  - Event: A happening at an instant.

- Cycle: A temporal sequence of events that arrives at a final state related to the initial state, from which the temporal sequence of events can be re-started
  - An example for a cycle is the rotation of a crankshaft in an automotive engine.
  - Although the duration of the cycle changes, the sequence of the significant events during a cycle is always the same.

Periodic Systems

Period: A cycle marked by a constant duration between the related states at the start and the end of the cycle.

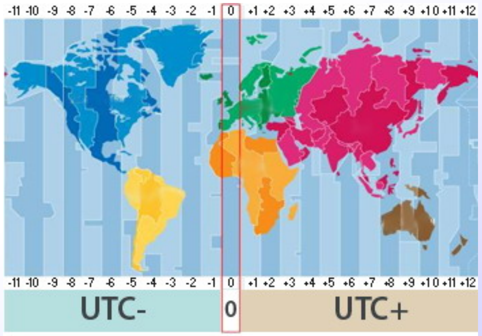
- Periodic Systems are of utmost relevance in control applications

Periodic System: A system where the temporal behaviour is structured into a sequence of periods.

- Periodicity is not mandatory, but often assumed as it leads to simpler algorithms and more stable and secure systems
  - the difference between cycle and period is the constant duration of the period.

Time Standards

- The physical second is the same in all UTC, TAI and GPS time standards
- UTC (Universal Time Coordinated) is an astronomical time standard aligned with the rotation of the earth.
  - Since the rotational speed of the earth is not constant, it was decided to base the SI second on atomic processes establishing the International Atomic Time TAI (*Temps Atomique International*).
    - On January 1, 1958 at 00:00:00 TAI and UTC had the same value.
    - TAI is distributed world-wide by the GPS (Global Positioning System) satellites.
  - GPS represents the TAI time in weeks and full seconds within a week.
    - The week count is restarted every 1024 weeks, i.e., after 19.6 years.



Coordinated Clocks

Clock: A (digital) clock is an autonomous system that consists of an oscillator and a register. Whenever the oscillator completes a period, an event (**tick**) is generated that increments the register.

- **Reference clock**: A hypothetical clock of a granularity smaller than any duration of interest and whose state is in agreement with TAI.
  - the reference clock has small granularity that digitalization errors are neglected,
  - the reference clock can observe every event of interest without any delay and
  - the state of the reference clock is always in perfect agreement with TAI time.

Coordinated Clock: A clock synchronized within stated limits to a reference clock that is spatially separated

Clock Drift

- Every good (fault-free) free-running clock has an individual granularity that can be different from the nominal granularity.

Drift: The drift of a physical clock is a quality measure describing the frequency ratio between the physical clock and the reference clock.

- Since the drift of a good clock is a number close to 1, it is conducive to introduce a drift rate by

$$\text{Drift Rate} = |\,\text{Drift} - 1\,|$$

- Typical clocks have a drift rate of 10^{-4} to 10^{-8}.

Data and State

Data and Information

- Systems-of-Systems (SoSs) come about by the transfer of information of one Constituent System (CS) to another CS.
  - But what is information? How is information related to data?

Data: A data item is an artefact, a pattern, created for a specified purpose.

- In cyber space, data is represented by a bit-pattern. To expand the meaning of the bit pattern we need to understand how to interpret the given bit pattern.

Information: A proposition about the state of or an action in the world.

Data Receivers

Such data can be intended either for a receiver human or a machine

- Human Receiver
  - the explanation must describe the data using concepts that are familiar to the intended human receiver.
- Machine
  - the computer instructions tell the computer system how the data bit-string is partitioned and how they have to be stored, retrieved, and processed.
  - the explanation of purpose is directed to humans who are involved in the design and operation of the SoS. Therefore, it should be understandable to the user/designer.

State

- Systems define their behaviour depending on interactions with the environment

State: The state of a system at a given instant is the totality of the information from the past that can have an influence on the future behaviour of a system.

- It is a data structure that characterizes the condition of a system at a given time.
- The concept of state is meaningless without a concept of time, since the distinction between past and future is only possible if the system is time-aware.
  - The variables that hold the stored state in a state-full system are **state variables**.

State Space: The state space of a system is formed by the totality of all possible values of the state variables within the domain

Actions and Behaviour

Event-Based View

We can observe the dynamics of a system that consists of discrete variables by an event-based view or by a state-based view.

- Event-based view:
  - we observe the value of relevant state variables at the beginning of the observation
  - record all events (i.e. changes of the state variables)
  - observe the time of occurrence of the events in a trace.
  - The value of all state variables at any past instant is defined by the recorded trace.
- However, if the number of events that can happen is not bounded, the amount of data generated by the event-based view **cannot be bounded**.

State-Based View

- Periodic State-based View (**Sampling**)
  - we observe the values of relevant state variables at selected observation instants (the sampling points) and record these values of the state variables in a **trace**.
  - The sampling interval, is critical for acquiring a satisfying image of the system.
  - The duration between two observation instants puts a limit on the amount of data generated by the state-based view.
  - Price to pay: events that happen between consequent samples may get lost.

Sampling: The observation of the value of relevant state variables at selected observation instants.

Execution Times

- **Execution Time**: The time needed to execute a specific action on a system.
  - The execution time depends on the performance of the available hardware and is also data dependent.

Worst Case Execution Time (WCET): The worst-case data independent execution time required to execute an action on a given computer.

- There are two possible sources for a start signal of an action.
  - Time-triggered (TT) Action: An action where the start signal is derived from the progression of time.
  - Event-triggered (ET) Action: An action where the start signal is derived from an event other than the progression of time.

Behaviour

- The behaviour of a system is of utmost interest to a user.
  - Function: A function is a mapping of input data to output data.
  - Behaviour: The timed sequence of the effects of input and output actions that can be observed at an interface of a system.
- A writing action and a producing output action have an observable effect.

Deterministic Behaviour: A system behaves deterministically if, given an initial state at a defined instant and a set of future timed inputs, the future states, the values and instants of all future outputs are entailed.

- A system may exhibit an **intended** or an **erroneous** behaviour.

Service: The intended behaviour of a system.

Communication

Targets of Communication

- A communication system transports a message from a sender to one or more receivers within a given duration and with a high dependability.
- By high dependability we mean that by the end of a specified time window
  - the message should have arrived at the receivers with a high probability,
  - the message is not corrupted, either by unintentional or intentional means,
  - the security of the message has not been compromised, and that
  - there might be other constraints (e.g., minimal energy consumption).

Communication Protocol: The set of rules that govern a communication action.

Messages

Message: A data structure that is formed for the purpose of the timely exchange of information among computer systems

- Note: a message combines the value domain and of the temporal domain

- **Datagram** : A best effort message for the transmission of sporadic messages.
  - **PAR-Message** : A PAR-Message (Positive Acknowledgment or Retransmission) is an error controlled transport service for the transmission of sporadic messages from a sender to a single receiver.
  - **TT-Message**: A TT-Message (Time-Triggered) is an error controlled transport service for the transmission of periodic messages from a sender to many receivers where the send instant is derived from the progression of the global time. (**TDMA**)

Comparison of Messages

Characteristic	Datagram	PAR-message	TT-message
Send Instants	Sporadic	Sporadic	Periodic
Data/Control Flow	Uni-directional	Bi-directional	Uni-directional
Flow Control	None	Explicit	Implicit
Message Handling	R/W Or C/P	C/P	R/W
Transport Duration	A-priori Unknown	Upper-limit Known	Tight-limit Known
Message Jitter	Unknown	Large	Small
Temporal Error Detection	None	At Sender	At Receiver
Example	Udp	Tcp/Ip	Tt-ethernet TDMA

Stigmergy (I)

- Constituent systems (CSs) that form the autonomous subsystems of SoSs can exchange information items via two different types of channels:
  - the conventional communication channels for the transport of messages and
  - the stigmergic channels that transport information via the change and observation of states in the environment.

Stigmergy: it is a mechanism of indirect coordination between agents or actions. The principle is that the trace left in the environment by an action stimulates the performance of a next action, by the same or a different agent.

Stigmergy (II)

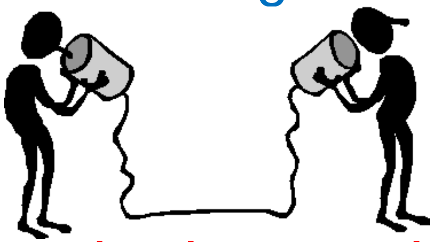
- The concept of stigmergy has been first introduced in the field of biology to capture the indirect information flow among ants working together.
  - Whenever an ant builds or follows a trail, it deposits a greater or lesser amount of pheromone on the trail, depending on whether it has found a prey or not.
  - If a prey is found, successful trails end up with a high concentration of pheromone.
  - The speed of the ants on a trail is a function of the pheromone concentration.
  - Since the trail-pheromone evaporates (we call this process environmental dynamics) unused trails disappear autonomously as time progresses.

Environmental Dynamics: Autonomous environmental processes that cause a change of state variables in the physical environment

Interfaces

Interactions over Channels

- Central to the integration of systems are their interfaces
  - points of interaction with each other and the environment over time.
  - a channel represents this exchange of information at connected interfaces.



Interaction: An interaction is an exchange of information at connected interfaces.

Channel: A logical or physical link that transport s information among systems at their connected interfaces.

- A channel is implemented by a communication system
  - e.g., a computer network, or a physical transmission medium
  - affecting the transported information, e.g., by introducing uncertainties
  - a channel model describes all channel effects relevant to the transfer of information.

RUIs

- We call an **interface** of a CS where the services are offered to other CSs a RUI.
  - the SoS as a whole relies on the services provided by the CSs across the RUIs.

Relied upon Interface (RUI): An interface of a CS where the services of the CS are offered to other CSs.

- **Relied upon Message Interface (RUMI):** A message interface where the services of a CS are offered to the other CSs of an SoS.
- **Relied upon Physical Interface (RUPI):** A physical interface where things or energy are exchanged among the CSs of an SoS.
- **Relied upon Service (RUS):** (Part of) a Constituent System (CS) service that is offered at the Relied Upon Interface (RUI) of a service providing CS under a Service Level Agreement (SLA).

Other Interfaces

Time-Synchronization Interface (TSI): The TSI enables external time-synchronization to establish a global time-base for time-aware CPSoSs.

Utility Interface: An interface of a CS that is used for the configuration, the control, or the observation of the behaviour of the CS.

- The purposes of the utility interfaces are to
  - configure, diagnose and update the system,
  - let the system interact with its physical environment.
- As example, we introduce
  - Diagnosis Interface (D-Interface): An interface that exposes the internals of a Constituent System (CS) for the purpose of diagnosis.
  - Monitoring CS: A CS of an SoS that monitors the information exchange s across the RUMI s of an SoS or the operation of selected CSs across the D-Interface

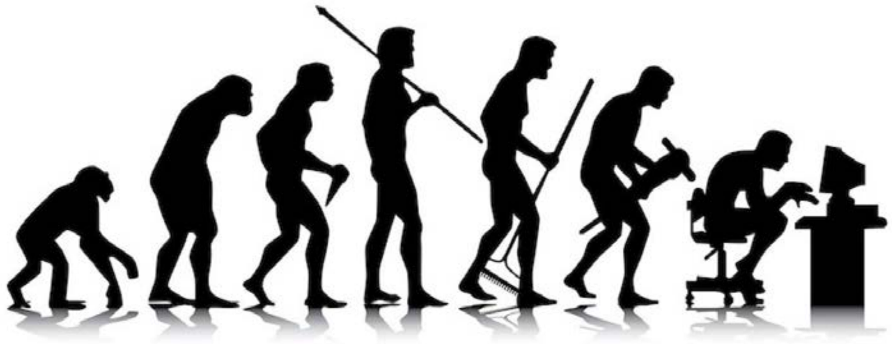
Evolution and Dynamicity

Dynamicity and Evolution

- Large scale Systems-of-Systems are designed for a long period of usage
  - Over time, the demands and the constraints put on the system will usually change, as will the environment in which the system is to operate.
  - Short-Term changes are referred as Dynamicity
  - Long-Term (planned) changes are referred as Evolution

Dynamicity: The capability of a system to react promptly to changes in the environment

Evolution: Process of gradual and progressive change or development, resulting from changes in its environment (primary) or in itself (secondary).



SoS Evolution

Although the term evolution in other contexts does not have a positive or negative direction, in SoSs evolution refers to maintaining and optimizing the system.

- More in detail, evolution is needed to cope with changes .
  - Managed evolution refers to the evolution guidance.
  - The goal can be anything like performance , efficiency , etc.

- **Manage d SoS evolution**: Process of modifying the SoS to keep it relevant in face of an ever-changing environment.
- **Unmanaged SoS evolution**: Ongoing modification of the SoS that occurs as a result of ongoing changes in (some of) its CSs.

Reconfigurability: The capability of a system to adapt its internal structure in order to mitigate internal failures or to improve the service quality.

SoS Authority

- Sometimes, governance-related facts may have impact on SoS evolution. Governance is generally related to

Authority: The relationship in which one party has the right to demand changes in the behaviour or configuration of another party, which has to conform to them.

(Collaborative) SoS Authority: An organizational entity that has societal, legal, and/or business responsibilities to keep a collaborative SoS relevant to its stakeholders. To this end it has authority over RUI specifications and how changes to them are rolled out.

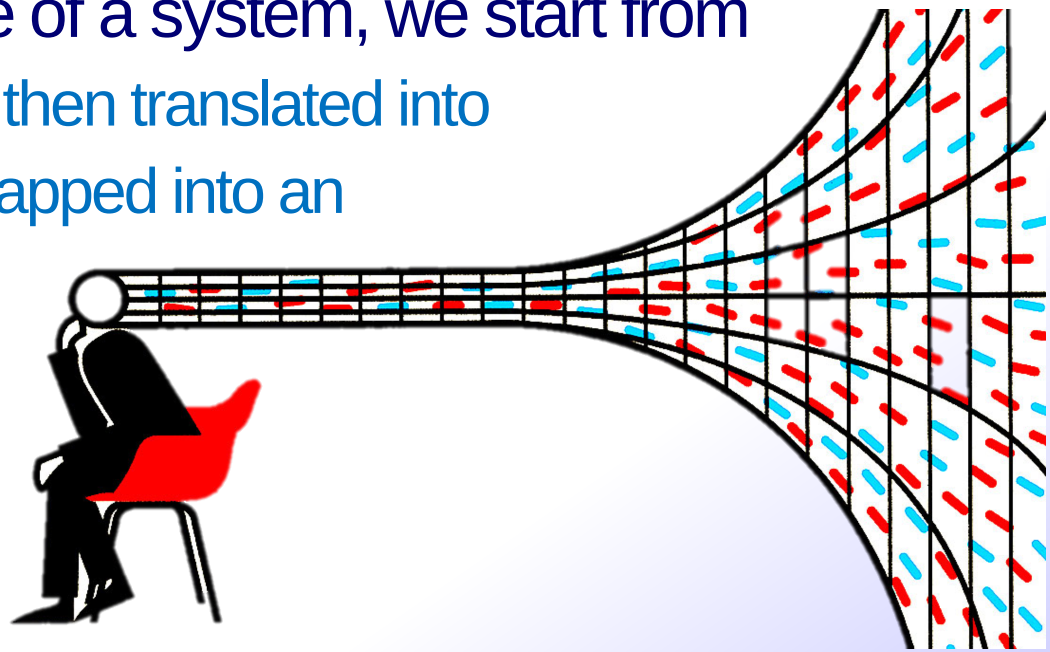
System Design and Tools

SoS Architecture

- **Problem**: some SoS requirements may not be fulfilled.
- **Solution**: The architecture of a system can have some variants or even can vary during its operation.
  - **Evolvable architecture**: it is adaptable and then is able to incorporate known and unknown changes in the environment or in itself.
  - **Flexible architecture**: it can be adapted to a variety of future possible developments.
  - **Robust architecture**: it performs well under a variety of possible future developments.
- The architecture then involves several components which interact with each other through interfaces.

Design Process

- During the development lifecycle of a system, we start from
  - **conceptual thoughts** which are then translated into
  - **requirements**, which are then mapped into an
  - **architecture**.



Design: The process of defining an architecture, components, modules and interfaces of a system to satisfy specified requirement.

Modularity: Engineering technique that builds larger systems by integrating modules.

Design Evolving Systems

Design for evolution: Exploration of forward compatible system architectures, i.e. designing applications that can evolve with an ever-changing environment.

- Design for evolution aims to achieve robust and/or flexible architectures.
- Principles of evolvability include modularity, updateability and extensibility.

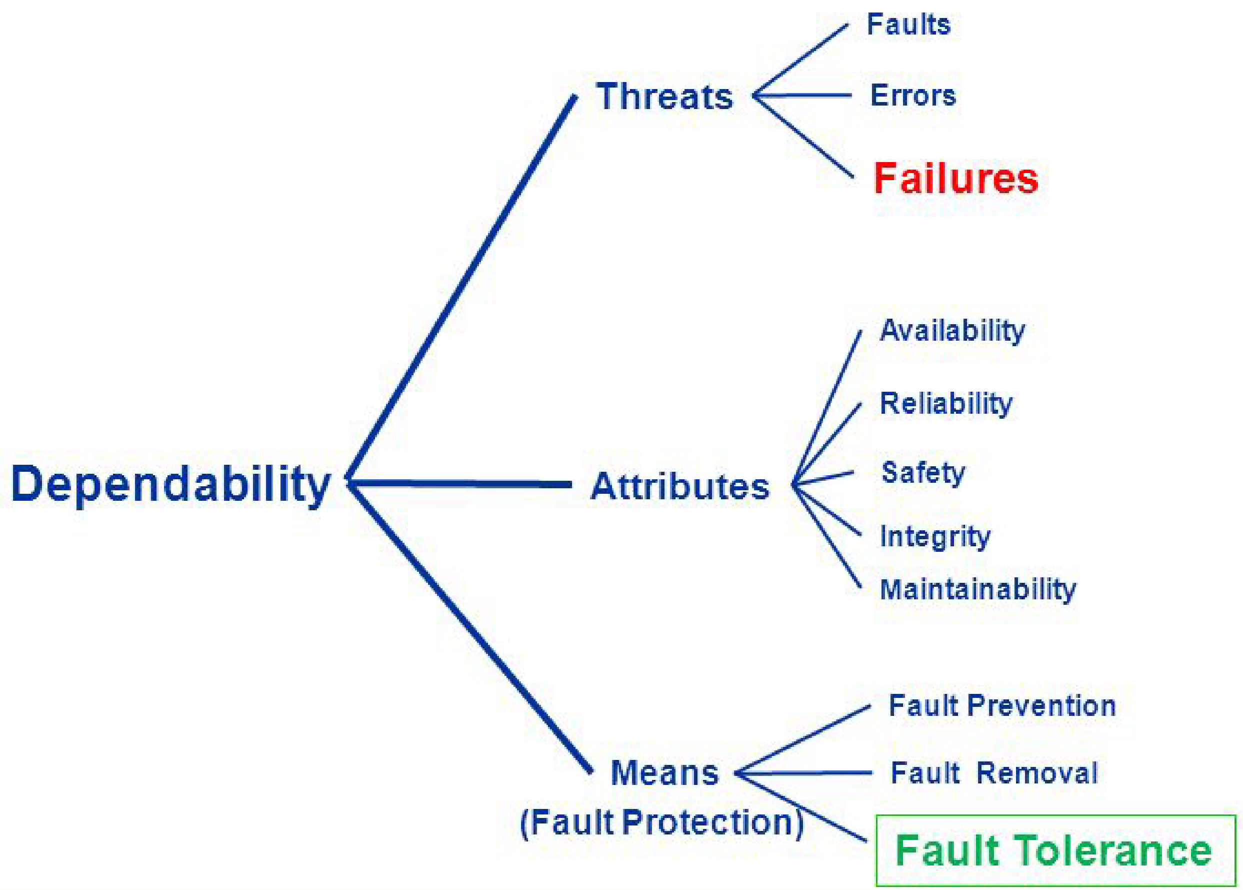
- In the context of SoS, design for evolution means that expected changes should be accommodated without any global impact on the architecture.
  - 'Expected' refers to the fact that changes will happen, it does not mean that these changes themselves are foreseeable.

Design for testability: The architectural and design decisions in order to enable easy and effective testing of the system.

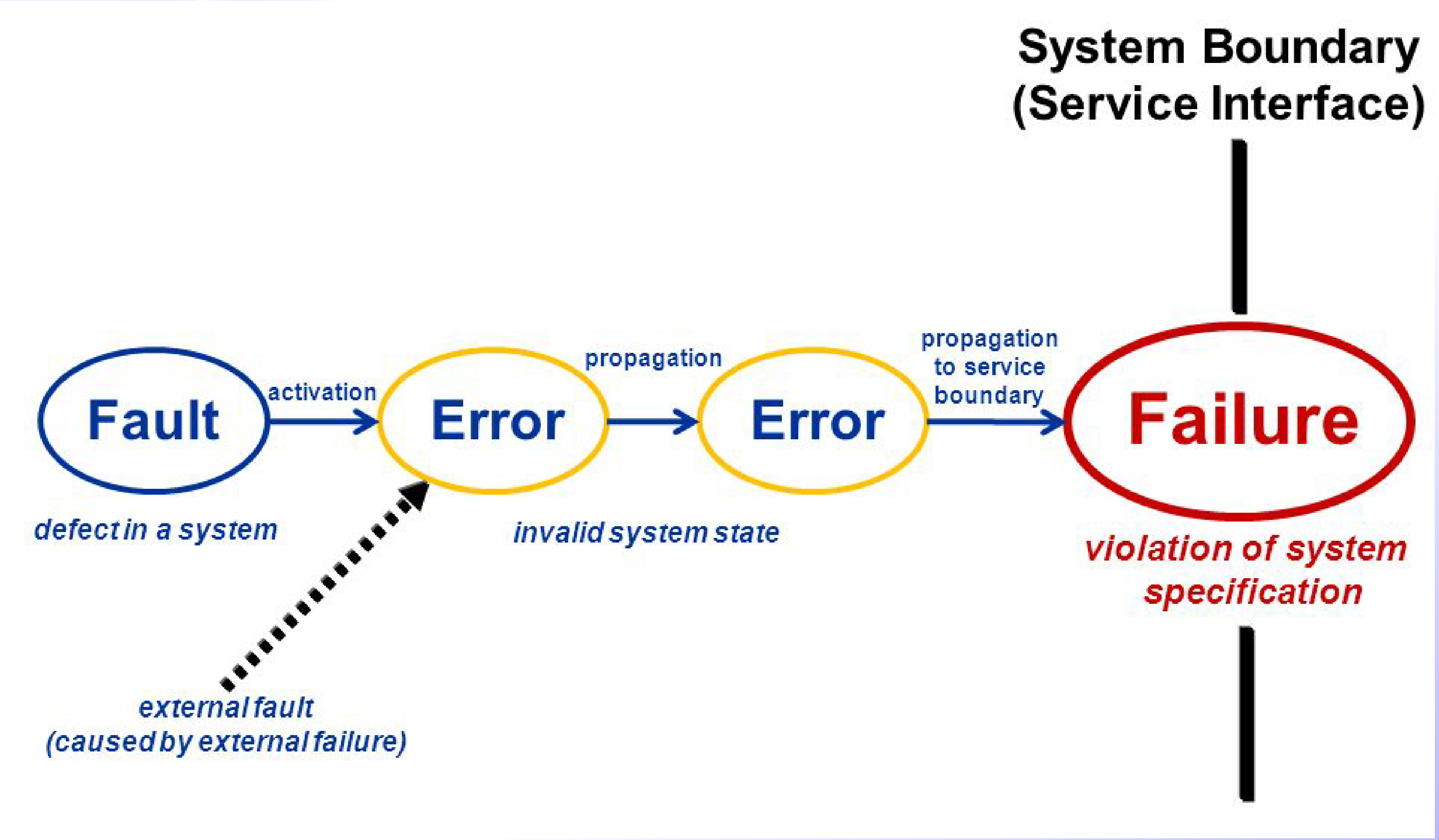
Dependability and Security

Dependability Overview

Dependability of a system *is the ability to deliver service that can justifiably be trusted"*

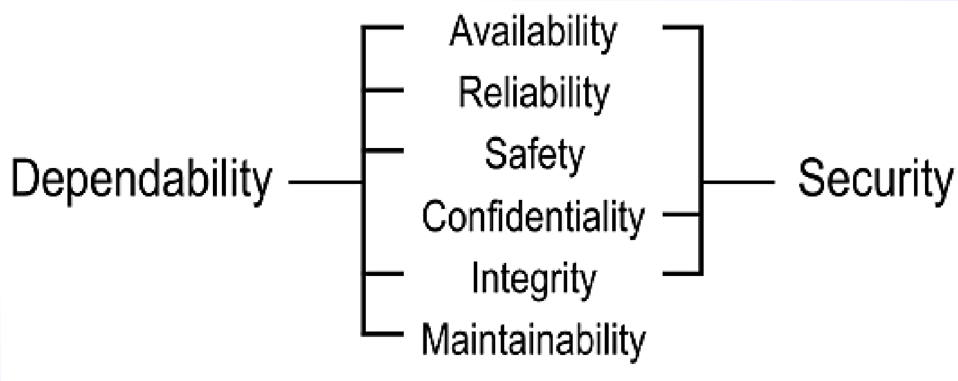


Dependability: Threats



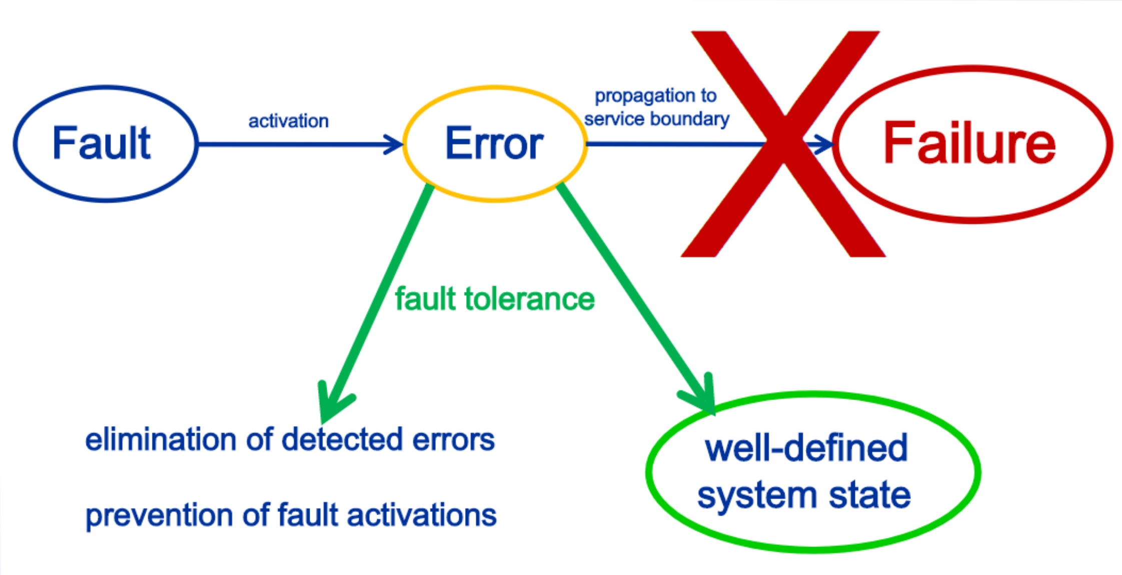
Dependability: Attributes

- **Availability**: Readiness for service.
- **Reliability**: Continuity of service.
- **Safety** : The absence of catastrophic consequences on the user(s) and on the environment.
- **Confidentiality**: The absence of unauthorized disclosure of information.
- **Integrity**: The absence of improper system state alterations.
- **Maintainability**: The ability to undergo modifications and repairs.
- **Robustness**: Dependability with respect to external faults (including malicious external actions).



Dependability: Means

- The means to attain dependability (and security) are grouped into four major dependability categories :
  - **Fault prevention**: The means to prevent the occurrence or introduction of faults.
  - **Fault tolerance**: The means to avoid service failures in the presence of faults.
  - **Fault removal**: The means to reduce the number and severity of faults.
  - **Fault forecasting**: The means to estimate the present number, the future incidence, and the likely consequences of faults.



Security (I)

Security: The composition of confidentiality, integrity, and availability;

- Security requires in effect the concurrent existence of availability for authorized actions only, confidentiality, and integrity (with “improper” meaning “unauthorized”)

- Security allows reducing the risk related to threats e.g., attacks, which may be conducted by external entities who exploit existing vulnerabilities.

Security (II)

- **Threat** : Any circumstance or event with the potential to adversely impact organizational operations / assets / individuals, via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

Vulnerability: Weakness in a system, in system security procedures, internal controls, or implementations that could be exploited by a threat.

- **Risk**: A measure of the extent to which an organization is threatened by a potential circumstance or event, and typically a function of
  - the adverse impacts that would arise if the circumstance or event occurs;
  - and the likelihood of occurrence

References

- Ceccarelli A., Bondavalli A., Froemel B., Hoeftberger O., Kopetz H. (2016) Basic Concepts on Systems of Systems. In: Bondavalli A., Bouchenak S., Kopetz H. (eds) Cyber-Physical Systems of Systems. Lecture Notes in Computer Science, vol 10099. Springer, Cham
- Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. 1(1), 11–33 (2004).
- Jamshidi, M.: Systems of Systems Engineering—Innovations for the 21st Century. Wiley, Cambridge (2009)
- Kopetz, H.: Real-Time Systems, 2nd edn. Springer, New York (2011)