Architecting Software-Intensive System-of-Systems

SEN Symposium

20 January 2017
Amsterdam, The Netherlands

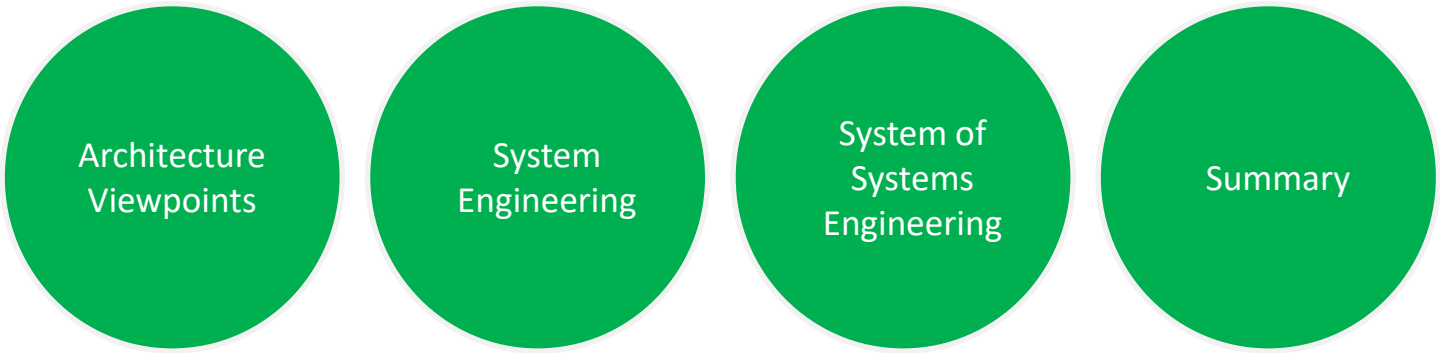
Prof.dr.ir. Bedir Tekinerdogan

Wageningen University, Chair Information Technology
Wageningen, The Netherlands

bedir.tekinerdogan@wur.nl
https://www.linkedin.com/in/bedir

WAGENINGEN UR
For quality of life

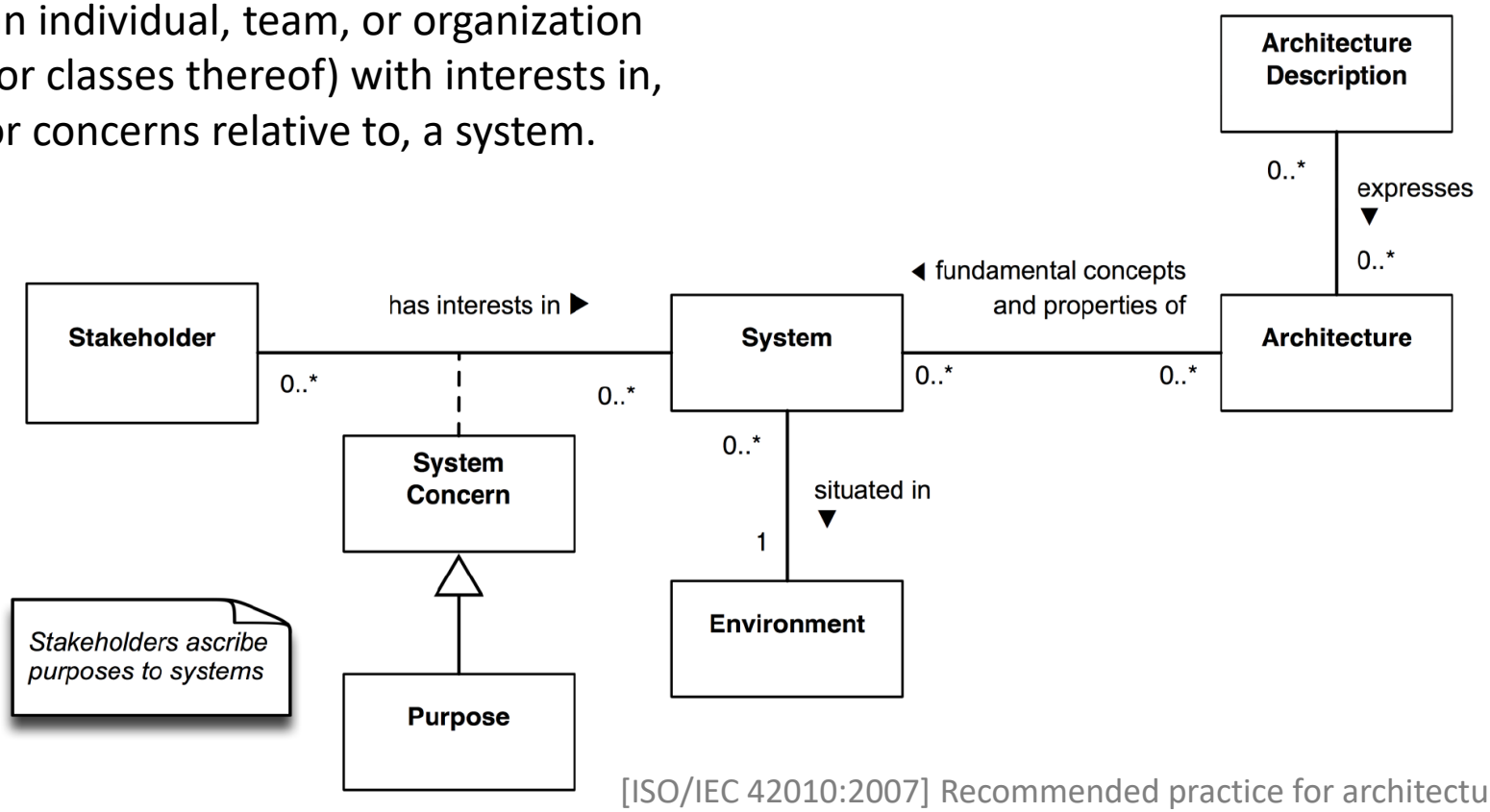
Contents


Architecture Viewpoints
System Engineering
System of Systems Engineering
Summary

Architecture Viewpoints

Architecture Stakeholders

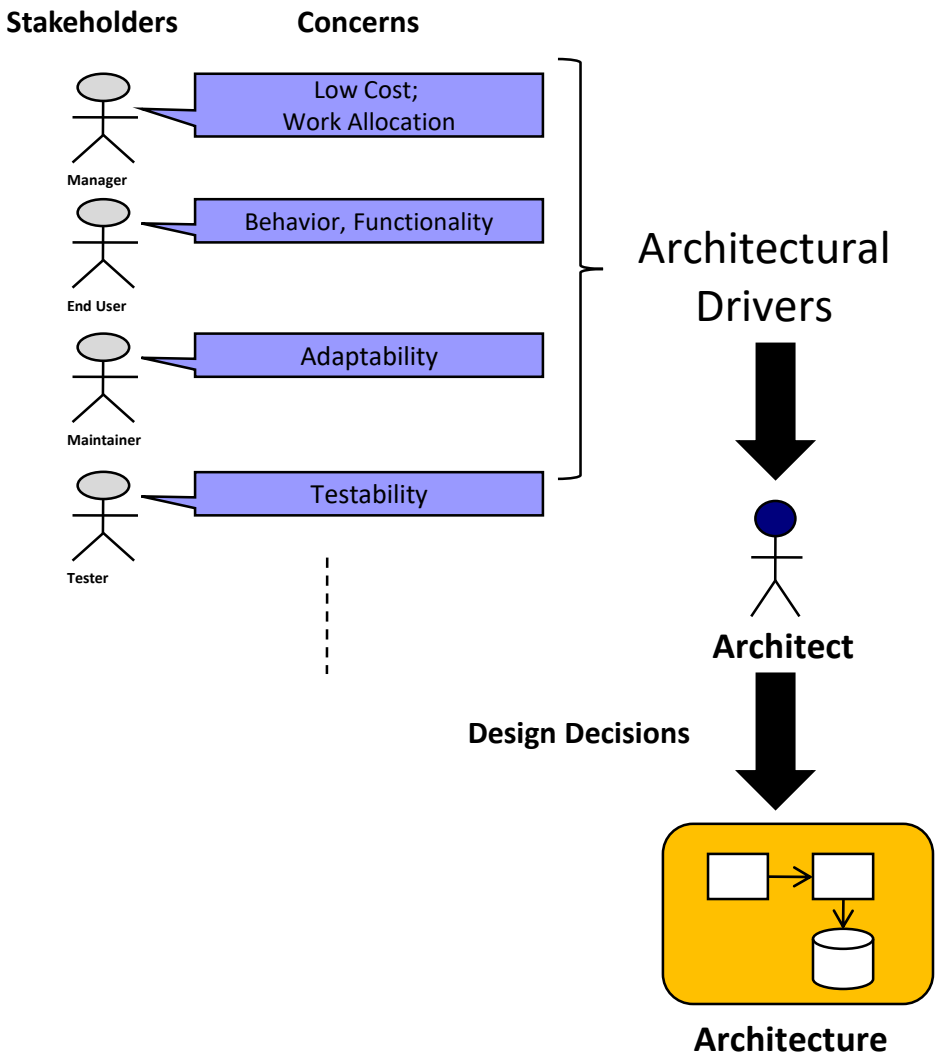
- **system stakeholder:**
  an individual, team, or organization
  (or classes thereof) with interests in,
  or concerns relative to, a system.



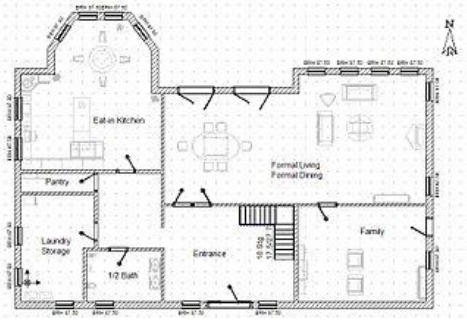
[ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010) July 2007.

Architectural Drivers

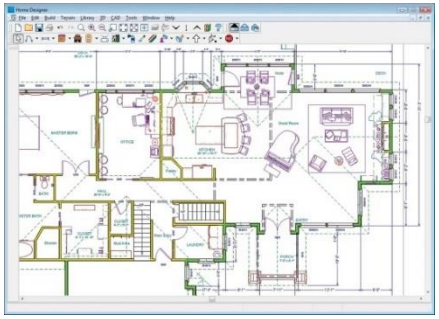
- **Stakeholder** is any person who has interest in the architecture
- Each stakeholder can have different **concerns**
- Each concern puts forces on the architect and influences the early design decisions that the architect makes.



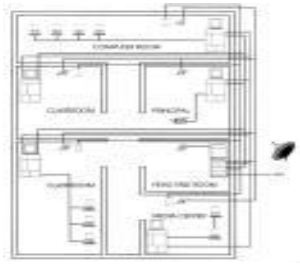
Multiple Views of the Architecture…



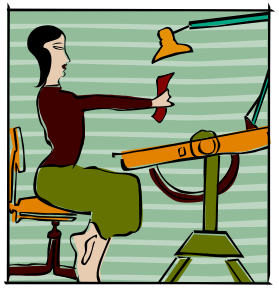
Floor plan



Interior Plan



Wiring Plan



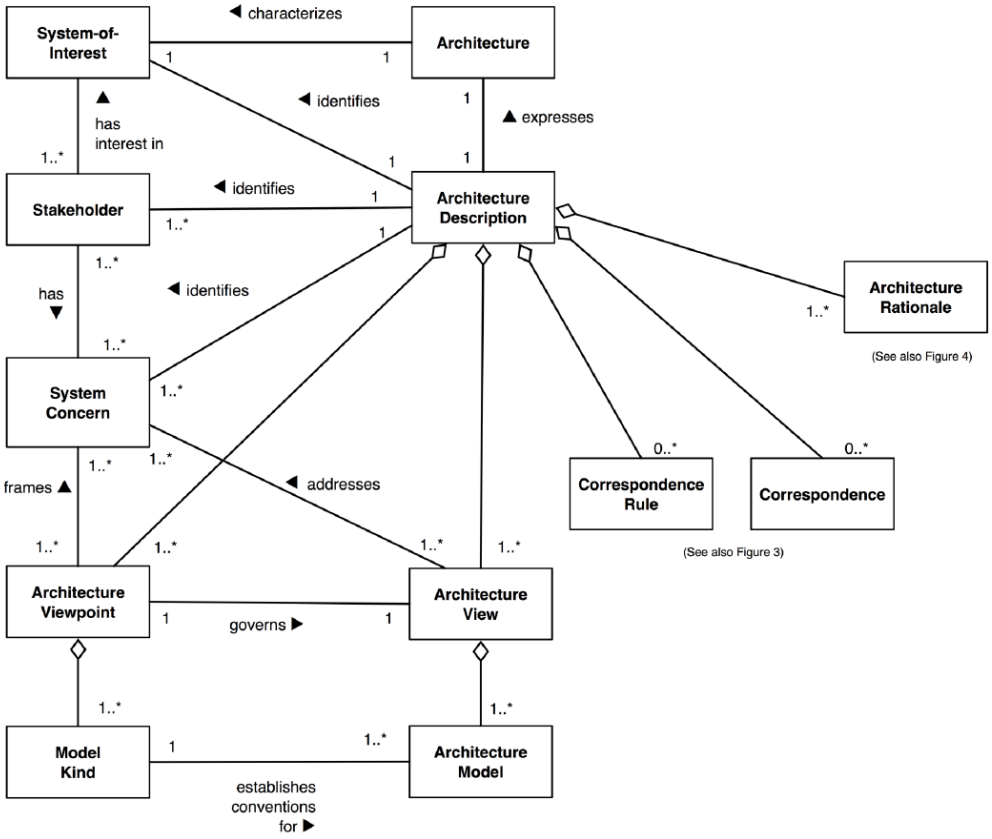
Architectural Viewpoints

View:

- a representation of a system from the perspective of one or more concerns which are held by one or more stakeholders.

Viewpoint:

- A pattern or template from which to construct individual views.



[ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010) July 2007.

Example – UML Deployment Viewpoint

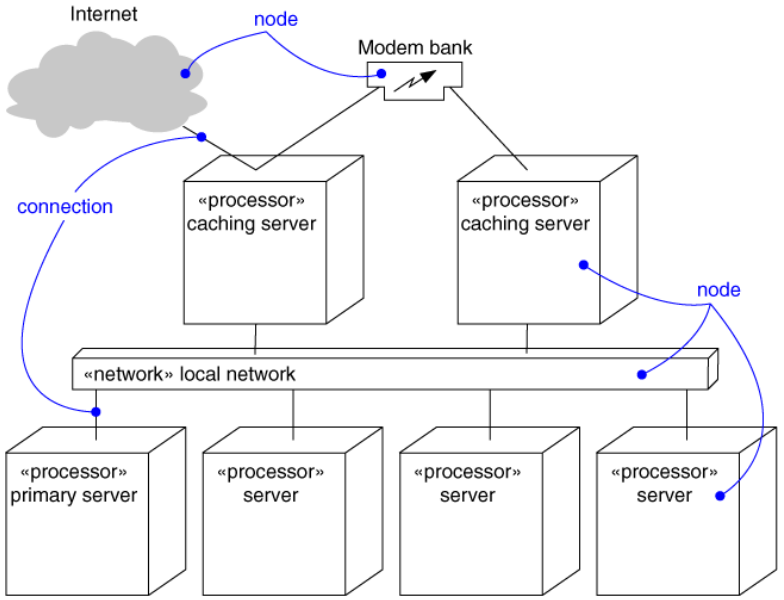
Viewpoint

- Name: Deployment Viewpoint
- Stakeholders:
  - System Designer
- Concerns:
  - System Design
- Components:
  - Processing Nodes
- Notation

Node

Connection

Deployment View - Example

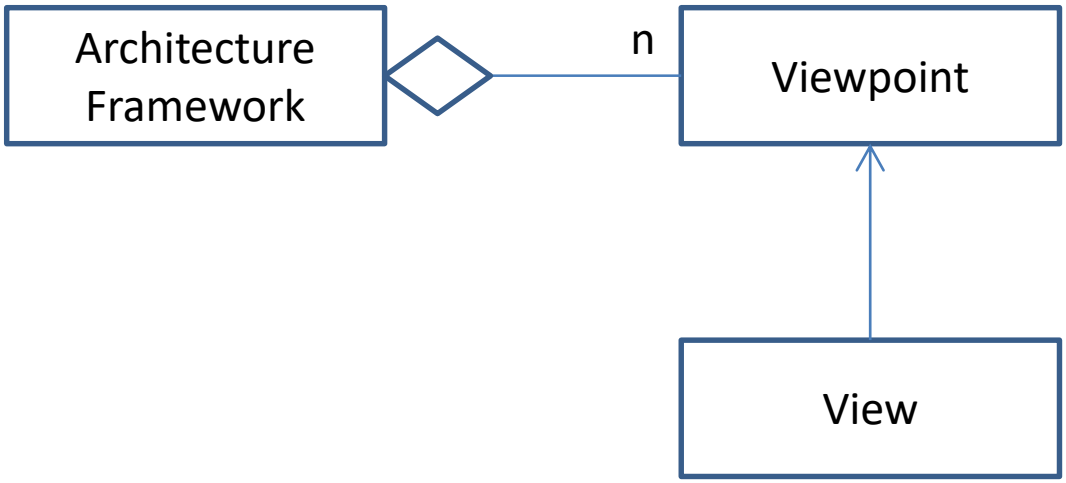


Rationale for Viewpoints

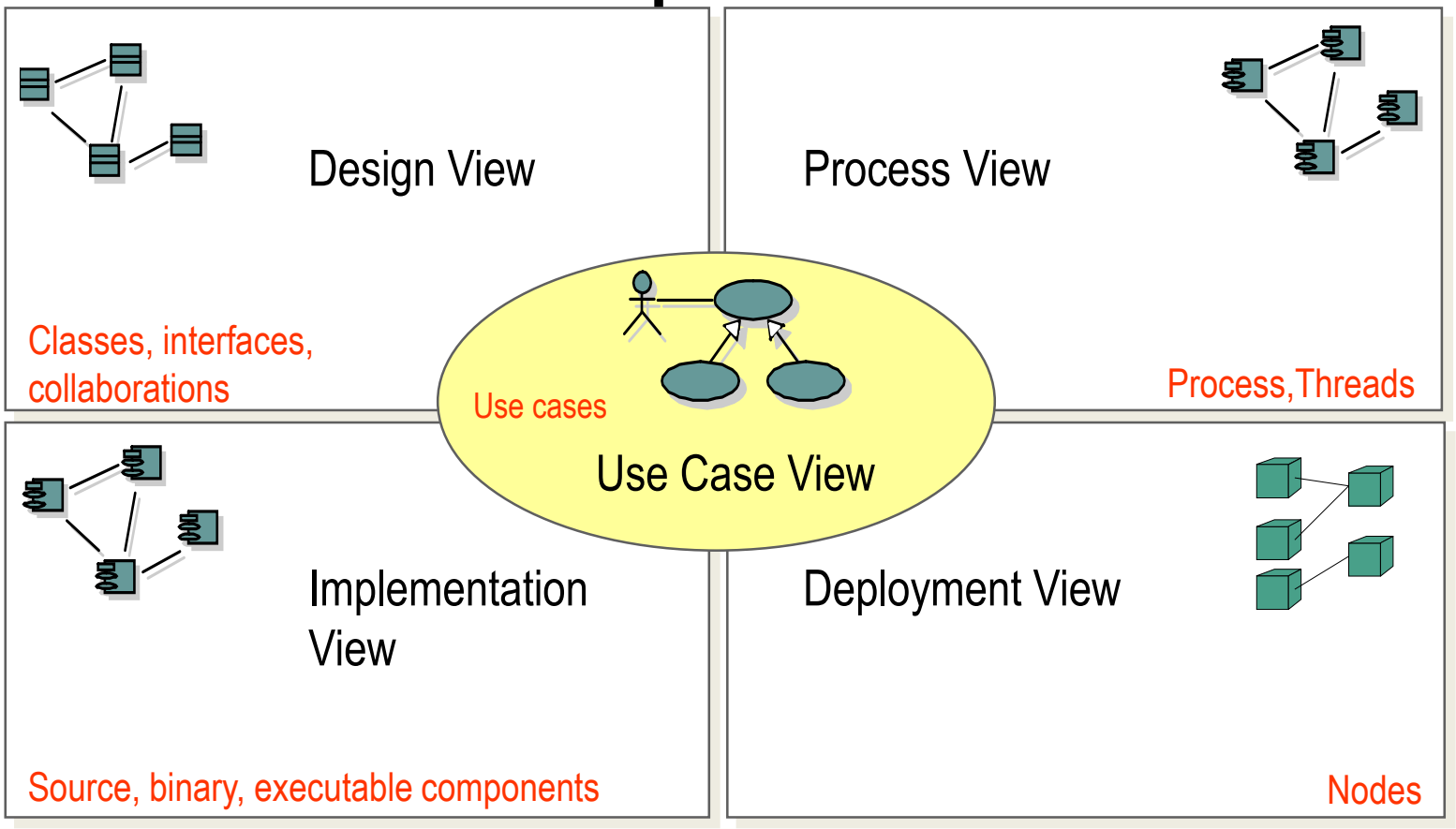
- **Communication**
  - With stakeholders about design decisions
- **Analysis**
  - to support design decisions and enhance design
- **Support Development**
  - To guide detailed design & implementation

Architecture Framework

- Coherent set of viewpoints



Example - UML

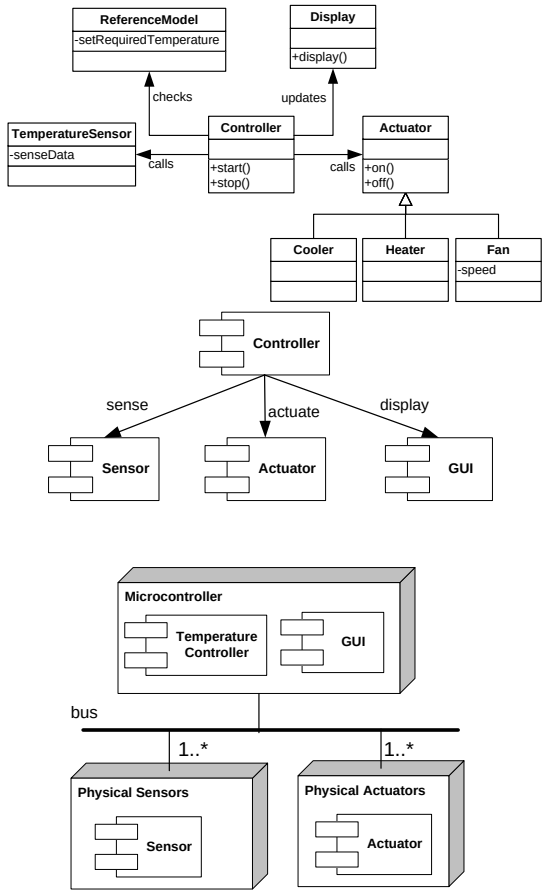


Organization
Package, subsystem

Dynamics
Interaction, State machine

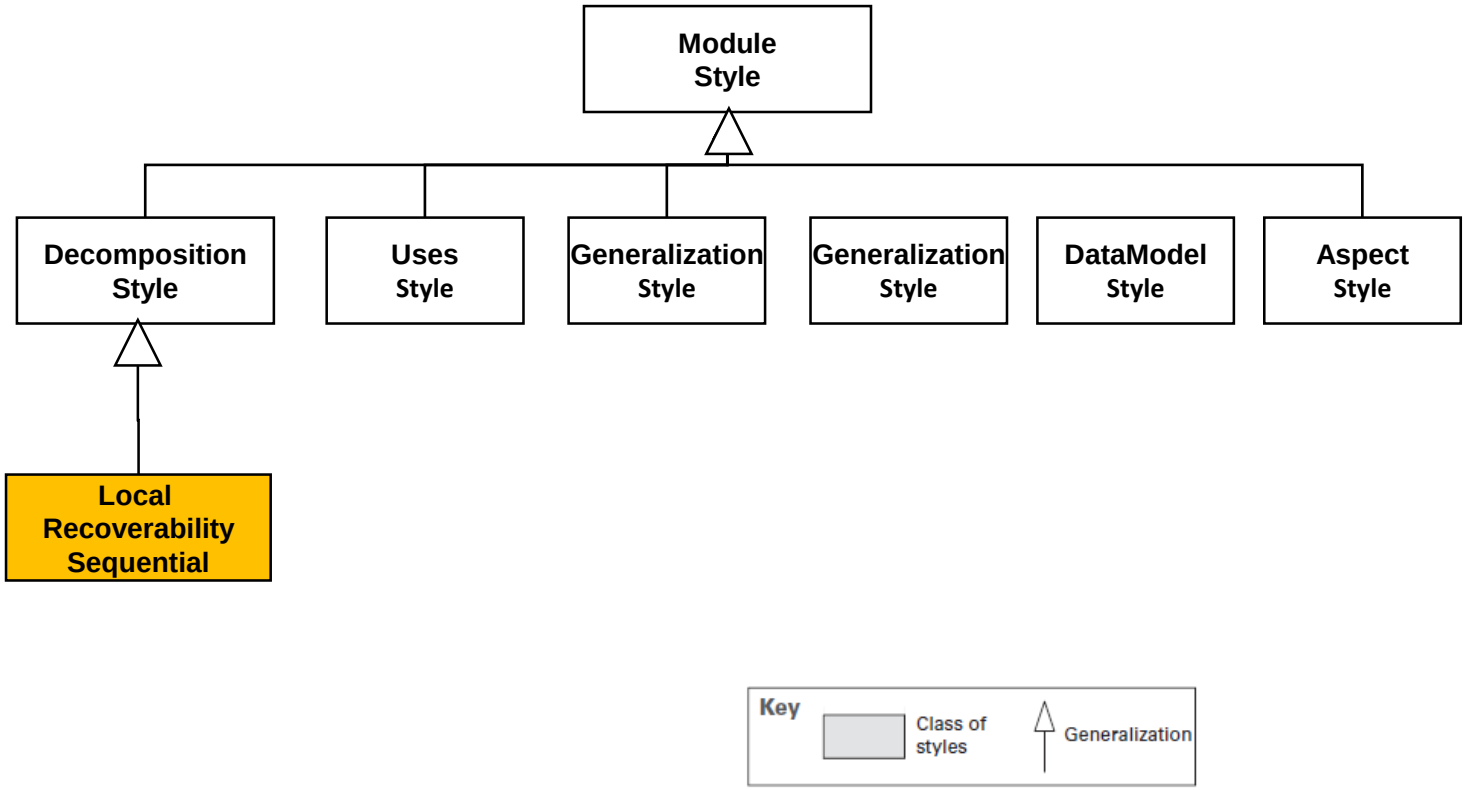
Example – Views and Beyond Approach

- **Module Styles**
  - How is the architecture structured as a set of implementation units?
- **Component-and-Connector Styles**
  - How is the architecture structured as a set of elements that have run-time behavior and interactions?
- **Allocation Styles**
  - How does the architecture relate to non-software structures in its environment?



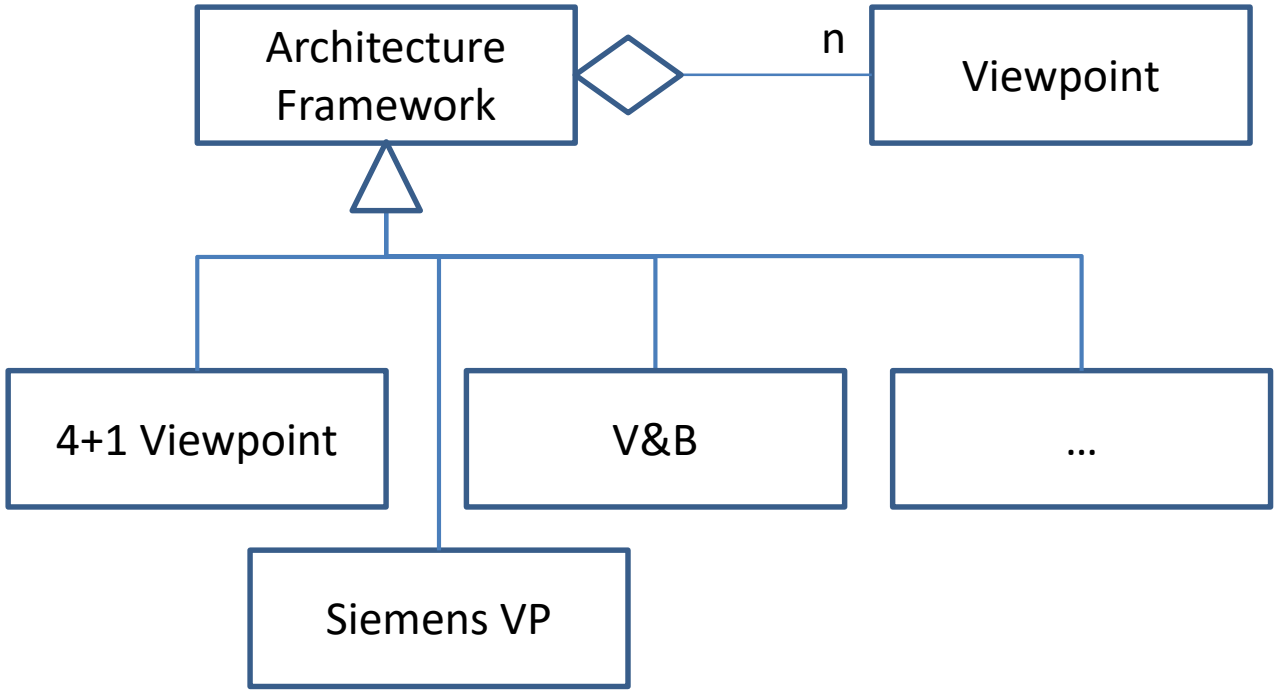
P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, J. Stafford. Documenting Software Architectures: Views and Beyond. Second Edition. Addison-Wesley, 2010

Extending Viewpoints

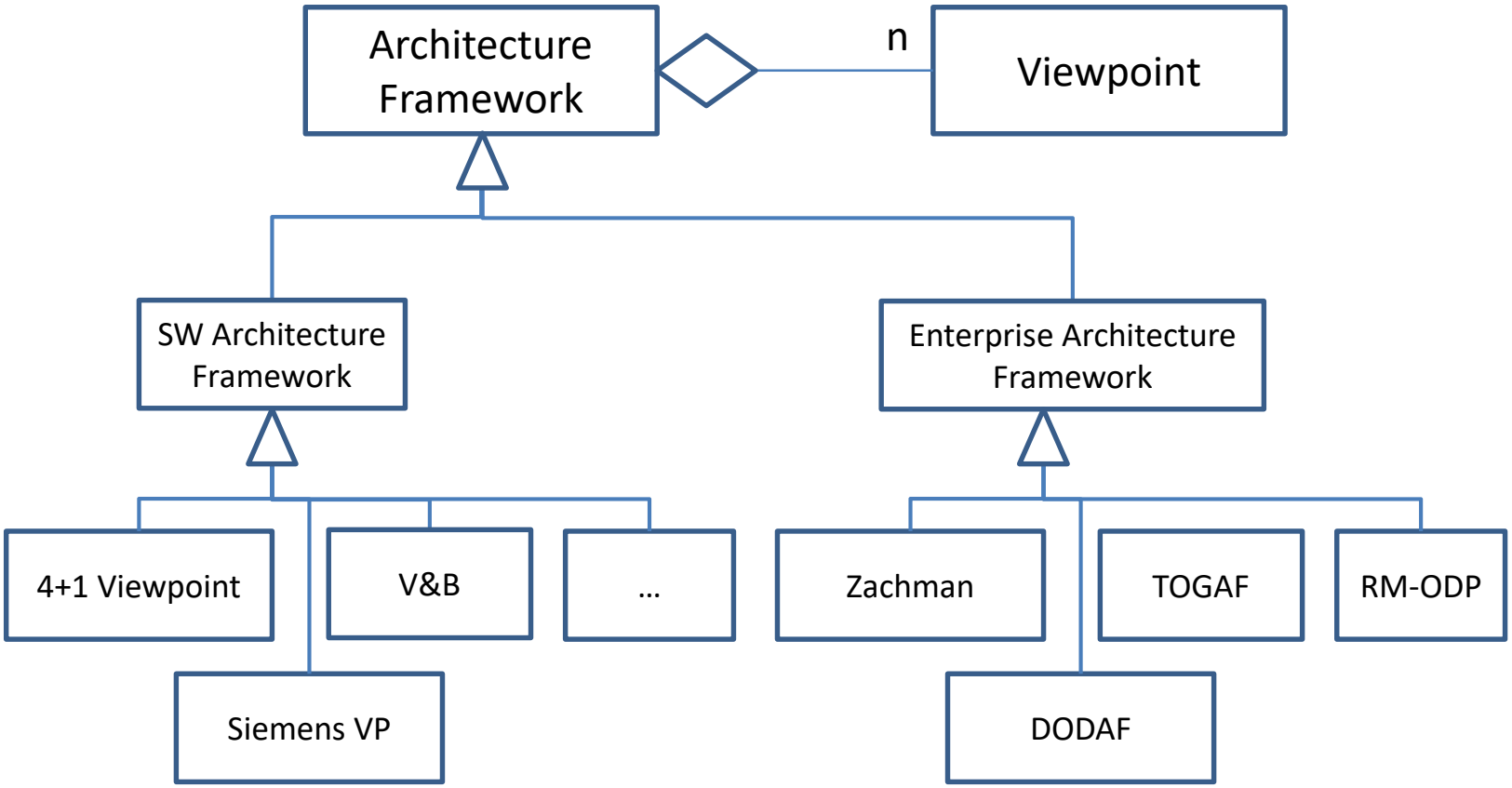


P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, J. Stafford. Documenting Software Architectures: Views and Beyond. Second Edition. Addison-Wesley, 2010

Architecture Framework



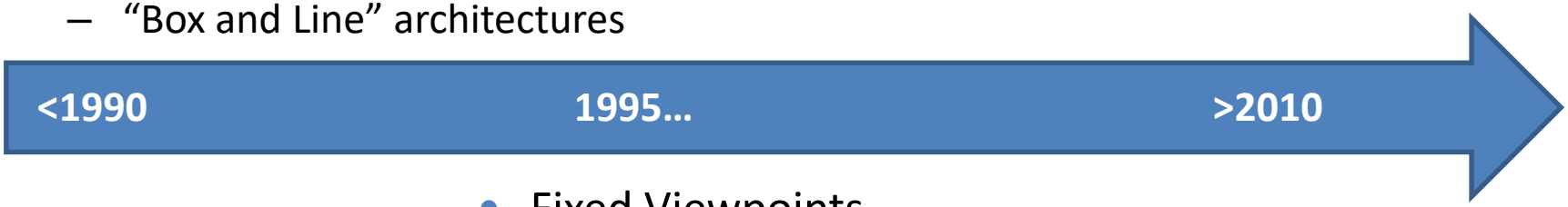
Architecture Framework



Evolution of Architectural Modeling -Viewpoints

- Open-ended viewpoints
  - Predefined Viewpoints
  - Plus mechanisms to add new viewpoints
  - E.g. Views and Beyond Approach

- No Viewpoint Approach
  - Informal Modeling
  - "Box and Line" architectures



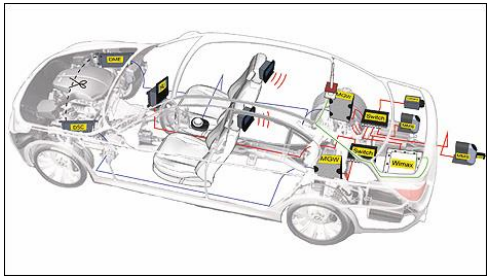
- Fixed Viewpoints
  - 4, 4+1, 6, …

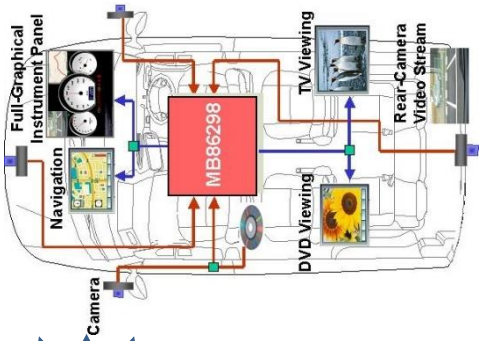
B. Tekinerdogan, Exploring Research Directions in Software Architecture Modeling, International Journal of Software Architecture, Vol. 1. No. 1, pp. 7-9, July 2010.

System Engineering

Beyond Software – System Engineering

- A system is a construct or collection of different elements that together produce results not obtainable by the elements alone.
- The elements, or parts, can include **people**, **hardware**, **software**, **facilities**, **policies**, and documents; that is, all things required to produce systems-level results.
- The results include **system level qualities**, properties, characteristics, functions, behavior and performance.
- The **value added by the system as a whole**, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected



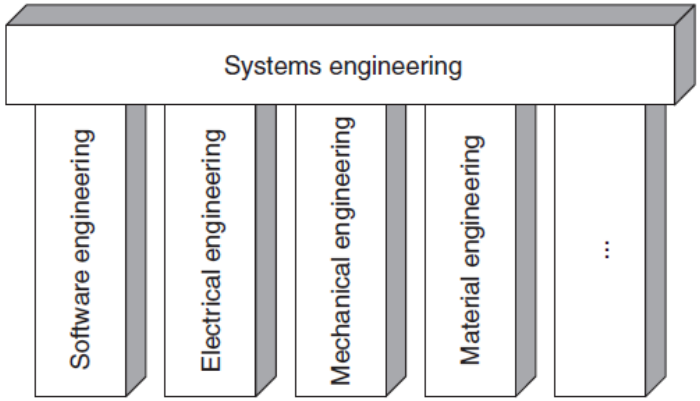


Not only Software!

International Council on System Engineering
http://www.incose.org/

The System Engineer

- A system engineer is the connecting link between the disciplines in a project, which are sometimes very different.
- System engineers think along the line of the entire system, independent of software, hardware, or other specific views.

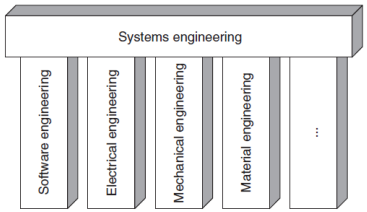


Questions…

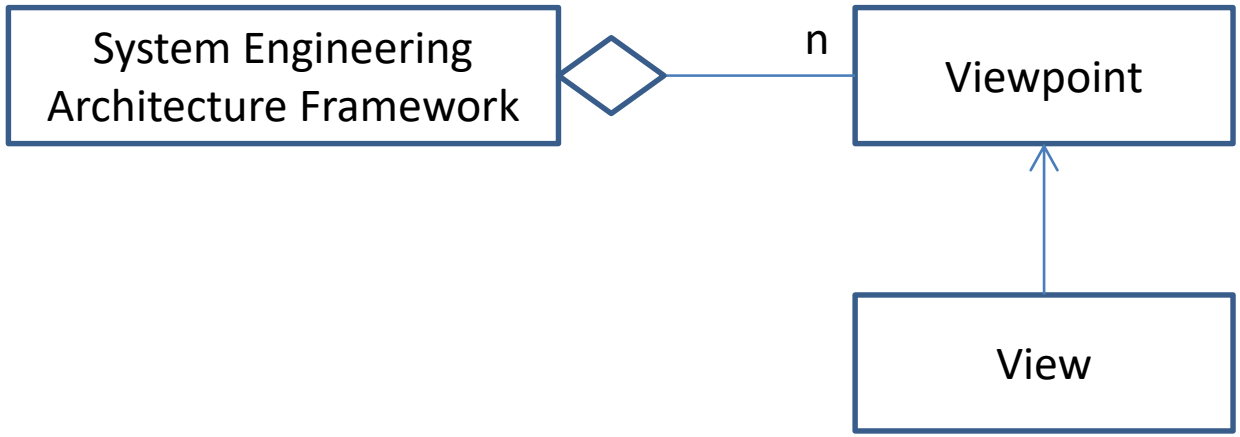
- Which domains are needed for system engineering?
- What are the stakeholders for each domain?
- What are the concerns?
- What are the required architecture viewpoints?
- Can we have a complete set for the whole system engineering?
- Or should we consider viewpoints per domain?
- If separate, how to integrate the different views?
- What should be the architecture design process using the viewpoints?
- ….

System Engineering AF

First Alternative



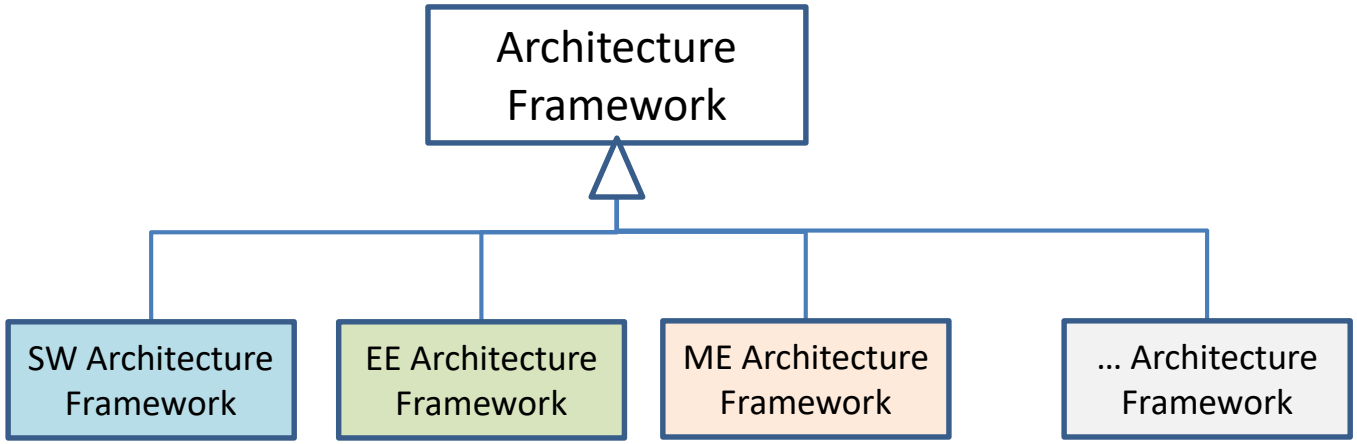
- One Architecture Framework
- What should be the viewpoints?
- What's the abstraction level? System components?
- Information loss due to higher abstraction level?



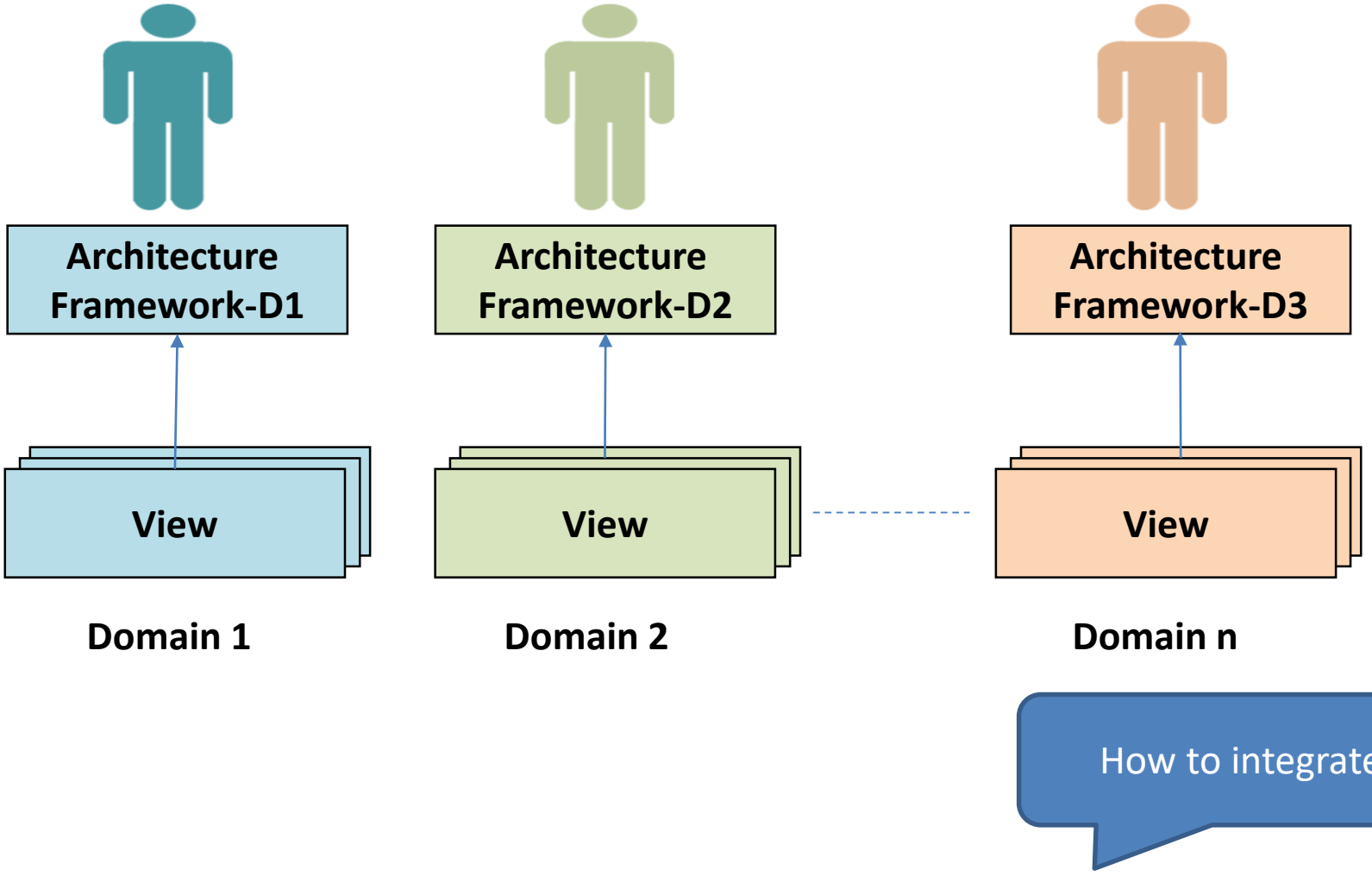
System Engineering AF

Second Alternative

- Architecture Framework for each domain/discipline…
- Separation of Concerns!

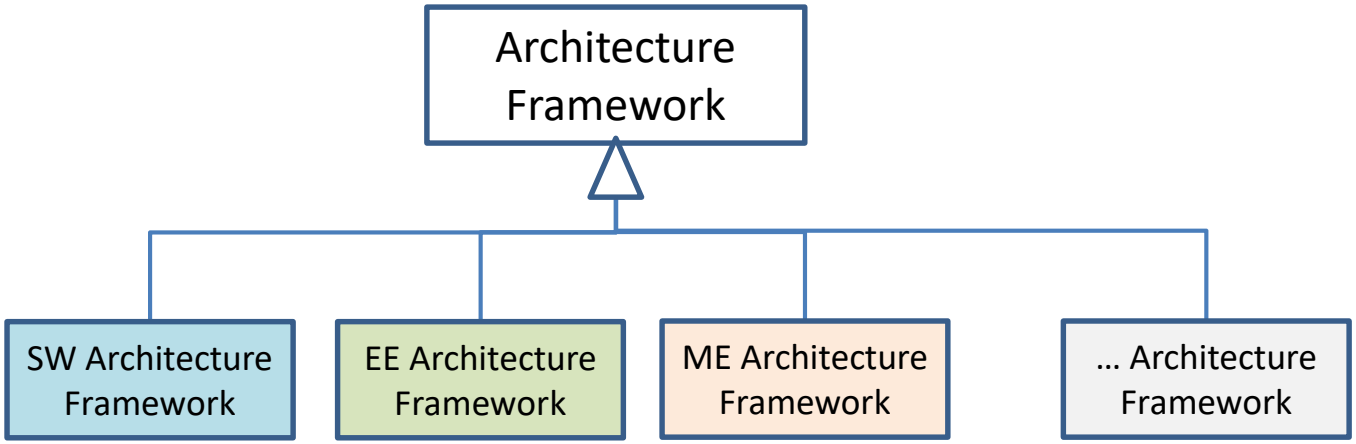


AF for each Domain



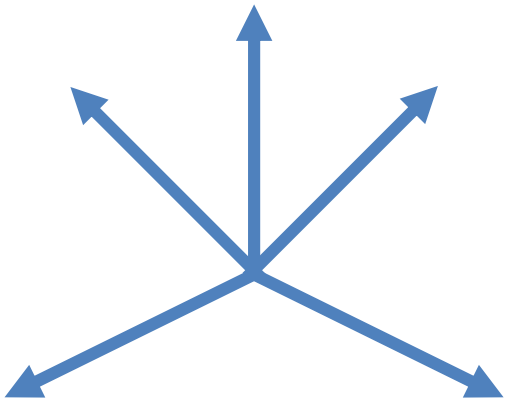
Dominant Decomposition?

- How to design the system?
- Which one is the dominating discipline/domain?
- The results will be different!



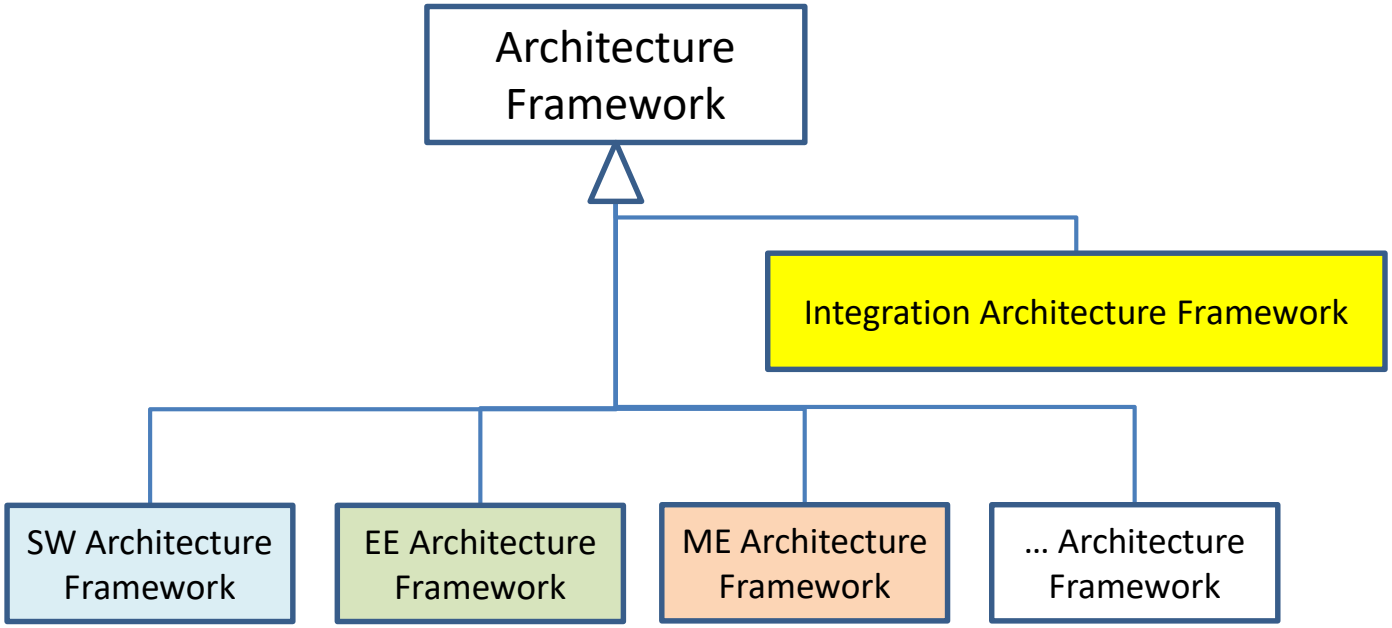
Multi-Dimensional Space

- Asymmetric vs. Symmetric Composition
- Identification, encapsulation of *any* kinds of concerns, *simultaneously*
  - Introduce new concerns or kinds of concerns at any time, as needed
- Flexible, concern-based composition
- Handling of overlapping, interacting and cross-cutting concerns
- *On-demand remodularization*

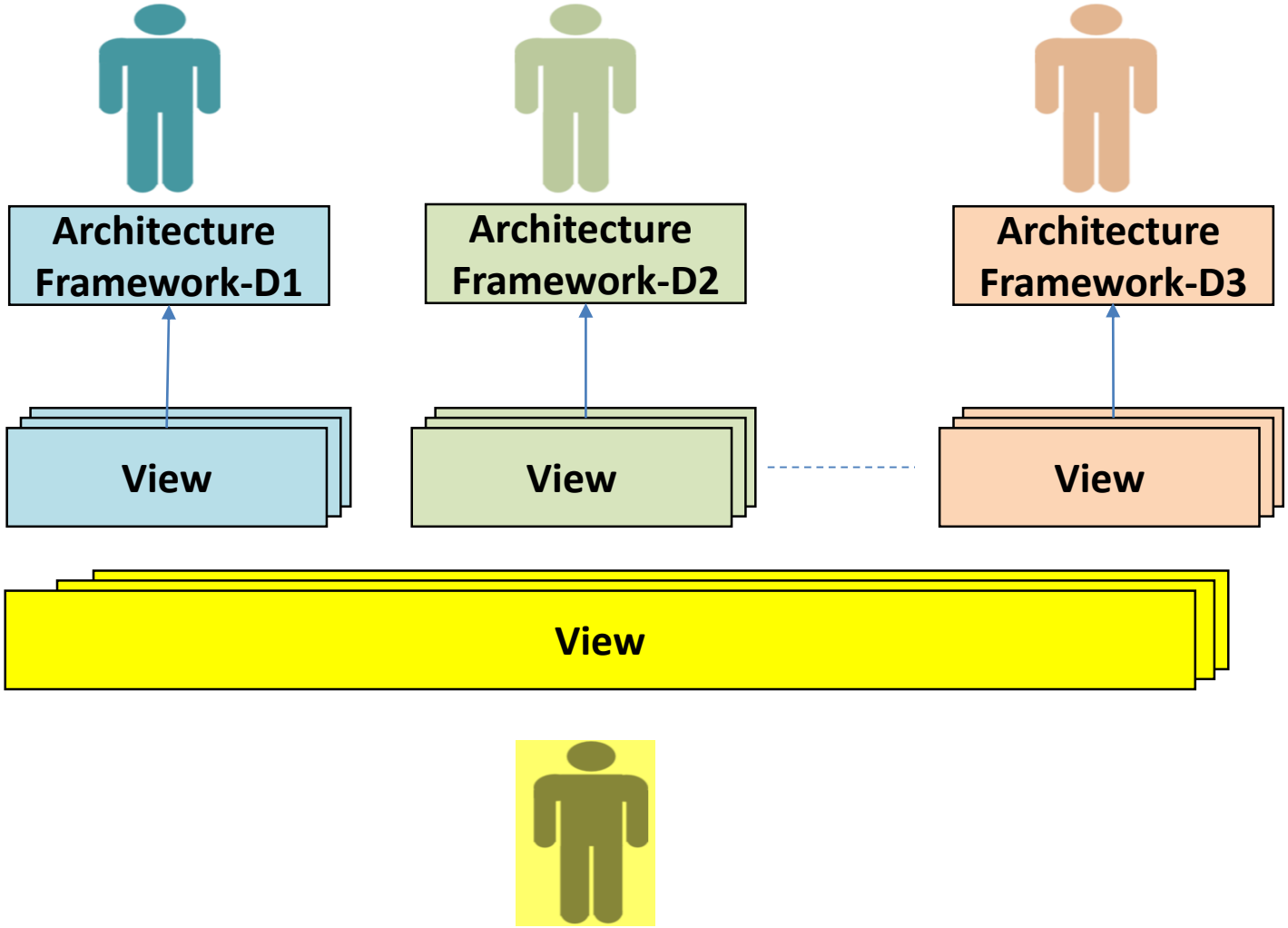


System Engineering AF

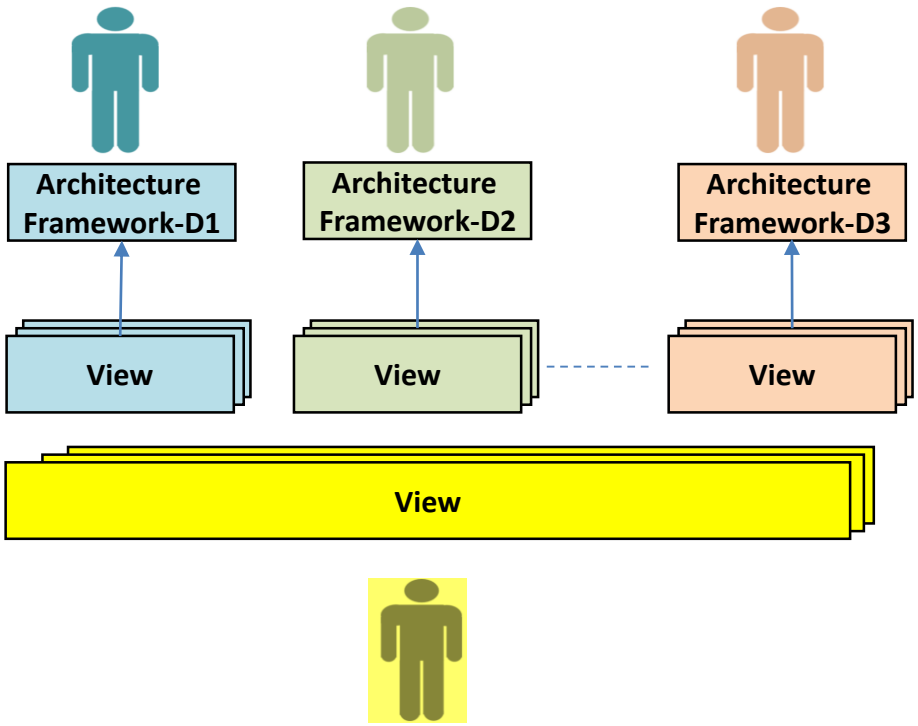
- Add one overarching Framework that is responsible for the trans-discipline/systemic perspective?



AF for each Domain



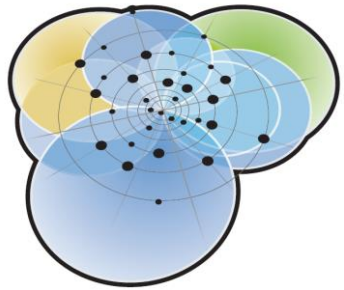
Many Viewpoints…



System-of-Systems Engineering

System vs. Systems of Systems

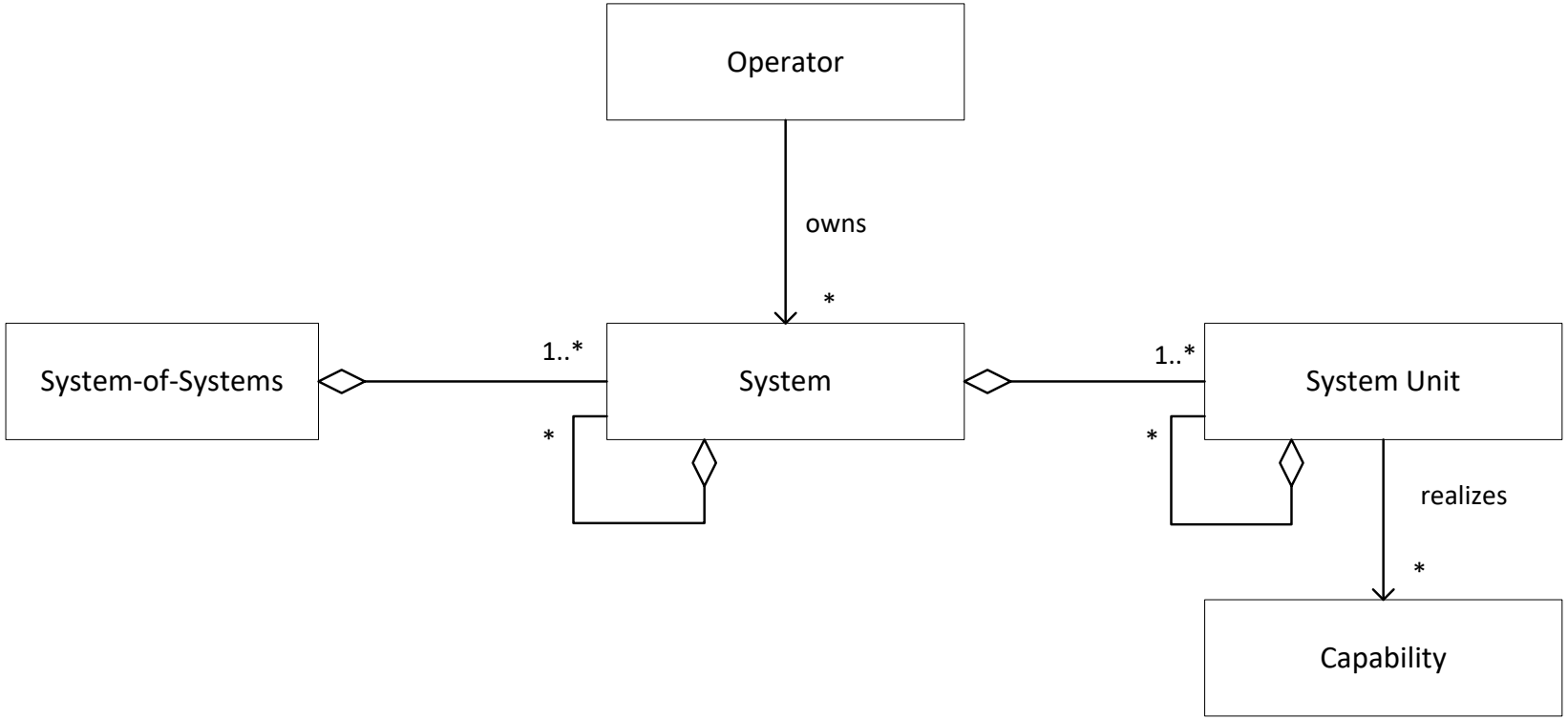
- **System**: A functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole.



- A **capability** is the ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks.

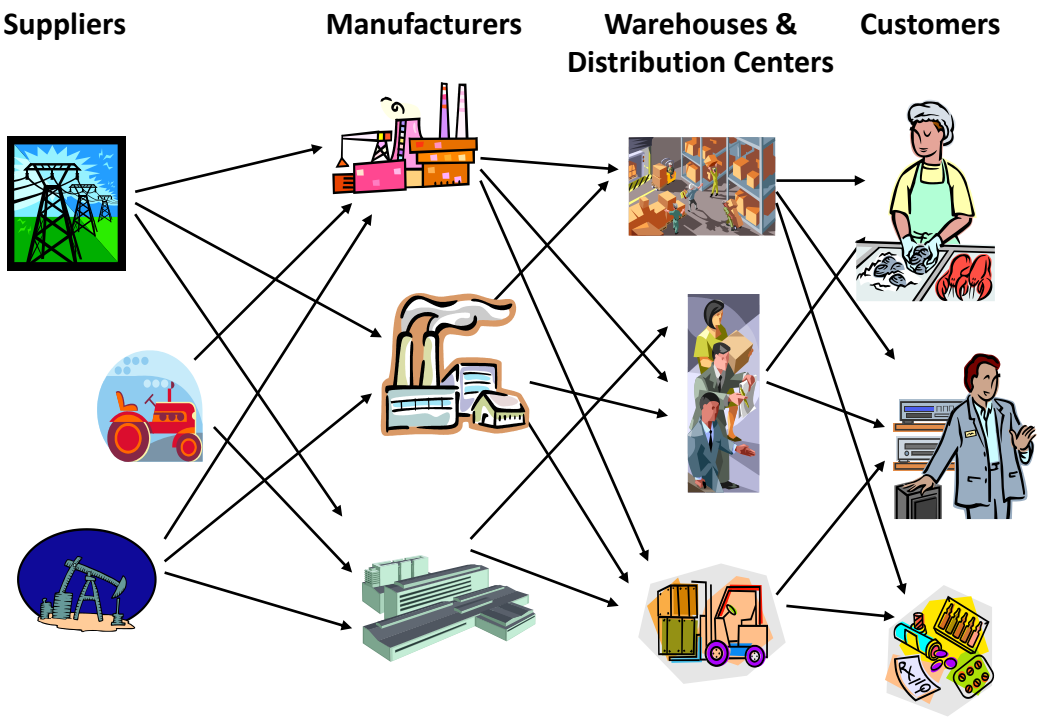
- **Systems of Systems (SoS)** is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities

System-of-Systems

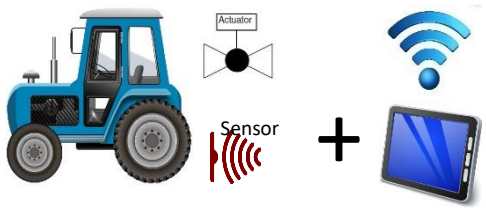


System of Systems - Fresh Logistics Management/Supply Chains

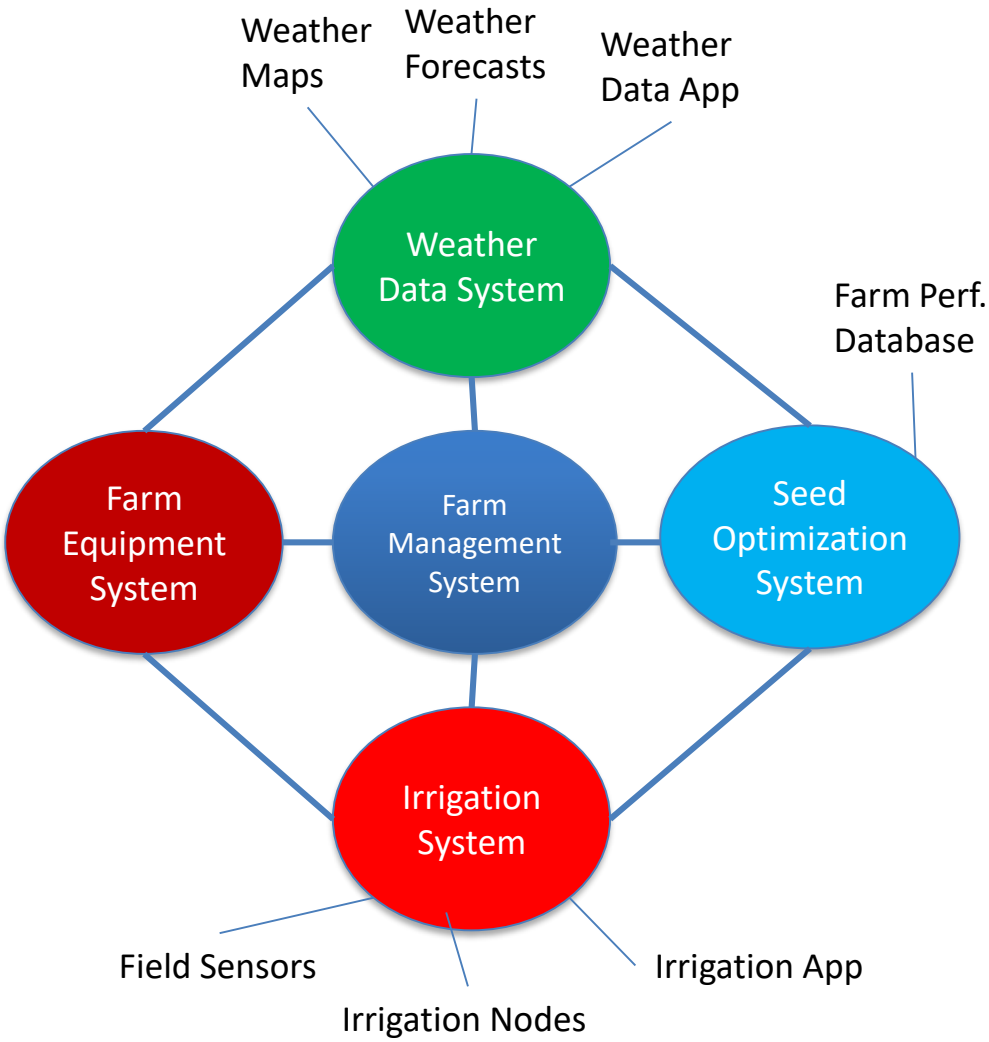
- Supply chains consist of autonomous organizations that independently make decisions
- Each organization can be considered as a complex system
- The operational performance of a supply chain relies on the proper integration of the activities of the organizations
- Hence, a supply chain system can be viewed as a system of systems
- System engineering approach can help the modeling and analysis of decentralized supply chain systems



Farming System-of-Systems



Smart Connected Product



System Engineering vs. SoS Engineering

	System Engineering	System of Systems Engineering
Stakeholder Involvement	Clearer set of stakeholders	Stakeholders at both system level and SoS levels, with competing interests and priorities; all stakeholders may not be recognized
Governance	Aligned PM and funding	Added levels of complexity due to management and funding for both the SoS and individual systems; SoS does not have authority over all the systems
Operational Focus	Designed and developed to meet operational objectives	Called upon to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives
Acquisition	Aligned to Milestones, documented requirements, SE with a Systems Engineering Plan	Added complexity due to multiple system lifecycles across acquisition programs, involving legacy systems, systems under development, new developments, and technology insertion; Typically have stated capability objectives upfront which may need to be translated into formal requirements
Test & Evaluation	Test and evaluation of the system is generally possible	Testing is more challenging due to the difficulty of synchronizing across multiple systems' life cycles; given the complexity of all the moving parts and potential for unintended consequences
Boundaries and Interfaces	Focuses on boundaries and interfaces for the single system	Focus on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control and functionality across the SoS while balancing needs of the systems
Performance and Behavior	Performance of the system to meet specified objectives	Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems

Systems Engineering Guide for Systems of Systems, DoD, 2008.

Types of SoS

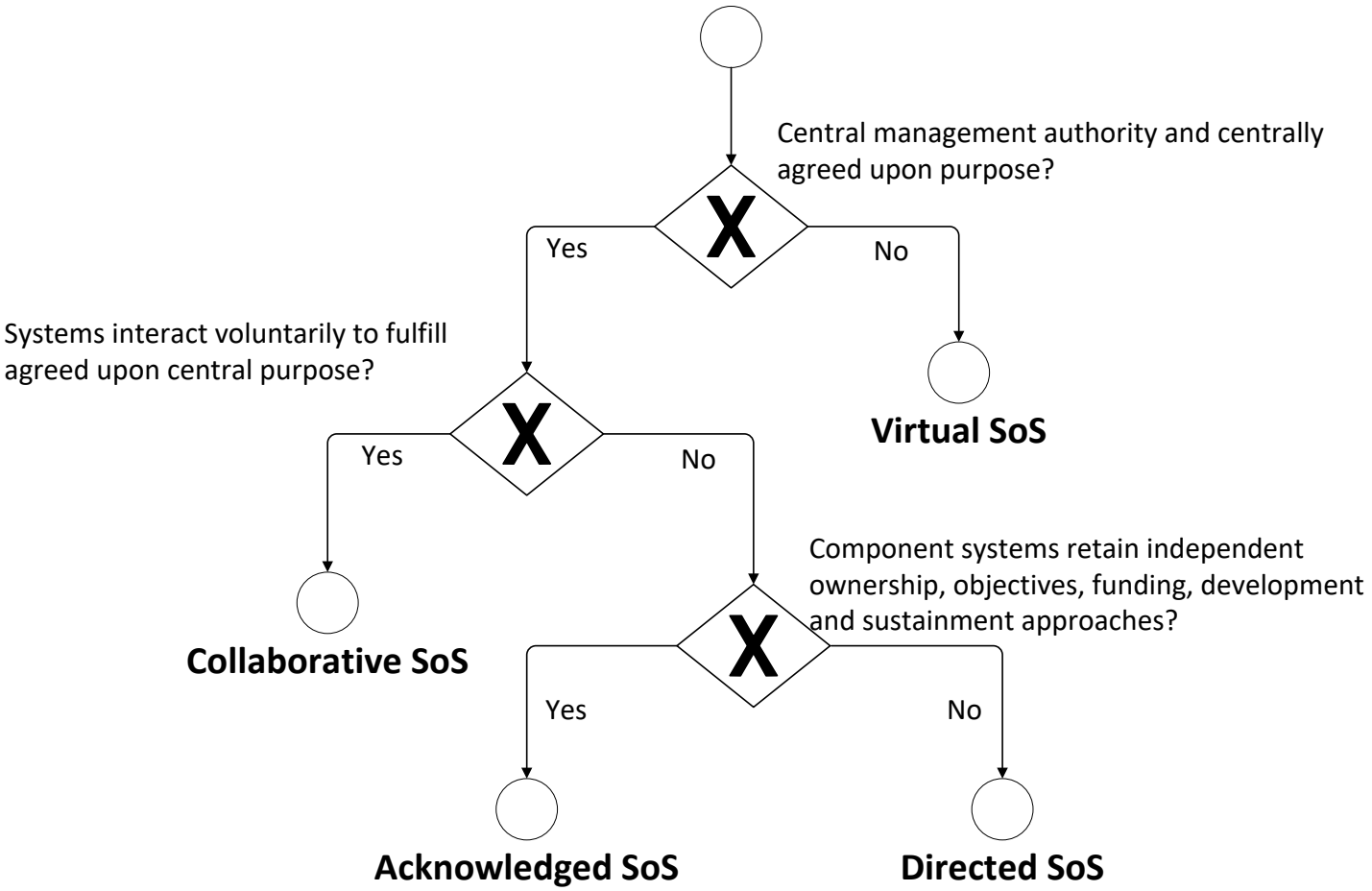
- **Virtual**. Virtual SoS lack a central management authority and a centrally agreed upon purpose for the system-of-systems. Large-scale behavior emerges—and may be desirable.

- **Collaborative.** Component systems interact more or less voluntarily to fulfill agreed upon central purposes. For example, the Internet is a collaborative system. The Internet Engineering Task Force works out standards but has no power to enforce them.

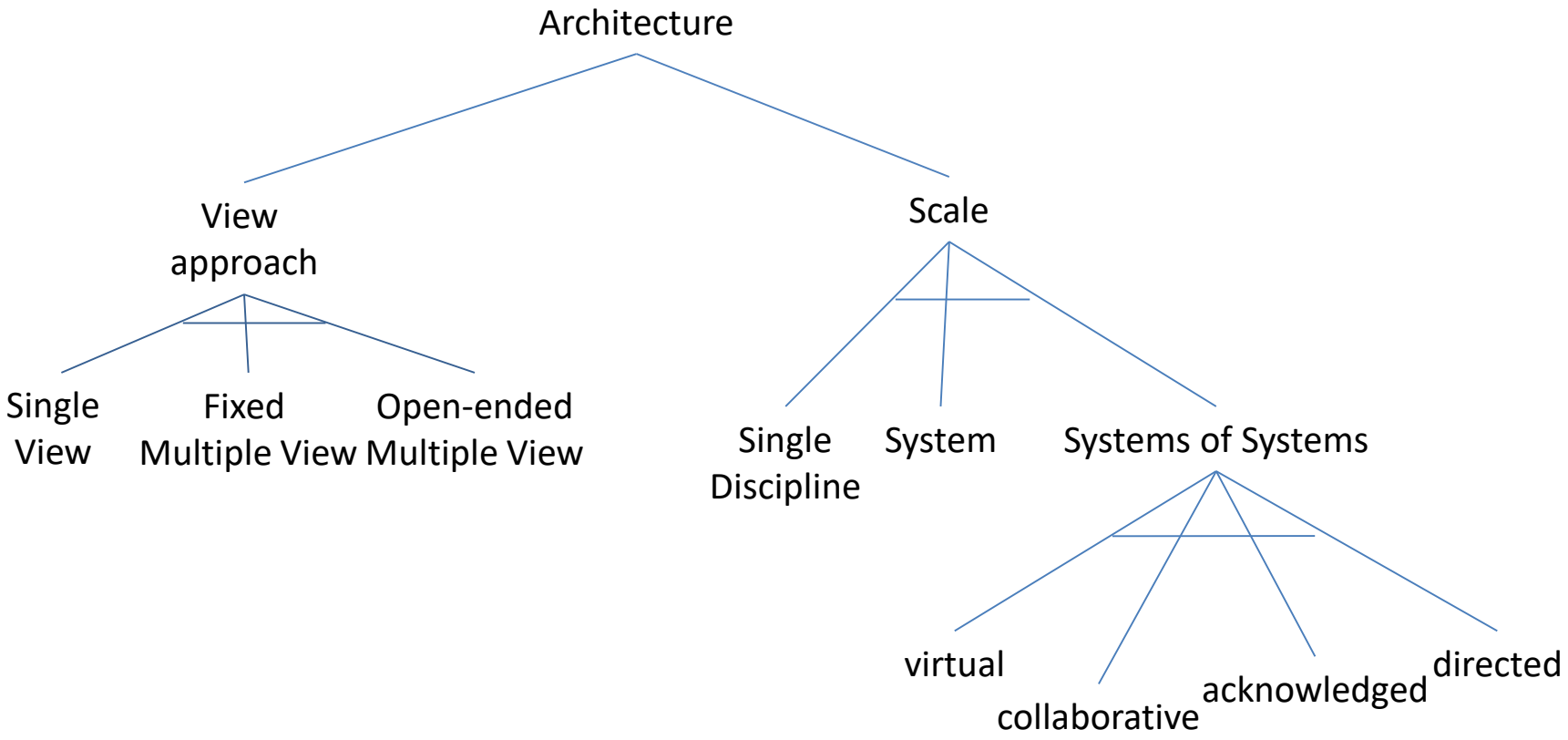
- **Acknowledged**. Recognized objectives, a designated manager, however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches.

- **Directed**. integrated system-of-systems is built and managed to fulfill specific purposes. Centrally managed; component systems subordinated to the central managed purpose.

Types of SoS


Central management authority and centrally agreed upon purpose?
Yes
No
Systems interact voluntarily to fulfill agreed upon central purpose?
Yes
No
Virtual SoS
Component systems retain independent ownership, objectives, funding, development and sustainment approaches?
Collaborative SoS
Yes
No
Acknowledged SoS
Directed SoS

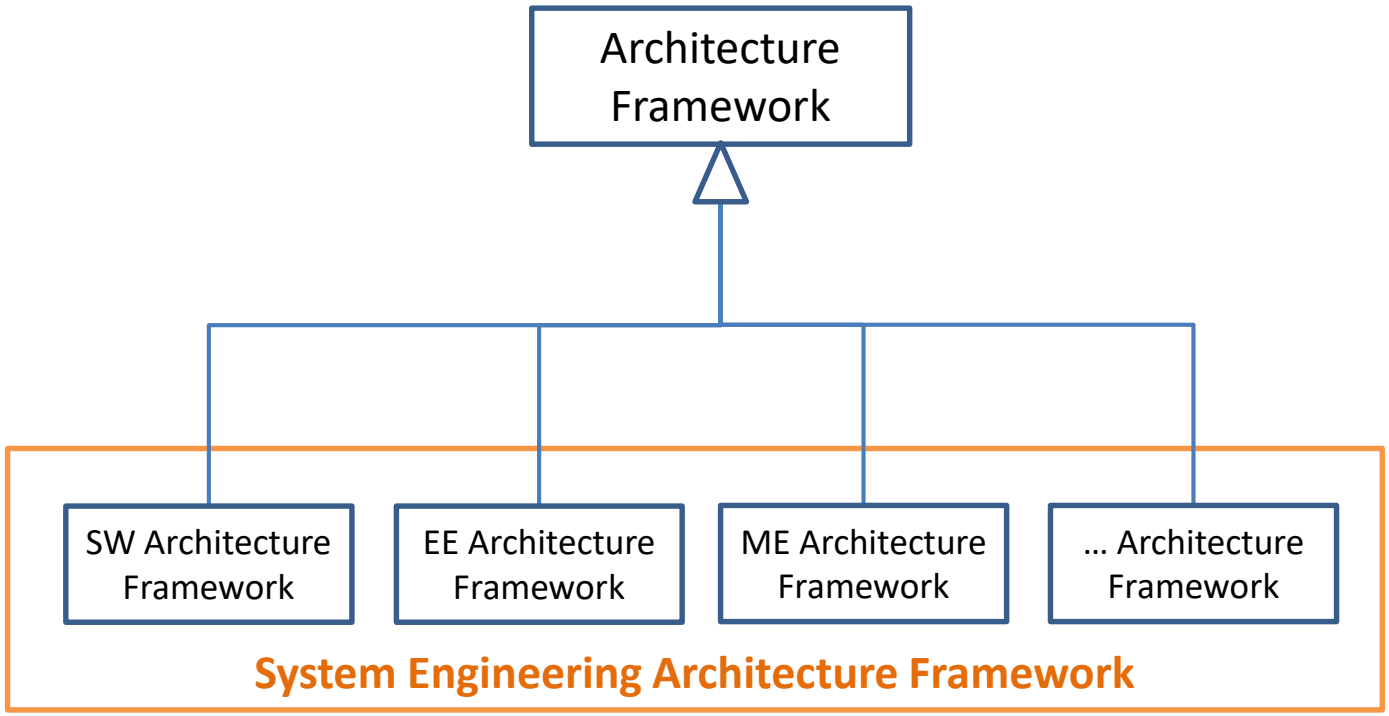
System Development



Questions…

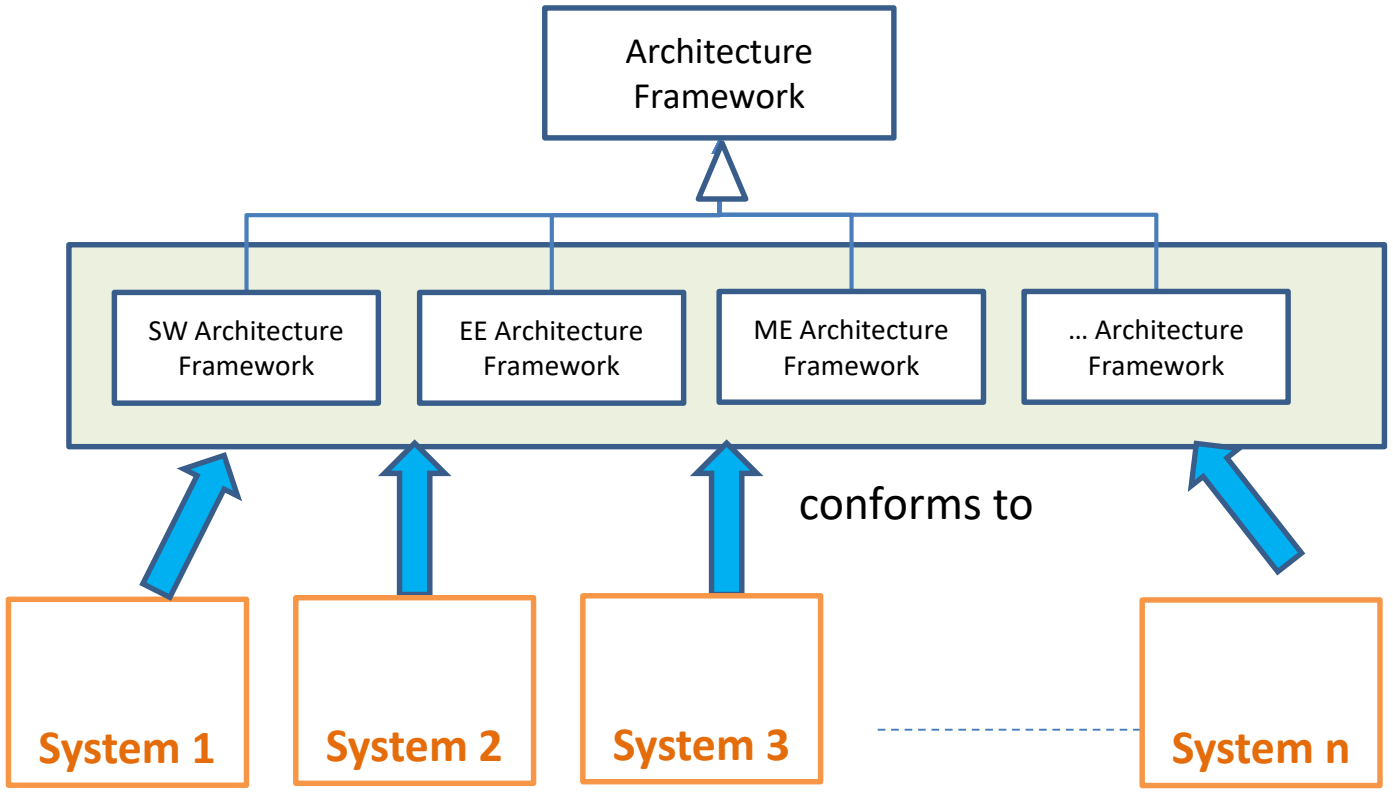
- Do we need additional viewpoints for SoS level?
- Or can we use the viewpoints for system engineering (SoS is an example of SE)?
- How to use viewpoints for different types of SoS?
- How to cope with uncertainty about stakeholders?

System Engineering AF



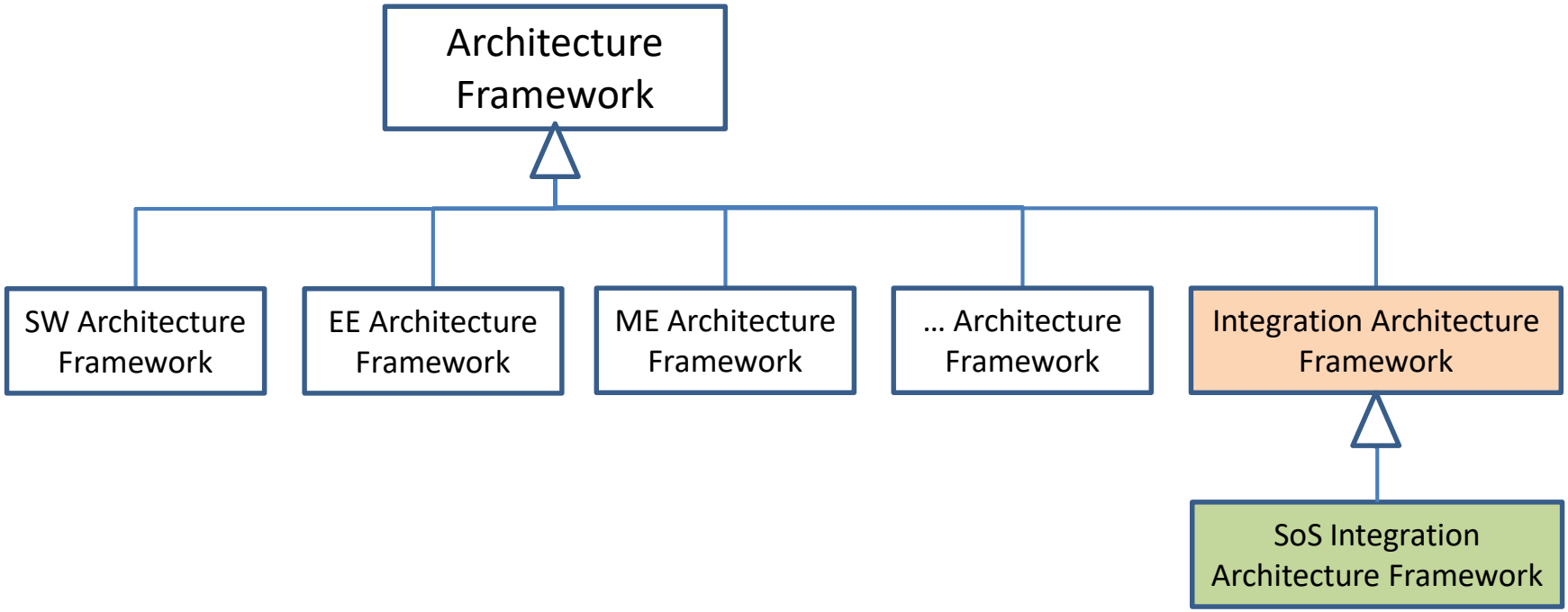
System Engineering AF

- Use same set of System Engineering viewpoints



System Engineering AF

- Use additional viewpoints



Summary

Summary

- System Engineering requires modeling the architecture from different perspectives
- Currently no coherent system engineering architecture framework exist yet
- Different alternatives can be defined for designing SE Architecture Framework
  - All-in-one approach in which one architecture framework is with viewpoints for representing the architecture of the system
  - Different architecture frameworks for different disciplines
- Important criteria to consider:
  - Separation of Concerns
  - Integrated View
  - Learning Curve

Summary

- System-of-Systems Engineering provides even a further broadening of the scope
- However SoS could be considered as system engineering
- As such viewpoints for system engineering could be reused
  - This needs to be justified
- Different SoS types might impose different constraints on the system