



Architecting Fundamentals

Integrated Modular Solution Architectures

May 2012
Version 0.8

***Rockwell
Collins***

What is System Architecture?

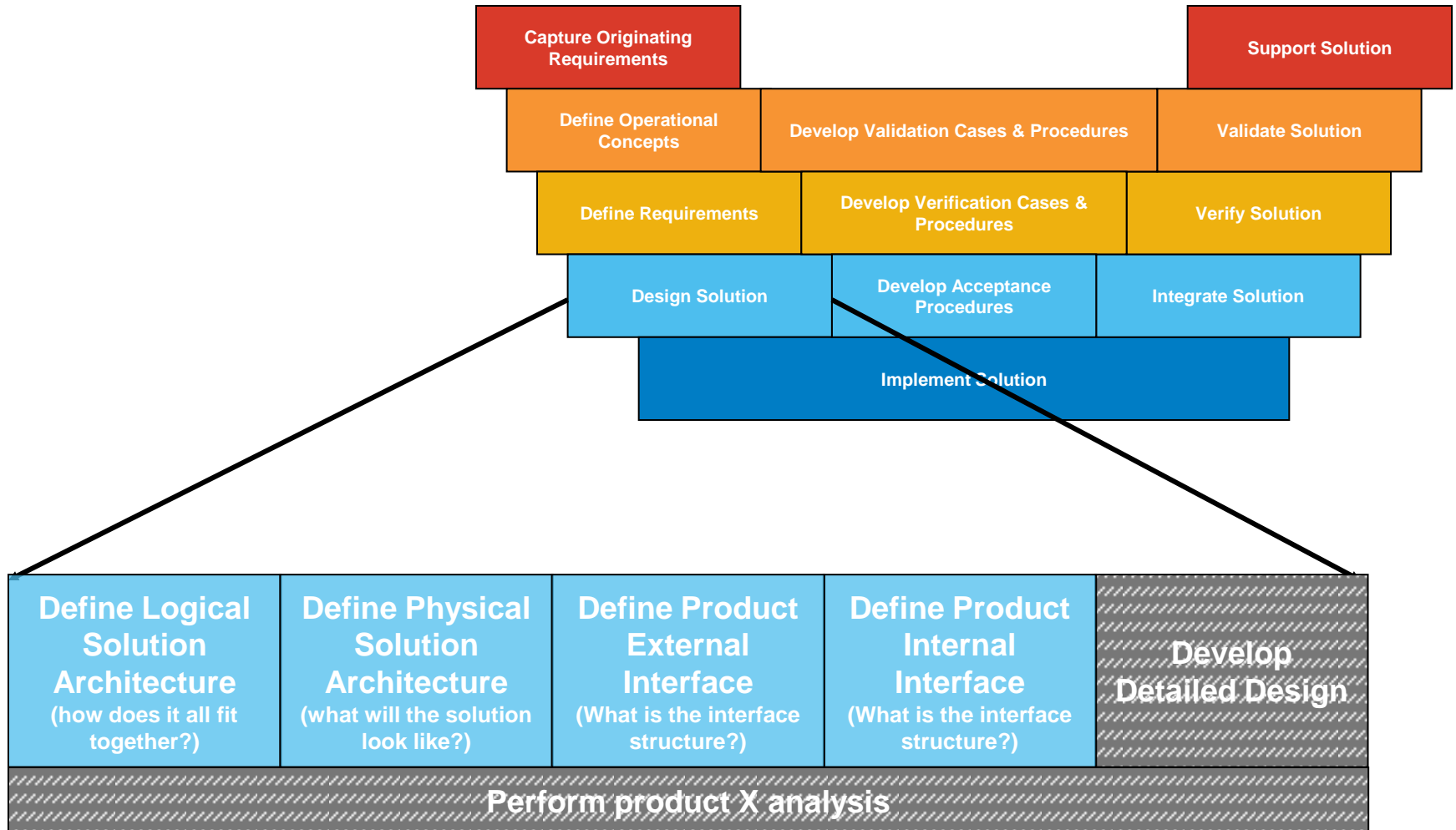
- Wikipedia:
 - A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system.
- IEEE:
 - The composite of the physical architectures for consumer products and their life-cycle processes. (P1220)
 - The organizational structure of a system or component. (STD 610.12)
 - A logical or physical representation of a product which depicts its structure, but, provides few or no implementation details. (P1220)
- DERA
 - The structure of levels and/or branches that partition a system into its constituent parts or components.
- NASA
 - How functions are grouped together and interact with each other. (MDP92)



Definition of System Architecture

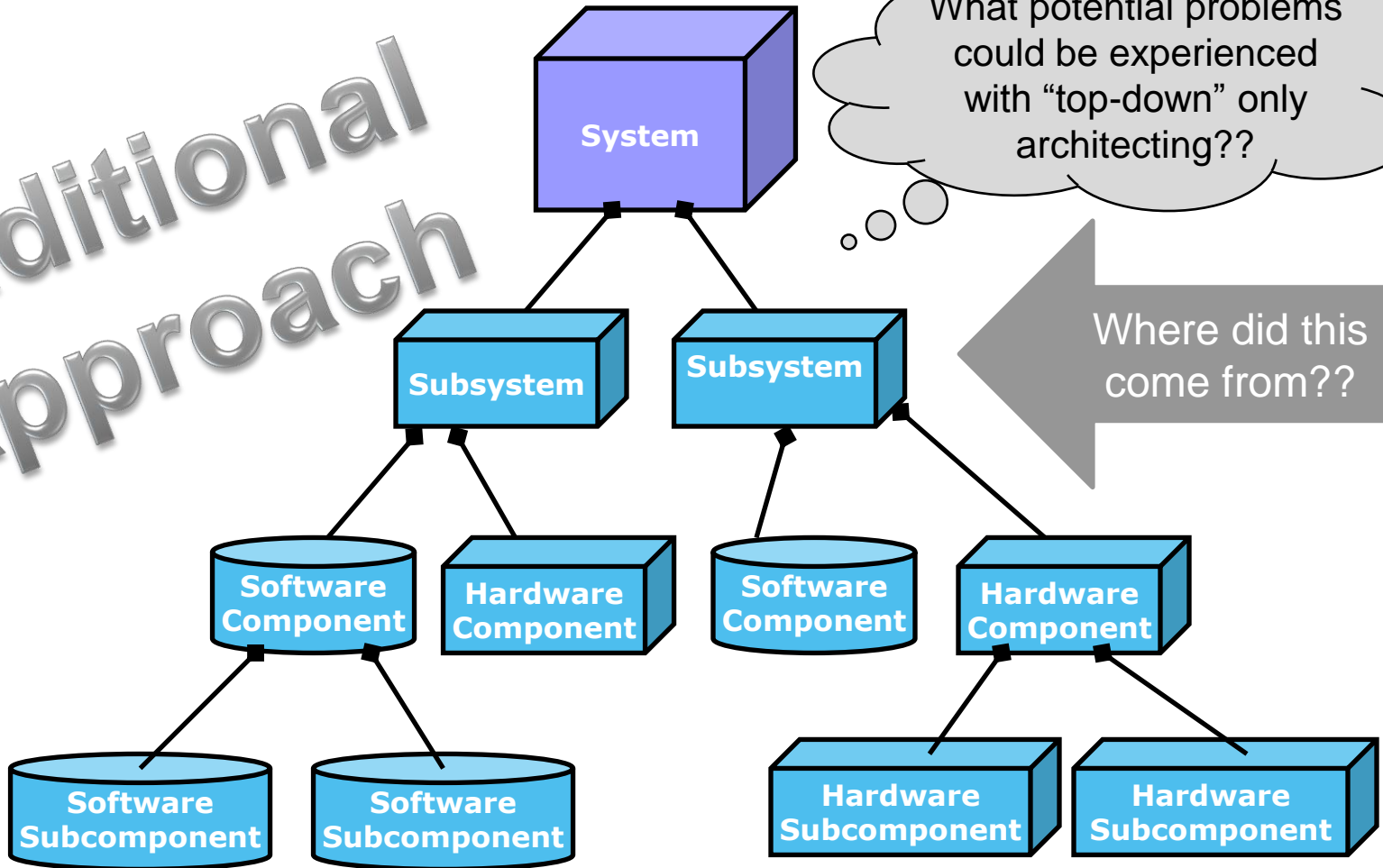
- As used in Rockwell Collins Architecture Standard
 - The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.

Architecture in the TCP



Top-Down Hierarchical System Architecture

Traditional
Approach



Traditional Top-Down System Approach



- What is a “subsystem”?
- How does a “system” differ from a “subsystem”?
- Can a system share components with other systems within the same hierarchy?
- What happens when the “subsystem” does not have unique components?
- How has “plug and play” and “modularity” changed the way we engineer or manage our products?

Is there a better way?

Characteristics of Integrated Modular Solution Architecture (IMSA)

- Independent software architectures
 - Software architecture can **exist** independently from underlying physical hardware architecture!
 - Two projects with identical software architecture could have very different hardware configurations.
 - Software applications/ programs can **reside** anywhere in overall physical hardware architecture
 - Application software design abstracts out hardware interface so that it is not tied to a single box
 - Software architecture **defined** independently from the hardware
 - Allocation of application software to specific processors requires analysis to assess resource usage
 - Concerns – latency, throughput, and processor loading – can the underlying architecture handle resource usage?

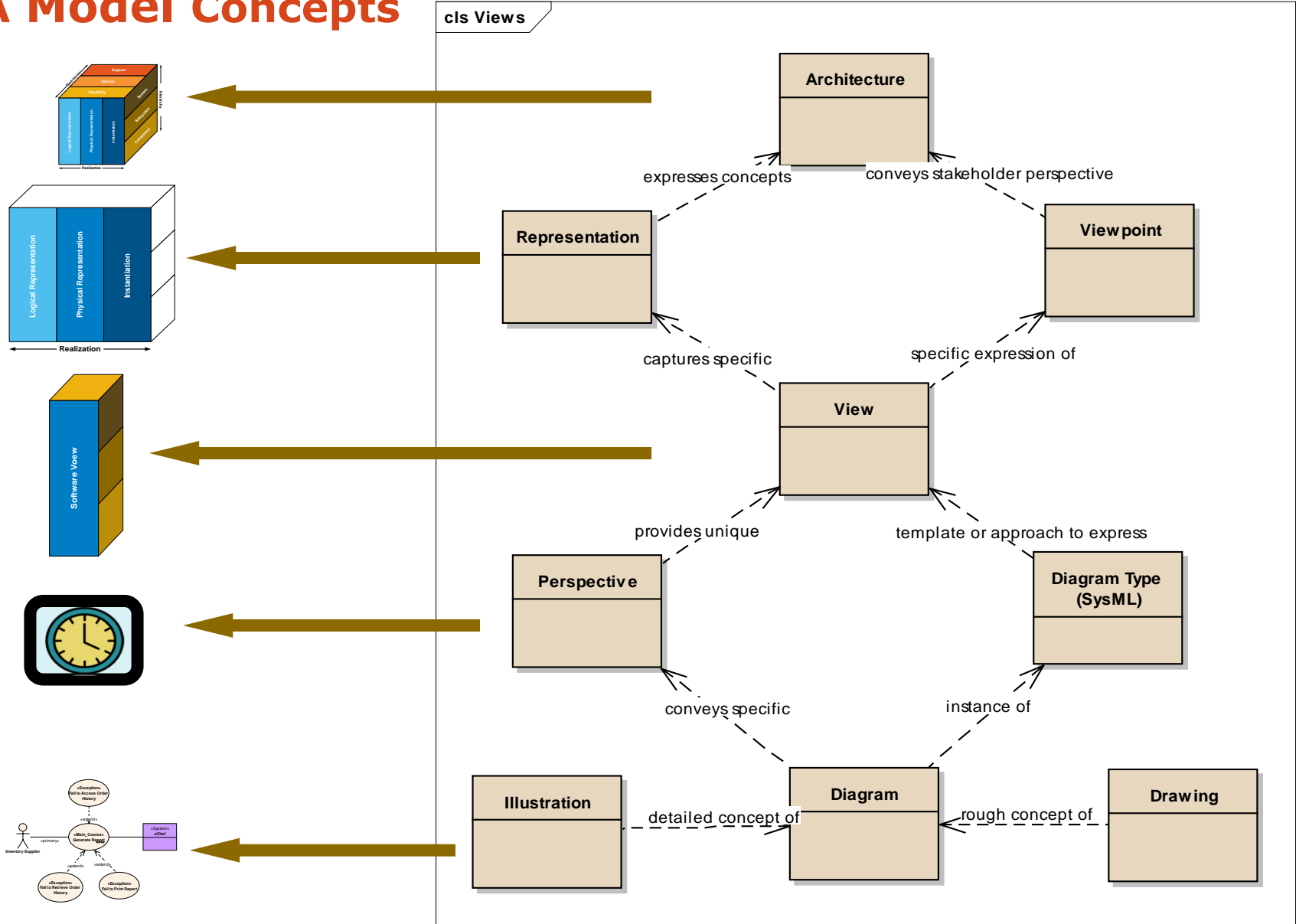
Characteristics of IMSA

- Processor architectures allowing more densely packed boxes
 - Will that be one processor? Or two? Or three? On one card? On one chip?
 - Shared memory? Dedicated memory?
 - Shared I/O? Dedicated I/O?
 - Processor architecture can be defined independently of mechanical architecture/ packaging
 - Allocation of processors to specific boards requires analysis to assess resource usage
 - Concerns – package density, EMI, thermal loading – can the underlying mechanical structure handle resource usage?

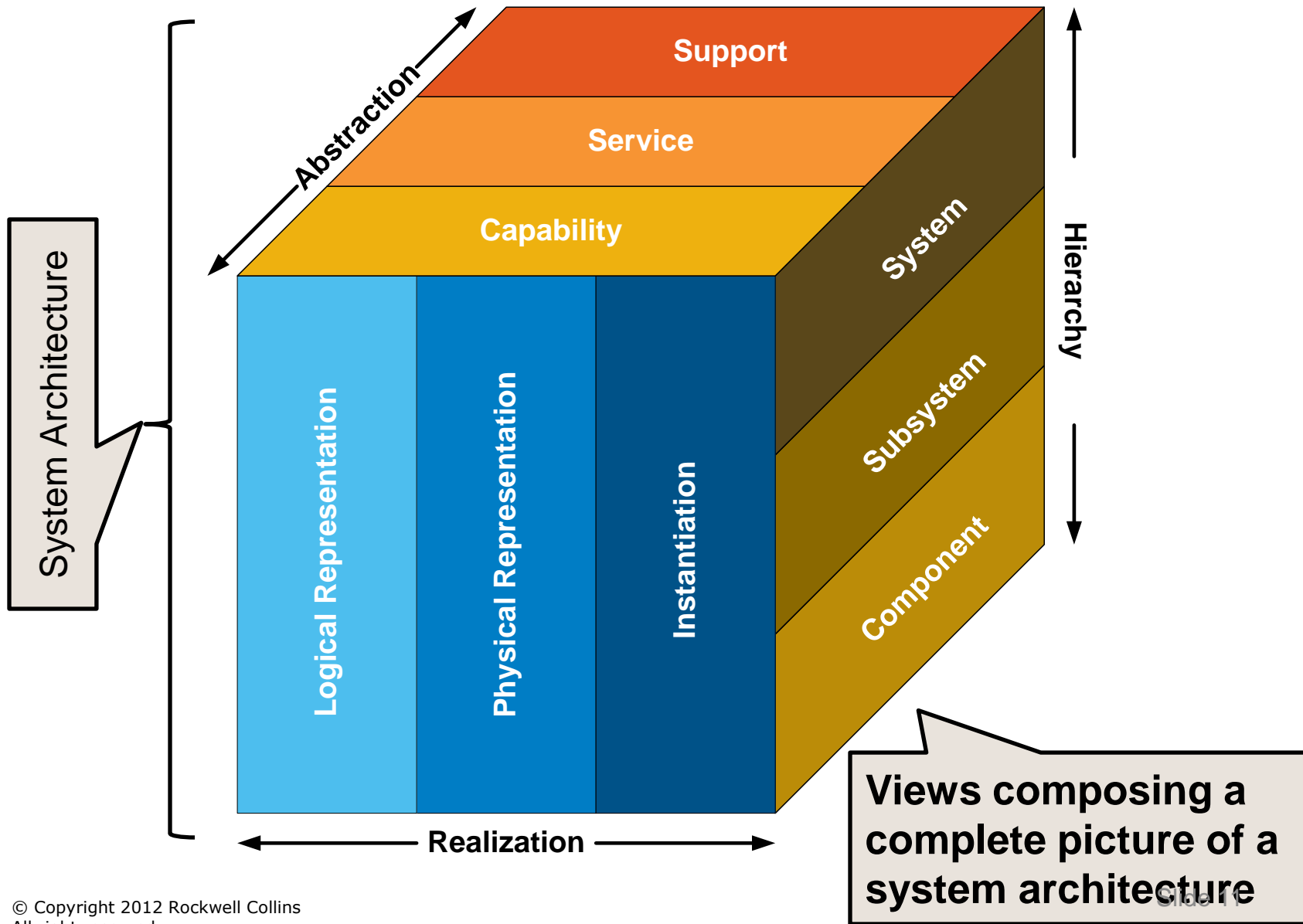
Solution Architecture Views

- No one “picture” shows the full architecture of the solution
 - How is the architecture captured and managed?
 - Is a hierarchy view enough?
- Architecture contains
 - **Components** with structure and composition
 - **Behavior** showing inputs, processing, and outputs
 - **Relationships** showing interconnectivity of the solution
- Multiple perspectives result in multiple views of the architecture
 - Software, Hardware, Behavioral, etc.
- Interdependencies between views must also be captured
 - Interfaces and relationships

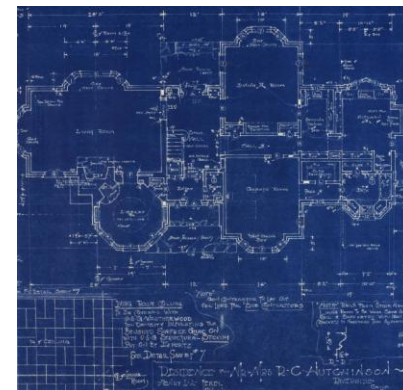
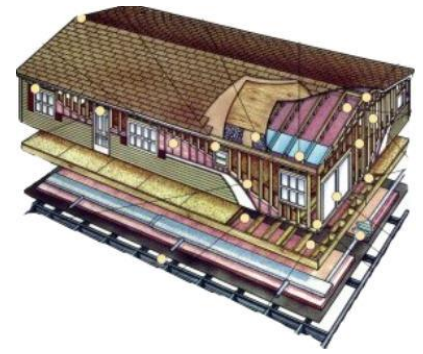
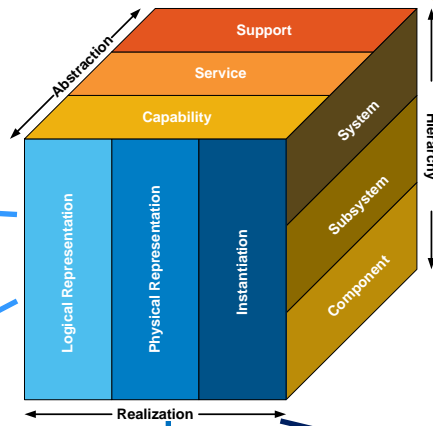
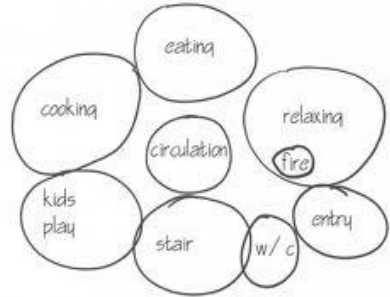
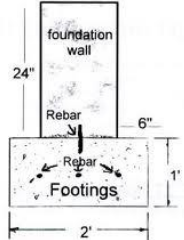
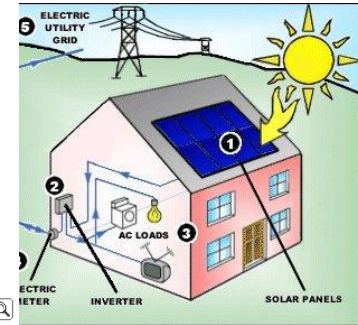
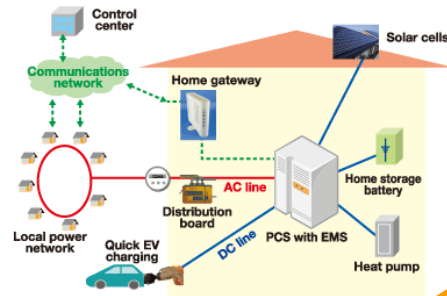
IMSA Model Concepts



Integrated Modular Solution Architecture (IMSA)



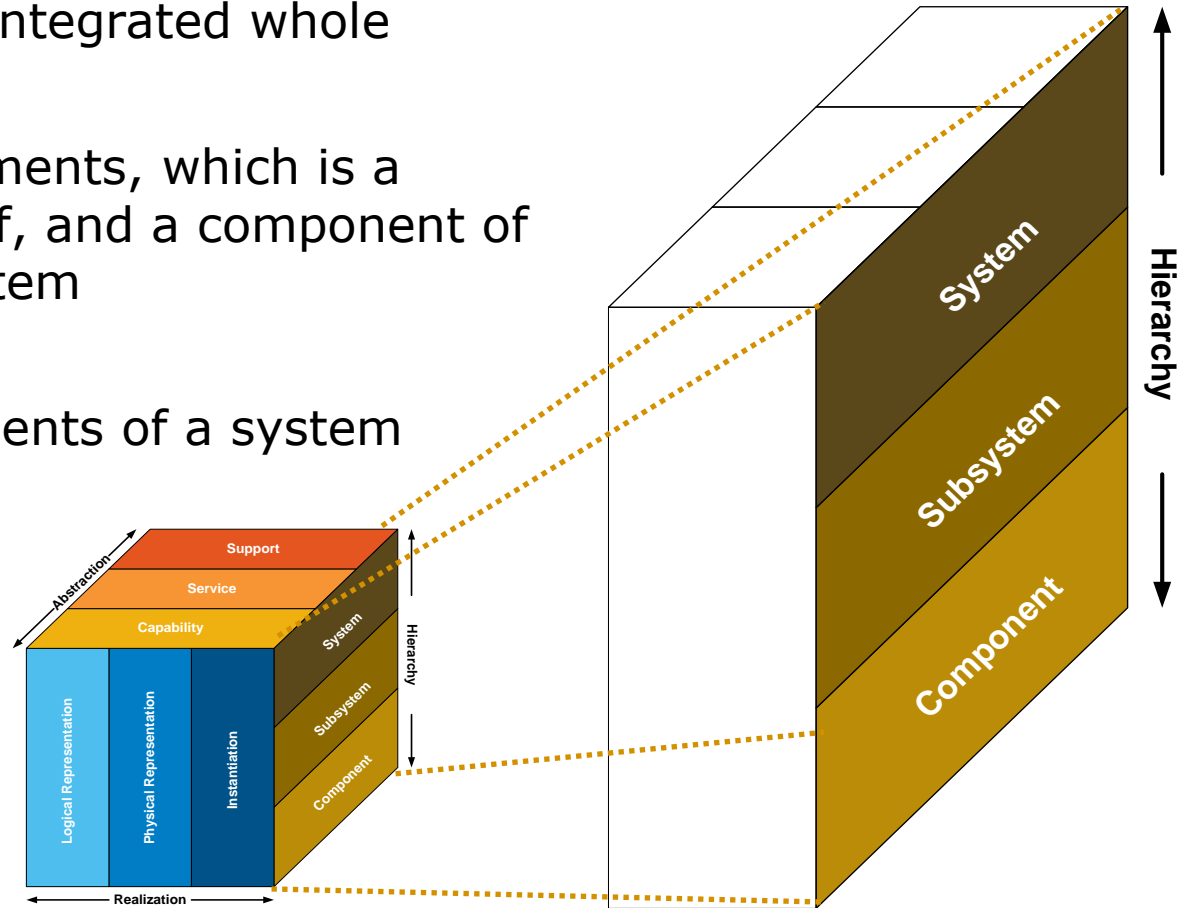
"Your House for Example"



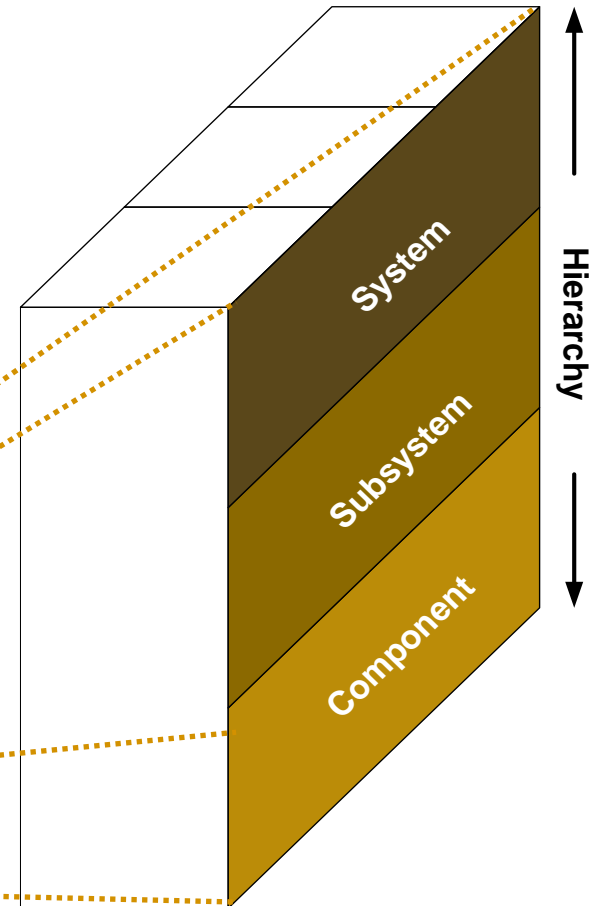
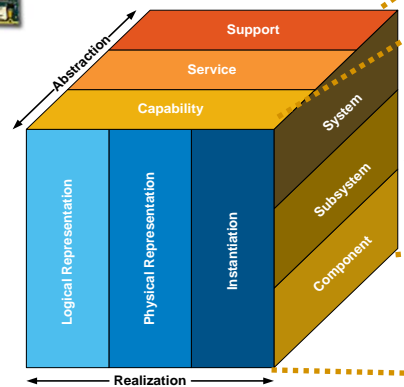
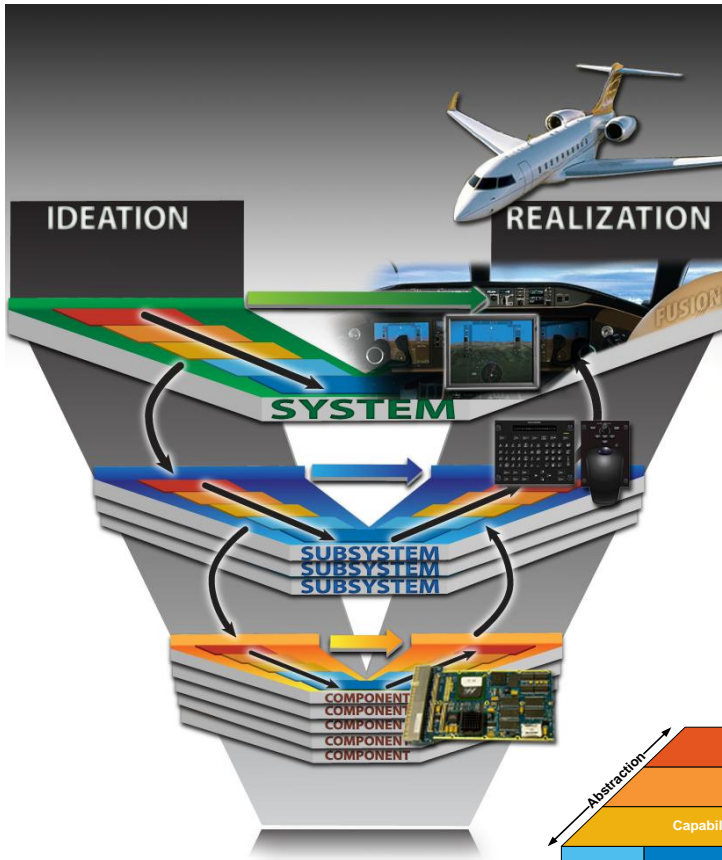
Hierarchy Tiers

- System
 - A set of interacting or interdependent components forming an integrated whole
- Subsystem
 - A set of elements, which is a system itself, and a component of a larger system
- Component
 - The constituents of a system

Definitions from
Wikipedia



Hierarchy Tiers in the TCP

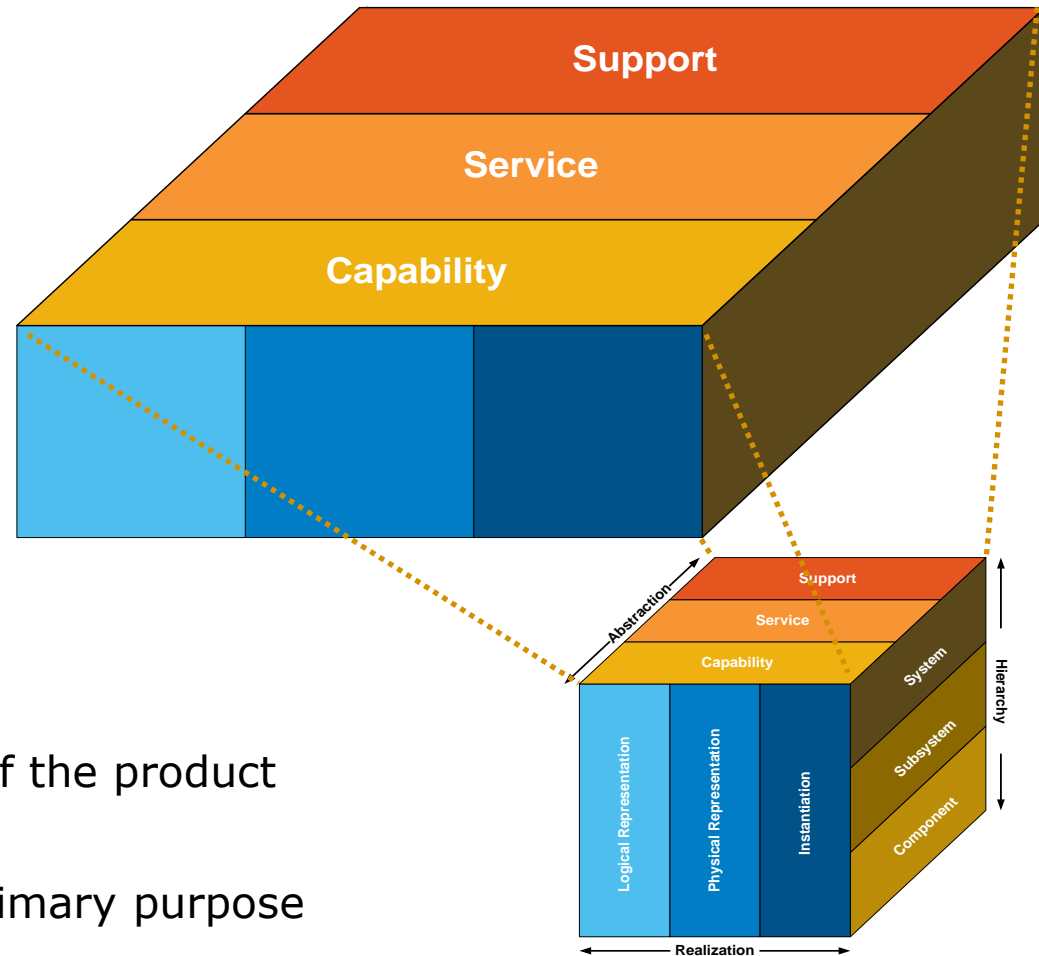


Abstraction Views

Abstraction layers partition architectural elements into isolated layers so that changes can be made in one layer without affecting the other layers

Enables modularity

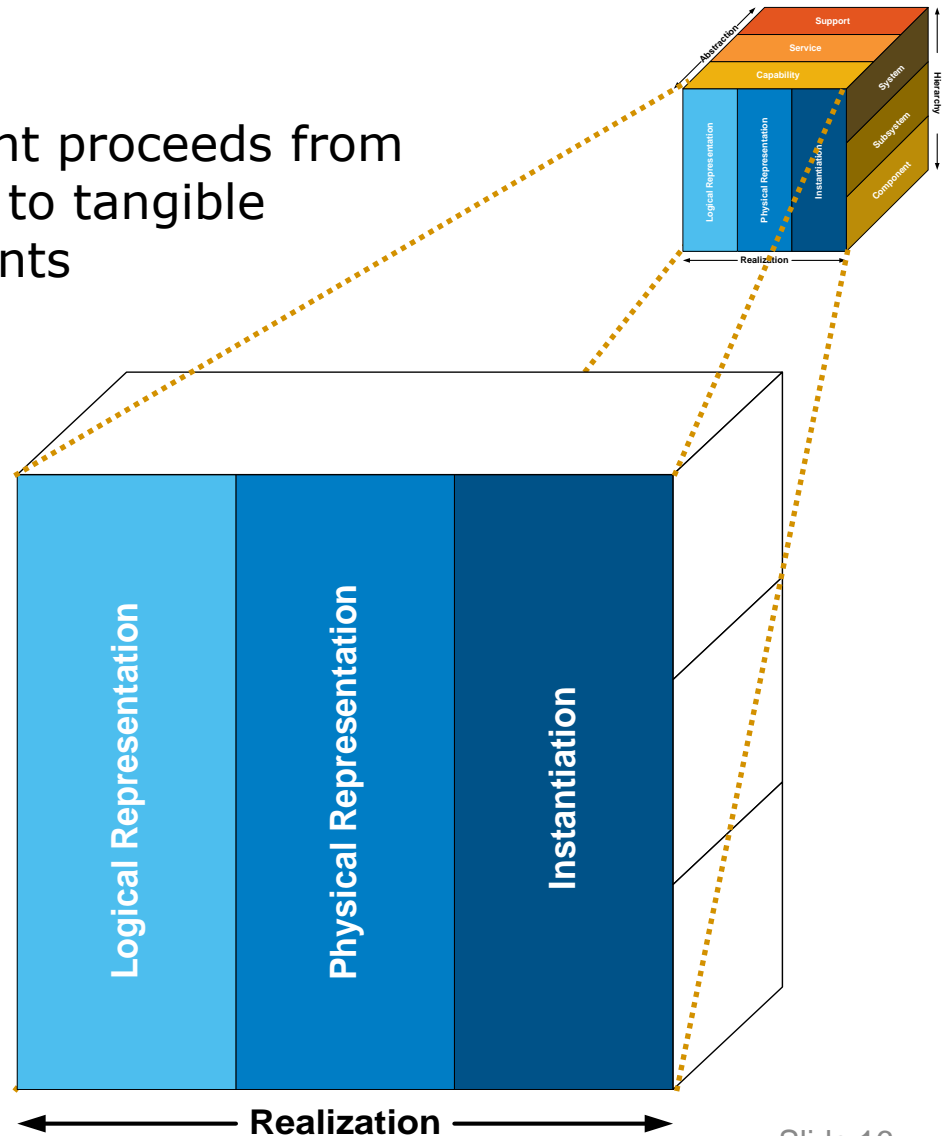
- **Capability Views**
 - Mission or primary purpose of the product
- **Service Views**
 - Needed services to enable primary purpose
- **Support Views**
 - Foundational elements to bind the solution together



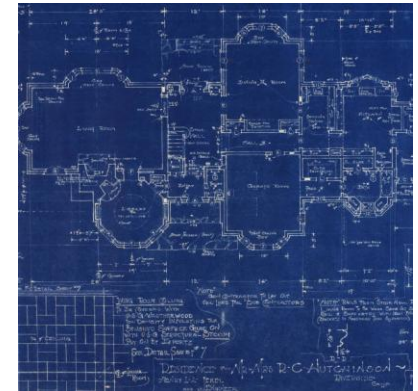
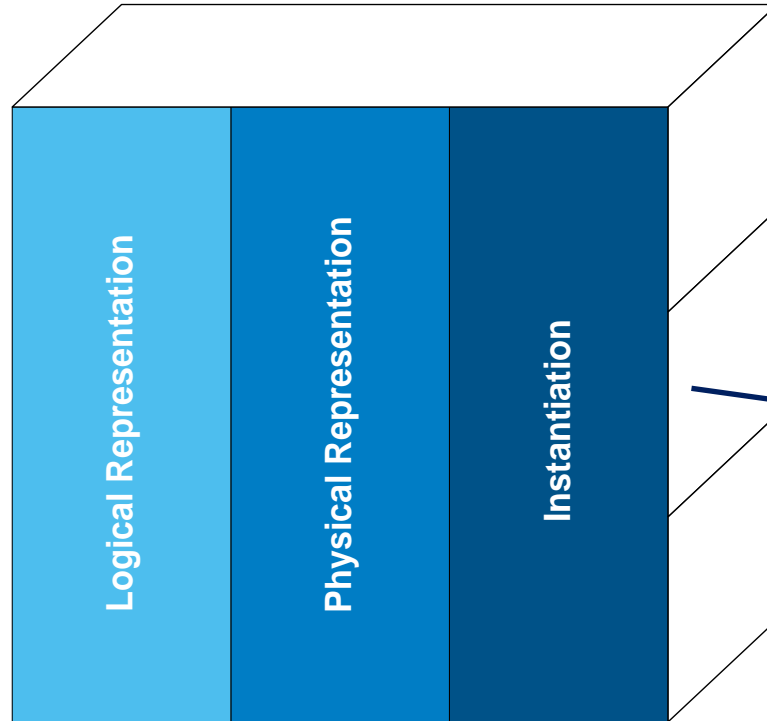
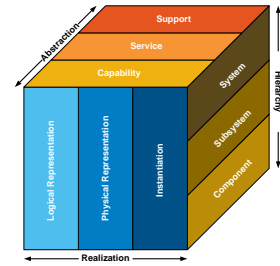
Service and Support layers are optional, depending on depth of abstraction needed in your architecture. *Slide 15*

Realization Views

- Progression from left to right proceeds from abstract, notional concepts to tangible assets to realized components



Realization Views



← Realization →



Architectural Perspectives



- Hierarchical Perspective
 - Focused on organization of things



- Chronological/ Sequential Perspective
 - Focused on the timing of things



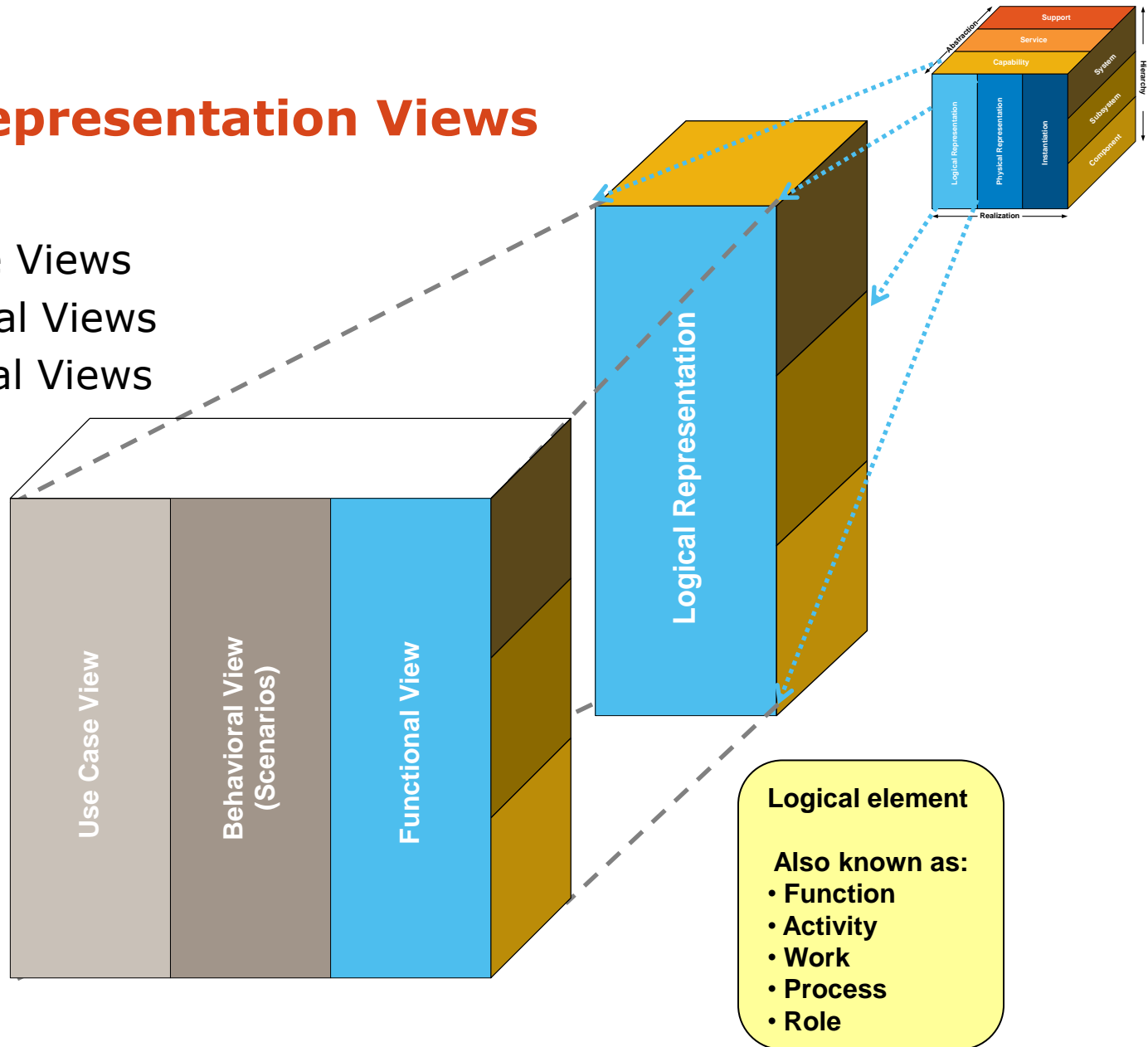
- Transactional Perspective
 - Focused on the transfer of interface objects



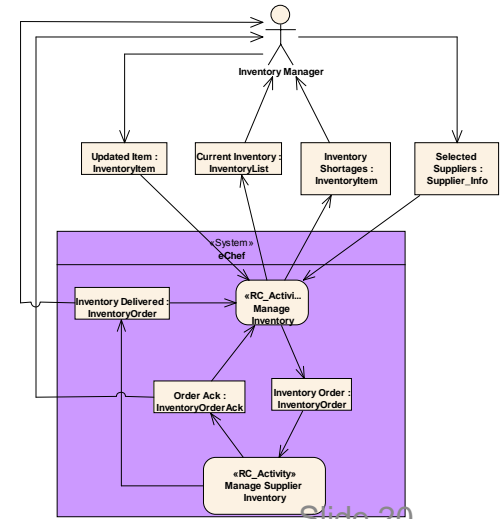
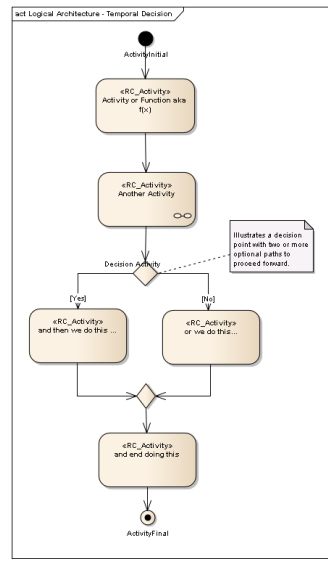
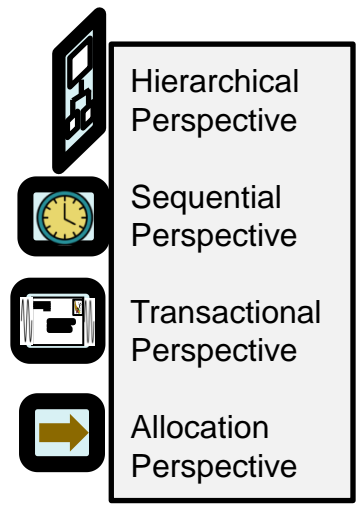
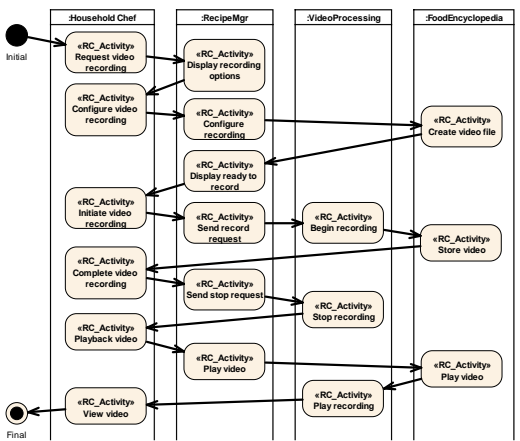
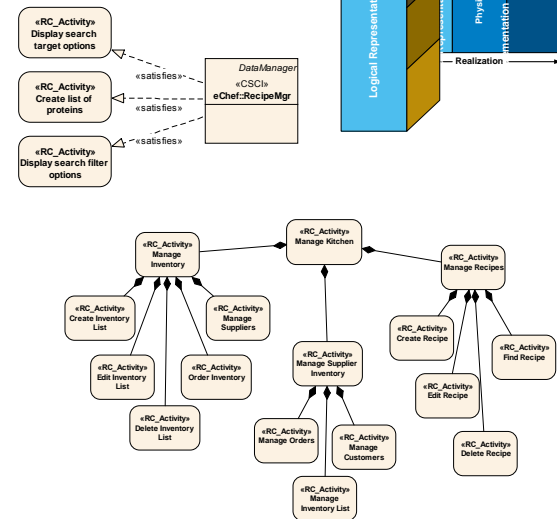
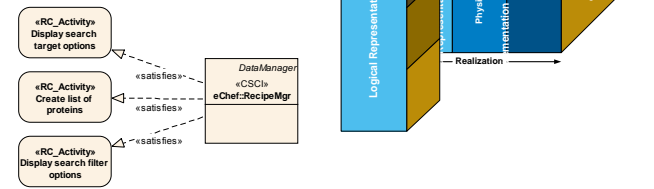
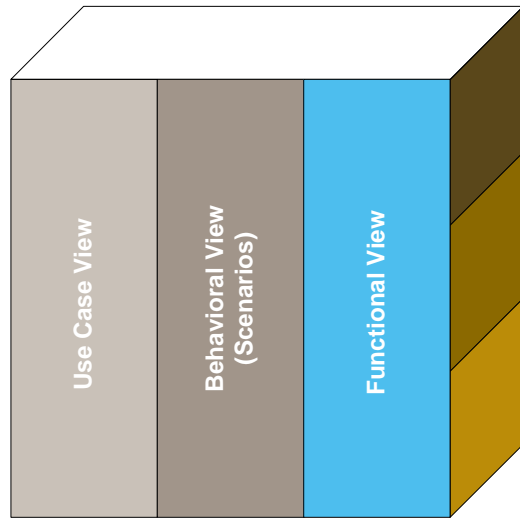
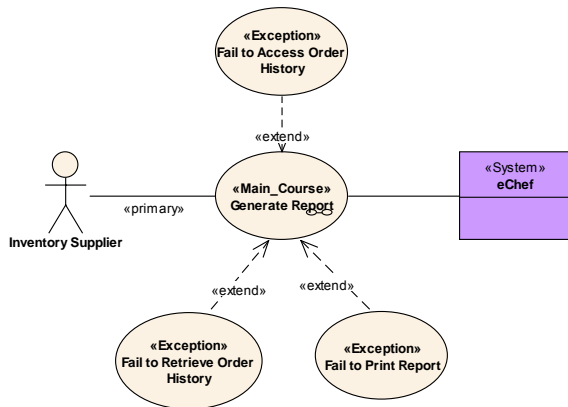
- Allocation (Deployment) Perspective
 - Focused on the realization of things – where does it live?

Logical Representation Views

- Use Case Views
- Behavioral Views
- Functional Views

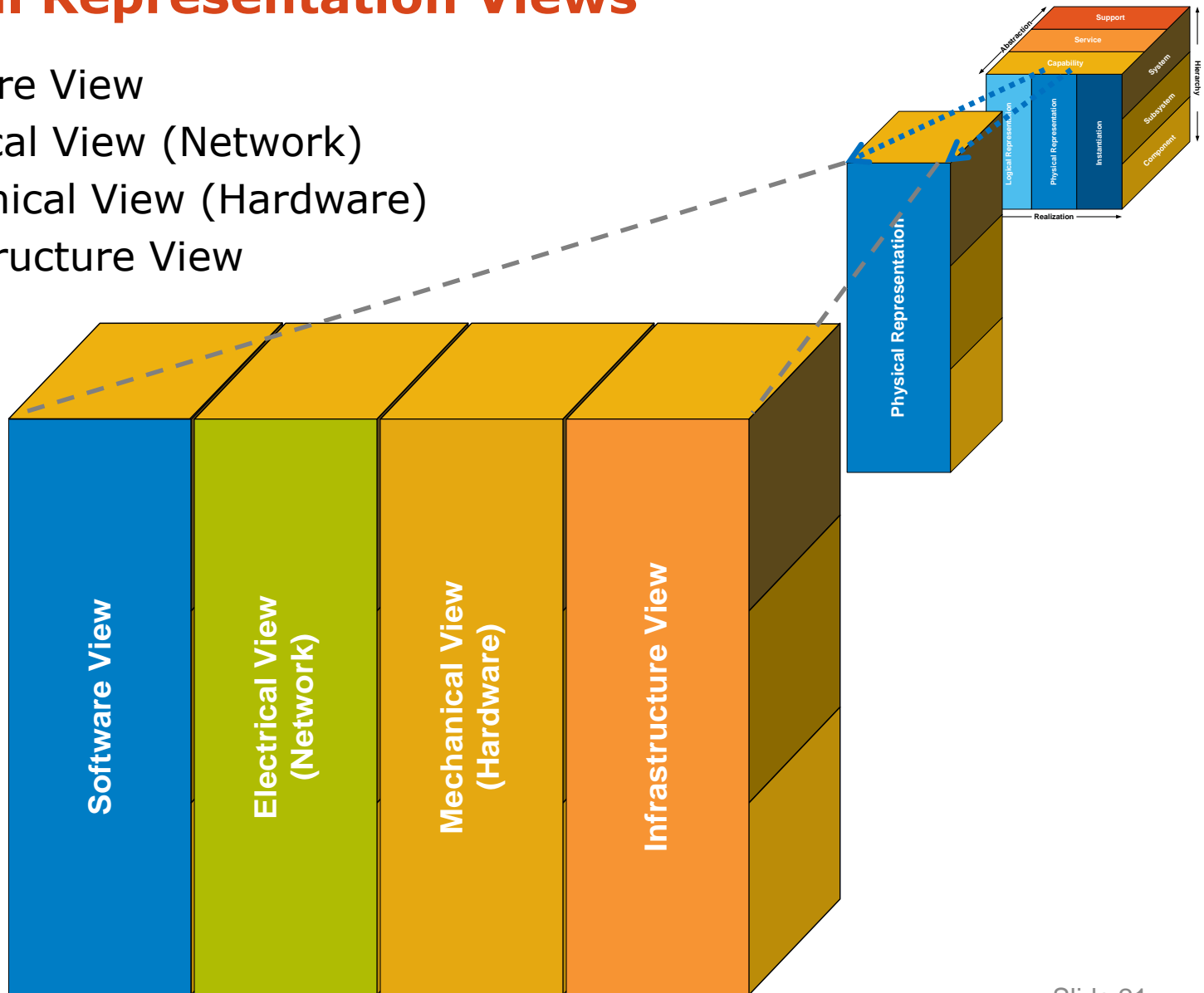


Logical Representations Summary

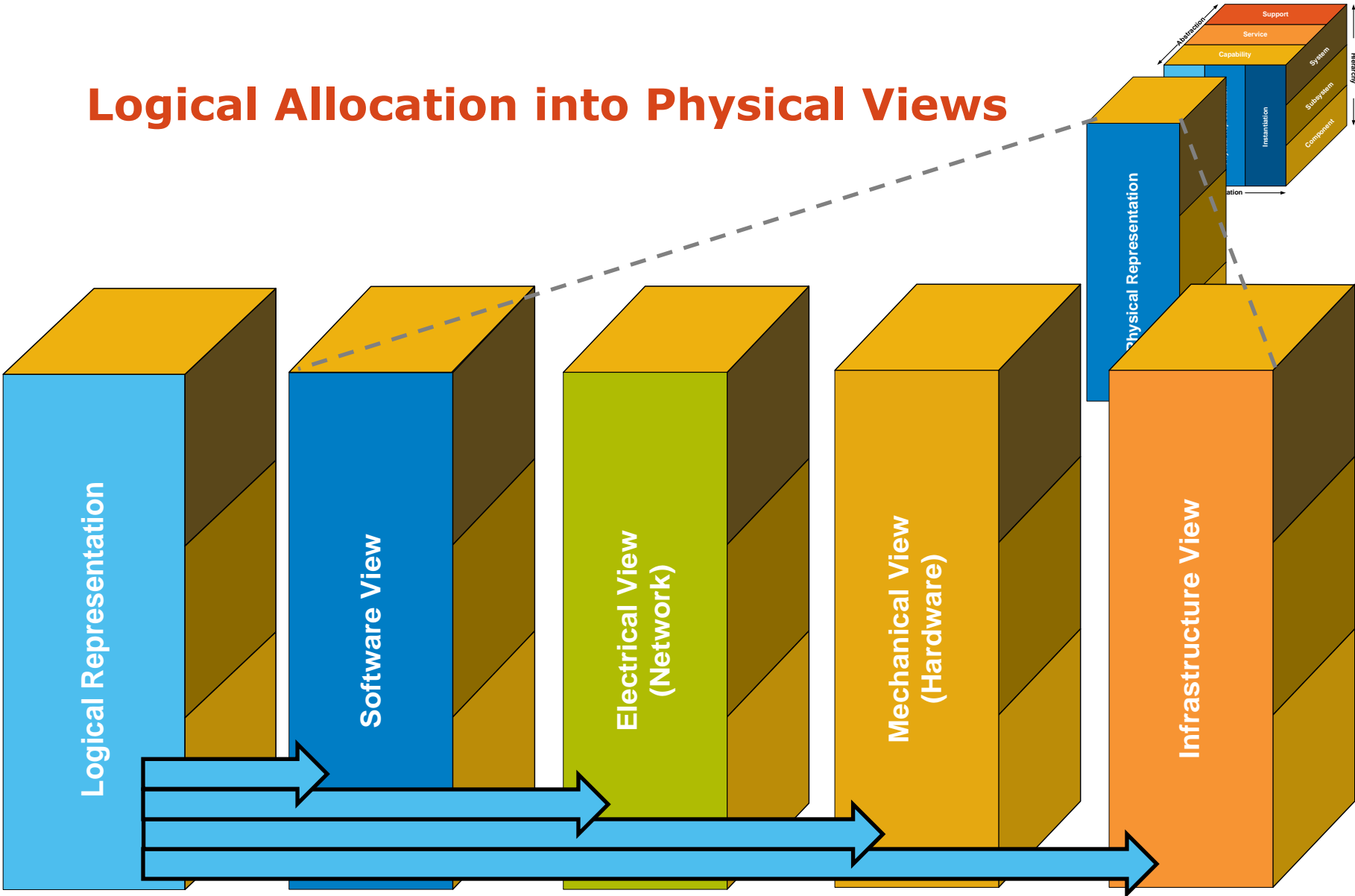


Physical Representation Views

- Software View
- Electrical View (Network)
- Mechanical View (Hardware)
- Infrastructure View



Logical Allocation into Physical Views



Physical Representation – Software View

- Software Views



Hierarchical Perspective



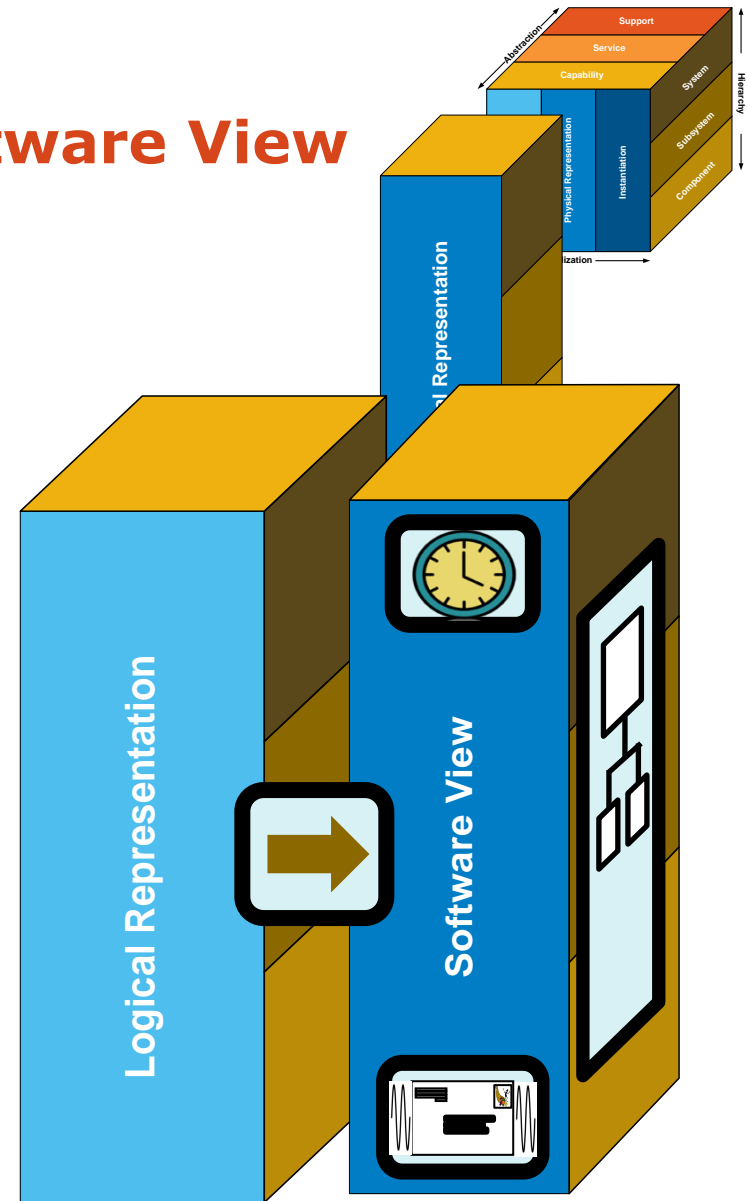
Sequential Perspective



Transactional Perspective (Messaging)

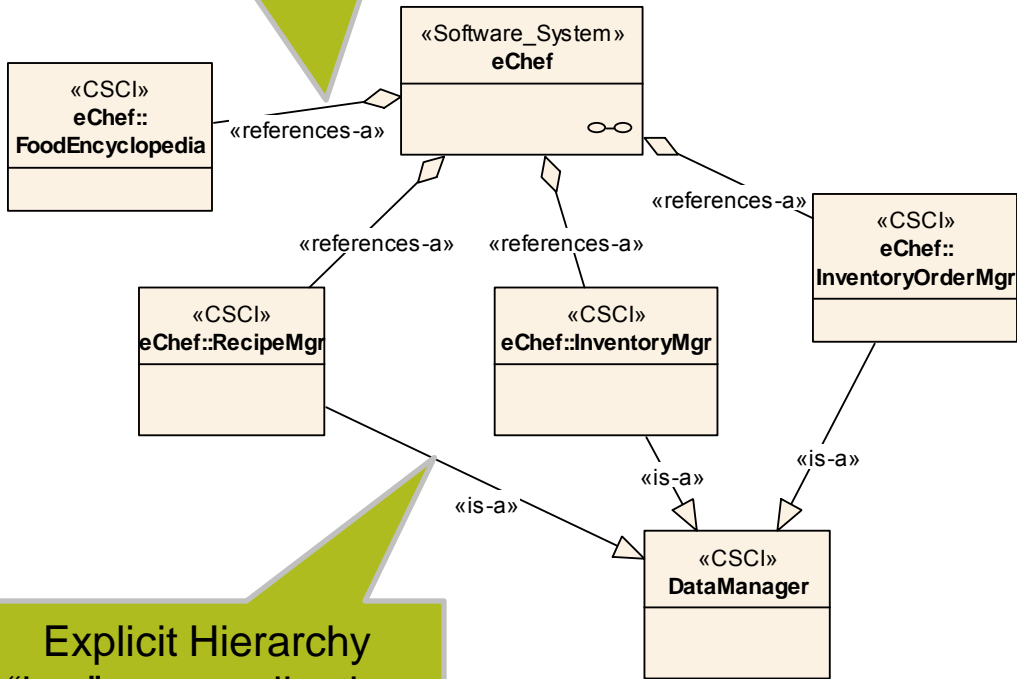


Allocation Perspective (from logical)

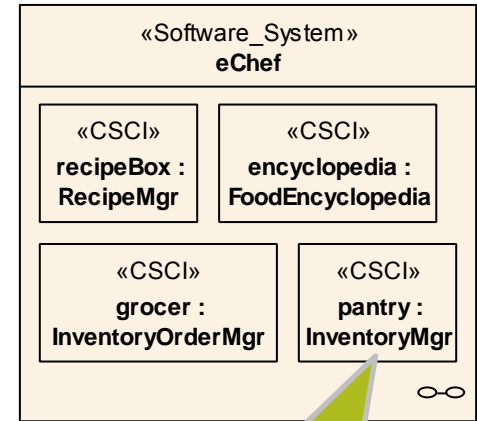


Software View – Hierarchical Perspective

Explicit Hierarchy
“a part of” -
aggregation

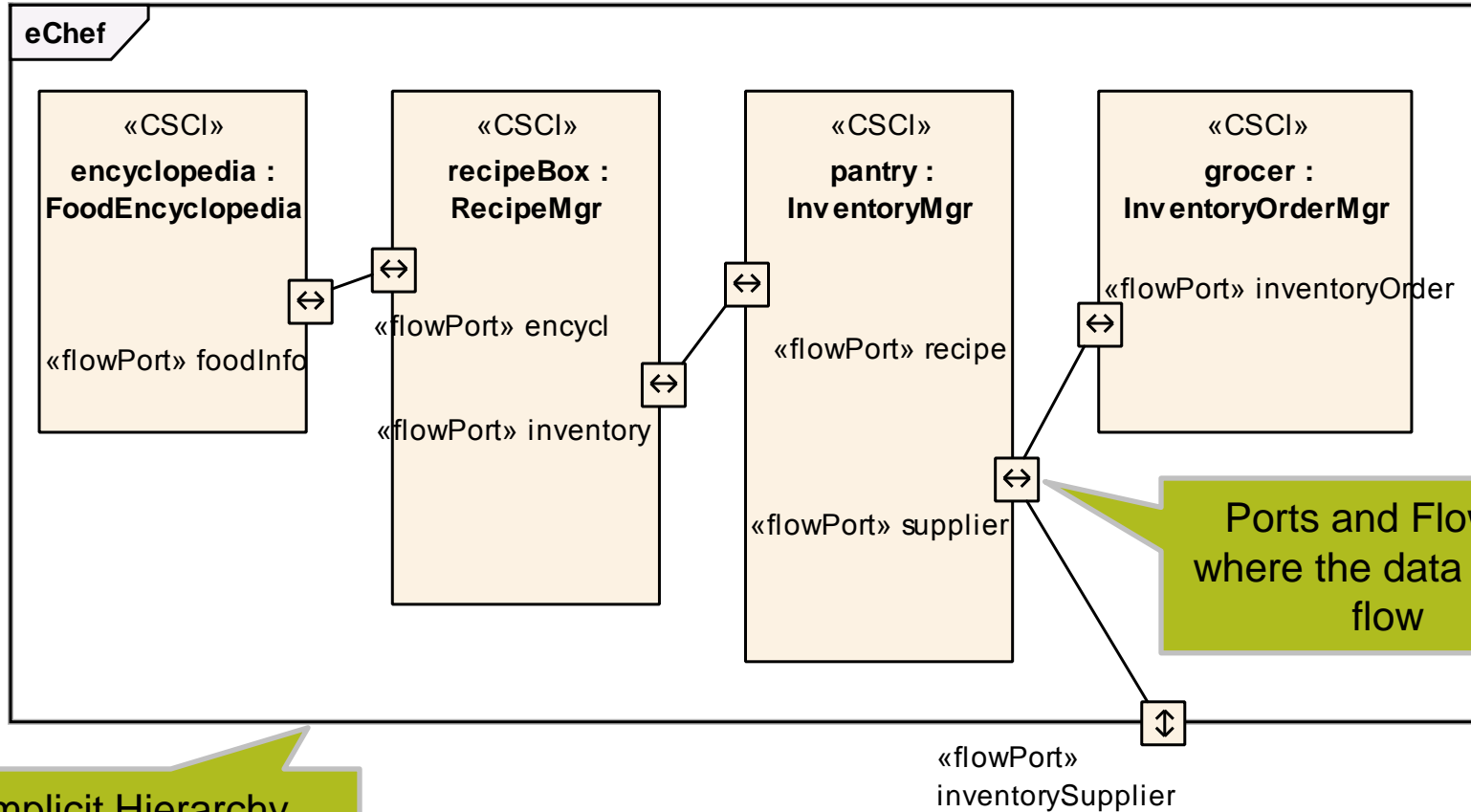


Explicit Hierarchy
“is a” - generalization



Implicit Hierarchy
superimposed

Software View – Transactional Perspective

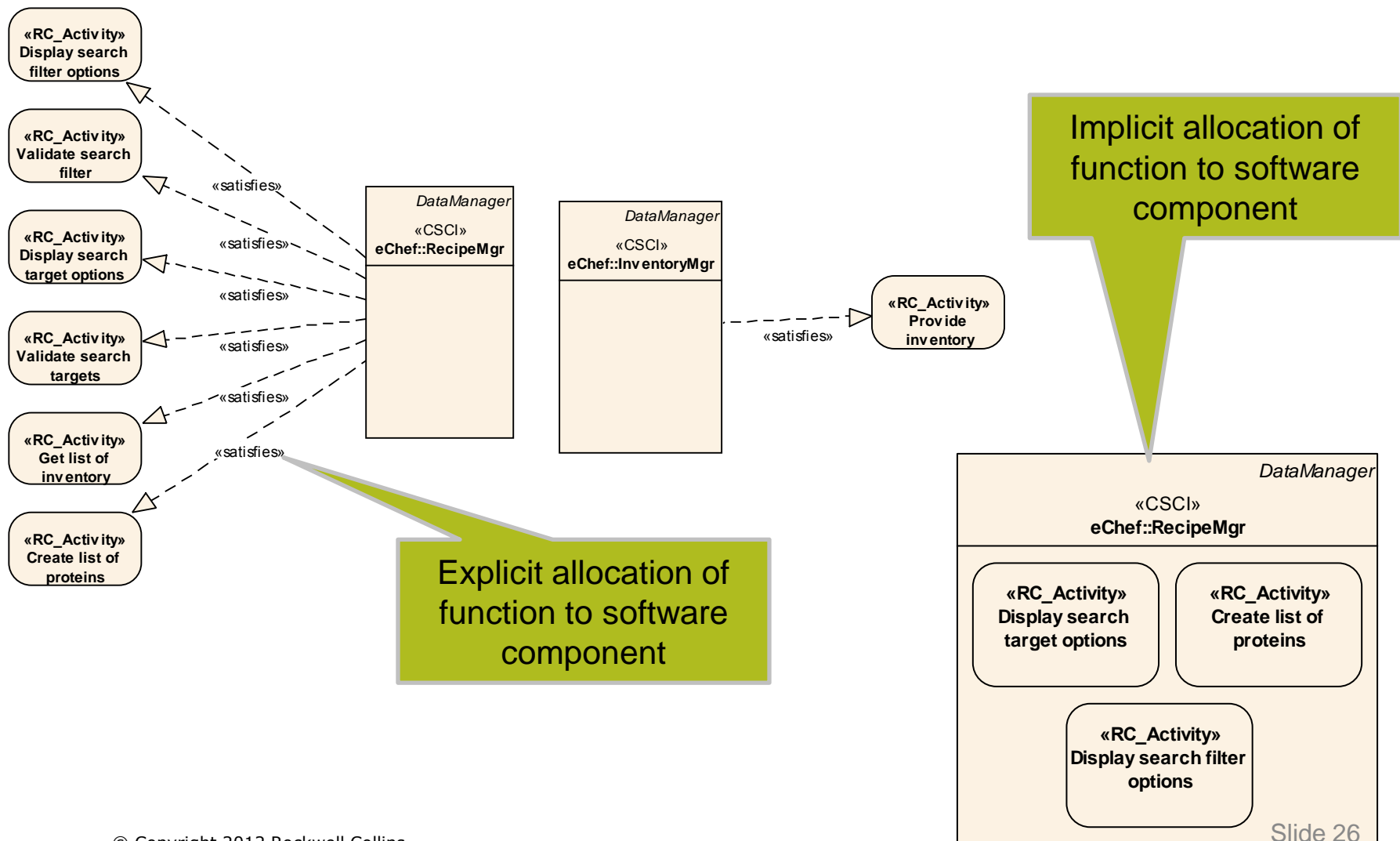


Implicit Hierarchy
- Showing with
superposition

Ports and Flows –
where the data words
flow

Software View – Allocation Perspective

Logical Elements to Software Elements

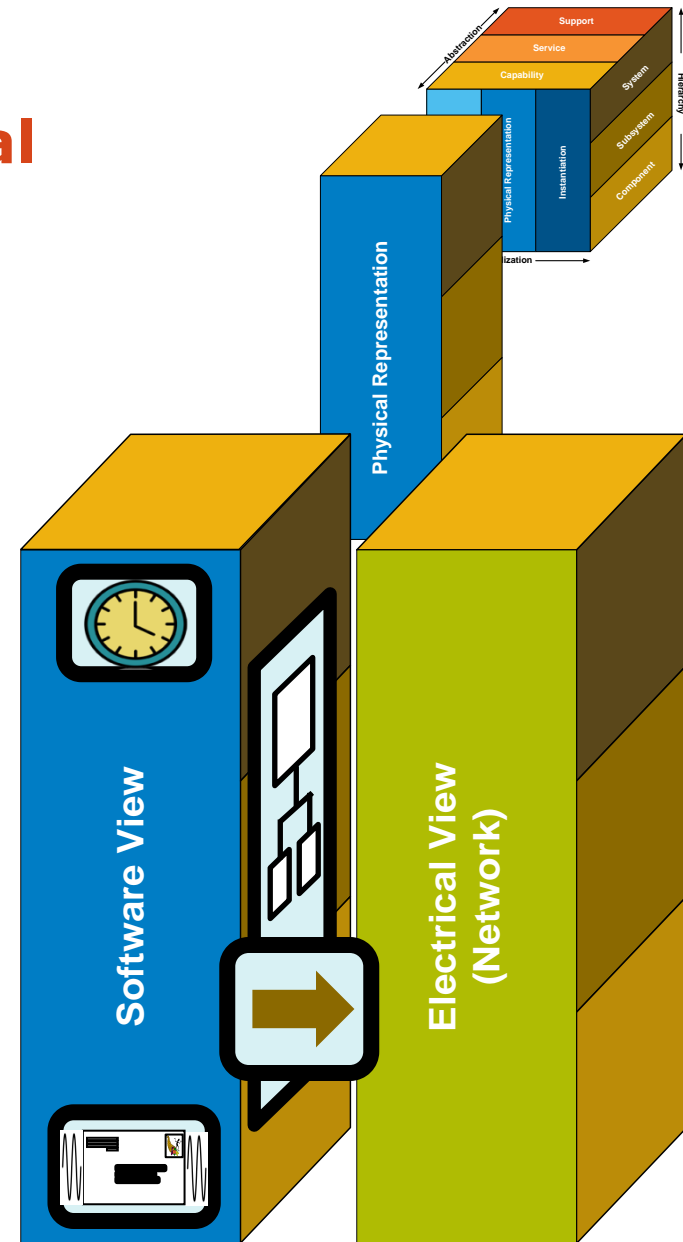


Software Allocation to Electrical

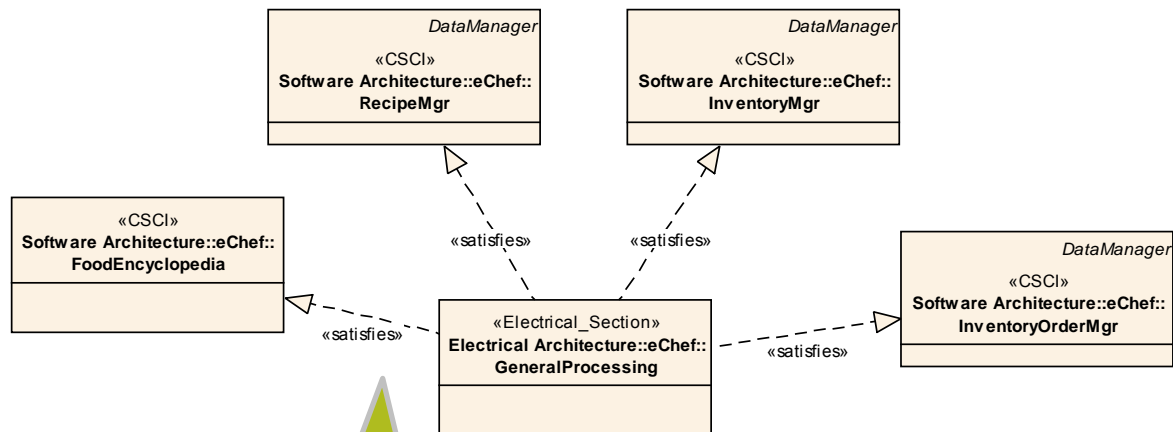
- Software View



Allocation (or Deployment) Perspective
(to Electrical)

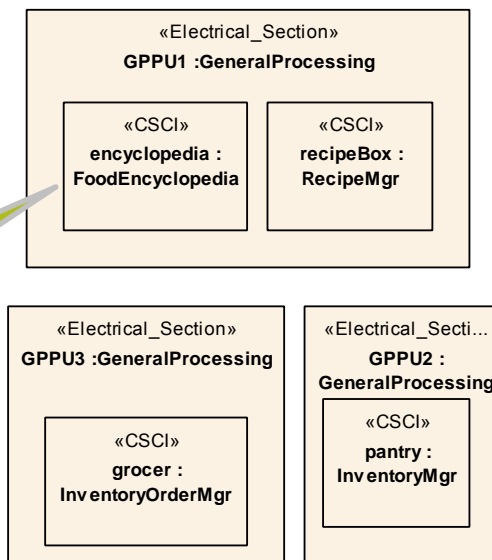


Software View – Allocation (or Deployment) Perspective



Explicit allocation/
deployment of
software components
onto processor.

Implicit allocation/
deployment of
software components
onto processor.



Physical Representation – Electrical View

- Electrical View (Network)



Hierarchical Perspective



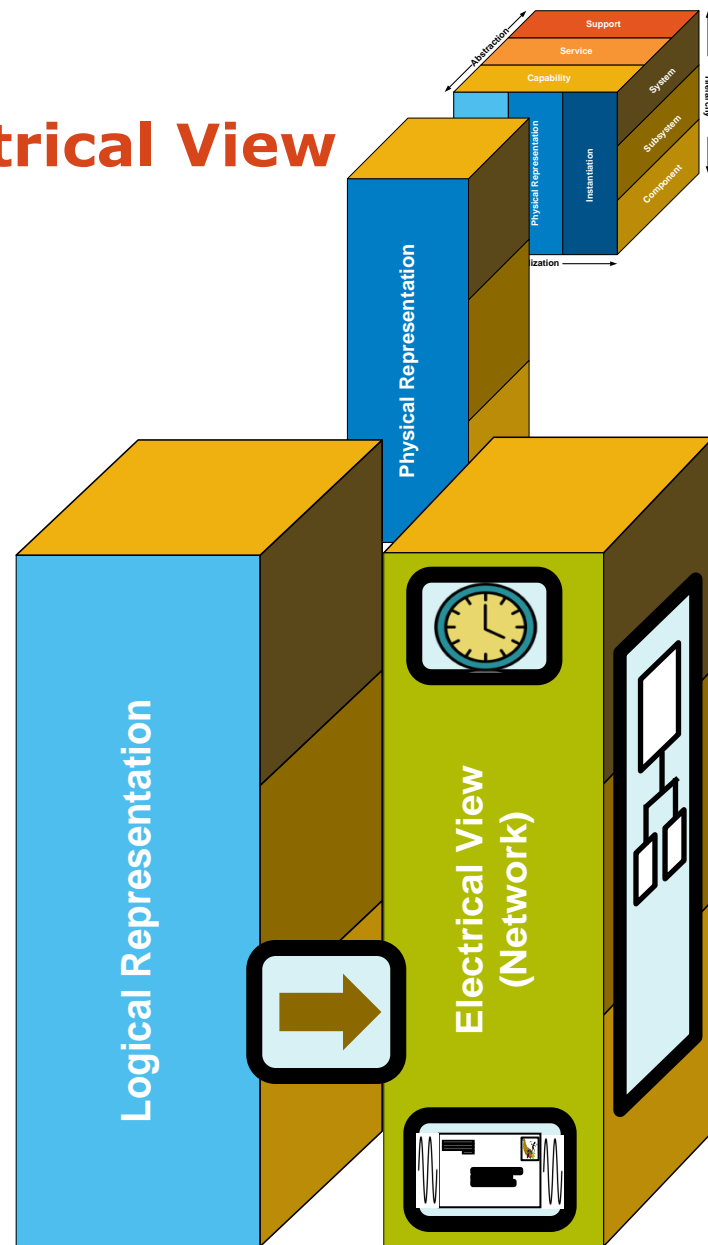
Sequential Perspective



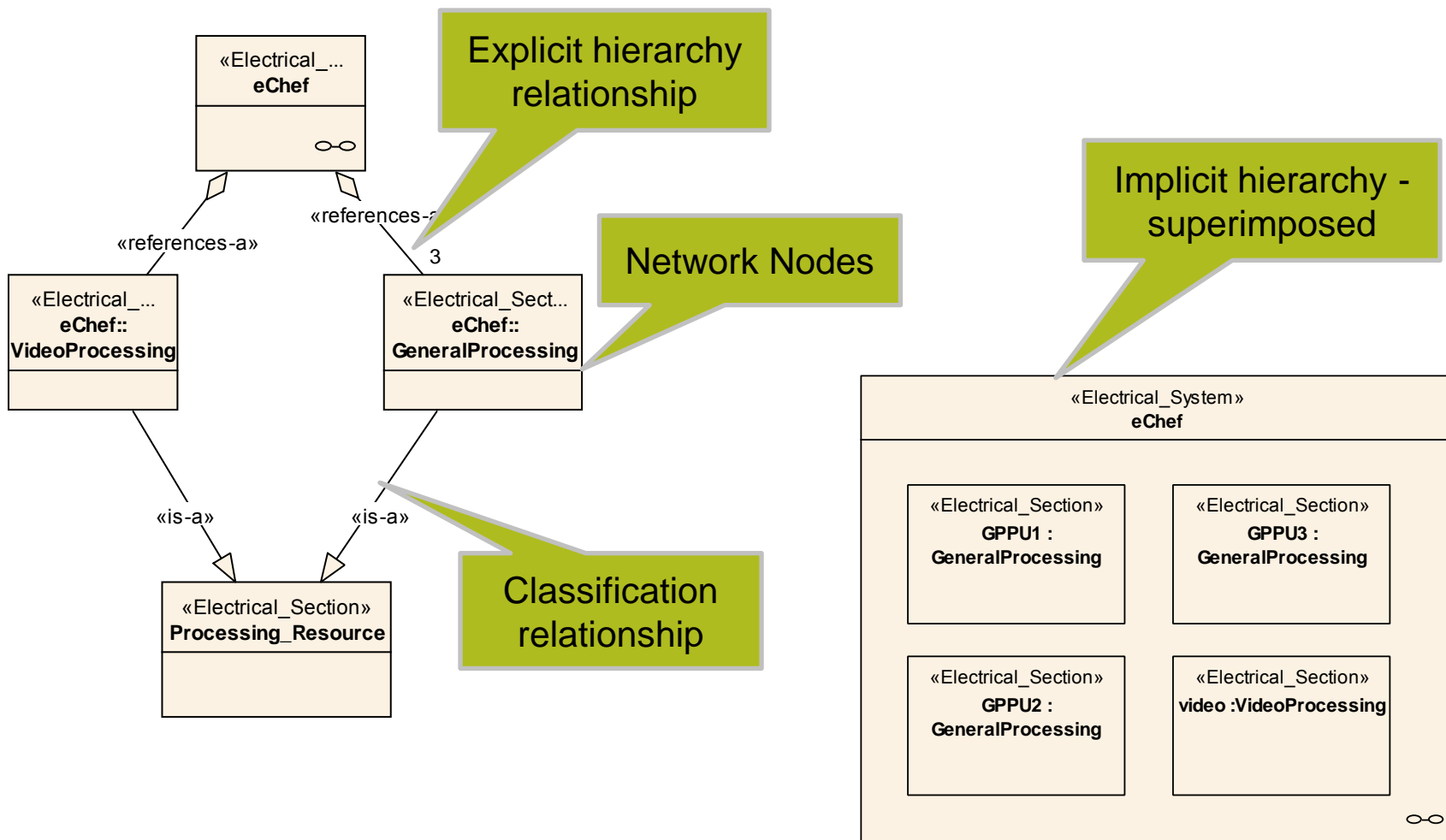
Transactional Perspective (Datalinks and Signals)



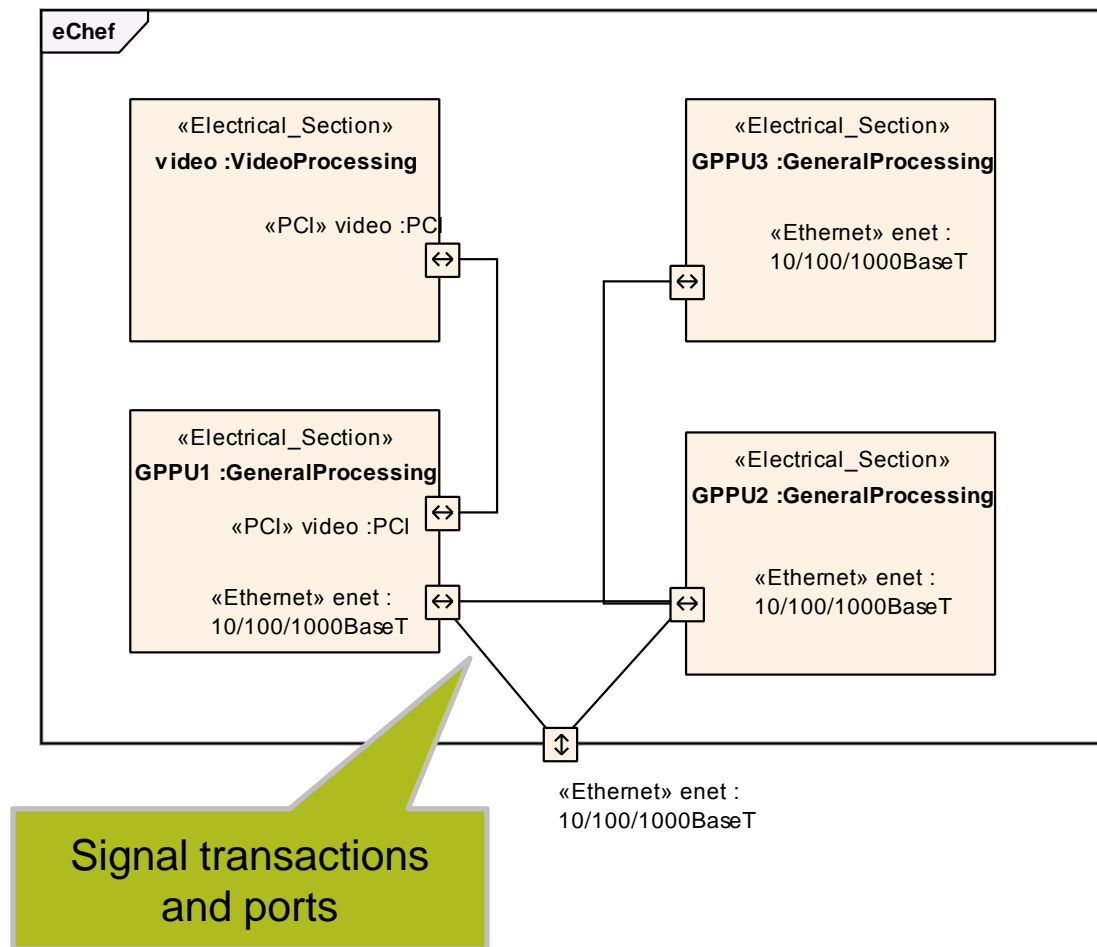
Allocation Perspective (from Logical)
Allocation (or Deployment) Perspective (from Software)



Electrical View (Network) – Hierarchical Perspective



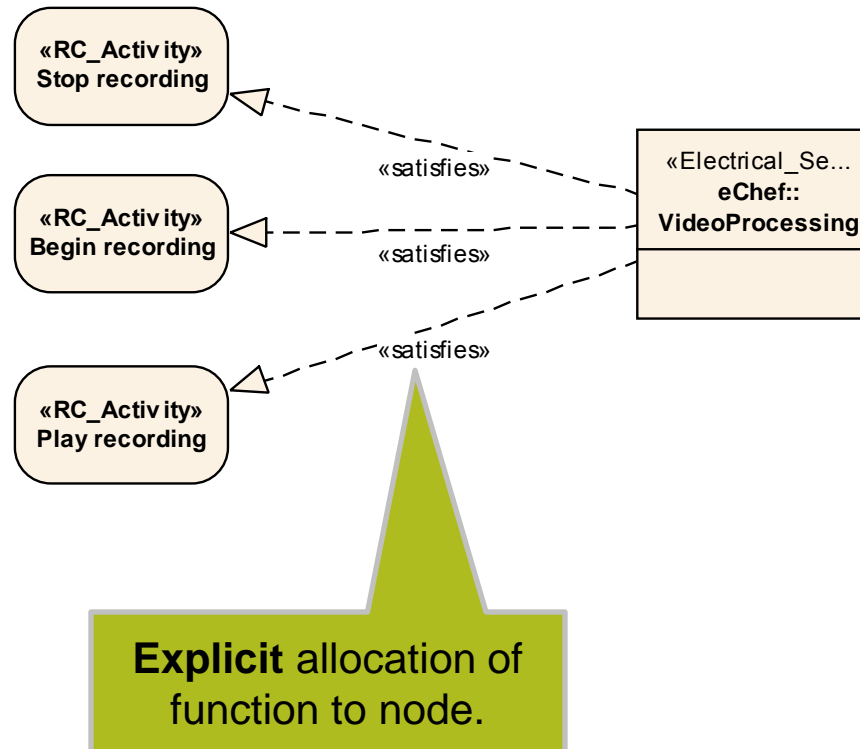
Electrical View (Network)– Transactional Perspective



Electrical View (Network) – Allocation Perspective

Logical to Electrical Node

- Only do this for logical behavior that does not have a software component trace

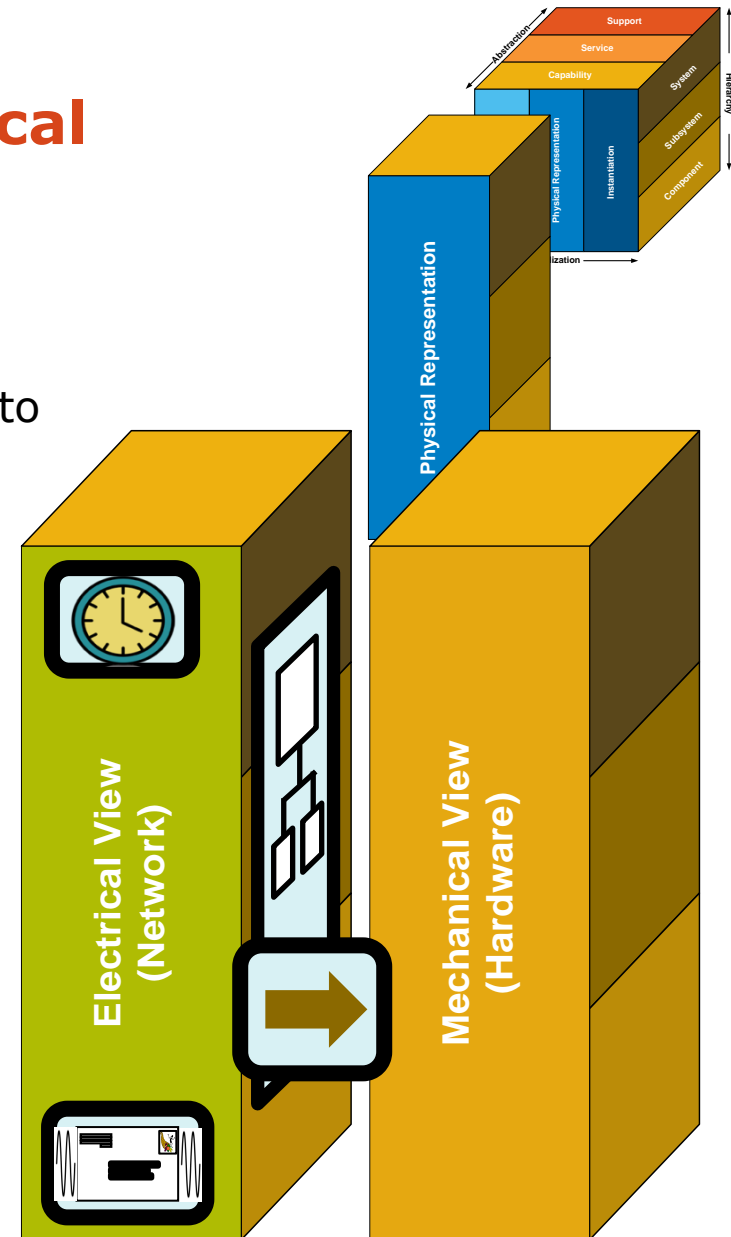


Electrical Allocation to Mechanical (Hardware)

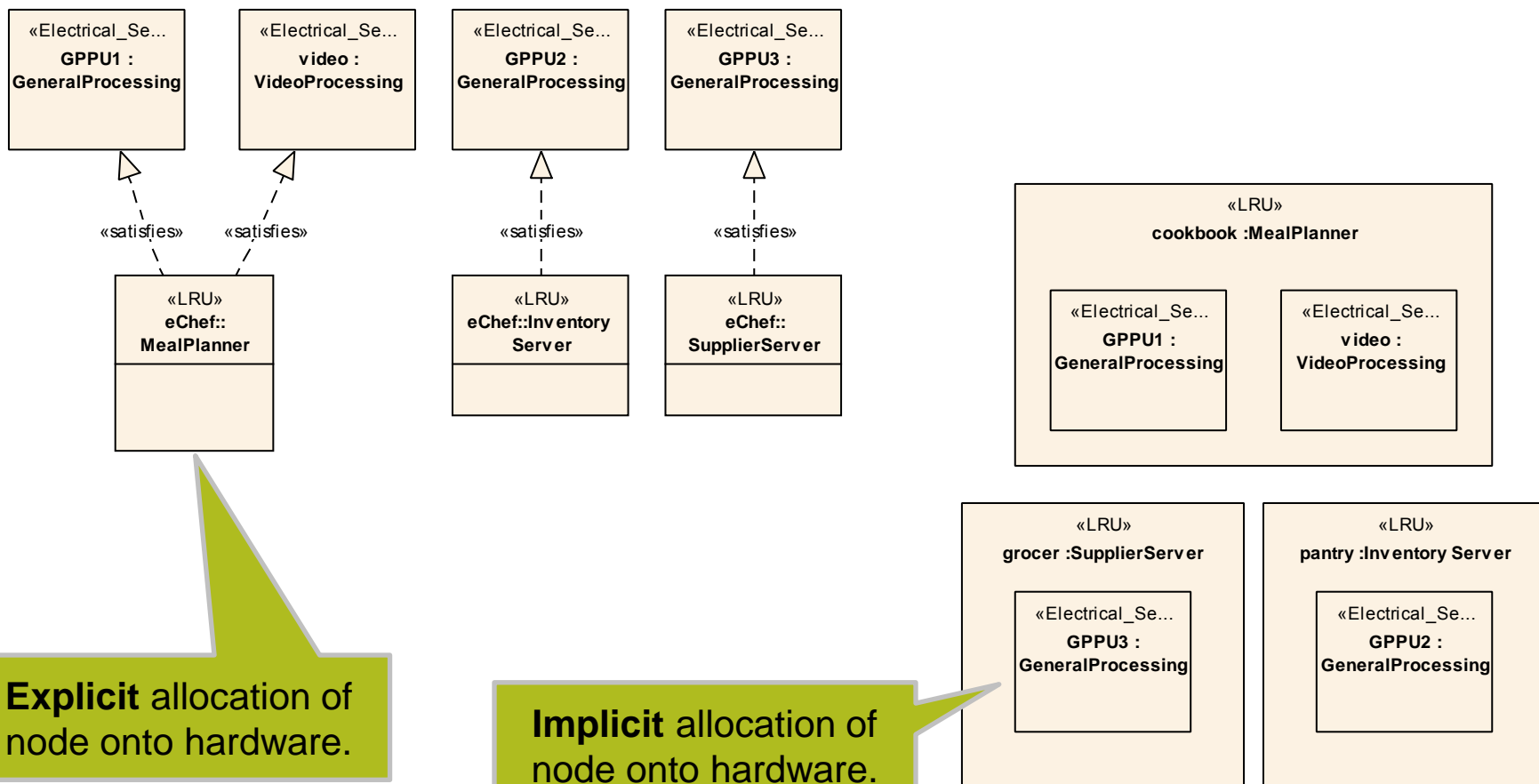
- Electrical View



Allocation (or Deployment) Perspective (to Hardware)



Electrical View (Network) – Allocation (Deployment) Perspective – Network Node to Mechanical Hardware



Physical Representation – Mechanical View



Mechanical (Hardware) View

Hierarchical Perspective



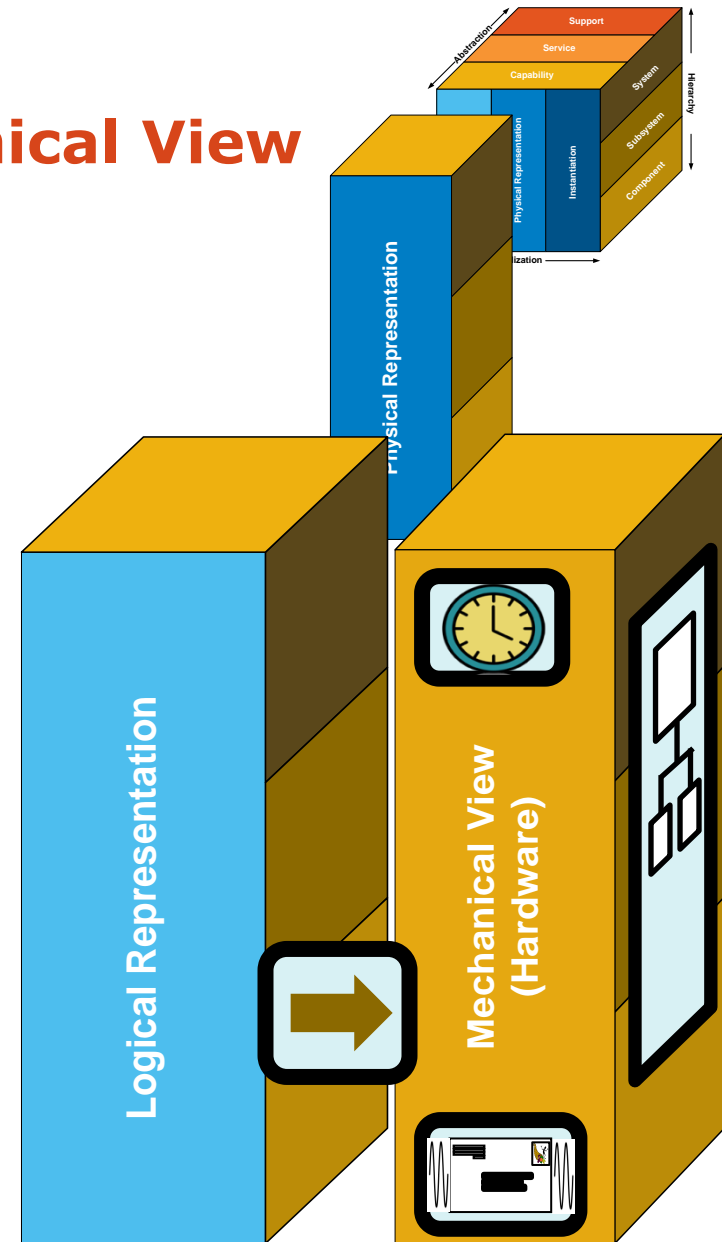
Sequential Perspective (uncommon in avionics)



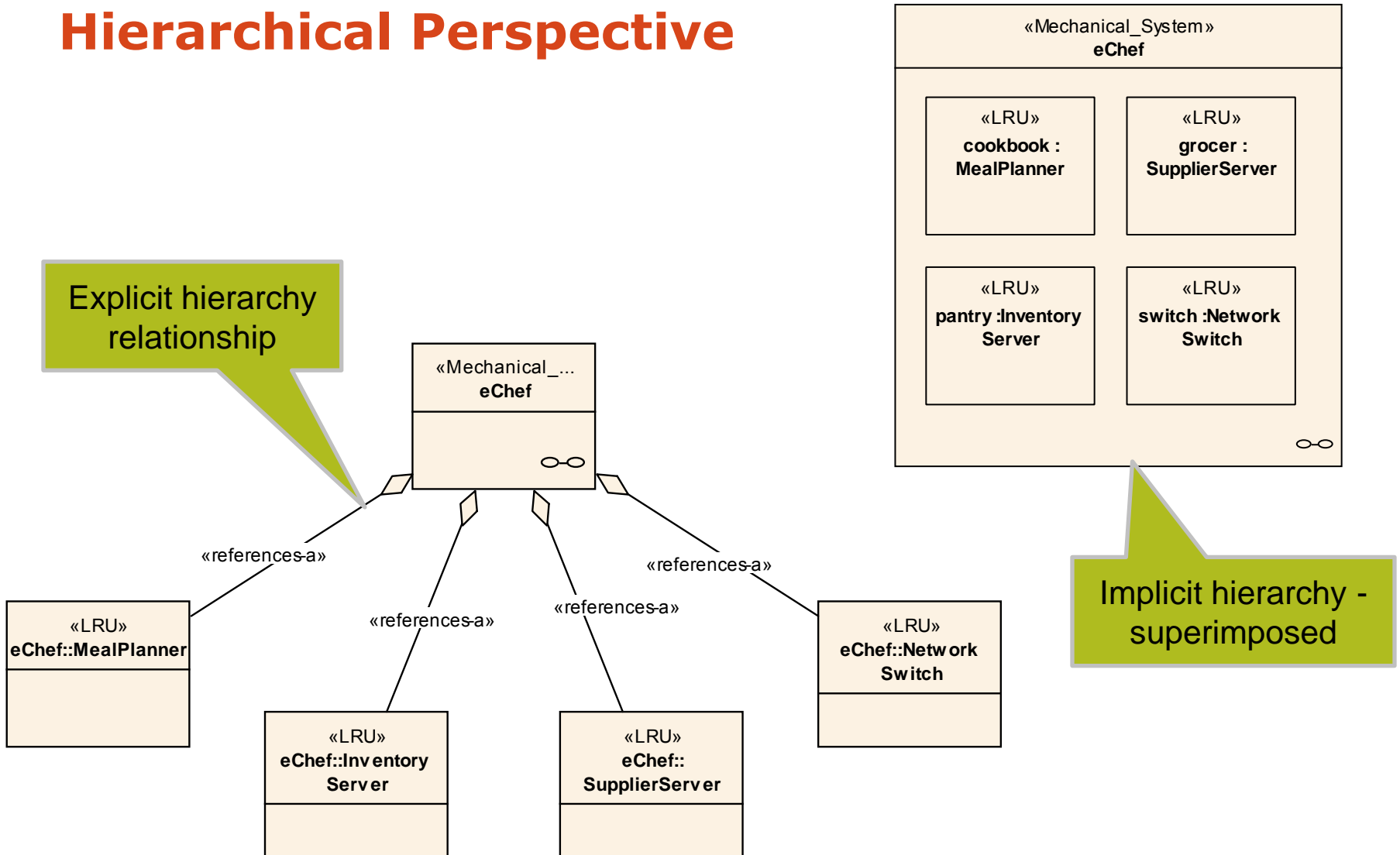
Transactional Perspective (connections)



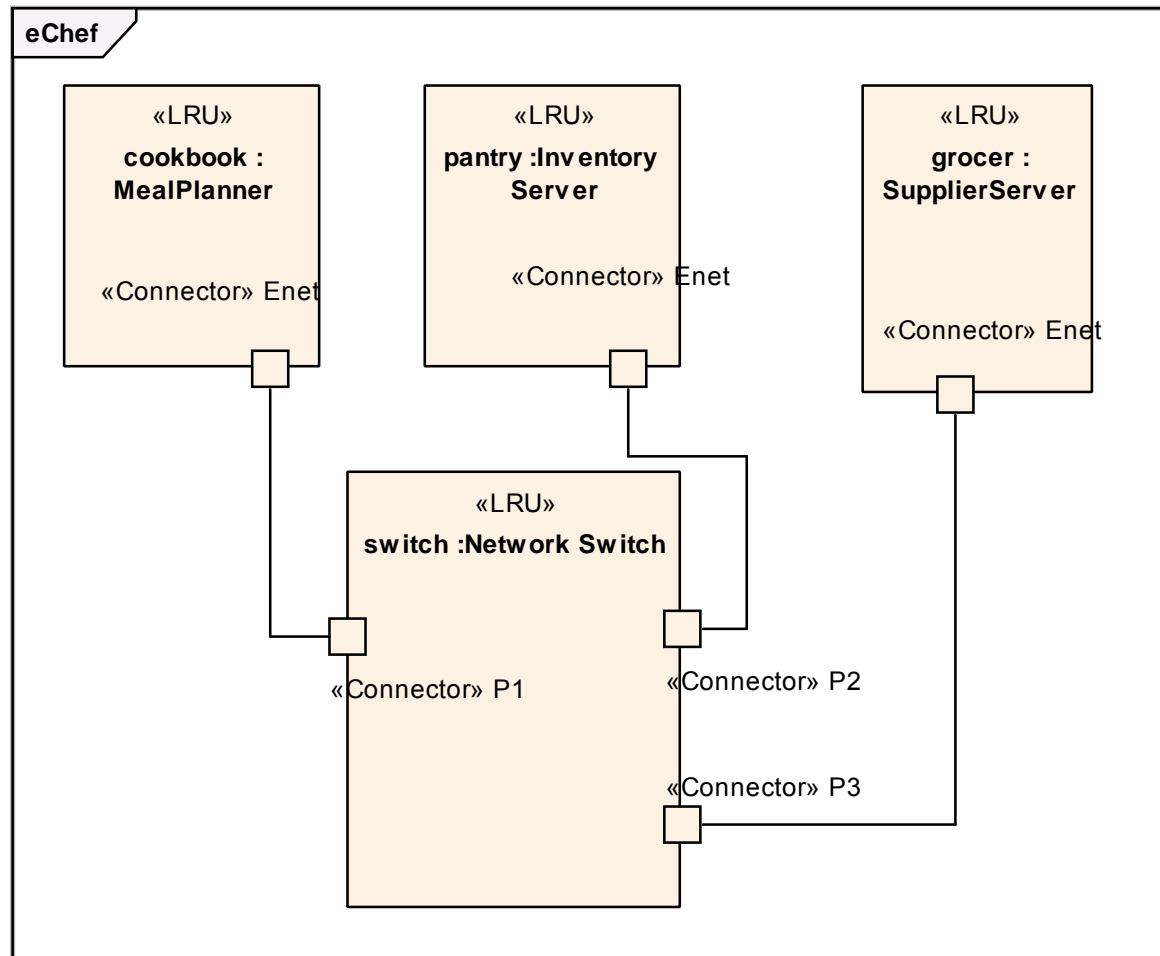
Allocation Perspective (from Logical)
Allocation (or Deployment) Perspective (from Network)



Mechanical View (Hardware) – Hierarchical Perspective

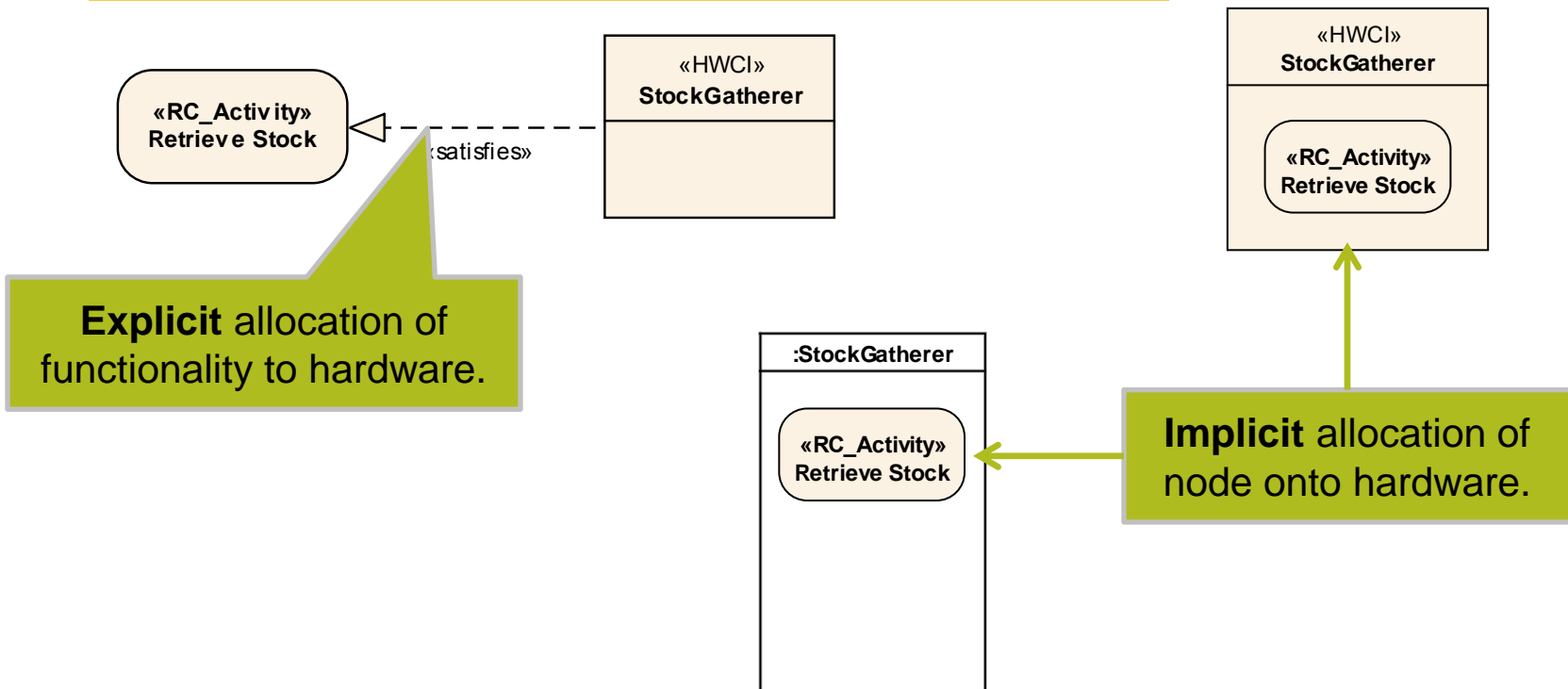


Mechanical View – Transactional Perspective



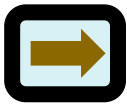
Mechanical Views – Allocation Perspective – Logical to Mechanical (Hardware)

Allocate logical elements that do not have a software or electrical component allocation

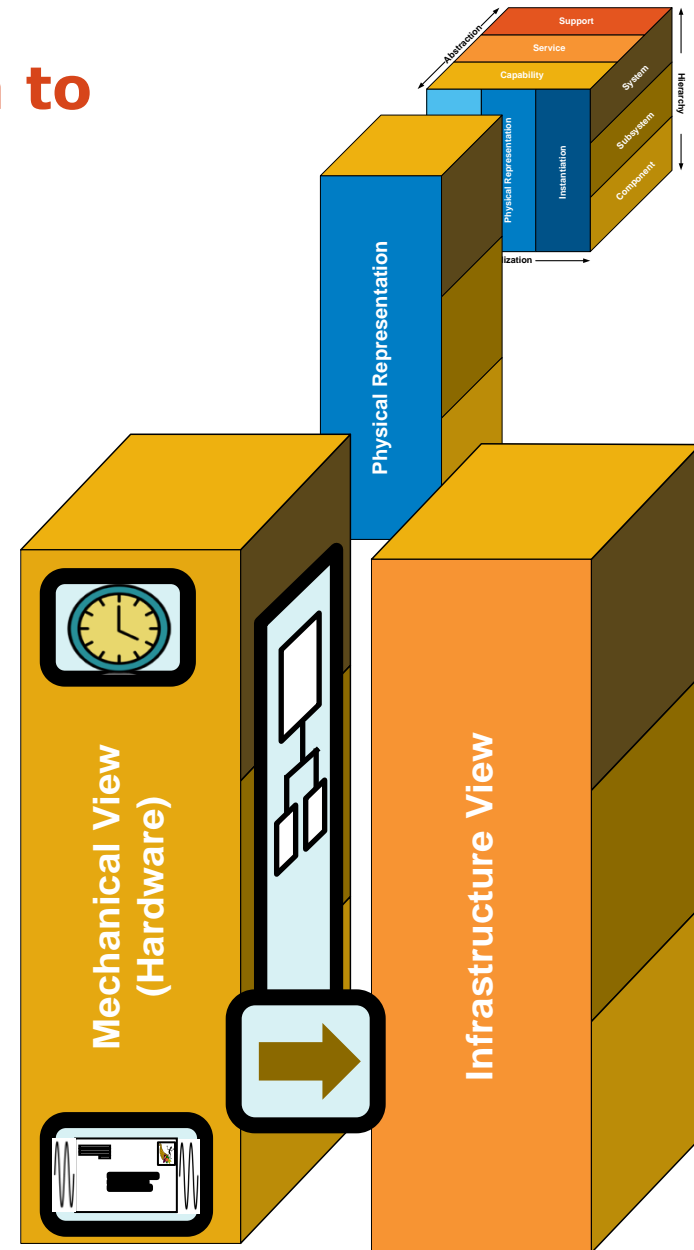


Mechanical (Hardware) Allocation to Infrastructure

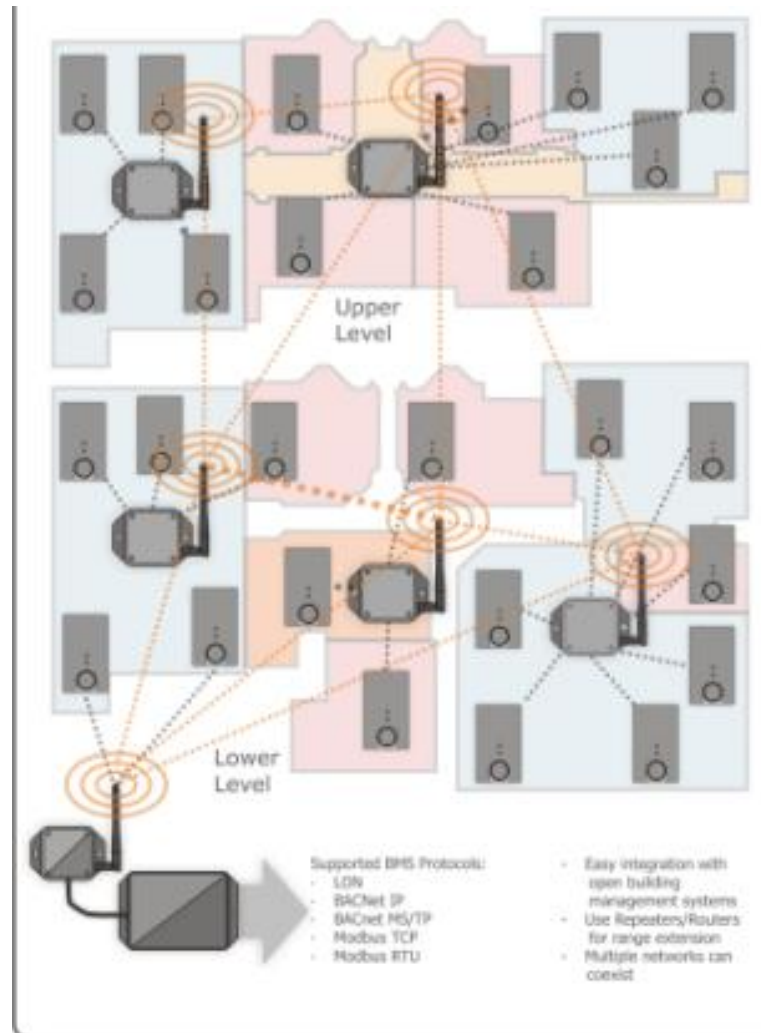
- Mechanical (Hardware) View



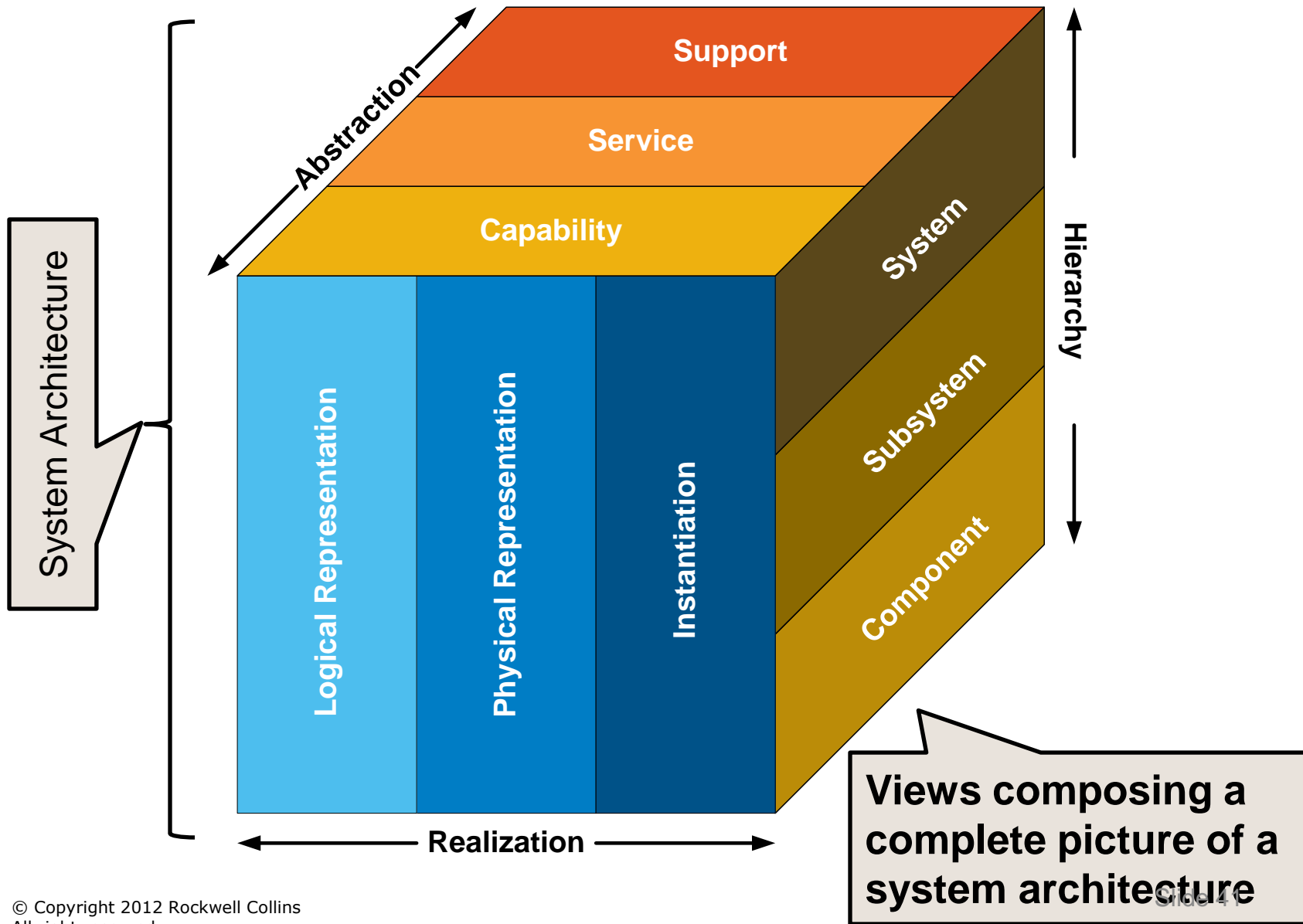
Allocation (or Deployment) Perspective (to Infrastructure)



Mechanical View – Allocation (Deployment) Perspective Mechanical (Hardware) to Infrastructure

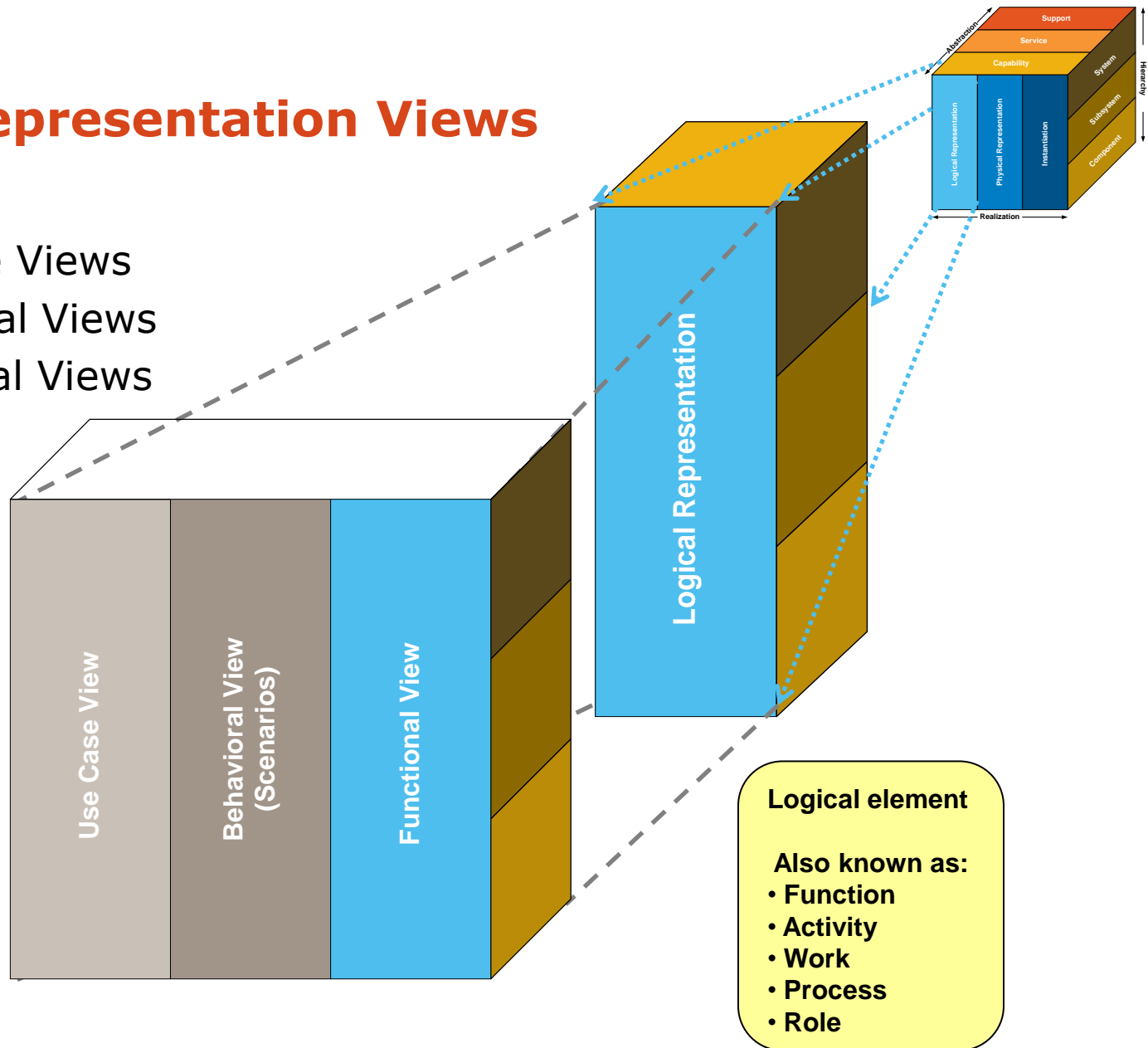


Integrated Modular Solution Architecture (IMSA)



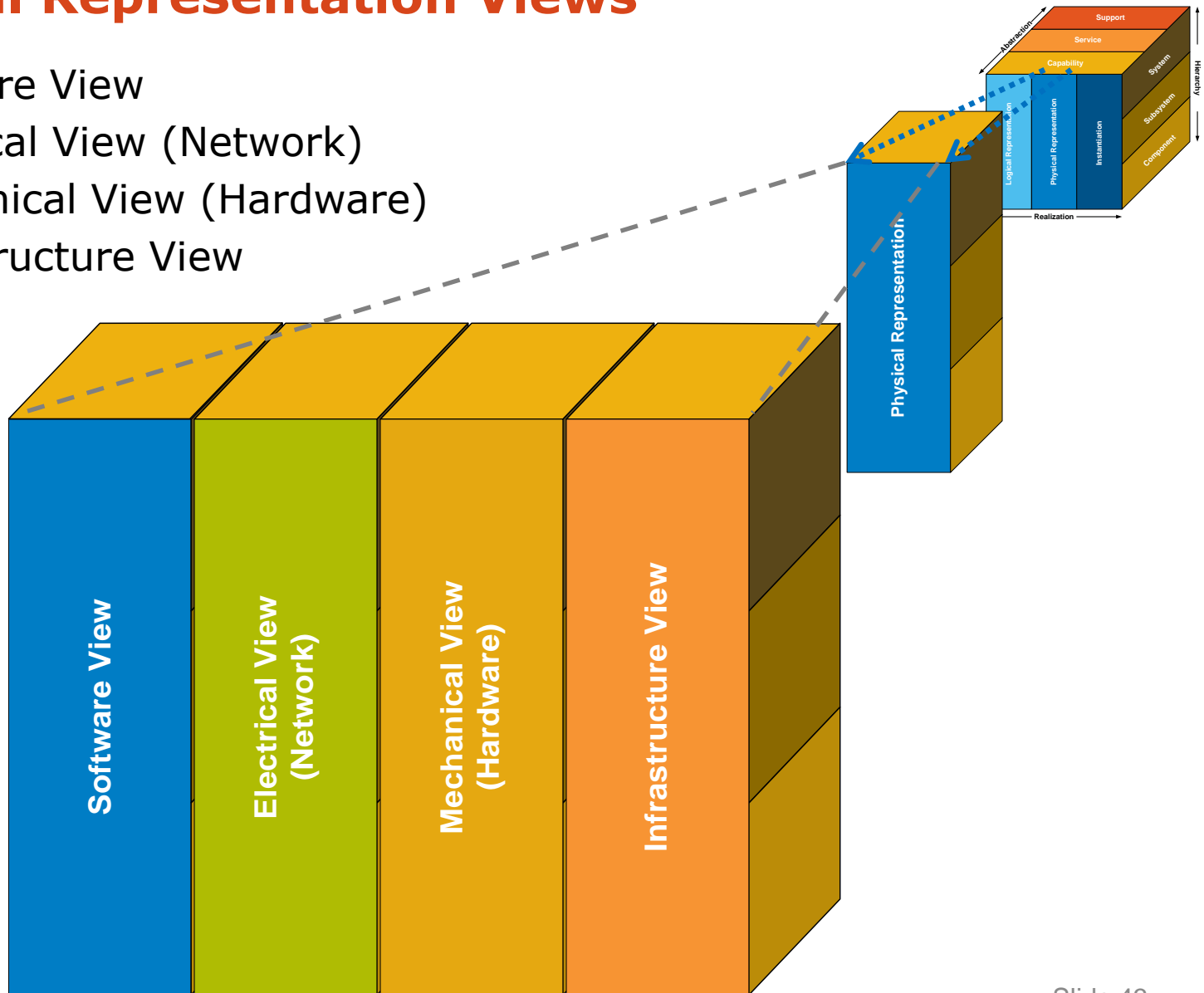
Logical Representation Views

- Use Case Views
- Behavioral Views
- Functional Views



Physical Representation Views

- Software View
- Electrical View (Network)
- Mechanical View (Hardware)
- Infrastructure View



Architectural Perspectives



- Hierarchical Perspective
 - Focused on organization of things



- Sequential Perspective
 - Focused on the timing of things



- Transactional Perspective
 - Focused on the transfer of interface objects



- Allocation (Deployment) Perspective
 - Focused on the realization of things – where does it live?

Presenter Information

- **Raymond Jorgensen**
- Systems Engineering Knowledge Architect
- Rockwell Collins, Inc
- 400 Collins Rd NE, MS 108-265
- Cedar Rapids, IA 52498
- rwjorgen@rockwellcollins.com
- 319-295-2615