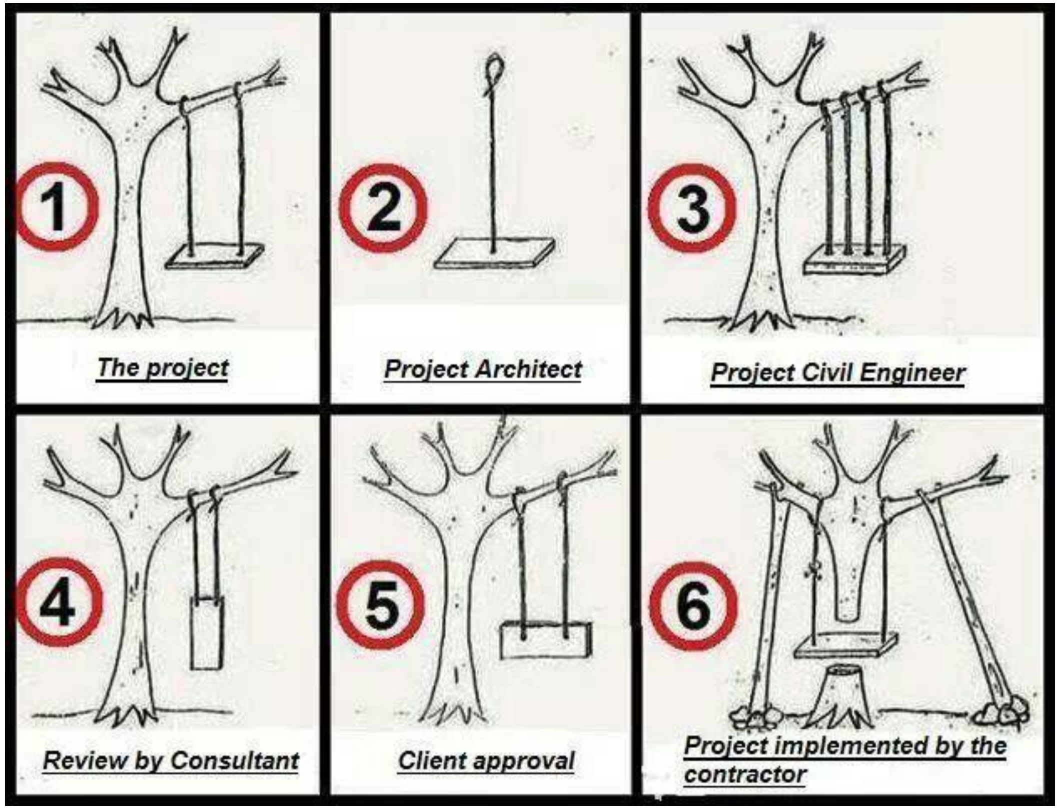

1
The project
2
Project Architect
3
Project Civil Engineer
4
Review by Consultant
5
Client approval
6
Project implemented by the contractor

- What is the distinction between - between architecting and engineering?
- Engineering deals almost entirely with measurables using analytic tools derived from mathematics and the hard sciences; that is, engineering is a **deductive** process.

- Architecting deals largely with unmeasurables using nonquantitative tools and guidelines based on practical lessons learned; that is, architecting is an **inductive** process

Table P.1 Characteristics of the Roles on the Architecting and Engineering Continuum

Characteristic	Architecting	Architecting and Engineering	Engineering
Situation/goals	Ill-structured	Constrained	Understood
	Satisfaction	Compliance	Optimization
Methods	Heuristics	⟵⟶	Equations
	Synthesis	⟵⟶	Analysis
	Art and science	Art and science	Science and art
Interfaces	Focus on "mis-fits"	Critical	Completeness
System integrity maintained through	"Single mind"	Clear objectives	Disciplined methodology and process
Management issues	Working for client	Working with Client	Working for builder
	Conceptualization and certification	Whole waterfall	Meeting project requirements
	Confidentiality	Conflict of interest	Profit versus cost

The art of systems architecture – Mark Maier

Four Architecting Methodologies

- Normative (solution based)

    Examples: building codes and communications standards or protocol standards

- Rational (method based)

    Examples: systems analysis and engineering

- Participative (stakeholder based)

    Examples: concurrent engineering and brainstorming

- Heuristic (lessons learned)

    Examples: Simplify. Simplify. Simplify. and SCOPE!

Selected Artifacts Created during the Architecture Process

- Define a **consistent logical architecture**—capture the logical sequencing and interaction of system functions or logical elements.
- **Partition system requirements** and allocate them to system elements and subsystems with associated performance requirements—evaluate off-the-shelf solutions that already exist.
- **Evaluate alternative design solutions** using trade studies.
- **Identify interfaces and interactions** between system elements (including human elements of the system) and with external and enabling systems.
- Define the **system integration strategy** and plan (to include human system integration).
- Document and maintain the architectural design and relevant decisions made to reach agreement on the baseline design.
- Establish and maintain the traceability between requirements and system elements.
- Define verification and validation criteria for the system elements.

Simplified architecture/definition

Entity Component System consists of three primary items:

- Components
  - A component simply holds a piece of data and does not contain any game logic. Your typical component will have fields for primitive values and data objects.
- Entities
  - An entity is a collection of components.
- Systems
  - A system is typically an implementation that iteratively operates on a group of entities that share a specific set of components.

Abstraction

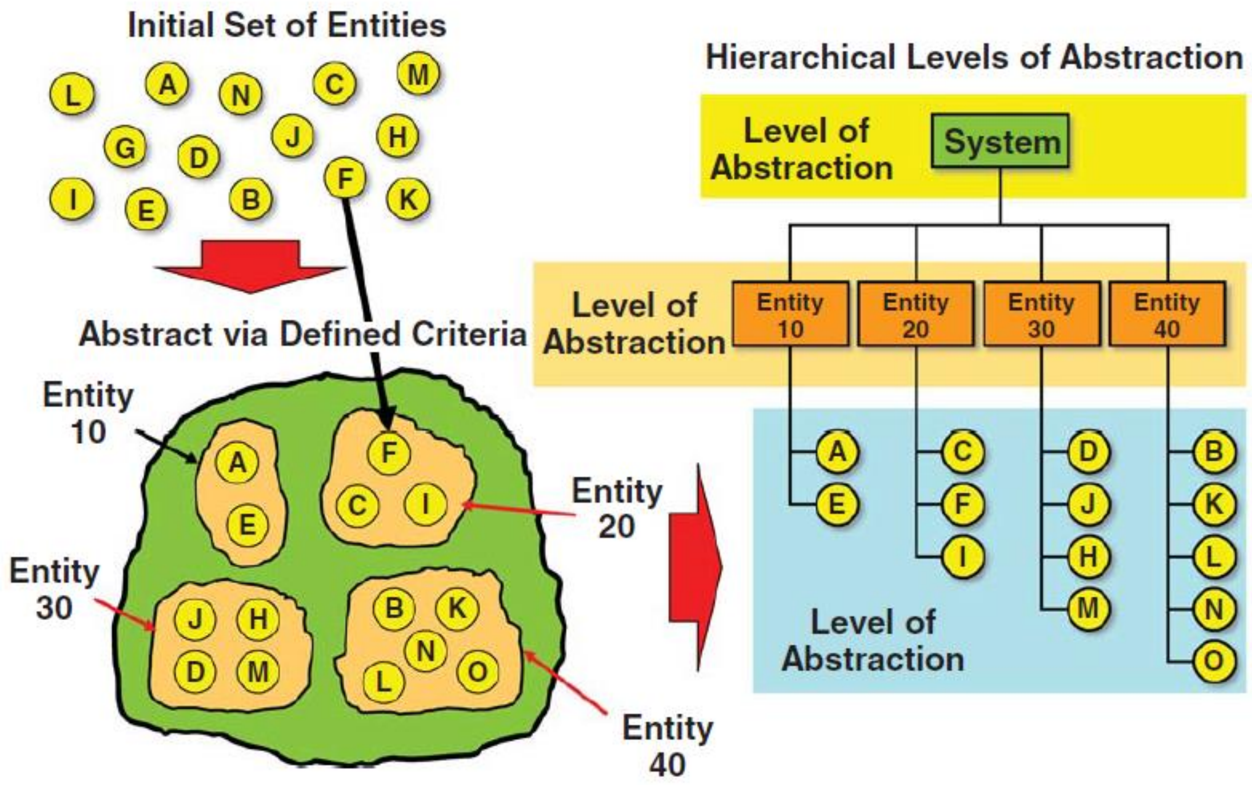


Figure 8.3 Abstracting Entities into levels of Abstraction

System Element Architecture

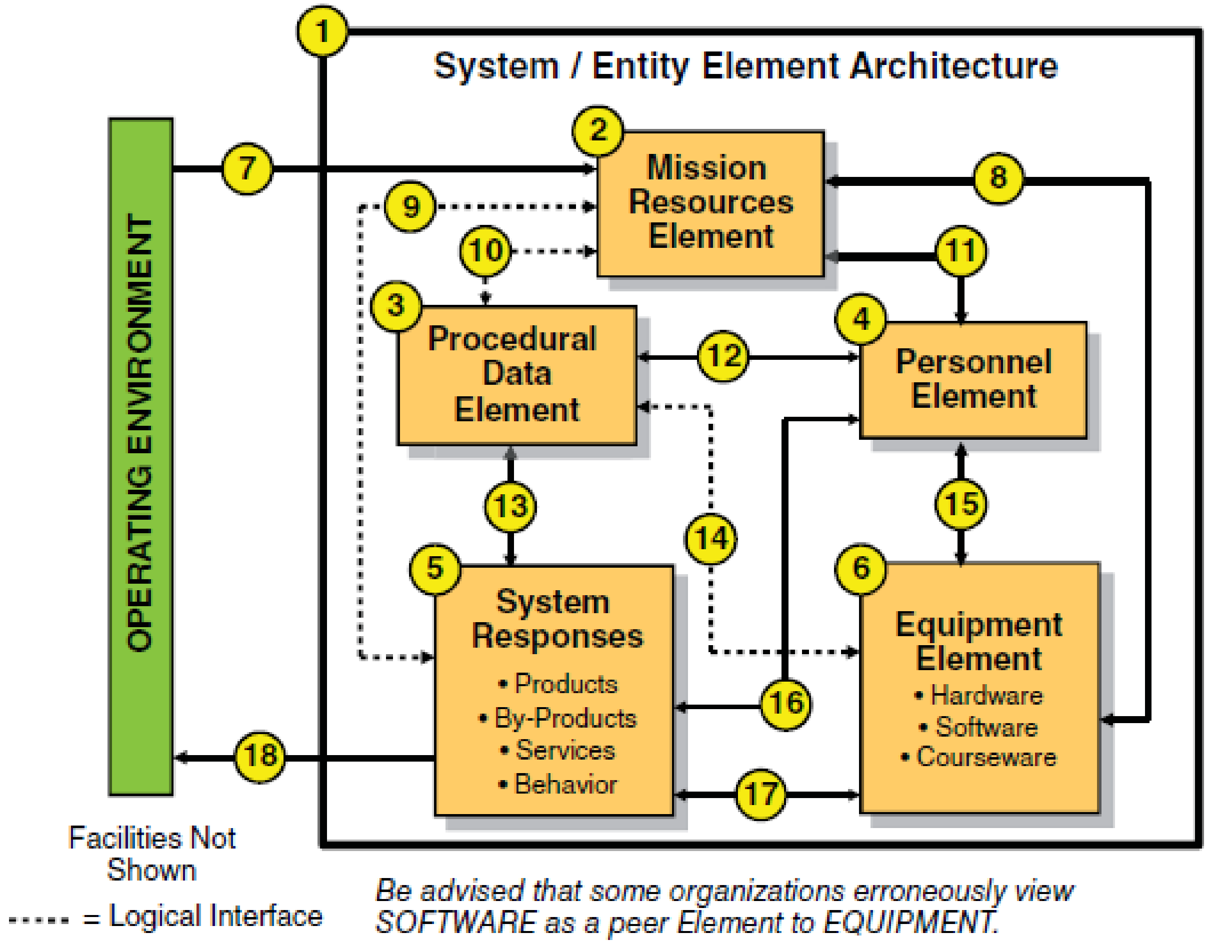


Figure 8.13 System Element Architecture (SEA) Construct

System level abstraction

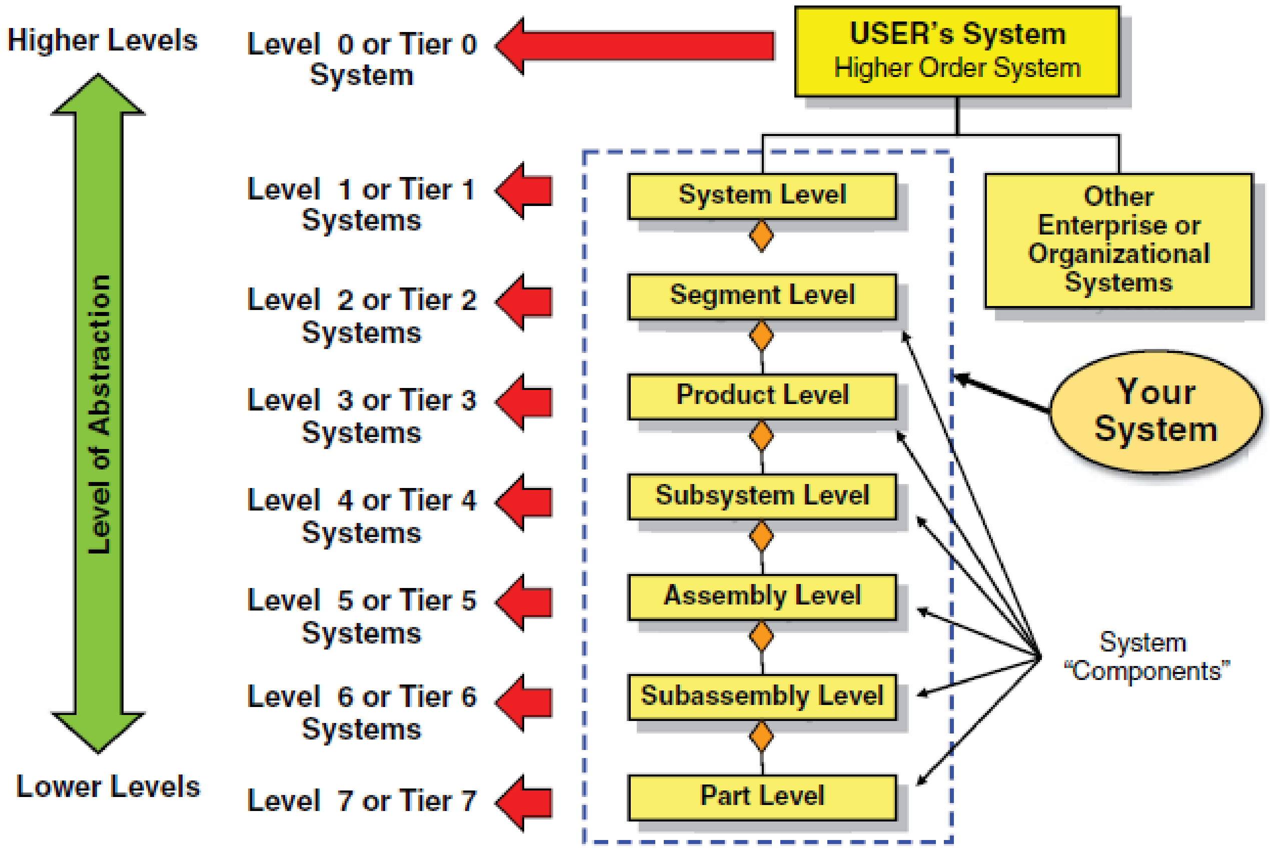
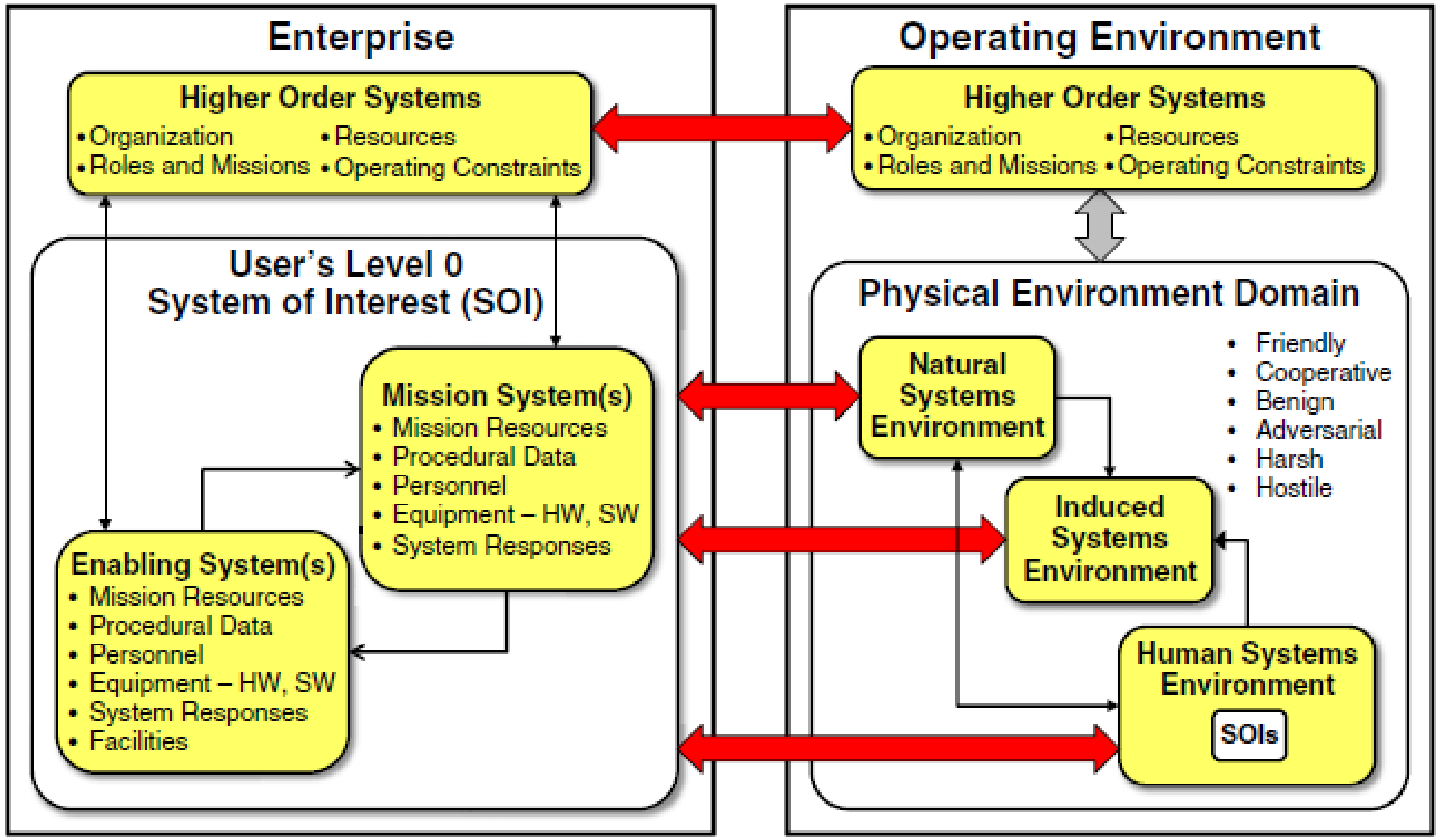
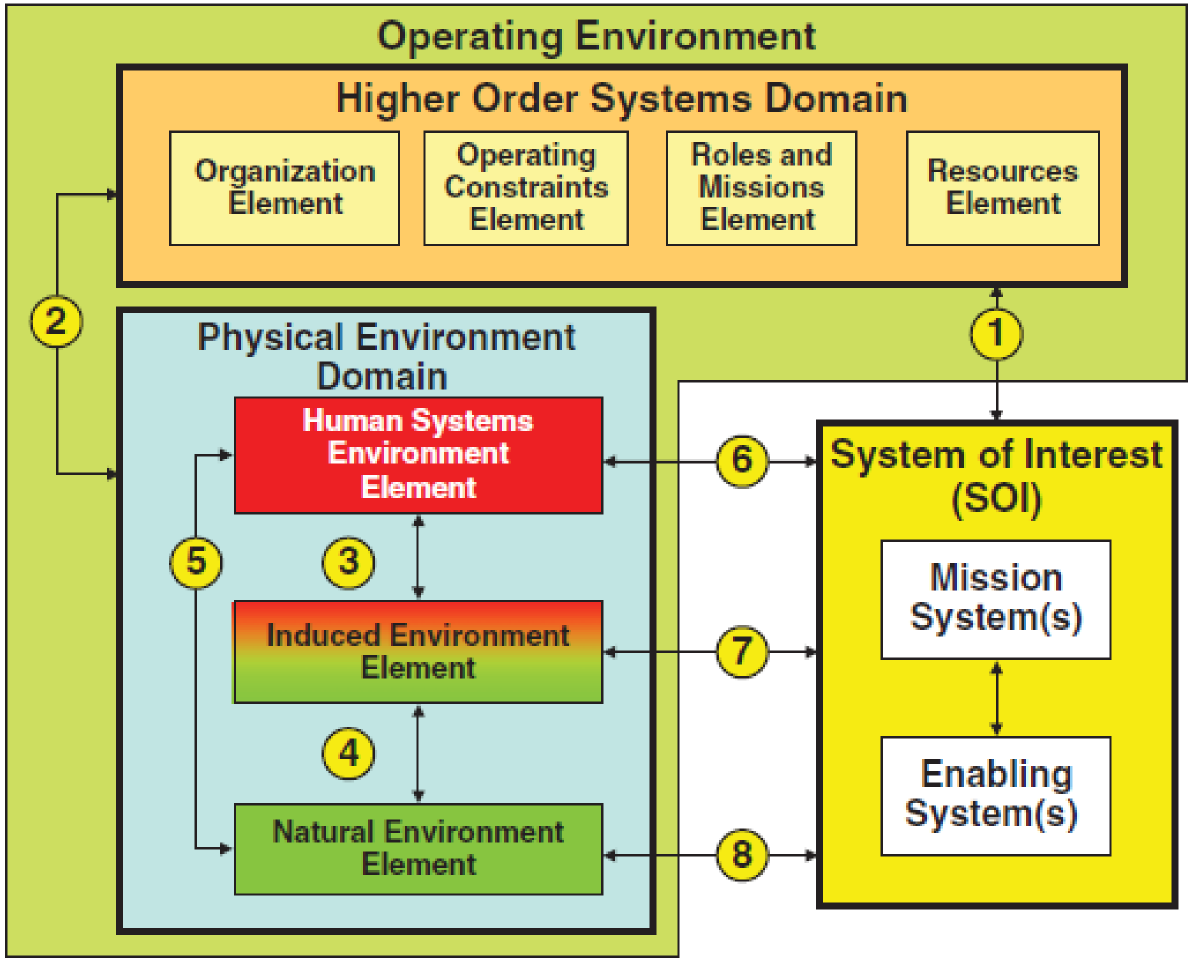


Figure 8.4 System Levels of Abstraction and Semantics Frame of Reference

System/Entity Architecture



SOI's Operating Environment (OE) Architecture



Levels of abstraction

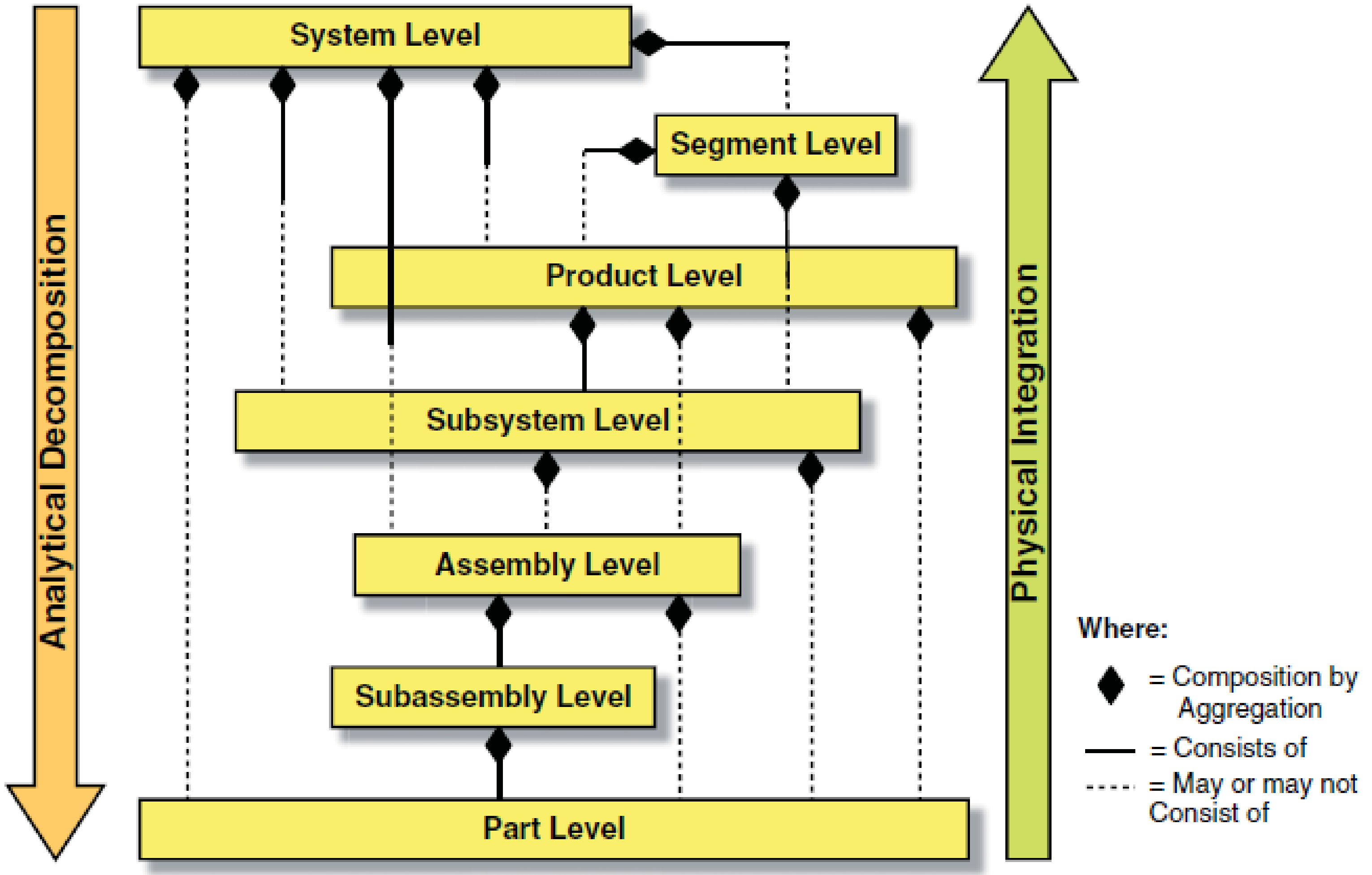


Figure 8.7 System Analytical Decomposition into Levels of Abstraction versus Physical

functional analysis and decomposition - list of tools and diagrams

- IDEF0 diagram
- Functional flow block diagram (FFBD)
- N^2 diagrams - diagram represents the logical data flow for a system or system segment.
- Timeline analysis
- Tree diagrams
- SysML (such as activity diagrams, sequence diagrams)

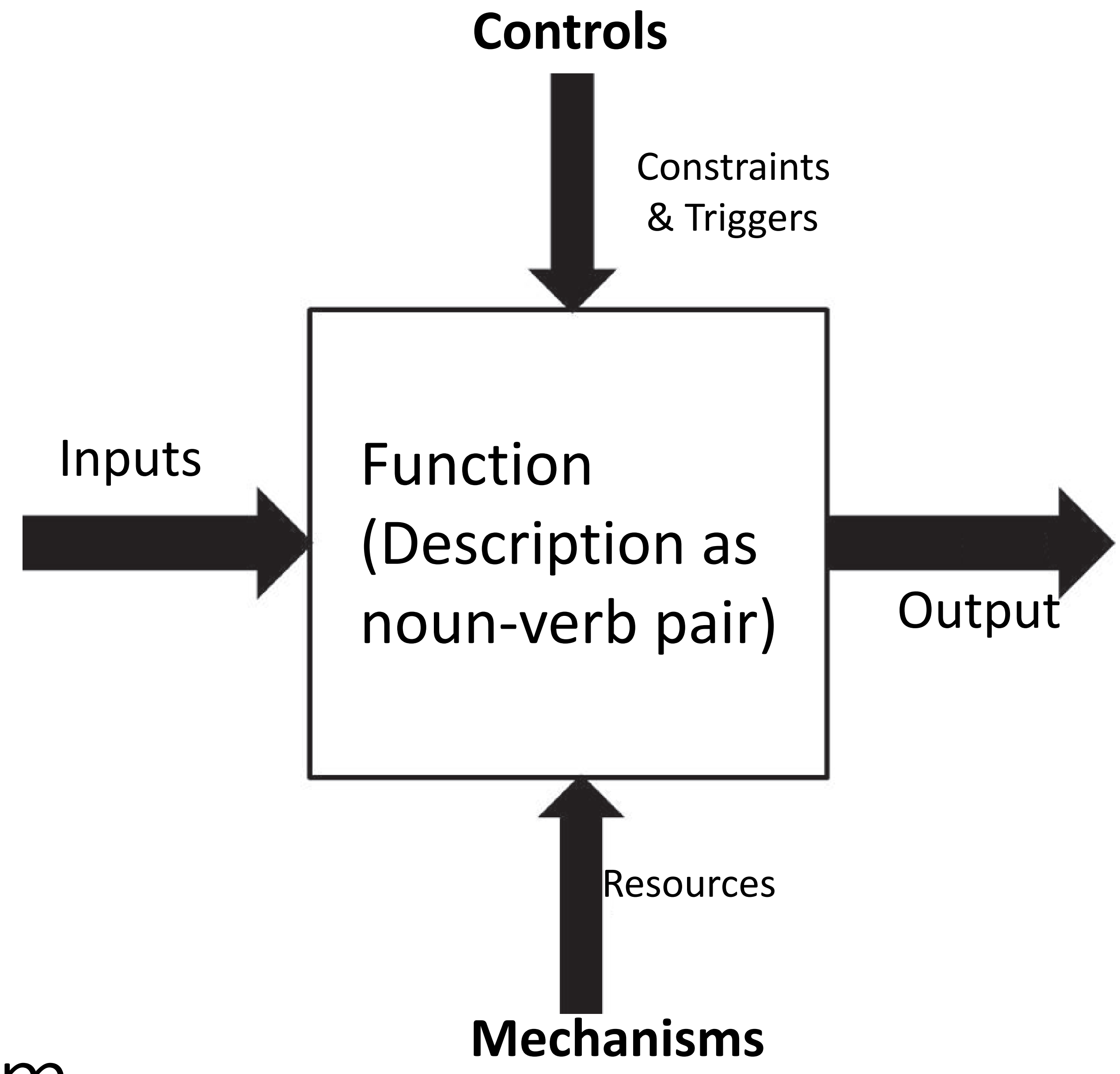
Example of functional specifications

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements

non-functional requirements are

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

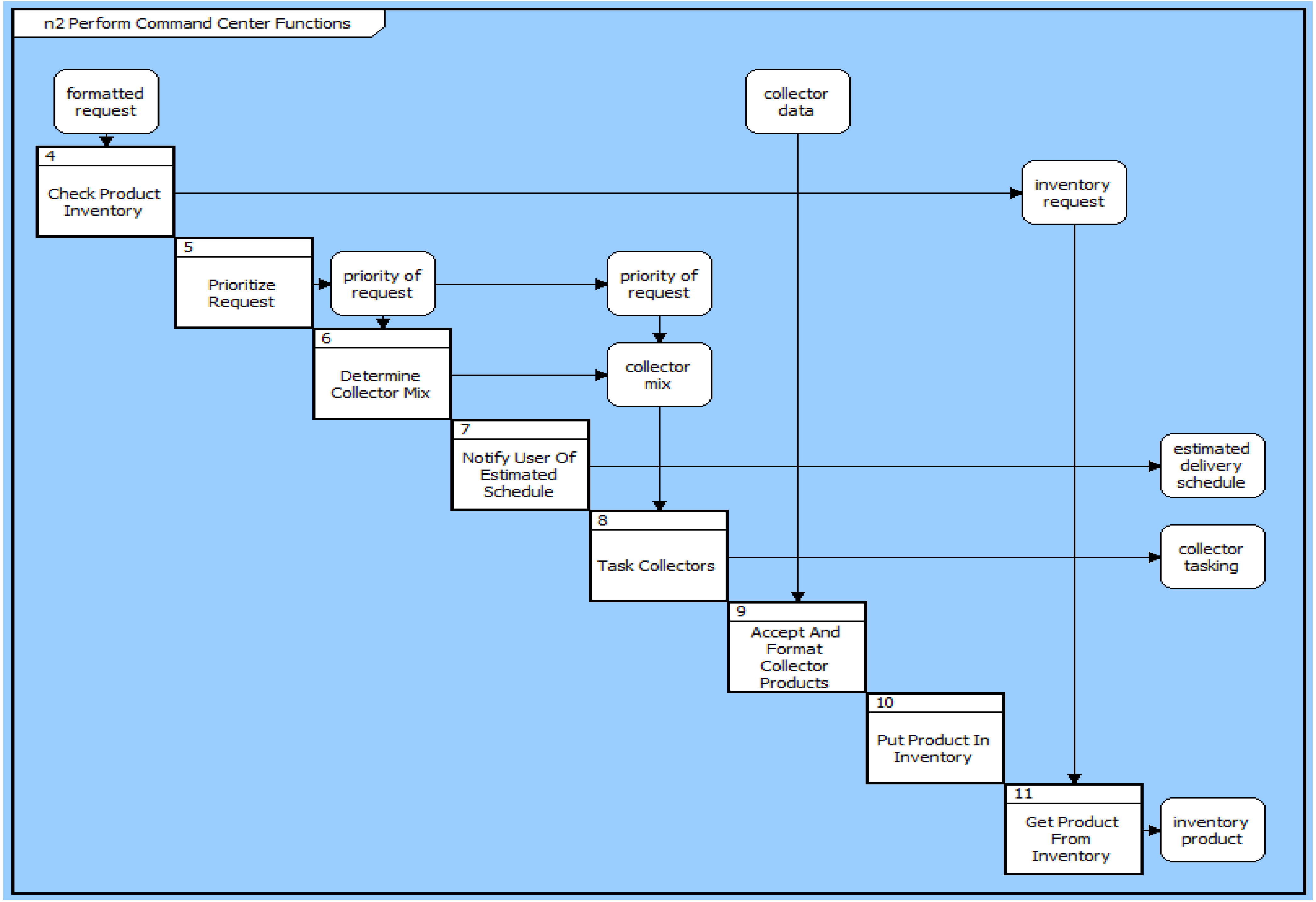
IDEF0 diagram



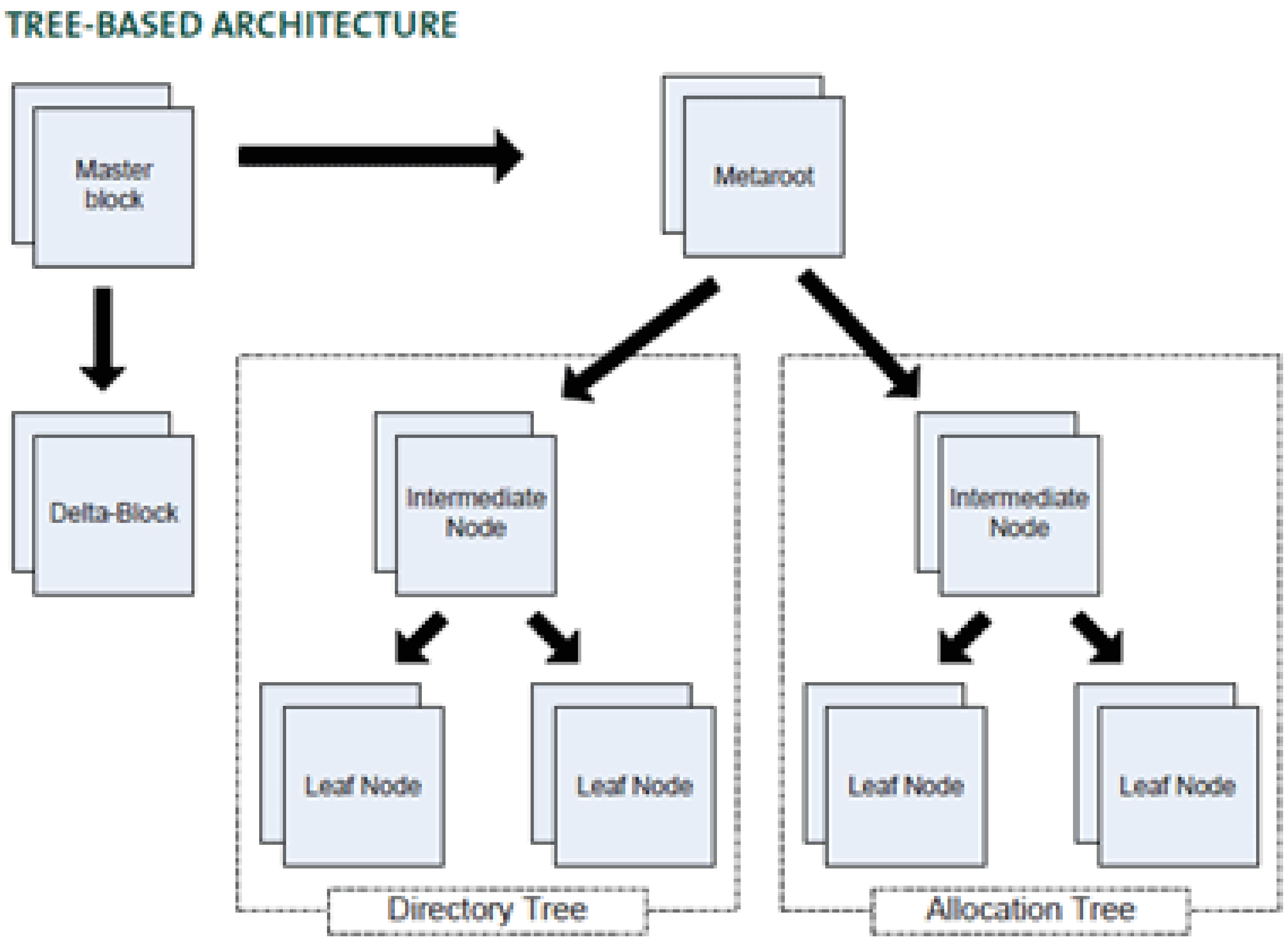
Clean Dirty Kitchen Items

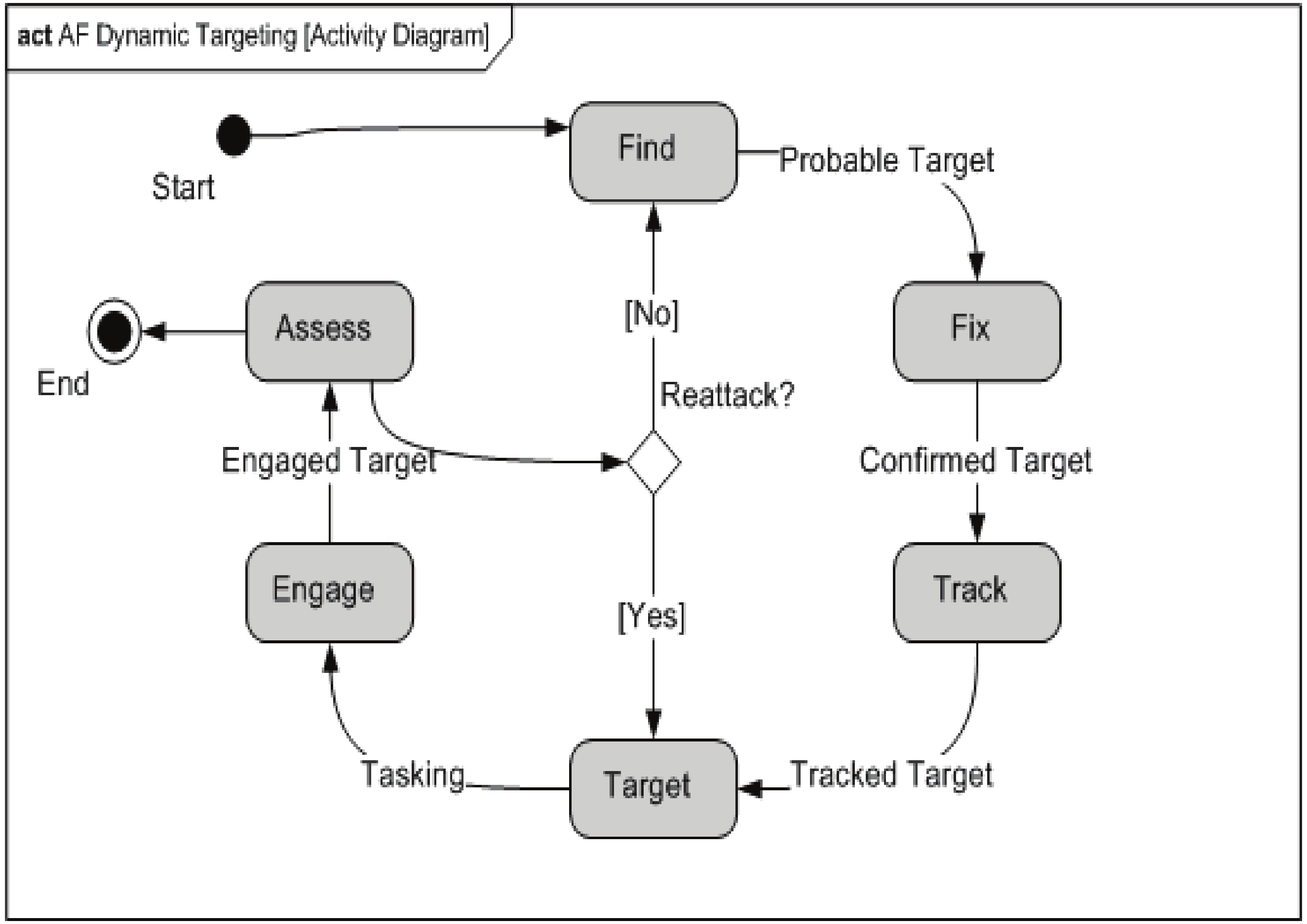
IDEF is an abbreviation of ICOM DEFinition

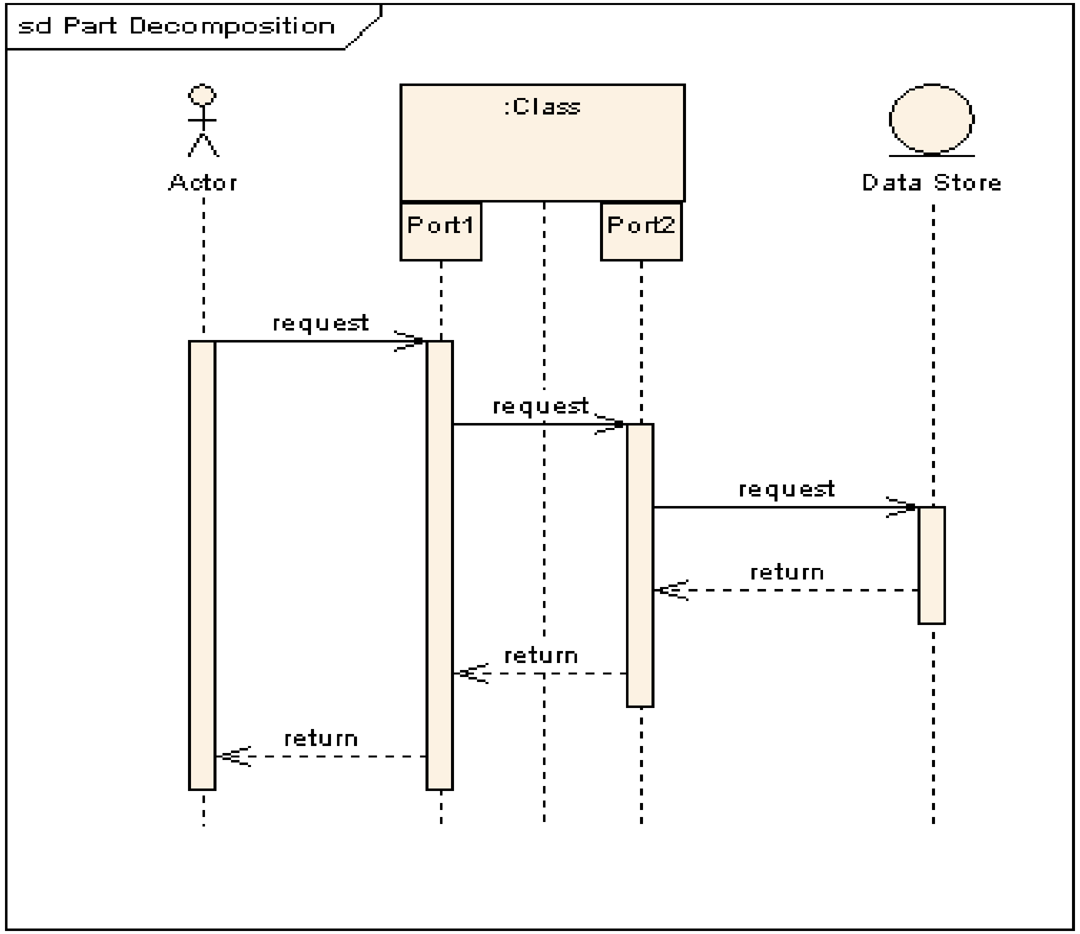
N^2 diagram


n2 Perform Command Center Functions
formatted request
collector data
4
Check Product Inventory
inventory request
5
Prioritize Request
priority of request
priority of request
6
Determine Collector Mix
collector mix
7
Notify User Of Estimated Schedule
estimated delivery schedule
8
Task Collectors
collector tasking
9
Accept And Format Collector Products
10
Put Product In Inventory
11
Get Product From Inventory
inventory product

Tree diagrams



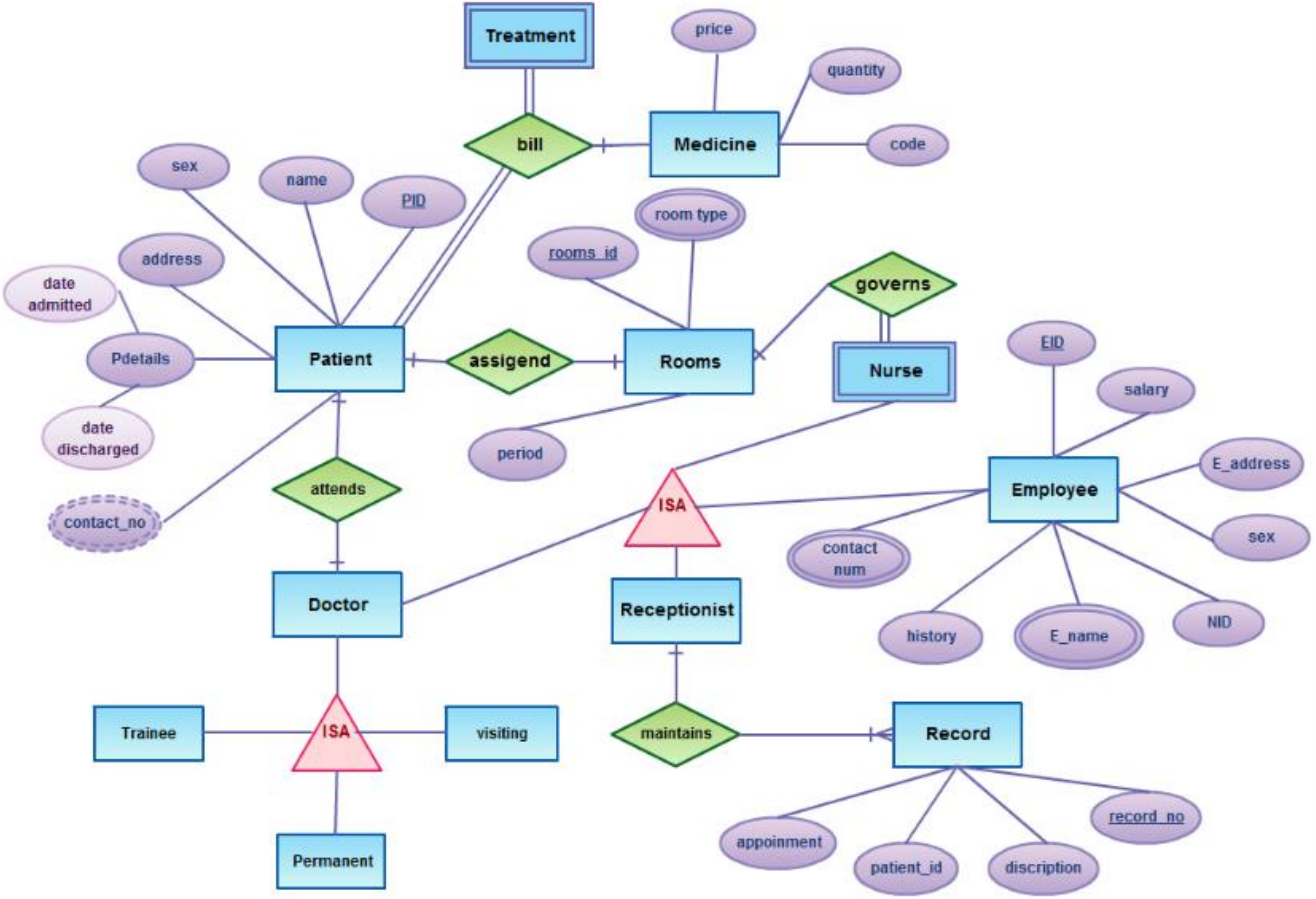




Activity Diagram shows a workflow - a starting point, actions, decisions, splits and joins to show concurrent activities, and ending points – used for **process (workflow) modeling**

A Sequence Diagram shows interactions between actors and objects and between two objects - **dynamic modeling purpose**

Entity F


Treatment
price
quantity
bill
Medicine
code
sex
name
PID
room type
rooms_id
address
date admitted
governs
Pdetails
Patient
assigend
Rooms
Nurse
EID
salary
date discharged
period
E_address
attends
ISA
Employee
contact_no
sex
contact num
Doctor
Receptionist
history
E_name
NID
Trainee
ISA
visiting
maintains
Record
record_no
appoinment
patient_id
discription
Permanent

context diagram

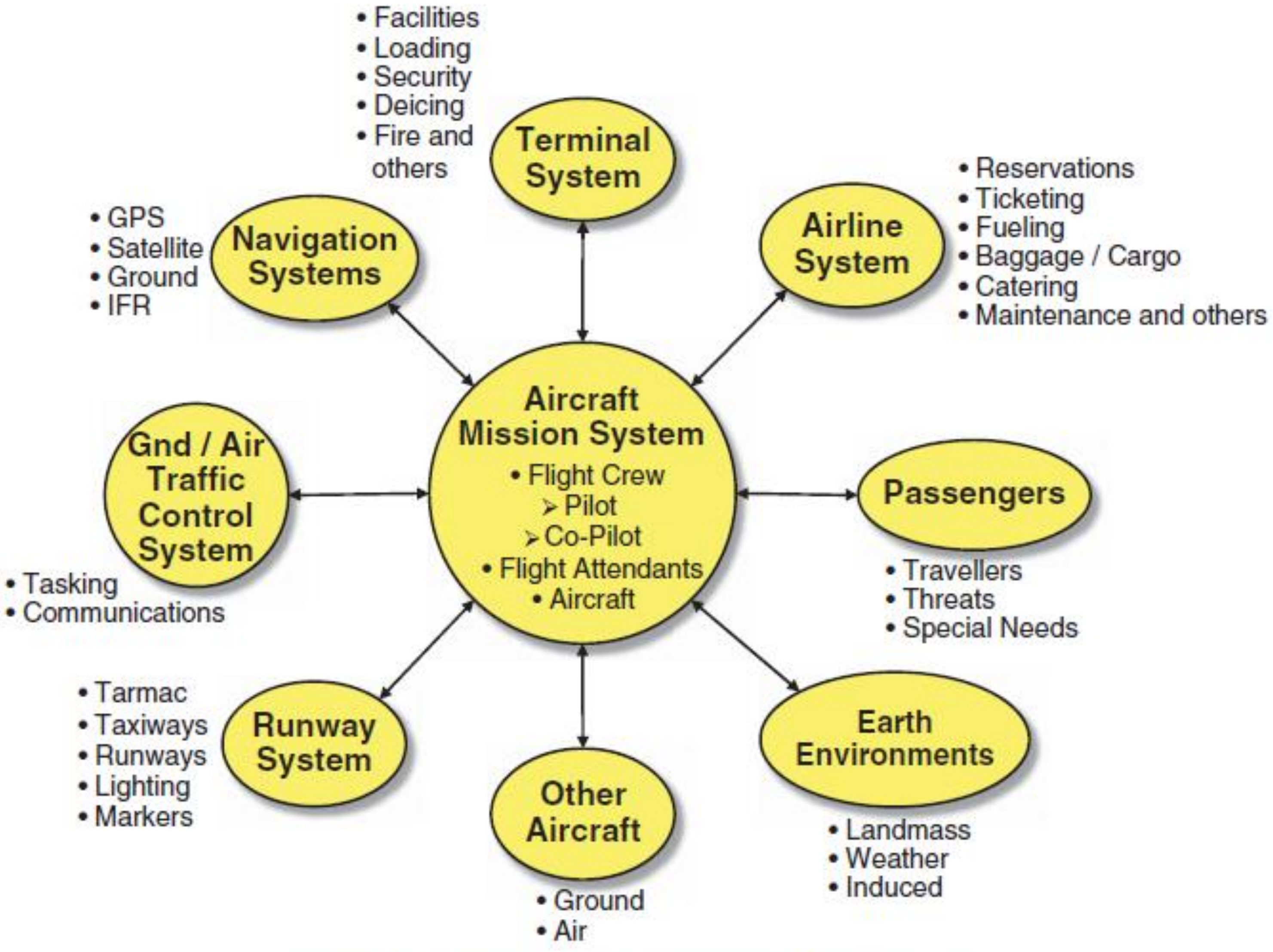
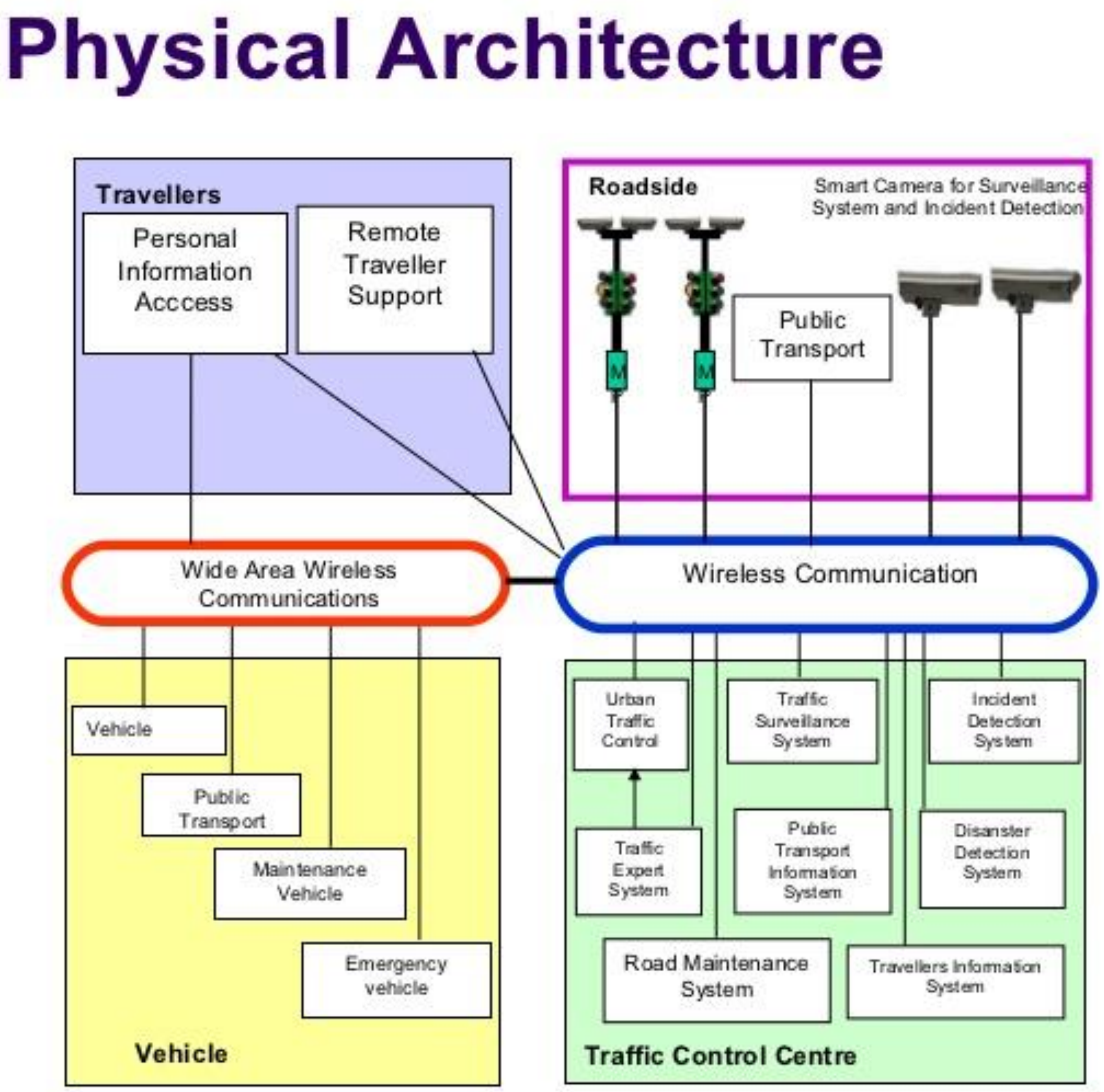


Figure 8.1 Context Diagram for an Aircraft Mission System

Source: Systems Engineering by Wasson

Examples – systems & product level

Intelligent Transport



UKM Architecture

Kumar N., Kumari N. (2012) Conceptual Architectural Design of Indian Railway Intelligent Transportation Systems. In: Vinel A., Mehmood R., Berbineau M., Garcia C.R., Huang CM., Chilamkurti N. (eds) Communication Technologies for Vehicles. Nets4Cars/Nets4Trains 2012. Lecture Notes in Computer Science, vol 7266. Springer, Berlin, Heidelberg

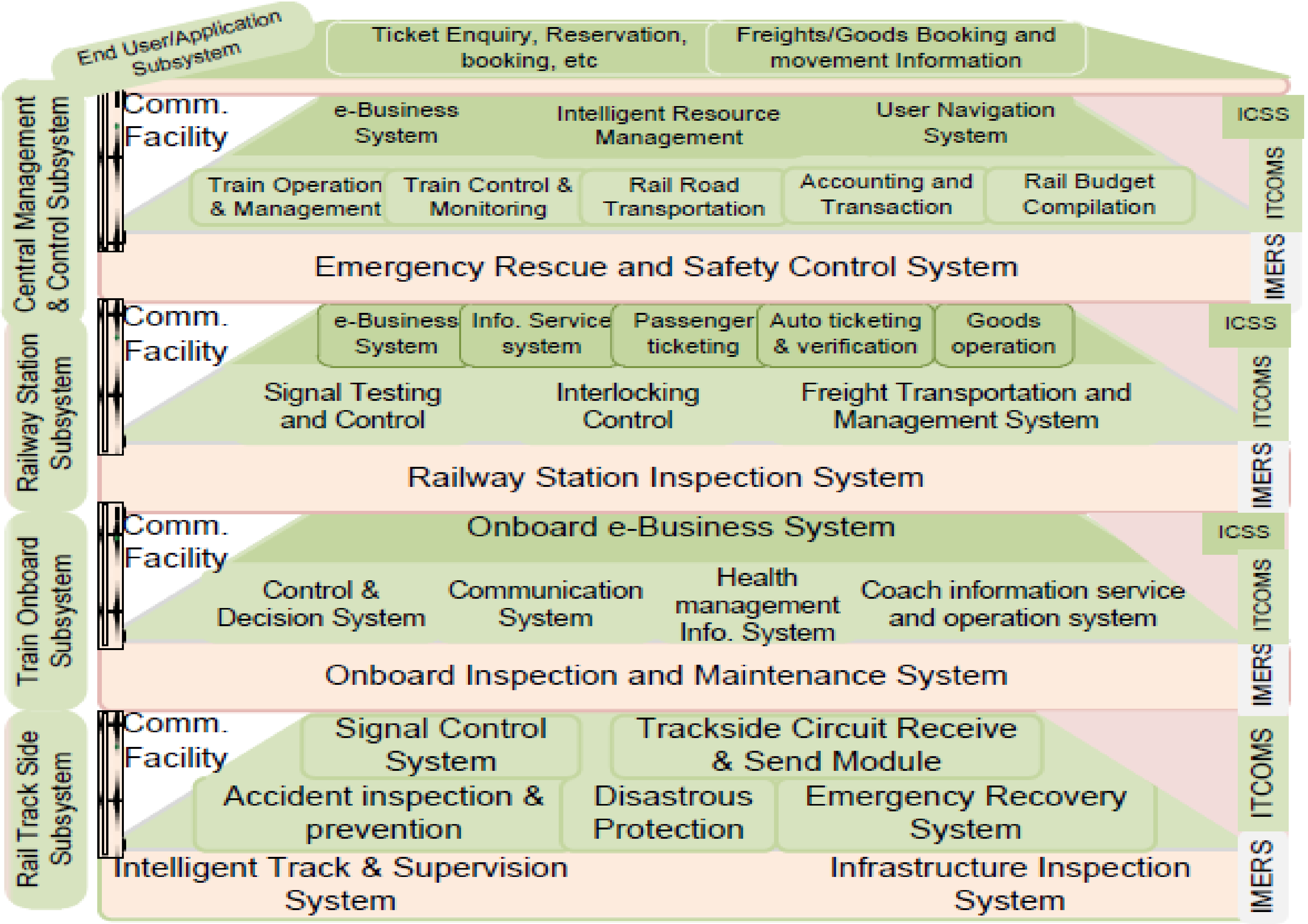
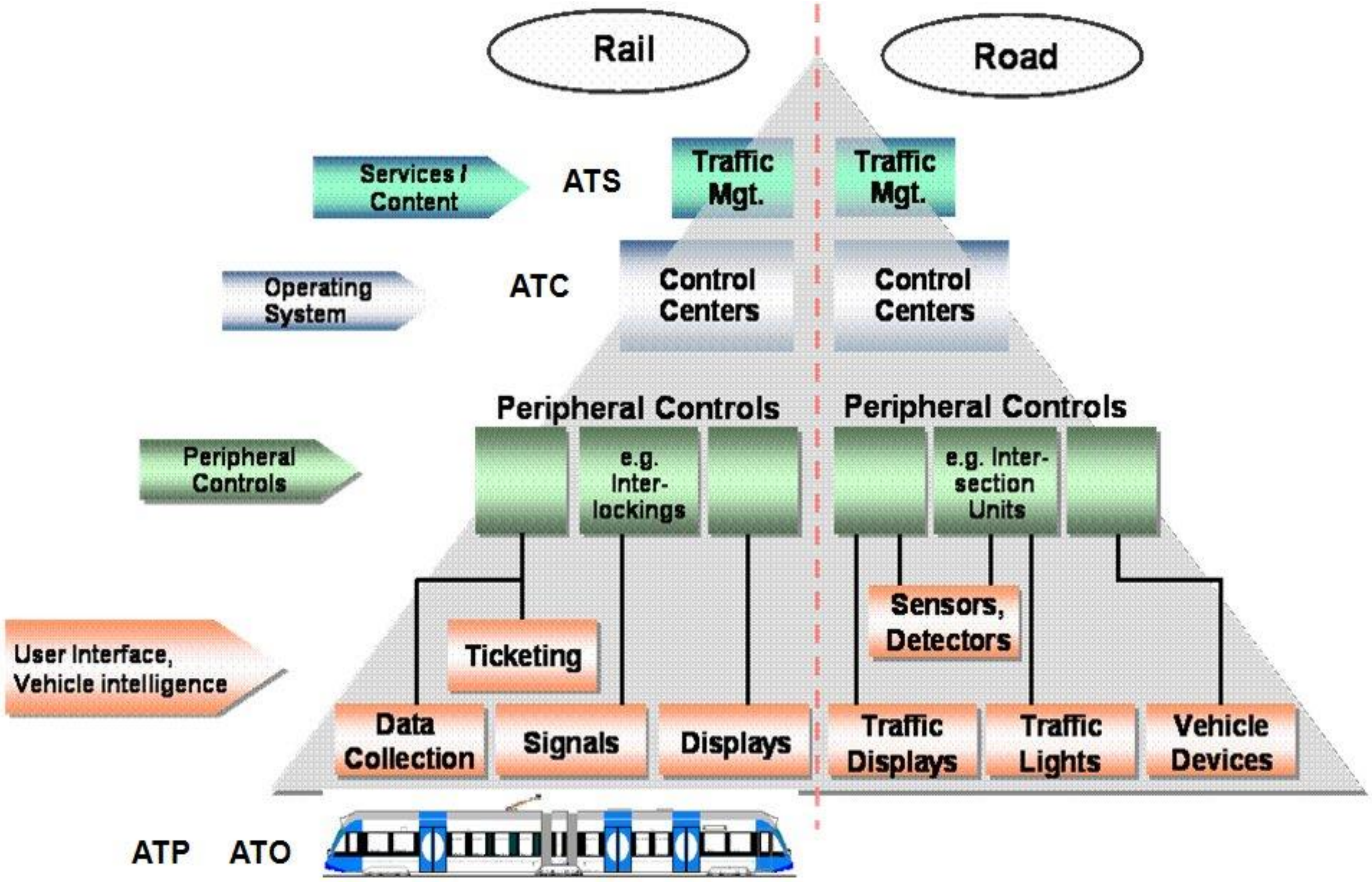
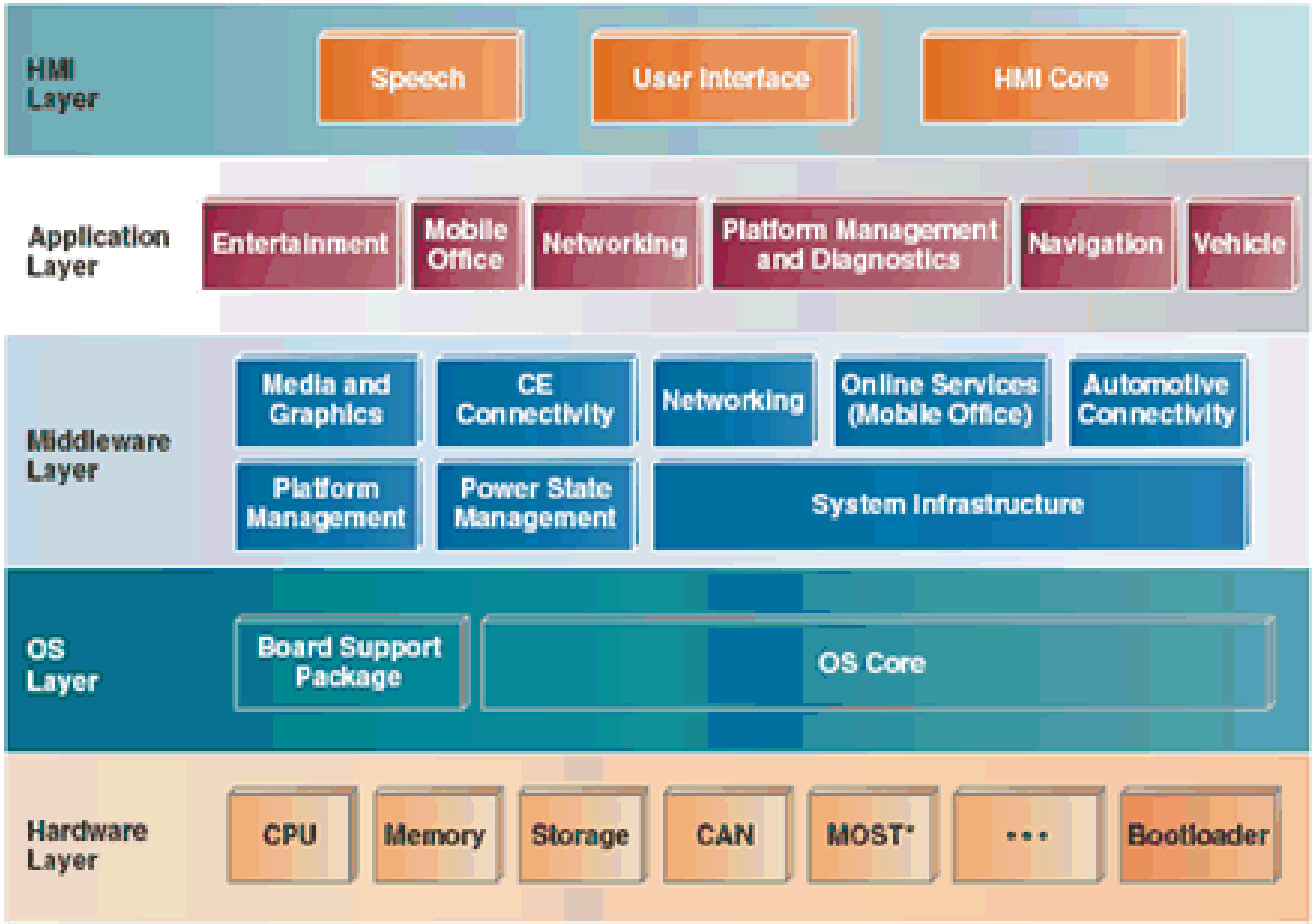


Fig. 2. Railway Intelligent Transportation Systems Architecture

Railway signal control architecture



An Architecture for In-Vehicle Infotainment Systems



Source: Intel team article in Dr Dobbs

Runtime Game Engine architecture