Engineering 180
Systems Engineering

Professor Kenneth Kung

kennethkung@ucla.edu

(Cell) 562-387-3493

Preliminary Design

Agenda

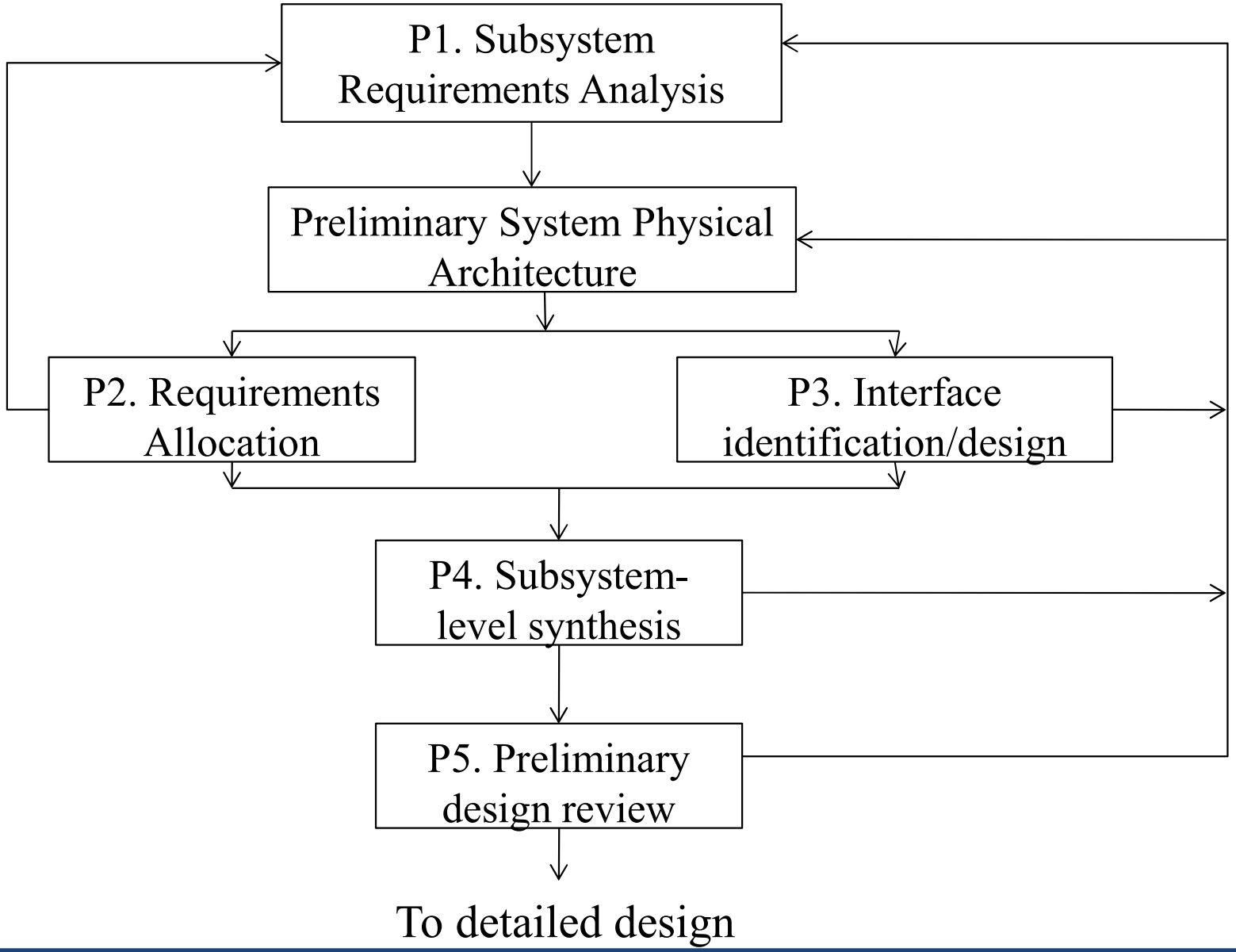
- Preliminary design process
- System architecture
  - A brief introduction

The Goal of Preliminary Design

- Understand and analyze the system level functional requirement
- Determine the top level system architecture
- Identify subsystems and interfaces
- Flow down system level requirements to these subsystems
- The completed and finalized design is called ***Allocated Baseline***, or physical design, or system view of the system architecture

Preliminary design is usually the first major task of the prime contractor

Preliminary Design Process



Preliminary Design Process – How is it really done?

This is an iterated process:

- Analyzes the System Functional Requirements to the next level of details, and choose a "preliminary" physical architecture, where the subsystems (configuration items – CIs) and interfaces are defined
- Allocates the resulting subsystem-level requirements derived from the System Functional Requirements into the subsystems
- Identify the interfaces between all subsystems
- Synthesize and evaluate the resulting architecture against the system Spec
- Back to step 1, until satisfied with the design.

Zachman Row 3

What is a "physical architecture"?

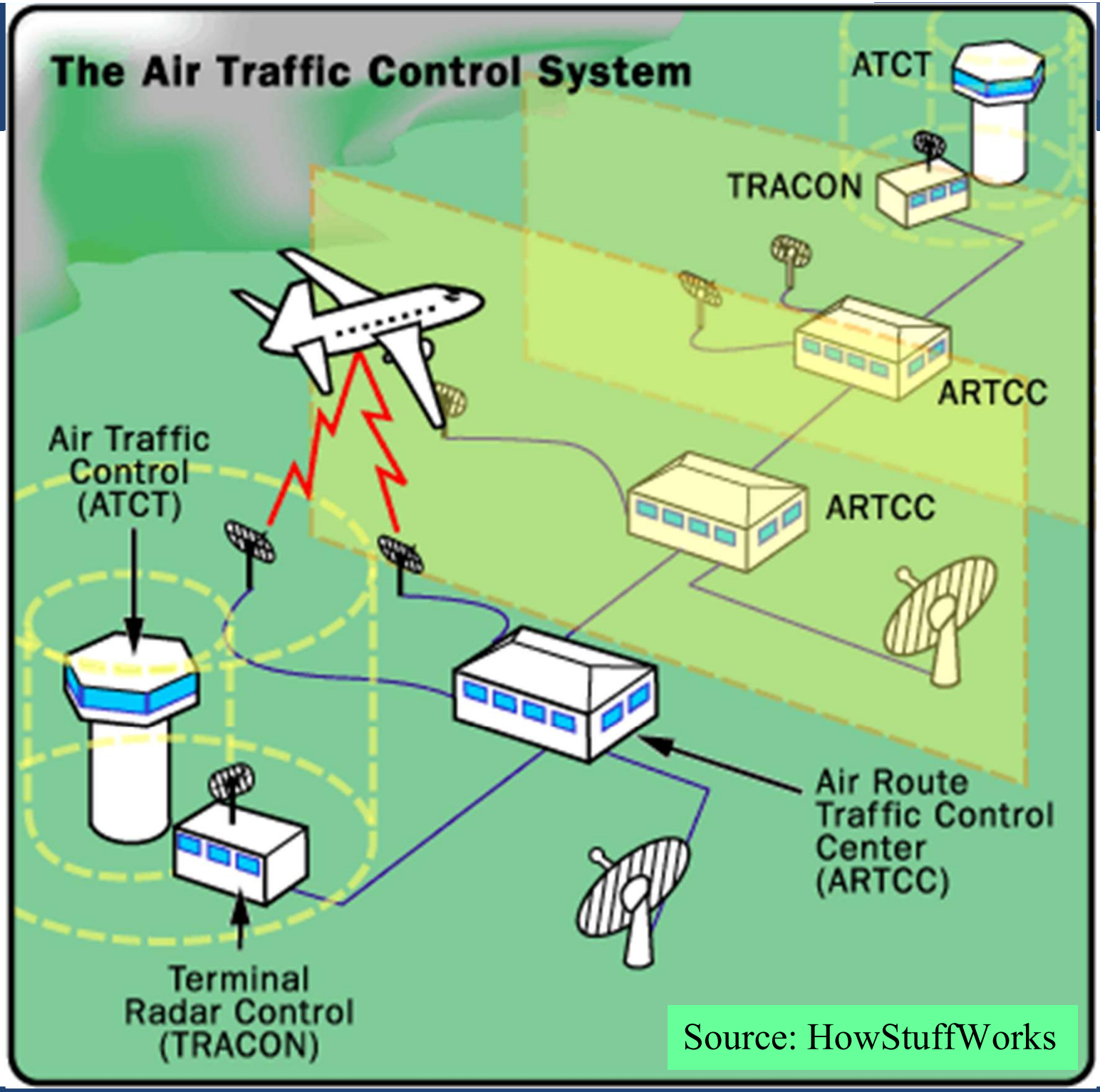
- More appropriately, it is called the "system view" of the system architecture
- It breaks down the system into major "*physical*" substructures that will be designed and built

List Some of FAA Airspace Management System Major Components (i.e., Configuration Items)

Communication	Automation
Surveillance	Navigation
Facilities	Information Systems Security
Aircraft	Weather
Personnel – HMI	Airport

From Takeoff to Landing

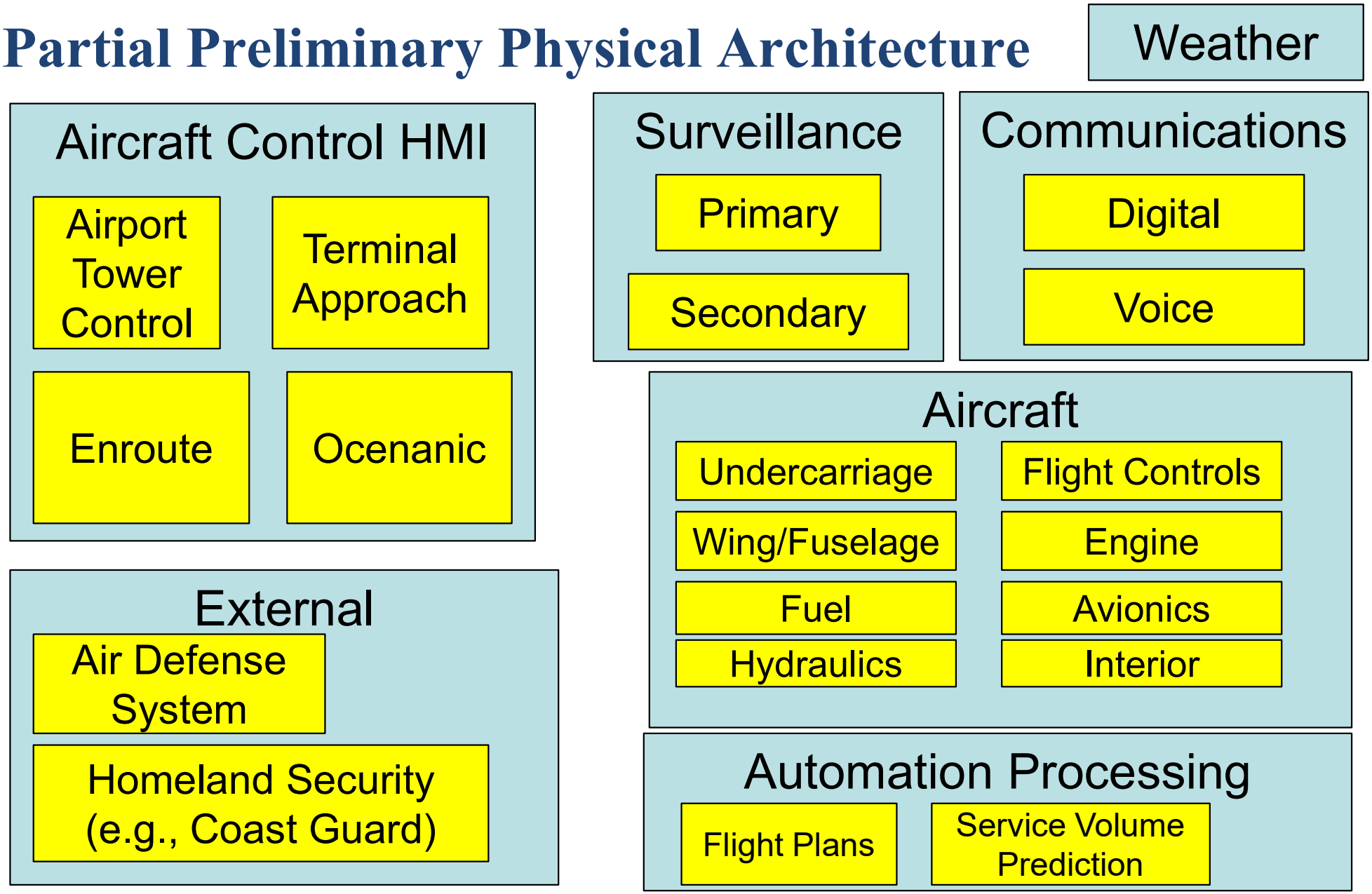
Operational Scenario



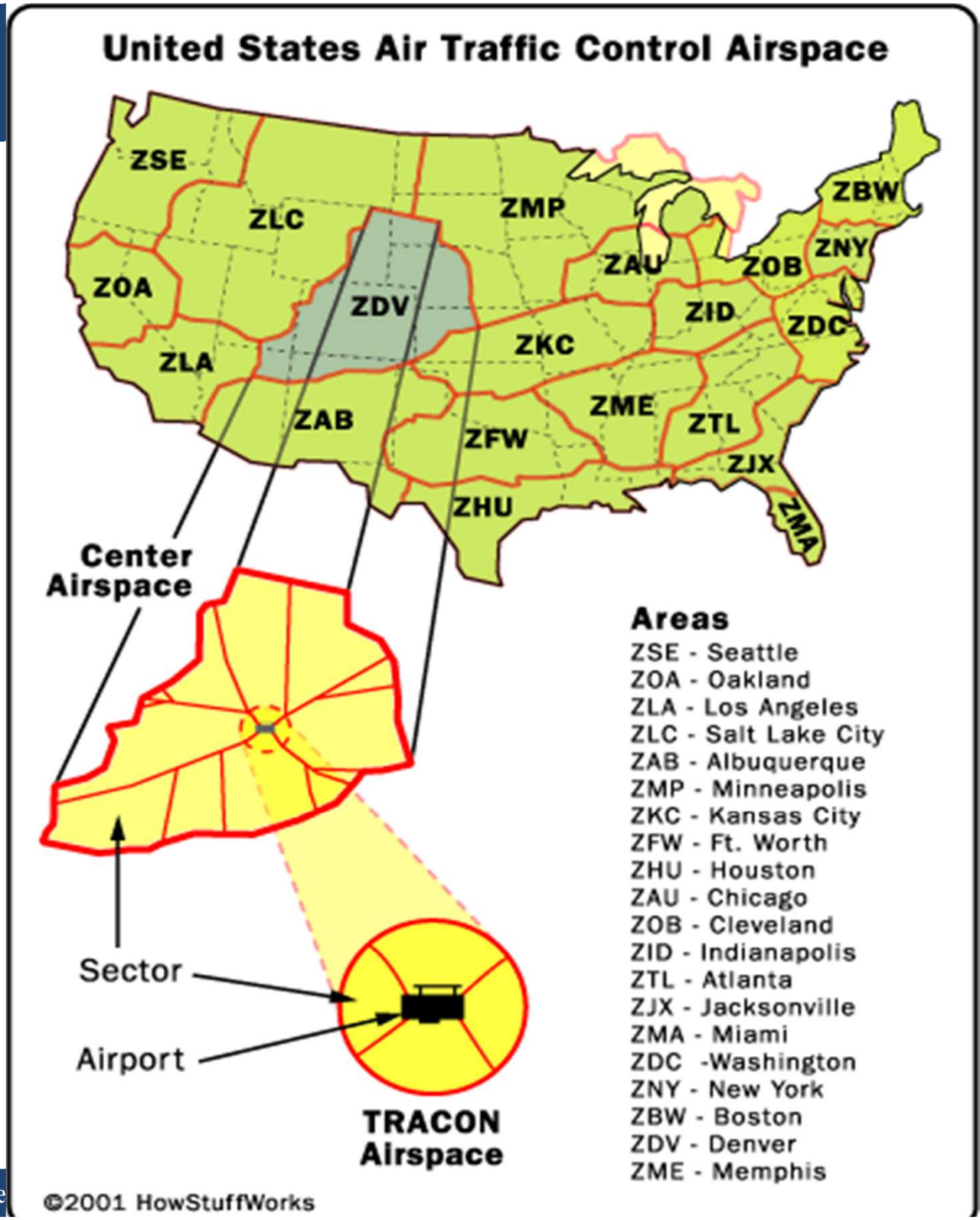
Physical Architecture

- These "*physical substructure*" are not conceptual entities anymore
  - They are concrete entities that will be design and built
  - The are also referred as Configuration Items (CIs)
    - Computer Software CIs (CSCIs) or Hardware CIs (HWCIs)
- HWCI Example: Radar hardware:
  - antenna, transmitter, receiver, processor, …
- SWCI Example: Radar software:
  - operational software, support software, diagnostic software, …

Partial Preliminary Physical Architecture



Air Traffic Control Airspace
- Air Route Traffic Control Center
- Enroute System



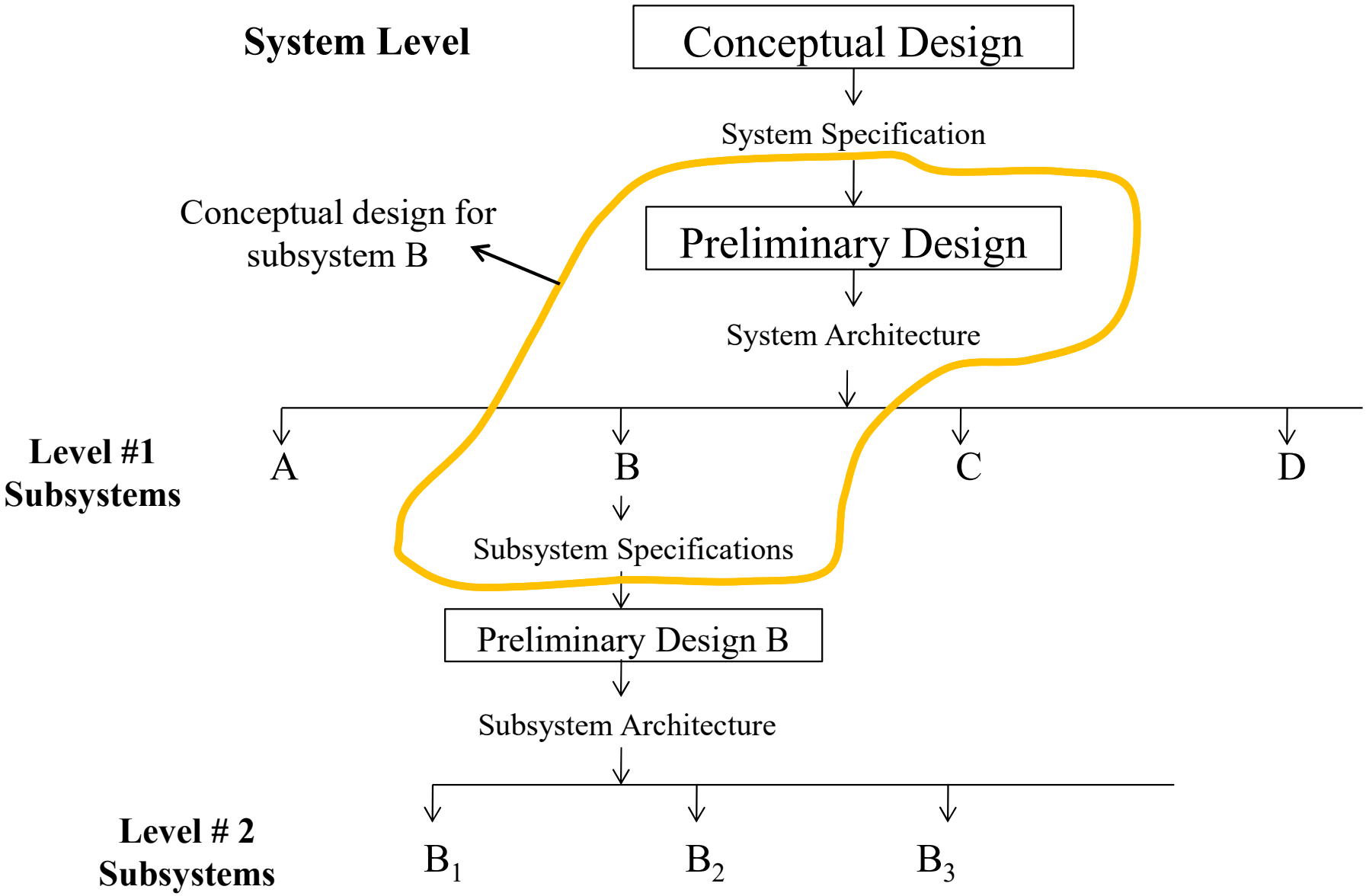
How to identify a physical architecture?
How the system architecture is created (designed)?

- Short Answer: It is based on **experience, knowledge, creativity**, and it is the result of **trade studies**, **reviews**, and **debates**
- System architecture is the heart of design effort!
- System architecture and supporting technologies are the technical foundation of product platform concept
- Usually at the beginning of preliminary design, contractors have their preferred architecture, existing products or product strategy

Caution

- It is important not to rush the selection of system architecture
  - The system requirements must be understood and analyzed
  - The proposed architecture must be carefully evaluated against all requirements before the final selection

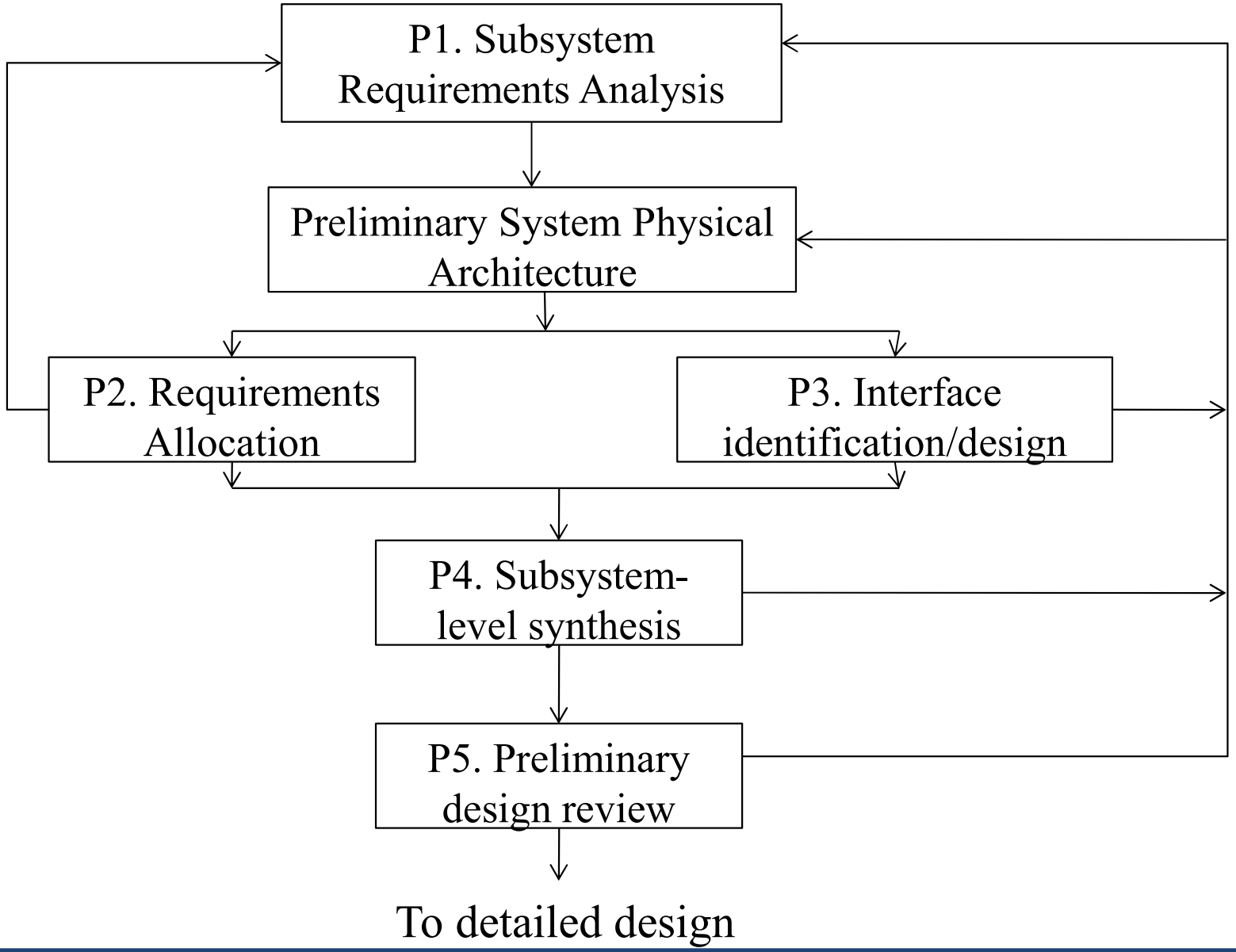
Nested Design Process for Complex Systems



Aircraft Configuration Item - Analysis

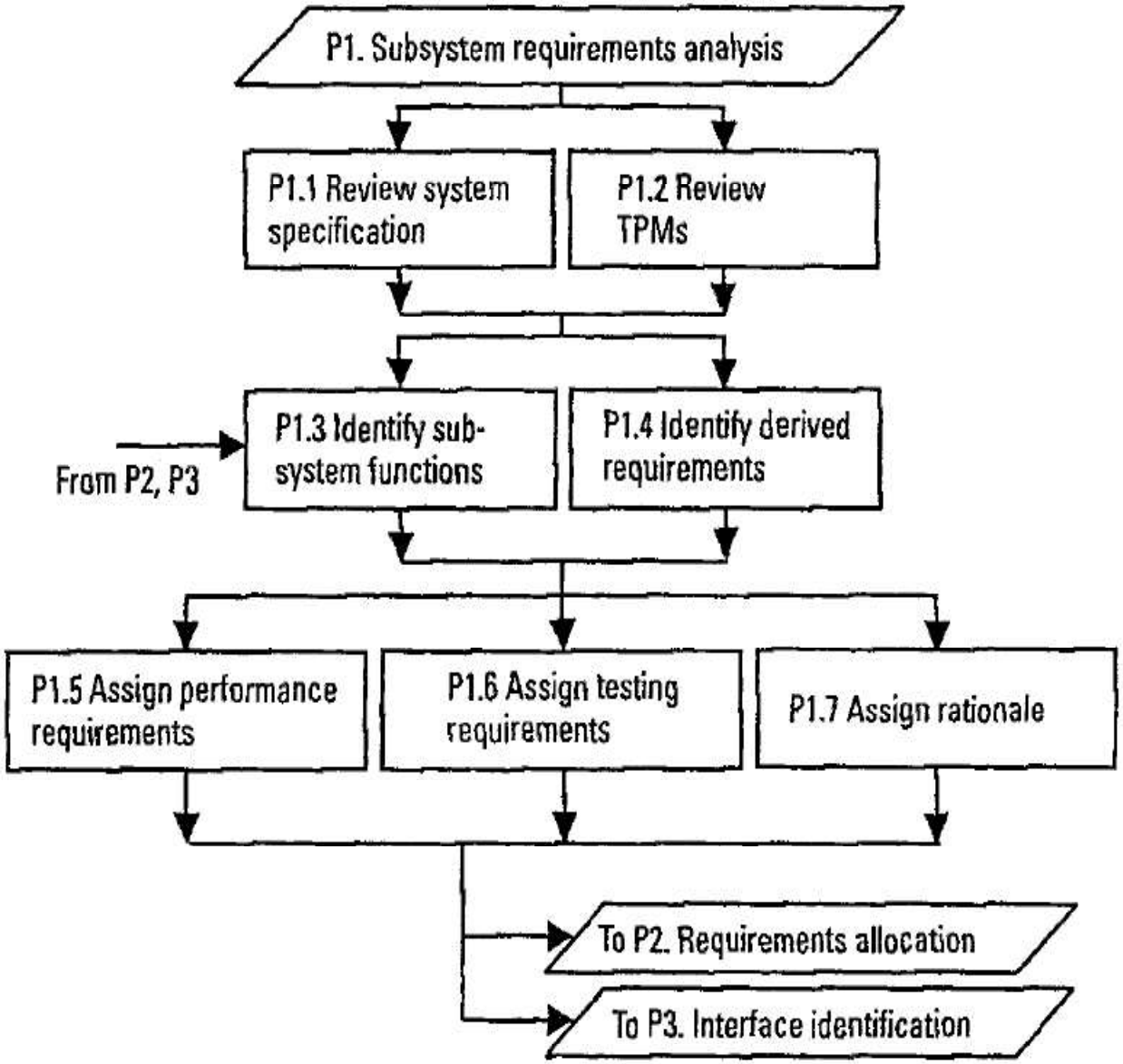
- From Falcombridge book
  - Undercarriage
  - Wing/fuselage
  - Fuel
  - Hydraulic
  - Flight controls
  - Engine
  - Avionics
  - Interior

Preliminary Design Process

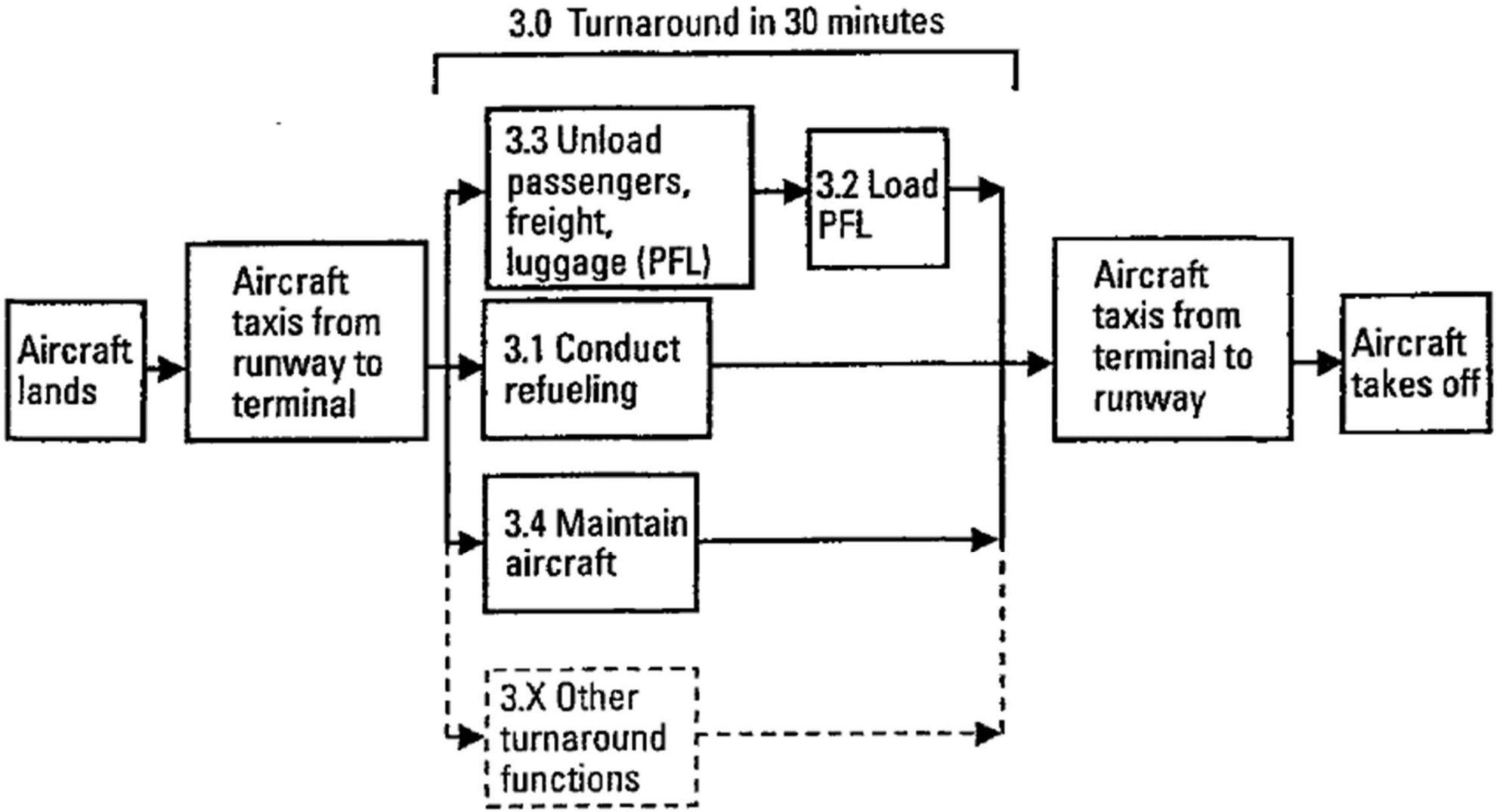


Subsystem Requirement Analysis (P1)

- The same requirement analysis process are used at all levels
- There could be several level of subsystems in a more complex system
  - This process will be done repeatedly
- Understand the higher level requirement is always the first step

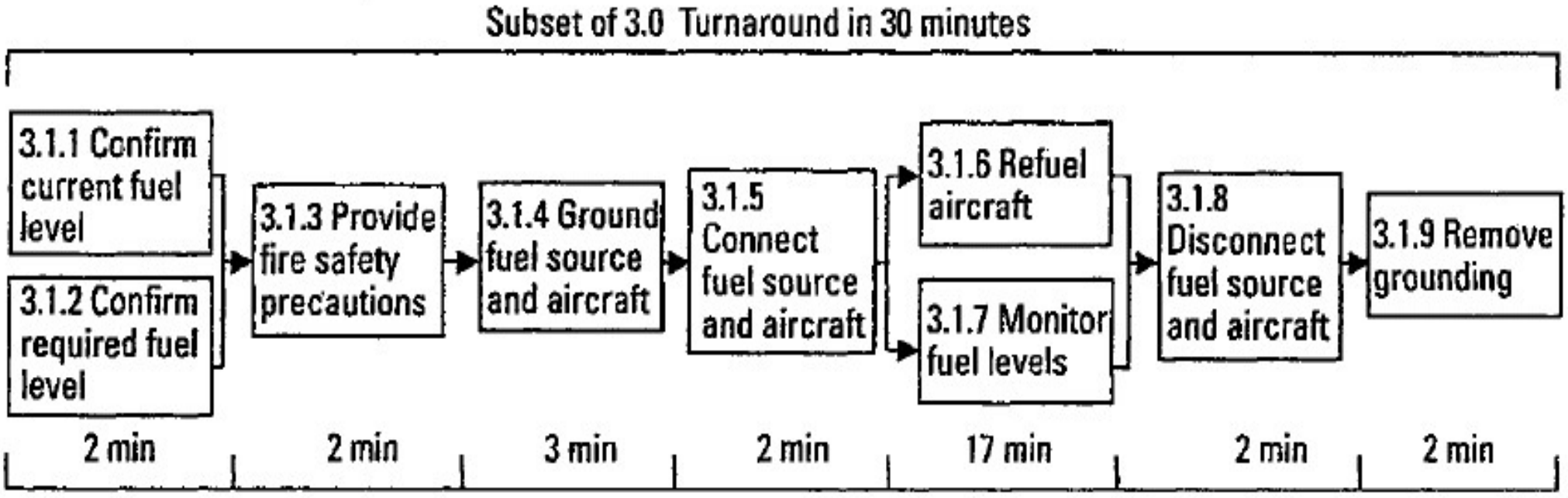


Functional decomposition and requirement flow down – Aircraft turn around in 30 minutes (1)

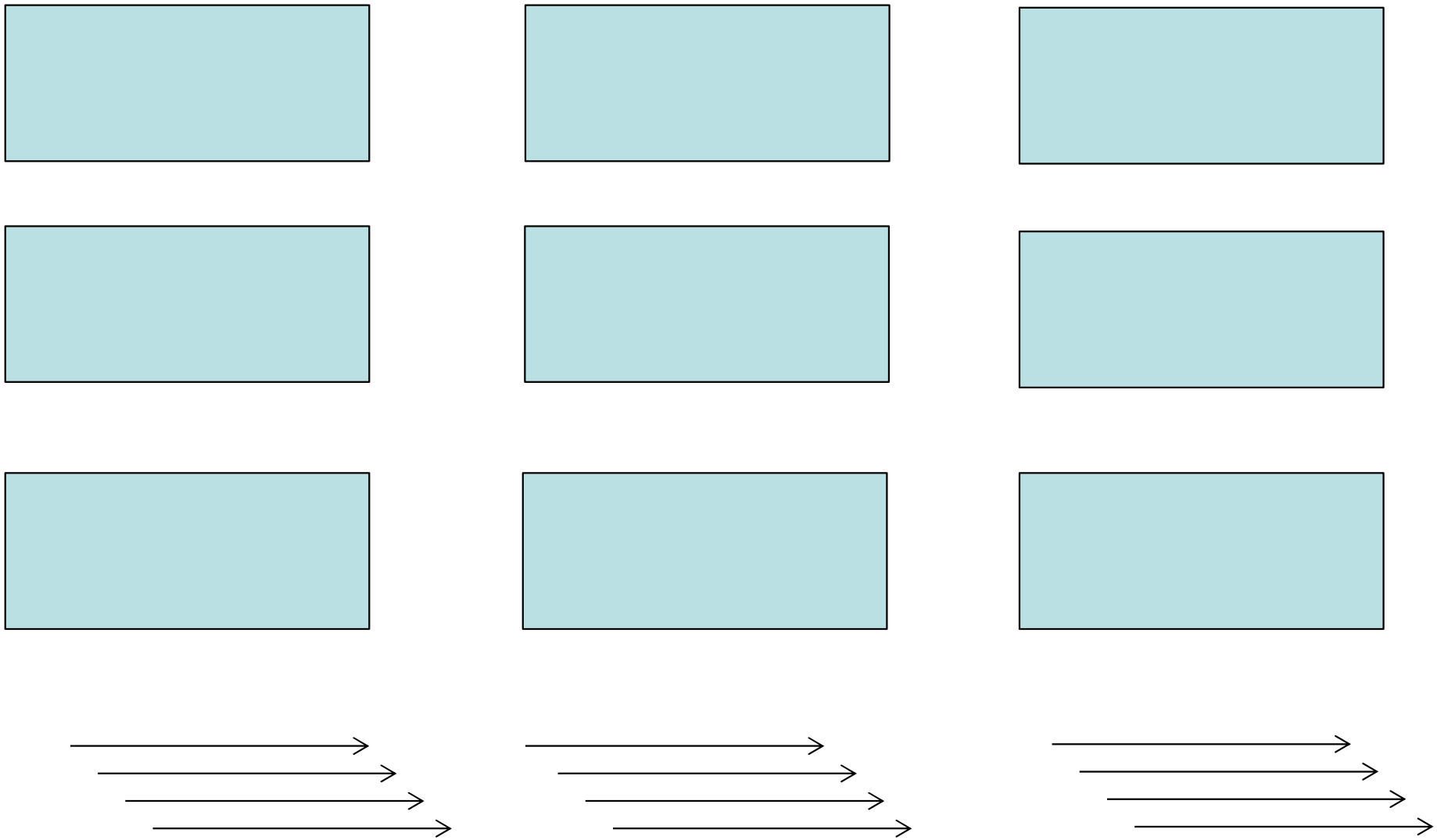


Subsystem requirement analysis example – aircraft refueling

- Aircraft must be able to be refueled, while the fuel level is monitored, from empty to maximum fuel level in less than 17 minutes



In Class Exercise (Choice 1)
Let us do an expansion of 3.1.6 Refuel Aircraft



In Class Exercise (Choice 2)

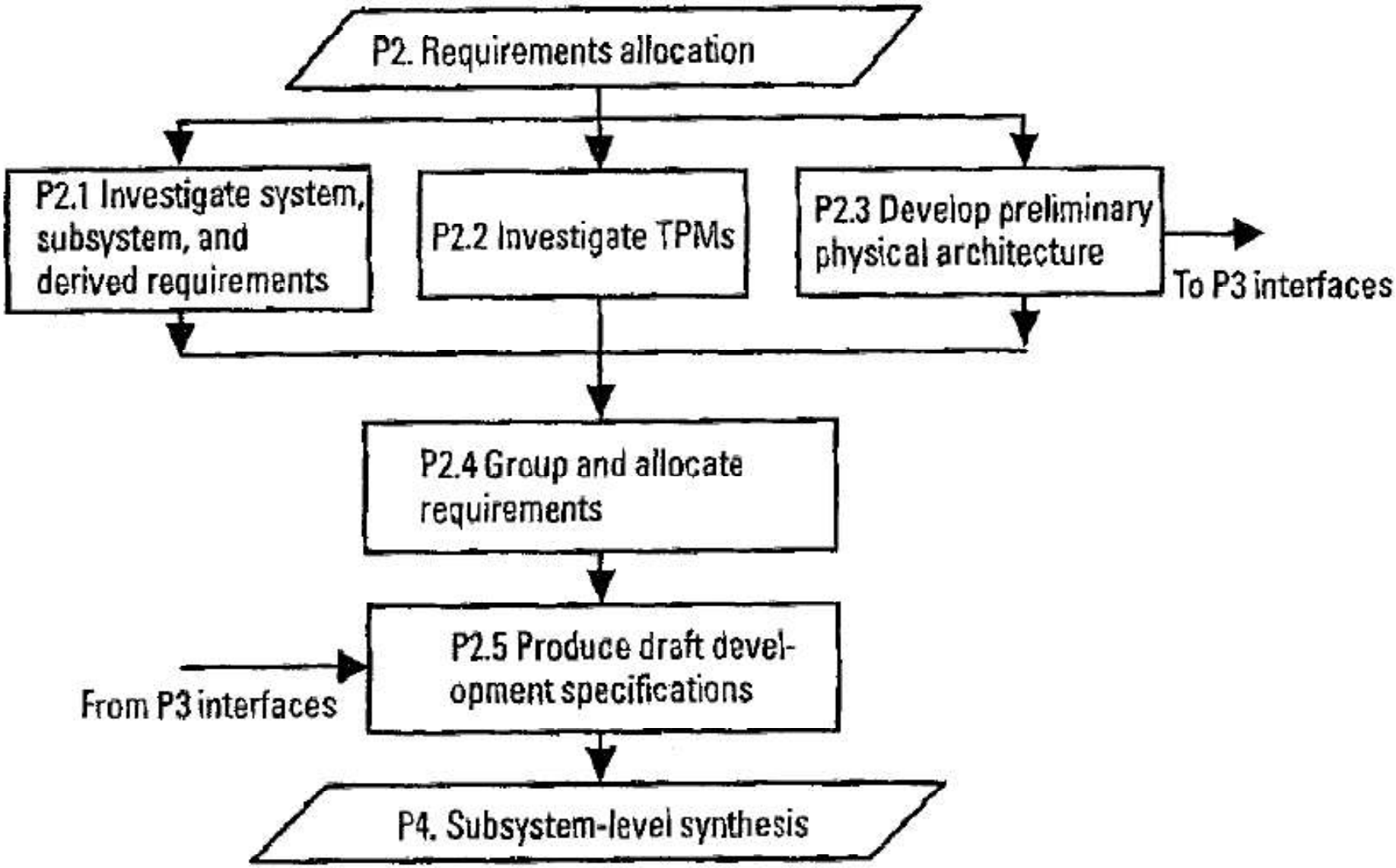
- Develop a FFBD for your group project
- Pick a requirement and develop the FFBD for it
  - Select one of the block and analyze its functions to the next level of details
- When your group project next meet, analyze each other's FFBD

Class Discussion – 15 minutes

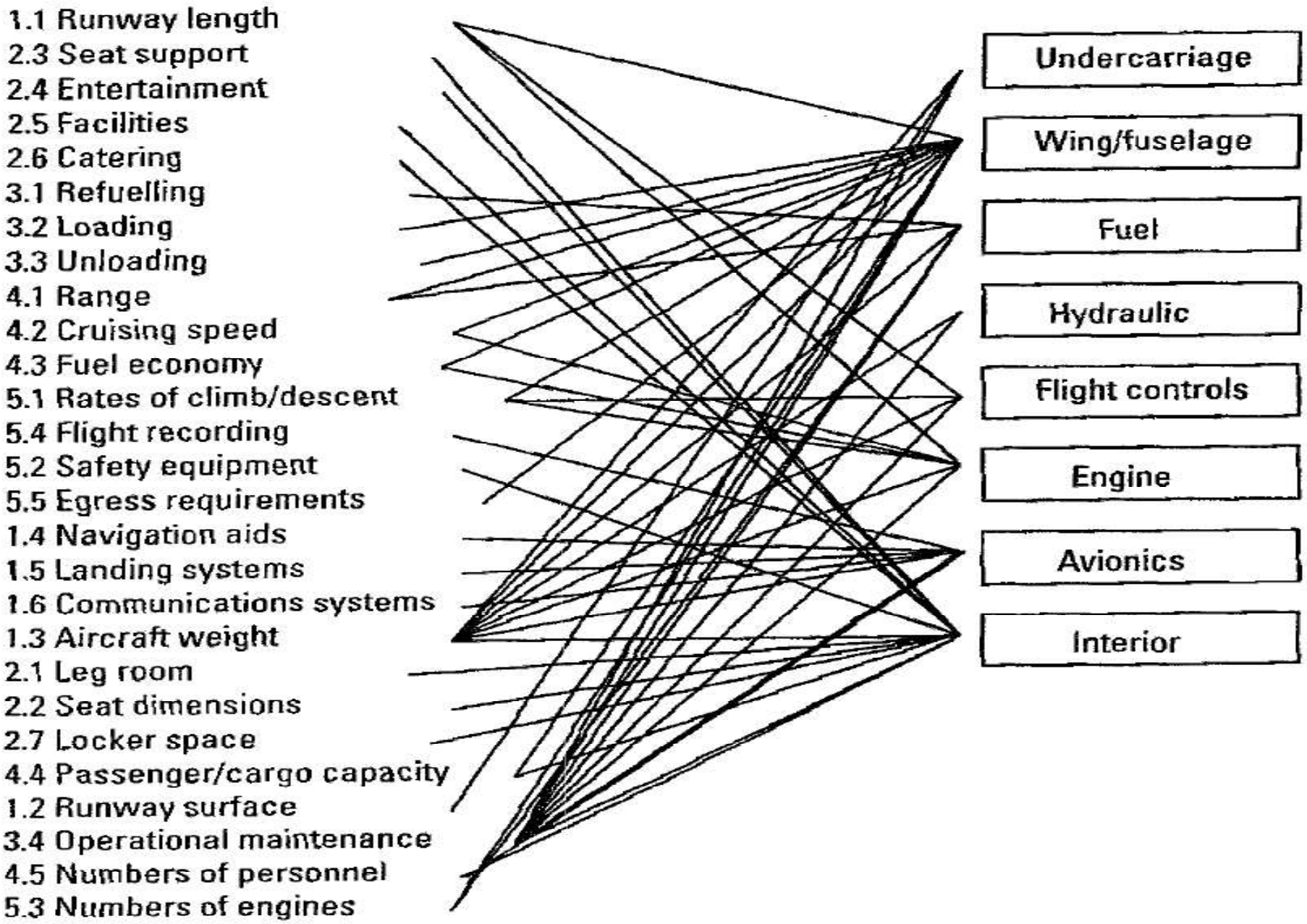
- Do a subsystem requirements analysis for
  - 3.4 Maintain Aircraft

    Or
  - 3.X Other turn around functions
- Draw a FFBD and be prepared to show at debriefing

Requirements Allocation (P2)



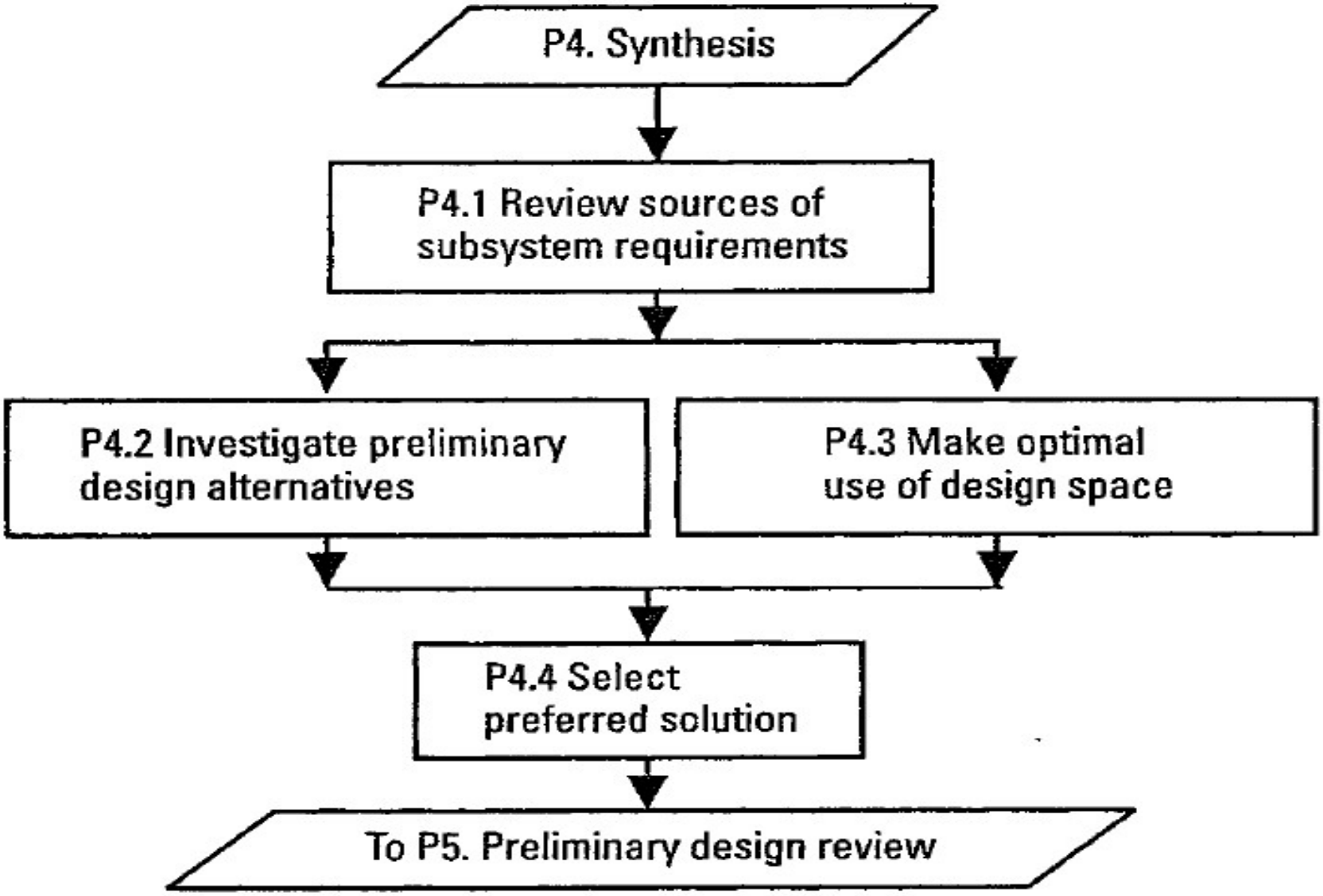
Requirement Allocation Example - Aircraft



Interface Identification and Design (P3)

- It defines the interfaces between the different subsystems
  - Physical interfaces
  - Electronic interfaces
  - Electrical interfaces
  - Hydraulic/pneumatic interfaces
  - Software (Data) interfaces
  - Environmental interfaces
- Interface Control Document (ICD) is one of the most important document. It is heavily used in every phase of system lifecycle.

Subsystem-level Synthesis and Evaluation (P4)



Class Discussion – 15 minutes

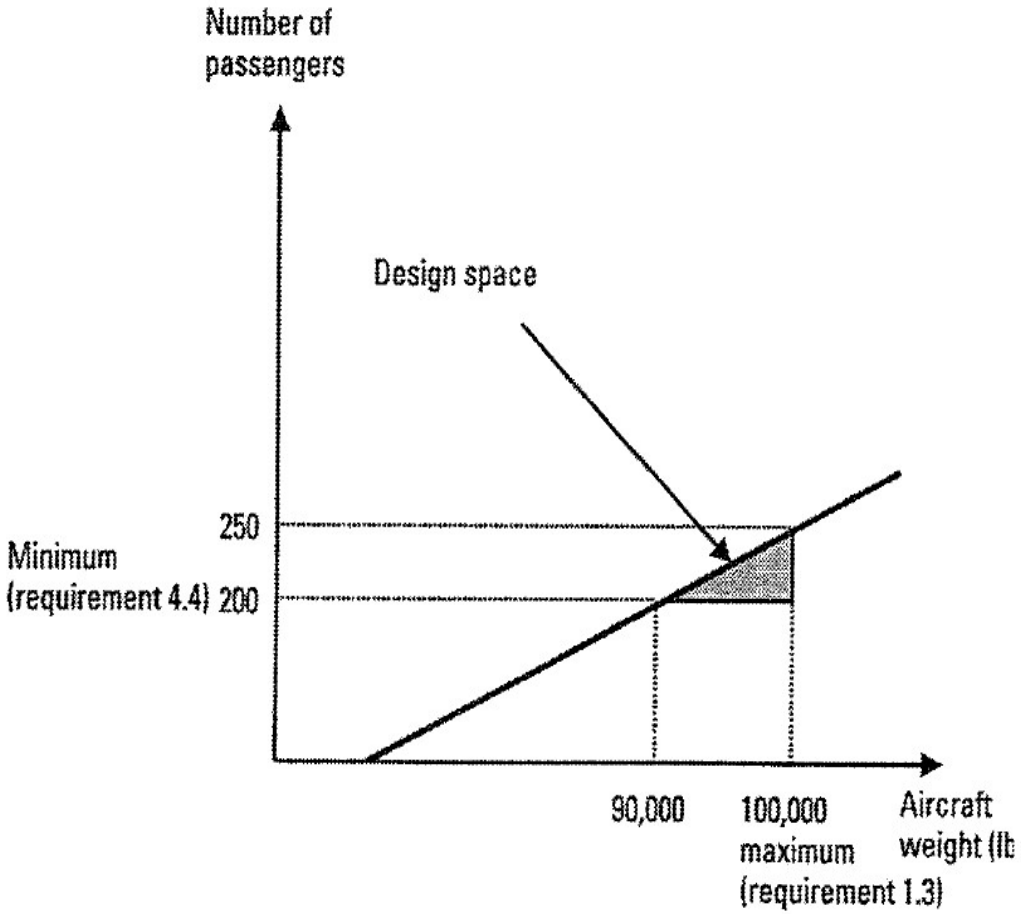
- Develop a preliminary physical architecture for your group project
  - 5 to 7 Configuration Items
  - Interfaces
- Use the requirements and operation scenario(s) that you developed during the conceptual design lecture
  - Allocate the system functions to the physical architecture components

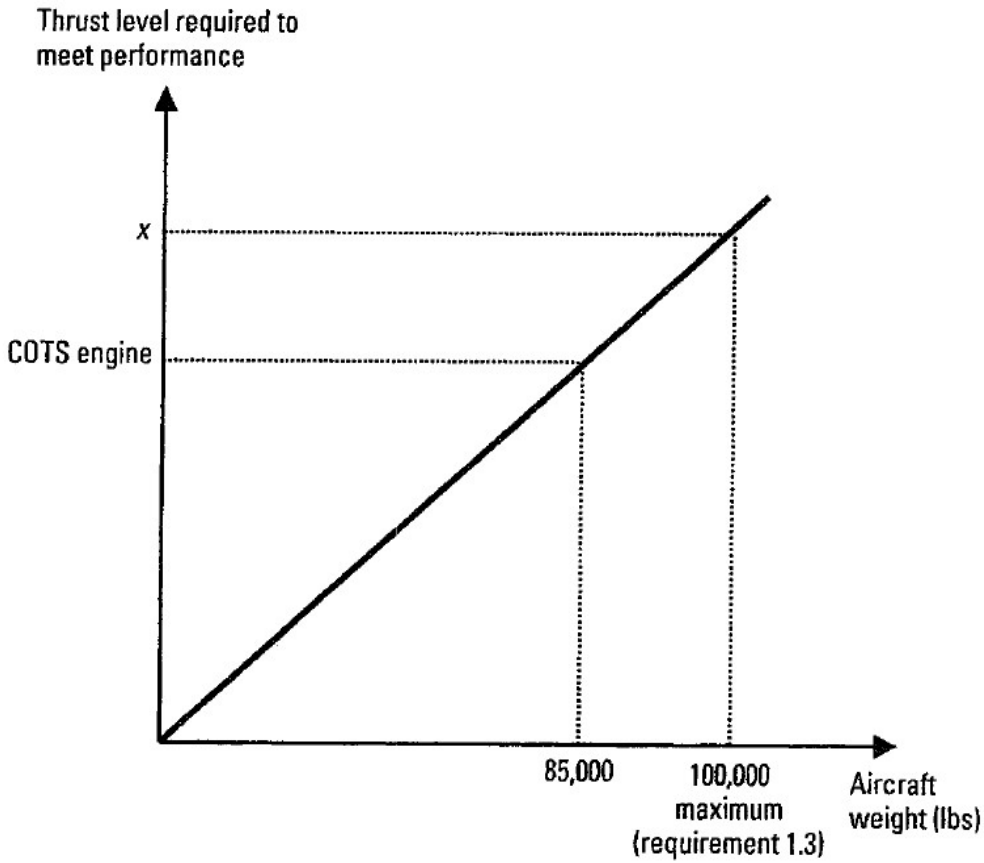
A special trade study – Existing equipment vs developmental item

- Existing equipment, such as off-the-shelf equipment and COTS are usually:
  - Lower cost (both development and procurement cost)
  - Lower risk (Technology, manufacturing, and schedule)

  But
  - May not meet all requirements
  - May have life cycle supportability issues – Technology and parts availability
- Directions of trade studies:
  - Could system requirements be relaxed to accommodate off-the-shelf equipment?
  - Could other related design parameters be tightened to allow the use of off-the-shelf equipment?

An example of COTS considerations – COTS engine
- Could the number of passengers requirement be reduced?
- Could the number of passengers/aircraft weight model be improved?

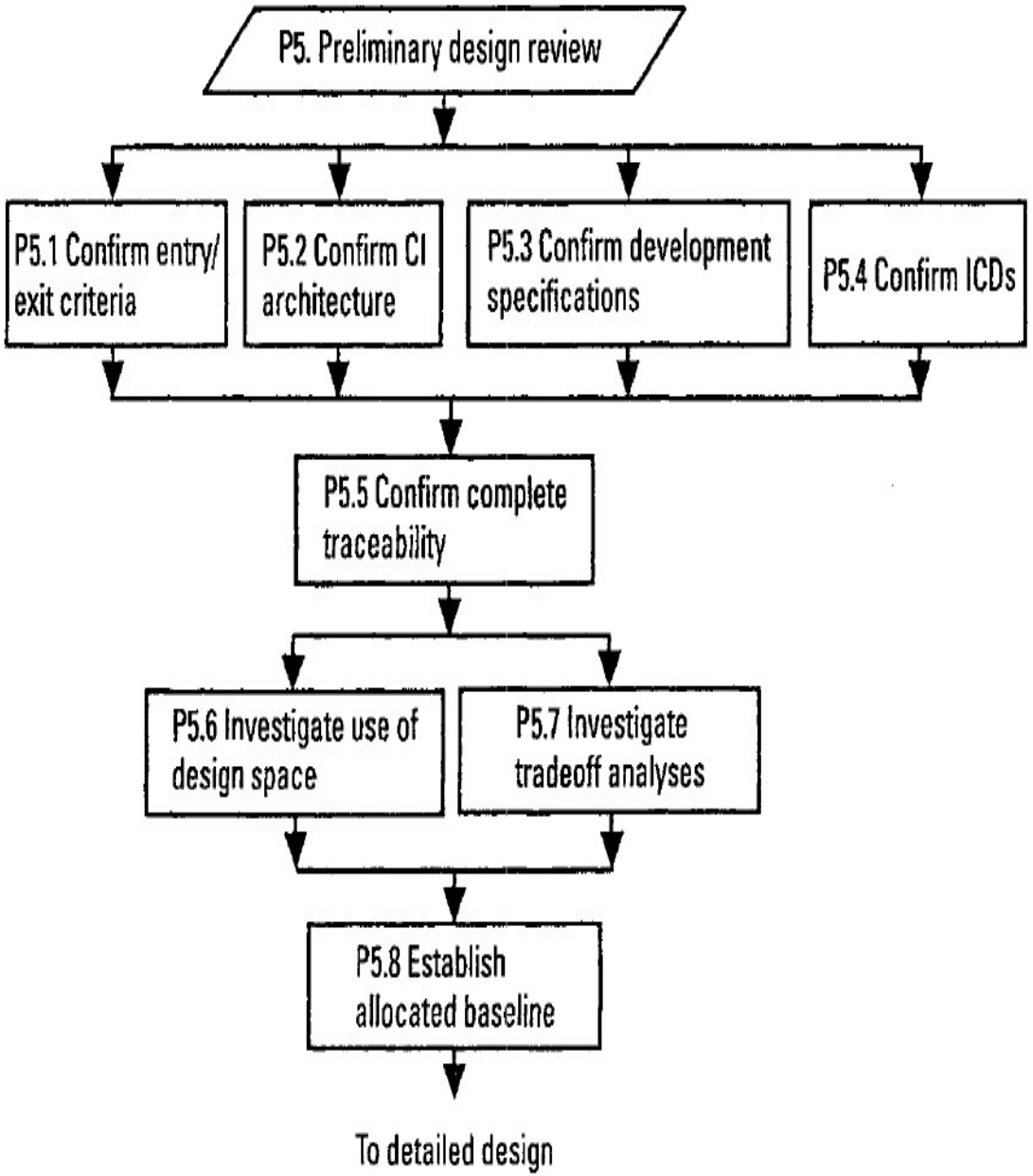




Preliminary design review (P5)

Objectives:

- Make sure the functional baseline requirements have been adequately addressed by the preliminary design
  - Physical architecture
  - Interfaces
  - Subsystem functional requirements
- Key documents:
  - Subsystem specifications
  - Interface control document (ICD)



A few remarks on design reviews

- Formal design reviews:
  - Systems Requirements Review – Stakeholder requirements
  - System Design Review (SDR) – Functional Baseline
  - Preliminary Design Review (PDR) – Allocated Baseline
  - Critical Design Review (CDR) – Product Baseline
- The purpose of formal design reviews are to make sure all design efforts and required documents are completed; and are ready to move to the next phase of development
  - Internal reviews, internal and external coordination, corrective actions should all be done prior to the formal design reviews
  - The action items from these formal design reviews are formally documented and tracked

Final Words on Attributes of Good Requirements

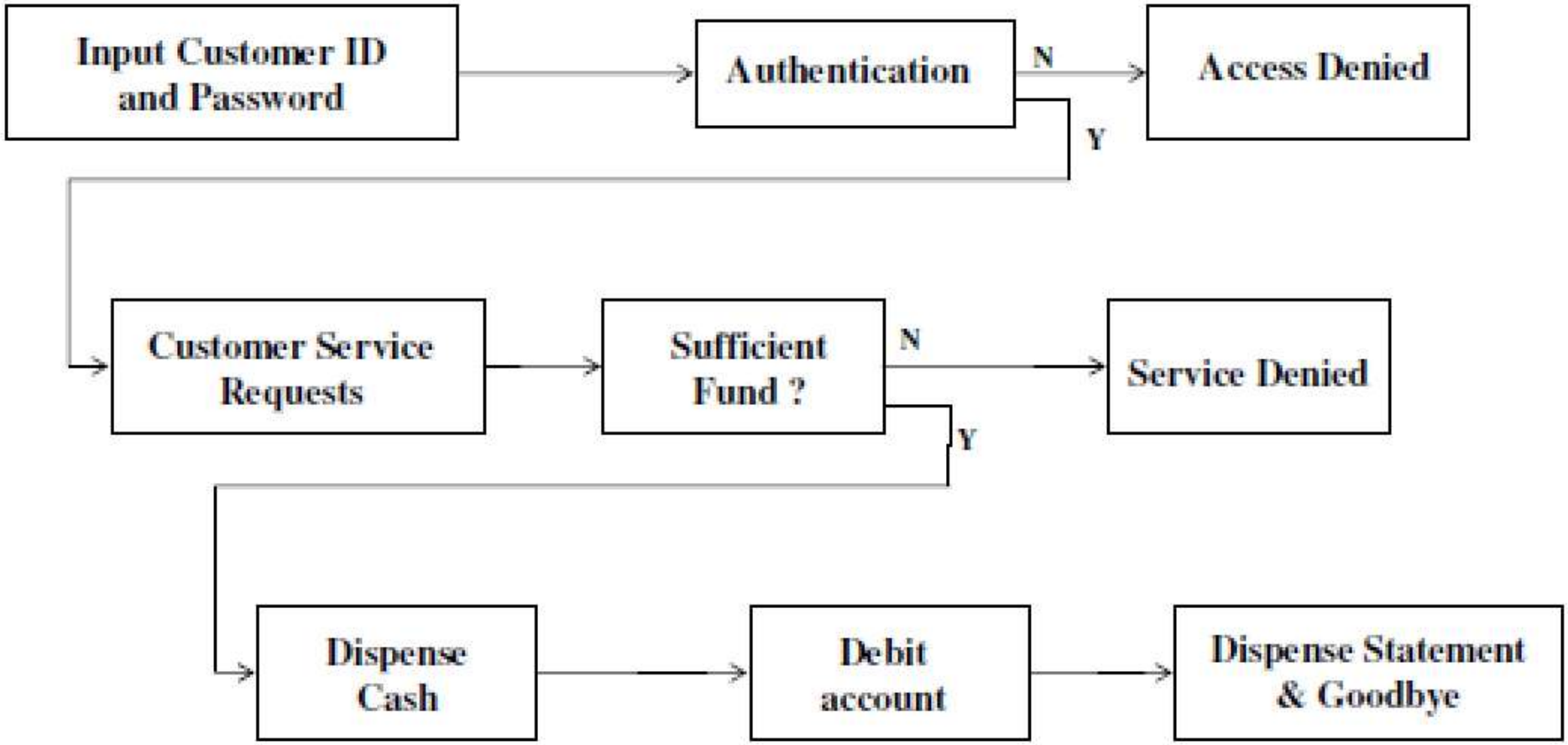
- Necessary and Traceable
- **Implementation-free**
  - **Define what need to be done, not how to do it**
- Provide rationales for requirements
- Concise and unambiguous
- Verifiable and Traceable
- Complete and consistent
- Feasible and affordable

Credit: Dr Pao and INCOSE

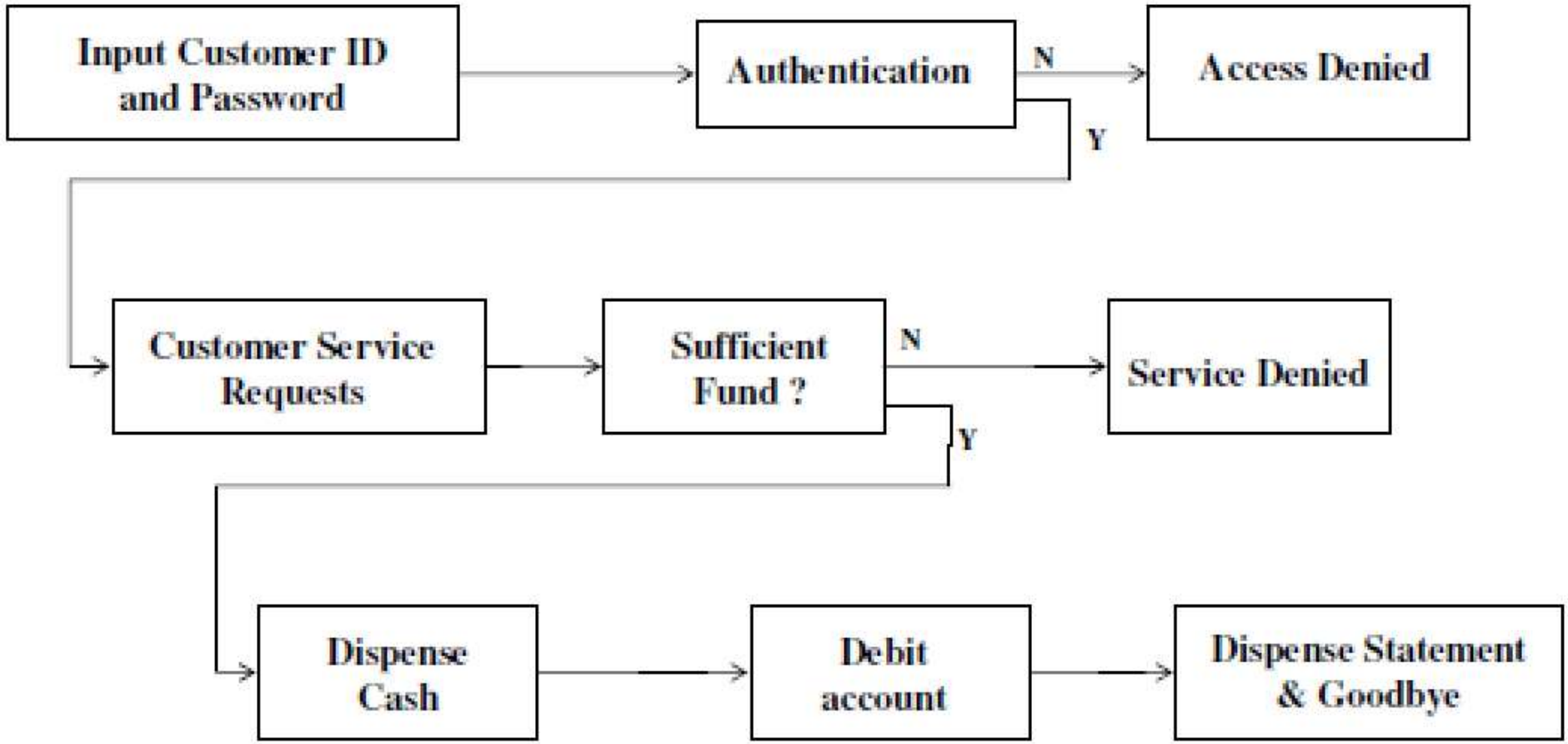
From Conceptual Design Lecture

- Assume you were a system engineer who is responsible to develop an ATM system with just one function:
  - Withdraw cash from customer's account
  - Requirements:
    - System response time for each interface cannot be longer than 30 seconds
    - No over drawn allowed
- Use FFBD to perform function analysis and requirement flow down

FFBD for ATM System (1) – From Conceptual Design Lecture

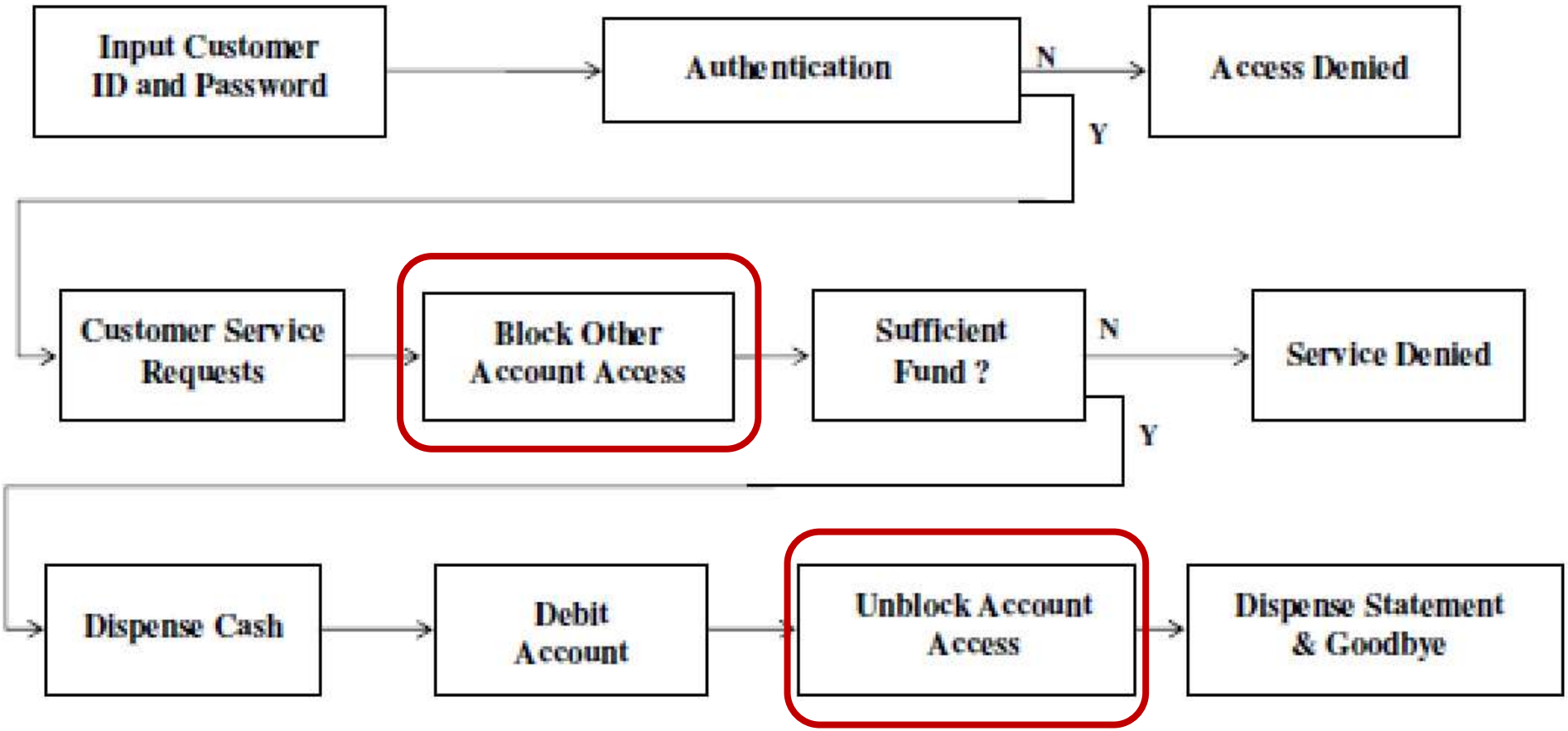


FFBD for ATM System (1)



What if two withdraws are made against one account at the "same time"?

FFBD for ATM System (2)

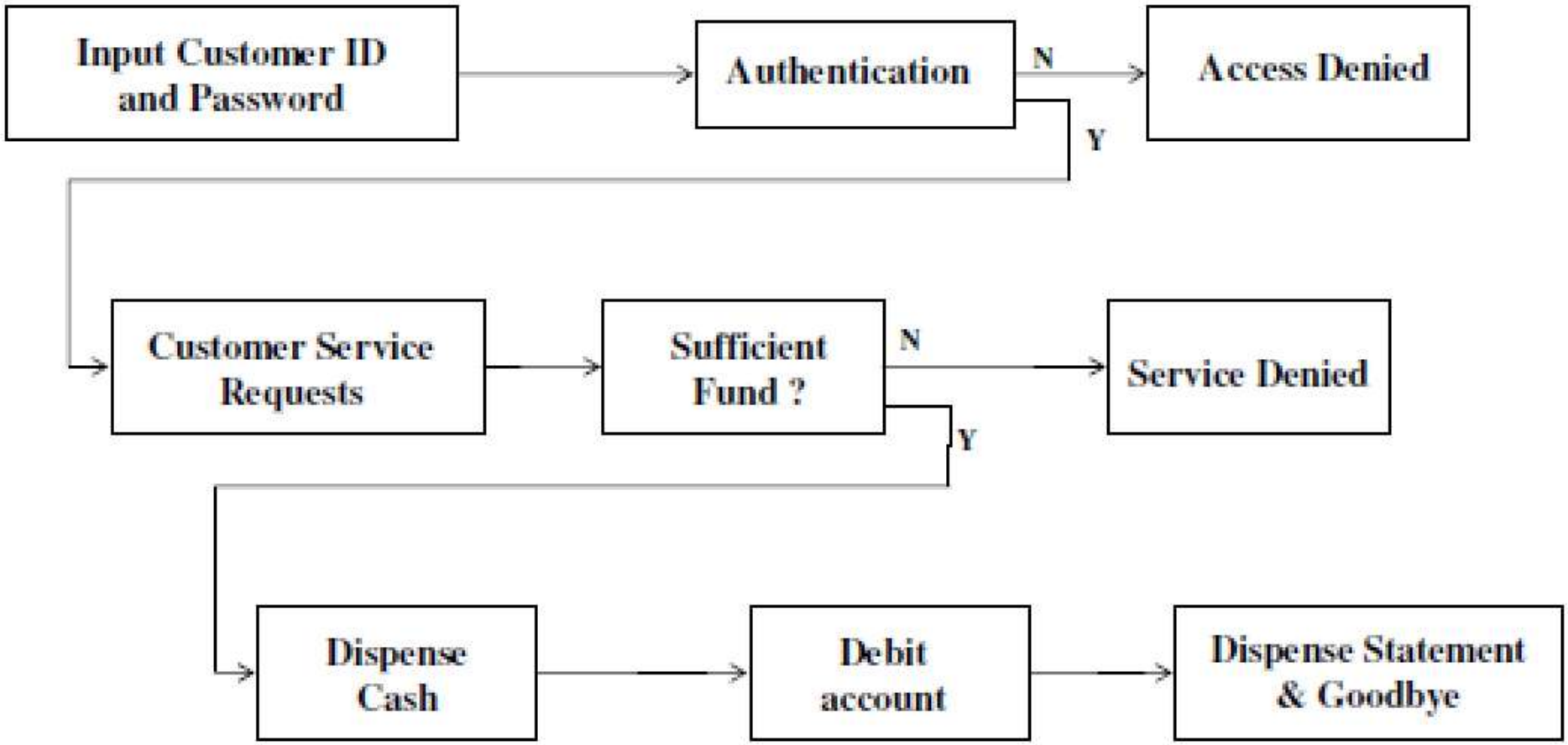


There is a problem with this design! Do you see it?

"Implementation" and "Requirement" don't mix

- Implementation issues should be considered after all requirements are understood
- Pre-matured implementation designs lead to
  - Fragmented solution
  - Added complexities
- Make system integrations, future system improvements and system migrations difficult

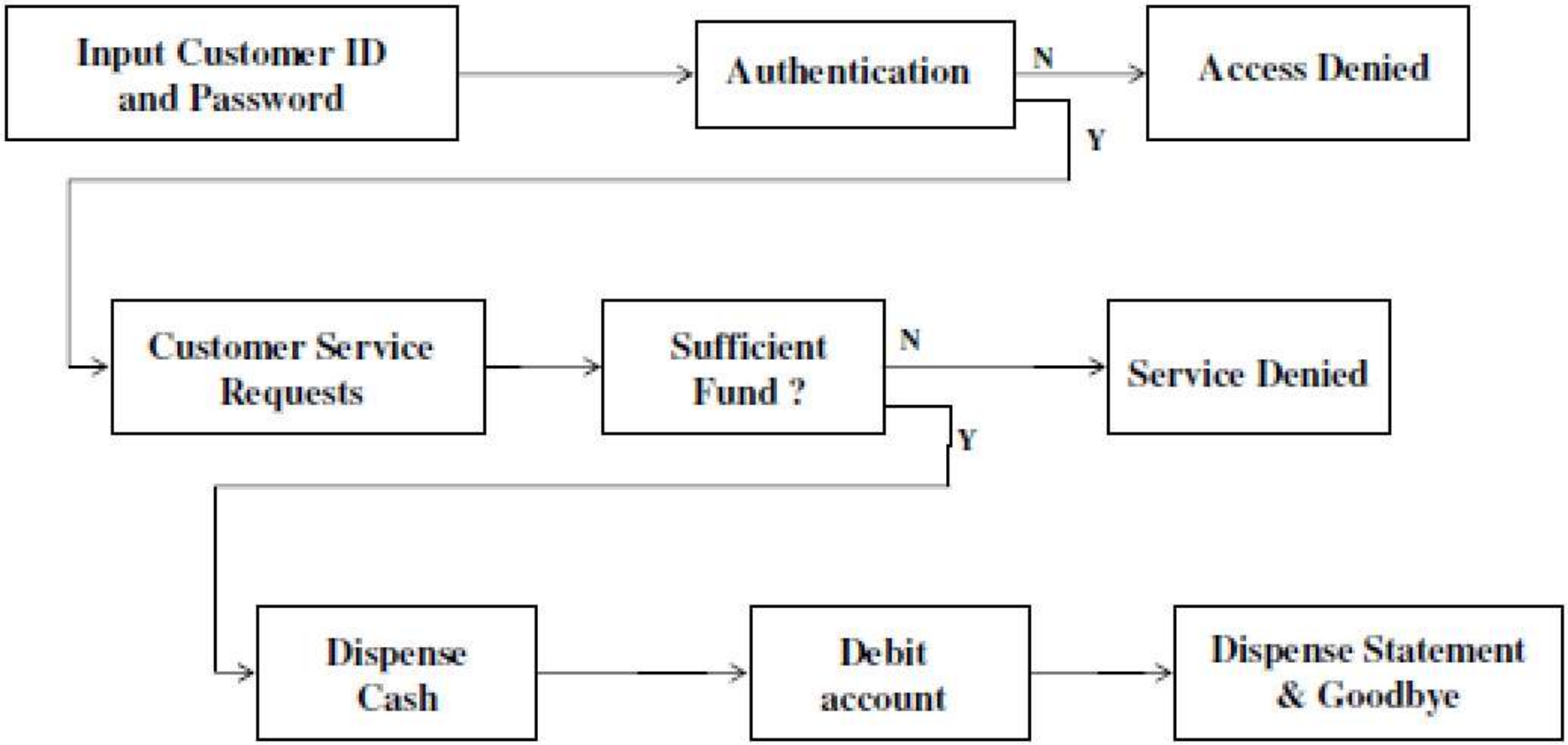
FFBD for ATM System (1)



Requirement Flow Down- ATM System

- **Requirement:** System response time for each interface cannot be longer than 30 seconds
  - (authentication + 2 way interface) <= 30 seconds
  - (sufficient fund? + **dispense cash** + debit account + interfaces) <= 30 seconds
  - Dispense statement & goodbye <= 30 seconds
- (Sufficient funds?): Shall prevent overdraws including possible overdraws caused by nearly simultaneous withdraws

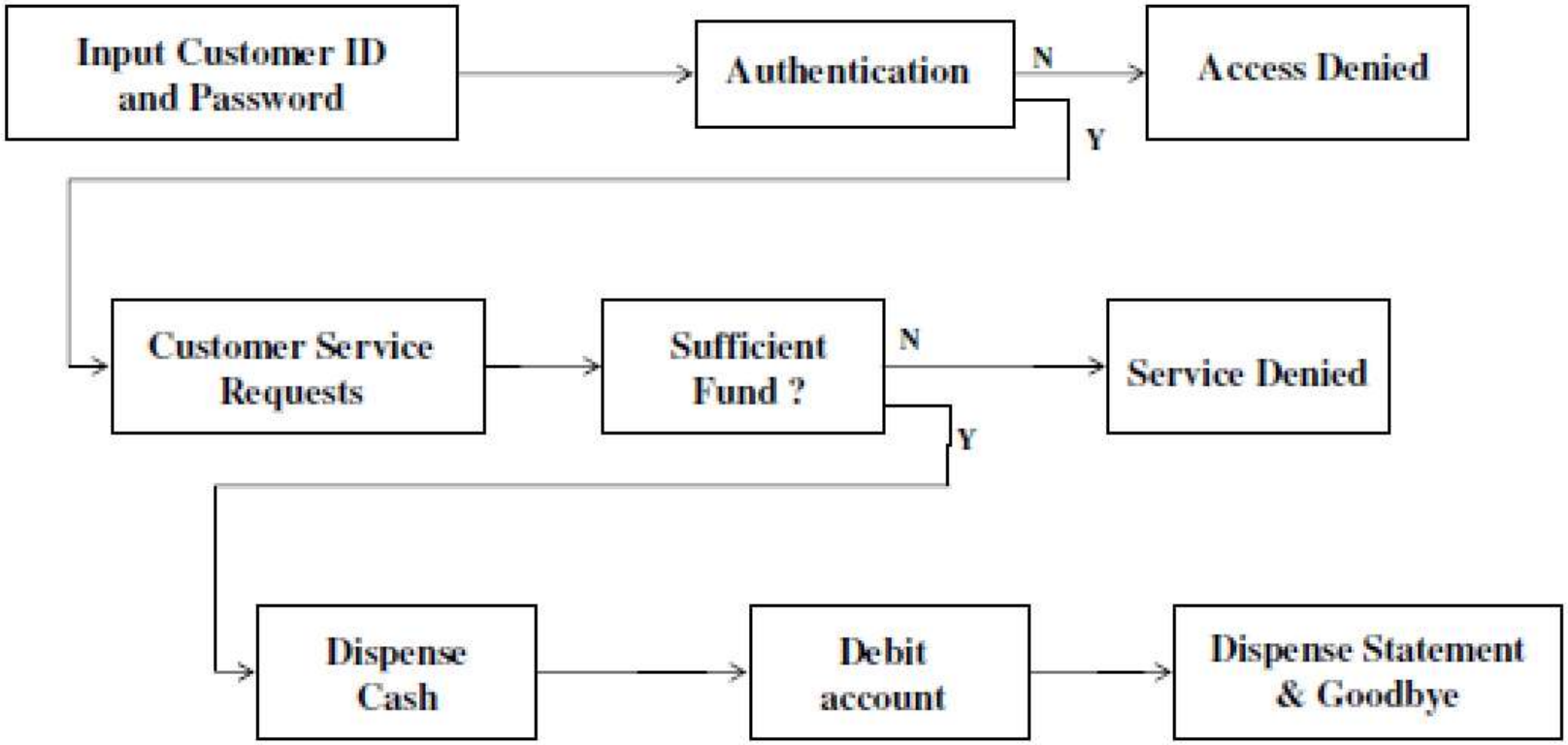
FFBD for ATM System (1)



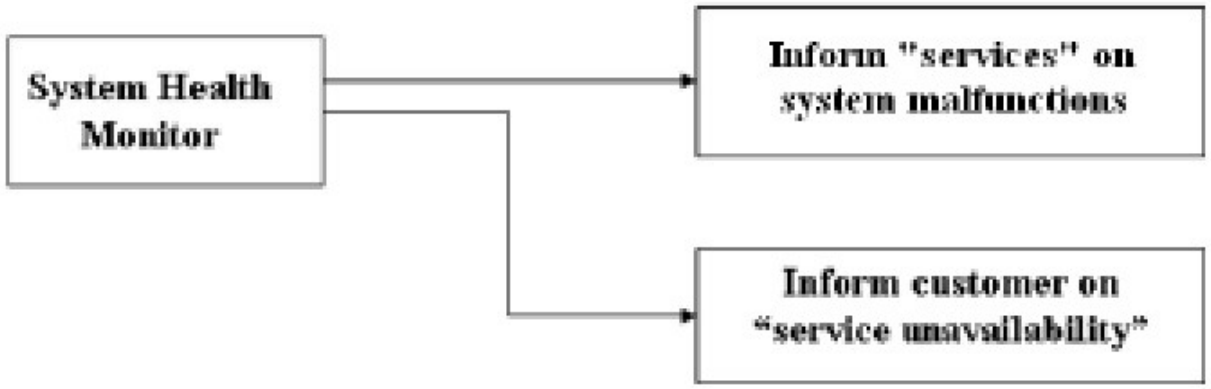
- What if the cash dispensing or other functions are not working properly?
- What if there is not enough cash in the dispenser?

FFBD for ATM System (1)

A



B

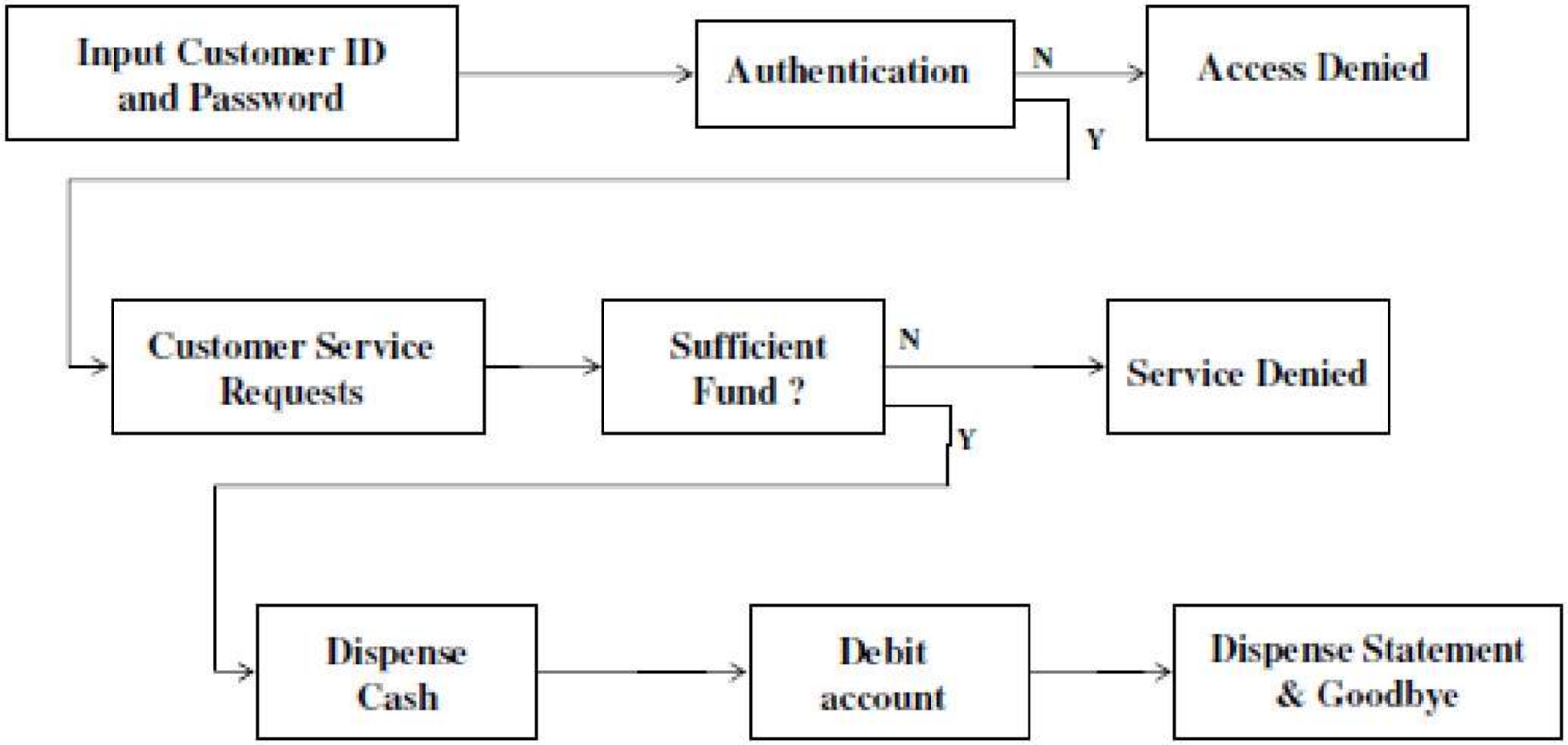


Requirement Flow Down- ATM System

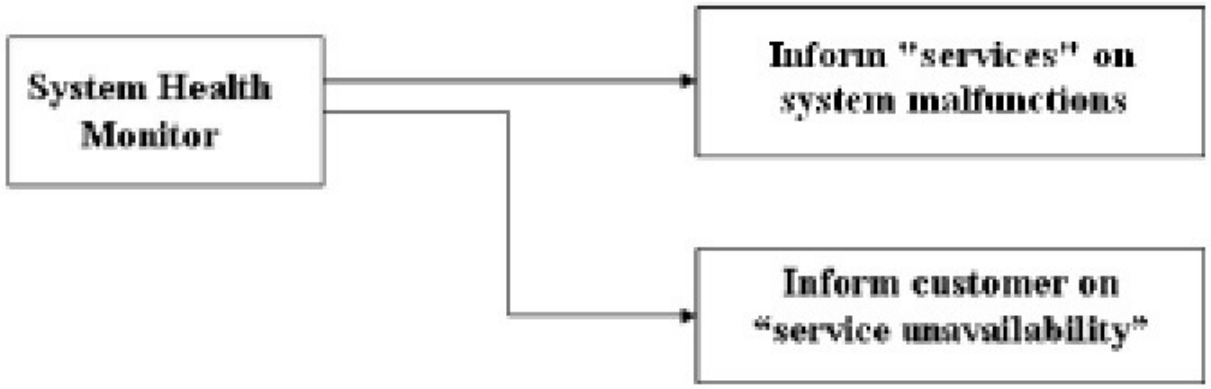
- **Requirement:** System response time for each interface cannot be longer than 30 seconds
  - (authentication + 2 way interface) <= 30 seconds
  - (sufficient fund? + **dispense cash** + debit account + interfaces) <= 30 seconds
  - Dispense statement & goodbye <= 30 seconds
- (Sufficient funds?): Shall prevent overdraws including possible overdraws caused by nearly simultaneous withdraws
- (System Health Monitor): Inform customers about the unavailability of service and inform "system services" regarding the system malfunctions within 30 seconds.

FFBD for ATM System (1)

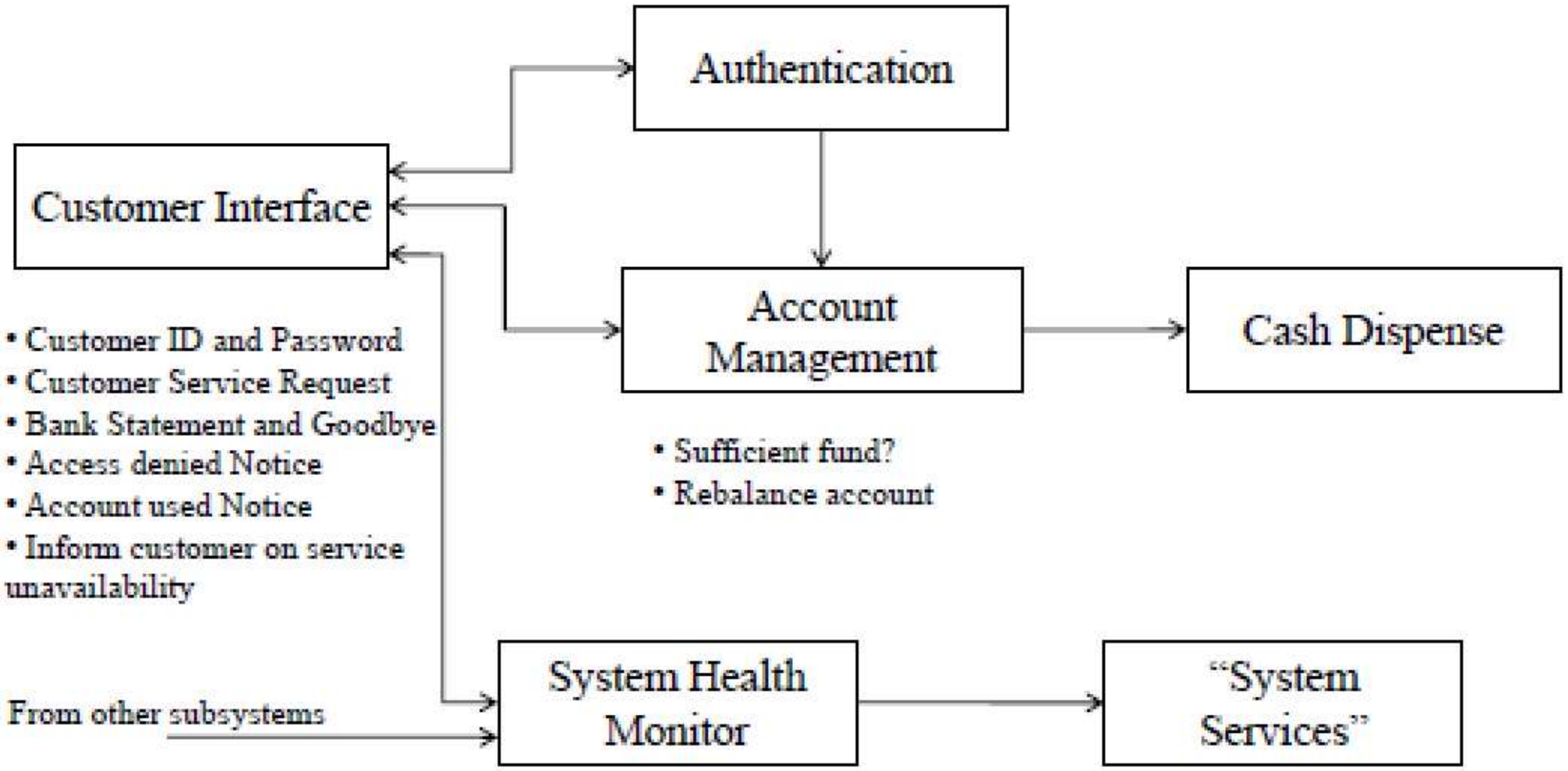
A



B



Preliminary System Physical Architecture for ATM System



Requirement Flow Down-ATM System

- **Requirement:** System response time for each interface cannot be longer than 30 seconds
  - (authentication + 2 way interface) ≤ 30 seconds
  - (sufficient fund? + **dispense cash** + debit account + interfaces) ≤ 30seconds
  - Dispense statement & goodbye ≤ 30 seconds
- (Sufficient funds?): Shall prevent over draws including possible over draws caused by nearly simultaneous withdraws
- (System Health Monitor): Inform customers about the unavailability of service and inform "system services" regarding the system malfunctions within 30 seconds

ATM Subsystem Requirement Flow Down

- Subsystem requirements flow down follows from the subsystem functions flow down and knowledge about these functions
- Most likely "authentication" and "account management" are **existing systems**
  - Performance of these systems should be understood
  - It simplifies the requirements allocations yet it also limits the trade space for other functions
- It is most likely that the "Customer Interface" and "Cash Dispenser" will be done in one "**off the shelf**" equipment
- The requirement flow down of these systems will become a part of the procurement requirement

Other Life Cycle Related Requirements

- Reliability – Measured using MTBF
- Maintenance concept – Example: On site service within x hours
- System Upgrade –Example: Open System interfaces

Class Discussion (about 15 minutes)

- Consider your group project preliminary design
- Design the physical system architecture for this system
- Identify subsystem functions, and flow down performance requirements

Architecture

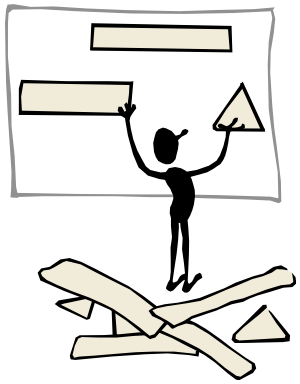
Agenda

- Definitions of Architecture, Views and Types
- Open Architecture
  - Definitions, Standards, Advantages
- DoD Architecture Framework
  - Multiple Views, Compliance

Architecture Definitions

- IEEE STD 610.12
  - Architecture is the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time
- INCOSE Systems Architecture Working Group Definitions
  - Systems Architecture: The fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors
  - System Design: An assemblage of realizable system elements and sub-elements and their interfaces
  - Product Line Architecture: The shared architecture of a related family of systems
- Rechtin
  - The Structure - in terms of components, connections, and constraints - of a product, process or element.

Architecture: a link between Business and Technology



Change

+

Complexity

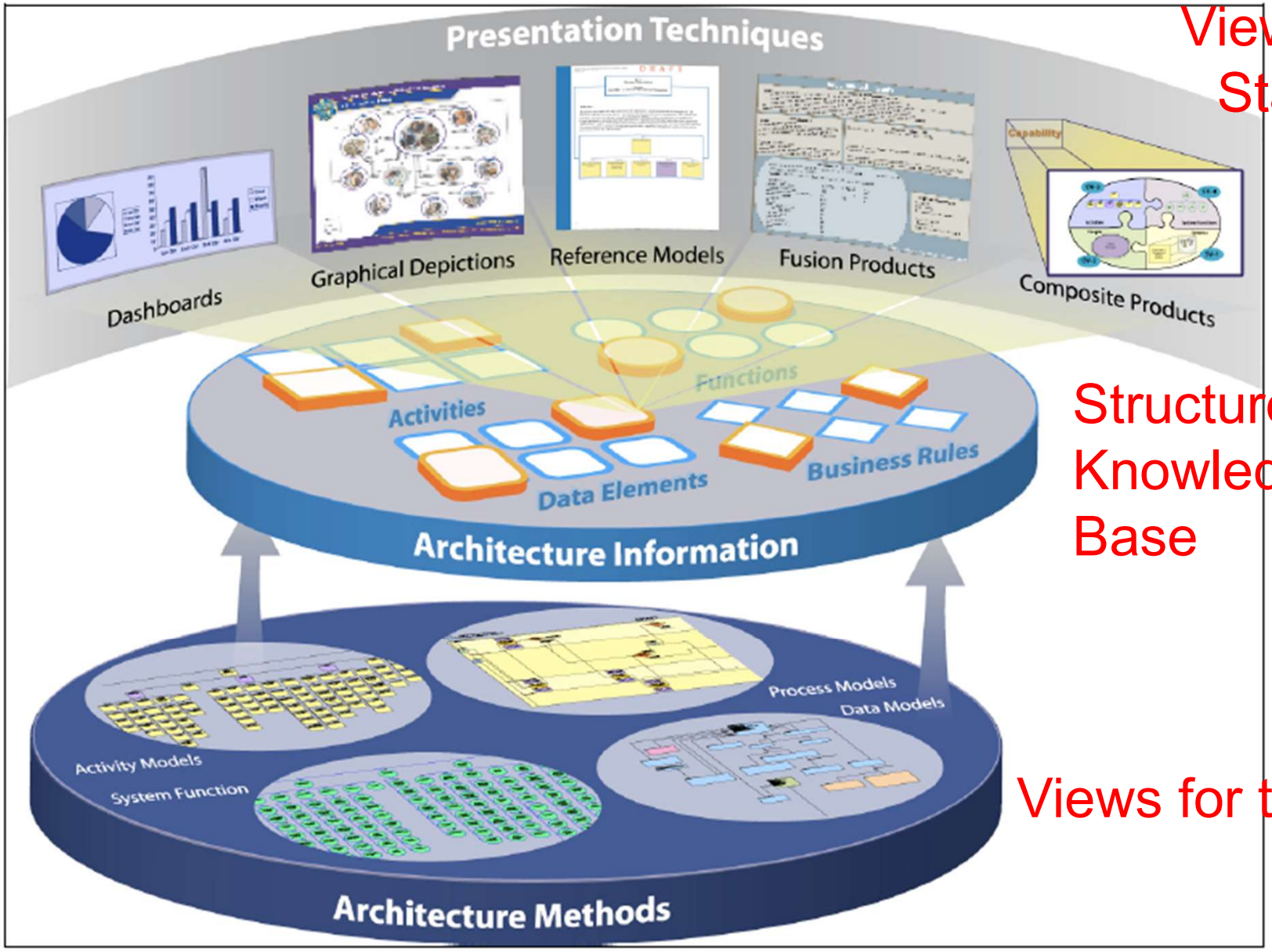
~~Uncertainty~~

Flexibility

Architecture

Source: Systems Architecture Forum

Apply Architecture Methods



Views for Other Stakeholders

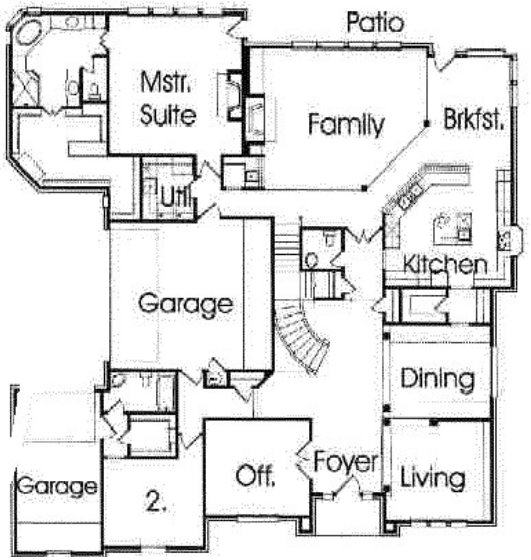
Structured Knowledge Base

Views for the Architect

55

What are Views?

- Architectures are Complex
- One Simple Model or View is inadequate to describe the complexity
- Multiple Views, each emphasizing different characteristics or attributes, is better able to describe overall architecture.



Architectural Views

- **From Rechtin**
  - Purpose/Objective — What the Client Wants
  - Form — What the System is
  - Behavioral or Functional — What the System Does
  - Performance Objectives or Requirements — How Effectively the System Does it
  - Data — Information retained and its interrelationships
  - Managerial — Process for construction and management

- **From DoD Architecture Framework (Version 1)**
  - Operational — Tasks and activities, operational elements, and information flows
  - System — Description, including graphics, of systems and interconnections
  - Technical — Minimal set of rules governing the arrangement, interaction, and interdependence

DoD Architecture Framework V1 – Three Views

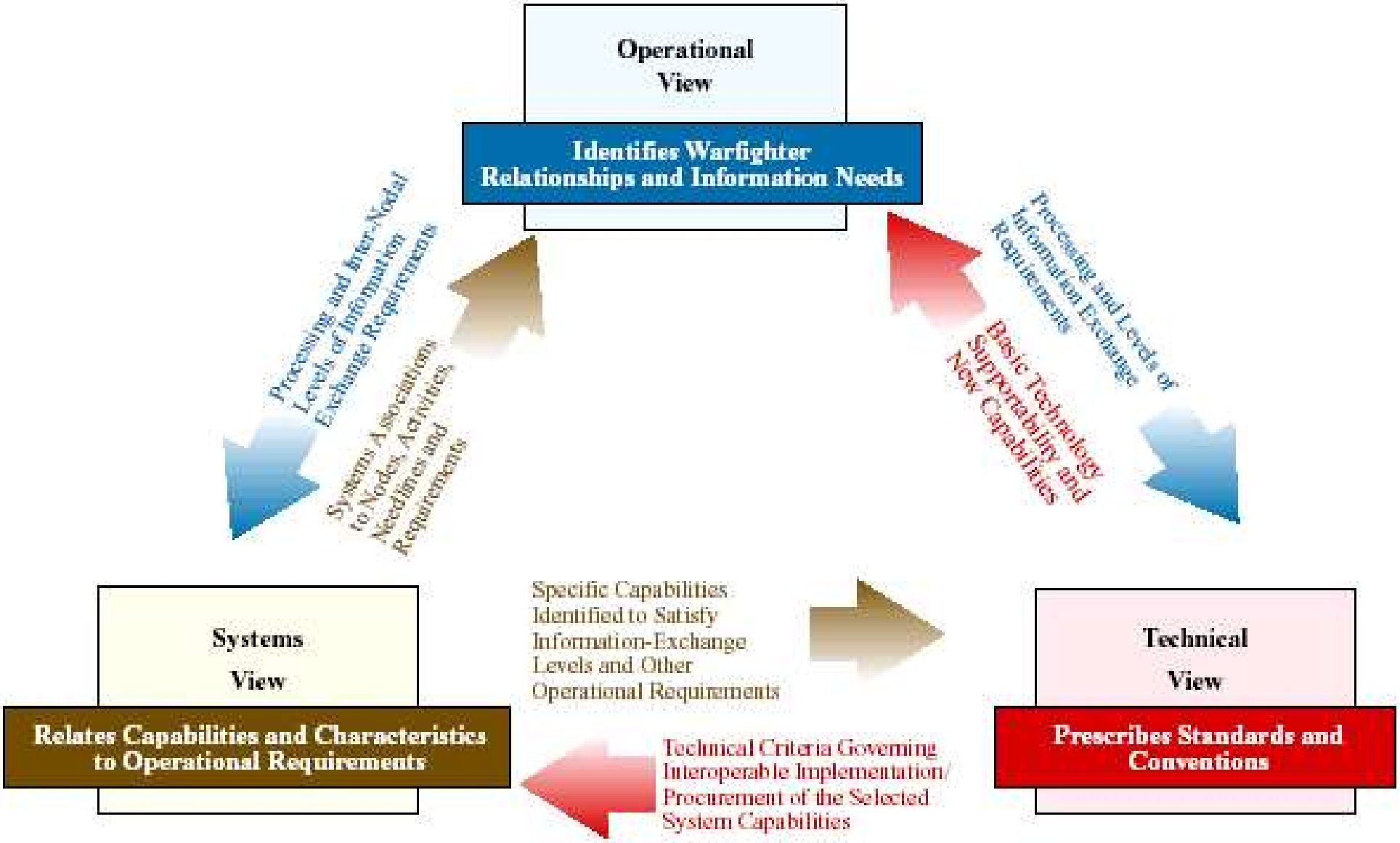
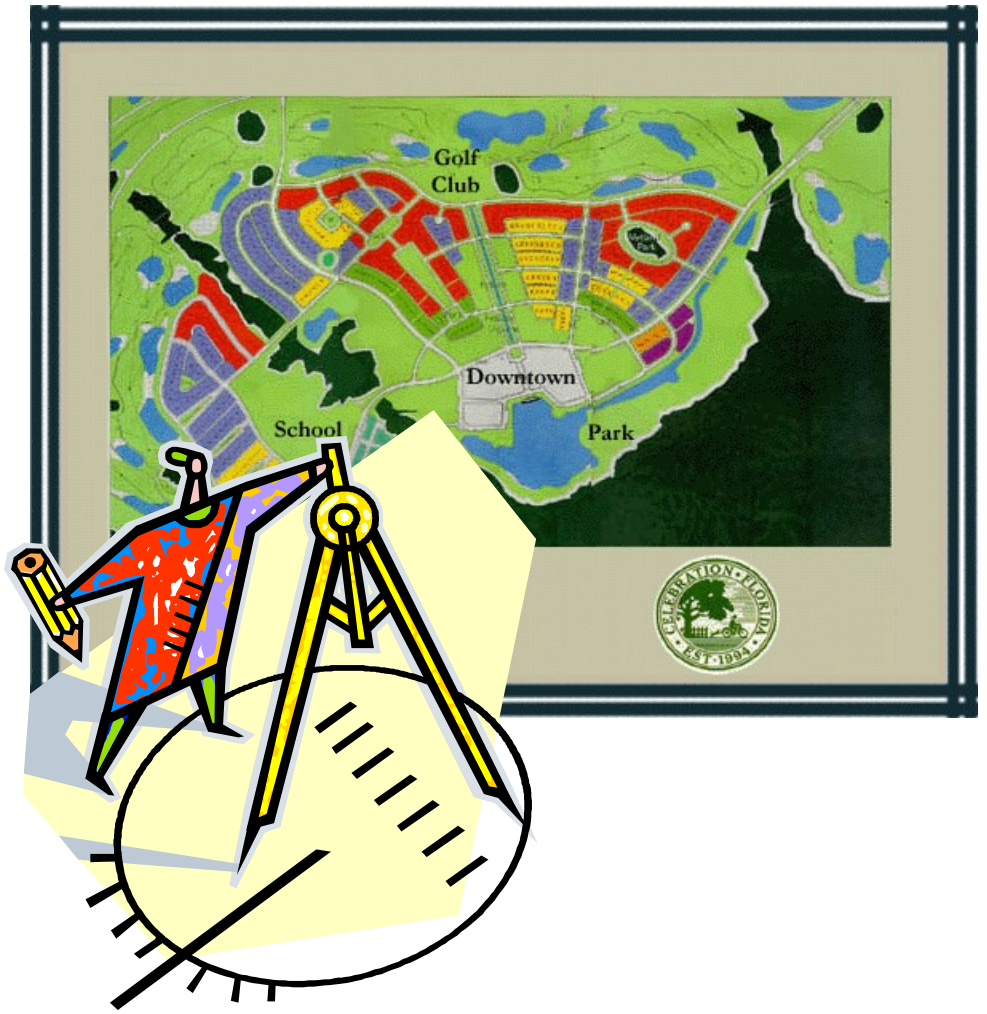


Figure 2-2. Fundamental Linkages Among the Views

Use Urban Planning to Explain the Views

- The "Big Picture"
  - Scope
  - Purpose
  - Intended users
  - Environment
  - "Context"



Source: Randy Case

Operational Viewpoint (OV)

- A community operates as a place to:
  - Work, shop, entertain
  - Raise a family
  - Help neighbors
  - Call home
- The community's citizens use the Operational Viewpoint
- For the citizens, the community's infrastructure is taken for granted; the systems and services are expected to seamlessly interact and inter-operate

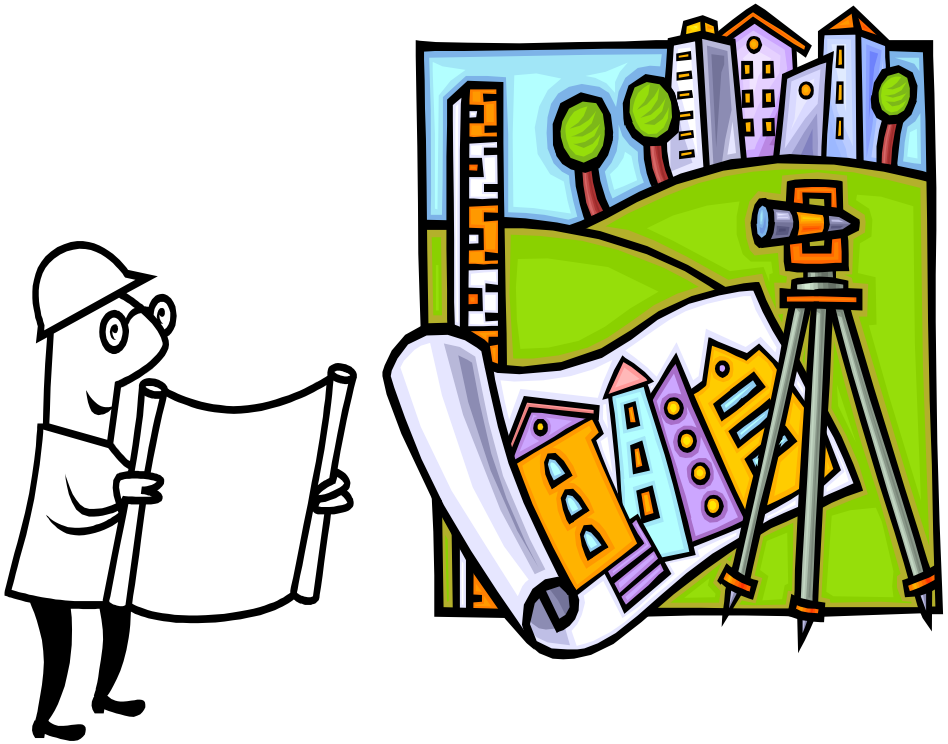
Source: Randy Case

Operational Architecture View

- The operational architecture view is a **description of the tasks and activities**, **operational elements, and information flows** required to **accomplish or support a intended system operation**
- It contains: operational elements, assigned tasks and activities, and information flows
- It defines: types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges
- Essential Information:
  - Customer Input
  - Operational Requirements
  - Operational Functional Flows (or Use Cases by Operation)
  - Definition of Elements (or Objects)
  - Allocation of Functions to Elements
  - Information Flow Between Elements

Systems Viewpoint

- A community is a system of systems:
  - Electrical
  - Water & sewer
  - Communications
  - Roads
- The community's builder uses the Systems Viewpoint
- For the builder the community's infrastructure is the main focus; the various systems and sub-systems must be carefully specified, designed, and installed



Source: Randy Case

Systems Architecture View

- The systems architecture view is a **description**, including graphics, of **systems and interconnections** providing for, or supporting, intended functions
- For a domain, the systems architecture view shows how multiple systems link and inter-operate
- For the individual system, the systems architecture view includes the physical connection, location, and identification of key nodes (including materiel item nodes), circuits, networks, platforms, etc., and specifies system and component performance parameters (e.g., mean time between failure, maintainability, availability).
- The systems architecture view associates physical resources and their performance attributes to the operational view and its requirements per standards defined in the technical architecture.
- Essential Information:
  - Block Diagrams, Interconnect Definition
  - Allocated Requirements
  - Physical Element Definition (Design)

BUILDING AN ARCHITECTURE IS LIKE BUILDING A HOUSE…

Technical Architecture
is the building code that contains technical specifications

Operational Architecture
is the architectural rendering with all the capabilities and functionality laid out in detail

System Architecture
is the building blueprint

Source: Army Futures Center, TRADOC

System architecture is like a house
Enterprise architecture is more like an urban design

Technical Architecture View

- The technical architecture view is the **minimal set of rules governing** the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.
- The technical architecture view provides the technical systems-implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed.
- The technical architecture view includes a collection of the technical standards, conventions, rules and criteria organized into profile(s) that govern system services, interfaces, and relationships for particular systems architecture views and that relate to particular operational views.
- Essential Information:
  - Architecture Requirements and Standards

Hardware vs. Software

- Hardware Architectures are Becoming Simpler, while Software Architectures are Becoming More Complex
- Hardware:
  - Network Connections and Power at the System Integration Level
  - Complex Information Flow, Once Allocated to Analog and Multiple Digital Lines is Now in Software
  - Well Known Subsystem Architectures from Experience (e.g. Radars and Missiles)
- Software:
  - Architecture is Highly Dependent Upon Data Structures and Software Implementation Approach
  - Typical Software ICD Runs for Thousands of Pages
  - Diagrams are Complex and Hard to Reduce

DoD Architecture Framework

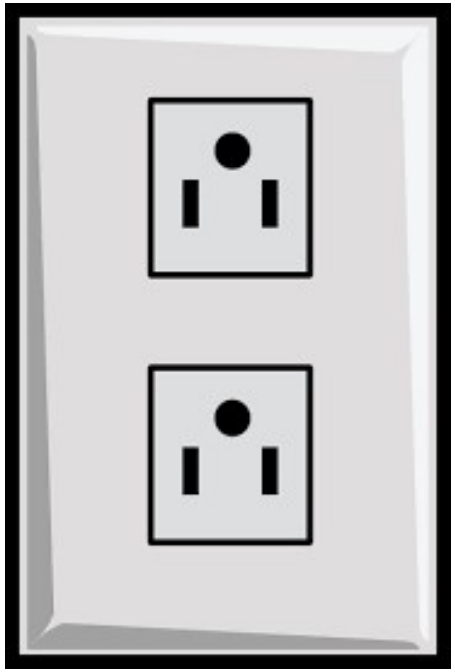
- Version 2 Added additional views
  - Capability – capabilities to be delivered and their relationship to operations and services
  - Data and Information – conceptual, logical and physical data models
  - Project – managing the project, timeline, and capabilities
  - Services – identification, description, relationship of services provided
- All Views – scope and purpose

Types of Architectures (Taxonomy)

- Systems of Systems (Domain)
  - Independently Developed and Managed, Distributed, Multi-Platform
- Distributed Systems
  - Multi-Platform at Geographically Separate Locations
- Platforms
  - Vehicles
    - Ships, Aircraft, Space Vehicles (e.g. Satellites), Trucks, Trailers
  - Buildings
    - Power Plant, Factory, Command Center, etc.

Definition: Open Systems

- Implements sufficient **open specifications** for interfaces, services, and supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and to interact with users in a style that facilitates portability



Key Characteristics of Open Systems

- Well-defined, widely used, **non-proprietary** standard interfaces, services, formats
- Standards that are developed by industrially recognized **standards bodies**
- Definition of system interfaces to facilitate adding new or additional capabilities
- Explicit **provision for expansion or upgrading** through incorporation of additional or higher performance elements with minimal impact on the system
- User portability

Advantages to Using Open Systems

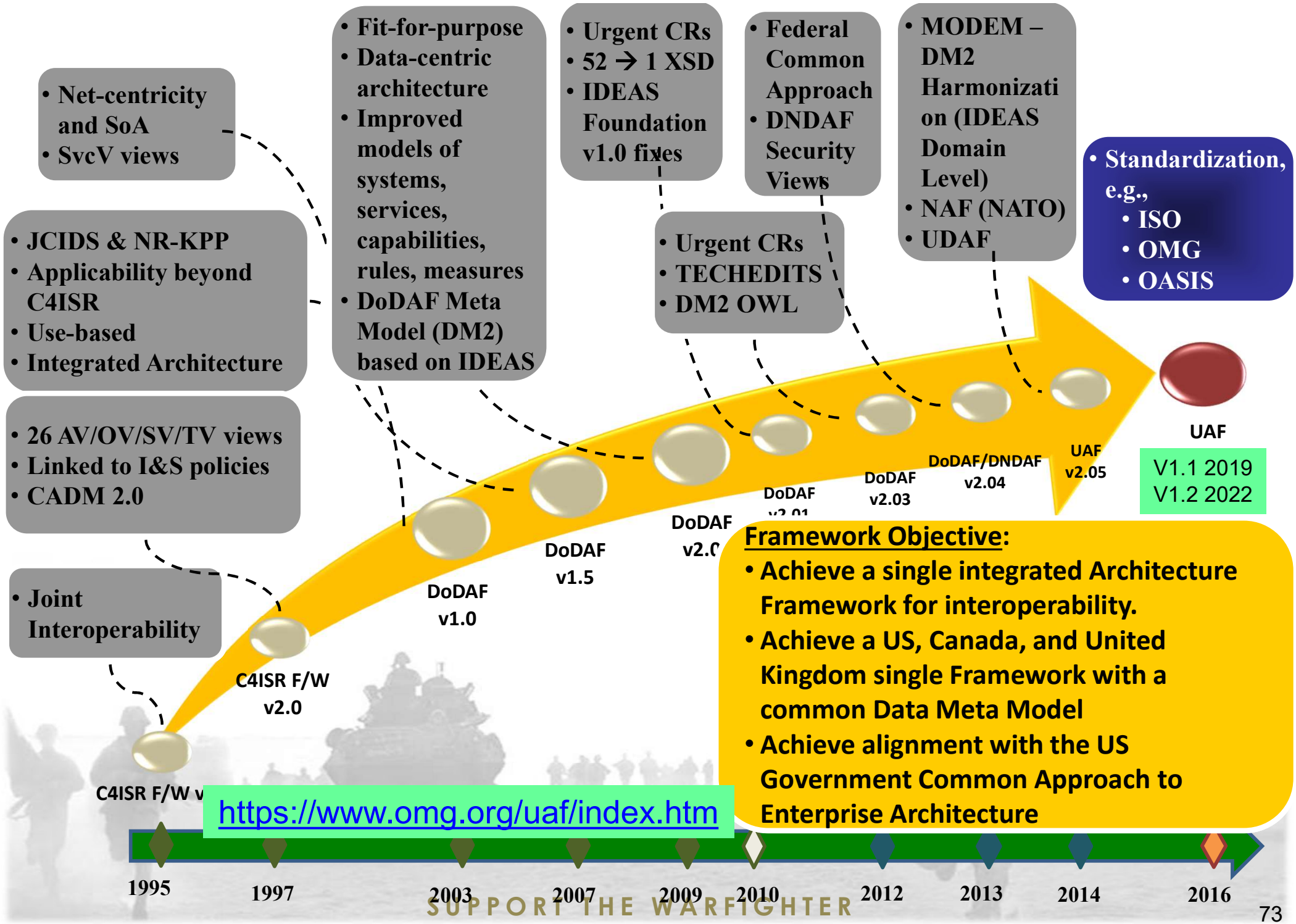
- Less Reliance on Proprietary Products
- More Competition Leading to Lower Cost
- Decreased Probability of Schedule Delay
- Better Tested Products (More Users)
- Portable Applications
- Interoperability
- Faster Technology Insertion
- Foundation for System Evolution

All Systems Should be Designed with Open System Architectures

Many Architecture Frameworks

- ATAM (Architecture Trade off and Analysis Method)
- Zachman Framework
- Department of Defense (US) Framework – DoDAF
- Ministry of Defense (UK) Framework – MoDAF
- The Object Management Group Framework – ToGAF
- Federal (US) Enterprise Architecture Framework – FEAF

Overall architecture framework convergence vision



Getting Good Architects

- In the Past it was All On the Job Training (OJT) with Experienced Experts
  - History provided the examples and lessons learned
  - Mostly hardware, technology limited the solutions
- Today Training is Required, Particularly in the Tools and the Methods
  - Systems engineering and software tools provide methods for constructing and examining architectures, must have expertise in these tools
  - Must understand the government and industry standards (more training)
  - Must understand the software methods for databases, distributed processing, and fault tolerance (more training)
- Finally, as Always, You can't be Good at Something Until You've Learned the Hard Lessons of Experience
  - Still need OJT

Knowing When It's Good

- "I'll Know it When I see It!" – don't recommend this approach, but for the experts it may work
- First Define the Characteristics of Good
  - Reliable, Fault Tolerant, Easy to Upgrade, Easy to Maintain, Runs Fast, Low Cost - What is the Definition of Good?
  - A Quality Function Deployment (QFD) Approach May be Required When the Customer Does Not Emphasize a Particular Aspect or Two
- Define Metrics that Relate Good to the Architecture at Hand
  - Implement Metrics in the Tools, so that as the Architecture and the Alternatives are Defined they Can be Compared Based on Goodness
  - Select the Architecture that Has the Best Goodness Metric
- Use the SEI Architecture Tradeoff Analysis Method (ATAM) Process to Accomplish this Goal

Backup

Architecture Considerations in Future Collaborative Information Sharing

DoDAF Version 2.03 Updates Information Sharing

for

NDIA Architecture Committee

18 April 2012

Walt Okon

Senior Architect Engineer

Architecture & Interoperability Directorate

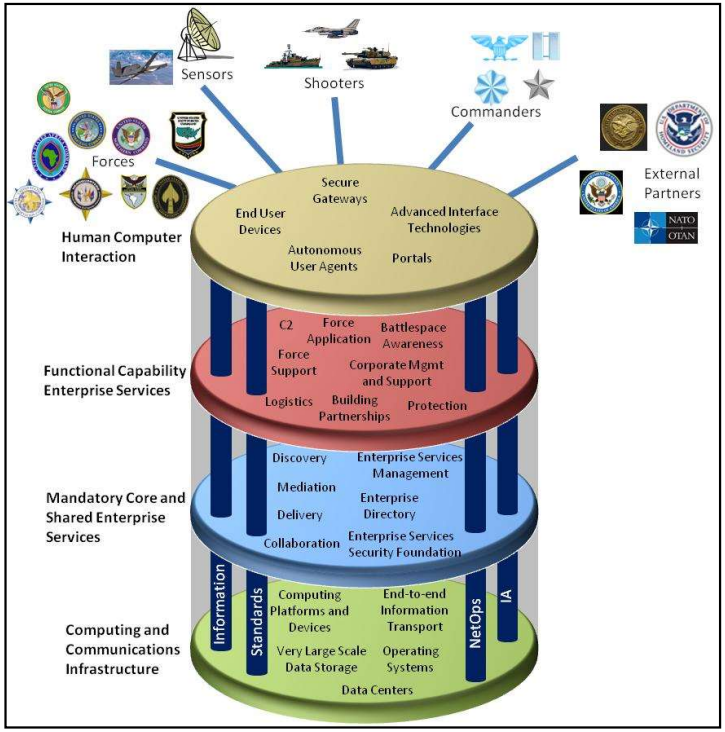
Office of the Secretary of Defense

E-Mail: walt.okon@osd.mil

Elements of Quality Architecture

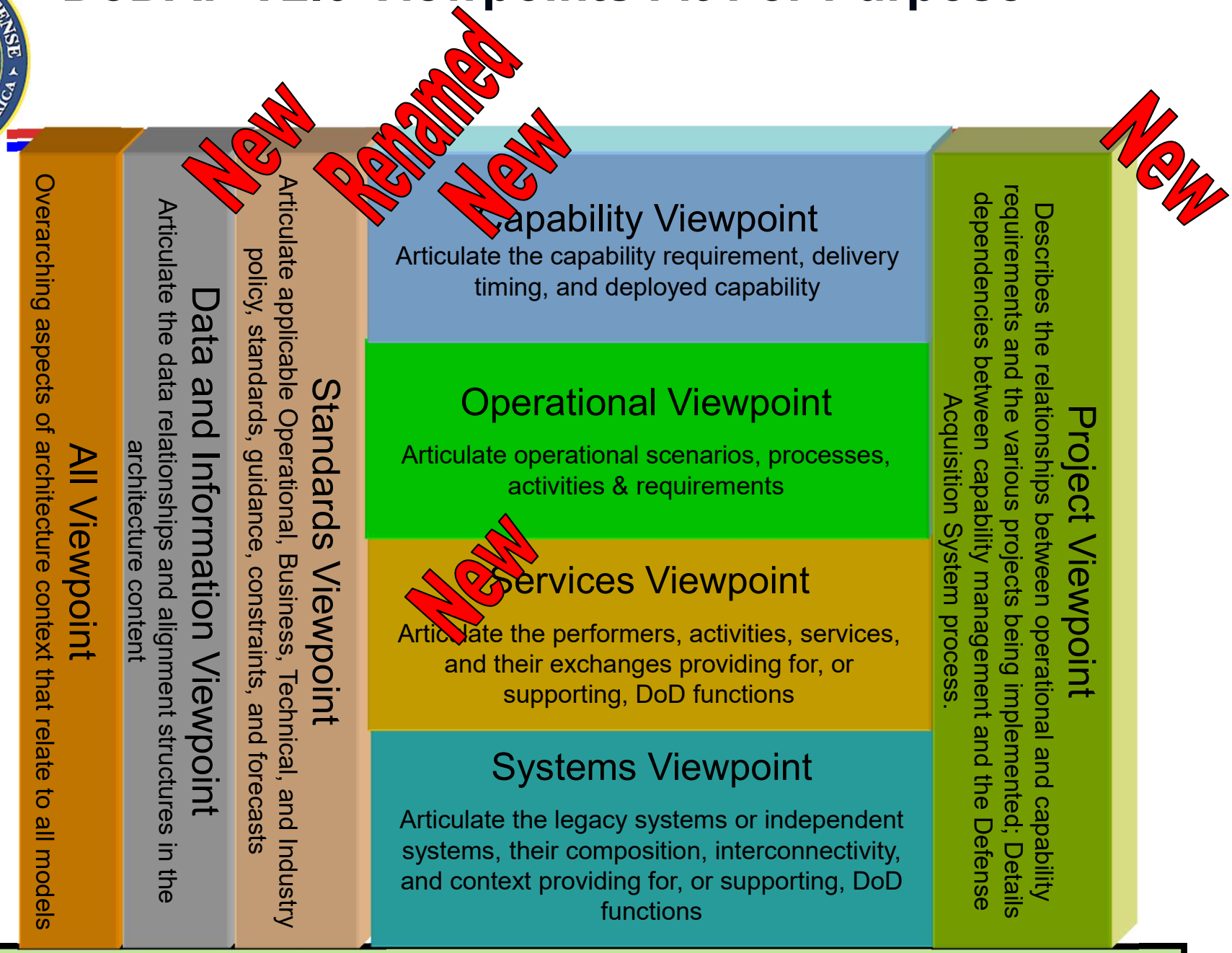
- *Single Architecture Framework*
- *Policy, Direction, Guidance*
- *Exchange - Standards*
- *Architecture Tools*
- *Certified Architects*

Enabling efficient and effective acquisition of hardware, software and services used by DoD and Partners in mission performance.



Unified Defense Architecture Framework

DoDAF V2.0 Viewpoints Fit-For Purpose



Architecture viewpoints are composed of data that has been organized to facilitate understanding.

Week 5 Homework 1

- Please read the paper
  - Enterprise Architecture: System & Management Framework for NYCT Vital Systems
- No need to turn in the homework
  - This is the mid terms week