CptS 487
Software Design and Architecture

Lesson 21

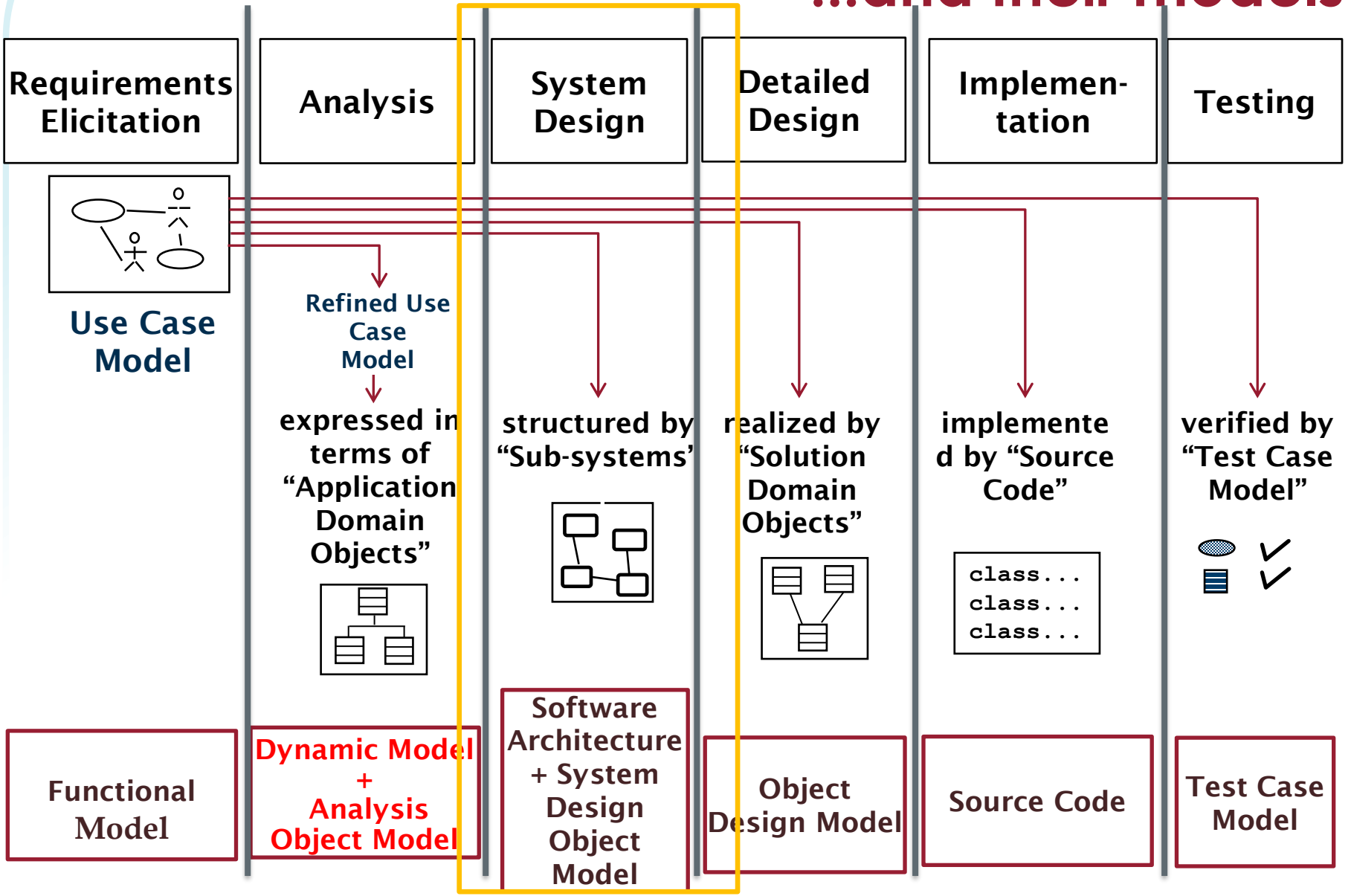
Key Issues in System Design 1:

Hardware/Software Mapping

WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Instructor: Bolong Zeng

Software Lifecycle Activities
...and their models



Overview

System Design I

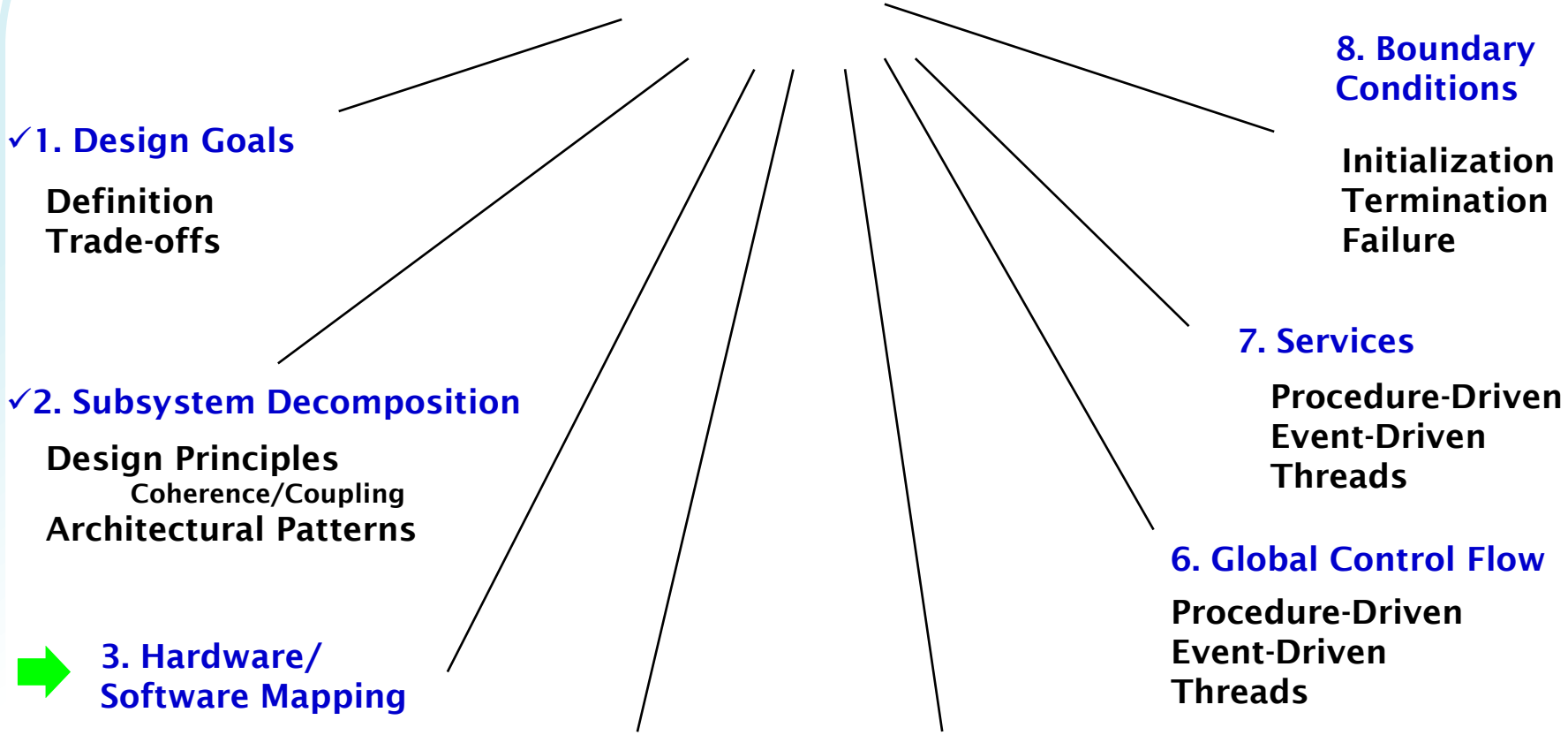
- ✓ 0. Overview of System Design
- 1. Design Goals
- 2. Subsystem Decomposition

  Architectural Patterns

System Design II

- 3. Hardware/Software Mapping
- 4. Persistent Data Management
- 5. Providing Access Control
- 6. Designing the Global Control Flow
- 7. Identifying Services
- 8. Identifying Boundary Conditions

System Design



3. Hardware/Software Mapping

- This system design activity addresses two questions:
  - How shall we realize the subsystems: With hardware or with software?
  - How do we map the object model onto the chosen hardware and/or software?
    - Mapping the Objects:
      - Processor, Memory, Input/Output
    - Mapping the Associations:
      - Network connections

Mapping Objects onto Hardware

- Control Objects -> Processor
  - Is the computation rate too demanding for a single processor?
  - Can we get a speedup by distributing objects across several processors?
  - How many processors are required to maintain a steady state load?
- Entity Objects -> Memory
  - Is there enough memory to buffer bursts of requests?
- Boundary Objects -> Input/Output Devices
  - Do we need an extra piece of hardware to handle the data generation rates?
  - Can the desired response time be realized with the available communication bandwidth between subsystems?

Hardware-Software Mapping Difficulties

- Much of the difficulty of designing a system comes from addressing externally-imposed hardware and software constraints
  - Certain tasks have to be at specific locations
    - Example: Withdrawing money from an ATM machine
  - Some hardware components have to be used from a specific manufacturer
    - Example: To send DVB-T signals, the system has to use components from a company that provides DVB-T transmitters.

Hardware/Software Mappings in UML

- A UML component is a building block of the system. It is represented as a rectangle with a tabbed rectangle symbol inside

- The Hardware/Software Mapping addresses dependencies and distribution issues of UML components during system design.

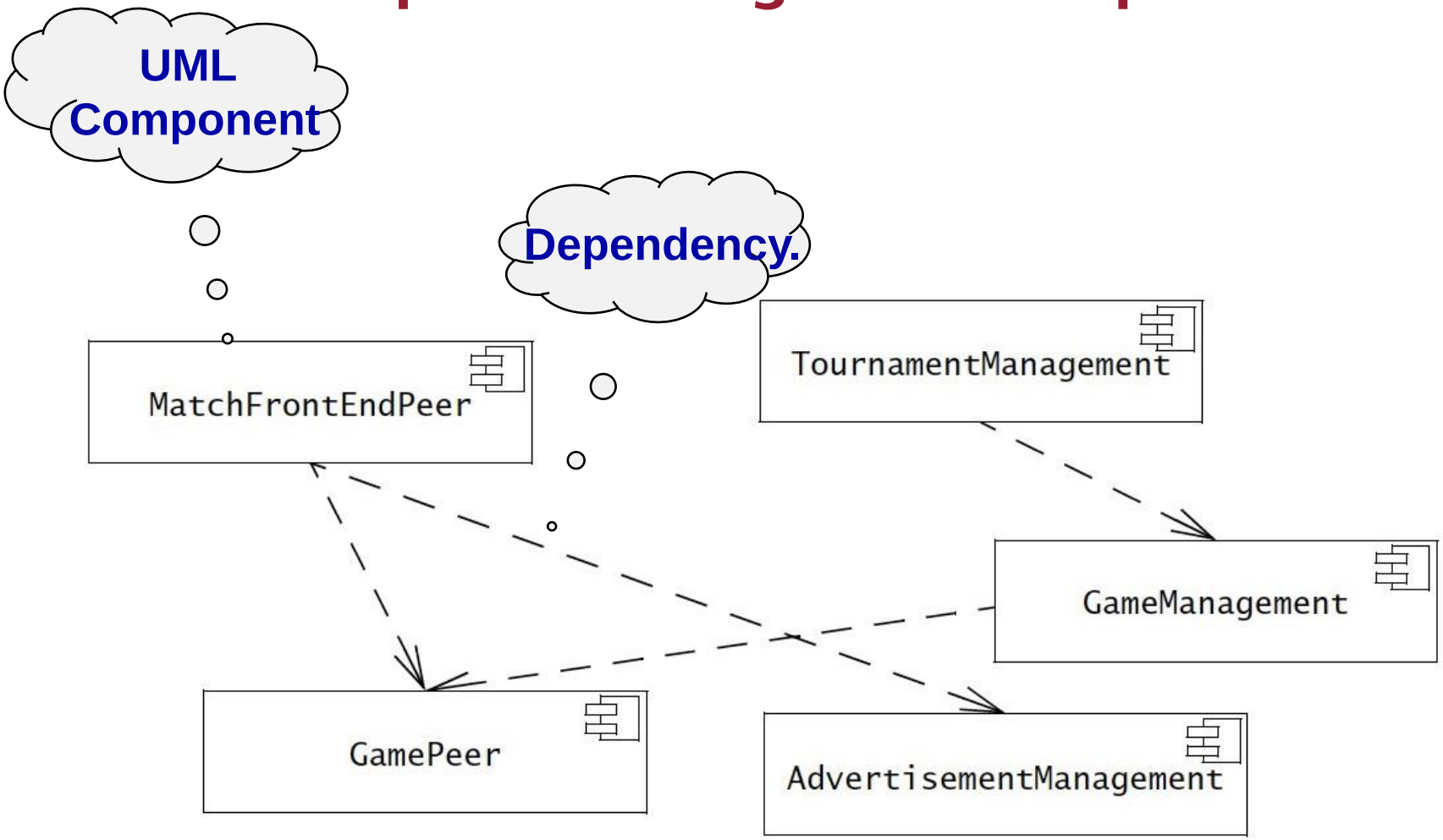
UML Diagram Types

- Deployment Diagram:
  - Illustrates the distribution of components at run-time.
  - Deployment diagrams use nodes and connections to depict the physical resources in the system.
- Component Diagram:
  - Illustrates dependencies between components

UML Component Diagram

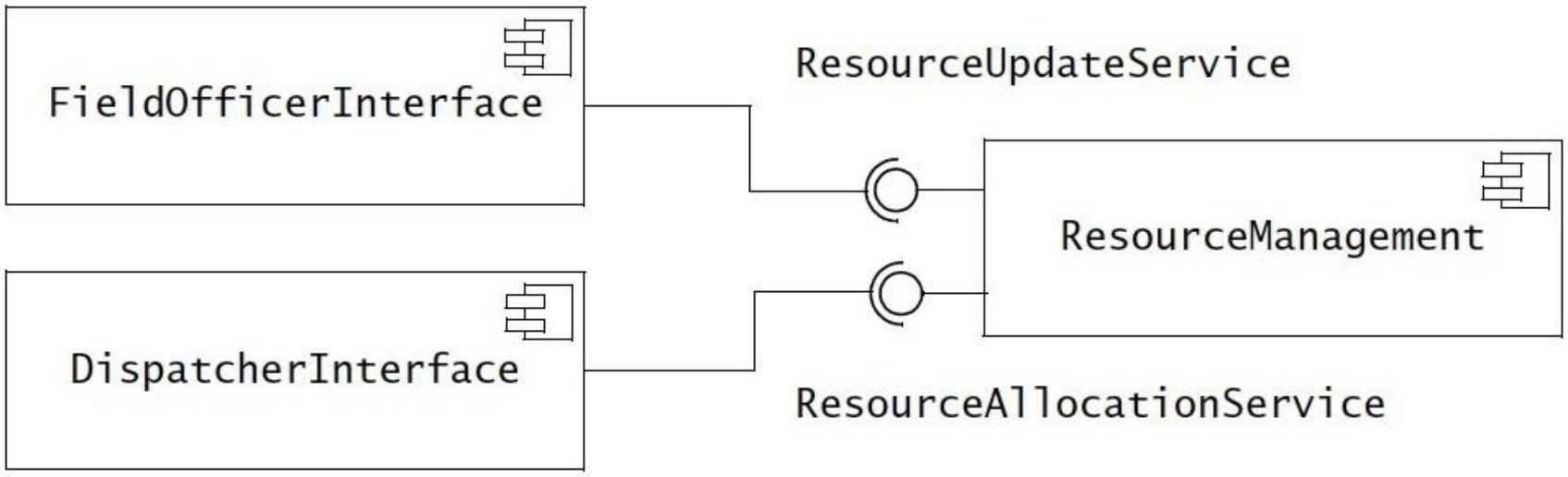
- Used to model the top-level view of the system design in terms of components and dependencies among the components.
- The dependencies (edges in the graph) are shown as dashed lines with arrows from the client component to the supplier component:
- Informally also called “software wiring diagram“ because it show how the software components are wired together in the overall application.

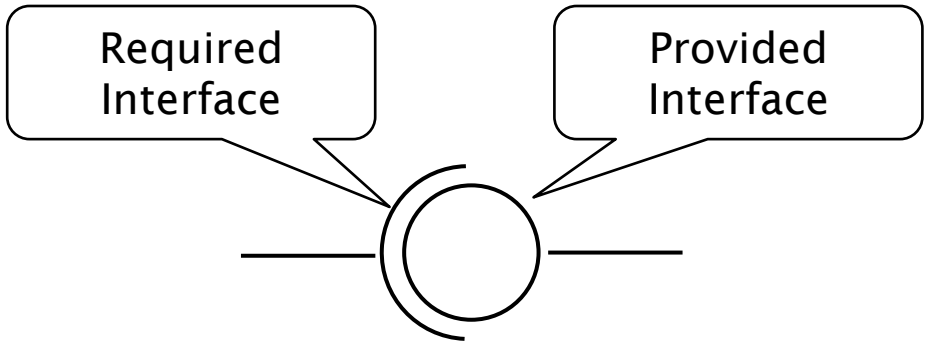
Component Diagram Example



- ARENA Subsystem Decomposition (UML Component Diagram)

Services and Subsystem Interfaces

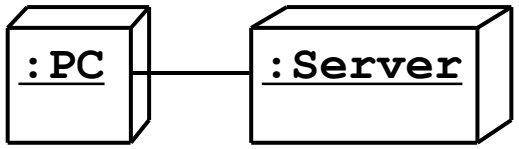




- Ball-socket notation showing provided and required interfaces

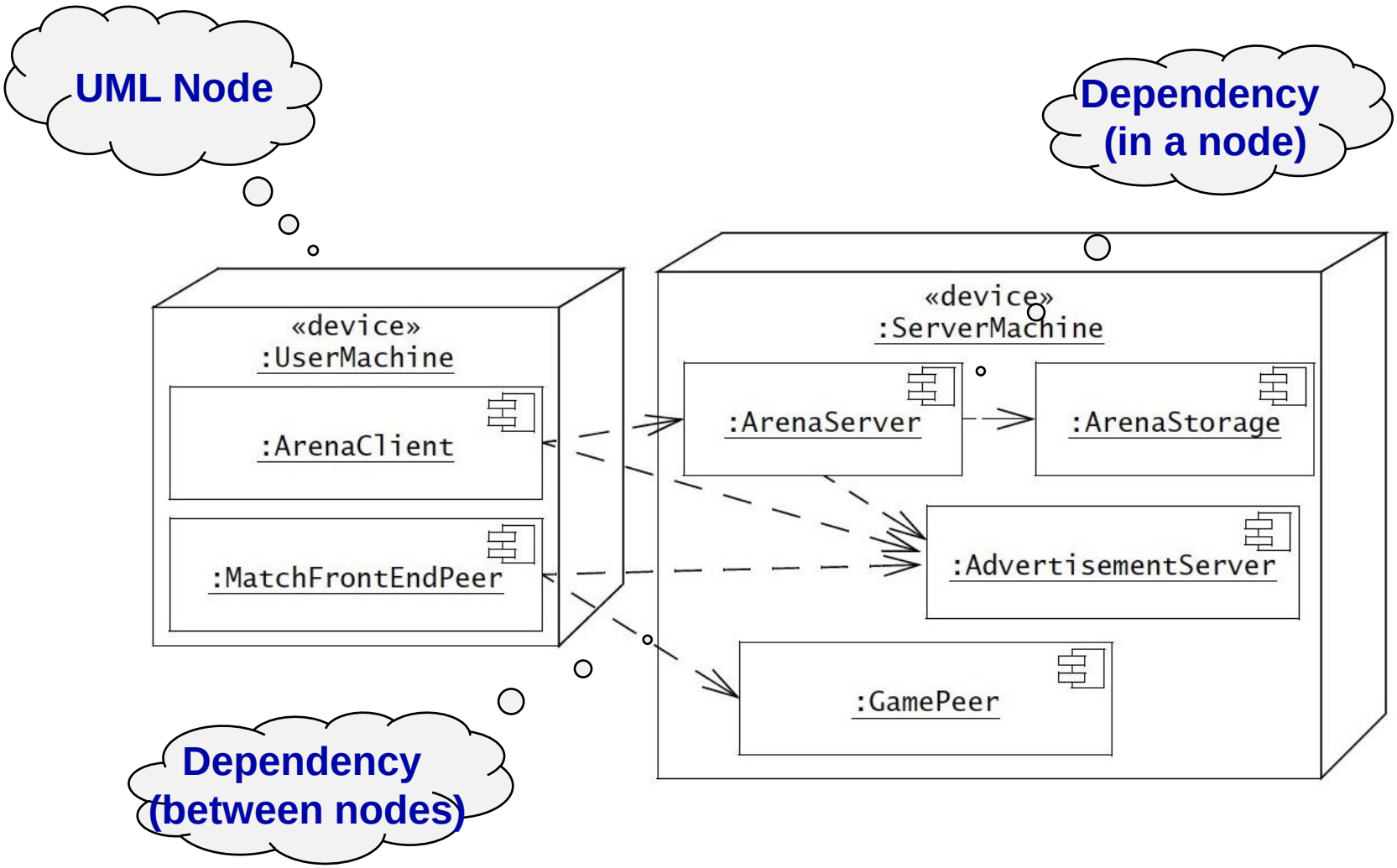
Deployment Diagram

- A deployment diagram is a graph of hardware nodes and connections ("communication associations")
  - Nodes are shown as 3-D boxes
  - Connections between nodes are shown as solid lines
  - Nodes may contain components
    - Components may contain objects (indicating that the object is part of the component).



- Deployment diagrams are useful for showing a system design after these system design decisions have been made:
  - Subsystem decomposition
  - Hardware/Software Mapping

Deployment Diagram Example



- ARENA Hardware/Software Mapping (UML Deployment Diagram)