

CSE347

Information System Analysis and Design

Nishat Tasnim Niloy

Lecturer

Department of Computer Science and Engineering

Faculty of Science and Engineering

Topic: 10

Deployment Diagram

Why We Need Deployment Diagrams

- What existing systems will system need to interact or integrate with?
- How robust does system need to be (e.g., redundant hardware in case of a system failure)?
- What and who will connect to or interact with system, and how will they do it?
- What middleware, including the operating system and communications approaches and protocols, will system use?
- What hardware and software will users directly interact with (PCs, network computers, browsers, etc.)?
- How will you monitor the system once deployed?
- How secure does the system need to be (needs a firewall, physically secure hardware, etc.)?

Deployment Diagrams

- Show the structure of the run-time system
- Capture the hardware that will be used to implement the system and the links between different items of hardware
- Model physical hardware elements and the communication paths between them
- Plan the architecture of a system
- Document the deployment of software components or nodes

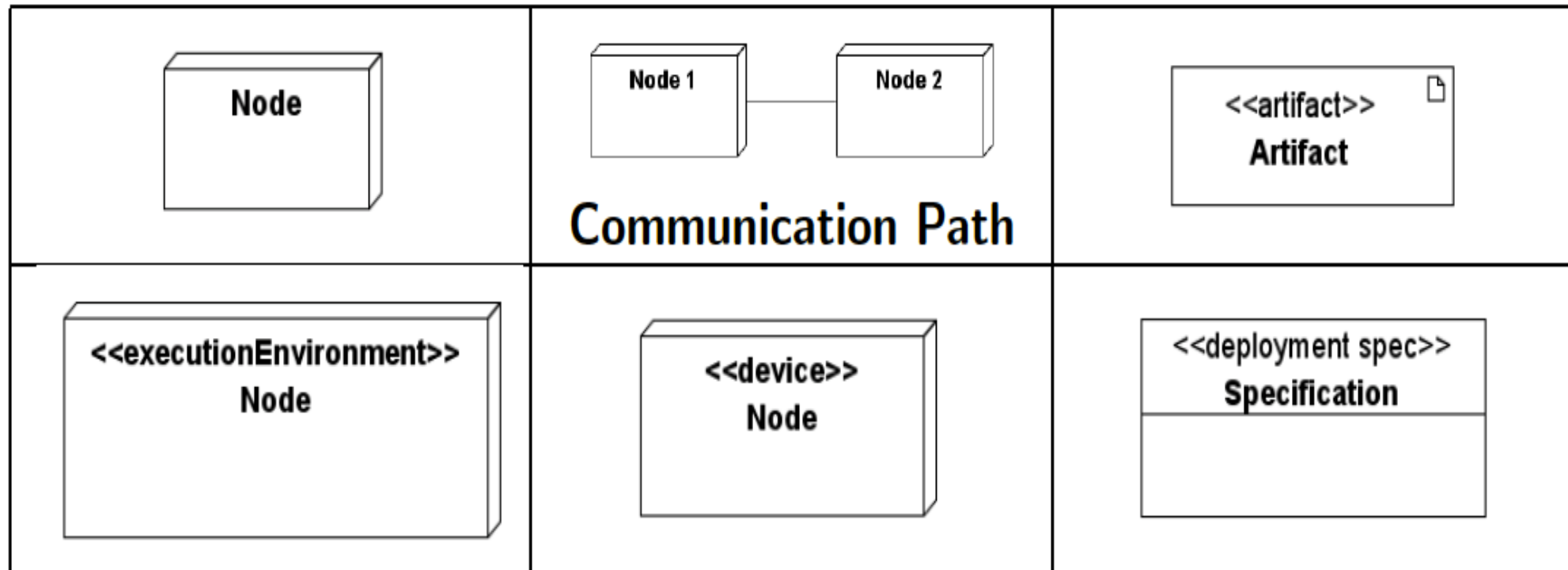
Deployment Diagrams

- Show the configuration of run-time processing elements and the software components, processes, and objects
 - Software component instances represent run-time manifestations of code units
- Capture only components that exist as run-time entities
- Show the systems hardware, the software installed on that hardware, and the middleware that connects the disparate machines together
- A collection of one or more deployment diagrams with their associated documentation
- Show the physical configurations of software and hardware

What is a Deployment Diagram?

- Deployment Diagram – a diagram that shows the physical relationships among software and hardware components in a system
 - Components – physical modules of code
 - Connections – show communication paths
 - Dependencies – show how components communicate with other components
 - Nodes – computational units, usually a pieces of hardware
- Show physical deployment of artifacts on nodes.
- To describe a web site, for example, a deployment diagram would show
 - what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server),
 - what software components ("artifacts") run on each node (e.g., web application, database), and
 - how the different pieces are connected (e.g. JDBC, REST, RMI).

Deployment Diagrams



DEPLOYMENT DIAGRAMS

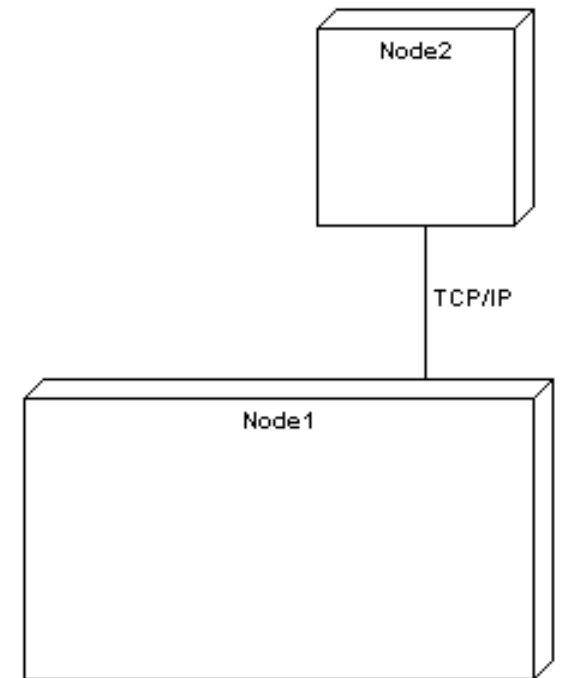
- There is a strong link between components diagrams and deployment diagrams
- Deployment diagrams
 - Show the physical relationship between hardware and software in a system
 - Hardware elements:
 - Computers (clients, servers)
 - Embedded processors
 - Devices (sensors, peripherals)
 - Are used to show the nodes where software components reside in the run-time system

DEPLOYMENT DIAGRAMS

- Deployment diagrams specify constructs that can be used to define:
 - ❖ the execution architecture of systems and
 - ❖ the assignment of software artifacts to system elements.
- Nodes in deployment diagrams represent either hardware devices or software execution environments.
- Artifacts are deployed over nodes
- Some item of information that is used or produced by a software development process or by operation of a system. Examples: model files, source files, scripts, executable files, database tables, development deliverables, word processing documents, and mail messages.

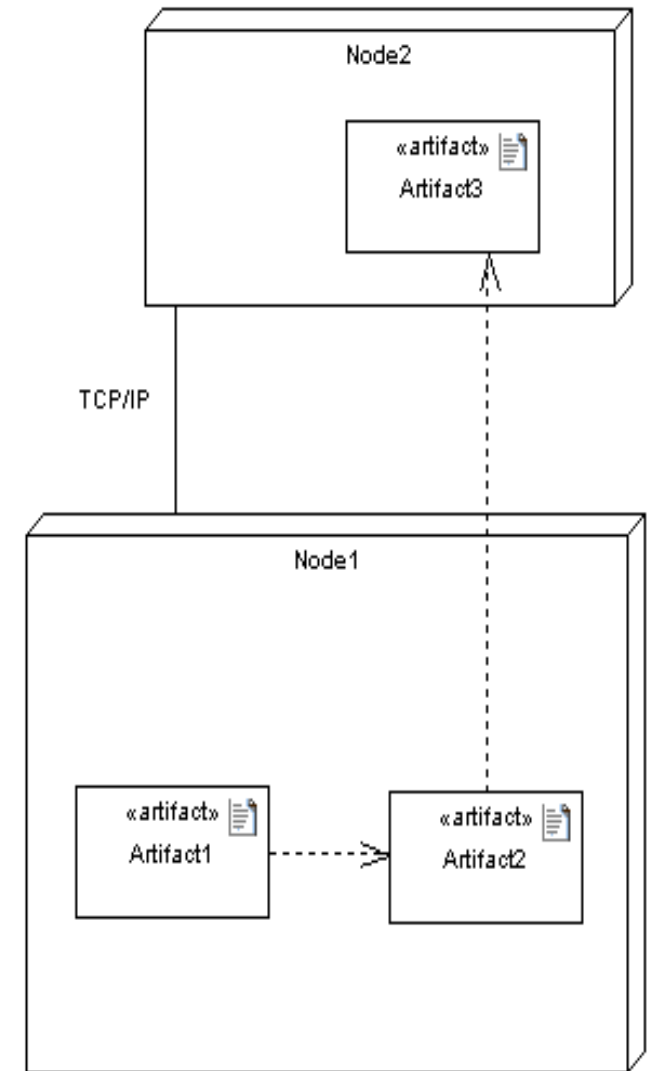
DEPLOYMENT DIAGRAMS

- Deployment diagram
 - Contains nodes and connections
 - A node usually represent a piece of hardware in the system
 - A connection depicts the communication path used by the hardware to communicate
 - Usually indicates the method such as TCP/IP

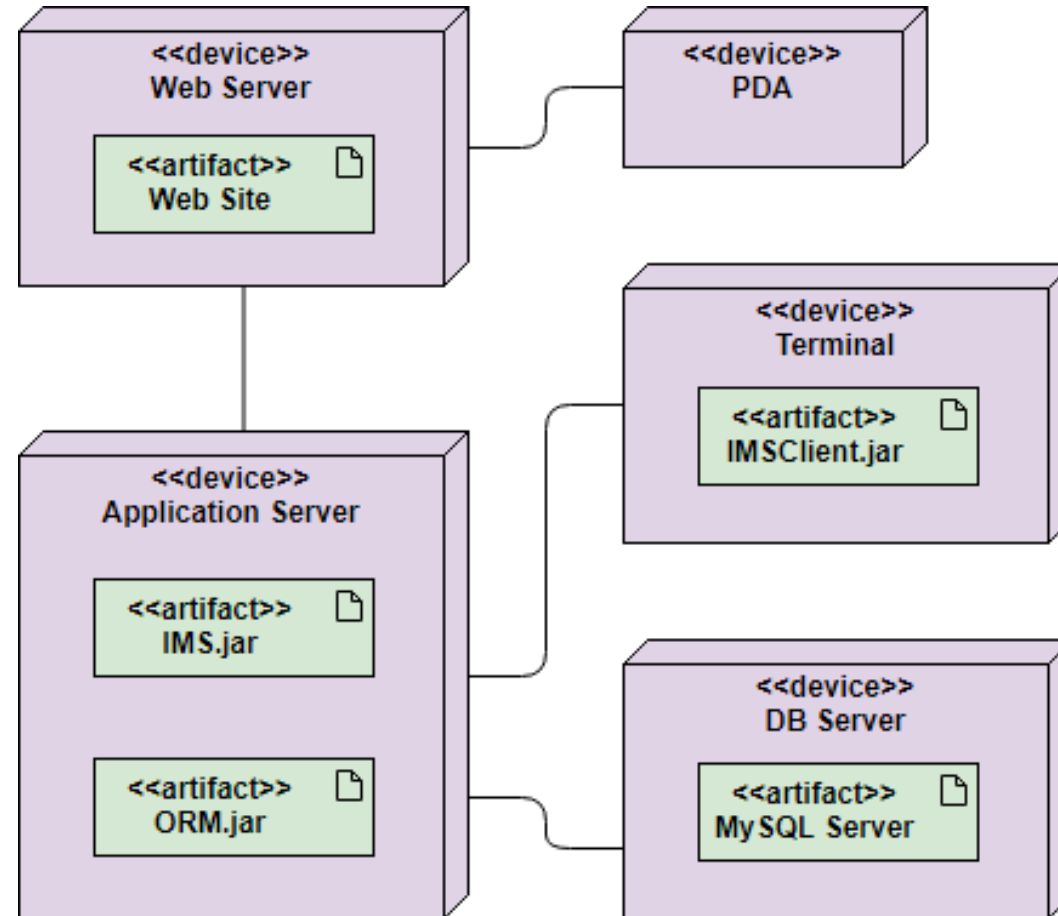


DEPLOYMENT DIAGRAMS

- Deployment diagrams contain artifact
- An artifact
 - Is the specification of a physical piece of information
 - Ex: source files, binary executable files, table in a database system,.....
 - An artifact defined by the user represents a concrete element in the physical world

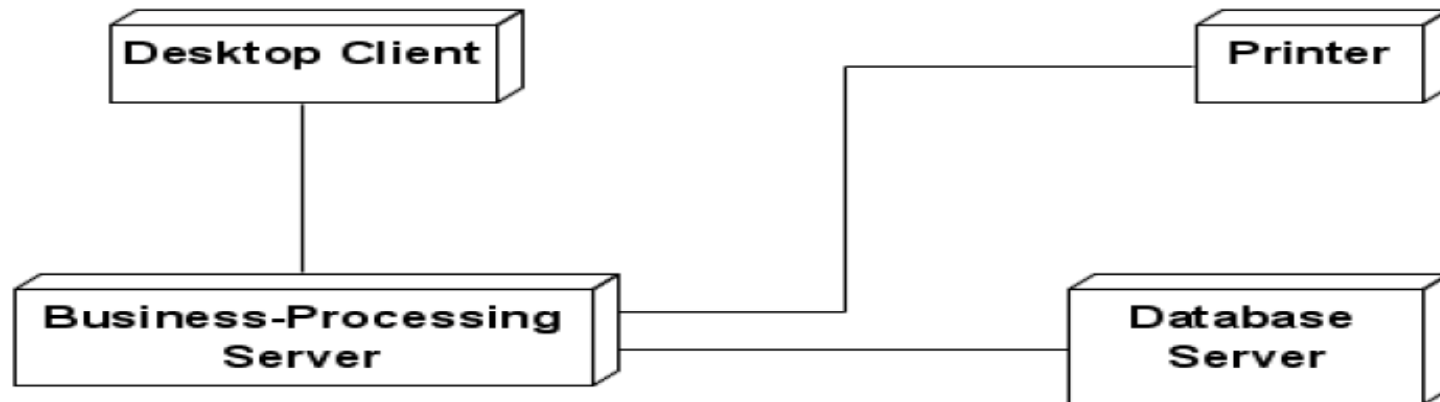


DEPLOYMENT DIAGRAMS



Communication Association

- A communication associations between nodes indicates a communication path between the nodes that allows components on the nodes to communicate with one another.



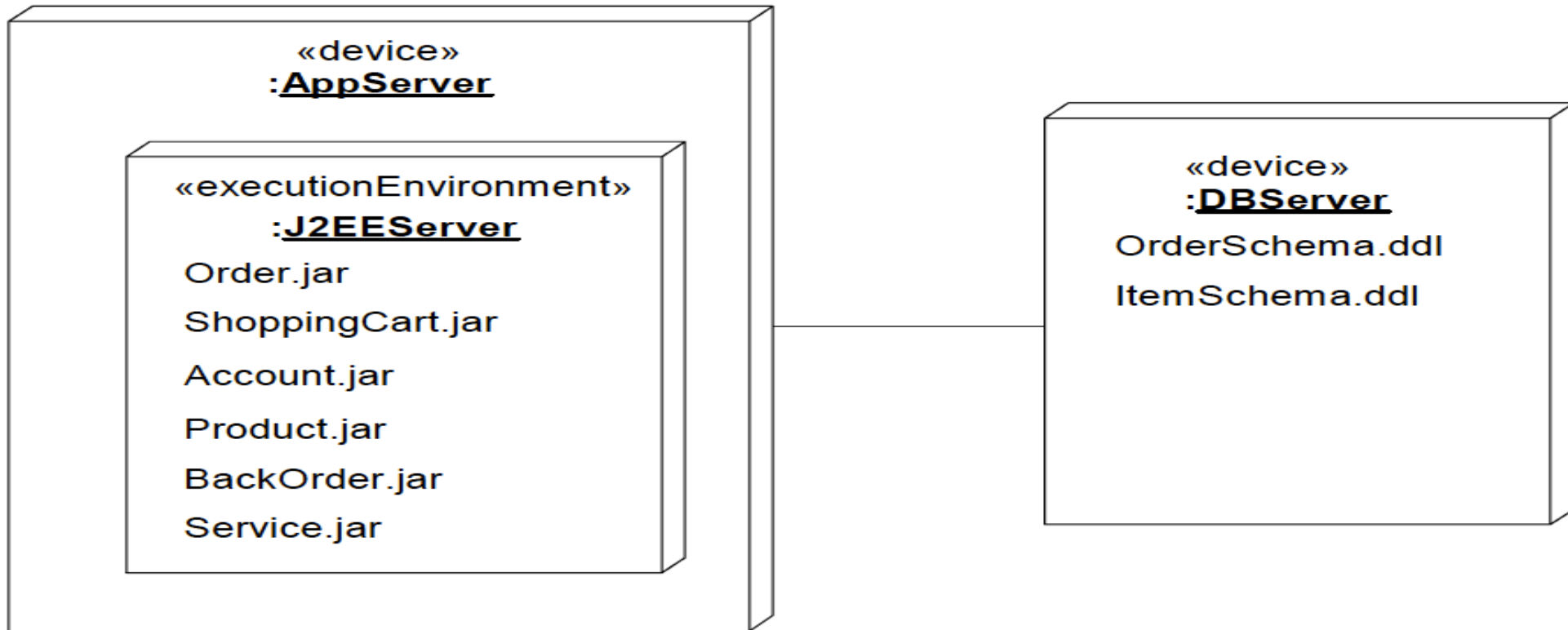
UML 2 Deployment Diagrams - nodes

- Two node types:
- **Device**
 - a physical computational resource with processing capability (memory and services to execute software) upon which artifacts may be deployed for execution. For example, typical computers or mobile phones.
- **Execution environment**
 - a node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts.
 - software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

UML 2 Deployment Diagrams - nodes

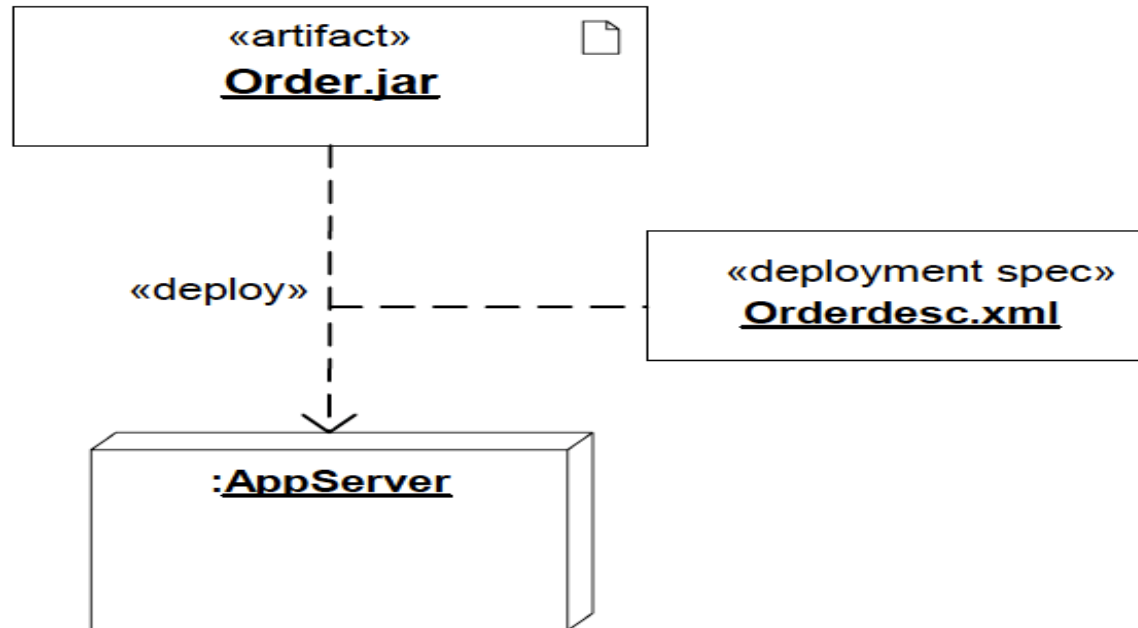
- The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes.
- Nodes may have sub-nodes, which appear as nested boxes.
- A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

UML 2 Deployment Diagrams - nodes



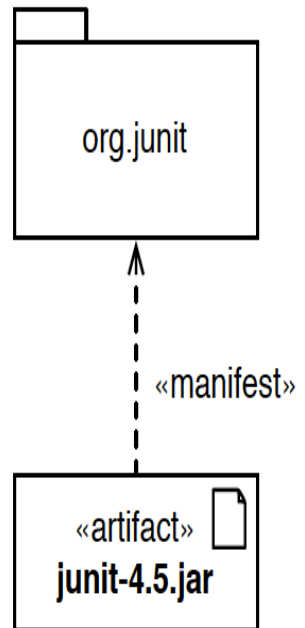
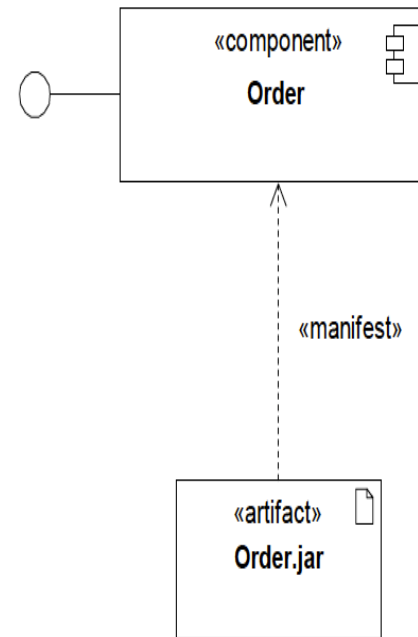
Deployment Specification

- A deployment diagram can have a deployment specification



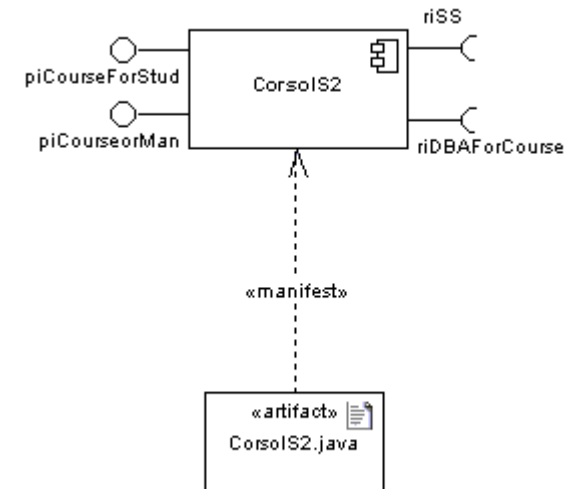
Manifestation

- Artifacts can now manifest any packageable element, not just components
- Manifestation (the concrete physical rendering of one or more model elements by an artifact) is shown by a dependency with keyword «manifest»

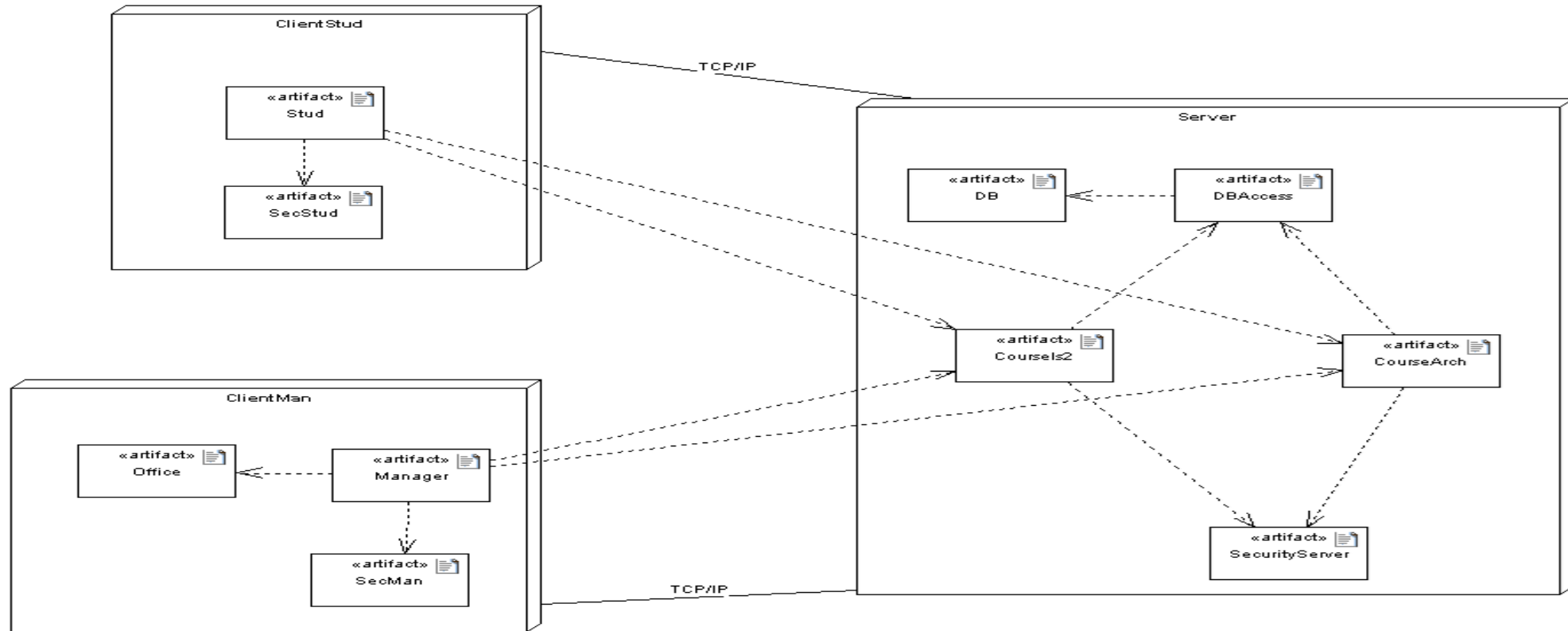


Manifestation

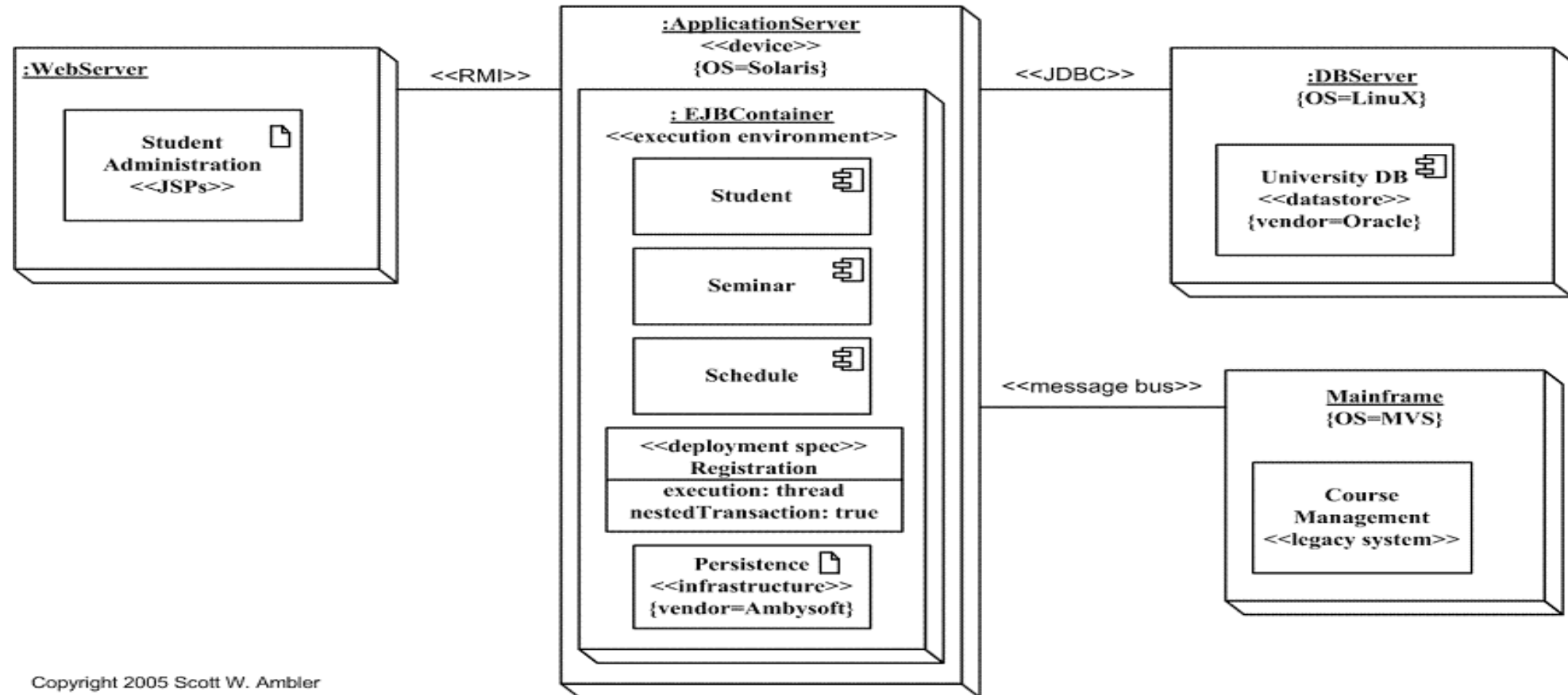
- An artifact manifest one or more model elements
 - A <<manifestation>> is the concrete physical of one or more model elements by an artifact
 - This model element is often a component
-
- A manifestation is notated as a dashed line with an open arrow-head labeled with the keyword <<manifest>>



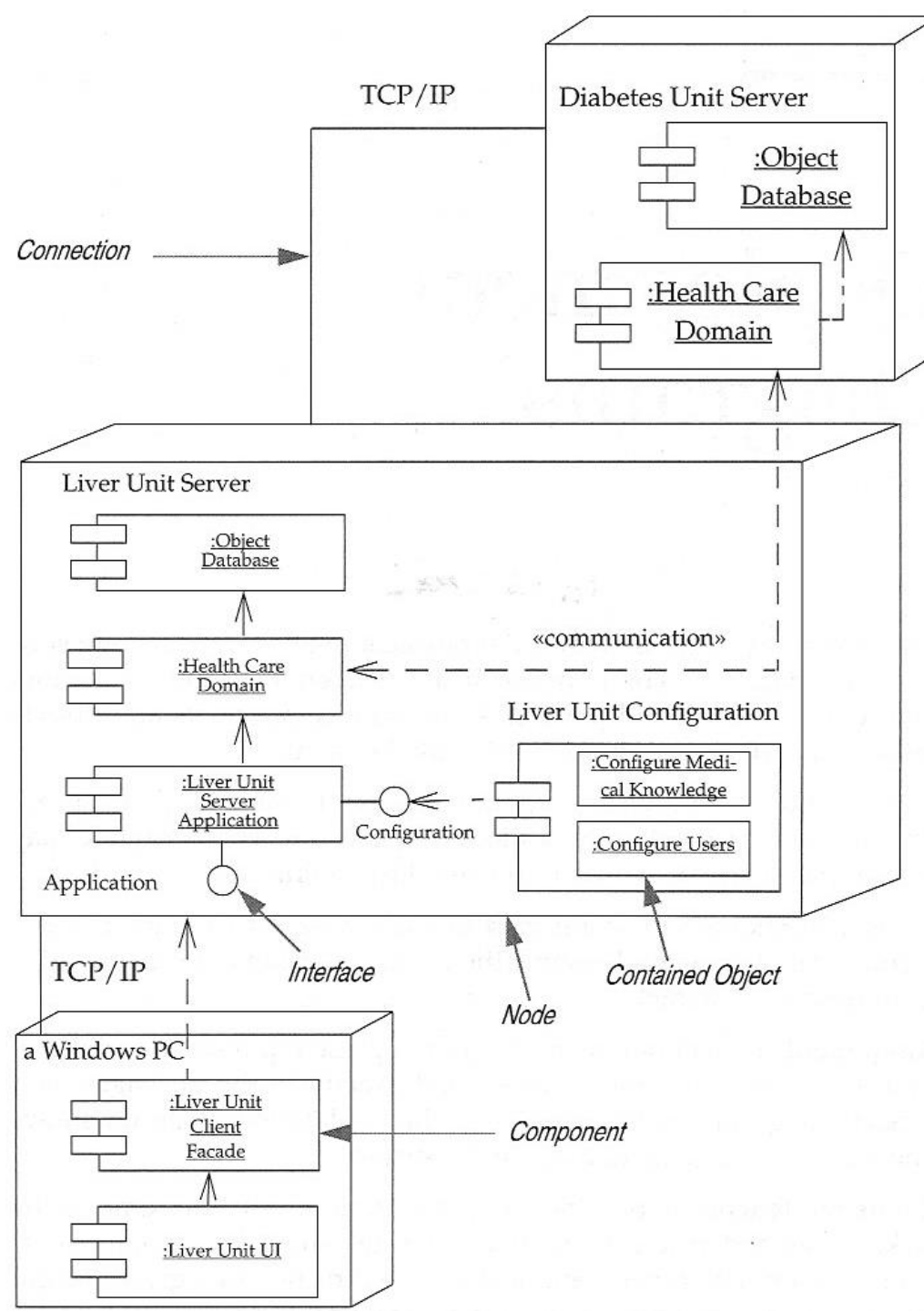
DEPLOYMENT DIAGRAMS



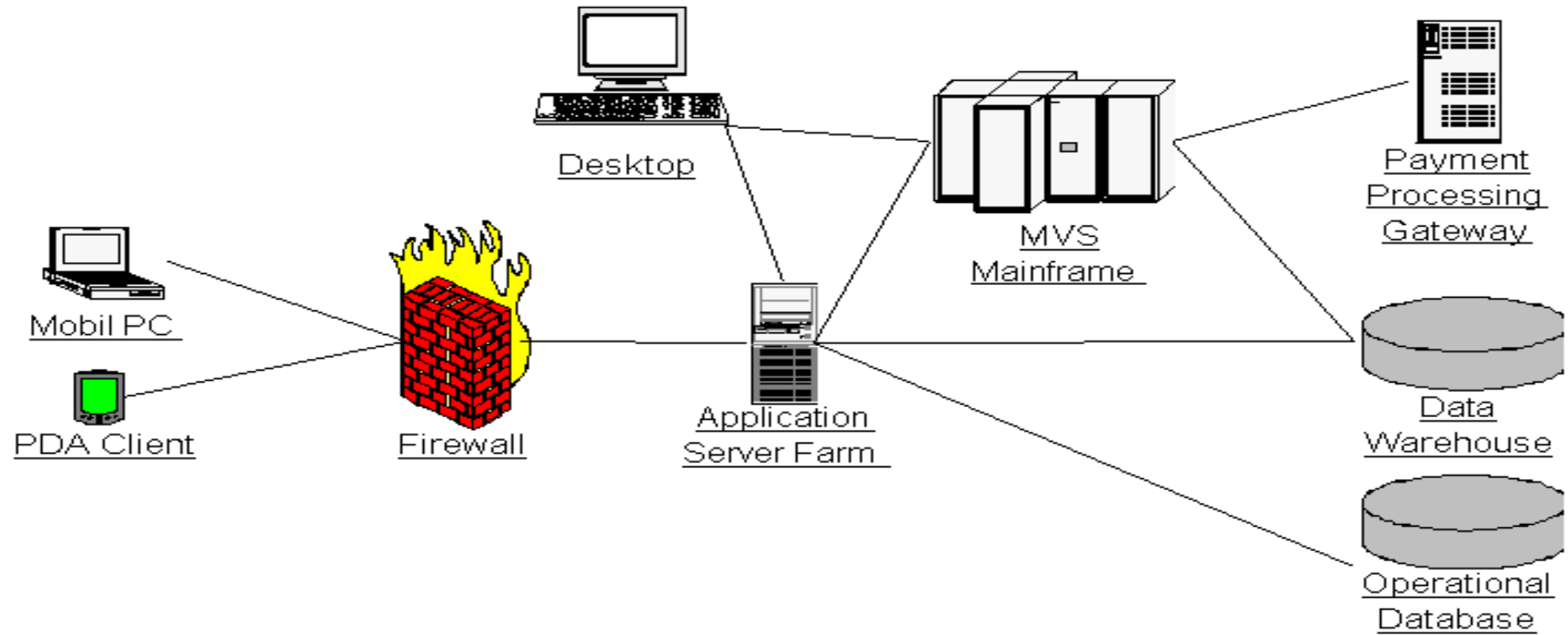
Deployment Diagrams



Deployment Diagrams Example



System Architecture



Deployment Planning

- How will your system be installed?
- If different versions of the system will be in production at the same time, how will you resolve differences?
- What physical sites do you need to deploy to and in what order?
- How will you train your users?

Deployment Planning

- How will your system be installed?
 - Who will install it? How long should it take to install?
 - Where the installation possibly fail? How do you back out if the installation fails? How long does it take to back out?
 - What is your installation window (during what time period can you install your system)?
 - What backups do you need before installation? Do you need to do a data conversion?
 - How do you know that the installation was successful?

Deployment Planning

- What physical sites do you need to deploy to and in what order?
 - What physical sites do you need to deploy to and in what order?
 - How will you train your support and operations staff?
 - Do you need to deploy a production support system so that the support staff uses their own environment to simulate problems?
- How will you train your users?
 - What documentation, and in what formats and languages, do your users, and support and operation staff need?
 - How will updates to documentation be deployed?

How to Produce Deployment Diagrams

- Decide on the purpose of the diagram
- Add nodes to the diagram
- Add communication associations to the diagram
- Add other elements to the diagram, such as components or active objects, if required
- Add dependencies between components and objects, if required

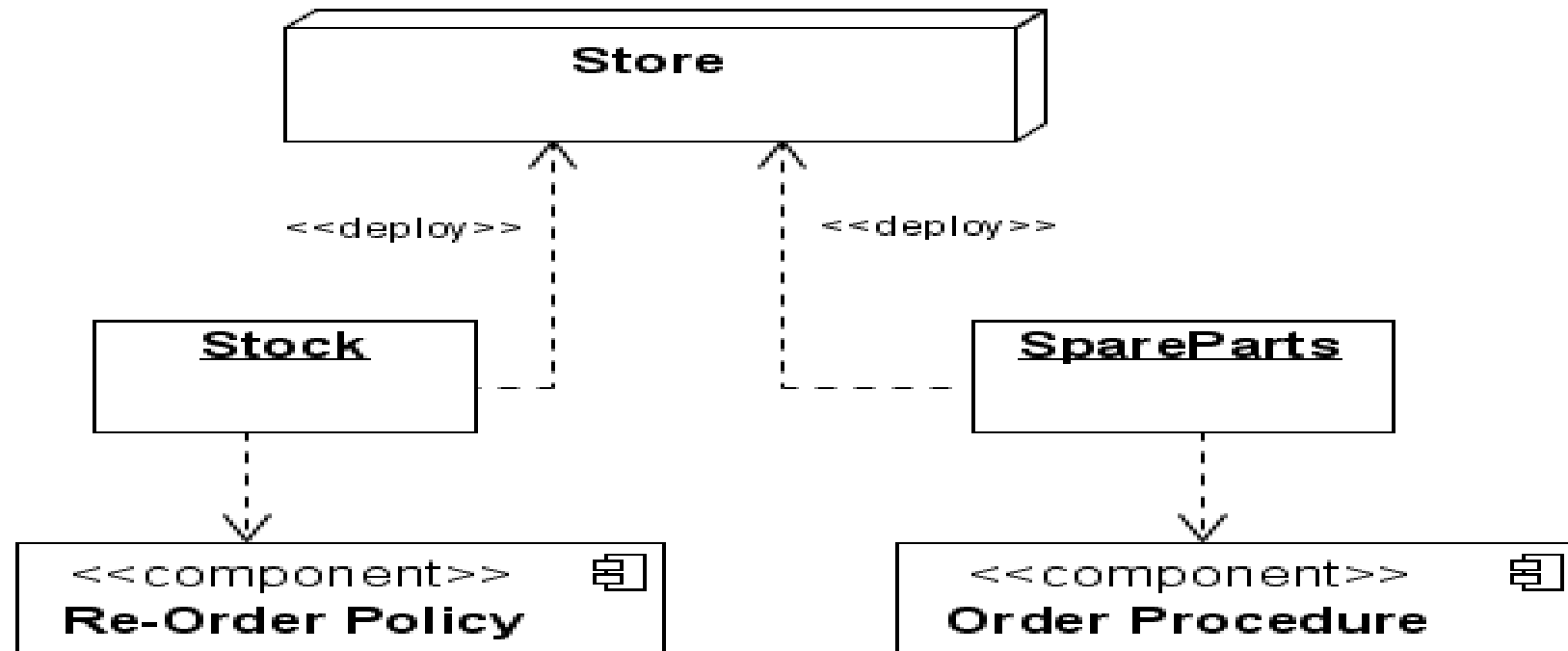
Allocating Artifacts to Nodes

- Consider a number of aspects when allocating components to nodes, e.g.:
 - Resource usage
 - Geographical location
 - Access to devices
 - Security
 - Performance
 - Extensibility and portability

Modelling Business Process

- Business modelling using nodes and components is an effective means of capturing non-computer based processes and entities
- This can be done very early in development, to complement the use case model and other business modelling
- Components are the business procedures and documents; the nodes ("run-time structure") are the organisation units and resources (human and other) of the business

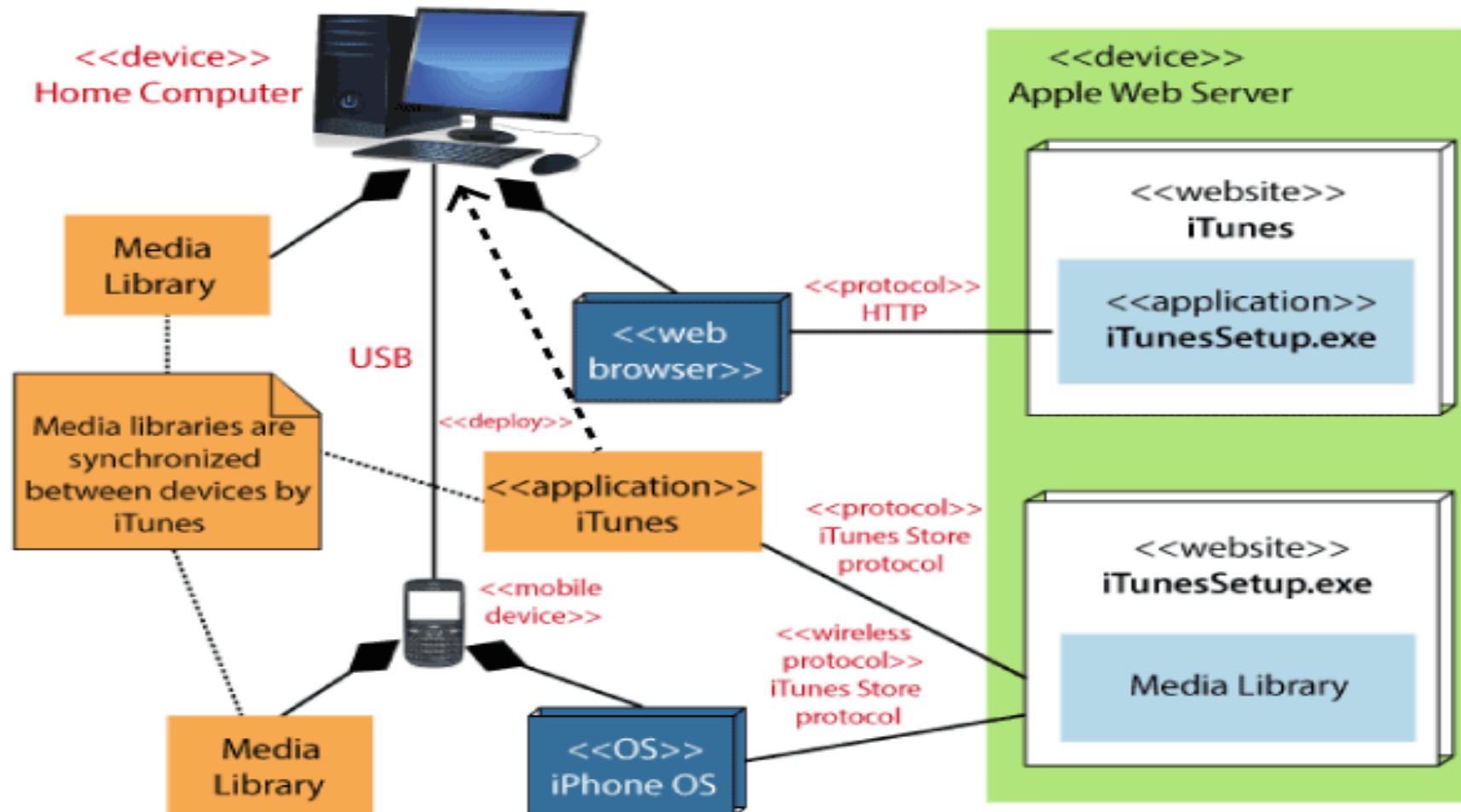
Modelling Business Process



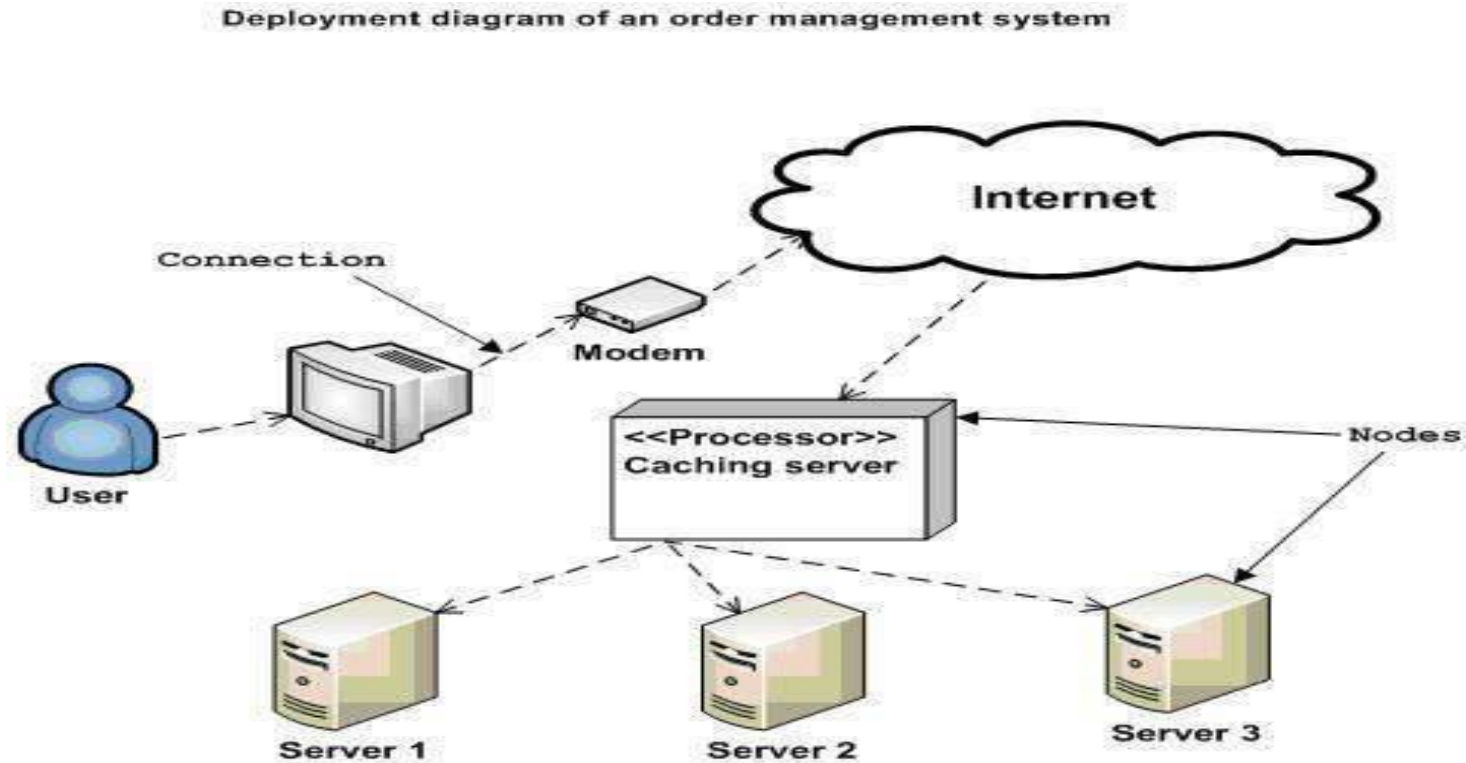
Example of a Deployment diagram

- A deployment diagram for the Apple iTunes application:
- The iTunes setup can be downloaded from the iTunes website, and also it can be installed on the home computer. Once the installation and the registration are done, iTunes application can easily interconnect with the Apple iTunes store. Users can purchase and download music, video, TV serials, etc. and cache it in the media library.
- Devices like Apple iPod Touch and Apple iPhone can update its own media library from the computer with iTunes with the help of USB or simply by downloading media directly from the Apple iTunes store using wireless protocols, for example; Wi-Fi, 3G, or EDGE.

A deployment diagram: Apple iTunes Application



Deployment diagram: Order Management System



Management Information System for a SME

