Object Oriented Analysis and Design with Java

UE20CS352

Prof. Nivedita Kasturi
Department of Computer Science and Engineering

UE20CS352: Object Oriented Analysis and Design with Java

Deployment Model

Prof. Nivedita Kasturi
Department of Computer Science and Engineering

Deployment Diagram

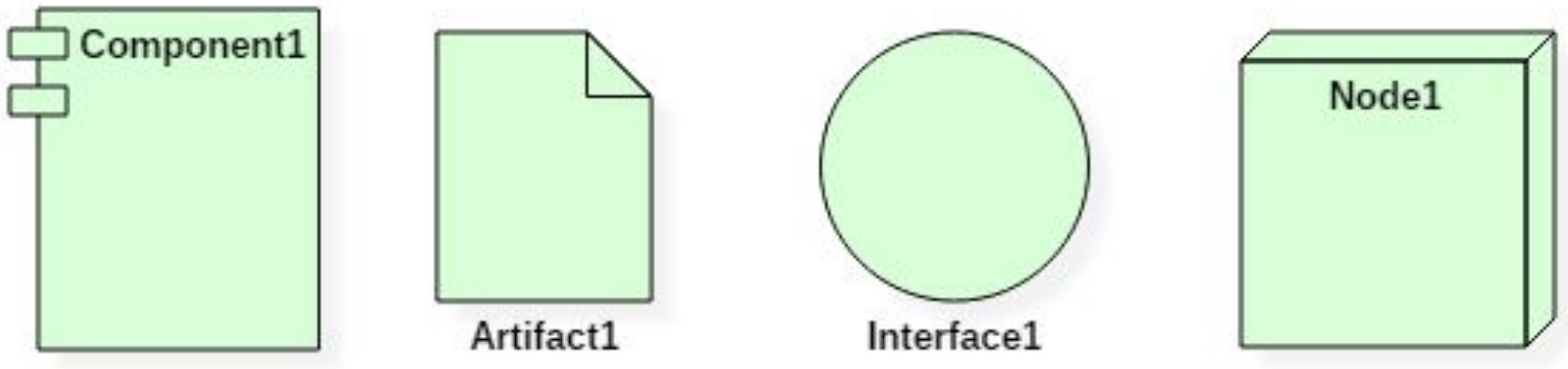
Introduction

- Software components developed as part of the development process would need to be deployed on some set of hardware or software to be executed.
- Deployment diagram maps the software architecture created in design to the physical system architecture that executes it. In distributed systems, it models the distribution of the software across the physical nodes.
- Deployment diagrams shows how the components described in component diagrams are deployed in hardware.
- Deployment diagrams are used to visualize
  - The topology of the physical components of a system, where the software components are deployed.
  - Describe the hardware components used to deploy software components.
  - Describe the runtime processing nodes (physical hardware) .

Deployment Diagram – Notations

- Deployment diagram, models the run-time architecture of a system.
- It shows the configuration of the hardware elements (nodes) and shows how software elements and artifacts are mapped onto those nodes.

Some of the deployment diagram symbols and notations



Deployment Diagram

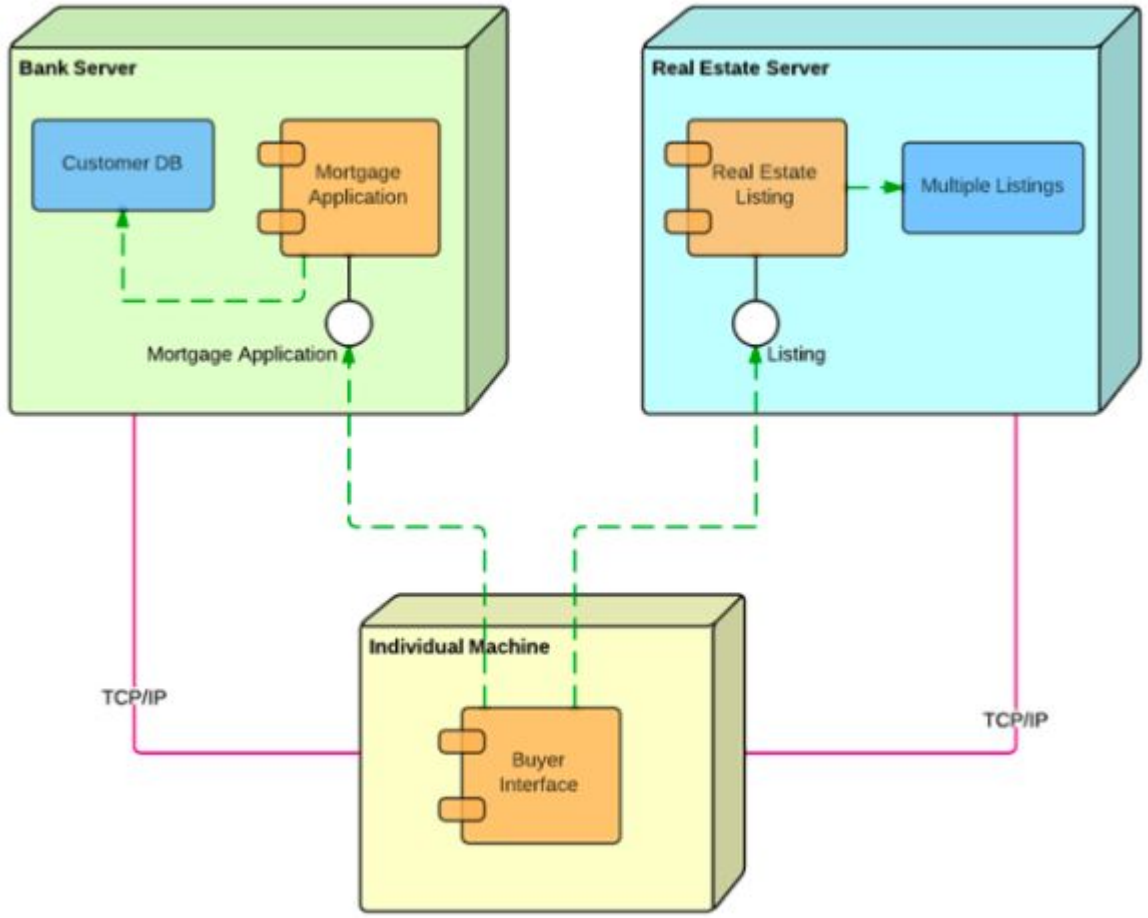
Uses of a Deployment Diagram

- Deployment diagrams are useful for system engineers.
- An efficient deployment diagram is very important as it controls the following parameters –
  - Performance
  - Scalability
  - Maintainability
  - Portability
  - Understandability
- Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering.

Deployment Diagram

Elements in a Deployment Diagram

- **Artifact:** A product developed by the software, symbolized by a rectangle with the name and the word "artifact" enclosed by double arrows.
- **Association**: A line that indicates a message or other type of communication between nodes.
- **Component:** A rectangle with two tabs that indicates a software element.
- **Dependency:** A dashed line that ends in an arrow, which indicates that one node or component is dependent on another.
- **Interface:** A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
- **Node:** A hardware or software object, shown by a three-dimensional box.

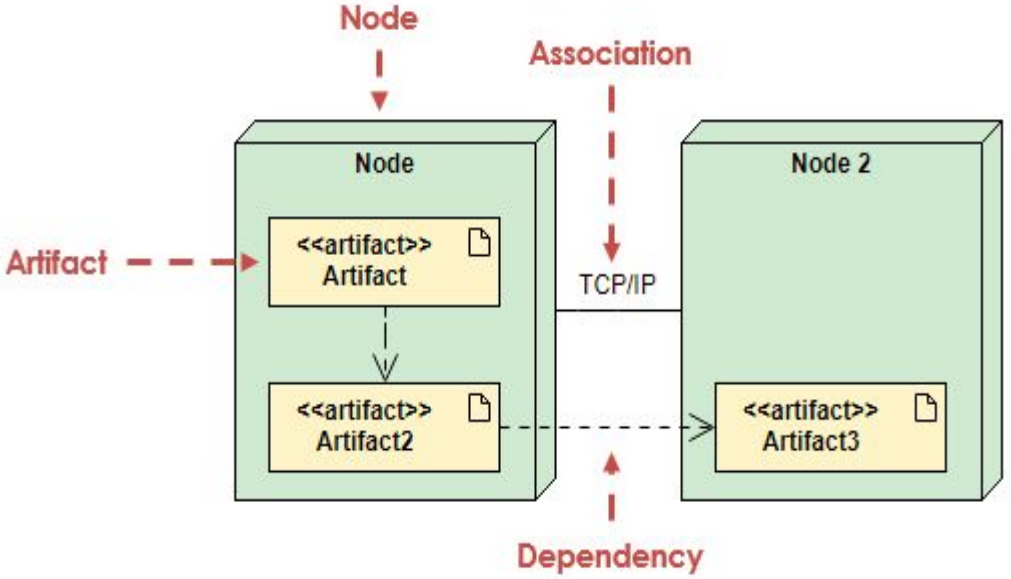


Deployment Diagram

Artifact

- Artifacts represent concrete elements in the physical world that are the result of a development process. Examples of artifacts are executable files, libraries, archives, database schemas, configuration files, etc.

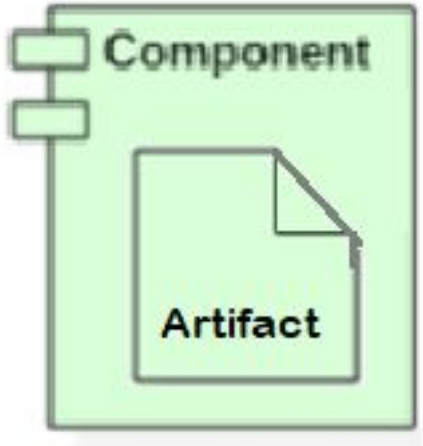
- Artifacts are concrete elements that are caused by a development process. Examples of artifacts are libraries, archives, configuration files, executable files etc.



Deployment Diagram

Component with Artifacts

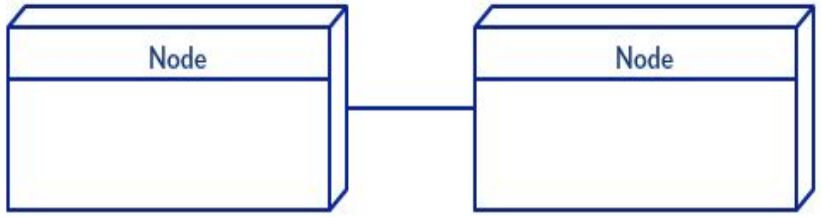
- Component with Artifacts represents the concrete real-world entity related to software development.
- Things that participate in the execution of a system or executable entities that reside in nodes.
- Represent the physical packaging of the logical elements.
- Artifacts are deployed on the nodes.
- Each artifact has a filename in its specification that indicates the physical location of the artifact.

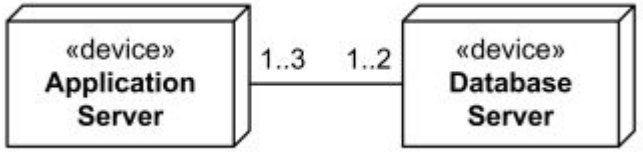


Deployment Diagram

Associations

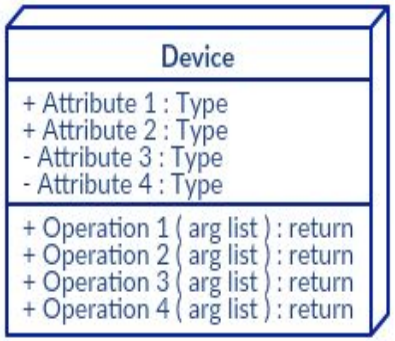
Communication Association

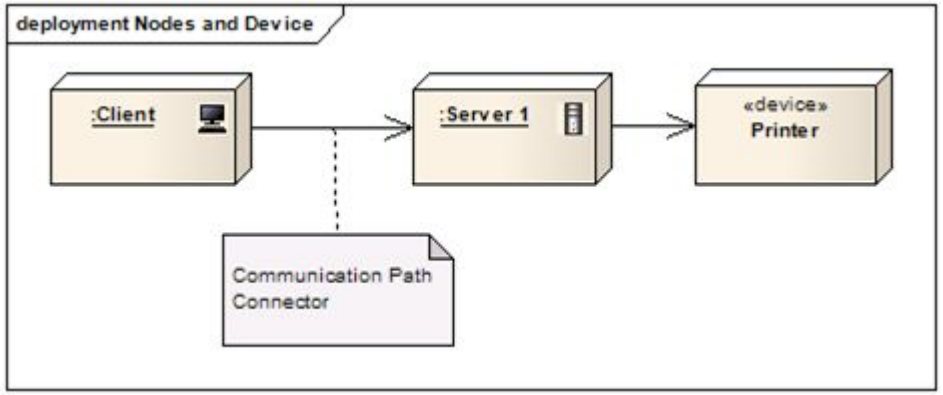




This is represented by a solid line between two nodes. It shows the path of communication between nodes.

Devices



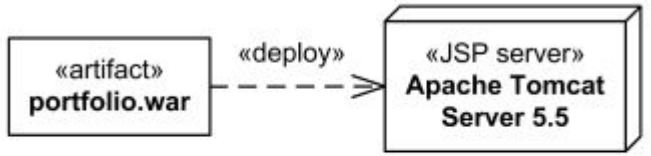


A device is a node that is used to represent a physical computational resource in a system. An example of a device is an application server.

Deployment Diagram

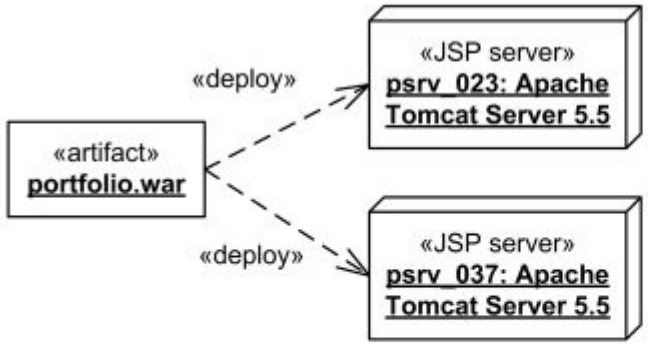
Associations…(cntd.)

- Deployment could be shown as a dependency that is drawn from the artifact (supplier) to the deployment target (client) and is labeled with «deploy».



J2EE web application archive portfolio.war deployed on Apache Tomcat JSP server.

- At the "instance level" instances of artifacts could be deployed to specific instances of the deployment target.

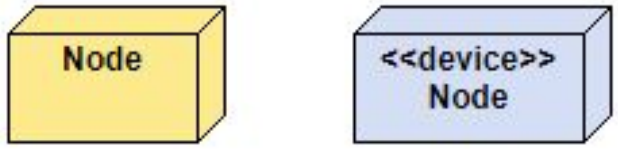


J2EE web application archive portfolio.war deployed on two instances of Apache Tomcat JSP server - psrv_023 and psrv_037.

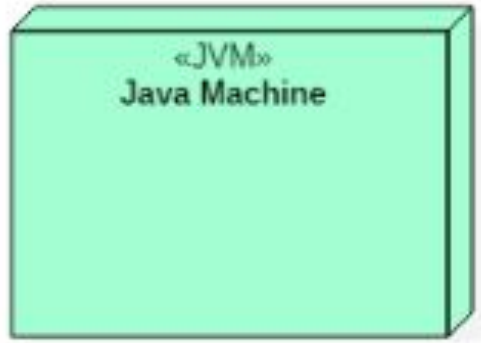
Deployment Diagram

Nodes

- Node : Node is a computational resource upon which component artifacts are deployed for execution.



device node



execution environment node

- There are two types of Nodes:
  - **Device Node**
  - **Execution Environment Node**

Device nodes are physical computing resources with processing memory and services to execute software, such as typical computers or mobile phones.

An execution environment node (EEN) is a software computing resource that runs within an outer node and which itself provides a service to host and execute other executable software elements.

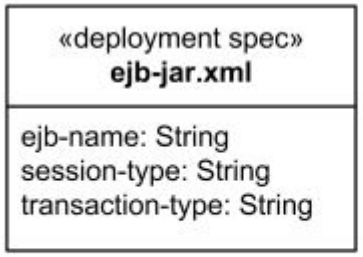
Deployment Diagram

Deployment Specifications

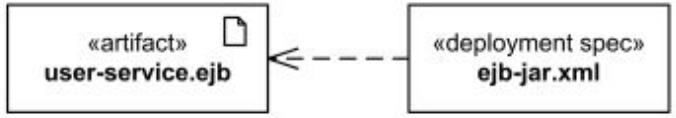
Deployment Specification

- Deployment specifications is a configuration file, such as a text file or an XML document. It describes how an artifact is deployed on a node.

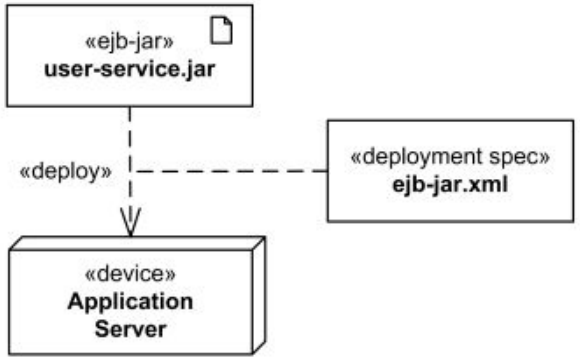
- A deployment specification at specification level is rendered as a classifier rectangle with optional deployment properties in a compartment.



The ejb-jar.xml deployment specification



The ejb-jar.xml deployment specification for user-service.ejb artifact.



The ejb-jar.xml deployment specification attached to deployment.

Deployment Diagram

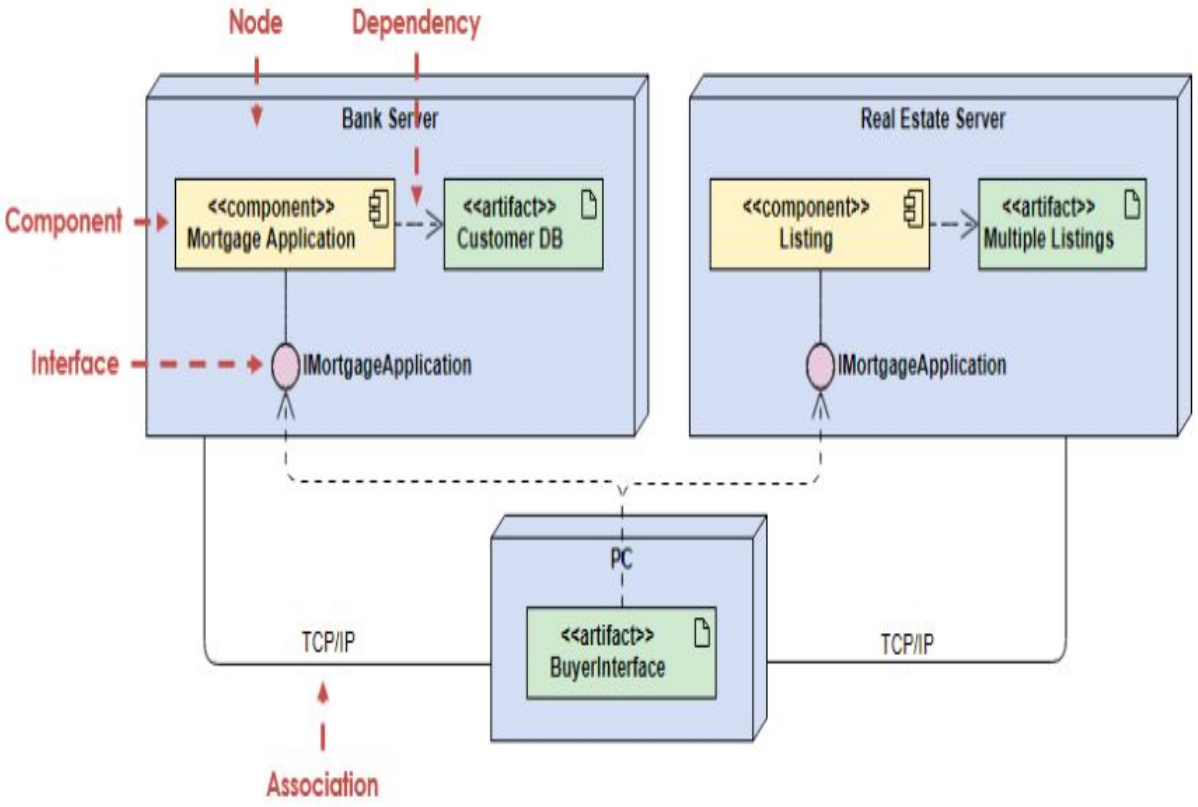
How to Draw deployment Diagram

Step 1: Identify the purpose of your deployment diagram. And to do so, you need to identify the nodes and devices within the system you'll be visualizing with the diagram.

Step 2: Figure out the relationships between the nodes and devices. Once you know how they are connected, proceed to add the communication associations to the diagram.

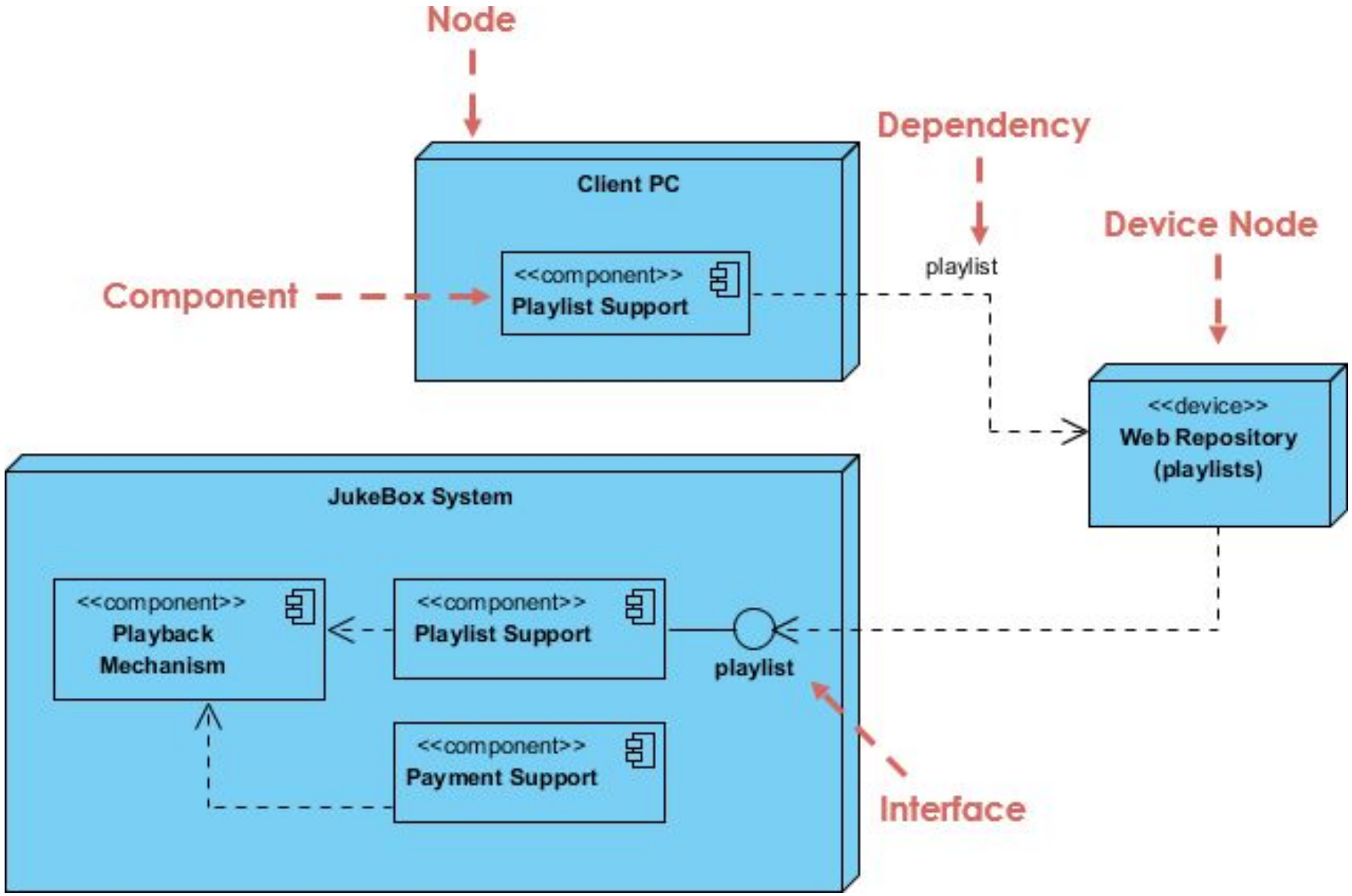
Step 3: Identify what other elements like components, active objects you need to add to complete the diagram.

Step 4: Add dependencies between components and objects as required.



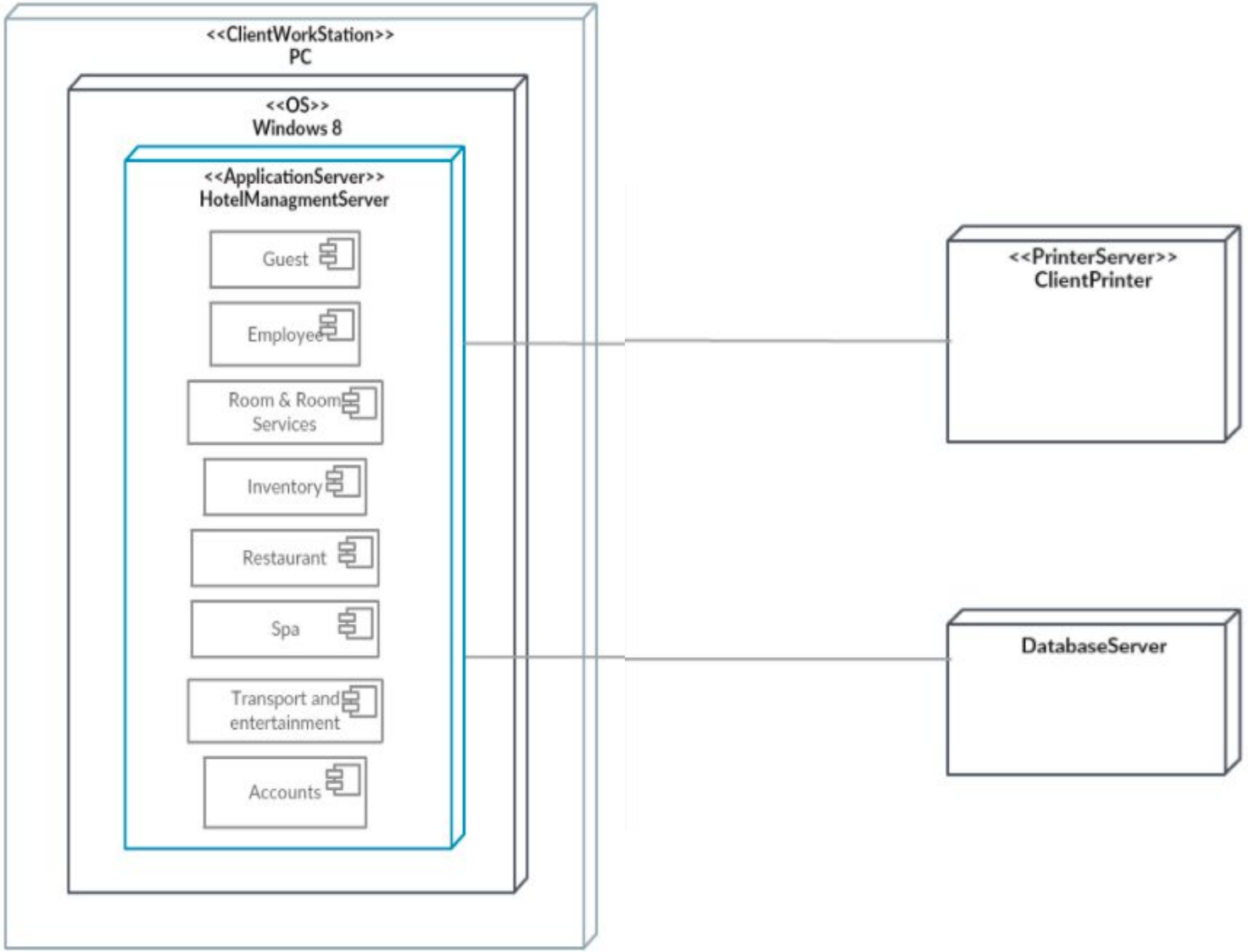
Deployment Diagram

Example – Jukebox



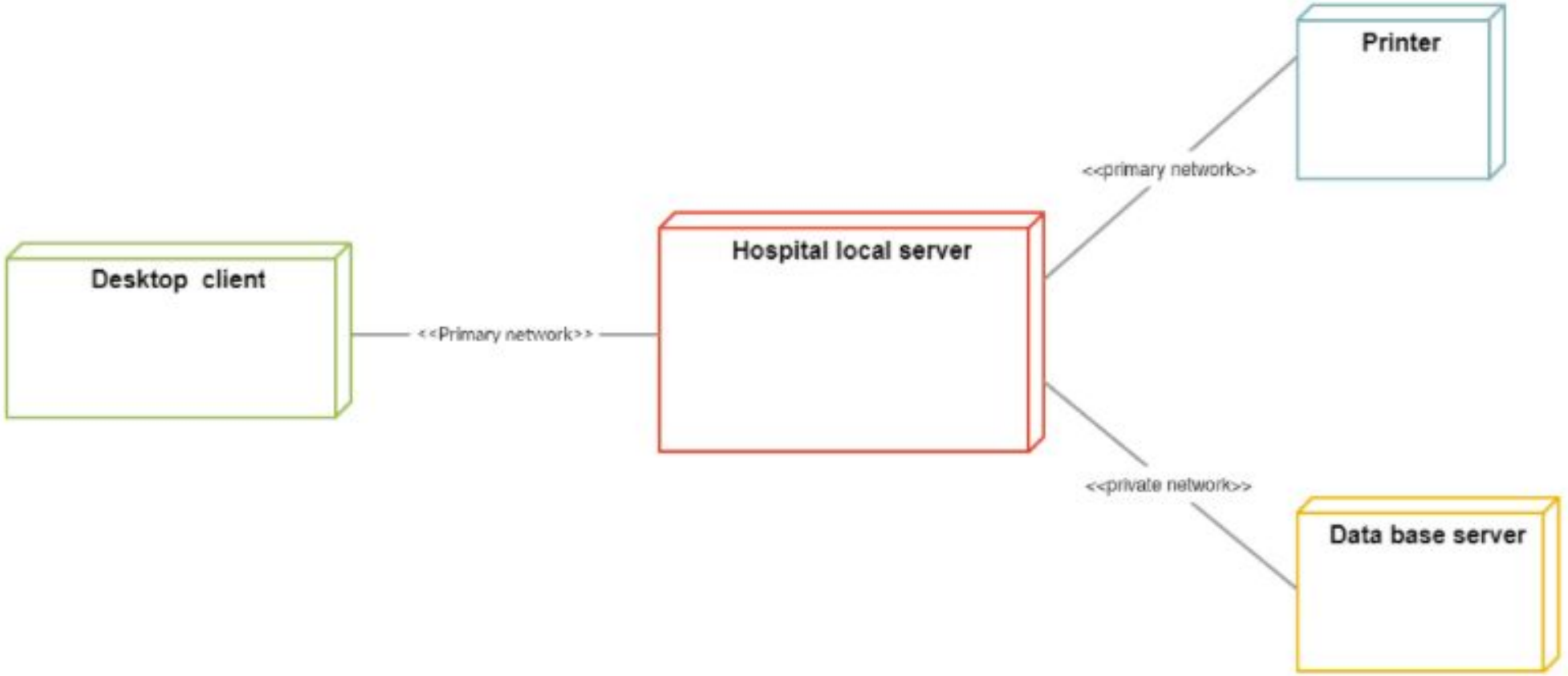
Deployment Diagram

Example – Hotel Management System



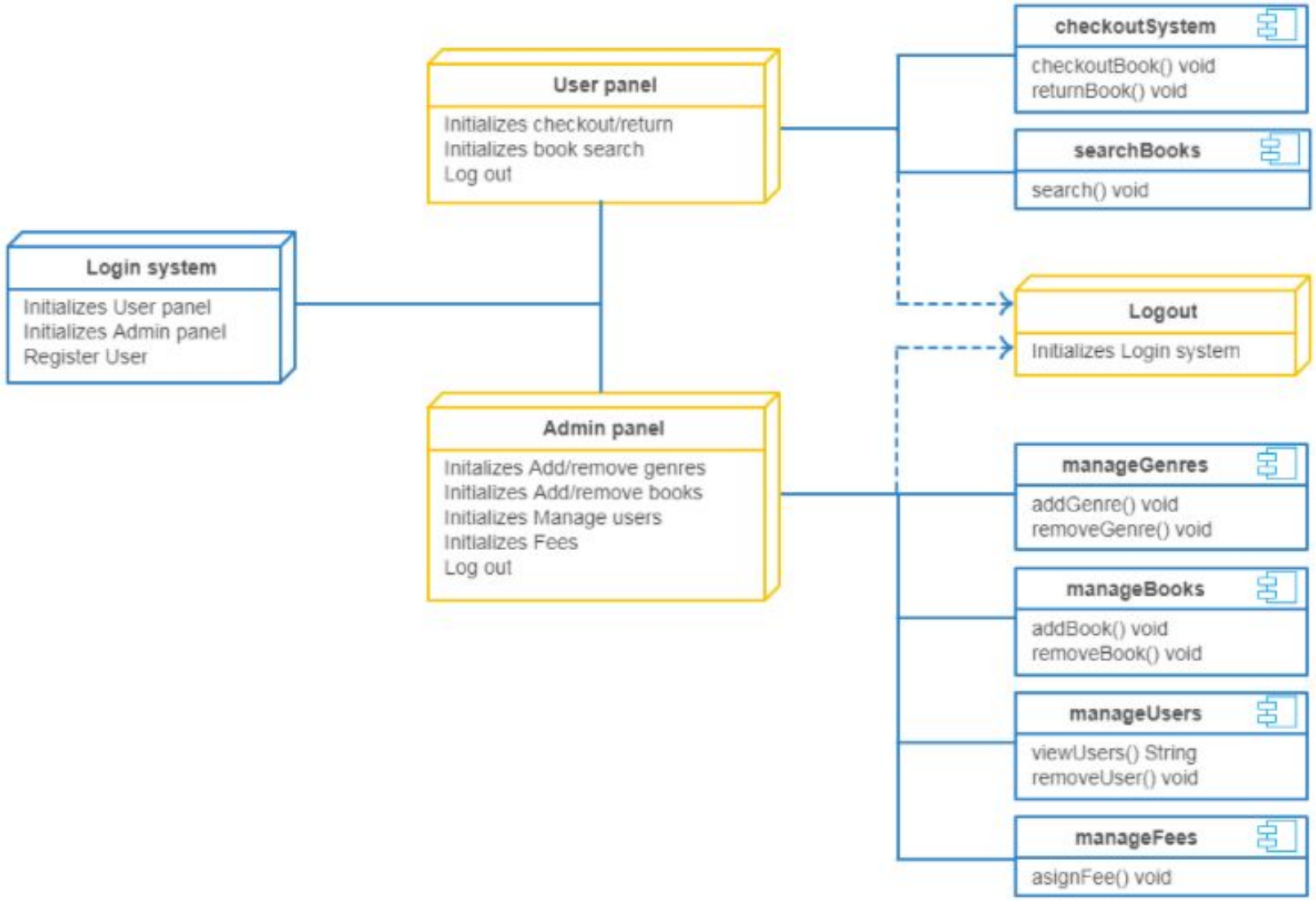
Deployment Diagram

Example – Hospital Management System



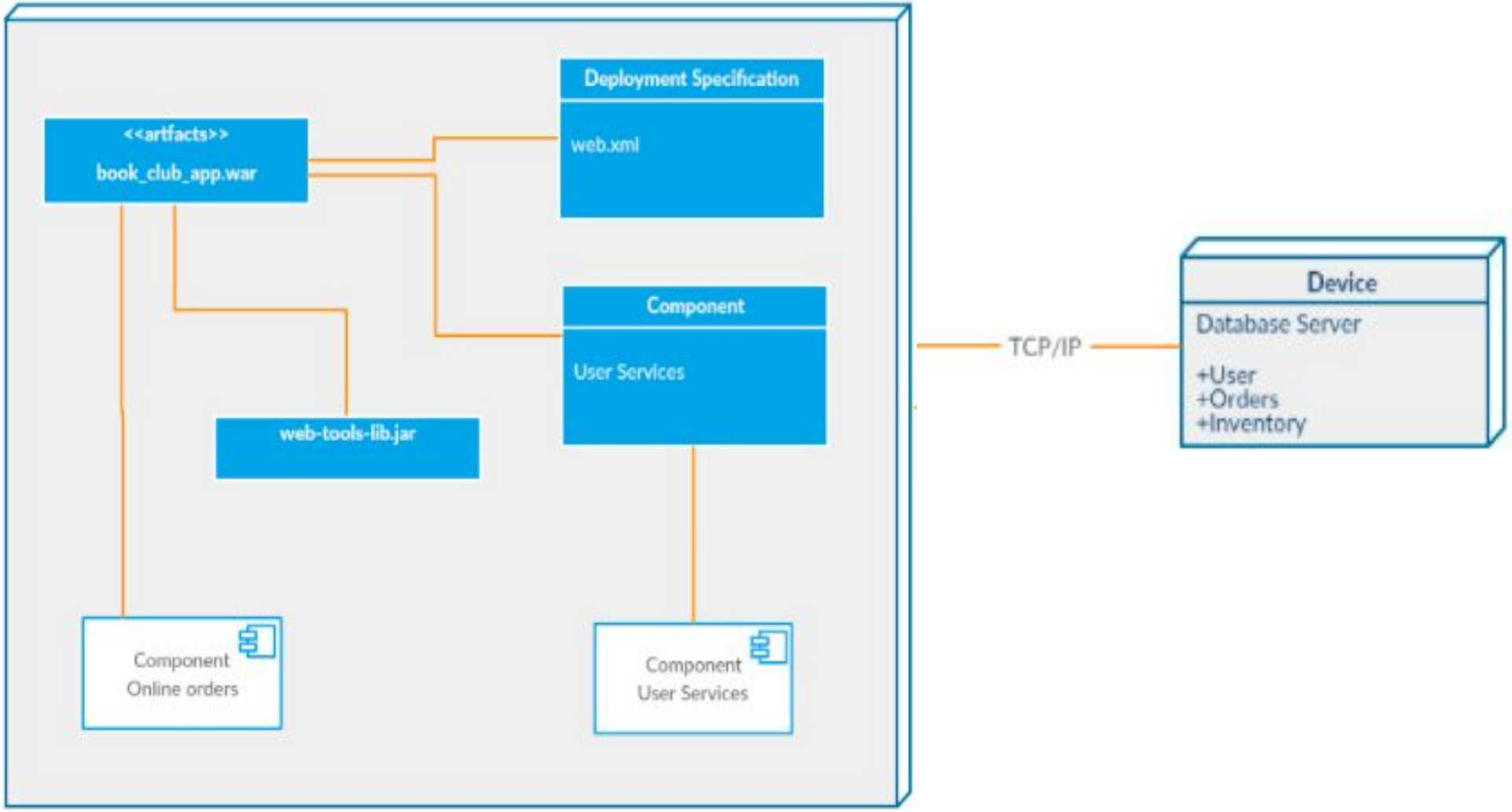
Deployment Diagram

Example – Library Management System



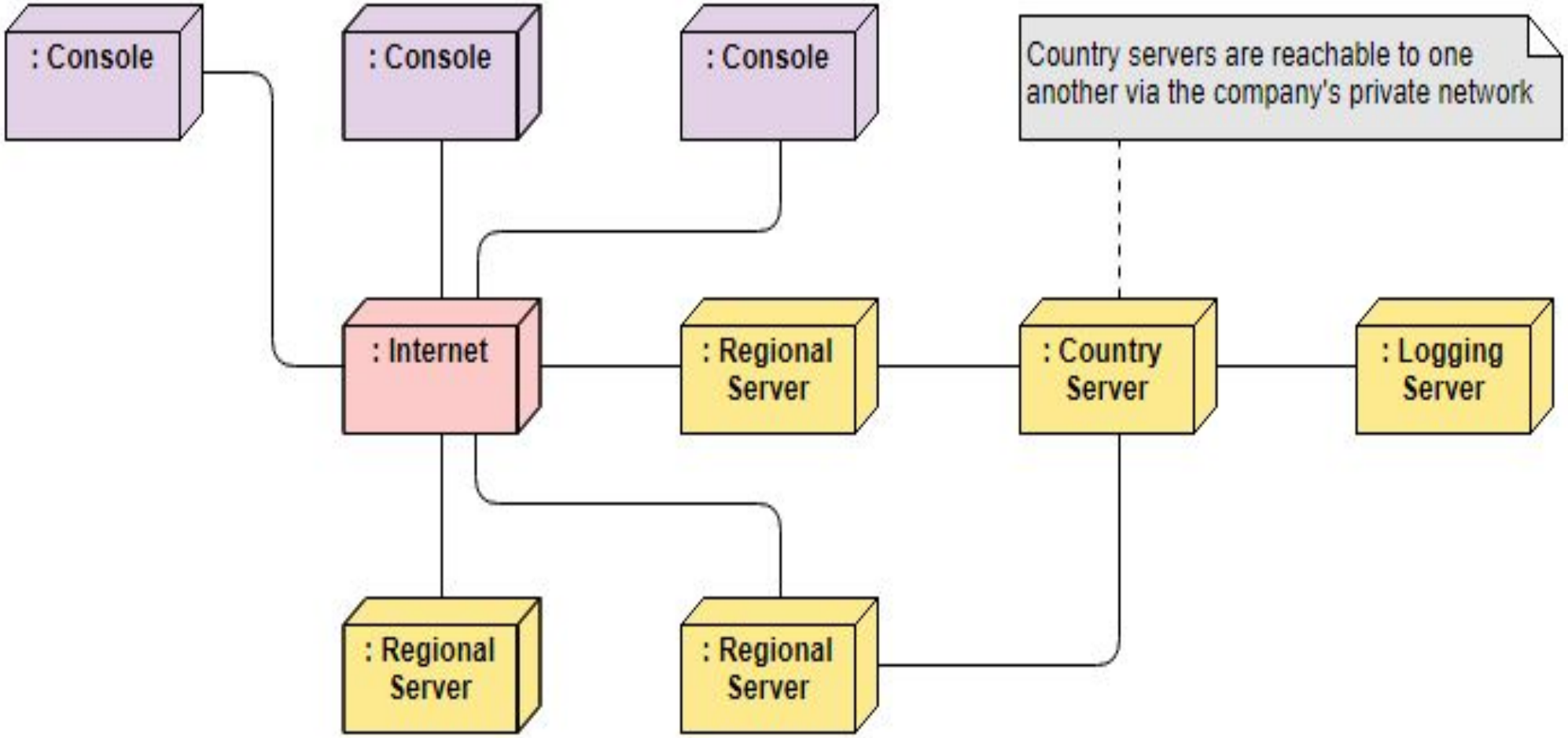
Deployment Diagram

Example – Online Shopping System



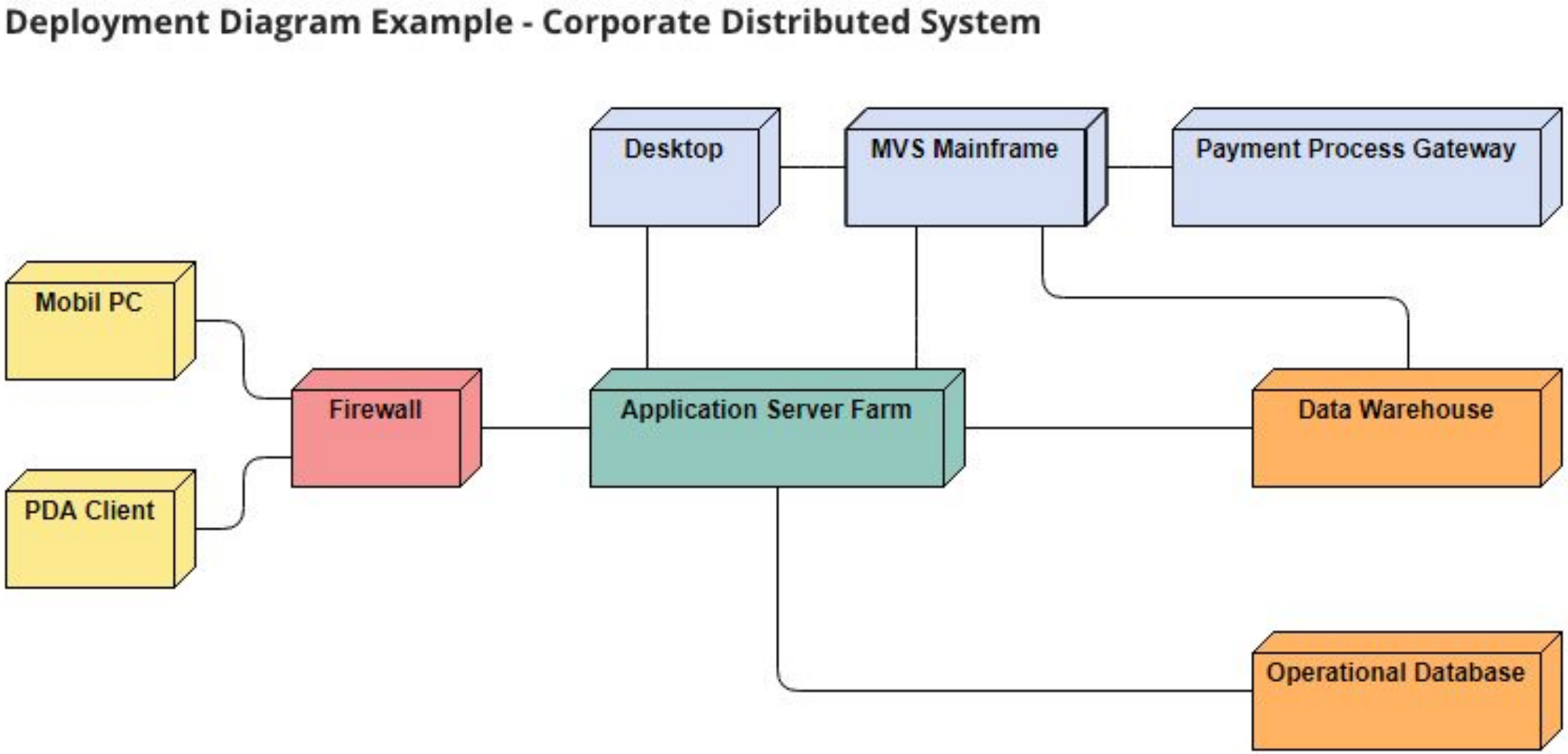
Deployment Diagram

Example – Modeling a Distributed System



Deployment Diagram

Illustration of a Deployment Diagram



THANK YOU

Prof. Nivedita Kasturi

Department of Computer Science and Engineering

Niveditak@pes.edu