Deployment Diagrams

Implementation Diagram

- show aspects of physical implementation:
- □ Structure of components.
- □ Run-time deployment systems.
- □ Two diagram types:
- □ **Component diagram** – show the structure of
- components, including the classifiers that specify
- them and the artifacts that implement them.
- □ **Deployment diagram** - show the structure of the
- nodes on which the components are deployed.
- □ These two diagrams are usually drawn together.

Need for Deployment Diagram

- **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

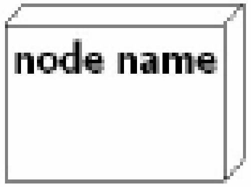
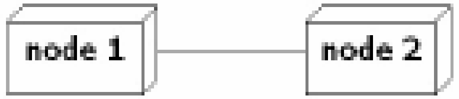
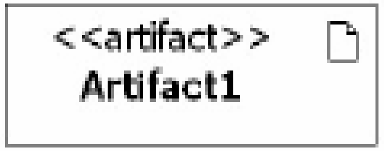
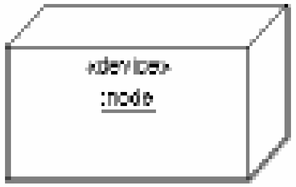
 Plan the **architecture** of a system

 Document the deployment of software components or nodes

What is a Deployment Diagram

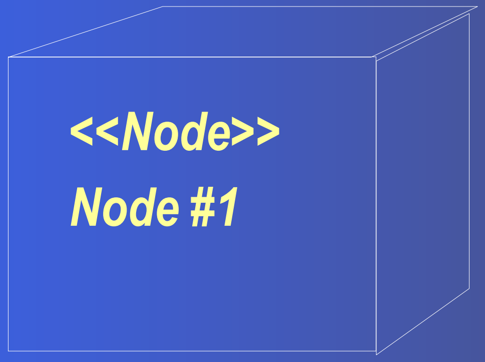
- The deployment diagram shows how a system will be physically deployed in the hardware environment. Its purpose is to show where the different components of the system will physically run and how they will communicate with each other. Since the diagram models the physical runtime, a system's production staff will make considerable use of this diagram.
- The notation in a deployment diagram includes the notation elements used in a component diagram, with a couple of additions, including the concept of a node. A node represents either a physical machine or a virtual machine node (e.g., a mainframe node). To model a node, simply draw a three-dimensional cube with the name of the node at the top of the cube. Use the naming convention used in sequence diagrams.

Deployment Diagrams - Notation

Node	Communication Path	Artefacts
		
<<device>> <<execution environment>>	Deployment Specifications	Deployment of Artefacts
	<<deployment spec>>	<<deploy>>

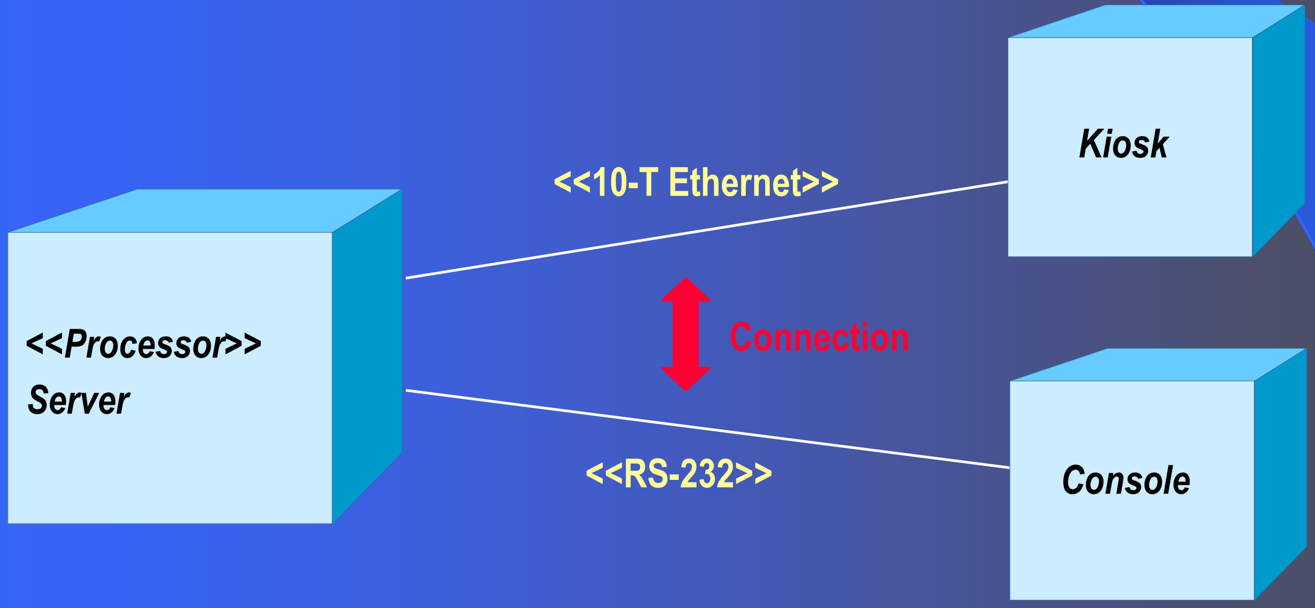
What is Node?

- a *node* is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability.
- A set of components may reside on a node and may also migrate from node to node.
- Graphically, a node is rendered as a cube, usually including only its name.



What Is a Connection?

- A connection represents a:
  - Communication mechanism
    - Physical medium
    - Software protocol



Deployment Diagram

- **A *deployment diagram* is a diagram that shows the configuration of run time processing nodes and the components that live on them.**
- **Captures the topology of a system's hardware**
- **Built as part of architectural specification**
- **Purpose**
  - Specify the distribution of components
  - Identify performance bottlenecks
- **Developed by architects, networking engineers, and system engineers**

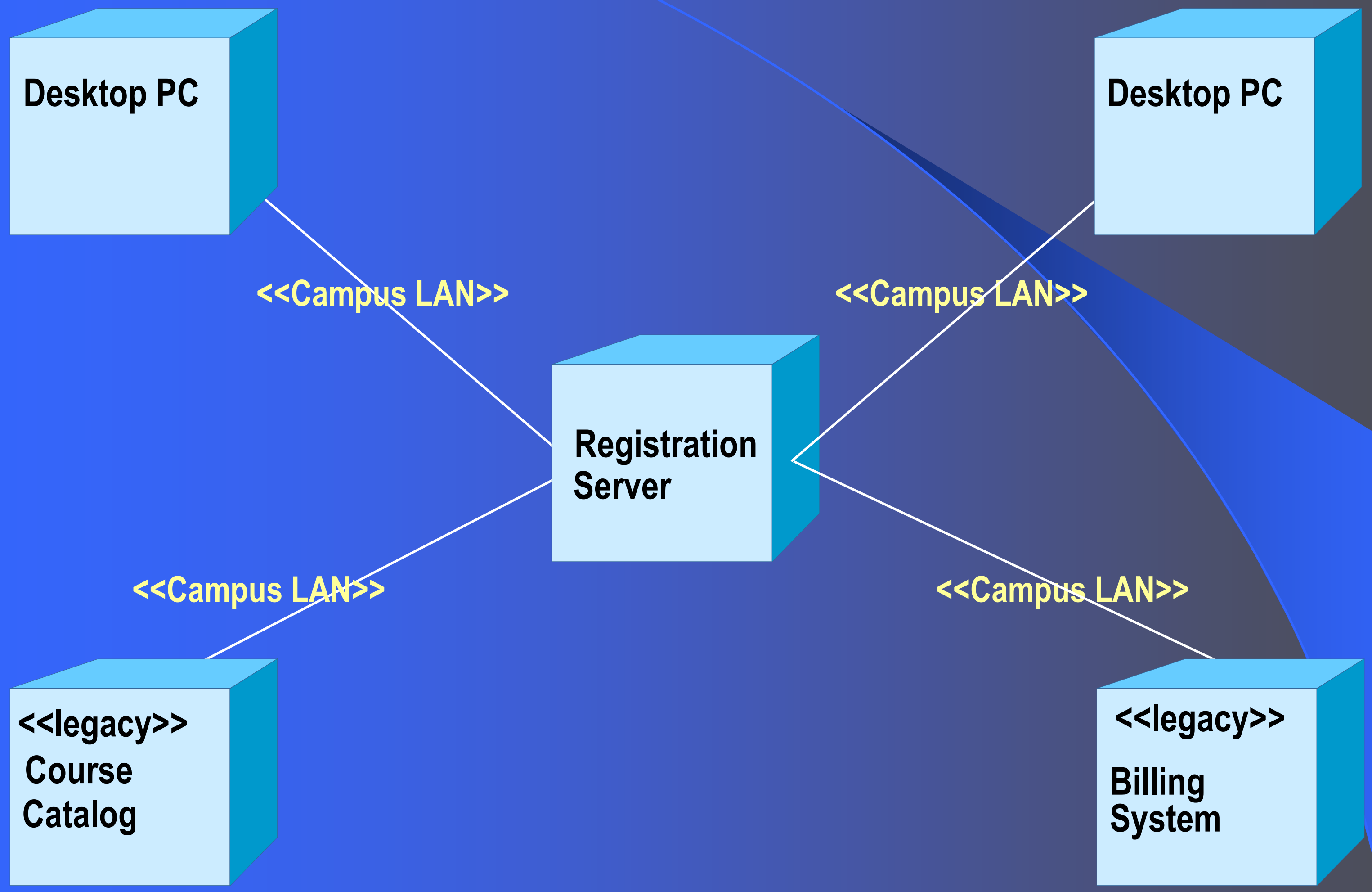
Purpose

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

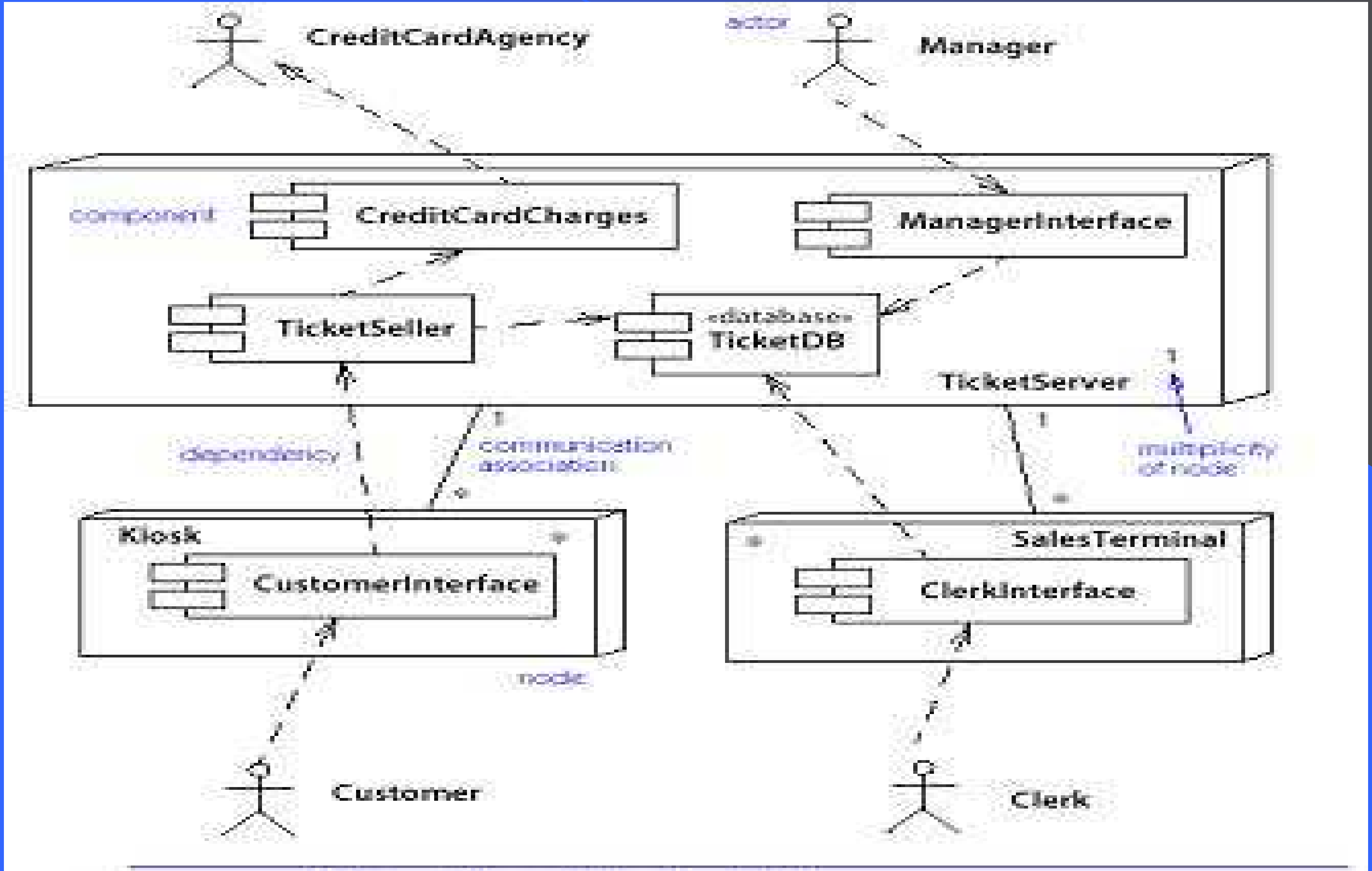
Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following parameters

- Performance
- Scalability
- Maintainability
- Portability

Deployment Diagram - Example

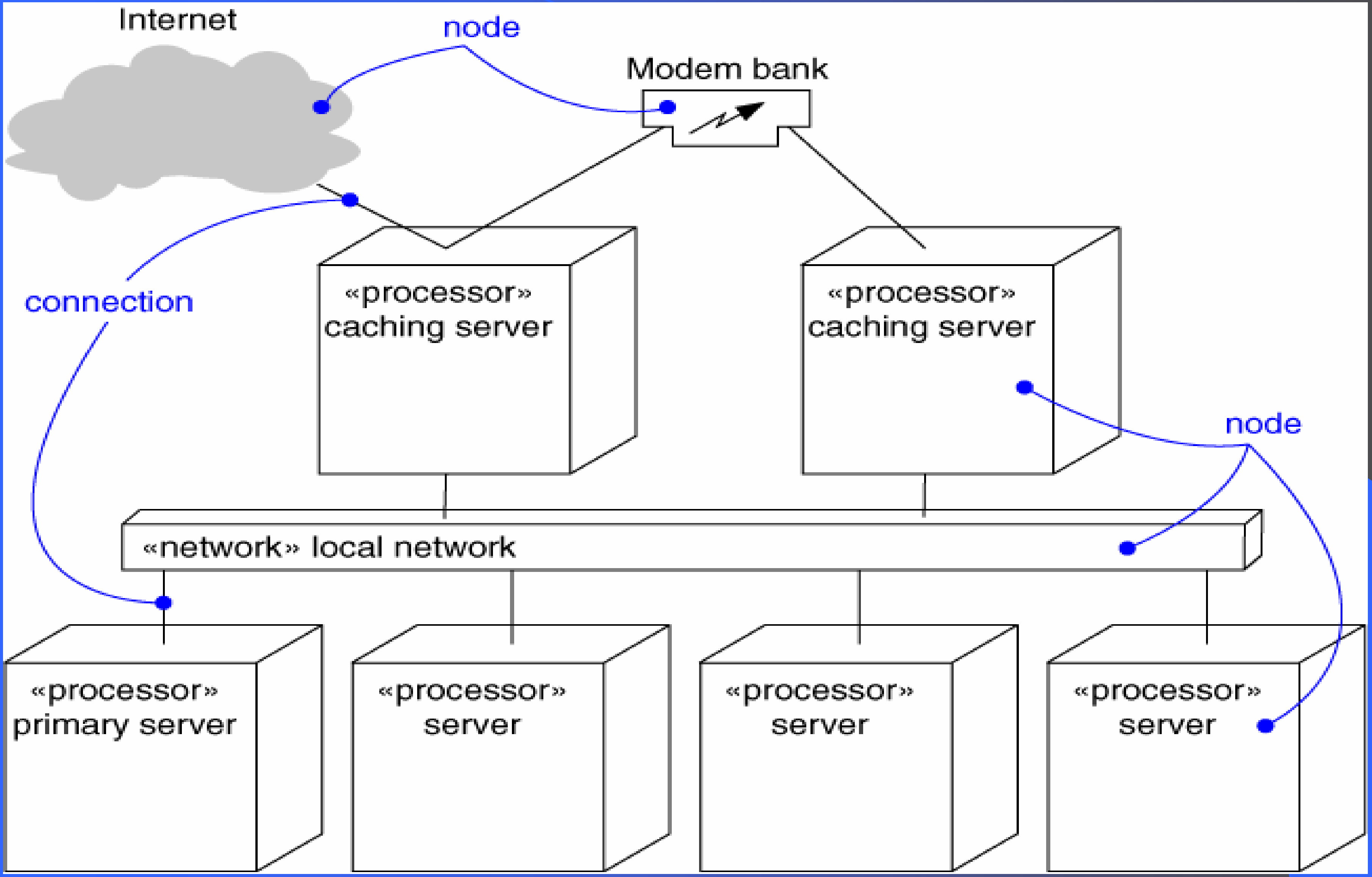


Deployment Diagram -Example



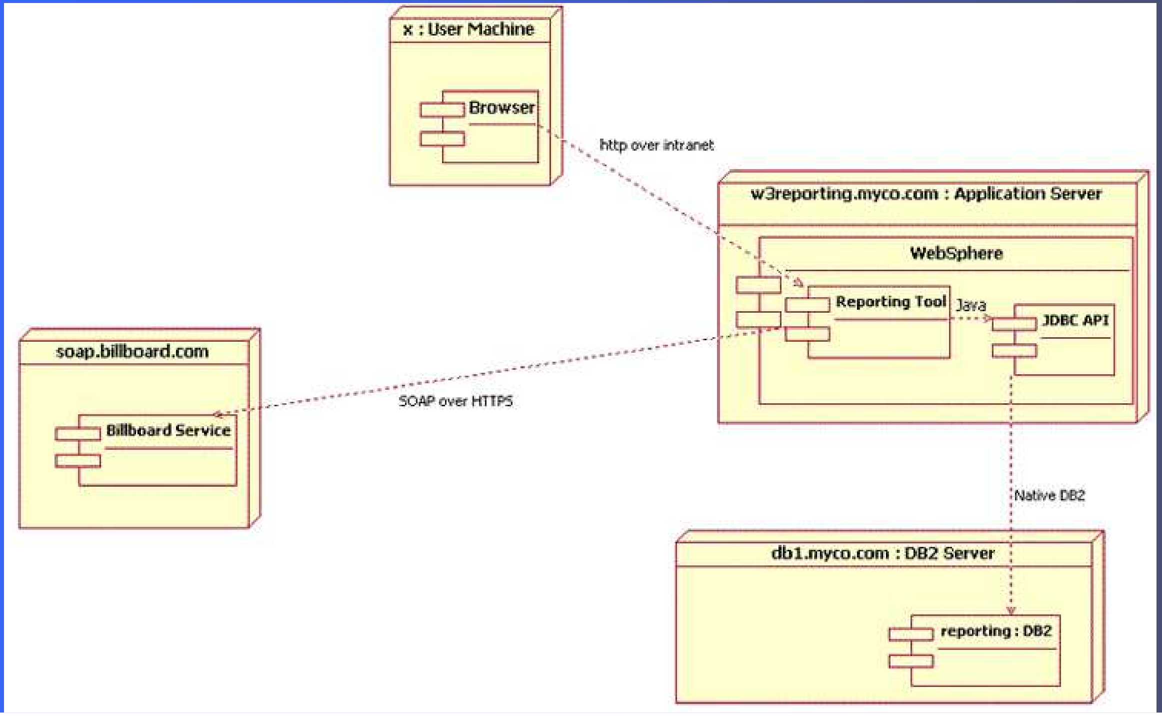
Deployment diagram (descriptor level)

Deployment Diagram -Example



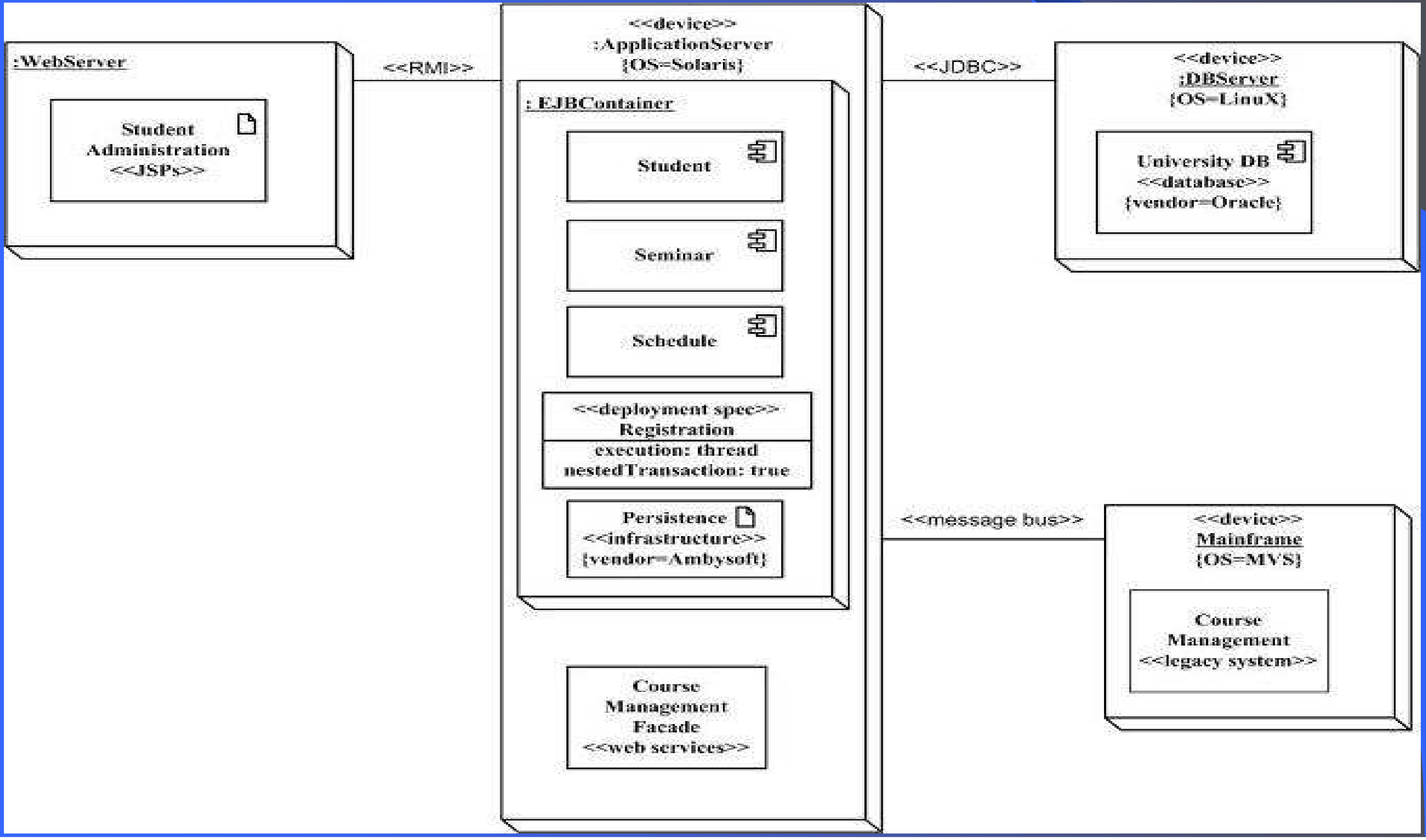
Deployment Diagram Example

The deployment diagram below shows that the users access the Reporting Tool by using a browser running on their local machine and connecting via HTTP over their company's intranet to the Reporting Tool. This tool physically runs on the Application Server named w3reporting.myco.com. The diagram shows the Reporting Tool component drawn inside of IBM WebSphere, which in turn is drawn inside of the node w3.reporting.myco.com. The Reporting Tool connects to its reporting database using the Java language to IBM DB2's JDBC interface, which then communicates to the actual DB2 database running on the server named db1.myco.com using native DB2 communication. In addition to talking to the reporting database, the Report Tool component communicates via SOAP over HTTPS to the Billboard Service.



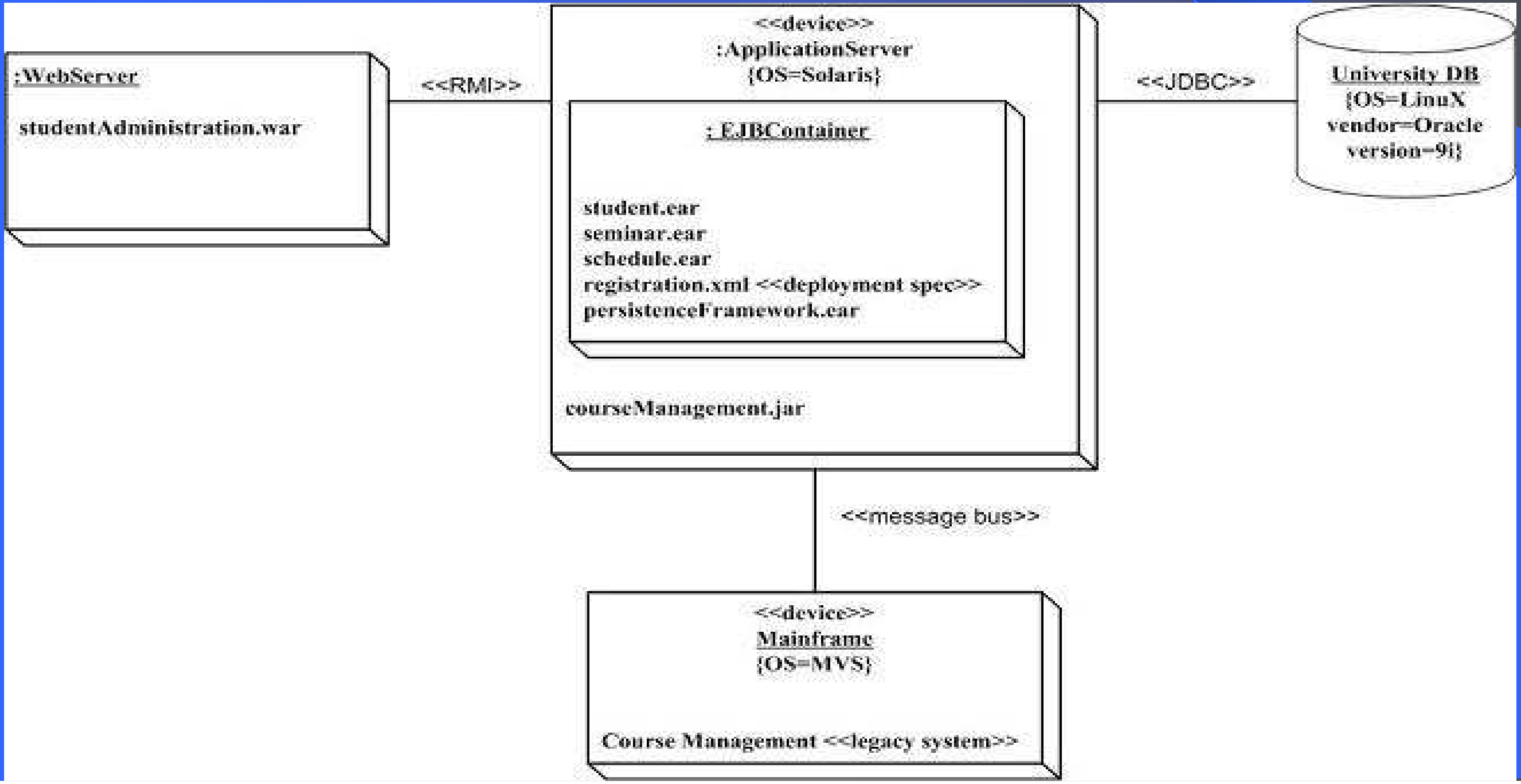
UML Deployment Diagram Example

- The three-dimensional boxes represent nodes, either software or hardware. Physical nodes should be labeled with the stereotype *device*, to indicate that it is a physical device such as a computer or switch.
- Connections between nodes are represented with simple lines, and are assigned stereotypes such as *RMI* and *message bus* to indicate the type of connection.

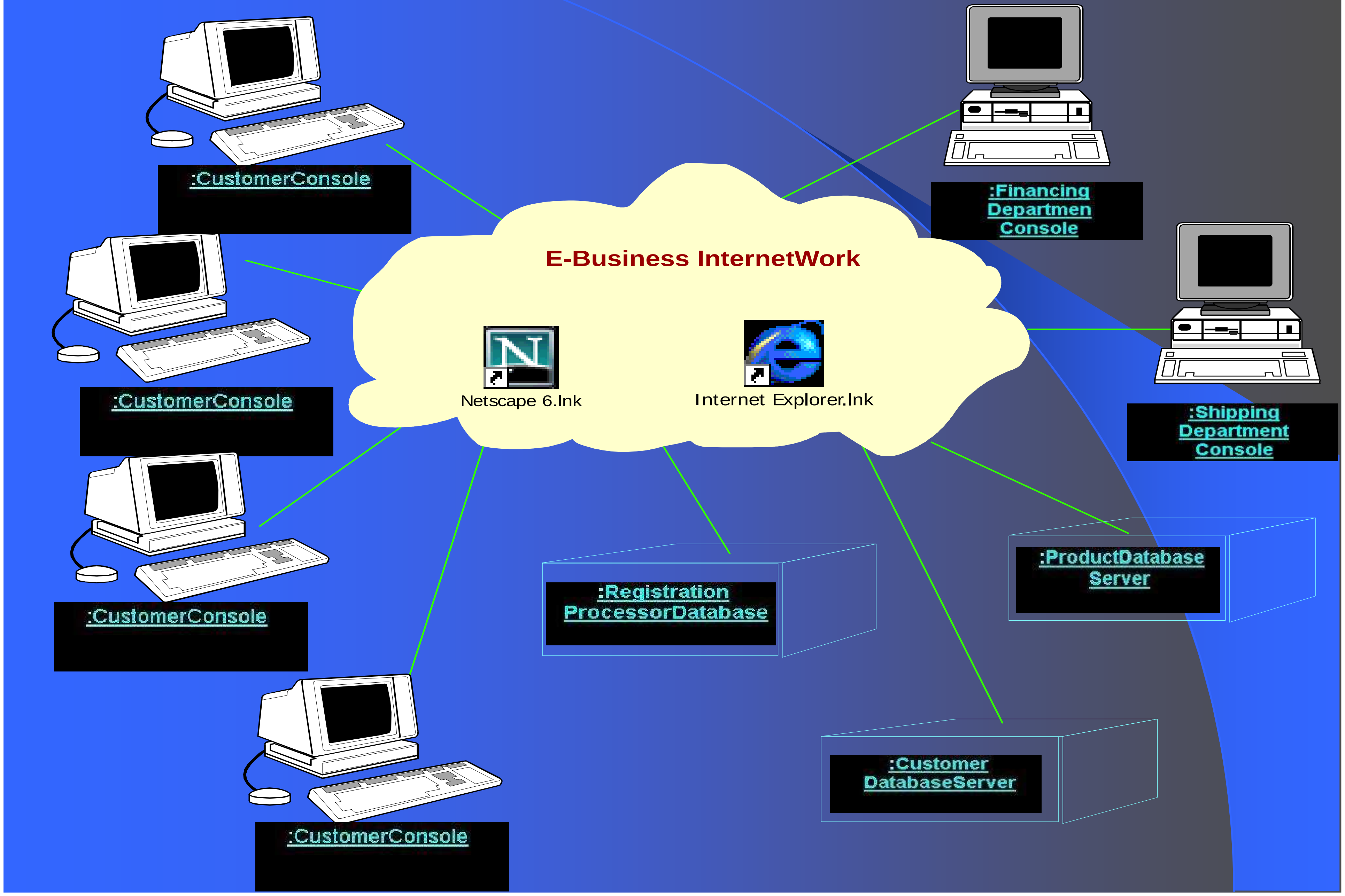


Concise UML 2 Deployment Diagram

- A better example is shown in the figure below. Software elements are now simply listed by their physical filenames, information that developers are very likely to be interested in, and thus a more compact diagram is possible.   A drum is used as a visual stereotype for the *University DB* database, making it easier to distinguish on the diagram. Another difference is that the concise version shows less details, not as many tagged values are shown as this information can be captured in either supporting documentation, configuration files, or source code.



Deployment Diagram

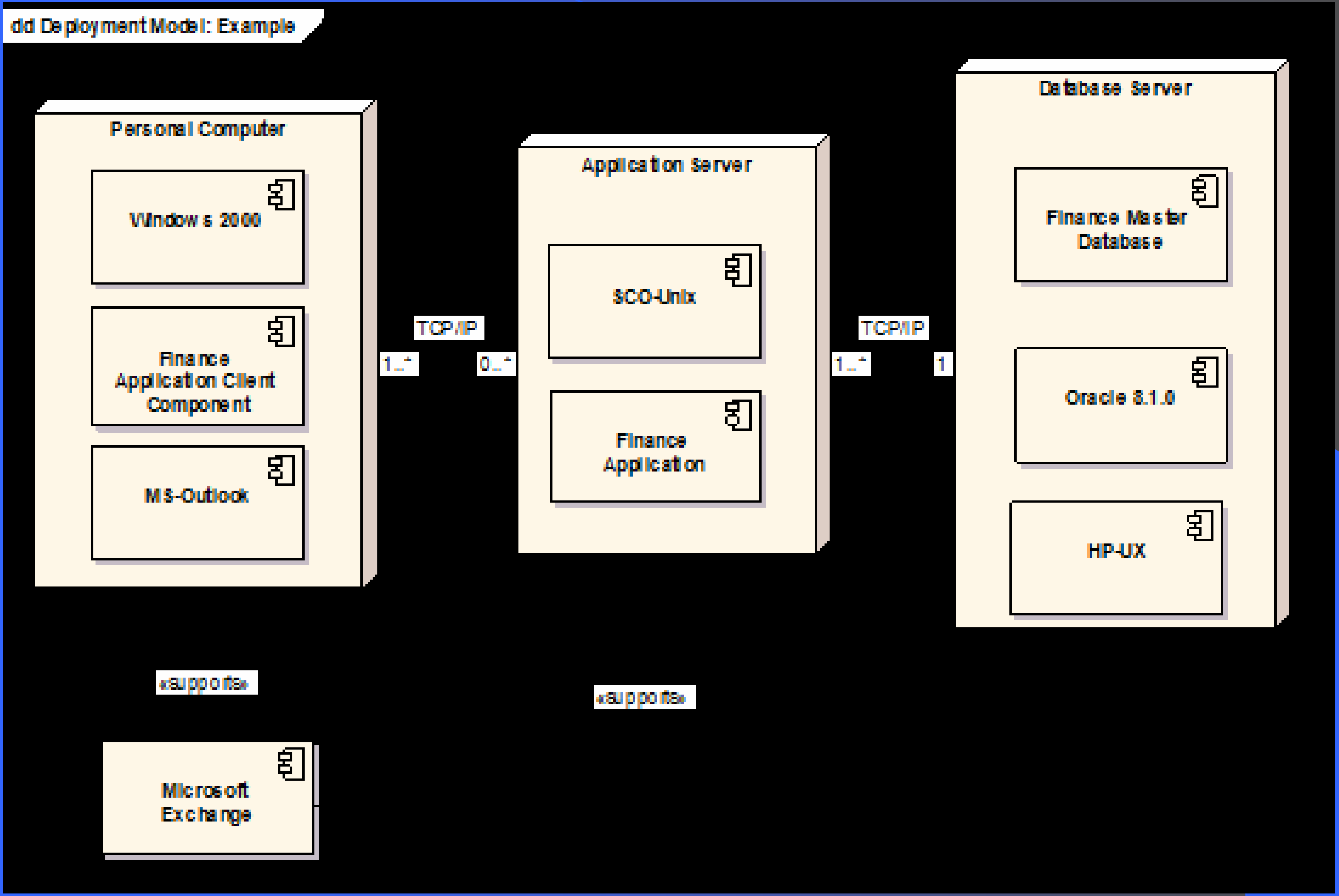


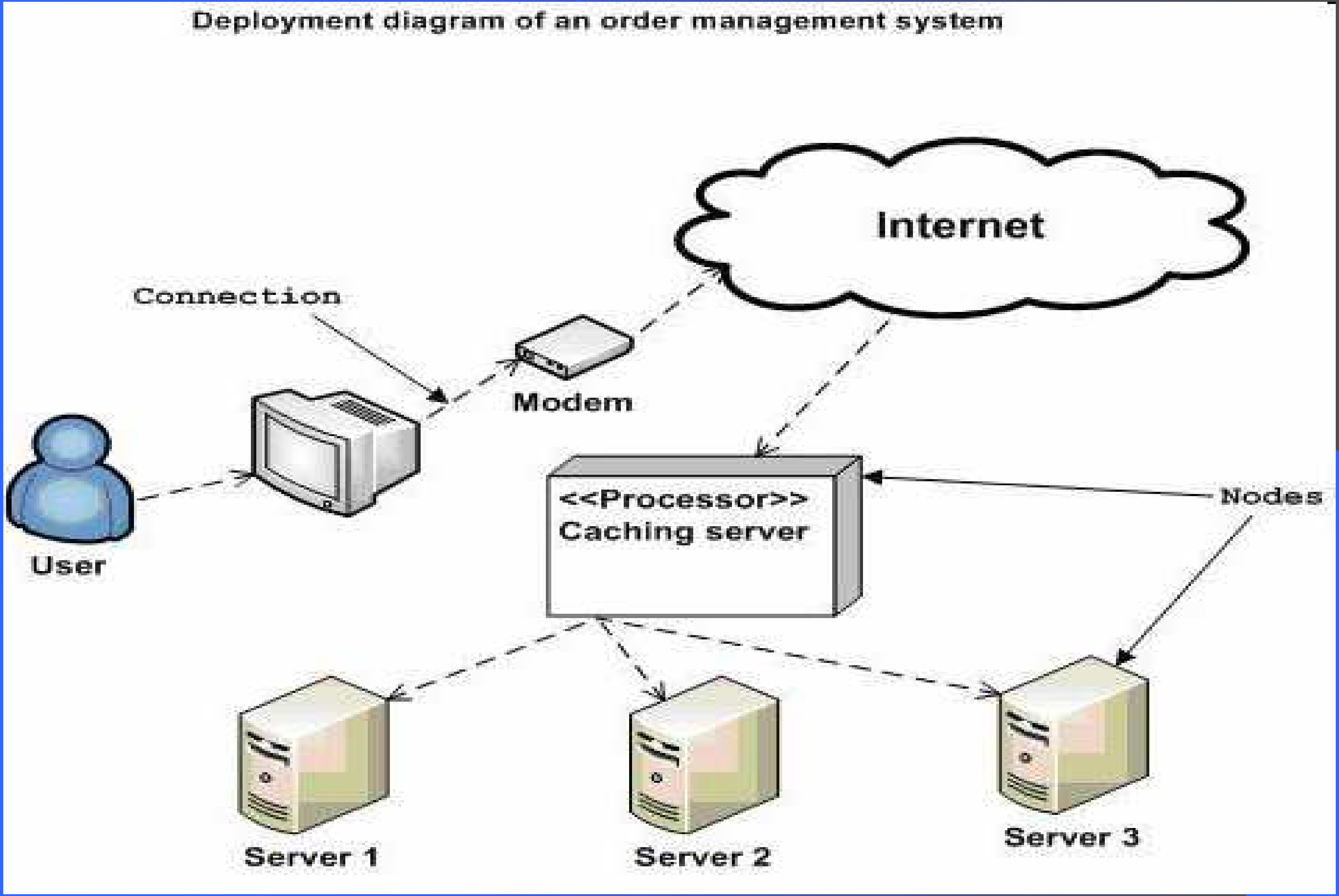
- The following deployment diagram is a sample to give an idea of the deployment view of order management system. Here we have shown nodes as:
- Monitor
- Modem
- Caching server
- Server
- The application is assumed to be a web based application which is deployed in a clustered environment using server 1, server 2 and server 3. The user is connecting to the application using internet. The control is flowing from the caching server to the clustered environment.
- So the following deployment diagram has been drawn considering all the points mentioned above:

Example

- The deployment model describes how an application is deployed a cross an infrastructure. The intention of the deployment model is n ot to describe the infrastructure, but rather the way in which specif ic components belonging to an application is deployed across it.
- In the example (Deployment diagram: Example) a physical deploy ment of a finance application is shown. Multiple customer/user per sonal computers with the runtime components Windows 2000 and the client component of the finance application can connect via TC P/IP to any application server, of which there are multiples. The ap plication server/s - running SCO-Unix and the finance application - connect via TCP/IP to the central database server - running HP-UX, Oracle and has the finance master database on it.

  Messaging and workflow between the client-PC's and between the application servers are performed using MS-outlook and MS-Exchange.
- MS- Exchange supports workflow and messaging.


dd Deployment Model: Example
Personal Computer
Windows 2000
Finance Application Client Component
MS-Outlook
Application Server
SCO-Unix
Finance Application
Database Server
Finance Master Database
Oracle 8.1.0
HP-UX
TCP/IP
1..*
0..*
TCP/IP
1..*
1
«supports»
«supports»
Microsoft Exchange


Deployment diagram of an order management system
Internet
Connection
Modem
<<Processor>>
Caching server
Nodes
User
Server 1
Server 2
Server 3

usage of deployment diagrams

- To model the hardware topology of a system.
- To model embedded system.
- To model hardware details for a client/server system.
- To model hardware details of a distributed application.
- Forward and reverse engineering.

- Deployment diagrams are distinct from component diagrams.
- A deployment diagram shows components and artifacts in relation to where they are used in the deployed system.
- A component diagram defines the composition of components and artifacts in the system.