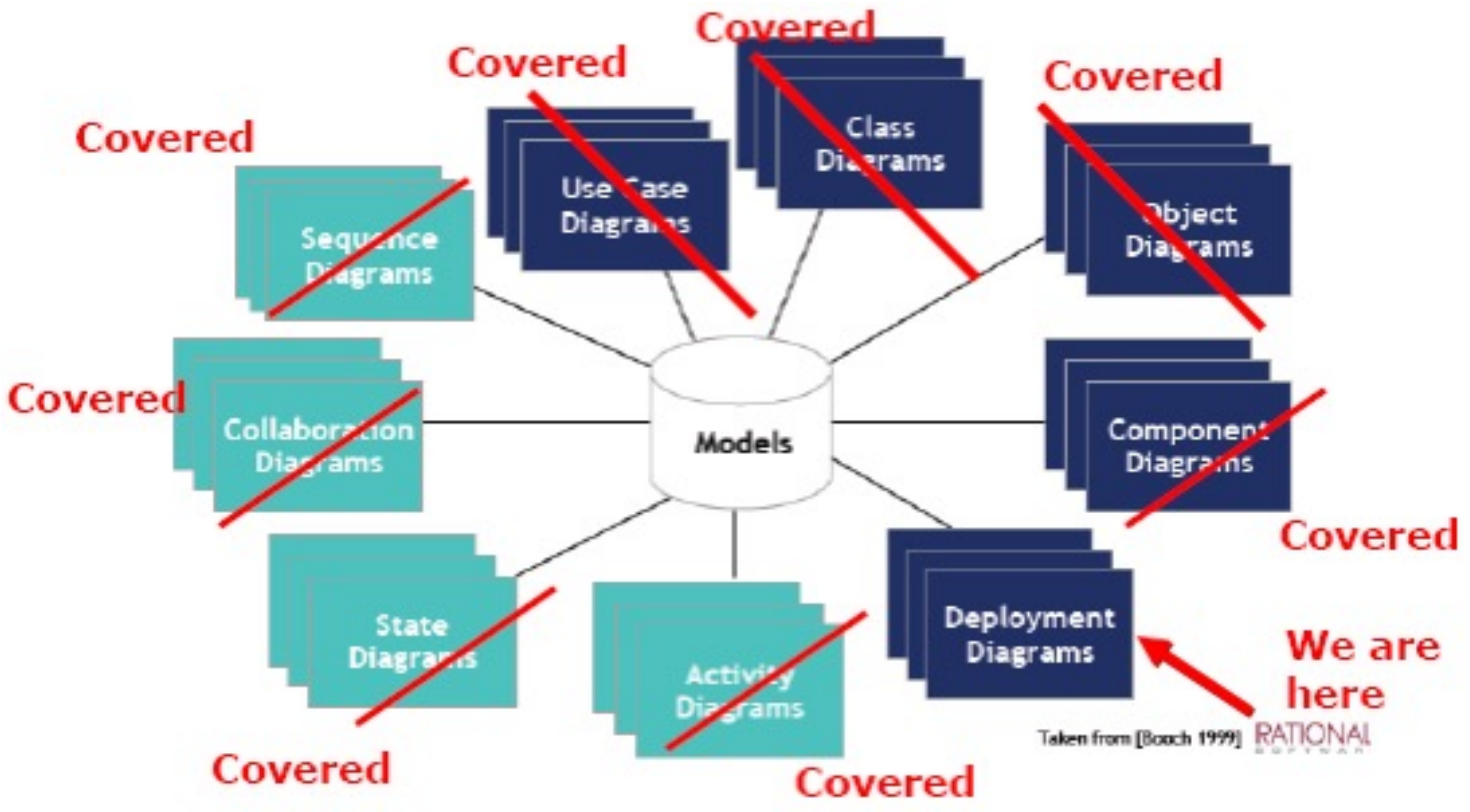
UML Diagrams



STUDENTS-HUB.com Uploaded By: anonymous

Deployment Diagram

Models the *run-time* configuration in a static view and visualizes the distribution of components in an application

It helps map between software components and hardware

A component is deployed part of the *software system architecture*

In most cases, it involves modelling the *hardware* configurations together with the *software* components that lived on

STUDENTS-HUB.com Uploaded By: anonymous

Deployment Diagram

Deployment diagram depicts a *static view* of the run-time configuration of processing nodes and the components that run on those nodes

Node: server, client etc.

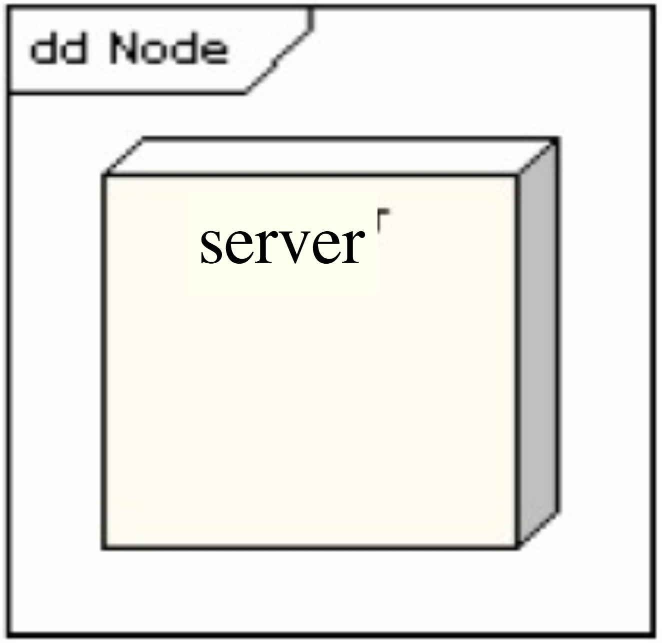
Deployment diagrams show the *hardware* for your system, the *software* that is installed on that hardware, and the *middleware* used to connect the disparate *machines* to one another!

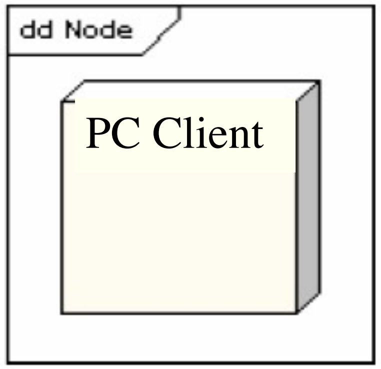
Visualizes the distribution of components in an application, it shows the configuration of the *hardware* elements (nodes) and shows how software elements and artifacts are mapped onto those nodes.

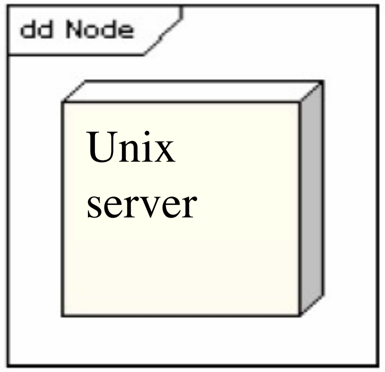
STUDENTS-HUB.com Uploaded By: anonymous

Node

A Node is either a hardware or software element. It is shown as a three-dimensional box shape, as shown below.





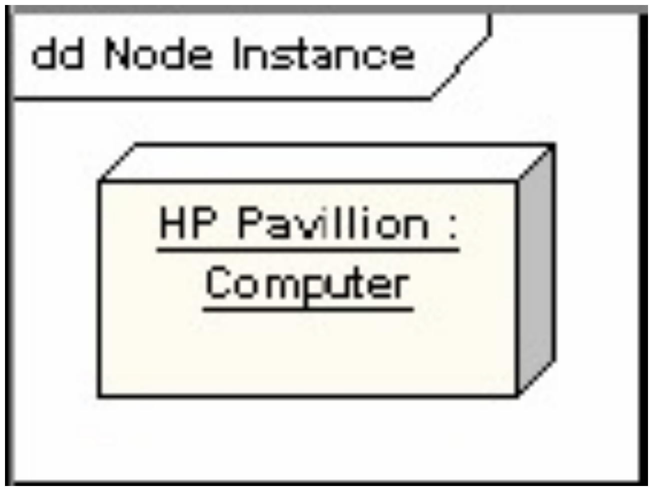


STUDENTS-HUB.com Uploaded By: anonymous

Node Instance

An **instance** can be distinguished from a node by the fact that its name is underlined and has a colon before its base node type. An instance may or may not have a name before the colon.

The following diagram shows a named instance of a computer

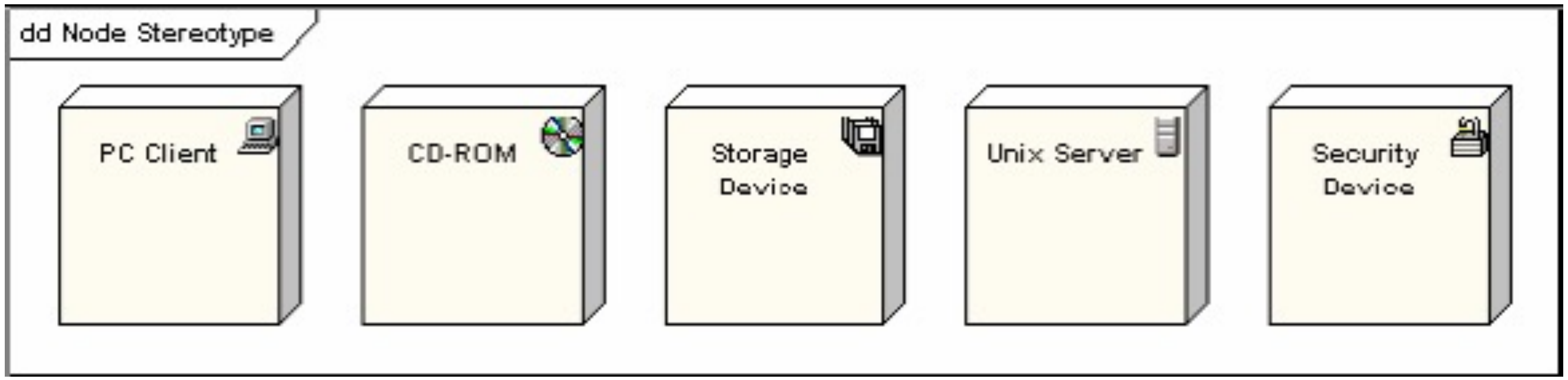


STUDENTS-HUB.com Uploaded By: anonymous

Node Sterotypes

In UML, a number of standard **stereotypes** are provided for nodes, namely «cdrom», «cd-rom», «computer», «disk array», «pc», «pc client», «pc server», «secure», «server», «storage», «unix server», «user pc».

These will display an appropriate icon in the top right corner of the node symbol

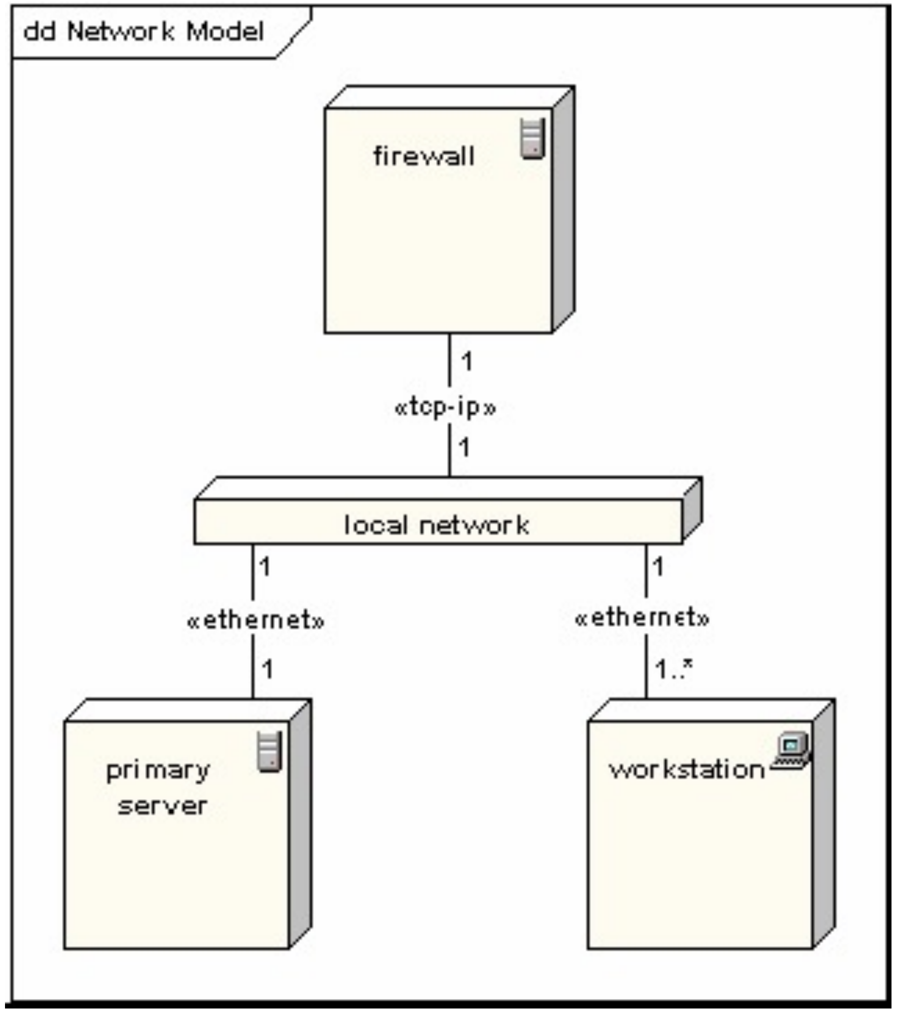


STUDENTS-HUB.com Uploaded By: anonymous

Association

In deployment diagram, an association represents a communication path between nodes.

The diagram shows a deployment diagram for a network, depicting network protocols as stereotypes, and multiplicities at the association ends.

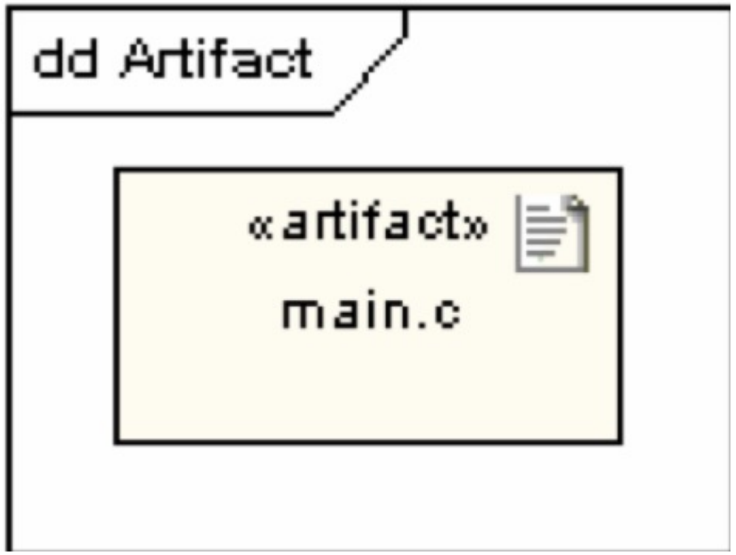


STUDENTS-HUB.com Uploaded By: anonymous

Artifact

An **artifact** is a product of the software development process. That may include process models (e.g. use case models, design models etc.), source files, executable files, design documents, test reports, prototypes, user manuals, etc.

An artifact is denoted by a rectangle showing the artifact name, the «artifact» keyword and a document icon, as shown.

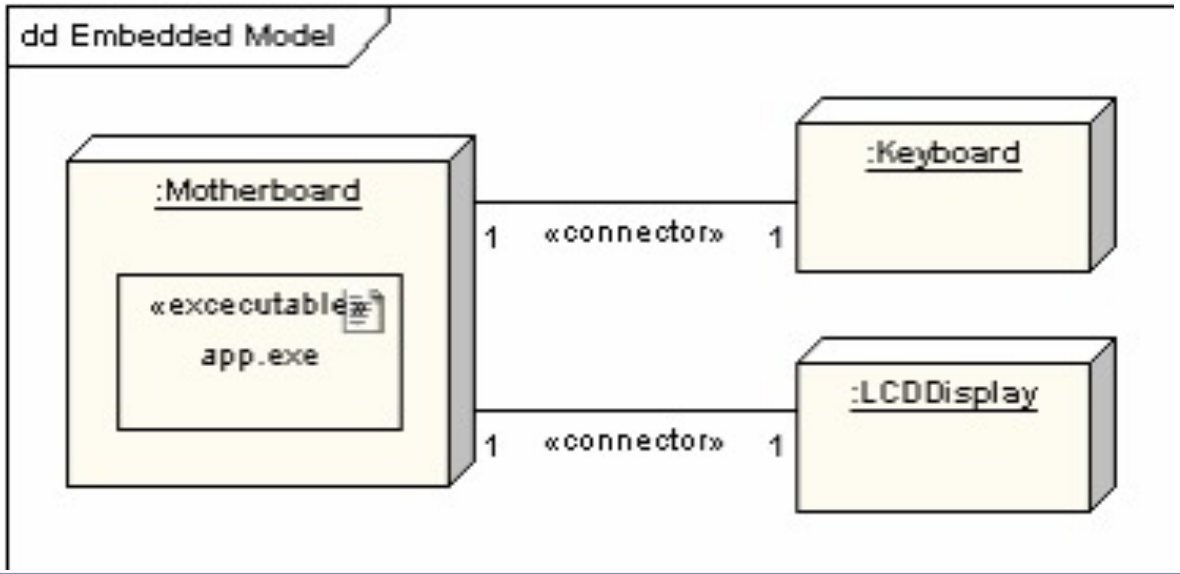


STUDENTS-HUB.com Uploaded By: anonymous

Node as Container

A node can contain other elements, such as components or artifacts.

The diagram shows a deployment diagram for part of an embedded system, depicting an executable artifact as being contained by the motherboard node.



STUDENTS-HUB.com Uploaded By: anonymous

Architectural Style vs Architecture

Architectural Style:

A pattern for a system layout

Software Architecture:

Instance of an architectural style.

STUDENTS-HUB.com Uploaded By: anonymous

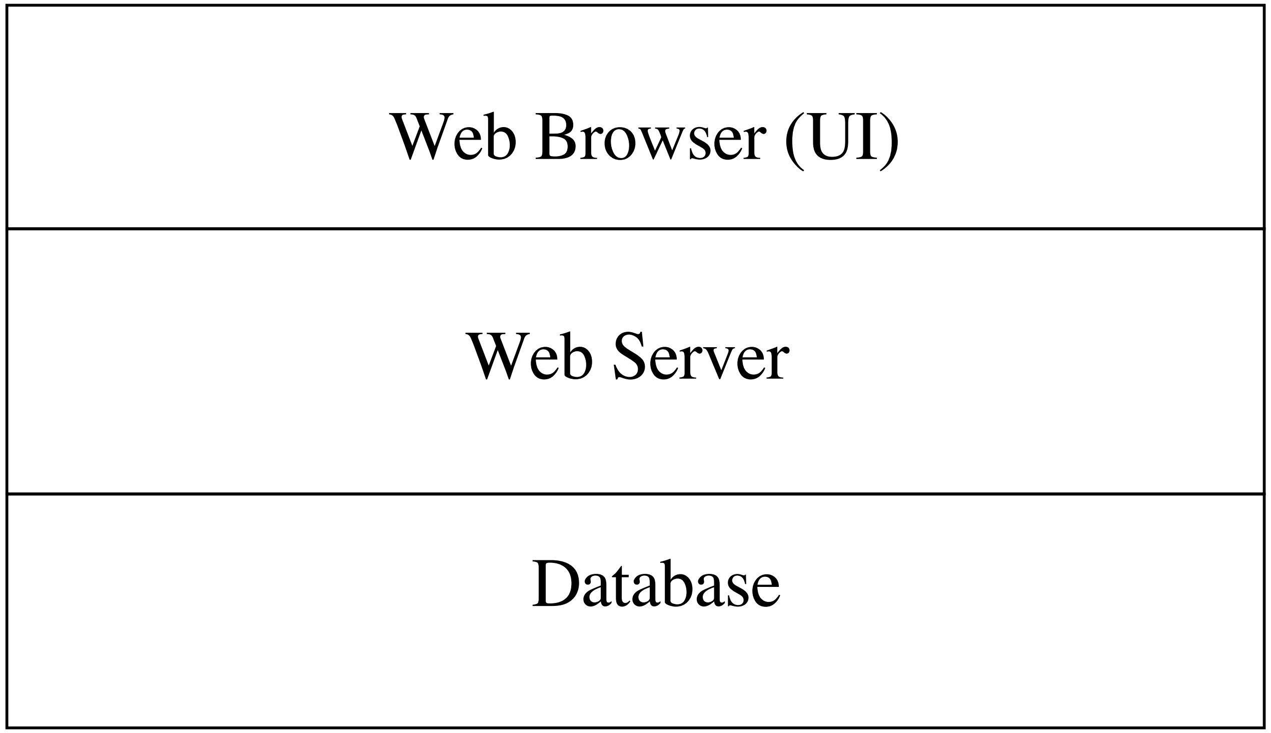
Examples of Architectural Styles

- Layered Architectural style
  - Service-Oriented Architecture (SOA)
- Client/Server
- Peer-To-Peer
- Three-tier, Four-tier Architecture
- Repository
- Model-View-Controller
- Pipes and Filters

STUDENTS-HUB.com Uploaded By: anonymous

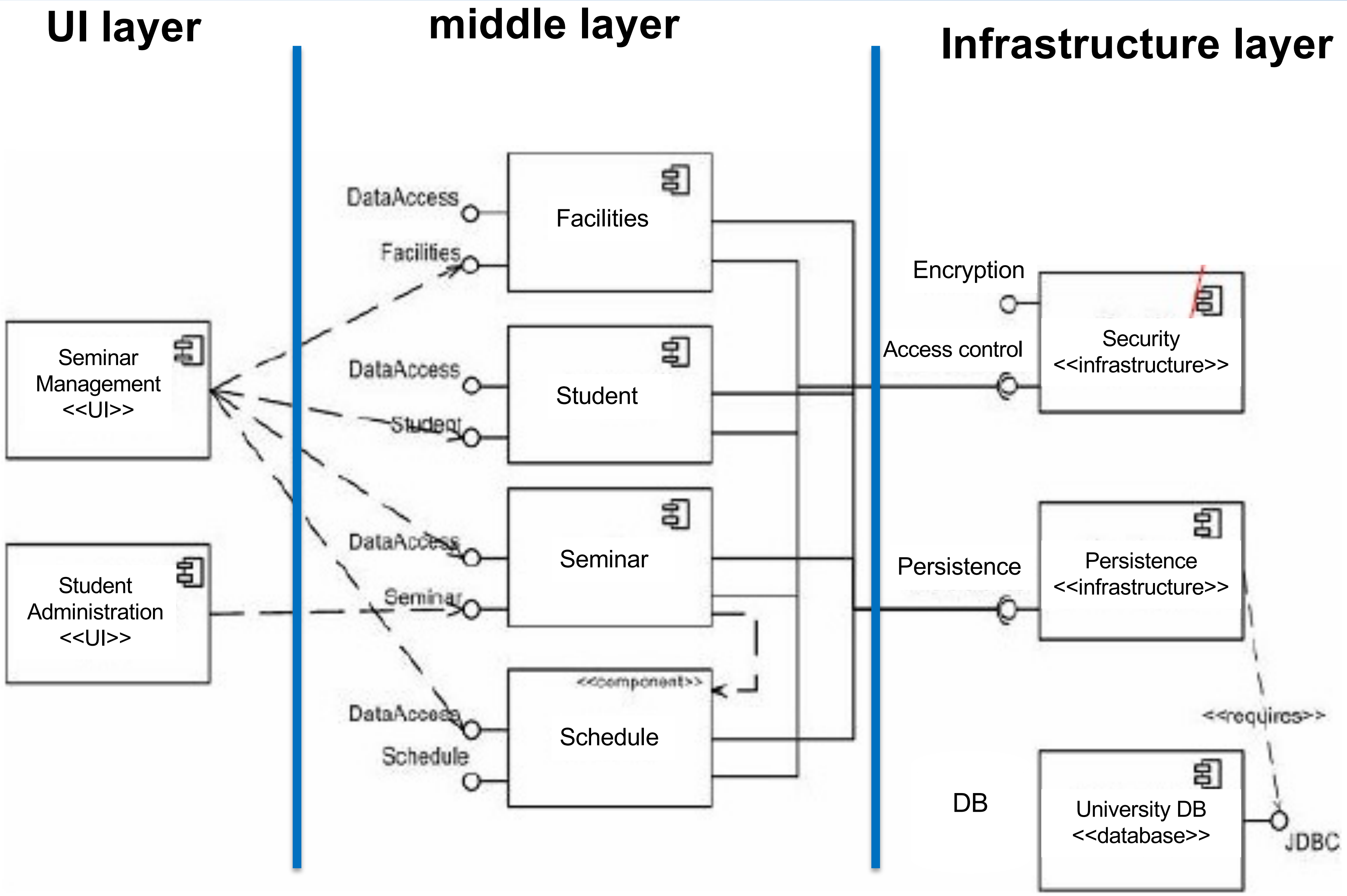
Example: Layered Architecture

Simplified architecture diagram
3-Layered Architecture:
often used for the development of Web-based Applications



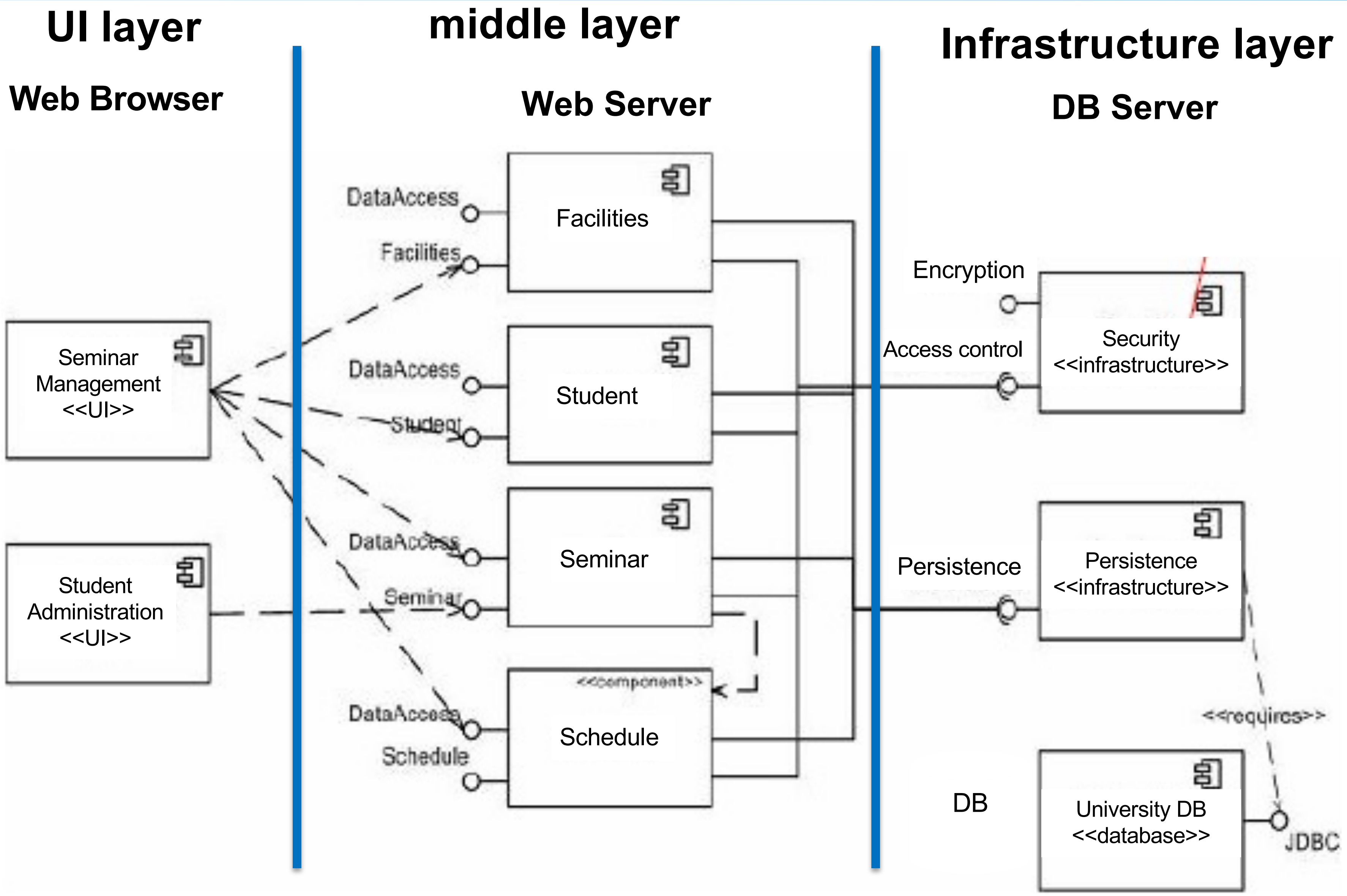
STUDENTS-HUB.com Uploaded By: anonymous

Example: Layered Architecture



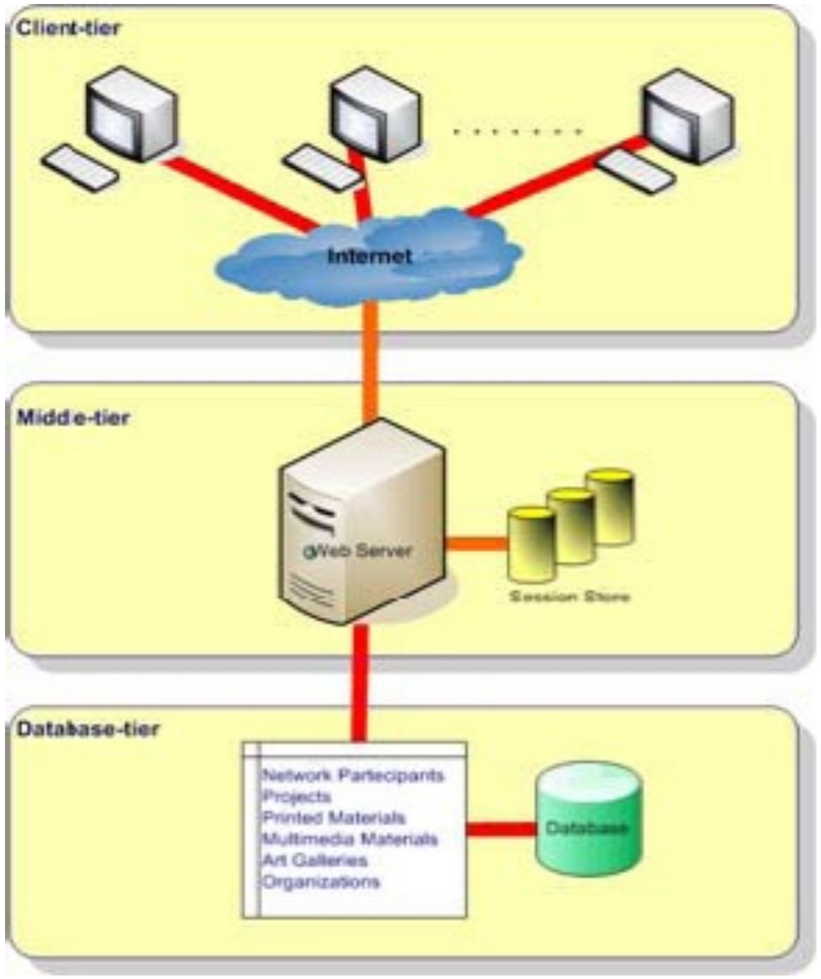
STUDENTS-HUB.com Uploaded By: anonymous

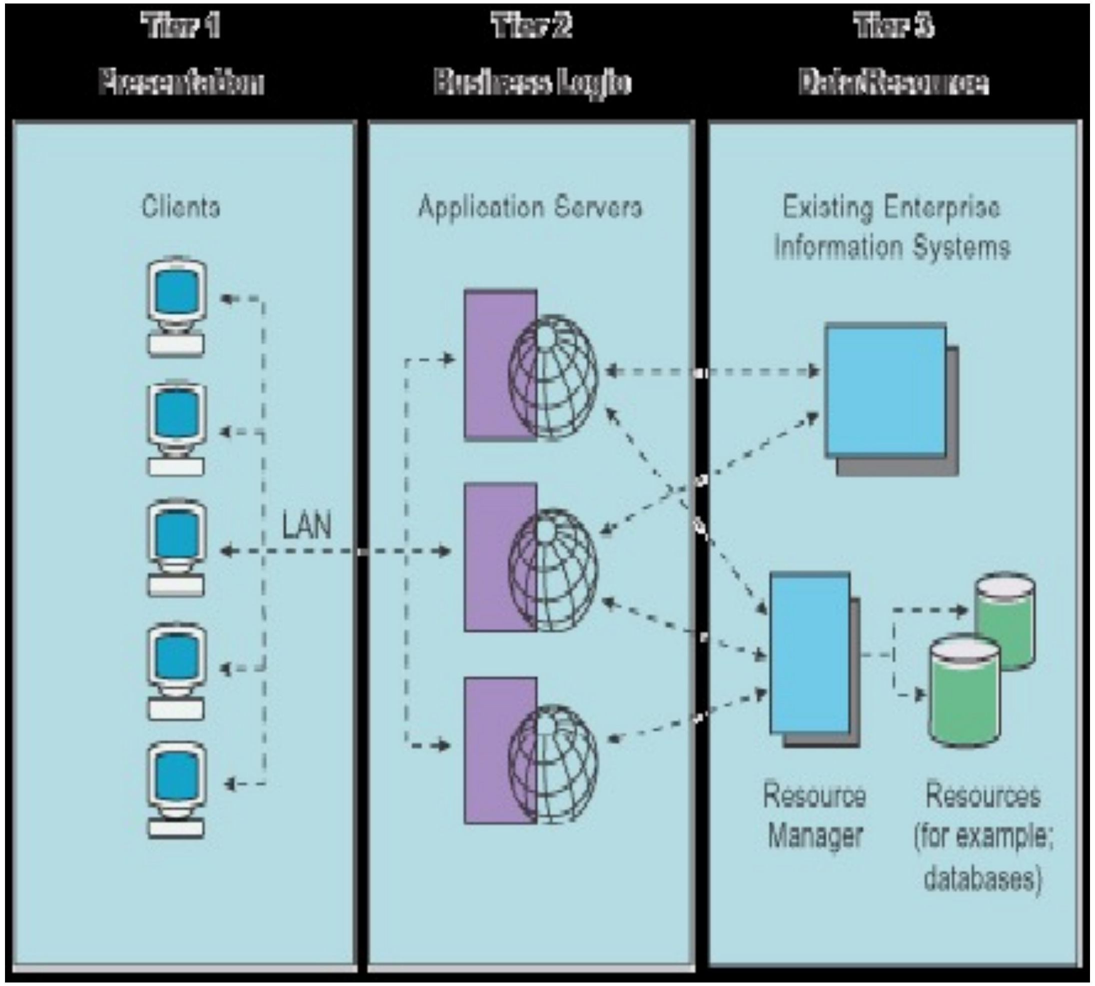
Example: Layered Architecture (Web App)



STUDENTS-HUB.com Uploaded By: anonymous

Example of three-tiers architectures



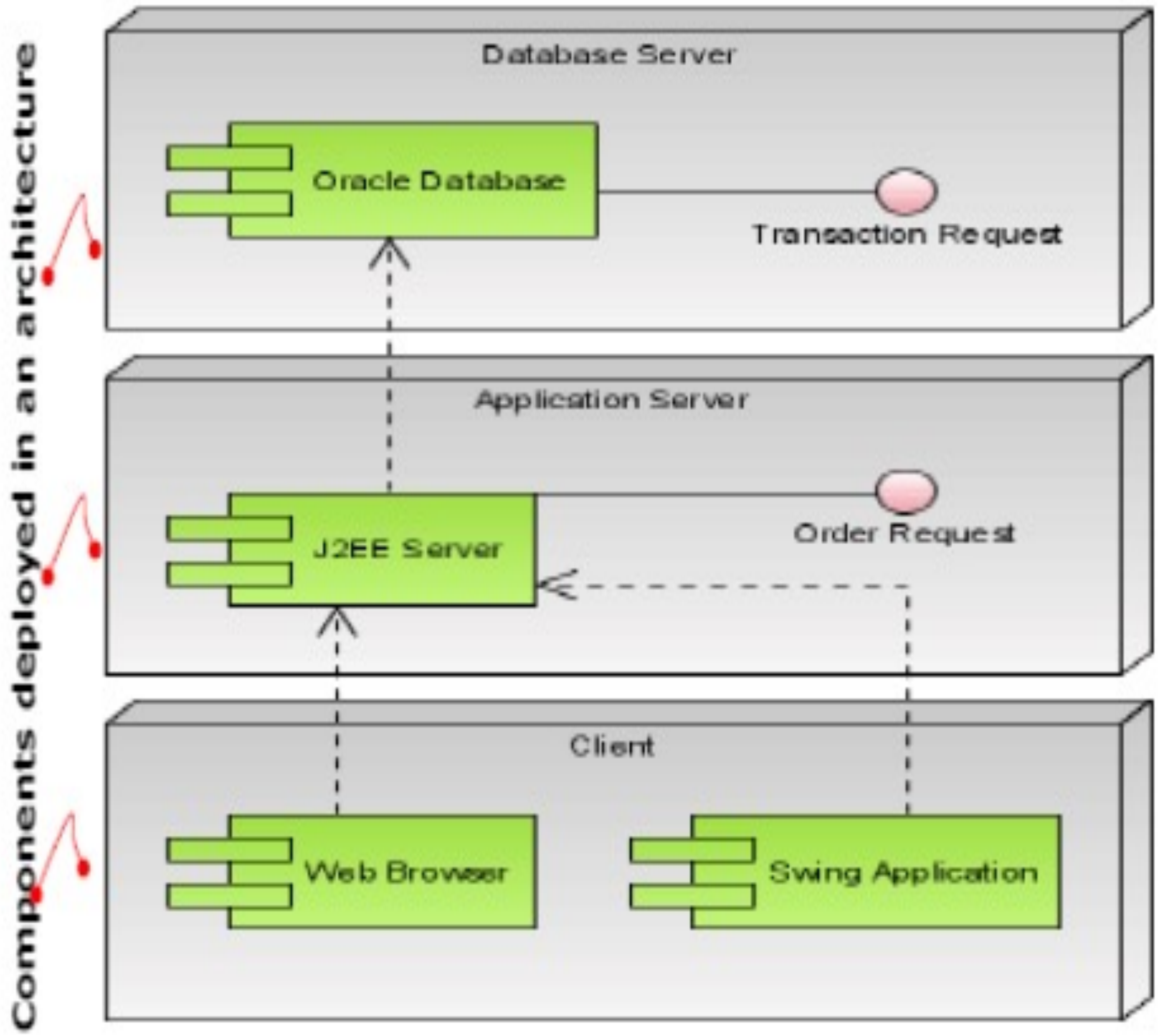


Many of real life web applications have three tier architectures

STUDENTS-HUB.com Uploaded By: anonymous

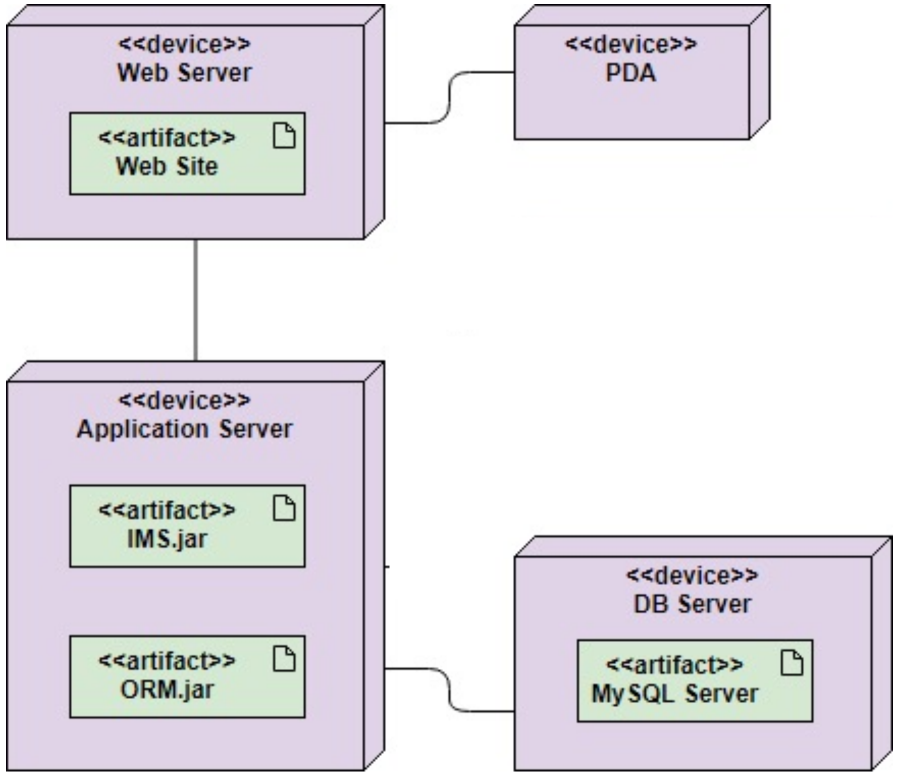
Deployment diagram for three tiers

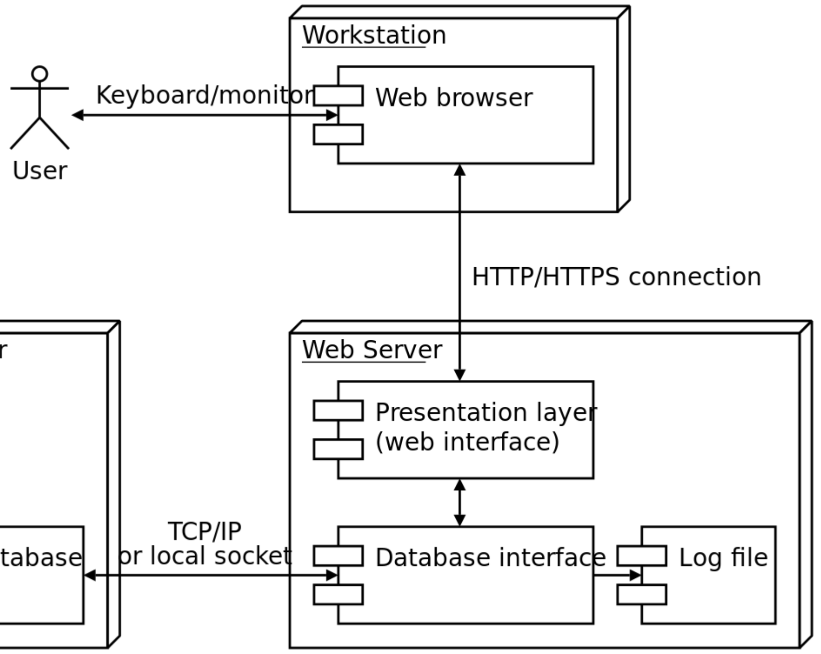
UML 1.4

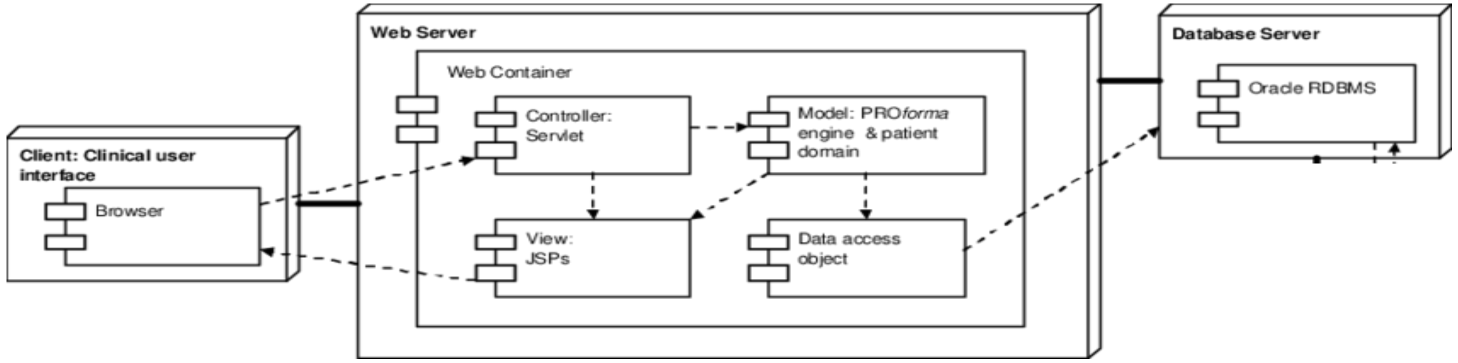


STUDENTS-HUB.com Uploaded By: anonymous

Examples: Deployment diagram for three/four tiers

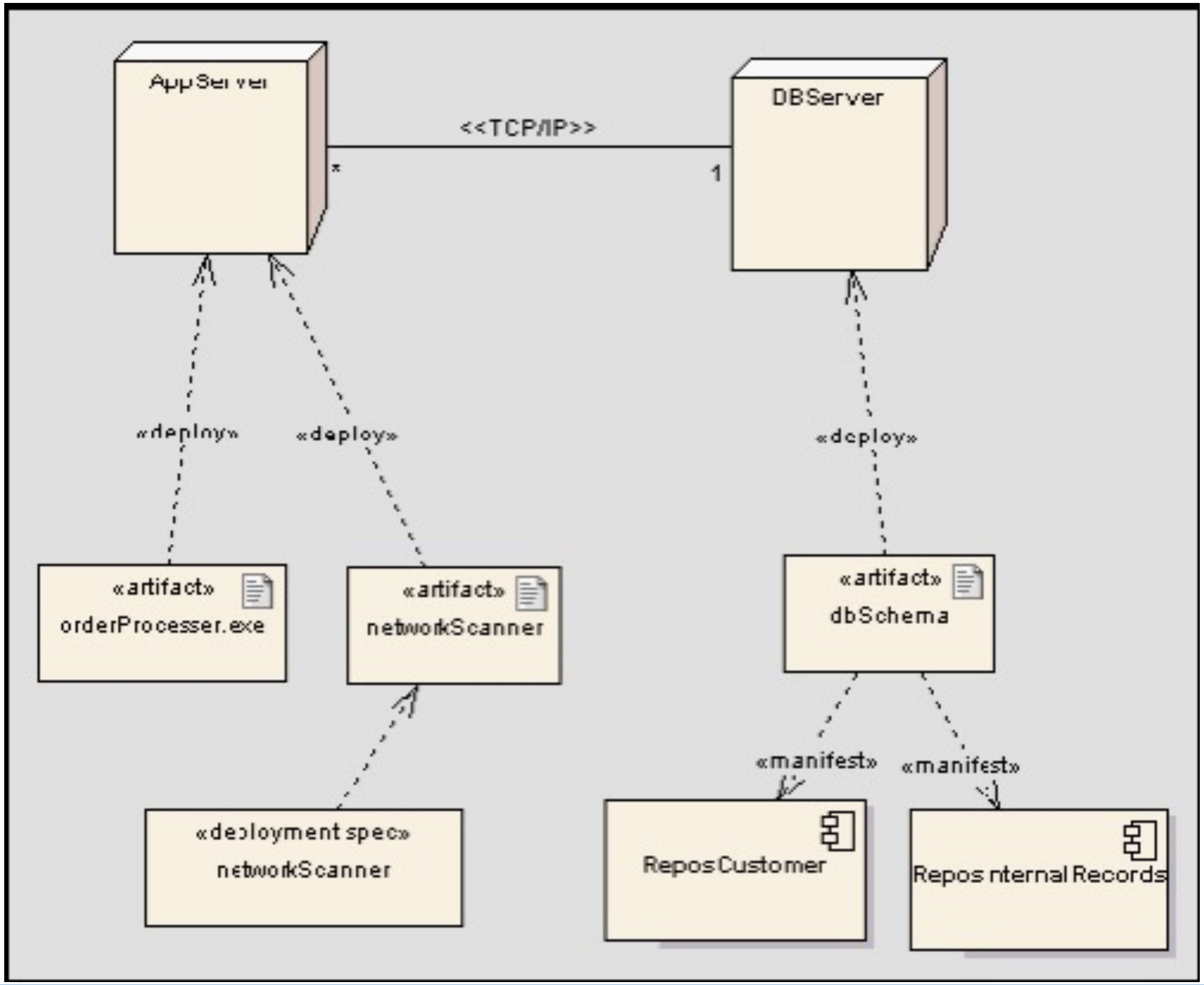






STUDENTS-HUB.com Uploaded By: anonymous

Example: Deployment Diagram for client server architectures



STUDENTS-HUB.com Uploaded By: anonymous

Exercise: Depict a deployment diagram for the following Scenario

A library System is to be developed and deployed as a web and a desktop UI applications to support Library Members and Librarians. The desktop UI connects to the library middleware directly, while the web UI requires to connect to the middleware through a web server (Apache).

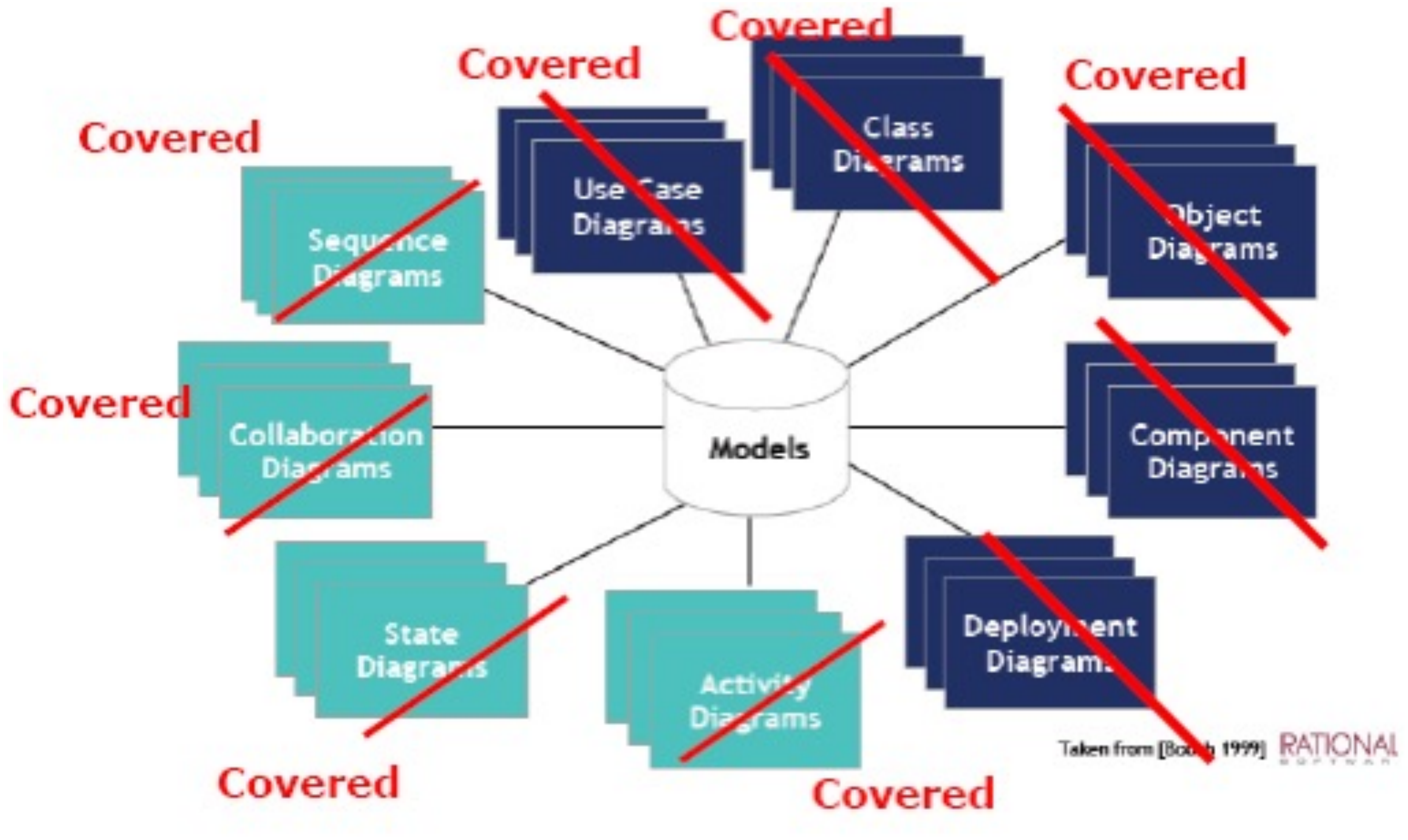
Desktop UI, Web UI and middleware are separate physical layers.

The Library system is required to have an operational DB and a backup DB, that both must be physically located in two separate locations.

The library system has two UI components: LibraryMembersUI and LibrarianUI; four application domain components (middleware): Books, Journals, LibraryMembers & Librarians, and three infrastructure components DBConnector & Authorization, that connect to the database servers, and LibraryDB, located in both databases.

STUDENTS-HUB.com Uploaded By: anonymous

Covered!?



STUDENTS-HUB.com Uploaded By: anonymous

Key points

A model is an abstract view of a system that ignores system details. Complementary system models can be developed to show the system's context, interactions, structure and behaviour.

Context models show how a system that is being modeled is positioned in an environment with other systems and processes.

Structural models show the organization and architecture of a system. Use cases describe interactions between a system and external actors. Class diagrams are used to define the static structure of classes in a system and their associations using both data-driven and executable view points.

Behavioural models show how how system elements interactions. Use case diagrams, activity diagrams and sequence diagrams are used to describe the interactions between users and systems in the system being designed taking the business view points or needs. Activity diagrams show how a business achieve its business process through interactions between use cases. Sequence diagrams show how a system achieve use cases through interactions between system objects.

STUDENTS-HUB.com Uploaded By: anonymous