# Deployment Diagram

# Deployment Diagram

- Purpose:
  - Visualize hardware topology of a system
  - Describe the hardware components used to deploy software components.

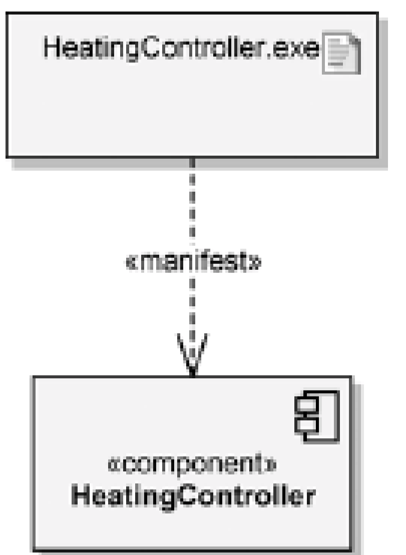
- Help system engineers understand:
  - Performance
  - Scalability
  - Maintainability
  - Portability

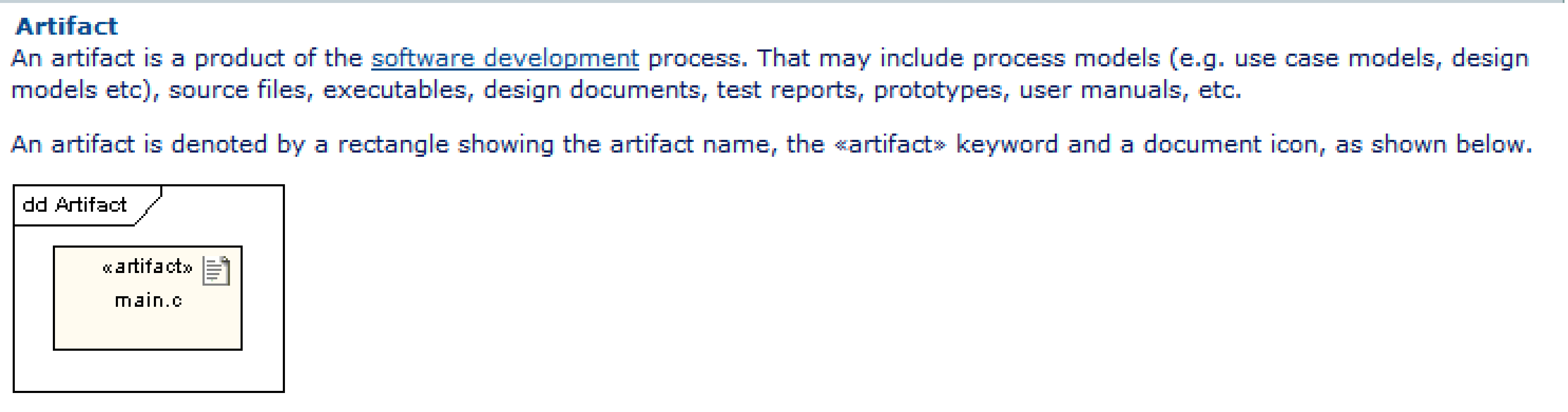
# Deployment Diagram

- Deployment diagrams indicate how the software is to be installed across systems—for example, what will be installed on the server and what will be installed on the admin PCs.
- A deployment diagram shows how the finished system will be deployed on one or more machines. A deployment diagram can include all sorts of features such as machines, processes, files and dependencies.
- A deployment diagram is used to show the allocation of artifacts to nodes in the physical design of a system. A  single deployment diagram represents a view into the artifact structure of a system. During development, we use deployment diagrams to indicate the physical collection of nodes that serve as the platform for execution of our system.

# Essentials: The Artifact Notation

- An artifact is a physical item that implements a portion of the software design. It is typically software code (executable) but could also be a source file, a document, or another item related to the software code. Artifacts may have relationships with other artifacts, such as a dependency or a composition.
- The notation for an artifact consists of a class rectangle containing the name of the artifact.





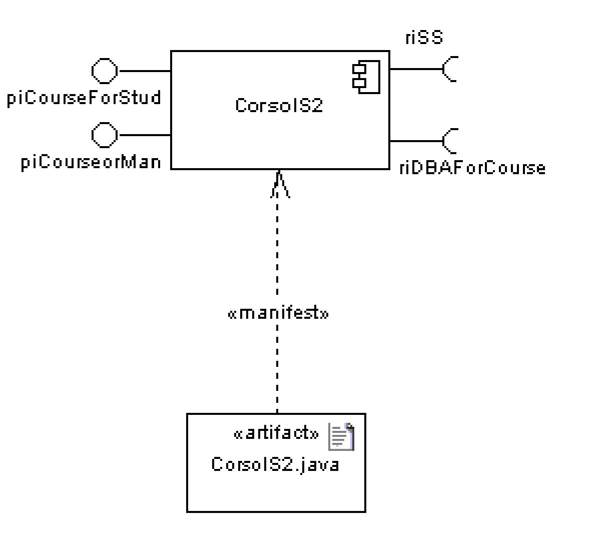
### Artifact

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, executables, 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 below.

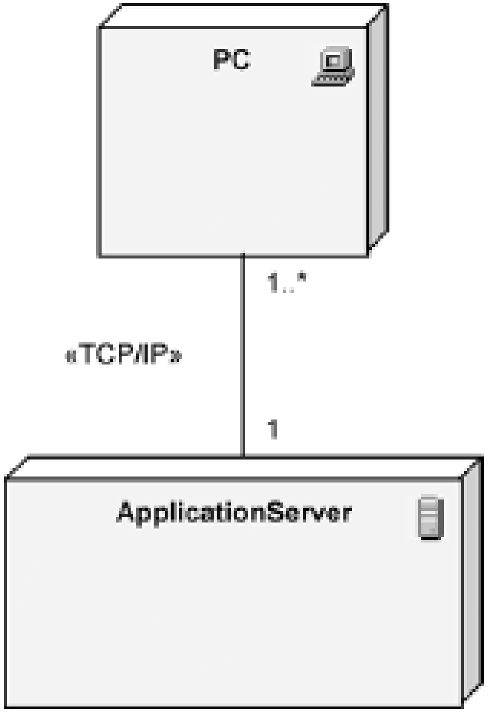
# DEPLOYMENT DIAGRAMS

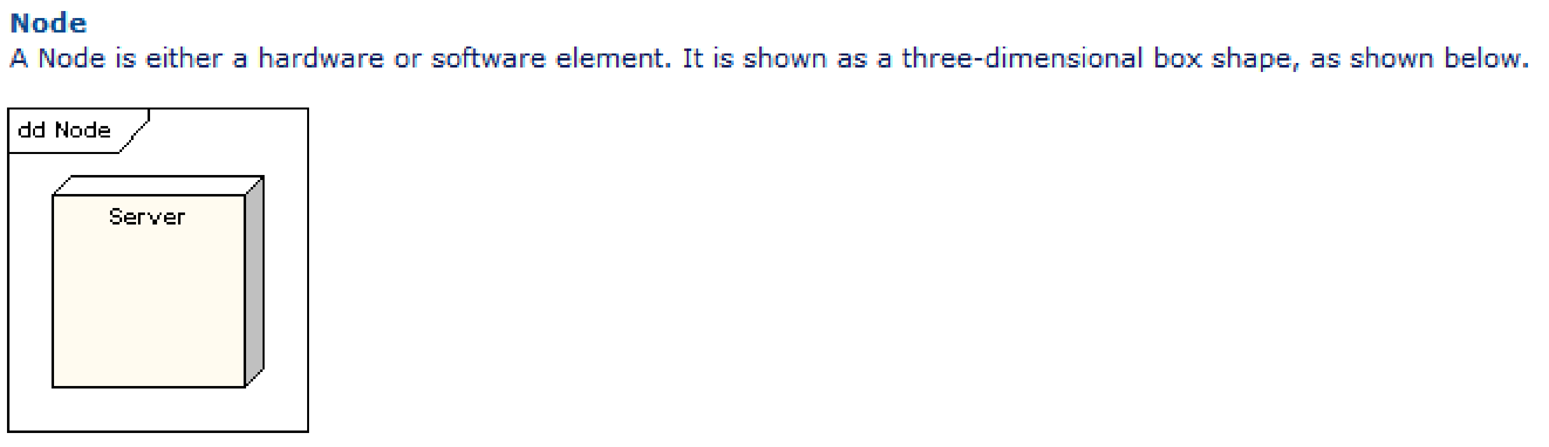
- An artifact manifest one or more model elements
- A <<manifestation>> is the concrete physical of one or more model elements by an artifact
- This model element often is a component
  - A manifestation is notated as a dashed line with an open arrow-head labeled with the keyword <<manifest>>



# Essentials: The Node Notation

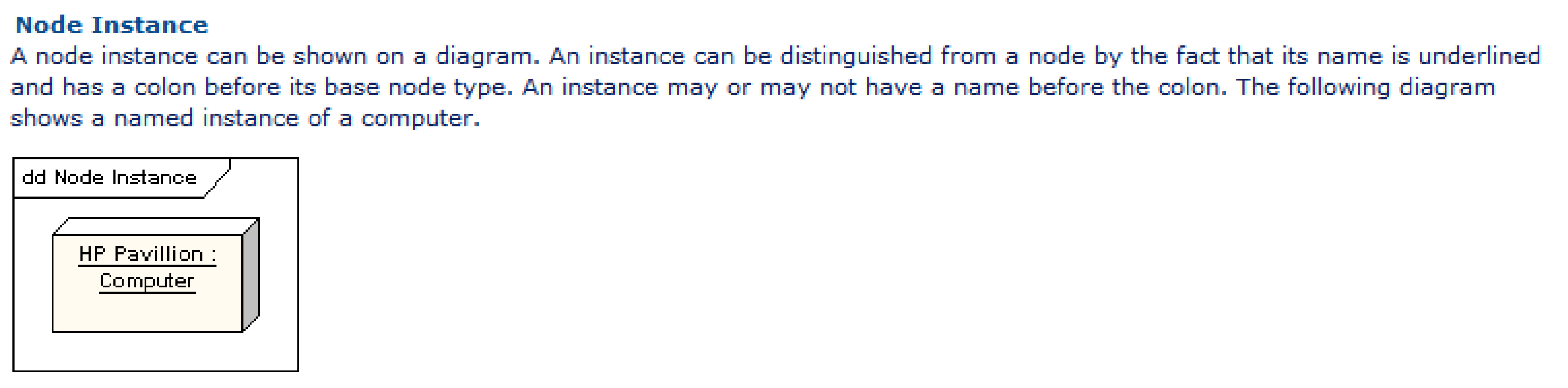
- A node is a computational resource, typically containing memory and processing,on which artifacts are deployed for execution. Nodes may contain other nodes to represent complex execution capability; this is shown by nesting or using a composition relationship. There are two types of nodes: devices and execution environments
- A device is a piece of hardware that provides computational capabilities, such as a computer, a modem, or a sensor. An execution environment is software that provides for the deployment of specific types of executing artifacts; examples include «database» and «J2EE server». Execution environments are typically hosted by a device.
- Nodes communicate with one another, via messages and signals, through a communication path indicated by a solid line. Communication paths are usually considered to be bidirectional, although if a particular connection is unidirectional, an arrow may be added to show the direction. Each communication path may include an optional keyword label, such as «http» or «TCP/IP», that provides information about the connection. We may also specify multiplicity for each of the nodes connected via a communication path.





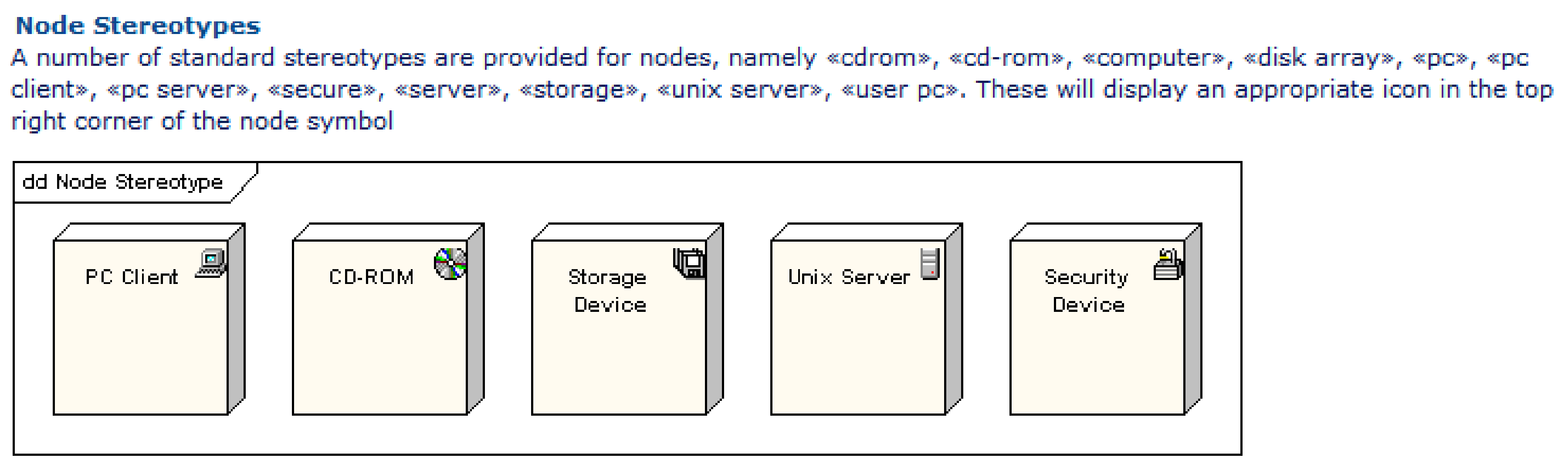
**Node**

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



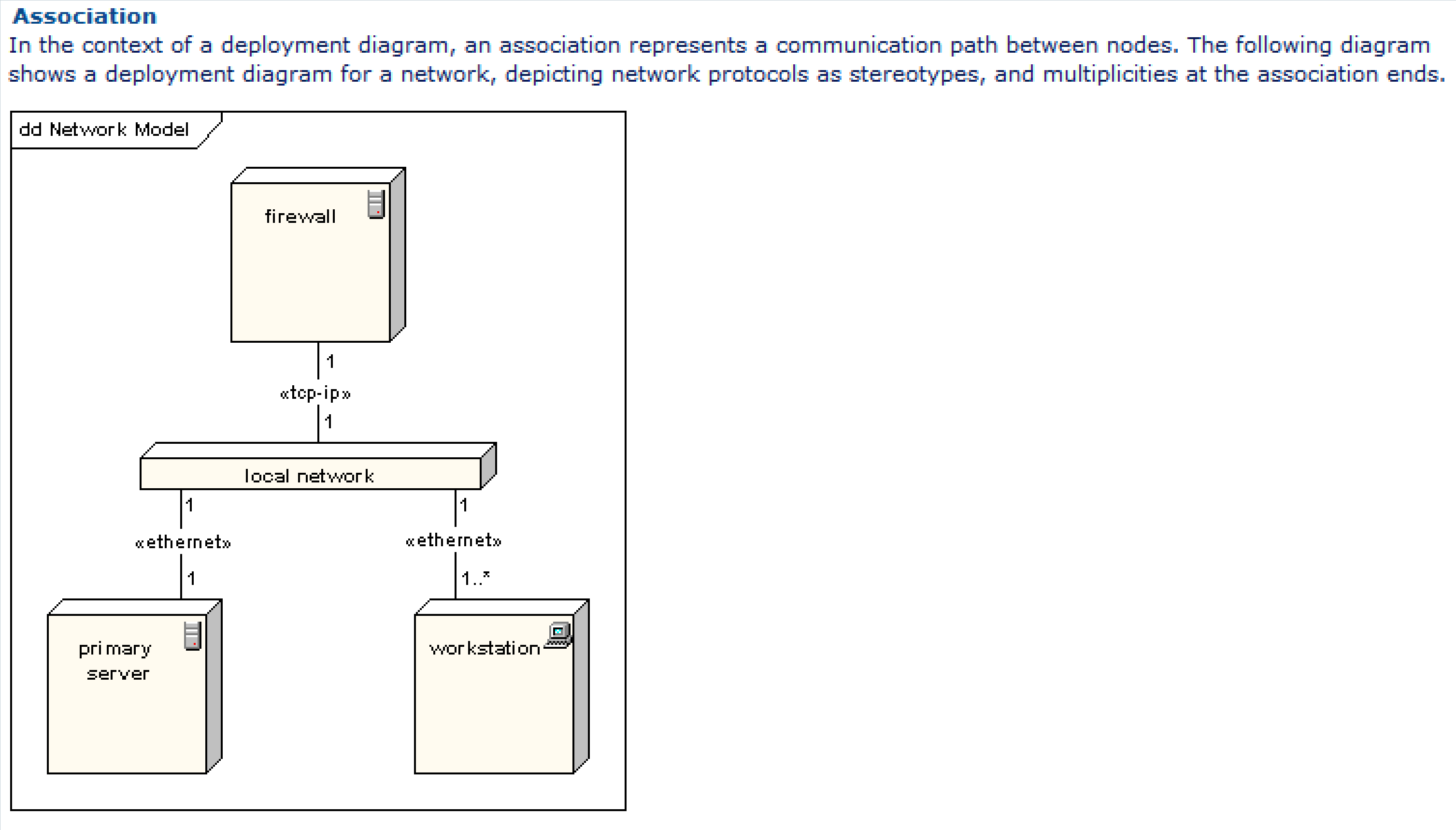
### Node Instance

A node instance can be shown on a diagram. 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.



**Node Stereotypes**

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

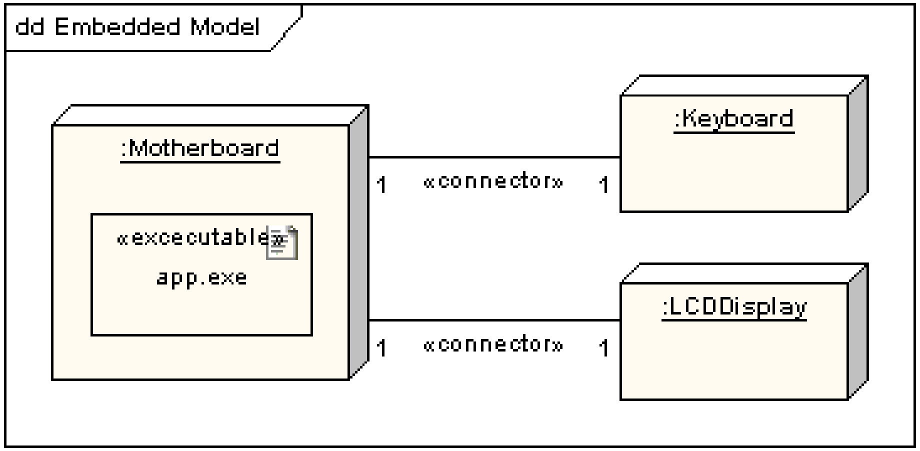


### Association

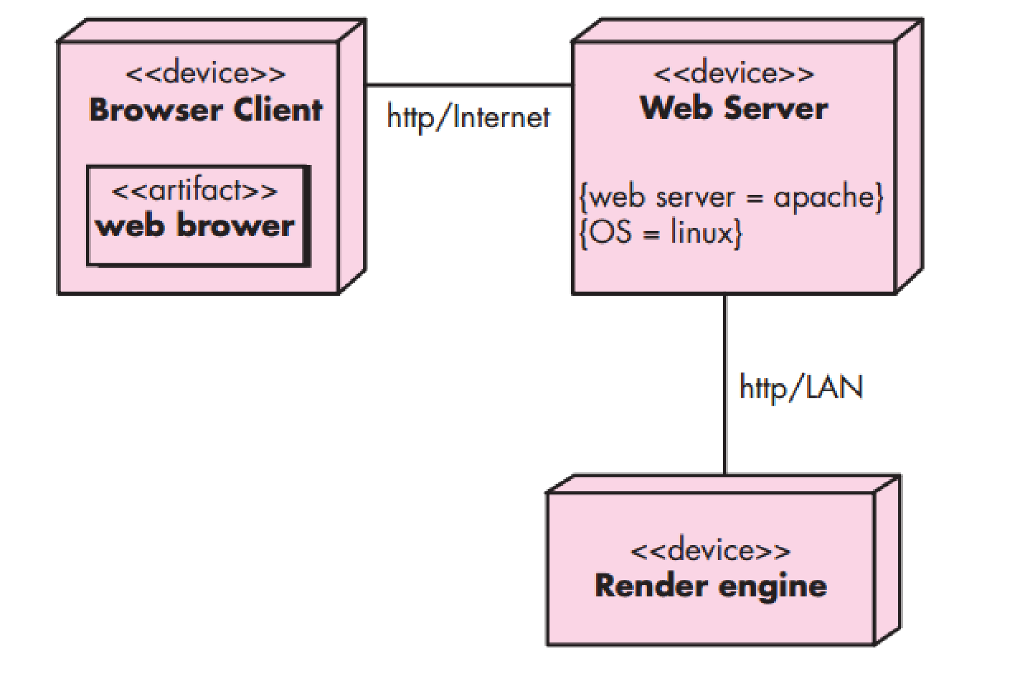
In the context of a deployment diagram, an association represents a communication path between nodes. The following diagram shows a deployment diagram for a network, depicting network protocols as stereotypes, and multiplicities at the association ends.

## Node as Container

A node can contain other elements, such as components or artifacts. The following diagram shows a deployment diagram for part of an embedded system, depicting an executable artifact as being contained by the motherboard node.



# Deployment Diagram (cont.)



- You can specify what exactly is running on that device(artifact).
- What's the environment of that device.

Clients use browser to request webserver to render something. After receiving the request from client, Web server will request the render engine to do the real job, get result and return to client.

- Development of an application collecting students' opinions about courses
- A student can
  - Read
  - Insert
  - Update
  - Make data permanent about the courses in its schedule using schedule component.
- The manager manages the statistics of the students opininion.
- A professor can only see statistic elaboration of the data .
- The professor application must be installed in pc client (ps1)
- The student application must be installed in pc client (sw1)
- The manager application must be installed in pc client (mg1)
- There is one or more server with database for courses management