15.
Modeling Your Deployed System: Deployment Diagrams

Shaoning Zeng, http://zsn.cc

15. Modeling Your Deployed System: Deployment Diagrams

- 15.1. Deploying a Simple System 部署简单系统
- 15.2. Deployed Software: Artifacts 部署软件：制品
- 15.3. What Is a Node? 节点
- 15.4. Hardware and Execution Environment Nodes
- 15.5. Communication Between Nodes
- 15.6. Deployment Specifications
- 15.7. When to Use a Deployment Diagram

15. Modeling Your Deployed System: Deployment Diagrams

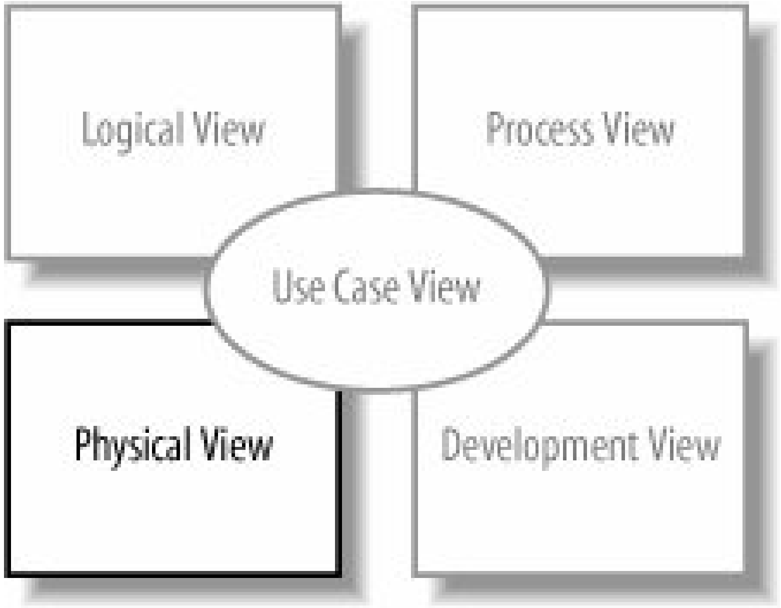


Figure 15-1. Deployment diagrams focus on the Physical View of your system

15.1. Deploying a Simple System

- To show computer **hardware**, you use a node, as shown in Figure 15-2. 使用节点表示硬件
  - This system contains a single piece of hardware: a Desktop PC.
- It's labeled with the stereotype **<<device>>** to specify that this is a hardware node.

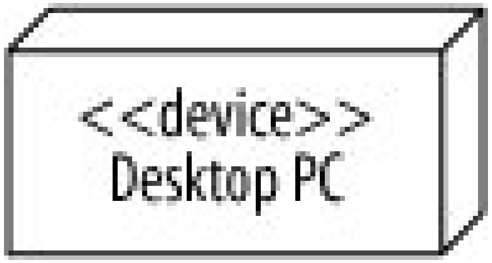


Figure 15-2. Use nodes to represent hardware in your system

15.1. Deploying a Simple System

- Modeling Software 软件 – 制品

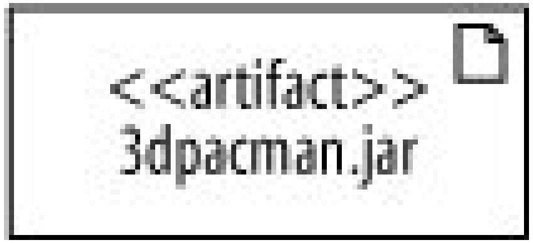


Figure 15-3.A physical software file such as a jar file is modeled with an artifact

15.1. Deploying a Simple System

- Draw the artifact inside the node to show that a software artifact is deployed to a hardware node.
  将软件制品置于节点之中

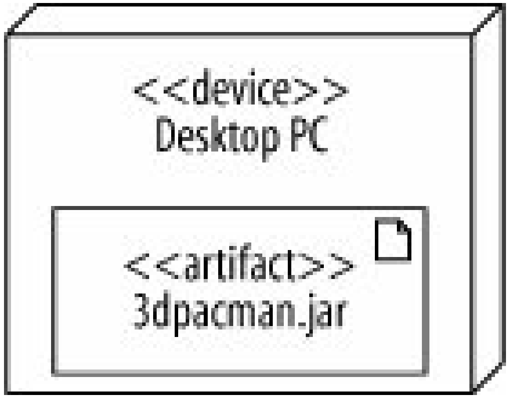


Figure 15-4. Drawing an artifact inside a node shows that the artifact is deployed to the node

15.2. Deployed Software: Artifacts

- ***Artifacts*** are physical files that execute or are used by your software. 制品是软件使用或执行的物理文件
- Common artifacts you'll encounter include:
  - Executable files, such as .exe or .jar files 可执行文件
  - Library files, such as .dlls (or support .jar files) 库文件
  - Source files, such as .java or .cpp files 源代码
  - Configuration files that are used by your software at runtime, commonly in formats such as .xml, .properties, or .txt 配置文件

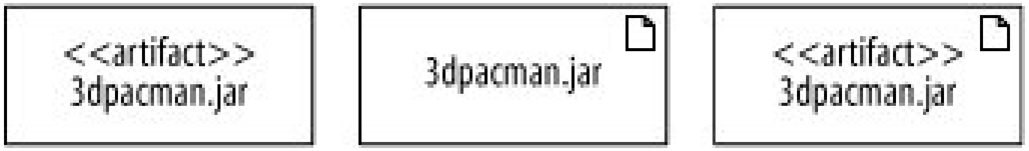


Figure 15-5. Equivalent representations of a 3dpacman.jar artifact

15.2.1. Deploying an Artifact to a Node

- An artifact is ***deployed*** to a node, which means that the artifact resides on (or is installed on) the node.
  制品部署于节点之中

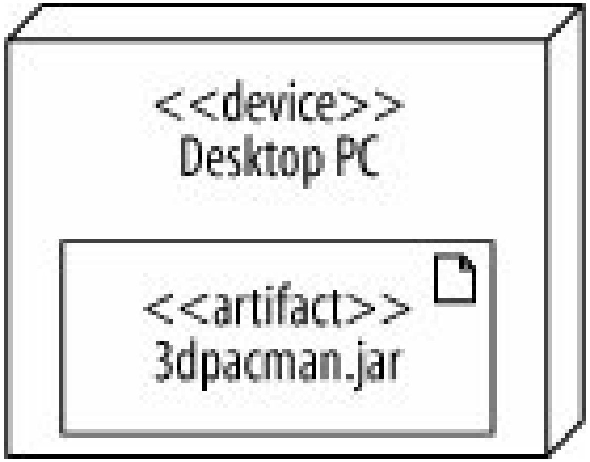


Figure 15-6.The 3dpacman.jar artifact deployed to a Desktop PC node

15.2.1. Deploying an Artifact to a Node

- You can also draw a dependency arrow from the artifact to the target node with the stereotype **<<deploy>>**, as shown in Figure 15-7. 通过依赖箭头描述

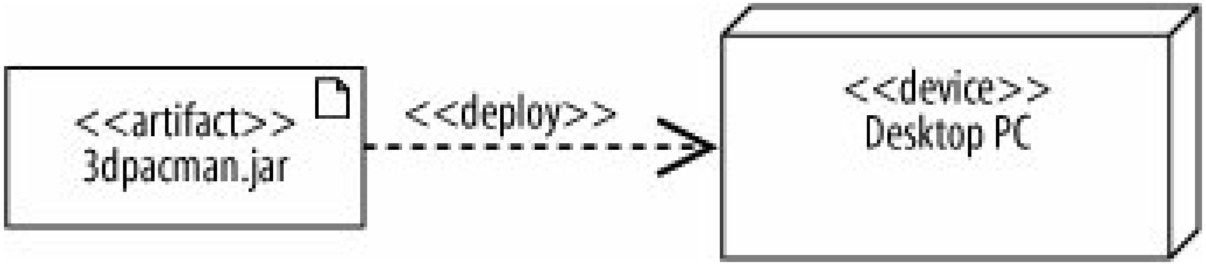
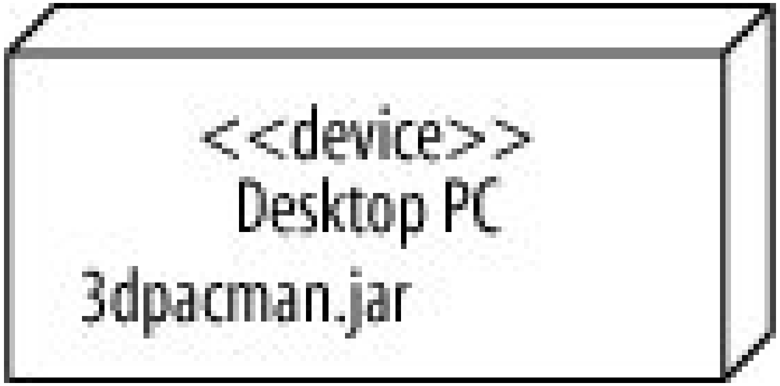


Figure 15-7.An alternate way to model the relationship deployment

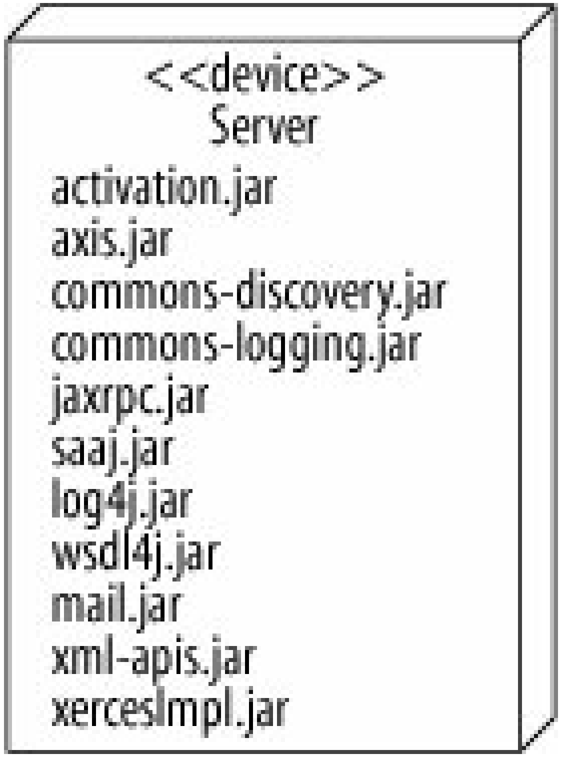
Figure 15-8. A compact way to show deployment is to write the name of the artifact inside the node 紧凑方式



15.2.2. Tying Software to Artifacts

- When designing software, you break it up into cohesive groups of functionality, such as components or packages, which eventually get compiled into one or more files or artifacts. 将软件绑定到制品（组件或包）
- In UML-speak, if an artifact is the physical actualization of a component, then the artifact manifests that component.
- An artifact can manifest not just components but any packageable element, such as packages and classes.

Figure 15-9. Listing artifact names inside a node saves a lot of space compared to drawing an artifact symbol for each artifact 在节点中列举所有制品（文字）



<<device>>
Server

activation.jar
axis.jar
commons-discovery.jar
commons-logging.jar
jaxrpc.jar
saaj.jar
log4j.jar
wsdl4j.jar
mail.jar
xml-apis.jar
xercesImpl.jar

Figure 15-10. A deployment notation that uses artifact symbols (instead of listing artifact names) allows you to show artifact dependencies
使用制品符号表示部署内容

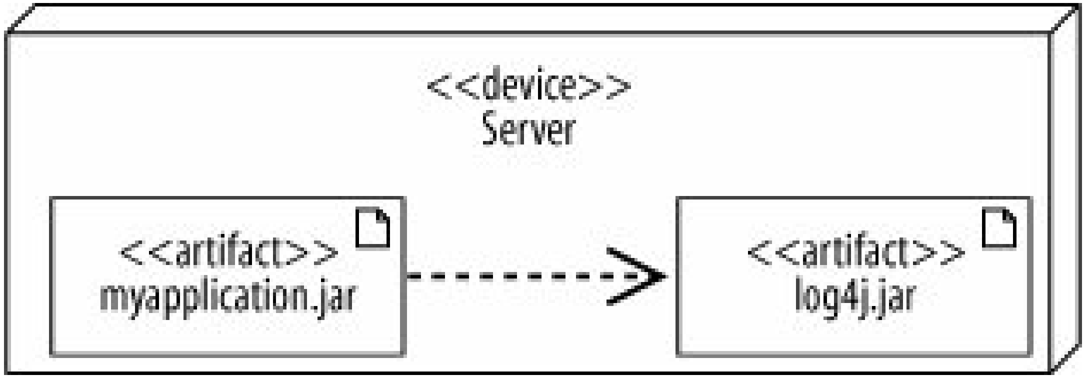
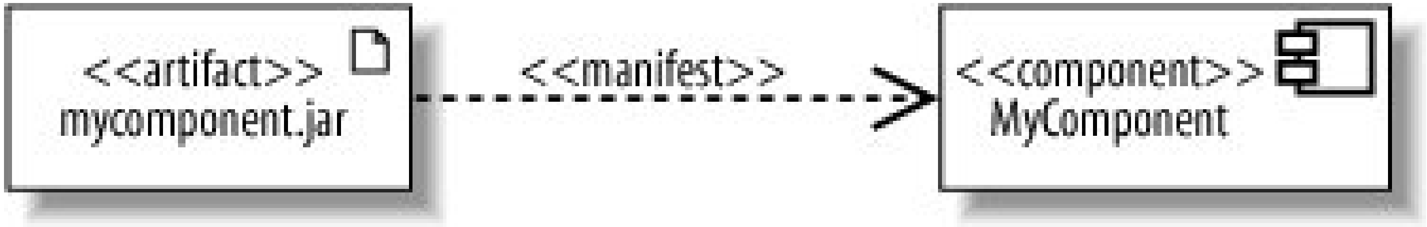


Figure 15-11. The artifact mycomponent.jar manifests the component MyComponent

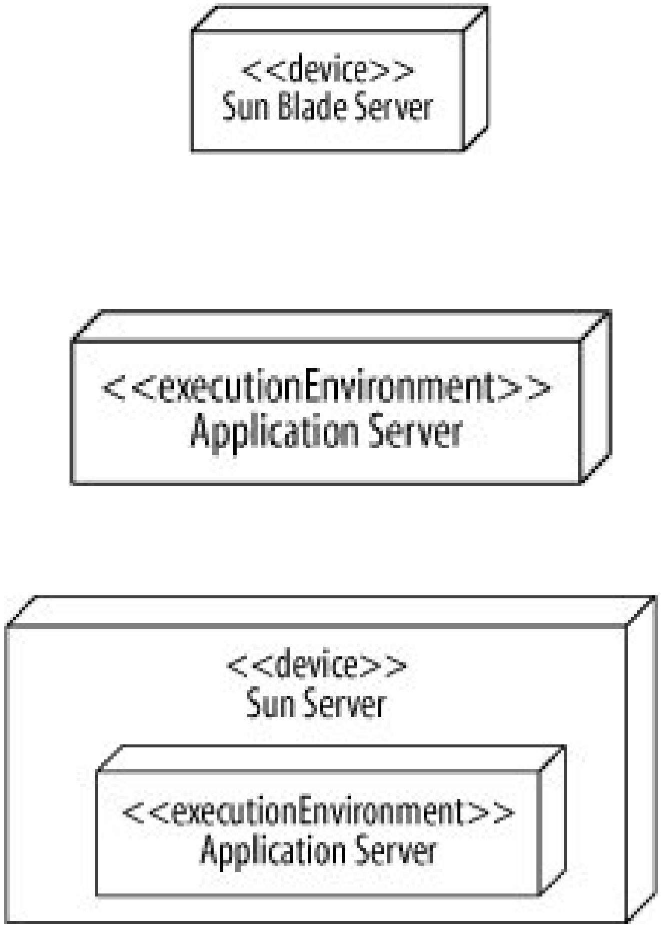
制品与组件的对应关系



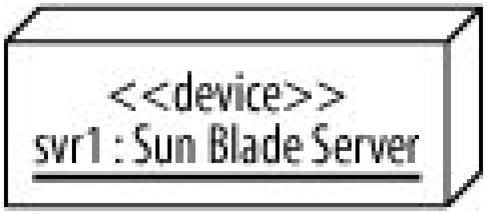
15.3. What Is a Node?

- A ***node*** is a hardware or software resource that can host software or related files. 节点是指硬件或软件
  - You can think of a software node as an application context; generally not part of the software you developed, but a third-party environment that provides services to your software.
- The hardware nodes: 硬件节点
  - Server
  - Desktop PC
  - Disk drives
- The execution environment nodes: 可执行环境节点
  - Operating system
  - J2EE container
  - Web server
  - Application server

15.4. Hardware and Execution Environment Nodes



15.4.1. Showing Node Instances



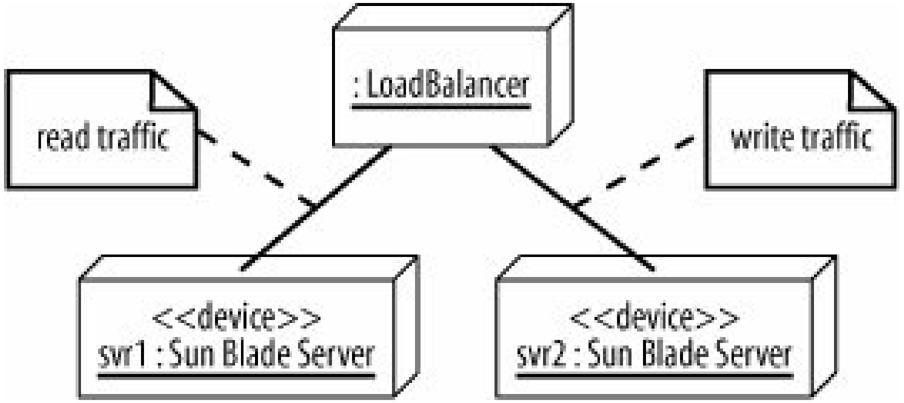


Figure 15-16. One node gets read traffic and the other gets write traffic

15.5. Communication Between Nodes

- Communication paths are used to show that nodes communicate with each other at runtime. 节点通信
- A communication path is drawn as a solid line connecting two nodes. 通信路径
- The type of communication is shown by adding a stereotype to the path. 通信类型

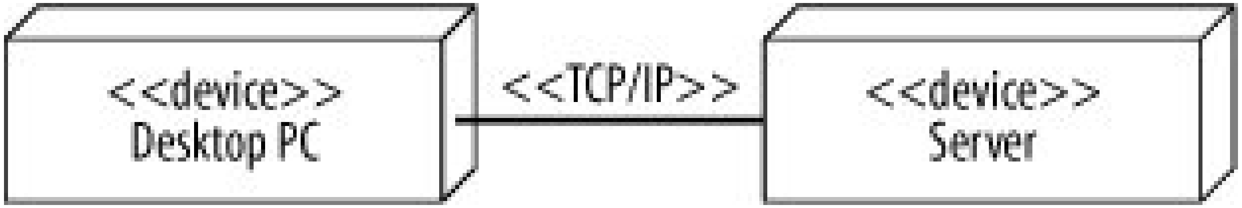
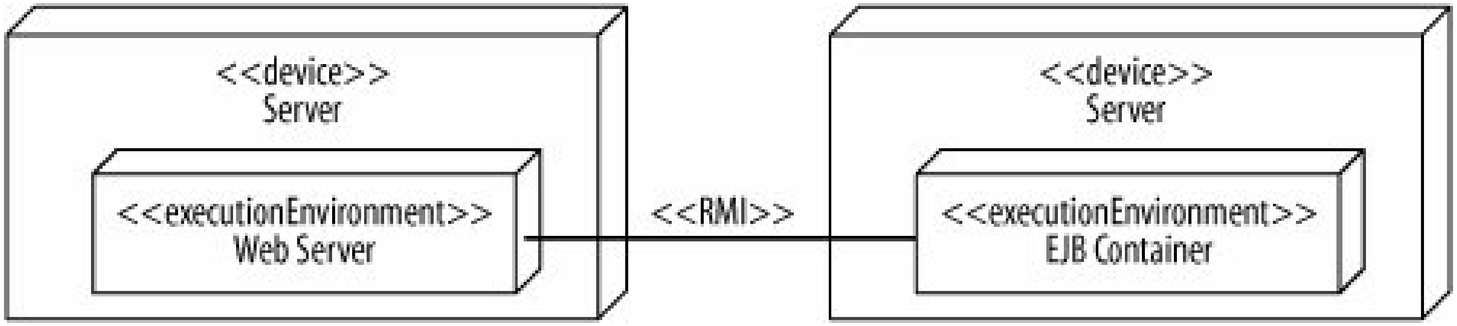


Figure 15-17.A Desktop PC and Server communicate via TCP/IP

Figure 15-18. You can also show communication paths between execution environment nodes
执行环境节点之间的通信



15.6. Deployment Specifications 部署规范

- Installing software is rarely as easy as dropping a file on a machine; often you have to specify configuration parameters before your software can execute. 安装软件
- A deployment specification is a special artifact specifying how another artifact is deployed to a node. It provides information that allows another artifact to run successfully in its environment. 部署规范描述如何将制品部署到节点上
- Deployment specifications are drawn as a rectangle with the stereotype **<<deployment spec>>**.
- There are two ways to tie a deployment specification to the deployment it describes: 绑定部署规范到所描述的部署
  - Draw a dependency arrow from the deployment specification to the artifact, nesting both of these in the target node.
  - Attach the deployment specification to the deployment arrow, as shown in Figure 15-19.

Figure 15-19. Equivalent ways of tying a deployment specification to the deployment it describes
将部署规范绑定到部署的两种方法

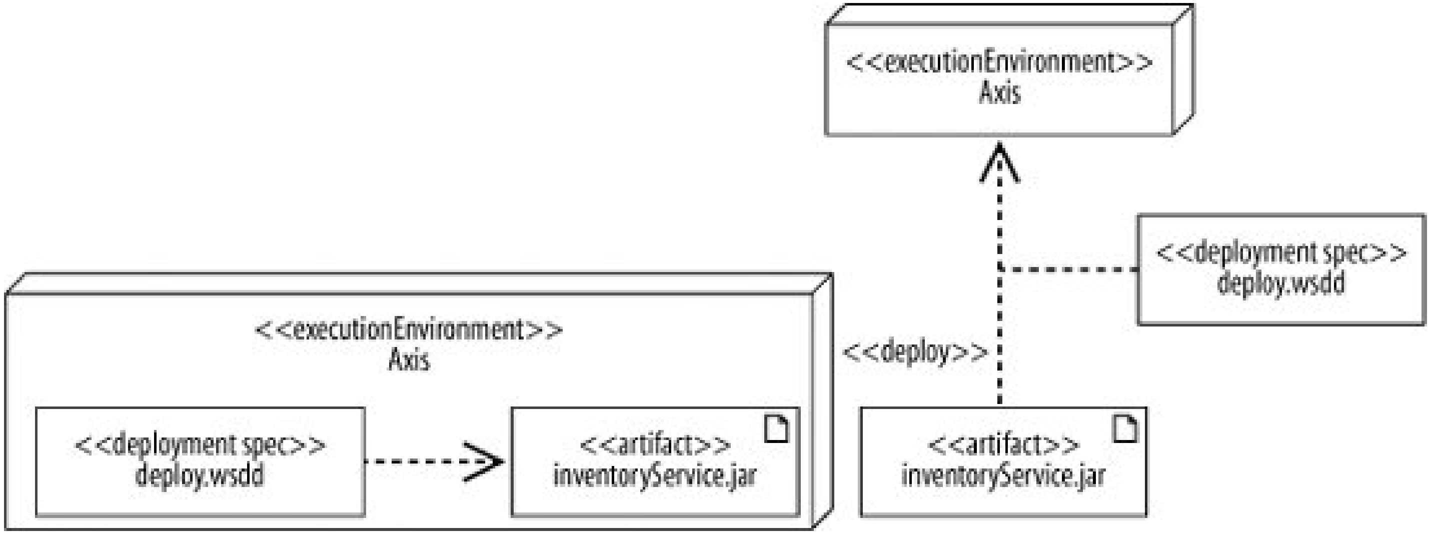
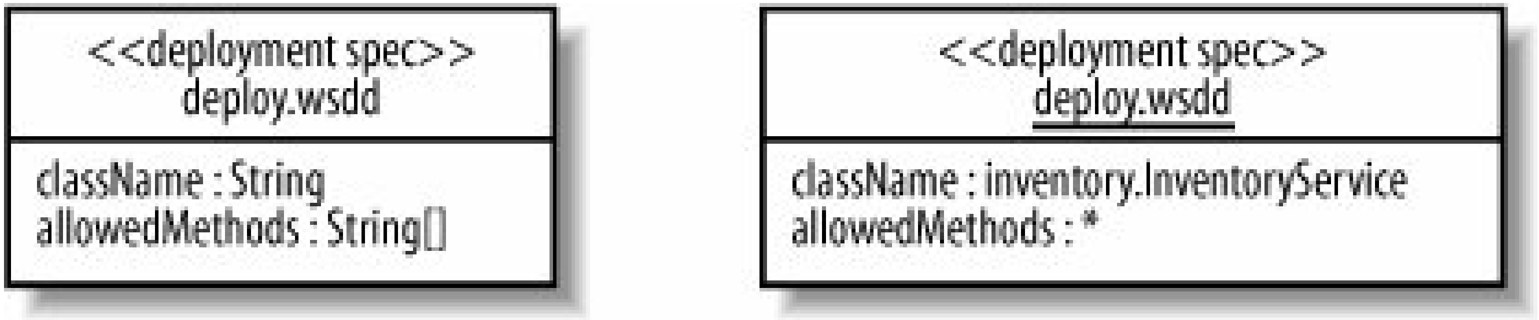


Figure 15-20. Showing the properties of a deployment specification: the notation on the right shows an instance populated with values
显示部署规范的属性（右图是实例）



<<deployment spec>>
deploy.wsdd

className : String
allowedMethods : String[]

<<deployment spec>>
deploy.wsdd

className : inventory.InventoryService
allowedMethods : *

15.7. When to Use a Deployment Diagram

- Deployment diagrams are useful at all stages of the design process. 部署图适用于所有设计阶段
  - When you begin designing a system, you probably know only basic information about the physical layout. 物理布局
  - For example, if you're building a web application, you may not have decided which hardware to use and probably don't know what your software artifacts are called.
- But you want to communicate important characteristics of your system, such as the following: 描述系统重要特性
  - Your architecture includes a web server, application server, and database. 服务器架构
  - Clients can access your application through a browser or through a richer GUI interface. 客户端访问方式
  - The web server is protected with a firewall. 防火墙配置

Figure 15-21. A rough sketch of your web application

Web应用程序草图

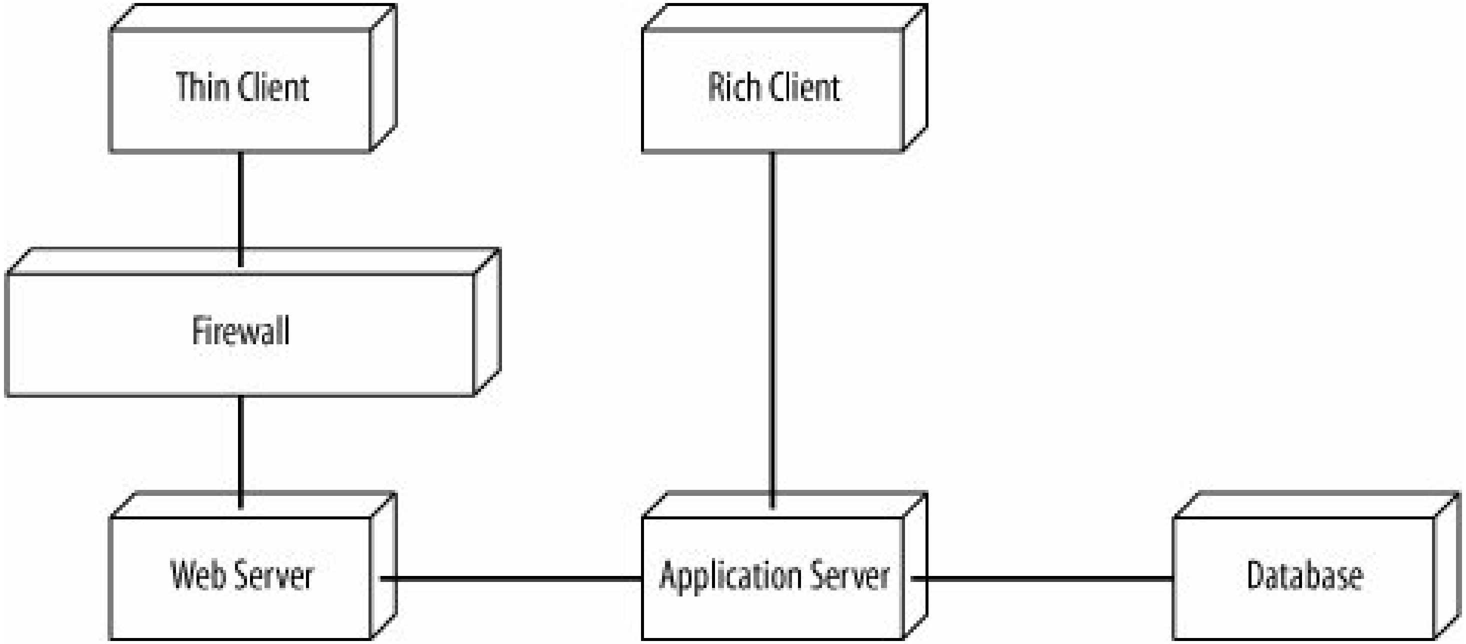
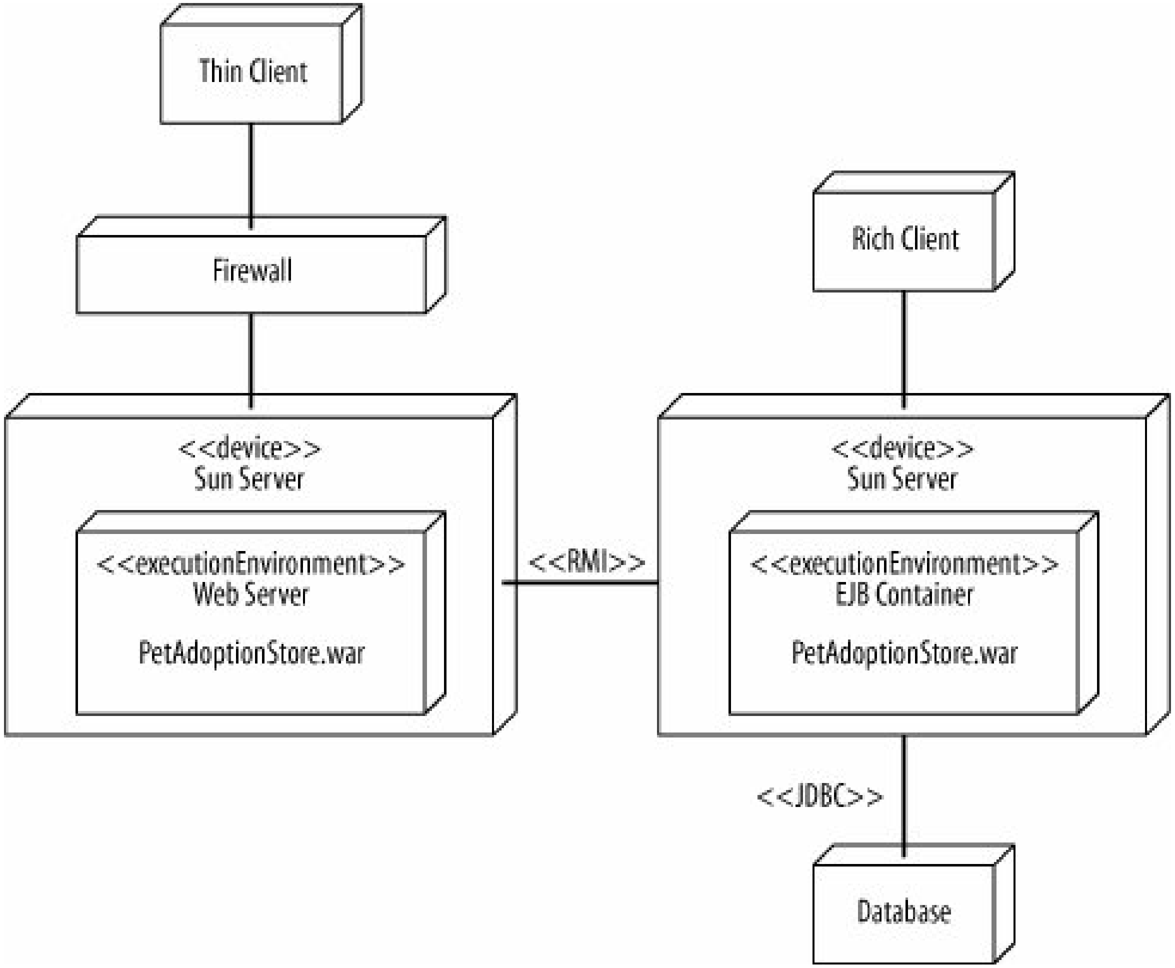


Figure 15-22. You can provide any amount of detail about the physical design of your system 关于系统物理设计的详细说明



Summary

- 15. Modeling Your Deployed System: Deployment Diagrams
  - 15.1. Deploying a Simple System
  - 15.2. Deployed Software: Artifacts
  - 15.3. What Is a Node?
  - 15.4. Hardware and Execution Environment Nodes
  - 15.5. Communication Between Nodes
  - 15.6. Deployment Specifications
  - 15.7. When to Use a Deployment Diagram

See you …