

第3章 构建WEB应用技术

第3章 构建WEB应用技术

■ 章节目标

- 1.掌握客户/服务器模型，HTTP协议,URL的格式，以及COOKIE和SESSION的运用。
- 2.了解WEB技术的发展过程。
- 3.了解WEB应用模型，掌握三层和四层架构模型。
- 4. 掌握HTML和XML的特点，能够使用HTML和XML做网页。
- 5.了解APPLET和脚本技术。掌握APPLET和脚本技术的优缺点。

第3章 构建WEB应用技术

■ 章节目标（续）

- 6. 了解CGI的工作原理和API的特点。掌握Servlet的生命周期和处理事件的四种方法。
- 7. 能够写简单的JSP页面，了解其特点。
- 8. 了解面向对象的概念和JAVA的特点以及常用的JAVA技术。
- 9. 了解MVC模式的概念和应用范围。

第3章 构建WEB应用技术

■ 章节目标（续）

- 10.掌握Web Service的体系架构和SOAP，UDDI，WSDL的概念和基本功能。
- 11. 掌握J2EE中各容器的功能，了解其构件技术、服务技术和通信技术。
- 12.了解EDI的特点以及它与INTERNET的结合方式。
- 13.了解无线技术。

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.1 WWW简介

- 3.1.1 WWW的概念
- 3.1.2 客户机/服务器模型
- 3.1.3 URL
- 3.1.4 HTTP协议
- 3.1.5 Cookie和Session

3.1.1 WWW的概念

- WWW（World Wide Web）简称W3，有时也叫WEB，中文译名为万维网、环球信息网等。
- WWW使得全球范围的用户可以利用Internet进行方便地通信、信息交流和信息查询。
- WWW采用客户机/服务器工作方式，以超文本信息的组织与传递为内容。
- 服务器端运行WWW服务器程序。用户通过WWW 客户程序向WWW服务器发出查询请求，WWW服务器则检索所用储存在服务器内的信息。
- WWW的客户程序也就是称为浏览器。

3.1.2 客户机/服务器模型

- WWW主要由三部分组成，服务器、客户机以及通信协议。
- 服务器端主要包括服务器软件、服务器的扩展应用软件、服务器端入口软件等。
- 通信协议中最常见的是HTTP协议。
- 客户端包括浏览器、用户工具等。网景公司开发的Netscape和微软的Internet Explorer是最著名的浏览器。
- WWW浏览器负责与WWW服务器建立连接，发送客户访问请求，处理HTML超媒体，提供客户界面。

3.1.3 URL

- URL（ Uniform Resource Locator ）称之为统一资源定位器，用来指示某一项信息(资源)的所在位置及存取方法。
- URL的格式：
使用协议://主机名称/文件路径/文件名： 端口号。
端口号缺省为80。
- 例如：
`http://www-900.ibm.com/developerWorks/cn/linux/1-cpconf/index.shtml`

3.1.4 HTTP协议

- HTTP: Hyper Text Transport Protocol超文本传输协议
- HTTP是用于在互联网上传输超文本文件的协议。
- 它是TCP/IP协议的应用层协议之一，是为Web定制的核心协议。
- 它是基于文本的简单协议，基于请求-应答的服务器/客户端工作模式，能够理解任意类型的对象

3.1.4.1 HTTP协议的工作过程

- 第一步：客户请求连接。
- 第二步：客户提出请求。
- 第三步：服务器应答。
- 第四步：断开连接。

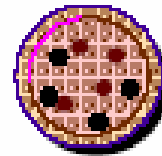
3.1.4.2 HTTP协议的特点

- HTTP协议具有无连接，无状态的特点。
- 无连接的含义是限制每次连接只处理一个请求。客户和服务器连接后提交一个请求，在客户接到应答后立即断开连接。
- 无状态是指协议对于事务处理没有记忆能力。

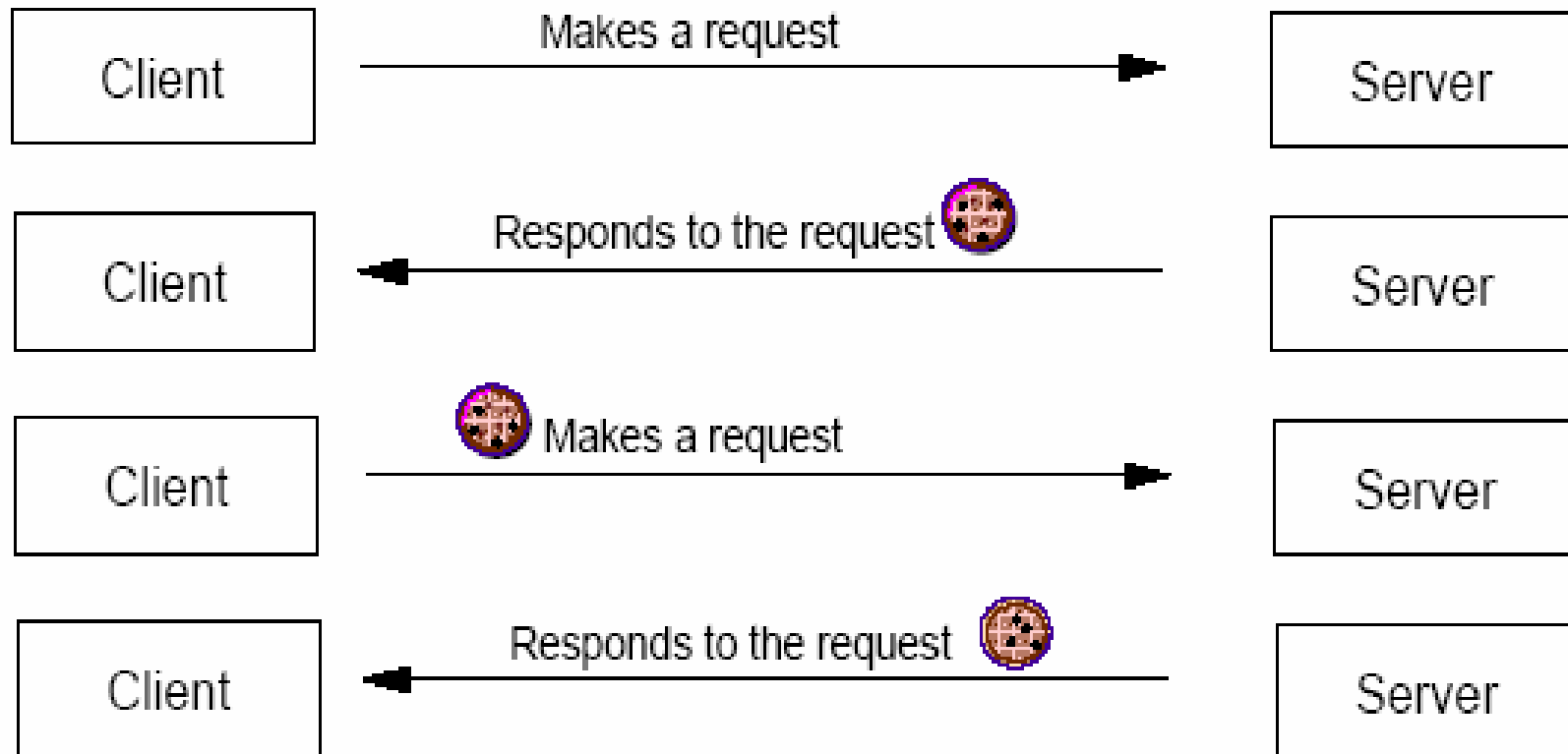
3.1.5 Cookie和Session

- 3.1.5.1 Cookie 技术
- 3.1.5.2 Session 技术

3.1.5.1 Cookie 技术



Cookie: A token word or words used to "remember" a larger set of information or state



3.1.5.1 Cookie 技术（一）

- Cookie是由服务器在用户的计算机上创建的一个文件。
- Cookie由客户端保存
- 每次客户端访问服务器时，将服务器在第一次连接时送到客户端的cookie传送给服务器端，对cookie信息校验后，合法则会响应该次请求,不合法则反馈相关错误信息

3.1.5.1 Cookie 技术（二）

- 在线购买应用可以使用Cookie存储当前选择的商品信息
- 网站可以使用Cookie在每个客户端存储该用户的个人喜好，并且使得用户能够在每次连接到该网站时就通过cookie将他们的个人喜好传递到网站服务器。

3.1.5.2 Session 技术（一）

- Session机制是一种服务器端的机制。
- 使用Session技术的本质，就是把要传递的参数保存在两个页面都能共享的对象Session中，前一个页面在这个对象中写入状态、后一个页面从这个对象中读取状态。
- Session对象相对于Cookie而言，最大的区别就在于Cookie是存储在客户端的，而Session是存储在服务器端的。
- 每个客户端在服务端都对应了一个session id，服务端维持了由session id标识的一系列session对象。

3.1.5.2 Session 技术（二）

- Session的这种特有的功能在电子商务解决方案中往往是非常有用的。
- 一个用户需要开发一个在线商店，它的长远目标是通过跟踪客户购买行为以实现网站的个性化等商业智能。
- 由于在后期的商业智能实现过程中需要对收集的客户购买行为的部分特定数据进行分析。
- 在这种解决方案需求中，使用**cookie**是不合适的，需要使用**session**在服务器端存储用户的购买过程信息，以供后期的商业智能分析过程和系统应用重写后的测试过程中使用。

本章目录

- 3.1 WWW简介
- **3.2 WEB技术的发展**
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.2 WEB技术的发展

- 3.2.1 静态WEB技术
- 3.2.2 动态WEB技术
- 3.2.3 事务WEB技术

3.2.1 静态WEB技术

- 静态WEB技术主要用于静态WEB页面(由文字和静态图像组成)的浏览。
- 静态WEB技术通常被人们称为第一代的WEB应用。
- 静态WEB应用技术的不足之处在于发布的信息是静态的、不变的。

3.2.2 动态WEB技术

- 动态WEB技术的主要内容：将WEB技术与Client/Server系统中的后台数据库连接起来，使用户能在浏览器这个统一界面下既能浏览网上信息又能访问数据库。

3.2.3 事务WEB技术

- **WEB技术必须解决的问题：**如何实现实时的事务处理，并具有可扩展性、可伸缩性、安全性及客户认证能力，以及与现有的Client/Server系统的集成等。
- **事务WEB技术**允许数据库根据用户的查询要求产生动态的内容，即可动态的将变化的内容，以页面的形式提供给用户，如报社的新闻站点。

本章目录

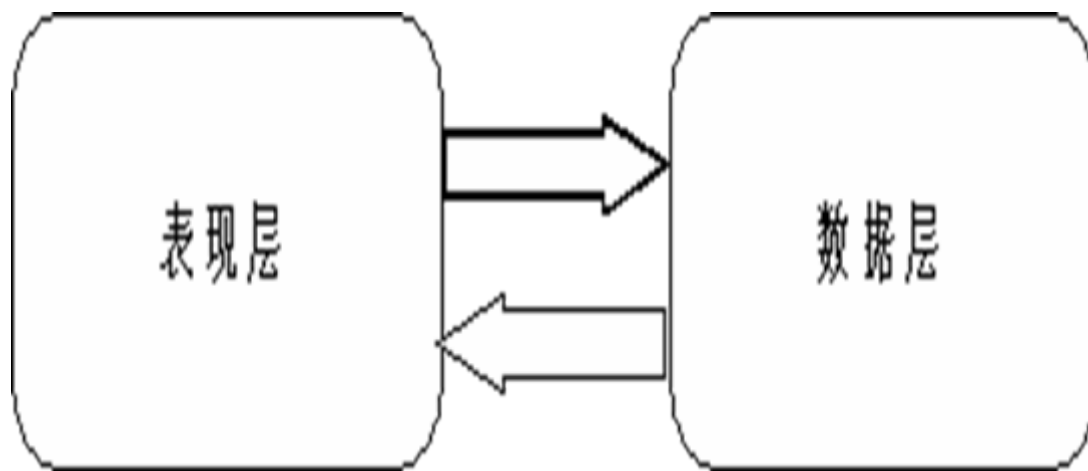
- 3.1 WWW简介
- 3.2 WEB技术的发展
- **3.3 WEB应用模型**
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.3 WEB应用模型

- 3.3.1 两层架构WEB应用模型
- 3.3.2 三层架构WEB应用模型
- 3.3.3 四层架构WEB应用模型

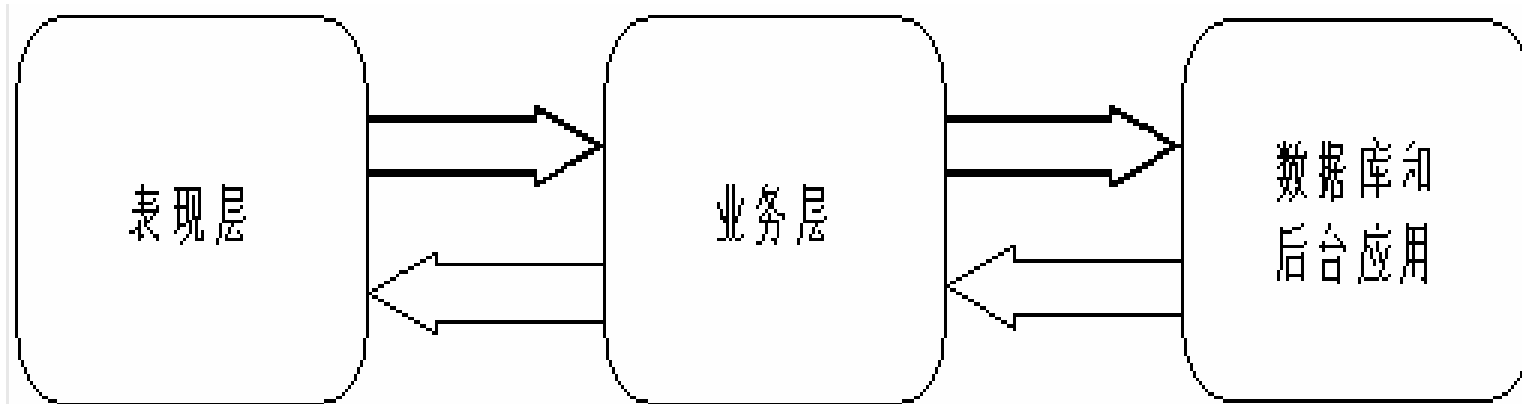
3.3.1 两层架构WEB应用模型

- 两层架构系统适用于简单的WEB应用，系统的设计实施简单快速，开发成本低，但是可扩展性和可维护性较差。



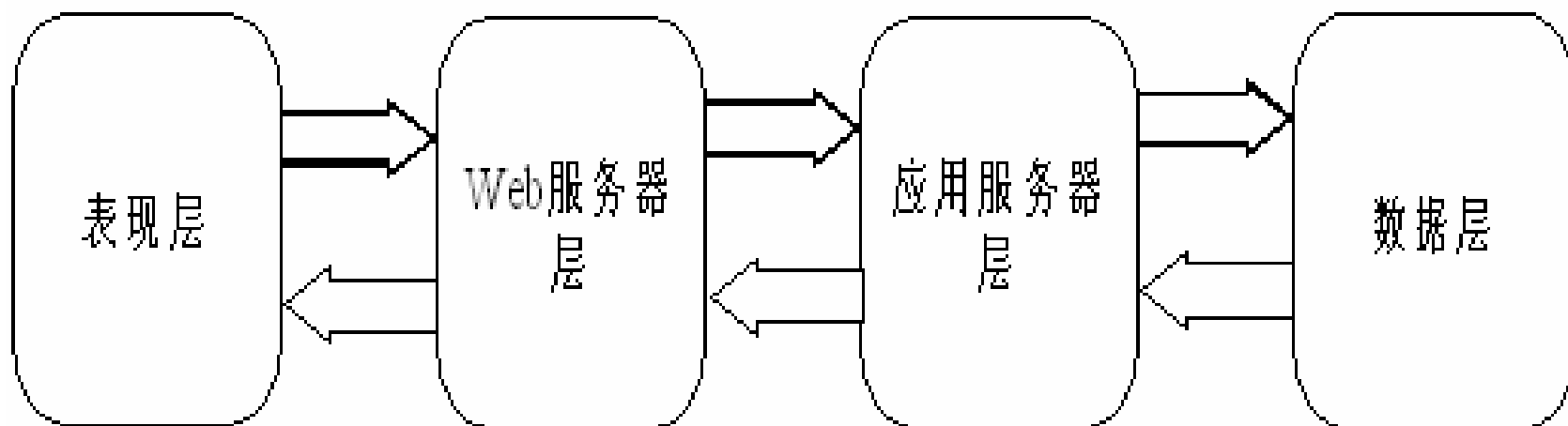
3.3.2 三层架构WEB应用模型

- 三层架构将业务逻辑从数据管理和客户端中分离出来。
- 从开发角度来看，三层架构适合团体协作开发，不同的人可以有不同的分工。
- 从应用角度来看，系统具有的好的可扩展性和可维护性，同时也增加了数据库和后台应用系统的安全性。



3.3.3 四层架构WEB应用模型

- 四层架构具有三层架构的全部优点。
- 系统可以实现负载均衡和分布处理应用。
- 它比三层架构具有更好的响应速度。
- 它比三层架构具有更好的可扩展性和可维护性。
- 它还能够预防单点失败，并具有更好的安全性。



本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- **3.4 客户端技术**
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.4 客户端技术

- 3.4.1 HTML
- 3.4.2 脚本语言JavaScript
- 3.4.3 Java Applets
- 3.4.4 XML

3.4.1 HTML

- 3.4.1.1 HTML是什么
- 3.4.1.2 HTML文件大致结构
- 3.4.1.3 HTML标记
- 3.4.1.4 HTML的特点和功能

3.4.1.1 HTML是什么

- HTML是Hypertext Markup Language（超文本标记语言）的缩写
- HTML是构成WEB页面的主要工具，是用来表示网上信息的符号标记语言。
- SGML：Standard Generalized Markup Language
- HTML是SGML的一个简化的实现

3.4.1.2 HTML文件大致结构

<HTML>

<HEAD>

<TITLE>网页标题</TITLE>

</HEAD>

<BODY>

网页的内容

</BODY>

</HTML>

3.4.1.3 HTML标记（一）

标记写法需要遵循以下规则：

- 任何标记皆由"<"及">"所围住，如 **<P>**。
- 标记名与小于号之间不能留有空白字符。
- 某些标记要加上参数，某些则不必。如
Hello
- 参数只可加于起始标记中。
- 在起始标记之标记名前加上符号"/"便是其终结标记，如 ****。
- 标记字母大小写皆可。

3.4.1.3 HTML标记（二）

- 文件标记：<HTML>，<HEAD>，<TITLE>，<BODY>
- 排版标记：<P>，
，<CENTER> 等
- 字体标记：<H1>，<H2>，<H3>，<H4>等
- 表格标记：<TABLE>，<TR>，<TD> 等
- 表单标记：<FORM>，<INPUT> 等
- 连结标记：<A>
- 框架标记：<FRAMESET>，<FRAME> 等

3.4.1.4 HTML的特点和功能

特点：

- 简单易学；
- 解释执行，无需编译；
- 独立于平台；

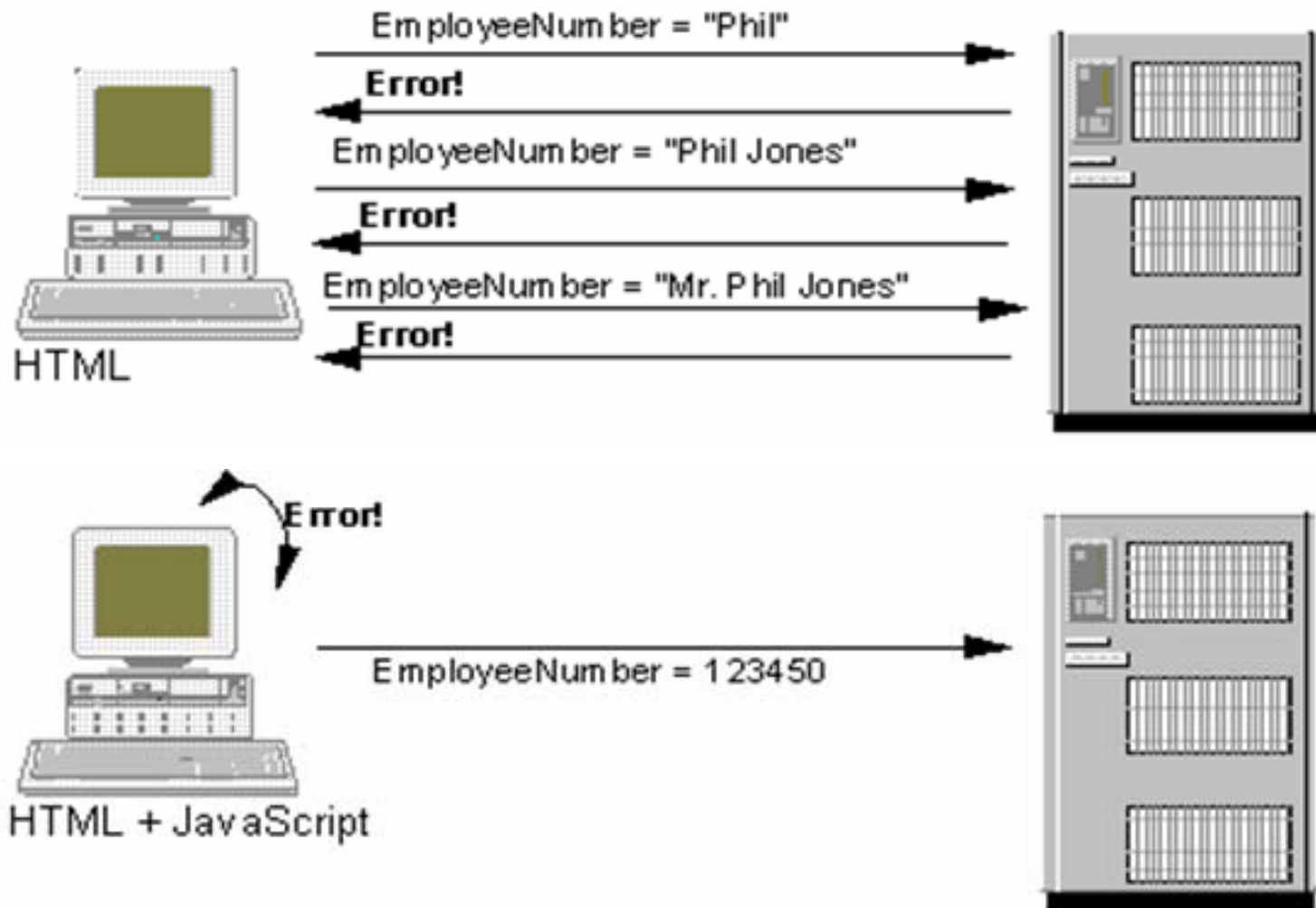
3.4.2 脚本语言JavaScript

- 3.4.2.1 JavaScript简介
- 3.4.2.2 JavaScript功能举例
- 3.4.2.3 JavaScript语言特点

3.4.2.1 JavaScript简介

- 由 Netscape公司开发
- 基于对象和事件驱动的编程语言
- 是一种解释执行的编程语言

3.4.2.2 JavaScript功能举例



3.4.2.2 JavaScript功能举例

- 通过上面的例子可以看出，使用JavaScript处理事务逻辑，可以有效地降低网络的通信量，减少服务器端处理负荷，改善服务器性能。

3.4.2.3 JavaScript语言特点

- 一种脚本编写语言
- 基于对象的语言
- 简单性
- 安全性
- 动态性
- 跨平台性

3.4.3 Java Applets

- 3.4.3.1 Applet实例
- 3.4.3.2 Applet的优缺点

3.4.3.1 Applet实例

```
<html>
<head>
<title>Calculator</title>
</head>
<body>
<applet id="Calculator" width="300" height="500"
code ="Calculator.class" codebase=".">
<param name="InitialMode" value="Normal">
</applet>
</body>
</html>
```

3.4.3.2 Applet的优缺点

- Applet可以在多种浏览器和平台上工作。
 - Applet下载起来比较快。
 - Applet具有好的安全性。
 - Applet在某些情况下能够与网页服务器通信，进行信息交互。
-
- 最大的缺点就是对浏览器有要求。它要求浏览器能够提供**Java**虚拟机环境。

3.4.4 XML

- 3.4.4.1 XML简介
- 3.4.4.2 DTD实例
- 3.4.4.3 XML实例
- 3.4.4.4 XML的特点
- 3.4.4.5 XML的使用前景

3.4.4.1 XML简介（一）

- XML: Extensible Markup Language可扩展标记语言
- 它是SGML（Standard Generalized Markup Language）的一个优化子集
- 1998年，W3C正式发布了XML 1.0标准。
- XML是一种元标记语言，使用者可按需创建新的标记
- 标记通过XML DTD（Document Type Definition，文档类型定义）来加以定义

3.4.4.1 XML简介（二）

- DTD描述了XML文件的文档结构。
- 它含有一系列关于元素类型（Element Type），属性（Attributes），实体（Entities）和符号（Notations）的定义。
- 它定义了文档所需的标记，可在文档里使用的元素类型，这些元素之间的联系，元素的属性。
- XML文档可以在它的文档类型声明(Document Type Declaration)里声明该文档遵循某个XML DTD。

3.4.4.2 DTD实例

messenger.dtd文件:

- `<!ELEMENT messenger (service*)>`
- `<!ELEMENT service (contactlist*)>`
- `<!ATTLIST service name CDATA #REQUIRED>`
- `<!ELEMENT contactlist (contact*)>`
- `<!ELEMENT contact (#PCDATA)>`

3.4.4.3 XML实例

messenger.xml文件:

- `<?xml version="1.0" ?>`
- `<!DOCTYPE messenger SYSTEM "../messenger.dtd">`
- `<messenger>`
- `<service name=".NET Messenger Service">`
- `<contactlist>`
- `<contact> rickzhang @sina.com</contact>`
- `</contactlist>`
- `</service>`
- `</messenger>`

3.4.4.4 XML的特点

- 可扩展性
- 灵活性
- 自描述性
- 简明性

3.4.4.5 XML的使用前景

- 商务的自动化处理
- 信息发布
- 智能化的Web应用程序和数据集成

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- **3.5 服务器端技术**
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.5 服务器端技术

- 3.5.1 CGI
- 3.5.2 API
- 3.5.3 Servlet
- 3.5.4 JSP

3.5.1 CGI

- 3.5.1.1 CGI简介
- 3.5.1.2 CGI工作原理
- 3.5.1.3 CGI的优缺点

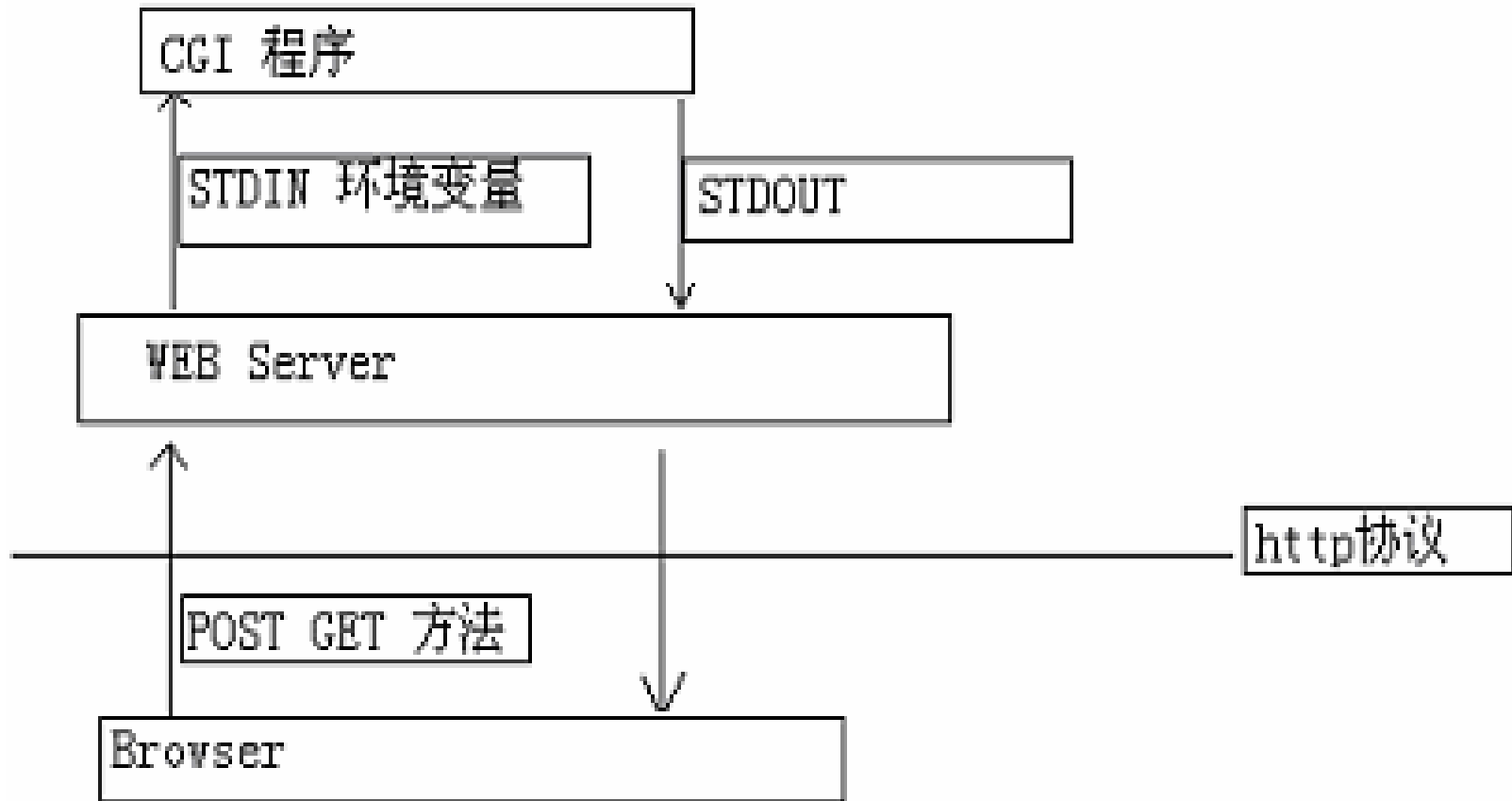
3.5.1.1 CGI简介（一）

- CGI (Common Gateway Interface) : 通用网关接口
- 通过CGI，Web服务器能够执行应用程序，并将应用程序的输出，如文字、图形、声音等传递给一个Web浏览器
- CGI程序语言:C、Shell、Perl和Visual Basic 等

3.5.1.1 CGI简介（二）

- CGI主要的功能就是在服务器应用程序和客户之间传递信息。
- CGI程序两种调用方式：
 - （1）URL调用：
`http://gjy.sic.o.ml.org/cgi-win/cgi.exe`
 - （2）HTML里的Form调用

3.5.1.2 CGI工作原理



3.5.1.3 CGI的优缺点

- 优点：CGI定义的是一种编程规范，实现方便，支持多种语言，易于程序员快速掌握。
- 不足之处：大量负载时多个CGI应用程序的执行将严重影响服务器系统性能。

3.5.2 API

- API（Application Program Interface）：应用程序接口
- API解决多线程问题，而CGI无法实现多线程
- API使用动态链接库
- 两种API：Netscape公司的NSAPI 和 Microsoft的ISAPI
- 只能用C语言编写，书写规范复杂，相较于CGI程序开发时间长，难度大。

3.5.3 Servlet

- 3.5.3.1 Servlet概述
- 3.5.3.2 Servlet的软件包
- 3.5.3.3 Servlet的特点

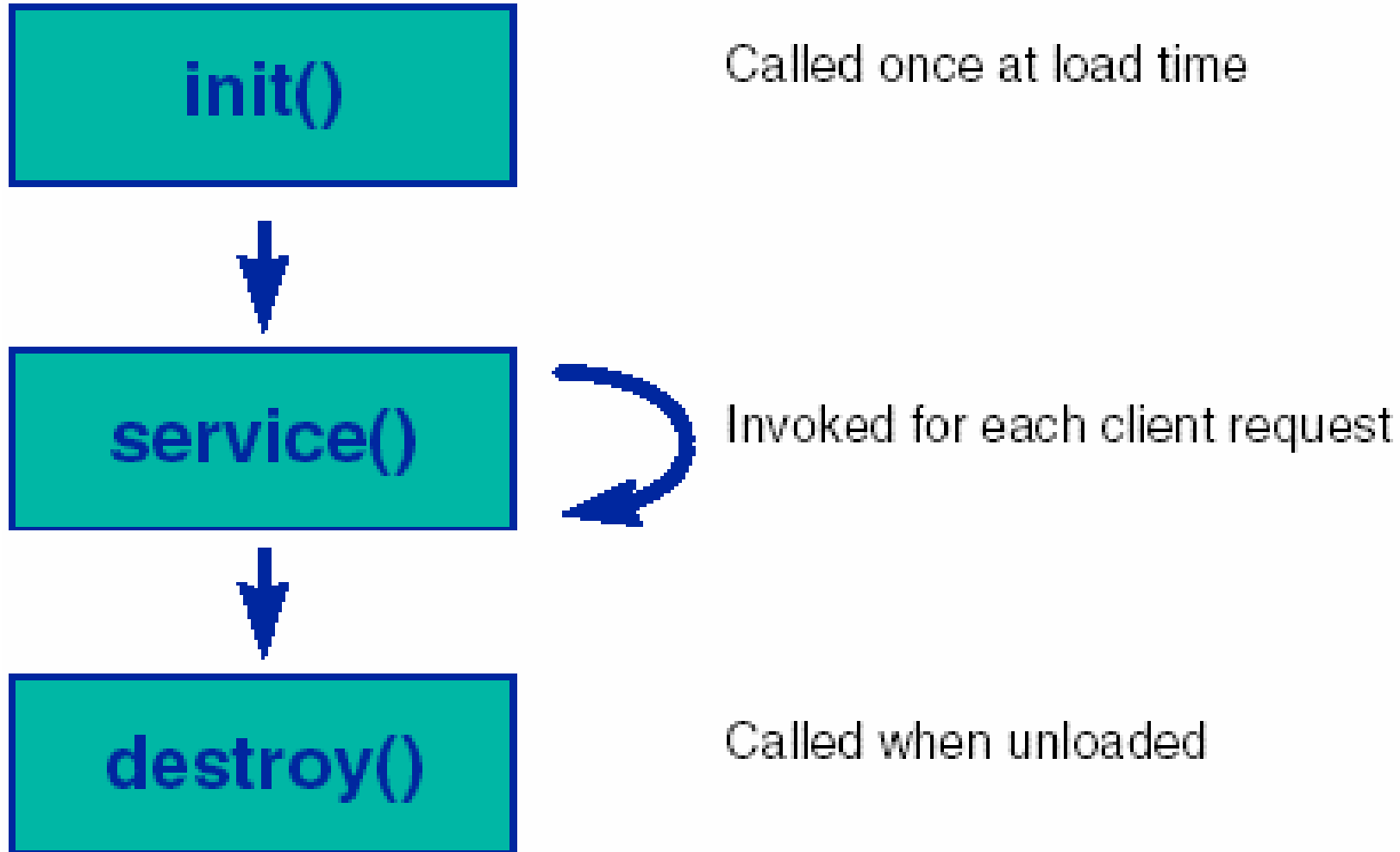
3.5.3.1 Servlet概述

- Servlet就是Java Servlet
- Servlet是一个Java类
- Servlet运行于由Servlet引擎所管理的Java虚拟机中，被来自Web客户机的请求所唤醒并用来处理请求
- Servlet的开发工具可J2SDK开发工具箱
- 主要有两个软件包：Javax.servlet包和Javax.servlet.http包

3.5.3.2 Servlet的软件包--生命周期（一）

- Servlet的生命周期：
 - （1）初始化时期：**init()**方法
 - （2）执行时期：**service()**方法
 - （3）结束时期：**destroy()**方法

3.5.3.2 Servlet的软件包--生命周期（二）



3.5.3.2 Servlet的软件包-- 四种方法

HttpServlet类中常用的四种方法:

- **doGet()**方法
- **doPost()**方法
- **doPut()**方法
- **doDelete()**方法

3.5.3.3 Servlet的特点

- 高效
- 方便
- 功能强大
- 可移植性好
- 节省投资

3.5.4 JSP

- 3.5.4.1 JSP概述
- 3.5.4.2 JSP语法
- 3.5.4.3 JSP实例
- 3.5.4.4 JSP技术特点

3.5.4.1 JSP概述

- **JSP** : Java Server Pages
- 用于创建可支持跨平台及跨Web服务器的动态网页
- 简单的说，一个**JSP**网页就是在**HTML**网页中包含了能够生成动态内容的可执行应用程序代码。
- **JSP**将应用程序逻辑和页面显示分离

3.5.4.2 JSP语法

■ JSP的基本语法：

(1) JSP指示：JSP页面指示；语言指示；包含指示；标识库指示

(2) JSP标识：核心标识为
`jsp:useBean`；`jsp:setProperty`；
`jsp:getProperty`

(3) 脚本元件：脚本在`<%和%>`标志中被描述，常用的脚本元件包括表达式和声明

3.5.4.3 JSP实例

- <HTML>
- <%@ page language=="java"
imports=="com.wombat.JSP.*" %>
- <H1>Welcome</H1>
- <P>Today is </P>
- <jsp:useBean id=="clock"
class=="calendar.jspCalendar" />
-
- Day: <%==clock.getDayOfMonth() %>
- Year: <%==clock.getYear() %>
-

3.5.4.3 JSP实例（续）

- `<% if (Calendar.getInstance().get(Calendar.AM_PM) == Calendar.AM) { %>`
- Good Morning
- `<% } else { %>`
- Good Afternoon
- `<% } %>`
- `<%@ include file=="copyright.html" %>`
- `</HTML>`

3.5.4.4 JSP技术特点

- 将内容的生成和显示进行分离
- 强调可重用的组件
- 采用标识简化页面开发

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- **3.6 面向对象的概念和java技术**
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

3.6 面向对象的概念和java技术

- 3.6.1 面向对象的基本概念
- 3.6.2 面向对象的特点
- 3.6.3 面向对象的优点
- 3.6.4 JAVA的产生和发展
- 3.6.5 Java的定义
- 3.6.6 Java的特点
- 3.6.7 常用JAVA技术

3.6.1 面向对象的基本概念

- 对象：由一个数据集以及对该数据集的操作组成
- 类：具有共同属性和行为的对象集合，是对象的抽象
- 方法：表明对象所具有的行为
- 消息：对象之间相互请求或相互协作的途径

3.6.2 面向对象的特点

- 封装：隐藏实现细节
- 继承性：指一个子类继承父类（或称为基类）的特征（数据结构和方法）
- 多态性：不同的对象收到同一个消息产生完全不同效果的特性

3.6.3 面向对象的优点

- 代码维护方便
- 具有好的可扩充性
- 代码重用性好

3.6.4 JAVA的产生和发展

- 1990年，Oak语言的产生
- 1995年1月，Oak 被改名为Java
- 1996年1月，Java开发工具包JDK 1.0推出
- 1996年10月，JavaBeans规范发布
- 1999年6月，Sun 发布JDK 1.3和Java Web Server 2.0

3.6.5 Java的定义

- Java是一种简单的，面向对象的，分布式的，解释的，健壮的安全的，结构的中立的，可移植的，性能很优异的多线程的，动态的语言。
- JDK是Java的基本开发环境

3.6.6 Java的特点

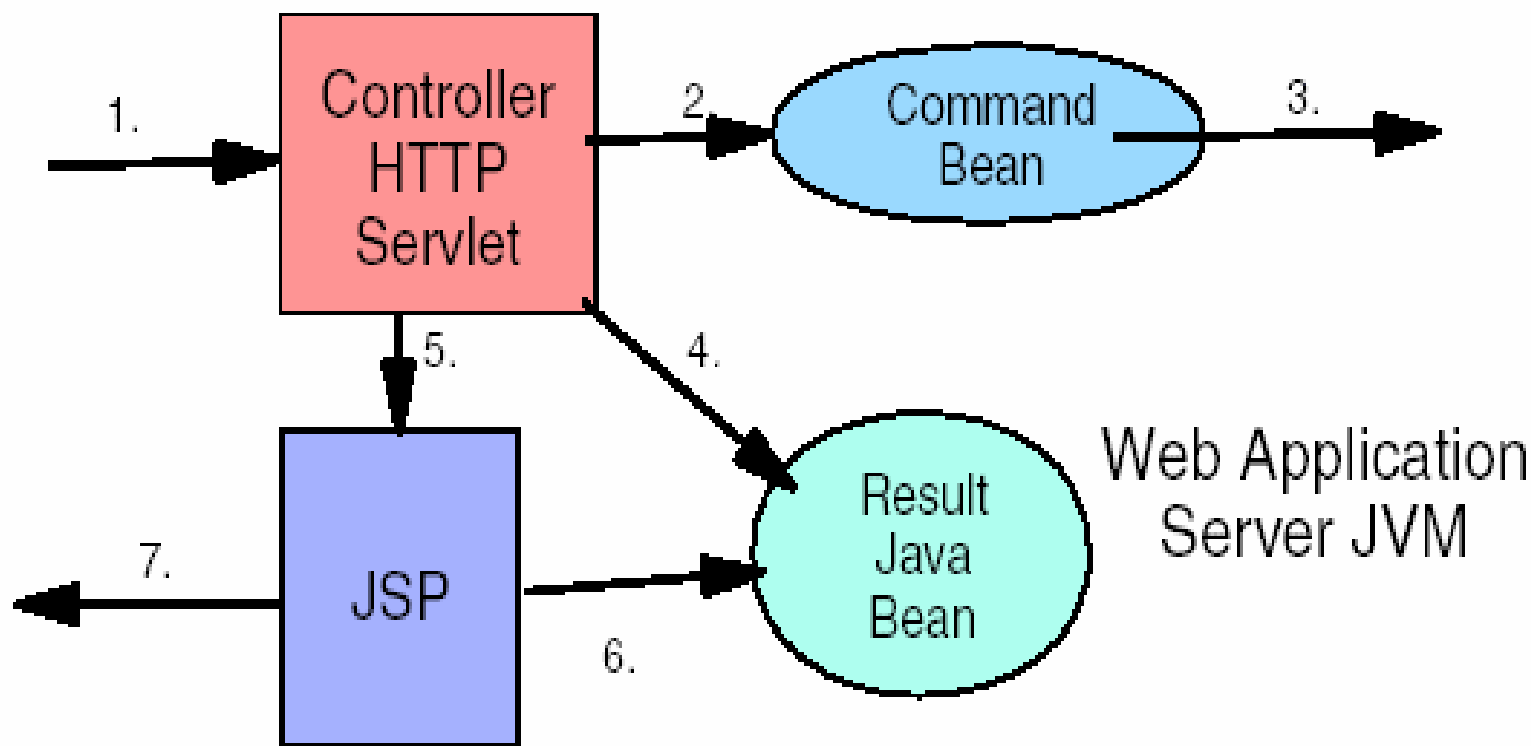
- 平台无关性
- 安全性
- 面向对象
- 分布式
- 健壮性

3.6.7 常用JAVA技术 （一）

- JavaApplet
- Java Servlet
- Java Server Pages （ JSP ）
- JavaBeans
- Enterprise JavaBeans （ EJB ）

3.6.7 常用JAVA技术（二）

JSP, Servlet, JavaBean执行关系图



本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- **3.7 MVC**
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

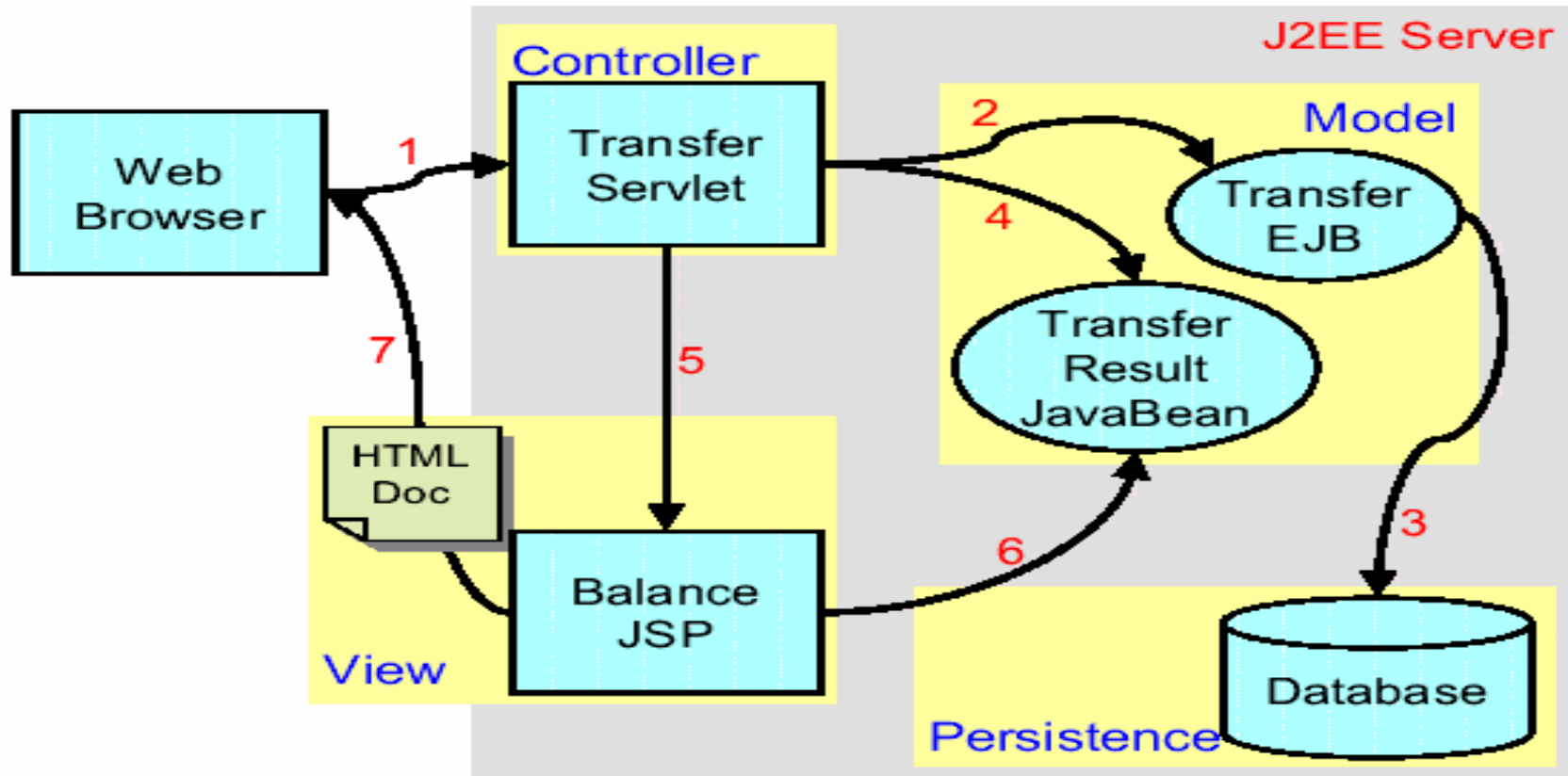
3.7 MVC

- 3.7.1 MVC模式的定义
- 3.7.2 MVC模式的设计实现
- 3.7.3 MVC的优点

3.7.1 MVC模式的定义

- MVC模型：Model-View-Controller 模型-视图-控制模型
- 模型部件（Model）封装了问题的核心数据、逻辑和功能的计算关系。
- 视图部件（View）把表示模型数据及逻辑关系和状态的信息以特定形式展示给用户。
- 控制部件（Controller）接受用户的输入，将输入反馈给模型，进而实现对模型的计算控制，是使模型和视图协调工作的部件。

3.7.2 MVC模式的设计实现



3.7.3 MVC的优点

- 表现与业务逻辑分离，使得表现部分的独立设计和修改维护更方便。
- **MVC**开发模式使得数据访问和数据表现分离开来，让开发人员可以开发一个可伸缩性的强、便于扩展的控制器，来维护整个流程。
- **MVC**模式可以被映射到多层企业级的**J2EE**应用上。

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- **3.8 Web Service**
- 3.9 J2EE
- 3.10 EDI
- 3.11 无线技术

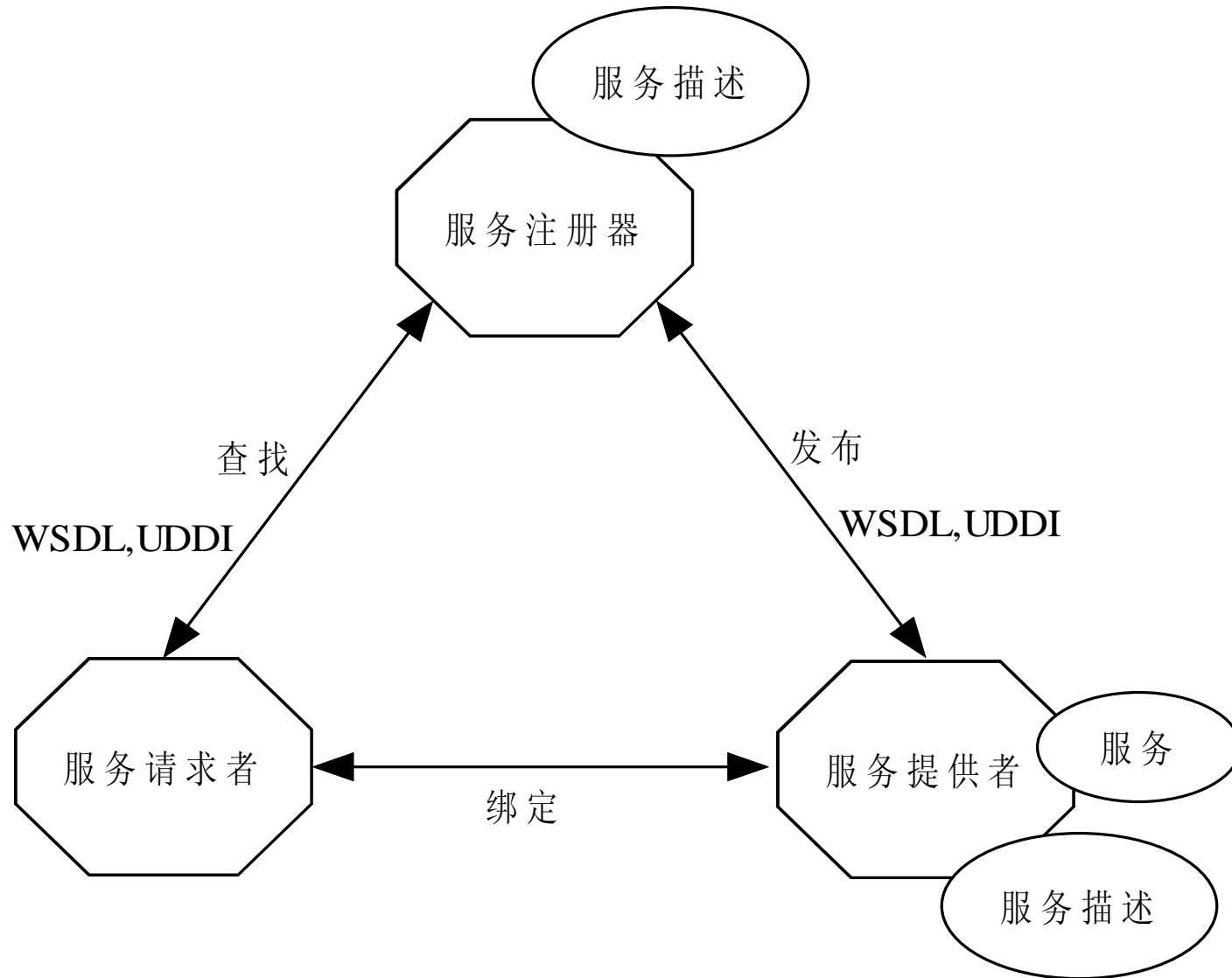
3.8 Web Service

- Web服务使用基于XML的消息处理作为基本的数据通讯方式
- 它消除使用不同组件模型、操作系统和编程语言的系统之间存在的差异
- 它使异类系统能够作为单个计算网络协同运行

3.8 Web Service

- 3.8.1 Web Service体系架构
- 3.8.2 简单对象访问协议SOAP
- 3.8.3 统一描述、发现和集成协议UDDI
- 3.8.4 Web服务描述语言WSDL
- 3.8.5 Web Service的特点

3.8.1 Web Service体系架构



3.8.2 简单对象访问协议SOAP

- SOAP (Simple Object Access Protocol):
简单对象访问协议
- 是一种基于XML的不依赖传输协议的表示层协议，用来在应用程序之间方便地以对象的形式交换数据
- SOAP包括三个部分：SOAP封装结构；SOAP编码规则；SOAP RPC表示

3.8.3 统一描述、发现和集成协议UDDI

- UDDI(Universal Description, Discovery and Integration)：统一描述、发现和集成协议
- 是一套基于Web的、分布式的、为Web Service提供的信息注册中心的实现标准规范。
- 核心组件是UDDI商业注册，它使用一个XML文档来描述企业及其提供的Web Service。

3.8.4 Web服务描述语言WSDL

- WSDL(Web Services Description Language):
Web服务描述语言
- 它把网络服务定义成一个能交换消息的通信
端点集
- 可重用抽象定义:
 - 消息，是需要交换的数据的抽象描述。
 - 端口类型，是操作的抽象集合。
- 定义网络服务时使用的元素：类型；消息；
操作；端口类型；绑定；服务。

3.8.5 Web Service的特点

- 完好的封装性
- 松散耦合
- 使用协约的规范性
- 使用标准协议规范
- 高度可集成能力
- 开放性

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- **3.9 J2EE**
- 3.10 EDI
- 3.11 无线技术

3.9 J2EE

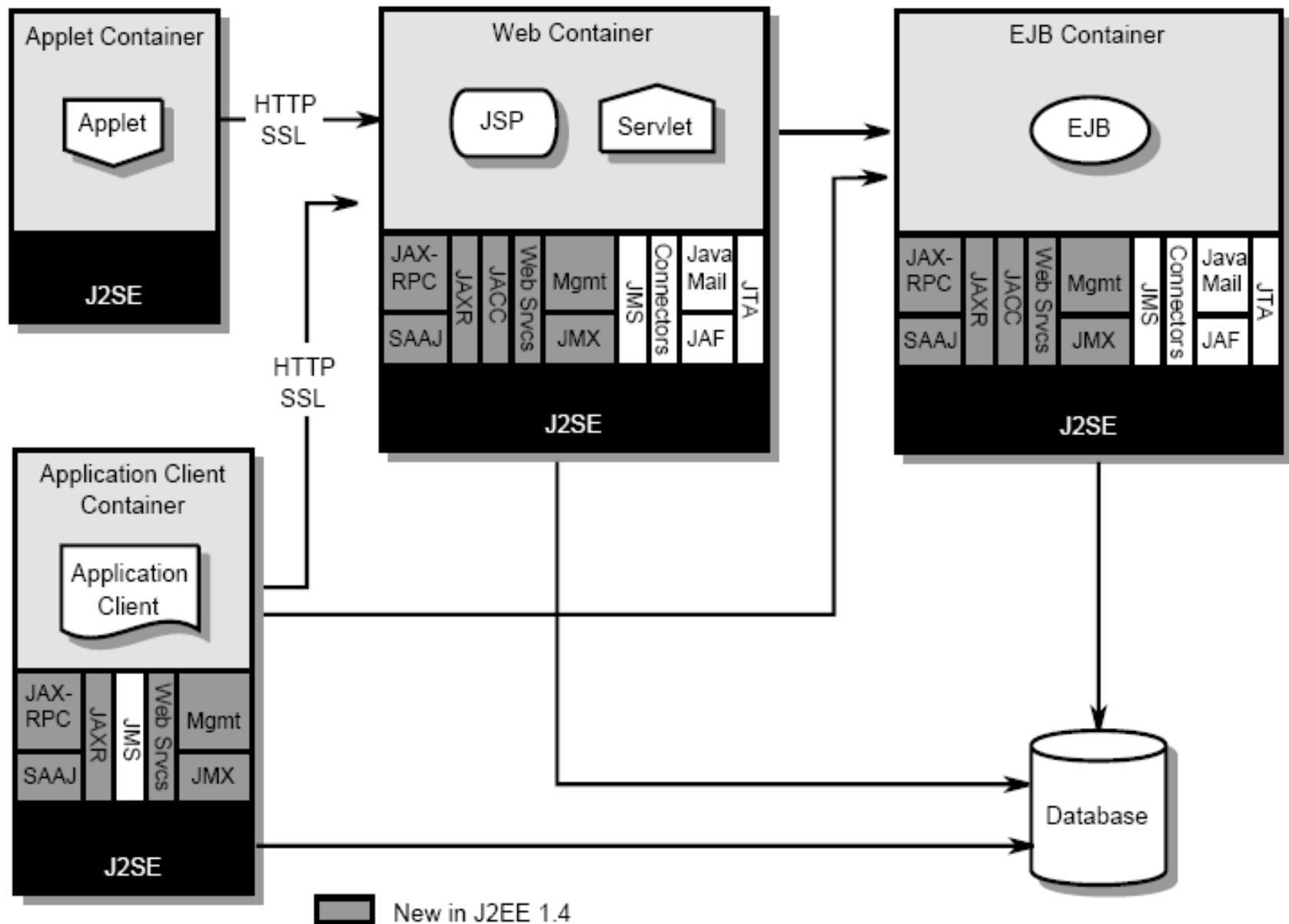
- 3.9.1 J2EE简介
- 3.9.2 J2EE的多层Web框架技术
- 3.9.3 构件技术

3.9.1 J2EE简介

- J2EE是纯粹基于Java的解决方案
- 1999年，Sun正式发布了J2EE的第一个版本。
- IBM的WebSphere、BEA的WebLogic
- 2003年时，J2EE升级到1.4版，其中，Servlet V2.4、JSP V2.0和EJB V2.1

3.9.2 J2EE的多层Web框架技术

- J2EE 1.4 由以下程序容器组成：
 - (1) EJB容器
 - (2) Web容器
 - (3) 应用客户端容器
 - (4) Applet容器
- J2EE 1.4多层次结构Web程序框架图如下一页图所示。



3.9.3 构件技术

- **J2EE**的客户
- **Web**构件
- 企业**JavaBeans(EJB)**构件

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- **3.10 EDI**
- 3.11 无线技术

3.10 EDI

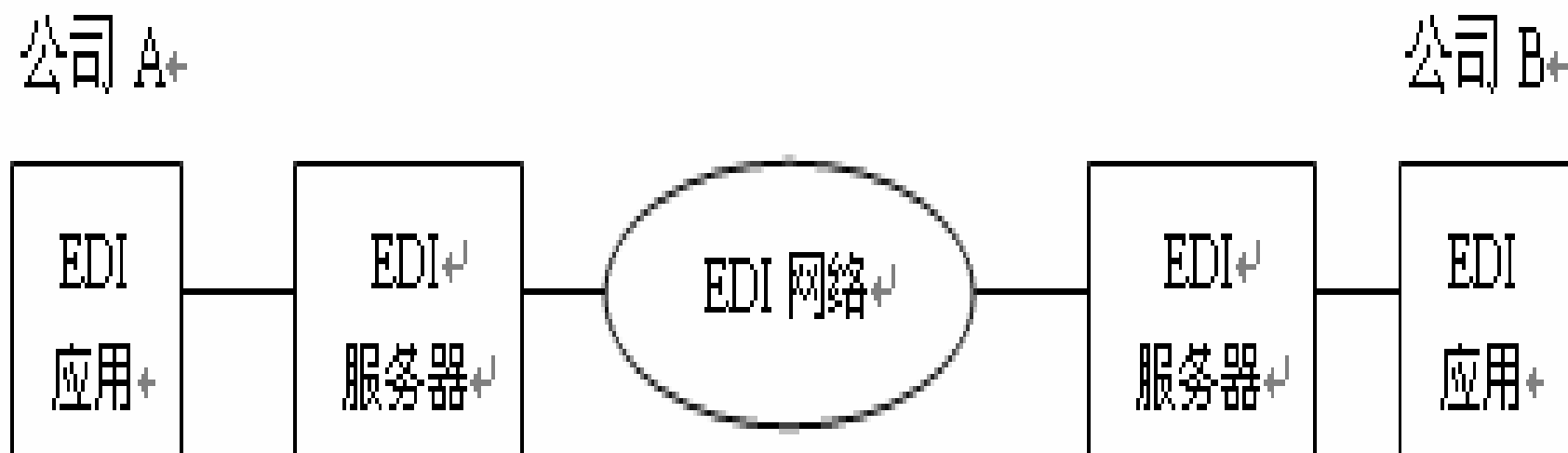
- 3.10.1 EDI简介
- 3.10.2 EDI标准
- 3.10.3 EDI的应用领域
- 3.10.4 EDI 与Internet的结合

3.10.1 EDI简介（一）

- EDI--Electronic Data Interchange 电子数据交换
- 它是一种在公司之间传输订单/发货通知、收据发票、保险单、进出口申报单等文件的电子化手段。
- 它通过计算机通信网络将贸易、运输、保险、海关等行业信息，用一种国际公认的标准格式，进行数据交换与处理，完成商务贸易。

3.10.1 EDI简介（二）

- EDI由通信网络、计算机应用系统和EDI标准化三个基本要素组成。



3.10.2 EDI标准

表 亚太地区 EDI 标准使用情况

国家或地区	使用标准	运营公司
澳大利亚	EDIFACT*	<u>Paxus</u>
新西兰	EDIFACT	GEIS, <u>Netway</u>
新加坡	EDIFACT	SNS
香港	EDIFACT	HKT- <u>CSL</u> , <u>INET</u> , <u>Gazatlenet</u>
日本	N/A	NTT, <u>Data</u> , <u>NEC</u> , <u>IBM</u> , <u>AT&T</u> 等
韩国	ANSI X.12	<u>Dacom</u> , <u>KT-Net</u>
台湾	N/A	DGT, TTN

3.10.3 EDI的应用领域

- 商业贸易领域
- 贸易运输业
- 运输业
- 通关自动化
- 其他领域

3.10.4 EDI 与Internet的结合

- 标准翻译（standard implementation conventions）
- Web_EDI方式
- XML/EDI方式

本章目录

- 3.1 WWW简介
- 3.2 WEB技术的发展
- 3.3 WEB应用模型
- 3.4 客户端技术
- 3.5 服务器端技术
- 3.6 面向对象的概念和java技术
- 3.7 MVC
- 3.8 Web Service
- 3.9 J2EE
- 3.10 EDI
- **3.11 无线技术**

3.11 无线技术

- 3.11.1 移动客户
- 3.11.2 WAP协议
- 3.11.3 WAP应用环境
- 3.11.4 WAP与Internet协议簇的对比

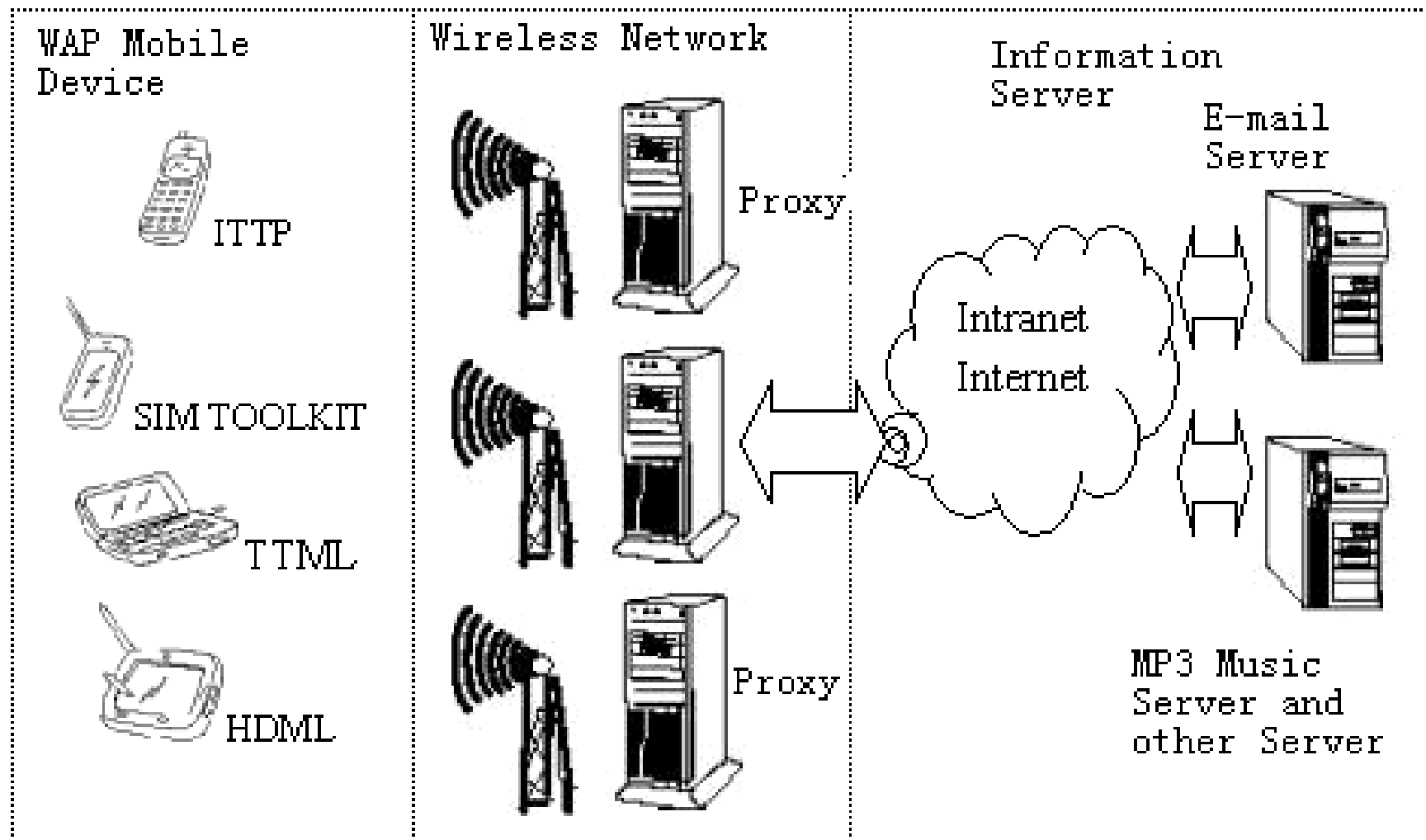
3.11.1 移动客户

- 移动客户通常是手机、寻呼机、PDA等设备。
- 特点：**CPU**不太强、存储容量较少、内存小，功耗受限、显示屏较小等。
- 移动设备有专门的操作平台，最常见的是 **Palm-OS**和 **Windows CE**。

3.11.2 WAP协议

- WAP协议是一个开放的，全球统一的标准。
- 它基于现有的Internet标准，支持移动用户接入Internet。
- WAP协议可以广泛地运用于GSM、CDMA、TDMA、3G等多种网络。

3.11.3 WAP应用环境（一）

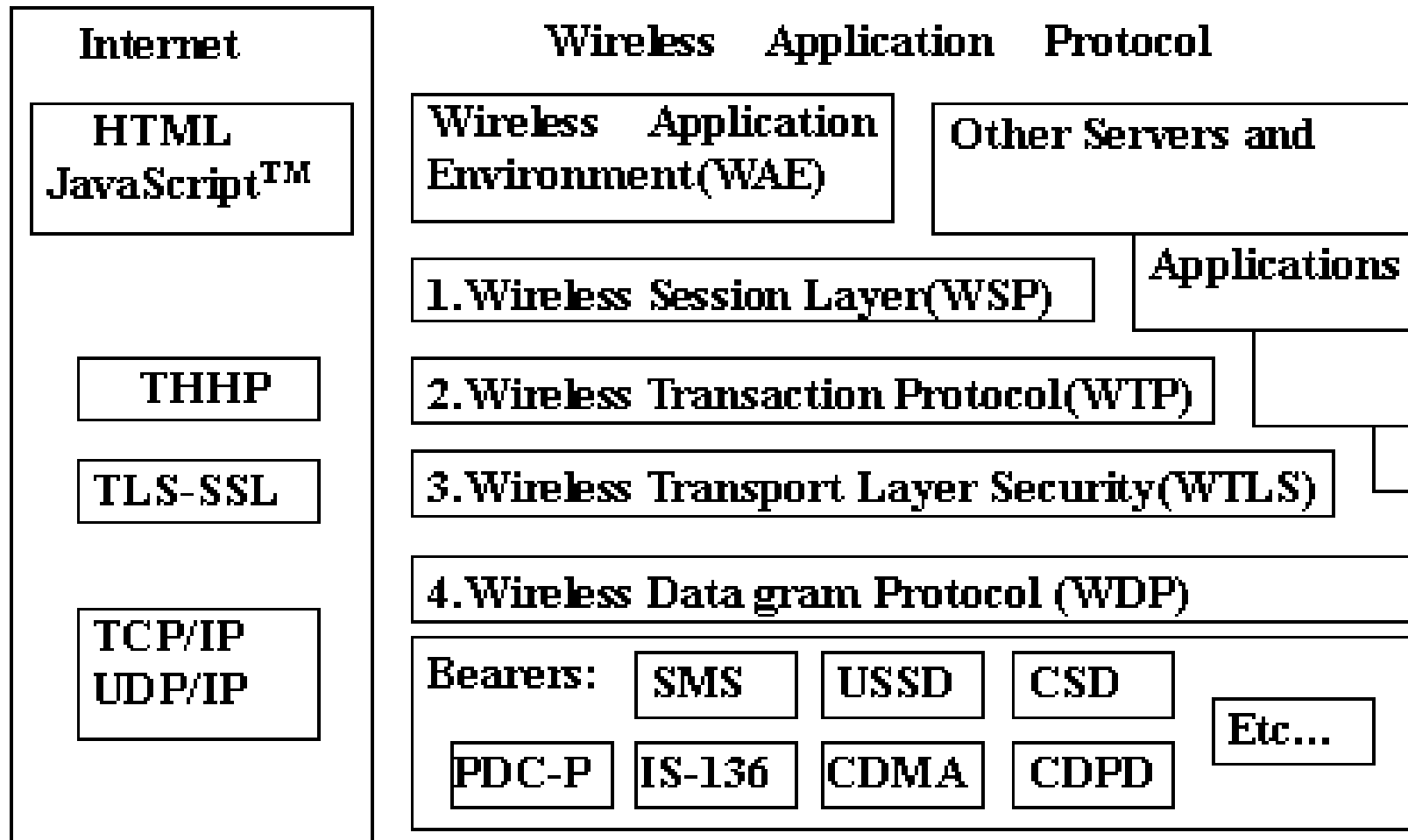


3.11.3 WAP应用环境（二）

WAP应用环境分为三个部分：

- WAP移动设备
- WAP网关
- 信息服务器

3.11.4 WAP与Internet协议簇的对比（一）



3.11.4 WAP与Internet协议簇的对比（二）

- 无线应用环境WAE
- 无线会话层WSL
- 无线传输协议WTP
- 无线传输安全层WTLS
- 无线数据报协议WDP
- 数据载体