

第一讲 内容复习

■ 何谓电子商务

- 指通过信息网络以电子数据信息流通的方式在全世界范围内进行并完成的各种商务活动、交易活动、金融活动和相关的综合服务活动。

■ 电子商务的分类方法

- 按交易对象
 - B2B , B2C , C2C
- 按使用的网络类型
 - EDI
 - Intranet
 - Internet
- 按交易涉及的商品内容
 - 直接与间接

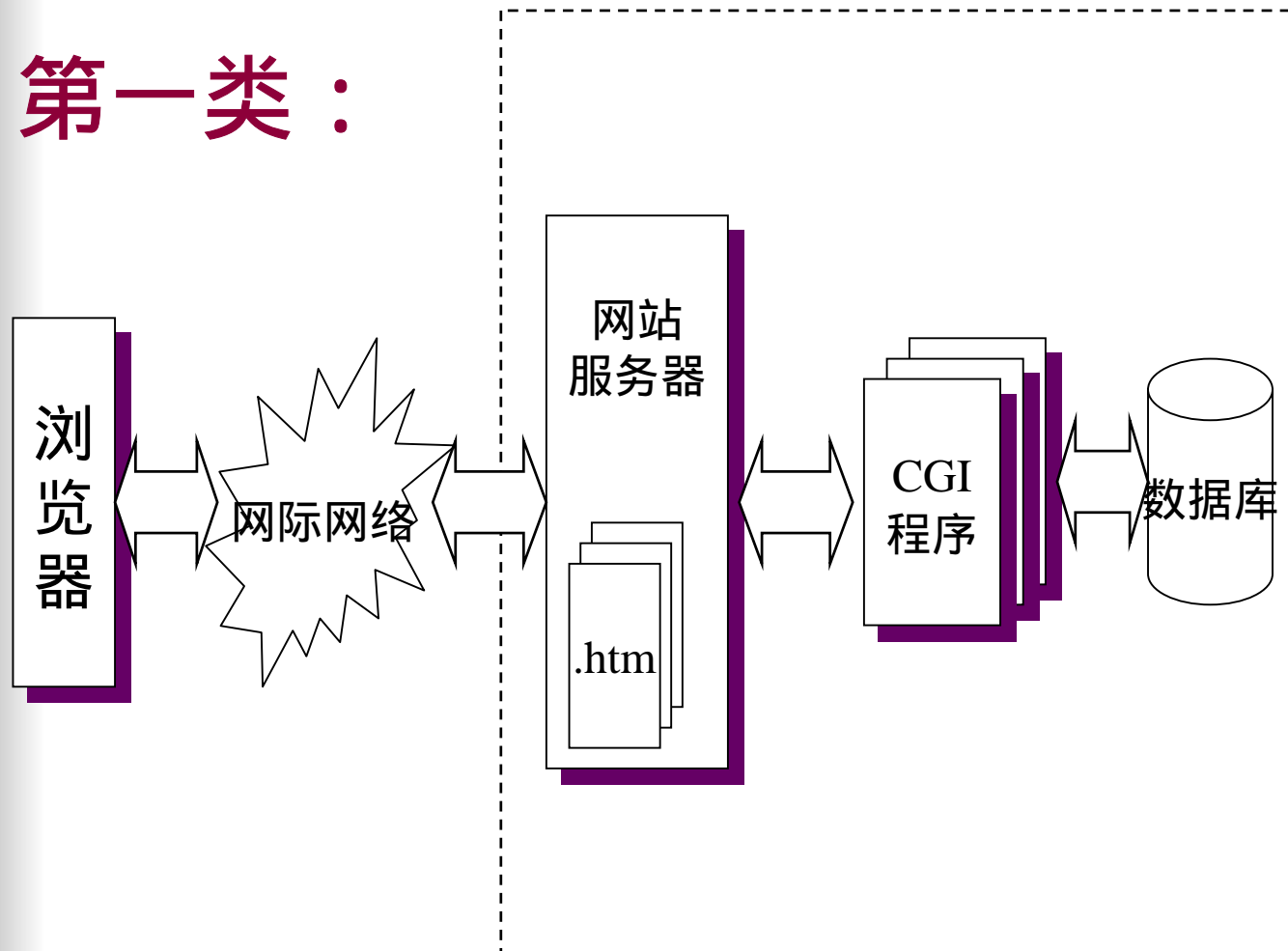


第一讲 内容复习

- e-Business与e-Commerce的区别
- 电子商务的特征
 - 功能特征
 - 经济特征
 - 结合经济
 - 缩短了生产与消费间的信息流、商流、物流距离
 - 网络经济
 - 广泛沟通、全球开放、动态更新、整体关联

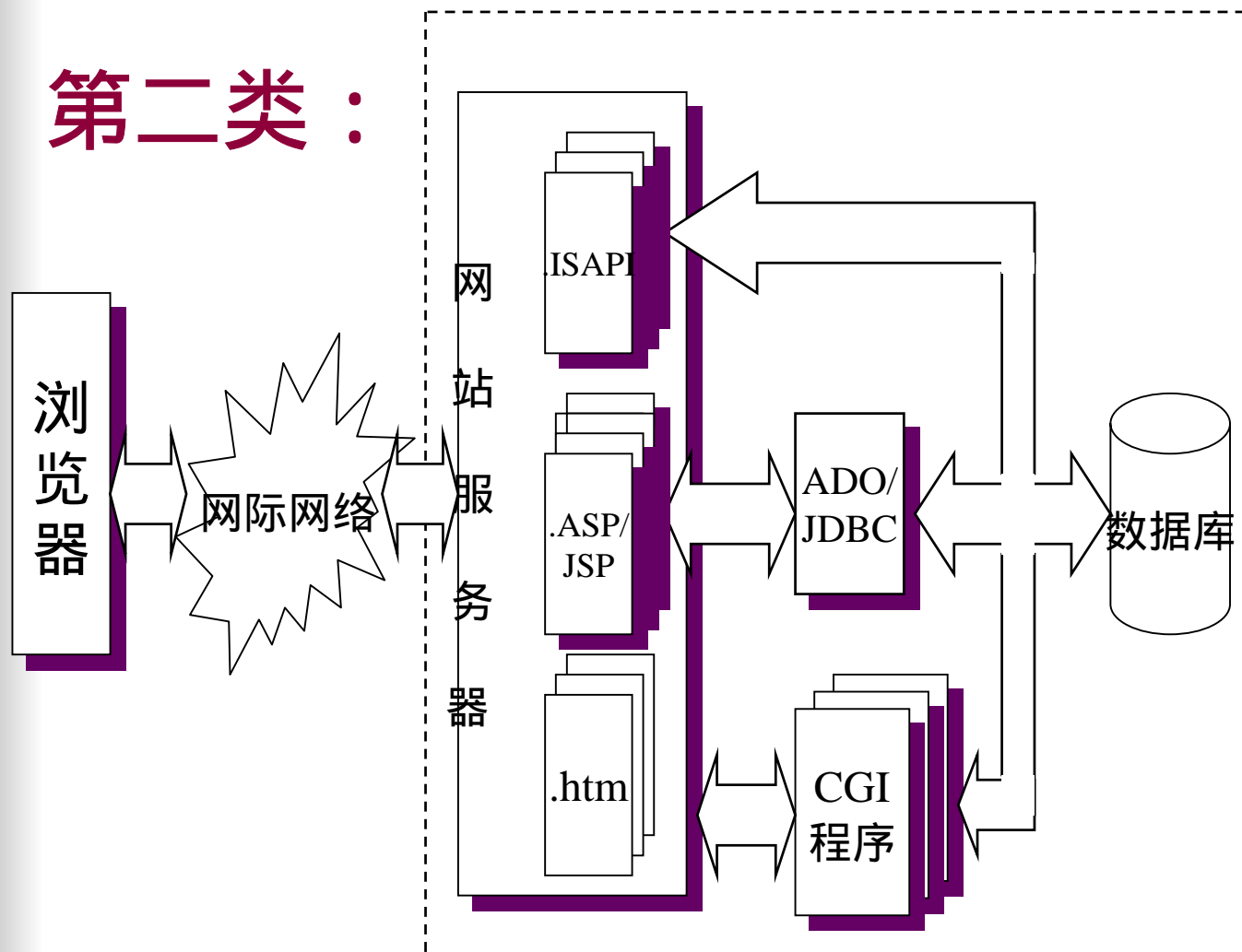
电子商务体系结构

第一类：

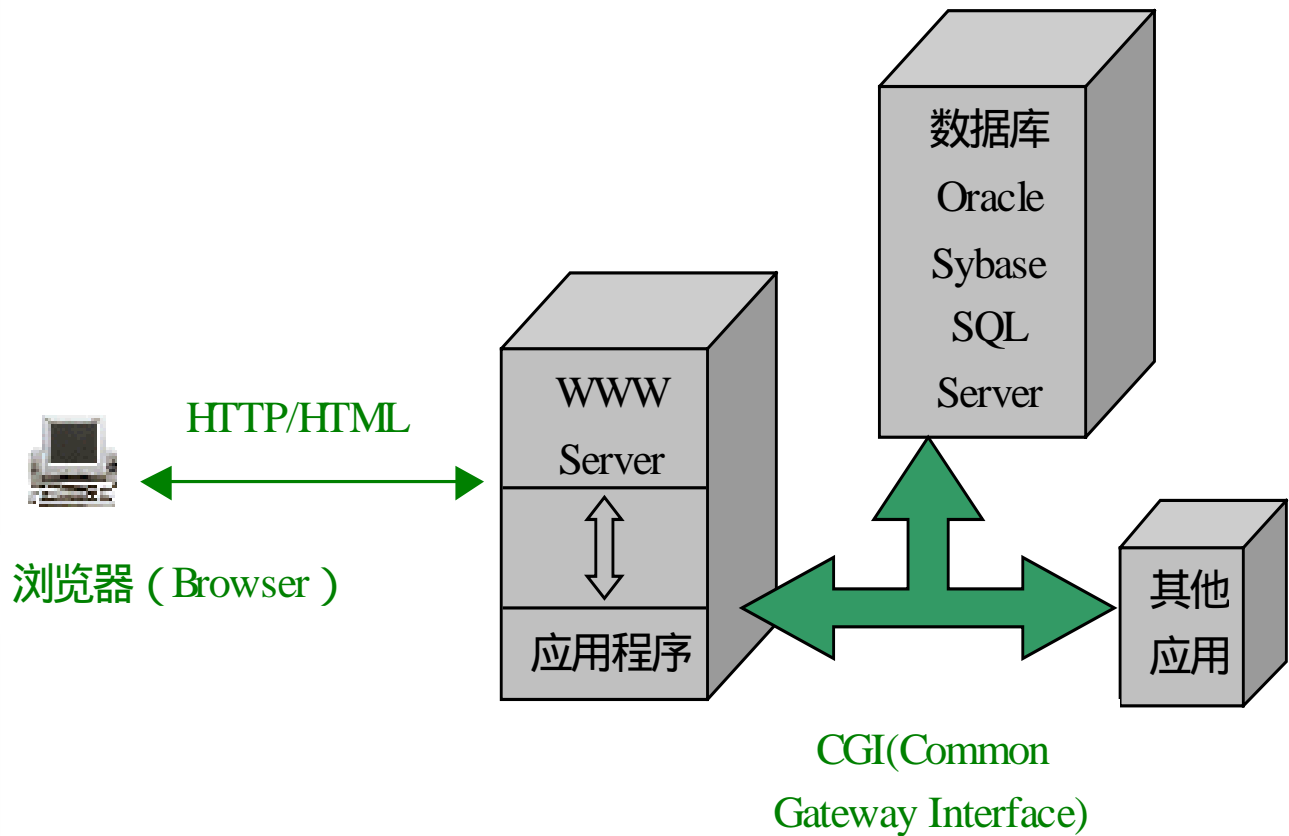


电子商务体系结构

第二类：

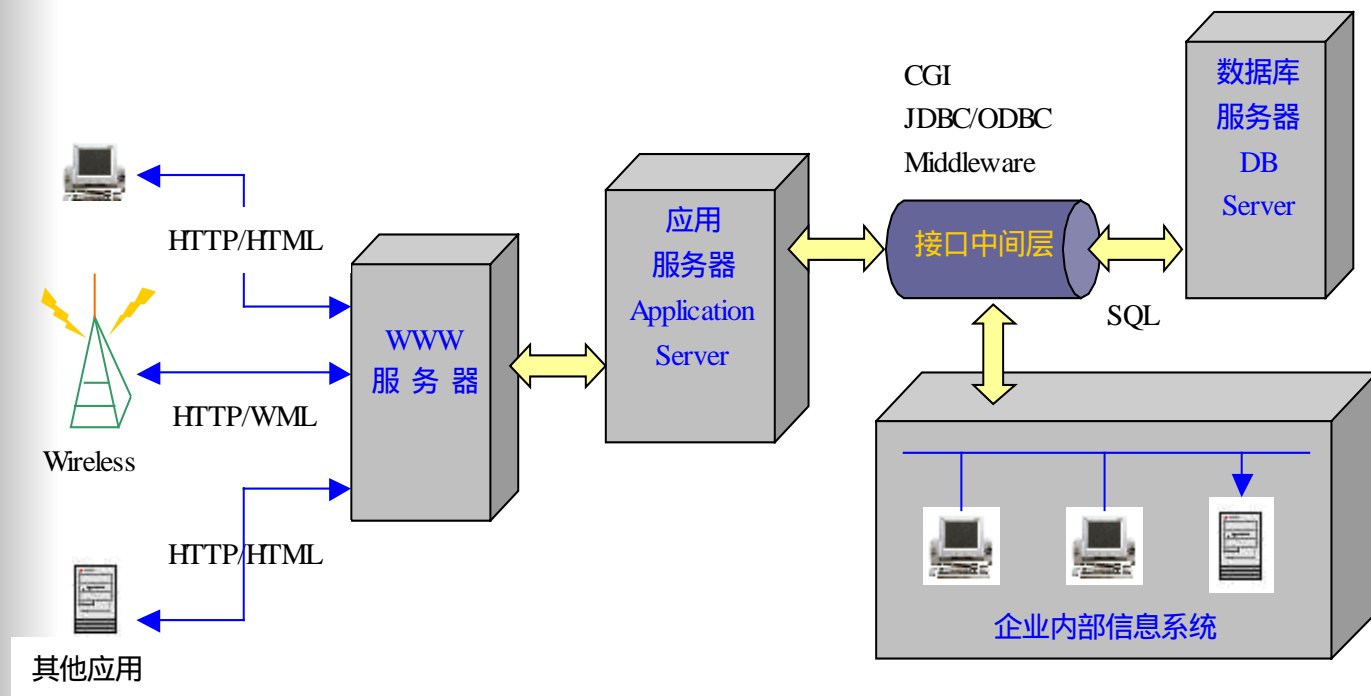


第二发展阶段的电子商务系统结构



■ 第三阶段：电子商务系统的发展阶段（1997年-现在）

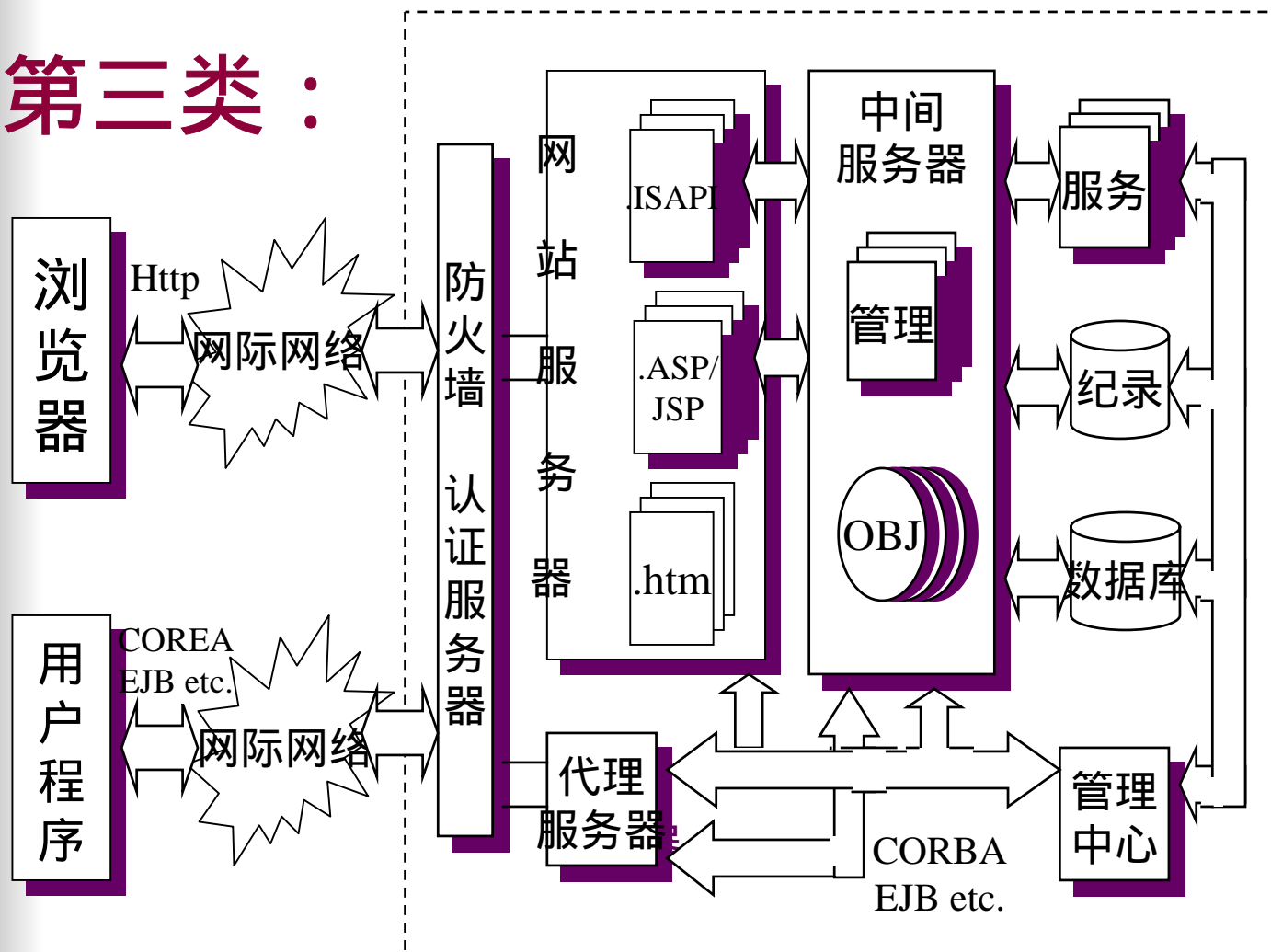
■ (1) 系统与企业内部的信息系统形成一个整体



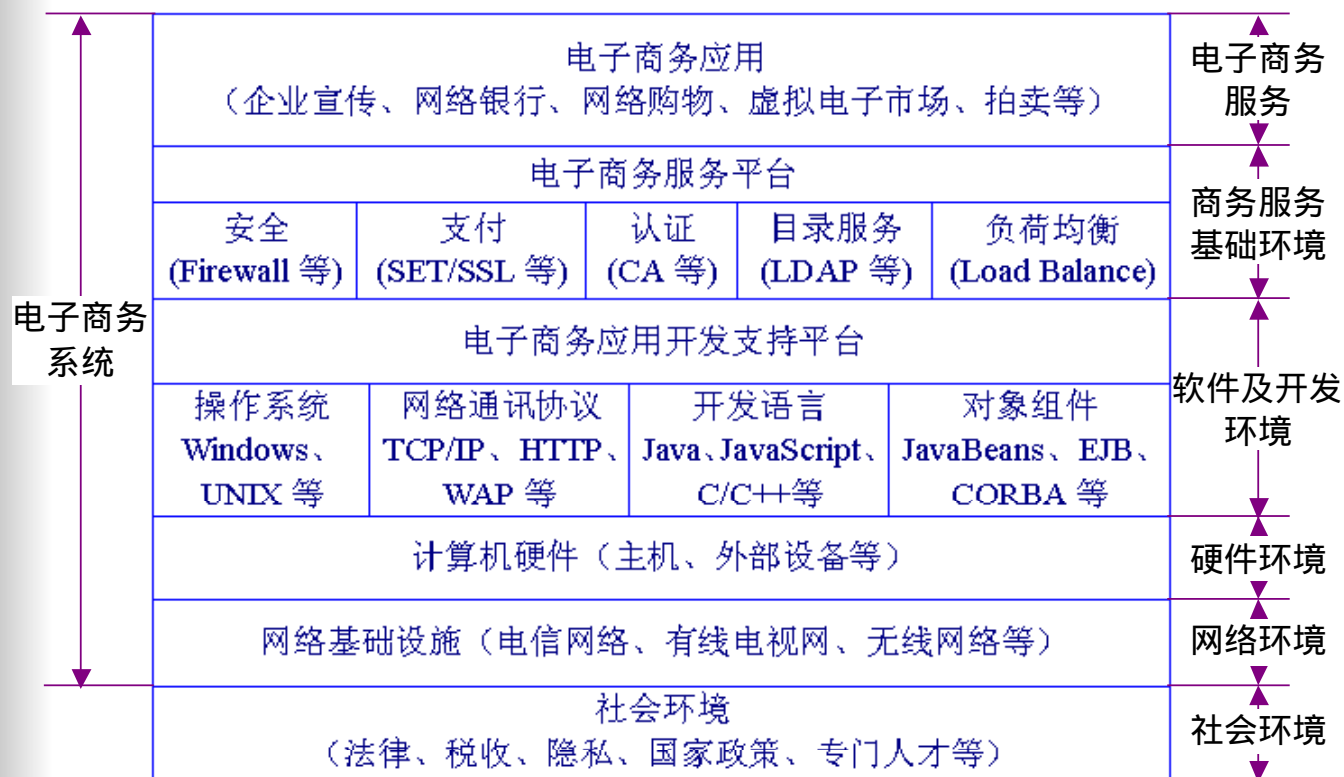
第三发展阶段的电子商务系统结构

电子商务体系结构

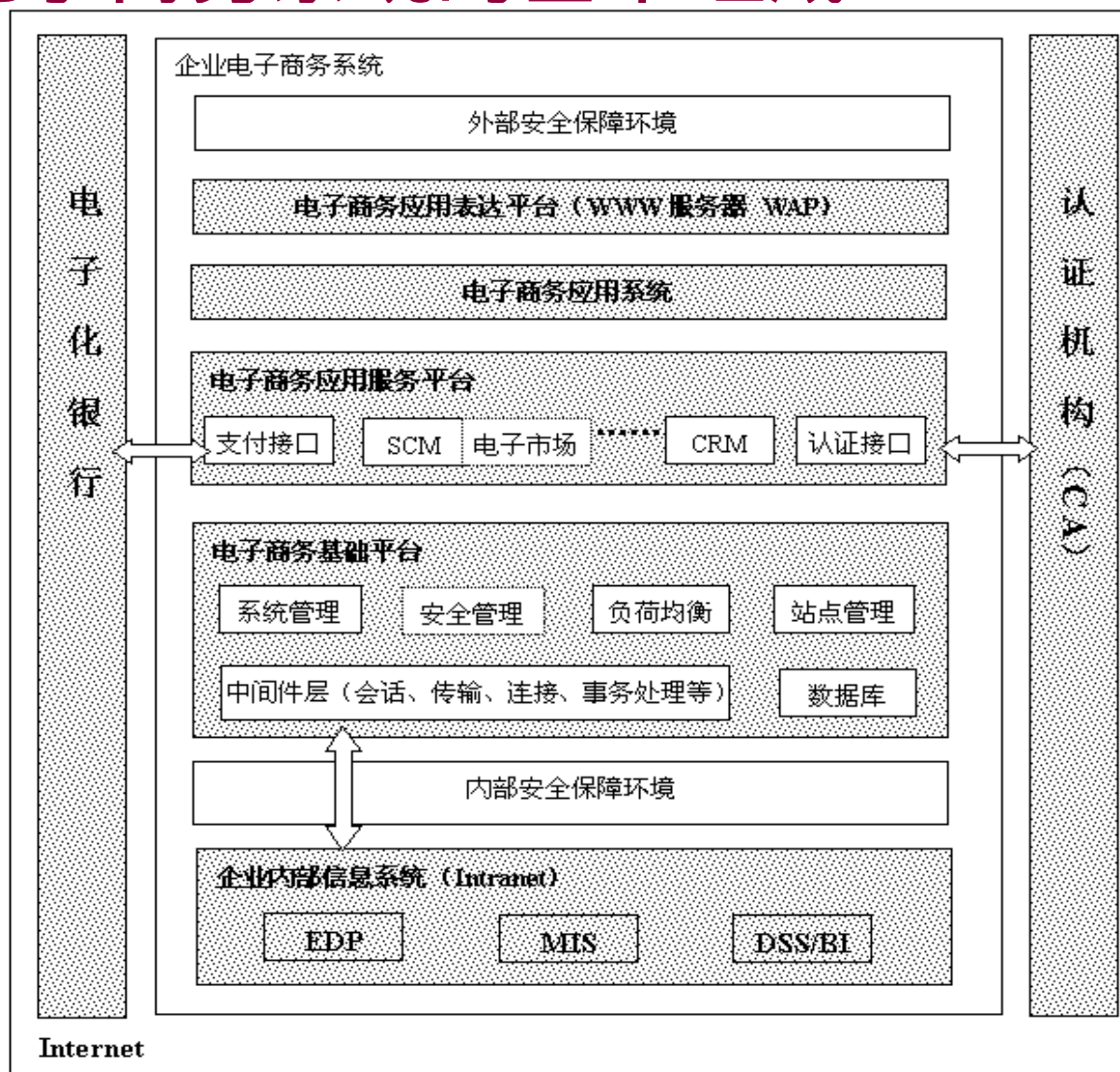
第三类：



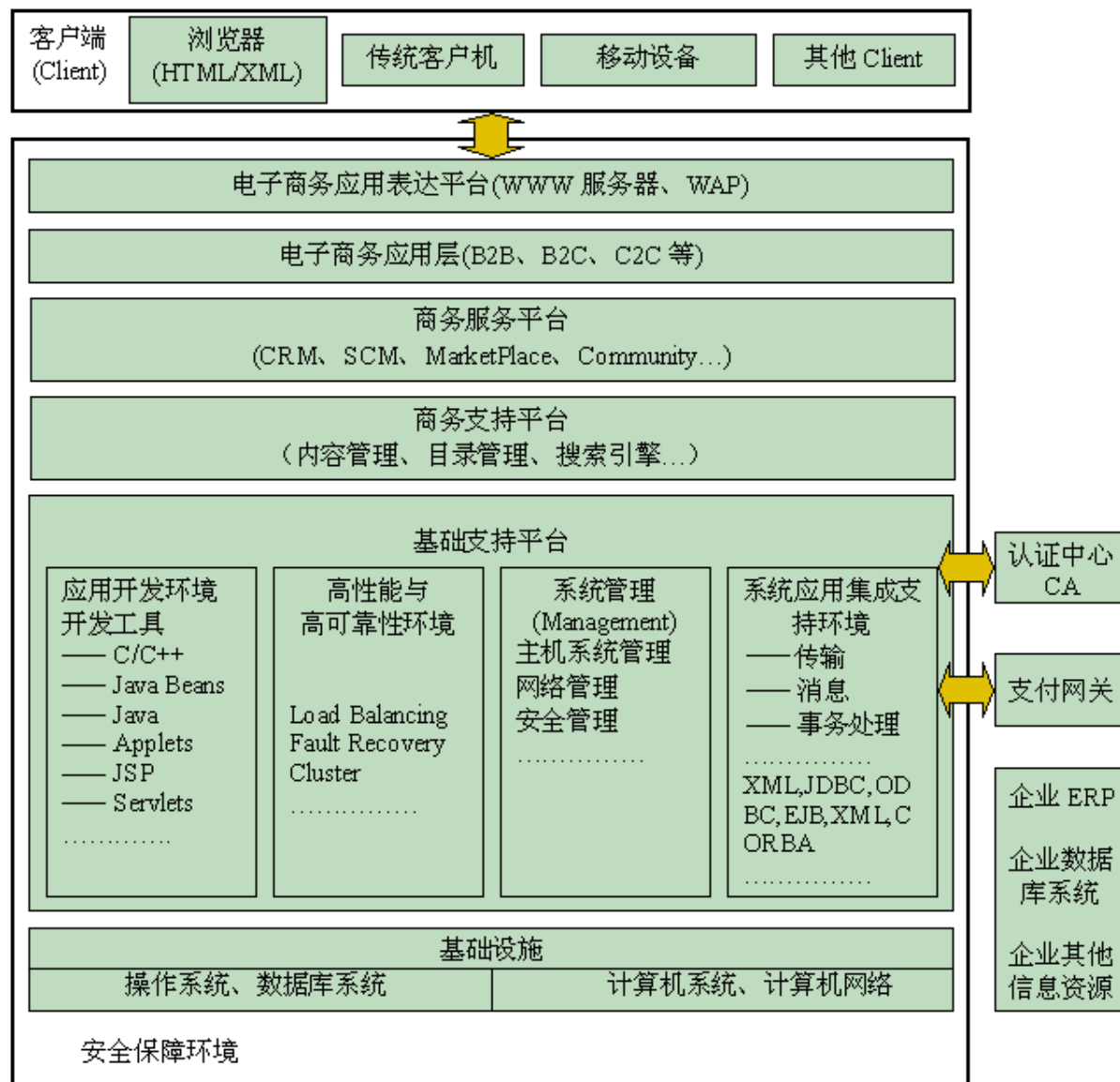
● 2 . 电子商务系统的框架结构



电子商务系统的基本组成



电子商务系统的体系结构





第二讲 电子商务实现相关的前后端技术框架

北京大学计算机系电子商务实验室

主要内容

- 2.1 基础环境技术
- 2.2 电子商务核心技术框架
- 2.3 客户端技术
- 2.4 服务器端技术
- 2.5 前后端数据的连接



2.1 基础环境技术

- 1.计算机网络及互联网络
- 2.TCP/IP协议族
- 3.计算机主机设备

2.2 电子商务核心技术框架

■ 一般构架的发展

在计算机应用，特别是数据库技术的发展过程中，站在平台的角度看，经历了

- 主机系统
- Client/Server (C/S) 架构
- B/W/D结构 (Internet计算环境)

三个阶段。



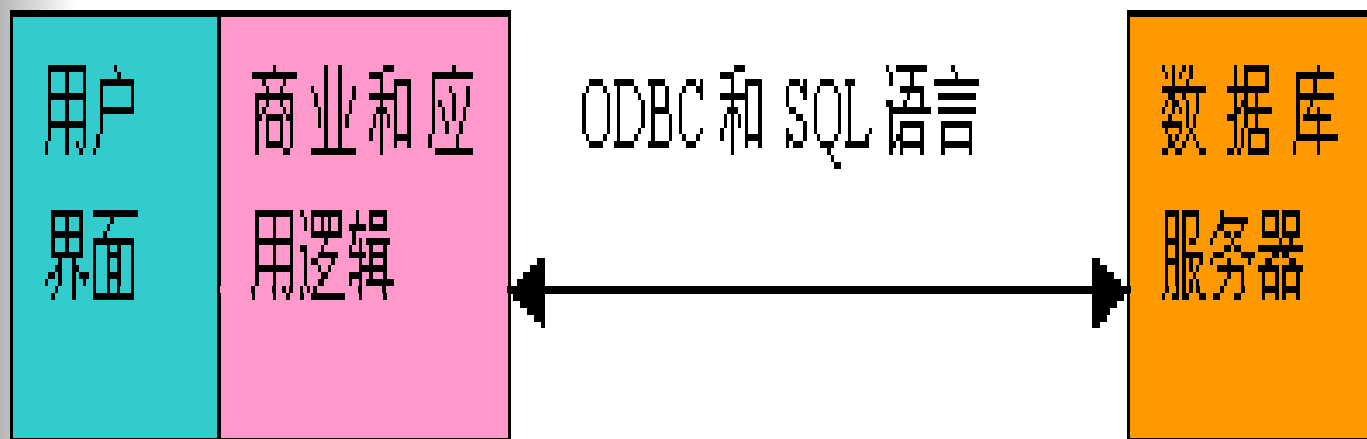
问题：

电子商务系统系统的发展分为几个阶段？

第一代Web应用技术

- 1990's : HTTP和HTML
- (静态的内容) 信息发布
- 内容为one-size fits all
- 多为文本、简单图形加一些超链
- 用户在终端上“冲浪”
- Browser向Server发送请求
- Server通过内部文件系统传送页面内容
- 问题：
 - 易于开发，但难于维护和扩展
 - 用户看到的是消极的 (passive) 页面表示

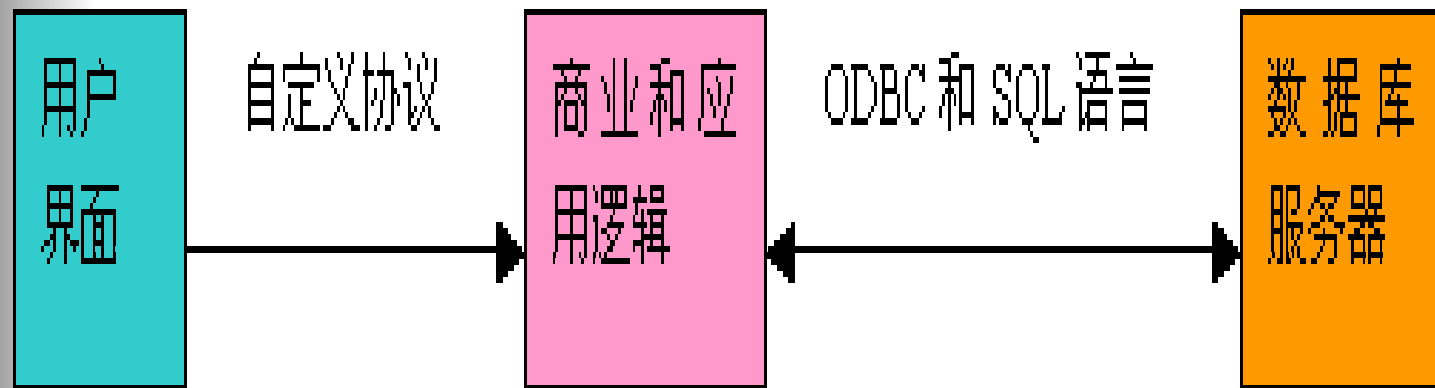
第一代系统基于两层结构



解决办法

- 在客户端嵌入Java Applet、Java Script和ActiveX控件；
- Applet的引入使得提供的浏览器、商业图形、多媒体等对用户更有吸引力；
- Applet代码可以在Web Server的一个类文件中或一个jar文件中；
- 用户在客户端运行Applet;
- 优点：改变了用户接口界面，增加数据的动态表示。

第二代系统是多层系统





三层架构比两层架构的优点

- 整个系统被分成不同的逻辑块，层次非常清晰，一层的改动不会影响其他层次；
- 能够使“肥胖”的客户机变得较“瘦”一些；
- 开发和管理工作向服务器端转移，使得分布的数据处理成为可能；
- 管理和维护变得相对简单。

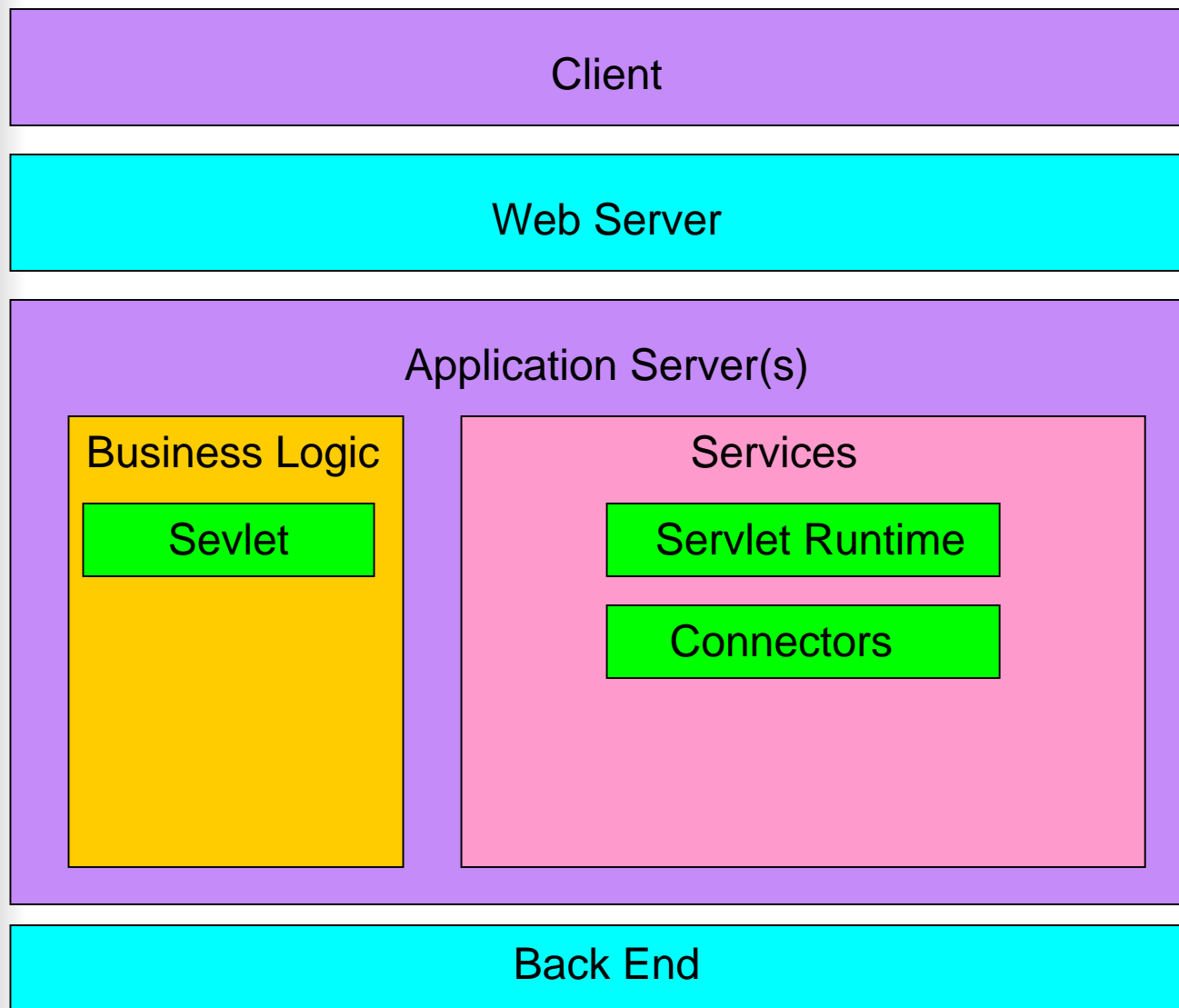
第二代的问题

- 无法从后端系统提取显示每分每秒变化的数据。
- 问题的关键：
 - 需要提供一个gateway将前后端系统连接起来，使得内容是动态的，并且可由用户客户化的。
- 解决方法：
 - 在Server端引入Servlet去解释用户端发来的请求并控制客户化响应的生成，从而将动态内容传给用户。

Servlet的运行机制

- Web Server将动态的内容服务请求发给Application Server；
- Application Server定位请求的Servlet，并且将它的请求参数传递出去；
- Servlet通过连接从后台数据库中检索内容；
- Servlet返回数据给Web Server作为一个HTTP回应。

Server Side Execution



问题

- HTML改变要求程序也跟着改变；
- 当你的Web Site应用逐步增多而需要一个团队保持经常性的修改工作。
- 解决途径：
 - 改进软件复用
 - 在一个清晰的应用结构内建立这些应用，并且使用软件构件
- 要求：
 - Application Server支持基于应用框架的组件；
 - 适应标准化

解决办法

■ 第一步：

- 首先使用Java Beans封装商业逻辑和表示数据；
- 使用Java Server Pages，可以使得其开发者的工作与Java Beans的开发者的工作同时进行；
- JSP与Servlet一起runtime执行。

■ 第二步：

- 增加组件说明商业处理和复用可持续的商业对象；
- 从而引出EJBs(Enterprise Java Beans)。

Server Side Scripting

Client

Web Server

Application Server(s)

Business Logic

Sevlet

JSP

Java Bean

Services

Page Compile

JSP

Servlet Runtime

Back End

Web Application Architecture

Client

Web Server

Application Server(s)

Business Logic

Servlet

JSP, Bean

EJB

Services

Servlet Runtime

Connectors

EJB

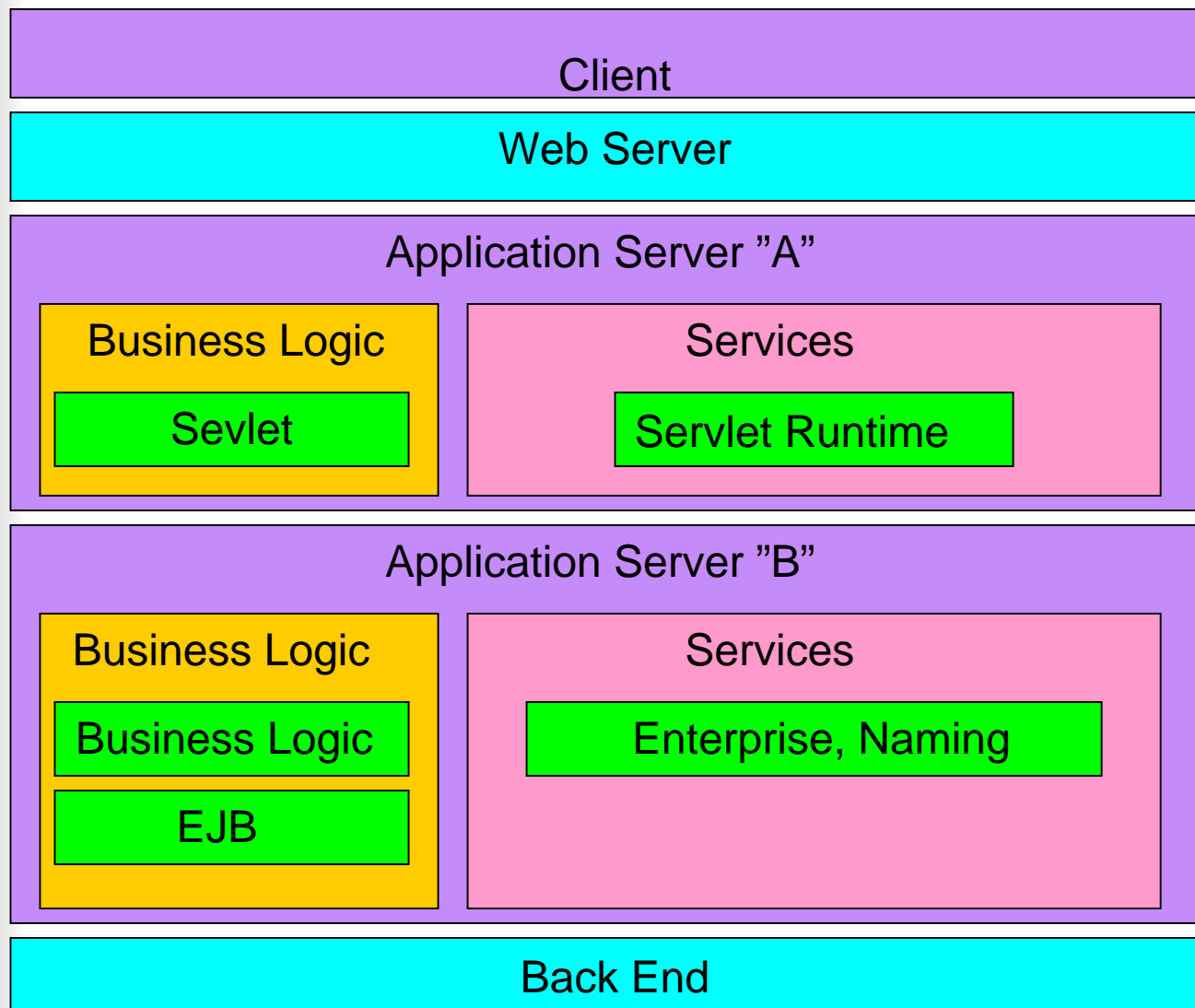
Enterprise
Naming

Back End

问题

- 处理器的能力是exceeded；
- 在几个计算机网络上如何支持多个用户（Multi-user）的交易（transaction）？
- 应用具有可扩展性（scalability）
- 解决途径：分而治之（divide and conquer），将接收的流量（incoming traffic）分发到各个server上：
 - 优点：cluster形式；增加应用的吞吐量；改进系统的可靠性。

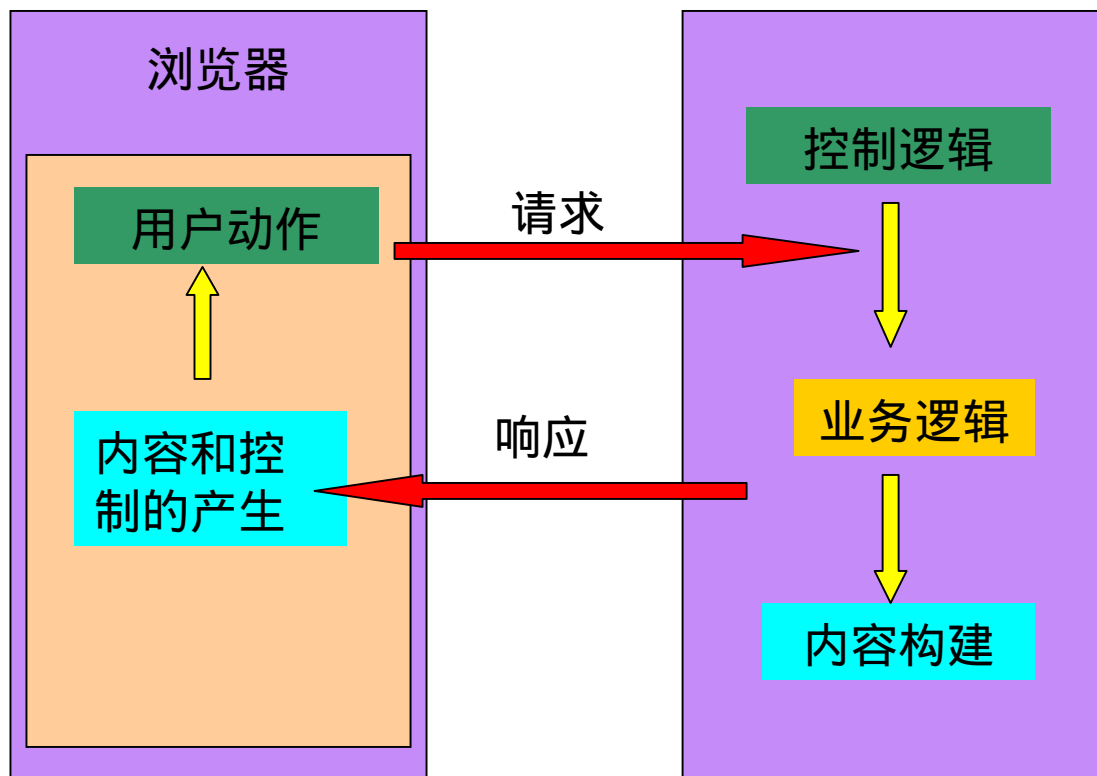
Distributed Web Application



服务器端逻辑体系结构

客户机

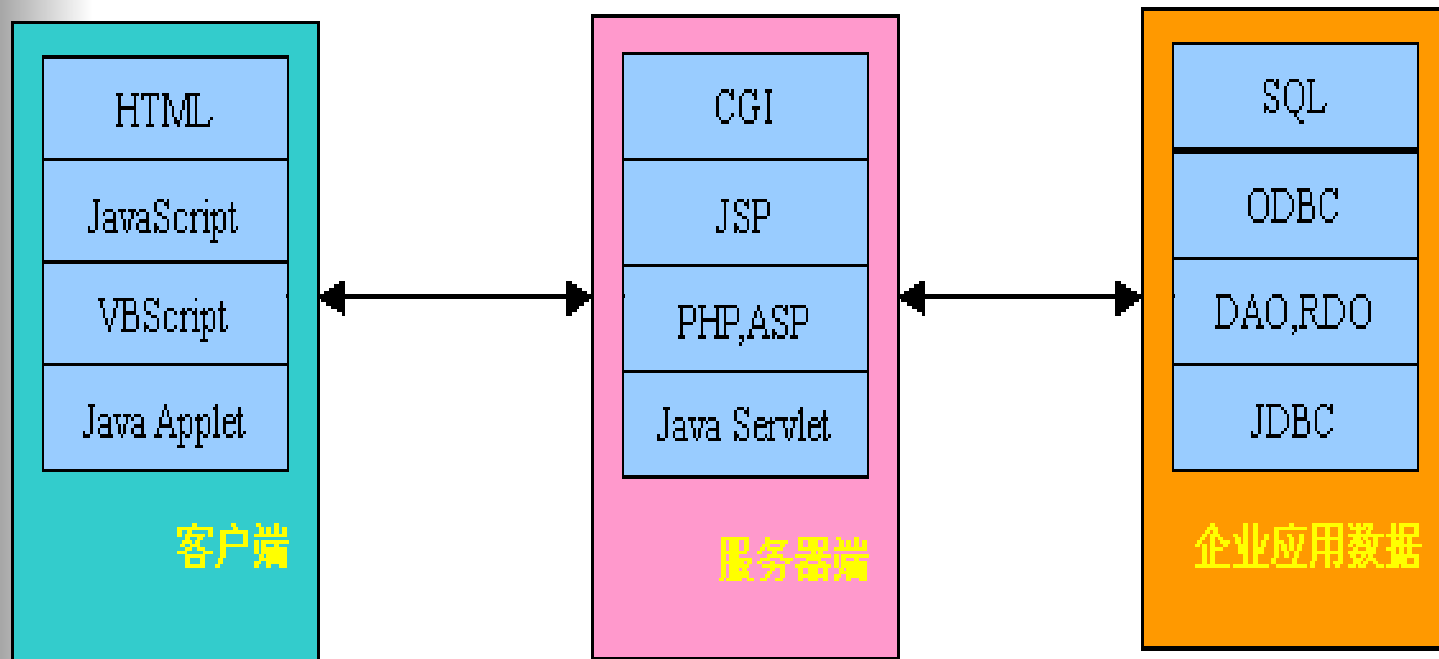
服务器



优点：所需的客户资源少，不用装入应用逻辑，启动用户交互所需的网络通信量少

其支持技术：Java Servlet，Java Server Pages

核心技术框架





问题：

电子商务系统系统的客户端技术有哪些？

2.3 客户端技术

■ 瘦客户与胖客户的比较

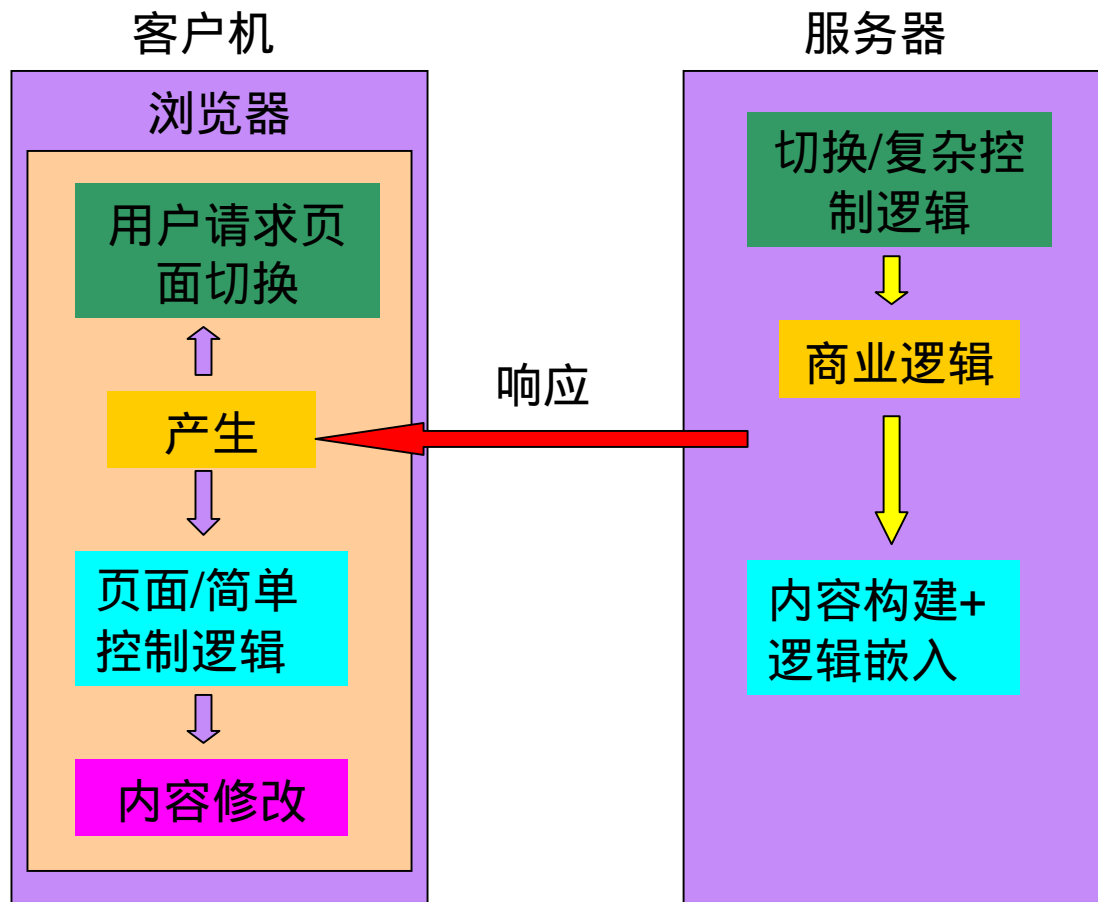
● 胖客户端

- 提供给终端用户较多的功能
- 减少服务器的负担
- 不易使用

● 瘦客户端

- 瘦客户可以集中管理，易部署
- 易使用
- 资源减少（意味着开销减少）

客户端脚本体系结构



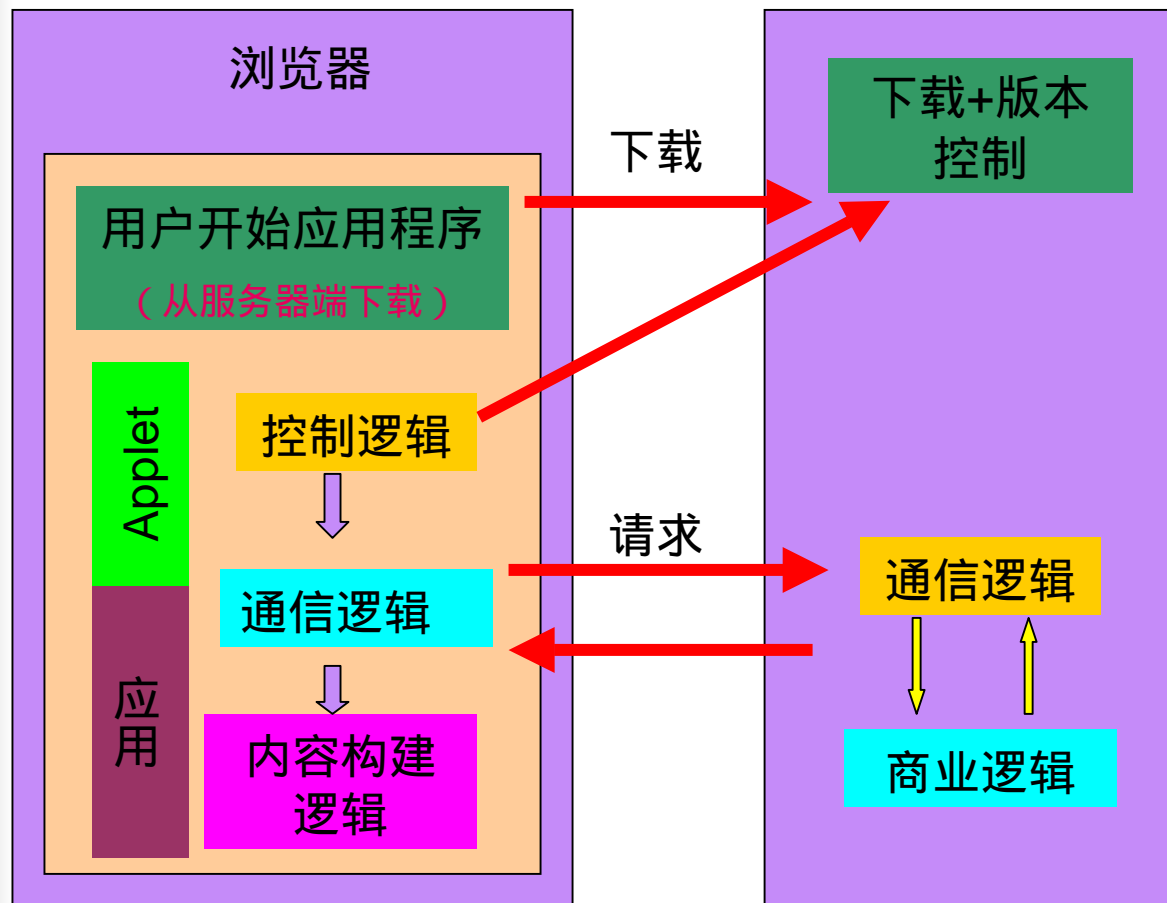
优点：服务器与用户的通信少，需要的服务器资源较少，可对用户的请求作更多响应。

其支持技术：JavaScript, VBScript, DHTML

客户端应用体系结构

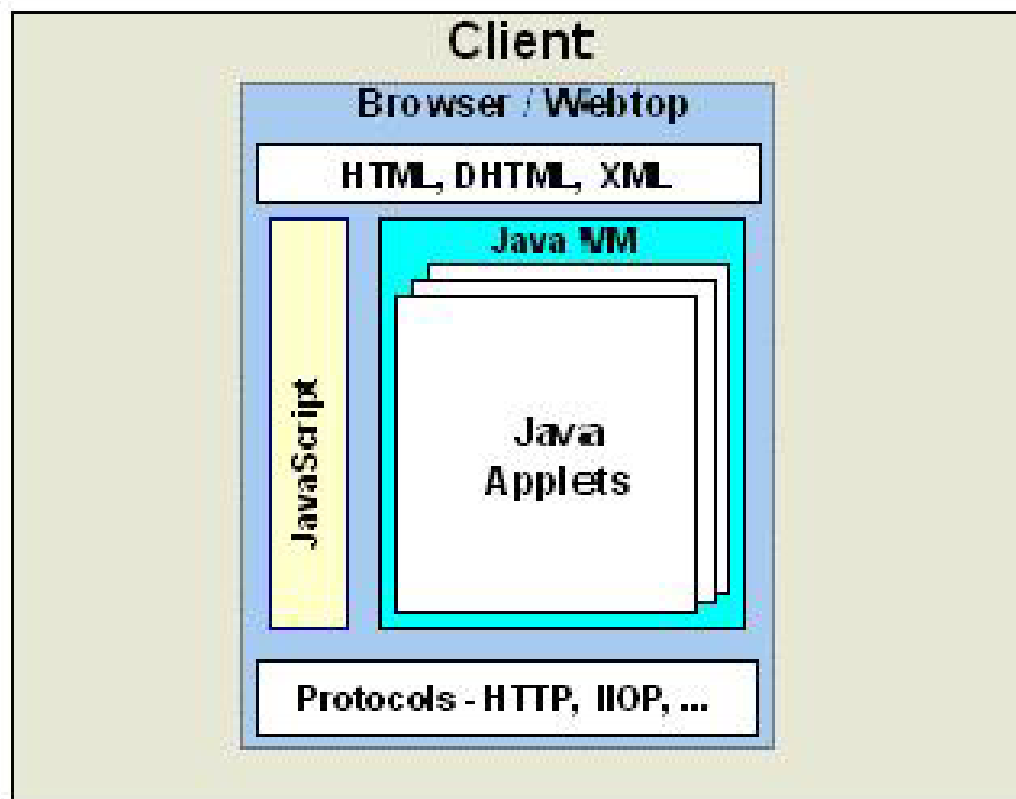
客户机

服务器



优点：省去用户界面和业务逻辑的区别；与用户交互时与服务器的通信较少；Web页面可离线浏览；无需很多的服务器资源。
其支持技术：Java applets、可下载的Java应用程序、ActiveX。

Client Technologies



客户端技术

HTML

HTML是 HyperText Markup Language的缩写，即超文本标记语言，是在WWW上建立超文本文件的语言，它通过标记和属性对一段文本的语言进行描述。HTTP协议接受来自于网络的内容并将接受到的HTML文档显示给用户。同时网络服务器也被设计成能够将HTML文件传送给使用HTTP协议的浏览器。

其主要特点如下：

- 基于文本
- 简易性
- 可扩展性
- 平台无关性
- 集中解释而不是内容方面
- 使用预定标记，主要供浏览使用



VBSCRIPT

VBScript 是微软开发的脚本语言，
同样基于对象，语法结构与VB语言类
似，与IE无缝集成，应用日渐广泛。
VBScript与Basic语言有密切关系。

用户界面体系结构的选择

■ 易用性 (M.E.Holzschlag)

- 隐语的使用
- 清晰性
- 统一性
- 定位性
- 导航性

用户界面体系结构的选择

■ 服务器端逻辑体系结构

- 只能容纳很简单的用户界面（超链接或表单）
- 每次交互均与服务器进行
- 用户难于使用导航帮助
- 仅通过静态表现，难于创建出视觉上的隐语（如图标）

用户界面体系结构的选择

■ 客户端脚本体系结构

- 比纯粹的静态方法提供更为复杂的导航和定位帮助
- 支持给界面元素添加简单的行为

用户界面体系结构的选择

■ 客户端应用体系结构

- 支持更为复杂的语言元素和标准库，以支持比脚本更为复杂的逻辑行为
- 支持比脚本更为复杂的和完美的定位和导航帮助

■ 多媒体内容

- 在提供隐语、导航、定位方面境界更高
- 采用VRML技术

用户界面体系结构的选择

■ 性能与瓶颈的考虑

- 下载时间（即初始化时间）
- 客户装入（即客户端处理用户交互）
- 服务器依赖度（与服务器交互的次数）
- 网络传输时间（客户端向服务器端传输消息和反馈结果所需时间）
- 服务器装入（服务器处理用户交互所需时间）



用户界面体系结构的选择

■ 性能与瓶颈的考虑

● 服务器端逻辑体系结构

- 用户逻辑不必下载
- 对服务器的依赖较高
- 服务器的响应不仅包含需要的数据，还有呈交这些数据所需的设计定义，从而导致大量的信息传输

● 客户端脚本体系结构

- 脚本提供了在客户端上运行一些控制逻辑，故与服务器的交互减少
- 较高级别的逻辑和与HCI不相关的逻辑仍需驻留在服务器上
- 大量的脚本下载需要一定的时间
- 对用户资源（如内存）较大

用户界面体系结构的选择

■ 性能与瓶颈的考虑

● 客户端应用体系结构

- 提供了最好的响应效果
- 与服务器交流更少
- 内容构建在客户端进行
- 仅传输网络数据
- 缺点是在第一次交互前需较长的下载时间，可采用以下技术改进：
 - 延迟模块装载
 - 应用程序分割
 - 客户端的安装过程
 - Applet缓存



问题：

电子商务系统系统的服务器端
技术有哪些？

2.4 服务器端技术

- 支持动态页面生成的主要技术
 - CGI
 - 服务器端脚本技术
 - 插件技术
 - Servlet技术
 - JSP技术



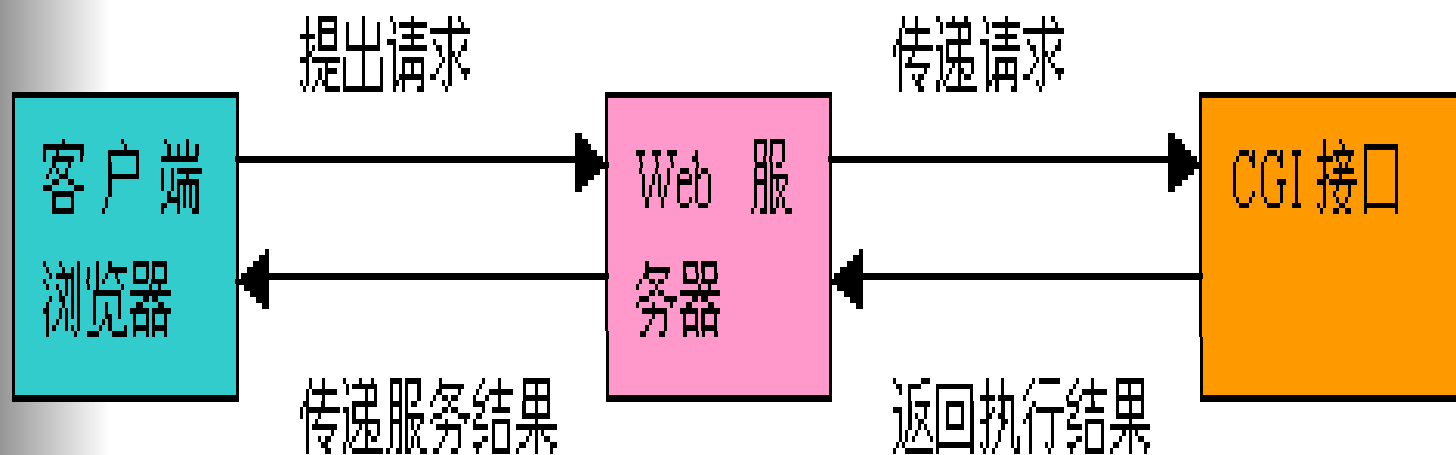
CGI

CGI是Common Gateway Interface的简称。其主要的功能是在WWW环境下，从客户端传递一些信息给WWW Server，再由WWW Server去启动所指定的程序码来完成特定的工作。所以更明确的说，CGI仅是在WWW Server上可执行的程序码，而它的工作就是控制信息要求而且产生并传回所需的文件。

- 用Perl或C编写
- 与特定的服务器平台相关

CGI

- 对每一次访问CGI程序的HTTP请求，服务器端均会产生一个新的进程来处理该请求，从而限制了服务器同时处理多个请求的能力；
- CGI程序无法与Web服务器交互，因为这两者运行在不同的进程中。



服务器端脚本技术

- 指服务器端脚本语言和环境，可以使Web开发者在Web页面中直接嵌入脚本来生成动态的页面，该脚本由服务器解释，从而产生动态内容。
- 典型代表技术：
 - IBM的Net.Data
 - Microsoft的ASP
 - Allaire的ColdFusion
- 缺点：
 - 无跨平台性，限于某类OS
 - 开发者必须掌握某种特定的脚本语言

插件技术

■ 产品

- Netscape的NSAPI
- Microsoft的ISAPI

■ 优点

- 性能好

■ 缺点

- 与服务器的耦合度较高
- 编程难度较大

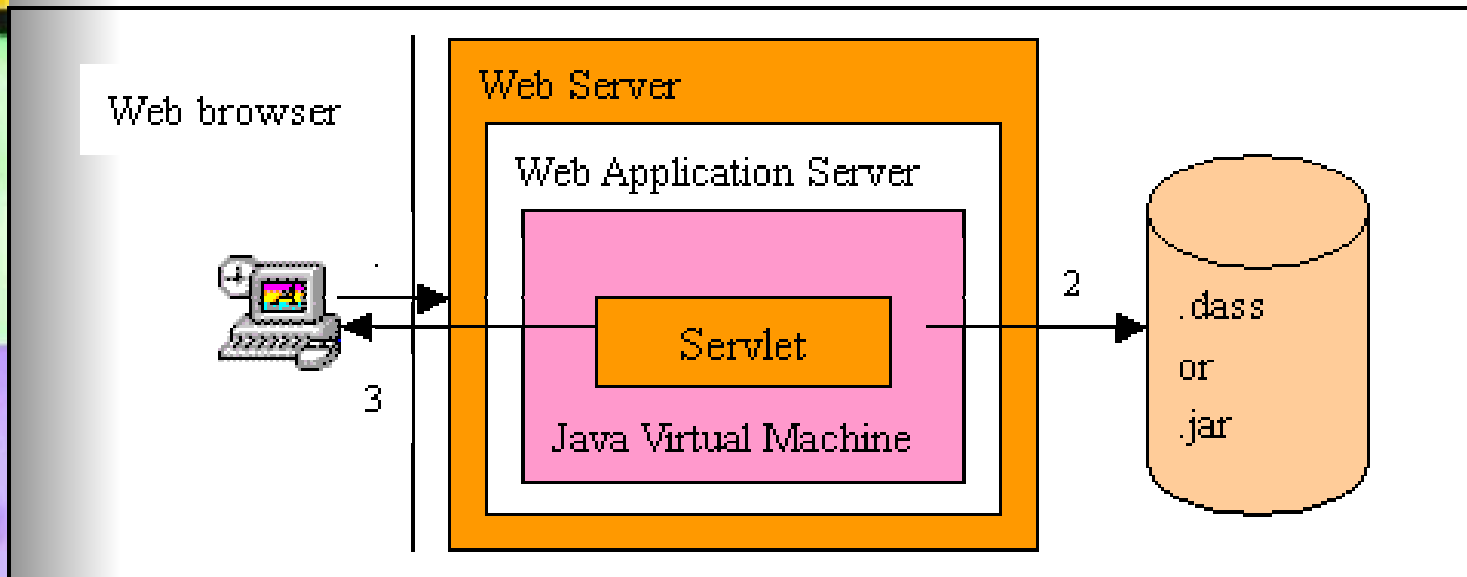


Java Servlet

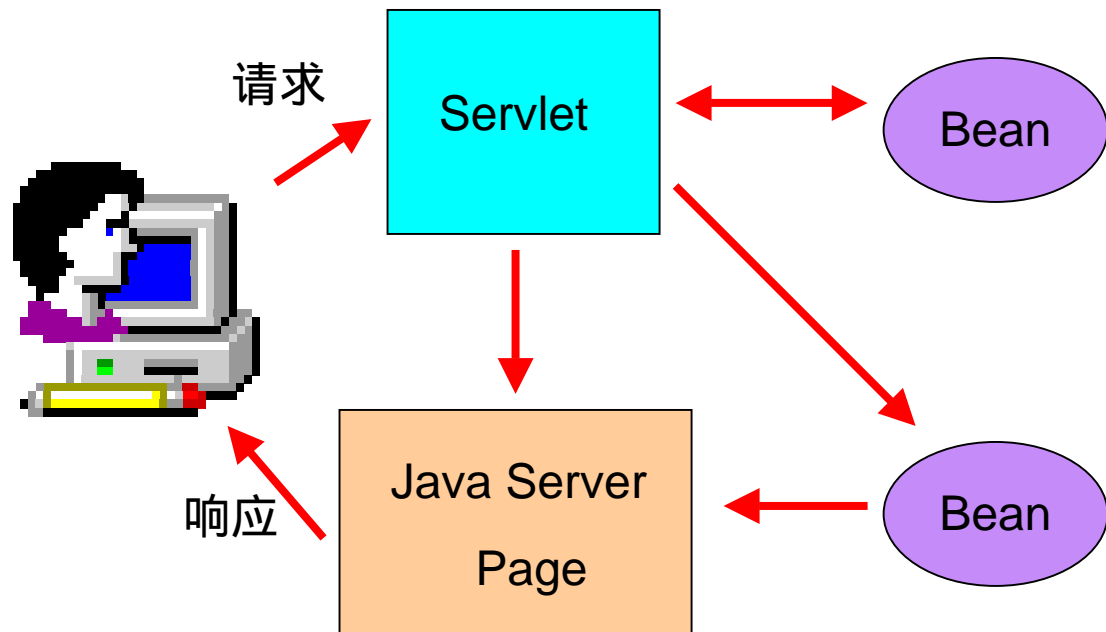
- Servlet可被认为是服务器端的applet
- Servlet是使用Java Servlet应用程序设计接口(API)及相关类和方法的Java程序
- Servlet用Java编写，与平台无关
(Write once, run anywhere)

Servlet的基本流程

- 客户端通过HTTP提出请求
- Web服务器接收该请求并将其发给Servlet；若此Servlet尚未被加载，则Web服务器将其加载到Java虚拟机上并执行它
- Servlet将接收该HTTP请求并执行某种处理
- Servlet将向Web服务器返回应答
- Web服务器将从Servlet收到的应答发送给客户端



Servlet作为控制器





Servlet的特点（CGI不具备）

- 持久性和高性能：加载一次，即可在不同请求之间保持服务
- 与平台无关性：基于Java
- 可扩展性：Java的优点
- 安全性：外部调用一个Servlet的唯一方法是通过Web服务器
- 可移植性：通过Servlet API定义的一个Servlet和Web服务器的软件接口，达到在多种类客户机上使用的目的。



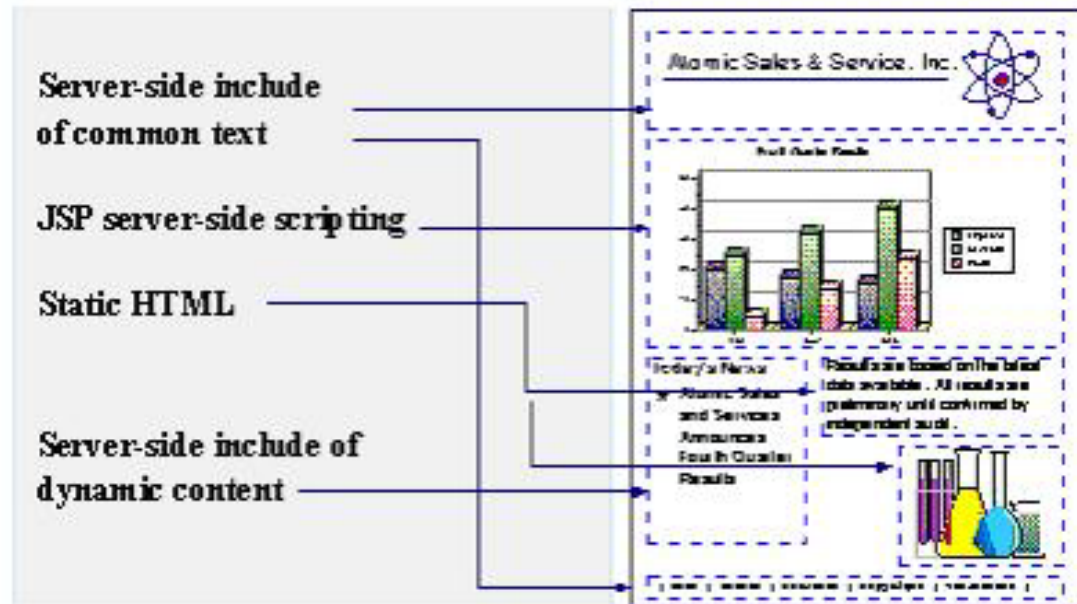
JAVA SCRIPT

Java Script是Netscape和SUN共同开发的脚本语言，语法结构与Java语言相似，并且是基于对象的，在互联网上应用最广泛。JavaScript 是一种新的描述语言，它可以被嵌入到HTML 的文件之中。

JSP

- JSP(Java Server Pages)是由Sun Microsystems公司倡导、许多公司参与一起建立的一种动态网页脚本技术标准。

Java Server Pages

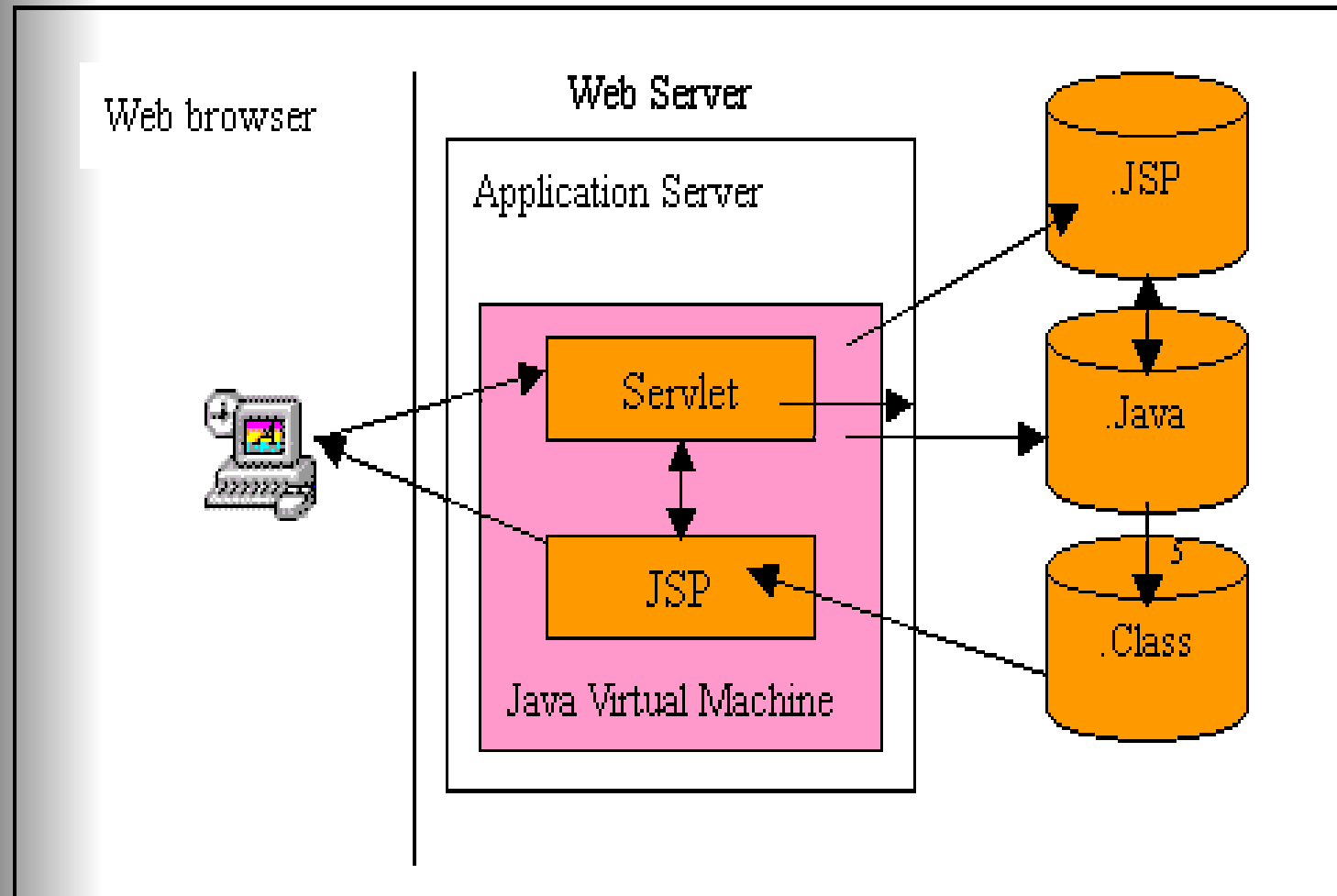


JSP

■ 其特性：

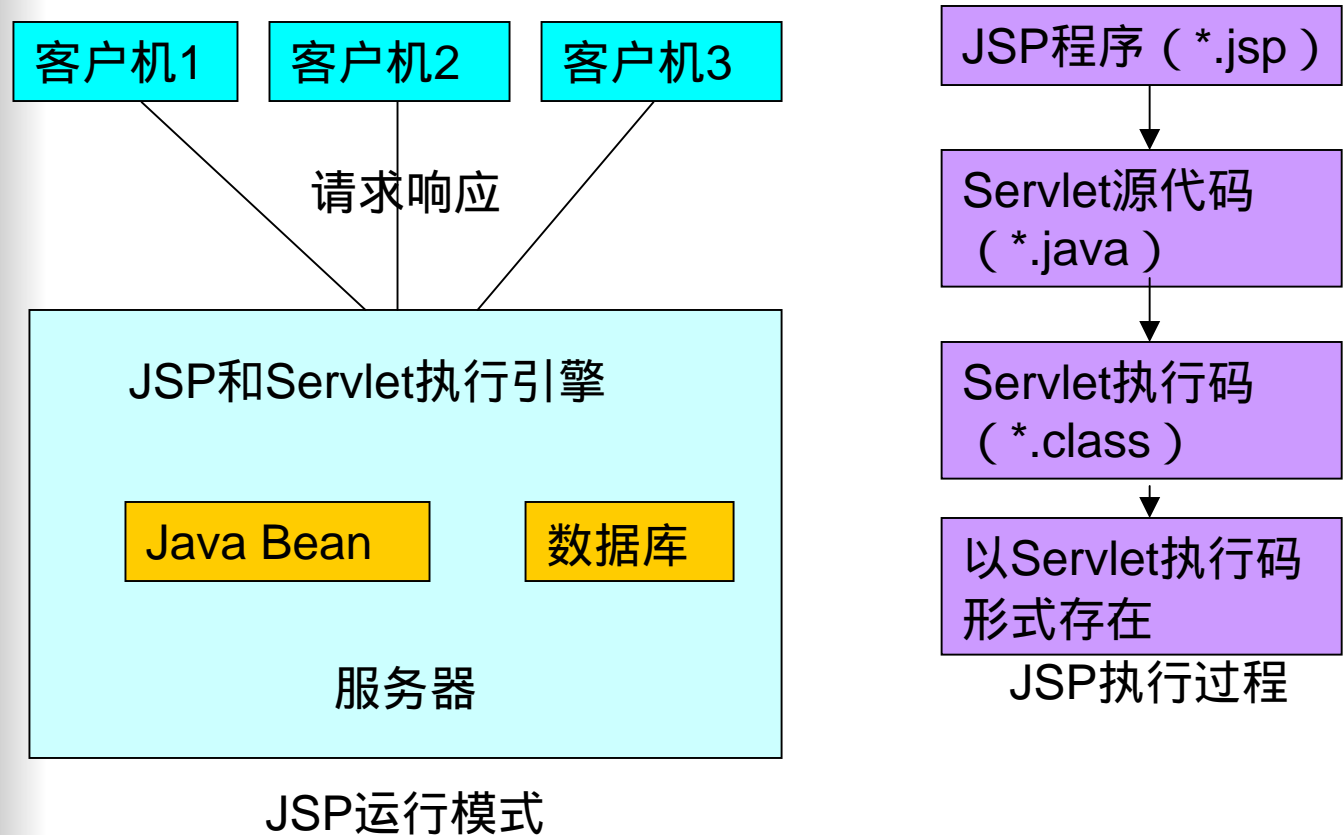
- 内容表达（由JSP负责）与数据生成（由服务器端的JavaBeans负责）的分离
- 更好的MVC(Model /View/Controller)分层体系结构：
 - 业务逻辑封装在组件（如JavaBeans）中
 - 控制逻辑有Servlet处理
 - JSP处理内容表达
 - 可以从JavaBeans中抽取有关内容并将其合并到某个HTML文档中
- 易用性：建立在Java编程和HTML内容表达等标准上
- 可移植性：跨平台
- 基于Java：强类型、OO、模块化、内存管理
- 支持协作开发

Java Server Pages



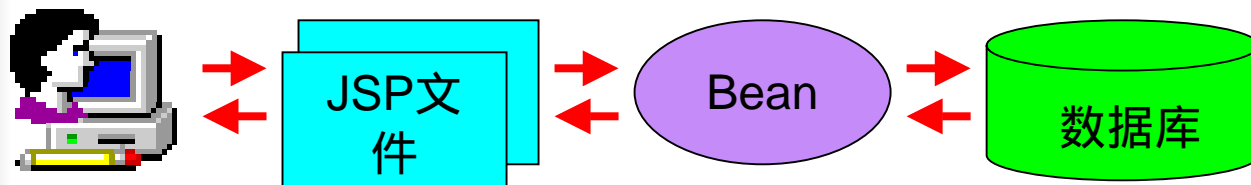
JSP的运行方式

- Web服务器遇到JSP网页请求时，首先执行其中的程序片段，然后将执行结果以HTML格式返回给客户。

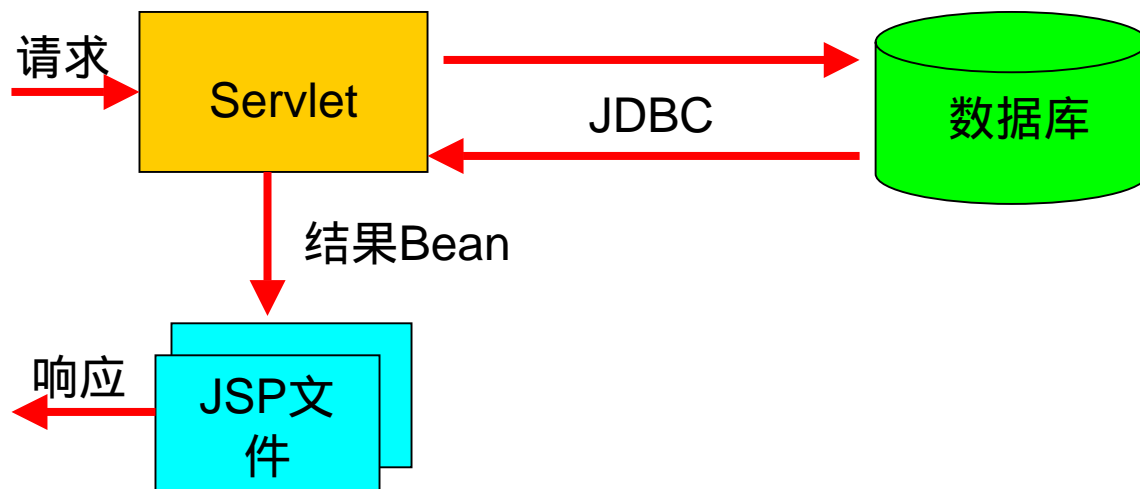


访问JSP文件的两种方式

(1) 浏览器发送JSP文件请求



(2) 发送至Servlet的请求

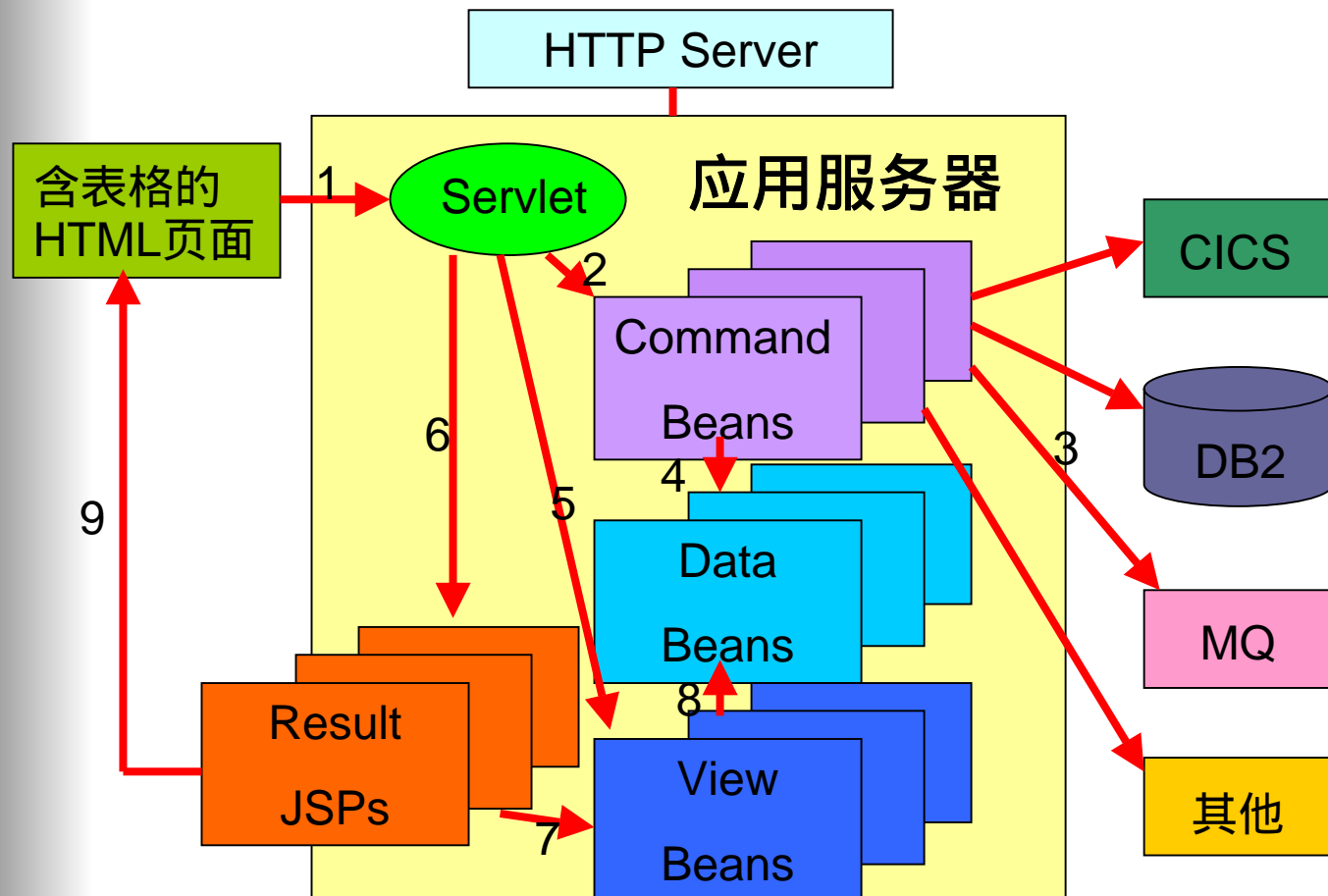




● JSP特点

- 将内容的生成和显示进行分离
- 生成可重用的组件
- 采用标识简化页面开发
- JSP页面就具有Java技术的所有好处，包括健壮的存储管理和安全性
- JSP拥有Java编程语言“一次编写，各处运行”的特点
- 程序执行效率高
- 具有强大的服务器端组件
- 具有强大的数据库支持

基于Servlet、JSP和JavaBeans的Web应用体系结构



- 1: 激活Servlet
- 2: 建立Command Beans
- 3: 嵌入到Command Beans中的业务逻辑交给后台的CICS、DB2、Mq等处理
- 4: Command Beans的结果放入Data Beans
- 5: 初始化View Beans
- 6: 输出处理和格式化
- 7: 通过有关标记声明View Beans，并获取需要显示在输出中的动态数据
- 8: 架起Data Beans与JSPs的桥梁，每个Data Beans有一个或多个View Bean以方便JSP存取数据。
- 9: 为浏览器生成有关输出



ASP

ASP全名Active Server Page，是一个WEB服务器端的开发环境，利用它可以产生和运行动态的、交互的、高性能的（都是微软说的）WEB服务应用程序。ASP只能用在下列Web Server上：

- IIS 3.0 on Windows NT
- Microsoft Peer Web Server v3.0 on NT Workstation
- Microsoft Personal Web Server on Win95
- **ASP技术主要以来微软的技术支持**

ASP和JSP技术比较

比较项目	ASP	JSP技术
Web 服务器	微软的IIS或Personal Web Server	任何Web服务器，包括Apache, Netscape和IIS
OS 平台	微软的 Windows 系统	绝大多数的流行平台，包括 Solaris, Windows, Mac OS, Linux
跨 平 台 访 问	需要第三方ASP的引入产品	支持Web系统中不同系列的计算机群，支持各种工具提供商提供的工具

CGI和JSP技术比较

- JSP隐藏了传统CGI编程中的大部分复杂性，使得服务器端的编程简单化
- JSP执行引擎与HTTP服务器一起常驻内存，每一时刻只有一个JSP执行引擎在内存，执行多个JSP程序，从而执行效率高
- 每执行一个CGI程序时，服务器均要启动一个进程来执行它（即每次系统均要进行加载和清除操作，从而消耗系统资源），执行效率低



■ PHP

PHP是一种服务器内置式的script语言,它的出现使得在UNIX上快速地开发动态web成为现实。

PHP3支持HTTP的身份认证、Cookie、GIF图象创建,它的一个最有代表性的特点是它的数据库层,它使得编写基于数据库的网页变得非常简单。

MVC设计模式

- 即Model-View-Controller设计模式，包含三个部件：
 - 模型
 - 是对有关数据以及与之有关联的事务处理，它是业务逻辑
 - 视图
 - 是针对数据的显示，它是页面的结构
 - 控制器
 - 是针对模型与外部世界的交互，它是交互的控制

MVC设计模式

- 优点：
 - 设计灵活
 - 用户界面的改变不影响应用程序
 - 模型、视图和控制器用不同的技术开发，从而方便项目管理
- 在基于Servlet、JSP和JavaBeans的Web应用中：
 - JSP和HTML提供视图
 - Servlet是控制器
 - Command Beans代表模型
 - Data Beans和View Beans在视图和模型之间提供数据通道
 - Servlet与视图和模型交互，并控制应用执行的流程



问题：

电子商务系统系统的前后端数据如何进行连接？

2.5 前后端数据的连接

■ 两类

- 以Web服务器为中介，将浏览器和数据源连接起来，在服务器端执行对数据的操作；
- 将应用程序和数据库下载到客户端，在客户端执行对数据库的访问。



■ 具体而言（四种）

● 利用CGI访问Web数据库

- 当浏览器发出HTTP请求时，Web服务器执行相应的程序访问站点数据库，并将结果以HTML格式送回浏览器。

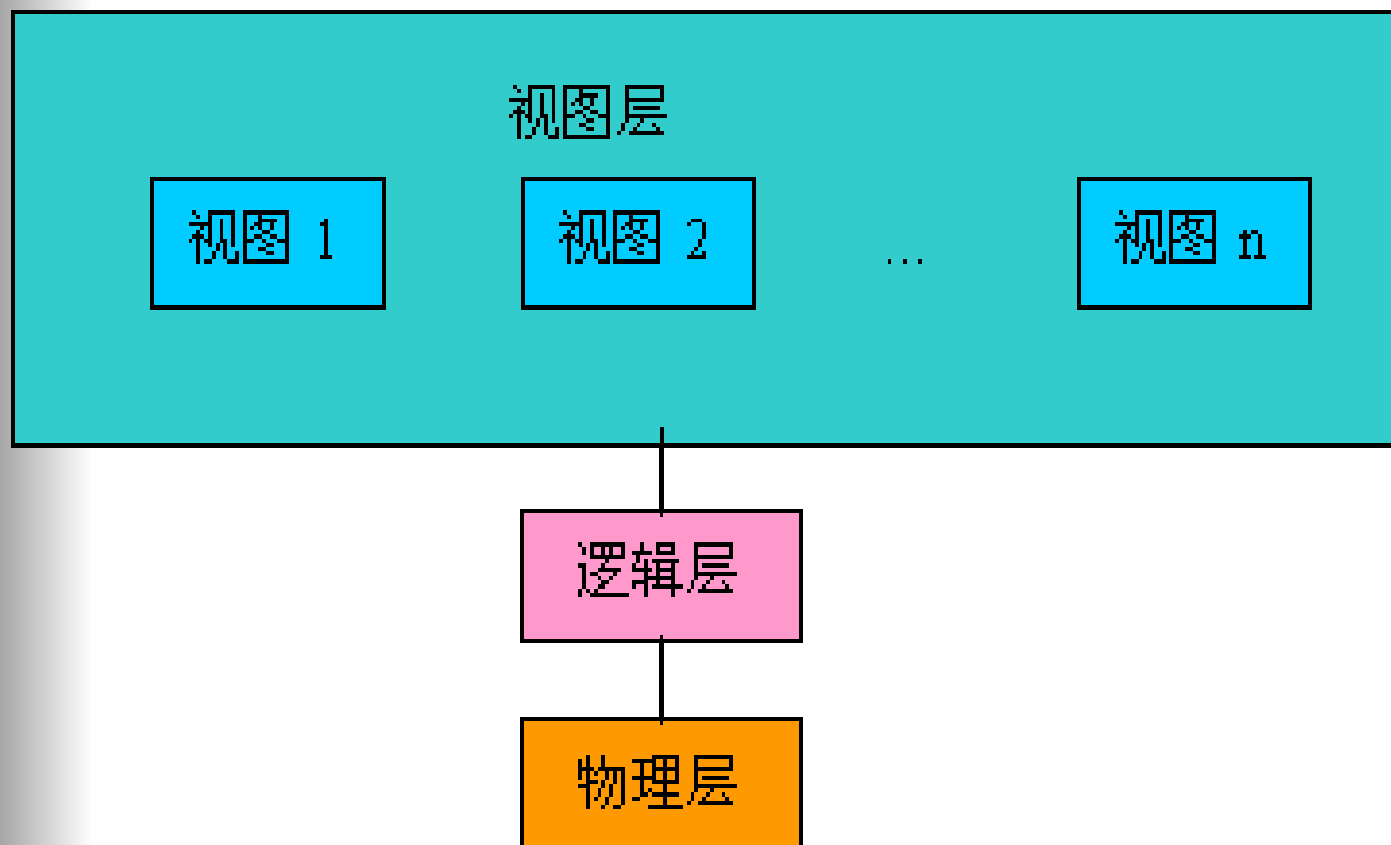
● 利IDC(Internet Database Connector)访问Web数据库

- 对数据库的操作仍然放在Web端，但可以通过SQL语句对数据库进行查询和维护等操作。

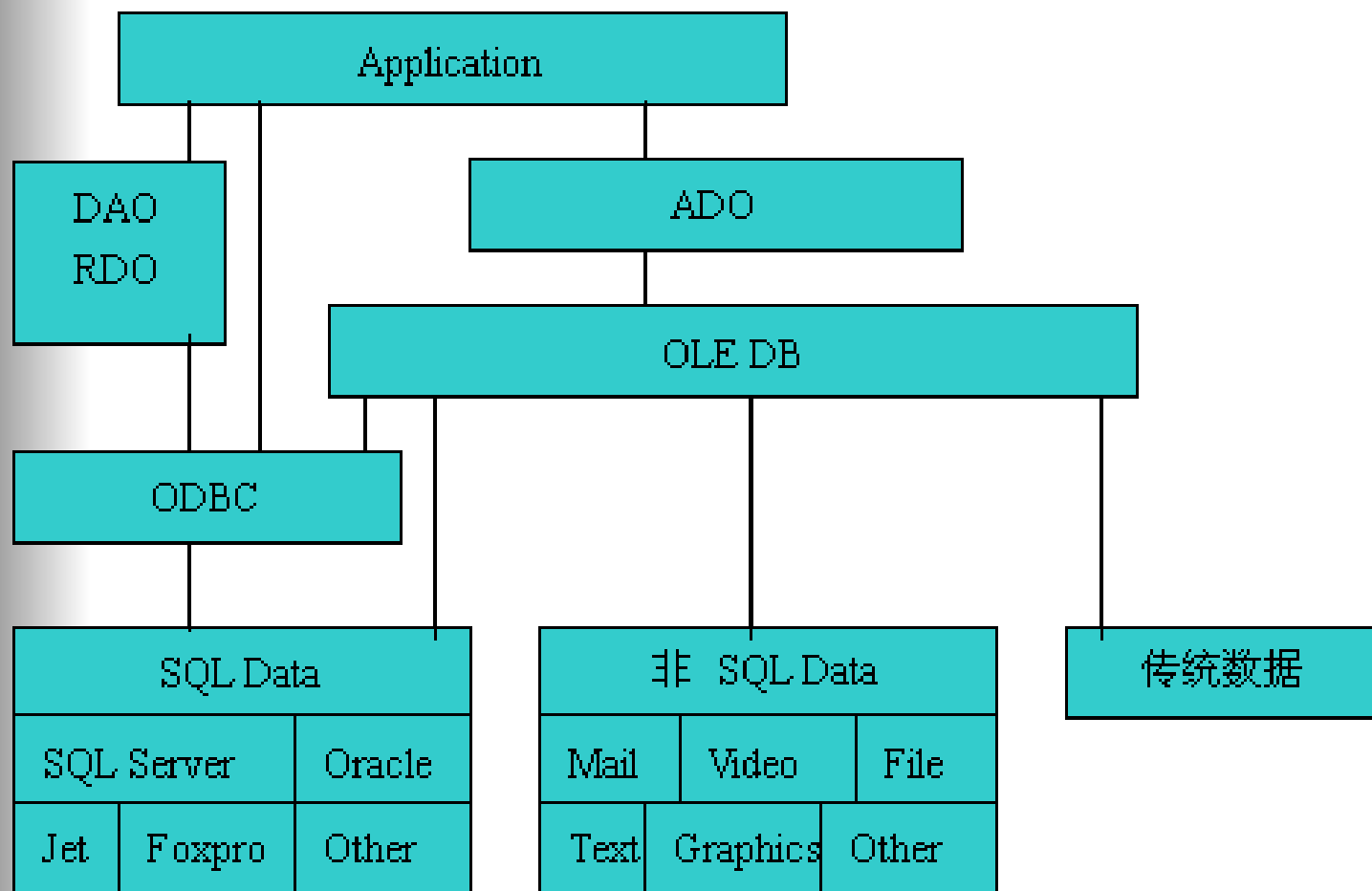


- 利用ADC(Advanced Data Connector)访问Web数据库
 - 在浏览器执行数据查询动作。当浏览器提出对数据库的操作请求时，ADC先将相应的数据库下载到浏览器端，然后在客户端对数据库进行操作（由ADC ActiveX控制完成）。
- 利用ADO(ActiveX Data Objects)和ASP访问Web数据库
 - 可以建立提供数据库信息的Web页面内容，通过在Web页面中执行SQL命令完成对数据库的搜寻和维护等操作；
 - ADO建立在OLE DB之上，可以连接多种支持ODBC的数据库；
 - 在ADO中，可以用VBScript或JavaScript来控制数据库的访问。

数据视图



微软的全面数据访问策略综述





DAO和RDO

数据访问对象（Data Access Objects）：
主要用于访问Access的底层数据库—
JET数据库引擎，将Access和Visual
C++捆绑在一起；

远程数据对象（Remote Data Objects）：
可使用微软的Visual FoxPro和SQL
Server等其它数据库。



OLE DB

一组数据对象，统称为OLE DB（数据库的对象链接与嵌入）。

- OLE DB能访问任何数据源，包括关系型和非关系型数据。
- OLE DB是一个基于C++ API的面向对象说明。
- 面向OLE的应用按对象的方式共享和操纵数据库。



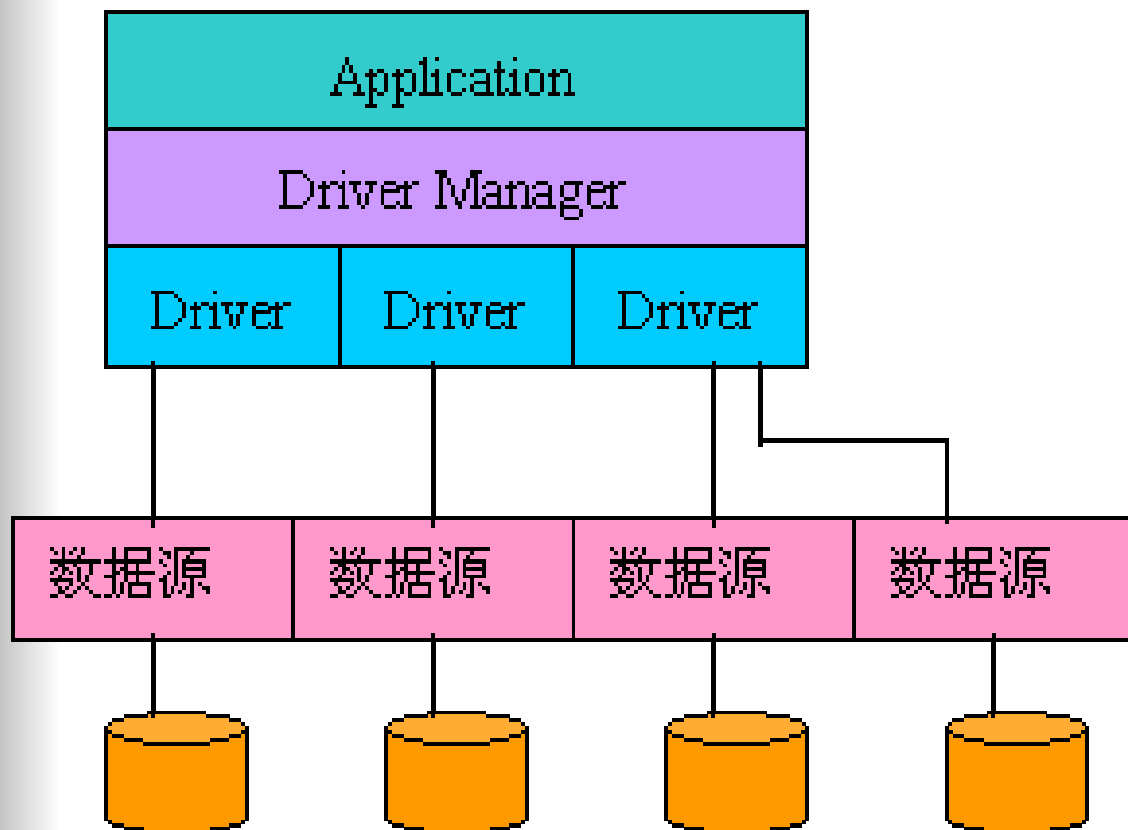
ADO

ADO (ActiveX Data Objects) 是对ASP (Active Server Pages) 的编程扩展，用于数据库连接。ADO的特点：

- 几乎兼容各种数据库系统
- 跨平台和程序语言开发环境
- 可在任何支持COM和OLE的服务器端操作系统上使用

开放数据库互连（ODBC）标准

ODBC技术提供了基于SQL语言访问异构SQL数据库的公共界面，使各应用程序能使用相同的API访问数据库。



数据库



思考题

1. 系统的一般构架经过了哪几个发展阶段？
2. 三层C/S架构比两层C/S架构的优点是什么？
3. 试比较“瘦”客户端与“胖”客户端的差异与优劣有哪些？
4. 客户端与服务器端所采用的技术有哪些？
5. 电子商务系统前后端数据是如何连接的？



OVER Lecture 2



Thank You