



电子商务系统的设计与实现

北京大学计算机系电子商务实验室



- 12.1电子商务的建设方式和技术选择
- 12.2电子商务系统的实现过程
- 12.3客户端的实现方式
- 12.4商务表达平台的实现
- 12.5商务支持平台与应用服务器
- 12.6基础支持平台的实现
- 12.7电子商务系统的集成
- 12.8实施及组织



12.1 电子商务系统的建设方式及技术选择

■ 12.1.1 电子商务系统的建设方式

电子商务系统目前的三类基本建设方式

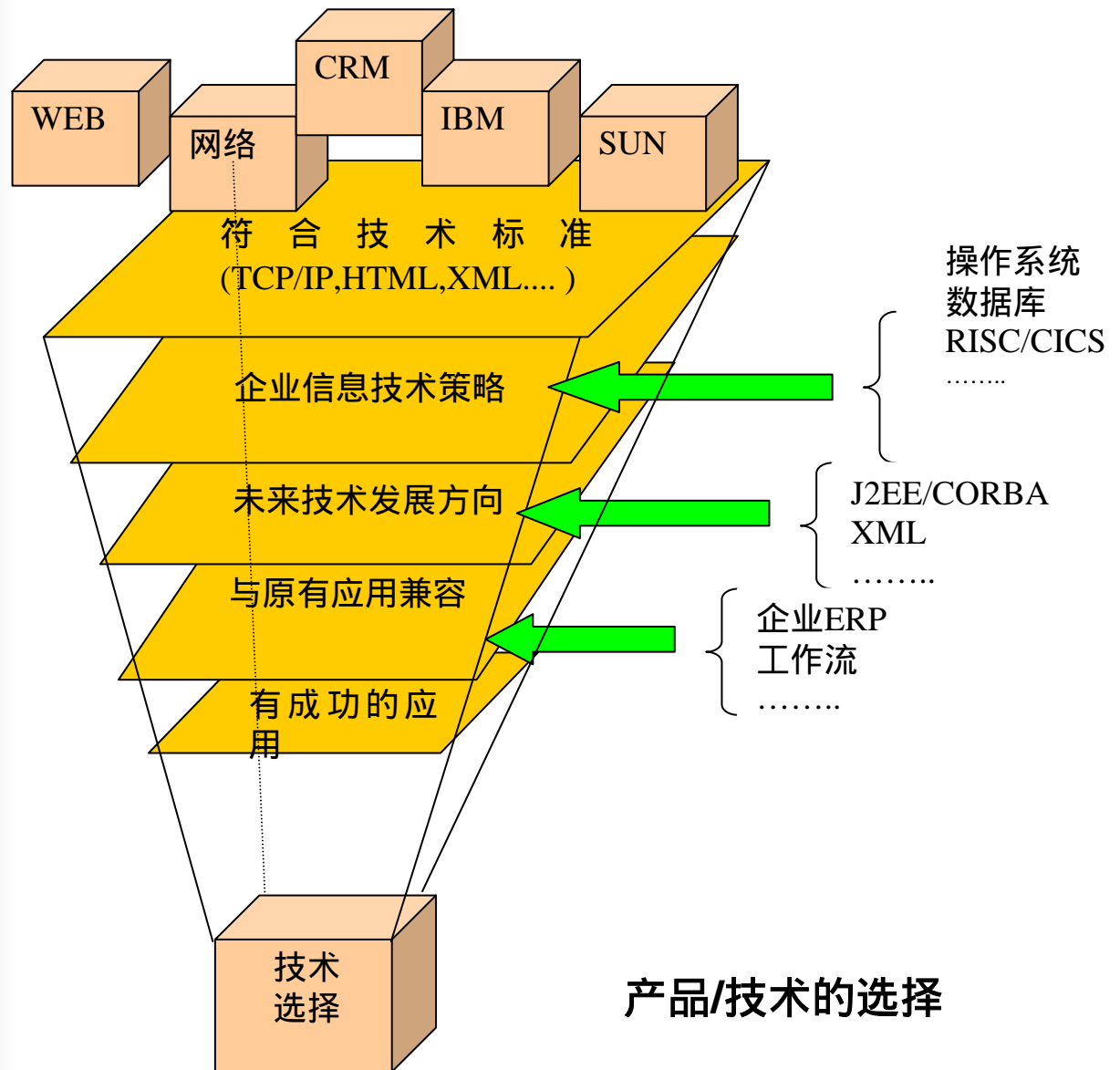
- 企业自己建设
- 委托专业的IT企业集成和开发
- 企业和IT公司合作建设



■ 12.1.2 电子商务系统构造技术的选择依据

● 12.1.2.1 选择技术产品的主要依据

1. 符合各种主流的技术标准
2. 符合企业信息化的整体技术战略
3. 符合未来技术的发展方向
4. 满足开放、可扩充的要求
5. 与现行的应用具有良好的兼容性
6. 具有成功的应用实例





■ 12.1.2.2 各个层次的主要技术标准

● 商务标准及规范

1. 国际数字保证商务总则
(GUIDEC, 1997)
2. EDI标准UN / EDIFACT(1990)
3. Internet开放贸易标准协议
IOTP(1998)
4. 联机交易及在线支付方面的标准(S-
HTTP, SSL, STT, SET)
5. 电子支付的有关标准(JEPT, FSML)
6. 数据自动识别技术标准



■ 技术标准及规范

1. 网络技术标准(ISO/OSI RM, TCP/IP)
2. 计算机系统及设备标准
3. 信息安全标准(DES, RSA, PKI, PGP)
4. 应用软件及开发标准(OSI/GDE, GIF, JPG, CDE)



12.2 电子商务系统的实现过程

■ 12.2.1 应用表达层的实现

- 由多客户端构成
- 确定客户端的类型并选择合适的设备
- 良好的接口



■ 12.2.2 商务逻辑层的实现

- 电子商务系统的核心
- 需要确定选择合适的服务器及其支持软件、开发实现商务逻辑的应用软件
- 包括商务应用、支持平台
 - Web服务器、商务支持软件、集成开发工具、主机、网络
 - 其他软件

■ 12.2.3 数据层的实现

● 数据源

- 系统既有信息系统的数据和企业的数据库
- 企业与合作伙伴间交换的数据
- 企业与银行间交换的数据
- 企业与CA间交换的认证数据
- 企业与其他商务中介交换的数据

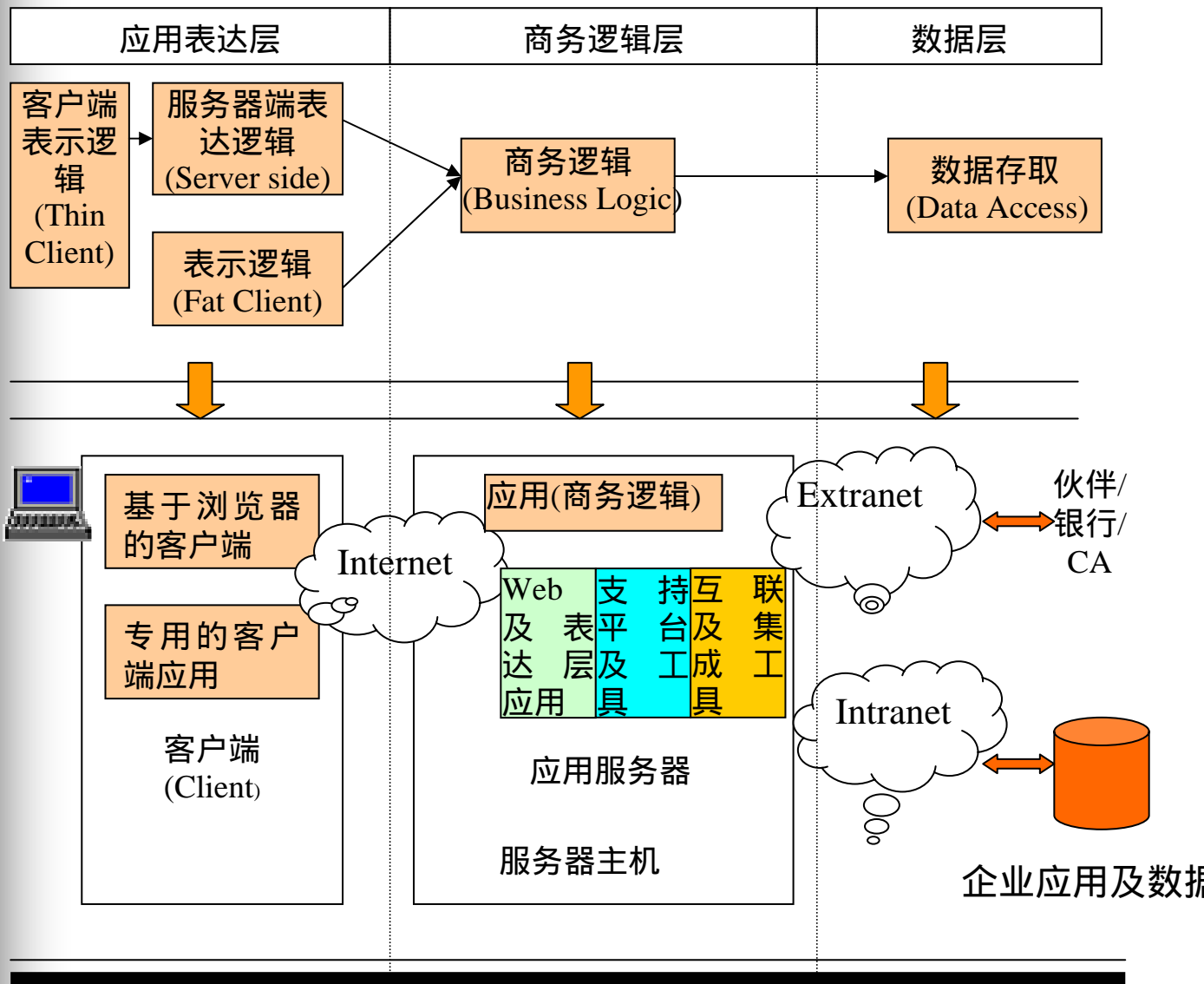
● 任务是

- 实现电子商务系统与企业内外部信息系统的网络互联
- 基于应用服务器平台的商务应用系统与企业内部数据的共享

● 实现的关键

- 技术支持平台的选择与构造
- 应用系统的开发
- 网络系统的集成
- 数据共享及应用的互操作

■ 实施过程的组织



电子商务系统的实现过程示意图



12.3 客户端的实现方式

■ 12.3.1 客户端的分类

- 从设备角度（各类设备）
- 从逻辑结构角度（胖、瘦客户端）
- 从应用角度（通用与专用）

■ 12.3.2 客户端的特点

- 和电子商务系统交互
- GUI
- 支持各种格式化数据表达标准
- 专用客户端



- 12.3.3 客户端设备及要求
 - 支持标准协议
 - Plug-ins
 - Online Update/Upgrade
- 12.3.4 客户端的选择
 - 选择通用浏览器作为客户端
 - 使用专门的客户端应用软件
 - Plug-in
 - 在线下载和升级



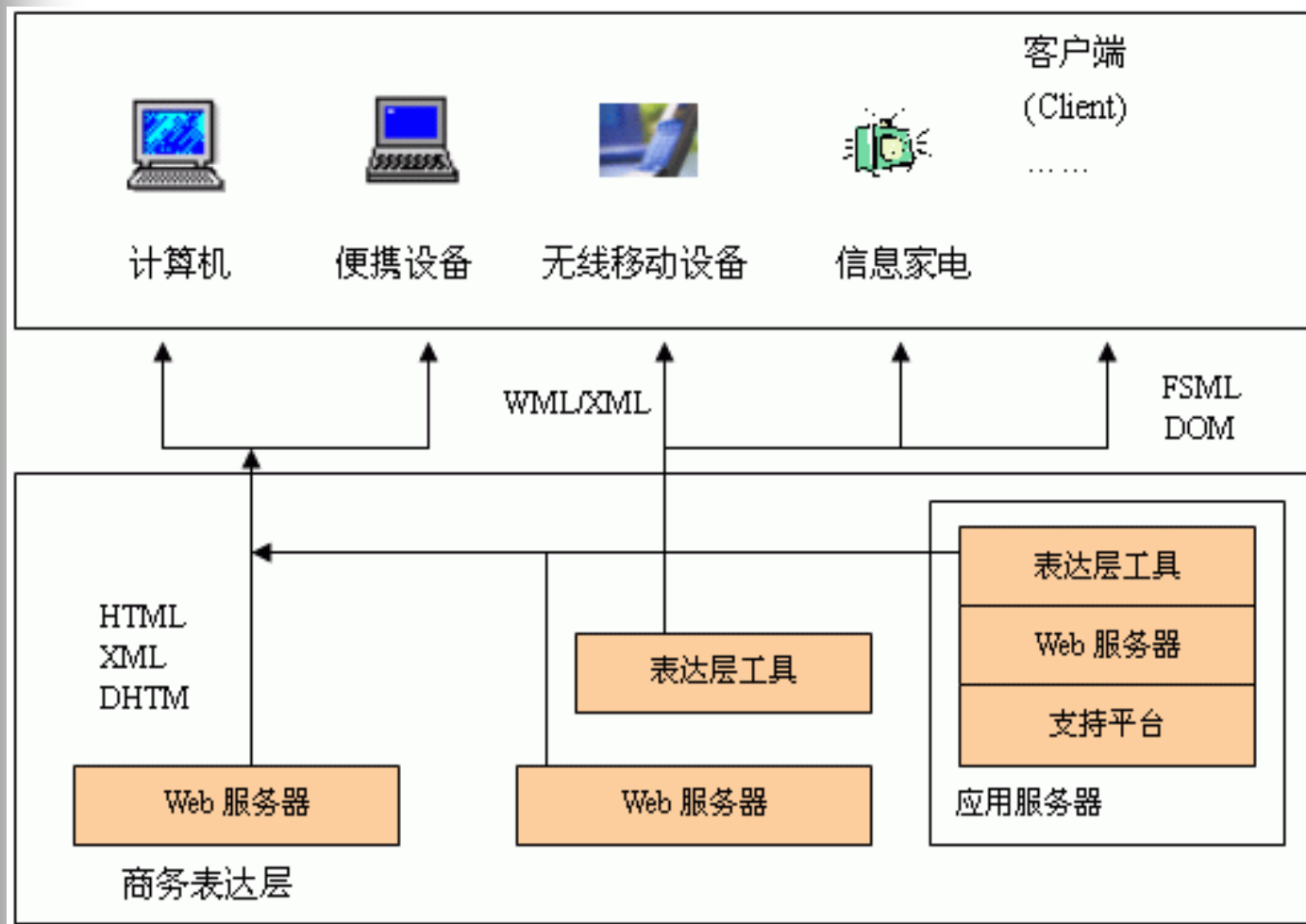
12.4 商务表达平台的实现

■ 12.4.1 商务表达层的主要标准

HTTP、DHTML、WAP、XML、
MIME、其他的一些数据表达协议

■ 12.4.2 商务表达层的实现

- 利用Web，支持以HTML为主的表达形式
- 在Web基础上增加表达工具，扩展Web的既有表达功能
- 利用应用服务器的数据发布功能



客户端及表达层实现方式示意图

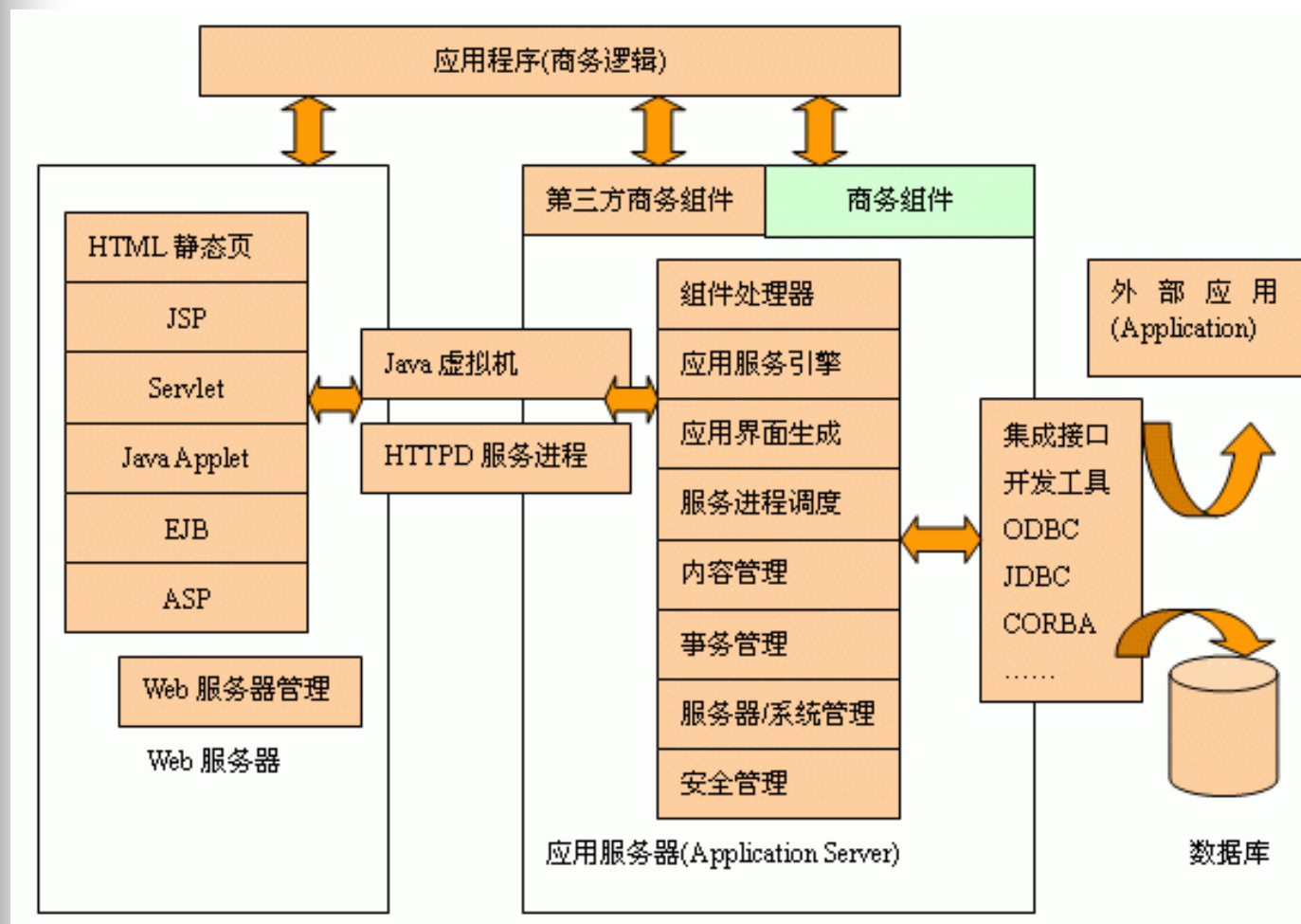


12.5 商务支持平台与应用服务器

- 应用服务器是一个将一些通用的、与商务应用无关的软件集成在一起的软件包，是商务支持层、商务服务层和基础服务层的某些功能的一个集合。

12.5.1 应用服务器基本结构

- 1．增强型Web服务器
- 2．应用服务器



Web服务器与应用服务器

应用服务器包括两部分

- 增强型Web服务器，其功能为：
 - 静态Web页面（HTML和DHTML等）发布
 - 动态页面脚本（JSP、ASP、Servlet、Java Applet等）
 - 用户自定义的MIME类型信息的发布
 - Java虚拟机（JVM）
 - 传统的HTTPD服务
 - 服务器管理功能（页面访问控制、Web访问统计、Web服务器配置等）
 - Web开发接口（NSPI、ISAPI等）



■ 应用服务器 (Application Server)

与Web服务器相互配合共同完成商务逻辑，Web服务器向应用服务器提供用户的请求，并表达处理结果；而应用服务器及其上层的应用负责完成商务逻辑的处理，并反馈处理的结果。两着差别为：

- Web服务器只能将应用程序处理的结果以HTML页面的形式发布给用户
- 应用服务器可为企业级应用提供一种可靠的、高性能的运行环境，而Web服务器无法做到这一点。
- 应用服务器可以提供很多预先安装（或预置）的服务，如搜索引擎、内容管理等。
- 通过预先配置的一些快速构造商务模型组件，企业可以迅速地建立起商务逻辑。



■ 12.5.2 应用服务器功能

- 高性能的应用程序运行环境
 - 内容缓存 (content cache)
 - 数据库连接缓存 (connection pool)
 - 支持多线程执行 (Multi-thread)
 - 负载均衡 (load balancing)
 - 支持分布式联机事务处理 (DOLAP)
- 为应用提供扩充性 (scalability)
 - 服务器集群技术
 - 支持多CPU系统
 - 动态负载均衡



- 会话管理 (Session Management)

- 记录和管理每次人机会话过程

- 支持多种应用编程模式

- 应用表达 : JSP、Servlet、Java应用
 - 商务逻辑 : Java Beans、EJB等
 - 数据库存取 : JDBC、ODBC等
 - 应用集成 : XML、CORBA等

- 目录及内容管理

- 完成用户访问内容的控制、分层数据组织、目录更新及控制服务



■ 商务引擎

- 为商务系统提供业务支持，包括个性化服务，CRM，SCM，eMarketplace等

■ 系统管理

- 性能配置管理
- 存取控制管理
- 系统日志管理

12.5.3 主流应用服务器产品

产品名称

支持

服 务

BEA WebLogic Server HTTP , HTML , XML,WML
JSP , SEVELET
SSL , X . 509 , ACL
J2EE
EJB
CORBA
RMI

静态页间
动态页面
安全
Java2 Enterprise Edition
Enterprise Java Bean
组件标准
远程方法调用

BEA WebLogic Personal Server Personalized Service

基于Java的可靠消息规则
内容管理

BEA WebLogic Commerce Server Busiffess Service

用户和组管理

Collaboration Server (含Session Mngt.
Production Mngt., CRM)

Web页面管理
会话管理、客户管理、
购物指导，赠品管理、
定单管理、产品管理、
库存管理、发票管理、
配送系统管理和客户服务



产品名称

支持

服 务

IBM Websphere Standard

HTTP , HTML , XML , WML

静态页面

WebSphere Advanced Server

JSP , Servlet

动态页面

Websphere Enterprise Server

HTTPS , SSL , X . 509 , ACL

安全

J2EE

Java2 Enterprise Edition

EJB

Enterprise Java Bean

CORBA

组件标准

XA

事务处理

MQSeries

传输中间件

SUN和Netscape联盟的i-Planet

HTTP , HTML , XML,WML

静态页面

JSP , Servlet

动态页面

SSL, X.509, ACL

安全

J2EE

Java2 Enterprise Edition

EJB

Enterprise Java Bean

NetscapeApp Server

应用集成

CORBA

组件标准

Netscape App. Server

应用集成

(NetDynamic)

Calendar Server

消息处理

Directory Server

LDAP

目录管理/内容管理

TradingXpert & ECXpert

销售、采购、库存、

市场等商务活动支持



产品名称

支持

服 务

Microsoft Internet
Information Server(IIS)
SiteSuite

HTTP , HTML , XML,WML

SSL , X.509 , ACL

ASP

COM

ISAPI

Active Server Page
Microsoft 组件模型
服务器开发接口

12.6 基础支持平台的实现

■ 12.6.1 网络环境的构建

分为Intranet、Extranet、Internet三部分

■ 要求：

- 支持网络的互联和应用的互操作
- 能够隔离和控制不同用户对系统的访问，保证网络设备的安全
- 网络环境是可以管理的

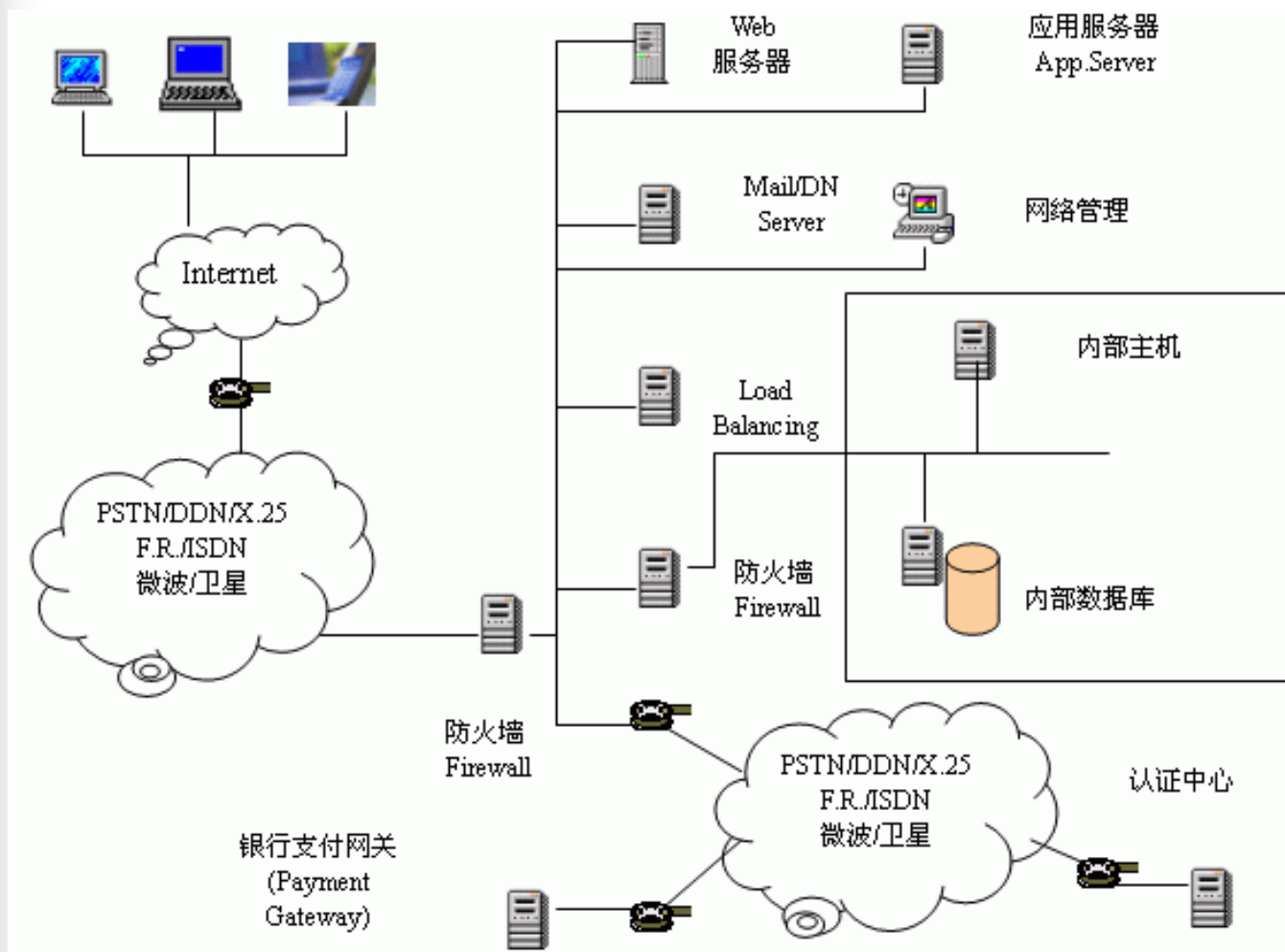


1) Internet部分

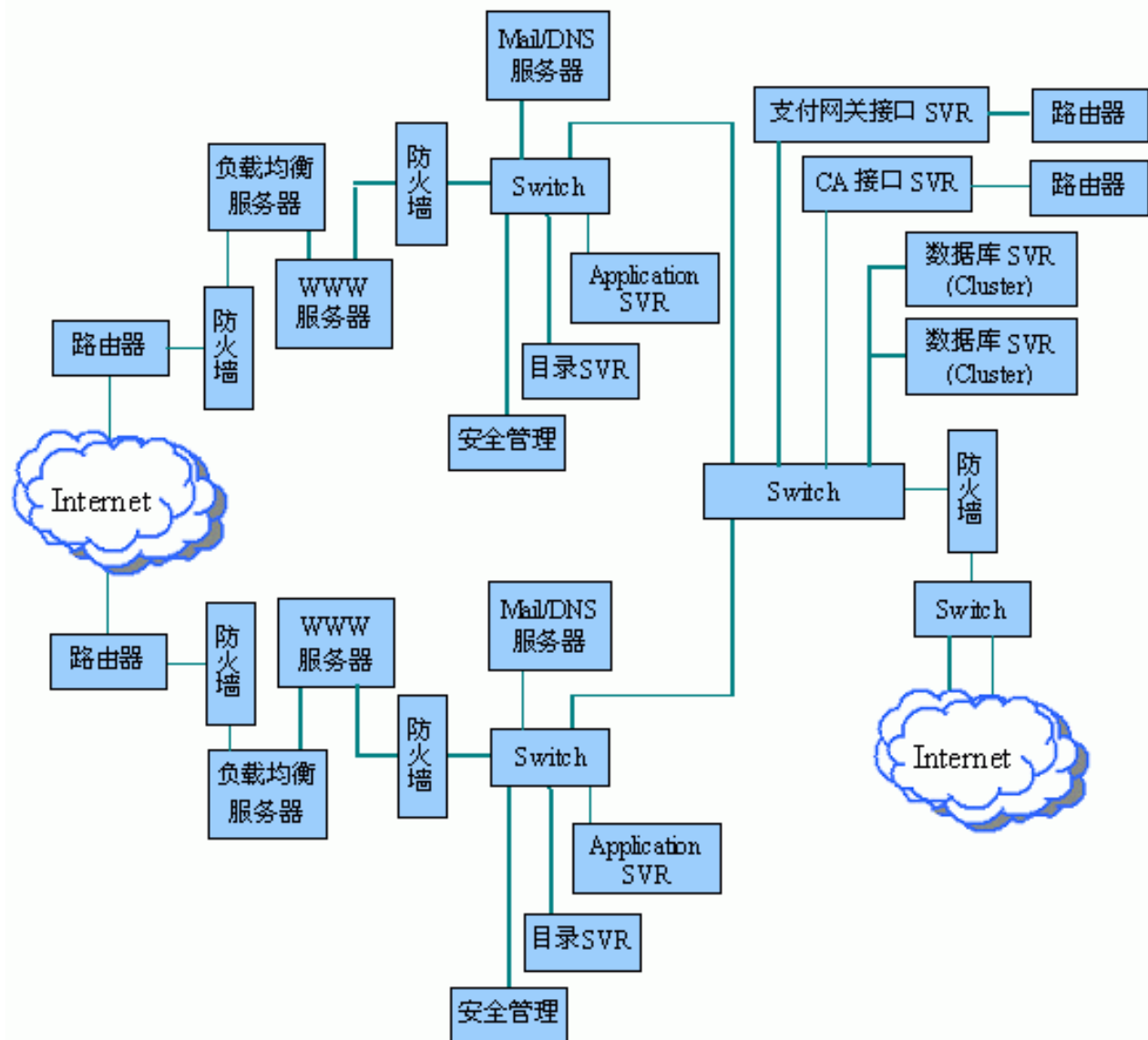
- 涉及接口方式、接口规格的设计
- 电子商务系统与Internet的互联方式：
 - 公共电话网
 - 数据数字网DDN（如ChinaDDN）
 - 分组交换网PSDN（如ChinaPAC）

2) 电子商务系统的局域网部分

- 要求较高的响应速度和带宽
- 一般通过LAN Switch构造
- 必须具备隔离措施，提供安全保证
 - 设置防火墙
 - 网络分为不同等级，通过路由设备隔离



电子商务系统网络结构示意图



一个电子商务系统LAN配置示意图



3) Intranet和Extranet

- 从技术角度看

- (1)低层互联：指通信子网的连通，采用VPN (Virtual Private Network)

- (2)高层互联：指应用程序之间的互通，解决应用的互操作及数据共享问题

- 银行支付网关接口
 - 认证中心接口
 - 企业合作伙伴接口

- 从实施的角度看

- (1)主机托管
 - (2)利用Internet数据中心IDC(Intelligent Data Center)建立电子商务系统的网络环境



■ 12.6.2 服务器主机及操作系统选择

● 1 . 服务器选择

- (1)可靠性高 (auto recover, dynamic reconfiguration, 模块化结构、冗余或热插拔部件, 在线升级等)、(对敏感数据的) 安全性要好
- (2)可扩展性 (增加系统配置或服务器集群)
- (3)网络吞吐量及网络接口能力
- (4)开放的体系结构

● 2 . 操作系统及系统管理软件的选择

- UNIX系统 : IBM AIX, SUN Solaris, HP-UX
- Windows系列 : Windows NT, Windows 2000 Server
- Linux系统 : RedHat, Core Linux, Open Linux

12.7 电子商务系统的集成

■ 电子商务系统的集成层次

1. 数据集成

- 通过消息、网络文件等方式获取外部数据

2. 网络集成

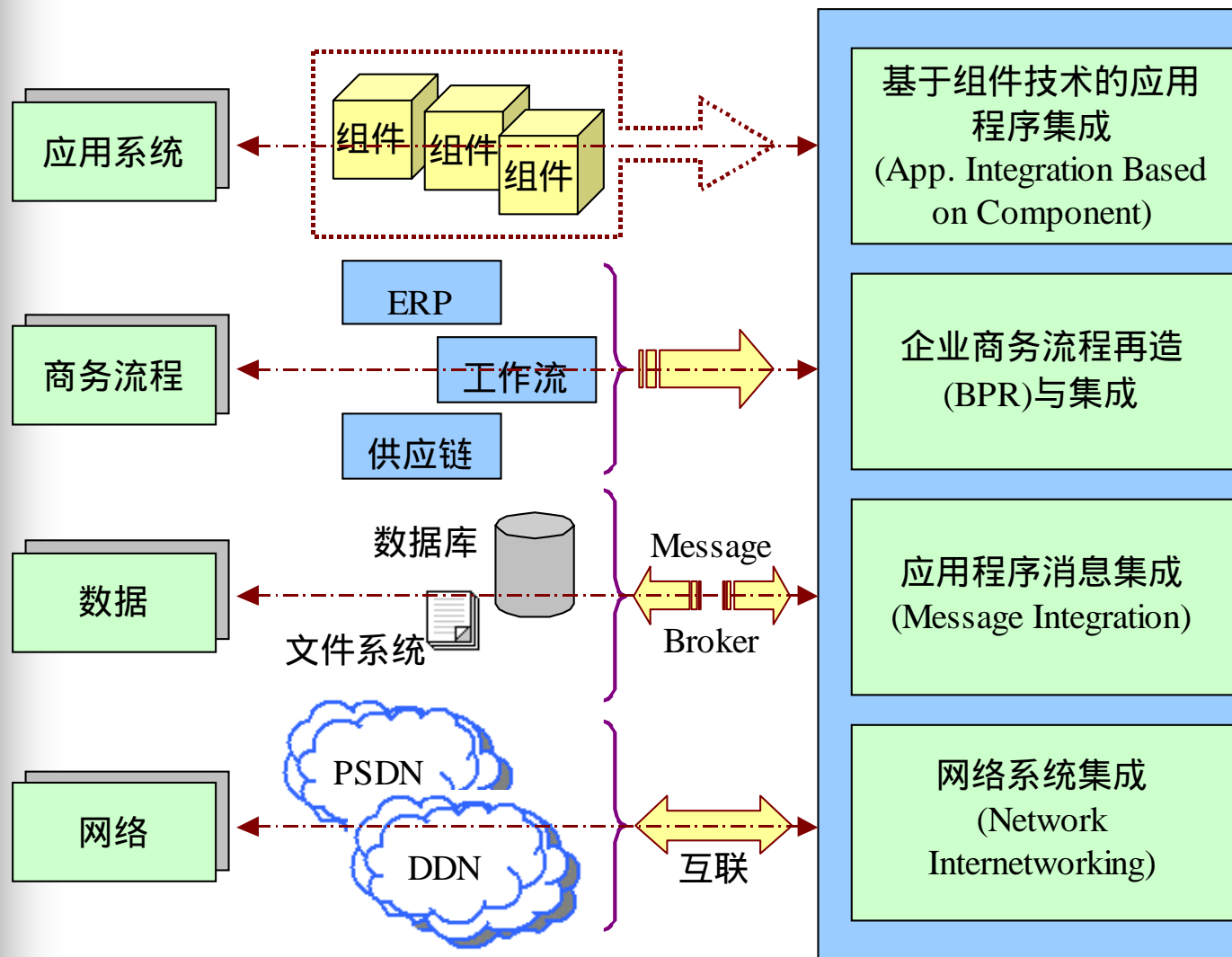
- 集成各种不同形式的数据通信网络，为应用之间的数据交换提供通信环境

3. 应用集成

- 通过组件搭建处理商务逻辑处理应用程序
- 应用程序之间的互操作

4. 企业商务流程集成

- 将企业业务流程统一到一个信息平台上，是一种商务逻辑集成



应用集成示意图

■ 12.7.1 网络集成技术

指将具有不同协议类型的电子商务系统的企业内部网络、电子商务系统的局域网、Internet和企业外部网络连接在一起构成互联的网络。

1. 规划网络集成方案：

- 目标：在调查各互联通信子网的基础上，确定未来网络的体系结构、协议体系、接口方式及互联设备。基本过程：
 - 针对网络集成的对象，调查各互联子网的结构、协议等背景。
 - 确定未来网络的边界以及各互联子网的边界。
 - 分析各子网互联的目的，确定各互联子网的接口层次
 - 根据商务系统的应用处理分布，分析未来网络中各需要互联的通信子网所要承担的网络流量，规划接口带宽要求。
 - 确定未来互联网络的体系结构及协议标准。
 - 确定各个互联接口设备。

2 . 低层网络互联

- 目的：实现通信子网的连通，包括物理层网络的扩充、链路层和网络层互联，其目标是解决通信子网的连接、寻址及路由问题。方式基本上可以分为两部分：
 - 物理层互联，基本上可以通过中继设备实现
 - 链路层和网络层的互联，一般可以通过各种路由设备来实现。

3 . 中、高层应用的互通及互操作

- 主要通过应用网关（ Application Gateway ）实现。
- 应用网关是一个逻辑概念，实际是一组应用程序，主要作用是在不同的高层服务协议之间进行转换。
- 目前网络体系结构中，中高层没有标准的协议，所以中高层集成也没有成熟的标准。因此，这一部分也是网络集成的难点。

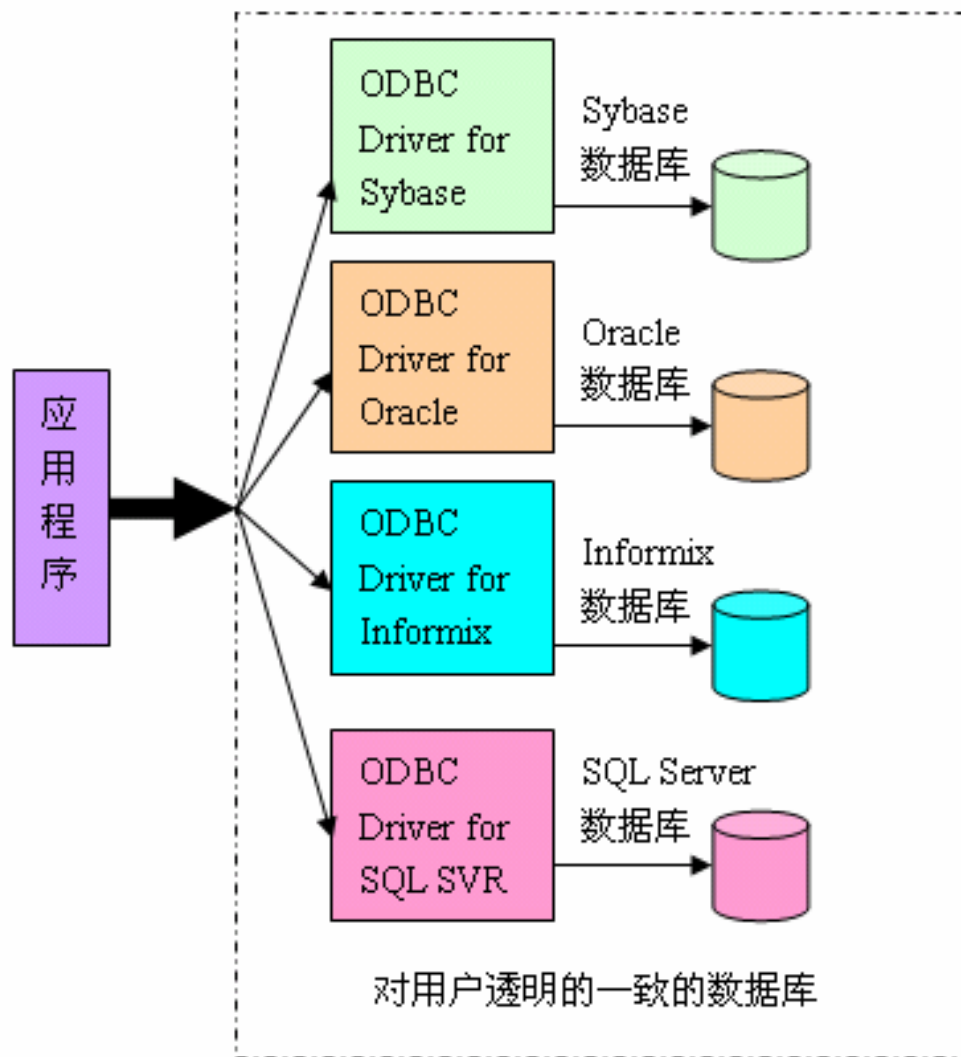


■ 12.7.2 数据集成

目的是使不同系统的、不同形式的数据集合能够统一在一起，为电子商务系统提供支持，实现分布式的数据共享。 其基本方法：

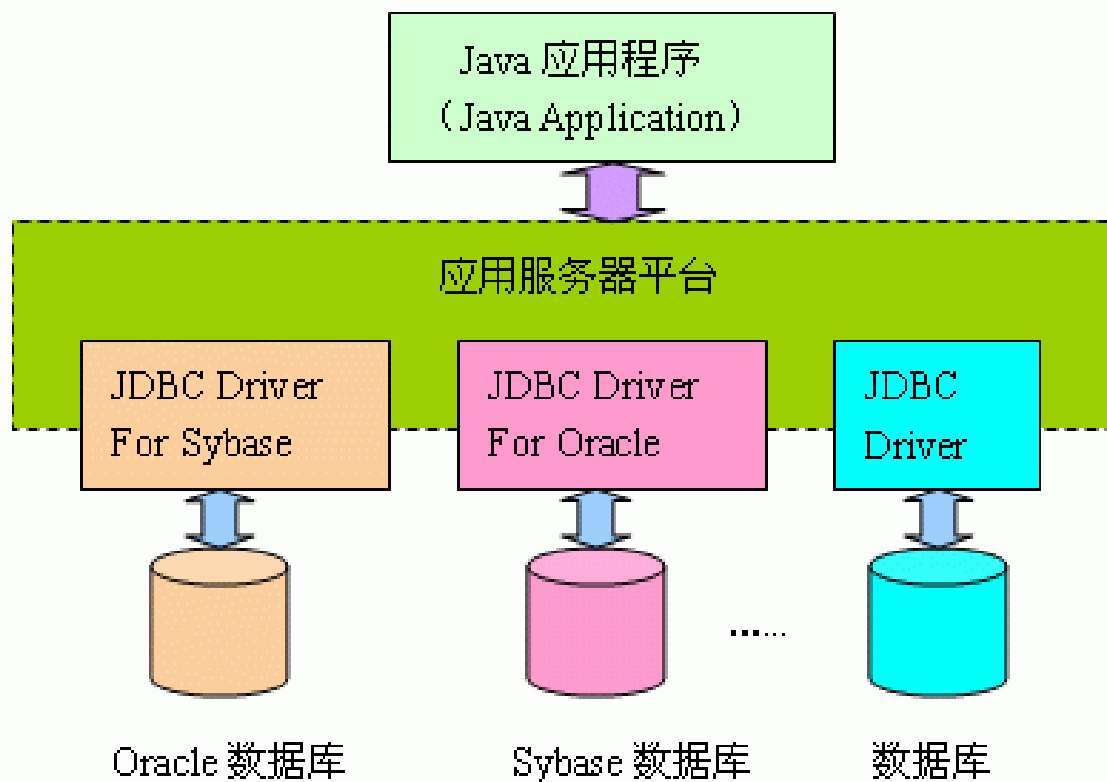
1 . 异构数据库的存取

- SQL(Structured Query Language)
- ODBC(Open Data Base Connectivity)
 - 提供ODBC API和ODBC Driver
- JDBC(Java Data Base Connectivity)
 - 提供JDBC API和JDBC Driver
- UDA (Universal Data Access)
 - 通过ADO(Advanced Data Object)访问数据库，而ADO通过OLE DB访问非关系型数据库或通过ODBC Driver访问不同的数据库。

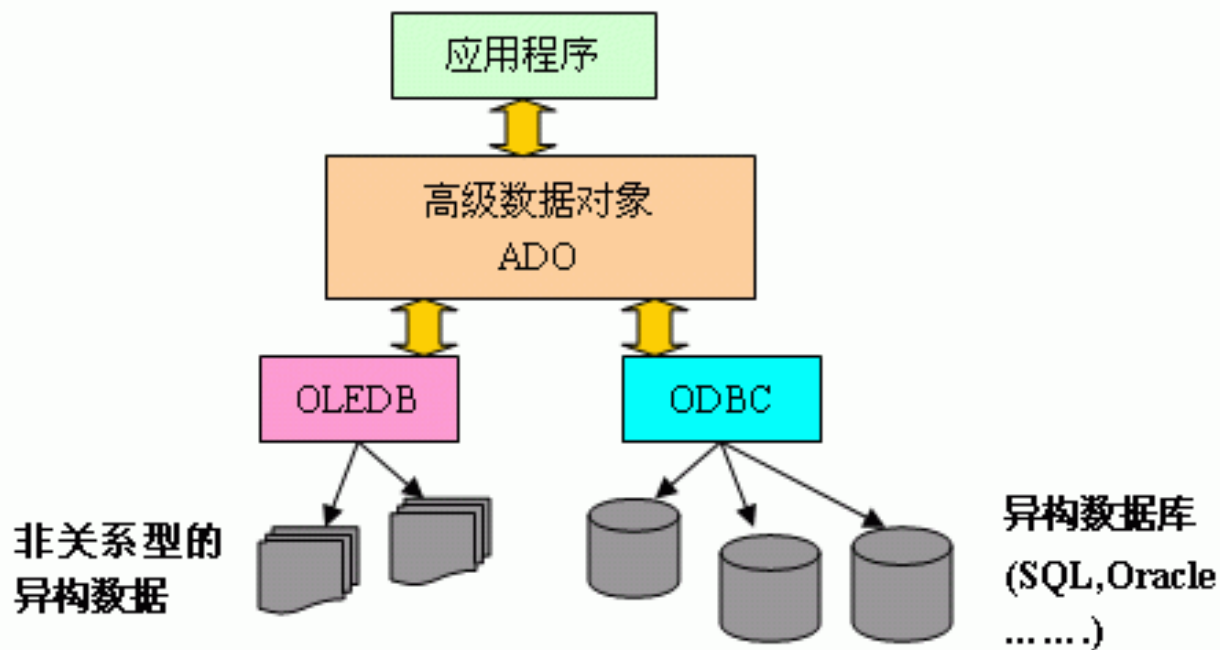


通过ODBC Drive访问异构数据库

北京大学计算机科学技术系(版权所有, 未经许可, 不得转载)



通过应用服务器的JDBC Driver访问不同的数据库



UDA结构示意图



2 . 异构数据的统一管理

指利用统一的操作方式管理不同格式的数据，并且实现不同数据之间的相互转换。标准和产品主要包括：

■ Microsoft UDA

- UDA实际上是将数据及其存储、表达格式和存取方式集成在一起，形成可被统一操作的数据对象（ADO对象）。在对象当中，数据及其操作方法被封装在一起，可以被透明地存取。

■ 扩展标记语言XML

- XML不仅可以定义数据表达的方式，而且可以描述数据的存取方法。




3 . 数据集成的中间件产品

除了上面阐述的标准外，很多中间件产品可以向开发人员提供比较完善的工具和接口，简化了API 接口及配置接口，帮助开发人员完成数据集成。这些中间件产品主要包括以下几类：

■ 基于远程过程调用(RPC)的中间件

- RPC机制使客户可以采用与调用本地进程一样的方式调用远程服务。
- 代表性的有Microsoft的DCOM机制、Inprise公司的Entera、NoblenetRPC。



■ 基于应用消息队列(message)的中间件

- 是在IPC (Internal Process Communication) 的消息队列基础上的经过扩充而实现的，它保证分布式应用可以在脱机情况下实现可靠的数据传输。
- 这类产品的代表是IBM的MQ Series，此外还包括BEA TUXEDO中的MessageQ、Microsoft的Message Queue，国产化产品的代表是东方通的TongLINK / Q。

■ 交易中间件

- 也将此类中间件称为交易处理监视器。它本质上是对联机事务处理提供服务的软件工具。
- 典型代表是BEA公司的TUXEDO、IBM的CICS等。
- 比较适合于构造要求在Internet环境中进行并发事务处理（例如订票、证券委托等）的电子商务系统。



■ 12.7.3 基于组件的应用集成

- CORBA(Common Object Request Broker Architecture), 具有两层含义：
 - 指对象管理组织 OMG (Object Management Group) 提出的分布式对象体系结构标准(或称规格)
 - 按这种体系结构开发的中间件产品
 - 其主要特征是：
 - 系统间实时互操作
 - 对象的封装性
 - 适合于分层结构的应用集成



■ EJB和J2EE

1 . EJB和J2EE

- EJB：定义了一个集成的编程模型和标准的API接口，是实现服务器端应用服务的组件。
- J2EE：J2EE（Java2 Enterprise Edition）定义了一个一致的环境，以支持企业级别应用的集成，确保应用的可移植性。J2EE被定义成为一个多层次的服务、开发平台，总共包括4个组成部分，其核心是J2EE应用开发模型和J2EE平台：
 - J2EE应用编程模型（J2EE Application Programming Model）
 - J2EE平台（J2EE Platform）
 - J2EE兼容性测试工具（J2EE Compatibility Test Suite）
 - J2EE参考实施（J2EE Reference Implementation）

2 . J2EE应用编程模型


在J2EE的应用编程模型中，商务逻辑处理被分成三个层次：

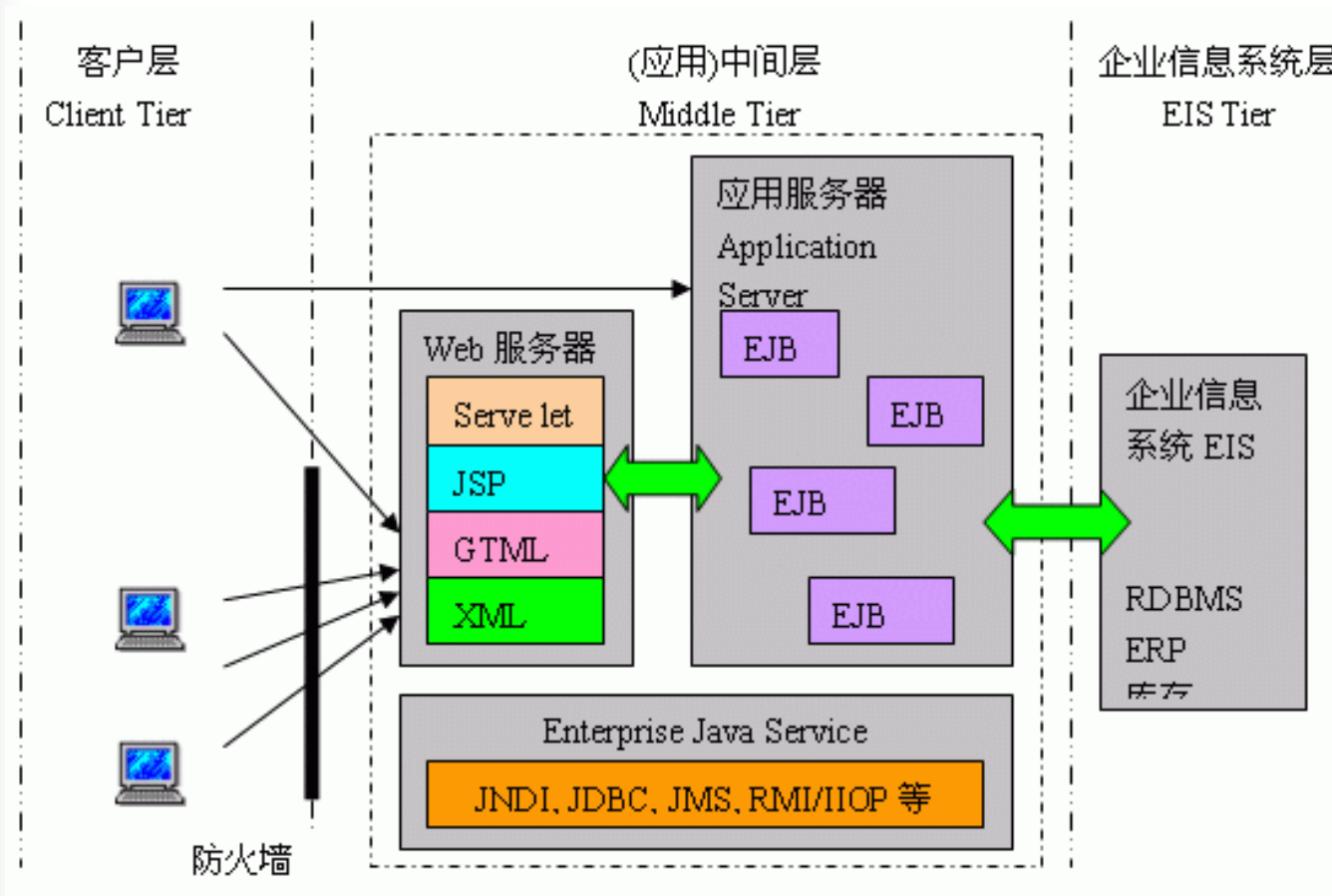
- 客户层 (Client Tier)
 - 支持不同的客户端，包括基于浏览器的瘦客户端及其他客户端
- 中间层 (Middle Tier)
 - 能够完成企业服务的存取
- 企业信息系统层 (EIS Tier)
 - 负责存储企业内部的关键商务数据

在J2EE模型当中，应用服务被分为两部分：

- 商务及表达逻辑，由开发人员实现
- 标准的系统服务，由J2EE平台提供

在J2EE模型当中，中间层的商务功能通过一系列的EJB组件实现。中间层使用JSP实现商务逻辑处理结果的动态发布，构成动态的HTML页面，中间层也可以使用Servlet实现更为灵活一些的动态页面。

- 
- 在上述模型中，中间层可以通过以下方式访问企业信息系统层(EIS Tier)中的信息资源：
 - JDBC：数据库访问接口API
 - Java名字及目录接口JNDI (Java Naming and Directory Interface)
 - Java消息服务JMS (Java Message Service)
 - Java Mail
 - Java IDL (Interface Definition Language)




J2EE应用编程模型



■ 3 . J2EE平台

是一个运行J2EE应用(EJB、JSP、Servlets等)的标准环境。它包括以下几个部分：

- J2EE(应用)分布规范(J2EE deployment specification)
- J2EE平台的Java技术标准
- J2EE平台的IETF标准
- J2EE平台的CORBA标准



Microsoft DCOM及Windows DNA

- DCOM(Distributed Common Object Modal)是微软公司以其公共对象模型COM(Common Object Modal)为基础提出的分布式应用集成框架。
- DNA(Windows Distributed InterNet Applications Architecture)结构的基本出发点是为了改善传统的客户机/服务器两层结构愈发表现出的明显的局限性，以适应更快更复杂的事务处理任务和快速开发的需要。



- 在Windows DNA结构中，分布式应用系统由表示层(Presentation)、事务逻辑层(Business Logic)和数据服务层(Data)三个组成部分：
 - 表示层：用户的界面部分
 - 商务逻辑层：负责处理表示层的应用请求，完成商务逻辑的计算任务，并将处理结果返回给用户
 - 数据服务层：为应用提供数据来源



Windows DNA的优势：

- 应用开发分工明确：

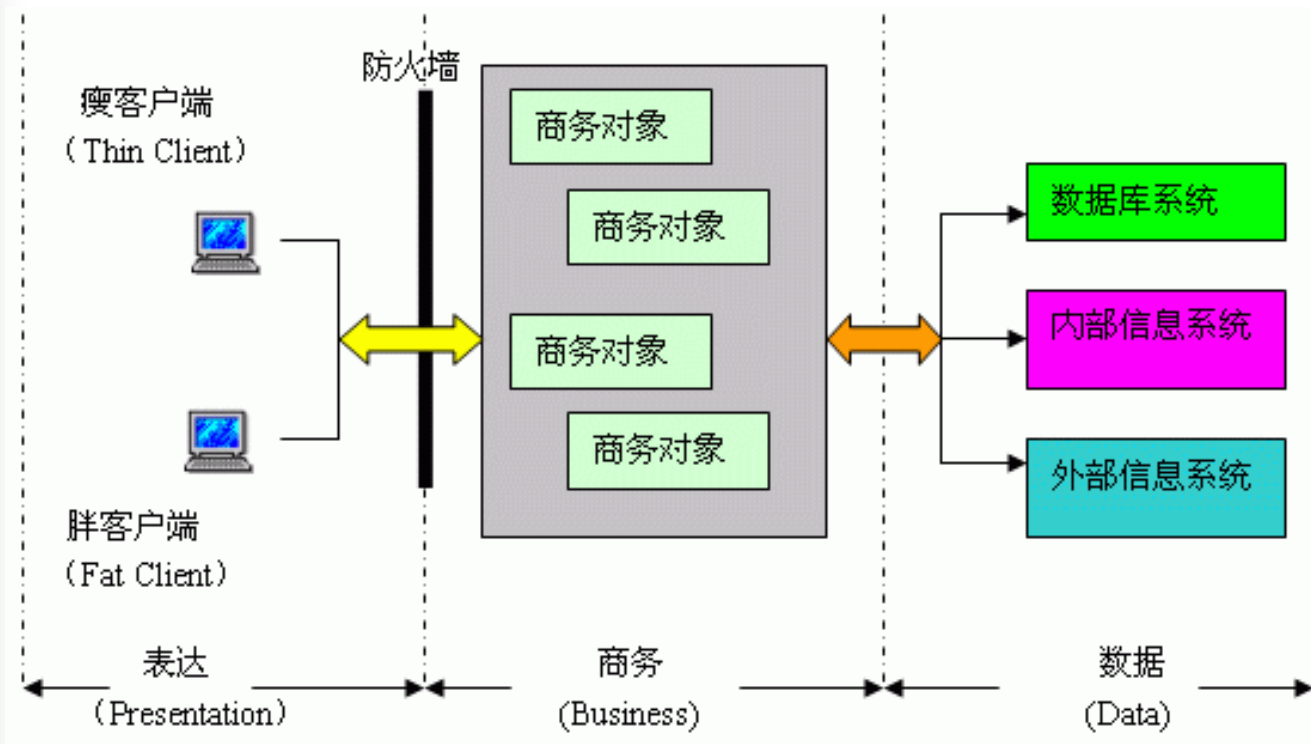
- 专注于事务逻辑层COM组件的开发和测试工作人员；
- 根据商务逻辑的需要选择和使用COM组件，而不需要了解组件功能实现的内部细节的人员；

- 将应用逻辑组件集中置于中间层：

- 组件对象COM的可重用性减少了应用系统整体的管理和维护费用
- 商务逻辑改变时，不必改变整个页面源代码，只需调整或替换中间层相应的COM组件，即可灵活适应商务逻辑的变化。



- 提高系统的运营效率 and 安全性：
 - 在中间层，IIS负责应用逻辑层Web页面的管理，MTS负责应用逻辑层COM组件的管理。MTS在多线程的支持下工作，实现对COM组件的分布式连接管理、线程自动管理及高性能事务处理的监视。
 - 应用程序使用组件可以共享与数据库的连接，使数据库不再和每个活动客户保持一个连接，而是若干个客户通过共享组件和数据库连接，降低了数据库的负担，提高了系统性能。
 - 客户通过组件访问数据库时，MTS的安全管理可以按权限将特定组件授给不同的用户组，使商务活动的安全性和系统结构有机地结合在一起。



Microsoft Windows DNA结构示意图



12.8 实施及组织

■ 12.8.1 电子商务系统开发集成的任务

- 特殊性：

- 开发工作可以分解成为客户端和服务器的应用开发两部分。
- 应用程序既包括编译型的程序，也包括解释型的脚本程序(例如Java Applet、JavaScript、VBScript等)。
- 借助组件技术搭建应用日益成为未来电子商务应用开发与集成的主流方向。



● 应用系统设计开发

■ 应用系统设计

- 包括应用系统的体系结构、数据流程、外部接口、数据结构、组件对象分析及定义等。其目标是清晰地描述应用系统的结构，以便为项目开发定义可分解的任务。

■ 服务器端组件、应用编程

- 按商务逻辑的功能进行划分。
- 开发组织方式是根据人员的知识及B / S结构的特点，将开发组织分成前、后台两个开发小组。

■ 客户端应用或脚本编写

- 针对浏览器编写的瘦客户应用，可以利用各种解释性语言编写，例如ActiveX、Java Applet等。
- 对需要进行数据处理的所谓胖客户程序用传统的C / C++等高级语言编写。



- 内容设计与编辑
 - 站点设计与策划
 - 页面编程
 - 内容创建或者信息采集编
- 软、硬件系统集成



■ 12.8.2 开发集成人员

- 项目管理人员：负责系统开发的组织协调
- 站点设计师：负责系统设计、策划
- 应用工程师：负责应用系统前、后台应用的编写、组件开发
- 系统工程师：负责系统软件(操作系统、数据库等)维护
- 网络工程师：负责网络集成
- 硬件工程师：负责硬件设备安装、调试与维护
- 美术创意人员：负责站点艺术包装，辅助页面开发工程师开发
- 页面开发工程师：负责HTML页面编写和页面内嵌脚本集成
- 内容创建或信息采编人员：数据采集、整理和加工



■ 12.8.3 项目管理

● 目标

- 付出较低的开发成本
- 应用系统的品质能够得到可靠保证
- 开发的应用易于移植
- 开发过程可控制、可管理
- 最终项目进度可保证
- 项目投产后易于维护



● 项目管理组织

- (1)项目建设领导小组
- (2)项目顾问与支持小组
- (3)项目负责人
- (4)项目实施小组
- (5)项目品质保证小组
- (6)项目开发小组



END



Thank You