

课程内容

第一章 软件构件与中间件

➤ 分布式软件的基本概念

- 三层结构的特点及相对于两层结构的优势

➤ 软件构件的基本概念

➤ 中间件的基本概念

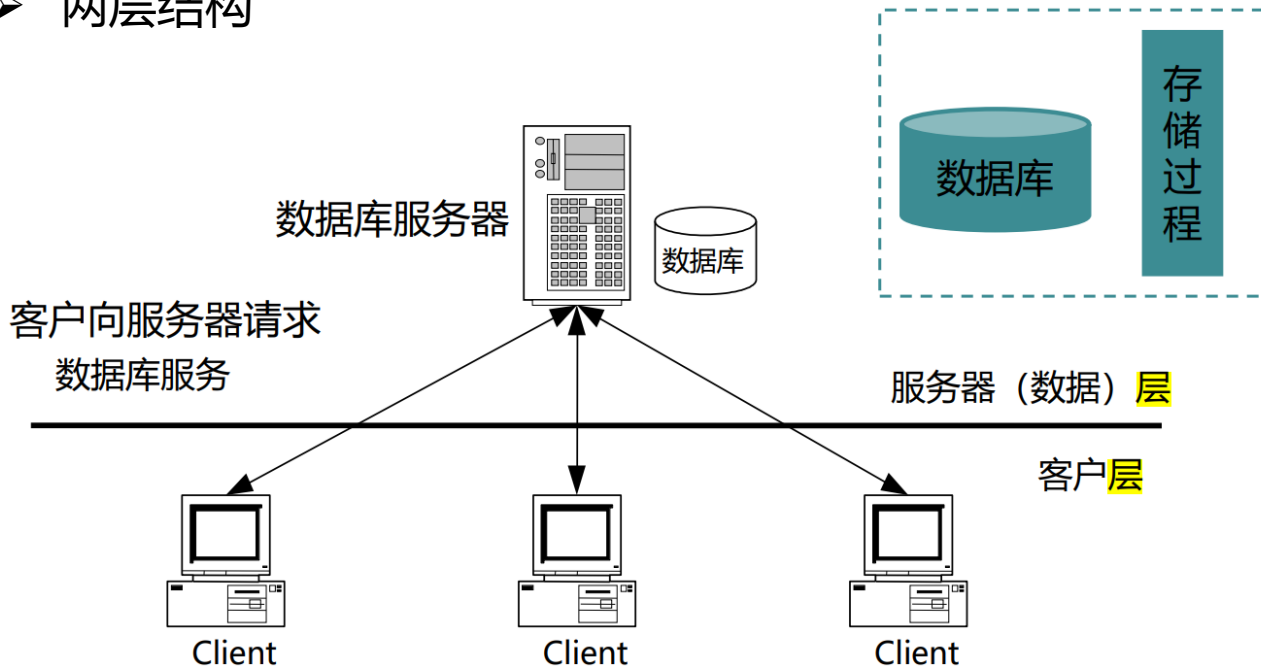
- 典型集成中间件为软件开发提供的三种基本支撑

➤ 互操的基本原理与实例

- 桩/框架结构(Stub/Skeleton)

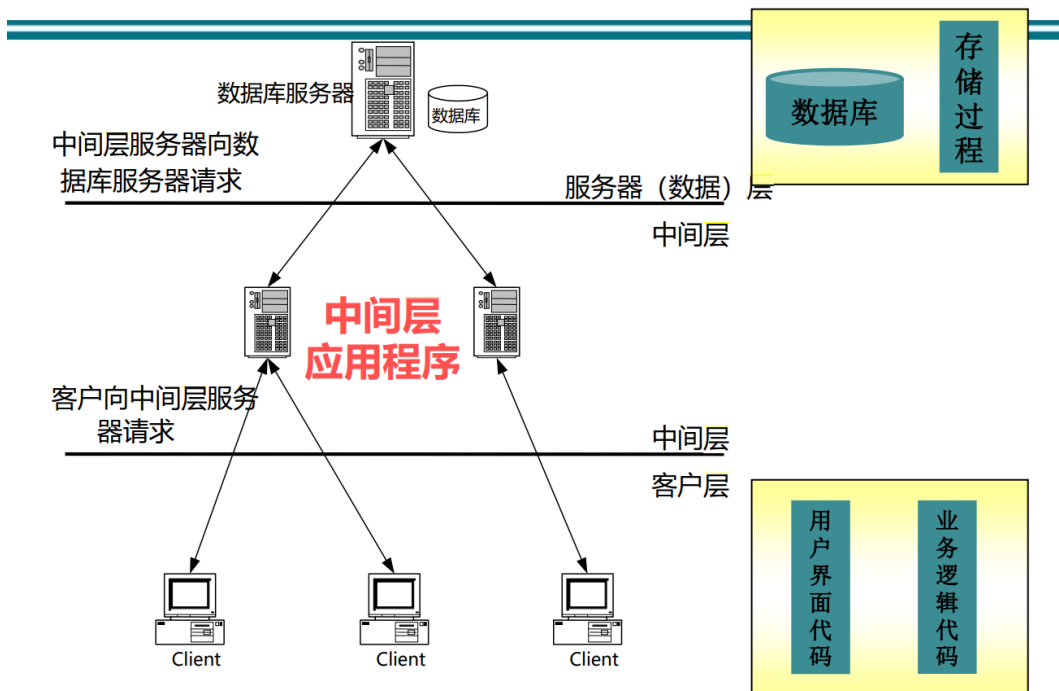
第一章 软件构件与中间件

➤ 两层结构



第一章 软件构件与中间件

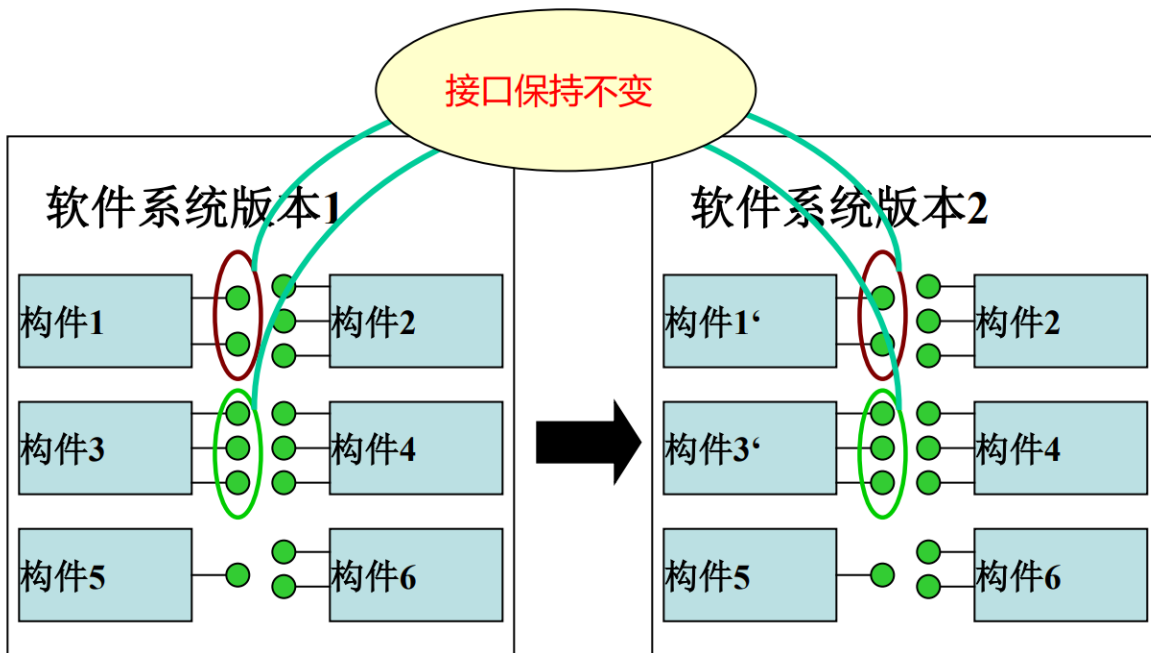
➤ 三层结构



第一章 软件构件与中间件

➤ 构件与软件构件

- **构件**指系统中可以明确辨识的构成成分,
- 而**软件构件**指软件系统中具有一定意义的、相对独立的构成成是可以被重用的软件实体



第一章 软件构件与中间件

➤ 中间件

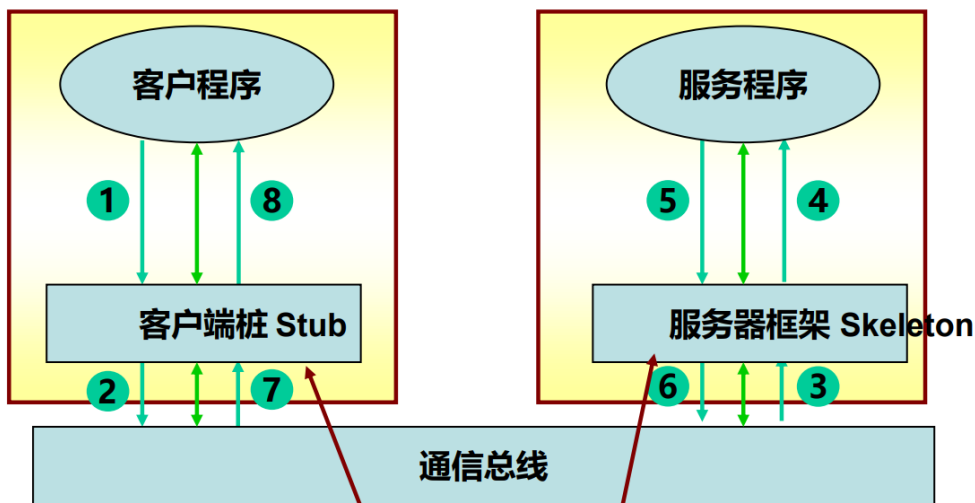
- 在操作系统与应用系统之间的一层软件，为分布式应用的开发、部署、运行与管理提供支持

➤ 中间件提供的主要公共服务

- 包括命名服务
- 事务服务
- 安全服务
- 持久性服务、消息服务、分布式垃圾回收服务、资源管理

第一章 软件构件与中间件

➤ Stub/Skeleton结构



可以自动生成

第一章 软件构件与中间件

➤ 中间件特性

- 平台性:中间件是必须独立存在,是运行时刻的系统软件
- 支撑性:通过抽象层度更高的构件模型,实现具备更高结构独立性、内容自包含性和业务完整性的可复用构件
- 复用性:

复用对象	复用范围
子程序	一个可执行程序内复用, 静态开发期复用
构件(DLL,COM等)	系统内复用, 动态运行期复用
企业对象组件(DCOM,.NET,EJB等)	企业网络内复用, 不同系统之间复用
服务 (如Web Service,SCA/SDO, 微服务)	不同企业之间, 动态可配置

- 解耦性
- 互操作性

第一章 软件构件与中间件

➤ 中间件类型

- 1.数据访问中间件
- 2.远程过程调用中间件
- 3.消息中间件
- 4.事务(交易)中间件
- 5.构件中间件
- 6.集成中间件

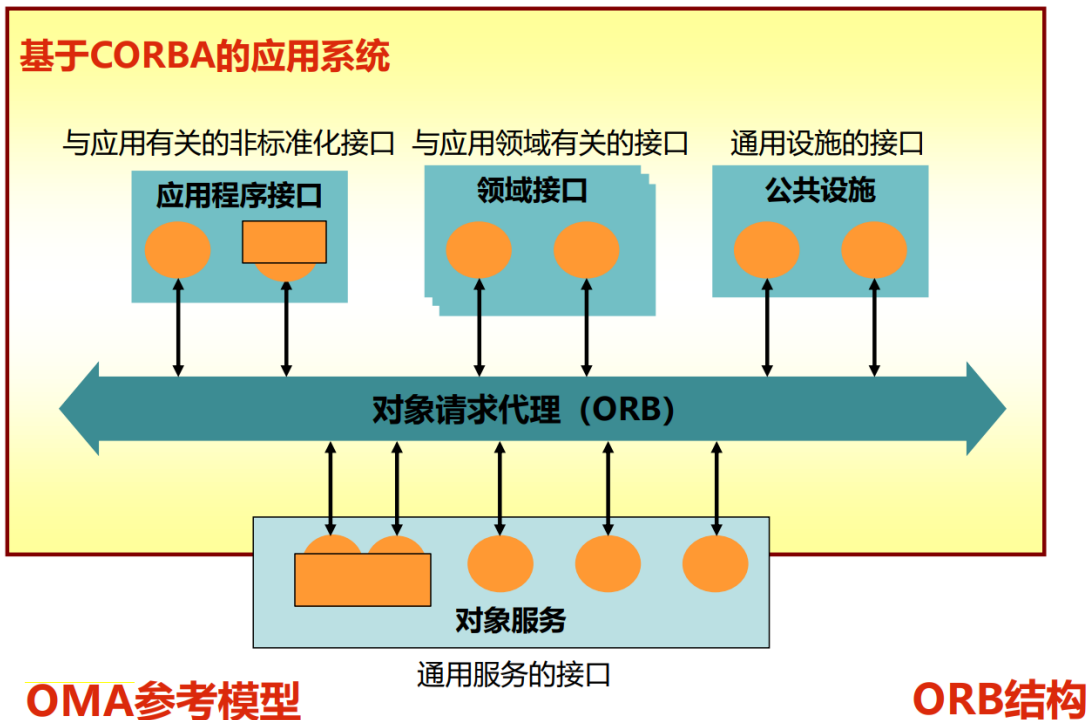
课程内容

第二章 CORBA基本原理

- 对象管理体系结构
 - OMA参考模型
 - CORBA中的公共服务
- ORB结构
 - 桩/框架结构
- CORBA对于可互操作的支持
 - 不同平台与语言之间的互操作性
 - 不同厂商的ORB产品之间的互操作性
 - 不同体系结构中的组件的互操作性

第二章 CORBA基本原理

➤ 对象管理体系结构——OMA

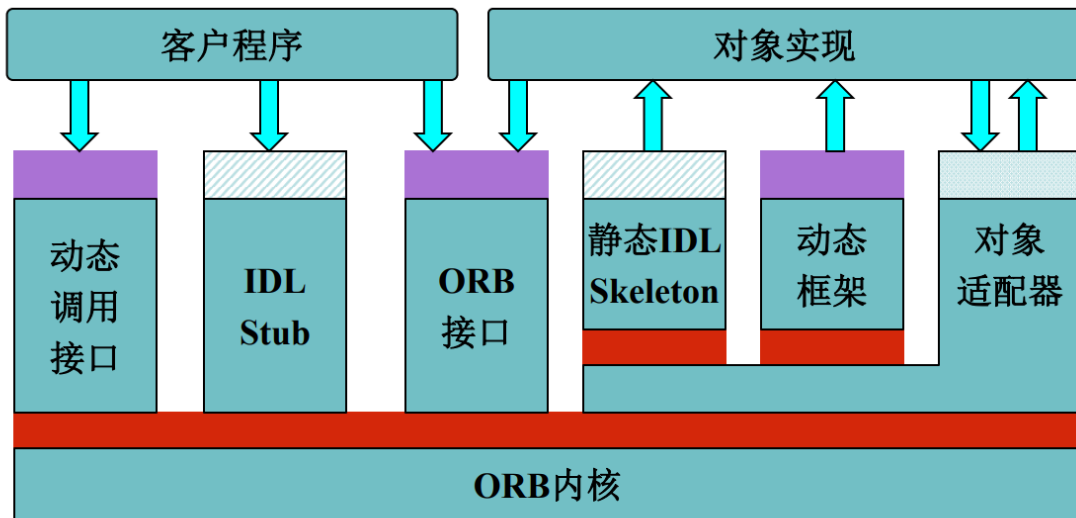


第二章 CORBA基本原理

- OMA: Object Management Architecture
 - OMA是CORBA所基于的概念框架，它描述了一个较高层次的分布式计算环境
- ORB(Object Request Broker,对象请求代理)
 - ORB是OMA参考模型的核心
 - 提供了分布式对象之间透明地发送请求或接收响应的基本机制,独立于实现对象的特定平台与技术
- 重要概念
 - 对象服务、公共设施、领域接口、应用程序接口、对象请求代理

第二章 CORBA基本原理

➤ 对象请求代理——ORB

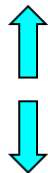


对于所有的ORB实现都相同的接口

每一对象类型都有桩和一个框架

可能有多对象适配器

依赖于ORB具体实现的接口



向上调用接口

向下调用接口

第二章 CORBA基本原理

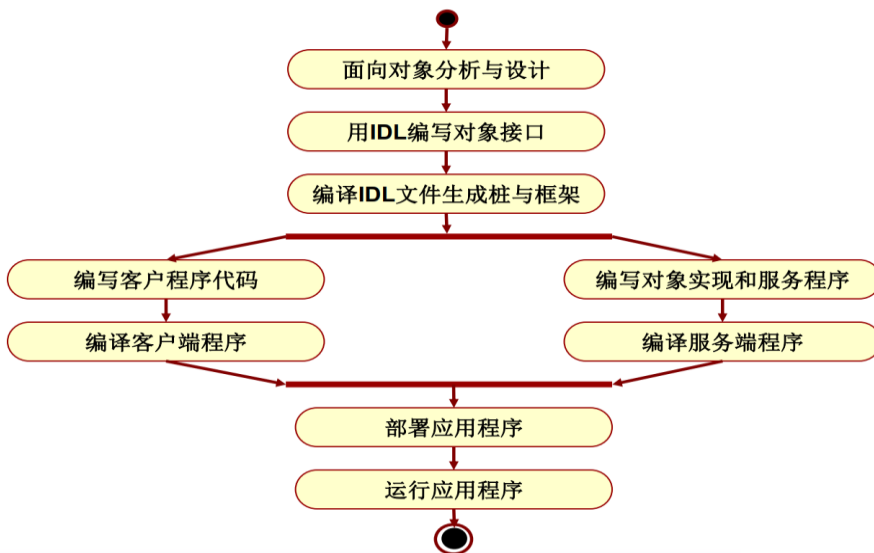
- 对象请求代理(ORB)结构中的重要概念
 - 对象接口、Stub/Skeleton结构(即静态调用方式动态调用接口(Dynamic Invocation Interface)、对象适配器(ObjectAdapter)
- 对象的可互操作性
 - 域(domain)
 - 域之间的桥接(Bridge)
 - CORBA对可互操作性的支持
 - GIOP——General Inter-ORB Protocol,通用ORB协议
 - IIOP——Internet Inter-ORB Protocol,因特网ORB间协议
 - ESIOP——Environment-Specific Inter-ORB Protocols

课程内容

第三章 基于CORBA的开发过程

➤ 基本开发过程

- 整体开发流程, 每一步的主要工作



课程内容

第四章 编写对象接口 (CORBA)

- OMG IDL的语法与语义
 - 模块、类型、常量、异常、接口、值类型
- 使用值类型(valuetype)
 - 对象副本的状态变化不会影响原来的对象
- 编写对象接口的准则(见课件)

第四章 编写对象接口 (CORBA)

➤ 接口定义语言

➤ OMG的接口定义语言(Interface Definition Language, IDL)

➤ IDL的规格说明:分为6类:模块、类型、常量、异常、接口以及值

```
module Bank{
```

```
//类型定义
```

```
//常量定义
```

```
//异常定义
```

```
//接口定义
```

```
//值定义
```

```
}
```


第四章 编写对象接口 (CORBA)

➤ 模块

- 语法:用关键字module来声明
- 语义:模块用于限制标识符的作用域

➤ 类型

- 语法:用关键字enum、struct、union等来声明自定义数据

➤ 常量

- 语法:用关键字const 来声明

➤ 异常

- 语法:用关键字exception来声明

➤ 接口

- 语法:用关键字interface来声明

➤ 值类型:

- 语法:采用高级语言类声明的形式使用关键字valuetype
- 值:值类型是一类特殊的类型声明
- 语义:值类型映射到Java语言的类

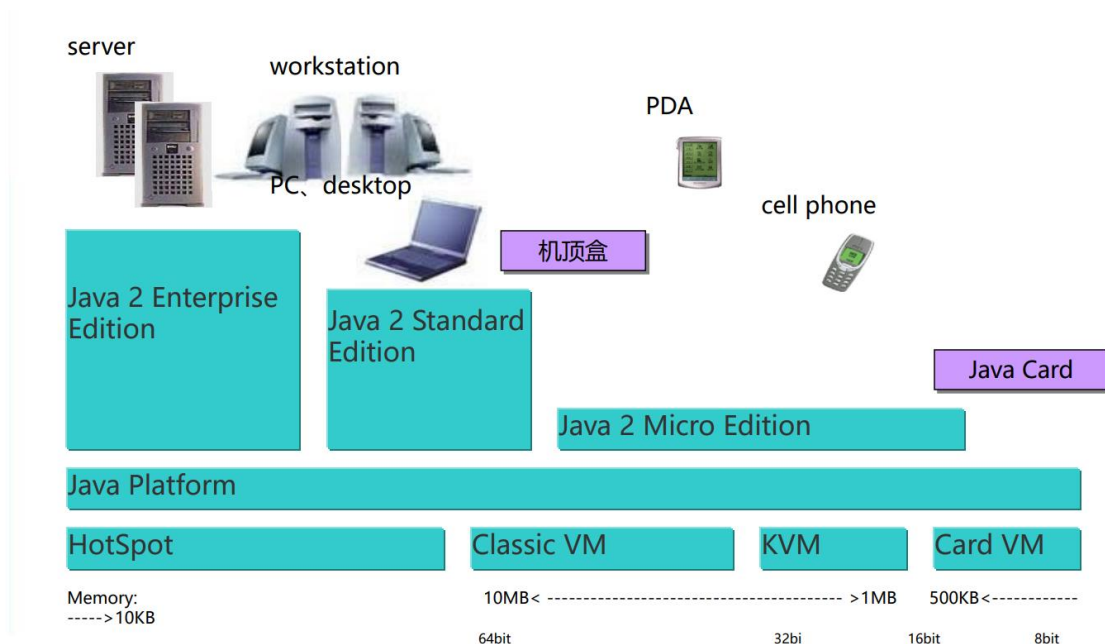
课程内容

第六章 J2EE体系架构

- J2EE构件
- J2EE公共服务
 - Service API
 - 运行时服务
- 通信支持
- J2EE应用程序
- 基于角色的开发

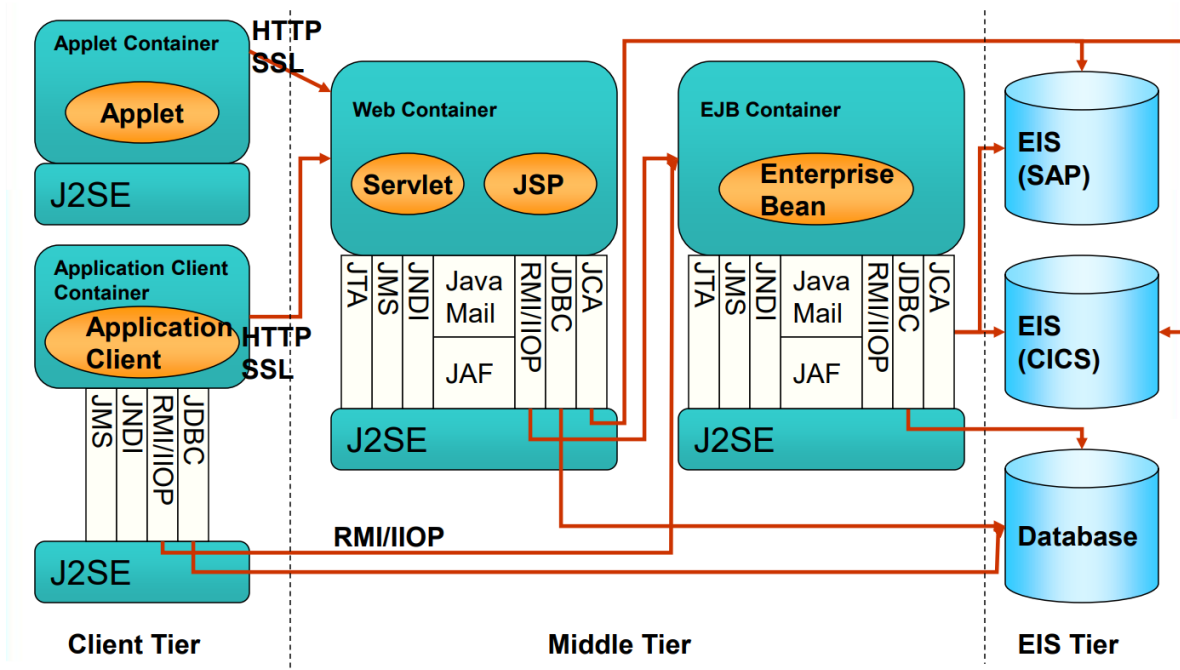
第六章 J2EE体系架构

- J2EE:支持以构件化的方法开发、部署、管理多层结构的分布式企业(Web)应用



第六章 J2EE体系架构

➤ J2EE的平台执行模型



第六章 J2EE体系架构

➤ J2EE Services

- JNDI、JDBC、JCA、JTA、Java Mail
 - JNDI: Java Naming and Directory Interface
 - JDBC: Java DataBase Connectivity
 - JCA: J2EE Connector Architecture
 - JTA: Java Transaction API

➤ J2EE Packaging

- Deployment Descriptors



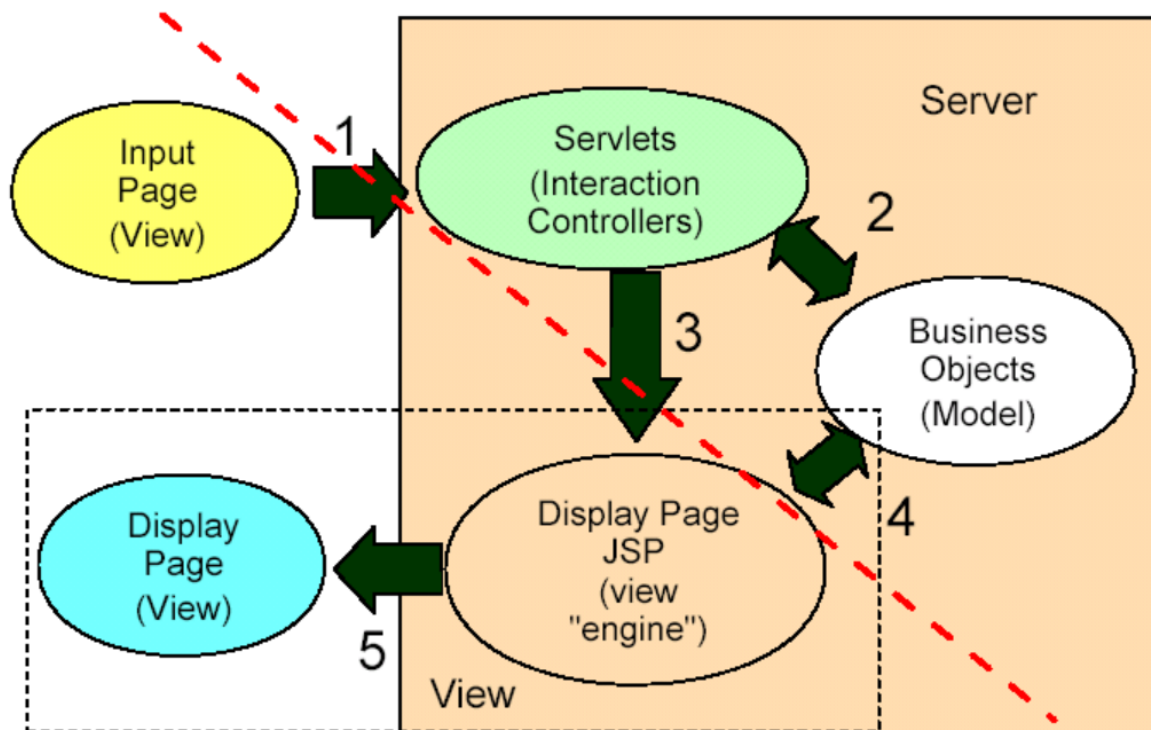
第六章 J2EE体系架构

➤ J2EE中的角色

- J2EE规范将应用系统的生命周期映射到了6个角色:
 - J2EE Product Provider
 - Tool Provider
 - Application Component Provider
 - Application Assembler
 - Deployer
 - System Administrator

第六章 J2EE体系架构

➤ MVC设计模式



课程内容

第七章 EJB技术基础

- EJB体系结构
 - EJB与Java Bean对比
 - EB体系结构中的构件(两个接口+一个类)
- EJB设计原则
 - 理解作用,了解对应关系

第七章 EJB技术基础

➤ EJB构件技术

- 分布式对象技术
- 服务端构件技术(Server-side Component Technology)
- CTM (Component Transaction Monitor)技术

➤ EJB的特点

- 公共服务框架
- 平台独立性
- 封装特性
- 可定制性(通过部署描述符实现)
- 协议无关性
- 通用性

第七章 EJB技术基础

➤ EJB vs JavaBean

- EJB用于服务端应用开发
- JavaBeans用于客户端应用开发

EJB	JavaBean
EJB 用于 服务端应用开发	JavaBean 用于 客户端应用开发
EJB 构件是可部署的	JavaBean 构件是不可部署的
EJB 支持使用部署描述符 对 EJB 应用进行定制化	JavaBean 中对应用的 定制化只能在开发阶段
EJB 构件是分布式对象	JavaBean 构件 不是分布式对象
EJB 构件对终端用户不可见	部分 JavaBean 构件 对终端用户可见

第七章 EJB技术基础

➤ EJB体系结构中的构件

➤ EJB体系结构由6种构件组成

- Enterprise Bean
- Home Interface
- Remote Interface
- EJB Container
- EJB Server
- EJB Client

➤ Enterprise bean包含三种类型

- Session bean、Entity Bean、Message Driven Bean
- Session bean
 - Stateless Session Bean
 - Stateful Session Bean

第七章 EJB技术基础

- EJB体系结构中的另两类构件是
 - Remote Interface
 - Home Interface
- 其它重要概念
 - EJB Container、EJB Server、EJB Client

第七章 EJB技术基础

➤ EB容器服务

- Remote Client Connectivity、Life Cycle Management、DeclarativeTransaction Management、Security Management、Access ControlList、Persistence Management、Resource Management

➤ EJB开发角色

- Enterprise Bean Provider
- Application Assembler
- EJB Deployer
- EB Server Provider
- EJB Container Provider
- System Administrator

课程内容

第八章 EJB构件开发过程

➤ 无状态会话构件开发与使用

- 生命周期特征
- 为什么会设计为无状态会话构件
- 客户端(对构件生命周期的影响)

➤ 有状态会话构件开发与使用

- 生命周期特征
- 为什么会设计为有状态会话构件
- 客户端(对构件生命周期的影响)

➤ 实体构件开发与使用

第八章 EJB构件开发过程

➤ 创建EJB需要做哪些事？

- Remote接口定义、Home接口、create方法定义、异常处理、Enterpbean定义与编写

➤ 无状态会话bean的生命周期包含两个状态

- Method Ready State
- No State

➤ JAR文件

➤ Java异常

第八章 EJB构件开发过程

➤ 有状态会话构件

- 有状态会话bean的生命周期包含三个状态
 - No State
 - Method Ready State
 - Passivated State

➤ 实体构件

- Entity bean的生命周期包含三个状态
 - Pooled State
 - Ready StateNo State

第八章 EJB构件开发过程

- Entity Bean----数据库表
- Entity Bean 实例/对象----数据库中记录
- 对应数据库表结构开发Entity Bean
 - Remote接口上约定记录上可以执行的操作(一个Remote接口操作个Entity Bean实例)
 - Home接口中包含插入记录、查找记录、删除记录的操作(Home接口包含Entity Bean工厂操作)

第三部分 Java EE与EJB \$3-2 Java Web工程

- 静态Web与动态Web
- Web服务器、应用服务器
- Servlet与JSP
- include编译指令与include动作指
- Cookie与Session

第十章Web Services概述

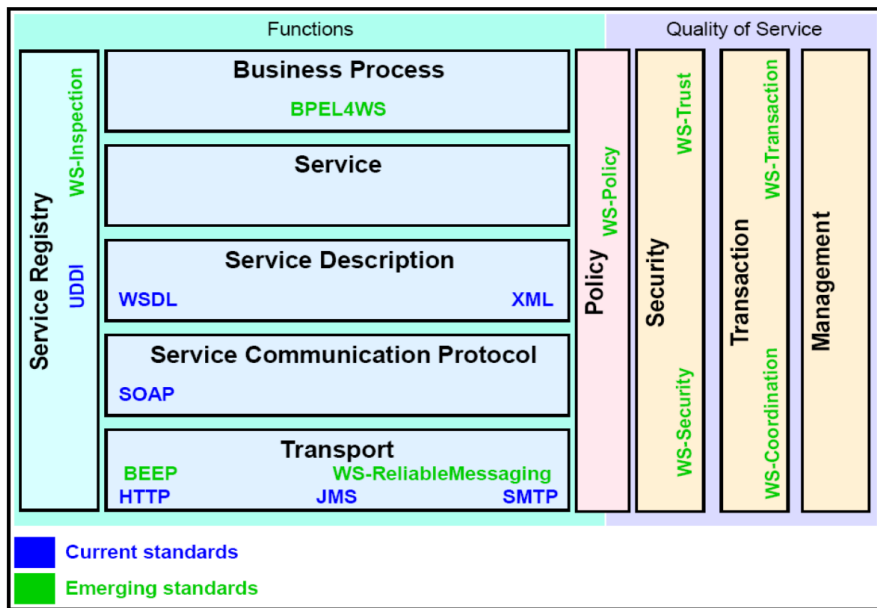
➤ Web服务体系结构

- SOA架构(三个参与者、三个基本操作、基本工作过程)
 - SOA中的三个参与者:服务提供者、服务代理者、服务请求者
 - 三个基本操作:发布服务、发现服务, 基于服务描述绑定或调用服务
- 使用的协议(基本作用)
- 应用方向

第十章 Web Services概述

➤ SOA: Service-oriented Architecture

- SOA是一种IT体系结构样式, 支持将业务作为链接服务或可重复业务任务进行集成,可在需要时通过网络访问这些服务和任务
- SOA服务栈元素



第十章 Web Services概述

- Web Service是SOA一种重要的实现方式
 - 提供了一种分布式计算的方法,通过网络集成异构的应用
 - 与实现语言、操作系统和硬件环境无关
 - 涉及到的技术
 - Extended Markup Language (XML)
 - Simple Object Access Protocol (SOAP)
 - Universal Description, Discovery and Integration(UDDI)
 - Web Services Description Language (WSDL)

第十章 Web Services概述

➤ WSDL - Web Service Description Language

➤ 基于XML格式描述Web Service, 包括

数据类型定义

输入、输出数据的抽象和类型定义

服务所处的网络位置

portType所对应具体的消息传输协议和数据格式说明

服务支持的操作集/功能

➤ BPEL4WS

➤ 是专为整合Web Services而制定的一项规范标准

➤ UDDI

➤ Universal Description, Discovery and Integration

➤ 统一描述、发现和集成协议(UDDI)是一套基于Web的、分布式的、为Web服务提供的信息注册中心的实现标准规范

RESTful API

➤ REST中的要素

- 资源(Resources)
- 表现层(Representation)
- 状态转化(State Transfer)

➤ HTTP协议的无状态性

➤ HTTP协议中的操作方式

http动作	对应的SQL	动作语义	http动作	对应的SQL	动作语义
GET	SELETE	从服务器取出资源	PATCH	UPDATE	在服务器更新资源(客户端提供改变的属性)
POST	CREATE	在服务器新建一个资源	HEAD		获取资源的元数据
PUT	UPDATE	在服务器更新资源(客户端提供改变后的完整资源)	OPTIONS		获取信息。关于资源的哪些属性是客户端可以改变的

\$4-4 微服务(概念、架构与实现)

- 微服务架构与单体架构(Monolith)的对比
- 微服务与SOA的区别

服务组合

- 服务编制(Service Orchestration)、服务编排(Service Choreography)
- BPEL: Business Process Execution Language
- 服务编制的运行模式:
 - 集中式的执行引擎
 - 基于Hub的分布式引擎
 - 无Hub的分布式引擎
- Orchestration vs. Choreograph
- BPEL4WS vs. WS-CDL
- 服务组合方法:静态组合VS.动态组合