



J2EE技术

Allen Long

Email: allen@huihoo.com

<http://www.huihoo.com>

2004-04



JAVA

内容安排



- J2EE介绍
 - 。 JAVA版本
 - 。 J2EE定义、组成
 - 。 J2EE架构
 - 。 J2EE应用编程模型
 - 。 EJB—J2EE的基石、EJB2.0
- 应用服务器—电子商务的基础平台
 - 。 什么是应用服务器
 - 。 应用模型
 - 。 常见的应用服务器
 - 。 应用服务器选型



JAVA

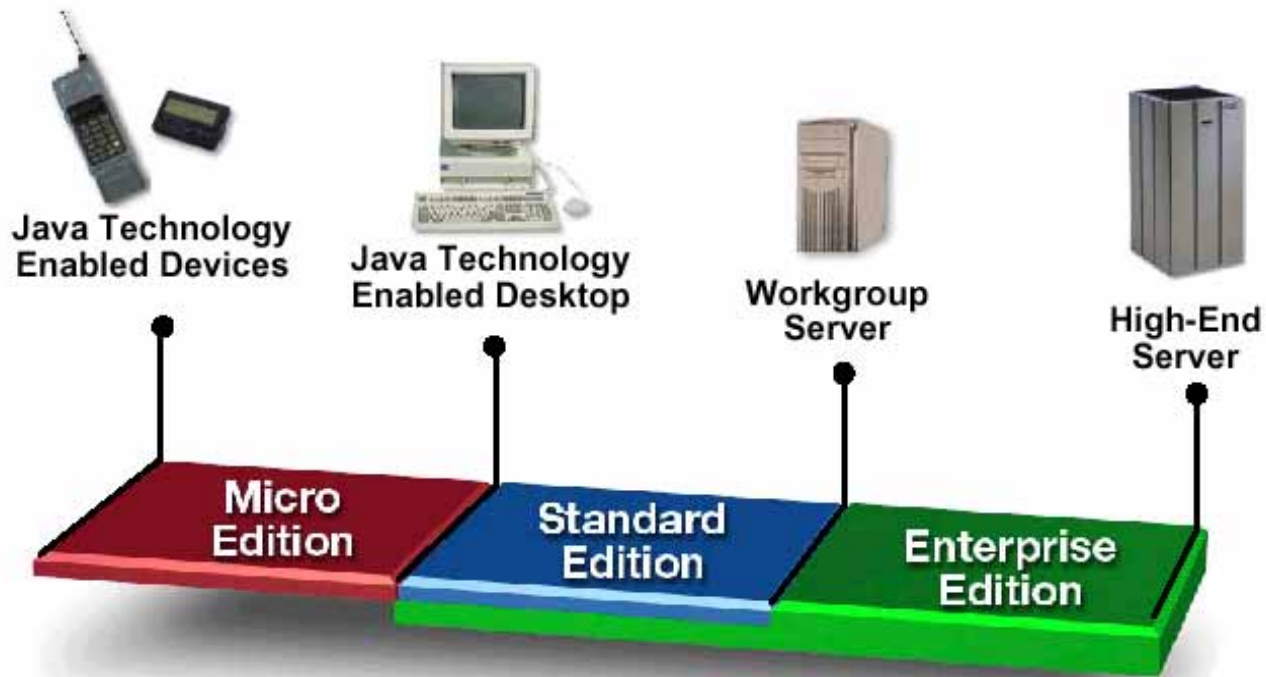
Java的三种版本



J2EE
Enterprise Java Software

J2SE
Core Java Software

J2ME
Java Mobility Software



什么是J2EE?

Java 2 Platform, Enterprise Edition (J2EE)



J2EE是一整套功能非常健壮的开发企业级应用的规范，是一个基于标准的开放平台：它包括以下技术：

- J2SE：J2EE基于J2SE之上
- EJB：组件标准
- RMI,远程方法调用
- JNDI,命名和目录服务
- JDBC,数据库连接
- JavaMail
- Jsp/Servlet
- Java的消息服务(JMS)
- JTA,事务处理API
- JCA,连接器：提供与现有企业系统的连接
- Java IDL
- XML

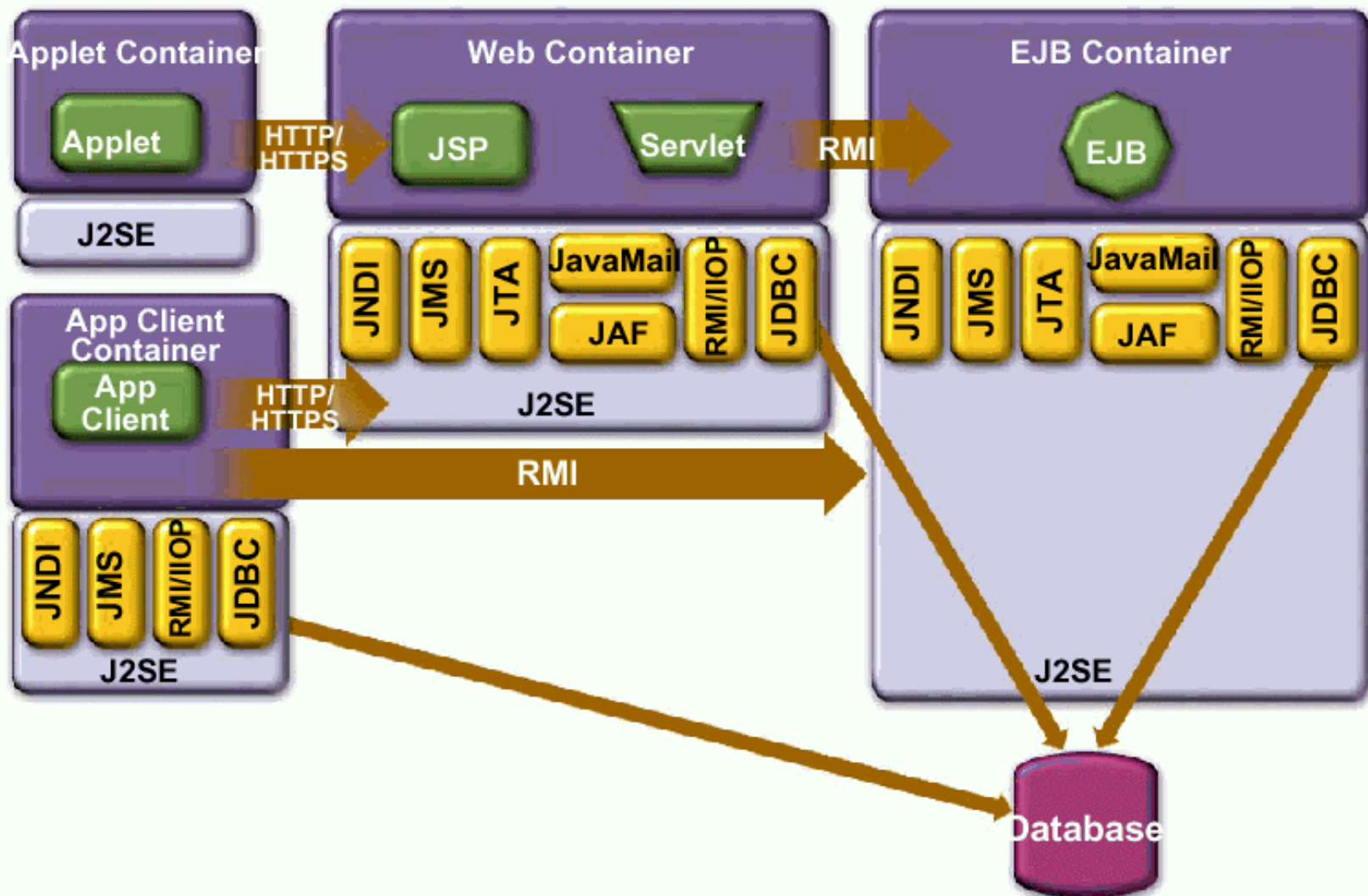
J2EE技术概观



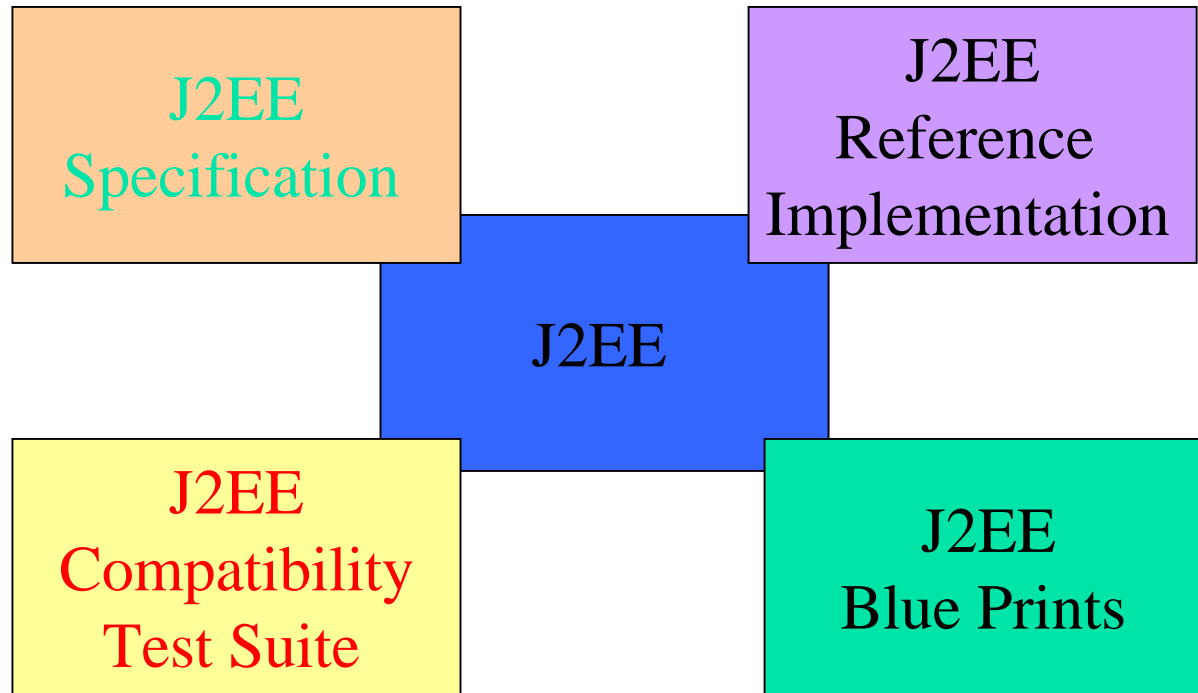
- 表现层 – Servlet/JSP
- 中间层 – EJB
- 中间层可用的企业服务
 - 事务服务 JTA
 - 目录服务 JNDI
 - 消息服务 JMS
 - 异步组件 Message-Driven EJB
- 数据层 - JDBC
- 远程调用 – RMI/IIOP
- 集成旧系统 - JCA



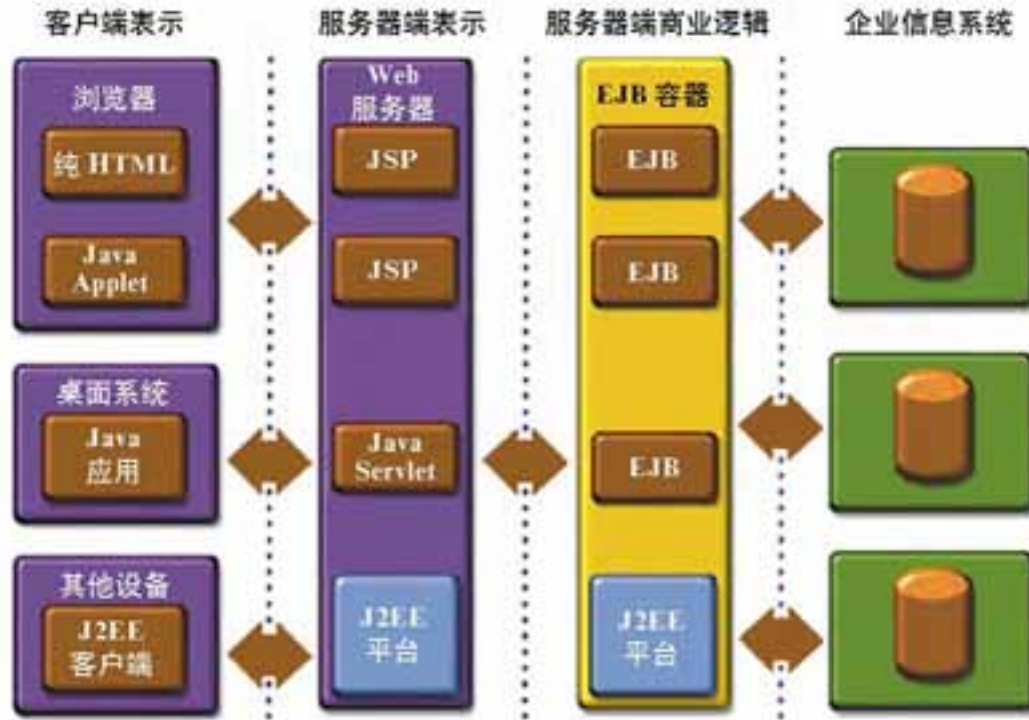
J2EE架构



J2EE组成部分



J2EE应用编程模型



客户层

商务逻辑层

数据层



JSP



JSP包含以下一些内容:

- 1、在JSP中使用的directive(指令)主要有三个：
 - 。 page指令
 - 。 include指令
 - 。 taglib指令

在JSP的任何地方，以任何顺序，一个页面可以包含任意数量的page指令

- 2、Scripting (脚本) 包括三种类型
 - 。 `<% ! declaraction %>`
 - 。 `<% scriptlet %>`
 - 。 `<%= expression %>`



JSP



■ 3、action (动作)

标准的动作类型有：

- <jsp:useBean>
- <jsp:setProperty>
- <jsp:getProperty>
- <jsp:param>
- <jsp:include>
- <jsp:forward>
- <jsp:plugin>

■ 4、Template data

除JSP语法外，JSP引擎不能解读的东西

JSP Custom tags



1. 标记库包含一个标记库描述符（TLD）和用于实现Custom tag的Java类
2. 在翻译阶段，JSP容器将使用TLD来验证页面中的所有的tag是否都被正确的使用。
3. 标记处理程序只是一个简单的适配器，而真正的逻辑是在另一个类中实现的，标记处理程序只是提供了一个供其他的可复用的类的JSP接口
4. 用法: `<%@ taglib prefix="huihoo" uri="/huihoo.tld" %>`
5. JAKARTA的I18N标记库
<http://www.huihoo.com/java/jsp/tablibs.zip>

Servlet



- 1 . ServletConfig
一个ServletConfig对象是servlet container在servlet initialization的时候传给servlet的

- 2 . 会话跟踪
 - 1) Session (推荐使用)
 - 2) cookie
 - 3) URL Rewriting

- 3 . SingleThreadModel
Servlet specification上建议，不要使用synchronized，而使用SingleThreadModel。



JAVA

JDBC



DataSource 接口 (JDBC 2.0) 或 DriverManager (JDBC 1.0)



- 1. DataSource

在JDBC2.0中，提供了透明的连接池（Connection pooling）。

一旦配置了J2EE应用服务器后，只要用DataSource获取连接(Connection)，连接池（Connection pooling）就会自动的工作。

- 2. ResultSet

java.sql.ResultSet接口中定义了三种类型的结果集

TYPE_FORWARD_ONLY

TYPE_SCROLL_INSENSITIVE 这种类型的结果集支持双向滚动

TYPE_SCROLL_SENSITIVE

- 3. JDBC驱动程序

- 4. 事务处理

--本地事务

java.sql.Connection接口可以控制事务边界（即开始和结束）。

--分布式事务

JDBC2.0 同JTA一起来实现分布式事务



JAVA

JDO



- 它具有所有必须的数据存储功能：增、删、改、事务、数据唯一性、缓冲
- JDO APIs (PersistenceManager, Query, Transaction)
- JDO提供了一个称为JDOQL的查询语言
- JDO可以支持在异种数据源中非常复杂的映射机制，然而，EJB/CMP模型仅适用于简单的JDBC模型。
- 使用JDO你没有业务对象的复杂性的限制（然而EJB/CMP不支持继承）。
- 使用JDO在你的业务数据对象中完全没有数据库代码（然而使用EJB/BMP 你的业务代码会掺杂入JDBC代码）
- JDBC只是面向关系数据库（RDBMS），而JDO更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML以及对象数据库（ODBMS）等等，使得应用可移植性更强。



什么是 EJB?

Sun's Enterprise JavaBeans™ spec



- SUN公司对EJB的定义是：
EJB的结构是开发和配置基于组件的分布式商务应用程序的一种组件结构。用EJB结构开发的应用程序是可伸缩的、事务型的、多用户安全的。这些应用程序可能只需编写一次，就可以在支持EJB规范的应用服务器平台上配置。
- 简单定义：
EJB是一个组件事务监控器的标准服务器端的组件模型



EJB的分类



- Entity Bean
 - Container-Managed Persistence Entity Bean
 - Bean-Managed Persistence Entity Bean
- Session Bean
 - Stateful Session Bean
 - Stateless Session Bean



实体bean (Entity Bean)



JAVA

定义：对于客户机，实体 enterprise bean 是一种持久性对象，它代表一个存储在持久性存储器（例如，一个数据库）中的实体的对象视图，或者是一个由现有企业应用程序实现的实体。

简单的讲实体bean 代表了一种数据集，可以访问这些数据集来满足用户请求。

特点：

- 代表数据库中的数据
- 是事务性的
- 允许多个用户共同访问
- 可以长期存在
- 持久性数据可以由容器管理
- 在 EJB 服务器失败后能继续生存



两种实体bean

- Bean-Managed Persistence (自管理的持久性)
 - 优点：
 - 开发者有完全的控制
 - 无需复杂的提供商支持
 - 缺点：
 - 编码复杂
 - 在改变时需要重新编码和部署
 - 影响可移植性
- Container-Managed Persistence (容器管理的持久性)
 - 优点：
 - 提供商解决，可能会有更好的缓冲和性能
 - 在部署描述符中进行改变即可
 - 可移植性好
 - 缺点：
 - 对容器提供商的工具依赖性强
 - 可能不易反映复杂的数据关系



会话bean(Session Bean)



定义：对于客户机，会话 enterprise bean 是一种非持久性的对象，它实现某些在服务器上运行的业务逻辑。想像一个会话对象的一种方式：会话对象是运行在服务器上的客户机程序的逻辑扩展。会话对象不在多台客户机之间共享

简单的讲会话 bean 代表这样的操作，它检索或存储数据以满足用户请求

特性：

- 代表单个客户机执行
- 可以是事务性的
- 可以更新共享数据库中的数据
- 生存期相对较短
- 其生存期通常就是客户机的生存期
- 任何持久性数据都由 bean 管理
- 可以依容器的判断予以删除
- 会在 EJB 服务器失败时被删除

两种类型的会话bean



- 无状态 (Stateless) bean
 - 表达一个无状态的服务 (如列表 , mail)
 - 不存储用户相关信息 , 进行对请求的响应
 - 暂态的
 - 可用来构造响应频繁而简单的访问的bean池
- 有状态 (Stateful) bean
 - 维护客户状态



何时使用会话Bean



- 使用会话bean 对针对于某一客户的处理或控制对象建模
- 对 workflow、任务和管理活动等建模（如订房、购物车等）
- 协调多个实体bean，控制实体bean之间的交互
- 将业务应用逻辑从客户端转移到服务器端



会话Bean与实体Bean的区别



会话bean

- 表示一个业务过程
- 每一客户一个实例
- Short-lived：与客户生命同步
- 暂态的
- 服务器崩溃后丢失
- 可以是事务性的

实体bean

- 表示业务数据
- 在多个客户间共享实例
- Long-lived：与数据库中数据同步
- 持久的
- 服务器崩溃后可重构
- 总是事务性的



消息驱动bean(Session Bean)



消息驱动Bean实际上实现一些业务逻辑的消息使用者。Bean在所选Queue或Topic中注册，用以实现MessageListener接口，并等待异步消息达到，引进了异步处理机制。

特性：

- 消息驱动bean不需要一个home或是远程的界面，这是由于他们仅仅是监听消息而不是响应来自客户机程序的直接调用。
- 消息驱动bean是异步消息的接受者（consumer），一个消息驱动bean只能接受一种类型的消息。
- EJB2.0的消息驱动bean，只接受源于JMS的消息。EJB2.1扩展该功能，使得消息驱动bean也可以从其它消息源产生的消息，如XML（JAXM）。



CMP(Container Managed Persistence)



EJB 容器需要两个条件才能支持容器管理持久性：

。映射 - 关于如何将实体 Bean 映射到某项资源（如关系数据库中的表）的信息

。运行时环境 - 一种容器管理持久性运行时环境，它使用映射信息在每个 Bean 上执行持久性操作

特性：

- 容器管理持久性过程包含三项操作：映射、部署和运行。



EJB Transactions



定义：对于客户机，会话 enterprise bean 是一种非持久性的对象，它实现某些在服务器上运行的业务逻辑。想像一个会话对象的一种方式：会话对象是运行在服务器上的客户机程序的逻辑扩展。会话对象不在多台客户机之间共享

简单的讲会话 bean 代表这样的操作，它检索或存储数据以满足用户请求

特性：

- 代表单个客户机执行
- 可以是事务性的
- 可以更新共享数据库中的数据
- 生存期相对较短
- 其生存期通常就是客户机的生存期
- 任何持久性数据都由 bean 管理
- 可以依容器的判断予以删除
- 会在 EJB 服务器失败时被删除



一个EJB的三个关键构件



作为Bean的开发者，主要关注于三个构件：

- EJBHome 接口（扩展javax.ejb.EJBHome接口）：使用了factory设计模式，定义了创建、查找EJB的方法。
- EJBObject接口（扩展javax.ejb.EJBObject接口）：使用了proxy设计模式，定义了(bean中实现的业务逻辑方法。
- Bean实现类（实现javax.ejb.EntityBean/SessionBean）：实现业务逻辑。



一个EJB的实现步骤



1. 创建Remote接口
2. 创建Home接口
3. 创建Bean的实现类
4. 编译远程接口、Home接口、bean实现类
5. 创建部署描述符
6. 将以上三个文件与部署描述符文件打包为一个ejb-jar文件
7. 部署EJB



Ejb-jar文件



- 。 Enterprise Bean
- 。 Remote interface
- 。 Home interface
- 。 Deployment descriptor

生成ejb-jar文件: `jar cmf HelloWorld.jar *`



EJB部署



将ejb-jar文件部署到应用服务器中，使它能被客户端访问

。核对Ejb-jar中文件

。 Ejb容器提供的工具生成EJB对象和Home对象

。 EJB容器提供的工具生成必需的RMI存根(stub)和骨架(skeletons)





EJB2.0



1. 新的CMP组件模型，持久性管理器
2. 新的支持JMS的Bean类型
3. 新的EJB2.0查询语言：EJB QL
4. 在本地接口中包含业务方法



J2EE对XML的支持



- J2EE标准未包括对XML Web服务的支持
- 各个厂商提供自己的方案
 - 支持UDDI /WSDL /SOAP
 - WSI组织保证互操作性
- SUN的方案
 - JAXP / JAXB / JAXM / JAXR / JWSDL
 - JAX / RPC
 - JAX * – Java API for XML *



应用服务器

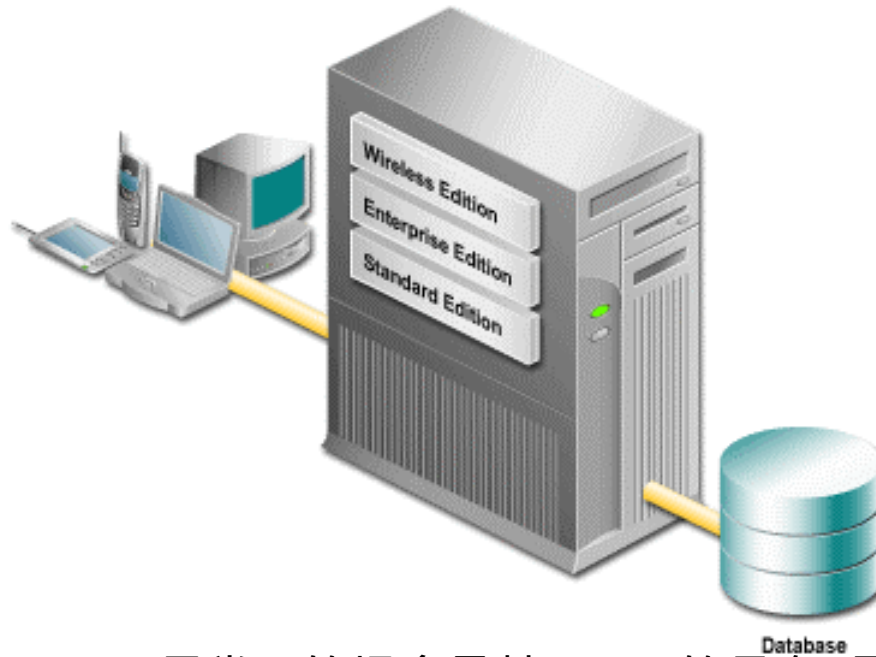
(Application Server)



- Application Server是一个设计、开发、调试、部署、运行、管理一个Internet电子商务应用系统的基础平台，它最适用于B2B或B2C电子商务网站的开发。
- Application Server是“Internet 上的操作系统”，是电子商务的基石。



应用模型



Application server最常用的场合是基于web的具有3层结构的应用。

第1层 前端： - Browser浏览器 (thin client瘦客户机),为用户提供的GUI图形接口

第2层 中间层： - Application Server - 应用程序、业务逻辑

第3层 后端： - Database Server



常用的应用服务器



JAVA

- BEA Systems Weblogic
- IBM WebSphere
- Sun iPlanet, SUN ONE Applications Server
- Oracle Oracle9iAs
- IONA iPortal Server
- Pramati Server
- Borland Appserver
- Apusic(国内)
- JBoss(开源)
- JOnAS(开源)
- JFox(开源,国内) ☺

应用服务器的选型



- 符合EJB规范(1.1,2.0)
- 支持实体Bean
- 持久性
- 选择基于CORBA与RMI的产品
- RMI-IIOP API的使用
- 经JNDI的JTA的可存取性
- 协议问题
- 集成层次的支持
- 内存数据的高速缓存
- 可测量性
- 高度的实用性
- 安全性
- IDE集成
- 智能负载均衡
- 无状态的透明的结束出错
- 分组(有状态的结束出错)
- 合理的关机
- 实时部署
- 分布式事务处理
- 与现有系统的集成
- 异步通信的支持
- 具有Bean产商的集成
- 专业化的服务
- 非技术标准



内容回顾



- J2EE介绍
 - 。 JAVA版本
 - 。 J2EE定义、组成
 - 。 J2EE架构
 - 。 J2EE应用编程模型
 - 。 EJB—J2EE的基石、EJB2.0
- 应用服务器—电子商务的基础平台
 - 。 什么是应用服务器
 - 。 应用模型
 - 。 常见的应用服务器
 - 。 应用服务器选型



JAVA

总结



- J2EE 是企业级应用的主流技术
- J2EE不仅仅是一种开发语言，更是一种基于标准的、开放的平台，提供的是一种构架
- J2EE是基于J2SE之上，捆绑了一些企业级的API
- EJB提供了健壮的、极具竞争力的组件开发模型
- 应用服务器是J2EE企业级应用的基础平台、它为EJB提供了运行的环境，是J2EE应用的关键。



参考资料



- <http://java.sun.com/j2ee>
sun公司的j2ee站点
- <http://www.huihoo.org/jfox>
JFox – J2EE开放源代码实现
- <http://www.huihoo.com>
国内一个关于中间件的专业站点



结束



谢谢大家！

Allen@huihoo.com

<http://www.huihoo.com>

