



JAVA

EJB技术

Allen Long

Email: allen@huihoo.com

<http://www.huihoo.com>

2004-04



内容安排



■ EJB介绍

- EJB是什么
- EJB架构
- EJB分类
- EJB容器
- EJB生命周期管理
- EJB事务处理
- EJB安全
- EJB设计策略
- EJB设计模式
- EJB性能与扩展性



JAVA

什么是 EJB?

Sun's Enterprise JavaBeans™ spec



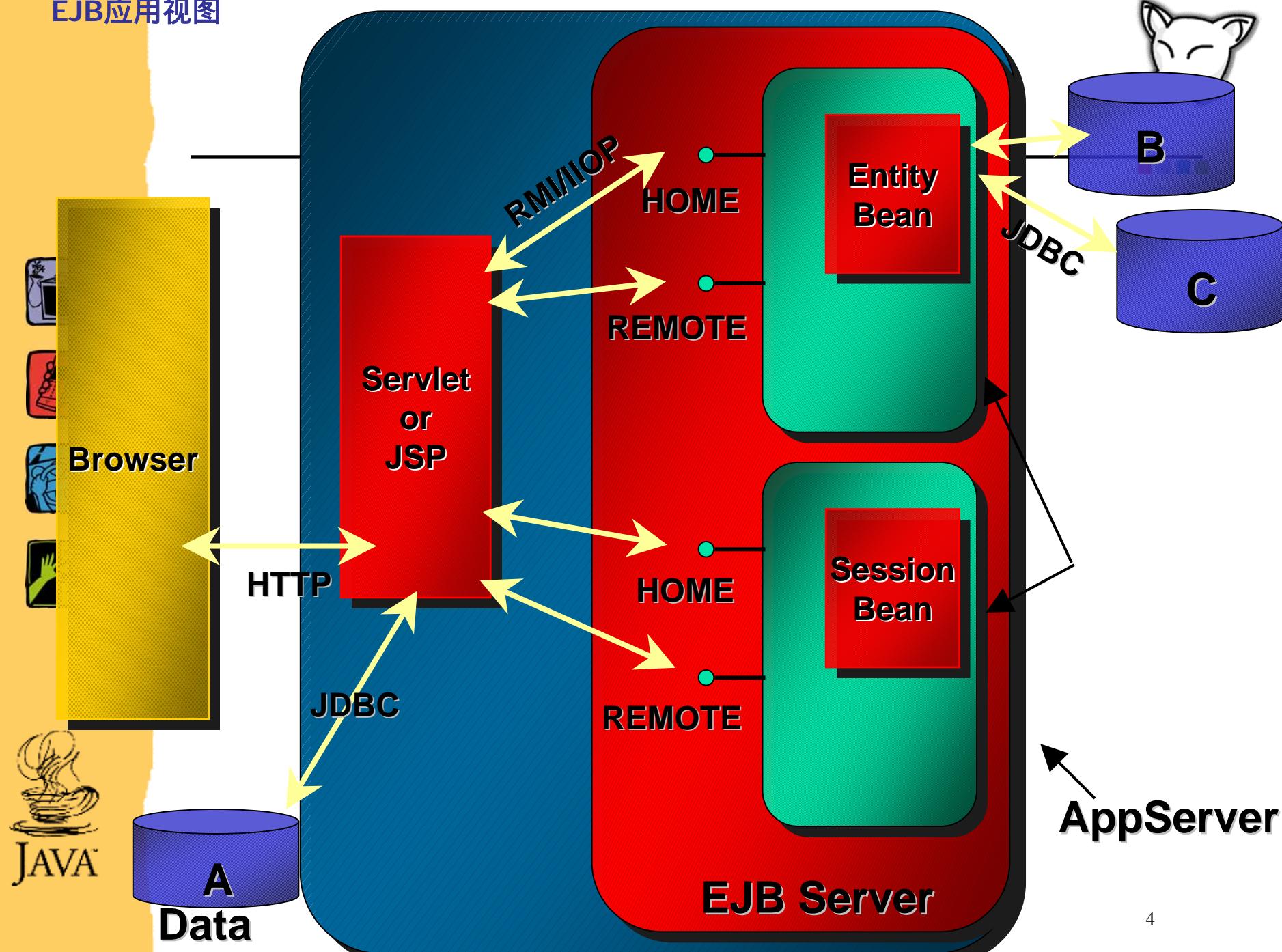
- SUN公司对EJB的定义是：

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。

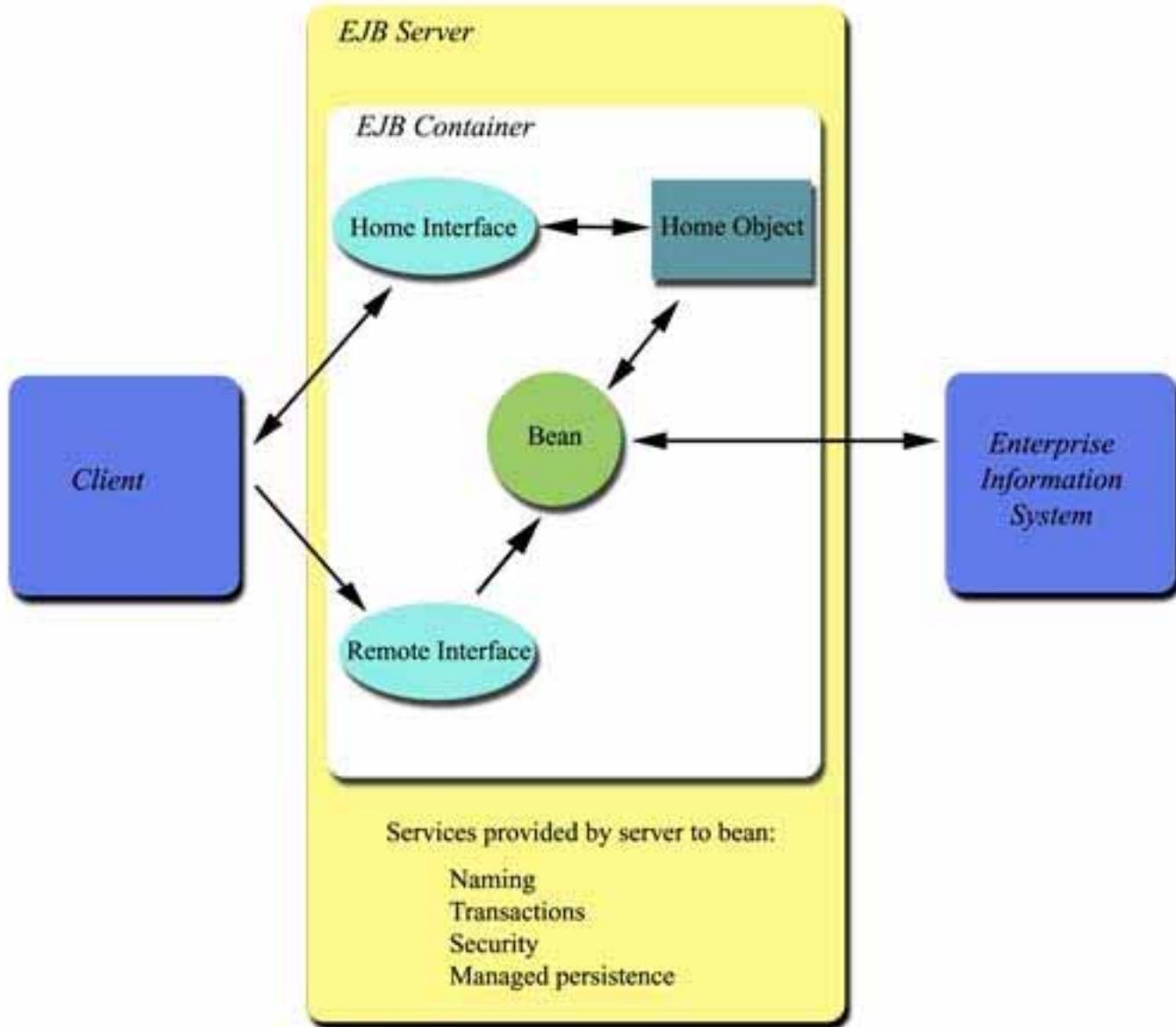
- 简单定义：

EJB是一个组件事务监控器的标准服务器端的组件模型

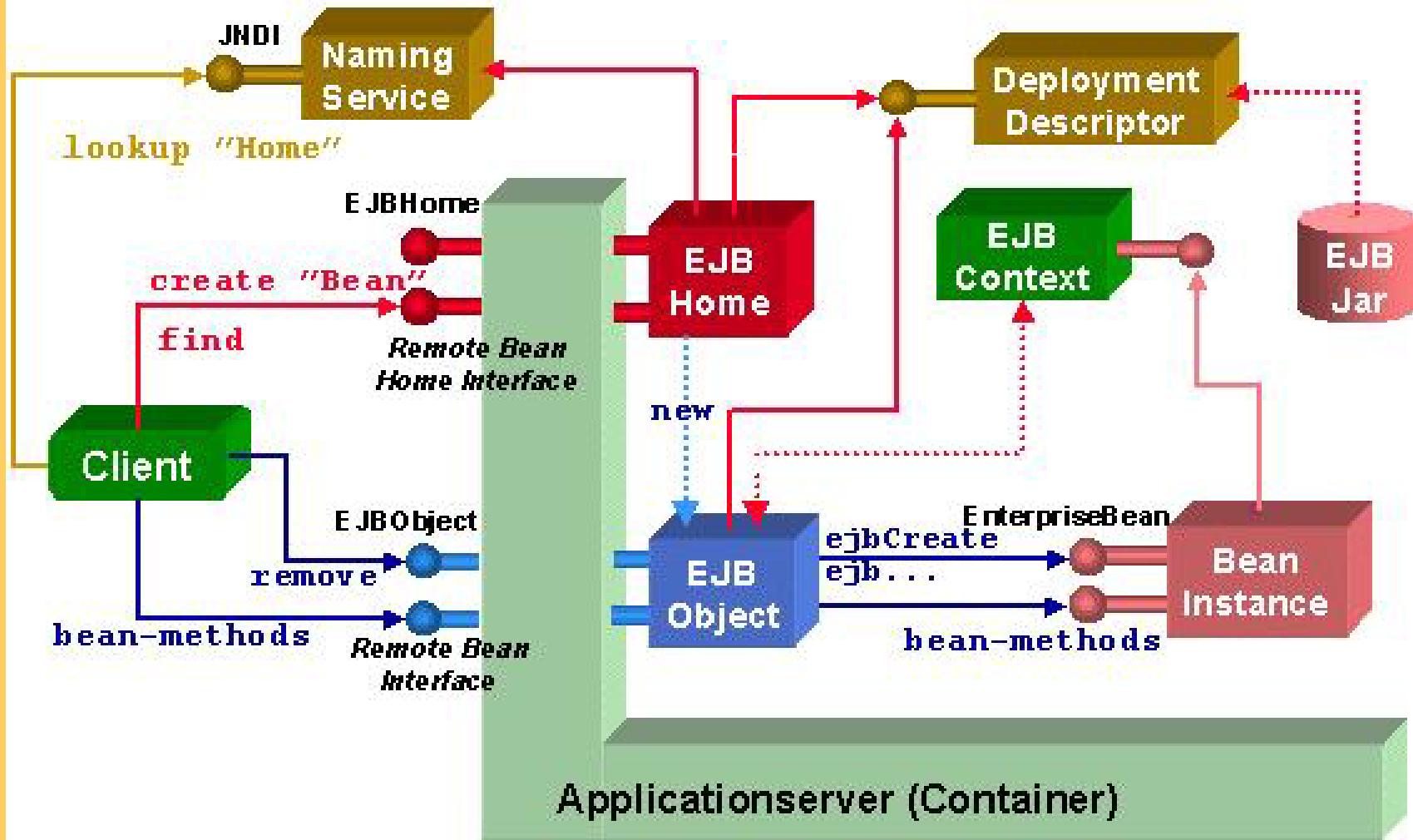
EJB应用视图



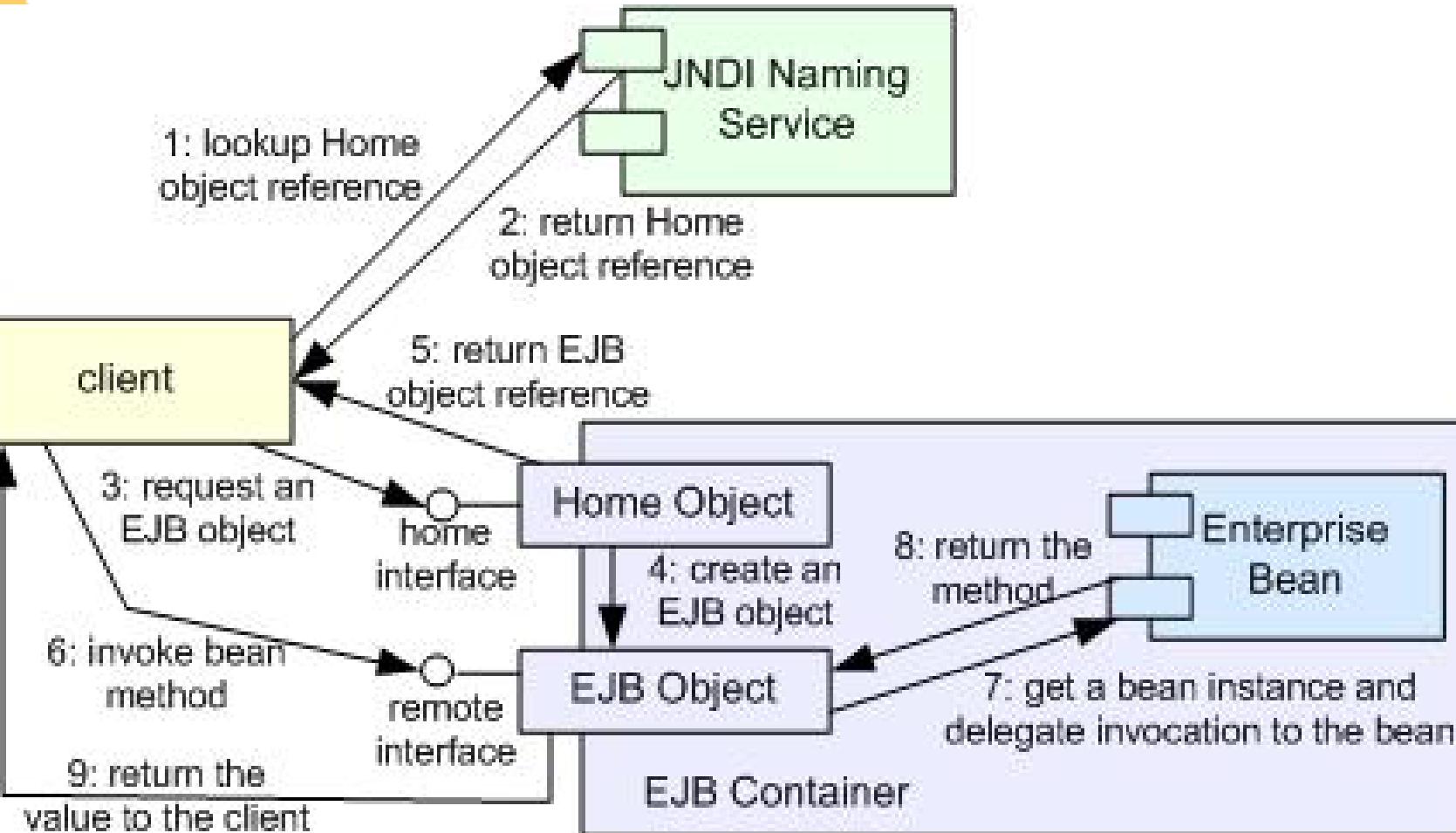
EJB架构图一



EJB架构图二



Client与EJB容器的交互过程



EJB的分类



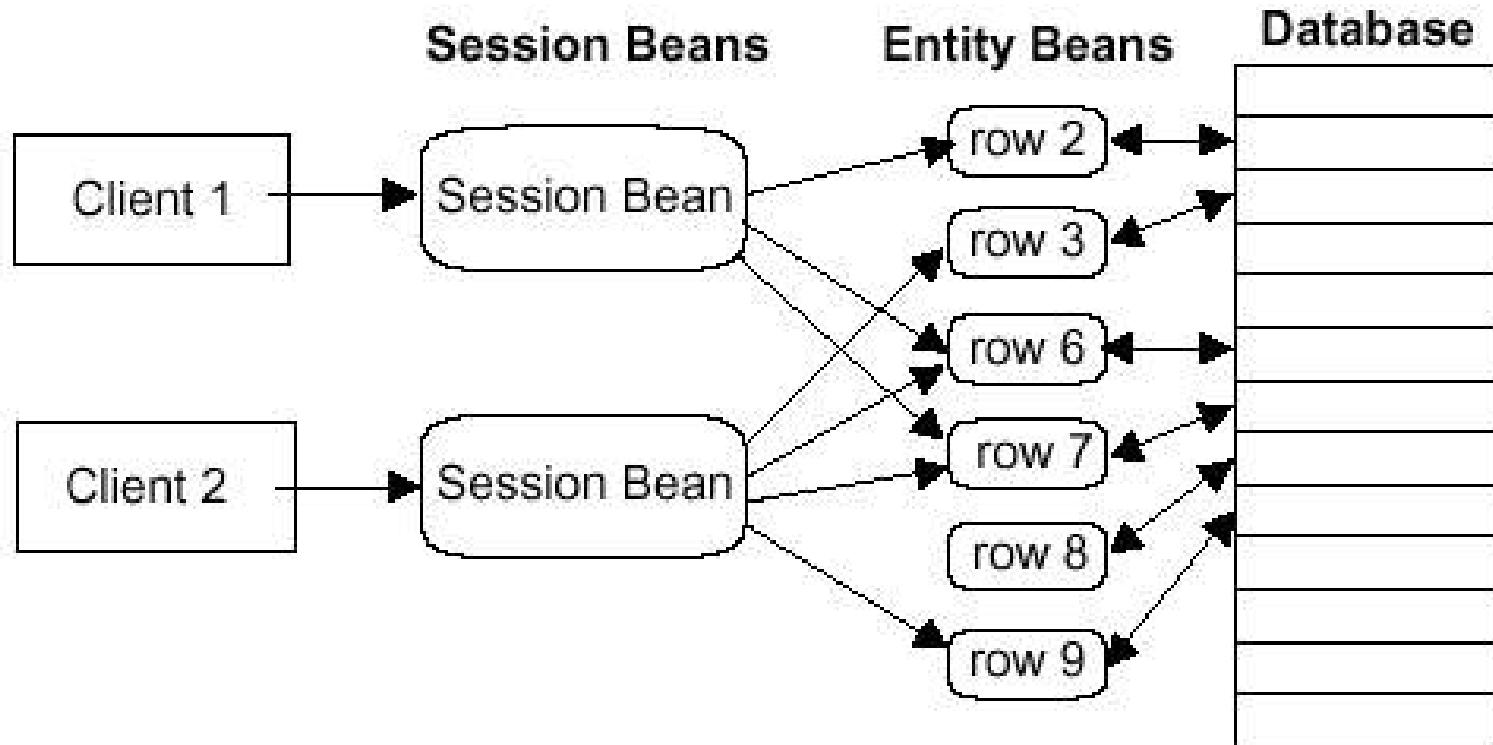
- Entity Bean
 - Container-**M**anaged Persistence Entity Bean
 - Bean-**M**anaged Persistence Entity Bean
- Session Bean
 - Stateful Session Bean
 - Stateless Session Bean



会话bean和实体bean应用层次



JAVA



实体bean(Entity Bean)



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



JAVA

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



JAVA



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。



JAVA

Entity Bean的一些优化



- Transacted Entity Bean
 - Property true 意味着在每个事务的开始和结束都要读数据和写数据 (即使数据没有变化)
 - Property false 指仅在生成时读一次数据, 以后再也不读(即使数据已陈旧)
- Read-only Bean (BEA 特有的功能)
 - BEA 特有的功能以防止在状态无变化时不调 Store 方法
 - BEA 亦有刷新的设置可允许数据按时间间隔刷新
 - 只读的, 事务的 entity bean 行为类似于 .NET 中运用 DataReader 的类 (每次读, 重不写)
- CMP Opt Tool 是 WebSphere 的优化
 - 读取 bean 的源代码, 判断数据访问模式, 删除不必要的 Store 调用
 - Option A, B and C caching 亦可规定 bean 的数据库访问是否是排他的
 - Option C意味着排他的数据访问, 所以 Store method 可以懒方式调用



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。

JDO(Java Data Object)



JAVA

- 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。

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是：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

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）。

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



JAVA

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
- 对工作流、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

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



JAVA

EJB-QL



JAVA

1. 基于SQL-92标准
2. EJB-QL主要用于finder和ejbSelect()方法
3. EJB-QL查询是由ejb-jar.xml文件的<query>定义

```
<query>
  <description>finds students in a given grade</description>
  <query-method>
    <method-name>findStudentsInGrade</method-name>
    <method-params>
      <method-param>int</method-param>
    </method-params>
  </query-method>
  <ejb-ql><![CDATA[FROM StudentCMPBean s WHERE s.grade = ?1]]></ejb-ql>
</query>
```

Local方法



- 1. EJB 2.0规范定义了两个附加的元素:<local-home>和<local>
- 2. 接口不扩展java.rmi.Remote
- 3. session和entity beans可以同时运用远程和本地接口,这带来了极大的灵活性

```
<enterprise-beans>
  <session>
    <ejb-name>...</ejb-name>
    <local-home>...</local-home>
    <local>...</local>
    <ejb-link>...</ejb-link>
    ...
  </session>
</enterprise-beans>
```



JAVA

EJB容器分为四种



EJB 服务器

EJB容器

StatelessSession
Container



StatefulSession
Container



EntityContainer



MDB
Container



部署

运行环境

检测、管理

事务服务

安全服务

其它服务



JAVA

EJB容器



EJB Container提供者负责如下功能：

- 。ejb部署
- 。对于部署的ejb提供必要的运行环境

从ejb来看，容器是目标操作环境的一部分。容器运行环境提供部署的ejb的事务，安全管理，远程客户分布式网络服务，资源的可扩展行管理，以及管理服务平台的一些其它服务。容器的主要责任是为EJB组件提供运行时期的环境，包括部署，bean的生命周期管理，以及服务的运行。因此如何实现这些功能，是设计EJB Container的目标。

在EJB的构架中定义的EJB Container提供者的责任，要求EJB Container和Server必须实现的。因为EJB规范并没有给出EJB Container和Server间的接口构架，所以依赖提供商来分隔EJB Container和Server之间的基本功能实现。

因此从EJB Container提供者的定义来说，EJB Container的必须功能如下：

1. 部署ejb
2. 提供ejb运行的基本环境，管理bean的生命周期
3. 管理和检测ejb的运行情况
4. 提供ejb的事务，安全等等服务





EJB容器的设计方法



JAVA

客户对于EJB方法的调用，实际上通过由容器提供的Home接口和EJBObject接口，因此在容器端，必须有这两个接口的实现，然后由其转交给bean实例，完成请求调用。通常根据生成实现类的过程不同，把EJB容器的设计方法分为两种：

- 1. 预编译方式
 - 通常在部署工具部署bean时，生成Home和EJBObject接口的实现类。在client方法调用时，装载和实例化这些类，从而提高EJB容器的性能。但是也有它的缺点：必须用特定的部署工具，并且也不支持热部署。例如JonAS EJB容器就采用的这种方式，它提供了一个工具GenIC，GenIC编译源码来生成实现类。一些商业EJB Container，象Websphere和Weblogic也是这种设计。（可以通过它的部署过程生成的文件看出）。
 - 2. 动态代理方式
 - 在容器内部署或者在调用时，用反射的方式动态生成Home和EJBObject接口的实现类，这种方式灵活，而且支持热部署。但是因为要大量用到java的反射机制，从而会降低EJB容器的性能。JBoss采用的就是这种方式，而且发挥了Dynamic Proxy的最大性能，特别是3.x版本，在结构、管理等等方面都采用了这种方式。在Sun的一篇关于reflect性能的文章中，jdk1.4x比以前的版本有很大的性能提高，因此容器的性能不会成为阻碍。

EJB 生命周期



实体Bean长于客户程序的生命周期,表示持久性数据,EJB容器崩溃后可恢复 (只要它存放在数据库中)

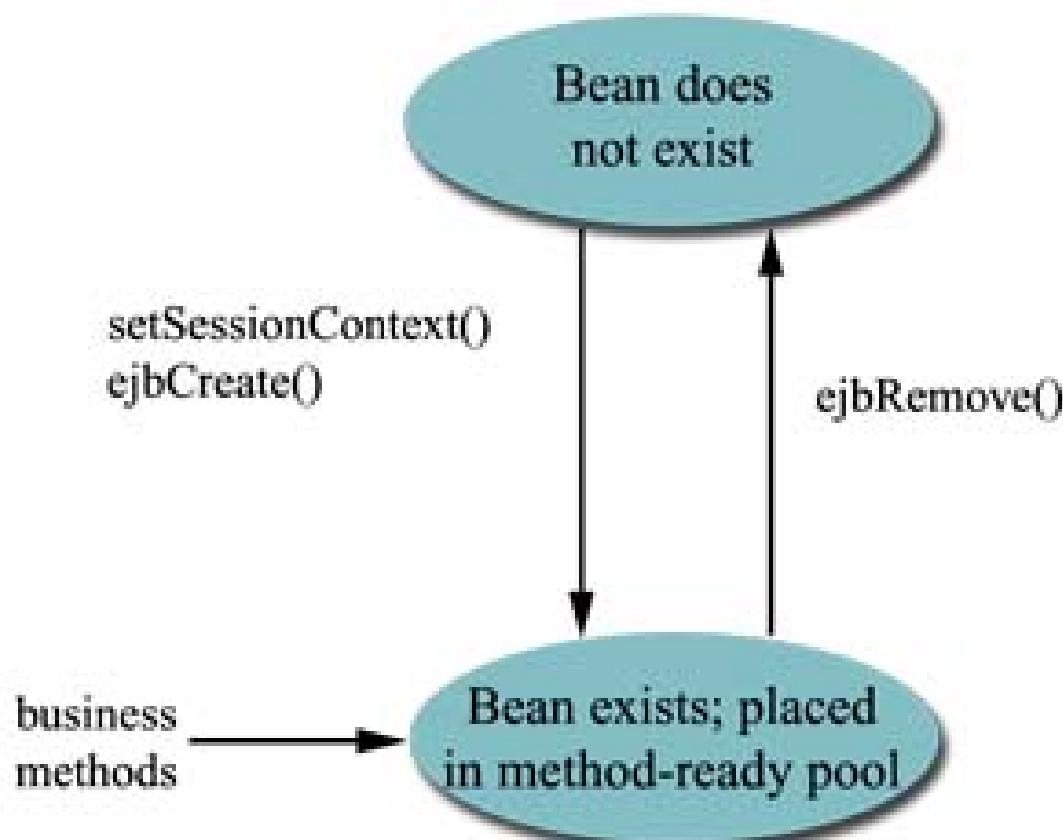
会话Bean通常由客户会话控制,表示非持久性数据.EJB容器崩溃后无法恢复.



无状态会话Bean生命周期



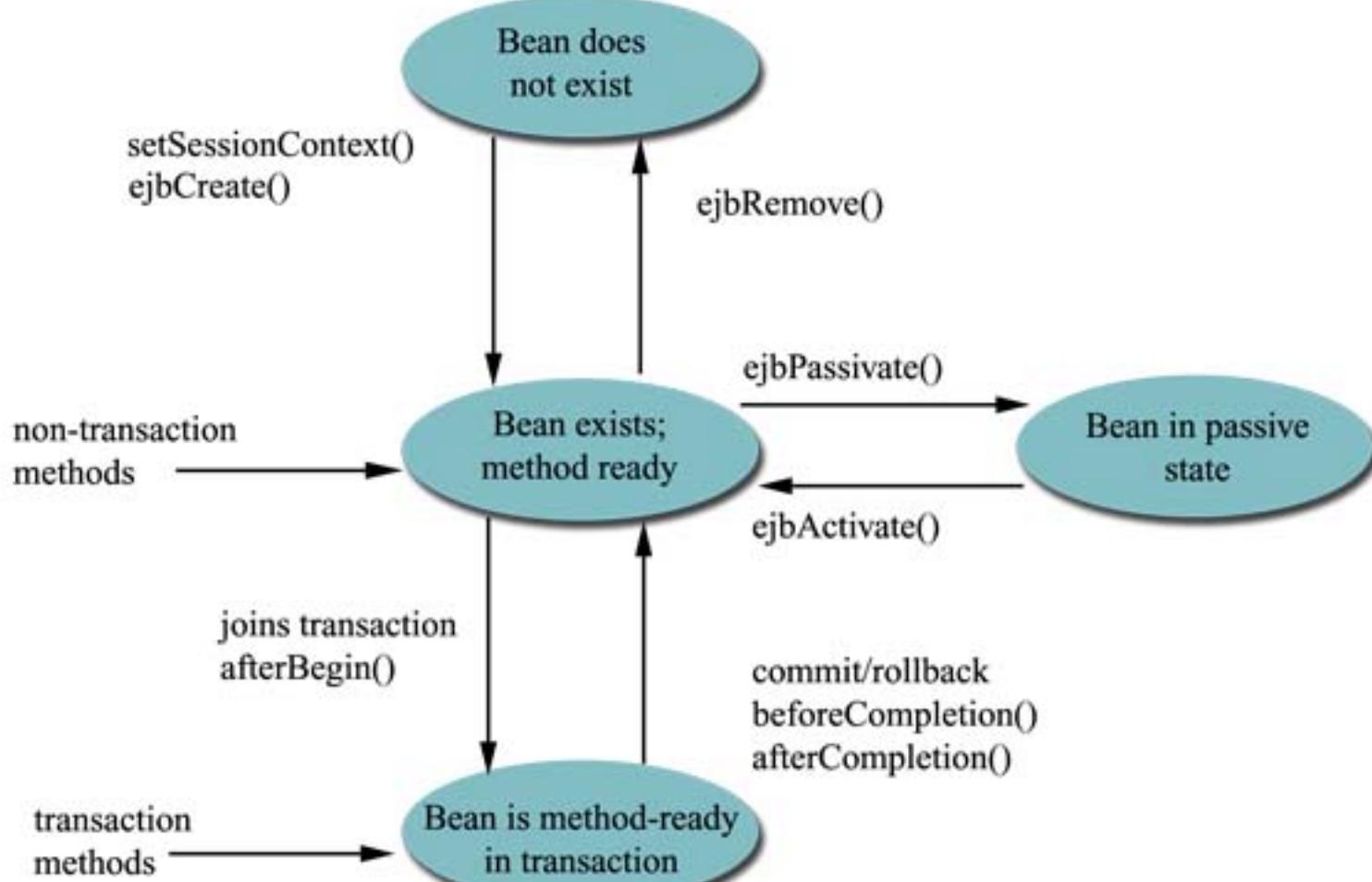
Stateless session bean life cycle



有状态会话Bean生命周期



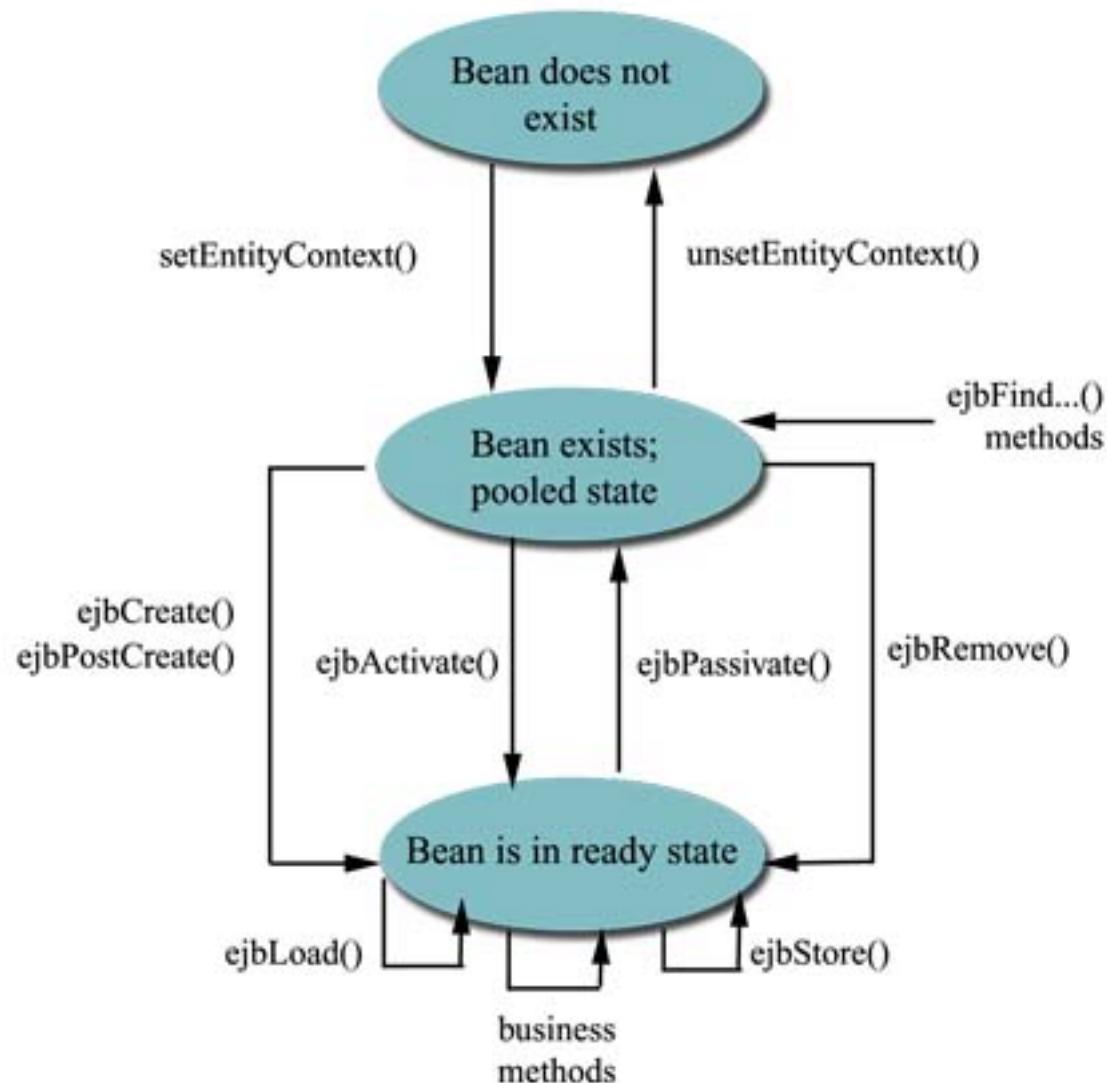
Stateful session bean life cycle



实体Bean生命周期



Entity bean life cycle

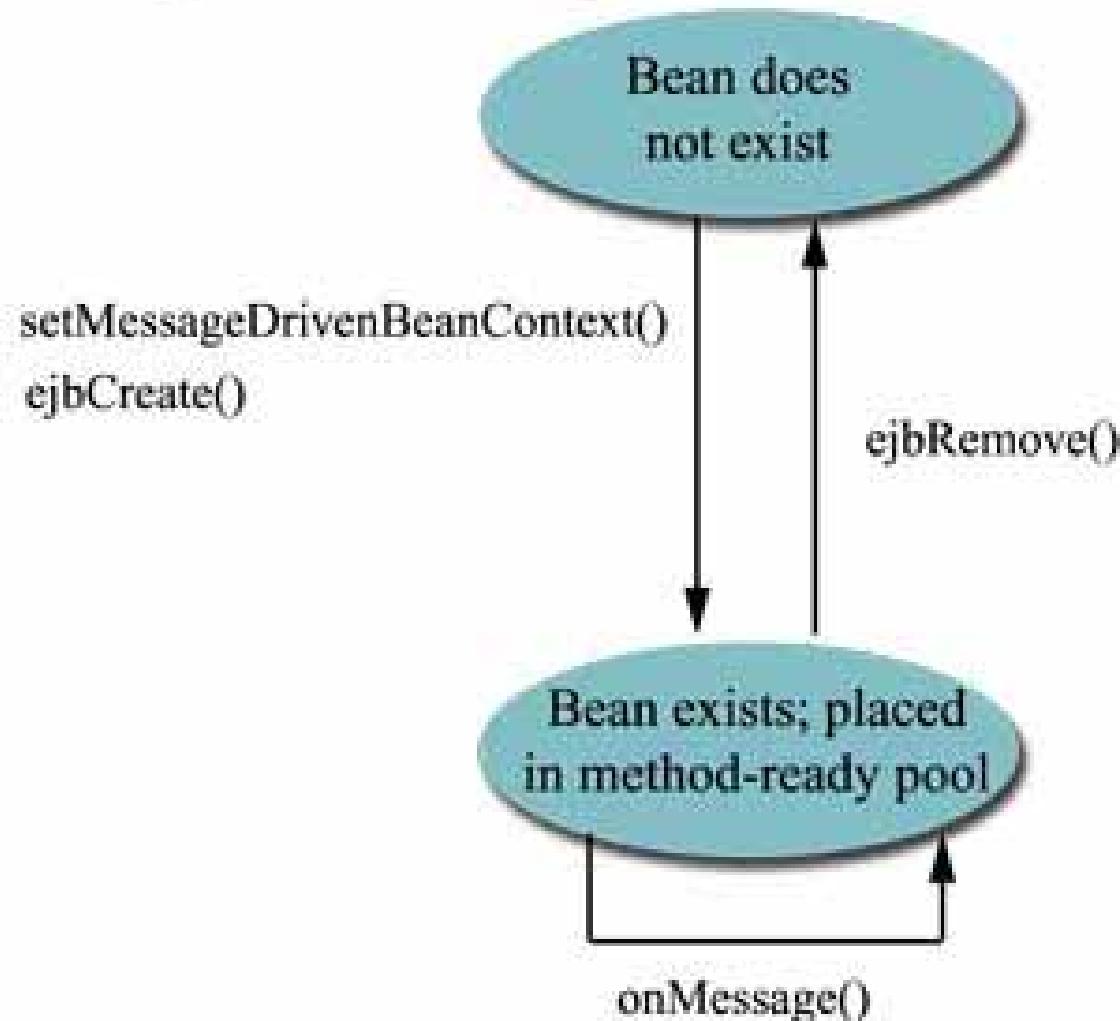


JAVA

消息驱动Bean生命周期



— Message-driven bean life cycle



JAVA

EJB Transactions

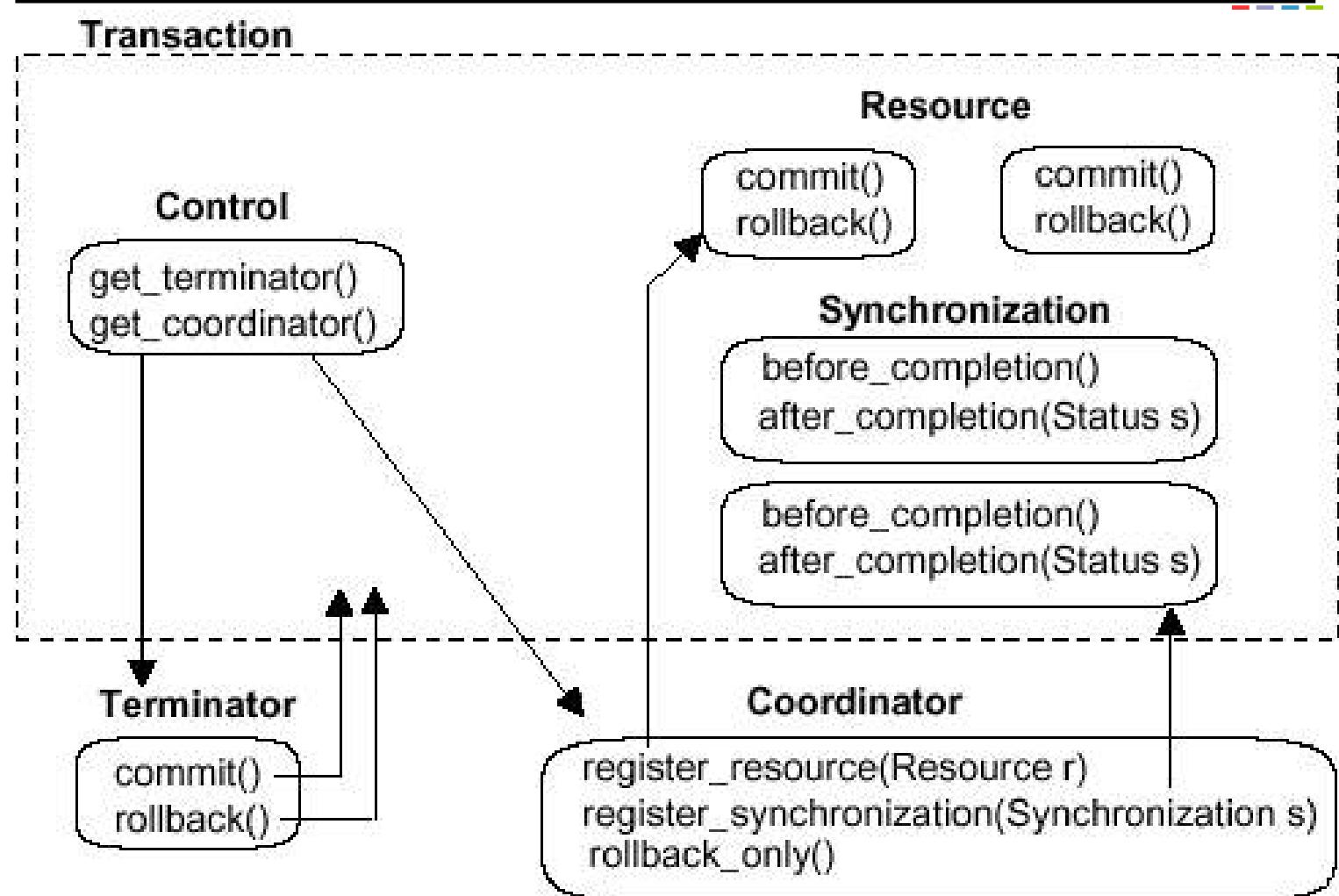


EJB的事务可以由容器管理,也可以由Bean自己管理.



关键字	说明
Never	表示相应的EJB调用不应当参与事务.如果在事务中调用了这个EJB,容器将抛出一个RemoteException的异常
NotSupported	表示不管调用者是否是事务性的,相应的EJB调用都不能参与事务.
Supports	如果调用者正在参与事务,相应的EJB调用也可以参与事务.否则,相应的EJB调用不能参与事务
Mandatory	如果调用者有一个事务,相应的EJB可以参与该事务,否则,容器将发出一个TransactionRequiredException异常信息
Required	如果调用者有一个事务,相应的EJB可以参与该事务。否则,容器将在调用相应的EJB方法之前,开始一个事务。当方法调用完成之后,即提交该事务
RequiresNew	在调用相应的EJB的方法之前,容器先开始一个新的事务。当方法调用返回时即提交这个事务

EJB事务处理过程



EJB安全



EJB的安全策略应在部署描述符中声明，而不是在Bean中硬编码

EJB 2.0规范中，EJB安全管理的三个目标：

- 。将应用程序安全方面的责任从Bean提供者转移到EJB角色上，如：
EJB容器，部署者和系统管理员
- 。Bean提供者在开发时不要硬编码安全策略，而是让应用程序汇编者和部署者设置安全策略。EJB容器实现安全的基础结构
- 。实现使用不同安全机制的多个EJB服务器之间的EJB移植性

ejb-jar.xml



声明安全角色引用

```
<security-role-ref>
    <role-name>Client</role-name>
    <role-link>Client</role-link>
</security-role-ref>
<security-role-ref>
    <role-name>Bank</role-name>
    <role-link>Bank</role-link>
</security-role-ref>
```



访问权限控制



保证只有Administrator角色能访问LogBean

```
<assembly-descriptor>
  <method-permission>
    <role-name>Administrator</role-name>
    <method>
      <ejb-name>LogBean</ejb-name>
      <method-name>*</method-name>
    </method>
  </method-permission>
</assembly-descriptor>
```



一些EJB的设计原则：

- 1: 只要可以,尽量使用CMP
- 2: 写代码时尽量保证对BMP和CMP都支持
- 3: 把ejbStore中的数据库访问减小到最少
- 4: 总是将从lookup和find中获得的引用进行缓存
- 5: 总是使用prepare statements
- 6: 完全关闭所有的Statement

EJB设计模式



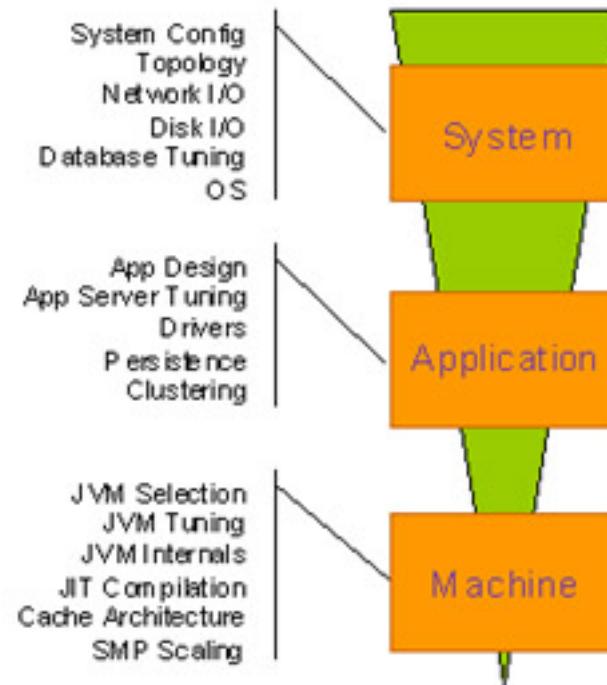
这部分请看J2EE模式一节的介绍。

这里可关注以下几种模式：

- 。数据访问对象 (Data Access Object)
- 。值对象 (Value Object)
- 。业务委托 (Business Delegate)
- 。会话外观 (Session Facade)



EJB性能与扩展性



在系统级、应用级和机器级三个自顶向下层次中的性能优化考虑



提升EJB性能的方法



1. 设计粗粒度(coarse-grained) 的EJB远程(remote)接口
2. 设计粗粒度的EJB远程接口方法
3. 减少JNDI查找次数
4. 利用会话bean的包从一个实体 bean中返回多行数据
5. 采用会话bean来处理大量的数据库操作
6. 尽可能采用CMP(container-managed persistence)容器管理bean代替BMP(bean-managed persistence bean)
7. 恰当的使用实体bean
8. 采用恰当的隔离级别
9. 模拟实际运行环境进行调整
10. 利用工具识别性能瓶颈
11. 调整系统参数
12. 采用群集来满足高负载或考虑升级硬件
13. 将远程方法调用减到最少：local接口的使用



JAVA

Java阵营对EJB的看法



- J2EE项目中只有10%使用EJB
 - “EJB makes Java look bad. ”
- Sun's Java PetStore:
 - “write your data persistence twice” . “bi-modal data access layer”
- IBM Redbook: 不要使用EJB, 用存储过程
- James Gosling, Borland Developer conference, May 2002
 - You have to manage it by ignoring it. The complexity of J2EE is pretty extreme...There's a dirty little secret about J2EE; most people don't need J2EE;



内容回顾



- EJB介绍
 - EJB是什么
 - EJB架构
 - EJB分类
 - EJB容器
 - EJB生命周期管理
 - EJB事务处理
 - EJB安全
 - EJB设计策略
 - EJB设计模式
 - EJB性能与扩展性

总结



- EJB 是主流的组件技术，EJB提供了健壮的、极具竞争力的组件开发模型，是J2EE核心所在
- EJB 的实体Bean看上去并十分成功
- 消息驱动Bean引进了异步处理机制



JAVA

参考资料



- <http://java.sun.com/products/ejb>
sun公司的EJB站点
- <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>