从Java走向Java EE

Java EE
或许并没有您想像的
那么难以入门

吴超◎编著

人民邮电出版社
POSTS & TELECOM PRESS

从Java走向Java EE

吴超◎编著

人民邮电出版社
北京

图书在版编目（CIP）数据

从Java走向Java EE / 吴超编著. —北京：人民邮电出版社，2009.2
ISBN 978-7-115-19192-2

Ⅰ. 从… Ⅱ. 吴… Ⅲ. JAVA语言－程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第176036号

内 容 提 要

Java EE 是目前企业级系统开发的最佳选择之一，其技术本身在不断发展，涌现出各种概念，其繁多的内容让很多初学者望而却步。特别对那些有了 Java 语言基础的读者（大多数计算机专业的学生在学校里只学习 Java 语言本身，其他读者一般也从 Java 语言开始学习）来说，从 Java 迈向 Java EE 是一个艰难的过程。本书就是为了满足这些读者的需要而编写的。

全书深入浅出地介绍 Java EE 各个方面的技术，覆盖从设计开发到测试部署的完整过程，展现 Java EE 的完整图景，通过基础的实例帮助读者上手，并利用 Eclipse 插件等帮助读者掌握利用 Java EE 开发的工具。

本书适合具有 Java 语言知识的读者阅读，尤其适合高等院校的师生及刚刚步入工作岗位的读者阅读。

从 Java 走向 Java EE

◆ 编　　著　吴　超
责任编辑　刘　浩

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京昌平百善印刷厂印刷

◆ 开本：787×1092　1/16
印张：22
字数：540 千字　　2009 年 2 月第 1 版
印数：1－4 000 册　　2009 年 2 月北京第 1 次印刷

ISBN 978-7-115-19192-2/TP

定价：42.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前 言

Java 语言不是很难学习的语言，但 Java EE（旧版本称为 J2EE）对很多人来说却是一个望而生畏的话题。

Sun 公司为 Java 推出了 3 个版本——Java SE，Java ME 和 Java EE，企图在移动设备、桌面设备和企业级应用领域都占得天下，可现实的结果是只有 Java EE 独放异彩，并且带动了 Java 语言的流行。可以说没有 Java EE 在企业级系统领域的成功就没有 Java 语言今天的流行。

所以，对于学习 Java 语言的人来说，掌握了语言仅仅是第一步，进入 Java EE 才是真正的目标所在。

Java EE 的学习范围从狭义来说就是 Sun 关于 Java EE 规范中的内容，比如 Servlet、EJB 等，但是在真实世界中，Java EE 的声音不仅来自 Sun 一家公司，还有其他的竞争者（比如 IBM）以及众多的开源项目（比如 Apache）。这使得 Java EE 领域分外热闹，但是对入门者来说，也显得格外繁杂而不知从何入手。

本书就是为了解决这个问题而编写的，作者的目的是帮助那些已经掌握了 Java 语言基础的人，能通过一本并不太厚的书，了解到 Java EE 开发的基本知识。

Java EE 包容了很广很深的内容，仅一本 Sun 官方的 Java EE 教程就有上千页厚，更别说加上其他产品和各式各样的开源框架了，它们中的每一个都能写成一本厚厚的书。所以说 Java EE 领域是一个广阔的天地，这是其强盛所在，但对入门者来说往往有高山仰止的畏惧感。本书就是帮助您攀上这座高山的第一级阶梯。

从作者使用 Java EE 的经验来说，Java EE 的学习不应该采用课堂上的方式，Java EE 没有一条固定的学习路线。有一些人更关注于实现的技术，比如 EJB 或者消息中间件，他们的目标是成为杰出的 Java EE 软件设计师；有一些人更着重整体解决方案的设计，对系统的接口和结构更感兴趣，那么他们将成为 Java EE 的系统架构师；还有一些人适合较严格的软件过程模型与质量保证，那么成为项目管理者或者测试人员会是一个良好的选择。对于这些不同的目的来说，没有一条万灵的学习道路。最好的办法就是在实践中学，在实践初期先掌握一些基础的知识，并了解一些常用工具的使用；在实践过程中轻装上阵，遇到问题就查文档；在实践之后作回顾，总结整个过程中成功和失败经验，并与项目中其他的人员交流。通过这么一个过程，你将会慢慢找到成为一名优秀 Java EE 开发者的感觉。

所以，在本书中，没有太深的知识，没有庞杂的介绍，我们用简单的描述把 Java EE 的知识展示给读者：这是什么？它是用来干什么的？为什么要采用它？如何上手开发第一个程序？如果读者还是不明白，可以翻看关于这个知识点的实例，这些都是入门实例，简单又不乏说服力，相信对于读者理解概念有很好的帮助。有时一个简单的 HelloWorld，能带给初学

者很多的启发，少走很多弯路。读者可以动手实验这些例子，边练边学。

我们把这本书定位成 Java EE 入门学习的起点，在每章的最后都列出一些资料的链接，鼓励您进一步去寻找更详细更深入的文档和参考资料，读者不一定要在看书之后就立即去浏览，而是当有实际需求时再看，它们将会提供非常有用的信息。

这本书的诞生得到了许多人的帮助，感谢我的导师，感谢关怀支持我的家人。

参加本书资料整理的有吴明晖、柯海丰、侯宏伦、沈明伟、钟晴江、姚慧、张琦、俞雪永、高建礼、胡彧恒、章欣、杨武飞、丁峰、李玉峰、李晓黎、李树有、张巍、张晓冬等，在此表示感谢！由于水平有限，书中难免存在不足，恳请读者批评指正（电子函件：book_better@sina.com）。

本书代码可以在 http://www.ptpress.com.cn/Resources.aspx 处下载。

作　者

2009.01

目 录

第1章 Java EE的基本知识

Java EE（Java Platform Enterprise Edition，Java 企业版）是建立在 Java 平台上的企业级应用的解决方案。Java EE 技术的基础是 Java，它不但拥有 Java SE（Java 标准版）平台的所有功能，同时还提供了对 EJB、Servlet、JSP、XML 等企业级技术的全面支持，形成了一个开发健壮且可移植的企业级应用系统的完整体系结构。Java EE 的出现极大地简化了企业级市场中各种解决方案的开发，并解决了之前非常棘手的系统部署和管理等问题，因此到目前为止，Java EE 已经成为企业级开发的工业标准和首选平台。

Java EE 并非一个单独的产品，而是由 Sun 公司提供的一系列标准组成，这些标准定义了 Java EE 各个组件的接口和其他一些规范。符合这个标准的产品叫做“实现”，很多厂商，包括 Sun 公司在内，都依据 Java EE 的规范提供了 Java EE 的实现或产品，由于这些产品都是符合 Java EE 规范的，所以企业级系统可以在这些不同的产品之间方便地移植。比如 Java EE 定义了应用服务器的标准，在此之上很多厂商开发了自己的产品，如 BEA 的 WebLogic、IBM 的 WebSphere 以及开源的 Jboss，等等，虽然有众多的应用服务器可以选择，但他们是符合 Java EE 规范的，所以应用系统可以轻松地在这些服务器之上部署并移植。

在本章中，我们将会介绍 Java EE 的一些基本概念，帮助读者了解 Java EE 的历史、平台的优良特性，以及平台的主要组成。

1.1 Java EE 的出现及其特点

很多人的印象中 Java 是一款较新的语言和平台，但实际上 Java 早在 20 世纪 90 年代初就开始酝酿了。1991 年 4 月，Sun 公司的 James Gosling 领导的 Green Project 项目开始着力发展一种分布式系统结构，其目的是为家用消费电子产品开发一个分布式代码系统，这样就可以通过电子邮件等方式来控制电冰箱、电视机等家用电器。项目组的成员一开始使用 C++语言来完成这个项目，但很快他们就感到 C++的很多不足，比如 C++太复杂，安全性差等，所以这个团队在 C++的基础上自行开发一种新的语言 Oak，后来它有了一个新的名字——Java。命名为 Java 的原因有好几个版本，有人说是因为项目组的窗外有一棵咖啡树，还有人说是 Java 这个名字是组员们开会的时候喝着 Java 咖啡想到的一杯飘香的咖啡成为了它的标志。

Green 的项目没有获得成功，但 Java 语言适合于网络编程、可移植性强的特性使其得到了 Sun 总裁 McNealy 的赏识。到 1994 年，随着 www 在 Internet 上的迅速发展，Sun 公司正式推出了 Java 语言。Java 语言一推出，便赢得许多著名公司的青睐，得到 Netscape、IBM、

Microsoft 和 Oracle 等大公司的支持，Java 得到迅速推广。到目前为止 Java 语言已经是世界上最流行的语言之一。

Sun 公司针对 Java 在不同领域的应用，将 Java 的标准和类库分为不同的平台版本。目前，Java 平台有 3 个版本。

- 适用于移动设备、小型设备和智能卡的 Java 平台 Micro 版（Java Platform Micro Edition，Java ME）。以前称为 J2ME。Java ME 为在移动设备和嵌入式设备（比如手机、PDA、电视机顶盒和打印机）上运行的应用程序提供一个健壮且灵活的环境。
- 适用于桌面系统的 Java 平台标准版（Java Platform Standard Edition，Java SE）。Java SE 以前称为 J2SE。它用于开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 为 Java EE 提供基础。
- 适用于创建服务器应用程序和服务的 Java 平台企业版（Java Platform Enterprise Edition，Java EE）。Java EE 在之前的版本中被成为 J2EE。

Java 最初是在浏览器和客户端机器中粉墨登场的。当时，很多人质疑它是否适合做服务器端的开发。现在，随着 Java EE 平台在实践中体现的强大特性，Java 被广泛接纳为开发企业级服务器端解决方案的首选平台。本书的读者应该是已经学习过了 Java 语言，并且对 Java SE 平台有一定的了解的，所以在本书中，我们集中介绍 Java EE 平台的内容而不会对虚拟机、Java 语言特性、Java SE 的类库等进行介绍。

Java EE 是建立在 Java SE 平台基础上的企业级平台开发标准，它利用 Java 平台来简化企业解决方案的开发、部署和管理等相关复杂问题。Java EE 不仅拥有 Java 的“编写一次、随处运行”的特性，还提供了方便的数据库存取方法、透明的远程调用、强大的消息机制、健壮的安全模式等特性，Java EE 还带来了包括 EJB（Enterprise JavaBeans）、Java Servlets、JSP（Java Server Pages）在内的众多优秀技术和框架。其最终目的就是使企业开发者大幅缩短开发成本和投放市场的时间。目前看来，经过多年的发展（见图 1-1），Java EE 已经达到了预期的目的，并且获得了巨大的成功。

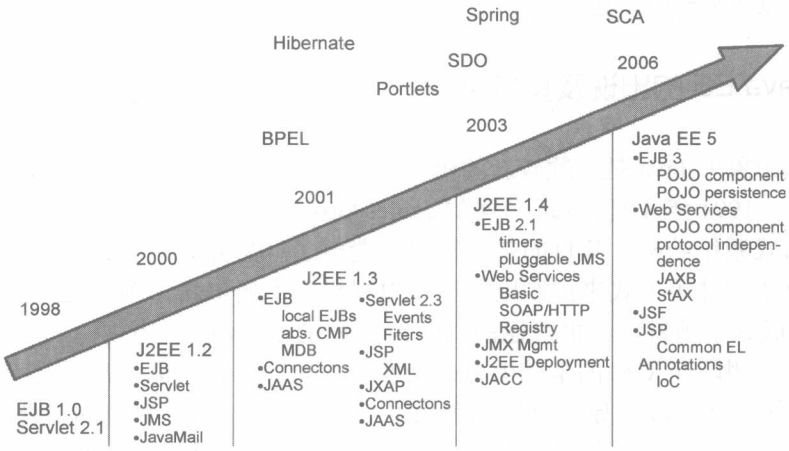

图 1-1　Java EE 的发展

Java EE 为搭建具有可伸缩性、灵活性、易维护性的企业级系统提供了良好的支持。

（1）加快了开发周期和速度。

客户从提出需求到得到响应的时间是个很重要的因素。企业需要更快地开发和部署它们

的企业应用程序，并根据用户的需求方便快速地对系统进行修改、完善和升级。所以这个系统需要简单而又有效率地把原来已有的系统集成起来，并且要具有良好的可扩展性，以迎接将来不断变化的市场需求。Java EE的分层体系结构就能很好地满足这个要求。

同时企业级应用程序需要很快地从原型发展到产品，并且快速地在产品的生命周期中不断完善。而“编写一次，随处运行”的特性，使得Java EE系统能更方便地测试与部署，从而大大提高了开发的效率。

（2）可扩展性。

作为企业级的应用程序需要有很强的扩展性，企业会面临需要将它们的Web应用程序扩大规模的情况，这个时候，如果采用了不合理的技术，就会造成很大的成本和技术风险。而Java EE平台标准保证了企业依据此标准建立的系统能比较方便地进行扩展，Java EE平台采用了多层结构，在各层之间提供了方便的资源管理和服务，比如数据库的连接等，这样开发人员就不用顾忌数据库连接的负载平衡等问题了。同时，可以根据客户端的类型和数量级来确定所用的服务器，并且在需要的时候变更服务器而不需要对系统本身作出太大的改动。

（3）组件模型带来简化的构架。

Java EE平台可以在任何符合标准的服务器上运行，并且基于组件的Java EE开发模型能更容易地将需求确定成功能，并且由于只需要升级必要的组件，所以系统的升级比较方便。

组件能在运行的时候根据配置来连接其他的组件，有了可配置的组件行为，开发人员就不需要重写代码，开发人员通过这些配置来与组件所在的服务器交流，这个过程也可以通过自动化的工具来实现。

（4）与现有系统集成。

企业的数据是多年积累的结果，往往位于一个比较古老的系统中，所以企业级应用系统的开发人员会面对一个重大的考验，就是如何利用并集成那些多年来一直使用的数据系统。为了实现这个目标，Java EE平台通过中间层和后台服务来访问这些现有系统。具体来说，Java EE平台提供了下面这些技术来与现有的信息系统集成。

- JDBC是用Java访问关系数据库的API。
- The Java Transaction API（JTA）是在各个异种的企业信息系统间管理和协调事务的API。
- The Java Naming and Directory Interface（JNDI）是访问现有信息系统的命名和目录服务API。
- The Java Message Service（JMS）是利用IBM MQ或TIBCO Rendezvous这些系统来发送和接收消息的API。
- Java IDL是调用CORBA的API。

（5）服务器、工具和组件的自由选择。

开发企业应用程序，需要根据手头的项目以及技术能力和投资成本，选择并配置应用系统，以得到最优化的效果。Java EE平台为此提供了多种自由的选择，这些选择包括服务器、组件和其他工具等。

（6）更专业化的开发人员。

Java EE的基于组件的开发模式能够根据不同的技能来细化开发人员的职能。所有参与开发过程的人员能更好地发挥他们的特长，JSP模板的设计者就能专心于他的工作，而商业逻

辑的开发者、部署人员也能专注于自己擅长的领域。这个特点对于程序的升级也有好处，一个 Java EE 系统最常改变的一般是前台的用户界面，这样网页设计师就可以着力于页面的设计而不需要编程能力。当然这些职能的划分在不同的开发小组中会有不同，在有些项目中，或许一两个人就能把所有的职能角色都分配掉了。

1.2 Java EE 的分层模型和平台组成

Java EE 平台提供了一种多层的分布式模型，这意味着企业级应用系统的各个部分可以运行在不同的设备上。

1.2.1 Java EE 的分层模型

Java EE 框架定义了客户端层、中间层（可以由一个或多个子层组成）和对现有数据系统提供支持服务的后端层。客户端层提供了对很多种类的客户端的支持，包括企业防火墙内部和外部的客户端。中间层通过在 Web 容器中的 Web 层提供与客户端的交互，并通过 EJB 等进行商业逻辑。后台的企业信息层（The Enterprise Information System，EIS）提供各种 API 以对现有的数据系统提供服务。

图 1-2 显示了 3 个层次的主要组件。

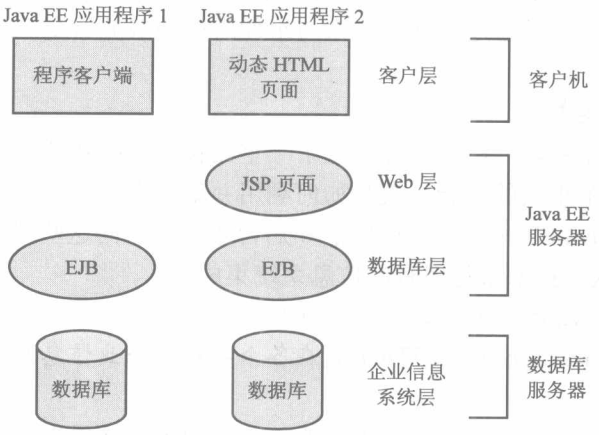

图 1-2　三层结构

图 1-3 所示是一个典型的 Java EE 系统体系图。

Java EE 客户端层支持企业防火墙内部和外部各种类型的客户端。客户端可以是浏览器（以得到纯 HTML 文件、由 JSP 或 Servlet 动态生成的 HTML 页面）或是 Java Applet，也可以是独立的 Java 程序。当然，当大家提到 Java EE 客户端时首先指的还是 Web 方式的。

为了提供比较复杂的用户交互，在客户端层需要直接提供一些功能性的服务，这通常是利用 Java Bean 组件来实现向中间层请求服务的。这些提供服务的客户端 Java Bean 通常以 Java Applet 形式自动下载到用户的浏览器。这些客户端的 bean 可以看作是用 Java 写的独立程序，它们的任务就是提供给客户端必要的服务。针对不同的平台，这些 bean 的提供者需要编写不同的安装文件并让它们能够自动下载到用户所在的平台，虽然这些 java bean 本身是和平台无关的，但是安装文件的代码一般都是和操作系统相关的。

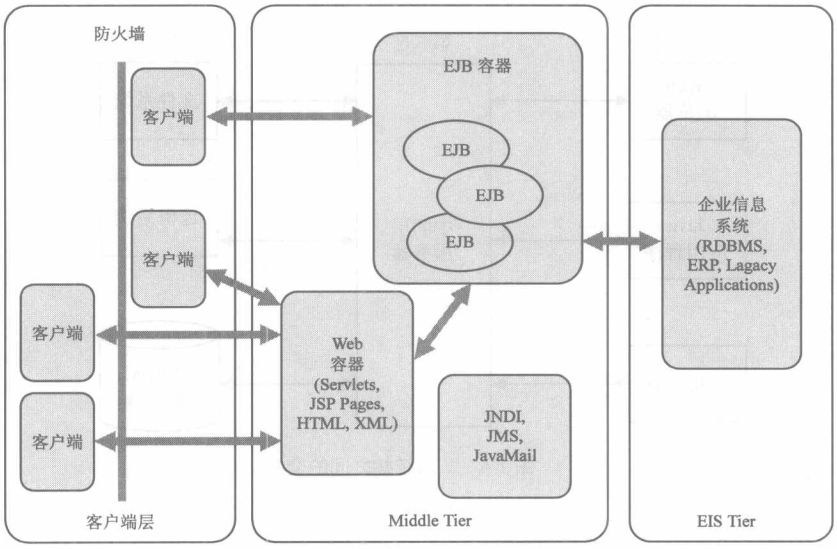

图 1-3　Java EE 体系结构

实际上，客户端也完全可以采用 Java 语言之外的其他语言编写，比如用 Visual Basic 写的程序同样可以访问用 servlet 实现的中间层——由于 servlet 接受标准的 HTTP 请求，所以可以用任何平台上的任何语言编写的程序来得到中间层的服务。

在多层结构（或一般三层结构）中，中间层扮演了关键的角色。它接收客户端的请求，把客户端从复杂的与数据库等后台的交互中解脱出来，这样客户端就不再需要作数据库查询，执行复杂的商业逻辑或连接到古老的 legacy 系统——中间层会替它们完成这些工作。

中间层的任务是执行商业逻辑，比如一个信用卡网站，所有业务相关的操作都由中间层完成。同时中间层也需要提供一些系统级的服务，这些服务包括：

- 提供客户端或到后台系统的远程访问能力；
- 线程和事务管理；
- 安全性；
- 资源池。

正是由于中间层提供了这些服务，才为瘦客户端的应用提供了可能，对用户来说他们能更轻巧、快速和简单地访问 Web 应用程序，而企业也节省了改造原有后台系统的成本。

图 1-4 所示是三层结构之间交互的示意图。

1.2.2　Java EE 的结构变形

用 Java EE 平台开发的 Web 应用程序，可以根据具体的需求和规模，采用不同的体系结构，主要包括以下几种类型。

（1）由浏览器和应用程序共同组成客户端，由 Web 组件和 EJB 组成中间层，如图 1-5 所示。

（2）由浏览器组成客户端，由 Web 组件和 EJB 组成中间层，如图 1-6 所示。

（3）由客户端的应用程序直接和 Web 组件、EJB 或后台数据系统交互的结构，如图 1-7 所示。

（4）客户端直接与 EJB 交互的结构，如图 1-8 所示。

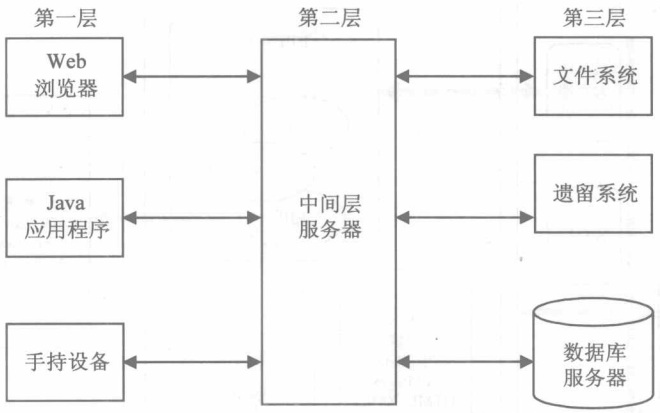

图 1-4　三层结构之间的交互

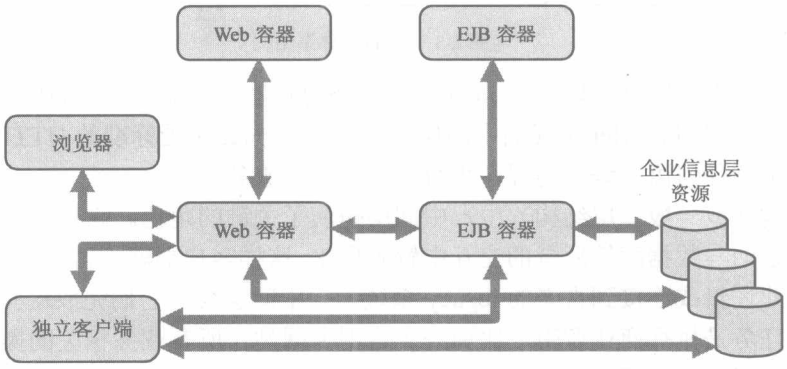

图 1-5　由浏览器和应用程序共同组成客户端，由 Web 组件和 EJB 组成中间层

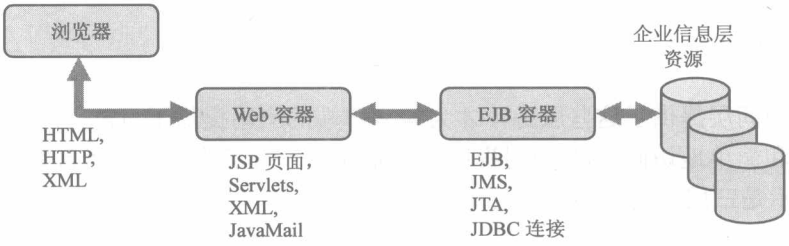

图 1-6　由浏览器组成客户端，由 Web 组件和 EJB 组成中间层

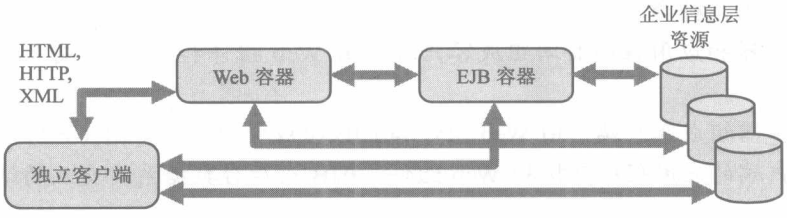

图 1-7　由客户端的应用程序直接和 Web 组件、EJB 或后台数据系统交互的结构

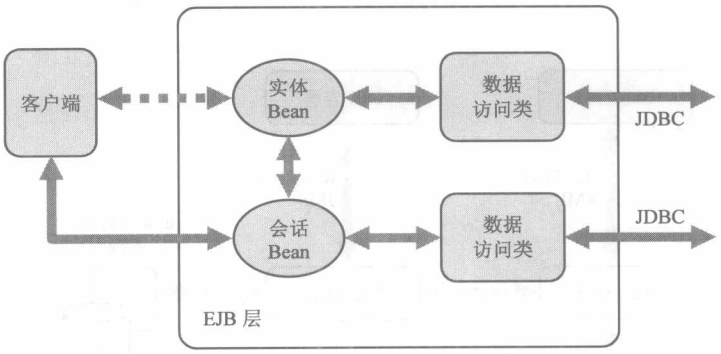

图 1-8　客户端直接与 EJB 交互的结构

（5）仅有 Web 组件的结构，如图 1-9 所示。

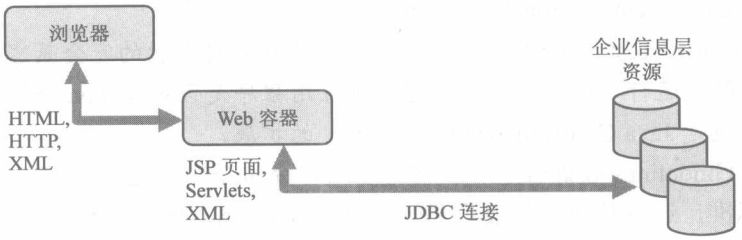

图 1-9　仅有 Web 组件

（6）采用其他数据访问方式的结构，如图 1-10 所示。

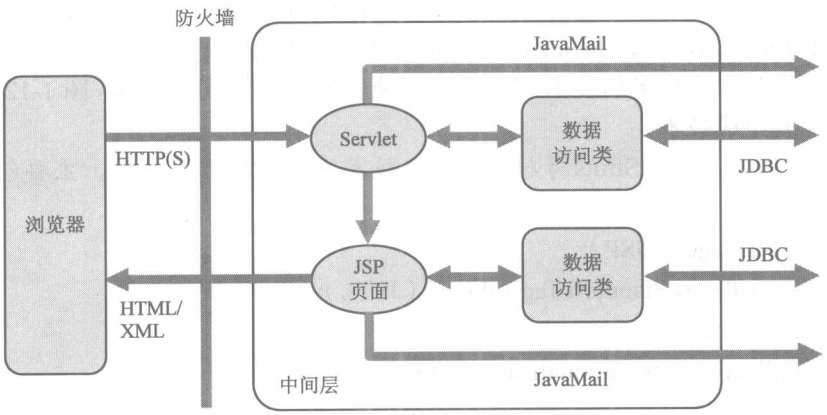

图 1-10　采用其他数据访问方式

（7）没有纯粹的客户端的结构，如图 1-11 所示。

可以看出，Java EE 灵活的结构变形使其具备很强的灵活性。

1.2.3　Java EE 平台的组成

Java EE 平台由 4 个关键部分定义：Specification、Reference implementation、Compatibility test suite 和 Java EE Blueprints design guidelines。

（1）Specification 定义了 Java EE 平台如何工作，是否需要包含应用服务器、数据库或

其他。

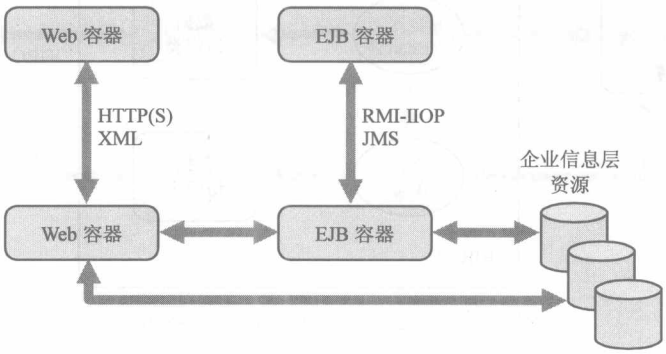

图 1-11　没有纯粹的客户端的结构

（2）Reference Implementation 提供了比较的标准。

（3）Compatibility Test Suite 保证 Java EE 厂商的产品对平台的完全兼容性以确保“一次编写，随处运行”的可移植性，它验证应用程序是否符合 Java EE 的实现标准。它可以看做是 Java Conformance Kit（JCK）的扩展。

（4）Design Guidelines 告诉程序员如何将各个部分组合到一起。

除了这些标准，还有其他一些很重要的补充，以对 Java EE 平台进行支持。这些补充包括 Java EE SDK 等。Java EE SDK 的出现有好几个目标：首先它提供了一个可操作的 Java EE 平台的定义，这样厂商就可以通过这个标准来决定它们产品所必需的功能实现；它也可以帮助开发人员判断他们的程序是否符合可移植性的标准；其次，Java EE SDK 作为开发者社区可以免费得到的 Java EE 平台实现，有助于加速 Java EE 标准的推广。虽然它不是商业化产品并且禁止用做商业用途，但是 Java EE SDK 是可免费得到的。

Java EE 平台包含了众多的技术，在这里可以把它们大致划分成 4 类，图 1-12 显示了 Java EE 平台中的主要组成技术。

（1）JSP、JSTL、JSF、Struts 等技术。这些技术关注 Web 层的组件，主要包括：

- Java Servlet;
- JavaServer Pages（JSP）;
- JavaServer Pages Standard Tag Library（JSTL）;
- JavaServer Faces;
- Web application internationalization and localization;
- Struts。

（2）EJB 等技术。Enterprise JavaBeans 技术是实现业务逻辑的组件，主要有以下类型：

- Session beans;
- Entity beans;
- Message-driven beans。

（3）JAXP、JAX-RPC、SAAJ 等技术。这方面技术的主要作用是处理 XML 文档和实现 web services 组件，主要包括：

- The Java API for XML Processing（JAXP）;
- The Java API for XML-based RPC（JAX-RPC）;