

J2EE权威指南

Designing Enterprise
Applications with the Java™ 2
Platform, Enterprise Edition

涂翔云 姚欣 李春阳 译

The Java Series
Enterprise Edition



... from the Source

中国电力出版社

内 容 提 要

本书介绍了用 J2EE (Java 2 平台企业版) 设计企业级应用程序的标准方法。它是由 Sun 公司 J2EE 项目组的核心成员集体撰写的, 其权威性毋庸置疑。

本书讨论了基于 Java servlet 和 JSP 的 Web 客户、使用 EJB 技术的中间层解决方案, 以及基于 JDBC 的后端连接。通过对客户层、Web 层、EJB 层、企业信息系统层的讨论, 指导读者在多层企业级应用程序的各层之间分配功能。并且还讨论了设计 Web 应用程序时遇到的安全、部署、事务管理等关键问题, 同时给出了一个生动的电子商务应用程序的例子。

本书适合 IT 管理者、系统工程师、使用 J2EE 平台的企业级应用程序开发者以及 J2EE 产品供应商阅读使用。

图书在版编目 (CIP) 数据

J2EE 权威指南/ (美) 卡赛姆等编; 涂翔云等译. —北京:
中国电力出版社, 2002.6

ISBN 7-5083-1095-0

I. J... II. ①卡...②涂... III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 036835 号

著作权合同登记号 图字: 01-2002-0704 号

本书英文版原名: Designing Enterprise Applications with the
Java™ 2 Platform, Enterprise Edition

Published by arrangement with Addison Wesley Longman, Inc.

All rights reserved.

本书由美国 Addison Wesley Longman 公司授权出版

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2002 年 7 月第一版 2002 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 18.5 印张 407 千字

定价 35.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

序

对于 Java™ 平台，我们可以将其想像成一个把门把手连接到电灯开关上的东西——即由智能化的门把手到智能化的电灯开关。虽然看起来是如此简单，但它却已广泛地应用于建立大型的服务器端应用程序中，这些应用程序通常运行于各地的大型计算机上。作为一项伟大的发明，它目前正在以一种甚至包括 Java 编程语言的创始人 James Gosling 都不能想像的方式被广泛地使用着。这也正是该项发明的伟大之处。

其实回顾一下，Java 平台在服务器上的成功似乎是必然的。毕竟，该平台提供了将 Internet 由一个单纯的信息发布媒体向事务处理平台转型所最需要的东西。在所有运行 Internet 应用程序的各种服务器上都提供了 Java 平台。Java 程序具有“编写一次，随处运行”的特性，且能够被迅速地测试和部署，正是这些特征，使得工程师们在 Java 平台上能够以成倍的效率编写代码。但在最合适的时间做最合适的事情并不意味着全部，我们还需要更多的东西：采用一种与大多数人都不同视角来看待问题的技术引路人，以及期待以一种全新的方式工作从而使得各种美好的想法能够成为现实的商务领导者。其直接结果便是导致具有跨平台一致性的产品开发缩短到一个令人不可思议的周期。

本人于 1995 年下半年加入了 Sun 公司的 JavaSoft 分部，受 Eric Schmidt 及 Bill Joy 之聘，开始领导一小批工程师及市场营销人员从事工作。我们力争以尽可能快的速度成长，但还是难以跟上 Java 平台以及围绕它所建立的业界联盟已获得的成功的步伐。在那个时候，普遍的焦点还正着重于运行于浏览器上的 applet，Java 1.0 版本也尚未发布，这个时候我们有两个聪明的工程师 Rick Cattell 及 Graham Hamilton 开始具有一种完全不同的想法。当时他们正在构思服务器上的 Java 运行时环境，乃至主框架。正是他们使得目前独一无二的 JDBC™ 技术成为 Java 平台的一个重要补充。后来，有许多天才的工程师们继续了 Rick 和 Graham 的工作，但当时正是他们做了一个良好的开端。我十分庆幸当时在改良平台时聪明地听取了他们的意见。

直到最近，才需要大量尖端科学家来创建工业上所要求的应用程序，即那些利用业已存在的资源（如数据库、事务处理系统、存货系统、发票系统及信用系统）并采取新的方式来处理 Internet 上事务的应用程序，这些应用程序通常需要规划数以千计，甚至百万计用户。它们必须要与大量的不可替代的旧技术打交道，并且时间也颇为紧迫。有能力建立这种应用程序的工程师只是少数的尖端科学家。不过，庆幸的是，Rick 和 Graham 看到了一种处理方法，即通过将尖端科学植入 Java 平台，从而带来了符合工业标准的可移植性和一致性，同时也能适应业已存在的系统，并通过服务端复杂的编程细节的自动化而使得应用程序开发的复杂性大为降低。这些想法便是构建 Java 2 平台企业版的

坚实基础。

JDBC 的出现是一个巨大的革命，在 Java 社区发布它之后，众多有关所有重要数据库的驱动程序便迅速出现在市场上，采用 JDBC 的应用程序也有如雨后春笋般出现。JDBC 的成功直接导致了其他中间件及数据库适配器工程的繁衍——如提供对命名和目录服务的一致性访问的 Java 命名和目录接口（Java Naming and Directory Interface™）API，提供对数据和事件的异步交换的 Java 消息服务（Java Message Service），用于事务处理的 Java 事务（Java Transaction）API 及 Java 事务服务（Java Transaction Service），用于创建动态 Web 页面的 JSP™（Java Server Pages）技术，用来开发面向 XML 的应用程序的 Java XML 和 Enterprise JavaBeans™ 体系结构，即用于服务端应用程序的一种组件模型。所有这些都采用一种由 Rick 和 Graham 创建的进程来协同进行开发，这一进程在后来被修正和正式化，成为 Java Community Process。

1996 年出现的 JDBC 不久便发展成为巨大数量的工具包，且均附带有自己的首字母缩略词和发行计划。这对于那些与各种“J*'s”接触不多的人来说也许会有些困惑。当销售商在其说明完成之前宣布对 Enterprise JavaBeans™ 1.0 提供支持时，我们便意识到是该使得这一目前十分成功的文档包变得更易理解和管理的时候了。

J2EE™ 平台（Java 2 平台企业版）将所有这些组件集成到了一起。J2EE 平台由四个关键部分进行定义：规范部分、参考实现、兼容性测试包及 J2EE Blueprints 设计向导。规范定义了 J2EE 平台将如何工作，无论其是否包含在一台应用程序服务器上、一个数据库中或任何其他的地方。参考实现对于与 J2EE 平台的结合试验来说是十分有用的，并且它给出了一个用于比较的标准。兼容性测试包确保了 J2EE 销售商实现平台的完整版本，从而确保了真正的“编写一次，随处运行”，另外，设计向导向开发人员说明了所有这些部分是如何集成到一起以建立起完整的应用程序的。核心的 J2EE 规范发布于《Java 2 Platform, Enterprise Edition: Platform and Components Specifications》（也可以从 Addison-Wesley 获得），而且可以从 <http://java.sun.com/j2ee/> 上获得有关说明的补充。用于创建本书中所包含实例的参考实现则可从 Sun Microsystems Java Software Web 站点 <http://java.sun.com/j2ee/download.html> 处获得。

许多人对 J2EE 做出了贡献。Mala Chandra 参加了该组并负责领导服务端事务，后迅速成为最主要的改革者之一。她高度的工作热忱及决策带领项目组克服了重重困难。Jeff Jackson、Connie Weiss、Karen Tegan 和 Tom Kincaid 提供了附加的工程管理。Mark Hapner、Vlada Matena、Bill Shannon、Shel Finkelstein、Eduardo Pelegri-Llopart、Larry Cable 和 Nick Kassem 等则提供了技术上的领导。Bill Roth、Gina Centoni、George Paolini 和 Kathy Knutsen 使得 Sun 团队保持与整个工业的紧密相连。

众多公司都对创建这一全新的服务器平台提供了极大的帮助。他们无论大小，无论新老，都紧密地团结在一起。BEA Systems、IBM、Oracle 及 Sun 公司是其中业务全面的代表，他们从来不会感觉到孤单。许多公司还提供了他们最为资深的设计师、工程师及最富经验的管理人员以迅速创建一个通用平台，并创建全新的标准以利于开发、伸缩

和应用。他们将这些最新的 Java 平台技术植入到他们老的或新的产品之中。来自硅谷的应用最为广泛的数据库及最新的开发工具开始共享相同的接口。我们大家都是其中的受益者。

在很多方面，本书代表了这些协同努力的结晶，有效地论述了这一新平台是如何简化和完善新一代企业级应用程序的设计、开发和部署的。

Jon Kannegaard

Sun 公司副董事长，常务董事

前言

本书介绍了一种用 J2EE 来设计多层企业级应用程序的标准方法。本书的目的不在于介绍如何用具体的 J2EE 技术开发应用程序，而是着眼于指导读者如何在层之间分配应用程序功能，以及如何在每一层的设计选项之间做出选择。

本书介绍了建立 J2EE 应用程序所使用的技术和原则，以及应用程序所采用的特定方法。处理好特性与通用原则之间的关系从来就不是一件容易的事情，我们希望所给出的原则既与书中的示例应用程序一致，又是对示例应用程序有益的补充。

本书非常适用于 IT 管理者、系统工程师、考虑向 J2EE 平台移植或使用 J2EE 平台的企业级应用程序开发者及提供 J2EE 产品的供应商。

本书的结构

本书包含以下章节：

- 第 1 章，“概论”，讨论了在企业级应用程序设计中的难题以及 J2EE 平台如何解决这些问题。同时也介绍了 J2EE 平台所支持的一些应用程序。
- 第 2 章，“J2EE 平台技术”，概述了 J2EE 平台所支持的组件、服务和通信技术。
- 第 3 章，“客户层”，给出了 J2EE 客户实现选项，并指导如何在这些选项中做出选择。
- 第 4 章，“Web 层”，介绍了可用于 Web 层开发的技术。还包括使用 J2EE Web 组件的指导和技巧，介绍了几个 Web 应用程序结构。
- 第 5 章，“EJB 层”，讨论了 J2EE 平台的 EJB 层的功能以及实现商务逻辑的设计选择。
- 第 6 章，“企业信息系统层”，介绍了访问企业信息系统的推荐方法，及如何配置 J2EE 组件来进行访问。
- 第 7 章，“包装与部署”，介绍了 J2EE 平台所提供的对 J2EE 应用程序进行包装和部署的功能，并给出一些使用这些功能的启示和经验，也给部署工具的提供者提出了一些建议。
- 第 8 章，“事务管理”，介绍了 J2EE 平台所提供的事务服务，并给出一些如何更好地使用这些服务的建议。
- 第 9 章，“安全”，介绍了 J2EE 安全模型到企业计算环境和基础结构的映射。

- 第 10 章, “示例应用程序”, 给出一个示例阐明 J2EE 编程模型, 深入地描述了多层 J2EE 应用程序。
- “词汇表”, 列出了本书中用到的一些词和短语及它们的定义。

如何获取示例应用程序

可以从以下网址下载本书中的示例应用程序:

<http://java.sun.com/j2ee/download.html>

此示例应用程序需要在 J2EE v1.2 兼容平台上运行。从本示例应用程序的下载网页也可以下载到 sun 公司的 J2EE SDK, 它是 J2EE v1.2 平台的免费实现。

相关信息

与 J2EE 相关的一些文档可以在这个网址获得:

<http://java.sun.com/j2ee/docs.html>

有关如何使用 J2EE SDK 构建多层企业级应用程序, 参看《J2EE Developer's Guide》, 可在以下网址获得:

<http://java.sun.com/j2ee/j2sdkee/techdocs/index.html>

本书提到的一些 J2EE 技术在它们的规范中有介绍:

- Java™ 2 平台企业版规范, 1.2 版 (J2EE 规范), ©1999, Sun 公司。可在 <http://java.sun.com/j2ee/download.html> 下载。
- Java™ 2 平台标准版, 1.2.2 版 (J2SE 规范), ©1993~1999, Sun 公司。可在 <http://java.sun.com/products/jdk/1.2/docs/api/index.html> 下载。
- Java™ Servlet 规范, 2.2 版 (Servlet 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/servlet> 下载。
- JavaServer Pages™ 规范, 1.1 版 (JSP 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/jsp> 下载。
- Enterprise JavaBeans™ 规范, 1.1 版 (EJB 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/ejb> 下载。
- JDBC™ 2.0 API (JDBC 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/jdbc> 下载。
- JDBC™ 2.0 标准扩展 API (JDBC 扩展规范), ©1998、1999, Sun 公司。可在

<http://java.sun.com/products/jdbc>下载。

- Java™ 事务 API, 1.0.1 版 (JTA 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/jta> 下载。
- Java™ 事务服务, 0.95 版 (JTS 规范), ©1997~1999, Sun 公司。可在 <http://java.sun.com/products/jts> 下载。
- Java 命名和目录接口, 1.2 版 (JNDI 规范), ©1998、1999, Sun 公司。可在 <http://java.sun.com/products/jndi> 下载。
- Java IDL, ©1993~1999, Sun 公司。可在 <http://java.sun.com/products/jdk/1.2/docs/guide/idl/index.html> 下载。
- RMI over IIOP 1.0.1, 可在 <http://java.sun.com/products/rmi-iiop> 下载。
- Java™ 消息服务, 1.0.2 版, (JMS 规范), ©1998, Sun 公司。可在 <http://java.sun.com/products/jms> 下载。
- JavaMail™ API 设计规范, 1.1 版 (JavaMail 规范), ©1998, Sun 公司。可在 <http://java.sun.com/products/javamail> 下载。
- JavaBeans™ 活动构架规范, 1.0.1 版 (JAF 规范), ©1998, Sun 公司。可在 <http://java.sun.com/beans/glasgow/jaf.html> 下载。

致谢

本书是许多人努力的结果。

每个企业小组成员主要负责某一章节, 并对其他章节做出了很大贡献。另外, Danny Coward 负责章节分配草案。

J2EE 规范的作者以及参考实现开发者也在许多方面对 J2EE 编程模型的开发提供了有用的帮助。

我们也感谢 Rick Cattell、Bill Shannon、Mark Hapner、John Crupi、Sean Brydon 以及看过我们的手稿并提出建议的人们。

感谢 Jim Inscore 和 Stephanie Bodoff 的编辑。

目 录

序 前 言

第 1 章 概论	1
1.1 企业级应用程序开发的挑战	2
1.2 用于企业解决方案的平台	5
1.3 J2EE 应用程序方案	11
1.4 总结	18
第 2 章 J2EE 平台技术	21
2.1 组件技术	21
2.2 平台的角色	25
2.3 平台的服务	27
2.4 服务技术	32
2.5 通信技术	34
2.6 总结	38
第 3 章 客户层	41
3.1 需求和限制	42
3.2 客户选择简述	44
3.3 Web 客户	45
3.4 EJB 客户	50
3.5 企业信息系统客户	55
3.6 面向多种类型客户的设计	55
3.7 总结	59
第 4 章 Web 层	63
4.1 Web 应用程序和 Web 容器	63
4.2 动态内容的创建	64
4.3 servlet 和 JSP 页面	67
4.4 JSP 页面的设计	71
4.5 国际化和本地化	74
4.6 应用程序设计	80

4.7	应用程序的移植	88
4.8	总结	90
第 5 章	EJB 层	93
5.1	商务逻辑	93
5.2	作为 J2EE 商务对象的企业 bean	96
5.3	实体 bean	99
5.4	会话 bean	102
5.5	设计的指导方针	106
5.6	总结	112
第 6 章	企业信息系统层	115
6.1	企业信息系统的性能及局限性	116
6.2	企业信息系统集成方案	116
6.3	关系数据库管理系统访问	119
6.4	其他企业信息系统访问	119
6.5	应用程序组件提供者的任务	120
6.6	应用程序编程模型	120
6.7	编程访问数据及函数	121
6.8	连接	124
6.9	安全	127
6.10	J2EE 连接器体系结构	130
6.11	总结	131
第 7 章	包装与部署	133
7.1	角色与任务	133
7.2	包装 J2EE 应用程序	136
7.3	部署描述符	141
7.4	部署工具	151
7.5	总结	156
第 8 章	事务管理	159
8.1	事务特性	159
8.2	J2EE 平台事务	160
8.3	方案	160
8.4	JTA 事务	162
8.5	applet 和应用程序客户端中的事务	163

8.6	Web 组件中的事务	163
8.7	企业 bean 中的事务	164
8.8	企业信息系统中的事务处理	167
8.9	总结	171
第 9 章	安全	173
9.1	安全威胁及机制	173
9.2	鉴别	174
9.3	授权	181
9.4	消息保护	188
9.5	审核	190
9.6	总结	191
第 10 章	示例应用程序	193
10.1	应用程序的功能	193
10.2	应用程序的体系结构	198
10.3	视图	203
10.4	模型	218
10.5	实现	223
10.6	控制器	225
10.7	MVC 总结	240
10.8	无状态服务	240
10.9	部署	243
10.10	事务	244
10.11	安全	244
10.12	总结	252
词汇表		255
后序		

第 1 章

概 论

Nicholas kassem

Internet 和 WWW 为企业提供了构建信息化经济的坚实基础和平台。在这种全新的经济模式下，信息与产品及服务有着同样重要的战略意义，成为市场中一个不可或缺的主体。信息化经济的出现对当今企业来说是一场新的挑战，企业将不得不对自己的商务运作模式重新进行审视和思考。

关于信息化经济的未来发展方向，人们做出了种种大胆的预言。既有对可无限扩展的动态的新型商务、工业及财务环境光辉前景的展望，也有对人们过度期望及不可承受的过量增长的隐忧。无论预言怎样，现实问题是：企业^①总是试图采用各种可能的合理手段，包括采纳最新的技术，以取得竞争优势。这其实是一个很自然的生存本能：任何一个有活力的企事业单位，不管是盈利公司、非盈利组织，还是政府机构，都在不断地寻找各种方式，以使自己与时俱进。在革新者致力于将新的挑战转变成商业上的成功的同时，那些过分自满的机构则往往难以逃脱倒闭的厄运。

在信息化经济里，信息资产对于一个机构来说有着深远的战略意义。其中，将其资本化的能力是成功的关键。一个成功的企业将通过把信息推进到市场，从而增长自己的生产力，来实现这一目标。

所有这些看来都会是新的挑战，在很多方面，Internet 和 WWW 只是使长期以来 IT 专业人士一直面对的挑战——“要求对信息资产反应迅速地进行管理”更加严峻。对于这一要求做出的第一反应应该是确保所有关键性的商务功能都能够进行计算机化管理。

① 本书中企业的原文是 enterprise，表示盈利机构，但英文原意准确地说，是企事业单位，这从下一句话可以看出来。所以 enterprise application 的合理译文是企业级应用（程序、系统）。——译者注

近来,人们已开始进一步要求实现这些系统在更高层次上的集成,并增加从分散的数据源整合数据的能力,从而服务于特定的战略需要。同时,各种公司兼并重组和结盟,也进一步推动了公司整合这些信息的需要。

分布式定制应用程序就是企业用来传递商品形式的信息的一些程序包。这些定制应用系统将使公司的信息资产增值并从中抽取价值。通过它们 IT 公司可以按用户的需求定制特定的功能。通过使信息在公司内部可用,战略性地增强了管理和规划进程。同时这些应用程序还有选择地向企业外部发布信息资源,增强了与客户、供应商以及企业自身的交互。

在信息化经济的竞争性环境里,响应时间对于定制应用程序在企业中的价值而言是一个关键方面。企业需要快速开发和部署这些定制应用程序,并且能够很方便地完善和增强它们的功能以适应需要。他们需要采用一些方法来简单而有效地将这些应用程序集成到现存的企业信息系统中去,并方便地规划它们以适应日益变化的需要。所有这些因素都影响了企业面对竞争性环境时快速响应的能力。

J2EE™ 的目标就是定义一个功能标准,以帮助适应这些挑战,从而提高企业在信息化经济中的竞争力。J2EE 平台通过基于组件的应用程序模型大大简化了开发过程,同时还支持使用范围广泛的各种新技术开发分布式应用程序。J2EE 模型所支持的应用程序的范围,从传统的企业内联网上的客户-服务器应用程序,到 Internet 上的电子商务 Web 站点。

本书描述了 J2EE™ Blueprints 编程模型,希望能为企业级应用程序的开发人员提供面临信息化经济挑战下的战略性认识,同时还探讨了 J2EE 平台是如何支持定制应用程序,以适应广泛的应用要求的。这些讨论背后的主题其实是 J2EE 平台提供了一个简单而统一的标准,从而为企业提供了更多超越传统边界设计商务信息系统的机会,同时也避免了内在的任务风险。

本书从合理的视角对企业级平台进行了考察,并对如何使用 J2EE 平台所提供的技术,最有效地划分应用程序的功能提出了建议。目的在于将采用 J2EE 设计和开发多层应用程序这一问题,划分为好管理的多个组成部分,然后将合适的技术应用到各个部分,最终得到维护性和伸缩性更好的解决方案。在处理上,本书做了一定程度的简化,不过分追究琐碎的细节,从而将主要精力放在核心的 J2EE 主题上。读者需要注意的是:对于本书的论述,不应纯粹地理解为严格意义上的要求,更适合作为一些建议或者忠告来接受。

1.1 企业级应用程序开发的挑战

目前,时间因素已经成为采纳新技术的一个关键性要素,而虚拟的信息驱动的商务

模型本身特有的迅猛步伐，却要求我们更加强调响应时间。为了利用 Internet 经济，所需要的不仅仅只是将企业级系统推向各种用户渠道，同时也需要及时反复地进行，要经常更新各种信息及服务。因此，最主要的挑战来自于怎样在充分维持和利用现有商务系统的基础上，跟上 Internet 激烈竞争的步伐。在这种环境下，时间因素对于获得和维持竞争力尤为重要。有许多种因素能够加强或阻碍企业迅速开发定制企业级应用程序并使其价值在整个生存期内达到最大化的能力。

1.1.1 编程生产力

毋庸置疑，如果要在信息化经济中取得成功，开发和部署应用程序的能力将是一个关键性的要素。应用程序从初具雏形到最终成为产品，应该是一个较为快速的过程，并且在它们进行部署之后继续发展。

这样，生产力对于反应灵敏的应用程序开发而言就显得尤为重要。有必要为应用程序开发团队提供一个访问为多层应用程序所要求的服务的标准方法以及支持各种不同的客户的标准方式，这将充分有助于迅速做出反应并保持灵活性。

Internet 及其他一些分布式计算应用程序中的一个不稳定因素便是当前在技术和编程模型之间存在的分歧。诸如 HTML 和 CGI 等历史性的（Web 用语）技术已经为分配动态内容提供了一种机制，而像事务处理器和数据库管理系统等后端系统则提供了对所给操作数据的控制性访问。这些技术提供了多样化的编程模型，其中一些基于定义良好的标准，其他一些基于更为特定的标准，还有一些则仍然停留在私有的层次上。

如果没有一个统一的应用程序模型，对于开发团队来说，要做到就应用要求有效地交流将十分困难。这样将会导致层次化的应用程序变得更为复杂，而且将这些技术完美地整合到一起也将变得困难，也就难以有效地分配劳动量。例如，CGI 开发要求代码编写者对将要显示在动态 Web 页面上的内容和布局都进行定义。

就应用程序的开发时间而言，另一个复杂的因素是客户选择。在我们能够通过静态或动态产生的 HTML 页面将许多应用程序分发给 Web 浏览器客户的同时，另外一些应用程序可能需要支持某种特定的客户，或者同时支持不同种类的用户。因此，编程模型需要支持多种客户配置，同时将对基本的应用程序架构或应用程序的核心商务逻辑的影响最小化。

1.1.2 对要求的响应

让我们设想一个砖瓦商业，如果它要试图将客户群增加 10% 的话，那么在重扩店面、建造新的仓库等上面需要花费多少时间和精力才能达到目的？一个事实便是：经常

性的重复劳动将严重地影响他们对所要吸引的客户群的服务质量。

对于信息化经济的商务来说，现实也是一样，应用程序的易于伸缩及自动容纳期望或不期望的增长的能力对于实现最终目标将是一个关键。所有在伸缩时要求重新构建或重新部署的系统都将阻碍增长并降低公司所期望达到的性能。

为了有效地进行伸缩，需要将系统设计为易于处理多用户交互的类型。它们需要提供种种机制来保证对系统资源及服务进行有效的管理，例如数据库之间的连接和事务处理等等，它们还需要具有无需应用程序开发者做出任何努力便实现自动负载平衡的能力。应用程序还应该能够运行在各种适应于所期望客户容量以及在需求增长时能够方便地转换机器配置的服务器上。

1.1.3 与现有系统的集成

对于许多企业和机构来说，现有信息系统多年来已经收集了大量的数据。大量的程序开发投资也都注入在运行于这些现存系统的各种应用程序中。因此，对于企业级应用程序开发者来说，所面临的挑战便是怎样对其进行重用并商业化。

为了达到这一目标，应用程序开发者需要获得访问中间层以及各种诸如数据库管理系统和事务监控器等后端服务的标准化方法。他们同时也需要一致性地提供这些服务的系统，从而使得新的编程模型并不是作为一个包含企业中各种系统的扩展性集成来实现。

1.1.4 选择的自由

对于应用程序的开发而言，要求具有混合并匹配解决方案的能力，从而达到对当前任务的最优化配置。企业级应用程序开发中选择的自由不仅仅局限于服务器，同时也包括各种工具和组件。

在服务器产品中的选择使企业能够选择适应于相应的应用程序要求的配置。同时也提供了应内部或外部要求从一种配置迅速且容易地转移到另一种配置的能力。

对工作中所需要的各种工具的访问则是另一个重要的选择，开发团队应该能够随着新的需求的增长而采用新的工具，包括来自于服务商和第三方工具开发商的各种工具。而且，开发团队的每一位成员都应该能够得到最适合他们相应技术的工具。

最后，开发者应该具有充分的自由从现有的应用程序组件市场进行选择，以充分利用外部知识并增强应用程序的开发能力。

1.1.5 保证安全

具有一定讽刺意味的是，分发信息资源从而提取有价值的东西同时也会导致一些损害。实际上，传统模式下，无论在服务器还是在客户，IT 部门都能够保证相当高水平的控制。然而当信息资源被分发到一个缺乏保护措施的环境中时，对于其中一些敏感性的资源，保证其安全便变得尤为重要，同时还必须保证其他资源能够顺利地被客户访问。

在集成分立的系统方面，存在的一个难题便是提供一个统一的安全模型。跨越内部应用程序及资源边界只需一次登录对于建立用户的应用程序体验来说十分重要。安全需要与现存机制相兼容。在客户需要访问可靠信息的情况下，这种机制需要保证高度的安全，同时尽可能保持非强制性和透明性。

1.2 用于企业解决方案的平台

J2EE 平台为实现和部署企业应用程序提供了一个标准。J2EE 平台是通过一个开放式的过程来设计的，并有一些企业级计算供应商的参与，以保证它能够适应尽可能广范围内的企业级应用程序的要求。这样，J2EE 平台所要解决的核心问题便是促进企业组织加快在信息化经济中竞争的步伐。

1.2.1 J2EE 平台概述

J2EE 的设计目的是提供对服务端和客户端的支持从而推动企业及企业中多层应用程序的发展。一般来说，这些应用程序都有一个客户层以提供用户界面，一个或多个中间层模块用来提供客户服务以及应用程序所要用到的商务逻辑，另外还有一个后端的企业信息系统用来提供数据管理。图 1.1 描述了组成一个典型的 J2EE 环境的各种组件和服务。

1.2.1.1 多层模型

正如图 1.1 中所描述的一样，J2EE 平台提供了一个多层分布式应用程序模型。这也就意味着应用程序的各个部分能够运行在不同的设备上。J2EE 结构中分别定义了客户层（client tier）、中间层（middle tier，由一个或多个子层组成）和用来提供现存信息

系统的各种服务的后端系统层。客户层可以支持各种客户类型，即包括公司防火墙内部和外部的用户。中间层通过 Web 层的 Web 容器来支持客户服务，并通过 EJB (Enterprise JavaBeans™) 层的 EJB™ 容器支持商务逻辑组件服务。企业信息系统 (EIS) 层能够通过标准 API 的方法来支持对现存信息系统的访问。

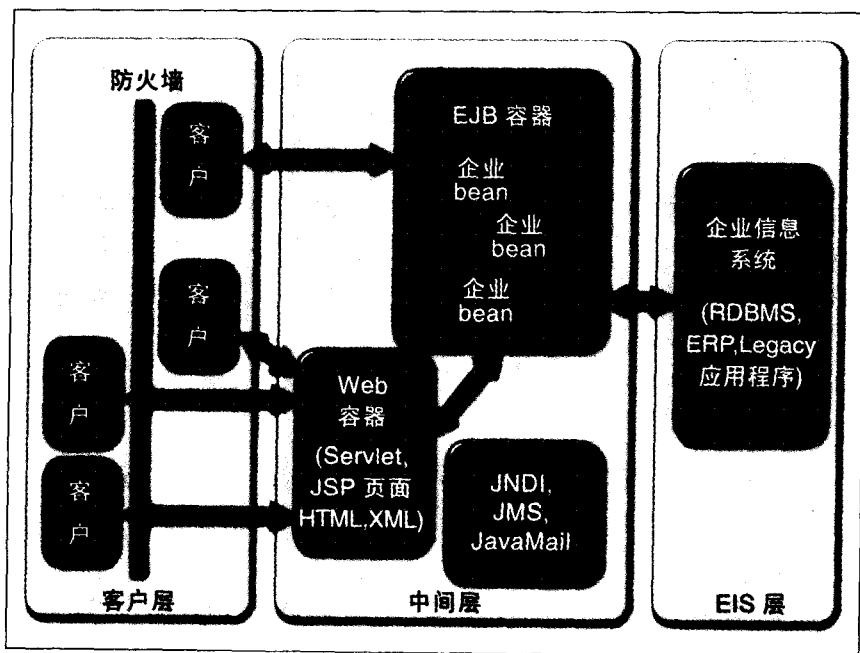


图 1.1 J2EE 环境

1.2.1.2 基于容器的组件管理

在 J2EE 基于组件的开发模型中，其中心是容器的概念。容器是提供特定组件服务的标准化运行时环境。通过这些组件，可以在任何供应商的任何 J2EE 平台上得到所期望的服务。例如，所有 J2EE Web 容器都提供响应客户要求、执行请求时间处理（例如激发 JSP 或 servlet 行为），以及将结果返回至客户的运行时支持。所有 EJB 容器都提供对事务和 EJB 组件的生命周期管理的自动化支持，同时支持 bean 查询以及其他服务。容器还提供了对企业信息系统的标准化访问，比如说，提供通过 JDBC API 进行的 RDBMS 访问。

此外，容器还提供了一种用来在汇编或部署时选择应用程序行为的机制。通过部署描述符的使用[根据精确定义的 XML 标签 (tag) 来指定组件行为的文本文件]，组件在部署时能够配置为一个特定的容器环境，而不是组件代码。这些在部署时能够配置的特性还包括安全性检查、事务控制以及其他管理功能。

J2EE 规范定义了必须支持的组件容器，却没有指定或者说限制对于这些容器的配