Java EE基础与应用

翁秀木

James Gosling

Joshua J. Bloch

Duke, the
Java mascot

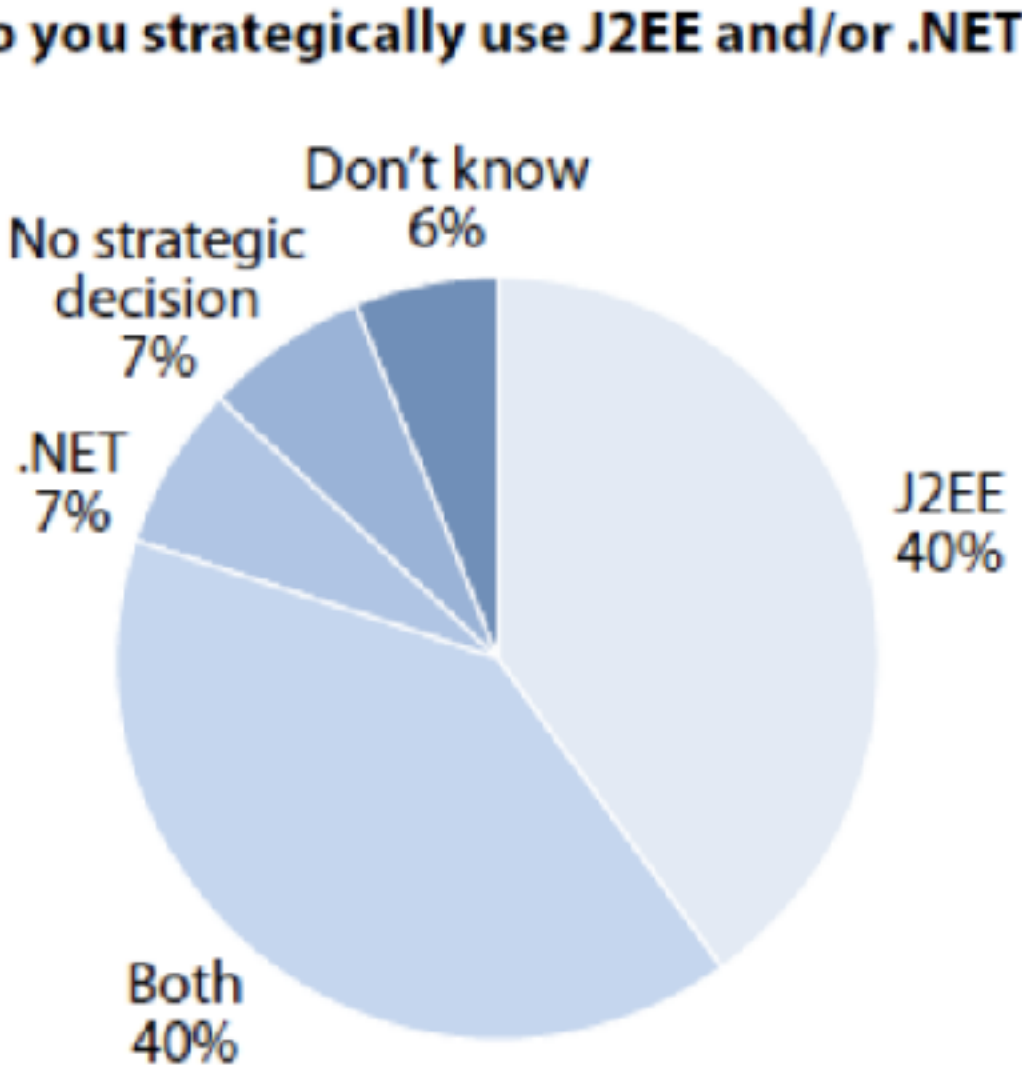
为什么要Java EE

- 益于创建大型、多层、可伸缩、可靠和安全的网络应用程序。

- 提供开发模型、API和运行环境，给出**客户-服务器端模式支持**、**安全**、**事务管理**、**对象-关系式映射**、**分布式计算**、**异构系统整合**等一系列问题的解决方案、使得开发者可更专注于业务逻辑，并提高生产率。

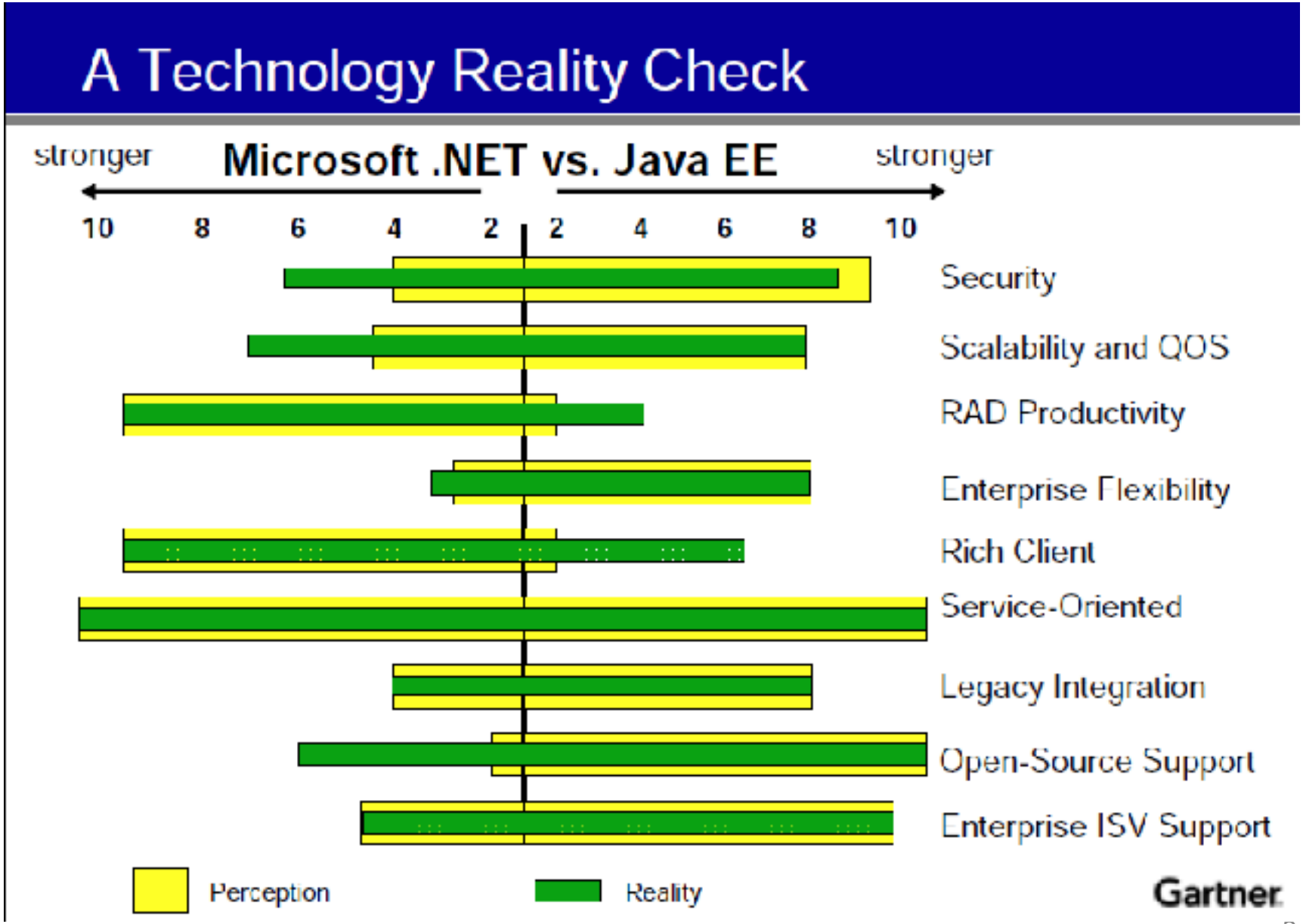
- 提供构建Java企业应用程序的**标准**，借此提高了**兼容性**，也提高了开发、测试、实施和维护的**效率**。

Java EE and .Net

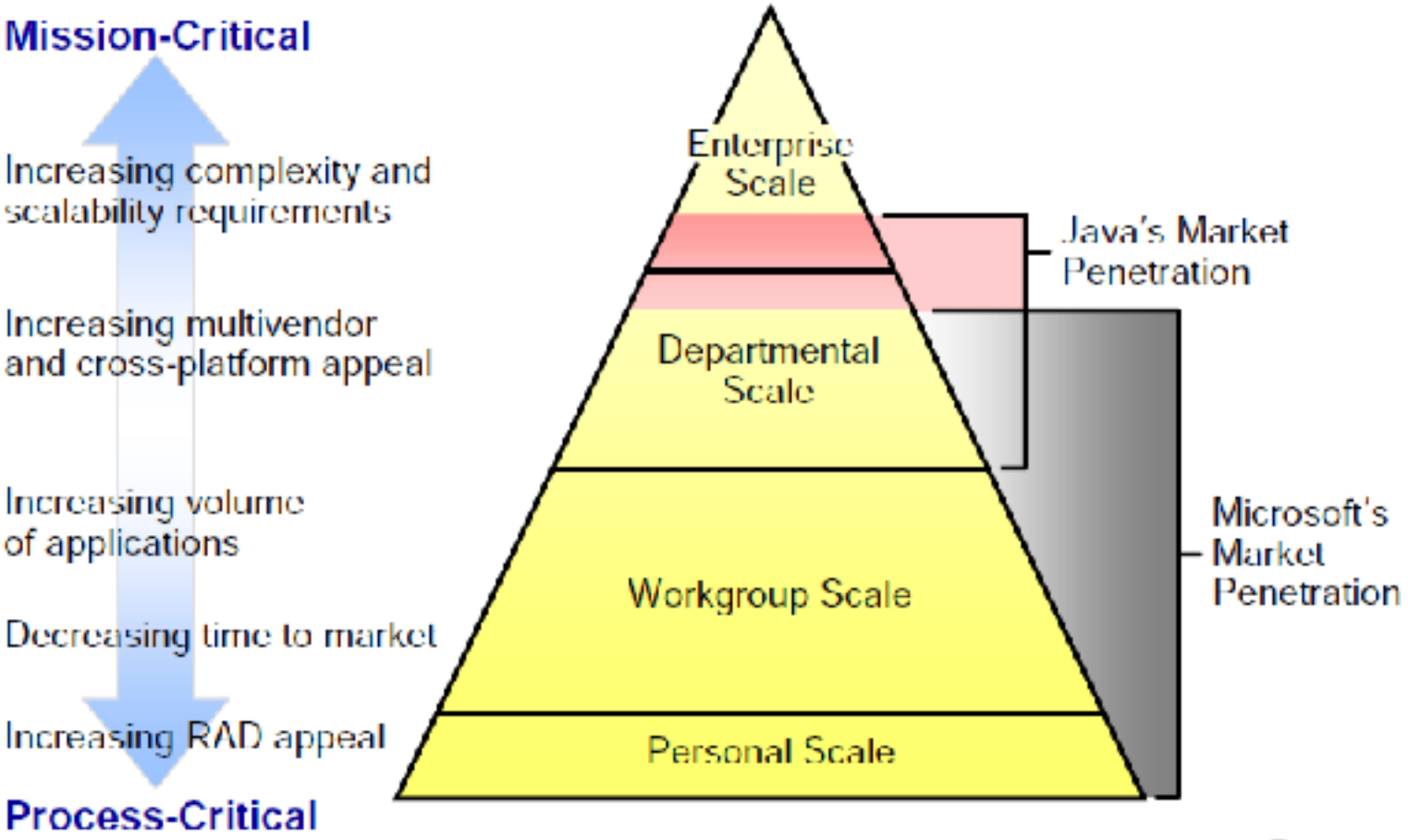


Base: 55 European enterprise architects at financial services firms

Java EE and .Net



Market Segmentation



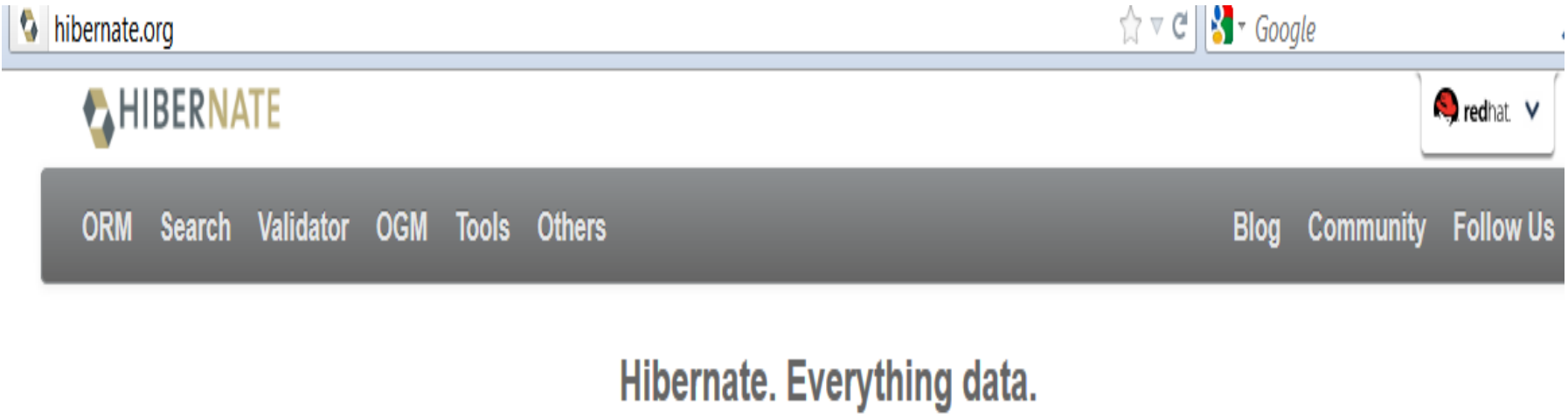
考核方案

- 学习表现20%
- 练习和作业（任务）25%
- 期中测试25%
- 期末考试30%

内容梗概

- Java SE的温故与知新
- Java EE的基础学习
- Java 开源项目的初涉与应用
- 应用服务器、IDE等工具的操练


spring.io
Google
spring
DOCS
GUIDES
PROJECTS
BLOG
FORUM
Let's build a better Enterprise.
Spring helps development teams everywhere build simple, portable, fast and flexible JVM-based systems and applications.


hibernate.org
Google
HIBERNATE
redhat.
ORM
Search
Validator
OGM
Tools
Others
Blog
Community
Follow Us
Hibernate. Everything data.

第一回 概述

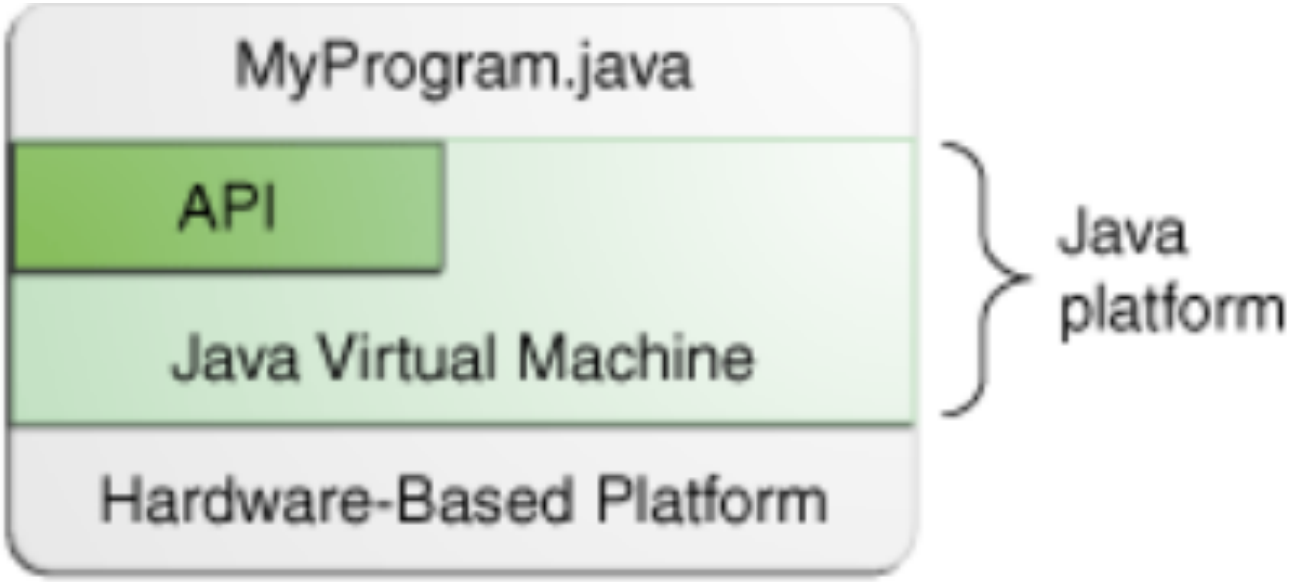
- Java EE的定义
- C/S模型和Peer to Peer（点对点）模式
- 企业应用程序（enterprise application）的分层
- Java EE应用实例

学习目标

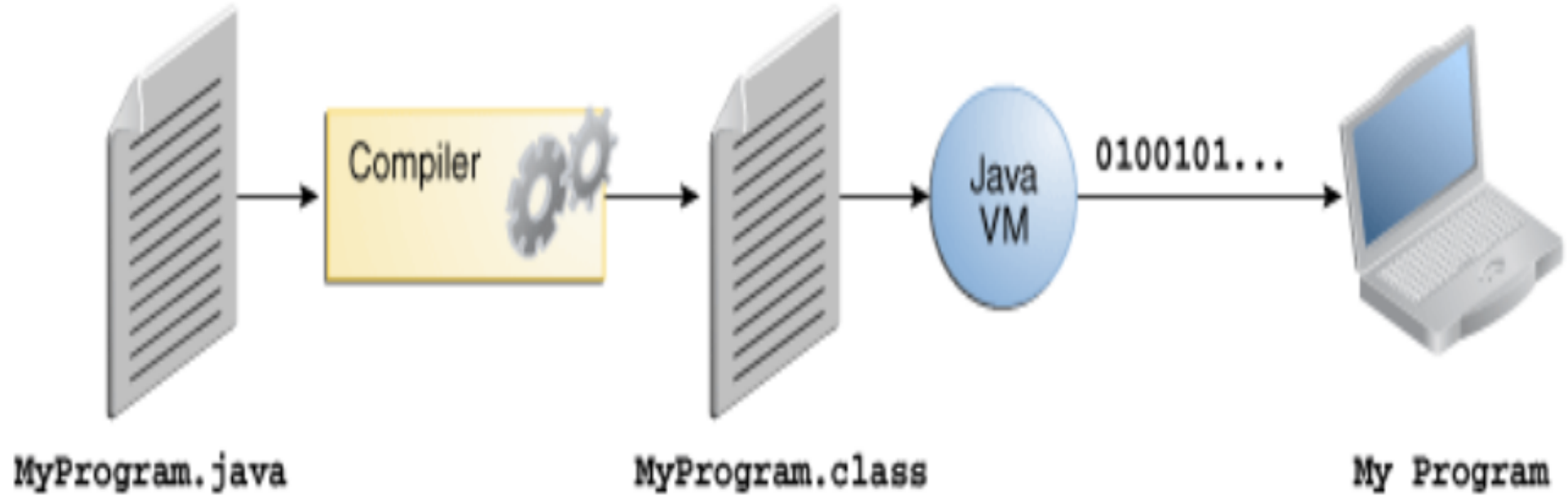
- 掌握为什么要Java EE
- 掌握Java EE的定义，掌握Java EE与Java SE的区别和联系。
- 掌握C/S模型，了解点对点（Peer to Peer）模型。
- 掌握企业应用程序的分层和各层的作用，了解各层所用的技术。
- 了解Java EE应用实例。

Java EE的定义

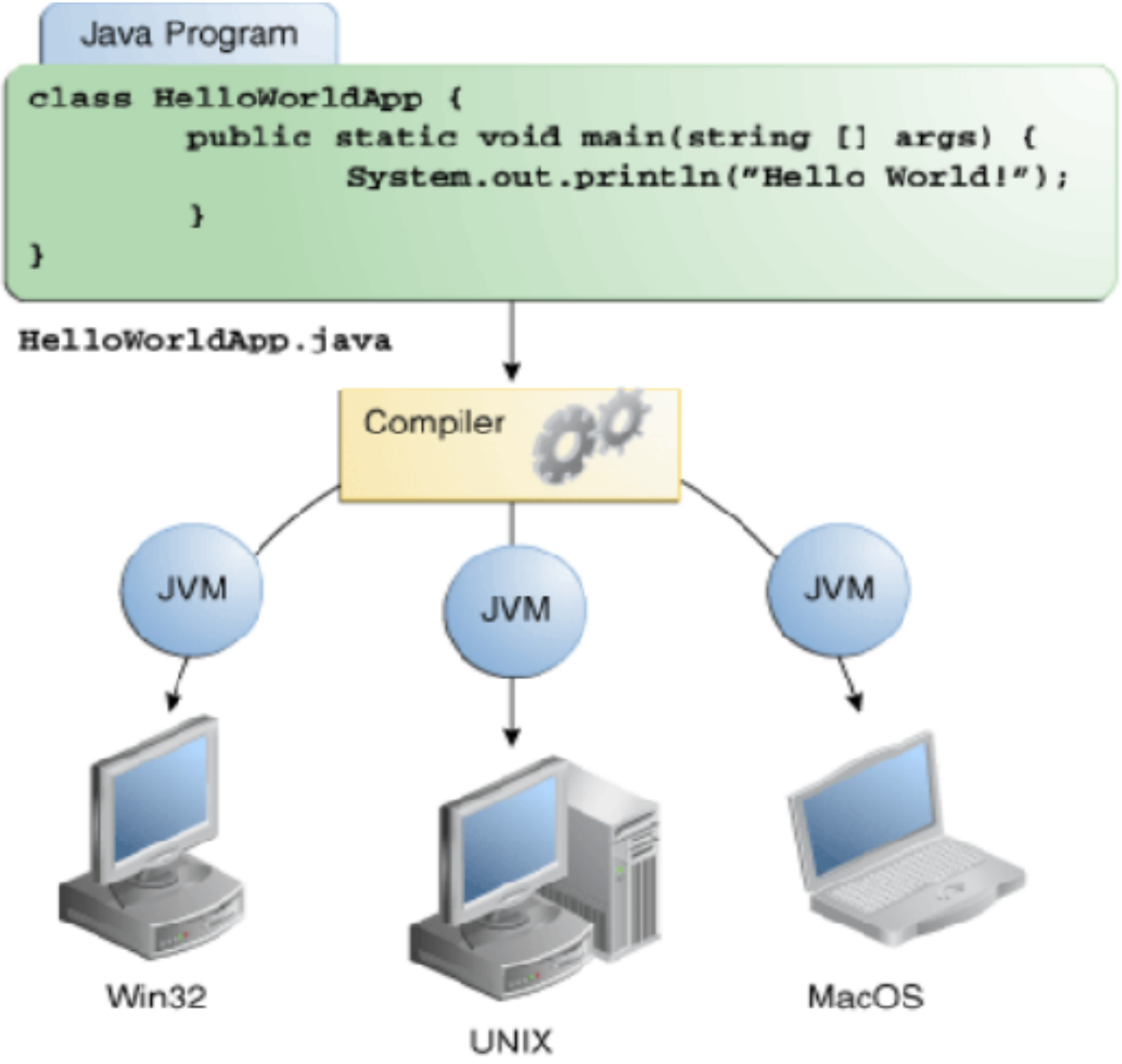
- Java EE=Java Platform，Enterprise Edition or Java Enterprise Edition Platform
- Java EE Platform = JVM + Java EE API

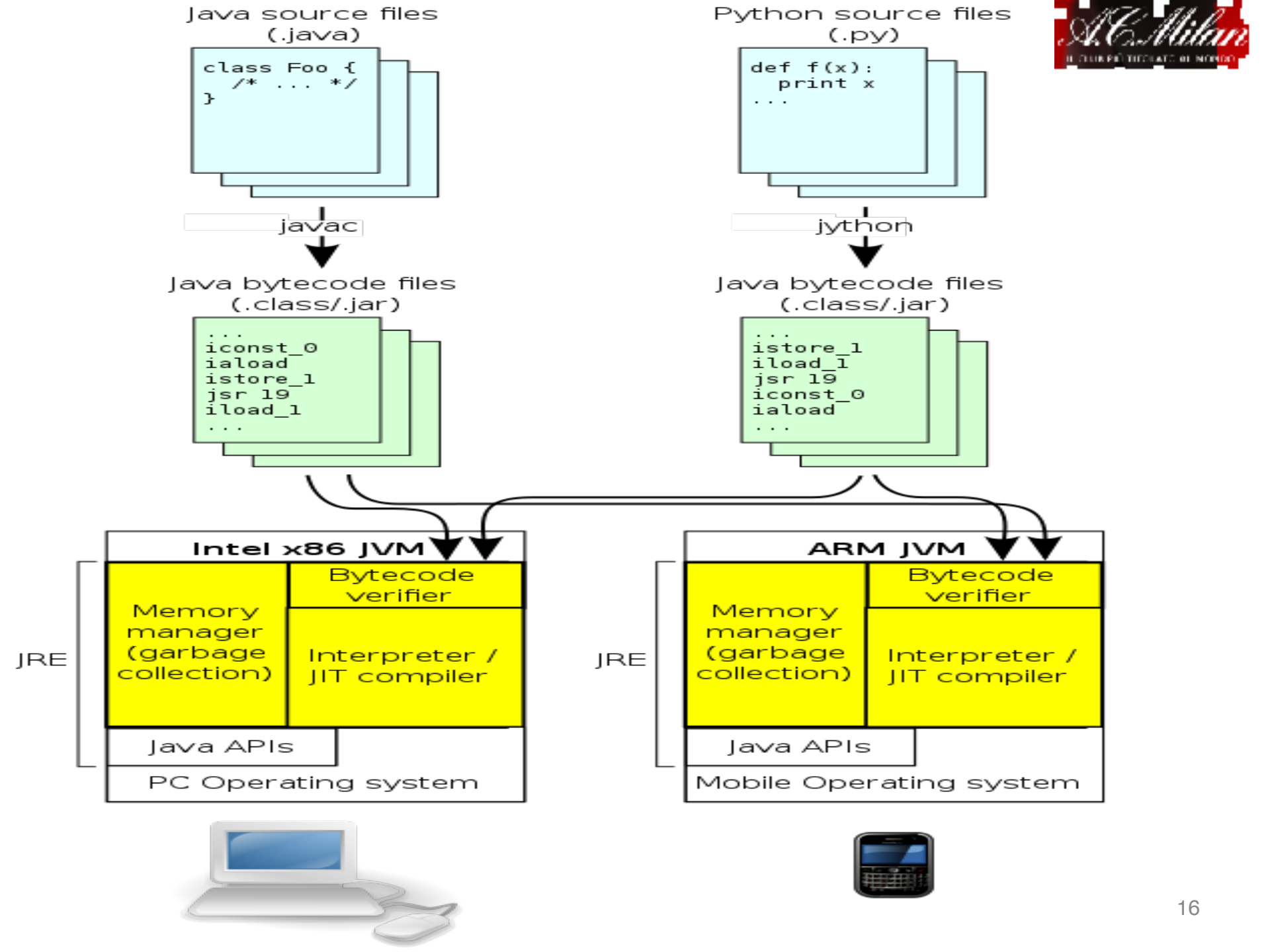


The API and Java Virtual Machine insulate the program from the underlying hardware.



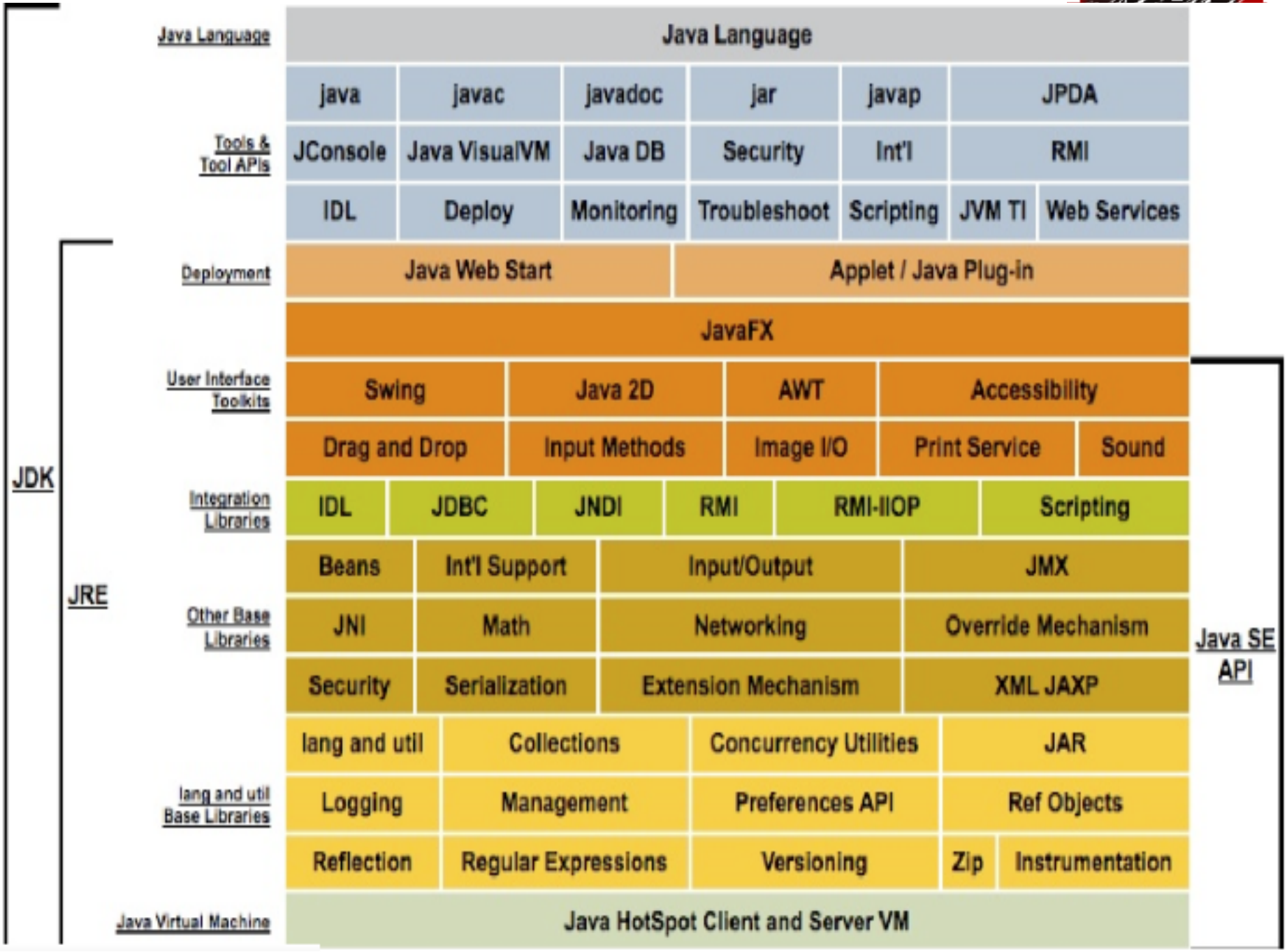
An overview of the software development process.


Java Program
class HelloWorldApp {
public static void main(string [] args) {
System.out.println("Hello World!");
}
}
HelloWorldApp.java
Compiler
JVM
JVM
JVM
Win32
UNIX
MacOS


Java source files
(.java)
class Foo {
/* ... */
}
javac
Java bytecode files
(.class/.jar)
...
iconst_0
iaload
istore_1
jsr 19
iload_1
...
Python source files
(.py)
def f(x):
print x
...
jython
Java bytecode files
(.class/.jar)
...
istore_1
iload_1
jsr 19
iconst_0
iaload
...
Intel x86 JVM
JRE
Memory manager (garbage collection)
Bytecode verifier
Interpreter / JIT compiler
Java APIs
PC Operating system
ARM JVM
JRE
Memory manager (garbage collection)
Bytecode verifier
Interpreter / JIT compiler
Java APIs
Mobile Operating system

As a platform-independent environment, the Java platform can be **a bit slower than native code**.

However, advances in compiler and virtual machine technologies are bringing **performance close to** that of **native code without threatening portability**.


Java Language
Java Language
java
javac
javadoc
jar
javap
JPDA
Tools & Tool APIs
JConsole
Java VisualVM
Java DB
Security
Int'l
RMI
IDL
Deploy
Monitoring
Troubleshoot
Scripting
JVM TI
Web Services
Deployment
Java Web Start
Applet / Java Plug-in
JavaFX
User Interface Toolkits
Swing
Java 2D
AWT
Accessibility
Drag and Drop
Input Methods
Image I/O
Print Service
Sound
Integration Libraries
IDL
JDBC
JNDI
RMI
RMI-IIOP
Scripting
Other Base Libraries
Beans
Int'l Support
Input/Output
JMX
JNI
Math
Networking
Override Mechanism
Security
Serialization
Extension Mechanism
XML JAXP
lang and util Base Libraries
lang and util
Collections
Concurrency Utilities
JAR
Logging
Management
Preferences API
Ref Objects
Reflection
Regular Expressions
Versioning
Zip
Instrumentation
Java Virtual Machine
Java HotSpot Client and Server VM
JDK
JRE
Java SE API

Java EE与Java SE的联系与区别

Java EE platform创建在**Java SE** platform之上.

Java EE 提供API和运行环境，以开发大型、多层、可伸缩、可靠和安全的网络应用程序。

Java EE 7部分技术

Java Servlet 3.1
JavaServer Pages 2.3
JavaServer Faces 2.2
Expression Language 3.0：表达式语言
JavaServer Pages standard Tag Library (JSTL) 1.2

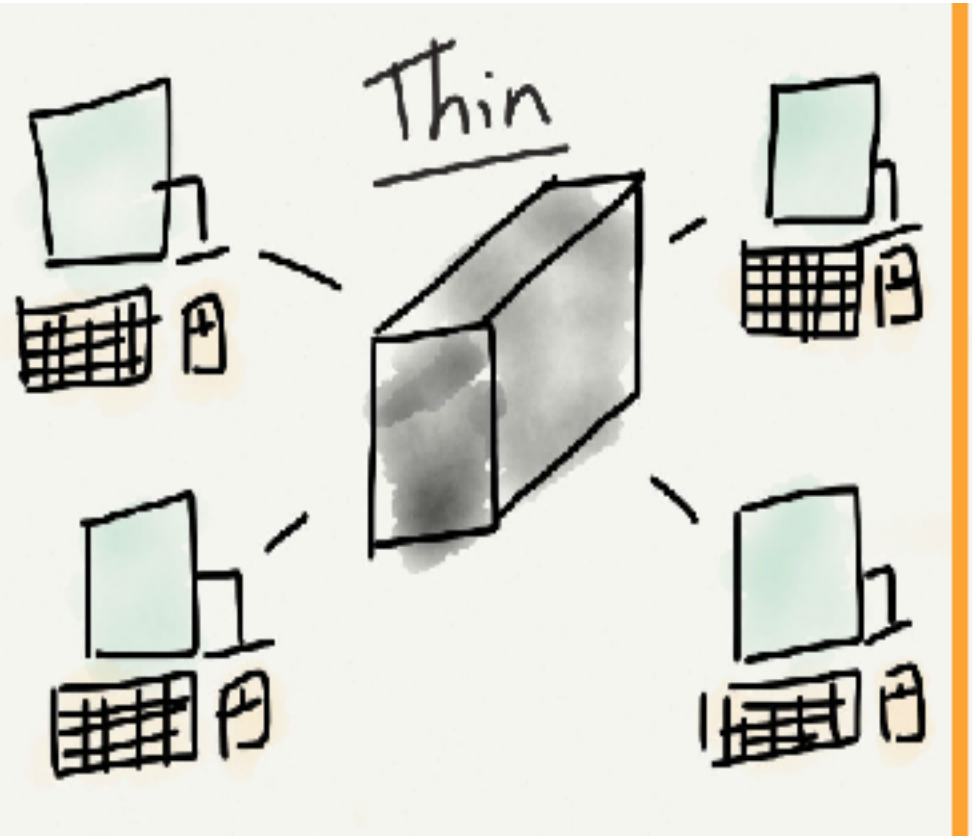
Enterprise JavaBeans 3.2
Java EE Connector Architecture (JCA)
Java Persistence 2.1
Java Message Service API 2.0： 消息服务
Java API for RESTful Web Services (JAX-RS) 2.0
Java API for XML-Based Web Services (JAX-WS) 2.2
Java Transaction API (JTA) 1.2 ： 事务

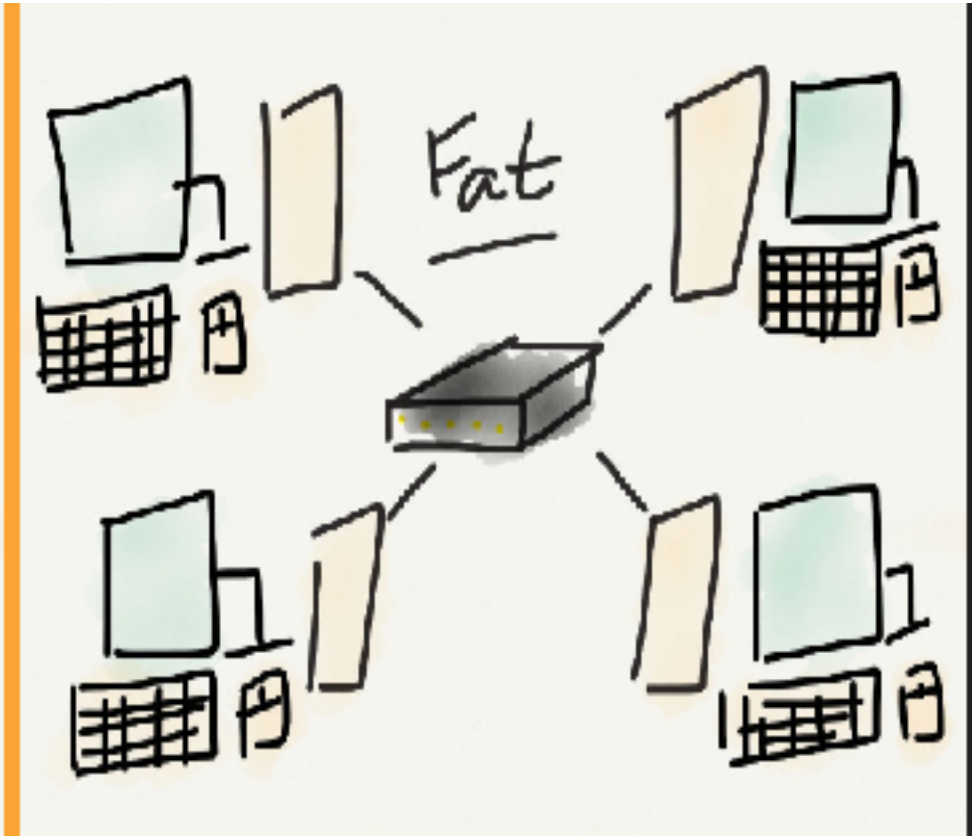
Java Database Connectivity (JDBC) 4.0
Java Naming and Directory Interface (JNDI)

Java EE 7部分API的package

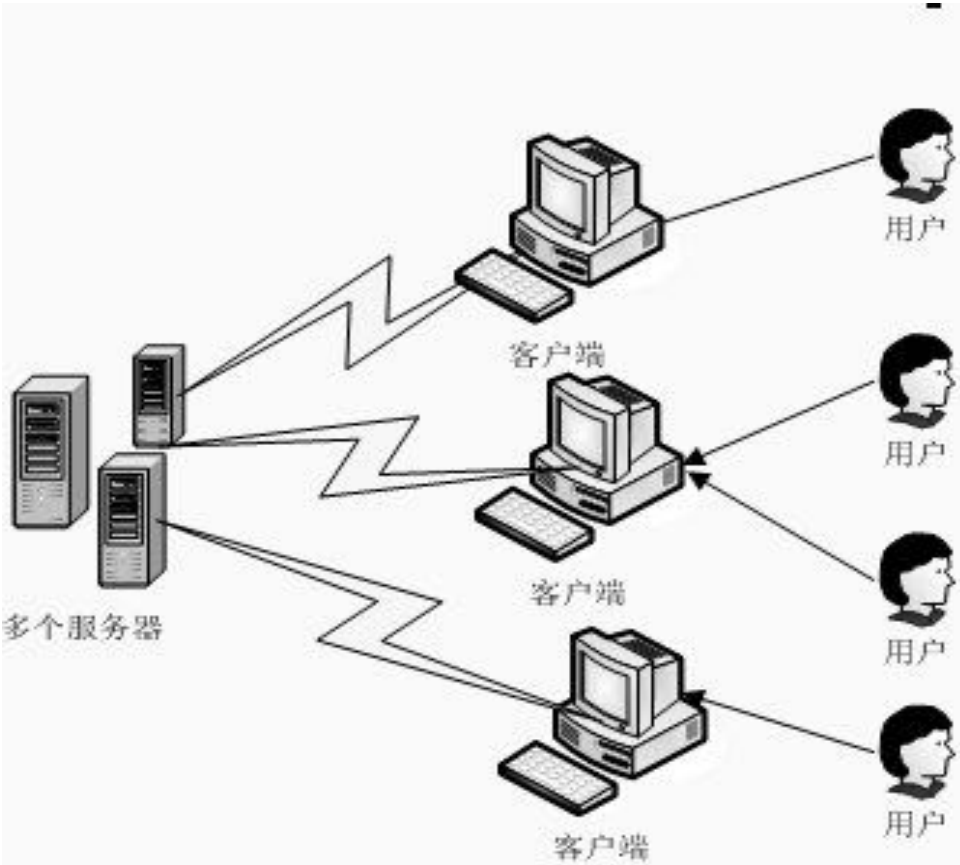
javax.jms
javax.resource ：Java EE Connector API
javax.servlet
javax.servlet.http
javax.servlet.jsp
javax.servlet.jsp.el
javax.servlet.jsp.jstl.core
javax.transaction.xa : JTA
javax.xml.ws ：to create web services

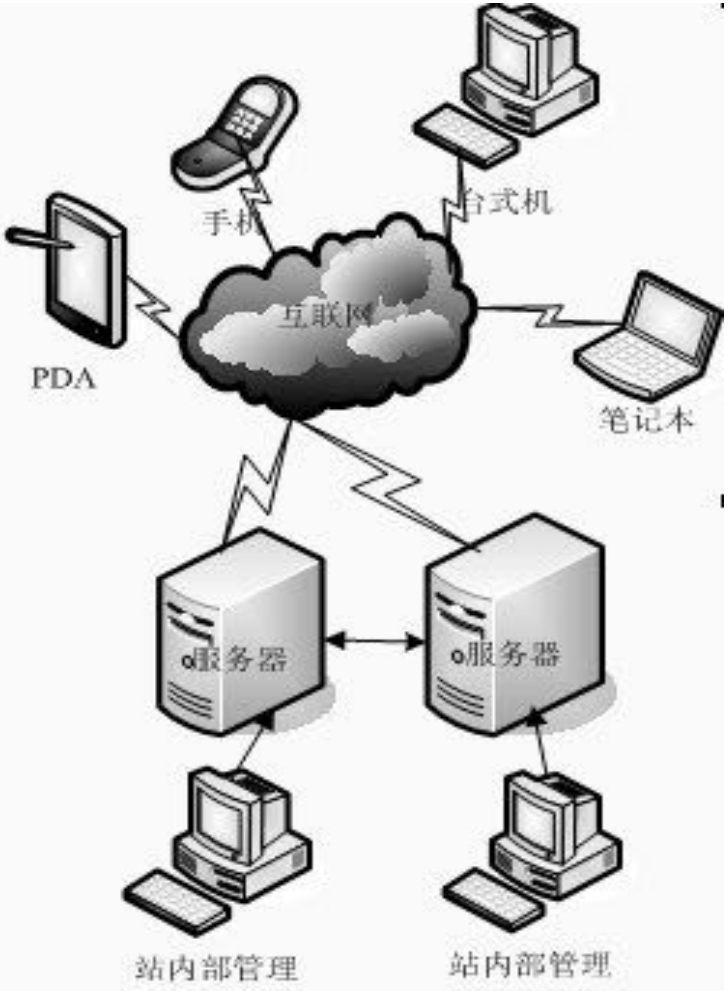
C/S模型: 瘦/肥客户端





C/S模型：本机应用程序客户端，or浏览器客户端+服务器（即B/S）





富互联网程序 Rich Internet Application

- 为让客户得到更好的应用程序体验，把一些功能放在客户端能完成，比如多媒体播放、缓存、2d/3d游戏等，从而减少访问服务器时的网络调用消耗。

- RIA运行可在Browser内（作为插件或用HTML5），也可独立于Browser，类似传统的本地客户端程序。

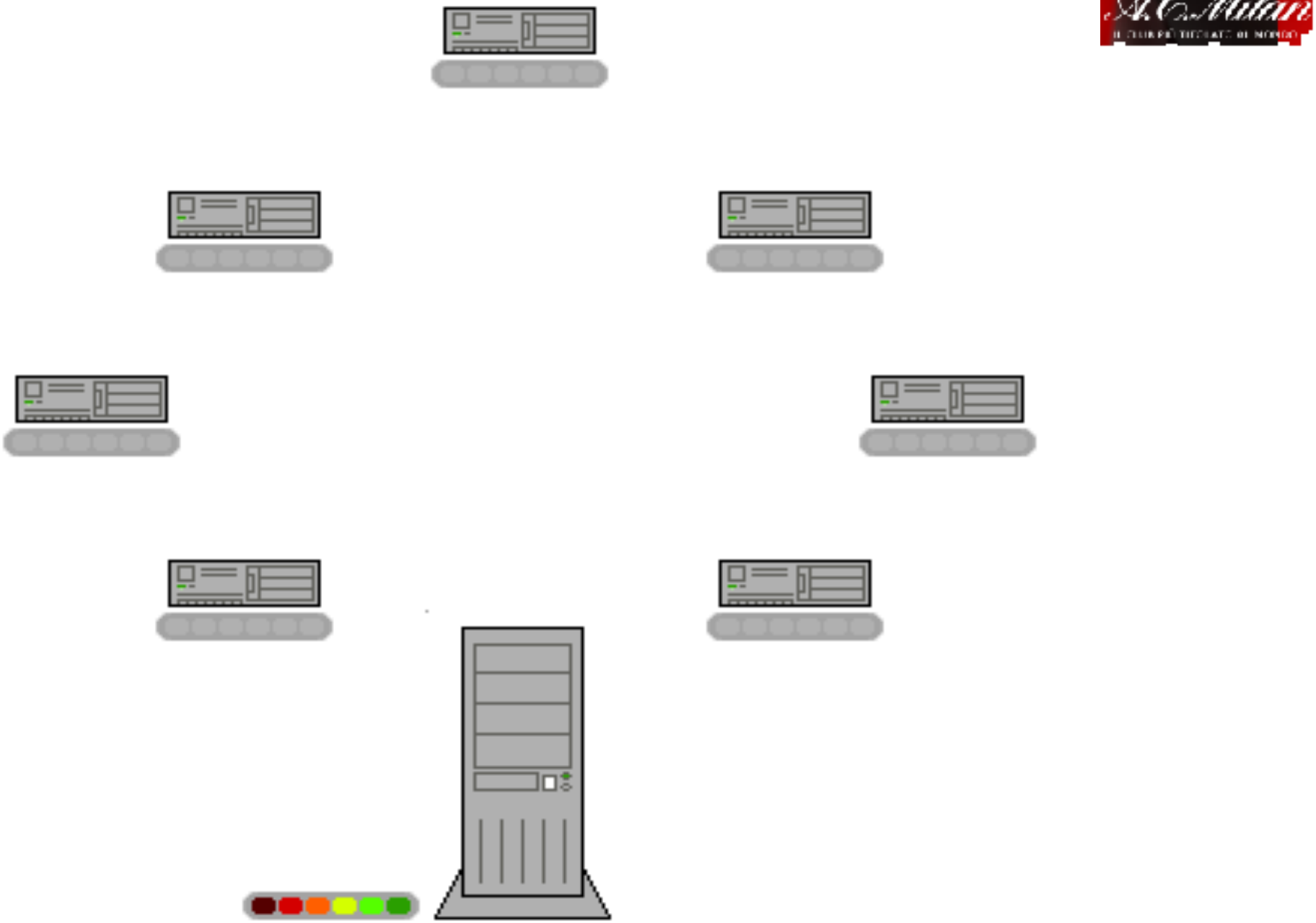
- 主要RIA平台：Adobe's AIR, Microsoft's Silverlight and Oracle's JavaFX

富互联网程序 Rich Internet Application

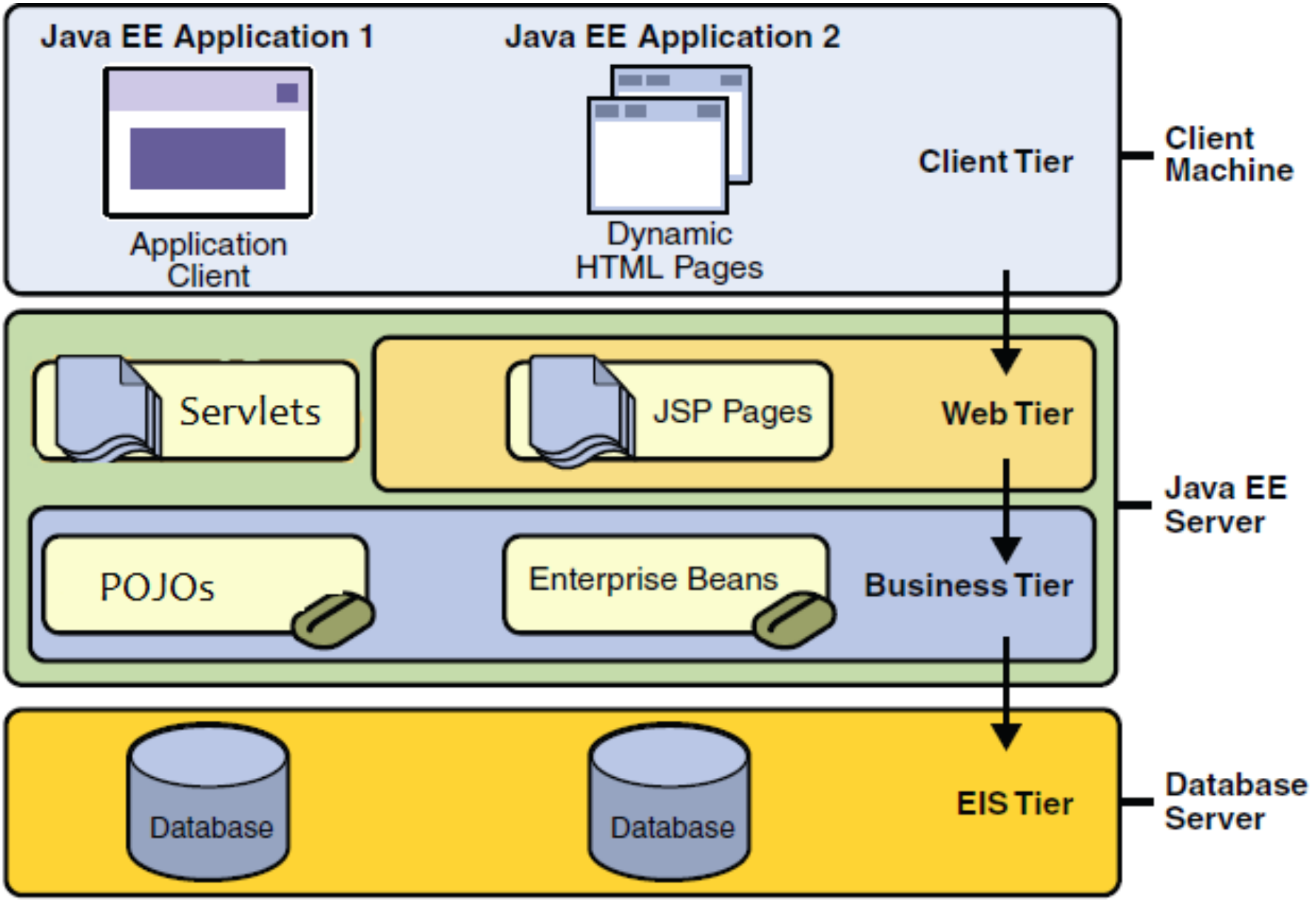
- JavaFX是一种富客户端平台（rich client platform），它目的是为了给企业应用程序提供一个轻量级、速度更快的Java用户界面。

- JavaFX 可以调用本地系统的功能，也可以无缝连接到服务器端的应用程序。

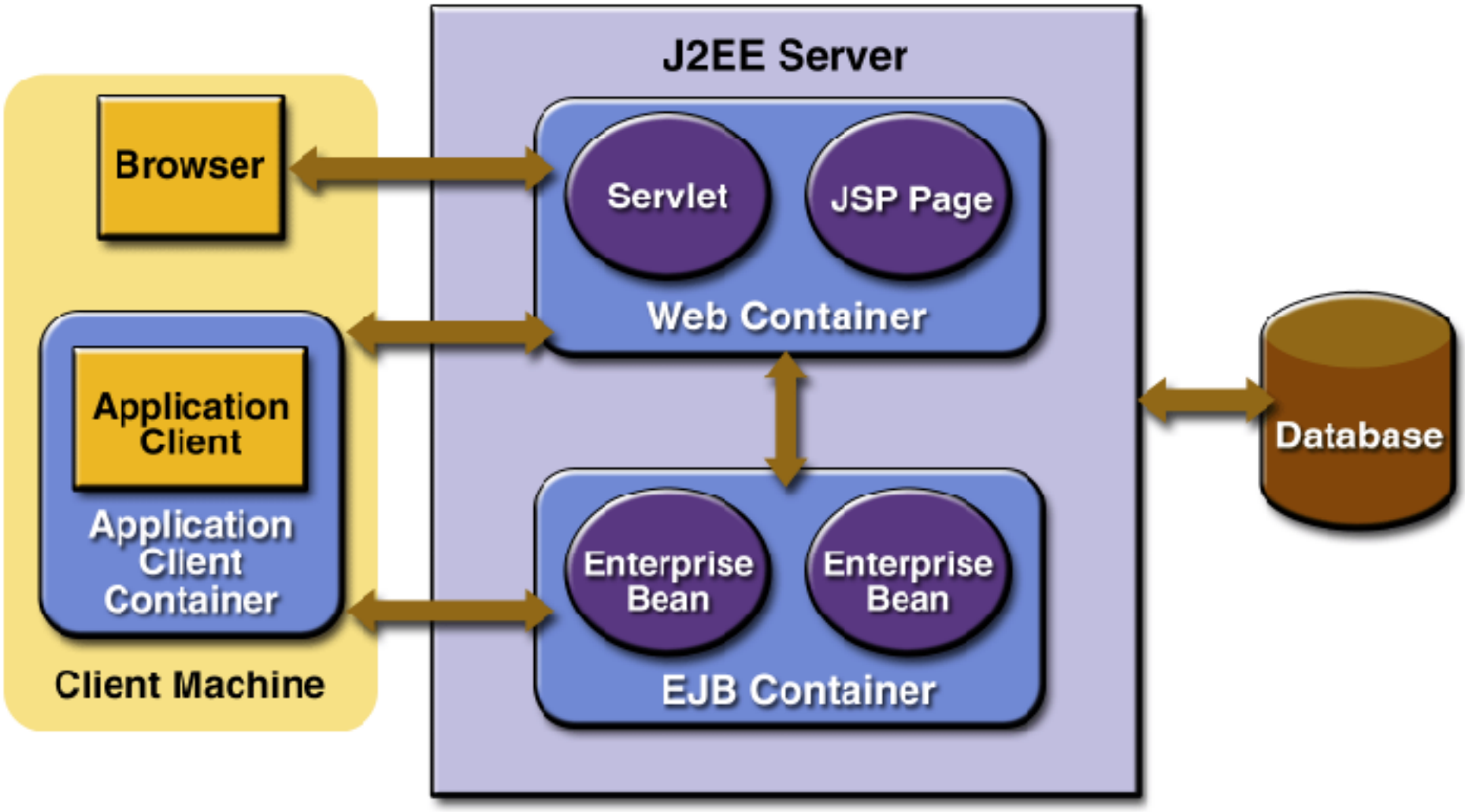
C/S（左）vs. peer to peer（右）

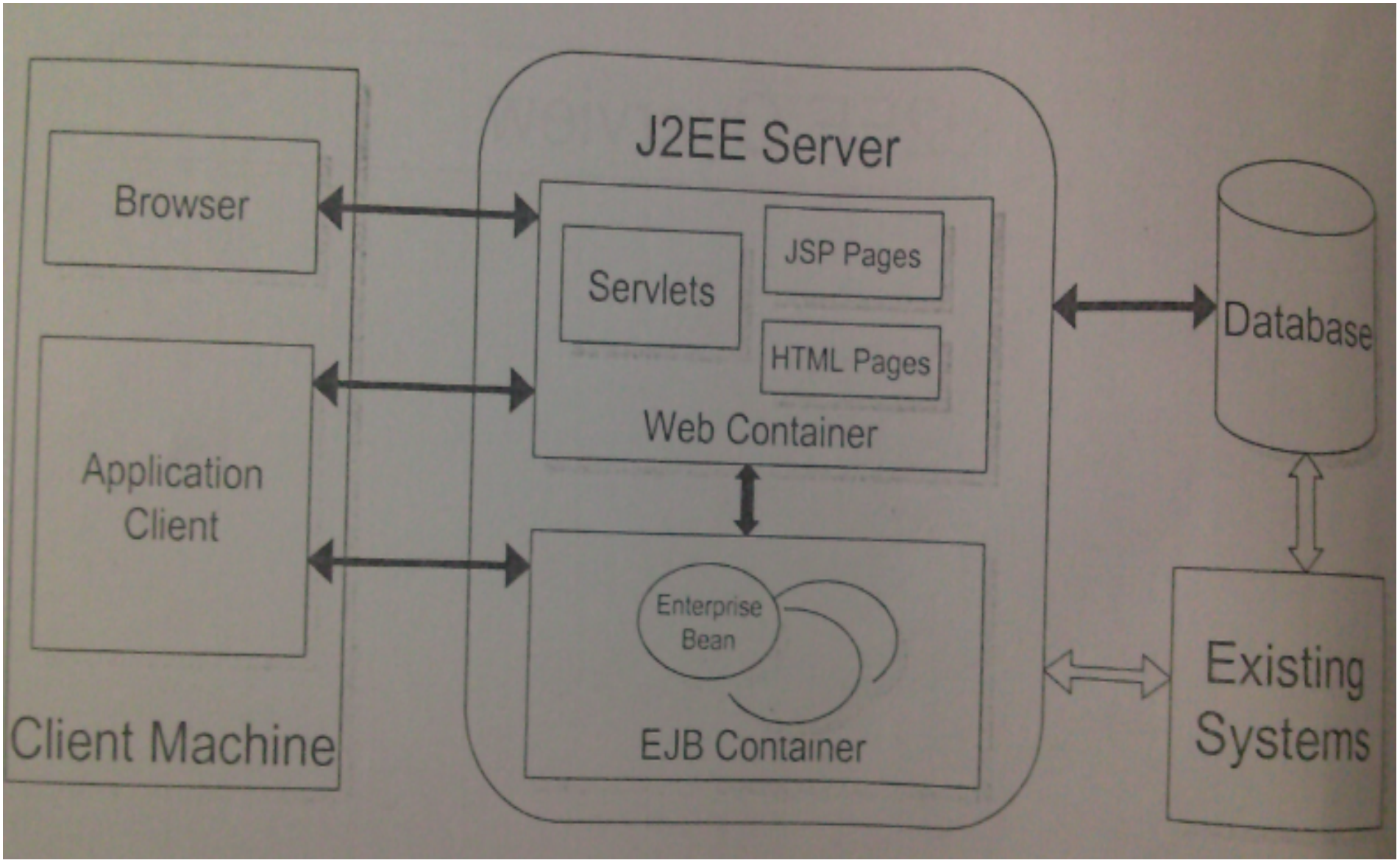


Java EE应用程序的分层结构



Java EE应用程序的分层结构




J2EE Server
Browser
Servlets
JSP Pages
HTML Pages
Web Container
Database
Application Client
Enterprise Bean
EJB Container
Existing Systems
Client Machine

客户端层（client tier）

客户端层由应用程序的客户端组成，客户端包括有万维网浏览器（web browser）和本机应用程序。 Java EE的本机应用程序客户端主要是Java程序，但也可以不是Java程序。

客户端通常与应用服务器不在一个机器上，可以是桌面电脑、手提电脑、手机、PAD等。

客户端向服务器发出请求，服务器处理请求后，再将结果返回给客户端。

Web层 （Web Tier or UI Tier）

Web层组件负责处理客户端和业务层之间的交互，其主要任务有：

- 动态产生不同格式的内容，供客户端使用。
- 获取客户端用户的输入，并调用业务层的组件，以返回适当的结果。
- 控制页面流转
- 维护用户会话（Session）中的数据。
- 执行基本逻辑并把一些数据暂存在JavaBean组件内。

业务层（Business Tier or Middle Tier）

业务层的组件负责提供业务逻辑。

业务逻辑是一些提供了某个业务领域功能的代码，例如金融行业或者电子商务网站。

设计企业应用程序时，应把业务领域功能放在业务层的组件中来实现。

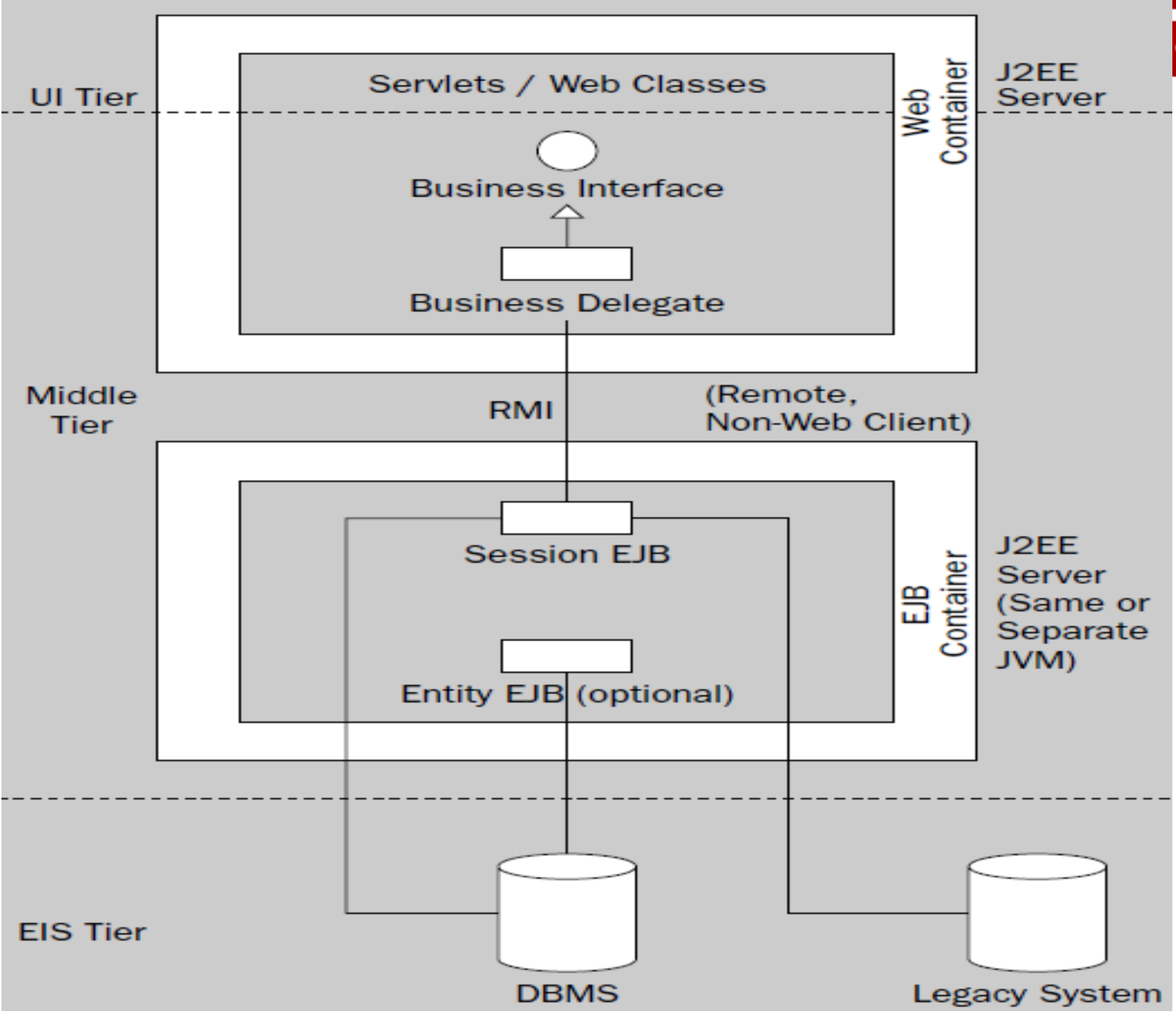
Java EE服务器为业务层提供事务管理和连接缓冲池

企业信息系统层
（Enterprise Information Systems Tier）

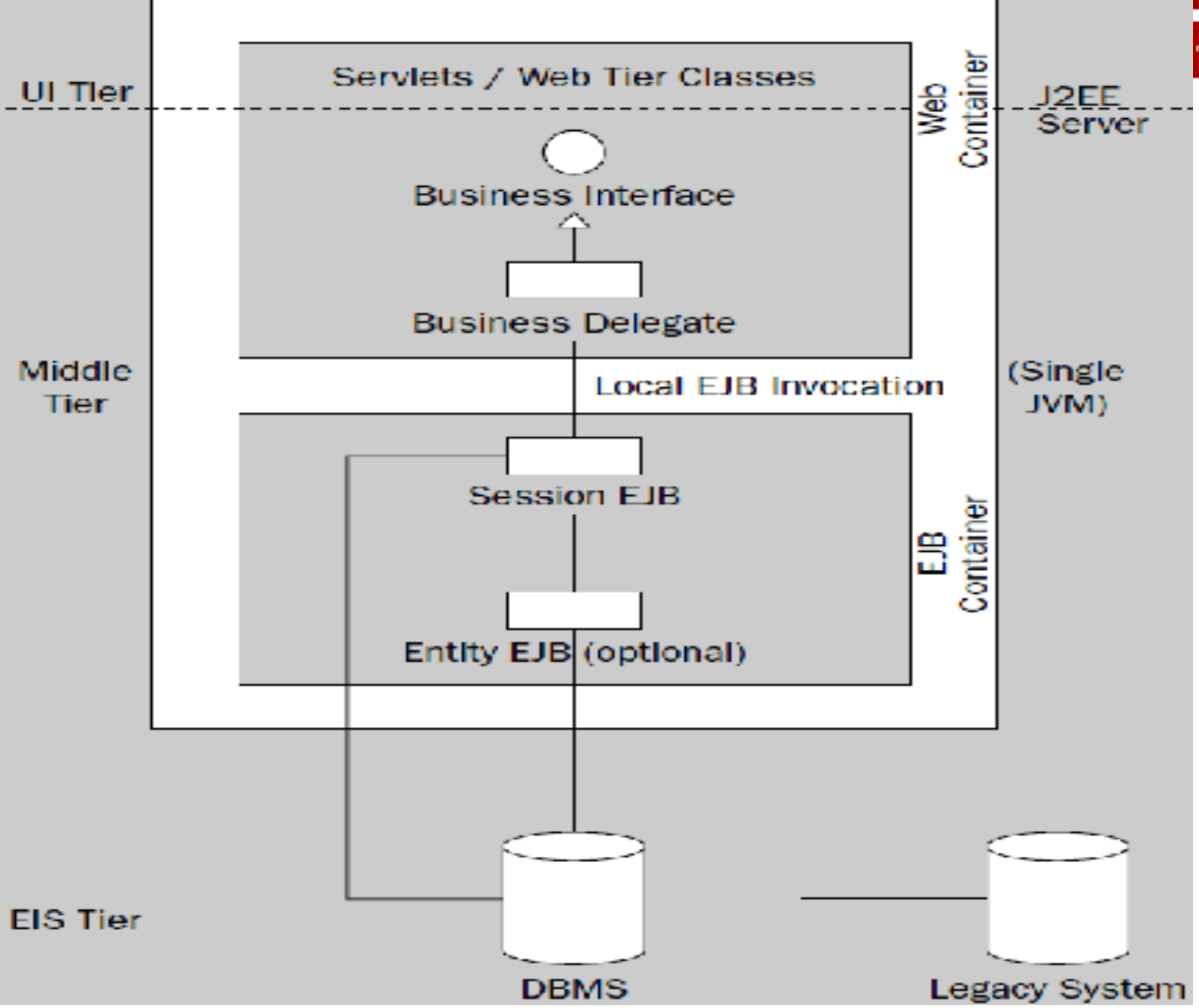
EIS层的组件（资源）通常包括数据库服务器，ERP系统，或其他的现有系统，例如大机（mainframe）上的系统.

EIS层的组件（资源）与Java EE server往往不在一个物理机器, 需要通过业务层的组件来访问它们。

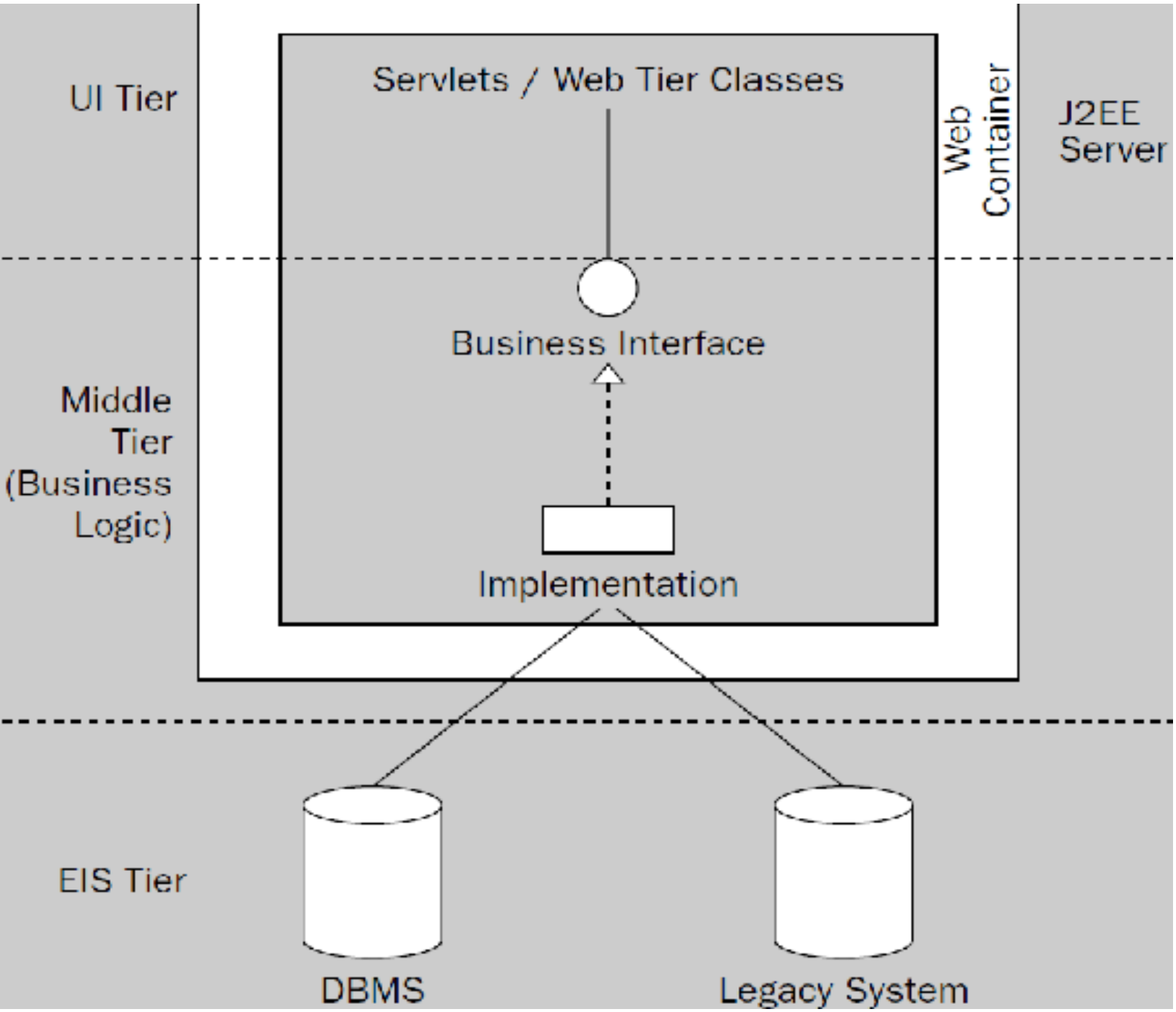
Java EE提供JDBC、JNDI、JCA、JMS等来访问EIS层



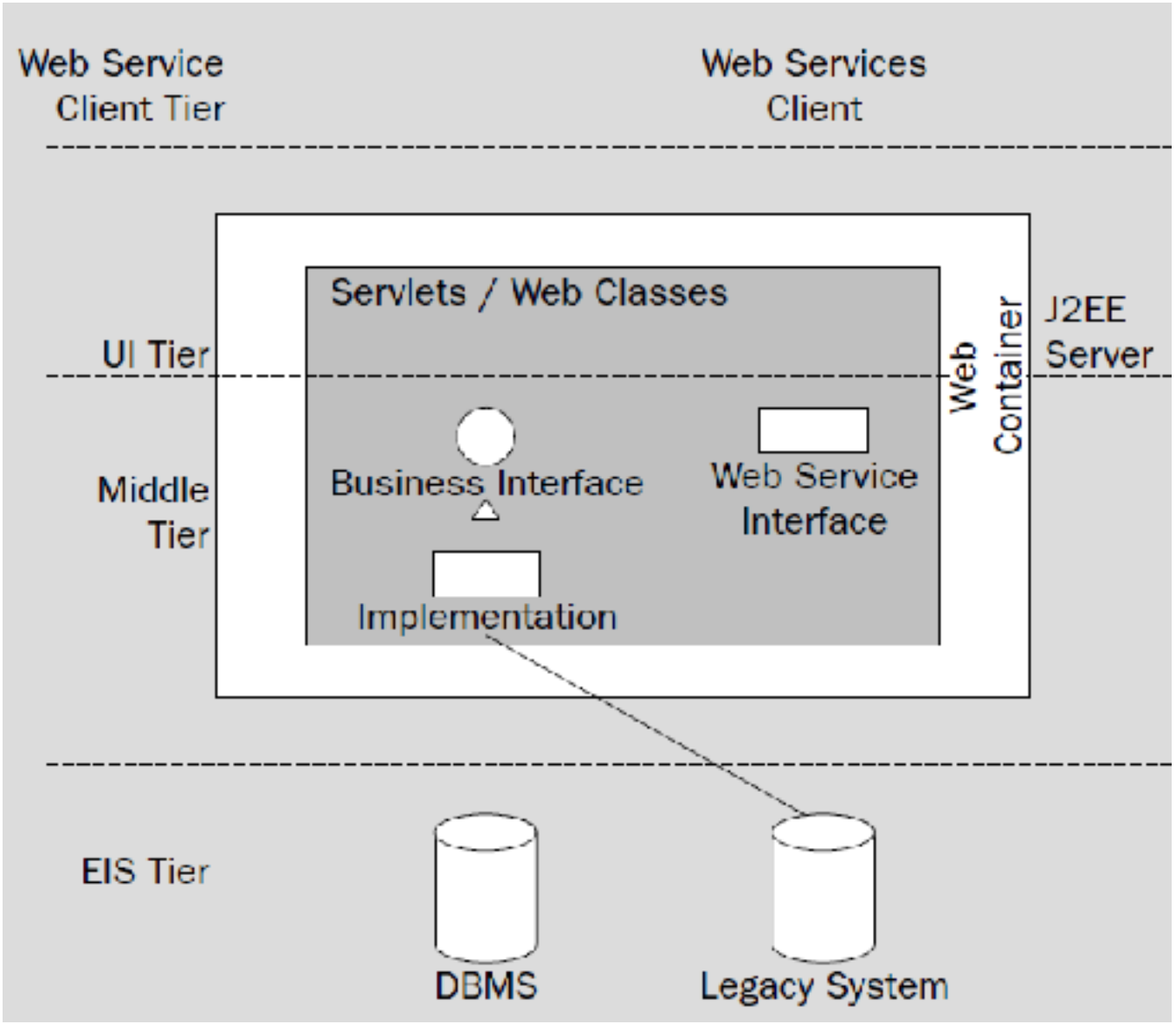
经典JavaEE架构 - 远程调用的EJB架构


UI Tier
Servlets / Web Tier Classes
Web Container
J2EE Server
Business Interface
Business Delegate
Middle Tier
Local EJB Invocation
(Single JVM)
Session EJB
EJB Container
Entity EJB (optional)
EIS Tier
DBMS
Legacy System

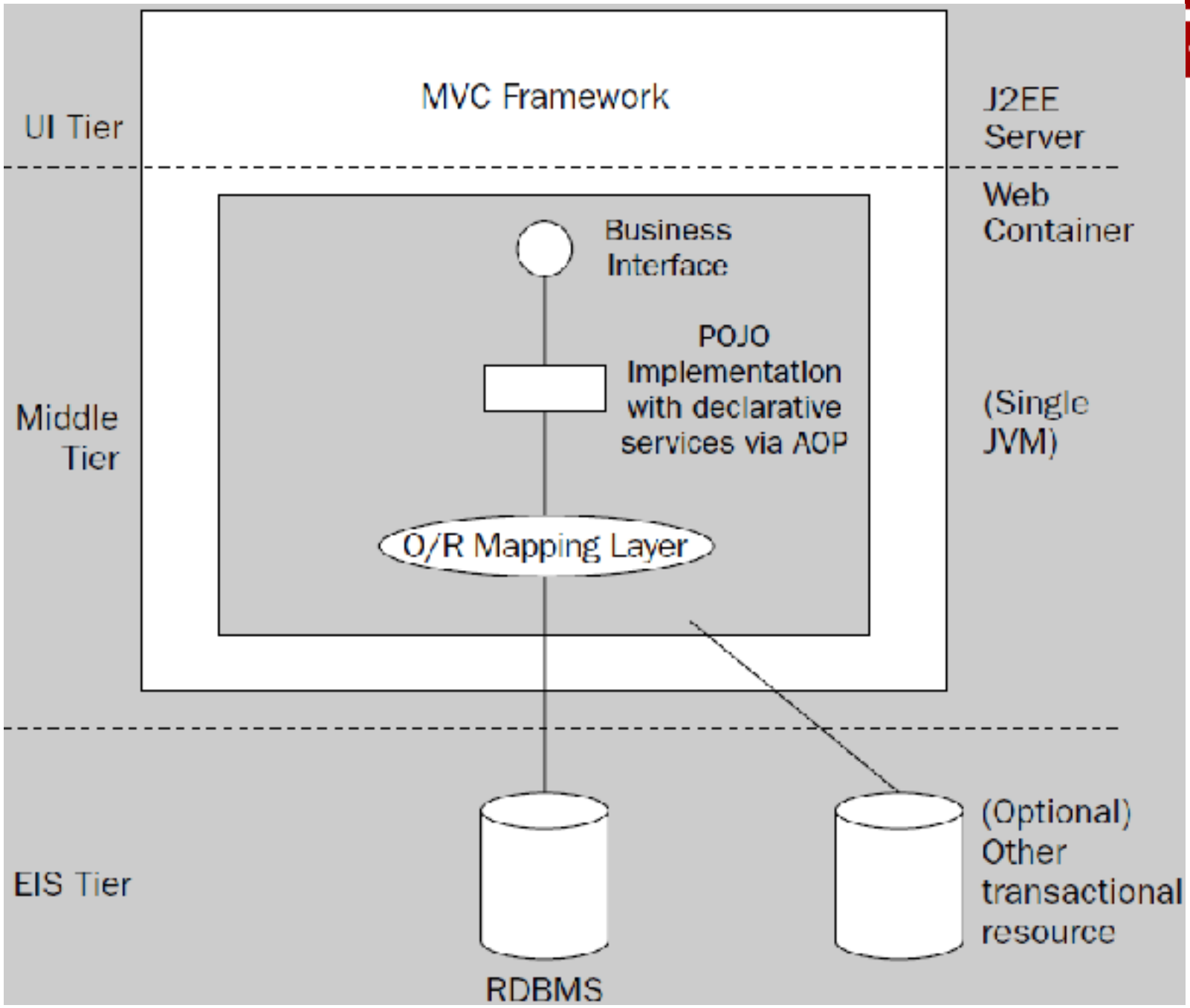
本地调用EJB的架构


UI Tier
Servlets / Web Tier Classes
Web Container
J2EE Server
Business Interface
Middle Tier (Business Logic)
Implementation
EIS Tier
DBMS
Legacy System

简单的架构，无EJB

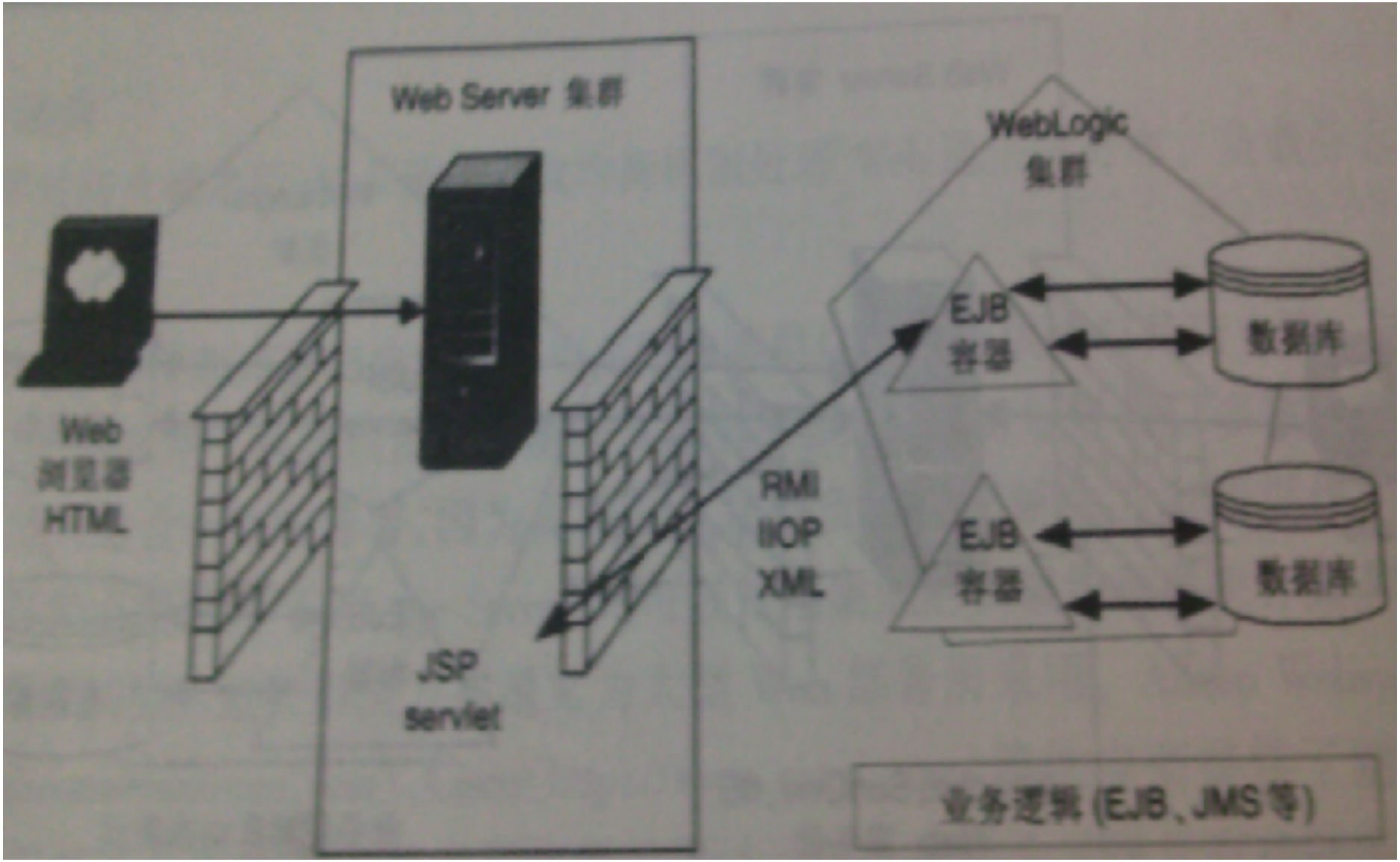


web service 架构，无EJB



轻量级容器的架构，无EJB

Java EE应用实例 1



未完待续，谢谢！