Java Web-based Programming & J2EE Architecture

Peter.Cheng

Email: founder_chen@yahoo.com.cn

http://www.huihoo.org

Course Goal

- The main goal of this course is to provide you with the knowledge for Java web-based programming (JSP, Servlet, Struts), MVC Design pattern, J2EE architectures

Course Overview

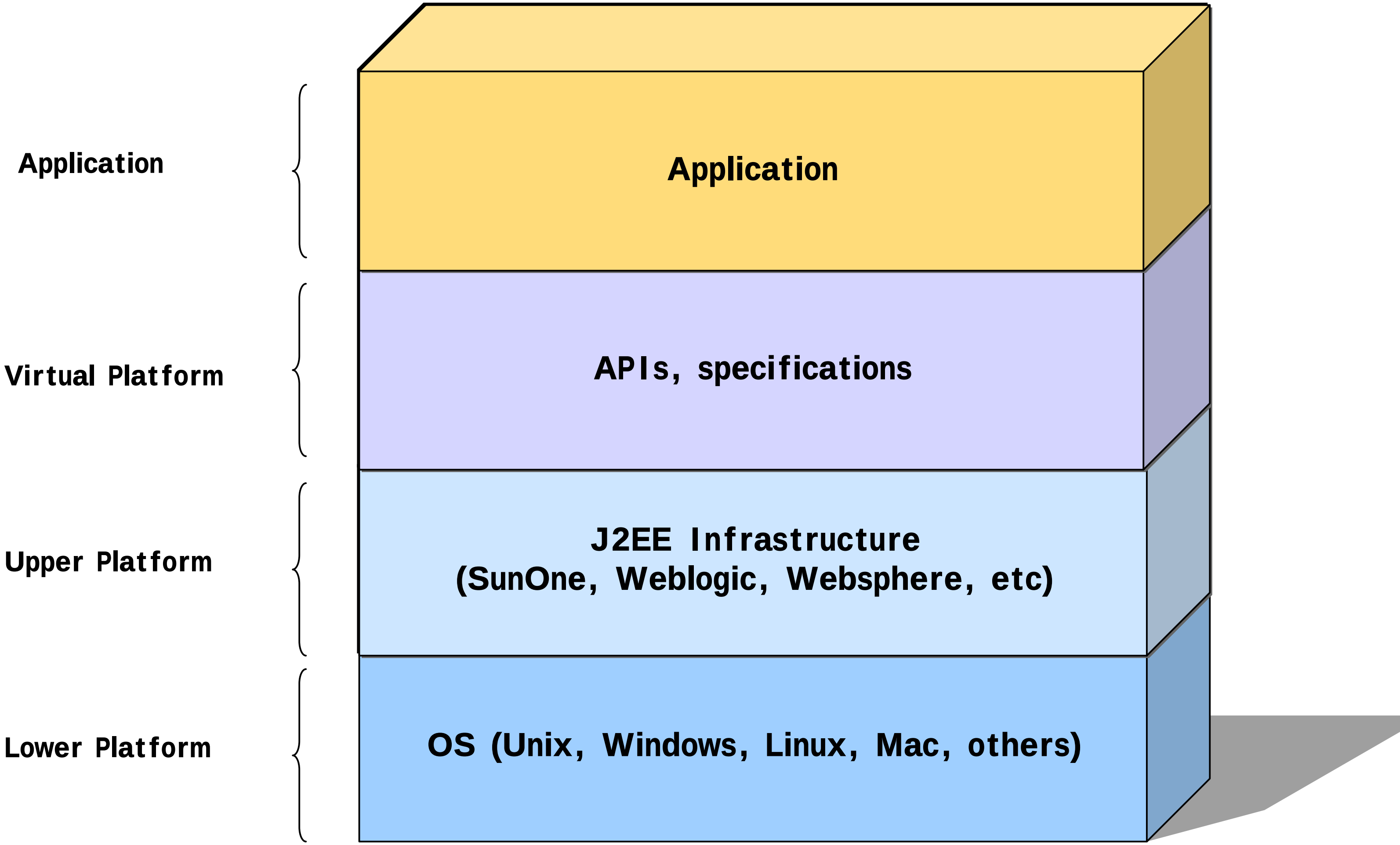
This course covers the following areas:

- J2EE Technology overview
- Web-based Programming Model
- MVC Design Pattern and Struts Framework
- J2EE Architectures
- Build a J2EE developer Team

How To Use Course Materials

- Lecture The instructor will present information specific to the topic of the module. This information will help you learn the knowledge and skills necessary to succeed with the exercises.
- Exercise Lab exercises will give you the opportunity to practice your skills and apply the concepts presented in the lecture.
- Think Beyond Thought-provoking questions are posed to help you apply the content of the module or predict the content in the next module.

Software Architecture



Application

Application

Virtual Platform

APIs, specifications

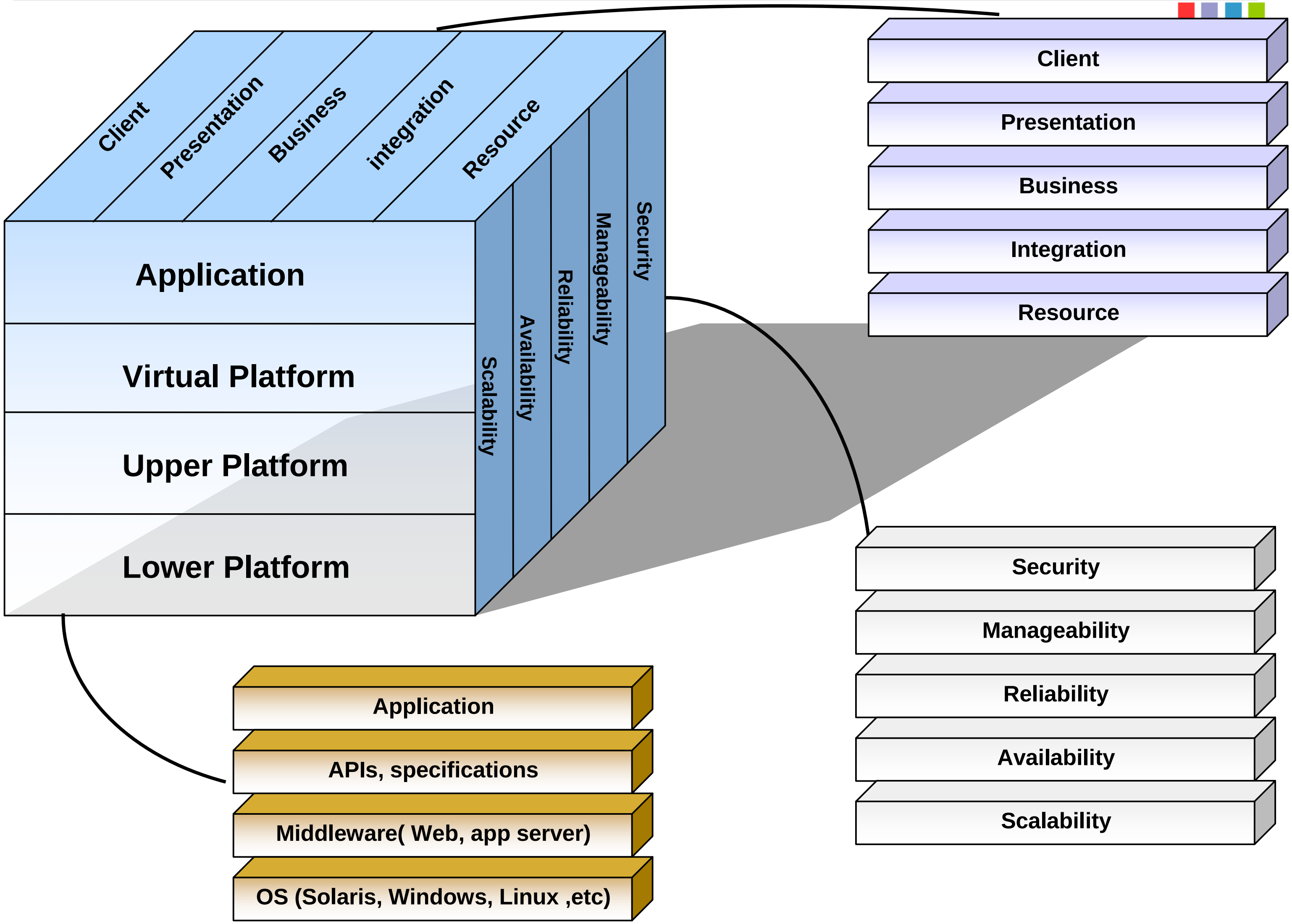
Upper Platform

J2EE Infrastructure
(SunOne, Weblogic, Websphere, etc)

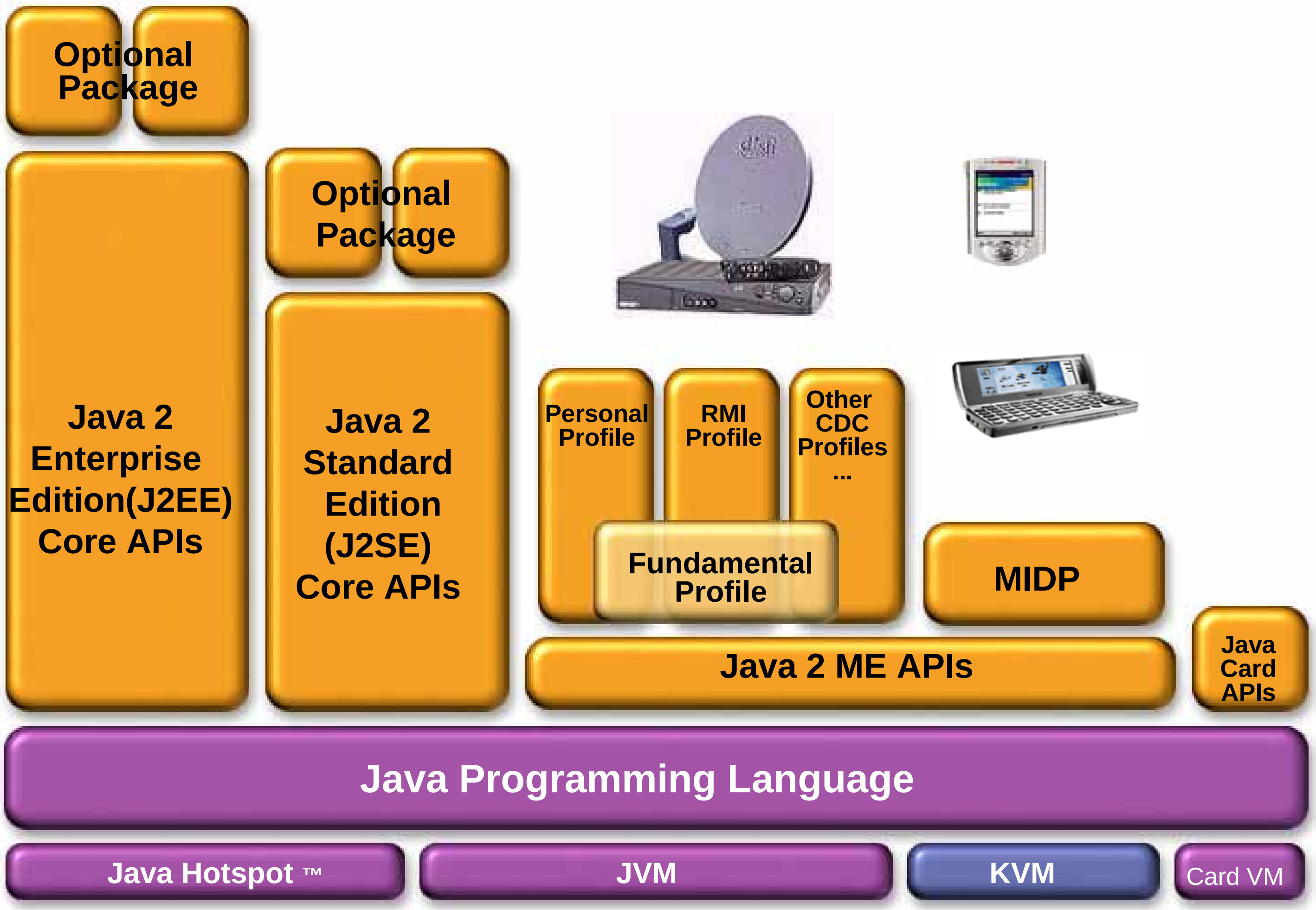
Lower Platform

OS (Unix, Windows, Linux, Mac, others)

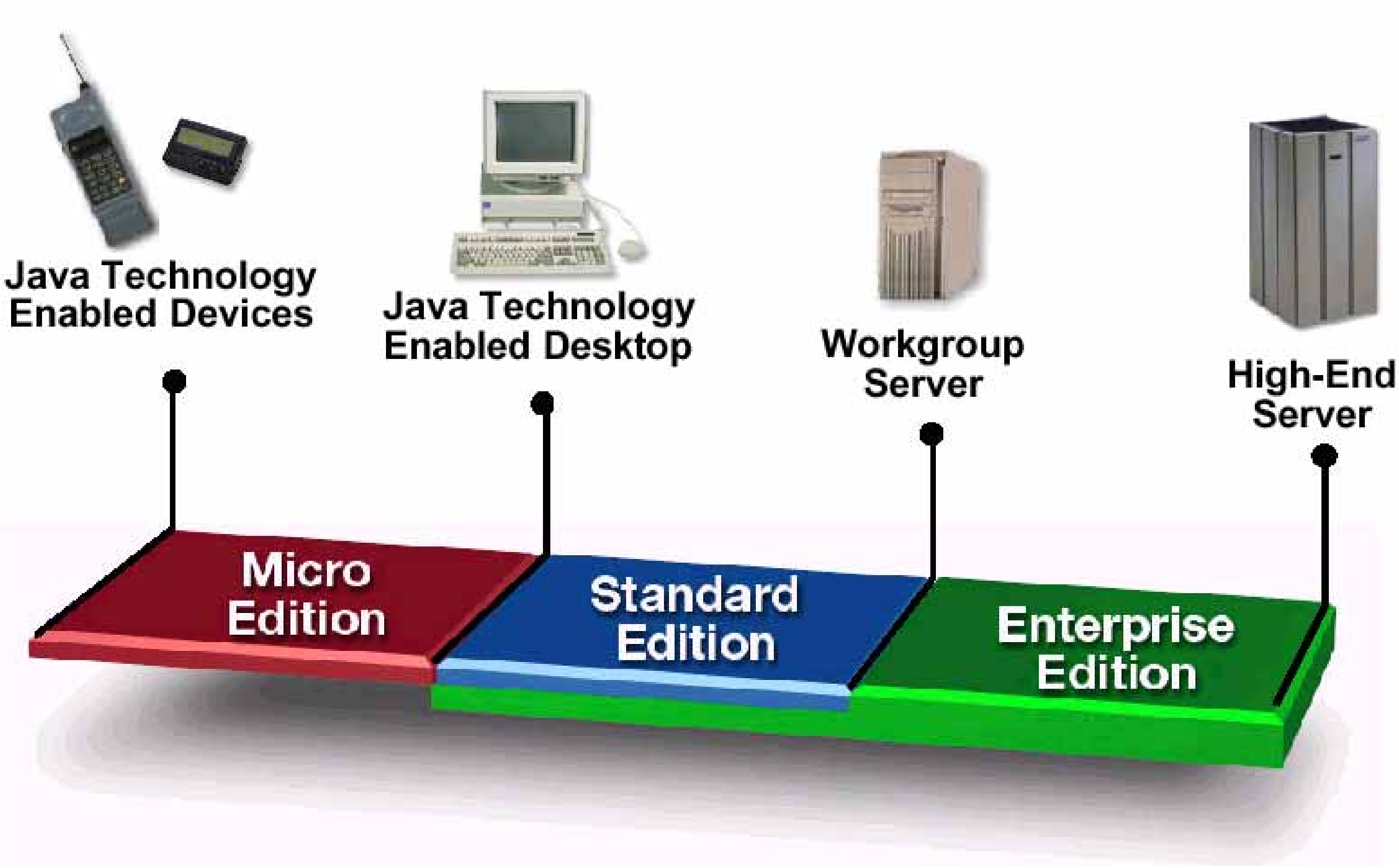
Architecture and the cube



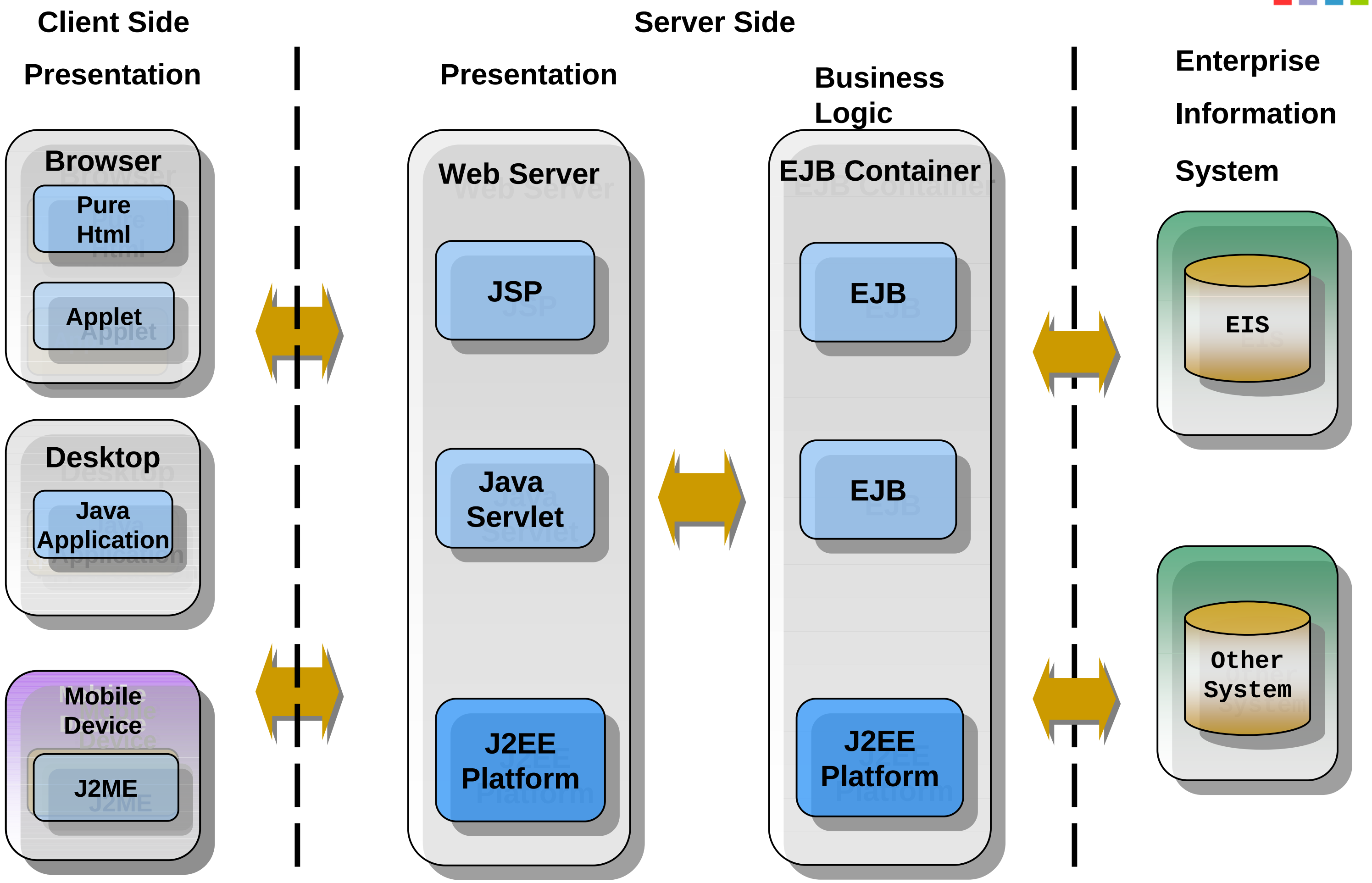
Java Platform Overview



Java is everywhere



J2EE 3-Tier Architecture

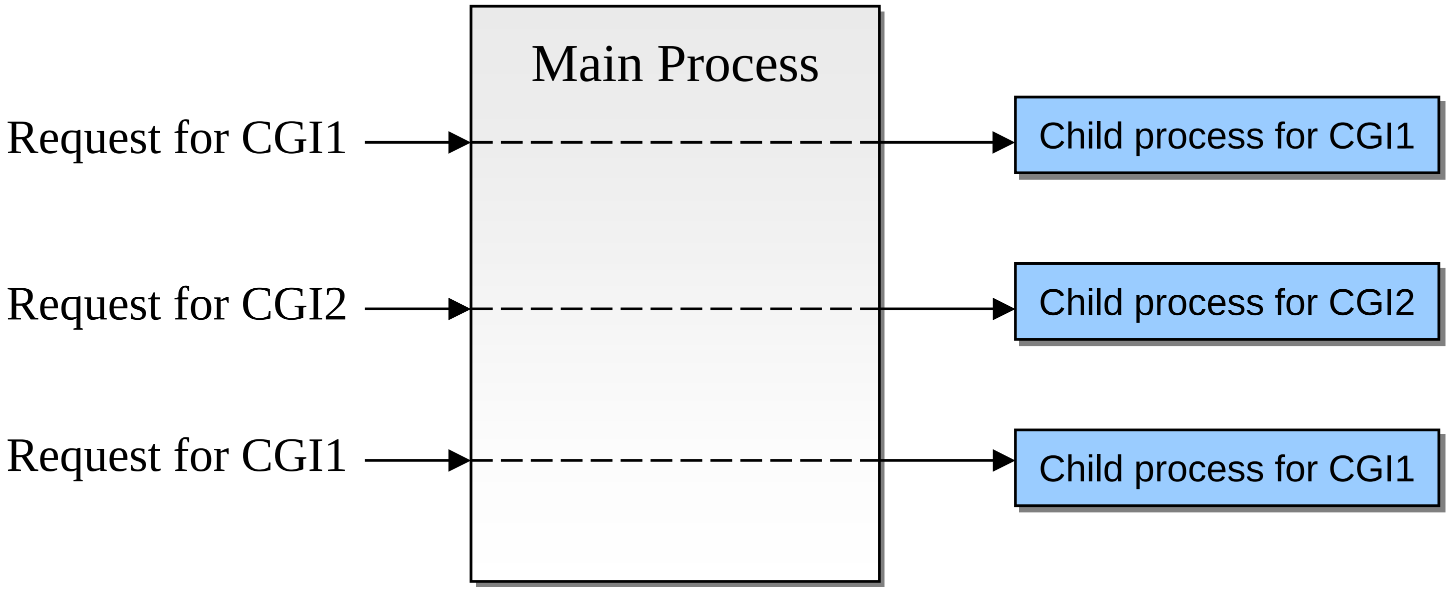


Computing Model is changing

Model	Client/Server	Browser/Server
UI	Application	Browser (html)
Maintain	Hard	Easy
Couple	High	Lower

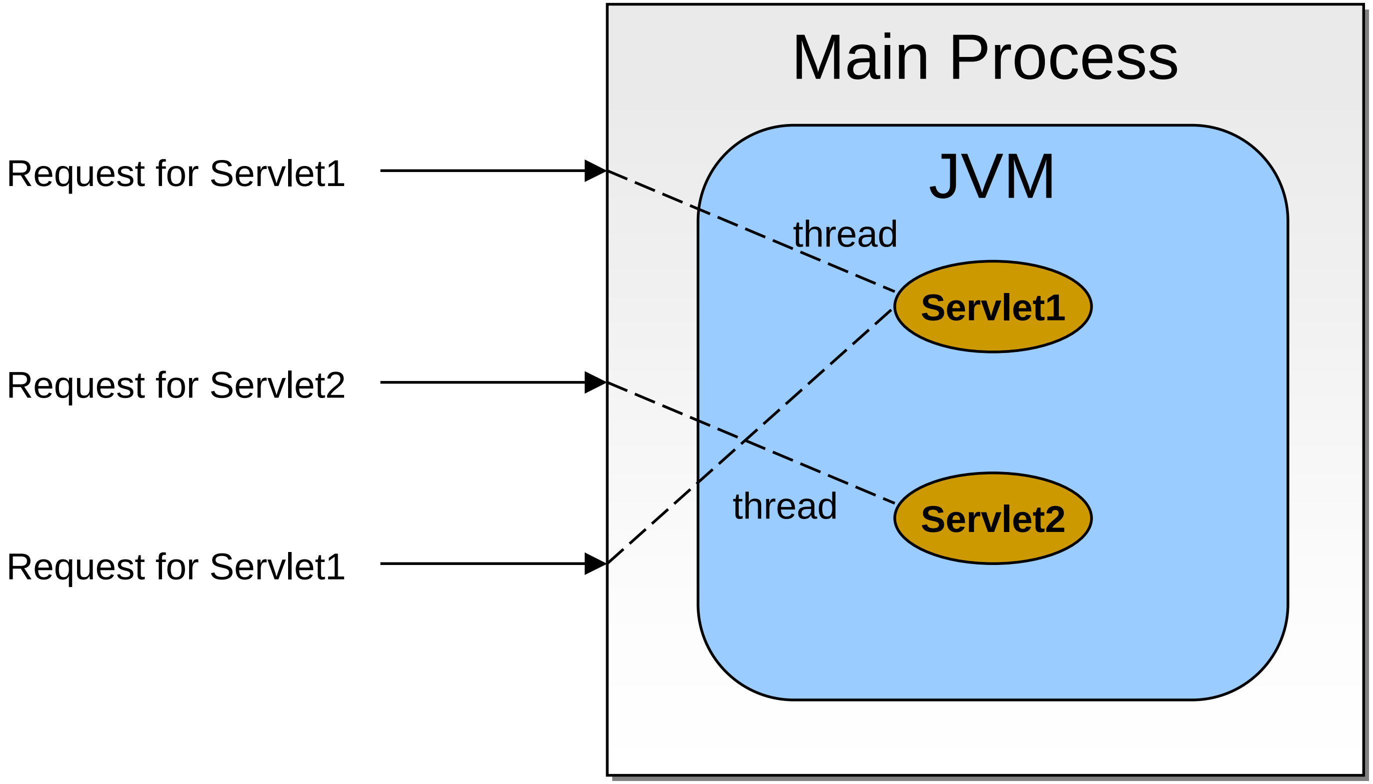
History of Web Applications

- Common Gateway Interface

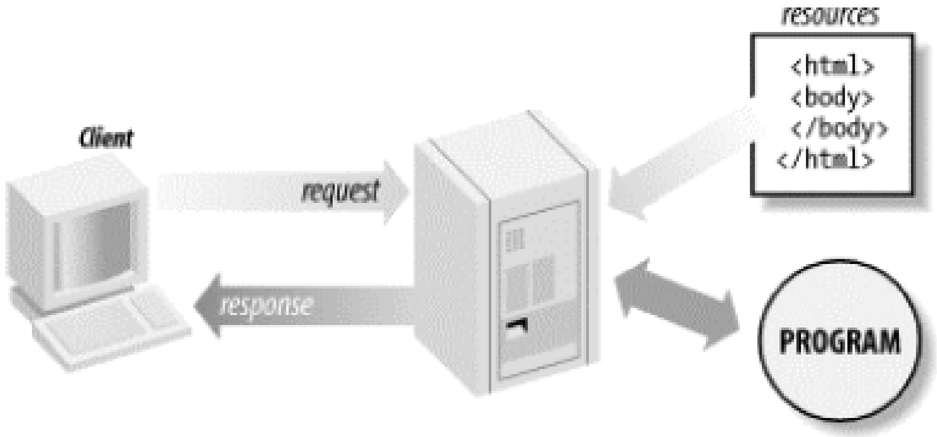


History of Web Applications

- Java Servlet

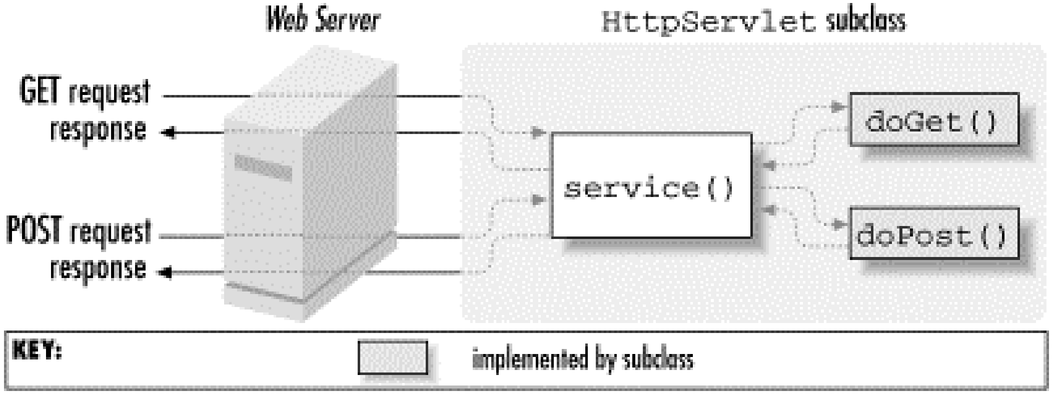


Web Tier – HTTP Request/Response Model



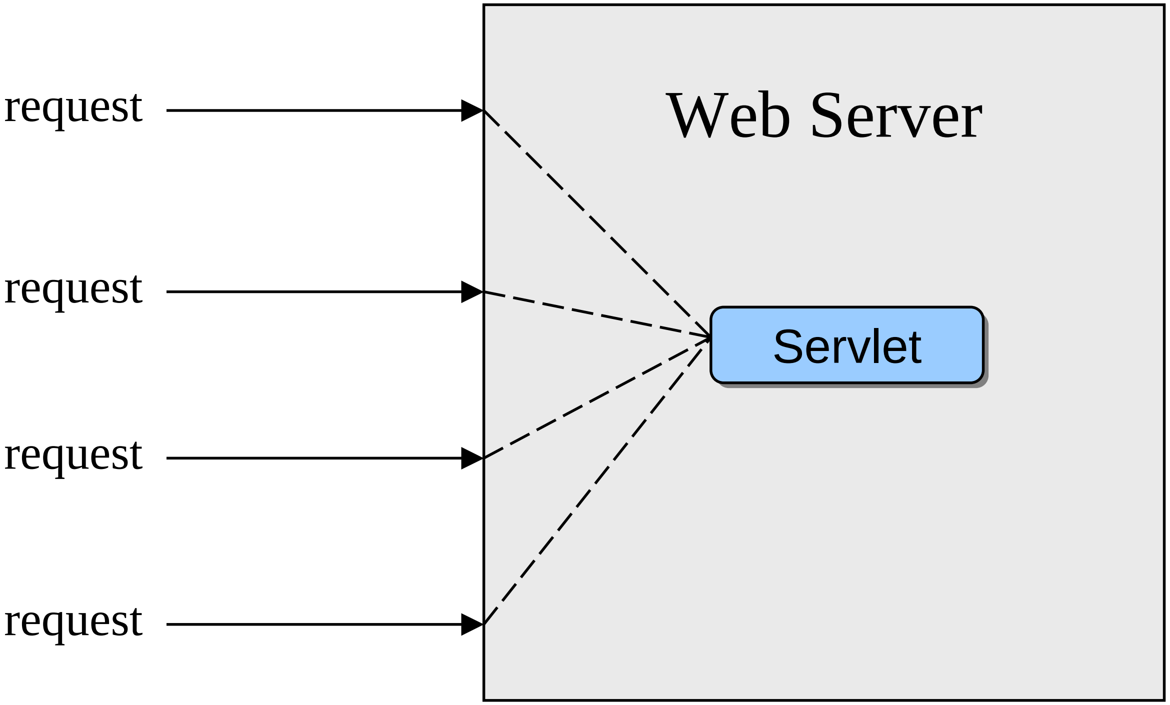
- A Client – a web browser
- Send a *request* for a resource to a server
- Server sends back a *response* corresponding to the resource

The Servlet



@see HelloWorldServlet.java
http://localhost:8081/servlet/HelloWorldServlet

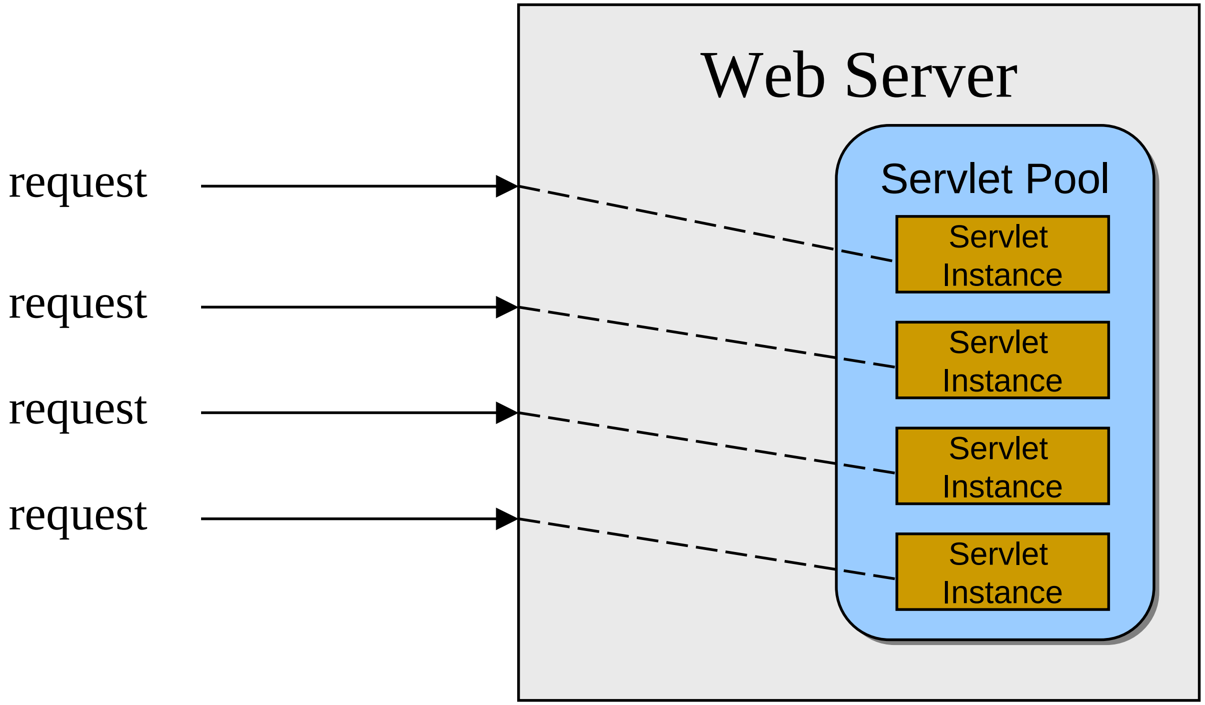
The Servlet – single instance



@see SingleInstance.java

http://localhost:8081/servlet/SingleInstance

The Servlet – Single Thread Model



Web Tier - What's JSP?

- JavaServer Pages ,for short—is a Java-based technology that simplifies the process of developing dynamic web sites.

- A JSP page contains standard markup language elements, such as HTML tags, just like a regular web page.

- A JSP page also contains special JSP elements that allow the server to insert dynamic content in the page.

HTML vs. JSP

HTML

```
<html>

<head>

<body>

<LI><A>Why Use JSP</A></LI>

</body>

</html>
```

JSP

```
<html>
<head>
<body>
<jsp:useBean…/>
</body>
</html>
```

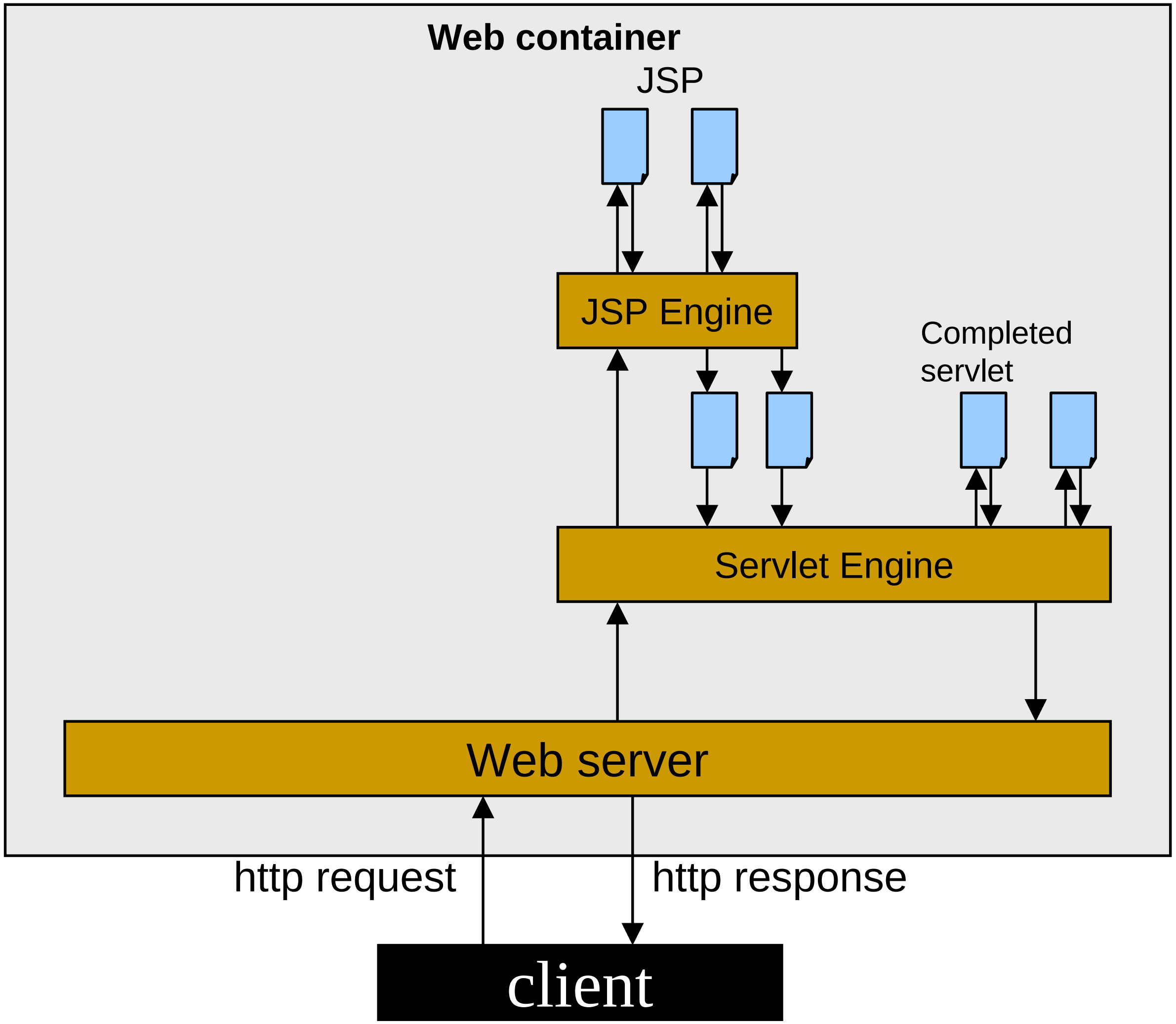
Why use JSP?

- Embedding Dynamic Elements in HTML Pages

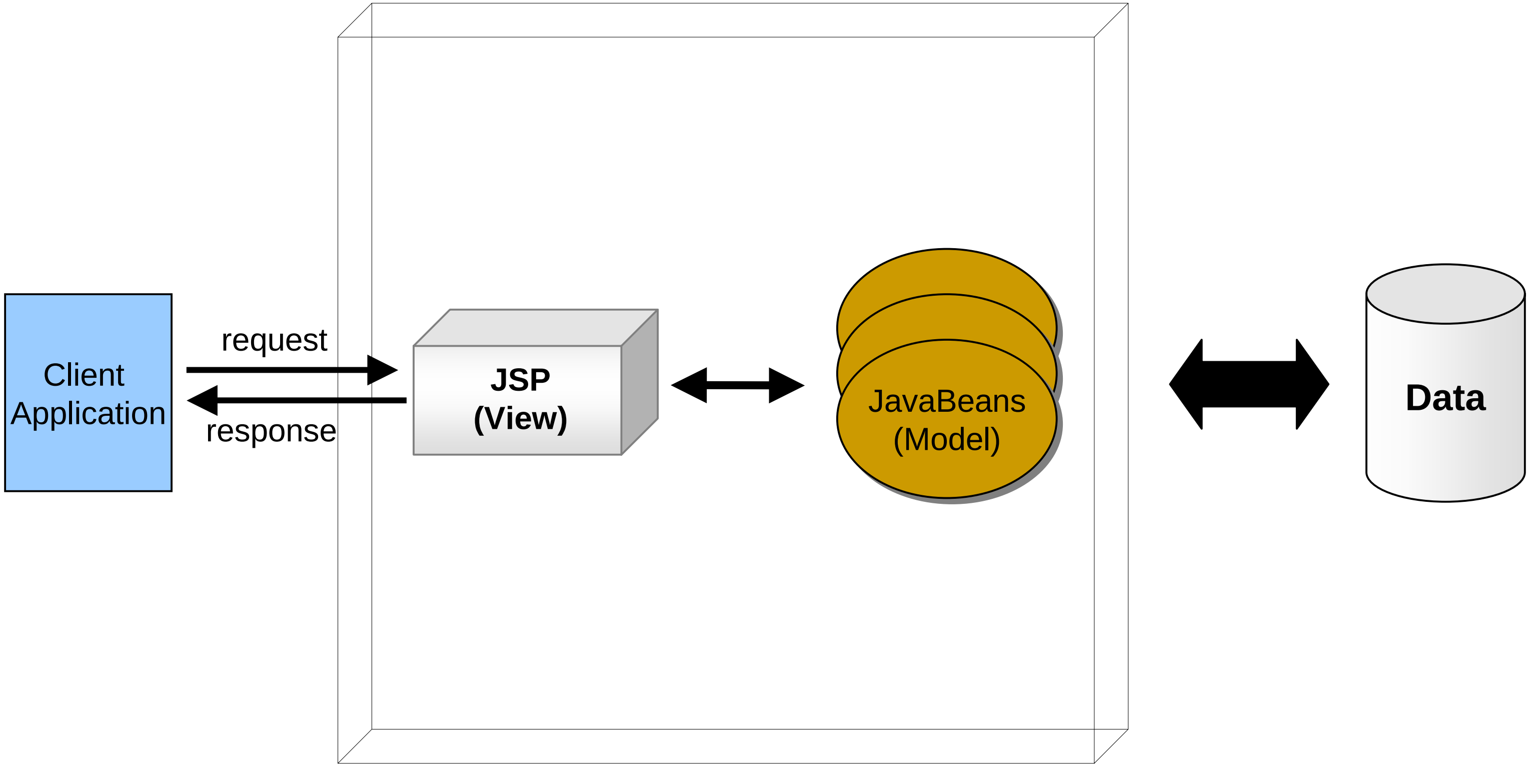
- Compilation
  - CGI/Perl require the server to load an interpreter and the target script each time the page is requested.
  - a JSP page is always compiled before it's processed by the server.

- Integration with Enterprise Java APIs
  - JDBC, JMS, EJB, etc

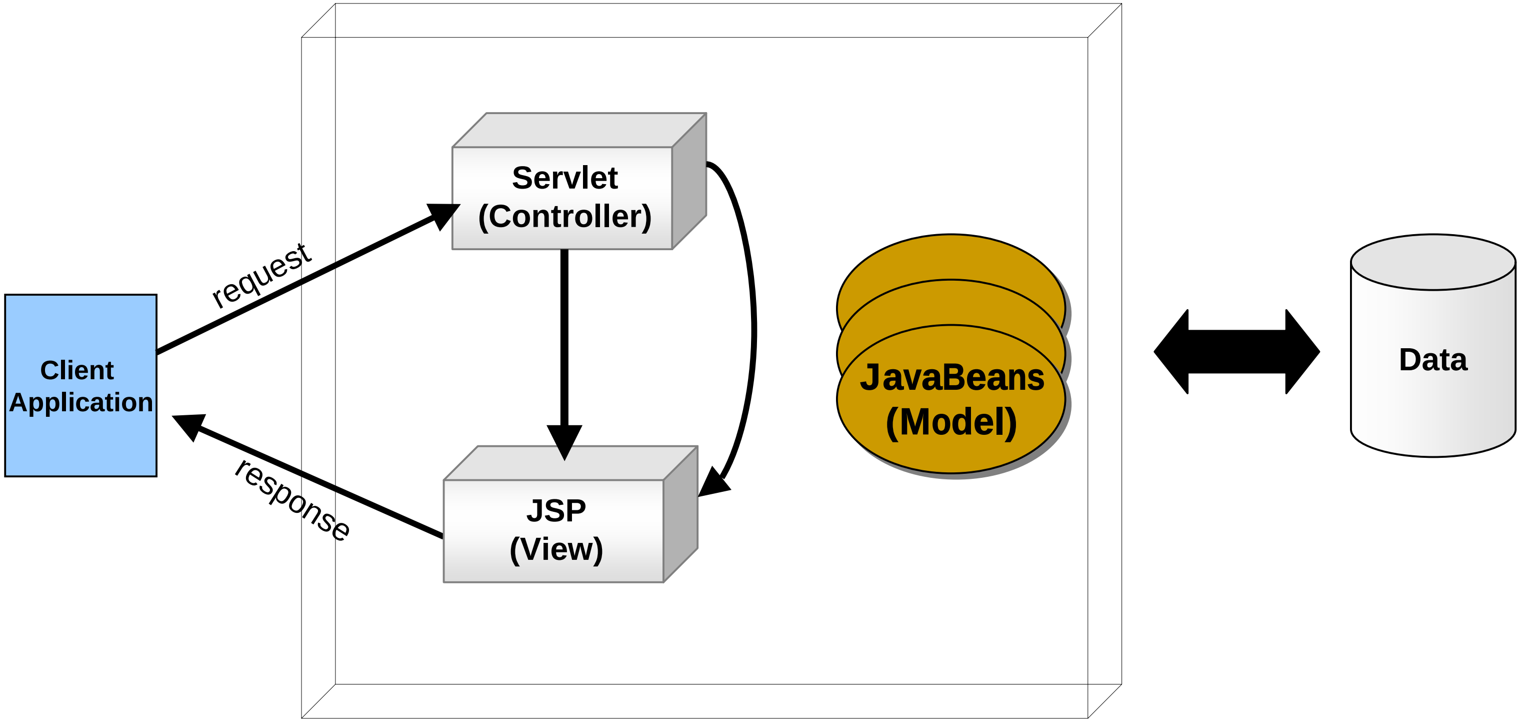
Inside the web server



Web Programming Model - 1

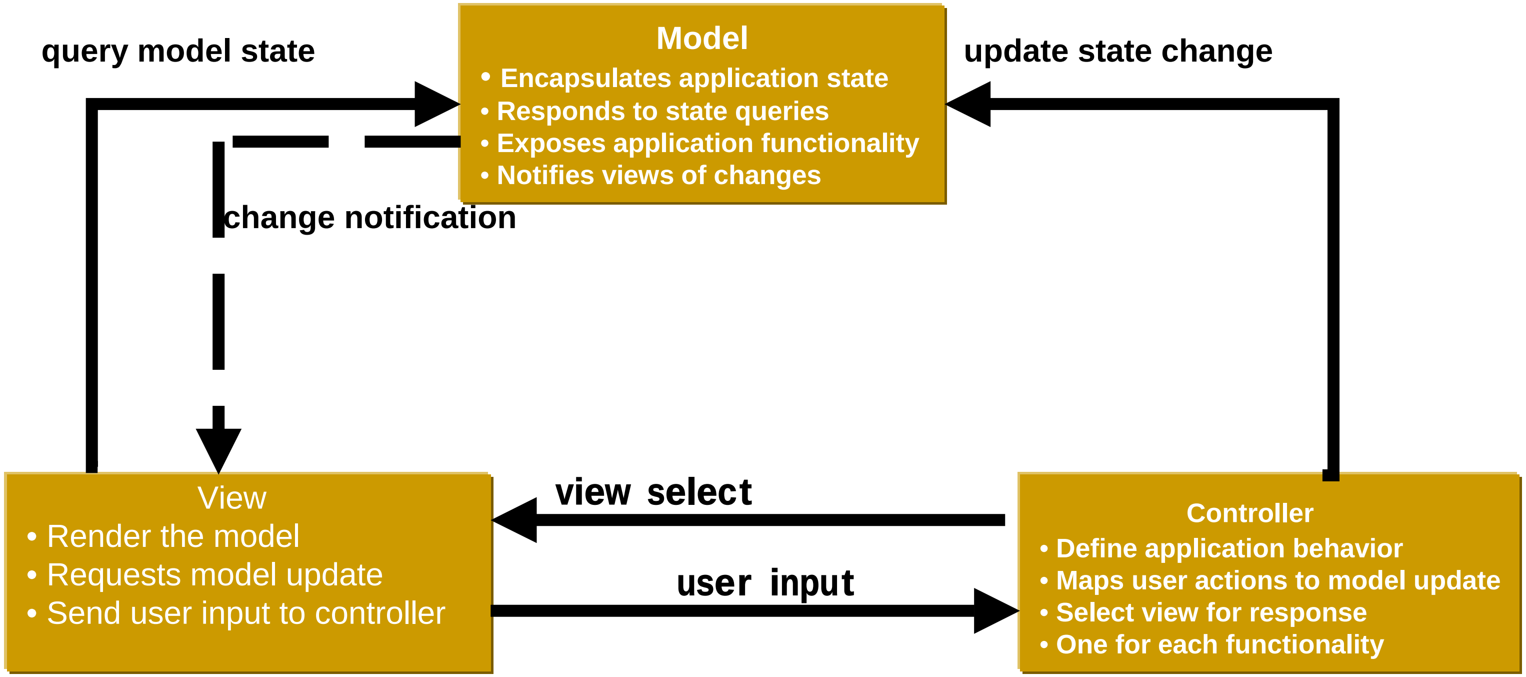


Web Programming Model - 2



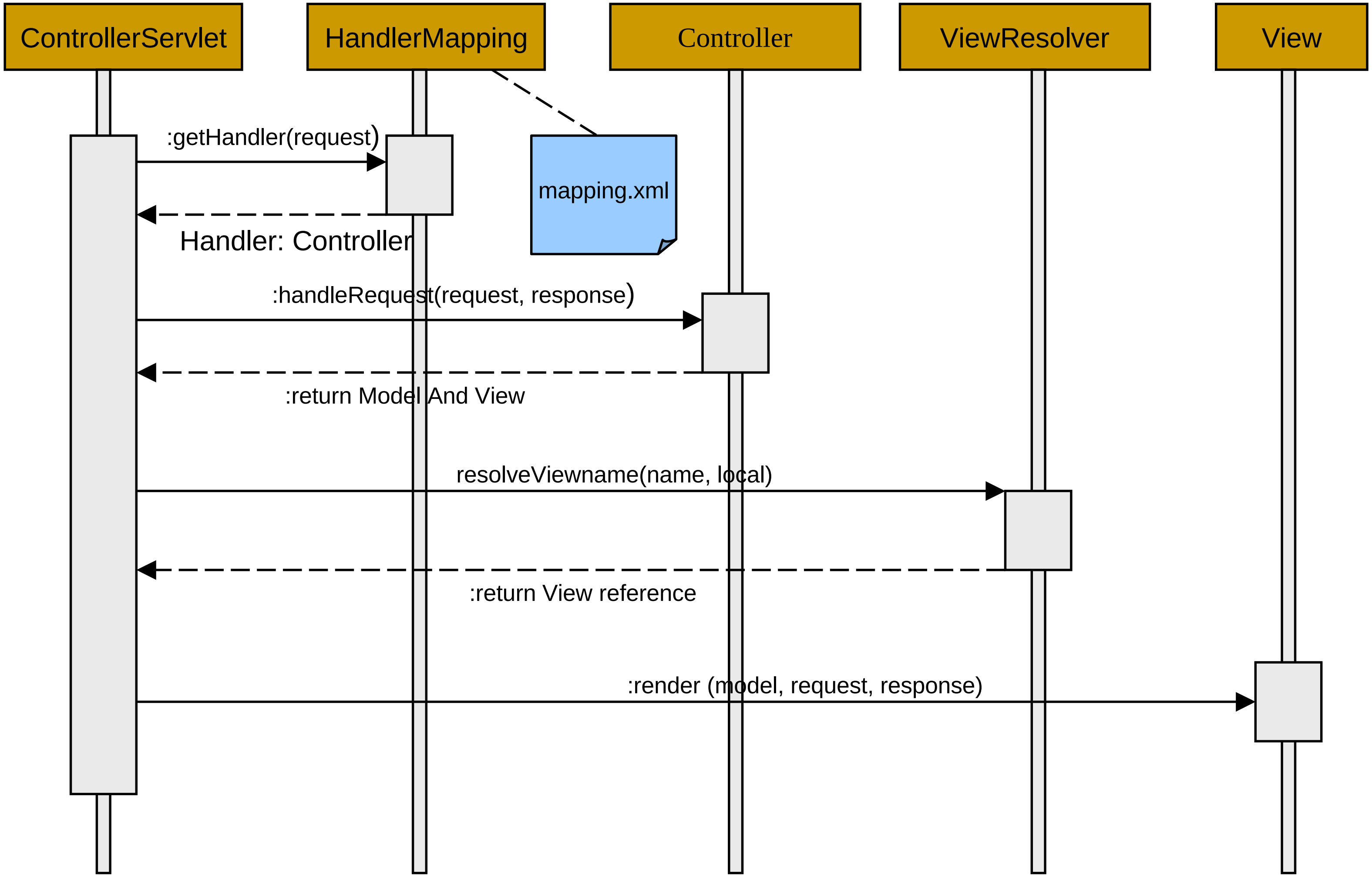
@see ControllerServlet.java welcome.jsp login.jsp

MVC (Model-View-Controller)



query model state
Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
update state change
change notification
View
• Render the model
• Requests model update
• Send user input to controller
view select
user input
Controller
• Define application behavior
• Maps user actions to model update
• Select view for response
• One for each functionality

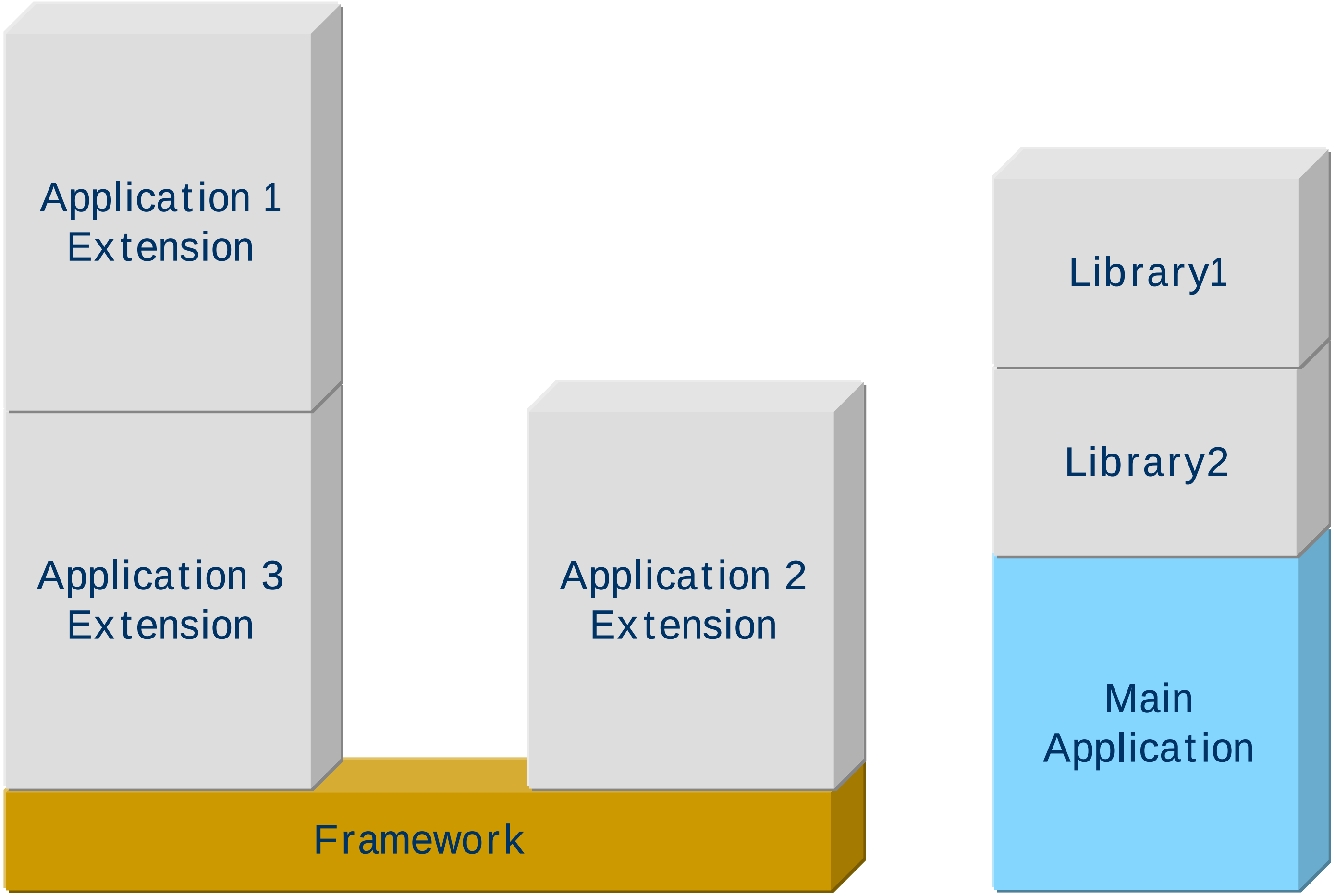
Basic MVC Control Flow



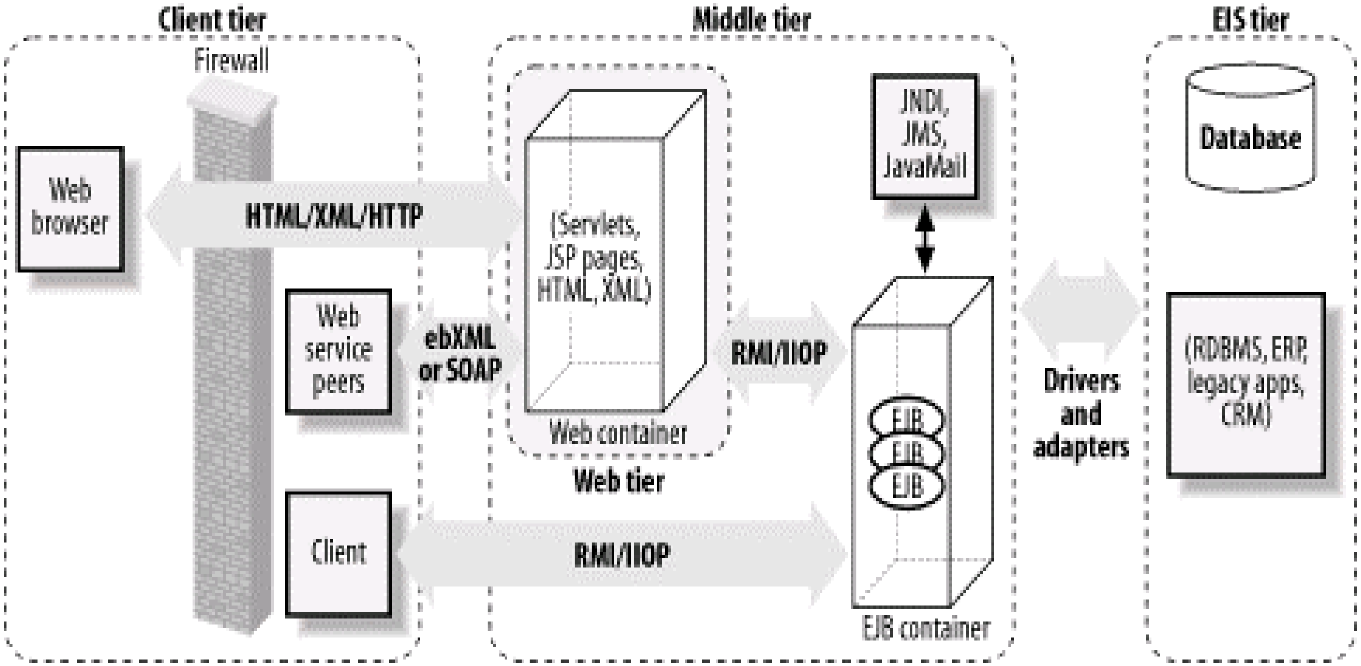
Open Source MVC Framework

- Struts www.apache.org
- WebWork www.opensymphony.com
- Maverick http://sourceforge.net/project/mav

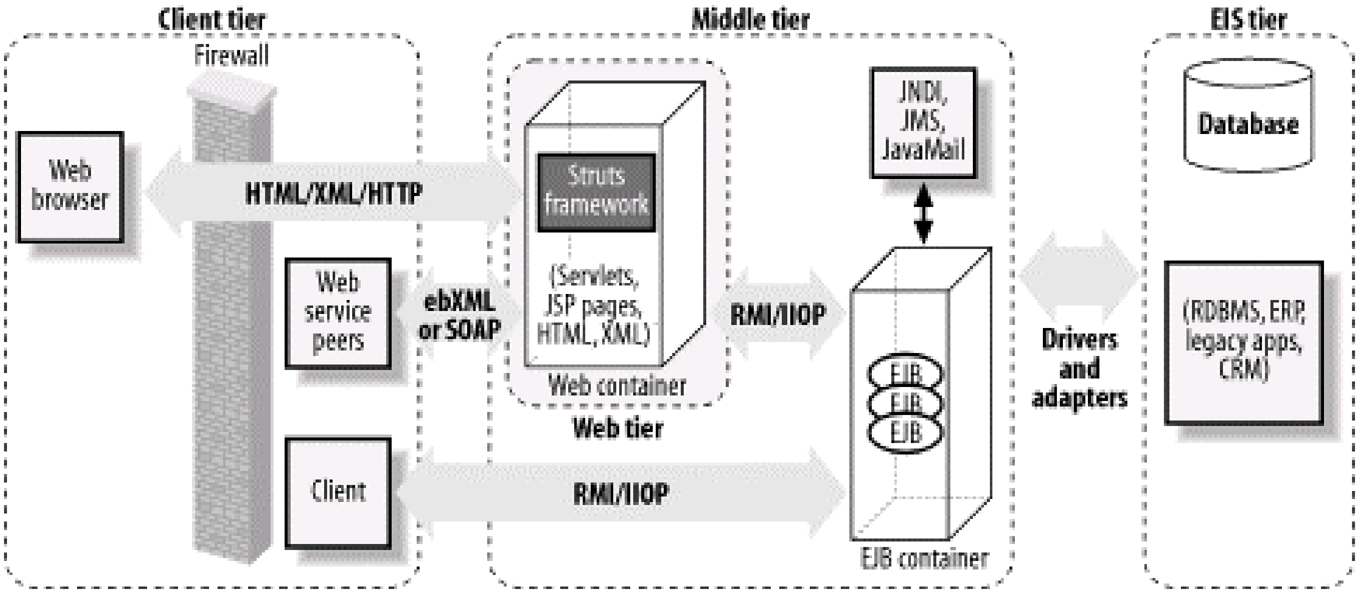
Framework vs. Component



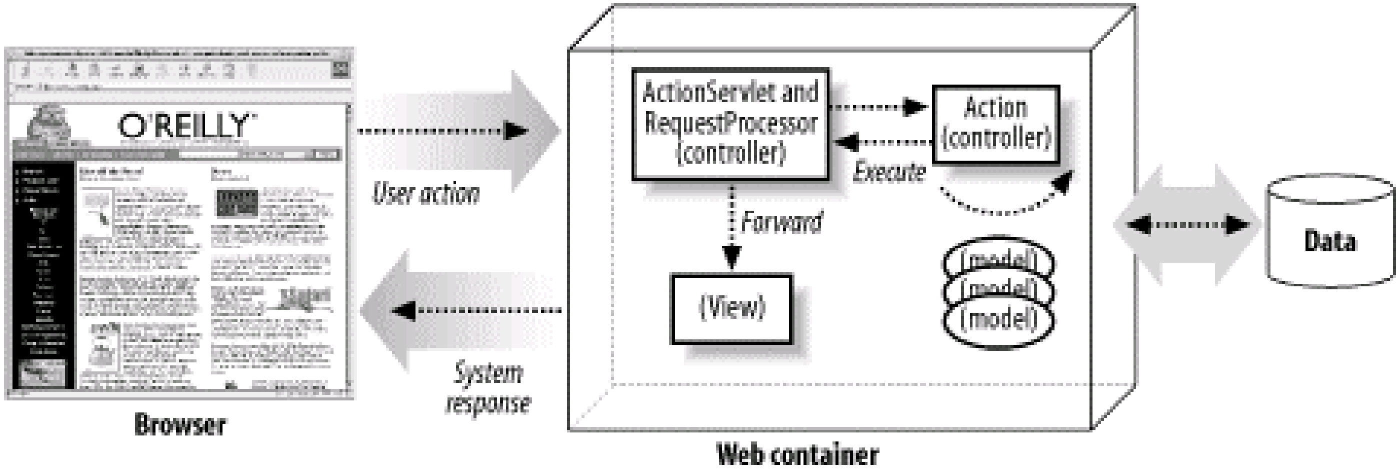
Functional Application Tier



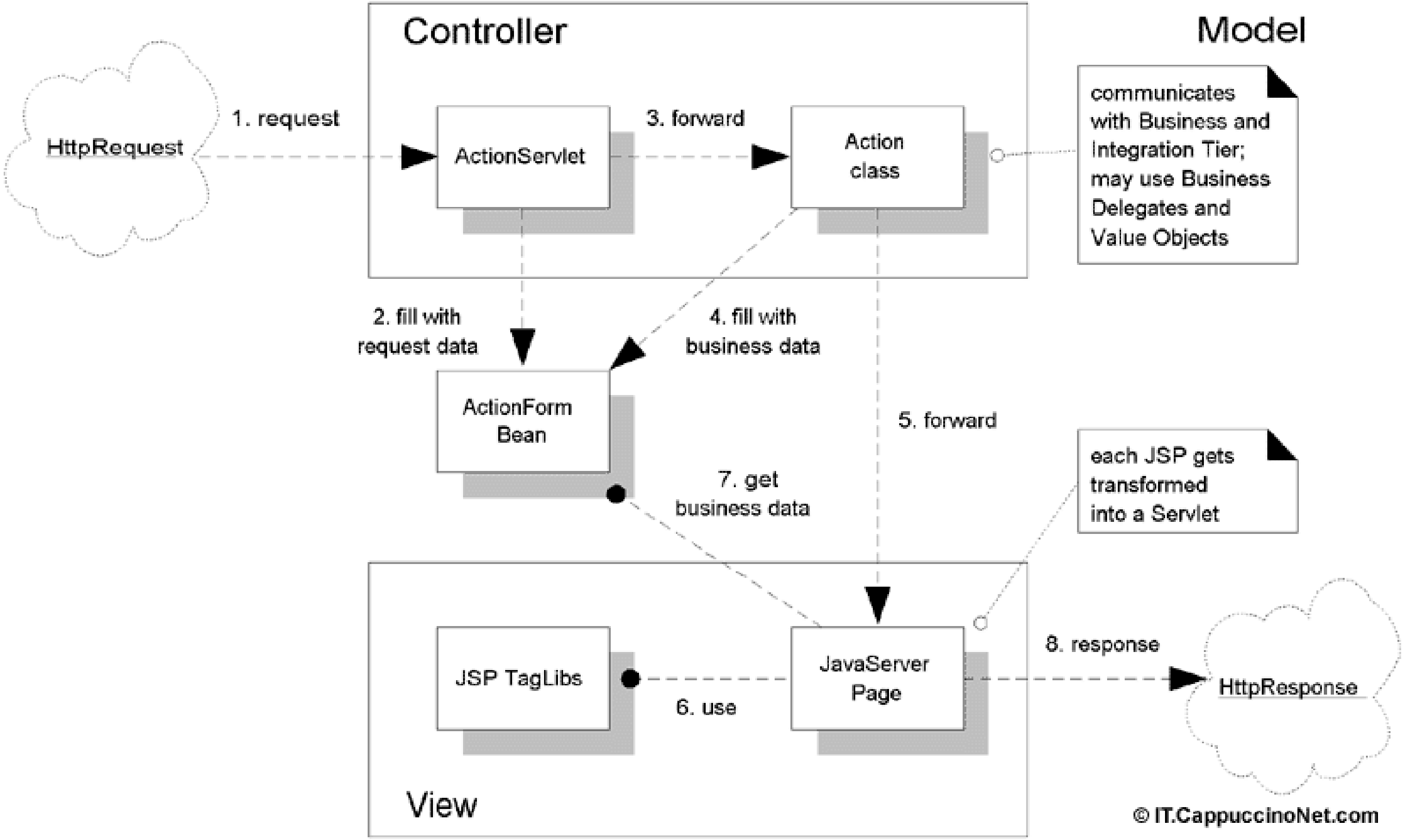
The Struts framework is used within the web tier



The execute() method call by controller



Struts Architecture

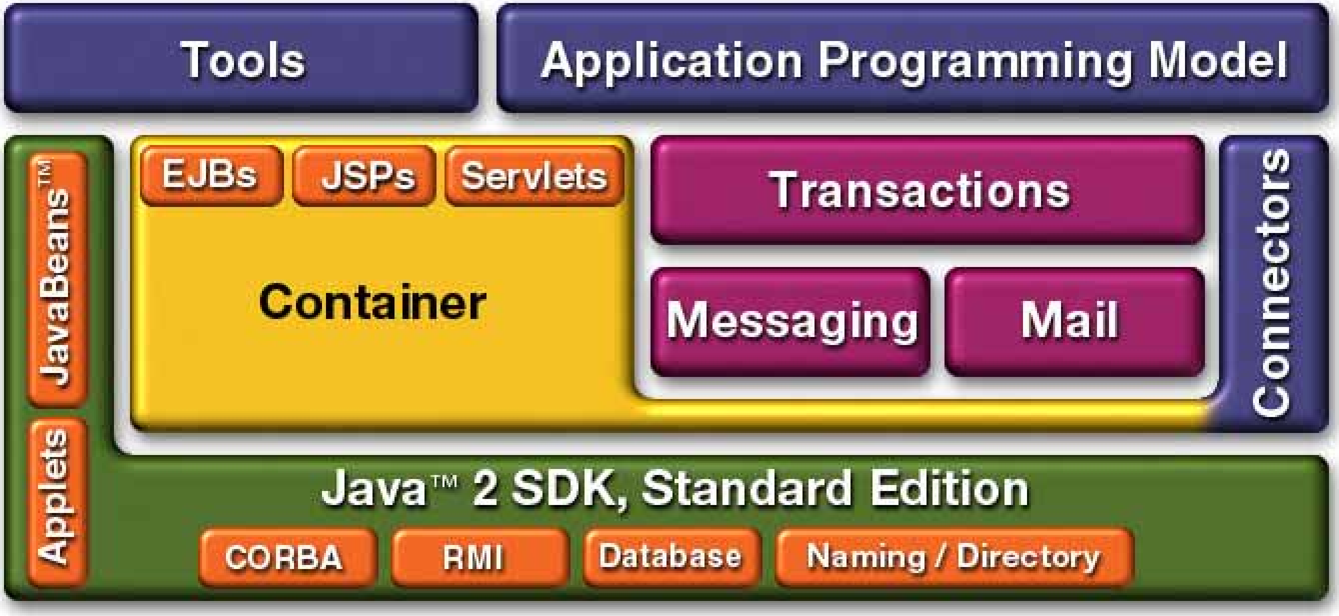

Controller
Model
HttpRequest
1. request
ActionServlet
3. forward
Action class
communicates with Business and Integration Tier; may use Business Delegates and Value Objects
2. fill with request data
4. fill with business data
ActionForm Bean
5. forward
each JSP gets transformed into a Servlet
7. get business data
JSP TagLibs
6. use
JavaServer Page
8. response
HttpResponse
View

© IT.CappuccinoNet.com

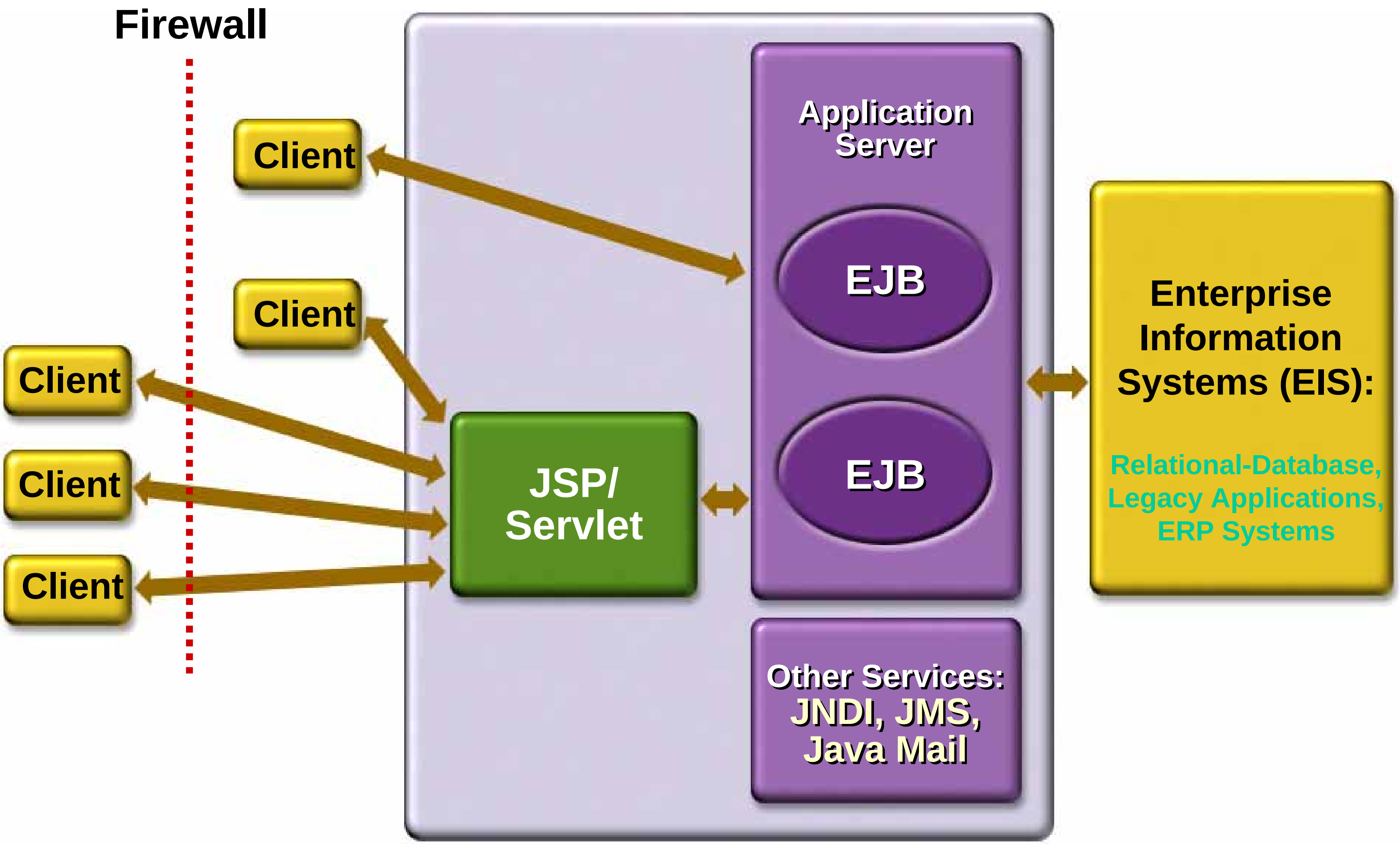
What’s J2EE Technology

- Components:
  - Application clients
  - Applets
  - Web components
  - Business components
- Containers
  - Manage lifecycle of business components
  - Provide a federated view of J2EE APIs
  - Provide runtime support for components

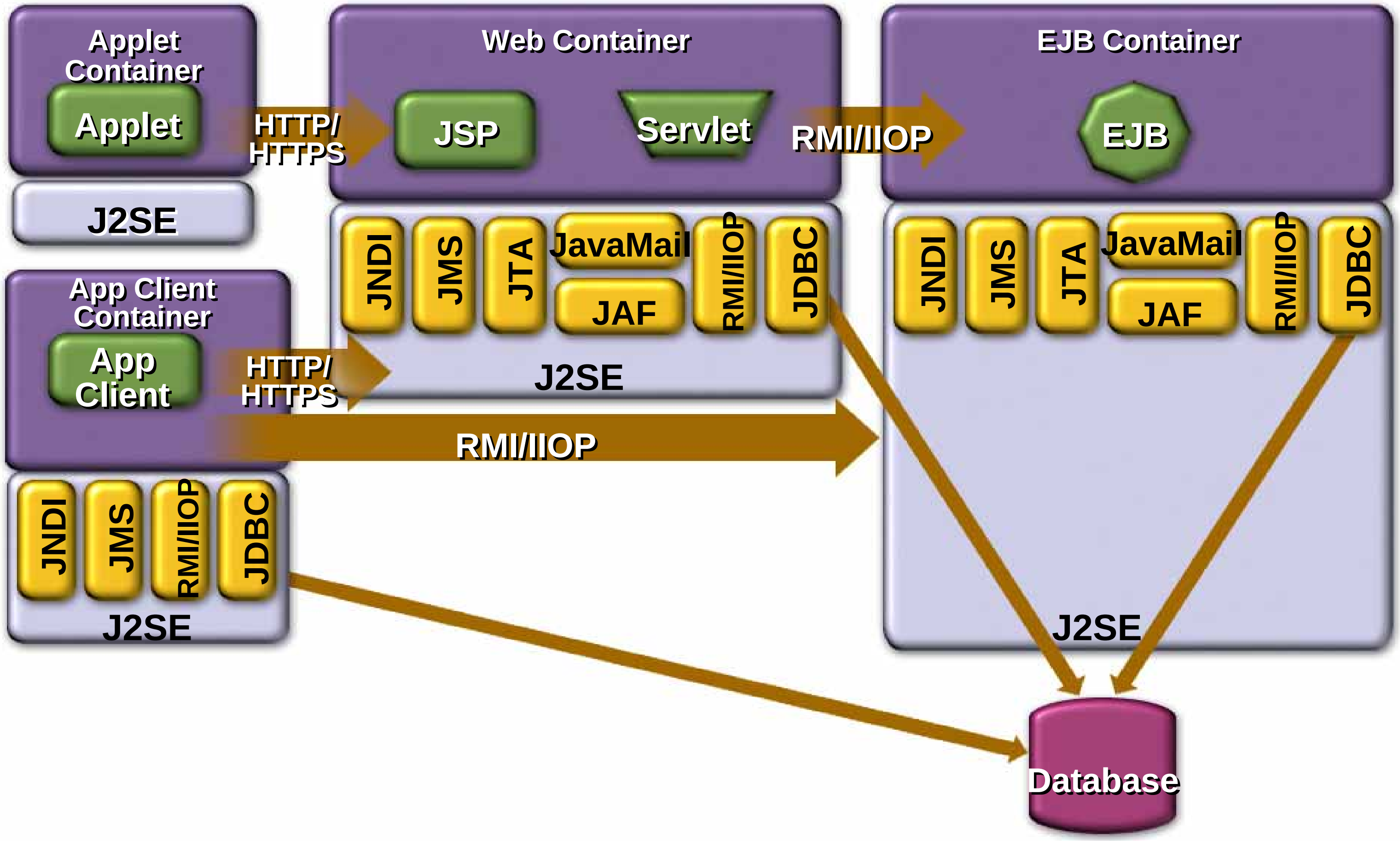
J2EE Architectures



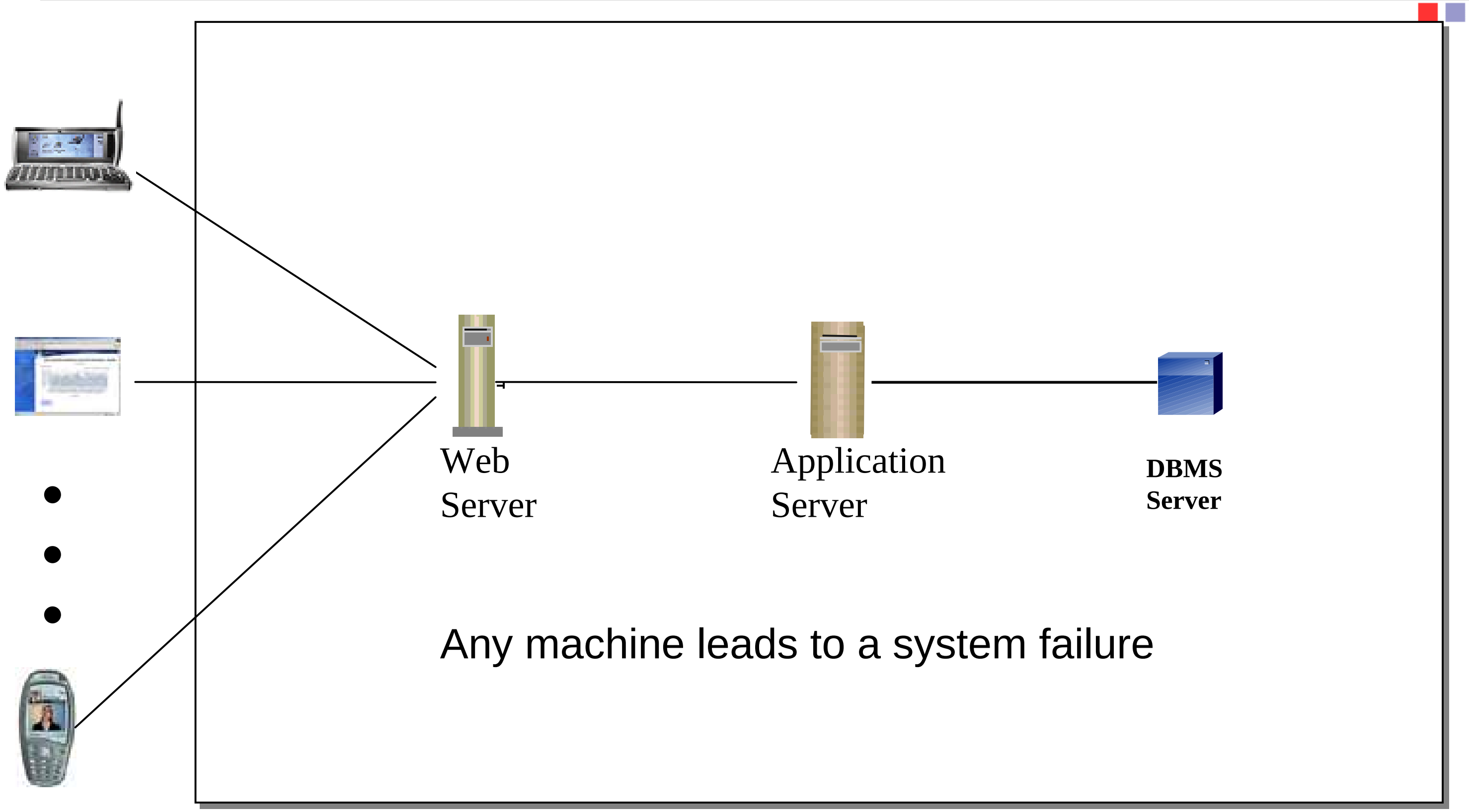
J2EE Application Model


Firewall
Client
Client
Client
Client
Client
Application Server
EJB
EJB
JSP/ Servlet
Other Services: JNDI, JMS, Java Mail
Enterprise Information Systems (EIS):
Relational-Database, Legacy Applications, ERP Systems

J2EE Platform Specification



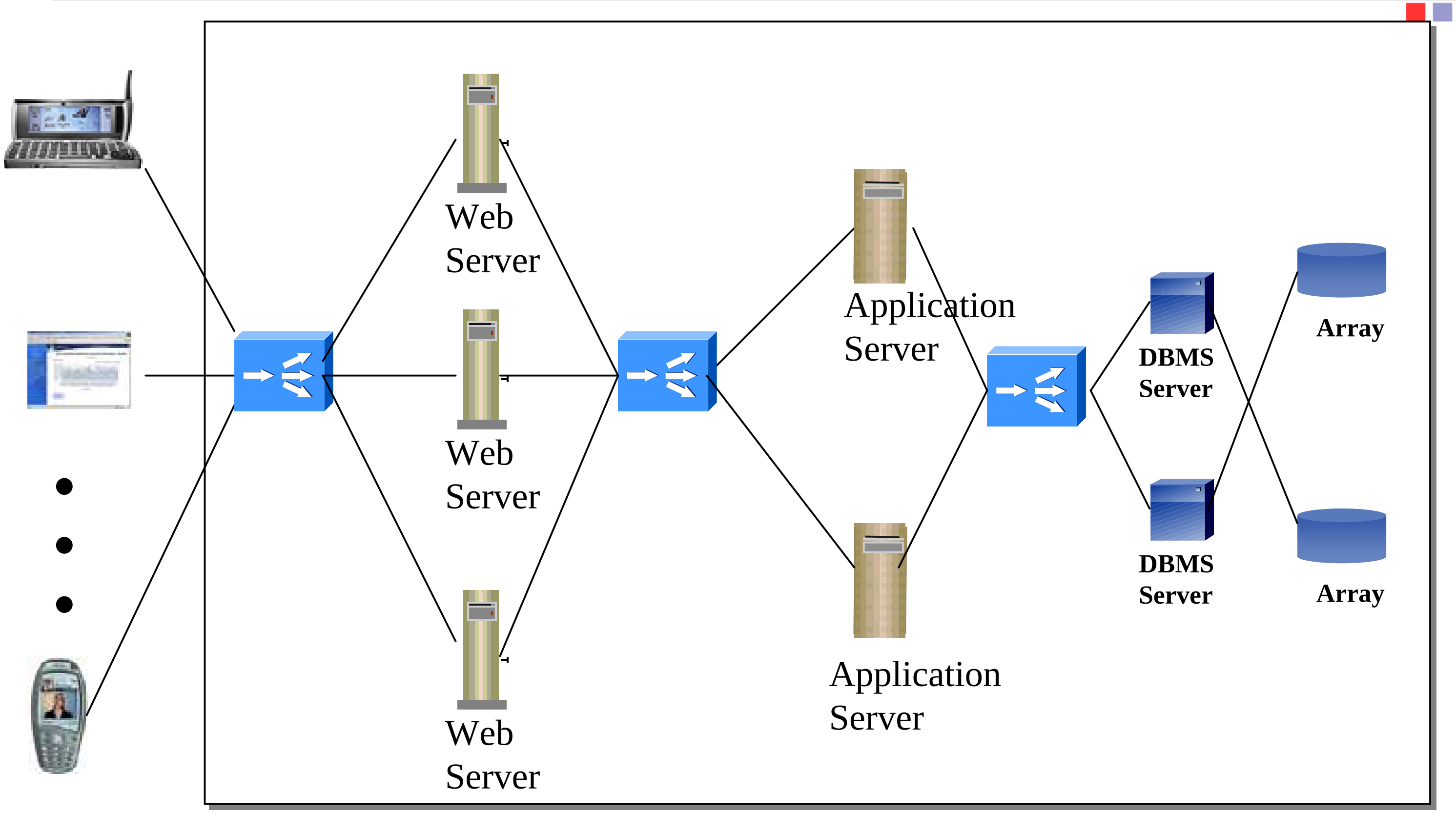
Vertical Scalability



Any machine leads to a system failure

Capacity	↑	Availability	↓
Reliability	↓	Manageability	↑

Horizontal Scalability



Capacity ↑

Reliability ↑↑

Availability ↑↑

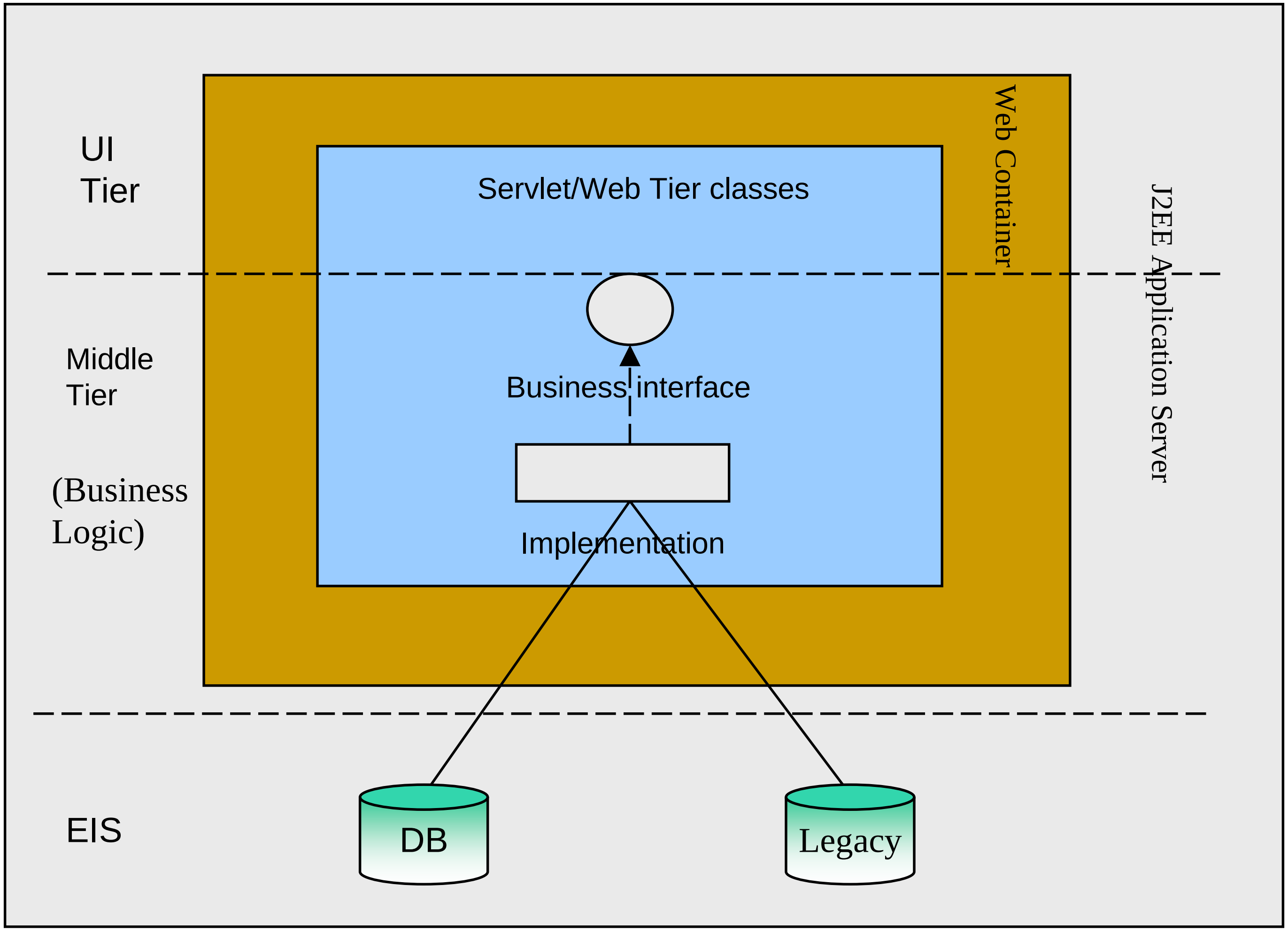
Manageability ↓

Design J2EE-Based Application

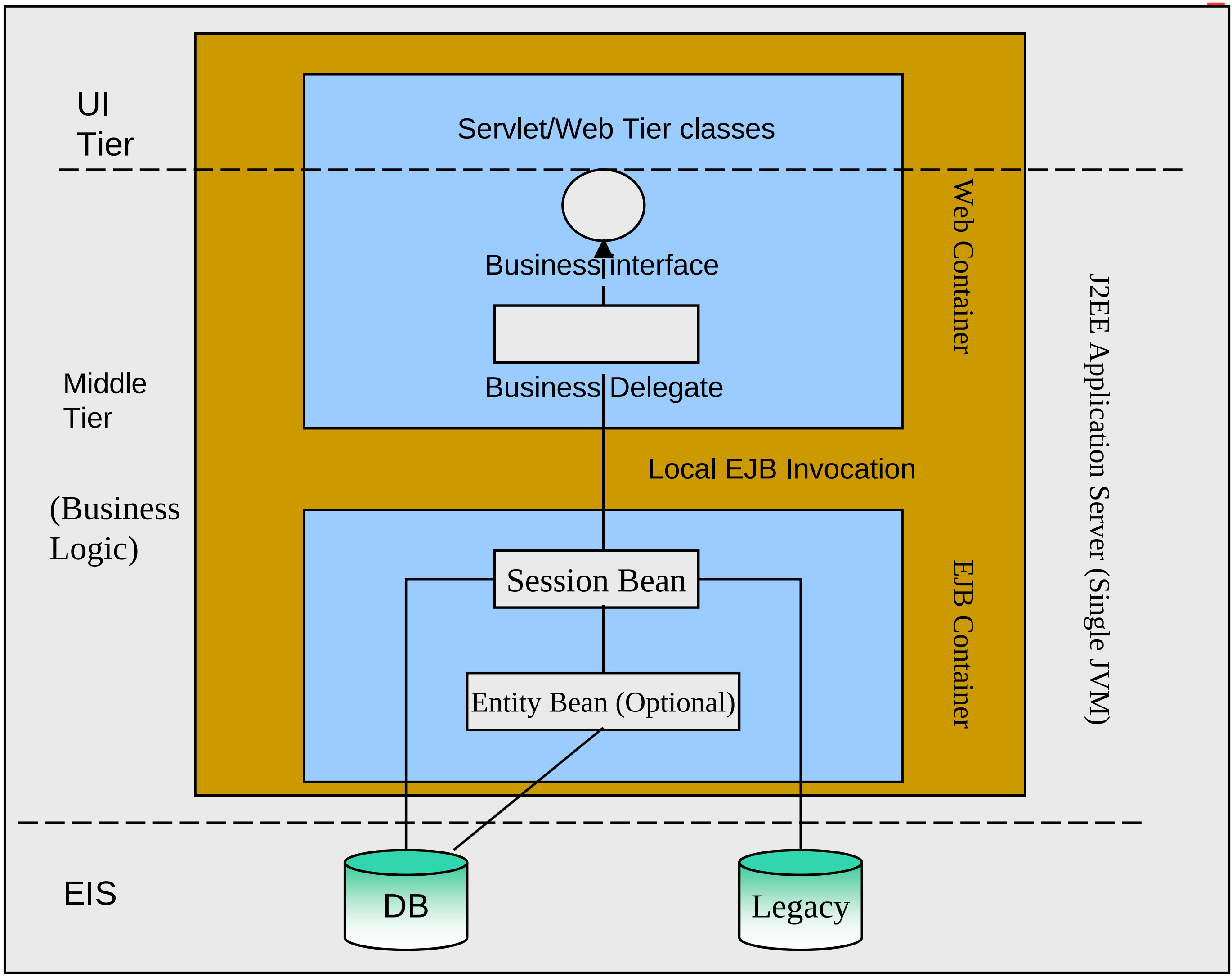
- EJB is not the only technology for implementing the middle tier in J2EE applications
- Design to Java interfaces, not concrete classes, and not technologies

Non-distributed Architectures

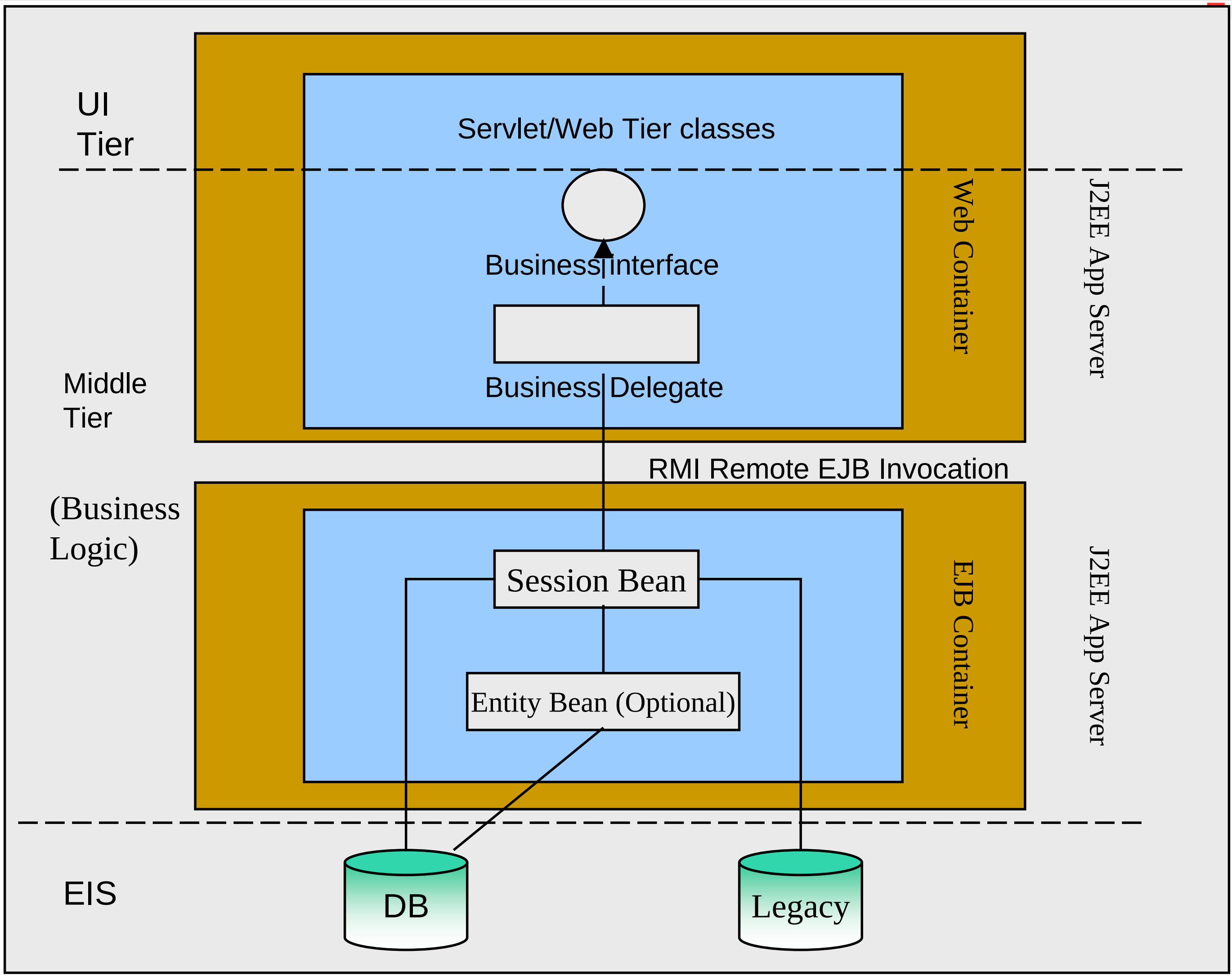
Web Application with Business Component Interfaces



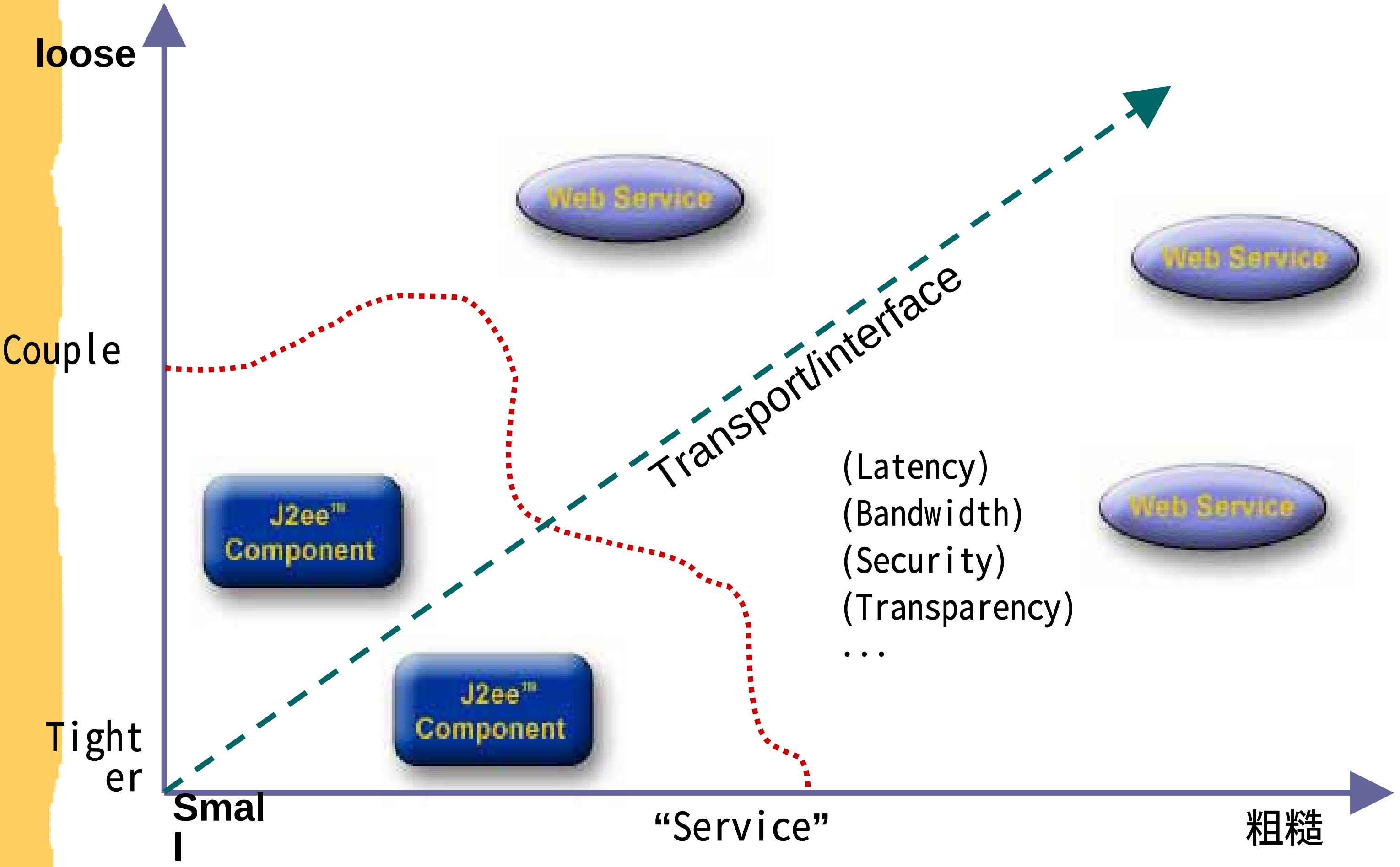
Web Application that Accesses Local EJBs



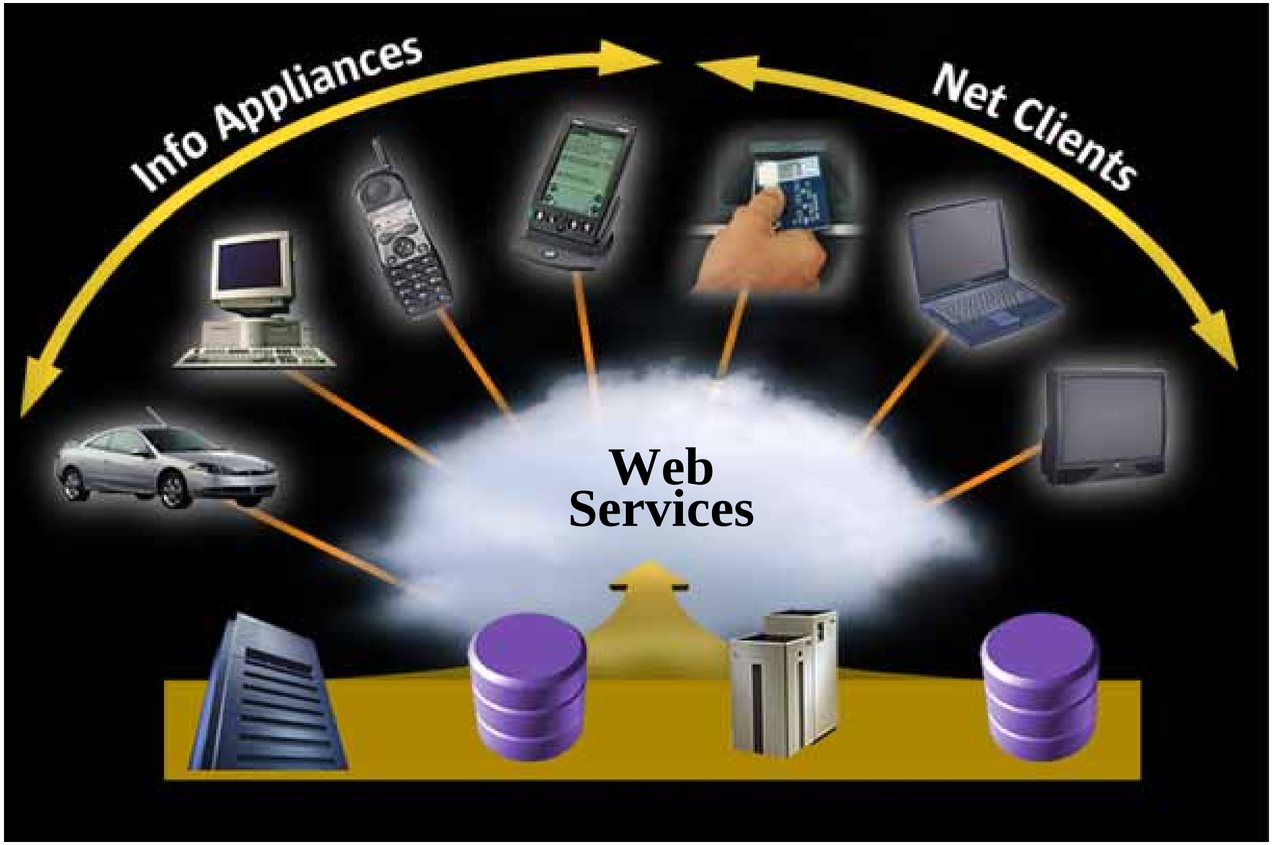
Distributed Architectures



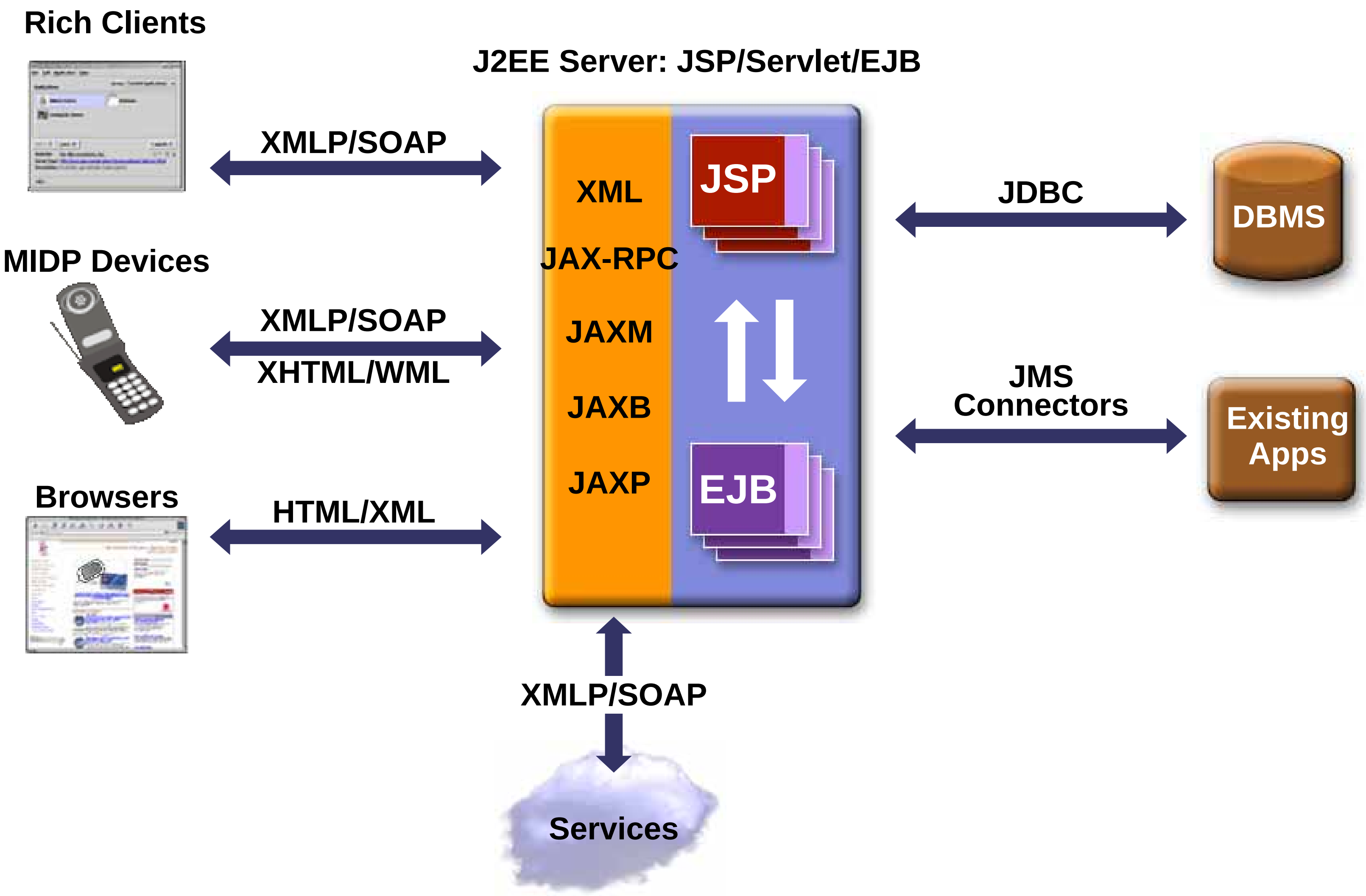
The future of J2EE



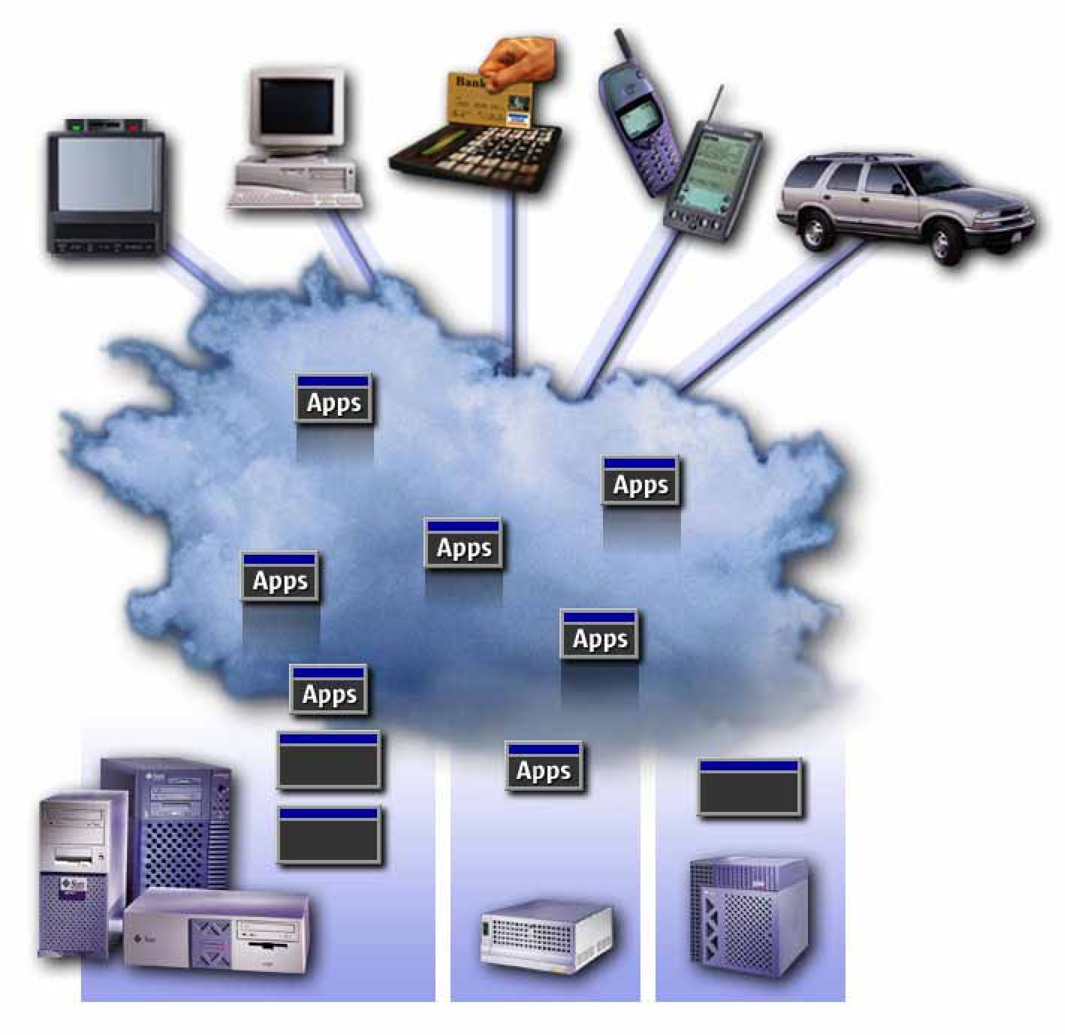
Service-Oriented Architecture



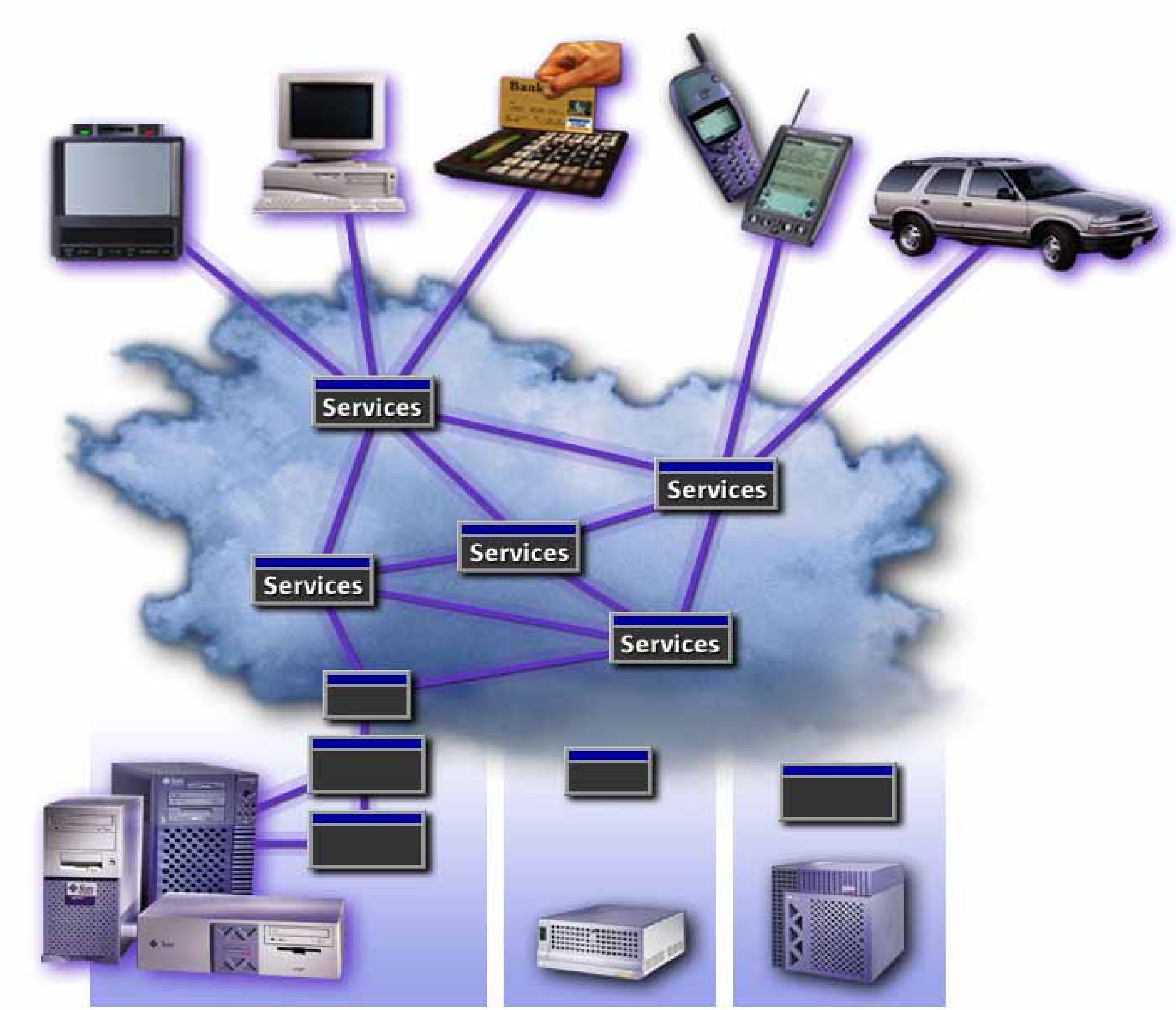
New J2EE Architecture



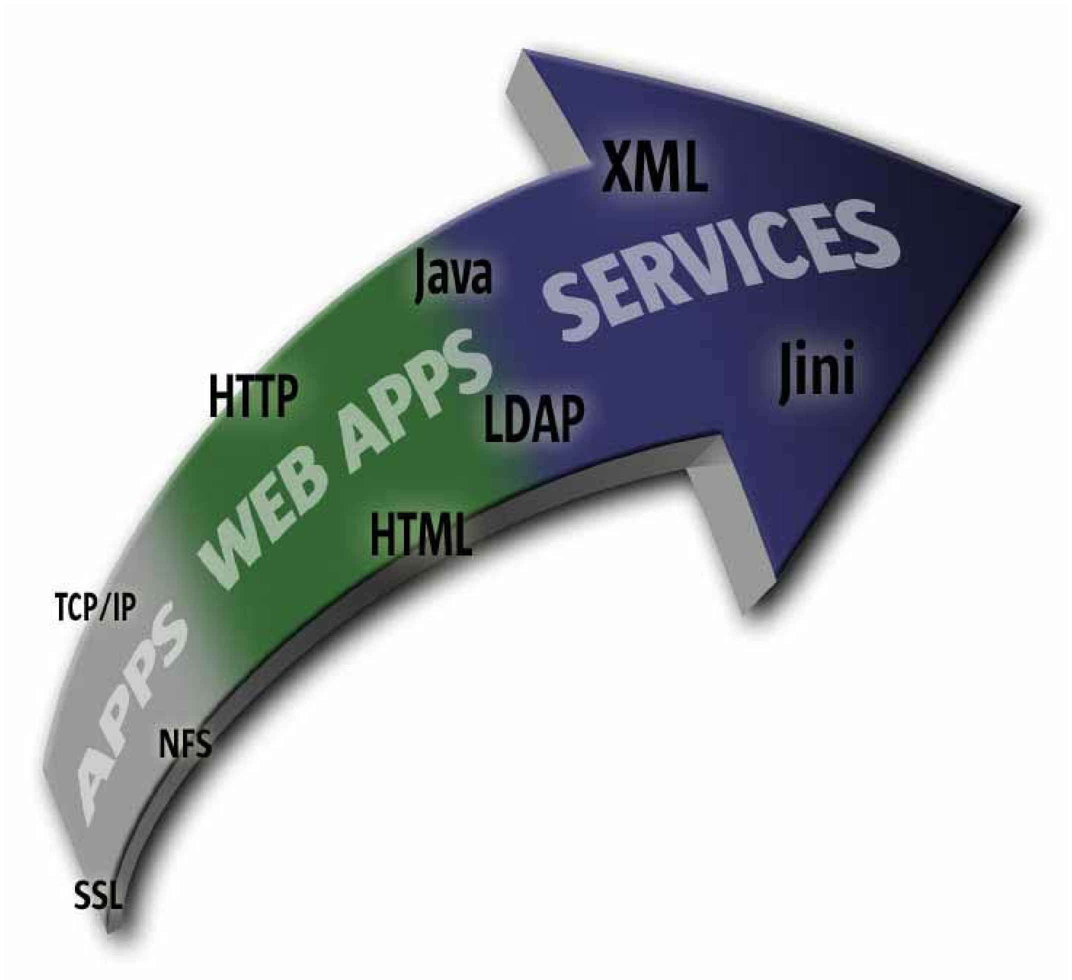
Application Driven



Service Driven



Service on demand



Builder a team

- **Markup developer**
  This will encompass HTML/XHTML development and possibly JavaScript.
- **Presentation developer**
  Responsible for more technical presentation-tier work, rather than designing good-looking HTML.
- **Web-tier Java developer**
  Responsible for MVC framework action classes, JSP tag handlers, and web-tier helper classes.

Builder a team

- **Business object developer**
Responsible for implementing application business logic. Business object developers will use EJB where appropriate, and should have a sound understanding of EJB.
- **Data access specialist**
Java developer responsible for efficient DBMS access. Often this role will be taken on by business object developers.
- **DBA**
Specialist in the underlying database. DBAs can provide valuable assistance to J2EE developers in vital areas such as ensuring good performance and correct locking behavior.

Exercises

- Rewrite, compile, and run a program that use the JSP, Servlet
- Rewrite, compile, and run a program that use the MVC design pattern

Think Beyond

- How to apply MVC framework to your project?

Common Abbreviations and Acronyms

- API - Application programming interface
- CORBA – Common Object Request Broker Architecture
- EIS – Enterprise Information System
- EJB – Enterprise JavaBeans
- ERP – Enterprise Resource Planning
- JAR – Java Archive
- J2EE – Java 2 Platform, Enterprise Edition
- JMS
- JNDI
- JSP
- JTA
- JTS
- OMG
- ORB

Further Reading

- Architecting and Design J2EE Applications SL-425 Sun Microsystems 2000
- Enterprise JavaBeans Programming SL-351 SunMicrosystems 2000
- Rod Johnson Expert One-on-One J2EE Design and Development Wrox Press 2003

Resources

- http://java.sun.com/j2ee
- http://java.sun.com/jcp
- http://www.w3.org/