INTRODUCTION TO JAVA EE (J2EE)
Enterprise Computing

**Challenges**
- Portability
- Diverse Environments
- Time-to-market
- Core Competence
- Assembly
- Integration

**Key Technologies**
- J2SE™
- J2EE™
- JMS
- Servlet
- JSP
- Connector
- XML
- Data Binding
- XSLT

**Products**
- App Servers
- Web Servers
- Components
- Databases
- Object to DB tools

**Legacy Systems**
- Databases
- TP Monitors
- EIS Systems
What Is the J2EE?

• Open and standard based platform for
  • developing, deploying and managing
  • n-tier, Web-enabled, server-centric, and component-based enterprise applications
The Java™ Platform

Java Technology Enabled Devices
Java Technology Enabled Desktop
Workgroup Server
High-End Server

THE JAVA™ PLATFORM

Java 2 Platform Micro Edition (J2ME™)

Java 2 Enterprise Edition (J2EE)
Java 2 Standard Edition (J2SE)
Foundation Profile
Personal Profile
Personal Basis Profile
MIDP
CDC
CLDC

Optional Packages

JVM
KVM
CardVM
Java Card APIs

Java Card APIs
Open and Standard Solution

• Use "component and container" model in which container provides system services in a well-defined and as industry standard

• J2EE is that standard that also provides portability of code because it is based on Java technology and standard-based Java programming APIs
J2EE TECHNOLOGIES
J2EE 1.4 APIs and Technologies

- J2SE 1.4 (improved)
- JAX-RPC (new)
- Web Service for J2EE
- J2EE Management
- J2EE Deployment
- JMX 1.1
- JMS 1.1
- JTA 1.0
- Servlet 2.4
- JSP 2.0
- EJB 2.1
- JAXR
- Connector 1.5
- JACC
- JAXP 1.2
- JavaMail 1.3
- JAF 1.0
Java EE 5

- JAX-WS 2.0 & JSR 181
- Java Persistence
- EJB 3.0
- JAXB 2.0
- JavaServer Faces 1.2 – new to Platform
- JSP 2.1 – Unification w/ JSF 1.2
- StAX – Pull Parser – new to Platform
What is a Servlet?

- Java™ objects which extend the functionality of a HTTP server
- Dynamic contents generation
- Better alternative to CGI, NSAPI, ISAPI, etc.
  - Efficient
  - Platform and server independent
  - Session management
  - Java-based
What is JSP Technology?

• Enables separation of business logic from presentation
  • Presentation is in the form of HTML or XML/XSLT
  • Business logic is implemented as Java Beans or custom tags
  • Better maintainability, reusability

• Extensible via custom tags

• Builds on Servlet technology
EJB

Enterprise Java Beans
What is EJB Technology?

• A server-side component technology
• Easy development and deployment of Java technology-based application that are:
  • Transactional, distributed, multi-tier, portable, scalable, secure, …
Why EJB Technology?
Why EJB Technology?

- Leverages the benefits of component-model on the server side
Why EJB Technology?

- Leverages the benefits of component-model on the server side
- Separates business logic from system code
  - Container provides system services
Why EJB Technology?

- Leverages the benefits of component-model on the server side
- Separates business logic from system code
  - Container provides system services
- Provides framework for portable components
  - Over different J2EE-compliant servers
  - Over different operational environments
Why EJB Technology?

- Leverages the benefits of component-model on the server side
- Separates business logic from system code
  - Container provides system services
- Provides framework for portable components
  - Over different J2EE-compliant servers
  - Over different operational environments
- Enables deployment-time configuration
  - Deployment descriptor
Enterprise JavaBeans

Synchronous communication
- Session Bean
- Entity Bean

Asynchronous communication
- Message-Driven Bean

Stateless
- Stateful

Bean managed Persistence (BMP)
- Container managed Persistence (CMP)
N-tier J2EE Architecture

Client-Side Presentation
- Browser
  - Pure HTML
  - Java Applet
- Desktop
  - Java Application
- Other Device
  - J2EE Client

Server-Side Presentation
- Web Server
  - JSP
- J2EE Platform

Server-Side Business Logic
- EJB Container
  - EJB
- EJB

Enterprise Information System

Web Tier

EJB Tier
J2EE Containers & Components

Applet Container
- Applet
- J2SE

App Client Container
- App Client
- HTTP/HTTPS
- JNDI
- JMS
- RMI/IIOP
- JDBC
- J2SE

Web Container
- HTTP/HTTPS
- JNDI
- JMS
- JTA
- JavaMail
- JAF
- RMI/IIOP
- JDBC
- J2SE
- Servlet

EJB Container
- HTTP/HTTPS
- JNDI
- JMS
- JTA
- JavaMail
- JAF
- RMI/IIOP
- JDBC
- J2SE
- EJB

Database

RMI

J2EE Containers & Components
<table>
<thead>
<tr>
<th>Containers Handle</th>
<th>Components Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concurrency</td>
<td>Presentation</td>
</tr>
<tr>
<td>Security</td>
<td>Business Logic</td>
</tr>
<tr>
<td>Availability</td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
</tr>
<tr>
<td>Persistence</td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
</tr>
<tr>
<td>Life-cycle</td>
<td></td>
</tr>
<tr>
<td>management</td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
</tr>
</tbody>
</table>
Containers & Components
Containers & Components

• Containers do their work invisibly
  – No complicated APIs
  – They control by interposition
Containers & Components

• Containers do their work invisibly
  – No complicated APIs
  – They control by interposition

• Containers implement J2EE
  – Look the same to components
  – Vendors making the containers have great freedom to innovate
Typical J2EE applications
J2EE Application Architectures

- 4-tier J2EE applications
  - HTML client, JSP/Servlets, EJB, JDBC/Connector

- 3-tier J2EE applications
  - HTML client, JSP/Servlets, JDBC

- 3-tier J2EE applications
  - EJB standalone applications, EJB, JDBC/Connector

- B2B Enterprise applications
  - J2EE platform to J2EE platform through the exchange of JMS or XML-based messages
Which One to Use?

• Depends on several factors
  - Requirements of applications
  - Availability of EJB tier
  - Availability of developer resource