

1 Enterprise Java

- The web or app server is called the **program**.
- Each instance of the program serving a customer is called a **container**.
- **Components** (JSPs/EJBs) are loaded into these containers.

1.1 Architecture

An Enterprise Java system is comprised of:

1. Web browser, or Java application.
2. Web server (html files, JSPs and/or servlets).
3. J2EE application server (EJBs).
4. Database, or enterprise information systems.

1.2 Interception

Components are ‘plumbed’ into their containers with additional code. This ‘Interception code’ is run when accessing component methods.

1.2.1 Declarative Transactions

If set up, Interception code containing a database transaction BEGIN is called when accessing a component. If everything goes well a COMMIT is called, else a ROLLBACK. No JTC (Java Transaction Code) knowledge required.

To add this behavior, a method is marked as TRANSACTIONAL in its XML deployment descriptor.

1.2.2 Declarative Security

In the same way, security permissions can be marked in a Bean’s deployment descriptor. The interceptor plumbing code will perform authorisation checks when methods are called.

1.2.3 Programmatic Security

Alternatively the developer can hard code the security using methods that get the current user’s role. This requires more than just an XML change but allows more granular security checks than methods or pages.

1.2.4 What Are Enterprise Java Beans?

Storage wrappers with Get and Set accessors.

1.2.5 How are they created?

EJBs are always created by a factory class that extends EJBHome. The factory must have a Create() method. You can then use a JNDI (Java Naming and Directory Interface) lookup to get a factory.

1.2.6 Types of Bean

Entity Beans: Persistent Data ie. Customer Details. These beans map attributes directly onto their database table, hiding the database query language. Each bean has a *serializable* primary key that matches the primary key in the database. You can find the bean using the FindByPrimaryKey() method. If the database mapping is too complex to define with the XML deployment descriptors, you can manage persistence by overriding the ejbLoad and ejbStore methods in the bean's Factory.

Stateful Session Beans: Session Data ie. Shopping Basket.

Stateless Session Beans: Request Information ie. Form Return Values.

Message Driven Beans: Message Queues ie. Business to Business order. These beans allow for *Asynchronous* communication. These beans implement an OnMessage() method.

1.3 Deployment

To deploy beans, package them and their XML descriptors in an EAR file (Enterprise Application Archive). To deploy JSPs package them in a WAR (Web Application Archive) file. Both are compressed archives.

1.4 Example

IBM Websphere. High performance, based on Apache.

1.5 Advantages

Role separation. Multi-Vendor. Transactional.

1.6 Disadvantages

Lots to learn. Not completely portable. .NET seems faster.

2 JSPs

Reference: <http://java.sun.com/products/jsp/pdf/card12.pdf>

- On first use, they are compiled into a servlet. A servlet is a normal java class that implements doPost() and doGet(). This is then run to produce normal html output.

- JSPs are HTML first and designers can just ignore the inline JSP
- Servlets are code first and are good for more complex logic.

2.1 JSP Tags

- `<%= Java expression %>`
- `<% Java code; // scriptlet %>`
- `<%@ page import="java.sql.*" %>`
- `<%@ include file="header.jsp"%>`
- `<jsp:forward page="other.html"/>`
- `<%! Java declaration %>`

You can also divert unhandled exceptions to an errorpage

```
<\%@ page errorPage="error.jsp"\%>
```

2.2 Provided Objects

The server will provide out, request, session and response objects. You can fill session with information that you want to be carried over to the next page.

3 JDBC

Wrapper language for database specific query languages. Commands are built as strings and passed to the database. (Slow and insecure). Improved by caching and preparing statements. JDBC can do transactions manually, helping make changes isolated. This can be achieved with beans too.