E05

IMS E-Business 101

James D. Polo

IMS

technical conference

Las Vegas, NV **September 15 – September 18, 2003**

RETURN TO INDEX

Agenda

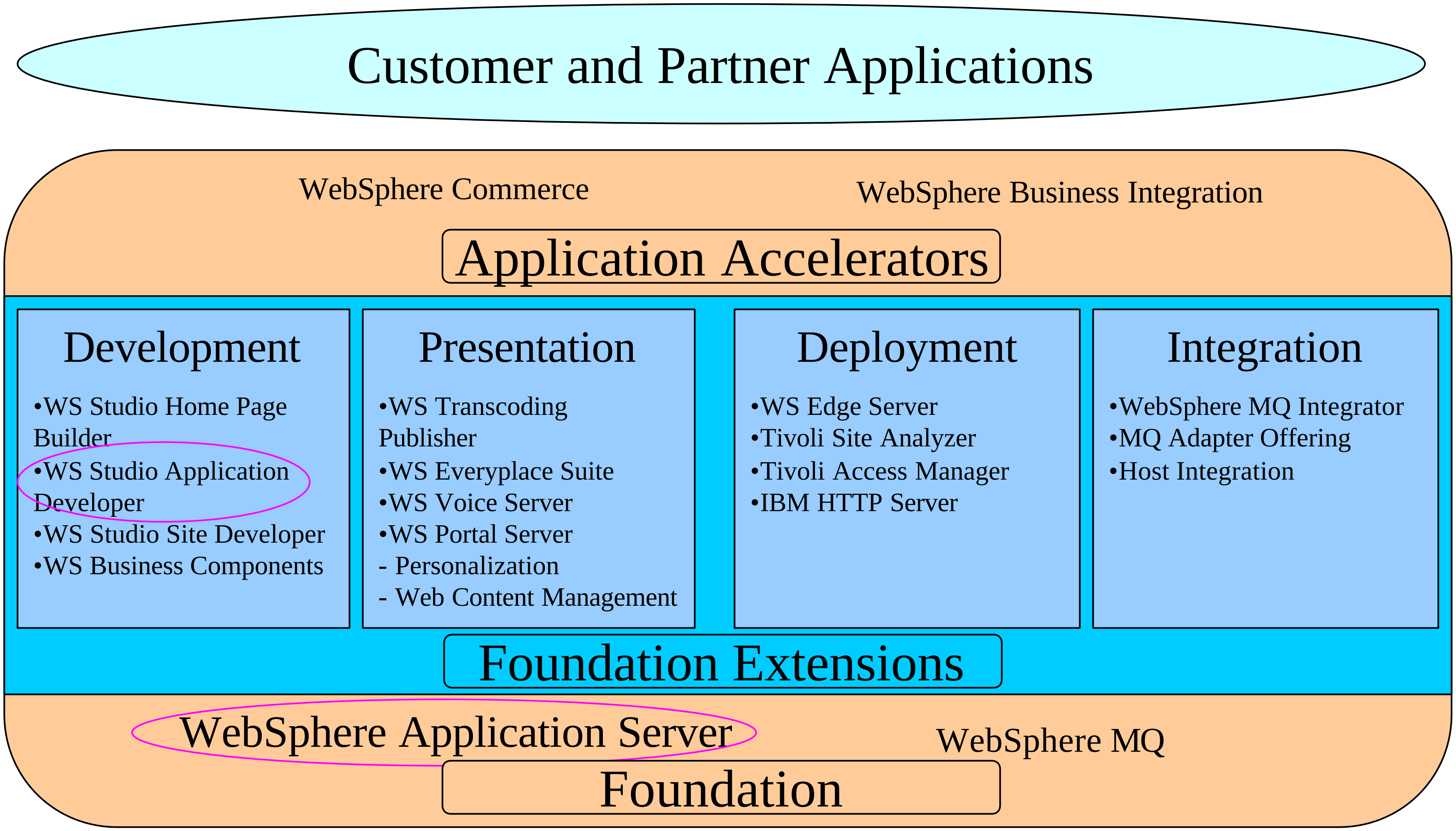
- What is E-Business?
  - WebSphere Solution
  - Runtime Architecture for a Standalone Server with IMS
- Enterprise Application
  - Enterprise Archive
  - Web Module
  - EJB Module
- J2EE Connector Architecture

E-Business

In its simplest sense, e-business is the use of Internet technologies to improve and transform key business processes.

- E-Commerce
- Business Intelligence
- Customer Relationship Management
- Supply Chain Management

WebSphere Solution



Customer and Partner Applications

WebSphere Commerce

WebSphere Business Integration

Application Accelerators

Development

- WS Studio Home Page Builder
- WS Studio Application Developer
- WS Studio Site Developer
- WS Business Components

Presentation

- WS Transcoding Publisher
- WS Everyplace Suite
- WS Voice Server
- WS Portal Server
  - Personalization
  - Web Content Management

Deployment

- WS Edge Server
- Tivoli Site Analyzer
- Tivoli Access Manager
- IBM HTTP Server

Integration

- WebSphere MQ Integrator
- MQ Adapter Offering
- Host Integration

Foundation Extensions

WebSphere Application Server

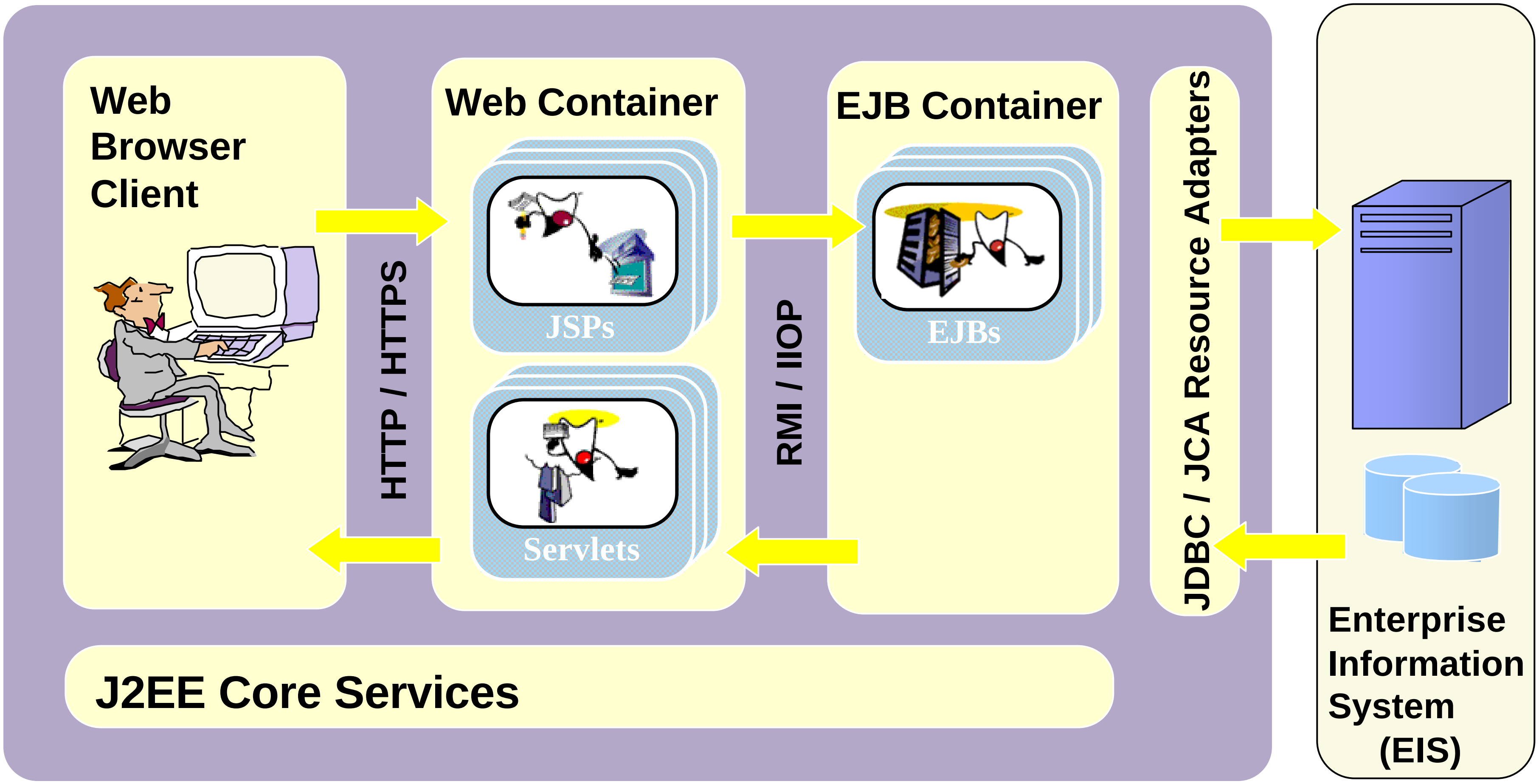
WebSphere MQ

Foundation

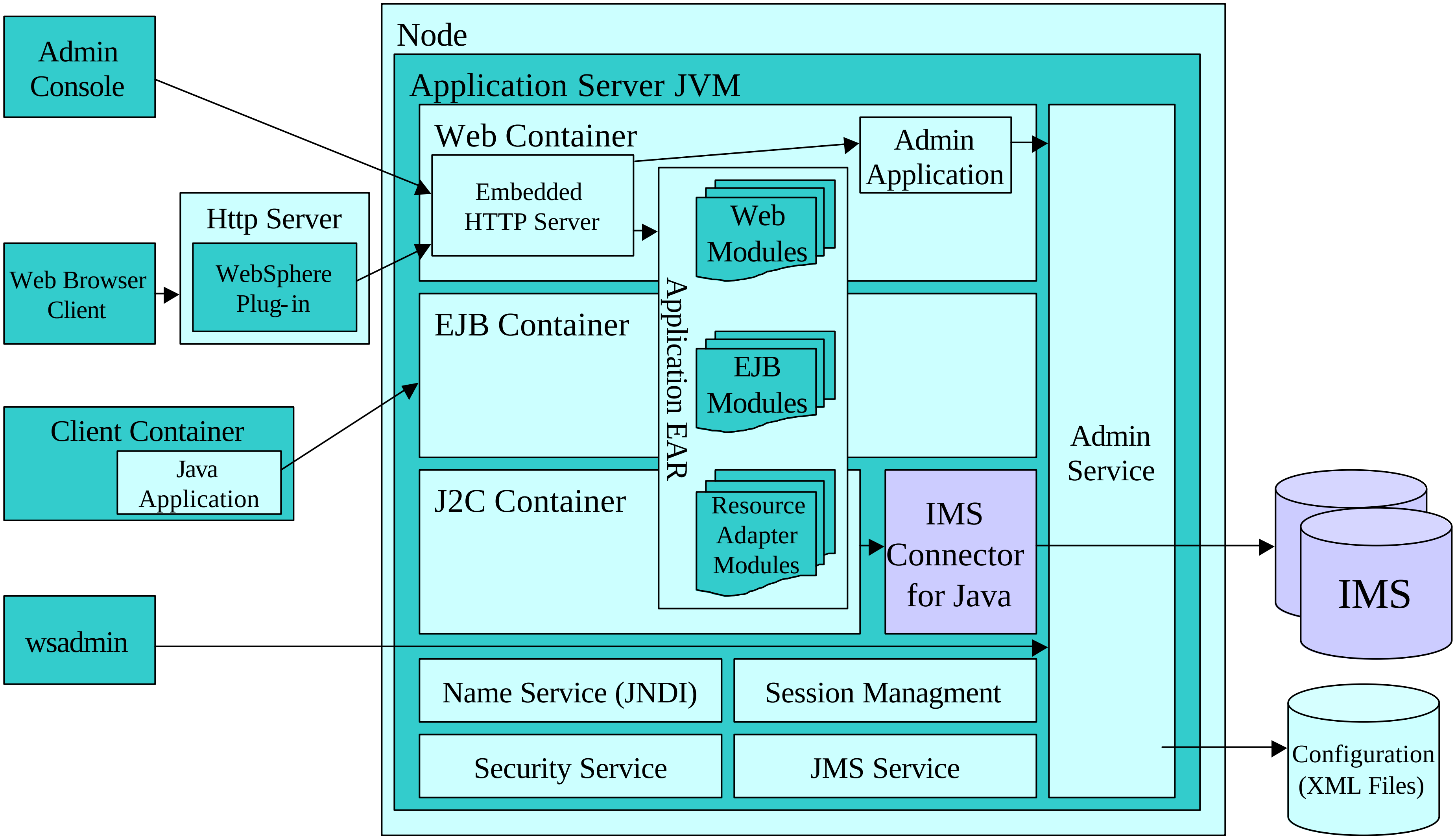
WebSphere 5.0 Packaging Scenarios

- WebSphere Application Server, Version 5
- WebSphere Application Server Network Deployment, Version 5
- WebSphere Application Server Enterprise, Verision 5
  - Focuses on J2EE Programming Model and Web Services
  - Includes the Network Deployment packaging

J2EE Server Model



Runtime Architecture for a Standalone Server with IMS



Agenda

- What is E-Business?
  - – WebSphere Solution
  - – Runtime Architecture for a Standalone Server with IMS

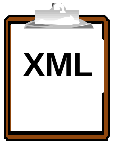
➤Enterprise Application

- Enterprise Archive
- Web Module
- EJB Module

- J2EE Connector Architecture

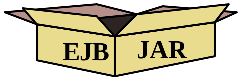
Enterprise Archive (EAR)

- Server deployable package
- Deployment Descriptor (XML file, application.xml)

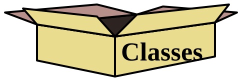


- Web Module (stored in Web Archive {WAR})
  - Servlets, JavaServer Pages (JSP) files
  - Utility classes
  - Static documents, such as HTML files, images, and sound.
  - Client-side applets, beans, and classes
  - Descriptive meta-information

- Enterprise Java Bean (EJB) Module (stored in a Java Archive {JAR})
  - One or more deployable enterprise beans.
  - Deployment Descriptor (XML file, ejb-jar.xml)



- Additional JAR files required by client or application



Extensible Markup Language (XML)

- Label and group information
- Consists of user defined tags
- International Organization of Standardizations
- Used to store WebSphere repository information and configuration information for application packages (EARs, WARs, JARs, etc.)

application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN"
   "http://java.sun.com/dtd/application_1_3.dtd">
<application id="Application_ID">
   <display-name>HelloWorld</display-name>
   <module id="JavaClientModule_1060255560122">
         <java>HelloWorldClient.jar</java>
   </module>
   <module id="EjbModule_1060255560132">
         <ejb>HelloWorldEJB.jar</ejb>
   </module>
   <module id="WebModule_1060255560142">
         <web>
                 <web-uri>HelloWorldWeb.war</web-uri>
                 <context-root>HelloWorldWeb</context-root>
         </web>
   </module>
</application>
```

HyperText Markup Language (HTML)

- Most Common Language for web page development
- Consists of a series of tags

```
<HTML>
<TITLE> My HTML </TITLE>
<B>WEB PAGE TEXT GOES HERE</B>
</HTML>
```

Servlets and Java Server Pages (JSPs)

- Servlets are Java classes that allow application logic to be embedded in HTTP request-response process
- JSPs are HTML documents with embedded JSP specific tags and inline Java code
  - On the server, JSPs are parsed and compiled into a Java Servlet

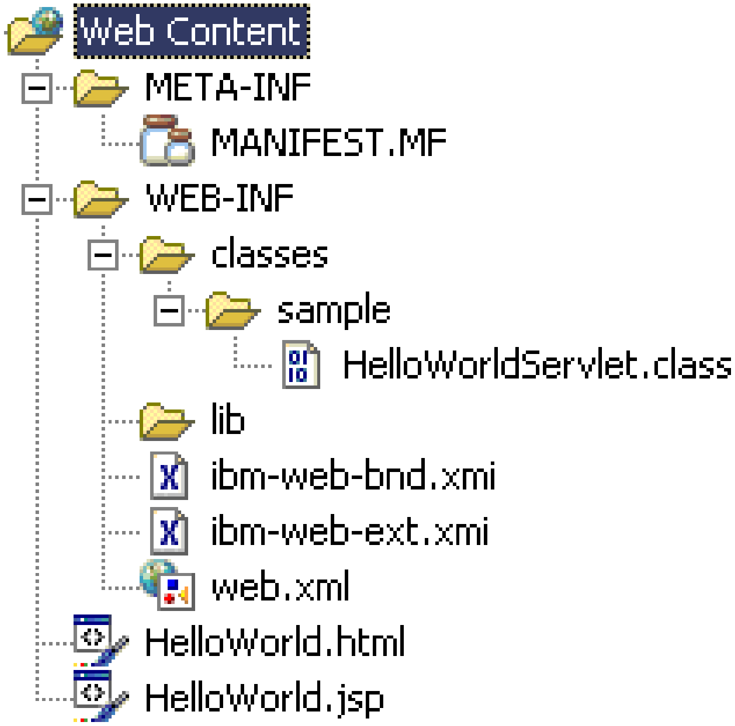
```
<HTML>
<TITLE> My HTML </TITLE>
<% String outputMessage = new String (“HELLO WORLD!!”); %>
<B> ><%=outputMessage%></B>
</HTML>
```

Web Archive

- IBM bindings and additional options beyond J2EE protocol are stored in the XML Metadata Interchange (XMI) format files

- Deployment Descriptor (web.xml)

- MANIFEST is a special file store in Java Archives (JARs) with information about the files in the jar file



```
Web Content
  META-INF
    MANIFEST.MF
  WEB-INF
    classes
      sample
        HelloWorldServlet.class
    lib
    ibm-web-bnd.xmi
    ibm-web-ext.xmi
    web.xml
  HelloWorld.html
  HelloWorld.jsp
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app id="WebApp">
    <display-name>HelloWorldWeb</display-name>
    <servlet>
        <servlet-name>HelloWorldServlet</servlet-name>
        <display-name>HelloWorldServlet</display-name>
        <servlet-class>sample.HelloWorldServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloWorldServlet</servlet-name>
        <url-pattern>/HelloWorldServlet</url-pattern>
    </servlet-mapping>
    <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Java Bean

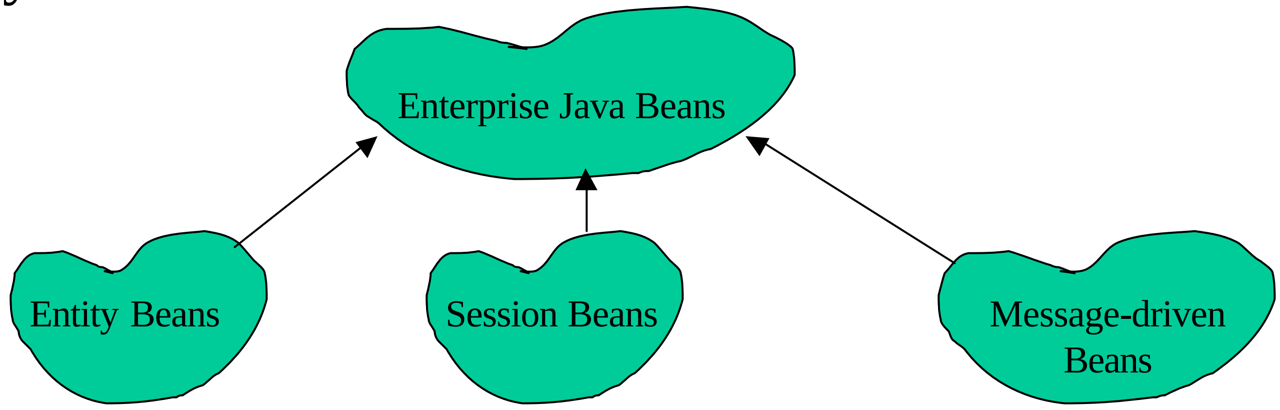
Java Bean – simply a Java component that adheres to a series of rules that allows users of the object to obtain information pertaining to that object for customization and visualization.

For example, a variable within a java class that can be modified by anyone should have a get and set method associated with it for ease of use and tooling purposes in order for the class to be considered a java bean.

```
public int value;
public int getValue() {
        return value;
}
public void setValue(int new_value) {
        value = new_value;
}
```

Enterprise Java Bean (EJB)

Enterprise Java Bean (EJB) – A java bean that is built for scaling and transaction support. It is packaged with an xml file (ejb-jar.xml) that contains information pertaining to the deployment of the bean.

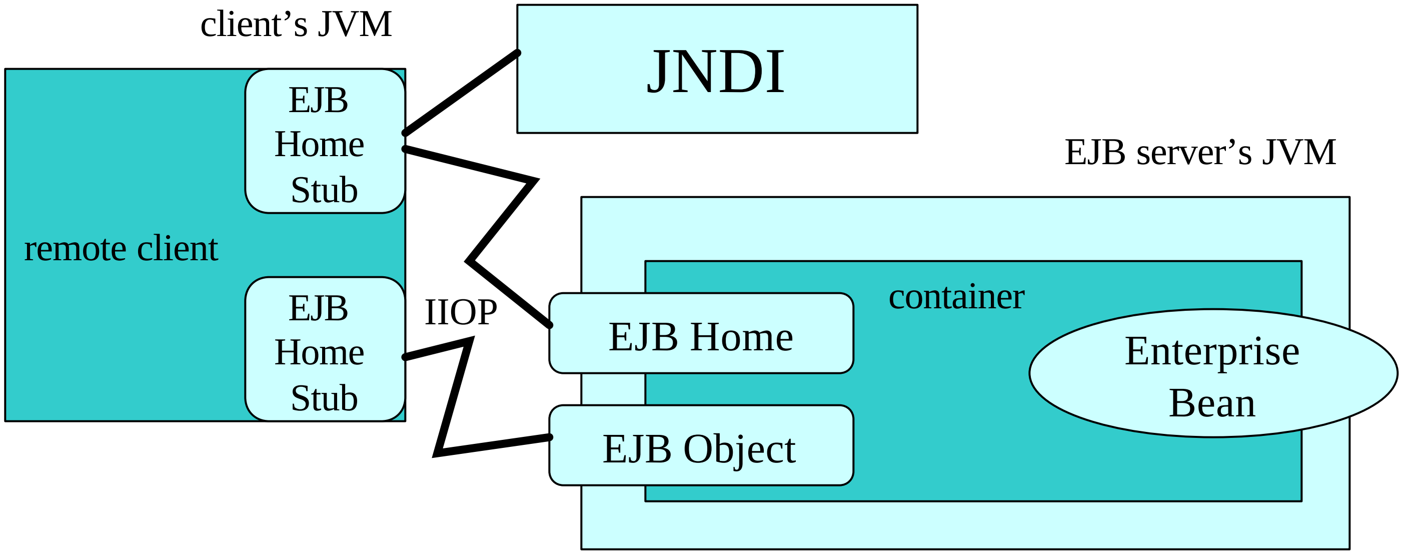


Container-Managed Persistance
Bean-Managed Persistence

Stateful
Stateless

An enterprise java bean, once deployed is handled by a container built into the application server (EJB Container).

Example of Session Bean



1. Client uses JNDI to find EJB Home
2. Client uses EJB Home create method to create a session EJB
3. Local EJB Home tell the container to create a remote EJB component
4. Container returns the EJB Object stub to the client
5. Client calls local methods on EJB Object, which are fowarded to remote EJB

EJB Module

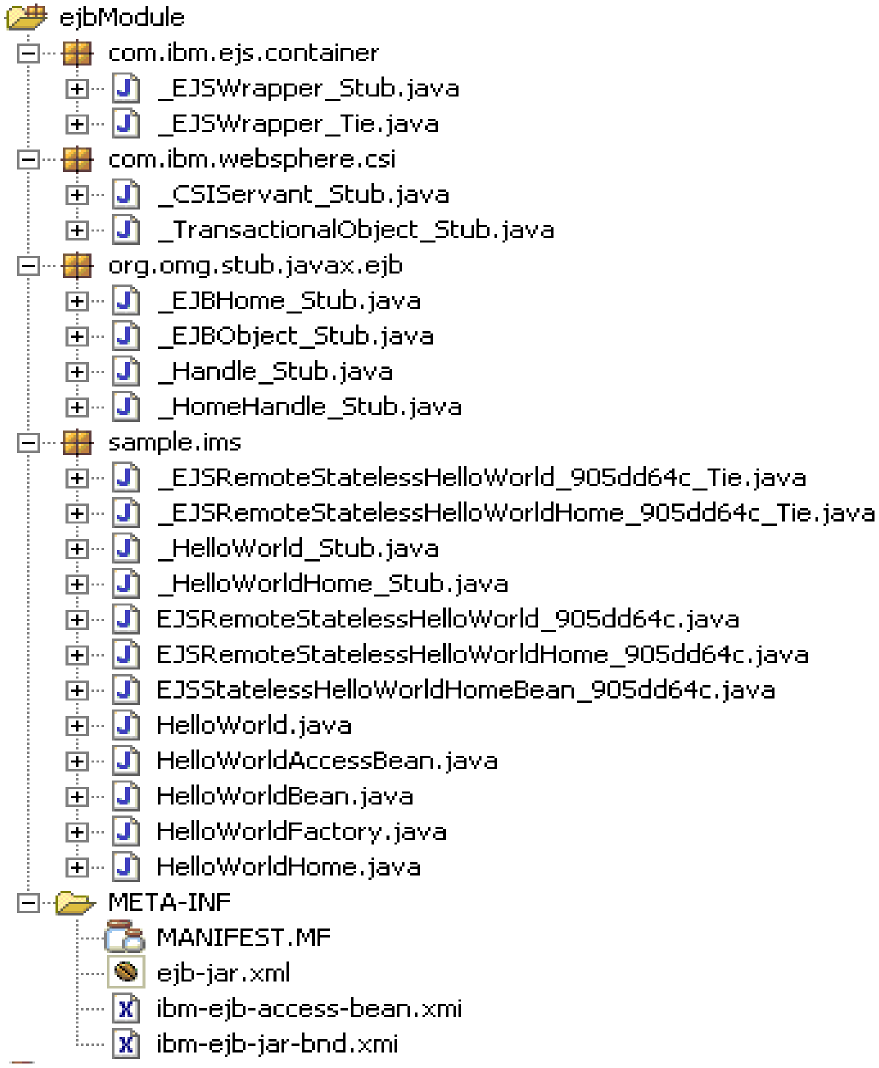
- Enterprise JavaBean
- Server (EJS) stub and tie
- files

- Remote Interface
- (HelloWorld.java)

- Home Interface
- (HelloWorldHome.java)

- Access Bean

- Deployment Descriptor (ejb.xml)



```
ejbModule
  com.ibm.ejs.container
    _EJSWrapper_Stub.java
    _EJSWrapper_Tie.java
  com.ibm.websphere.csi
    _CSIServant_Stub.java
    _TransactionalObject_Stub.java
  org.omg.stub.javax.ejb
    _EJBHome_Stub.java
    _EJBObject_Stub.java
    _Handle_Stub.java
    _HomeHandle_Stub.java
  sample.ims
    _EJSRemoteStatelessHelloWorld_905dd64c_Tie.java
    _EJSRemoteStatelessHelloWorldHome_905dd64c_Tie.java
    _HelloWorld_Stub.java
    _HelloWorldHome_Stub.java
    EJSRemoteStatelessHelloWorld_905dd64c.java
    EJSRemoteStatelessHelloWorldHome_905dd64c.java
    EJSStatelessHelloWorldHomeBean_905dd64c.java
    HelloWorld.java
    HelloWorldAccessBean.java
    HelloWorldBean.java
    HelloWorldFactory.java
    HelloWorldHome.java
  META-INF
    MANIFEST.MF
    ejb-jar.xml
    ibm-ejb-access-bean.xmi
    ibm-ejb-jar-bnd.xmi
```

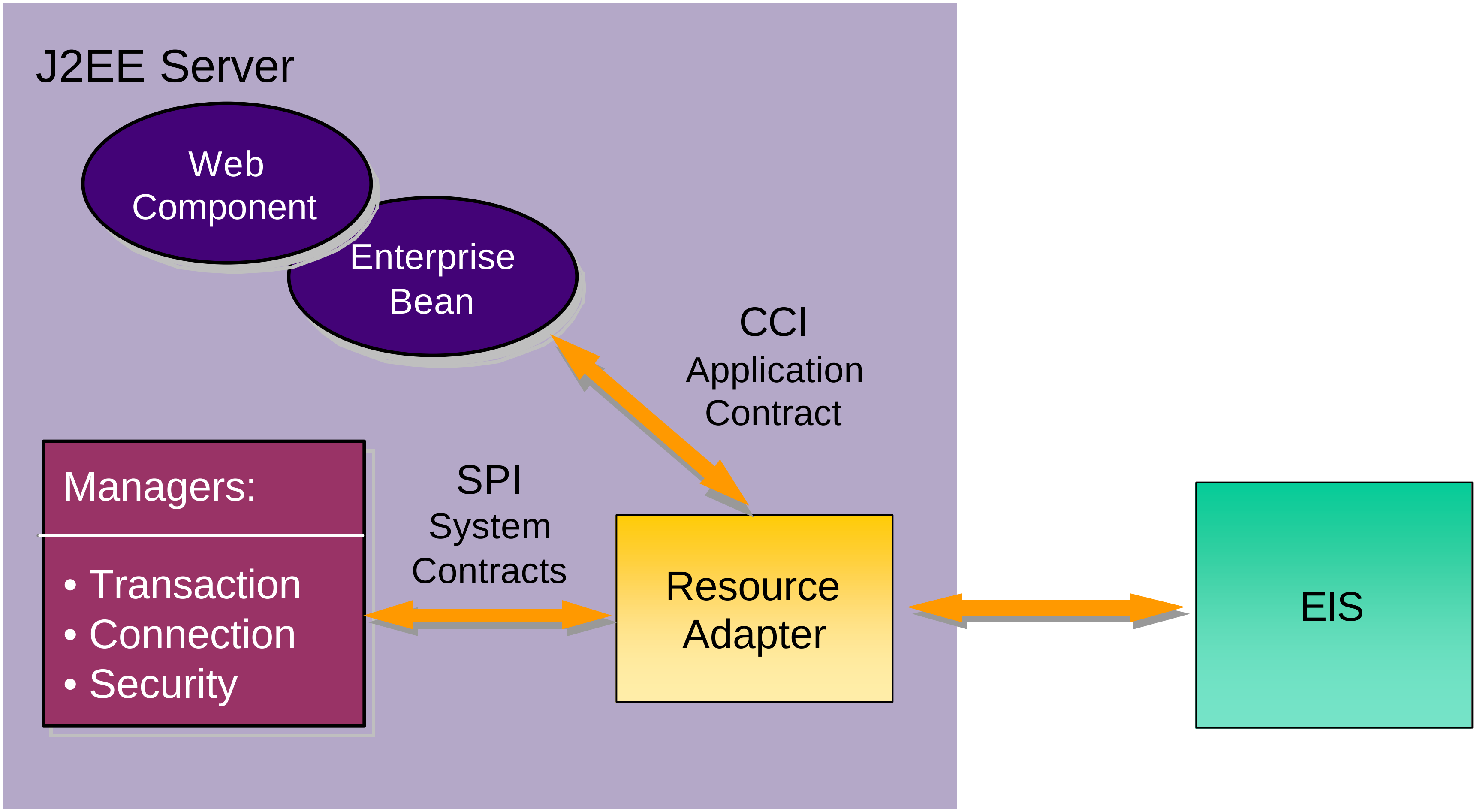
ejb.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans
   2.0//EN" "http://java.sun.com/dtd/ejb-jar_2_0.dtd">
<ejb-jar id="ejb-jar_ID">
   <display-name>HelloWorldEJB</display-name>
   <enterprise-beans>
        <session id="HelloWorld">
                <ejb-name>HelloWorld</ejb-name>
                <home>sample.ims.HelloWorldHome</home>
                <remote>sample.ims.HelloWorld</remote>
                <ejb-class>sample.ims.HelloWorldBean</ejb-class>
                <session-type>Stateless</session-type>
                <transaction-type>Container</transaction-type>
        </session>
   </enterprise-beans>
</ejb-jar>
```

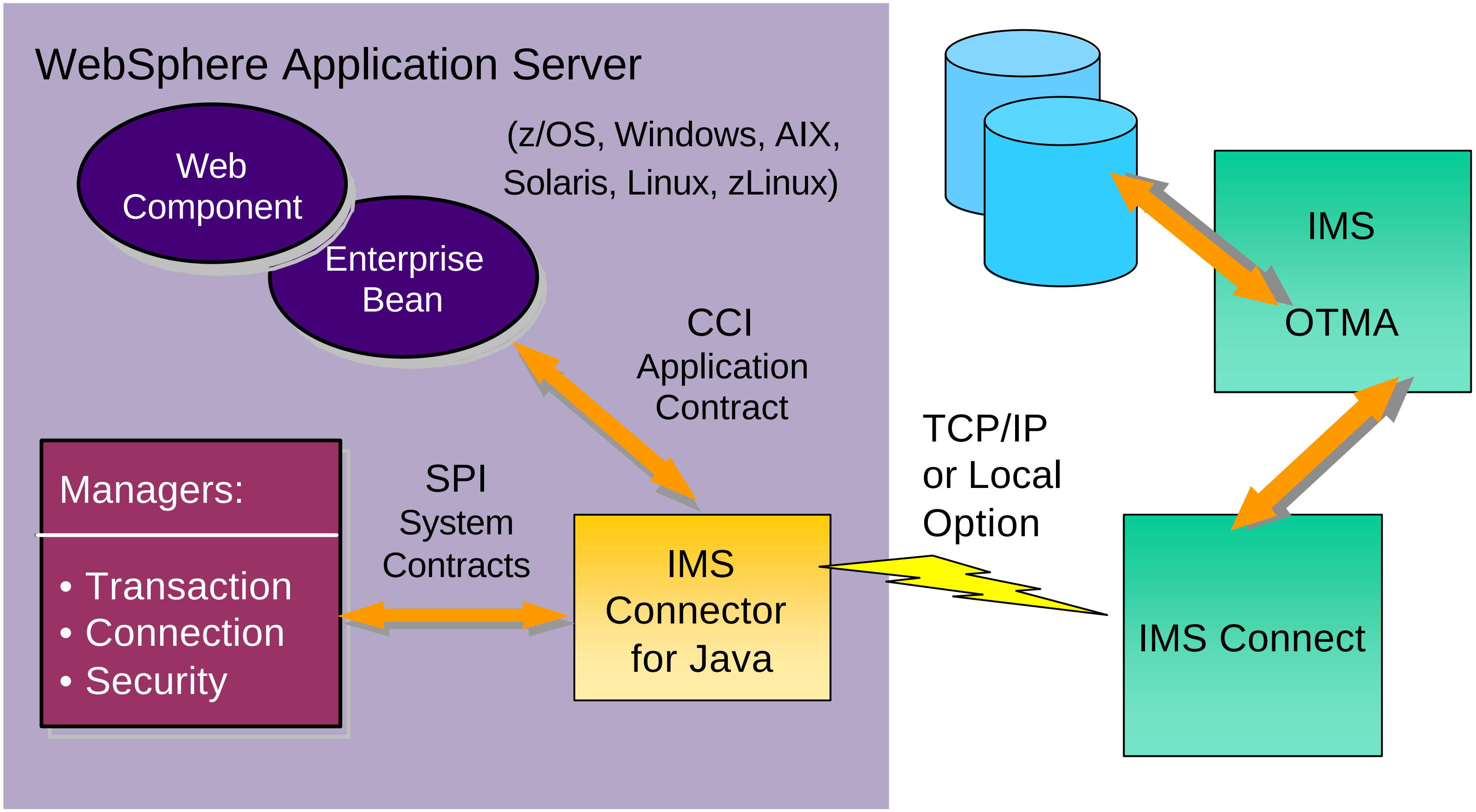
Agenda

- What is E-Business?
  - WebSphere Solution
  - Runtime Architecture for a Standalone Server with IMS
- Enterprise Application
  - Enterprise Archive
  - Web Module
  - EJB Module
- J2EE Connector Architecture

J2EE Connector Architecture



J2EE Connector Architecture



Getting More Information

- The Java Language Specification (second edition)
  - http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html
- JavaServer Pages Standard Tag Library
  - http://java.sun.com/products/jsp/jstl/index.html
- Enterprise JavaBeans Technology Downloads & Specifications
  - http://java.sun.com/products/ejb/docs.html
- IMS, IMS Connect, IMS Connector for Java
  - http://www.ibm.com/ims
- WebSphere Application Server
  - http://www.ibm.com/software/webservers/appserv/was/
- WebSphere Studio Application Developer Integration Edition
  - http://www.ibm.com/software/awdtools/studiointegration/

Glossary

API (Application Programming Interface) – Application programming interfaces allow users to program to a specific environment or model. They consist of contracts that provide guidelines for application programmers to adhere to. They reduce confusion and provide a common programming model. For example, the Java^{TM} API provides users descriptions of classes and examples on how to use the Java programming language to develop applications.

CCF (Common Connector Framework) – IBM's Common Connector Framework is a common client programming model for connectors. It is a common infrastructure programming model that gives a component environment a standard view of a connector and vice versa. This was adhered to prior to the introduction of the J2C architecture, but continues to be supported.

CCI (Common Client Interface) – The common client interface is a J2EE standard application interface used by clients to communicate to resource adapters. It is primarily bundled into code (services) generated by enterprise tools.

EAB (Enterprise Access Builder) - Enterprise Access Builder for Transactions is a set of frameworks and tools for creating Java applications that access existing host applications and data, for example using CICS, IMS, MQSeries, SAP or Encina. The Visual Age for Java tooling uses this framework to generate an Enterprise Access Bean Command which is composed of the connection specification, interaction specification, and input and output records to perform a single interaction with a host system. This was replaced by the services protocol and tooling supplied by WSADIE.

EAR (Enterprise Archive) – Enterprise Archive files are standard Java archive files that are bundled together and have the file extension .ear. An EAR file represents a J2EE application that can be deployed in a WebSphere^{TM} Application Server. An EAR file can consist of the following:

- One or more Web modules;
- One or more EJB modules;
- One or more application client modules;
- Additional JAR files required by the application;
- Any combination of these.
- EAR files also contain a deployment descriptor (an XML file) that describes the contents of the application and contains instructions for the entire application, such as security settings to be used in the run-time environment.

EIS (Enterprise Information System) – Enterprise Information Systems categorize any suite existing enterprise applications that are within a company (including Siebel, PeopleSoft, SAP, MQ Series, IMS, etc.).

EJB (Enterprise Java Bean) – An Enterprise Java Bean is a component architecture defined by Sun Microsystems for the development and deployment of object-oriented, distributed, enterprise-level applications. An EJB consists of a Java bean extended by a remote and home interface with a deployment descriptor to describe such things as naming, resource references, transactions, security management, etc.

Enterprise Services – Enterprise Services are the business functions of your enterprise, or of your business partners. At the heart of the Application Developer Integration Edition programming model are *business services*, or services for short. The *service* is the key element in the integration architecture, it binds everything together. When we say service we mean any kind of business service not just one that you can access via the Web. Services are used to model different kinds of service providers in a consistent way. For example, there are Web services, processes, EIS (J2EE Connector) services, and so on.

HTML (HyperText Markup Language) – The HyperText Markup language is the most commonly language used for generating web pages. It consists of a series of tags that aid in the visual organization of images and data; it provides the capability of links and forms; and allows the incorporation of media and various other files.

HTTP/HTTPS (HyperText Transfer Protocol/Secure) – In the Internet suite of protocols, the HyperText Transfer Protocol is the protocol that is used to transfer and display HTML and XML documents, exchange information, and securing transactions on the Web. When HTTP is used over SSL it is titled HTTPS to indicate a secure medium.

J2C/JCA (J2EE Connector Architecture) – The J2EE Connector Architecture Specification is a standard architecture for integrating Java applications with existing enterprise information systems (EIS).

J2C Connection Factory – A J2EE Connector Architecture Connection Factory is a list of properties that will configure a connection to a specific EIS. They also include some standard properties defined by the J2C architecture to manage connection pooling.

J2EE (Java 2 Platform, Enterprise Edition) – The Java 2 Platform, Enterprise Edition, defines a standard that applies to all aspects of designing, developing, and deploying multi-tier, server-based applications.

JAR (Java Archive) – The Java archive (JAR) file format that enables you to bundle multiple files into a single secure, compressed, versioned, portable unit. Usually JAR files will contain class files and other resources associated with an Java application.

Java Bean – A Java Bean is a customizable Java software component that adheres to a given standardization to provide introspection, event monitoring and handling, reusability, and store state information.

JNDI (Java Naming and Directory Interface) – The Java Naming and Directory Interface is an extension to the Java platform that provides Java applications with a standard interface to heterogeneous naming and directory services.

JMS (Java Message Service) – The Java Message Service is an API that allows separate application systems to reliably communicate asynchronously across many different message systems and products such as IBM MQSeries.

JSP (Java Server Pages) – JavaServer Pages are HTML documents with embedded JSP specific tags and inline Java code that enable dynamic web content, such as HTML, DHTML, XHTML, and XML files, to be included in a Web application. JSP files are one way that the WebSphere Studio implements server-side dynamic page content. JSP files allow an Web server, such as WebSphere Application Server or Apache Tomcat, to dynamically add content to your HTML pages before they are sent to a requesting browser. When you deploy a JSP file to a Web server that provides a servlet engine, it is preprocessed into a servlet that executes on the Web server.

Local Option – An IBM private protocol that can be used between IMS Connector for Java and IMS Connect when they both reside on the same system.

RAR (Resource Adapter Archive) – A Resource Adapter Archive file is a Java archive (JAR) file used to package a resource adapter for the J2EE Connector Architecture. A RAR file can contain the following:

- Enterprise information system (EIS) supplied resource adapter implementation code in the form of JAR files or other executables, such as dynamic link lists (DLL);
- Utility classes;
- Static documents, such as HTML files, images, and sound files;
- J2C common client interfaces, such as *cci.jar*.

RMI-IIOP (Remote Method Invocation – Inter-Orb Protocol) – The remote method invocation protocol is used to communicate method invocations over a network. Java Remote Method Invocation is a distributed object model in which the methods of remote objects written in the Java programming language can be invoked from other Java virtual machines, possibly on different hosts. When the RMI technology IS run over Internet Inter-Orb Protocol (RMI-IIOP) it uses the Common Object Request Broker Architecture (CORBA) to adhere to its standardizations and distributed computing capabilities to the Java 2 platform. Some of the features include the following: Enables RMI programmers to use IIOP as the transport mechanism; Enables interoperability between Java RMI programs and programs written in languages other than Java (e.g. C++); Compatible with Enterprise Java Beans; RMI-IIOP reference implementation runs on Windows (NT, 95, and 98) and Solaris. RMI-IIOP also runs on OS/390, AIX, and OS/2 from JDK 1.1.8 onwards; Provides a new-generation Java ORB that allows passing of objects by value; Standard extension to JDK 1.1.6 onwards, including Java 2; Available for free download.

QoS (Quality of Service) – Quality of Service determines the service usability and utility, both of which influence the popularity of the service. It's factors includes such qualities as availability, accessibility, integrity, performance, reliability, regulatory, and security.

Servlet – Servlets are Java programs that use the Java Servlet Application Programming Interface (API). You must package servlets in a Web Archive (WAR) file or Web module for deployment to the application server. *Servlets* run on a Java-enabled Web server and extend the capabilities of a Web server, similar to the way applets run on a browser and extend the capabilities of a browser. Servlets can support dynamic Web page content, provide database access, serve multiple clients at one time, and filter data.

SOAP (Simple Object Access Protocol) – The Simple Object Access Protocol is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP can be used to query and return information and invoke services across the Internet. A SOAP message is a transmission of information from a sender to a receiver. SOAP messages are combined to perform request/response patterns. SOAP is transport protocol independent. SOAP can be run over existing Internet infrastructure (for example SOAP over HTTP). SOAP enables the binding and usage of discovered Web services by defining a message path for routing messages. SOAP is used to query UDDI for Web services. It uses XML to describe of three parts to every message, the envelope, encoding rules, and communication styles.

SPI (Service Provider Interface) – Service provider interfaces allow vendors to integrate their software with users of a specific environment or model. They consist of contracts that provide guidelines for service providers to adhere to. They reduce confusion and provide a common interface. For example, an SPI for containers would indicate to providers how to perform such things as transaction, connection, and security management.

SSL (Secure Socket Layer) – The secure socket layer is a security protocol developed by Netscape to authenticate clients and servers and to encrypt Internet transactions.

TCP/IP (**Transmission Control Protocol/Internet Protocol**) **–** The Transmission Control Protocol/Internet Protocol is set of communication protocols that provide peer-to-peer connectivity functions for both local and wide area networks.

UDDI (**Universal Description, Discovery, and Integration) –** The Universal Description, Discovery, and Integration is a set of standards-based specifications for service description and discovery over the Internet. UDDI has two functions: (1) it is a SOAP-based protocol that defines how clients communicate with UDDI registries, and (2) it is a particular set of global replicated registries. UDDI includes an XML schema for SOAP messages that defines a set of documents to describe business and services information, a common set of APIs for querying and publishing information to the directories, and an API for replicating directory entries between peer UDDI nodes. For more information on UDDI refer to www.uddi.org

WAR (Web Archive) – A Web archive file is a packaged Web application. A Web application is a group of HTML pages, JSP pages, servlets, and other resources, along with source files, optionally, that can be managed as a single unit. WAR files can be used to import a Web application into a Web server. In addition to project resources, the WAR file includes a Web deployment descriptor file. The Web deployment descriptor is an XML file that contains deployment information, MIME types, session configuration details, etc., for a Web application. The Web deployment descriptor file (web.xml) provides information about the WAR file that must be shared with the developers, assemblers, and deployers in a J2EE environment.

WSAD-IE (WebSphere Studio Application Developer Integration Edition) – The WebSphere Studio Application Developer Integration Edition is a tool based off of the Eclipse Project. It is a suite of tools that aid in developing business integration applications and generating code based on specific architectures or models.

WSDL (Web Services Description Language) – The Web Services Description Language is used to define a service. In WSDL a service is defined in three distinct parts:

- The **PortType**. The PortType defines the abstract interface offered by the service. A PortType defines a set of *Operations*. Each operation can be In-Out (request-response), In-Only, Out-Only and Out-In (Solicit-Response). Each operation defines the input and/or output *Messages*. A message is defined as a set of *Parts* and each part has a schema-defined type.
- The **Binding**. A binding defines how to map between the abstract PortType and a real service format and protocol. For example the SOAP binding defines the encoding style, the SOAPAction header, the namespace of the body (the targetURI), and so on.
- The **Port**. This defines the actual location (endpoint) of the available service - for example, the HTTP URL on which a SOAP service is available.

WSIF (Web Services Invocation Framework) – WSIF provides a Java API for invoking Web services, independent of the format of the service or the transport protocol through which it is invoked. WSIF provides the following features: It has an API that provides binding-independent access to any Web service; It is closely based on WSDL, so it can invoke any service that can be described in WSDL; It allows stubless (completely dynamic) invocation of a Web service; You can plug a new or updated implementation of a binding into WSIF at runtime; You can defer the choice of a binding until runtime. WSIF is designed to work both in an unmanaged environment (standalone) and inside a managed container. You can use JNDI to find the WSIF service, or else read in the WSDL definition.

XML (Extensible Markup Language) – The Extensible Markup Language is a programming language that enables web authors and developers to create customized tags. It is used to label and group various chunks of information to be used by applications. It consists of a series of labels for general topics and detailed topics that are encompassed by the larger ones. It is derived from the standard markup language described by the international organization of standardizations.

RETURN TO INDEX