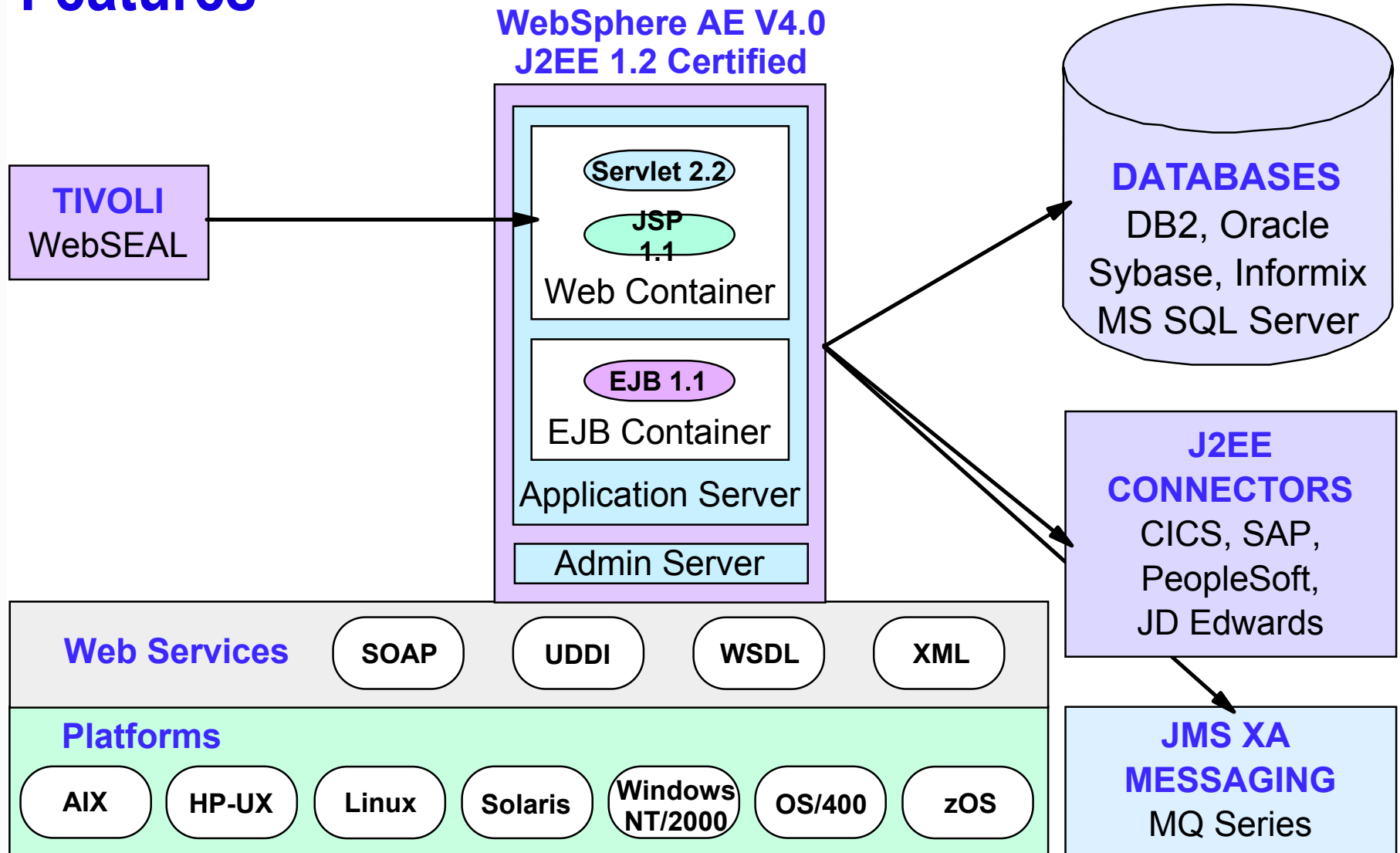




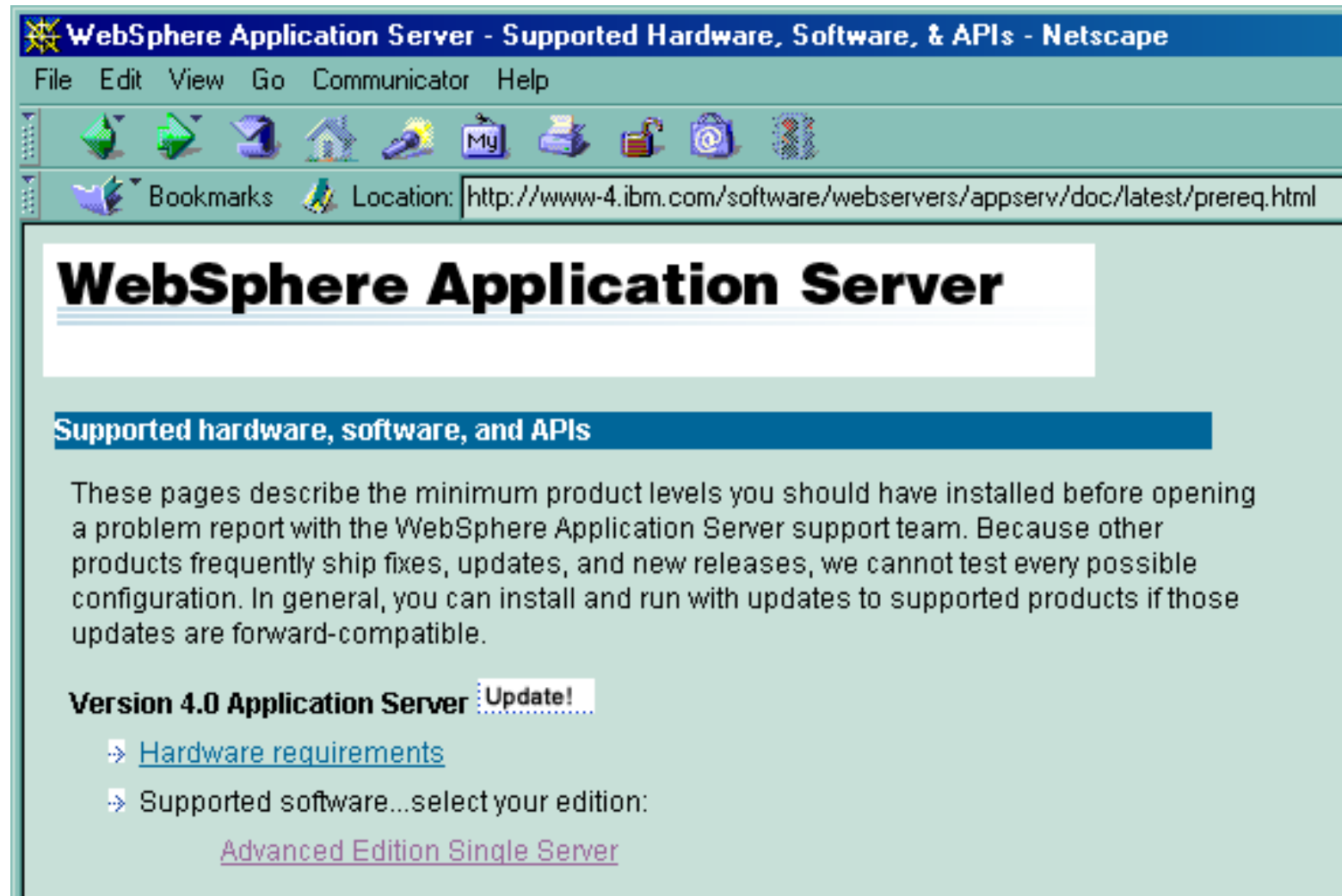
WebSphere

■ the fastest way to dynamic e-business ■

Features



Windows, Linux, Solaris, HP-UX, OS/400, AIX, z/OS



The screenshot shows a Netscape browser window with the title "WebSphere Application Server - Supported Hardware, Software, & APIs - Netscape". The address bar displays the URL "http://www-4.ibm.com/software/webservers/appserv/doc/latest/prereq.html". The main content area features the heading "WebSphere Application Server" and a section titled "Supported hardware, software, and APIs". This section contains a paragraph explaining that the pages describe minimum product levels and that updates to supported products are forward-compatible. Below this, there is a link to "Version 4.0 Application Server" with an "Update!" button next to it. Further down, there are two links: "Hardware requirements" and "Supported software...select your edition:", with the latter leading to the "Advanced Edition Single Server" page.

WebSphere Application Server - Supported Hardware, Software, & APIs - Netscape

File Edit View Go Communicator Help

Bookmarks Location: <http://www-4.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

WebSphere Application Server

Supported hardware, software, and APIs

These pages describe the minimum product levels you should have installed before opening a problem report with the WebSphere Application Server support team. Because other products frequently ship fixes, updates, and new releases, we cannot test every possible configuration. In general, you can install and run with updates to supported products if those updates are forward-compatible.

Version 4.0 Application Server [Update!](#)

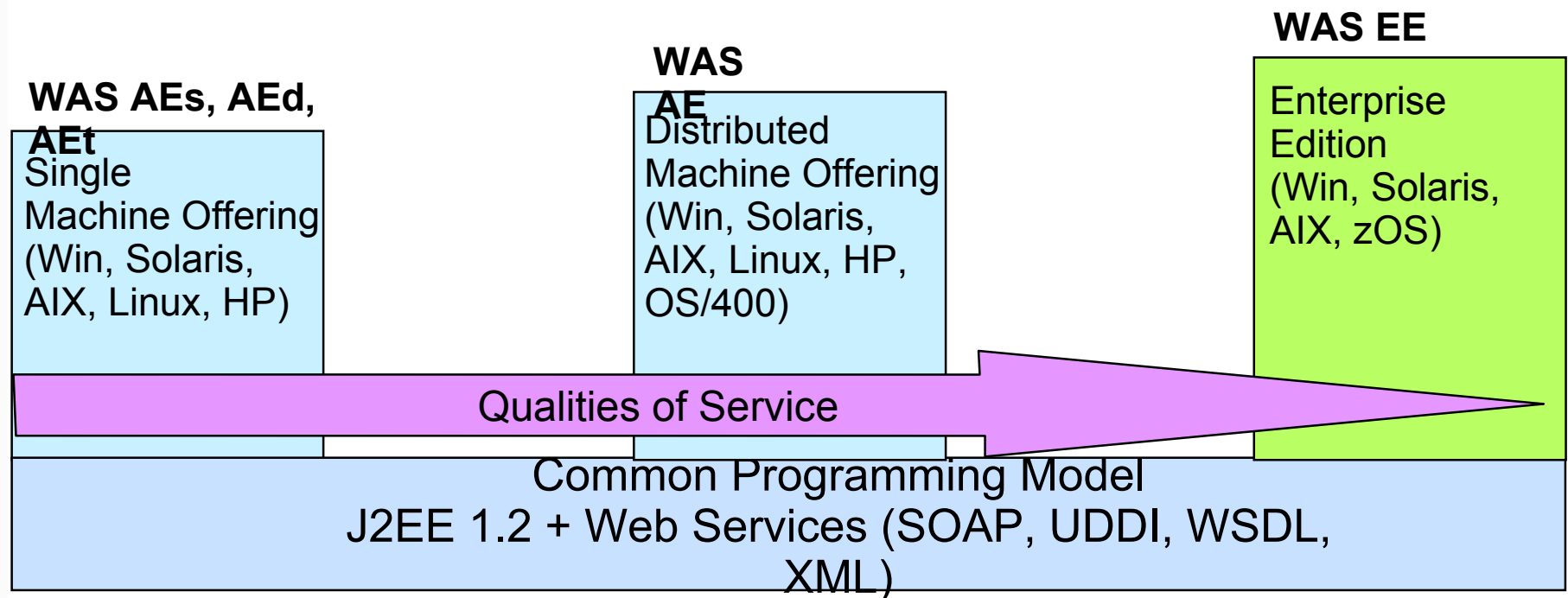
- [Hardware requirements](#)
- Supported software...select your edition:
 - [Advanced Edition Single Server](#)

<http://www-4.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

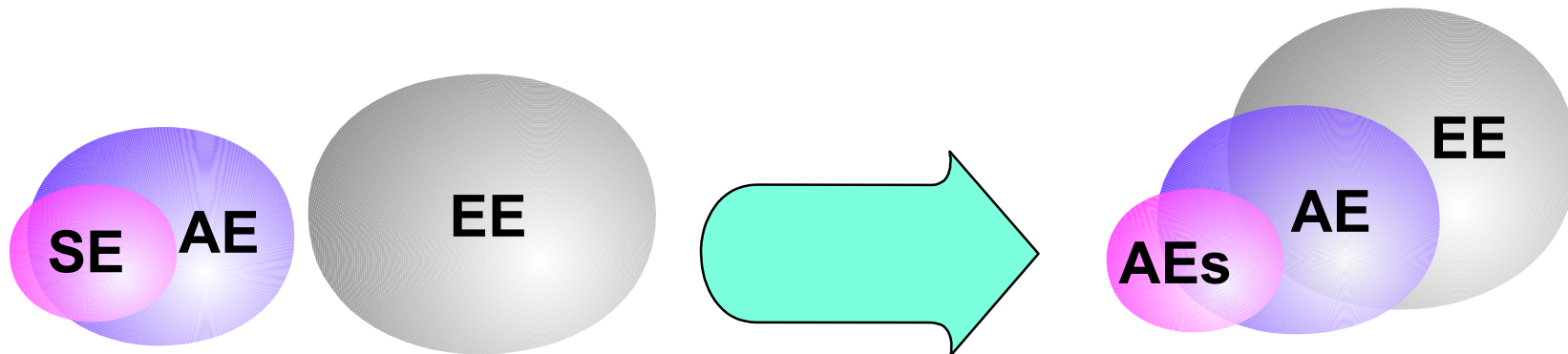
- **Advanced Edition Version 4.0, Single Server (AEs)**
 - ▶ Lightweight, simplified implementation
 - ▶ No relational database requirement for repository
 - ▶ Intended for development and unit testing
 - ▶ Departmental or smaller companies production and deployment
- **Advanced Edition Version 4.0, Development Only (AE_d)**
 - ▶ Identical to Single Server Option, but licensing restricts usage to non-production
 - ▶ Packaged with new developer tools
- **Advanced Edition Version 4.0, Trial Program (AE_t)**
 - ▶ Identical to Single Server Option
 - ▶ Free download for Windows and Linux platforms
- **Advanced Edition Version 4.0 (AE)**
 - ▶ Quality of Service enhancements
 - ▶ Multiple/Distributed Server support
 - ▶ Intended for production and deployment
- **Enterprise Edition**
 - ▶ High-end functions, beyond the J2EE specifications
 - ◆ Examples: internationalization, ActiveX to EJB bridge, ...
 - ▶ AE, TXSeries, MQSeries, Enterprise Services

V4 Editions and Platforms

- Identical programmer experience
- Identical end user experience
- Platform unique operational experience
- Differentiate on Qualities of Service



WebSphere Evolution

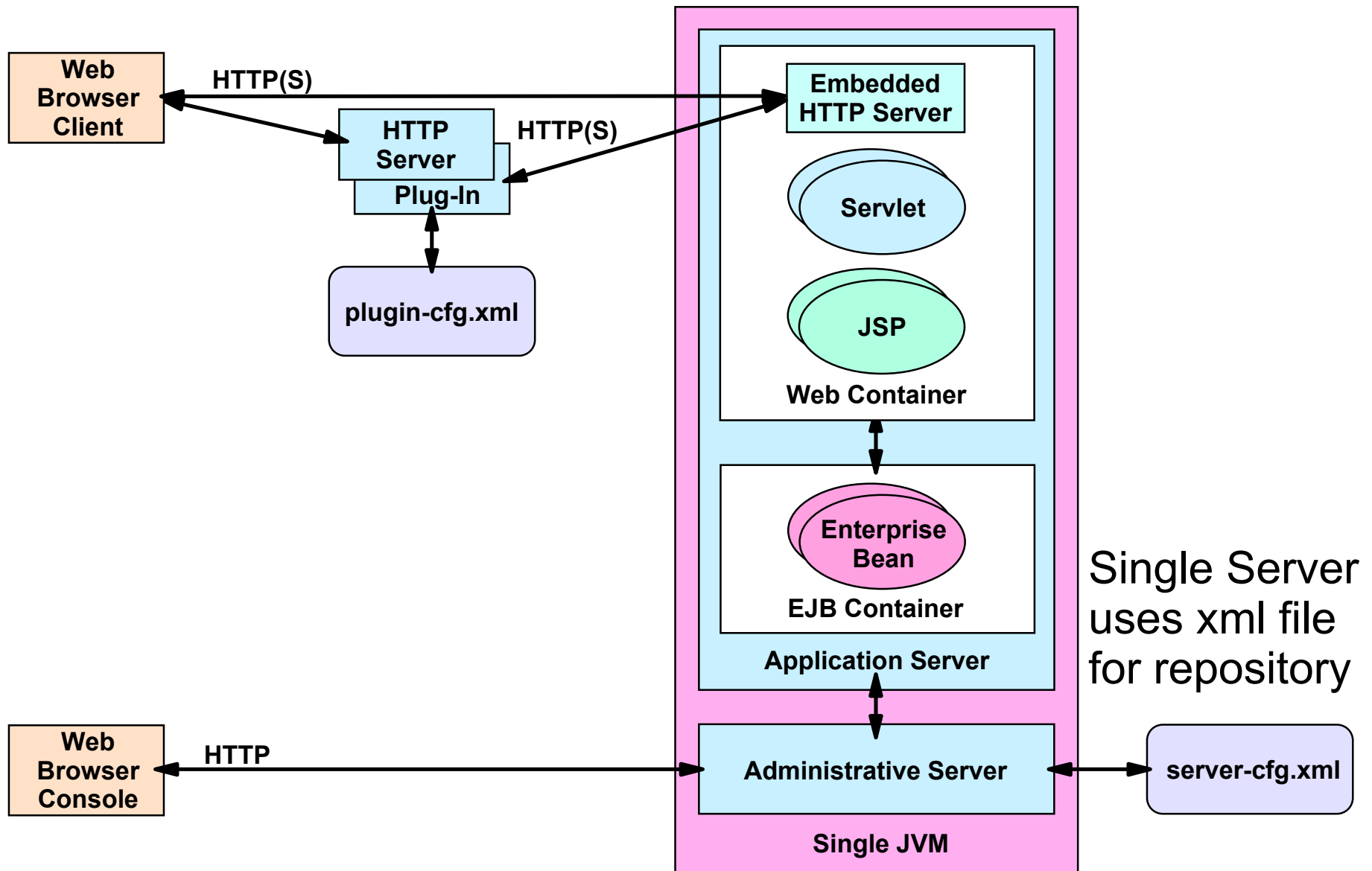


- Development View before V4:
 - Separate code bases
 - Not extendable
- Development View with V4:
 - Move to common code base
 - Enterprise Extensions (EEx) for added value and quality of service

AEs Version 4.0, Single Server Option

- Single machine, single process
- Full J2EE Programming Model
- Full Web Services Enablement
- XML-based Repository
- HTTP/HTTPS support for Plug-in transport
- Roles-based security - limited to Local OS
- Simplified administration via web browser
- DataSource support for DB2, Oracle
- AEd (Development Only) has matching functionality, but Development only licensing
- AEt (Trial Program) has matching functionality, but timebombed and can be downloaded at no charge

WebSphere AE Single Server Architecture



WebSphere Advanced Edition (AE)

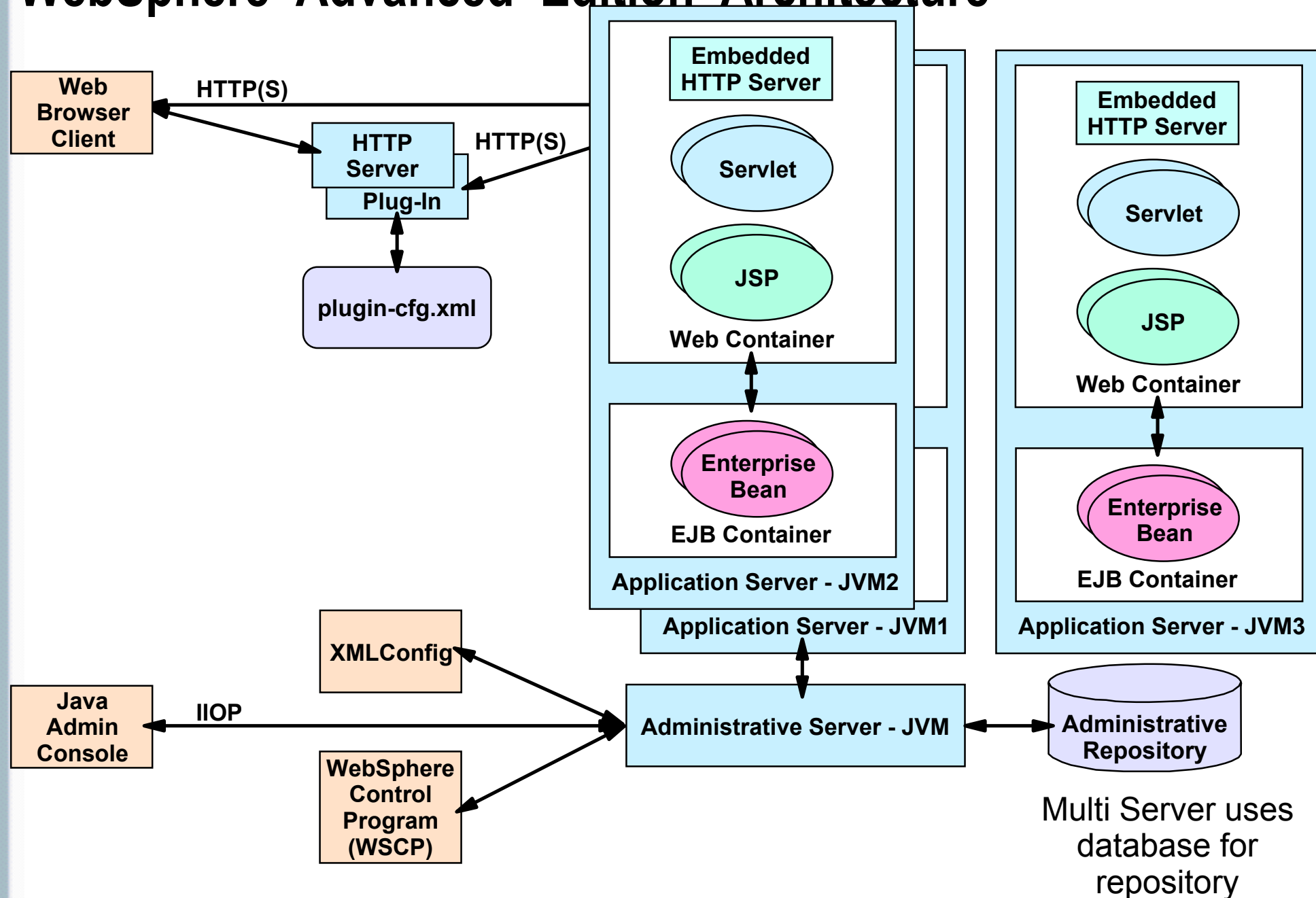
- Multiple application server processes
- Support for **multiple nodes** (machines)
- Full-blown workload management facilities
- J2EE 1.2 +
 - **Java 2 Connectors**
 - JMS/XA interface to MQ Series
- Expanded JDBC **database** support
 - Sybase
 - Informix
- Expanded options for **security** registry
- Migration tools

Workload Management

- Sharing requests across multiple servers
- Configuration options that improve
 - ▶ Scalability - serve more users
 - ▶ Load balancing - share work fairly
 - ▶ Availability - system runs if server fails

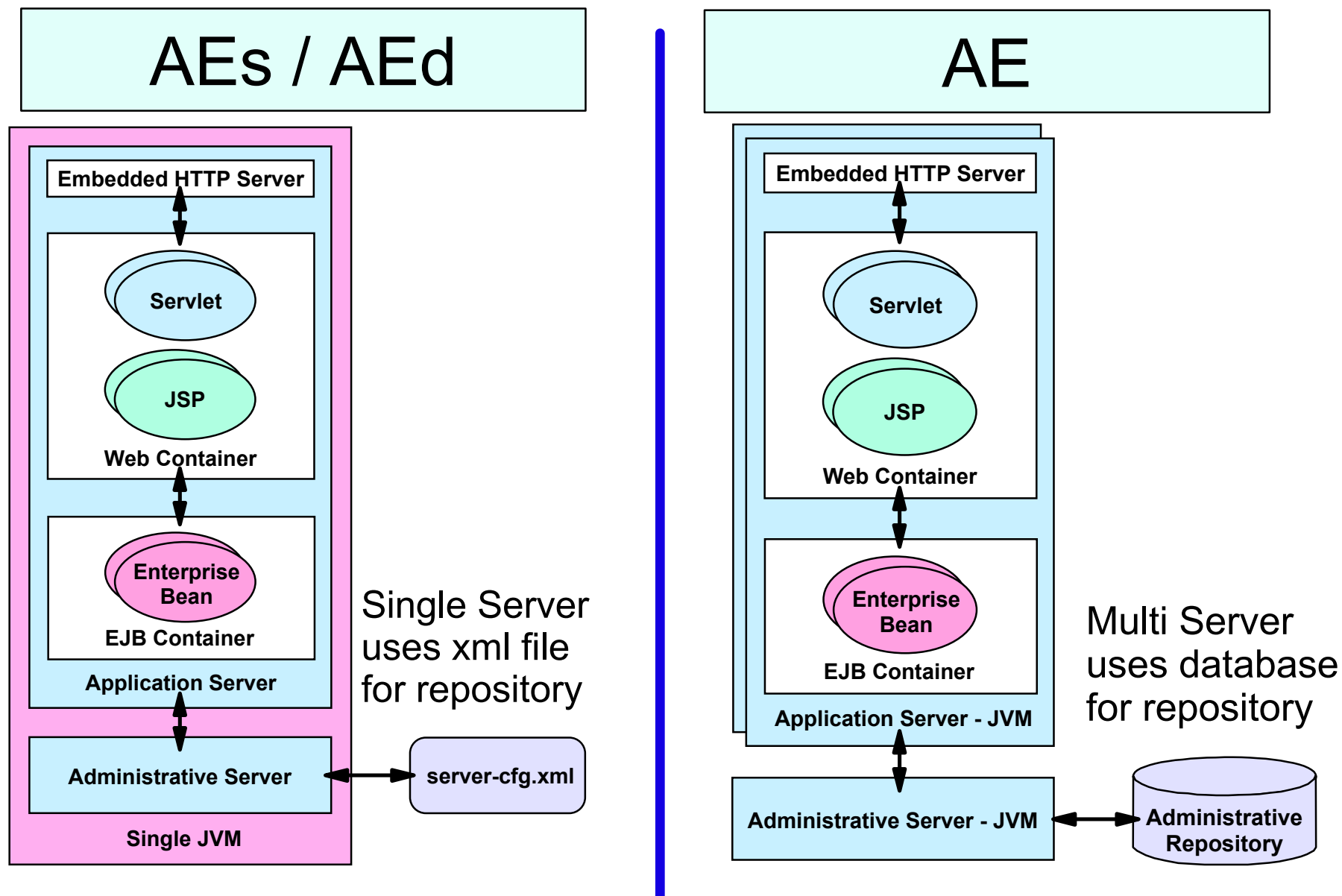


WebSphere Advanced Edition Architecture



AE R4.0 vs. AEs R4.0

Item	AE	AEs
J2EE Compliant	yes	yes
admin repository	relational DB	XML file
admin console	application (thick)	web-based (thin)
Dynamic caching	yes	no
Clustering (WLM)	yes	no
Security	pluggable	local OS
J2C	yes	no
JMS/XA	yes	no



IBM.

WebSphere

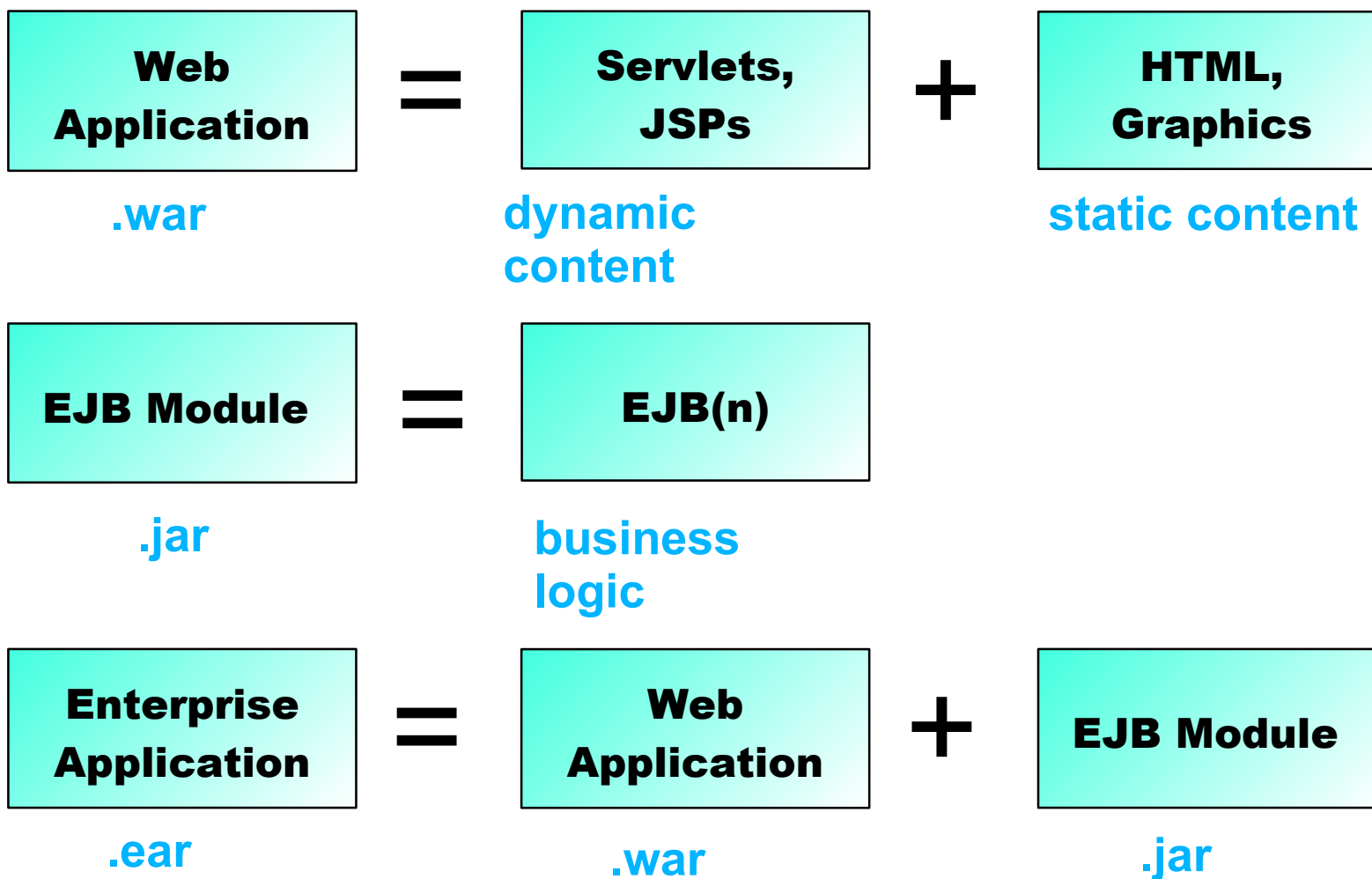
the fastest way to dynamic e-business



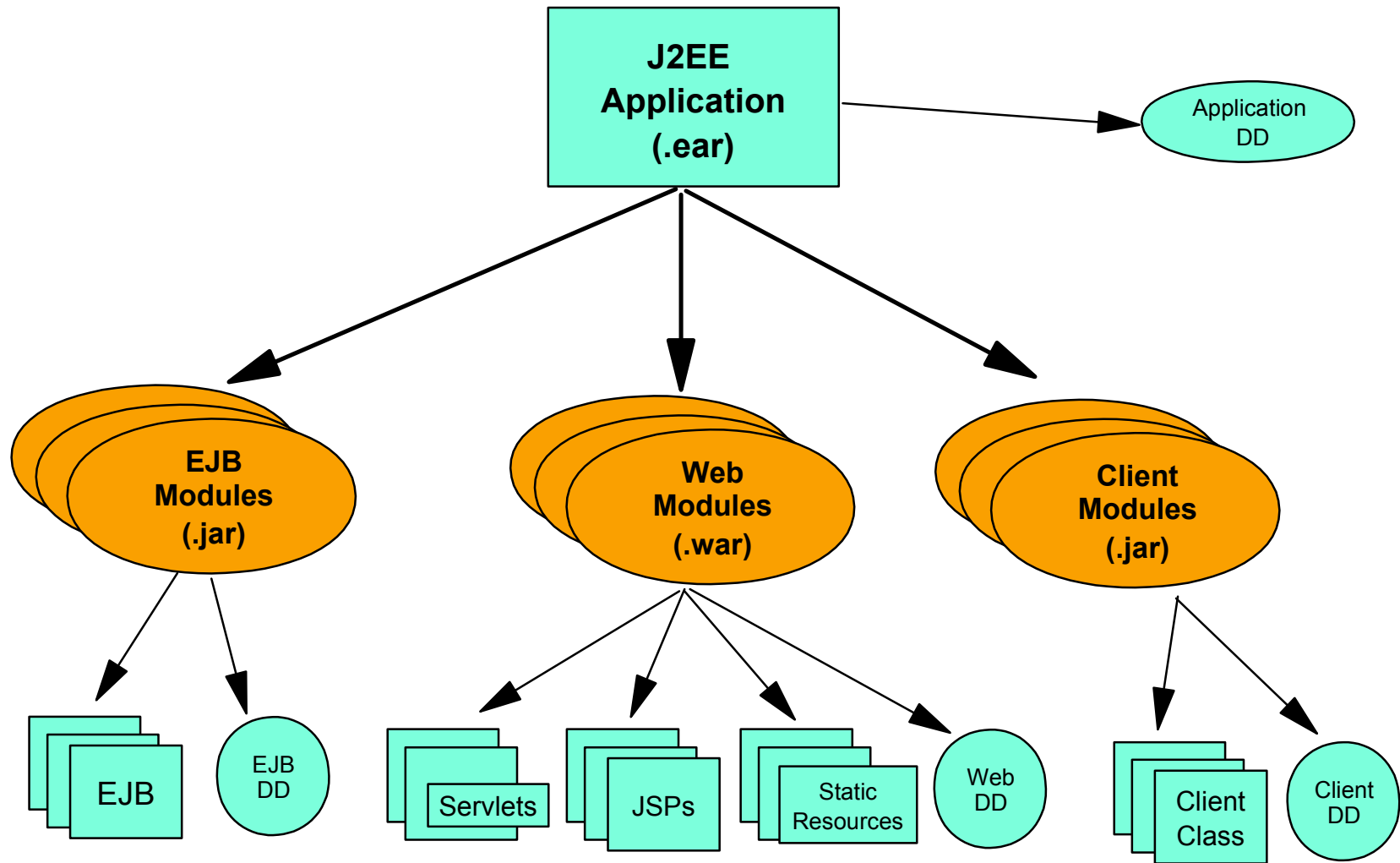
J2EE Separation

- J2EE creates a clear separation between development (creating the application) and administration (installing and managing the application). This separation enables the development of applications that are independent from the environments in which they are deployed. In addition, J2EE task separation simplifies the process of promoting an application from initial development up through production, or of moving an application from one server to another. In each of these cases, changes to application code are not necessary; only deployment parameters might change.
- Version 4.0 supports J2EE task separation through reorganized interfaces. In Version 3.x, developers used the administrative console to create, edit, and view applications. In Version 4.0, developers use the Application Assembly Tool (AAT) to create, edit, and view J2EE applications.
- In Version 4.0, each application is installed into the server domain and bound to an environment when the application is installed. This enables administration at the application and module level. Administrators no longer need to manage individual servlets, JSPs, or beans.
- The relationship between applications and application servers has changed in J2EE. After a J2EE application is created, you install it onto application servers through the administrative console. Through the administrative console, you can view installed modules either by the application to which they belong or by the application server on which they are installed. Modules can be started and stopped individually as well as collectively. Modules can be started collectively by either starting the application to which they belong or starting the application server on which they are installed. Modules can be stopped in a similar way.

J2EE Packaging



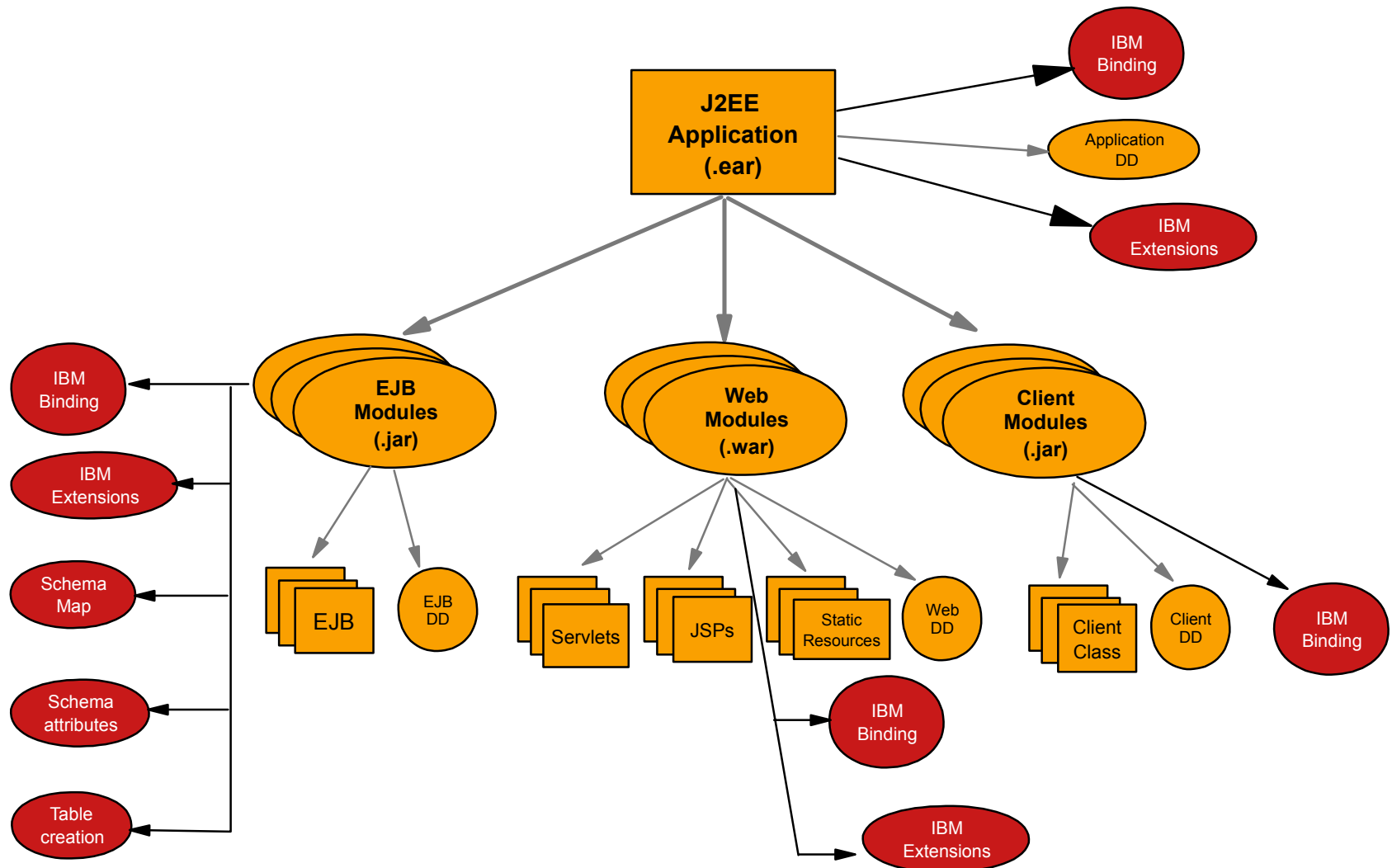
J2EE Packaging



WebSphere 4.0 Bindings

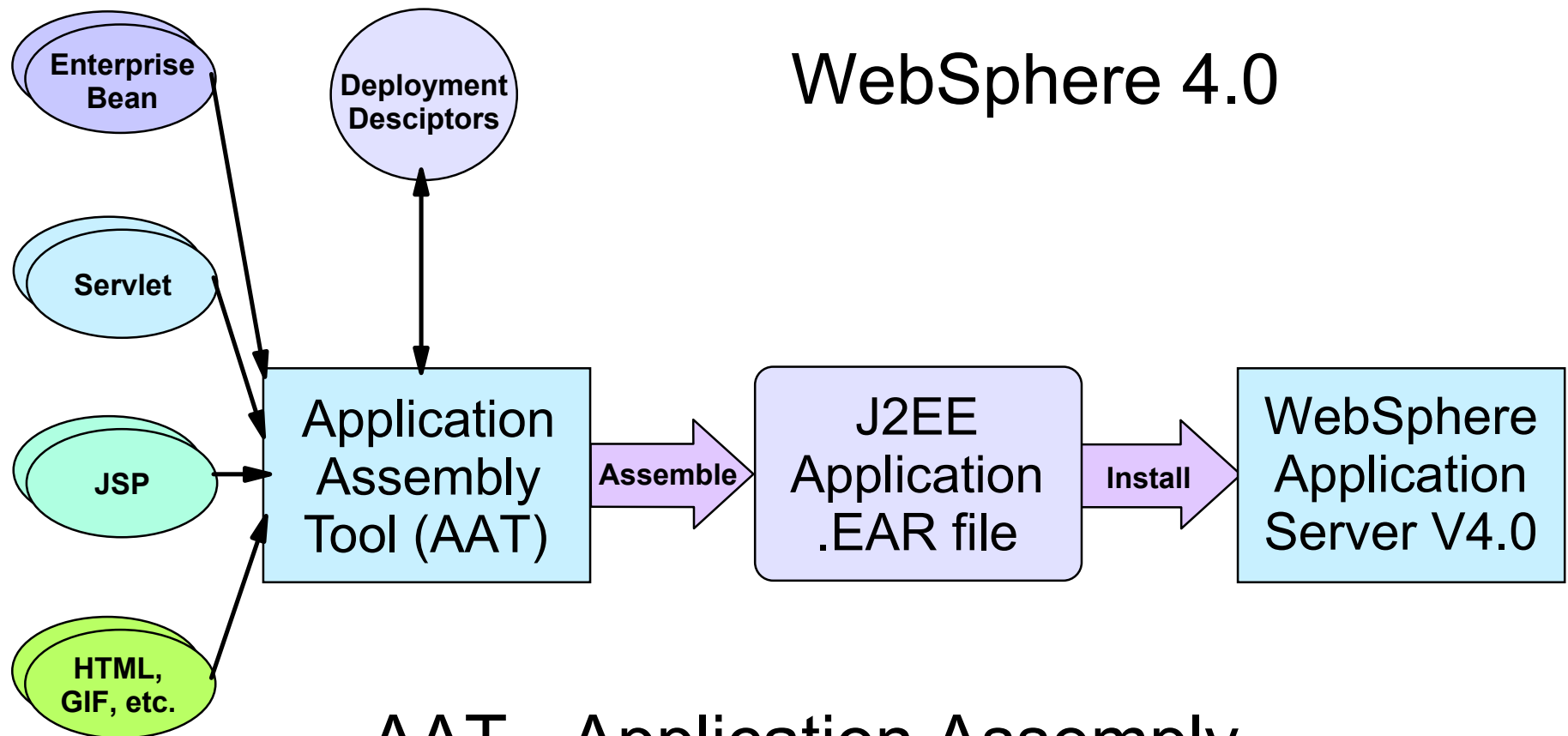
- J2EE provides a mechanism to use local names for external EJB and Resource (for example, JDBC, JavaMail, JMS) objects
- The J2EE specification does not define how objects are referenced in the runtime
- Each J2EE vendor has their own implementation
- WebSphere 4.0 defines *bindings* for this purpose
 - ▶ bindings are configured in the AAT
 - ▶ stored in the EJB, Web, or Application Archive in a file called *ibm-type-bnd.xmi*

WebSphere Application Packaging



Application Assembly and Deployment

WebSphere 4.0



AAT - Application Assembly

Tool

- Creates, Maintains EAR, War, and Jar files
- Performs assembly
- Prepares application for deployment

Administration Console

WebSphere Advanced Administrative Console

Console View Tools Help

WebSphere Administrative Domain

- Virtual Hosts
- ServerGroups
 - Default Model
 - DummySGAppServer
 - DummySGInstalledEJBs
 - DummySGInstalledWebs
 - Nodes
 - wsbeta004
 - Application Servers
 - Default Server
 - Installed EJB Modules
 - Installed Web Modules
 - Default Clone1
 - Installed EJB Modules
 - Installed Web Modules
 - Generic Servers
 - EnterpriseApps
 - Resources

Name	Status	Startup Attempts	Debug Enabled	Thread Pool Size
Default Server	Running	2	False	20
Default Clone1	Stopped	2	False	20

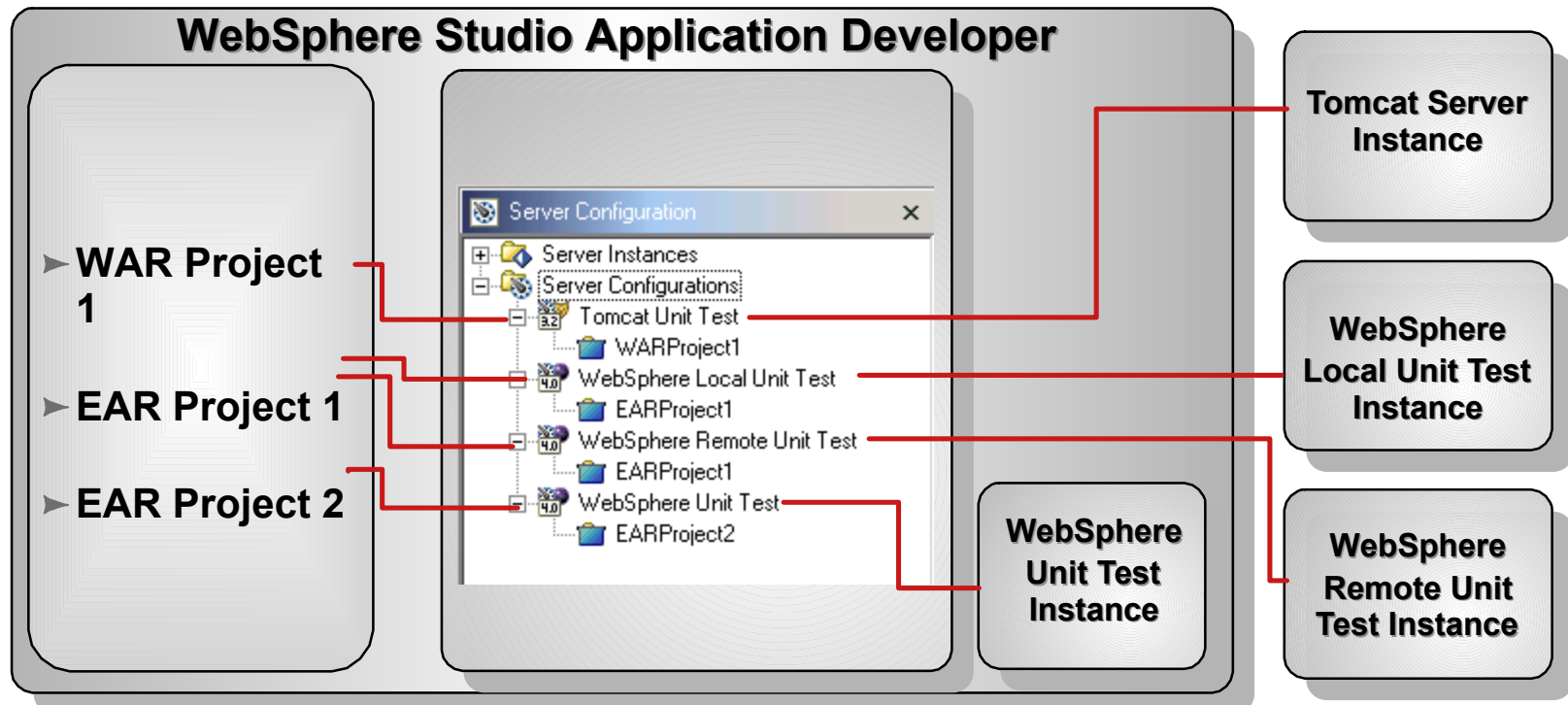
IBM WebSphere
Systems Management



Steps for Configuring WebSphere for J2EE Application

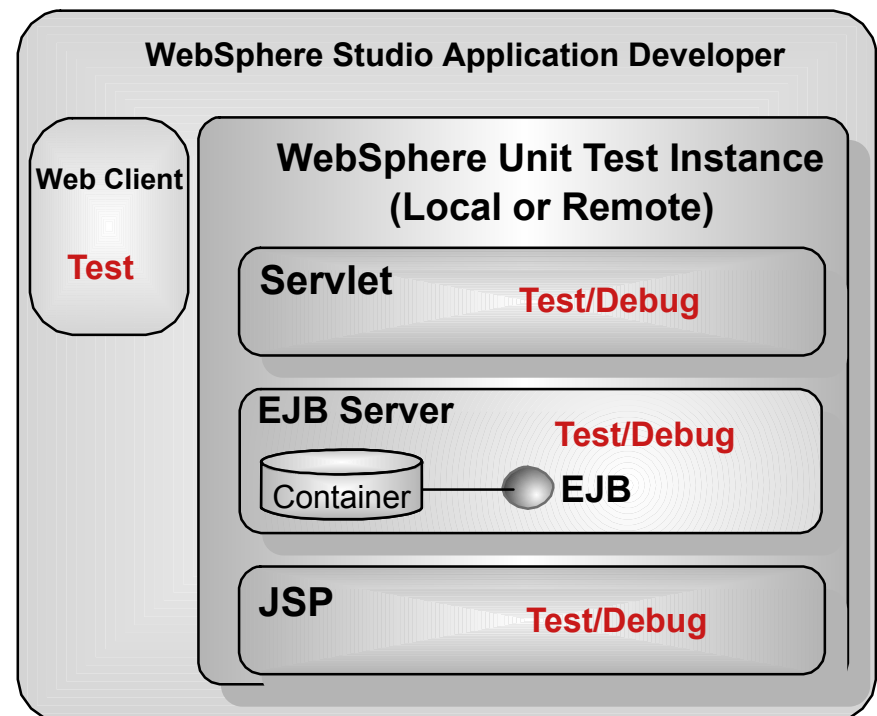
1. Prepare the environment (datasources, virtual hosts, etc.)
 - [Admin Console](#)
2. Create Application Server - [Admin Console](#)
3. Create Application Bindings - [AAT](#)
4. Define EJB JNDI Names - [AAT](#)
5. Define DataSources for Entity Beans - [AAT](#)
6. Bind EJB References to EJB JNDI Names - [AAT](#)
7. Bind Web Modules to Virtual Hosts - [AAT](#)
8. Configure Enterprise Application Security Roles - [AAT](#)
9. Move EAR file created by AAT to installableApps
 - [Admin Console](#)
10. Open EAR file and deploy it - [Admin Console](#)

WebSphere Studio Application Developer



Unit Testing a J2EE Application

- Unit Test Instance can be run in debug mode
 - ▶ Test/Debug Servlets
 - ▶ Test/Debug EJBs
 - ▶ Test/Debug JSPs
- Web Browser runs in source view in workbench
 - ▶ Internet

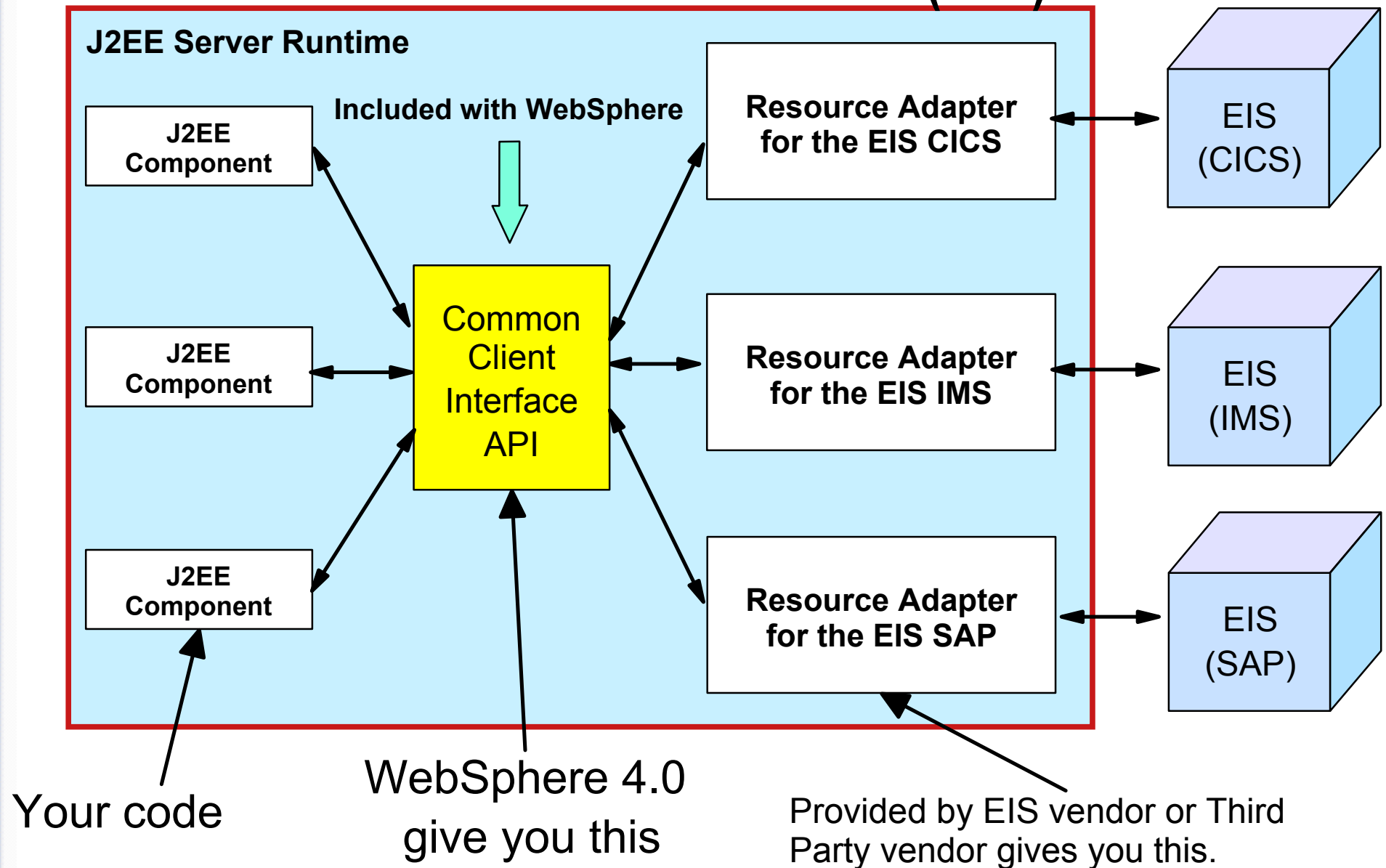


WebSphere Tooling Support

- Built-in WebSphere Unit Test Environment
 - ▶ WebSphere 4.0.1 AEs
- Built-in support for local/remote testing with WebSphere Application Server
- Full support for generating WebSphere deployed J2EE modules:
 - ▶ EAR, WAR, deployed EJB JAR
 - ▶ Command line generation of WebSphere deployed EJB JAR
 - ▶ Support for WebSphere J2EE extension bindings

***Runtime+Tools=Successful
e-business applications***

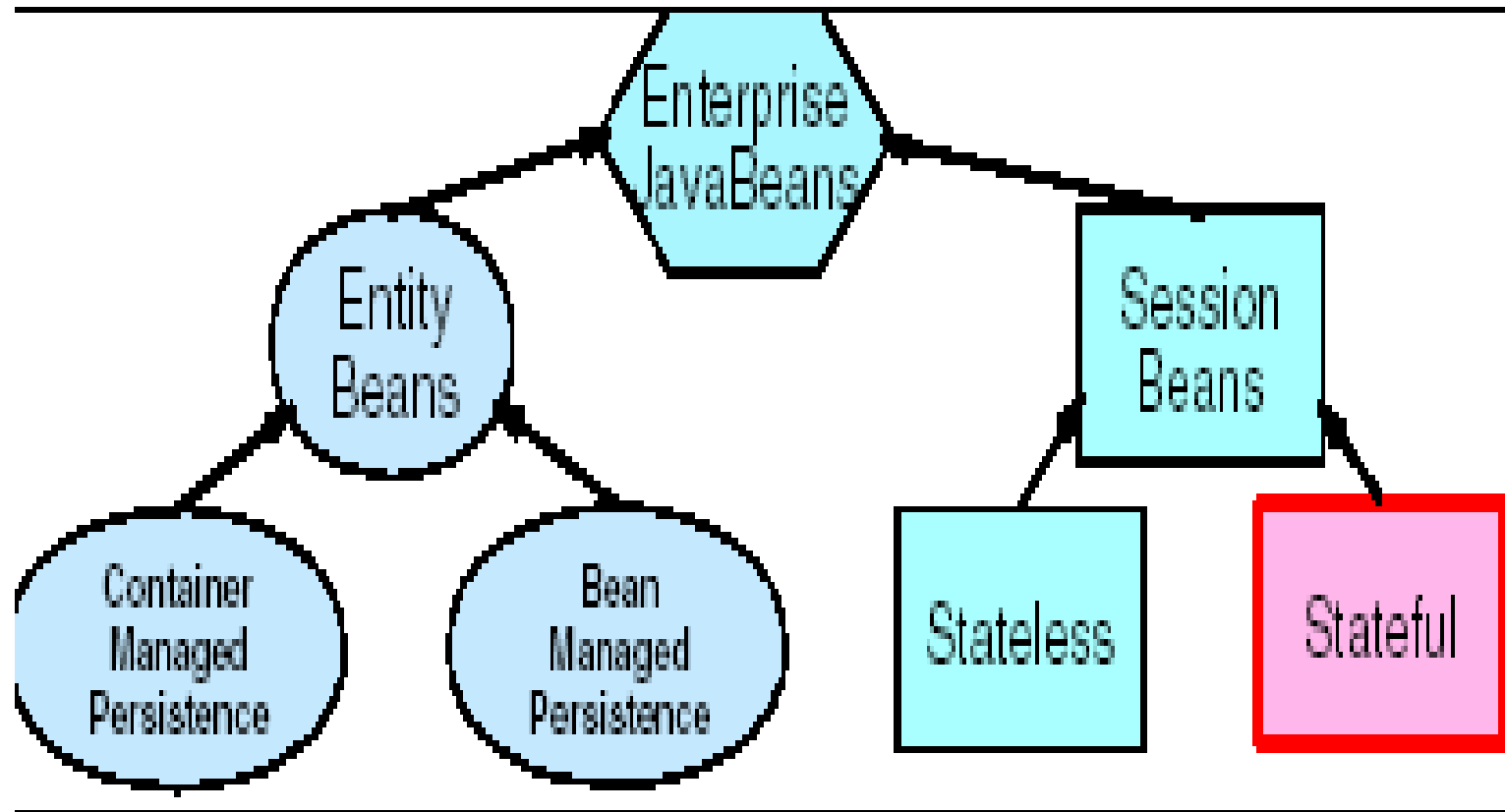
J2EE Connector Architecture (J2C)



WebSphere JCA Support

- Connection Management
 - ▶ ConnectionManager implementation
 - ▶ ConnectionEventListener implementation
 - ▶ Pool Manager implementation
- Transaction Management
 - ▶ Satisfied by JTA implementation
 - ▶ Connection sharing will be assumed within global transactions
 - ▶ CCI local transactions
- Security Management
- Logging/tracing for Resource Adapters

Enterprise Bean



Developing Entity Beans

Unless you have a good reason for implementing BMP, it is recommended that you design your entity beans to use CMP. The code for an enterprise bean with CMP is easier to write and does not depend on any particular data storage product, making it more portable between EJB servers.

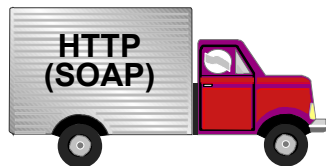
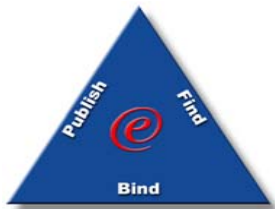


- In an entity bean with container-managed persistence (CMP), the container handles the interactions between the enterprise bean and the data source. In an entity bean with bean-managed persistence (BMP), the enterprise bean must contain all of the code required for the interactions between the enterprise bean and the data source. For this reason, developing an entity bean with CMP is simpler than developing an entity bean with BMP.
However, you must use BMP if any of the following is true about an entity bean:
 - ▶ **The bean's persistent data is stored in more than one data source.**
 - ▶ **The bean's persistent data is stored in a data source that is not supported by the EJB server that you are using.**
- In entity beans with CMP, the persistence service handles nearly all of the tasks required to manage persistent data. In entity beans with BMP, the bean itself handles most of the tasks required to manage persistent data.
- BMP is ideal when data needs to be stored in legacy storage systems or file systems. However, the increased flexibility comes at the price of decreased portability.

WebSphere Persistence

- When persistent sessions are enabled, the Session Manager will persist session information into the data source specified by the data source connection settings. Otherwise, the Session Manager will discard the session data when the server shuts down.
- Whenever data is shared across multiple application servers (JVMs), persistent sessions are used as a method to share that data. This occurs when cloning application servers (WLM), or having two applications, in separate application servers, talk to each other using sessions, or when configuring WebSphere clusters with fail-over support.
- The persistence service ensures that the data associated with entity beans is properly synchronized with their corresponding data in the data source. To accomplish this task, the persistence service works with the transaction service to insert, update, extract, and remove data from the data source at the appropriate times.
 - ▶ The persistence service uses the following components to accomplish its task:
 1. The Java Database Connectivity (JDBC), which gives entity beans a common interface to relational databases.
 2. Java transaction support (JTS). The EJB server ensures that persistent data is always within the appropriate transactional context.





Web Services

J2EE without XML and Web services is an incomplete environment**J2EE APIs:**

- Servlet API 2.2 web device support
- JSP Java ServerPage 1.1 dynamic data display
- EJB Enterprise JavaBeans 1.1
- JDBC 2.0 RDB access
- JNDI Java Naming and Directory Interface 1.2
- JavaIDL java to corba communications
- RMI-IIOP corba based transport
- JTA/JTS Java Transaction API 1.0
- JMS Java Message Service 1.0

J2EE standards relevant to implementing business services

Web Services standards:

- **XML** XML4J 3.1.1 (Xerces 1.2)
Extensible Markup Language
Inter-system data exchange
- HTML/WML/VXML Pervasive device interface
- **SOAP** 2.2 MS COM+/Java integration
etc. component model independent
- **UDDI** UDDI4J 1.0.3 Universal
Description, Discovery, and Integration
- **WSDL** Web Services Definition
Language

Web services provide access to business services

What are Web Services?



- A way to catalog and describe services
 - ▶ **WSDL** - Web Services Description Language
- A way to find out about available services
 - ▶ **UDDI** - Universal Discovery Description and Integration
- Network-neutral service access protocol
 - ▶ **SOAP** - Simple Object Access Protocol
- A common language for exchanging information
 - ▶ **XML** - eXtensible Markup Language



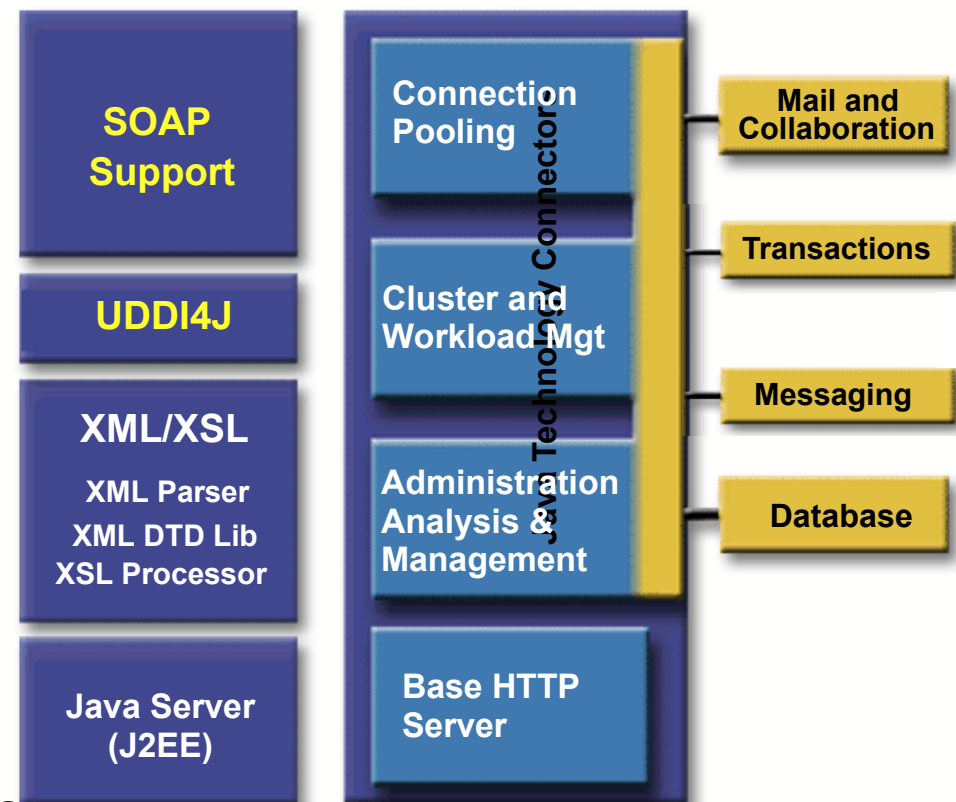
Web Services Assumptions

- Business functions
 - ▶ invoked over the Internet
 - ▶ use industry standard transports
 - ▶ typically use XML for information interchange
- Building blocks for complex web applications
 - ▶ low technology barriers (XML, HTTP)
 - ▶ few temporal constraints
 - no long locks
 - allow think time and latency
 - enable independent development cycles
- Interoperability must be preserved

WebSphere 4.0 Support for Web Services

- Integration of SOAP Support
 - ▶ SOAP server and client environments
 - ▶ Enables WebSphere applications to send and receive SOAP messages, and leverage WSDL
- Integration of UDDI4J
 - ▶ Java interface to UDDI registries
 - ▶ Enables WebSphere applications to communicate with UDDI-compliant registries to publish and find Web services

WebSphere Application Server 4.0



Web Services Value for Developers

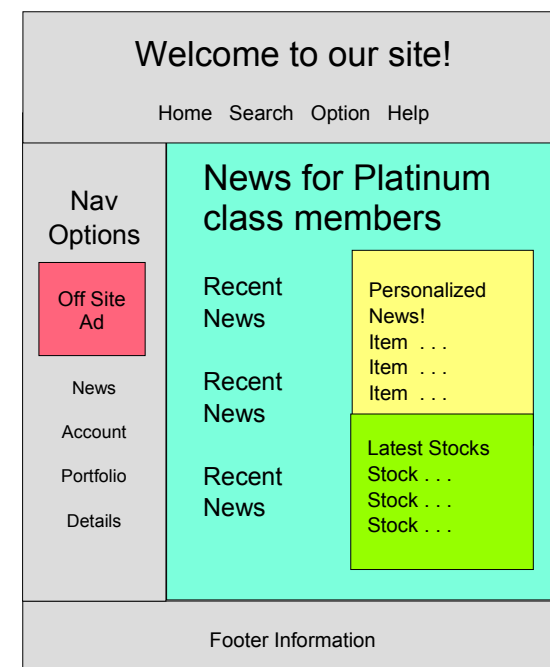
- Backed by key software vendors (IBM, Microsoft, Oracle, etc.) and Open Source organizations (Apache)
 - Web Services tooling
 - developers can concentrate only on writing the business logic
 - developers do not need to write the Web Services "plumbing" code
- Implementation is de-coupled from interface
 - developers that create or use a Web Service are **not restricted** to any particular language, operating system or object model

Web Services Success

- Other distributed technologies are not optimized for the Internet because they required strong coupling:
 - ▶ required the use of the same transport technologies
 - ▶ required knowledge of each others' implementation
- Limitations of other distributed technologies
 - ▶ **DCOM - requires Windows at each endpoint**
 - ▶ **CORBA - requires compatible ORBs at each endpoint**
 - ▶ **RMI - requires Java at each endpoint**
- XML/WSDL/SOAP/UDDI advantages
 - ▶ implementation choices are de-coupled from interfaces
 - any language can be used to write and consume Web Services
 - more than one open standard transport technology can be specified
 - operating system differences are not a factor

- WebSphere can cache fragments of pages, notably individual Servlet and JSP requests
- Reduces both load and response time
- Performance gains with:
 - ▶ Static Fragments
 - header JSPs, navigation bars, etc.
 - ▶ Dynamic Fragments/Pages
 - **stock quotes**, **search results**, **levels of service**
 - **personalized pages** using shared information (e.g. MyNews)
- Application developers control how fragments are cached
 - ▶ Define rules based on Servlet, URI, request/session variables, etc.
- Rule-based, time-based, and programmatic techniques for invalidating cache entries. Can control external caches, e.g. IBM Edge Server

Cache entries created based on unique values or rules supplied by end-user
eg. <http://www.ibm.com/servlet/stockquote?symbols=IBM>



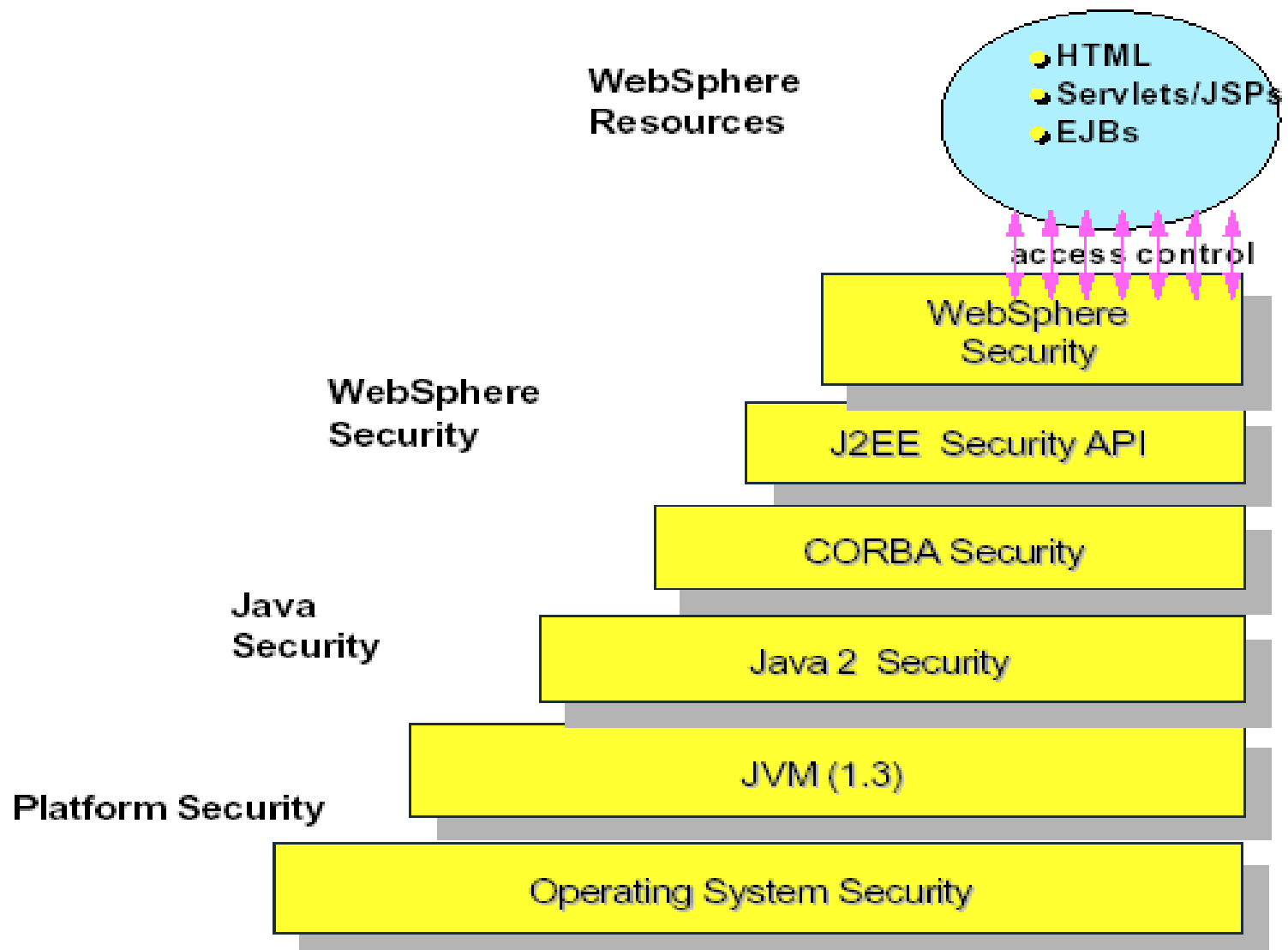
Building a Secure Environment



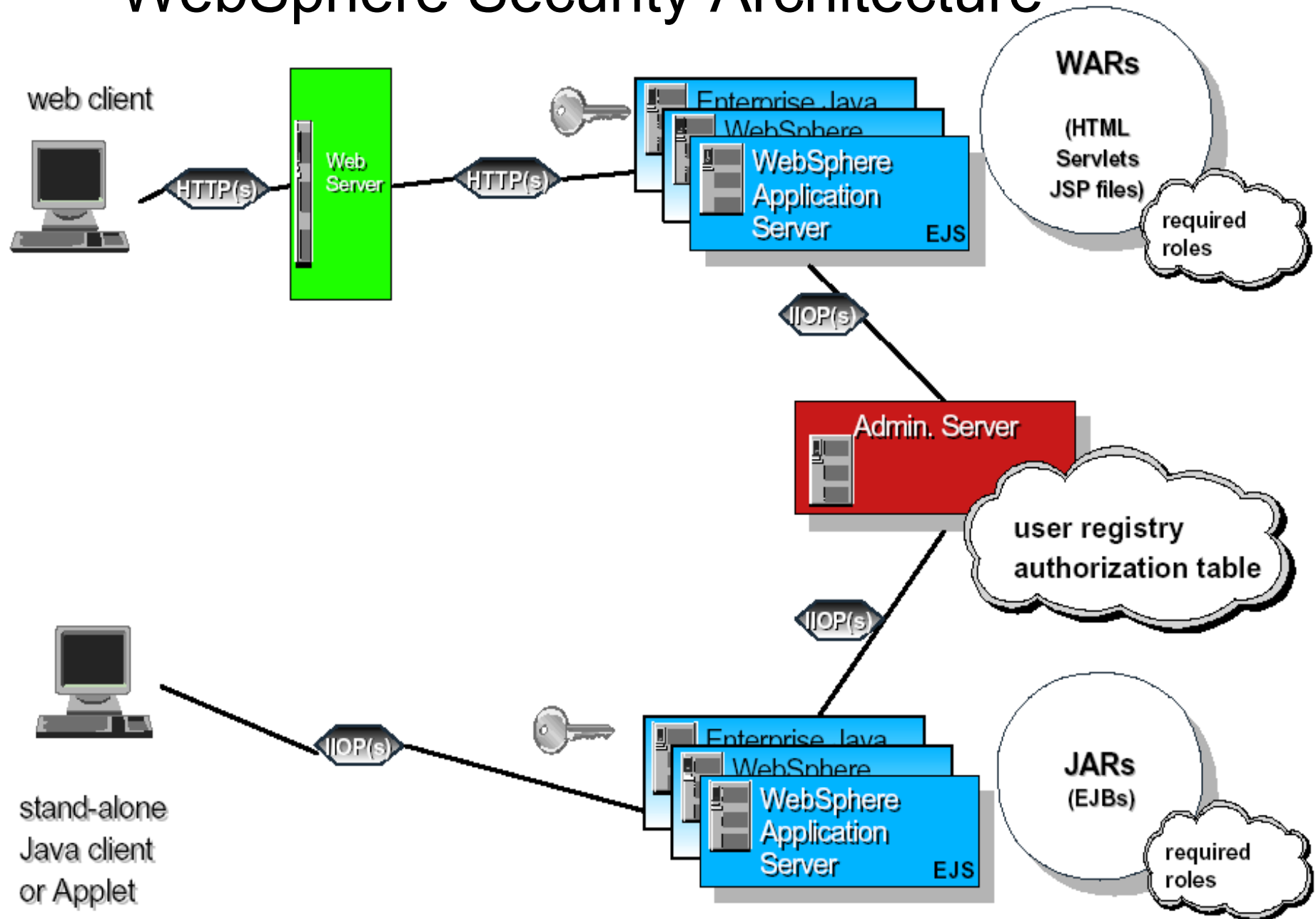
Identification, Authentication, Authorization

- Operating System
- J2EE
- Transports, Layers and Protocols
- Client Browser
- Web Server
- WebSphere
- User Registry
- Application Components

WebSphere Security Layers



WebSphere Security Architecture



WebSphere Security

- WebSphere provides security components that provide or collaborate with other services to provide authentication, authorization, delegation, and data protection.
- WebSphere also supports the security features described in the J2EE specification.
- Security is established at two levels.
 1. The first level is global security. Global security applies to all applications running in the environment and determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults.
 2. The second level is application security. Application security, which can vary with each application, determines the requirements specific to the application. In some cases, these values can override global defaults. Application security includes settings like mechanisms for authenticating users and authorization requirements.

WebSphere Security Components

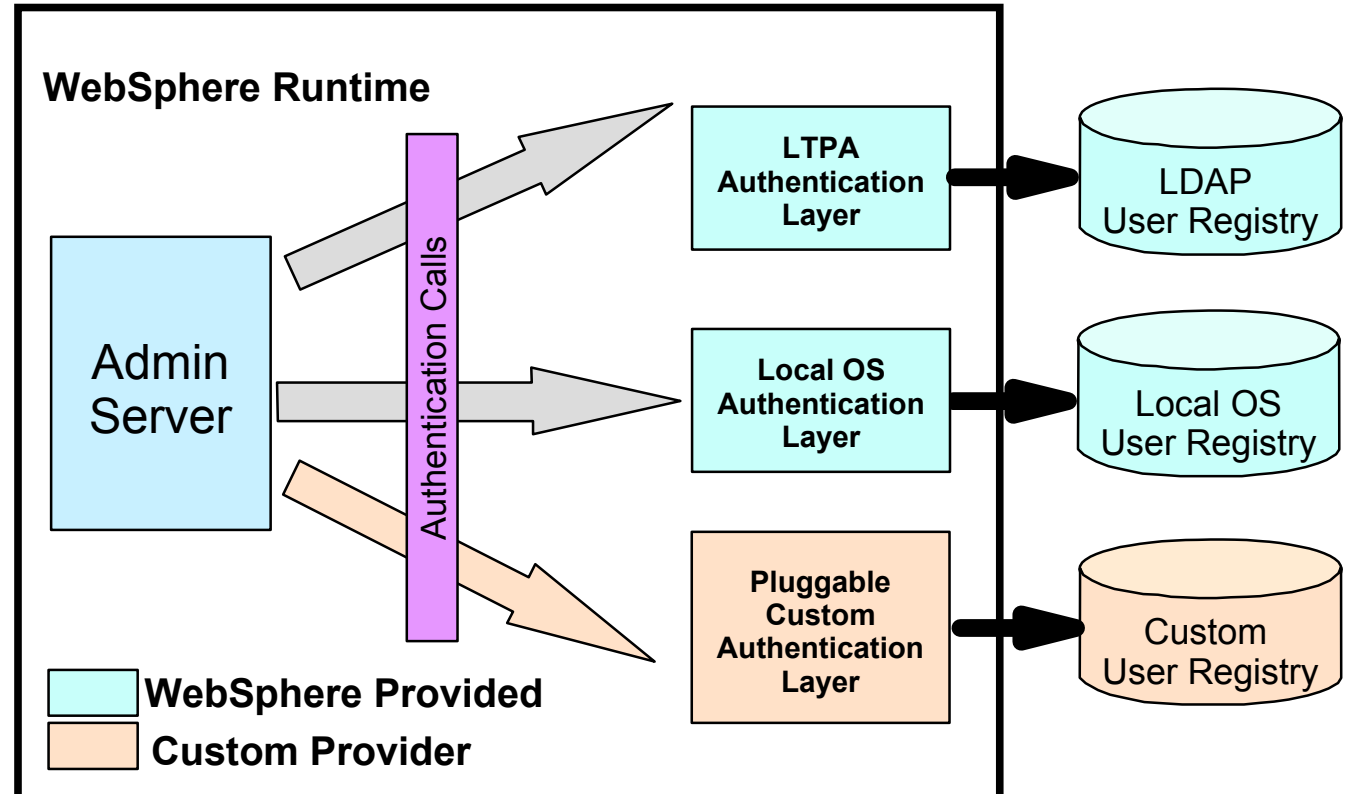
- **Security Collaborators** - Enforce the security constraints and attributes specified in the deployment descriptors.
- **Security Policies** - Specify security attributes include roles, method permissions, the run-as mode or delegation policy, login-configuration or challenge type, and data-protection (confidentiality and integrity) settings.
- **The Secure Association Service (SAS)** - performs authentication for Java clients of enterprise beans and helps to provide message protection or encryption between such clients and WebSphere application servers using RMI/IIOP over SSL for communication. SAS also provides message protection between WebSphere application services.
- **The User Registry**
- **SSL**

WebSphere Authentication

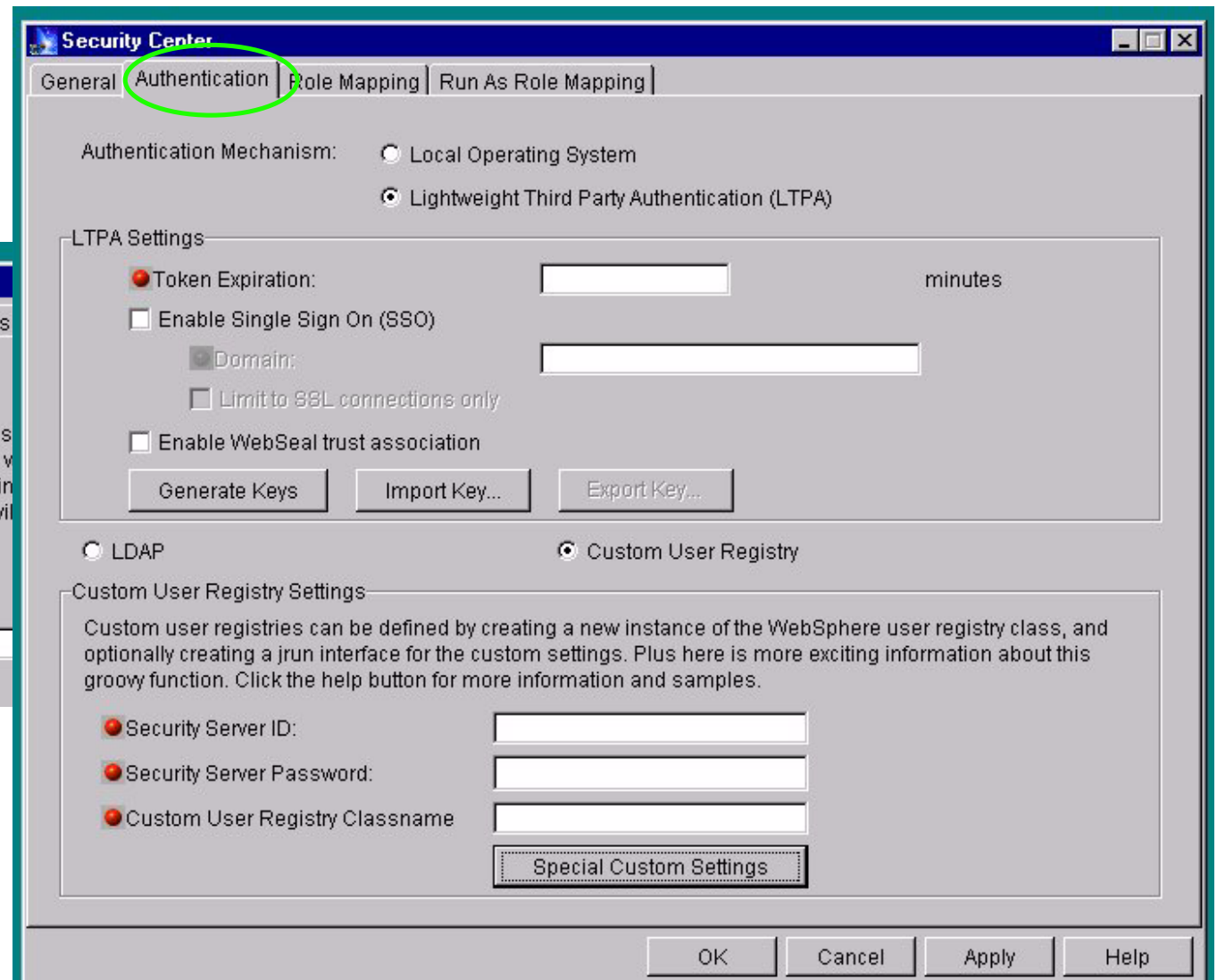
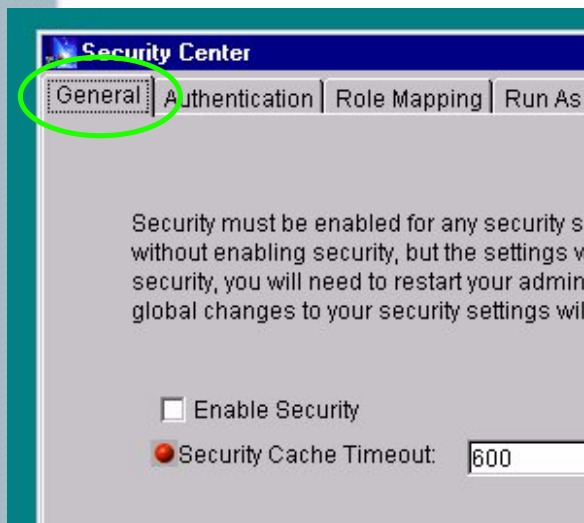
- WebSphere security authenticates a principal based on the authentication policy associated with the resource the principal has requested.
- The WebSphere server supports authentication mechanisms based on validating credentials, such as certificates, tokens, or user ID and password pairs. Credentials are verified against a user registry that supports such a scheme.
- The WebSphere server also supports a third-party authentication scheme.
 - ▶ The client and server principals are authenticated to a mutually trusted third party, a Lightweight Third Party Authentication (LTPA) server.
 - ▶ Any authentication scheme must assume that the client does not trust the server, and the server does not trust the client.
 - ▶ The main benefit of a third-party authentication mechanism is that the user registry is administered centrally.
 - ▶ The authentication policy for performing authentication between a user and a WebSphere server can be specified in terms of the J2EE login configuration tags as part of a Web application's deployment descriptor

WebSphere provides a common security directory based on LDAP for customers to address the proliferation of application-specific directories, a major driver of high costs. WebSphere Supports LDAP directories by:

- Netscape Directory Server
- Lotus Domino Enterprise Server
- Microsoft Active Directory
- Novell Directory Services (NDS) eDirectory
- IBM SecureWay Directory - bundled with WS for free



Security Center GUI



Tivoli Security Solution

Tivoli software

- SecureWay Policy Director
- SecureWay Policy Director for MQSeries
- SecureWay User Administration
- SecureWay Privacy Manager
- SecureWay Global Sign-On
- SecureWay Public Key Infrastructure
- Tivoli Risk Manager
- Tivoli Identity Director

Additional Information and Trial Code

- **V4 Handbook Redpiece - SG24-6176**
www.redbooks.ibm.com
- **V4 AD Redbook - SG24-6134**
www.redbooks.ibm.com
- **WebSphere InfoCenter**
www.software.ibm.com/webserver
- **IBM WebServers Site**
www.software.ibm.com/webserver
- **WebSphere Developers Domain (WSDD)**
www.software.ibm.com/webserver
- **VisualAge for Java Developers Domain (VADD)**
www.software.ibm.com/ad
- **IBM Partner World for Developers**
www.developer.ibm.com
- **WebSphere Central Site**
www.WebSphereCentral.com



Thank You!

