



IBM Software Group

IBM® WebSphere® Application Server V6

Architecture Overview



@business on demand.

© 2004 IBM Corporation
Updated February 2, 2005

This presentation will focus on the Architecture of IBM WebSphere Application Server, Version 6.

Goals

- Understand the high level architecture of WebSphere Application Server V6
- How it fits in the Enterprise environment



The goal of this presentation is to provide high level architecture overview of WebSphere Application Server V6 and of Java™ 2 Enterprise Edition (J2EE) applications for new customers. It also provides an overview of what types of clients can be used to call applications running in WebSphere V6.

Agenda

- Application Server in Enterprise environment
- WebSphere Application Server Architecture
 - ▶ Major components inside the Server
 - ▶ Types of Request/Response
 - Supported protocols like HTTP, RMI-IIOP, SOAP/HTTP, etc.

The agenda for this presentation is to briefly examine the positioning of WebSphere Application Server V6 in the enterprise computing environment, and then to discuss its architecture.

Overview

- IBM WebSphere Application Server provides the foundation to run many high-volume, business-critical enterprise applications
 - ▶ It provides an environment to run J2EE applications
 - ▶ WebSphere Application Server V6 delivers a Services Oriented Architecture hosting infrastructure
- Multiple offerings to fit the needs of different market segments
 - ▶ From small businesses to large enterprises
- Middleware foundation for IBM and 3rd party products
 - ▶ Example: WebSphere Portal, WebSphere Business Integration, etc.



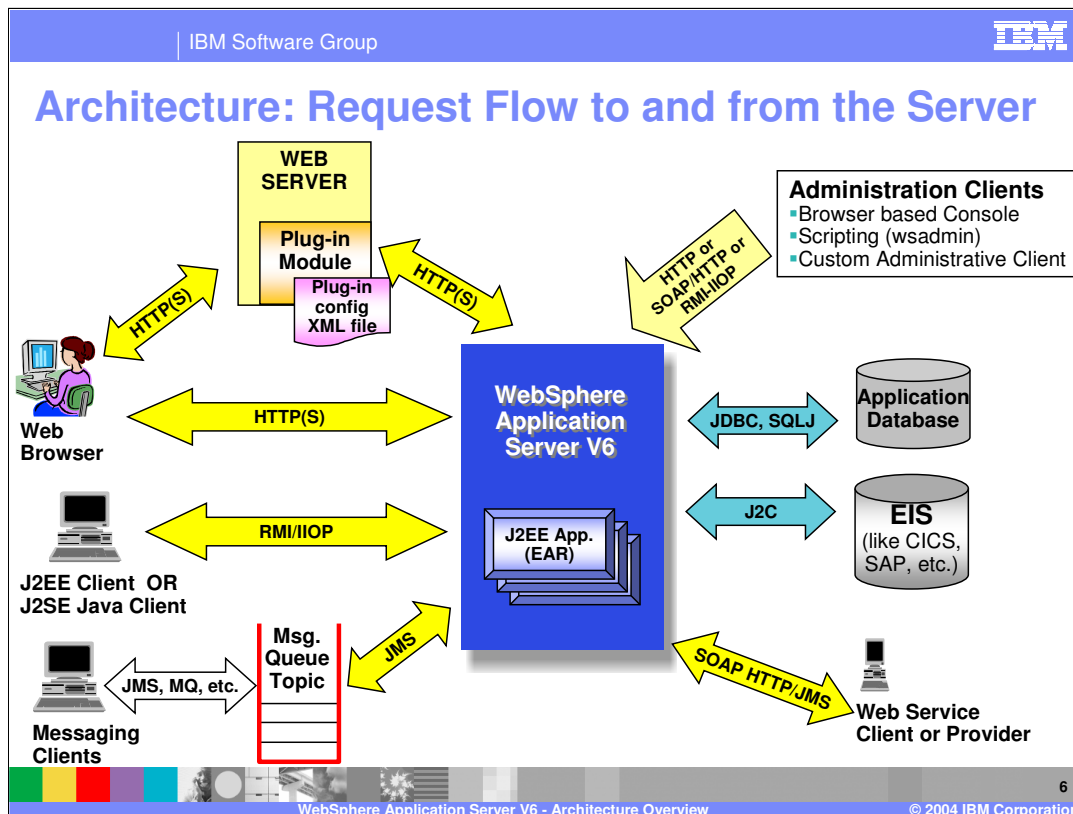
WebSphere Application Server V6 continues and expands on the tradition of providing a platform for you to host your critical business applications that require secure, transactional capability and being able to access host of back ends.

Section

The Role of the Application Server in Enterprise Computing



This section examines the positioning of WebSphere Application Server in the enterprise computing environment.

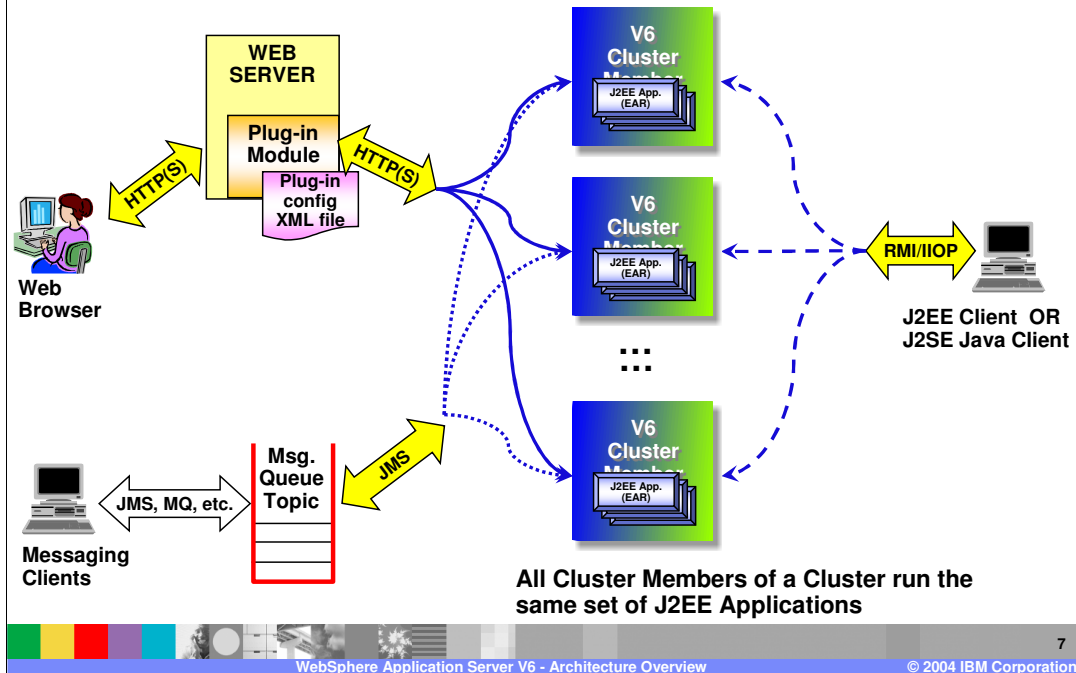


There are various entry points or clients to interact with the J2EE applications running with the WebSphere Application Server. The clients could be Web based, stand-alone Java or J2EE clients, Messaging Clients using Java™ Messaging Service (JMS), or Web Services clients like another J2EE server or .Net.

Your applications can interact (read/write) dynamic data hosted on back ends like Databases, or Enterprise Information Service providers like CICS, IMS or third party EIS systems like Siebel, PeopleSoft, and so on.

WebSphere Application Server also provides an open set of Java™ Management Extensions (JMX) APIs to manage and administer it in a distributed environment. The Administrative clients provided by WebSphere use the same JMX APIs.

Workload Management



WebSphere also provides clustering capability whereby you can have multiple application servers running the same application, sharing the work load, and providing failover capability. This will allow very high throughput and by distributing the Application servers across multiple machines, you could provide a high degree of failover.

This feature is present in the Network Deployment and higher packages.

Section

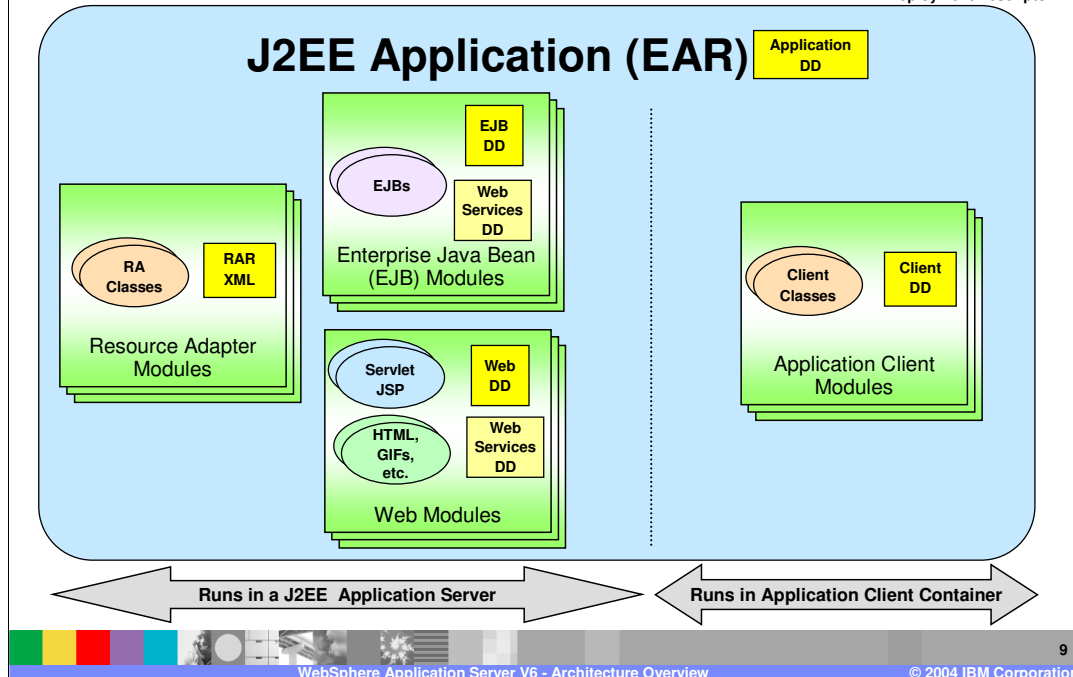
J2EE Application Components and Application Server Components



This section illustrates the relationship between J2EE Components and WebSphere Application Server components.

J2EE 1.4 Application Packaging

DD = Deployment Descriptor



A structure of a J2EE application is shown here. All modules, except the Application Client module run within the Application Server. The modules running within the Application Server are Web Modules, EJB modules and Resource adapters. Each module has a standard J2EE defined deployment descriptor that defines the content of the module and its deployment properties

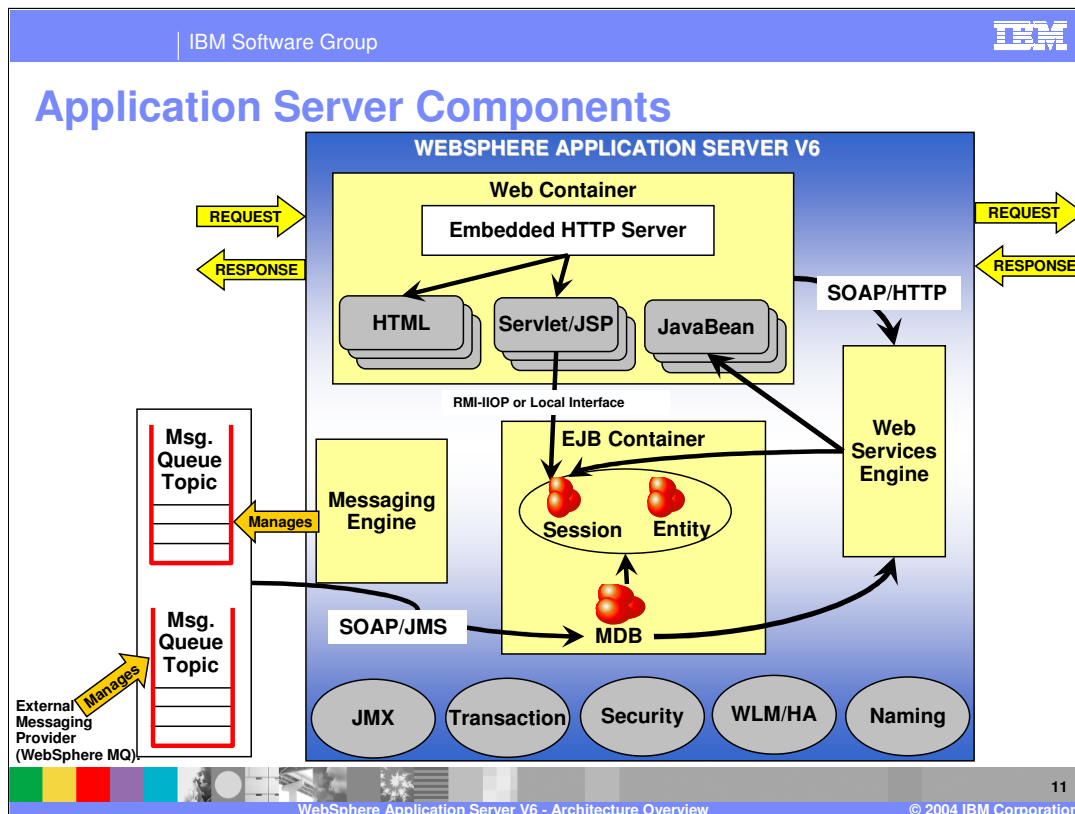
Section

WebSphere Application Server

Types of Request/Response



This section covers some of the requests and responses supported by WebSphere Application Server V6. Through this, you will see the variety of clients that interact with the server.



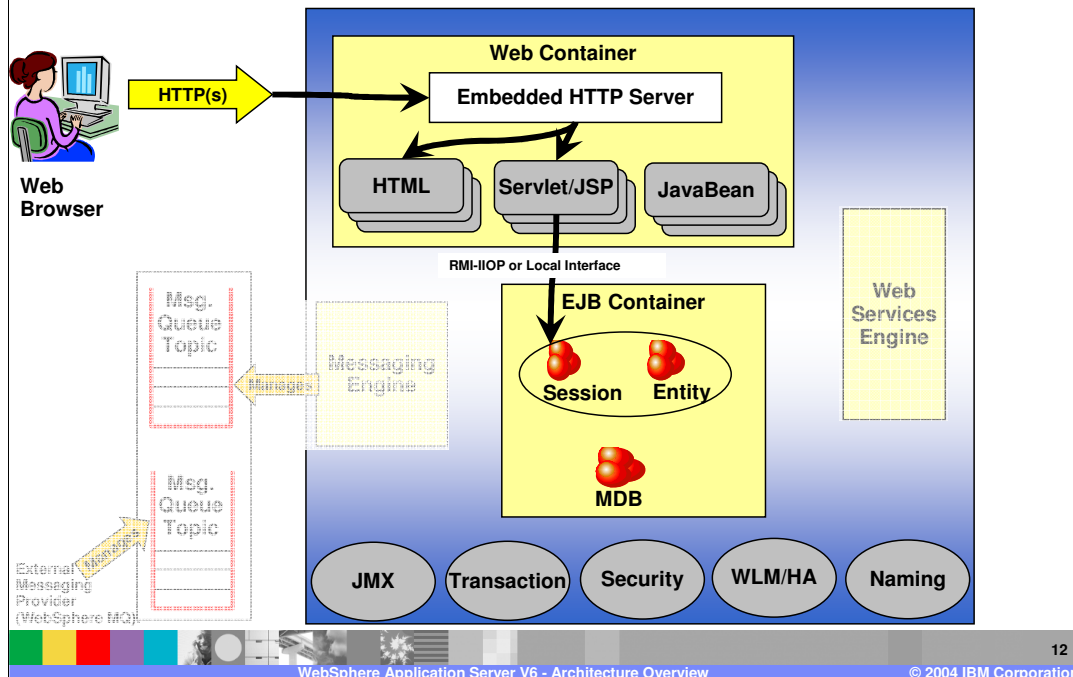
Here you see the major components of the WebSphere Application Server, but not all of them.

The Web container runs the Servlets, JSPs, and HTMLs present in the Web Modules. It also contains an embedded HTTP port through which the Web clients (like browsers) send HTTP requests.

The Enterprise Java™ Bean (EJB) container runs the EJB modules. There is a Web Service engine to process the Web Service request into the WebSphere Application Server and send a response back.

The Service Integration Bus provides the foundation of Enterprise Service Bus capability and Service Oriented applications

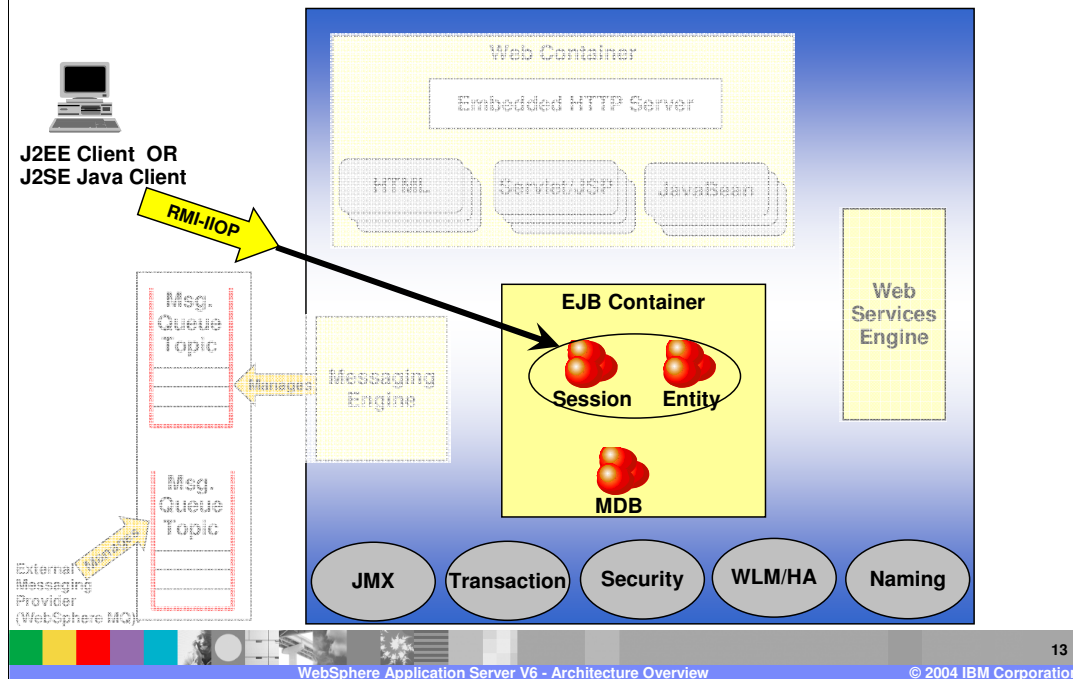
Web Application HTTP) Request flow



The HTTP or HTTPS request coming directly from a Web client (like a Browser) or through a Web Server like the IBM HTTP server is received by the embedded HTTP server within the Web container. The request then flows to the appropriate Web artifact – servlet, Java™ Server Page (JSP) or HTML. The response generated by these Web artifacts flows back the same route to the client.

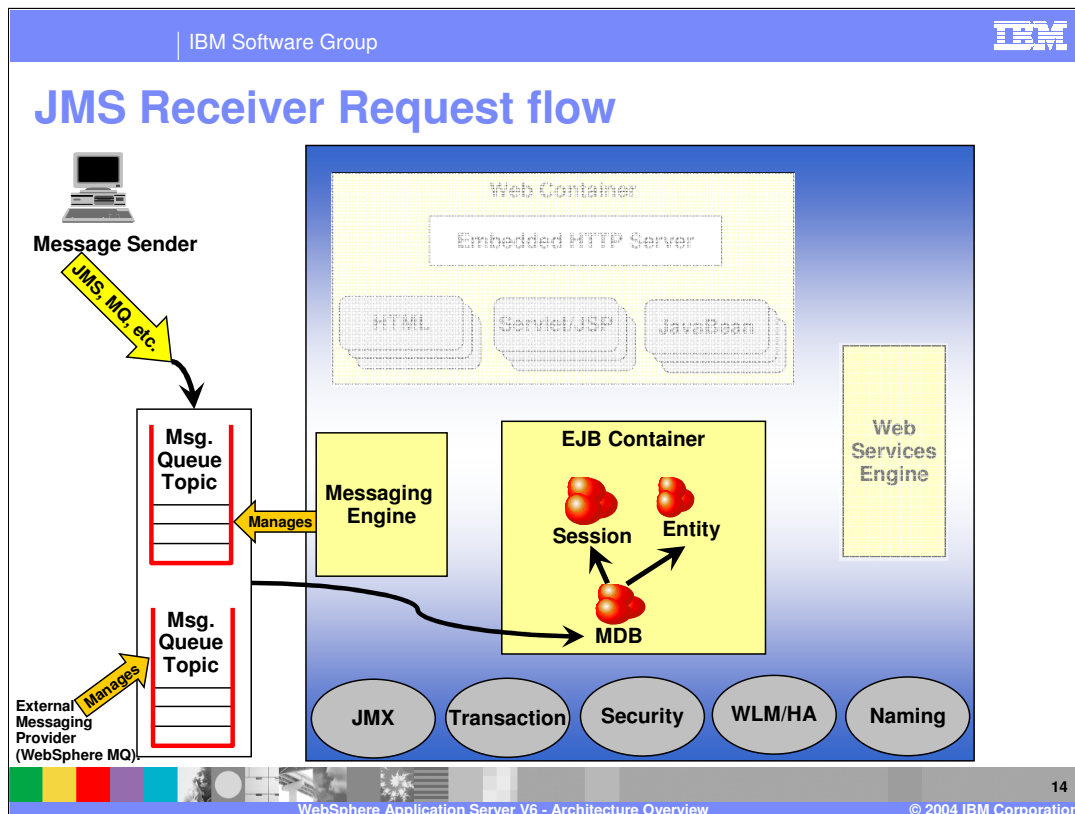
The Web artifact can call other artifacts like an EJB on the same server or different server.

RMI-IIOP Request flow



Here, the stand-alone Java or J2EE client sends a request to a session or Entity EJB running with the Server via RMI-IIOP protocol.

Even though not shown here, you could have another J2EE server send a request to an EJB via RMI-IIOP.



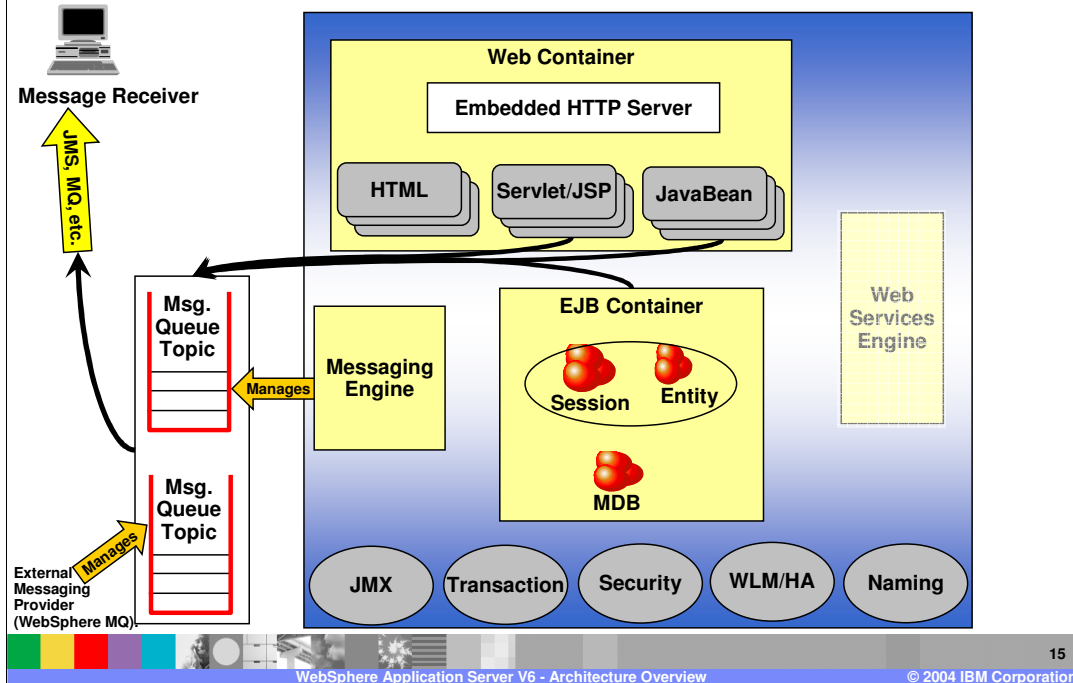
Included in WebSphere Application Server V6 is the Server Integration Technology that contains Service Integration Bus that provides the embedded JMS Messaging functionality within the server.

A Message Sender puts a message on a Destination (Queue or Topic) managed by the System Integration Bus.

An Alternative could be to use WebSphere MQ and receive messages from MQ.

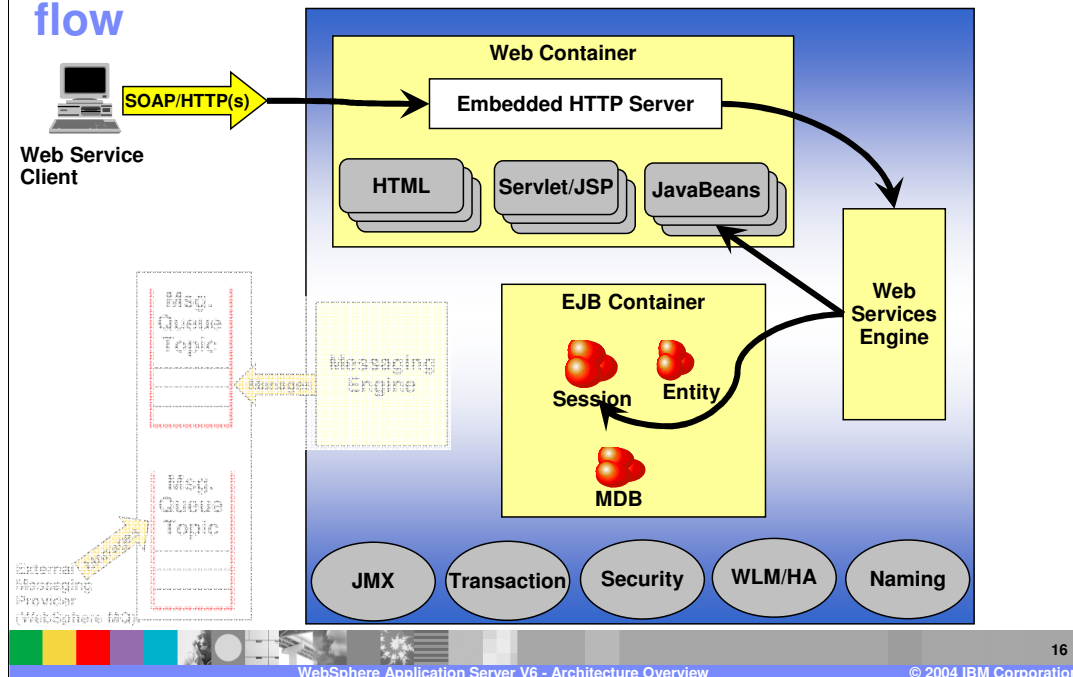
The receiver of the message would be the Message Driven EJB, where the container can handle working with the Destination. You could write your code to poll the Destination and get the message. However, Message Driven EJB is a better implementation and lets the container manage the destination.

JMS Sender Request flow



Any application artifact (Servlet, JSP, EJB, and so on) running within the WebSphere Application Server has the ability to put a message on a destination that is managed by the Service Integration Bus or that is managed externally by WebSphere MQ. Any WebSphere or non-WebSphere client can retrieve the message.

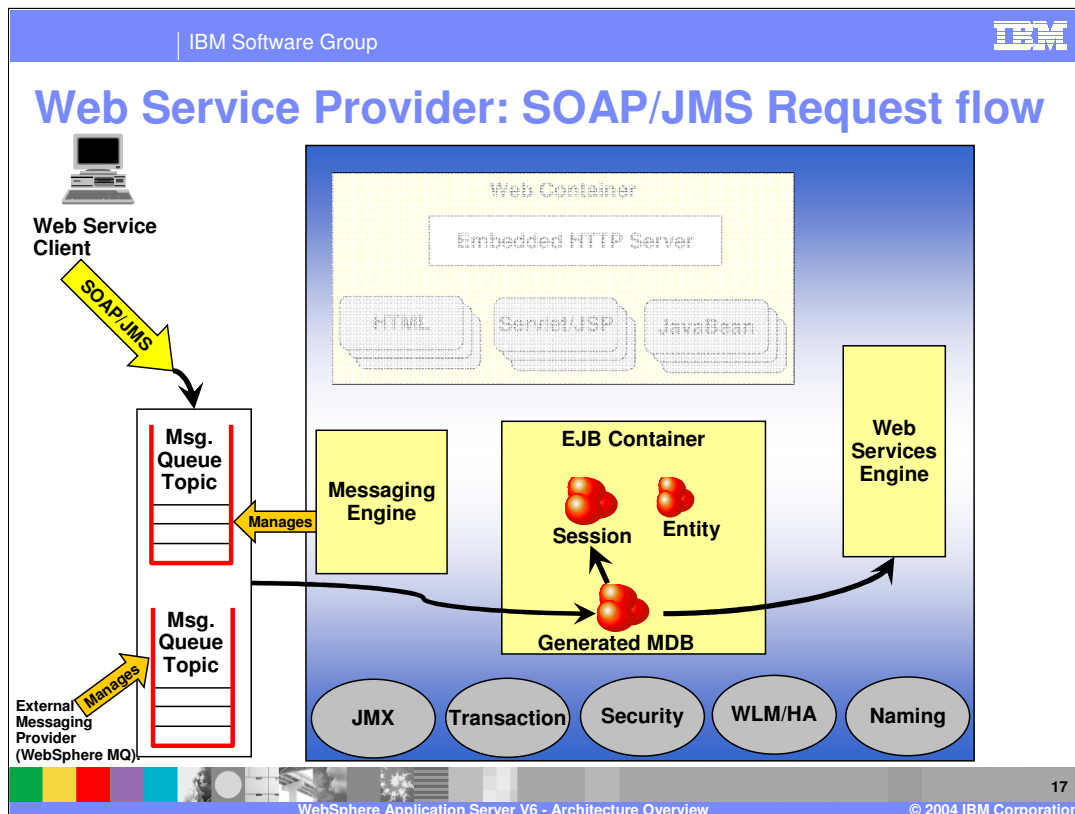
Web Service Provider: SOAP/HTTP Request flow



WebSphere Application Server V6 supports exposing Javabeans™, or a Stateless Session EJB, or both to be Web Service providers. For EJB Web Service providers, Web Services clients can then send SOAP requests using HTTP or JMS transport mechanisms.

The EJB container running the EJB provider does not have the ability to receive HTTP requests. Only the Web container can receive the HTTP request.

For an EJB Web Service provider, a Web module is generated to be the receiver of the HTTP request. That Web Module can then call the target EJB.

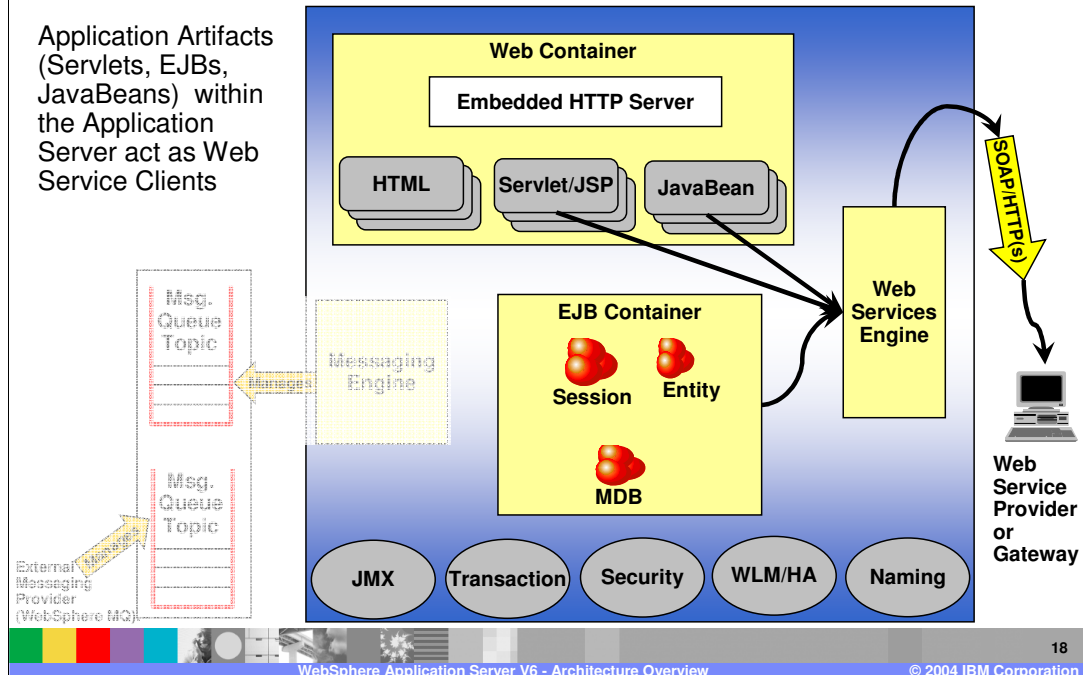


SOAP over JMS supports only Stateless Session EJBs as Web Service Providers.

The EJB container running the EJB provider has the ability to receive a JMS message using a Message Driven Bean (MDB)

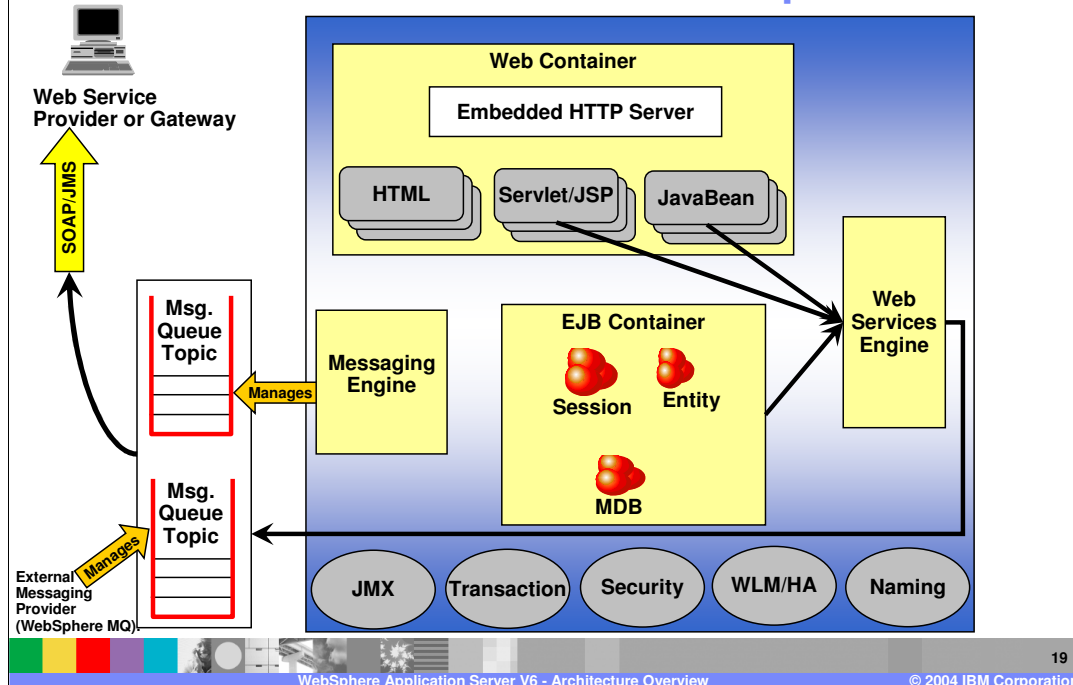
For an EJB Web Service provider, development tools can generate an MDB that can be receiver of the SOAP/JMS message which can then call the target EJB.

Web Service Client: SOAP/HTTP Request flow



An application artifact running within WebSphere Application Server can be a Web Service client sending SOAP requests over HTTP or JMS to a Web Service provider (like another J2EE Server or .Net or other Web Service provider). The Web Service engine creates the SOAP message based on the request from the artifact and sends it to the provider.

Web Service Client: SOAP/JMS Request flow



A Web Service client within WebSphere Application Server V6 can send SOAP request over JMS to a Web Service provider listening on the same JMS queue. Either the destination that is managed by Systems Integration Bus or that is managed externally by WebSphere MQ can be used.

Section

Summary

Summary

- Introduced high level architecture of WebSphere Application Server V6
- Discussed different protocols for request/response to WebSphere Application Server

In summary, this presentation has covered the Application Server architecture at a high level and shown the types of clients and other enterprise infrastructure that WebSphere Application Server can interact with to provide a rich distributed application server environment.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eIogo business	DB2	Series	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

