



# An Overview of Red Hat Application Server

By the Red Hat Application Server Engineering Team

## Abstract

This white paper provides information on Red Hat Application Server. It describes the components of Red Hat Application Server and how they fit into the Open Source Architecture.

Revision 1.0. August 2004

## Table of Contents

Introduction	2
Overview	2
What is an application server?	2
Where does it fit in the Open Source Architecture?	4
What is Red Hat Application Server?	5
JOnAS	6
JOnAS Features	7
JOnAS Architecture	9
Tomcat	11
Struts	12
Apache Web Server	13
Red Hat Developer Suite	13
Summary	14



## Introduction

Red Hat Application Server is an open source middleware platform, layered between the operating system and applications. It is comprised of a runtime system and associated development libraries for creating and deploying Java-based Web applications with dynamic content (e.g., dynamic Web sites, portal servers, and content management systems). These applications might retrieve, display, or update data in database management systems such as PostgreSQL and Oracle, or communicate with standard application software such as ERP systems and with proprietary legacy applications. Red Hat Application Server is based on projects from the ObjectWeb Consortium (<http://www.objectweb.org>) and the Apache Foundation (<http://www.apache.org>).

Included with Red Hat Application Server is Red Hat Developer Suite, which gives application developers a full-featured Integrated Development Environment (IDE) based on the open source Eclipse project (<http://eclipse.org>).

The goal of Red Hat Application Server is to encourage widespread adoption of open source middleware and to simplify Enterprise Linux Web application development and deployment. A Red Hat Application Server subscription includes support at the level of the underlying Enterprise Linux product subscription. Red Hat Application Server is available as an optional Red Hat Network download for Red Hat Enterprise Linux subscribers on IA-32, Itanium, and PPC architectures. Red Hat Application Server is also available without a subscription on <ftp.redhat.com>.

## Overview

This section provides a brief description of application servers in general, shows where application servers fit into the Open Source Architecture, and gives an overview of Red Hat Application Server.

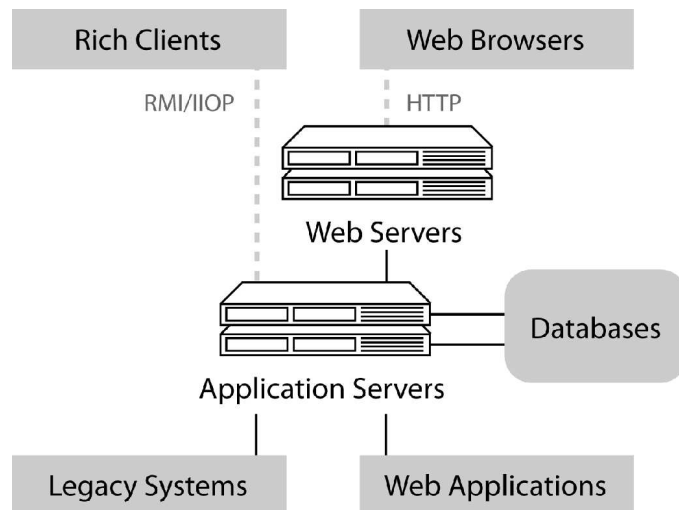
### What is an application server?

An application server provides a more efficient platform for developing Web-based applications by isolating the application from the operating system through standard APIs. Web application development becomes less complex, allowing developers to focus on business logic code rather than infrastructure code such as memory management, multi-threading, and resource allocation. In addition, an application server improves runtime efficiency by simultaneously managing access to distributed resources for many Web applications.



Application servers are typically used in a single or multi-tiered distributed application model in which business applications are divided into components that can then be deployed on different machines. In the examples below, a three-tiered application represents any application distributed over three or more locations, including desktop machines, server machines and database server machines. A Web application can consist of more than three tiers.

- Client (Single-tiered): Client components running on a desktop machine.
- Web tier (or two-tiered): Components running on an application server machine and a desktop.
- Business-tier (multi-tiered): Components running on a desktop, an application server machine and a database server machine.



The J2EE specification and other related specifications define an architecture and interfaces for developing and deploying distributed Internet Java server applications based on a multi-tier architecture. The resulting applications are typically:

- Web-based
- secure
- database-oriented
- transactional
- scalable
- multi-user
- portable

The specifications define both a runtime environment and associated programmer and user information. For a runtime environment, J2EE specifies the execution environment and required system services including transaction, persistence, messaging, and security services. For programmers and users, it



specifies how applications should be developed, deployed and used.

A typical J2EE application has two types of components:

- Web components, or presentation components (Servlets and JavaServer Pages (JSP))
- Enterprise components, or application business logic and data (Enterprise JavaBeans (EJB))

Java applications and applets are Web components that run on the client machine. An applet is an application written in Java that executes in a Web browser. Business applications requiring a richer user interface, such as a graphical user interface created from Swing or the AWT (Abstract Window Toolkit) API, or applications needing HTTP connections often use these components. Java servlet and JSP technology components are Web components that run on the server.

EJB components are business components that run on the server. These components manage the data flow between an application client or applet and components running on the J2EE server or between server components and a database.

A J2EE application server provides containers for hosting Web components (Web containers) and enterprise components (EJB containers). The containers provide component lifecycle management and interface components with services provided by the J2EE server.

### **Where does it fit in the Open Source Architecture?**

In the Summer of 2003 Red Hat announced its long term vision for open source software development in a comprehensive strategy called the Open Source Architecture. The Open Source Architecture (OSA) is designed to meet four basic requirements:

- To create products and services that bring the benefits of standards-based and open source solutions to complete software solutions.
- To stimulate broad participation and collaboration across hardware and software vendors and the open source community.
- To extend the security and reliability of transparent computing as broadly as possible.
- To extend Linux, open source solutions, and ISV partner applications to cost-effectively provide massive horizontal and vertical scalability.

The OSA calls for the provision of a suite of standard technologies on which open source and ISV partner applications

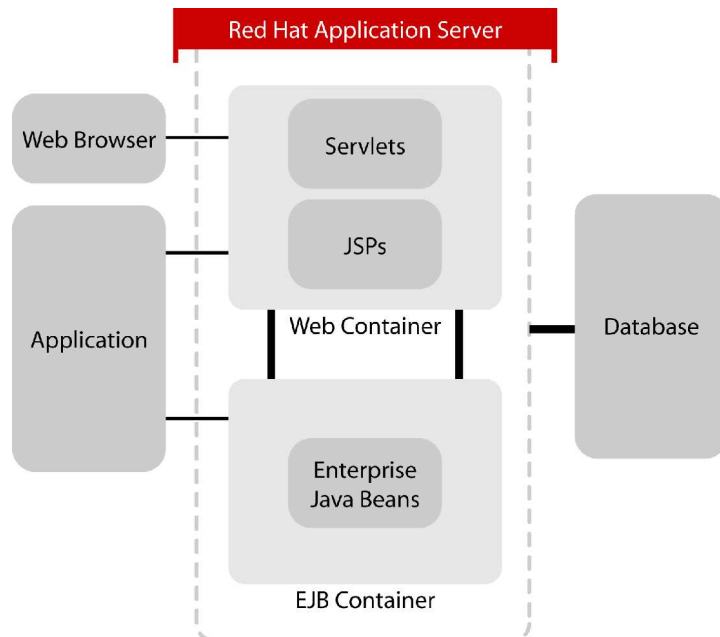


can be deployed in a predictable, stable environment. Essentially, the goal is to extend Red Hat's proven ability to create commercial-quality products based on open source community projects from the operating system to other pieces of the solution.

A core feature of the Open Source Architecture is the delivery of standardized, high-quality, open source middleware that can be used by end-user application providers. The first product designed for this market is Red Hat Application Server.

### What is Red Hat Application Server?

Red Hat Application Server integrates a number of open source technologies from ObjectWeb and Apache projects that have undergone a full product testing cycle. Red Hat Application Server is designed to be a robust platform for the development and deployment of Web applications written in the Java programming language and built with JSP, Servlet, and EJB technologies. This product enables new applications written for the Web to connect to legacy databases, file systems, ERP systems, and other applications through resource adapters. Red Hat Application Server has been built to standard protocols and APIs that have emerged from the various Java, J2EE, Web Services, SOAP (Simple Object Access Protocol), XML (Extensible Markup Language) and CORBA (Common Object Request Broker Architecture) standards groups. Developers build their applications using these standards, while Red Hat's middleware infrastructure ensures compatibility with the guidelines set forth by the J2EE specifications.





The major components of Red Hat Application Server include:

- JOnAS, a standards-based middleware platform
- Tomcat, a Java Servlet and JavaServerPages (JSP) container
- Struts, a framework for building Java Web applications

A Red Hat Application Server subscription also includes support for Apache Web Server, included in Red Hat Enterprise Linux.

Red Hat Application Server requires a Java environment, which can be any of the commercial JVM offerings from IBM, BEA, or Sun Microsystems.

Red Hat Application Server has been tested and is supported with Oracle, DB2, Sybase, PostgreSQL, and MySQL databases. Any JDBC-compliant driver, available from database vendors, can be used.

The following sections describe each of the main components of Red Hat Application Server in more detail.

## JOnAS

The main feature of Red Hat Application Server is JOnAS, ObjectWeb's standards-based middleware implementation<sup>1</sup>, with Web-based administration.

JOnAS can be used as:

- A J2EE server to run applications with both Web and Enterprise components
- An EJB container to run applications with only Enterprise components
- A Web container to run only Web components (Servlets and JSPs)

JOnAS supports the following standards:

J2EE	JTS/JTA	JCA	JMS
EJB	JDBC	JMX	JNDI
JavaMail	Servlet	JSP	
JAF	CMP	JAAS	

---

<sup>1</sup> JOnAS is currently being tested against the J2EE 1.4 compatibility test suite.



## JOnAS Features

JOnAS provides the following important features:

- *Management.*  
JOnAS server management uses JMX and has a servlet-based management console. The management console provides unified JOnAS and Tomcat management.
- *Service-based Architecture.*  
JOnAS's service-based architecture provides for high modularity and configurability of the JOnAS server. This service-based approach provides a way to start only the services needed by a particular application, thus saving valuable system resources. It also allows the developer to apply a component-model approach at the middleware level, and makes the integration of new modules easy. You can manage JOnAS services through JMX.
- *Scalability.*  
JOnAS integrates optimization mechanisms for increasing server scalability. These mechanisms include a pool of stateless session beans, a pool of message-driven beans, a pool of threads, a cache of entity beans, activation/passivation of entity beans, a pool of connections (for JDBC and JMS), and storage access optimizations.
- *Clustering.*  
JOnAS clustering solutions provide load balancing and high availability at the Web and EJB container levels and failover at the Web container level.
- *Distribution.*  
JOnAS works with several distributed processing environments, through the integration of ObjectWeb's CAROL (Common Architecture for RMI ObjectWeb Layer) project (<http://www.objectweb.org/carol/index.html>). CAROL enables simultaneous support of several communication protocols:
  - CMI (Cluster Method Invocation), the cluster-aware distribution protocol of JOnAS
  - RMI (Java Remote Method Invocation) using the Sun proprietary protocol JRMP (Java Remote Method Protocol)
  - RMI on IIOP (Java Remote Method Invocation over Internet Inter-Orb Protocol)
- *Web Services.*  
Using Axis from the Apache Jakarta Project, JOnAS allows J2EE components to access Web services (i.e., to be Web services clients), and allows J2EE components to be deployed as Web services endpoints. Standard Web services



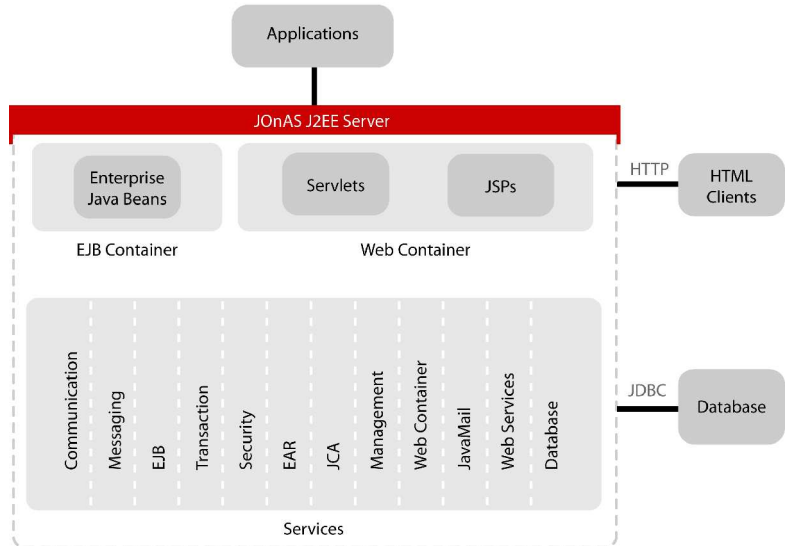
clients and endpoints deployment, as specified in J2EE 1.4, are supported.

- *JDO (Java Data Objects).*  
JOnAS provides the capability of using JDO within J2EE components by integrating the ObjectWeb implementation of JDO, SPEEDO (<http://speedo.objectweb.org>), and its associated J2EE Connector Architecture (CA) Resource Adapter.
- *JCA (J2EE Connector Architecture).*  
JOnAS applications can easily access Enterprise Information Systems (EIS). By supporting the JCA, JOnAS allows deployment of any J2EE CA-compliant Resource Adapter (connector), which makes the corresponding EIS available from J2EE application components. Resource adapters are the standard way to plug JDBC drivers and JMS implementations into J2EE platforms. A JDBC Resource Adapter available with JOnAS provides JDBC PreparedStatement pooling and can be used in place of the JOnAS Database Service. A JORAM JMS Resource adapter is also available.
- *JMS (Java Messaging Service).*  
JMS implementations can be easily plugged into JOnAS. They run as a JOnAS service either in the same JVM (Java Virtual Machine) or in a separate JVM. JOnAS provides administration facilities that hide the JMS proprietary administration APIs. Currently, three JMS implementations can be used:
  - the JORAM open-source JMS implementation from ObjectWeb (<http://joram.objectweb.org/>) and included with Red Hat Application Server
  - SwiftMQ (<http://www.swiftmq.com/>)
  - Websphere MQJ2EE CA Resource Adapters are also available, providing a more standard way to plug JORAM or SwiftMQ into JOnAS.
- *JTA (Java Transaction API).*  
The JOnAS platform supports distributed transactions that involve multiple components and transactional resources through the JOTM (Java Open Transaction Manager) project from ObjectWeb. The JTA transactions support is provided by a Transaction Monitor that has been developed on an implementation of the CORBA Object Transaction Service (OTS).



## JOnAS Architecture

The JOnAS architecture is shown in Figure 4. It shows all of the JOnAS services, although the Transaction Service is the only mandatory service. The others can be included as needed for the Web applications being deployed.



JOnAS services can be managed through the administrative interface Servlet. The services include the following:

- *Registry Service.*  
The Registry Service launches the RMI, CMI, or CosNaming registry, depending on the JOnAS / CAROL configuration. It provides the Java Naming and Directory Interface (JNDI) to application components and other services.
- *EJB Container Service.*  
The EJB Container Service loads EJB components (EJB-JAR files) and their containers. This service can be configured to automatically load a set of EJB-JAR files at server startup time or can load EJB-JAR files on-the-fly at server runtime. JOnAS uses JORM and MEDOR, two ObjectWeb projects, for Container-Managed Persistence of EJB 2.0 (CMP 2).
- *Web Container Service.*  
The Web Container Service runs a Web container (Tomcat) in the JVM of the JOnAS server. This service loads Web components (WAR files) within the Web container. It can be configured to automatically load a set of WAR files at server startup time or can be used to load WAR files on-the-fly at server runtime.
- *EAR Service.*  
The EAR Service deploys complete J2EE applications,



packaged as Enterprise ARchive (EAR) files. It delegates deployment of WAR files to the Web Container Service, EJB-JAR files to the EJB Container Service, and Resource Archives (RAR files) to the Resource Service.

- *Transaction Service.*  
The Transaction Service provides transaction management for EJB components deployed with the EJB Container Service. The Transaction Service uses a Transaction Manager that may be in the local JVM or distributed across multiple JVMs. This service handles distributed transactions. The Transaction Manager, JOTM (Java Open Transaction Manager), an ObjectWeb project, will be enhanced in the near future to provide advanced transaction features such as nested transactions and Web services transactions.
- *Database Service.*  
The Database Service creates and loads Java datasources on the JOnAS server, either at startup or at runtime. Datasources are standard JDBC administrative objects for handling connections to a database. This service also provides database connection pooling. The Database Service can be replaced by the JDBC Resource Adapter.
- *Security Service.*  
The Security Service provides a role-based authorization mechanism for accessing EJB components. JAAS (Java Authentication and Authorization Service) login modules are provided for user authentication of Web container and Java clients.
- *JMS Service.*  
The JMS (Java Message Service) Service launches or establishes a connection to the integrated Message Oriented Middleware (MOM), or JMS, server. The JMS server may or may not be in the same JVM as JOnAS. JOnAS provides wrappers on JMS administrative APIs that allow the application server to perform simple administrative operations automatically. The JMS Service also provides connection pooling and thread pooling for Message-Driven Beans (MDBs). The JMS server integrated with JOnAS is JORAM. Either SwiftMQ or IBM WebSphere MQ can be integrated with JOnAS in place of JORAM. Alternatively, JORAM or SwiftMQ can be used with a JMS resource adapter.
- *Resource Service.*  
The Resource Service is a J2EE CA resource service that deploys J2EE CA-compliant resource adapters (RAR files). Resource adapters connect Enterprise Information Systems (EIS) to the application server, allowing applications to perform business logic on EIS data.



- *JMX Service.*  
The JMX Service is JONAS's management service. This service provides administration of one or more JOnAS servers and Tomcat Web containers from the JOnAS administration console. From the administration console, the administrator can configure services or deploy applications and resources. Either MX4J (included with Red Hat Application Server) or the Sun RI server can be used as the JMX server.
- *Mail Service.*  
The Mail Service provides resources to J2EE applications that use JavaMail.
- *Web Services Service.*  
The Web Services Service supports the deployment of Web services through Axis from the Apache Jakarta project.
- *DB Service.*  
The DB Service is an optional service that can be used to start a Java database server in the same JVM as JOnAS. The Java database included with Red Hat Application Server is currently HSQL.

## Tomcat

Tomcat, from the Apache Jakarta Project, is the Web container in the official Reference Implementation of the Servlet and JavaServerPages standards. JOnAS, as provided in Red Hat Application Server, includes Tomcat as the embedded Web container. A Web container provides Web components (Servlets and JSPs) with access to application server resources in a standard way.

JOnAS provides load balancing, high availability, and failover at the Web container level using the Apache / Tomcat mod\_jk plugin and the Tomcat in-memory session replication mechanism. The plugin uses round-robin and weighted round-robin load balancing algorithms to dispatch HTTP requests from the Apache Web server to multiple Tomcat instances running as JOnAS Web containers.

Red Hat Application Server also provides support for installing Tomcat in a stand-alone configuration without JOnAS.



## Struts

Struts, from the Apache Foundation, provides an open source framework for building Java Web applications. Struts provides a framework for developing applications with architectures based on the Model 2 approach, a variation of the Model-View-Controller (MVC) framework. Core features of Struts include:

- *The MVC framework.*  
Struts provides the Controller component of the MVC framework and integrates with other technologies to provide the Model (e.g., JDBC, EJB) and View (e.g., JSP) components. The Struts controller is a Servlet that intercepts requests and dispatches them to Actions.
- *XML-based configuration.*  
Typical Web applications use an XML document known as a deployment descriptor (web.xml) to initialize resources. Similarly, Struts uses an XML configuration file (struts-config.xml) to initialize its resources, including ActionForms to collect input from users, ActionMappings to direct input to server-side Actions, and ActionForwards to select output pages.
- *Automatic Form Bean Population and Form Repopulation.*  
A main strength of Struts is its form processing capabilities. A Struts form-bean is a JavaBean with properties corresponding to the form's input fields. An application developer creates the form-bean and Struts automatically sets the properties of the form-bean with values from the form input. If a user is returned to a form, due to invalid inputs for example, the input fields of the form are automatically repopulated with the user's previous inputs.
- *Duplicate submission prevention.*  
Struts allows the application developer to easily prevent duplicate form submissions from occurring using a token-based system.
- *Internationalization facilities.*  
Application messages are stored in a Struts resource file. When Struts receives a request, it checks the client's locale and displays the appropriate message text for that locale.

## Apache Web Server

Apache Web Server (or Apache HTTP Server) from the Apache Software Foundation is the most common Web server in use on the Internet. An October 2003 survey found that more than 64% of the Web sites on the Internet were using Apache.

Using the Apache Web Server, Tomcat and the Apache/Tomcat plug-in mod\_jk, also provided with Red Hat Application Server, JOnAS can be configured in a clustered environment to provide load balancing and high availability at the JSP/Servlet level. The mod\_jk plug-in enables the use of the Apache HTTP server in front of one or more Tomcat JSP/Servlet engines, and provides the capability of forwarding some of the HTTP requests (typically those concerning dynamic pages such as JSP and Servlet requests) to Tomcat instances.

This configuration can also use Tomcat's In-Memory-Session-Replication technique based on the group communication protocol JavaGroups to provide failover at the Servlet/JSP level. For more information on configuring Red Hat Application Server in a clustered environment, see the *Red Hat Application Server JOnAS User Guide*.

Apache Web Server 2.0 is included in Red Hat Enterprise Linux. Support for Apache Web Server is included with a Red Hat Application Server subscription.

## Red Hat Developer Suite

Red Hat Developer Suite includes:

- Eclipse framework
- Java Development Tools (JDT) plugin
- C/C++ Development Tools (CDT) plugin
- Plugin Development Environment (PDE)
- RPM/SRPM plugin
- Profiling plugin
- Enhanced documentation
- Lombok plugin

Eclipse provides an environment for developers to efficiently create a wide variety of applications in a rapidly growing set of languages. Eclipse supports a wide range of operating systems beyond Red Hat Enterprise Linux, including Windows (XP, 2000, 98, ME) and Solaris 8. This simplifies the migration of applications and developer skills from these platforms to Red Hat Enterprise Linux.

For more information on Red Hat Developer Suite, see *An Overview of Red Hat Developer Suite*, at <http://www.redhat.com/software/rhel/whitepapers/>.

## Summary

Red Hat Application Server is an open source middleware platform, layered between the operating system and applications. Red Hat Application Server comprises a runtime system and associated development libraries for creating and deploying Java-based Web applications with dynamic content.

Red Hat Application Server includes:

- JOnAS, a standards-based middleware platform
- Tomcat, a Java Servlet and JavaServerPages (JSP) container
- Struts, a framework for building Java Web applications

A Red Hat Application Server subscription also includes support for Apache Web Server, included in Red Hat Enterprise Linux, and Red Hat Developer Suite, an Eclipse-based IDE.

## For more information visit

[www.redhat.com/software/rha/appserver/](http://www.redhat.com/software/rha/appserver/)

## Contact information:

U.S. and Canada:  
1-888-REDHAT1  
(1-888-733-4281)  
+1-919-754-3700

Worldwide:  
[www.redhat.com/about/corporate/wwoffices/](http://www.redhat.com/about/corporate/wwoffices/)