

Overview of JBoss Application Server

JBoss Organization

- JBoss - The Professional Open Source Company
- Focuses on middleware software and services - JBoss Enterprise Middleware Suite (JEMS)
- Software is open source and free
- Acquired by Red Hat in April 2006 for \$420M

JBoss Enterprise Middleware Suite (JEMS):

- Application Server (JBoss AS, Tomcat)
- O/R Mapping and Persistence (Hibernate)
- Portal Platform (JBoss Portal)
- Business Process Management and Rules (JBoss jBPM, JBoss Rules)
- Object/Data Cache (JBoss Cache)
- Distributed Transaction Management (JBoss Transactions)
- Development Tools (JBoss Tools plugin for Eclipse)

JBoss AS Background

- Established in 1999 as an open-source EJB container
- Version 2.x becomes a full J2EE application server
- Version 3.x is the production series - based on JMX microkernel and service-oriented architecture (May 2002)
- Version 4.x explores the aspect-oriented middleware services, adds support for EJB 3 (September 2004)
- Version 5.x comes with JBoss Microcontainer (also a stand-alone project) and replaces JMX Microkernel of the 3.x and 4.x JBoss series (December 2008)
- Version 6.x to add support for Java EE 6 APIs and Profiles. Replaces JBoss Messaging with HornetQ (6.0.0.M3 released in April 2010)

Highlights of JBoss AS

- Enterprise-class reliability, performance, scalability, and high-availability
- Zero-cost product license, to download, use, embed, and distribute
- Open-source
- Built for standards - provides a safe choice for portable applications (interoperable)

- Service-oriented architecture provides consistency, makes it embeddable
- Aspect-oriented architecture simplifies interaction with middleware services
- 24x7 professional support by the core development team

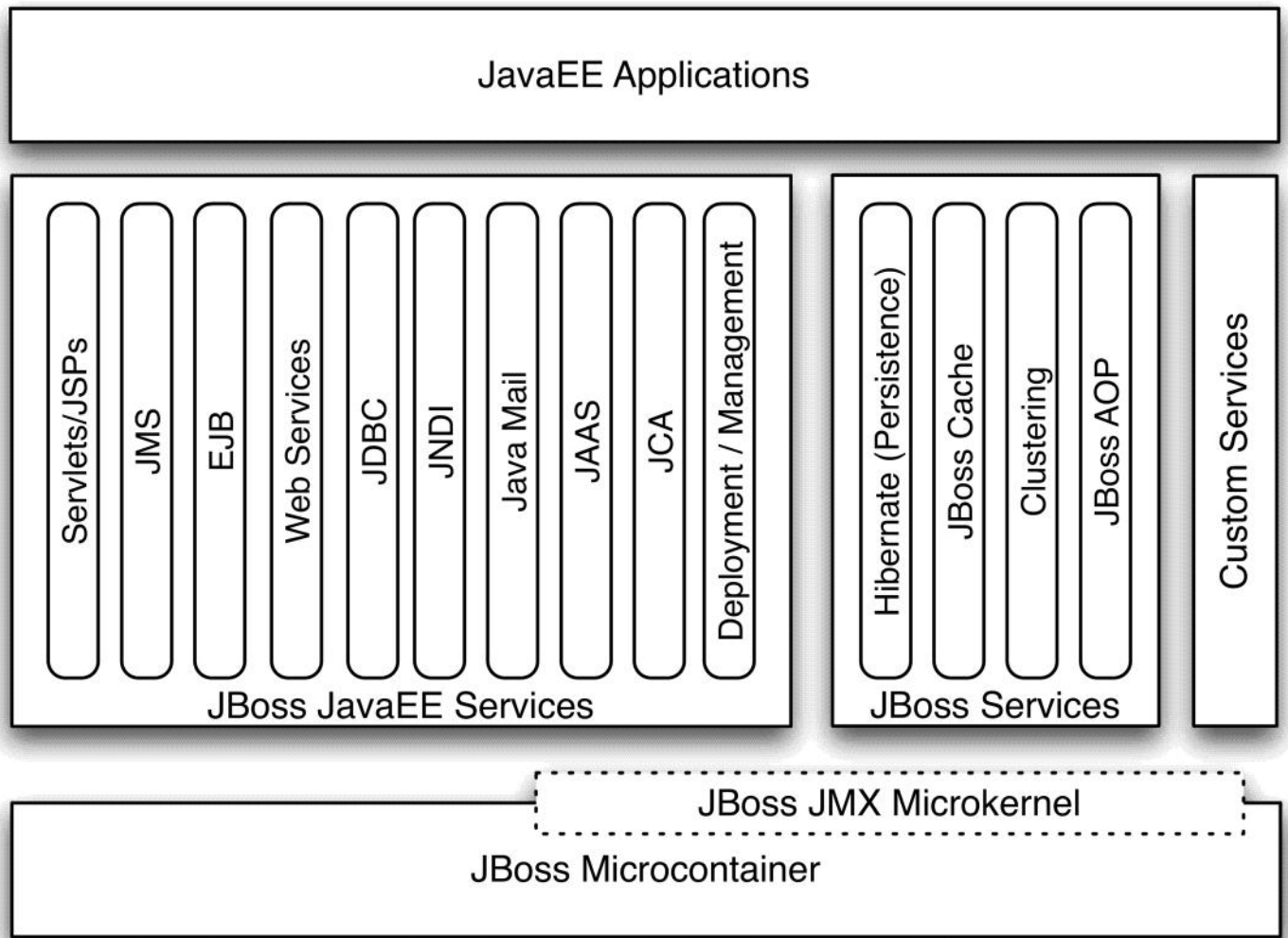
What is new in JBoss AS 5

- New kernel ⇒ JBoss Microcontainer
 - is a refactoring of old JMX Microkernel (JBoss AS 3.x and 4.x)
 - the core of JBoss AS 5
- New messaging provider ⇒ JBoss Messaging
- One of the first application servers to implement EJB 3.0 specification
- Reliable transaction manager ⇒ JBoss TS
- JBoss Web based on Apache Tomcat 6.0
- JBoss WS 3.0 (support for JAX-WS/JAX-RPC)
- Two new configurations:
 - standard: Java EE compliant configuration.
 - web: provides support for JTA/JCA and JPA in addition to the Servlet/JSP container. The server can only be accessed through the http port.

JBoss AS Architecture

"JBoss AS is assembled from a set of independent, yet cooperating components and services that are neatly packaged and fully hot-deployable. It is architected to be seamlessly embeddable in applications, and the nature of its embedding is completely customizable to the requirements of the application itself. Only the critical and necessary application server components, therefore, need to be brought along as part of the application's baseline footprint. Developers can also easily create and add their own services to the system, thus ensuring that custom services exhibit the same consistent behavior as the JBoss standard set of services."

JBoss AS 5.x Architecture



JBoss Microcontainer Layer

JBoss Microcontainer is an inversion of control (IoC) framework. The objects created usually represent the modules of your application.

- Replaces JMX-based Microkernel, though still supports all JMX Microkernel features
- IoC framework similar to Spring IoC
- POJO based kernel (no need for Standard/XMBean or MBeanProxy)
- Transparent AOP integration
- Virtual File System (VFS)
- class-loading

Services Layer

- Service-oriented architecture - service is either defined as a POJO or a JMX Managed Bean (use the JMX kernel, still available in JBoss 5.x but is created by JBoss Microcontainer).
- Services are hot-pluggable
- Makes it possible to tune the system for just the required services to lower the overall footprint (easier to secure and tune)
- Easy to define new services and package them as SARs (service archives) or JARs (Java ARchives)
- Examples: Servlet/JSP container, EJB container, transaction management, messaging, connection pooling, security etc.

Aspect Layer

Increase the modularity of an application by allowing the separation of cross-cutting concerns (e.g logging is often required in many parts of your application).

- Based on aspect-oriented programming model (AOP)
- Defines cross-cutting simple-to-use services
- Makes it possible to add object persistence, caching, replication, remoteness, security, etc. late in the development cycle by annotating existing plain-old-java-objects (a.k.a POJOs)

Application Layer

- This is where Java EE applications reside
- This layer deals with the business logic while leaving the container services up to JBoss AS
- Portable - Independent of JBoss AS

JBoss AS Services

- JBoss Microcontainer - POJOs services container
- JBoss Microkernel - JMX MBean server (One of the primary POJOs created by JBoss Microcontainer)
- Aspect-oriented Framework
- Web Application Services - based on Tomcat (Servlet, JSP, JSF)
- Enterprise Services: EJB, ORB, JNDI, JTA

- Web Services - based on SOAP, WSDL, UDDI, and XML
- Messaging Services: JMS, JDBC, JCA
- Persistence Services - Hibernate O/R mapping and transparent persistence
- HA Services: clustering, fail-over, load-balancing, distributed deployments
- Security Services - based on JAAS
- Console Services - monitoring, configuration, deployment, management, lifecycle

. JBoss AS Requirements

| | JBoss 3.2.x | JBoss 4.x | JBoss 5.x |
|---------------|--------------------|--------------------|--------------------|
| Java (JDK) | 1.3+ | 1.4-1.5+ | 1.5+ |
| OS/Platform | Any Java-compliant | Any Java-compliant | Any Java-compliant |
| Memory - RAM | 128MB | 512MB | 512MB |
| CPU | 266MHz | 400MHz | 400MHz |
| Disk | 100MB | 100MB | 100MB |
| DB (Optional) | Any JDBC-Compliant | Any JDBC-Compliant | Any JDBC-Compliant |