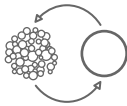


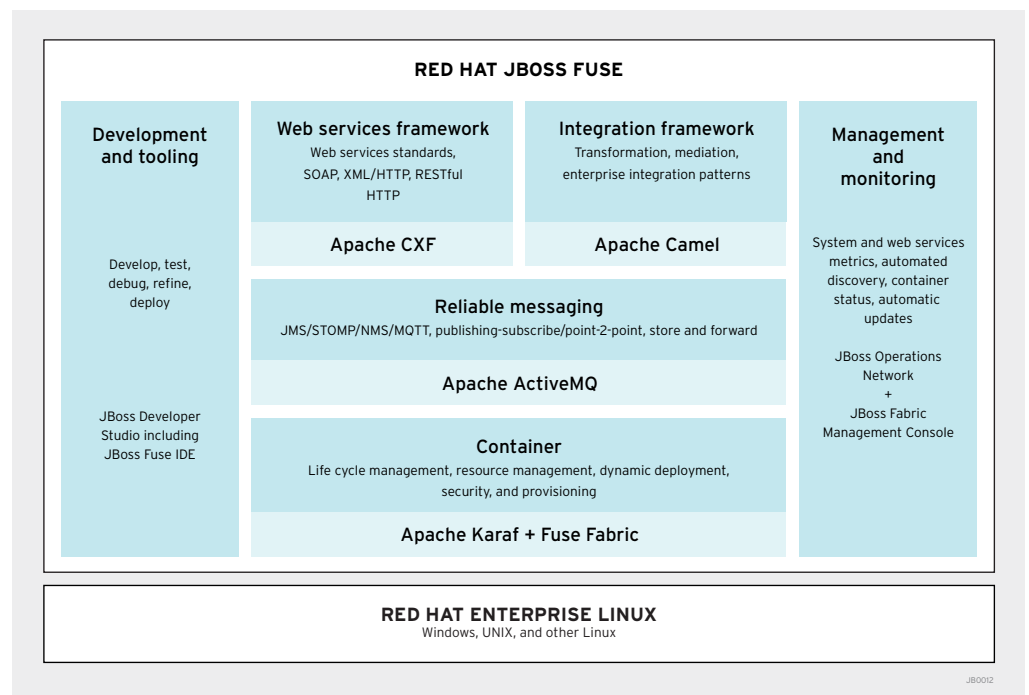
# RED HAT JBOSS FUSE

## TECHNOLOGY OVERVIEW



A SMALL-FOOTPRINT,  
FLEXIBLE,  
OPEN SOURCE ESB

Red Hat® JBoss® Fuse is a comprehensive, standards-based integration platform that can be configured with any combination of components for a customizable IT footprint.



### THE FUNCTIONAL COMPONENTS OF RED HAT JBOSS FUSE INCLUDE:

- **Container** - You can create your choice of containers at each endpoint. The container layer uses Apache Karaf and is enhanced by Fabric Management Console, which simplifies the management of large numbers of distributed containers.
- **Integration framework** - You can use a standard notation to go from diagram to implementation with coding. This layer is based on Apache Camel.
- **Web service framework** - You can turn any application into a service for inclusion in your architecture. Service enablement technology is based on Apache CXF.
- **Reliable messaging** - Red Hat JBoss Fuse includes a small-footprint, standards-based message broker. The messaging layer is based on Apache ActiveMQ.
- **Development and tooling** - Red Hat JBoss Fuse is supported by tooling to help you with development. JBoss Developer Studio includes Fuse IDE.
- **Management and monitoring** - Production environments are supported by Fabric Management Console for management and JBoss Operations Network for monitoring



facebook.com/redhatinc  
@redhatnews  
linkedin.com/company/red-hat

redhat.com

## KEY FEATURES AND BENEFITS

### CONTAINER LAYER

FEATURE	BENEFIT
<b>Dynamic configuration</b> Change the configuration while the container is running	<b>Increased system availability</b> Change the configuration at an endpoint without having to stop and restart the ESB
<b>Hot deployment</b> Deploy or update services while the ESB is running	<b>Increased system availability</b> Change an integration route without affecting other services or endpoints
<b>Custom deployers</b> Deploy POJOs as dynamic services (Blueprint, Spring DM)	<b>Reduced development time</b> Faster and easier development of services without the complexity of creating OSGi bundles
<b>Centralized logging backend</b> Logging through multiple common logging APIs (SLF4J, JCL, Avalon, Tomcat, OSGi)	<b>Reduced development and maintenance costs</b> Reduces the need to refactor services written for a particular logging API when deploying in the ESB

## CONTAINER LAYER (CONTINUED)

### Extensible shell console

Use to manage runtime and control services' life cycle; can be dynamically extended to control custom features or functions of a deployed service

### Better control over services

Gives interactive control of deployed services and features; shell extensions provide additional control options, eliminating the need to write a custom console

### Remote access

Secure access to the ESB runtime console from any SSH client

### Simplified administration of large applications

Location-independent management of the ESB

### Security framework

Access control to the ESB through JAAS, SSL encryption, and plug-in points to support custom and third-party authentication providers, firewalls, proxy servers, HTTP(s) tunneling, and DMZ products

### Simplified security administration

Uses a single security framework

### Clustering and failover

Shares loads across brokers and containers in a cluster; failover is supported through multiple master-slave configuration options

### Increased system availability

Enables deployments to scale to support large numbers of messages, users, and applications, with high performance and high availability

## INTEGRATION LAYER

FEATURE	BENEFIT
<b>Enterprise integration router</b> Leverages Apache Camel to provide a full-featured, easy-to-use, and intuitive framework for integration	<b>Go from diagram to deployment</b> Makes it easy to prototype and test enterprise integration patterns in a fluent Java DSL or through IoC using Spring-based deployments
<b>Web services</b> Easy-to-use and intuitive JAX-WS-compliant web services stack	<b>Reduces development time</b> WSDL-first or Java-first creation of web services
<b>RESTful services</b> Easy-to-use and intuitive JAX-RS front end	<b>Reduces development time</b> Simple java-first development of RESTful services
<b>JMS service</b> Full-featured JMS 1.1-compliant broker and client infrastructure	<b>Integrates with existing IT infrastructure</b> Supports asynchronous communication between services within the ESB or from outside the ESB
<b>Extensive connectivity</b> Uses Apache Camel to provide connectivity to external applications with connectors for JDBC, FTP/SFTP, HTTP/HTTPS, file, and many more	<b>Broader integration</b> Simplified integration with many diverse sources and targets

## MESSAGE BROKER

FEATURE	BENEFIT
<b>Cross-language clients</b> Providing connectivity from client programs written in languages other than Java	<b>Supports many development environments</b> Allows native connectivity from applications written in non-Java languages like C or C++
<b>Pluggable transports</b> Multiple transport protocols for exchanging data between the broker and client, or between multiple brokers	<b>Supports many networking environment</b> Flexibility to meet the demands of different networking environments and use cases
<b>Flexible persistence</b> Supports a variety of persistence options, from no persistence at all, to using a JDBC database directly, to enabling high-performance persistence using the journal plus a JDBC database; long-term persistence is enabled through a JDBC-compliant storage database	<b>Balances reliability and performance</b> Allows the user to maximize reliability and performance for individual applications
<b>REST API</b> A technology-neutral, web-based API to the message broker service	<b>Simplified integration</b> Easy integration with RESTful web services
<b>Ajax support</b> Support for streaming to web browsers using pure DHTML	<b>Increased integration options</b> Allows web developers to use the browser as a messaging client
<b>JMS Streams for very large messages</b> When sending extremely large messages, JMS streams eliminate the bottleneck that would occur as the JMS client tries to keep an entire 1+ GB message in memory	<b>Supports application scalability</b> Allows the messaging platform to deliver truly massive files (many GBs) across the network in a reliable manner
<b>Message compression</b> GZIP compression allows highly verbose messages to be compressed	<b>Supports application scalability</b> Message compression helps organizations efficiently transport large amounts of data encapsulated in SOAP and other XML formats

## MANAGEMENT

FEATURE	BENEFIT
<b>Cluster configuration manager</b> Mechanism for provision and configuration of ESB nodes of ESB in a cluster of ESBs	<b>Simplifies management of multiple nodes</b> Centralized configuration and management of ESB cluster nodes
<b>Configuration profiles</b> Method of defining the configuration of an ESB node; changes and updates to profiles are applied across all nodes	<b>Simplifies management of uniquely configured nodes</b> Guaranteed consistency between identically configured nodes and simplified maintenance
<b>Runtime registry</b> Provides information about ESB instances, including the services they host, the location of service endpoints, and the status of all services and endpoints	<b>Supports node auto-discovery</b> Services or consumers can discover services available anywhere in the cluster without needing to know specific location information

## RED HAT JBOSS FUSE - INTEGRATION BEYOND THE DATACENTER

Some integration solutions require comprehensive integration capabilities, while others need a lightweight, low-footprint, easy-to-manage integration platform. Some integration solutions require both. Red Hat JBoss Fuse can be deployed and easily managed in any configuration. This way, you can have a different configuration for every endpoint, and can deploy a network of configurations across your infrastructure – whether it is on premise, in the cloud, or in a hybrid configuration.

With this cost-effective and flexible integration platform, businesses can finally have integration everywhere.

Visit us to:

- **Download Red Hat JBoss Middleware products** – Our products are open source. Install them, run a demo, develop a proof-of-concept project. Eliminate risk by trying the software before you buy.  
[redhat.com/jbossfuse/download](https://redhat.com/jbossfuse/download)
- **Learn how to make the most of Red Hat JBoss Middleware products** – Explore the various resources and get up to speed fast. View a webinar, tune in for a Getting Started tutorial, watch a demo, and more.  
[redhat.com/products/jbossenterprisemiddleware/fuse](https://redhat.com/products/jbossenterprisemiddleware/fuse)
- **Get involved** – Open source software belongs to you. Blog, join a user group, contribute code, or test upcoming releases.  
[jboss.org/contribute](https://jboss.org/contribute) and [apache.org](https://apache.org)



## ABOUT RED HAT

Red Hat is the world's leading provider of open source solutions, using a community-powered approach to provide reliable and high-performing cloud, virtualization, storage, Linux, and middleware technologies. Red Hat also offers award-winning support, training, and consulting services. Red Hat is an S&P company with more than 70 offices spanning the globe, empowering its customers' businesses.



facebook.com/redhatinc  
@redhatnews  
linkedin.com/company/red-hat

**NORTH AMERICA**  
1-888-REDHAT1

**EUROPE, MIDDLE EAST  
AND AFRICA**  
00800 7334 2835  
europe@redhat.com

**ASIA PACIFIC**  
+65 6490 4200  
apac@redhat.com

**LATIN AMERICA**  
+54 11 4329 7300  
latammktg@redhat.com