

# JBoss ESB vs. Oracle Service Bus Shoot out

Steffen Miller  
Oracle Deutschland B.V. & Co.KG  
Geschäftsstelle Frankfurt

## Schlüsselworte

ESB, Enterprise Service Bus, Switchyard, JBoss, OSB, Integration.

## Einleitung

Der Oracle Service Bus stellt eine zentrale Komponente im Portfolio von Oracles SOA Suite dar. Auch in Projekten, die „nur“ einen Bus brauchen findet er Verwendung, und trifft dabei auf Marktbegleiter aus dem OpenSource Umfeld wie den JBossESB. Nachfolgend wird in verschiedenen Bereichen untersucht, welche Unterschied sich in der Architektur, Installation und Out-Of-the-Box Experience inkl. Beispielen bestehen.

## Architektur

Bevor auf die spezifischeren Fähigkeiten der beiden ESBs eingegangen wird, beleuchten wir die grundlegenden Architekturen. Diese unterscheiden sich im wesentlichen durch die Wahl der Realisierung der Services innerhalb des Produktes.

## Oracle Service Bus

Der OSB setzt dabei auf eine Architektur, die auf dem Weblogic Server als Applikations Server basiert und den ESB als J2EE Applikation darauf deployed. Nachfolgende Abbildung gibt einen Überblick über die verschiedenen realisierten Funktionsbereiche, die für den Betrieb eines ESB notwendig sind.

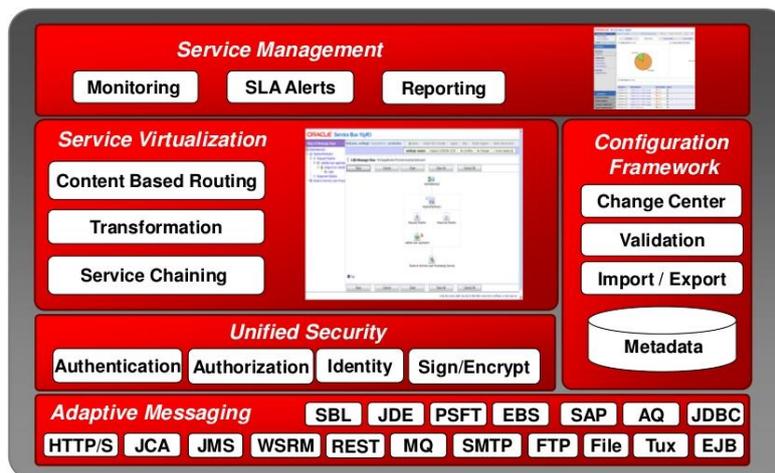


Abb. 1

Die Services werden entweder über die Webapplikation direkt im laufenden OSB definiert, oder mit Hilfe der IDE auf Eclipse Basis. Nachfolgende Abbildung( Abb. 2 )gibt einen Überblick über den Aufbau, bzw. die Elemente der Services. Services werden dabei mit Hilfe vordefinierter Aktionen

(Actions) in Stages konfiguriert. Dabei gibt es für den Request-Weg einen Request-Pfad in der Pipeline des Message Flows und einen Response-Weg für die Antwort. Für den Fehlerfall existieren auf verschiedenen Ebenen (Stage, Pipeline, Service) separate Error-Handler, die wiederum mit Stages strukturiert und Aktionen konfiguriert werden.

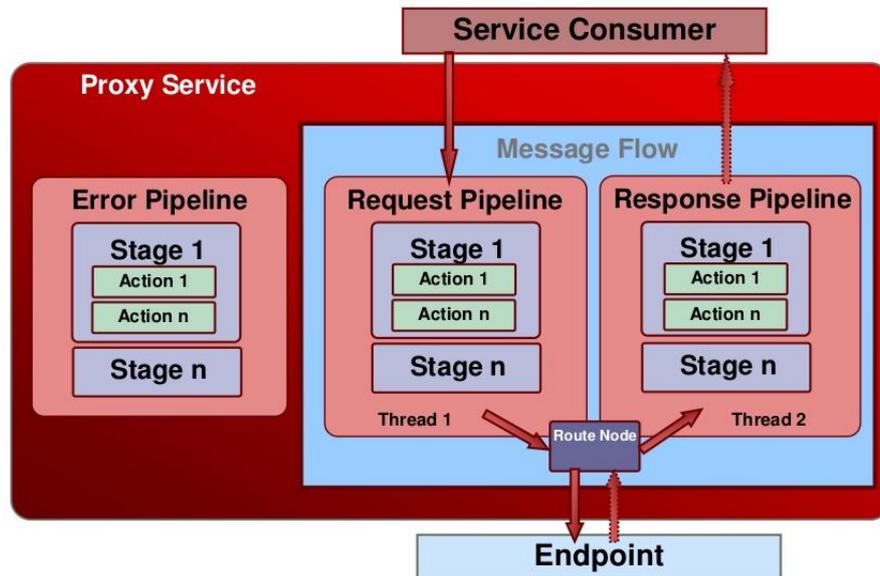


Abb. 2

Wie schon erwähnt liegt der Hauptfokus auf der Konfiguration von Services auf Basis vorhandener Aktionen wie Replace, Assign, Delete und Kontrollstrukturen/Schleifen wie for-each, if-then etc. Zusätzlich sind Aktionen für den Aufruf weiterer Services (Service Chaining) und Split-Join vorhanden. Fehlende Funktionen können diese auch über Java-callouts eingebunden werden.

## JBOSS ESB

Auch der JBoss ESB basiert als Deployment auf einem Applikationsserver, hier natürlich dem JBoss AS. Im Standardumfang ist auch hier eine Administrationsoberfläche enthalten, die einen Überblick über die deployten Services gibt. ESB Services werden beim JBoss ESB als .esb Archive, und JBoss Services als .sar Archive deployed. D.h. die Services werden mit z.B. der Eclipse IDE entwickelt, paketiert und deployed.

Nachfolgende Grafik erläutert den Zusammenhang der verschiedenen ESB Komponenten, wie Passive- und Active Listener aber auch die Gruppierung von Aktionen in sogenannte Processing Pipelines.

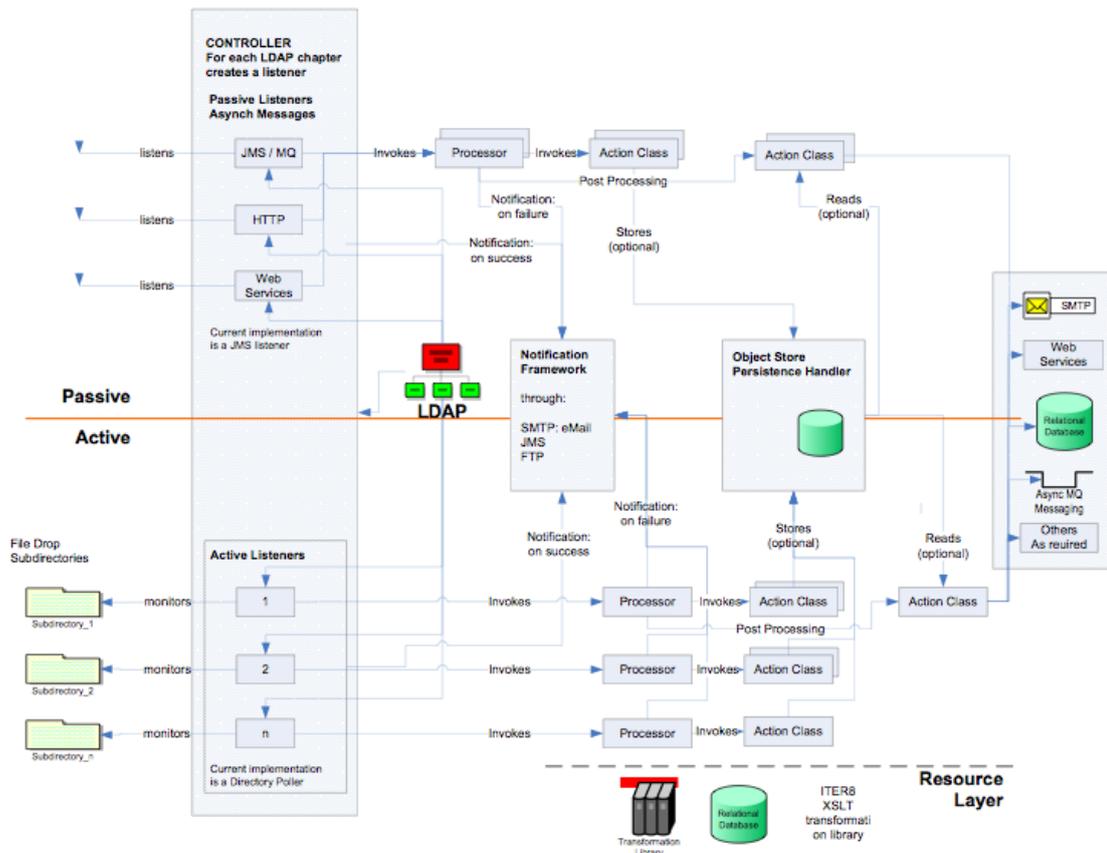


Abb. 3

D.h. die Begrifflichkeit beider Produkte ist ähnlich, der Hauptunterschied, der hier deutlich wird, liegt in der Art der Definition von Services. Hier verwendet der JBoss ESB den Ansatz, die Services in einer XML Datei (jboss-esb.xml) zu definieren und schwerpunkt-mässig Java-Klassen für die eigentlichen Funktionalitäten einzubinden. Zentrale Bestandteile sind

- MessageListener, die Nachrichten entgegennehmen und an
- Message Processing pipelines weiter reichen. Diese können
- Daten Transformationen auf Basis von Aktionen vornehmen.
- Aktionen liegen zum Teil als Produktbestandteil (Http-Router, Smooks-action etc.) vor, bzw. können/müssen per Java-Code erstellt werden.

## Installation

### Oracle Service Bus

Die Installation besteht aus 2 Paketen, dem WebLogic Server (WLS) inkl. Oracle Enterprise Pack for Eclipse (OEPE) und dem Oracle Service Bus (<http://bit.ly/UoAnFk>). Zuerst muss der WLS installiert

werden, dieser legt ein sogenanntes „Middleware-Home“ an, mit einem frei wählbaren Namen wie „OSB\_11.1.1.6“ o.ä.. Anschliessend wird in dieses Middleware-Home das OSB Paket installiert. Nach Abschluss der Installation wurde eine Verzeichnisstruktur nach **Error! Reference source not found.** erstellt.

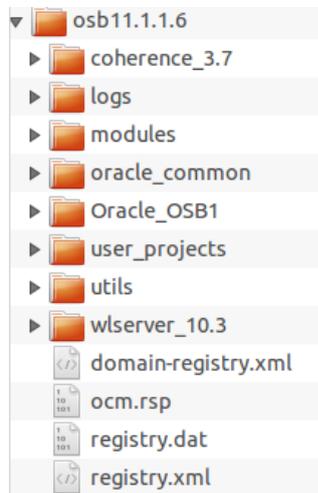


Abb. 4

## JBOSS ESB

Die Software kann als zip-Paket von <https://access.redhat.com/downloads> geladen werden. Zuvor muss man einen Account erstellen und erhält dann die Möglichkeit eine 30-Tage Testversion zu erhalten. Ein alternativer Weg (ohne Account) geht über <http://www.jboss.org/jbossesb/downloads> und dort z.B. die jbossesb-server-4.11.zip laden. Nach dem Entpacken sieht die Struktur so aus.

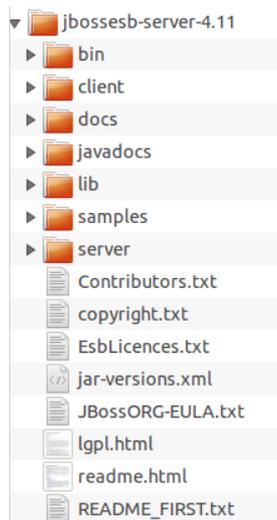


Abb. 5

Allerdings ist dies eine ältere Version, und verwendet JBoss AS 4.2.3. Möchte man einen aktuelleren JBoss verwenden, wählt man jbossesb-4.11.zip, dann muss allerdings zuvor ein „JBoss AS z.B. 6.0 final“ zuvor installiert werden. Der JBoss ESB wird anschliessend in den JBoss AS installiert.

Als IDE kommt Eclipse zum Einsatz. Auch hier gibt es zwei Möglichkeiten die benötigten Paket zu erhalten. Einmal geht man den Weg über ein Eclipse von eclipse.org und installiert die benötigten Pakete separat nach. Oder man wählt das JBoss Developer Studio (JBDS) , das man nach einer kostenlosen Registrierung herunterladen kann (<http://bit.ly/Q0mAR0>). In beiden Fällen fehlt der ESB Editor, der bisher nur in einer älteren Version (Eclipse 3.6.2) released wurde, und für die aktuellen Eclipse Versionen nur in einer beta zur Verfügung steht.

## **Wertung**

Der grösste Unterschied bei der Installation betrifft die Bereitstellung der „Installationsmedien“. Bei Oracle werden Executables geliefert, die mit einem grafischen Wizard durch den Installationsprozess führen.

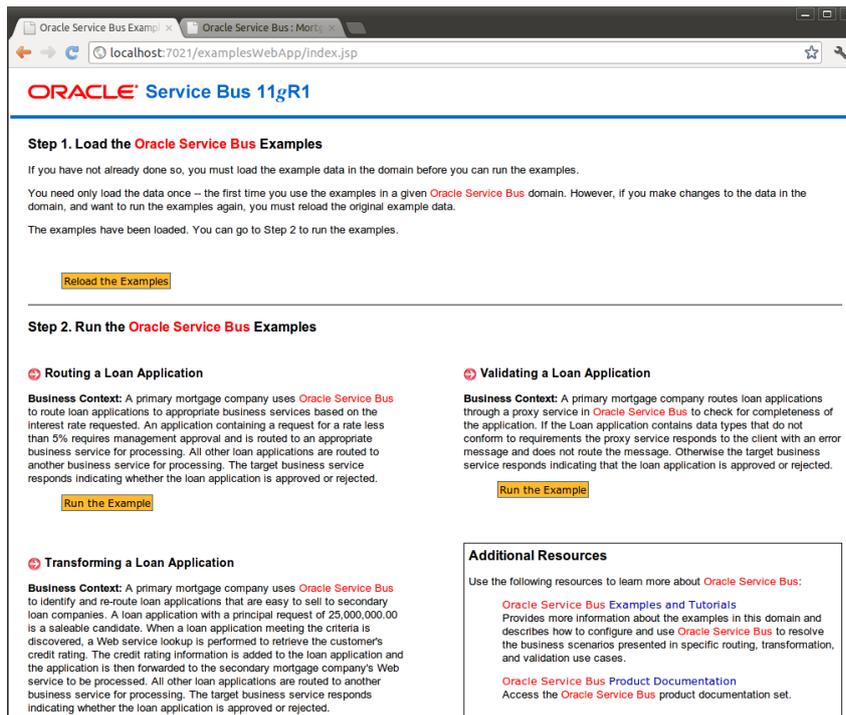
Bei JBoss Esb spaltet sich die „Lieferung“ in verschiedenen „Qualitäten“ auf. Geht man den „Distributionsweg“, d.h. man kauft eine Subscription, oder lädt eine 30 Tage Version der JBoss EAP, erhält man eine Zip Datei, die nur entpackt wird.

Wählt man allerdings den OpenSource-Weg hat man wieder mehrere Möglichkeiten, eine „fertig“ Version (jbossesb-server-4.11.zip), die allerdings veraltet (JBoss AS 4.2.3) ist und auch nicht alle Module enthält, z.B. fehlt die ESB Admin-Console. Oder man wählt die Server Variante (jbossesb-4.11.zip), dann muss allerdings zusätzlich noch ein JBoss AS (5 od. 6) installiert werden. In diesen „separaten“ AS wird dann über eine Property-Datei der ESB installiert. Das gleiche Vorgehen trifft auch auf die Verwendung von Eclipse als IDE zu.

## **Beispiele / OOTB Experience**

### **Oracle Service Bus**

Der OSB kommt mit einem Beispielservers, der während der Installation als zusätzliches Feature ausgewählt werden kann. Dieser befindet sich anschliessend unter `$MiddlewareHome/user_projects/samples/domains/servicebus` und lässt sich mit dem Script `startWebLogic.sh` starten. Das Script startet nach Abschluss des Bootvorgangs automatisch einen Browser mit einer Beispielapplikation (Abb. 6) .



**Abb. 6**

Die Console zur Bearbeitung der Beispiel und zur Konfiguration des Oracle Service Buses ruft man über <http://localhost:7021/sbconsole> auf, Username und Passwort sind weblogic/welcome1 .

## **JBoss ESB**

Auch der JBoss ESB kommt mit einem vorkonfigurierten Beispielservers und dieser wird mit `jbosessb-server-4.11/bin/run.sh` gestartet. Die mitgelieferten Beispiele können nun über ant Commandos aus dem Verzeichnis `jbosessb-server-4.11/samples/quickstarts` deployed und gestartet werden.

## **Wertung**

In beiden Fällen sind zahlreiche Beispiel für einen Beispiel-Server mitgeliefert. Im Oracle Service Bus ist dies eleganter über eine Weboberfläche realisiert. Hier werden alle Beispiele mit einem Knopfdruck installiert und stehen zur Dateieingabe per Webfrontend zur Verfügung. Die Beispiel beim JBoss ESB sind nur über ant Kommandos verwendbar, dafür aber zahlreicher.

Bei der OOTB –Experience hat in der JBoss ESB „CommunityEdition“, Oracle die Nase vorn, da sich hier ein schnellerer Start und Überblick bzw. Einblick in die Beispiel erreichen lässt. Hier spielt die Weboberfläche zur Konfiguration, Änderung und Erstellung von Services ihre Stärken aus, und ermöglicht einen schnelle Ein-und Überblick über die Service Bus Funktionalitäten.

In der Subscription Variante liegen beide gleich auf, wobei der JBoss ESB durch die konsequente Verwendung von Konfigurationsdateien und Verzicht auf eine grafische Entwicklungsumgebung Punkte liegen lässt. Es existiert lediglich ein rudimentäres Frontend für die `jboss-esb.xml` Datei. So besteht ein einfacher HelloWorld Service aus 12 Dateien, darunter 3 Java-Dateien, der Rest xml Konfigurationsdateien.

**Kontaktadresse:**

Steffen Miller

Oracle Deutschland B.V.&Co.KG  
Geschäftsstelle Frankfurt  
Robert-Bosch Strasse 5  
D-63303 Dreieich

Telefon: +49 (0) 6103-397 778  
E-Mail [steffen.miller@oracle.com](mailto:steffen.miller@oracle.com)  
Internet: [www.oracle.com/de](http://www.oracle.com/de)