

Dreikampf: Eine Testfahrt mit Glassfish, JBoss und Geronimo

Autor(en): Frank Pientka, MATERNA GmbH

Ende des letzten Jahres wurden einige Applikationsserver für Java EE 6 zertifiziert. Den neuen Flaggschiffen steht ihre Bewährungsprobe jedoch noch bevor. Ein guter Zeitpunkt sich Glassfish, JBoss und Geronimo näher anzuschauen.

Java EE 6

Für Java EE 6 gilt der Satz, was lange währt, wird endlich gut. Die lange Wartezeit zwischen Veröffentlichung der Spezifikation und der Referenzimplementierung vor mehr als 2 Jahren haben die meisten Applikationsserver-Anbieter genutzt, um ihre Produkte auf die Anforderungen der Zukunft auszurichten und dazu komplett umzubauen.

Dieser Wechsel stellt nicht nur für die Hersteller, sondern auch für die Nutzer einige Herausforderungen parat. Doch der Umstieg auf Java EE 6 lohnt sich. Alle drei hier vorgestellten Applikationsserver verwenden intern OSGi, damit wird nicht nur der Server schlanker, sondern es lassen sich flexiblere Anwendungen damit entwickeln. Ein weiterer gemeinsamer Punkt ist, dass die Administrierbarkeit und die Integration in bestehende System- und Entwicklungs-Umgebungen verbessert wurden. Gerade Betriebsthemen, wie Überwachung, Skalierbarkeit und Automatisierbarkeit spielen hier eine große Rolle.

GlassFish – die Referenzimplementierung

Bei Glassfish handelt es sich um die Java EE-Referenzimplementierung. Kein Wunder, dass er immer die aktuellsten Standards unterstützt.

Die Zukunft von Glassfish war nach der Übernahme von SUN durch ORACLE etwas ungewiss. Doch spätestens nach der Vorstellung von GlassFish 3.1.1 im Februar 2011 hat sich die Aufregung gelegt, da dieser endlich die lang ersehnte Clusterfähigkeit enthält. Genau ein Jahr später erschien die aktualisierte und fehlerbereinigte Version 3.1.2 des GlassFishs. Neben der OpenSource-Variante bietet Oracle eine gleichnamige kommerzielle Variante mit Support an. Die Hoffnung, dass im GlassFish neben Java, andere Java-Skriptsprachen ausgeführt werden hat sich nicht erfüllt und wurde in den aktuellen Versionen nicht mehr unterstützt.

Für die kommende Java EE 7 Versionen existiert bereits eine erste Implementierung mit dem GlassFish 4.0. Doch bevor Java EE 7 erscheint wird es noch eine Version

3.2 geben. Der GlassFish 3.1 wird sicher für die nächsten Jahre die am meisten eingesetzte GlassFish-Version bleiben, da im Bundle mit NetBeans oder dem Java EE 6 SDK regelmäßige Aktualisierungen geliefert werden. Ebenso gibt es Erweiterungen, wie das Oracle Enterprise Pack für Eclipse, Integrationen in die Fusion Produktreihe oder Migrationsassistenten für WebLogic. Gerade die OpenSource-Variante ist bei vielen Neu- und Altkunden von ORACLE beliebt.

JBoss – der Platzhirsch

JBoss ist der älteste OpenSource-Applikationsserver. Dadurch ist er am Markt weit verbreitet. Technologisch ist er in den letzten Jahren etwas ins Hintertreffen geraten. Das hat sich mit der Zwischenversion 6 und mit der gerade frisch zertifizierten Version 7.1 geändert. Neben einer modernisierten Administrationsoberfläche, wurde vor allem unter der Haube mit OSGi einiges geändert. Neben Felix, Tomcat und Aries werden hier einige Apache Komponenten verwendet. Dadurch dass die CDI-Referenz-Implementierung Weld an Apache übergeben wurde und das eigene WebService-Framework zugunsten von Apache CXF aufgegeben wurde, scheint es bei JBoss ein Umdenken hinsichtlich der Zusammenarbeit mit Apache zu geben. Gerade mit den weiteren JBoss-Produkten, wie Hibernate, SEAM, RichFaces, Jgroups, Infinispan, DROOLS, jBPM, HornetQ oder GatIn integriert sich JBoss AS gut, wenn man die Versionsabhängigkeiten beachtet. Hier bietet JBoss mit seiner Mutter Red Hat einen bunten Strauß an Produkten an. Die Administrierbarkeit und Überwachung von großen JBoss-Installationen ist mit RHQ möglich. Die Entwicklung wird durch Werkzeuge, wie die Eclipse JBoss Tools oder Testframeworks, wie JSFUnit oder Arquillian, gut unterstützt. Die am meisten eingesetzte Version ist 5.1. Ein Versionswechsel auf 7.1 ist nicht ohne Probleme. Gerade da die EJB-Abwärtskompatibilität erst recht spät eingebaut wurde, muss hier mit Überraschungen gerechnet werden. Eine Einschränkung der OpenSource-Varianten von JBoss bleibt, dass die Entwicklung nicht wirklich öffentlich passiert und dass Anwender keine Fehlerkorrekturen für alte Produkte erhalten. Etwas kurios mutet die große Lizenzsammlung mit 22 Lizenzen von GPL bis MIT an. Trotzdem zählt RedHat zu den großen OpenSource-Namen, die sich im Java Community Prozess an der Weiterentwicklung von Java EE beteiligen. Fehlerkorrekturen werden immer nur über die aktuellste Community-Version angeboten oder der Kunde muss auf die

erst später erscheinende stabile aber kostenpflichtige Enterprise Version wechseln. Wer jedoch mit dieser Unsicherheit leben kann, für den ist JBoss immer noch ein gutes Rundumangebot. Wobei hier der Schritt auf die nächste Applikationsserver-Version, durch die Architektur und Versionsaktualisierung der verwendeten Komponenten, nicht ohne Stolperfallen ist. Spätestens mit der für Mitte des Jahres geplanten Enterprise Applikationsserver Version 6, die wiederum auf der Community-Version 7.1 beruht, sollten die meisten aktuellen Probleme beseitigt sein.

Geronimo - der Nachzügler

Apache Geronimo ist vor allem als freie Herausforderung zu JBoss angetreten. Deswegen setzen viele bewährte andere OpenSource-Komponenten ein. So hat es Geronimo seit der Version 1.0 und der Zertifizierung für J2EE 1.4 immer als zweiter OpenSource-Server, nach Glassfish, geschafft jede JEE-Zertifizierung zu erfüllen. Die am meisten eingesetzte Version ist 2.1.8, wobei bei dieser Version seit der Vorstellung Anfang 2008 immer wieder Fehlerkorrekturen erschienen sind. Offizielle existiert für Geronimo 5 Jahre offizieller Support und weitere 3 Support-Jahre sind bei IBM buchbare. Damit bietet Geronimo, typisch Apache, von allen drei Produkten die längsten Supportzeiträume bei regelmäßigen Weiterentwicklungen, was eine gute Planungssicherheit ergibt. Jedoch darf nicht verschwiegen werden, dass die Entwicklung und die Unterstützung der Community in den letzten Jahren etwas nach gelassen hat. Trotzdem bleibt Geronimo, gerade als Integrationsplattform für die vielen Apache Java EE-Projekte wichtig.

Geronimo bietet gerade für Eclipse und Maven eine gute Integration an. Es gibt viele Möglichkeiten das Deployment zu automatisieren oder den Server über eine komfortable Oberfläche zu konfigurieren oder zu überwachen.

Wer sich zutraut selbst Hand anzulegen und trotz unvollständiger Dokumentation sich auf OpenSource einzulassen, erhält mit Geronimo einen verlässlichen und innovativen Begleiter. Etwas Konkurrenz im eigenen Haus hat Geronimo durch den ebenfalls zertifizierten TomEE-Server und dem demnächst geplanten KarafEE-Server bekommen. Für die hybride Entwicklung von OSGi und JavaEE-Anwendungen scheint Geronimo eine gute Wahl zu sein, da die Beispiele von Karaf und Aries vom gleichen Hersteller ohne zusätzliche Anpassungen in Geronimo lauffähig sind. Durch die stagnierenden Committer-Zahlen und dem Ausstieg von

Apache aus dem Java-Community-Prozess sind die Apache-Projekte zwar nicht gefährdet, doch die Weiterentwicklung etwas gebremst. Umso erfreulicher ist dass große Unterstützer des JCP, wie IBM und JBoss, sich immer mehr bei Apache Projekten engagieren und so die OpenSource-Fahne weiter wehen lassen.

Zusammenfassung

Die Unterschiede zwischen den Produkten werden immer weniger. Trotzdem kann die folgende Übersicht bei der ersten Orientierung helfen. Interessant ist der Vergleich der Entwicklung der Codegröße der jeweiligen Produkte. Hier zeigt sich die Tendenz, dass die Produkte, ähnlich wie der Java-Standard, immer größer werden. Durch die neue Modularität sind die neuen Applikationsserver immer anpassbarer geworden, was sich nicht zu Letzt auf das schnelle Startverhalten und den geringen Ressourcenverbrauch auswirkt.

<Vergleich der Codegrößen.jpg>

Abbildung 1 Vergleich der Codegrößen (Ohloh.net)

	Glassfish	JBoss	Geronimo
Lizenz	CDDL1.1, GPL 2	LGPL 2.1	ASF
Beurteilung	<ul style="list-style-type: none"> + sehr gute Dokumentation, viele Beispiele + Cluster Support + Migrations-, Upgrade-Unterstützung + gute IDE-, BUILD-Integration + gute Administrier-, Automatisierbarkeit (GUI, CLI) + Supportmöglichkeit + aktuelle Referenzimplementierung + regelmäßige Release auch ältere mit klarer 	<ul style="list-style-type: none"> + gute Dokumentation + verbesserte Administrierbarkeit + gute Integration in JBoss, RedHat-Produkte + gute Cloud-Integration + gute Testbarkeit Arquillian + gute IDE-, BUILD-Integration - Webcontainer: Tomcat - seltenere und unregelmäßige Release nur für aktuellste Version - große Community, jedoch mit wenig 	<ul style="list-style-type: none"> + stabiles, schlankes und gut anpassbares Produkt mit ausgereiften Best-Of-Breed-Komponenten + gute IDE-, BUILD-Integration + Langzeitsupport + gute Mitwirkungsmöglichkeit, jedoch kleine Community + regelmäßige Release auch ältere + hybride Version für OSGi, Jetty/Tomcat,,

	Roadmap + mittlere Produktgröße 83 MB	Mitwirkungsmöglichkeiten bei der Produktweiterentwicklung - relative neue Architektur - eingeschränkte Abwärtskompatibilität - Langzeitsupport nur für kommerzielle Variante - größtes Produkt 103 MB	CXF, Axis2 - wenige Dokumentation, Beispiele + kleinstes Produkt mit 74-91MB + sehr freie ASF-Lizenz
Plattformen	LINUX, Windows, SOLARIS, AIX	LINUX und Windows	LINUX, Windows, SOLARIS, AIX
Webcontainer	Grizzly	Tomcat	Jetty, Tomcat
OSGi	Felix	Felix, Karaf	Equinox, Felix, Karaf, Aries
JSF	Mojarra	Mojarra, Richfaces, SEAM	myFaces, OpenWebBeans
CDI	Weld	Weld	OpenWebBeans
JAX-WS, JAX-RS	Metro, Jersey	CXF, RESTEasy	CXF, Axis2, WINK
JPA	EclipseLink, Hibernate Validator	Hibernate, Hibernate Validator	OpenJPA, BVal

Abbildung 1: Übersicht und Bewertung

Fazit

Durch die Übernahme von SUN durch ORACLE hat Java wieder eine Zukunft. Der aktuelle und der kommende Java EE-Standard haben zu einer Renaissance der Applikationsserver geführt. Trotzdem weisen die Applikationsserver mit OSGi den Weg zu mehr Modularität. Dadurch dass der Java-Markt gesättigt ist, wird es hier wenig Bewegung geben.

Für viele Projekte steht der Schritt in die JavaEE 6-Zukunft noch bevor.

Trotz vieler Unterschiede in Lizenz, Support, Architektur und den verwendeten Komponenten haben die hier vorgestellten Produkte, durch den Java-EE-Standard

viele Gemeinsamkeiten. Kurz zusammen gefasst kann man sagen, dass Glassfish sich etabliert hat und für die JavaEE-Funktionen am weitesten ausgereift ist. JBoss und Geronimo bieten erst seit kurzem eine volle JavaEE 6-Unterstützung, so dass hier noch einige Fehler schlummern, die im Laufe dieses Jahres behoben sein werden. Als OpenSource-Applikationsserver bieten alle drei Kandidaten interessante und innovative Alternativen zu kommerziellen Anbietern oder reinen Webservern an, die einen Wechsel attraktiv machen. Erste Ansätze, wie der Weg der Applikationsserver in die Cloud aussehen könnte, sind zwar sichtbar, doch noch nicht wer hier das Rennen machen könnte.

Link:

[http://www.ohloh.net/p/compare?project_0=GlassFish&project_1=Apache+Geronimo
&project_2=JBoss+Application+Server](http://www.ohloh.net/p/compare?project_0=GlassFish&project_1=Apache+Geronimo&project_2=JBoss+Application+Server)

<http://geronimo.apache.org>

<http://www.jboss.org/jbossas>

<http://glassfish.java.net/>

Kontakt:

Frank Pientka

Frank.Pientka@materna.de