

1 Einleitung

Dieses Kapitel gibt Ihnen einen kurzen Überblick über den Apache-Geronimo-Server. Neben seiner Historie werden Produktvarianten sowie die unterstützten Standards und Produkte vorgestellt. Abgerundet wird das Kapitel mit einem Vergleich zu anderen Open-Source-Java-EE-Anwendungsservern. Der Inhalt dient eher als Hintergrundinformation zu einer ersten Orientierung oder einer Entscheidung, ob Geronimo oder sein Bruder WAS CE besser auf die Anforderungen Ihrer Systemumgebung passen. Daher ist dieses Kapitel für das weitere Verständnis des Buches nicht Voraussetzung und kann auch überfliegen werden.

1.1 Übersicht

Dieses Buch stellt den jüngsten der zertifizierten Open-Source-Java-EE-Anwendungsserver vor: den Apache Geronimo 2.0. Zusätzlich wird auch auf den darauf basierenden WAS CE eingegangen, den IBM WebSphere Application Server Community Edition.

*Apache Geronimo 2.0 und
WAS CE*

Es wird außerdem die zum Zeitpunkt der Bucherstellung erschienene Geronimo-Version 2.1 behandelt. Die in diesem Buch gezeigten Beispiele wurden mit Geronimo 2.0.2 getestet. Deshalb sollten diese bis auf die Versionsnummern in den Modulen auch auf aktuelleren Geronimo-Versionen lauffähig sein. Dazu müssen Sie nur die Versionsnummern der Module auf die von Ihnen verwendete Version anpassen.

Apache Geronimo 2.1

Die vorgestellten Beispiele verwenden schwerpunktmäßig folgende Softwareprodukte: Geronimo 2.0.2, Sun JDK 5.0, Apache Tomcat 6.0, Apache Derby 10.2, Apache Directory Server 1.0, Apache-HTTP-Server 2.0, Eclipse 3.3 + WTP 2.0, Maven 2.0.8 und Subversion Client 1.4.5.

*Verwendete
Softwareprodukte*

Betriebssysteme

Durch Java sollten alle Komponenten plattformunabhängig sein, auch wenn diese für das Buch ausschließlich unter Linux und Windows getestet wurden.

Aufbau des Buches

Nach einer Einführung in Geronimo werden dessen grundlegende Konzepte und Einsatzmöglichkeiten beschrieben. Neben den zugrunde liegenden Lizenzmodellen werden je nach Anwenderkreis die einzelnen Aufgabenfelder Installation, Konfiguration, Entwicklung, Deployment und Betrieb behandelt. Zur praktischen Verdeutlichung der verschiedenen Themen wird auf die online verfügbaren Beispiele verwiesen, die kurz vorgestellt werden. Etwas ausführlicher wird die Erstellung eines EJB-3.0-Beispiels mit Eclipse erörtert.

Migration und Aktualisierung

Einen speziellen Raum nimmt das Thema Migration und Aktualisierung bestehender Java-EE-Anwendungen ein. Dies wird anhand einer Fallstudie mit den von IBM für Apache Geronimo frei zur Verfügung gestellten Referenzanwendungen »Plants by WebSphere« und »DayTrader« Schritt für Schritt vorgestellt. Abschließend werden auch fortgeschrittene Konzepte wie Optimierung und Erweiterungsmöglichkeiten besprochen.

Abgeschlossen wird das Ganze mit einem Blick auf die kommenden Versionen und die dort geplanten Funktionen.

1.2 Geschichte

JBoss

Wie so oft in der Geschichte von Softwareprodukten ging auch Geronimo aus einem Streit zwischen Entwicklern hervor. In diesem Fall waren es die Entwickler des JBoss-Applikationsservers, eines Geronimo-Vorgängerproduktes. Neben technischen Kontroversen spielten hier unterschiedliche Auffassungen über die richtige Lizenz eine Rolle. Statt der GNU Lesser General Public License (LGPL) wollten einige Entwickler die freiere und weniger restriktive Apache-Lizenz verwenden. Sie versprach breitere Einsatzmöglichkeiten, ist jedoch nicht kompatibel zur LGPL.

Apache-Lizenz

Geronimo wird daher unter der Apache License 2.0 veröffentlicht. Eine Software, die damit lizenzierten Code verwendet, muss nicht erneut unter diese Lizenz gestellt werden, was die Einsatzmöglichkeiten der neuen Produkte in der Regel erhöht. Außerdem erlaubt die Apache-Lizenz, ähnlich der BSD-Lizenz, Veränderungen am Quellcode und die ausschließliche Weitergabe in Binärform, sofern die Lizenzbedingungen beigelegt werden oder darauf hingewiesen wird. Es muss dabei kein Quellcode beigelegt werden. Damit ist sie die freieste der hier erwähnten Lizenzen. Ein Grund, warum diese gerade bei

Softwareherstellern, die Apache-Software in ihre Produkte integrieren, beliebt ist.

Bevor es bei Geronimo so weit war, mussten jedoch sowohl technisch als auch lizenzrechtlich geeignete Bibliotheken zur Erstellung von Geronimo entwickelt und eingebunden werden. Mittlerweile ist Geronimo jedoch ein Top-Level-Projekt, das auch von vielen anderen Apache-Projekten verwendet und unterstützt wird.

Eine Variante von Geronimo existiert ebenfalls als kostenloses Produkt unter dem Namen IBM WebSphere Application Server Community Edition (WAS CE). Hintergrund ist die Firma GlueCode, die an der Entwicklung des Apache-Geronimo-Servers mitwirkte, dafür Support anbot und 2005 von IBM übernommen wurde. Sie kümmerte sich seitdem vor allem um die Anpassung von Geronimo an die IBM-spezifischen Anforderungen. IBM stellte jedoch auch einen großen Teil der über 50 Geronimo Committer.

WAS CE

WAS CE wird unter IBMs International License Agreement for Non-Warranted Programs (ILAN) und nicht unter der Apache-Lizenz zur Verfügung gestellt. Die IBM-Lizenz erlaubt die Verwendung und Veränderungen des Programms und schließt Haftung von IBM-Seite aus. Dadurch ist der WAS CE zwar kostenlos verfügbar, jedoch ohne den dazugehörigen Quellcode. IBM bietet allerdings längerfristig Fehlerkorrekturen an, im Gegensatz zu der in der Entwicklung weiter vorne liegenden Geronimo-Version.

ILAN-Lizenz

Trotz seines geringen Alters hat Geronimo bereits eine recht dynamische Geschichte hinter sich. Die wichtigsten Geronimo-Versionen waren sicher die erste für J2 EE 1.4 zertifizierte Geronimo-Version 1.0 und die für Java EE 5.0 (Tomcat 6, Jetty 6) zertifizierte Version 2.0. Die am weitesten verbreitete Version ist zurzeit 1.1, da für sie viele Plugins und Beispiele erhältlich sind. Wir verwenden in diesem Buch hauptsächlich die Version 2.0, gehen jedoch auch auf Migrationsmöglichkeiten von der Vorgängerversion ein und erwähnen an einzelnen Stellen die Version 2.1.

Geronimo-Versionen

Die Tabelle 1–1 fasst die noch recht junge Versionsgeschichte von Apache Geronimo zusammen. Hier können Sie sehen, wie dieser kontinuierlich weiterentwickelt wurde, und daraus den groben Veröffentlichungszyklus für Ihre eigene Planung abschätzen.

Tab. 1-1

*Geronimo-Versionen von
Februar 2004 bis heute*

26. Februar 2004	Wechsel vom Apache-Inkubator zum Top-Level-Projekt
5. Oktober 2005	Version 1.0-M5: Zertifizierung nach dem Standard J2EE 1.4, Maven1
25. Oktober 2005	Bekanntgabe der IBM WebSphere Community Edition
5. Januar 2006	Version 1.0: nur JDK-1.4-Support, configld und parentld
26. Juni 2006	Version 1.1: Adminkonsole, Little-G, Plugins, Server-cloning, Module IDs, Offline-Deployment
18. September 2006	Version 1.1.1: kleinere Fehlerkorrekturen
22. Dezember 2006	Version 1.2: Java 5.0, OSGI, Yoko ORB, OpenEJB 3.0, JMX Viewer, LDAP Viewer, Keystores, Certificate Authority, globaler JNDI-Support, OpenJPA, EJB Cluster
4. Juni 2007	Version 2.0: M6 Java-EE-5.0-zertifiziert (TCK), Maven2
20. August 2007	Version 2.0.1: offizielle Java-EE-5.0-Unterstützung mit angepasster Dokumentation, Beispielen und Werkzeugen
19. Oktober 2007	Version 2.0.2: kleinere Fehlerkorrekturen, neues Format für die EJB-JNDI-Namen, Tomcat 6.0.13
18. Februar 2008	Version 2.1: komplett auf Plugins, Assemblies umgestellt, besseres Monitoring
28. April 2008	Version 2.1.1: Assemblies, flexiblere Webkonsole, GShell, Cluster-Support, Monitoring-Plugin, Groovy 1.1, Dojo 1.0.2, Tomcat 6.0.14
5. August 2008	Version 2.1.2: Beispiele werden erstmalig als Archiv gesammelt zur Verfügung gestellt ^a . WADI-Cluster-Unterstützung für Tomcat. Fehlerkorrekturen, aktualisierte Bibliotheken für Ant 1.7.1, Tomcat 6.0.16, CXF 2.0.8, JavaMail 1.5, MyFaces 1.2.3, WADI 2.0, DWR 2.0.3, und Yoko 1.0, Aspectj 1.5.3, Groovy 1.5.6, Plexus
15. September 2008	Version 2.1.3: Fehlerkorrekturen ^b , Aktualisierungen (z.B. Derby 10.4.1.3, Tomcat 6.0.18, DWR 2.0.5, Genesis 1.3.1, OpenJPA 1.0.3, MS SQLServer XA-Konnektor), funktionsfähige Beispiele
voraussichtlich 1. Quartal 2009	Version 2.2 ^c : Verbesserte GShell, Fehlerkorrekturen, Aktualisierungen (z.B. Derby 10.4.2, Dojo 1.1.1, Java 6.0, XStream, JAX-WS 2.1/JAX-B 2.1, Axis2 1.4, CXF 2.1.x, Jetty 7

a. <http://geronimo.apache.org/plugins/samples-2.1.2/>

b. <http://cwiki.apache.org/GMOxPMGT/geronimo-213-release-status.html>
http://svn.apache.org/repos/asf/geronimo/server/branches/2.1.3/RELEASE_NOTES-2.1.3.txt

c. <http://cwiki.apache.org/GMOxPMGT/geronimo-22-release-status.html>
<http://cwiki.apache.org/GMOxPMGT/geronimo-22-release-roadmap.html>

Generell ist es schwierig, die Installationszahlen für Open-Source-Produkte zu schätzen. Dies trifft nur eingeschränkt auf Apache Geronimo zu, denn die regelmäßig unter <http://people.apache.org/~vgritsenkol/>

stats/projects/geronimo.html veröffentlichten Download-Zahlen zeigen deutlich ein wachsendes Interesse an Geronimo (siehe Abb. 1–1). Das gilt insbesondere für die Java-EE-5.0-zertifizierte Geronimo-Version 2.x, die seit August 2007 verfügbar ist.

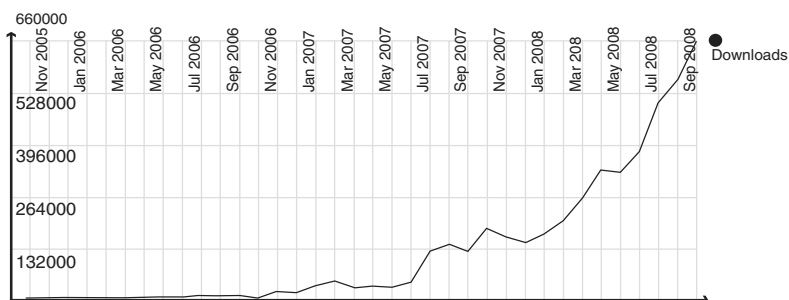


Abb. 1–1
Geronimo-Download-
Statistik

1.3 Mögliche Einsatzszenarien für Geronimo

Da Geronimo ein ausgewachsener und auch zertifizierter Applikationsserver ist, kann er für zahlreiche Java-Unternehmensanwendungen, wie sie die Java-EE-Plattform adressiert, eingesetzt werden.

Möchte man Geronimo in seine Anwendung integrieren und den Server darin komplett verschwinden lassen, so ist dies mit der sehr freien Apache-Lizenz problemlos möglich. Hierfür bietet sich die Geronimo-Variante Micro-G an (siehe Abschnitt 1.4.2).

In einer anderen Variante namens Little-G (siehe Abschnitt 1.4.2) steht Geronimo als schlanke Ablaufumgebung in direkter Konkurrenz zu den spezialisierten Webservern Tomcat und Jetty, bietet jedoch die von Applikationsservern bekannten Möglichkeiten.

Sein bevorzugtes Einsatzgebiet sind sicherlich serviceorientierte Webarchitekturen. Hier setzt Geronimo auf ausgereiften und weitverbreiteten Komponenten wie Tomcat oder Axis auf und integriert diese zu einer einheitlichen, erweiterbaren und skalierbaren Plattform.

Durch sein Mandantenkonzept, das in Abschnitt 6.3 ausführlicher vorgestellt wird, ist Geronimo ein idealer Kandidat für das Hosting von Java-EE-Anwendungen. Dadurch können Sie mit einer einzigen physischen Geronimo-Installation mehrere logische Serverinstanzen erstellen, die getrennt administriert werden, zentrale Ressourcen jedoch gemeinsam nutzen können. Demzufolge reduziert sich nicht nur der Installations-, sondern auch der Pflegeaufwand zum Verwalten der Geronimo-Server.

*Geronimo-Framework als
Teil der Anwendung*

Webarchitekturen

*Serviceorientierte
Architekturen*

*Hosting von Java-EE-
Anwendungen*

*Java-EE-Client-Server-
Anwendungen in
Entwicklung, Test und
Produktion*

Insbesondere für die Entwicklung und den Test von Java-EE-Anwendungen bietet Geronimo für weitverbreitete Entwicklungswerkzeuge, wie Eclipse und NetBeans bzw. Ant und Maven, eine gute Integration und Unterstützung. Da diese auch bei der eigenen Entwicklung von Geronimo eingesetzt werden, lässt er sich vorzüglich in den eigenen automatisierten Build- und Testprozess integrieren. Die Möglichkeit, nicht nur die Anwendung, sondern auch deren Konfiguration automatisiert und transparent zu deployen, erleichtert nicht nur die Entwicklung und den Betrieb, sondern hilft auch dabei, eine Test- und Integrationsumgebung schnell und in konsistenter Weise zu pflegen.

Den Anforderungen, die an den produktiven Einsatz gestellt werden, wie Ausfallsicherheit und Skalierbarkeit, wird Geronimo mit entsprechenden Konzepten gerecht. Besonders gelungen ist hier die konfigurierbare Überwachung vieler Parameter in der komfortablen Verwaltungsoberfläche.

Durch seine Erweiterungsmöglichkeiten und die unterstützten Produkte lässt er sich auch gut in bestehende Umgebungen integrieren.

1.4 Produktvarianten

*Mit Jetty und CXF oder
mit Tomcat und Axis*

Apache Geronimo wird in zwei Varianten als Binärversion angeboten: einmal mit dem Webcontainer Jetty und der Webservice-Engine CXF (CXF steht für CeltiXFire, da es aus den Projekten Celtix und Xfire hervorgegangen ist) und alternativ mit dem Webcontainer Tomcat und der Webservice-Engine Axis.

Obwohl beide die Java-EE-Zertifizierung bestanden haben, ist die Tomcat-Variante, hauptsächlich wegen der größeren Verbreitung, die am meisten getestete und genutzte Variante. Aus diesem Grunde wird sie auch von der IBM WebSphere Community Edition verwendet, wobei dann nur die IBM JVM unterstützt wird.

1.4.1 Geronimo mit Jetty oder mit Tomcat

Als echter Java-EE-5.0-Anwendungsserver sind für Apache Geronimo folgende Webcontainer und Webservice-Implementierungen zertifiziert:

- Apache Geronimo mit Tomcat-Webcontainer, Axis2 für Webservices und OpenJPA als Persistenzschicht
- Apache Geronimo mit Jetty-Webcontainer, CXF für Webservices und OpenJPA als Persistenzschicht

Welche Variante Sie einsetzen möchten, hängt auch von den eigenen Anforderungen und Wünschen ab. Die derzeit am meisten eingesetzte Variante ist sicherlich die mit Tomcat und Axis. Sie ist auch die Grundlage für den von IBM herausgegebenen Geronimo-Klon WAS CE, den wir in Abschnitt 1.4.3 vorstellen und auf den wir auch im Laufe des Buches immer wieder eingehen werden.

1.4.2 Geronimo als Micro-G, Little-G oder Big-G

Um für verschiedene Anforderungen einsetzbar zu sein, wird Geronimo in drei Paketgrößen angeboten.

Möchte man Geronimo komplett in seine Anwendung integrieren, so kann man die Variante Micro-G einsetzen, die nur das Geronimo-Plugin-Framework, den Kern, die JMX- und RMI-Konnektoren verwendet. Mit diesem ca. 22 MB großen Framework können Sie komplette Java-EE-Anwendungen realisieren.

Micro-G

Wer nicht alle Java-EE-Funktionalitäten benötigt und mit einem einfachen Java-Webserver auskommt, für den steht Geronimo als Little-G-Variante zur Verfügung. Den abgespeckten Geronimo-Server Little-G gibt es mit Tomcat oder Jetty als Webcontainer. Er ist mit ca. 17 MB wesentlich kleiner als der komplette Geronimo (Big-G) mit 55 MB oder der WAS CE mit 60 MB. Der Vorteil einer solchen Variante gegenüber einer eigenen Tomcat- oder Jetty-Installation besteht darin, dass es sich bei dem Little-G genannten Server um einen echten Geronimo-Server handelt, der jedoch nur die minimalen Dienste für Webanwendungen anbietet. So kann dieser mit den Anforderungen mitwachsen. Zusätzliche Serverfunktionalität bis zu einem kompletten Applikationsserver kann zur Laufzeit nachträglich hinzuinstituiert werden. Damit kann das bereits vorhandene Wissen wiederverwendet werden und es fällt auch kein zusätzlicher Testaufwand, wie bei einem Wechsel zu einem anderen Applikationsserver, an.

Little-G

Wenn man die Dummy-Module für nicht vorhandene Funktionalitäten abzieht, werden nur 14 Module geladen. Als einzige Webanwendung steht der Remote Deployer zur Verfügung, sodass man sowohl Anwendungen als auch Plugins einsetzen kann.

Doch selbst in dieser Konfiguration kann man noch die Protokolle bzw. Konnektoren Tomcat-Konnektor AJP, JMX-Remoting-Konnektor und Tomcat-Konnektor HTTPS abschalten. RMI und HTTP werden jedoch für weitere Deployments benötigt (eine minimale Konfiguration wird in Anhang C beschrieben).

Als letzte und umfangreichste Variante bleibt der große Geronimo-Server, den Sie mit Jetty oder Tomcat für die Realisierung und

Big-G

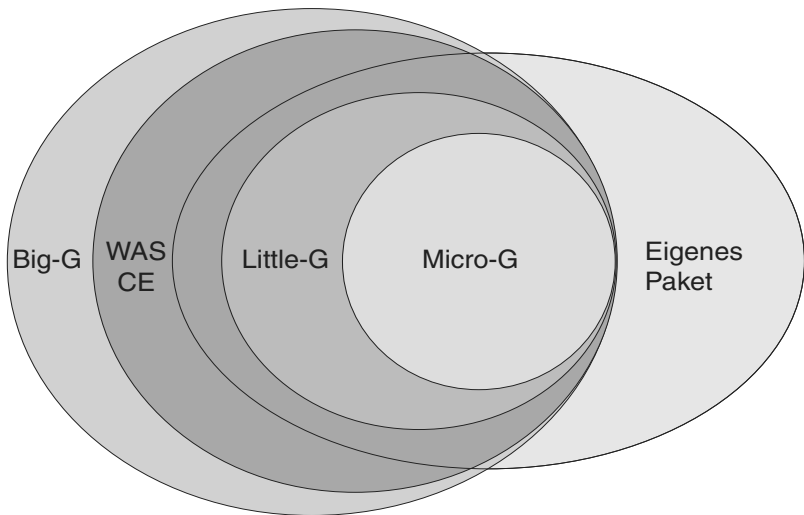
Verwaltung kompletter Java-EE-Anwendungen einsetzen können. Dieser wird auch Big-G genannt und verfügt neben einer Willkommenseite auch über eine komfortable Webadministrationskonsole, einen integrierten Datenbankserver sowie einen JMS-Server und bietet bereits die Möglichkeiten, EJBs und Webservices mit den dafür benötigten Protokollen und Bibliotheken zu verwenden.

*Eigene Pakete
(Assemblies)*

Neben diesen »offiziellen« Paketen von Apache Geronimo können Sie sich auch eigene aus den Einzelkomponenten Ihrer Geronimo-Installation zusammenstellen. Diese Art von benutzererzeugten Geronimo-Paketen werden auch Assemblies genannt.

Alle diese Pakete von Geronimo sind in der Abbildung 1–2 im Überblick dargestellt.

Abb. 1–2
*Geronimo-Pakete im
Überblick*



1.4.3 WAS CE

Da IBM schon recht früh die Geronimo-Entwicklung unterstützte, bietet IBM mit dem WebSphere Application Server Community Edition (WAS CE) auch eine eigene Variante an, die bis auf wenige eigene Fehlerkorrekturen auf dem Geronimo-Code beruht. Der Einsatz des WAS CE bietet sich dann an, wenn Sie Plattformen wie AIX oder Solaris einsetzen, die offiziell von Geronimo nicht unterstützt werden. Das Gleiche gilt für den Einsatz der IBM JVM oder die Integration von weiteren Produkten wie LDAP-Server oder Datenbanken.

Die Installation wird jedoch um eine eigene Dokumentation, proprietäre JDBC-Treiber sowie Supportskripte, die von der WebSphere-

Vollversion bekannt sind, erweitert. Die genauen Versions- und Bibliotheksunterschiede können Sie entweder den Dateien `CHANGES.txt` oder `client_artifact_aliases.properties` entnehmen. Weitere Informationen finden sich auch in der sehr guten mehrsprachigen Online-Dokumentation des WAS CE. Alternativ kann man diese Dokumentation auch als Offline-Dokumentation herunterladen.

Im Gegensatz zu Geronimo enthält WAS CE eine größere Unterstützung von Systemen und Plattformen, was eine Integration in eine bestehende Umgebung erleichtert, und das nicht nur für treue IBM-Kunden. Ein weiterer Grund für den Einsatz von WAS CE könnte sein, ihn als Entwicklungs- und Testumgebung für WebSphere-Anwendungen einzusetzen. Hier kann man statt der bereits enthaltenen Java-Datenbank Derby auch die kostenlose und umfangreichere DB2-Community-Express-Version verwenden, die für Produktionsumgebungen geeigneter ist. WAS CE wird sowohl offiziell von IBM für die Entwicklung von Anwendungen mit dem WebSphere Portal 6.1 als auch mit dem WebSphere Web Feature Pack unterstützt.

Einen Überblick über die WAS-CE-Versionen, die bisher parallel zum Apache Geronimo veröffentlicht wurden, bietet Tabelle 1–2.

Professioneller Open-Source-Server

Datum	WAS CE	
15. November 2005	Version 1.0	basiert auf einer angepassten Geronimo-1.0-Version
15. Februar 2006	Version 1.0.0.1	basiert auf einer angepassten Geronimo-1.0-Version
4. April 2006	Version 1.0.1	basiert auf einer angepassten Geronimo-1.0-Version
10. April 2006	Version 1.0.1.1	basiert auf einer angepassten Geronimo-1.0-Version
12. Mai 2006	Version 1.0.1.2	basiert auf einer angepassten Geronimo-1.0-Version
19. September 2006	Version 1.1	basiert auf einer angepassten Geronimo-1.1.1-Version
3. November 2006	Version 1.1.0.1	basiert auf einer angepassten Geronimo-1.1.1-Version
19. Juni 2007	Version 1.1.0.2	basiert auf einer angepassten Geronimo-1.1-Version
25. September 2007	Version 2.0.0.0	basiert auf einer angepassten Geronimo-2.0.1-Version →

Tab. 1–2

Bisher veröffentlichte WAS-CE-Versionen

Datum	WAS CE	
13. November 2007	Version 2.0.0.1	basiert auf einer angepassten Geronimo-2.0.2-Version, OpenJPA 1.0.0
15. April 2008	Version 2.0.0.2	basiert auf einer angepassten Geronimo-2.0.2-Version, Apache Axis2 1.3
25. Juli 2008	Version 2.1.0.0	basiert auf einer angepassten Geronimo-2.1.1-Version
4. September 2008	Version 2.1.0.1	basiert auf einer angepassten Geronimo-2.1.1-Version, Tomcat 6.0.18

In der Tabelle erkennt man auch die Releasestrategie, jährlich (2005, 2006, 2007, 2008) ein größeres Release und dazwischen mehrere Updates herauszubringen.

Eine detaillierte Liste der Änderungen gegenüber Apache Geronimo finden Sie für jede WAS-CE-Version unter der URL

<http://publib.boulder.ibm.com/wasce/changes/wasce-changes.htm>

*Der kommerzielle IBM
WebSphere Application
Server*

Obwohl die WebSphere Community Edition eine andere Basis und Zielgruppe hat als der kommerzielle IBM WebSphere Application Server, ergeben sich Gemeinsamkeiten. So platziert IBM den WAS CE als Einsteiger- und Entwicklerversion für neue Technologien und Anwendungen. Gleichzeitig bietet IBM ausführliche Informationen und Werkzeugunterstützung, die einen Umstieg auf die größeren WebSphere-Applikationsserver erleichtern.

Auch wenn die Migration zwischen dem »großen« WebSphere Application Server und der Community Edition gut beschrieben ist, bedeutet das jedoch nicht, dass der Umstieg ohne Probleme oder Aufwand möglich wäre, denn beide WebSphere-Server haben bis auf den Namen nur wenig technologische Gemeinsamkeiten. WAS CE unterstützt z.B. aktuellere JDK-, Java-EE- und Frameworkversionen als der große Bruder. Das Buch widmet diesen Themen deshalb auch ein eigenes Kapitel 9.

Die Abb. 1–3 gibt eine grobe Einordnung von WAS CE in die komplette WebSphere-Familie.

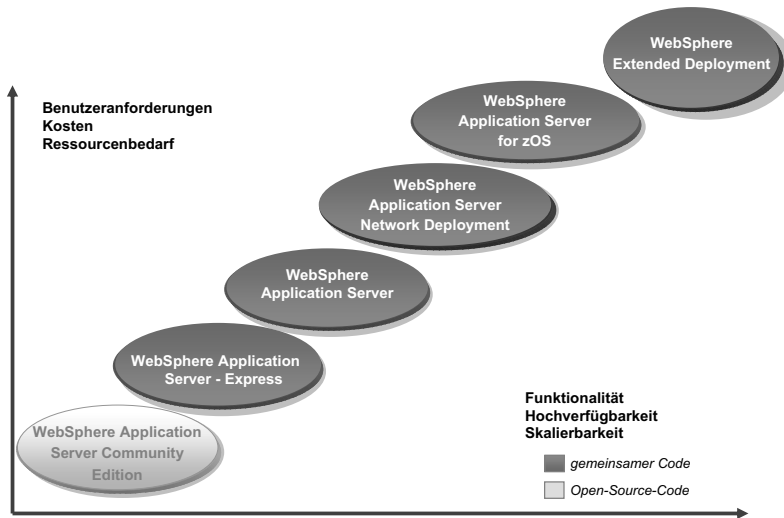


Abb. 1-3
Integration von
WebSphere CE in die
WebSphere-Familie
(Quelle: IBM)

1.4.4 Vergleich der Varianten

Die kleinste Variante ist Micro-G, mit der Sie sich Ihren eigenen Geronimo-Server zusammenstellen können. Dabei handelt es sich um das Geronimo-Framework, mit dem Kern und der Möglichkeit, Plugins nachzuinstallieren, jedoch ohne einen Webcontainer.

Micro-G

Für die Entwicklung und die ausschließliche Verwendung für Webanwendungen ist die Little-G-Version die geeignete Variante. Ob man hier Tomcat oder Jetty als Webcontainer verwenden möchte, hängt von den eigenen Erfahrungen und Rahmenbedingungen ab.

Little-G

Wird jedoch eine komfortable und erweiterbare Administrationsoberfläche bevorzugt, mit der man Java-EE-Anwendungen und ihre Ressourcen verwalten und überwachen kann, so sollte man unbedingt die große Geronimo-Version Big-G einsetzen.

Big-G

Wenn man eine zugesicherte Unterstützung für spezielle Plattformen und Integrationsprodukte haben möchte, entscheidet man sich für WAS CE. In diesem Fall kann man auch mit einem kommerziell verfügbaren Support und einem längerfristig stabileren Releaseplan rechnen. Der einzige Nachteil dieser Variante ist, dass hierfür der zugrunde liegende Quellcode nicht vollständig öffentlich verfügbar ist. Dies ist der Vorteil von Apache Geronimo, denn dort können Sie sich aus den vorhandenen Quellen Ihren eigenen Geronimo-Server aus sogenannten Assemblies zusammenkonfigurieren.

WAS CE

1.5 Unterstützte Standards

Dieses Unterkapitel stellt die für Geronimo relevanten Standards kurz vor. Für eine vertiefende Einführung in die Standards wird auf die entsprechende Literatur verwiesen.

Java EE 5.0 Die für eine Java-EE-5.0-Zertifizierung wichtigen Standards – Java Platform Standard Edition (Java SE 5.0) und Java Platform Enterprise Edition 5.0 (Java EE 5.0) – fassen einige Einzelstandards zusammen, die die Bereiche Web, Webservices, Unternehmensanwendungen und Management von Java-Anwendungen betreffen.

Java EE 6.0 Der kommende Standard Java EE 6.0 befindet sich noch in der Abstimmungsphase, sodass dessen genauer Inhalt erst zu einem späteren Zeitpunkt feststeht, um dann auch wieder von Geronimo unterstützt zu werden. Als Ablaufumgebung für den Geronimo 2.0 setzt die Apache Foundation derzeit eine JVM 5.0 voraus; allerdings läuft er auch schon mit der JVM 6.0 gut.

Abb. 1-4

Von Geronimo
unterstützte Java-EE-
Standards im Überblick

Web	Web Services	Enterprise
Servlet 2.5 JSP 2.1 JSTL 1.2 JSF 1.2 Portlet 1.0 Dojo 0.4.3 DWR 1.1.4	WS-Metadata JAXB 2.0 JAX-WS 2.0 SAAJ 1.1/1.3 SOAP 1.1/1.2 WSDL 1.1/2.0	EJB 3.0 JPA 3.0 JAF 1.1 JTA 1.1 JACC 1.1 JavaMail 1.4

1.5.1 Webstandards

Bei den Standards zur Realisierung von Weboberflächen handelt es sich mit der Servlet-API, neben den JDBC-Datenbank-Verbindungsstandards, um die ältesten Schnittstellen für Java-Unternehmensanwendungen. Danach kamen mit der Beschreibungssprache für Webseiten (JavaServer Pages, JSP) und den Tag-Makros (JavaServer Pages Standard Tag Library, JSTL) als letzter Standard die JavaServer Faces (JSF) für die Erstellung von Web-Oberflächenelementen hinzu. Dabei sind folgende aktuellen Versionen relevant, die bis auf beschriebene Ausnahmen abwärtskompatibel sind:

- Java Servlet 2.5 (JSR-154)
- JavaServer Pages 2.1 (JSP, JSR-245)
- JSP Debug 1.0 (JSP Debug, JSR-45)

- JavaServer Faces 1.2 (JSF , JSR-252)
- JavaServer Pages Standard Tag Library (JSTL 1.2, JSR-52)

Das Zusammenspiel dieser wichtigen Java-Webstandards ist in Abbildung 1–5 dargestellt.

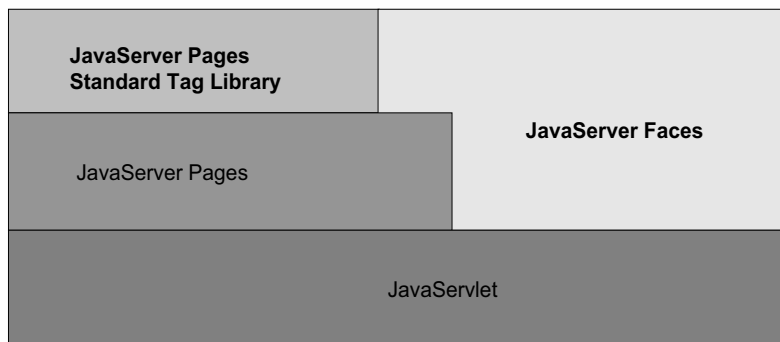


Abb. 1–5

Zusammenspiel der wichtigen Java-Webstandards

Als Ablaufumgebung wird ein zur jeweiligen JSP-/Servlet-Version konformer Webcontainer benötigt. Der Apache Tomcat Webserver, der dafür bisher die Referenzimplementierung war, unterstützt z.B. in der Version 6.0 Servlet 2.5 und JSP 2.1 und in der Version 5.5 Servlet 2.4 und JSP 2.0. Die mit Tomcat ausgelieferten JSP-Beispiele sind deshalb sowohl in der Version JSP 1.2 als auch 2.0 ausführbar. Bei einzelnen JSF- und JSTL-Funktionen gibt es jedoch Überschneidungen, sodass man hier beim Mischen der Versionen aufpassen muss.

1.5.2 Enterprise-Standards

Die Standards zur Java Enterprise Edition (Java EE) werden seit der Version J2EE 1.2 in einer Plattform mit SDK zusammengefasst und ausgeliefert. Für die Verwendung benötigt man einen Applikationsserver als Ablaufumgebung, der die jeweilige Java-EE-Version unterstützt. Die aktuelle Java Platform Enterprise Edition 5.0 (JSR-244) beinhaltet im Teilbereich Enterprise folgende Standards, in denen es vor allem um die Integration von Backend-Systemen geht (Datenbanken, Messaging-Systeme, Transaktionssysteme und Sicherheit).

- JavaBeans Activation Framework 1.1 (JAF, JSR-925)
- JavaMail 1.4 (JSR-919)
- Java Persistence API 1.0 (JPA, JSR-220)
- Java Transactions API 1.1 (JTA, JSR-907)
- Java Message Service API 1.1 (JMS, JSR-914)
- J2EE Connector Architecture 1.5 (JCA, JSR-112)

- Java Authorization Contract for Containers 1.1 (JACC, JSR-115)
- Common Annotations for the Java Platform 1.0 (JSR-250)
- Enterprise JavaBeans 3.0 (EJB, JSR-220)

1.5.3 Managementstandards für Java-Anwendungen

Der wichtigste und älteste Standard für das Management von Java-Anwendungen sind, wie bereits die JSR-Nummer 3 zeigt, die Java Management Extensions (JMX). Diese sind seit Java 5.0 sogar Teil der Standard Edition (SE). Die anderen Standards behandeln eher den Server- und damit den Java-EE-Bereich. Apache Geronimo ist hier eines der ersten Open-Source-Produkte, das alle drei Java Specification Requests fast vollständig implementiert.

- Java Management Extensions 1.4 (JMX, JSR-3)
- J2EE Management 1.1 (JSR-77)
- Java EE Application Deployment 1.2 (JSR-88)

1.5.4 XML- und Webservices-Standards

Im Java-EE-5.0-Standard wird großer Wert auf das Management, das Deployment von Webservices und die Unterstützung aktueller Standards gelegt. Daher finden sich hier auch die meisten neuen Standards im Bereich Webservices. Inzwischen sind davon sogar einige Standards in die Java-SE-6.0-Version eingeflossen.

Da Webservices nicht nur mit Java, sondern auch mit anderen Sprachen realisiert werden können, wird deren Standardisierung durch Organisationen wie W3C und OASIS.org vorangetrieben. Auch wenn einige der über 50 Spezifikationen veraltet oder kaum praxisrelevant sind, ist es schwierig, hier einen Überblick zu behalten. Die wichtigsten Standards werden für Java mit JAX{B, M, P, R, WS} abgekürzt. Nachdem in Java SE 5 nur JAXP integriert war, wurden in Java SE 6 auch die Webservice-Standards JAX-WS, JAXB, SAAJ und Java Webservices Metadata (JSR-181) aufgenommen. Das zeigt auch, wie wichtig dieser Bereich für Java in der Zukunft ist. Für die Standards JAX-WS 2.0, WS Metadata 2.0 und SAAJ 1.3 bietet Geronimo mit Axis2 oder CXF alternative Implementierungen an. Dadurch wird die Integration von weiteren Frameworks, die eine von beiden Implementierungen verwenden, vereinfacht.

- WSDL 1.1/2.0 (www.w3.org)
- Implementing Enterprise Web Services 1.2 (EWS, JSR-109)
- Java API for XML-Based Web Services 2.0 (JAX-WS, JSR-224)

- Java Architecture for XML Binding 2.0 (JAXB, JSR-222)
- Java API for XML based RPC 1.1 (JAX-RPC, JSR-101)
- Java API for XML Registries 1.0 (JAXR, JSR-93)
- Webservices Metadata for the Java Platform 2.0 (WS Metadata, JSR-181)
- SOAP (SOAP 1.1/1.2) (www.w3c.org)
- Java APIs for XML Messaging (SAAJ 1.3, JAXM 1.0, JSR-67)
- Streaming API for XML 1.0 (STAX, JSR-173)
- Java API for XML Processing 1.3 (JAXP, JSR-206, auch Teil von Java SE 5.0)
- XML Digital Signature APIs (JSR-105)
- XML Digital Encryption APIs (JSR-106)
- UDDI (Universal Description Discovery and Integration, www.oasis-open.org)

1.6 Unterstützte Produkte

Geronimo unterstützt folgende Produkte und Umgebungen. Falls die WebSphere Community Edition weitere Produkte unterstützt, ist dies in Klammern angegeben¹:

- **Java:** Sun, IBM 5.0 (nur WS CE), Java-EE-5.0-zertifiziert
- **Plattformen:** Windows XP/2003, Linux (Ix86/PowerPC64), AIX (nur WS CE)
- **Datenbanken:** Prinzipiell werden hier alle Datenbanken unterstützt, die einen JDBC-2.0-fähigen Treiber besitzen. Für ca. 34 Datenbanken sind auch bereits die korrekten Verbindungsparameter eingepflegt, und auch zum Herunterladen der jeweiligen Datenbanktreiber wird eine Unterstützung angeboten.
- **Webserver:** Jetty (Geronimo), Apache, Tomcat
- **LDAP-Server:** OpenLDAP, Apache Directory Server, Tivoli Directory Server, Microsoft Active Directory 2000

In den folgenden Abschnitten gehen wir näher auf die einzelnen Themen Java-Umgebung, Plattformen, Datenbanken, Webserver und LDAP-Server ein. Speziell werden die von Geronimo verwendeten Open-Source-Produkte behandelt.

1. Die detaillierten Systemanforderungen für WAS CE 2.1 sind unter <http://www-1.ibm.com/support/docview.wss?uid=swg27006834> beschrieben.

1.6.1 Java-Umgebung

Offiziell ist für Geronimo und WAS CE nur die Java-Version 5.0 getestet. Da auch viele Standards von Java EE diese Version erwarten, ist dies die Mindestvoraussetzung für Geronimo oder WAS CE 2.0. Jedoch kann sowohl bei Sun als auch bei IBM die schnellere JVM 6.0 jeweils verwendet werden. Sie wird aber nicht offiziell unterstützt. Da Geronimo nicht unterscheidet, ob die JVM als 32- oder 64-Bit-Version läuft, könnte sogar hier die 64-Bit-Version verwendet werden. Offiziell wird leider nur die IBM-JVM-64-Bit-Version für Linux-Intel/AMD (x-86_64) unterstützt. Für den normalen Betrieb reicht eine JVM aus. Möchte man jedoch z.B. Java-Sourcen kompilieren lassen, so benötigt man dafür ein JDK. Das ist z.B. für die Generierung der Java-Klassen aus einer WSDL und deren Übersetzung der Fall. Für die Übersetzung von JSPs gilt das unter Tomcat 5.5 nicht, da dieser bereits intern den Eclipse-JDT-Kompiler (`org.apache.jasper.compiler.Compiler`) einschließt. Dieser wird auch von Jetty 6 eingesetzt, sodass auch dafür kein JDK nötig ist. Bei beiden Webcontainern besteht die Möglichkeit, alternative Kompiler wie z.B. JAVAC aus dem JDK, Jikes oder den GlassFish-JSP-Compiler zu nutzen.

Da in den Geronimo-Versionen 1.x als ORB der in der JVM enthaltene ORB verwendet wurde, war das neben der angepassten XML-Parser-Version einer der Hauptgründe, warum man bei Geronimo an die JVM von Sun und bei WAS CE an die JVM von IBM gebunden war. Bei der aktuellen Geronimo- und WAS-CE-Version sind mir bisher keine solchen Einschränkungen bekannt. Obwohl beispielsweise die Variante WAS CE mit der Sun JVM 6.0 nicht offiziell unterstützt wird, funktioniert sie bisher ohne Probleme.

1.6.2 Unterstützte Plattformen

Bei den Betriebssystemen und Prozessorarchitekturen unterstützt Geronimo eine breite Auswahl.

Entwicklungsumgebung

Als JDK wird sowohl das von Sun als auch das von IBM (WAS CE) ab Version 5.0 unterstützt, allerdings als 64-Bit-Variante nur das IBM SDK. Je nach JDK können Intel-, AMD-, PowerPC- oder SPARC-Prozessoren als Ablaufumgebung verwendet werden.

Betriebssysteme

Die am weitesten verbreiteten Betriebssysteme werden in mehreren Varianten unterstützt.

Dies sind für Linux: Asianux Server Version 3 (x86-32, x86-64), RedHat Enterprise Linux Version 4 Update 6 bzw. Version 5 Update 1 (x86-32, x86-64, System p), Novell Suse Linux Enterprise Server, Version 9 SP 4 bzw. 10 SP1 bzw. Desktop 10 (x86-32, x86-64 und System

p), RedHat Fedora 8, 9, OpenSuse 10.3 sowie Ubuntu 6.06 LTS, 6.10, 7.04, 7.10 und 8.04. Für Windows sind dies: XP Pro SP2, 2003 Server SP1 sowie 2003 Server x64 SP2. Darüber hinaus gibt es Unterstützung für AIX Version 5 Update 3 ML4, 6 Update 1 sowie Solaris 10 oder höher. Geronimo wurde zusätzlich noch mit Mac OS X getestet.

1.6.3 Kompatible Datenbankmanagementsysteme

Die minimale Voraussetzung für die Unterstützung einer Datenbank in Geronimo ist das Vorhandensein eines geeigneten JDBC-Treibers im Klassenpfad. Möchte man jedoch auch eine Datenquelle anlegen, so verwendet man dafür einen generischen oder spezifischen TranQL-Ressourcen-Adapter-Konnektor.

Die Ressourcen-Adapter-Konnektoren für Apache Derby, IBM DB2, MySQL, PostgreSQL und Oracle sind sowohl für einfache als auch für XA-konforme Transaktionen vorhanden. Zu ihren speziellen Eigenschaften gehören Lastverteilung und Ausfallübernahme. Möchten Sie diese auch für jene Datenbanken einsetzen, die nur einen generischen Adapter verwenden, so benötigen Sie dazu den speziellen C-JDBC-Treiber.

Falls keine Treiber von Geronimo mitgeliefert werden, stehen einige zum automatischen Download bereit. Für HSQLDB, MaxDB, McKoi, MySQL, MS SQL jTDS, Sybase und PostgreSQL können die Datenbanktreiber aus dem Internet geladen werden. Bei den anderen JDBC-Treibern sind zumindest die Links zum Herunterladen angegeben, sodass Sie diese separat herunterladen und in das Repository kopieren müssen, bevor Sie diese verwenden können.

Sollte kein datenbankspezifischer Ressourcen-Adapter-Konnektor vorhanden sein, besteht immer noch die Möglichkeit, den generischen Ressourcen-Adapter zu verwenden, auch wenn dieser keine spezifischen Datenbankeigenarten unterstützt.

Offiziell unterstützt Geronimo 2.0 folgende Datenbankversionen und -systeme:

- Apache Derby 10.2.2.0
- IBM DB2 8.1 ab Fixpack 11, IBM DB2 8.2 ab Fixpack 4 sowie IBM DB2 9.1 und 9.5
- MS SQL Server 2000 ab SP 3a, MS SQL Server 2005 ab SP 1
- MySQL Community Server Edition ab Version 4.1.10a bzw. ab Version 5.0.41
- Oracle 10g Release ab der Version 10.2.0.1.0 auch mit RAC-Unterstützung
- PostgreSQL ab Version 7.3 bzw. ab Version 8.0

Die Datenbank, die am besten und komfortabelsten unterstützt wird, ist Apache Derby. Sie wird neben dem JDBC-Treiber bereits als Teil von Geronimo mitgeliefert.

1.6.4 Kompatible Webserver

Zur Ausführung von Servlets und für die darauf aufbauenden Webstandards wie JSP oder JSF benötigt Geronimo einen Webcontainer. Hier unterstützt Geronimo sowohl Apache Tomcat als auch Jetty als Ablaufumgebung. Am meisten wird hier sicher der Tomcat-Container (genannt Jasper) verwendet. Er unterstützt auch die optimale Integration in den Apache-HTTP-Server mit dem JK-Konnektor. Dieser kann dann bei besonderen Anforderungen an Sicherheit, Performance oder Ausfallsicherheit auch als Cluster und Lastverteiler eingesetzt werden.

Mit Geronimo 2.0 können die aktuellen Versionen Tomcat 6, Jetty 6, Apache 1.x und 2.x genutzt werden.

1.6.5 Kompatible LDAP-Server

Bei LDAP, dem Lightweight Directory Access Protocol, handelt es sich sowohl um ein Protokoll als auch um einen Standard zum Aufbau eines Verzeichnisdienstes. Er wird durch einen Server zur Verfügung gestellt. Da es trotz Standards je Produkt Abweichungen geben kann, wäre es wichtig, dass Geronimo möglichst viele Verzeichnisserver unterstützt. Leider tut er das jedoch offiziell nur für den eigenen Apache Directory Server.

Für den WAS CE 2.0 sind folgende Server freigegeben, die gerade für die Integration in Unternehmensinfrastrukturen relevant sind:

- Apache Directory Server 1.0
- IBM Tivoli Directory Server 5.1, 5.2, 6.0
- Microsoft Active Directory 2000
- Open LDAP Server 2.2
- Sun One Directory Server 5.2

1.7 Dokumentation

Geronimo-Wiki

Die Dokumentation des Geronimo ist zurzeit nur online als HTML in der Form eines Wikis unter <http://cwiki.apache.org/geronimo> verfügbar. Neben allgemeinen Informationen zum Projekt finden sich hier Beispiele sowie eine Benutzer- und Entwicklerdokumentation für die jeweilige Version. Dabei ist das Wiki wie folgt aufgebaut:

Es gibt einen Bereich »Available Releases«, der sich nach Versionen geordnet mit der Benutzung des Geronimo beschäftigt. Für die Geronimo-Version 2.0 heißt der Bereich z.B. GMOxDOC20 (User's Guide). Neben diesen Dokumentationen für Benutzer finden sich hier außerdem zwei Dokumentationen für Entwickler (Developer's Guide), die allerdings nur für die Version 1.0 und 1.1 gültig sind². Einige Dokumentationen sind auch in Spanisch, Französisch, Polnisch und Japanisch verfügbar.

Ein zweiter Bereich des Wiki trägt die Überschrift »Development« und richtet sich an die Entwickler und an Architektur-Interessierte. Dieser Bereich heißt unabhängig von der Geronimo-Version GMOxDEV. Hier werden sowohl Konzepte von Geronimo vorgestellt als auch Vorschläge für eigene Anpassungen gemacht.

Für beide Bereiche – »Available Releases« und »Development« – gibt es jeweils eine dazugehörige Benutzer-Diskussionsgruppe, an der man sich beteiligen kann.

Der dritte Wiki-Bereich betrifft die Projektleitung (»Project management«). Dieser Bereich heißt unabhängig von der Geronimo-Version GMOxPMGT. Hier werden die Protokolle der monatlichen Planungssitzung (ASF Board), Projektstatusberichte zur Entwicklerversion und der Releaseplan beschrieben.

Ein weiterer Wiki-Bereich ist die Geronimo-Wissensdatenbank (GMOxKB-FAQ) für häufige Fragen und Antworten. In der Zukunft sollen die Beispiele nicht mehr unter GMOxDOCNN, sondern je nach Version unter GMOxSAMPLES gesammelt werden.

Zuletzt wird GMOxSBOX genannt, die sich mit experimentellen Erweiterungen, veralteten Dokumenten oder Vorschlagsdokumenten beschäftigt.

Für jede Version werden auch aktualisierte Beispiele angeboten. Obwohl die Geronimo-Beispiele aktueller sind, empfiehlt es sich hier, die Beispiele von WAS CE zu benutzen, da diese nicht einzeln heruntergeladen, übersetzt oder angepasst werden müssen.

Gut strukturiert und online durchsuchbar ist das WAS-CE-Infocenter (<http://publib.boulder.ibm.com/wascel>). Neben einer Übersicht bietet das Infocenter Informationen zur Installation und Verwaltung des Servers bzw. zur Entwicklung von Anwendungen. Als einzige derzeit aktuelle Dokumentation zu Geronimo kann dieses zum Offline-Lesen auch heruntergeladen werden. Darüber hinaus stellt IBM über »developerWorks« aktuelle Artikel und Tutorials mit Codebeispielen für Geronimo zur Verfügung.

WAS-CE-Infocenter

2. Für Geronimo 2.1 wird der Developer's Guide aktuell erstellt:
<http://cwiki.apache.org/GMOxDOC21/developers-guide.html>

Fehlermeldungen

Um den Stand bereits gemeldeter Fehler zu beobachten oder nach Informationen zu deren Behebung zu suchen, können Sie unter <https://issues.apache.org/jira/browse/GERONIMO> nachschauen.

1.8 Support

Neben der Dokumentation ist eine Unterstützung bei Fragen oder Problemen mit Geronimo wichtig. Neben den kostenpflichtigen Angeboten gibt es, wie bei Open-Source-Produkten üblich, Unterstützung in Form von Newsgruppen und einem Wiki.

Newsgruppen

Bei Entwicklungsproblemen sind die Newsgruppen der Community die beste Möglichkeit, schnell und unbürokratisch Hilfe zu bekommen.

Die wichtigsten Newsgruppen für Geronimo bzw. WAS CE sind:

- Entwicklerliste:

<http://www.mail-archive.com/dev@geronimo.apache.org/>

- Benutzerliste:

<http://www.mail-archive.com/user@geronimo.apache.org/>

- Das Nabble-Forum fasst diese Entwickler- und Benutzerliste zusammen. Außerdem kann im Netz ein Beitrag direkt erfasst werden: <http://www.nabble.com/Geronimo-f134.html>

- Forum für WAS CE und Geronimo: <http://www.ibm.com/developerworks/forums/forum.jspa?forumID=541>

- Änderungen am Quellcode, Build-Nachrichten:

http://mail-archives.apache.org/mod_mbox/geronimo-scm/

Da bei Geronimo sowohl für Benutzer als auch für Entwickler Newsgruppen existieren, die auch größer und lebendiger als das WAS-CE-Forum sind, sollte man dort zuerst nach einer Lösung für sein Problem suchen, bevor man dort seinen eigenen Beitrag platziert. Bei WAS-CE-spezifischen Fragen ist wiederum das dafür vorgesehene Forum hilfreicher.

Online-Dokumentation,
Wiki und developerWorks-
Artikel

Oft hilft einem die z.T. sogar mehrsprachig vorhandene Online-Dokumentation weiter. Eine gute, wenn auch z. T. lückenhafte Informationsquelle ist das Geronimo-Wiki. Besser strukturiert und online durchsuchbar ist das WAS-CE-Infocenter. Darüber hinaus stellt IBM über developerWorks aktuelle Artikel und Tutorials mit Codebeispielen zu Geronimo zur Verfügung (siehe Abschnitt 1.7)

Kostenpflichtiger Support

Für den Betrieb eines Servers ist darüber hinaus ein professioneller Support von großer Wichtigkeit. Für WAS CE und Geronimo bietet IBM selbst und über Partner einen kostenpflichtigen Support (24x7h) mit vereinbarten Servicezeiten und einer festen Anzahl von Support-

anfragen in drei Stufen (Entry, Enhanced, Elite) pro Server (mit bis zu 4 CPUs) und Jahr an³. Die Stufen beinhalten folgende Leistungen, die sich im Wesentlichen in den Bereitschafts- und Antwortzeiten sowie in der Art und Anzahl der Anfragen unterscheiden:

- Entry Support: Der Basistarif beinhaltet eine Bereitschaftszeit von Montag bis Freitag 8 bis 17 Uhr und eine Antwortzeit von acht Stunden. Anfragen können nur elektronisch gestellt werden. (Zurzeit 969 Euro ohne MwSt.)
- Enhanced Support: Als Erweiterung zum Basistarif beinhaltet er dieselbe Bereitschaftszeit mit einer auf vier Stunden verkürzten Antwortzeit. Es können auch telefonische Anfragen gestellt werden, zusätzlich auch zu Entwicklungs- und Performance-Problemen. (Zurzeit 3.499 Euro ohne MwSt.)
- Elite Support: Die höchste Stufe beinhaltet eine weltweite 24x7 x 365-Verfügbarkeit mit einer definierten Antwortzeit von unter zwei Stunden und einer höheren z.T. unbegrenzten Anzahl von Unterstützungsanfragen. (Zurzeit 6.461 Euro ohne MwSt.)

Da der Support pro Server unabhängig von der Anzahl der darauf laufenden Anwendungen oder Instanzen abgeschlossen werden kann, ist das vor allem für Firmen interessant, die keine große Betriebsmannschaft haben oder aufbauen wollen. Um die Leistungsfähigkeit zu überprüfen, kann man diesen Support auch 30 Tage kostenlos in Anspruch nehmen. Unabhängig davon garantiert IBM für WAS CE zwei Versionen pro Jahr und eine Unterstützung von minimal 18 Monaten. Ein so definierter und bisher auch eingehaltener Releaseplan bietet eine Planungssicherheit, die bei Open-Source-Server-Produkten selten vorhanden ist. Anders als bei kommerziellen Produkten wird man jedoch nicht zu einem Releasewechsel gezwungen. Trotzdem werden Fehlerkorrekturen nur noch für 18 Monate an Vorversionen vorgenommen. Benötigt man diese in älteren Versionen, so muss man dort selbst die Korrekturen vornehmen oder auf eine aktuelle Version wechseln, um diese in Zukunft einfacher zu erhalten.

Es gibt zudem auch andere Geronimo-Partner, wie z.B. Covalent, die einen ähnlichen Support anbieten.

Erste Anlaufstelle für den Geronimo-/WAS-CE-Support ist folgende Seite:

Erste Anlaufstelle

<http://www-306.ibm.com/software/webservers/appserv/geronimo/support/>

3. *<http://www-01.ibm.com/software/webservers/appserv/community/detail/table.html>*

Hier können Sie die offiziellen Troubleshooting Technotes, FAQs, APAR-Fehlerberichte (Authorized Program Analysis Reports), Red-books, White Papers oder Newsletter für IBM WAS CE durchsuchen.

Der Stand der aktuellen Fehler in Geronimo wird im JIRA-Fehlerverwaltungssystem festgehalten, das über <http://issues.apache.org/jira/browse/GERONIMO> angefragt werden kann.

1.9 Konkurrenten

Geronimo ist nicht nur für die Platzhirsche von BEA, IBM und JBoss eine Konkurrenz, sondern muss sich z.B. auch mit seinem Open-Source-Altersgenossen GlassFish von Sun messen lassen. Im Folgenden wollen wir kurz die Konkurrenz vorstellen und deren Unterschiede beschreiben.

Lässt sich eine Marktverteilung bei kommerziellen Applikationsservern über den Umsatz leicht ermitteln, so ist das bei Open-Source-Produkten schwieriger. Bei den kommerziellen Applikationsservern hält IBM größere Marktanteile als BEA. Da Apache Tomcat auch in vielen Open-Source-Applikationsservern wie JBoss und Geronimo verwendet wird, sind die Installationszahlen von Tomcat hier am höchsten. Dieser liegt insbesondere aus historischen Gründen vor seinen ähnlich weit verbreiteten jüngeren Konkurrenten GlassFish und Geronimo, die jedoch mit stark steigenden Download-Zahlen glänzen, was auf großes Interesse schließen lässt.

1.9.1 Tomcat

Der Apache Tomcat war lange Zeit die Referenzimplementierung für Java Servlet und JavaServer-Pages. Sun hat dafür sowohl den Quellcode des Servers als auch der Java-API der Apache Software Foundation als Open Source zur Verfügung gestellt. Deshalb wird der Tomcat-Server von vielen Open-Source-Applikationsservern als Standardwebcontainer verwendet. Das aktuelle Java Web Services Developer Pack (JWSDP 2.0) unterstützt zwar immer noch Tomcat als Webcontainer, jedoch in einer alten und speziell angepassten Version 5.0.19. In Zukunft wird JWSDP offiziell nur noch für GlassFish zur Verfügung stehen.

Version 6.0

Tomcat ist in seiner neuesten Version 6.0 sehr modular und flexibel erweiterbar aufgebaut. Dabei ist neben einer verbesserten Ausfallsicherheit und Skalierbarkeit (NIO-Konnektor) die Unterstützung der Spezifikationen für Java Servlet 2.5 und JavaServer Pages 2.1 sowie für das JSDK 6.0 hinzugekommen. Tomcat bleibt auch in Zukunft als fle-

xibler und skalierbarer Webcontainer die erste Wahl. Er ist jedoch bewusst kein Applikationsserver, obwohl er in einigen Bereichen, wie z.B. Web- und Webservice-Anwendungen mit diesen konkurriert. Hinsichtlich komfortablen Verwaltens und Überwachsens der Anwendungen fehlen hier leider Funktionen. Tomcat steht in mehreren getrennten Paketen zur Verfügung (Embedded, Server, Dokumentation, Webadministrationskonsole, Loadbalancer, APR). Es setzt stark die Jakarta-Commons-Bibliotheken ein und verträgt sich gut mit anderen Apache-Produkten.

1.9.2 JBoss

JBoss wurde 1999 als kommerzielle Firma gegründet, um einen Open-Source-Applikationsserver unter der GNU Lesser General Public License (LGPL) zu entwickeln und um mit passendem Support und Service Geld zu verdienen. Das Geschäftsmodell erinnert ein wenig an RedHat oder MySQL. JBoss 4.0 war der erste Open-Source-Applikationsserver mit einer J2EE-1.4-Zertifizierung (EJB 2.1). Für die Vorgängerversion 3.x der aktuellen Version 4.x wurde keine eigene J2EE-1.3-Zertifizierung (EJB 2.0) durchgeführt. Für die aktuelle Java-EE-5.0-Spezifikation gibt es sowohl eine Version 4.2 als auch eine Version 5.0.

Durch die Übernahme von JBoss durch RedHat im Jahr 2006 hat sich jedoch das Geschäftsmodell geändert: Der Quellcode und die Entwicklungsversionen werden immer noch kostenlos zur Verfügung gestellt. Die offiziell unterstützten und qualitätsgesicherten Versionen sind aber nur noch über ein kostenpflichtiges Abonnement erhältlich, ähnlich wie bei RedHat Linux.

JBoss ist in Bezug auf seine Konzepte sehr innovativ. So verwendet er wie Geronimo eine JMX-Architektur mit einem Mikro-Kernel. Für JBoss 5.0 wurde die JMX-Architektur stärker modularisiert und auf PlainOldJava-Objekte (POJOs) umgestellt, um so den Server mit verschiedenen Profilen starten zu können oder ihn auch einfacher um neue Funktionalitäten zu erweitern.

Als Webcontainer können Jetty oder Tomcat integriert werden, die beide auch ein Hot-Deployment unterstützen. JBoss bot als Erster einen vollständig in Java implementierten Anwendungsserver mit Cluster und Fail-over an. Er kann in sieben festen Varianten (21 MB bis 73 MB) installiert werden, die jedoch nicht so flexibel und dynamisch wie bei Geronimo angepasst werden können. Außerdem hinkt der JBoss-Applikationsserver im Bedienungskomfort und den Automatisierungsmöglichkeiten Geronimo hinterher, da sich hier RedHat mehr auf die Zusatzprodukte konzentriert.

1.9.3 GlassFish

Der Open-Source-Anwendungsserver GlassFish wird unter zwei alternativen Lizenzen veröffentlicht; der Common Development and Distribution License (CDDL) oder der GPL-Lizenz, wobei einige Teile der Java-EE-Bibliotheken unter der Suns Binary Code License (BCL) veröffentlicht sind. Deshalb musste das Geronimo-Projekt z.B. auch eine eigene quellfreie Java-EE-Implementierung erstellen, da weder CDDL noch BCL mit der Apache-Lizenz kompatibel sind. Sun ist jedoch bestrebt, immer mehr Teile von JSE und Java EE nicht nur unter den eigenen restriktiven Lizenzen CDDL bzw. BCL, sondern auch unter der freieren GPL-Lizenz zu veröffentlichen. So werden in Zukunft immer mehr Teile von Java unter dieser Lizenz zur Verfügung stehen.

Für Java EE 5.0 zertifiziert

GlassFish ist nicht nur für die derzeitige, sondern auch für die zukünftigen Java-EE-Spezifikationen die bevorzugte Plattform. So ist GlassFish der erste Open-Source-Anwendungsserver, der für Java EE 5.0 zertifiziert ist.

Seine Architektur ist JMX-basiert mit eigenen Erweiterungen (Appserver Management Extensions, AMX) und erfüllt die Java-EE-Management-Spezifikation nach JSR-77. GlassFish verwendet eine angepasste Tomcat-Version als Webcontainer, der um einen eigenen HTTP-Server ergänzt wurde, der auf dem Grizzly-Framework beruht und auch NIO-Sockets unterstützt.

Die Referenzimplementierung des JSR-220 (EJB-3.0-Persistenz) beruht auf TopLink Essentials, die auch als Open Source unter GPL verfügbar ist.

Da es sich bei GlassFish bzw. der kommerziellen Variante Sun Java System Application Server (SJAS, auch als GlassFish Enterprise bezeichnet) um die Java-EE-Referenzimplementierung handelt, gibt es sehr viele Beispiele, Handbücher und Tutorials für die Entwicklung und die Administration. Außerdem besteht eine breite Unterstützung von Frameworks und Nicht-Java-Sprachen für GlassFish. Für GlassFish liegt, ähnlich wie für die WebSphere Application Server Community Edition, sogar ein offizieller SPECjAppServer-2004-Benchmark (<http://www.spec.org/jAppServer2004/results>) vor.

Version 3.0 ab 2009

Die auf die derzeit aktuelle Version 2.0 folgende Version 3.0 soll, ähnlich wie bei Little-G, aus einem sehr kleinen Kern HK2 (100 KB) bestehen, der sehr flexibel auch außerhalb von Java einsetzbar wäre, z.B. für Sprachen wie JRuby und PHP. Diese für das kommende Java EE 6.0 vorgesehene Version ist für 2009 angekündigt und durch ihren modularen Aufbau bereits für die darin geplante Unterstützung von Java-EE-Profilen vorbereitet. GlassFish hat eine komfortable Administrationsoberfläche und erfüllt die Ansprüche an eine hohe Ausfall-

sicherheit und Skalierbarkeit. Auch wenn administrative Tätigkeiten über eine Skriptschnittstelle automatisierbar sind, fehlt ihm allerdings ein durchgängiges einheitliches Deploymentkonzept, wie es Geronimo hat.

1.9.4 Kommerzielle Applikationsserver

Kommerzielle Applikationsserver sind und waren maßgeblich an der Verbreitung des Java-EE-Standards beteiligt. Sie haben bei der Realisierung ihrer Applikationsserver ihr Wissen und ihre Erfahrung mit verteilten Anwendungen und Servern eingebracht. Daher besitzen diese Produkte auch eine hohe Reife. Durch die Standardisierung ist es prinzipiell einfacher geworden, auch zwischen den Produkten zu wechseln, solange man die Standards einhält. Die kommerziellen Applikationsserver von BEA, IBM oder SAP spielen jedoch – auch preislich – in einer anderen Liga als Apache Geronimo.

Am besten verträgt Geronimo sich noch mit der IBM-Verwandtschaft, dem WebSphere-Applikationsserver, da IBM selbst den WAS CE als Test- und Entwicklungsumgebung unterstützt und auch eine Migration durch Informationen und Werkzeuge erleichtert.

WebSphere

Auch mit BEA gibt es einige Gemeinsamkeiten. Da BEA selbst Open-Source-Bibliotheken in seinen Produkten einsetzt und zur Verfügung stellt, verwendet z.B. Geronimo die ursprünglichen BEA-Bibliotheken XMLBeans und OpenJPA (vormals als Kodo bekannt). Gespannt darf man darauf sein, wie sich die Übernahme von BEA durch Oracle auf die Weiterentwicklung des BEA-Weblogic-Applikationsservers auswirkt.

BEA

Architekturtechnisch sind die genannten kommerziellen Applikationsserver für den Betrieb großer Installationen optimiert, sodass sie für die Entwicklung eher schwerfällig sind. Außerdem hinken diese Produkte, was die Unterstützung aktueller Standards und Frameworks betrifft, immer noch den Open-Source-Alternativen hinterher. Da viele Firmen auch die auf dem Applikationsserver des gleichen Herstellers beruhenden Zusatzprodukte, wie Portal- oder Prozessserver einsetzen, ist ein Wechsel auf einen anderen Applikationsserver dafür kaum möglich. Eine Firma wird hier zumindest in der Produktion versuchen, die Versions- und Produktvielfalt auf das Nötigste zu beschränken, um die Administrationskosten zu reduzieren und das erworbene Wissen besser zu nutzen.

Vor- und Nachteile

Trotzdem können Open-Source-Applikationsserver immer noch in der Entwicklungsumgebung eine Alternative darstellen, um hier effizienter, kostengünstiger und flexibler zu sein. Da der Applikationsser-

ver-Markt zurzeit in Bewegung ist, werden hier nur wenige große Anbieter übrigbleiben. Für Kunden, die ihre bestehenden Anwendungen auch in Zukunft einsetzen wollen, stellt sich die Frage, welcher Server diese Aufgabe in den nächsten Jahren am besten und kostengünstigsten erledigen kann. Da eine große Versionsaktualisierung ähnlich aufwendig wie eine Migration sein kann, bietet hier Geronimo eine interessante Alternative. Die Verbreitung von Apache-Software wird auch durch den vermehrten Einsatz von Linux als Serverbetriebssystem gefördert. Neben Windows ist Linux eine der meistgenutzten Serverplattformen von Apache Geronimo. Bei einigen Linux-Distributionen wie Suse wird Geronimo bereits als Software mitausgeliefert. Bei RedHat-Linux steht er ein wenig im Schatten des RedHat-JBoss-Applikationsservers.

1.10 Zusammenfassung

Mit Geronimo steht nach GlassFish ein zweiter zertifizierter Java-EE-5.0-Open-Source-Applikationsserver zur Verfügung. Auch wenn Apache Geronimo durch sein junges Alter noch nicht mit großen Installationszahlen glänzen kann, zeigen Performance-Tests, dass er durchaus für produktive Anwendungen ein Potenzial hat.

Besonders interessant für Administratoren und Entwickler sind sein flexibler Konfigurationsmechanismus und das automatisierte Deployment. Dadurch ist es möglich, Geronimo einfach und schnell dynamisch zu erweitern und anzupassen. Wer nicht auf den vollständigen Java-EE-5-Stack setzen möchte, kann sich auch der abgespeckten Implementierung Little-G bedienen. Nicht nur bei der Unterstützung mehrerer Webservices-Implementierungen, sondern auch bei den Webcontainern bietet Geronimo ein hohes Maß an Wahlfreiheit. Gegenüber seinen Open-Source-Konkurrenten punktet er durch die gute Integration in die weitverbreiteten und ausgereiften Apache-Frameworks sowie mit der sehr liberalen Apache-Lizenz. Durch eine verlässliche Releasepolitik verspricht Geronimo eine langfristige Planungssicherheit. Als Neuling bietet Geronimo in dem gerade sich konsolidierenden Applikationsserver-Markt Beispiele und Werkzeuge, die den Umstieg erleichtern. Auch was Dokumentation, Support und Entwicklungsunterstützung anbelangt, braucht er sich nicht hinter seiner Konkurrenz zu verstecken. Gerade gegenüber den kommerziellen Produkten bietet er eine günstige und verlässliche Alternative. Die Erweiterung der Serverfunktionalität zur Laufzeit, das komplett automatisierbare Deployment oder die Verwendung mehrerer Instanzen sind Alleinstellungsmerkmale von Apache Geronimo.