Java Plattform Strategie

Wolfgang Weigend, Peter Doschkinow

ORACLE Deutschland B.V. & Co. KG

Bestandteile der Java-Plattform, OpenJDK, JDK 7, JDK 8, JDK 9, Entwicklungsumgebungen, Open Source Community, JavaFX, Java EE 7, Java EE 8

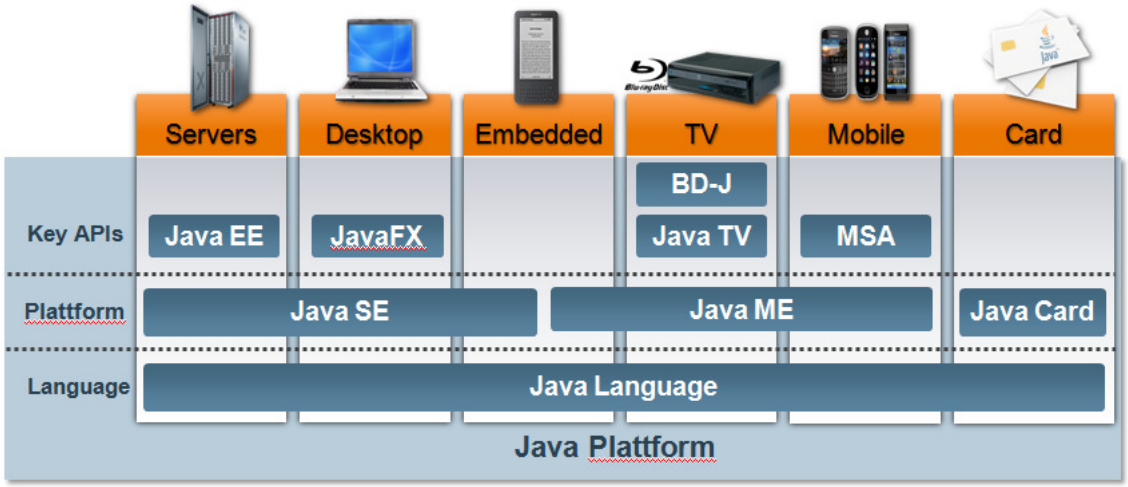
Java ist seit mehr als 1,5 Jahrzehnten im unternehmensweiten Einsatz und gerade jetzt sind die Firmen dabei das JDK 7 zu verwenden. Da steht bereits das JDK 8 mit größeren Veränderungen vor der Tür. Projekt Lambda zeigt konkrete Ansätze, die konvergierende JVM macht Fortschritte, die JavaScript Interoperabilität mit der JVM und weitere kleinere Sprachverbessungen runden die Änderungen ab. Doch damit nicht genug, JavaFX ist nach dem Start vom OpenJFX Projekt auf dem Sprung ins JDK 8 und wird sich dort als JavaFX 8 für Rich Internet Anwendungen manifestieren. Im Vortrag wird die gesamte Java Plattform Strategie besprochen, vom OpenJDK, Java SE 7, Java SE 8, Java SE 9 bis zur Java Enterprise Edition und die Zusammenarbeit mit der Java Community betont. Es geht im folgenden um die nahe und ferne Zukunft der Java-Sprache und der Java-Plattform.

Wie sich Java schrittweise verändert

Der Vortrag beginnt mit der Java Plattform und dem bestehenden Ökosystem und unterstreicht die hohe Verbreitung der Java-Technologieplattform mit weltweit über 9 Millionen registrierten Java Entwicklern, 930 Millionen Java Runtime Environment (JRE) Downloads pro Jahr, mehr als 1.1 Milliarden Java Desktops, 3 Milliarden mobilen Endgeräten mit Java und 1.4 Milliarden Java Cards die Jahr für Jahr hergestellt werden. Java führt die Liste der am häufigsten verwendeten Programmiersprachen an und ist in allen Schulen und Universitäten fester Bestandteil der Ausbildung. Darin liegt auch die Stärke von Java: Die Community vergrößert sich und tauscht sich global über die Kodierung aus, sodass ohne komplexe Einarbeitung der Programmiercode gelesen, nachvollzogen und verändert werden kann. Aus diesem Grund wird der freie Zugang zu aktuellen Informationen für die Weiterentwicklung der Java Community uneingeschränkt zur Verfügung gestellt. Die einheitliche Entwicklungsbasis bildet dabei das offene und frei verfügbare OpenJDK, als zentrale Grundlage für die Aktivitäten der Java Standard Edition 7 (Java SE 7) und der Java Standard Edition 8 (Java SE 8). Java ist der technologische Ausgangspunkt der meisten Hardware- und Software-Hersteller und bildet auch die Basis für die Oracle Fusion Middleware mit Oracle Fusion Applications. Dies verdeutlicht auch das Geschäftsmodell für die Java-Entwickler, das die anhand der gelernten Programmiersprache und der frei zugänglichen Java-Technologie die von ihnen erstellte Programmierlogik in Form von Anwendungen und neuen Services in die kommerzielle Vermarktung bringt. Die Verwendung von Java in Open Source Projekten macht einen Großteil der IT-Landschaft aus, bietet doch der

kommerzielle Einsatz des Java-Programmier-Codes die Möglichkeit einer Einnahmequelle für die Entwickler. Bereits bei der Verwendung vom OpenJDK ist der Entwickler integraler Bestandteil einer klar umrissenen Java-Strategie. Die neuen OpenJDK Community Richlinien wurden in Zusammenarbeit mit IBM, Eclipse Foundation, Oracle und Einzelpersonen erstellt und führen die Arbeiten des OpenJDK Governance Board weiter. Das Gremium hat Regeln aufgestellt, die den langfristigen Bestand und das Wachstum der OpenJDK Community fördern und sicherstellen, dass die Mitglieder in klarer und offener Weise agieren und die administrative Governance nach dem Leistungsprinzip erfolgt. So wird ein hohes Qualitätsniveau für das OpenJDK erreicht. An der Weiterentwicklung vom OpenJDK sind neben Oracle große Hersteller wie IBM, Apple, SAP, HP, Twitter, Azul Systems und VMware beteiligt. Alle setzen auf die einheitliche Java Plattform, die aus der Java Sprache, der Java Virtual Machine (JVM) und den Java APIs für unterschiedliche Funktionalitäts- und Hardware-Anforderungen wie Java Enterprise Edition (Java EE), Java Standard Edition (Java SE) und Java Micro Edition (Java ME) besteht (siehe Abbildung 1).

Abbildung 1: Die Java Plattform



Java Virtual Machine und Java Development Kit

Durch die Akquisition von SUN Microsystems durch Oracle, sind die beiden Java Virtuellen Maschinen HotSpot und JRockit unter einem Dach. Die Hotspot JVM ist allgemein einsetzbar, parametrisierbar, qualitativ hochwertig und am Markt sehr stark verbreitet. Die JRockit JVM ist eine spezielle Server-JVM mit hoher Leistungsfähigkeit, die für den Serverbereich und die Oracle Fusion Middleware optimiert wurde. Mit JRockit verbessert sich das Laufzeitverhalten von Anwendungen die effizienter mit den Ressourcen: Speichergröße, Anzahl der Threads, Netzwerk, IO-Aktivitäten umgehen und durch deterministische Garbage-Collection und minimalen Pausezeiten niedrige Antwortzeiten erreichen. Zudem werden alle Bestandteile der JRockit JVM (Code Generierung, Speicher Management, Thread Management, I/O, Reflection) optimiert. Die beiden JVMs JRockit und HotSpot werden in einem mehrjährigen Entwicklungsprozess zu einer einheitlichen JVM unter dem Namen „HotRockit“ verschmolzen, die sich dann aus den besten Funktionsmerkmalen beider JVMs zusammensetzen wird. Die Arbeitsergebnisse sind inkrementell dem OpenJDK Projekt zugeflossen, wobei die JRockit Funktionalität den bisher größten Code-Beitrag zum OpenJDK darstellt.

Vorhandene JRockit basierte Produkte, wie JRockit Mission Control, JRockit Real Time, JRockit Virtual Edition bleiben separat und werden weiterhin kommerziell lizensiert. Das Java Development Kit (JDK) und Java Runtime Environment (JRE) steht weiterhin kostenfrei zur Verfügung und Oracle wird eine offene Java Implementierung (OpenJDK) dauerhaft unterstützen, die reine Open Source Komponenten beinhaltet. Mit dem zentralen Einstiegspunkt vom OpenJDK gelangen die erreichten Ergebnisse in die Java SE 7 und in die Java SE 8. Diese Java SE Versionen bieten eine höhere Entwicklerproduktivität, bessere Ausnutzung von Multi-Core Prozessoren und Unterstützung großer Hauptspeicher und Hochgeschwindigkeitsnetze. Das JDK 7 wurde fertiggestellt und auf openjdk.java.net veröffentlicht. Der Funkftionsumfang vom JDK 7 kann mit NetBeans 7.2 und GlassFish 3.1.2 getestet werden. JDK 7 umfasst im wesentlichen Sprachverbesserungen aus dem Projekt Coin, die Concurrency und Collections Updates und die Unterstützung für dynamisch typisierte Sprachen. Das JDK 7 ist seit dem 28. Juli 2011 verfügbar, JDK 8 soll im September 2013 zur Verfügung stehen. Die wichtigsten Inhalte des JDK 8 werden die Lambda-Ausdrücke (vormals Closures), die Fertigstellung der Oracle JVM Konvergenz und auch die verbesserte Interoperabilitität von Java Script mit der JVM sein. Vorab hatte sich das JCP-SE/EE-Executive-Committee in der Abstimmung mit teils deutlichen Mehrheiten für die zugrundeliegenden Java Specification Requests entschieden: JSR 334 „Small Enhancements to the Java Programming Language“, JSR 335 “ Lambda Expressions for the Java Programming Language“, JSR 336 "Java 7 SE Release Contents“, JSR 337 "Java 8 SE Release Contents. Die wichtige Java Plattform Modularisierung im Projekt Jigsaw wird voraussichtlich im JDK 9 enthalten sein, welches zwei Jahre nach der Veröffentlichung vom JDK 8 erscheinen soll.

Java Micro Edition

Oracle beginnt mit der Modernisierung der Java Micro Edition (Java ME) Plattform und arbeitet mit der Java Mobile Community gemeinsam am Projekt Java ME.next, der evolutionären Weiterentwicklung der neuen Version von Java ME. Ziel ist es, die zugrundeliegende Sprachspezifikation zu aktualisieren und moderne Geräte-/Hardware-Funktionalität wie Near Field Communication, IP Multimedia Subsystem (IMS), Sensoren, Telefonie und Lokation durch neue Java APIs besser zu unterstützen. Wie auf dem Java Client werden sowohl native Java-Anwendungen, als auch auf Web-Technologien basierende Anwendungen unterstützt.

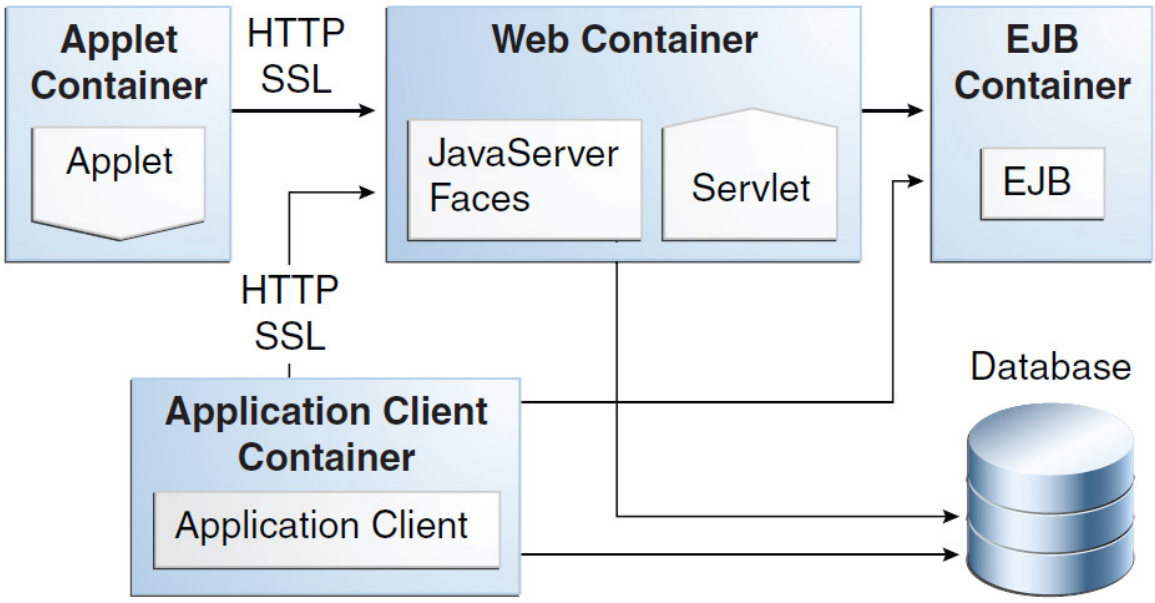
JavaFX

Für die Entwicklung von Rich Internet Anwendungen (RIA) mit Unterstützung von Multimedia und modernen Hardware-GPUs (Graphics Processing Unit) wird JavaFX angeboten. Sie bestand früher aus der JavaFX Script Sprache, den JavaFX Script APIs, den Runtime-Libraries und lief auf der Java Virtual Machine. Für die neue JavaFX Plattform kommt ein Sprachwechsel. Damit wird seit JavaFX 2.0 JavaFX Script nicht mehr fortgeführt. Die JavaFX APIs sind vollständig in Java implementiert. Damit stehen viele Vorteile der Java Plattform wie Generics, Annotations und Multithreading unmittelbar auch für JavaFX zur Verfügung. Java Programmierer können JavaFX nutzen, ohne eine weitere Programmiersprache lernen zu müssen und es wird leichter JavaFX innerhalb von Swing zu benutzen. Andere Scripting-Sprachen wie JRuby, Groovy und Scala, die auf der JVM laufen können für JavaFX Anwendungen benutzt werden und vergleichbare Merkmale wie JavaFX Script bereitstellen.

Java Enterprise Edition 6

Java EE 6 bietet neue Funktionalität mit der Einführung von Java EE Profilen für Web-Anwendungen, der Erweiterbarkeit/Plugin-Fähigkeit, „Contexts and Dependency Injection for the Java EE Platform“ (CDI), Managed Beans mit POJO-Modell, RESTful Web Services (JAX-RS), Schichtenübergreifende Validierung (Bean Validation), erweiterte APIs mit EJB 3.1, JSF 2.0, JPA 2.0, Servlet 3.0 und verbesserte Nutzbarkeit durch Konventionen anstatt Konfigurationen (weniger XML) und Annotationen basiertem Programmiermodell (decorate und inject). Damit wird Java EE 6 mit seinen Innovationen und neuen APIs (siehe Abbildung 2) leichtgewichtiger und flexibler und kann schneller von den Java Entwicklern in Projekten eingesetzt werden. Der GlassFish Anwendungsserver ist die Java EE 6 Referenz Implementierung und der WebLogic Server 12c ist seit Dezember 2011 auch für die Java EE 6 zertifiziert.

Abbildung 2: Java EE 6 API‘s



Java Enterprise Edition 7 und Java Enterprise Edition 8

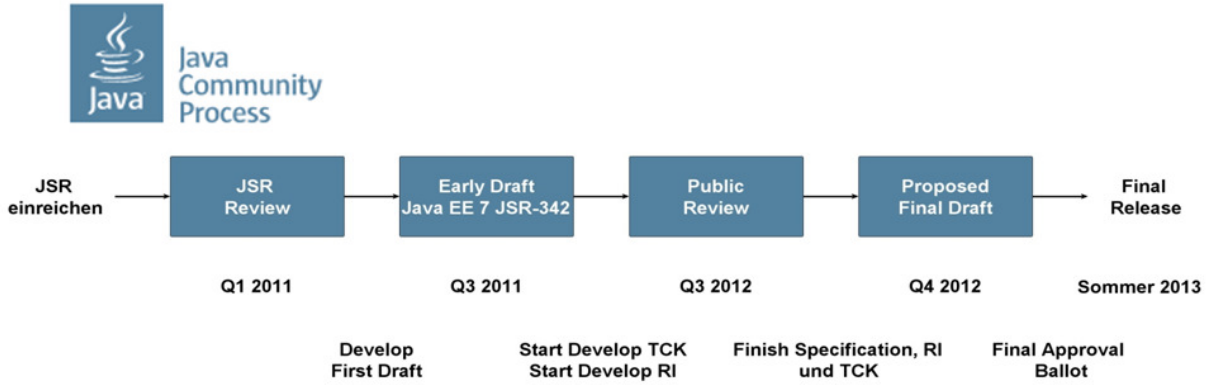
An der nächsten Version der Java Platform Enterprise Edition 7 (Java EE 7) wird intensiv gearbeitet. Alle Executive Committee Mitglieder für SE/EE haben dem JSR 342 zugestimmt, um die Inhalte von Java EE 7 zu entwickeln. Die Java EE Plattform wird sich langfristig den Anforderungen von Cloud-Umgebungen anpassen und einfacher mit Private und Public Clouds zusammenspielen. Java EE 8 wird die Funktionalität durch ein Servicekonzept unterstützen, um Mandantenfähigkeit und horizontale Skalierbarkeit (Elasticity) als kompletten Plattformdienst abzudecken. Java EE 8 Anwendungen werden damit die Vorteile von Cloud Umgebungen besser ausnutzen können. Im Cloud Platform as a Service (PaaS) Modell können verschiedene Anwendungskomponenten mit unterschiedlichem Quality-of-Service und Sicherheitsbereichen getrennt voneinander betrieben werden, wie es bei Mandantenfähigkeit notwendig ist. Bisher hatte Java EE eine Container basierte Umgebung

angeboten, die im Einzelbetrieb oder großen Cluster-Anwendungen den Zugriff auf das System oder externe Ressourcen steuerte, ohne das Programmiermodell ändern zu müssen. Hierbei wirkt der Cloud-Ansatz evolutionär, um einige inkrementelle Änderungen am existierenden Java EE Programmiermodell vorzunehmen. Die Java EE Plattform wird erweitert, um die Belange vom PaaS Betriebsmodell aufzunehmen. Dafür werden neue Rollen, wie der PaaS-Administrator eingeführt und die Java EE Security-Architektur angepaßt. Der GlassFish Applikationsserver dient auch weiterhin als Referenz Implementierung für die Spezifikationen Java EE 7 / Java EE 8 und bietet in Teilschritten einen praktischen Einblick in die technische Umsetzung der jeweiligen PaaS-Funktionalität.

Zeitlicher Ausblick und Java Technologie Spezifikationen

Mit der vorliegenden Early Draft Version der Java EE 7 kann der aktuelle Stand der Spezifikation JSR 342 jederzeit nachvollzogen werden und mit dem Jahresende 2012 kann der Übergang vom Public Review zum Proposed Final Draft verfolgt werden. Ziel ist es, das Final Releae im Sommer 2013 anzubieten (siehe Abbildung 3).

Abbildung 3: Java EE 7 Zeitplan



Zur besseren Übersicht folgt eine Auflistung der Java Technologie Spezifikationen unter dem Community Development der letzten beiden Jahre:

- *JSR 359: SIP Servlet 2.0*
- *JSR 358: A major revision of the Java Community Process*
- *JSR 357: Social Media API*
- *JSR 356: Java API for WebSocket*
- *JSR 355: JCP Executive Committee Merge*
- *JSR 354: Money and Currency API*
- *JSR 353: Java API for JSON Processing*
- *JSR 352: Batch Applications for the Java Platform*
- *JSR 351: Java Identity API*
- *JSR 350: Java State Management*
- *JSR 349: Bean Validation 1.1*
- *JSR 348: Towards a new version of the JCP*
- *JSR 347: Data Grids for the Java Platform*
- *JSR 346: Contexts and Dependency Injection for Java EE 1.1*
- *JSR 345: Enterprise JavaBeans 3.2*
- *JSR 344: JavaServer Faces 2.2*
- *JSR 343: Java Message Service 2.0*
- *JSR 342: Java EE 7 Specification*
- *JSR 341: Expression Language 3.0*
- *JSR 340: Java Servlet 3.1 Specification*
- *JSR 339: JAX-RS 2.0: The Java API for RESTful Web Services*
- *JSR 338: Java Persistence 2.1*
- *JSR 337: Java SE 8 Release Contents*
- *JSR 336: Java SE 7 Release Contents*
- *JSR 335: Lambda Expressions for the Java Programming Language*
- *JSR 334: Small Enhancements to the Java Programming Language*

Kontaktadresse:

Wolfgang Weigend

ORACLE Deutschland B.V. & Co. KG

Robert-Bosch-Strasse 5

63303 Dreieich

Telefon: **+49 (0) 6106-397-785**

Fax: **+49 (0) 6106-397-105**

E-Mail: **wolfgang.weigend@oracle.com**

Internet: www.oracle.com

Peter Doschkinow

ORACLE Deutschland B.V. & Co. KG

Riesstr. 25

D-80992 München

Telefon: +49 (0) 1802672253

Fax: +49 (0) 1802672329

E-Mail peter.doschkinow@oracle.com

Internet: www.oracle.com