

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 ME, JEE 7, JEE 8

Java ist eine der bekanntesten Marken der IT und eine der am häufigsten bereitgestellten Technologien. Sie wird durch eine Vielzahl von Anwendungen und Services repräsentiert, die in der Java-Sprache geschrieben sind. Mit der Übernahme von Sun wird Oracle mit Innovationen und Investitionen in die Java-Technologie zum Vorteil der Java-Community und der Kunden fortfahren. Oracle hat bei der Unterstützung von Java - seit der Einführung im Jahr 1995 - eine führende Rolle eingenommen und wird mit hohem Engagement und Klarheit dazu beitragen, die Java-Technologie zusammen mit der Community entscheidend voran zu bringen. Es geht im folgenden um die nahe und ferne Zukunft der Java-Sprache und der Java-Plattform.

Die Oracle Java Plattform Strategie im Java Ökosystem

Der Vortrag beginnt mit den Zahlen und Fakten zu Java 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 Milliarde 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 ein klar umrissenen Java-Strategie. Die neuen OpenJDK Community Richtlinien wurden in Zusammenarbeit mit John Duimovich (IBM), Jason Gartner (IBM), Mike Milinkovich (Eclipse), Prof. Doug Lea (State University NY Oswego) und Adam Messinger (Oracle) erstellt und führen die Arbeiten des OpenJDK Interim 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 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).

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 Pausenzeiten 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 lizenziert. 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 Funktionsumfang vom JDK 7 kann mit NetBeans 7.0.1 und GlassFish 3.1 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 Mitte 2012 zur Verfügung stehen. Die wichtigsten Inhalte des JDK 8 werden die Java Plattform Modularisierung und die Lambda-Expressions (Closures) sein. Vorab hatte sich das JCP-SE/EE-Executive-Committee in der Abstimmung mit teils deutlichen Mehrheiten für die jeweiligen 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.

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, die programmatisch untereinander kommunizieren können.

JavaFX

Für die Entwicklung von Rich Internet Anwendungen (RIA) mit Unterstützung von Multimedia und modernen Hardware-GPUs (Graphics Processing Unit) wird JavaFX angeboten. Sie besteht jetzt noch aus der JavaFX Script Sprache, den JavaFX Script APIs, den Runtime-Libraries und läuft auf der Java Virtual Machine. Für die neue JavaFX Plattform kommt ein Sprachwechsel. Damit wird in JavaFX 2.0 JavaFX Script nicht mehr fortgeführt. Die JavaFX APIs werden künftig 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 sein 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 basierendem Programmiermodell (decorate und inject). Damit wird Java EE 6 mit seinen Innovationen und neuen APIs 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 wird mit dem nächsten Major-Release ebenfalls J EE 6 zertifiziert sein (siehe Abbildung 2).

Java Enterprise Edition 7

An der nächsten Version der Java Platform Enterprise Edition 7 (Java EE 7) wird bereits gearbeitet (JSR 342). Alle Executive Committee Mitglieder für SE/EE haben dem JSR 342 zugestimmt, um die Inhalte von Java EE 7 zu entwickeln. Der Schwerpunkt vom neuen Java EE 7 Release ist die Cloud. Die Java EE Plattform wird sich besser an die Anforderungen von Cloud-Umgebungen anpassen und einfacher mit Private und Public Clouds zusammenspielen. Java EE 7 wird die Funktionalität als Service unterstützen, um Mandantenfähigkeit und horizontale Skalierbarkeit (Elasticity) abzudecken.

Java EE 7 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. JSR 342 wird die Java EE Plattform Architektur erweitern, um die Belange vom PaaS Betriebsmodell aufzunehmen. Dafür werden neue Rollen, wie der PaaS-Administrator eingeführt und die Java EE Security-Architektur angepaßt. Der JSR soll auch Voraussetzungen für Anwendungen bereitstellen, die PaaS spezifische Funktionalität wie Mandantenfähigkeit nutzen möchten, sich selbst als Cloud fähige Anwendungen ausweisen und sich an die definierten Regeln halten. Die verwendeten Plattformtechnologien müssen an die neuen Cloud-Modell Anforderungen angepaßt werden. Beispielsweise die Ressource-Manager basierten APIs wie JPA, JDBC und JMS. Das Programmiermodell soll soweit verbessert werden, dass die meisten APIs verbindungslos agieren. Die in den Anwendungen verwendeten Modell-Ressourcen Data Sources und Message Queues werden für mehrere Anwendungen als Shared gekennzeichnet. Die Expert Groups steuern die Anforderungen für die einzelnen Technologien und bewerten sie, um deren konkreten Umsetzung sicherzustellen. Der JSR definiert einen Deskriptor für Anwendungsmetadaten, damit die Entwickler ihre Anwendungscharakteristik in der PaaS-Umgebung beschreiben können. Diese Eigenschaften können wie folgt lauten: Mandantenfähigkeit, Resources-Sharing, Quality-of-Service Information, Anwendungsabhängigkeiten. Die Anwendungsmetadatenstruktur wird erweiterbar sein und soll Standard Metadatenattribute bereitstellen, die mit künftigen Java EE Plattform Versionen erweitert werden.

Ausblick

Die Java EE 7 Plattform wird aktuelle Web-Standards wie HTML 5 und Web Sockets unterstützen und ein HTTP Client API mit JAX-RS 2.0 enthalten.

Zur verbesserten Entwicklung mit Java EE 7 wird das überarbeitete JMS 2.0 API enthalten sein. Die Dependency Injection Struktur wird mit einem API zur Injector-Konfiguration erweitert, welches in DI 1.1 definiert und in CDI 1.1 supported wird.

Das Managed Bean Modell wird überarbeitet und verschiedene Inkonsistenzen, wie sie bei Managed Beans, EJB, Servlets, JSF, CDI und JAX-RS auftreten können, werden entfernt.

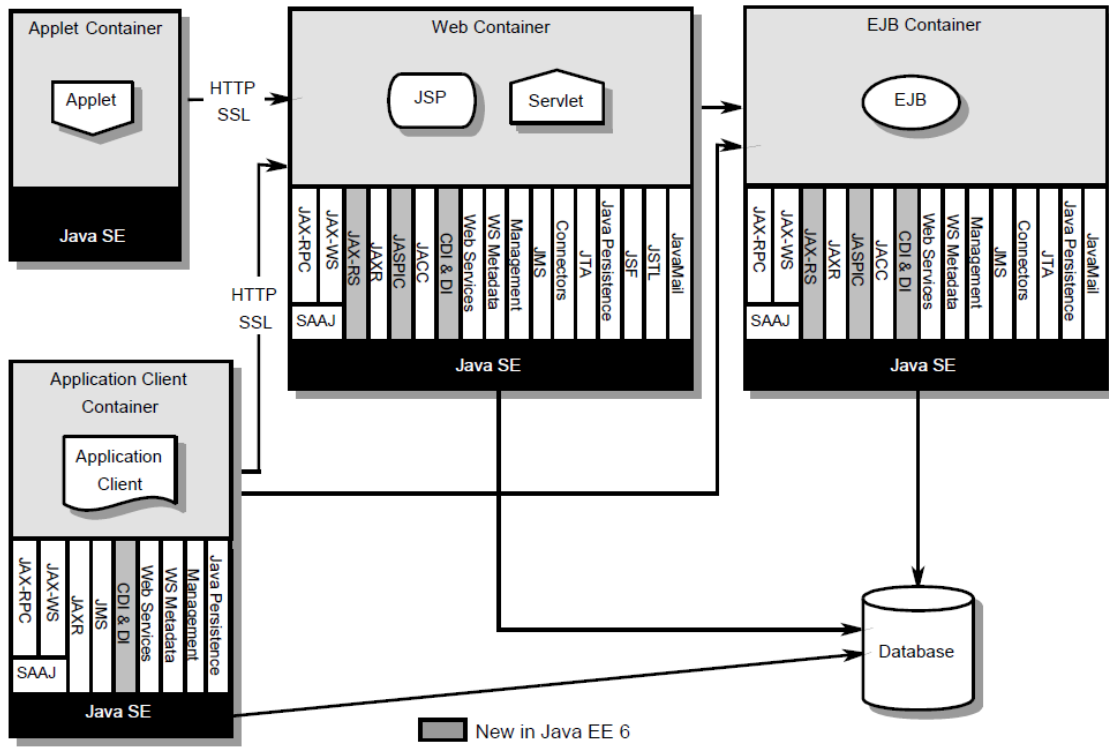
Java EE 7 wird mit Java SE 7 aufgebaut und alle beteiligten Java-Spezifikationen werden dahingehend analysiert und verbessert, um ihre Schnittstellen auf die kommenden Java SE 8 Sprachänderungen auszurichten. Die Entwickler bekommen so den größt möglichen Nutzen aus der neuen Funktionalität (siehe Java EE 7 Plattform JSRs und Technologie Übersicht).

Der Zeitplan für die Entwicklung der Java EE 7 Plattform sieht ein Early Draft im dritten Quartal 2011 vor. Dem folgt ein Public Review im ersten Quartal 2012 und ein Final Release für JSR 342 im dritten Quartal 2012.

Abbildung 1: Die Java Plattform



Abbildung 2: Java EE 6 Architektur Diagramm



Java EE 7 Plattform JSRs und Technologie Übersicht

Die folgenden Java Specification Requests und Technologien werden für die Java EE 7 Plattform verwendet und noch angepaßt:

- Concurrency Utilities for Java EE (JSR-236)
- JCache (JSR-107)
- Java Persistence API (JPA)
- Java API for RESTful Web Services (JAX-RS)
- JavaServer Faces (JSF)
- Servlets
- Enterprise JavaBeans (EJB)
- JavaServer Pages (JSP)
- Expression Language (EL)
- Java Messaging Service (JMS)
- Java API for XML-based Web Services (JAX-WS)
- Contexts and Dependency Injection for Java EE (CDI)
- Bean Validation
- Dependency Injection for the Java Platform (JSR-330)
- Common Annotations (JSR-250)
- Java Connector Architecture
- Java Web Sockets API (TBD)
- Java JSON API (TBD)

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