

Java aktuell

Praxis. Wissen. Networking. Das Magazin für Entwickler
Aus der Community – für die Community

Java aktuell



Java im Mittelpunkt

Big Data
Predictive Analytics
mit Apache Spark

Microservices
Lagom, das neue
Framework

Achtung, Audit
Nutzung und Verteilung
von Java SE

Achtung, Audit!

Markus Karg, JUG Goldstadt

Java ist Open Source, dann darf ich es doch wohl auch frei benutzen und verteilen? Weit gefehlt: Wenn ein Oracle-Auditor ins Haus kommt, sollte man Bescheid wissen! Die Fallstricke bei der Nutzung und Verteilung von Java



Auditor

Varianten von Java SE

Oracle verpackt das eigentlich freie und kostenlose OpenJDK in verschiedene Pakete, die in überwiegender Mehrzahl kostenpflichtig sind. Alle Varianten enthalten ein JRE und ein JDK sowie eine Reihe zusätzlicher Beigaben:

- Das Projekt OpenJDK bietet JRE und JDK selbst nicht als kompilierten Download an, sondern verweist auf die Produktpalette von Oracle. Es ist jedoch in Debian und anderen Distributionen enthalten. Die JVM ist weniger leistungsstark als jene, die im Oracle-Java-SE-Download enthalten ist. Daher enthält beispielsweise Raspbian (ein speziell auf den Raspberry Pi zugeschnittenes Debian-Derivat) zusätzlich das proprietäre Oracle Java SE inklusive dessen leistungsfähiger Hot Spot Virtual VM (siehe unten).
- Oracle Java Platform Standard Edition (Java SE) ist das, was allgemein als „Java Download“ bekannt ist und vom Normal-Anwender heruntergeladen wird. Es ist kostenlos (aber nicht frei!) für „general purpose computing“ unter der Oracle Binary Code License (BCL). Es darf – unter Beachtung der in der BCL

genannten Regeln – auch weitergegeben werden. Sehr wahrscheinlich ist davon auszugehen, dass Oracle mit „general purpose computing“ nur PCs und Server meint. Wie beispielsweise der Rechtsstreit mit Google über Java auf Android zeigt, sind jedoch bereits die Nutzung auf Mobiltelefonen sehr vermutlich und jede Art von Embedded-Anwendung in Gerätschaften jeglicher Couleur – Stichwort „IoT“ – sogar ganz explizit von der kostenfreien Nutzung ausgeschlossen. Für Mobiltelefone wirbt die Firma Gluon mit Oracles Segen mit der Gluon VM, einem OpenJDK-Port auf Android und iOS.

- Oracle Java SE Embedded enthält eine spezielle, für die Nutzung in SBCs und Mikro-Controllern optimierte JVM sowie Zusatzbibliotheken. Das Produkt steht unter OTNLA. Die Entwicklung damit ist kostenlos, die Weitergabe an den Endkunden ist allerdings zu bezahlen, und zwar pro Gerät, auf dem sich später die Embedded Runtime befindet.
- Oracle Java SE Advanced Desktop (optional for ISVs) sowie Oracle Java SE Advanced (optional for ISVs) enthalten die im Kasten „Kostenpflichtige Beigaben“ genannten Entwicklungs- und Admi-

nistrations-Werkzeuge und sind zunächst pro Entwickler zu bezahlen. Die Variante „Desktop“ beschneidet die Client/Server-Fähigkeit der enthaltenen Monitoring-Werkzeuge. Der Zusatz „for ISVs“ erlaubt dem Lizenznehmer darüber hinaus die Weitergabe von Java Flight Recorder und Java Mission Control sowie Advanced Management Console und Enterprise JRE Installer an Endkunden.

- Oracle Java SE Suite entspricht Oracle Java SE Advanced, enthält aber eine JVM mit Soft-Real-Time-Fähigkeit. Somit benötigt jeder Entwickler eine kostenpflichtige Lizenz.
- IBM und weitere Anbieter haben darüber hinaus eine Reihe kostenloser und kostenpflichtiger Produkte im Angebot, die wiederum proprietäre Erweiterungen gegenüber OpenJDK enthalten (etwa andere JVMs wie IBM J9 oder Unterstützung für spezielle Betriebssysteme wie z/OS).

Der geneigte Leser möge sich auf den entsprechenden Produktseiten selbst informieren, da dies sonst den Umfang dieses Artikels sprengen würde.

Kurz vor Weihnachten geriet die mediale Java-Welt in Aufruhr: Angeblich sei es dem Durchschnitts-Java-Programmierer unmöglich, sich ein JDK zu besorgen, ohne damit praktisch schon mit einem Fuß in der Oracle-Abzock-Falle zu stehen. Tatsächlich hat sich das Ganze schon bald als unbegründete Panikmache eines offenbar schlecht recherchierenden Online-Redakteurs herausgestellt. Trotzdem bleibt das flaue Gefühl, ob an der Sache nicht doch etwas dran ist. Tatsächlich: Wer nicht aufpasst, kann beim Besuch des Auditors in echte Argumentationsnot geraten. Dieser Artikel klärt auf und gibt Tipps, wie Lizenz-Risiken leicht zu vermeiden sind.

Was ist eigentlich ein Lizenz-Audit?

Viele Softwarehersteller behalten sich das Recht vor, beim Kunden zu überprüfen, ob nicht etwa mehr Anwender die Software

verwenden, als Lizenzen eingekauft wurden. Hierzu ist es, neben Selbstauskünften und Fernzugriff, teilweise auch gängige Praxis, einen Auditor vor Ort zu senden, der mit entsprechenden Werkzeugen ausgestattet die tatsächliche Nutzung im Firmennetz zählt. Diese Prüfung muss nicht immer negativ sein: Teilweise stellt sich dabei auch heraus, dass bestimmte Produkte schon lange nicht mehr im Einsatz sind und man sich deren Aktualisierung und Support künftig sparen kann. Trotzdem sind Administratoren und Geschäftsleitung gut beraten, mit solchen Prüfungen zu rechnen und ein ordentliches Lizenz-Management einzuführen. Denn je nach Grad des Verschuldens können unter Umständen auch strafrechtliche Konsequenzen drohen.

Steht der Auditor erst einmal vor der Tür, ist es meist zu spät. Eine Nutzung nicht lizenzierter Software lässt sich schwer begründen und ist kein Kavaliersdelikt. Ehr-

lichkeit ist sicherlich die bessere Lösung: Wer Tools verwendet, die Geld kosten, sollte die typischerweise sowieso moderaten Kosten nicht scheuen, sondern nutzenorientiert denken und aktiv entsprechende Lizenzen erwerben.

Leider hat es Oracle einem bislang sehr leicht gemacht, sich nicht lizenzierte Software mit dem JDK- und JRE-Download auf den Rechner zu ziehen. Das Gute ist, dass das kein Problem darstellt – solange man diese Software nicht regelmäßig verwendet. Doch um welche Software geht es dabei überhaupt und wieso ist im JRE nicht-lizenzierte Software drin?

Die Sache ist eigentlich ganz einfach. Sun Microsystems hat das JRE (also die von Sun entwickelte HotSpot JVM sowie die Standard-Bibliotheken, die zu Java SE zählen) als Open Source freigegeben und, wie auch Oracle heute noch, auf jegliche Lizenzgebühren verzichtet. Jeder darf diese freien

Bestandteile im Rahmen der vorliegenden Lizenzbedingungen nutzen, ändern und weitergeben. Sie werden durch das Projekt „OpenJDK“ verwaltet und weiterentwickelt und sind beispielsweise in Debian wie auch in vielen anderen Linux-Distributionen enthalten. Wer also nur das OpenJDK verwendet, ist immer auf der sicheren Seite.

Oracle verteilt jedoch kein reines OpenJDK. Um den Entwicklern, Endanwendern und Administratoren das Leben zu vereinfachen, hat Oracle weitere Werkzeuge entwickelt beziehungsweise eingekauft (unter anderem durch die Übernahme der JRockit-Produktfamilie). Einige dieser Features und Werkzeuge sind in den Open-Source-Zweig eingeflossen, andere sind weiterhin Closed-Source, werden jedoch kostenfrei verteilt, und wiederum andere sind und bleiben kostenpflichtige Zusatzprodukte. Es wäre nett gewesen, Oracle hätte diese in einen separaten Download verpackt. Leider ist dem nicht so.

Der Original-Download des JRE und des JDK enthält daher kostenpflichtige Zusatzprodukte, die der Auditor solange ignoriert, wie sie nicht nachweislich verwendet wurden. Startet man diese allerdings regelmäßig, wird der Kaufpreis fällig! Daher ist es wichtig zu wissen, welche Tools kostenpflichtig sind und welche nicht.

Kostenpflichtige Beigaben

Oracle verteilt unter dem Namen „Java SE“ OpenJDK mit weiteren Beigaben. Diese sind kostenpflichtig und sollten nur dann regelmäßig genutzt werden, wenn entsprechende Lizenzen erworben wurden:

- Java Mission Control
- Java Flight Recorder
- Java Advanced Management Console
Microsoft Windows Installer (MSI)
Enterprise JRE Installer
- Java Usage Tracker

Nähere Informationen sind im Web zu finden, unter anderem in der Oracle-Produktübersicht (siehe „<https://www.oracle.com/de/products/index.html>“) und in der Java SE FAQ (siehe „<http://www.oracle.com/technetwork/articles/javase/faqs-jsp-136696.html>“).

Es besteht grundsätzlich keine Gefahr, wenn die normalen Downloads für Java SE JRE beziehungsweise Java SE JDK genutzt werden und man von den kostenpflichtigen Beigaben einfach die Finger lässt. Vermutlich wird sich auch kein Auditor daran stören, dass man sich diese Tools, da sie ja schon einmal auf der Platte sind, einmal in Ruhe anschaut, um über deren Kauf zu entscheiden. Ein Recht darauf, dass die ersten paar Klicks kostenlos sind, hat man jedoch keineswegs. Wer also gar nicht wirklich am Kauf dieser Produkte interessiert ist, sollte sie auch wirklich nicht starten.

Wer hingegen interessiert ist, dem bietet Oracle mehrere Optionen an. Teilweise bedingen diese zusätzliche Downloads, teilweise handelt es sich lediglich um das Abschließen eines zusätzlichen Lizenzvertrags. Gerade in Letzterem liegt der Grund, weshalb die Downloads des „eigentlich freien“ Java SE die „eigentlich kostenpflichtigen“ Tools bereits enthalten. Der Kasten „Varianten von Java SE“ erklärt, welche Optionen derzeit angeboten werden und warum es so viele gibt.

Fazit

Auch wenn die Panikmache in den Medien übertrieben war – es gilt: „Augen auf beim Java-Kauf!“ Selbst wenn kein Auditor seinen Besuch angekündigt hat, sollte man sich überlegen, welche kostenlosen/kostenpflichtigen Werkzeuge wirklich benötigt werden, und dann entweder auf OpenJDK umsteigen, nur die kostenlosen Oracle-Bestandteile nutzen und weitergeben oder einen (kostenpflichtigen) Lizenzvertrag mit Oracle abschließen.

Markus Karg

markus@headcrashing.eu



Markus Karg fasziniert das Programmieren, seit er in grauer Vorzeit einen Sinclair ZX Spectrum in die Hand bekam. Heute verantwortet er die Entwicklung eines unabhängigen Software-Herstellers. JAX-RS begleitet der gebürtige Pforzheimer seit Version 0.8 durch Feature Proposals, Mitarbeit an der Referenz-Implementierung „Jersey“ und zuletzt in der Expert Group JSR 370. Einige Features von JAX-RS, aber auch von anderen Java-APIs, stammen aus der Feder des JCP-Mitglieds. Seine Freizeit widmet er der Kunst seiner Frau sowie einer nachhaltigen Gesellschaft.



APEX Connect 2017

9. bis 11. Mai 2017 in Berlin