



VON JAVA ZU JAKARTA

EIN INTERVIEW MIT WERNER KEIL

Geführt von Alexander Salvanos, Senior IT Consultant bei msg systems ag

Seit 1998 ist die Java Enterprise Edition (Java EE) der Industriestandard, der es sich unter anderem zur Aufgabe gemacht hat, die besonderen Sicherheitsprobleme der geschäftskritischen Unternehmensanwendung von Industrie und Behörden zu lösen. Und auch wenn dieser Industriestandard im Jahr 2018 an die Eclipse Foundation ging und dort auch noch mit Jakarta EE einen neuen Namen bekam, so ist es immer noch der gleiche Industriestandard. Das Ziel von Jakarta EE (ehemals Java EE) ist es nach wie vor, eine Plattform anzubieten, mit der sich hochskalierbare, zuverlässige und sichere Unternehmensanwendungen erstellen lassen.

Und genau wie Java EE ist auch Jakarta EE Teil des Java Community Process (JCP). Sie erinnern sich: Beim JCP werden Java-Neuerungen als Spezifikationsanfrage (JSR) eingereicht. Nimmt das Exekutivkomitee die Anfrage an, so durchläuft sie einen formalen Prozess, bei dem sie eine Expert Group genauer

spezifiziert. Auch alles andere, wie die technologischen Kompatibilitätstests (TCK), wird auf gleiche Weise erstellt. Dennoch gibt es einen kleinen Unterschied, und der liegt im Führungsmodell. Denn Jakarta EE ist nun Community basiert. Hiermit ist nicht nur gemeint, dass sich jeder Java-Programmierer an der Programmierung als gleichgestellter Entwickler beteiligen kann, sondern auch, dass die neuen Spezifikationen und Entwicklungsprozesse für Jakarta EE offen und herstellerneutral sind und gleiche Wettbewerbsbedingungen für alle Teilnehmer bieten. Das klingt alles sehr positiv. Dennoch gab es Hürden: Beispielsweise stellte der Transfer der Java-EE-Quelltexte von Oracle an die Eclipse Foundation eine große Herausforderung dar. Gerade hierbei ist es daher sehr wichtig, dass erfahrene Java-EE-Persönlichkeiten auf dem Schiff mit der nun neuen Flagge mitreisen. Eine der herausragenden Personen hierbei ist Werner Keil, der bereits zu Zeiten von Sun Microsystems Mitglied des Java Community Process Executive Committee

und der Java EE Expert-Group tätig war und dort mit zahlreichen Awards prämiert wurde. Seine Auszeichnungen füllen Regale: JCP Award Winner, JCP Spec Lead/JSR of the Year, Speaker of All Time, JCP Award Winner, Member of the Year JCP Award Winner, Most Significant JSR Spec Lead at JSR 385 JCP Award Winner, JSP Executive Committee Member, Java EE Expert Group Member und nun auch Jakarta EE Specification Committee Member und Committer Representative der Jakarta EE Working Group. Wir hatten die Gelegenheit, mit Werner Keil zu sprechen.

msg: Vielen Dank, Werner, dass Du Dir heute die Zeit genommen hast für uns. Wann und wie bist Du erstmalig mit Java EE in Berührung gekommen und wie bist Du zu diesen hohen Positionen in der Java- und Java-EE-Community gekommen?

Werner Keil: Erstmals bei einer JavaOne-Konferenz war ich 1998, nachdem mir mein Kunde aus dem öffentlichen Sektor, nämlich die österreichische Pensionskasse, freundlicherweise ein Ticket mit Flug zur JavaOne finanzierte. Auf der Konferenz wurde auch das spätere Java EE erstmalig vorgestellt. In das JCP Executive Committee und in die Java EE Expert Group wurde ich aber erst im Jahre 2007 gewählt, nachdem ich für Sun Microsystems und BEA/Oracle gearbeitet habe. Diese Tätigkeit hat mich dann auch inspiriert und in die Lage versetzt, die Standards später aktiv mitzugestalten.

msg: 2018 ging Java EE 8 von Oracle an die Eclipse Foundation über, wo sie kurze Zeit später den Namen Jakarta EE 8 erhielt. Welche Auswirkungen, denkst Du, hat diese Entscheidung für den Industriestandard und somit für unternehmenskritische Geschäftsanwendungen gehabt?

Werner Keil: Ein großer Vorteil des Übergangs von Oracle an die Eclipse Foundation ist, dass seitdem Entscheidungen demokratischer gefällt werden. Vorher hatte Oracle stets das letzte Wort, denn es waren ja meistens Oracle-Mitarbeiter an Java EE beteiligt. Und jede Spezifikation hatte auch wieder einen Spezifikationsleiter von Oracle usw. Dadurch hatte Oracle ein Vetorecht. Das gibt es in dieser Form jetzt nicht mehr. Denn durch den Wechsel haben nun alle das gleiche Mitbestimmungsrecht. Dadurch kann auch niemand mehr diktieren, was er im Standard haben will.

Wenn beispielsweise CORBA oder EJB aus dem Standard herausgenommen werden sollen, wird diese Entscheidung demokratisch gefällt. Und auch ich als Committer Representative, also als Vertreter der Jakarta EE Committer Members, habe

eine Stimme, genauso wie die Strategic Enterprise Members, die einen Jahresbeitrag bezahlen. Das sind beispielsweise die Unternehmen IBM, Red Hat usw. Zwar hat Oracle als Industrieunternehmen hierdurch auch eine Stimme, aber es hat kein Vetorecht mehr. Und auch die verschiedenen Arbeitsgruppen, wie das Specification Committee, oder führende Funktionäre der Eclipse Foundation, können nicht über diesen demokratischen Prozess hinweg entscheiden.

msg: Das hört sich ja gut an. Hatte diese Änderung auch Nachteile zur Folge?

Werner Keil: Ja, schon ein bisschen. Denn jetzt müssen alle Mitglieder zur Weiterentwicklung des Jakarta-EE-Standards beitragen. Man kann also nicht mehr sagen, dass Oracle und vielleicht auch noch IBM die Arbeit machen und die anderen höchstens mal alle paar Monate ihre Stimme abgeben. Sondern es ist sehr viel mehr aktive Gestaltung gefragt. Früher waren es ja vor allem zahlreiche Mitarbeiter von Oracle, die an der Programmierung beteiligt waren. Jetzt sind es gefühlt nur noch zwei bis vier Oracle-Mitarbeiter, die mitarbeiten.

msg: Sind auch Behörden in irgendeiner Weise an der Weiterentwicklung von Jakarta EE beteiligt?

Werner Keil: Ja, zumindest indirekt. So sind beispielsweise das Deutsche Zentrum für Luft- und Raumfahrt (DLR) und das Bundesinstitut für Risikobewertung (BfR) Eclipse-Mitglieder, ebenso die Deutsche Bahn oder die Schweizer Bundesbahnen (SBB), auch wenn diese derzeit wahrscheinlich nicht aktiv an Jakarta EE mitwirken. Und ich habe gerade heute mit den Eclipse-Vertretern über das von der EU gehostete und gesponserte DSS-Projekt¹ (Digital Signature Service) gesprochen. Aber in der Regel sind Behörden vorwiegend Nutzer.

JAKARTA EE – EIN SOLIDER INDUSTRIE-STANDARD FÜR BEHÖRDEN

msg: Für Behörden waren die standardisierten Java-Technologien stets von großer Bedeutung, denn ihre geschäftskritischen Unternehmensanwendungen müssen besonderen Sicherheitsanforderungen gerecht werden. Denkst Du, dass Jakarta EE immer noch den soliden Industriestandard darstellt, wie er von den Behörden für ihre unternehmenskritischen Geschäftsanwendungen benötigt wird?

Werner Keil: Ja, nun sogar umso mehr. Denn erst jetzt ist vorhersehbarer geworden, wohin sich der Standard bewegt.

Zum Beispiel wird jetzt niemand mehr aus wirtschaftlichen Gründen die Technologien in eine eigene Cloud befördern, nur um seinen eigenen Gewinn zu steigern.

msg: In der Vergangenheit wurden Java-EE-Server von den Großen der IT-Industrie zur Verfügung gestellt. Dadurch konnten sich Behörden darauf verlassen, dass sie ein solides Fundament vorfinden, auf dem sie ihre Anwendungen laufen lassen können. Ist das noch immer so, oder hat sich hieran etwas geändert?

Werner Keil: Hieran hat sich nichts geändert. Ganz im Gegenteil, denn es sind noch mehr Unternehmen wie Fujitsu und weitere Anbieter als solides Fundament hinzugekommen. Gerade aus dem asiatischen Raum kamen zuletzt besonders viele Jakarta-EE-Mitglieder dazu – neben Fujitsu TmaxSoft aus Südkorea sowie sechs Mitglieder aus China, darunter die beiden einzigen Enterprise-Mitglieder der Jakarta EE Working Group.



Fast alle alteingesessenen Hersteller haben ihre Produkte auch schon kurze Zeit nach der Ankündigung von Jakarta EE 8 zertifizieren lassen, so auch IBM WebSphere – jetzt Liberty – und JBoss beziehungsweise Wildfly. Mit Ausnahme von Tomitribe, wo man zwar oft etwas „laut“ ist, was aber verglichen mit Oracle, IBM oder Fujitsu bestenfalls ein KMU (kleine und mittlere Unternehmen) darstellt, haben alle Strategic Members der Jakarta EE Working Group ihre Produkte bereits für Jakarta EE 8 zertifizieren können.

Auch für Jakarta EE 9 ist mit IBM Open Liberty bereits der erste Server, abgesehen von GlassFish, als kompatibel zertifiziert. Andere hingegen sind etwas in den Hintergrund geraten. Beispielsweise ist GlassFish zwar die primäre kompatible Implementation. Aber dadurch, dass bei Jakarta EE der Begriff der Referenz-Implementation weggefallen ist, spielt GlassFish keine so eine große Rolle mehr. Einige Nutzer wie beispielsweise BMW sind dadurch zu Payara gewechselt. Oracle WebLogic ist inzwischen kompatibel mit Jakarta EE 8, aber bei Oracle ist zumindest ein gewisses Fragezeichen vorhanden, weil sie die

Technologien lieber in ihrer eigenen Cloud anbieten. Wobei Oracle hier allerdings weniger erfolgreich unterwegs ist als große Mitbewerber wie Microsoft, Google oder Amazon, die oftmals auch ihre eigenen OpenJDK-Distributionen bereitstellen, wie Coretto bei Amazon.

ÄLTERE, AUF JAVA EE BASIERENDE INSTALLATIONEN WERDEN IMMER WENIGER MIT JAKARTA EE KOMPATIBEL SEIN

msg: Seit vielen Jahren setzt sich Apache Tomcat in Kombination mit Spring und Hibernate in Behörden immer weiter durch. Genauso wird als Frontend immer mehr auf JavaScript-basierte Frameworks wie Angular gesetzt. Ist das nicht ein Trend, der die ganze Idee eines Industriestandards wie Java EE oder jetzt Jakarta EE infrage stellt?

Werner Keil: Es war ja ohnehin schon in der Vergangenheit so, dass Java-EE-Versionen stets neue Java-Server-Versionen mitbrachten und dies auch stets zu neuen Tomcat-Hauptversionen führte. Und Tomcat ist und war ja auch immer eine Implementierung zahlreicher Java-EE-Technologien. Somit sollte man Tomcat nicht als Konkurrent zu Java EE betrachten. Und dieser Anschluss zum Enterprise-Standard setzt sich auch mit Jakarta EE fort. So basieren auch die aktuelleren Tomcat-Versionen mehr und mehr auf Jakarta-EE-Versionen.

Ab Apache Tomcat Version 10 wird die Angleichung zum Jakarta Servlet und somit Jakarta-EE-Standard komplett sein. Ältere, auf Java EE basierende Installationen werden immer weniger mit Jakarta EE kompatibel sein. Dies betrifft im Übrigen auch Spring-Technologien wie Spring Boot, die sich ebenso heute schon auf Jakarta EE beziehen. Was Angular anbelangt, ist es schon so, dass Angular eine bedeutende Rolle spielt. Allerdings handelt es sich bei Angular ja nicht um eine Application-Server-basierte, sondern um eine clientseitige Technologie, die nicht ohne ein serverseitiges Pendant benutzt wird. Somit ist es zwar richtig, dass der Trend zu Angular Bestand hat, aber nicht als Konkurrenz gesehen werden kann, weil für das Backend stets ein Application Server eingesetzt werden muss.



DIE HÖHERE VERBREITUNG VON SKRIPT- SPRACHEN UND DIE INTERPRETATION IHRES QUELLCODES ZUR LAUFZEIT MACHEN DAS EINSCHMUGGELN BÖSARTIGER CODESEQUENZEN LEICHTER.

msg: Für mich ist eine Frage noch besonders interessant. Wenn Du jetzt eine Behörde beraten solltest: Welche Technologien würdest Du zur Nutzung empfehlen, wenn es sich um eine transaktionale, geschäftskritische, sichere Webanwendung handeln würde.

Werner Keil: Es gibt heutzutage immer mehr Programmiersprachen und Technologien, die für sich in Anspruch nehmen, dass mit ihnen geschäftskritische Webanwendungen und APIs besonders gut und effizient entwickelt werden können. Auch durch die meist steilen Lernkurven sind dabei Skriptsprachen wie JavaScript beziehungsweise TypeScript oder Python besonders populär. Die steile Lernkurve und höhere Verbreitung sowie die Tatsache, dass die Skriptsprachen per Definition eben nicht in ein binäres Artefakt wie eine ausführbare Datei oder ein Archiv münden, sondern der Quellcode zur Laufzeit interpretiert wird, machen das Einschmuggeln böser Codesequenzen dort leichter.

Die Verbreitung von Angular oder NodeJS gerade auch im unternehmenskritischen Umfeld führt immer öfter zu Angriffen oder der Ausnutzung von Sicherheitslücken in diesem Bereich. Der Node Package Manager (npm),

De-facto-Standard für das JavaScript-Ökosystem, war in den vergangenen ein bis zwei Jahren mehrfach Ziel von Angriffen, die ich beispielsweise im Gespräch mit Speaker-Kollegen auf der Developer Week bereits ein paar Jahre früher vorhergesehen habe, als sie mir damals erzählten, dass die Pakete bei npm ohne jede Authentifizierung anonym hochgeladen und beliebig oft ersetzt werden könnten.

Das mag sich seither gebessert haben, aber dank sogenannter „Dependency

Confusion“ konnte sich ein Sicherheitsexperte, wie eben erst Anfang Februar berichtet², in einige der größten IT- und Cloud-Unternehmen weltweit, wie Apple, Microsoft, PayPal, Netflix oder Uber, hacken. Währenddessen setzen die binären Repositories des De-facto-Standards für die Java Virtual Machine, Sonatype Nexus beziehungsweise MavenCentral, seit Jahren auf Zugangsdaten und digitale Signaturen zum Schutz ihrer Artefakte. Und selbst das Deployment eines Pakets in einem neuen oder unbekanntem Namensraum ist nicht ohne vorherige Genehmigung durch die Urheber der jeweiligen Projekte, beispielsweise den Domain-Eigentümer, möglich.

Auch wenn Java und Java EE beziehungsweise Jakarta EE schon mehrfach totgesagt wurden, halte ich auch aus diesem Grund beide für eine solidere Basis, um geschäftskritische, transaktionale und vor allem sichere Webanwendungen zu entwickeln. Es spielt dabei keine so große Rolle, ob man auf Java, Scala, Kotlin oder Clojure schwört und welches Framework letztlich zum Einsatz kommt – ob „pure“ Jakarta EE oder darauf basierende Lösungen wie Spring Boot, Eclipse MicroProfile, Micronaut oder sonstige – solange sie auf der JVM laufen, was dank GraalVM übrigens auch für die meisten Skriptsprachen heute möglich ist.

msg: Welche Rolle wird eine Modularisierung von Jakarta EE in Zukunft spielen?

Werner Keil: Bereits mit Jakarta EE 9 habe ich im Specification Committee geholfen, gut zwei Drittel der Jakarta-EE-Spezifikation-

APIs mit JPMS-Moduldefinitionen (seit Java 9 „Jigsaw“ Modulsystem) zu versehen. Daneben enthalten auch etliche eine OSGi-Bundle-Deklaration. Da die beiden Systeme konkurrieren und manche Committer insbesondere aus dem Apache-Lagerhistorisch bedingt ihre API JARs wiederpacken (seit sie mit Apache Harmony lange Zeit auf Konfrontationskurs mit dem JCP und Sun beziehungsweise Oracle waren), statt auf die offiziellen Java EE oder Jakarta EE JARs zu setzen, ist das nicht immer so einfach.



ZUR PERSON

WERNER KEIL ist Cloud Architekt, API-Designer, Java- und Microservice-Berater für die unterschiedlichsten Branchen. Er entwickelt Enterprise-Systeme mithilfe von Java, Java EE, Jakarta EE, Oracle, IBM oder Microsoft, betreibt Webentwicklung mit JavaScript und anderen dynamischen oder funktionalen Sprachen. Er ist seit über dreißig Jahren als Projektleiter, Software-Architekt, Analyst und Technologie-Berater für führende Unternehmen tätig.

Werner Keil ist unter anderem Committer in der Apache und der Eclipse Foundation, Committer Vertreter im Jakarta EE Specification Committee sowie aktives Mitglied der Java Community Process.

Neben seiner Tätigkeit für große Unternehmen leitet oder unterstützt Werner Keil zahlreiche Open-Source-Projekte, schreibt Songtexte, Romane, technische Bücher und Artikel. Er ist ein gefragter Speaker auf Konferenzen weltweit.

Spätestens mit dem Java-Modulsystem wurde aber vom JDK eine gewisse Grundlage bereitgestellt, von dem nicht nur Oracle selbst in seinem Helidon Microservice Framework intensiven Gebrauch macht. Auch die erste zertifizierte Implementierung einer einzelnen Jakarta-EE-Spezifikation (JSON Processing) „Joy“ setzt ganz auf Java 14 und höher mit JPMS.

msg: Ich habe kürzlich davon gehört, dass Du zusammen mit anderen Jakarta-EE-Experten an einem Buch schreibst. Worum geht es dabei?

Werner Keil: Das Buch „The Definitive Guide to Jakarta EE Security“ im Apress Verlag, an dem ich zusammen mit dem Leiter der Utrecht Java User Group Thodoris Bais und Java Champion Arjan Tijms arbeite, ist für Frühjahr 2021 geplant. Wie der Titel

erahnen lässt, deckt es alle Facetten und wesentlichen Technologien zum Thema Sicherheit auf Jakarta-EE-Basis ab – angefangen bei Java-Security-Grundlagen, digitaler Signatur oder der Java Cryptography Extension über JSON Web Token (JWT), OAuth beziehungsweise OpenID Connect bis zu den Jakarta-Sicherheitsstandards Authentication, Authorization und Jakarta Security API. Wen das Thema interessiert, der kann sehr gerne dem entsprechenden Twitter-Account³ folgen, dort finden sich die neuesten Ankündigungen und, sobald Apress die Buch-Website freigeschaltet hat, auch Links darauf.

msg: Vielen Dank, Werner, für das sehr informative Interview.

Werner Keil: Vielen Dank für die Einladung zu diesem Interview. ●



msg: Bist Du immer noch als Speaker so aktiv wie in der Vergangenheit? Welche IT-Konferenzen würdest Du unseren Lesern empfehlen?

Werner Keil: Ich wurde in den letzten zehn bis fünfzehn Jahren auf so gut wie alle größeren Entwickler- und IT-Konferenzen eingeladen. Mit Ausnahme von Australien/Ozeanien auf jedem Kontinent und auf alle größeren Veranstaltungen, die 2000 oder mehr Besucher anziehen, in der D-A-CH-Region. Deshalb wäre es fast etwas unfair oder auch zu umfangreich, hier alle zu erwähnen oder einzelne hervorzuheben. Ich möchte dennoch, mit Schwerpunkt auf Mitteleuropa, einige erwähnen, abhängig von der jeweiligen Plattform und Thematik. Wer größtenteils auf der Java Virtual Machine (JVM) zu Hause ist, der kommt an Devovx (ehemals JavaPolis) in Antwerpen oder Javaland in Brühl bei Köln kaum vorbei. Devovx hat seit einigen Jahren auch mehrere Ableger in England, Frankreich oder Osteuropa sowie kleinere – Vovxed Days genannte – meist eintägige Gastkonferenzen in anderen Städten und Ländern. Das JavaLand ist zwar noch nicht ganz so groß wie die größten Devovx-Events, die meist um die 4.000 Besucher anziehen, wird aber – mit Ausnahme von 2020 – jedes Jahr etwas größer. Zuletzt kamen 2019 wohl schon über 2.000 Besucher ins Phantasia-land. Im Coronajahr 2020 wurden bis auf die Devovx Ukraine online fast alle anderen Konferenzen abgesagt. In diesem Jahr findet das JavaLand corona-bedingt online statt. Welchen Einfluss das auf die Teilnehmerzahlen hat, bleibt abzuwarten. Aber da Anreise und Hotel wegfallen, könnte es Manchen die Teilnahme sogar erleichtern. In deren Schatten ist die JCON

in Düsseldorf, die sich in einem großen Kino vom Ambiente her an der belgischen Devovx orientiert, in den letzten Jahren auch beständig gewachsen. Und konnte trotz oder gerade wegen Corona 2020 als Onlineevent mit mehr Besuchern als je zuvor punkten. Die 2.000er-Marke hat sie ebenfalls überschritten.



Abseits der JVM ist die Codemotion-Serie ebenfalls zu empfehlen. Sie ist mehr polyglott und deckt alle wesentlichen Plattformen von JavaScript/NodeJS über Python oder .NET bis zu JVM-Sprachen ab. Sie hat in Rom den größten Standort mit mehr als 4.000 Besuchern. Nach weitgehenden Absagen finden alle Codemotion-Konferenzen nun wohl online statt. Ansonsten gab es unter anderem in Berlin meist eine etwas kleinere Codemotion. Ähnlich polyglott, mit mehr als 2.500 Besuchern in einem normalen Jahr, sind die Developer Week (DWX) in Nürnberg oder die OOP in München. Die IT-Tage in Frankfurt waren mit 1.000 Besuchern vor Ort etwas kleiner. Sie fanden 2020 auch virtuell statt und sind für Ende 2021 geplant. Ich kenne hier keine Teilnehmerstatistik, aber ähnlich wie bei der JCON dürfte sich das Ausweichen in die Cloud kaum negativ auf die Teilnehmerzahl ausgewirkt haben.

1 <https://ec.europa.eu/cefdigital/DSS/webapp-demo> (abgerufen am 29.03.2021).

2 <https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610> (abgerufen am 05.03.2021).

3 <https://twitter.com/jakartasecbook/> (abgerufen am 05.03.2021).