



30 Jahre Java

Patrick Baumgartner, Heinz Kabutz, Sandra Parsick, Peter
Roßbach, Lars Vogel, Eberhard Wolff, Sebastian Meyen

Phasen

1991

Die Anfänge: Set Top Boxen / Web Browser

1998

Die Erfolgsgeschichte: Server mit lang laufenden Prozessen

2013

Die Moderne: Container mit schnell startenden Prozessen

Modell

Sprache	Java nebst alternativer JVM-Sprachen
Plattform	Runtime, Bibliotheken, Tools
Community	Entwickler:innen, JUGs, Industrie

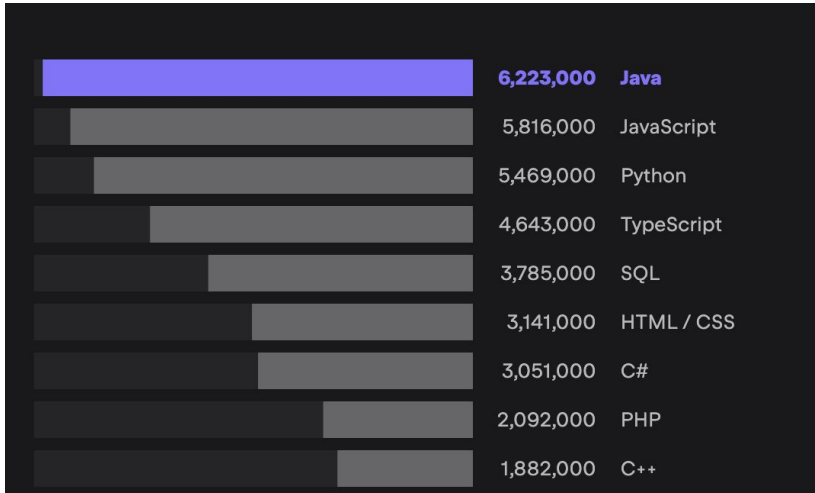
Anfänge (1)

- 1995
 - Java als erste Beta
- 1996
 - Java 1.0 – Applets im Browser
 - Sun versammelt Partner, die Java lizenzieren mit dem gemeinsamen Ziel einer universellen Programmier- und Ablaufplattform
- 1997
 - Microsoft als Lizenznehmer modifiziert Java durch proprietäre Elemente: WFC und Windows APIs
 - Rechtstreit Sun/Microsoft: Kein Microsoft-Java mehr

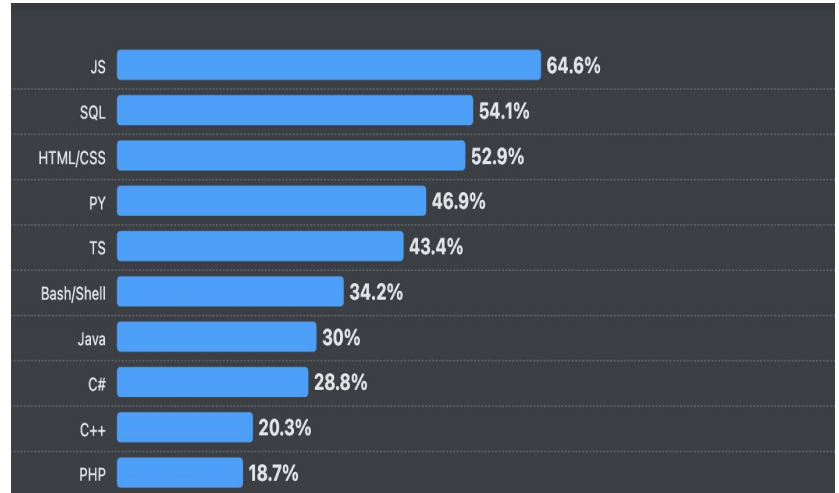
Anfänge (2)

- 1998
 - Servlet Spec, EJB
 - JCP 1.0: strukturierter kollaborativer Entwicklungsprozess (Spec first). Sun: volle Kontrolle.
- 1999
 - Kooperation mit ASF, Tomcat
 - Java 2 Platform: J2SE, J2EE, J2ME
- 2000
 - JCP 2.0 inklusiver transparenter. Partner können JSRs übernehmen.

Heute



JetBrains



StackOverflow



Kooperatives Entwicklungsmodell

Heinz Kabutz

Von proprietärer Entwicklung zum kooperativen Entwicklungsmodell

- **1995 - Java 1.0 veröffentlicht** als Mosaic und Netscape beliebt wurden.
- **1997 - Microsoft J++ proprietären Erweiterungen** wurden von Sun Microsystems verklagt
- **1998 - Java Community Process (JCP) mit JSRs** als Standardisierungsmodell
- **2006 - OpenJDK** als Open-Source-Implementierung der Java-Plattform
- **2010 - Oracle kauft Java** und öffnet den Prozess noch mehr mit JEPs
- **Alternative JVM-Sprachen:** Groovy (2003), Scala (2004), Clojure (2007), Kotlin (2011)
- **Java als Sprache für verteilte Systeme:** Performanz, threads, OSGi & Jini



Innovation durch Offenheit

Die Kraft von Open Source

Peter Roßbach

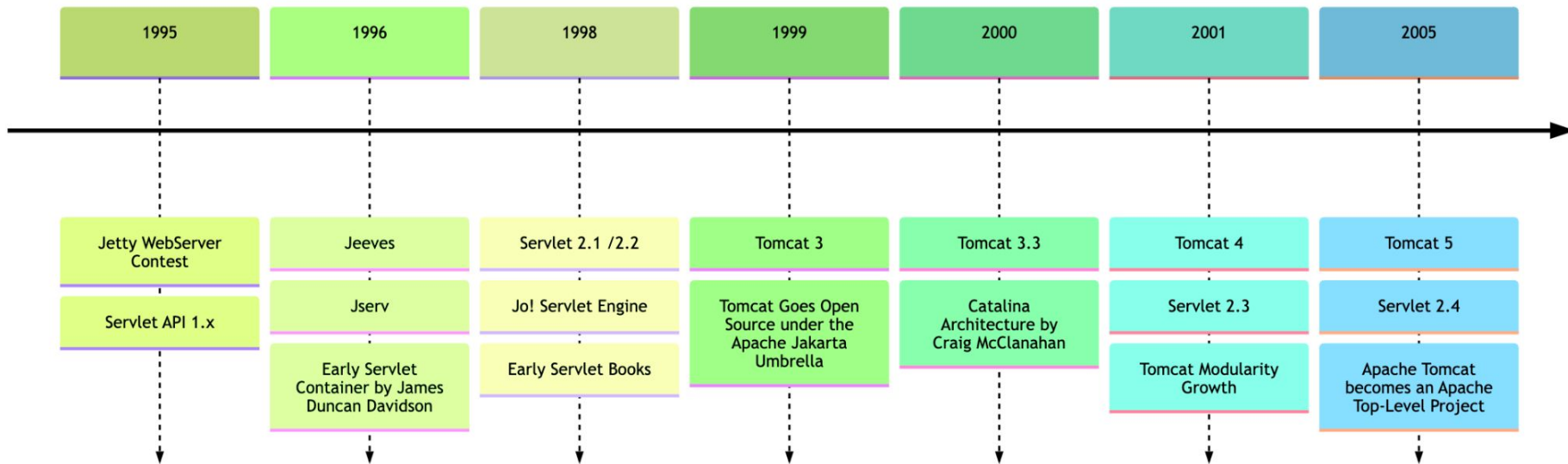
Open Source als Motor der Innovation: Apache Software Foundation

Während sich die Sprache Java stetig weiterentwickelte und selbst nicht Open Source vorlag, entstanden parallel mächtige Open-Source-Initiativen. Die Apache Software Foundation spielte eine zentrale Rolle dabei, Java in der realen Softwareentwicklung zu etablieren, indem sie Werkzeuge und Frameworks bereitstellt, die Java weltweit produktiv nutzbar machten.

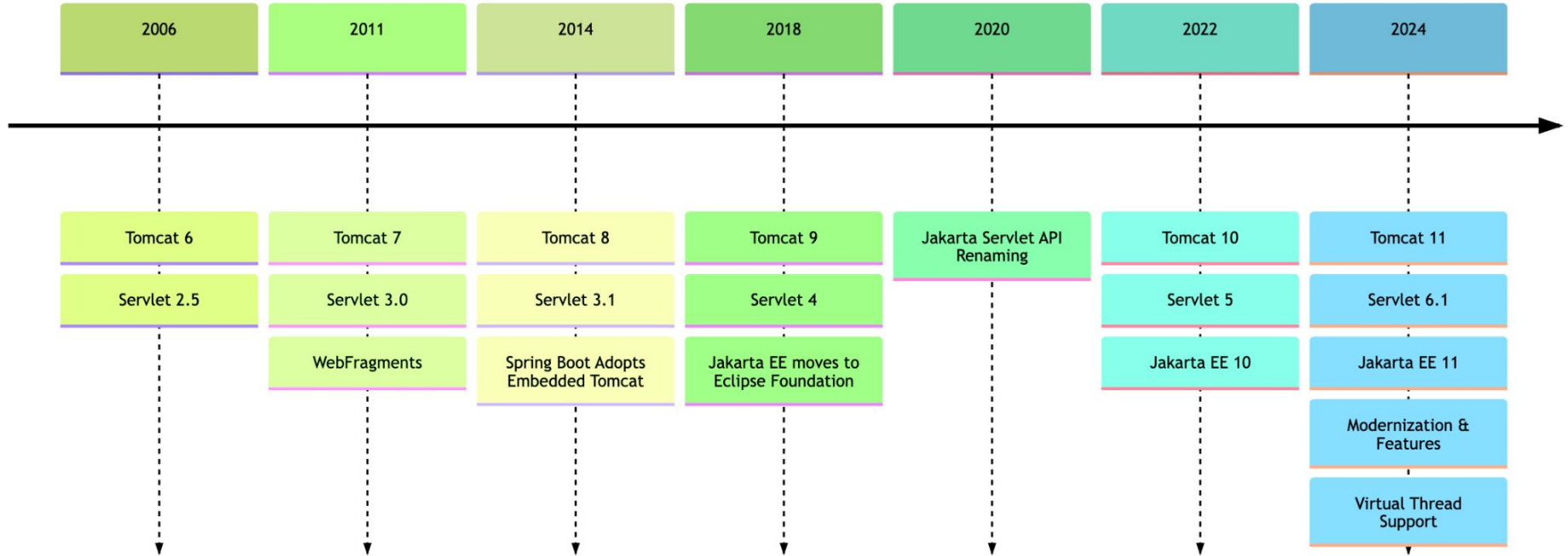
- **Apache Tomcat:** Referenzimplementierung der Servlet API und JSP, der Java-Webserver
- **Apache Commons:** Basis Java Frameworks
- **Apache Log4j:** Das erste effiziente Logging- Framework
- **Apache Struts & Cocoon:** frühe MVC-Frameworks für Webanwendungen
- **Apache Maven & ANT:** Build- und Projektmanagement-Werkzeuge
- **JCP, J2EE & JEE, SpringBoot:** Erfolgreiche Adaptation



Tomcat wird ein Top Level - Apache Software Foundation Projekt



Tomcat Adaption und Skalierung



Apache Software Foundation

- **Mehr als 350 Open-Source-Projekte**
- **Gegründet 1999 – heute über 8.000 Committer**
- **100% gemeinnützig und unabhängig**
- **Milliarden Nutzer erfreuen sich täglich**
- **Mitarbeit jederzeit willkommen!**



Community over code!

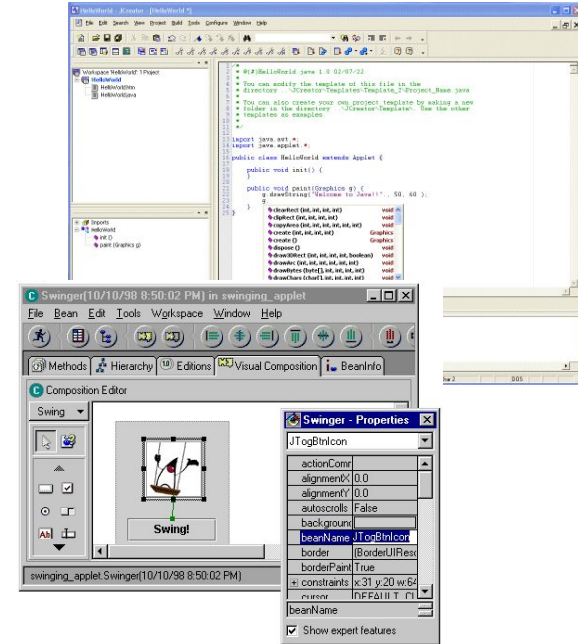
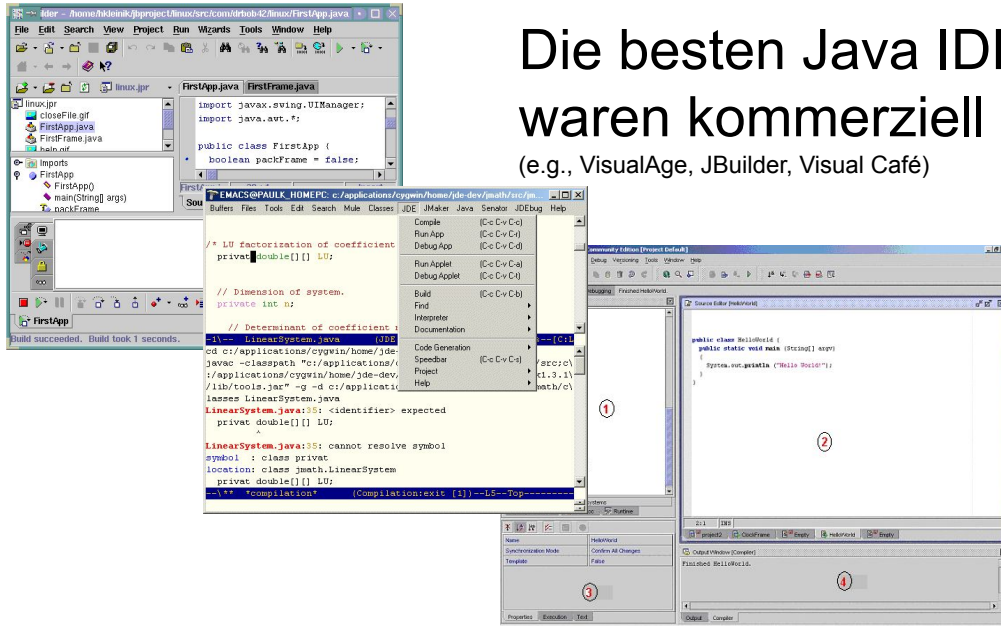
Eclipse: Toolchains als Disruption



Lars Vogel, Eclipse Developer und
langjähriger Projektleiter

IDE Alternativen vor 25 Jahren

Die besten Java IDE's
waren kommerziell
(e.g., VisualAge, JBuilder, Visual Café)



2002 - Eclipse IDE wird “de facto” Java IDE standard

Modular und
erweiterbar

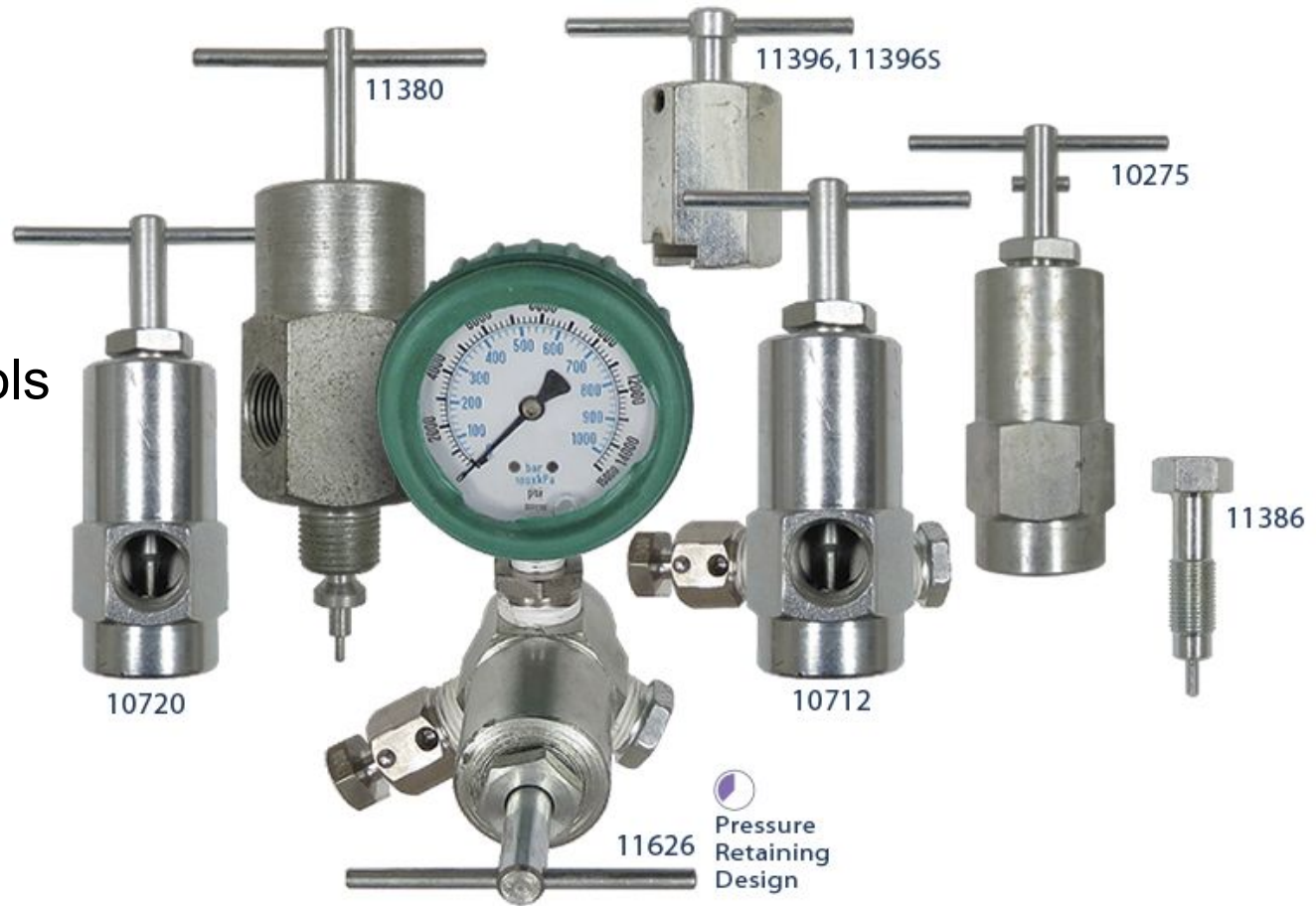
Schnell und stabil



IBM VisualAge for Java
wird Eclipse in 2001 und
von der IBM “unabhängig”
in 2004

Eclipse is **completely free** and
open source under the
Eclipse Public License.

Druck auf andere Tools
wie z.B. IntelliJ

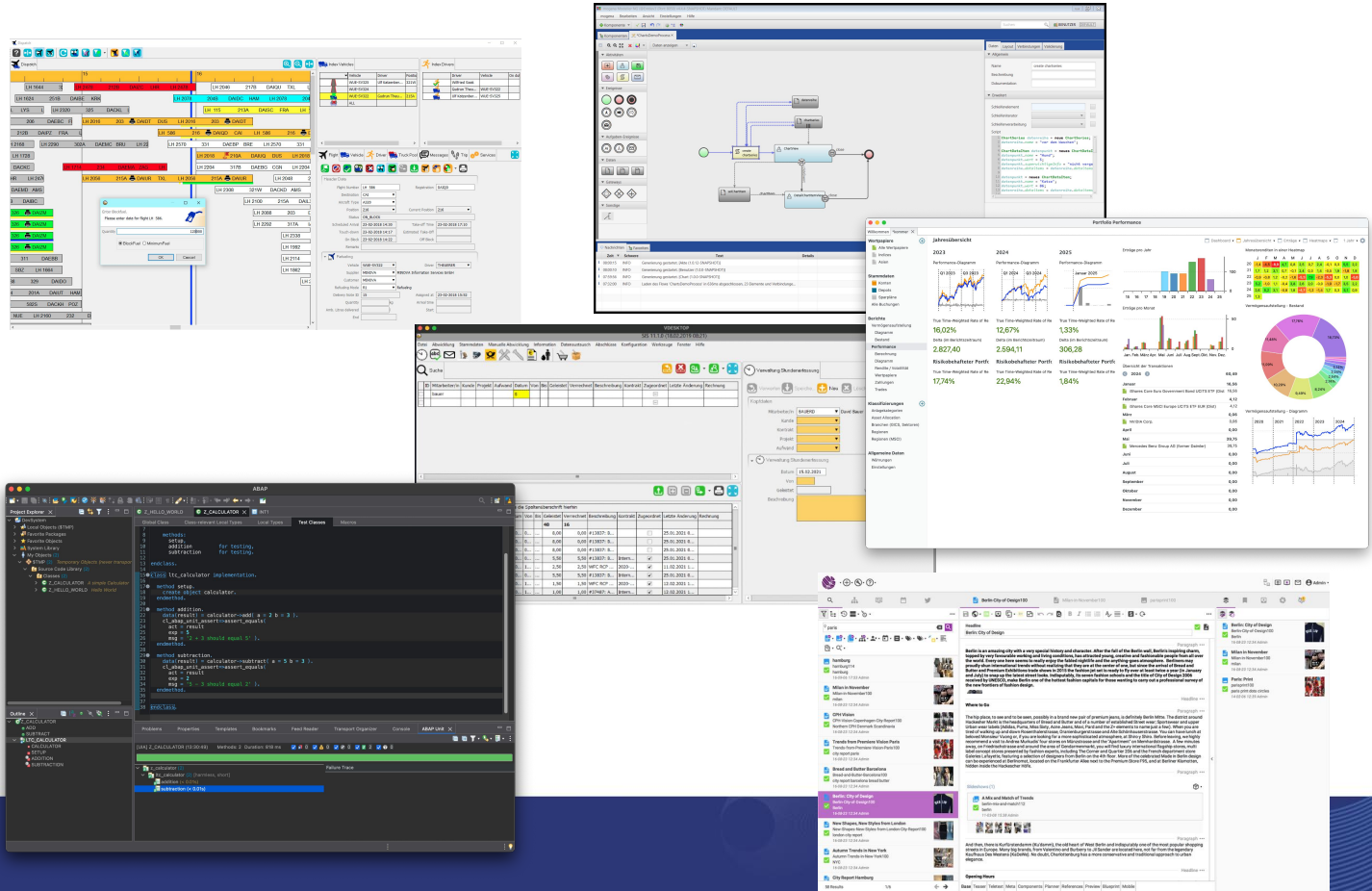


SWT und RCP helfen Java den Desktop zu erobern

SWT for building native looking desktop applications

Eclipse RCP as rapid development framework

OSGi as module system



Eclipse als “business friendly” Community



Eclipse **Foundation** mit
Ansprechpartnern, Förderung
durch Sponsoren, Sammeln von
Geldern etc

Definierte Prozesse und
Ansprechpartner bei Fragen

Mehr als 400 Open Source Projekte

Heimat für **Jakarta EE** nachdem
Oracle Interesse an Java EE
verliert

Heimat für die wichtige JRE
Implementierung **Adoptium** oder
Eclipse Temurin



Java und Tools in 2025

Viele, viele **Alternativen**, wie Visual Studio Code, IntelliJ, Eclipse, etc

Austausch zwischen den Tools basierend auf dem **Language Server Protokoll**

Schnelle Releases z.B. Eclipse IDE alle 3 Monate

Super **KI Unterstützung** für Java und Tools

A close-up photograph of various carpenter's tools. In the foreground, a hammer with a wooden handle and a metal head lies on a wooden workbench. Next to it are a pair of worn, light-colored leather work gloves. A brown leather tool belt with white stitching is also visible, containing a yellow tape measure and other small tools. In the background, a circular saw with a red blade guard is partially visible. The scene is set in a workshop with a warm, slightly blurred background.

**Java ist und bleibt stark
durch seine starken Tools**



Enterprise Java in der Krise

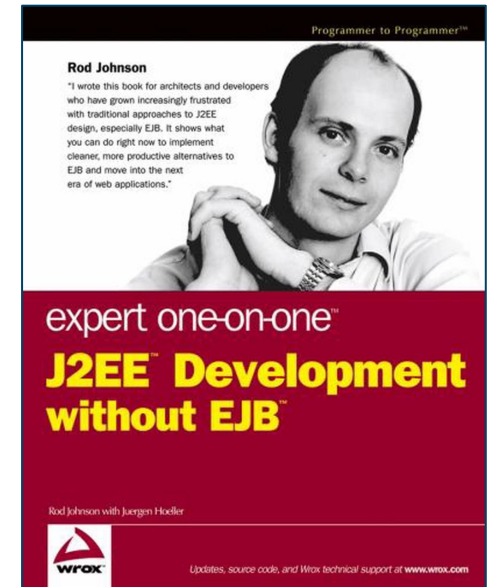
Eberhard Wolff

Enterprise Java steckt in der Krise: Spring eröffnet neue Horizonte

J2EE und die damit verbundenen **Standards** - vor allem **EJB** - waren in der Praxis schwierig einzusetzen. Mit **Spring** entstand 2002 ein leichtgewichtiger, flexibler Ansatz.

- **Dependency Injection** zum Zusammenstecken und Konfigurieren
- **AOP (Aspektororientierte Programmierung)** als Basis für Transaktionen etc
- **Spring MVC** für Webanwendungen
- **Abstraktionen und Vereinheitlichungen** über APIs (JDBC, Hibernate, JTA, JMS ...)

Ergebnis: Der Tod der Application Server und damit der J2EE-Container.



Spring wird zum Ökosystem

- **Spring Data** für Persistenzschichten
- **Spring Integration**
- **Spring Batch**
- **Spring Security**
- **Spring Modulith** für die Strukturierung von Anwendungen
- **Spring Boot** für Convention over Configuration und weitere Vereinfachungen



Cloud & Container

Sandra Parsick

Cloud & Container

Als Cloud-Technologien ihre Macht entfalten, schien Java ins Hintertreffen zu geraten. Doch es waren Open-Source-Projekte und Community-Initiativen, die mit Spring Boot, Quarkus und Kubernetes-Integration Lösungen für moderne Cloud-native Architekturen bereitstellen.

- **Spring Boot (2014)** als leichtgewichtige Plattform für Microservices
- **Docker (2013) & Kubernetes (2014)** als Basis für containerisierte Java-Anwendungen
- **GraalVM (2019)** - Native Java Compiler zur Optimierung der Startzeit und Steigerung der Performance
- **Micronaut (2018) & Quarkus (2020)** für Java-Anwendungen mit geringerem Ressourcenverbrauch

Docker & JVM

- Testcontainers
- Native Container Support seit Java 10
- Container erhöhte den Wunsch
 - nach schnelleren Start-Zeiten und
 - kleineren Anwendungen
- Kubernetes
 - “erleichterte” den Betrieb von Microservices-Landschaften
 - vereinfacht den Betrieb von verschiedenen Ökosystemen



Java & AI

Patrick Baumgartner



- Erstellt 2023 von Mark Pollack und Christian Tzolov
- Fokus: Einfache Integration von LLMs in Spring-Anwendungen
- Inspiriert von LangChain und LlamaIndex
- Aktuelle Version: Spring AI 1.0.0-M8 / 1.0.0-SNAPSHOT
- Basiert aktuell auf Spring Framework 6.2 und Spring Boot 3.4

Langchain4j

- Start: Anfang 2023, inspiriert vom Python-Original
- Fokus: Einfache Integration von LLMs in Java-Anwendungen
- Aktuelle Version: Langchain4j 1.0.0-beta / 1.0.0-rc1
- Integration mit Spring Boot möglich, Quarkus und Java EE
- Aktive Open-Source-Community



AI & JAVA

- KI wird allgegenwärtig in der IT-Landschaft
- Java ist die Sprache der Unternehmen
- Die Entwicklung von Java-KI-Anwendungen ist eine neue Anforderung
- Modernisierung bestehender Java-Anwendungen
- Neue Use Cases, Automatisierung von Prozessen
- Starke Integration mit unstrukturierten Daten
- Treibt Innovation, Time-to-Market, Entwickler-Produktivität
- Faktische Standardisierung von AI-Modellen und Schnittstellen
- AI-Community und Ökosystem





Ausblick

Patrick Baumgartner, Sandra Parsick, Peter Roßbach,
Lars Vogel, Eberhard Wolff, Sebastian Meyen