



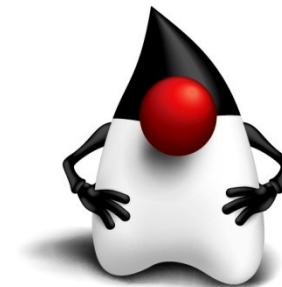
Java - Quo vadis?

ORACLE®

JAVA FORUM 2011
stuttgart am 7. Juli 2011

Die Java Plattform Strategie

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technologie und Architektur

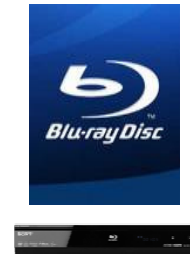


Java in Zahlen und Fakten



- 1.1 Mrd. Desktops verwenden Java
- 930 Millionen Java Runtime Environment Downloads pro Jahr
- 3 Mrd. Mobile Telefone verwenden Java
- 31-fach höhere Auslieferung von Java Telefonen pro Jahr im Vergleich zu Apple und Android zusammen
- 100 Prozent aller Blu-Ray Player verwenden Java
- 1.4 Mrd. Java Cards werden jedes Jahr hergestellt

Die Java Plattform



APIs

Java EE

JavaFX

Java SE

Java ME & Java TV

Java Card

Java VM

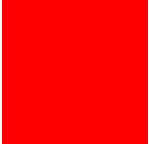
Hotspot Java VM

Lightweight Java VM

Language

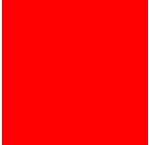
Java Language

ORACLE®



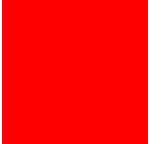
Was die Zukunft bringt

- **Java Strategie Treiber**
- **JDK 7 Review**
- **JDK 8 und darüber hinaus**
- **FX & Mobile**
- **Java Community**



Java Strategie

- Höchste Priorität: Java muß leben!
 - Zunehmendes Investment in die Plattform
 - Kontinuierliche Bewegung in Richtung "free & open" (OpenJDK)
 - Verbesselter Support für Entwickler Community, Open Source Community und JCP
- Niedrigere Priorität: Direkter Umsatz mit Java
 - Support für Oracle Software und Hardware
 - Mehrwert auf Basis der Java Plattform, Enterprise Support



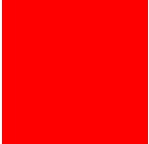
JVM Strategie: HotSpot und JRockit Konvergenz

- Existenz von zwei Java Virtuellen Maschinen
 - HotSpot – Allgemeine Einsatzfähigkeit, starke Verbreitung, hohe Qualität und Performance
 - JRockit – Spezialisierte Server-JVM mit optimierter Performance für den Server-Bereich und Oracle FMW
 - JRockit JVM als Basis für Mehrwertprodukte JRockit Mission Control, JRockit Real Time und JRockit Virtual Edition
- Vorhandene JRockit-basierte Produkte, wie JRockit Mission Control, JRockit RT, JRockit VE bleiben separat und werden weiterhin kommerziell lizenziert



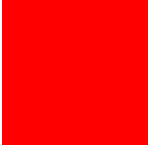
JVM Strategie: HotSpot und JRockit Konvergenz

- JRockit und HotSpot werden in einem mehrjährigen Prozess zu einer einheitlichen JVM verschmolzen: *“HotRockit”*
 - unter Berücksichtigung der besten Funktionsmerkmale beider JVM's
- Die Arbeitsergebnisse fließen inkrementell zum OpenJDK
 - JRockit Performance Funktionalität - Bereits im JDK 7
- Ziele
 - Maximale Performance und Skalierbarkeit
 - Multi-Core und Parallelisierungs Optimierungen
 - Feingranulares Monitoring, Profiling und Management
 - Extrem effiziente Garbage Collection



Evolutionäre Weiterentwicklung der Sprache

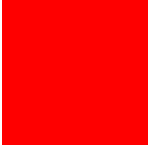
- **Langzeit Perspektive (20+ Jahre)**
 - Ideen reifen in ausgeprägten experimentellen Sprachen
 - Abwärtskompatibilität bleibt extrem wichtig für die Sprache
- **Periodische Anreicherung von
ausgewählter Funktionalität**
 - Verbesserte Entwicklerproduktivität
 - Bewahrt Klarheit und Einfachheit



JDK 7 Zeitplan

Open-Source Implementierung der Java SE Platform JSR 336

23. Dezember 2010	Feature Complete (M11)
17. Februar 2011	Developer Preview (M12)
12. April 2011	Ramp down start: P1-P3 bugs only
28. April 2011	API/interface changes: Showstoppers only
11. Mai 2011	All targeted bugs addressed
18. Mai 2011	Bug fixes: Showstoppers only
2. Juni 2011	Last scheduled build (M13)
	Final test cycle started
28. Juli 2011	General Availability



JDK 7 Funktionsumfang

- Coin – Kleine Sprachverbesserungen (JSR 334)
- Dynamic Language Support (JSR 292)
- Concurrency und Collections Updates (JSR 166y)
- Netzwerk und File System (JSR 203)
- Sicherheit
- Internationalisierung
- Weitere Verbesserungen
- JVM Konvergenz

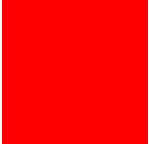




JDK 8 Ausrichtung

- Modularisierung
- Projekt Lambda (JSR 335)
- Annotationen on Java Types (JSR 308)
- Weitere kleine Sprachverbesserungen (Projekt Coin Teil 2)

8



JDK 9 und darüber hinaus

Die Oracle Wunschliste

- Zurück zum Ursprung: Zur tatsächlichen Wirkung der Plattform-Unabhängigkeit von Java
- Verbesserte Interoperabilität mit Non-Java, nativen Sprachen
- Verbesserte Datenintegration von SQL über Name/Value Stores zu Online Feeds
- Verbesserte Device Unterstützung
- Und vieles mehr ... die Arbeiten haben gerade erst angefangen!

9+



JavaFX

- **Java/JavaFX als geeignete Plattform für Rich Client Appli.**
- **Einheitliche Applikationsentwicklung für native und Web-Lösungen**
 - Browser Plug-in, Web Start, Native Executables
 - Hardware Accelerated Graphics (DirectX, OpenGL)
 - Nahtloser DOM Zugriff zwischen HTML5 & Java
- **JavaFX 2.0 Plattform Sprachwechsel**
 - Java als native Sprache - anstatt JavaFX Script
 - JavaFX APIs werden in Java implementiert
 - Vorteile bei der Nutzung von Generics, Annotations und Multithreading für JavaFX
- **JavaFX 2.0 Release für 2011**
 - JavaFX 2.0 Public Beta 26. Mai 2011 und GA am 29. September 2011



Mobile Java

- **ME.next zur Plattformmodernisierung**
- **Integration von Web Technologien**
 - LWUIT xHTML Support
 - Web Content Rendering in Java ME
 - JavaScript Zugriff auf Java Geschäftslogik
- **Innovations-Vorsprung mit Feature Phones Services**
 - Portfolio aus besonderen und kritischen lokalen Mobile Value Added Services (MVAS)
 - Einfacher Zugriff & Auffinden von Apps
 - Lösungen für Device Content Management
- **Skalierbare, leistungsfähige Laufzeit-Lösungen**
 - Ausgezeichnete Java Performance, Footprint und Applikationsfähigkeiten - Java ME Runtime basiert auf MSA
 - Für alle Geräte: Vom Low-End Feature Phone bis zum Smartphone





Teilnahme erwünscht!

OpenJDK

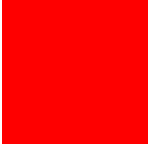
- Oracle verpflichtet sich zur besten Open-Source Java Implementation
- Weitere Software-Hersteller beteiligen sich:
 - “Oracle and **IBM** Collaborate to Accelerate Java Innovation Through OpenJDK”
 - “Oracle and **Apple** Announce OpenJDK Project for Mac OS X”
 - “**Red Hat** and Sun (Oracle) Collaborate to Advance Open Source Java Technology”
- Zwei neue OpenJDK Releases in 2011 und 2012
 - Liste der Features unter openjdk.java.net/projects/jdk7/features





Neue Richtlinien für OpenJDK-Teilnahme

- Community Richtlinien (Draft) in Zusammenarbeit mit
 - John Duimovich und Jason Gartner (IBM)
 - Mike Milinkovich (Eclipse)
 - Prof. Doug Lea (State University NY Oswego)
 - Adam Messinger (Oracle)
- Ziel der OpenJDK Richtlinien
 - Fördern vom langfristigen Bestand & Wachstum der Community
 - Basis stellen für offene, transparente und leistungsbezogene Aktivitäten der Mitglieder
- Beschreibung
 - Allgemeine Rollen, Abstimmung, Gruppen & Gruppenrollen, Projekte & Projektrollen, OpenJDK Teilnehmer, Governing Board, Reporting und Entscheidungswesen



JCP Checkliste für mehr Klarheit

- Namen der Expert Group Mitglieder
- Expert Group Aktivitäten berichten an öffentlich einsehbaren Alias
- Zeitplan ist öffentlich und aktuell mit regelmäßigen Anpassungen
- Öffentliche Teilnahme über read/write Wiki
- Diskussions-Board auf jcp.org
- Öffentlicher Read-Only Issue Tracker zur Nachverfolgung
- Ausrichtung einer neuen JCP Version mit JCP.next

NetBeans IDE 7.0

- **Entwicklungswerkzeug für Desktop, mobile und Web Anwendungen**

- Java EE 6, Java SE 6, Swing, (Java FX)
- Java ME und Embedded
- PHP, Ruby, Groovy, C / C++
- HTML5 Editing, JSON Formatter
- Debugger, Profiler, Refactoring



- Läuft auf MS Windows, Linux, Mac OS X und Solaris
- **NetBeans IDE ist open-source und frei verfügbar**
- **Nächstes NetBeans Release 7.1 in 2011**
- Feature Liste & Builds
 - netbeans.org/community/releases/roadmap.html

NetBeans IDE 7.0

- **JDK 7**

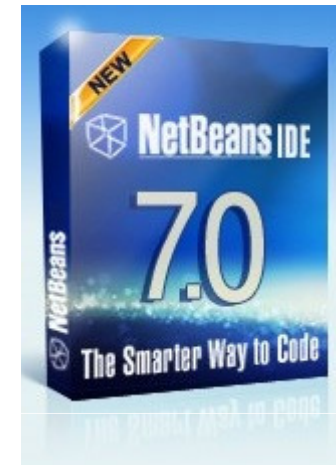
- Project Coin Unterstützung
- Editor Erweiterungen: Code Completion, Hints

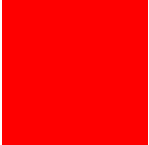
- **Java**

- Maven 3 Support
- Verbesserte Visual Customizer für GridBagLayout

- **Java EE**

- Bessere Unterstützung für CDI, REST, JPA
- Unterstützung für Bean Validation
- Unterstützung für JSF Comp Libraries, PrimeFaces Bundled
- Bessere JSF EL Verarbeitung mit Code Completion





Unterstützung für eclipse IDE

- Eclipse Projekte mit Werkzeugen und Frameworks
- Ganzheitliche Unterstützung vom Software Development Lebenszyklus
 - Modellierung
 - Entwicklung
 - Deployment Werkzeuge
 - Reporting
 - Daten Manipulation
 - Testing und Profiling
- Primärer Fokus zur Entwicklung von Anwendungen
 - Java EE, Web Services und Web Anwendungen
- Eclipse Unterstützung für andere Programmiersprachen
 - C/C++, PHP, andere



GlassFish 3.1 Applikationsserver

- **GlassFish Server Open Source Edition** 
 - Freier Open Source Applikationsserver mit Community-Support
 - Volle Java EE 6 Plattform Unterstützung
 - Modulare und erweiterbare Architektur (OSGi)
 - Clustering
- **Oracle GlassFish Server** 
 - GlassFish Server Distribution mit Kunden-Support
 - Open Source Edition & GlassFish Server Control enthalten
 - Active Cache – Native Integration mit Oracle Coherence*Web
 - Verwendbar zur Evaluation via OTN Lizenz
 - Benötigt eine Lizenz zur produktiven Nutzung

GlassFish und WebLogic

Java Foundation und Community



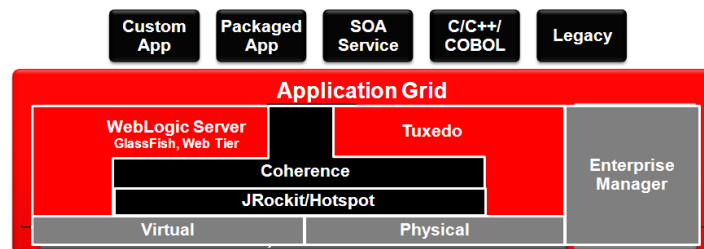
Java Entwicklung und Deployment



Fusion Middleware Foundation



Oracle Middleware Application Grid Infrastruktur



Optimierte ExaLogic Hardware/Software Plattform



ORACLE®

Java EE 6 Technologie

Java EE 6: Flexibilität, Erweiterbarkeit und Entwicklereffizienz

- **Neue Funktionalität**

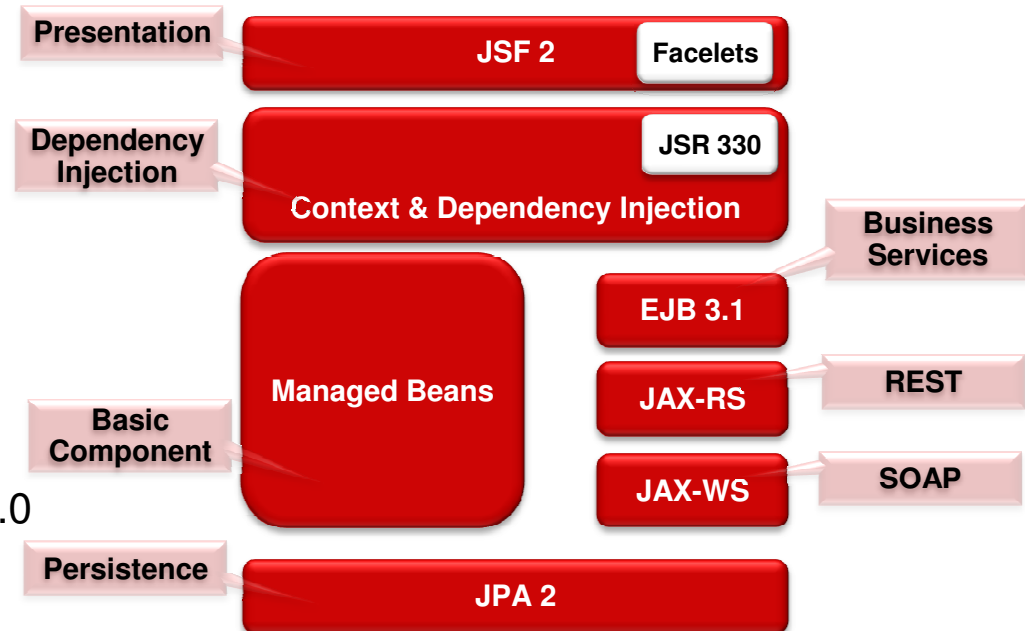
- Web Profile
- Erweiterbarkeit/Plugin-fähig
- Context und Dependency Injection, Managed Beans
- REST, Validation

- **Erweiterte APIs**

- EJB 3.1, JPA 2.0, JSF 2.0, Servlet 3.0

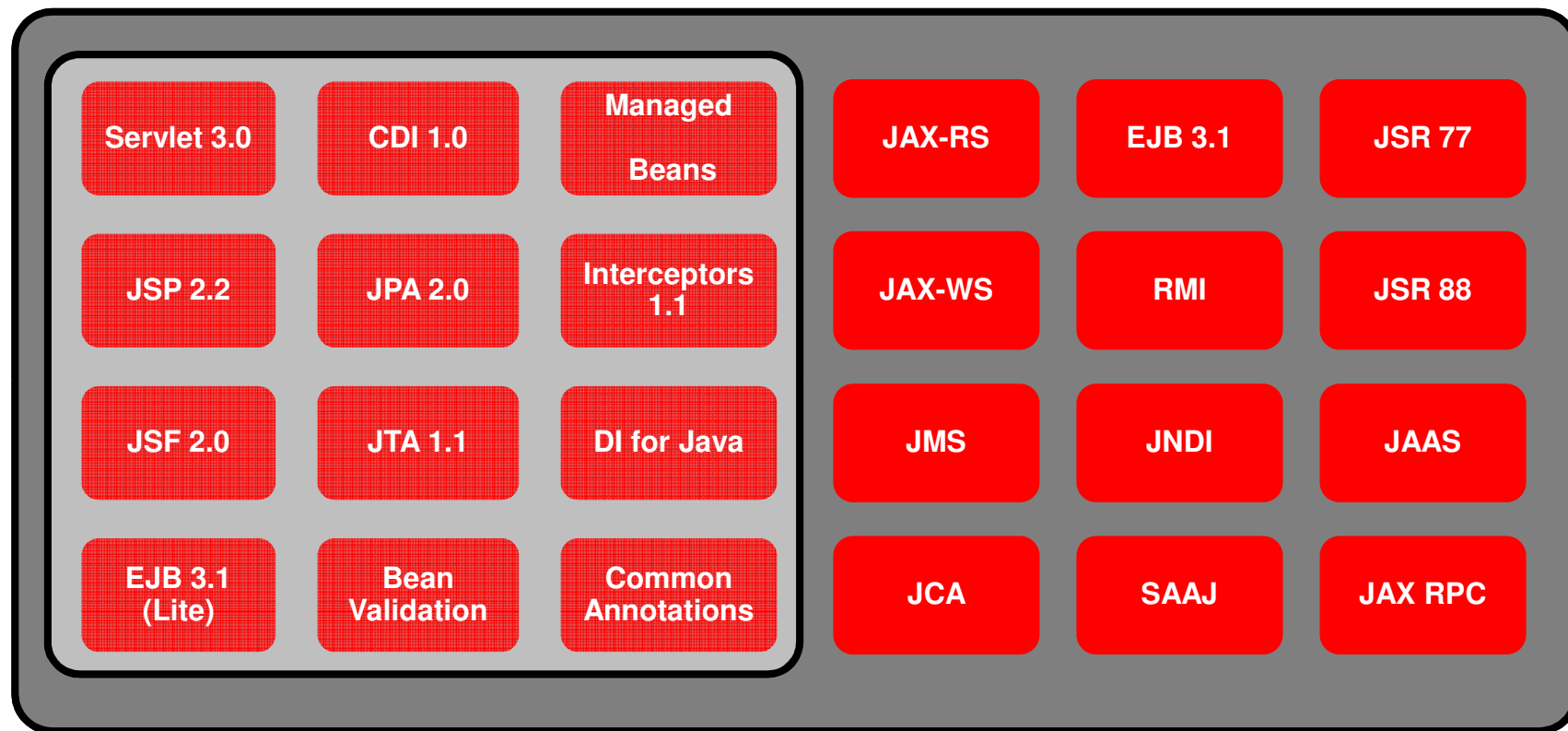
- **Verbesserte Nutzbarkeit und vereinfachte Konfiguration**

- Konventionen über Konfigurationen (weniger XML)
- Annotationen-basiertes Programmiermodell (Decorate und Inject)
- POJO Modell - Managed Beans zum Test von Komponenten
- Bessere Testmöglichkeiten und Zusammenspiel mit Betriebssystem-Werkzeugen
 - Ant / Maven / Hudson
- Keine Deployment Deskriptoren (optional)



Java EE 6 Web Profile

Web Profile



Vollständige Plattform

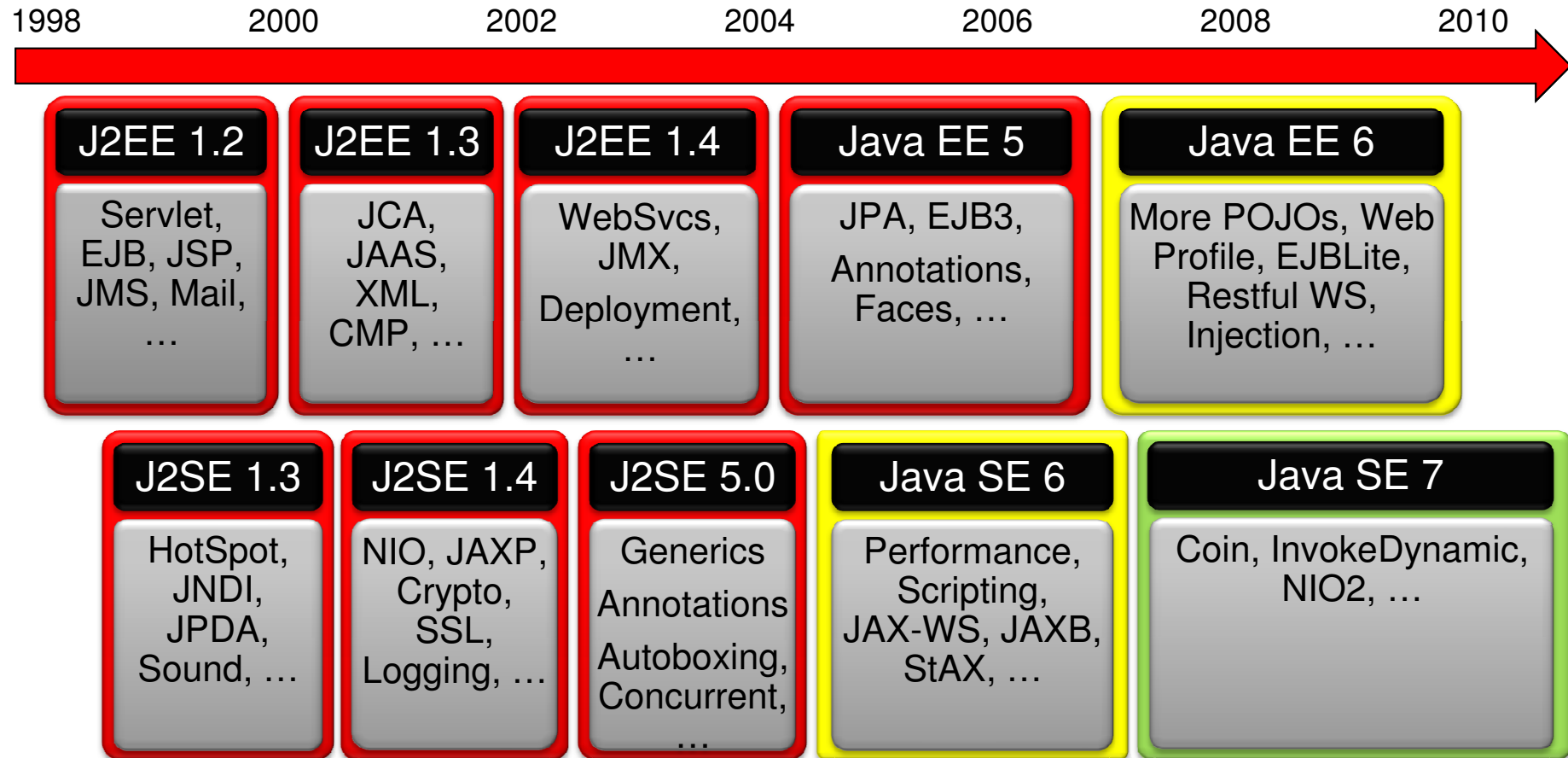
Java EE 6 Neuerungen (1)

- Einführung von Java EE Profilen für Web-Anwendungen
- Verbesserte Konfiguration mit durchgehender Unterstützung von Annotationen, und weniger XML-Konfiguration
- Verbesserte Basis API's (JSF 2.0, Servlet 3.0, JPA 2.0, ..)
- Einführung eines optionalen, leichtgewichtigen Komponentenmodells mit "Contexts and Dependency Injection for the *Java* EE Platform" *CDI* (JSR 299 und JSR 330)
- RESTful Web Services Unterstützung durch JAX-RS
- Automatische Web-Framework-Einbindung durch automatischer Erkennung von Servlet API 3.0 Komponenten
- Expression Language 2.2 mit Benutzer-definierten Functions und Parameter Support (JSR-245 JSP 2.2)
- Unterstützung von asynchronen Calls für Stateless Session Beans und Servlets

Java EE 6 Neuerungen (2)

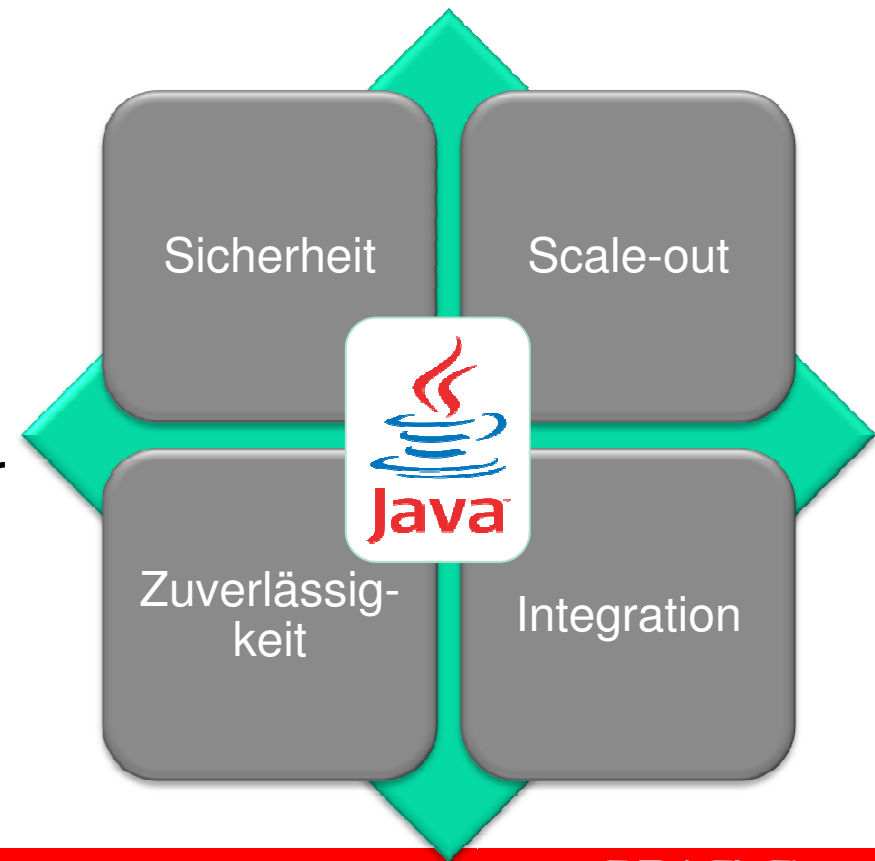
- Singleton EJB mit Container und Bean Managed Concurrency
- Criteria API in JPA 2.0 (dynamic, typesafe queries in Java)
- Schichtenübergreifende Validierung mit Beans Validation 1.0 (JSR-303)
- Verbessertes Testing von EJB's mit POJO-Ansatz und Embedded Container
- ProjectStage für JSF 2.0 Web-Anwendungen für Entwicklung, Unit Test, System Test und Produktion
- Source Code Debugging Support für anderer Sprachen, die in der virtuellen Maschine ablaufen
- Als *pruned* markierte API's und Spezifikationen können in zukünftigen Versionen optional werden (JAX-RPC, JAXR, JSR 88, EJB Entity Beans)

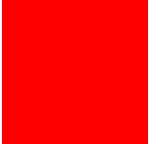
Der evolutionärer Pfad ..



.. und was dabei erkannt wurde

- Mehrwert liefern
- Innovation wahrnehmen
- Plattform muß sich verändern um lebensfähig zu bleiben
- Unternehmen verwenden weiter Java EE wegen technologischem Fortschritt der sich auszahlen muß
- Kunden profitieren vom Herstellerwettbewerb auf einer Standard Plattform





Zukünftiger Plan für Java EE 7

- **Schwerpunkt ist Zusammenspiel mit Cloud Plattformen**
 - Container-basiertes Modell, Abstraktion von Ressourcen, private und public Cloud Services, Mandantenfähigkeit, Elastizität (horizontale Skalierbarkeit) und Sicherheit
 - Bessere Trennung/Isolation von Java EE 7 Anwendungen
 - Potenzielle Standard Schnittstellen für “No SQL Datenbanken” (Not Only SQL), Caching, ..
 - HTML5 und WebSocket Unterstützung
 - Einheitliche Management- und Monitoring-Schnittstellen
- **Modularität und Versionierung**
 - Basierend auf den Arbeitsergebnissen der Java SE
 - Anwendungserstellung über Module
 - Explizite Abhängigkeiten und Versionierung
- **Evolution, anstatt Revolution**

Zustimmung für Java EE 7

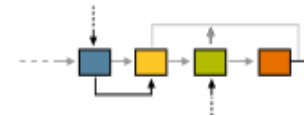
JSR 342: Java™ Platform, Enterprise Edition 7 (Java EE 7) Specification

SE/EE				Icon Legend					
Credit Suisse		Eclipse Foundation, Inc		Ericsson AB		Fujitsu Limited		Yes	
Google Inc.		Hewlett-Packard		IBM		Intel Corp.		No	
Keil, Werner		Oracle		RedHat		SAP AG		Abstain	
VMWare									

- Specification Lead:
 - Roberto Chinnici, Bill Shannon (Oracle)
- Expert Group im März 2011 etabliert
- Zeitplan für die Entwicklung definiert
 - Q3 2011 Early Draft
 - Q1 2012 Public Review
 - Q3 2012 Final Release

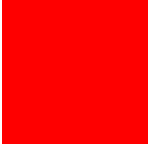
Community Development of Java Technology Specifications

- 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



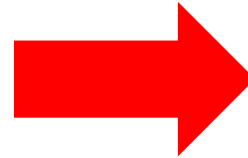
Entwickler möchten flexibel sein ..

- Entwickler für Unternehmensanwendungen wollen Cloud-Lösungen von der eigenen IT-Abteilung
 - IT liefert Service Level Agreement (SLA)
 - Entwickler fordern:
 - “Amazon-Modell”
 - Hosted Cloud Services
 - On-Demand Provisioning
 - Zuverlässige Skalierbarkeit
- Virtualisierung ist ein wertvolles Mittel, aber keine Lösung
- Unternehmenskritische Lösungen basieren auf Java
- Java EE hat vergleichbare Herausforderungen für die IT bereits gelöst



Wie soll sich Java EE für die Cloud weiterentwickeln?

- Entwicklung
- Betrieb/Deployment
- Management



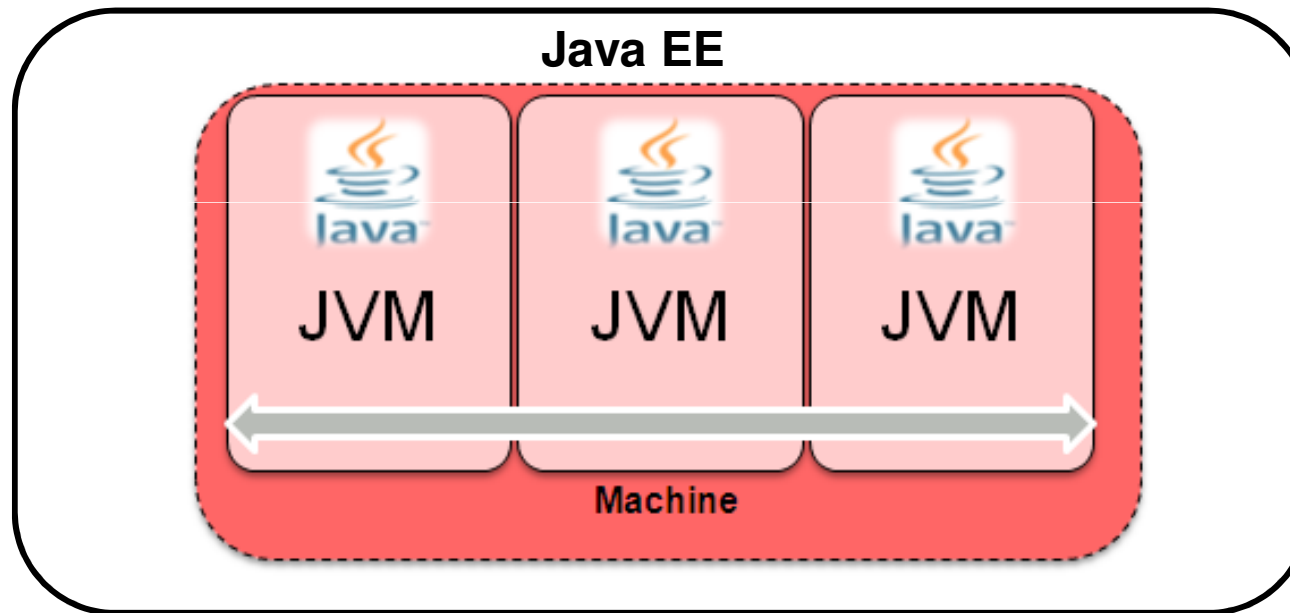
Veränderung

Java EE benutzt derzeit Rollen und Verantwortlichkeiten

Entwickler



Betrieb/Administrator



Container Provider



ORACLE®

Die Cloud benötigt Rechenzentrums- Rollen und Mandanten

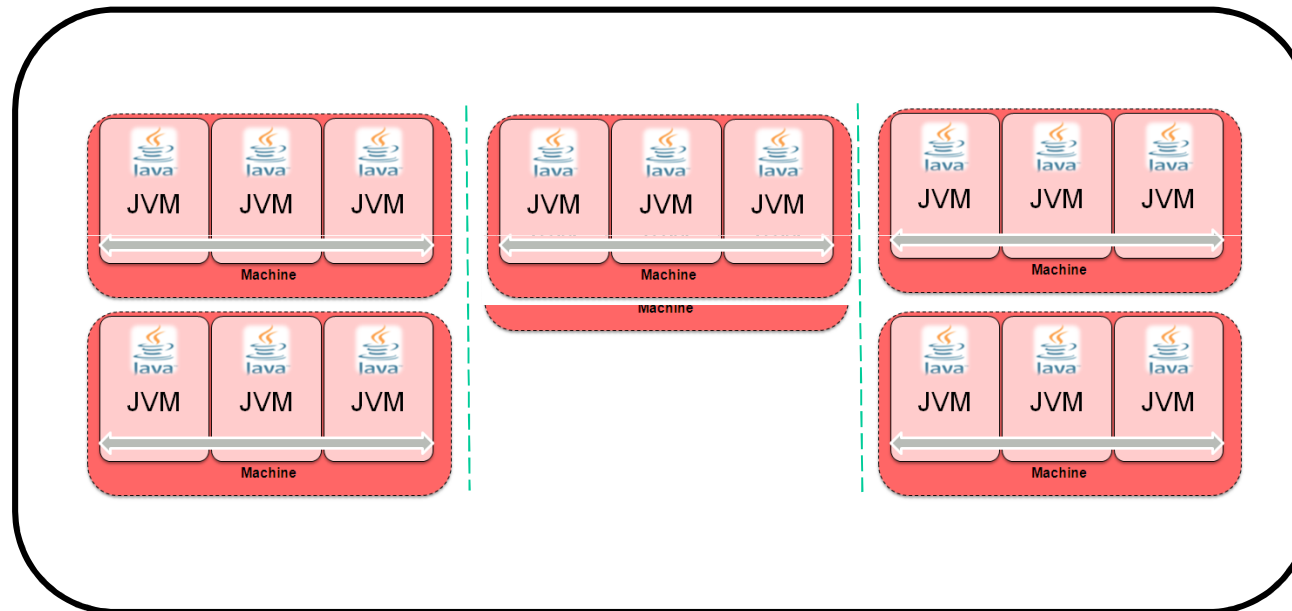
Entwickler



Anwendungs-Administrator



Container/Service
Provider



Applikations-
Deployment-
Betrieb



PaaS Administrator



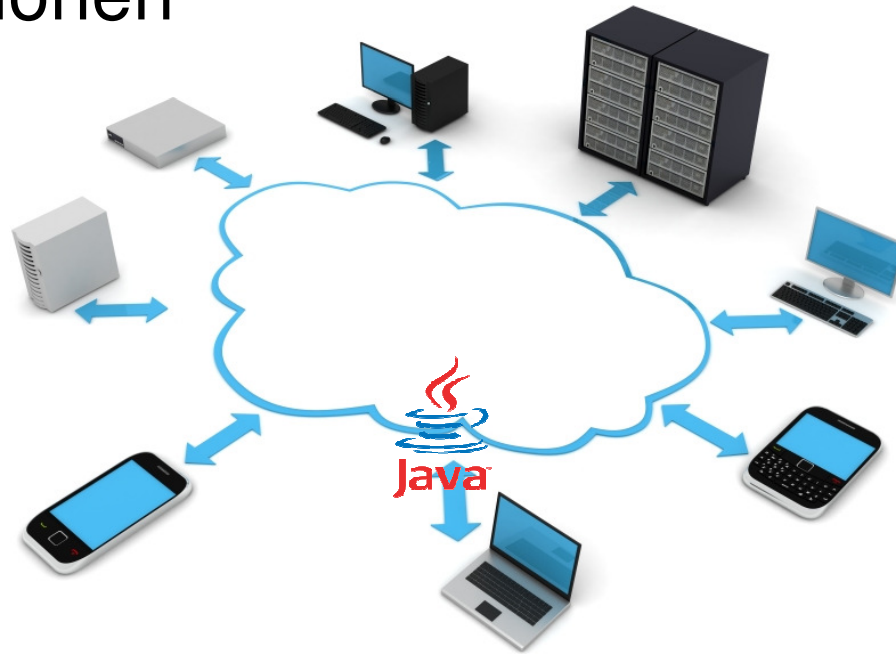
Modell mit Cloud Services – Keine APIs

- Mandanten verwenden Services als Verbraucher
- PaaS Administratoren für Bereitstellung, Verteilung, und Management von Services verantwortlich
- Existierende Java EE APIs müssen erneuert werden, damit sie Service-fähigkeit und Mandantenfähigkeit besitzen

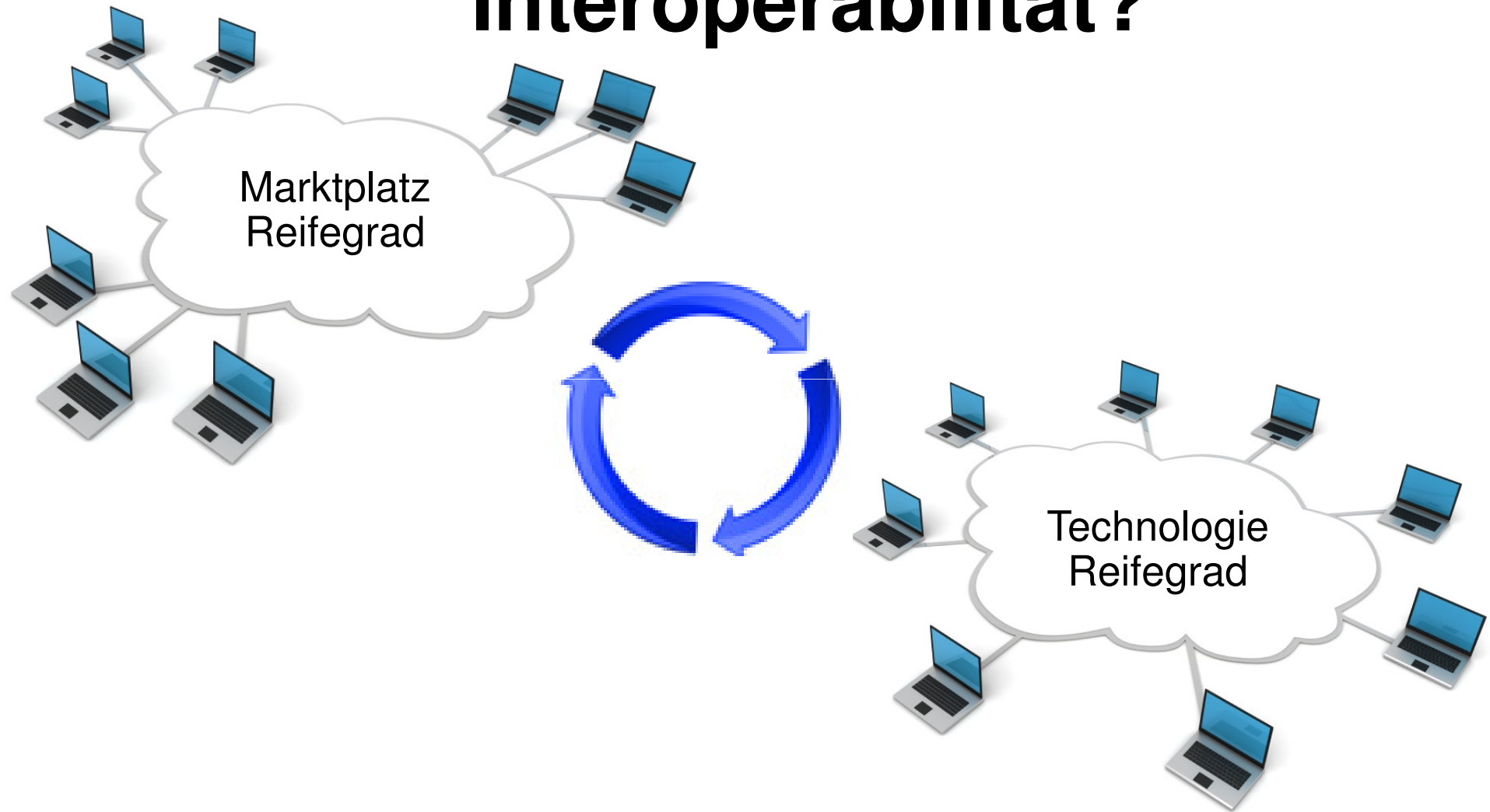


Java EE 7: Cloud Basis

- Neue formalisierte Rollen
- “Pluggable” Cloud Services, Keine APIs
- Revision der draunterliegenden Java EE Spezifikationen

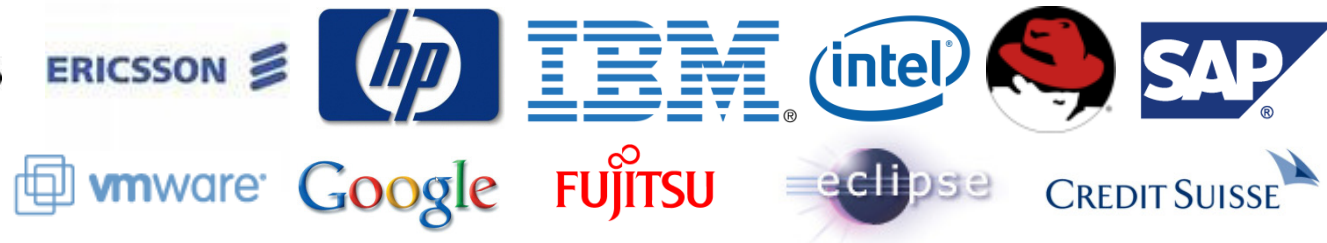
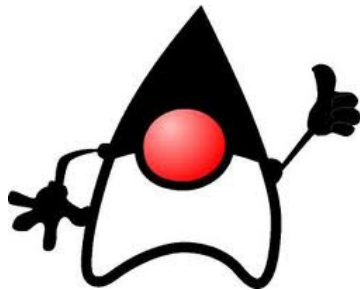


Java EE 8: Cloud Reifegrad und Interoperabilität?



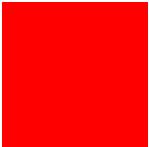
Welche Vorteile können wir daraus ziehen?

- ✓ Echten Mehrwert liefern
- ✓ Plattform nutzt Innovation um uns herum
- ✓ Zeitnahe Plattform-Evolution, Platz für mehr ..
- ✓ Unternehmen nutzen existierende EE Investitionen
- ✓ Supporters und Mitbewerber können den Marktplatz benutzen, ihn innovativ und erweiterungsfähig gestalten





ORACLE®

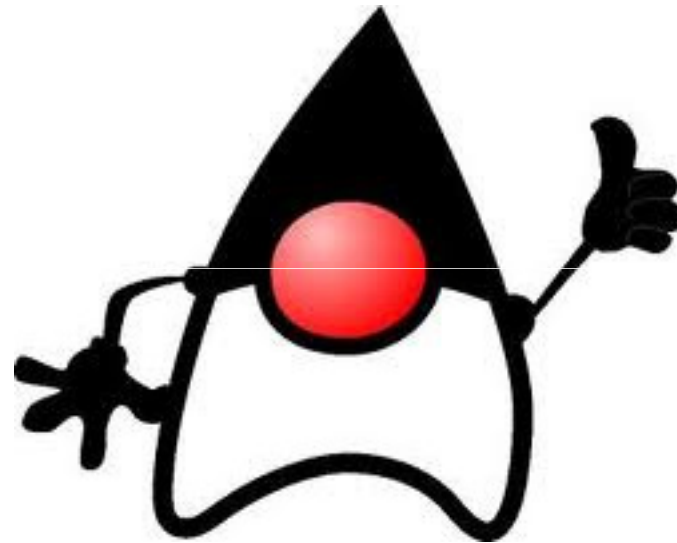


Vielen Dank für Ihre Aufmerksamkeit!

Wolfgang.Weigend@oracle.com



Java[™]



 **JAVA FORUM** *stuttgart* **2011**
am 7. Juli 2011

ORACLE[®]