

# ENTWICKLUNGS- UND LAUFZEITUMGEBUNG DER CSE: ECLIPSE UND JBOSS

Piotr Kasprzak

# Agenda



- Laufzeitumgebung
  - Java EE (J2EE)
    - Motivation
    - APIs / Technologien
  - JBoss
- Entwicklungsumgebung
  - Eclipse
  - Ausblick

# Java EE - Motivation

- Entwickler von unternehmenskritischen Applikationen stehen häufig vor denselben Herausforderungen:
  - ▣ Skalierbarkeit
  - ▣ Ausfallsicherheit
  - ▣ Verteilte Strukturen
  - ▣ Datenkonsistenz
  - ▣ Integration von „Legacy“-Systemen
  - ▣ Sicherheit

...unter den Bedingungen einer chronischen Ressourcenknappheit (Geld, Personal) und geforderter hoher Entwicklungsgeschwindigkeit / Flexibilität („Time to market“)
- Eine Entwicklungsplattform, die hilft, diesen Anforderungen gerecht zu werden, wäre hilfreich...

# Java EE



- Java EE definiert eine Plattform zur Entwicklung von komponentenbasierten, serverseitigen, verteilten Applikationen
  - ▣ Offen und standardisiert
  - ▣ Definiert eine Reihe von APIs und Technologien (Sicherheit, Transaktionen, Persistenz, Load-Balancing...)
  - ▣ Diese werden zur Laufzeit durch einen Applikation-Server bereitgestellt

# Java EE



- Vorteile:
  - ▣ Reduktion der Entwicklungskomplexität
    - Grundlegende Technologien für robuste, skalierende, verteilte Applikationen sind nun Teil der Infrastruktur - Das Rad muss nicht ständig neu erfunden werden!
    - Konzentration auf das eigentliche zu lösende Problem
    - Reduktion von Kosten
    - Hohe Entwicklungsgeschwindigkeit
  - ▣ Standardisierung verhindert „Vendor Lock-In“

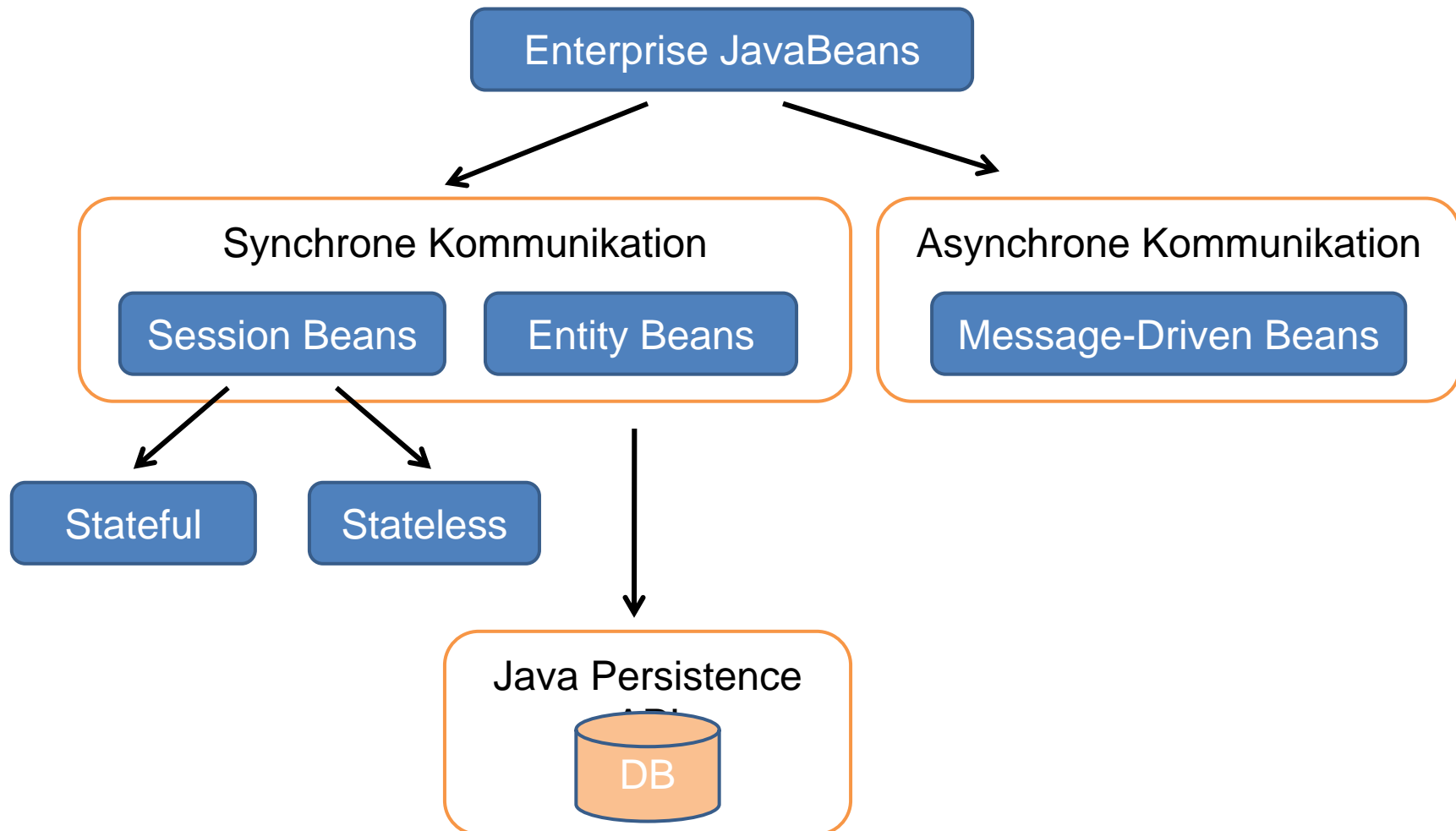
# Java EE - APIs / Technologien

API /	Beschreibung
EJB	Komponenten-Architektur zur Implementierung der Applikationslogik
JPA	Persistenz
JNDI	Verzeichnisdienst
JTA	Transaktionsverwaltung
JMS	Nachrichtenaustausch
JAXB	Verbinden von XML-Schema und Java-Klassen
JDBC	Zugriff auf relationale Datenbanken
JAX-WS, JAX-RPC, SAAJ	Implementierung von Web-Services
JAAS	Authentifikation / Autorisation
JAXP	XMP Bearbeitung
...	...

# Java EE – EJB3

- Enterprise JavaBeans (EJB3): Serverseitige Komponentenarchitektur
  - ▣ Kern von Java EE
  - ▣ Implementierung der Applikationslogik als Komponenten, die innerhalb eines Containers ausgeführt werden
  - ▣ Container kümmert sich um
    - Lebenszyklus der Komponenten (erzeugen, aktivieren / deaktivieren, zerstören)
    - Zugriff von außen (RMI)
    - Skalierbarkeit
    - Sicherheit
    - Persistenz
  - ▣ Verschiedene Arten von Komponenten

# Java EE – EJB3





# Java EE - JPA

- Java Persistence API (JPA) / Entity Beans:  
Bequeme Persistenz
  - ▣ Assoziation von Java-Klassen mit Datenbank-Tabellen (object-relational mapping (ORM)): Java-Objekte werden transparent in einer relationalen Datenbank persistent gehalten
  - ▣ Ursprünglich als Teil des EJB-Standards entwickelt, aber auch separat verwendbar
  - ▣ Unabhängigkeit von der verwendeten Datenbank
  - ▣ An SQL angelehnte Abfrage-Sprache: EJB-QL
    - Unterstützt objektorientierte Programmierung (z.B. Polymorphismus)

# Java EE - JMS

- Java Message Service (JMS): API für asynchrone Kommunikation über Nachrichtenaustausch
  - ▣ Senden / Empfangen von Nachrichten
  - ▣ Unterstützung verschiedener Kommunikationsmodelle
    - Point-to-Point:
      - Sender verschickt Nachricht an eine Warteschlange – Empfänger holt diese später ab
      - Direkte Beziehung zwischen Sender und Empfänger
    - Publish-Subscribe:
      - Ein oder mehrere Sender verschicken Nachrichten an eine spezielle Adresse („publish“),
      - Empfänger müssen sich für das Abholen der Nachrichten von dieser Adresse registrieren („subscribe“)
      - Fluktuation der Sender / Empfänger
      - Entkoppelte Kommunikation (Unabhängigkeit der Sender von den Empfängern)
  - ▣ Zustellungsgarantien für Empfänger: Nachricht wird genau einmal zugestellt
  - ▣ Transaktionsfähigkeit

# Java EE - JAXB



- Java Architecture for XML Binding (JAXB):  
Bequeme Verarbeitung von XML
  - ▣ Assoziation von XML-Schemata mit Java-Klassen
  - ▣ Automatisches Erzeugen von JAVA-Objekten aus XML-Dokumenten und umgekehrt
  - ▣ Java-Klassen können automatisch aus XML-Schemata erzeugt werden und umgekehrt
  - ▣ Flexibilität
    - Validierung optional
    - Anpassung der Umsetzung durch Annotationen / XML-Konfigurationsdateien

# JBoss



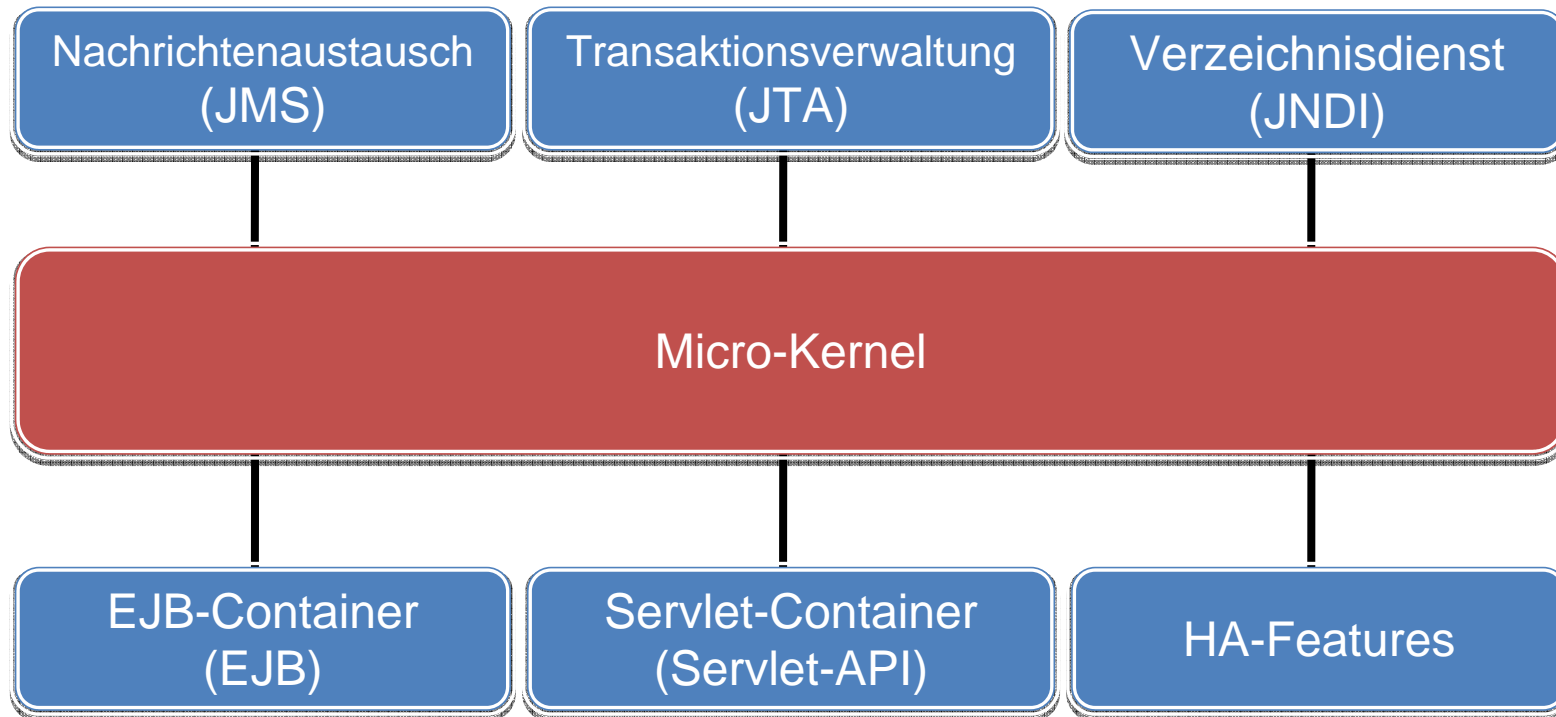
- Java EE Laufzeitumgebung wird durch einen Applikation-Server zur Verfügung gestellt:
  - Open-Source Implementierungen:
    - Apache Geronimo (Apache Foundation)
    - JBoss (Redhat)
    - Glassfish (Sun)
    - ...
  - Kommerzielle Implementierungen:
    - WebLogic Server (BEA)
    - WebSphere (IBM)
    - Oracle OC4J (Oracle Corporation)
    - ...

# JBoss



- Unsere Wahl fiel auf JBoss:
  - ▣ Open Source
    - Große Community
    - Gute Dokumentation
    - Transparenter Entwicklungsprozess
    - Schnelle Unterstützung neuer Standards
  - ▣ Große Verbreitung
  - ▣ Ausgereift
  - ▣ Hot-Deployment von Diensten und Applikationen
  - ▣ Kommerzieller Support
  - ▣ Micro-Kernel Architektur

# JBoss - Micro-Kernel- Architektur



# JBoss



- Vorteile der Micro-Kernel-Architektur:
  - ▣ Dienste lassen sich einfach ein- / ausschalten
    - Reduktion der Komplexität durch Abschalten nicht benötigter Dienste
    - Später bei Bedarf Aktivierung von Load-Balancing, HA-Features, zusätzlichen Schnittstellen, Caching etc.
  - ▣ Manche Dienste lassen sich auch außerhalb des Applikation-Server nutzen (EJB3 und JMS durch JBoss Micro-Container)

# Eclipse



- State-of-the-art Entwicklungsumgebung:
  - ▣ Open Source: ursprünglich von IBM entwickelt als Nachfolger von „IBM Visual Age for Java“, 2001 freigegeben
  - ▣ In Java implementiert: läuft unter Windows / Linux / Mac OS X
  - ▣ Am bekanntesten als Java-IDE
  - ▣ Wesentlich mehr als nur einfache IDE:  
Offene Plattform zur Entwicklung von IDEs /  
Container für Plugins:  
Java-IDE ist nur ein Plugin!



# Eclipse - Plugins

- Open-Source-Plugins:
  - Java / C / C++ / COBOL / PHP IDEs
  - SQL Development Tools
  - SOA / BPEL
  - GUI-Designer
  - UML
  - Mobile Linux
  - ...
- Kommerzielle Plugins:
  - BEA Workshop Studio
  - Borland Together Edition for Eclipse
  - IBM Rational Rose XDE Modeler
  - ...
- Momentan > 995 Plugins verfügbar  
(<http://www.eclipseplugincentral.com>)

# Eclipse



- Entwicklungsumgebung CSE
  - ▣ Java Development Tools (JDT)
  - ▣ JBoss Tools
    - Start / Stop von JBoss
    - JBoss Konfiguration / Status
    - (Remote) Debugging von JBoss
    - Packaging / Deployment von Java EE-Komponenten
    - Generierung von XML-Artefakten zur Konfiguration von Java-EE Komponenten
  - ▣ Web Tools Platform (WTP)
    - XML / XML-Schema Editoren
    - Web-Services / WSDL