

# Einführung in die Java / Jakarta Enterprise Edition

## Kapitel 1 - Übersicht

1

# ÜBERSICHT

1.1

## **BESTANDTEILE**

# Überblick: Die Java Enterprise Edition

## Jakarta EE Platform

### Jakarta EE Web Profile

### Jakarta EE Core Profile



COPYRIGHT (C) 2023, ECLIPSE FOUNDATION. | THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)

# Varianten

## ▪ Profile

- Core
- Web
- Platform

## ▪ Artefakttypen

- EAR - Enterprise Application Archive
  - kann WAR und JAR Dateien beinhalten
- WAR - Web Application Archive
- RAR - Resource Archive
- JAR - Java Archive

## ▪ Auslieferungsmöglichkeiten

- Installation von WAR oder EAR Dateien auf einen JEE Server
- Fat-JAR/Uber-JAR, welches Anwendung + JEE-Plattform enthält
  - sinnvoll für Installationen als Docker-Image (z.B. für Kubernetes Cluster) Was sind Ihre Erwartungen?

- Connectors
  - Über Inbound Connectors kommunizieren entfernte Clients mit dem Applikationsserver
    - Sie öffnen einen abhörenden Netzwerk-Socket
    - Jeder ankommende Request wird der passenden Anwendung zugeordnet
  - Outbound Connectors binden externe Systeme an
    - Dazu baut der Outbound Connector Socket-Verbindungen auf, die in einem Connection Pool gehalten werden
  - Zusätzliche Inbound und Outbound Connectors können installiert werden

- Enterprise JavaBeans
  - Kümmern sich um Authentifizierung und Autorisierung
  - Realisieren horizontale und vertikale Skalierbarkeit
  - Können direkt durch gängige Kommunikationsprotokolle angesprochen werden
    - und machen damit die Anwendung verteilbar

**Enterprise JavaBeans sind technische Komponenten, aber mittlerweile so einfach zu definieren, dass sie problemlos auch direkt im Anwendungs-programm benutzt werden können!**

- Servlets
  - Kümmern sich um Authentifizierung und Autorisierung
  - Realisieren horizontale und vertikale Skalierbarkeit
  - Können durch das http/https-Protokolle angesprochen werden
    - Ursprünglich rein Request-Response-orientiert
    - Ab der JEE 7 auch asynchrone Aufrufe mit Server-Callbacks möglich
      - Unterstützung von Web Sockets

**Servlets bieten ein mächtiges Low-Level-API und werden von Frameworks wie JavaServer Faces oder Web Services intern benutzt.**



- Context & Dependency Injection (CDI)
  - Der Context erzeugt sämtliche relevante Fachklassen
    - Die Lebensdauer wird durch "Scopes" definiert
  - Setzt deren Abhängigkeiten
  - Macht die Fachlogik transaktionsfähig
  - Realisieren mit Interceptors Anwendungsfallübergreifende Querschnittsfunktionen
    - Aspekt-orientierte Programmierung
  - Stellt einen Event-Bus zur losen Kommunikation zur Verfügung

**Mit CDI und Interceptors unterstützt die JEE die neuesten anerkannten Design-Regeln für die Anwendungsprogrammierung!**

- **JavaServer Faces**
  - Bilden Browser-basierte Anwendungen in Komponenten ab
  - Abstrahieren und Kapseln die technischen Details der http-Technologie
  - Stellen mit einem Event-Modell, AJAX-Unterstützung und Navigationsregeln ein hochwertiges Programmier-Modell zur Verfügung
  - Erstellen mit Layouts Web-Seiten mit einheitlicher Gestaltung
  - Auch JSF-Komponenten sind automatisch CDI-Konform

**Die direkte Verwendung eines Servlets zur Programmierung von Web Anwendungen ist in der Regel viel zu kompliziert!**

- Java Persistence API (JPA)
  - Ein EntityManager verwaltet Entity-Objekte
    - Create-, Read-, Update-, Delete-Metoden
    - Synchronisation der Entity-Objekte mit der Datenbank-Einträgen
    - Objektorientierte Abfragen mit Objekt-orientiertem SQL und Criteria-Objekten
  - Auch direkte SQL-Befehle können ausgeführt werden

**Das JPA macht die direkte Verwendung des JDBC-APIs `java.sql` überflüssig.**

- Verteilte Anwendungen
  - Remote Method Invocation (RMI)
    - Notwendig ist nur ein Remote-kompatibles Interface
  - http/https
    - Servlet-API mit `HttpServletRequest`, `HttpServletResponse`, `HttpSession`,  
...
  - Messaging
    - JMS-API
  - SOAP-basierte Web Services
    - JAX-WS
  - RESTful Web Services
    - JAX-RS
  - Anbindung an Mail-Server

- Scheduling
  - Definition von wiederkehrenden Aktionen mit Timer und TimerTask
- Java Batch
  - Ein komplettes Job-basiertes Framework zur Ausführung komplexer Batch-Anwendungen
  - Definition durch die Job Specification Language
    - eine XML-basierte Beschreibungssprache

# 1. Übersicht - Wissenscheck

---

- Nennen Sie 6 Bestandteile von Jakarta EE
- Welche Profile gibt es. Nennen Sie pro Profil ein Bestandteil, was enthalten ist und eines was nicht enthalten ist.
- Welche Auslieferungsmöglichkeiten gibt es?

1.2

## **SPEZIFIKATION UND HERSTELLER**

- Eindeutige Definition des Laufzeitverhaltens und der zur Verfügung gestellten Dienste
- Entwickler und Architekten müssen für unterschiedliche Projekte/Kunden nur eine einzige Umgebung lernen
- JEE-Anwendungen laufen in allen JEE-konformen Applikationsserver
  - 100 % Pure Java läuft in "Java Compatible" Umgebungen
- Garantierte Abwärts-kompatibilität
- Herstellerunabhängigkeit garantiert fruchtbare Konkurrenz zwischen JEE-Anbietern



# Die Bürde einer Spezifikation

- JEE-Provider müssen einer formalen Spezifikation genügen
  - Damit ist die Frequenz von neuen Versionen und damit die Integration neuer Anforderungen und Features relativ gering
    - JEE 1.4 2001
    - JEE 5 2006 (!)
    - JEE 6 2009
    - JEE 7 2013
    - JEE 8 als JSR 366 in der Findungsphase
- Die Spezifikation erfolgt unter Mitarbeit verschiedener Hersteller und ist deshalb stets als Schnittmenge aktueller Technologien zu verstehen
  - Im Gegensatz dazu können unabhängige, Implementierungs-getriebene Frameworks jederzeit neue Features einbringen
    - Mit allen daraus resultierenden Vor- und Nachteilen!

- Kommerziell
  - IBM WebSphere
  - Oracle WebLogic
  - ...
- Open Source
  - JBoss
  - Apache Geronimo bzw. IBM Developer
  - Oracle Glassfish
  - Apache Tomcat
  - ...

# Aktuelle Jakarta EE 10 zertifizierte Applikationsserver

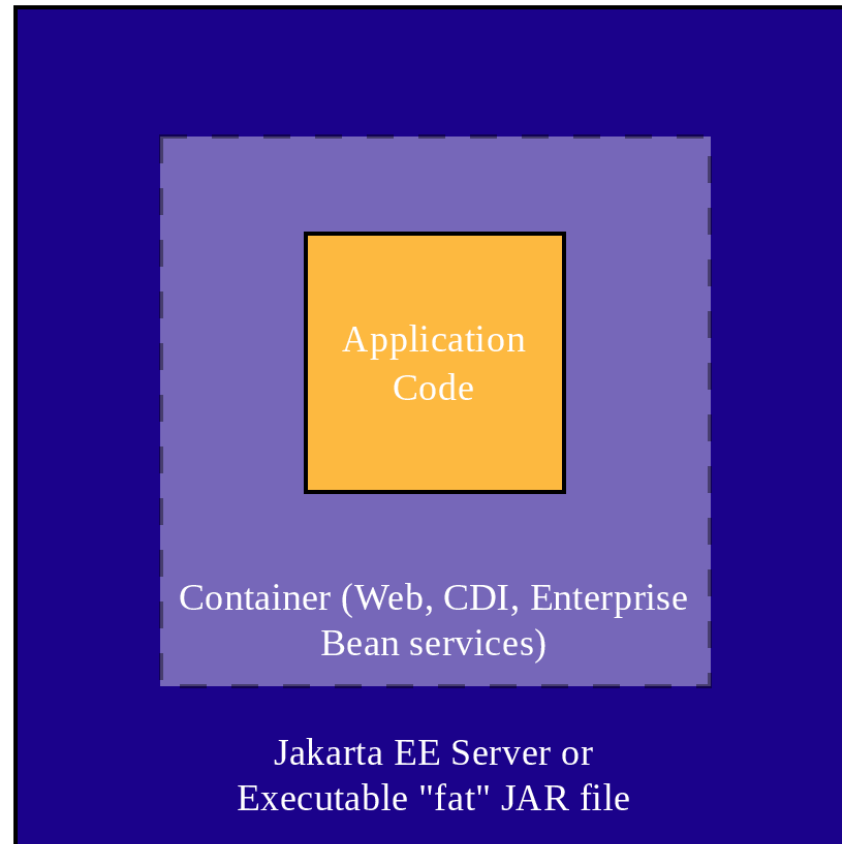
Product	Download
Eclipse GlassFish	7
FUJITSU Software Enterprise Application Platform	1.2.0 1.1.0
IBM WebSphere Liberty	23.0.0.3, Java 17 23.0.0.3, Java 11
Open Liberty	23.0.0.3, Java 17 23.0.0.3, Java 11
Payara Server Community	6.2023.12 6.2023.7 6.2022.1 6.2022.1.Alpha4
Payara Server Enterprise	6.9.0, Web Profile 6.9.0 6.4.0
JBoss Enterprise Application Platform	8.0.0, Java 17 8.0.0, Java 11
WildFly	27.0.0.Alpha5, Java SE 17 27.0.0.Alpha5, Java SE 11

<https://jakarta.ee/compatibility/download/> | Spezifikationen | Jakarta EE 10

1.3

# **ARCHITEKTUREN**

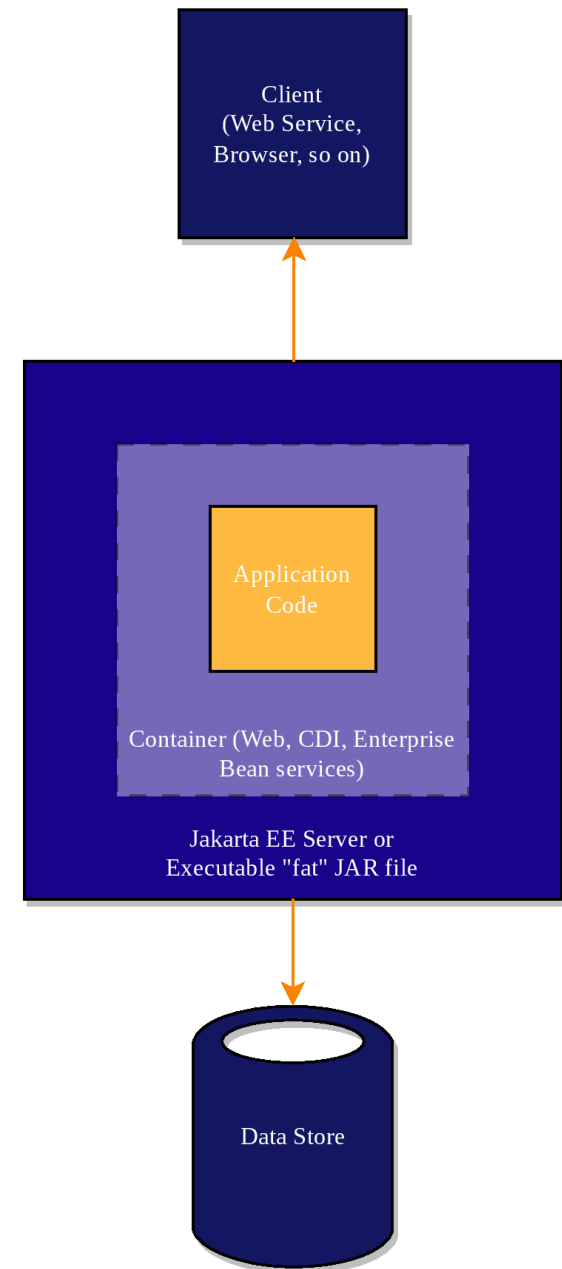
## 1.3 Architekturen – Applikation & Runtime



[https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#\\_footnotedef\\_3](https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#_footnotedef_3)

## 1.3 Architekturen – 3 Schicht

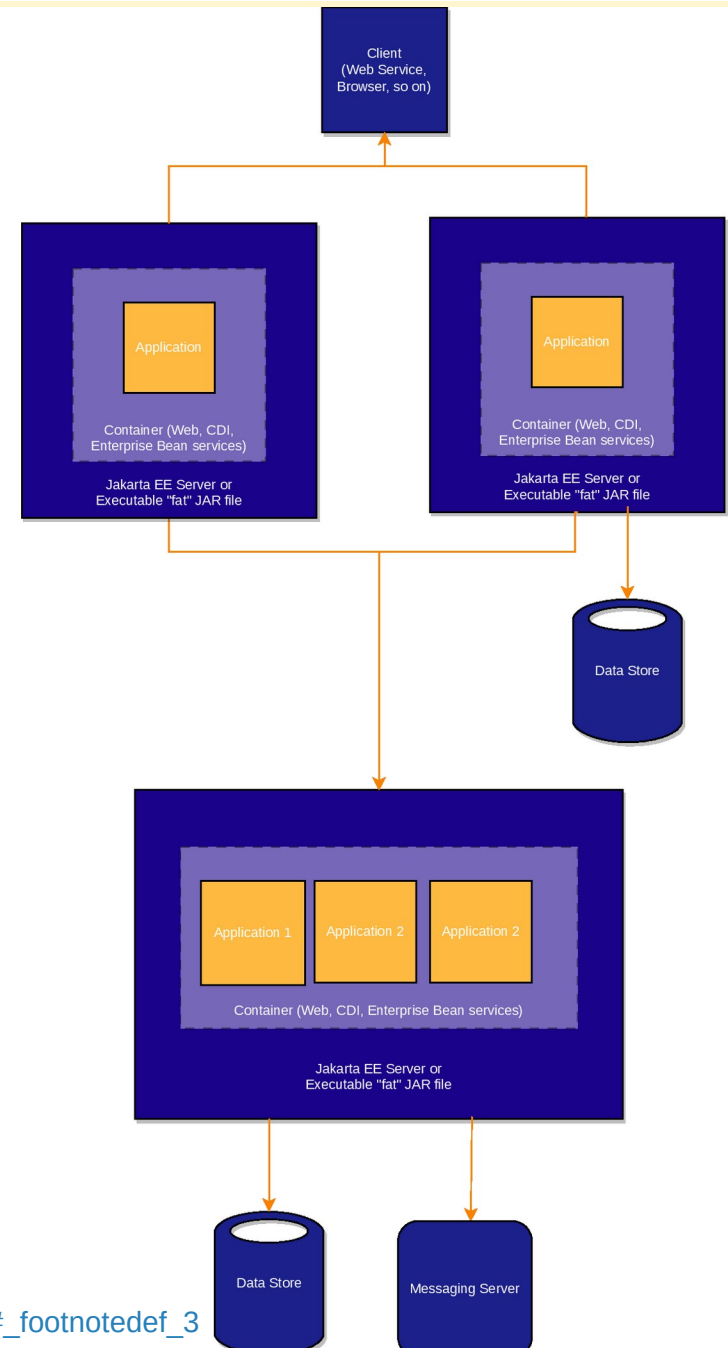
- Standardarchitektur für viele Anwendungen
- z.B. Self-Contained-Systems



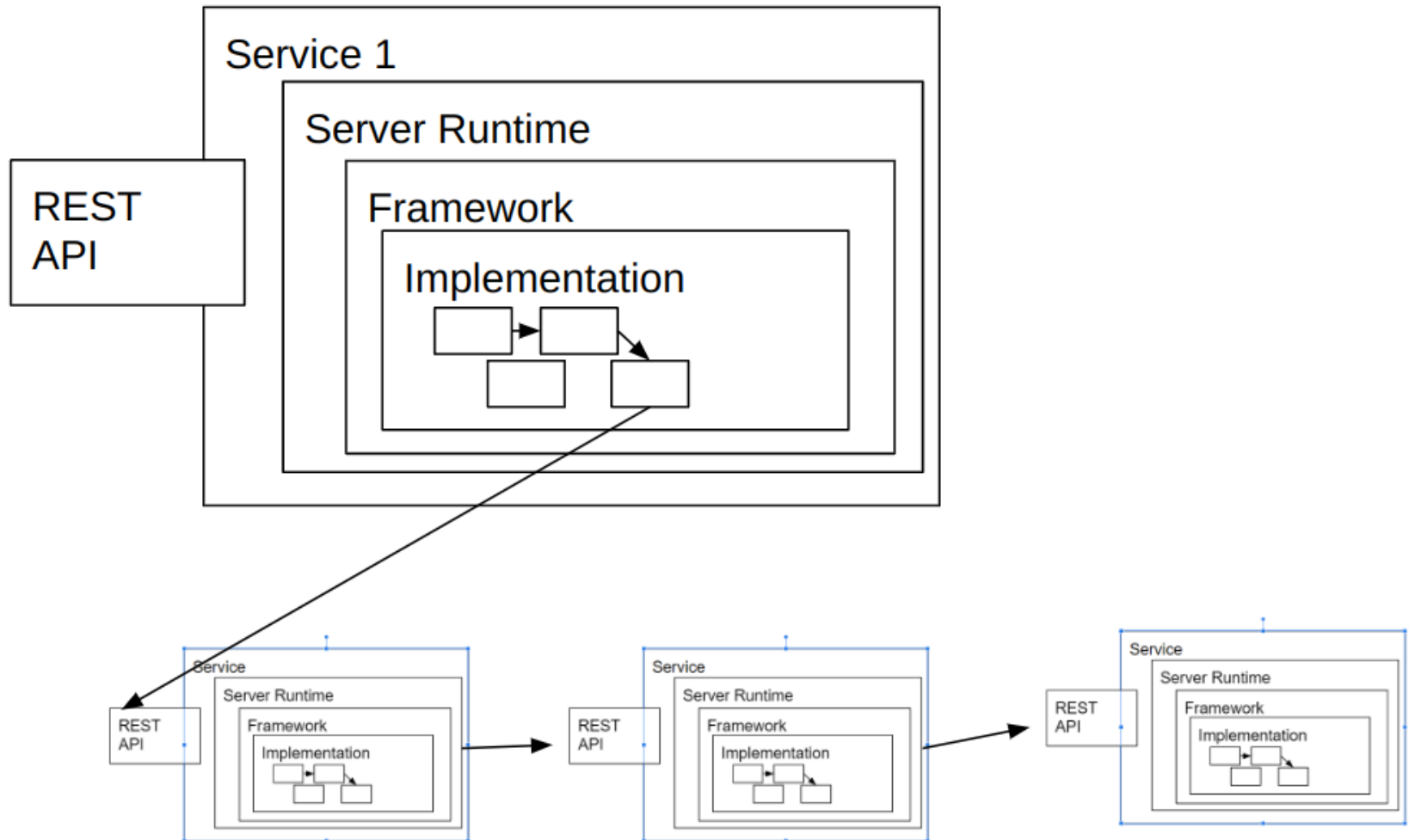
[https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#\\_footnotedef\\_3](https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#_footnotedef_3)

## 1.3 Architekturen – Multitier-App

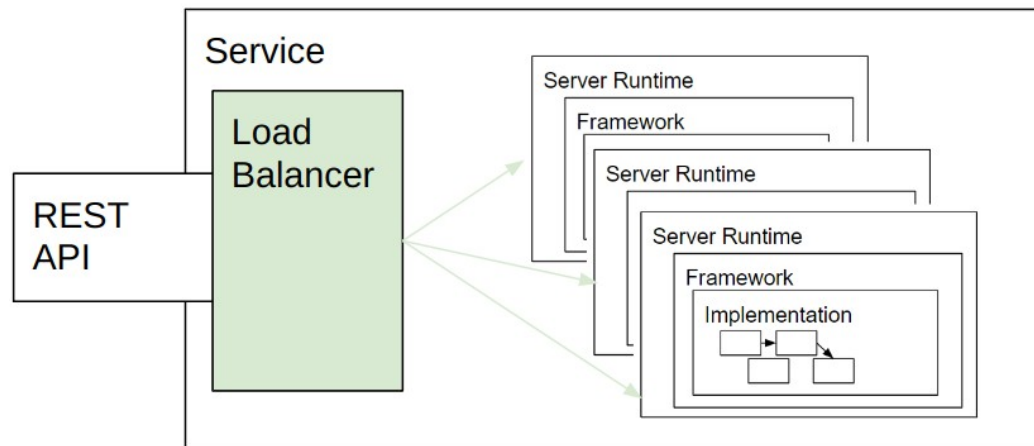
- z.B. Microservices
- Komplexität wird auf die Orchestrierungsebene der Services verschoben



[https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#\\_footnotedef\\_3](https://jakarta.ee/learn/docs/jakartaee-tutorial/current/intro/overview/overview.html#_footnotedef_3)







Orchestrierung und Monitoring

# 1. Übersicht - Wissenscheck

---

- Nennen Sie 6 Bestandteile von Jakarta EE
- Welche Profile gibt es. Nennen Sie pro Profil ein Bestandteil, was enthalten ist und eines was nicht enthalten ist.
- Welche Auslieferungsmöglichkeiten gibt es?
- Nennen Sie 2 Applikationsserver die Jakarta EE 10 zertifiziert sind
- Beschreiben Sie typische Architekturansätze im Umfeld von Jakarta EE