



# Java Enterprise Edition – Einführung

## Java EE Grundlagen

## Inhalte dieses Kapitels

Motivation

Konzepte

Architektur

Zusammenfassung

# Motivation

## Anforderungen an Enterprise-Anwendungen

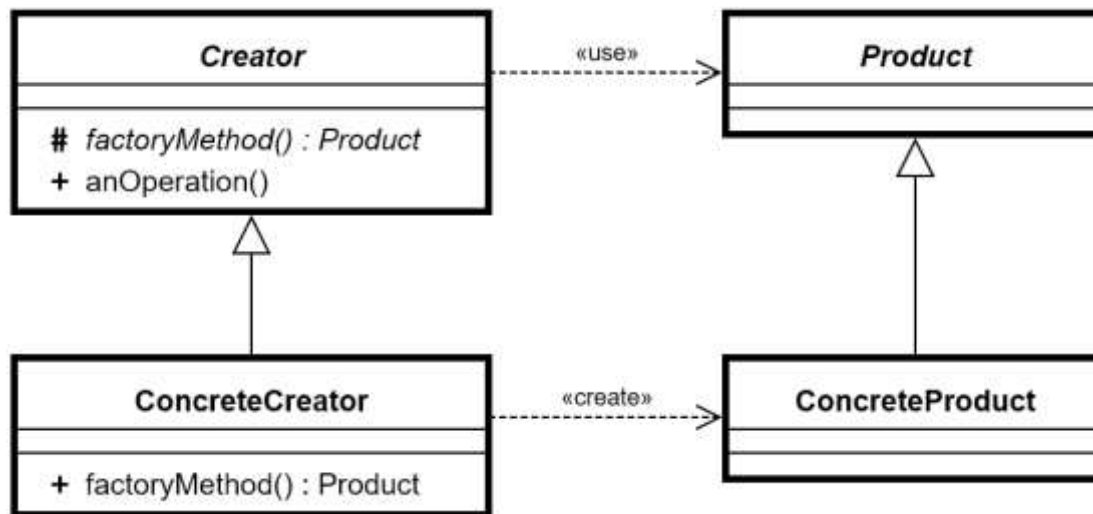
- Verteilte Architektur
- Parallele Verarbeitung
- Skalierbarkeit
- Ausfallsicherheit
- Modularität
- Persistenz
- Sicherheit
- ...

→ Umfang der Java Standard Edition nicht ausreichend

# Konzepte [1/7]

## Programmieren gegen Schnittstellen

- Spezifikation definiert *Interfaces*
  - zur Implementierung eigener Komponenten
  - zur Nutzung von Basisdiensten (Persistenz, Transaktionen, ...)
- Application Server als Implementierung der Basisdienste
  - Laufzeitumgebung als Notwendigkeit
  - Austauschbarkeit (Herstellerunabhängigkeit)



## Konzepte [2|7]

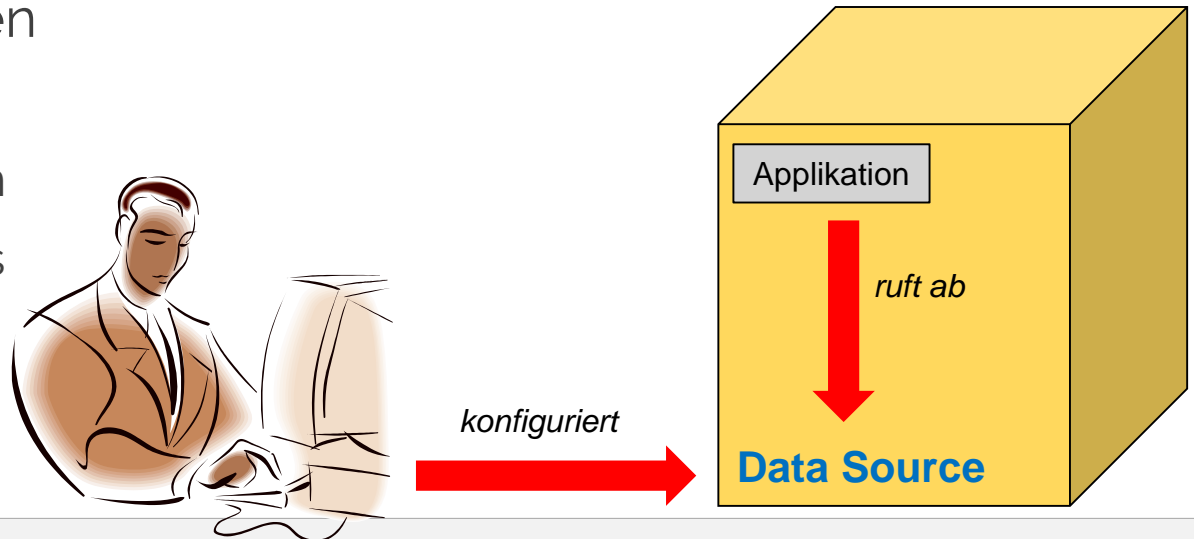
### Application Server

- Laufzeitumgebung für Java-EE-Anwendungen
- Bereitstellung von
  - Schnittstellen und Diensten nach Java-EE-Spezifikation
  - Schnittstellen zur
    - Konfiguration
    - Administration
    - Analyse
- Produkte
  - IBM WebSphere Application Server (WAS), wlp (*Liberty Profile*)
  - Red Hat WildFly, ehem. JBoss Application Server (*JBoss AS*)
  - Oracle GlassFish
  - OW2 JOnAS, BEA Weblogic, SAP NetWeaver Application Server
  - ...

# Konzepte [3/7]

## Konfigurierbarkeit

- Ressourcen werden am Application Server konfiguriert
- Application Server verwaltet Ressourcen (Lifecycle, Pooling)
- Applikationen können Ressourcen verwenden
  - Abfrage über Java Naming and Directory Interface (JNDI)
- Unterschiedliche Konfigurationen auf unterschiedlichen Zielsystemen
- Arten von Ressourcen
  - Data Sources
  - Umgebungsvariablen
  - JMS Queues / Topics
  - Mail Sessions
  - URLs



## Konzepte [4|7]

### Hollywood-Prinzip

- „*Don't call us, we call you!*“
- Entwicklung passiver Komponenten
  - Verwaltung (*Lifecycle Management*) und Aufruf durch Container
- Entwickler muss Komponente
  - implementieren (ggf. Interface als Vorlage)
  - registrieren (Annotation, alternativ: Deskriptor-Datei)
- Entwickler muss dabei wissen,
  - wann*
  - wie oft (parallel)*
  - in welcher Reihenfolge*
  - mit welchen Parametern*die von ihm geschriebenen Klassen instanziiert und deren Methoden aufgerufen werden

# Konzepte [5|7]

## Dependency Injection

- Komponente hat Abhängigkeiten (*Dependencies*) zur Umgebung
  - Ressourcen (Datenbanken, URLs, Mail Server, Variablen)
  - Referenzen auf austauschbare Schnittstellenimplementierungen

	ohne DI	mit DI
Ermitteln der Abhängigkeit	eigenständig	Übergabe von außen ( <i>Injection</i> ) durch DI Container
Austauschbarkeit der Abhängigkeit	schwer möglich	<i>Injection</i> einer anderen Implementierung → Programmieren gegen Schnittstellen
Lauffähigkeit der Komponente	nur in vollständiger Umgebung	in dedizierter Umgebung
Fehlersuche	einfach	schwer





## Konzepte [6|7]

### Dependency Injection – Umsetzung im Code

- Klassen müssen Injizieren benötigter Dienste, Ressourcen u.a. Informationen von außen ermöglichen
  - Instanzvariablen (*Field Injection*)
  - Kontruktorparameter (*Constructor Injection*)
  - Methodenparameter (*Parameter Injection*, *Setter Injection*)

```
public class TrainingModuleDAOImpl {  
  
    private DataSource ds;  
  
    public void setDataSource(DataSource ds) {  
        this.ds = ds;  
    }  
  
    // ...  
}
```

# Konzepte [7|7]

## Ressource Injection

- Annotation `@Resource`
  - JNDI-Lookup durch Container
  - Teil der *Common Annotations for Java* (JSR-250)

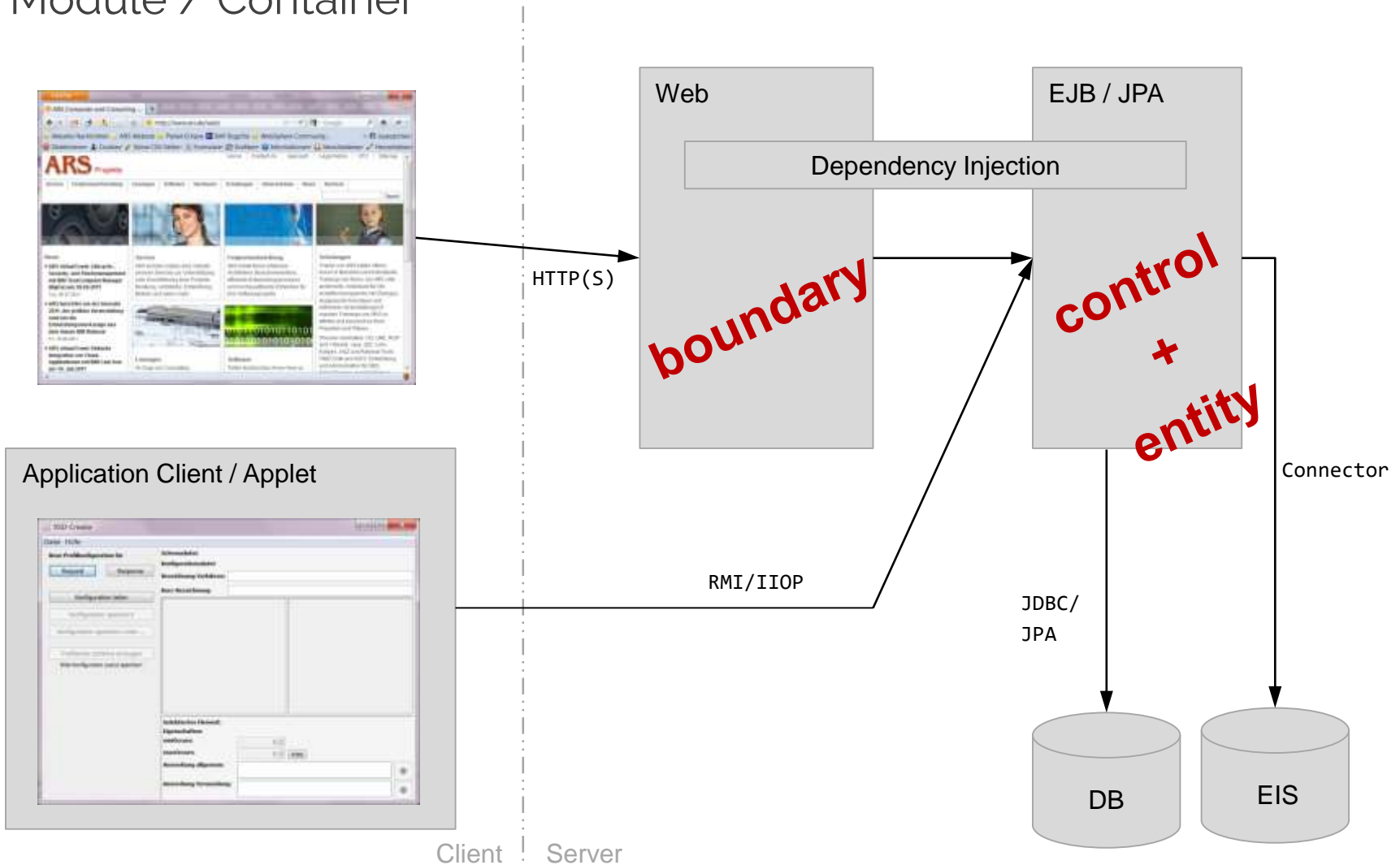
```
import javax.annotation.Resource;  
import javax.sql.DataSource;
```

```
...
```

```
@Resource(name="jdbc/LibraryDB")  
private DataSource ds;
```

# Architektur

## Module / Container



# Profile

- Java EE Spezifikation = Agregation mehrere (Teil-)Spezifikationen
- „Java EE compliant“
  - Umsetzung aller Spezifikationen
  - Laufzeitumgebung sehr umfangreich
- Definition von Profilen als Teilmenge
  - kleinere Laufzeitumgebung (Embedded Runtime möglich)
  - Beispiel: Web Profile

Overview Downloads Documentation Community Technologies Training

### Java™ EE 8 Technologies

Learn more about the technologies that comprise the Java EE 8 platform using the specifications, and then apply them with the Java EE 8 SDK.

Specification downloads are the final releases. Please check the individual JSR pages for download updates such as maintenance releases.

Java EE 8 Technologies			
Technology	JSR	Download	Web Profile
<b>Java EE Platform</b>			
Java Platform, Enterprise Edition 8 (Java EE 8)	JSR 308	Download spec	
<b>Web Application Technologies</b>			
Java API for WebSocket 1.1	JSR 356	Download spec	✓
Java API for JSON Binding 1.0	JSR 367	Download spec	✓
Java API for JSON Processing 1.1	JSR 314	Download spec	✓
Java Servlet 4.0	JSR 369	Download spec	✓
JavaServer Faces 2.3	JSR 312	Download spec	✓
Expression Language 3.0	JSR 341	Download spec	✓
JavaServer Pages 2.3	JSR 345	Download spec	✓
Standard Tag Library for JavaServer Pages (JSTL) 1.2	JSR 52	Download spec	✓
<b>Enterprise Application Technologies</b>			
Batch Applications for the Java Platform 1.0	JSR 352	Download spec	
Concurrency Utilities for Java EE 1.0	JSR 236	Download spec	
Contexts and Dependency Injection for Java 2.0	JSR 269	Download spec	✓
Dependency Injection for Java 1.0	JSR 220	Download spec	✓
Bean Validation 2.0	JSR 303	Download spec	✓
Enterprise JavaBeans 3.2	JSR 348	Download spec	✓
Interceptors 1.2	JSR 216	Download spec	✓
Java EE Connector Architecture 1.7	JSR 222	Download spec	
Java Persistence 2.2	JSR 223	Download spec	✓
Common Annotations for the Java Platform 1.3	JSR 250	Download spec	✓
Java Message Service API 2.0	JSR 343	Download spec	✓
Java Transaction API (JTA) 1.2	JSR 907	Download spec	✓
JavaMail 1.6	JSR 919	Download spec	
<b>Web Services Technologies</b>			
Java API for RESTful Web Services (JAX-RS) 2.1	JSR 310	Download spec	✓
Implementing Enterprise Web Services 1.3	JSR 109	Download spec	
Web Services Metadata for the Java Platform 2.1	JSR 181	Download spec	
Java API for XML-based RPC (JAX-RPC) 1.1 (Optional)	JSR 101	Download spec	
Java API for XML Negotiation (JAX-N) 1.0 (Optional)	JSR 92	Download spec	
<b>Management and Security Technologies</b>			
Java EE Security API 1.0	JSR 375	Download spec	✓
Java Authentication Service Provider Interface for Containers 1.1	JSR 198	Download spec	✓
Java Authentication Contract for Containers 1.0	JSR 115	Download spec	
Java EE Application Deployment 1.2 (Optional)	JSR 88	Download spec	
J2EE Management 1.1	JSR 77	Download spec	
Debugging Support for Other Languages 1.0	JSR 43	Download spec	✓
<b>Java EE-related Specs in Java EE</b>			
Java Management Extensions (JMX) 2.0	JSR 3	Download spec	
SOAP with Attachments API for Java (SAAJ) Specification 1.3	JSR 67	Download spec	
Streaming API for XML (SAX) 1.0	JSR 173	Download spec	
Java API for XML Processing (JAXP) 1.6	JSR 226	Download spec	
Java Database Connectivity 4.0	JSR 221	Download spec	
Java Architecture for XML Binding (JAXB) 2.2	JSR 222	Download spec	
Java API for XML-based Web Services (JAX-WS) 2.2	JSR 224	Download spec	
JavaBeans Activation Framework (JAF) 1.1	JSR 225	Download spec	

<https://www.oracle.com/technetwork/java/javaee/tech/index.html>

## Kontrollfragen

- Was ist Java EE?
- Was ist ein Application Server?
- Wie werden Java-EE-Anwendungen konfiguriert?
- Welchen Vorteil bietet Dependency Injection?

