



Business
Technology Days

Jens Seekamp, Dirk Weil
GEDOPLAN GmbH
W-JAX 2017

Java EE 7/8 Enterprise-Anwendungen ohne Ballast

Jens Seekamp & Dirk Weil

- ≡ GEDOPLAN GmbH, Bielefeld
 - ≡ GEDOPLAN IT Consulting
Konzeption, Realisierung von IT-Lösungen
 - ≡ GEDOPLAN IT Training
Seminare in Berlin, Bielefeld, on-site
- ≡ Java EE seit 1998
- ≡ Vorträge, Veröffentlichungen



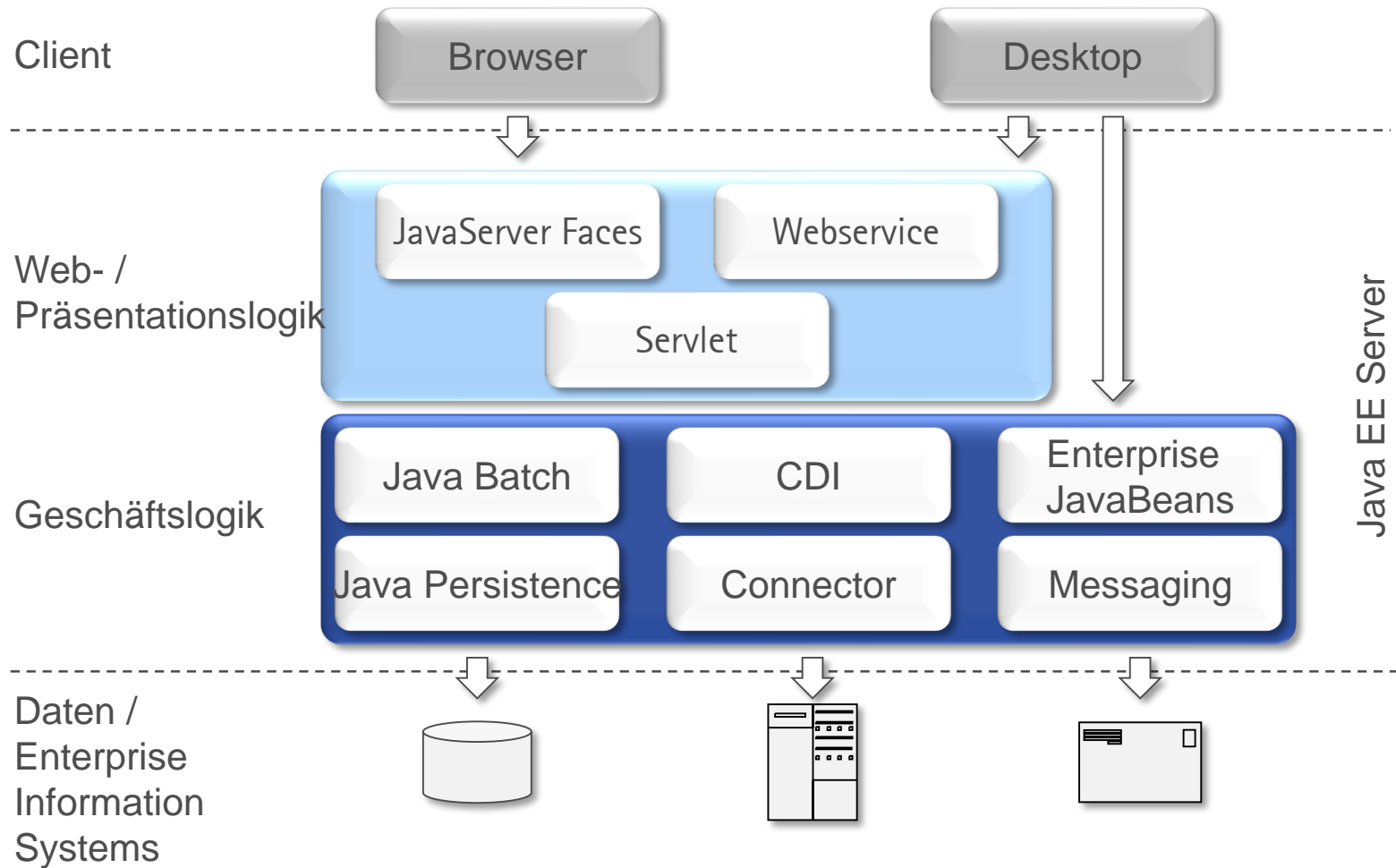
Was ist Java EE?

- ≡ = Java Platform, Enterprise Edition
- ≡ Plattform für mehrschichtige Anwendungen
- ≡ Spezifikationen und Interfaces als SE-Ergänzung
- ≡ Implementierung durch Produkte
 - ≡ GlassFish / Payara – Oracle WebLogic
 - ≡ JBoss / WildFly – IBM WebSphere
 - ≡ Apache TomEE – ...

Was ist Java EE?

- ≡ Vereinfacht die Entwicklung von Enterprise-Anwendungen durch
 - ≡ ein standardisiertes, modulares Komponenten-Modell
 - ≡ einen kompletten Satz von Infrastrukturdiensten für diese Komponenten
 - ≡ automatische Abwicklung von Problemen ohne komplexe Programmierung

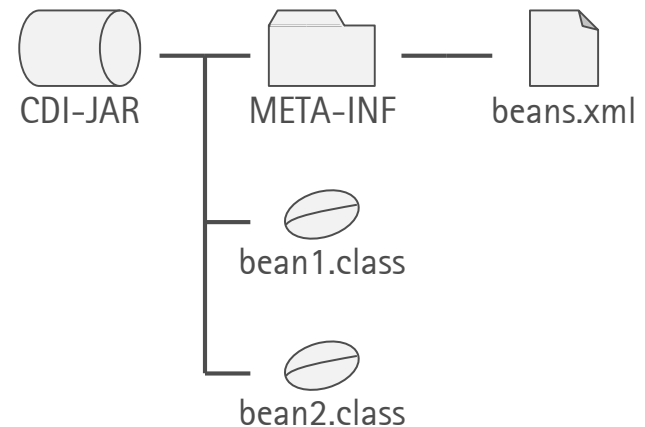
Architektur-Modell



Java-EE-Bestandteile

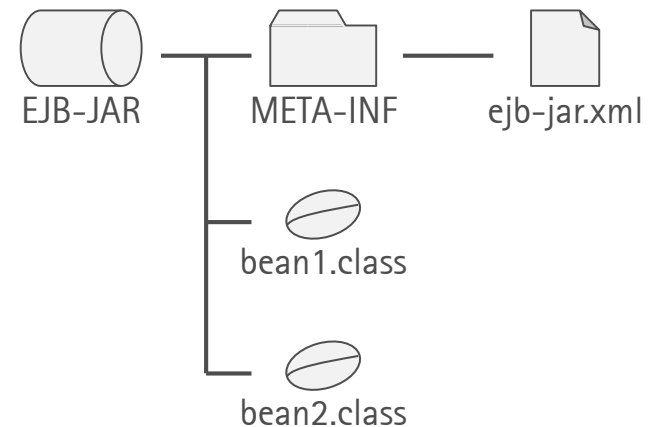
≡ CDI

- ≡ Beans, Interfaces, Interceptors etc.
- ≡ Deployment Descriptor `beans.xml`
- ≡ Paketierung als `.jar`-Datei
oder Integration in
EJB / Webanwendung



Java-EE-Bestandteile

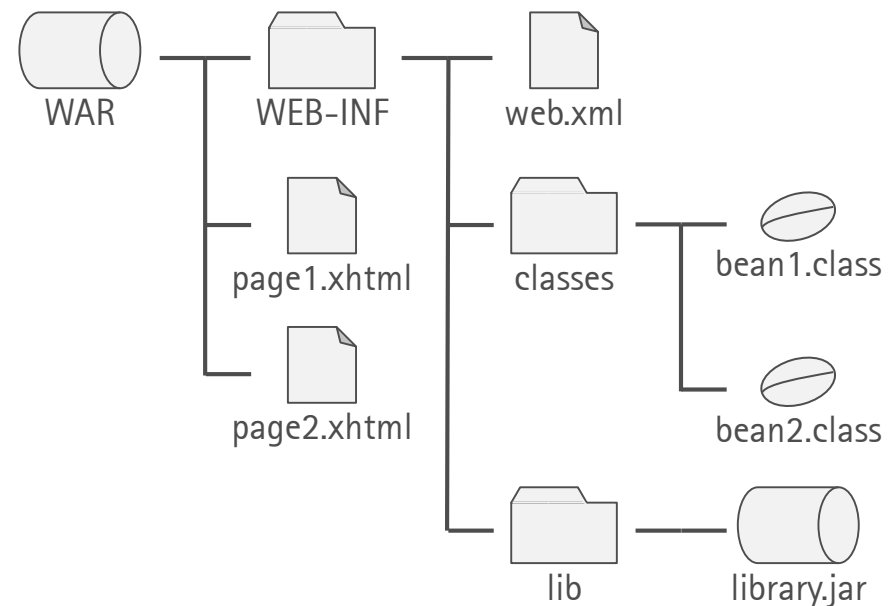
- Enterprise JavaBeans
 - Beans, Interfaces, Interceptors etc.
 - Deployment Descriptor `ejb-jar.xml` (optional)
 - Paketierung als `.jar`-Datei
oder Integration in
Webanwendung



Java-EE-Bestandteile

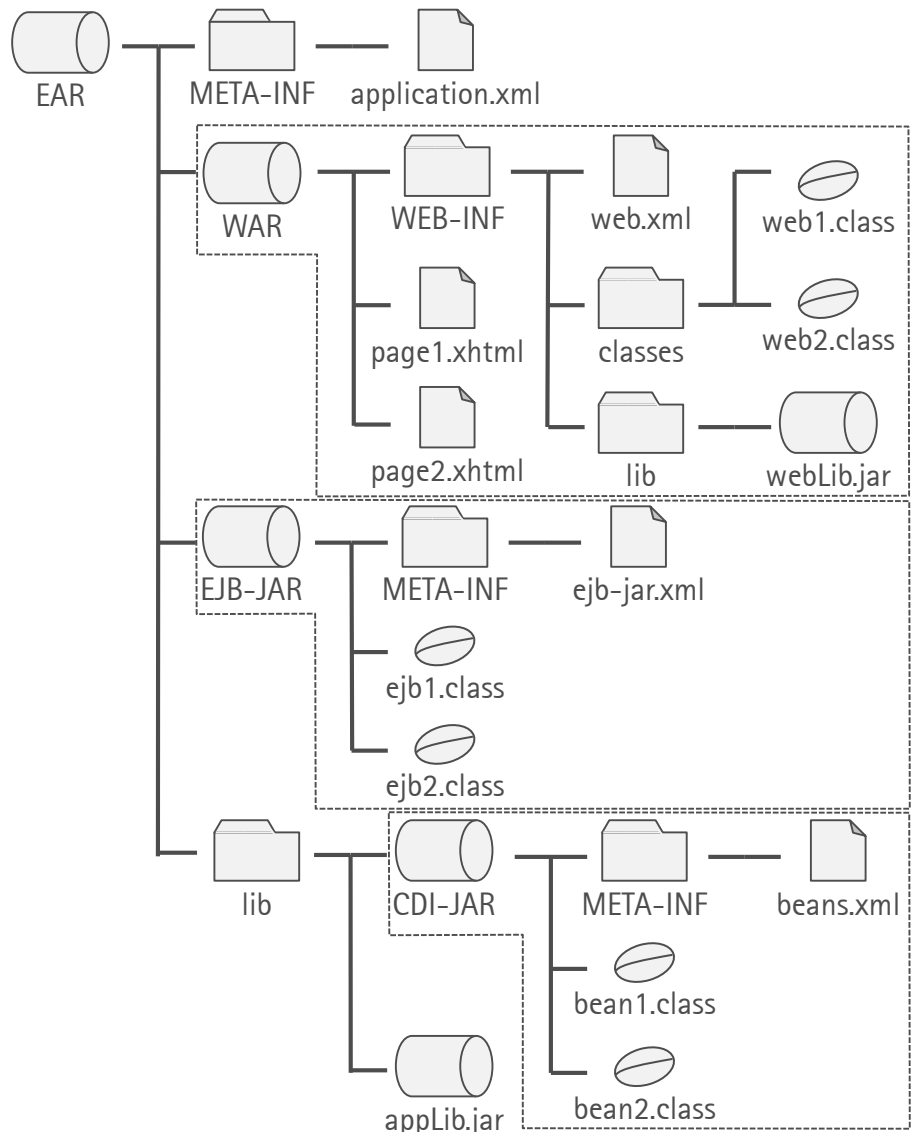
Webanwendungen

- Web-Dokumente, Servlets, JSP, JSF
- Deployment Descriptor `web.xml` (optional)
- Paketierung
als `.war`-Datei



Enterprise- Anwendungen

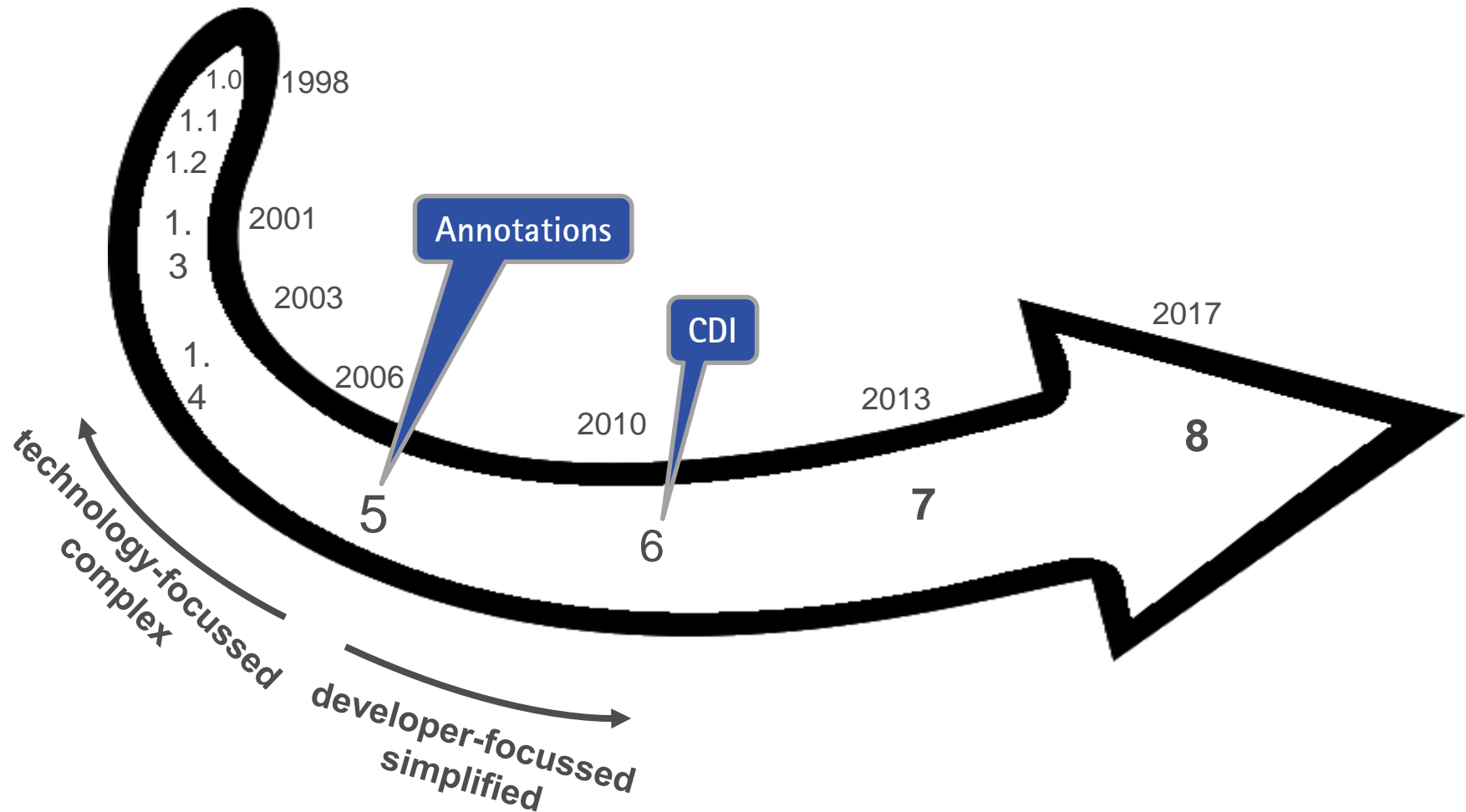
- Zusammenfassung der Bestandteile
- Deployment Descriptor `application.xml` (optional)
- Paketierung als `.ear-Datei`



Deployment

- ≡ Erlaubte Formate:
 - ≡ WAR
 - ≡ EAR
- ≡ Serverabhängige Deployment-Verfahren:
 - ≡ Auto Deployment
 - ≡ Administrationswerkzeuge

Historie



Technologien

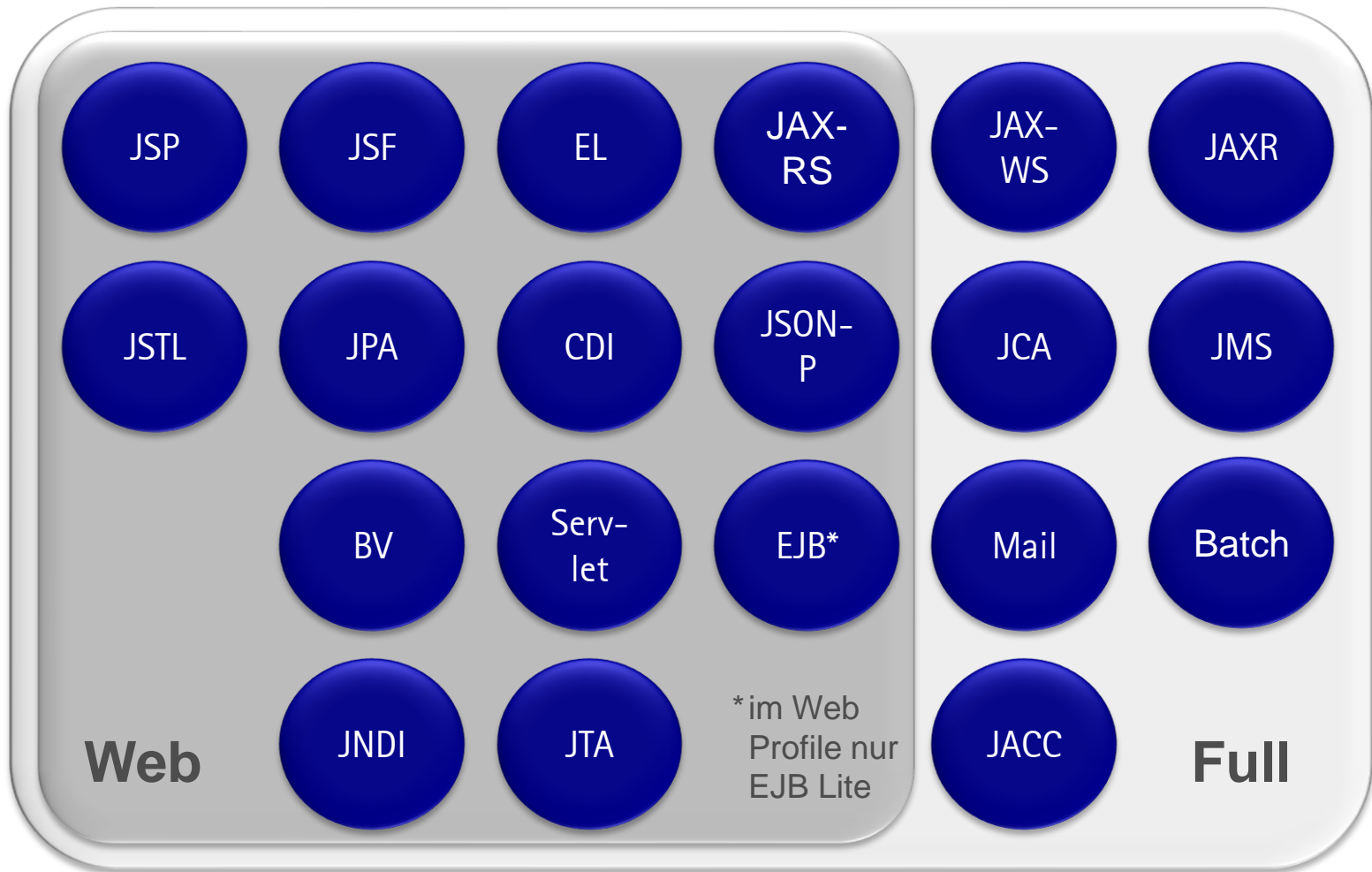
	Specification	Web Profile?	Version / JSR		
Java EE Platform	Java Platform, Enterprise Edition		8	7	6
	Java EE		366	342	316
Web Application Technologies	Java API for WebSocket	✓	1.1	1.0	
			356	356	
	Java API for JSON Binding	✓	1.0		
	JSON-B		367		
	Java API for JSON Processing	✓	1.1	1.0	
	JSON-P		374	353	
	Java Servlet	✓	4.0	3.1	3.0
	Servlet		369	340	315
	JavaServer Faces	✓	2.3	2.2	2.0
	JSF		372	344	314
	Expression Language	✓	3.0	2.2	
	EL		341		245
	JavaServer Pages	✓	2.3	2.2	
	JSP		245		245
Enterprise Application Technologies	Standard Tag Library for JavaServer Pages	✓	1.2		
	JSTL		52		
	Batch Applications for the Java Platform		1.0		
	Batch		352		
	Concurrency Utilities for Java EE		1.0		
			236		
	Contexts and Dependency Injection for Java	✓	2.0	1.1	1.0
	CDI		365	346	299
	Dependency Injection for Java	✓	1.0		
			330		
	Bean Validation	✓	2.0	1.1	1.0
			380	349	303
	Enterprise JavaBeans	✓	3.2	3.1	
	EJB		345		318
	Interceptors	✓	1.2	1.1	
			318		318
	Java EE Connector Architecture		1.7		1.6
	JCA		322		322
	Java Persistence	✓	2.2	2.1	2.0
	JPA		338	338	317
	Common Annotations for the Java Platform	✓	1.3	1.2	1.1
			250	250	250
	Java Message Service API		2.0		1.1
	JMS		343		914
	Java Transaction API	✓	1.2		1.1
	JTA		907		907
	JavaMail		1.6	1.5	1.4
			919	919	919

Versionsnummer in Klammern = Spezifikation in Java EE optional

	Specification	Web Profile?	Version / JSR		
Java EE Platform	Java Platform, Enterprise Edition		8	7	6
	Java EE		366	342	316
Web Services Technologies	Java API for RESTful Web Services	✓	2.1	2.0	1.1
	JAX-RS		370	339	311
	Implementing Enterprise Web Services		1.3		
			109		
	Web Services Metadata for the Java Platform		2.1		
			181		
	Java API for XML-Based Web Services		2.2		
	JAX-WS		224		
	Java API for XML-Based RPC		(1.1)		1.1
	JAX-RPC		101		101
	SOAP with Attachments API for Java (aka Java APIs for XML Messaging)		1.3		
			67		
Management and Security Technologies	Java API for XML Registries		(1.0)	1.0	
	JAXR		93	93	
	Java EE Security API	✓	1.0		
			375		
	Java Authentication Service Provider Interface for Containers		1.1		1.0
			196		196
	Java Authorization Contract for Containers		1.0		
	JACC		115		
	Java EE Application Deployment		(1.2)		1.2
			88		88
Java EE-related Specs in Java SE	J2EE Management		1.1		
			77		
	Debugging Support for Other Languages	✓	1.0		
			45		
	Java Management Extensions	✓	2.0		
	JMX		3		
	Streaming API for XML	✓	1.0		
	StAX		173		
	Java Architecture for XML Binding	✓	2.2		
	JAXB		222		
	Java API for XML Processing	✓	1.6	1.3	
	JAXP		206	206	
Java EE-related Specs in Java SE	Java Database Connectivity	✓	4.0		
	JDBC		221		
	JavaBeans Activation Framework	✓	1.1		
	JAF		925		



Profile



Beispielanwendung

- ≡ Conference Management
- ≡ `https://github.com/GEDOPLAN/javaee-workshop`

Beispielanwendung

≡ Java Persistence

- ≡ Entity Mapping
- ≡ Entity Manager
- ≡ Relations
- ≡ ...

≡ CDI

- ≡ Beans, Injektion
- ≡ Producer
- ≡ TX Interceptor
- ≡ Scopes
- ≡ ...

≡ REST

- ≡ Ressourcen
- ≡ JSON

≡ (AngularJS)

≡ JavaServer Faces

- ≡ MVC / Front Controller
- ≡ Facelets
- ≡ EL
- ≡ ...



Beispielanwendung

- ≡ Weitere Themen je nach Zeit Wunsch
 - ≡ JPA: Attribute Converter, ...
 - ≡ CDI: Fachliche Injektion, ...
 - ≡ Bean Validation
 - ≡ EJB
 - ≡ ?

Beispielanwendung

≡ Entwicklungsumgebung:

- ≡ Java 8
- ≡ Git Client (z. B. Git Bash git-scm.com/downloads)
- ≡ Maven 3.x (maven.apache.org/download.html)
- ≡ IDE, z. B. JBoss Developer Studio
(developers.redhat.com/products/devstudio/download/)
- ≡ DB-Tool, z. B. SQL Workbench/J
(www.sql-workbench.net/)

```
git clone https://github.com/GEDOPLAN/javaee-workshop.git
git checkout wjax2017
cd javaee-workshop/backend
mvn

# Import Maven project into IDE
```

Beispielanwendung

≡ Laufzeitumgebung:

≡ WildFly 10.1.0.Final

(<http://www.wildfly.org/downloads>)

```
# Server an beliebigen Platz entpacken ...

# Server starten
wildfly-10.1.0.Final/bin/standalone.sh bzw. .bat

# Datasource einrichten
cp javaee-workshop/backend/additional/wildfly/conference-ds.xml \
  wildfly-10.1.0.Final/standalone/deployments

# Anwendung deployen
cp javaee-workshop/backend/target/javaee-workshop.war \
  wildfly-10.1.0.Final/standalone/deployments
```

Beispielanwendung

≡ DB-Tool, z. B. SQL Workbench

Treiber-Jar	<code>h2-1.3.173.jar</code> (z.B. aus Projektverzeichnis <code>additional/db</code>)
URL	<code>jdbc:h2:~/h2/conference;AUTO_SERVER=TRUE</code> (am besten aus Projektdatei <code>additional/wildfly/conference-ds.xml</code> kopieren)
User	<code>conference</code>
Password	<code>conference</code>

Beispielanwendung

≡ Project Lombok

- ≡ <https://projectlombok.org/>
- ≡ Generierung von Gettern, Settern etc.
- ≡ Polarisiert stark („genial“ ⇔ „Teufelszeug“)
- ≡ Integration in Maven unproblematisch (Dependency)
- ≡ Integration in JBDS (Eclipse)
 - ≡ `lombok-1.16.12.jar` (aus Projektverzeichnis `additional/ide`) in IDE-Verzeichnis kopieren (wo auch `devstudio.ini` bzw. `eclipse.ini` liegt)
 - ≡ In `devstudio.ini` bzw. `eclipse.ini` im Abschnitt `-vmargs` Lombok-Agenten ergänzen

```
-vmargs
-javaagent:lombok-1.16.12.jar
-Xms512m
-Xmx1024m
...
```

Beispielanwendung

≡ „nur gucken“

- ≡ GitHub-Repo clonen
- ≡ Änderungen dann und wann pullen

≡ „selber basteln“

- ≡ GitHub-Repo clonen
- ≡ Arbeitskopie davon erstellen und darin arbeiten
- ≡ Änderungen dann und wann pullen
- ≡ Gewünschte Teile davon in Arbeitskopie übernehmen

1) Demo: Hello World, Talk

- ≡ Hello-World-Webservice erstellen
- ≡ Domänenklasse **Talk** entwickeln
- ≡ Webservice für **Talk** entwickeln
 - ≡ zunächst **GET** mit JSON-Repräsentation
- ≡ Repository für **Talk** zunächst als Mock entwickeln
- ≡ Webservice komplettieren
 - ≡ „not found“
 - ≡ **GET** für alle Talks
 - ≡ **PUT**
 - ≡ **DELETE**
 - ≡ **POST**
 - ≡ „bad request“
- ≡ Nutzung der Angular-UI für Talks („**talk**“)

```
{  
  "id": 1,  
  "title": "Java EE ohne Ballast",  
  "type": "WORKSHOP",  
  "start": 1509955200000,  
  "minutes": 480  
}
```



2) Übung: Person

- ≡ Fachklasse **Person** entwickeln

```
id:           Integer
lastName:     String
firstName:    String
```

- ≡ Repository für **Person** entwickeln (Interface + Mock)

 - ≡ Copy-and-Paste von **Talk...** ist erlaubt ;-)

- ≡ Webservice für **Person** entwickeln

 - ≡ `http://localhost:8080/javaee-workshop-backend/rest/person ...`

- ≡ UI „**speaker**“ sollte damit funktionieren

```
{
  "id": 1,
  "lastName": "Weil",
  "firstName": "Dirk",
}
```

3) Demo: DB-Anbindung für Talk

- ≡ **Talk** als JPA-Entity annotieren
- ≡ JPA-Implementierung des Repositories entwickeln
- ≡ Persistence Unit definieren
- ≡ Zwischen JPA- und Mock-Version per **@Alternative** umschalten

4) Übung: DB-Anbindung für Person

- ≡ `Talk` als JPA-Entity annotieren
- ≡ JPA-Implementierung des Repositories entwickeln
- ≡ Mock-Version zur `@Alternative` machen

5) Demo: Producer, equals/hashCode

- ≡ Producer für `EntityManager` einführen
- ≡ Entity-Klassen mit `equals/hashCode` versehen
- ≡ ggf.: Gemeinsame Aspekte in Basisklassen verschieben

6) Demo: Relation Talk ⇔ Person

- ≡ **Talk** um Speaker-Zuordnung ergänzen
 - ≡ zunächst **EAGER**
- ≡ UI „**talks-speaker**“ sollte nun funktionieren

- ≡ Speaker-Zuordnung **LAZY** machen
 - ≡ erzeugt Fehler bei **GET** von **Talk**
- ≡ Extended **EntityManager** Producer einsetzen
- ≡ UI „**talks-speaker**“ sollte nun wieder funktionieren

7) Demo / Übung (falls ausreichend Zeit)

- ≡ JSF-View für **Talk**

- ≡ Managed Bean / Presenter
- ≡ Facelet
- ≡ Expression Language

- ≡ Analog: Für **Person**

What's new in Java EE 8 (Auszug)

- Java API for JSON Binding (JSON-B) 1.0
 - Binding Java ↔ JSON analog zu JAXB
 - Referenzimplementierung Eclipse Yasson noch sehr buggy
- JavaServer Faces (JSF) 2.3
 - Deprecate `javax.faces.bean`
 - Java Time Support
 - Multi-field Validation
 - CDI Inject support in converters
 - Websocket support

What's new in Java EE 8 (Auszug)

- Contexts and Dependency Injection for Java (CDI) 2.0
 - API to boot CDI in Java SE
 - Observer ordering
 - Asynchronous events
- Bean Validation 2.0
 - Java 8 alignment
 - `@Email`, `@NotEmpty`, `@NotBlank`, `@Positive`,
`@PositiveOrZero`, `@Negative`, `@NegativeOrZero`,
`@PastOrPresent`, `@FutureOrPresent`

What's new in Java EE 8 (Auszug)

- Java Persistence (JPA) 2.2
 - Java 8 alignment
 - `getResultStream()`
 - CDI Inject support in attribute converters
- Java API for RESTful Web Services (JAX-RS) 2.1
 - Non-blocking I/O
 - Reactive support
 - Server side event support
- Java EE Security API 1.0
 - Standardize proprietary solutions (identity store, security context, ...)



More

- www.gedoplan-it-training.de
Seminare in Berlin, Bielefeld, Köln, Inhouse
- www.gedoplan-it-consulting.de
Reviews, Coaching, Development

- javaeeblog.wordpress.com/
- gedoplan-it-consulting.de/expertenkreis-java/

-  dirk.weil@gedoplan.de

-  [@dirkweil](https://twitter.com/dirkweil)

