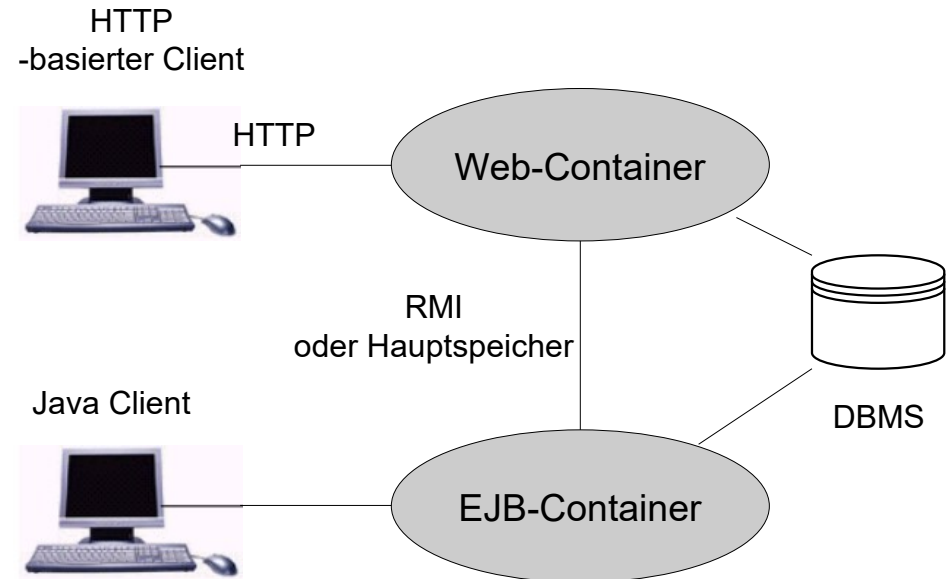


Java – JEE im Überblick

Stephan Karrer

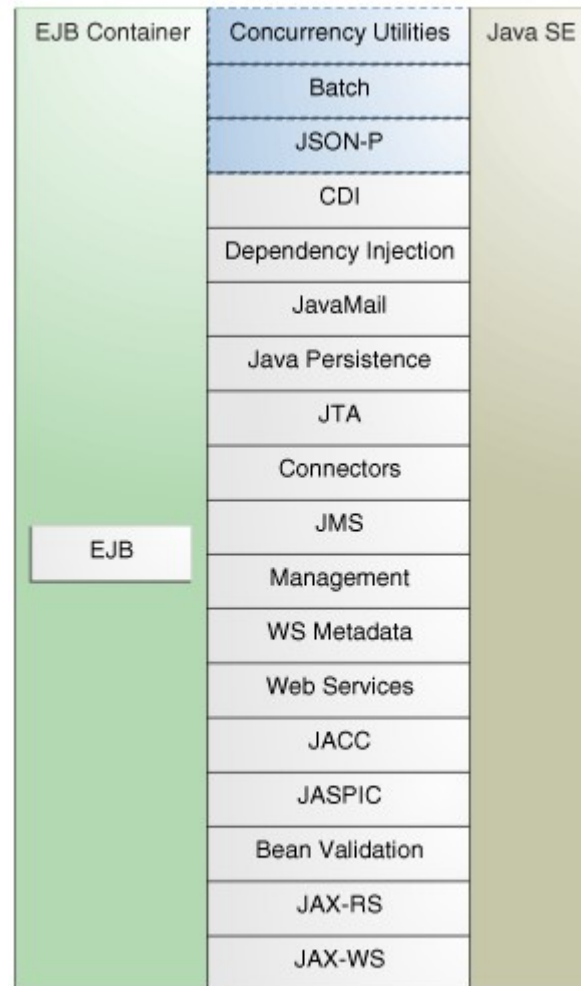
Java auf der Serverseite

- Web- oder auch Servlet-Container
 - HTTP-Server ist dabei
 - Aufgebohrte VM mit Netzwerk-Handling und Thread-Pool
 - Basis aller web-basierten Zugriffe
- EJB-Container
 - Aufgebohrte VM mit Netzwerk-Handling und Thread-Pool
 - Bietet Proxy-Mechanismus und Dependency-Injection
 - Verschiedene Lifecycle-Modelle für die EJBs (Enterprise Java Beans)
 - „Schwergewichtig“
 - Tendenz: Mit dem Web-Container auskommen

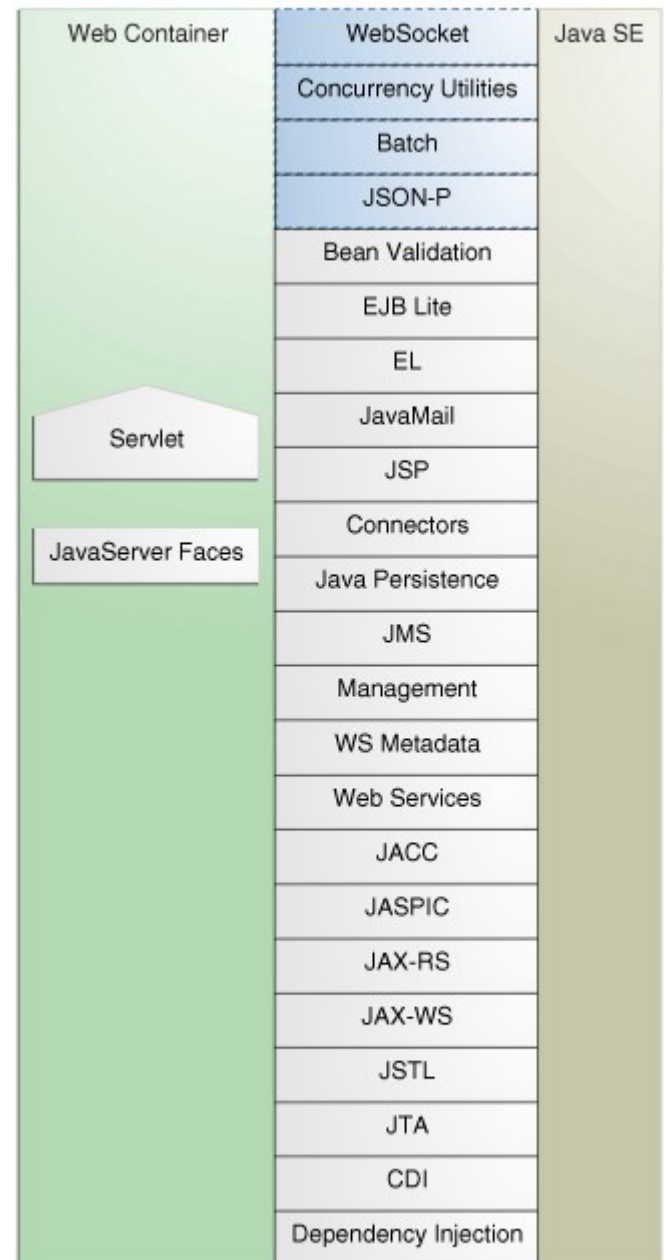



- Wichtige Server-Produkte:
 - Web-Container: Apache Tomcat, Apache Jetty
 - Full Server: RedHat Wildfly bzw. EAP, Oracle Weblogic, IBM Websphere, ...

Bestandteile des JEE 7-Standards

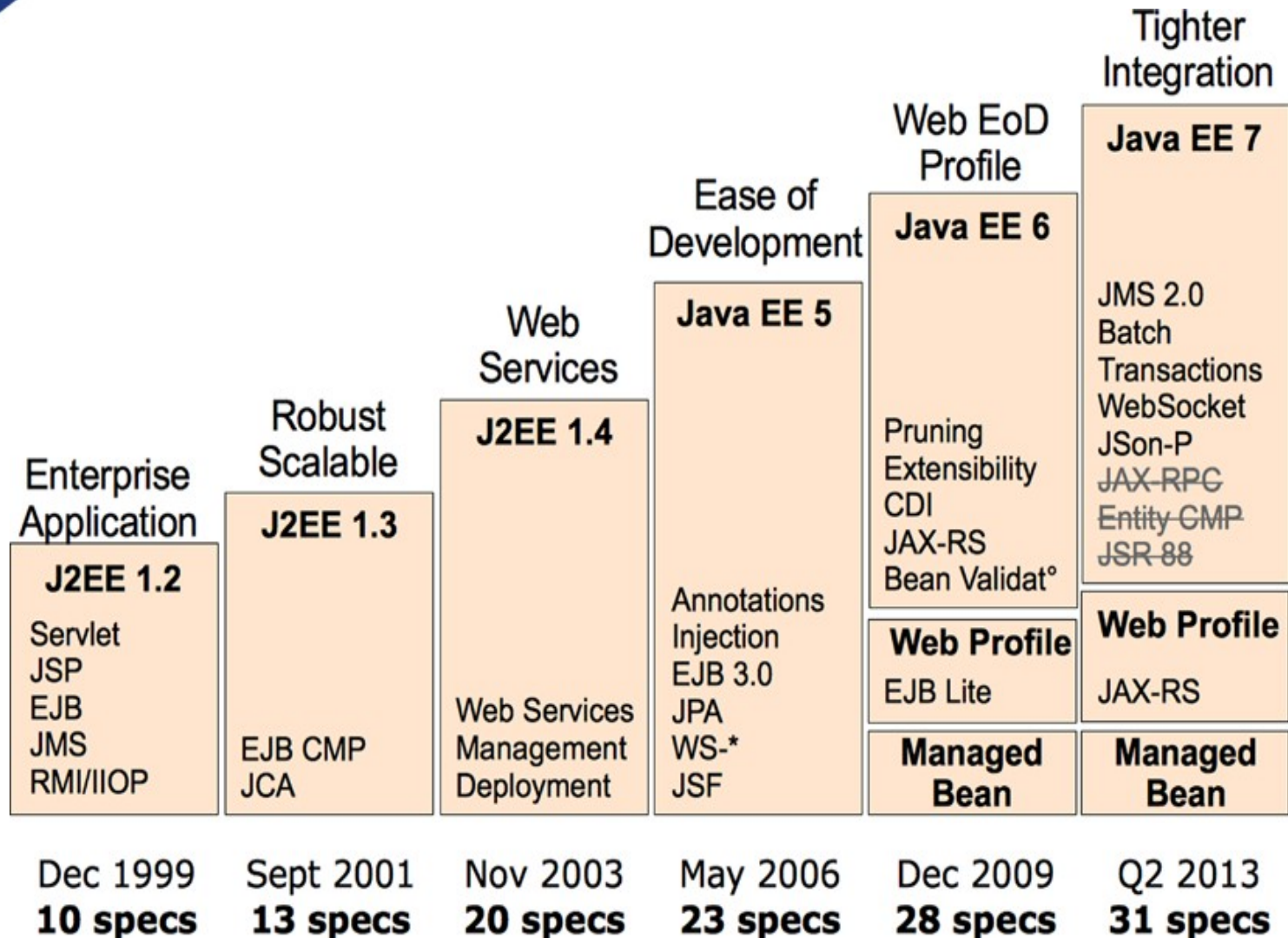


 New in Java EE 7



 New in Java EE 7

Einzelspezifikationen



Bibliotheken und Container auf der Serverseite

■ Vorteile

- Zwei Container, einer hält die Applikationslogik in Form von EJBs, der andere stellt den web-basierten Zugriff auf die Service-Bausteine zur Verfügung
- Alle möglicherweise benötigten Bibliotheken stehen bereits in miteinander kompatiblen Versionen zur Verfügung

■ Nachteile

- Der EJB-Container wird eventuell nicht benötigt
- In der Regel benötigt man nur einen Bruchteil der Bibliotheken

■ Tendenz

- Übertragung der programmiertechnischen Funktion der EJBs in den Web-Container (EJB-Light bzw. CDI-Framework). Somit genügt der Web-Container.
- Zerlegen des monolithischen Servers in einzelne Bestandteile, so dass wir mit dem Web-Container und lediglich den benötigten Bibliotheken für das Projekt ausgestattet sind. (Mikro-Profil-Ansatz)

Neu in JEE 7 (2013), Java 7 ist Basis

- JSON Processing (JSON-P, JSR 353)
- WebSockets (JSR 356)
- Batch Applications for Java (JBatch, JSR 352)
- Concurrency Utilities (JSR 236)

Updates in JEE 7

- JAX-RS (Client-API, Asynchrone Verarbeitung)
- Servlet 3.1 (Asynchrone Verarbeitung)
- JSF-Erweiterungen (JSF 2.2)
- CDI-Erweiterungen (CDI 1.1)
- JPA-Erweiterungen (JPA 2.1)
- Ansonsten kleinere Updates

Neu in JEE 8 (2017), Java 8 ist Basis

- JSON Binding API (JSON-B, JSR 367)
- Java EE Security API (JSR 375)

Updates in JEE 8

- Servlet 4.0 (HTTP/2, PushBuilder)
- Bean Validation 2.0
- CDI-Erweiterungen (CDI 2.0)
- Ansonsten kleinere Updates

Jakarta EE 8 (2019)

- Full Compatibility with Java EE 8: d.h. keine Neuerungen!

Jakarta EE 9 (2020)

- „The goal of the Jakarta EE 9 release is to deliver a set of specifications functionally similar to Jakarta EE 8 but in the new Jakarta EE 9 namespace jakarta.“
- Java SE 8 ist Basis
- Keine funktionalen Erweiterungen !

Jakarta EE 9.1 (2021)

- Java SE 11 ist Basis
- Keine funktionalen Erweiterungen !

Jakarta EE 10 (2022)

JAKARTA EE 10 PLATFORM

JAKARTA EE 10 WEB PROFILE

JAKARTA EE 10 CORE PROFILE

	Expression Language 5.0	Server Pages 3.1	
Authorization 2.1	Authentication 3.0	CDI 4.0	CDI Lite 4.0
Activation 2.1	Concurrency 3.0	WebSocket 2.1	JSON Binding 3.0
Batch 2.1	Persistence 3.1	Bean Validation 3.0	Annotations 2.1
Connectors 2.1	Faces 4.0	Debugging Support 2.0	Interceptors 2.1
Mail 2.1	Security 3.0	Enterprise Beans Lite 4.0	RESTful Web Services 3.1
Messaging 3.1	Servlet 6.0	Managed Beans 2.0	JSON Processing 2.1
Enterprise Beans 4.0	Standard Tag Libraries 3.0	Transactions 2.0	Dependency Injection 2.0

 Updated  Not Updated  New

Wichtige Neuerungen bzw. Updates

- Jakarta Contexts and Dependency Injection (CDI) 4.0, inkludiert CDI-Lite für Verwendung im MicroProfile (Core Profile)
- Jakarta Security 3.0 unterstützt OpenID Connect
- Jakarta Servlet 6.0 mit vereinfachter Programmierung und verbesserter Sicherheit
- Jakarta Faces (JSF) 4.0 mit aktualisierter API inklusive CDI-Nutzung
- Jakarta JSON Binding (JSON-B) 3.0 unterstützt polymorphe Typen
- Jakarta RESTful Web Services mit Java SE Bootstrap und Unterstützung für Multipart-Form-Daten
- Jakarta Persistence mit neuem Basistyp UUID und Erweiterungen der Query-Sprache
- Jakarta Concurrency 3.0 steht auch im Web-Profile zur Verfügung

Programmierung: Annotations Everywhere

```
@Stateful @SessionScoped @Model
public class Login {

    @Inject Credentials credentials;
    @Inject @Users EntityManager userDatabase;
    ...

    private User user;

    @Inject
    void initQuery(@Users EntityManagerFactory emf) {
        ...
    }

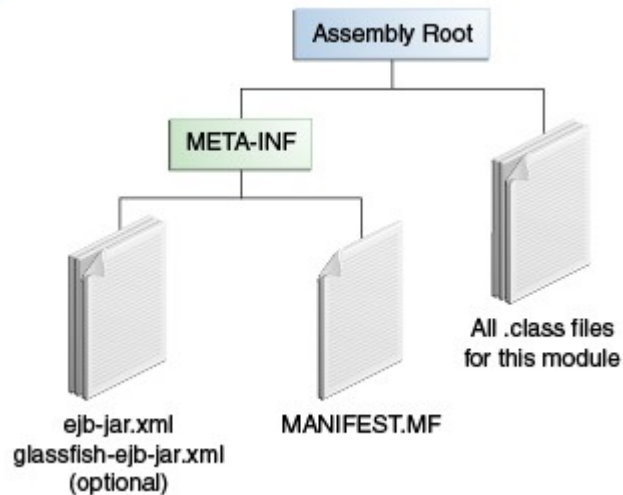
    @TransactionAttribute(REQUIRES_NEW)
    @RolesAllowed("guest")
    public void login() {
        ...
    }
    ...
}
```

XML-Konfigurationsdateien

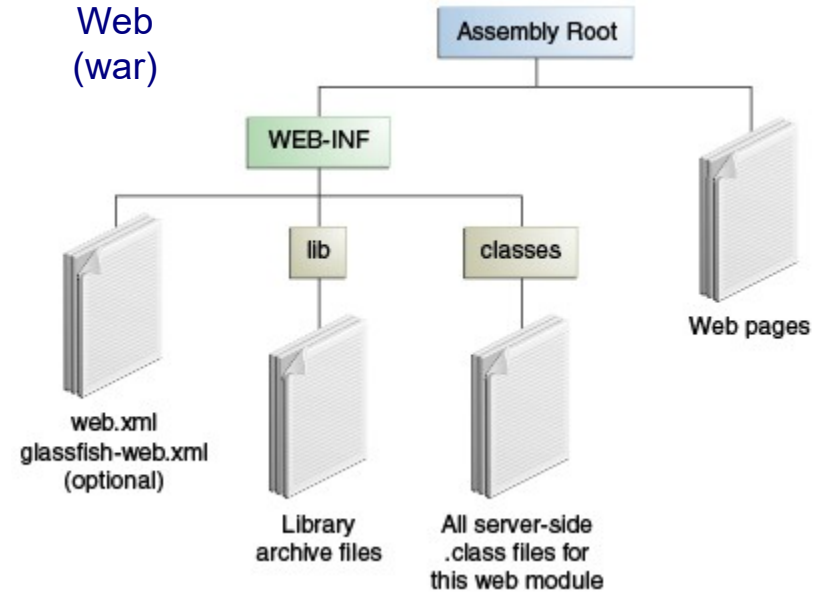
File	Specification	Paths
application.xml	Java EE	META-INF
application-client.xml	Java EE	META-INF
beans.xml	CDI	META-INF or WEB-INF
ra.xml	JCA	META-INF
ejb-jar.xml	EJB	META-INF or WEB-INF
faces-config.xml	JSF	WEB-INF
persistence.xml	JPA	META-INF
validation.xml	Bean Validation	META-INF or WEB-INF
web.xml	Servlet	WEB-INF
web-fragment.xml	Servlet	WEB-INF
webservices.xml	SOAP Web Services	META-INF or WEB-INF

Packaging

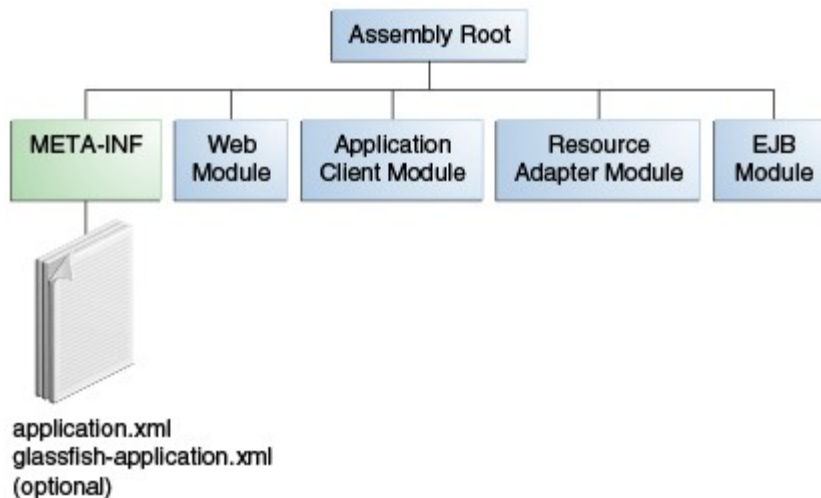
EJB
(ejb-jar)



Web
(war)



EAR



- EJBs können ab JEE 7 direkt mit der Web-Applikation in ein WAR gepackt werden