

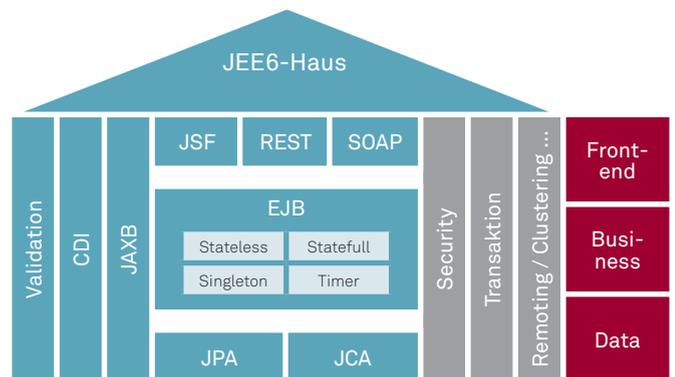
# JEE – Java Enterprise Edition

Die Java-Plattform für Enterprise-Business-Anwendungen

Die Java Enterprise Edition ist eine Plattform und Programmier-Modell für die Entwicklung von production-ready Enterprise-Business-Anwendungen, wie Web-GUI-, Web-Service-, Restful- und andere transaktionsbasierte, multi-tier, skalierbare, zuverlässige und sichere Anwendungen. Die Plattform basiert auf und nutzt die APIs und Dienste der Java Standard Edition und erweitert diese um relevante Spezifikationen und APIs.

## Definition

Aus Sicht des Programmier-Modells betrachten die Spezifikationen der JEE den kompletten Lebenszyklus einer Anwendung, von der Entwicklung, über das Auslieferungsformat für Komponenten einer Anwendung bis hin zum Deployment (Installation) in einem kompatiblen JEE-Container (aka Application Server). Der JEE-Container stellt die Laufzeitumgebung für die Komponenten einer Anwendung sowie direkt oder indirekt nutzbare Dienste zur Verfügung. Direkt nutzbare Dienste sind solche, die die Komponenten per API ansprechen können, wie zum Beispiel Transaktions-Management in Verbindung mit der Persistenz von Daten in die Datenbank.



Anwendung in einem Cluster von JEE-Containern auszuführen, um Anforderungen wie Ausfallsicherheit und Lastverteilung umsetzen zu können. Dazu zählt auch das Management von Ressourcen oder das Management des Lebenszyklus von Enterprise Java Beans (EJB). Der Begriff Komponente kann aus Sicht von Deployment-Artefakten definiert werden: Man kann reine Web-, EJB- oder Enterprise-Anwendungen – beide vorgenannten kombiniert – unterscheiden. Wichtigster Bestandteil einer EJB-Anwendung ist das Enterprise Java Bean – es ist das Plattform-Konstrukt, um transaktionsbasierte, sichere und skalierbare Business-Funktionalität zu entwickeln. EJBs sind Java-Klassen, die bestimmten Konventionen/Restriktionen genügen, so dass sie in einem JEE-Container ausführbar sind und mit diesem interagieren können.

<b>Dienste</b> <ul style="list-style-type: none"> <li>• Transaktions-Management</li> <li>• Clustering / Loadbalancing</li> <li>• Security (AAA)</li> <li>• Monitoring</li> <li>• Lebenszyklus von EJBs</li> <li>• Ressourcen-Management</li> <li>• Zeitsteuerung</li> <li>• Clustering</li> </ul>	<b>Container Anbieter</b> <ul style="list-style-type: none"> <li>• Oracle Weblogic / Glassfish</li> <li>• JBoss</li> <li>• IBM Websphere</li> <li>• Apache Geronimo</li> </ul>
<b>Java EE</b>	
<b>Begleitende Technologie-Trends</b> <ul style="list-style-type: none"> <li>• RESTful Webservices</li> <li>• Object-Relational-Mapping mit JPA</li> <li>• RIA / Mobile Anwendungen</li> </ul>	<b>Steigerung der Entwicklerproduktivität mit JEE6</b> <ul style="list-style-type: none"> <li>• Convention over Configuration</li> <li>• Dependency Injection per CDI</li> <li>• Vereinfachung – weniger Boilerplate Code</li> </ul>

Indirekt nutzbare Dienste sind zum Beispiel die aus Entwicklersicht transparente Möglichkeit, eine

## Reifegrad

Die Java Enterprise Edition ist bedingt durch ihre lange Historie eine Plattform mit sehr hohem

