

Java EE Basics

FWP Aktuelle Technologien zur Entwicklung
verteilter Java-Anwendungen

EINFÜHRUNG IN DIE JAVA EE-PLATTFORM

Brandaktuell: Java EE wird EE4J

- Oracle übergibt in Absprache mit IBM und Red Hat die Weiterentwicklung von Java EE an die Eclipse Foundation
 - ⦿ keine Abhängigkeit von einer einzelnen Partei mehr
 - ⦿ Entwicklungsprozess soll transparenter, flexibler und agiler werden
- Oracle unterstützt weiterhin aktuelle Java EE Lizenzen
- Oracle setzt weiterhin auf WebLogic als Application Server
- Neuer Projektname lautet **Eclipse Enterprise for Java (EE4J)**
- Homepage unter <https://projects.eclipse.org/projects/ee4j>

Die Java EE Plattform

- Java EE Spezifikation definiert
 - ◉ ein Programmiermodell für Applikationen
 - ◉ die Eigenschaften einer Laufzeitumgebung für Applikationen (Application Server)
- Hersteller liefern Application Server
- Fokus auf server-seitige Applikationen mit web-basierter Benutzeroberfläche
- Aktuelle Version: JavaEE 8 (September 2017)

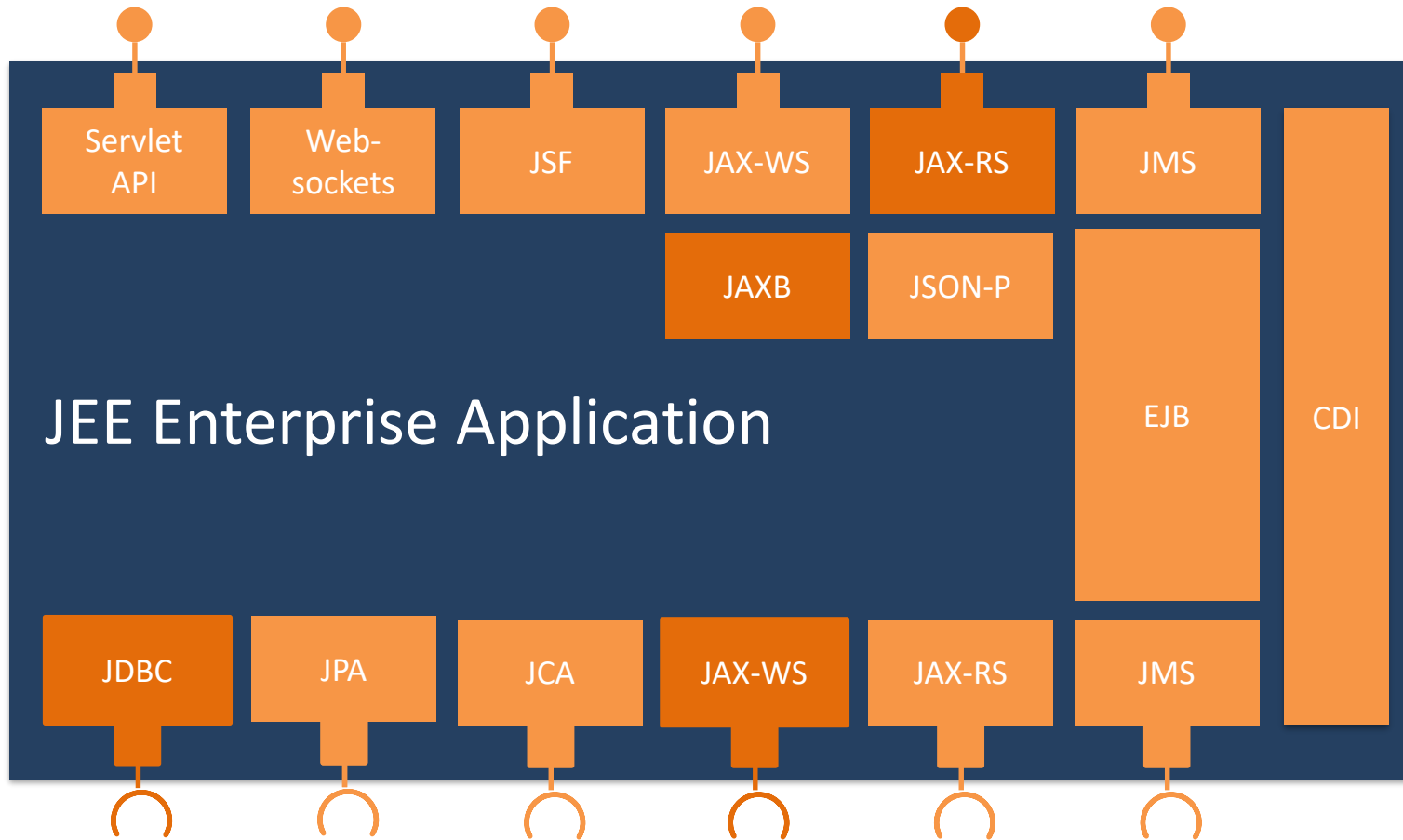
Ziele der Java EE Plattform

- Standardisierter Applikationscontainer mit essentiellen Infrastrukturdiensten
 - ⊙ Remoting, Transaktionen, Security, Persistenz
- Leichte und schnelle Entwicklung von portablen, serverseitigen Applikationen
 - ⊙ POJO-basiertes Modell, Annotations, Convention over Configuration, Dependency Injection, AOP
- Flexibler Technologiestack über Profile
- Ausgelegt auf Erweiterbarkeit

Profiles – Tailoring der Java EE Plattform

- Profile ermöglichen flexiblen Technologiestack
- Java EE kommt standardmäßig mit zwei Profilen
 - ⊙ **Full Profile** = komplette Laufzeitumgebung
 - ⊙ **Web Profile** = reduzierte Laufzeitumgebung („Tomcat mit EJBs“)
- Community (Eclipse Foundation) liefert drittes Profil
 - ⊙ **Micro Profile** = reduzierte Laufzeitumgebung für Microservices

Übersicht JEE Technologien



Servlet API: Support für (HTTP-)Endpunkte

Websockets: Support für Websockets

JSF: Java Server Faces; webbasierte Uis

JAX-WS: Java API for XML Web Services (SOAP)

JAXB: Java Architecture for XML Binding; schon aus Java SE

JAX-RS: Java API for RESTful Web Services (REST)

JSON-P: Java API for JSON Processing

JMS: Java Message Service; Integration von Messaging Systemen

EJB: Enterprise Java Beans; Support für serverseitige transaktionale Komponenten

CDI: Contexts and Dependency Injection

JDBC: Java Database Connectivity; Integration von relationalen Datenbanken aus Java SE

JPA: Java Persistence API; Standard für persistente Java-Objekte

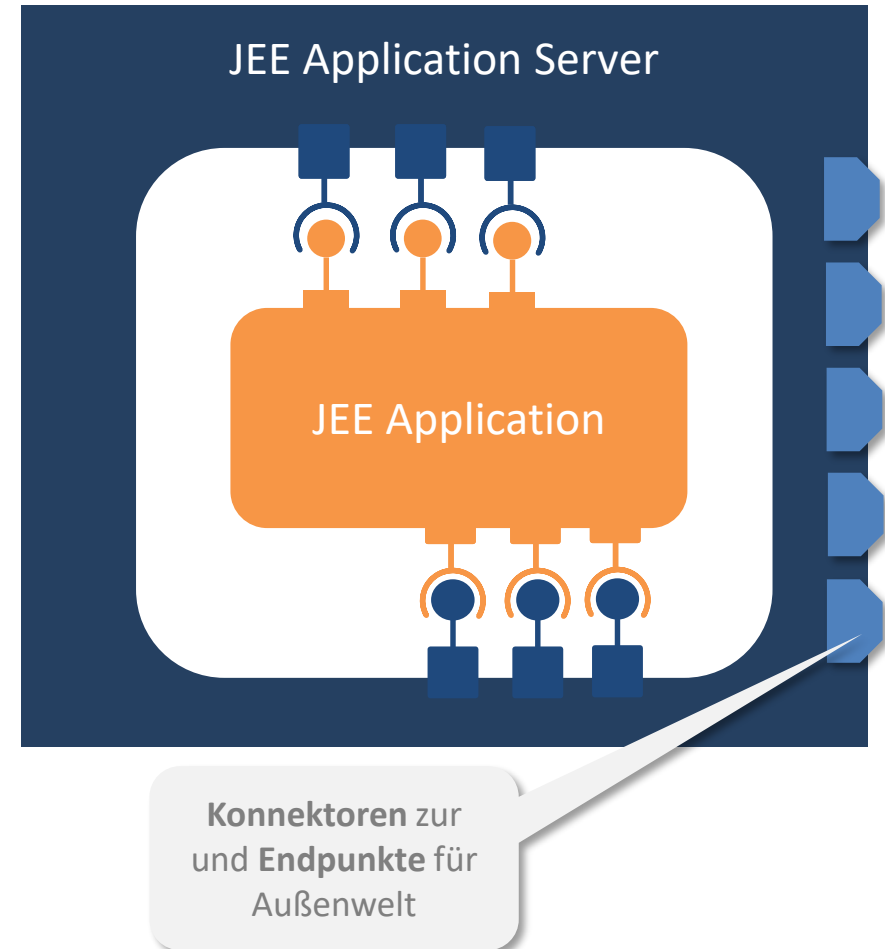
JCA: Java EE Connector Architecture; Integration von transaktionalen Enterprise Information Systemen

Rolle eines JEE Application Servers und dessen Zusammenspiel mit JEE Applikationen

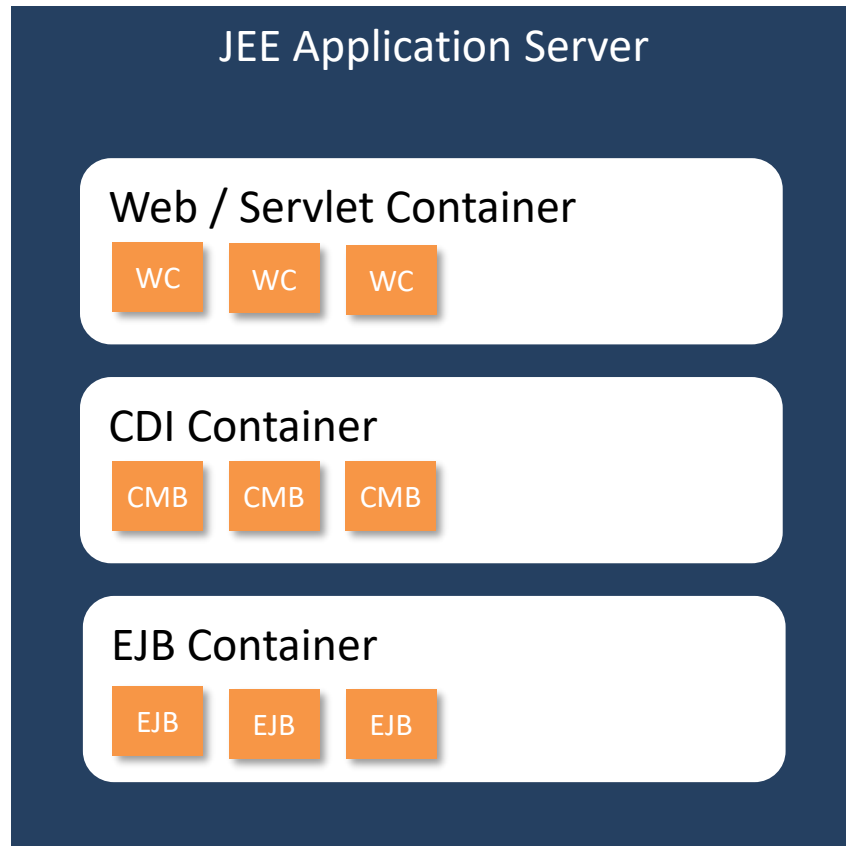
GRUNDLAGEN ZU JEE APPLICATION SERVERN

Container für JEE Applikationen

- JEE Application Server stellt Laufzeitumgebung für JEE Applikationen dar
- JEE Application Server muss vorgegebene Schnittstellen anbieten, damit JEE Applikationen darin laufen können
- JEE Applikation muss bestimmte Schnittstellen unterstützen, damit der JEE Application Server sie ausführen kann



Container aus Containern



- JEE Applikation bestehen aus Komponenten von bestimmten Typen
- Pro Komponententyp gibt es eigene Container im JEE Application Server:
 - Web-Komponenten (WC) laufen im Servlet Container
 - CDI-Container verwaltet alle CDI managed Beans (CMB)
 - EJB-Container kontrolliert alle Enterprise Java Beans (EJB)
- Container kennen sich untereinander

Bekannte JEE Application Server (Auszug)



Glassfish

- Open Source Referenz Implementierung
- Aktuelle Version: 5.0 (JEE 8)
- Webseite: <https://javaee.github.io/glassfish/>



Payara

- Glassfish Klon mit kommerziellem Support
- Aktuelle Version: 4.1 (JEE 7), 5 Beta (JEE 8)
- Webseite: <https://www.payara.fish/>

Oracle WebLogic Server

WebLogic

- Kommerzielles Server-Flagschiff von Oracle
- Aktuelle Version: 12.c (JEE 7)
- Webseite: <https://www.oracle.com/de/middleware/weblogic/index.html>

IBM WebSphere Application Server

WebSphere

- Kommerzielles Server-Flagschiff von IBM
- Aktuelle Version: 9 (JEE 7)
- Webseite: <https://www.ibm.com/de-de/marketplace/java-ee-runtime>



WildFly

- Open Source Community Server von Red Hat
- Aktuelle Version: 11 (JEE 7)
- Webseite: <http://wildfly.org/>



JBoss EAP

- Kommerzieller Server von Red Hat
- Aktuelle Version: 7.1 (JEE 7)
- Webseite: <https://www.redhat.com/de/technologies/jboss-middleware/application-platform>

Aufbau von JEE Applikationen

ANATOMIE EINER JAVA EE-APPLIKATION

Module einer Enterprise Applikation



- Eine typische Java EE Applikation wird durch ein **Enterprise Application Archive (EAR)** repräsentiert
 - ZIP-Datei mit standardisiertem Inhalt
 - In sich vollständige Installationseinheit
- Ein EAR besteht im allgemeinen
 - aus einer Webapplikation (WAR), die die Benutzeroberfläche repräsentiert
 - aus einem EJB JAR mit Enterprise Java Beans, die die Businesslogik der Applikation repräsentieren
 - aus einem EJB Client JAR, welches die Interfaces zu den EJBs zur Verfügung stellt
 - aus mehreren Utility JARs, die Querschnittsfunktionalität für alle Module zur Verfügung stellt (Frameworks, Security, Logging...) (/lib)

Physische Struktur eines EARs

`${appName}.ear`

`| ${webModName}.war`

`| ${ejbClientModName}.jar`

`| ${ejbModName}.jar`

`| /META-INF`

`| application.xml`

`| glassfish-application.xml`

`| /lib`

`| ${utilityModName}.jar`

`| ${utilityModName}.jar`

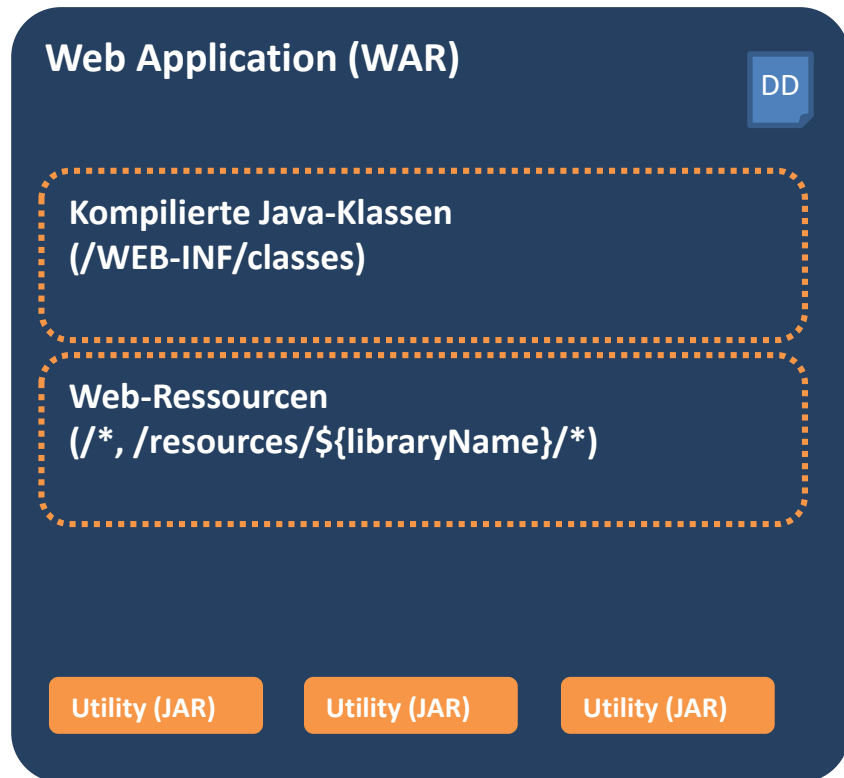
`| ...`

Applikationsmodule: Web Application Module (WAR), EJB Client Module (JAR), EJB Module (JAR)

Deployment Deskriptoren (optional):
application.xml = Standard-Deskriptor
glassfish-application.xml = Hersteller-spezifischer Deskriptor

Alle Utility-JARs, die keine Web Application Module, keine EJB (Client) Module sind

Module einer Web Applikation



- Eine typische Java EE Web-Applikation wird durch ein **Web Application Archive (WAR)** repräsentiert
 - ZIP-Datei mit standardisiertem Inhalt
 - In sich vollständige Installationseinheit
- Ein WAR besteht im allgemeinen
 - aus kompilierten Java-Klassen (/WEB-INF/classes)
 - aus Web-Ressourcen wie (X)HTML, Grafiken, CSS, JavaScript (/*)
 - aus JSF-Ressource-Bibliotheken (/resources/\${libraryName}/*)
 - aus mehreren Utility-JARs (/WEB-INF/lib)
- Enterprise Java Beans (EJB) werden auch unterstützt (EJB Lite = EJB ohne Remote Interfaces)

Physische Struktur eines WARs

`${webAppName}.war`

[static web resources]

Statische Web-Ressourcen (HTML, CSS, JS, Bilder) mit beliebiger Verzeichnisstruktur

/WEB-INF

web.xml

glassfish-web.xml

beans.xml

faces-config.xml

persistence.xml

Deployment Deskriptoren und Konfigurationsdateien (optional):

web.xml = Standard-Deskriptor

glassfish-web.xml = Hersteller-spezifischer Deskriptor

beans.xml = CDI-Konfigurationsdatei

faces-config.xml = JSF-Konfigurationsdatei

persistence.xml = JPA-Konfigurationsdatei

/classes

[compiled classes].class

/lib

`${utilityModName}.jar`

`${utilityModName}.jar`

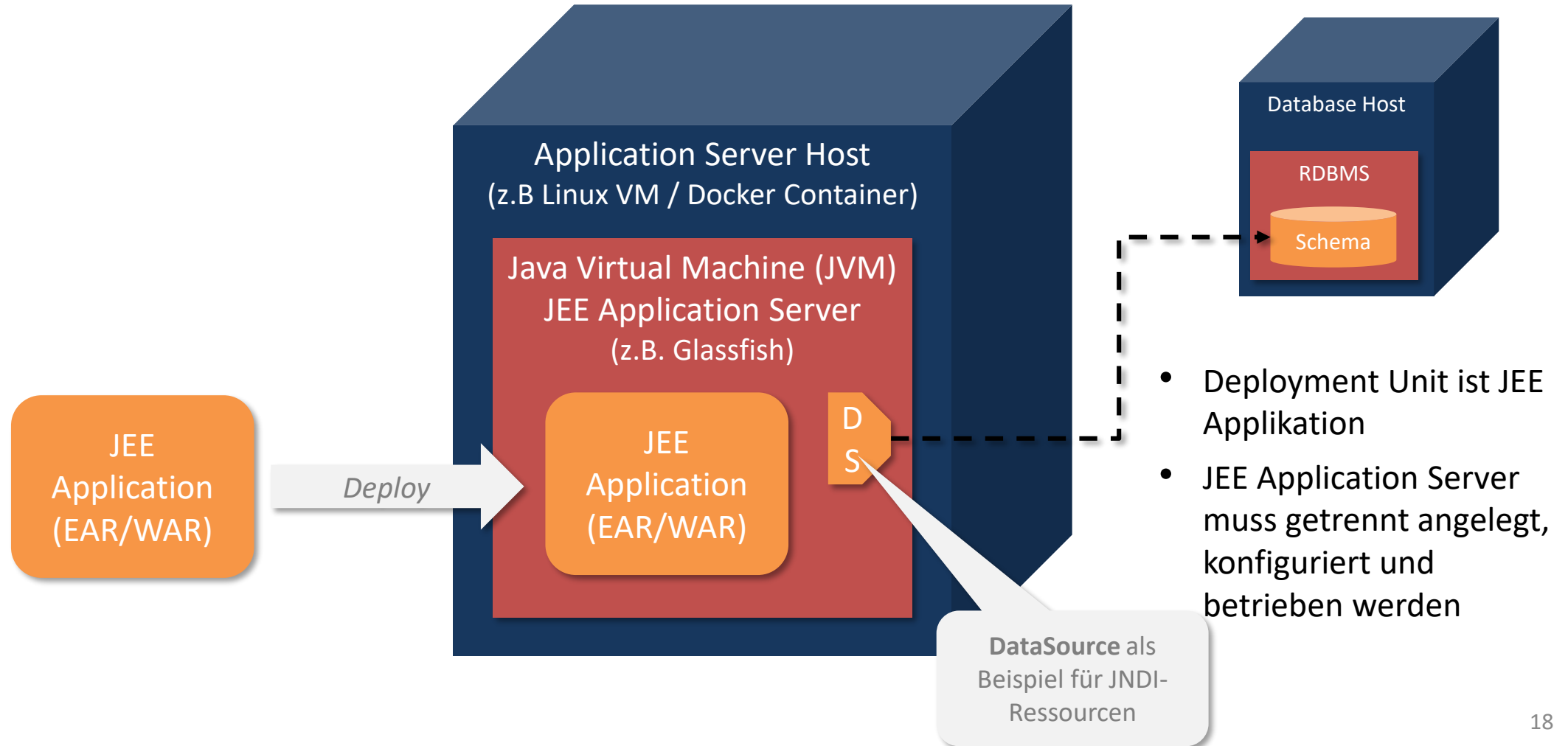
...

Alle Utility-JARs

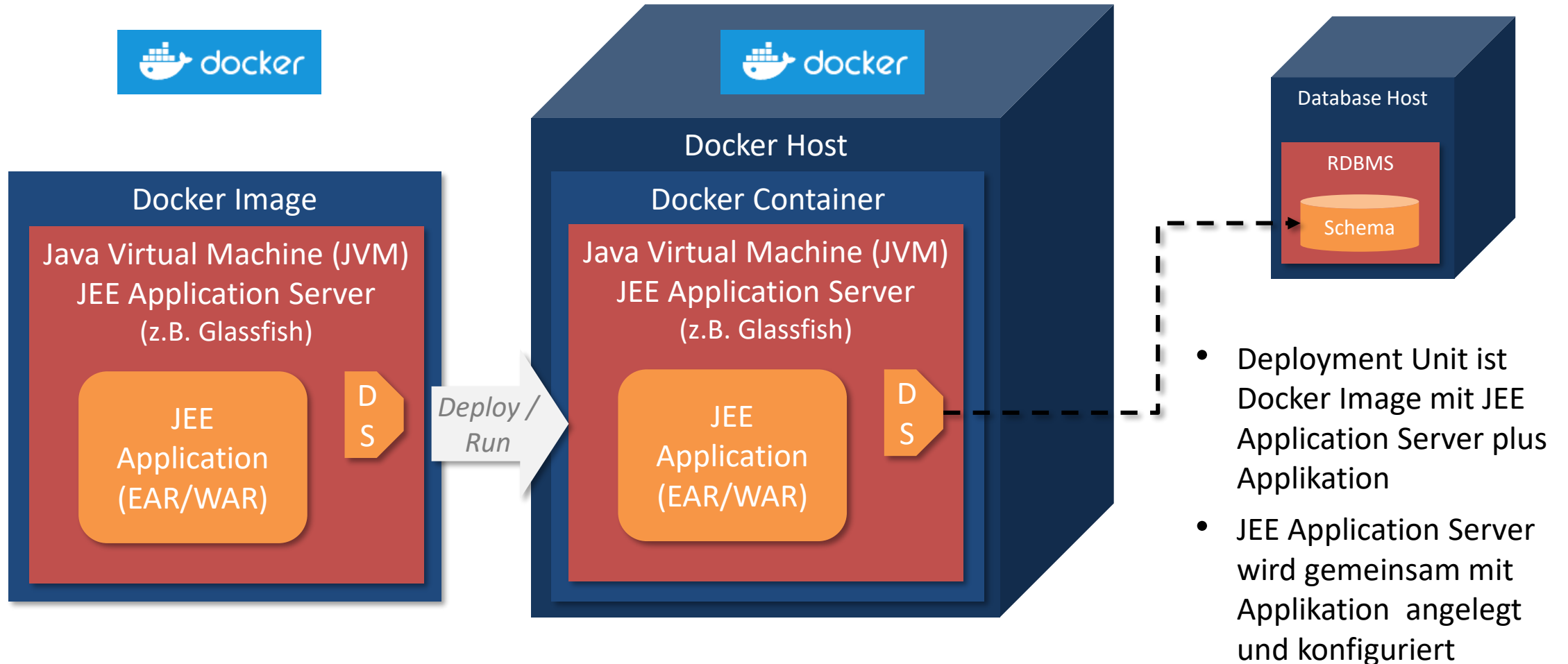
Bereitstellung von Apps zur Ausführung in einem App Server

DEPLOYMENT VON JEE-APPLIKATIONEN

Betrieb einer klassischen JEE App



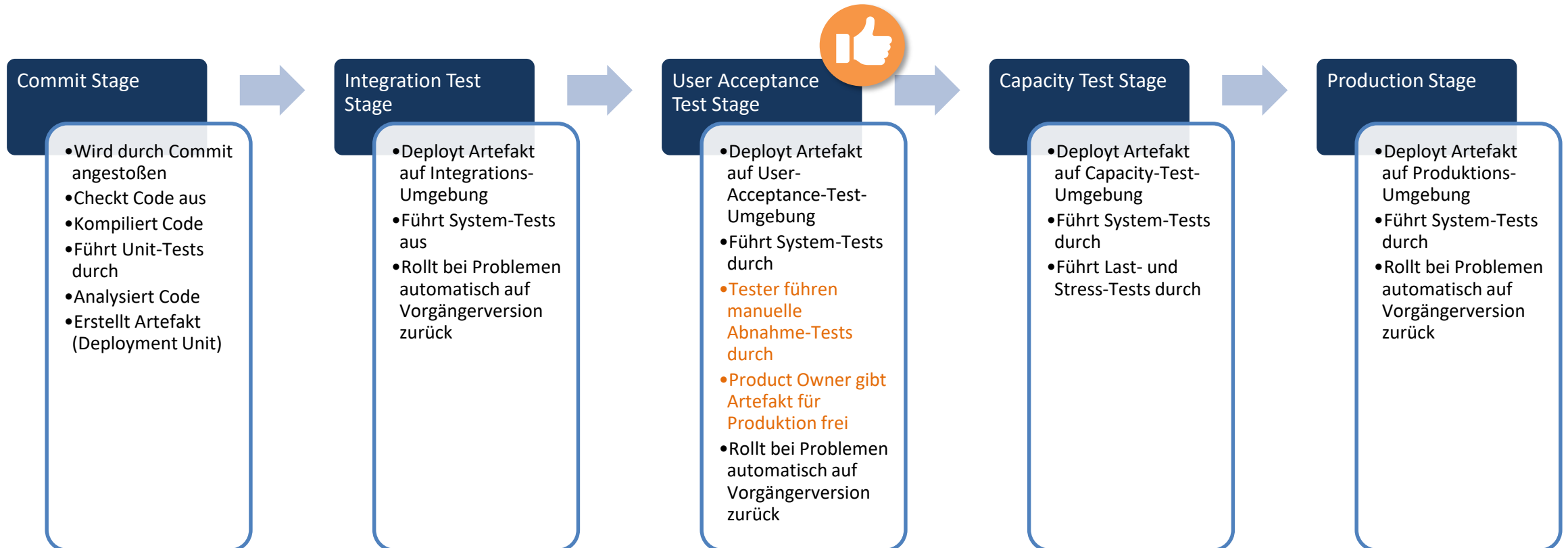
Betrieb einer containerisierten JEE App



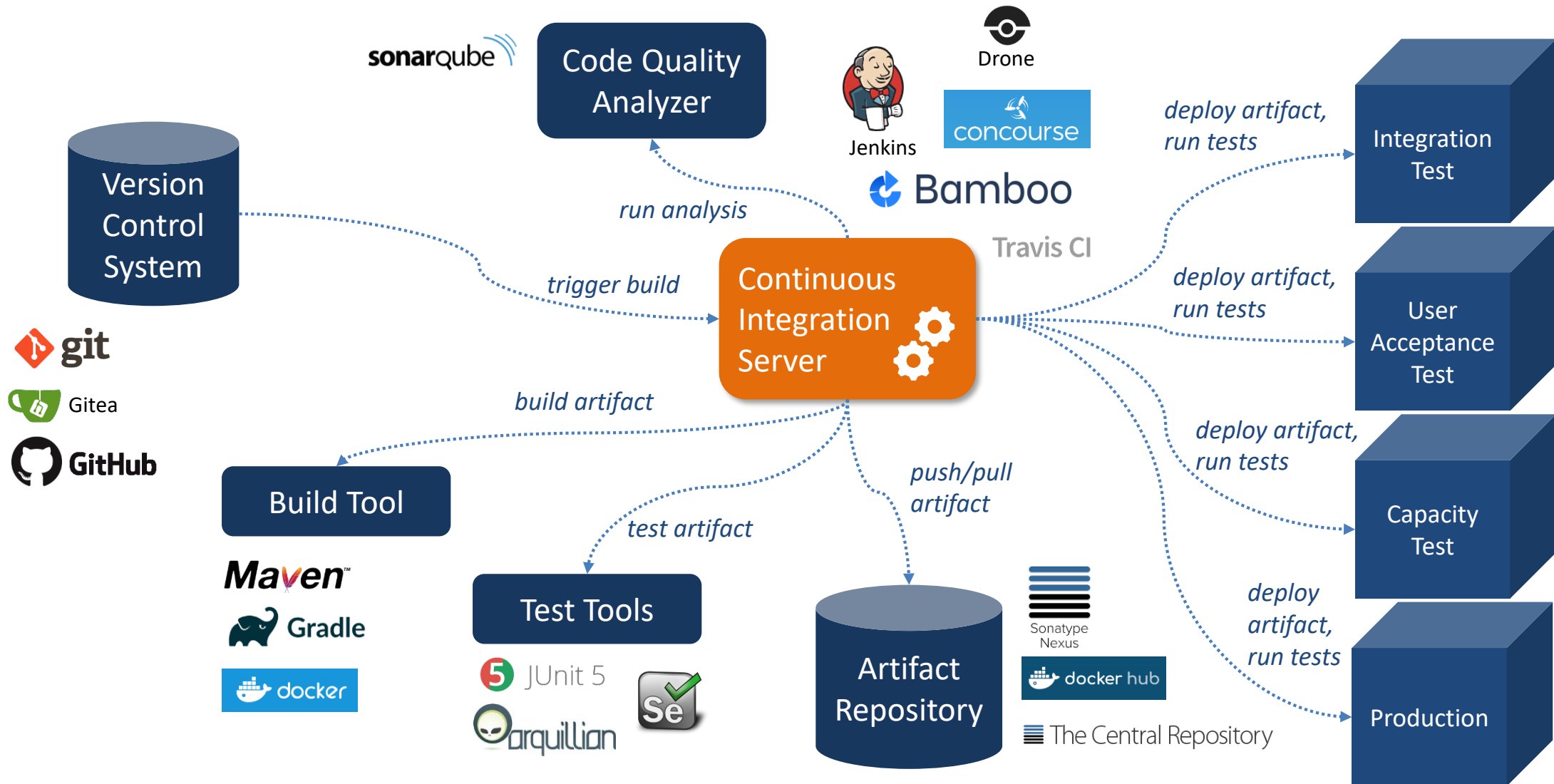
Möglichkeiten des Deployments

- Manuell über
 - ⊙ IDE (lokal)
 - ⊙ Admin Konsole des App Servers
 - ⊙ Command Line Interface (CLI) des App Servers
- Automatisiert über
 - ⊙ Maven Plugin
 - ⊙ Build Step im Continuous Integration Server (Jenkins u.ä.)

Vollautomatische Deployment Pipeline



Tools für Continuous Deployment



Fragen?



ANHANG

Quellen

- ***Java Platform Enterprise Edition (Java EE)***
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>
Offizielle Homepage für Java EE von Oracle
- ***Eclipse Enterprise for Java (EE4J)***
<https://projects.eclipse.org/projects/ee4j/>
Offizielle Homepage für Java EE der Eclipse Foundation
- ***Java EE Tutorial***
<https://javaee.github.io/tutorial/>
Java EE Tutorial Project auf GitHub (gesponsort von Oracle)
- Jez Humble, David Farley
Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation
Pearson Professional; Auflage: 01 (27. Juli 2010)
ISBN: 978-0321601919



Kontakt



Michael Theis

Lehrbeauftragter Hochschule München

email michael.theis@hm.edu

mobile + 49 170 5403805

web <http://www.tschutschu.de/Lehrauftrag.html>