

# Einführung in die Java / Jakarta Enterprise Edition

Kapitel 3 – Dienste des Applikationsservers

# 3

## **DIENSTE DES APPLIKATIONSSERVERS**

3.1

## **CONNECTORS**

- Empfängt Requests über eine Netzwerk-Verbindung
  - Auch Datei-basiertes Connectors sind möglich
- Benutzt für die Verarbeitung einen freien Thread aus dem Thread Pool
  - Durch den Pool wird eine Überlastung der Server-Ressourcen vermieden
- Delegiert die Aufrufe an einen JEE Container weiter
- Schreibt das Ergebnis zurück in das Netzwerk-Protokoll

- Die folgenden Protokolle müssen von einem JEE Applikationsserver durch Inbound Connectors unterstützt werden:
  - http/https
    - Das Internet-Protokoll für Web-Anwendungen
  - Java RMI
    - Ein Protokoll für die Kopplung zweier Java-Prozesse
  - JMS
    - Messaging

- SOAP/http
  - SOAP-basierte Web Services
- IIOP
  - Kommunikation mit CORBA-Anwendungen
- REST
  - RESTful Web Services

# Eigene Inbound Connectors

- Es können jederzeit eigene Inbound Connectors realisiert werden
  - Die JEE legt ein Programmier-API und ein Deployment-Verfahren fest
- Beispiele:
  - Raw TCP/IP
  - Datei-basierte Connectors
  - Mail-gesteuerte Connectors
  - ...

- Bindet ein Backend-System in den Applikationsserver ein
  - Zugriffe auf Backend Systeme erfolgen nie direkt aus dem Programm heraus
- Hält einen Pool von Connections und stellt diese der aufrufenden Anwendung zur Verfügung
  - Diese Connections werden in der Regel bereits beim Start des Servers angelegt
  - Die Pool-Größe kann dynamisch der Last angepasst werden
- Initiiert und steuert Transaktionen im Backend-System
- Unterstützt das vom Backend-System geforderte Authentifizierungsprotokoll



- Die folgenden Backend-Systeme müssen von Applikationsservern eingebunden werden können:
  - Datenbanken
    - Die `javax.sql.DataSource`
  - Messaging-Systeme
    - Die `javax.jms.ConnectionFactory`

# Eigene Outbound Connectors

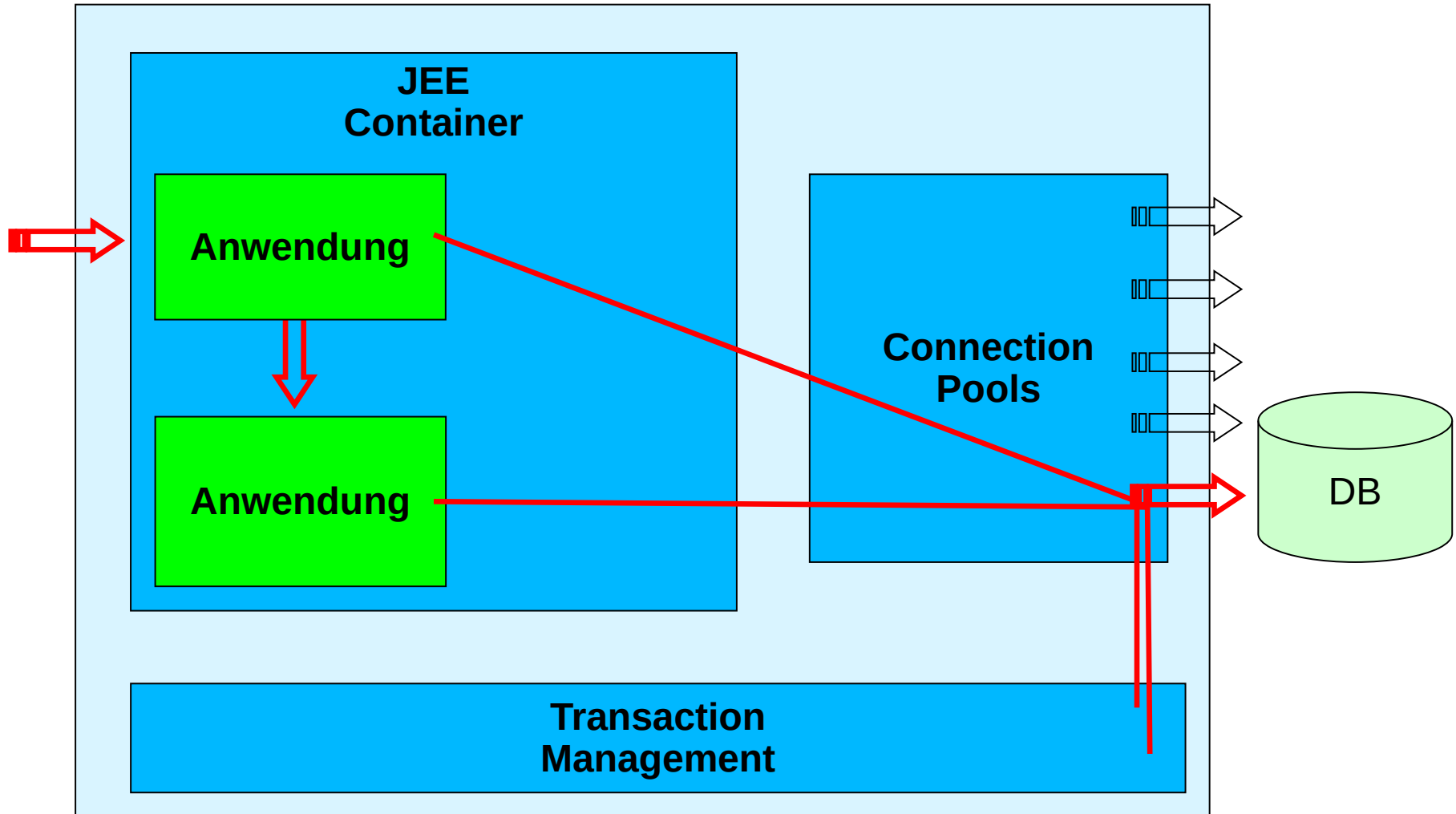
- Es können jederzeit eigene Outbound Connectors realisiert werden
  - Die JEE legt ein Programmier-API und ein Deployment-Verfahren fest
- Im Gegensatz zu Inbound Connectors sind Outbound-Connectors deutlich aufwändiger
  - Allerdings existiert ein breiter Markt von Drittanbietern
  - Auch JEE Provider statten ihre Produkte häufig mit besonderen Connectors aus
    - IBM WebSphere mit Host-Anbindung
- Beispiele:
  - Raw TCP/IP
  - File Connectors
  - SAP Connector
  - CICS-Connector
  - ...

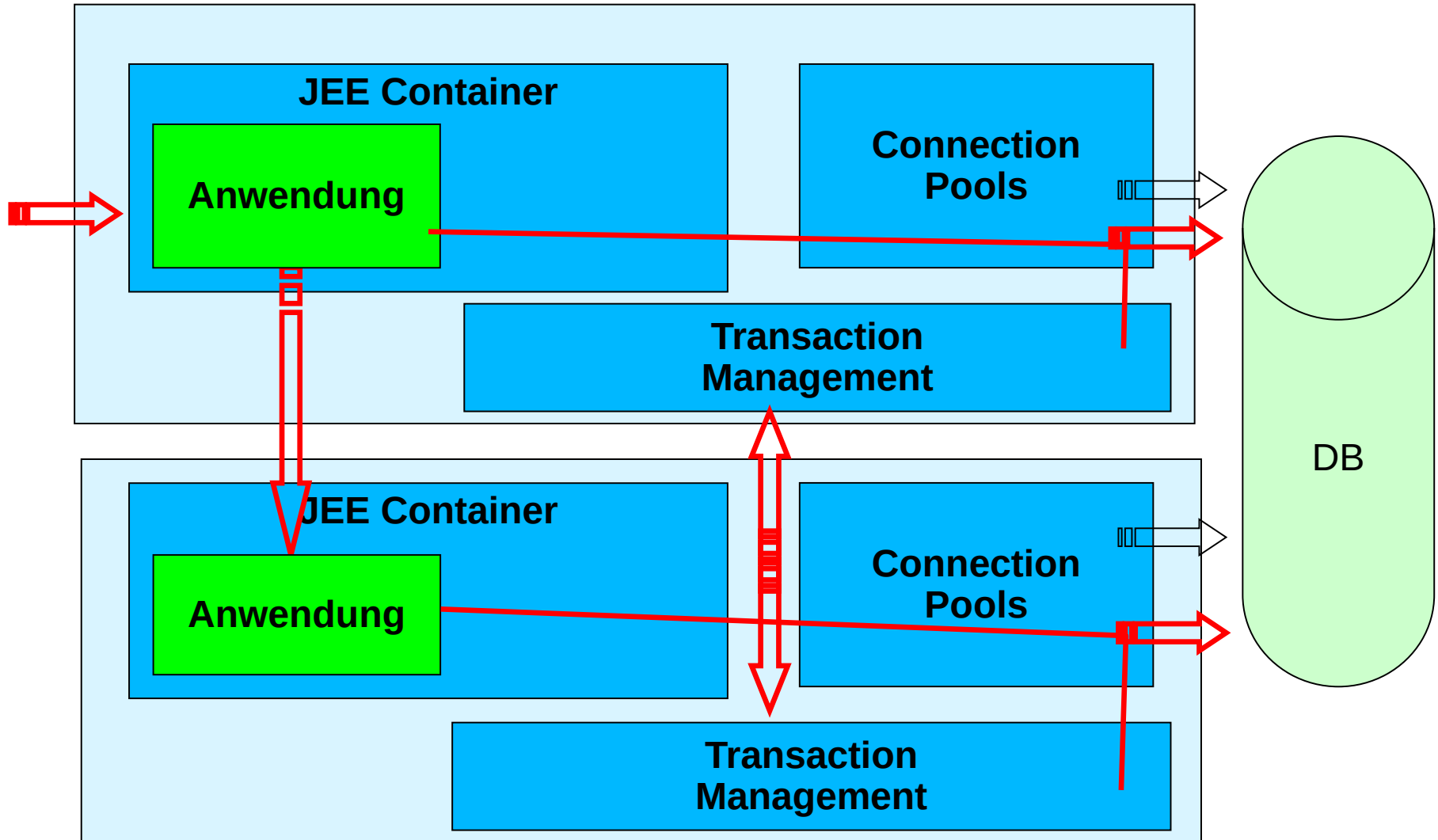
3.2

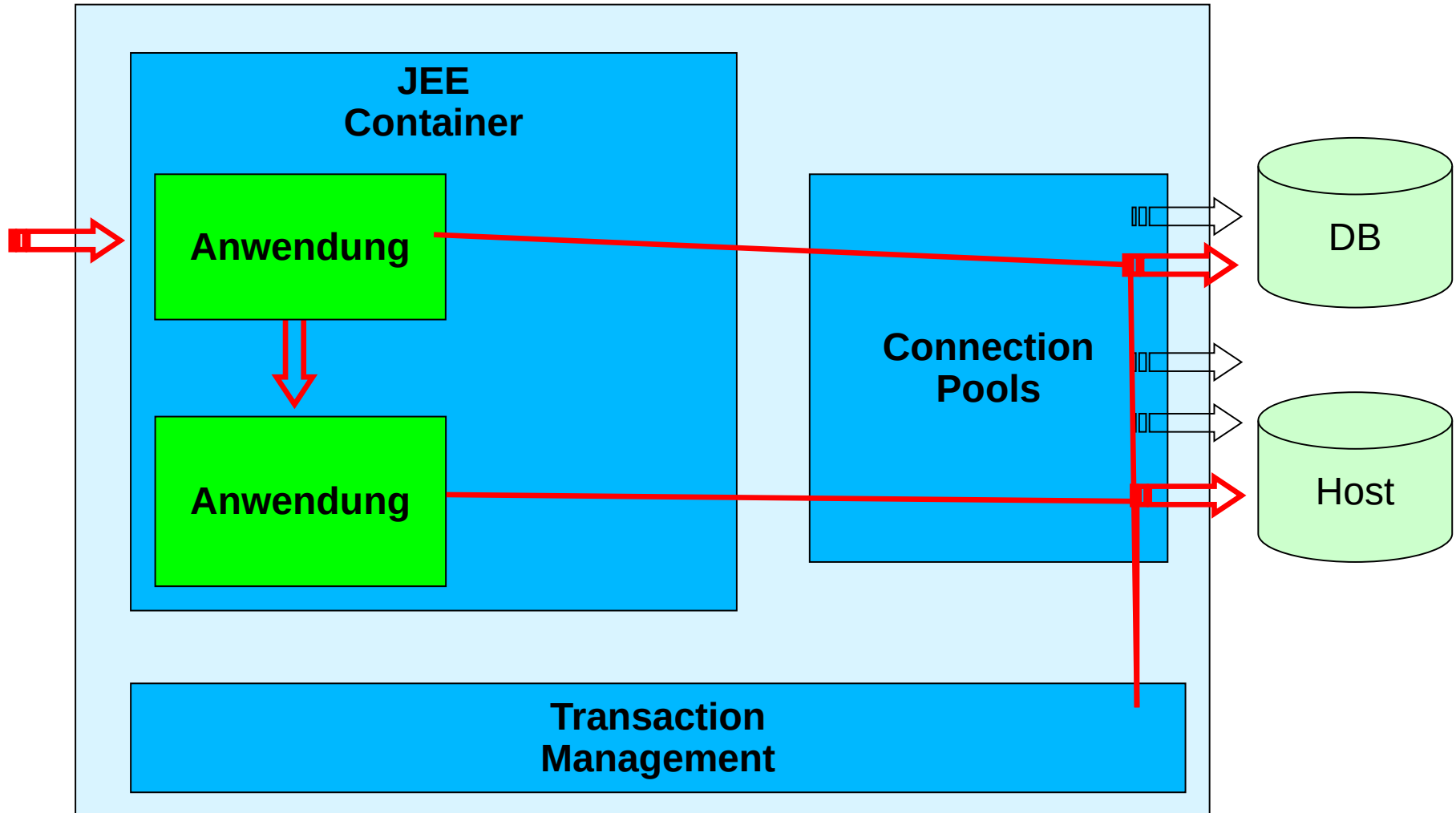
## **DER TRANSACTION MANAGER**

- Der Applikationsserver ist in der Lage, Transaktionen in Backend-Systemen zu starten und zu koordinieren
  - Dies ist die eigentliche Kern-Funktion des JEE Applikationsservers
  - Reine Web Server müssen keinen Transaktionsmanager zur Verfügung stellen
- Über Inbound Connectors kann auch eine Transaktion übernommen werden
  - Dies ist allerdings abhängig vom zu Grunde liegenden Protokoll
- Sämtliche Ressourcen-Zugriffe über Outbound Connectors werden vom Transaktionsmanager kontrolliert

- Lokale Transaktionen
  - Umspannen Aktionen innerhalb eines Applikationsservers
  
- XA Transaktionen sind
  - Verteilt über mehrere Server-Instanzen oder
  - Ressourcen-übergreifend









3.3

## **SESSION MANAGEMENT**

# Serverseitiger Zustand

- Ein Client kann bei Bedarf Zustand auf dem Server ablegen
  - Dazu definiert er eine Session
- Gründe für Serverseitigen State sind:
  - Entlastung der Client-Ressourcen
  - Verringerung der Netzwerklast
- Der Client muss den Zustand auf dem Server auch definiert wieder freigeben
  - Sonst droht eine Überlastung des Servers!
    - Auslagerung von Sessions aus dem Speicher
    - Timeout

# Identifikation des Clients

- Web Anwendungen
  - Senden pro http-Request eine Session-ID
    - Als Request-Parameter
    - Als Cookie-Parameter
- RMI-Clients
  - Können eine stehende TCP/IP-Verbindung benutzen
  - Alternativ kann auch hier eine Session ID benutzt werden
- JMS und SOAP
  - Können im aktuellen Stand der Spezifikation keinen Serverseitigen State benutzen

3.4

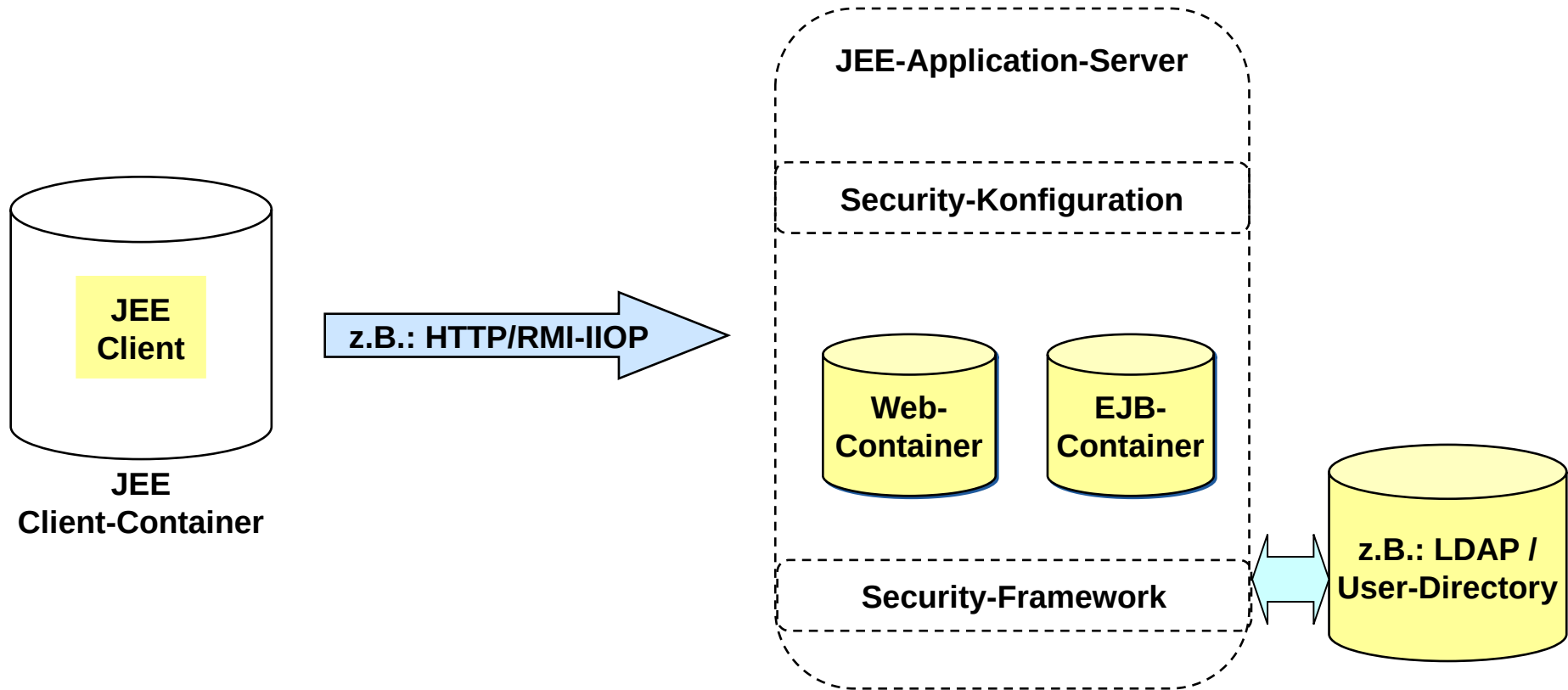
## **AUTHENTIFIZIERUNG UND AUTORISIERUNG**

- Authentifizierung
  - Ein Benutzer muss seine Identität nachprüfen lassen
- Zugriffskontrolle
  - Zuordnung von Benutzern zu Rechten in einem System
- Sichere Kommunikation
  - Die Verbindung zwischen Client und Server wird verschlüsselt

- JEE Anwendungen definieren eigene Anwendungs-Rollen
  - PowerUser
  - User
  - Guest
- Ein Benutzerverwaltungssystem definiert User und Gruppen
  - Administrator
  - Sachbearbeiter
  - „Georg Metzger“
- Ein Realm mapped die Anwender-Rollen auf User und Gruppen

# Authentifizierung und Autorisierung

- Jede Anwendung kann so konfiguriert werden, dass eine Anmeldung erforderlich ist
  - Dazu definiert sie das zu benutzende Realm
  - Das eigentliche Authentifizierungs-Verfahren wird komplett vom Applikationsserver übernommen
- Die Autorisierung erfolgt auf funktionaler Ebene
  - Seiten einer Web Anwendung
  - Methodenaufruf einer Business-Klasse





- Nennen Sie die 4 vorgestellten Dienste des Applikationsservers und erklären Sie, was darunter zu verstehen ist. Nennen Sie je ein Beispiel für ihre Verwendung