



Enterprise JavaBeans™ 2.0

Vortrag im Seminar
Komponentenbasierte Softwareentwicklung

Von
Hauke Traulsen und Sebastian Lempert



Überblick

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Unternehmensanwendungen
 - Motivation
 - Anforderungen
- Allgemeines
- J2EE
 - Einführung
 - Bestandteile
 - Architektur
 - Rollen



Überblick

Einführung

J2EE

EJB

EJB 2.0

Literatur

■ EJB

- Motivation
- Einführung
- Spezifikation
- Rollen
- Container
- Beantypen
- Bestandteile einer Bean



Überblick

Einführung

J2EE

EJB

EJB 2.0

Literatur

- EJB 2.0
 - JMS
 - MDB
 - Local Interfaces
 - CMP/CMR
 - EJB QL
 - Home Methods



Enterprise Motivation (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Verteilte transaktionsorientierte Anwendungen für Unternehmen
- serverseitige
 - Geschwindigkeit
 - Sicherheit
 - Zuverlässigkeit



Enterprise Motivation (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Schnelllebigkeit
- Bedarf in Design und Aufbau
 - Schneller
 - Weniger Geld
 - Weniger Ressourcenverbrauch



Multi-tier Architektur (1)

Einführung

J2EE

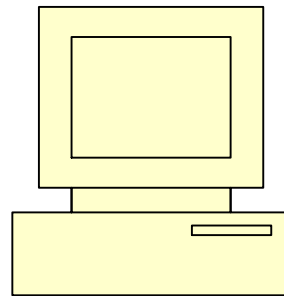
EJB

EJB 2.0

Literatur

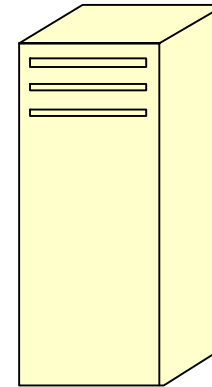
Beispiel: 3-tier-Architektur

Präsentation
(first tier)



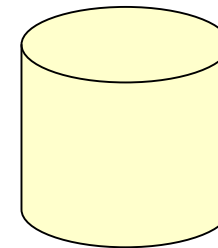
GUI

Geschäftslogik
(middle tier)



Server

Daten
(third tier)



Datenbank



Multi-tier Architektur (2)

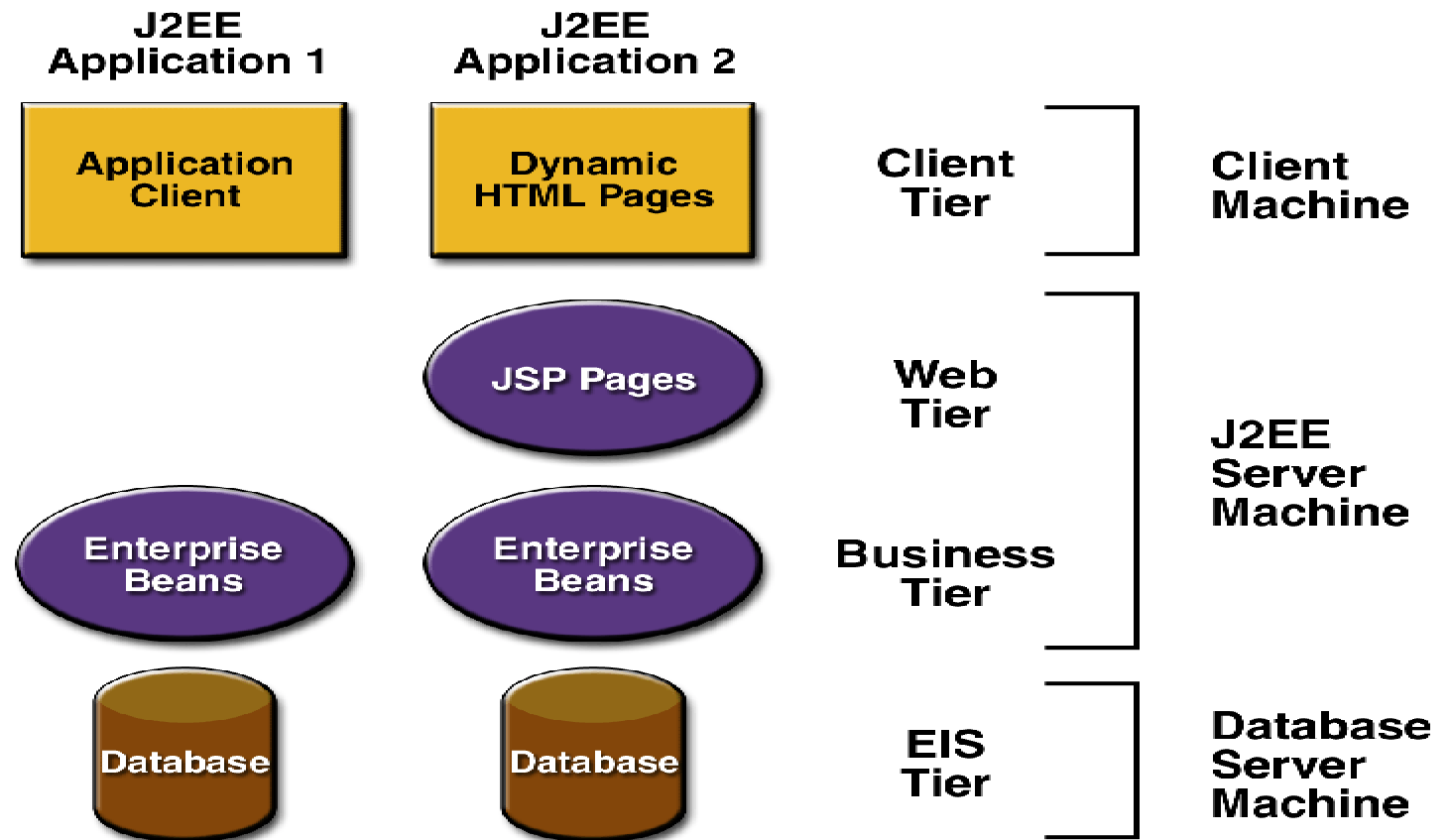
Einführung

J2EE

EJB

EJB 2.0

Literatur





J2EE – Einführung (1)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Java 2 Enterprise Edition

- Reduktion der Kosten
- Beschleunigung von Design und Entwicklung
- Komponentenbasierter Ansatz
 - Design
 - Entwicklung
 - Zusammenbau
 - Einsatz



J2EE – Einführung (2)

Einführung
J2EE

EJB

EJB 2.0

Literatur

- Verteiltes Multitier- Anwendungsmodell
- Wiederverwendbarkeit von Komponenten
- Entscheidungsfreiheit bei Wahl der Hersteller verschiedener Komponenten



J2EE - Bestandteile (1)

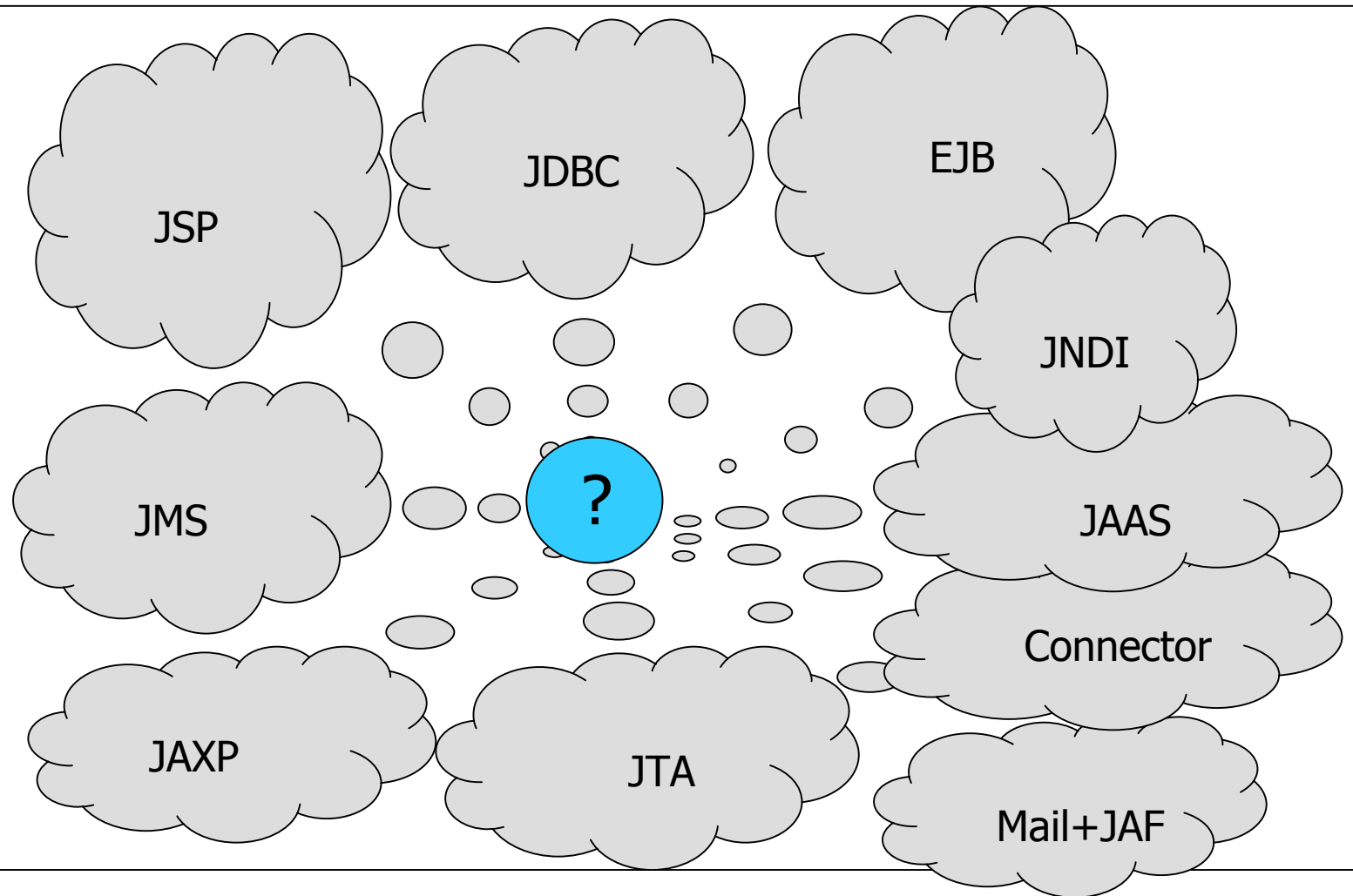
Einführung

J2EE

EJB

EJB 2.0

Literatur





J2EE – Bestandteile (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Enterprise JavaBeans (EJB)

Implementierung der Geschäftslogik

Java Server Pages (JSP)

Verbindung von Web und Geschäftslogik



J2EE – Bestandteile (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Java Database Connectivity (JDBC)

Zugriff auf Datenbank aus EJB oder JSP heraus

Java Message Service (JMS)

asynchrone Interprozess-Kommunikation



J2EE – Bestandteile (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Java Naming And Directory Interface (JNDI)

Lookup und Assoziation von Objekten durch deren Attribute

Java Transaction API (JTA)

Transaktionsmanagement (RBs,Commits)



J2EE – Bestandteile (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Java Mail und Java Activation Framework (JAF)

E-Mails versenden

Java Mail benutzt JAF (Datenzugriff)

Java API for XML Processing (JAXP)

XML Verarbeitung



J2EE – Bestandteile (6)

Einführung

J2EE

EJB

EJB 2.0

Literatur

J2EE Connector Architecture

Ressourcenzugriff aus J2EE heraus auf
andere Informationssysteme (EIS-Tier)

Java Authentication and Authorization Service (JAAS)

Überprüfung von Aufrufsberechtigungen eines Nutzers
oder einer Nutzergruppe



J2EE - Architektur

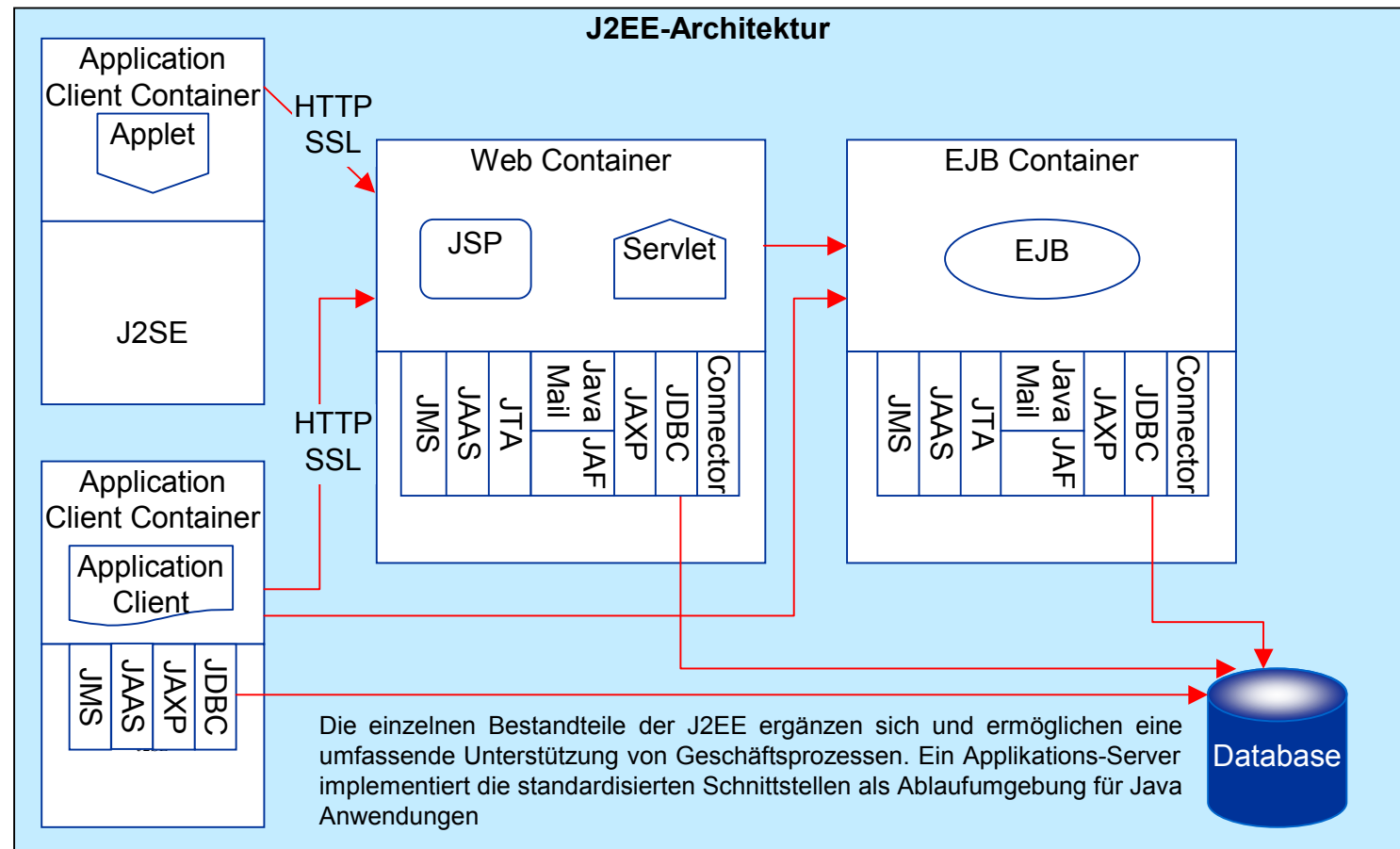
Einführung

J2EE

EJB

EJB 2.0

Literatur





J2EE – Rollen (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Module
- Teilung des Entwicklungsprozesses
- Teilung des Einsatzprozesses



J2EE – Rollen (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

J2EE Product Provider

- Hersteller von OS, DBS, App.-/Web-Server
- Implementieren J2EE-Spezifikation
- → J2EE – Server

J2EE Tool Provider

- Tools für Entwicklung, Zusammenbau und Einsatz



J2EE – Rollen (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Application Component Provider

Anbieter von Enterprise Beans, Web Komponenten, Clients, etc

Application Assembler

Komponenten werden zusammengeführt
(EJB-JARs, WARs → EAR)



J2EE – Rollen (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Application Deployer

Einbau der J2EE Anwendung, Administration,
Kontrolle der Laufzeitumgebung



J2EE – Spezifikation

Einführung

J2EE

EJB

EJB 2.0

Literatur

- J2EE

Spezifikation

1.3 vom 27.07.2001

- J2EE SDK

Referenzimplementierung

SDK 1.3.1 vom 31.01.2002

inklusive Dokumentation 25 MB



EJB - Motivation

Einführung

J2EE

EJB

EJB 2.0

Literatur

Vereinfachte Entwicklung kommerzieller Anwendungen

Entwickler von kommerziellen Anwendungen sollen sich auf die Implementierung der Geschäftslogik, d. h. das Codieren von Geschäftsregeln, -logik und -wissen, konzentrieren können.



EJB - Die EJB Spezifikation (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

... ist eine

Standardisierte API für die Entwicklung
komponentenbasierter Software

... ist

ein Bestandteil der J2EE-Spezifikation

... wird entwickelt von

vielen Leuten, vielen Firmen (Borland, Fujitsu, IBM,
Oracle, Sun, Sybase etc.) und dem Java Community
Process



EJB - Die EJB Spezifikation (2)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Überblick über die Versionen

- Version 1.0, 21.03.1998, 181 Seiten
- Version 1.1, 17.12.1999, 314 Seiten
- Version 2.0, 22.08.2001, 572 Seiten
- Version 2.1, 17.06.2002, 636 Seiten (Public Draft)



EJB – Einführung (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Was sind EJB?

„The Enterprise JavaBeans architecture is a *component architecture* for the development and deployment of component-based *distributed business applications*. Applications written using the Enterprise JavaBeans architecture are *scalable, transactional, and multi-user secure*. These applications may be *written once, and then deployed on any server platform* that supports the Enterprise JavaBeans specification.“

Quelle: Sun microsystems
(EJB Specification 2.0 vom 22.08.2001)



EJB – Einführung (2)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Was sind EJB?

„Enterprise JavaBeans (EJB) components execute in a *managed environment* that supports *transactions*. Enterprise beans typically contain the *business logic* for a J2EE application.“

Quelle: Sun microsystems
(J2EE Specification 1.3 vom 27.07.2001)



EJB – Einführung (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- **Komponentenmodell**

Serverseitige Komponentenarchitektur für die J2EE-Plattform

- **Verteilte Anwendungen**

Ein Client nutzt Funktionalität eines Servers über Fernaufrufe (RPC / ROI), hier Java RMI

- **Write once, run anywhere (WORA)**

Ein EJB läuft in jedem Server, der sich an die Spezifikation hält.



EJB – Einführung (4)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Transaktionen und EJB

- EJB laufen in einem Container, welcher laut J2EE Transaktionen unterstützen muss
- Protokoll: Two-Phase-Commit (2PC) basierend auf dem OMG Transaction Service (OTS)
- implementiert im Java Transaction Service (JTA)
- Unterstützung für mehrere Transaktionsmanager (transactional resource manager)
- dadurch Überführung von Datenbanktransaktionen auf Objekte



EJB Architektur (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

... besteht aus

- einem EJB-Server
- EJB-Containern, die auf diesem Server laufen
- sog. "Home Objects" , sog. "Remote EJB Objects" und Enterprise JavaBeans die in den Containern laufen
- mindestens einem EJB-Client
- weiteren Hilfssystemen wie JNDI, JTS, Sicherheitsdiensten etc.



EJB Architektur (2)

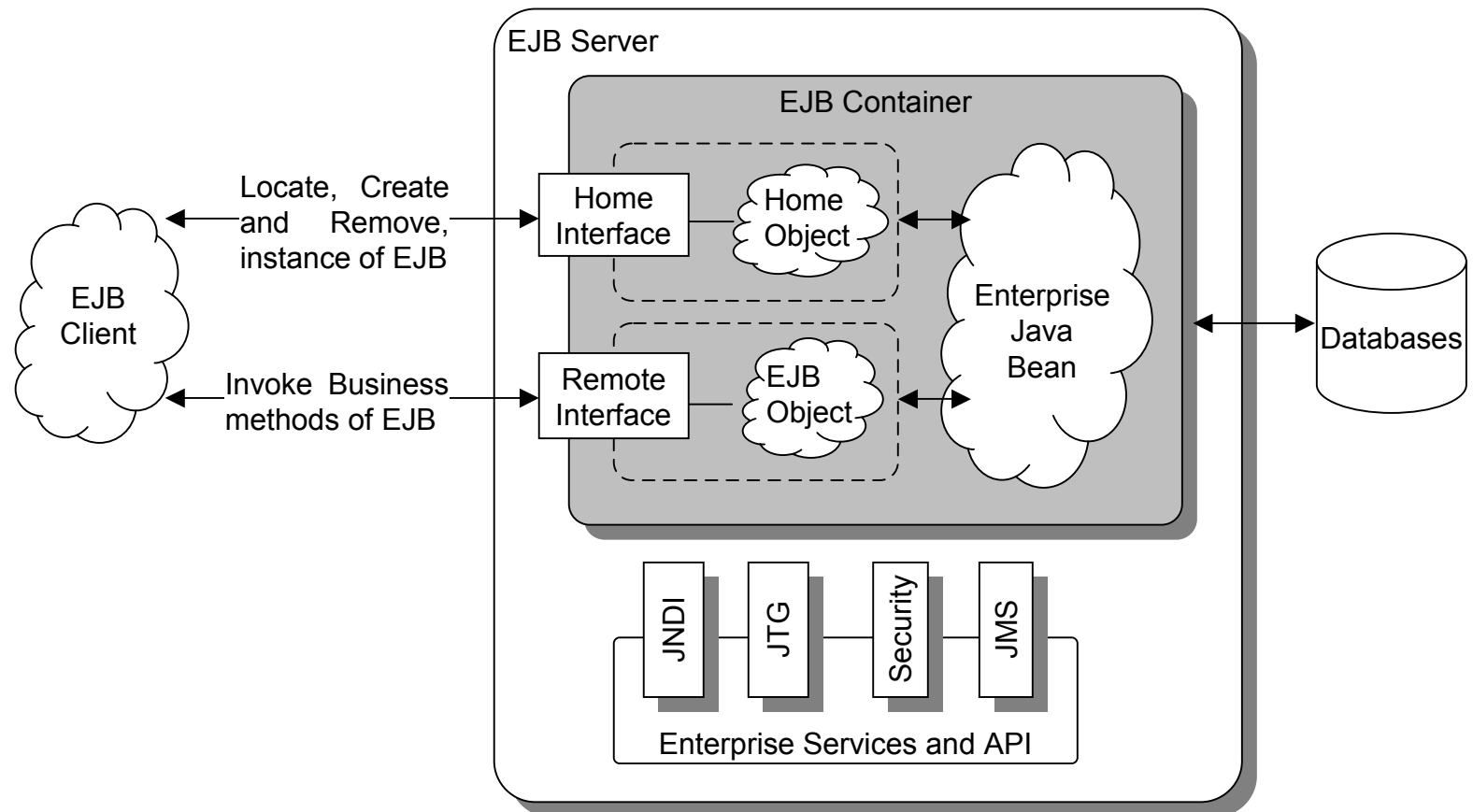
Einführung

J2EE

EJB

EJB 2.0

Literatur





EJB – Container: Spezifikationsanforderungen (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Ressourcen und Lebenszyklusmanagement

- Verteilung der Ressourcen auf die unterschiedlichen EJB eines Containers.
- Erzeugung und Entfernung von Bean-Instanzen.

Zustandsverwaltung / Persistenz

- Der Zustand einer zustandsbehafteten Bean kann abgespeichert bzw. persistent gemacht und wieder geladen werden.



EJB - Container: Spezifikationsanforderungen (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Transaktionsmanagement

Sicherheit

- Für den Ablauf einer EJB können verschiedene Sicherheitsklassen definiert werden.
- Durch Java Security Manager implementiert.

RPC und Ortstransparenz

- Fernaufrufe sind durch RMI implementiert.
- Ortstransparenz ist gegeben.



EJB - Container: Spezifikationsanforderungen (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Hilfswerkzeuge

- Ein EJB Container besitzt eine gewisse Grundausstattung an Hilfswerkzeugen.
- Erzeugung von Stubs und Skeletons.
- Logik, die für die verschiedenen Dienste notwendig ist (z.B. Implementierung eines Namensdienstes)



EJB Container: Zusätzliche Features (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Lastbalancierung

- Ressourcenaufteilung von gleichartigen EJB-Containern untereinander.

Komponentenaustausch zur Laufzeit

- Einspielung neuer Versionen zur Laufzeit.

Verteilte Transaktionen

- Transaktionen über mehrere Container und Server hinweg.



EJB Container: Zusätzliche Features (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

XML-Unterstützung

- ... zur Repräsentation der Daten

Integration von CORBA

- CORBA-Unterstützung durch das Protokoll RMI-IIOP:
CORBA-Objekte kommunizieren mit Java Remote Objects



EJB Architektur: Beantypen (3)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Session Beans

- Modellierung von Geschäftsprozessen
- enthalten Geschäftslogik
- Lebenszeit entspricht in etwa einer Client-Session
- sind nicht nebenläufig, pro Session Bean immer nur ein Client zur gleichen Zeit



EJB Architektur: Beantypen (4)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Typen von Session Beans

- *Stateful Session Beans*: Zustandsbehaftete Beans, z.B. für einen elektronischen Warenkorb
- *Stateless Session Beans*: Zustandslose Beans.



EJB Architektur: Beantypen (5)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Entity Beans

- Modellierung von Daten: Objektorientierte Ansicht der gerade benötigten Daten aus der Datenbank.
- Somit nur lokale Kommunikation beim Zugriff auf die Daten = erhöhte Performance.
- Daten lassen sich leichter abfragen und modifizieren.
- nebenläufig: Gleichzeitige Nutzung durch mehrere Clients



EJB Architektur: Beantypen (6)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Typen von Entity Beans

- *Bean-Managed Persistent Entity Bean*: Persistenzmanagement von Hand
- *Container-Managed Persistent Entity Bean*: ...



Bestandteile einer EJB (1)

Einführung
J2EE

EJB

EJB 2.0

Literatur

... laut J2EE Spezifikation

- die eigentliche Enterprise JavaBean
- das EJB Object samt Remote Interface
- das Home Object samt Home Interface
- der Deployment Descriptor / das Manifest (Verwaltungsinformationen)
- die EJB-Jar-Datei



Bestandteile einer EJB (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

EnterpriseBean Interface

```
public interface javax.ejb.EnterpriseBean  
extends java.io.Serializable {}
```

- muss von jeder EJB implementiert werden
- ist ein Marker Interface: Jede Klasse, es implementiert ist automatisch eine EJB
- erbt von Serializable, somit ist jede EJB serialisierbar.



Bestandteile einer EJB (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

SessionBean Interface

```
public interface SessionBean  
extends EnterpriseBean {...}
```

EntityBean Interface

```
public interface EntityBean  
extends EnterpriseBean {...}
```



Bestandteile einer EJB (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Remote Interface und EJB Object

```
public interface EJBObject extends Remote {...}
```

- Das EJB Object ist ein Stellvertreter für die EJB mit einer Schnittstelle nach außen; ähnlich Skeleton aus CORBA
- Aufrufe werden über das EJB Object und den Container an die EJB weitergeleitet.
- Das EJB Object wird durch das Remote Interface beschrieben und automatisch vom Container generiert.
- Das EJB Object ist Container-abhängig



Bestandteile einer EJB (5)

Einführung

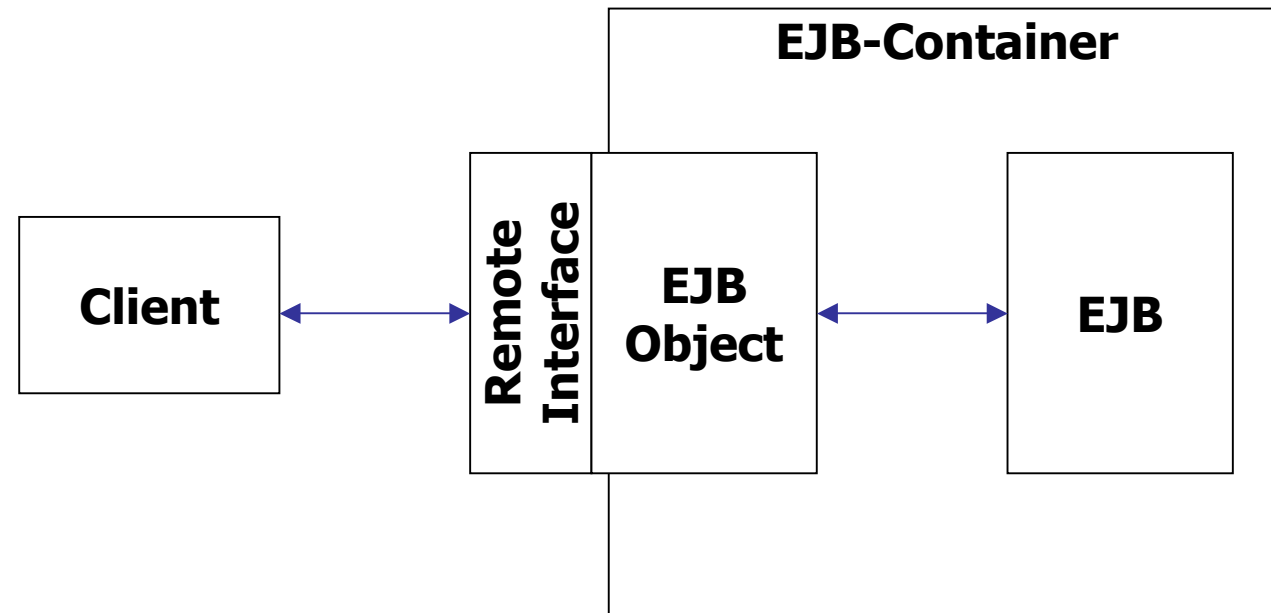
J2EE

EJB

EJB 2.0

Literatur

Kommunikation mit einer EJB





Bestandteile einer EJB (6)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Home Interface und Home Object

```
public interface EJBHome extends Remote {...}
```

- Dem Home Object kommt die Rolle einer Object Factory zu: Sie liefert eine Referenz auf ein EJB Object zurück.
- Das Home Objects wird durch das Home Interface beschrieben und automatisch vom Container generiert.
- Das Home Object ist Container-abhängig.



Bestandteile einer EJB (7)

Einführung

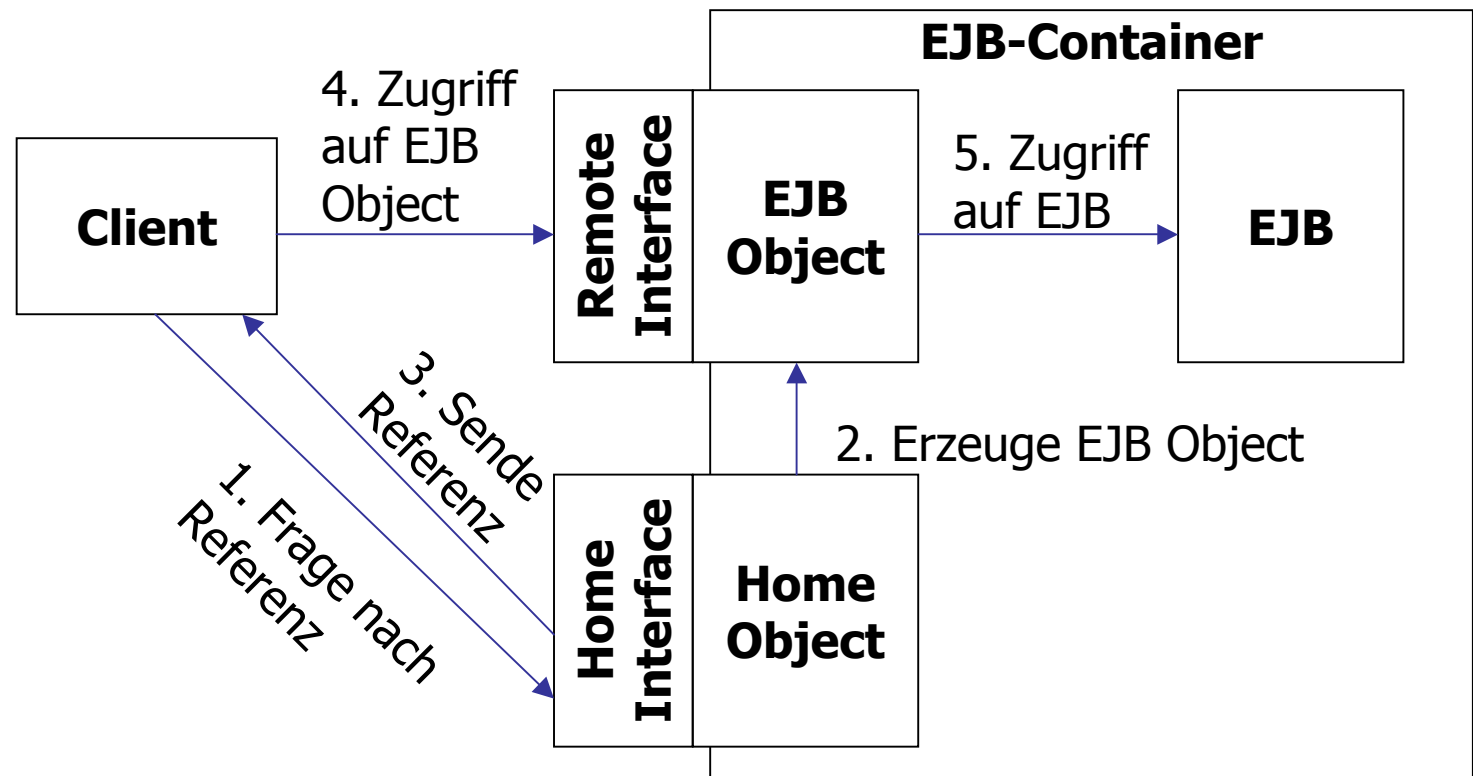
J2EE

EJB

EJB 2.0

Literatur

Erzeugung von EJB-Instanzen





Bestandteile einer EJB (8)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Deployment Descriptor

- Für Verwaltungsinformationen zuständig.
- legt fest, welche Dienste des EJB-Containers durch die Bean genutzt werden sollen.
- Bean-Management: Namen der EJBs, zugehörige Objekte und Schnittstellen, Bean-Typ
- Persistenz: Betrifft Entity Beans (container managed?)
- Transaktionsmanagement
- Sicherheit



Bestandteile einer EJB (10)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Die EJB-Jar-Datei

- Auslieferungsformat einer EJB: Sämtliche Dateien werden in eine JAR-Datei gepackt.
- EJB Object bzw. Home Object sind nicht enthalten, da vom Container automatisch generiert ... dafür aber notwendigerweise Remote Interface bzw. Home Interface!



EJB - Die sechs Rollen (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Enterprise Bean Provider

- Bean Hersteller. Erzeugnis: EJB-JAR-File.

Application Assembler

- Fügt mehrere EJB zu einer größeren Anwendung zusammen. Erzeugnis: EJB-JAR-File.

Deployer

- Integriert eine gelieferte EJB in das System, bringt die EJB in einem Container zum Laufen.



EJB - Die sechs Rollen (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

EJB Server Provider

- Spezialisiert auf das Gebiet des verteilten Transaktionsmanagements (z.B. OS-, Middleware- bzw. Datenbankhersteller).
- Momentan ist diese Rolle gleichbedeutend mit der des EJB Container Providers.

EJB Container Provider

- Stellt Deployment-Tools und die Laufzeitumgebung für EJB zur Verfügung
- Häufig Server/Container Provider in Einem



EJB - Die sechs Rollen (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

System Administrator

- Verantwortlich für die Konfiguration und Administration der Rechner- und Netzwerkinfrastruktur.
- Überwachung von EJB Server und Container.



EJB Architektur: Namensdienst (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

JNDI

- Java Naming and Directory Interface
- Wird vom Client angesprochen, um das Home Object finden zu können.
- Eigentlich nur eine Spezifikation: Muss von dem Container-Hersteller bereitgestellt werden.



Zusammenfassung

Einführung

J2EE

EJB

EJB 2.0

Literatur

Vorteile

- Unabhängigkeit von zugrundeliegender Middleware
- Plattformunabhängigkeit und WORA
- Strukturierte Anwendungsentwicklung durch Rollen
- (verteiltes) Transaktionsmanagement
- Portierbarkeit und Skalierbarkeit
- Nahtlose Integration von CORBA



EJB 2.0

Einführung

J2EE

EJB

EJB 2.0

Literatur

EJB 2.0 Spezifikation

- Final Release vom 22.08.2001
- 2/3 umfangreicher als 1.1
- 572 Seiten



EJB 2.0 – JMS (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Java Message Service
- Seit 2.0 in J2EE integriert
- Jedes Element von J2EE →
Producer / Consumer von Nachrichten



EJB 2.0 – JMS (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Konzept eines JMS-Servers
- Häufig in den EJB-Applicationsserver integriert
- Verwaltet **Topics** und **Queues**



EJB 2.0 – JMS (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

■ Pub/Sub

- Übermittlung einer Nachricht an mehrere Consumer (publish)
- Kommunikationskanal: „Topic“
- An-/Abmeldung von Consumer an einem Topic (subscribe)
- Listenerkonzept



EJB 2.0 – JMS (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

■ P2P

- Point To Point
- Auswahl eines von mehreren potentiellen Empfängern
- Viele Produkte: Auswahl nach Lastverteilung
- Kommunikationskanal: „Queue“



EJB 2.0 – JMS (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Queues und Topics per Naming-Service verfügbar
- Steuerung von Web-Komponenten, Session Beans und Entity Beans über RMI-IIOP: Synchron
- Client-Applikation und **Message Driven Beans:** Asynchron



EJB 2.0 – MDB (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

Message Driven Beans

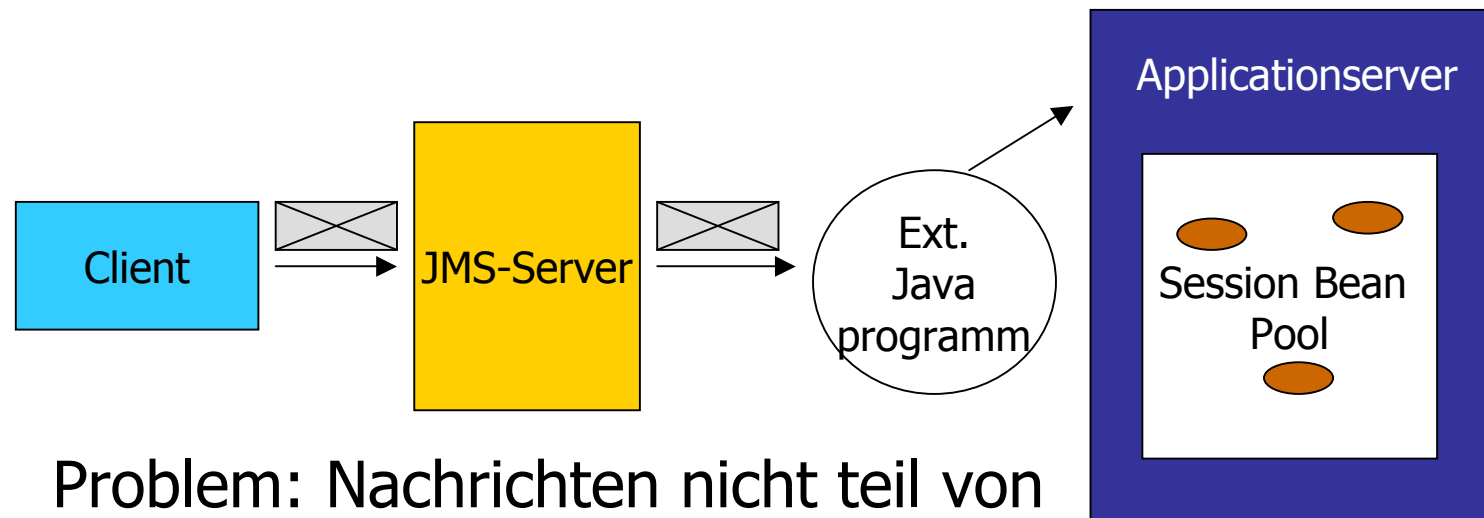
- Neuer (dritter) Beantyp
- Möglich durch Integration von JMS
- Asynchrone Methodenaufrufe
- Per Definition: JMS-Consumer
 - Kann dennoch Nachrichten senden



EJB 2.0 – MDB (2)

Einführung
J2EE
EJB
EJB 2.0
Literatur

Bis Spezifikation 1.1:



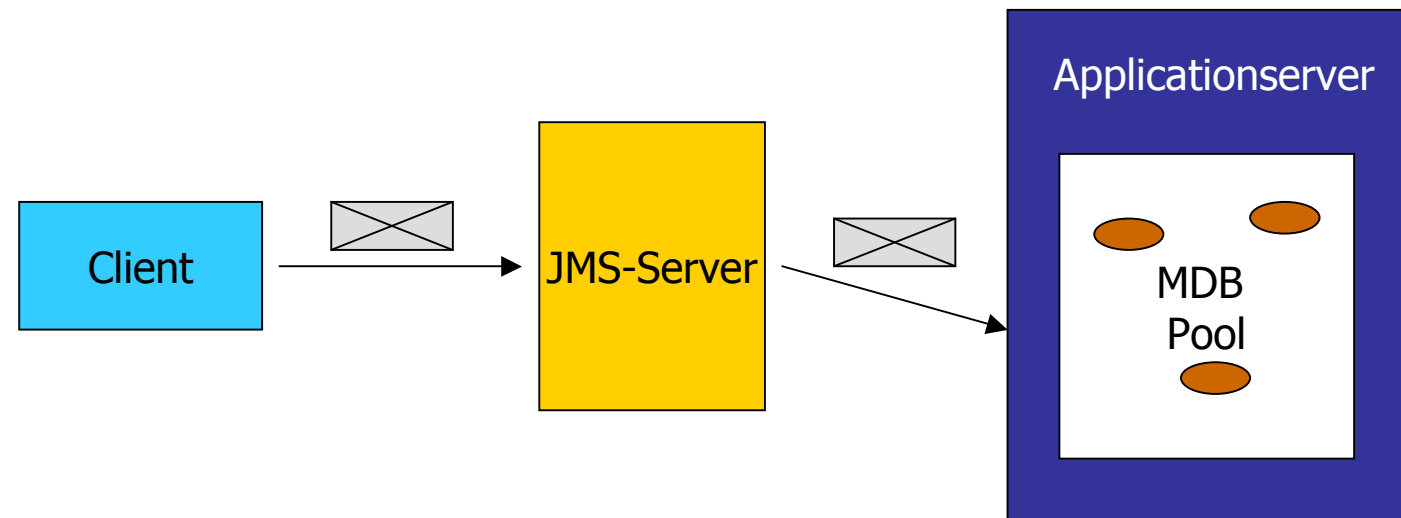
Problem: Nachrichten nicht teil von
Transaktionen



EJB 2.0 – MDB (3)

Einführung
J2EE
EJB
EJB 2.0
Literatur

■ Spezifikation 2.0





EJB 2.0 – MDB (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Implementiert zwei Interfaces
 - `javax.ejb.MessageDrivenBean`
 - Methoden zum Erzeugen neuer Beans
 - Vorbereitung der Verarbeitung von Nachrichten
 - `javax.jms.MessageListener`
 - *`public void onMessage(Message m);`*
 - Vom Container genutzt, um Bean Nachricht zu übermitteln



EJB 2.0 – MDB (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- MDB ähnlich **Stateless Session Beans**
- geeignet zum Instance - Pooling
- Lastverteilung

- Immer vom Container aufgerufen
 - Darum: kein Home-Interface
 - Unsichtbar für Client



EJB 2.0 – MDB (6)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Definition von Selector möglich
 - Problem: Erzeugung der Instanzen unsichtbar für den Entwickler durch den Container
 - Lösung: Deployment Descriptor bietet Elemente zur Definition

<container-transaction>

<exclude-list>

<method-permission>



EJB 2.0 – MDB (7)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Concurrency hier erlaubt
 - Keine Garantien über die Ausführungsreihenfolge von eintreffenden nebenläufig verarbeiteten Nachrichten
 - Vorbereitung der Bean auf Nebenläufigkeit

- Sonst nicht erlaubt
 - Container verliert Teil der Kontrolle



EJB 2.0 – Local Interfaces (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Spezifikation 1.x
 - Home-Interface (Lebenszyklus)
 - Remote-Interface (Geschäftslogik)
 - Generelle Ortsunabhängigkeit (RMI)
 - Kommunikation zwischen Beans auch in der gleichen JVM
- immer über RMI
- Serialisierung
- bis 10000 mal langsamer als lokal



EJB 2.0 – Local Interfaces (2)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Spezifikation 2.0
 - Einführung des Local Interface – Konzeptes
 - Parameterübergabe innerhalb einer JVM damit per Referenz
 - Somit Aufgabe der Ortstransparenz



EJB 2.0 – Local Interfaces (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

■ Motivation

- Sowohl grobe als auch feine Sicht der Beans möglich
- fein = stark voneinander abhängige Beans
- Feine Sicht wird durch Local Interface beschleunigt



EJB 2.0 – Local Interfaces (4)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Neu: Zwei Gruppen von Interfaces
 - Home – Interfaces
 - Erzeugen, Suchen, Löschen, Neu: „Home-Methods“
 - Component Interfaces
 - Ersetzt Begriff „Remote-Interface“
 - Beschreibt Zugriff auf die Geschäftslogik



EJB 2.0 – Local Interfaces (5)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Interfaces dieser Gruppen wahlweise als **Local Interface** oder **Remote Interface**
- Resultat: Vier mögliche Interfaces
 - Local Home
 - Local Component
 - Remote Home
 - Remote Component



EJB 2.0 – Local Interfaces (6)

Einführung
J2EE

EJB

EJB 2.0

Literatur

- Möglichkeit vorhanden, nur Remote- bzw. Local-Interfaces zu verwenden
- Rückgabewerte entsprechend
 - *Local*/Home Methoden geben *Local*/Component Interfaces zurück



EJB 2.0 – Local Interfaces (7)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Kritische Betrachtung
 - Stark vernetzte Beans mit Local Interfaces (Clustering)
 - Fassaden- Bean
 - Kommuniziert mit Cluster-externem
 - Cluster-interne Bean
 - Wird nur von Beans desselben Clusters angesprochen



EJB 2.0 – Local Interfaces (8)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Forced Facades

- Cluster-interne EJBs, die als Rückgabewert an Elemente ausserhalb des Clusters auftreten, brauchen (z.B. im Falle eines Remote Clients) ein erzwungenes Remote-Interface
- Problem: Abhängige Local Interface-Rückgabewerte führen zu weiteren Forced Facades



EJB 2.0 – Local Interfaces (9)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Problem bei Dual Interfaces:
 - Cluster-Bean erschafft via LocalHome eine Instanz einer anderen Bean
 - Folge: Hat ein Local Interface bekommen
 - Rückgabe zum Remote Client: Transformation einer Lokalreferenz in eine Remotereferenz notwendig



EJB 2.0 – Local Interfaces (10)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Transformationstypen
 - Methode steht bei erschaffener Instanz zur Verfügung (getRemote...)
 - ... sonst: Eigenregie
 - Via Local Interface Primary Key der Instanz erfragen
 - Via JNDI das Remote Home Interface holen
 - Via Find-Methode mit Primary Key das Remote Component Interface holen



EJB 2.0 – Local Interfaces (11)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Schwierigkeit vorherzusehen, welche Methoden in welchem Interface (= Remote oder Local) benötigt werden
- EJB Designer realisiert Grossteil der Methoden in beiden Interfaces
- Mehr Entwicklungsarbeit, unterschiedliche Semantik (?) nach außen gleicher Methoden
- größerer Deployment Descriptor



EJB 2.0 – Local Interfaces (12)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Anderer Ansatz:
 - Cluster-Bean-Hersteller soll nicht zu einem Remote Interface gezwungen werden
 - Definition: Aktivität = Menge von Methoden
 - Einführung einer Facade Bean für jede Aktivität: Session Bean, die das Remote Interface stellt und intern auf das Local Interface zugreift



EJB 2.0 – Local Interfaces (13)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Neues Problem:
 - Jede neue Aktivität führt
 - zu neuer Session Bean
 - wahrscheinlich zur Verschmelzung von Clustern (Schnittmenge der benötigten Entity Beans)
 - zu riesigen Clustern (z.B. Session Bean greift auf mehrere Cluster-Beans zu, die in unterschiedlichen Clustern stecken)



EJB 2.0 – Local Interfaces (14)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Weitere Nachteile dieses Ansatzes
 - Alle Session Beans könnten auf alle Entity-Beans zugreifen: „**komponentenbasierter Spaghetti-Code**“
 - Facade Session Beans sind extra konstruiert
 - nicht wiederverwendbar
 - Komponenten ?
 - Gesteigerte Komplexität in Entwicklung und Wartung



EJB 2.0 – Local Interfaces (15)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Letzter Versuch
 - Kontra Bildung eines Riesencusters
 - Kommunikation zwischen Facade Beans unterschiedlicher Cluster
 - Aber Problem:
 - Zyklen bei Session Beans sind verboten
 - Wiederum steigende Komplexität bei den Session Beans



EJB 2.0 – Local Interfaces (16)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Local Interfaces nur unter bestimmten Gegebenheiten eine gute Lösung
- Alternativlösung entwickelt ... siehe Literatur: H. A. Schmid



EJB 2.0 – CMP (1)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Container Managed Persistence → CMP EntityBeans
- Seit EJB 1.1
- In EJB 1.1 Problem der schlechten Portierbarkeit



EJB 2.0 – CMP (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Spezifikation 2.0
 - Bean Klassen und Persistenz genügend spezifiziert
 - Portierung von Sourcecode leicht
 - Generator (kommt mit Container) erzeugt aus Deployment Descriptor- Beschreibung Container-spezifische Hilfsklasse



EJB 2.0 – CMP (3)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- Weitere Vorteile
 - Generatorwerkzeug erzeugt Unabhängigkeit von der Persistenzschicht
 - Generierung vermindert Testbedarf



EJB 2.0 – CMP/CMR (1)

Einführung
J2EE
EJB
EJB 2.0
Literatur

- CMR =
Container Managed Relations
- Verwaltung von Relationen zwischen Beans durch Container



EJB 2.0 – CMP/CMR (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- 1.x

- Abfrage der Fremdschlüsselbeziehungen durch Find-Methoden

- 2.0

- Beziehungen definiert im Deployment Descriptor



EJB 2.0 – CMP/CMR (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Abstrakte get/set Methoden in den Beans
- Implementierung durch Generator (beim Container)



EJB 2.0 – CMP/CMR (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Kardinalitäten: 1:1, 1:M, M:N
- Local Interface, `java.util.Set` bzw. `java.util.Collection` als Parameter in den `get()` und `set()` Methoden
- Nur einsetzbar für CMP Entity Beans



EJB 2.0 – CMP/CMR (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- M:N Relationen

- Müssen in relationalen Datenbanken mit dritter Verbindungstabelle dargestellt werden
- Neu: Nur Angabe im Deployment Descriptor, vorher explizite Bean-Modellierung



EJB 2.0 – CMP/CMR (6)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- 1:M bzw. 1:1 Beziehungen
 - erlauben das „Cascade Delete“, wenn das löschende Element einzeln ist (die 1-Seite)
 - Vergleiche Datenbanken
 - Spezifikation 2.0 bietet Technik an:
Aufruf von `ejbRemove()` am anderen Ende der Beziehung



EJB 2.0 – CMP/CMR (7)

Einführung
J2EE

EJB

EJB 2.0

Literatur

- Container ist für referentielle Integrität zuständig
- Grosse Rolle von CMR bei Anfragen per EJB QL (folgt)



EJB 2.0 – CMP – Dependent Value Classes

Einführung

J2EE

EJB

EJB 2.0

Literatur

- instanziiierbare und serialisierbare Java Klassen
- werden serialisiert in der Datenbank gespeichert
- Nachteil: Schlechte Performance bei Anfragen



EJB 2.0 – EJB QL (1)

Einführung
J2EE

EJB

EJB 2.0

Literatur

Enterprise **J**ava**B**eans **Q**uery **L**anguage

- Ist eine „query specification language“
- Benutzt zum Auffinden von Daten
- Keine Manipulation ! (vgl. SQL)



EJB 2.0 – EJB QL (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Bis Spezifikation 1.1:
 - Eingebettete SQL-Anfragen im Sourcecode
 - Verlust der Unabhängigkeit gegenüber der Persistenzschicht
- EJB QL arbeitet als Anfragesprache auf dem abstrakten Schema der Bean im Deployment Descriptor



EJB 2.0 - EJB QL (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Suchfunktionalität
 - Jeweils Signaturen im Quellcode:
 - Find-Methoden im Home Interface
 - Select-Methoden in der Bean-Klasse
 - Signaturen wiederholen im Deployment Descriptor
 - Ebenfalls im DD: Anfrage
 - (angelehnt an SQL92)



EJB 2.0 – EJB QL (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- `Select_clause`
`from_clause`
`[where_clause]`
- Methodenparameter: ?1, ?2,...



EJB 2.0 – EJB QL (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- TRUE, FALSE
- +, -, *, /
- =, >, >=, <, <=, <>
- NOT, AND, OR
- BETWEEN, IN, LIKE, NULL, EMPTY, MEMBER
- CONCAT, SUBSTRING, LENGTH,...
-



EJB 2.0 – EJB QL (6)

Einführung
J2EE
EJB
EJB 2.0
Literatur

Beispiel für find-Methode
(im Home Interface folgende Signatur):

```
public Collection findXXX(int id, Date maxDate)  
    throws FinderException
```





EJB 2.0 – EJB QL (7)

Einführung
J2EE
EJB
EJB 2.0
Literatur

```
<query>
  <query-method>
    <method-name>findXXX</method-name>
    <method-params>
      <method-param>int</method-param>
      <method-param>java.sql.Date</method-param>
    </method-params>
  </query-method>
  <ejb-ql> <![CDATA[SELECT OBJECT(v) FROM ABCBean v
    WHERE v.id = ?1
    AND v.gueltigbis = ?2 ]]>
  </ejb-ql>
</query>
```



EJB 2.0 – EJB QL (8)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Select Methoden
 - Neu in 2.0
 - Abstrakt in der Bean mit Prefix:
- „ejbSelect...”
 - im Gegensatz zu ejbFind() nicht nach außen sichtbar
 - Umsetzung wie bei ejbFind()



EJB 2.0 – Home Methods (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Zusätzliche Methoden
 - Definiert im Home-Interface
 - Neben create-, remove-, find-
 - Beziehen sich auf eine Menge von Instanzen
 - Wie static-Methoden in Java (Klassenmethoden)



EJB 2.0 – Home Methods (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Implementierung
 - „ejbHome“-Präfix für den Methodennamen
 - Public – muss sein
 - Static – darf nicht sein



EJB 2.0 – Home Methods (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Beispiel: Remote-Interface

```
public int berechneXY() throws RemoteException;
```

- Implementierung

```
public int.ejbHomeBerechneXY() {  
    ....  
}
```



EJB 2.0 – Home Methods (4)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- **WICHTIG**

- Keine Instanzmethoden aus einer ejbHome-Implementierung aufrufen (vgl. static, wo der Compiler protestieren könnte)



EJB 2.0 – Home Methods (5)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Lebenszyklus einer Entity-Bean
 - *does not exist* (keine Methodenaufrufe möglich)
 - *pooled* ← (alle Instanzen noch gleich: Zustand für den Aufruf von ejbSelect- und Home-Methoden)
 - *ready* (ejbCreate, usw. sorgen für Attributbelegung einzelner Instanz, jetzt Component-Interface-Methoden aufrufbar)



EJB 2.0 – Run-As-Security (1)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Erweiterung des bestehenden Aufruf-Berechtigungskonzept
- Definieren von „Rollen“, die Aufrufer innehaben kann
- Rollendefinition im Deployment-Descriptor
- Weiterreichen des *Security-Contextes* bei inneren Methodenaufrufen, welcher die Benutzeridentität beinhaltet



EJB 2.0 – Run-As-Security (2)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Run-As steht für:
 - Zuweisung einer Identität an eine Bean
 - 1) Nutzer ruft Methode X der Bean auf
 - 2) Überprüfung, ob er dies darf
 - 3) Bean hat Identität „admin“ (z.B.)
 - 4) Alle Methodenaufrufe in X werden gegen „admin“ auf Berechtigung geprüft.



EJB 2.0 – Run-As-Security (3)

Einführung

J2EE

EJB

EJB 2.0

Literatur

- Run-As verhindert Mehrfach-implementierung von Methoden.



Literatur

Einführung

J2EE

EJB

EJB 2.0

Literatur

A. Eberhart, S. Fischer (2001)

Java-Bausteine für E-Commerce-Anwendungen

Hanser

G. Seshadri (1999)

Enterprise Java Computing – Applications and Architecture

Cambridge University Press

V. Gruhn, M. Schneider (2002)

EJB 2.0 Anwendungen

Addison-Wesley



Literatur

Einführung

J2EE

EJB

EJB 2.0

Literatur

V. Matene, M. Hapner (1998)

Enterprise JavaBeans

<http://www.java.sun.com/products/ejb/docs.html>

V. Matene, M. Hapner (1999)

Enterprise JavaBeans Specification, v1.1

<http://www.java.sun.com/products/ejb/docs.html>

L. DeMichiel, L. Yalçinalp, S. Krishnan (2001)

Enterprise JavaBeans Specification, Version 2.0

<http://www.java.sun.com/products/ejb/docs.html>



Literatur

Einführung

J2EE

EJB

EJB 2.0

Literatur

L. DeMichiel (2002)

Enterprise JavaBeans Specification, Version 2.1 (Public Draft)

<http://java.sun.com/products/ejb/docs.html>

B. Shannon (2001)

Java 2 Platform Enterprise Edition Specification, v1.3

<http://java.sun.com/j2ee/download.html>

M. Pawlan, D. Green, K. Haase, S. Bodoff,

E. Jendrock, B. Stearns (2002)

The J2EE Tutorial

<http://java.sun.com/j2ee/download.html>



Literatur

Einführung

J2EE

EJB

EJB 2.0

Literatur

E. Pelegri-Llopart (2001)

JavaServer Pages Specification Version 1.2

<http://java.sun.com/products/jsp/download.html>

D. Coward (2001)

Java Servlet Specification Version 2.3

<http://java.sun.com/products/servlet/download.html>

H.A. Schmid (2002)

An Evaluation of the Use of Enterprise JavaBeans 2.0

Local Interfaces

University of Applied Sciences, Konstanz



Anbieter von EJB Komponenten

Einführung
J2EE
EJB
EJB 2.0
Literatur

Weitere Informationen unter ...

<http://www.componentsource.com/java>

<http://www.flashline.com/components/>

➔ Marketplace ➔ EJB

<http://industry.java.sun.com/solutions/>

➔ Enterprise JavaBeans Components



Ende der Präsentation

Vielen Dank für Ihre
Aufmerksamkeit!