

Enterprise Java Beans

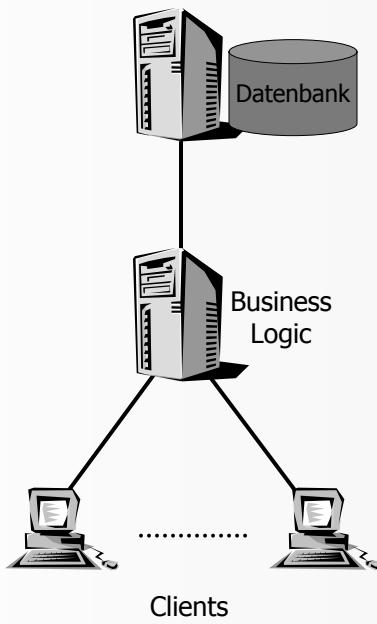
„Die hohe Kunst der
aromatischen Bohnenmischung“

oder

„Replikator: Einmal Kaffee, Brasilia Highland Blend, Heiß“

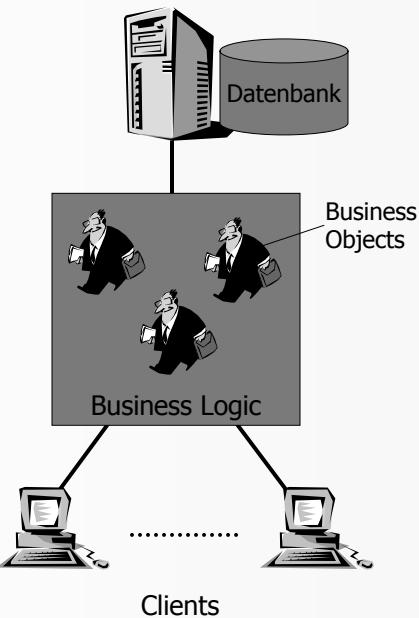
Motivation

- Bean = Komponente
- Zielgruppe
 - Kommerzielle Anwendungen
 - E-Commerce
- Evolution früherer
 - Client/Server-Systeme
 - Transaktionsmonitore
- 3-Tier-Anwendungen
 - Client
 - Business Application
 - Database



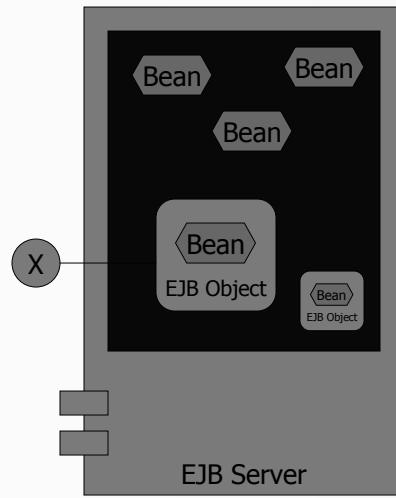
Business Objects

- Komponenten-basierte Anwendung
- Varianten
 - Eigenständige Komponenten
 - Wrapper für Legacy Software
 - Client-Transaktionen
- Laufzeitumgebung
 - EJB Server
 - vergleichbar MTS bei COM+



Aufbau des EJB-Server

- Laufzeitumgebung
 - Namensverwaltung
 - Anbindung an DB
 - Nachrichtenkommunikation
- EJB Object = Bean Wrapper
 - Indirektionsstufe
 - Zugriff über Reflection
- Beantypen
 - Entity Bean
 - Zustand, Persistent
 - Session Bean
 - Transient
 - Stateless, Stateful



Aufgaben des Servers

- Resource Management
 - Instance Pooling: Beans vorinstanziert
 - Instance Swapping: Wechsel der EJB-Object-Bindung
 - Activation: Stilllegen und Reaktivieren von Beans
- Primary Services
 - Concurrency: Single-Threaded Bean-Implementierung
 - Transactions: ACID-Prinzip
 - Persistence: Container-Managed, Bean-Managed
 - Distribution: RMI over JRMP oder RMI-IIOP
 - Naming: JNDI unterstützt LDAP, NIS+, DNS, ...
 - Security: Methodenbasierte Zugriffskontrolle

Enterprise Java Beans

Entity Bean

Bean-Struktur

- Remote Interface
 - Die eigentlichen Zugriffsfunktionen der Bean
 - Setters und Getters :-)
 - `public interface X extends javax.ejb.EJBObject { ... };`
- Home Interface
 - Bean-Erzeugung: `public X create (...)`
 - Beans wiederfinden: `public X findByPrimaryKey (key)`
 - `public interface XHome extends java.ejb.EJBHome { ... };`
- Primary Key
 - Speichern und Auffinden der Bean in Datenbank
 - `public class XPK implements java.io.Serializable { ... };`
- Bean Class
 - `public class XBean implements javax.ejb.EntityBean { ... };`

EntityBean: Funktionen

- ejbActivate
 - EJBObject-Bean-Bindung wiederhergestellt
- ejbPassivate
 - EJBObject-Bean-Bindung wird abgebaut
- ejbLoad
 - Bean wurde geladen
- ejbStore
 - Bean wird gespeichert
- ejbRemove
- setEntityContext
unsetEntityContext
 - Informationen über EJBObject, Client, PK

Exkurs: Reflection

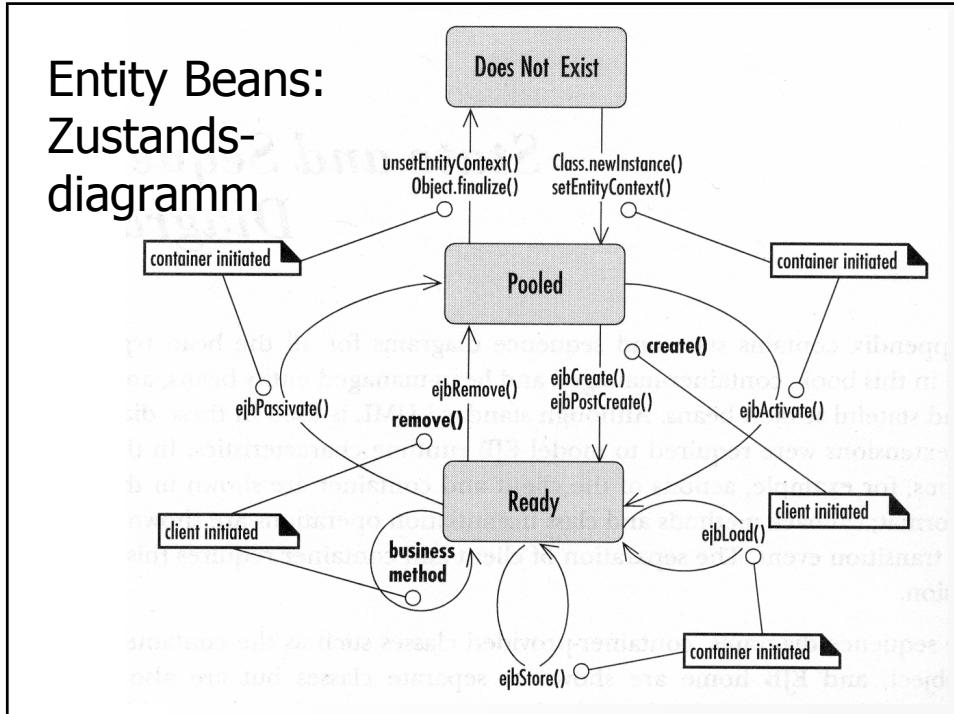
- Bean-Klasse implementiert weder Remote- noch Home-Interface
 - keine direkte Bindung zum Client vorhanden
- Bean-Klasse muß aber Methoden mit identischer Signatur enthalten
- EJB-Object greift über Reflection-Mechanismus darauf zu
 - vgl. JavaBeans
- Beispiel „ShowClass“ aus
 - Java in a Nutshell
 - D. Flanagan
 - O'Reilly, 2. Auflage



Deployment Descriptor

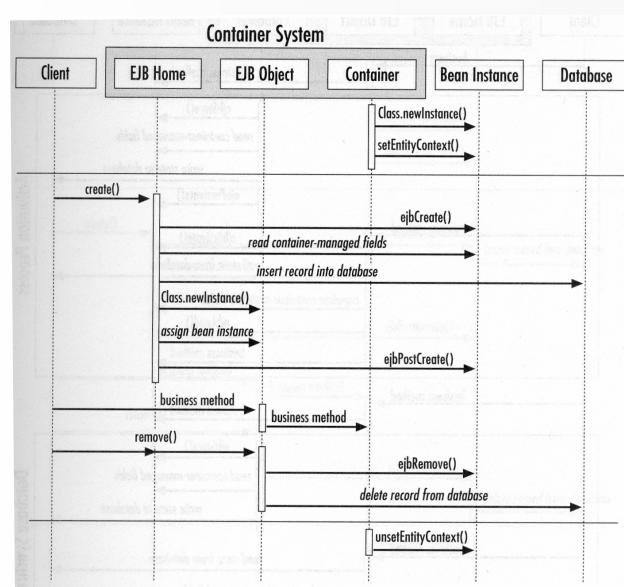
- XML-Dokument (EJB Version 1.1)
 - Bean-Beschreibung
 - Namen der Interfaces
 - Klassename
 - Persistenztyp (Container oder Bean)
 - Welche Bean-Attribute werden persistent gespeichert
 - Zugriffsrollen von Clients beim Bean-Zugriff
 - Zugriffskontrolle auf Methoden
 - Welche Rollen dürfen zugreifen
 - Transaktionen
- Bean = jar-File bestehend aus XML-DD und *.class Dateien

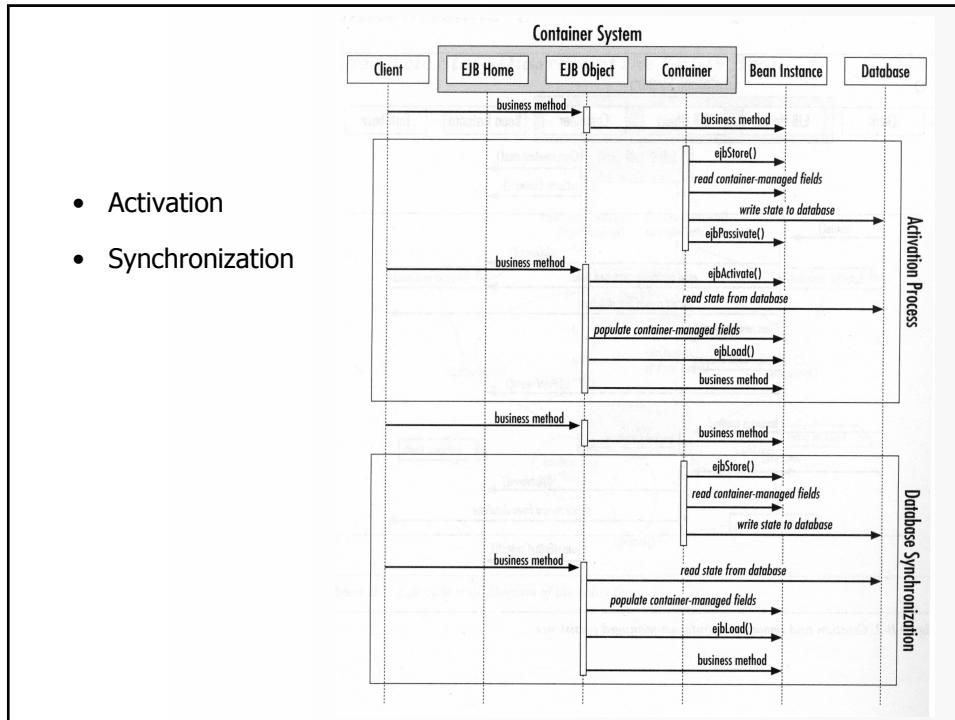
Entity Beans: Zustandsdiagramm



Container-Managed Persistence

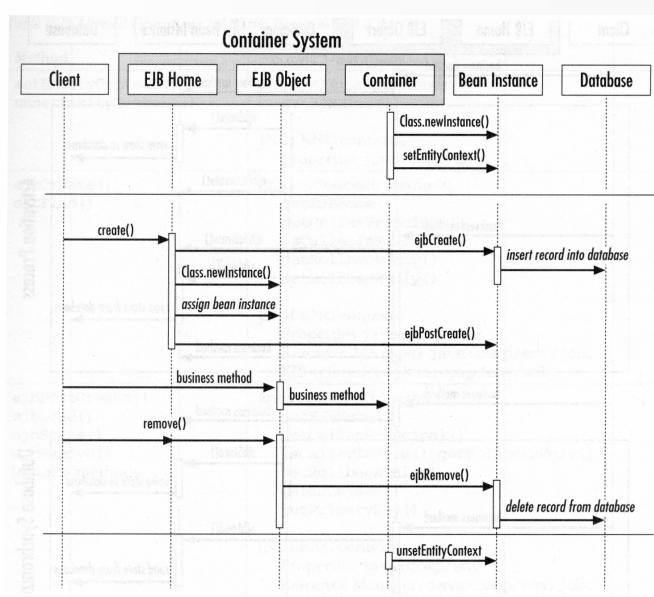
- Erzeugen
- Löschen

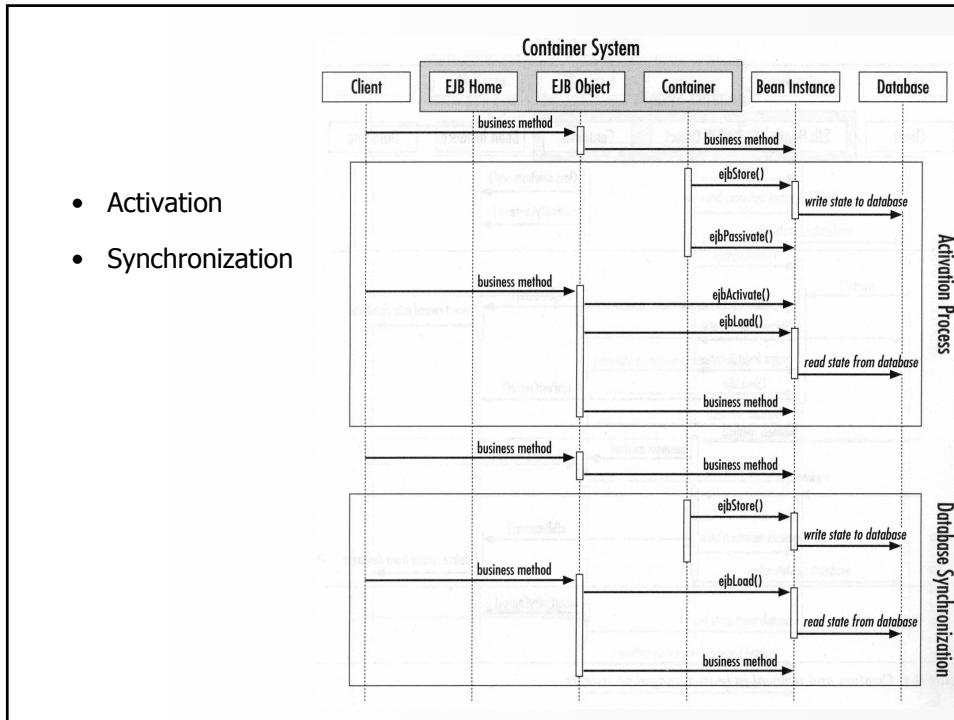




Bean-Managed Persistence

- Erzeugen
- Löschen





Enterprise Java Beans

Session Bean

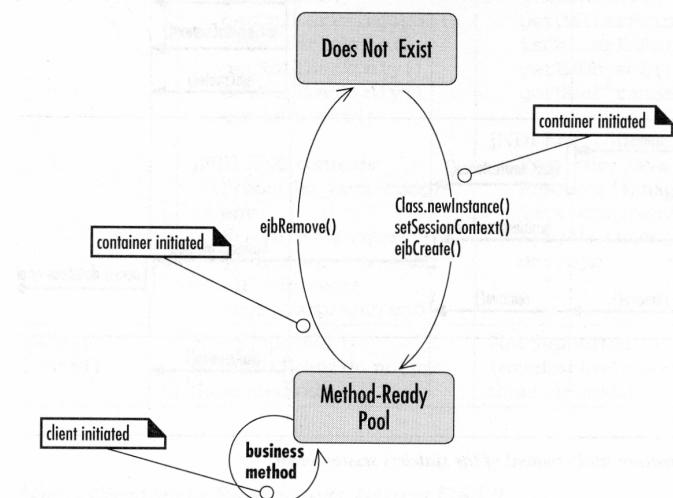
Session Beans

- Session Beans besitzen keinen persistent gespeicherten Zustand
- Zwei Varianten
- Stateless Session Bean
 - Kein sichtbarer Zustand zwischen Methodenaufrufen
 - Pooling und Swapping möglich
- Stateful Session Bean
 - Feste Bindung zu einem Client
 - Conversational State zwischen Methodenaufrufen
 - Swapping nicht möglich

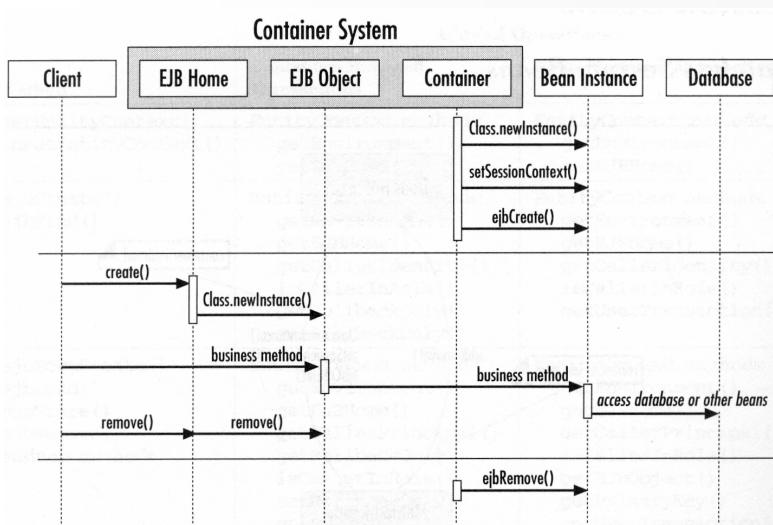
Bestandteile

- Remote Interface
 - Eigentlichen Interaktionsfunktionen
- Home Interface
 - Nur Erzeugung
 - Kein Auffinden über Primary Keys
- Bean-Klasse
 - Erweitert SessionBean
- Deployment Descriptor
 - XML-Dokument
 - u.a. Lokalisierungsinformation benötigter Beans
 - Zugriffskontrolle

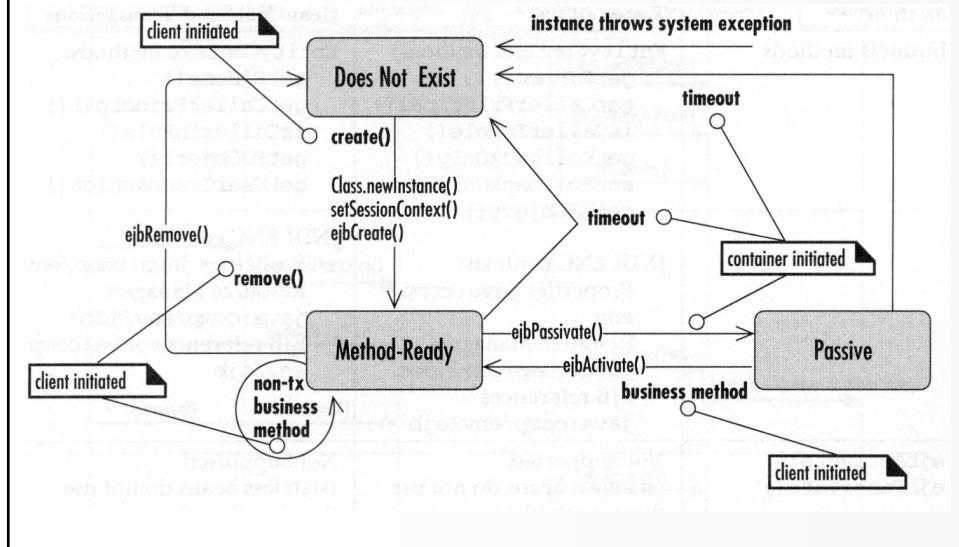
Session Bean: Stateless



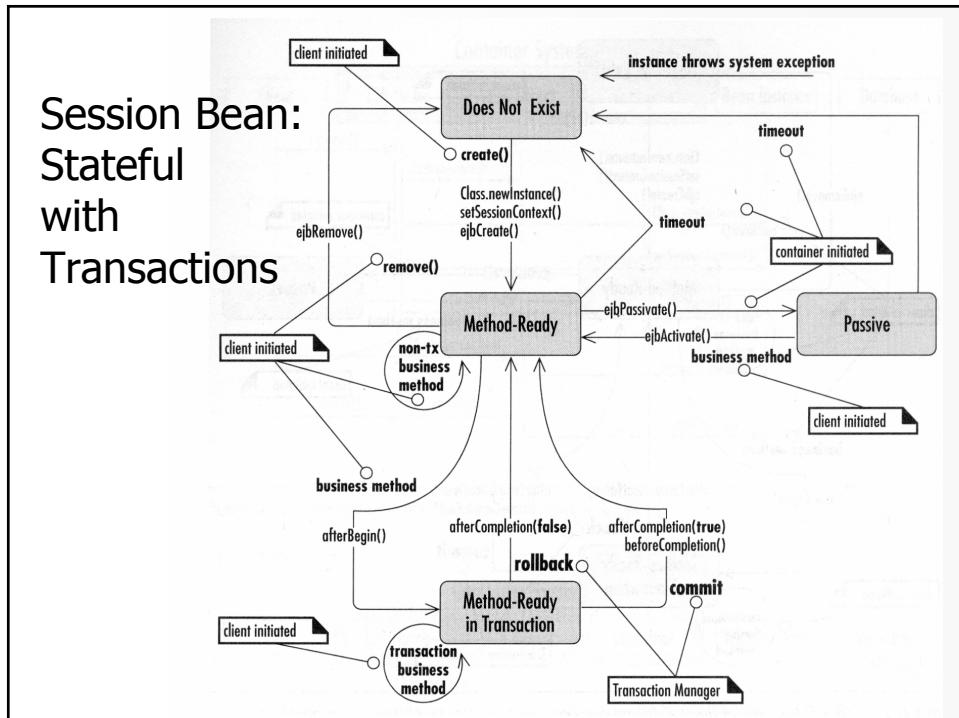
Stateless: `create()` und `remove()`



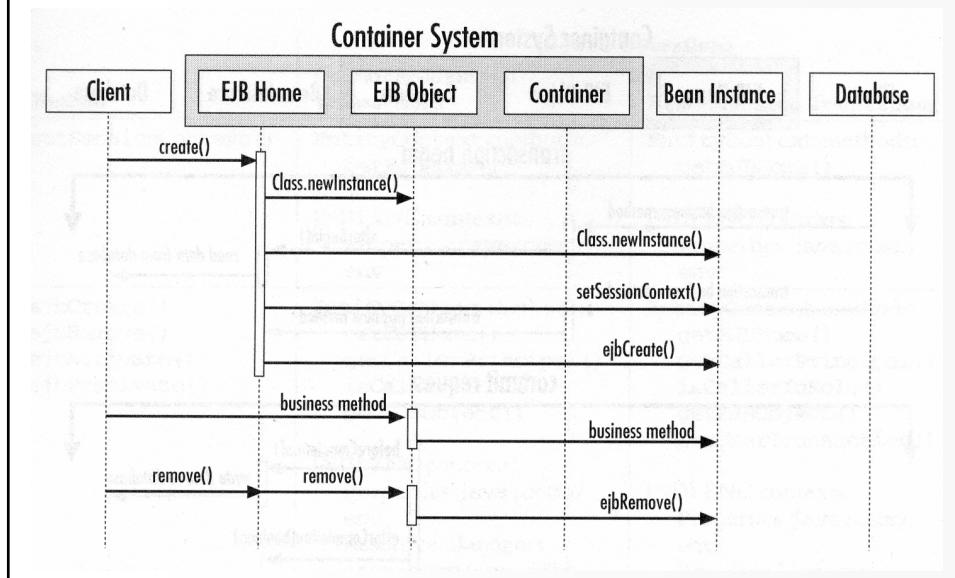
Session Bean: Stateful



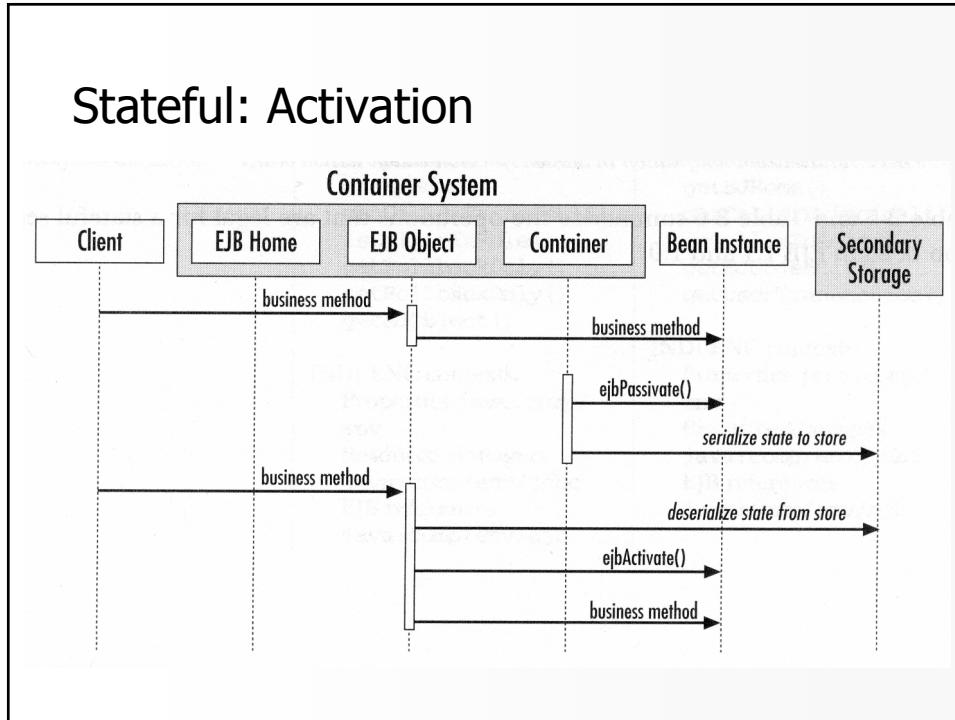
Session Bean: Stateful with Transactions



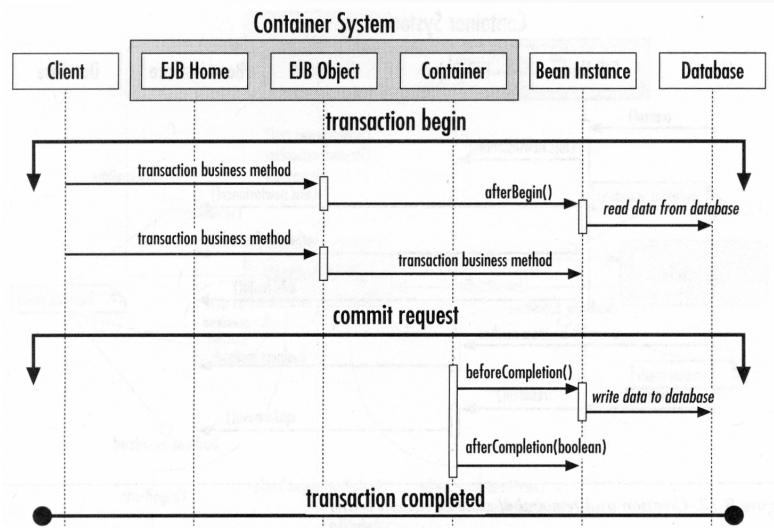
Stateful: `create()` und `remove()`



Stateful: Activation



Stateful: Transaction Notification



Literatur

- Richard Monson-Haefel
Enterprise Java Beans
O'Reilly, 2. Auflage, 2000
 - Quelle der gezeigten Zustands- und Interaktionsdiagramme