

presentation.



# Avantis Unisuite for EJB objekt-relacionales Mapping für EJBs

Ralf Schippert  
ralf.schippert@avantis.de

Avantis GmbH  
Max-Lang-Straße 24  
D-70771 Leinfelden  
fon +49 (0)711.903748-0  
fax +49 (0)711.903748-10  
web [www.avantis.de](http://www.avantis.de)

v 2.2



# Inhalt

- Object-relationales Mapping
  - Avantis Unisuite
  - UML-Bridge for EJB
    - Modellierung
    - Import
    - Prüfung
    - Generierung
  - Persistency-Bridge for EJB
  - EJB Komponenten mit dependent Objects
  - Features

# Ausgangssituation

Anforderungen an heutige Softwareprojekte für Internet und e-Commerce

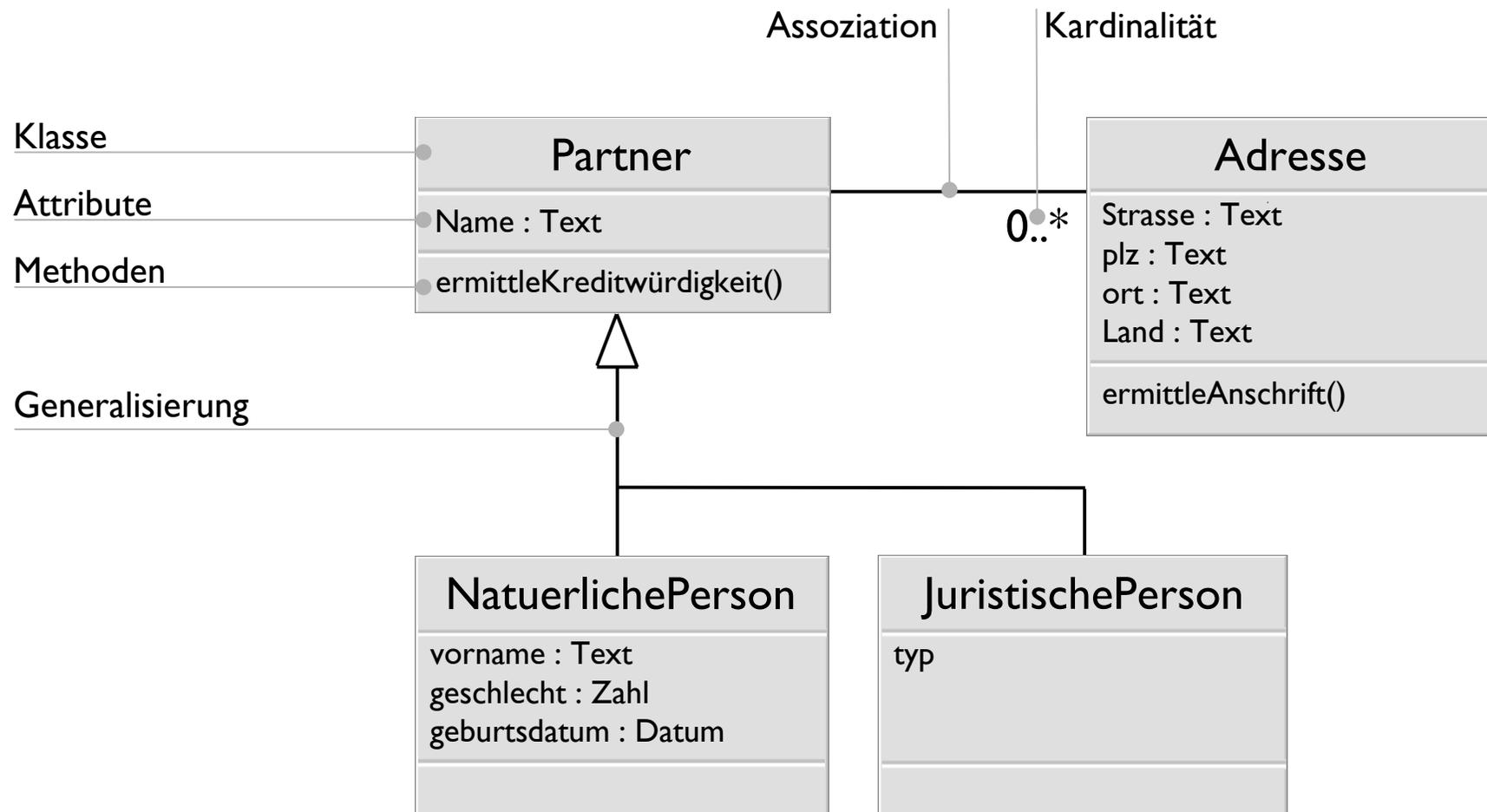
- kurze Entwicklungszyklen
- moderne, mehrschichtige Architekturen
- moderne Modellierungstechniken
- moderne Programmiersprachen

objekt-relationales  
Mapping-Werkzeug

Randbedingungen in konkreten Projekten

- Kosten- und Termindruck
- wenig Erfahrung mit neuen Technologien
- bestehende relationale Datenbestände

# OR-Mapping – Objektorientierung



# OR-Mapping – Datenmodellierung

Relation  
(Tabelle)

Attribut/Spalte  
(Merkmal, Eigenschaft)

Tupel  
(Satz, Record)

Attributwert  
(Ausprägung)

PERSON			
NAME	VNAME	GEBDAT	GESCHL
Duck	Donald	01.05.1956	0
Duck	Daisy	05.08.1961	1
Duck	Dagobert	13.11.1938	0
Düsentrieb	Daniel	25.01.1954	0
...			
...			
...			

# OR-Mapping – Anforderungen

## Anforderungen an objekt-relationales Mapping

- Klassen, Attribute, Assoziationen und Generalisierung abbilden
- Kein SQL in fachlichen Klassen
- Über Assoziationen des Modells navigieren
- Objektmengen über Anfragen ermitteln
- Identität von Objekten gewährleisten
- Transaktionen auf Objektebene
- Verwaltung der Nebenläufigkeit
- Integritäts- und Plausibilitätsprüfungen

# Inhalt

- Objekt-relationales Mapping
- **Avantis Unisuite**
- UML-Bridge for EJB
  - Modellierung
  - Import
  - Prüfung
  - Generierung
- Persistency-Bridge for EJB
- EJB Komponenten mit dependent Objects
- Features

# Avantis Unisuite

Avantis Unisuite besteht aus zwei Teilprodukten:

## UML-Bridge

Programmcode-Generator

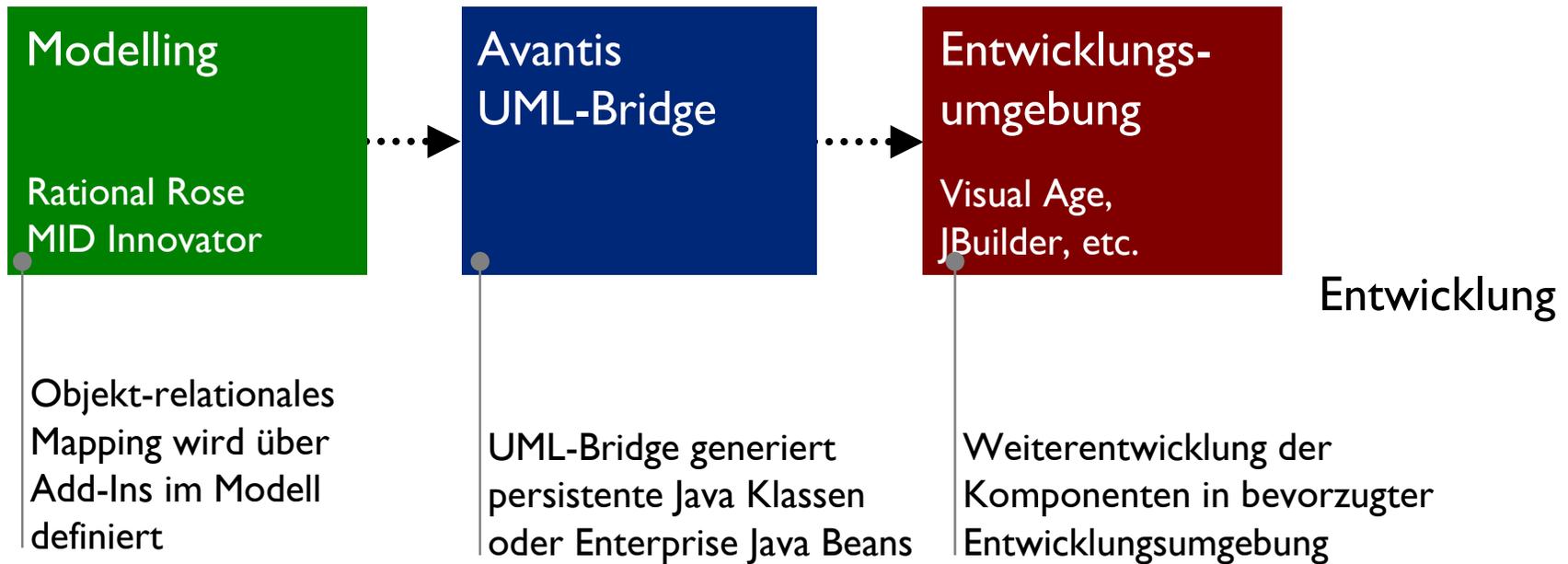
Generiert Softwarekomponenten für die  
Persistency-Bridge direkt aus dem UML-Modell des  
Modellierungswerkzeuges

## Persistency-Bridge

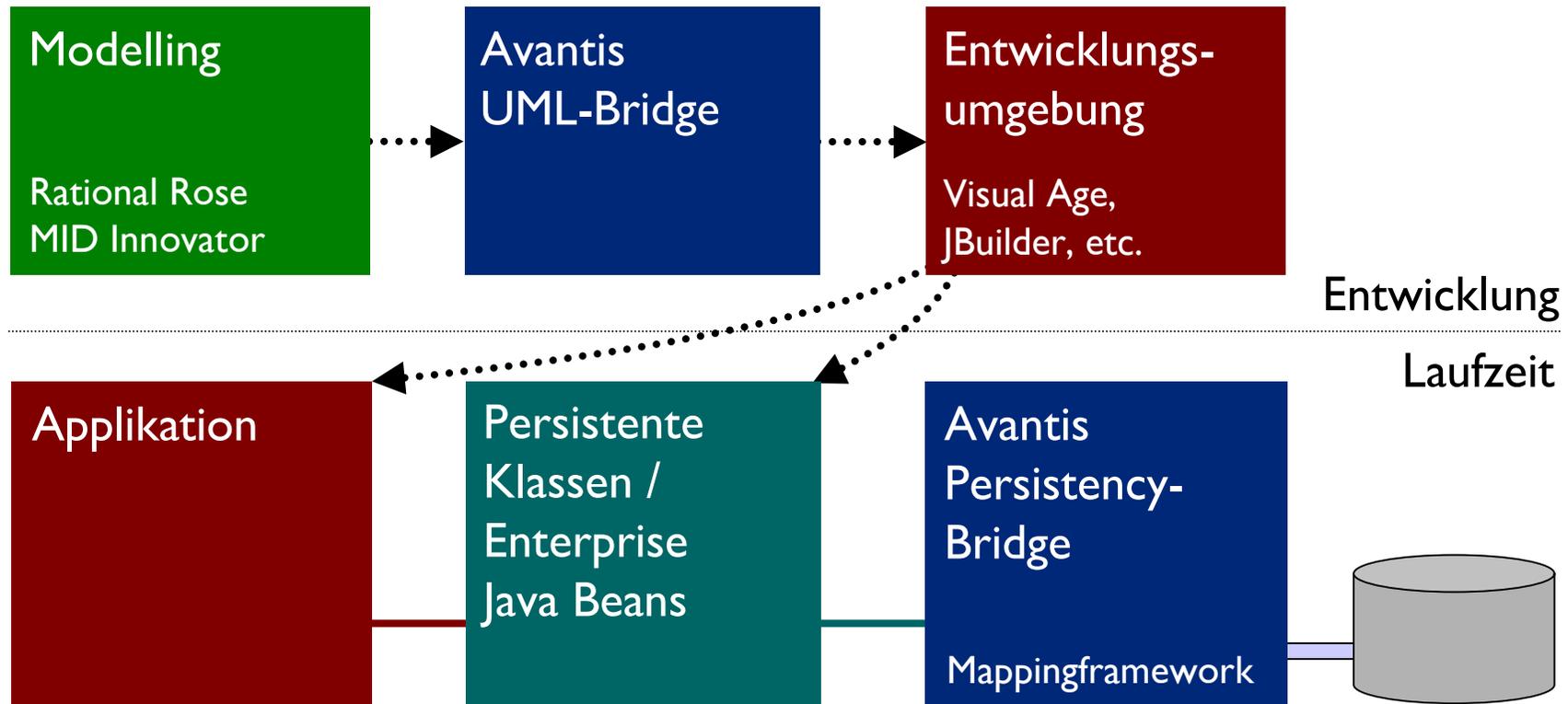
objekt-relationales Mappingwerkzeug

Bindet objektorientierte Anwendungsprogramme  
an neue oder bestehende Datenbanksysteme

# Avantis Unisuite



# Avantis Unisuite

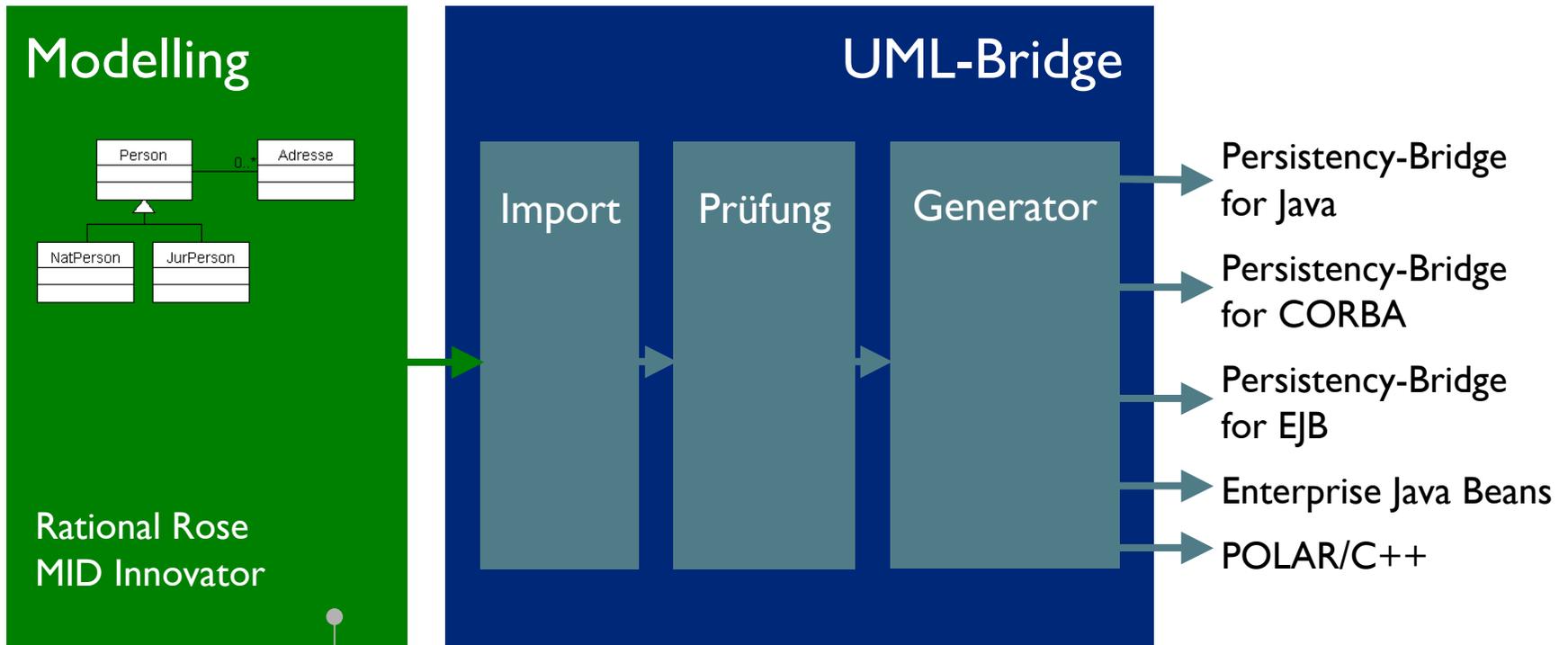


Infrastruktur: Applikationsserver, Object Request Broker, etc.

# Inhalt

- Objekt-relationales Mapping
- Avantis Unisuite
- **UML-Bridge for EJB**
  - Modellierung
  - Import
  - Prüfung
  - Generierung
- Persistency-Bridge for EJB
- EJB Komponenten mit dependent Objects
- Features

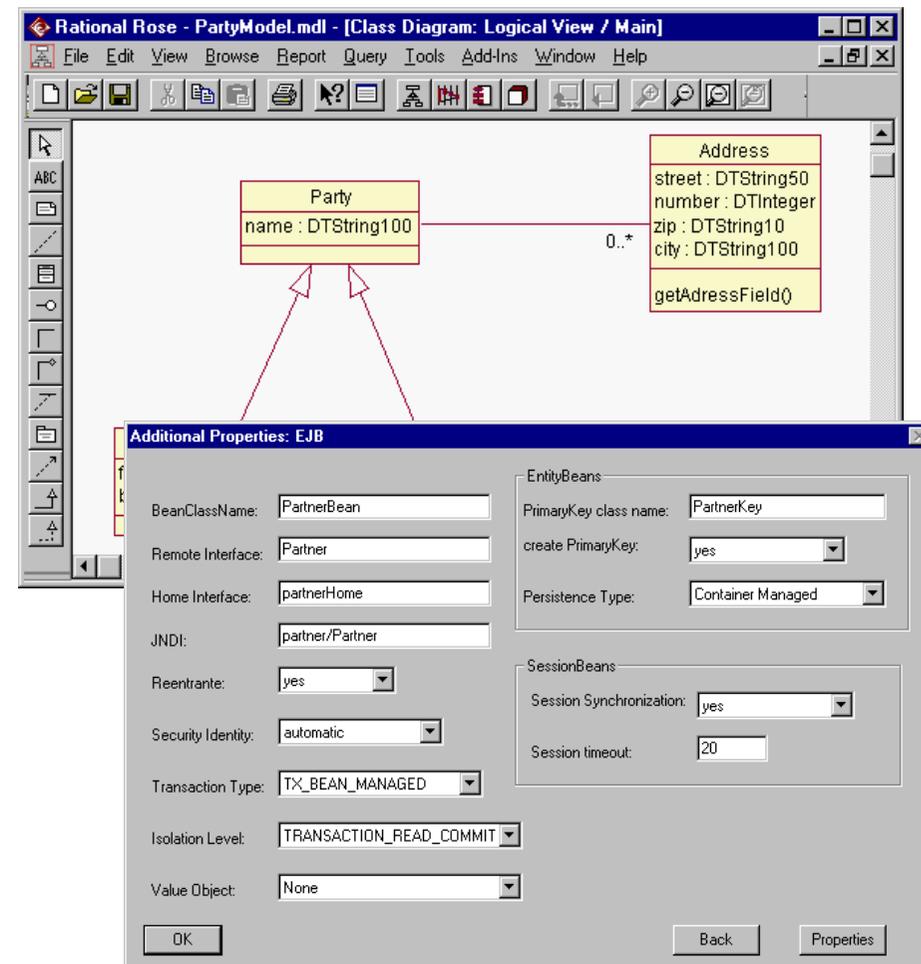
# UML-Bridge



OO-ER Mapping  
wird bereits im  
Modell festgelegt

# UML-Bridge – Modellierung

- Der Entwickler entwirft die Softwarekomponenten visuell im Modellierungswerkzeug.
- Avantis Add-Ins erweitern die Modellierungswerkzeuge, so dass alle Informationen für die Codegenerierung im Modell erfasst werden können.

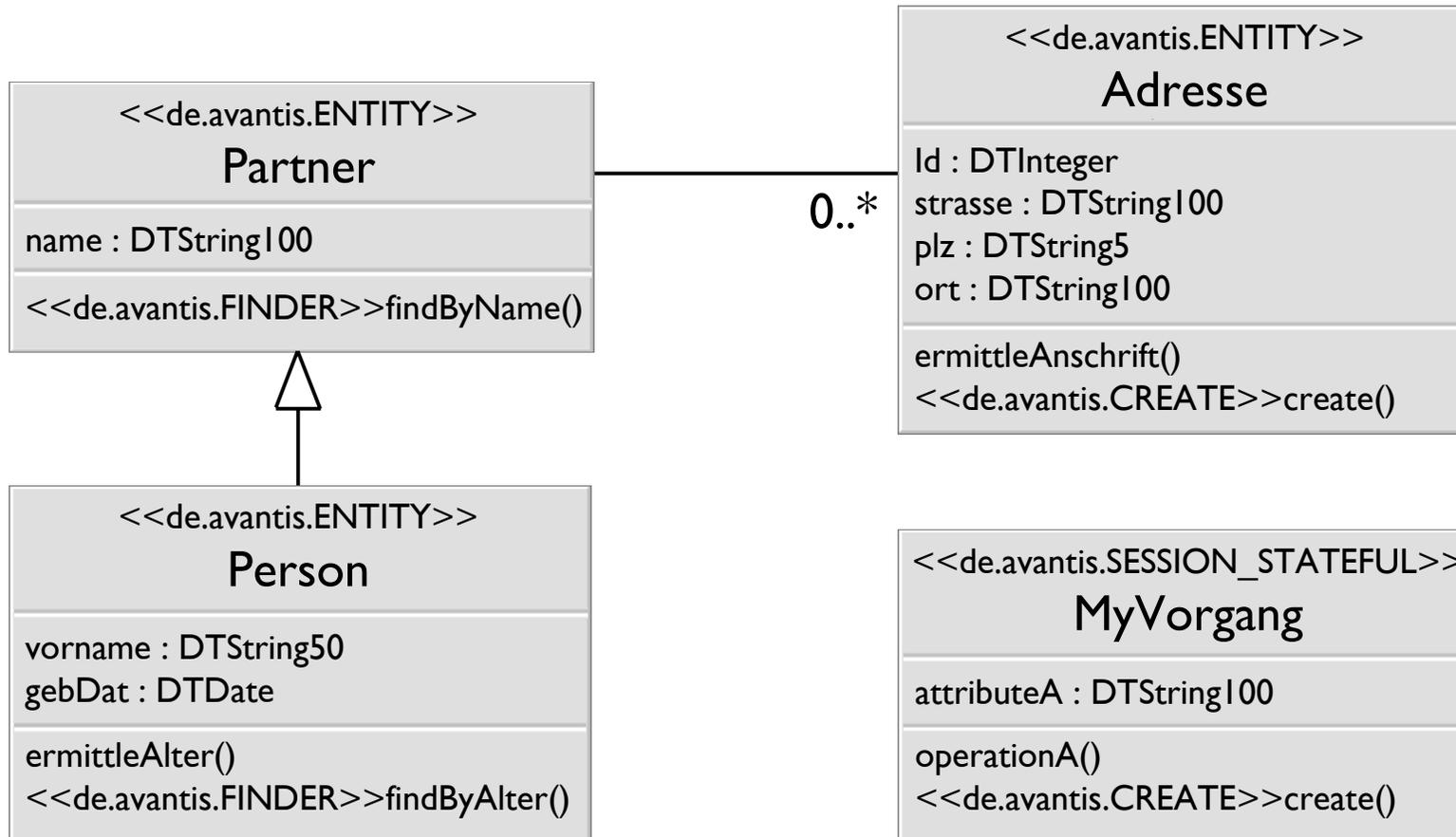


# UML-Bridge for EJB – Modellierung

- EJBs werden in Form von Klassen im UML-Modell beschrieben
- Stereotypen legen den EJB-Typ fest
  - Stateless Session-Bean
  - Stateful Session-Bean
  - Entity-Bean
- finder- und create-Methoden können modelliert werden
- Beziehungen zwischen Enterprise Java Beans werden vollständig unterstützt
- Über Add-Ins werden EJB-spezifische Informationen erfasst

# UML-Bridge for EJB – Modellierung

## Klassenmodell



# UML-Bridge for EJB – Modellierung

## EJB-Spezifische Eingaben

**Additional Properties: EJB**

BeanClassName:	<input type="text" value="PartnerBean"/>	EntityBeans	PrimaryKey class name:	<input type="text" value="PartnerKey"/>
Remote Interface:	<input type="text" value="Partner"/>	create PrimaryKey:	<input type="text" value="yes"/>	
Home Interface:	<input type="text" value="partnerHome"/>	Persistence Type:	<input type="text" value="Container Managed"/>	
JNDI:	<input type="text" value="partner/Partner"/>	SessionBeans	Session Synchronization:	<input type="text" value="yes"/>
Reentrante:	<input type="text" value="yes"/>	Session timeout:	<input type="text" value="20"/>	
Security Identity:	<input type="text" value="automatic"/>			
Transaction Type:	<input type="text" value="TX_BEAN_MANAGED"/>			
Isolation Level:	<input type="text" value="TRANSACTION_READ_COMMIT"/>			
Value Object:	<input type="text" value="None"/>			

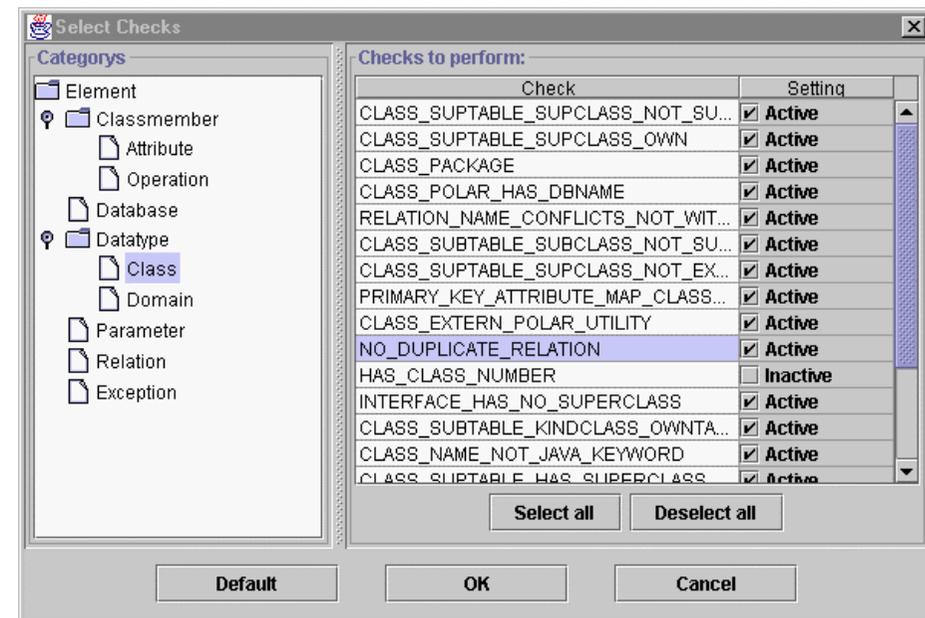
OK Back Properties

# UML-Bridge – Import

- Die UML-Bridge importiert das Objektmodell. Alle für die Generierung benötigten Informationen stehen damit zur Verfügung.
- Nach dem Import vervollständigt die UML-Bridge das Modell, dabei werden alle im UML-Modell nicht angegebenen Einstellungen mit Standardwerten gesetzt. Die Standardwerte sind frei konfigurierbar.

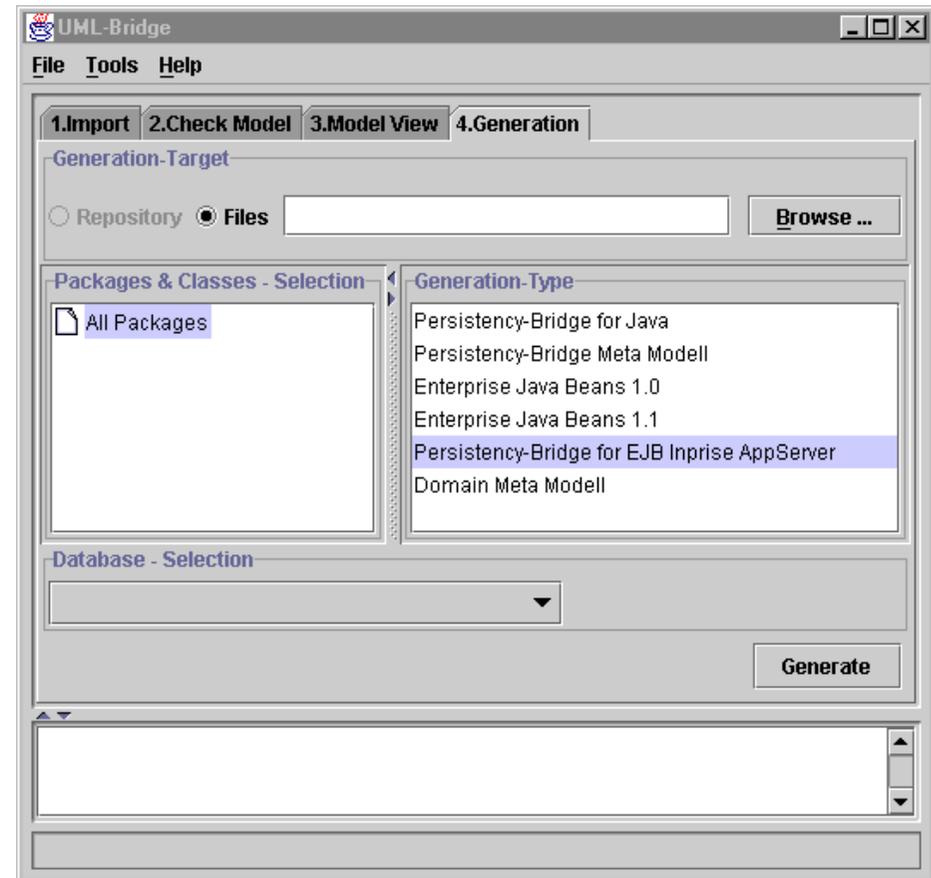
# UML-Bridge – Prüfung

- Nach dem Einlesen des Modells kann der Entwickler verschiedene Prüfungen durchführen lassen.
- Diese Prüfungen stellen ein in sich konsistentes Modell sicher und vermeiden unnötige Generierungsläufe.
- Die zahlreichen Prüffälle können wahlweise aktivieren oder deaktiviert und durch eigenentwickelte ergänzt werden.



# UML-Bridge – Generierung

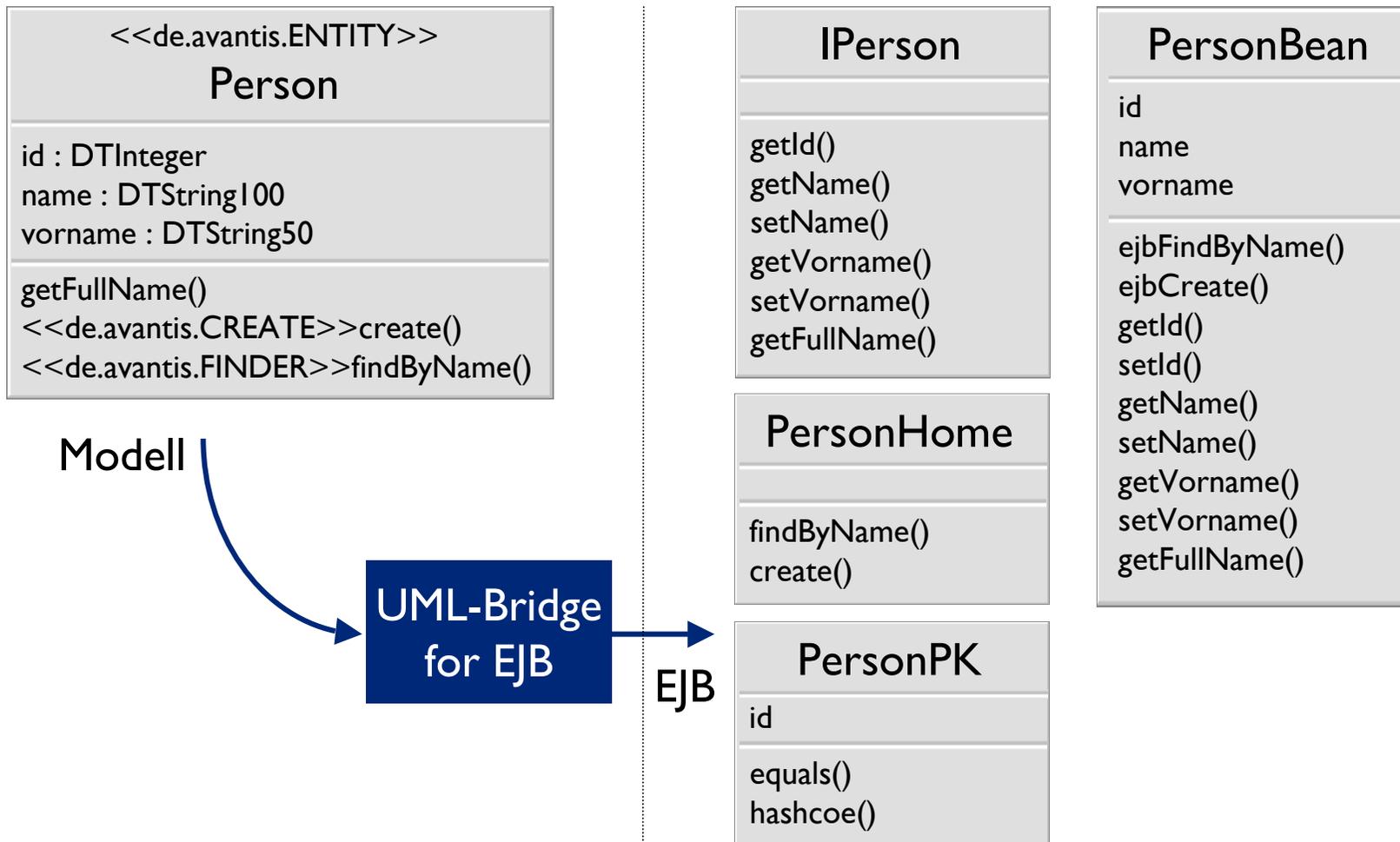
- Liegt ein fehlerfreies Modell vor, generiert die UML-Bridge Java-Objekte, CORBA-Objekte mit IDL-Beschreibung oder Enterprise Java Beans.
- Änderungen können jederzeit visuell im UML-Modell vorgenommen werden, ohne dass der eigenentwickelte Programmcode bei der nächsten Generierung überschrieben wird.



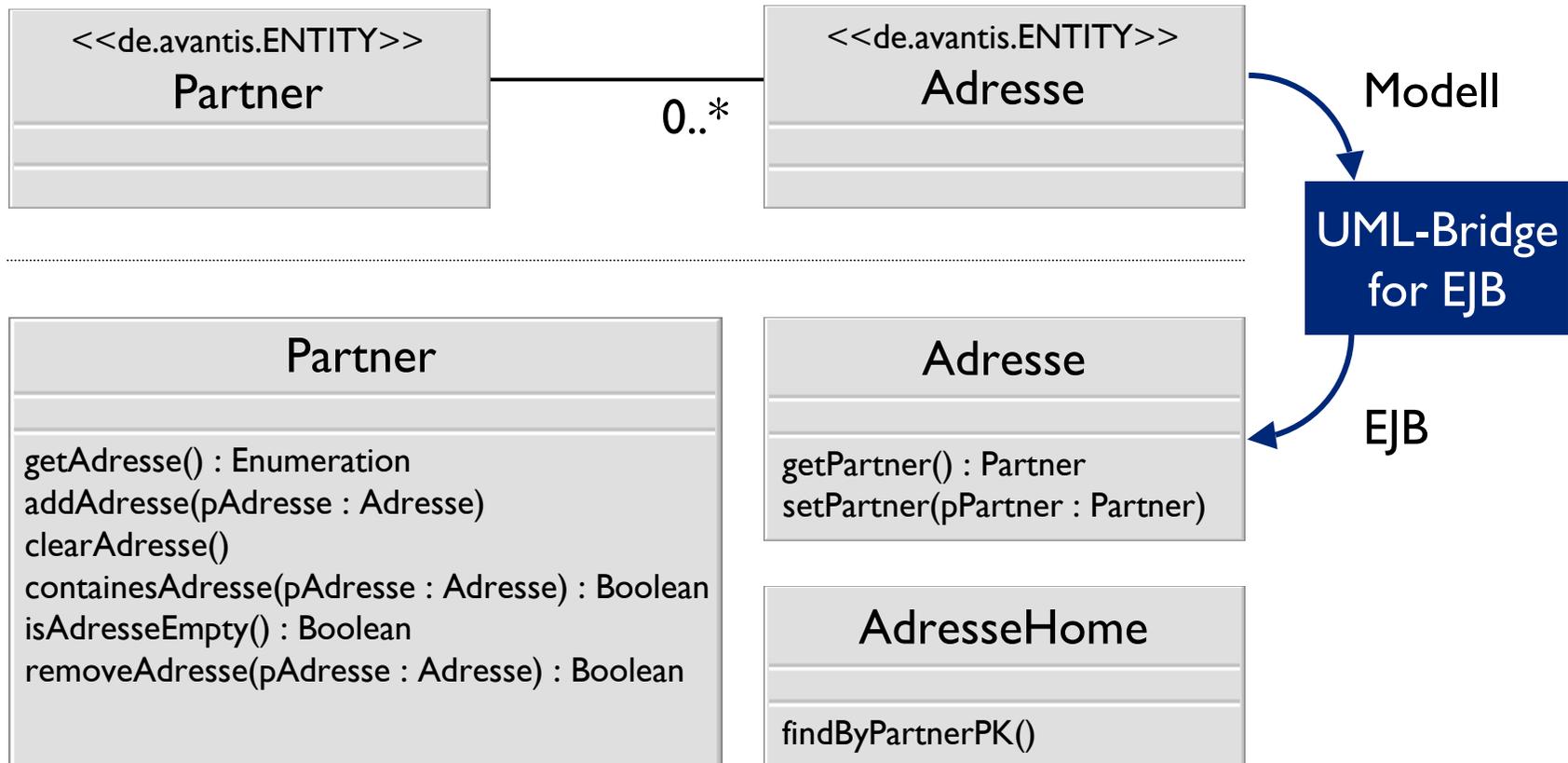
# UML-Bridge for EJB – Generierung

- UML-Bridge for EJB generiert
  - Remote-Interfaces
  - Implementierungsklassen
  - Home-Interfaces
  - Primärschlüsselklassen
  - Deployment-Deskriptoren

# Generierung – Entity-Bean



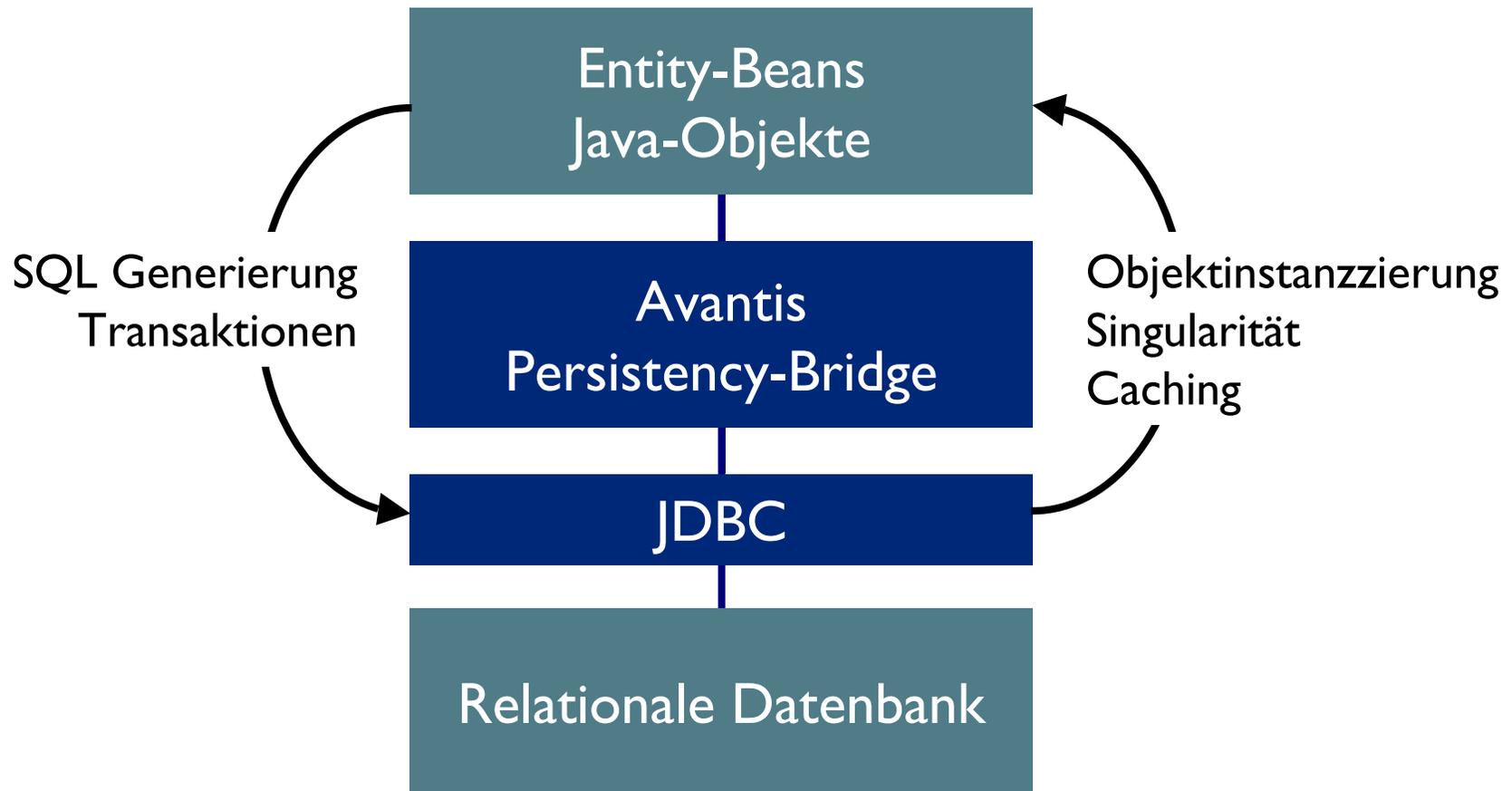
# Generierung – Beziehungen



# Inhalt

- Objekt-relationales Mapping
- Avantis Unisuite
- UML-Bridge for EJB
  - Modellierung
  - Import
  - Prüfung
  - Generierung
- **Persistency-Bridge for EJB**
- EJB Komponenten mit dependent Objects
- Features

# Persistence-Bridge



# Persistence-Bridge – Mapping

- Datentypen
  - alle Standard-Datentypen relationaler Datenbanken
- Beziehungen zwischen Klassen
  - 1:1, 1:n
- Vererbung
  - Datenhaltung in einer eigener Tabelle
  - Datenhaltung in der Tabelle der Unterklasse
  - Datenhaltung in der Tabelle der Oberklasse
- Technische Schlüssel über Surrogate oder fachliche Schlüssel

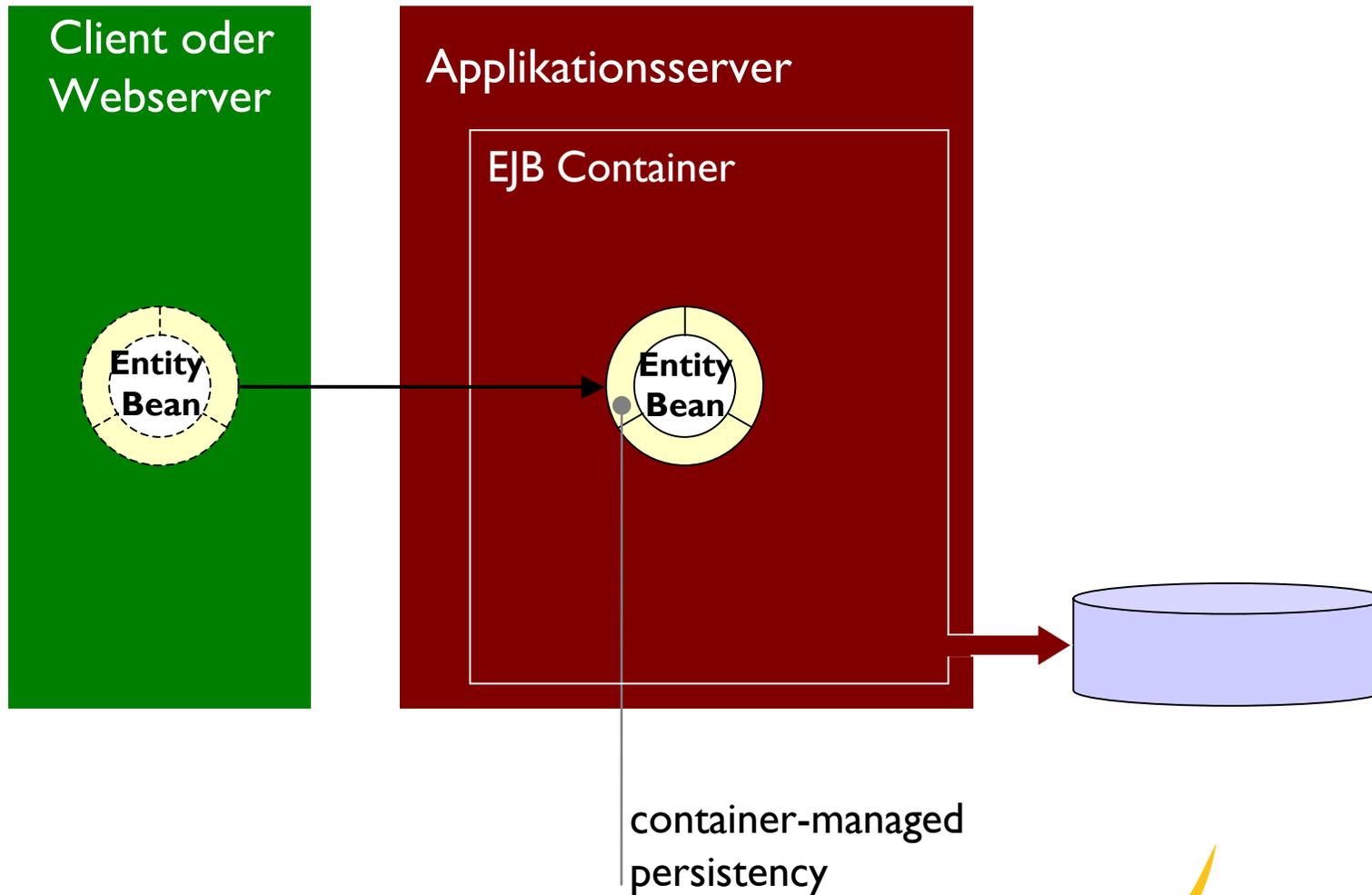
# Persistence-Bridge – Anfragen

- Objektübergreifende Abfragen auf der Ebene des Objektmodells
- Kein SQL notwendig
- Query-By-Example oder OQL für dependent Objects
- OQL für die Definition der finder-Methoden
- Optimierung durch seitenweises Lesen und Einschränken auf bestimmte Attribute

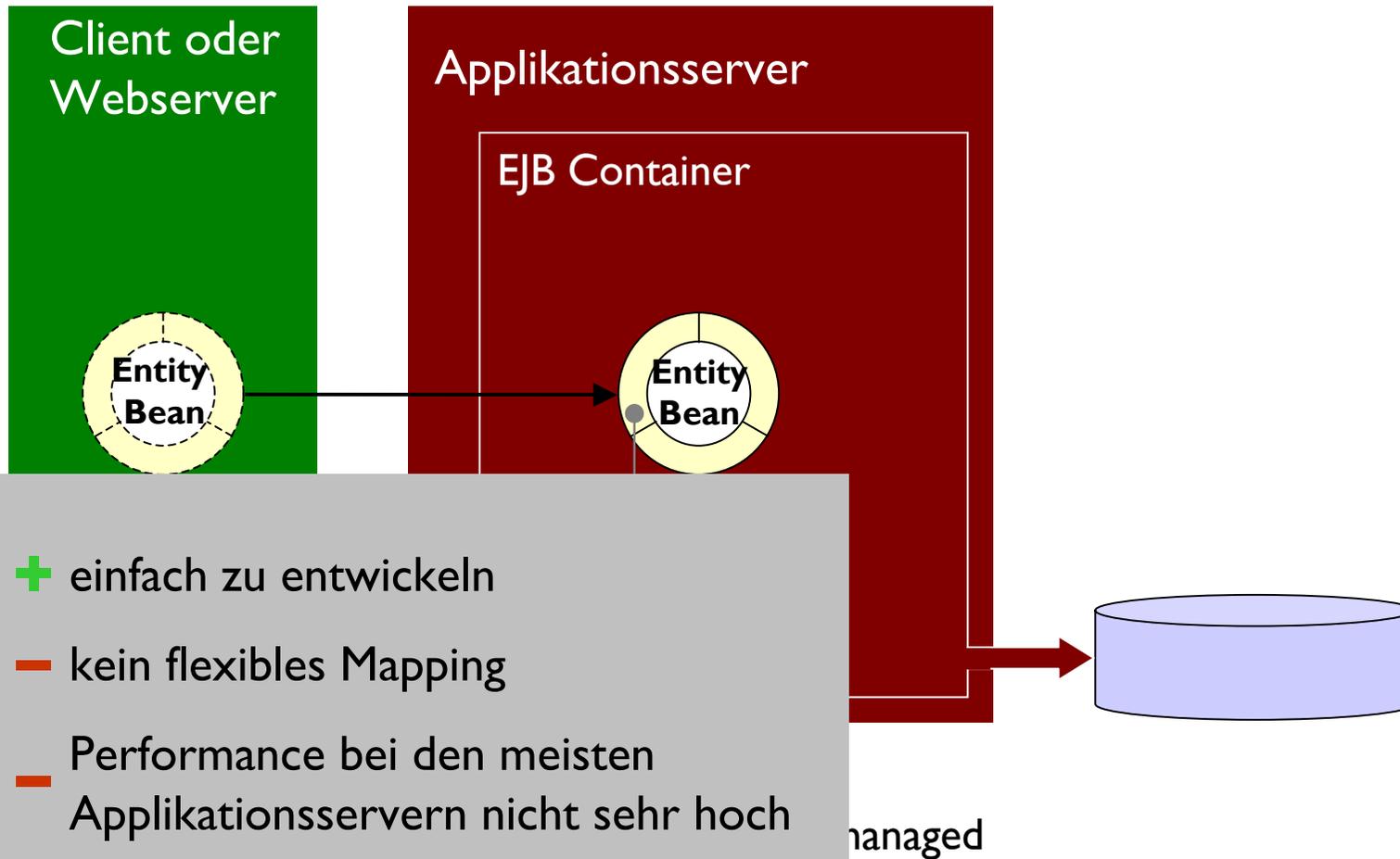
# Persistence-Bridge – Transaktionen

- Transaktionen können auf Objekt-Ebene verwaltet werden
- Wird eine Transaktion abgeschlossen, werden alle Änderungen mit einer Datenbank-Transaktion geschrieben
- Integration mit dem Java Transaction Service (JTS)

# container-managed persistency

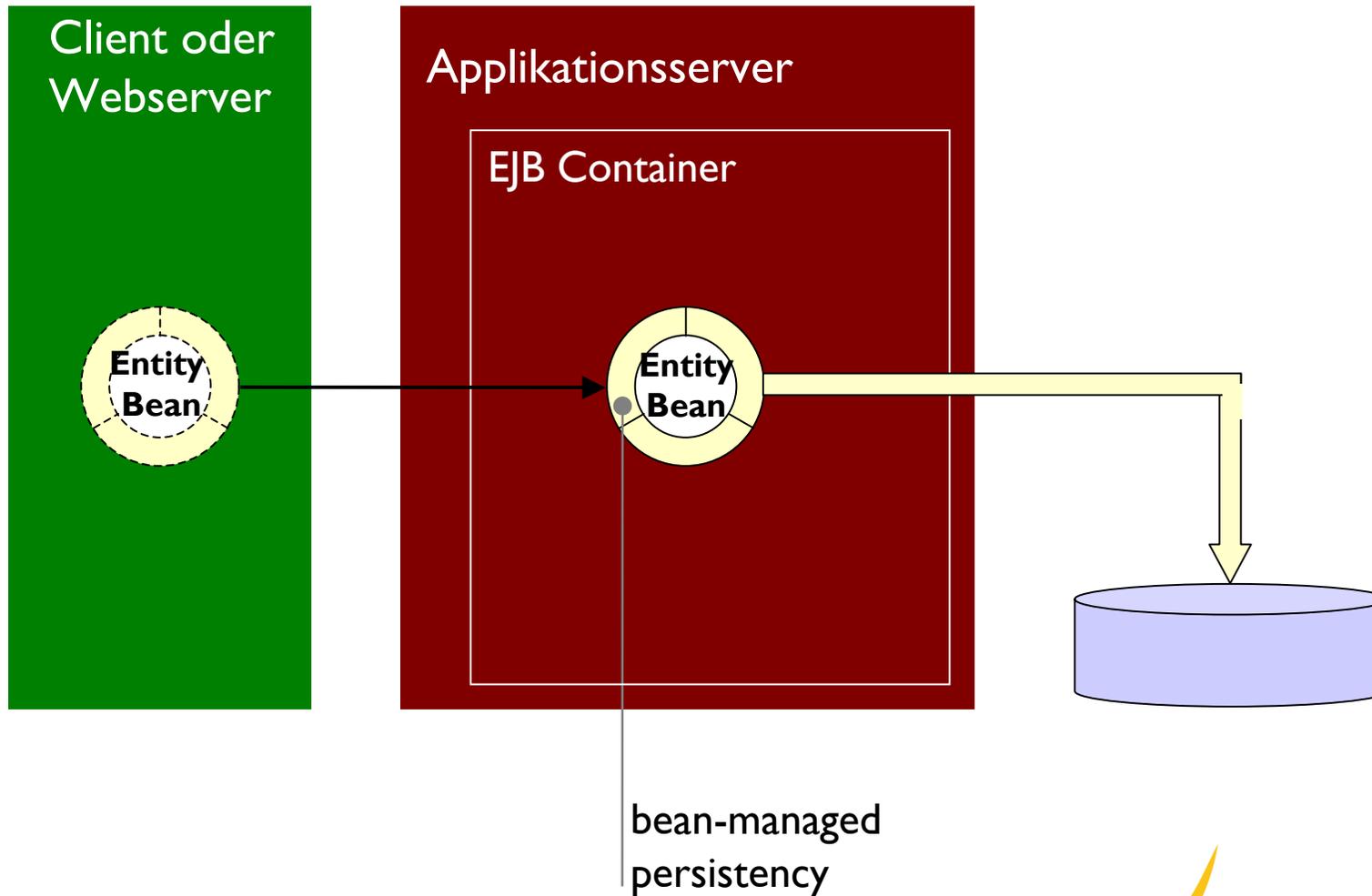


# container-managed persistency

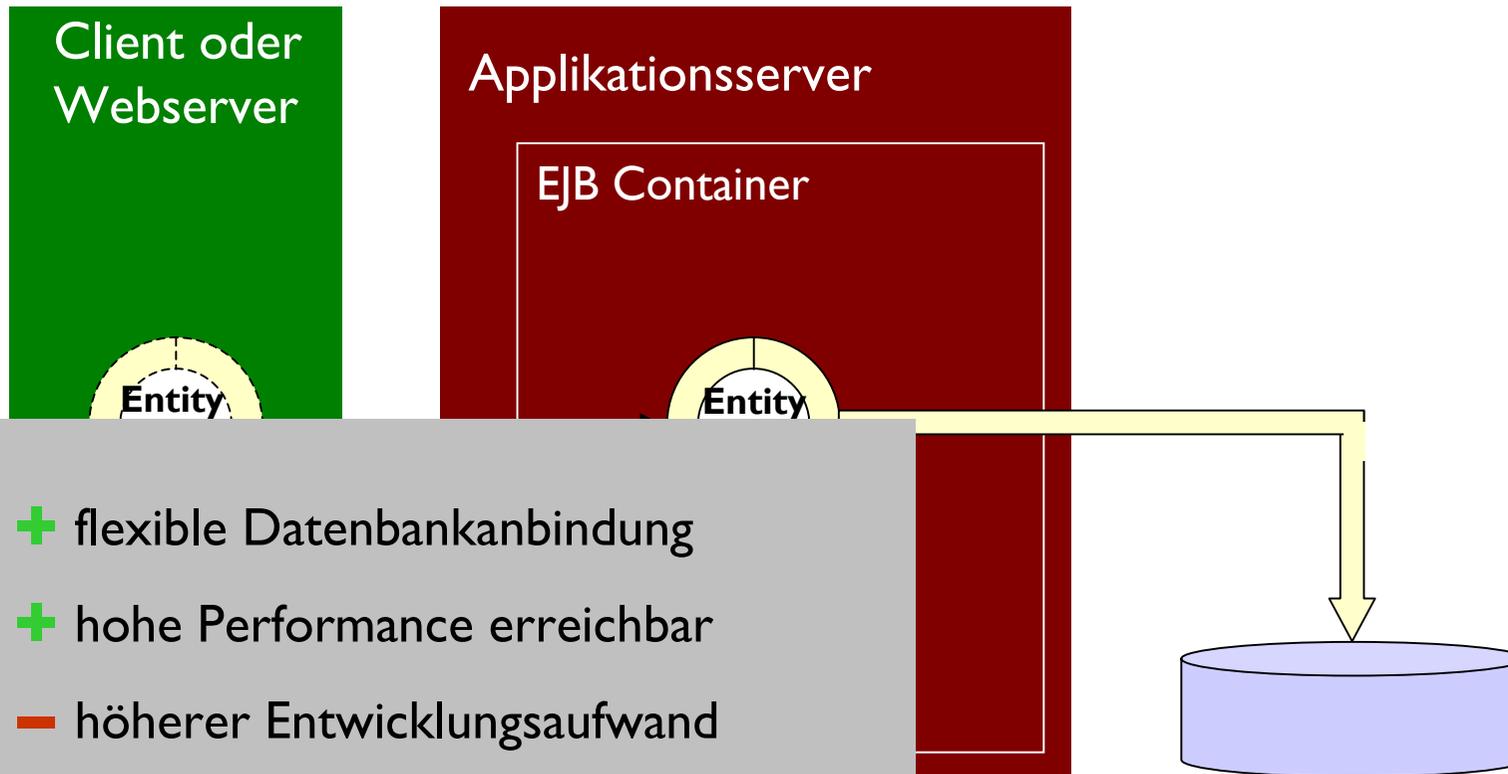


- + einfach zu entwickeln
- kein flexibles Mapping
- Performance bei den meisten Applikationsservern nicht sehr hoch

# bean-managed persistency

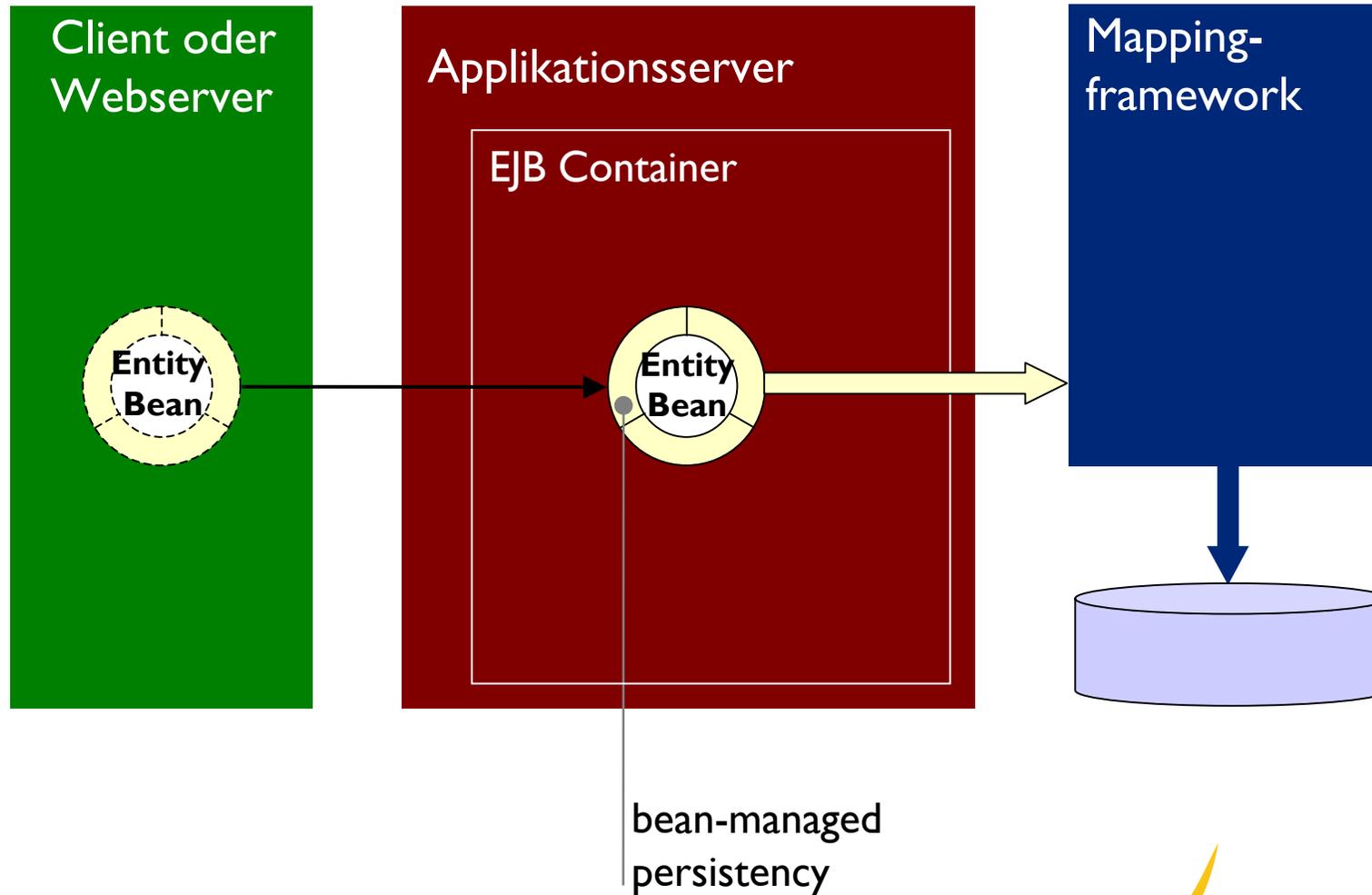


# bean-managed persistency

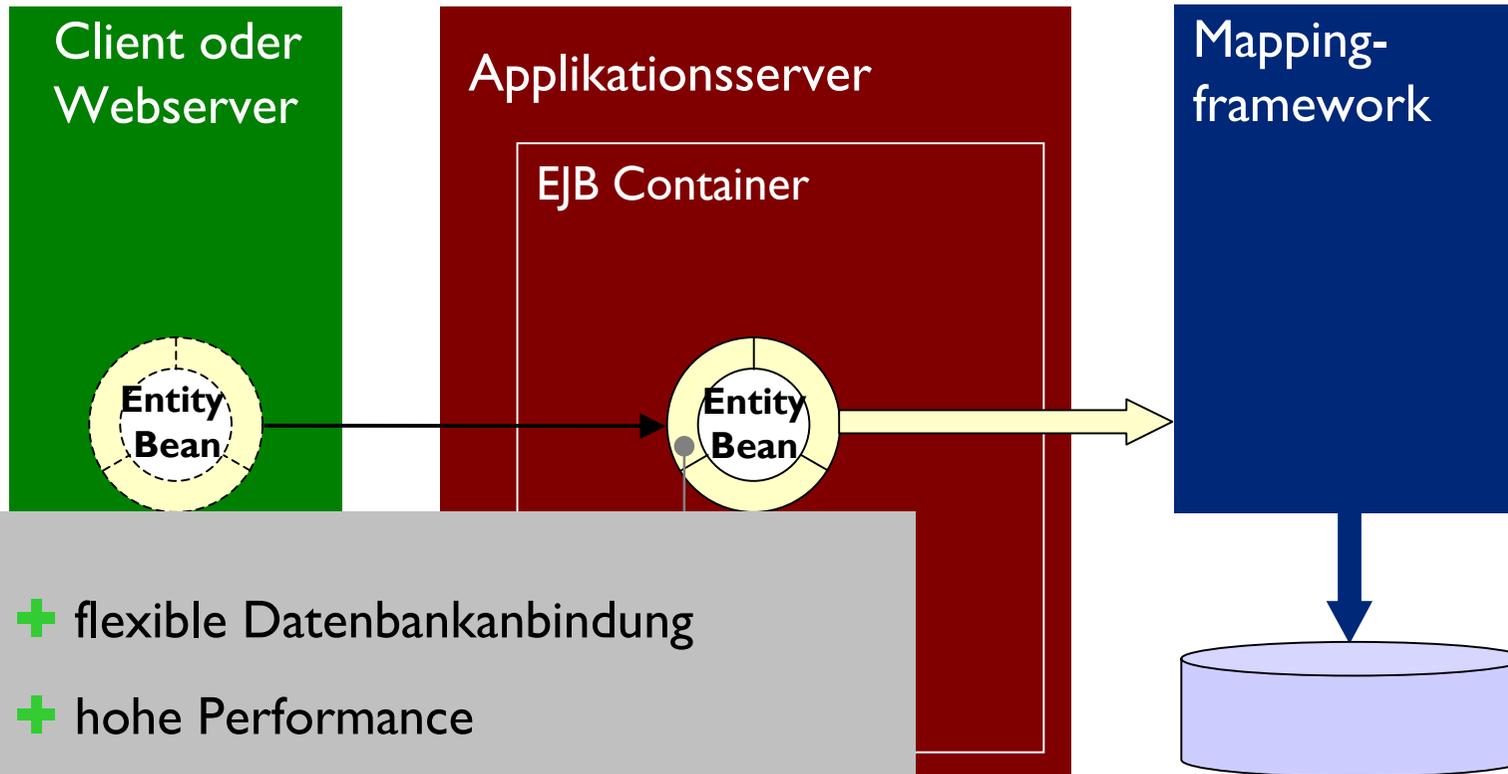


- + flexible Datenbankbindung
- + hohe Performance erreichbar
- höherer Entwicklungsaufwand
- schlechte Wartbarkeit, da EJB seine Datenbankabbildung kennt

# objekt-relacionales Mappingframework

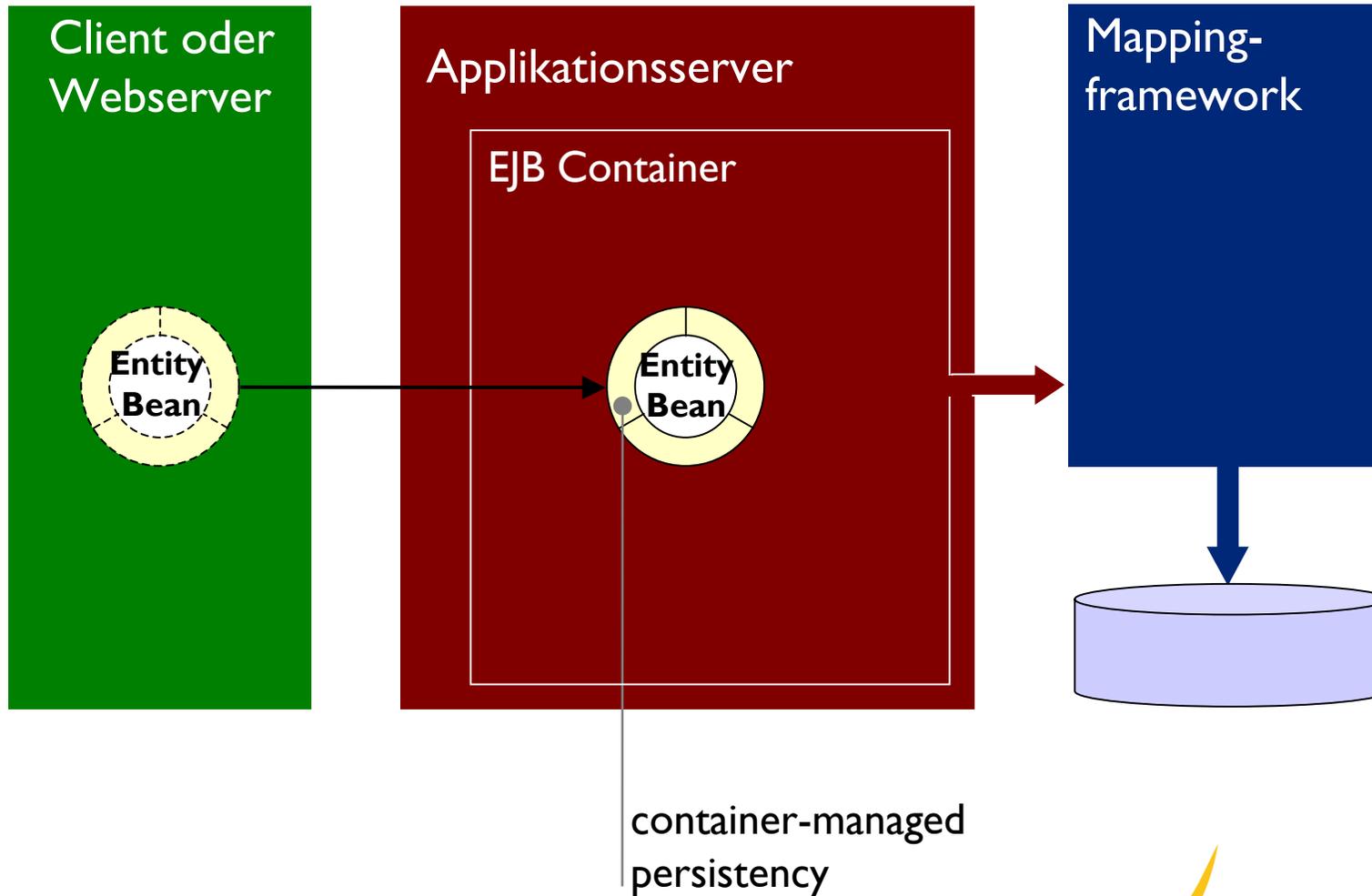


# objekt-relacionales Mappingframework

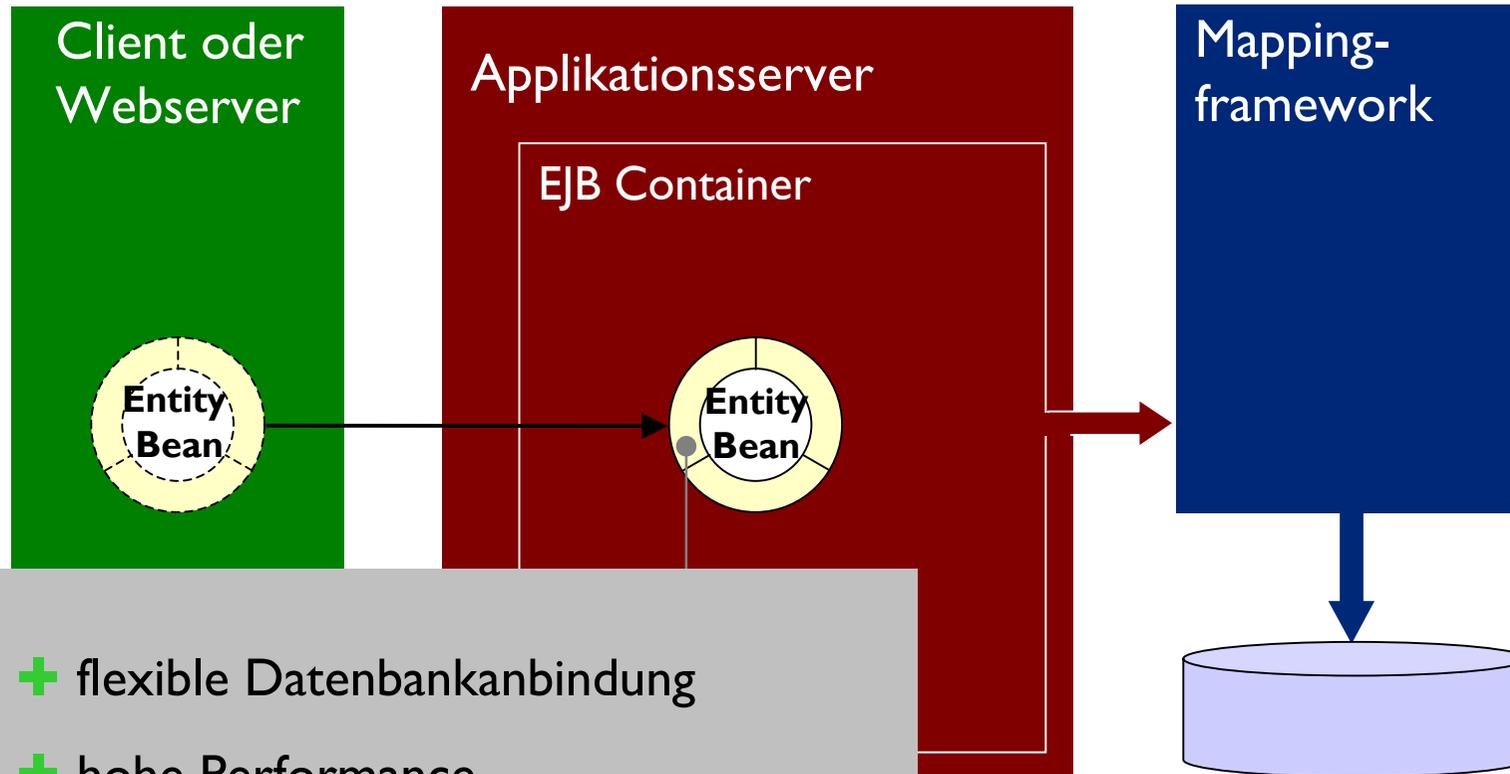


- + flexible Datenbankbindung
- + hohe Performance
- EJB enthält Mappingframework-spezifischen Programmcode

# objekt-relacionales Mappingframework



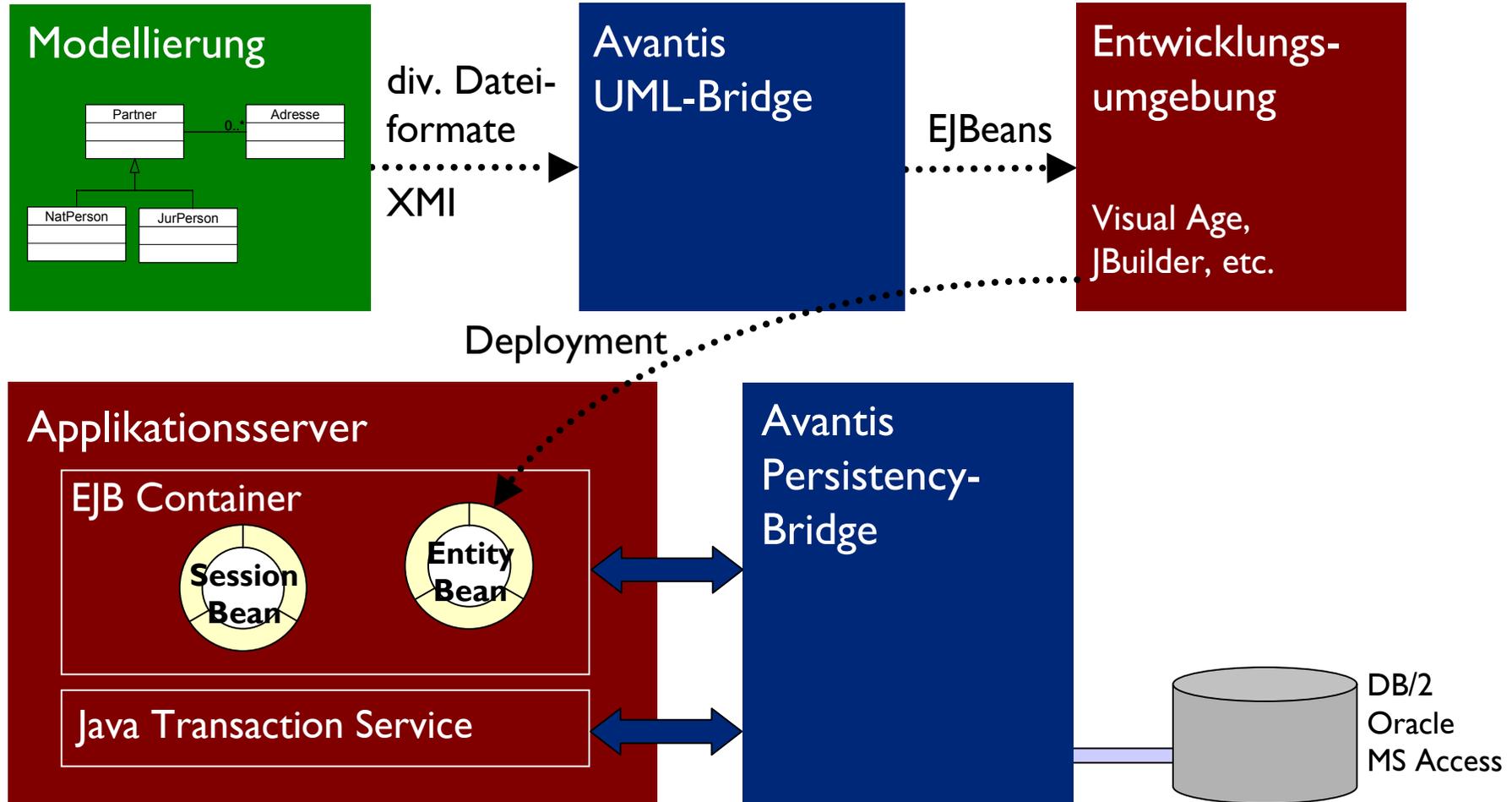
# objekt-relacionales Mappingframework



- + flexible Datenbankbindung
- + hohe Performance
- + EJB kennt Mappingframework nicht

managed

# Entwicklungsprozess für EJBs



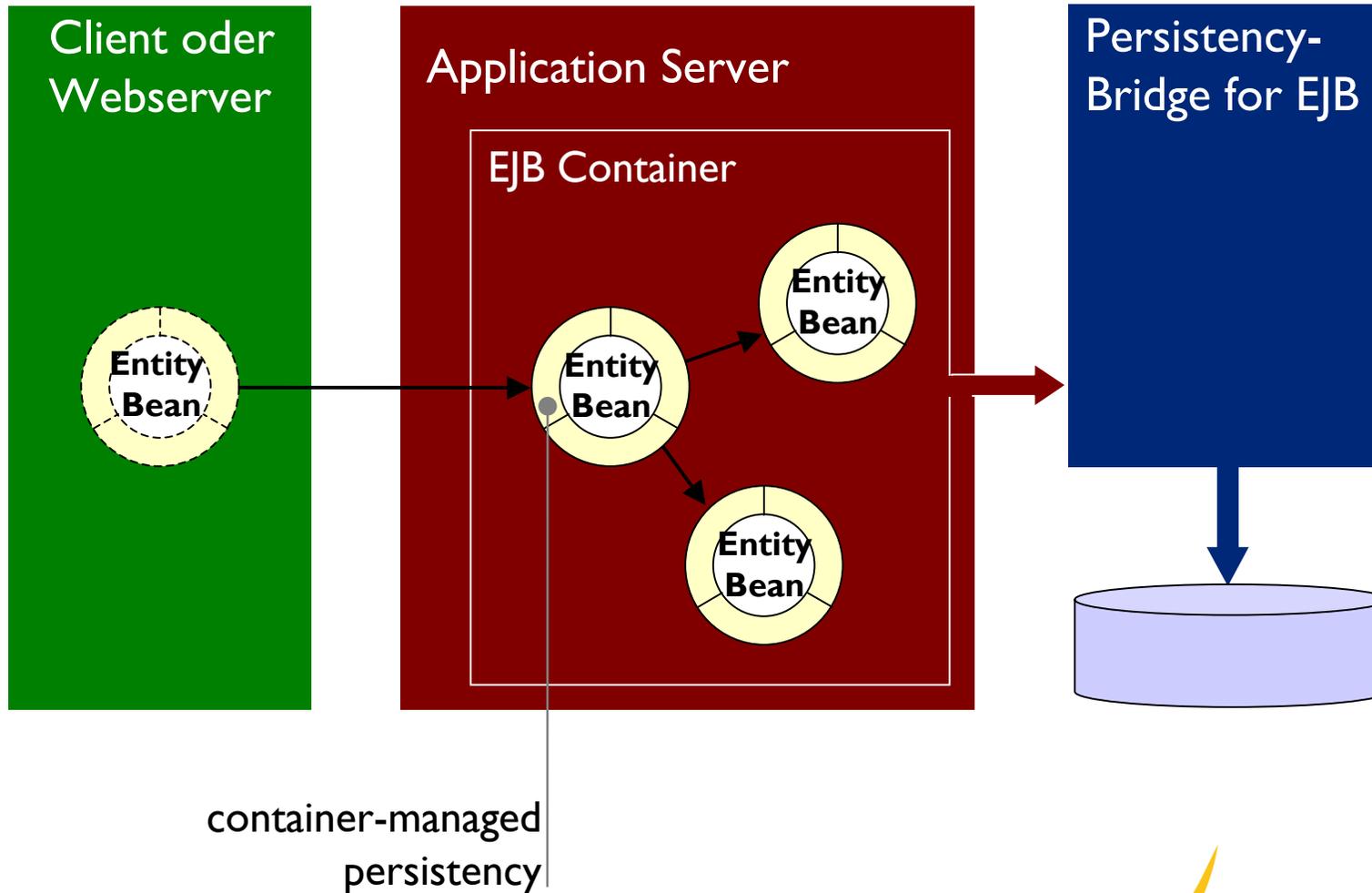
# Inhalt

- Objekt-relationales Mapping
- Avantis Unisuite
- UML-Bridge for EJB
  - Modellierung
  - Import
  - Prüfung
  - Generierung
- Persistency-Bridge for EJB
- **EJB Komponenten mit dependent Objects**
- Features

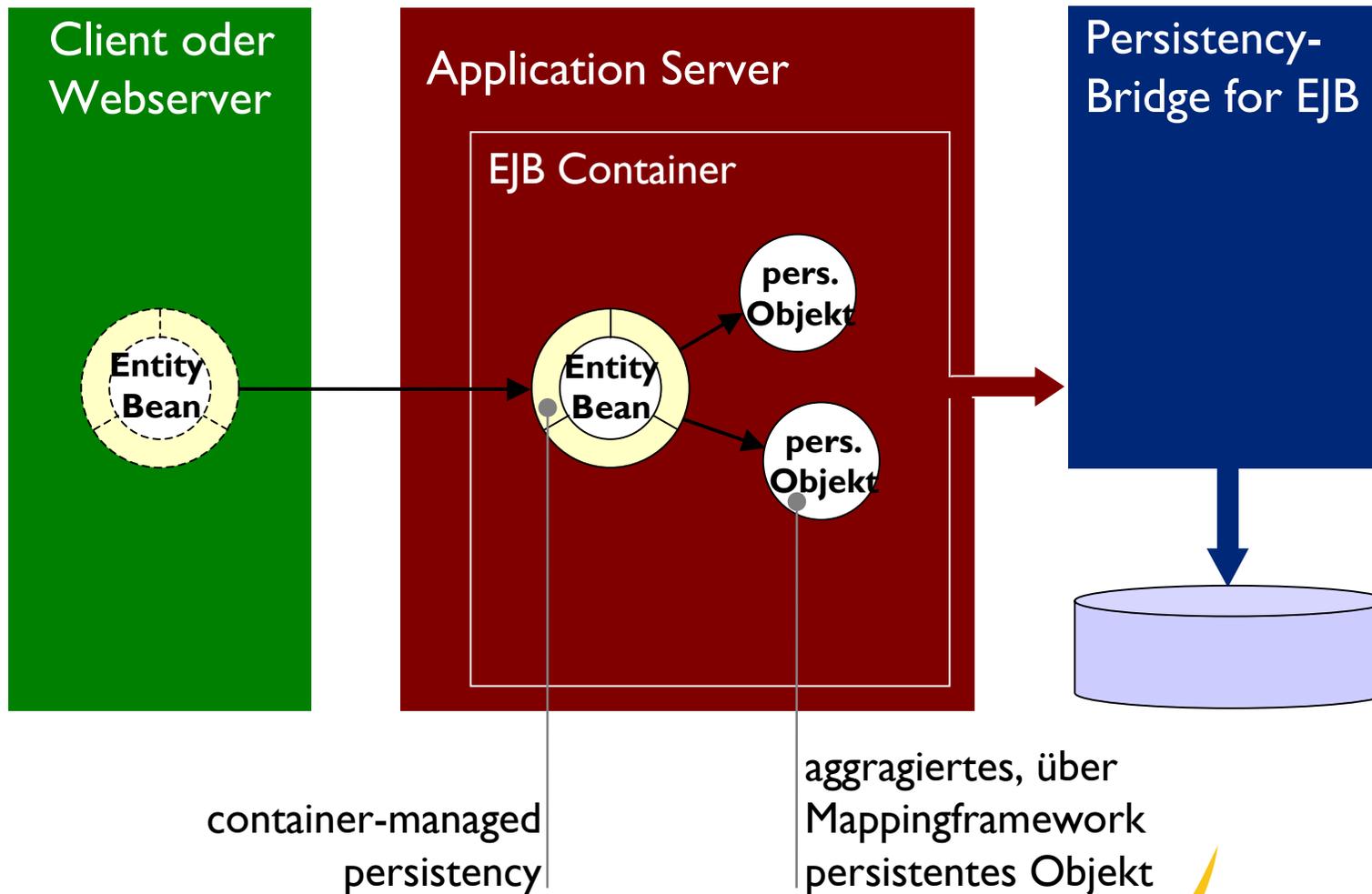
# Performance von Entity-Beans

- Wenn Entity Beans direkt Entitäten des Datenbankschemas darstellen, erhält man eine sehr fein-granulare Entity Bean Struktur.
- Da Entity Beans verteilte Objekte sind, führen fein-granulare Entity Bean Strukturen sehr schnell zu einem Performanceengpass, da die Kommunikation zwischen EJBs über Interprozesskommunikation und häufig unter Verwendung des Naming-Service abläuft.

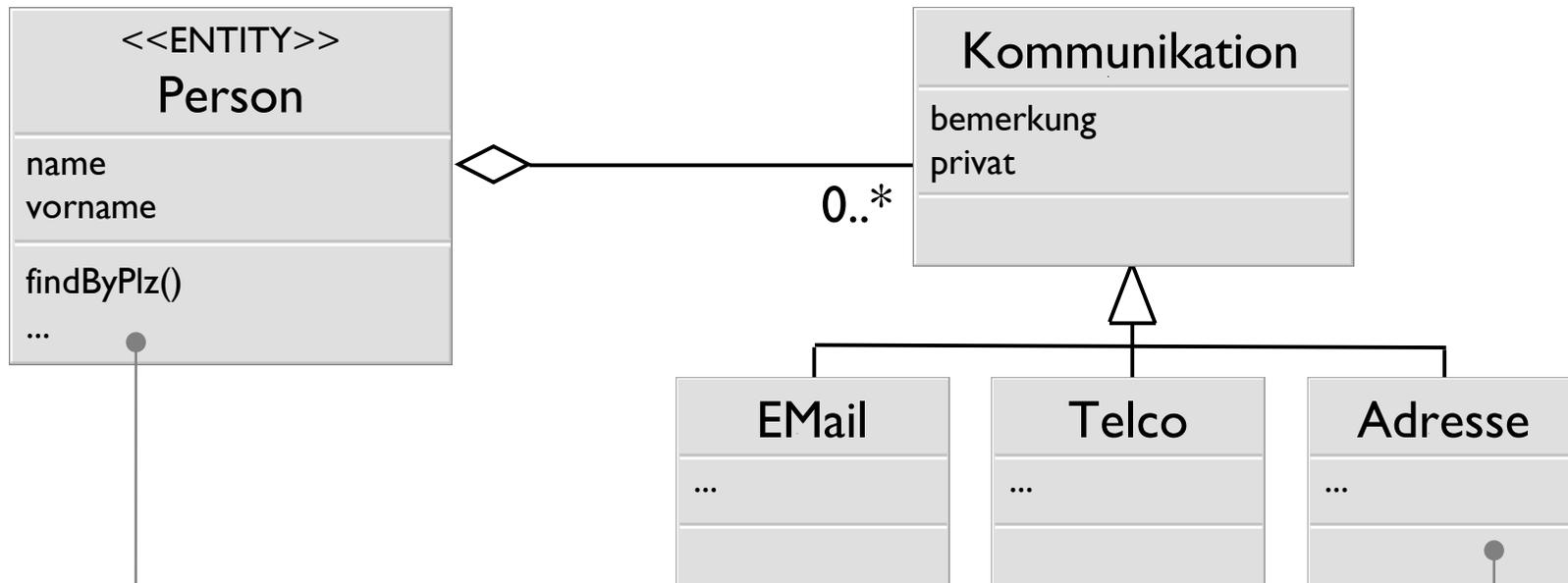
# Fein-granulare EJB Struktur



# EJB Komponente mit dependent, persistenten Java-Klassen



# EJB Komponente Beispiel

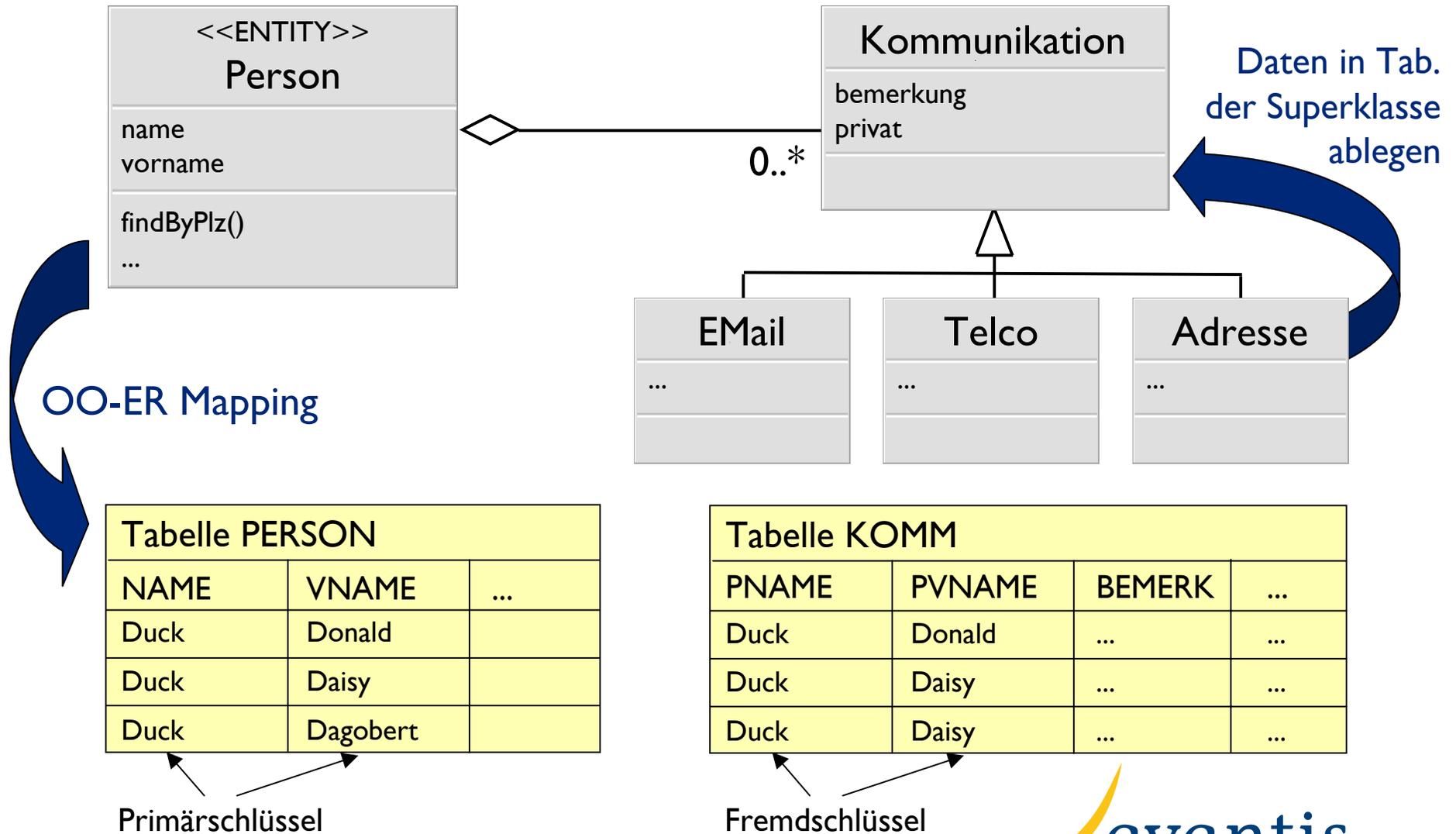


Client hat ausschließlich Zugriff auf EJB Person. Person bietet Methoden für den Zugriff auf die Kommunikations-Objekte.

Persistente  
Dependent Objects

# EJB Komponente

## OO-ER Mapping



# Inhalt

- Objekt-relationales Mapping
- Avantis Unisuite
- UML-Bridge for EJB
  - Modellierung
  - Import
  - Prüfung
  - Generierung
- Persistency-Bridge for EJB
- EJB Komponenten mit dependent Objects
- **Features**

# features ...

	Unisuite for EJB
<b>Design im UML-Modell</b>	
Rational Rose	✓
MID Innovator	✓
<b>Generierung</b>	
Session Beans	✓
Entity Beans	✓
Dependent Objects	✓
Deployment Descriptor	✓
<b>Persistenz</b>	
bean managed persistency	✓
container managed persistency	✓*
dependent Objects	✓
<b>Transaktionsverwaltung</b>	
Transaktionen auf Objekteben	✓
Java Transaction Service (JTS) Integration	✓

\* Derzeit nur für Inprise Application Server, geplant: WebSphere, WebLogic, iPlanet

# ... features ...

	Unsuite for EJB
<b>Mapping</b>	
Flexibles Mapping von Attributen	✓
Flexibles Mapping von Vererbung	✓
Assoziationen zwischen Java Klassen	✓
Assoziationen zwischen Entity Beans	✓
Assoziationen zwischen Entity Beans und Java Klassen	✓
<b>Abfrageservice</b>	
Abfrage auf Objekt-Ebene	✓
Query-By-Example (dep. Objects)	✓
OQL-Support (EJB und dep.Objects)	✓
<b>Sicherstellen der Objektidentität</b>	
Unterstützung von fachlichen Schlüsseln	✓
Unterstützung von technischen Schlüsseln	✓

# ... features.

	Unisuite for EJB
<b>Nebenläufigkeitsverwaltung</b>	
Optimistisches Sperren mit Zeitstempel	✓
Optimistisches Sperren mit Versionsnummer	✓
Pessimistisches Sperren	✓
<b>Performance</b>	
Seitenweises Lesen von großen Ergebnismengen	✓ *
Unvollständiges Lesen mit Nachladen	✓ *
Object Caching	✓
Caching von Assoziationen und Aggregationen	✓
<b>Datenbanken</b>	
DB/2	✓
Oracle	✓
MS Access	✓
weitere auf Anfrage	✓

\* Derzeit nur für depended objects

# Avantis Unisuite



Fragen?

Ralf Schippert

ralf.schippert@avantis.de

Avantis GmbH

Max-Lang-Straße 24

D-70771 Leinfelden

fon +49 (0)711.903748-0

fax +49 (0)711.903748-10

web [www.avantis.de](http://www.avantis.de)

