



JDO – Java Data Objects

Ralf Degner, Chief Consultant

Ralf.Degner@poet.de

Be Persistent.

FAST OBJECTS™

Agenda

- POET
- Motivation
- Geschichte
- Einführung
- Architekturen
- FastObjects


POET

- Gegründet 1993
- Zwei Produktlinien
 - eSupplier Solutions: Kataloglösungen für B2B-Onlinehandel
 - FastObjects: Objektdatenbank
- Hauptstandort Hamburg Volksdorf mit ca. 70 Mitarbeiter

Motivation

- Persistenz in Java™
 - Serialisierung
 - ✍ transparent
 - ✍ nicht transaktional, keine Queries, kein Concurrency-Handling
 - JDBC – relationale Datenbanken
 - ✍ Transaktionsschutz, Queries, Concurrency-Handling
 - ✍ explizite Datenbankoperationen
- Ziel von JDO: transparente, transaktionale Schnittstelle zu Datenquellen (wie Datenbanken, ERP-Systeme, ...)

Geschichte

-  – Object Data Management Group
 - 1993 Standard für transparente Persistenz von Objekten (C++ und Smalltalk)
 - 1997 Language Binding für Java
 - Basis für JDO
- **JDO**
 - JSR 012 verabschiedet April 2002
 - Version 1.0.1 in Kürze
 - Spezifikation wesentlich genauer als ODMG



Einführung - Persistente Klassen

- Keine Änderung am Quelltext

- Zusatzinformationen in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jdo SYSTEM "jdo.dtd">
<jdo>
  <package name="com.poet.example">
    <class name="TestClass">
      </class>
    </package>
  </jdo>
```

- `.class`-Dateien werden von „Enhancer“ gepatched
 - Objekte werden beim ersten Zugriff automatisch geladen
 - Geänderte Objekte werden beim commit der Transaktion automatisch geschrieben

Einführung - Objektidentität

- **ObjectId identifiziert Objekt eindeutig**
 - Datastore identity: Identität wird von Datenbanksystem erzeugt (Objektdatenbanken)
 - Application identity: Identität durch Anwendung (primary key, OR-Mapper)
- **Zwei Klassen von Objekten**
 - First Class Objects (FCOs): Objekte mit eigener Identität
 - Second Class Objects (SCOs): keine eigene Identität, abhängig von FCO (sinnvoll für ValueObjects)

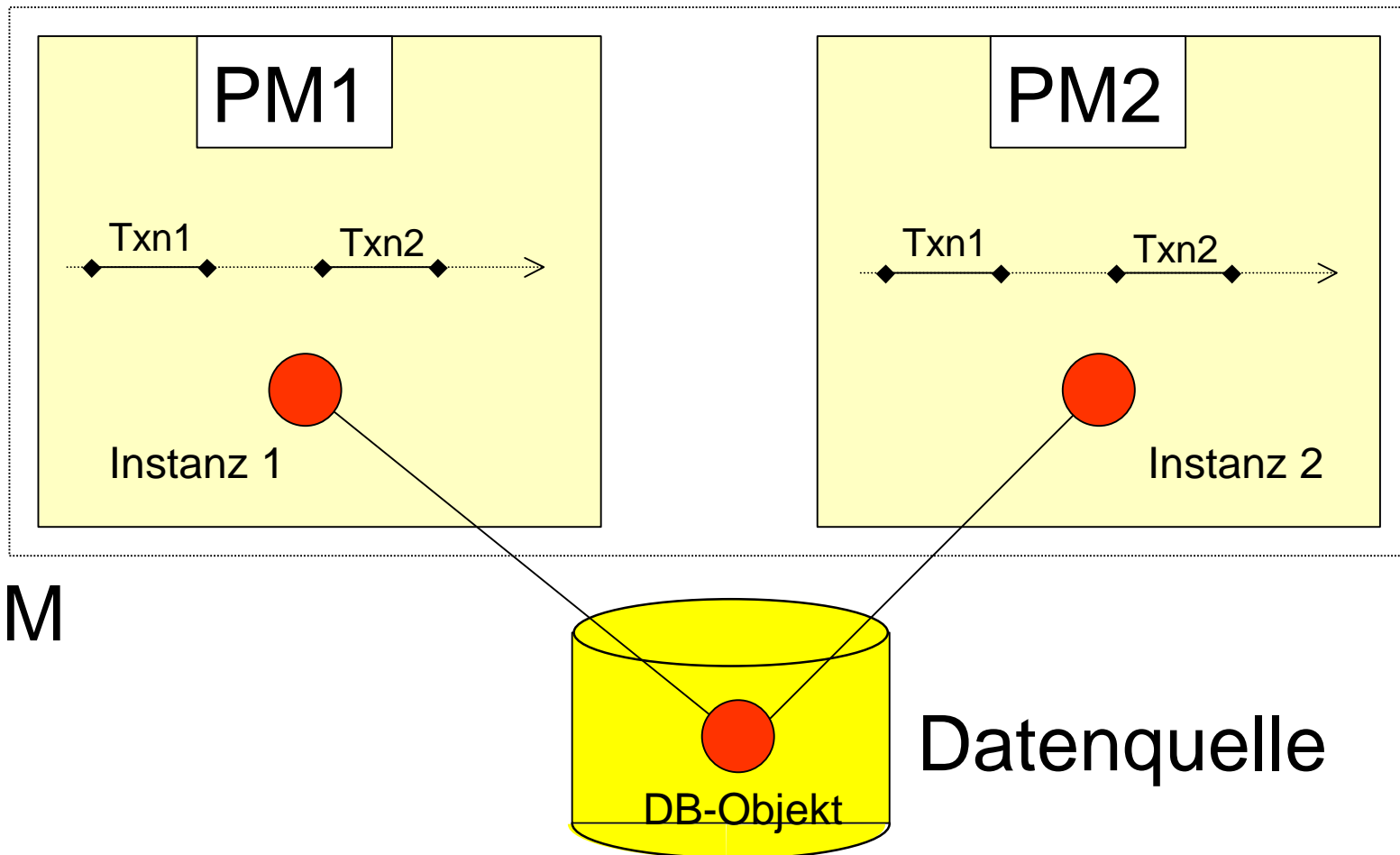
Einführung – PersistenceManager (I)

- Verwalter von JDO-Objekten
- Erzeugt über PMFactory, PM gehört zu einer Datenquelle
- Pro PersistenceManager max. eine Transaktion zur Zeit

```
PersistenceManager pm = pmf.getPersistenceManager();  
Transaction txn = pm.currentTransaction();  
txn.begin();  
TestClass testObject = new TestClass();  
pm.makePersistent(testObject);  
txn.commit();
```

- Persistence by Reachability: Objekt persistent, wenn explizit persistent oder von persistentem Objekt erreichbar

Einführung - PersistenceManager (II)



Einführung – weitere Begriffe

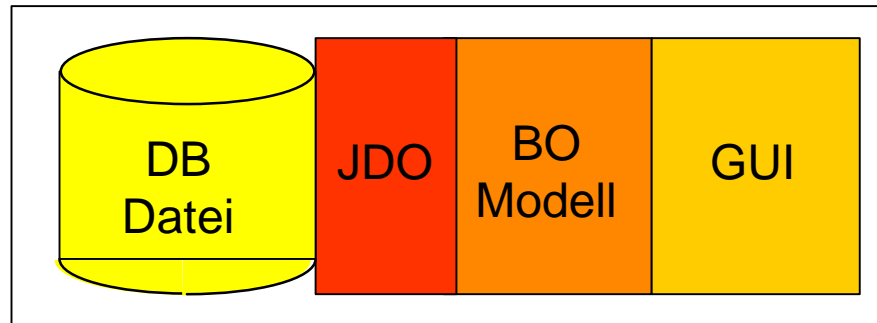
- Extent – Menge aller Instanzen eines Typs
- JDOQL – JDO Query Language
- Optional
 - andere Query Language (z.B. OQL oder SQL)
 - Transaktionschutz für transiente Objekte
 - optimistische Transaktionen
 - Arrays (kein change tracking)

Einführung - Zusammenfassung

- **Transparent**
 - Alle Möglichkeiten von Java werden unterstützt
 - Keine expliziten Operationen zum Laden oder Speichern
- **Schlank**
 - API
 - Laufzeitumgebung

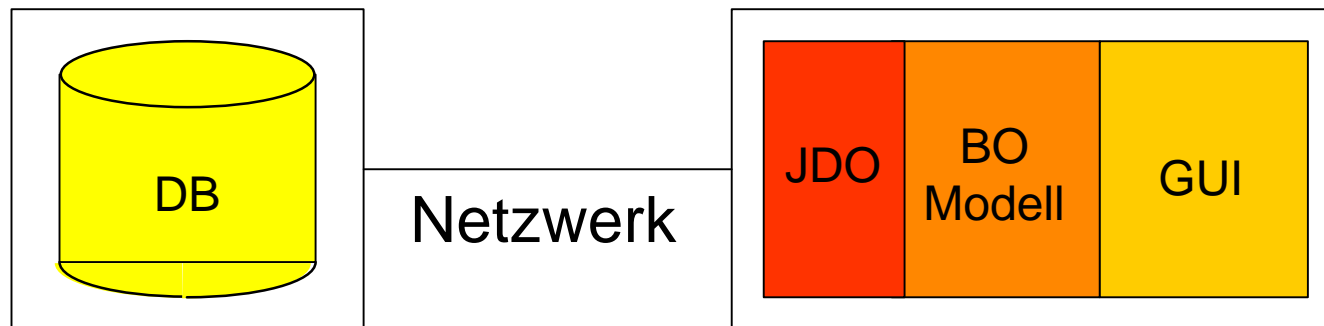
Architekturen - klassisch

- Embedded – direkter Zugriff auf Datenbankdatei



Client

- Client/Server

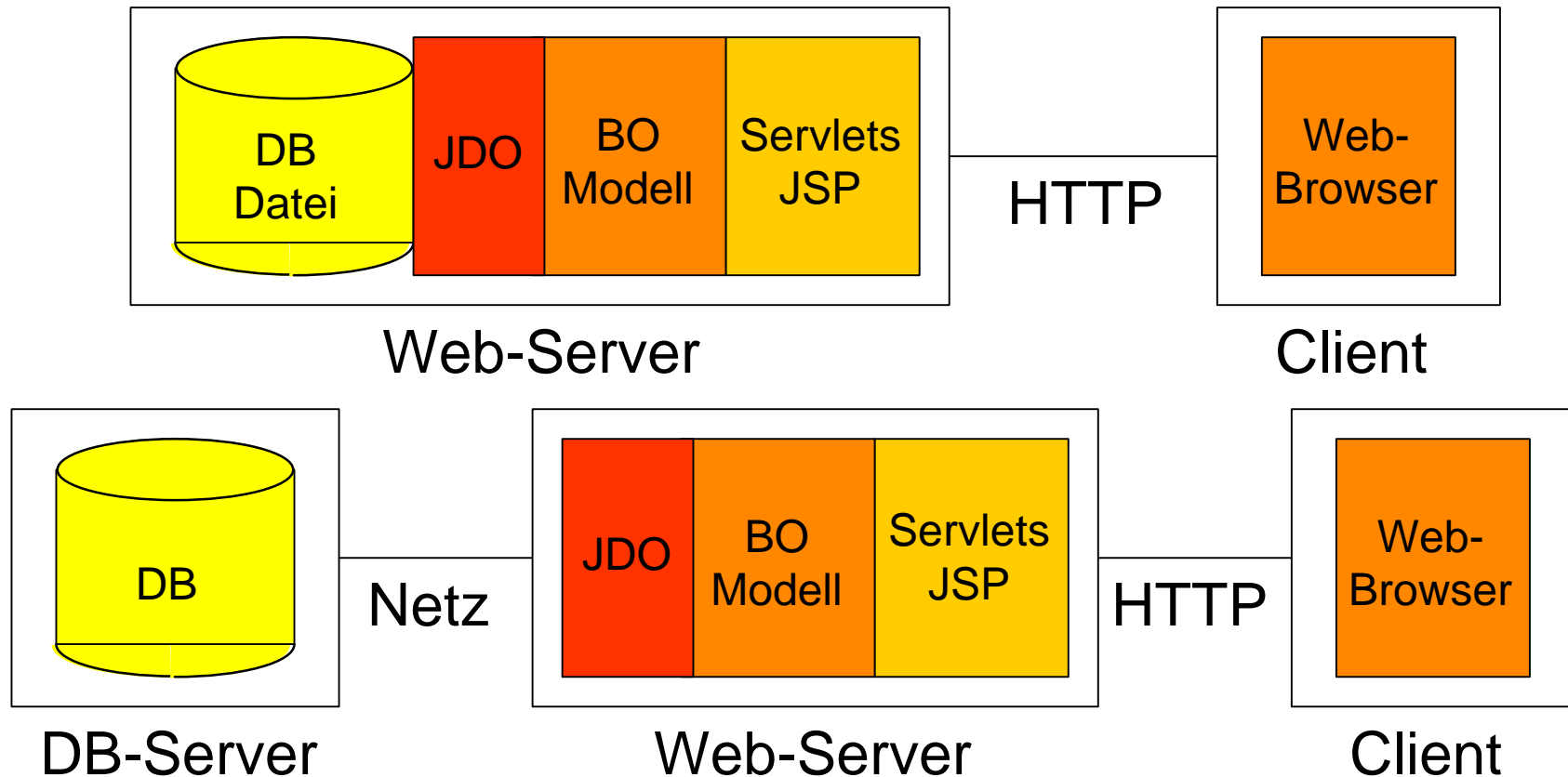


DB-Server

Client

Architekturen - Web

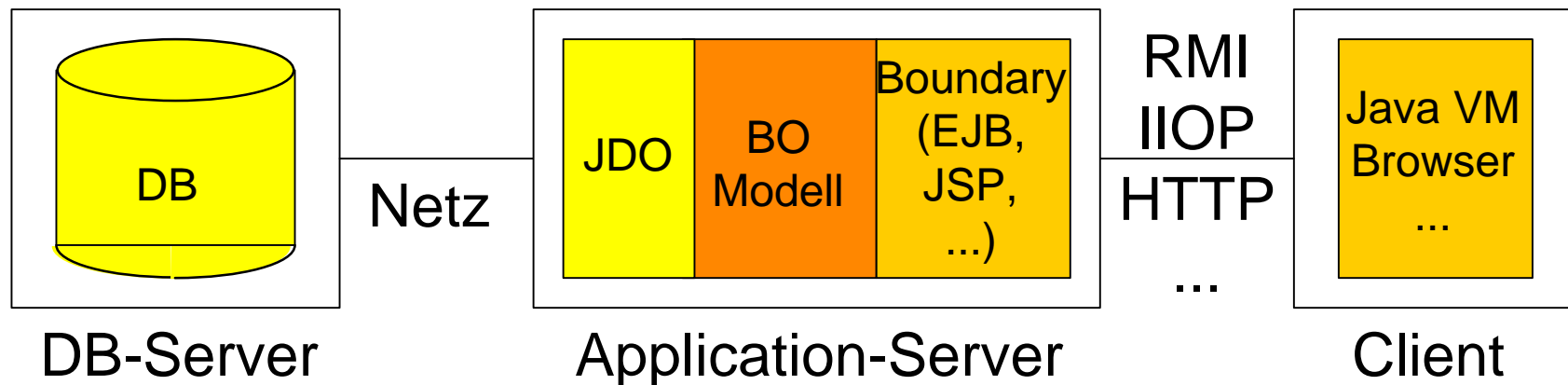
- Servlets oder JSPs



Architekturen – Application-Server

- **Managed JDO**
 - Integration in J2EE Connector Architecture als Resource Adapter
- **EJB**
 - Session Beans - JDO Objekte als abhängige Objekte
 - Entity Beans - JDO Objekte als Delegates für BMP und CMP
- **Alternative**
 - CORBA oder RMI mit direkter Nutzung von JDO

Architekturen – Mehrschicht verallgemeinert



- Trennung von BO-Modell und vom Klienten abhängiger Boundary
- Boundaries: JSPs/Servlets, EJBs, CORBA oder RMI Objekte, Message-Service, ...

FastObjects - Features

- Zero Administration
- RDBMS als Storage-Engine
- Schema Evolution
- Hochverfügbarkeit (Replikation, Fail-Over)
- Active Cache
- Verschlüsselung von Datenbank und C/S-Kommunikation
- Nested Transactions
- Windows, Linux, SUN Solaris, HP-UX, IBM AIX
- ...
- Demnächst: Microsoft .NET

FastObjects - Produkte

- j1 – Community Edition, pure Java™
- j2 – pure Java™, eingebettet, sehr schlank
- e7 – Java und C++, eingebettet
- t7 – Java und C++, Client/Server und Application Server

- Ein API für alle aller FastObjects Produkte (für Java™ ODMG/JDO).

Links

- www.javasoft.com/products/jdo
- www.jdocentral.com
 - JDO-Portal
- www.fastobjects.de
 - Download von j1 und e7 Trial
- community.fastobjects.com
 - Entwickler-Portal



Ende

Vielen Dank für Ihre
Aufmerksamkeit!