



**Analyse und praktischer
Vergleich von neuen Access-
Layer-Technologien in
modernen Webanwendungen
unter Java**

Oliver Kalz





Agenda

- **Grundlagen**
- Objektpersistenz
- Objektrelationales Mapping
- Performance
- Fazit



Das Client/Server-Modell

- Server
 - stellt Dienst für andere Rechner zur Verfügung
- Client
 - nutzt bereitgestellte Dienste
- Aufteilung der Programmfunktionalität
 - Präsentationsschicht (Presentation Layer)
 - Geschäftslogik (Business Logic Layer)
 - Persistenzschicht (Access Layer)



Java & XML

- Java
 - moderne objektorientierte Sprache
 - Interpretation durch virtuelle Maschine
- XML
 - textbasierte Beschreibungssprache
 - Strukturierung der Daten durch Tags und Attribute
- Java & XML
 - portabler Code & portable Daten



Agenda

- Grundlagen
- **Objektpersistenz**
- Objektrelationales Mapping
- Performance
- Fazit



Objektpersistenz

- Mechanismus, um Objekte über die Laufzeit einer Anwendung hinaus zu erhalten
- persistent vs. transient
- Speicherung im Sekundärspeicher
 - Dateien
 - Datenbanken



Objektpersistenz: relationale Datenbanken

- etabliertes DB-System
- Organisation der Daten in Tabellen
- Identifizierung über Schlüssel
- Anfragesprache SQL
- Impedance Mismatch

R

a_1	...	a_n
	...	
	...	
	...	

R Relationenname

a_x Attribute



Objektpersistenz: relationale Datenbanken

- Schwächen
 - ▶ Segmentierung
 - ▶ künstliche Schlüsselattribute ohne Anwendungssemantik
 - ▶ Verhalten nicht speicherbar
 - ▶ keine Darstellung von Vererbungsstrukturen
 - ▶ externe Programmierschnittstelle für die Einbettung von SQL in andere Sprachen



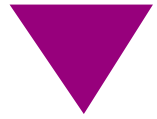
Objektpersistenz: objektorientierte Datenbanken

- Datenbanksysteme mit objektorientiertem Datenmodell
- Entwicklung seit Mitte der 1980er Jahre
- bisher keine Konkurrenz für relationale Datenbanken
- Standards
 - Database Manifesto (1989)
 - ODMG (1993-2000)
 - JDO (2002)



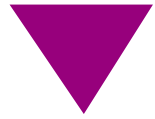
Objektpersistenz: objektorientierte Datenbanken

- Entwicklungslinien
 - evolutionäre Linie
 - revolutionäre Linie
 - Objektdatenbanken
- Vorteil
 - Speicherung von Struktur und Daten
- Nachteile
 - Migration schwierig
 - Systeme nicht austauschbar



Agenda

- Grundlagen
- Objektpersistenz
- **Objektrelationales Mapping**
- Performance
- Fazit

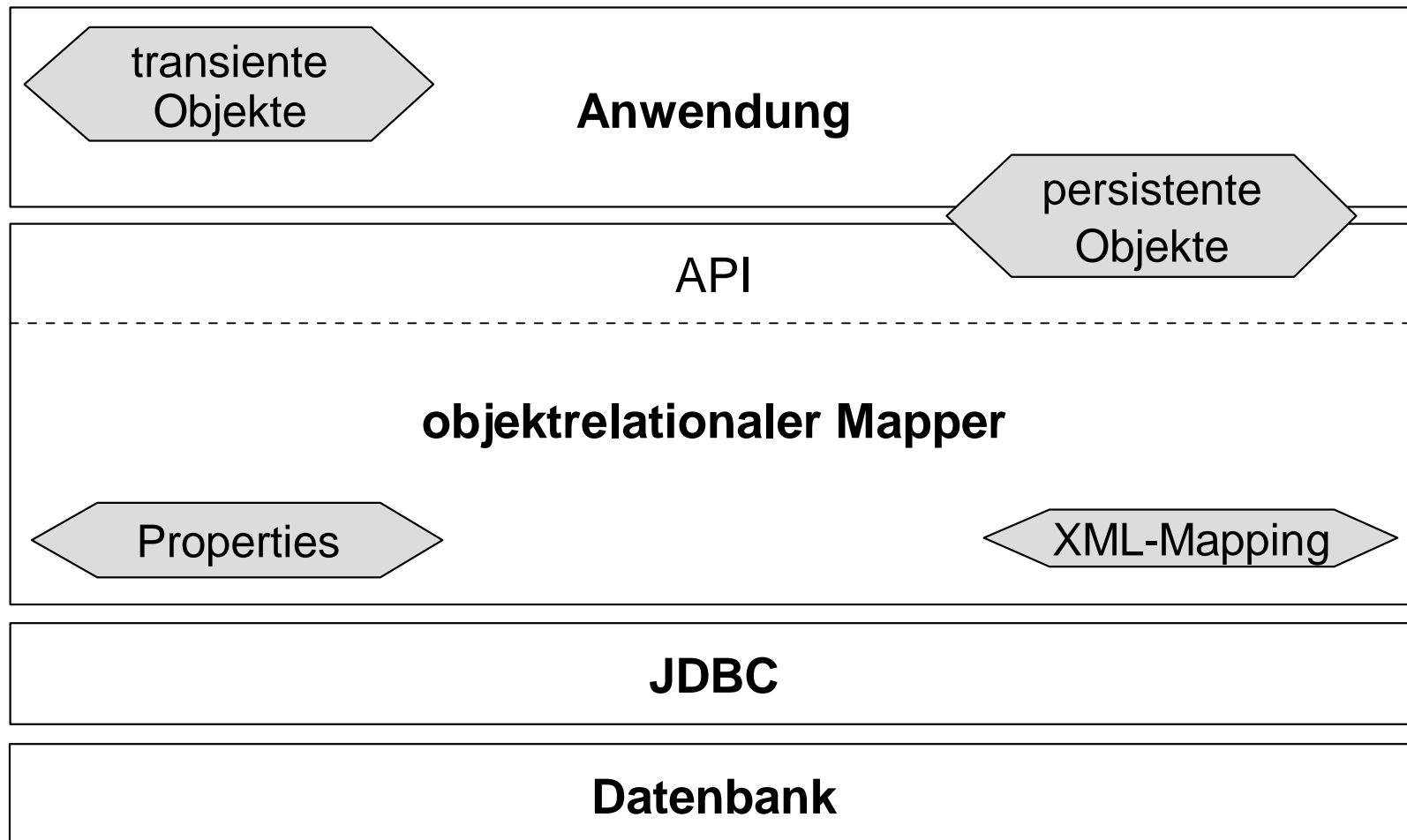


Objektrelationales Mapping

- Abbildung von Objekten auf relationale Strukturen
 - Klasse -> Tabelle
 - Attribut -> Spalte
- O/R-Mapper als Persistenzschicht zwischen Anwendung und Datenbank
- Schemageneratoren, Codegeneratoren
- Persistence by Reachability



Objektrelationales Mapping





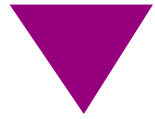
Objektrelationales Mapping vs. JDBC

■ JDBC

```
Statement s = c.createStatement();
ResultSet rs = s.executeQuery(
    "SELECT * FROM states WHERE id=42");
while ( rs.next() ) {
    // Daten holen u. Objekt(e) erzeugen
}
```

■ O/R-Mapper (hier: Hibernate)

```
State s = session.load(State.class,
    new Long(42));
```



Enterprise JavaBeans

- serverseitiges Komponentenmodell für verteilte Geschäftsanwendungen
- Enterprise JavaBeans Specification
- EJB-Container stellt Laufzeitumgebung zur Verfügung
- Arten von EJBs
 - Session Beans
 - Entity Beans
 - Message Driven Beans



Entity Beans

- Attribute, Methoden, Primärschlüssel
- Persistenzmechanismen
 - Bean Managed Persistence (BMP)
 - Container Managed Persistence (CMP)
- CMP 2.0
 - deklarativ im Deployment-Deskriptor
 - Beziehungen: CMR
 - Anfragesprache: EJB-QL



Entity Beans: CMP 2.0 Beispiel

```
public abstract class StateBean implements EntityBean
{
    public abstract Long getId();
    public abstract void setId(Long id);

    public abstract String getName();
    public abstract void setName(String name);

    public Long ejbCreate(Long id, String name) {
        setId(id);
        setName(name);
        return null;
    }
}
```



Entity Beans: CMP 2.0 Beispiel

```
<entity>
  <ejb-name>State</ejb-name>
  [...]
  <persistence-type>Container</persistence-type>
  <cmp-version>2.x</cmp-version>
  <cmp-field>
    <field-name>id</field-name>
  </cmp-field>
  <cmp-field>
    <field-name>name</field-name>
  </cmp-field>
  <primkey-field>id</primkey-field>
</entity>
```



Java Data Objects

- Standard für transparente Persistenz von Objekten
- Konzepte
 - XML-Metadaten
 - Bytecode Enhancement
 - JDO Query Language (JDOQL)
- JDO-API
 - transaktionsbasiert
 - Anfragen über Extents



O/R-Mapper: Hibernate

- Open Source
- Forward- und Reverse Engineering
- Mapping im XML-Format
- unterstützt alle Arten von Beziehungen
- ODMG-API
- Hibernate-API
 - sitzungsbasiert
 - Hibernate Query Language (HQL)



O/R-Mapper: Oracle TopLink

- kommerziell, Teil von Oracle9iAS
- Forward- und Reverse-Engineering
- Mapping Workbench
- Session
 - Suchen und Laden von Objekten
 - Auswahlkriterien über Expression Builder
- Unit of Work
 - Überwachung registrierter Objekte
 - Änderungen an Objekt-Klon

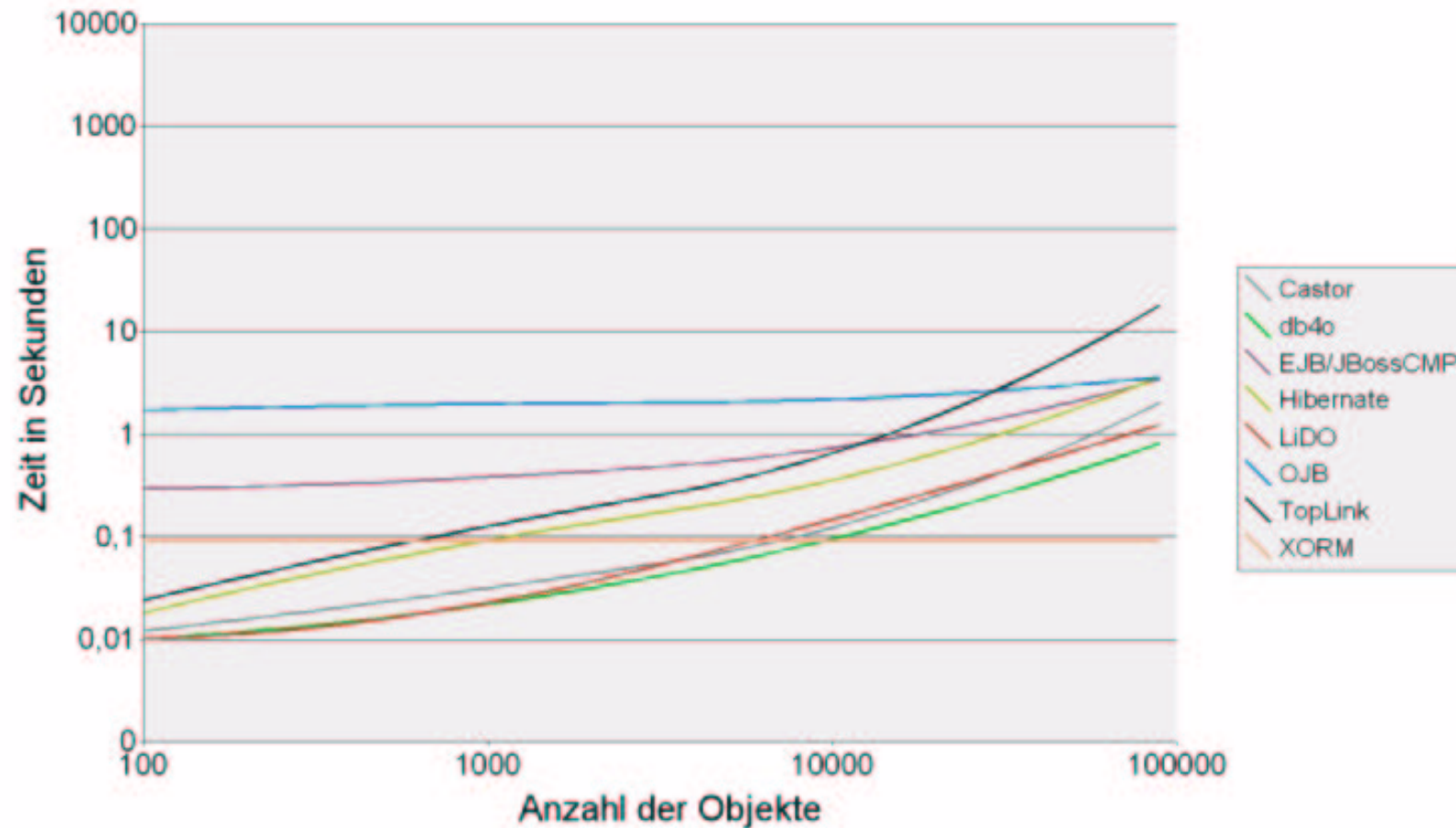


Agenda

- Grundlagen
- Objektpersistenz
- Objektrelationales Mapping
- **Performance**
- Fazit

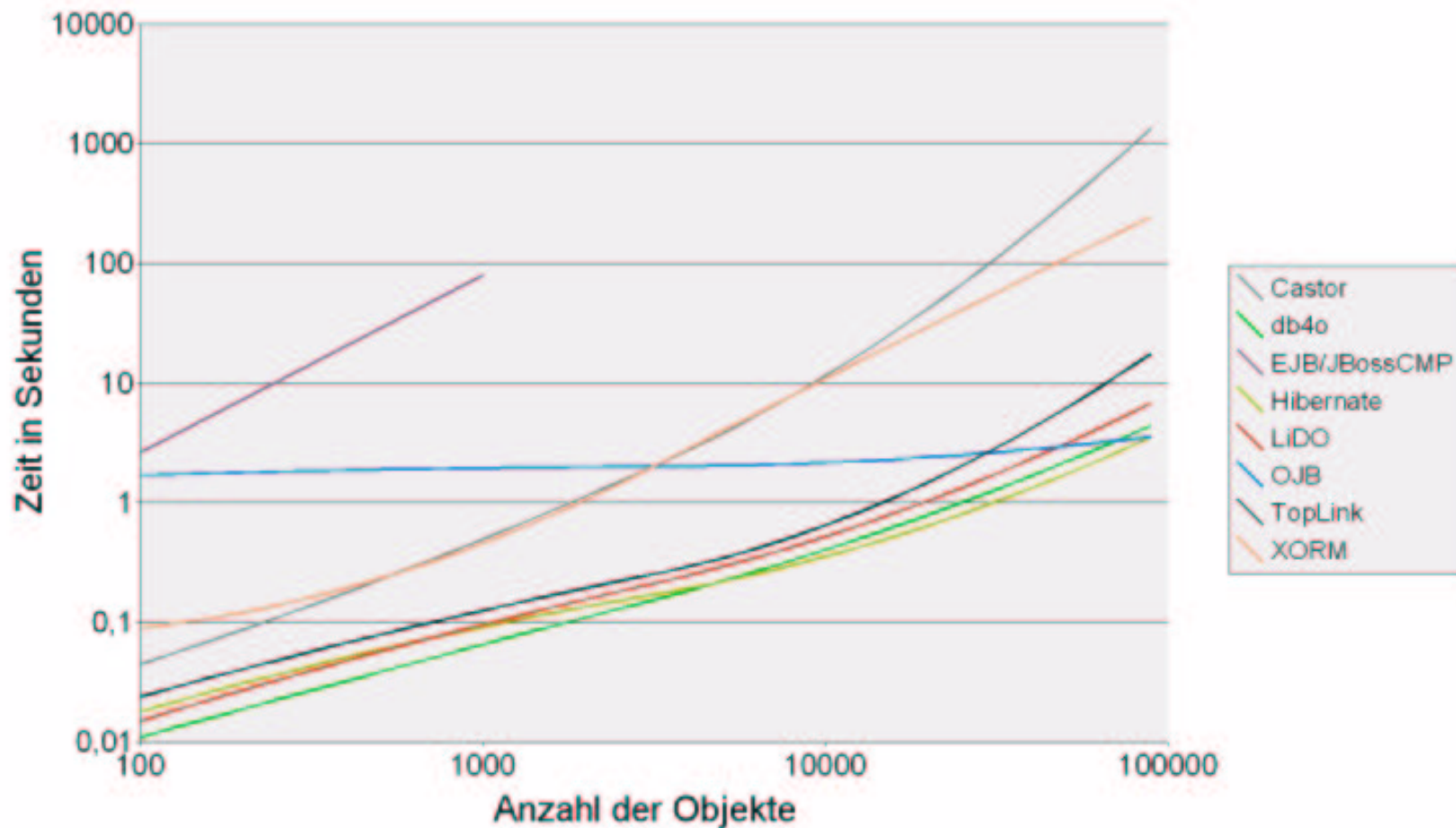


Performance: Lazy Loading





Performance: vollständiges Laden





Fazit

- relationale Datenbanken sind Standard für Datenspeicherung
- mangelnde Verbreitung der OODB führte zu Entwicklung der O/R-Mapper
- Auswahlkriterien
 - ▶ Projektumfeld
 - ▶ Einarbeitungsaufwand
 - ▶ Standard-APIs



Diskussion

?