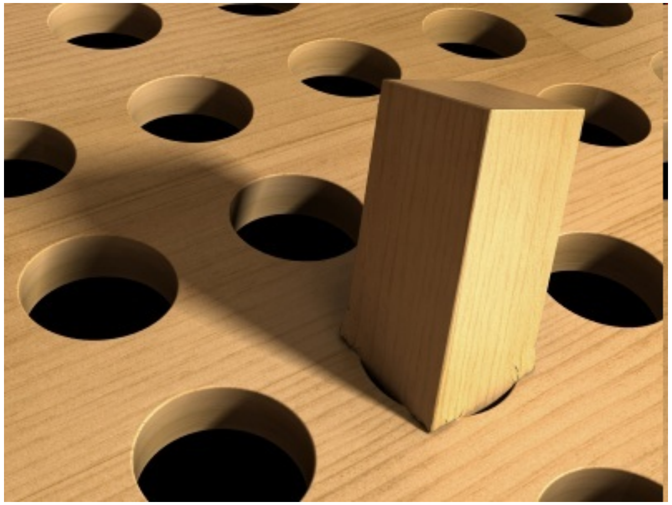
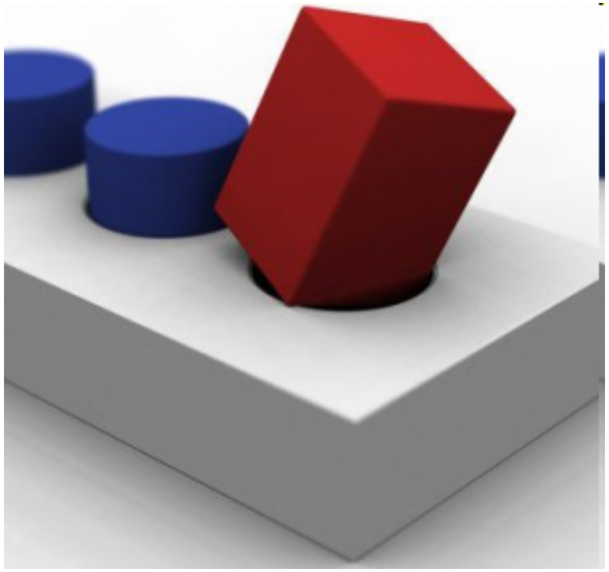
Architecture des Systèmes d' Information

JPA

Java Persistence API (JPA)
Object Relational Mapping (ORM)

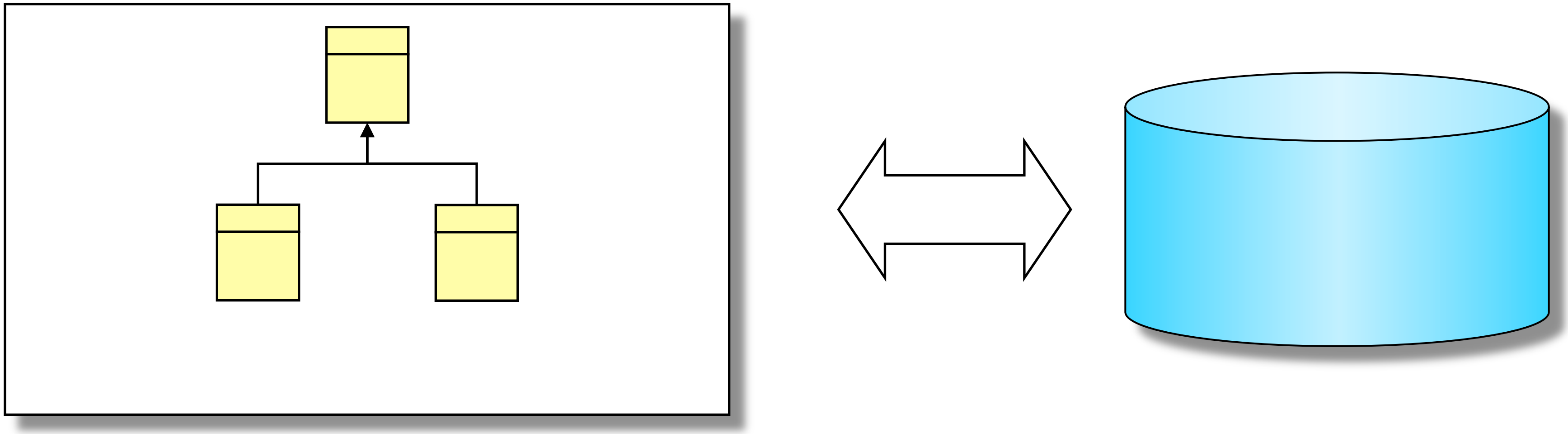
Traduction en cours





Mapping, Round Pegs into Square Holes

Object Persistence

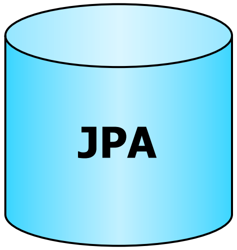


Objects

Disks

EJB persistence component

- The model for persistence and object/relational mapping has been considerably revised and enhanced in the Enterprise JavaBeans 3.0 release.
- An EJB 3.0 entity is a lightweight persistent domain object.



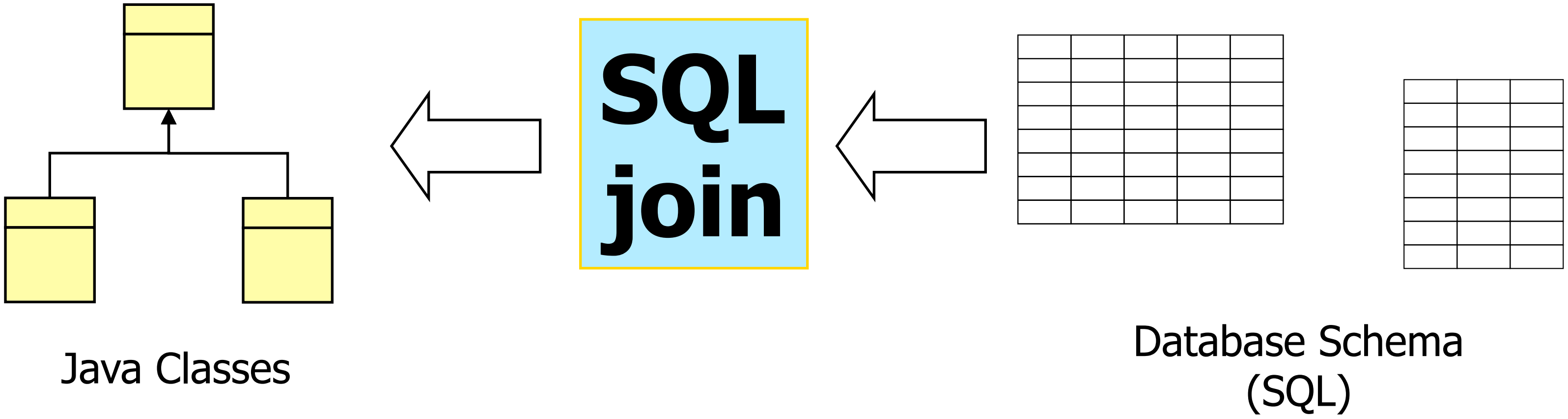
Object Persistence

- Object Relational Mapping (ORM)
  - TOPlink
  - Hybernet
  - EJB

- Object Database Management System (ODBMS)
  - Versant
  - O2
  - GemStone

RDBMS (relational databases mangement system)

- In relational databases relationships are calculated using JOIN operations.
- In ODBMS relationship are objects collections



Book

Id
Value

Book	
ID	**Value**

ODBMS : Native object persistence

- Objects in memory are stored directly on disk.
- No Object break down to relational database fixed type system.
- Object relationship is implemented by Object Data Base mechanisms.
- Disk lay out is object based.
- Tables model is replaced by objects collections
- The Data Definition Language (DDL) is defined in native object language not in SQL.

The problem

- Paradigm mismatch between how data is represented in objects versus relational databases

- Transparent object/relational mapping paradigm

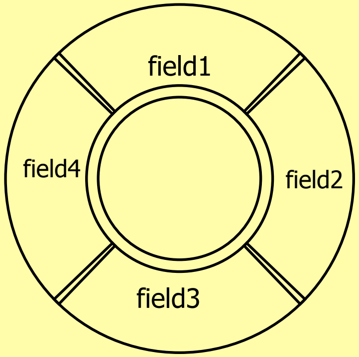
RDBMS and Object Paradigm mismatch

- Cobol
- Fortran
- Pascal
- C

Record, Structure

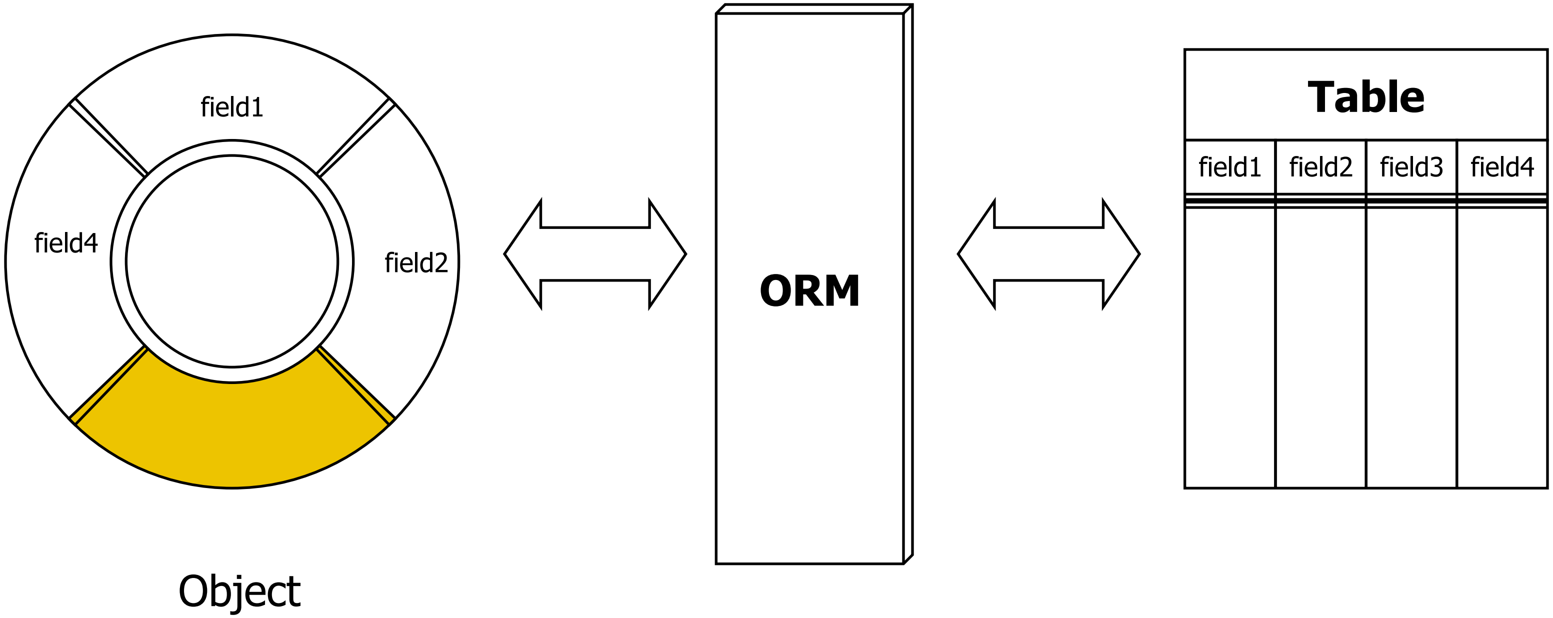
field1	field2	field3	field4

- Smalltalk
- C++
- Java



Object

Object Relational Mapping (ORM)

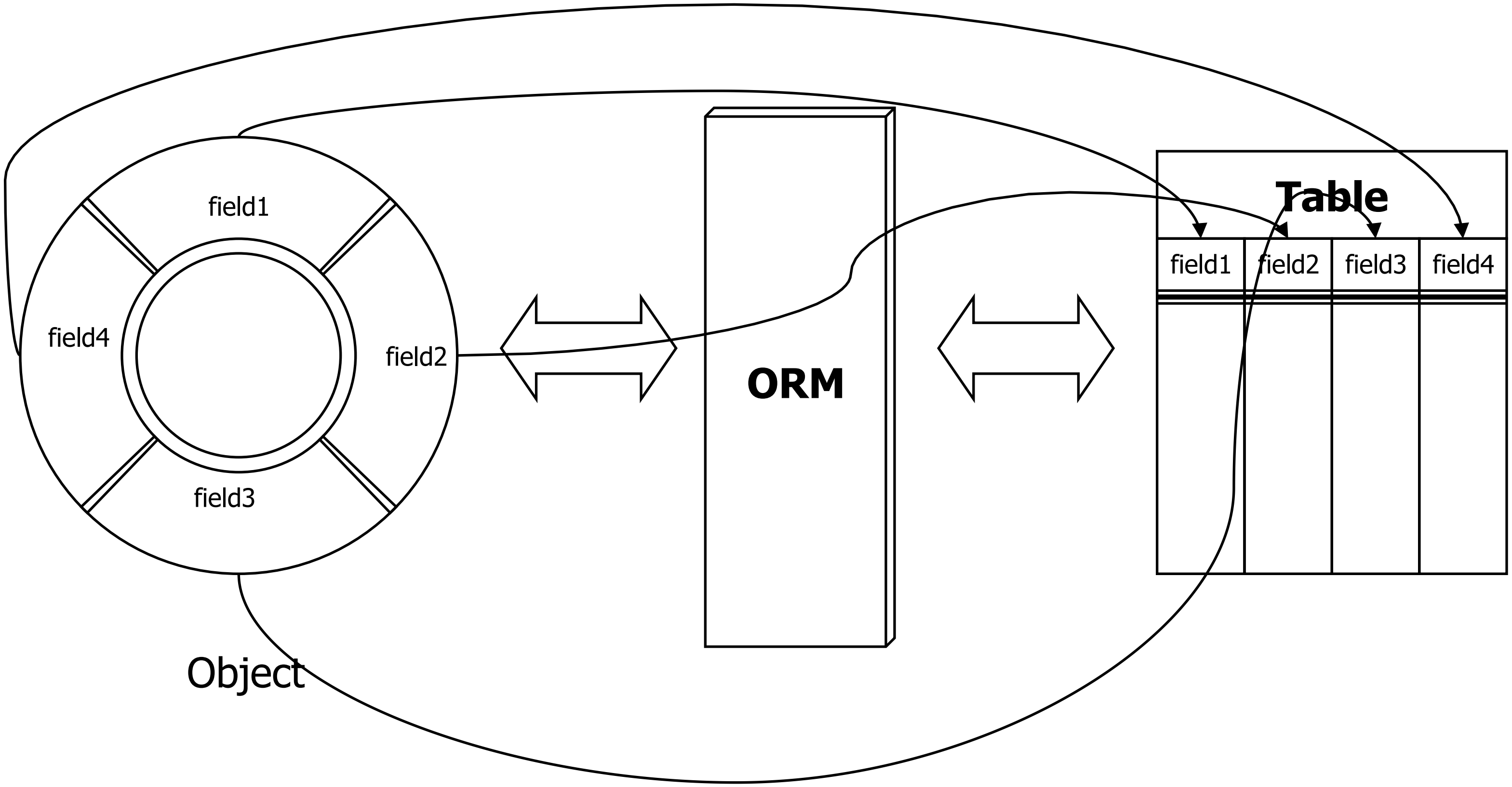


Table

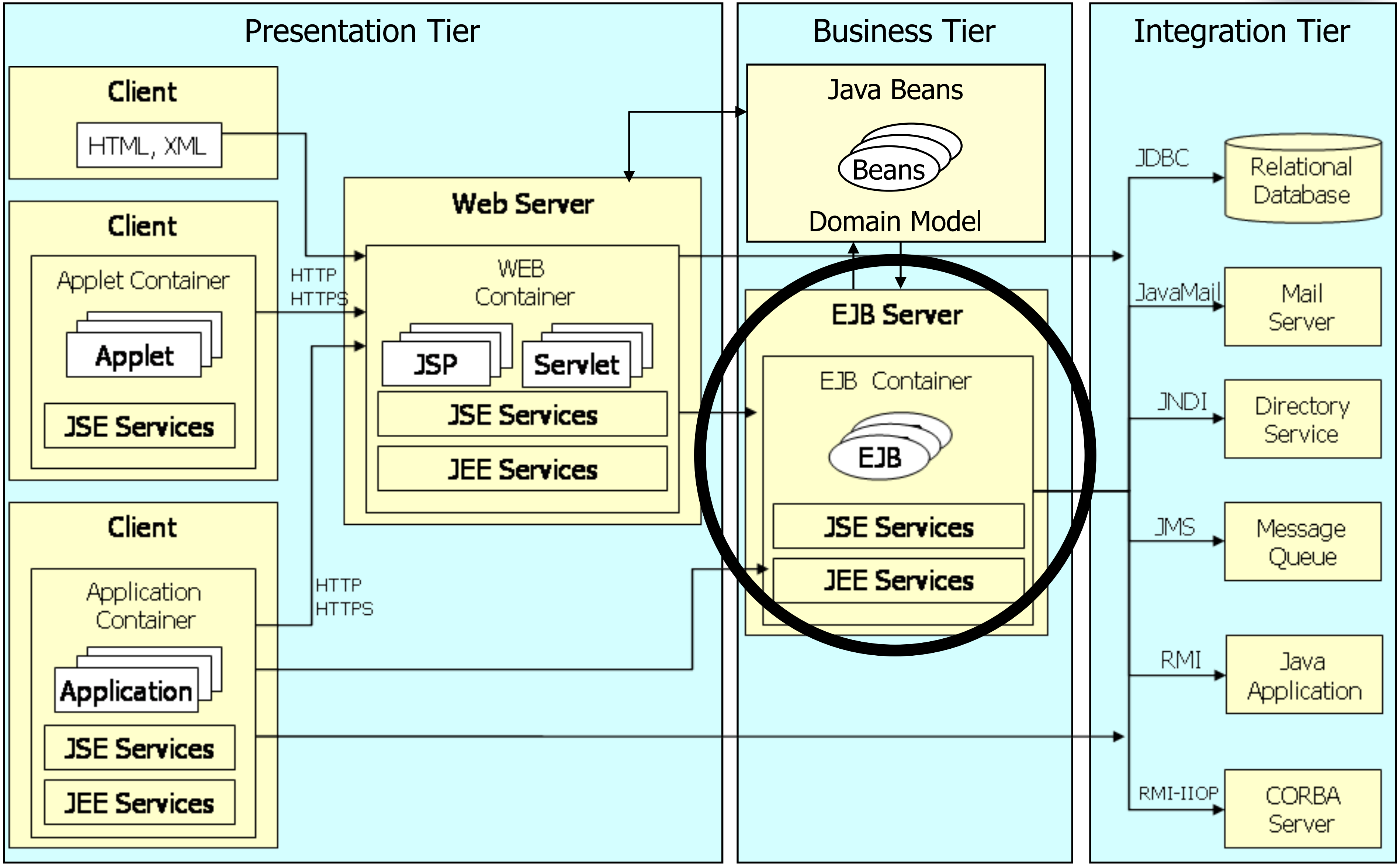
field1	field2	field3	field4

RDBMS seen as ODBMS

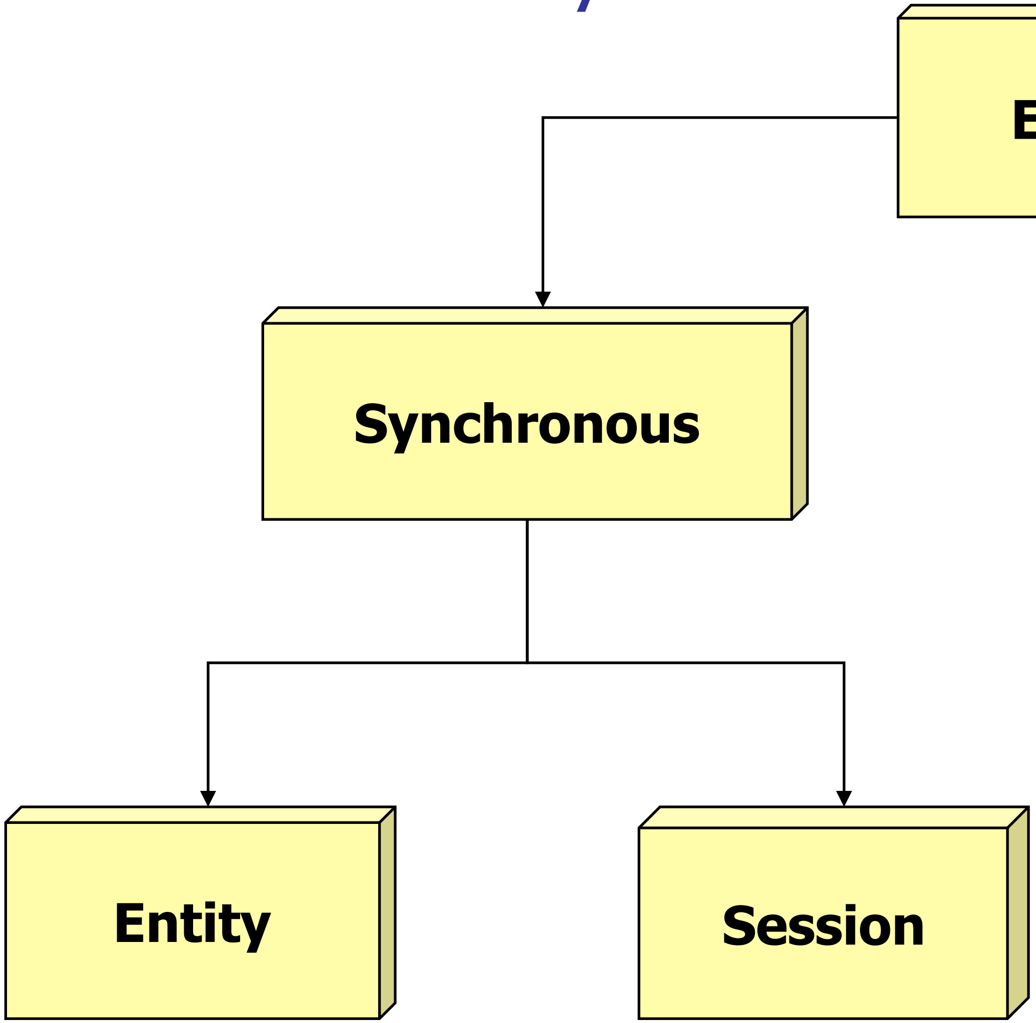
Object Relational Mapping (ORM)



JEE JSE and EJB



EJB Taxonomy



EJB

Synchronous

Asynchronous

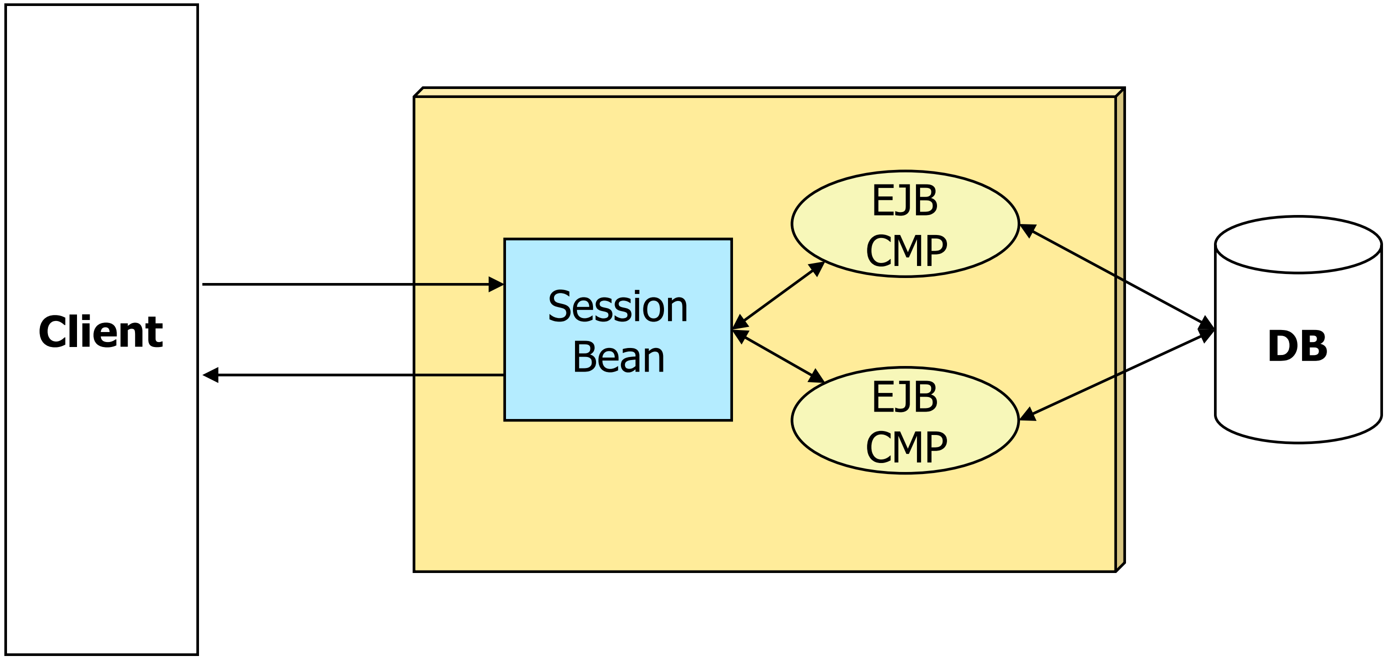
Entity

Session

ORM

Bean

Java Persistence API



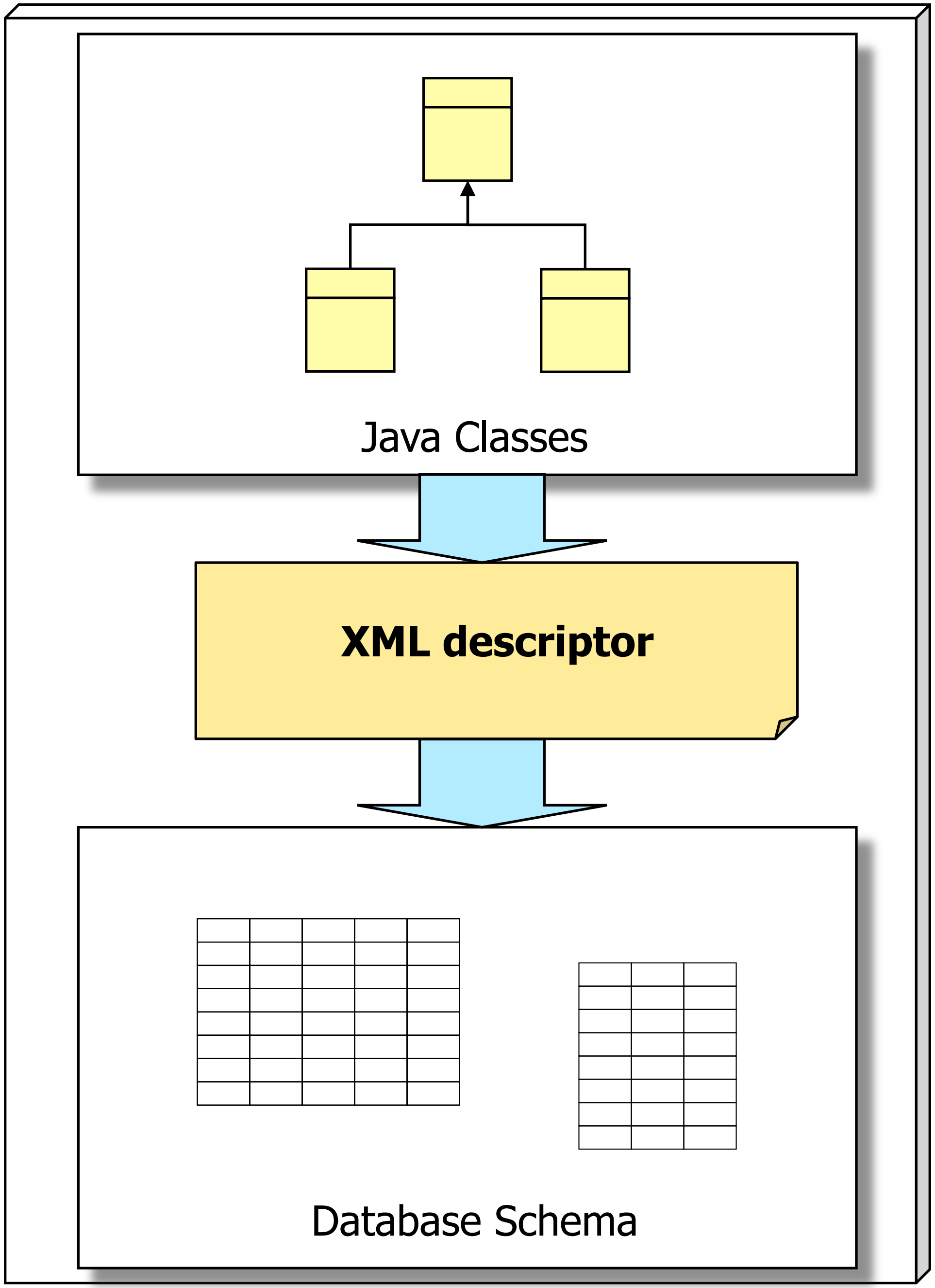
Bean

EJB

- Container-Managed Persistence (CMP)
- Bean-Managed Persistence (BMP)

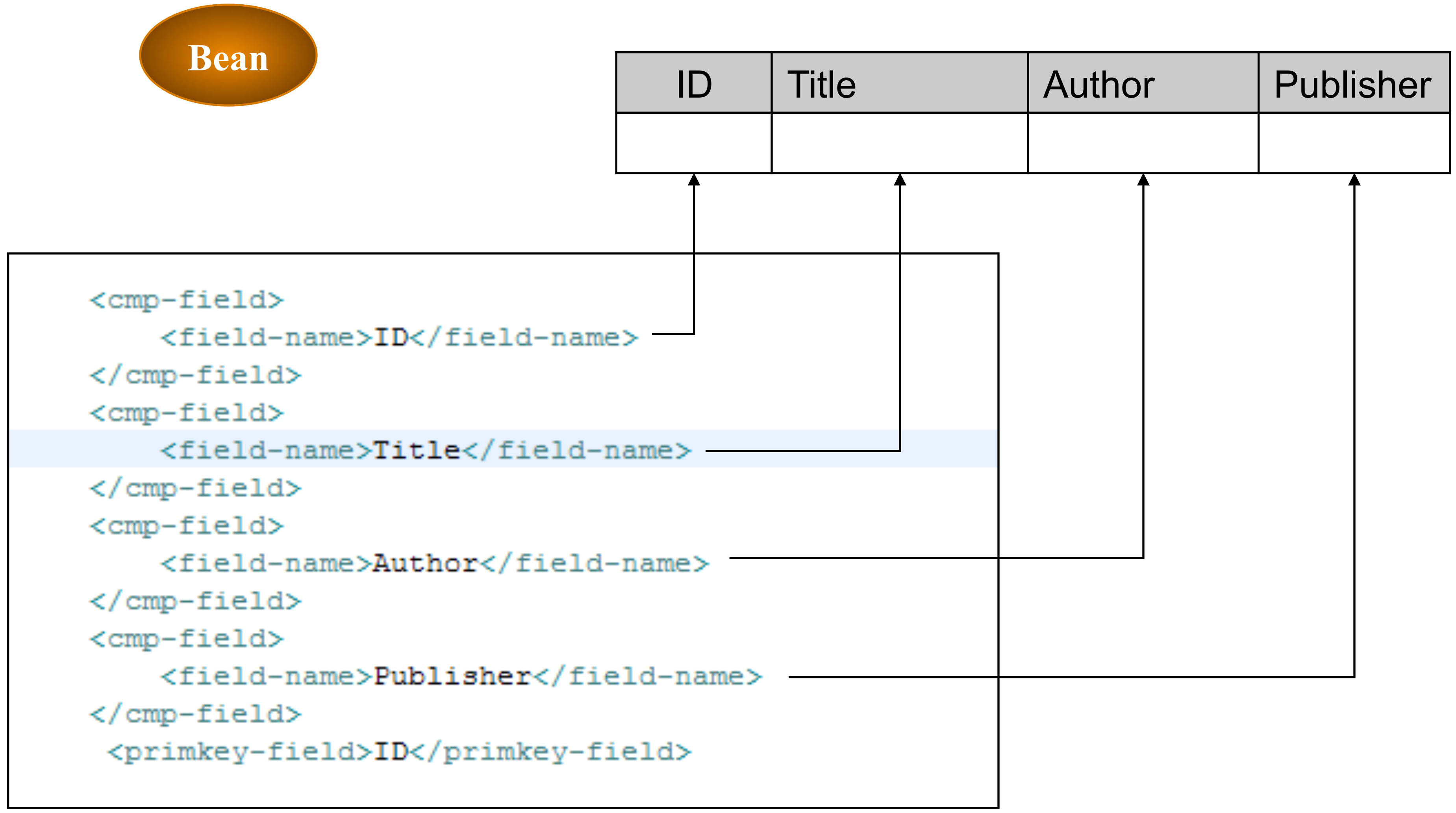
Bean

EJB 2.0 Mapping



Bean

EJB 2. XML Mapping

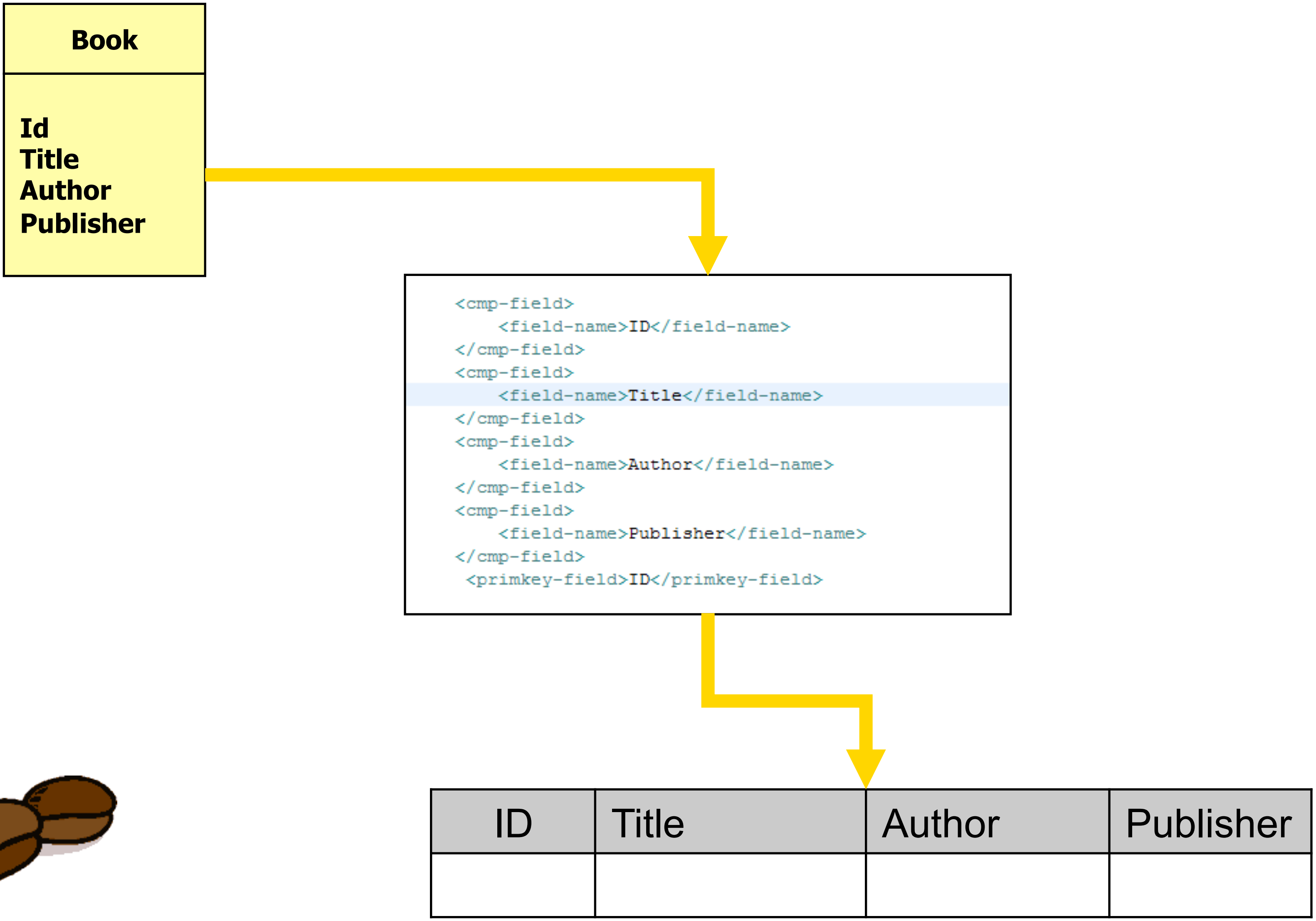


ID	Title	Author	Publisher

```
<cmp-field>
    <field-name>ID</field-name>
</cmp-field>
<cmp-field>
    <field-name>Title</field-name>
</cmp-field>
<cmp-field>
    <field-name>Author</field-name>
</cmp-field>
<cmp-field>
    <field-name>Publisher</field-name>
</cmp-field>
 <primkey-field>ID</primkey-field>
```

EJB 2. XML Mapping

Bean

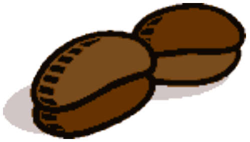


Book

Id
Title
Author
Publisher

```
<cmp-field>
    <field-name>ID</field-name>
</cmp-field>
<cmp-field>
    <field-name>Title</field-name>
</cmp-field>
<cmp-field>
    <field-name>Author</field-name>
</cmp-field>
<cmp-field>
    <field-name>Publisher</field-name>
</cmp-field>
 <primkey-field>ID</primkey-field>
```

ID	Title	Author	Publisher

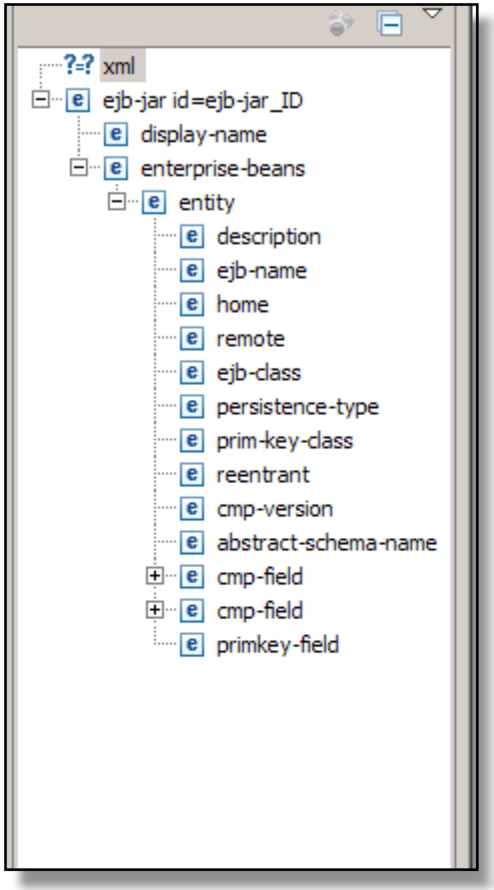


ejb-jar.xml

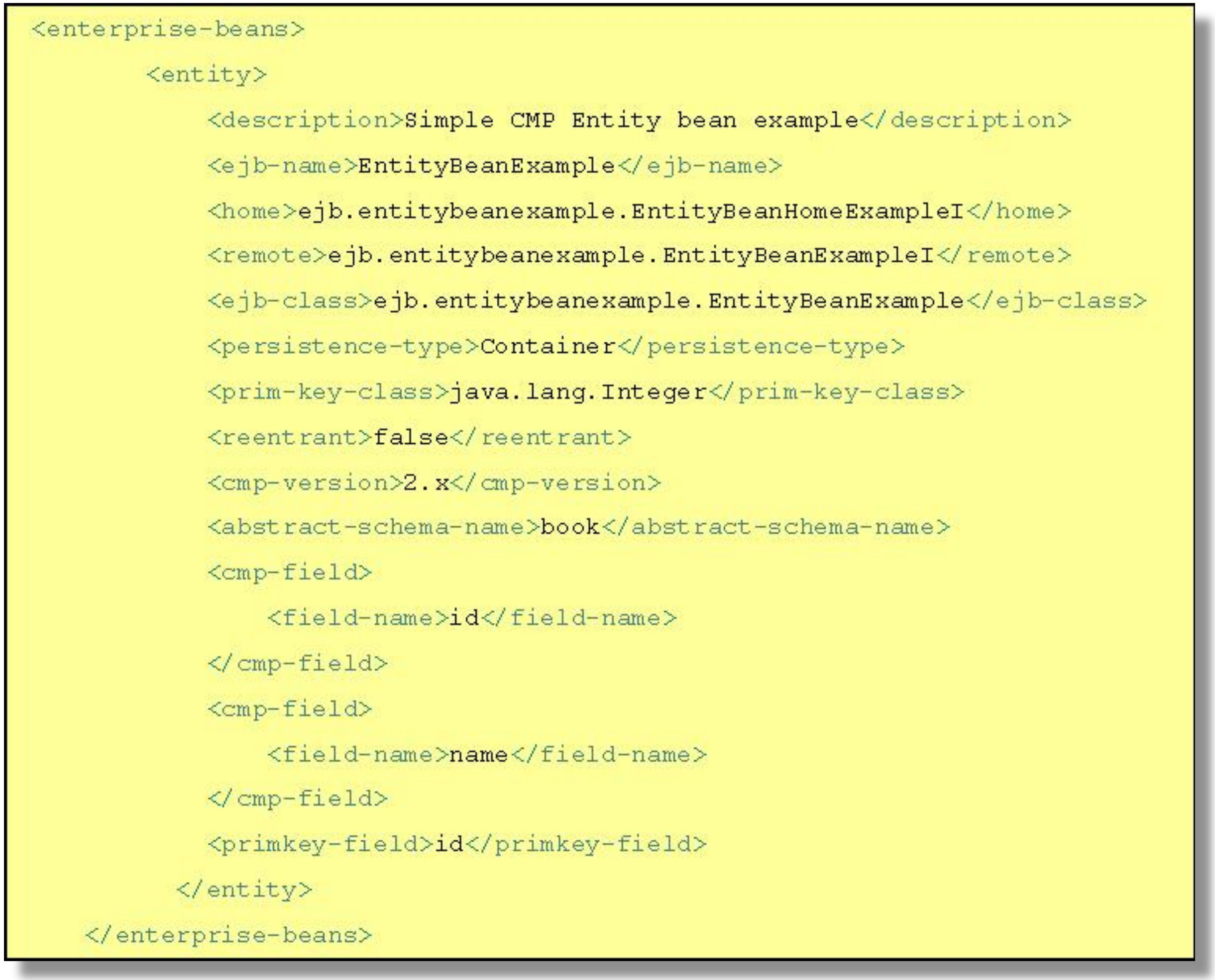
```xml
<enterprise-beans>
        <entity>
            <description>Simple CMP Entity bean example</description>
            <ejb-name>EntityBeanExample</ejb-name>
            <home>ejb.entitybeanexample.EntityBeanHomeExampleI</home>
            <remote>ejb.entitybeanexample.EntityBeanExampleI</remote>
            <ejb-class>ejb.entitybeanexample.EntityBeanExample</ejb-class>
            <persistence-type>Container</persistence-type>
            <prim-key-class>java.lang.Integer</prim-key-class>
            <reentrant>false</reentrant>
            <cmp-version>2.x</cmp-version>
            <abstract-schema-name>book</abstract-schema-name>
            <cmp-field>
                <field-name>id</field-name>
            </cmp-field>
            <cmp-field>
                <field-name>name</field-name>
            </cmp-field>
            <primkey-field>id</primkey-field>
          </entity>
    </enterprise-beans>
```

Bean

Eclipse



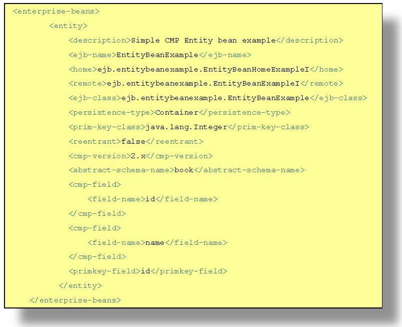
```
?=? xml
ejb-jar id=ejb-jar_ID
    display-name
    enterprise-beans
        entity
            description
            ejb-name
            home
            remote
            ejb-class
            persistence-type
            prim-key-class
            reentrant
            cmp-version
            abstract-schema-name
            cmp-field
            cmp-field
            primkey-field
```



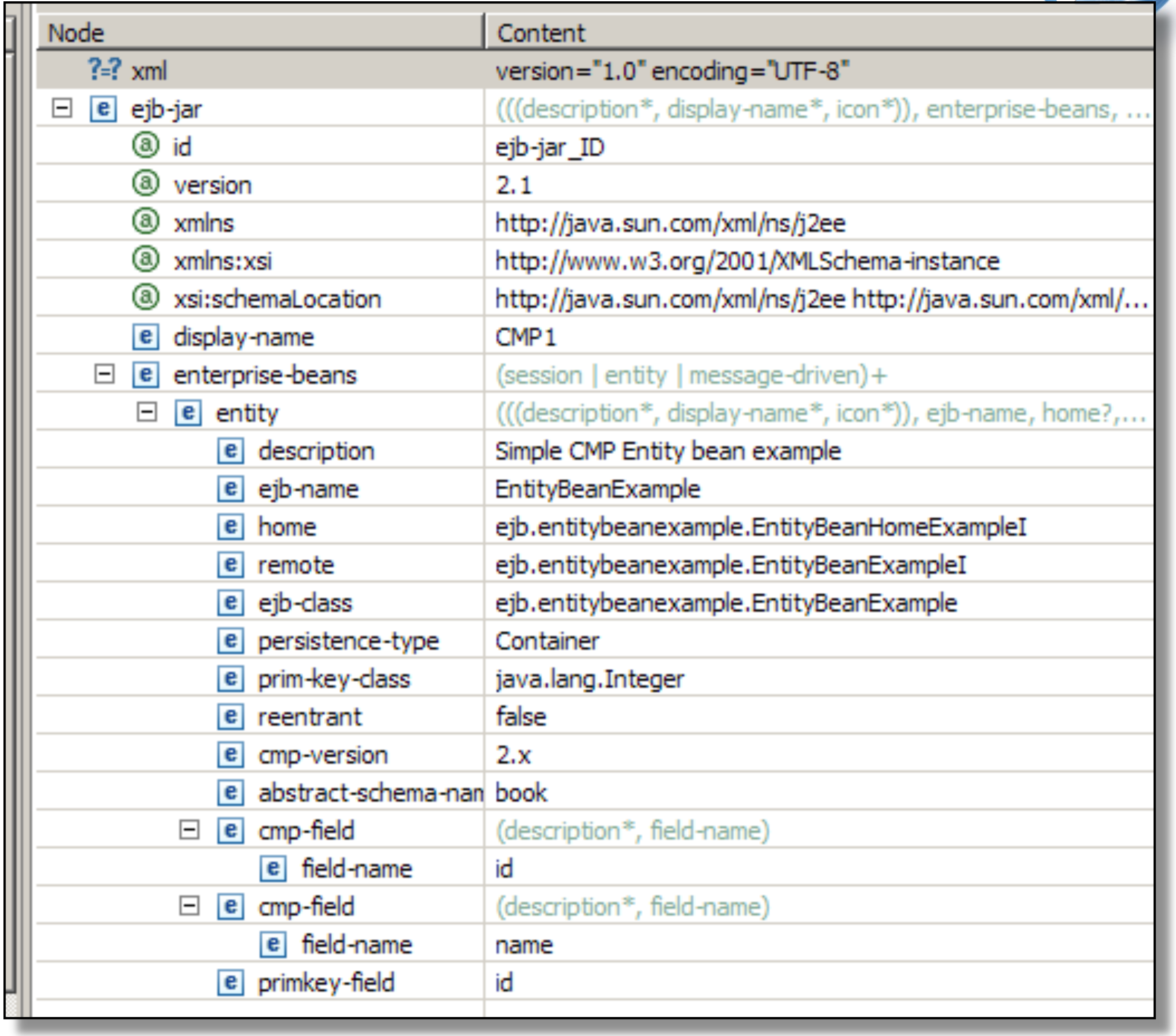
```
<enterprise-beans>
       <entity>
           <description>Simple CMP Entity bean example</description>
           <ejb-name>EntityBeanExample</ejb-name>
           <home>ejb.entitybeanexample.EntityBeanHomeExampleI</home>
           <remote>ejb.entitybeanexample.EntityBeanExampleI</remote>
           <ejb-class>ejb.entitybeanexample.EntityBeanExample</ejb-class>
           <persistence-type>Container</persistence-type>
           <prim-key-class>java.lang.Integer</prim-key-class>
           <reentrant>false</reentrant>
           <cmp-version>2.x</cmp-version>
           <abstract-schema-name>book</abstract-schema-name>
           <cmp-field>
               <field-name>id</field-name>
           </cmp-field>
           <cmp-field>
               <field-name>name</field-name>
           </cmp-field>
           <primkey-field>id</primkey-field>
         </entity>
   </enterprise-beans>
```

Eclipse

Bean



```
<enterprise-beans>
      <entity>
          <description>Simple CMP Entity bean example</description>
          <ejb-name>EntityBeanExample</ejb-name>
          <home>ejb.entitybeanexample.EntityBeanHomeExampleI</home>
          <remote>ejb.entitybeanexample.EntityBeanExampleI</remote>
          <ejb-class>ejb.entitybeanexample.EntityBeanExample</ejb-class>
          <persistence-type>Container</persistence-type>
          <prim-key-class>java.lang.Integer</prim-key-class>
          <reentrant>false</reentrant>
          <cmp-version>2.x</cmp-version>
          <abstract-schema-name>book</abstract-schema-name>
          <cmp-field>
              <field-name>id</field-name>
          </cmp-field>
          <cmp-field>
              <field-name>name</field-name>
          </cmp-field>
          <primkey-field>id</primkey-field>
        </entity>
   </enterprise-beans>
```



Node	Content
?=? xml	version="1.0" encoding="UTF-8"
⊟ ejb-jar	(((description*, display-name*, icon*)), enterprise-beans, …
id	ejb-jar_ID
version	2.1
xmlns	http://java.sun.com/xml/ns/j2ee
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/…
display-name	CMP1
⊟ enterprise-beans	(session | entity | message-driven)+
⊟ entity	(((description*, display-name*, icon*)), ejb-name, home?,…
description	Simple CMP Entity bean example
ejb-name	EntityBeanExample
home	ejb.entitybeanexample.EntityBeanHomeExampleI
remote	ejb.entitybeanexample.EntityBeanExampleI
ejb-class	ejb.entitybeanexample.EntityBeanExample
persistence-type	Container
prim-key-class	java.lang.Integer
reentrant	false
cmp-version	2.x
abstract-schema-nan	book
⊟ cmp-field	(description*, field-name)
field-name	id
⊟ cmp-field	(description*, field-name)
field-name	name
primkey-field	id

First ORM : TOPlink

- TOPLink is an object-relational mapping tool that has versions in Smalltalk and Java.
- The "TOP" in TopLink is an acronym for "The Object People".

- Now EclipseLink
  - Open source version of TopLink
  - Reference implementation of the EJB 3.0 Java Persistence API (JPA)

Hibernate

- Started in 2001 as an alternative to using EJB2-style entity beans.
- To offer better persistence capabilities than offered by EJB2.
- Based on Xdoclet (attribute Oriented Programming).
- Main influence for JPA.

JPA

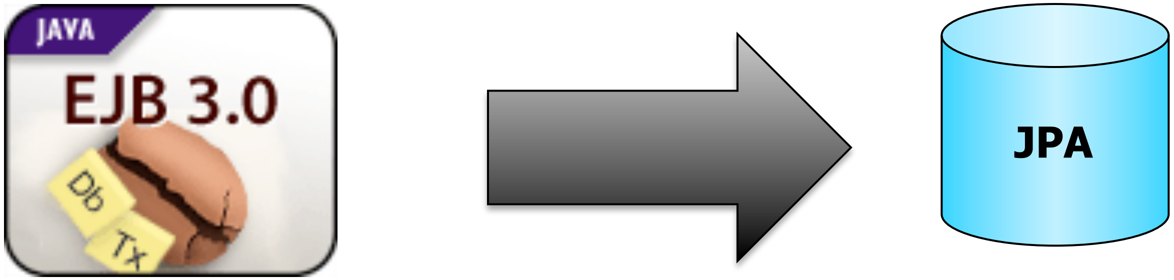
Xdoclet : attribute Oriented Programming

- XDoclet is an open source code generation engine.
- It enables *"Attribute-Oriented Programming"* for java.
- Annotation ancestor
- Based on special JavaDoc tags.

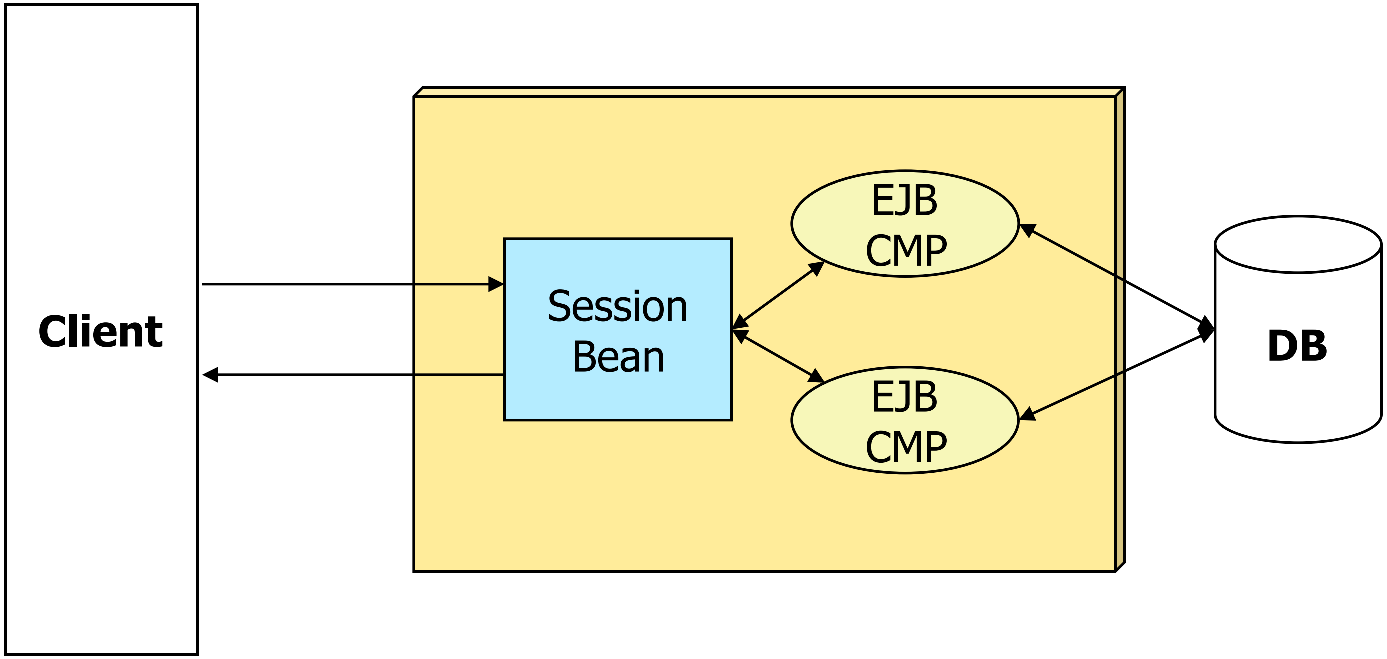
```
/**
    * @hibernate.id
    *  generator-class="native"
    *  column="CAT_ID"
    */
```

Persistance Java API

- JDO : Java Data Objects
  - XML
  - Web service
  - ORM
- JPA : Java Persistence API
  - ORM only

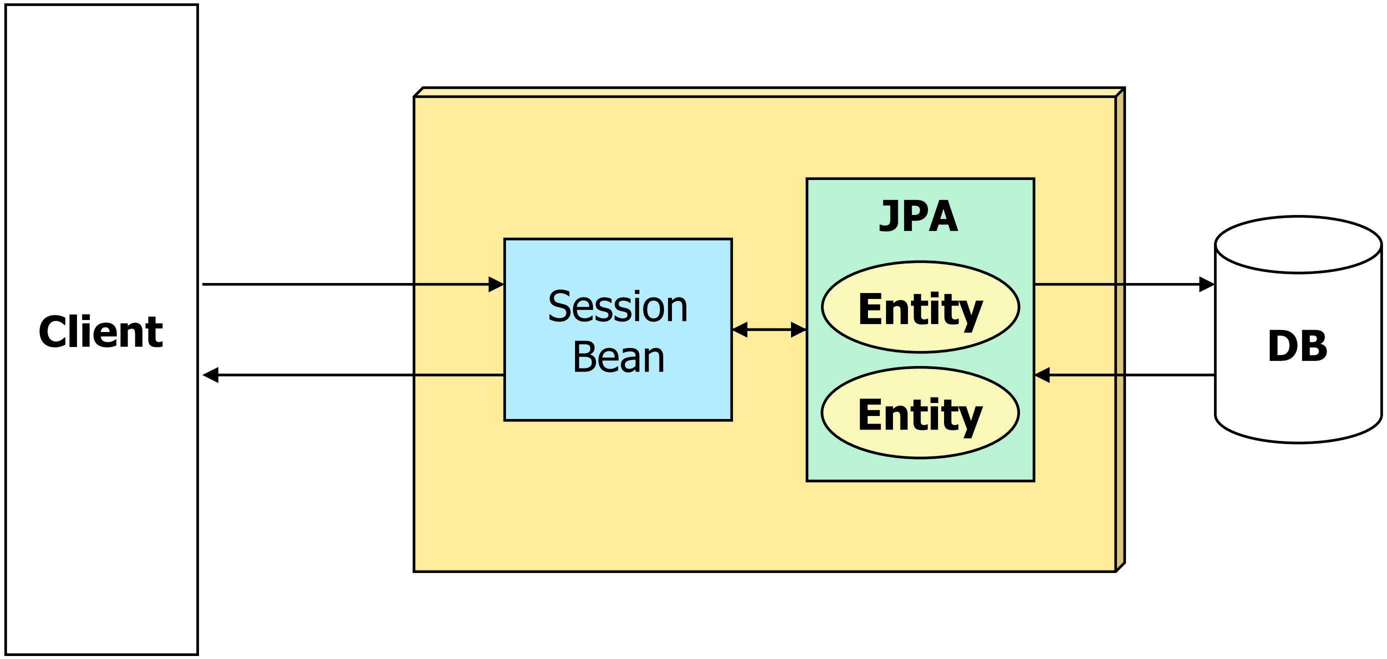


Java Persistence API



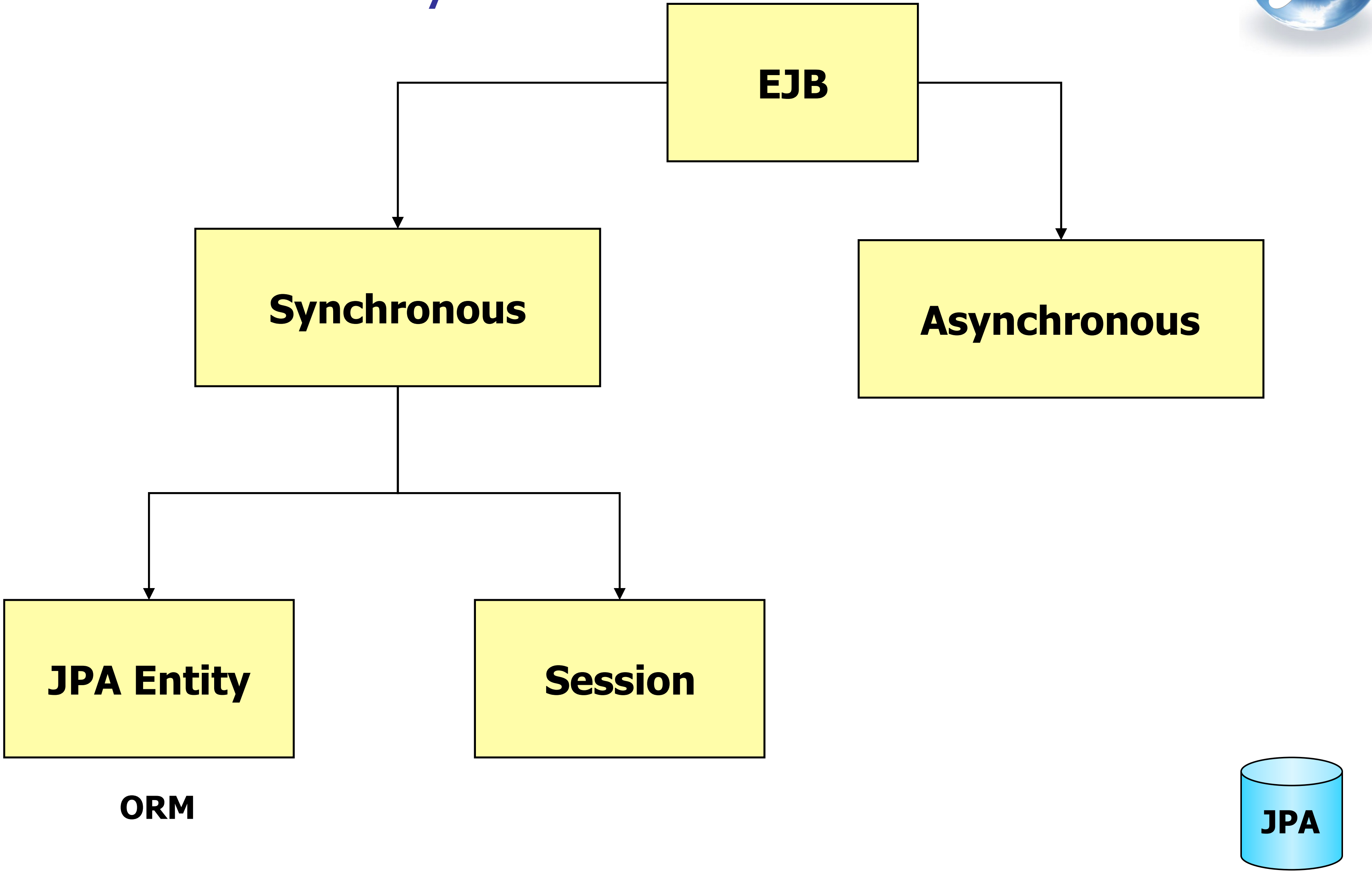
Bean

Java Persistence API



EJB CMP Replaced By JPA

EJB Taxonomy



EJB

Synchronous

Asynchronous

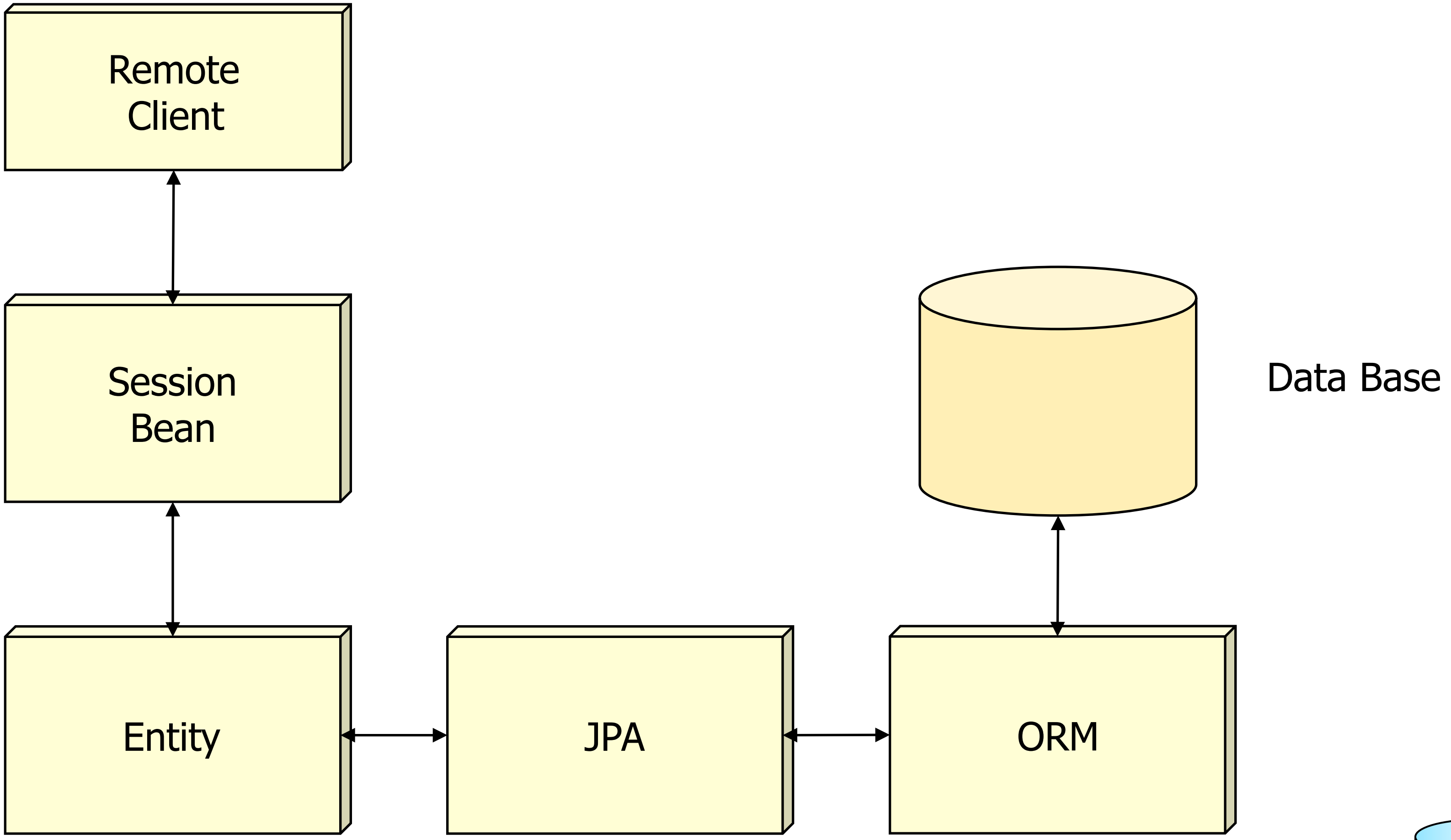
JPA Entity

Session

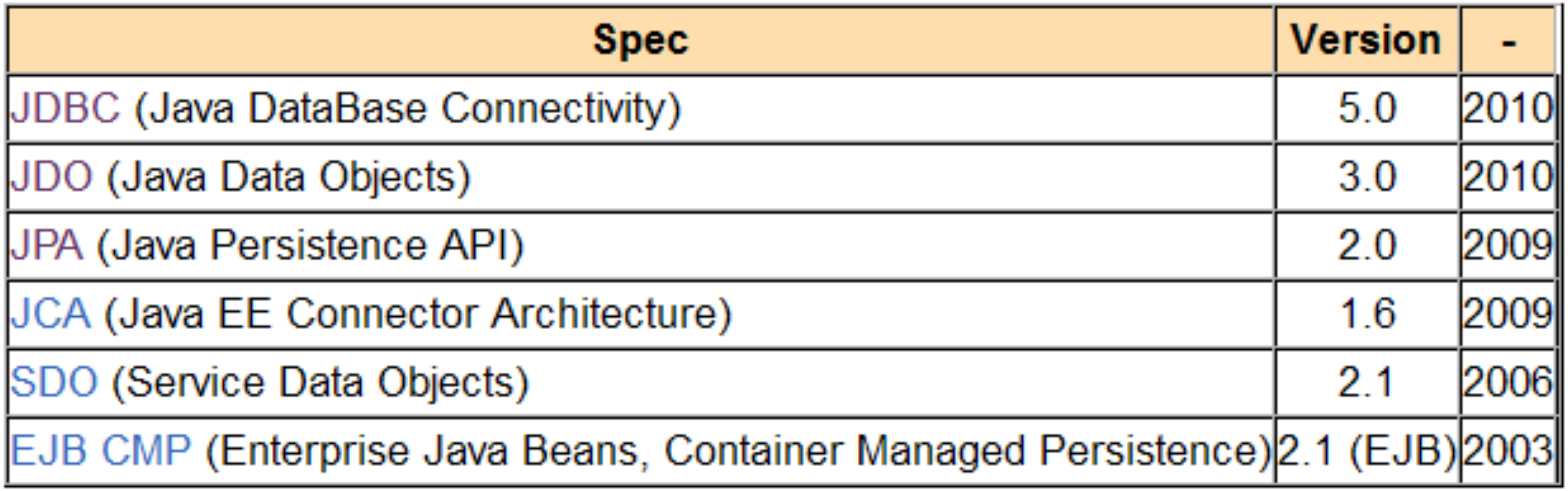
ORM

JPA

EJB and Persistance API



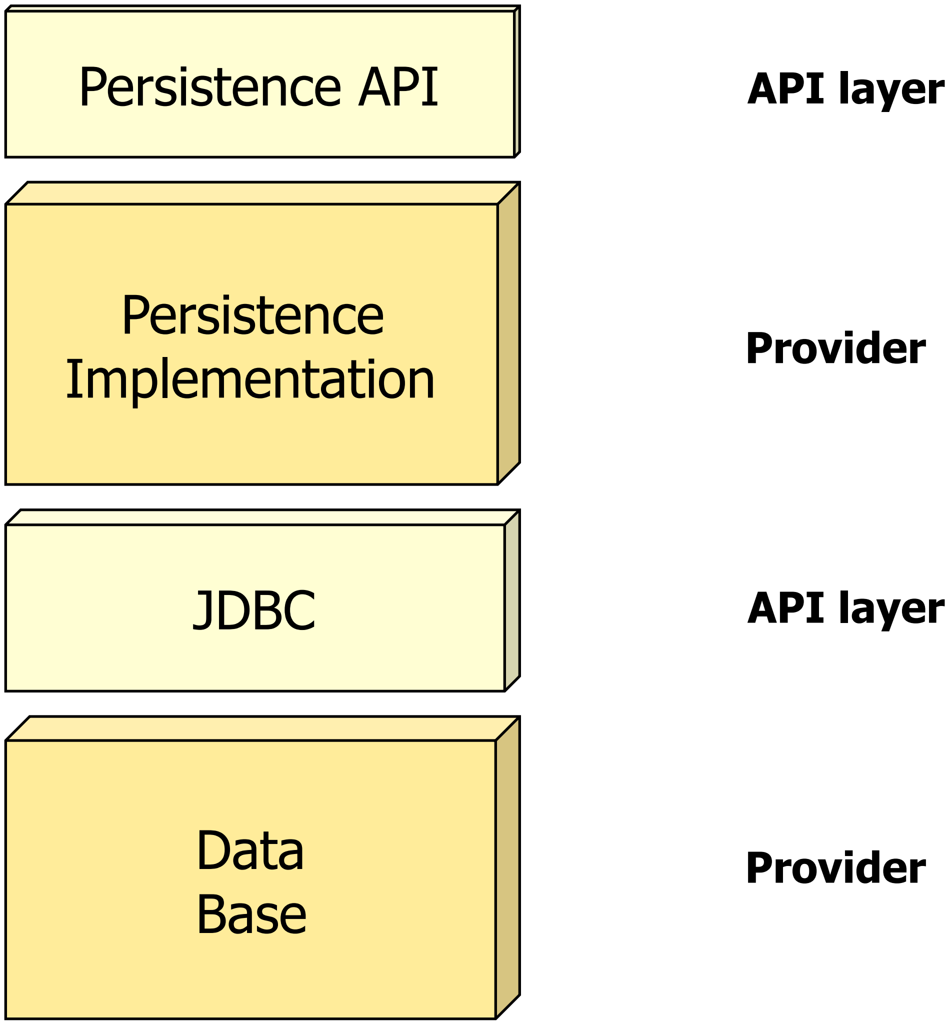
Persistence API specifications



Spec	Version	-
JDBC (Java DataBase Connectivity)	5.0	2010
JDO (Java Data Objects)	3.0	2010
JPA (Java Persistence API)	2.0	2009
JCA (Java EE Connector Architecture)	1.6	2009
SDO (Service Data Objects)	2.1	2006
EJB CMP (Enterprise Java Beans, Container Managed Persistence)	2.1 (EJB)	2003

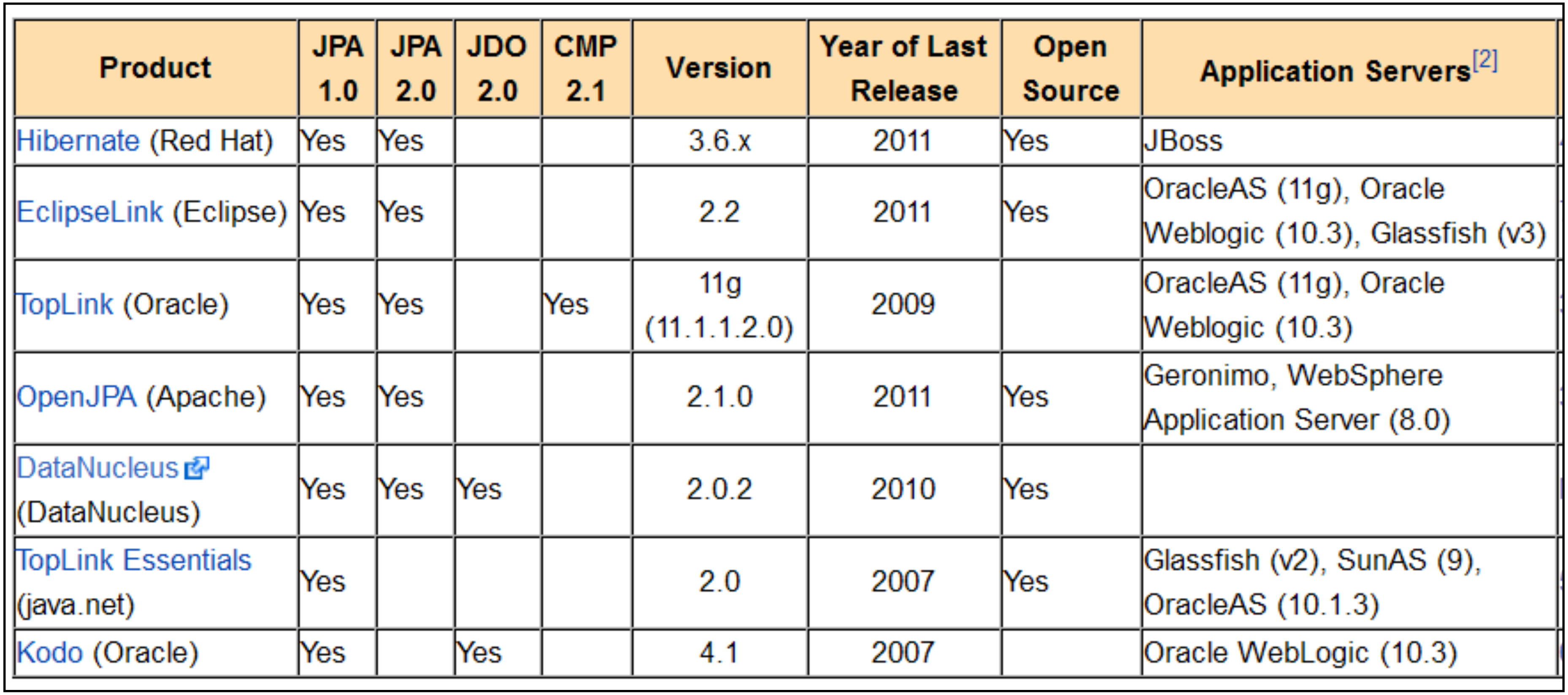
EJB CMP Replaced By JPA

Persistance framework architecture



JPA

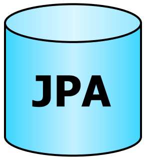
Persistence Products



Product	JPA 1.0	JPA 2.0	JDO 2.0	CMP 2.1	Version	Year of Last Release	Open Source	Application Servers[2]
Hibernate (Red Hat)	Yes	Yes			3.6.x	2011	Yes	JBoss
EclipseLink (Eclipse)	Yes	Yes			2.2	2011	Yes	OracleAS (11g), Oracle Weblogic (10.3), Glassfish (v3)
TopLink (Oracle)	Yes	Yes		Yes	11g (11.1.1.2.0)	2009		OracleAS (11g), Oracle Weblogic (10.3)
OpenJPA (Apache)	Yes	Yes			2.1.0	2011	Yes	Geronimo, WebSphere Application Server (8.0)
DataNucleus (DataNucleus)	Yes	Yes	Yes		2.0.2	2010	Yes	
TopLink Essentials (java.net)	Yes				2.0	2007	Yes	Glassfish (v2), SunAS (9), OracleAS (10.1.3)
Kodo (Oracle)	Yes		Yes		4.1	2007		Oracle WebLogic (10.3)

Java Persistence API (JPA).

- Java Persistence API (JPA).
- JPA uses the javax.persistence package.
- first specified in a separate document within the EJB3, but was later moved to its own specification.
- javax.persistence do **not** require an EJB container.
  - Java SE environment (“Bootstrap API”)
  - Java EE environment



JPA vs JDO

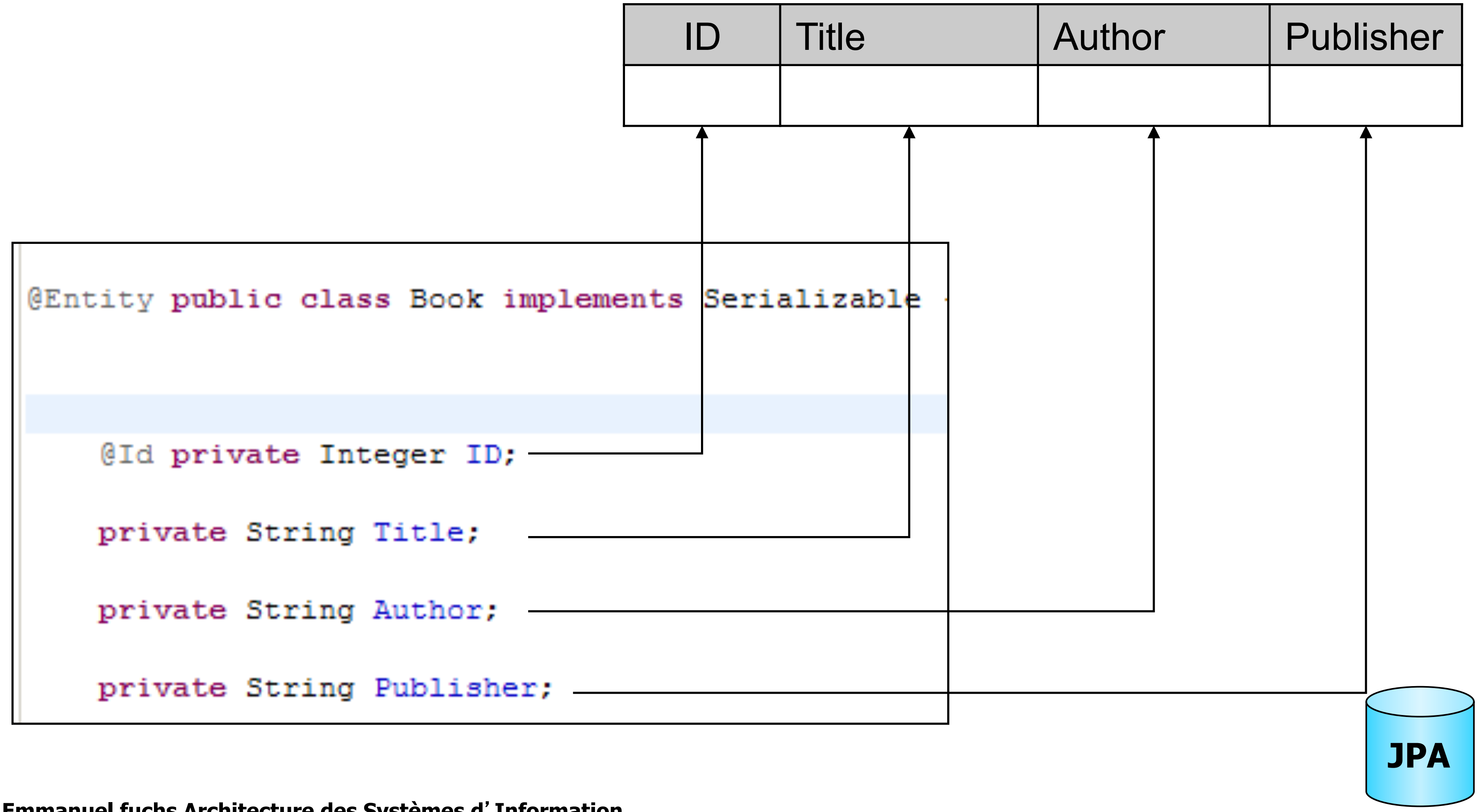
- JDO is agnostic to the technology of the underlying datastore.
- JPA is targeted to RDBMS datastores.

JDO™

Java™ Data Objects

Java™ Data Objects

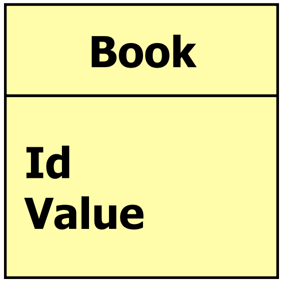
JPA Mapping Annotation



ID	Title	Author	Publisher

```
@Entity public class Book implements Serializable


    @Id private Integer ID;

    private String Title;

    private String Author;

    private String Publisher;
```

HelloJPA entity bean



```
package helloJPA;

import java.io.Serializable;
import java.lang.Integer;
import java.lang.String;
import javax.persistence.*;

@Entity
public class HelloJPAentityBean implements Serializable {

            @Id
            @GeneratedValue(strategy = GenerationType.AUTO)
            private Integer ID;
            private String value;
            private static final long serialVersionUID = 1L;

            public HelloJPAentityBean() {
                        super();
            }

            public Integer getID() {
                        return this.ID;
            }

            public void setID(Integer ID) {
                        this.ID = ID;
            }

            public String getValue() {
                        return this.value;
            }

            public void getValue(String Lesson) {
                        this.value = Lesson;
            }
}
```