
ORACLE®

ORACLE

Java Persistence and More

Michael Bräuer
Oracle Deutschland B.V. & Co. KG

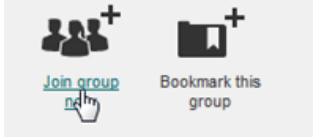
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Java Application Server Plattform Community

Eine Community von ORACLE für Kunden, Partner und Interessierte

Code Camps
Demos
WebLogic Server
GlassFish Server
Community Treffen
Java EE
JSRs
Vorträge
Serverseitige Entwicklung
Administration
Wissensaustausch

Registrierung: https://www.xing.com/net/oraclejavaappserver
Blog: http://fmwtech.wordpress.com
Ansprechpartner: michael.braeuer@oracle.com
peter.doschkinow@oracle.com



Java EE Hackaton

Tagesveranstaltung

Die Agenda

9:00–12:00
Java EE 7 und GlassFish Server 4 – Schnelleinstieg
WebLogic Server 12.1.2 – Product Update

13:00–16:00 Hands–On
HTML5 Coding mit JavaFX und Java EE 7
WebLogic Server 12.1.2 – New Features praktisch ausprobieren

16:00–16:30 Diskussion Java Application Server Plattform Community – Nächste Schritte und Veranstaltungen

Wo und Wann?
Nach einer Abstimmung über den Ort und Termin werden wir das Ergebnis allen Interessenten am 26.11.2013 mitteilen. Bitte selektieren Sie bis zum 25.11.2013 die Termine und Orte, die für Sie akzeptabel wären, unter folgendem Link, indem Sie im Namen–Feld auch Ihre Email–Adresse, durch Komma getrennt, angeben.

Anmeldung
Nach Bekanntgabe des Ortes und Datums melden Sie sich bitte per Mail bei Barbara.Frank@oracle.com an. Die Veranstaltung ist kostenlos. Sie müssen kein Mitglied der Community sein, können ihr jedoch hier beitreten. Dann werden Sie regelmäßig über alle Aktivitäten der Community informiert.

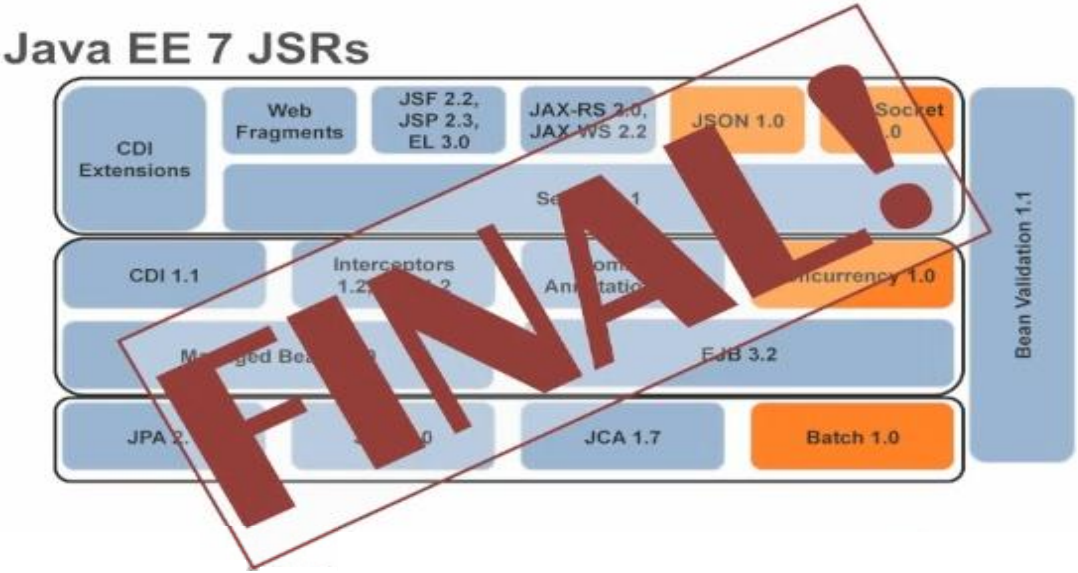
http://fmwtech.wordpress.com/2013/10/23/java-ee-hackaton-fur-devops-anmeldung-jetzt/

Agenda

- Java EE 7: JPA 2.1
- Trends of Java Persistence (with EclipseLink)

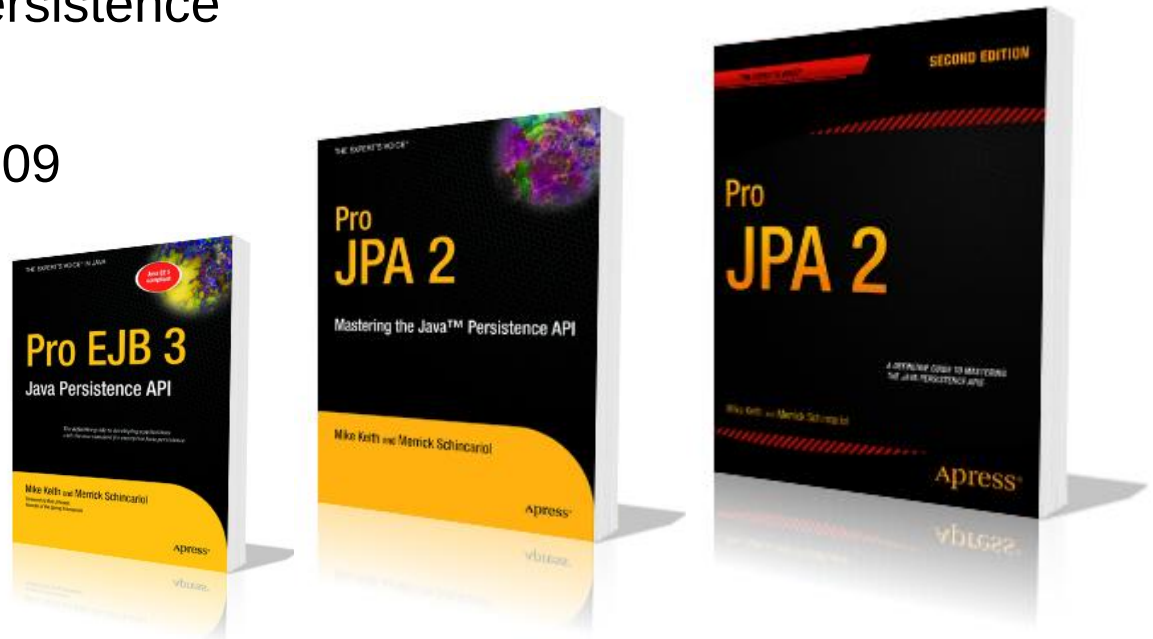
Java EE 7

JPA 2.1

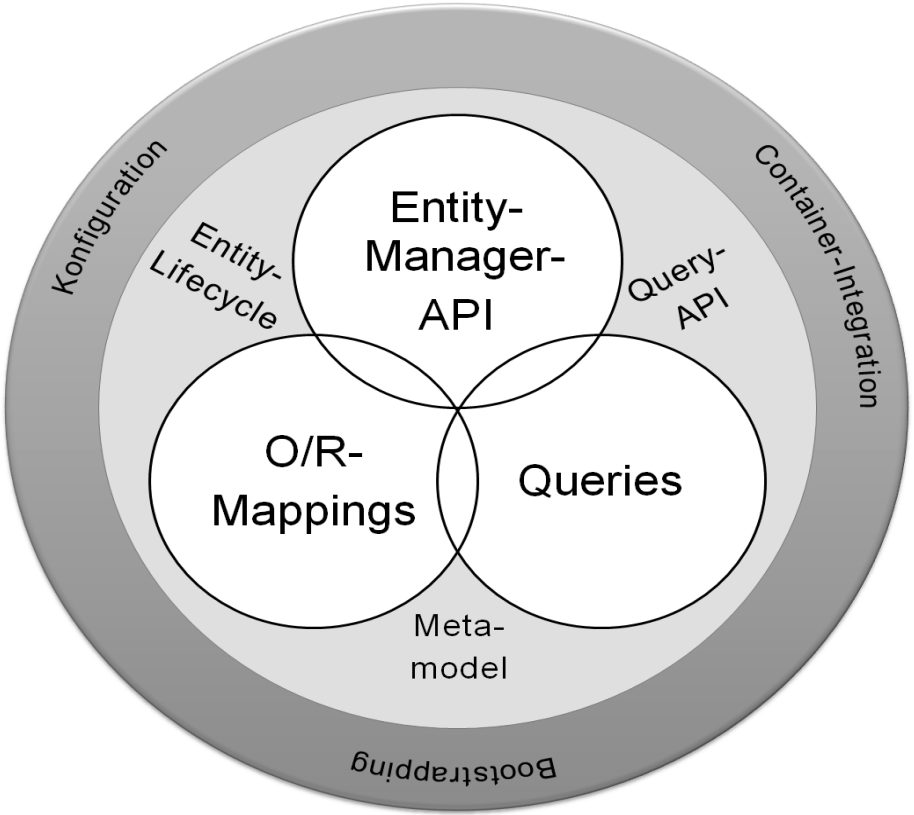


Java Persistence API

- Standardizing Java Persistence
- JPA 1.0: May 2006
- JPA 2.0: December 2009
- JPA 2.1: Juni 2013



Java Persistence API



Java Persistence API 2.1

Highlights

- Schema Generation
  - javax.persistence.schema-generation.* properties
- Unsynchronized Persistence Contexts
- Query Enhancements, e.g.
  - Bulk update/delete using Criteria
  - User-defined functions using FUNCTION
  - Stored Procedure Query, etc.
- Converter
- Entity Graphs

Java Persistence API 2.1

Entity Graphen Ergänzungen

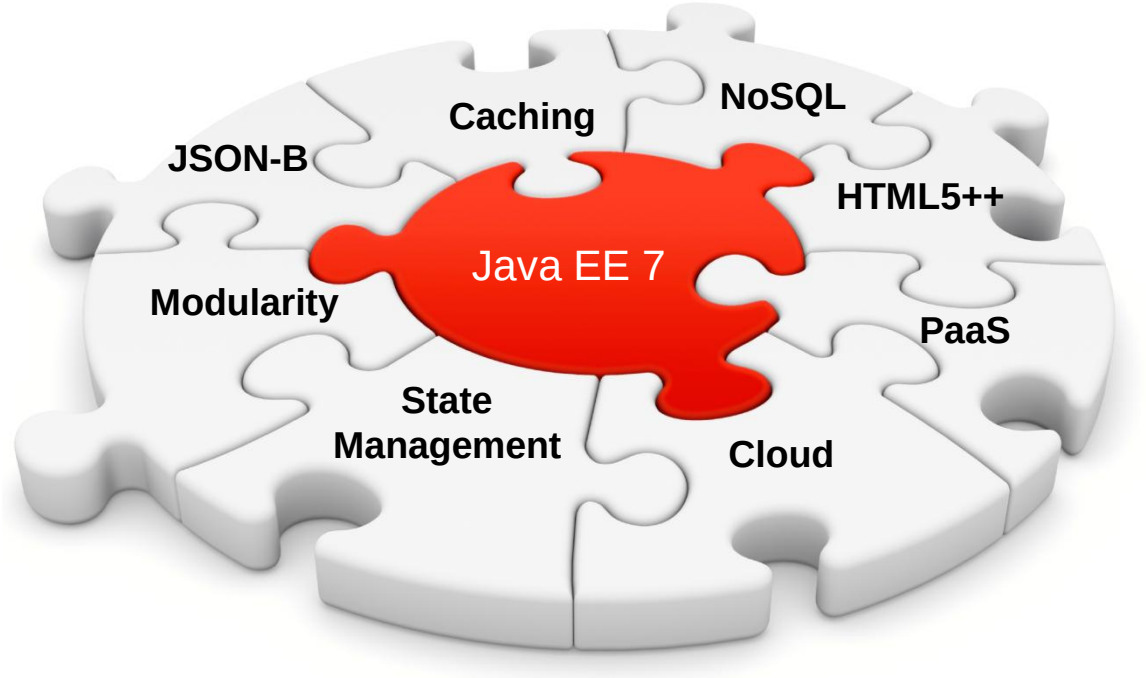
- Nachladen von Attribut-Knoten, die nicht Teil des ursprünglichenEntityGraphen waren: in Eclipselink wird das Attribut bei Bedarf nachgeladen (lazy loading)

Java Persistence API 2.1

Examples

- Look at https://bugs.eclipse.org/bugs/show_bug.cgi?id=415742
- http://wiki.eclipse.org/EclipseLink/Release/2.5/JPA21

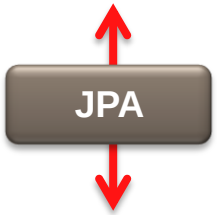
Java EE 8 and Beyond



Java Persistence mit EclipseLink

Customer
id: int name: String creditRating: int

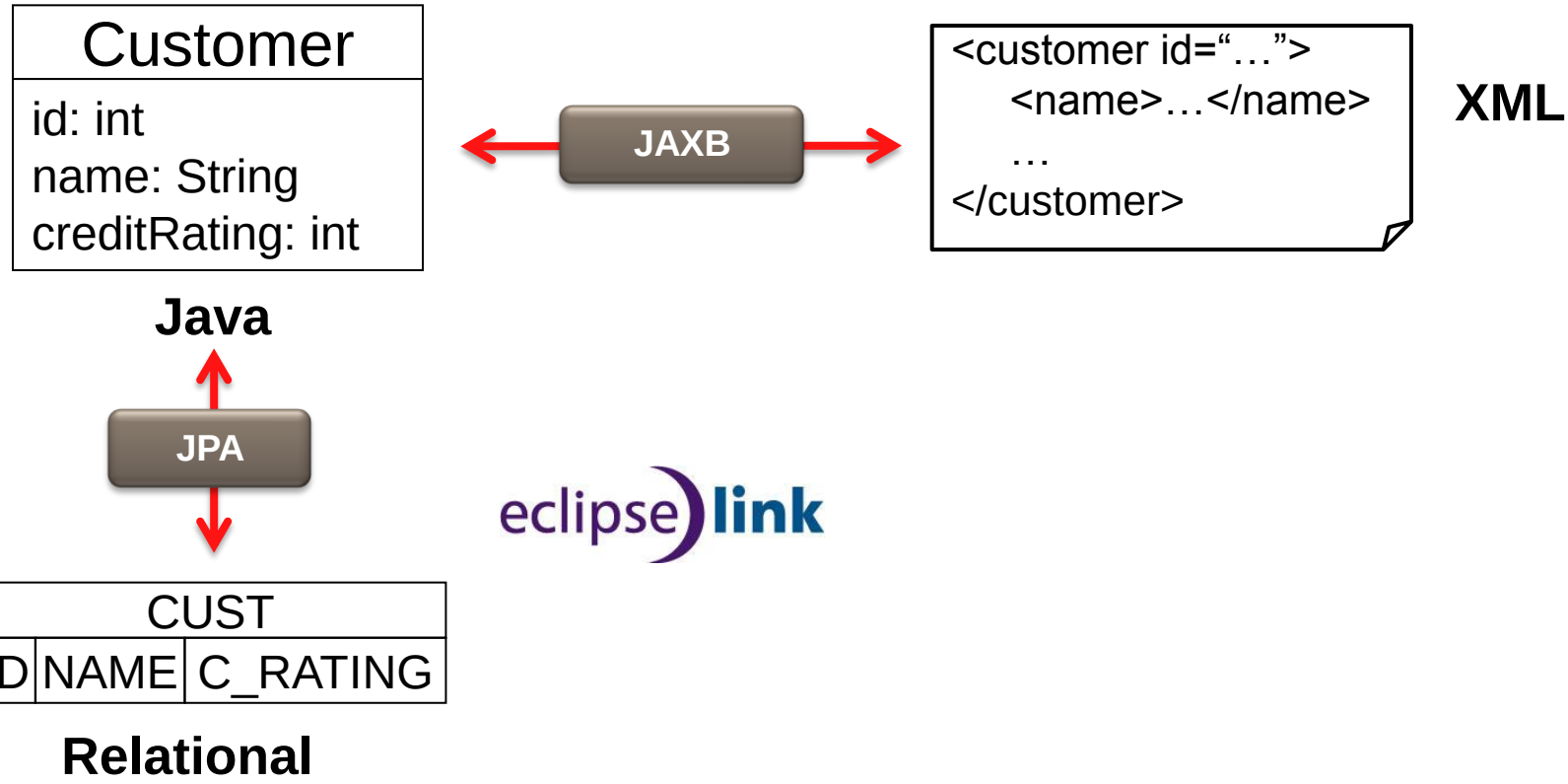
Java



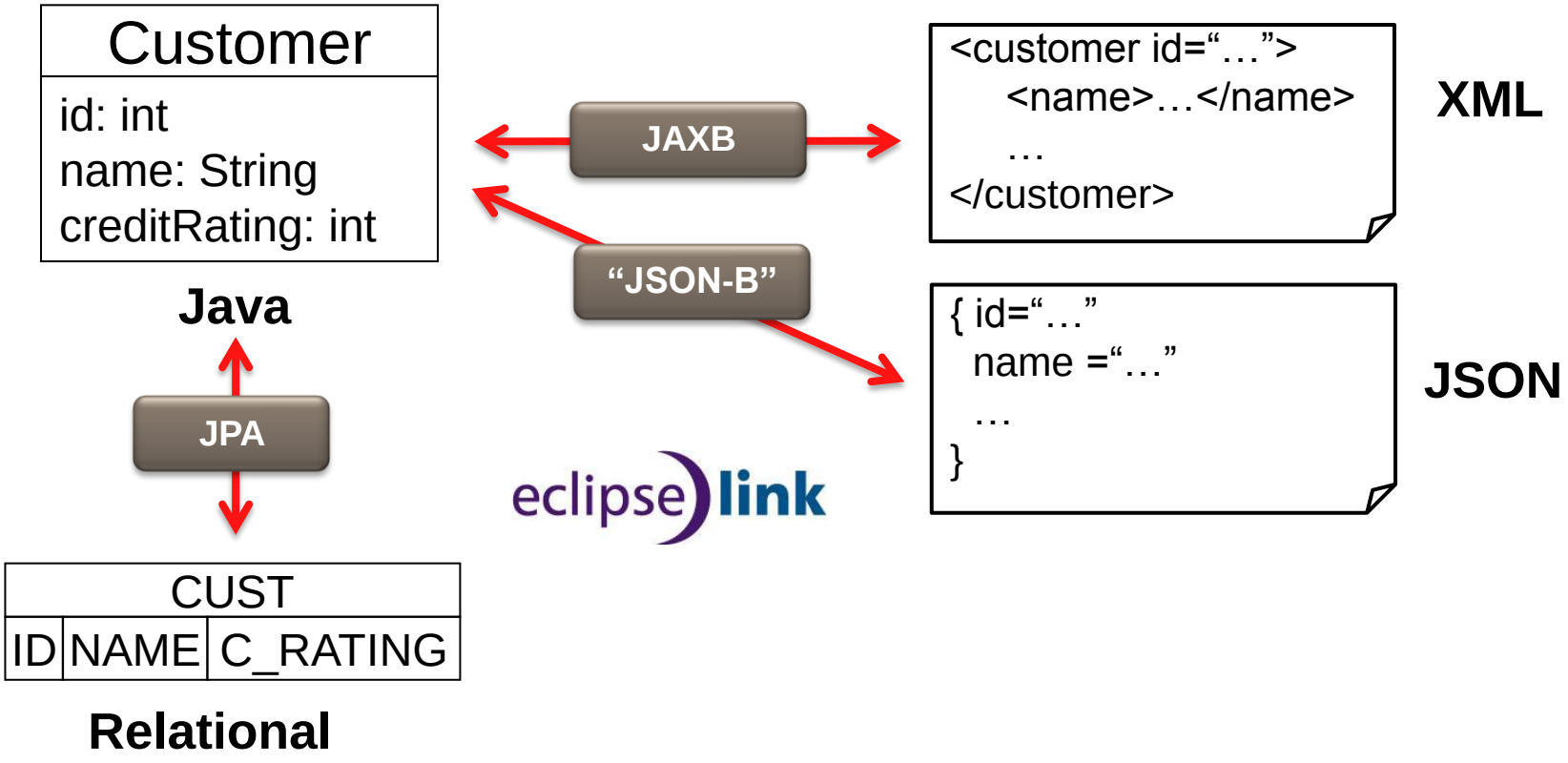
CUST		
ID	NAME	C_RATING

Relational

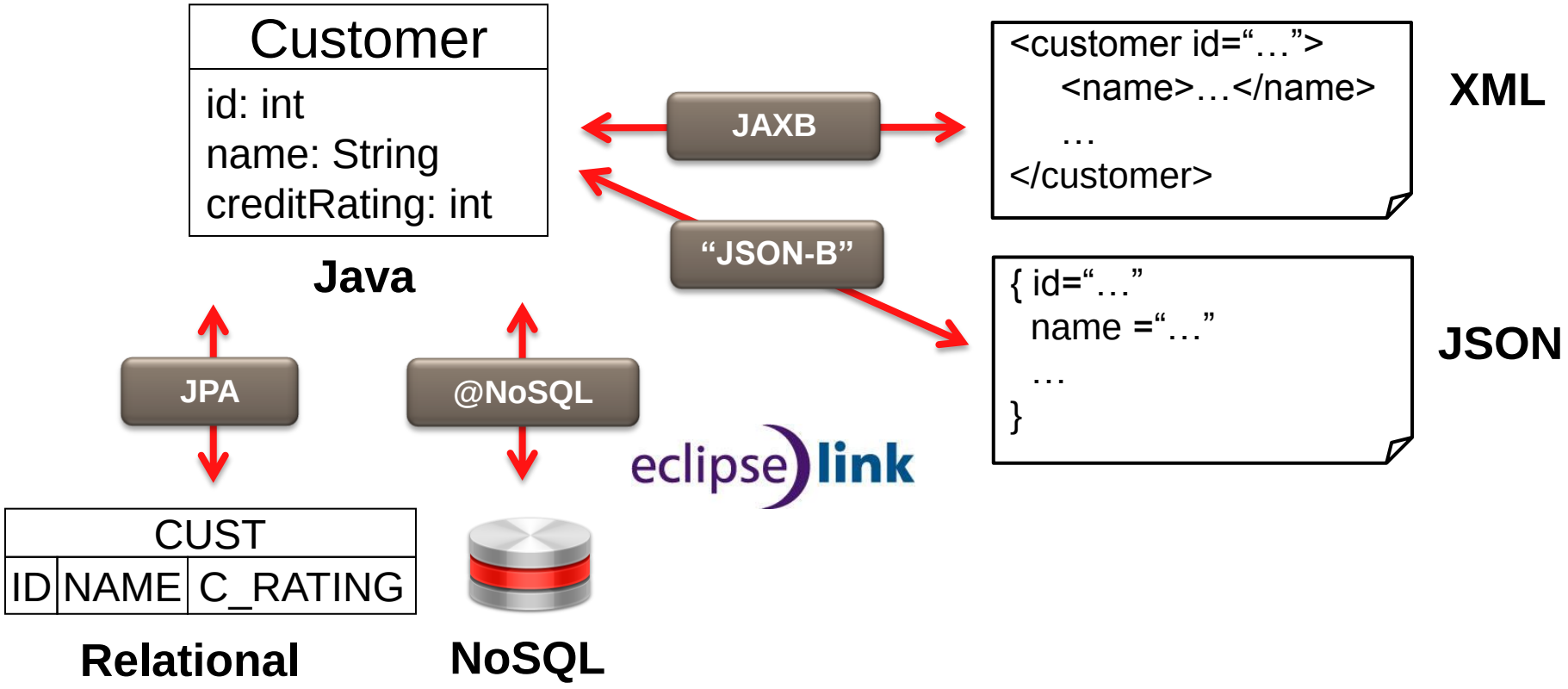
EclipseLink: more than JPA



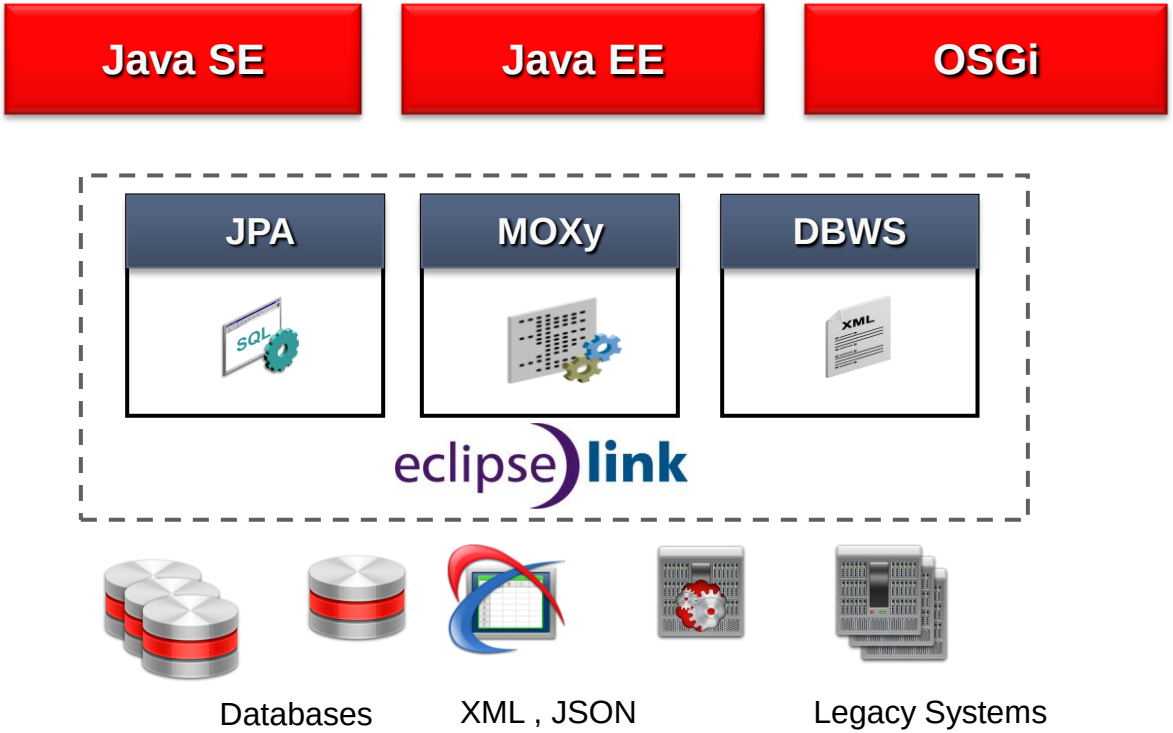
EclipseLink : more than JPA



EclipseLink : more than JPA



EclipseLink Project



eclipse

ORACLE
FUSION MIDDLEWARE
WEBLOGIC

ORACLE
TOPLINK

...

EclipseLink

- www.eclipselink.org
- Links: http://git.eclipse.org/c/eclipselink/
  - GIT → Runtime, Examples etc.
- Docs: http://www.eclipse.org/eclipselink/documentation/

EclipseLink vs. TopLink

- TopLink ships EclipseLink
- support.oracle.com, Notes: 1392592.2, 1069115.1
- TopLink also includes TopLink Grid (Coherence and JPA), TopLink Data Services

Now Some Code Demos on ...

Standards

Relational
- JPA 2.1

XML
- JAXB 2.2
- SDO 2.1.1

DBWS
- JAX-WS

Recent

- JSON Binding
- Dynamic JPA
- Tenant Isolation
- RESTful JPA
- NoSQL

Future

- JSON-B
- Java Standard for NoSQL?

JSON Binding

JSON Binding / EclipseLink “JSON-B”

- Provides Java-JSON binding similar to EclipseLink JAXB’s Java-XML binding
- Marshall Java domain model to and from JSON
- Currently no Java standard—EclipseLink interprets JAXB XML bindings for JSON
- Content-type selectable by setting property on Marshaller/Unmarshaller

EclipseLink JSON-B Goals

- Offer the same flexibility as object-to-XML mappings
- Support both XML and JSON with one set of mappings
- No additional compile time dependencies over the JAXB APIs

XML and JSON from JAXB Mappings

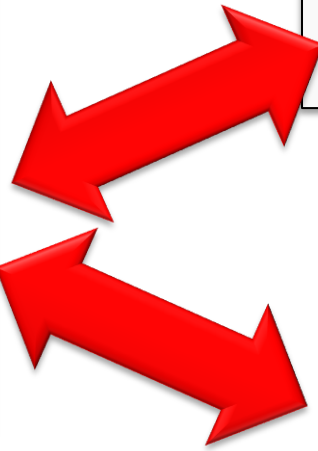
```java
@XmlRootElement(namespace="urn:example")
public class Foo {

    @XmlAttribute
    private int id;

    @XmlElement(namespace="urn:example")
    private String bar;

}
```

JAXB mapped Java



```xml
<foo xmlns="urn:examle" id="123">
   <bar>Hello World</bar>
</foo>
```

XML

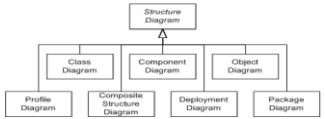
```
{"foo" : {
     "id" : 123,
     "bar : "Hello World"
}}
```

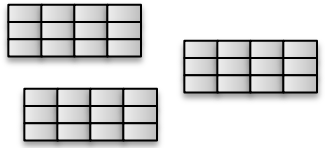
JSON

```
Marshaller marshaller = getJAXBContext().createMarshaller();
marshaller.setProperty(MarshallerProperties.MEDIA_TYPE, MediaType.APPLICATION_JSON);
marshaller.setProperty(MarshallerProperties.JSON_INCLUDE_ROOT, false);
marshaller.marshal(entity, writer);
```

Challenges – Mapping JPA Entities to XML

```
<?xml version="1.0" ?>
<employee>
  <first>Mark</first>
  <last>Twain</last>
  <id>1</id>
</employee>
```

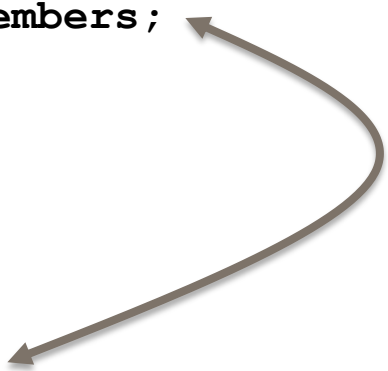




- Bidirectional/Cyclical Relationships
- Composite Keys/Embedded Key Classes
- Byte Code Weaving

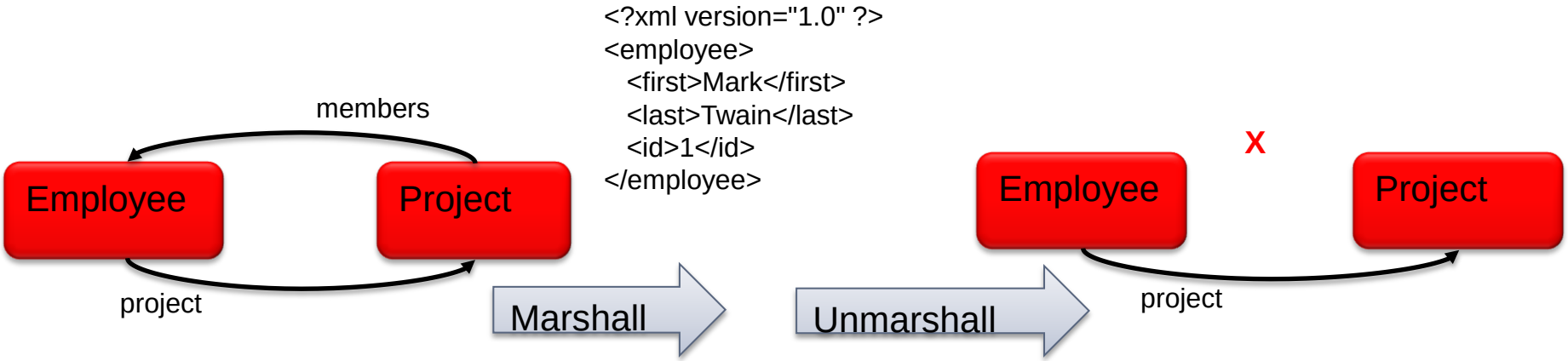
Bidirectional Relationship

```
@Entity
public class Project{
    ...
    @OneToMany(mappedBy="project")
    private List<Employee> members;
}
@Entity
public class Employee{
    ...
    @ManyToOne
    private Project project;
}
```



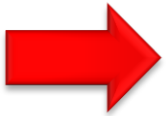
Bidirectional Relationships in JAXB

- JAXB specification does not support bidirectional relationships. One side must be marked **`@XmlTransient`**.
- But that loses the relationship!



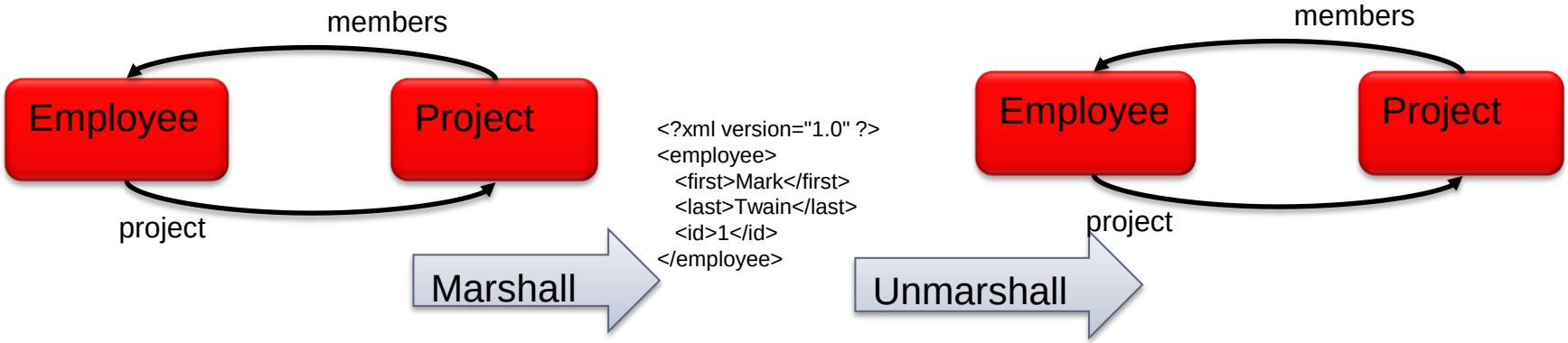
EclipseLink XmlInverseReference

```
@Entity
public class Project{
    ...
    @OneToMany(mappedBy="project")
    private List<Employee> members;
}
@Entity
public class Employee{
    ...
    @ManyToOne
    @XmlInverseReference(mappedBy="members")
    private Project project;
}
```



EclipseLink XmlInverseReference

- EclipseLink restores relationships on unmarshall!



JPA-RS

JAX-RS with JPA Example – GET Invoice

```
public class InvoiceService {...



 public Invoice read(int id) {
    return null;
 }
...
```

JAX-RS with JPA Example – GET Invoice

```
@Stateless
public class InvoiceService {...



 public Invoice read(int id) {
    return entityManager.find(Invoice.class, id);
 }
...
```

JAX-RS with JPA Example – GET Invoice

```
@Path("/invoice")
@Stateless
public class InvoiceService {...



 public Invoice read(int id) {
    return entityManager.find(Invoice.class, id);
 }
...
```

JAX-RS with JPA Example – GET Invoice

```
@Path("/invoice")
@Stateless
public class InvoiceService {...

  @GET
  @Path("{id}")
  public Invoice read(@PathParam("id") int id) {
     return entityManager.find(Invoice.class, id);
  }
...
```

JAX-RS with JPA Example – GET Invoice

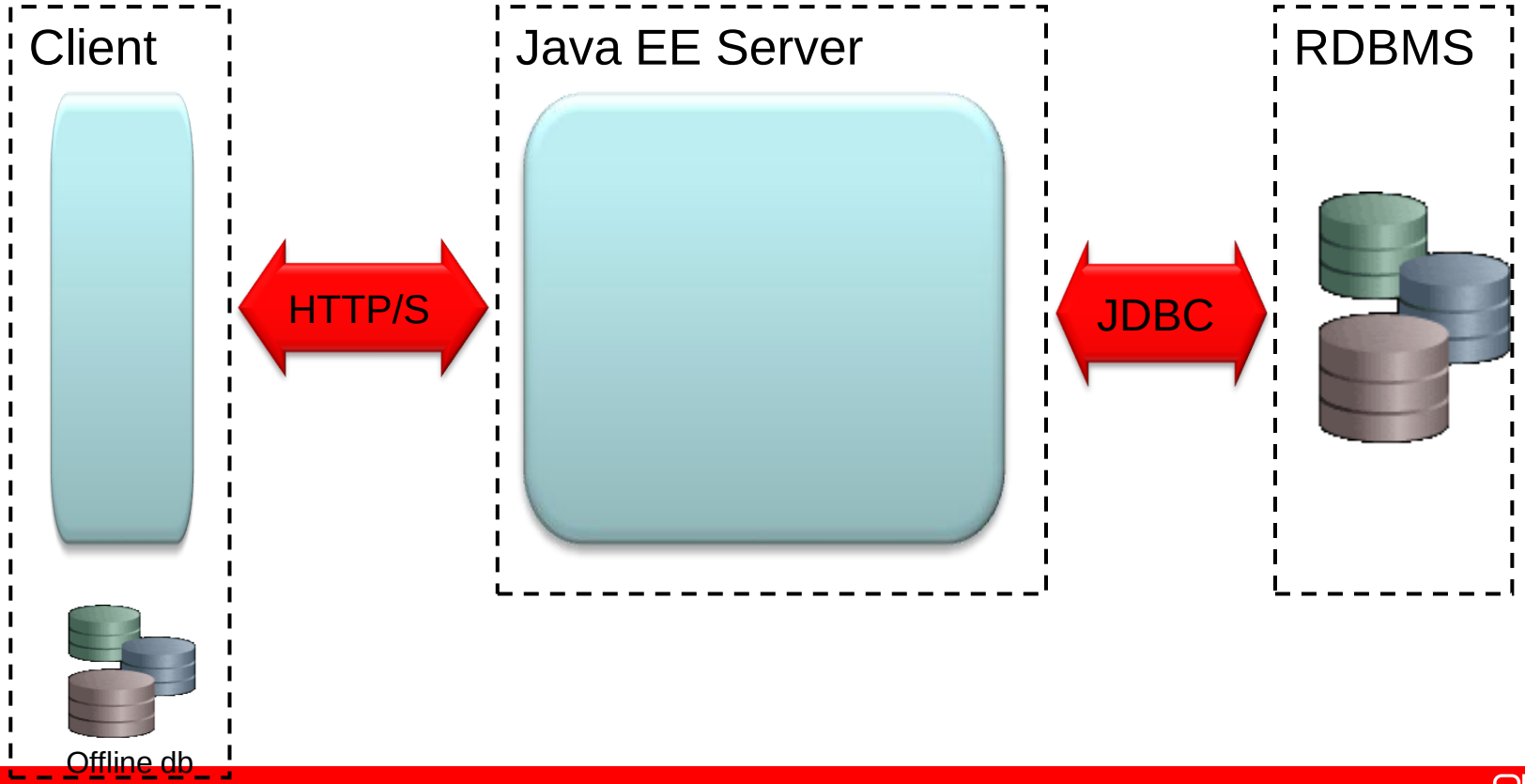
```
@Path("/invoice")
@Stateless
public class InvoiceService {...
 @GET
 @Path("{id}")
 @Produces({"application/xml", "application/json"})
 public Invoice read(@PathParam("id") int id) {
    return entityManager.find(Invoice.class, id);
 }
...
```

JAX-RS with JPA Example – GET Invoice

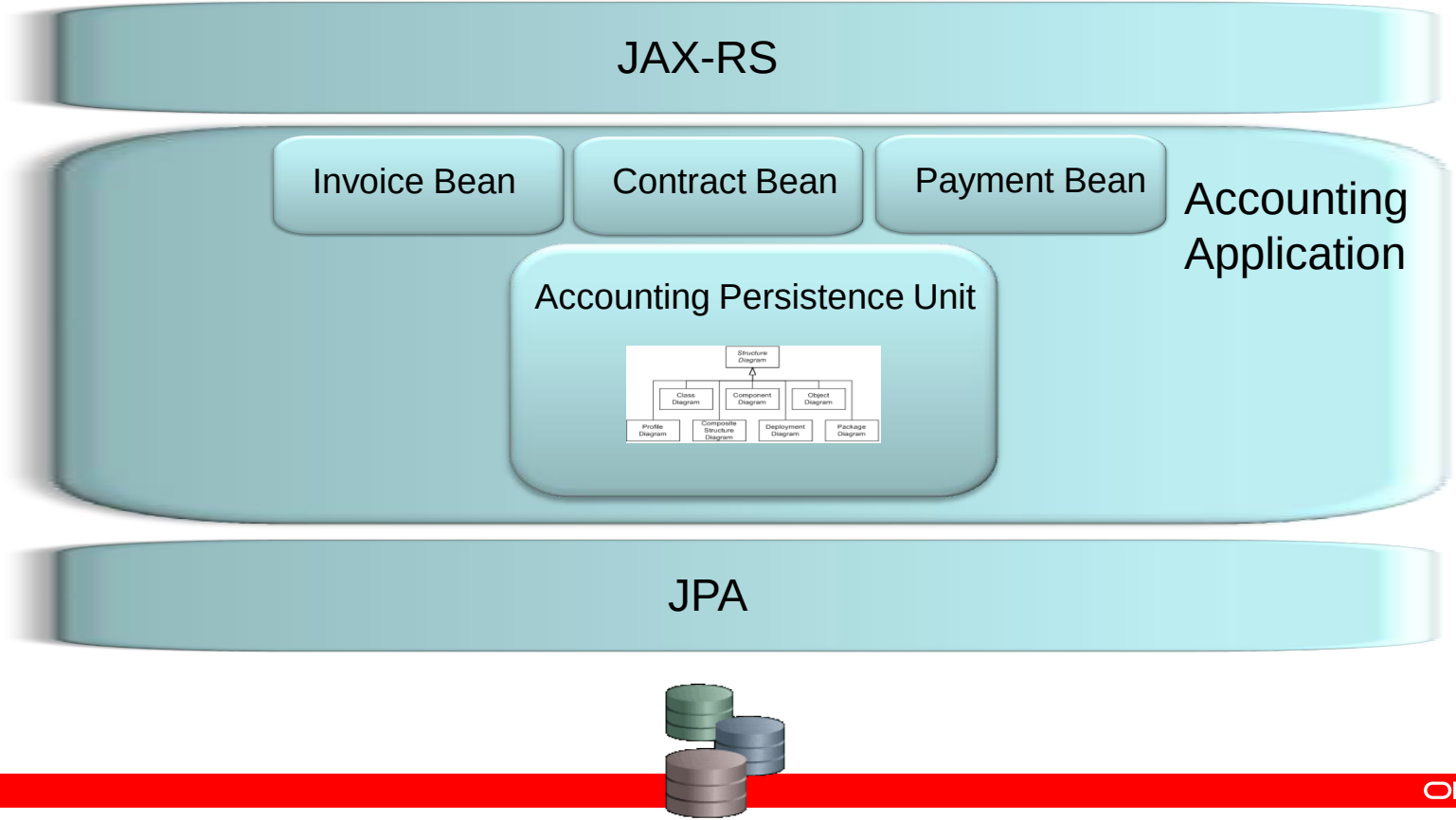
```
@Path("/invoice")
@Stateless
public class InvoiceService {...
 @GET
 @Path("{id}")
 @Produces({"application/xml", "application/json"})
 public Invoice read(@PathParam("id") int id) {
    return entityManager.find(Invoice.class, id);
 }
...
```

GET http://[*machine*]:[*port*]/[*web-context*]/invoice/4

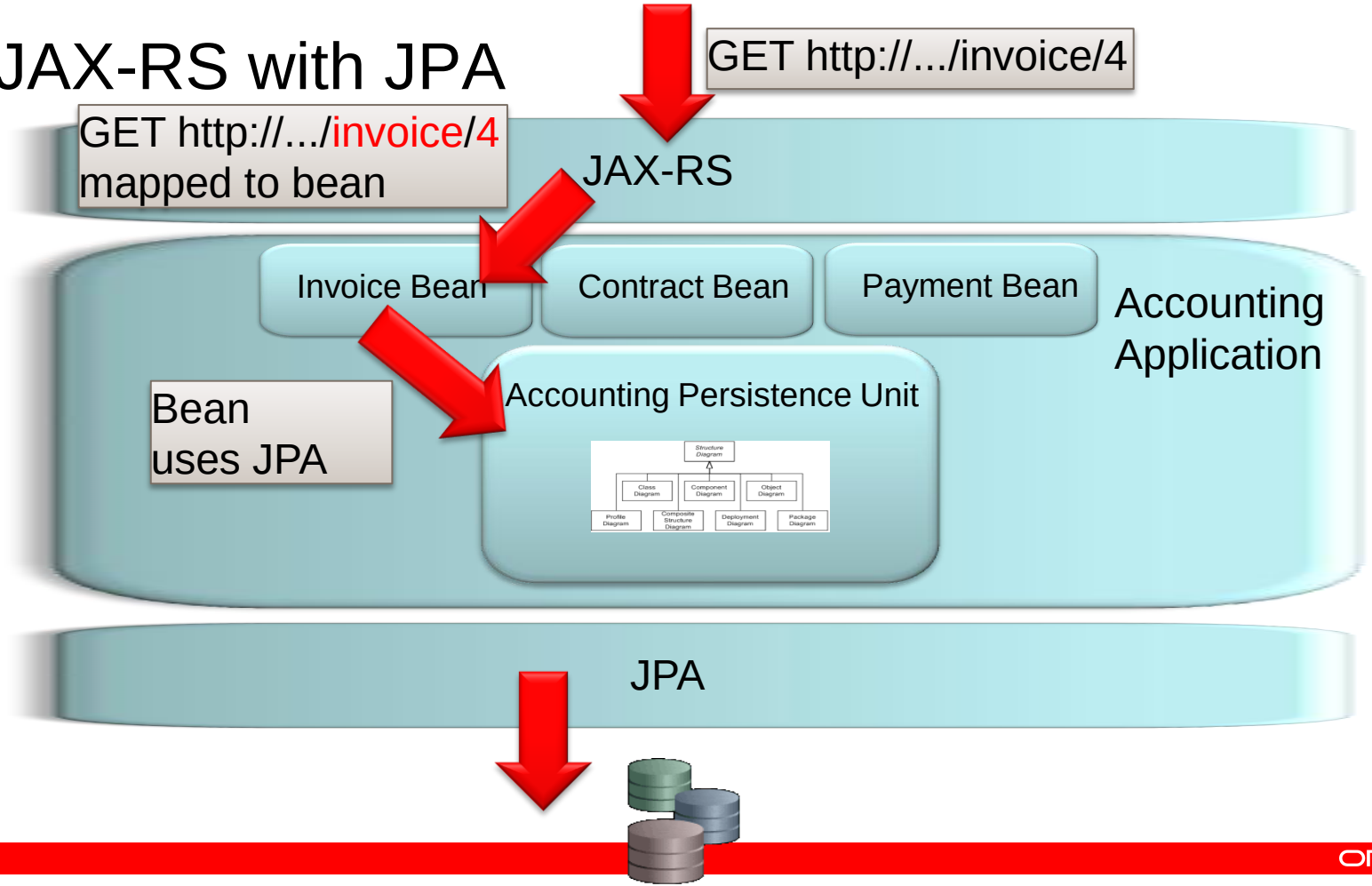
JAX-RS with JPA—High Level Architecture



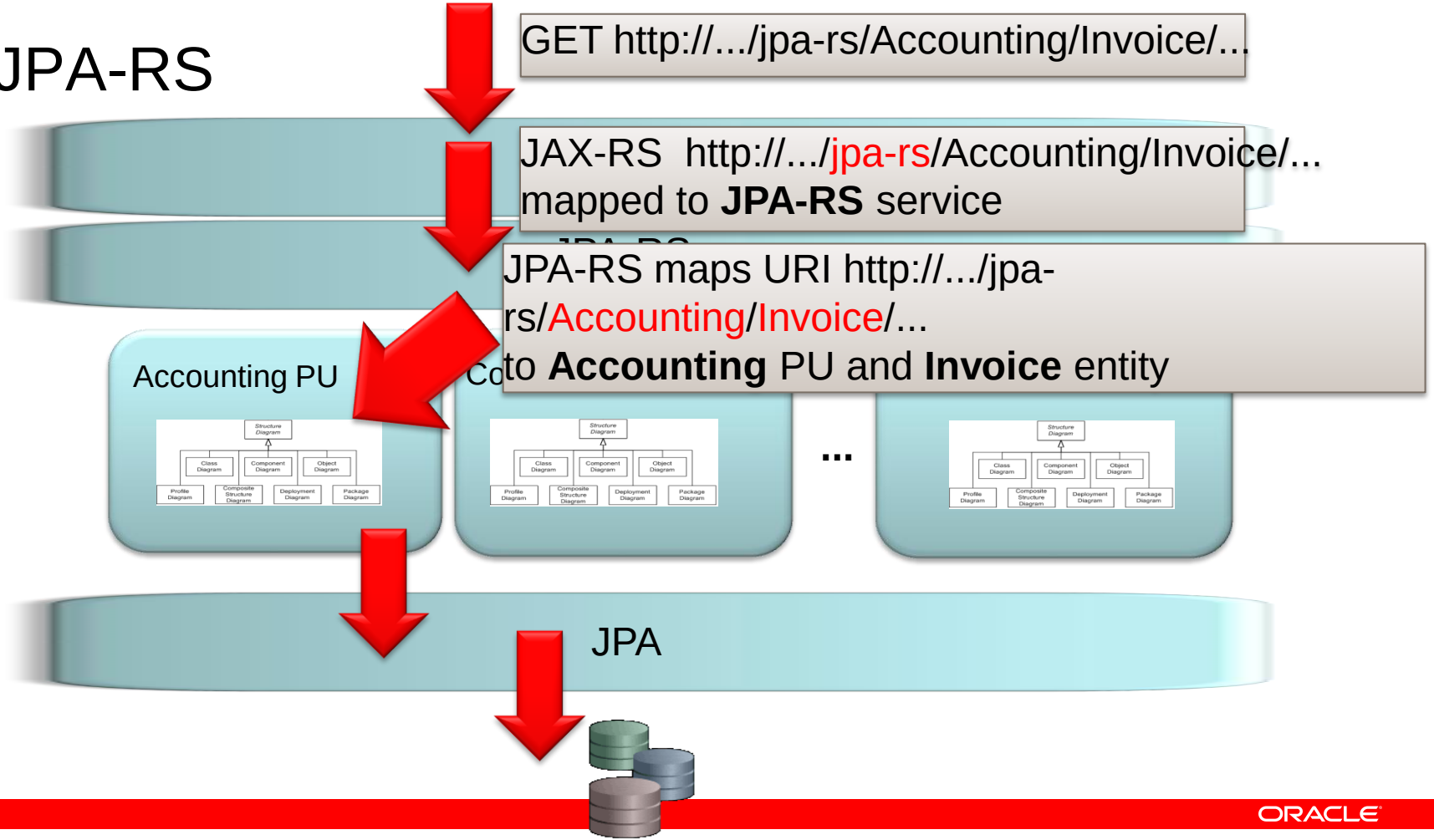
JAX-RS with JPA Example



JAX-RS with JPA



JPA-RS


GET http://.../jpa-rs/Accounting/Invoice/...
JAX-RS http://.../jpa-rs/Accounting/Invoice/...
mapped to **JPA-RS** service
JPA-RS maps URI http://.../jpa-rs/Accounting/Invoice/...
to **Accounting** PU and **Invoice** entity
Accounting PU
...
JPA

JPA-RS Features

- Access relational data through REST with JSON or XML
- Provides REST operations for entities in persistence unit (GET, PUT, POST, DELETE)
- Supports invocation of named queries via HTTP
- Server Caching—EclipseLink clustered cache
- Dynamic Persistence also supported
  - **Entities defined via metadata—no Java classes required**
  - Enables persistence services for HTML 5/JavaScript applications

NoSQL Java Persistence

NoSQL Databases

- NoSQL (i.e., non-relational) database are increasingly popular
- No standards
- Differing APIs and feature sets
- Some offer query language/API—some not

EclipseLink NoSQL

- Support JPA-style access to NoSQL databases
  - Leverage non-relational database support for JCA (and JDBC when available)
- Define annotations and XML to identify NoSQL stored entities (e.g., @NoSQL)
- Support JPQL subset for each
  - Key principal: leverage what's available
- Initial support for MongoDB and Oracle NoSQL.
- Support mixing relational and non-relational data in single composite persistence unit


ORACLE®