

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.





**MOVING JAVA
FORWARD**

ORACLE

Java Persistence in the Cloud

Shaun Smith

shaun.smith@oracle.com /  @shaunMsmith

Application Development and the Cloud

- Today
 - Single Tenant or non-Tenant Applications
 - Dedicated application instance and database
- Future
 - Support multiple tenants
 - Support extensibility (custom fields per tenant)
 - Support various deployment architectures
 - Dedicated or shared application instances
 - Dedicated or shared databases

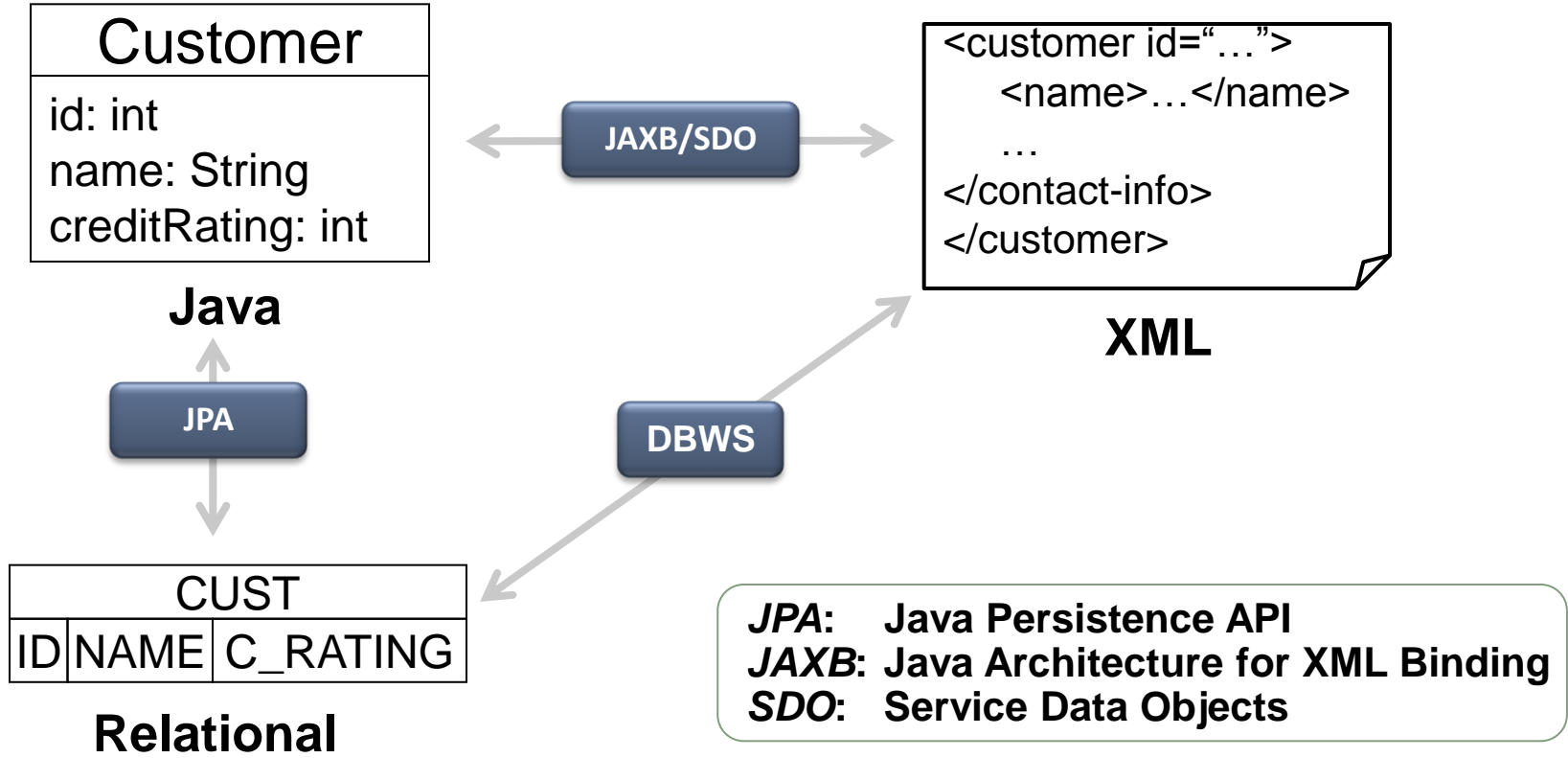


ORACLE

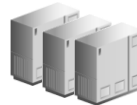
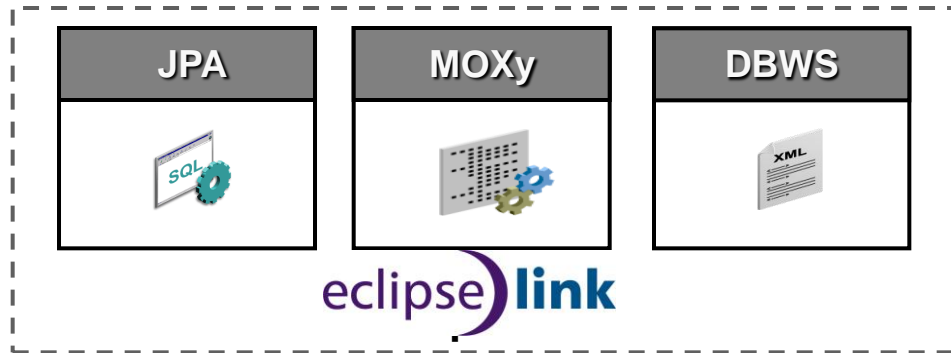
Developing for the Cloud?

- What is the Cloud?
 - Infrastructure - IaaS
 - Platform - PaaS
 - Software – SaaS
- What is a tenant?
- Persistence Challenges
 - Same application used for multiple tenants
 - Flexible domain models: Extensibility
 - Multiple deployment architectures
 - Shared versus Isolated container and database

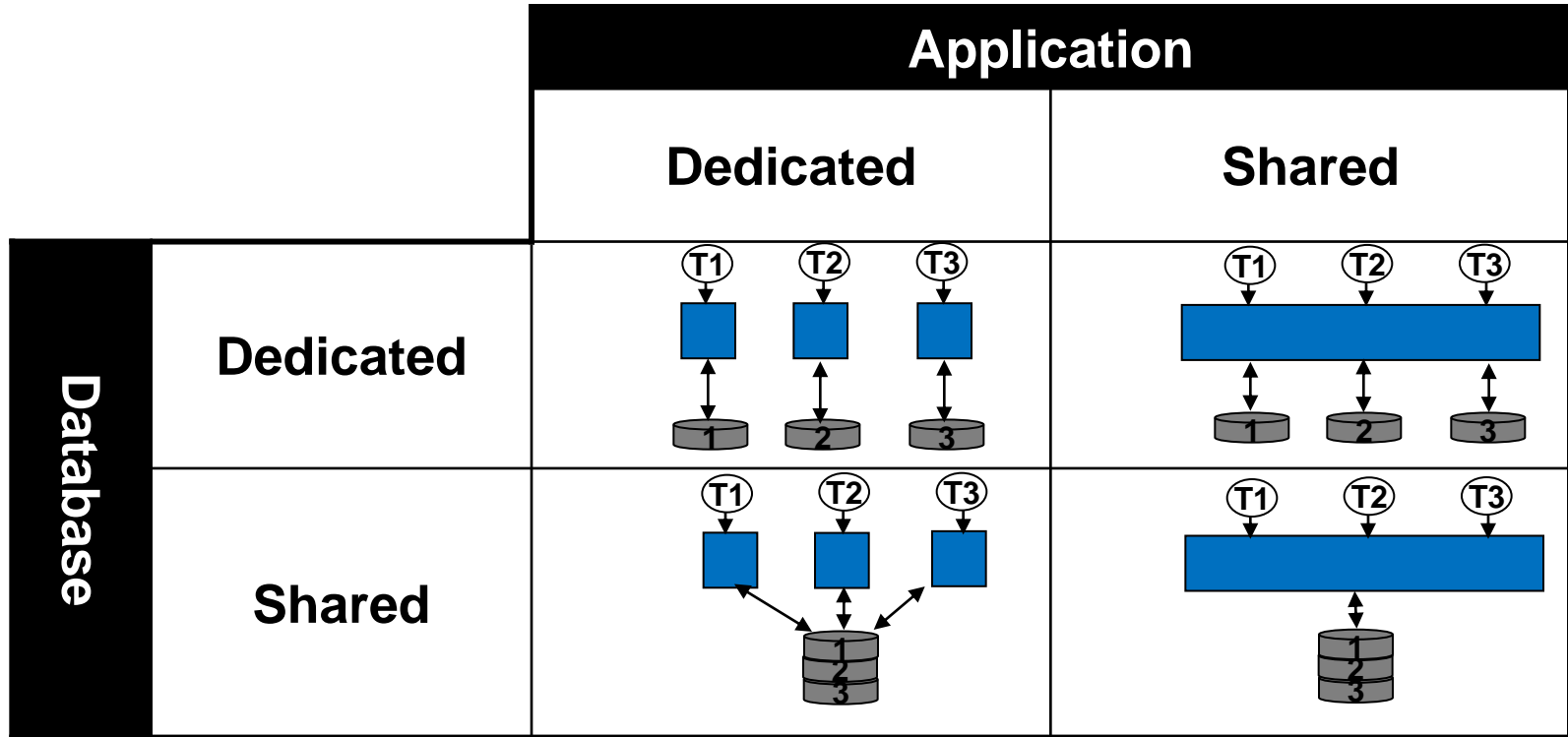
Java Persistence: The Problem Space



EclipseLink Project



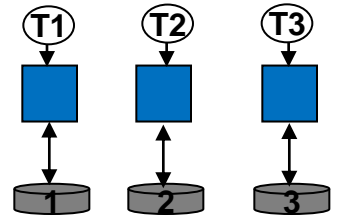
Multitenant Topologies



Note: Single application deployed to support various MT architectures

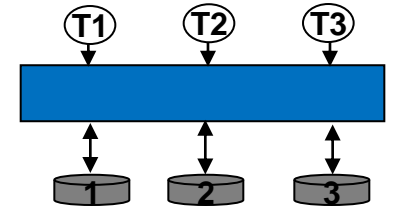
Multitenant: Dedicated Application Dedicated DB

- Dedicated application Instance
 - Application instance per tenant
 - unique container or application class-loader
 - Caching supported
- Dedicated database
 - Unique tables (tablespace/schema/db) per tenant
 - Tenant specific data source required



Multitenant: Shared App Dedicated DB

- Shared Application Instance
 - Application instances handle multiple tenants
 - Caching must isolate by tenant
- Dedicated Database
 - Common data source
 - Unique schema/tablespace per tenant
 - Common schema with table per tenant (partitioning)
 - Proxy Authentication
 - Data source per tenant



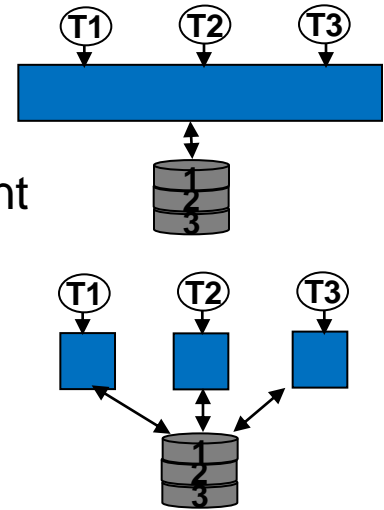
Shared Database

- **@Multitenant**

- Application's persistence layer manages access
- Row data includes tenant identifier values
- Queries augmented to limit results based on current tenant
- Database vendor independent

- **@Multitenant(VPD)**

- Row data includes tenant identifier values
- Database provides client limited view of database tables
 - Shared solution for all database clients
 - Native queries (SQL) supported



Tenant Customized Applications

- Multitenant
 - Tenant Shared Schema or Tenant per Schema
 - Tenant Specific Extensions
- Deployment Customization
 - External metadata overrides
 - Cached metadata
 - Composite Persistence Units

Multitenant Entity Types

- Support storage of entities from multiple tenants in a shared schema
- @Multitenant Strategies
 - @Multitenant(SINGLE_TABLE) - default
 - @Multitenant(VPD)
 - SINGLE_TABLE + includeCriteria=false
 - SET_IDENTIFIER(property) & CLEAR_IDENTIFIER
 - DDL Gen of predicate function and ADD_POLICY
 - @Multitenant(TABLE_PER_TENANT)

In the beginning...

- Application dedicated for tenant
- All rows available to all queries

```
@Entity
@Table(name="MYS_PLAYER")
public class Player {
```

MYS_PLAYER

EMP_ID	VERSION	F_NAME	L_NAME	GENDER	LEAGUE
1	1	John	Doe	M	HTHL
2	3	Jane	Doe	F	OSL

Multitenant: SINGLE_TABLE

- Simple configuration: Annotation or XML
- Flexible tenant identifier support
- EclipseLink augments generated SQL

```
@Entity
```

```
@Table(name="MYS_PLAYER")
```

```
@Multitenant
```

```
@TenantDiscriminatorColumn(name="league-id", columnName="LEAGUE")
```

```
public class Player {
```

MYS_PLAYER

EMP_ID	VERSION	F_NAME	L_NAME	GENDER	LEAGUE
1	1	John	Doe	M	HTHL
2	3	Jane	Doe	F	OSL

Multitenant using Oracle VPD

- Leverage the Oracle Database

```
@Entity
@Table(name="MYS_PLAYER")
→ @Multitenant(VPD)
@TenantDiscriminatorColumn(name="league-id", columnName="LEAGUE")
public class Player {
```

MYS_PLAYER

EMP_ID	VERSION	F_NAME	L_NAME	GENDER	LEAGUE
1	1	John	Doe	M	HTHL
2	3	Jane	Doe	F	OSL

Multitenant: TENANT_PER_TABLE

- Future Plans

```
@Entity
@Multitenant(TABLE_PER_TENANT)
public class Player {
```

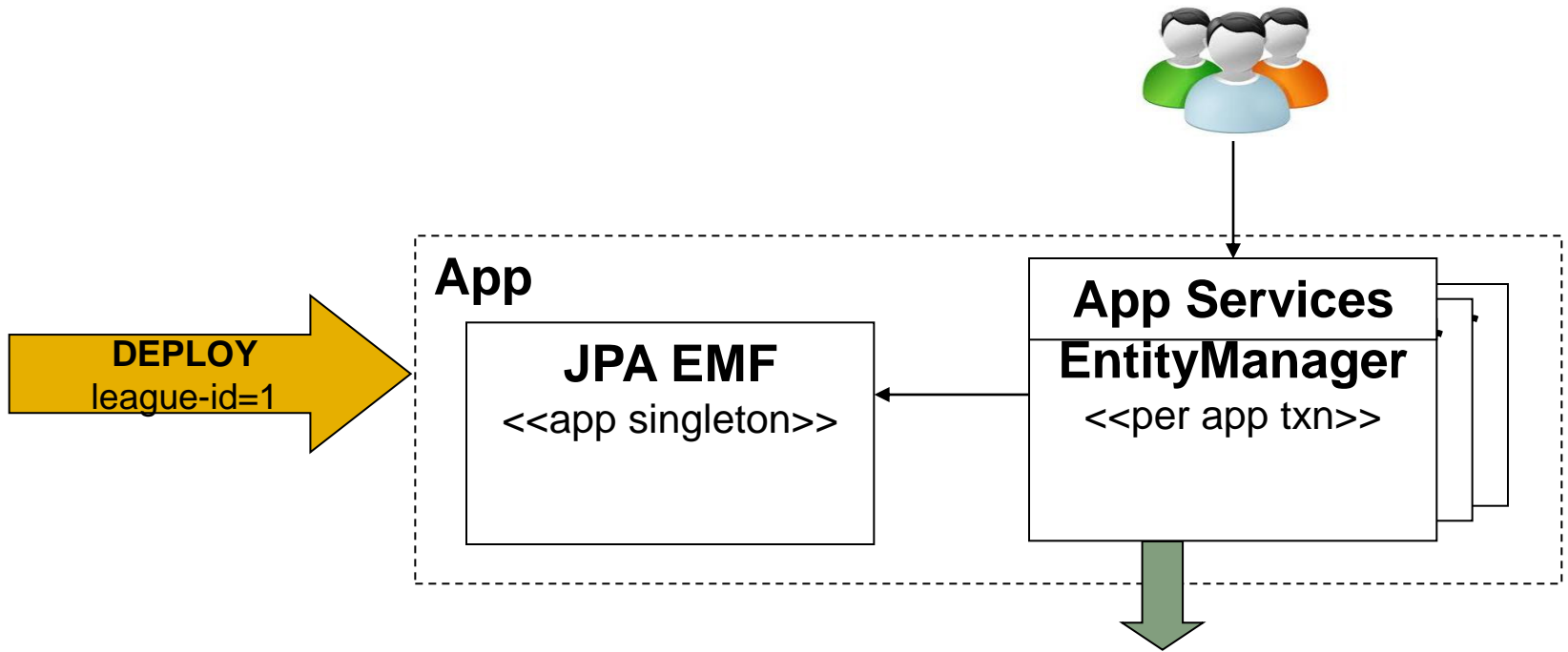
HTHL.MYS_PLAYER

EMP_ID	VERSION	F_NAME	L_NAME	GENDER
1	1	John	Doe	M

OSL.MYS_PLAYER

EMP_ID	VERSION	F_NAME	L_NAME	GENDER
2	3	Jane	Doe	F

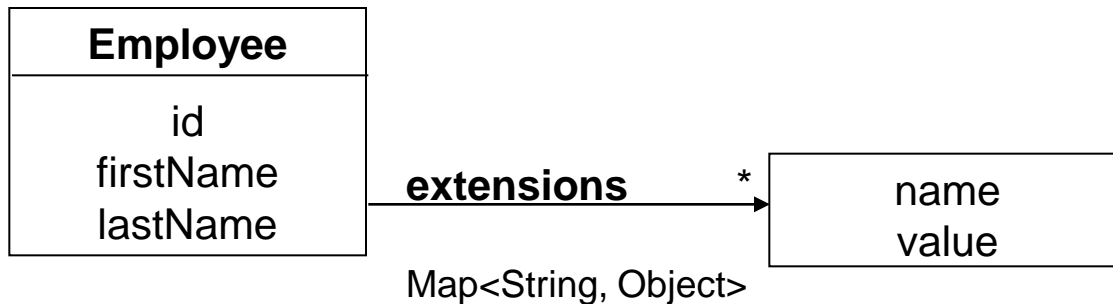
Multi-Tenancy at Runtime



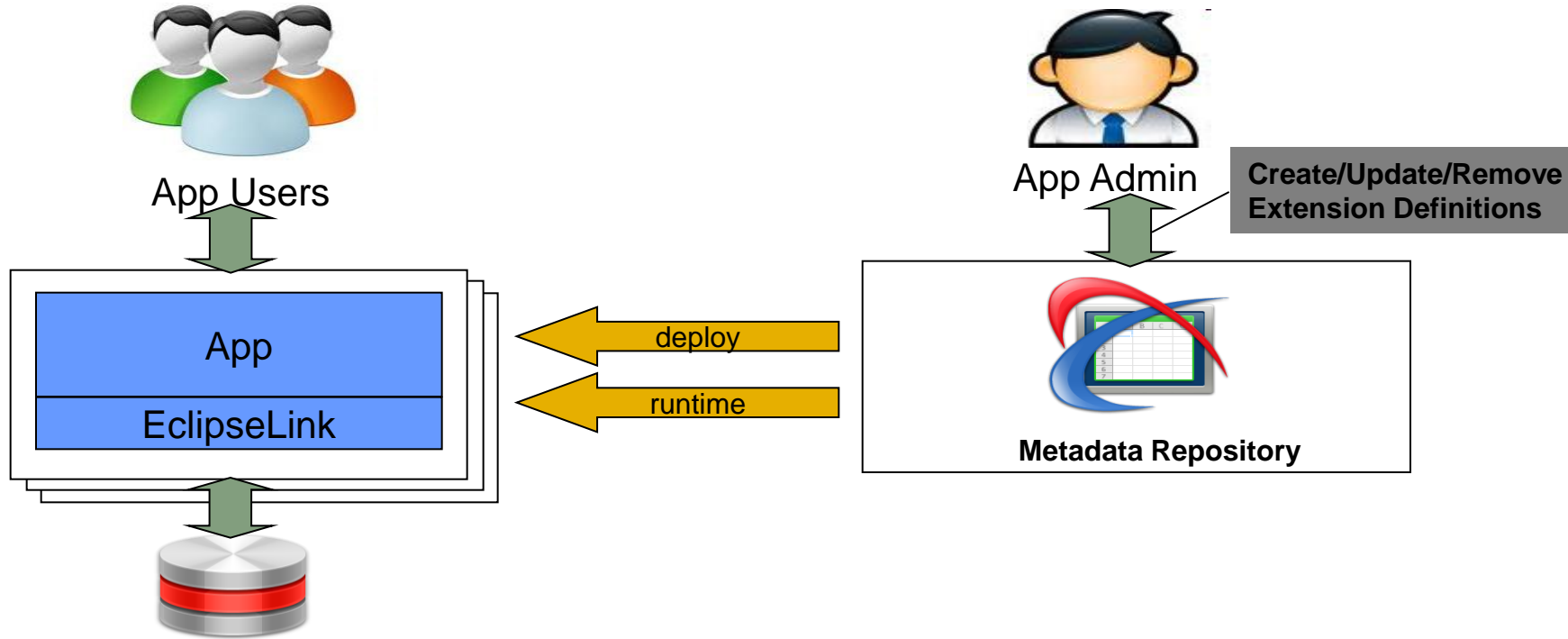
```
SELECT * FROM MYS_PLAYER WHERE L_NAME = 'Doe' AND LEAGUE = 1
```

Domain Model Extensions

- Storage and querying of extended properties
 - Application developer enables extensions in entity
 - Schema created with extension columns/table(s)
 - Application Admin stores extension definitions
 - Application instances made aware of extension definitions
 - Application users make use of extensions



Tenant/Extension Metadata Management



Flex Extensions

```
@VirtualAccessMethods
public class Player{
...
@Transient
private Map<String, Object> attributes;

public <T> T get(String attributeName) {
    return (T) this.attributes.get(attributeName);
}

public Object set(String attributeName, Object value) {
    return this.attributes.put(attributeName, value);
}
```

MYS_PLAYER

EMP_ID	F_NAME	L_NAME	FLEX_1	FLEX_2
1	John	Doe	'R'	'22'
2	Jane	Smith	'NONE'	

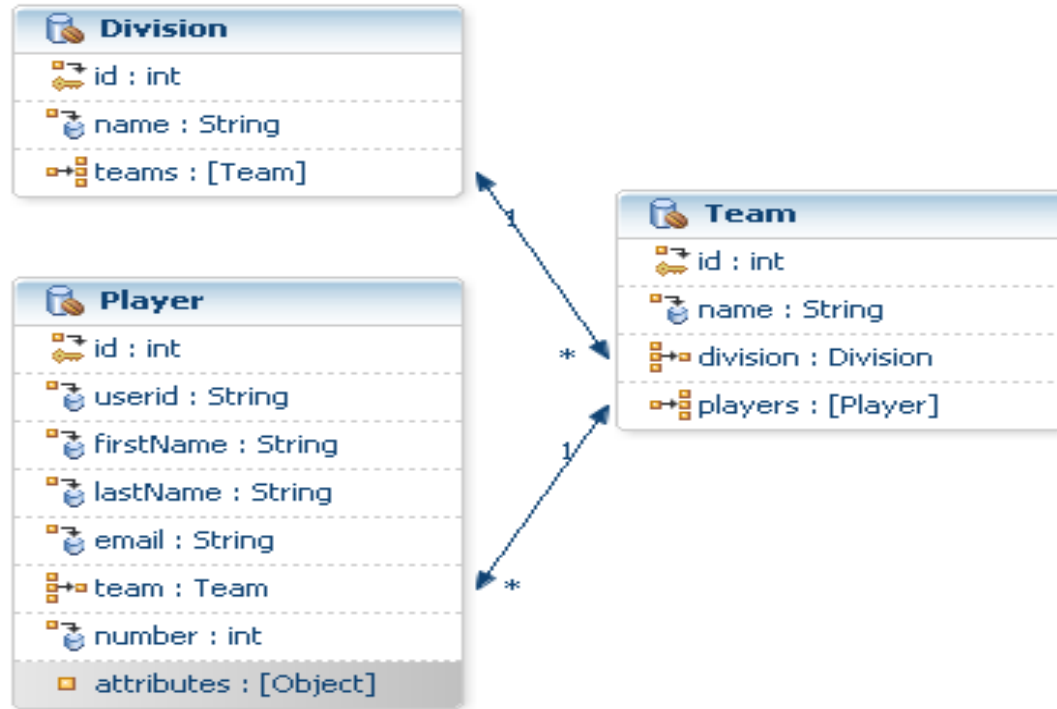
Definition: Virtual Access Mappings

```
<entity class="example.mysports.model.Player">
  <attributes>
    <basic name="penaltyMinutes" access="VIRTUAL"
      attribute-type="java.lang.Integer">
      <column name="flex_1"/>
    </basic>
    <basic name="position" access="VIRTUAL"
      attribute-type="java.lang.String">
      <column name="flex_2"/>
    </basic>
  </attributes>
</entity>
```

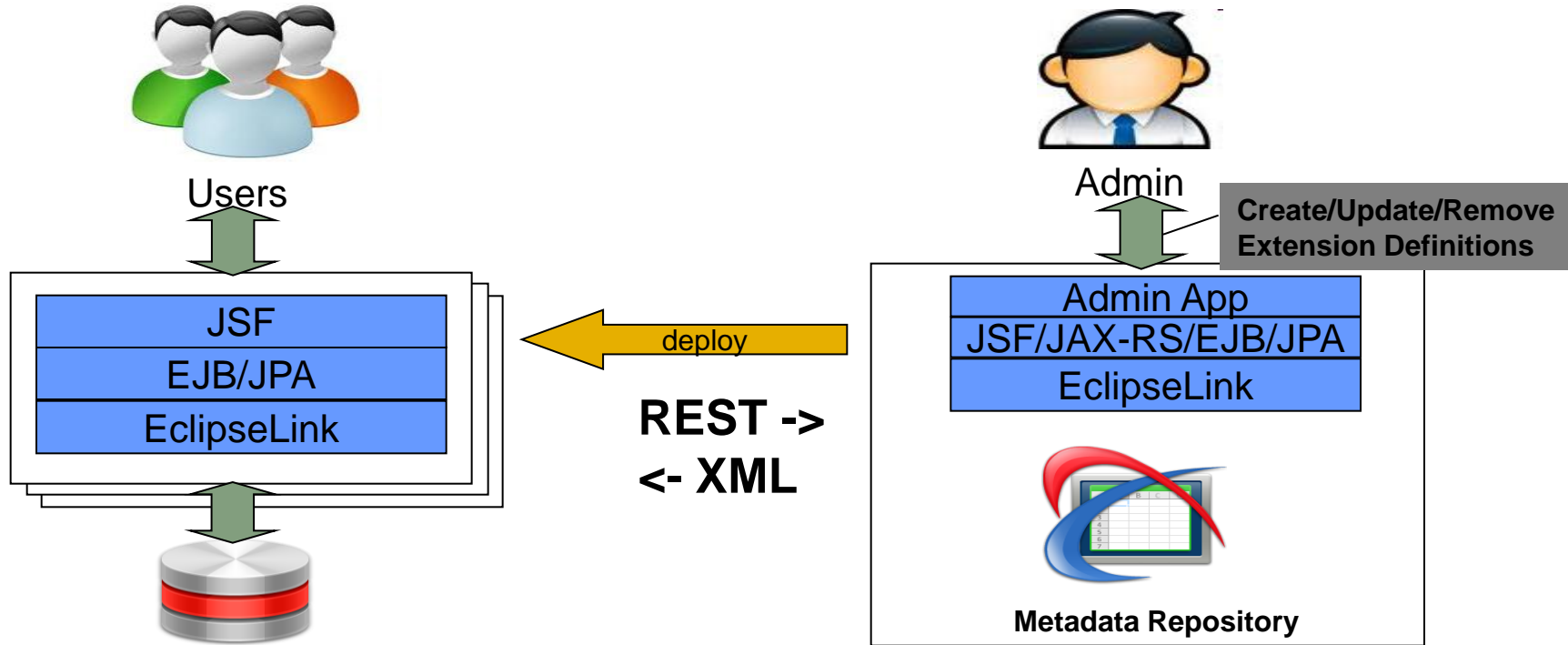
MySports Demo

- Introduced in EclipseLink Indigo (2.3)
- Features
 - @Multitenant
 - @VirtualAccessMethods (Extensions per Tenant)
 - External Metadata Sources
 - JSF, EJB, JPA
 - Admin: JSF + JAX-RS + JPA
- Wiki
 - <http://wiki.eclipse.org/EclipseLink/Examples/MySports>

MySports Demo Model



MySports Architecture



Summary: Java Persistence in the Cloud

- Starting thinking about SaaS and PaaS in your applications
- Java EE 7 and the Cloud
- Call to Action
 - Try it out
 - Download EclipseLink Indigo (2.3.0) Release Candidates
 - www.eclipse.org/eclipselink/downloads/milestones.php
 - Try out the MySports Demo
 - wiki.eclipse.org/EclipseLink/Examples/MySports
 - Provide Feedback, Get Involved!

Q&A



