Hardware and Software

ORACLE

Engineered to Work Together

ORACLE®

Java COMPATIBLE ENTERPRISE EDITION

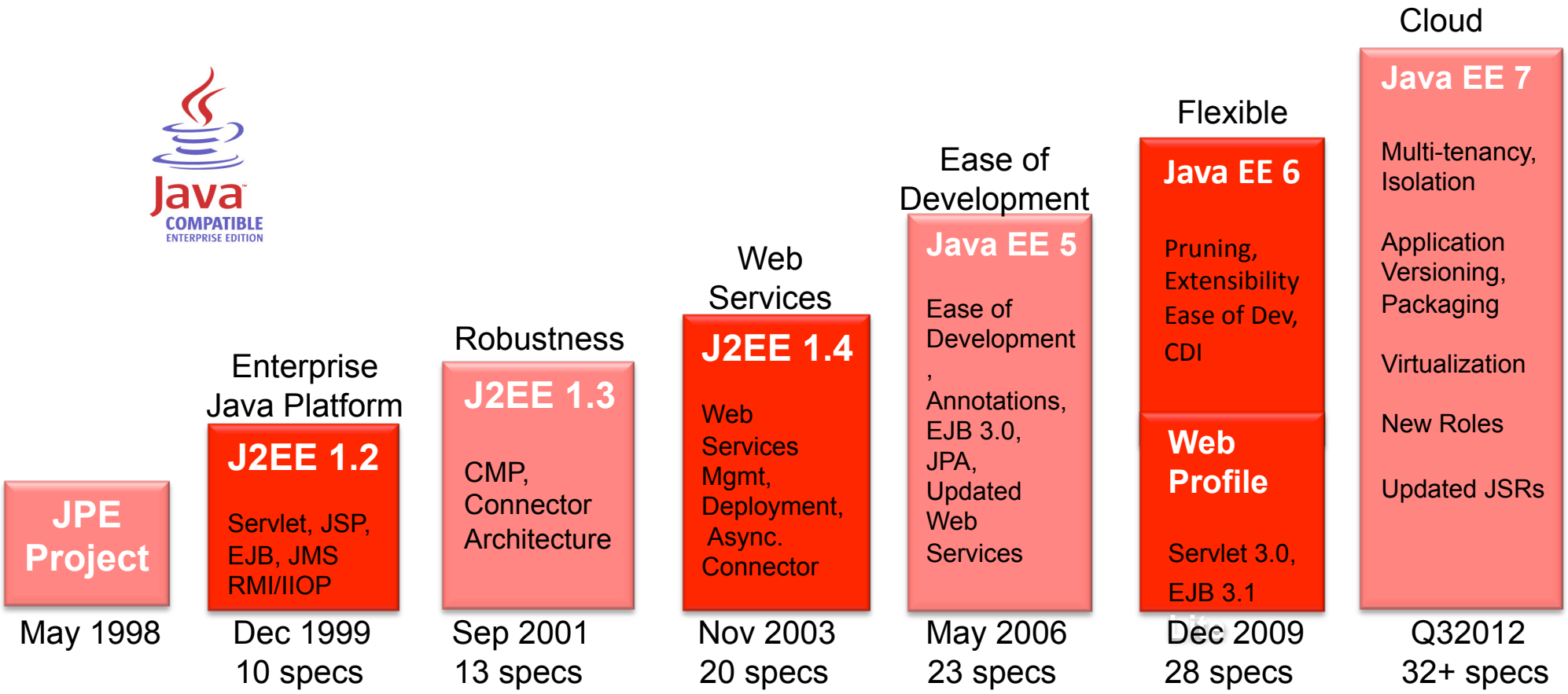
The Java EE 7 Platform: Developing for the Cloud

Adam Leftik, Java EE and GlassFish Product Management

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.

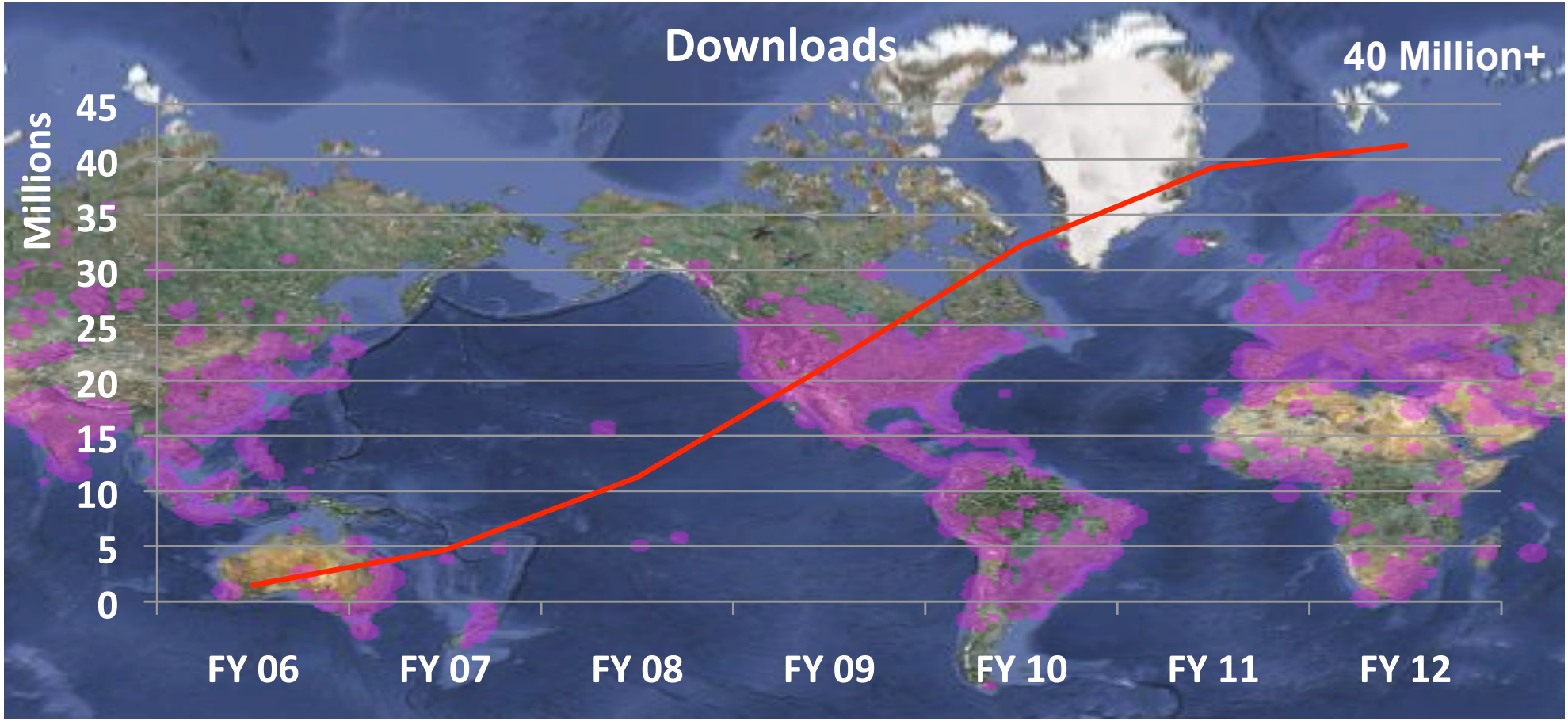
Java EE Roadmap

Java EE Past, Present, & Future



 | |

Java EE – Developer Adoption



Today's Cloud Offerings Are Vendor-Specific

Infrastructure as a Service

Platform as a Service

Software as a Service

. . .

Java EE 7 Focus: Platform as a Service

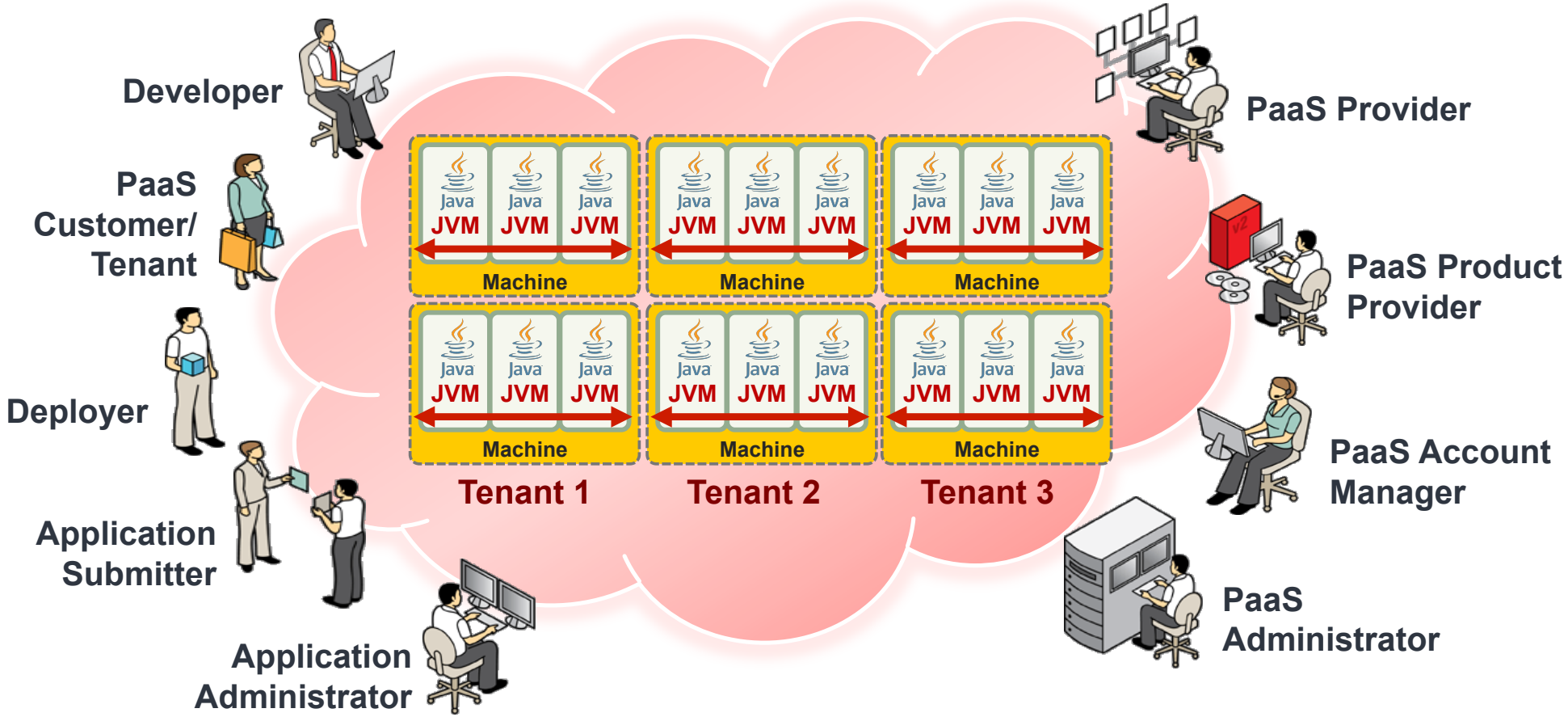
- Provide way for customers and users to leverage public, private, and hybrid clouds
- PaaS support entails evolutionary change
- Next logical step for Java EE
  - J2EE → Java EE 6 : The Java EE Platform provides services
  - Java EE 7 : The Java EE Platform IS a service

Java EE 7 PaaS Roadmap

- Define new platform roles to accommodate PaaS model
- Add metadata
  - For service provisioning and configuration
  - For QoS, elasticity
  - For sharing of applications and resources
  - For (re)configurability and customization
- Add useful APIs for cloud environment
  - JAX-RS client API, Caching API, State Management, JSON,…
- Extend existing APIs with support for multi-tenancy

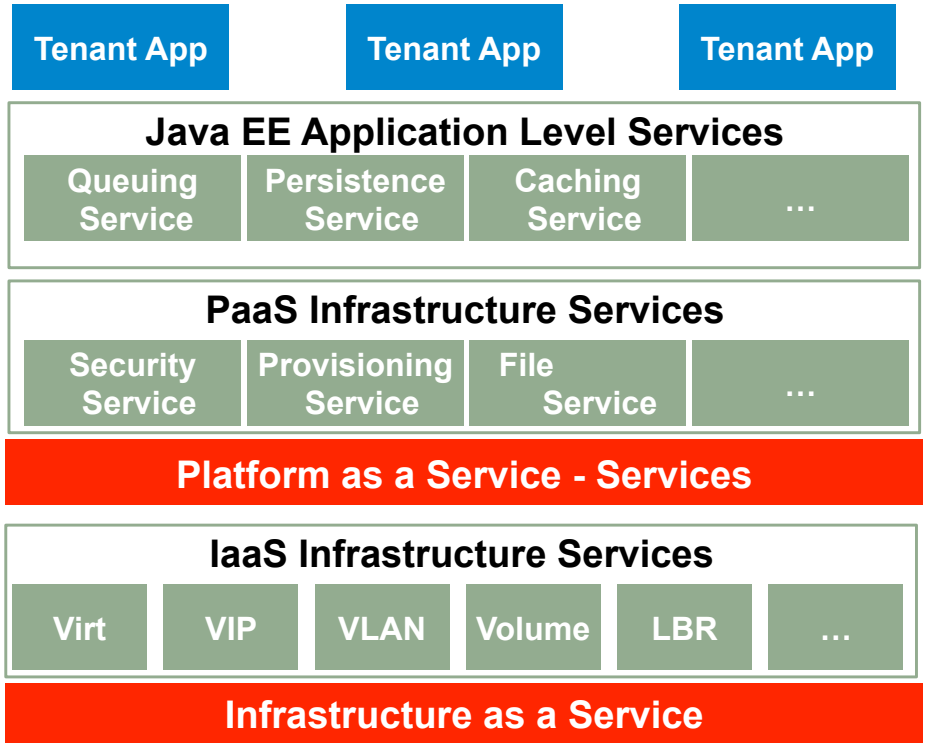
Java EE Platform Focus Areas

Roles



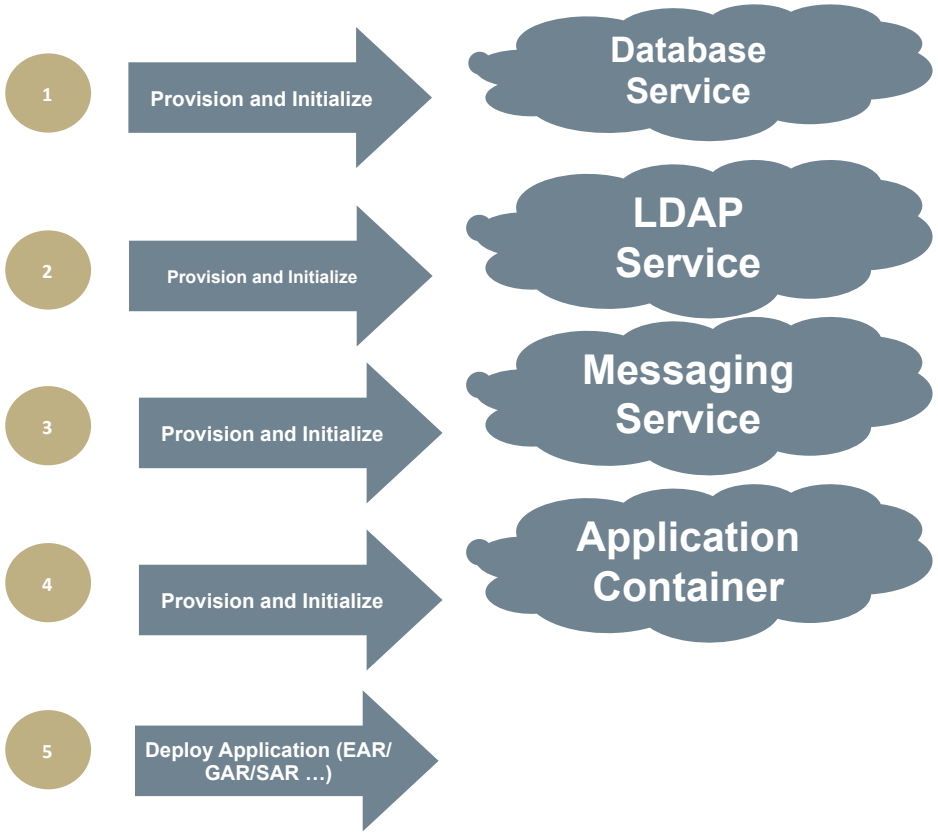
Java EE 7 Focus Area: Cloud Services, Not Just APIs

- In the cloud tenant applications consume services
- PaaS administrators host, configure, and manage application and infrastructure services
- Existing APIs in Java EE need to be updated to be service-enabled and tenant-aware
  - Example: pluggable services, late binding and tenant id injection



Old Java EE model

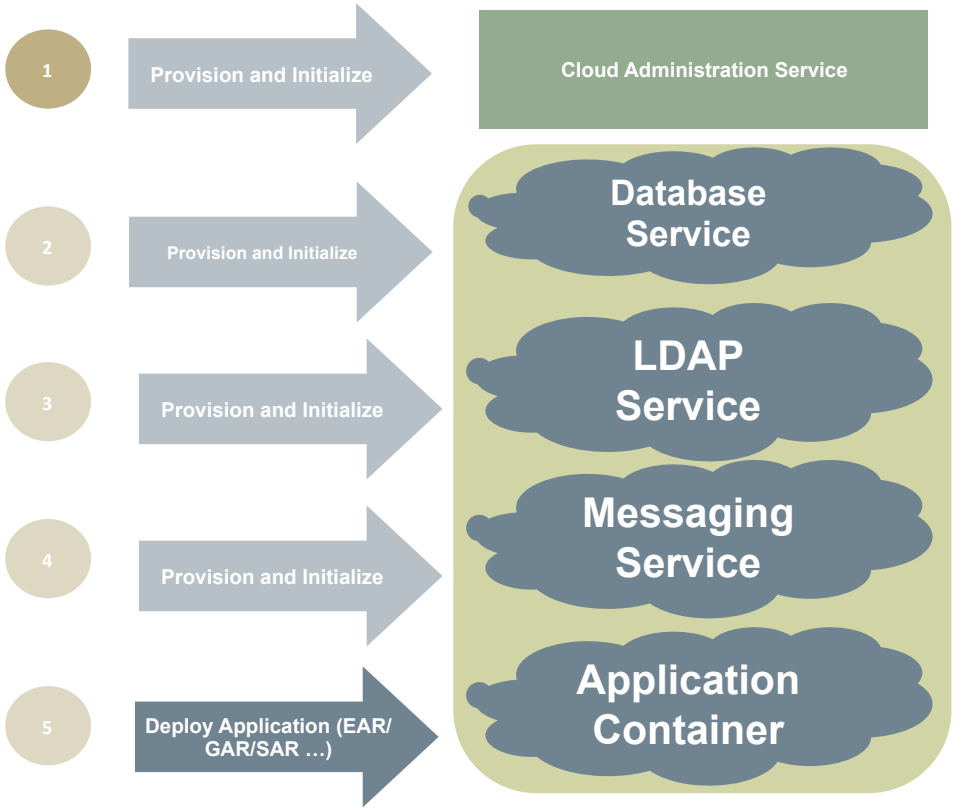
- Configure Java EE resources – JDBC, JMS etc
- Deploy Application EAR



 | |

Java EE 7 Model: Auto-Provision Services from Application Dependencies

- Provision and deploy application resources (e.g. LDAP stripe, data source instantiation and connection …)
- Extensible Deployment Models Supporting Multiple Frameworks
  - Spring, Seam, Play …



 | |

Services

- Cloud apps consume services
  - Persistence, queueing, mail, caching, …
- Service metadata facilitates ease of use when deploying into the cloud

```
@DataSourceDefinition(
    name="java:app/jdbc/myDB",
    className="oracle.jdbc.pool.OracleDataSource",
    isolationLevel=TRANSACTION_REPEATABLE_READ,
    initialPoolSize=5
)
```

Services

- Cloud apps consume services
  - Persistence, queueing, mail, caching, …
- Service metadata facilitates ease of use when deploying into the cloud

```
@JMSConnectionFactory (
   name="java:app/myJMSConnectionFactory",
   resourceType="javax.jms.QueueConnectionFactory")

@JMSDestination(
   name="java:app/myQueue",
   resourceType="javax.jms.Queue")
```

Services

- Cloud apps consume services
  - Persistence, queueing, mail, caching, …
- Service metadata facilitates ease of use when deploying into the cloud

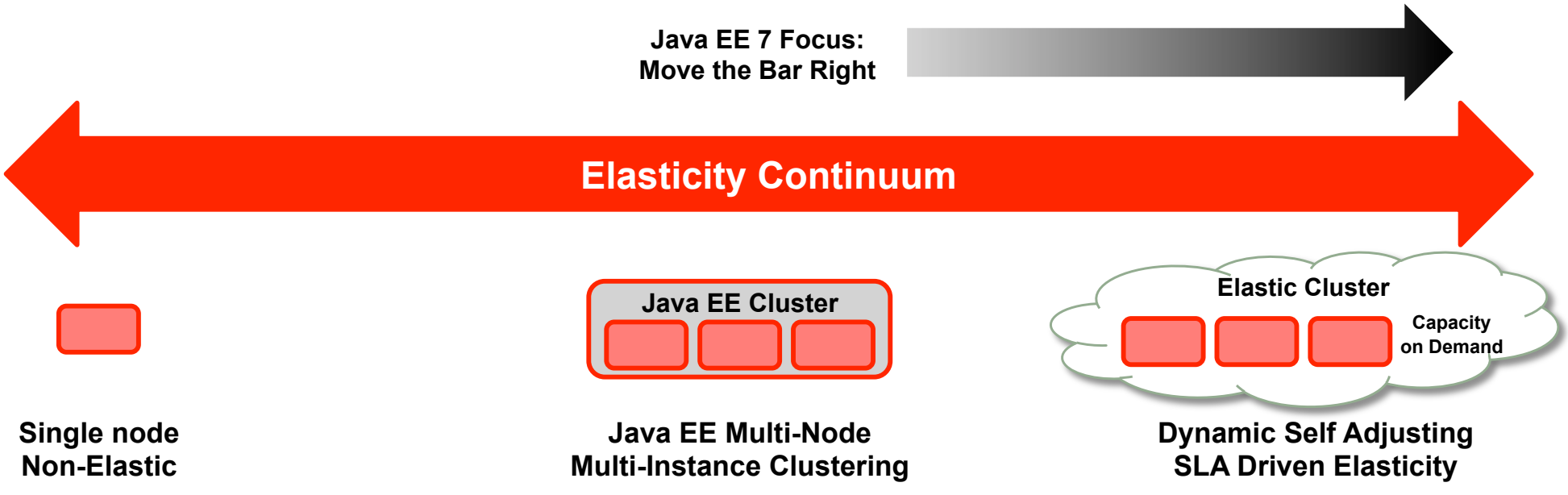
```
@MailSession (
    name="java:app/mailSession",
    from="MyService@ExtraServices.com"
)
```

Services

- Cloud apps consume services
  - Persistence, queueing, mail, caching, …
- Service metadata facilitates ease of use when deploying into the cloud

```
@ConnectorService (
    name="java:app/myCustomConnector",
    type="com.extraServices.customConnector.class",
    properties = {…}
)
```

Elasticity



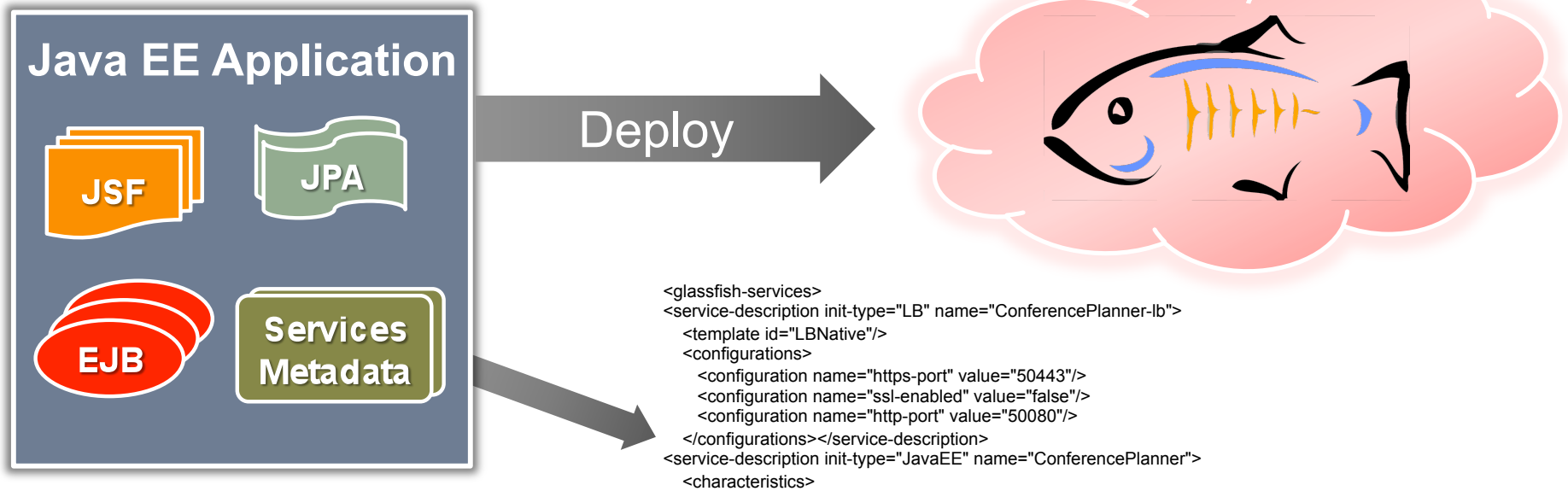
- Capacity On Demand
- Autonomic Service Level Management
- Targeting deployment from single machine to IaaS

DEMO

PaaSing a Java EE Application in the Cloud

ORACLE

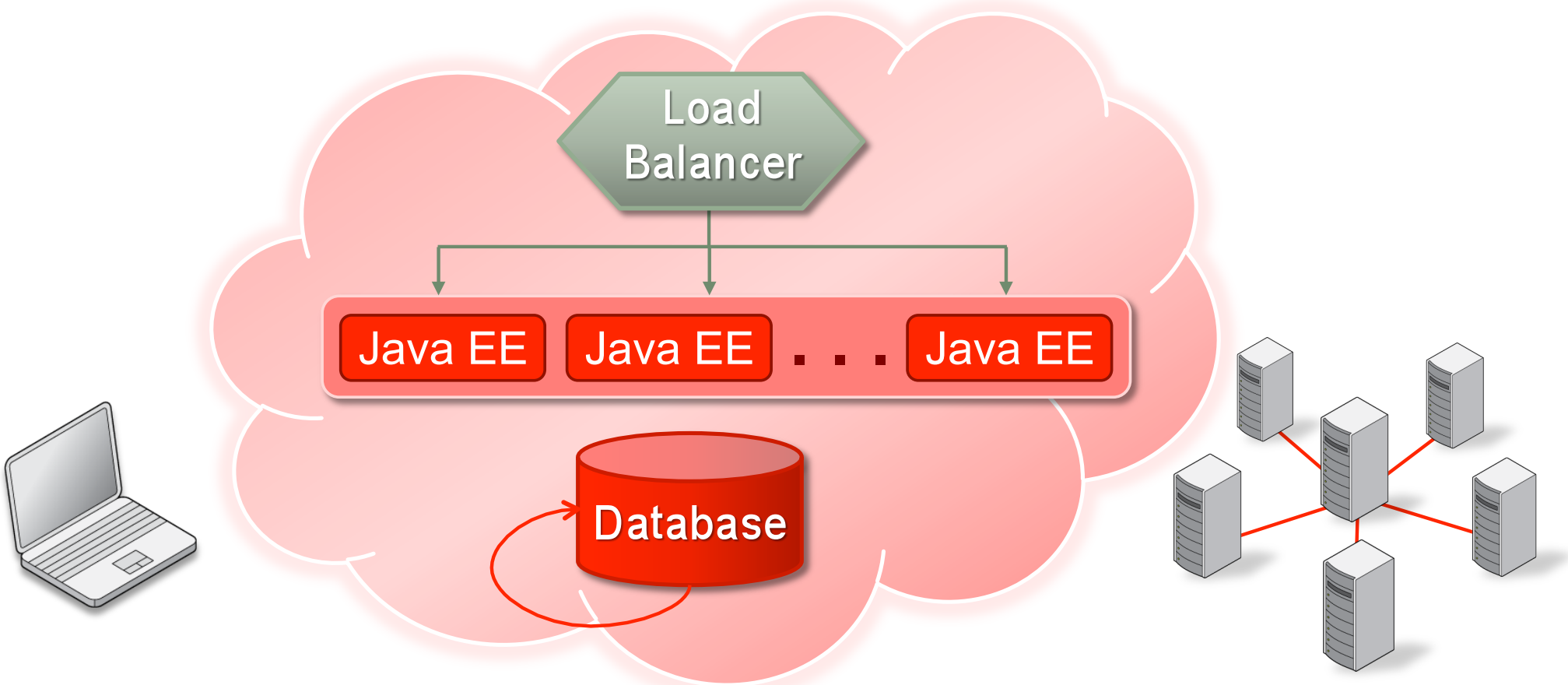
Conference Planning in the Cloud




```
<glassfish-services>
<service-description init-type="LB" name="ConferencePlanner-lb">
   <template id="LBNative"/>
   <configurations>
      <configuration name="https-port" value="50443"/>
      <configuration name="ssl-enabled" value="false"/>
      <configuration name="http-port" value="50080"/>
   </configurations></service-description>
<service-description init-type="JavaEE" name="ConferencePlanner">
   <characteristics>
      <characteristic name="service-type" value="JavaEE"/>
   </characteristics>
   <configurations>
      <configuration name="max.clustersize" value="4"/>
      <configuration name="min.clustersize" value="2"/>
   </configurations>
</service-description>
. . .
</glassfish-services>
```

http://glassfish.org/javaone2011

Service Provisioning



Multi-tenancy in Java EE 7

- Support for separate isolated instances of the same app for different tenants
  - One application instance per tenant
  - Tenants correspond to units of isolation
  - Multitenant apps are declared as such
  - Each instance customized and deployed for a single tenant
  - Limited form of SaaS
- Mapping to tenant done by the container
- Tenant id available to application
  - E.g., under java:comp/tenantId or by injection

Java EE 7 Focus Area: Application Level Multi Tenancy

- Goal: Simple configuration
- Flexible tenant discriminator support
- Other areas impacted - JNDI, JMS, EJB …

```
@Entity
@Table(name="EMP")
@MultiTenant(SINGLE_TABLE)
@TenantDiscriminator(name="company-id", columnName="COMPANY")
public class Employee {
```

EMP

EMP_ID	VERSION	F_NAME	L_NAME	GENDER	DEPT_ID
1	1	John	Doe	M	1
2	3	Jane	Doe	F	2

depart_id → DEPT_ID

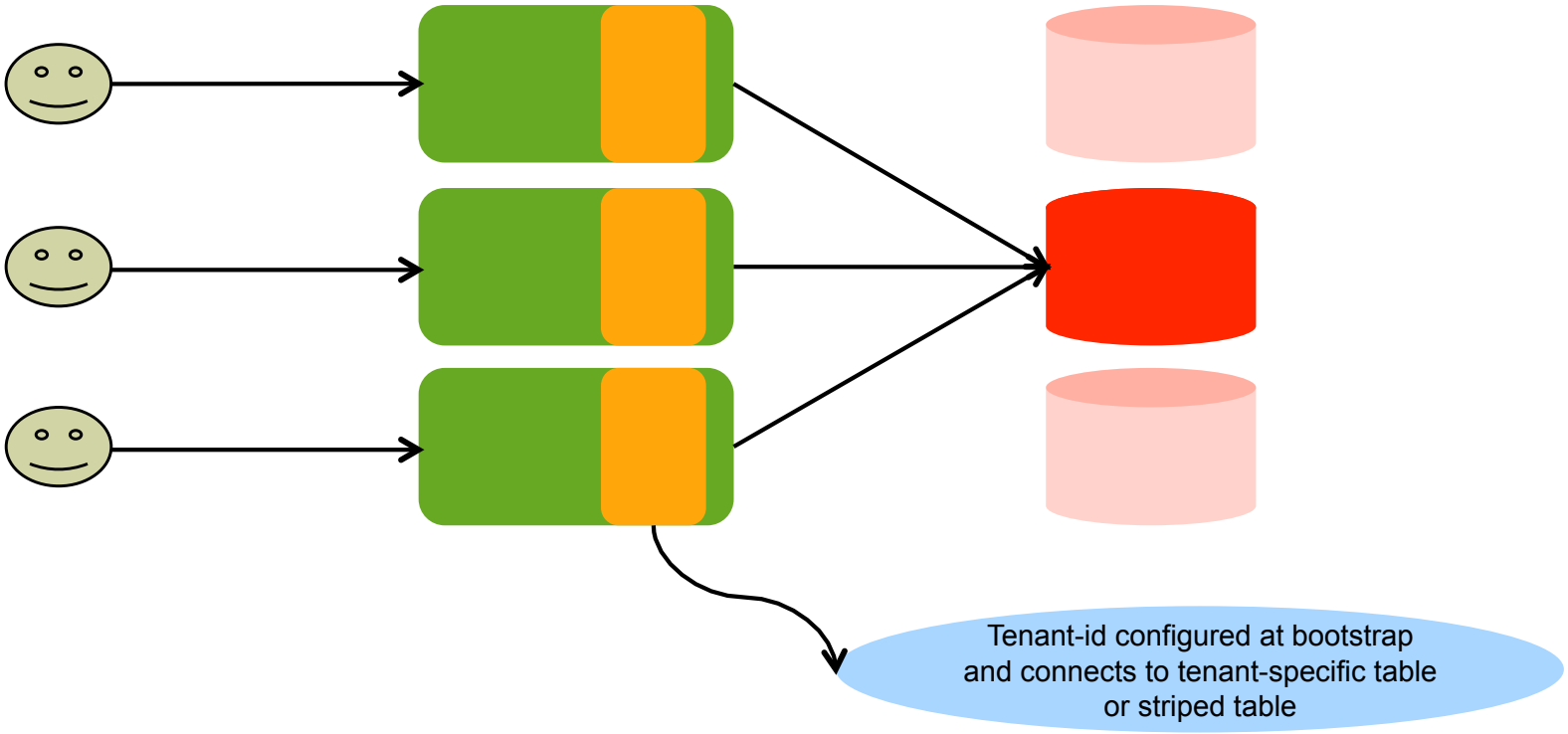
```
SELECT * FROM EMPLOYEE WHERE L_NAME LIKE 'D%' AND DEPT_ID= 1
```

Persistence Layer

- Multi-tenant Taxonomies
- Dedicated App, Dedicated Database
- Shared App, Dedicated Database
- Dedicated App, Shared Database
- Shared App, Shared Database

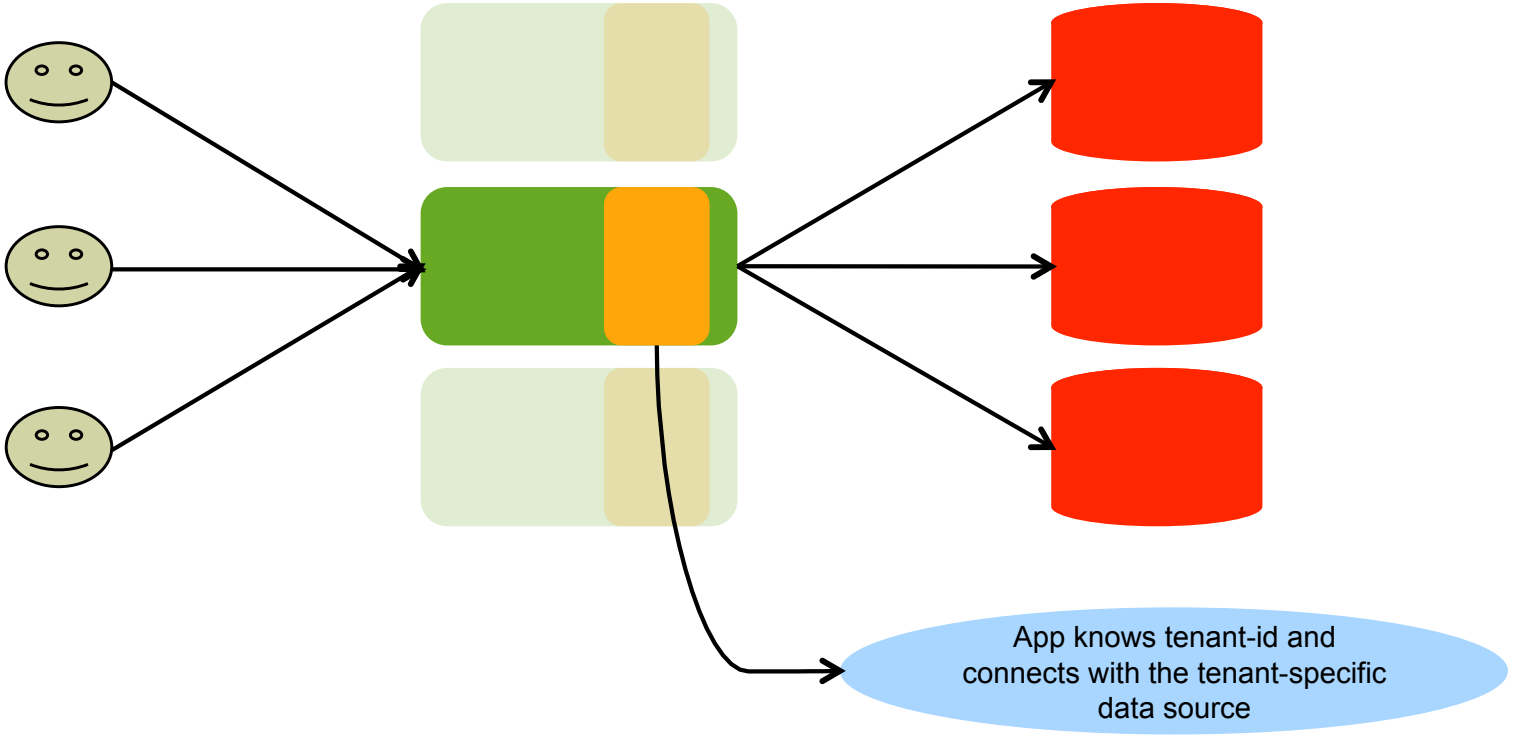
Dedicated App, Shared Database

- Persistence Layer Multi-Tenant Taxonomies



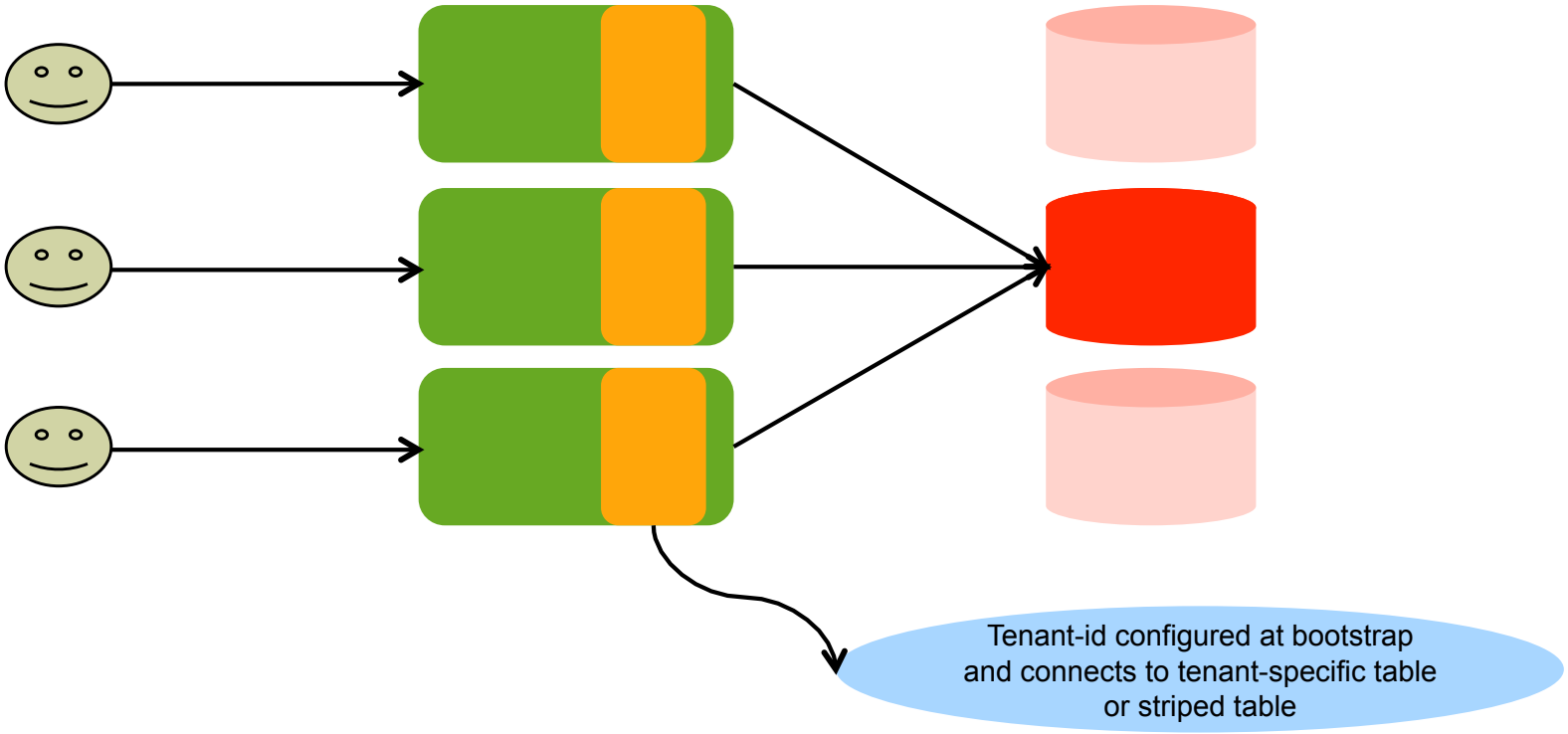
Shared App, Dedicated Database

- Persistence Layer Multi-Tenant Taxonomies



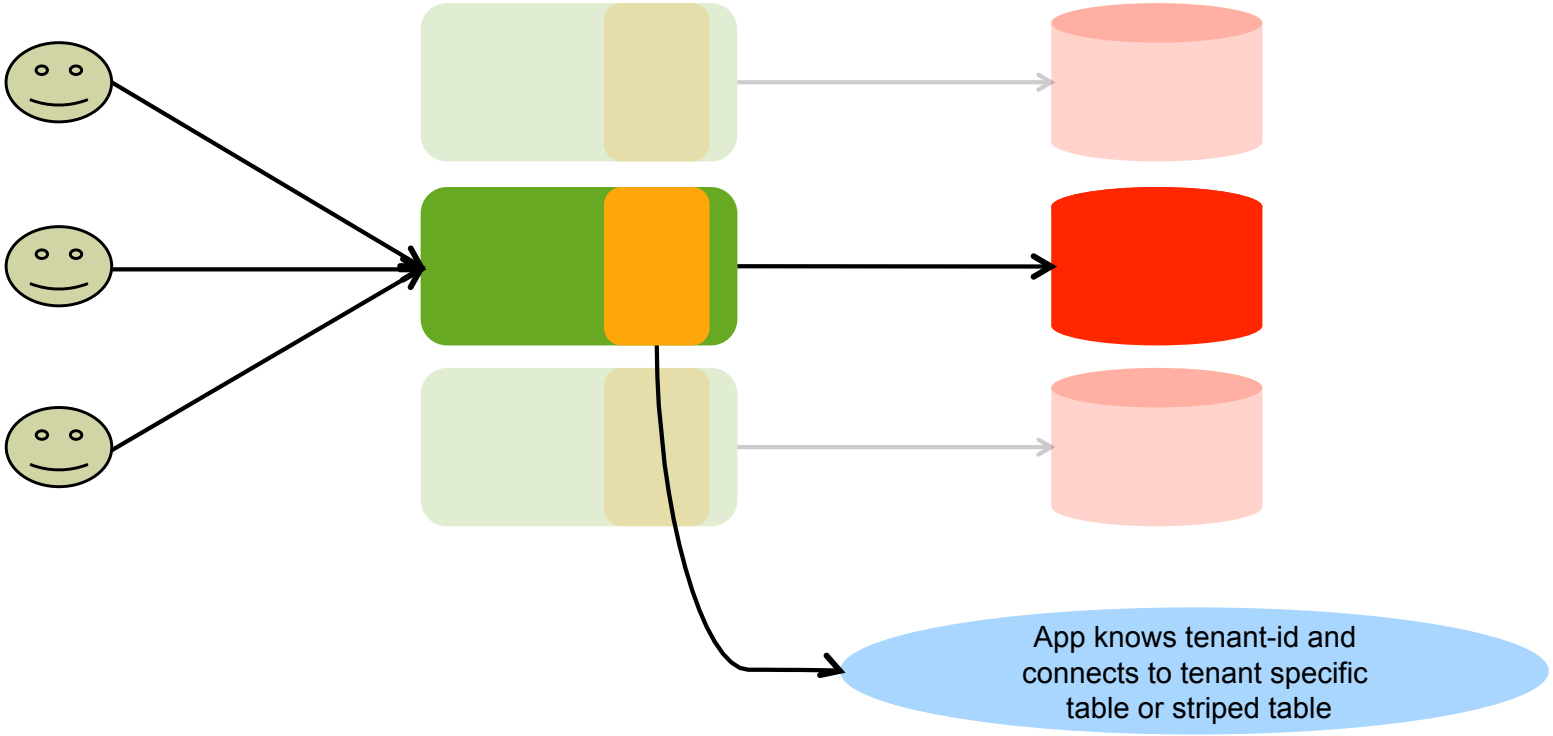
Dedicated App, Shared Database

- Persistence Layer Multi-Tenant Taxonomies



Shared App, Shared Database

- Persistence Layer Multi-Tenant Taxonomies



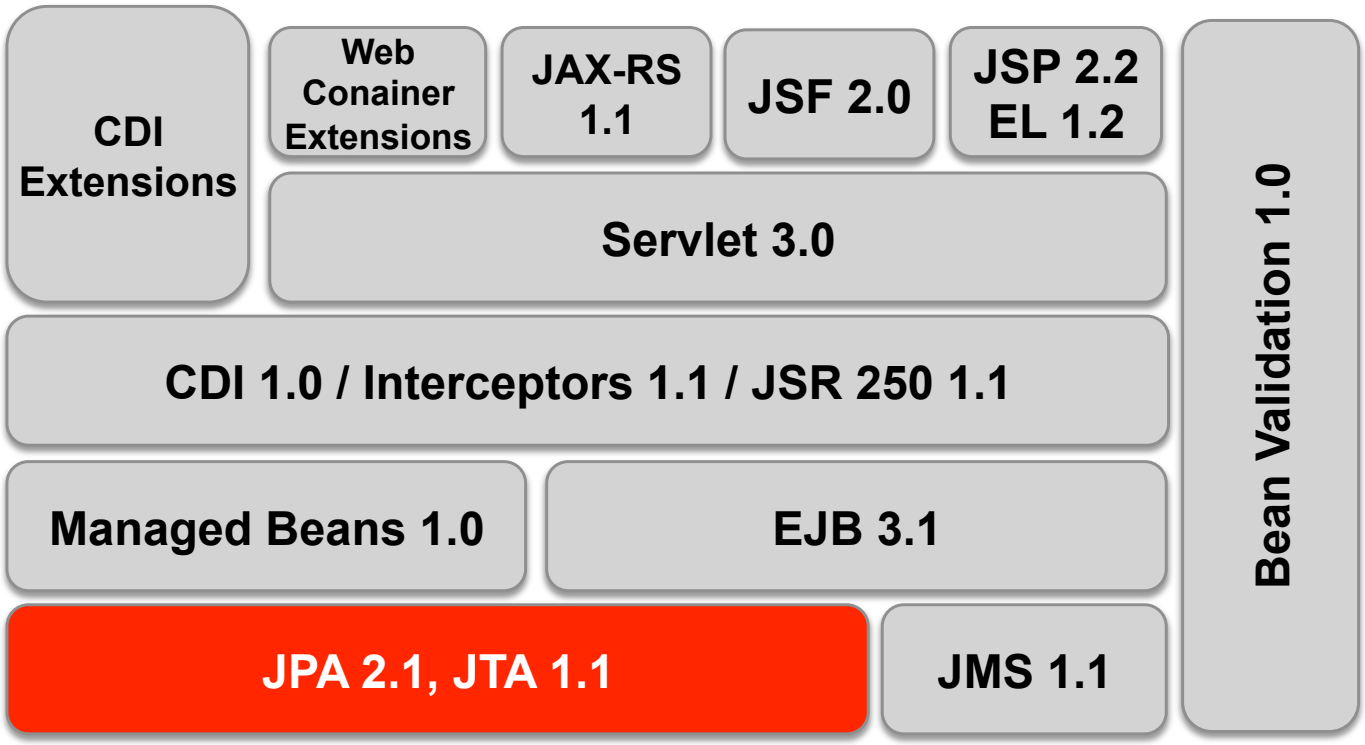
Java EE 7 Is Not Just PaaS Focused

- Alignment of ManagedBeans across CDI, EJB, JSF,…
  - POJO → ManagedBean → Enterprise JavaBean
  - Extension of container-managed transactions beyond EJB
- Further simplifications for ease-of-development
  - JMS 2.0 focus on ease-of-development
  - Expanded use of dependency injection
  - Expanded service metadata; improved configuration
- Pruning
  - EJB CMP and BMP, JAX-RPC, Deployment API
- Update to Web Profile

Java EE 7 By JSR

ORACLE OPEN WORLD

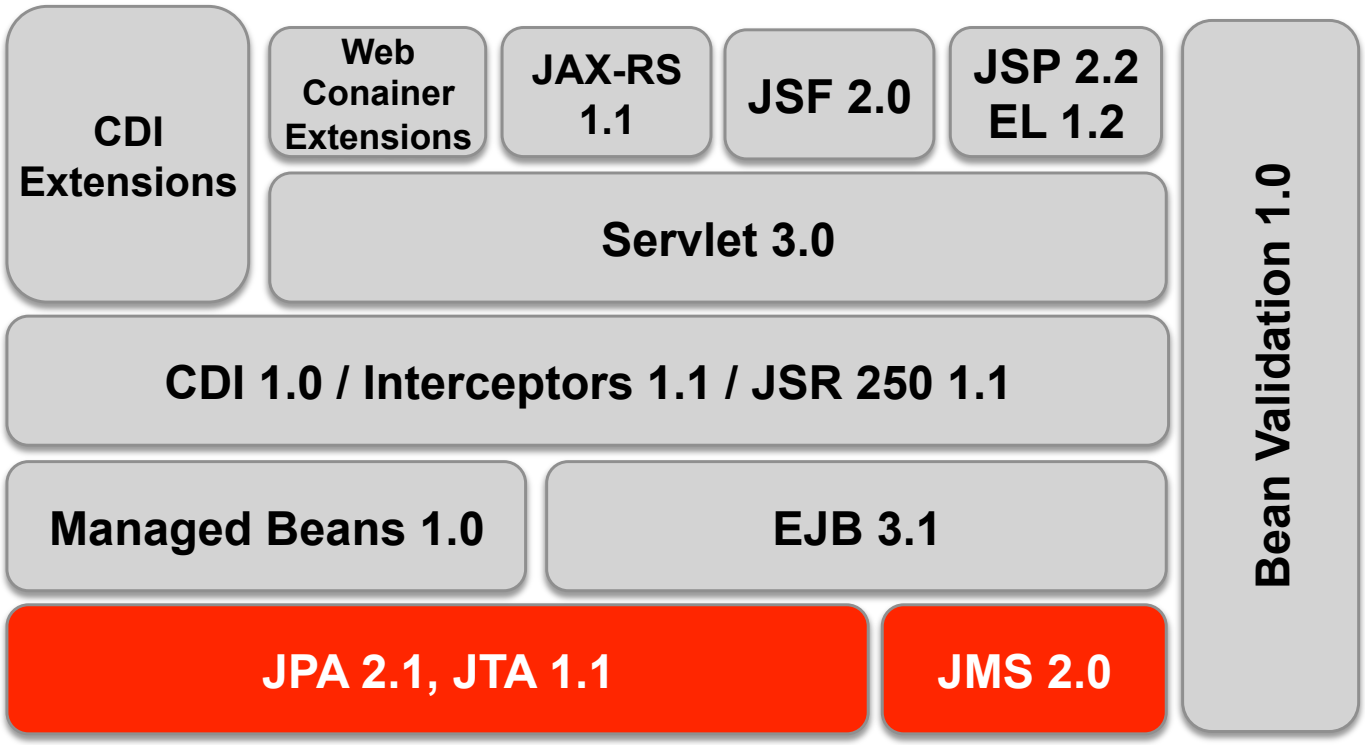
Java EE 7 – JPA 2.1 (JSR 338)



JPA 2.1

- Multi-Tenancy
- Stored Procedures
- Query by Example
- Dynamic PU Definition
- Schema Generation

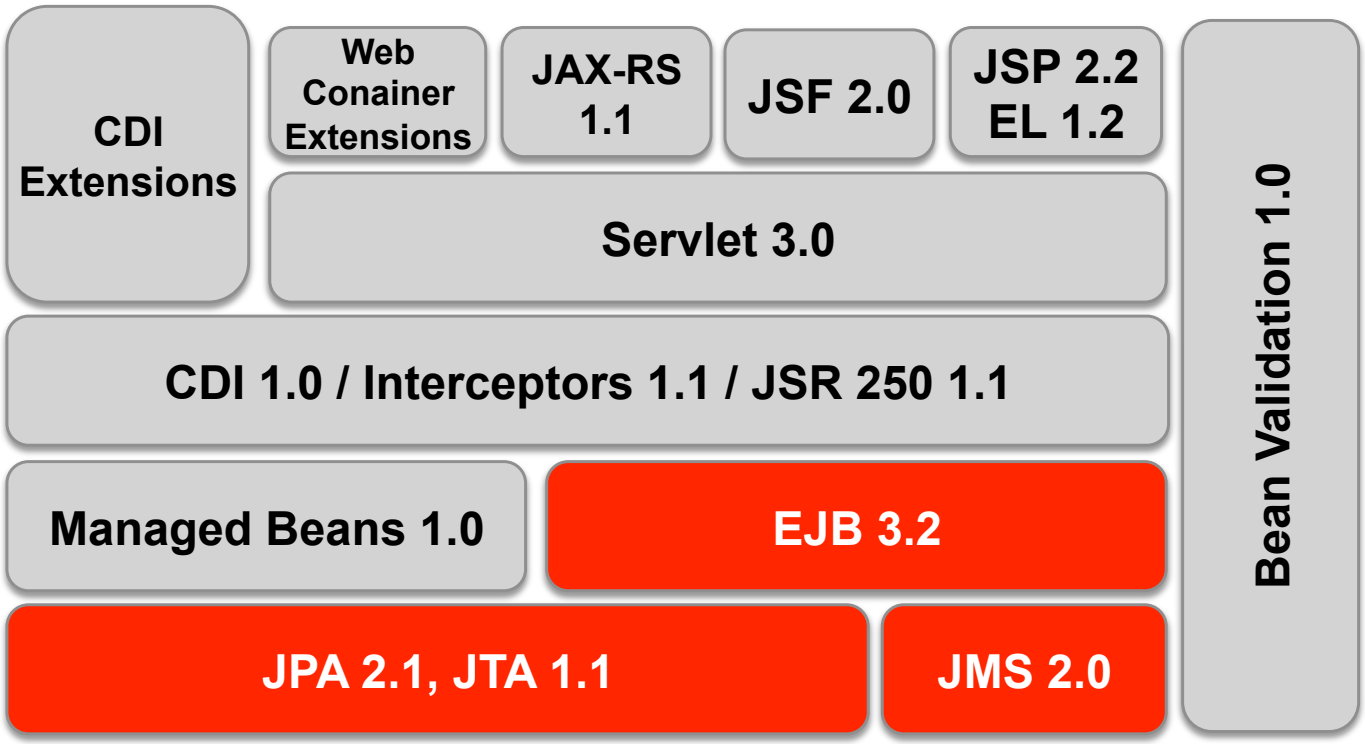
Java EE 7 – JMS 2.0 (JSR 343)



JMS 2.0

- 9 years since last update
- Modest scope
- Ease of development
- Pluggable JMS provider
- Extensions to support “Cloud”

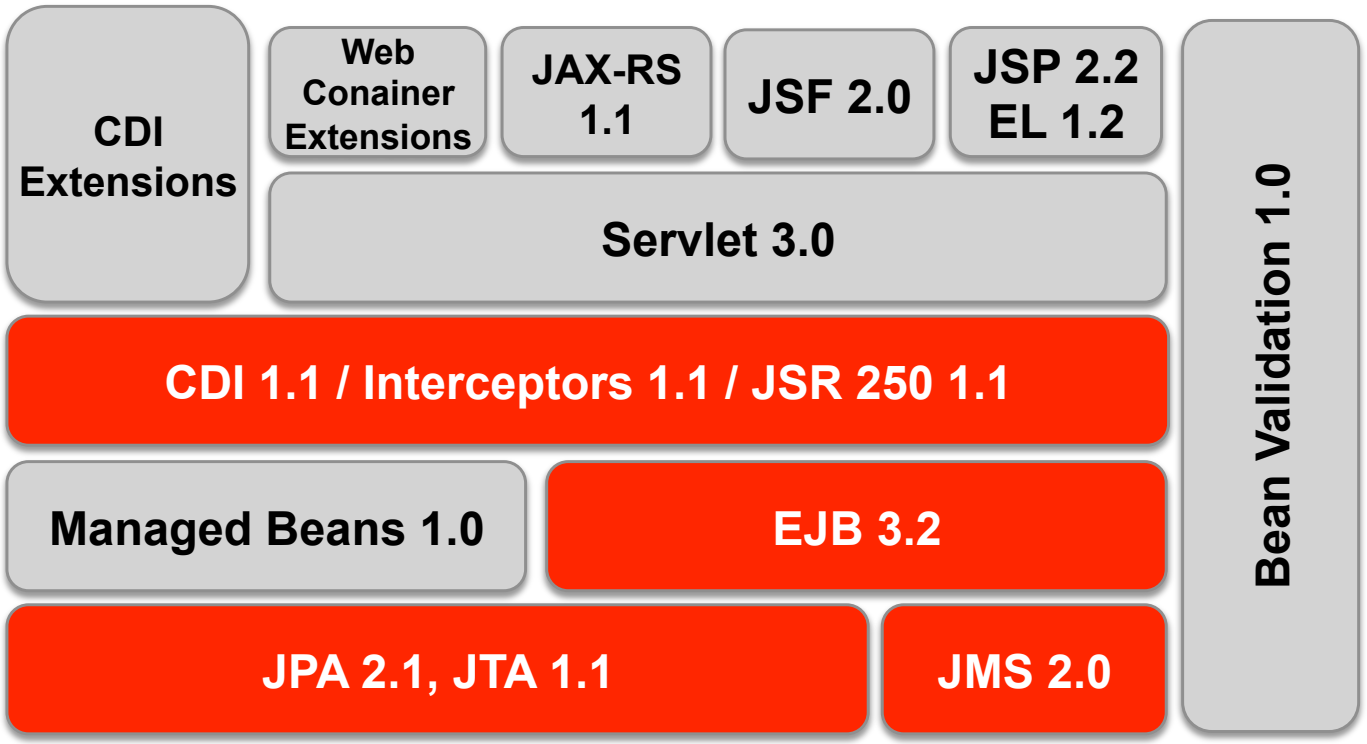
Java EE 7 – EJB 3.2 (JSR 345)



EJB 3.2

- Incremental factorization
- More annotations
- Optional
  - BMP/CMP
  - Web Svcs invocation using RPC

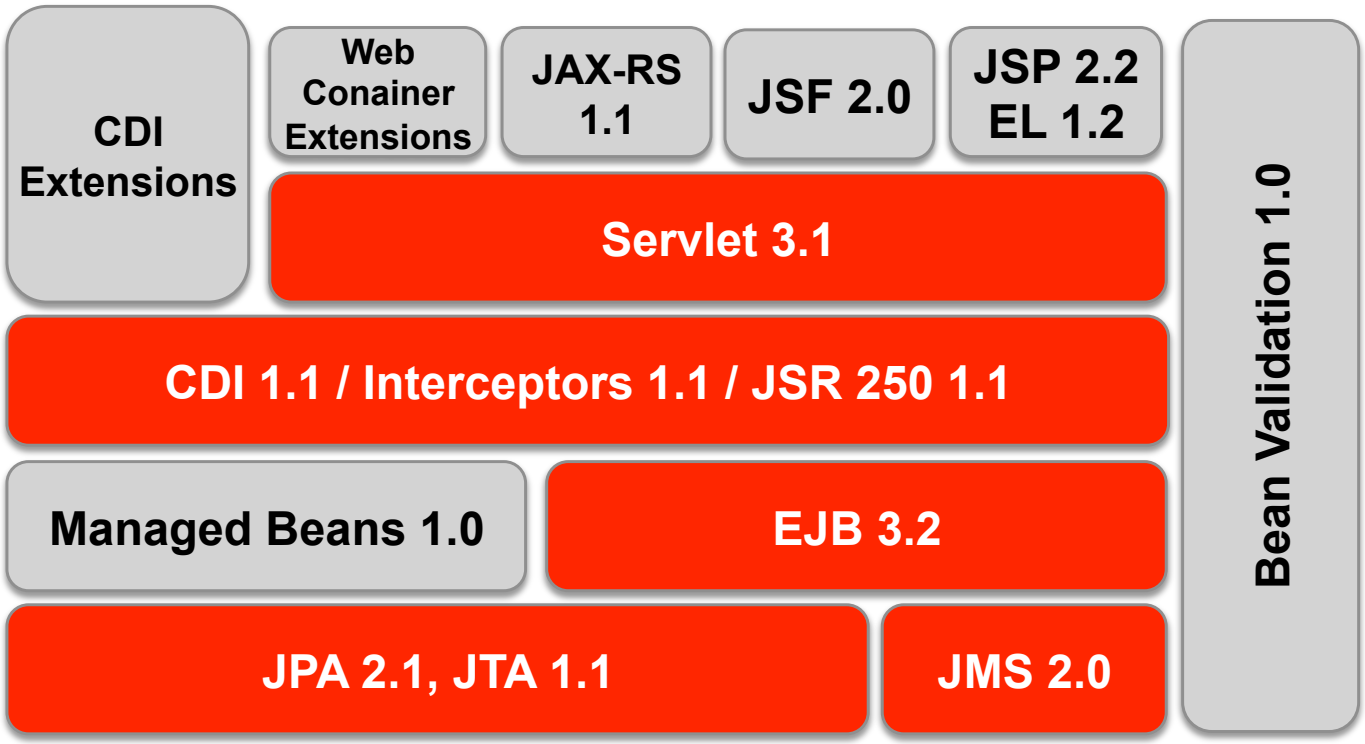
Java EE 7 – CDI 1.1(JSR 346)



CDI 1.1

- Embedded mode
- Lifecycle Events
- Declarative package scanning
- Global ordering of interceptors and decorators
- Injection Static Variables

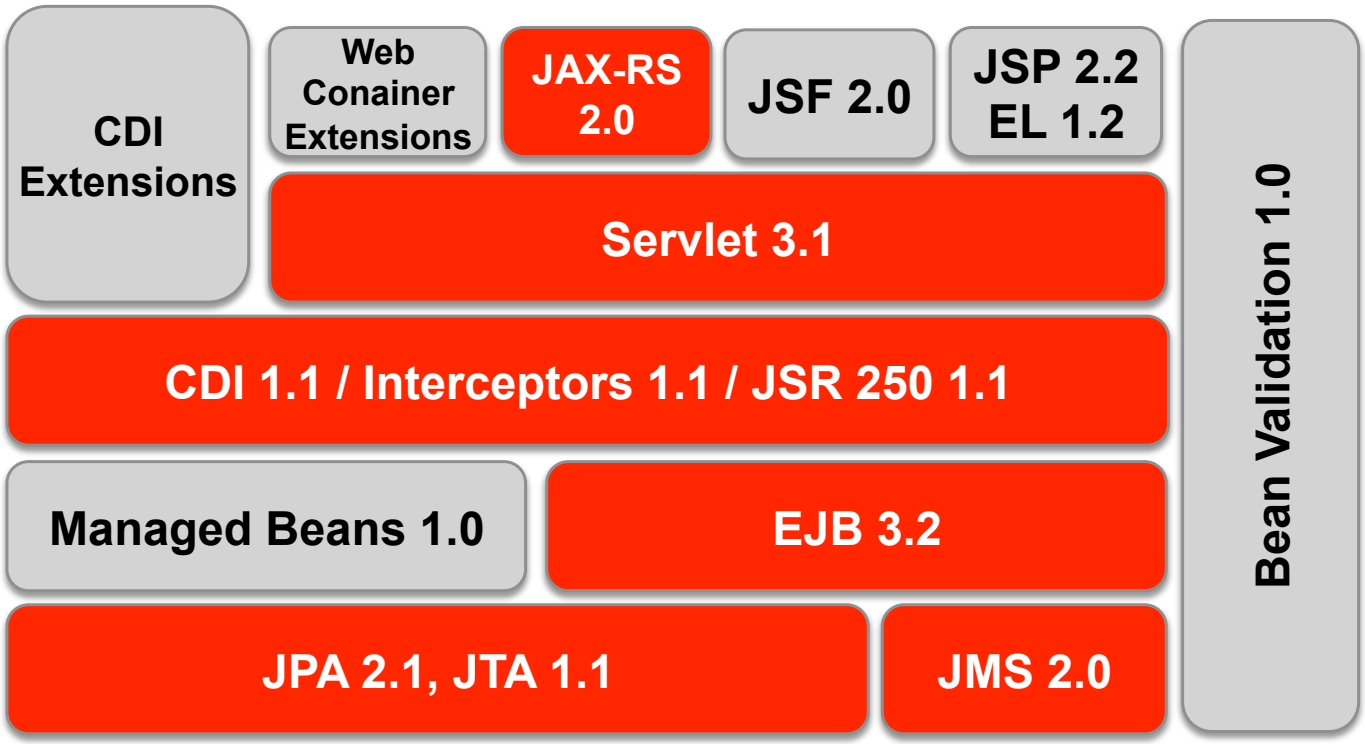
Java EE 7 - Servlet 3.1 (JSR 340)



Servlet 3.1

- NIO.2 Async I/O
- Leverage Java EE concurrency
- Web Sockets support
- Ease of Use

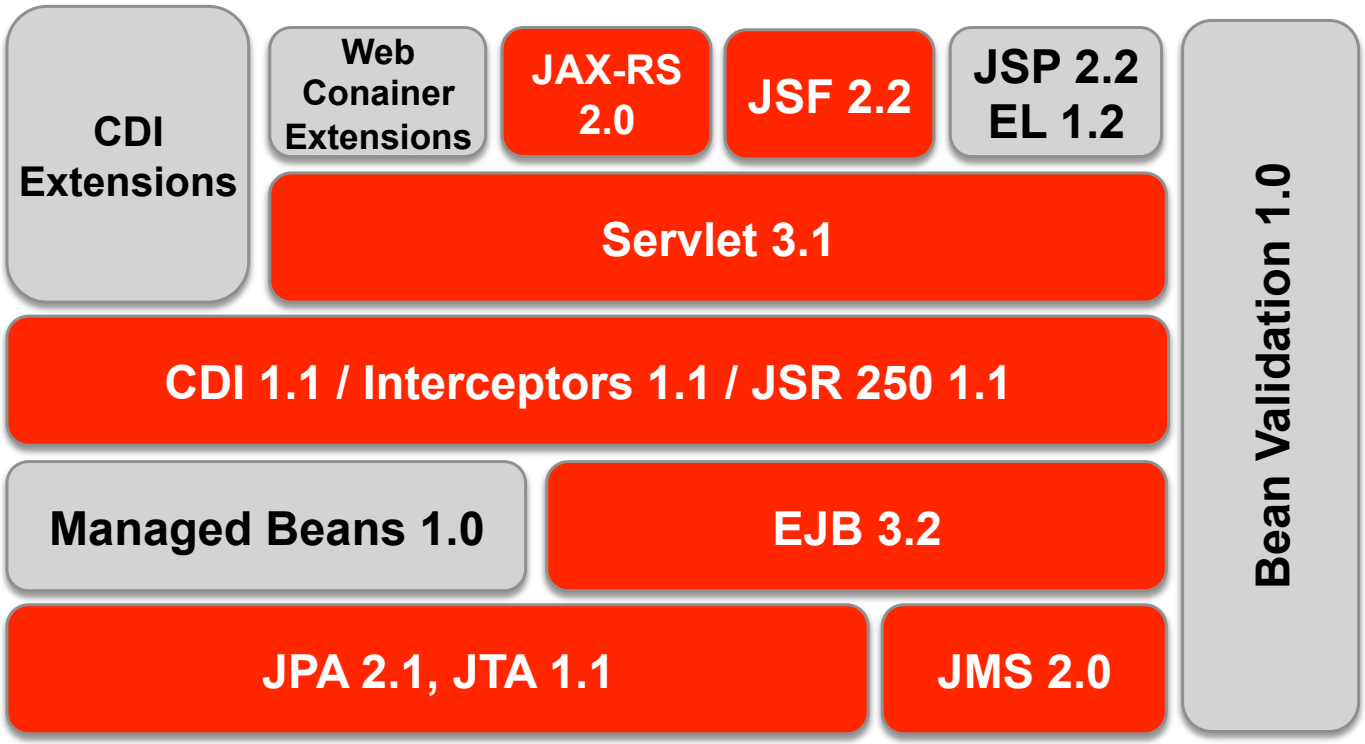
Java EE 7 – JAX-RS 2.0 (JSR 339)



JAX-RS 2.0

- Client API
- Hypermedia
- Bean Validation

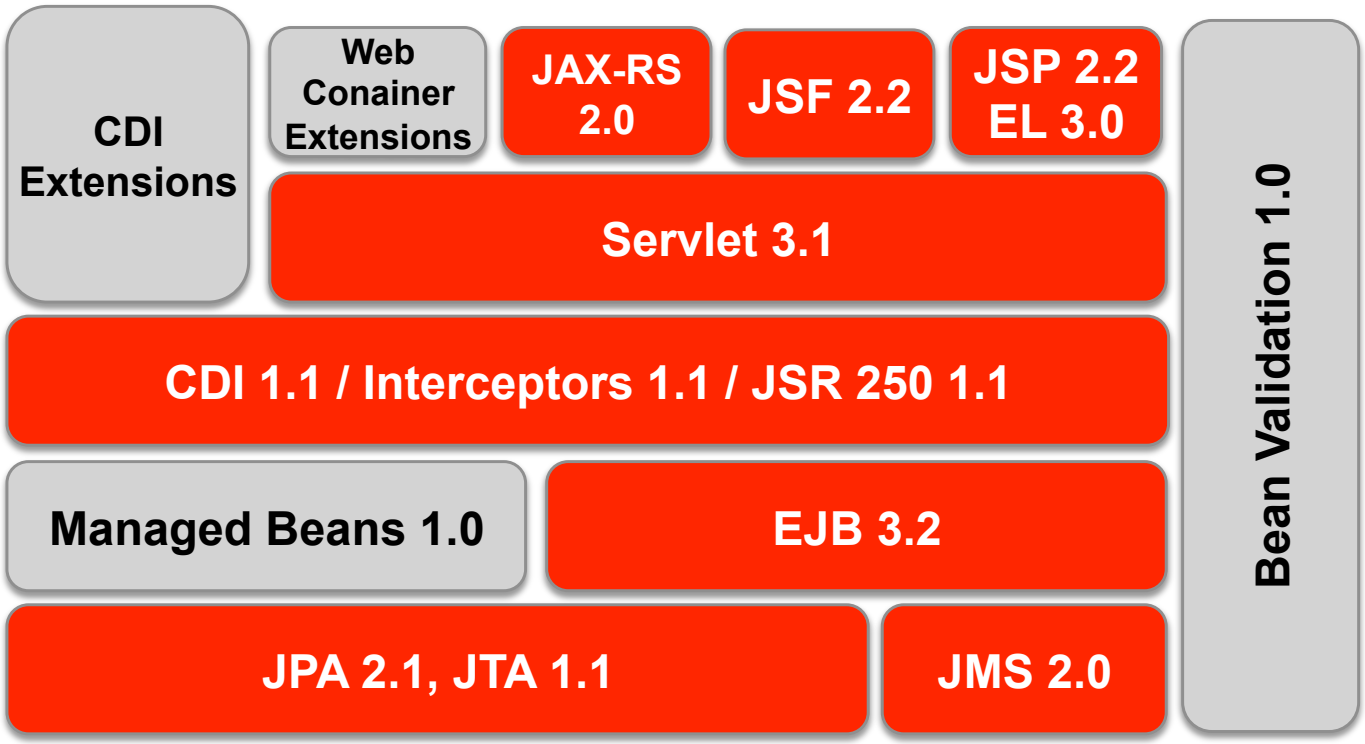
Java EE 7 – JSF 2.2 (JSR 344)



JSF 2.0

- Ease of development
- HTML 5 support
- New components
- Portlet Integration

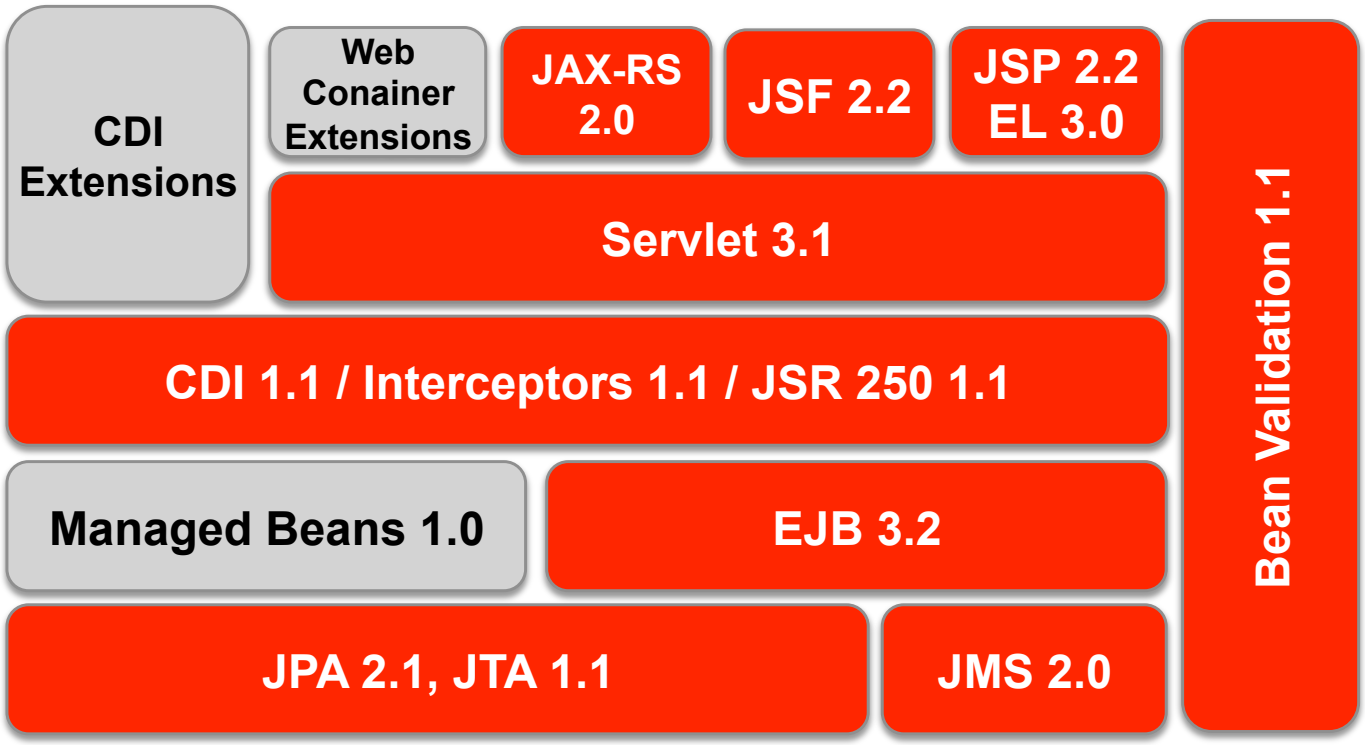
Java EE 7 – Expression Language 3.0 (JSR 341)



EL 3.0

- Standalone JSR
- Easier to use outside container
- Criteria-based Collection selection
- New operators
- CDI events for expression eval

Java EE 7 – Bean Validation 1.1 (JSR 349)



Bean Validation 1.1

- Integration with other JSRs (JAX-RS, JAXB, JPA)
- Method-level validation
- Compound Expressions

Transparency

- Our Java EE 7 JSRs are run in the open on java.net
  - http://javaee-spec.java.net
  - One project per spec – e.g., jpa-spec, jax-rs-spec, jms-spec…
- Publicly viewable Expert Group mail archive
  - Users observer list gets copies of all Expert Group emails
- Publicly viewable download area
- Publicly viewable issue tracker
- Commitment to match JCP 2.8 Process

Status and Schedule

- Nearly all JSRs up and running
- Remaining ones to be filed in next few weeks
- Final release target: Q3 2012
- Date-driven release: anything not ready will be deferred to Java EE 8

How to Get in the Loop

- Java EE 7 Expert Group Project
  - http://javaee-spec.java.net
- Java EE 7 Reference Implementation
  - http://glassfish.org
- The Aquarium
  - http://blogs.oracle.com/theaquarium

Hardware and Software

ORACLE

Engineered to Work Together

