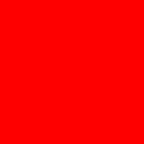


ORACLE®

Java EE: Wohin die Reise geht

Peter Doschkinow
Senior Java Architect

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technologie und Architektur



The following/preceding 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.

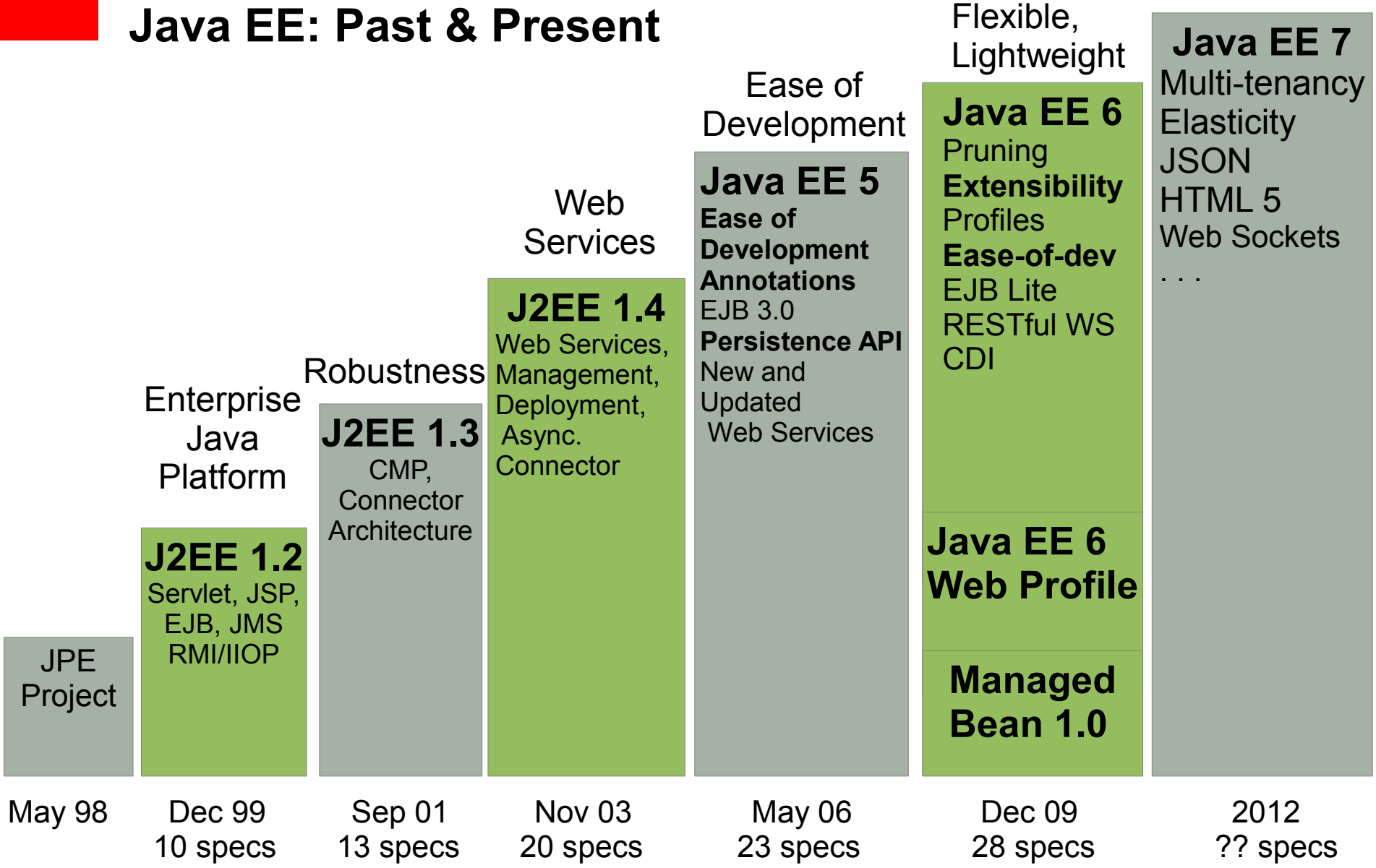
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Java EE 6 adoption and state-of-the-art
- Java EE 7 and the road ahead

Java EE: Past & Present

Cloud



Java EE 6: Fast Uptake

Within Both Open Source and Commercial Community

Today:



Oracle GlassFish 3.x



JEUS 7



Resin 4.0.17



AS7

Web Profile Only



WAS 8.0

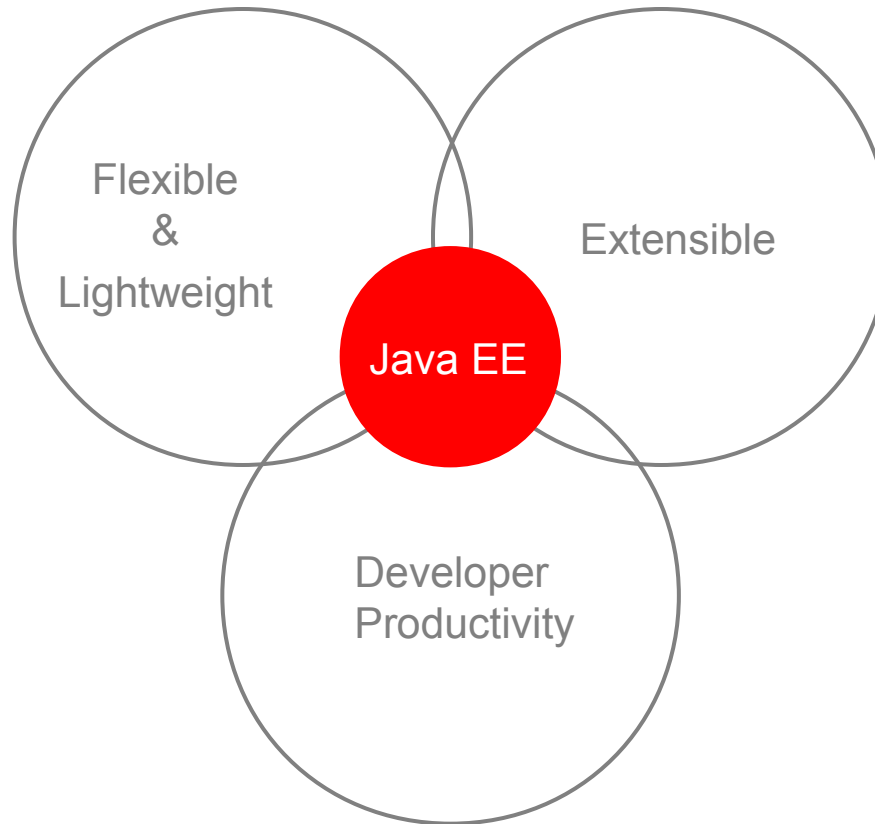
Oracle WebLogic Suite 11g

Announced:



Major Themes of Java EE 6

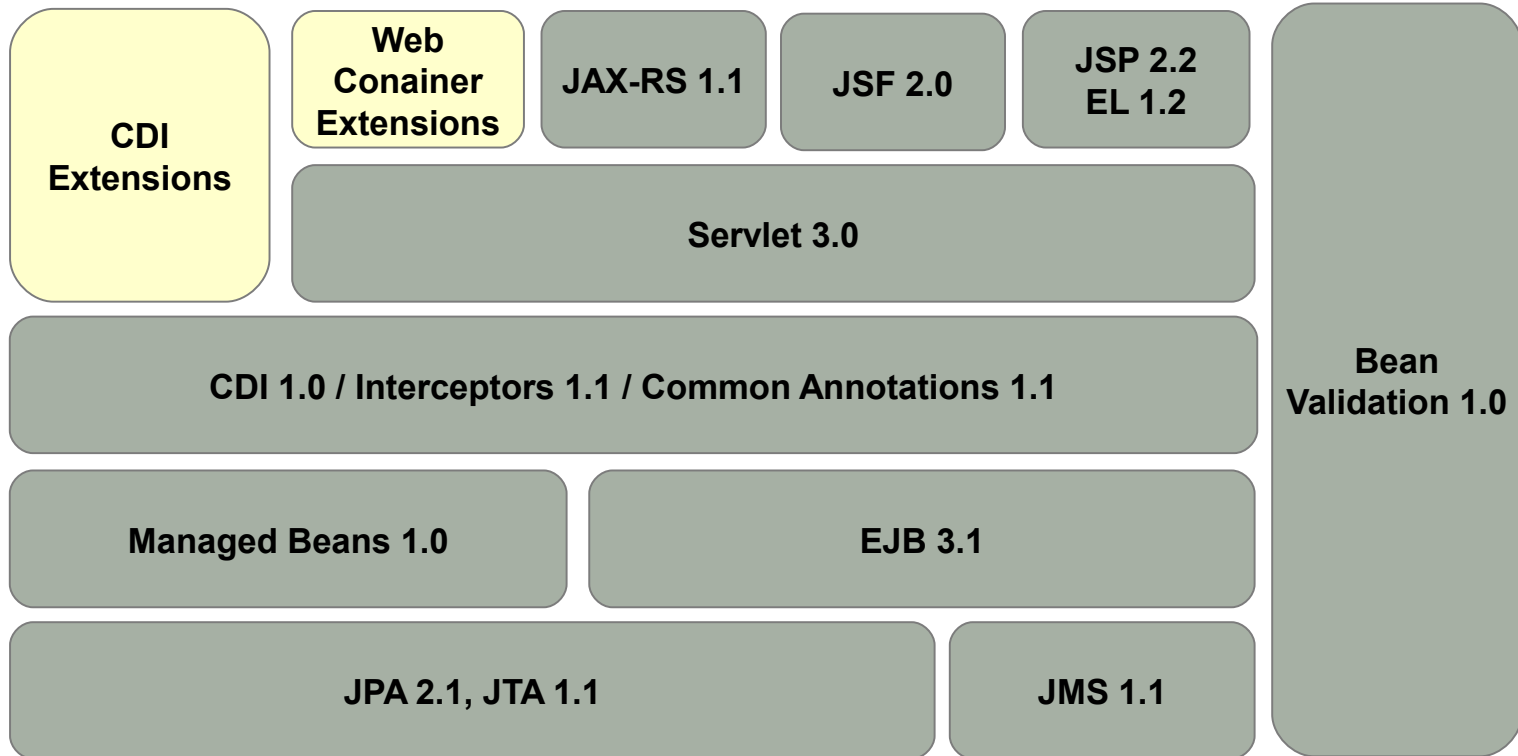
- Web Profile
- EJB Lite
- Pruning



- Embrace open source frameworks
- Enables Drag & Drop framework installation

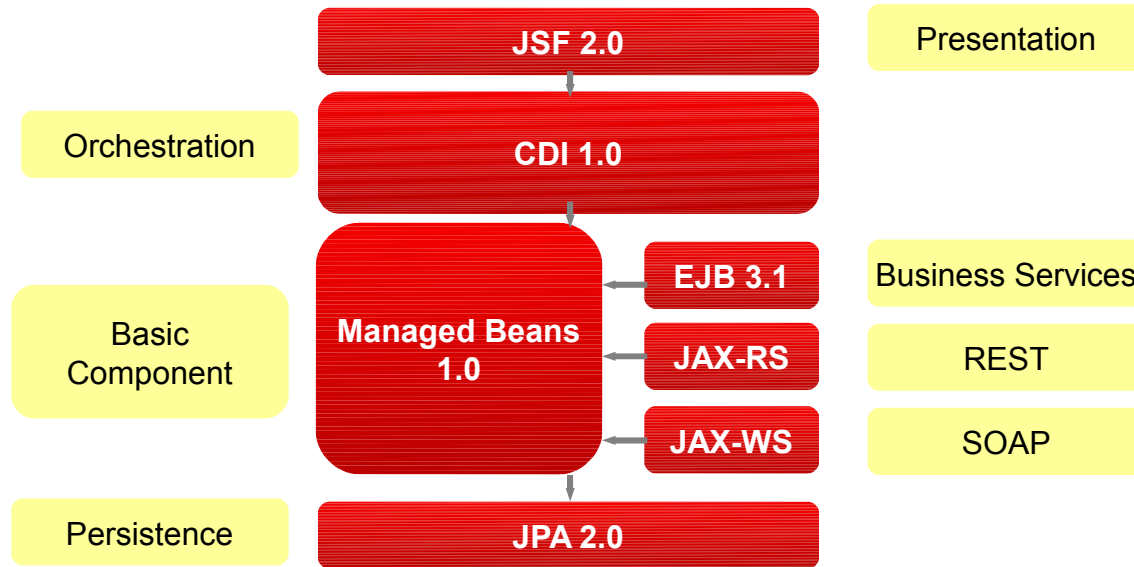
- More annotations
- POJO development
- Less XML configuration

Java EE 6 Core Technologies



- Extension hooks provided to keep up with new technologies

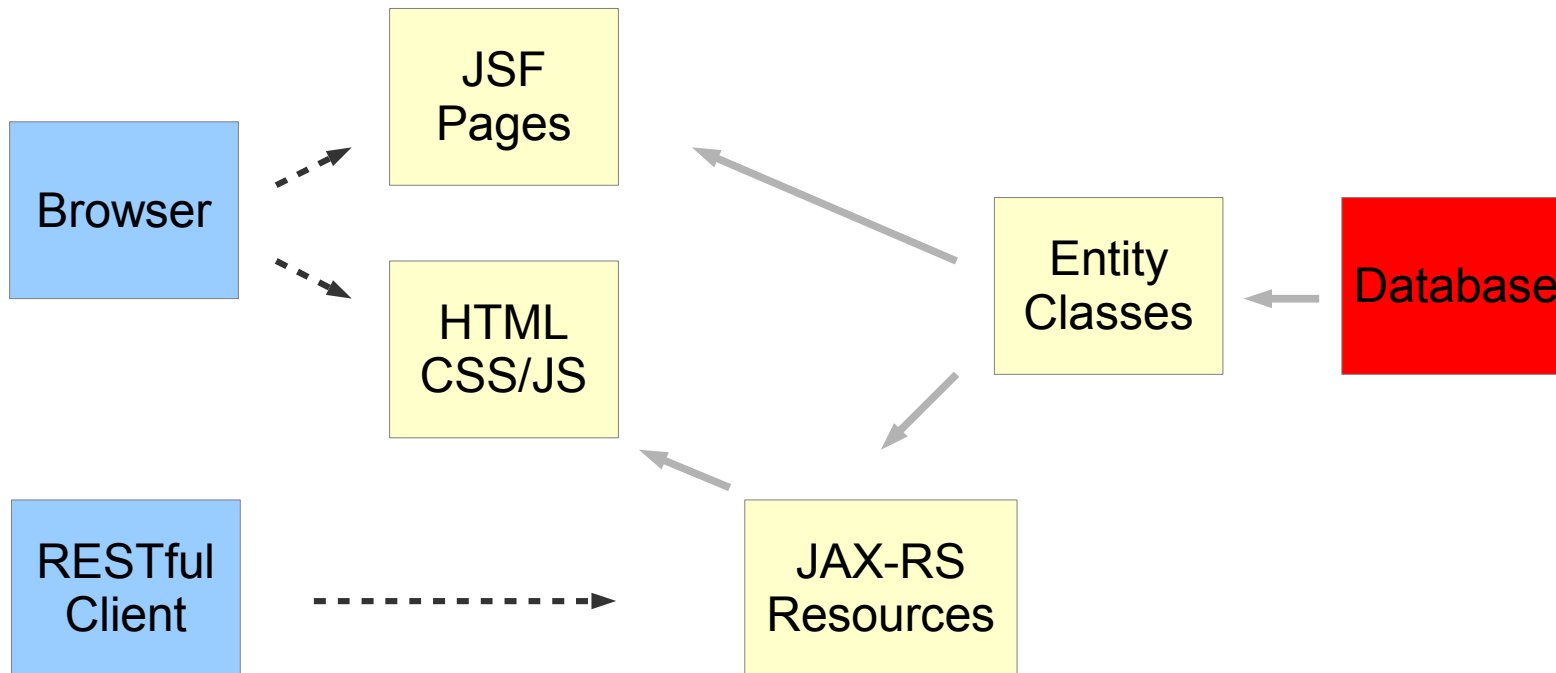
The Java EE 6 Programming Model



- Complementary: declarative and programmatic layers
- Annotations are additive
- Incremental programming: code evolves gradually
 - Managed beans as a common foundation

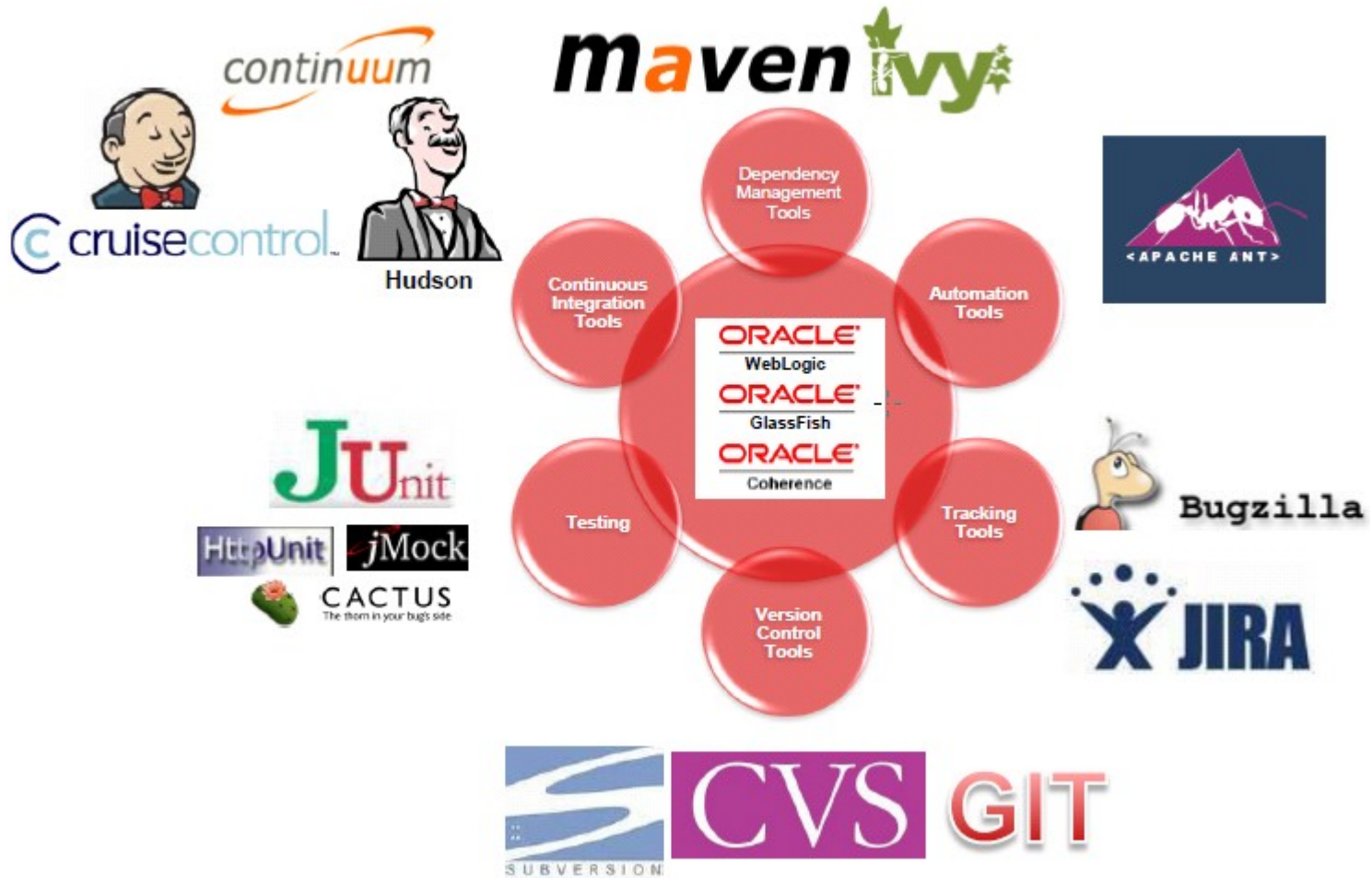
IDE Support for Java EE 6

Extensive Tooling from Schema to GUI



F/OSS Java tools ecosystem

Leverage free and open source software



8 Reasons to use Java EE 6

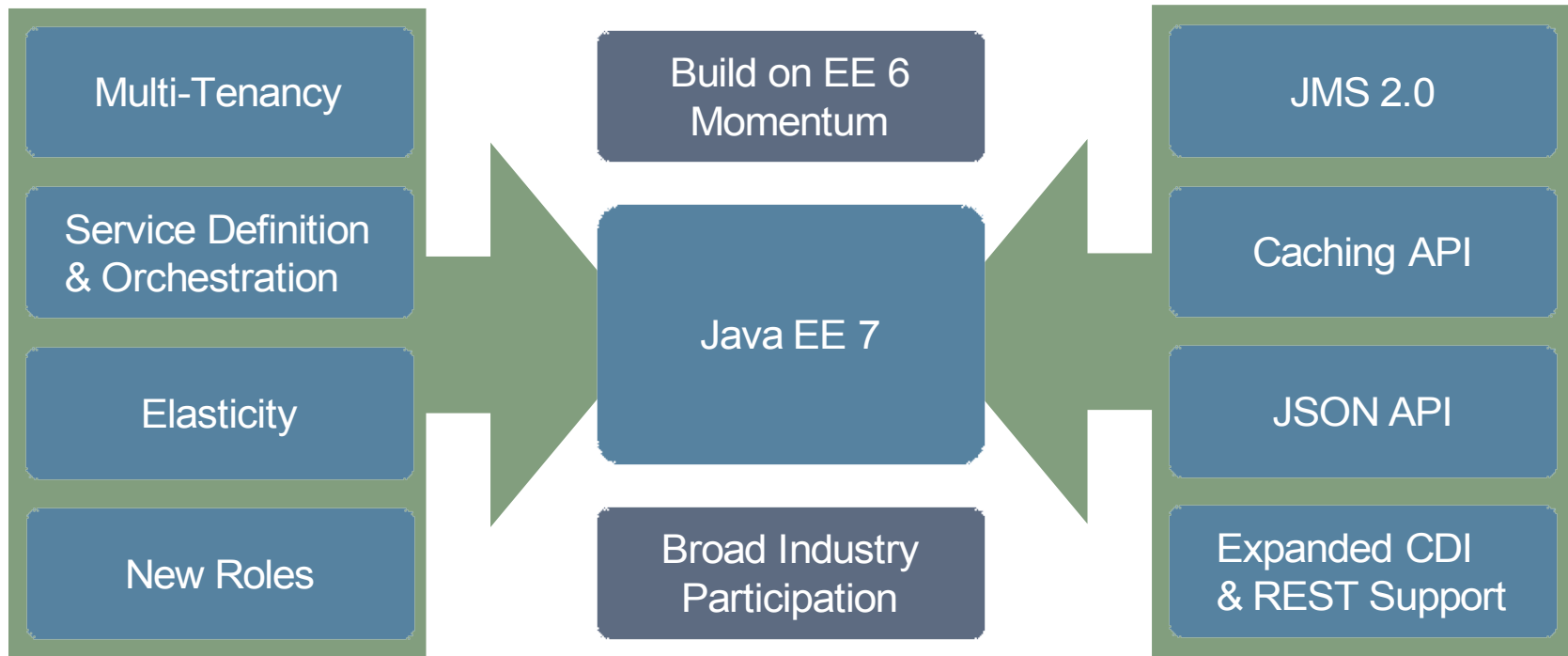
- Prototyping (multiple IDEs)
- Development (~30MB, incremental deployment, ...)
- Production (Variety, Start small/then scale)
- Support (Pick the best one)
- Training (“Only” Java EE 6 APIs)
- Portability (Backwards compatibility)
- Adoption (Growing)
- Freedom of choice (Multiple vendors)

http://www.adam-bien.com/roller/abien/entry/8_reasons_why_java_ee

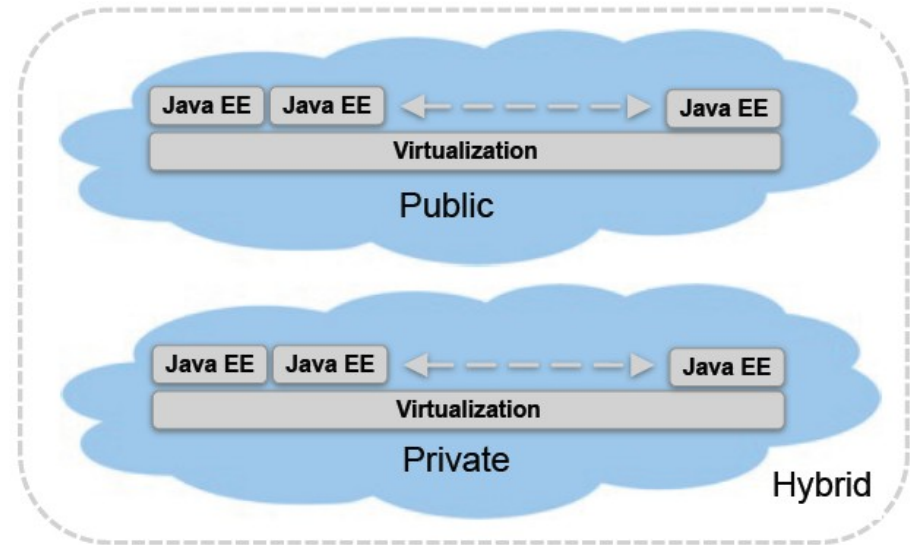
Java EE 6 Platform Summary

- More powerful
- More flexible
- More extensible
- Easier to use

Java EE 7 Design Goals

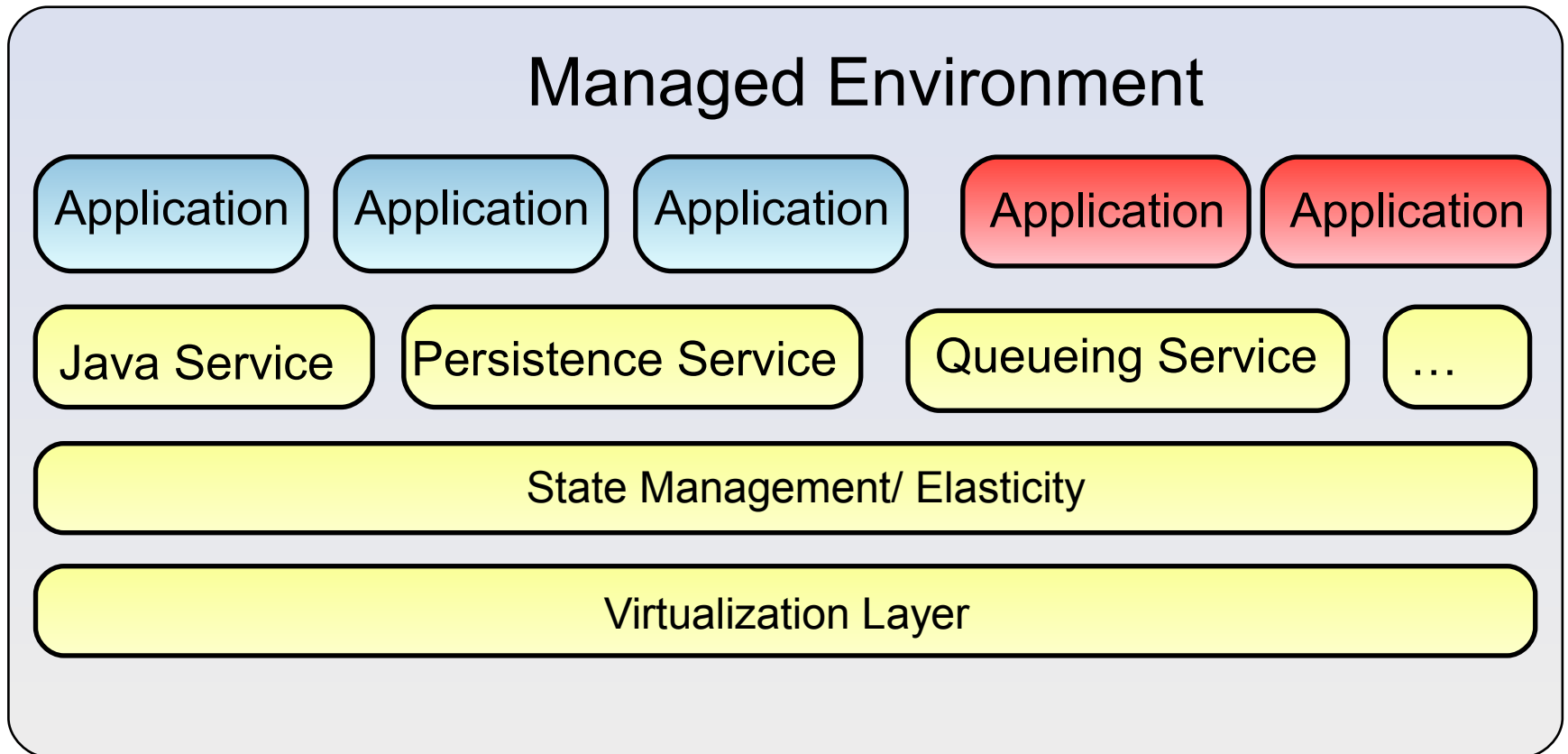


Java EE 7 Focus: PaaS



- 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 as PaaS



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 multitenancy

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")
```

PaaS and Multi-tenancy: Some Models

- PaaS Platform on Demand
 - New runtime stack for each tenant
- PaaS Multitenant Containers
 - Isolated application partitions per tenant with shared runtime
- SaaS Multitenant Applications (SaaS-full)
 - Shared application instances, with tenant-specific customization
- SaaS-limited
 - Separate application instances, with tenant-specific customizations

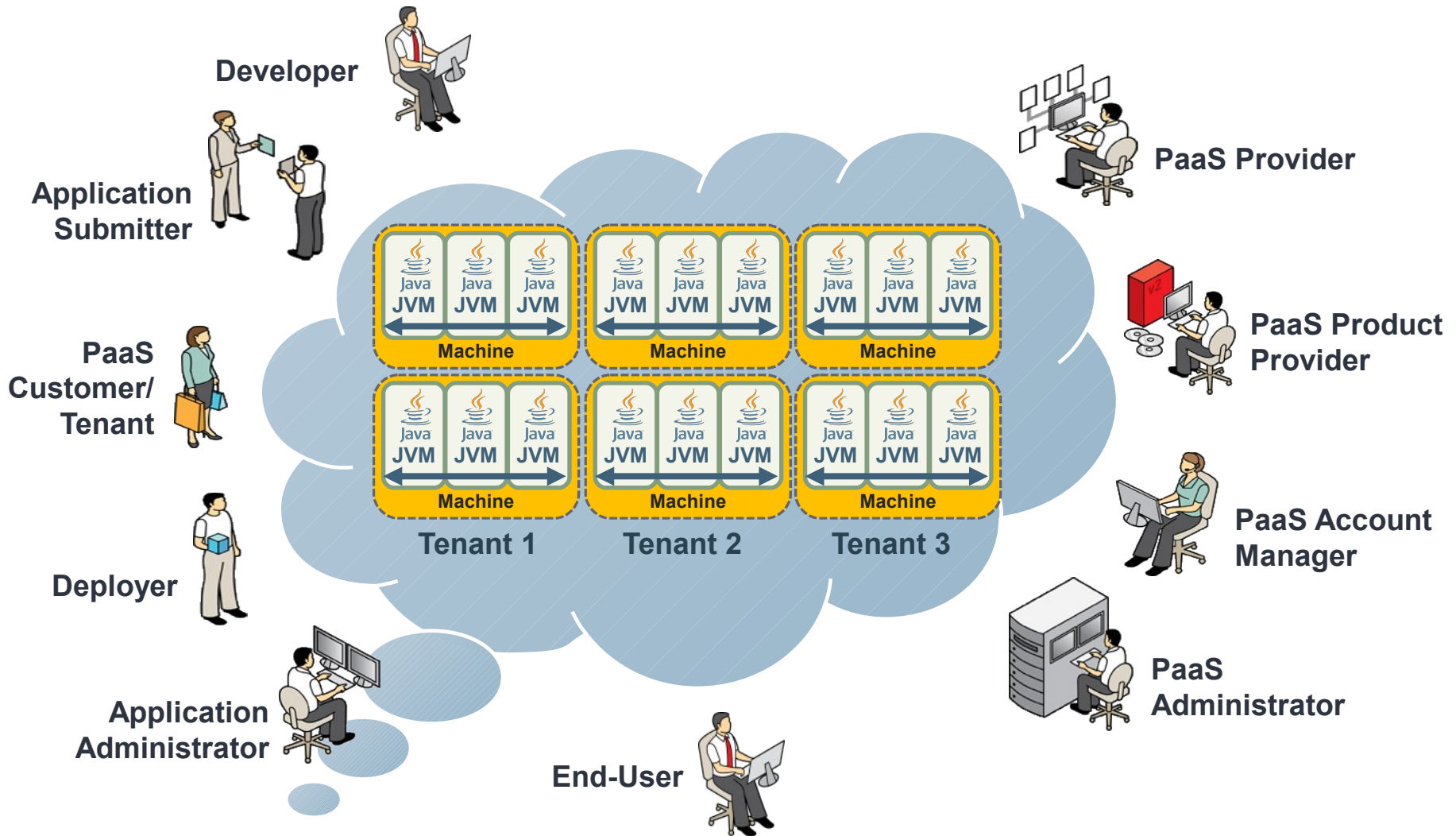
Java EE 7 Multitenancy

- 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 is not just Cloud-y

- Alignment of ManagedBeans across CDI, EJB, JSF,...
 - POJO \square ManagedBean \square 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 Roles



Java EE 7 JSRs

- Platform 7 / Web Profile 7
- JPA 2.1
- JAX-RS 2.0
- EJB 3.2
- JMS 2.0
- Servlet 3.1
- EL 3.0
- JSF 2.2
- CDI 1.1
- Bean Validation 1.1
- JCache 1.0 (JSR 107)
- Concurrency Utilities 1.0
- State Management 1.0
- Batch Processing 1.0
- JSON 1.0
- Common Annotations 1.2 MR
- JAX-WS 2.3 MR
- JTA 1.2 MR
- JSP 2.3 MR
- JASPIC 1.2 MR

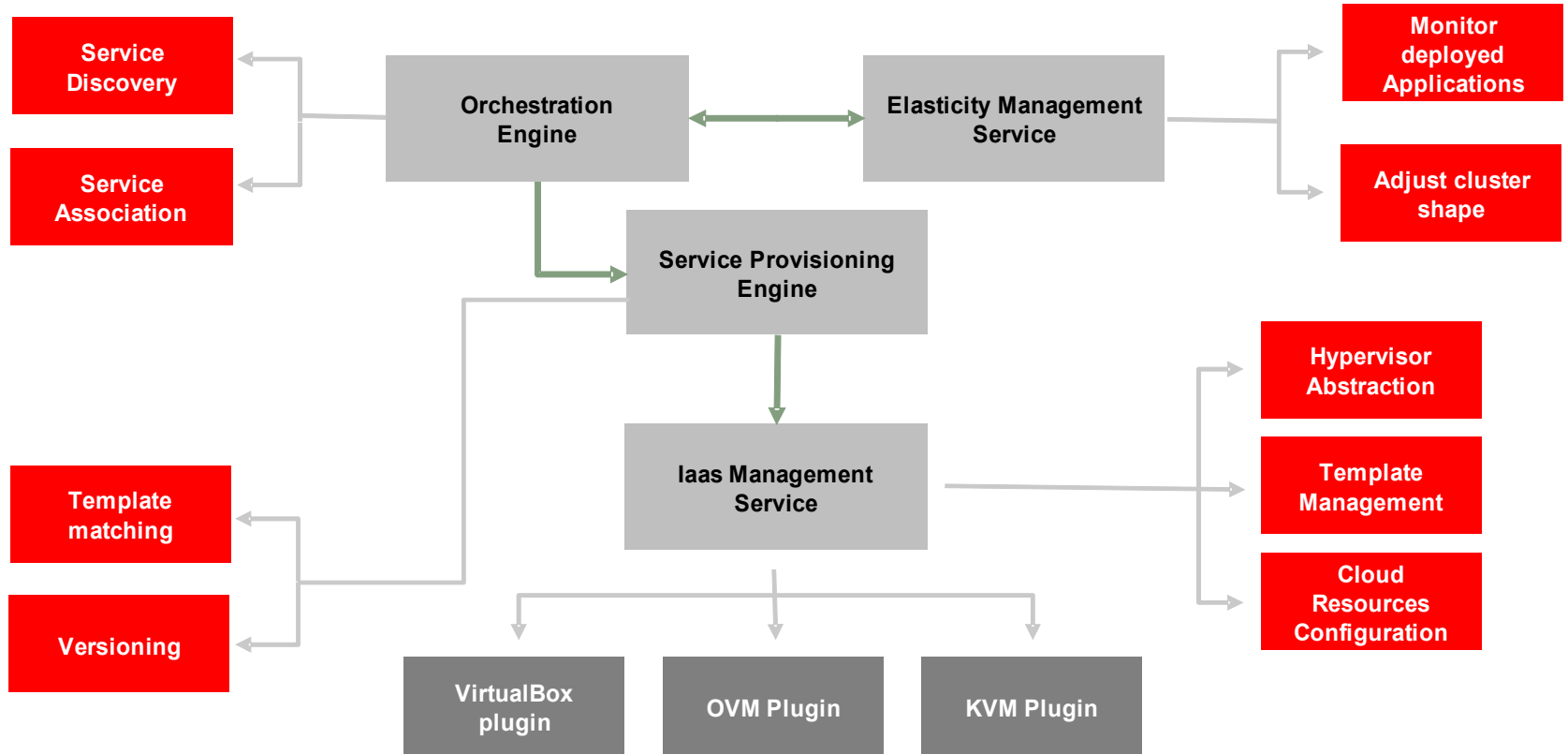
Transparency Checklist

<http://jcp.org/en/resources/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

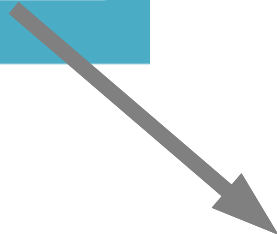
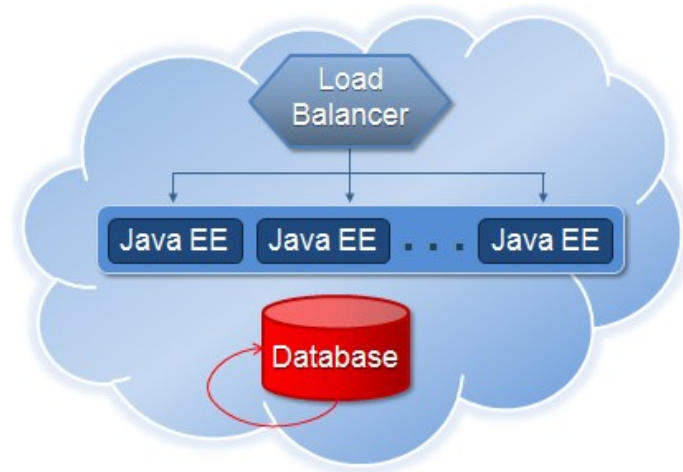
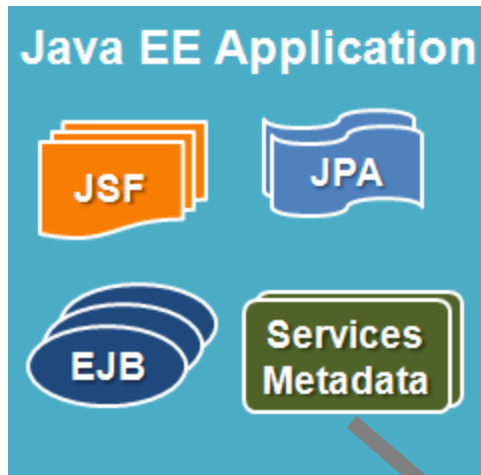
- Java EE 7 reference implementation
- Flexible deployment topology
 - Traditional GlassFish Server deployment
 - PaaS deployment
- Transparent development
 - GlassFish 3.2 ~ GlassFish 4.0

GlassFish in the Cloud



GlassFish 4.0 Demo at JavaOne

<http://glassfish.org/javaone2011>



```
<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>
```

Java EE 7 Status and Schedule

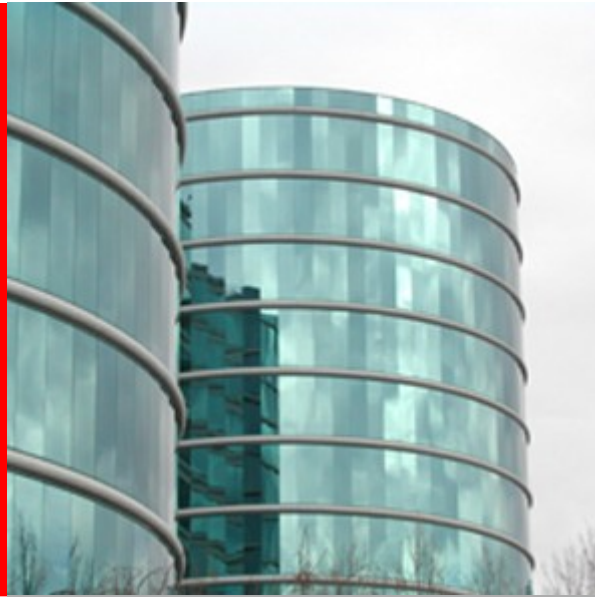
- Nearly all JSRs up and running
- Remaining ones to be filed in the coming weeks
- Final release by Q3 2012
- Date-driven release: anything not ready will be deferred

Futures Roadmap: Java EE 8

- Modularity based on Java SE 8
- More cloud-related APIs
 - NoSQL ?
 - Management ?
- Cloud profile ?
- Explicit tenant API in support of SaaS

Links

- Java EE 7 java.net project
 - Archives, documents, mailing lists,...
 - <http://java.net/projects/javaee-spec>
- Component projects
 - <http://java.net/projects/XXX-spec>
(where XXX = jpa, ejb, jms, servlet, jax-rs, jsf,...)
- Feedback
 - users@javaee-spec.java.net



ORACLE®

Java EE: Wohin die Reise geht

Peter Doschkinow
Senior Java Architect

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technologie und Architektur