



**MOVING JAVA  
FORWARD**

**ORACLE®**

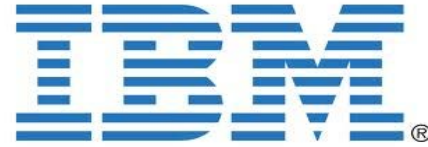
## **The Java EE 7 Platform: Developing for the Cloud**

Simon Ritter

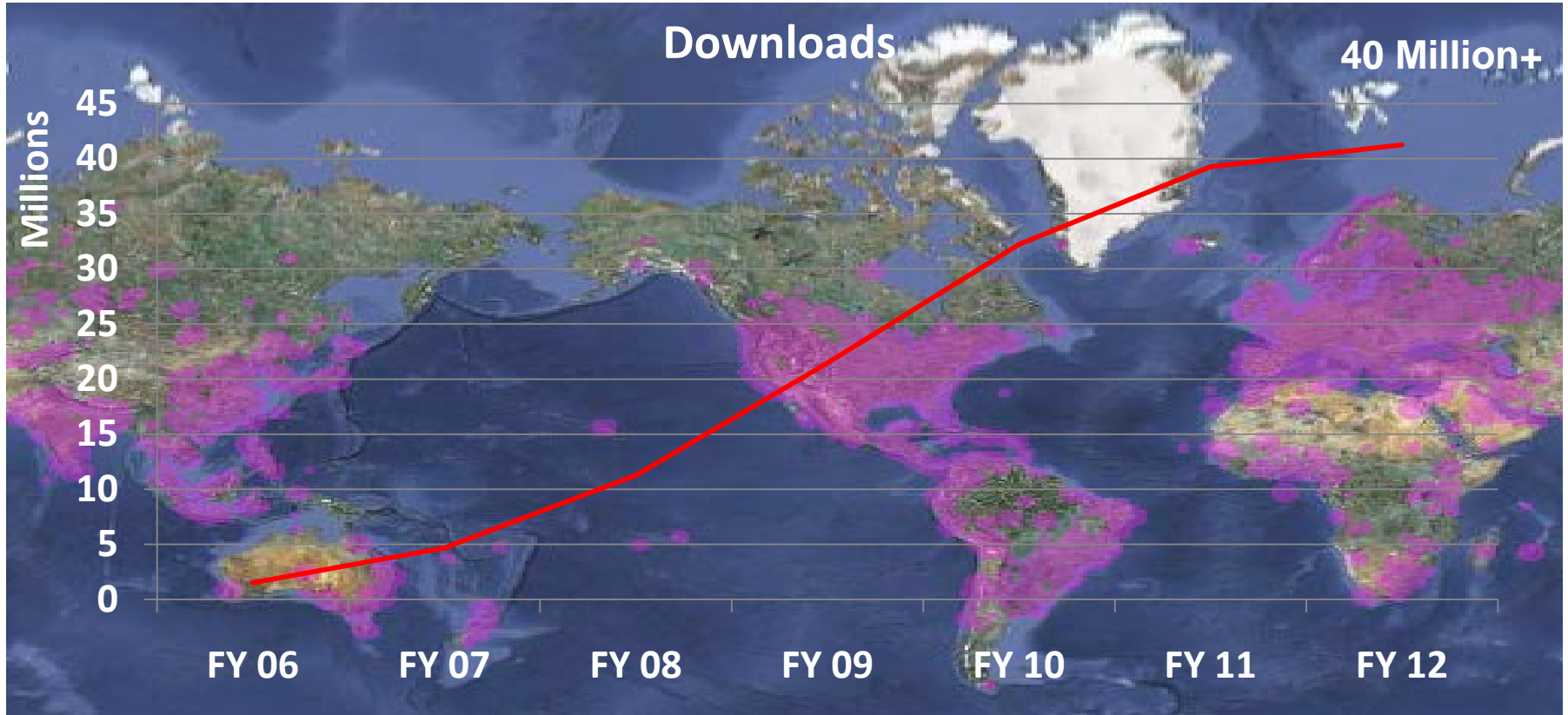
Technology Evangelist, Oracle Corporation

# Java EE 6 Implementations

Fastest implementations of a Java EE release ever!



# Java EE – Developer Adoption



We're moving Java EE  
into the Cloud !

# Today's Cloud Offerings Are Vendor-Specific

Infrastructure as  
a Service

Platform as  
a Service

Software as  
a Service



...

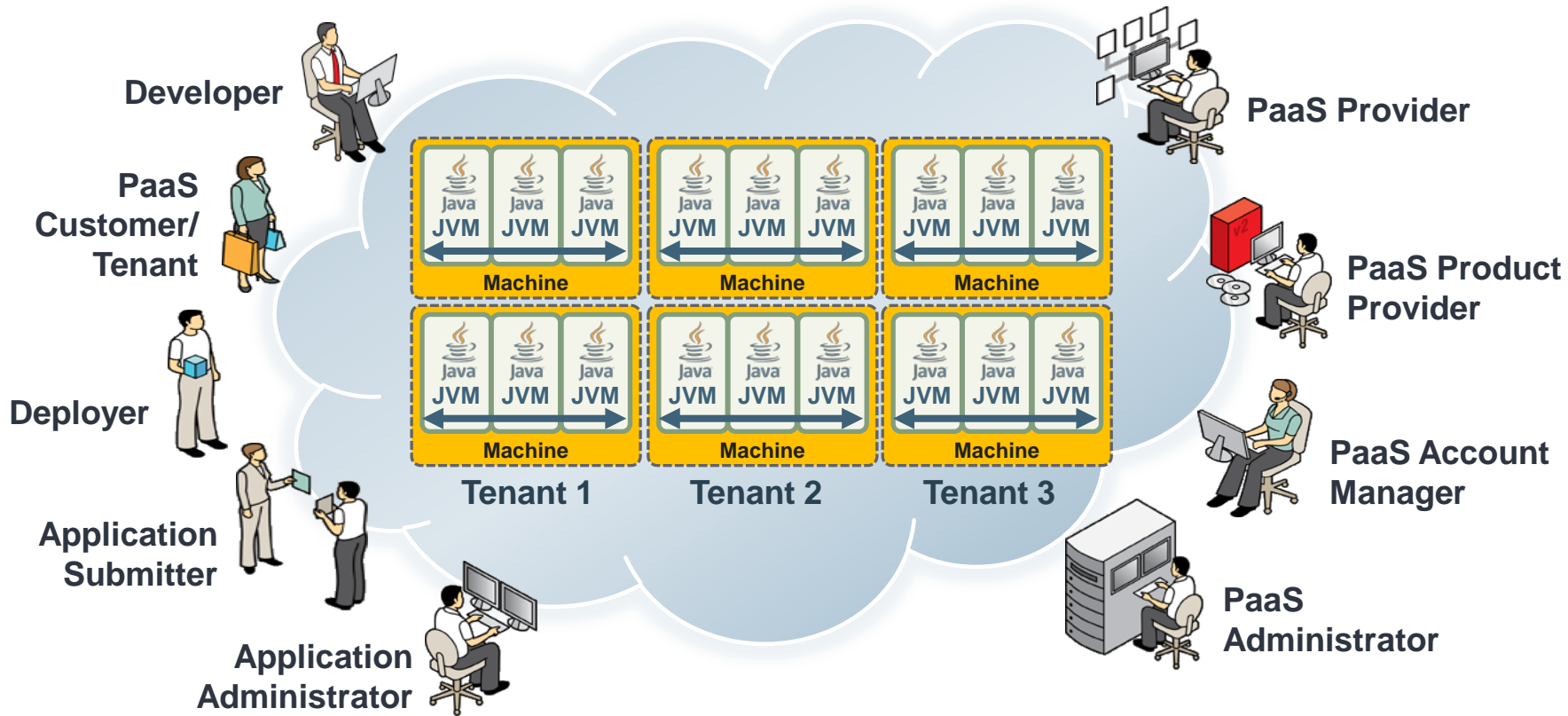
# Java EE 7 Focus: Platform as a Service

- 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

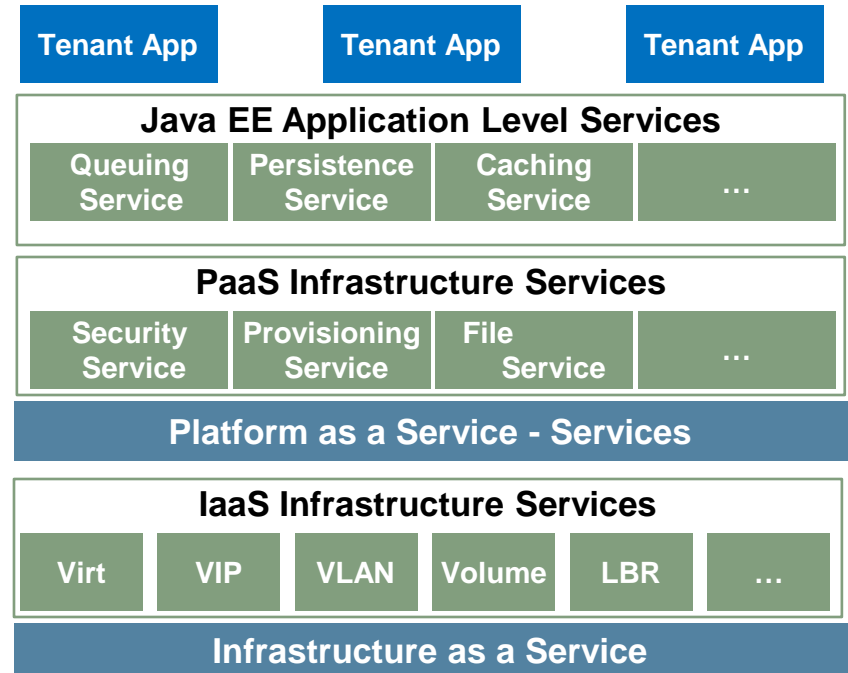
- Define new platform roles to accommodate PaaS model
- Add metadata
  - For service provisioning and configuration
  - For QoS, elasticity
  - For sharing of applications and resources
  - For (re)configurability and customization
- Add useful APIs for cloud environment
  - JAX-RS client API, Caching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

# Roles



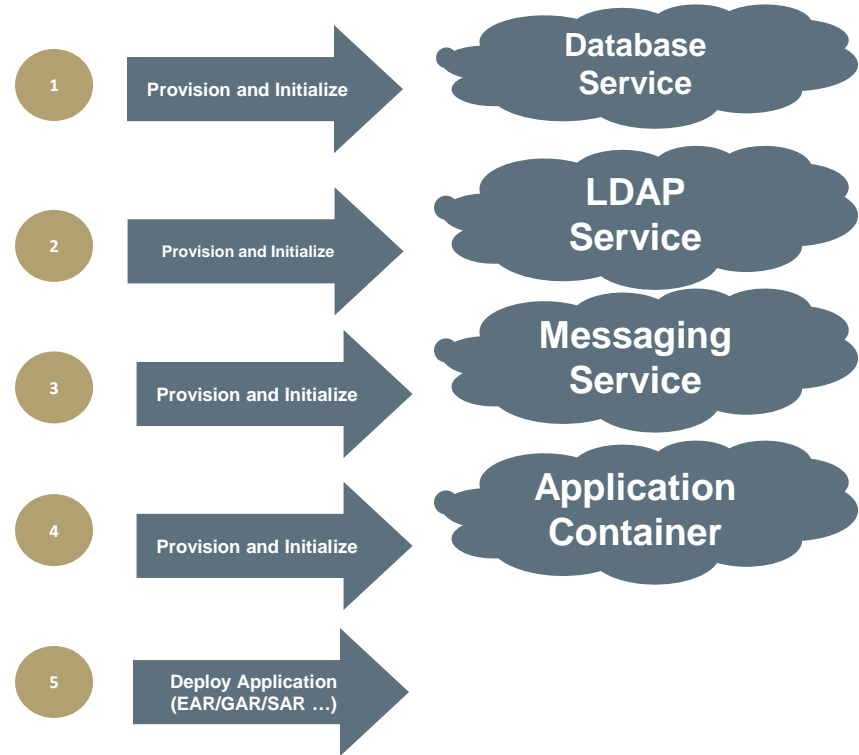
# Java EE 7 Focus Area: Cloud Services, Not Just APIs

- 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



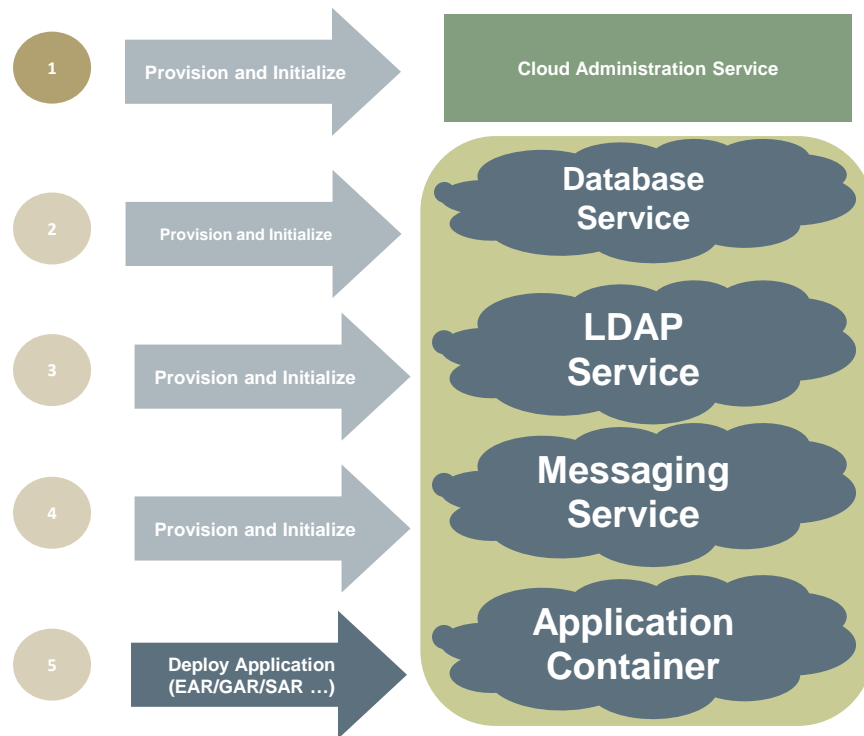
# Old Java EE model

- 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 ...



# Java EE 7 Focus Area: Elasticity

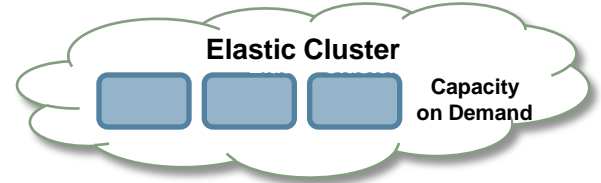
Java EE 7 Focus:  
Move the Bar Right



Single node  
Non-Elastic



Java EE Multi-Node  
Multi-Instance Clustering



Dynamic Self Adjusting  
SLA Driven Elasticity

- Service Levels
- Minimum and Maximum Instances
- Futures – Self Adjustment, Capacity On Demand

# Example Scenario

“A software company writes an application, submits it to a PaaS provider, then any number of tenants sign up for the application, deploy it, their end users access it”

# Walkthrough (1)

SimplyCRM

DiabloCloud

# Walkthrough (2)

SimplyCRM

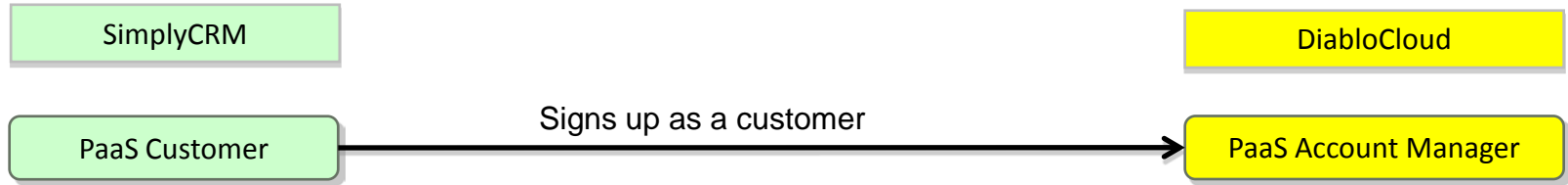
DiabloCloud

Application Developer

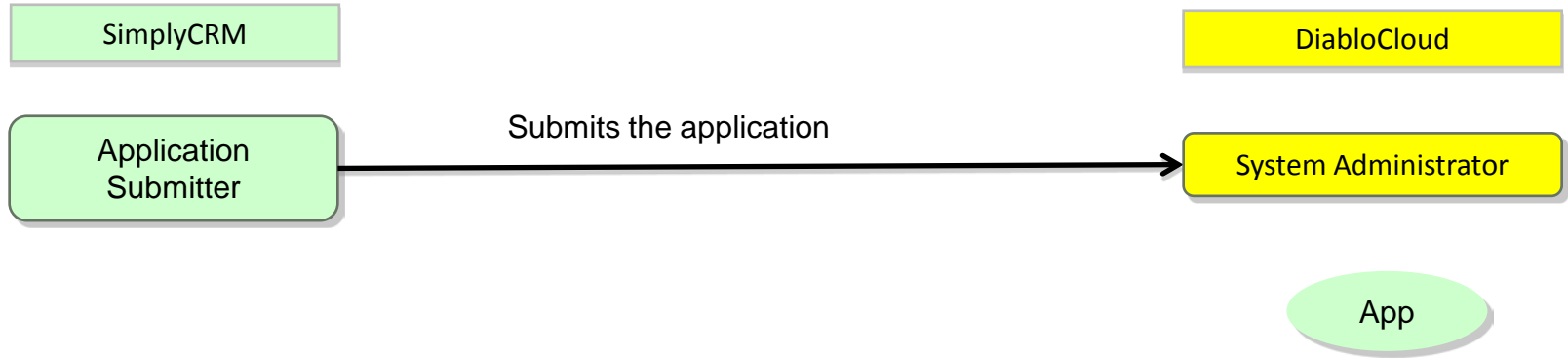
Writes

App

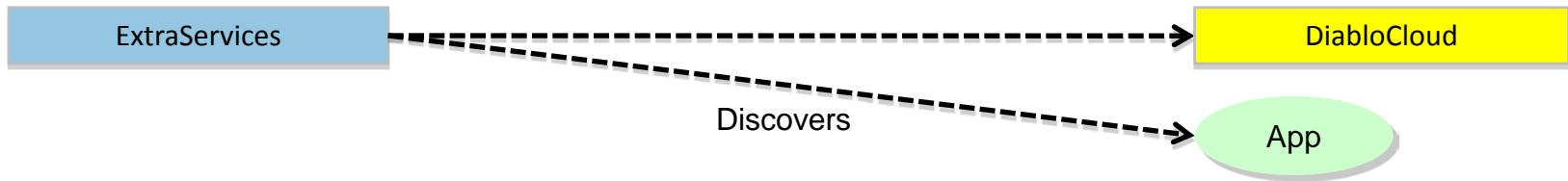
# Walkthrough (3)



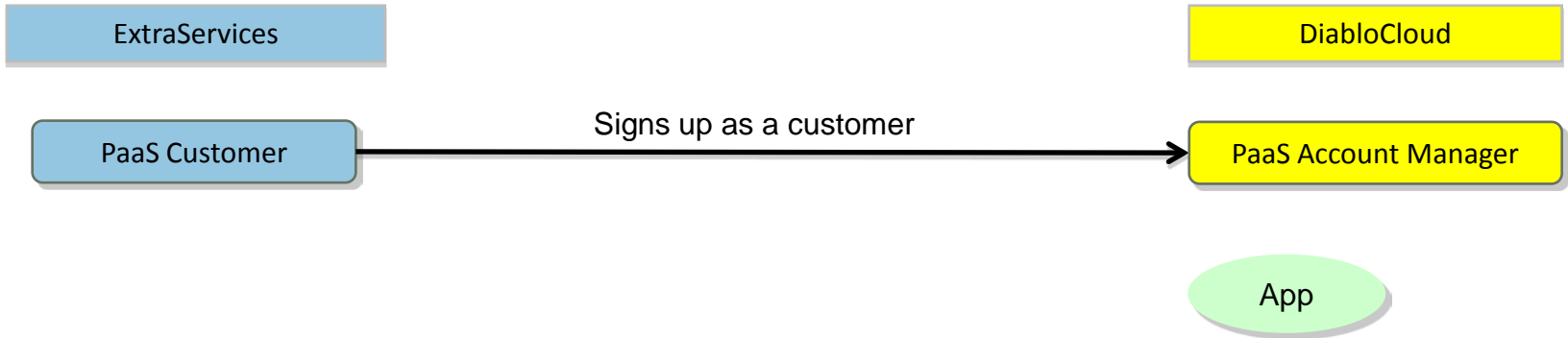
# Walkthrough (4)



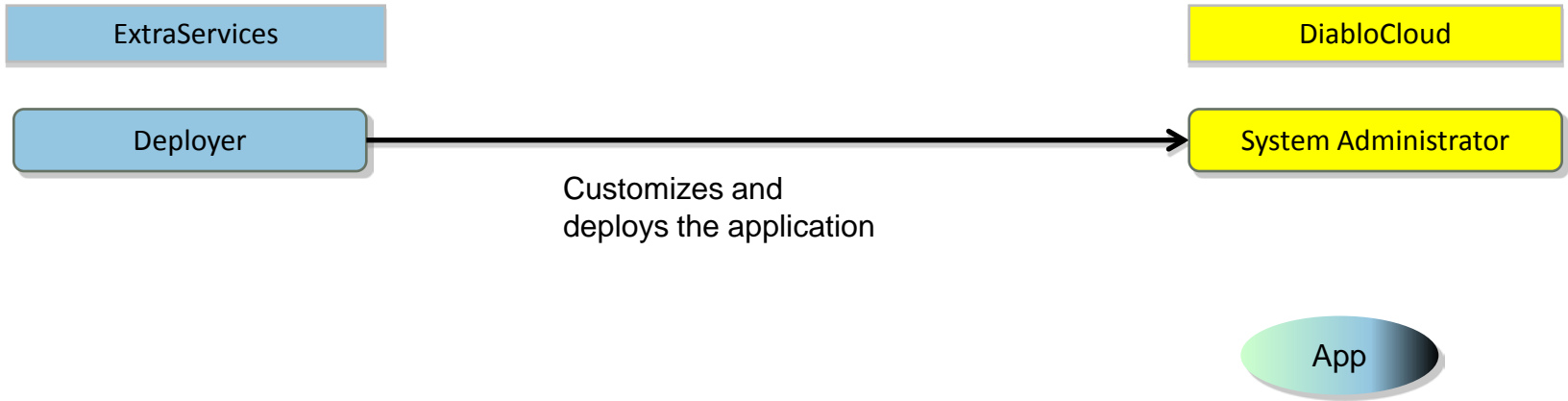
# Walkthrough (5)



# Walkthrough (6)



# Walkthrough (7)



# Walkthrough (8)

ExtraServices

Deployer

DiabloCloud

System Administrator

Provisions on  
Cloud  
infrastructure

App

# Walkthrough (9)

ExtraServices

Deployer

DiabloCloud

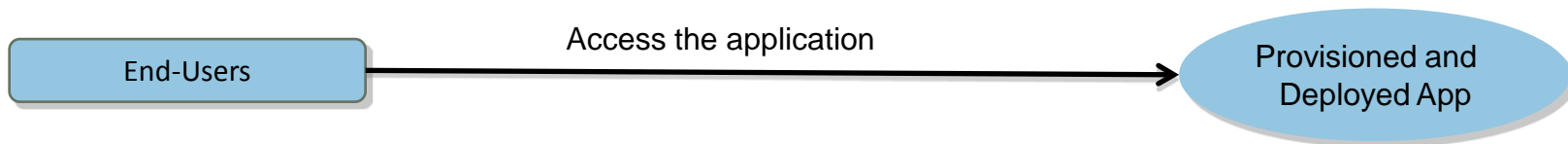
System Administrator

Provisioned and  
Deployed App

# Walkthrough (10)

ExtraServices

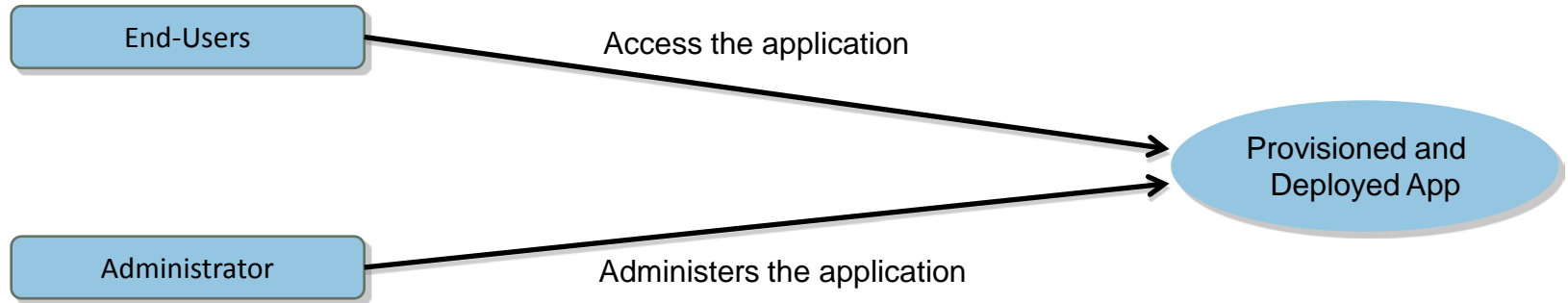
DiabloCloud



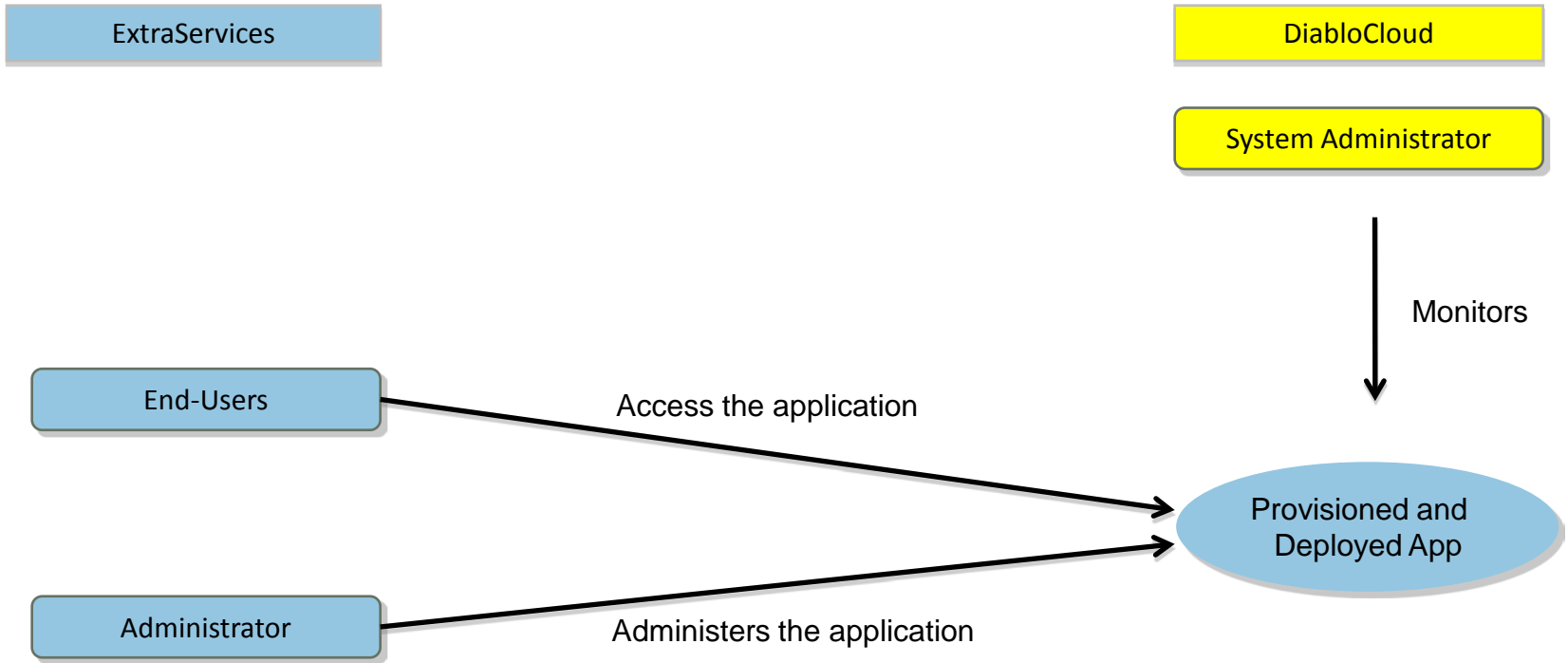
# Walkthrough (11)

ExtraServices

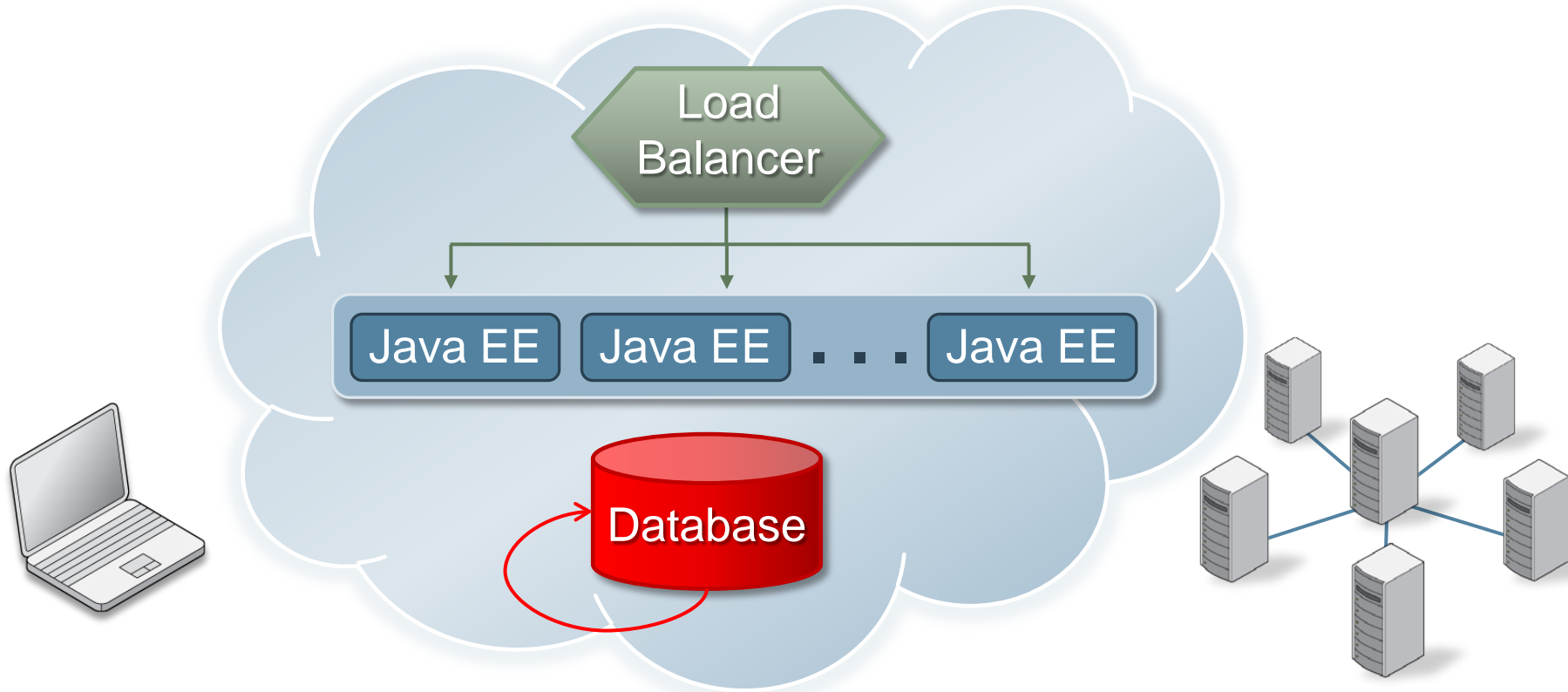
DiabloCloud



# Walkthrough (12)



# Service Provisioning

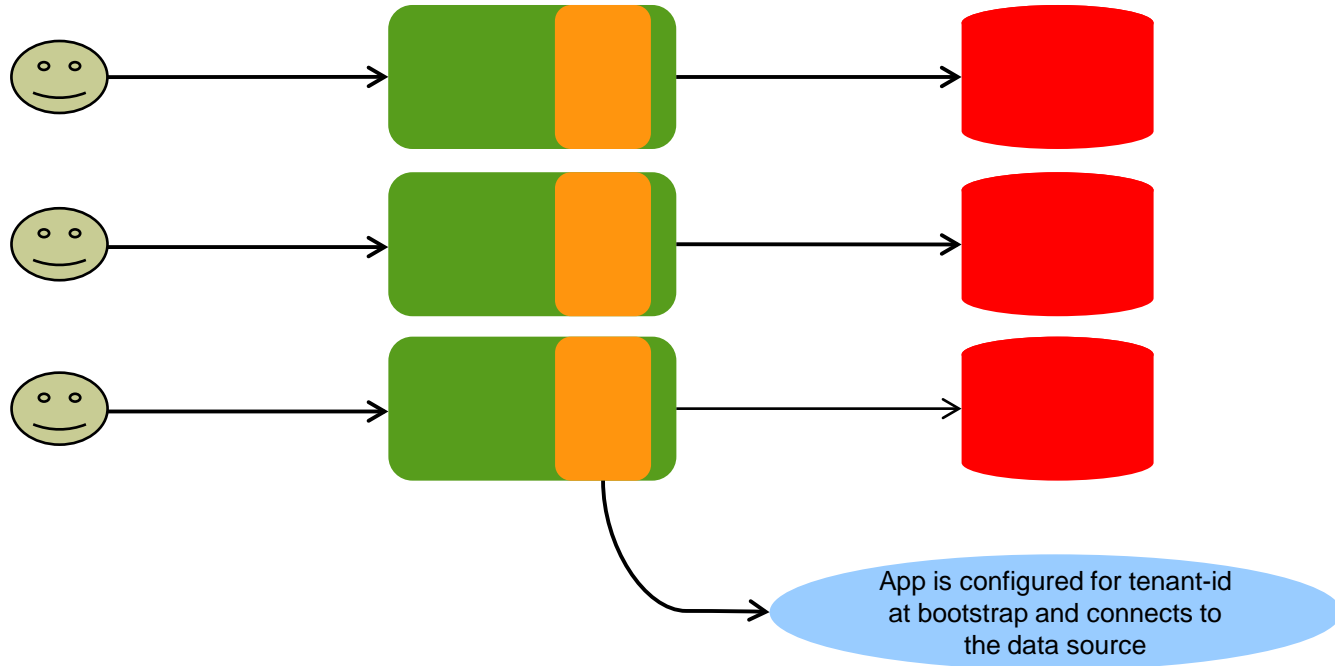


# Multi-tenancy in Java EE 7

- 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

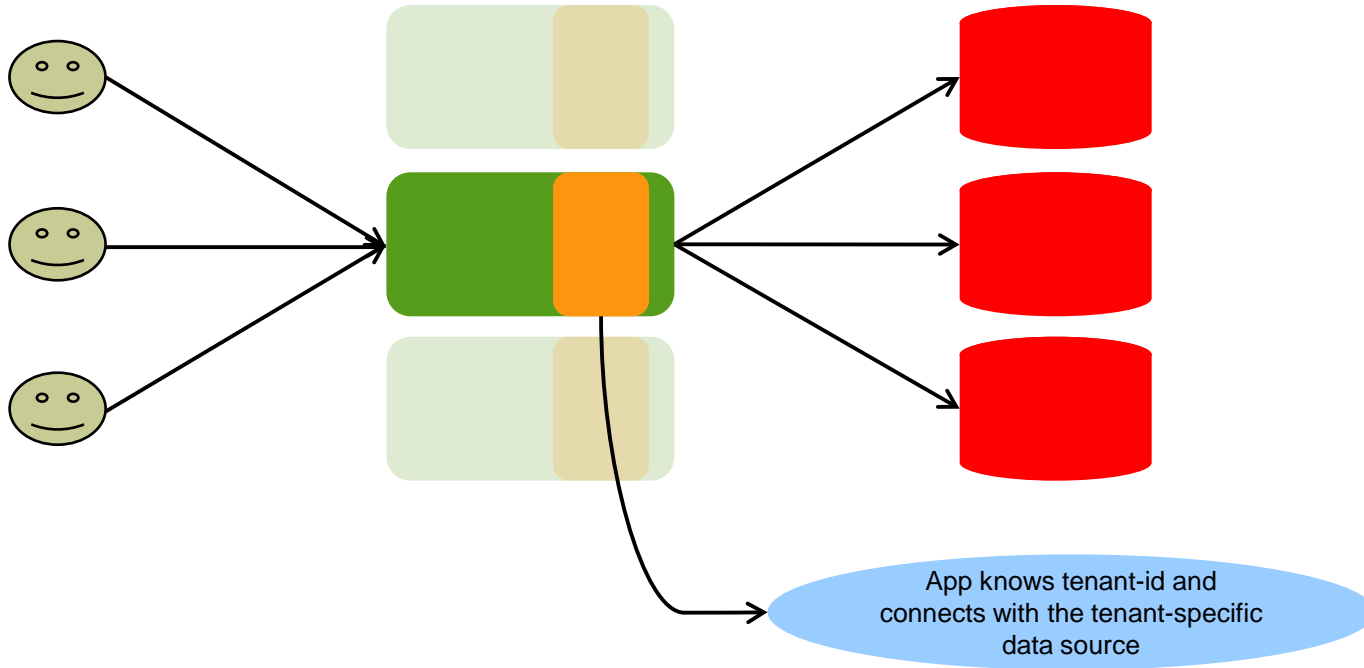
# Dedicated App, Dedicated Database

Persistence Layer Multi-Tenant Taxonomies



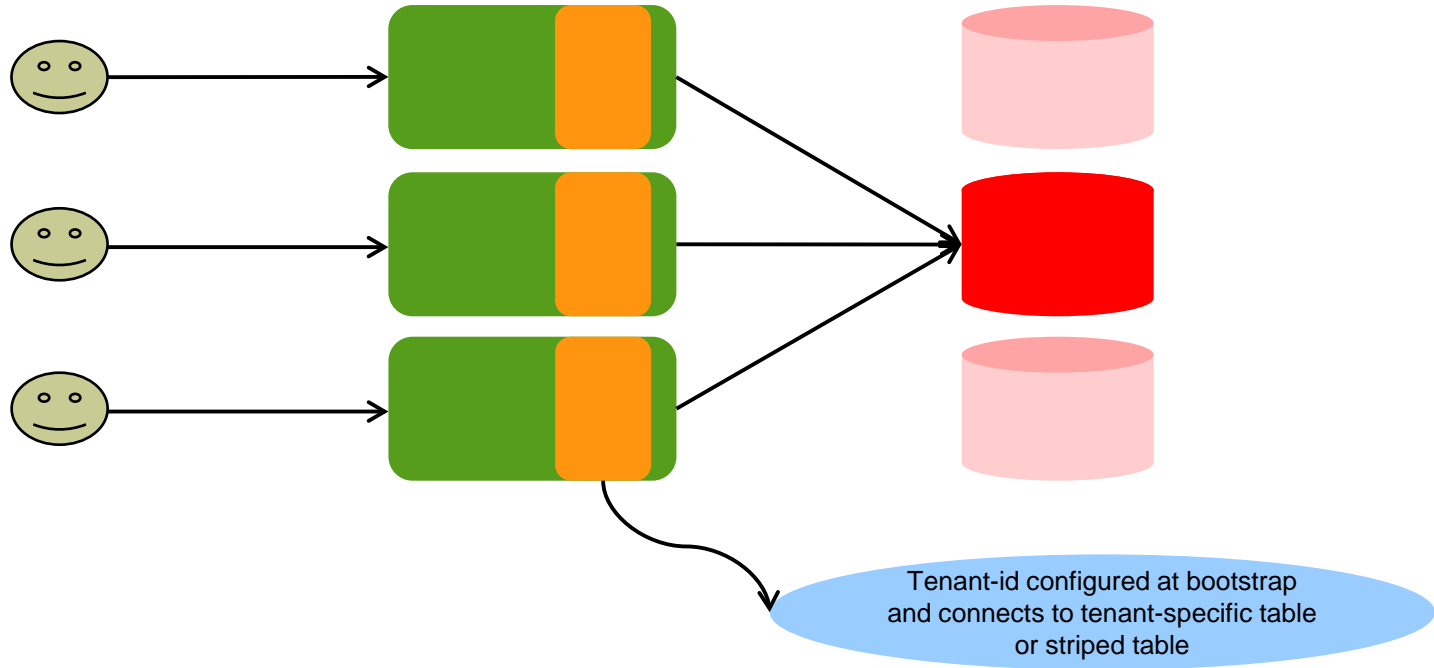
# Shared App, Dedicated Database

Persistence Layer Multi-Tenant Taxonomies



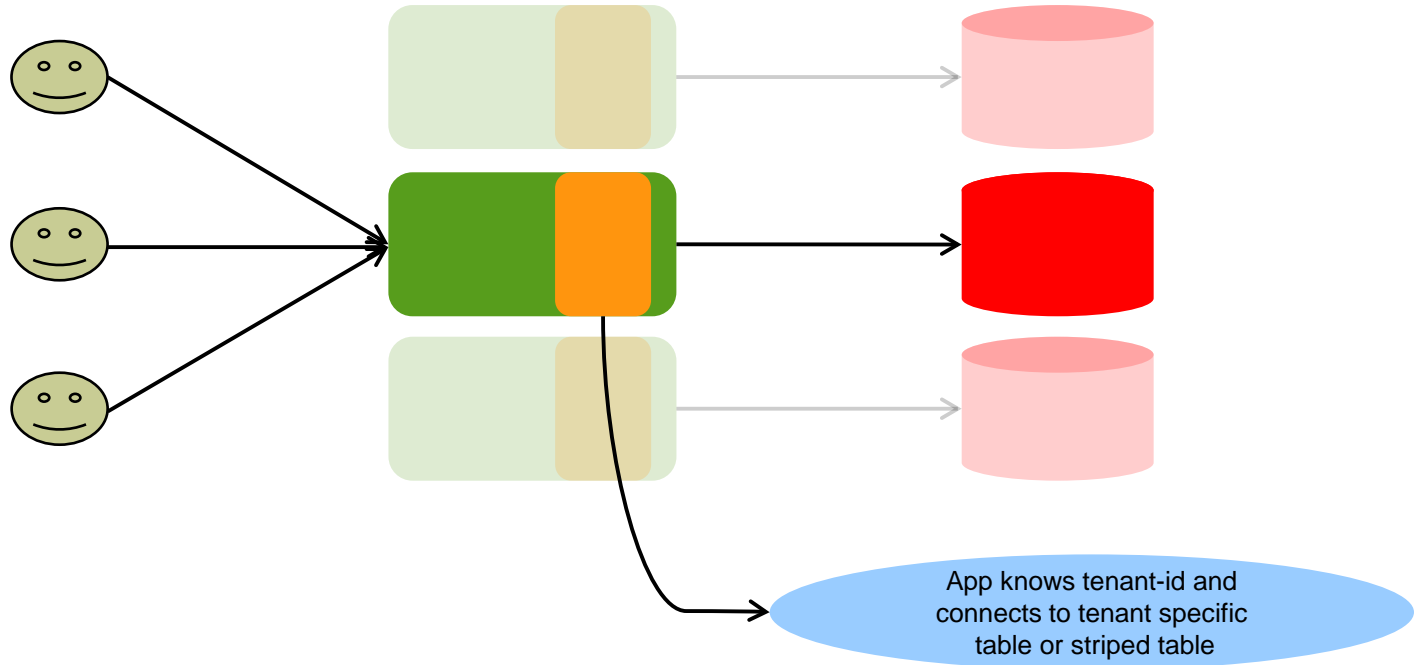
# Dedicated App, Shared Database

Persistence Layer Multi-Tenant Taxonomies

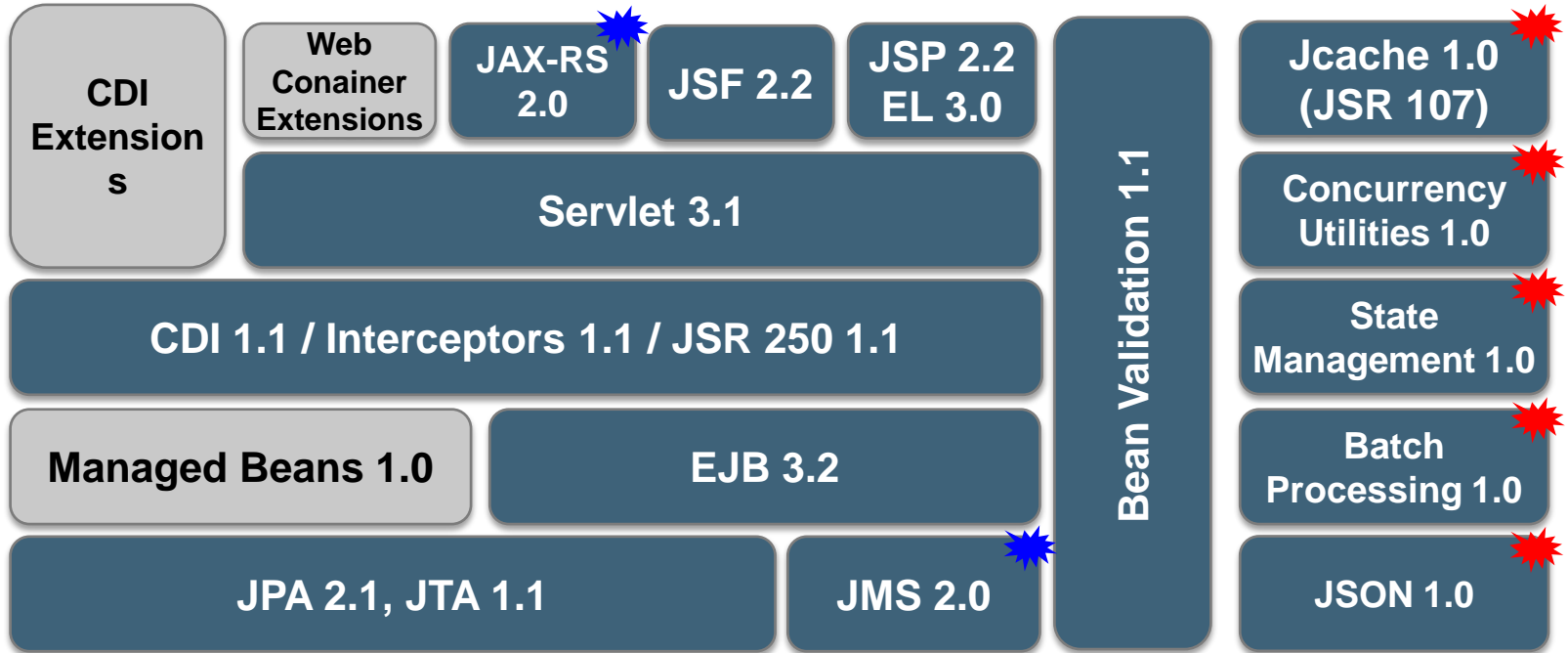


# Shared App, Shared Database

Persistence Layer Multi-Tenant Taxonomies



# Java EE 7 – Candidate JSRs



# 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>

The 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.



**MOVING JAVA  
FORWARD**

**ORACLE®**

## **The Java EE 7 Platform: Developing for the Cloud**

Simon Ritter

Technology Evangelist, Oracle Corporation