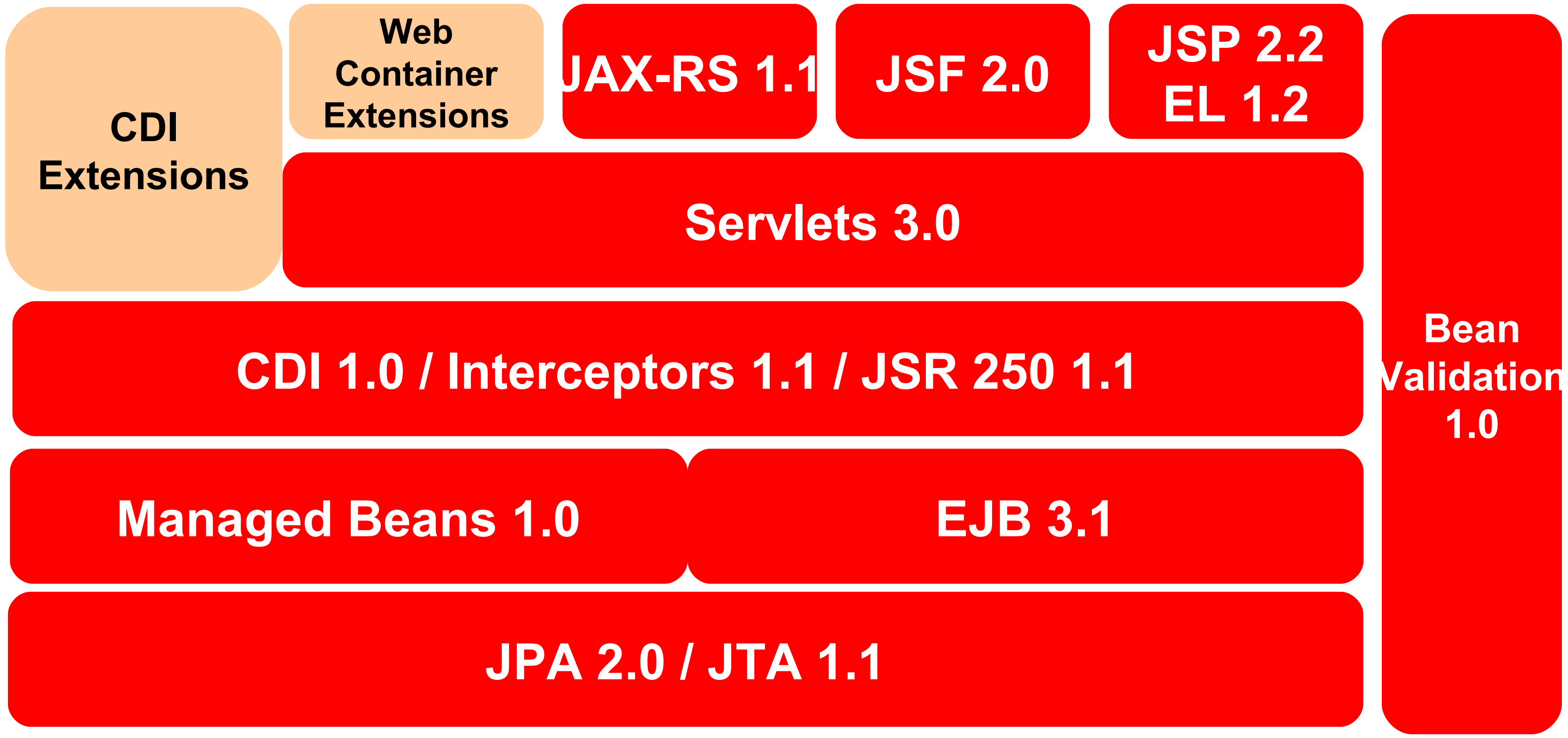
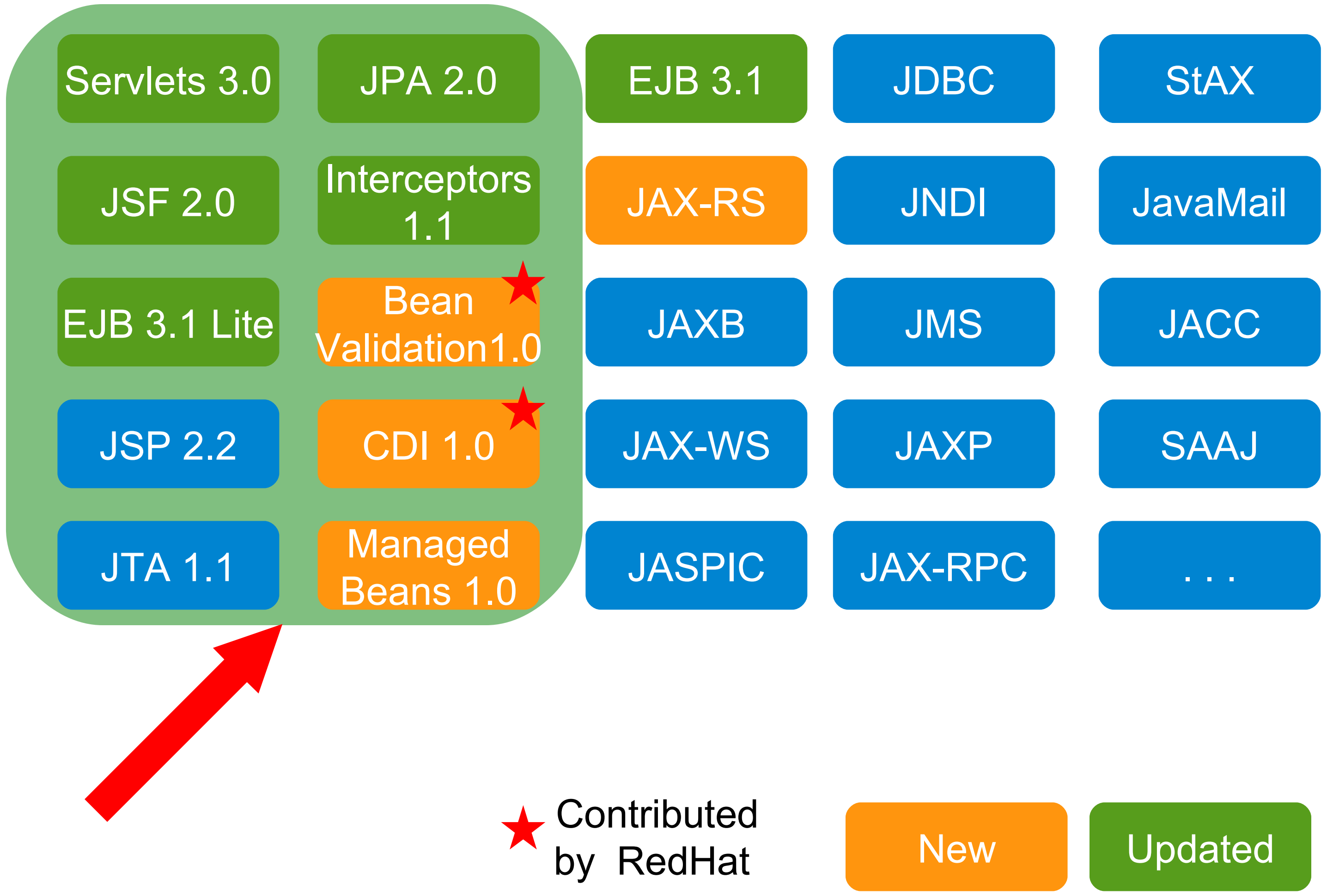
The Java EE 7 Platform: Developing for the Cloud

Arun Gupta, Java EE & GlassFish Guy
blogs.oracle.com/arungupta, @arungupta

The Core Java EE 6 Programming Model



The Java EE 6 Web Profile 1.0



Compatible Java EE 6 Impls

Today:

Tmax Soft

JBoss a division of Red Hat

Web Profile Only

caucho

WebSphere.

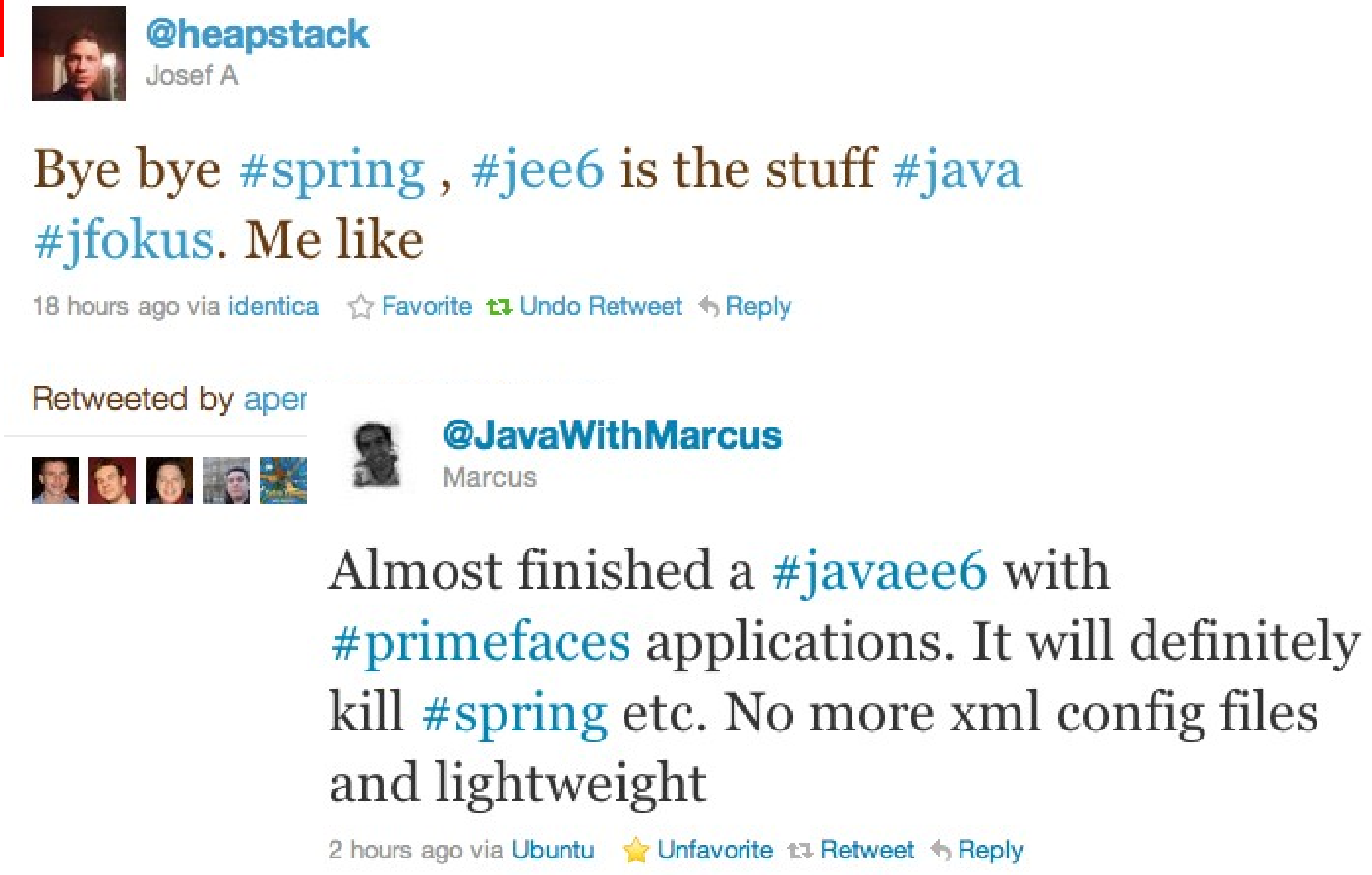
Java COMPATIBLE ENTERPRISE EDITION

Announced:

Oracle WebLogic Suite 11*g*

APACHE GERONIMO

Siwpas



@heapstack
Josef A

Bye bye #spring , #jee6 is the stuff #java #jfokus. Me like

18 hours ago via identica ☆ Favorite Undo Retweet Reply

Retweeted by aper

@JavaWithMarcus
Marcus

Almost finished a #javaee6 with #primefaces applications. It will definitely kill #spring etc. No more xml config files and lightweight

2 hours ago via Ubuntu ★ Unfavorite Retweet Reply

9 Reasons why Java EE 6 will save $$

- 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)
- Plan B (Similar component models)

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

From the real users ...

Jigsaw puzzle, Modular, standard, **less xml, easy, easy**, have I said easy?

Developers can concentrate on business logic, Java EE 6 is providing a **standard for the infrastructure**.

Standards compliance, vendor independence, **milliseconds and kilobyte deployment**

Faster development, **less frameworks, less complexity**, more great code shipped

Higher integrated specs, simple and annotation driven, **single-classloader WARs**, next level of industry standard

Definite excuse to **avoid Spring forever**

Simplified Java Development, **Focus on building great products**

Not your fat grandfather's enterprise Java anymore, **enterprise Java renaissance**

http://blogs.oracle.com/arungupta/tags/community+feedback

Avoid “framework explosion”

In selecting an application server our main goal was to avoid the framework explosion that happens when you use a **"custom" Enterprise stack like Tomcat + Spring + Hibernate + Myfaces** +... Java EE 6 had 80% of what we needed out of the box: strong persistence support (JPA), inversion of control (CDI), and a lightweight component model (EJB 3.1)

http://blogs.oracle.com/stories/entry/egesa_engineering_avoids_framework_explosion

What does Java EE offer to Cloud ?

- Containers
- Injectable services
- Scale to large clusters
- Security model
- . . .

Java EE for the Cloud : JSR 342

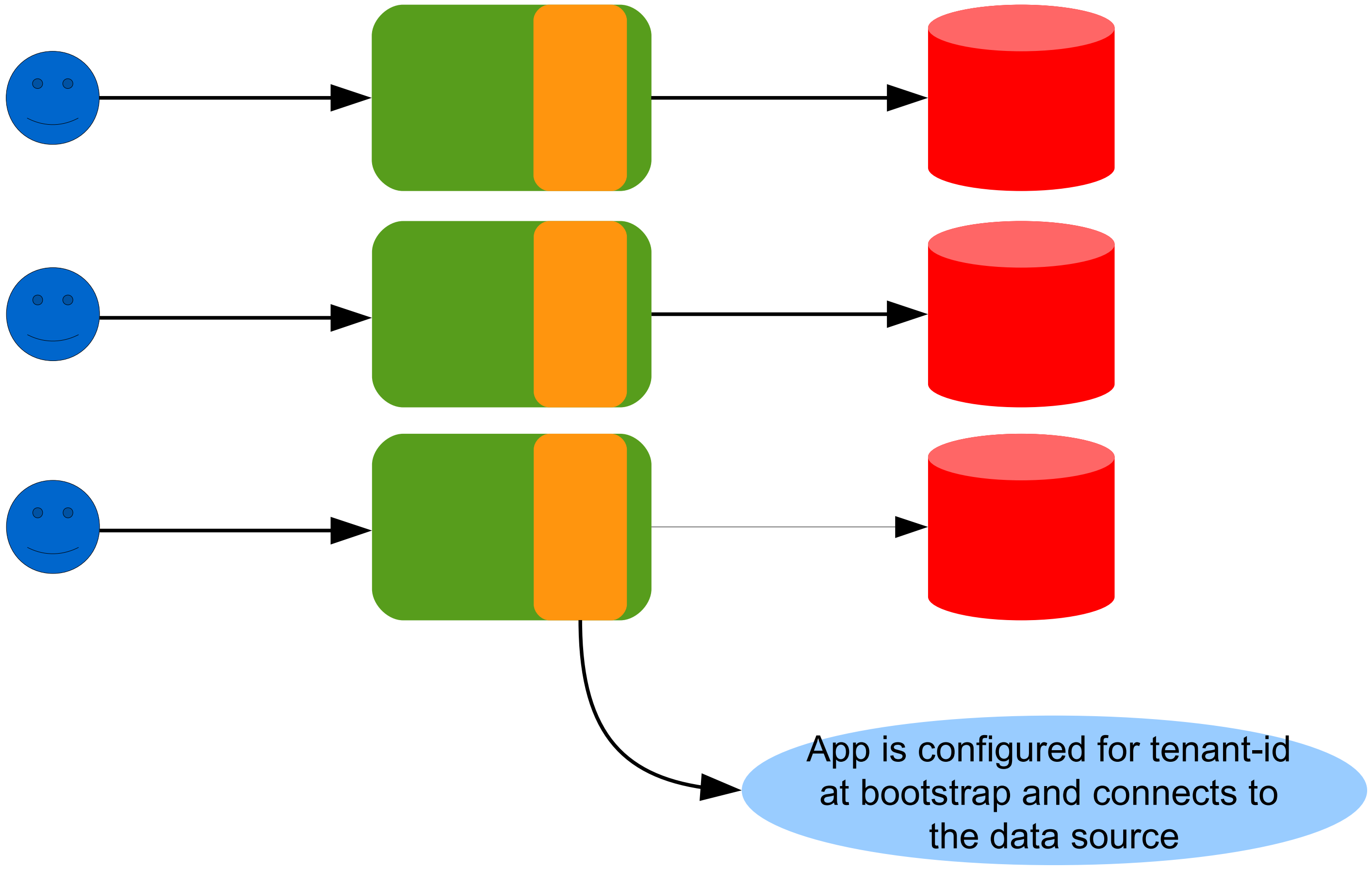
- More easily operate on private/public clouds
  - Multi-tenancy
  - Elasticity
- Tighter requirements for resource and state management
- Better isolation between applications
- Potential standard APIs for NRDBMS, Caching, other
- Common management and monitoring interfaces
- Better packaging
- Evolution, not revolution

Persistence Layer
Multi-Tenant Taxonomies

- Dedicated App, Dedicated Database
- Shared App, Dedicated Database
- Dedicated App, Shared Database
- Shared App, Shared Database

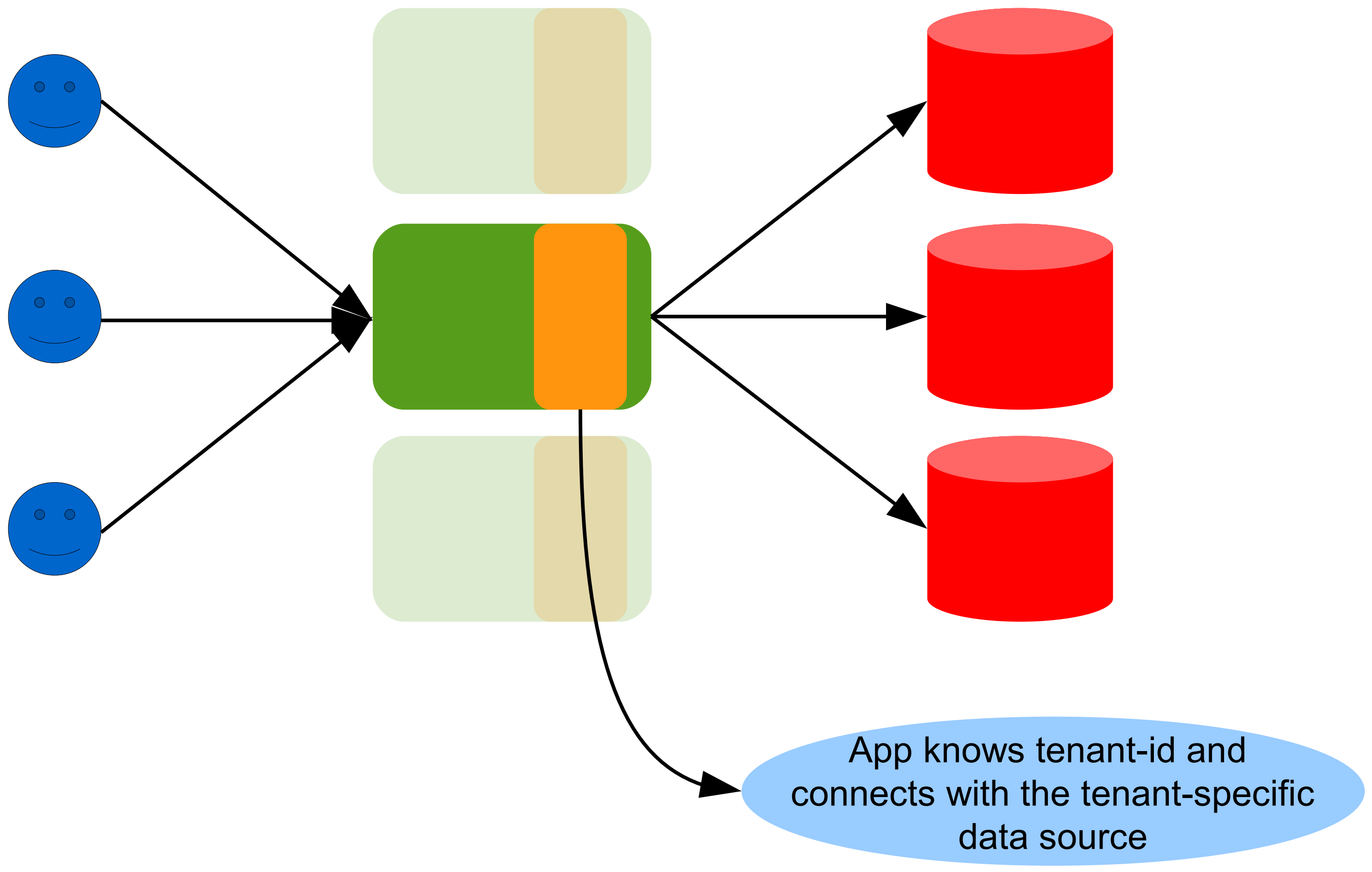
Dedicated App, Dedicated Database

Persistence Layer Multi-Tenant Taxonomies


App is configured for tenant-id at bootstrap and connects to the data source

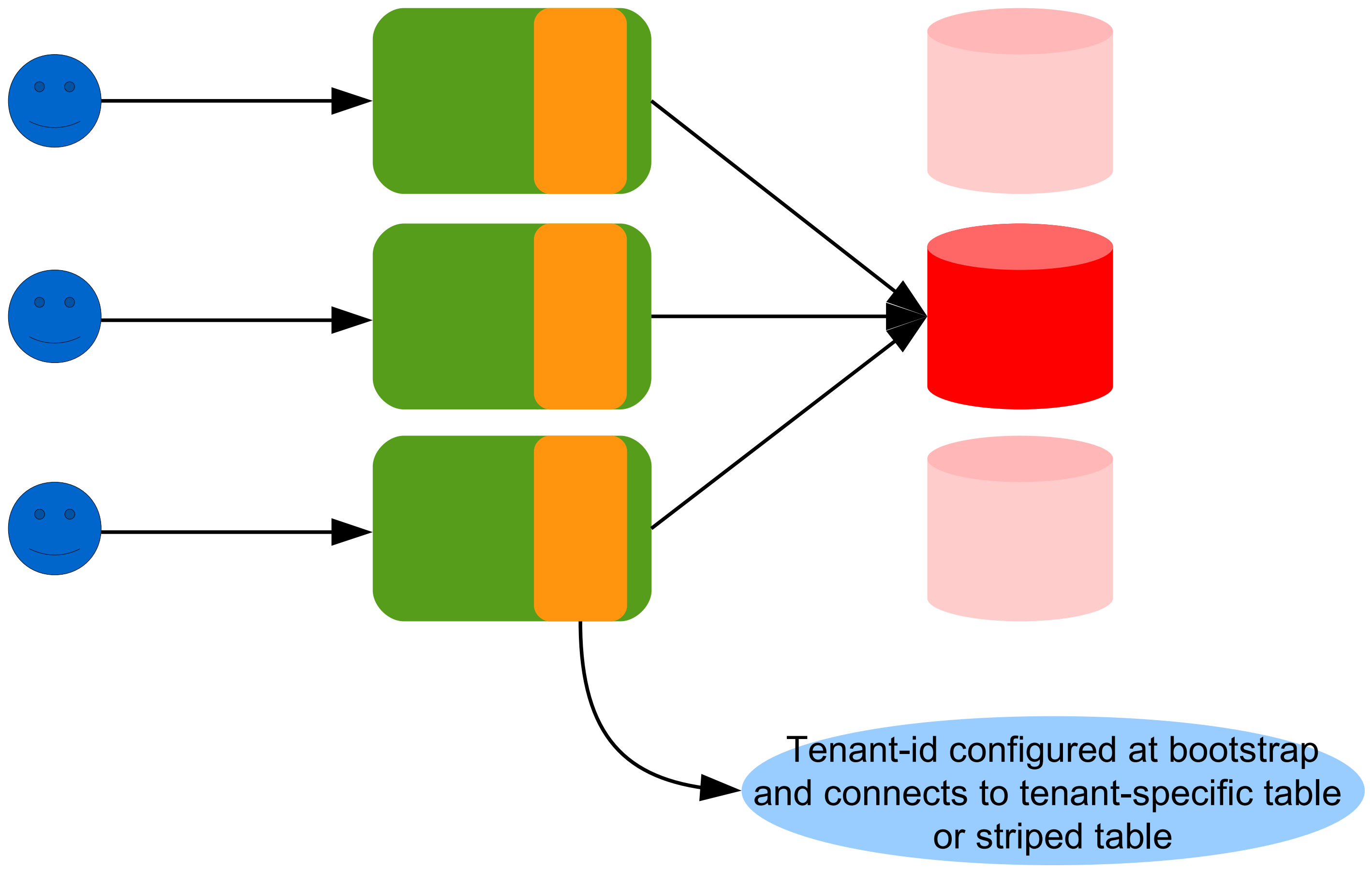
Shared App, Dedicated Database

Persistence Layer Multi-Tenant Taxonomies


App knows tenant-id and connects with the tenant-specific data source

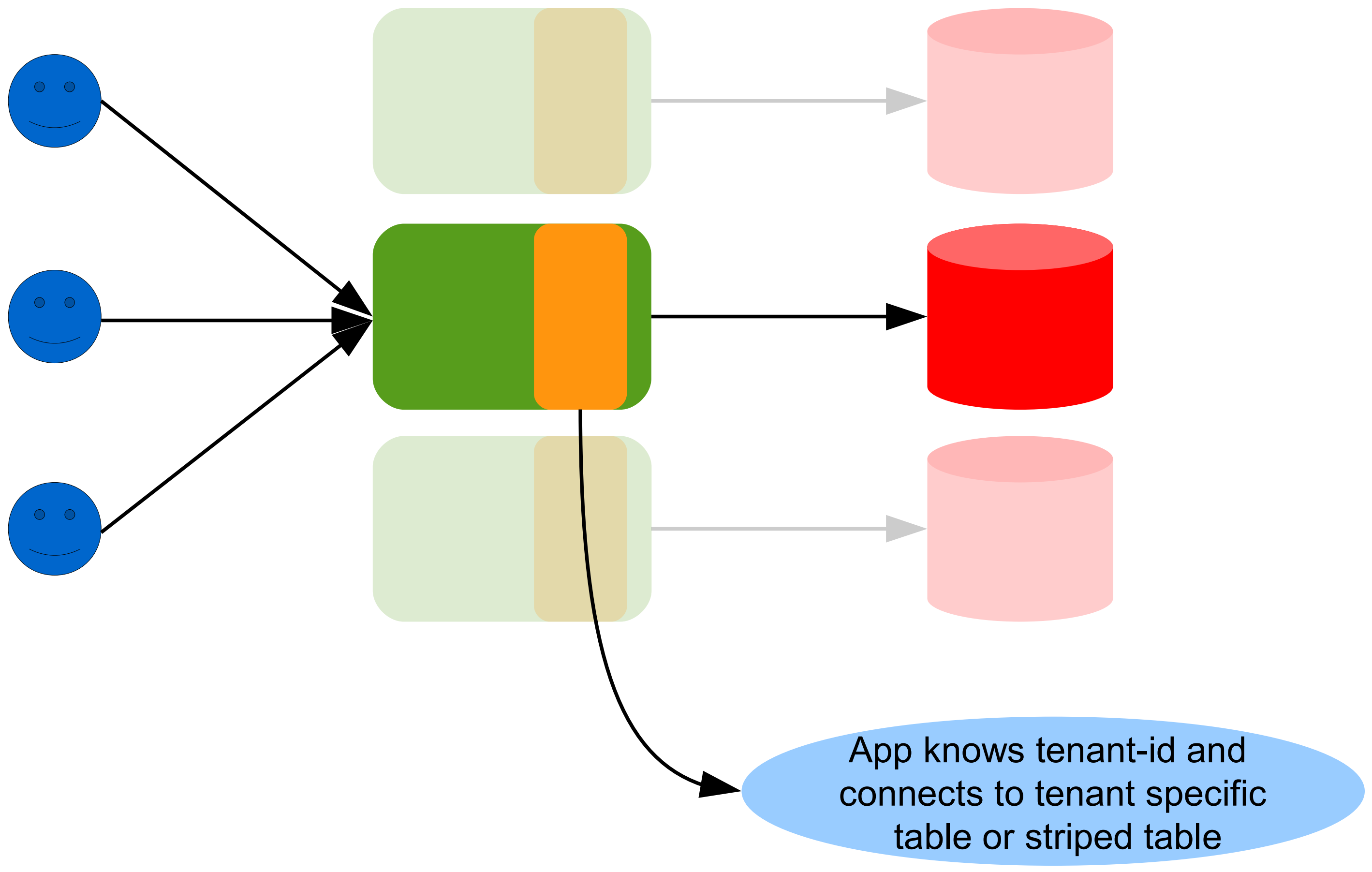
Dedicated App, Shared Database

Persistence Layer Multi-Tenant Taxonomies

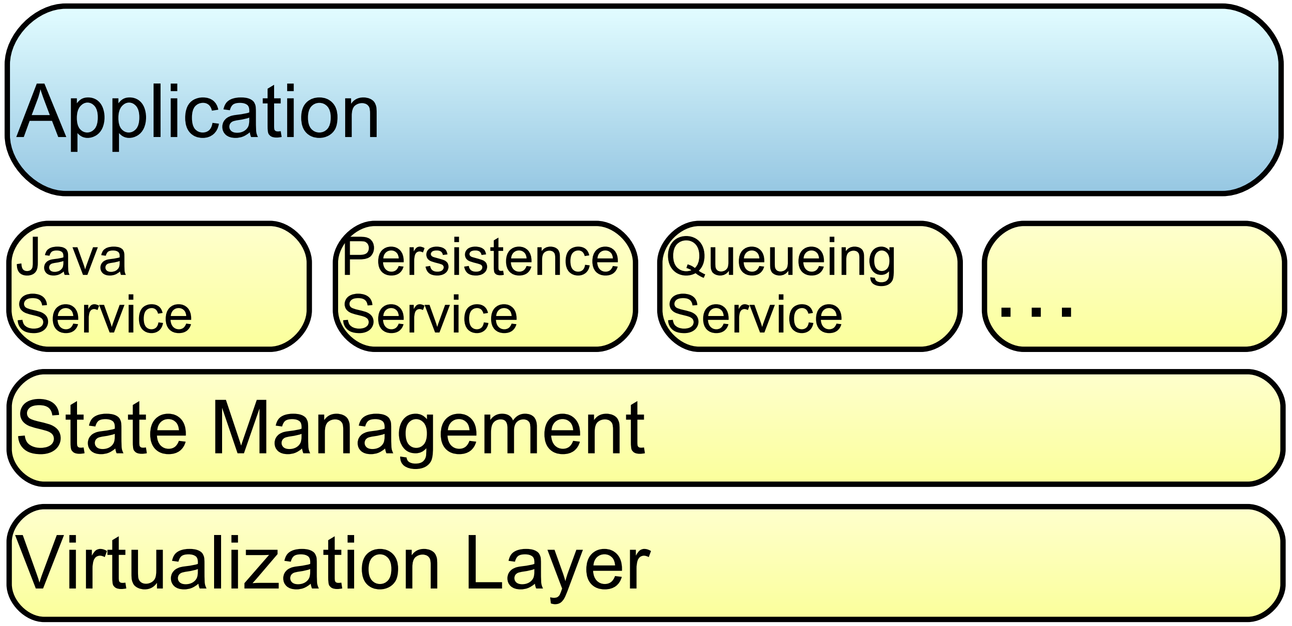

Tenant-id configured at bootstrap and connects to tenant-specific table or striped table

Shared App, Shared Database

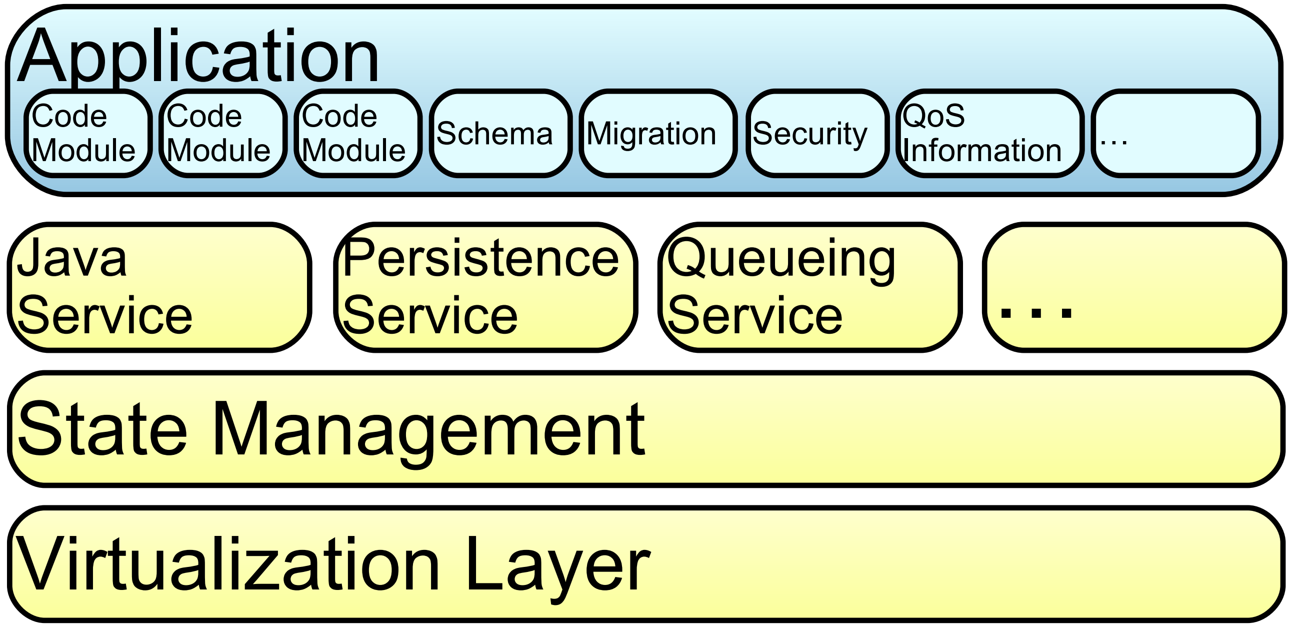
Persistence Layer Multi-Tenant Taxonomies


App knows tenant-id and connects to tenant specific table or striped table

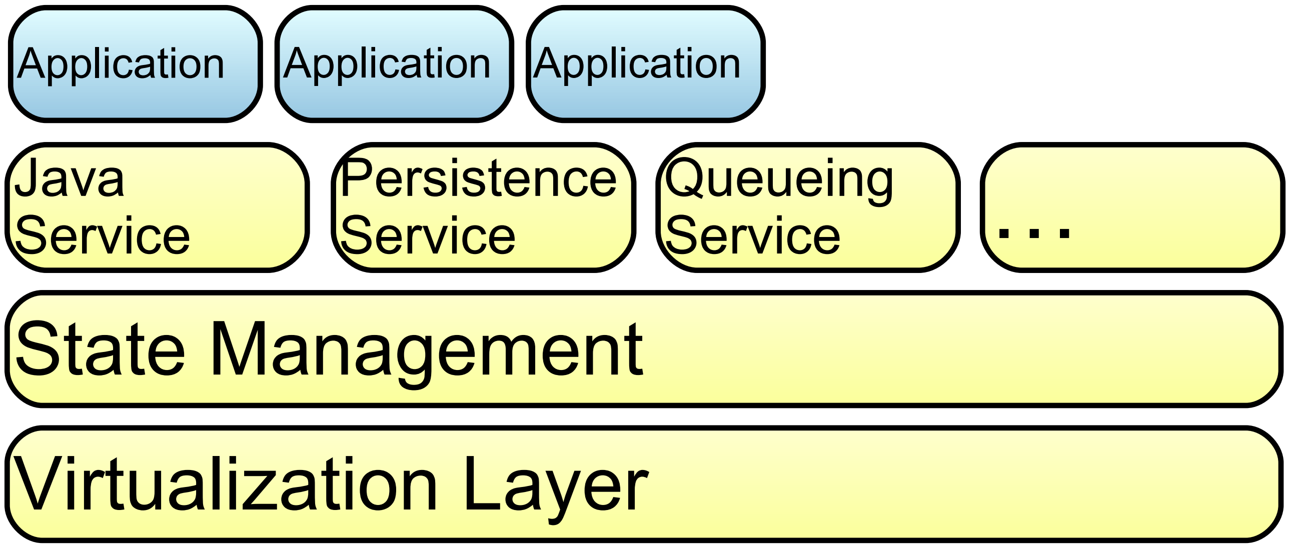
Cloud Platform



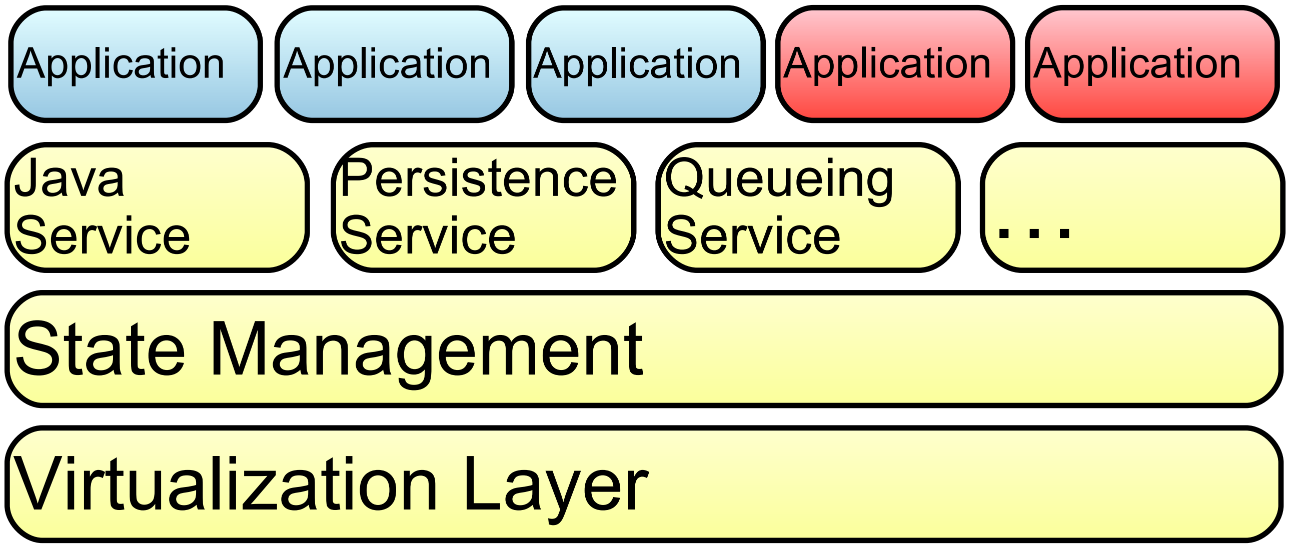
Cloud Platform



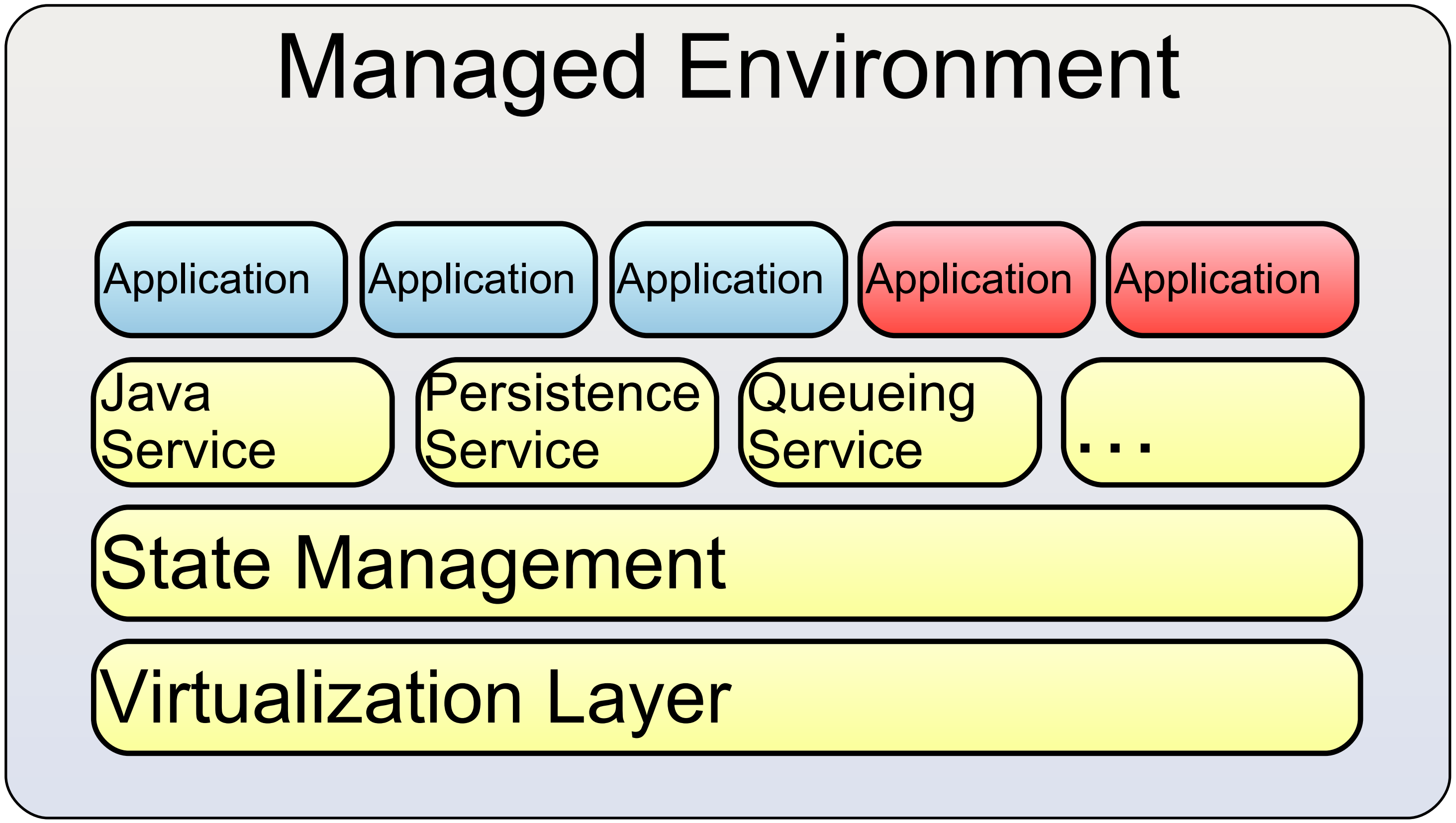
Cloud Platform



Cloud Platform



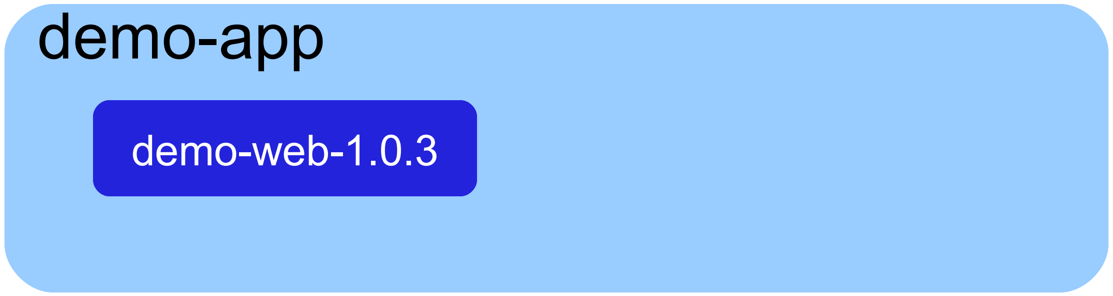
Cloud Platform



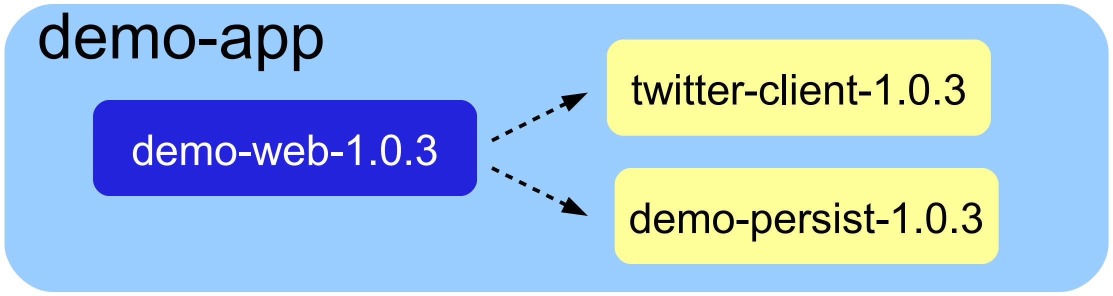
The Java EE 7 Modularity

- Built on Java SE 8 work
- Applications made of modules
- Dependencies are explicit
- Versioning is built-in
- Classloaders are straightened

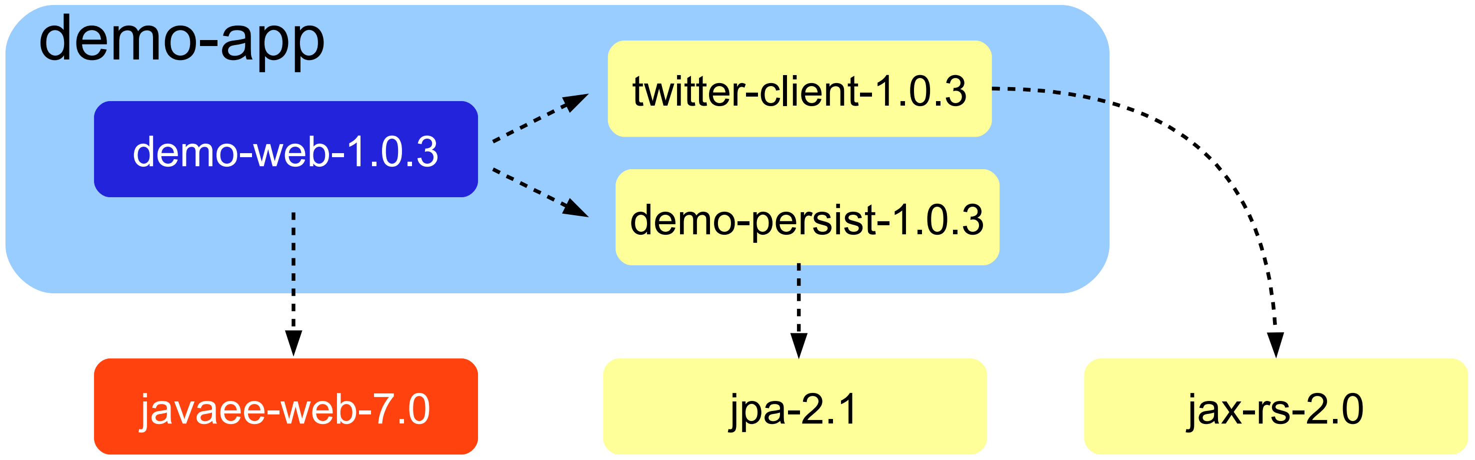
Modular Applications



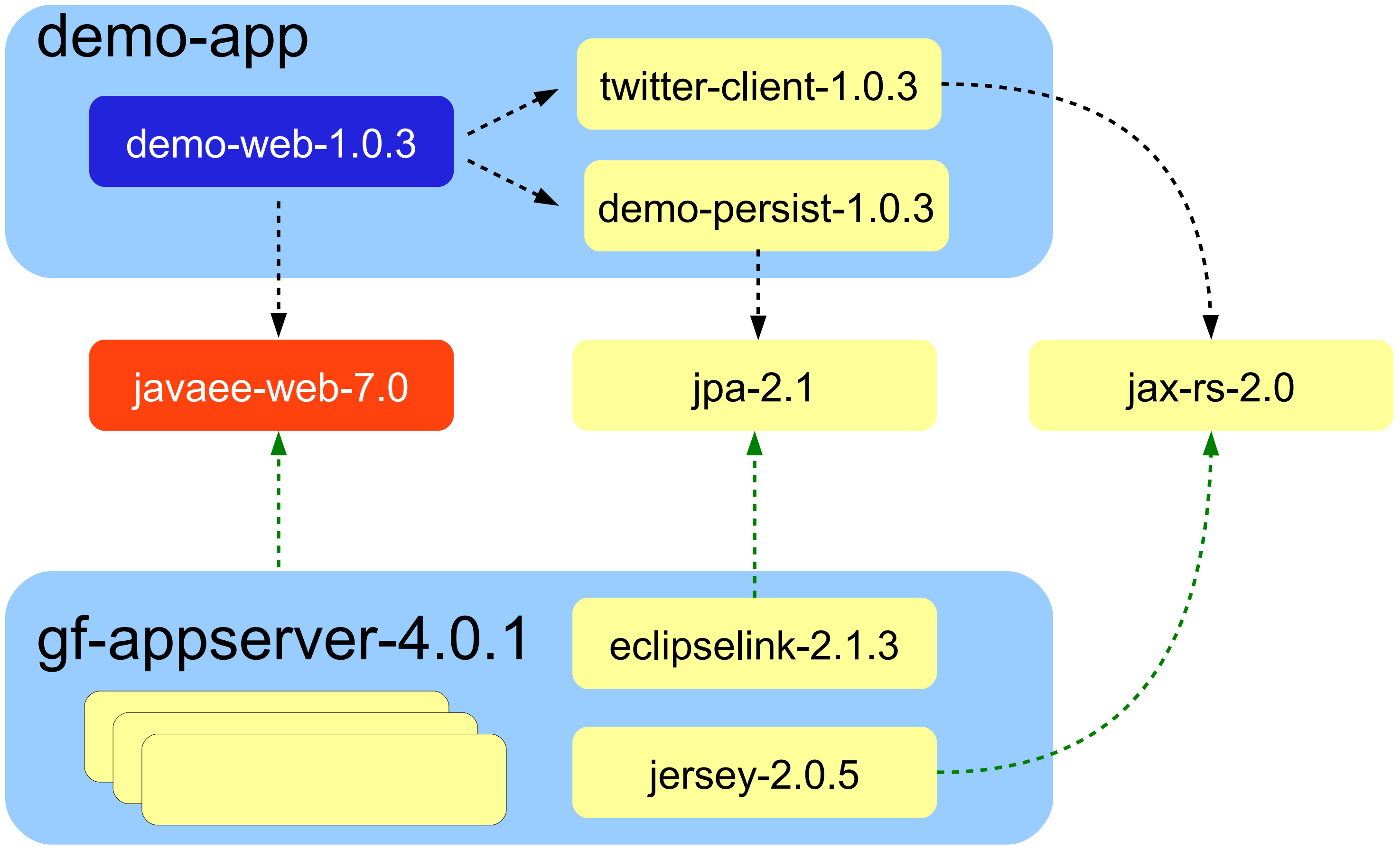
Modular Applications



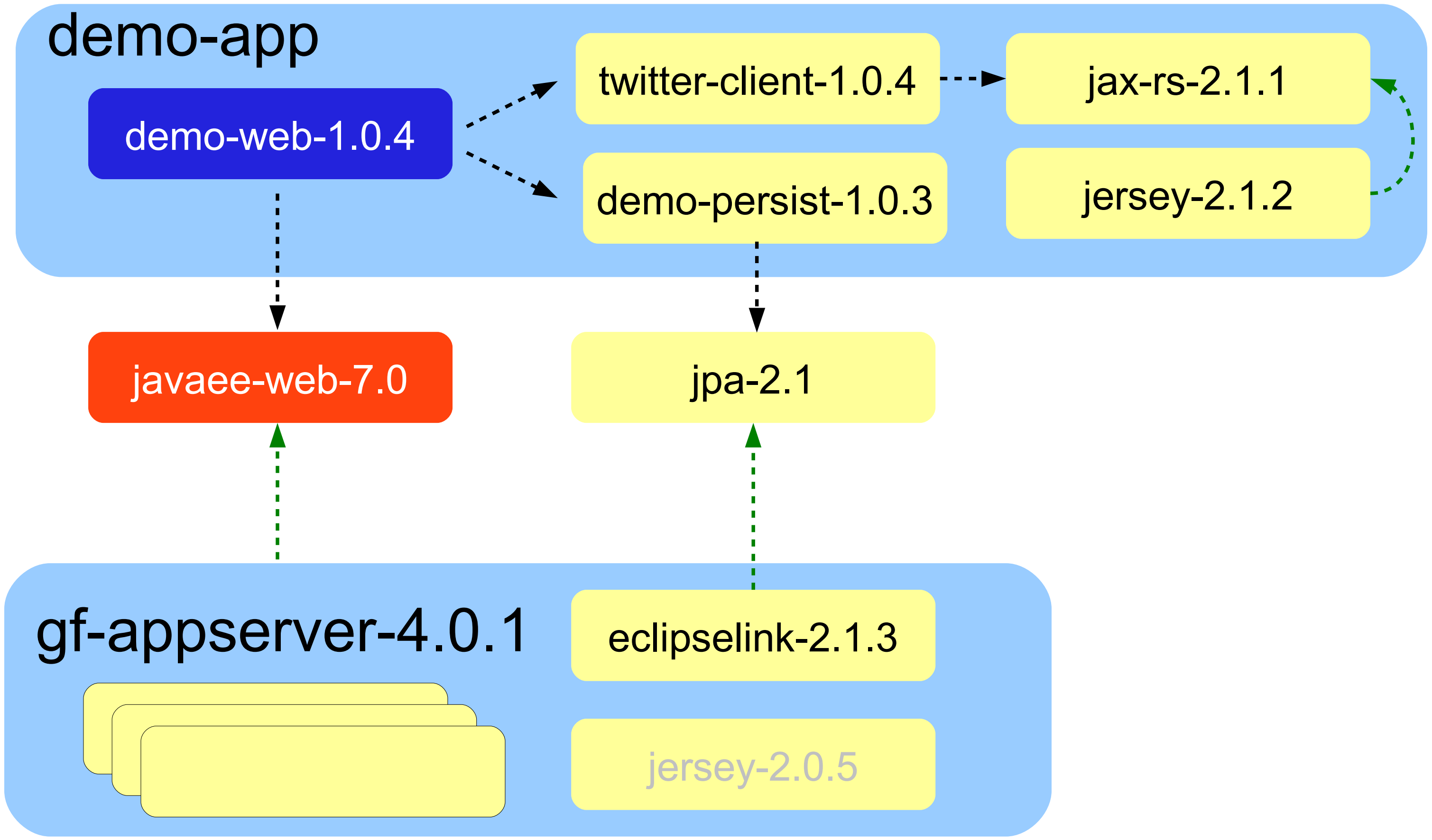
Modular Applications



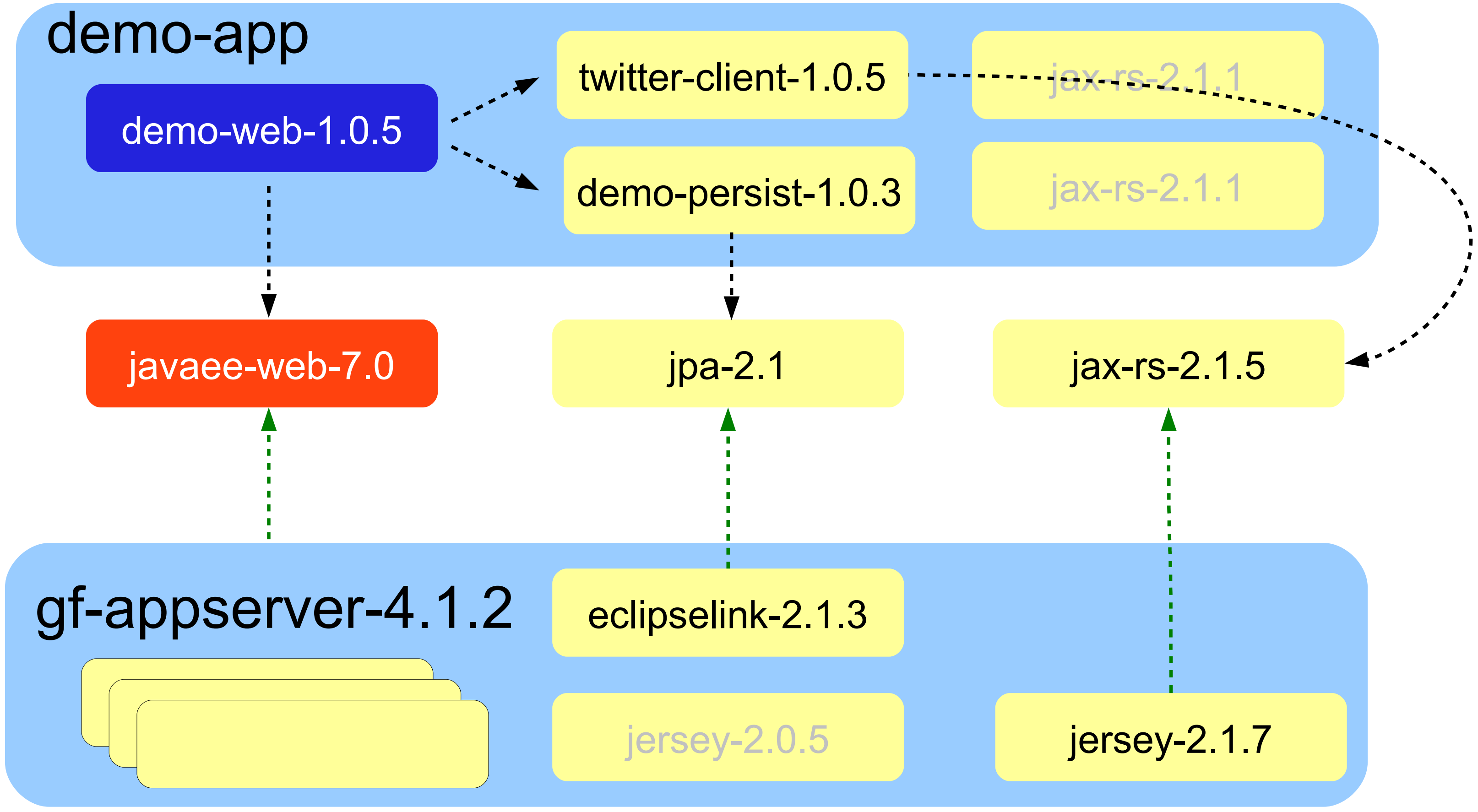
Modular Applications



Modular Applications



Modular Applications



Java EE 7 JSR Soup

- Java Persistence API 2.1 – JSR 338
- JAX-RS 2.0 – JSR 339
- Servlets 3.1 – JSR 340
- Expression Language 3.0 – JSR 341
- Java EE 7 – JSR 342
- Java Message Service 2.0 – JSR 343
- Java Server Faces 2.2 – JSR 344
- EJB 3.2 – JSR 345
- CDI 1.1 – JSR 346
- JCache – JSR 107
- Bean Validation 1.1 – JSR 349
- ...

Servlets 3.1 (JSR 340)

http://jcp.org/en/jsr/detail?id=340
http://servlet-spec.java.net

- Cloud support
- Multi-tenancy
  - Security / Session state / Resources isolation
- Asynchronous IO based on NIO2
- Simplified Asynchronous Servlets
- Utilize Java EE concurrency utilities
- Enable support for Web Sockets

JPA 2.1 (JSR 338)

http://jcp.org/en/jsr/detail?id=338
http://jpa-spec.java.net

- Multi-tenancy
- Support for stored procedures, vendor function
- Update and Delete Criteria queries, JPQL ↔ Criteria
- Query by Example
- Support for schema generation
- Persistence Context synchronization control
- Dynamic definition of PU
- Additional event listeners

EJB 3.2 (JSR 345)

http://jcp.org/en/jsr/detail?id=345

- Enablement for use in cloud
- Factorization of the EJB technology
  - Interceptors was the first example
  - Container-managed transactions as target
- Alignment with other specifications
- Mark “pruned” technologies as optional
  - EJB 1.x and 2.x entity beans
  - Web service invocation using JAX-RPC

JAX-RS 2.0 (JSR 339)

http://jcp.org/en/jsr/detail?id=339
http://jax-rs-spec.java.net

- Client API
  - Low level using Builder pattern, Higher-level
- Hypermedia
- MVC Pattern
  - Resource controllers, Pluggable viewing technology
- Bean Validation
  - Form or Query parameter validation
- Closer integration with @Inject, etc.
- Server-side asynchronous request processing
- Server-side content negotiation

CDI 1.1 (JSR 346)

http://www.jcp.org/en/jsr/proposalDetails?id=346

- Global ordering of interceptors and decorators
- API for managing built-in contexts
- Embedded mode to startup outside Java EE container
- Send Servlet events as CDI events

Expression Language 3.0 (JSR 341)

http://jcp.org/en/jsr/detail?id=341
http://el-spec.java.net

- A JSR by itself
- Make EL easier to use outside EE container
  - Simplified to use in Java SE
- EL Context is split into Parsing and Evaluation context
- Explicit coercion rules using API
- Criteria-based selection from Collection
- Operators: ==, concat, sizeof
- CDI events for expression evaluation

JMS 2.0 (JSR 343)

http://jcp.org/en/jsr/detail?id=343
http://jms-spec.java.net

- Long overdue – after 9 years
- Modest scope, major extensions deferred to a subsequent revision
- Ease-of-development
- Clarification of relationship with other Java EE specs
- New mandatory API for pluggable JMS provider

Bean Validation 1.1 (JSR 349)

http://jcp.org/en/jsr/detail?id=349

- Integration with other specs
  - JAX-RS: Validate parameters on HTTP calls
  - JAXB: convert into XML schema descriptor
  - JPA: DDL generation
- Method level validation

```
public void processOrder(@Valid Order order,
                         @Min(0) @Max(30) int retry) {
}
```

- @Valid and group propagation
- Apply constraints on element collection

JSF 2.2 (JSR 344)

http://jcp.org/en/jsr/detail?id=344
http://jsf-spec.java.net

- Ease of development
  - cc:interface is optional
  - JSF lifecycle is CDI aware
  - Runtime configuration options change
- Support implementation of Portlet Bridge 2.0
- Support for HTML5 features
  - Forms, Heading/Section content model, ...
- New components like FileUpload and BackButton

Java EE 7 : Technology Refresh

- Ease-of-development: JMS 2.0
- Latest web standards
  - New JSRs: Web Sockets, Java JSON API
  - HTTP Client API (JAX-RS 2.0)
- Possible JSRs inclusion
  - Concurrency Utilities for Java EE (JSR 236)
  - JCache (JSR 107)

Transparency Checklist

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

NEW

- EG members names
- EG business reported on publicly readable alias
- Schedule is public, current and updated regularly
- Public can read/write to a wiki
- Discussion board on jcp.org
- Public read-only issue tracker

Java EE 7 – When ?

- Late 2012
- Date-driven release
  - Anything not ready will be deferred to Java EE 8
- Participate
  - Expert Groups forming
  - Public discussion lists
  - JCP membership free for individuals

GlassFish Server Distributions

Distribution	License	Features
GlassFish Server Open Source Edition 3.1 *Web Profile*	CDDL & GPLv2	• Java EE 6 compatibility • Web Profile support • In-memory replication / clustering • Centralized Administration
GlassFish Open Source Edition 3.1	CDDL & GPLv2	• Java EE 6 compatibility • Full Java EE distribution • In-memory replication / clustering • Centralized Administration
Oracle GlassFish Server 3.1 *Web Profile*	Commercial	• Adds • Oracle GlassFish Server Control • Patches, support, knowledge base
Oracle GlassFish Server 3.1	Commercial	• Adds • Oracle GlassFish Server Control • Patches, support, knowledge base

References

- oracle.com/javaee
- glassfish.org
- oracle.com/goto/glassfish
- blogs.oracle.com/theaquarium
- youtube.com/GlassFishVideos
- Follow @glassfish

The Java EE 7 Platform: Developing for the Cloud

Arun Gupta, Java EE & GlassFish Guy
blogs.oracle.com/arungupta, @arungupta