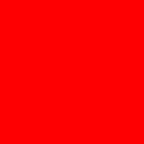




The Java EE 7 Platform: Reaching for the Cloud

Roberto Chinnici
Java EE Platform Lead



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.

Agenda

- Java EE 7 theme
- Main features
- Highlights of some new APIs
- Component JSRs
- Schedule

Java EE 7 Theme : The Cloud

- “We’re making Java EE ready for the cloud”
- Support for Platform-as-a-Service (PaaS) model

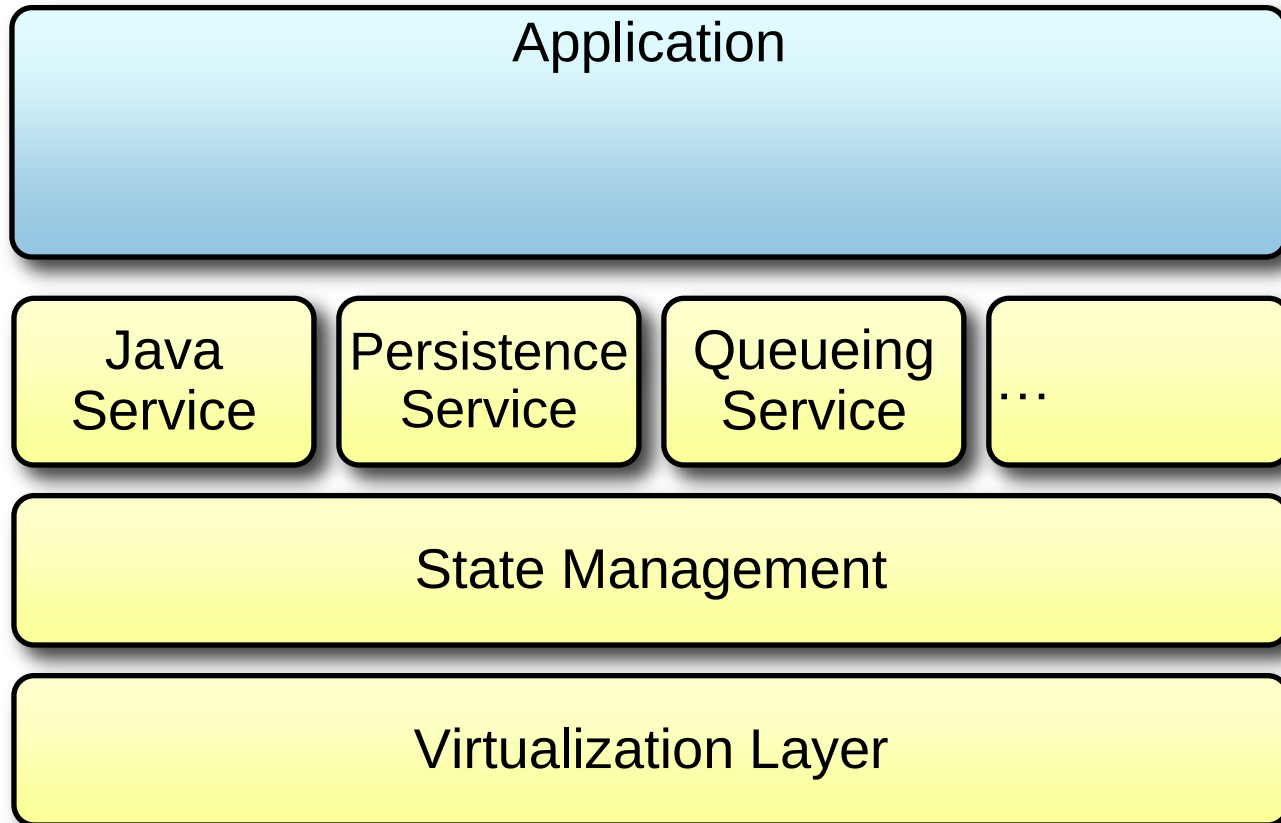
Java EE for the Cloud

- Tighter requirements for resource and state management
- Better isolation between applications and their data, using tenants as an isolation unit
- Standard APIs for caching in EE 7, perhaps MapReduce, NORDBMS in the future
- Common management and monitoring interfaces
- Evolution, not revolution

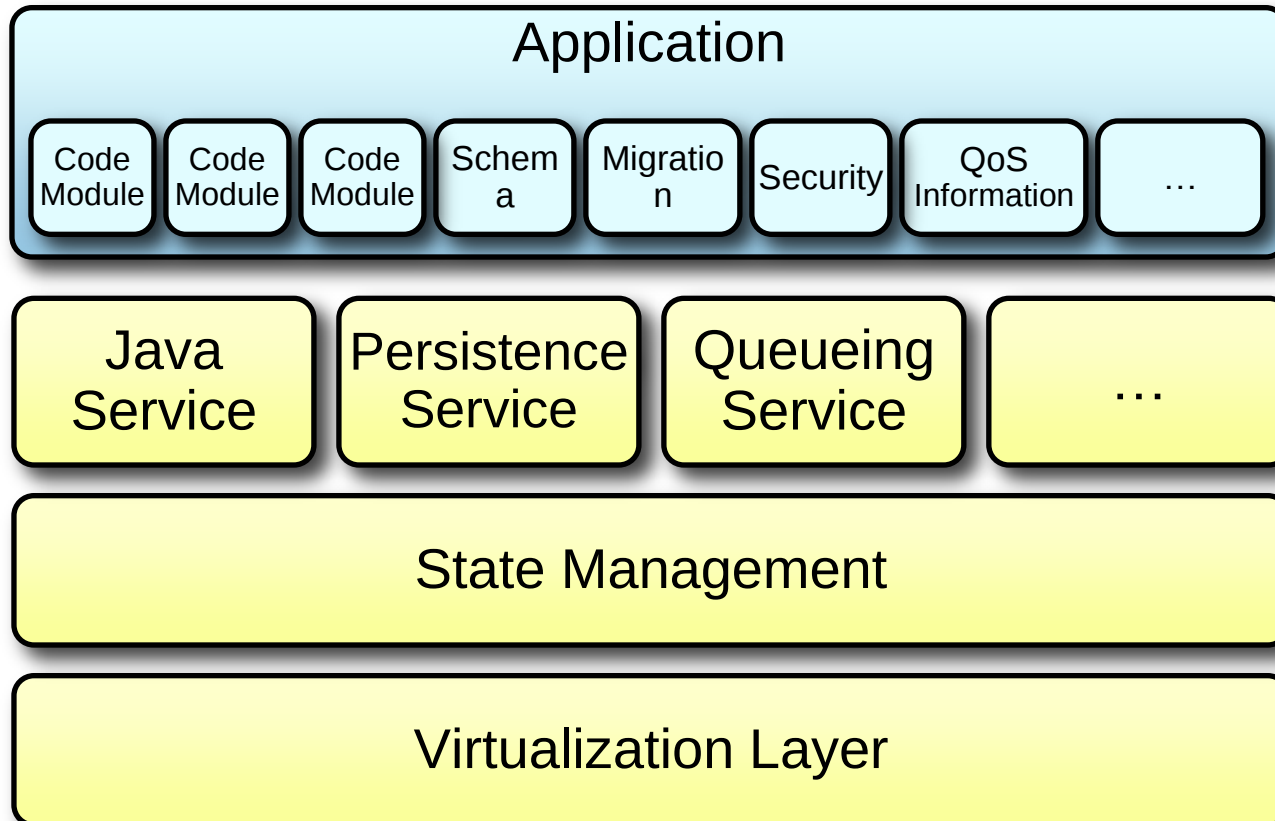
Better Packaging for the Cloud

- Apps are versioned
- Multiple versions can coexist
- Must deal with data versioning, upgrades, etc.
- Need ability to specify QoS properties
- Apps both expose and connect to services

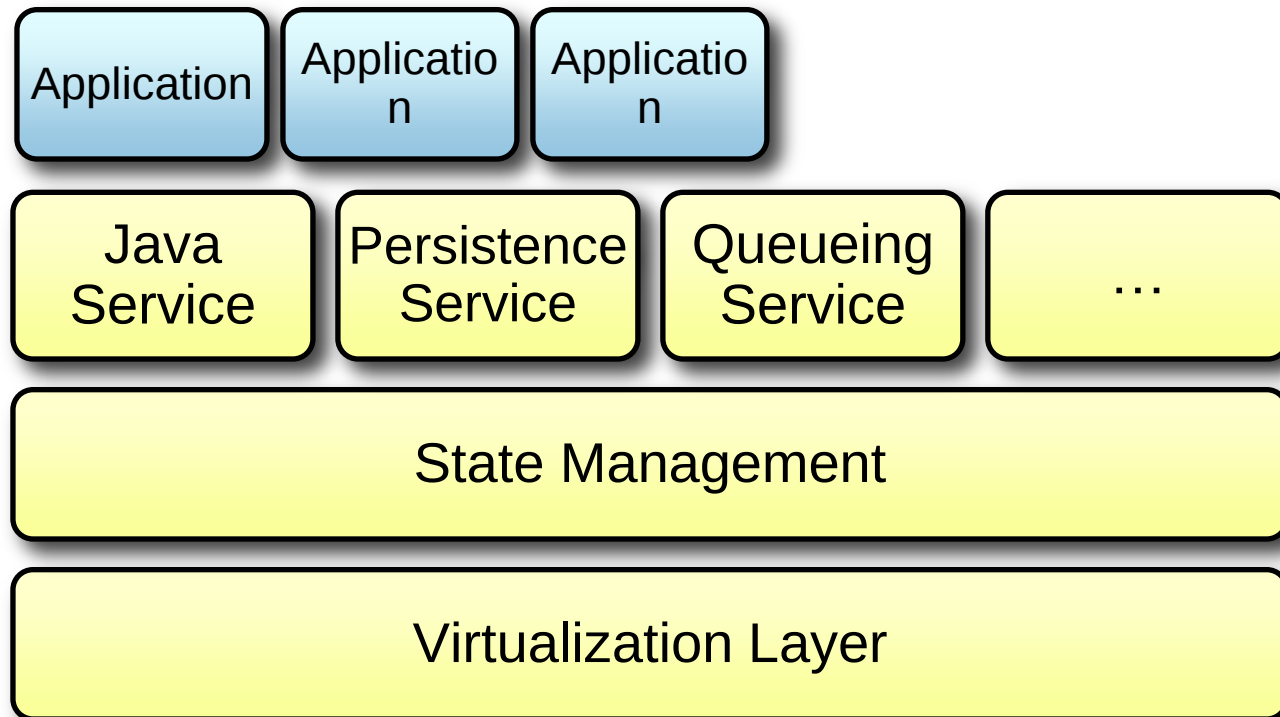
Cloud Platform



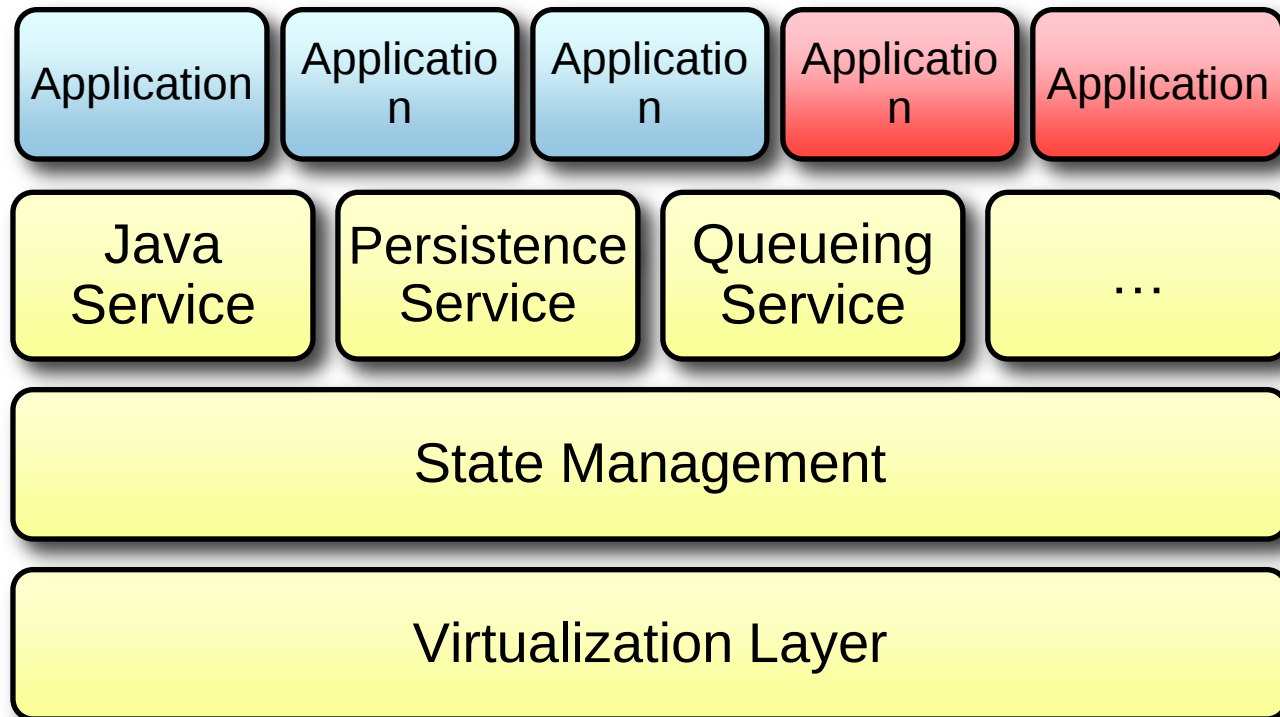
Cloud Platform



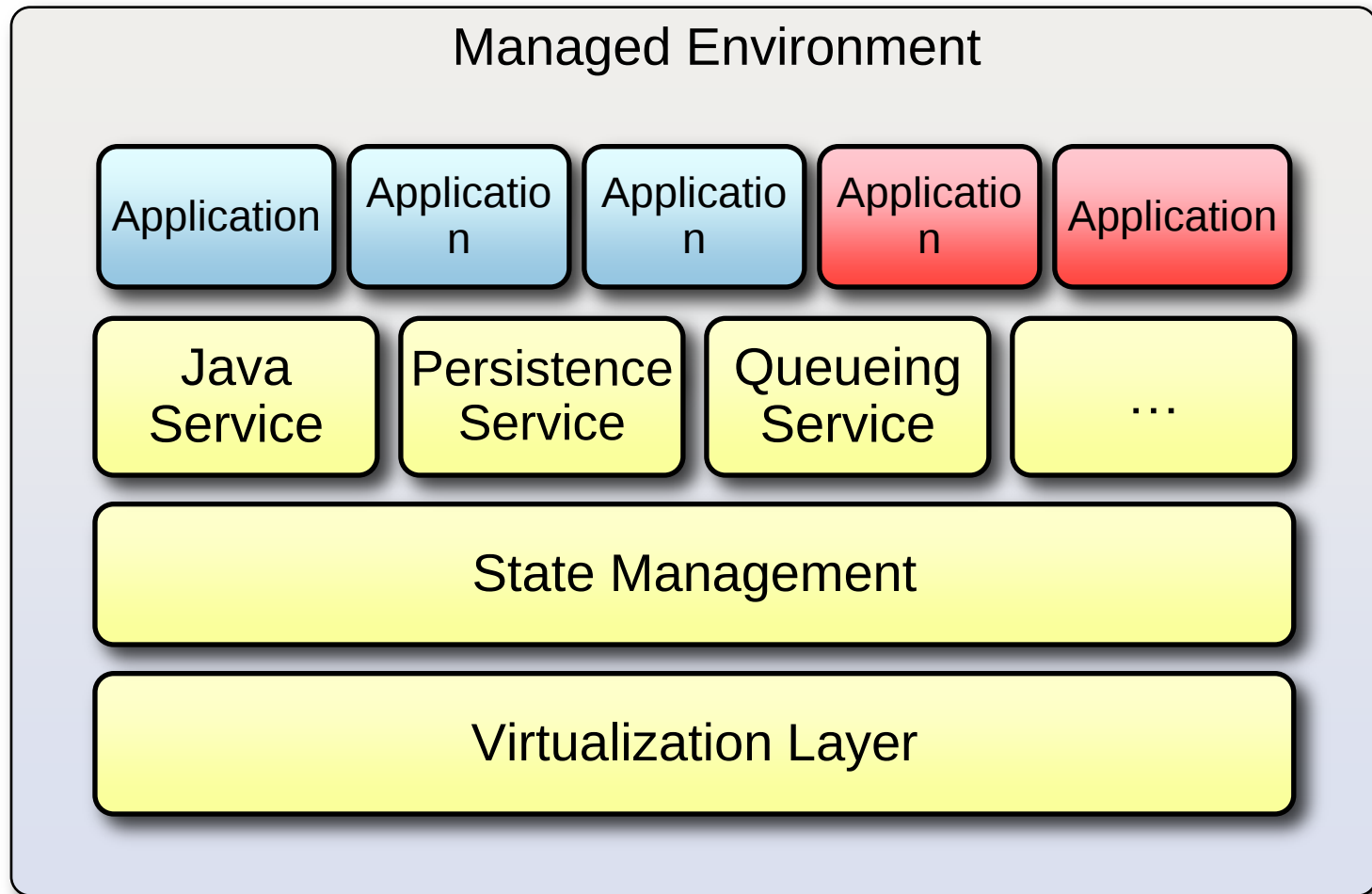
Cloud Platform



Cloud Platform



Cloud Platform



PaaS Support

- New roles:
 - PaaS Provider, PaaS Customer (Tenant), Application Administrator, End-User
- Multitenancy
 - Application does not directly manage the association to a tenant
 - Containers may use e.g. virtual servers to map a request to a tenant
 - Containers and resource managers cooperate to propagate tenant identity across tiers
- Extensible descriptor for service metadata (e.g. QoS)
- Connectionless APIs for convenience

Modularity

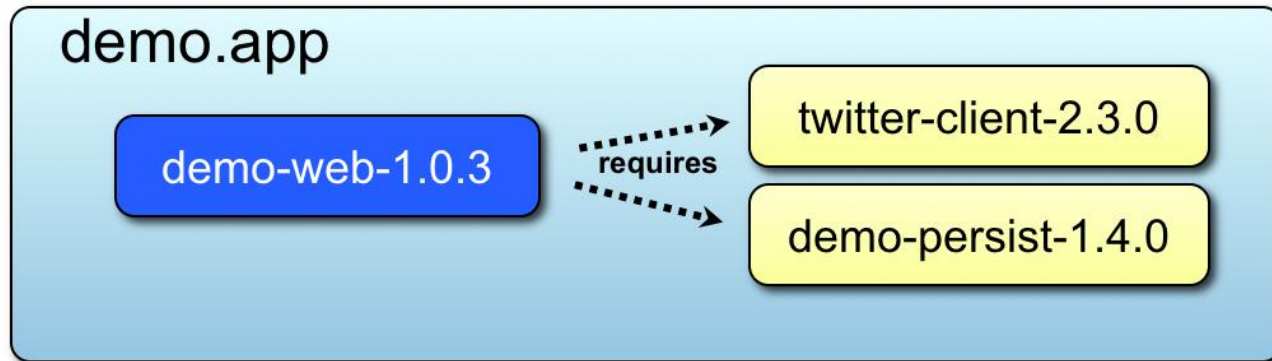
- Deliver modularity metadata ahead of Java SE 8
- Alignment with SE 8 is essential
- Use cases:
 1. Enable easy updating of an API to the latest version
 2. Support applications composed of versioned modules
- Full modularity support deferred to EE 8
 - Classloader changes

Modular Applications

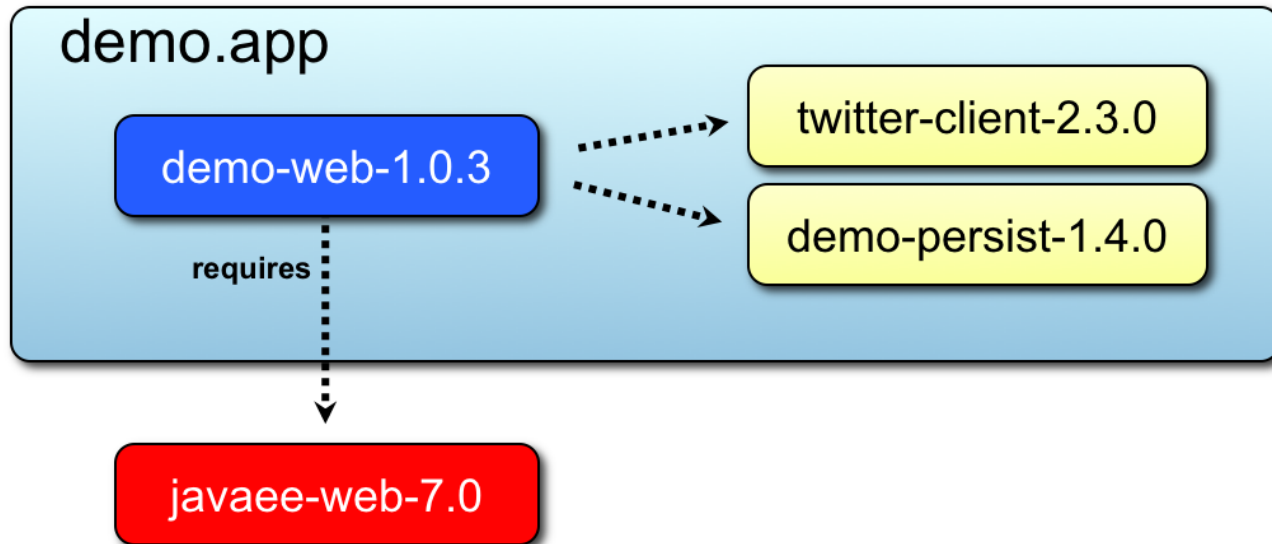
demo.app

demo-web-1.0.3

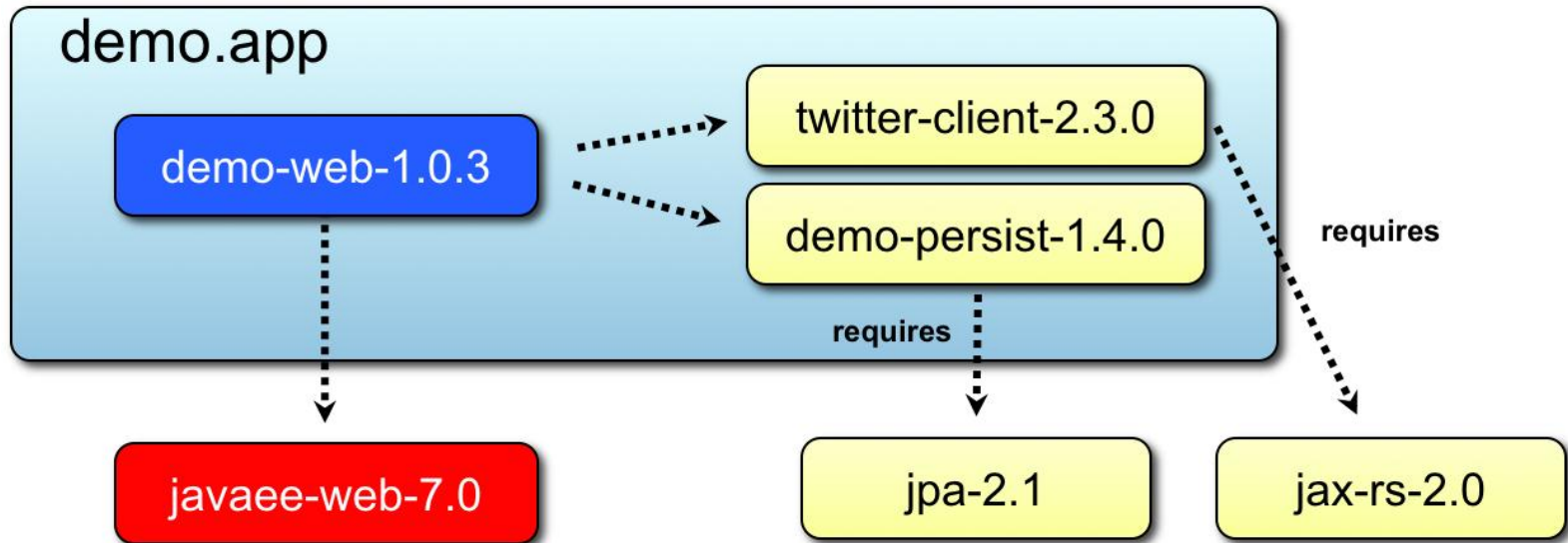
Modular Applications



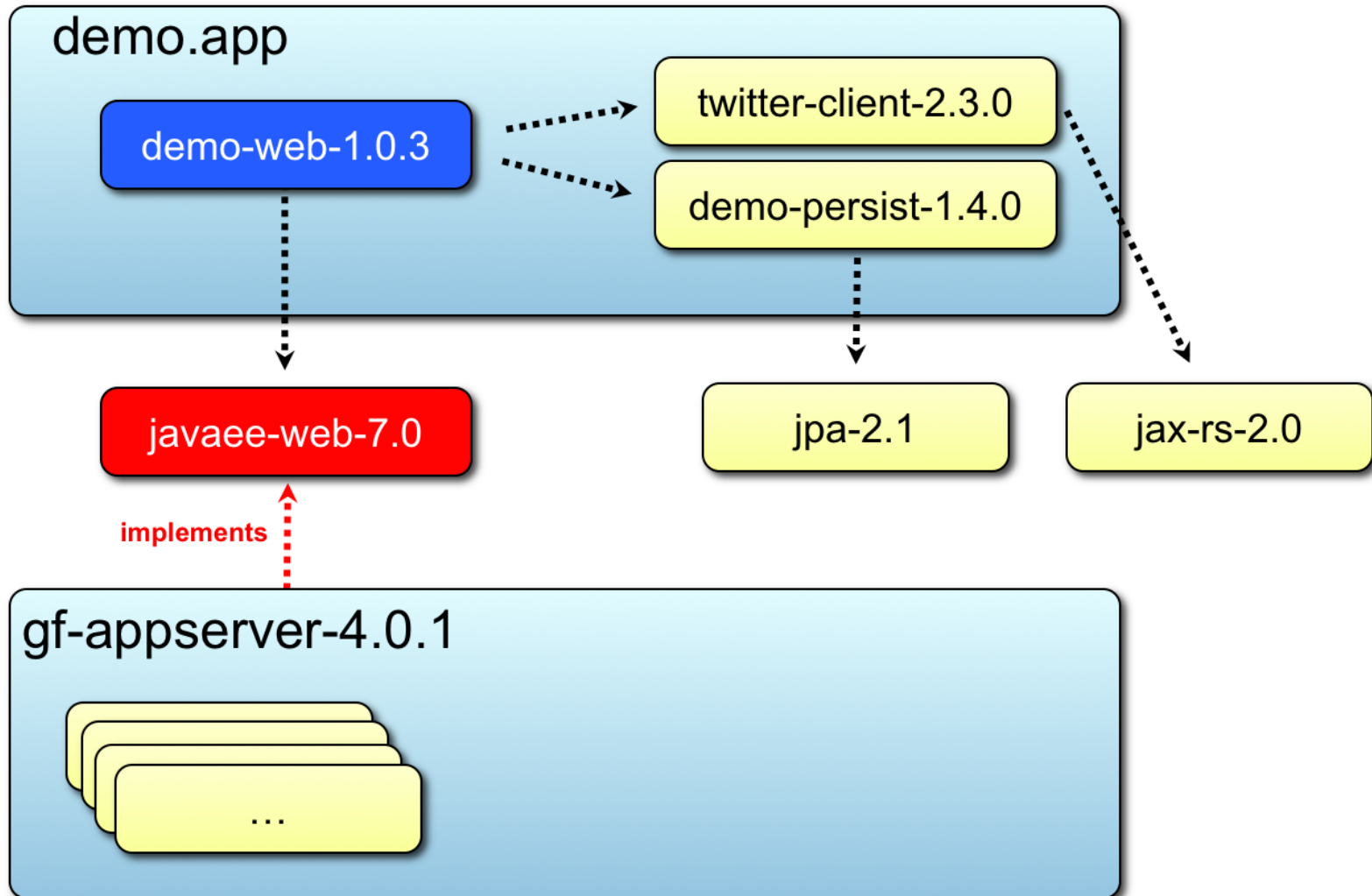
Modular Applications



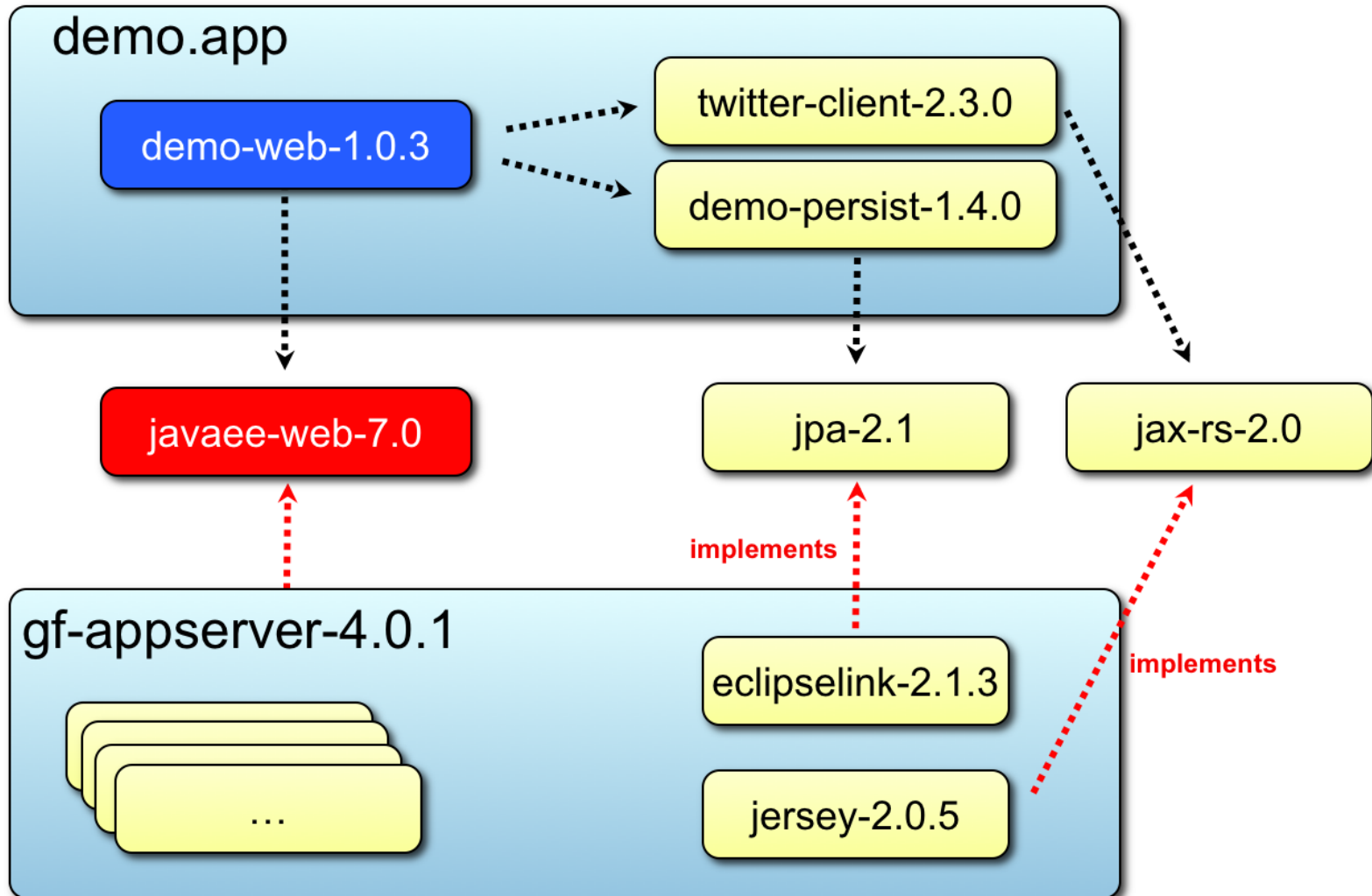
Modular Applications



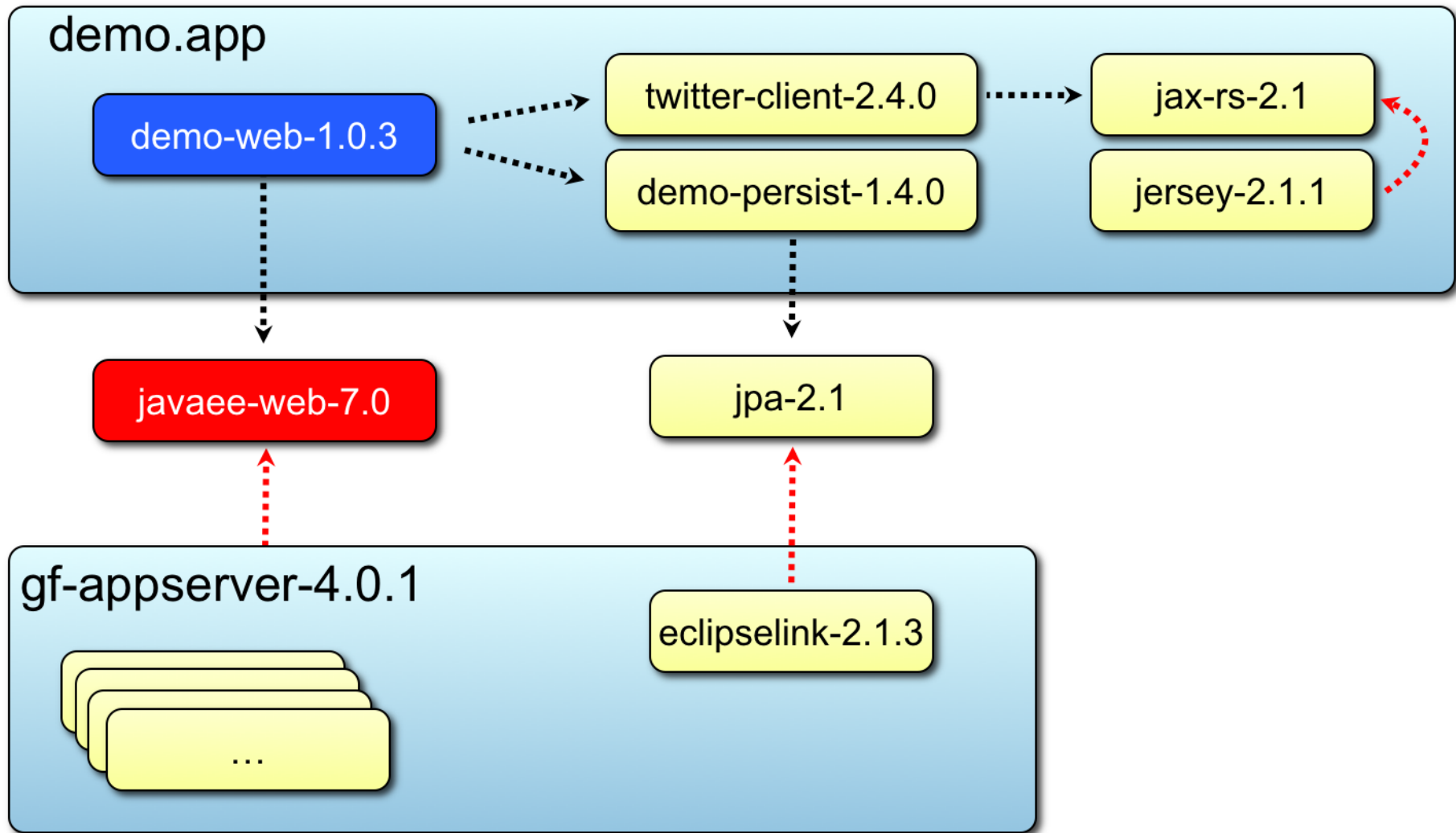
Modular Applications



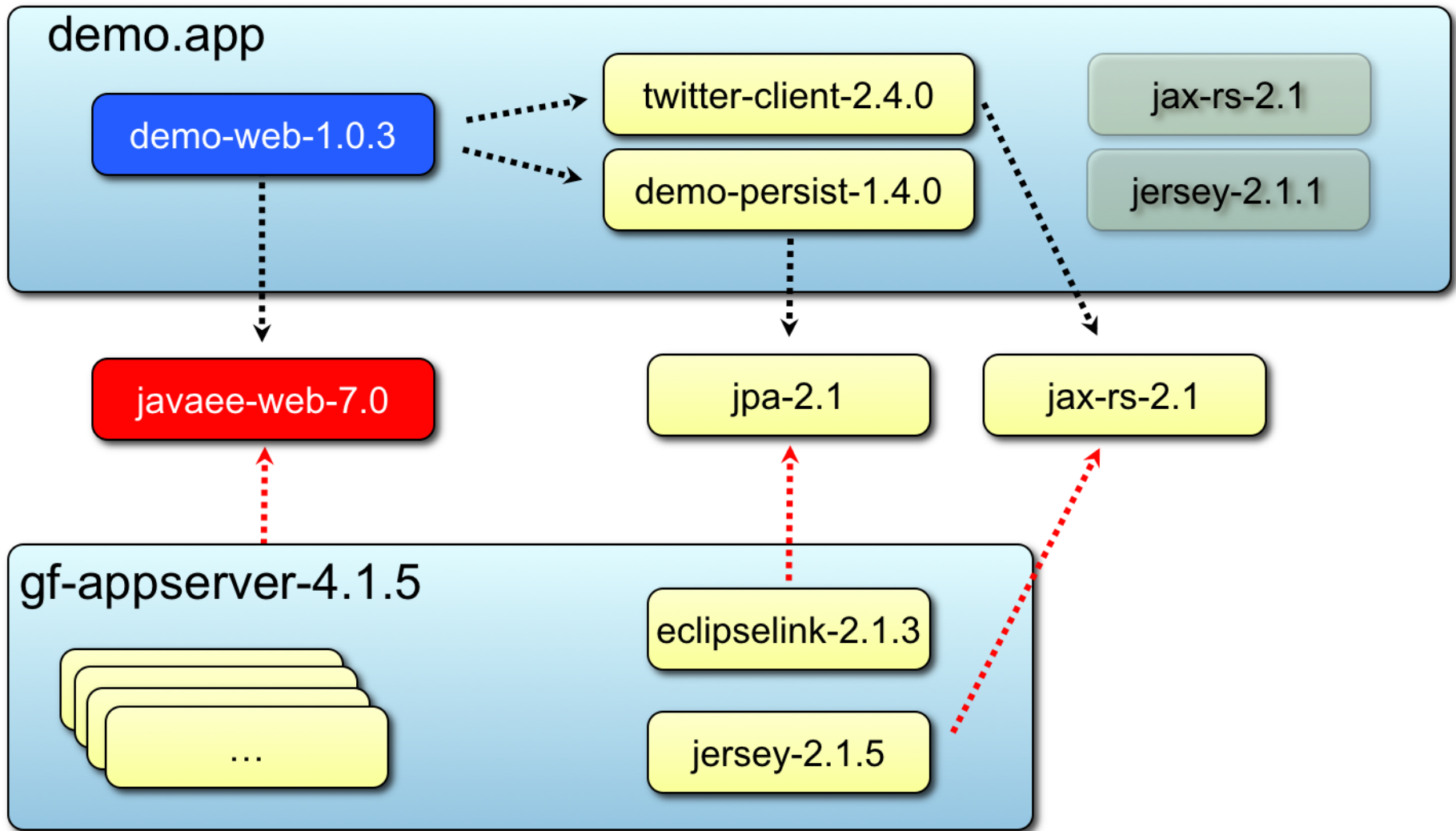
Modular Applications



Modular Applications



Modular Applications



Web Tier

- Web socket support in the servlet container
- Standard JSON API
- HTML5 support in JSF
- NIO.2-based web container
- Update the Web Profile definition
 - JAX-RS 2.0 will be a required component

New APIs

- Concurrency Utilities (JSR-236)
 - E.g. a transaction-, security-, naming-aware ExecutorService
- JCache (JSR-107)
 - Local cache API, a key-value map
 - Needs to be aligned with the JPA Cache API
- JMS 2.0
 - Brings annotation-based programming model to JMS
- JSON 1.0
- RESTful client API in JAX-RS 2.0

RESTful Client API (using Jersey)

```
WebResource r = Client.create().resource("http://example.com");
WebResource content = r.path("containers").path("quotes");
content.path("1")
    .type(MediaType.TEXT_PLAIN)
    .put("Something is rotten in the state of Denmark");

ClientResponse response = content.path("1")
    .header("If-Modified-Since", lastModified)
    .header("If-None-Match", etag)
    .get(ClientResponse.class);

assertEquals(Response.Status.NOT_MODIFIED,
    response.getResponseStatus());
```

Web Socket Sample (using Grizzly WebSocket API)

```
public class ChatApplication
    extends WebSocketApplication<ChatWebSocket> {
    protected ChatWebSocket createWebSocket(Connection c,
                                           ServerWebSocketMeta m) {
        return new ChatWebSocket(c, m, this);
    }
    public void onMessage(ChatWebSocket s, DataFrame frame) {
        String msg = frame.getAsText();
        s.sendJson(s.getUser(), msg);
    }
    public void onClose(ChatWebSocket s) {
        s.sendJson("system", s.getUser() + " left the chat");
    }
}
```

Async Web API Sample (with Atmosphere)

```
@Path("/{topic}")
@Produces("text/plain;charset=ISO-8859-1")
public class PubSub {
    private @PathParam("topic") Broadcaster topic;

    @GET @Suspend
    public Broadcastable subscribe() {
        return new Broadcastable("", topic); }

    @POST @Broadcast
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    public Broadcastable publish(
        @FormParam("message") String message) {
        return new Broadcastable(message, topic);
    }
}
```

JPA 2.1 Candidate Features

- Converters / custom types
- “Fetch” plans / profiles
- Additional mapping metadata
- User-defined naming strategies
- More flexibility in use of generated values
- Immutable attributes; readonly entities
- More flexible XML descriptors
- Additional event listeners and callbacks
- Improved control of persistence context synchronization
- Support for dynamic definition of persistence unit
- Extension of Metamodel API to ORM
- Methods for dirty detection
- Additional unwrap methods
- Support for stored procedures
- Additional built-in functions
- Database and vendor function invocation support
- Downcasting
- Outer joins with ON conditions
- Update and delete criteria queries
- Mapping between JPQL and criteria queries
- Improved result type mapping for native queries

Java EE 7 JSRs

Approved

- JPA 2.1
- JAX-RS 2.0
- Servlet 3.1
- EL 3.0
- Platform 7 / Web Profile 7
- JMS 2.0
- JSF 2.2

Ongoing

- Concurrency Utilities 1.0
- JCache 1.0

Yet to be filed

- EJB 3.2
- CDI 1.1
- JSR-330 1.1
- Bean Validation 1.1
- JSON 1.0

Transparency

- High level of transparency for all EE JSRs
- One java.net project per specification
 - E.g. jax-rs-spec, jpa-spec
- Publicly viewable, archived EG mailing list
- Jcp.org mailing list for administrative/confidential info
- Users list gets a copy of all emails to the EG
- JIRA for issue tracking
- Download area

Schedule

- First seven JSRs already approved
- Remaining ones to be filed in the coming weeks
- Final release by Q3 2012
- Date-driven release: anything not ready will be deferred to Java EE 8



Questions?

ORACLE®



北京站 · 2012年4月18~20日
www.qconbeijing.com (11月启动)

QCon杭州站官网和资料
www.qconhangzhou.com

全球企业开发大会

INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE