Java EE 8 Update

Linda DeMichiel
Java EE 8 Specification Lead
Oracle
September 19, 2016
Safe Harbor Statement

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.
Program Agenda

1. Road to Java EE 8
2. Java EE 8 JSRs (Original Proposal)
3. Proposed Shift in Focus
4. Where to Learn More at JavaOne
Java EE 7

**Productivity**
- Annotated POJOs
- Less boilerplate code
- Integrated
- Excellent tool ecosystem

**HTML5-Ready**
- JSON
- WebSockets
- JAX-RS

**Meets Enterprise Demands**
- Java Message Service
- Batch processing
- Distributed transactions

**Scalable**
- Multi-threaded
- Asynchronous APIs (Servlet, EJB, JAX-RS)
- Concurrency utilities for Java EE

**Community Driven**
- Java Community Process
- Adopt-a-JSR
- Open Source RI (GlassFish)

**Industry Standard**
- Lowers risk
- Vendor choice
- Implementation choice
- Operating system choice
- Portable applications
The Vibrant Java EE Community

Publications

Java EE Developers

Career Opportunity

Java EE Compatible Application Servers

User Groups
Program Agenda

1. Road to Java EE 8
2. Java EE 8 JSRs (Original Proposal)
3. Proposed Shift in Focus
4. Where to Learn More at JavaOne
Java EE 8
Contents Driven by Community Input

Java EE 8 Community Survey 2014

- JSONB 14%
- Security Simplification 11%
- JCache 9%
- Security Interceptors 8%
- MVC 8%
- Config JSR 8%
- Mgt/Monitoring API 7%
- Embedded 7%
- SSE 7%
- Cloud 7%
- Logging 6%
- EJB Timer 4%
- Pruning 4%
Java EE 8

JSRs

• Java EE 8 Platform and Web Profile
• Contexts and Dependency Injection 2.0 (CDI)
• Java API for JSON Binding 1.0 (JSON-B)
• Java Message Service 2.1 (JMS)
• Java Servlet 4.0
• Java API for RESTful Web Services 2.1 (JAX-RS)
• Model-View-Controller 1.0 (MVC)
• JavaServer Faces 2.3 (JSF)
• Java EE Management API 2.0
• Java API for JSON Processing 1.1 (JSON-P)
• Java EE Security API 1.0
• Bean Validation 2.0
CDI 2.0

• Define behavior of CDI outside of a Java EE container
• Make CDI more modular to help other Java EE specs better integrate with it
• Spec split into 3 parts:
  – Core CDI
  – CDI in Java SE
  – CDI in Java EE
• API to bootstrap a CDI container in Java SE
• Observer ordering
• Asynchronous event firing
JSON-P 1.1

• Update JSON-P spec to stay current with emerging standards
• Support for IETF standards
  – JSON Pointer, JSON Patch, and JSON Merge Patch
• Add editing operations to JsonObject and JsonArray
• Helper classes and methods to better utilize Java SE 8 Stream operations
JSON-B 1.0

- JAXB-like API to marshal/unmarshal Java objects to/from JSON
- Default mapping between classes and JSON
- Customization APIs
- Standard support to handle “application/json” media type for JAX-RS
- Natural follow on to JSON-P – closes the JSON support gap
JAX-RS 2.1

• Server-sent events
• Non-blocking I/O in providers (filters, interceptors...)
• Reactive programming paradigm to improve JAX-RS asynchronous clients
• Hypermedia API enhancements
• Integration with other JSRs and frameworks
Servlet 4.0

• Support for HTTP/2
  – Request/response multiplexing
  – Server push
  – Upgrade from HTTP 1.1
• Compatibility with latest HTTP 1.1 RFCs
• Smaller community-requested improvements (JIRA issues)
JSF 2.3

• Better CDI integration
• WebSocket integration
• Ajax method invocation
• Class-level Bean Validation
• Java Date/Time support
MVC 1.0

• Provide action-based MVC framework
  – HTTP requests are routed to controllers and turned into actions by application code
  – Alternative/complement to JSF's component-based MVC framework

• Leverage existing technologies:
  – CDI, Bean Validation, Facelets, JSPs
JMS 2.1

• New API for receiving messages asynchronously
  – More flexible and general than MDBs
  – Alignment with CDI

• Improved portability of JMS providers between appservers

• Improved support for using JMS in XA transactions

• Improved handling of "bad" messages
Java EE Management API 2.0

•REST-based management APIs
  – Supersede current management EJB-based APIs of J2EE Management 1.0
  – Superset of functionality of J2EE Management 1.0
•Simple deployment API
Java EE Security 1.0

• API for managing users and groups
• Support for password aliasing
• API for role mapping
• Metadata and API for authentication
• Interceptors for authorization, with CDI support
Bean Validation 2.0

- Constraints applied to collection elements
- Support for new Date/Time API
- Integration with Optional wrappers
- Repeatable annotations
- Additional features requested from community
Where Are We Now?
Java EE 8 Progress to Date

**CDI 2.0 (JSR 365)**
- Bootstrap API for Java SE
- Async events
- Observer ordering

**Servlet 4.0 (JSR 369)**
- HTTP/2 support

**JSF 2.3 (JSR 372)**
- Small-scale new features
- Community-driven improvements

**Security 1.0 (JSR 375)**
- Authentication/authorization APIs

**JSON-B 1.0 (JSR 367)**
- JSON <-> object mapping

**JAX-RS 2.1 (JSR 370)**
- Reactive enhancements
- Server-sent events
- Non-blocking I/O

**Management 2.0 (JSR 373)**
- REST-based APIs

**Bean Validation (JSR 380)**
- Collection constraints
- Date/Time support
- Community-requested features

**JMS 2.1 (JSR 368)**
- Flexible JMS MDBs
- Improved XA support

**MVC 1.0 (JSR 371)**
- Action-based MVC framework

**JSON-P 1.1 (JSR 374)**
- JSON Pointer and Patch
- Java Lambda support
Where Are We Now?

The World Has Changed

• Focus on deployment into the Cloud
• Focus on microservices
• Emphasis on more rapid evolution of applications
Program Agenda

1. Road to Java EE 8
2. Java EE 8 JSRs (Original Proposal)
3. Proposed Shift in Focus
4. Where to Learn More at JavaOne
Revised Java EE 8 Proposal

Modernizing Java EE for Cloud and Microservices

Need to retarget Java EE to address these trends with 2-fold approach

1. Java EE 8 adjustment in focus
2. Java EE 9 – longer term, more in-depth work targeted at enhanced support for Cloud and microservices, leveraging work done in Java EE 8
Revised Java EE 8 Proposal
Modernizing Java EE for Cloud and Microservices

Goals

• Migration path to cloud development and deployment models for Java EE customers
• Migration path to microservices-based architecture for Java EE applications
• Backwards compatibility with Java EE
Revised Java EE 8 Proposal

<table>
<thead>
<tr>
<th>CDI 2.0 (JSR 365)</th>
<th>JSON-B 1.0 (JSR 367)</th>
<th>JMS 2.1 (JSR 368)</th>
<th>JAX-RS 2.1 (JSR 370)</th>
<th>MVC 1.0 (JSR 371)</th>
<th>JSON-P 1.1 (JSR 374)</th>
<th>Management 2.0 (JSR 373)</th>
<th>Bean Validation 2.0 (JSR 380)</th>
<th>Security 1.0 (JSR 375)</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap API for Java SE</td>
<td>JSON &lt;&gt; object mapping</td>
<td>Flexible JMS MDBs</td>
<td>JSON-B 1.0 (JSR 367)</td>
<td>Action-based MVC framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Standard for externalizing application configuration</td>
</tr>
<tr>
<td>Async events</td>
<td></td>
<td>Improved XA support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observer ordering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REST-based APIs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servlet 4.0 (JSR 369)</td>
<td>Flexible JMS MDBs</td>
<td>JSON-&lt;-&gt; object mapping</td>
<td>Reactive enhancements</td>
<td>Action-based MVC framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/2 support</td>
<td></td>
<td></td>
<td>Server-sent events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Non-blocking I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Client-side circuit breakers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAX-RS 2.1 (JSR 370)</td>
<td>JSON-&lt;-&gt; object mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSF 2.3 (JSR 372)</td>
<td>Reactor enhancements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small-scale new features</td>
<td>Server-sent events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Community-driven improvements</td>
<td>Non-blocking I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client-side circuit breakers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security 1.0 (JSR 375)</td>
<td></td>
<td>Flexible JMS MDBs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rationale for Proposed Changes

New Functionality

- Cloud apps make many remote REST calls. Need a **client-side circuit breaker** added to JAX-RS
- Need a **secret vault** because there’s no way to do this today using **standards**
- Need **OAuth and OpenID** support because those technologies have rapidly emerged as standards
- Need **externalized configuration store** to make applications retargetable across environments
- Need basic **multi-tenancy** support to accommodate needs of more complex apps and offer higher density
- Need **standard way of health checking** Java-based apps

Dropped Functionality

- **JMS** is no longer very relevant in cloud. Proposed to stay at JMS 2.0 standard (vs. upgrading to JMS 2.1).
- Cloud apps often ship headless, making **MVC** largely irrelevant
- Current **Management** JSR not widely used
Technical Focus Areas

Programming Model
- Extend for reactive programming
- Unified event model
- Event messaging API
- JAX-RS, HTTP/2, Lambda, JSON-B, ...

Key Value/Doc Store
- Persistence and query interface for key value and document DB

Eventual Consistency
- Automatically event out changes to observed data structures

Configuration
- Externalize configuration
- Unified API for accessing configuration

Resiliency
- Extension to support client-side circuit breakers
- Resilient commands
- Standardize on client-side format for reporting health

Packaging
- Package applications, runtimes into services
- Standalone immutable executable binary
- Multi-artifact archives

Serverless
- New spec – interfaces, packaging format, manifest
- Ephemeral instantiation

Multitenancy
- Increased density
- Tenant-aware routing and deployment

Security
- Secret management
- OAuth
- OpenID

State
- API to store externalized state

Eventual Consistency
- Automaically event out changes to observed data structures

Key Value/Doc Store
- Persistence and query interface for key value and document DB

Configuration
- Externalize configuration
- Unified API for accessing configuration

Resiliency
- Extension to support client-side circuit breakers
- Resilient commands
- Standardize on client-side format for reporting health

Packaging
- Package applications, runtimes into services
- Standalone immutable executable binary
- Multi-artifact archives

Serverless
- New spec – interfaces, packaging format, manifest
- Ephemeral instantiation

Multitenancy
- Increased density
- Tenant-aware routing and deployment

Security
- Secret management
- OAuth
- OpenID

State
- API to store externalized state

Eventual Consistency
- Automaically event out changes to observed data structures
## Proposal for Reactive Programming

### Problem Statements

- Need to incorporate evolving reactive/async-style programming model
- More common in Cloud because apps are distributed (split into microservices) and there is increased latency
  - Many remote calls.
  - Synchronous request-handling blocks threads with remote calls

### Proposal

- Migration path to fuller reactive programming model in Java EE 9
- Improve JAX-RS to support reactive programming for client side (e.g., async "orchestration" as in RXJava or in Jersey)
## Proposal for Circuit Breaker – Resiliency

### Problem Statements

- Prevent request-handling threads from being consumed while making requests to remote systems
- Ease up on requests to remote system as it's having problems
- Allow system time to recover
- Prevent cascading failures. Isolates failures in the source system
- Use circuit breaker without writing extensive boiler-plate code.

### Proposal

- Extension to JSR 339 - JAX-RS Client
- Several possible approaches:
  - Programmatic – change in JAX-RS Client API
  - Declarative – registering @Provider classes to the Client
  - Other...
- Configurable -- potential parameters might include:
  - Sampling frequency
  - Sampling time period
  - Performance threshold (milliseconds)
  - % error threshold
  - ...

---

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | 31
Proposal for Health Checking – Resiliency

Problem Statements

- No standard for health is being reported
- Applications, resources, servers, services, micro-services, etc. will report health differently
- Traditional health check just returns opaque up/down messages. If an instance is having problems, it should report the root cause(s) where possible to help diagnose/locate problems

Proposal

- Define standard for how individual instances should report health
  - REST API with structure format in JSON
- Define configurable context path – e.g. /healthcheck
- Define semantics for reporting health
  - Health status codes
  - Reason(s) for failures, warnings, etc.
  - Health of dependencies
- Enhancements to Java APIs to facilitate easier development
  - New annotations and descriptors to specify health endpoints
- Circuit breaker could poll /healthcheck rather than waiting for HTTP requests to fail first
Proposal for Security

Problem Statements

- No standard way of connecting an application to a key service
- Need to keep sensitive stored data secret
- Hard to use OAuth
- OpenID is emerging as the default authentication standard

Proposal

- Standard way of connecting an application to a key service
- Encryption service for stored data
- Improved OAuth support
  - Registration and Discovery of Resources to Request Scopes
  - Authorization Model
- OpenID support for authentication
## Problem Statements

- No standard way of working with configuration in applications
- No easy way of moving applications between environments without hacking into their packages
- Changing application configuration without redeployment of app
- Reconfiguring multiple instances of an application at once
- Externalized configuration is the standard for cloud

## Proposal

- Standardize a mechanism of defining, injecting and using configuration within an application
- Define configuration persistence mechanisms, formats, and bindings
- Provide an ability to externalize application configuration from the application package
- Support for merging, overriding, and federating configurations from different sources
- Provide a standard mechanism for working with mutable/dynamic configuration
## Proposal for Multi-tenancy

<table>
<thead>
<tr>
<th>Problem Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>- No concept of customer-facing tenancy within Java today</td>
</tr>
<tr>
<td>- One customer writing a multi-tenant App or Service has to define the concept of tenancy from scratch – hard and error-prone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>- <strong>Optional</strong> feature for servers supporting multi-tenancy</td>
</tr>
<tr>
<td>- Specification for mapping an external inbound request back to a tenant</td>
</tr>
<tr>
<td>- API for applications to find out which tenant the current request corresponds to</td>
</tr>
<tr>
<td>- Tenant-aware routing and deployment within an instance of an app server</td>
</tr>
</tbody>
</table>
Java EE 7

- Batch
- Dependency Injection
- JACC
- JAXR
- Management
- Bean Validation
- Deployment
- JASPIC
- JMS
- Servlet
- CDI
- EJB
- JAX-RPC
- JTA
- Web Services
- Common Annotations
- EL
- JAX-RS
- JSON-P
- Web Services Metadata
- Concurrency EE
- Interceptors
- JAX-WS
- JSP
- Managed Beans
- Connector
- JSP Debugging
- JAXB
- WebSocket
- Web Services
### Java EE 8 (Revised Proposal, 2016)

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Batch</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Bean Validation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>CDI</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Common Annotations</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Concurrency EE</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Dependency Injection</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EJB</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EL</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Interceptors</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JACC</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JASPIC</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JAX-RPC</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JAX-RS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JAX-WS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JAXB</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JAXR</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JMS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JTA</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JSF</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JSP</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JSTL</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JSON-B</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JSON-P</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JavaMail</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Managed Beans</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Web Services</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Web Services Metadata</strong></td>
<td></td>
</tr>
<tr>
<td><strong>WebSockets</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Health Check</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td></td>
</tr>
</tbody>
</table>

*Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |*
Java EE Roadmap

Engage Java EE Community
• Feedback through Survey
• Launch Java EE Next JSRs

Java EE 8
• Specs, RI, TCK complete
• Initial microservices support
• Define Java EE 9
• Early access implementation of Java EE 9

Java EE 9
• Specs, RI, TCK complete
• Modular Java EE runtime
• Enhanced microservices support
Next Steps

Give us your feedback

• Take the survey
  – http://glassfish.org/survey

• Send technical comments to
  – users@javaee-spec.java.net

• Join the JCP – come to Hackergarten in Java Hub

• Join or track the JSRs as they progress
  – https://java.net/projects/javaee-spec/pages/Specifications

• Adopt-a-JSR
  – https://community.oracle.com/community/java/jcp/adopt-a-jsr
Program Agenda

1. Road to Java EE 8
2. Java EE 8 JSRs (Original Proposal)
3. Proposed Shift in Focus
4. Where to Learn More at JavaOne
<table>
<thead>
<tr>
<th>Session Number</th>
<th>Session Title</th>
<th>Day / Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON7975</td>
<td>Enterprise Java for the Cloud</td>
<td>Monday 4:00 p.m.</td>
</tr>
<tr>
<td>CON1558</td>
<td>What's New in the Java API for JSON Binding</td>
<td>Monday 5:30 p.m.</td>
</tr>
<tr>
<td>BOF7984</td>
<td>Java EE for the Cloud</td>
<td>Monday 7:00 p.m.</td>
</tr>
<tr>
<td>CON4022</td>
<td>CDI 2.0 Is Coming</td>
<td>Tuesday 11:00 a.m</td>
</tr>
<tr>
<td>CON7983</td>
<td>JAX-RS 2.1 for Java EE 8</td>
<td>Tuesday 12:30 p.m.</td>
</tr>
<tr>
<td>CON8292</td>
<td>Portable Cloud Applications with Java EE</td>
<td>Tuesday 2:30 p.m.</td>
</tr>
<tr>
<td>CON7980</td>
<td>Servlet 4.0: Status Update and HTTP/2</td>
<td>Tuesday 4:00 p.m.</td>
</tr>
<tr>
<td>CON7978</td>
<td>Security for Java EE 8 and the Cloud</td>
<td>Tuesday 5:30 p.m.</td>
</tr>
<tr>
<td>CON7979</td>
<td>Configuration for Java EE 8 and the Cloud</td>
<td>Wednesday 11:30 a.m</td>
</tr>
<tr>
<td>CON7977</td>
<td>Java EE Next – HTTP/2 and REST</td>
<td>Wednesday 1:00 p.m.</td>
</tr>
<tr>
<td>CON7981</td>
<td>JSF 2.3</td>
<td>Thursday 11:30 a.m</td>
</tr>
</tbody>
</table>
Integrated Cloud
Applications & Platform Services
## Technical Focus Areas

### Programming Model
- Extend for reactive programming
- Unified event model
- Event messaging API
- JAX-RS, HTTP/2, Lambda, JSON-B, ...

### Key Value/Doc Store
- Persistence and query interface for key value and document DB

### Eventual Consistency
- Automatically event out changes to observed data structures

### Configuration
- Externalize configuration
- Unified API for accessing configuration

### Multi-tenancy
- Increased density
- Tenant-aware routing and deployment

### Serverless
- New spec – interfaces, packaging format, manifest
- Ephemeral instantiation

### State
- API to store externalized state

### Resiliency
- Extension to support client-side circuit breakers
- Resilient commands
- Standardize on client-side format for reporting health

### Security
- Secret management
- OAuth
- OpenID

---

[Image of JavaOne logo]
## Java EE 7

<table>
<thead>
<tr>
<th>Batch</th>
<th>Dependency Injection</th>
<th>JACC</th>
<th>JAXR</th>
<th>JSTL</th>
<th>Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bean Validation</td>
<td>Deployment</td>
<td>JASPIC</td>
<td>JMS</td>
<td>JTA</td>
<td>Servlet</td>
</tr>
<tr>
<td>CDI</td>
<td>EJB</td>
<td>JAX-RPC</td>
<td>JSF</td>
<td>Java Persistence</td>
<td>Web Services</td>
</tr>
<tr>
<td>Common Annotations</td>
<td>EL</td>
<td>JAX-RS</td>
<td>JSON-P</td>
<td>JavaMail</td>
<td>Web Services Metadata</td>
</tr>
<tr>
<td>Concurrency EE</td>
<td>Interceptors</td>
<td>JAX-WS</td>
<td>JSP</td>
<td>Managed Beans</td>
<td>WebSocket</td>
</tr>
<tr>
<td>Connector</td>
<td>JSP Debugging</td>
<td>JAXB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Java EE 8 (Revised Proposal, 2016)

<table>
<thead>
<tr>
<th>Batch</th>
<th>Dependency Injection</th>
<th>JACC</th>
<th>JAXR</th>
<th>JSTL</th>
<th>Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bean Validation</td>
<td>Deployment</td>
<td>JASPIC</td>
<td>JMS</td>
<td>JTA</td>
<td>Servlet</td>
</tr>
<tr>
<td>CDI</td>
<td>EJB</td>
<td>JAX-RPC</td>
<td>JSF</td>
<td>Java Persistence</td>
<td>Web Services</td>
</tr>
<tr>
<td>Common Annotations</td>
<td>EL</td>
<td>JAX-RS</td>
<td>JSON-B</td>
<td>JavaMail</td>
<td>Web Services Metadata</td>
</tr>
<tr>
<td>Concurrency EE</td>
<td>Interceptors</td>
<td>JAX-WS</td>
<td>JSON-P</td>
<td>Managed Beans</td>
<td>WebSocket</td>
</tr>
<tr>
<td>Connector</td>
<td>JSP Debugging</td>
<td>JAXB</td>
<td>JSP</td>
<td>Security</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Configuration</td>
<td>Health Check</td>
</tr>
</tbody>
</table>

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.