Java EE – eine Zwischenbilanz

Peter Doschkinow
Senior Java Architect

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technologie und Architektur
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 Overview
- Getting started with Java EE 7
  - Available bundles and sample applications
- Using Java EE 7 for HTML5 application development
  - HTML5 and Thin Server Architecture
  - Demo
- Project Avatar
The Java EE Journey

1998-2004

- Enterprise Java Platform
- Robustness
- Web Services

Ease of Development
Lightweight

2005-2012

EE 5, EE 6

- 20 specs

Java EE 7

Developer Productivity & HTML5

2013 - Future

- 32 specs
Java EE Community Momentum

- **Active and transparent mailing lists**: 26
- **JSRs Adopted**: 22
- **JUGs**: 19
- **Promoted Builds**: 89
- **Companies**: 32
- **Experts**: 187
- **Active JSRs**: 14
- **Spec Leads**: 16
- **Experts**: 32

**Java.net**

**Adopt a JSR**

**GlassFish**

**Active Projects**

**Specify Process**
Java EE 7 Themes

- More annotated POJOs
- Less boilerplate code
- Cohesive integrated platform
- JAX-RS
- WebSockets
- JSON
- Servlet NIO

- Batch
- Concurrency
- Simplified JMS
Top Ten Features in Java EE 7

- WebSocket client/server API
- Batch Applications
- JSON Processing
- Concurrency Utilities
- Simplified JMS API
- New JTA Annotations
- JAX-RS Client API
- Default Resources
- More annotated POJOs
- Faces Flow
Java EE 7 JSRs

CDI Extensions
- Web Fragments
- Servlet 3.1
- EJB 3.2
- Managed Beans 1.0
- JPA 2.1

JSF 2.2, JSP 2.3, EL 3.0
- JAX-WS 2.2
- JAX-RS 2.0
- Interceptors 1.2, JTA 1.2
- Common Annotations 1.1
- JMS 2.0
- JCA 1.7

CDI 1.1
- Bean Validation 1.1
- Batch 1.0
- Concurrency 1.0
- EJB 3.2

JSON 1.0
- WebSocket 1.0
GlassFish Server 4.0
Open Source Project

- Built in open source
- World’s first Java EE 7 App Server
- Lightweight, modular, easy to use
Developer Tools

- Since NetBeans 7.3.1, currently 7.4
  - Adds support for Java EE 7 profiles
  - Bundle GlassFish 4.0 runtime
  - New/Updated Wizards

- Eclipse 4.3 Kepler
  - Add Java EE 7 support to Eclipse WTP
  - GlassFish Plugin via Eclipse Market (OEPE 12.1.2)
Java EE 7 Implementation Deliverables


- **Java EE 7 SDK**
  - With GUI installer for Windows, Linux and Mac OS X
  - Web and full profile, English and multi-language
  - API docs, tutorial and samples

- **GlassFish 4.0 OSE** with GUI installer or as Zip archive

- **Java EE 7 RI** binaries and sources for the web and full profile

- **Maven dependencies and javadocs**
  - javaee-api-7.0.jar, javaee-web-api-7.0.jar, javaee-api-7.0-javadoc.jar
Java EE 7 Samples

- Included in Java EE 7 SDK or available through GlassFish Update Center
- 41 example applications for the new and updated technologies
  - Common documentation structure
    - Description, key features
    - Building, deploying and running
    - Troubleshooting
  - Maven based
    - Parent project does not aggregate all examples!
- Can be run from command line or IDE

file:///C:/<javaee7-sdk>/samples/docs/list.html
Java EE 7 Tutorial

file:///C:/<javaee7-sdk>/docs/javaee-tutorial/examples

- Included in Java EE 7 SDK or available through GlassFish Update Center
- More complete coverage of Java EE technologies
  - Some are migrated from Java EE 6 tutorial
  - Maven based
- Can be run from command line or IDE
Java EE 7 Demo

Build, deploy and run a sample application

- Build, deploy and run a sample application
  - Using NetBeans 7.4
  - GlassFish 4.0
- Java EE 7 features in Netbeans 7.4
- Java EE 7 features in GlassFish admin console
HTML5 Architectural Implications

The Browser Is the Platform

- HTML5 is the new UI across devices
  - Designed to address the cross-platform jungle
    - Multimedia, Graphics, Offline, Real-time Communication, Device Access, File access, Semantic markup, CSS3
    - Applications == HTML5 + JavaScript + CSS3 + Server Resources
- Requires a different programming approach
  - Servers no longer generating markup language
  - Clients responsible for presentation logic and execution
  - JavaScript is part of the domain model, JSON is the payload
  - No need for browser plugin
Thin Server Architecture

Runtime application presentation

User Interface
- DOM API
- HTML5
- JavaScript
- CSS

Browser

App download
- HTTP
- XHR
- WebSocket
- Server-Sent-Events

App Server
- Static Resource Services
- RESTful Data Services
- WebSocket Server Push
- Data Access

Input
- display

Output
- EIS
- DB

Web Storage
Thin Server Architecture
With Java EE

Clients
- HTTP/S
- JSON
- SSE
- XML
- Web Sockets

Java EE Server
- JAX-RS
- Data Services
  - JMS
  - JAXB
  - POJO/EJB
  - JPA
  - JCA

Data Sources
- EIS

Oracle Coherence
- SQL

EIS
Drawing Board Demo

http://github.com/jersey/hol-sse-websocket

- Collaborative drawing
- Two-page application
  - List of drawings
  - Drawing
- Demonstrating
  - Server-side: JAX-RS, JSON, WebSocket, SSE Java API
  - Client-side: HTML5 with AngularJS, JAX-RS, WebSocket, SSE Java and JavaScript API
  - JavaFX hybrid Java/HTML5 application
Drawing Board Demo

TSA - Architecture

Clients

- HTML5 Browser
- JavaFX

GlassFish 4.0

Data Service

- DataProvider
- POJO

JAX-RS/SSE
- Jersey

HTTP/S
- JSON

SSE
- JSON

WebSocket
- Endpoint

WebSocketSend.send(...)
- onEvent(...)
Drawing Board Demo

Technology usage

- JAX-RS: CRUD for drawings
- SSE: distributing the list of drawings to all connected clients
- WebSocket: distributing the updates of a drawing to all connected clients
- JSON: implementing of encoder/decoder of the WebSocket server endpoint
- Java – JavaScript bridge (WebEngine): modifying the AngularJS client by replacing the WebSocket/SSE JavaScript client communication with a Java implementation in the JavaFX client
Future of Java EE

- Cloud / PaaS
- HTML5 ++
- JSON Binding
- NoSQL
- Configuration
- State Management
- JCACHE

Java EE 8 and Beyond
- End-to-end TSA-framework for HTML5 applications
- Service components implemented in JavaScript
  - Using a node.js implementation with Nashorn
- View components implemented in JavaScript
  - Using HTML5 + Widgets + Data Binding with EL
  - Minimal JavaScript code needed
Avatar Architecture

Client

- DOM
- Generated JavaScript
- Application CSS
- Application JavaScript
- Application/Generated HTML
- Browser
- local storage

Server

- Application Services
- Avatar Modules
- Node Modules
- Avatar Runtime
- Avatar Compiler
- Server Runtime
- Nashorn JavaScript Engine
- JDK 8

Connections:

- REST/WS/SSE
- HTML/CSS/JS
Links

- **Java EE 7**
  - [http://javaee-spec.java.net](http://javaee-spec.java.net)

- **GlassFish 4.0**
  - [http://wikis.oracle.com/display/GlassFish](http://wikis.oracle.com/display/GlassFish)
  - [http://glassfish.java.net/quality/portal](http://glassfish.java.net/quality/portal)
  - [http://www.youtube.com/GlassFishVideos](http://www.youtube.com/GlassFishVideos)

- **HTML5**
  - [http://www.w3.org/TR/html5/](http://www.w3.org/TR/html5/)

- **Thin Server Architecture**
  - [http://www.thinserverarchitecture.com](http://www.thinserverarchitecture.com)

- **Avatar**
  - [http://avatar.java.net](http://avatar.java.net)
  - [http://avatar-js.java.net](http://avatar-js.java.net)
Hardware and Software

Engineered to Work Together
Eclipse Kepler

https://blogs.oracle.com/arungupta/entry/java_ee_7_support_in1

- Create Java EE 7 Eclipse projects or using Maven
- New facets for JPA 2.1, JSF 2.2, Servlet 3.1, JAX-RS 2.0, EJB 3.2
- Schemas and descriptors updated for Java EE 7 standards (web.xml, application.xml, ejb-jar.xml, etc)
- Tolerance for JPA 2.1 such as features can be used without causing invalidation and content assist for UI (JPA 2.1)
- Support for NamedStoredProcedureQuery (JPA 2.1)
- Schema generation configuration in persistence.xml (JPA 2.1)
- Updates to persistence.xml editor with the new JPA 2.1 properties
- Existing features support EE7 (Web Page Editor, Palette, EL content assist, annotations, JSF tags, Facelets, etc)
- Code generation wizards (New EJB, Servlet, JSP, etc.)
NetBeans 7.3.1

- Support for creating Java EE 7 projects using Maven and Ant
- Develop, Deploy, and Debug using GlassFish 4
- Bundled Java EE 7 javadocs
- CDI is enabled by default for new Java EE 7 projects (CDI 1.1)
- Create database scripts from Entity Classes (JPA 2.1)
- Java Persistence Query Language (JPQL) testing tool (JPA 2.1)
- RESTful Java client creation using JAX-RS 2.0 Client APIs (JAX-RS 2.0)
- New templates for JAX-RS 2 Filter and Interceptor (JAX-RS 2.0)
- New templates for WebSocket endpoints (WebSocket 1.0)
- JMS messages are sent using JMS 2 simplified API (JMS 2.0)
- Pass-through attributes are supported during Facelet page editing (JSF 2.2)
- Resource Library Contracts (JSF 2.2)
- @FlowScoped beans from editor and wizards (JSF 2.2)
- Support for EL 3.0 syntax in editor (EL 3.0)
- JSON APIs can be used with code completion (JSON 1.0)