Lessons learned from real-world deployments of Java EE 7

Arun Gupta, Red Hat
@arungupta
Java EE 7

- More annotated POJOs
- Less boilerplate code
- Cohesive integrated platform

WebSockets
- JSON
- Servlet 3.1 NIO
- REST

Batch
- Concurrency
- Simplified JMS

HTML 5

DEVELOPER PRODUCTIVITY

MEETING ENTERPRISE DEMANDS
• UN agency to protect and support refugees

• Help UN refugees to get benefits through secured card wallets
  • Register, issue card, POS, ATM, Settlement
  • Refugees get cash or use cards at merchants
Architecture
Technology

- **JSON**: Notification messages to agents (simple messages created using **JSONBuilder**, No dependency on 3rd party library)

- **Batch**: End of day tasks like posting of accounts, generation of reports, email to clients and banks

- **Concurrency**: Managed threads pool for work

- **EJB**: Backend services (locally and remotely)

- **JAX-WS**: Integration with WFP/UNHCR systems (exposed interface)
Technology

• JPA
  • Stored procedure calls (Oracle and SQL Server)
  • Two-way schema (model using JPA, generate schema, tune it, and regenerate entities)
    • Change requests
  • Default **CDI** enabled (ease of integration between different layers)
Additional Libraries

- PrimeFaces
- Great support in NetBeans
- Generator plugin (in NetBeans) that facilitate and streamline development of pages based on entity beans (e.g. Admin and Operations page)
- jQuery stock library (streamline UI development in few lines)
Runtime

- GlassFish 4
  - 2 instances deployed locally
  - Mandate to use commercial app server
- Java SE 7
- Oracle 11g and Microsoft SQL Server 2008
Tools

- NetBeans 7.4
  - Integrated GlassFish management
  - Many templates to start from
  - PrimeFaces generator plugin
  - Great compatibility with HTML5
Acknowledgments

@_tamanm
miles2run.org

- Track your daily runs
- Day or distance-based goals or community runs
- Social network for runners
- It's free
Join Miles2Run Today

Capture all your runs, anytime, anywhere.

Miles2Run is a fast and fun way to share your running activities with friends and family. Sign up, enter an activity, share, and track your progress. It's easy.
Architecture Diagram

Client - HTML 5 Web Application

Container


Profile View  Goal View  Authentication Callback View  Home View  Others

DATABASES

DB1  DB2  DB3
Technology Diagram
Technology

• User Interface
  • Thymeleaf: Java-based HTML5 templates
  • AngularJS based multiple SPA, consuming JAX-RS endpoints
  • Twitter Bootstrap theme
  • Authentication using JAX-RS filters
  • D3.js and C3.js for pretty graphs
Technology

• Middleware

• EJB + JPA services for RDBMS

• CDI services for NoSQL datastores

• Joda-Time API for working with Date

• Authenticating with various OAuth providers like Twitter using respective client API. Planning to upgrade to Keycloak.

• SFL4J with logback for logging
Additional Libraries

- Thymeleaf template engine for server side HTML5 views
- OAuth providers with client libraries
- Joda-Time for working with date/time
- jadira.usertype for storing date as UTC in database
- Database drivers for Redis and Mongo
- JUnit with Mockito for Unit testing
- Arquillian for integration testing
Runtime

- JDK 8
- WildFly 8.1.0 Final: 2 scalable instances running on OpenShift with HA Proxy Load Balancer

Data Stores

- MySQL - All business entities like Activity, Goal, User profile, etc.
- Redis - Generate timeline data using sorted sets and storing all counters
- MongoDB - Location based user recommendations, follower/following stored using in-built array types
Tools

- IntelliJ IDEA
- Git as version control
- Jenkins - planned for CI and deployment
Wish list for Java EE 8

- Support for NoSQL database
- Easy to add OAuth support to application and use existing OAuth providers
- Modern MVC framework with support for pluggable template engines
Acknowledgments

@shekhargulati
tipi.camp

• Campground management
  • Reservations, portal, email, monitoring, …
  • Targeting 25k+ campgrounds in Europe
• Customer-friendly booking portal
• On-premise or SaaS
• €1 or $1 / day
Technology

• Presentation
  • JavaServer Faces + Expression Language
  • Security: JAAS principal + JDBC-realm

• Middle Tier
  • EJB for business boundaries
  • Servlets with JSON for web hooks: response from Paypal and Sendgrid
  • JAX-RS for exposing API to partners
Technology

• Backend
  • JPA + Bean Validation, use @Index for database generation
  • Concurrency: Calculation of statistics and sending messages (Using @Startup @Singleton @Schedule, Sendgrid takes ~1-4 seconds)
• JDK-8 specific features: Lambdas, parallel streams, Date/Time API
General Principles

• All components used with @Inject
  • bean-discovery-mode="all"
• All named beans are JSF 2.2 @ViewScoped
• All boundaries are only @Stateless, no @Remote or @Local interfaces
• All modules in entity-control-boundary pattern
Entity-Control-Boundary Pattern
Additional Libraries

• PrimeFaces5 and Bootstrap for the responsive portal

• Sendgrid API for sending emails

• Flyway for database migrations

• Misc Apache Commons (FileUtils.writeByteArrayToFile, IOUtils.toByteArray)

• JUnit with Derby for testing
Runtime

- JDK 8
- WildFly 8.1.0 - super, unique product
- MySQL 5.1.73
Tools

• IntelliJ 13.1
• Jenkins
• loader.io for load testing
• Self-hosted Linux server in hetzner.de data center
  • Planning to migrate to OpenShift in 2015
Wishlist for Java EE 8

- Apache Commons Library
- Generics in Batch Processing
- `@Temporal` should work with `LocalDate` and `LocalDateTime`.

I have no great wishes, the technology is perfect. I’m with Java/EE7 very productive, i’m and our customers are with Java ee7 happy. With the technology, really everything is possible.
Acknowledgments

@ChristophGapp

@AdamBien
Resources

• **Java EE 7 tutorial**: [http://docs.oracle.com/javaee/7/tutorial/doc/home.htm](http://docs.oracle.com/javaee/7/tutorial/doc/home.htm)

• **~200 code samples**: [github.com/javaee-samples/javaee7-samples](https://github.com/javaee-samples/javaee7-samples)

• Lots of books