

# EJB 3.0 for EJB3CA 4

Simplify maintenance with migration

Johan Eklund

*johan at primekey dot se*

# Presentation content

- ◆ Motivation for migration
  - ◆ Expected EJB3 4 platform support
  - ◆ Migration overview
  - ◆ Migration details
  - ◆ EJB3 4.1
- 
- ◆ Please interrupt w questions, remarks and/or suggestions

# Background, MSc Thesis Project

- ◆ Workflow evaluation
  - Not covered in this presentation
- ◆ Java framework evaluation
  - EJB 3.0 vs. Spring 2.0
  - Practical evaluation
  - Migration strategy

# Background, EJBICA

- ◆ First EJBICA release in 2001
- ◆ >204 400 lines of real code in 3.9.0!
- ◆ Works with several application servers
- ◆ Works with several databases

# Background, Problems

- ◆ EJB 2.1 is hard to maintain
  - XDoclet tool for generating interfaces was discontinued
  - XDoclet prohibits use of neat Java features
  - Support for different databases requires a lot of manual configuration
  - Support for different application servers requires a lot of manual configuration and extra work

# Framework evaluation, Spring 2.0

- ◆ Based on regular JavaBeans
- ◆ Integration framework
- ◆ Loose coupling with XML "wiring"
- ◆ Tons of configuration
- ◆ Fast moving

# Framework evaluation, EJB 3.0

- ◆ Standard
- ◆ Simplified database access using Java Persistence API (JPA)
- ◆ Configuration by exception w annotations.
- ◆ Loose coupling w dependency injection
- ◆ Less room for vendor lock-in
- ◆ EJB 3.1 around the corner

- ◆ EJB 3.0 is better suited for EJBCA
  - XDoclet JavaDoc-style comments translates directly to EJB 3.0 annotations
  - Easier configuration for a huge project
  - A stable standard is easier to maintain

# Framework evaluation, Test application

- ◆ Test EJB 3.0 support
- ◆ Test JPA providers' support for MySQL, Postgres, Oracle, DB2 and Derby
- ◆ Test JSF config
- ◆ Test Web Service support
- ◆ Test Remote EJB invocation
- ◆ Test workaround for JBoss serialization
- ◆ Uses isolated piece of code from EJBCA

# Framework evaluation, Results

- ◆ Sufficient application servers support
- ◆ Hibernate is the best JPA provider
- ◆ No extra libraries for JSF

	EJB 3.0 core	Working JPA providers	Req ORM XML	JSF w/o libs	WS	EJB remote	EJB 2.1 Serialization
Geronimo 2.1.3	w proprietary config	added Hibernate lib	no	works	works	w proprietary config	
Glassfish v2	works	added Hibernate lib, TopLink	only for Toplink	works	works	works	normal
(Glassfish v3 Prelude)	no, rebuilt for EJB 3.1						
JBoss 4.2.3.GA	not fully supported	Hibernate	no				proprietary
JBoss 5.0.0.GA	works	Hibernate	no	works	works	works	proprietary
(JOnAS 5.1-M2)	works	Hibernate	no	incomplete			
OC4J 10.1.3.4.0	not supported						normal
WebLogic 10.3 Exp	w proprietary config	added Hibernate lib	no	w proprietary config	works	no, proprietary solution	normal
WebSphere 7.0 Exp	not fully supported						normal

# Migration, Preparations

- ◆ Prepare new project layout
  - Done
- ◆ Commons Configuration
  - In progress (EJB3A 3.10.0, ECA-1380)
- ◆ Separate unit tests and system tests
- ◆ Move EJB code from JSP pages (if any)
- ◆ Use JPA in ExtRA client library

# Migration, Overview (1/3)

- ◆ Bransch SVN (1d)
- ◆ Remove XDoclet (3d)
- ◆ Drop Java 1.5 support (3d)
- ◆ New project layout and build (8d)
- ◆ Setup util JUnit tests (2d)
- ◆ Convert EJB interfaces (15d)
- ◆ Convert Entity Beans (20d)

# Migration, Overview (2/3)

- ◆ Convert Session Beans (45d)
- ◆ Timer Service and WS (4d)
- ◆ Rewrite system JUnit tests (3d)
- ◆ Rewrite CLI (3d)

# Migration, Overview (3/3)

- ◆ Update GUI and OCSP (5d)
- ◆ Write upgrade CLI (10d)
- ◆ Convert JDBC to JPA QL (10d)
- ◆ Review documentation (10d)
  
- ◆ System testing (20d+ ?)

# Migration, Post migration tasks

- ◆ Additional (stress-)testing (10d+)
- ◆ Additional improvements (?)
- ◆ Review documentation (10d)
- ◆ Add must have features (?)
  
- ◆ Handle bug reports (20d+ ?)

# EJB 3.0 migration, Branch SVN

- ◆ Branch ejbca-4 is created in SVN
- ◆ Feature freeze during migration
  - Bug fixes could be applied to 3.x branch
  - A new issue should be created for relevant patches that cannot be applied to ejbca-4

# EJB 3.0 migration, Remove XDoclet

- ◆ Generate EJB interfaces and XML files one last time
- ◆ Add to version control
- ◆ Remove ant-scripts for generation and deletion of interfaces
- ◆ Remove XDoclet libraries

# EJB 3.0 migration, Drop Java 1.5 Support

- ◆ Sun's support ends 2009-10-30
- ◆ Replace all 1.5 libraries with 1.6 version
  - BouncyCastle
- ◆ Use Console class for CLI passwords
  - Remove third party Java 1.5 library

# EJB 3.0 migration, New project layout

## ◆ Separate into modules

- Entity beans
- Bean interfaces
- Session beans
- Tools, helpers etc
- Libraries built from source
- One module per WAR

## ◆ New build scripts

# EJB 3.0 migration, Convert Entity Beans (1/2)

- ◆ Move to "entity"-module
- ◆ Convert EB to POJO
  - Implements Serializable and Clonable
  - Add a default constructor
  - XDoclet comments → annotations
  - Abstract getters and setters → implement
  - Annotate CLOB fields w largest (?) size from merge-files
  - Change create methods required by EJB 2.1 to regular constructor

# EJB 3.0 migration, Convert Entity Beans (2/2)

- ◆ XDoclet finder queries → static methods
  - DAO pattern is overkill
  - JPA QL
  - EntityManager method argument
  - Instead of EJB 3.0 named queries
  - Use generics for returned Lists
  - Validate method parameters if possible

# EJB 3.0 migration, Convert EJB Interfaces

- ◆ Common interface `XxSession`
  - Common methods
- ◆ Local interface `XxSessionLocal`
  - Annotated with `@Local`
  - Local-only methods
- ◆ Remote interface `XxSessionRemote`
  - Annotate with `@Remote`
  - Remote-only methods

# EJB 3.0 migration, Convert Session Beans

- ◆ XDoclet comments → annotations
  - Transaction attributes
  - Bean type
- ◆ Manual lookup → @EJB-injection
- ◆ Change database access pattern
  - Use JPA POJOs instead of entity beans
  - Keep value objects for now if used in EJB interfaces

# EJB 3.0 migration, Timer Service and WS

## ◆ Timer Service

- Old implementation can still be used
- Easy to use EJB 3.0 style timer instead

## ◆ Web Services

- Convert to an annotated Session Bean
- Drop unused libraries
- Generate WSDL and client lib without running application server

# EJB 3.0 migration, Rewrite JUnit test and CLI

- ◆ Lookup of remote EJB 3.0 beans instead of EJB 2.1 beans
- ◆ Functionality shouldn't have changed!

# EJB 3.0 migration, Update GUI and OCSP

- ◆ Move EJB 2.1 code from JSP
  - To backing bean
  - Use EJB 3.0 lookup instead
  - Refactor as JSF if major reconstruction is required
- ◆ Update Servlets to use EJB injection

# EJB 3.0 migration, System Testing

- ◆ Unit tests available early on
- ◆ System test on Hudson during conversion later on
- ◆ Test GUI manually
  - Same cases as release testing
  - Try to cover all common use-cases
- ◆ Test other interfaces (WS, CMP, ExtRA)
  - Manually using clients
  - JUnit tests

# EJB 3.0 migration, Upgrade

- ◆ Can we disallow database upgrades in the EJBCA 3.x branch?
  - Require partial upgrade to latest 3.x first
  - Only upgrade code from latest 3.x to 4.0
- ◆ If not
  - Rewrite old upgrade code
  - Write upgrade code from latest 3.x to 4.0

# EJB 3.0 migration, Improve (1/2)

- ◆ Convert JDBC to JPA QL
  - Mixing JDBC and JPA is not a good idea.
  - Mixing with EJB 2.1 Entity Beans isn't.
  - More maintainable code.
- ◆ Use detached JPAs instead of VOs
  - Keep old interfaces and deprecate properly
  - Wait for Web Beans? Same object from database to web?

# EJB 3.0 migration, Improve (2/2)

- ◆ Use stand-alone JPA for ExtRA client library
- ◆ Other necessary core changes?
  - GUI changes for EJBCA 4.1
  - Possible to do more advanced stand-alone upgrade tools with JPA in the future.

# EJB 3.0 migration, Review documentation

- ◆ For both EJB3CA and OCSP
- ◆ [www.ejbca.org](http://www.ejbca.org)
- ◆ [wiki.ejbca.org](http://wiki.ejbca.org)
- ◆ [/howto/](#)
  - Move to generated HTML documentation
- ◆ JavaDoc

# The future of EJB3, EJB 3.1 for EJB3 4.1?

- ◆ Web Beans web framework
  - Subset of Seam framework
  - Possible to run already on supported AS?
- ◆ Cron-like timers
  - Introduce Quartz today?
- ◆ Asynchronous bean methods
  - Useful for tasks started from GUI
- ◆ Simplified WAR configuration

## ◆ EJB 3.0

- Good choice for EJBCA
- Sufficient application server support
- Migration from EJB 2.1 possible
- Requires a lot of work (~ $\frac{1}{2}$  to 1 man-year?)

## ◆ Additional questions or ideas?

- later? → johan at primekey dot se