JUDCon

JBoss Users & Developers Conference

2012:India

Agenda

Migration from Weblogic to JBoss

Vadi Ranganathan
System Architect/Project Manager
Ventera Corp, USA

Vision

•Migrate Business Critical system supporting 1000+ users from WebLogic to Open Source J2ee application server using the latest technologies.

Scope:

•Anything and everything which can be upgraded to the latest version or migrated to the vendor neutral open source technologies.

Constraints/considerations/expectations:

•3 – 6 months timeframe and limited budget.

•Current WebLogic version is no longer supported and was limiting the use of newer technologies.

•No database level changes as that will multiply the work and impact both the timeline and budget.

•Limit the code changes.

•Migration should be transparent to the users & no user interface changes.

Why JBoss

- Open choice, maximum flexibility, and productivity at all levels of your middleware architecture.
- Enterprise-class features, management, and support at a fraction of the price.
- High quality solution with lower Total Cost of Ownership

Scope and Timeline

- Migrate to JBoss 5.1 from Weblogic 8.1
- Upgrade JDK to 1.6 from JDK 1.4
- EJB 2.1
- Struts1.37
- Spring 3.0.5
- Hibernate 3.3.2
- Windows to RHEL 6.0 64-bit Platfom
- fnscore foundation code.
- 3 months of timeline
- Business continuity was vital since the application was in production.

Planning Phase

- Application Code
- Setting up development environment
- Did pick up a small application to migrate to Jboss from Weblogic.
- Production environments.
- Review the build and deployment process.
- Testing the entire application was a challenge.
- Schedule issues.
- Resources.
- Application consolidation.

Planning Phase

- Application Code
  - Code Organization
  - Deployment Descriptors
  - Security
  - JNDI
  - EJB Migration
  - Third Party libraries.
- Resources
  - Existing knowledge of application was vital.
- Production Environment
  - Clustering
  - Application load
  - External dependencies

Planning Phase

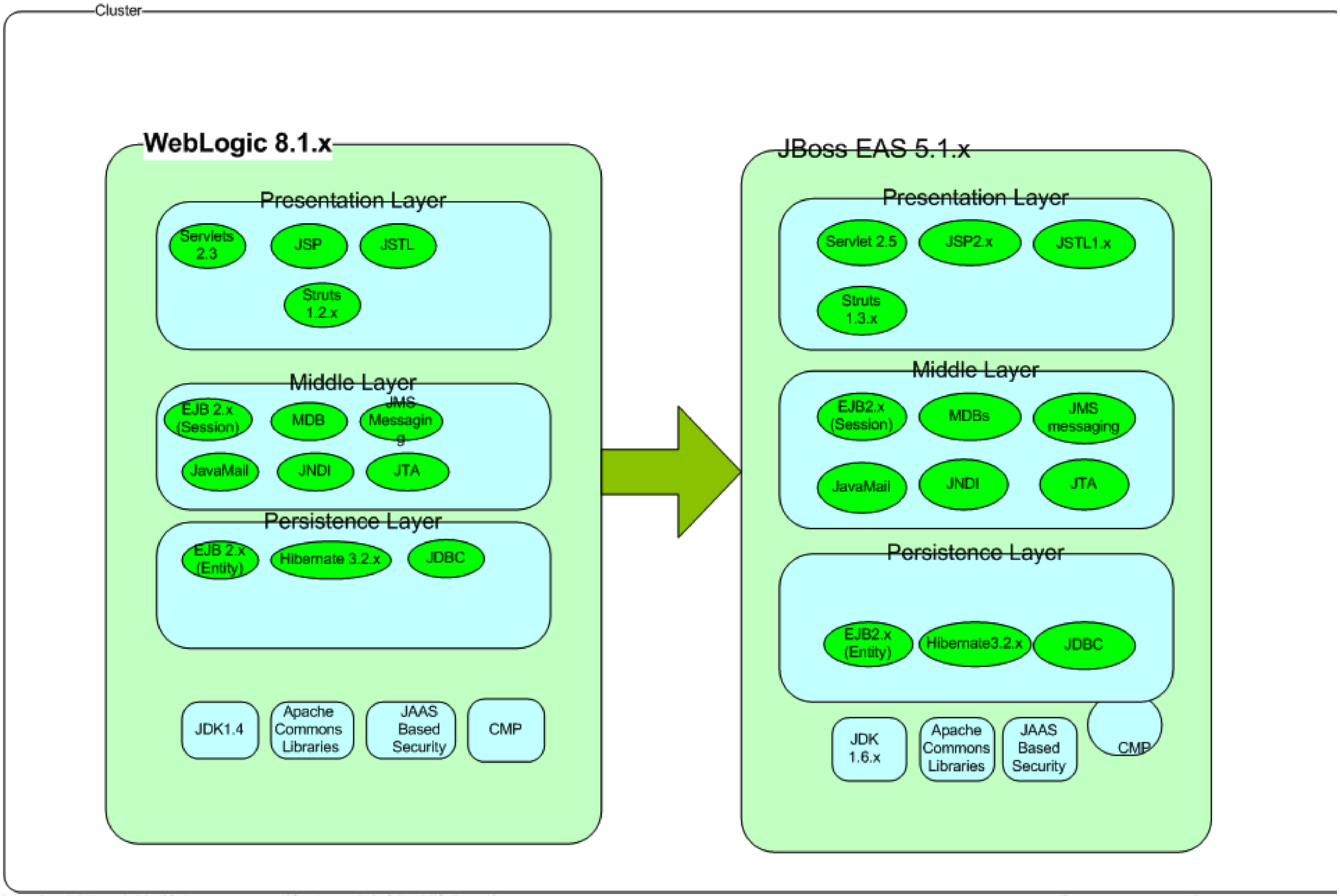
- Testing Practice
  - Coverage provided was considered adequate.
  - Well-defined testing procedure will help during migration.
  - Identified key functionalities
  - To determine pre-existing issues.
- Schedule Issues
  - no set of enterprise application is static, migration plans included changes to existing code base.
- Environments
  - Build new environments Integ, Pre-prod and Production
  - Deployment

Technologies Adopted

- RHEL 6.0
- JBoss clustering
- ANT Build
- RPM deployment
- MS SQL Server 2008 R2
- SSRS Reporting
- Axis 2

Application Server specifics

- Included more than 300 EJBS (SESSION, ENTITY and MDB)
- xdoclet to generate ejb-jar.xml to ejb 2.1 spec to replace wlgen.
- Authentication and Authorization using JAAS:security-jboss-beans.xml
- weblogic timer was replaced using quartz.



Cluster

WebLogic 8.1.x

Presentation Layer

Servlets 2.3

JSP

JSTL

Struts 1.2.x

Middle Layer

EJB 2.x (Session)

MDB

JMS Messagin g

JavaMail

JNDI

JTA

Persistence Layer

EJB 2.x (Entity)

Hibernate 3.2.x

JDBC

JDK1.4

Apache Commons Libraries

JAAS Based Security

CMP

JBoss EAS 5.1.x

Presentation Layer

Servlet 2.5

JSP2.x

JSTL1.x

Struts 1.3.x

Middle Layer

EJB2.x (Session)

MDBs

JMS messaging

JavaMail

JNDI

JTA

Persistence Layer

EJB2.x (Entity)

Hibernate3.2.x

JDBC

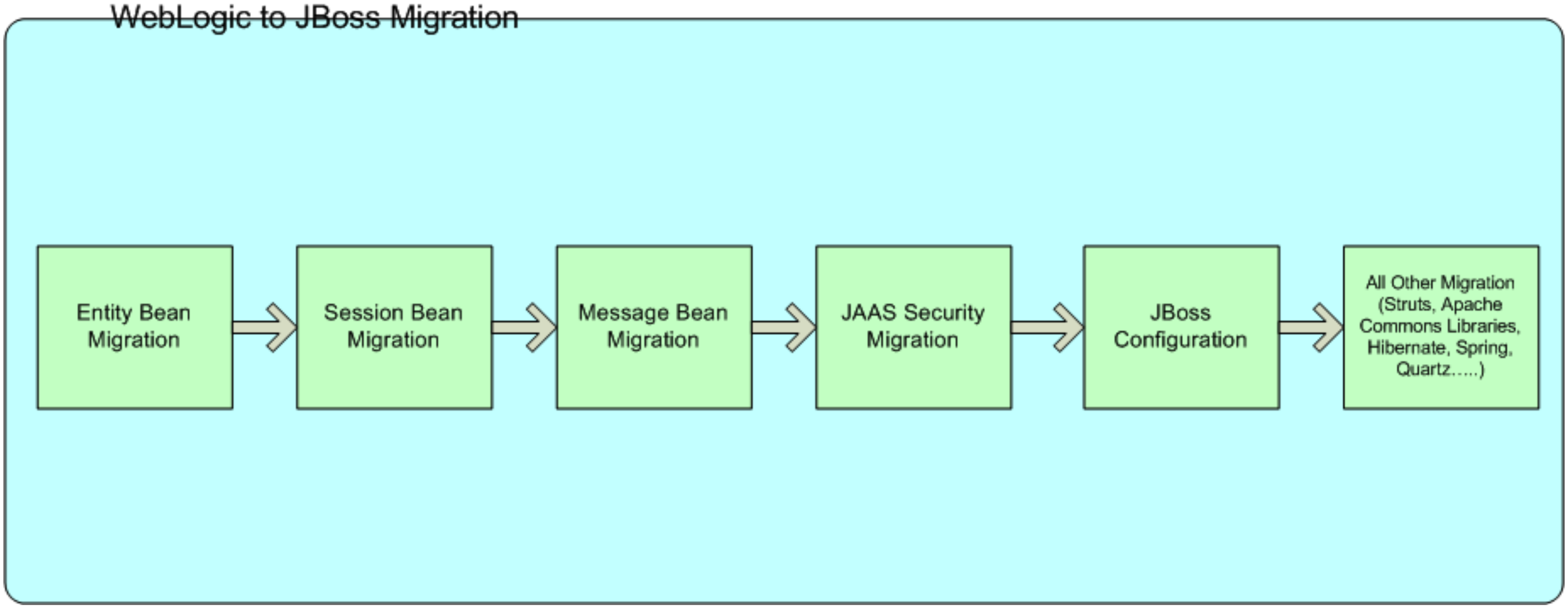
JDK 1.6.x

Apache Commons Libraries

JAAS Based Security

CMP

How did we go about!



WebLogic to JBoss Migration

Entity Bean Migration → Session Bean Migration → Message Bean Migration → JAAS Security Migration → JBoss Configuration → All Other Migration (Struts, Apache Commons Libraries, Hibernate, Spring, Quartz…..)

How did we go about!

- We started with the persistence layer.
  - Entity Beans were migrated first.
  - We have 160+ Entity Beans and over 80% of them have 2nd or 3rd level CMR defined.
  - Entity Beans was using EJBGen annotations for generating the deployment descriptors and EJB interfaces and stub implementations.
  - To migrate the Entity Beans to JBoss, we used XDoclet tool to generate Jboss annotations.

How did we go about!

- We then went on deploying the Entity Beans onto the JBoss.
- Initial errors were resolved by fixing/using the correct XDoclet/JBoss annotations.
- These steps were repeated until all the Entity Beans were deployed successfully.
- The important thing to remember here is that we did not have to make any changes to our existing database or relationships to support this migration, which was one of the key constraints we put in place before we started on migration path.

Concerns/Challenges

- CMRs posed a biggest challenge during this migration of Entity Beans. However using the correct XDoclet/JBoss annotations, we were able to achieve the same CMRs what we had with webLogic.
- Cascade deletion also posed a challenge and required a little bit of code changes.

How did we go about!

- Next we migrated session beans & message beans
  - We had 80+ session beans and 10+ Message beans.
  - These beans required only EJBGen to XDoclet/JBoss annotation changes.
  - We did not make any single line of code change to achieve the session and message bean migration.

How did we go about!

- Next JBoss configration changes
  - Made JBoss configuration changes for data source, JMS queues, JMS data store and transaction support.

Testing

- Clustering including session replication.
- load testing.
- tomcat server.xml configuration with mod_jk.
- security vulnerability scans using IBM App Scan.

Class loading Mechanism

Jboss-classloading.xml:

```
<classloading xmlns="urn:jboss:classloading:1.0"
             name="starsweb.war"
             domain="DefaultDomain"
             parent-domain="Ignored"
            export-all="NON_EMPTY"
             import-all="true">
</classloading>
```

Authentication and Authorization

- SecurityFilter
- SecurityDispatchAction

```
String fullname = request.getParameter("j_username");
String password = request.getParameter("j_password");
//authenticate internal user
WebAuthentication webA = new WebAuthentication();
if(webA.login(username.toString(), password)){
            setLastLoginDate(request);
            return mapping.findForward("gotowarning");
}
```

- EAuthLoginModule extends DatabaseServerLoginModule for SSO

JAAS Programatic Login

- org.jboss.web.tomcat.security.login.WebAuthentication
- org.jboss.security.auth.spi.UsersLoginModule
- org.jboss.security.auth.spi.DatabaseServerLoginModule

security-jboss-beans.xml

```
<authentication>
            <login-module code = "gov.fns.platform.security.auth.EAuthLoginModule"
            flag = "sufficient">
            <module-option name = "dsJndiName">java:CommonSecurityDS</module-option>
            <module-option name="principalsQuery">select eauthentication_id from fns_user where eauthentication_id = ?</module-option>
            <module-option name="rolesQuery">select rsr.security_role_name, 'Roles' from ref_security_role rsr
             inner join user_security_role usr on rsr.security_role_id = usr.security_role_id
             inner join fns_user fu on usr.user_id = fu.user_id
             where fu.eauthentication_id = ?</module-option>
            </login-module>
            <login-module code = "gov.fns.platform.security.auth.NameLoginModule"
            flag = "sufficient">
            <module-option name = "dsJndiName">java:CommonSecurityDS</module-option>
            <module-option name="principalsQuery">select password from fns_user_detail fud where fud.user_name = ?</module-option>
            <module-option name="rolesQuery">select rsr.security_role_name, 'Roles' from ref_security_role rsr
            inner join user_security_role usr on rsr.security_role_id = usr.security_role_id
            inner join fns_user_detail fud on usr.user_id = fud.user_id
            where fud.user_name = ?</module-option>
            <module-option name="unauthenticatedIdentity">nobody</module-option>
            </login-module>
</authentication>
<authorization>
<policy-module code="gov.fns.platform.security.authorization.JBossAuthorizationModule" flag="required">
<module-option
         name="exemptWebResources">/common/*,/images/*,/scripts/*,/css/*,/warning.do,/login.do,/logout.do,/security.do,/secureLogin.do,/welcome.do,/internalLogin.
         do,/eauthLoginInfo.do,/accessViolation.do</module-option>
</policy-module>
 </authorization>
```

web.xml form-based authentication

```
<distributable/>
<context-param>
        <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring-application-context.xml</param-value>
    </context-param>
<filter-mapping>
<filter-name>SecurityFilter</filter-name>
<url-pattern>*.do</url-pattern>
<url-pattern>*.action</url-pattern>
```

web.xml

```
<security-constraint>
<web-resource-collection>
<web-resource-name>Secure Content</web-resource-name>
<url-pattern>*.do</url-pattern>
<url-pattern>*.jsp</url-pattern>
</web-resource-collection>
<auth-constraint>
<role-name>FNS User</role-name>
</auth-constraint>
</security-constraint>

<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/secureLogin.action</form-login-page>
<form-error-page>/common/accessViolation.action</form-error-page>
</form-login-config>
</login-config>

<security-role>
  <description>The role required to access restricted content</description>
  <role-name>FNS User</role-name>
</security-role>
</web-app>
```

jboss-web.xml

```
<?xml version='1.0' encoding='UTF-8' ?>

 <!DOCTYPE jboss-web PUBLIC
   "-//JBoss//DTD Web Application 2.6//EN"
   "http://www.jboss.org/j2ee/dtd/jboss-web_4_0.dtd">

<jboss-web>

  <security-domain flushOnSessionInvalidation="true">java:/jaas/BYPASSED-SECURITY</security-domain>
  <replication-config>
    <replication-trigger>SET_AND_NON_PRIMITIVE_GET</replication-trigger>
    <replication-granularity>SESSION</replication-granularity>
  </replication-config>
</jboss-web>
```

- flushOnSessionInvalidation :caches authentication results to avoid constant access of the security store associated with login modules
- Didn't opt for declarative programming model.

JNDI

```
public Context getInitialContext() throws PlatformException {
Context ctx = null;

try {
            Properties p = new Properties();
            p.put(Context.INITIAL_CONTEXT_FACTORY, "org.jnp.interfaces.NamingContextFactory");
            p.put(Context.URL_PKG_PREFIXES, "jboss.naming:org.jnp.interfaces");
            String bindAddress = System.getProperty("jboss.bind.address");
            p.put(Context.PROVIDER_URL, "jnp://"+bindAddress+":1100/"); // HA-JNDI port.
            ctx = new InitialContext(p);
} catch (NamingException ne) {
           CommonExceptionHandler.handleException("ContextFactoryHAServiceImpl#getInitialContext()#1; " +
           "The initial context could not be instantiated: ",ne);
}

return ctx;
}
```

EJB Session

```
*
 * @ejb.bean type="Stateless" name="ContactManager" display-name="ContactManager" description="This
 *          stateless session bean for ContactManager" view-type="remote"
 *          jndi-name="ejb/ContactManager"
 *          transaction-type="Container"
 * @ejb.home extends="javax.ejb.EJBHome"
 *          remote-class="gov.fns.starsii.business.store.fspEntity.ContactManagerHome"
 *
 * @ejb.interface extends="javax.ejb.EJBObject"
 * remote-class="gov.fns.starsii.business.store.fspEntity.ContactManager"
 * @ejb.transaction type="Required"
 * @ejb.permission role-name="FNS User"
 *
 * @jboss.clustered True
 * @jboss.cluster-config home-policy="org.jboss.ha.framework.interfaces.RoundRobin"
 *                      bean-policy="org.jboss.ha.framework.interfaces.RoundRobin"
 *                      partition-name="${jboss.partition.name:DefaultPartition}"
 */
```

EJB Entity

```
/**
 * @ejb.bean
 *name="LabelSequence"
 *     local-jndi-name="ejb/LabelSequence"
 *     cmp-version="2.x"
 *     primkey-field="labelSequenceID"
 *     schema="LabelSequence"
 * type="CMP"
 *view-type="local"
 *
 * @ejb.persistence
 *table-name="label_sequence"
 *
 * @ejb.interface
 * local-extends="javax.ejb.EJBLocalObject"
 * local-class="gov.fns.starsii.persistence.brsb.printShop.LabelSequence"
 *
 * @ejb.home
 * local-extends="javax.ejb.EJBLocalHome"
 * local-class="gov.fns.starsii.persistence.brsb.printShop.LabelSequenceHome"
 *
```

EJB Entity

```
* @ejb.transaction
 *     type="Required"
 *
 * @ejb.permission
 * role-name="FNS User"
 *
 *
 * @jboss.persistence
 * datasource="java:/STARSDataSource"
 * datasource-mapping="MS SQLSERVER2000"
 *
 * @jboss.unknown-pk
 * class="java.lang.Integer"
 * auto-increment="true"
 *
 * @jboss.entity-command
 * name="mssql-fetch-key"
 *
 */
```

EJB MDB

```
* @ejb.bean
 * name="EmailMessageBean"
 *  destination-type="javax.jms.Queue"
 *  transaction-type=="Container"
 *  destination-jndi-name="jms/PlatformEmailQueue"
 *  impl-class-name="gov.fns.platform.service.messaging.EmailMessageBean"
 *
 * @ejb.transaction
 *  type="NotSupported"
 *
 * @jboss.destination-jndi-name name="jms/PlatformEmailQueue"
 */
```

References

http://docs.redhat.com/docs/en-US/index.html

Thanks

Ventera FNS Team.