
JBoss and Spring Head to Head

By : Yanai Franchi, Senior Software Engineer “Tikal”



Open Source World is on Fire

- ▶ JBoss : Not Just an Application Server – A large community of open source projects.
 - » Hibernate
 - » Seam
 - » JBoss Cache
 - » JBPM
 - » JBoss Rules
 - » ...
- ▶ Spring Framework - leading full-stack Java/JEE application framework
 - » Dependency Injection
 - » AOP
 - » Service abstraction
- ▶ Spring and JBoss folks engage in a public high-pitch discussion

Spring, JBoss and The Elephant

- ▶ Bruce Tate once mentioned *"Don't make me eat the elephant again"*.
- ▶ Should I go for the most trusted IoC container (a.k.a. Spring) and run the risk of locking myself to a concrete implementation OR try to eat the elephant again ?
- ▶ Are Spring trying to trade in their "quasi-EJB3" container with the standards based implementations used by the huge customer base of BEA and WLS ?



Recall our Spring DaoSupport

```
public class EntityDaoImpl extends HibernateDaoSupport
    implements EntityDao{

    ...
    public Entity findEntity(Long id) {
        return (Entity)
            getHibernateTemplate().load(Entity.class, id);
    }
}
```

```
public void testFindEntityUsingInvalidIdentifier() {
    try {
        dao.findEntity(-9999L);
        fail("Should have thrown an Exception here");
    } catch (HibernateObjectRetrievalFailureException e) {
        //It's OK - swallow it.
    }
}
```

Failed

Oops, What Went Wrong Here ?

- ▶ Spring load() javadoc : "Return the **persistent instance** of the given entity class with the given identifier, throwing an exception if not found."
- ▶ Hibernate 3 migration guide : "Hibernate3 now **defaults to lazy="true"** for everything, both classes and collections "
- ▶ Hibernate3 reference guide : "If the class is mapped with a proxy, **load() just returns an un initialized proxy** and does not actually hit the database until you invoke a method of the proxy."
- ▶ Solutions
 - » Change the class lazy load to false
 - » Use the get() method instead of load()

Should we Use Abstraction Frameworks to Maintain Control?

- ▶ Emmanuel (Hibernate) Blog key points:
 - » *We lost the semantic of the load() method*
 - » *You may be waiting for new API integration.*
 - » *HibernateTemplate does NOT hide any boilerplate on Hibernate 3.x.*
 - » *Use JPA to abstract ORM and EJB3 container for Dependency Injection.*
 - » *ORM switch is orders of magnitude less common than using advanced features of an ORM.*

Spring Developers Responds

- ▶ *HibernateTemplate throws DataAccessExceptions from both JDBC ,Hibernate or any data access that contain the same error info*
- ▶ *HibernateTemplate is not an abstraction to shield you from hibernate 3.x.*
- ▶ *As for JPA - Not everybody wants to buy into the EJB3 thing. For many of us using a simple lightweight container to glue POJOs together is fine.*

Spring Developers Responds – Cont'

- ▶ If you don't need this kind of abstraction you can fall back to Hibernate API:

@Repository

```
public class EntityDaoImpl implements EntityDao{
    private SessionFactory sf;
    public void setSf(SessionFactory sf) {
        this.sf = sf;
    }
    ...
    public Entity findEntity(Long id) {
        return (Entity)
            sf.getCurrentSession().load(Entity.class, id);
    }
}
```

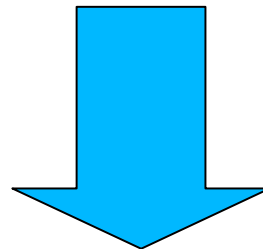
```
<bean class="org.springframework.dao.annotation.
PersistenceExceptionTranslationPostProcessor"/>
```


New JBoss Projects

- ▶ JBoss Microcontainer - JBoss provides its own lightweight IoC container.
 - » Lightweight, IoC container similar in concept to Spring, Pico Container, and Plexus.
 - » Can be used standalone outside the JBoss Application Server - JUnit tests, Swing applications, Tomcat, and more.
- ▶ JBoss EJB3
 - » **JEE Application Server**
 - » **Embeddable EJB 3.0** - standalone applications, JUnit tests, Tomcat, or even other non-JBoss application servers.

R. Johnson Reaction...

- ▶ Rod is understandably laying on the sarcasm:
 - » *“Wow. What a great idea! Imagine if someone had thought of that back in 2002. Imagine if there was a project used very widely that gave you the same model inside and outside the app server across a wide range of functionality. There’s a whole world outside JBoss technology :-)”*
- ▶ Not only has JBoss flipped and embraced the POJO, they are trying to make it look like they came up with the whole idea.



Both the Spring and JBoss camps agree on something – POJO is back !

Pitchfork

- ▶ Developed by Interface21 and BEA Systems.
- ▶ An add-on for the Spring Framework performing EJB 3.0-style interception.
 - » EJB3 beans can be run inside Tomcat as Spring beans.
 - » Allows mixing and matching the powerful Spring programming model with Java EE 5 constructs.
 - » Every EJB can use Spring backed services.
- ▶ *“You don't have to settle for EJB 3's invasive and weak, interceptor-only AOP functionality “ - (Spring dev member).*

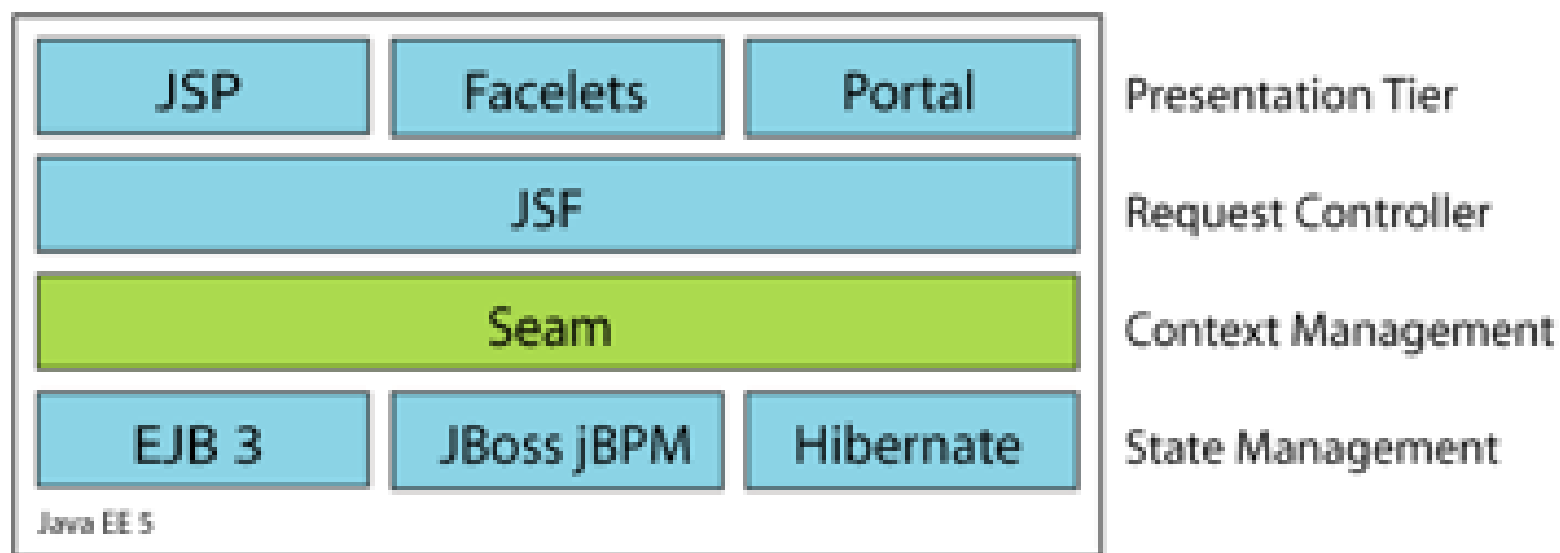
Gavin King on Pitchfork

- ▶ *“This is not even remotely new. You've been able to do exactly this for the past year using the JBoss Embeddable EJB3 container .”*
- ▶ *“...full AOP has basically zero traction in the developer community”*
- ▶ *“...interceptors + annotations - an approach that is much more understandable to non-geniuses. “*
- ▶ *“...To me, AOP looks like totally over hyped, failed technology right now. “*



JBoss Seam

- ▶ Seam is an application framework for Java EE 5:
- ▶ Seam works in any application server that supports EJB 3.0.
- ▶ Can be used in a servlet, or in any JEE AS, by leveraging the new JBoss Embeddable EJB3 container.
- ▶ Can also use JavaBean and Hibernate



Summary

- ▶ If you hear anyone from JBoss say that they are OK with Spring and they don't see it as competition etc. – DON'T BELIEVE IT!
- ▶ Technology needs to carry the endorsement of Sun for it to reap the commercial benefits.
 - » EJB3, JPA
 - » WebBeans
- ▶ Sun Specification is NOT necessarily -
 - » De facto standard
 - » Quality Assured
 - » Going to last long
 - » Superior to other technologies



Q&A



Thank You

yanai@tikalk.com