

EJB 3.0 and Spring

June 2008

presented by: Reza Ghaffaripour

www.RezaGh.com

Topics?

- EJB 2.1 and 3.0
- JPA
- EJB 3.0 and Spring comparison
- JPA and Hibernate comparison

What is EJB?

- An Enterprise Java Bean (EJB) is a server-side component
- **EJB container provides system-level services to beans**
- EJB needs EJB container to run in
- **System-level services are transaction management, security, authorization, pooling, caching, ...**

Types of EJBs?

Session Bean

Performs a task for a client; optionally may implement a web service. Can be **Stateful or Stateless**

Entity Bean

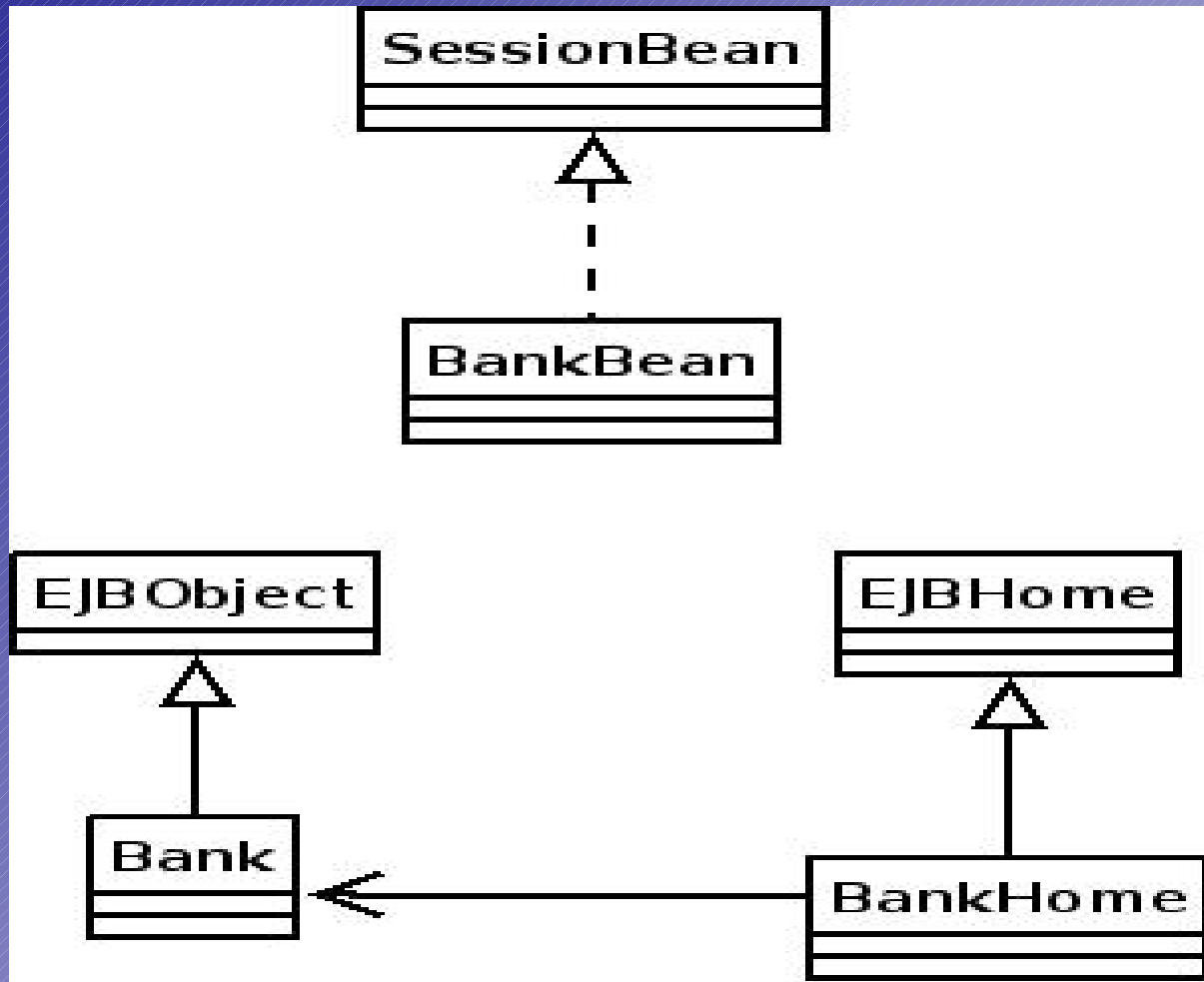
No more in EJB 3.0; instead we have JPA

Message Driven

Acts as a listener for a particular messaging type, such as JMS

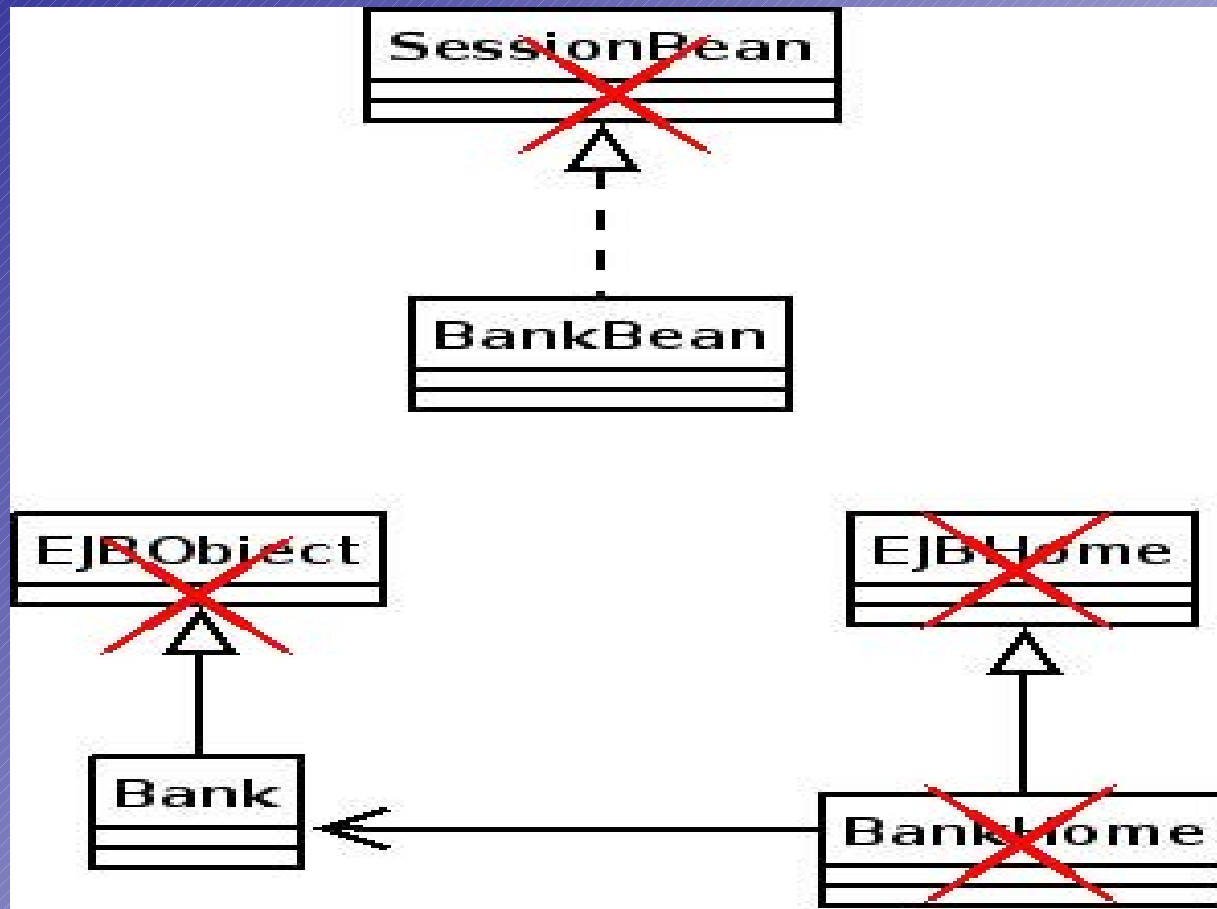
EJB 2.1

- Is part of J2EE 1.4
- Is heavier than EJB 3.0



EJB 3.0

- Is part of Java EE 5 and needs JDK 5 +
- EJB 3.0 makes another attempt to reduce EJB's complexity
- EJB 3.1 is on the way (JSR 318)!



What is new in EJB 3.0 ? Annotations

@Stateless

```
public class SomeBean implements Some{  
    public String getMessage(){  
        return "Test Message";  
    }  
}
```

@Remote

```
public interface Some{  
    public String getMessage();  
}
```

What is new in EJB 3.0 ? Annotations

@Stateless

@Remote

```
public class SomeBean{  
    public String getMessage(){  
        return "Test Message";  
    }  
}
```

But developers can still use XML files optionally

What is new in EJB 3.0 ? Interceptors

- The runtime services like transaction and security are applied to the bean objects at the method's invocation time
- These services are often implemented as the interceptor methods managed by the container

What is (Java Persistence API) JPA ?

- Is part of Java EE 5
- **Provides an O/R mapping like Hibernate**
- Has a query language like HQL called JPQL
- **Uses annotations**
- An entity is a domain object POJO
- **Entities are managed by the entity manager (em)**

JPA

@Entity

@Table(name="BOOK_TABLE")

```
public class Book implements Serializable {  
    private String bookId;  
    private String title;  
    public Book() { }  
    public Book(String bookId, String title, ...) {  
        this.bookId = bookId;  
        this.title = title;  
    }  
}
```

@Id

```
public String getBookId() {  
    return this.bookId;  
}  
  
public void setBookId(String id) {  
    this.bookId=id;  
}  
  
...  
}
```

@PersistenceContext

```
EntityManager em;  
  
...  
Book book = em.find(bookID);  
em.persist(book);  
em.remove(book);  
em.merge(book);  
  
...
```

EJB 3.0 and Spring Comparison Aspects

- **Persistence**
- **Transaction**
- **State Management**
- **Messaging**
- **Remoting**
- **Dependency Injection**
- **Aspects**
- **XML or Annotation**
- **Standard**

EJB 3.0 and Spring: Persistence

- Spring persistence is based on JDBC, Hibernate, JDO, iBatis, and JPA(as of Spring 2.0)
- **EJB by default uses JPA but can use others too**

EJB 3.0 and Spring: Transaction

- Spring supports JTA, JDBC and Hibernate transactions
- EJB 3.0 supports JTA and can support others if based on JTA

EJB 3.0 and Spring: State Management

- EJB provides Stateful session beans
- Spring has Web Flow for Spring MVC
- Spring's other approaches include putting state in the database, the HTTP Session, or in an in-memory cache
- Spring emphasizes on being stateless

EJB 3.0 and Spring: Messaging

- Both Spring and EJB 3.0 enable the sending and receiving of JMS messages
- In terms of implementation MDBs are a JCP standard whereas Spring MDPs (Message Driven POJO) are not

EJB 3.0 and Spring: Remoting

- EJB remoting is built on RMI
- EJB remoting includes support for security and transaction
- Stateless EJBs may also be exposed as Web services
- Spring 2.0 supports various forms of remoting including RMI, Web services, Hessian, Burlap, and HTTP invokers
- Spring remoting does not support out of the box security and transaction propagation

EJB 3.0 and Spring: Dependency Injection

- EJB 3 supports injection of primitive types and Java EE resources from JNDI
- Spring supports injection of anything

EJB 3.0 and Spring: Aspects

- EJB 3 supports interceptors
- Spring is tightly integrated with AspectJ with all of its features

EJB 3.0 and Spring: XML Versus Annotation

- XML files can express complex relationships
- Annotations are simple and concise
- **Both Spring and EJB support XML and annotation**

EJB 3.0 and Spring: Standard

- Spring is **not** a Java standard
- **If standardization is important to you then consider EJB 3.0**
- EJB has wider vendor support

When to Use Spring Alone

- **Your application requires fine-grained control, a lot of configuration and gluing things together**
- You need to build your own stack with different frameworks
- **You need advanced AOP features**
- You need faster development process
- **You don't need appserver**
- ...

When to Use EJB Alone

- You need appserver
- **Your application is very stateful**
- Standardization is an important consideration
- **Application should be scalable out of the box**
- ...

EJB 3.0 and Spring Together!

They are not enemies! They can work together!

Spring in EJB scenario

- Can be used as the glue in all tiers
- Support for other persistence methods

EJB in Spring scenario

- Immediate use of appserver facilities (load balancing, fail over,...)

EJB 3.0 and Spring in the same Stack

Presentation Tier	JSP Tags	Spring Tags
Controller	Servlet	Spring Controller
Delegate\Facade	EJB	EJB
Business Tier	Spring POJO	**
Integration Tier	*	*

* JPA or Spring with Hibernate/JDBC/...

** Spring POJO or EJB

JPA and Hibernate

- JPA is (by reputation at least) slow
- **JPA have to reside within a JEE app server**
- You can use Hibernate without a JEE container
- **JPA is a standard; Hibernate is not**
- Gavin King is an active member of JPA and EJB 3.0 spec!

The End?

Questions?