

# Transitioning to Seam

---

Brian Leonard

Angela Caicedo

Sun Microsystems, Inc.

# Your Speakers

---



# Agenda

---

- Introduction to Java EE 5
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and Resources

# Agenda

---

- Introduction to Java EE
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and Resources

# Java EE 5: High Level Features

---

- EJB 3.0
  - ✓ Java Persistence API (JPA)
- Dependency Injection
- Simplified Web Services Support
- JavaServer Faces (JSF)
- JSP Standard Tag Library (JSTL)
- Fully J2EE 1.4 Compatible

# Agenda

---

- Introduction to Java EE 5
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and Resources

# NetBeans 5.5: Java EE 5 IDE

---

- Java EE 5 end to end development
  - ✓ EJB 3.0
  - ✓ Java Persistence API (JPA)
  - ✓ Web Tier
  - ✓ Web Services
- Same Ease of Use as NetBeans 5.0
  - ✓ Ant based project system
  - ✓ SJS AS, JBoss, WebLogic, Tomcat
  - ✓ Java EE Application Profiler

# NetBeans 5.5 JPA

---

- Wizards
  - ✓ Entity classes, Persistence Units
  - ✓ Entity classes from database, JSF pages and session facade from entity classes
- Editor improvements (editor is the mapping tool)
  - ✓ Annotations code completion, code hints, inline verifier, editor popup actions (eg: use EntityManager)
- Visual editor for persistence.xml
- Standalone Persistence runtime from GlassFish



# NetBeans 5.5 Web Tier

---

- JSF 1.2
- Editor
  - ✓ JSP/JSF Expression Language Code Completion
  - ✓ Visual security configuration
  - ✓ Palette, hyperlinks
- Web Services and JPA support
- CRUD application generation
- HTTP monitor

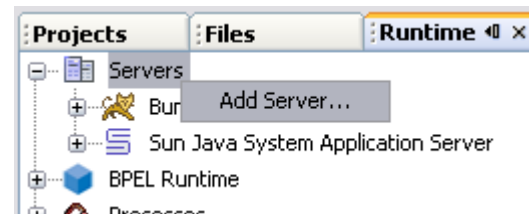
# NetBeans 5.5 Web Services

---

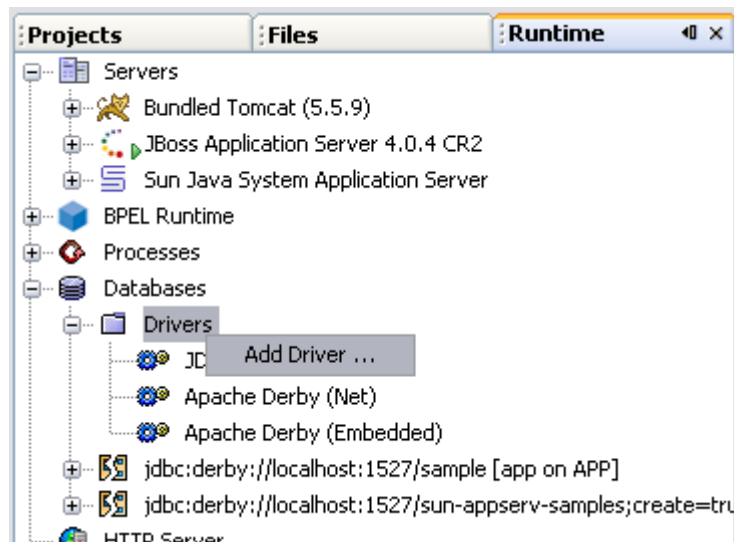
- JAX-WS 2.0, JAXB 2.0
- Wizards (in Java SE, Web, EJB projects)
  - ✓ WS, Clients, Handlers, Operations
  - ✓ Start from annotated Java classes
  - ✓ Start from WSDL and XML schema
- Editor improvements
  - ✓ Using WS Clients from another project
  - ✓ Drag & Drop for Web Service Operations
  - ✓ Annotations code completion
- GlassFish Tester capability integration
- Asynchronous Web Service Invocation

# Setting up JBoss in NetBeans

## 1. Add the JBoss Server



## 2. Set Up the HSQLDB Database



# Creating a Java EE 5 Application

---

1. Create a Web Application Project
2. Create the Persistence Unit
3. Create the Entity Classes
4. Create the JSF Pages

You're Done!

# DEMO:CRUD Application

## Listing Cds

[New Cd](#)

[Back to index](#)

Item 1..8 of 15 [Remaining 7](#)

CdId	Title	Author	YearCreated	Rating	Location	
2	Have a Drink on Me	AC/DC	1987	5	Taqueria el Balazo	<a href="#">Destroy</a> <a href="#">Edit</a>
3	You Shook Me All Night Long	AC/DC	1980	8	Limon Restaurant	<a href="#">Destroy</a> <a href="#">Edit</a>
4	Back in Black	AC/DC	1980	9	Home	<a href="#">Destroy</a> <a href="#">Edit</a>
5	Let Me Put My Love into You	AC/DC	1980	9	Home	<a href="#">Destroy</a> <a href="#">Edit</a>
6	Givin the Dog a Bone	AC/DC	1980	6	Taqueria el Balazo	<a href="#">Destroy</a> <a href="#">Edit</a>
7	What Do You Do for Money Honey	AC/DC	1980	5	Office	<a href="#">Destroy</a> <a href="#">Edit</a>
8	Shoot to Thrill	AC/DC	1980	8	Limon Restaurant	<a href="#">Destroy</a> <a href="#">Edit</a>
9	Battery	Metalica	1986	9	Cha Cha Cha bar	<a href="#">Destroy</a> <a href="#">Edit</a>

## New cd

CdId:

Title:

Author:

YearCreated:

Rating:

Location:

[Create](#)

[Show All Cd](#)

[Back to index](#)

# Agenda

---

- Introduction to Java EE 5
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and resources

# JBoss Seam

---

- RAD programming model for data-driven applications without sacrificing the full power of Java EE 5
- Framework for integrating JSF and EJB3 component models.
  - ✓ Bridge web-tier and EJB tier session contexts
  - ✓ Enable EJB 3.0 components to be used as JSF managed beans.
- Prototype for JSR 299: Web Beans

# JBoss Seam

---

- Some Key Concepts:
  - ✓ Eliminate the ManagedBean – bind directly to our entity and action classes.
  - ✓ Enhanced context model
    - Conversation
    - Business process
  - ✓ Depend less on xml (faces-config) – use annotations instead
  - ✓ *Bijection* – for stateful components - dynamic, contextual, bidirectional
  - ✓ Constraints specified on the model, not in the view.



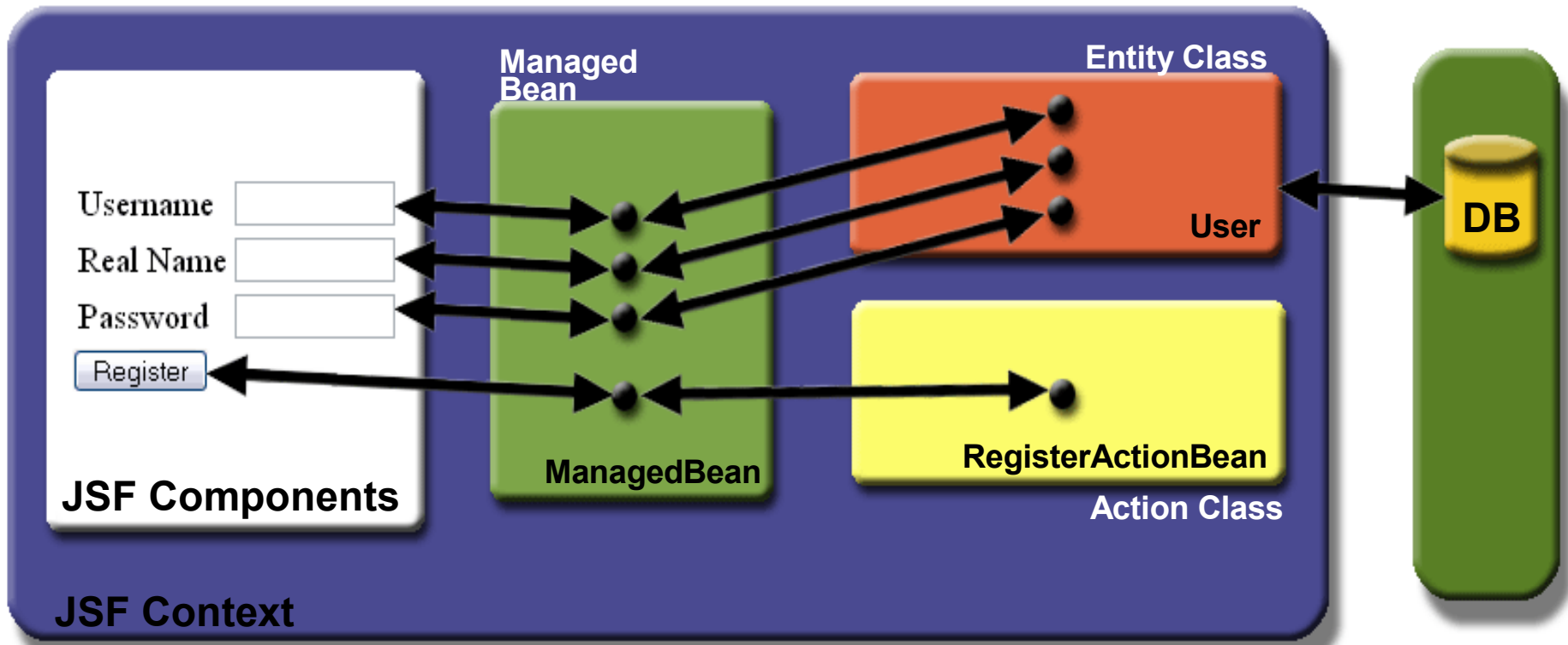
# Agenda

---

- Introduction to Java EE 5
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and resources

# Standard JSF / EJB 3.0 Application

## Registration



# register.jsp: JSF In Action

```
1. ...
2. <td>Username</td>
3. <td><h:inputText id="userName"
4.         value="#{user.username}"
5.         required="true">
6.         <f:validateLength minimum="5"
7.             maximum="15"/>
8.     ...
9. <td>Real Name</td>
10. <td><h:inputText id="name"
11.        value="#{user.name}"
12.        required="true"/></td>
13. ...
14. <td>Password</td>
15. <td><h:inputSecret id="password"
16.        value="#{user.password}"
17.        required="true" maxlength="15">
18.        <f:validateLength minimum="5"
19.            maximum="15"/>
20. ...
21. <h:commandButton id="registerCommand"
22.        type="submit" value="Register"
23.        action="#{user.register}"/>
24. ....
```

Register New User - Mozilla Fire

File Edit View Go Bookmarks T

Username

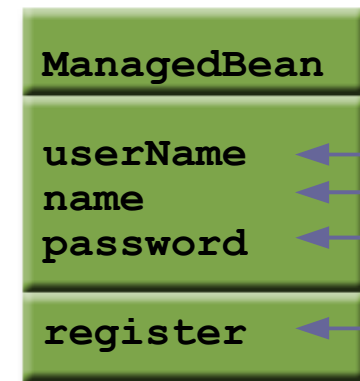
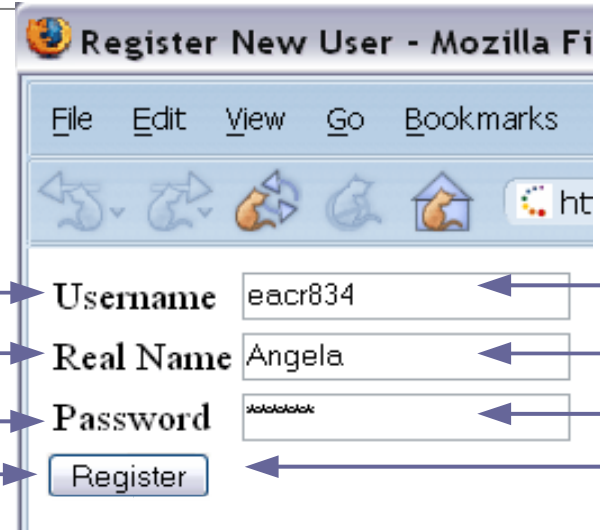
Real Name

Password

Register

# register.jsp & BackingBean

```
1. ...
2. <td>Username</td>
3. <td><h:inputText
4.     id="userName"
5.     value="#{user.username}"
6. ...
7. <td>Real Name</td>
8. <td><h:inputText
9.     id="name"
10.    value="#{user.name}"
11. ...
12. <td>Password</td>
13. <td><h:inputSecret
14.     id="password"
15.     value="#{user.password}"
16. ...
17. <h:commandButton
18.     id="registerCommand"
19.     type="submit" value="Register"
20.     action="#{user.register}"/>
21. ...
```



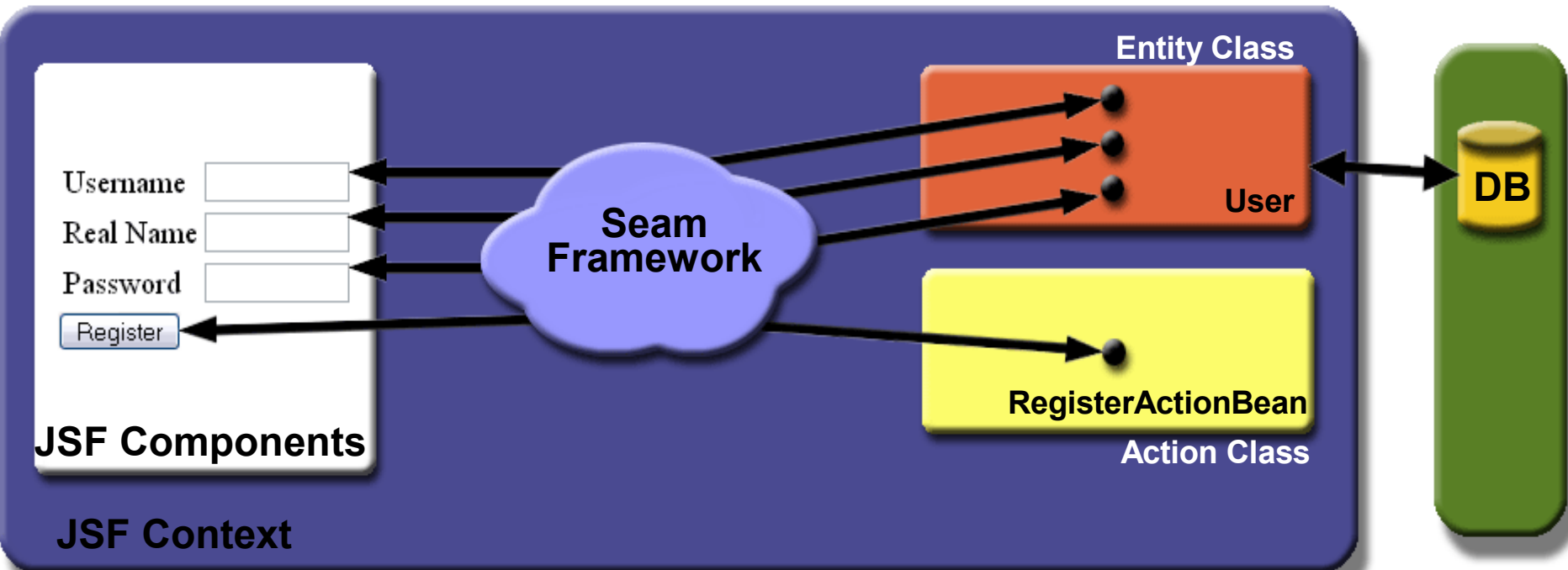
# Manage Beans Configuration

---

## faces-config.xml

```
1. ...
2.   <managed-bean>
3.       <managed-bean-name>user</managed-bean-name>
4.       <managed-bean-class>
5.           org.examples.jsf.ManagedBean
6.       </managed-bean-class>
7.       <managed-bean-scope>request</managed-bean-scope>
8.   </managed-bean>
9. ...
```

# Seam Registration Application

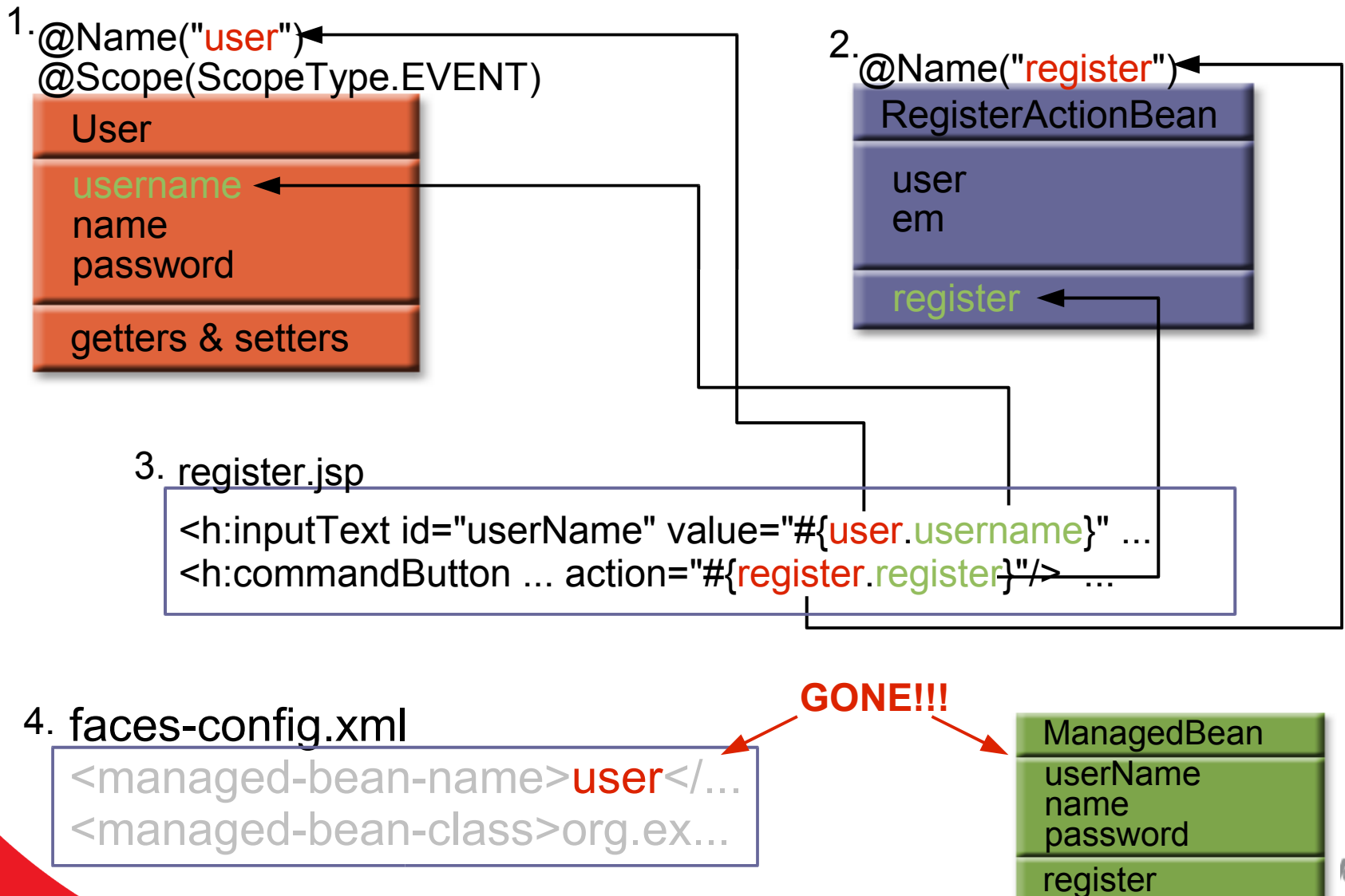


# Integrating The Seam Framework

## Additions...

- EJB Module (jar)
  - ✓ seam.properties
- Web Module (war)
  - ✓ faces-config.xml
    - SeamPhaseListener
  - ✓ web.xml
    - IndiPattern
    - SeamListener

# Eliminating Your ManagedBean





# User.java (1 of 2)

---

```
1. @Entity
2. @Name ("user") → Component name for Seam
3. @Scope (ScopeType.Event) → Default component context
4. @Table (name="users")
5. public class User implements Serializable{
6.
7.     private String username;
8.     private String password;
9.     private String name;
10.
11.     public User(String name, String password,
12.                 String username) {
13.         this.name = name;
14.         this.password = password;
15.         this.username = username;
16.     } ...
```

# User.java (2 of 2)

---

```
1.     public User() {}
2.
3.     @Length(min=5, max=15)
4.     public String getPassword() {
5.         return password;
6.     }
7.
8.     public String getName() {
9.         return name;
10.    }
11.
12.    @Id @Length(min=5, max=15)
13.    public String getUsername() {
14.        return username;
15.    }
16.
```

Hibernate  
Validator  
framework



# RegisterAction.java

---

```
1. @Stateless
2. @Name("register")
3. public class RegisterAction implements Register{
4.     @In ←────────────────────────────────── Attr. Injected by Seam
5.     @Valid ←──────────────────────────────── Hibernate validator
6.     private User user;
7.     @PersistenceContext
8.     private EntityManager em;
9.
10.    public String register() {           (6)
11.        List existing = em.createQuery("select username
12.            from User where username=:username")
13.            .setParameter("username", user.getUsername())
14.            .getResultList();
15.        if (existing.size() == 0) {
16.            em.persist(user);
17.            return "success";
18.        } else {
19.            FacesMessages.getInstance().add("User
20.                #{user.username} already exists"); (8)
21.            return null;
22.        }
    }
```

# Agenda

---

- Introduction to Java EE 5
- Creating a standard Java EE 5 Application
  - ✓ JSF
  - ✓ EJBs
  - ✓ Persistence
- Introduction to Seam
- Refactor to use the Seam Framework
- Summary and Resources

# Summary

---

- Hopefully you've learned how to start using the Seam framework in your existing JSF / EJB 3.0 applications.
- There's much more to Seam, I've just touched the surface.

# Resources

---

- Java EE
  - ✓ <http://java.sun.com/javaee>
  - ✓ <http://glassfish.dev.java.net>
- NetBeans IDE
  - ✓ <http://www.netbeans.org>
- Everything presented here today is documented on my blog on java.net
  - ✓ <http://weblogs.java.net/blog/bleonard>

# Sun Developer Network

---

- Free resources for all developers
  - Tools (Creator, Java Studio Enterprise Edition)
  - Forums
  - FAQs
  - Early Access
  - Newsletters
- Join Today!
  - <http://developers.sun.com>



## Get Connected

Sun Developer Network connects you to what you need, when you need it.

[Join Now >>](#)

# Questions & Answers

---





# Thank You

---

Brian Leonard  
Angela Caicedo  
Sun Microsystems, Inc.

