

JUDCon

JBoss Users & Developers Conference

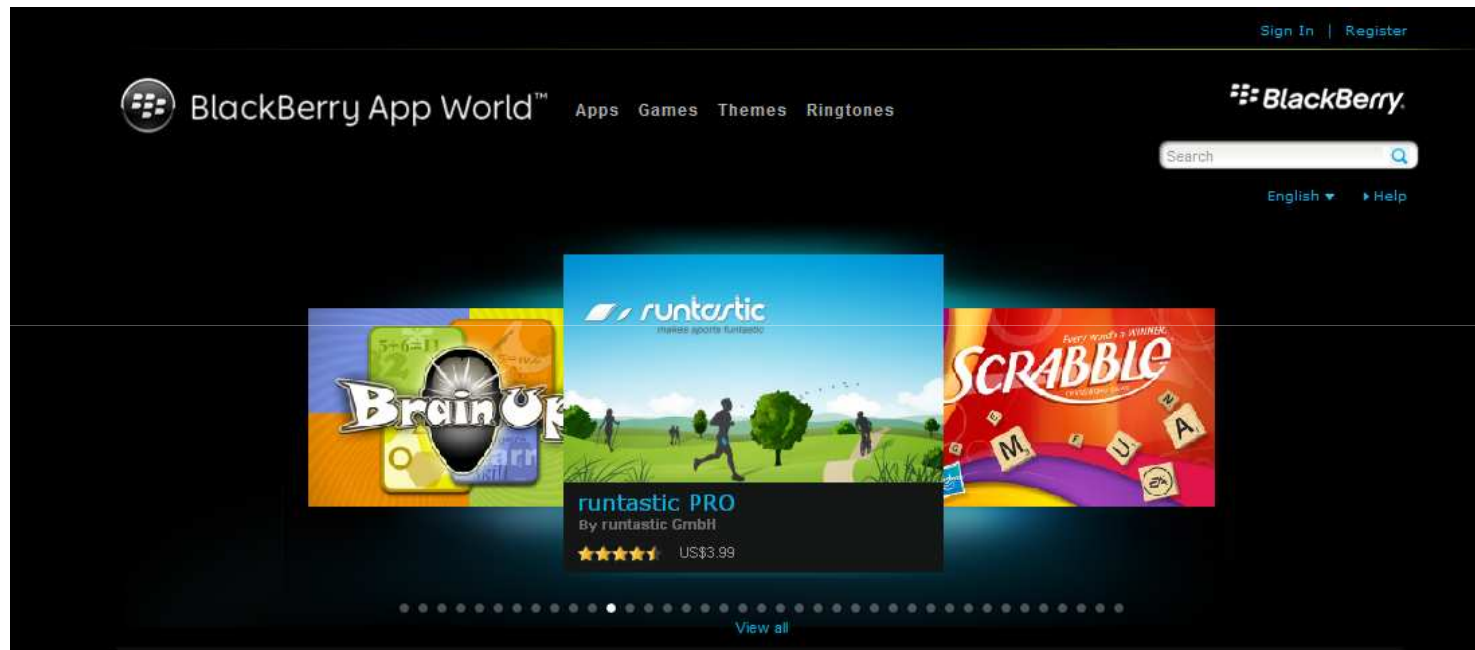
2012: Boston

Seam 3 Migration Planning To Production

- Trevor Sweeney, Team Lead App World™ Webstore Development
- trsweeney@rim.com

- What is the App World™ Webstore?
- How we began migration
- Code changes
- Common Errors
- Moving to production

What is the BlackBerry App World™ Webstore?



Top Free

1 - 5 of 25 | [View all](#)



Where we are today

- JBoss AS 7.0.1
- Seam 3.1.0
- Hibernate 4.0.0
- JSF 2.0
- Richfaces 4.1.0

Where were we last year

- JBoss AS 5.1.0
- Hibernate 3.5.1
- Seam 2.2.1
- JSF 1.0
- Richfaces 3.3.1

Why Migrate?

Why Migrate?

- JSF/Richfaces AJAX issue in IE9

Why Migrate?

- JSF/Richfaces AJAX issue in IE9
- Codebase outside of upgrade path

Why Migrate?

- JSF/Richfaces AJAX issue in IE9
- Codebase outside of upgrade path
- Community development moving to new technology

Proposed Solutions

Proposed Solutions

- Use AJAX outside of framework

Proposed Solutions

- Use AJAX outside of framework
- Maintain custom framework

Proposed Solutions

- Use AJAX outside of framework
- Maintain custom framework
- Add new features agnostic to framework

Proposed Solutions

- Use AJAX outside of framework
- Maintain custom framework
- Add new features agnostic to framework
- Migrate to Jee6 environment

To migrate or not to migrate.

- Risks
 - Unable to estimate size and scope of changes
 - Lack of knowledge/training
- Benefits
 - Upgrades and features available without development work
 - Bug fixes provided by community

How we began

How we began

- Seam 2 Structure
 - Resources
 - Meta-inf
 - Web-inf
 - Src
 - View
 - Build.xml

How we began

- Seam 2 Structure
 - Resources
 - Meta-inf
 - Web-inf
 - Src
 - View
 - Build.xml
- Seam 3 Structure
 - Java
 - Resources
 - Meta-inf
 - Webapp
 - Meta-inf
 - Web-inf
 - Build.pom

Simple Code Changes

Simple Code Changes

- @In vs @Inject

Simple Code Changes

- @In vs @Inject
- @Factory vs @Produces

Simple Code Changes

- @In vs @Inject
- @Factory vs @Produces
- @Name vs @Named

Simple Code Changes

- @Out
 - No longer needed
 - When using @Inject a direct link to the context is obtained
 - Changing the injected instance is immediately reflected in the context

Scopes

- Seam 2 to CDI Scope Conversion
 - @Event, @Page to @RequestScoped
 - @Conversation to @ConversationScoped
 - @Session to @SessionScoped
 - @Application to @ApplicationScoped
- Default Scope in Seam 2 should not map to @Dependant scope in CDI

Dependant Scope

Dependant Scope

```
@Conversation
public class ConvBean{
    @In ReqBean req;
    @In Worker worker;
}
```

```
@Event
public class ReqBean{
    @In Worker worker;
}
```

```
public class Worker{
    ...
}
```

Dependant Scope

```
@Conversation
public class ConvBean{
    @In ReqBean req;
    @In Worker worker;
}
```

```
@Event
public class ReqBean{
    @In Worker worker;
}
```

```
public class Worker{
    ...
}
```

```
@ConversationScoped
public class ConvBean{
    @Inject ReqBean reqBean;
    @Inject Worker worker;
}
```

```
@RequestScoped
public class ReqBean{
    @Inject Worker worker;
}
```

```
public class Worker{
    ...
}
```

Conversations

Conversations

- Conversation management

Conversations

- Conversation management
- Working without nested conversations

Conversations

- Conversation management
- Working without nested conversations
- Understanding the contexts

Conversation Management

Conversation Management

```
@Name("convScope")
@Conversation
public class ConvBean {

    @Begin
    public void beginConversation()
    {
        ...
    }

    @End
    public void endConversation()
    {
        ...
    }
}
```

Conversation Management

```
@Name("convScope")
@Conversation
public class ConvBean {

    @Begin
    public void beginConversation()
    {
        ...
    }

    @End
    public void endConversation()
    {
        ...
    }
}
```

```
@Named("convScope")
@ConversationScoped
public class ConvBean implements Serializable
{
    @Inject
    private Conversation conversation;

    public void beginConversation()
    {
        if (conversation.isTransient())
        {
            conversation.begin();
        }
    }

    public void endConversation()
    {
        if (!conversation.isTransient())
        {
            conversation.end();
        }
    }
}
```

Working without nested conversations

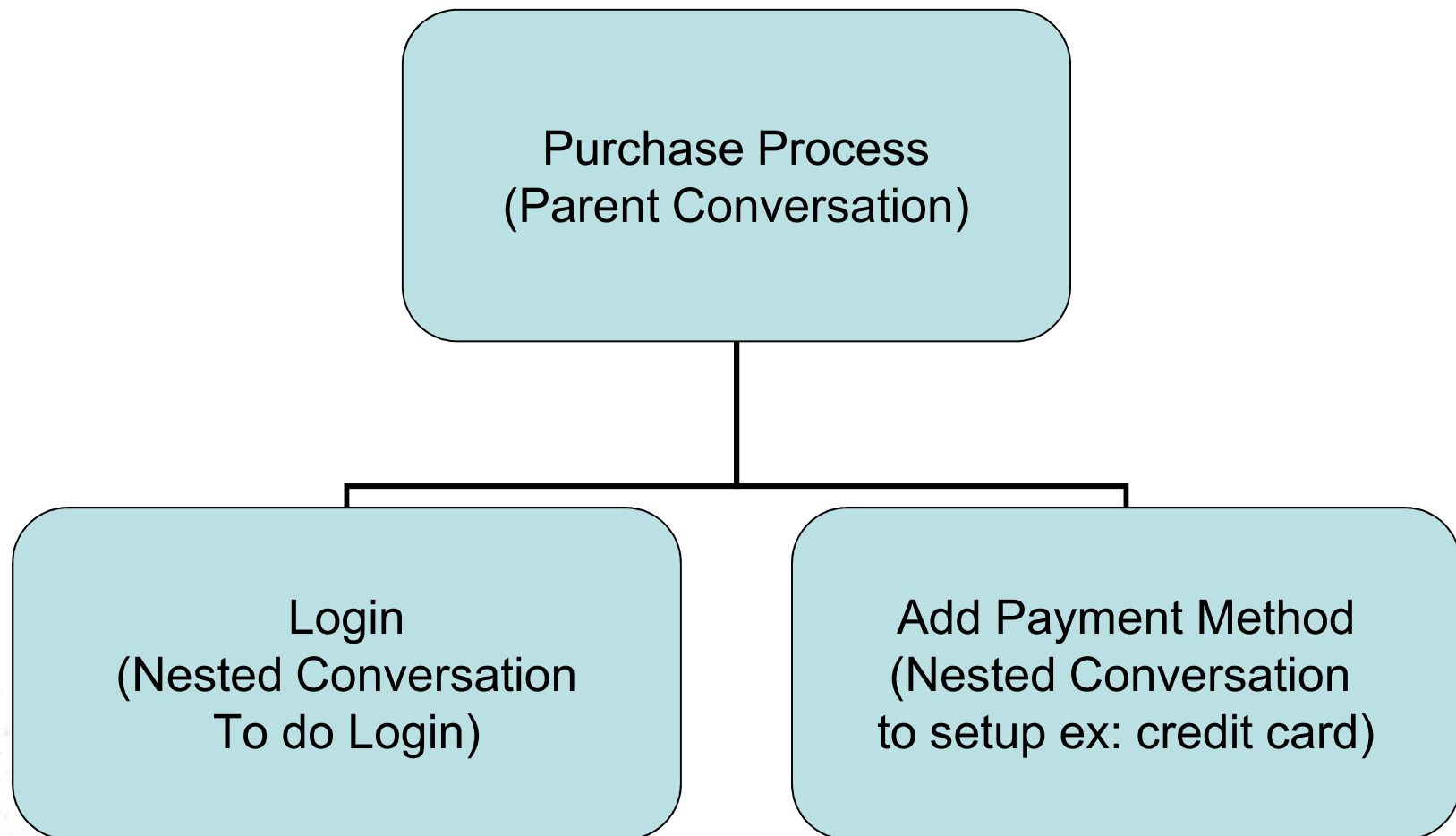
Working without nested conversations

- Nested conversations provided easy method of creating different work flows

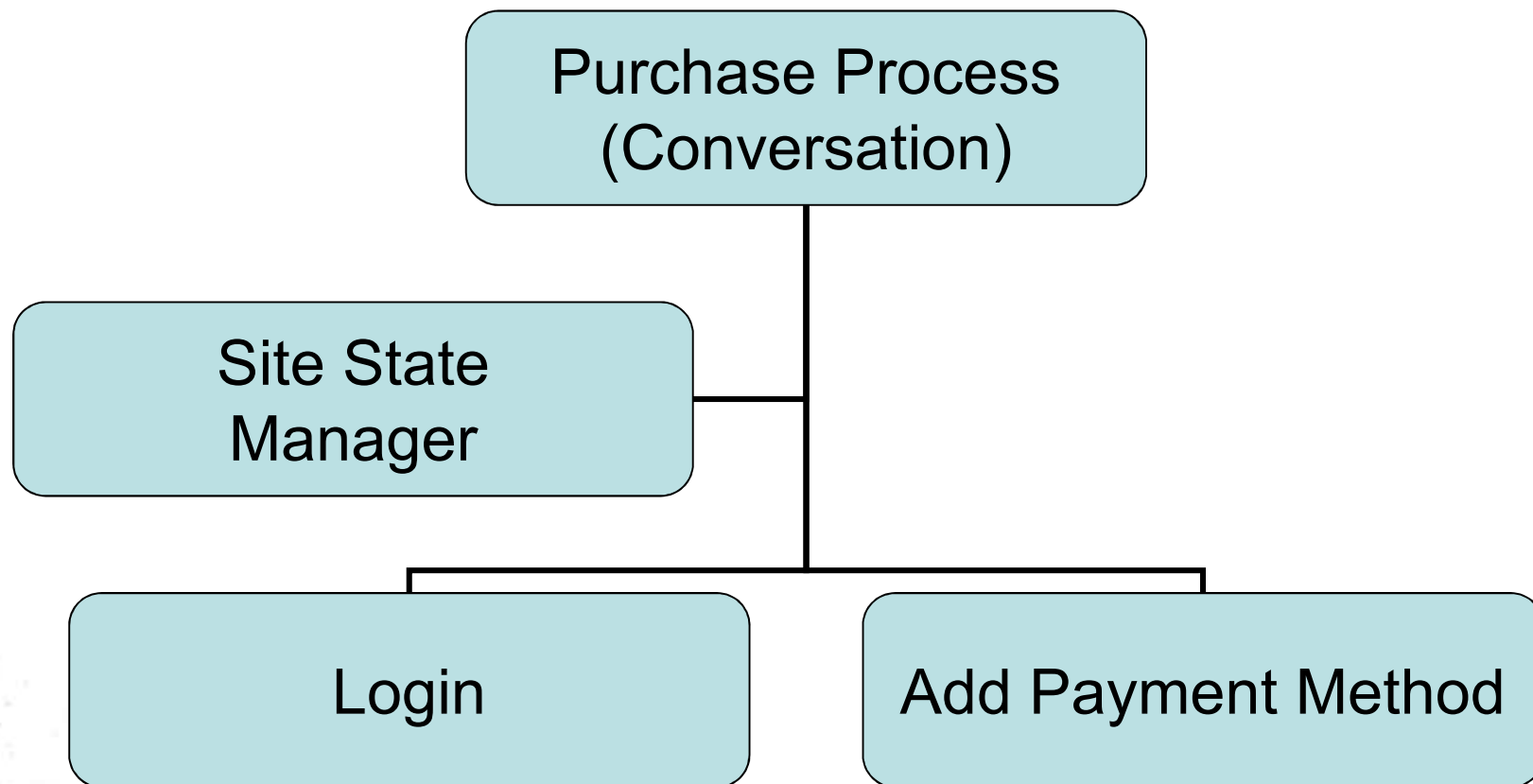
Working without nested conversations

- Nested conversations provided easy method of creating different work flows
- Re-architecture was required during migration

Webstore Purchase Process seam 2.2.1



Webstore Purchase Process Seam 3/CDI



Understanding The Context

Understanding The Context

```
@Conversation  
public class ConvBean{  
    @In ReqBean req;  
}
```

```
@Event  
public class ReqBean{  
    int someImportantState  
}
```

Understanding The Context

```
@Conversation
public class ConvBean{
    @In ReqBean req;
}
```

```
@Event
public class ReqBean{
    int someImportantState
}
```

```
@ConversationScoped
public class ConvBean{
    @Inject ReqBean reqBean;
}
```

```
@RequestScoped
public class ReqBean{
    int someImportantState;
}
```

JBoss EL Parser Changes

- Minor annoyances caused big changes

JBoss EL Parser Changes

- Minor annoyances caused big changes
 - Bean methods must begin with get/set

`#{list.size}`

`#{bool.isSet}`

JBoss EL Parser Changes

- Minor annoyances caused big changes
 - Bean methods must begin with get/set
 - `#{list.size}`
 - `#{bool.isSet}`
 - *Var args methods not supported*
 - `#{bean.setParams(x, y) }`
 - `#{bean.setParams(x, y, z) }`

Servlet Vs CDI Filters

Servlet Vs CDI Filters

- Servlet Filters
 - Loaded using web-fragment.xml
 - Outside of CDI

Servlet Vs CDI Filters

- Servlet Filters
 - Loaded using web-fragment.xml
 - Outside of CDI
- CDI Filters
 - Observer\Observable interface
 - Allows access to contexts

Servlet Vs CDI Filters

- Webstore usage of Filters

Servlet Vs CDI Filters

- Webstore usage of Filters
 - Setting cache control headers

Servlet Vs CDI Filters

- Webstore usage of Filters
 - Setting cache control headers
 - Removing JSessionID param on outbound links
 - Only way to override the HttpServletResponseWrapper to remove the JSessionID during the encodeURL method

Restricting Access To Pages

- Seam 2
 - pages.xml
 - `<page view-id="/someid.xhtml" login-required="true">`

Restricting Access To Pages

- Seam 3/CDI

```
@ViewConfig
public interface Pages {
    public static enum Pages1 {
        @ViewPattern("/someid.xhtml")
        @SOMEROLE
        SOMEPAGE
    }
}
```

Restricting Access To Pages

- Seam 3/CDI

```
@Secures @SOMEROLE
```

```
public boolean isUserAuth(Identity identity){
```

```
    if (identity.isLoggedIn() && identity.hasRole("SOMEROLE", "users",  
        "webstore") && ctx != null){
```

```
        return true;
```

```
    }else{
```

```
        return false
```

```
    }
```

Restricting Access To Pages

- Seam 3/CDI

```
private void authenticationPass(String username){  
    setStatus(AuthenticationStatus.SUCCESS);  
    identity.addRole("SOMEROLE", "users", "webstore");  
    setUser(new SimpleUser(username));  
}
```


Common Errors

- WELD 1409
 - Ambiguous dependencies for type
- WELD 1408
 - Unsatisfied dependencies

Moving To Production

Moving To Production

- Test, Test, Test
 - This is essentially a new product

Moving To Production

- Test, Test, Test
 - This is essentially a new product
- Higher than expected bug rate
 - Higher than expected churn was reflected in bug rate

Moving To Production

- Test, Test, Test
 - This is essentially a new product
- Higher than expected bug rate
 - Higher than expected churn was reflected in bug rate
- Unexpected performance gains!

Questions?