

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

Eye on CDI:

Weld, Seam 3 and Beyond

Bill Bendrot

GLS JBoss Curriculum Manager, Red Hat

June 2010

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Objectives

- Long-standing Challenges of JEE
- We have a solution, but....
- What is CDI?
- Weld
- Seam 3

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Long Standing Challenges of JEE

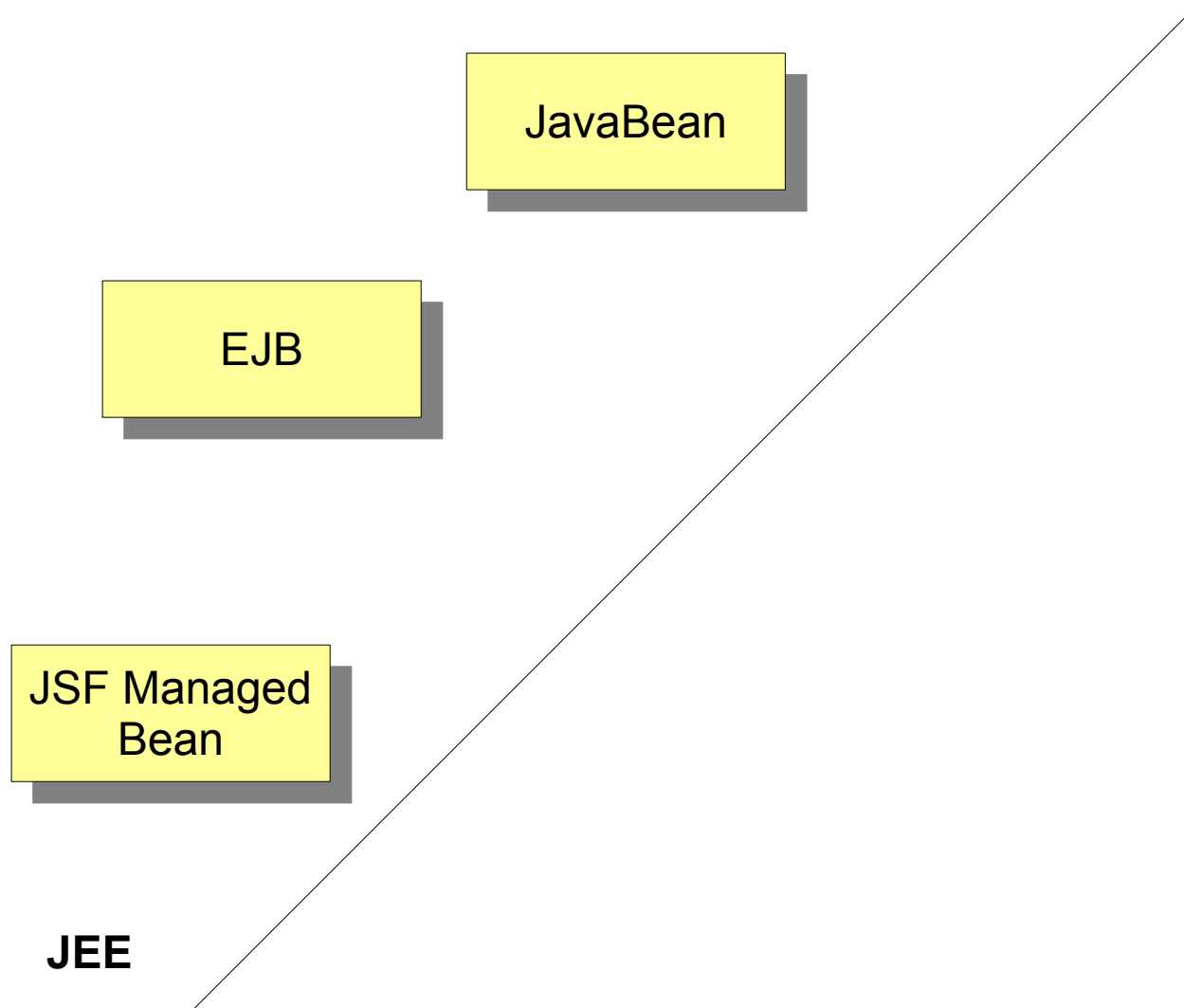
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



So many beans, so little common ground....



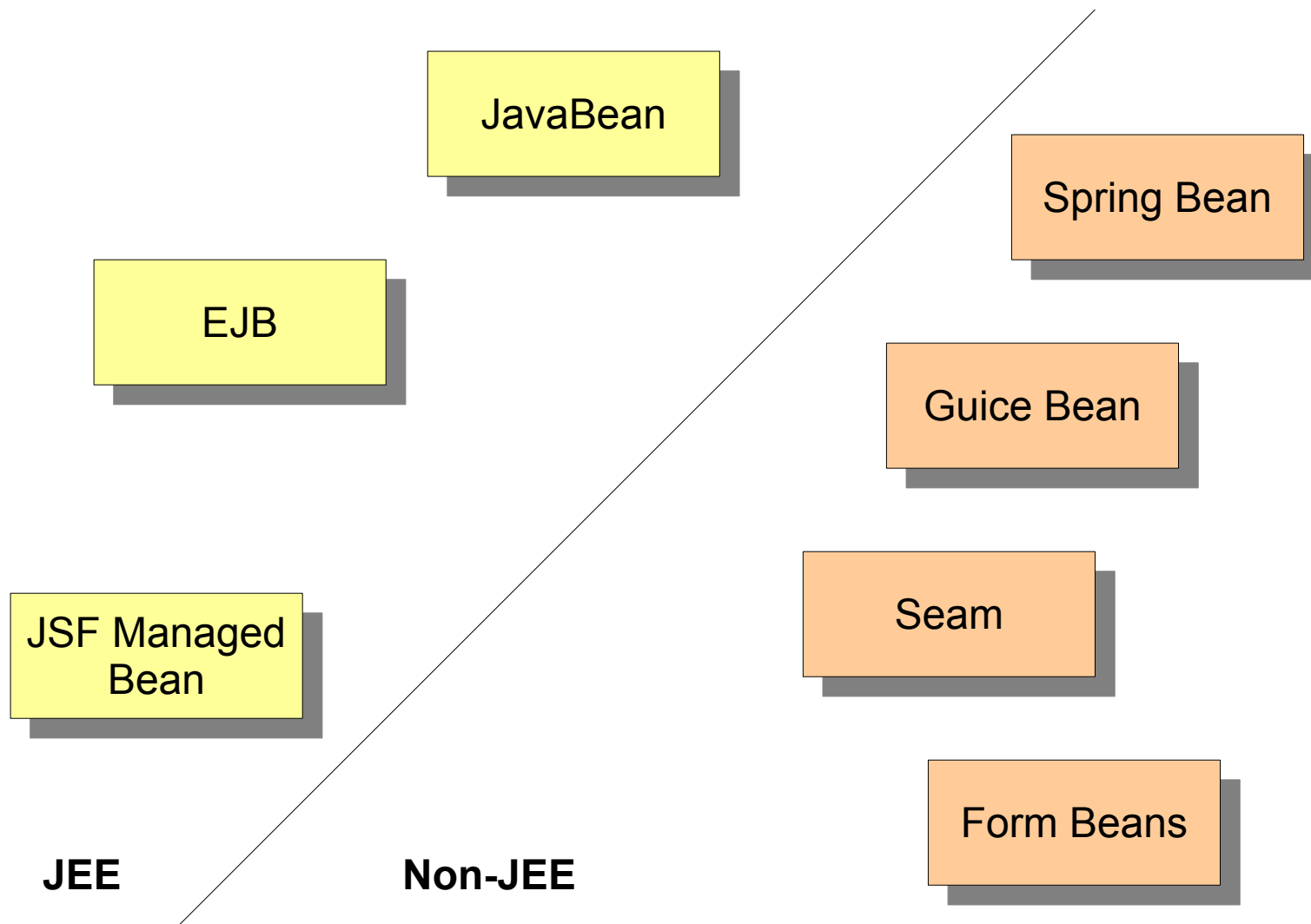
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



So many beans, so little common ground....



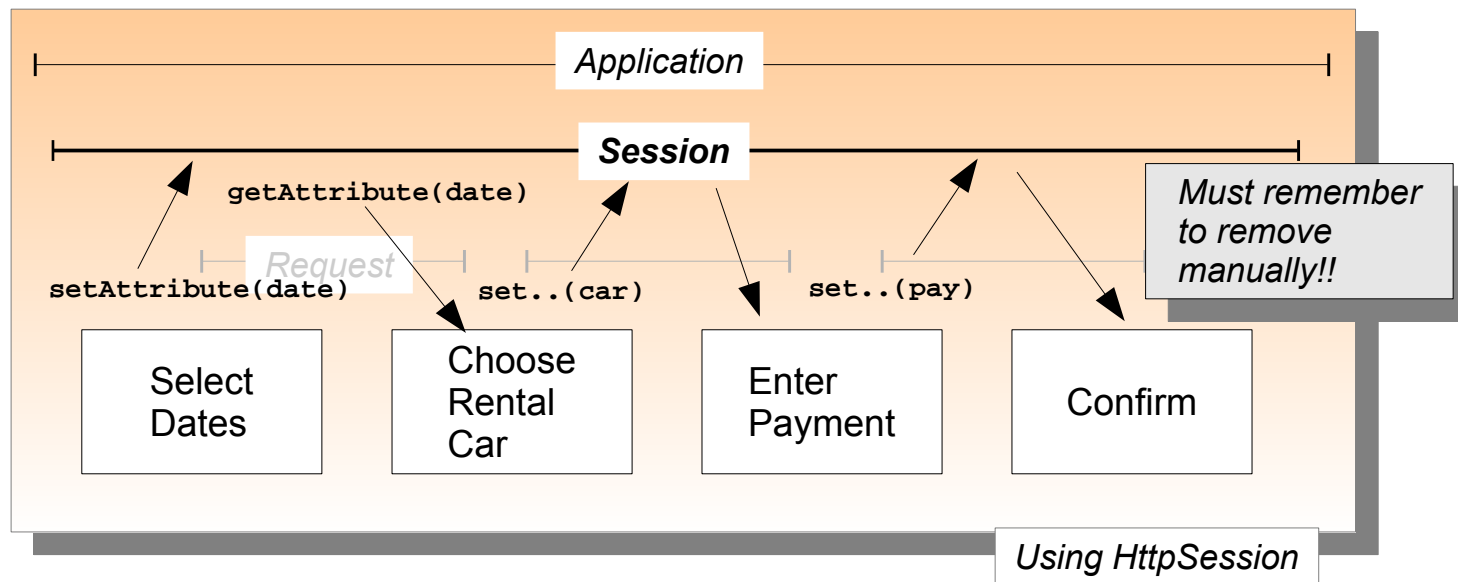
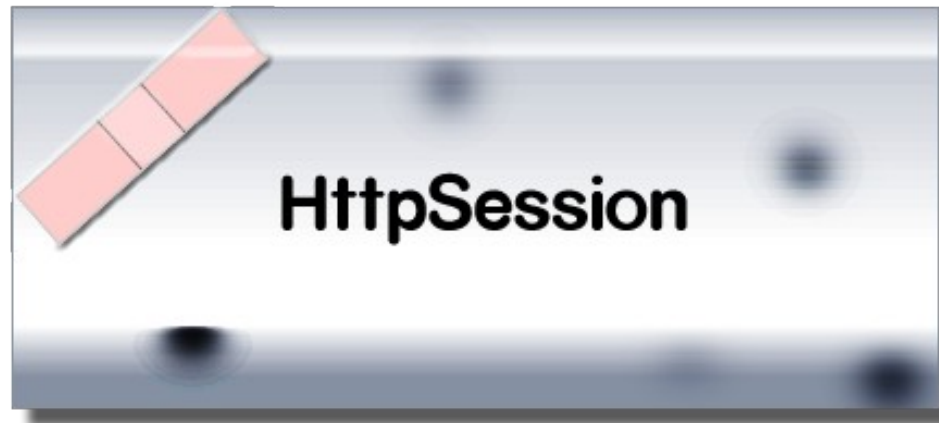
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Context Abuse! - HttpSession



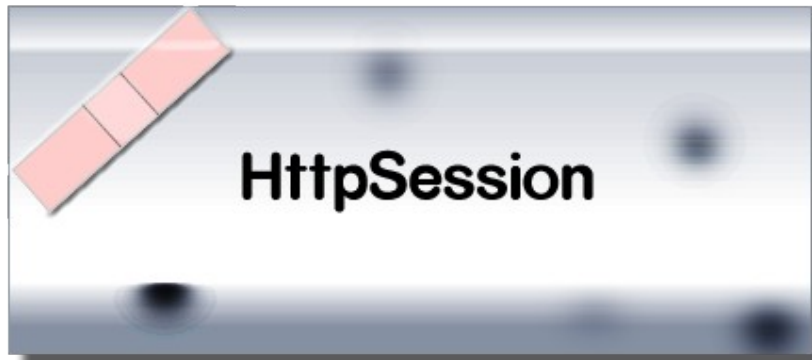
SUMMIT

JBoss
WORLD

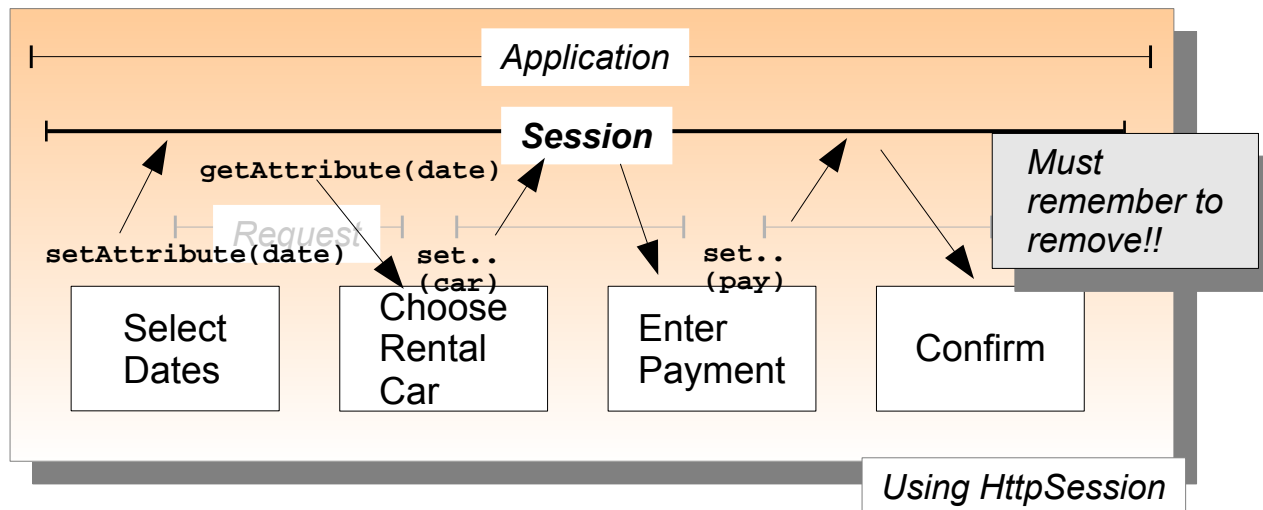
PRESENTED BY RED HAT



Context Abuse! - HttpSession - Why?



- Servlet Context Models are too coarse.
- Don't reflect real world usage.



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



We have a solution, but....

- Seam Introduced:
 - Turns everything into a “component”
 - One kind of “stuff”
 - Introduces Bijection
 - Makes dependency injection Stateful and Dynamic
 - Brings JSF and EJB together
 - Adds Conversation context and Business Process Context
- But....
 - Seam is still not JEE.



The Solution

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Solution: Contexts and Dependency Injection

- JSR-299 – Contexts and Dependency Injection for JEE
 - Core part of JEE 6
 - Assists in unifying the Bean model
 - Well defined contexts and the ability to bind beans statefully to them and manage the lifecycle.
 - Adds dependency injection to JEE and makes it type-safe.
 - Introduces an event notification system
 - Uses interceptors to foster loose coupling
 - Refines interceptors into decorators for finer grained control
 - Integrates with the Unified EL to bridge JSF
 - Introduces an SPI to extend JEE – Roll your own JEE7!

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



CDI Relations

- Spec – JSR-299 – CDI
 - *Also integrates with elements of JSR-330*
 - Reference Implementation – Weld
 - TCK – Technology Compatibility Kit
 - Implementation – Seam 3

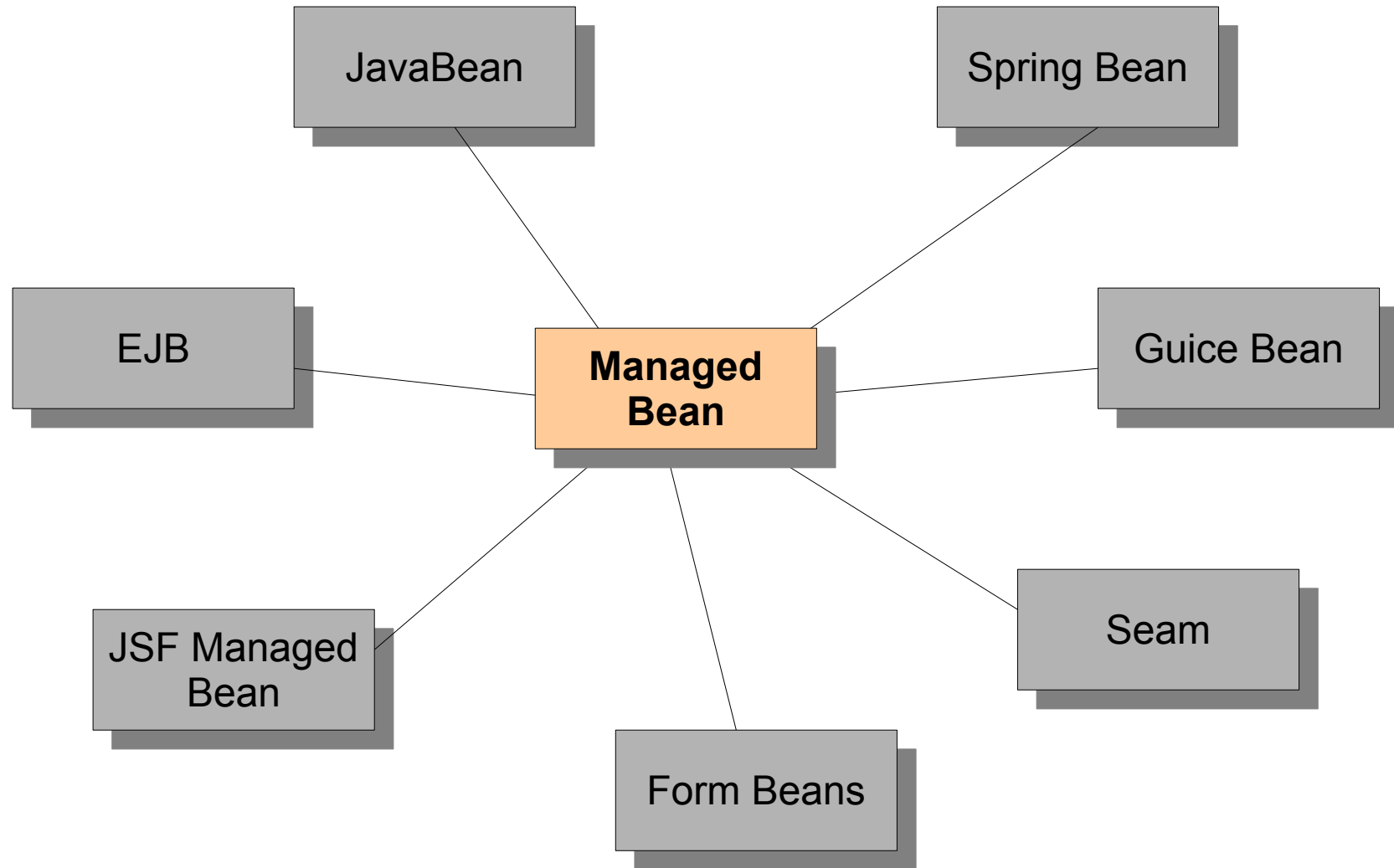


What's in a name? History of the Solution.

- Started as WebBeans
- Involved more than “Web” and more than just “Beans”
- Spec changed to CDI
- Reference Implementation initially called WebBeans.
- Objections on committee
- Referenced Implementation renamed to Weld
- *You will still see references to “WebBeans” when Googling for info and is even still in the filename of the spec!*



Unified Bean Model



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Managed Bean Details

- Any component managed by container is a Bean.
- Managed Bean baseline spec established.
 - Interceptors
 - Lifecycle Callbacks
 - Resource Injections
- In JEE6 and beyond, other bean specs build on this foundation.
 - CDI
 - EJB
 - etc...



CDI Beans – *We're back to “Managed Beans”*

- CDI adds the following to the Managed Bean services:
 - Auto-discovery
 - opt-in via beans.xml
 - Bean types
 - Ability to have a 'family' of bean types
 - Qualifiers (choose at design time)
 - Alternatives (choose at deploy time)
 - Producers (choose at run time)
 - Scope awareness and attachment
 - EL name (optional)
 - Exposes beans directly to JSFs and JSPs
 - Interceptor bindings



CDI Bean Example

```
public class Bike {  
    public String returnManufacturer() {  
        return "Made by JBoss";  
    }  
}
```



CDI Bean Example

```
@Stateful public class Bike {  
    . . . . .
```

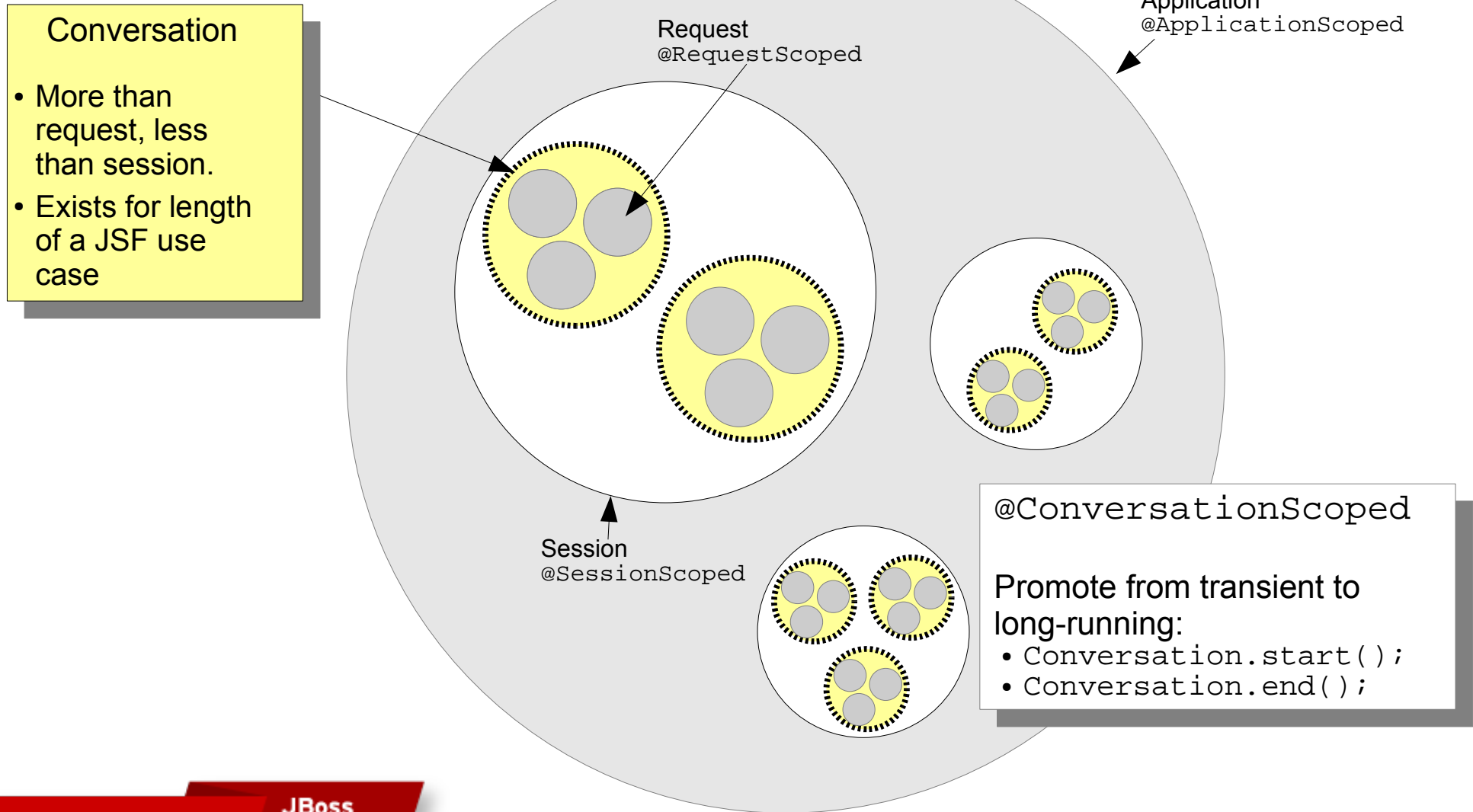
Same bean as an EJB

```
@Named  
public class Bike {  
    . . . . .
```

Same bean exposed to a JSF



Well Defined Contexts – Addition of Conversation



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Contextual Model

@SessionScoped

```
public class Race() {  
  
    @Inject @Motocross Motorcycle racer1  
    . . .  
}
```

Pseudo Scope:

- @Dependent

Servlet Scopes:

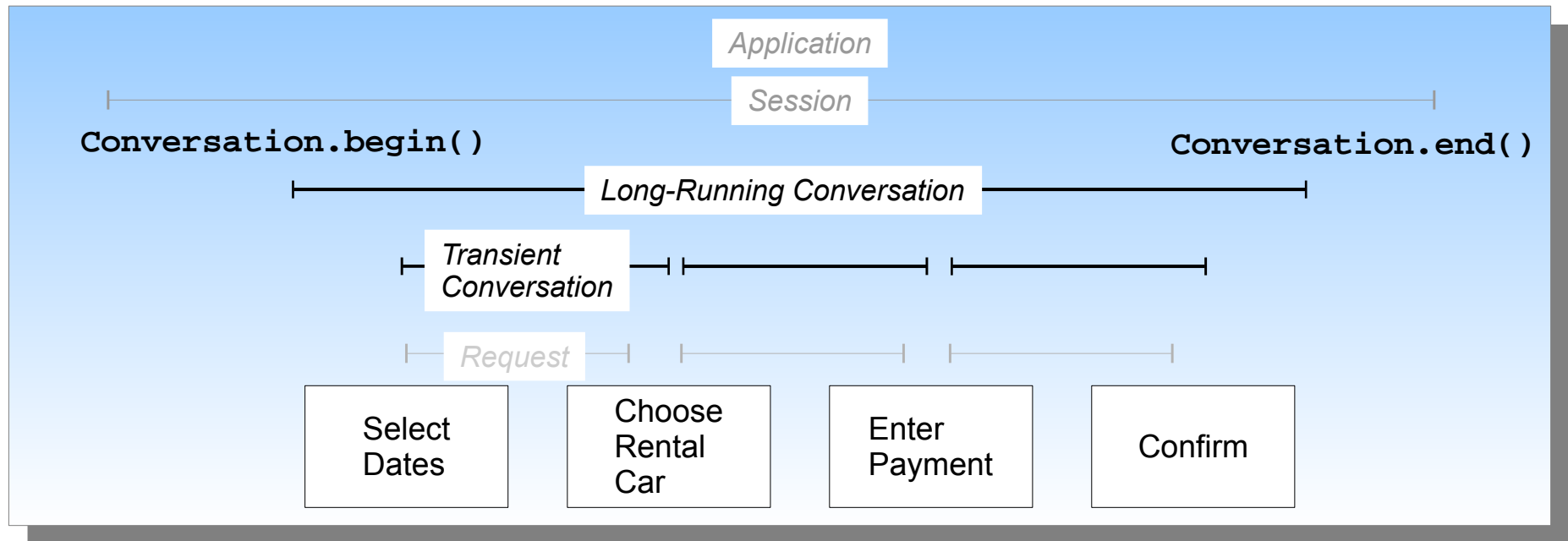
- @RequestScoped
- @SessionScoped
- @ApplicationScoped

JSF Scope:

- @ConversationScoped



Conversation Scope



```
@ConversationScoped
public class RentCar {
    @Inject Conversation conversation;

    public void rentCar(){
        CODE TO RENT CAR
        conversation.begin();
    } ...
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Type Safe Dependency Injection

- Injection is now Type Based.
Bean injection typing based on combination of:
 - Annotation
 - Type
 - Type Params
- Eliminates name/string based injection.



A Closer Look – CDI Injection

```
@Motocross  
public class Motorcycle extends Vehicle  
implements Bike<Motorized> {  
    ...  
}
```

4 ways this Bean could be injected:

- Motorcycle
- Vehicle
- Wheel<TwoWheel>
- @Motocross



Qualifiers

@Motocross is a **Qualifier**. When you have more than one bean that implement the same interface, this is a way to distinguish them. Acts as a binding type.

@Motocross

```
public class Motorcycle extends Vehicle
implements Bike<Motorized> {
    ...
}
```

@Qualifier

```
@Retention(RUNTIME)
```

```
@Target ({TYPE, METHOD, FIELD})
```

```
public @interface Motocross { }
```



@Alternative

Alternatives act as a deploy-time override must be specifically enabled. They then take the place of the matching type.

```
@Alternative @Motocross
public class TestBike extends Vehicle
implements Bike<Motorized> {
    ...
}
```

Enabled by...

```
<alternatives>
  <class>
    org.jbossmotors.mock.TestBike
  </class>
</alternatives>
```

beans.xml

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Producers

A **Producer method** is a method that acts as a source of bean instances. The method declaration itself describes the bean and the container invokes the method to obtain an instance of the bean when no instance exists in the specified context

```
@ApplicationScoped
public class RandomNumberGenerator {
    private Random random =
        new Random(System.currentTimeMillis());

    @Produces @Named @Random int getRandomNumber() {
        return random.nextInt(100);
    }
}
```

Used to dynamically fulfill an injection at Runtime.

Producer fields are a simpler alternative to methods. Are particularly good for JEE component reference injection.

```
@Produces @WebServiceRef
(lookup="java:app/service/Catalog") Catalog catalog;
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Event Notifications

Represent a way to completely decouple the event producer from the observer and vice versa.

```
@Inject Event<Package> delivered;  
  
public void delivery(String recipient) {  
    delivered.fire(new Package(recipient, timeStamp) );  
}
```

Producer: DeliveryTracking.java

```
void onDelivery(@Observes Package package,  
               @Default Recipient recipient) {  
    System.out.println(recipient +  
        " recieved " + package.getContents());  
}
```

Receiver: DeliveryNotifier.java

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



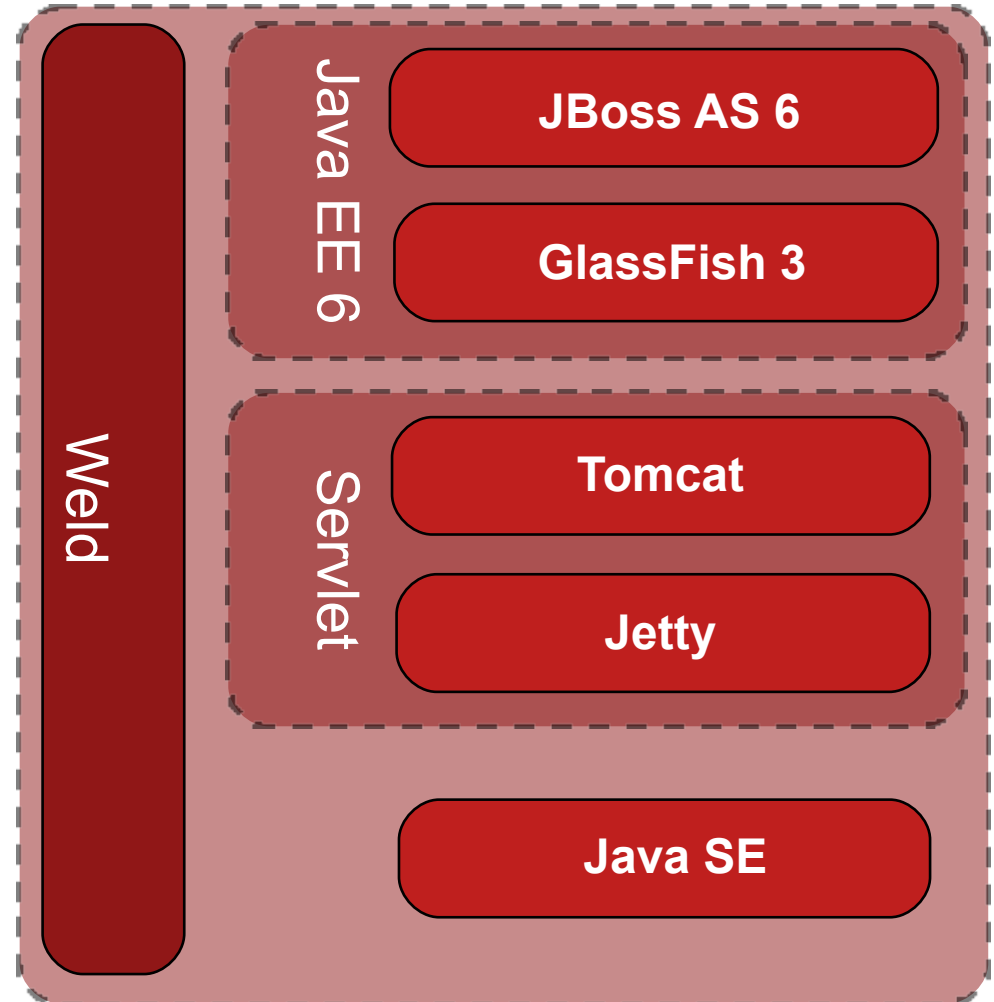
Unified EL Integration

- Unified EL exposes Managed Beans to JSF and JSP views.
 - All that is required is the `@Named` annotation.
- Many resources are injectable:
 - Managed Beans
 - Web service references
 - Local or Remote EJBs
 - Producer-created objects
 - etc...



Weld

- Reference implementation of CDI
 - “Vanilla” CDI
- Is the CDI implementation in JBoss AS 6 and GlassFish 3
 - But also can support Tomcat and Jetty and even JavaSE
- Is Mavenized
- Provides the foundation for extensions.
 - Weld extensions are for extending the core.



Where Is It All Going Next? - Seam 3

- Seam is now modularized with CDI at the core.
 - Modules have an independent release cycle.
 - Regular Snapshot/Packaged Releases
- The Seam stack consists of...
 - Java EE integration within platform
 - Declarative security
 - Page flows (jPDL) and business processes (jBPM)
 - JavaScript remoting
 - "Render" e-mail, graphics, PDF, and XLS
 - Spring integration and more...
- Tooling

SUMMIT

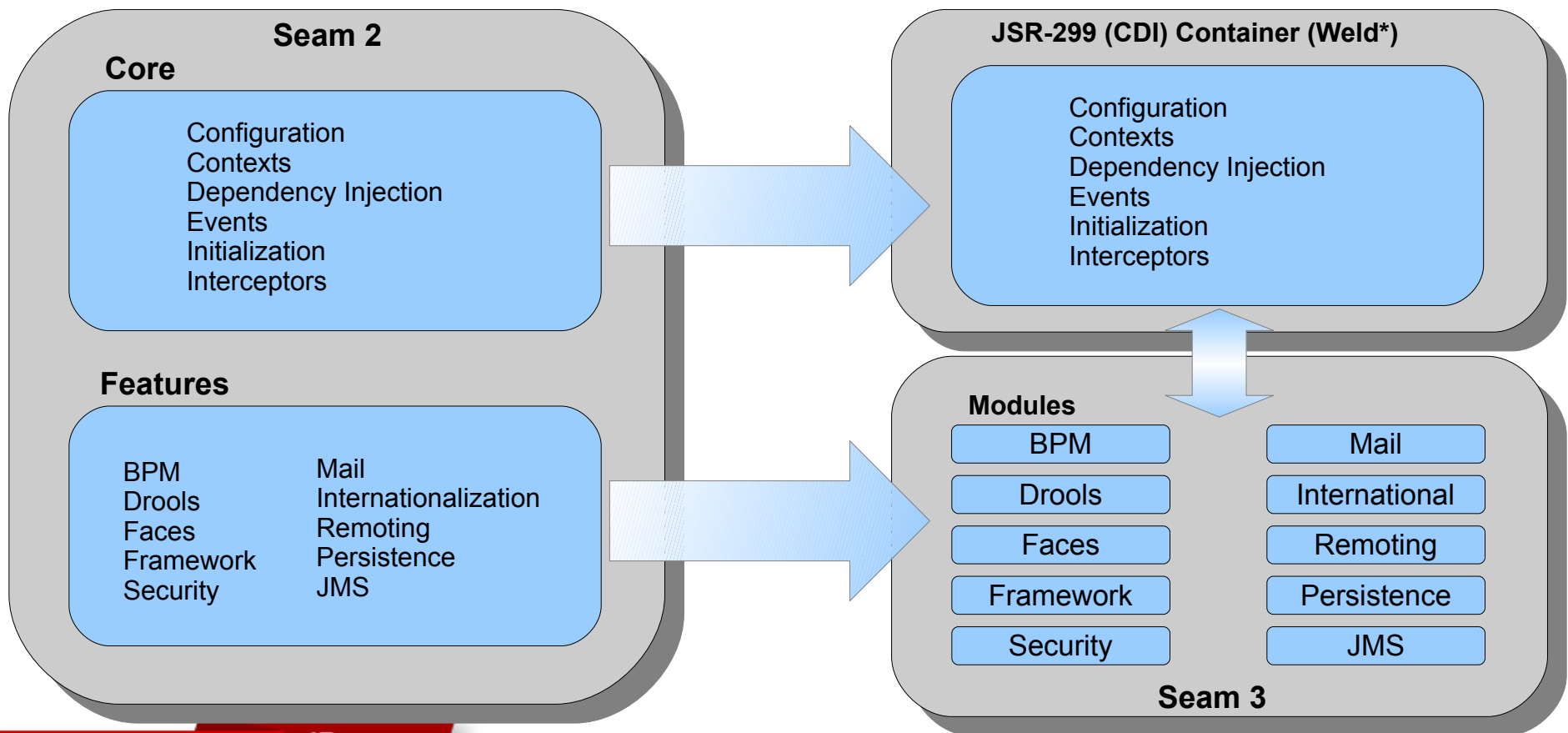
**JBoss
WORLD**

PRESENTED BY RED HAT



Seam 2 to Seam 3

- Seam 3 = Portable Extensions for Java EE.
- Built on top of the CDI foundation
- Now organized as modules.



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Seam 3 Modules

Name	VCS Folder	Lead(s)	Purpose	Issue tracker	Functional?	Build status	Release Available
Drools	drools	Tihomir Surdilovic	Drools integration	SEAMDROOLS	✓		✗
Faces (JSF)	faces	Lincoln Baxter III Dan Allen	JSF enhancements	SEAMFACES	✓		✓
International (i18n & l10n)	international	Lincoln Baxter III Ken Finnigan	Contextual internationalization and localization support	SEAMINTL	✓		✓
JMS	jms	Jordan Ganoff	JMS resource injection and CDI event bridging	SEAMJMS	✓		✓
Transactions & persistence	persistence	Dan Allen	Transaction and persistence for managed beans	✗	✗	n/a	✗
JavaScript remoting	remoting	Shane Bryzak	JavaScript remoting (ala DWR)	SEAMREMOVING	✓		✓
Security	security	Shane Bryzak	Authentication, identity and permissions management, rule-based security, ACLs	SEAMSECURITY	✗		✗
Servlet	servlet	Nicklas Karlsson Lincoln Baxter III	Utilities and functionality to complement the Servlet API	SEAMSERVLET	✓		✗
Wicket	wicket	Clint Popetz	Wicket integration	SEAMWICKET	✓	n/a	✗
XML configuration	xml	Stuart Douglas	XML-based bean configuration	SEAMXML	✓		✓

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Seam 3 vs. Seam 2

- Annotations have changed.
 - ex. `@In` now `@Inject`, `@Name` now `@Named`
 - ex. `@ScopeType` replaced by scope annotations - `@RequestScope`
 - ex. `@Begin/@End` now called from Conversation interface - `Conversation.begin()`.
- Some concepts/implementations replaced:
 - Outjection – Instead of outjecting something into scope, make sure you are reading from the scope
 - Also, safer alternative using Producer Fields
 - Factories
 - Safer result from Producer Methods



For Latest Information

- <http://www.seamframework.org>
 - Downloads
 - Documentation
 - Forums and Community
 - Contribute!

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Summary

- JSR-299 introduces a flexible set of JEE services:
 - Managed Beans that are Strong Typed
 - Contexts
 - Dynamic Dependency Injection
 - Numerous avenues for loose coupling
 - Extendable SPI
- Weld is the reference implementation of CDI and is the implementation in JBoss 6 and GlassFish3.
 - Can be added to Tomcat, Jetty
- Seam 3 – Portable Extensions for JEE (via CDI)



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

