



Java. Cloud. Leadership.

Introducing Contexts and Dependency Injection 1.1

Pete Muir
Principal Software Engineer
Red Hat, Inc.
October 2011

@plmuir

- CDI 1.0
 - Insanely productive
- CDI 1.1
 - (Global) Ordering
 - Java SE support
 - Extensions & Deployment
 - Context/Scope improvements
 - Tracking



Java. Cloud. Leadership.



Java. Cloud. Leadership.



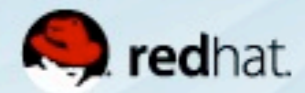
Java. Cloud. Leadership.

<http://bit.ly/OpenShiftAcme>

<http://acme-pmuir.rhcloud.com>

What you will learn?

- **What** problems we want to solve
- Our **ideas** on how to solve them
- How you can **influence** the CDI 1.1



Java. Cloud. Leadership.



Issue tracker



Issue tracker
Mailing list



Issue tracker

Mailing list

Wiki



Issue tracker

Mailing list

Wiki

Review process





Issue tracker

Mailing list

Wiki

Review process

Spec (git)



Community Sponsored Issues



Community
Sponsored
Issues

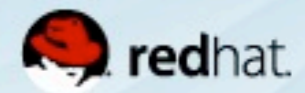
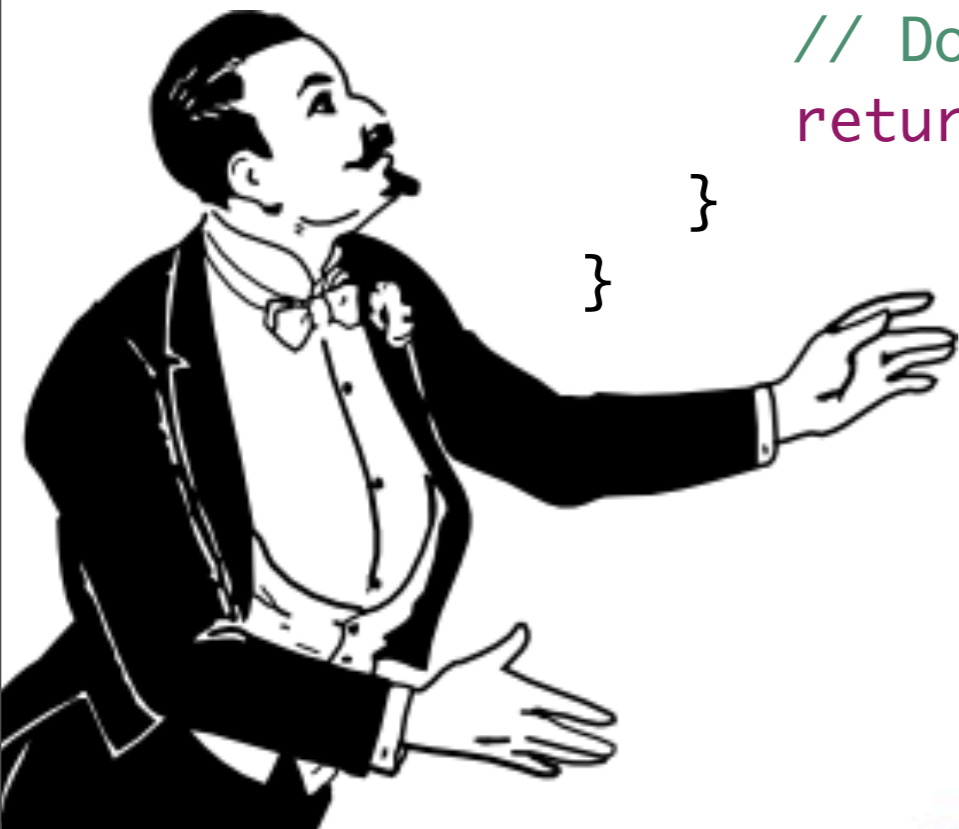
(Global) Ordering

Option 1: Magic numbers

```
@Shot @Interceptor(37)
public class Goalkeeper {

    @AroundInvoke
    public Object catchBall(InvocationContext ctx)
        throws Exception {

        // Do something
        return invocationContext.proceed();
    }
}
```



Java. Cloud. Leadership.

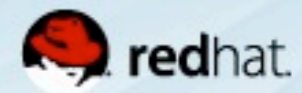
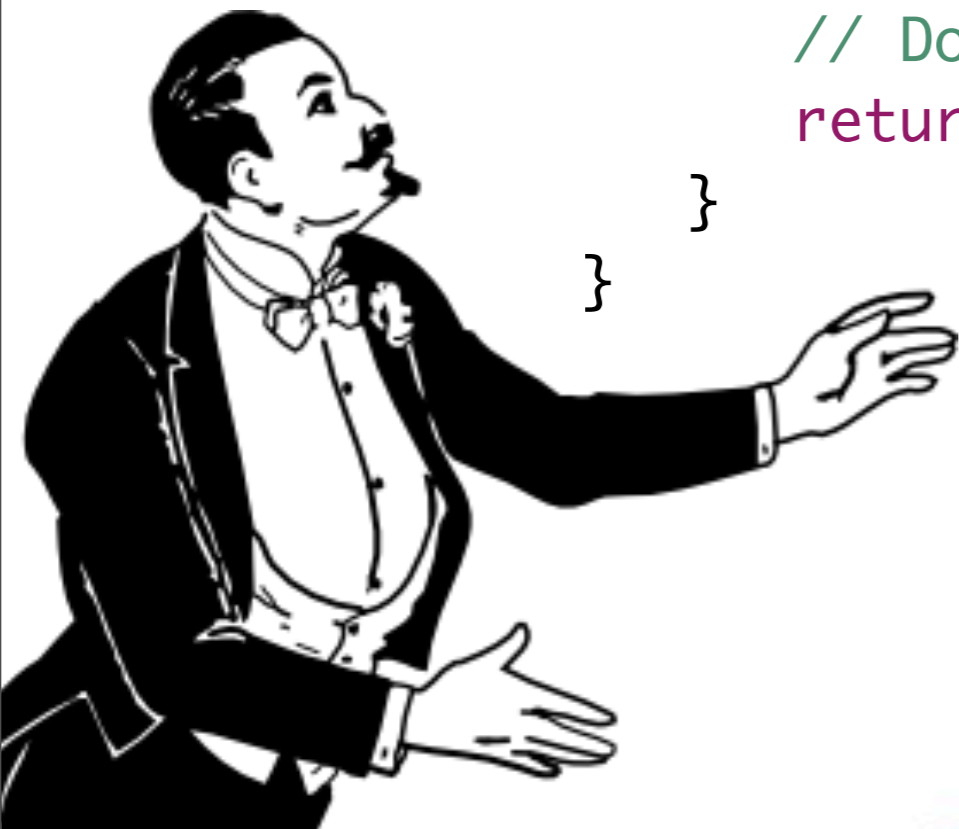
(Global) Ordering

Option 1: Magic numbers

```
@Shot @Interceptor(37)
public class Goalkeeper {

    @AroundInvoke
    public Object catchBall(InvocationContext ctx)
        throws Exception {

        // Do something
        return invocationContext.proceed();
    }
}
```



Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
@LastHope @Interceptor  
public class Goalkeeper {
```

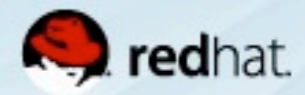
```
    ...
```

```
}
```

```
@Interceptor @Before(LastHope.class)  
public class Defender {
```

```
    ...
```

```
}
```



Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
@LastHope @Interceptor  
public class Goalkeeper {
```

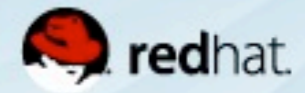
```
...
```

```
}
```

```
@Interceptor @Before(LastHope.class)  
public class Defender {
```

```
...
```

```
}
```



Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
@LastHope @Interceptor  
public class Goalkeeper {
```

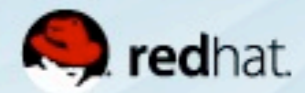
```
...
```

```
}
```

```
@Interceptor @Before(LastHope.class)  
public class Defender {
```

```
...
```

```
}
```

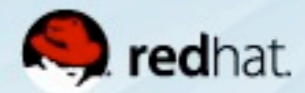


Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
@InterceptorBinding  
@OrderingQualifier  
@Retention(RUNTIME)  
@Target({METHOD, TYPE})  
@Documented  
public @interface LastHope {}
```

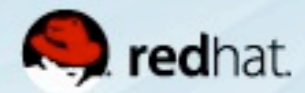


Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
@InterceptorBinding  
@OrderingQualifier  
@Retention(RUNTIME)  
@Target({METHOD, TYPE})  
@Documented  
public @interface LastHope {}
```



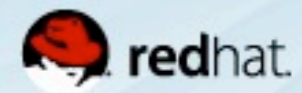
Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
public class Linesman {  
  
    @Before(Official.class)  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```

```
public class Referee {  
  
    @Official  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```



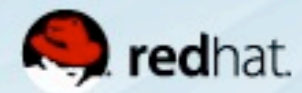
Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
public class Linesman {  
  
    @Before(Official.class)  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```

```
public class Referee {  
  
    @Official  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```



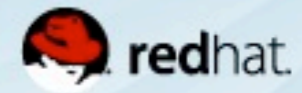
Java. Cloud. Leadership.

(Global) Ordering

Option 2: Relative ordering

```
public class Linesman {  
    @Before(Official.class)  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```

```
public class Referee {  
    @Official  
    public void observeGoal(@Observes Goal goal) {  
        ...  
    }  
}
```



Java. Cloud. Leadership.

(Global) Ordering

Option 3: Imperative ordering

```
public class ReorderExtension implements Extension {  
    public void reorder(@Observes AfterDeploymentValidation adv) {  
        Collections.sort(adv.getInterceptors(),  
            new InterceptorOrderingComparator());  
    }  
}  
  
public interface AfterDeploymentValidation {  
    public List<Class<?>> getInterceptors();  
    ...  
}
```



(Global) Ordering

Option 3: Imperative ordering

```
public class ReorderExtension implements Extension {  
  
    public void reorder(@Observes AfterDeploymentValidation adv) {  
        Collections.sort(adv.getInterceptors(),  
            new InterceptorOrderingComparator());  
    }  
}  
  
public interface AfterDeploymentValidation {  
    public List<Class<?>> getInterceptors();  
    ...  
}
```



(Global) Ordering

- Option 2 & Option 3
- Global enablement now possible

```
<beans xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

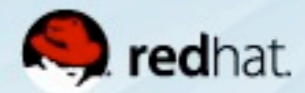
```
  <interceptors>  
    <class>com.javaone.CheckBadge</class>  
  </interceptors>
```

```
</beans>
```

Java SE

Get a handle

```
public class SoccerHacker {  
  
    public void fixTheMatch() {  
        CDI.current()  
            .select(new TypeLiteral<Event<Goal>>(){})  
            .get()  
            .fire(new Goal());  
    }  
}
```

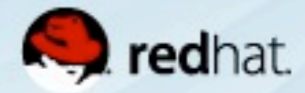


Java. Cloud. Leadership.

Java SE

Get a handle

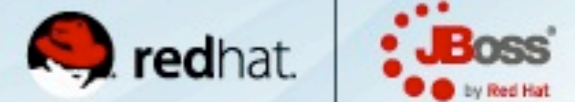
```
public abstract class CDI<T> implements Instance<T> {  
    public static <T> CDI<T> current() {  
        ...  
    }  
  
    public abstract BeanManager getBeanManager();  
}
```



Java. Cloud. Leadership.

Java SE

Boot the container

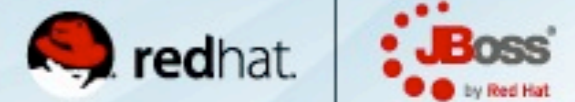


Java. Cloud. Leadership.

TODO



Java SE

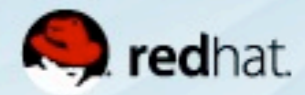


Java. Cloud. Leadership.

Java SE



Core



Java. Cloud. Leadership.

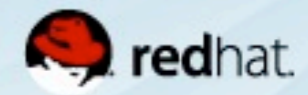
Java SE



Core

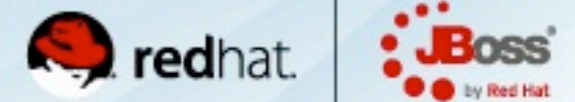


Java EE
integrations



Java. Cloud. Leadership.

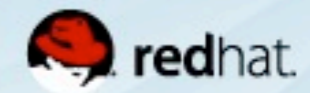
Java SE



Java. Cloud. Leadership.

Java SE

DSL for bean deployment



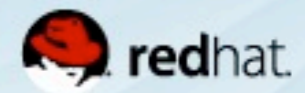
Java. Cloud. Leadership.

Java SE

DSL for bean deployment



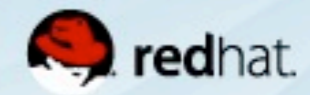
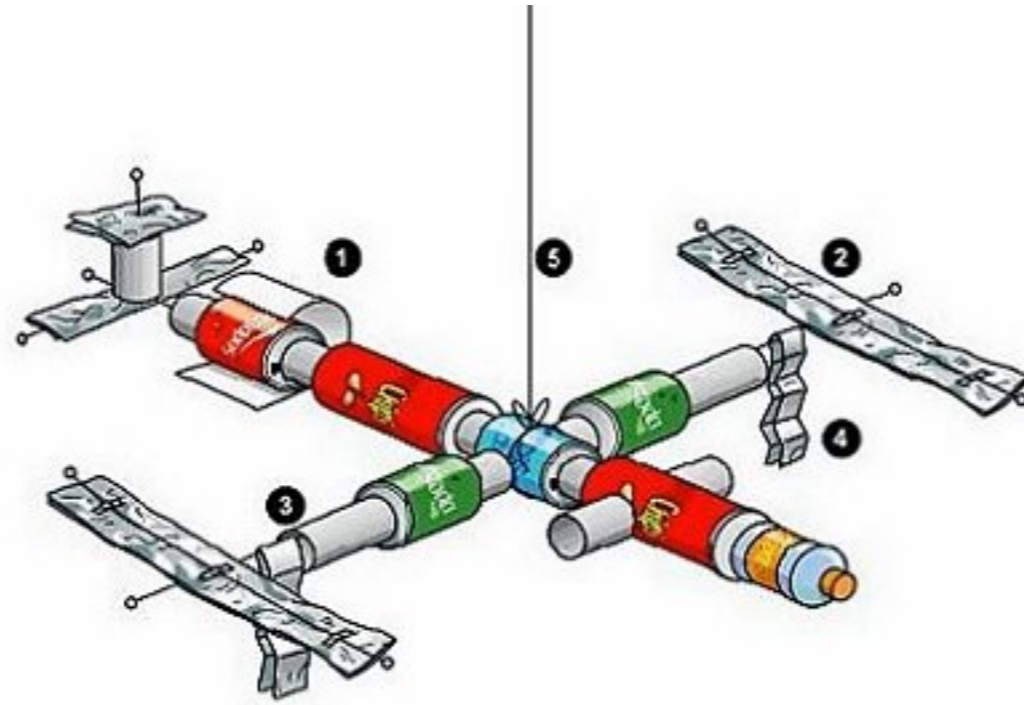
Define context lifecycles



Java. Cloud. Leadership.

Extensions & Deployment

ProcessModule



Java. Cloud. Leadership.

Extensions & Deployment

ProcessInjectionPoint



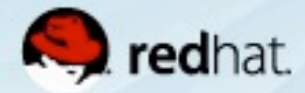
Extensions & Deployment

Declarative control of scanning

```
<beans xmlns="http://java.sun.com/xml/ns/javaee"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <scan>
    <exclude name="com.acme.customer.**">
      <if-class-available name="com.acme.Customers">
        </exclude>
      </if-class-available>
    </exclude>
  </scan>

</beans>
```



Java. Cloud. Leadership.




```
@Veto  
public class Goalkeeper {  
  
    ...  
  
}
```



Extensions & Deployment

Service Handler

```
@QueryService
interface Users {

    @Query("select u from User u")
    public List<User> getAllUsers();

}
```

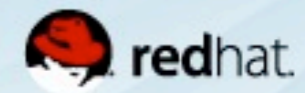
```
class UserListManager {

    @Inject
    UserQuery q;

    List<User> users;

    @PostConstruct
    void create() {
        users = q.getAllUsers();
    }

}
```



Java. Cloud. Leadership.



Extensions & Deployment



Service Handler

```
@QueryService
interface Users {

    @Query("select u from User u")
    public List<User> getAllUsers();

}
```

```
class UserListManager {

    @Inject
    UserQuery q;

    List<User> users;

    @PostConstruct
    void create() {
        users = q.getAllUsers();
    }

}
```



Java. Cloud. Leadership.



Extensions & Deployment

Service Handler

```
@ServiceHandlerType(QueryHandler.class)
@Retention(RUNTIME)
@Target(TYPE)
@interface QueryService {}
```

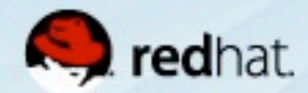
```
class QueryHandler {

    @Inject EntityManager em;

    @AroundInvoke
    Object handle(InvocationContext ctx) {
        return em.createQuery(ctx.getMethod()
                               .getAnnotation(Query.class)
                               .value())
                   .getResultList();
    }
}
```

```
@Retention(RUNTIME)
@Target(METHOD)
@interface Query {

    String value();
}
```



Java. Cloud. Leadership.



Extensions & Deployment

Service Handler

```
@ServiceHandlerType(QueryHandler.class)
@Retention(RUNTIME)
@Target(TYPE)
@interface QueryService {}
```

```
class QueryHandler {

    @Inject EntityManager em;

    @AroundInvoke
    Object handle(InvocationContext ctx) {
        return em.createQuery(ctx.getMethod()
                               .getAnnotation(Query.class)
                               .value())
               .getResultList();
    }
}
```

```
@Retention(RUNTIME)
@Target(METHOD)
@interface Query {

    String value();
}
```



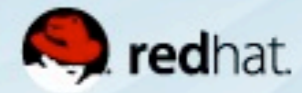
Java. Cloud. Leadership.

Context/Scope improvements

Lifecycle events

```
‘ public void onStartUp(@Observes
    @Initialized(ApplicationScoped.class)
    Object payload,
    EntityManager em) {
    ...
}

public void onLogin(@Observes
    @Initialized(SessionScoped.class)
    HttpSession session, ...) {
    ...
}
```

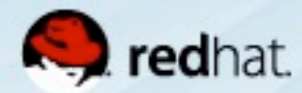


Java. Cloud. Leadership.

Context/Scope improvements

Unmanaged instances

```
Instance<Bar> instance = CDI.current().select(Bar.class);  
Bar bar = instance.get();  
bar.doSomething();  
instance.destroy(bar);
```



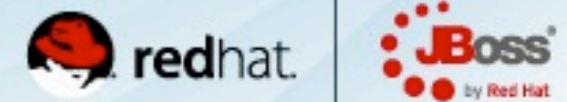
Java. Cloud. Leadership.

Context/Scope improvements



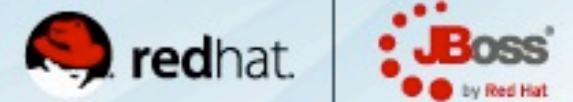
Unmanaged instances

```
Instance<Bar> instance = CDI.current().select(Bar.class);  
Bar bar = instance.get();  
bar.doSomething();  
instance.destroy(bar);
```



Java. Cloud. Leadership.

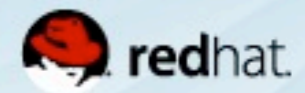
Context/Scope improvements



Java. Cloud. Leadership.

Context/Scope improvements

Conversations extended to servlet

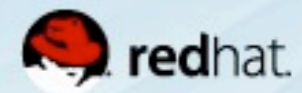


Java. Cloud. Leadership.

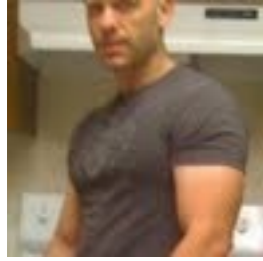
Context/Scope improvements

Conversations extended to servlet

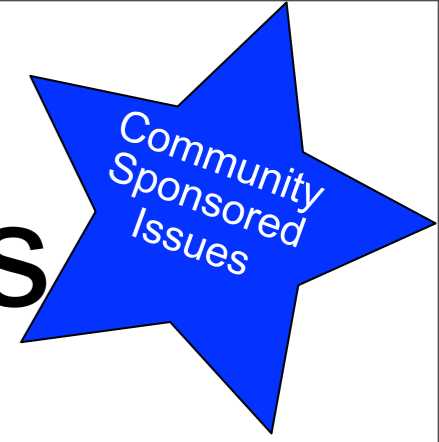
Conversation propagation control



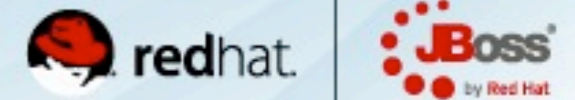
Java. Cloud. Leadership.



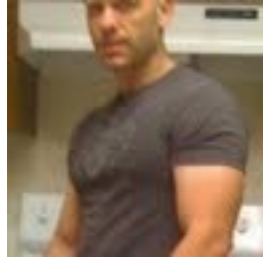
Context/Scope improvements



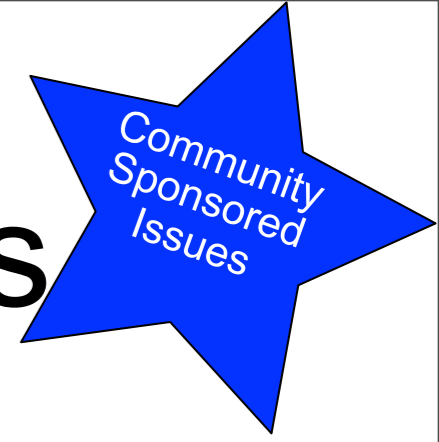
Transaction Scope



Java. Cloud. Leadership.



Context/Scope improvements



Transaction Scope



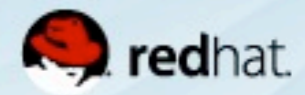
Java. Cloud. Leadership.

Tracking

Declarative transactions

```
@TransactionAttribute(TransactionAttributeType.REQUIRED)
public void doSomeWork() {

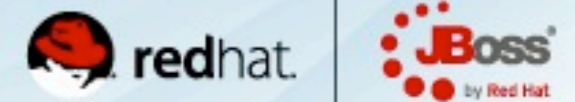
}
```



Java. Cloud. Leadership.

Tracking

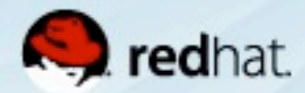
JMS integration



Java. Cloud. Leadership.

And more

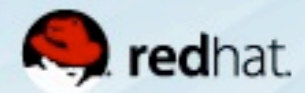
- Serialization/Passivation tidy up
- Injection of Manifest
- Static & enum injection



Java. Cloud. Leadership.

More sessions at JavaOne

- CDI Today and Tomorrow - panel session on CDI with **David Blevins, Arun Gupta, Sivakumar Thyagarajan** and **Reza Rahman** - 1:30pm on Thursday Imperial Ballroom A



Java. Cloud. Leadership.



Java. Cloud. Leadership.

Q&A

cdi-dev@lists.jboss.org

<https://issues.jboss.org/browse/CDI>

<http://github.com/jboss/cdi>

<https://github.com/jboss/cdi/wiki>

@plmuir