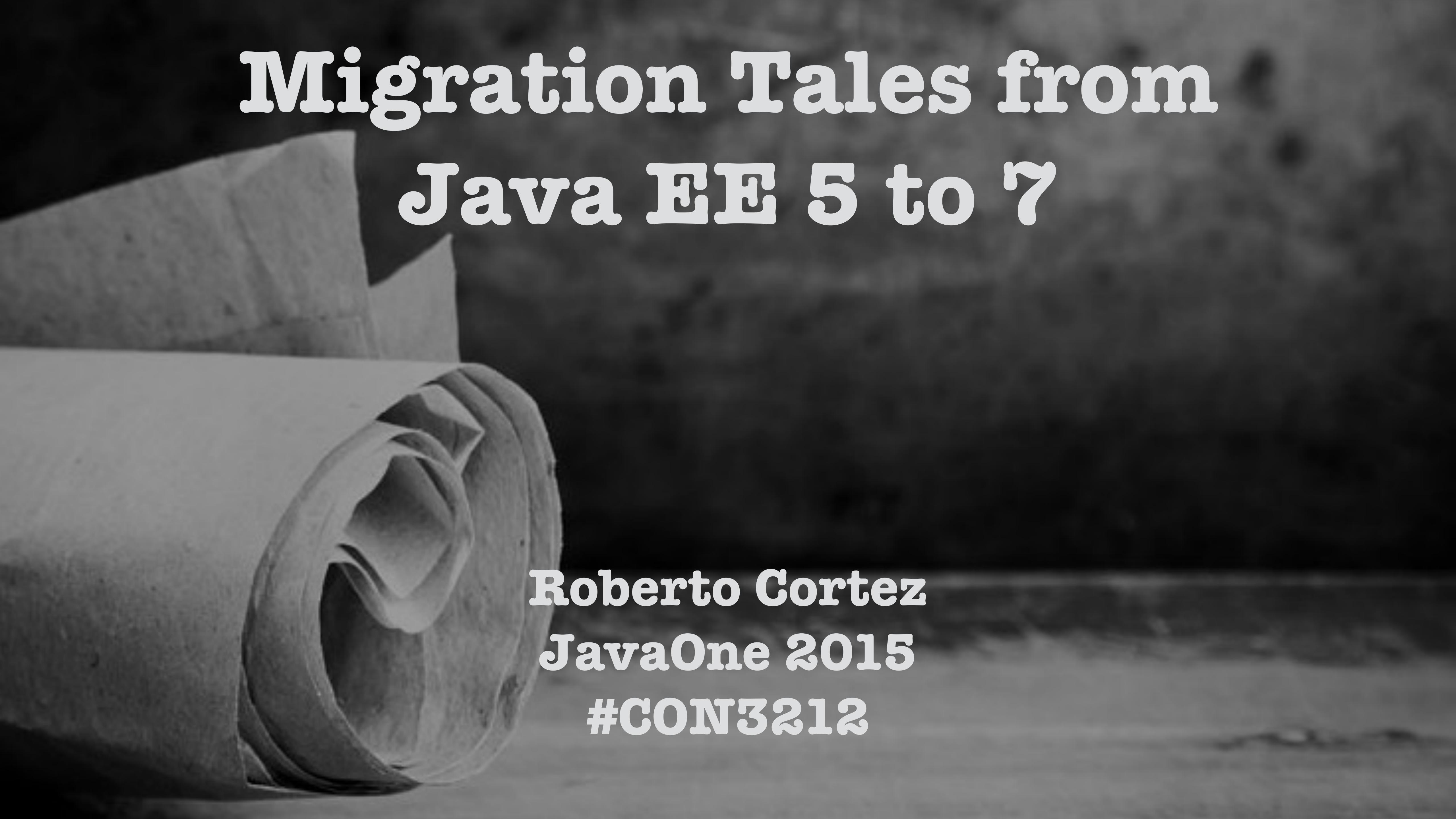


# Migration Tales from Java EE 5 to 7



Roberto Cortez  
JavaOne 2015  
**#CON3212**

# Roberto Cortez

Speaker, Blogger, Passionate Developer

**twitter:**

@radcortez

**mail:**

radcortez@yahoo.com

**blog:**

<http://www.radcortez.com>



# This presentation is NOT about:

- Technology X versus Y
- Follow a step by step plan (**BANG!** You're done)
- Each case is different

# It's about:

- Sharing knowledge
- Telling our story
- Learning about possible issues



# Questions?

A black and white photograph showing a close-up of a person's hand. The hand is resting on a dark, textured surface, likely a desk or table. A pen is held vertically between the thumb and forefinger. In the background, a portion of a lined notebook is visible, showing several horizontal lines. The lighting is dramatic, with strong highlights and shadows.

# Questions?

As soon as you have them!

# A little bit of History

- Java EE 5 (2006) and Java EE 7 (2013) are 7 years apart
- 7 years in technology is like a century
- Major changes introduced between 5 and 7

# A little bit of History

JSF JSP  
EL

JAX-WS

Servlet

Interceptors  
JTA

Common  
Annotations

EJB

JPA

JMS

JCA

# A little bit of History

CDI  
Extensions

Web  
Fragments

JSF JSP  
EL

JAX-WS  
JAX-RS

Servlet

CDI

Interceptors  
JTA

Common  
Annotations

Bean  
Validation

Managed Beans

EJB

JPA

JMS

JCA

# A little bit of History

CDI  
Extensions

Web  
Fragments

JSF JSP  
EL

JAX-WS  
JAX-RS

JSON

WebSocket

Servlet

CDI

Interceptors  
JTA

Common  
Annotations

Concurrency

Bean  
Validation

Managed Beans

EJB

JPA

JMS

JCA

Batch

# The Platform

- Segurnet acts as an integration network for the Insurance sector in Portugal.
- It's central to the Business
- New challenges to keep the platform going

# The Platform

- SOA based Architecture
- XML Messages by WS or Batch
- Online Application

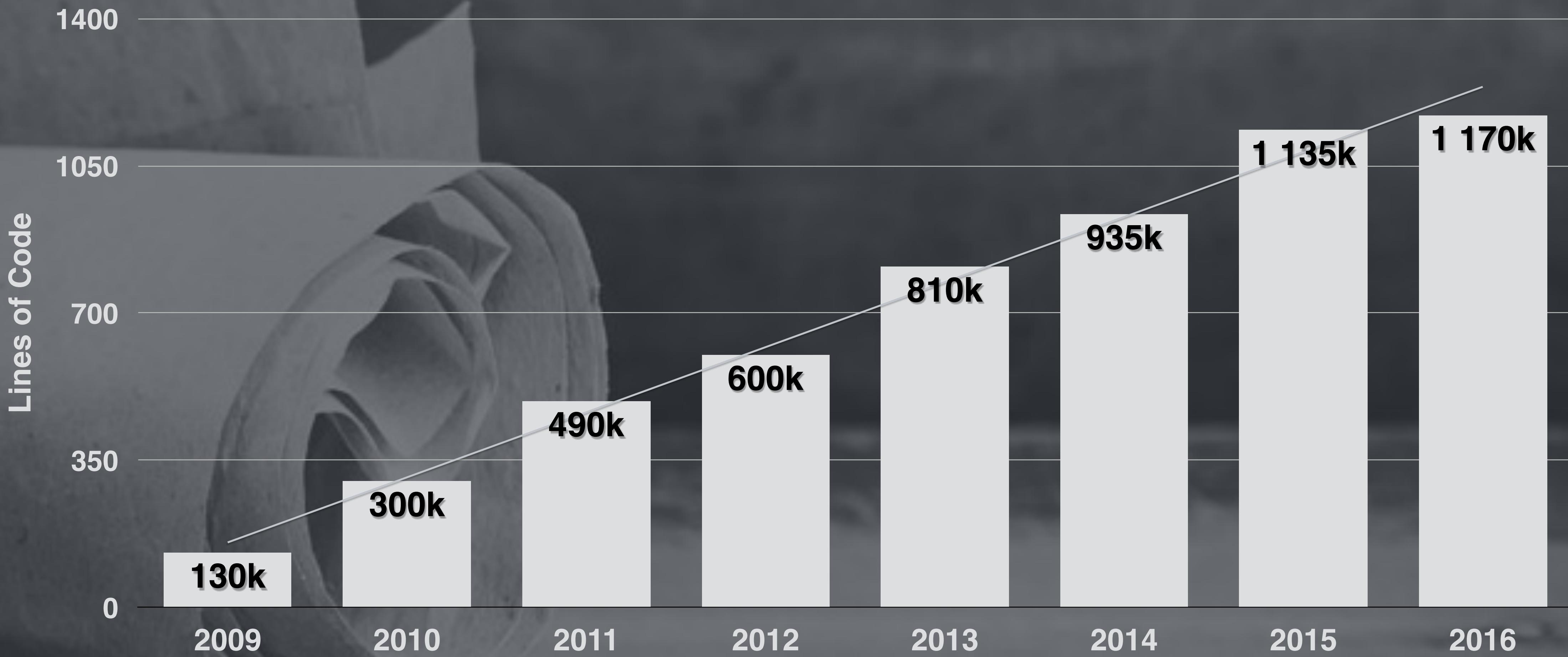
# A few Numbers

- 65 Companies
- 32 258 Active Users
- 141 084 250 Page Views
- 325 605 701 Server Requests

# Legacy

- Since 2007 the platform is using Java EE 5 technology in a Java 6 environment
- Other technologies are also used: GWT, Spring, Batch, Apache Projects, Velocity, BIRT
- Runs in a JBoss 4.2.3 server

# Stop the Legacy



# Motivations to Migrate

- Benefit from the latest improvements
- Java 6 EOL in Q1 2013
- Increased maintenance costs
- Hard to keep developers interested



# The Prelude

# On with the Journey

- Migrations are not straightforward
- It's difficult to know the path
- Expect the unexpected

# Strategy: Feature Freeze?

- Delays new Business Features
- Non negotiable deadline
- One Shot

# Strategy: Combined?

- Keep Development going
- Best for Business
- Requires more discipline and planning
- By Module, Canary, Application

# First steps

- Application Server Upgrade to Wildfly 8.2.0
- Update / Duplicate Environments
- Support Tools also need Migration / Tweaking  
(Maven, Jenkins, Sonar)

A dark, atmospheric forest scene. The foreground is filled with dense, dark foliage and ferns. In the background, tall, thin trees stand silhouetted against a lighter sky, creating a sense of depth and mystery.

# Walking the Path

# Dependency Chaos

- Code tied to the Application Server
- Code tied to specific Java EE implementations  
(especially Hibernate)
- In house code to bypass Java EE 5 limitations

# Dependency Chaos



# Dependency Chaos

```
<dependency>
  <groupId>javax.ejb</groupId>
  <artifactId>ejb-api</artifactId>
  <version>3.0</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
</dependency>

<dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>persistence-api</artifactId>
    <version>1.0</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
</dependency>

<dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>persistence-api</artifactId>
    <version>1.0</version>
</dependency>

<dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
</dependency>

<dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>persistence-api</artifactId>
    <version>1.0</version>
</dependency>

<dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
</dependency>

<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
</dependency>

<dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>persistence-api</artifactId>
    <version>1.0</version>
</dependency>

<dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
</dependency>
```

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
</dependency>

<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet</groupId>
                           <artifactId> servlet-api</artifactId>
                           <version>2.5</version>
                         </dependency>

<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>1.0</version>
</dependency>

  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet.jsp</groupId>
                           <artifactId> jsp-api</artifactId>
                           <version>2.1</version>
                         </dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet</groupId>
                           <artifactId> servlet-api</artifactId>
                           <version>2.5</version>
                         </dependency>

<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>1.0</version>
</dependency>

  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet.jsp</groupId>
                           <artifactId> jsp-api</artifactId>
                           <version>2.1</version>
                         </dependency>

<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-annotations-ejb3</artifactId>
  <version>4.2.3.GA</version>
</dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet</groupId>
                           <artifactId> servlet-api</artifactId>
                           <version>2.5</version>
                         </dependency>

<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>1.0</version>
</dependency>

  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet.jsp</groupId>
                           <artifactId> jsp-api</artifactId>
                           <version>2.1</version>
                         </dependency>

<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-annotations-ejb3</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
    <version>3.2.4.sp1</version>
  </dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet</groupId>
                           <artifactId> servlet-api</artifactId>
                           <version>2.5</version>
                         </dependency>

<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>1.0</version>
</dependency>

  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet.jsp</groupId>
                           <artifactId> jsp-api</artifactId>
                           <version>2.1</version>
                         </dependency>

<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-annotations-ejb3</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate</artifactId>
    <version>3.2.4.sp1</version>
  </dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>

  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet</groupId>
                           <artifactId> servlet-api</artifactId>
                           <version>2.5</version>
                         </dependency>

  <dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>persistence-api</artifactId>
    <version>1.0</version>
  </dependency>           <dependency>
                           <groupId>jboss</groupId>
                           <artifactId>jboss-common</artifactId>
                           <version>4.2.3.GA</version>
                         </dependency>

  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>           <dependency>
                           <groupId>javax.servlet.jsp</groupId>
                           <artifactId>jsp-api</artifactId>
                           <version>2.1</version>
                         </dependency>

<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-annotations-ejb3</artifactId>
  <version>4.2.3.GA</version>
</dependency>           <dependency>
                           <groupId>org.hibernate</groupId>
                           <artifactId>hibernate</artifactId>
                           <version>3.2.4.sp1</version>
                         </dependency>

  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
    <version>3.2.4.sp1</version>
  </dependency>
```

# Dependency Chaos

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-ejb-api</artifactId>
  <version>4.2.3.GA</version>
</dependency>
```

```
  <dependency>
    <groupId>javax.ejb</groupId>
    <artifactId>ejb-api</artifactId>
    <version>3.0</version>
  </dependency>
```

```
<dependency>
  <groupId>javax.persistence</groupId>
  <artifactId>persistence-api</artifactId>
  <version>1.0</version>
</dependency>
```

```
  <dependency>
    <groupId>javax.transaction</groupId>
    <artifactId>jta</artifactId>
    <version>1.1</version>
  </dependency>
```

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-annotations-ejb3</artifactId>
  <version>4.2.3.GA</version>
</dependency>
```

```
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
    <version>3.2.4.sp1</version>
  </dependency>
```

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jbosssx</artifactId>
  <version>4.2.3.GA</version>
</dependency>
```

```
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId> servlet-api</artifactId>
    <version>2.5</version>
  </dependency>
```

```
<dependency>
  <groupId>jboss</groupId>
  <artifactId>jboss-common</artifactId>
  <version>4.2.3.GA</version>
</dependency>
```

```
  <dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
  </dependency>
```

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate</artifactId>
  <version>3.2.4.sp1</version>
</dependency>
```

```
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
    <version>3.2.4.sp1</version>
  </dependency>
```

# One JAR to rule them all!

```
<dependency>
  <groupId>javax</groupId>
  <artifactId>javaee-api</artifactId>
  <version>7.0</version>
</dependency>
```

# Dependencies Updates

- Maven 2.2.1
  - Java EE 5
  - Spring 2.5.6
  - Ehcache 1.5.0
  - GWT 1.6.4
  - Maven 3.2.3
  - Java EE 7
  - Spring 4.1.4
  - Ehcache 2.8.3
  - GWT 2.0.0
- 

# Dependencies Resolution

- Try to keep with the Standards
- Introduce additional libraries only if absolutely necessary
- Easier to move on to new versions

# Classloading Hell

- **ClassNotFoundException**
- **NoClassDefFoundError**
- **ClassCastException**
- **JBoss dreaded UnifiedClassLoader**

# Classloading Hell

- Servers change their libraries
- These can clash with your Application libraries
- Need to tune dependencies or rewrite code

# Simplify Packaging

- **EAR to WAR**
- Other packages too: **SAR, RAR**
- **Enough for the majority of Projects**

A dark, atmospheric forest scene. In the center, a deer stands in profile, facing right, its silhouette stark against a bright, hazy background. The background shows tall trees and a misty sky. The overall mood is mysterious and contemplative.

# Keep on Walking

# Easy to use Goodies

- CDI
- No more Local Interfaces
- Singletons
- Simplified JPA
- Validations

```
public static <T> T getLocalBean(final Class<T> klass) {  
    try {  
        LocalBinding localBinding =  
            klass.getAnnotation(LocalBinding.class);  
        if (localBinding == null) {  
            throw new BeanNotFoundException("...");  
        }  
        return (T) initialContext.lookup(localBinding.jndiBinding());  
    } catch (Exception e) {  
        throw new BeanNotFoundException("...");  
    }  
}
```

A dark, atmospheric forest scene featuring a large stag standing in the center. The stag is silhouetted against a bright, hazy background, its antlers reaching upwards. The forest is dense with tall trees, their trunks and branches silhouetted against the light. The overall mood is mysterious and ethereal.

```
LocalBinding localBinding =  
klass.getAnnotation(LocalBinding.  
}  
}  
}
```

The background of the image is a dark, moody forest scene. In the center, there is a silhouette of a horse standing in a clearing. The forest is filled with tall trees, their trunks and branches silhouetted against a lighter sky. The overall atmosphere is mysterious and dramatic.

LocalBinding localBinding =  
klass.getAnnotation(LocalBinding.

}

}

}

# @Inject

```
@Stateless  
@Local(UserBusiness.class)  
public class UserBusinessBean implements UserBusiness {  
    ...  
}
```

The background of the slide is a dark, moody photograph of a forest. A stag is standing in the center, its silhouette clearly defined against a bright, hazy sky. The forest floor is covered in fallen leaves and branches. The overall atmosphere is mysterious and ethereal.

@Stateless

public class UserBusinessBean implements UserBusiness {

...

}

The background of the slide is a dark, moody photograph of a forest. A large stag with prominent antlers is silhouetted against a lighter area in the center, appearing to stand in a clearing. The forest is filled with tall trees and dense foliage, creating a sense of depth and mystery.

**@Stateless**

**public class UserBusinessBean**

...

}

{

```
public class ModuleListener {  
    private static ModuleListener moduleListener;  
    private static ModuleBusiness moduleBusiness;  
  
    private ModuleListener() {  
        moduleBusiness =  
            BeanFactory.getLocalBean(ModuleBusinessBean.class);  
    }  
  
    public static ModuleListener getInstance() {  
        if (moduleListener == null) {  
            moduleListener = new ModuleListener();  
        }  
        return moduleListener;  
    }  
}
```

A dark, atmospheric forest scene featuring a large stag standing in the center. The stag is silhouetted against a bright, hazy background, possibly a sunrise or sunset. The forest is dense with tall trees, their trunks and branches silhouetted against the light. The overall mood is mysterious and ethereal.

```
public class
{
    private static
    private static
    private
    BeanFactory.getLocalBean(ModuleBusinessBean.
}
public static
if
return
```

```
public class
    private static
    private static
        private
    }
    BeanFactory<? extends LocalBean> ModuleBusinessBean.
public static
if
return
```

# @Singleton

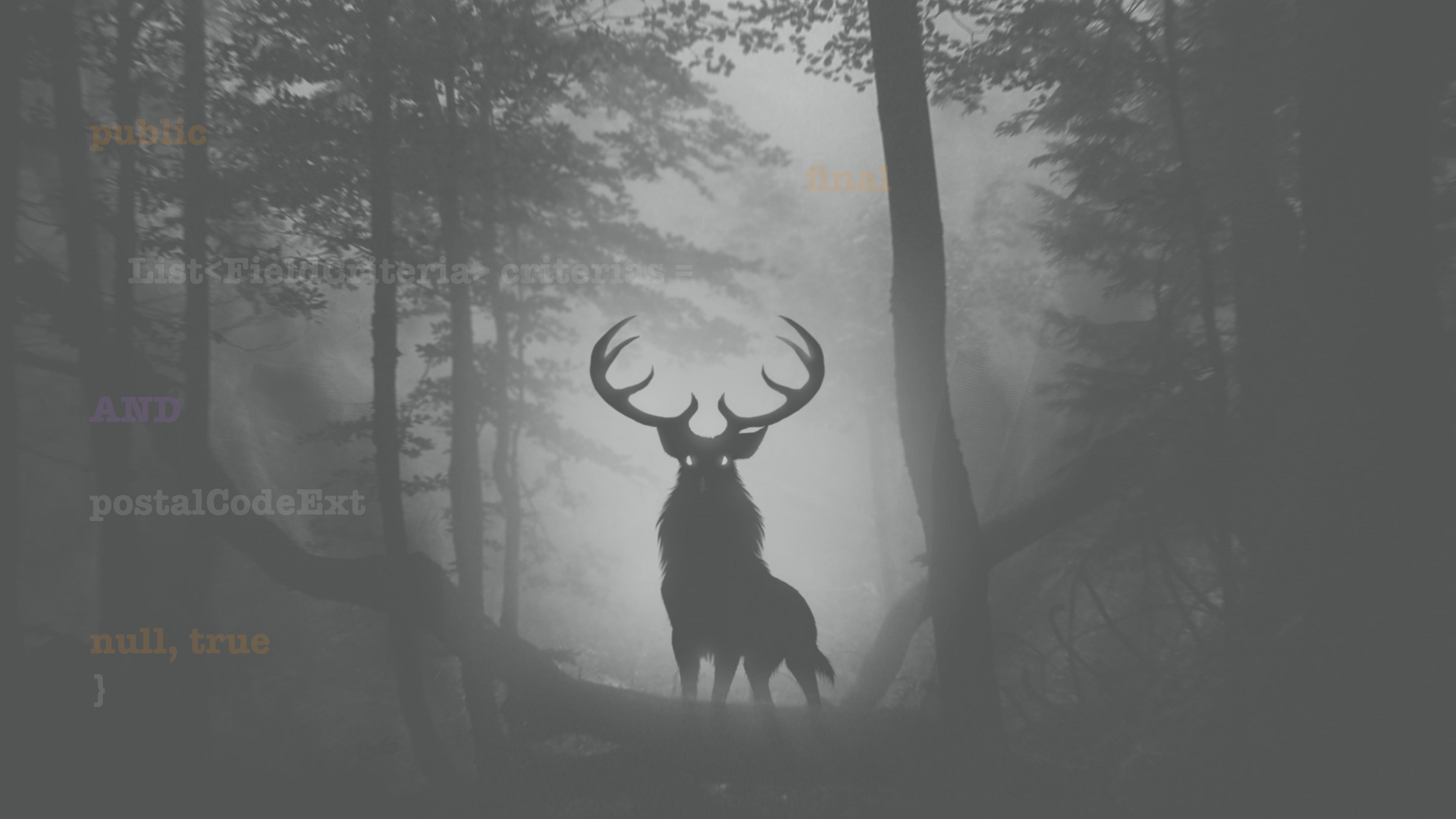
A dark, atmospheric forest scene with a stag silhouette in the center.

```
@Singleton
@Lock(LockType.READ)
public class ModuleListener {

    @EJB
    private ModuleBusiness moduleBusiness;

}
```

```
public List<PostalCode> findAddress(final String postalCode,  
final String postalCodeExt) {  
  
    List<FieldCriteria> criterias = new ArrayList<FieldCriteria>();  
  
    criterias.add(new FieldCriteria("code", EQUALS, postalCode,  
AND));  
    criterias.add(new FieldCriteria("extension", EQUALS,  
postalCodeExt, AND));  
  
    return postalCodesDAO.findAll(PostalCode.class, null, criterias,  
null, true);  
}
```

The background of the image is a dark, moody forest scene. In the center, there is a large, dark silhouette of a stag with prominent antlers, standing on a path or clearing. The forest is filled with tall trees, their trunks and branches silhouetted against a lighter sky. The overall atmosphere is mysterious and ethereal.

public

final

List<FieldCriteria> criterias =

AND

postalCodeExt

null, true

}

# JPA Criteria API

public

final

List<F<sup>7</sup>

AND

postalCodeExt

null, true

}

```
public List<PostalCode> findAddress(final String postalCode,  
final String postalCodeExt) {  
  
CriteriaBuilder cb = em.getCriteriaBuilder();  
CriteriaQuery<PostalCode> cq = cb.createQuery(PostalCode.class);  
  
Root<PostalCode> root = cq.from(PostalCode.class);  
cq.select(root).where(  
    cb.equal(root.get(PostalCode_.code), postalCode),  
    cb.equal(root.get(PostalCode_.extension), postalCodeExt));  
  
TypedQuery<PostalCode> query = em.createQuery(cq);  
return query.getResultList();  
}  
}
```

```
Query query = manager.createNativeQuery(  
    "BEGIN PAC_GS_COMPENSATION_CHAMBER.  
    execute_compensation_chamber(  
    :protocol, :eligibility, :identifier, :userId); END;");  
  
query.setParameter("protocol", protocol);  
query.setParameter("eligibility", eligibility);  
query.setParameter("identifier", identifier);  
query.setParameter("userId", userId);  
query.executeUpdate();
```

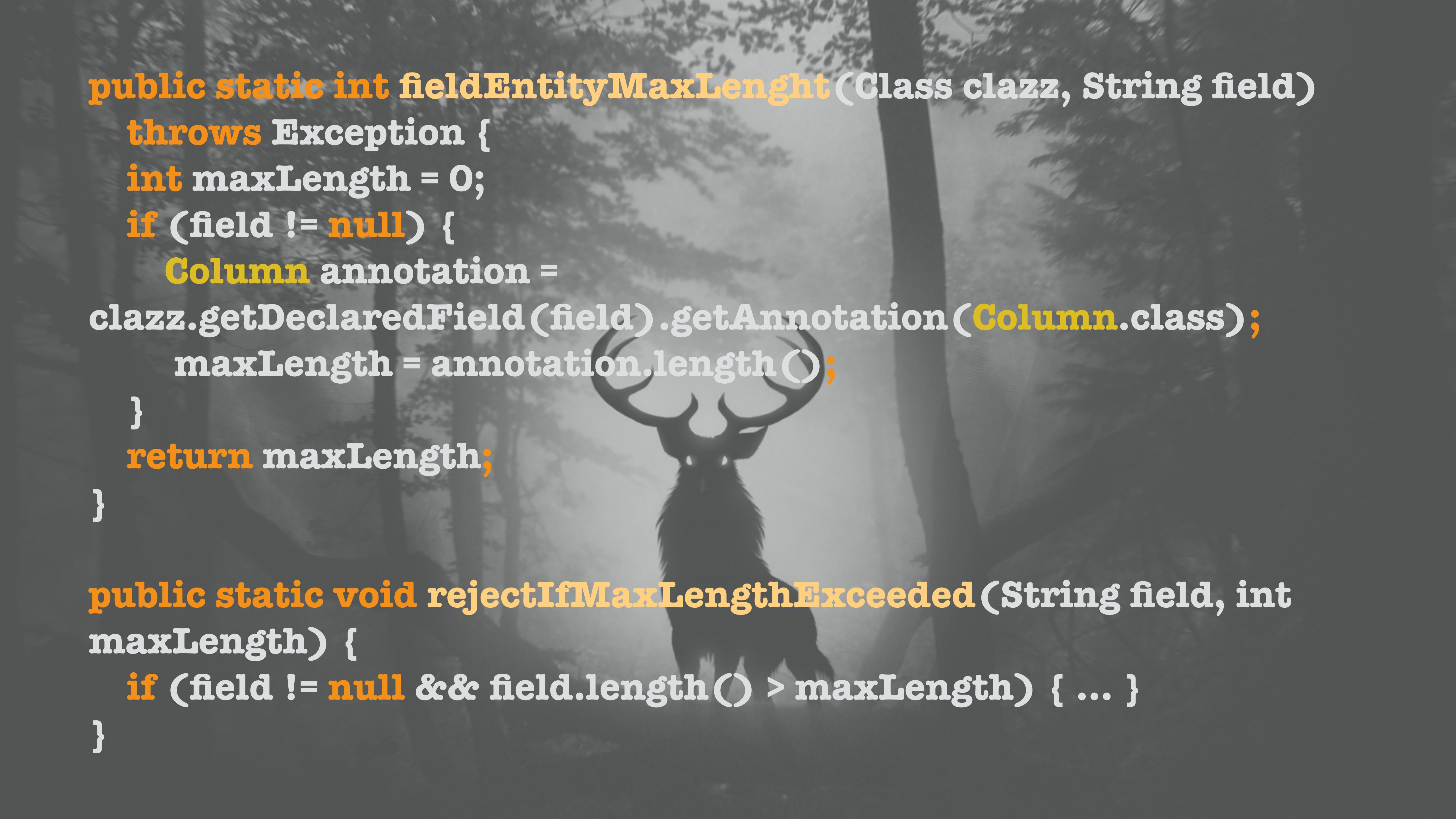
```
Query query =  
    execute_compensation_chamber(  
        :protocol, :eligibility, :identifier, :userId); END;"  
  
query.setParameter(  
query.setParameter(  
query.setParameter(  
query.setParameter(  
query.executeUpdate();
```



```
Query query =  
    execute_compensation_chamber(  
        :protocol, :eligibility, :identifier, :userId); END;"
```

**@NamedStoredProcedureQuery**

```
query.setParameter(  
query.setParameter(  
query.setParameter(  
query.executeUpdate();
```



```
public static int fieldEntityMaxLength(Class clazz, String field)
throws Exception {
int maxLength = 0;
if (field != null) {
    Column annotation =
    clazz.getDeclaredField(field).getAnnotation(Column.class);
    maxLength = annotation.length();
}
return maxLength;
}
```

```
public static void rejectIfMaxLengthExceeded(String field, int
maxLength) {
    if (field != null && field.length() > maxLength) { ... }
}
```

A dark, atmospheric forest scene featuring a stag silhouette standing in the center. The background consists of tall, thin trees with sparse foliage, creating a misty and mysterious atmosphere.

```
public static int  
        clazz.getDeclaredField(field).getAnnotation(  
            maxLength = annotation.length()  
    }  
  
}  
  
public static void  
maxLength) {  
  
}
```

The background of the slide is a dark, moody photograph of a forest. In the center, a deer stands facing away from the viewer, its silhouette clearly defined against a lighter path or opening in the trees. The forest is dense with tall, thin trunks and intricate branch patterns.

**public static int**

**clazz.getDeclaredFields();  
maxLength = annotation.getAnnotation(maxLength.class).  
maxLength;**

**}**

**}**

**public static void**

**maxLength) {**

**}**

**@NotNull  
@Max**

A black and white photograph of a white unicorn standing in a dense forest. The unicorn is positioned in the center of the frame, facing towards the right. Its long, flowing mane and tail are clearly visible. The background consists of many tall, thin trees, creating a sense of depth and a magical atmosphere.

The End of a Journey

# Was it worth it?

- Removed Code
- Code is growing at a slower pace
- Easier to maintain

# Was it worth it?

- Productivity (1 minute deploy time vs 5)
- Performance JVM (10 to 25% increase)
- Developers Happy :)

# Setbacks

- Started in May 2013, finished October 2015
- Underestimate the Job
- Business demands priorities

# Setbacks

- Vendors moved slowly to EE '7
- Blocked by Wildfly bugs
- Missed dates multiple times
- Not enough confidence in tests

# New Horizons

- Jumped straight away to Java 8
- Ready to face new Java EE versions
- Easier to upgrade or change Server

# New Horizons

- Proper Testing frameworks (**Arquillian**)
- Keep simplifying the code (**JMS, Batch**)
- Adopt new specifications (**Websockets**)

A black and white photograph of a white horse standing in a dense forest. The horse is positioned in the center-left of the frame, facing towards the right. The background is filled with tall, thin trees, their trunks creating vertical lines that converge towards the top of the image. The ground is covered with fallen leaves.

# Thank you for Attending!

**twitter:**

**@radcortez**

**mail:**

**radcortez@yahoo.com**

**blog:**

**<http://www.radcortez.com>**