

JAVA CLASSLOADERS

SIMON MAPLE

@SJMAPLE

 ZEROTURNAROUND

ABOUT ME



SIMON MAPLE
@SJMAPLE



BACKGROUND: **J**REBEL

- Hooked into class loading on the JVM level
- Integrated with the class loading mechanism in more than 20 different servers
- Solved hundreds of issues connected to class loading
- Learned a lot more about class loaders than we wanted to 😊

PLAN

- Basics
- What is class loading?
- How was it meant to work?
- Problems and solutions
- How do class loaders leak?



BASICS

JAVA.LANG.**CLASSLOADER**

```
public abstract class ClassLoader {  
    public Class loadClass(String name);  
    protected Class defineClass(byte[] b);  
    public URL getResource(String name);  
    public Enumeration getResources(String name);  
    public ClassLoader getParent()  
};
```

EXAMPLE 0

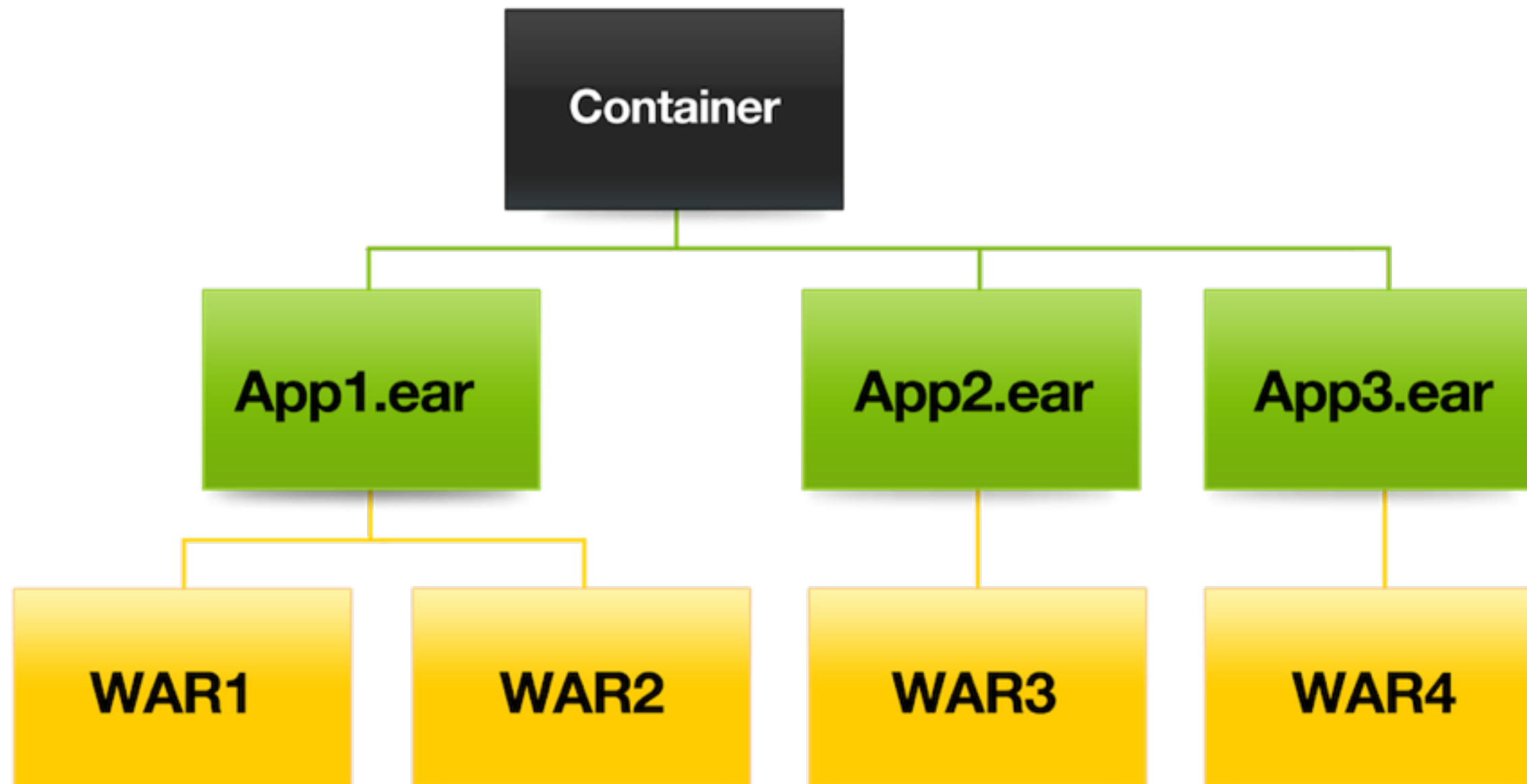
```
public class A {  
    public void doSmth() {  
        B b = new B();  
        b.doSmthElse();  
    }  
}
```


EXAMPLE 0

```
public class A {  
    public void doSmth() {  
        B b = new B();  
        b.doSmthElse();  
    }  
}
```



Causes a call to
`A.class.getClassLoader().loadClass("B");`



CL HIERARCHY

INTENTIONS

- Class loaders have a parent class loader
- The parent is usually consulted first
- However in a Java EE web module local classes are searched first
- Each WAR module of EAR gets a classloader



PROBLEMS

NOT ENOUGH CLASSES

- ClassNotFoundException
- NoClassDefFoundError

TROUBLESHOOTING

- IDE class lookup (Ctrl+Shift+T in Eclipse)
- `find *.jar -exec jar -tf '{}' \; | grep MyClass`
- `URLClassLoader.getUrls()`
- Container specific logs

ALIEN CLASSES

- IncompatibleClassChangeError
- AbstractMethodError
- NoSuch(Method | Field)FoundError
- ClassCastException, IllegalAccessException

TROUBLESHOOTING

- `-verbose:classpath`
- `ClassLoader.getResource()`
- `javap -private MyClass`

MORE **CLASSES** THAN **YOU** NEED

- LinkageError (class loading constraints violated)
- ClassCastException, IllegalAccessException

TROUBLESHOOTING

- -verbose:class
- `ClassLoader.getResource()`

Shared ClassLoader

Util3 Factory3

new Util3()

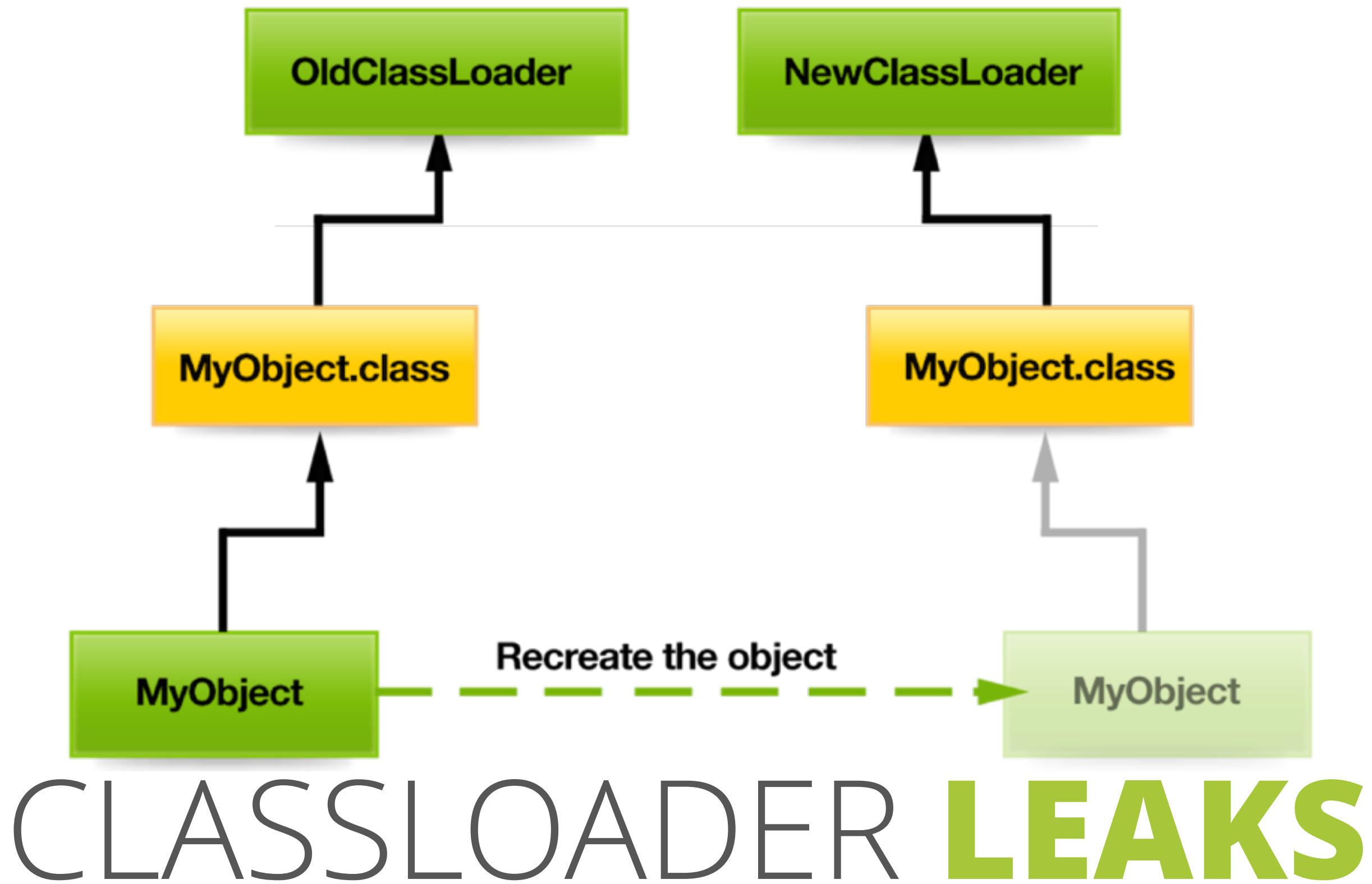


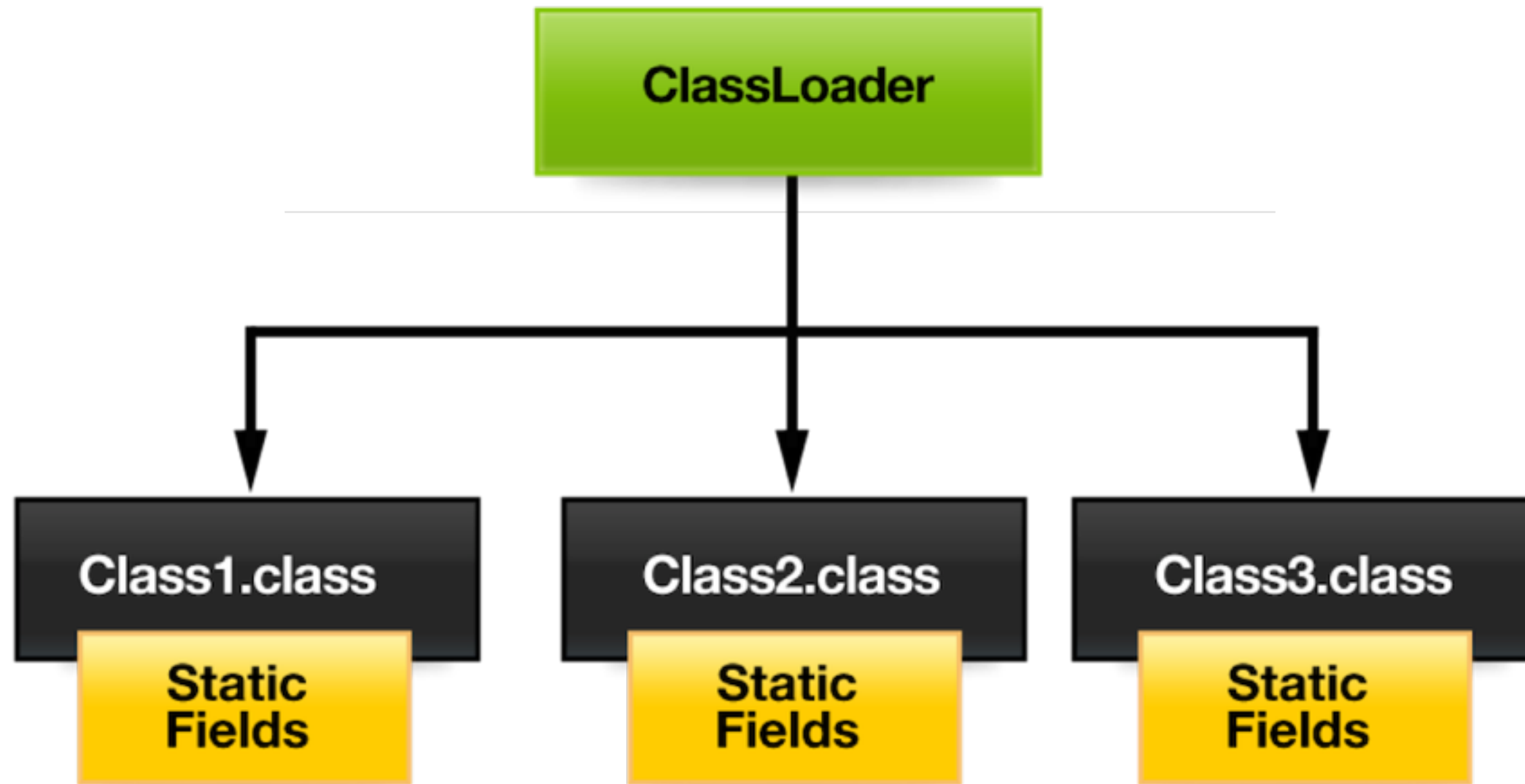
Factory3.instanceUntyped();

Util3 Test3

Web ClassLoader

Util3 u = (Util3) Factory3.instanceUntyped();





CLASSLOADER **LEAKS**

TAKE AWAY

- Validate assumptions
- Leaking ClassLoader is easy
- Isolation works for processes
- **JREBEL** rocks!



HOPE YOU ENJOYED
THE PRESENTATION

IF NOT, AND YOU WOULD LIKE TO
CLAIM YOUR 60 WASTED MINUTES BACK,
PLEASE REDEEM VIA

JRebel