

# Polyglot Java

Using multiple languages in your Java code



# @honzam399

- Technical director at Cake Solutions
- Author of Pro Spring, Pro Spring 2.5 and other books & articles
- Contributor to Spring Extensions, Specs2 Spring, Scalad, Spock Spring Integration
- Editor of the Open Source Journal
- Road racer for the Oxford City Road Club

# Polyglot Java

- Java platform is not just the Java language
- We have dynamically typed languages  
(Clojure, Groovy, JRuby, Jython)
- We have statically typed languages  
(Java, Scala)

# Polyglot Java

- Java platform is not just the Java language
- We have dynamically typed languages  
(Clojure, Groovy, JRuby, Jython)
- We have statically typed languages  
(Java, Scala)
- Because th

# Polyglot Java

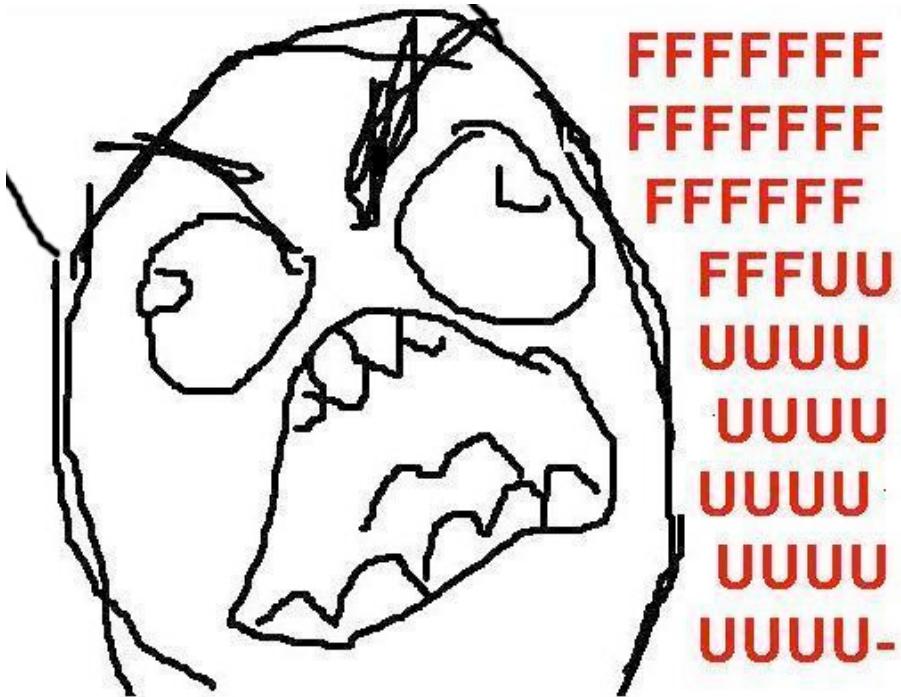
- Java platform is not just the Java language
- We have  
(Clojure,
- We have  
(Java, Scal
- Because th

You need to restart your computer. Hold down the Power button for several seconds or press the Restart button.

Veuillez redémarrer votre ordinateur. Maintenez la touche de démarrage enfoncée pendant plusieurs secondes ou bien appuyez sur le bouton de réinitialisation.

Sie müssen Ihren Computer neu starten. Halten Sie dazu die Einschalttaste einige Sekunden gedrückt oder drücken Sie die Neustart-Taste.

コンピュータを再起動する必要があります。パワーボタンを数秒間押し続けるか、リセットボタンを押してください。

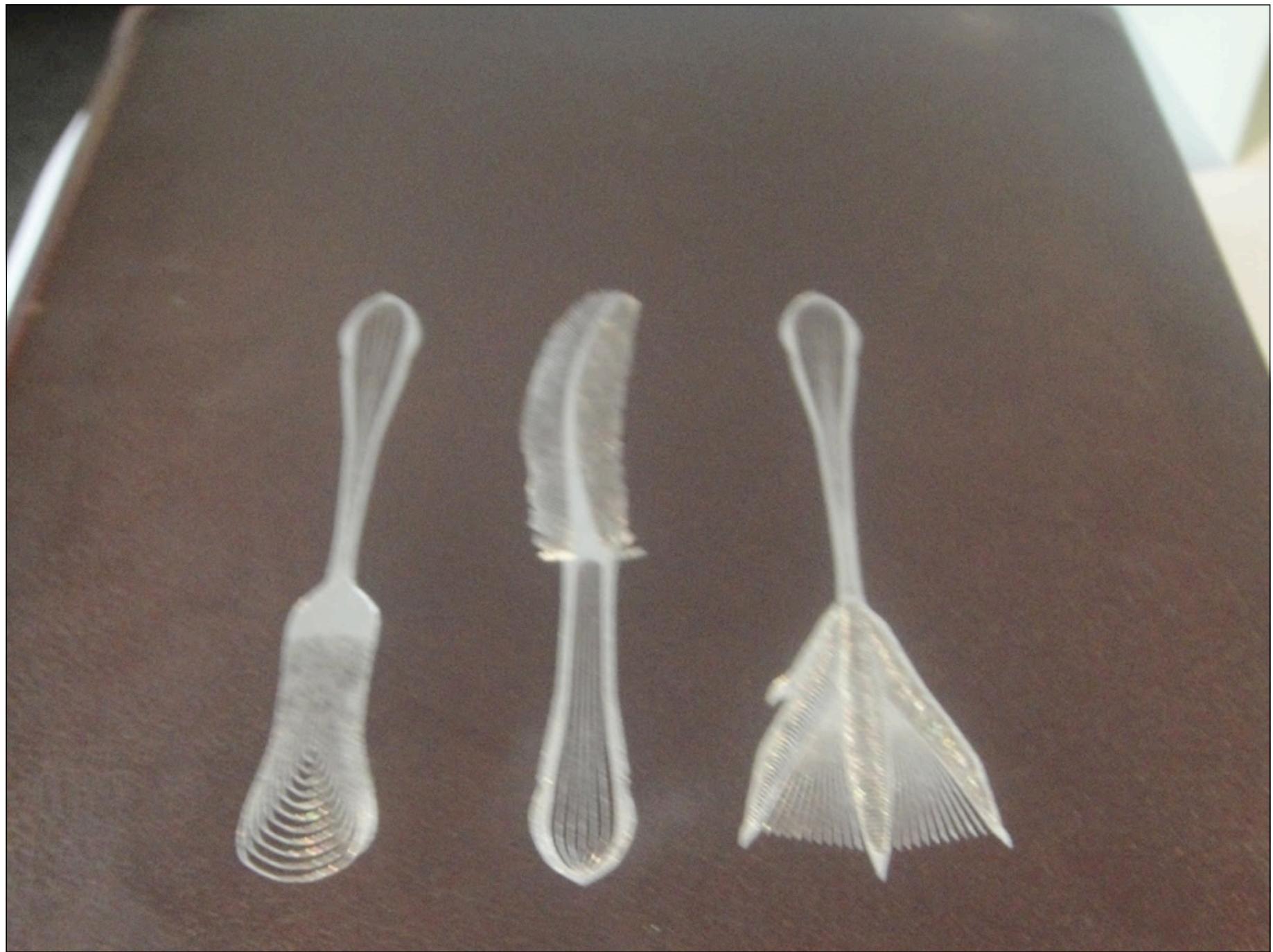


FFFFFFF  
FFFFFFF  
FFFFFFF  
FFFUUU  
UUUU  
UUUU  
UUUU  
UUUU  
UUUU-

# DOPRDELE PRÁCE!

Only the right language allows us to express the right sentiment!





## **Microsoft MS-DOS 6 Setup**

---

---

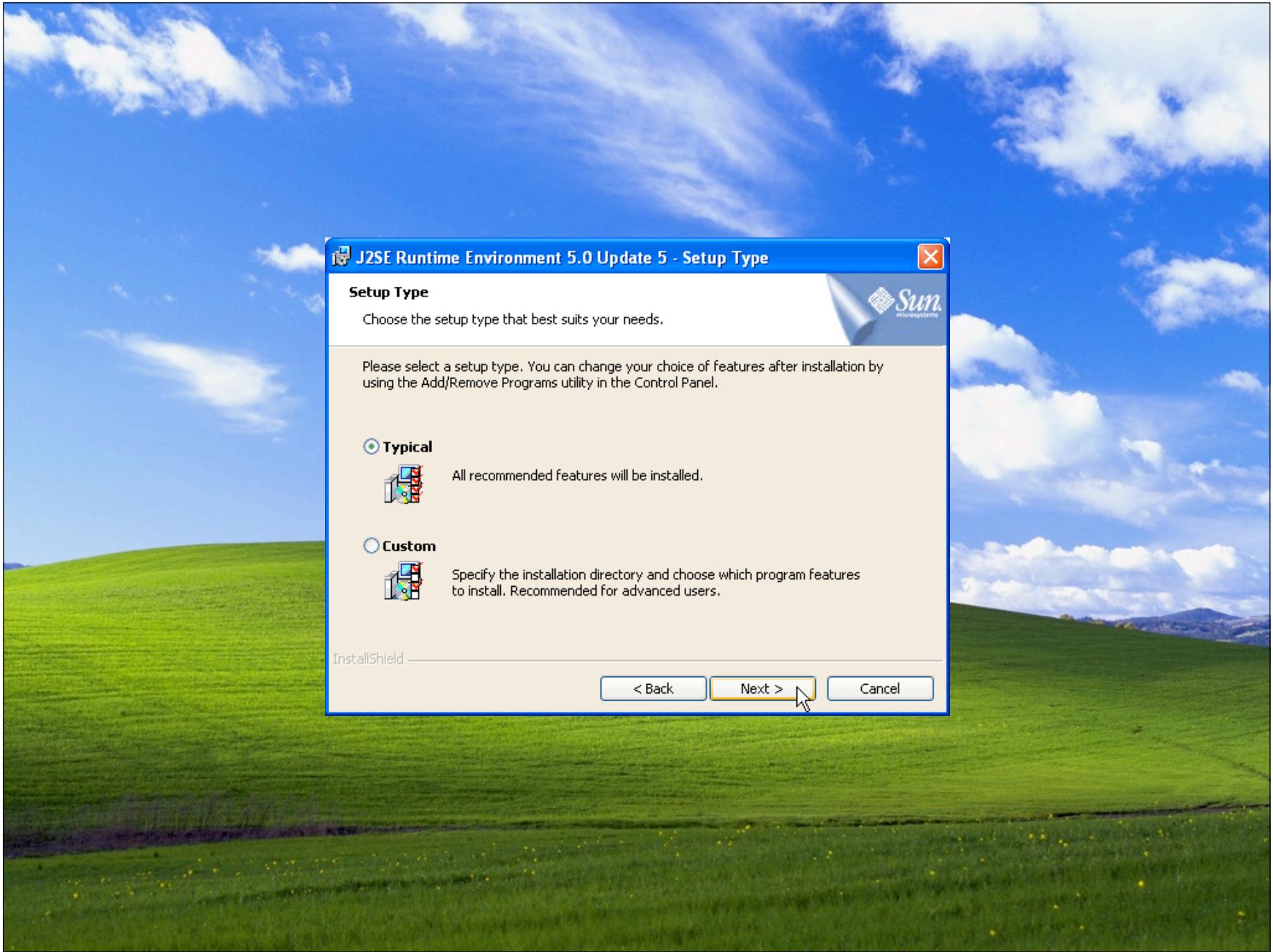
**Double your hard disk with DoubleSpace.** MS-DOS 6 gives you a safe, easy way to increase your disk capacity by integrating data compression into the operating system.

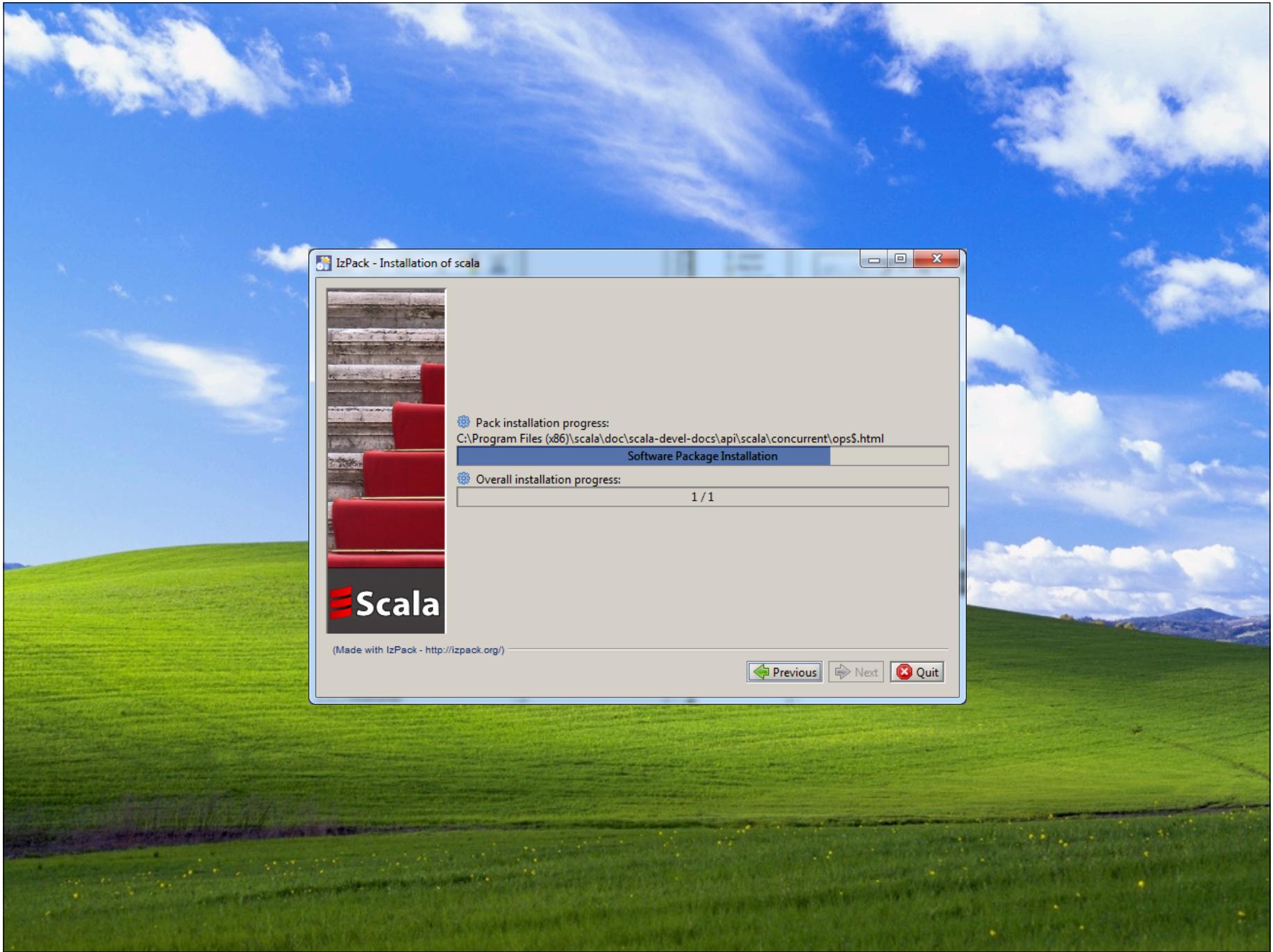
You can double your hard disk by typing DBLSPACE at the command prompt as soon as you complete this setup program.

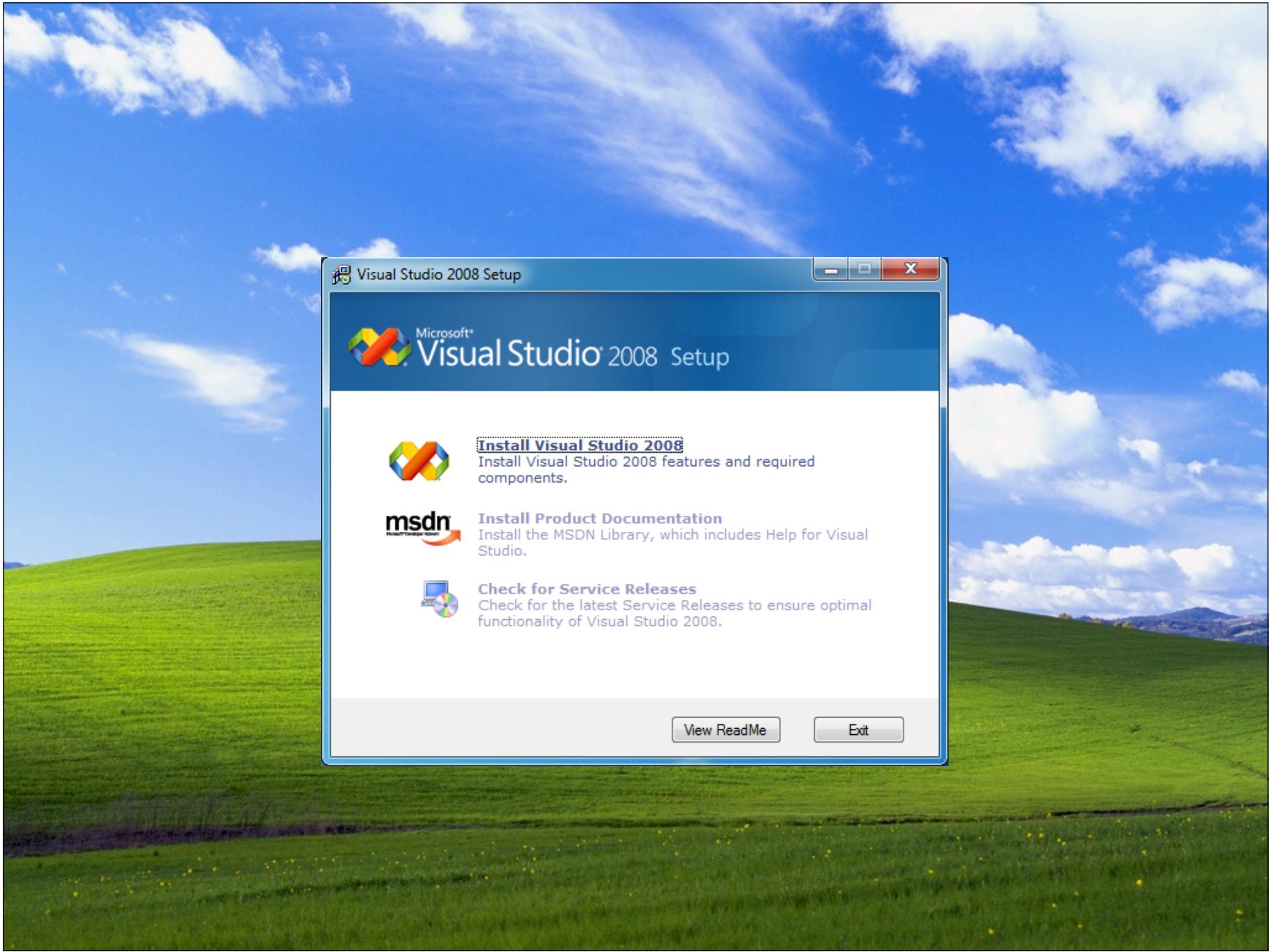
25% complete













```
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ session1 ---
[INFO] No sources to compile
[INFO]
[INFO] --- gmaven-plugin:1.4:testCompile (default) @ session1 ---
[INFO] No sources found to compile
[INFO]
[INFO] --- maven-scala-plugin:2.15.3-SNAPSHOT:testCompile (default) @ session1 ---
[INFO] Checking for multiple versions of scala
[INFO] includes = [**/*.scala,**/*.java,]
[INFO] excludes = []
[WARNING] No source files found.
[INFO]
[INFO] --- maven-surefire-plugin:2.7.2:test (default-test) @ session1 ---
[INFO] Surefire report directory: /Users/janmachacek/Training/pvjm/project/session1/target/surefire-reports
```

---

## T E S T S

---

There are no tests to run.

Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

```
[INFO]
[INFO] --- clojure-maven-plugin:1.3.6:test (test) @ session1 ---
```

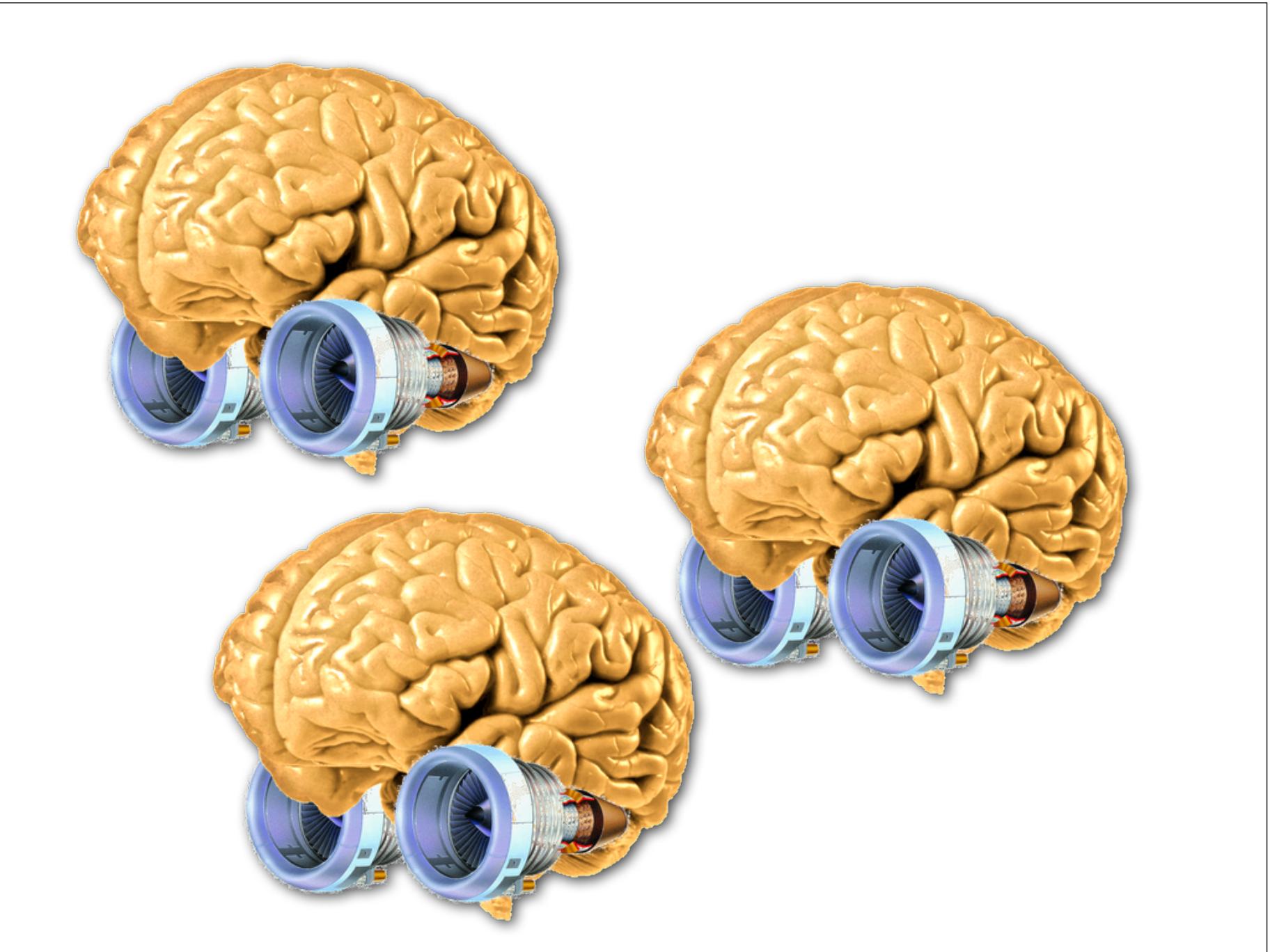
Testing com.theoryinpractise.clojure.testrunner

Ran 0 tests containing 0 assertions.

0 failures, 0 errors.

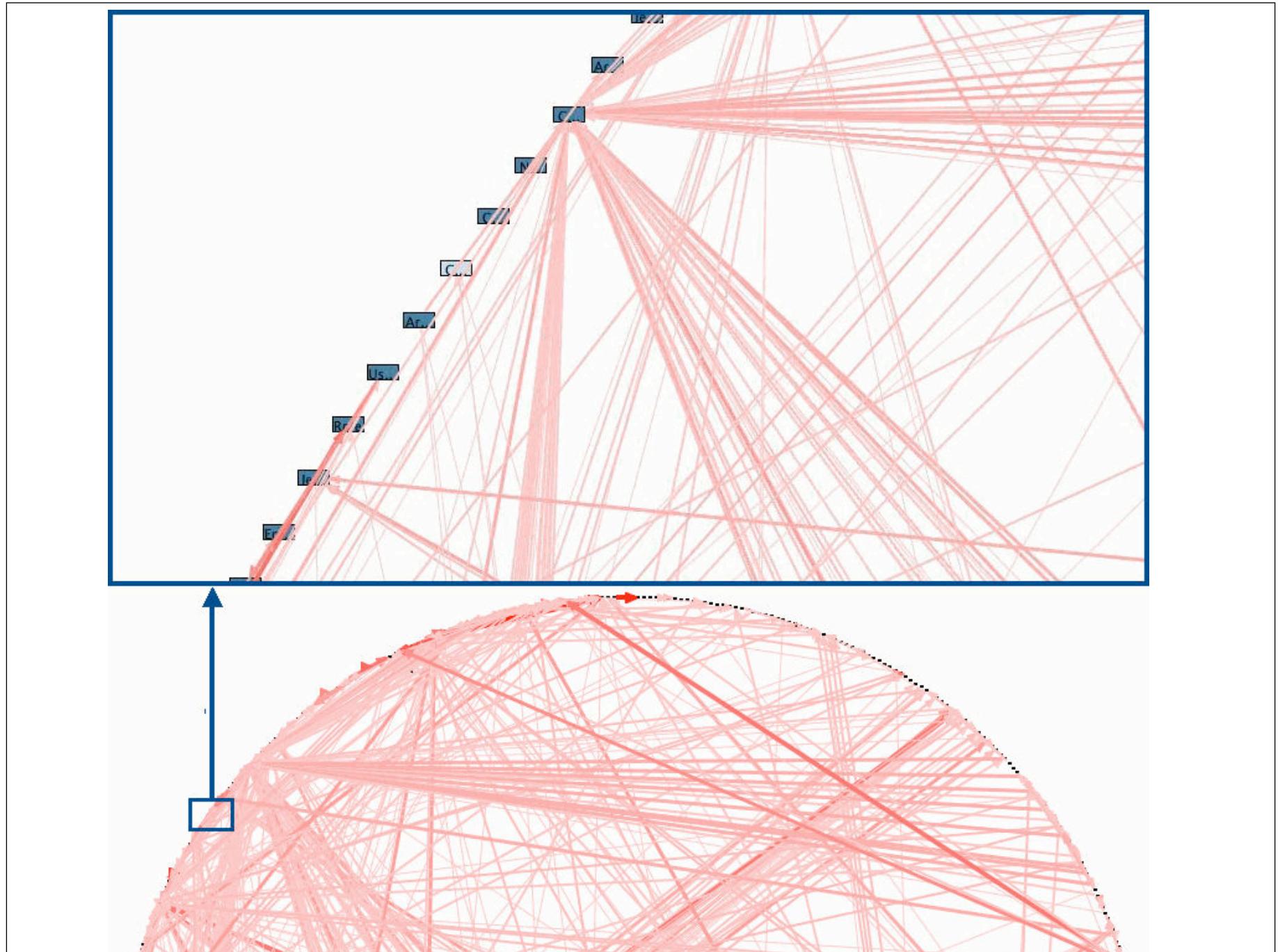
```
[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ session1 ---
[INFO] Building jar: /Users/janmachacek/Training/pvjm/project/session1/target/session1-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.927s
[INFO] Finished at: Tue May 01 12:01:35 BST 2012
[INFO] Final Memory: 8M/81M
[INFO] -----
```

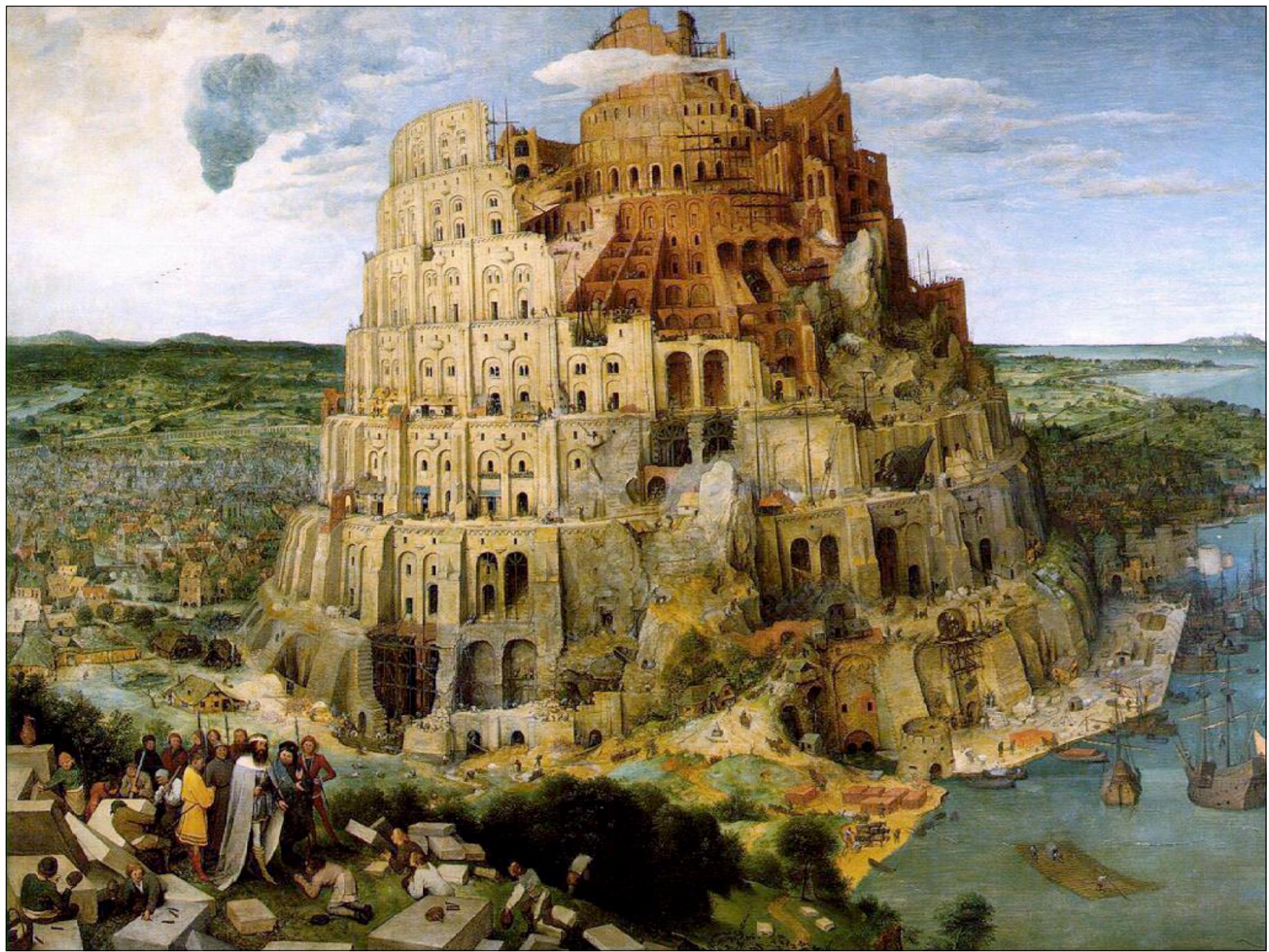
janmachacek@janmachacek:~/Training/pvjm/project/session1 (master)\$ █



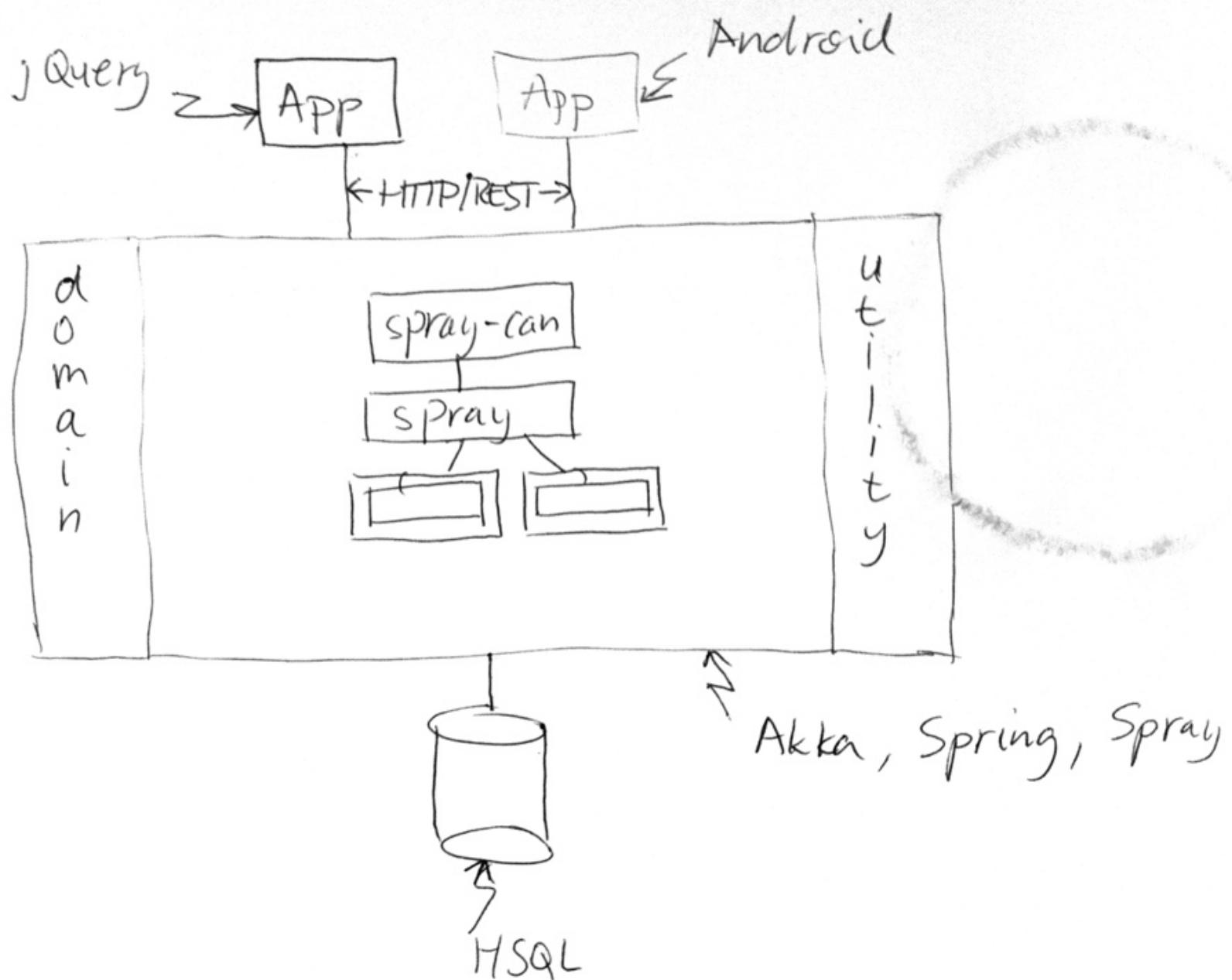
# We Can Do It!











# SESSION 2

src

main

groovy

...

java

...

resources

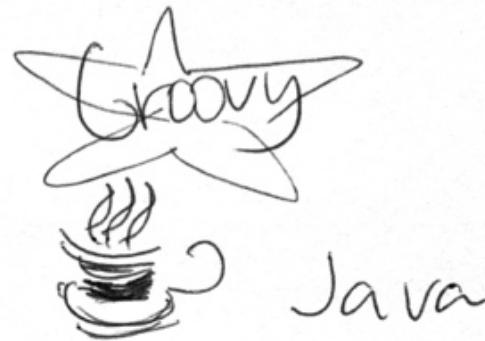
...

scala

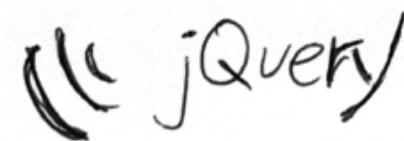
...

webapp

...



Java



```
object Main extends App {  
    ...  
    new Boot(system)  
    new SprayCanBoot(system)  
    ...  
}
```

```
class Boot(system: ActorSystem) {  
    val spring = ...  
    val application = ...  
  
    Await.ready(  
        spring ? Start(), timeout.duration)  
    Await.ready(application ? Start(), ...)  
}
```

```
class SpringContextActor extends Actor {
    protected def receive = {
        case Start() =>
            val ac = new GenericXmlApplicationContext("...")
            context.actorOf(
                Props(new BeanLookupActor(ac)),
                name = "bean lookup")
            sender ! Started()
        case Stop() =>
            context.children.foreach(- ! Stop())
    }
}
```

```
class Application Actor extends Actor {
```

```
protected def receive = {
```

```
case Start() =>
```

```
    context.actorOf(
```

```
        Props(new Entity Actor),  
        name = "entity")
```

```
// same for User Actor
```

```
    sender ! Started()
```

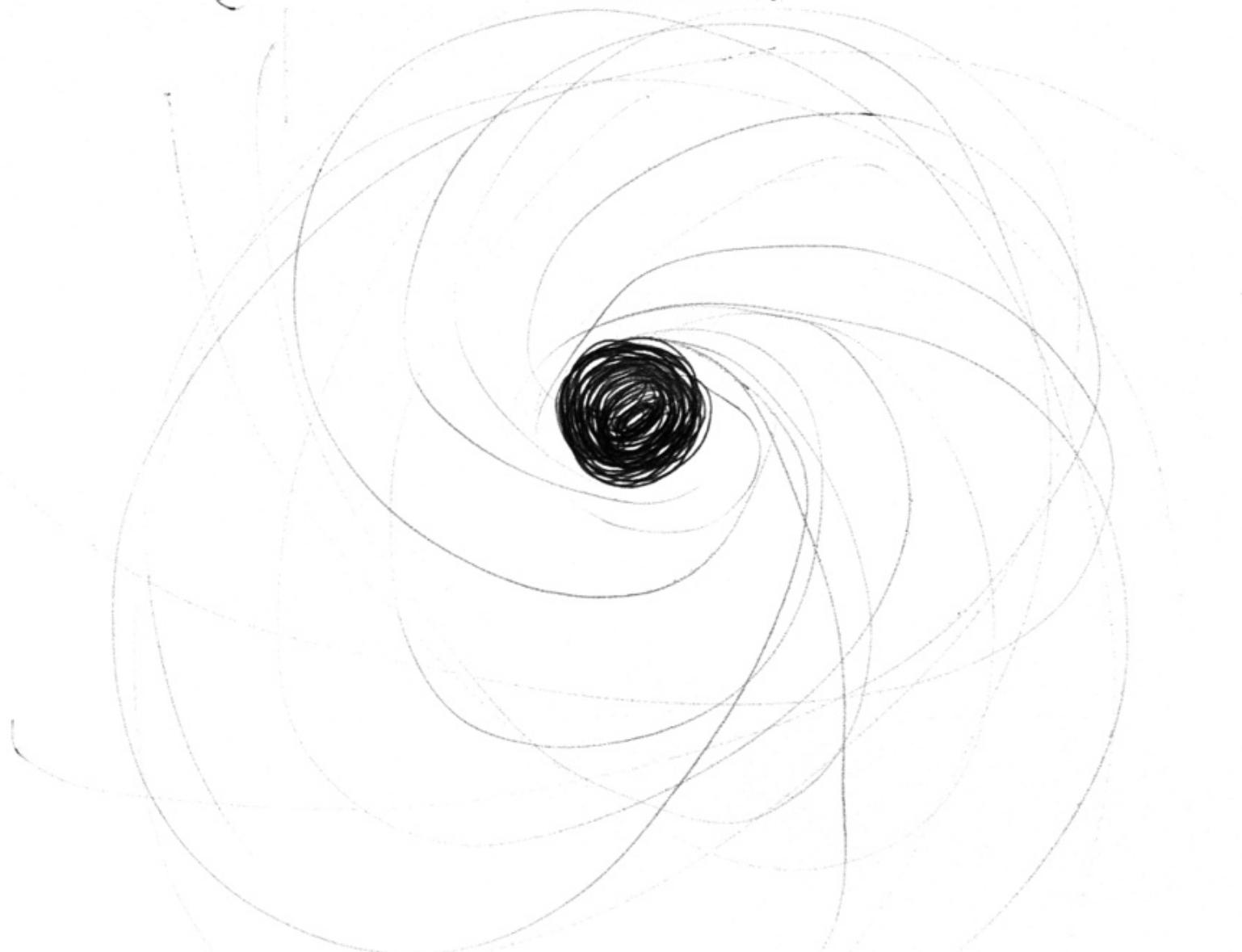
```
case Stop() =>
```

```
}
```

```
}
```

```
@Configurable  
class EntityActor extends Actor {  
    @Autowired  
    var sessionFactory: SessionFactory = -  
    protected def receive = {  
        case GetEntity(clazz, id) =>  
            ...  
        case CountEntities(clazz) =>  
            ...  
        case ListEntities(clazz, first, max) =>  
            ...  
    }  
}
```

The null hole



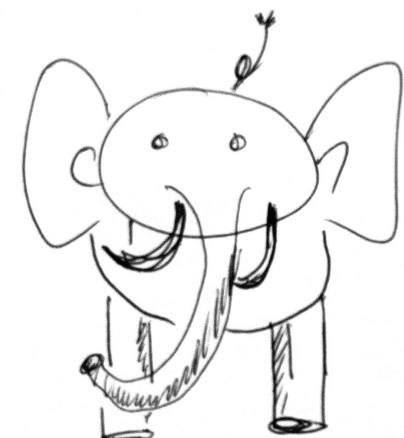
```
case EntityActor extends Actor {  
    ...  
    protected def receive = {  
        case GetEntity(clazz, id) =>  
            val e = sessionFactory.getCurrentSession.  
                get(clazz, id)  
            if (e == null) {  
                sender ! None  
            } else {  
                sender ! Some(e)  
            }  
        ...  
    }  
}
```

```
for (i < 1 to count) {  
    val user = new User  
    user.setUsername("...")  
    ...  
    session.getCurrentSession().saveOrUpdate(user)  
}
```

```
val reader = new UserReader ↳ Groovy!  
val importOp =  
for {  
    input ← bufferFile(new File(source))  
    op ← io { reader.read(input) } ≫= { users ⇒  
        io { users.foreach(sessionFactory.  
            getCurrentSession.  
            saveOrUpdate(-) }  
    }  
    - ← closeReader(input)  
} yield op  
importOp.unsafePerformIO
```

```
val reader = new UserReader  $\Leftarrow$  Groovy!  
val importOp =  
for {  
    input  $\leftarrow$  bufferFile(new File(source))  
    op  $\leftarrow$  io { reader.read(input) }  $\gg=$  { users  $\Rightarrow$   
        io { users.foreach(sessionFactory.  
            getCurrentSession.  
            saveOrUpdate(-) }  
    }  
    -  $\leftarrow$  closeReader(input)  
} yield op
```

```
importOp.unsafePerformIO
```



```
trait UserService extends Directives  
    with DefaultMarshallers  
    with CustomMarshallers {
```

...

```
val user Service = {
```

```
get {
```

```
    path("user/list") {
```

```
        completeWith {
```

```
            val actor = ...
```

```
Await.result((actor ? ListEntities(classOf[User])).
```

```
mapTo[List[User]])timeout.duration)
```

```
}
```

```
} ~
```

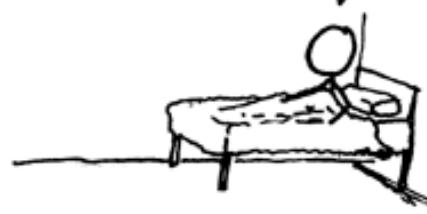
```
path("...") ...
```

```
}
```

```
}
```

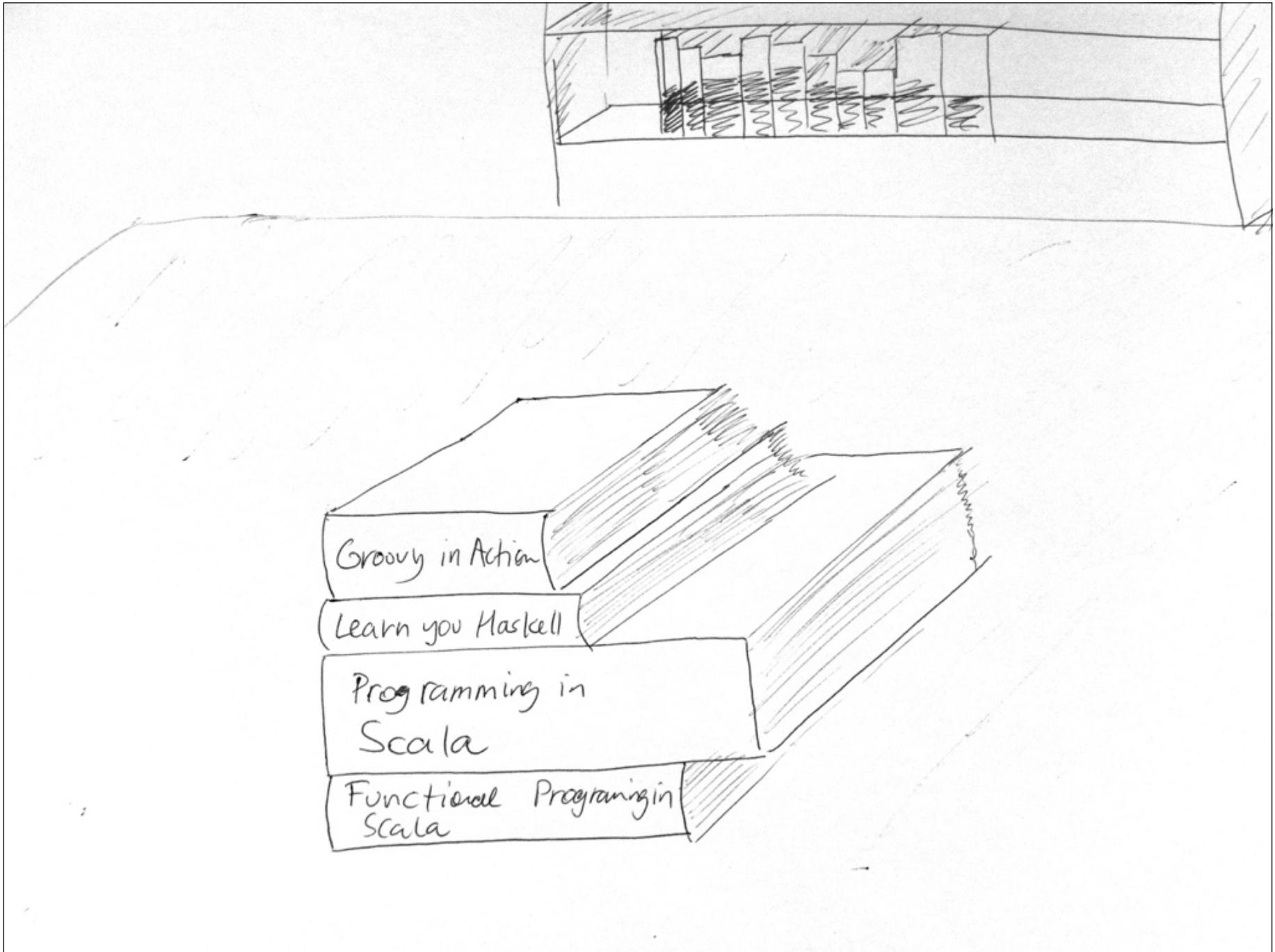
SOMETIMES, WHEN I FIRST WAKE UP, I AM  
CAUGHT IN THE HORRIBLE GRIP OF PERSPECTIVE:

IT MAY BE A JEWEL OF OPEN SOURCE, BUT  
FIREFOX IS JUST A BROWSER. IT SHOWS  
WEBPAGES. WHATTHEHELL IS WRONG WITH US?



FORTUNATELY, THIS SUBSIDES QUICKLY.















Groovy











