



Java oder Scala? Scala und Java!

Erfahrungen aus 2 Jahren Reengineering

Agenda

1. **Java oder Scala? Scala und Java!**
2. "Unsere Scala Geschichte"
3. CLX.Box Reengineering
4. Erfahrungen
5. Fragen/Diskussion

Was ist Scala?

Scala ist eine neue Programmiersprache, welche:

- OO-Konzepte und Funktionale Konzepte vereint
- Statisch typisiert ist und ein mächtiges Typsystem aufweist
- Nach Java Bytecode kompiliert

- **Higher Order Functions**
- Classes, Objects (Singletons), Traits
- Case Classes, Pattern Matching
- Standard Klassen (u.a. Collection Library)
- For Expressions, Monads
- Implicit Conversions
- Actors, Futures, Parallel Collections...

Higher Order Functions

Higher Order Functions Scala:

1. `val photos = List(...)`
2. `val smallPhotos = photos.filter(p => p.sizeKb < 10)`

Lambda Expressions Java 8:

1. `List<Photo> photos = Arrays.asList(...);`
2. `List<Photo> smallPhotos = photos.filter(p -> p.getSizeInKb() < 10);`

Quelle: <http://www.infoq.com/articles/java-8-vs-scala>

Function Chaining

Function Chaining Scala:

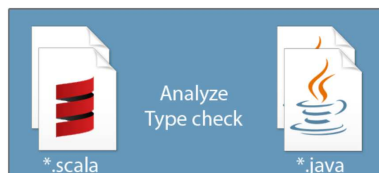
1. `val fatPhotos = photos.filter(p => p.sizeInKb > 10000)`
2. `.map(p => p.name)`

Function Chaining Java 8:

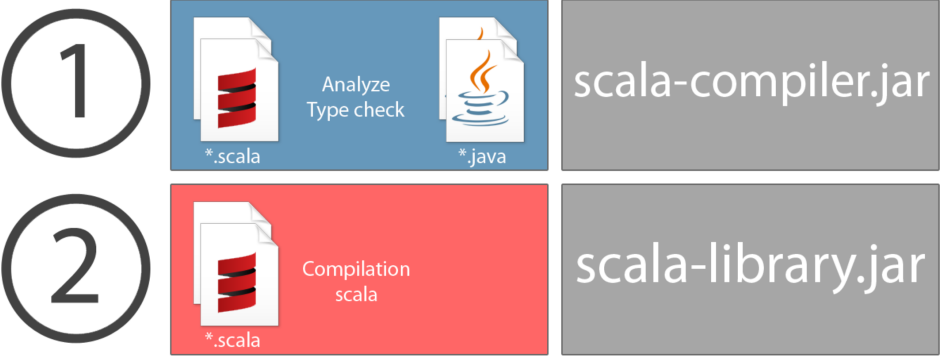
1. `List<Photo> photos = Arrays.asList(...)`
2. `List<String> fatPhotos = photos.filter(p -> p.getSizeInKb() > 10000)`
3. `.map(p -> p.name)`

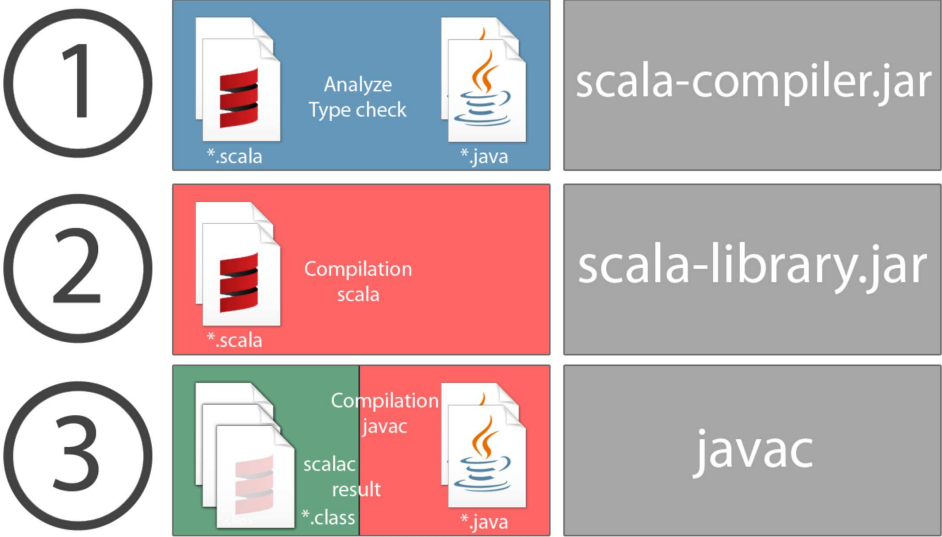
Quelle: <http://www.infoq.com/articles/java-8-vs-scala>

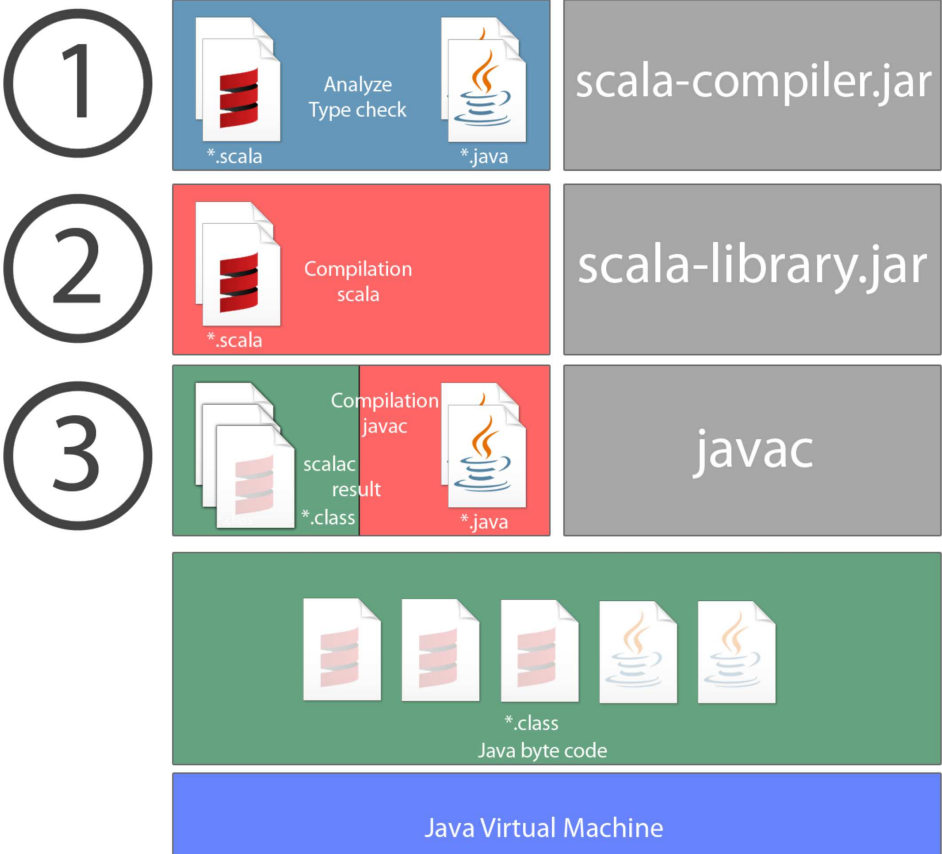
1



scala-compiler.jar



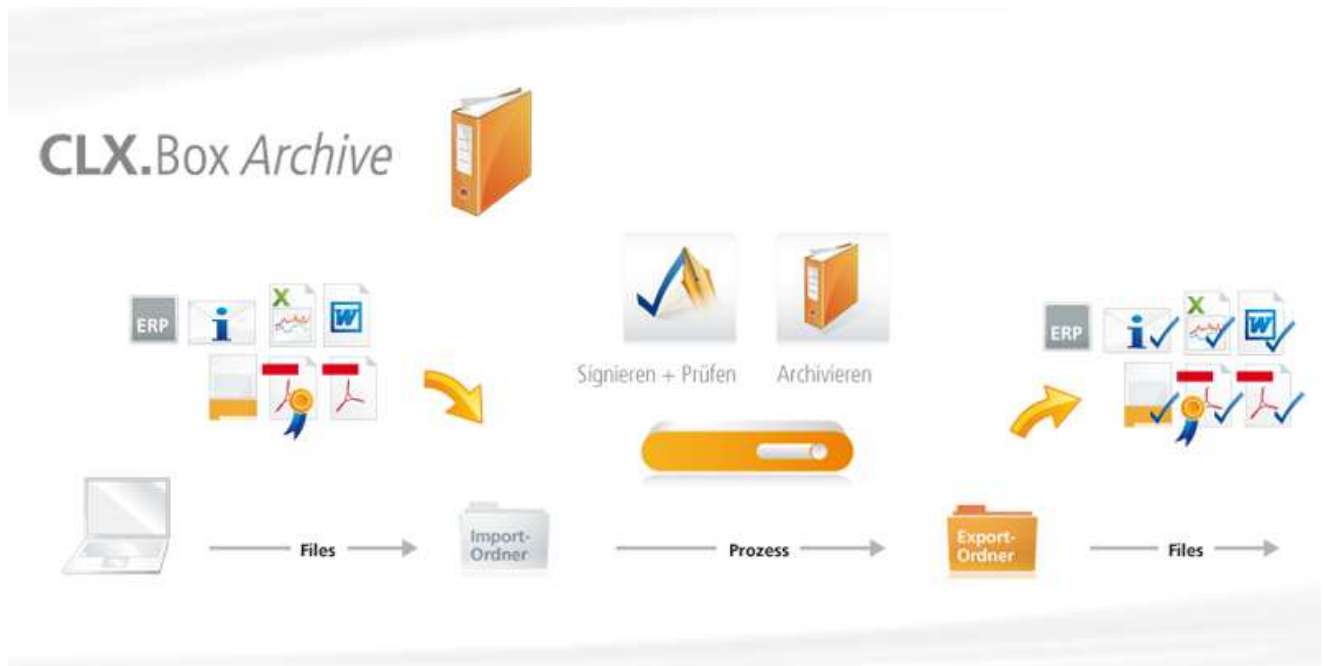
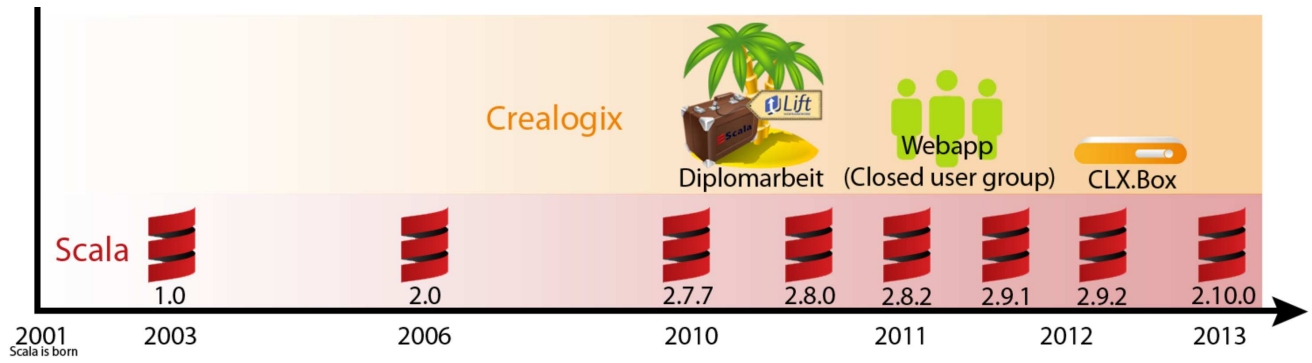




Agenda



- 1. Java oder Scala? Scala und Java!
- 2. "Unsere Scala Geschichte"
- 3. CLX.Box Reengineering
- 4. Erfahrungen
- 5. Fragen/Diskussion



CLX.Box

Hauptmenü

Meldungen

Dashboard

Einstellungen

Archivkonfiguration

Archiv Suche

Status am 28.02.2013 15:48:55: Dokumentverarbeitung

Willkommen, admin | Logout

Dashboard

Letzter Login: 28.02.2013 12:55

Archivbereiche

Archivbereich	Anzahl Dokumente	Weitergeleitete Dokumente	Empfangene Dokumente	Neue Dokumente *	Letzte Verarbeitung
CLX_Systest_10	130622	0	0	0	28.02.2013 13:33
CLX_Systest_4	402589	0	0	2743	28.02.2013 12:44
CLX_Systest_6	35000	0	0	0	28.02.2013 12:44
CLX_Systest_7	35000	0	0	0	28.02.2013 12:44
CLX_Systest_8	35000	0	0	0	28.02.2013 12:44
CLX_Systest_9	35000	0	0	0	28.02.2013 12:44

Archiv Suche

Dokumentverarbeitung Status

Letzte Verarbeitung

28.02.2013 12:44

Nächste Verarbeitung

01.03.2013 03:01

Dokumentverarbeitung

Backup Status

Letztes vollständiges Backup

23.02.2013 07:25

Nächstes vollständiges Backup

02.03.2013 00:00

Letztes differenzielles Backup

Bisher wurde kein differenzielles Backups durchgeführt.

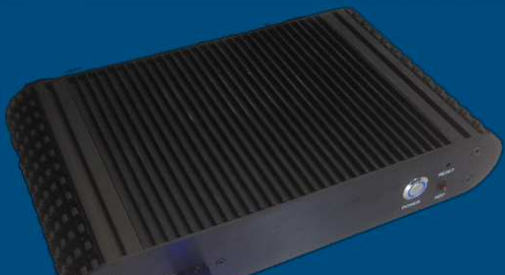
Nächstes differenzielles Backup

Es sind keine differenzielle Backups geplant.

Backup

Supportunterstützung

Exportieren Sie die Systemlogs oder senden Sie diese an den Crealogix-Support.



Agenda

1. Java oder Scala? Scala und Java!
2. "Unsere Scala Geschichte"
3. **CLX.Box Reeengineering**
4. Erfahrungen
5. Fragen/Diskussion

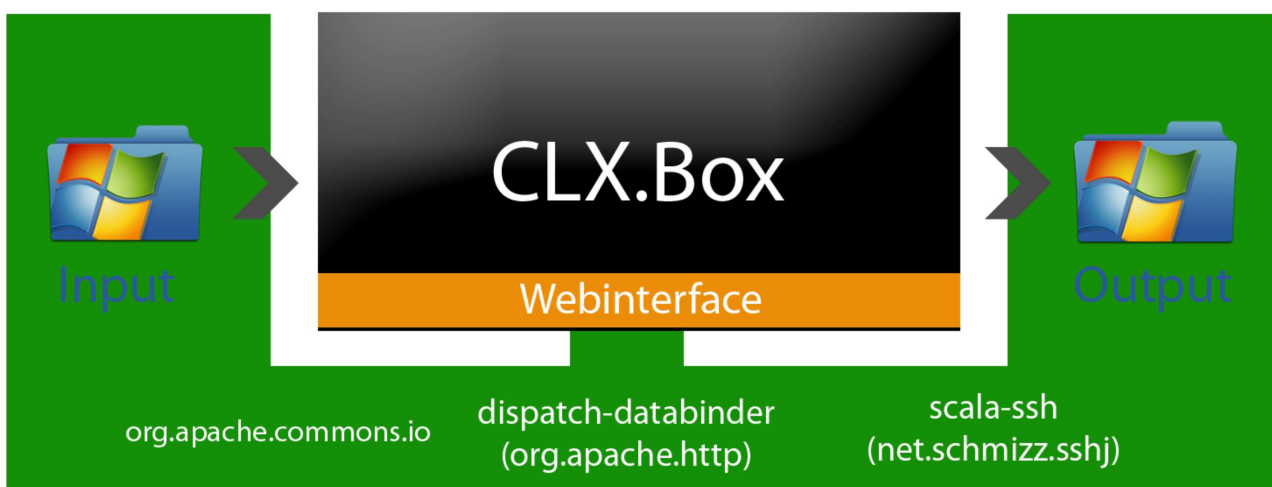


System-Testsuite (specs2)



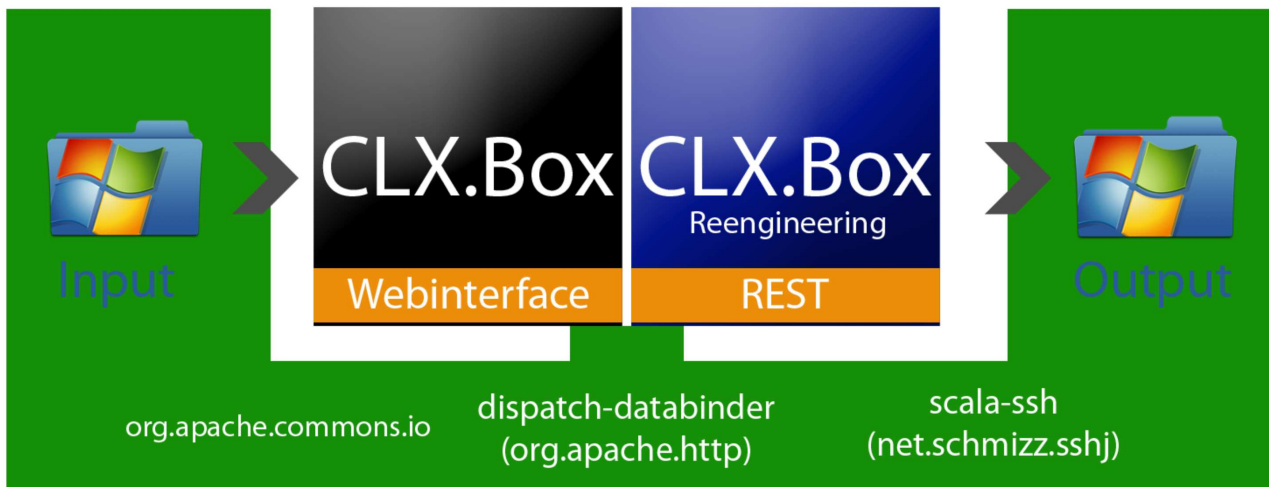
23 von 61

System-Testsuite (specs2)



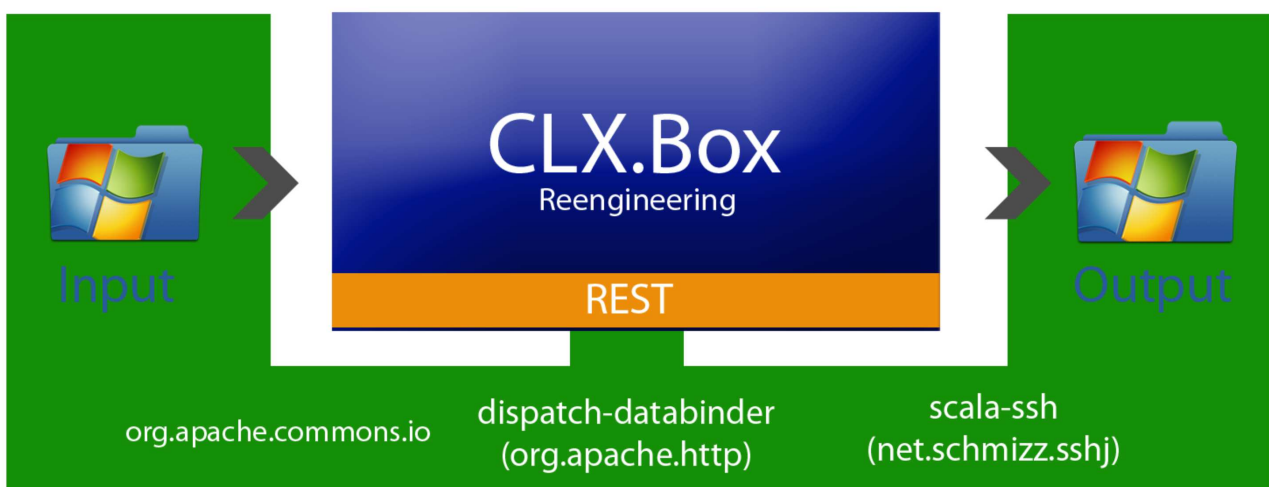
24 von 61

System-Testsuite (specs2)



25 von 61

System-Testsuite (specs2)



26 von 61

Gutes Beispiel?

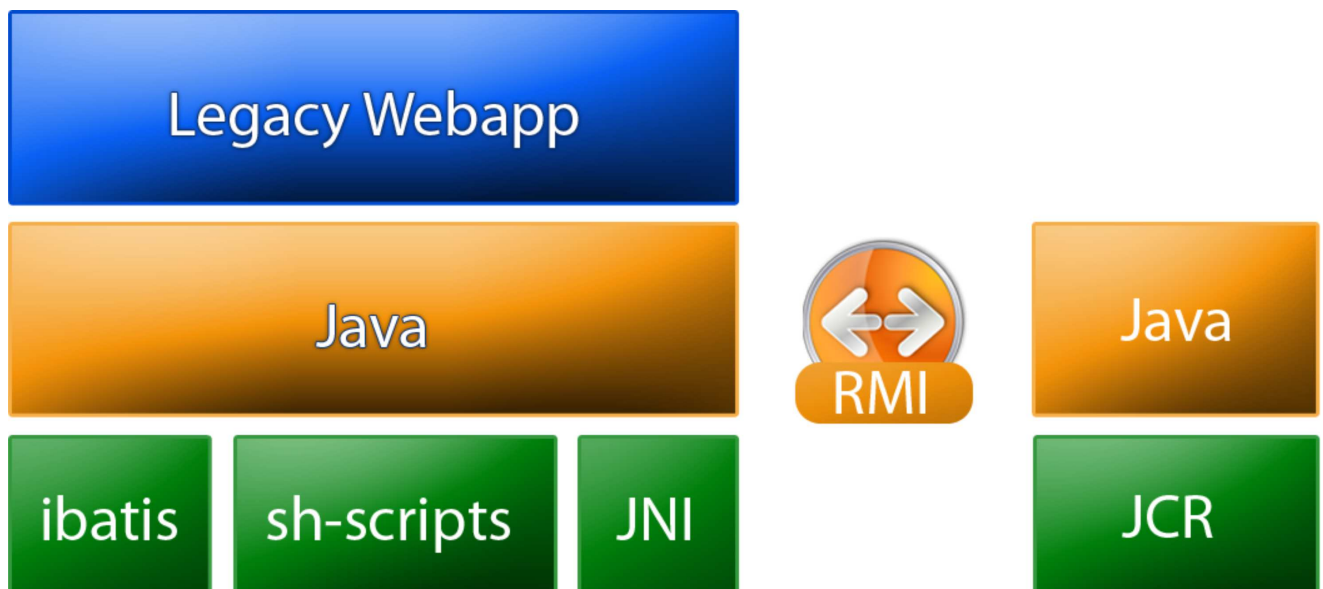
```
1. "CT1001" can {
2.   "copy testFolder to " + inShare in {
3.     Utils.copyTestFolder(directory).to(testingShare) must succeed
4.   }
5.
6.   testInputFile + " exists in " + inShare in {
7.     testInputFile must beAvailableIn(inShare)
8.   }
9.
10.  "starting job" in {
11.    Utils.startWork must haveResponseCode("302")
12.  }
13. ...
14. }
```

Schlechtes Beispiel?

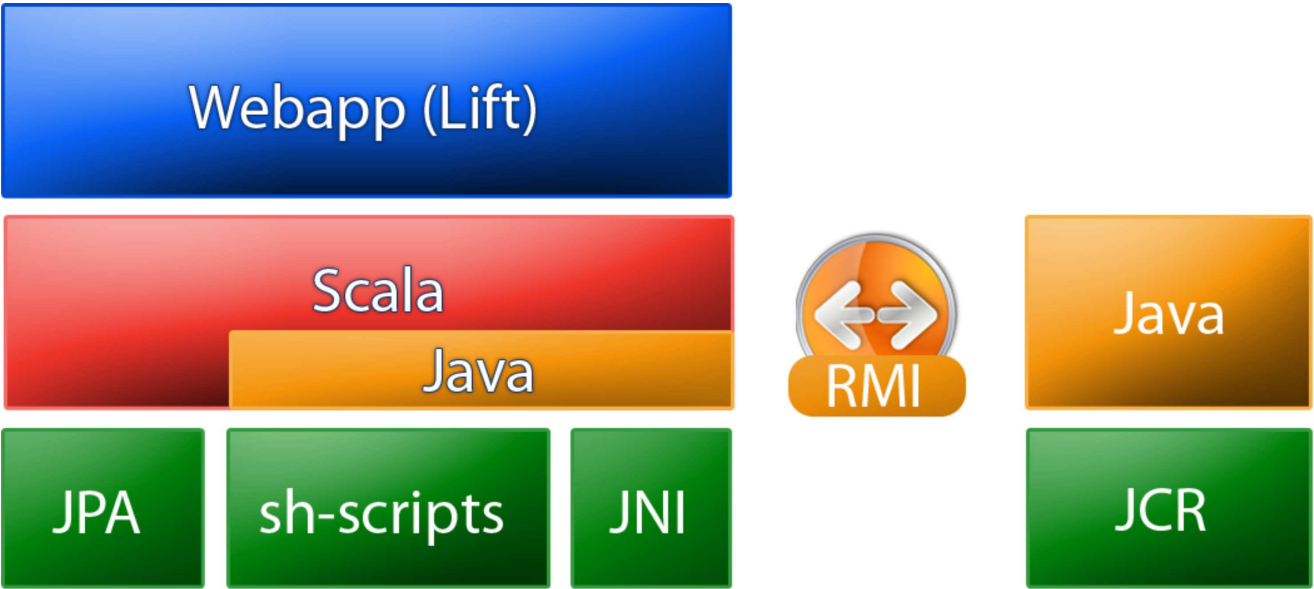
```
1. def downloadArchiveFile[T](cookie: Map[String, String],
2.   downloadUrl: String,
3.   is: InputStream => T): T = {
4.   http(req.as("admin", "passwort") / downloadUrl <:< cookie >+ {
5.     handlers => (handlers >> is, handlers >:> {
6.       map => logger.info("Download map: " + getDownloadName(map))
7.     })
8.   })._1
9. }
```

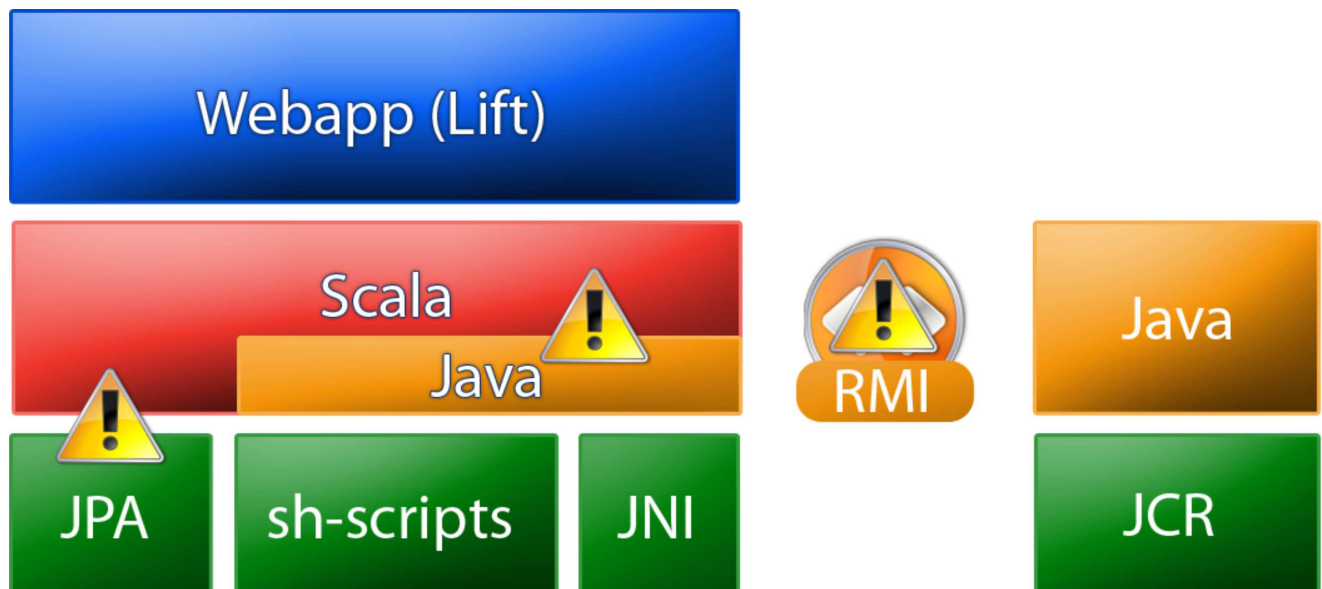


29 von 61



30 von 61





33 von 61

JPA / SPA



1. Kein Scala Enum Support
2. Kein Scala Collection Support
3. Felder können "null" sein

34 von 61

1. Kein Scala Enum Support
2. Kein Scala Collection Support
3. Felder können "null" sein

-
1. Verwendung von Java Enums
 2. **Konversion von Scala zu Java Listen**
 3. Verwendung von Option (Some/None) auf Zwischenschicht

```
1. //Scala class
2. @Entity
3. class ArchiveSpace {
4. ...
5. @OneToMany(cascade = Array(CascadeType.ALL), ...)
6. var processingRules: java.util.List[ProcessingRule] =
7.     new ArrayList[ProcessingRule]()
8. ...
9. }
10.
```

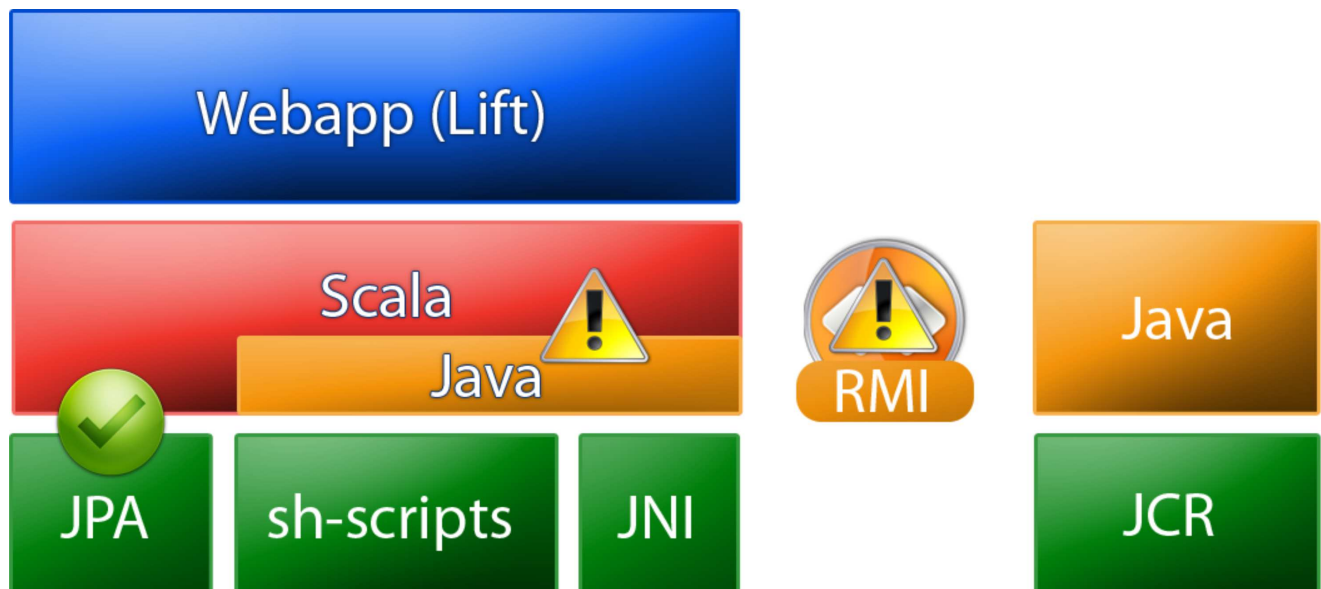
- Scala Collections sind immutable (default)
- JPA erwartet jedoch eine mutable Collection (clone/add/remove)

```
1. import scala.collection.JavaConversions._
2.
3. val as = new ArchiveSpace
4. val scalaList = List(new ProcessingRule)
5. as.processingRules = scalaList //will compile but...
6. Model.merge(as) ///..could throw an exception here at runtime
7.
```

```
1.
2. def convertScalaToJava(scalaList: List[ProcessingRule]) = {
3.   val javaList = new java.util.ArrayList[ProcessingRule]()
4.   scalaList.foreach(elem => javaList.add(elem))
5.   javaList
6. }
7.
8. def convertScalaToJava(scalaList: List[ProcessingRule]) = {
9.   import scala.collection.JavaConversions._
10.  bufferAsJavaList(ListBuffer(scalaList:_*))
11. }
12.
```

```
1. import scala.collection.JavaConversions._
2.
3. val as = new ArchiveSpace
4. val scalaList = List(new ProcessingRule)
5. as.processingRules = scalaList //will compile but...
6. Model.merge(as) //..could throw an exception here at runtime
7.
```

```
1. import scala.collection.JavaConversions._
2.
3. val as = new ArchiveSpace
4. val scalaList = List(new ProcessingRule)
5. as.processingRules = convertScalaToJava(scalaList) //will compile and...
6. Model.merge(as) //..work as expected
7.
```



41 von 61

RMI

CREALOGiX 

- Jede VM hat ihren eigenen Classloader
- Die transferierten Typen müssen beiden VM's bekannt sein.



42 von 61

- Unproblematisch für Standardtypen (String, Long, Integer, ...)

```
1. //Java method on java remote interface  
2. public String addArchiveSpaceToTenant(long tenantId,  
3.     AbstractArchiveSpace archiveSpace)  
4.     throws RemoteException;
```

```
1. //Scala code  
2. archive.addArchiveSpaceToTenant(1L, new InvoiceArchiveSpace)
```

- Problematisch bei Collections (z.B. List)

```
1. //Java method on java remote interface  
2. public AbstractVirtualFile getFile(List<AbstractArchiveSpace> aS)  
3.     throws RemoteException;
```

```
1. //Scala code  
2. import scala.collection.JavaConversions._  
3.  
4. val scalaList = List(new InvoiceArchiveSpace)  
5. archive.getFile(scalaList) //exception
```

Implementationstyp muss auf Seite Java bekannt sein:

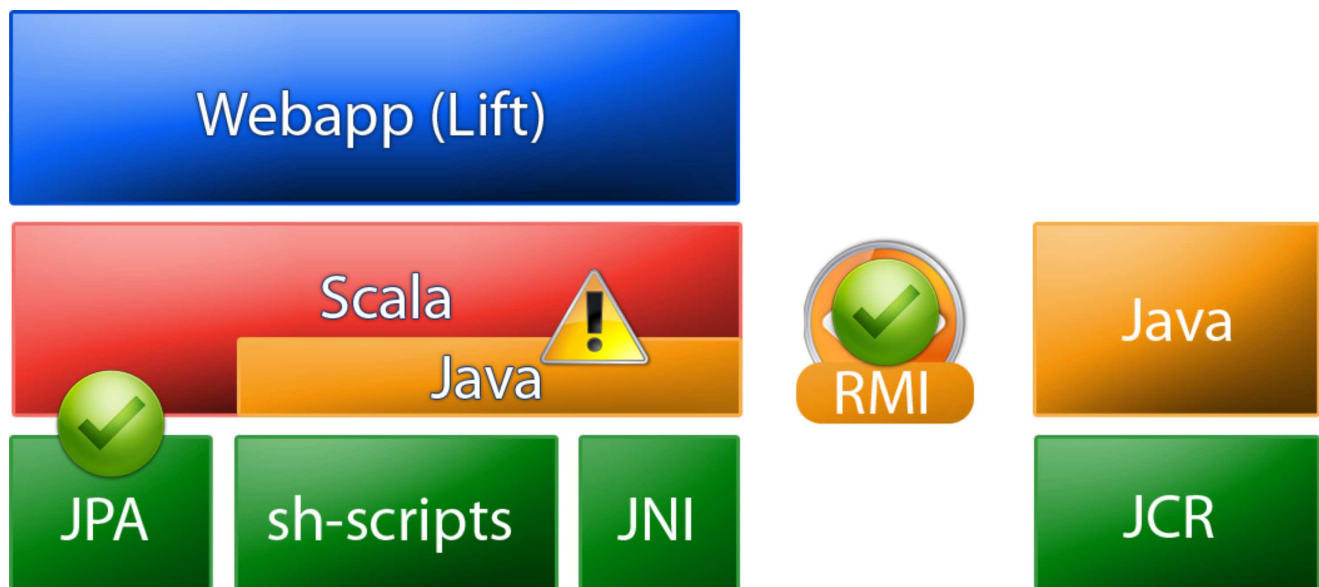
```
1.  
2. def convertScalaToJava(scalaList: List[AbstractArchiveSpace]) = {  
3.   val javaList = new java.util.ArrayList[AbstractArchiveSpace]()  
4.   scalaList.foreach(elem => javaList.add(elem))  
5.   javaList  
6. }  
7.
```

```
1. //Java method on java remote interface  
2. public AbstractVirtualFile getFile(List<AbstractArchiveSpace> aS)  
3. throws RemoteException;
```

```
1. //Scala code  
2. import scala.collection.JavaConversions._  
3.  
4. val scalaList = List(new InvoiceArchiveSpace)  
5. archive.getFile(scalaList) //exception
```

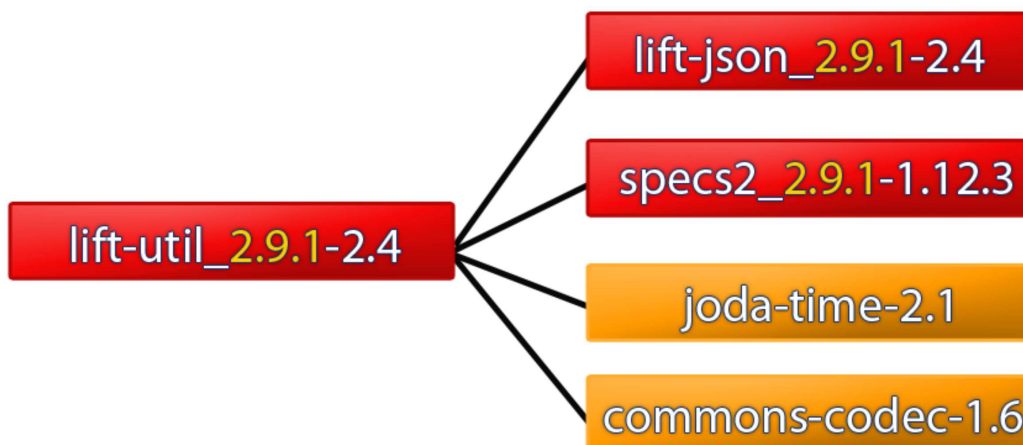
```
1. //Java method on java remote interface  
2. public AbstractVirtualFile getFile(List<AbstractArchiveSpace> aS)  
3. throws RemoteException;
```

```
1. //Scala code  
2. //import not needed anymore  
3.  
4. val scalaList = convertScalaToJava(List(new InvoiceArchiveSpace))  
5. archive.getFile(scalaList) //works
```



- Aufruf Scala zu Java (n-Mal)
- Aufruf Java zu Scala (2 Mal)

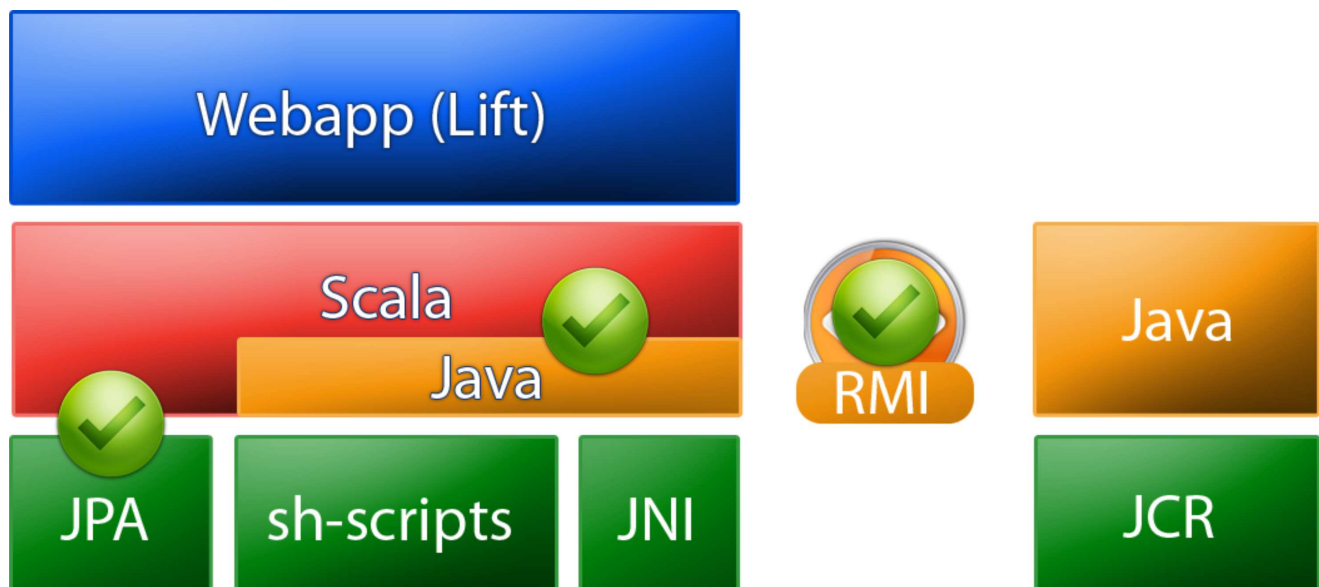
```
205 |      * @param searchRequest the request
    |      net.liftweb.common
    |      sealed abstract class Box[+A]() extends Object
    |      with Product with ScalaObject with ScalaObject
209 |      */
210 |      public Box<Boolean> exportArchiveEntries
211 |
```



Zusammenspiel Scala / Java - Version fragility **CREALOGiX**

- Maven mit Variablen
- sbt mit "cross build" - Funktion
- Binärkompatibilität bei Minor Releases von Scala
- In den meisten Fällen sind die Bibliotheken "source compatible"

51 von 61



52 von 61



Agenda



1. Java oder Scala? Scala und Java!
2. "Unsere Scala Geschichte"
3. CLX.Box Reengineering
4. **Erfahrungen**
5. Fragen/Diskussion

Erfahrungen - Komplexität von Scala meistern **CREALOGiX**

- Verzicht auf komplexe Konzepte (z.B. eigene Implicit Conversions)
- Verzicht auf exotische Scala Libraries
- Reengineering Java nach Scala: Von oben nach unten

Erfahrungen - Scala in der Firma **CREALOGiX**

- Management Support
- Infrastruktur: Schnelle PCs, IntelliJ (Eclipse), git, Maven/sbt
- Lernkurve berücksichtigen: Vom kleinen zum grossen Projekt

- Offizielle Style Guides, Code Conventions/Corrections in IDE
- Code Walkthroughs: Zusammen Scala Code lesen
- Pair Programming, Refactoring

>> Collective Code Ownership <<



- Schnellerer Compiler, bessere Doc
- (Noch) besserer Support für Multicore Programmierung: Actors, Futures/Promises, Parallel Collections
- Konsolidierung der neuen 2.10 Features (u.a. String Interpolation, Value Classes, Language Imports, Implicit Classes, Macros)

- Die Kombination Scala / Java funktioniert für das Produkt CLX.Box
- Scala kann gut für Teilaspekte eingesetzt werden (z.B. für Testing)
- Scala 2.10 verspricht bessere Stabilität bez. dem Hauptschwachpunkt "Version Fragility"

- Was sind die Wow-Erlebnisse und die Fallstricke beim Einsatz von Scala?
- Wie kann man die zusätzliche Komplexität von Scala schrittweise meistern und gewinnbringend einsetzen?
- Wie kann in einem Java-Projekt Scala für Teilaspekte eingesetzt werden?
- Welche Versprechen macht Scala für die Zukunft?

<http://www.scala-lang.org>
<http://docs.scala-lang.org>
<http://typesafe.com>

Programming in Scala - Martin Odersky et al.
http://booksites.artima.com/programming_in_scala

Programming Scala - Dean Wampler
<http://ofps.oreilly.com/titles/9780596155957/index.html>

Kurs Funktionale Programmierung - Martin Odersky
<https://www.coursera.org/course/progfun>