



# Liftweb in the Real-World

## A hyper-polyglot's Tale

Franz Bettag <franz@bett.ag>

<http://uberblo.gs>

<http://www.bett.ag>

<https://twitter.com/fbettag>

# I am..

- Franz Bettag (franz@bett.ag)
- from Nuremberg, Germany
- 26 years old (it's true!)
- proud owner of a small company ([www.bett.ag](http://www.bett.ag))
- \*BSD Admin, code fetishist and polyglot



# I am NOT..

- a blogger (but i do blog, [www.uberblo.gs](http://www.uberblo.gs))
- an expert in all the languages i code/speak
- saying you must believe me, but you should

# Languages i've coded

- Pascal
- JavaScript/HTML, LotusScript/Notes DBs
- C, C++, VC++
- Perl, PHP, Python
- Ruby, Java (later JRuby)
- Objective-C, Erlang
- Scala!



# Scala

- Typesafe! Typesafe! Typesafe!
- Write less tests
- Write more meaningful code
- Less Boilerplate code
- High concurrency
- Massive amounts of libs (Java compat)



# Why Type-safety helps

```
def billing(totalprice, myTax = 19.0)
  myTax.to_f! if myTax.class != Float
  totalprice.to_f! if totalprice.class != Float
  (myTax / 100 + 1) * totalprice
end
```

# Success!

>> billing(100.00)

=> 119.00

^ works!



# Invoicing jacked!



```
>> billing("100.00")
```

```
TypeError: String can't be coerced into Float
```



# Catching the Culprit

```
street_num = item[24].match(/d+\s*w?(?:\s*[-|\/]\s*d+\s*w?)?$/)[0].strip rescue ''
delivery_address = {
  :third_party_customer_id => tpc.id,
  :street => item[24].gsub(/(S|s)tr\./, '\1trasse').
    gsub(/d+\s*w?(?:\s*(?:[-|\/])\s*d+\s*w?)?$/, '').to_s.upwords,
  :street_add => item[26].to_s.upwords,
  :street_num => street_num,
  :zip => item[29].gsub(/^\s+/, '').strip,
  :city => item[30].to_s.upwords,
  :city_add => item[32] ? item[32].to_s.upwords : nil,
  :country_id => Country.find_by_short(item[34].downcase).id,
  :created_by => 2,
  :updated_by => 2,
}
```

# Expect the unexpected

- Wrong invocation?
- Opened a Box or got Some(None)?
- Null Pointer Exception?

# Nipped in the butt

- String \* Float
- \$wait=0; .... code .... code .... sleep(\$wait);
- include(\$\_POST['iamsostupid'])
- `cat {\$\_GET['iamsogifted']} | wc -l`
- SQL INSERT \$browser\_string \*poof\*

# Liftweb

- Typesafe! Typesafe! Typesafe!
- Highly scalable
- Write even more meaningful code
- Write less JavaScript Wiring
- View first
- CSS Selectors



# Ajax Textfield

```
val myF = ""
```

```
def setF(a: String) = {myF = a; Noop}
```

```
def render =
```

```
  “.foo” #> SHtml.ajaxText(“myfield”, setF _)
```

# With fancyness

```
def setF(a: String): JsCmd = {  
  myF = a  
  doSomethingWith(a)  
  JsRaw(“$(‘#bar’).effect(..)”).cmd  
}  
  
def render =  
“#bar” #> SHtml.ajaxText(myF, setF _, “id” -> “bar”)
```

# Playing with Callbacks

```
(SHtml.ajaxText("foo", myFunc _) \\ "@onblur")  
    .toString.replaceAll(  
        "this.value",  
        "\\$( '#bar' ).val()"  
    )
```

# The Akka Lift Example

- CometActor
- Local LiftActor
- Local Akka Actor
- Remote Akka Bridge
- Remote Akka Actor
- jQuery-UI ProgressBar





# Comet Actor

- `val targetActor =`  
`LADemoLiftActor`
- `val targetActor =`  
`LADemoAkkaActor.actor`
- `val targetActor = Actors.actorOf(`  
`classOf[LADemoAkkaRemoteProxyService]`  
`).start()`

# How it's done

```
var copyQueue: Map[CometActor, LADemoFileCopyInfo] = Map()
```

```
sealed trait LADemoFileStat
```

```
sealed trait LADemoFileCopyInternal extends LADemoFileStat
```

```
case class LADemoFileCopyInternalWait extends LADemoFileCopyInternal
```

```
case class LADemoFileCopyInternalStart(source: Path, target: Path, actor: CometActor) extends LADemoFileCopyInternal
```

```
sealed trait LADemoFileCopyInfo extends LADemoFileStat
```

```
case class LADemoFileCopyRequest(actor: CometActor, file: String) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyQueue(waiting: Int) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyStatus(file: String, percent: Int, speed: String, remaining: String) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyDone(success: Boolean, when: DateTime) extends LADemoFileCopyInfo
```

# Lift Actor

```
object LADemoLiftActor extends LiftActor

  with LADemoStatMethods with LADemoFileCopyMethods {

    protected def messageHandler = {

      case a: LADemoStatGather =>          a.actor ! sysStatInfo

      case a: LADemoFileCopyRequestList =>  a.actor ! copyFileList

      case a: LADemoFileCopyAbortRequest => copyDequeue(a.actor)

      case a: LADemoFileCopyRequest =>

        copyQueueWithInfo(a) match {

          case stat: LADemoFileCopyInternalStart =>

            LAScheduler.execute(() => copyFileStart(stat))

          case _ =>

            } } }

  }
```

# Akka Actor

```
object LADemoAkkaActorService extends Actor

  with LADemoStatMethods with LADemoFileCopyMethods {

  override def receive = {

    case a: LADemoStatGather =>          a.actor ! sysStatInfo

    case a: LADemoFileCopyRequestList =>  a.actor ! copyFileList

    case a: LADemoFileCopyAbortRequest => copyDequeue(a.actor)

    case a: LADemoFileCopyRequest =>

      copyQueueWithInfo(a) match {

        case stat: LADemoFileCopyInternalStart =>

          LAScheduler.execute(() => copyFileStart(stat))

        case _ =>

      } } }
```

# Akka Remote Actor

- Problem 1: Serializing a CometActor?
  - Problem 2: Bridging Actor w/ Mapping?
  - Solution: per CometActor Bridge
- 
- Problem 3: Can't do self.channel ! SomeReply
  - Solution: per CometActor Bridge is blocking (reply)

# Akka Remote Actor

```
class LADemoAkkaRemoteActorService extends Actor
  with LADemoStatMethods with LADemoFileCopyRemoteMethods {

  override def receive = {

    case LADemoStatGatherRemote =>                self.reply(sysStatInfo)

    case LADemoFileCopyRequestListRemote =>        self.reply(copyFileList)

    case a: LADemoFileCopyAbortRequestRemote =>    copyDequeue(a.actorRef)

    case a: LADemoFileCopyRequestRemote =>

      var result:Any = copyQueueWithInfo(a) match {

        case stat: LADemoFileCopyInternalStartRemote =>

          LAScheduler.execute(() => copyFileStart(stat)); "starting ;)"

        case any:Any => any

      }

      self.reply(result)

  } }

}
```

# Akka Remote Bridge

```
class LADemoAkkaRemoteBridgeService extends Actor {  
  val remoteHost = Props.get("akka.remote.host") openOr("127.0.0.1")  
  val remotePort = Props.get("akka.remote.port").openOr("2552").toInt  
  val remoteActor = remote.actorFor("lift-akka-example-service", remoteHost, remotePort)  
  val actorRef = scala.util.Random.nextLong  
  
  override def receive = {  
    case a: LADemoStatGather =>  
      sendToComet(a.actor, replyFromRemote(LADemoStatGatherRemote))  
  
    case a: LADemoFileCopyRequestList =>  
      sendToComet(a.actor, replyFromRemote(LADemoFileCopyRequestListRemote))  
  
    case a: LADemoFileCopyAbortRequest =>  
      remoteActor ! LADemoFileCopyAbortRequestRemote(actorRef); self.stop()  
  }  
}
```

# Akka Remote Bridge

- ```
protected def sendToComet(cometActor: CometActor, x: Option[Any]) {  
  x match {  
    case Some(a:Any) => cometActor ! a  
    case _ =>  
  }  
}
```
- ```
protected def sendToComet(cometActor: CometActor, x:Any) {  
  sendToComet(cometActor, Some(x))  
}
```



# Akka Remote Bridge

- ```
protected def replyToComet(cometActor: CometActor, x: Option[Any]) {  
  x match {  
    case Some(a:Any) => cometActor ! a  
    case _ =>  
  } }  
  
protected def replyFromRemote(x:Any): Option[Any] = remoteActor !! x
```

# How Remote is done

```
var copyQueue: Map[Long, LADemoFileStat] = Map()
```

```
sealed trait LADemoFileStat
```

```
sealed trait LADemoFileCopyInternal extends LADemoFileStat
```

```
case class LADemoFileCopyInternalWait extends LADemoFileCopyInternal
```

```
case class LADemoFileCopyInternalStart(source: Path, target: Path, actor: CometActor) extends LADemoFileCopyInternal
```

```
case class LADemoFileCopyInternalStartRemote(source: Path, target: Path, actorRef: Long) extends LADemoFileCopyInternal
```

```
sealed trait LADemoFileCopyInfo extends LADemoFileStat
```

```
case class LADemoFileCopyRequest(actor: CometActor, file: String) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyRequestRemote(actorRef: Long, file: String) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyQueue(waiting: Int) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyStatus(file: String, percent: Int, speed: String, remaining: String) extends LADemoFileCopyInfo
```

```
case class LADemoFileCopyDone(success: Boolean, when: DateTime) extends LADemoFileCopyInfo
```

# The Comet-magic

```
case a: LADemoFileCopyStatus =>
```

```
  cachedStat = Full(a)
```

```
  partialUpdate(
```

```
    SetHtml("comet_copy_list", Group(
```

```
      <div id="progressbar"></div>
```

```
      <strong>Copy for file '{a.file.name}' @ {a.speed} in progress! {a.remaining} remaining.</strong>
```

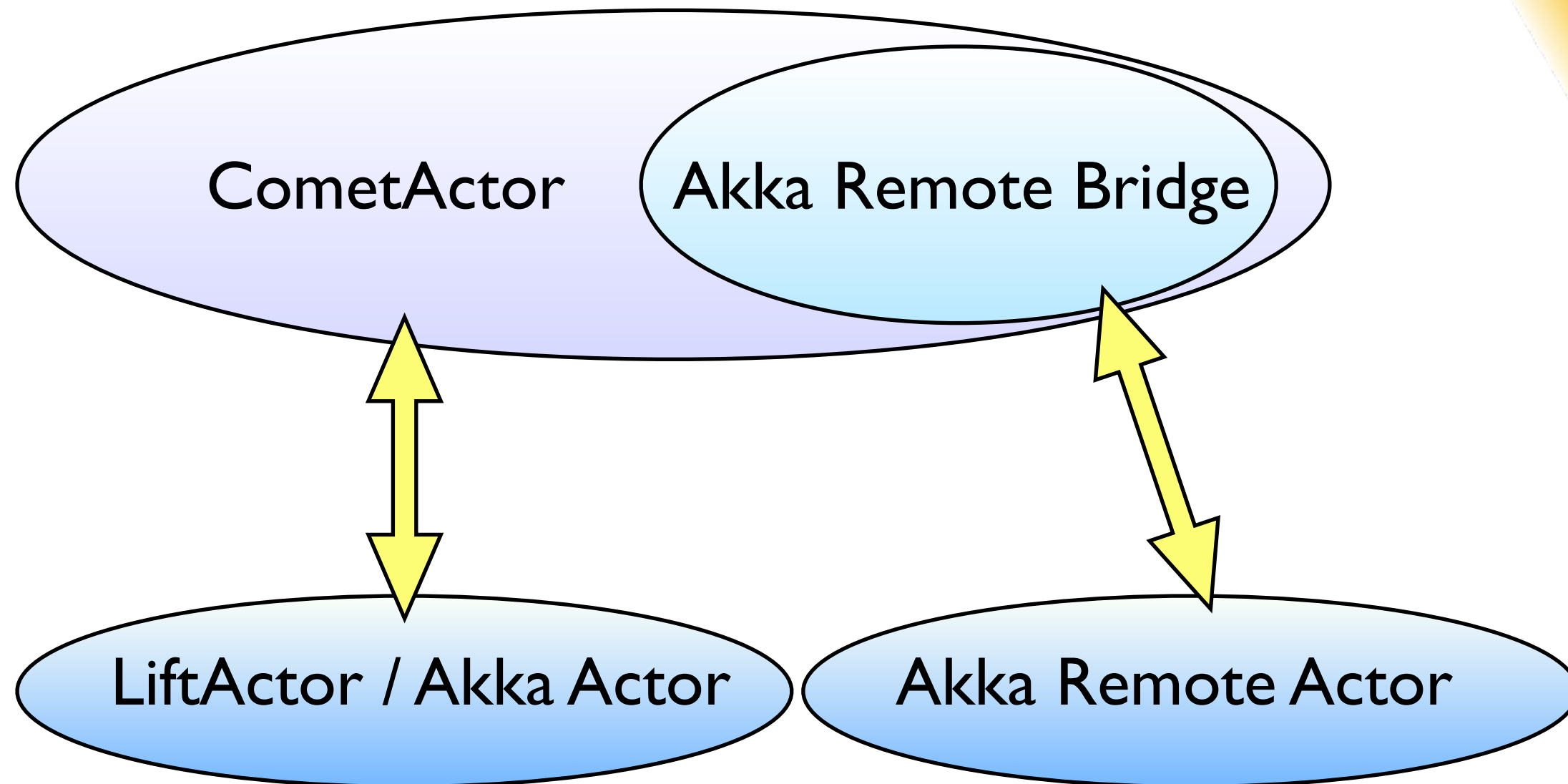
```
    )) &
```

```
    JsRaw("lademo.copyStatus(%s)".format(write(a)))
```

```
  )
```

```
LADemoFileCopyStatus(file: String, percent: Int, speed: String, remaining: String)
```

# Let's throw it together



Enough babbling



It's time for a live-demo. ;)

# Break it yourself!



<http://scalalol2011.bett.ag>

# Dear Santa...

- Arbitrary JavaScript-Callbacks
- Simpler Howtos and Examples
- more of them!





**BETTAG**  
SYSTEMS

Thank you  
for your joining me!





# Consultant for Hire!

Franz Bettag <franz@bett.ag>

<http://uberblo.gs>

<http://www.bett.ag>

<https://twitter.com/fbtag>