

Liftweb

Łukasz Kuczera



Scala

Statically typed with types inference

Object-functional

Compiles to Java bytecode

And CLI .Net



Scala - Basics

```
class Test {  
  // variable definition  
  var n = 5  
  // value definition  
  val i = 10  
  // function definition  
  def printit(x: Any) = println(x)  
  def sum(i: Int, n: Int) = i+n  
}  
object Test extends Test {  
  def main(args: Array[String]): Unit = {  
    printit(sum(i,n))  
  }  
}
```



XML

```
val xml = <a href="http://example.com">
    <customTag b="foo">
        <deepTag b="bar" />
        Custom content
    </customTag>
</a>

println(xml \ "customTag")
println(xml \\ "deepTag")
println(xml \\ "@b")
println(xml \ "customTag" \ "deepTag" \ "@b")
```

Outputs:

```
<customTag b="foo"><deepTag b="bar"></deepTag>Custom content</customTag>
<deepTag b="bar"></deepTag>
foobar
bar
```



XML

```
val xml = <div class="rsswidget">
  <ul>
    <li class="rsswidgettitle"><a href="http://www.quotationspage.com/qotd.html">Quotes of the Day</a></li>
    <li class="rsswidgetitem"><a href="http://www.quotationspage.com/quotes/C._P._Snow">C. P. Snow</a></li>
    <li class="rsswidgetitem"><a href="http://www.quotationspage.com/quotes/unknown">unknown</a></li>
    <li class="rsswidgetitem"><a href="http://www.quotationspage.com/quotes/Frederick">Frederick</a></li>
  </ul> ...
</div>

(xml \\ "@href").slice(1,4).foreach(url =>
  println("URL:"+url)

  val con = new URL(url.toString).openConnection
  val reader = new BufferedReader(new InputStreamReader(con.getInputStream, "ISO-8859-1"))
  var line = "";
  while(line != null) {
    line = reader.readLine
    if(line != null) println(line)
  }
  con.getInputStream.close
)
```



Actors

Easier concurrency

Message passing instead of locks

Asynchronous

Implemented using Java threads

ScalaActors, Akka, Lift - CometActor



Actors

```
class ActorUser {  
    def send(msg: Any) = ScalaActor ! msg  
}  
  
object ScalaActor extends Actor {  
    /* The $actor's mailbox. */  
    // private[actors] val mailbox = new MQueue[Msg]("Reactor")  
    def act() {  
        receive {  
            case "ping!" => println("pinged!");  
            case ("ping!", actor: Actor) => actor ! "pong!"  
            case msg => println("unhandled message type:"+msg)  
        }  
    }  
}
```



Lift

History

Chat Application Using Comet



History

Ruby conf 2006

Problems with ruby runtime

Webframework manifesto

Tim O'Reilly writes post about Lift



Build Tools

SBT

Maven

Gradle

JRebel



IDE

IntelliJ IDEA

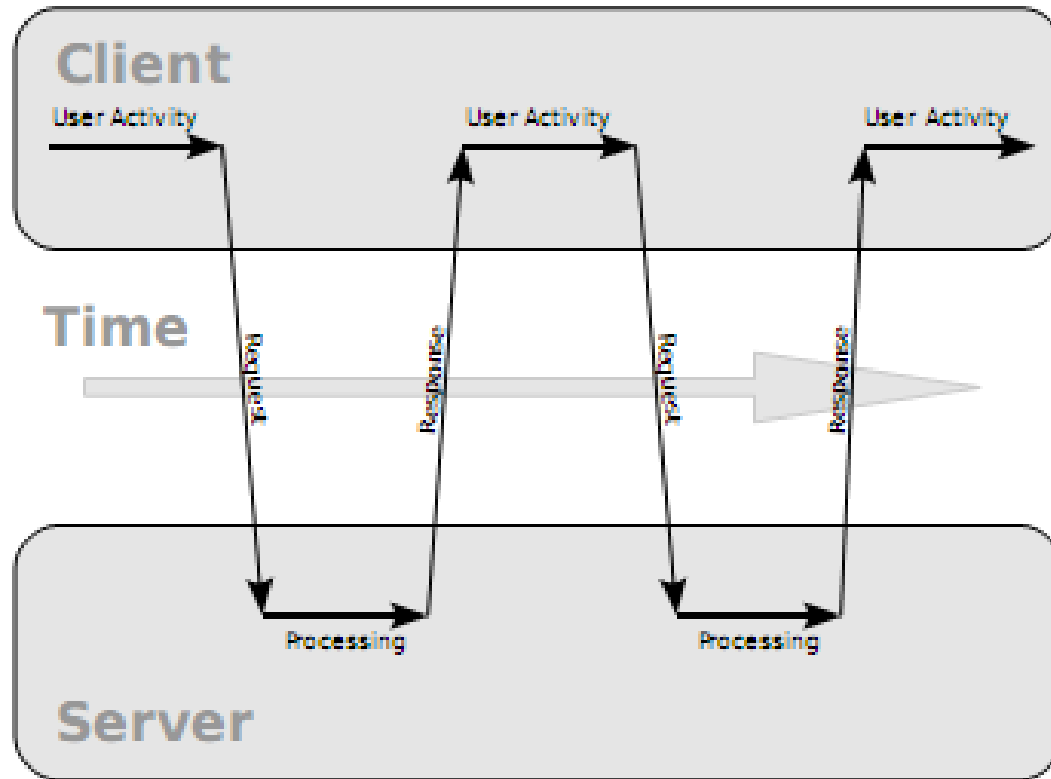
Netbeans

Eclipse

Emacs - Ensimae



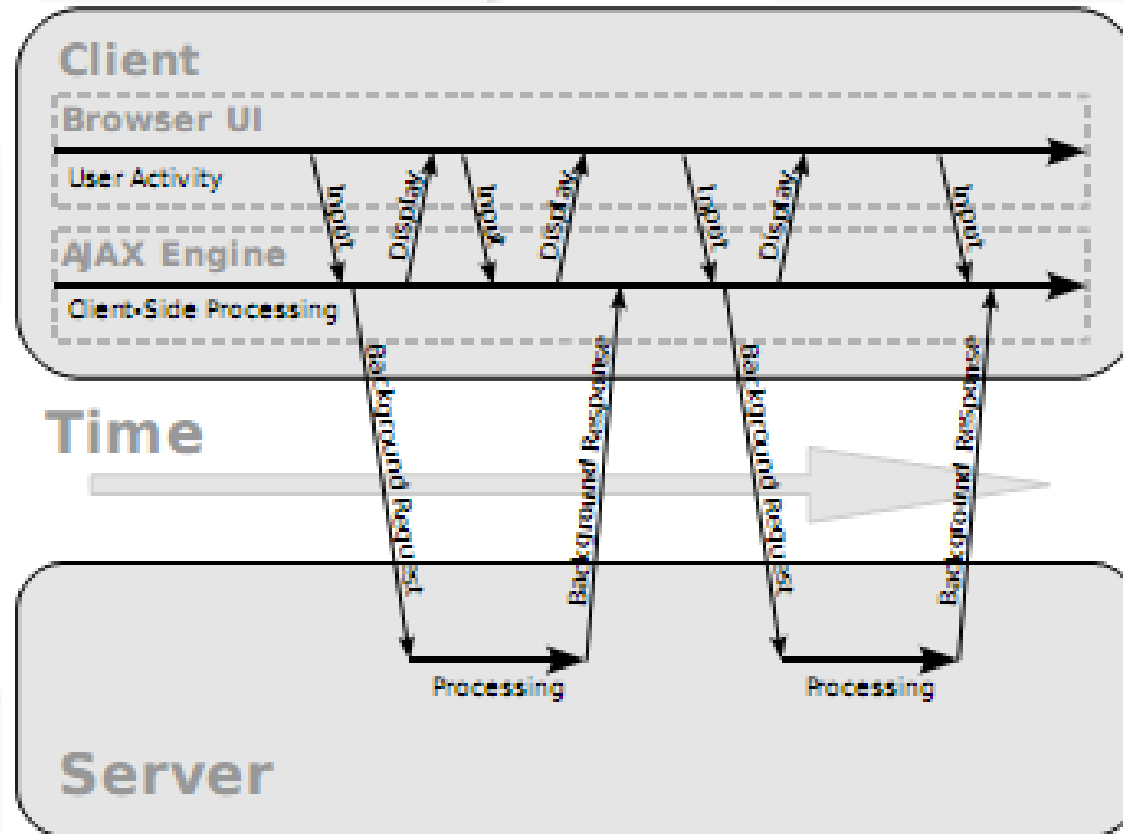
HTTP GET



(a) Traditional Application Model



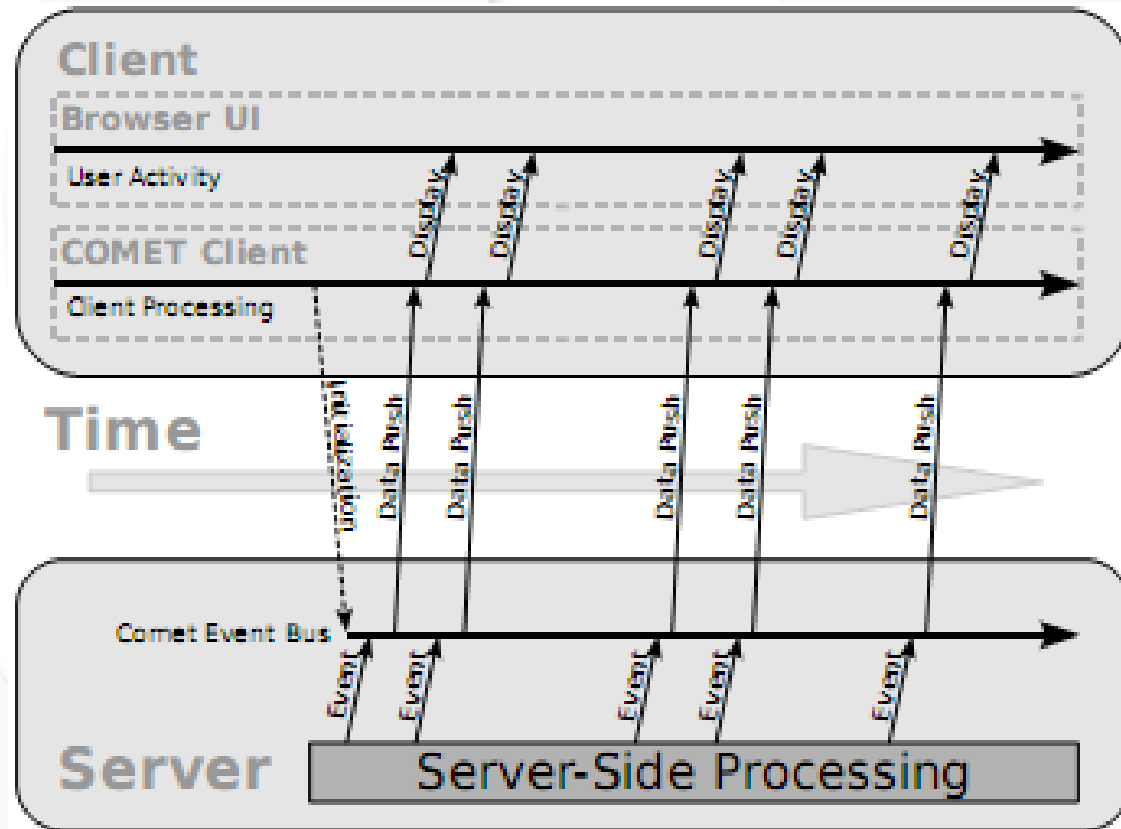
Ajax



(b) AJAX Application Model



Comet



(c) Comet Application Model





041209

Tassopulos

Questions



Contact

email: l.kuczera@jextreme.pl

blog: <http://acidbits.org/blog>

project: <http://github.com/kukems/lift-blog>

