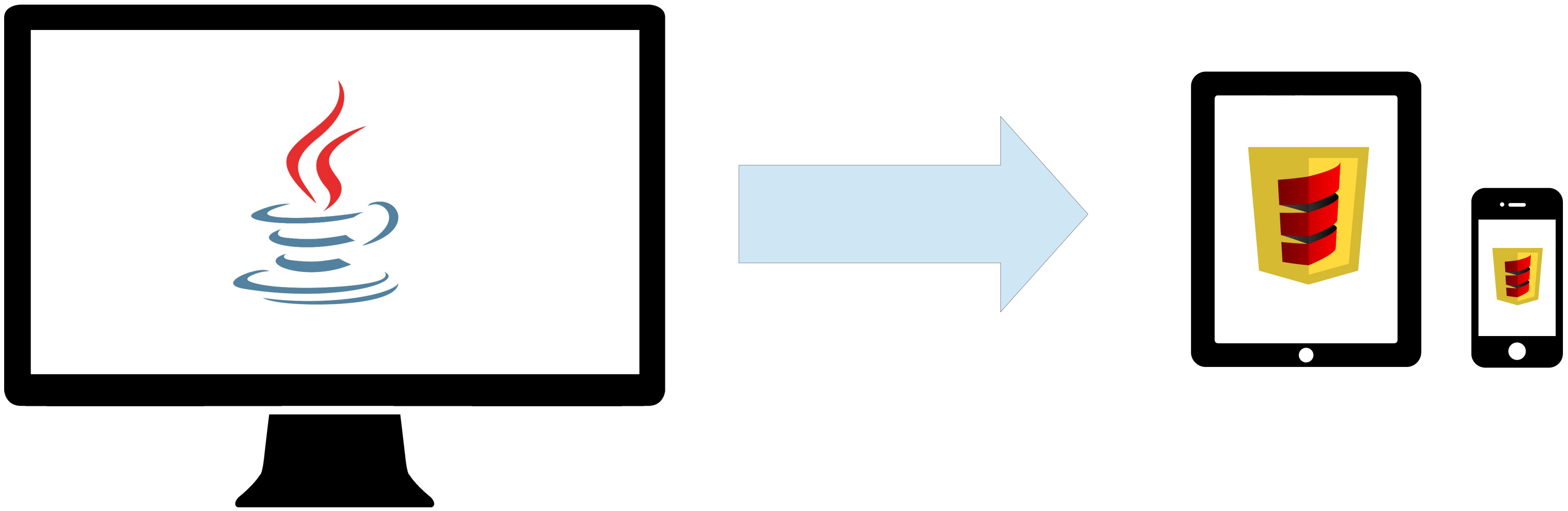
From Swing to Mobile



Technologies

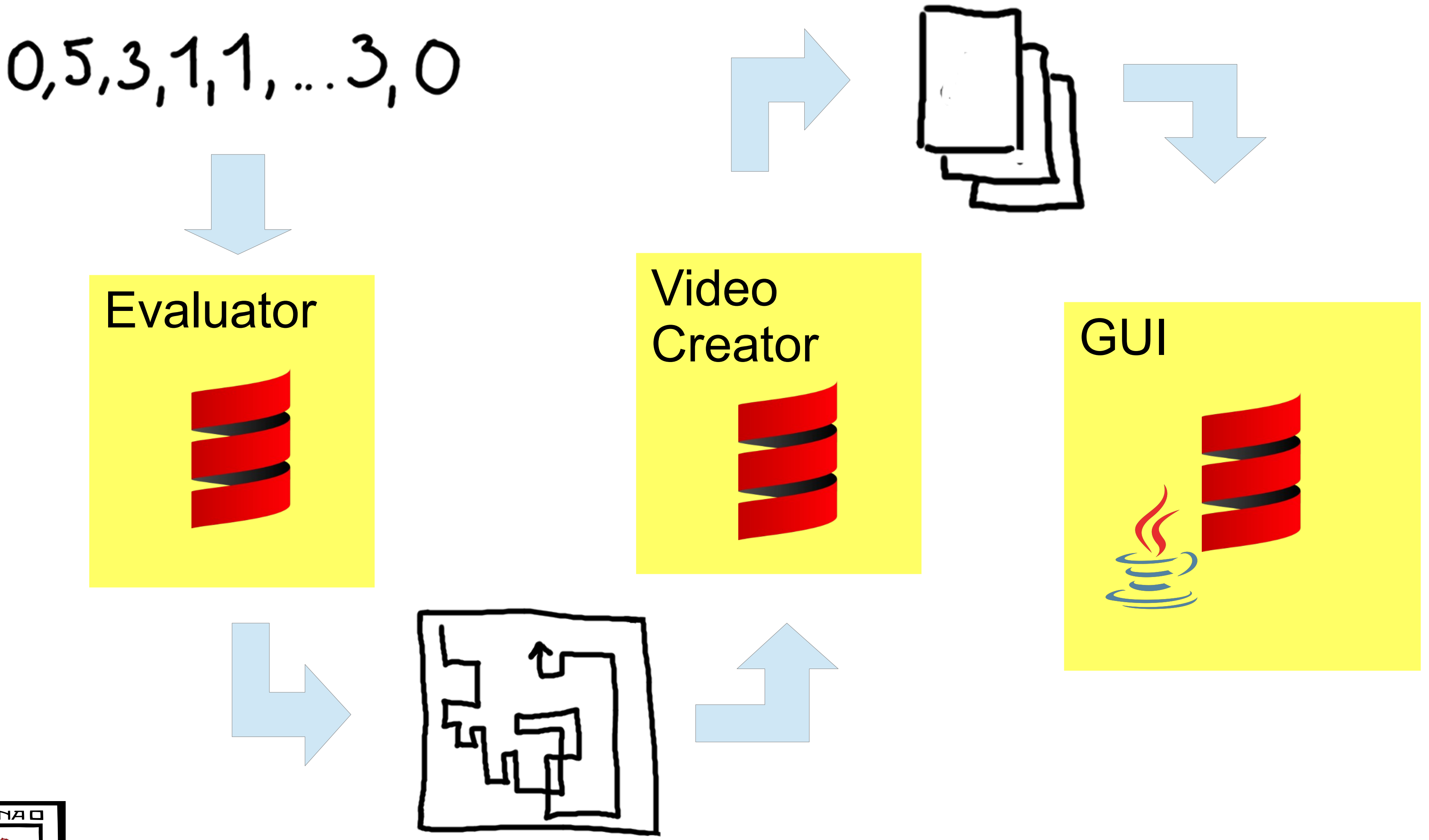
jQuery mobile

jQuery write less, do more.

PhoneGap

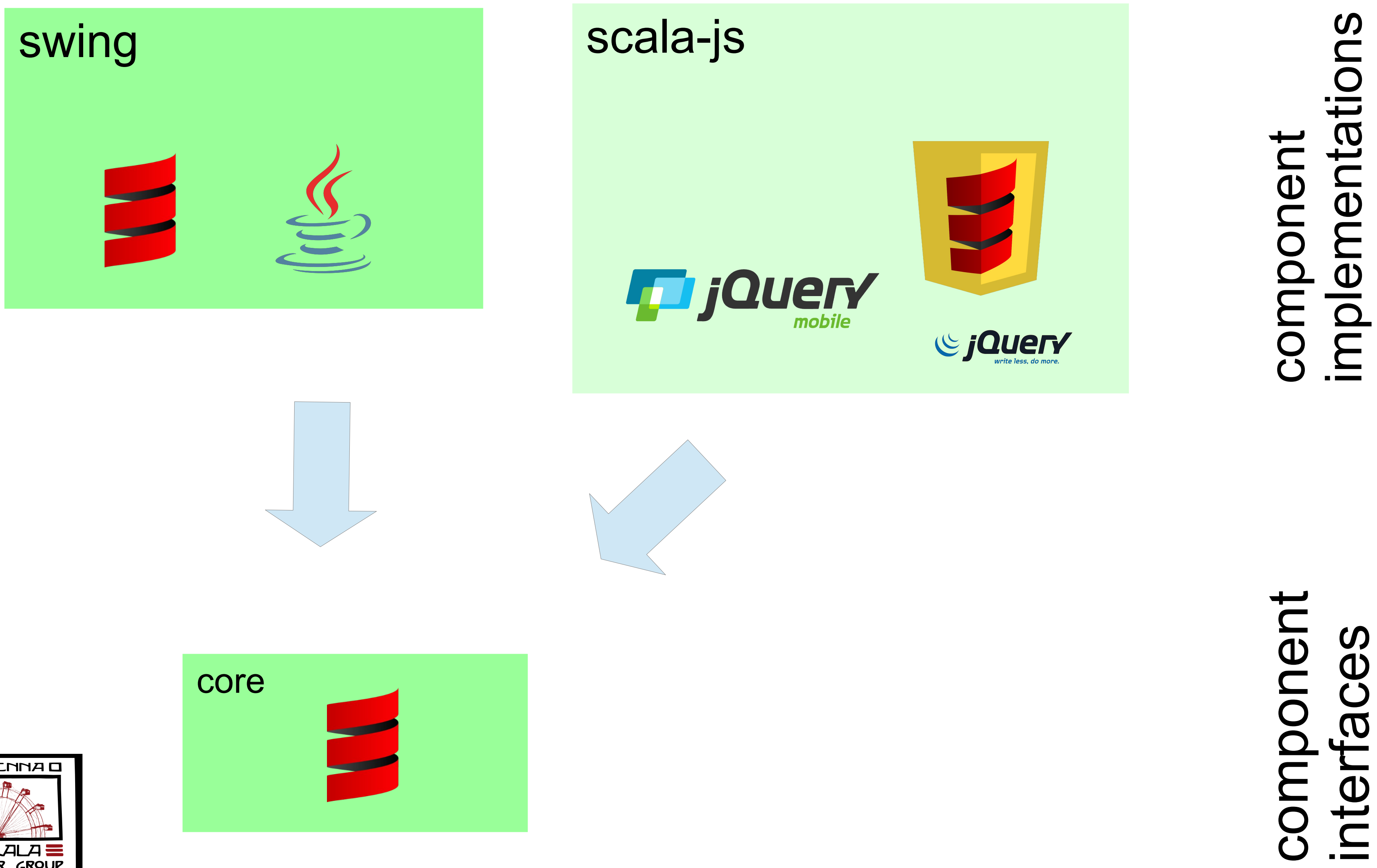
APACHE CORDOVA™

Program Components



VIENNA SCALA USER GROUP SCALA-VIENNA.ORG

Modularisation



GUI Logic (in core module)

```
case class GuiController(
    canvas: DoctusCanvas, selectBox: DoctusSelect[Video],
    startButton: DoctusButton, scheduler: DoctusScheduler,
    framesPerSecond: Int, stageParams: StageParams, allVideos: List[Video]) {

  // Global state
  var index = 0
  var stagesOpt: Option[List[NumberedStage]] = None

  // Fill the select box
  allVideos.zipWithIndex.foreach {
    case (video, index) => selectBox.addItem(index, video)
  }

  // Register callback for start button
  startButton.onClick { () =>
    val video = selectBox.selectedItem
    index = 0
    stagesOpt = Some(VideoCreator.create(List(video), framesPerSecond))
  }

  // Start scheduler
  scheduler.start(() => canvas.repaint, (1000.0 / framesPerSecond).toInt)

  // Register callback for repaint
  canvas.onRepaint { (cg: DoctusGraphics) =>
    {
      val da: DrawArea = DrawArea(Pos(0, 0), Rec(canvas.width, canvas.height))
      stagesOpt match {
        case Some(stages) => {
          stages(index).stage.paint(cg, da, stageParams)
          if (index >= stages.size - 1) stagesOpt = None
        }
        case None => { Intro.stage(index).paint(cg, da, stageParams) }
      }
      index += 1
    }
  }

}
```

Only interfaces

Swing Implementation

```
object VideoSwingMain extends App {

  val canvas = new DoctusPanel()
  val comboBox = new ComboBox(List.empty[Video])
  val startButton = new Button("Start")

  val mf = new MainFrame() {
    // Define the layout of the components
    contents = new BorderPanel() {
      val compPanel = new FlowPanel(comboBox, startButton)
      add(compPanel, BorderPanel.Position.North)
      add(canvas, BorderPanel.Position.Center)
    }
    title = "Akka Workshop Reloaded"
    iconImage = new ImageIcon(getClass.getClassLoader().getResource("icon.png")).getImage
    size = new Dimension(800, 600)
  }


  val framesPerSecond = 20
  val params = StageParams(10, ImageProvider_V01, 0.8, 0.05)
  val videos = AkkaWorkshopResultsVideos.all

  GuiController(
    SwingCanvas(canvas),
    SwingSelect[Video](comboBox),
    SwingButton(startButton),
    SwingScheduler(canvas),
    framesPerSecond,
    params,
    videos)

  mf.visible = true;

}
```

Wrapper are here created

Scala-JS Implementation 1/2

```
<!DOCTYPE html>
<html>
<head>
<title>canrob</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="js/jquery.mobile-1.4.1.js"></script>
<link rel="stylesheet" type="text/css" href="js/jquery.mobile-1.4.1.css">
</head>
<body>
    <div id="centerDiv" style="padding: 3px; max-width: 900px; margin-left: auto; margin-right: auto;">
        <h1 style="background-image: url('src/main/resources/background.png'); ">
            Top 100 Results of the Scala Akka Workshop Reloaded
        </h1>
        <p>
            <select id="selectBox"></select>
            <button id="startButton">Start</button>
        </p>
        <canvas  id="canvas"  style="padding: 3px; width: 99%; max-width: 900px;">Canvas tag not supported</canvas>
        <p style="padding-bottom: 30px">For details about the workshop
            <a href="http://www.meetup.com/scala-vienna/events/162110952/">see...</a>
        </p>
    </div>
    <script type="text/javascript" src="target/scala-2.10/canrob-scalajs-extdeps.js"></script>
    <script type="text/javascript" src="target/scala-2.10/canrob-scalajs-intdeps.js"></script>
    <script type="text/javascript" src="target/scala-2.10/canrob-scalajs.js"></script>
    <!--
    <script type="text/javascript" src="js/canrob-scalajs-opt.js"></script>
     -->

</body>
</html>
```

Components are all created here

Scala-JS Implementation 2/2

```
object VideoScalajsMain {

  // Comes here on every refresh (update)
  def main(): Unit = {
    // GUI Components form the HTML-Page
    val center: HTMLDivElement = dom.document.getElementById("centerDiv").asInstanceOf[HTMLDivElement]
    val canvas: HTMLCanvasElement = dom.document.getElementById("canvas").asInstanceOf[HTMLCanvasElement]
    val selectBoxElem = dom.document.getElementById("selectBox")
    val selectBox: JQuery = jQuery(selectBoxElem.asInstanceOf[HTMLSelectElement])
    val startButtonElem = dom.document.getElementById("startButton")
    val startButton: JQuery = jQuery(startButtonElem.asInstanceOf[HTMLButtonElement])

    canvas.width = center.clientWidth
    canvas.height = canvas.width * 0.7

    // Some configuration
    val framesPerSecond = 15
    val params = StageParams(10, ImageProvider_V01, 0.7, 0.07)
    val allVideos = AkkaWorkshopResultsVideos.all

    // Wrap the javascript components
    val dcanvas: DoctusCanvas = ScalajsCanvas(canvas)
    val dselectBox: DoctusSelect[Video] = ScalajsSelect[Video](selectBox, (v: Video) => v.text)
    val dstartButton: DoctusButton = ScalajsButton(startButton)
    val dscheduler: DoctusScheduler = ScalajsScheduler(canvas)

    // Start the platform independent controller
    GuiController(dcanvas, dselectBox, dstartButton, dscheduler, framesPerSecond, params, allVideos)

  }

}
```

Components are all created here

Going Mobile

APACHE CORDOVA™

PhoneGap

Plus/Minus/Unknown

Almost only Scala to write

Easy testable

Easy debuggable

No Java libraries can be used

Access to mobile-APIs

Remote access (Akka remote)