Scala for GUI

Aurelijus Banelis

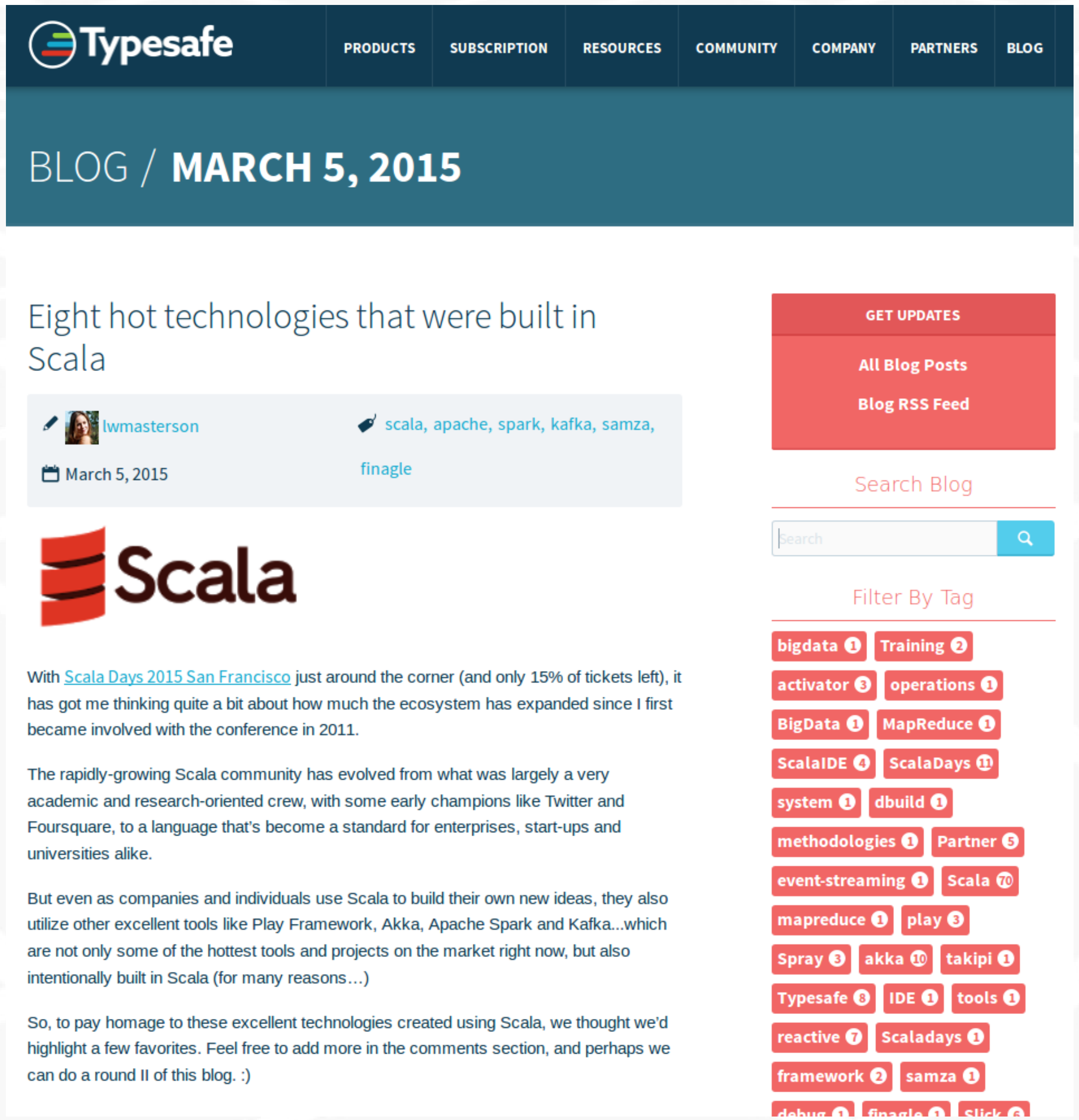
About me

Aurelijus Banelis

aurelijus@banelis.lt
aurelijus.banelis.lt

Using Scala for personal project

Scala for GUI? Really?



PRODUCTS SUBSCRIPTION RESOURCES COMMUNITY COMPANY PARTNERS BLOG

BLOG / **MARCH 5, 2015**

Eight hot technologies that were built in Scala

lwmasterson

March 5, 2015

scala, apache, spark, kafka, samza, finagle

With Scala Days 2015 San Francisco just around the corner (and only 15% of tickets left), it has got me thinking quite a bit about how much the ecosystem has expanded since I first became involved with the conference in 2011.

The rapidly-growing Scala community has evolved from what was largely a very academic and research-oriented crew, with some early champions like Twitter and Foursquare, to a language that's become a standard for enterprises, start-ups and universities alike.

But even as companies and individuals use Scala to build their own new ideas, they also utilize other excellent tools like Play Framework, Akka, Apache Spark and Kafka...which are not only some of the hottest tools and projects on the market right now, but also intentionally built in Scala (for many reasons…)

So, to pay homage to these excellent technologies created using Scala, we thought we'd highlight a few favorites. Feel free to add more in the comments section, and perhaps we can do a round II of this blog. :)

GET UPDATES

All Blog Posts

Blog RSS Feed

Search Blog

Search

Filter By Tag

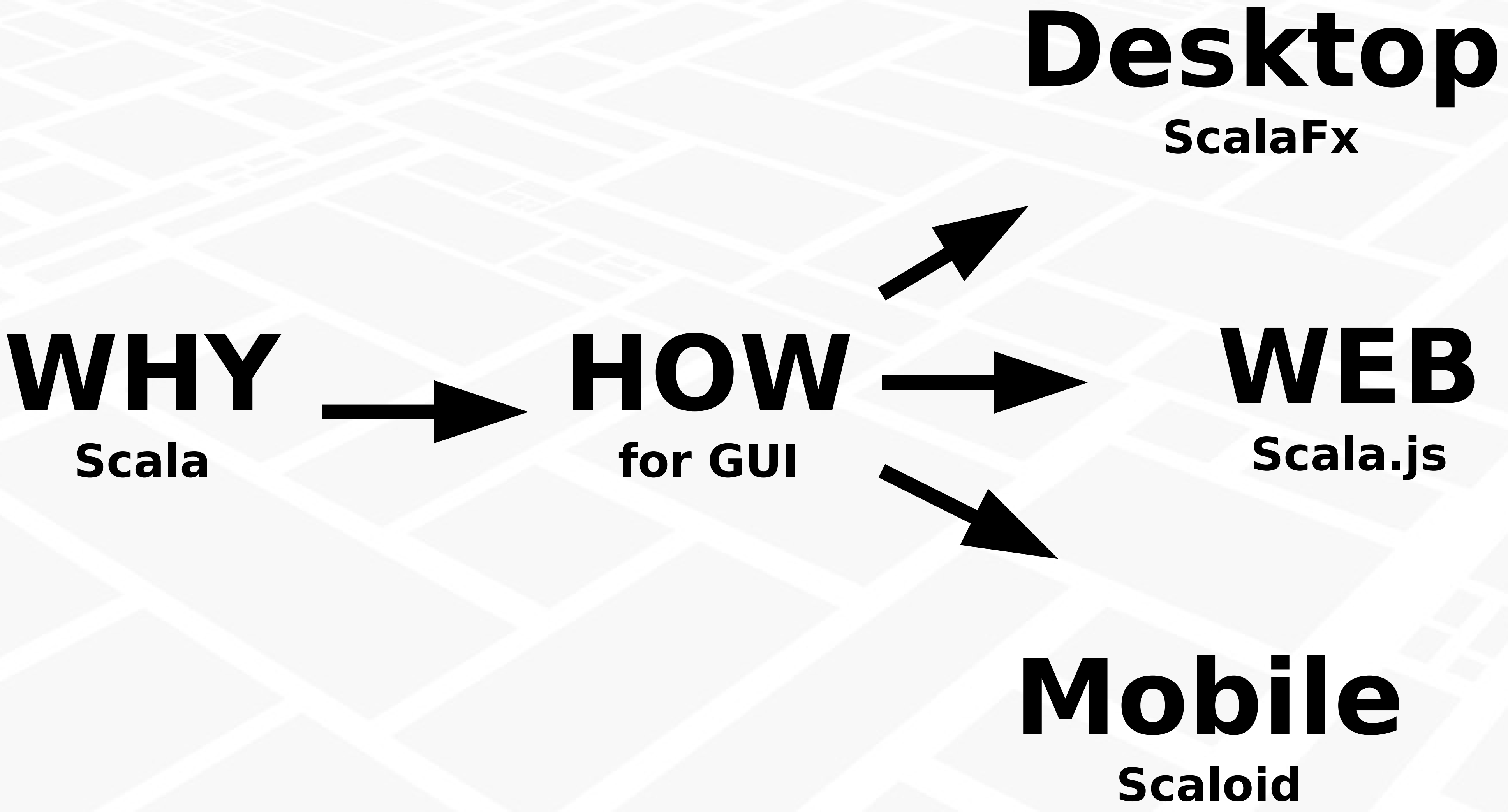
bigdata 1 Training 2 activator 3 operations 1 BigData 1 MapReduce 1 ScalaIDE 4 ScalaDays 11 system 1 dbuild 1 methodologies 1 Partner 5 event-streaming 1 Scala 70 mapreduce 1 play 3 Spray 3 akka 10 takipi 1 Typesafe 8 IDE 1 tools 1 reactive 7 Scaladays 1 framework 2 samza 1 debug 1 finagle 1 Slick 6

https://typesafe.com/blog/eight-hot-technologies-that-were-built-in-scala

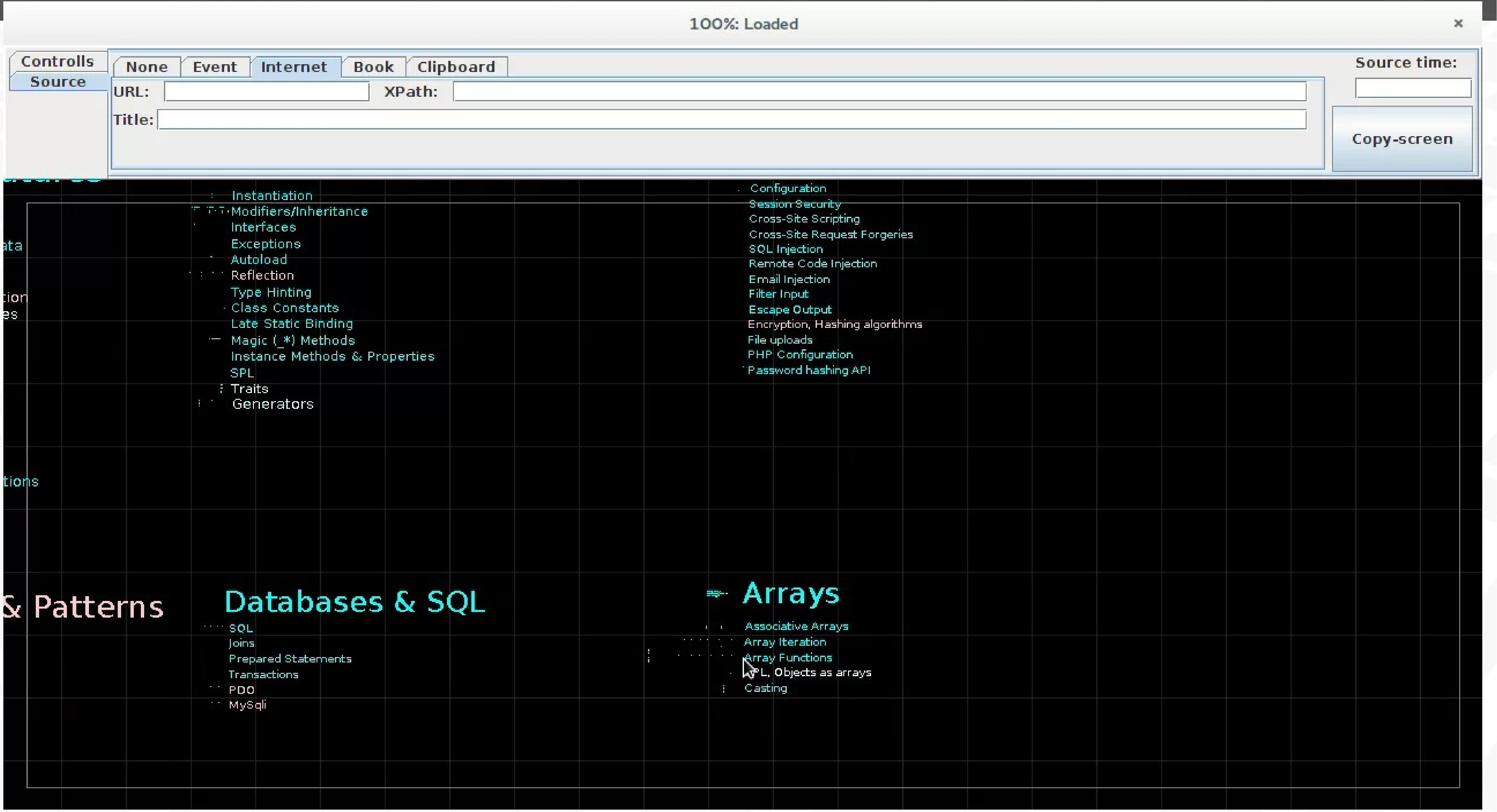
GUI

not listed

You will learn



Context: Knowledge management tool



100%: Loaded

ata
tion
es
tions
& Patterns

Instantiation
Modifiers/Inheritance
Interfaces
Exceptions
Autoload
Reflection
Type Hinting
Class Constants
Late Static Binding
Magic (_*) Methods
Instance Methods & Properties
SPL
Traits
Generators

Configuration
Session Security
Cross-Site Scripting
Cross-Site Request Forgeries
SQL Injection
Remote Code Injection
Email Injection
Filter Input
Escape Output
Encryption, Hashing algorithms
File uploads
PHP Configuration
Password hashing API

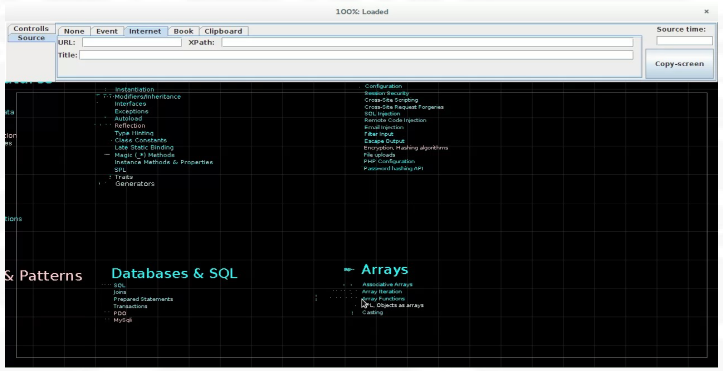
Databases & SQL

SQL
Joins
Prepared Statements
Transactions
PDO
MySqli

Arrays

Associative Arrays
Array Iteration
Array Functions
SPL, Objects as arrays
Casting

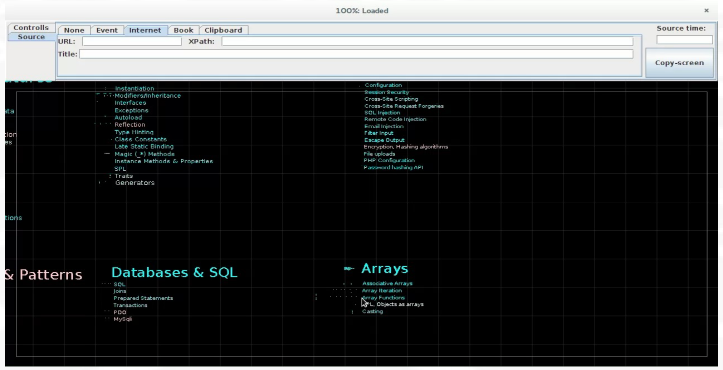
Context: When GUI matters



GUI = Added value

Prevent cognitive overhead
Boost visual memory
Faster perception

Context: When GUI matters



GUI = Added value

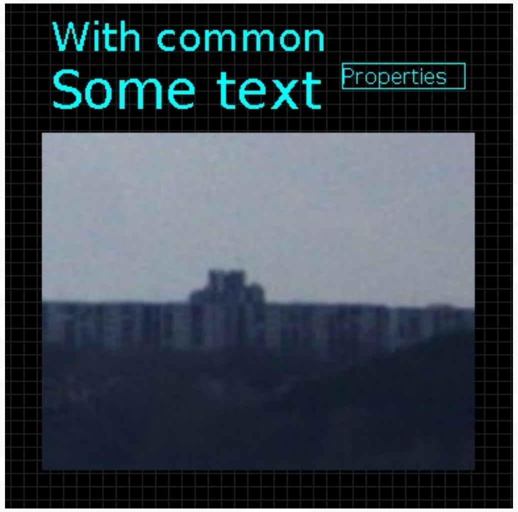
Prevent cognitive overhead
Boost visual memory
Faster perception

Java 6 + Swing

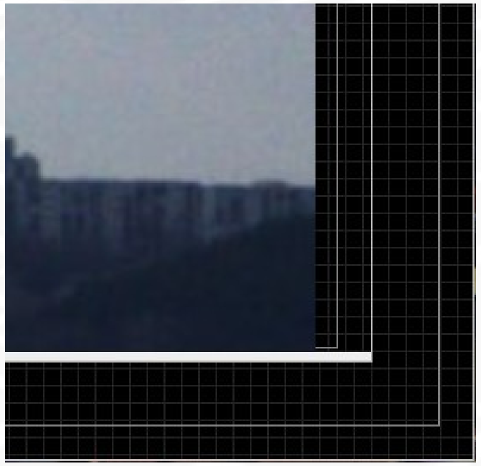
Just get things done

Personal use: **more like <u>prototype</u>**

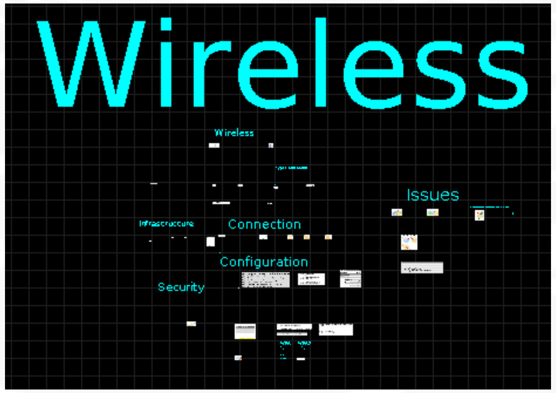
WHY: Java → Scala



Common

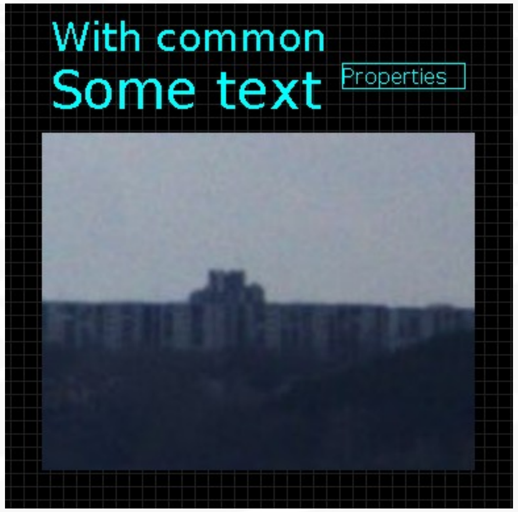


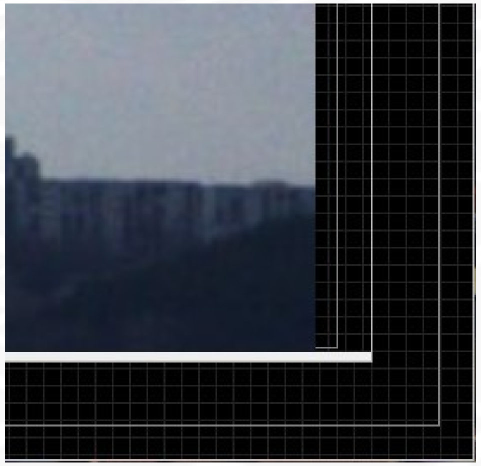
Async

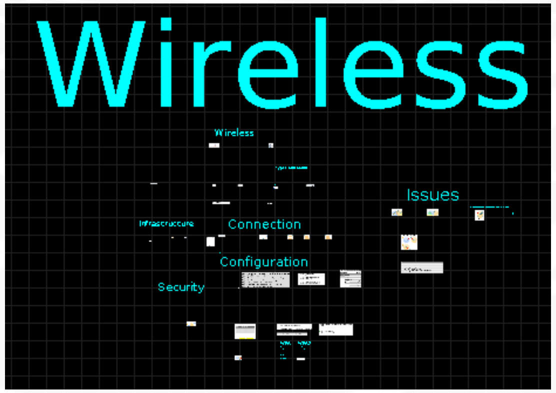


Zoom

WHY: Java → Scala



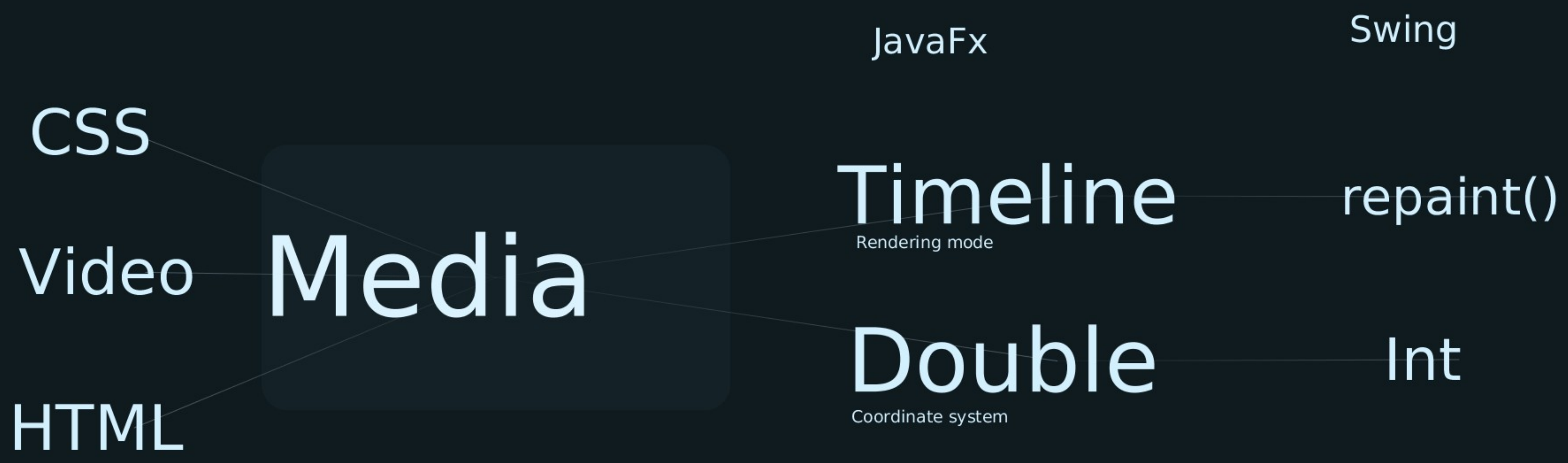




Common **Async** **Zoom**

Traits **Immutability** **@tailrec**

WHY: Swing → JavaFx



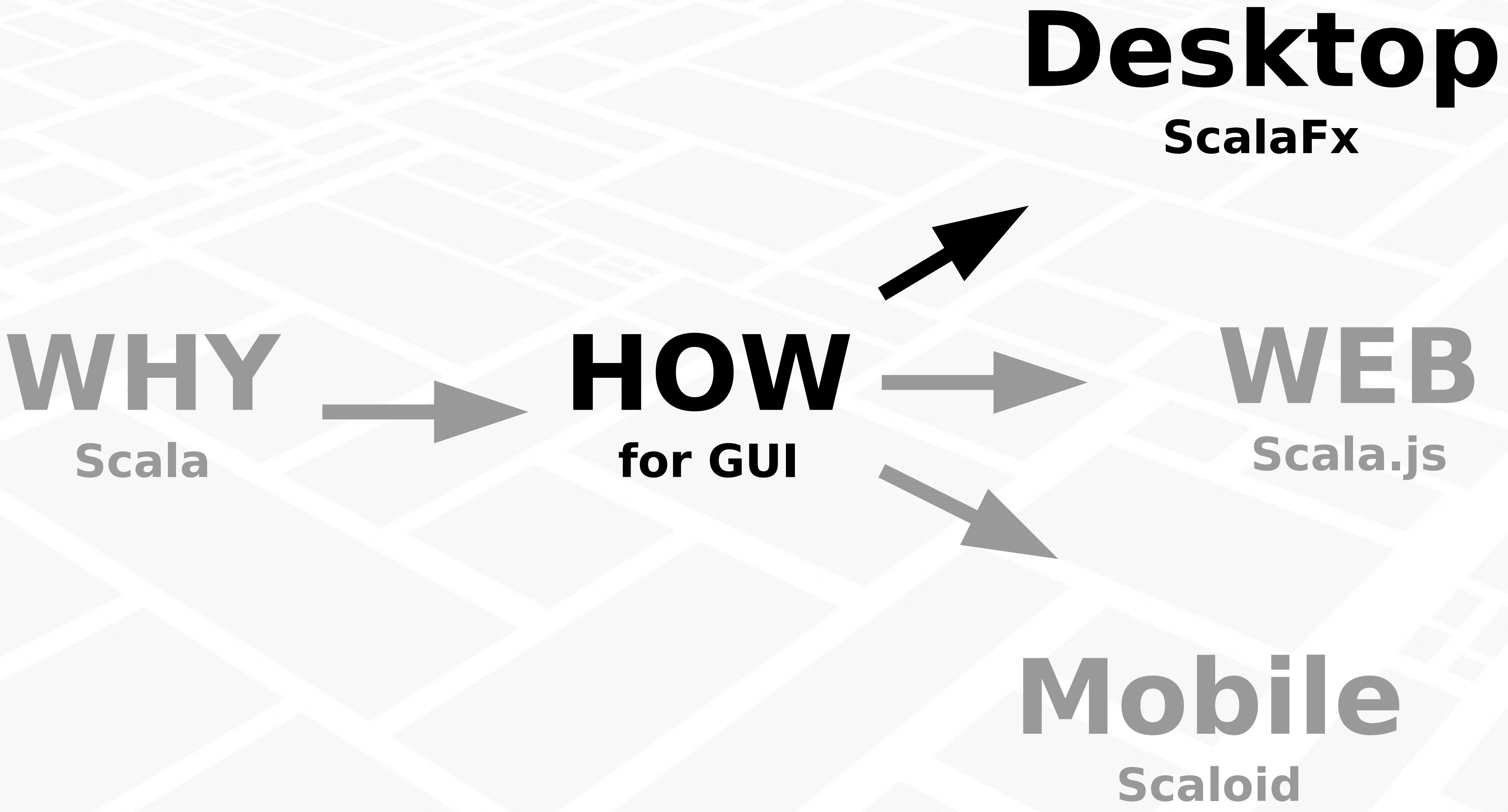
Context: JavaFx example

HOW: JavaFx ↔ ScalaFx

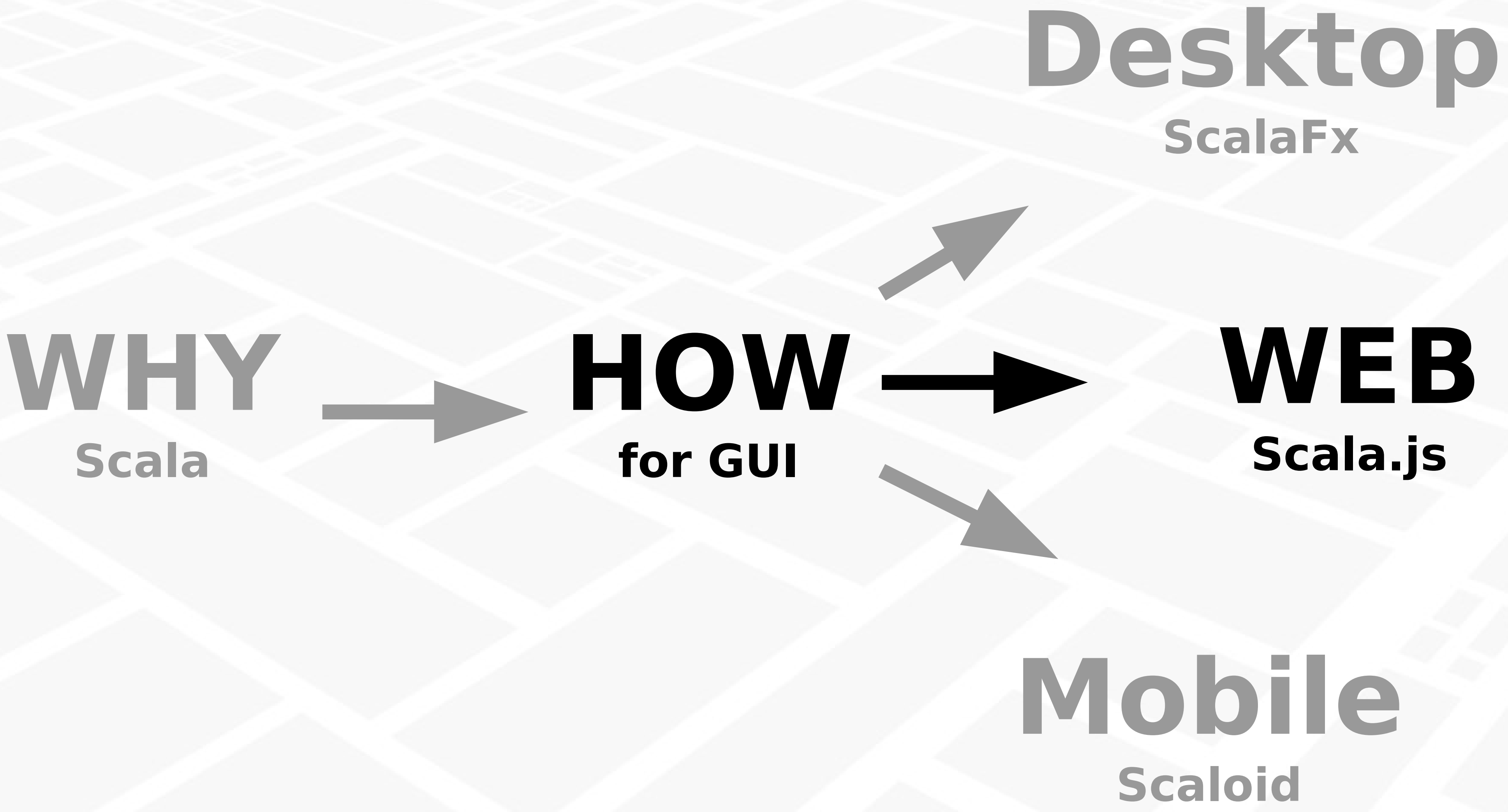
```
class Label(val _text: String)
extends RichJPane
with ViewableNode
with HaveOperations
with DragableNode[jp]
with ZoomableNode[jp]
with ScalableElement[jp]
with Data
with Transformable[Label]
with EditableNode {

class Image
extends RichImageView
with ViewableNode
with HaveOperations ...
```

```
mousePressed += beginDrag

mouseReleased += endDrag

mouseDragged += {
 (e: MouseEvent) =>
  if (beingDragged) {
   endDrag(e)
   beginDrag(e)
  }
}
```

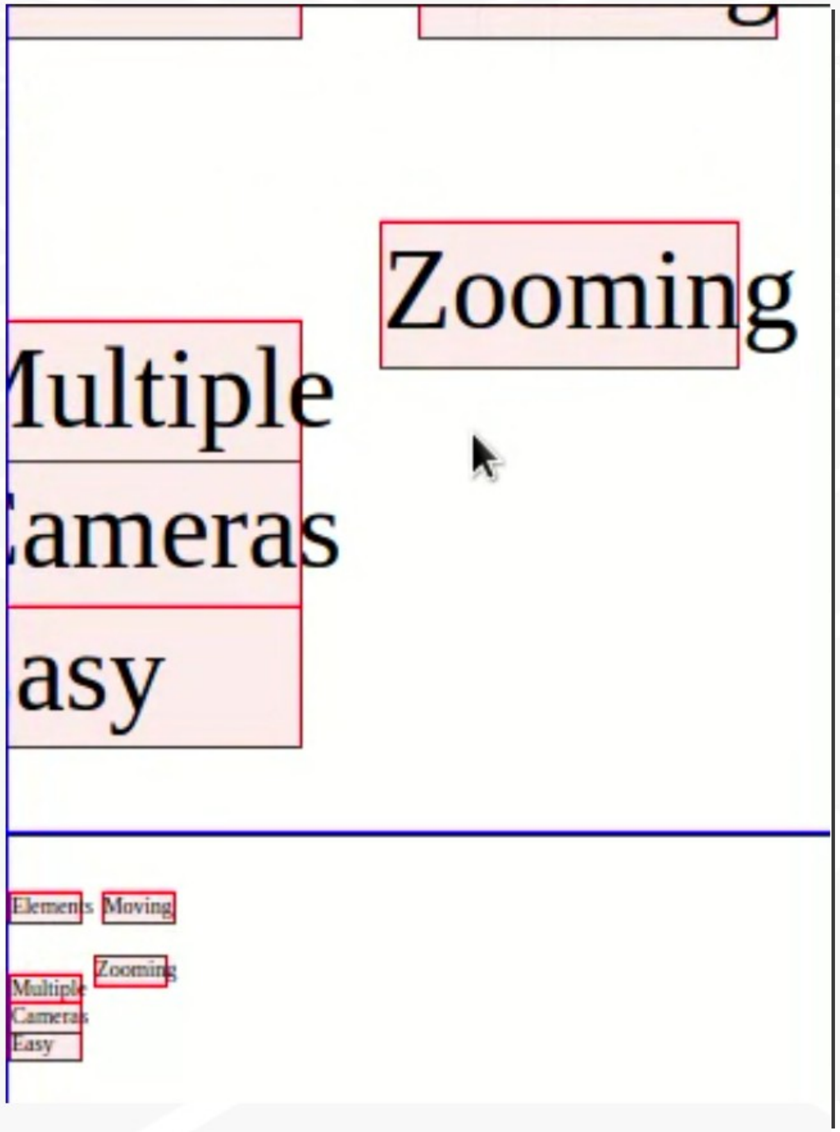
HOW: Multi module project



HOW: Multi module project



HOW: Scala.js + React



Scala.js

Compile Scala → JavaScript
Type safety for complex GUI
Access to native JavaScript

React

Forces immutability
Direct data flow
Virtual DOM

https://github.com/japgolly/scalajs-react

HOW: Scala.js + React

State

Logic **Flow** Render

Events

HOW: Scala.js + React

State

Logic Flow Render

Events

Deeper structure

```
case class State(camera: Camera) {
  def inCamera(converter: Camera => Camera) =
    copy(camera = converter(camera))
}
```

Immutable

HOW: Scala.js + React

State

Logic Flow **Render**

Events

Parameters

```
.render { (P, S, B) =>
  <.span(
    P.element.text,
    ^.`class` := "dragable noselect",
    ^.left := (P.element.x - P.camera.x) / P.camera.scale,
    ^.fontSize := s"${1.0 / P.camera.scale}em",

    ^.onMouseDown ==> P.receive,
    ^.onTouchStart ==> P.receive,
  )
}
```

Callbacks

HOW: Scala.js + React

State

Logic Flow Render

Events

Event propagation

```
def beginDrag(e: PointerEvent): Unit =
  preventDefault(e) {
    selectedElement(e) match {
     case Some(element) =>
      elements.Dragging.begin(element, e)
     case None =>
      view.Dragging.begin(e)
    }
}
```

```
def touch(reactEvent: ReactMouseEvent): Unit =
 event(reactEvent) match {
  case e: TouchStart if e.touchEvent.touches.length == 1 =>
    preventDefault(reactEvent) {
      elements.Dragging.begin(element, e.touchEvent.touches(0))
    }
...
```

Different parameters

HOW: Scala.js + React

State

Logic Flow Render

Events

All of same type

```
def drag(event: ScreenPosition): T =
  moveElement(event) andThen savePosition(event)
```

Easy combine

```
private def savePosition(position: ScreenPosition): T = {
 State:State =>
  state inSelected (_ inElements (
     _ withPosition inCamera(state, position)
  ))
}
```

Returns function

Rendering techniques

State

Logic Flow Render

Events

Functional style

Transformation based

JavaFx

Thread updates

Observers

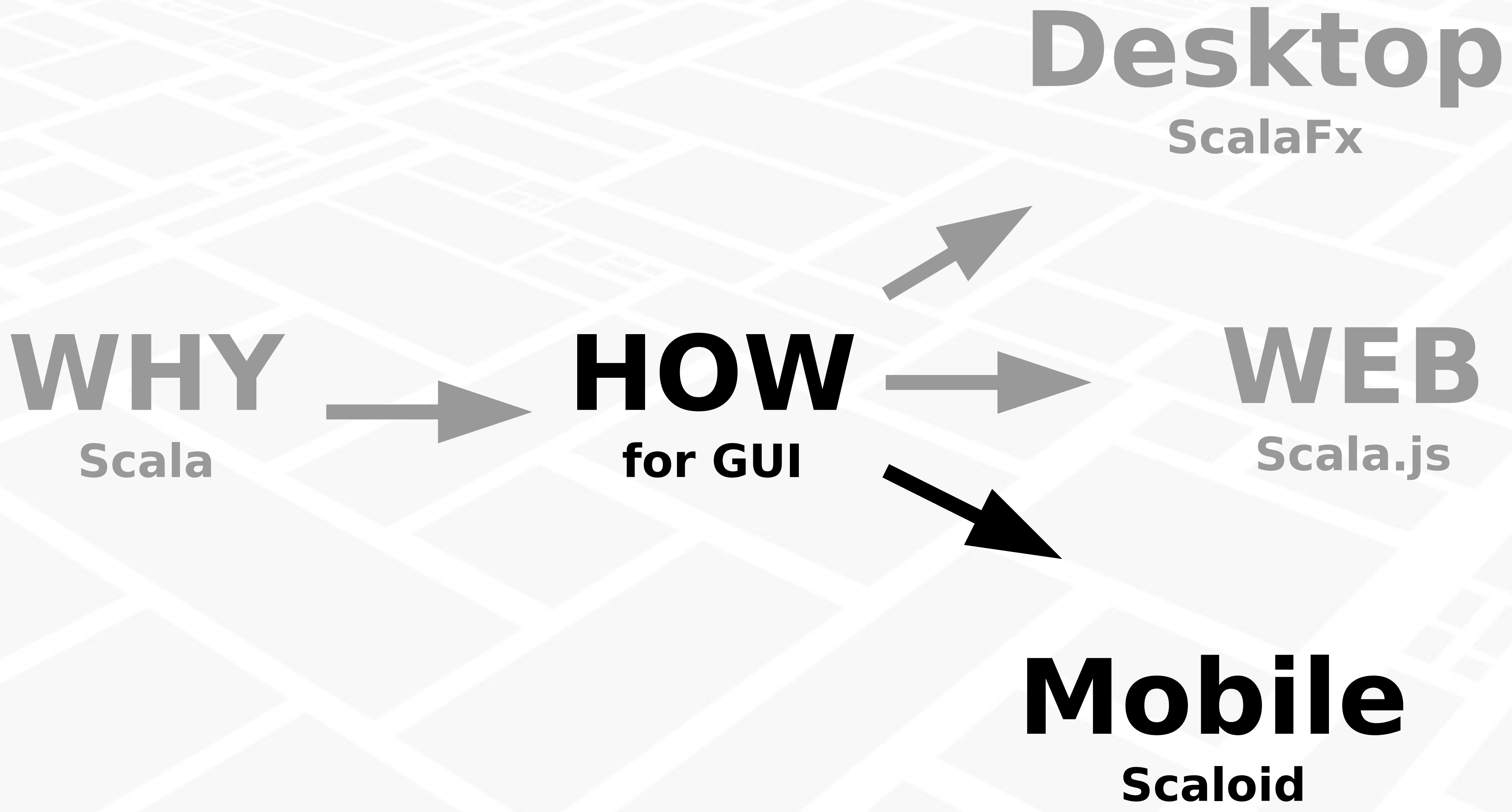
Swing

Event dispatch

Force update

From OS/callback

HOW: Multi module project



Reuse in Multi module project

Use tools

Scala.js — **crossProject**

Android — **scaloid**

Not user-friendly

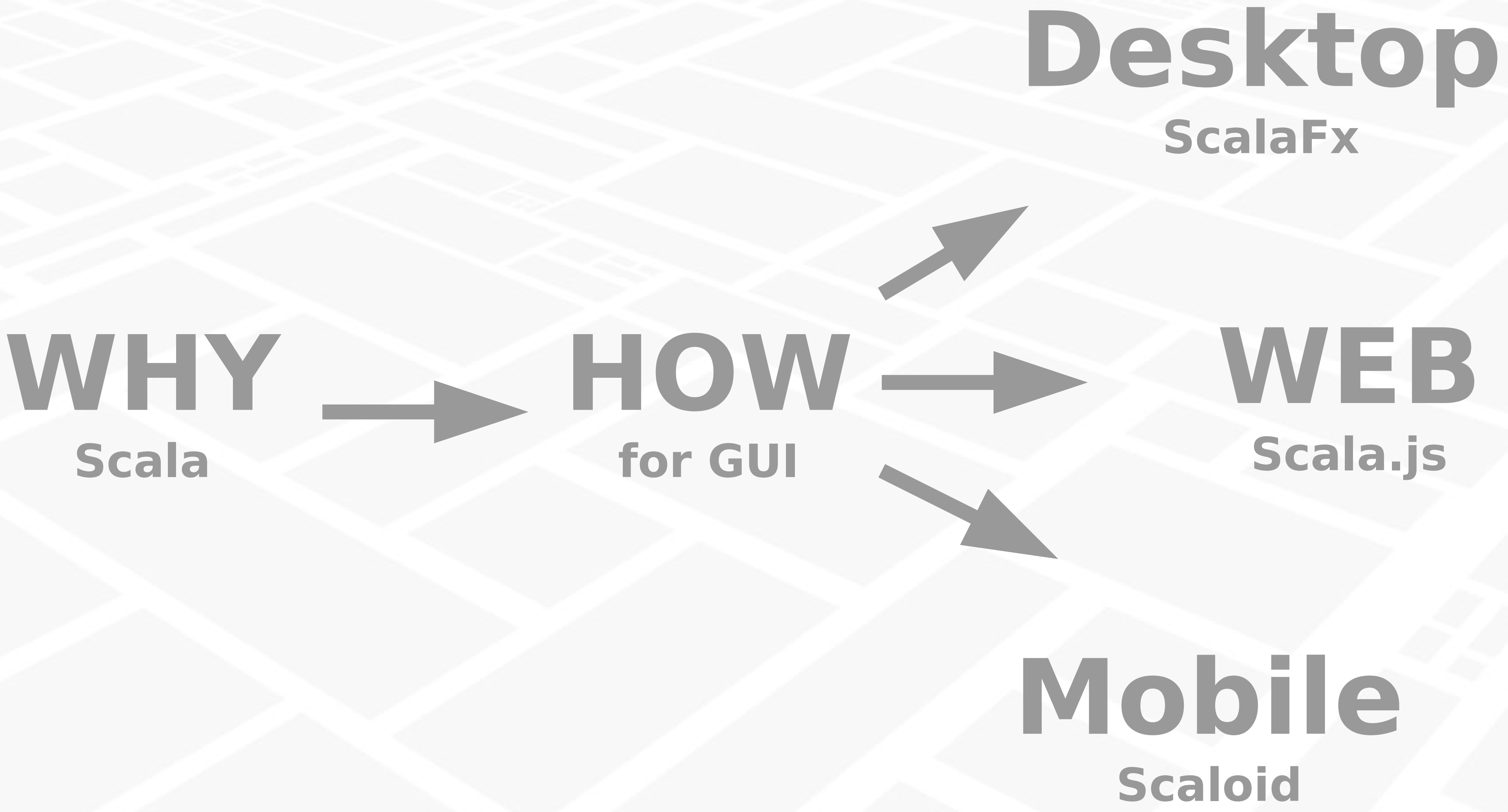
SBT vs IDEA

Copy with symlinks

Dirty, but works

Code completion

Questions?


WHY
Scala
HOW
for GUI
Desktop
ScalaFx
WEB
Scala.js
Mobile
Scaloid

References and useful links

•http://auginte.com/
•http://www.scala-lang.org/
•http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html
•https://github.com/scalafx/scalafx
•http://www.scala-js.org/
•https://github.com/japgolly/scalajs-react
•https://www.youtube.com/watch?v=KVZ-P-ZI6W4
•http://www.scala-js.org/doc/sbt/cross-building.html