

Scala Training 1

Why Scala + Values, variables, lazy and def

A large, stylized number 1 is centered on a red ribbon that spirals around it. The ribbon has a 3D effect with shadows. The background of the entire slide is red with a repeating pattern of small, faint icons representing various electronic devices like smartphones, cameras, and game consoles.

1

Contents

- Why Scala
- Values, variables, lazy and def
- Workshop



Why Scala

Why Scala

🔍 why scala|

🔍 why scala - Google Search

👎 why scala is **bad**

🔍 **scala** for java

🔍 why scala is **better** than java

🔍 why scala **programming**

👍 why scala is **awesome**

Why Scala

OOP - Functional

- Promotes and make it easier to deal with immutability
 - Immutable collection APIs by default
 - Case Classes

```
case class Message(text: String)

val helloMessage = Message("Hello")

val byeMessage = helloMessage.copy(text = "Bye")
```

- Higher-order functions

Why Scala

JVM - Async

- JVM performance and ecosystem
- Type inference (avoid Java verbosity)

```
case class Message(text: String) {  
  def messageLength = text.length // inference Int  
}
```

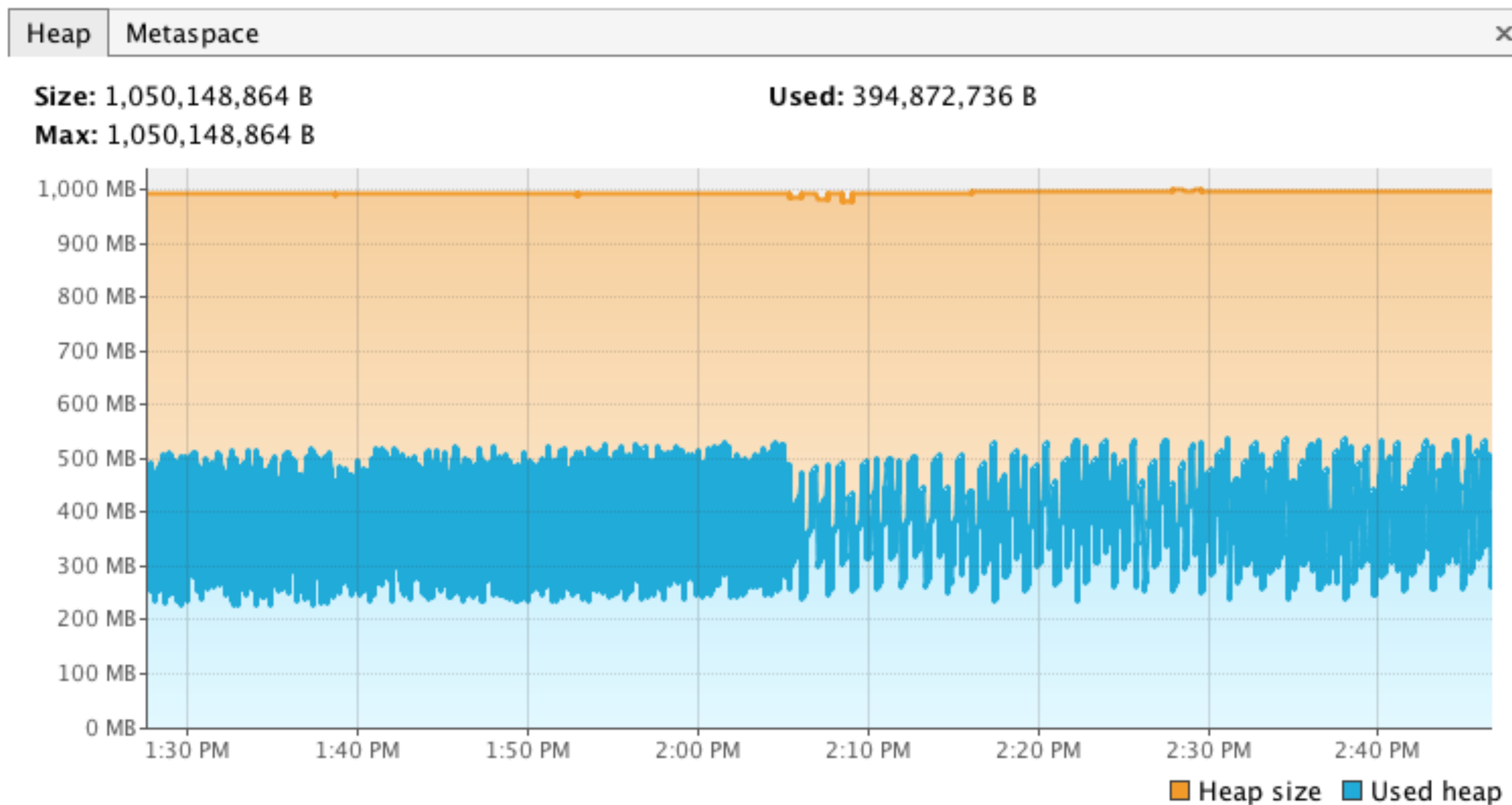
- Easy to use async APIs (Future)

💡 Bonus tracks:

- [Scala functional programming features and more!](#)

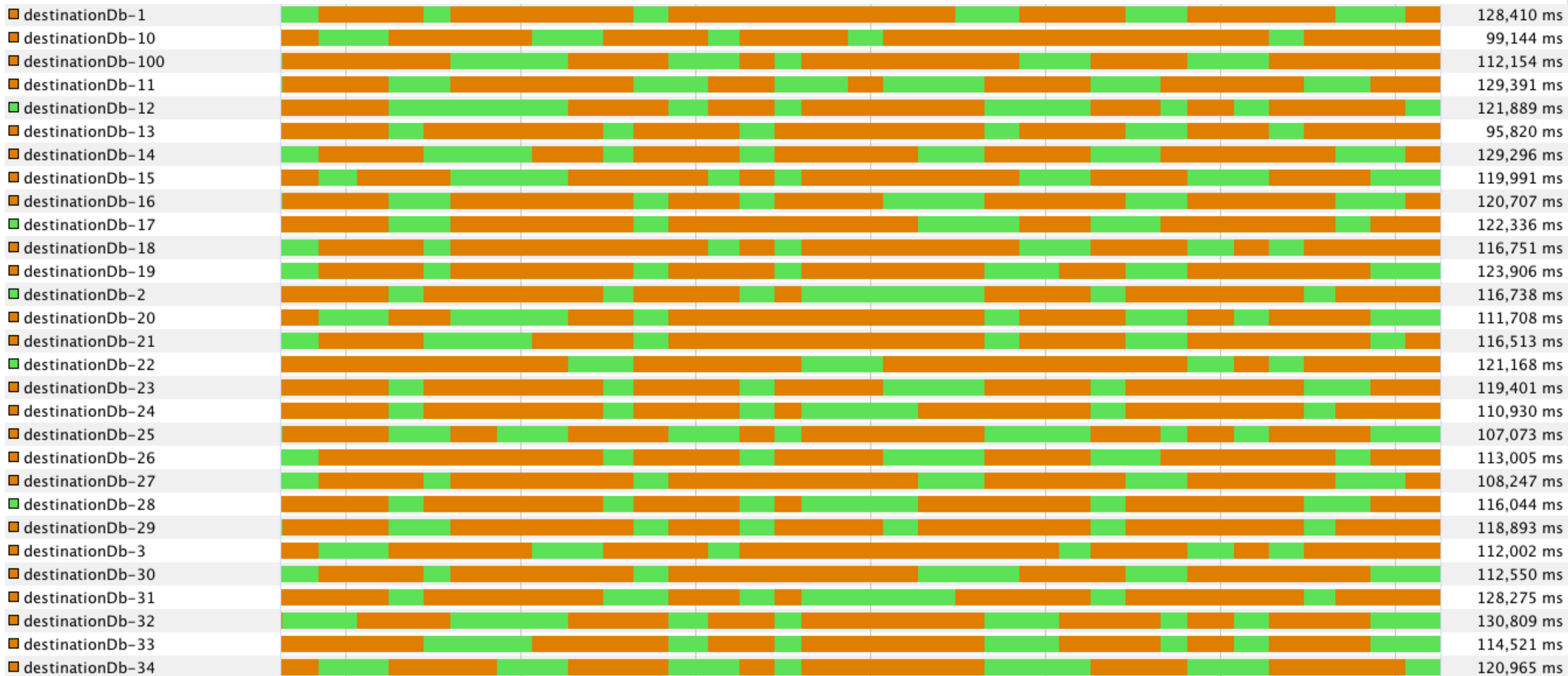
Why Scala

JVM



Why Scala

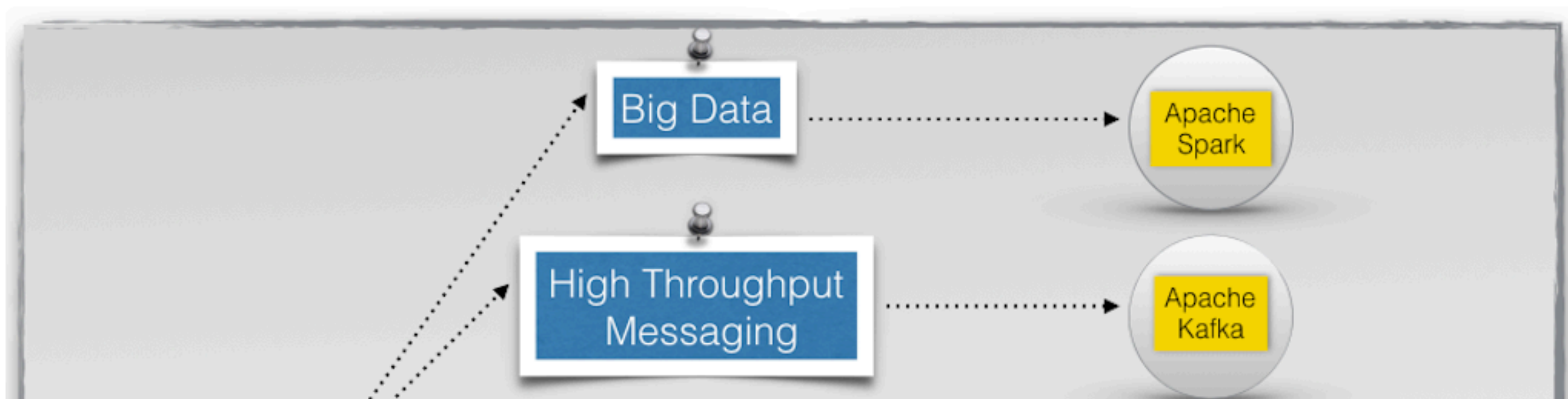
JVM



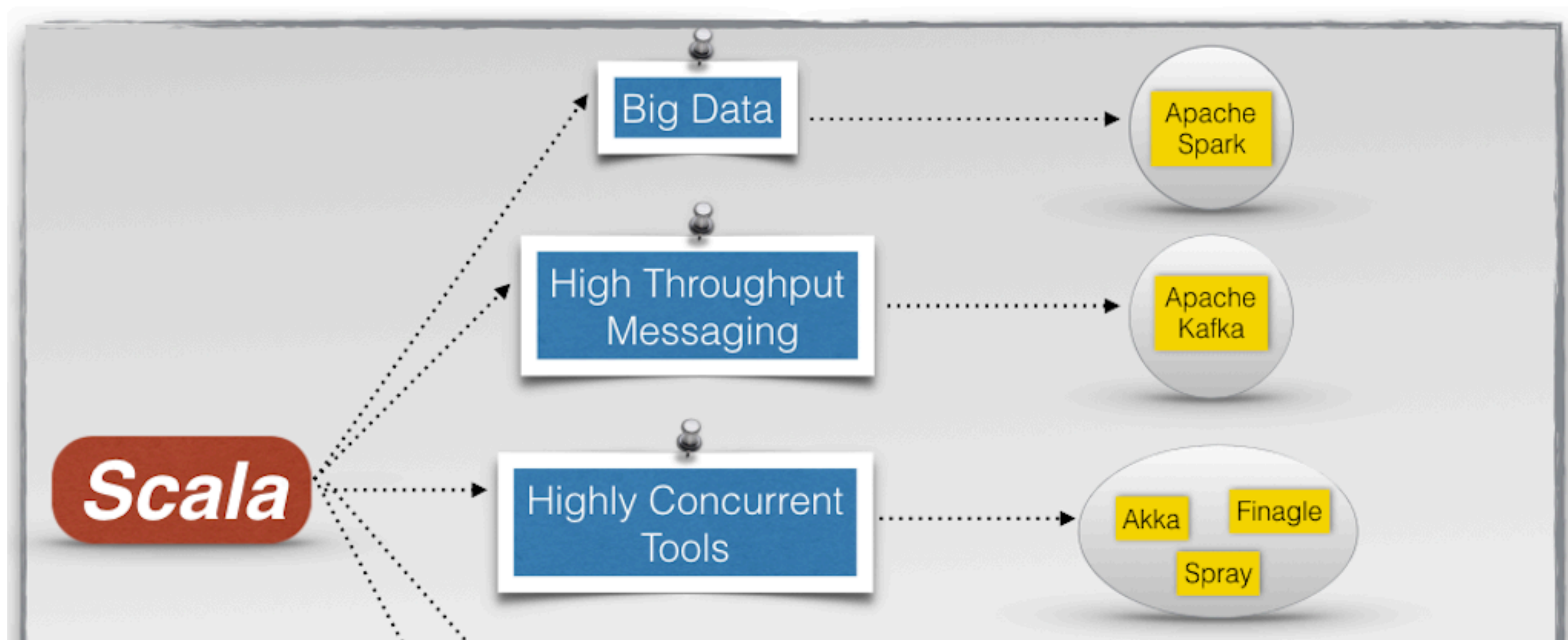
Why Scala



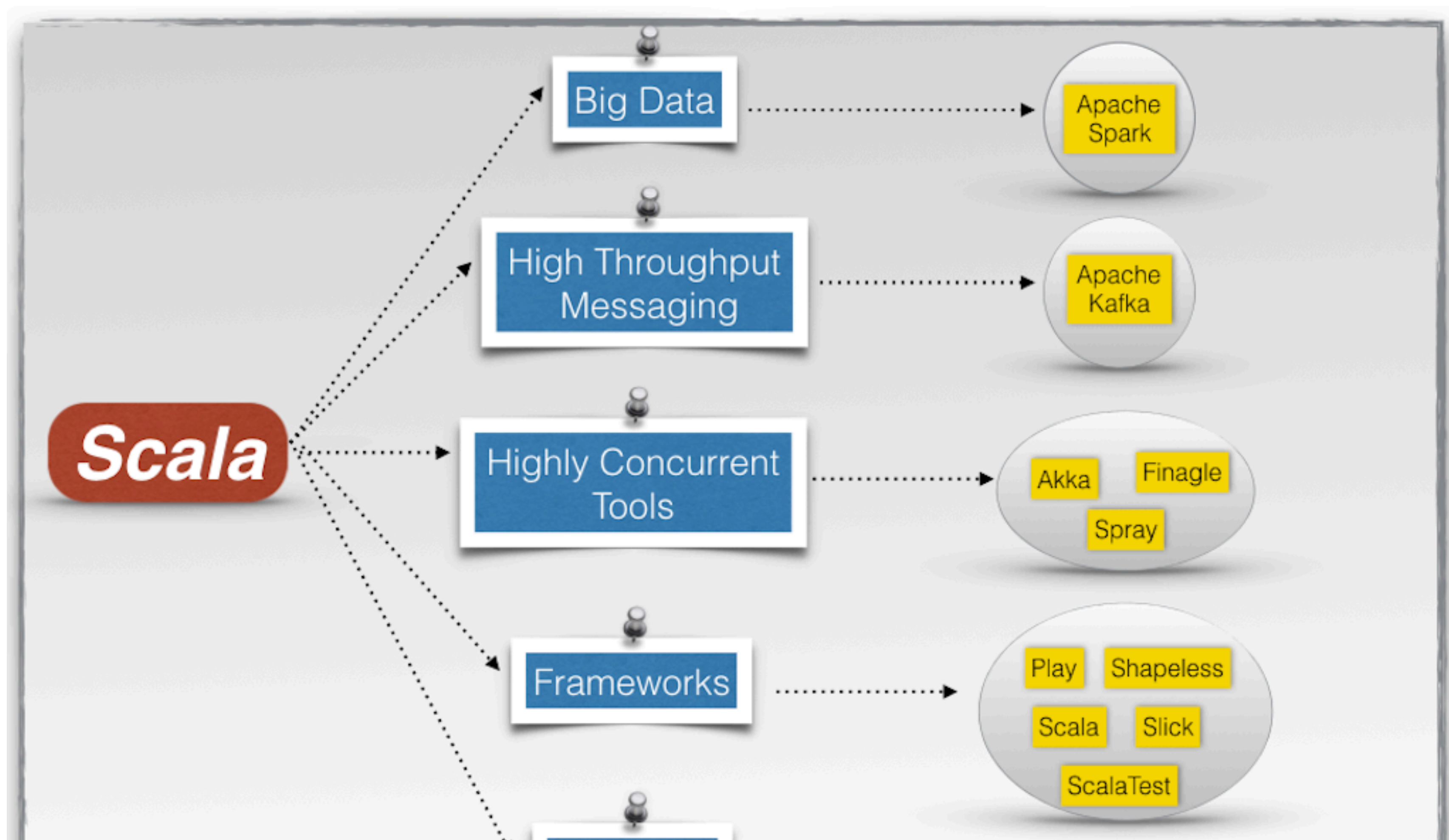
Why Scala



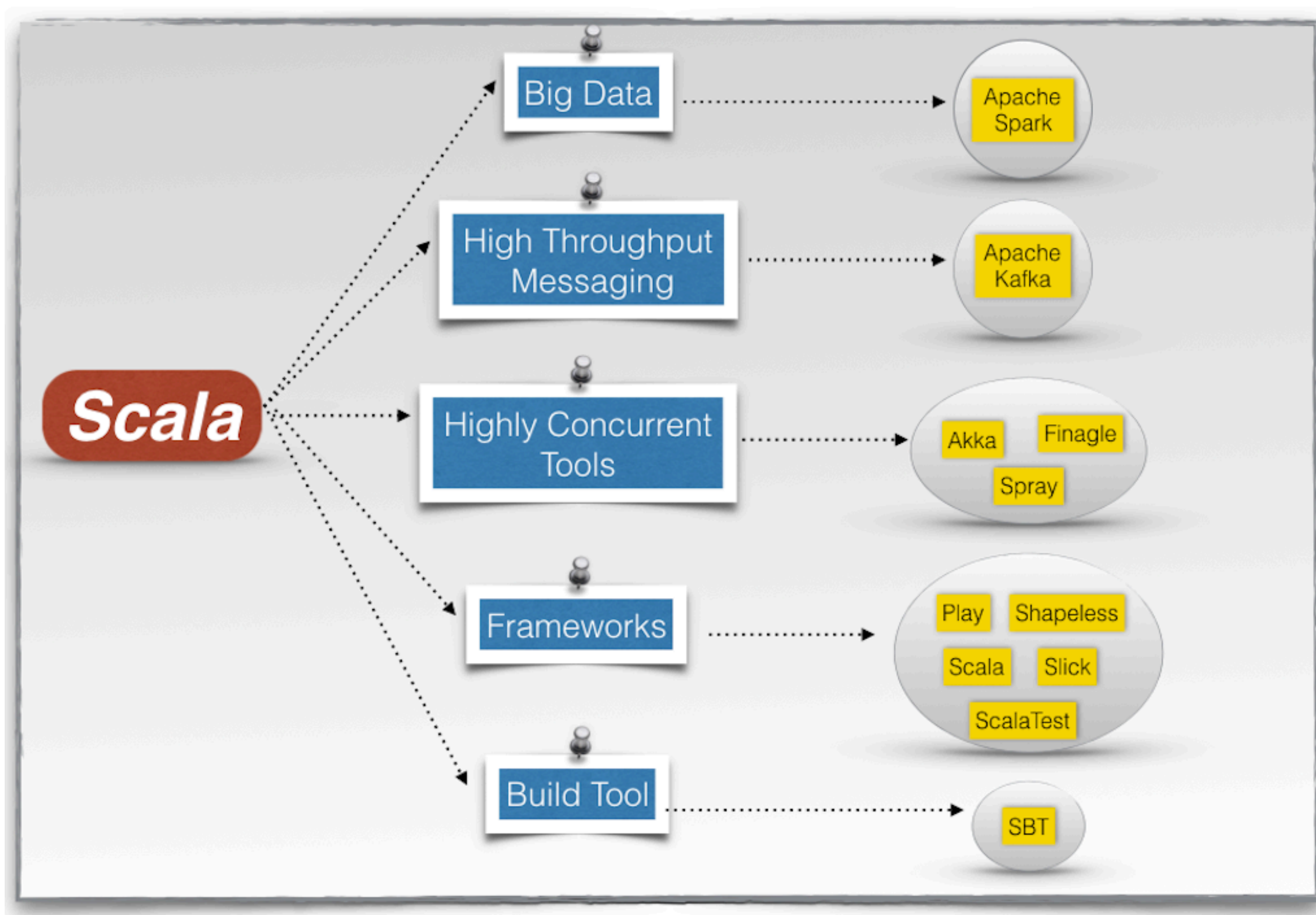
Why Scala



Why Scala



Why Scala



Values, variables, lazy and def

Variables

```
case class Message(text: String)

var questionMessage: Message = Message("Ola k ase?")

questionMessage = Message("No ase nada?")
```

Values

```
val helloMessage = Message("Hello")

helloMessage = Message("Bye") // error: Reassignment to val
```

Values

```
val helloMessage = Message("Hello")
```

```
helloMessage = Message("Bye") // error: Reassignment to val
```

Lazy values

```
val allMessages = Seq(helloMessage, byeMessage, questionMessage)
```

```
val randomMessageAsVal: Message = allMessages(Random.nextInt(allMessages.size))
```

```
lazy val randomMessageAsLazyVal = allMessages(Random.nextInt(allMessages.size))
```

Lazy values

```
val allMessages = Seq(helloMessage, byeMessage, questionMessage)
```

```
val randomMessageAsVal: Message = allMessages(Random.nextInt(allMessages.size))
```

```
lazy val randomMessageAsLazyVal = allMessages(Random.nextInt(allMessages.size))
```

Functions

```
def randomMessageAsDef: Message = allMessages(Random.nextInt(allMessages.size))
```

Test

```
randomMessageAsVal  
randomMessageAsVal
```

```
randomMessageAsLazyVal  
randomMessageAsLazyVal
```

```
randomMessageAsDef  
randomMessageAsDef
```



Workshop

Workshop

- https://github.com/letgoapp/scala_course/blob/2-var_val_def_lazy/doc/lessons/2-values-variables-lazy.md

Bonus tracks

- <http://www.scala-lang.org/old/node/8610>
- <https://github.com/mushtaq/scala-basics-9/tree/master/worksheets/day1>
- <https://www.linkedin.com/pulse/scala-syntax-7-images-michele-sciabarr%C3%A0>