

Why you might like **Scala.js**

Li Haoyi, Scaladays 17-March-2015

0 Who Am I?

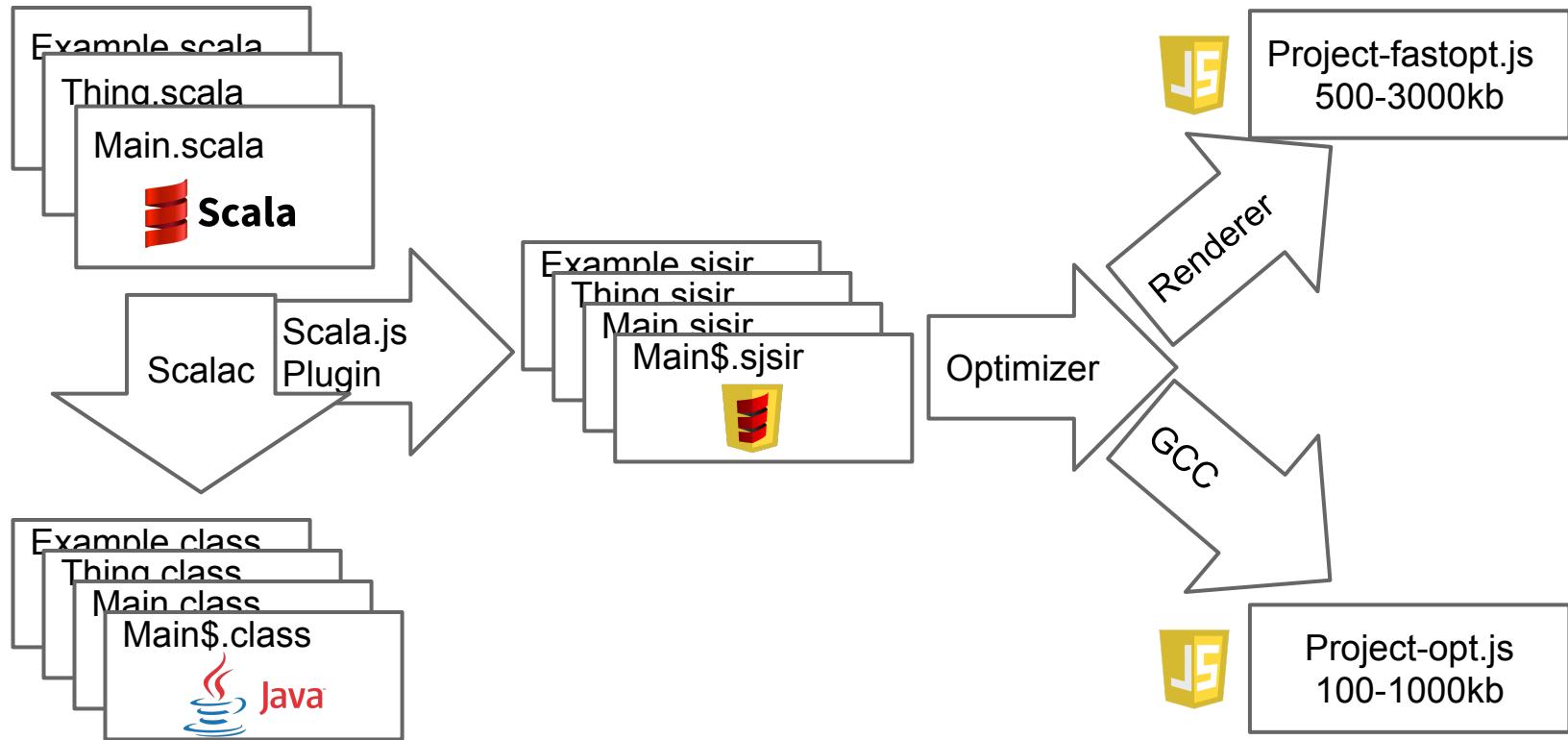
- Li Haoyi
- Work at Dropbox
 - Fixing legacy Python/Coffeescript
 - Writing legacy Python/Coffeescript
- Do a lot of free-time Scala work
- Early tester/contributor for Scala.js



1 What is Scala.js

- "Scala.js is a compiler that converts Scala code into the equivalent, executable Javascript"
- Write Scala, not Javascript
- Compiler handles the rest

1 What is Scala.js



Live Demo

2 Notes from the Demo

- It works seamlessly!
- Really nice IDE experience
- Compiled executable is reasonable

3 Why should I care?

- Depends on who “I” am...
- Who am I?

3 Why should I care?

- Depends on who “I” am...
- Who am I?



4 “I” am a...

- Scala dev, who works on web apps
- Scala dev, who's never touched a web app
- Compiler writer, who likes doing fancy optimizations
- Going to ignore: Javascript Developer, CTO, Professor, Newbie Programmer...

4 “I” am a...

- Scala dev, who works on web apps
- Scala dev, who's never touched a web app
- Compiler writer, who likes doing fancy optimizations

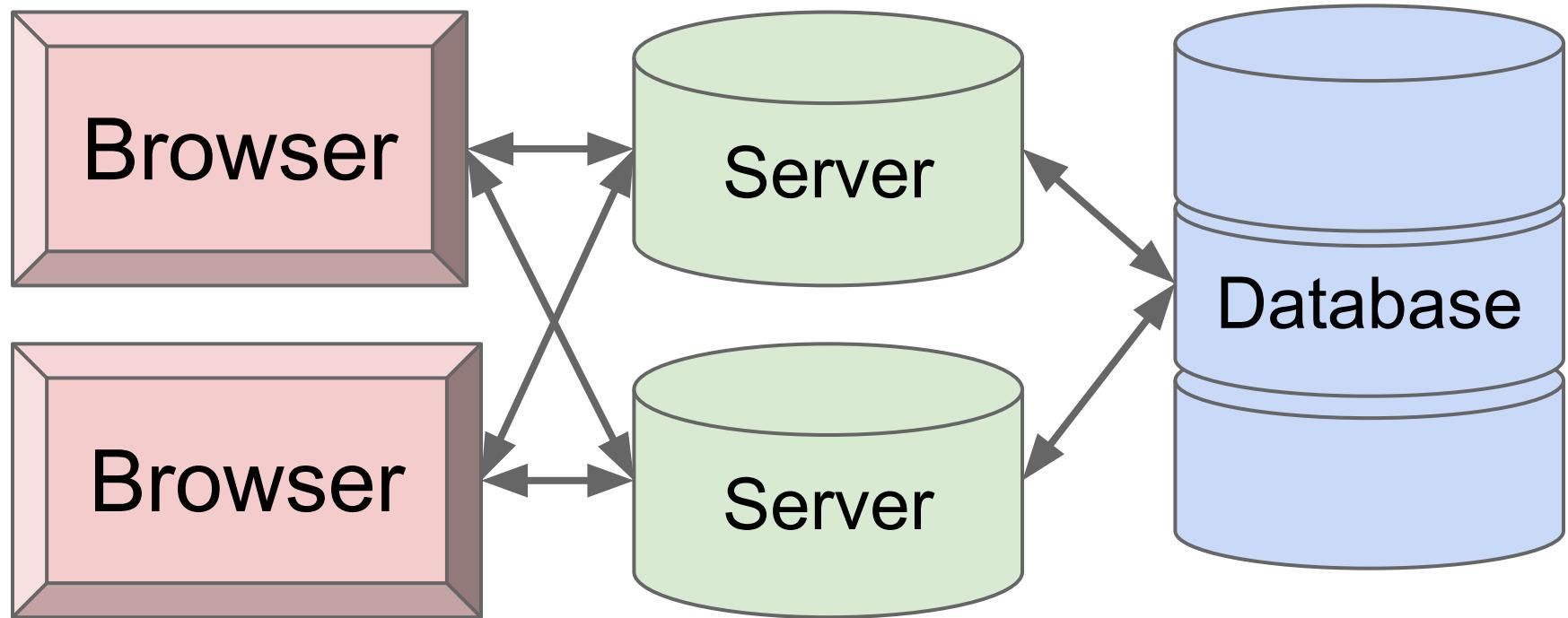
5.1 I am a Scala developer

- I sometimes work on Web Applications
 - Making the world more open and connected
 - Help people watch events unfold, in real time.
 - Cataloguing the world's knowledge
- *Why should I care about Scala.js?*

5.2 What is a web application?

- Client-server model
- Usually written in two (or more) languages
 - Scala on the server?
- Complicated!

5.3 What is a web application?



5.4 What's wrong with Web Apps?

- No code re-use!
 - Find two sets of libraries to do the same thing
 - Learn two languages
 - Write your algorithms twice
- Alternative: pepper awkward/slow RPCs everywhere
 - Also known as “API first” design

5.5 What's wrong with Web Apps?

- Everything `String`/`Map[String, String]`
 - URLs
 - Ajax arguments/return-value
- Compiler cannot help you!
 - `document.getElementById("my-id")`
 - `document.getElementsByClassName("my-cls")`
 - `throw new Exception()`

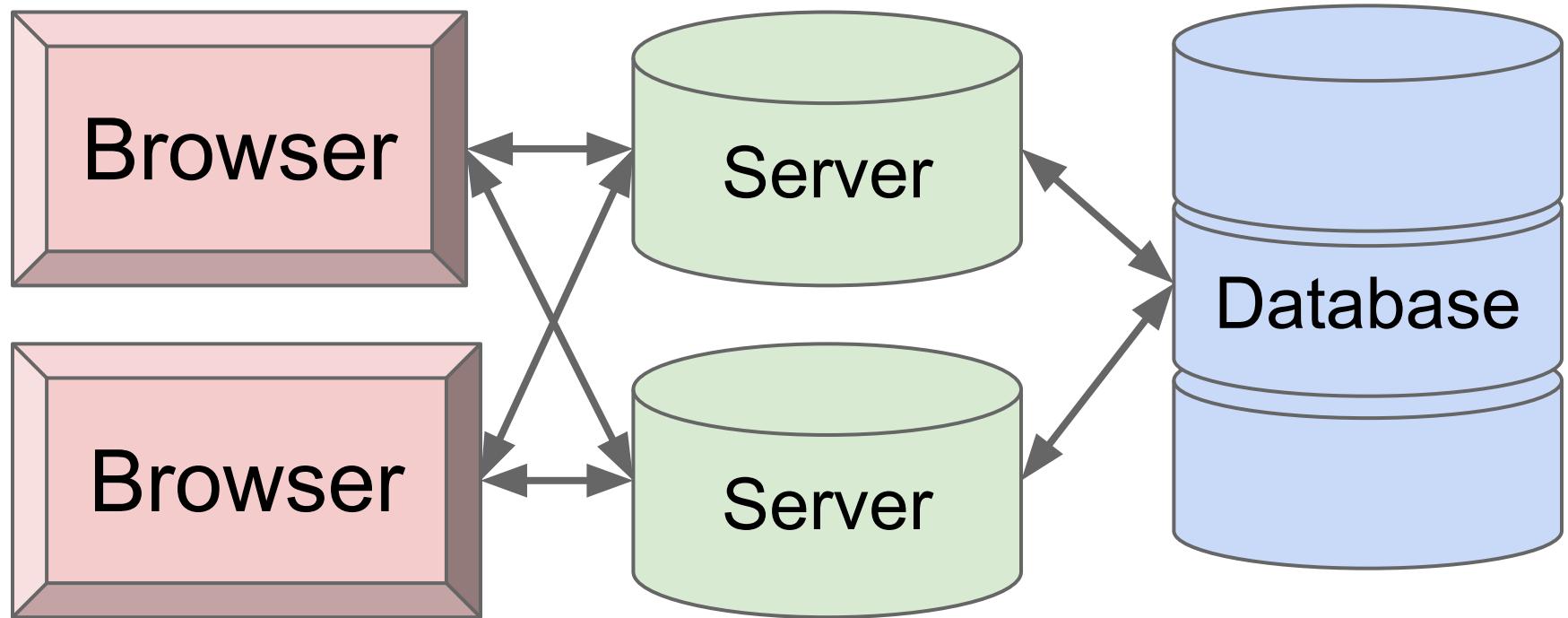
5.6 What's wrong with Web Apps?

- Javascript!

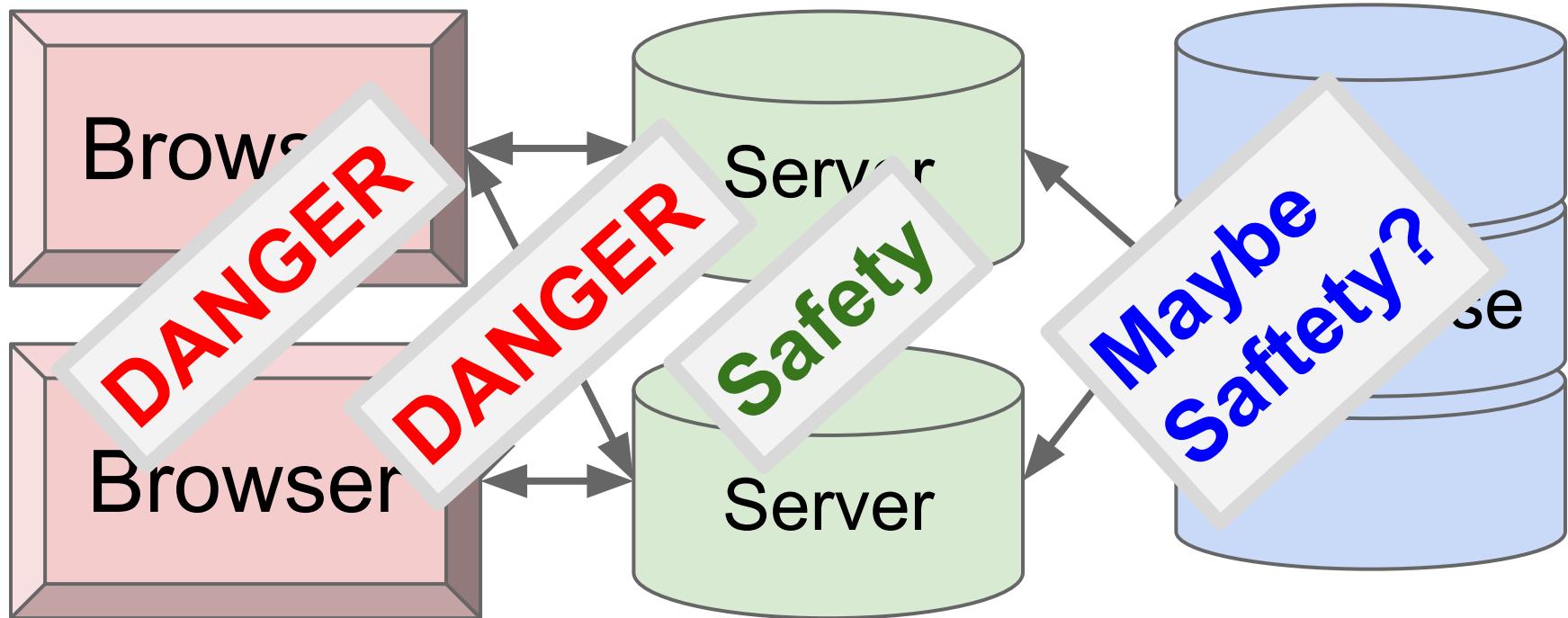
```
javascript> ["10", "10", "10", "10"].map(parseInt)  
[10, NaN, 2, 3] // WTF
```

- **Yes** this is well defined/documentated
- **No** that does not excuse its stupidity

5.7 What is a web application?



5.7 What is a web application?



5.8 Scala.js lets you...

- Write the web application in one language
 - That's not Javascript
- Swap String-typing for Strong-typing
 - In the Browser just as on the Server
 - And in between!

5.9 Scala.js: Not Javascript!

- **Scala.js -> Scala(is not)Javascript**

```
javascript> ["10", "10", "10", "10"].map(parseInt)  
[10, NaN, 2, 3] // WTF
```

```
scala.js> List("10", "10", "10", "10").map(parseInt)  
List(10, 10, 10, 10) // Yay!
```

5.10 Scala.js: Type Safety!

```
javascript> document.getElementById("Foo")
```

5.10 Scala.js: Type Safety!

```
javascript> document.getElementById("Foo")  
undefined is not a function // Gee, thanks
```

5.10 Scala.js: Type Safety!

```
javascript> document.getElementById("Foo")
```

```
undefined is not a function // Gee, thanks
```

```
scala.js> document.getElementById("Foo")
```

```
value getElementById is not a member of Document
```

Compilation failed

5.10 Scala.js: Type Safety!

```
javascript> document.getElementById("Foo")
```

```
undefined is not a function // Gee, thanks
```

```
scala.js> document.getElementById("Foo")
```

```
value getElementById is not a member of Document
```

Compilation failed

5.11 Scala.js: Reduce boilerplate

```
// Javascript
$j.ajax("/api/list", {
  data: inputBox.value,
  onComplete: function(res){ ... }
})
```

5.11 Scala.js: Reduce boilerplate

```
// Coffeescript  
$j.ajax "/api/list",  
  data: inputBox.value  
  onComplete: (res) => ...
```

5.11 Scala.js: Reduce boilerplate

```
// Coffeescript  
  
$j.ajax "/api/list",  
  data: inputBox.value  
  onComplete:(res) => ...
```

```
// Scala.js  
  
val res = Ajax[Api].list(inputBox.value).call()
```

5.12 Scala.js: Type Everything!

```
val res: Future[Seq[String]] =  
  Ajax[Api].list(inputBox.value).call()
```

5.12 Scala.js: Type Everything!

```
val res: Future[Seq[String]] =  
  Ajax[Api].lsit(inputBox.value).call()
```

value lsit is not a member of Api

Compilation failed

5.13 Scala.js: Type Everything!

```
val res: Future[Seq[String]] =  
  Ajax[Api].list(inputBox.value, "arg").call()
```

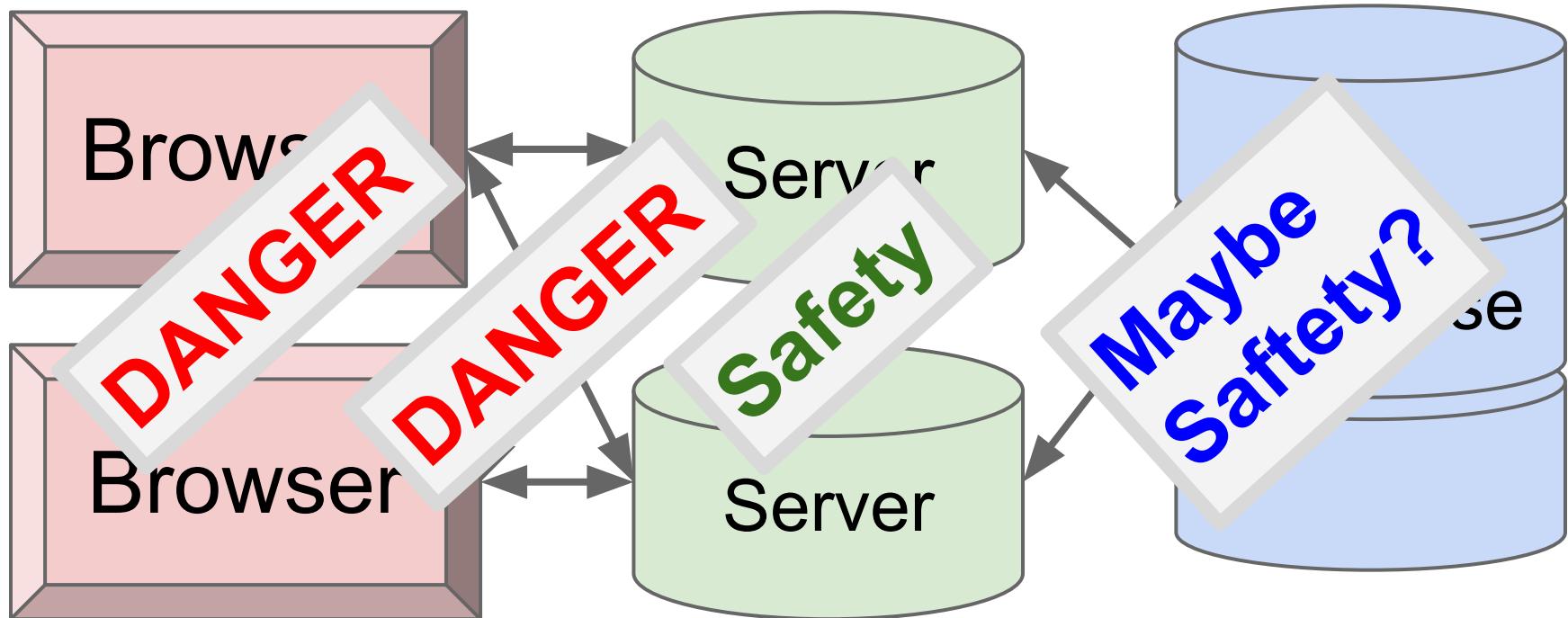
too many arguments for method list(value: S...)

Compilation failed

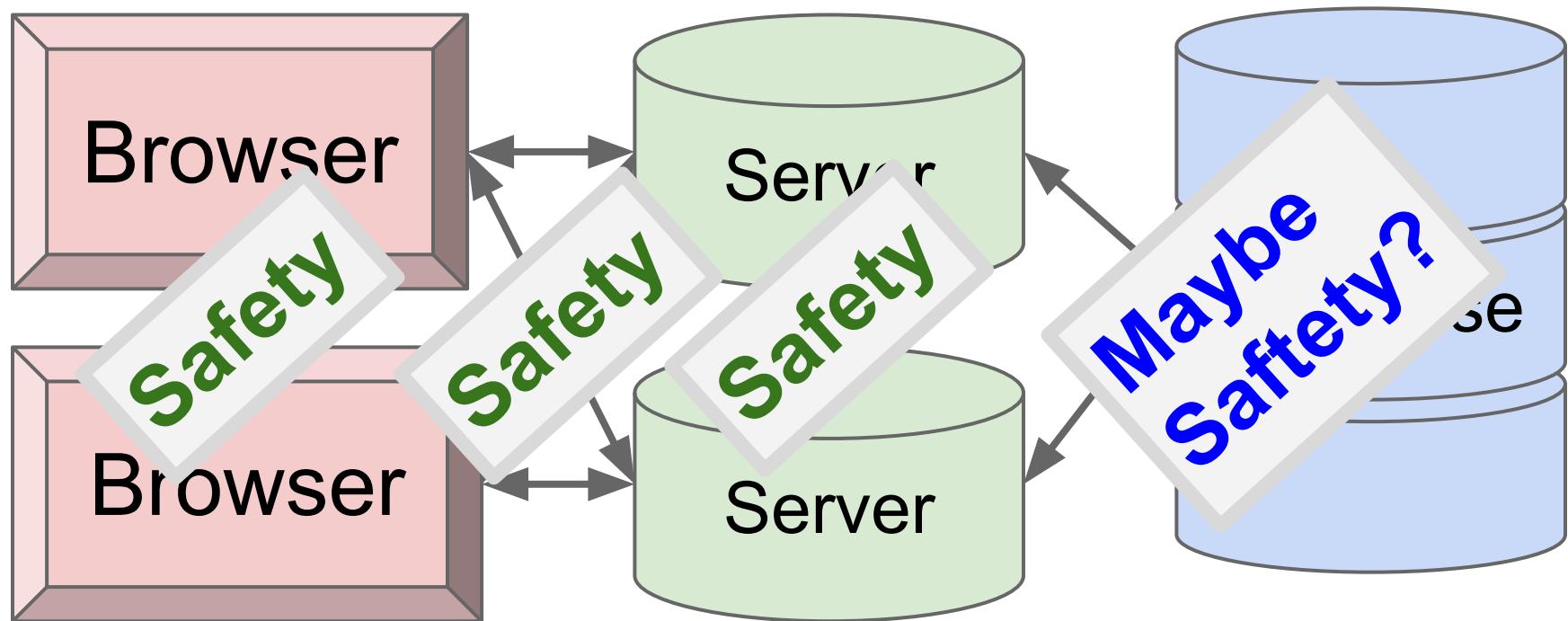
5.13 Scala.js: Type Everything!

```
val res: Seq[String] =  
    Ajax[Api].list(inputBox.value).call()  
type mismatch; found: Future[Seq[String]] ...  
Compilation failed
```

5.14 What is a web application?



5.14 What is a web application?



5.15 Scala.js gives you...

*“thanks to all ScalaJS contributors, this is really
a great system to develop with. **Lowers heart
rate and reduces adrenaline compared to
the usual JSfrontend development”***

- Otto Chrons

6 “I” am a...

- Scala dev, who works on web apps
- *Scala dev, who's never touched a web app*
- Compiler writer, who likes doing fancy optimizations

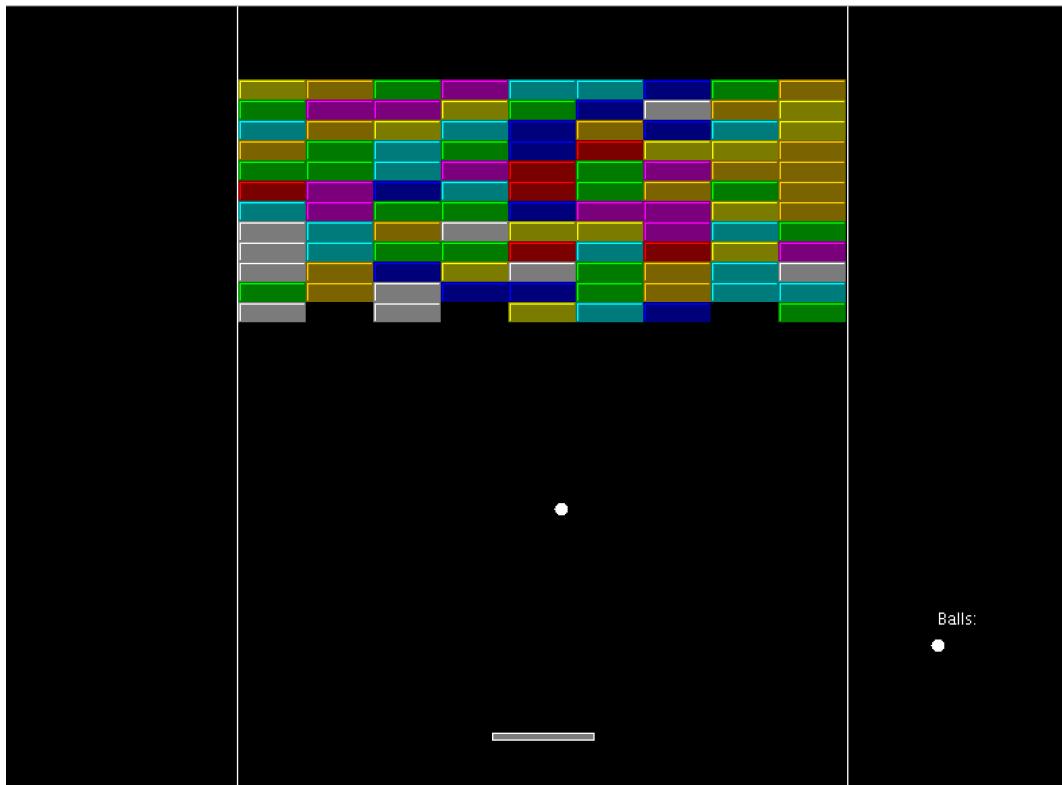
6.1 I am a Scala developer

- I have never touched a Web Application
 - I live in the terminal
 - I am a distributed-systems master
 - Headless Ubuntu is my OS of choice
- *Why should I care about Scala.js?*
 - Or: *What's wrong with Scala-JVM*

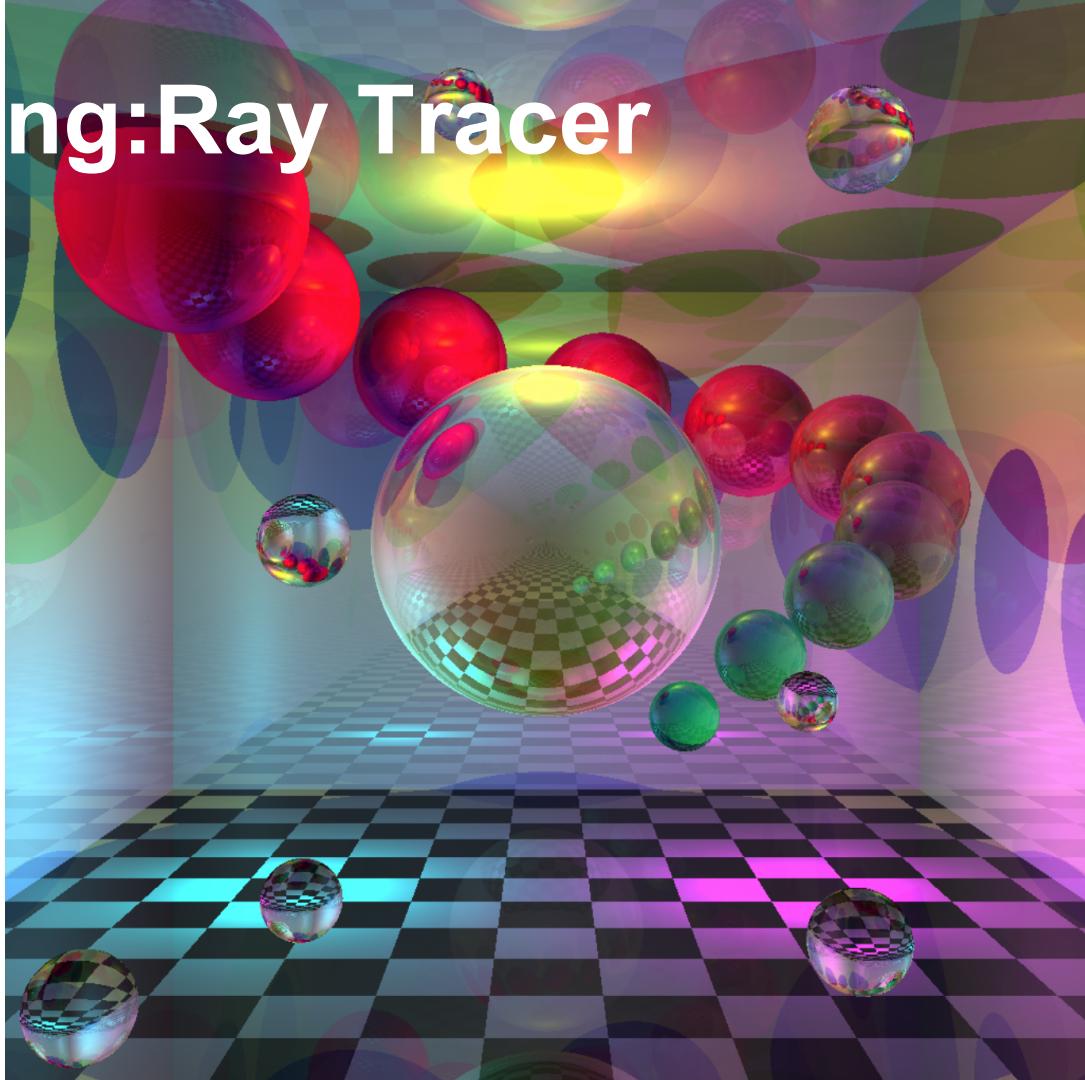
6.2 Case Study: I made a Thing

- Let's imagine I am a developer and I wrote some code
- I want to send it to someone to see it run!
- How do I do that?

6.3 Possible Thing: Game



6.4 Possible Thing:Ray Tracer



6.5 How do I let people run it?



Hey guys, anyone want to try my game? Download this
<https://dl.dropboxusercontent.com/u/7997532/Game.jar> and install Java and
run from terminal using java .jar Game.jar



Add photo



Location disabled

1



Tweet

6.6 Nobody's going to run it

- To the rest of the world...
 - Java is an island next to Sumatra
 - A terminal is where the bus driver changes shift
 - Jars are where you put cookies
- Where's the game???
- Only techies will know how to run it

6.7 End Result?

- You'll stop making fun/pretty things
- You will take a job at a big company
- You will live in the command line
- You'll forget the joy of programming

6.8 Scala.js lets you...



Hey guys, anyone want to try my game? Go to lihaoyi.github.io/scala-js-games/



Add photo



Location disabled

74



Tweet



Hey guys, anyone want to try my game? Download this
<https://dl.dropboxusercontent.com/u/7997532/Game.jar> and install Java and run from terminal using java .jar Game.jar



Add photo



Location disabled

1



Tweet



Hey guys, anyone want to try my game? Go to lihaoyi.github.io/scala-js-games/



Add photo



Location disabled

74



Tweet

6.9 Scala.js sets your Scala free

- This has always been a dilemma:
 - Making pretty things that other people can run/appreciate is cool
 - Making things using Scala is cool
 - Can only pick one
- Not anymore!
 - <http://www.scala-js-fiddle.com/gist/9443f8e0ecc68d1058ad/RayTracer.scala>

7 “I” am a...

- Scala dev, who works on web apps
- Scala dev, who's never touched a web app
- *Compiler writer, who likes doing fancy optimizations*

8 I am a Compiler Writer

- I like doing fancy optimizations
 - My first true love is transforming trees
 - Type-checking is enough for me, why run it?
 - Dead code is my sworn enemy
- *Why should I care about Scala.js?*

8.2 Scala.js is Fun to Compile

- v.s. Scala-JVM
 - More static
 - Easier to optimize
- v.s. Other Compile-2-JS languages:
 - *Much* more static
 - *Much* easier to optimize

8.3 Scala.js Static Discipline

- No separate-compilation/open-packages
 - Whole-program optimization
 - Must be explicit to compile multiple whole-programs
- No reflection
 - Private is really-truly private
- No stacktrace-introspection/sun.misc.Unsafe

8.3 Nope

```
getClass.getMethods()(0).invoke(null) // Nope
```

```
Class.forName("com.lihaoyi.MyClass").newInstance() // Nope
```

```
sun.misc.Unsafe.getUnsafe // Nope
```

8.4 No Open-Packages/Reflection

- Need to explicitly mark entry-points
- Everything else *will* be optimized/eliminated
- Classes, methods, variables, lambdas, ...

```
@JSExport  
object Main{  
    @JSExport  
    def main() = {  
        ...  
    }  
}
```

8.4 Scala-JVM: Slow for-loops

```
def count(): Int = {                                var i = IntRef.create(0);  
    var i = 0  
    for(j <- 0 until 10) i += j  
    i // 45  
}  
  
RichInt.until(intWrapper(0), 10)  
    .foreach(new $count$1(i));  
return i.elem  
  
class $count$1 extends AbstractFun1{  
    def <init>(i$1: IntRef) = {  
        this.i$1 = i$1; super.<init>();  
    }  
    def apply(j: Int) =  
        i$1.elem = i$1.elem.+ (j);  
}
```

8.4 Scala-JVM: Slow for-loops

```
def count(): Int = {  
    var i = 0  
    for(j <- 0 until 10) i += j  
    i / 45  
}
```

What if **foreach** or other helpers change?

What if someone calls them using reflection?

```
var i = IntRef.create(0);  
RichInt.until(intWrapper(0), 10)  
    .foreach(new $count$1(i));  
return i.elem  
  
class $count$1 extends AbstractFun1{  
    def <init>(i$1: IntRef) = {  
        this.i$1 = i$1; super.<init>();  
    }  
    def apply(j: Int) =  
        i$1.elem = i$1.elem.+ (j);  
}
```

8.7 Why is Scala-JVM so fat/slow?

- Nothing can be Inlined/Optimized
- Nothing can be Eliminated
- Scala.js shares none of these problems

8.5 Scala.js: Fast-loops since 2014

```
def count(): Int = {  
    var i = 0  
    for(j <- 0 until 10) i += j  
    i / 45  
}  
  
var elem$1 = 0;  
var i = 0;  
var count = 0;  
while ((i !== 10)) {  
    var v1 = i;  
    elem$1 = ((elem$1 + v1) | 0);  
    count = ((1 + count) | 0);  
    i = ((1 + i) | 0)  
};  
return elem$1
```

8.6 Compiler Output Numbers

- Scala-JVM Benchmarks:
 - ~5x slower than hand-written Java
 - Can reach ~1x if written in Java-style
 - ~7mb Hello World

8.6 Compiler Output Numbers

- Scala-JVM Benchmarks:
 - ~5x slower than hand-written Java
 - Can reach ~1x if written in Java-style
 - ~7mb Hello World
- Scala.js Benchmarks:
 - ~1x as fast as hand-written Javascript
 - No need to compromise style!
 - ~100kb Hello World

8.4 Optimization

```
var i = IntRef.create(0);
RichInt.until(intWrapper(0), 10)
    .foreach(new $count$1(i));
return i.elem

class $count$1 extends AbstractFun1{
    def <init>(i$1: IntRef) = {
        this.i$1 = i$1; super.<init>();
    }
    def apply(j: Int) =
        i$1.elem = i$1.elem.+(j);
}
```

```
var elem$1 = 0;
var i = 0;
var count = 0;
while ((i !== 10)) {
    var v1 = i;
    elem$1 = ((elem$1 + v1) | 0);
    count = ((1 + count) | 0);
    i = ((1 + i) | 0)
};
return elem$1
```

8.8 Other languages have it harder

```
# Opal: Ruby -> Javascript  
  
def count  
  i = 0  
  (0 ... 10).each{|x| i += x}  
  i  
end
```

- 100x slower than raw JS!
- Can be optimized, but dynamic ruby semantics will be broken
- Python, e.t.c.

```
var $a, $b, TMP_1, self = this, i = nil;  
  
i = 0;  
(  
  $a = ($b = ($range(0, 10, true))).$each,  
  $a.$$p = (  
    TMP_1 = function(x){  
      var self = TMP_1.$$s || this;  
      if (x == null) x = nil;  
      return i = i['$+'](x)  
    },  
    TMP_1.$$s = self, TMP_1  
  ), $a  
).call($b);  
return i;
```

8.9 Other languages have it harder

- “It depends what you are looking for. The closer you get to 100% support for Python, the more weight you "pay".”
- ClojureScript: gave up Vars, eval
- Dart: Reflection makes the output huge!
- Dynamic features are *expensive!*

8.10 Scala.js gives you...

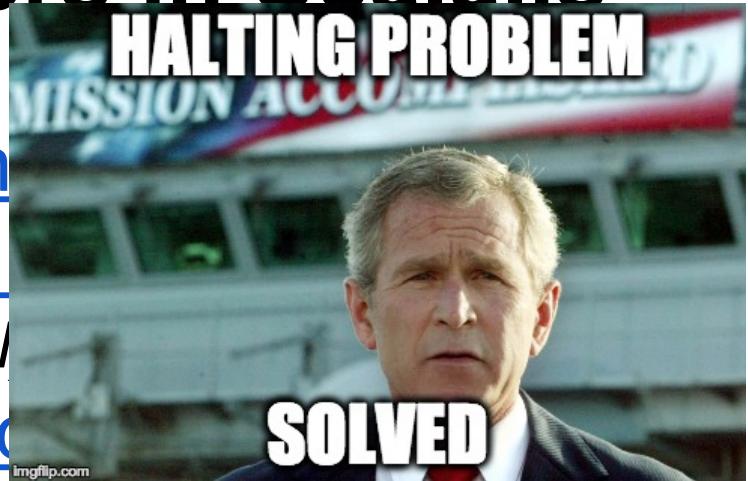
- A well-specified language with specified semantics
 - If you think the Scala spec isn't good, look at the Python or Ruby specs! [Oh wait...](#)
- Static-analyzable semantics
 - Far more so than Scala-JVM
- Tons of opportunity for interesting work!
 - We had Typed Trees before it was cool

8.11 Fun with Compilers in Scala.js

- Guaranteeing-termination via turing-completeness-removal for Scala applications
 - ~30LOC, *mostly regexes (lol)*
- JRebel-style live-editing for Scala.js
 - ~200LOC, *also mostly regexes*
- Try doing *that* on Scala-JVM!

8.11 Fun with Compilers in Scala.is

HALTING PROBLEM



- Guaranteeing-termination completeness-removal for
 - ~30LOC, *mostly regexes* (*lol*)
- JRebel-style live-editing for
 - ~200LOC, *also mostly regexes*
- Try doing *that* on Scala-JVM!

9 Many things to Many people

- To a Web Engineer...
 - Scala.js is a breath of safety in a sea of danger
 - Do your work without Adrenaline!
- To a Scala Programmer...
 - Scala.js sets your Scala free
 - No longer is your work trapped in the command line!
- To a Compiler Writer...
 - Scala.js is an easily approachable compilation target
 - .. with solid semantics and lots of room for fun!

10 What's Scala.js to you?

- Questions?

