

# Intro to Scala.js

Singapore-Scala, 28 Mar 2017

Li Haoyi

haoyi.sg@gmail.com

Bright Technology Services  
[www.bright.sg](http://www.bright.sg)



# About me

Previously software engineer at Dropbox



Currently at Bright technology services ([www.bright.sg](http://www.bright.sg))

- data-science/software consulting

Early contributor to Scala.js, author of [Ammonite REPL](#), [Scalatags](#), [FastParse](#), ...

@li\_haoyi on Twitter, @lihaoyi on Github



Bright

technology services

# Agenda

What is this Scala.js thing?

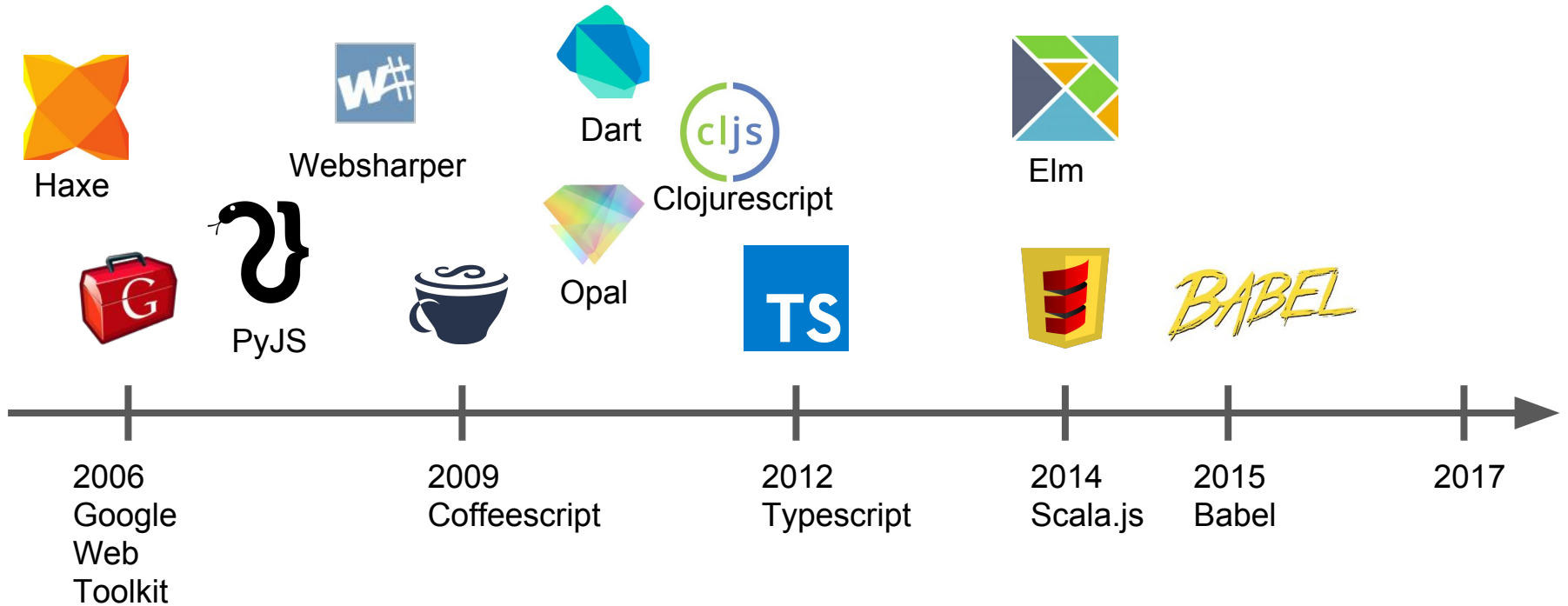
Why should I care?

How does it work?



# Compiling to Javascript

# Timeline of Compile-to-Javascript Languages



Why Compile to Javascript?

# Why Compile to Javascript?

Share expertise between Client and Server

Share code between Client and Server

TECHNOLOGY LAB —

**How Google Inbox shares 70% of its code across Android, iOS, and the Web**

Google's open source tools allow it to use Android code on iOS and the Web.

RON AMADEO - 11/21/2014, 1:10 AM

Choose what language to build your website in



Scala.js



[www.scala-js.org](http://www.scala-js.org)



# Scala.js: What

```
def main() = {  
  var x = 0  
  while(x < 999){  
    x = x + "2".toInt  
  }  
  println(x)  
}
```

# Scala.js: What

```
ScalaJS.c.LExample$.prototype.main__V = (function() {  
  var x = 0;  
  while ((x < 999)) {  
    x = ((x + new ScalaJS.c.sci_StringOps().init__T(  
      ScalaJS.m.s_Predef().augmentString__T__T("2")  
    ).toInt__I()) | 0)  
  };  
  ScalaJS.m.s_Predef().println__O__V(x)  
});
```

# Scala.js: What

```
be.prototype.main=function(){
  for(var a=0;999>a;)
    a=a+(new de).g(S(L(),"2")).ne()|0;
  ee(); L();
  var b=F(fe); ge();
  a=(new he).g(w(a)); b=bc(0,J(q(b,[a])));
  ie(bc(L(),J(q(F(fe),[je(ke(ge()).Vg),b]))))
}
```

# Scala.js: Examples

Ray Tracing [scalafiddle.io/sf/4beVrVc/1](https://scalafiddle.io/sf/4beVrVc/1)

Online Games [www.lihaoyi.com/roll](http://www.lihaoyi.com/roll)

Web Apps [demo.fluentcode.com](http://demo.fluentcode.com)

# Common patterns of using Scala.js

Client-side Scala.js



Client-Server Scala/Scala.js



# Client-side Scala.js

# Live Demo: Client-side Scala.js

[github.com/lihaoyi/workbench-example-app](https://github.com/lihaoyi/workbench-example-app)

# Type-checked by default

```
var paragraph = document.body
console.log(paragraph.children.length)
```

```
val paragraph = document.body
console.log(paragraph.children.length)
```

Cannot resolve symbol children

ScalaJSExample.scala:12: value  
children is not a member of  
org.scalajs.dom.raw.Element

```
console.log(paragraph.children.length)
```

^

Compilation failed

✘ ▼ Uncaught TypeError: Cannot read property 'length' of undefined  
(anonymous function) @ [index-fastopt.html:22](#)

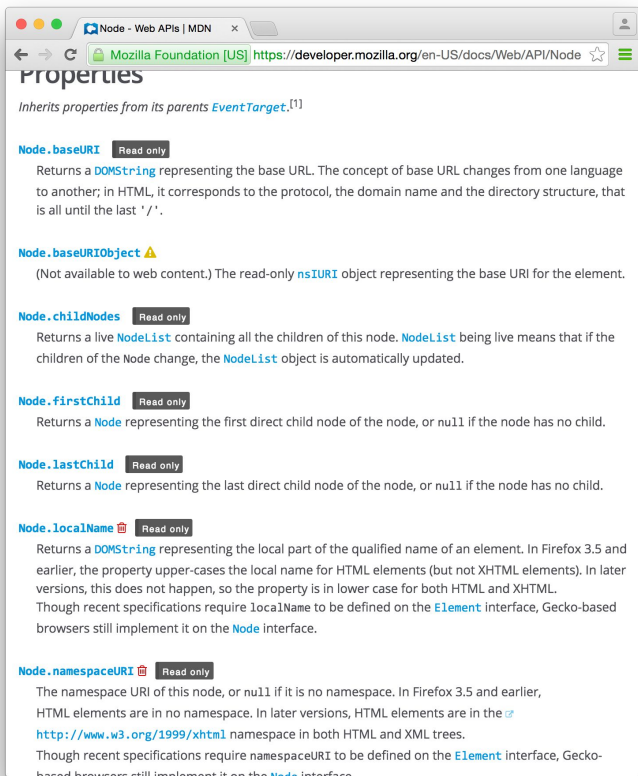


# Fewer Warts

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

```
scalajs> Seq("10", "10", "10", "10").map(Integer.parseInt)  
List(10, 10, 10, 10)
```

# Great editor support



Node - Web APIs | MDN

MoZilla Foundation [US] <https://developer.mozilla.org/en-US/docs/Web/API/Node>

## Properties

Inherits properties from its parents [EventTarget](#).<sup>[1]</sup>

**Node.baseURI** Read only

Returns a [DOMString](#) representing the base URL. The concept of base URL changes from one language to another; in HTML, it corresponds to the protocol, the domain name and the directory structure, that is all until the last `'/'`.

**Node.baseURIObject** ⚠

(Not available to web content.) The read-only [nsIURI](#) object representing the base URI for the element.

**Node.childNodes** Read only

Returns a live [NodeList](#) containing all the children of this node. [NodeList](#) being live means that if the children of the Node change, the [NodeList](#) object is automatically updated.

**Node.firstChild** Read only

Returns a [Node](#) representing the first direct child node of the node, or `null` if the node has no child.

**Node.lastChild** Read only

Returns a [Node](#) representing the last direct child node of the node, or `null` if the node has no child.

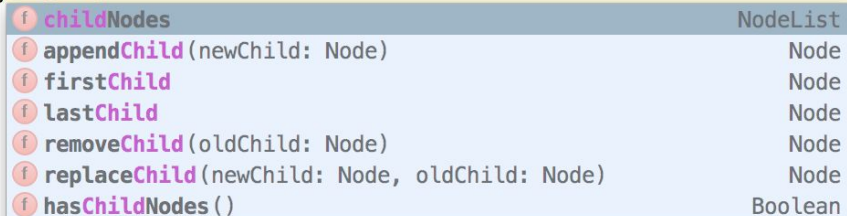
**Node.localName** Read only

Returns a [DOMString](#) representing the local part of the qualified name of an element. In Firefox 3.5 and earlier, the property upper-cases the local name for HTML elements (but not XHTML elements). In later versions, this does not happen, so the property is in lower case for both HTML and XHTML. Though recent specifications require `localName` to be defined on the [Element](#) interface, Gecko-based browsers still implement it on the [Node](#) interface.

**Node.namespaceURI** Read only

The namespace URI of this node, or `null` if it is no namespace. In Firefox 3.5 and earlier, HTML elements are in no namespace. In later versions, HTML elements are in the <http://www.w3.org/1999/xhtml> namespace in both HTML and XML trees. Though recent specifications require `namespaceURI` to be defined on the [Element](#) interface, Gecko-based browsers still implement it on the [Node](#) interface.

```
def doThing(target: dom.Node) = {  
  target.child  
}
```



<code>childNodes</code>	<code>NodeList</code>
<code>appendChild</code>	<code>Node</code>
<code>firstChild</code>	<code>Node</code>
<code>lastChild</code>	<code>Node</code>
<code>removeChild</code>	<code>Node</code>
<code>replaceChild</code>	<code>Node</code>
<code>hasChildNodes</code>	<code>Boolean</code>

Press `^`. to choose the selected (or first) suggestion and insert a dot afterwards [>>](#) [π](#)

## Documentation for childNodes

[←](#) [→](#) [↑](#) [📄](#) SBT: org.scala-js:scalajs-dom\_sjs0.6\_2.11... [⚙](#)

[org.scala.js.dom.raw.Node](http://org.scala.js.dom.raw.Node)

def `childNodes`: [NodeList](#)


Returns a live [NodeList](#) containing all the children of this node. [NodeList](#) being live means that if the children of the Node change, the [NodeList](#) object is automatically updated. [MDN](#)

# Library Ecosystem







Use any JS library

- `val xhr = new XMLHttpRequest()`
- React
- D3
- ...

Along with lots of Scala libraries...

<a href="#">scalajs-java-time</a>	Port of the <a href="#">java.time API</a> of JDK8 for Scala.js	
<a href="#">scalajs-jsjoda-as-java-time</a>	Implementation of JDK8's <a href="#">java.time API</a> in Scala.js by wrapping <a href="#">js-joda</a> classes	
<a href="#">scala-java-time</a>	Platform-independent implementation of <a href="#">java.time</a>	
<a href="#">scala-java-locales</a>	Implementation of JDK8's <a href="#">java.util.Locale API</a> and parts of <a href="#">java.text API</a>	
<a href="#">scalajs-java-logging</a>	Port of the <a href="#">java.util.logging API</a> of JDK 8 for Scala.js	

## Functional programming

<a href="#">Scalaz</a>	Library for functional programming.	
<a href="#">Each</a>	A macro library that converts native imperative syntax to scalaz's monadic expressions.	
<a href="#">Shapeless</a>	Generic programming for Scala.	
<a href="#">Cats</a>	Lightweight, modular, and extensible library for functional programming.	
<a href="#">Monocle</a>	Optics library strongly inspired by Haskell <a href="#">Lens</a> .	
<a href="#">Quicklens</a>	Modify deeply nested fields in case classes.	

## Web libraries/frameworks

<a href="#">Udash</a>	A Scala(.js) framework for building beautiful and maintainable Web Applications	
<a href="#">Binding.scala/dom</a>	Reactive web framework for Scala.js	
<a href="#">ScalaTags</a>	HTML templating library/DSL that works on both Scala/JVM and Scala.js	
<a href="#">statictags</a>	Write HTML in both Scala JVM & JS. Extend tags and attributes with ease!	

# Client-side Scala.js: Limitations

## Can use:

- Most of `java.lang.*`
- Almost all of `scala.*`
- Some of `java.util.*`
- Scala Macros: `upickle`, `scala-async`, `scalaxy`, etc
- Pure-Scala ecosystem: `shapeless`, `scalaz`, `scalatags`, `utest`

## Can't use:

- `j.l.Thread`, `j.l.Runtime`, ...
- `s.c.parallel`, `s.tools.nsc`
- `org.omg.CORBA`, `sun.misc.*`
- Reflection: `scala-pickling`, `scala-reflect`
- Java-dependent: `Scalatest`, `Scalate`

# Client-side Scala.js: Limitations

## Can use:

- JS stuff: XMLHttpRequest, Websockets. Localstorage
- HTML DOM, Canvas, WebGL
- JavaScript libraries: chipmunk.js, hand.js, react.js, jquery
- IntelliJ, Eclipse, SBT
- Chrome console, firebug

## Can't use:

- JVM stuff: Netty, akka, spray, file IO, JNI
- AWT, Swing, SWT, OpenGL
- Java ecosystem: guice, junit, apache-commons, log4j
- Yourkit, VisualVM, JProfiler

# Client-side Scala.js: using Javascript libraries

```
// Chipmunk.js definition in javascript
cp.Vect = function(x, y){
    this.x = x;
    this.y = y;
};
```

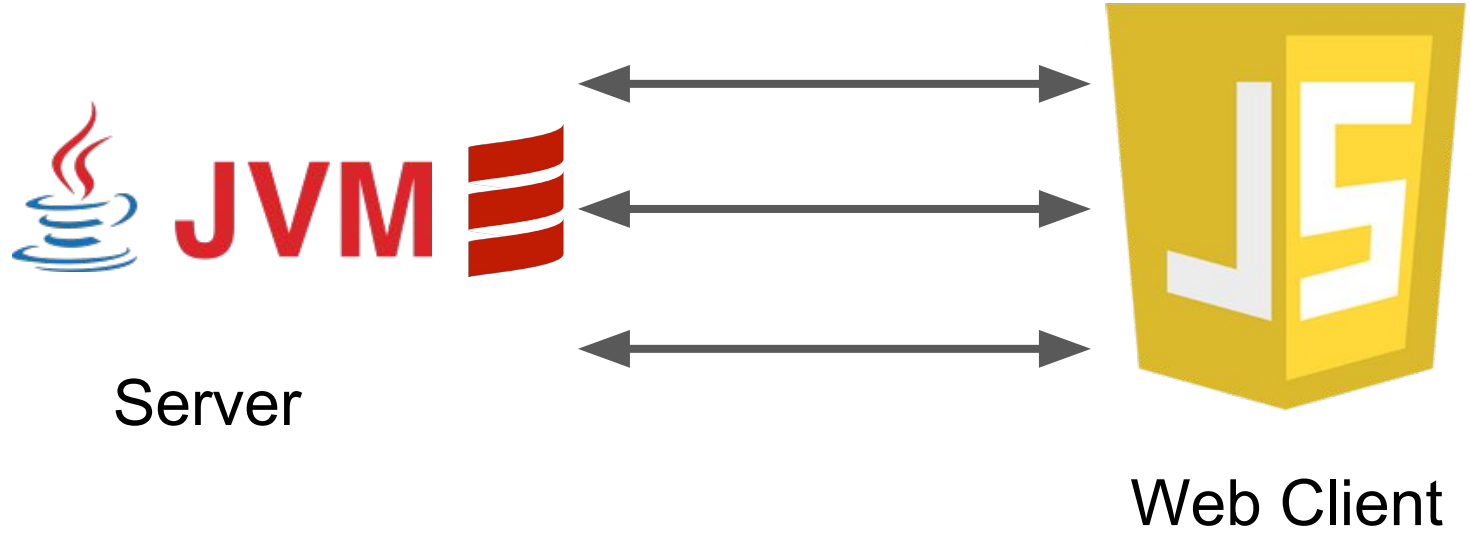
```
// using Chipmunk.js in javascript
var p = new cp.Vect(50, 100)
console.log(p.x + p.y) // 150
```

```
// Chipmunk.js definition in Scala
package cp
@JSName("cp.Vect")
class Vect(var x: Double,
           var y: Double)
           extends js.Object
```

```
// using Chipmunk.js in Scala
val p = new cp.Vect(50, 100)
println(p.x + p.y) // 150
```

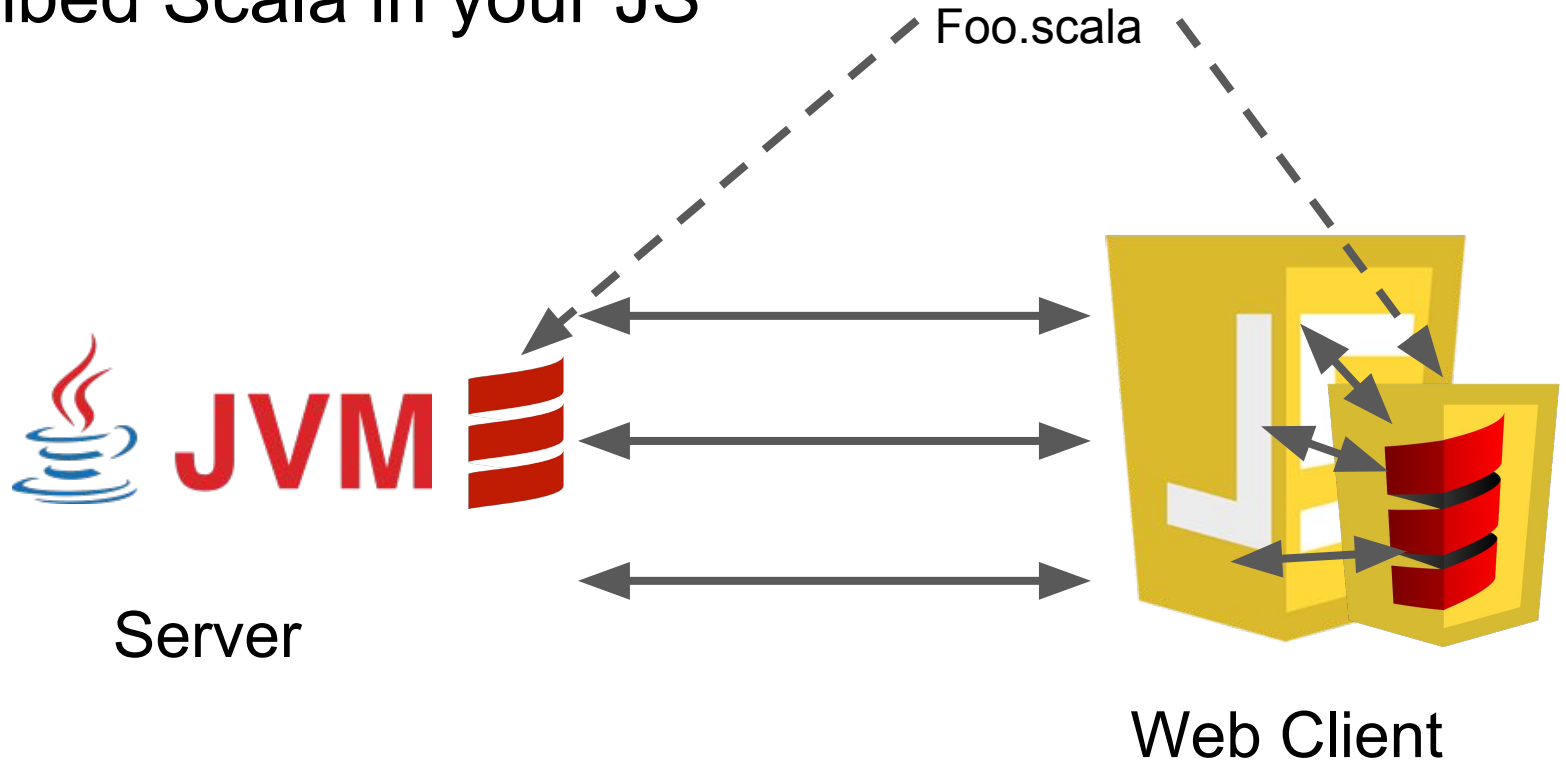
Client-Server Scala.js

# Embed Scala in your JS





# Embed Scala in your JS



# Embed Scala in your JS

```
@JSExportTopLevel("Foo")
class Foo(val x: Int) {
  @JSExport
  def square(): Int = x*x
  @JSExport("foobar")
  def add(y: Int): Int = x+y
}
```

```
var foo = new Foo(3);

console.log(foo.square());
// 9

console.log(foo.foobar(5));
// 8
```

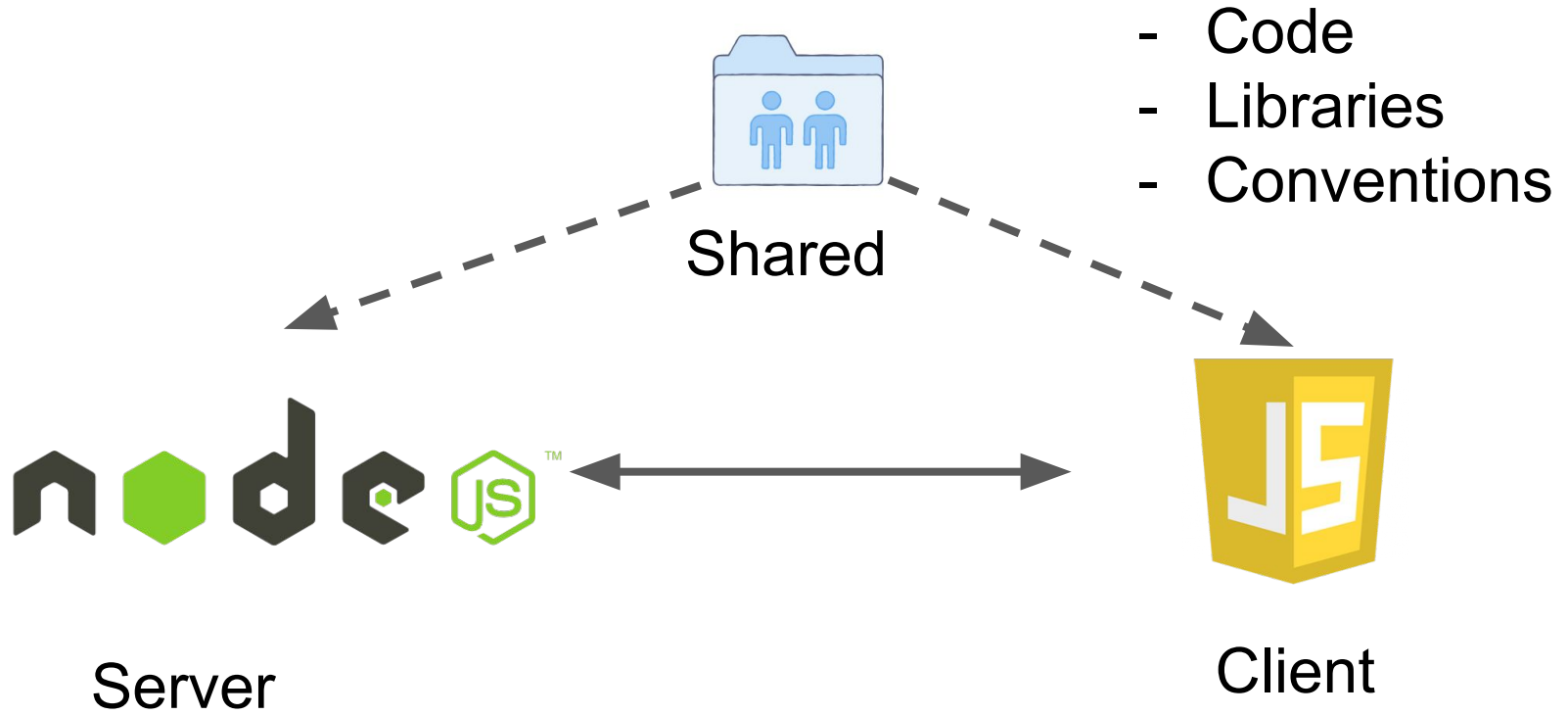
# Embed Scala in your JS

[NetLogo Web](#): agent-based simulation engine

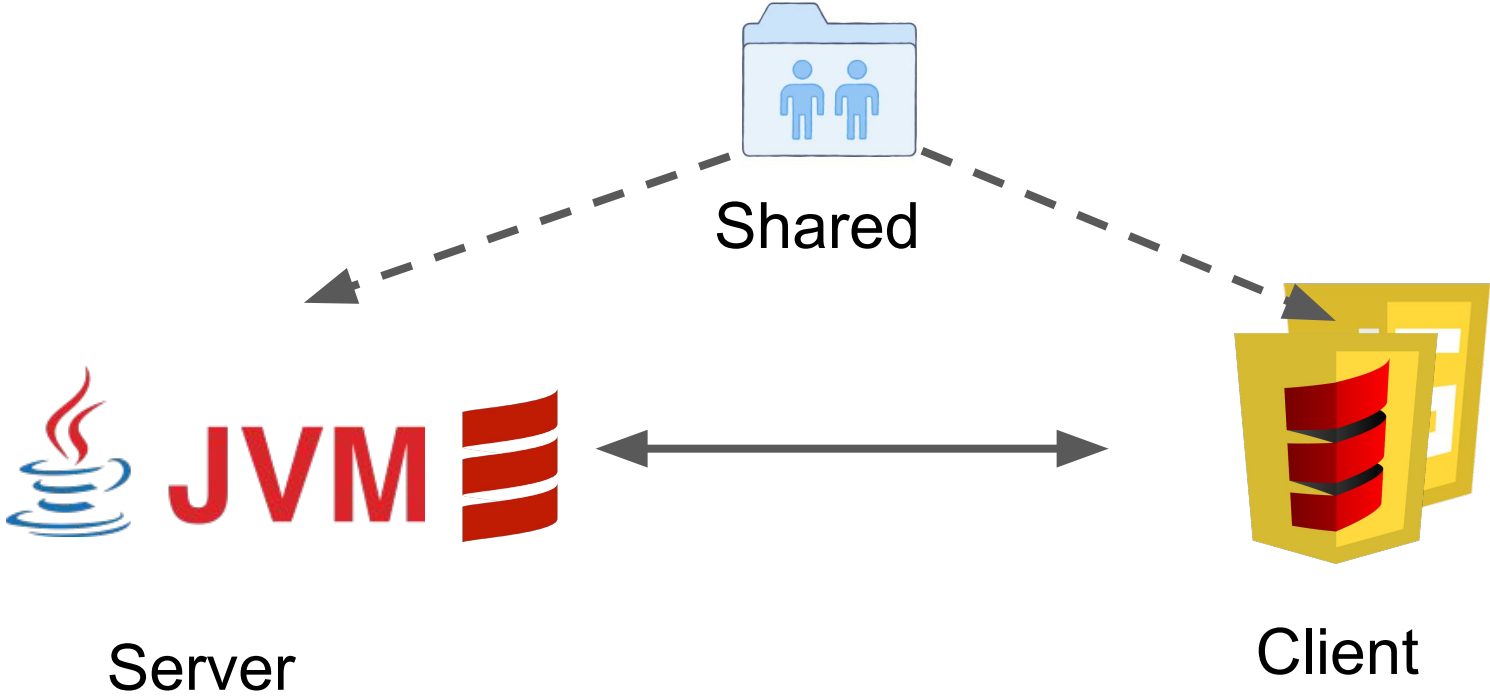
[marianogappa/ostinato](#): re-usable chess engine

[fommil.com/kerbal](#): kerbal space program calculator

# Isomorphic Javascript

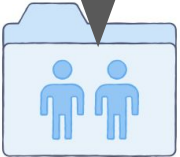


# Isomorphic Scala



Scalatags, uPickle, Scalatest

Akka-HTTP  
javax.mail  
Scopt



Shared

React  
Angular  
jQuery



Server



Client

# Isomorphic Scala: Shared Libraries

[Scalaz](#): functional programming

[Scalatest](#): test framework

[Shapeless](#): generic programming

[Akka](#): actors

[Accord](#): data validation

[Monix](#): asynchronous streaming

[Scala Async](#): async & await

[QuickLens](#): lenses for updating case classes

[Scalatags](#): HTML templating

[uPickle](#): JSON serialization

[BooPickle](#): binary serialization

[Circe](#): JSON handling

[Autowire](#): type-safe routing

[RosHTTP](#): HTTP client

[FastParse](#): parser combinators

[Shocon](#): Typesafe “HOCON” config parser

Why Scala.js?



# What do Javascript developers get from Scala.js?

Everything in ES6/7 (=>, destructuring, string-interpolation, ...)

Everything in TypeScript (types, generics, ...)

Everything in Immutable.js (immutable collections)

+

Great language, extensive standard library, functional programming, shared client-server code, access to all JS libs, fearless refactoring

# What do Scala developers get from Scala.js?

You can already write back-end servers



And compilers



You can now program web front-ends,



Browser extensions

Node.js servers: [github.com/scalajs-io/nodejs](https://github.com/scalajs-io/nodejs)



Microcontrollers ([www.espruino.com](http://www.espruino.com), [tessel.io](http://tessel.io))



Mobile apps with React-Native



# Scala.js: Performance

- **Relatively quick:** 1-2s warm turnaround
- **Acceptable size:** small apps start at ~70kb, grow to 100s of kb pre-gzip
- **Efficient Code:** ~1-2x slower than “raw” Javascript

# Intro to Scala.js

[scala-js.org](http://scala-js.org)



Bright

technology services

[haoyi.sg@gmail.com](mailto:haoyi.sg@gmail.com)

[www.bright.sg](http://www.bright.sg)

