Scala.js

Safety & Sanity in the wild west of the web

Li Haoyi, Philly ETE 8/3/2015

1.1 Who am I

Work at Dropbox on Web Infrastructure, writing Python/Coffeescript

Using Scala open-source since 2011

Contributor to Scala.js since 2013

1.2 What is Scala.js

```
val (obj, misc) = objects(i)
val t = obj.intersectionTime(ray)
if (t > Epsilon &&
    t < length - Epsilon){
  visible = false
}
```

```
var tup = self.Ve.objects[i]
if (null !== tup)
  obj = tup._1, misc = tup._2
else
  throw (new MatchError).init(tup)

var t = obj.intersectionTime(ray)
t > Example$().Epsilon &&
t < length - Example$().Epsilon &&
(visible = !1)
```

1.3 What is Scala.js?

Write Scala, Run Javascript, Make Website!

Compiler takes care of in between

100s of kb of code, ~1x as fast as “raw” JS

Supports entire Scala language, many libraries

1.4 Why Scala?

Concise code due to powerful abstractions

Extreme Type safety!

Good tooling, broad ecosystem to depend on

Live Demo

github.com/lihaoyi/workbench-example-app

1.5 Notes from the Demo

Fast turn-around time

Compile errors when you make a mistake

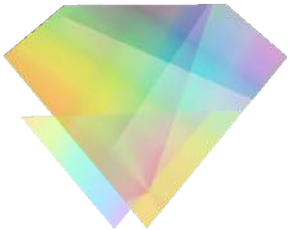
Accurate in-editor autocomplete

2.1 How does Scala.js compare to

CoffeeScript

HAXE

Opal



Dart

clojure / clojurescript

WebSharper

elm

TypeScript

opa

Emscripten: An LLVM-to-JavaScript Compiler

2.2 Everyone wants a better web

Safer

More modular, expressive, reusable code

One language across client/server

Async support

More tool-able & better tooling

Fewer warts

2.3 Safety

```
Uncaught TypeError: undefined is not a function
  o.extend.trim
  b
  d.fx.step.(anonymous function)
  o.fx.update
  o.fx.step
  F
  o.fx.custom
```

2.4 More Expressive

```
race = (winner, runners...) ->
  print winner, runners
```

```
race = function() {
  var winner = arguments[0]
  var runners =
    2 <= arguments.length ?
    slice.call(arguments, 1) : [];
  print(winner, runners);
};
```

2.5 One language for client/server

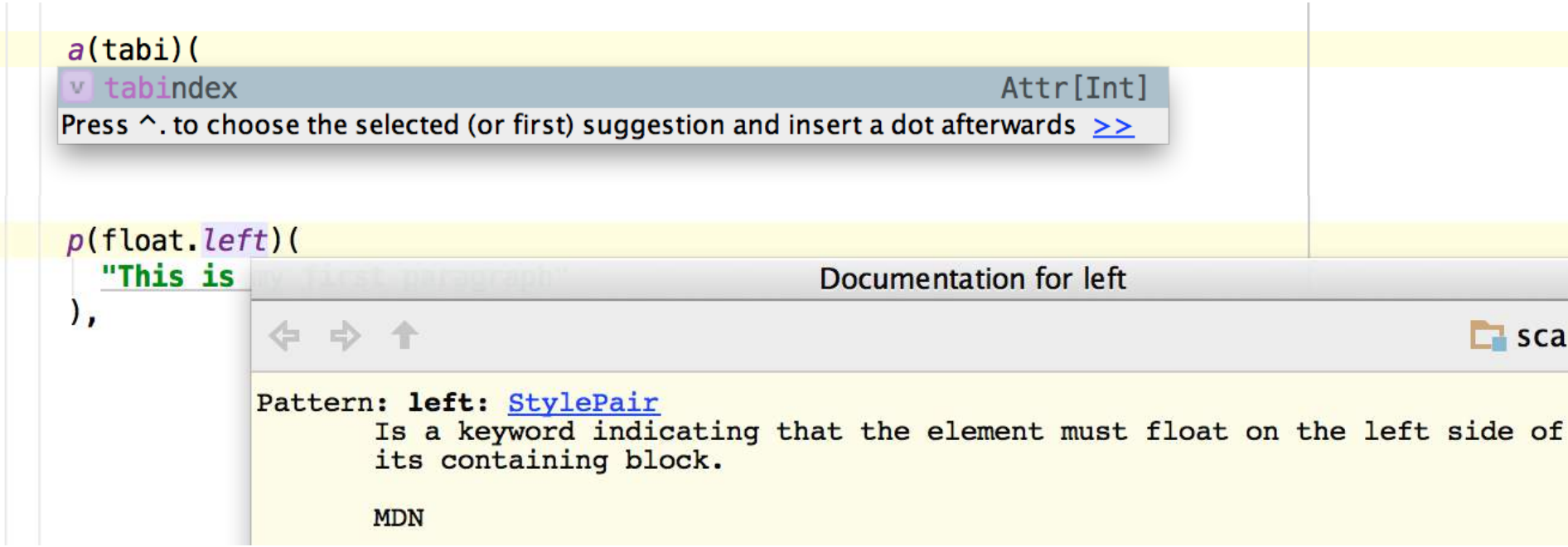
```
# This has been ported to our Python Emstring class
# Please keep them both in sync if you need to change something!
class Emstring
  @em_snippet: (s, maxchars=50, location=0.75) ->
    new Emstring(s.toString()).snippet(
      maxchars, location
    ).toString()
```

2.6 Async

```
ajaxFoo((a) =>
  bar(a, (b) =>
    baz(a, (c) =>
      b + c
    )
  )
)
```

```
async{
  var a = wait(ajaxFoo())
  wait(bar(a)) + wait(baz(a))
}
```

2.7 More Toolable/Better Tooling



2.8 Fewer Warts

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

3.1 How does Scala.js compare to

Javascript ES6/7

Coffeescript

Clojurescript

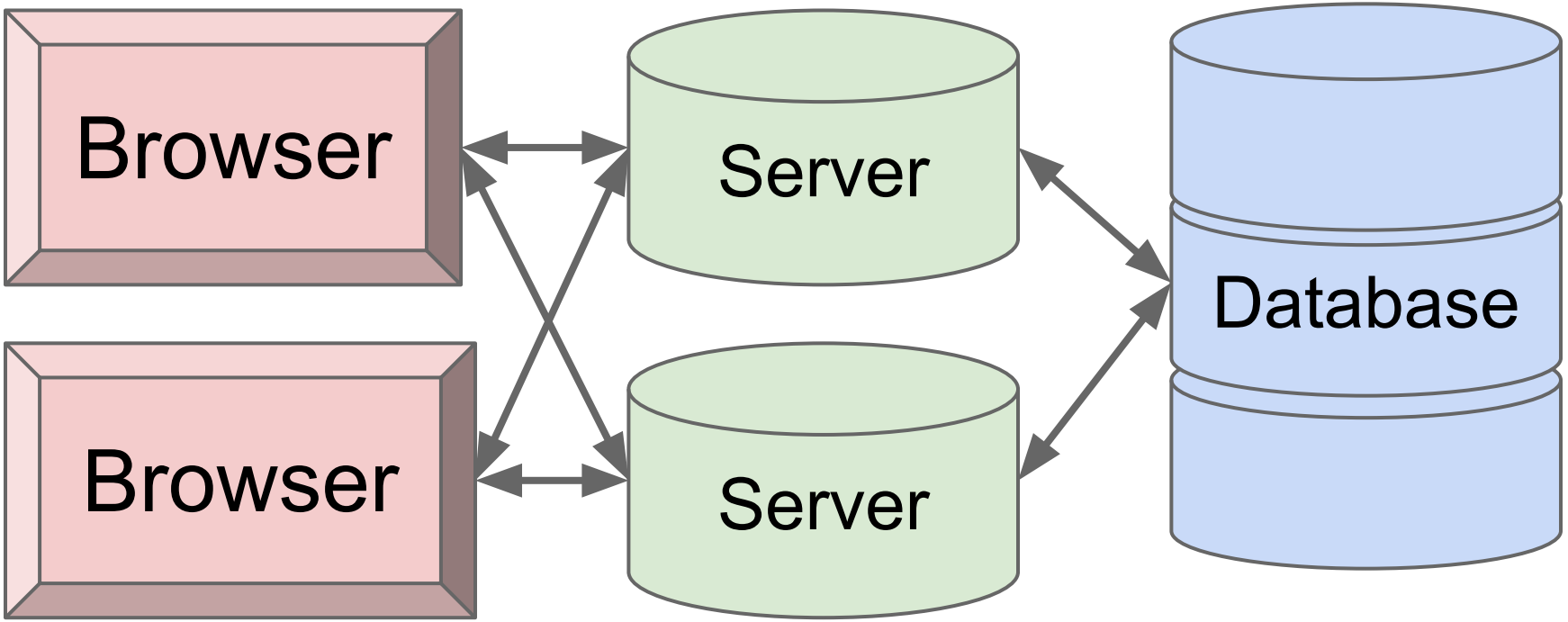
Typescript

Dart

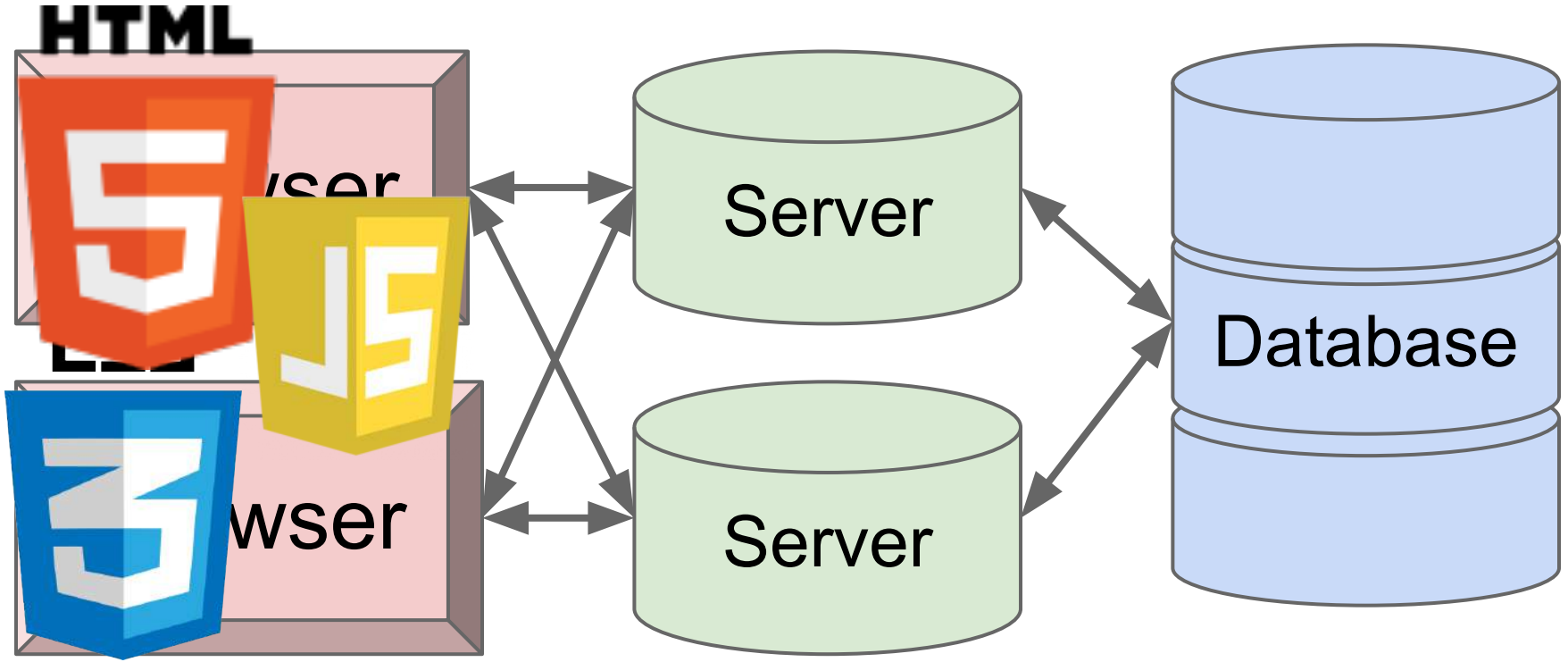
WARNING: mostly an outside perspective

	Safer	Expressive	Async	Toolable	Wart--
ES6/7	Still Un-typed	Some amount of new syntax	Generators!	Still Javascript	Still Javascript
Coffee	Still Un-typed	Ridiculously concise	Nope	Similar to Javascript	New syntaxes avoid warts
CLJS	Still Un-typed	Very powerful language	core/async	Similar to Javascript	Whole new language
TS	Types!	Some amount of new syntax	Generators …later, in 1.6	Visual Studio IntelliJ	Still Javascript
Dart	Types!	Half Javascript Half Java	async/await, new in 1.9	Mostly Eclipse IntelliJ works	Whole new language
Scala.js	Very Strong Types!	Very powerful language	async/await	Great support: Eclipse/IntelliJ	Whole new language

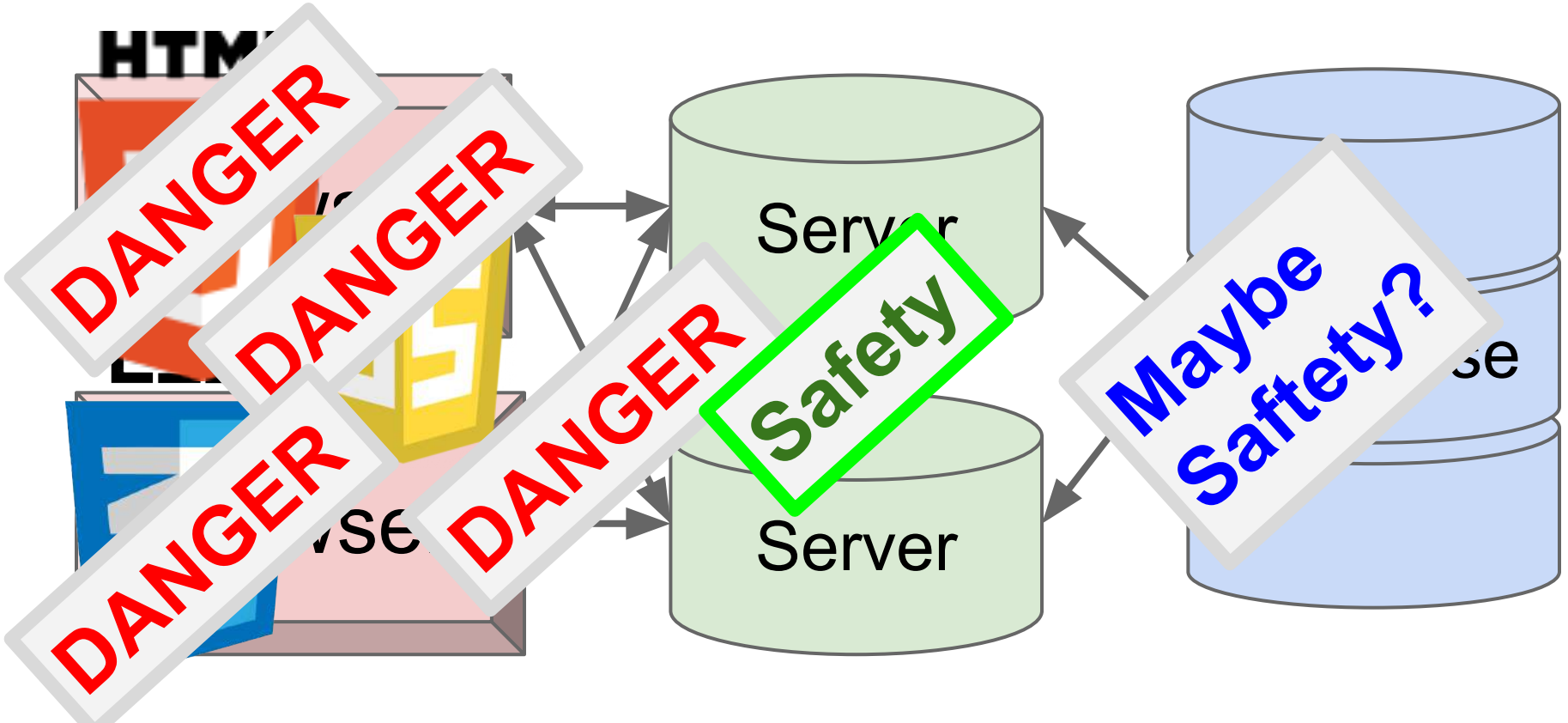
4.1 What is a web application?



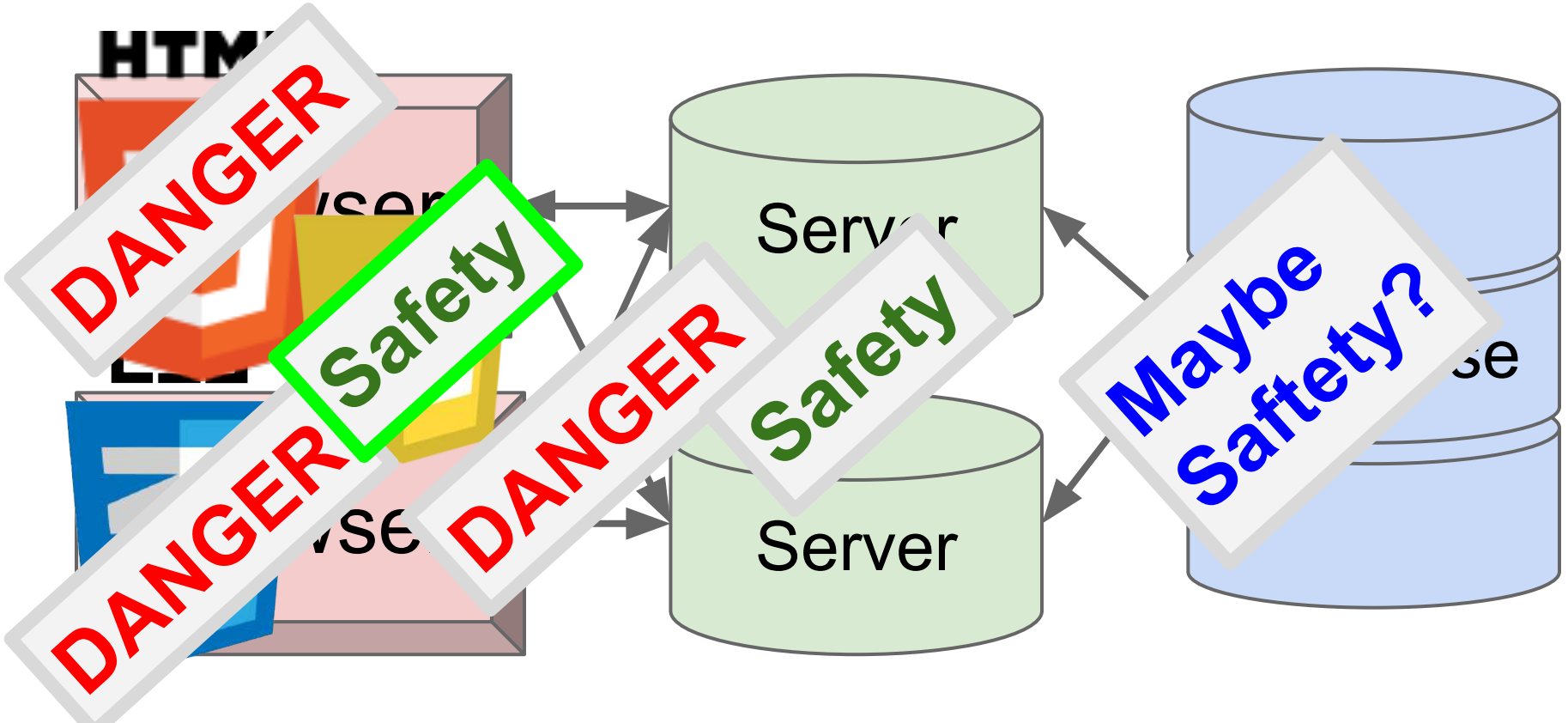
4.1 What is a web application?



4.1 What is a web application?



4.1 What is a web application?



4.2 Typed HTML!

```
div(
  float.left,
  p("I am cow"),
  p("Hear me moo")
)
```

```
<div
  style="float: left">
  <p>I am cow</p>
  <p>Hear me moo</p>
</div>
```

4.2 Typed HTML!

```
div(
  float.elft,
  p("I am cow"),
  p("Hear me moo")
)
```

```
value elft is not a
member of object float
  float.elft,
        ^

Compilation failed
```

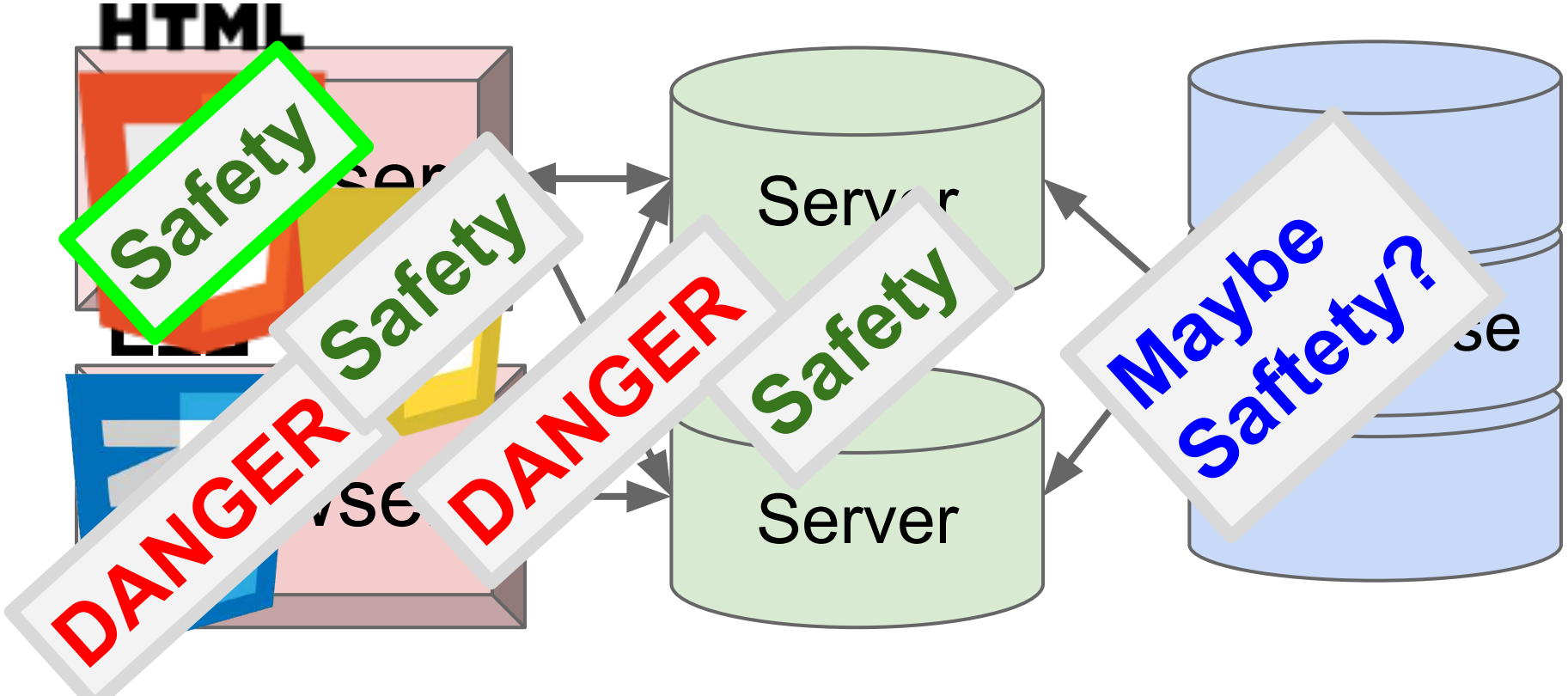
4.2 Typed HTML!

```
dvi(
  float.left,
  p("I am cow"),
  p("Hear me moo")
)
```

```
Not found: value dvi
dvi(
^

Compilation failed
```

4.3 What is a web application?



4.4 Hygienic, Typed CSS!

```
trait Simple{
  def btn = cls(
    color := "red",
    height := 125
  )
  def fade = cls.hover(
    opacity := 0.5
  )
}
```

```
.$pkg-Simple-btn{
  color: red;
  height: 125px;
}
.$pkg-Simple-fade:hover{
  opacity: 0.5;
}
```

4.4 Hygienic, Typed CSS!

```
trait Simple{
  def btn = cls(
    colro := "red",
    height := 125
  )
  def fade = cls.hover(
    opacity := 0.5
  )
}
```

```
Not found: value colro
    colro := "red"
    ^

Compilation failed
```

4.4 Hygienic, Typed CSS!

```
trait Simple{
  def btn = cls(
    color := "red",
    height := 125
  )
  def fade = cls.hovre(
    opacity := 0.5
  )
}
```

```
value hovre is not a member of
object cls
  def fade = cls.hovre(
                       ^

Compilation failed
```

4.5 Hygienic, Typed CSS!

```
val x = div(
  cls := """
    $pkg-Simple-btn
    $pkg-Simple-fade
  """,
  h1(...),
  p(...)
)
```

```
<div class="
  $pkg-Simple-btn
  $pkg-Simple-fade">
  <h1>...</h1>
  <p>...</p>
</div>
```

4.5 Hygienic, Typed CSS!

```
import Simple._
val x = div(
  btn,
  fade,
  h1(...),
  p(...)
)
```

```
<div class="
  $pkg-Simple-btn
  $pkg-Simple-fade">
  <h1>...</h1>
  <p>...</p>
</div>
```

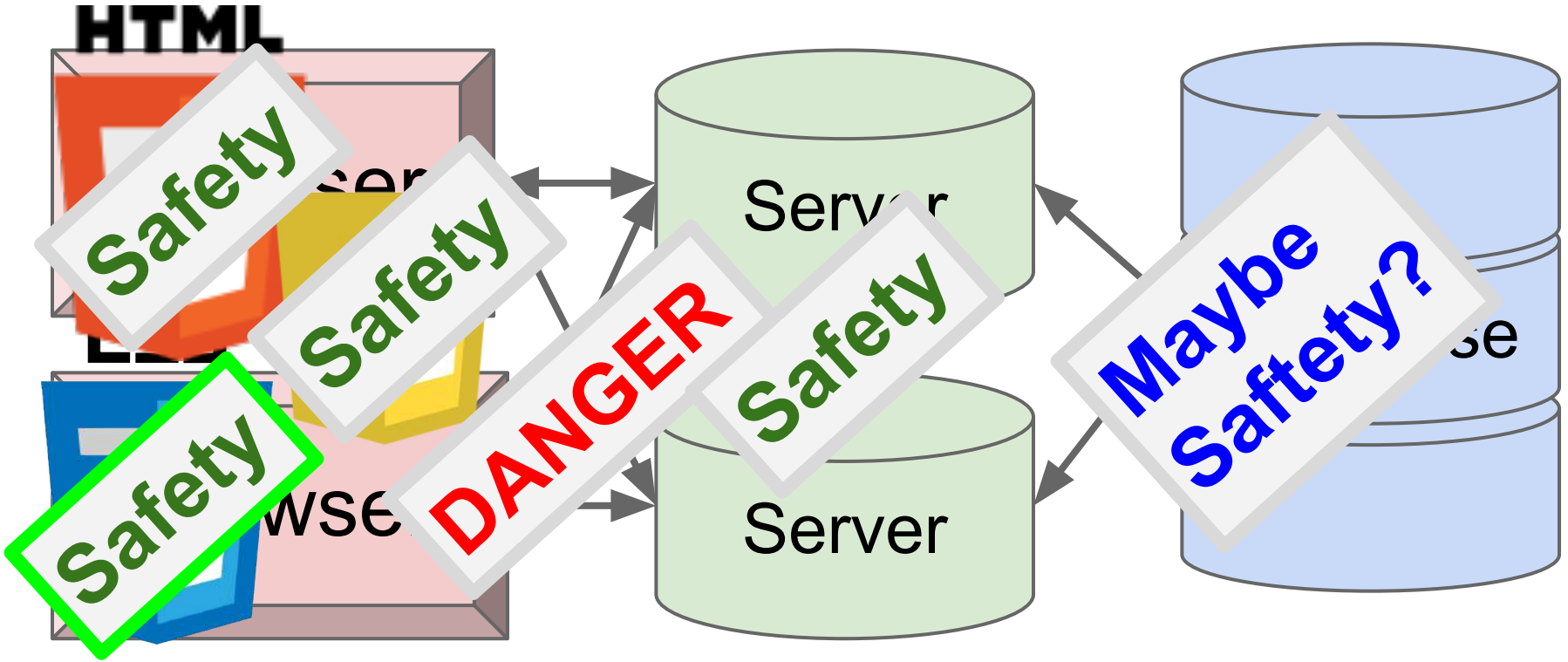
4.5 Hygienic, Typed CSS!

```
import Simple._
val x = div(
  btn,
  fadee,
  h1(...),
  p(...)
)
```

```
Not found: value fadee
  fadee,
  ^

Compilation failed
```

4.6 What is a web application?



4.7 Ajax!

```
var xhr = new XMLHttpRequest()
xhr.open("http://www.bit.ly")
xhr.onload = (x) => {
  ...
}
xhr.send()
```

```
import dom._
var xhr = new XMLHttpRequest()
xhr.open("http://www.bit.ly")
xhr.onload = (x:dom.Event) => {
  ...
}
xhr.send()
```

4.7 Ajax!

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

4.7 Ajax!

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

What if we could check this path?

4.7 Ajax!

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

What if we could check this path?

And this value?

4.7 Ajax!

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

What if we could check this path?

And this value?

And that we're using this res the right way?

4.7 Typed Ajax!

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

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

4.8 Typed Ajax!

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

4.8 Typed Ajax!

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

```
value lsit is not a member of Api
```

Compilation failed

4.8 Typed Ajax!

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

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

Compilation failed

4.8 Typed Ajax!

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

```
type mismatch; found: Future[Seq[String]] ...
```

Compilation failed

4.9 Typed Ajax!

```
// API implementation
object Server extends Api{
  def list(value: String) = ...
}
// API definition
trait Api{
  def list(value: String): Seq[String]
}
```

4.9 Typed Ajax!

```
// API implementation
object Server extends Api{
  def list(value: String, idx: String) = ...
}
// API definition
trait Api{
  def list(value: String): Seq[String]
}
```

4.9 Typed Ajax!

```
// API implementation
object Server extends Api{
  def list(value: String, idx: String) = ...
} Compilation failed: method list(...) in trait Api is not defined...
// API definition
trait Api{
  def list(value: String): Seq[String]
}
```

4.9 Typed Ajax!

```
// API implementation
object Server extends Api{
  def list(value: String, idx: String) = ...
}
// API definition
trait Api{
  def list(value: String, idx: String): Seq[String]
}
```

4.10 Typed Ajax!

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

Not enough arguments for method list(value: String, idx: Int)...

Compilation failed

4.10 Typed Ajax!

```
// API Usage
val res: Future[Seq[String]] =
  Ajax[Api].list(inputBox.value, 123).call()
```

4.10 Typed Ajax!

```
// API Usage
val res: Future[Seq[String]] =
  Ajax[Api].list(inputBox.value, 123).call()
```

Got Int, expected String

```
 Ajax[Api].list(inputBox.value, 123).call()
                                    ^
```

Compilation failed

4.10 Typed Ajax!

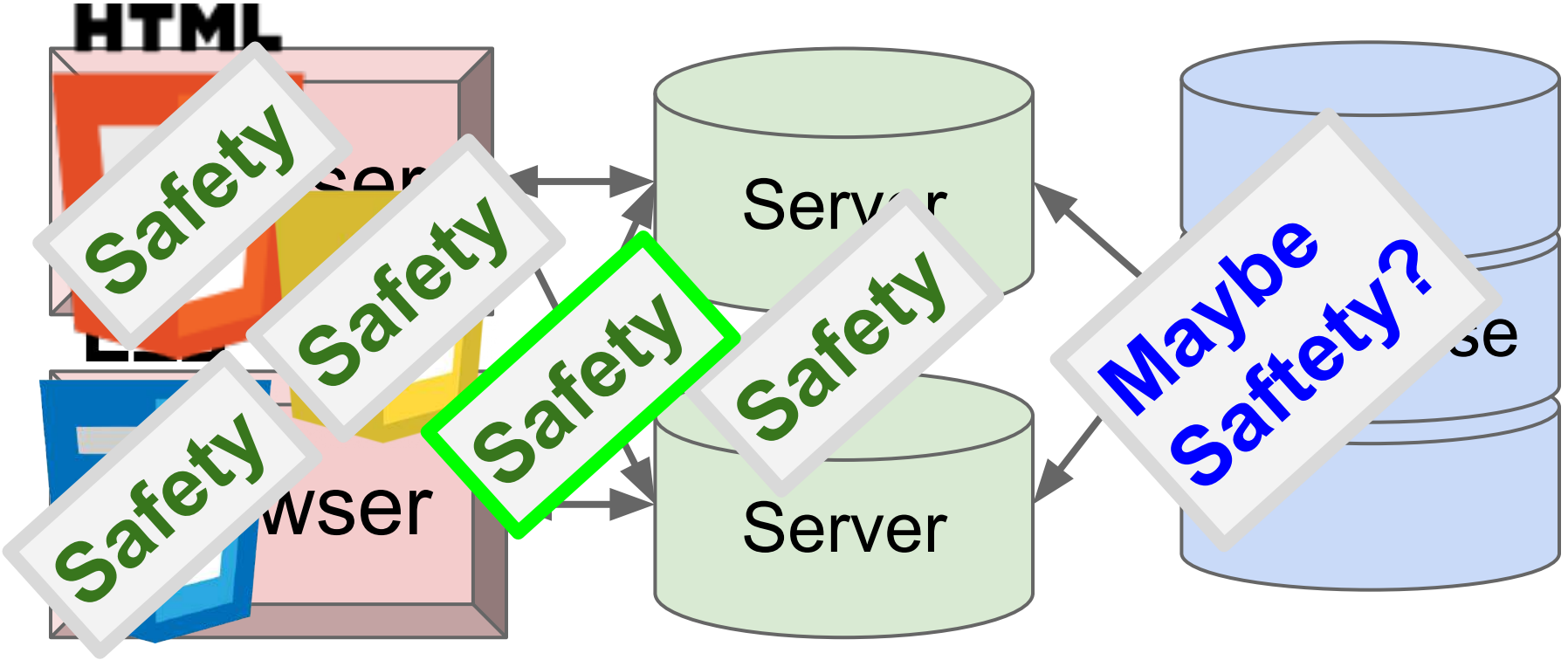
```
// API Usage
val res: Future[Seq[String]] =
  Ajax[Api].list(inputBox.value, "123").call()
```

4.10 Typed Ajax!

```
// API Usage
val res: Future[Seq[String]] =
  Ajax[Api].list(inputBox.value, "123").call()
```

Compilation Succeeded ^_^

5.1 What is a web application?



5.1 What is a web application?

5.2 Scala.js gives you...

Everything that

- ES6 gives (string-interp, const, class, =>, …)
- Typescript gives (types, generics, …)

- Monads
- Higher Kinded Types
- Type-level Compuations

5.3 Failure-modes we Conquered

undefined is not a function

Mal-formed HTML

Un-used CSS classes

Using un-defined CSS classes

CSS class-name collisions

Mal-formed Ajax requests

5.4 Safety & Sanity on the Web

Scala.js: expressive, safe web applications

Enforce safety throughout the entire application, not just the Javascript

Not 12 months from now, but today!