1024

X

768

Towards Browser and Server Utopia

Richard Dallaway, @d6y

underscore.io

127.0.0.1:8080

Editing: default

```
1  Ah freddled gruntbuggly,
2  Thy micturations are to me,
3  As plurdled gabbleblotchits,
4  On a lurgid bee,
5
```

Agenda

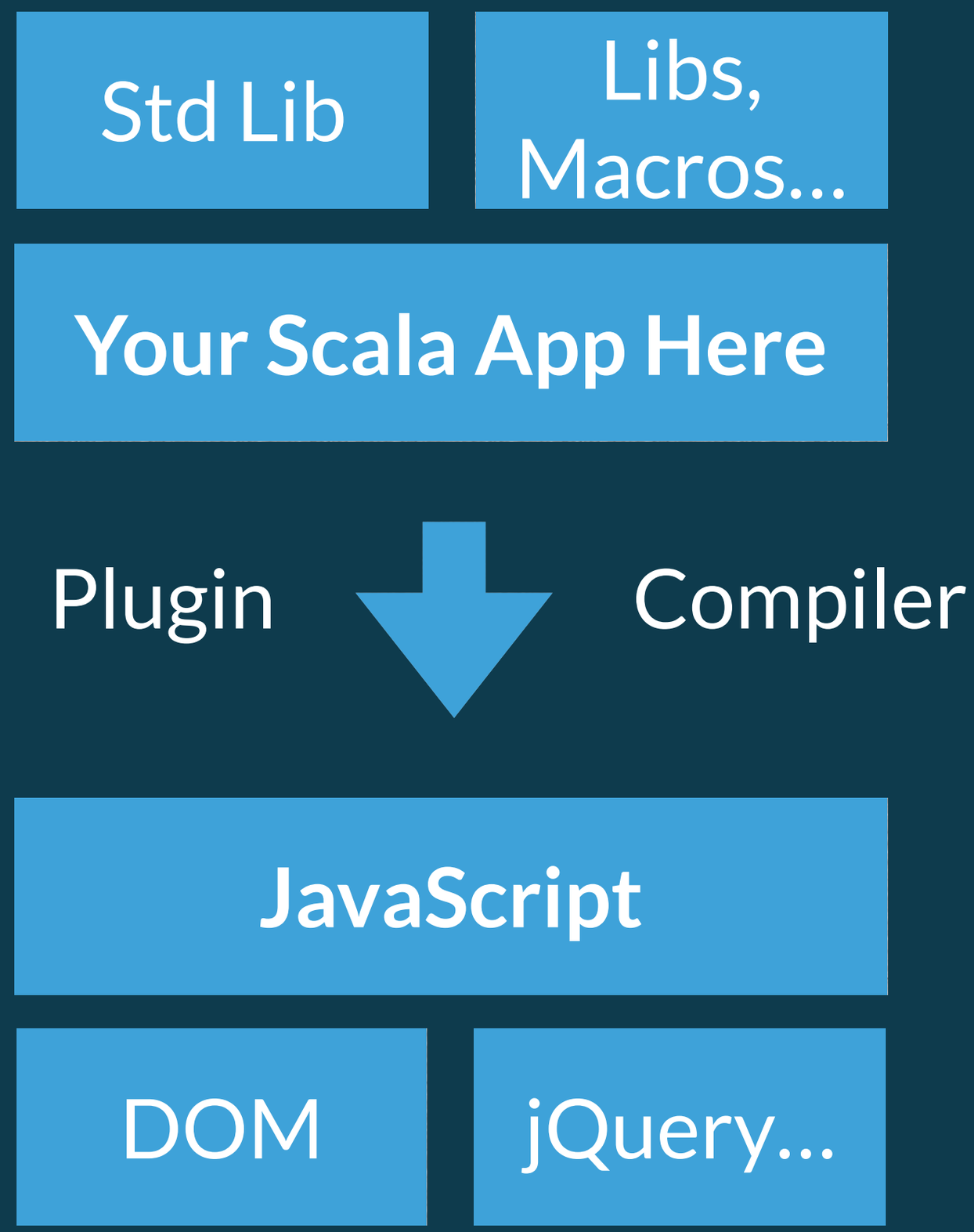
Why?
Scala.js motivation & introduction

What?
Collaborative text editing with WOOT

How?
SBT, calling to and from Javascript…

— Part 1 —

Scala.js Crash Course


Std Lib
Libs, Macros...
Your Scala App Here
Plugin
Compiler
JavaScript
DOM
jQuery...

Features

Fast Compiler

Optimized Output (this app: 295k)

Resulting JS is Fast

Ecosystem of bindings

Great interop with JS

Yes, all of Scala

But not Java or reflection-based code

Ported	New	Typed JS Bindings
Scalaz	Scalatags	scalajs-dom
ScalaCheck	uTest	scalajs-jquery
shapeless	uPickle	scalajs-react
...	...	...

```scala
// Scala
def answer = Option(41).map(_ + 1)
```

```javascript
// JavaScript
(function() {
  var o = Option().apply(41);
  if (o.isEmpty()) {
    var answer = None()
  } else {
    var arg1 = o.get();
    var answer =
      new Some().init(1 + arg1);
  };
  return answer
});
```

```scala
// Scala
def answer = Option(41).map(_ + 1)
```

```javascript
// JavaScript
(function() {
  var this$1 = $m_s_Option().apply__O__s_Option(41);
  if (this$1.isEmpty__Z()) {
    var answer = $m_s_None$()
  } else {
    var arg1 = this$1.get__O();
    var x$1 = $uI(arg1);
    var answer =
      new $c_s_Some().init___O(((1 + x$1) | 0))
  };
  return answer
});
```

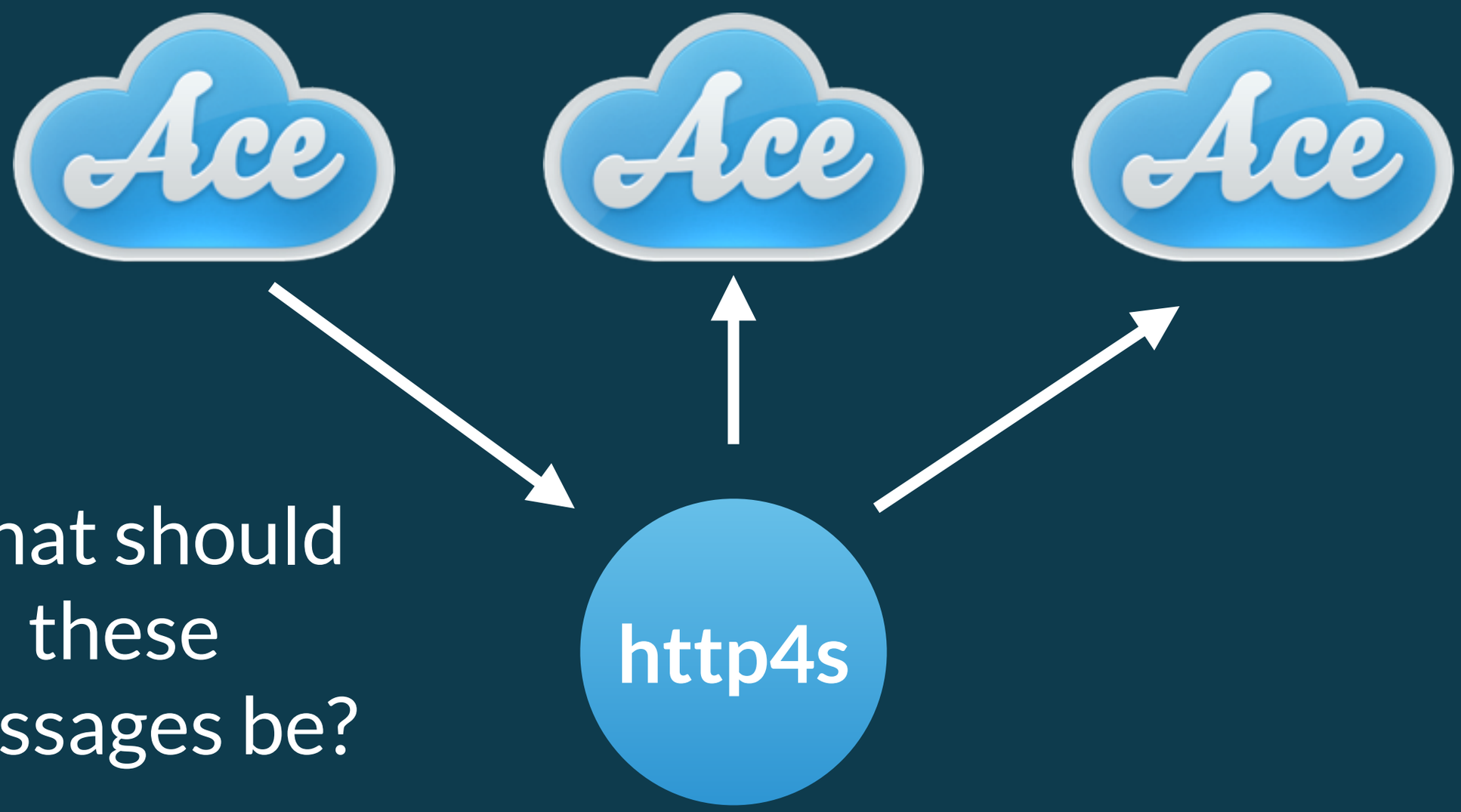
Not a framework

— Part 2 —

CRDTs

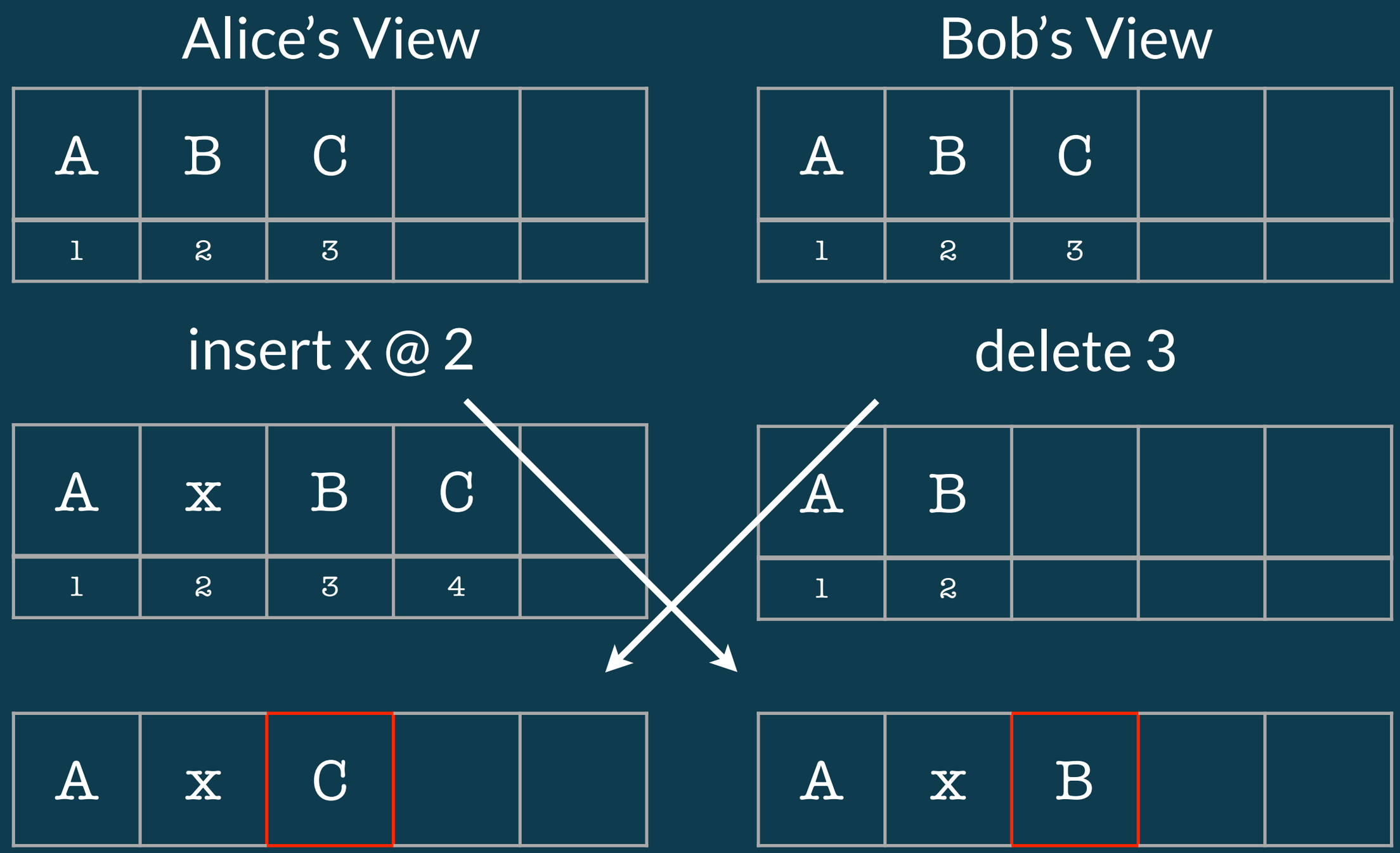
Commutative Replicated Data Type

What we want to build



What should these messages be?

Don’t Do This


Alice’s View
A B C
1 2 3
insert x @ 2
A x B C
1 2 3 4
A x C
Bob’s View
A B C
1 2 3
delete 3
A B
1 2
A x B

WOOT

Alice's View

A	B	C		
1	2	3		

Bob's View

A	B	C		
1	2	3		

A < x < B

A	x	B	C	
1	2	3	4	

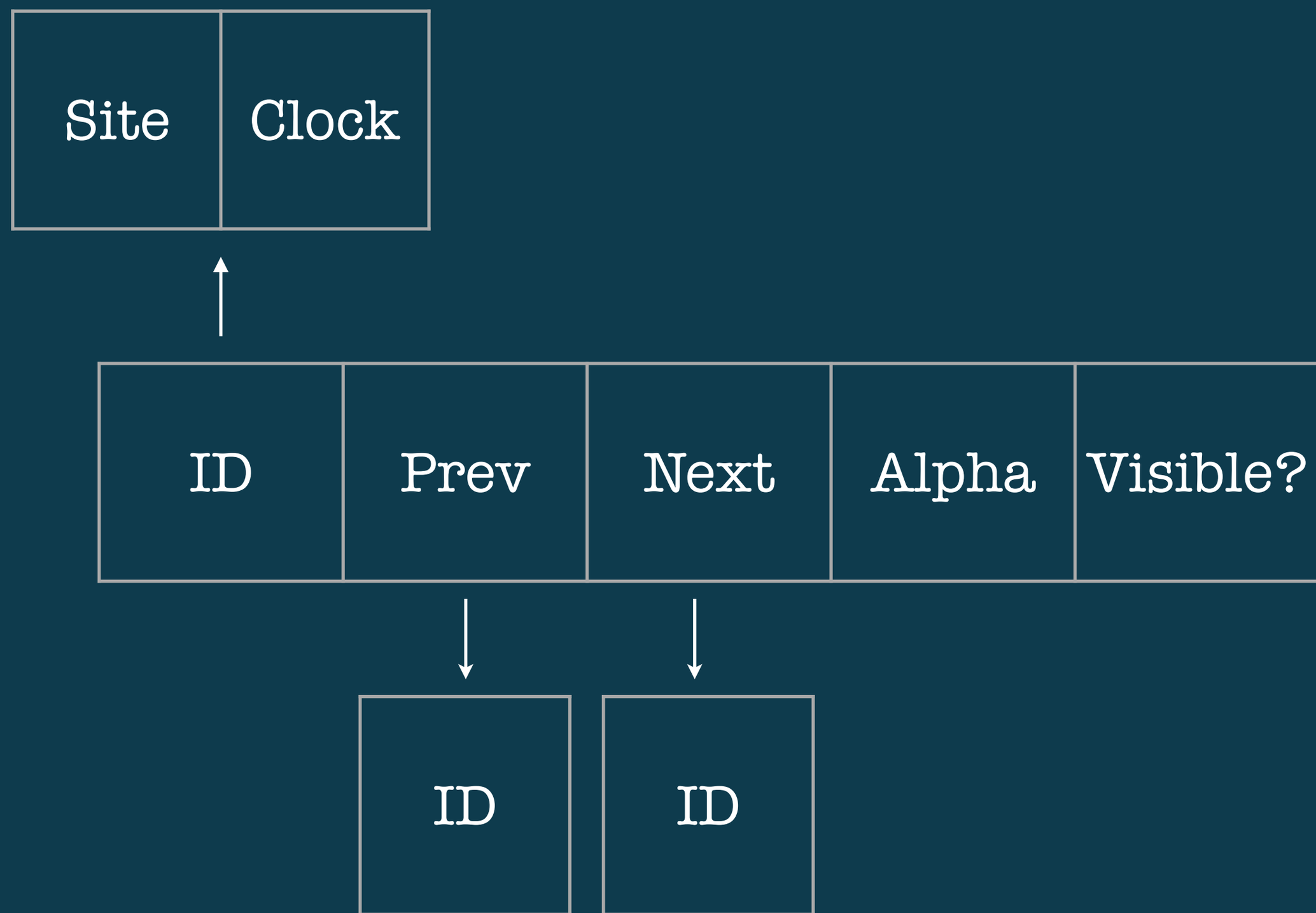
Delete C

A	B			
1	2			

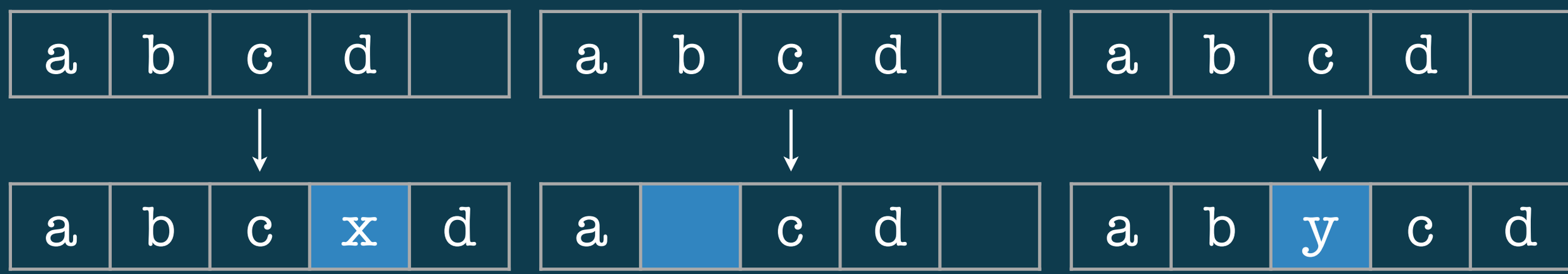
A	x	B		

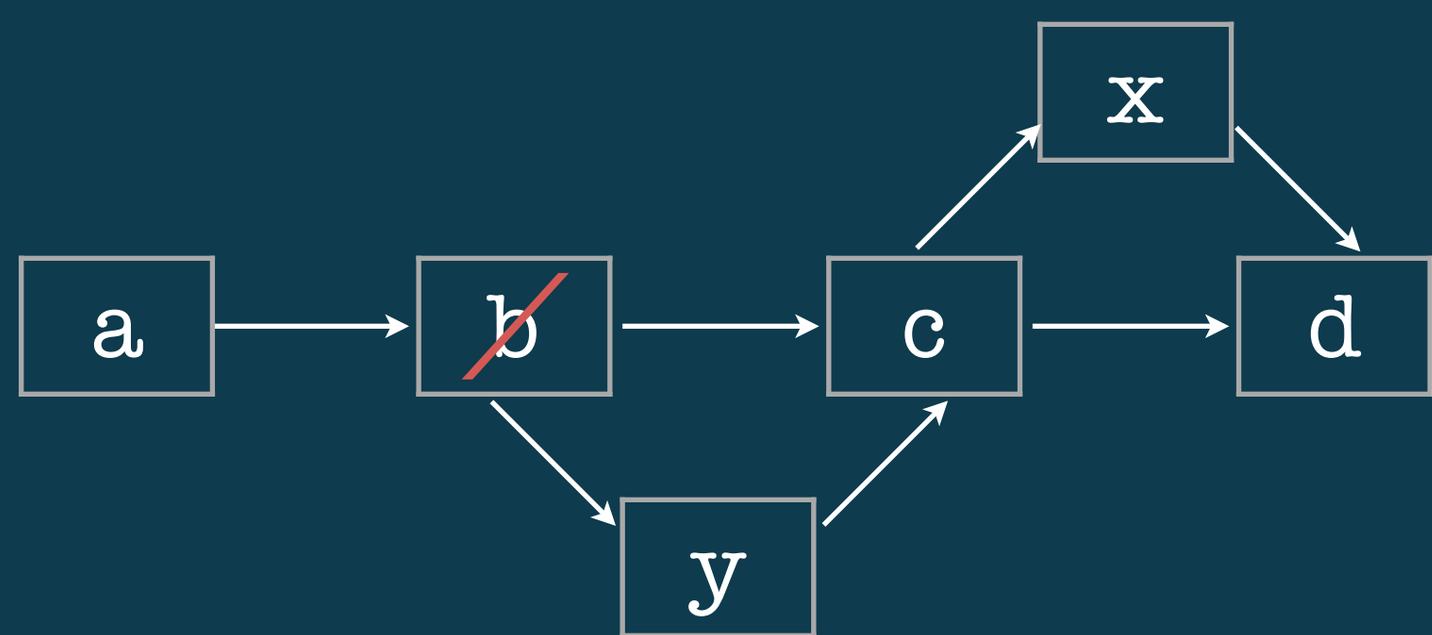
A	x	B		

Representation



Algorithm





Algorithm

Local Integration

```
(WString,Char,Pos) => (WString,WChar)
```

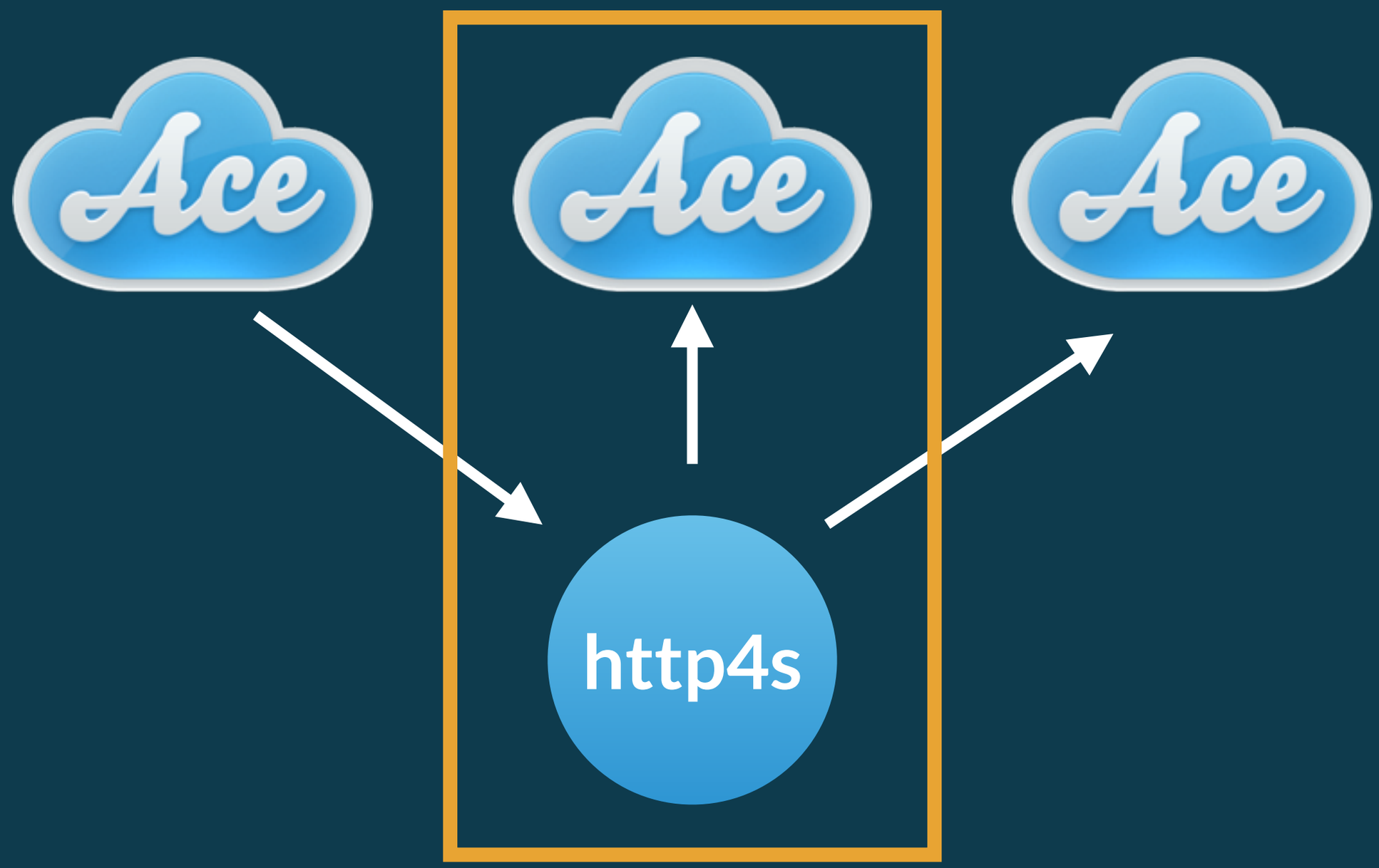
Remote Integration

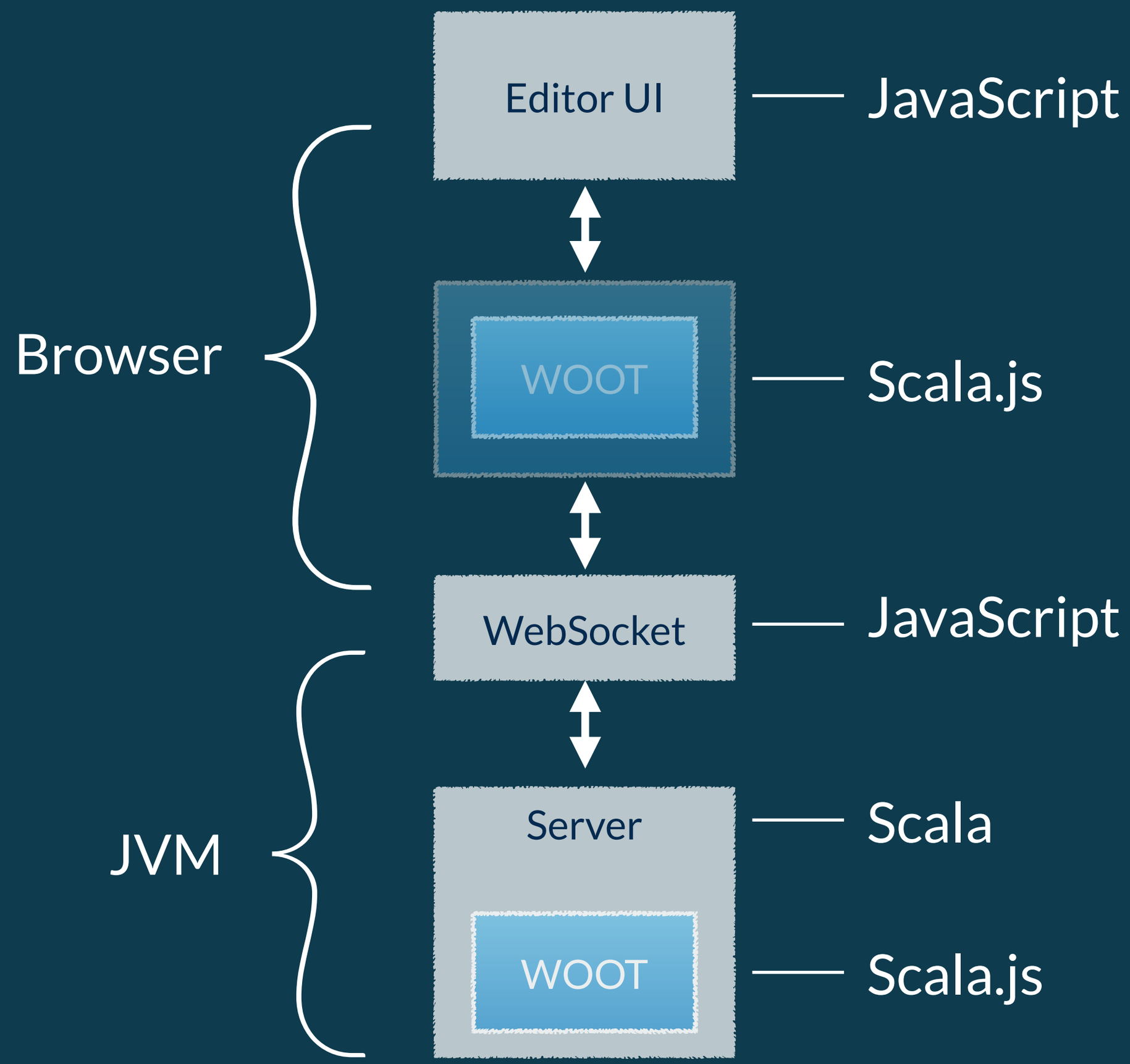
```
(WString,WChar) => WString
```

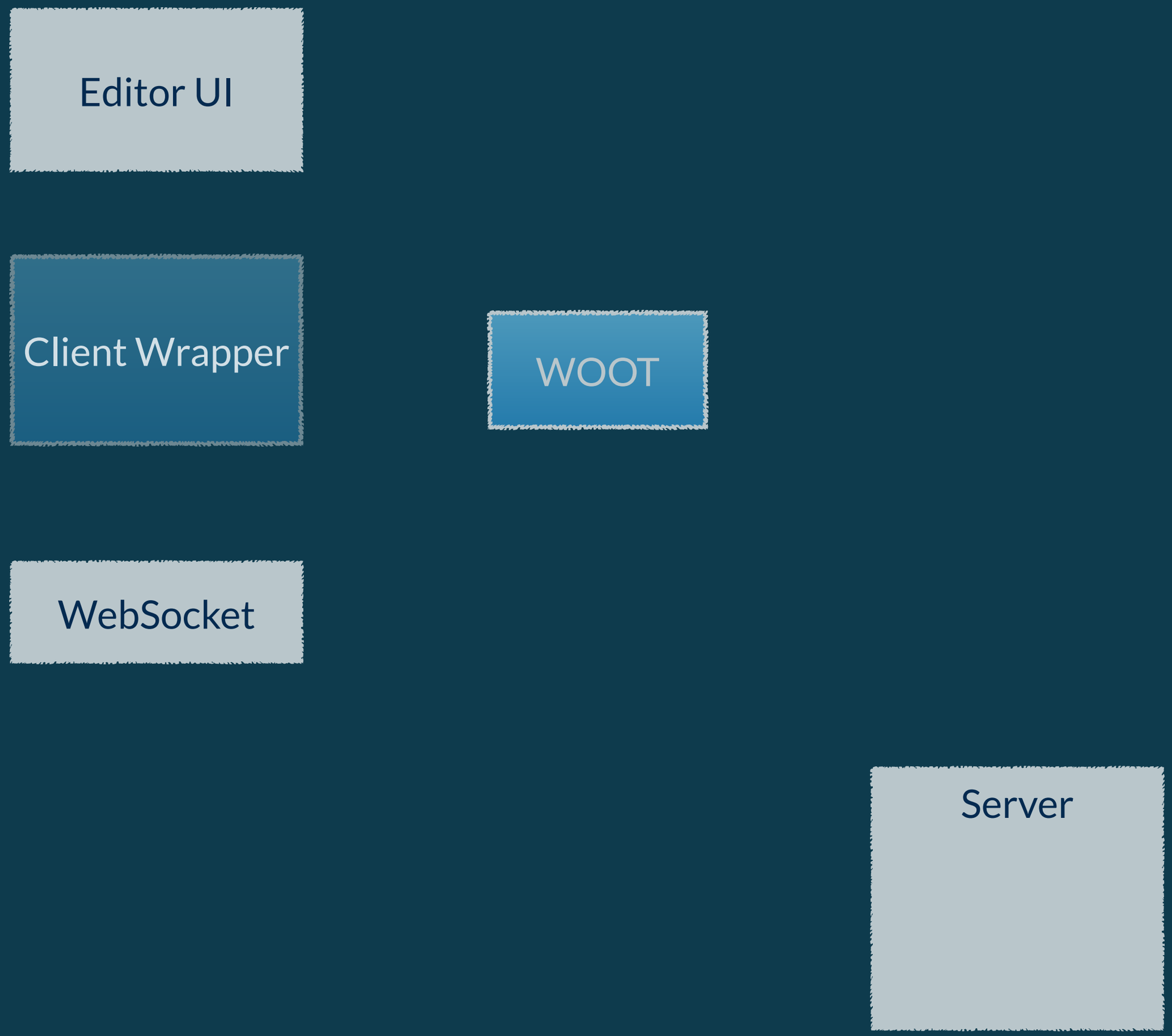
— Part 3 —

Migrating to Scala.js

What we want to build

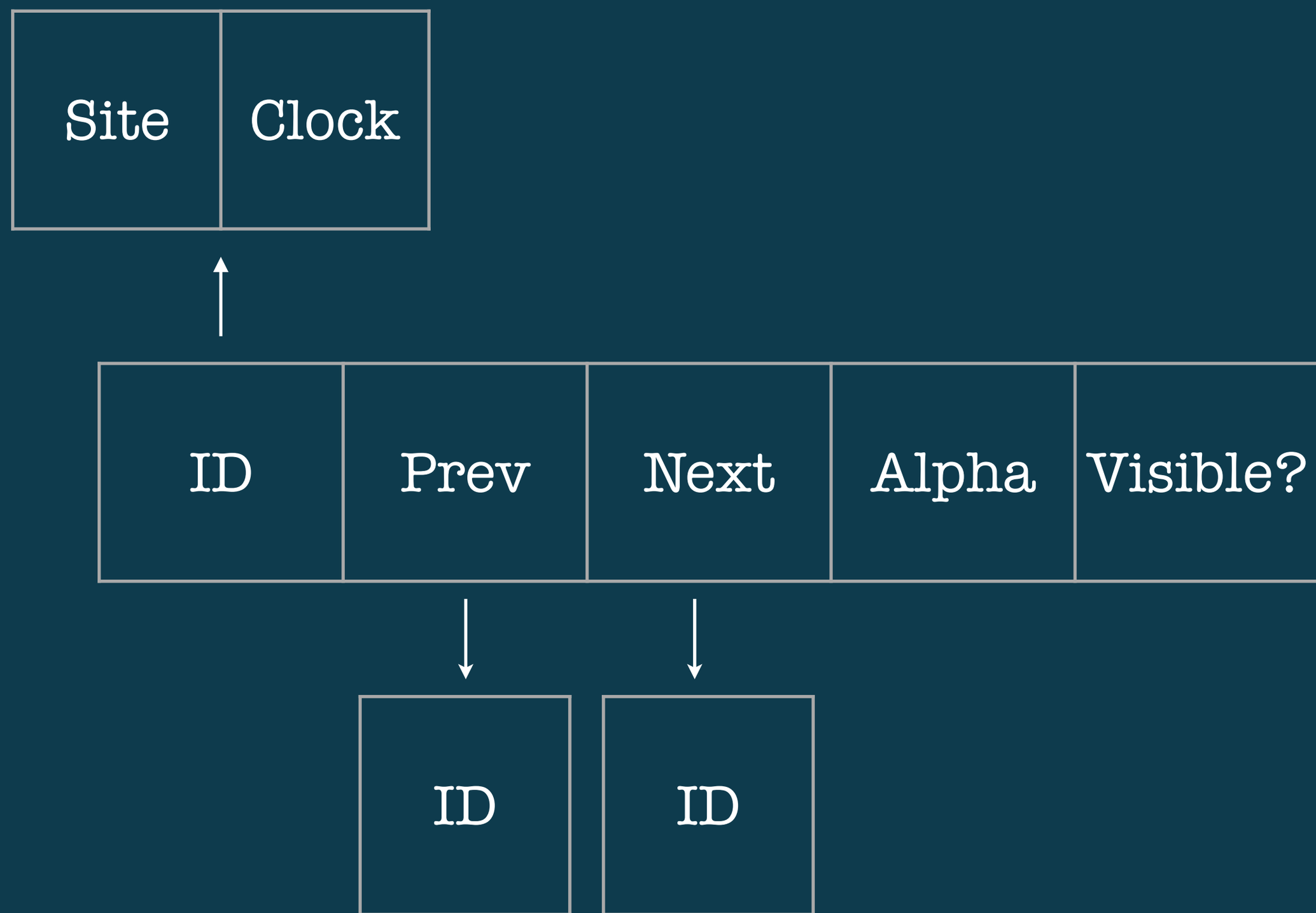



Editor UI
JavaScript
Browser
WOOT
Scala.js
WebSocket
JavaScript
JVM
Server
Scala
WOOT
Scala.js


Editor UI
Client Wrapper
WOOT
WebSocket
Server

build.sbt


shared
client
server
.sjsir
.class
.class
.js
.js
> compile
> fastOptJS
> fullOptJS


Site
Clock
ID
Prev
Next
Alpha
Visible?
ID
ID

```scala
// Scala
import js.annotation.JSExport

@JSExport
case class WChar(
          id: Id,
       alpha: Byte,
        prev: Id,
        next: Id,
   isVisible: Boolean = true)
```

OK, but...

Should be Char?

```
case class WChar(
        id: Id,
     alpha: Byte
      prev: Id,
      next: Id,
 isVisible: Boolean = true)
```

Can’t this whole API be easier to use?

Differences

Semantics = Scala Semantic

But…

JavaScript has no Char

toString differences

…not as many as you think

Wrapper Solution

```
package client

import js.annotation.JSExport
import woot.WString

@JSExport
class WootClient() {

  var doc = WString.empty()

  @JSExport
  def insert(s: String, pos: Int): Json = ???

  @JSExport
  def ingest(json: Json): Unit = ???
}
```

uPickle

```scala
sealed trait Operation {
  def wchar: WChar
}

case class InsertOp(override val wchar: WChar) extends Operation
case class DeleteOp(override val wchar: WChar) extends Operation
```

```
["woot.InsertOp", { "wchar": {
        "id": ["woot.CharId", {...}],
     "alpha": "*",
      "prev": ["woot.Beginning",{}],
      "next": ["woot.Ending",   {}]}
  }
]
```

```scala
@JSExport
class WootClient() {

  var doc = WString.empty()

  @JSExport
  def insert(s: String, pos: Int): Json = ???


  @JSExport
  def ingest(json: Json): Unit = ???
}
```

```scala
@JSExport
class WootClient() {

  var doc = WString.empty()

  @JSExport
  def insert(s: String, pos: Int): Json = {
    val (op, wstring) = doc.insert(s.head, pos)
    doc = wstring
    write(op)
  }

  @JSExport
  def ingest(json: Json): Unit = ???
}
```

Effecting the DOM

```
import org.scalajs.dom
val element = dom.document.getElementById("editor")
```

Effecting the DOM

```scala
import org.scalajs.dom
val element = dom.document.getElementById("editor")
```

Effecting the DOM

```scala
import org.scalajs.dom
val element = dom.document.getElementById("editor")
```

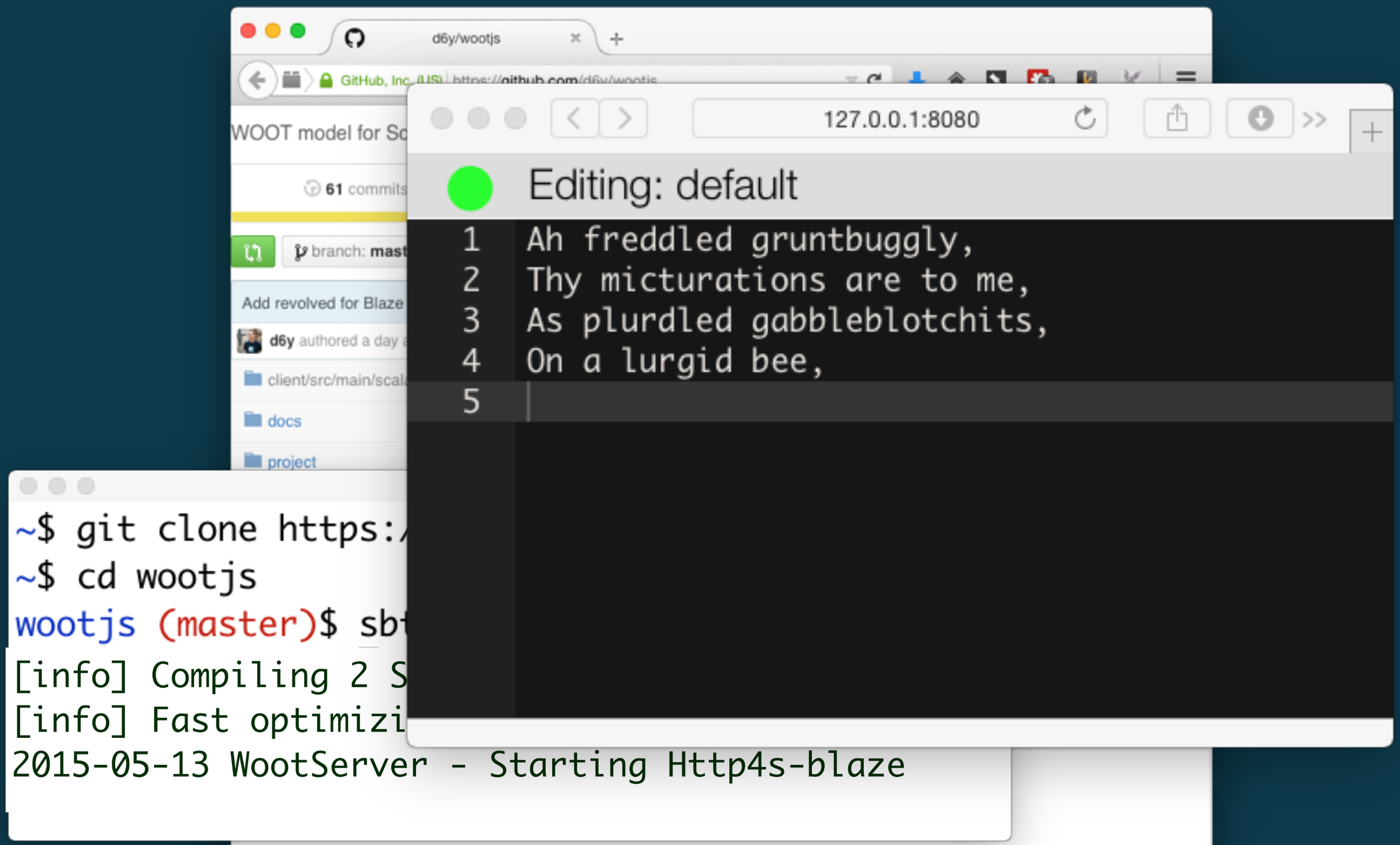
Effecting the DOM

```
// JavaScript
var updateEditor = function(ch, isVisible) {
  var delta = convertWootToAceCoordinates(…);
  editor.getSession()
        .getDocument().applyDeltas([delta]);
}

jQuery(document).ready(function() {
  client = new client.WootClient(updateEditor);
});
```

```
@JSExport
class WootClient() {

  var doc = WString.empty()

  @JSExport
  def ingest(json: Json): Unit = ???
}
```

```scala
@JSExport
class WootClient(f: js.Function2[String,Boolean,Unit]) {

  // JavaScript
  var updateEditor = function(ch, isVisible) {…

  def applyOperation(op: Operation): Unit = {
    val (ops, wstring) = doc.integrate(op)

    // Become the updated document:
    doc = wstring

    // Side effects:
    ops.foreach {
      case InsertOp(ch) => f(ch.alpha.toString, true)
      case DeleteOp(ch) => f(ch.alpha.toString, false)
    }
  }
}
```

github.com/d6y/wootjs



Talk about ScalaCheck
here if time

What we've Seen

Multi-project build ✓

Wrote Scala, ran it both places ✓

Great interop (call JS, be called by JS) ✓

Dirty dirty dynamic calls ✓

Used cross-compiled libraries ✓

Benefits?

IDE support

Single language

Write once, run both places

“It’s the types, stupid”

Thanks!

Richard Dallaway, @d6y

https://github.com/d6y/wootjs

underscore.io