ESUG 2022, Нови Сад (Novi Sad), Србија (Serbia)

PharJS for Real World Applications

Noury Bouraqadi & Dave Mason

IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

Toronto Metropolitan University

New to Smalltalk ?

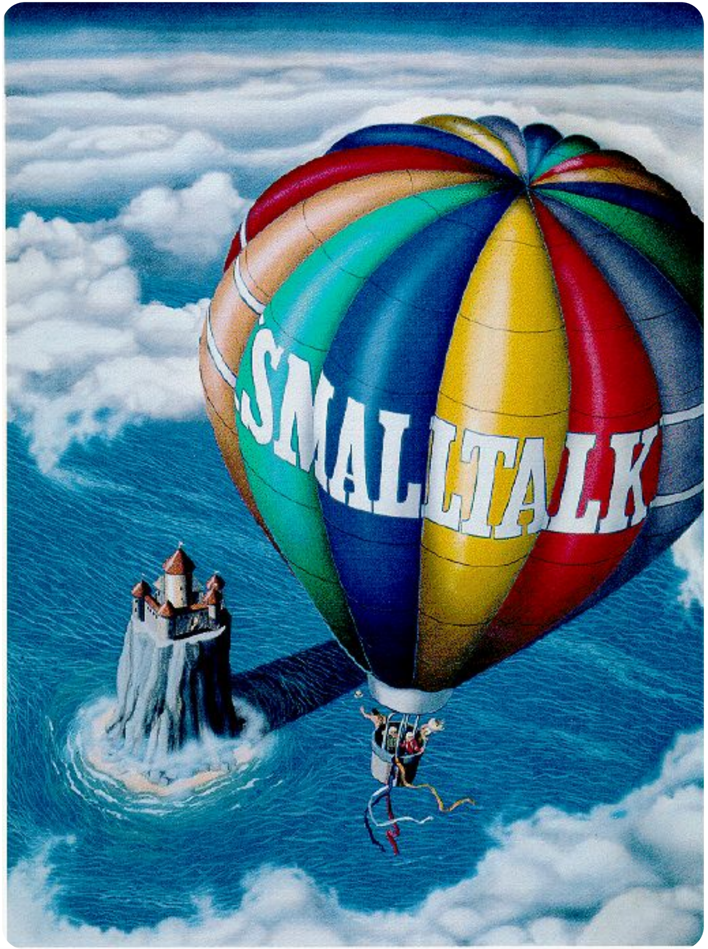
Smalltalk is dangerous.
It is a drug.

My advice to you would be

don’t try it.

It could ruin your life

– Andy Bower, CEO of Object Arts Ltd.



= Great Language, Libraries, Tools, Community

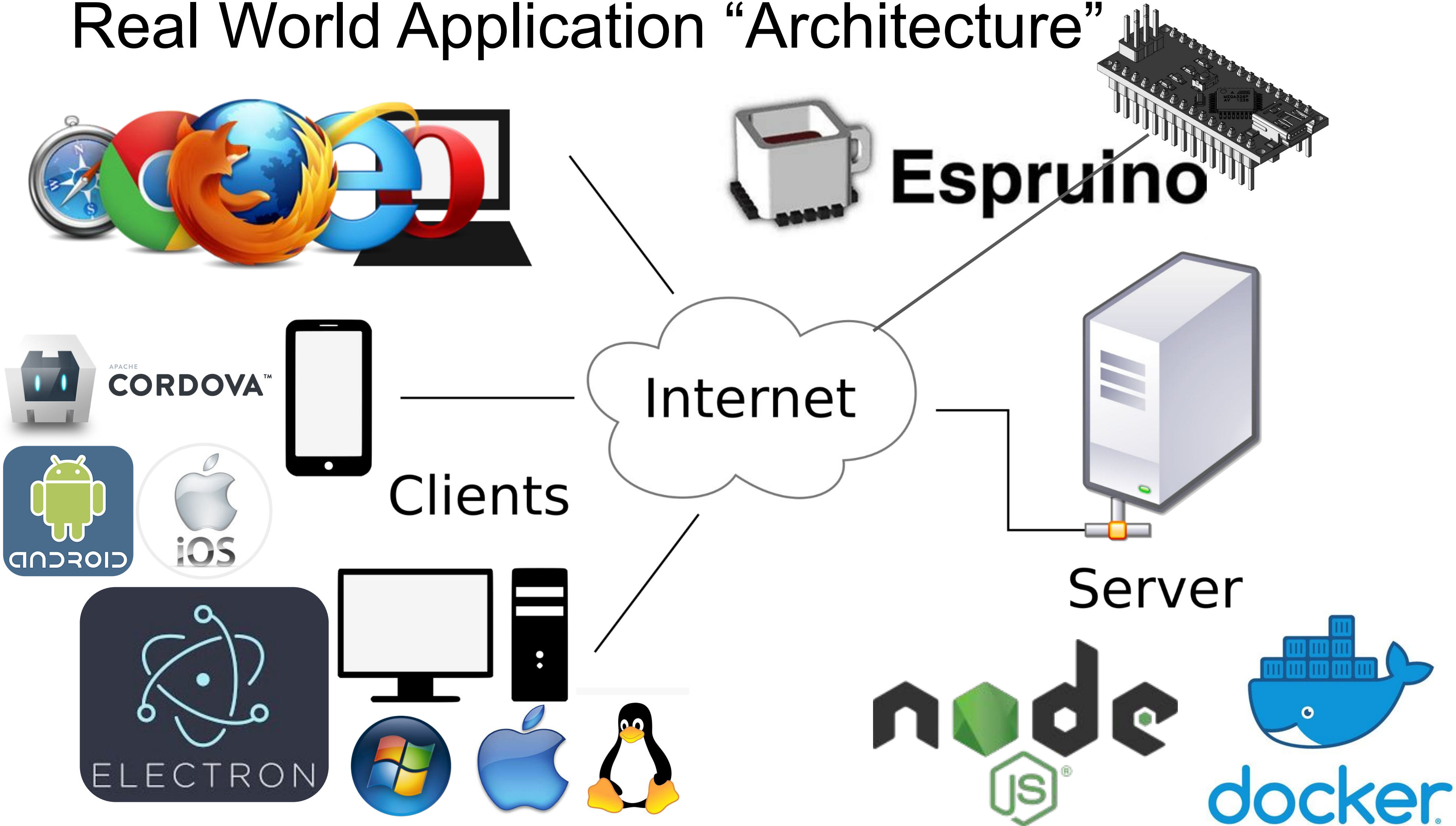
We want to develop in

Smalltalk
All the Time
Everywhere

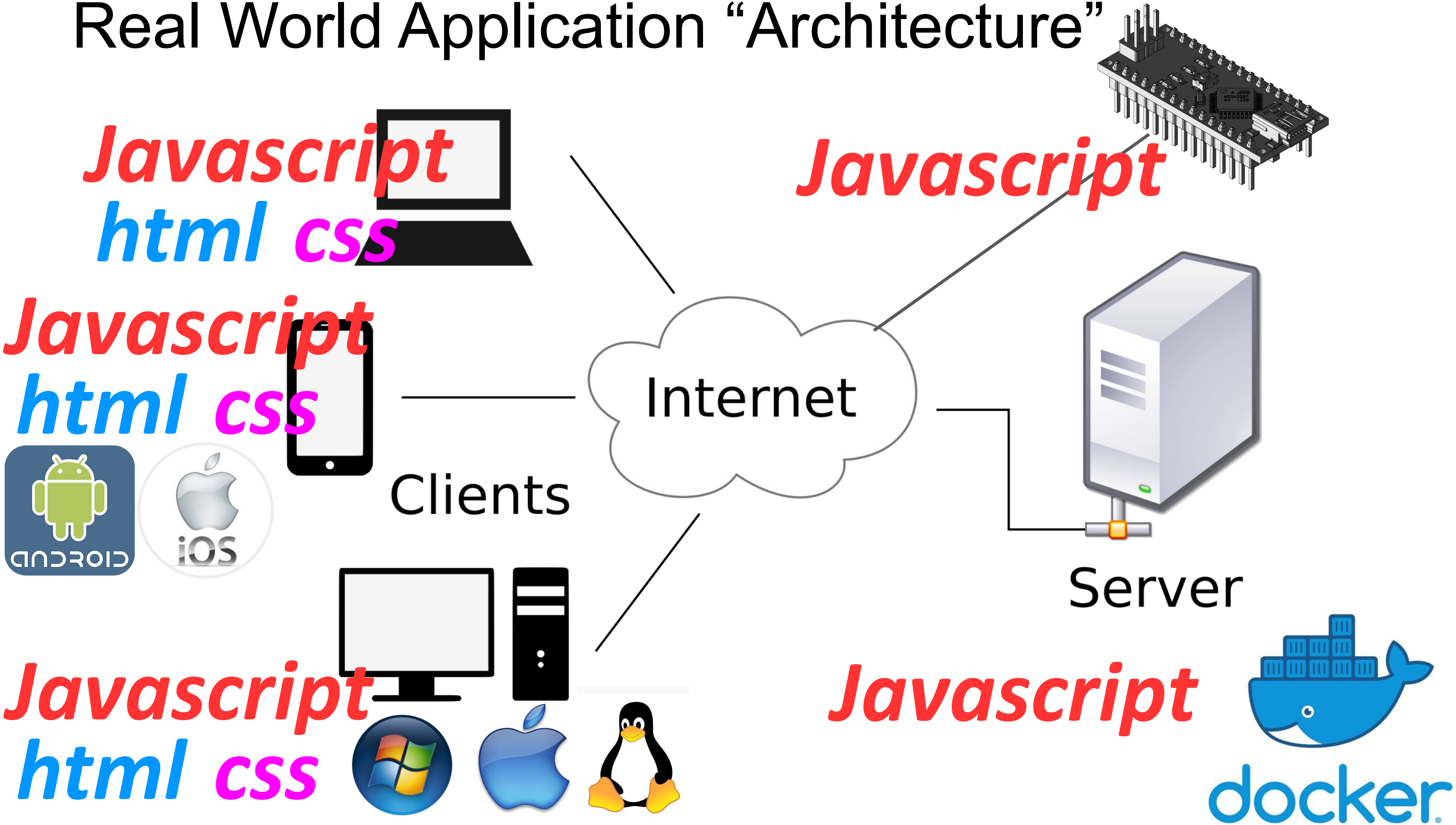
What to do with non-Smalltalk Resources?

HTML
5
CSS
3
JS
5
php
SQL
++

Real World Application “Architecture”



Real World Application “Architecture”



Javascript

html css

Javascript

html css

Clients

Internet

Javascript

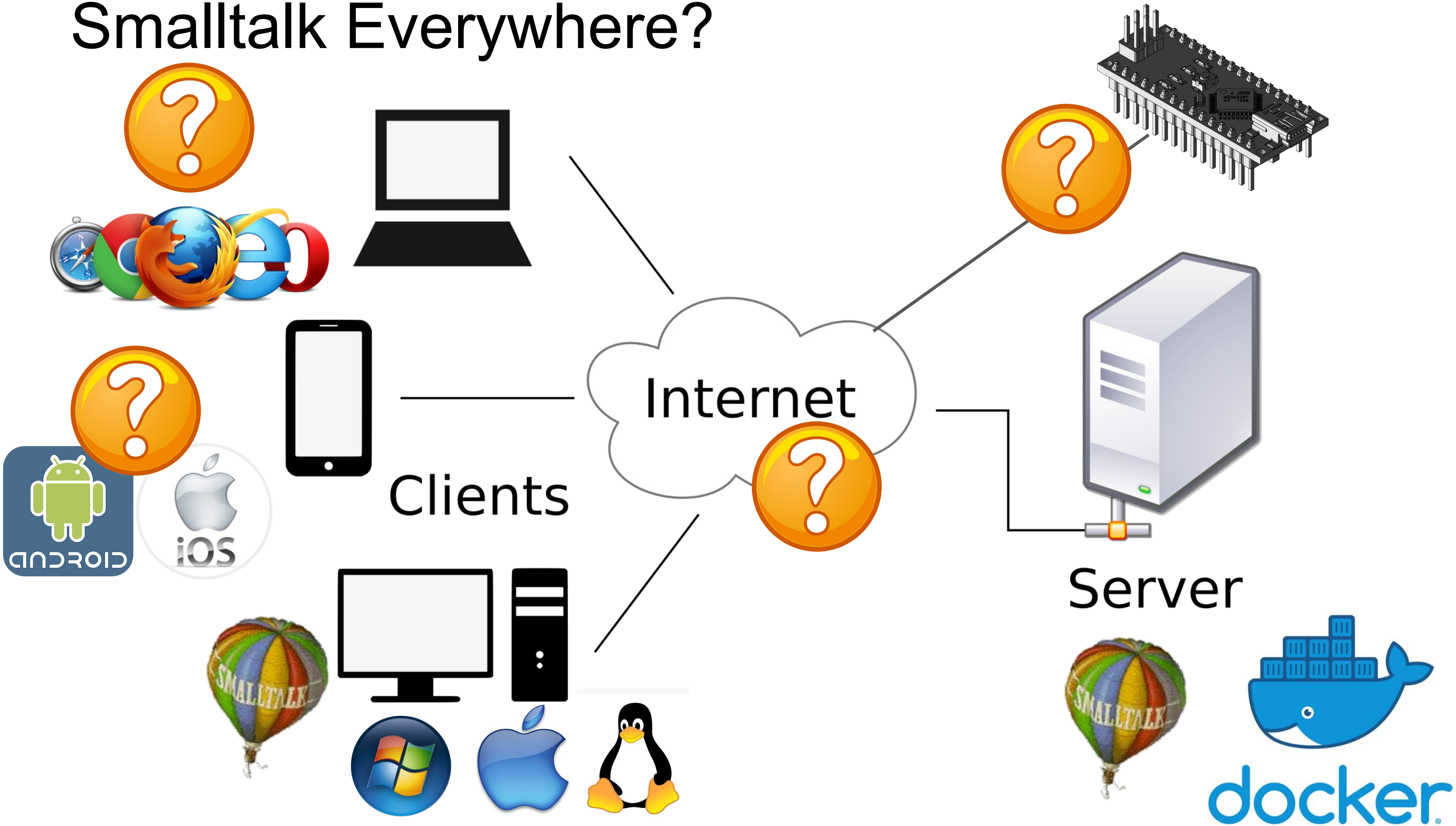
Server

Javascript

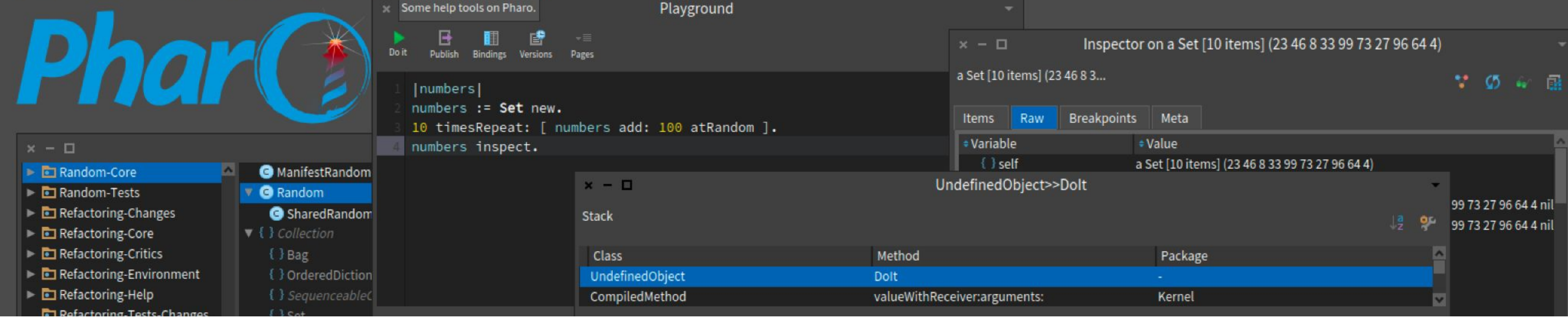
html css

Javascript

Smalltalk Everywhere?



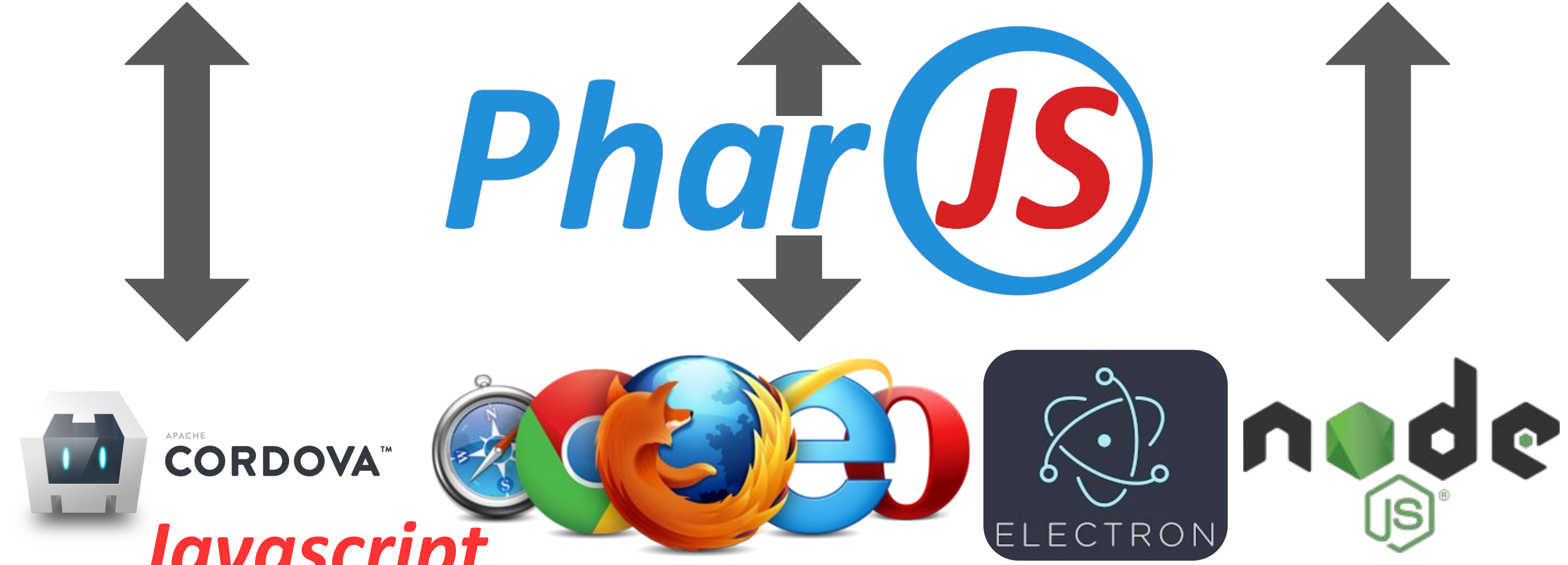
PharoJS



```
|numbers|
numbers := Set new.
10 timesRepeat: [ numbers add: 100 atRandom ].
numbers inspect.
```

Variable	Value
self	a Set [10 items] (23 46 8 33 99 73 27 96 64 4)

Class	Method	Package
UndefinedObject	DoIt	-
CompiledMethod	valueWithReceiver:arguments:	Kernel



Javascript

PharoJS

- Develop in Pharo Smalltalk all the time!

Espruino

- Reuse existing JS libraries

- JS Portability

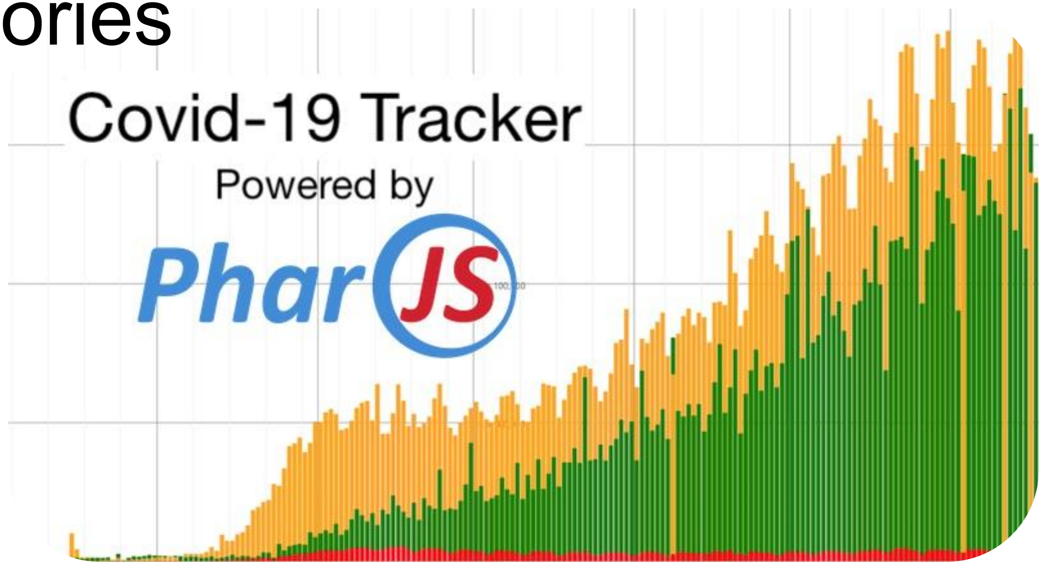
- JS Run-time Speed

```
me@linuxbox:~$
```

PharJS

- **Transpiler:** Converts Pharo Code to JavaScript
- **Framework:** Develop JS applications in Pharo
- **Libraries:** Extend JS Objects with Pharo’s Behavior
- **Tools:** Playground + Inspector for JS Objects
- **Test Framework:** Test JS Code

PharJS Success Stories



Covid-19 Tracker

Powered by

PharJS

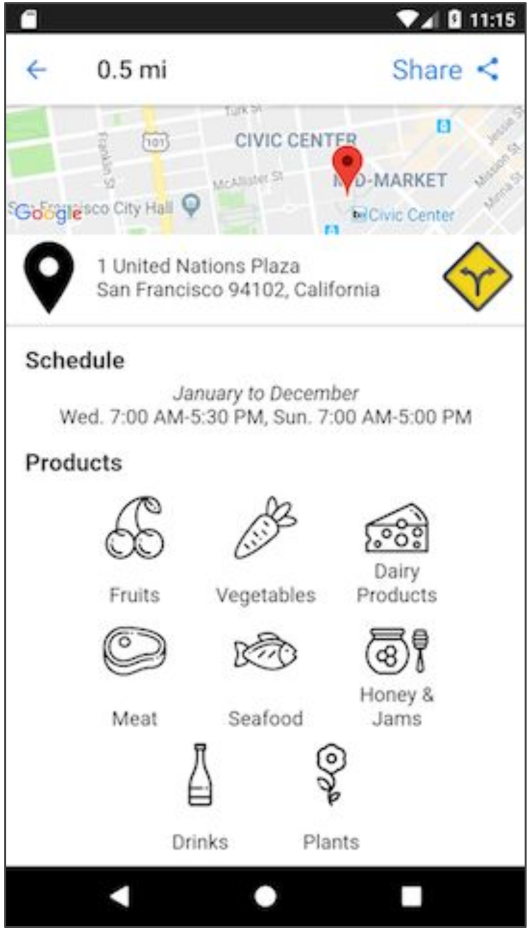
PharJS

Smalltalk REPL

PLC3000

Teaching PLC Automation
Made Easy

PharoJS Success Stories



11:15

0.5 mi Share

1 United Nations Plaza
San Francisco 94102, California

Schedule

January to December
Wed. 7:00 AM-5:30 PM, Sun. 7:00 AM-5:00 PM

Products

Fruits
Vegetables
Dairy Products
Meat
Seafood
Honey & Jams
Drinks
Plants

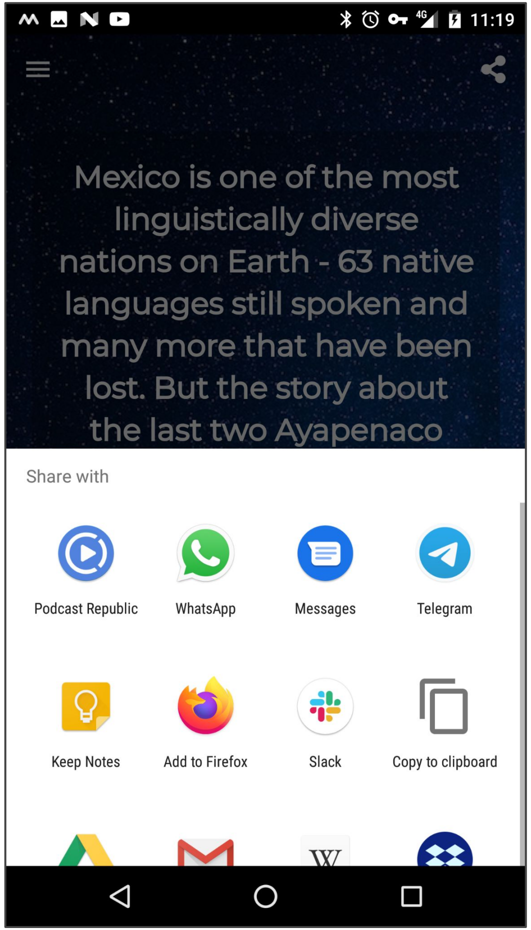
android

iOS

Farmers Market

Brain Treats

Fun Facts Everyday



11:19

Mexico is one of the most linguistically diverse nations on Earth - 63 native languages still spoken and many more that have been lost. But the story about the last two Ayapenaco

Share with

Podcast Republic
WhatsApp
Messages
Telegram
Keep Notes
Add to Firefox
Slack
Copy to clipboard

Pharo 100%

Development

Production

APACHE CORDOVA™

100% Javascript

Pharo
100%

1. Write Tests
2. Pass the tests
3. Export to JS

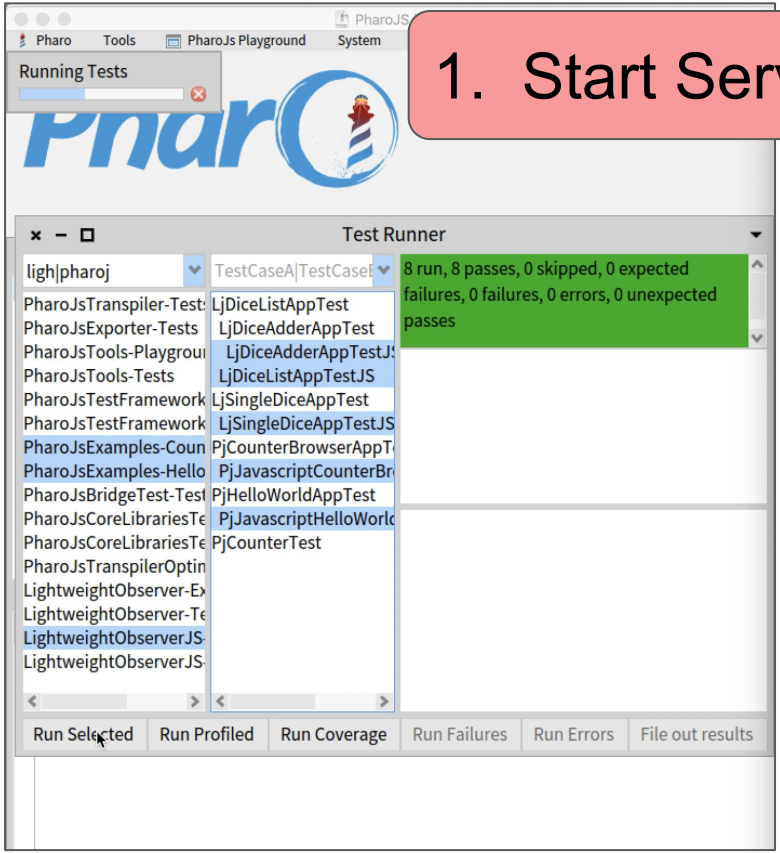
APACHE
CORDOVA™

100%
Javascript

PharJS

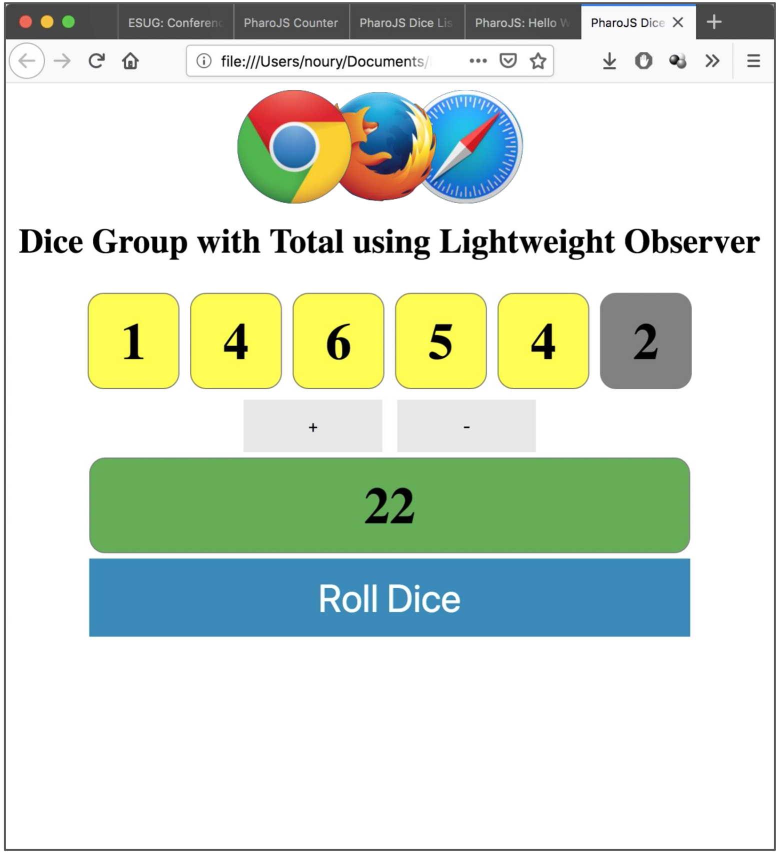
Testing JS Generated Code

PharoJS Tests Talk to Web Browsers

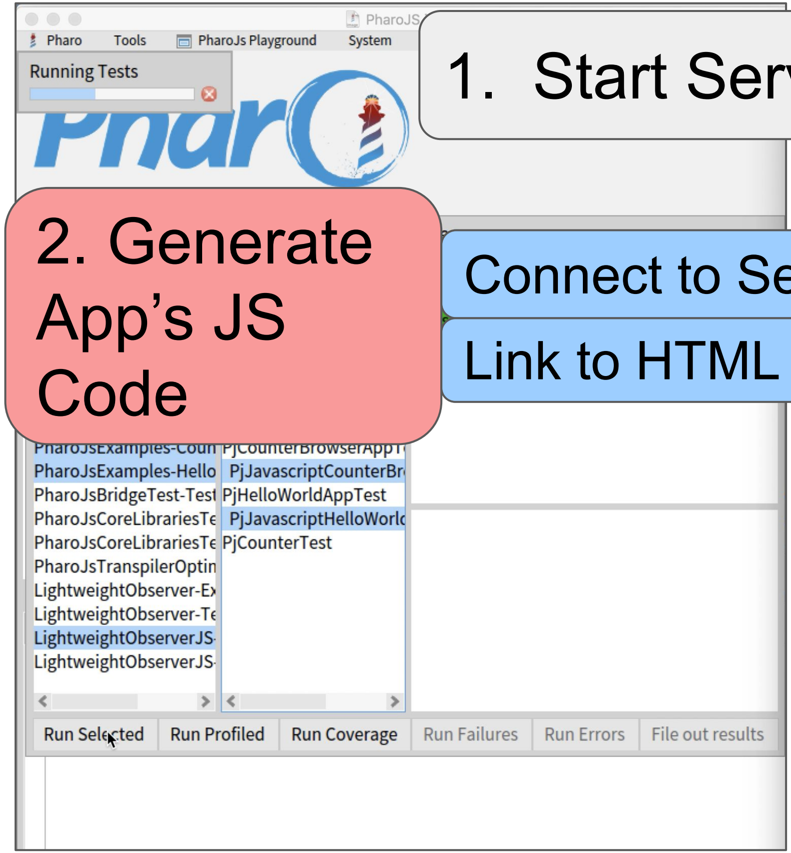


8 run, 8 passes, 0 skipped, 0 expected failures, 0 failures, 0 errors, 0 unexpected passes

1. Start Server



PharoJS Tests Talk to Web Browsers



PharoJsExamples-Coun
PharoJsExamples-Hello
PharoJsBridgeTest-Test
PharoJsCoreLibrariesTe
PharoJsCoreLibrariesTe
PharoJsTranspilerOptin
LightweightObserver-Ex
LightweightObserver-Te
LightweightObserverJS-
LightweightObserverJS-

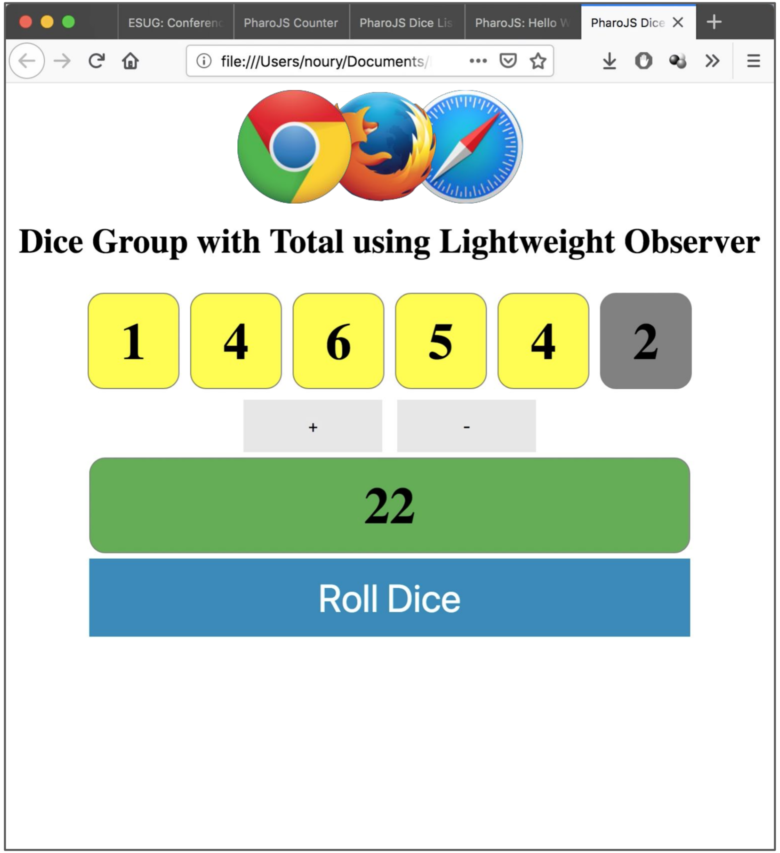
PjCounterBrowserAppT
PjJavascriptCounterBr
PjHelloWorldAppTest
PjJavascriptHelloWorld
PjCounterTest

1. Start Server

2. Generate App's JS Code

Connect to Server

Link to HTML

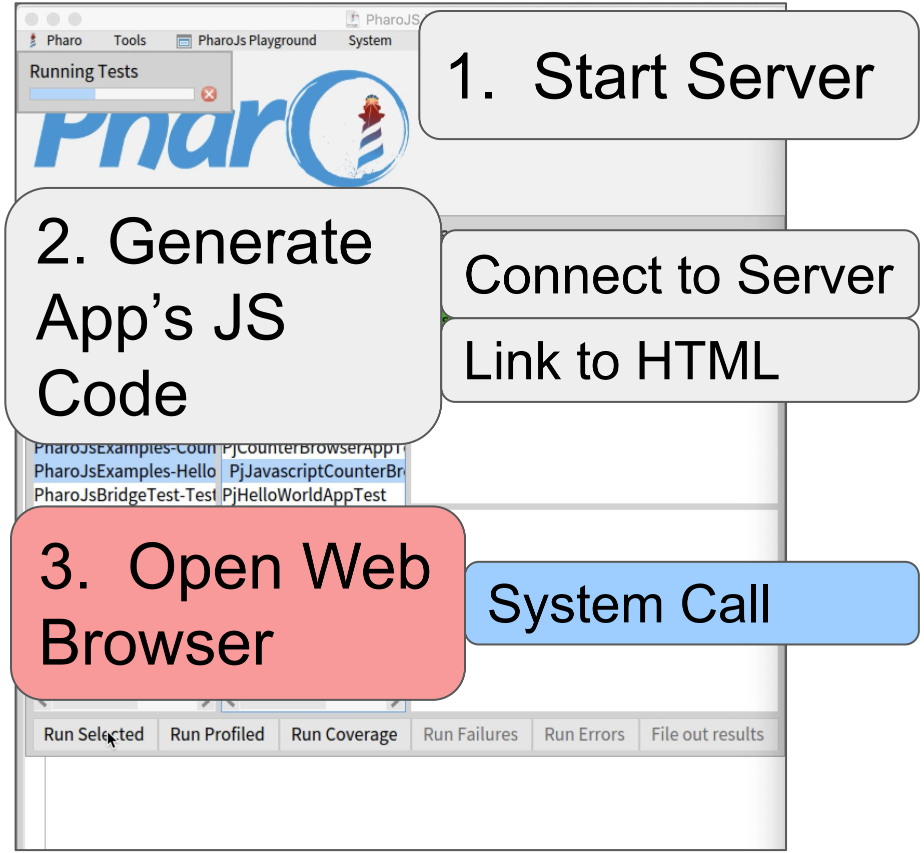


Dice Group with Total using Lightweight Observer

1	4	6	5	4	2

22

PharoJS Tests Talk to Web Browsers



1. Start Server

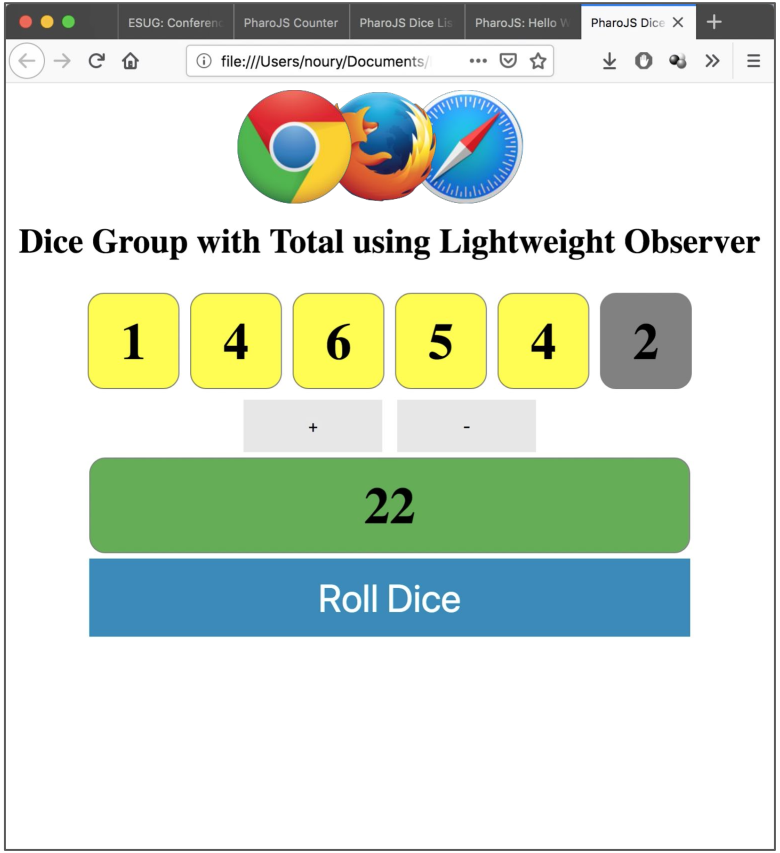
2. Generate App's JS Code

Connect to Server

Link to HTML

3. Open Web Browser

System Call



Dice Group with Total using Lightweight Observer

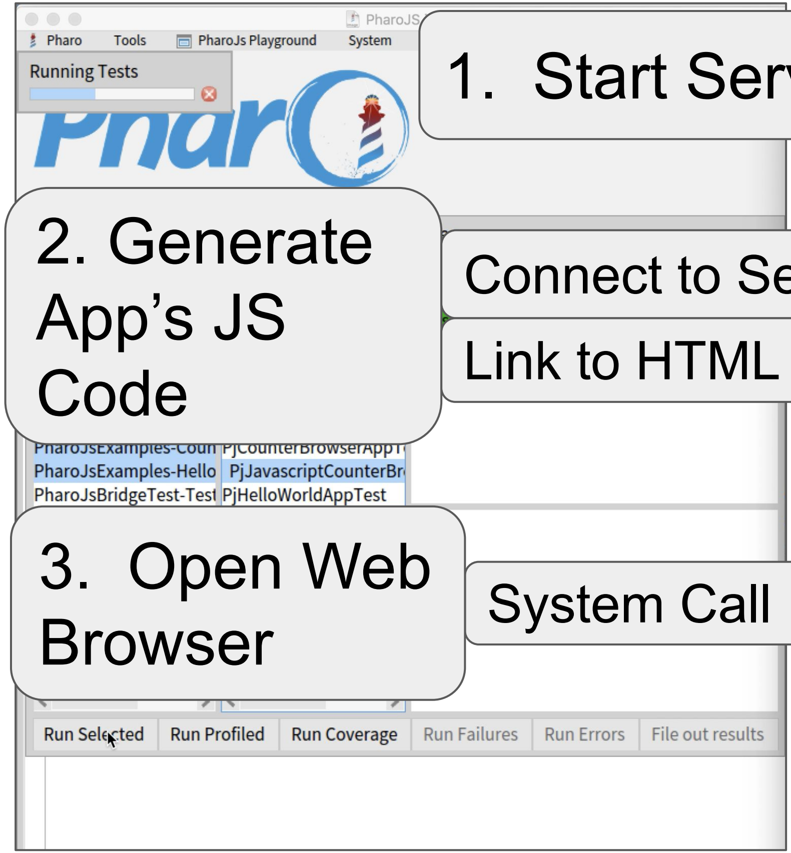
1	4	6	5	4	2

+ -

22

Roll Dice

PharoJS Tests Talk to Web Browsers



1. Start Server

2. Generate App's JS Code

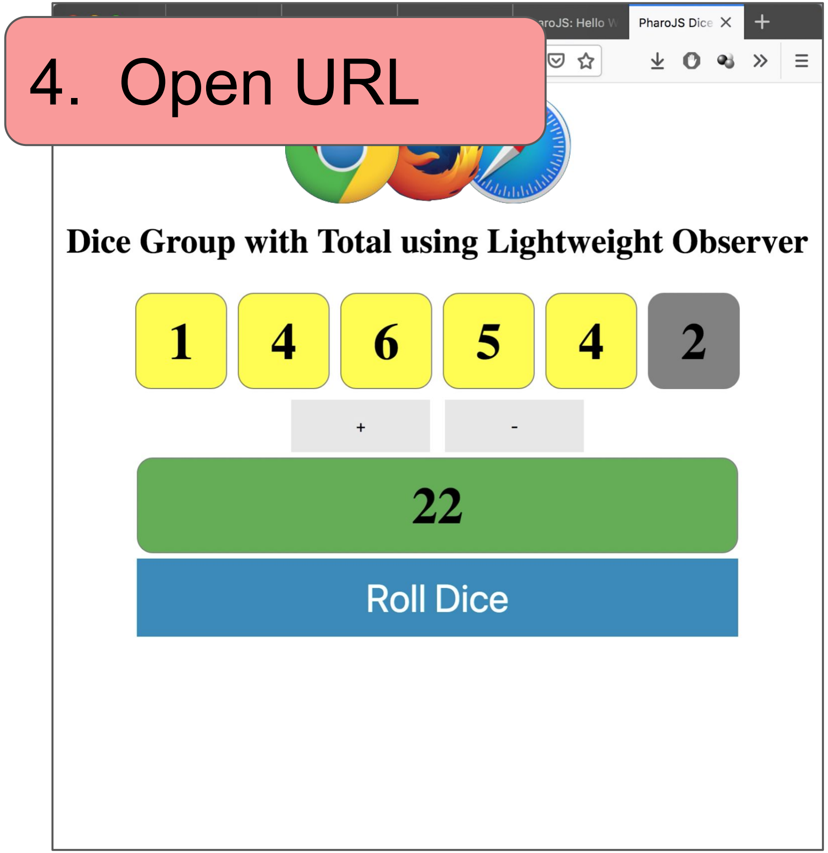
Connect to Server

Link to HTML

3. Open Web Browser

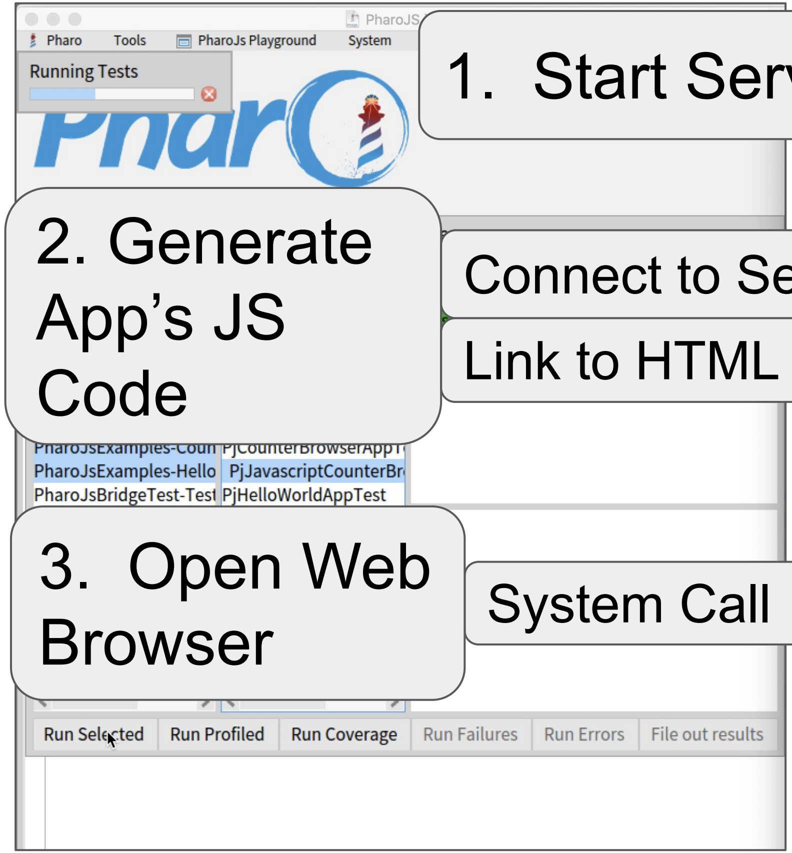
System Call

4. Open URL



Dice Group with Total using Lightweight Observer

Tests Talk to Web Browsers



1. Start Server

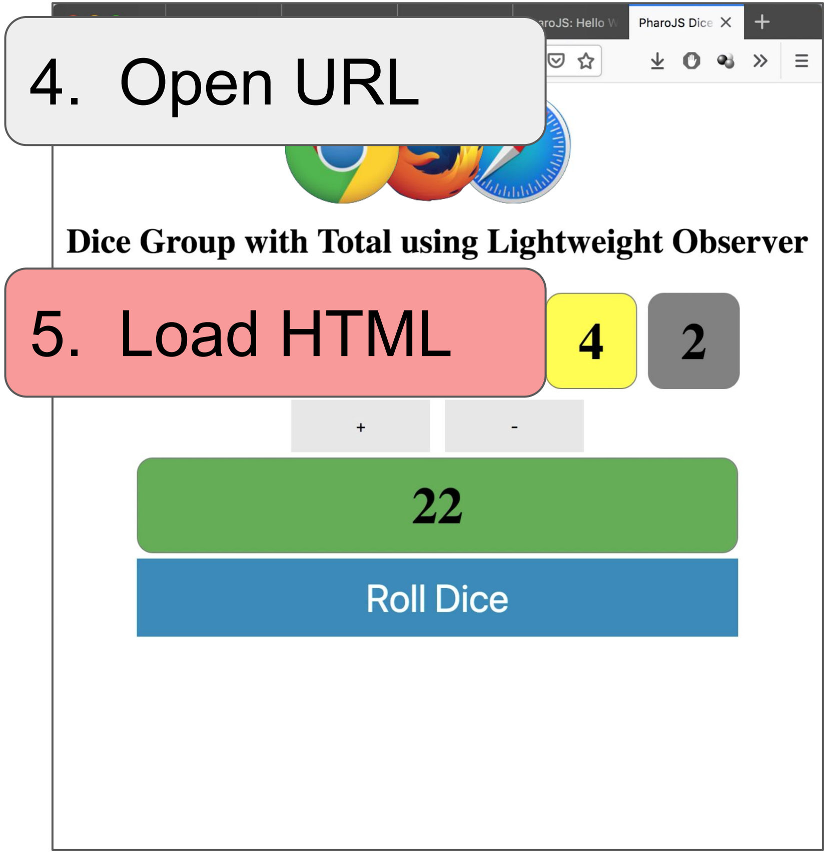
2. Generate App's JS Code

Connect to Server

Link to HTML

3. Open Web Browser

System Call

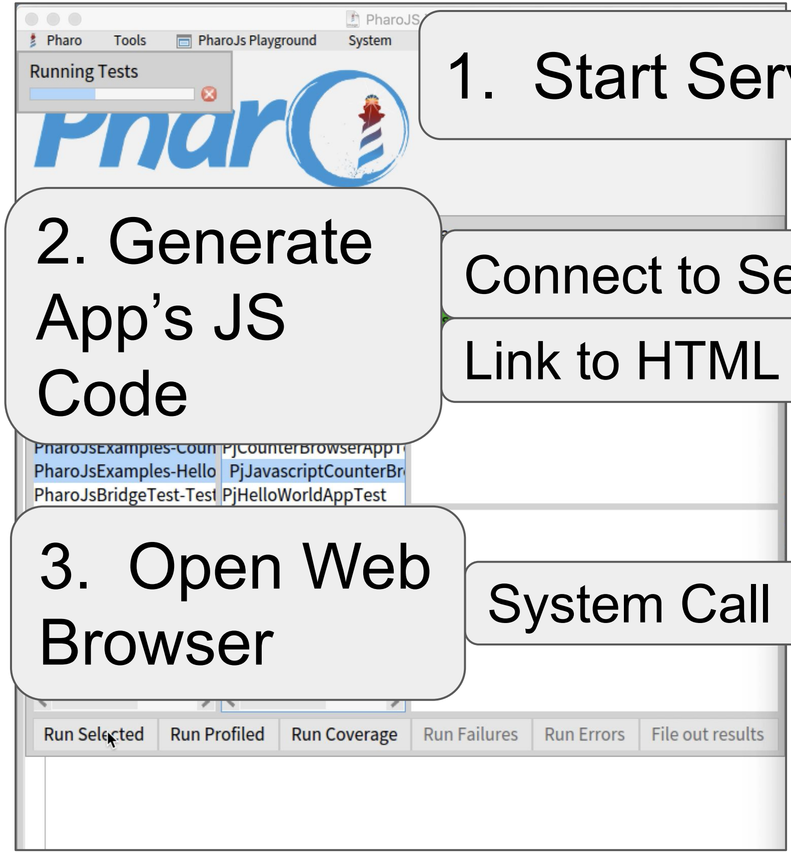


Dice Group with Total using Lightweight Observer

4. Open URL

5. Load HTML

PharoJS Tests Talk to Web Browsers



1. Start Server

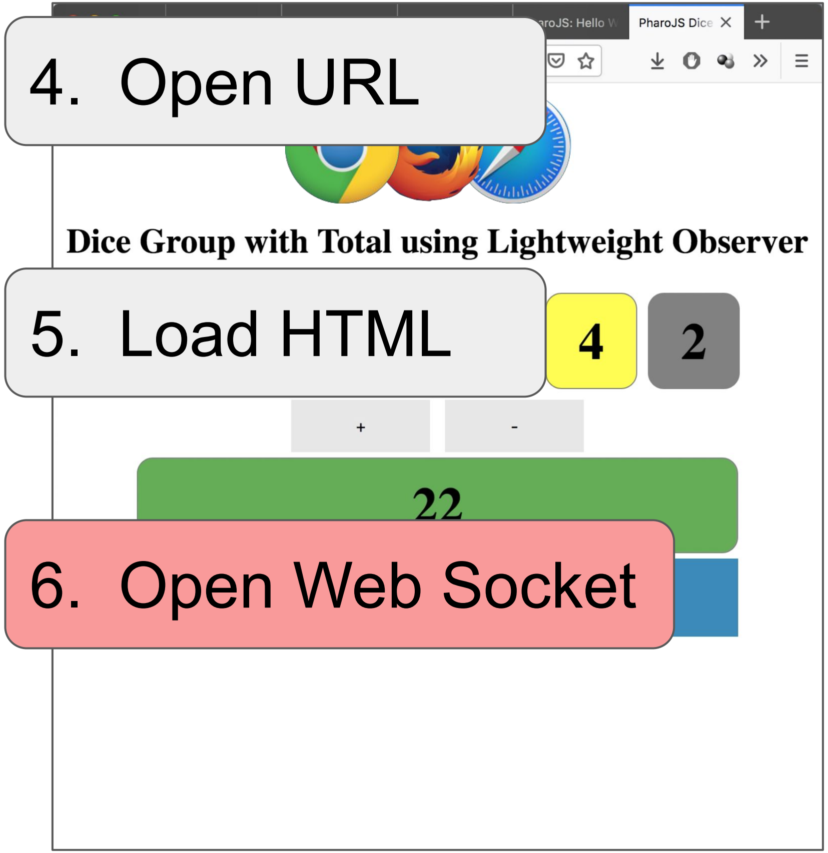
2. Generate App's JS Code

Connect to Server

Link to HTML

3. Open Web Browser

System Call

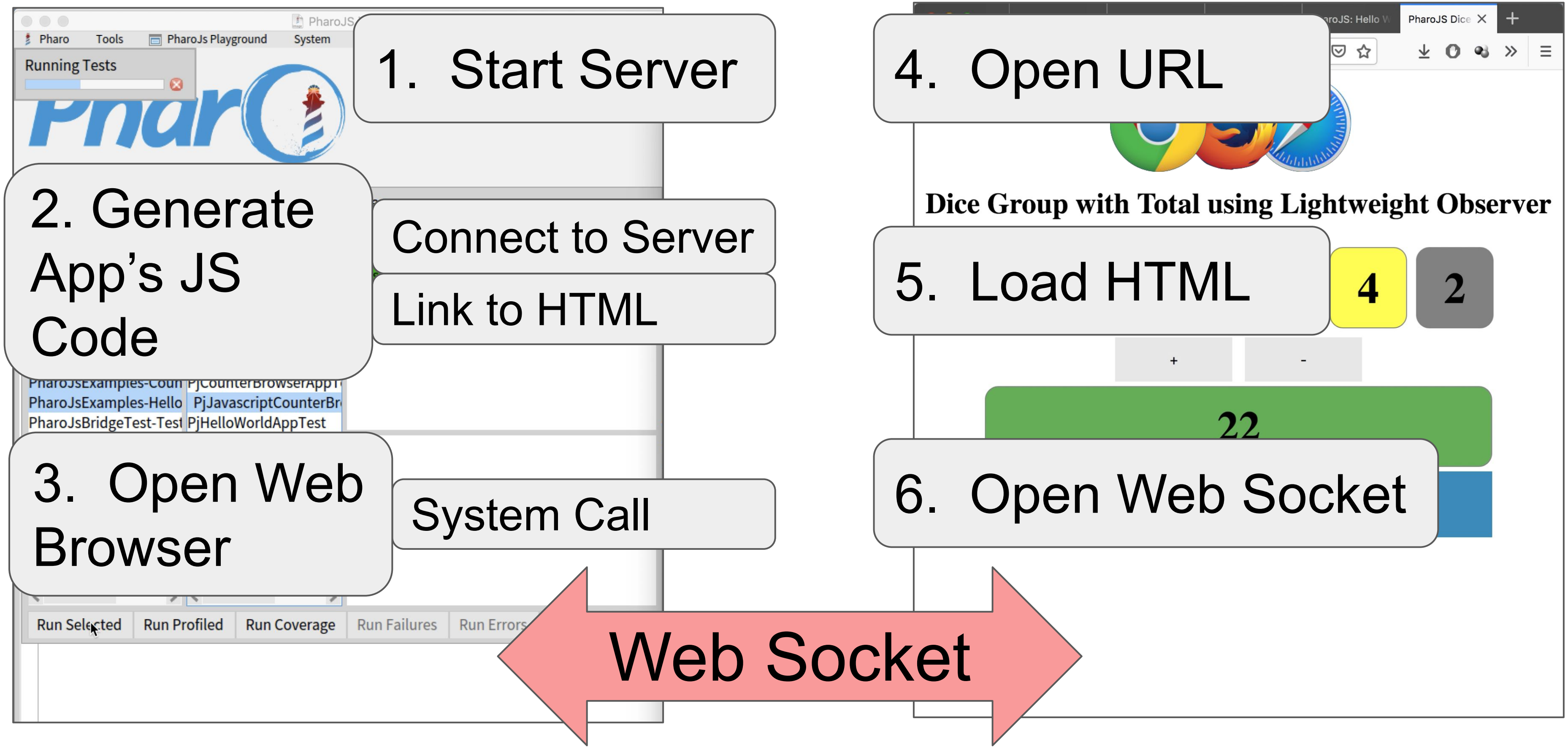


4. Open URL

5. Load HTML

6. Open Web Socket

Tests Talk to Web Browsers



1. Start Server

2. Generate App's JS Code

Connect to Server

Link to HTML

3. Open Web Browser

System Call

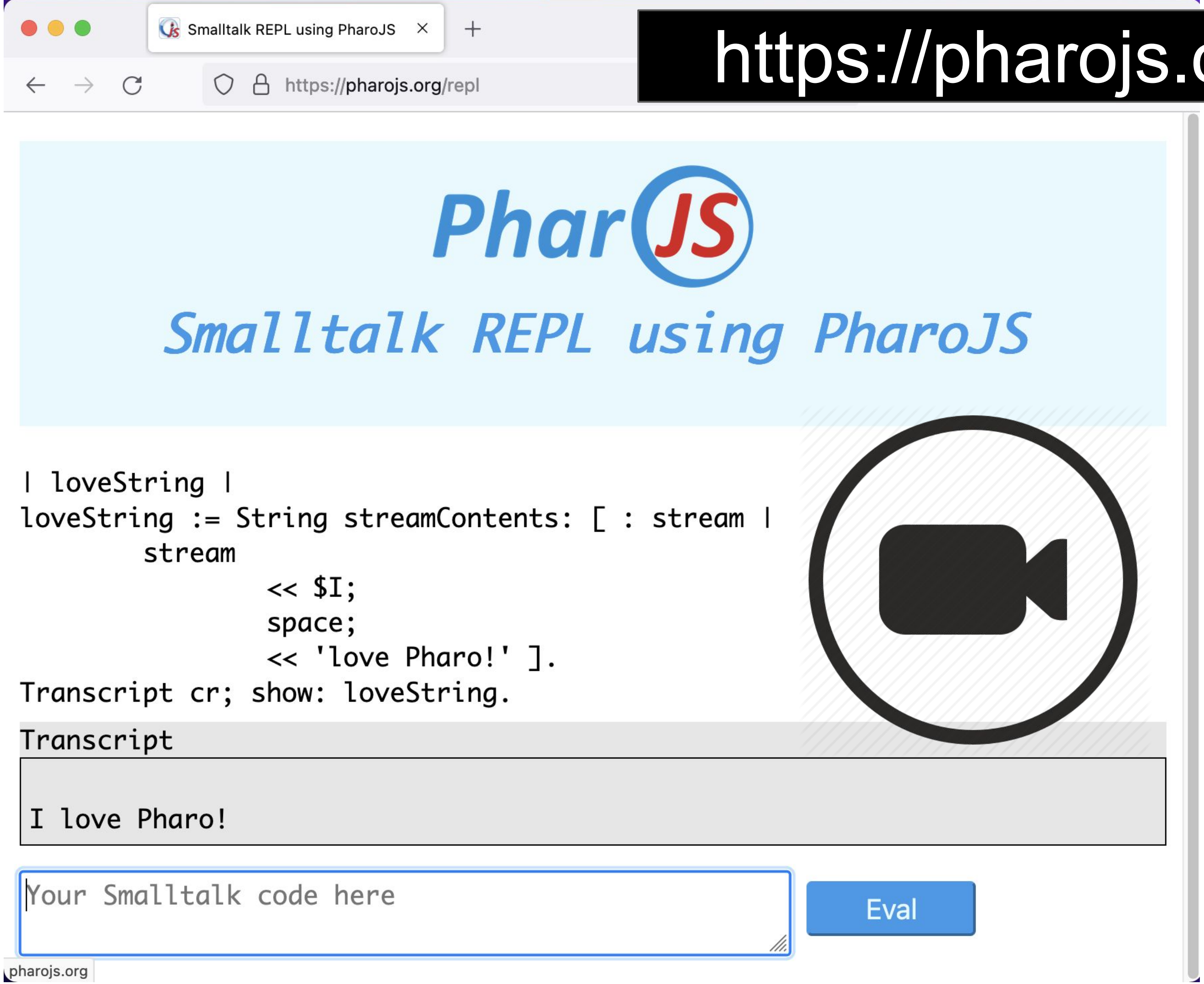
4. Open URL

5. Load HTML

6. Open Web Socket

Web Socket

https://pharojs.org/repl



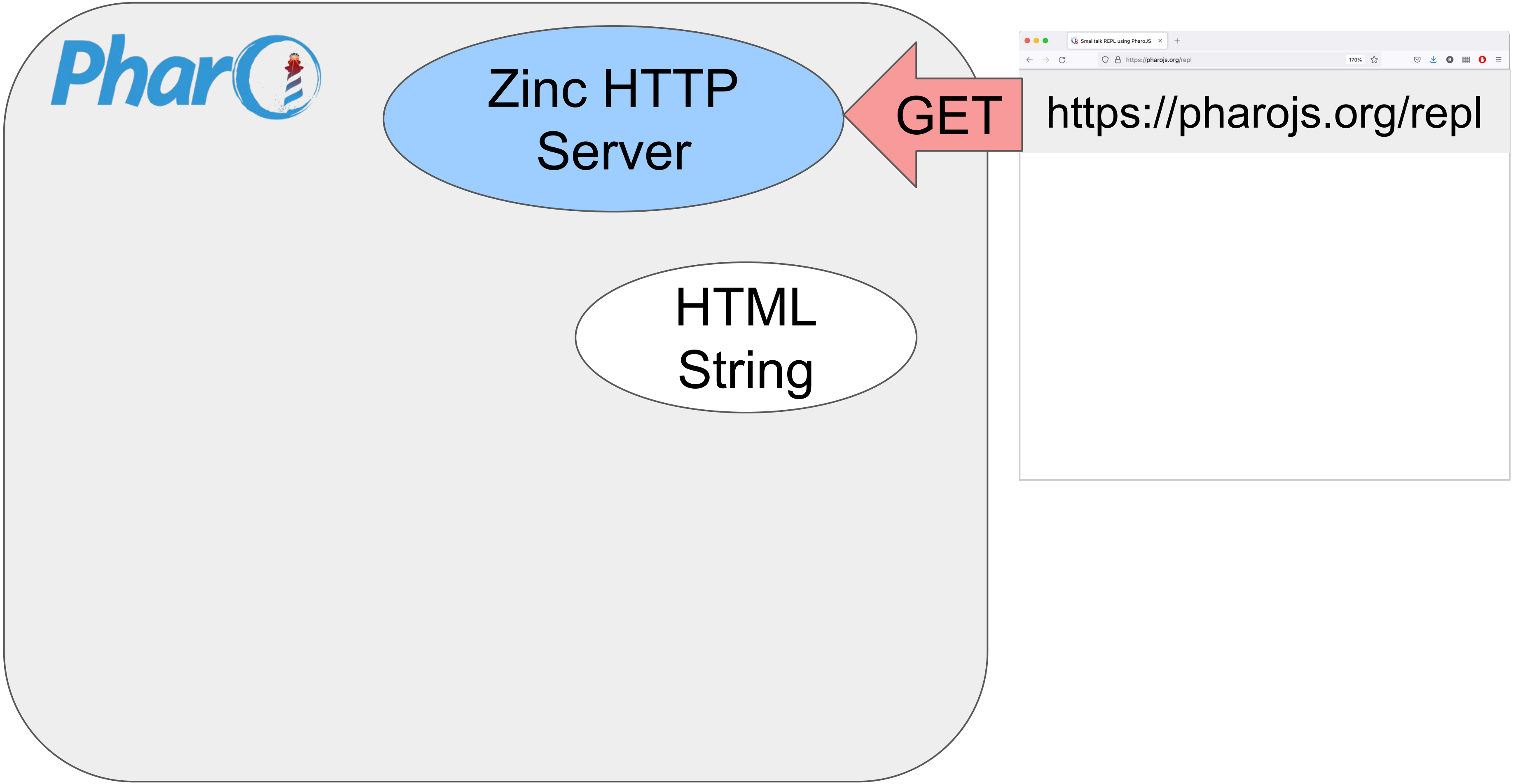
```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

Transcript

```

I love Pharo!
```

HTML is a String in the image



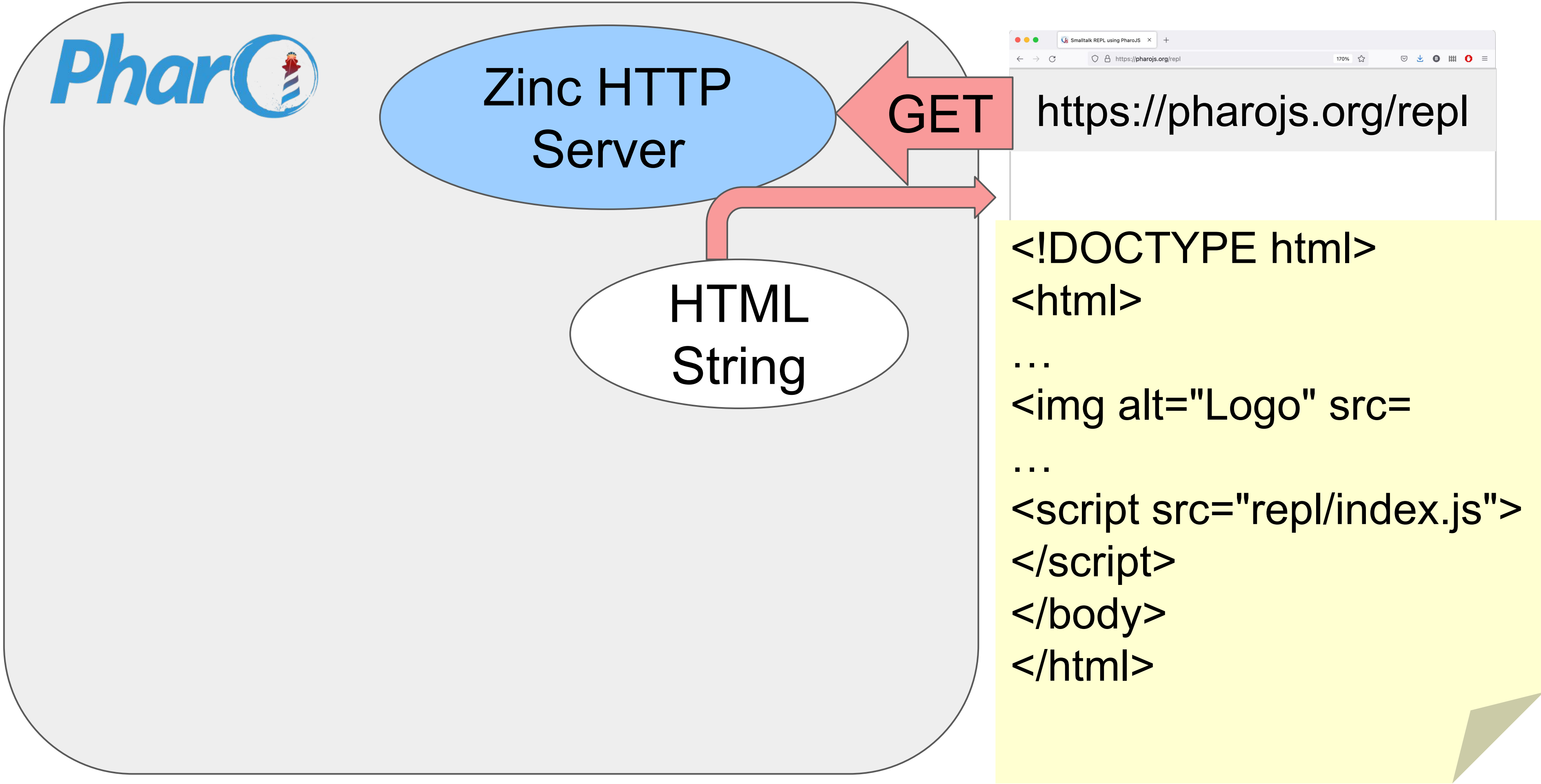
Zinc HTTP Server

HTML String

GET

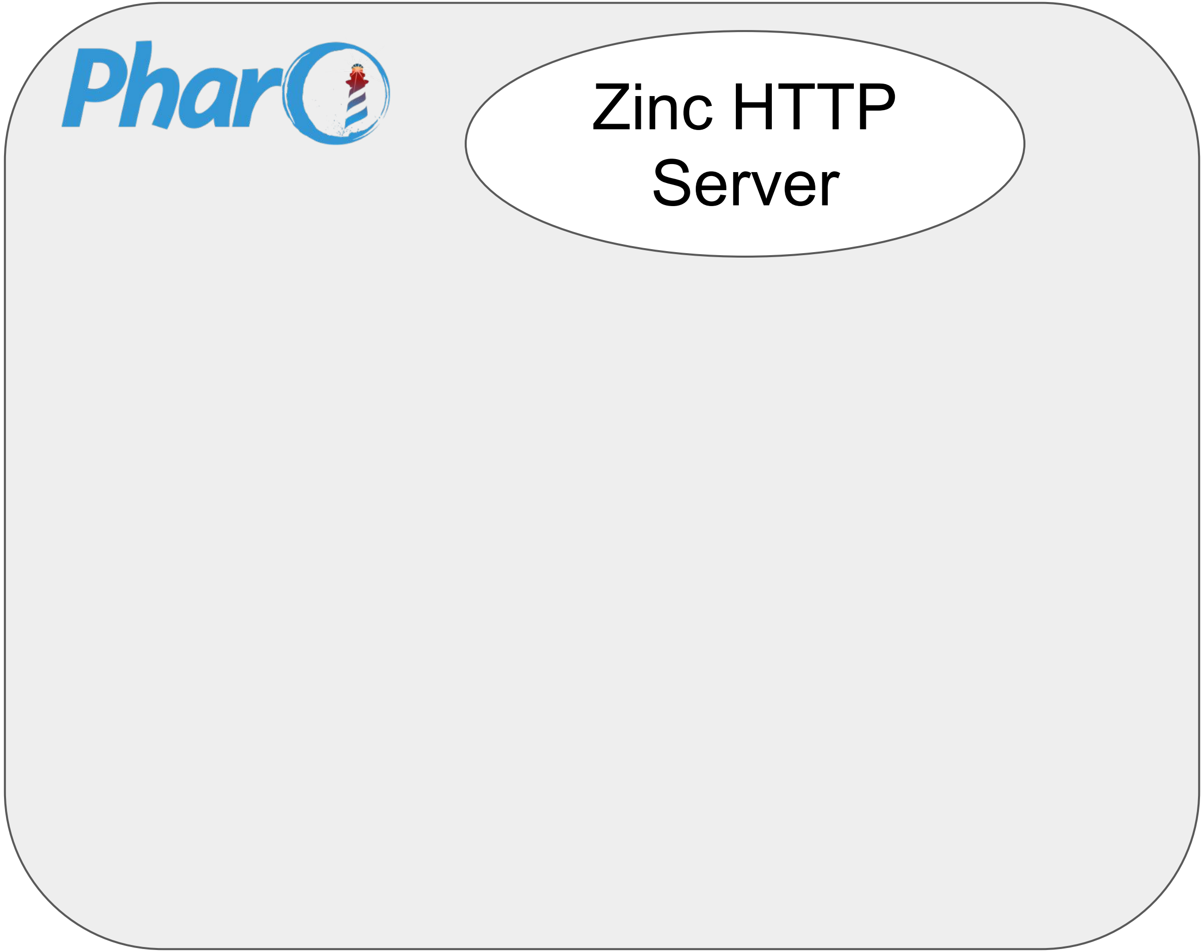
https://pharojs.org/repl

HTML is a String in the image



```
<!DOCTYPE html>
<html>
…
<img alt="Logo" src=
…
<script src="repl/index.js">
</script>
</body>
</html>
```

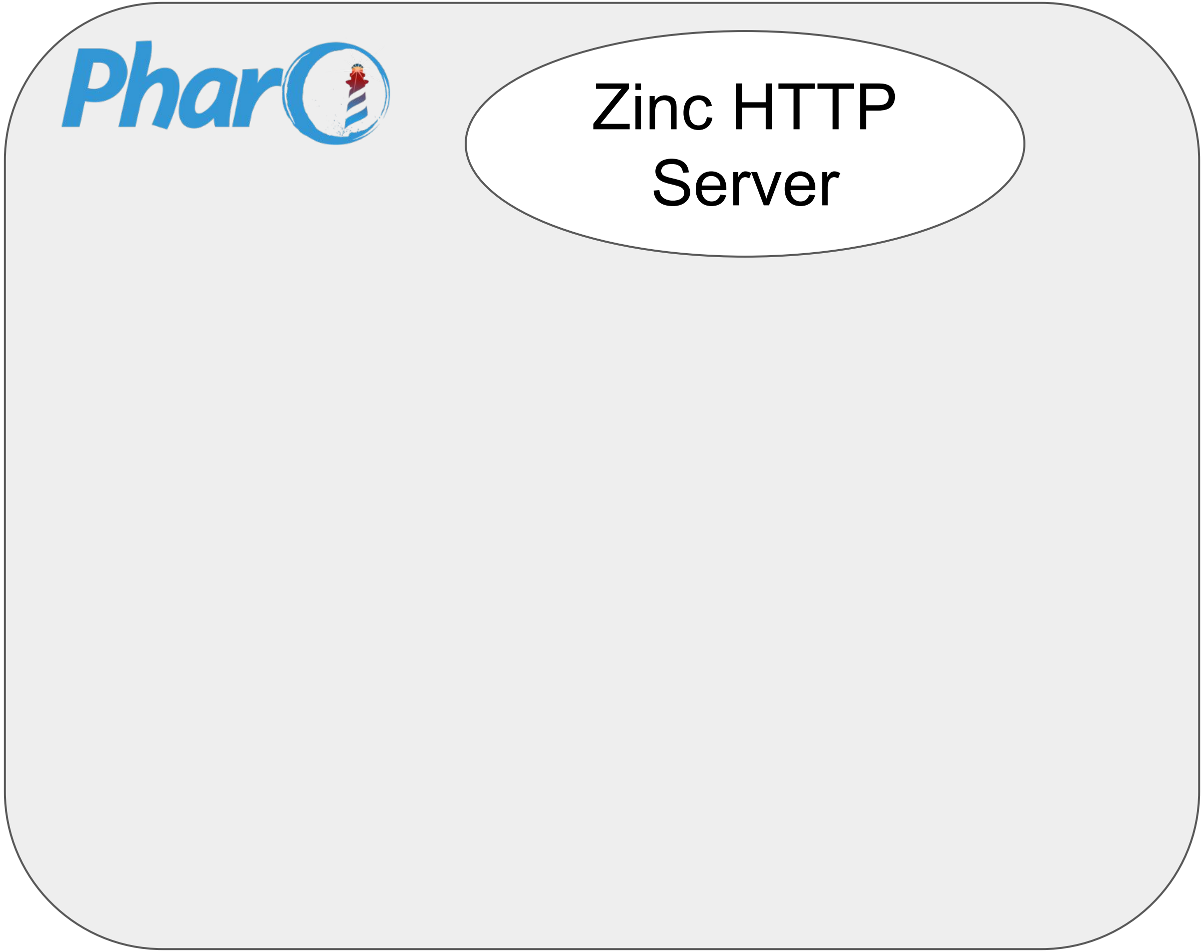
Browser Processes the HTML



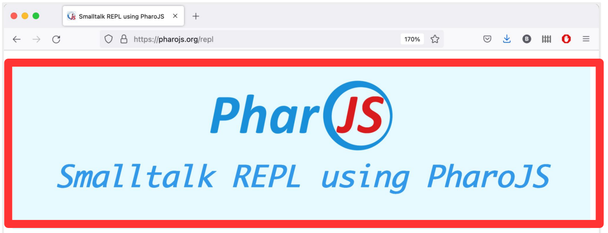
https://pharojs.org/repl

```
<!DOCTYPE html>
<html>
...
<img alt="Logo" src=...
...
<script src="repl/index.js">
</script>
</body>
</html>
```

Browser Loads Resources



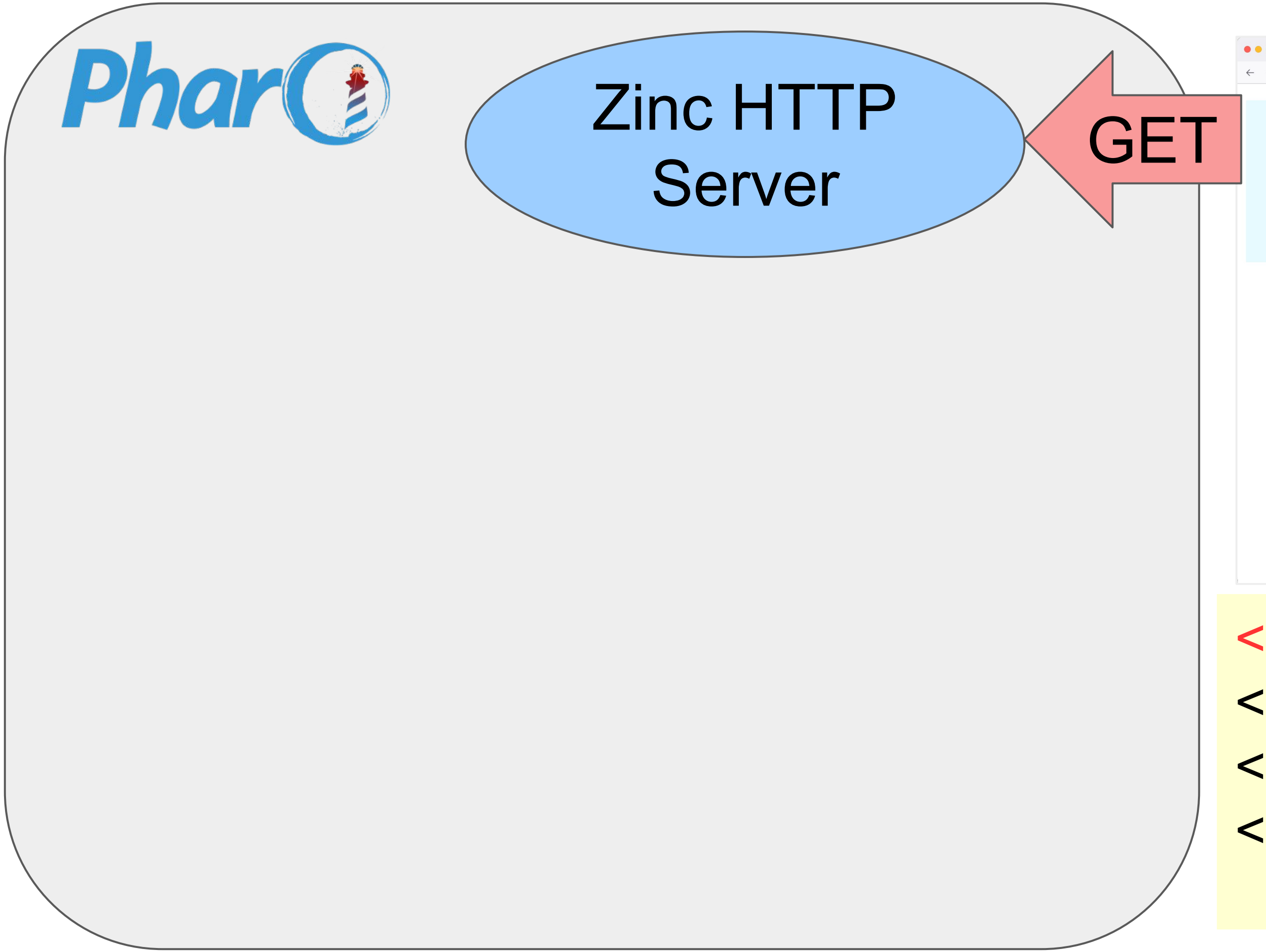
Zinc HTTP Server

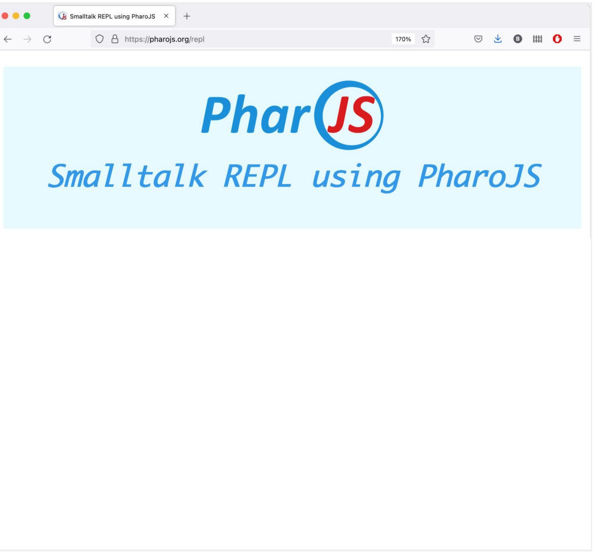


Smalltalk REPL using PharoJS

```
<!DOCTYPE html>
<html>
…
<img alt="Logo" src=
…
<script src="repl/index.js">
</script>
</body>
</html>
```

Browser Requests JavaScript Code

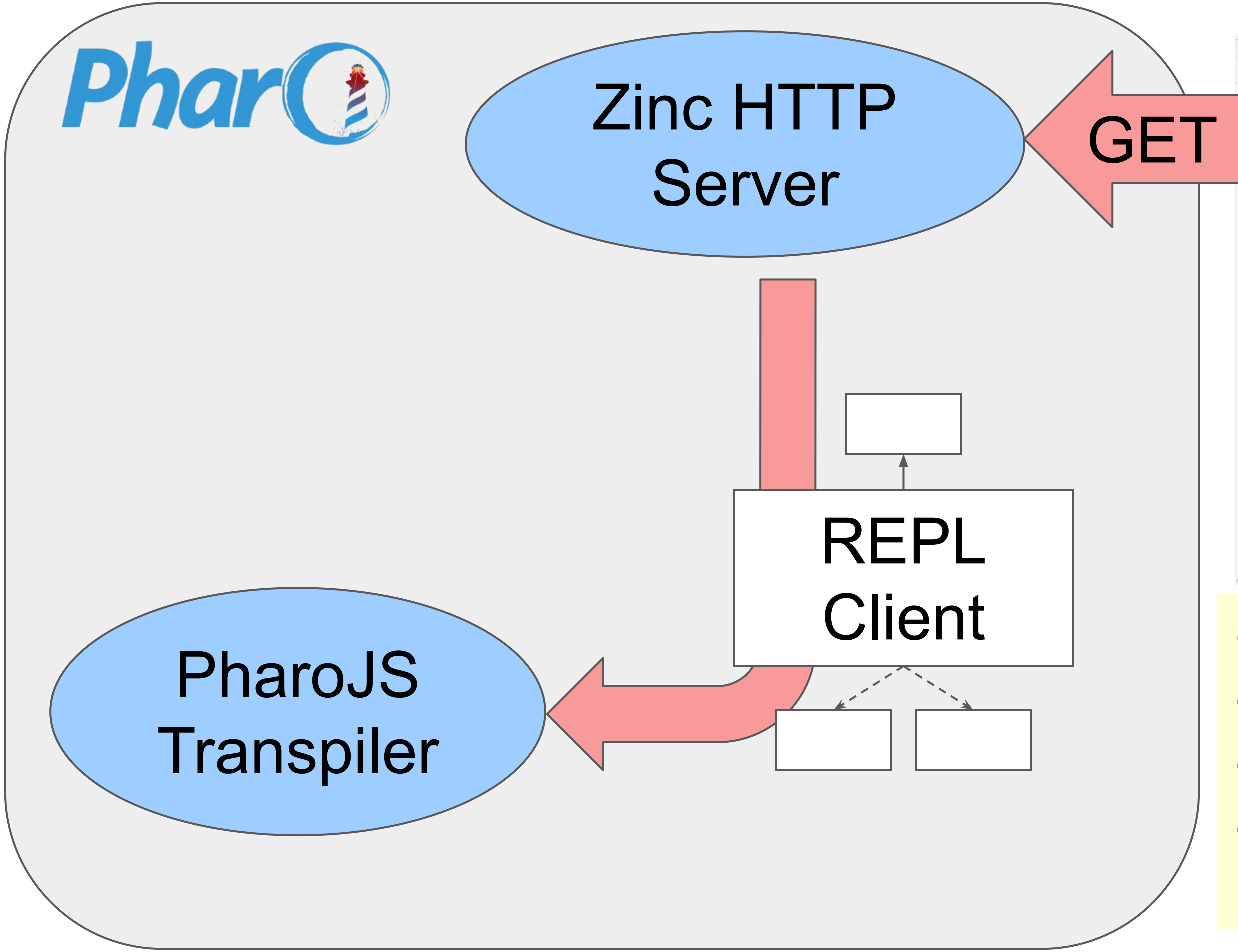


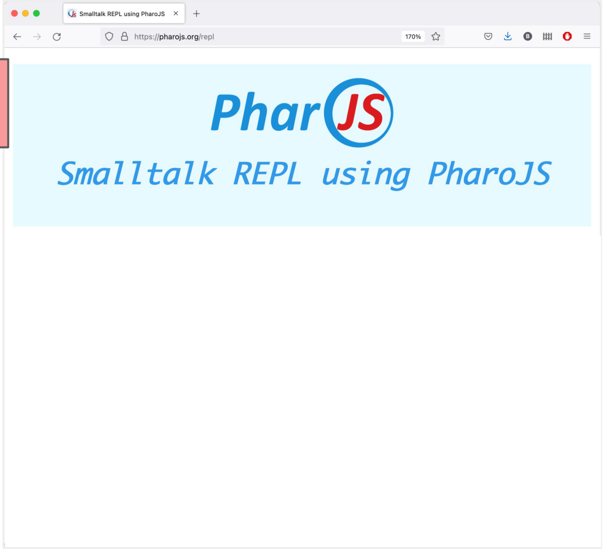


Smalltalk REPL using PharoJS

```
<script src="repl/index.js">
</script>
</body>
</html>
```

REPL Client JS Code is Generated

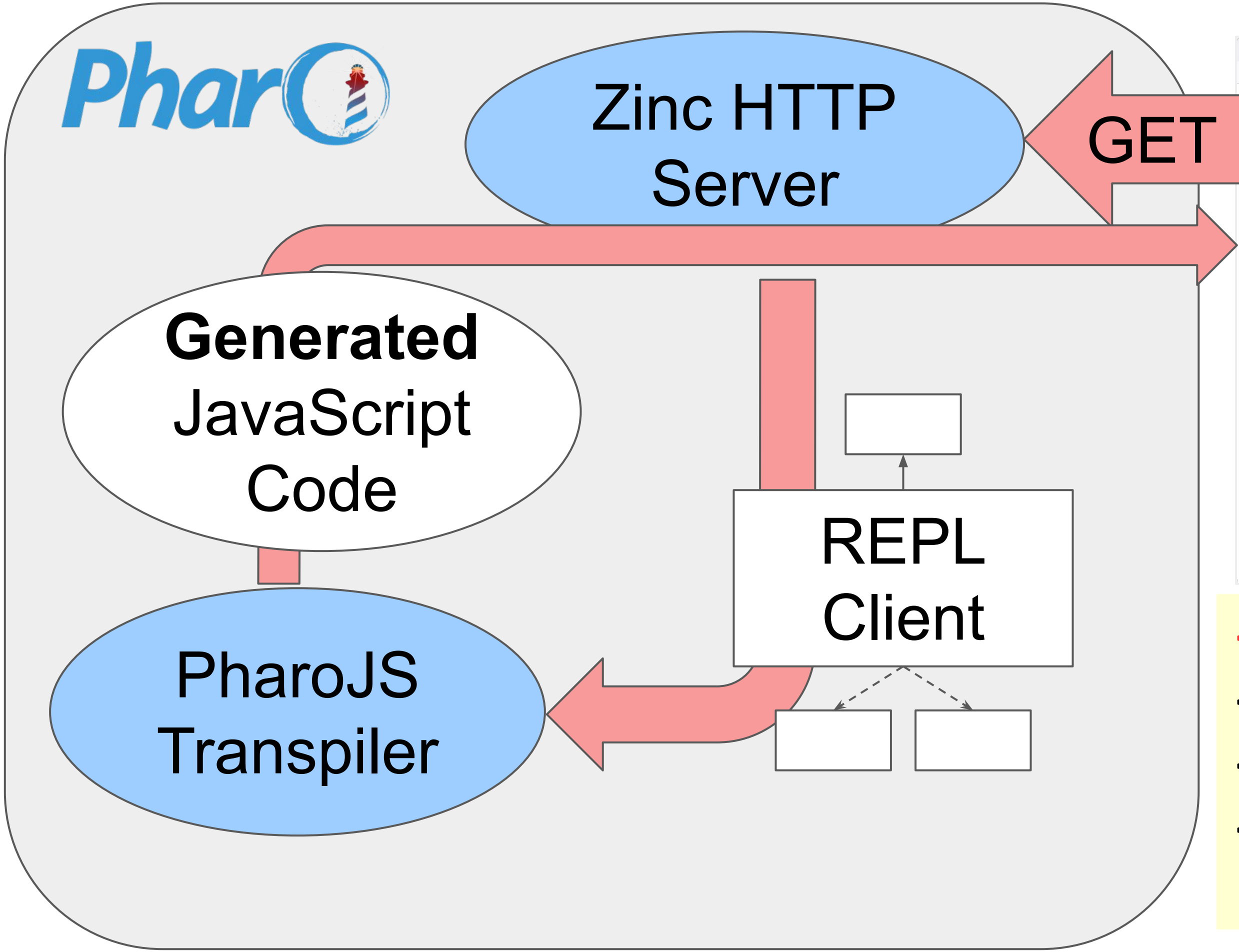


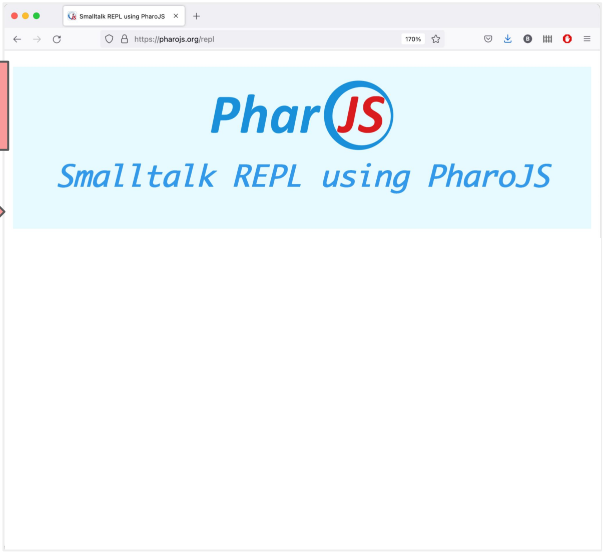


Smalltalk REPL using PharoJS

```
<script src="repl/index.js">
</script>
</body>
</html>
```

REPL Client JS Code is Generated




Smalltalk REPL using PharoJS

https://pharojs.org/repl

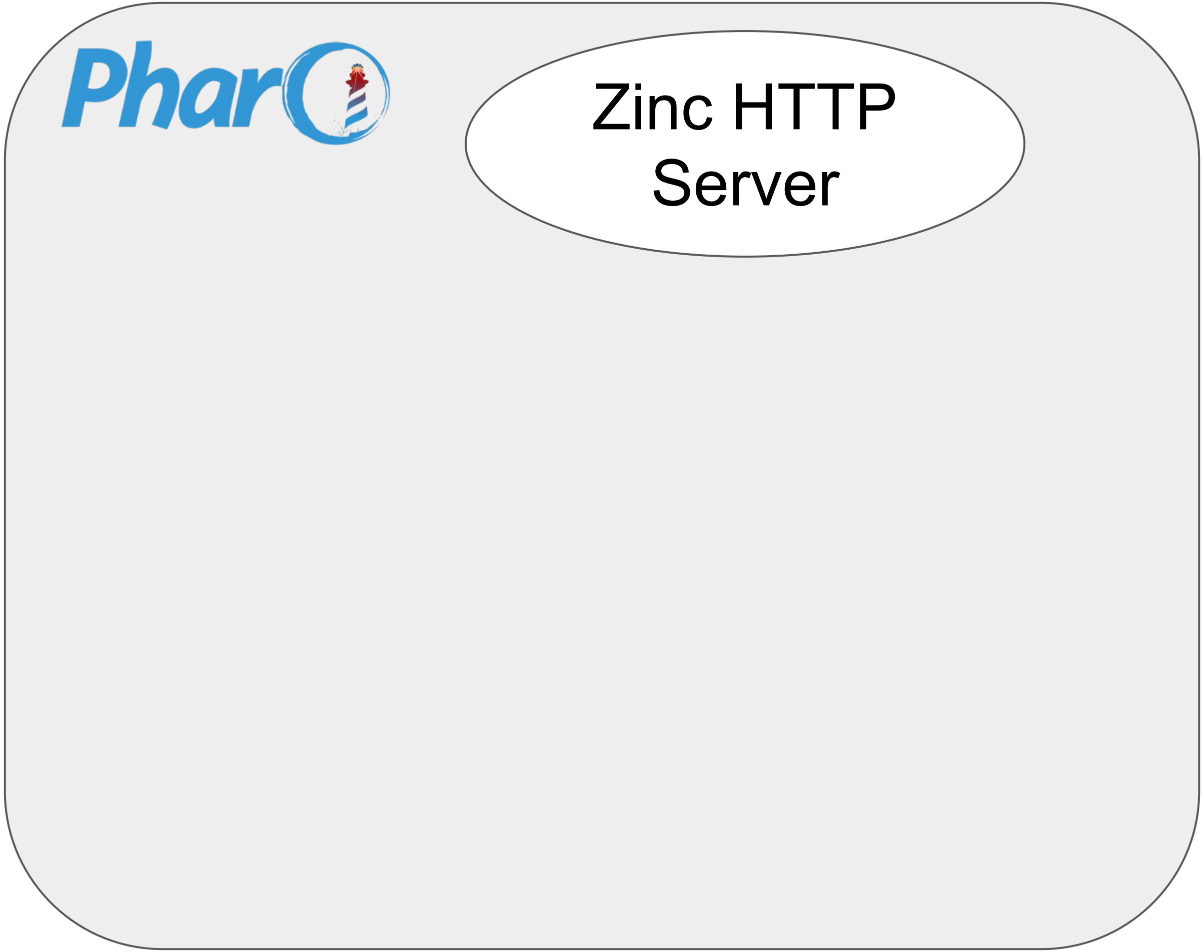
170%

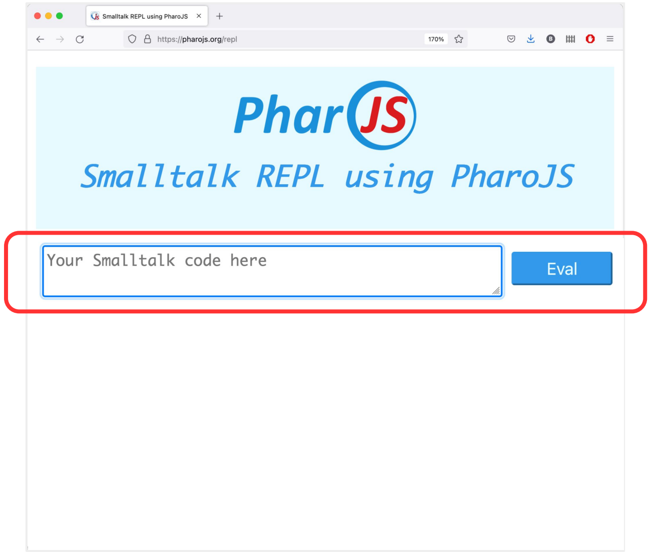
PharoJS

Smalltalk REPL using PharoJS

```
<script src="repl/index.js">
</script>
</body>
</html>
```

Client Creates and Links DOM Elements



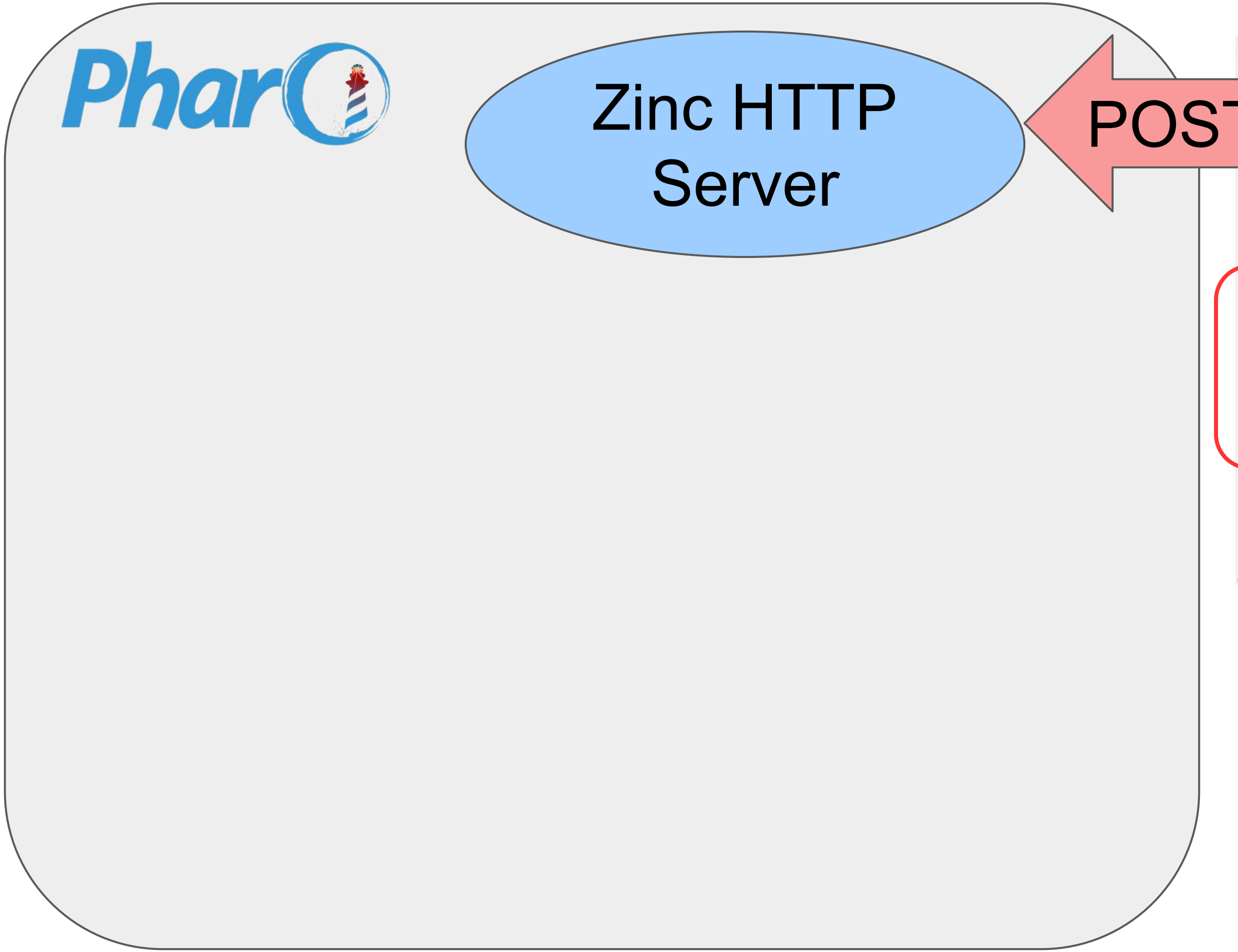


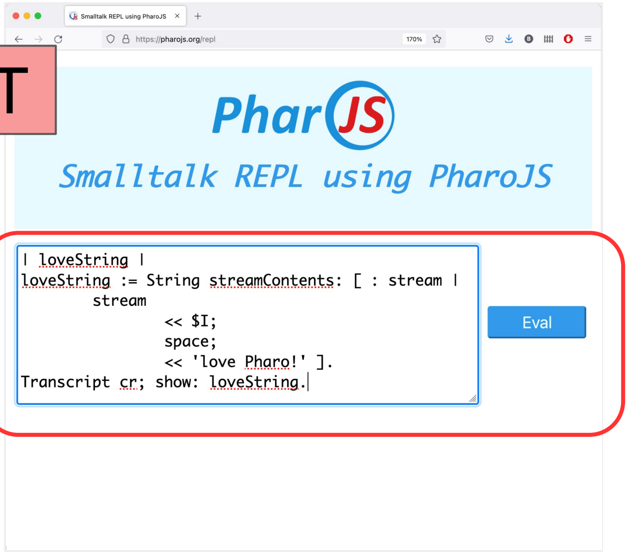
Smalltalk REPL using PharoJS

Your Smalltalk code here

```
<script src="repl/index.js">
</cript>
</body>
</html>
```

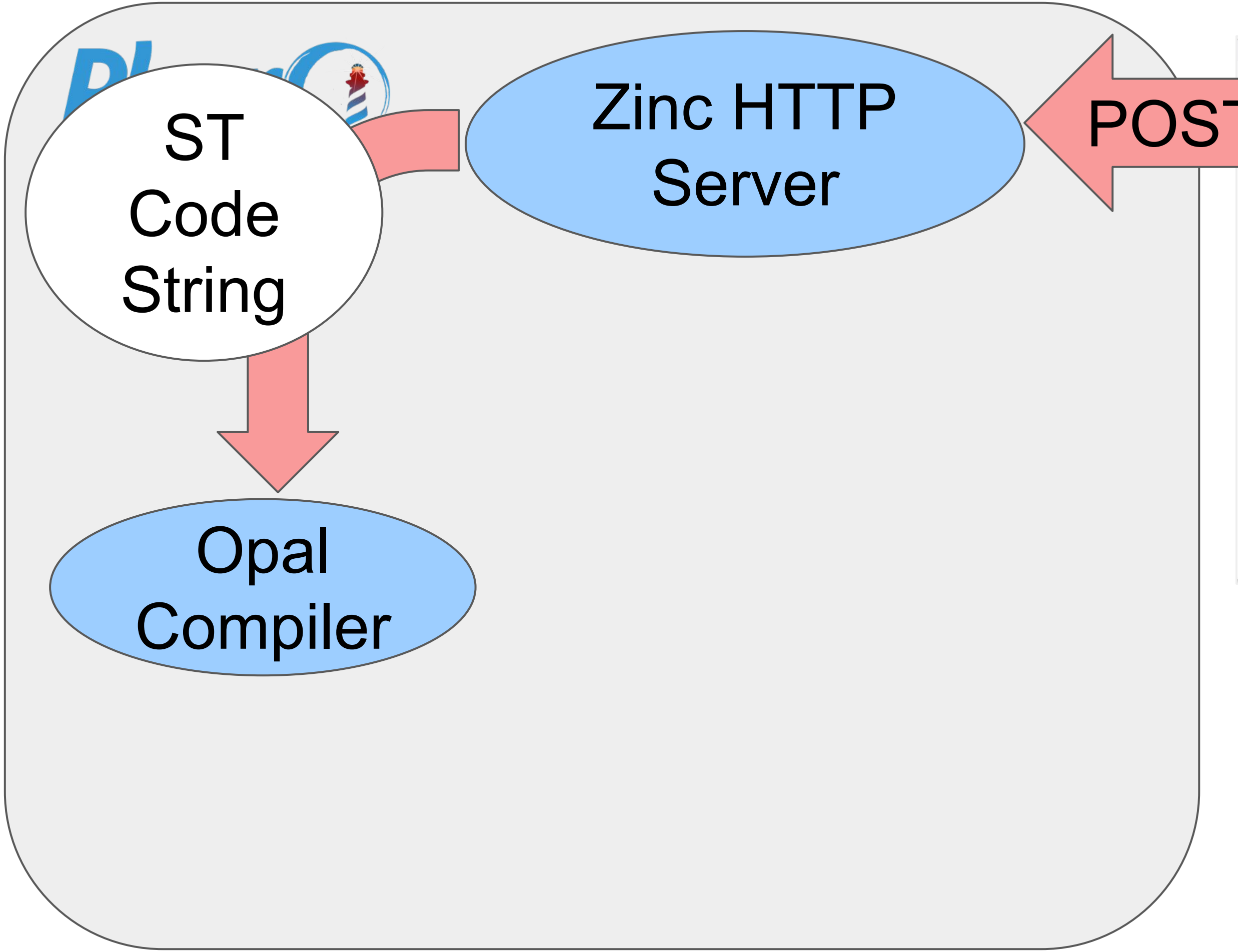
Client Sends ST Code Snippet

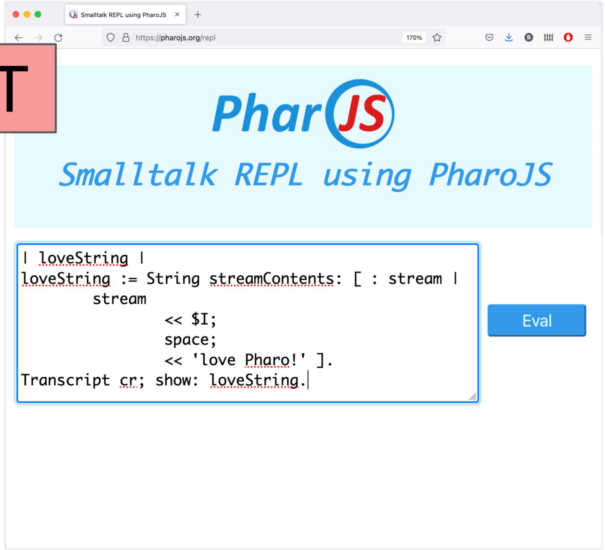




```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

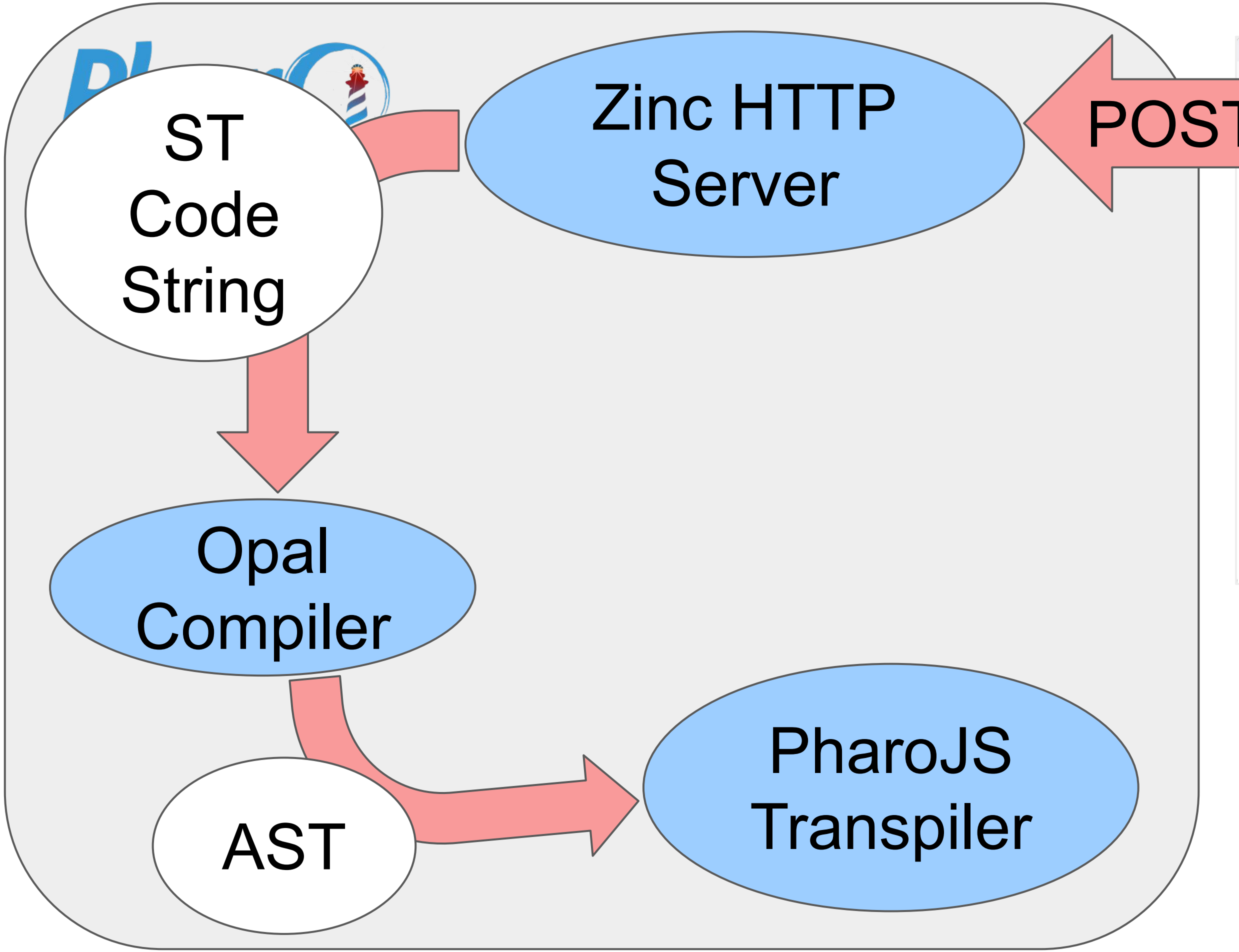
Server Compiles ST Code Snippet

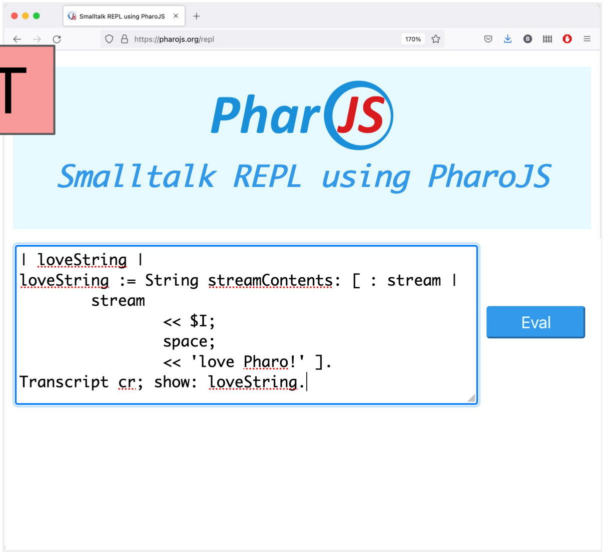




```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

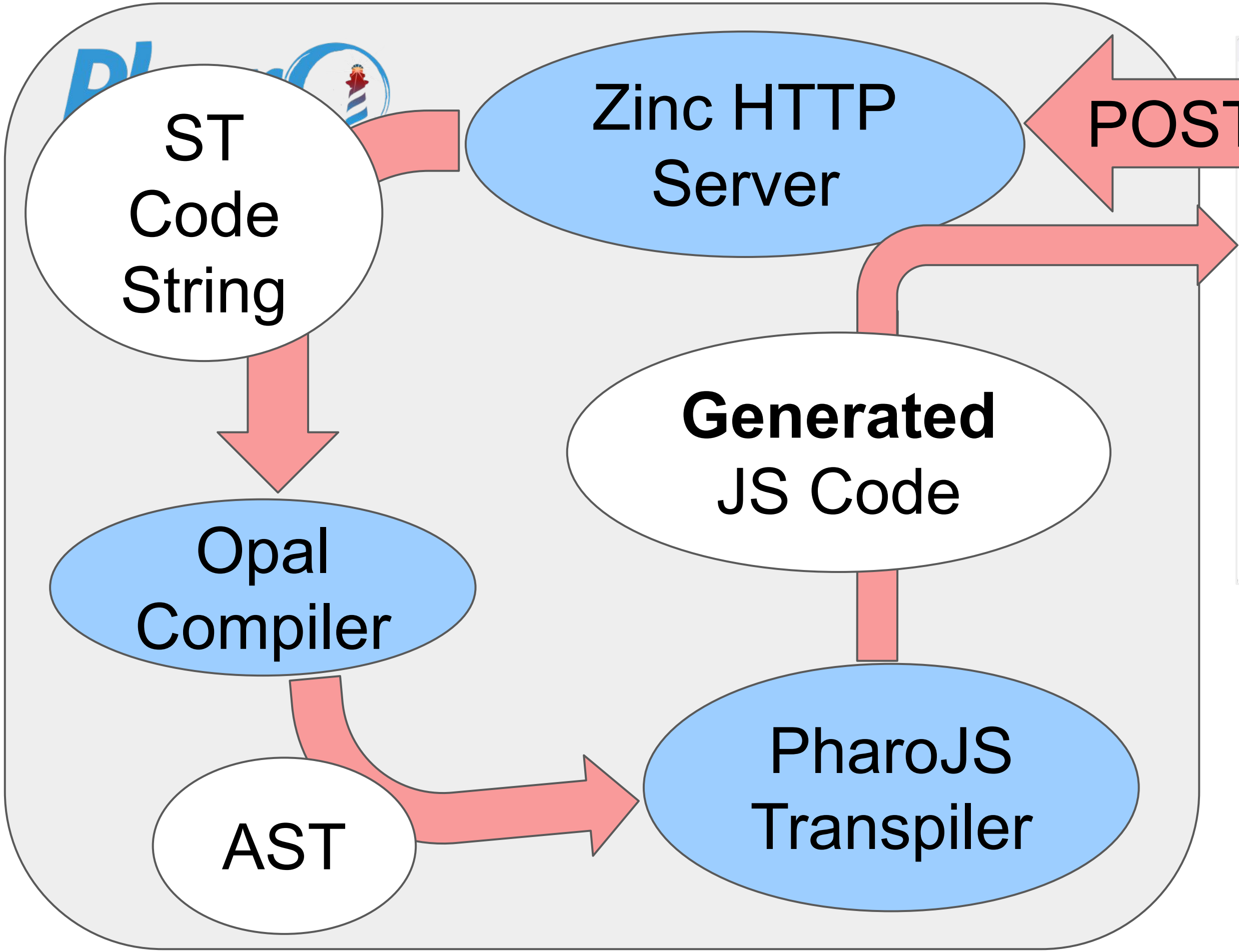
PharoJS Transpiles AST

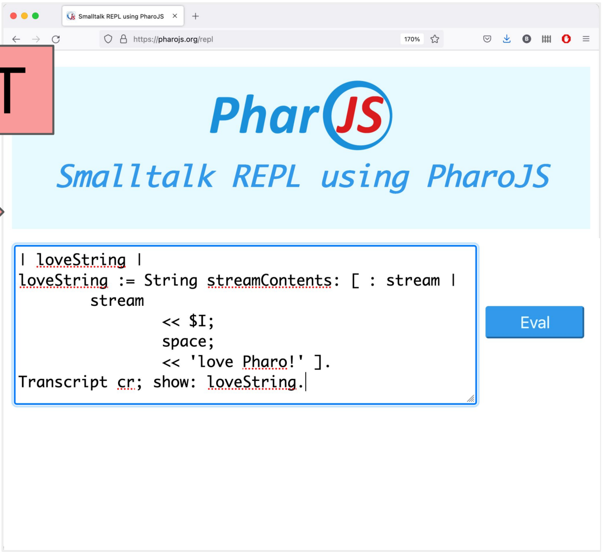




```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

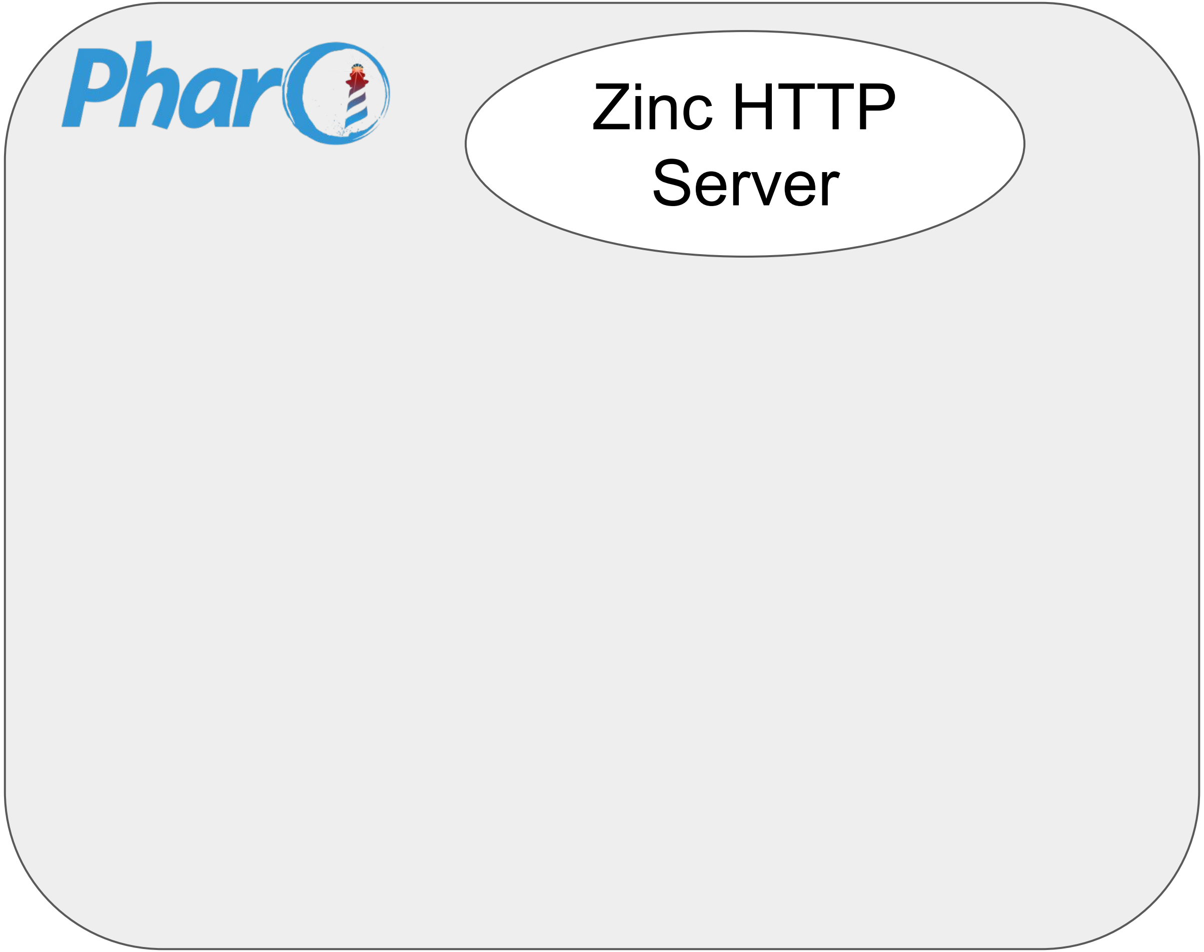
Server Sends Generated JS



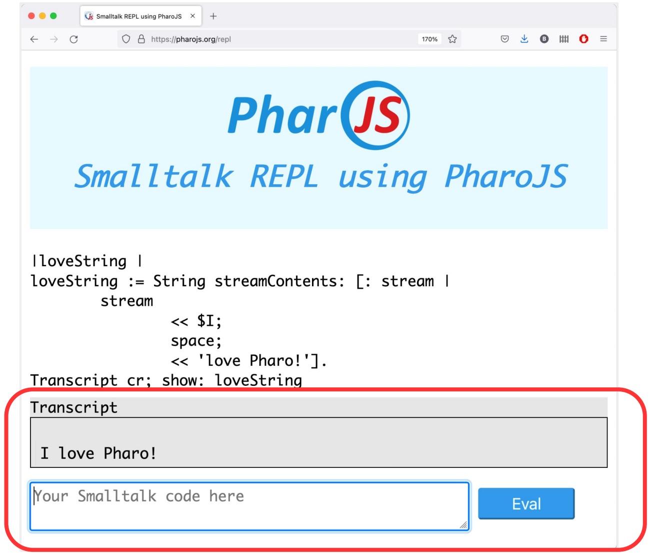


```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

Client Executes Generated JS Code



Zinc HTTP Server



PharJS

Smalltalk REPL using PharoJS

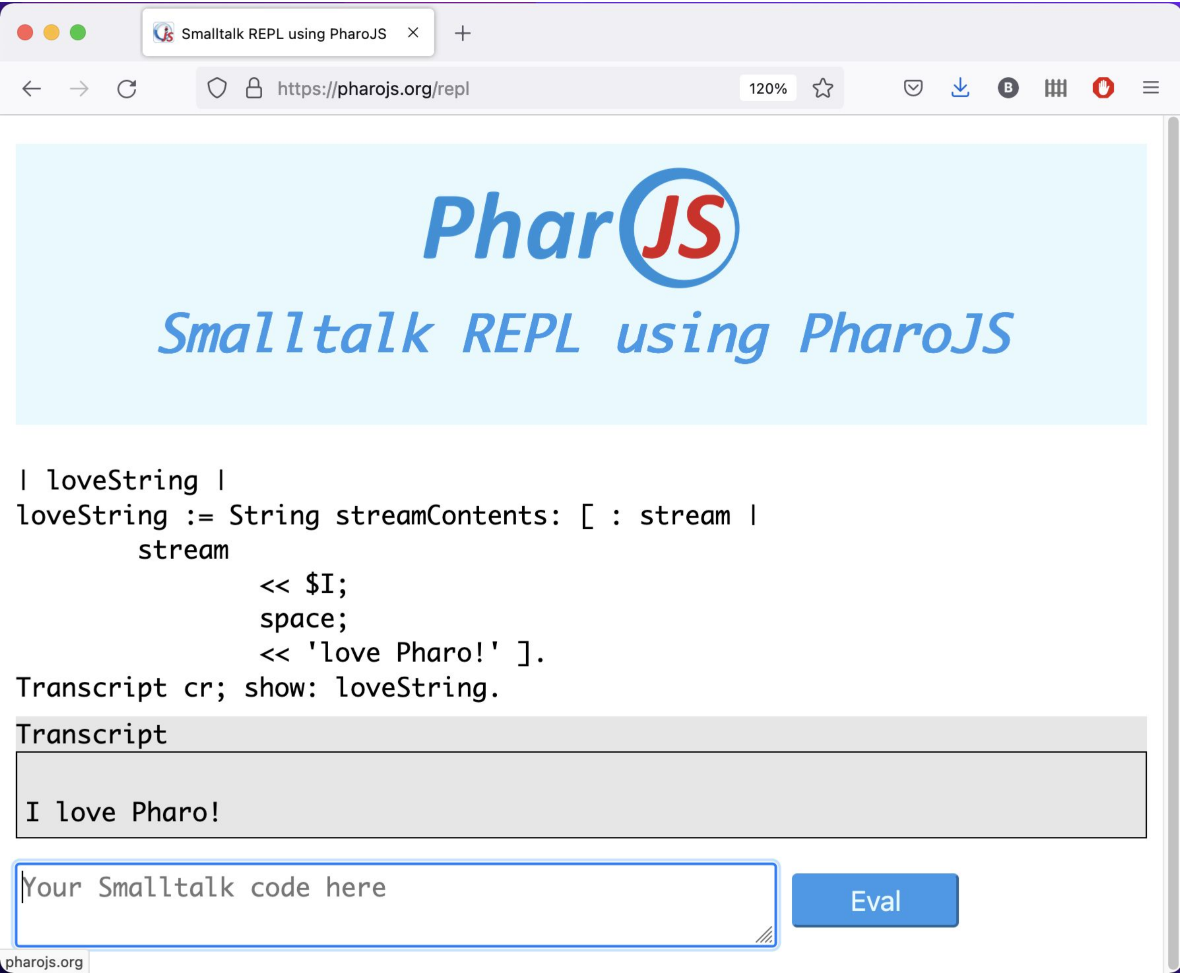
```
|loveString |
loveString := String streamContents: [: stream |
        stream
                << $I;
                space;
                << 'love Pharo!'].
Transcript cr; show: loveString
```

Transcript

```

 I love Pharo!
```

Small App



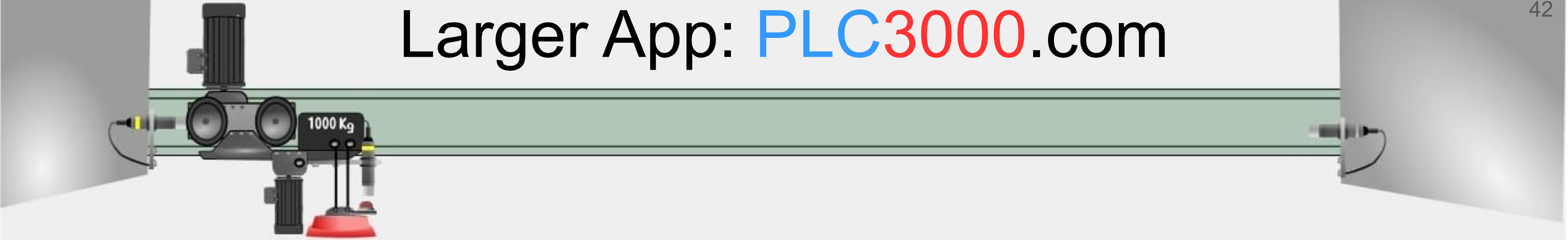
```
| loveString |
loveString := String streamContents: [ : stream |
        stream
                << $I;
                space;
                << 'love Pharo!' ].
Transcript cr; show: loveString.
```

```
I love Pharo!
```

- Client+Server
  - 10 classes
  - 64 methods
- Tests
  - 1 class
  - 20 methods

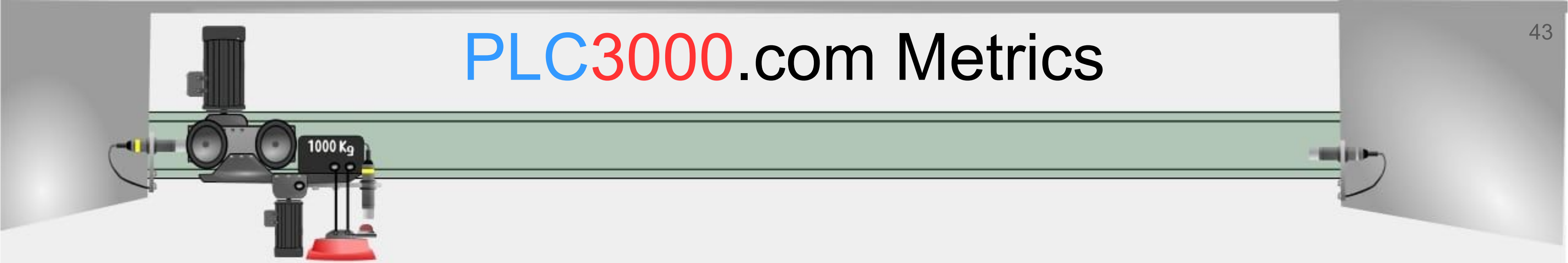
Generated JS
267KB (+3KB)

Larger App: PLC3000.com



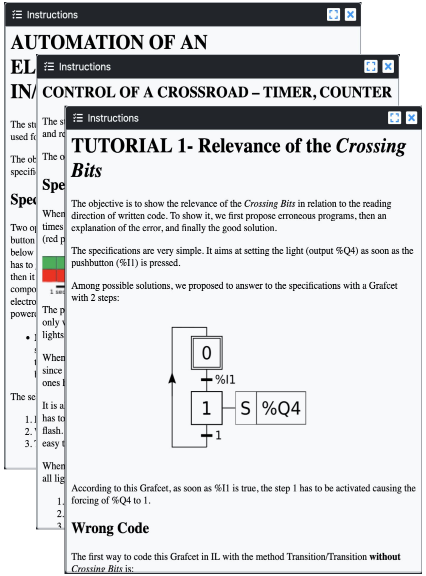
Teaching
PLC Programming & Factory Automation

PLC3000.com Metrics



- Client+Server
  - 342 classes
  - 2529 methods
- Tests
  - 108 classes
  - 1184 methods
  - 876 test runs

PLC3000.com = Educational Software + Contents



Instructions

AUTOMATION OF AN EL… IN…

Instructions

CONTROL OF A CROSSROAD – TIMER, COUNTER

Instructions

TUTORIAL 1- Relevance of the *Crossing Bits*

The objective is to show the relevance of the *Crossing Bits* in relation to the reading direction of written code. To show it, we first propose erroneous programs, then an explanation of the error, and finally the good solution.

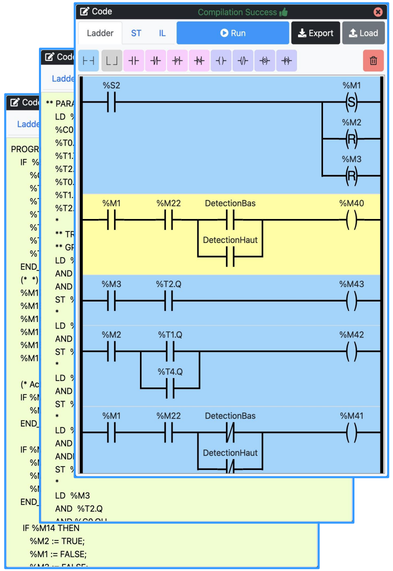
The specifications are very simple. It aims at setting the light (output %Q4) as soon as the pushbutton (%I1) is pressed.

Among possible solutions, we proposed to answer to the specifications with a Grafcet with 2 steps:

According to this Grafcet, as soon as %I1 is true, the step 1 has to be activated causing the forcing of %Q4 to 1.

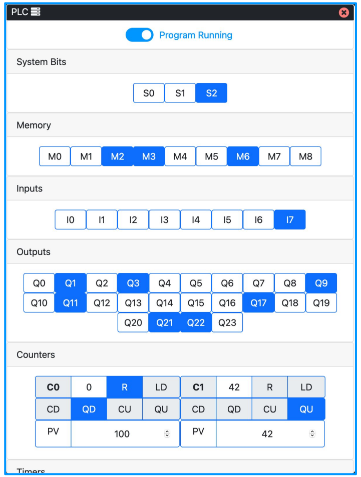
Wrong Code

The first way to code this Grafcet in IL with the method Transition/Transition **without** *Crossing Bits* is:



```
LD  %M3
AND  %T2.Q
```

```
IF %M14 THEN
  %M2 := TRUE;
  %M1 := FALSE;
```



PLC

Program Running

System Bits

S0 S1 S2

Memory

M0 M1 M2 M3 M4 M5 M6 M7 M8

Inputs

I0 I1 I2 I3 I4 I5 I6 I7

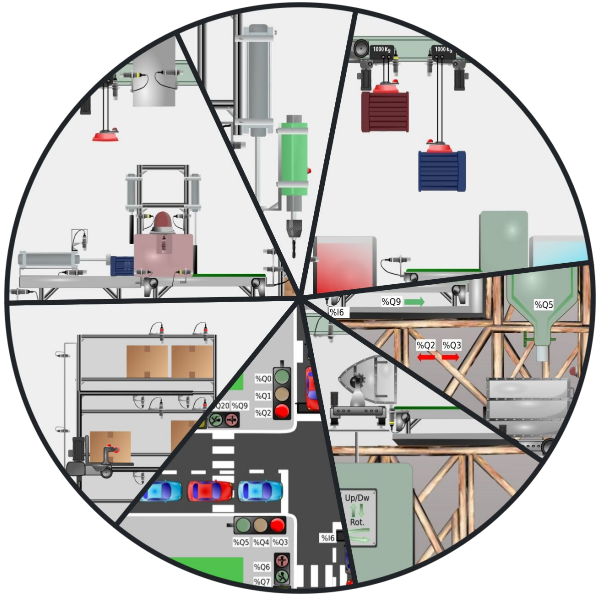
Outputs

Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9
Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19
Q20 Q21 Q22 Q23

Counters

C0	0	R	LD	C1	42	R	LD
CD	QD	CU	QU	CD	QD	CU	QU
PV	100			PV	42		

Timers



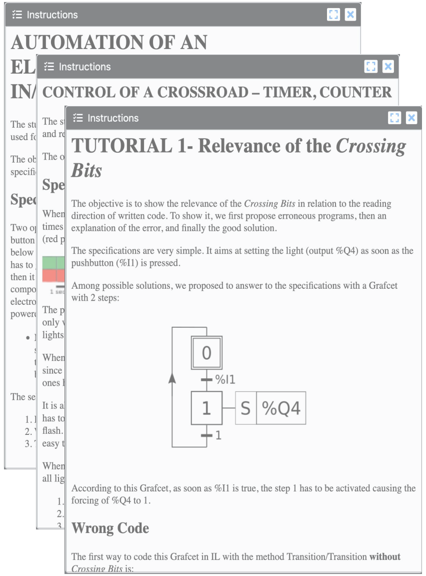
27 (13 + 14) Exercises & Tutorials

3 Programming Languages

PLC Simulator

7 (4+3) Physics Simulations

PLC3000.com = Educational Software + Contents



TUTORIAL 1- Relevance of the *Crossing Bits*

The objective is to show the relevance of the *Crossing Bits* in relation to the reading direction of written code. To show it, we first propose erroneous programs, then an explanation of the error, and finally the good solution.

The specifications are very simple. It aims at setting the light (output %Q4) as soon as the pushbutton (%I1) is pressed.

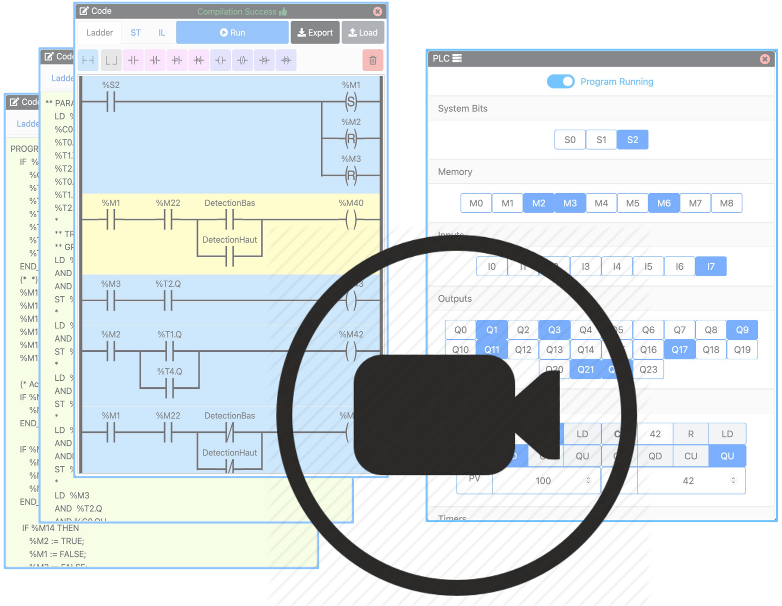
Among possible solutions, we proposed to answer to the specifications with a Grafcet with 2 steps:

According to this Grafcet, as soon as %I1 is true, the step 1 has to be activated causing the forcing of %Q4 to 1.

Wrong Code

The first way to code this Grafcet in IL with the method Transition/Transition **without** *Crossing Bits* is:

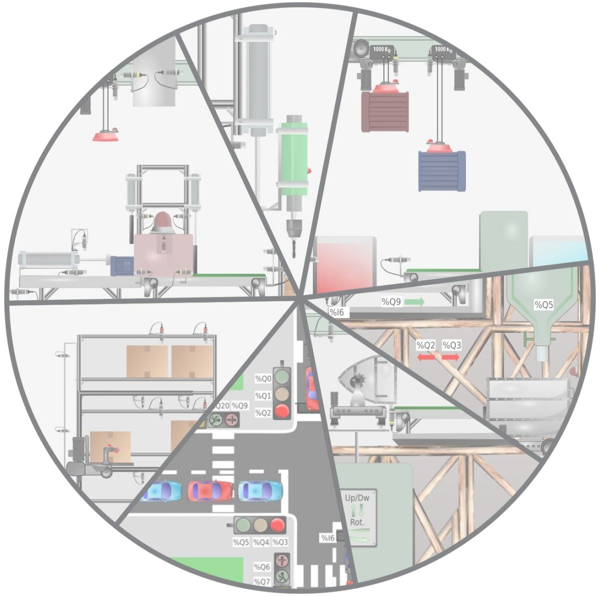
27 (13 + 14) Exercises & Tutorials



```
LD  %M3
AND  %T2.Q
IF %M14 THEN
  %M2 := TRUE;
  %M1 := FALSE;
```

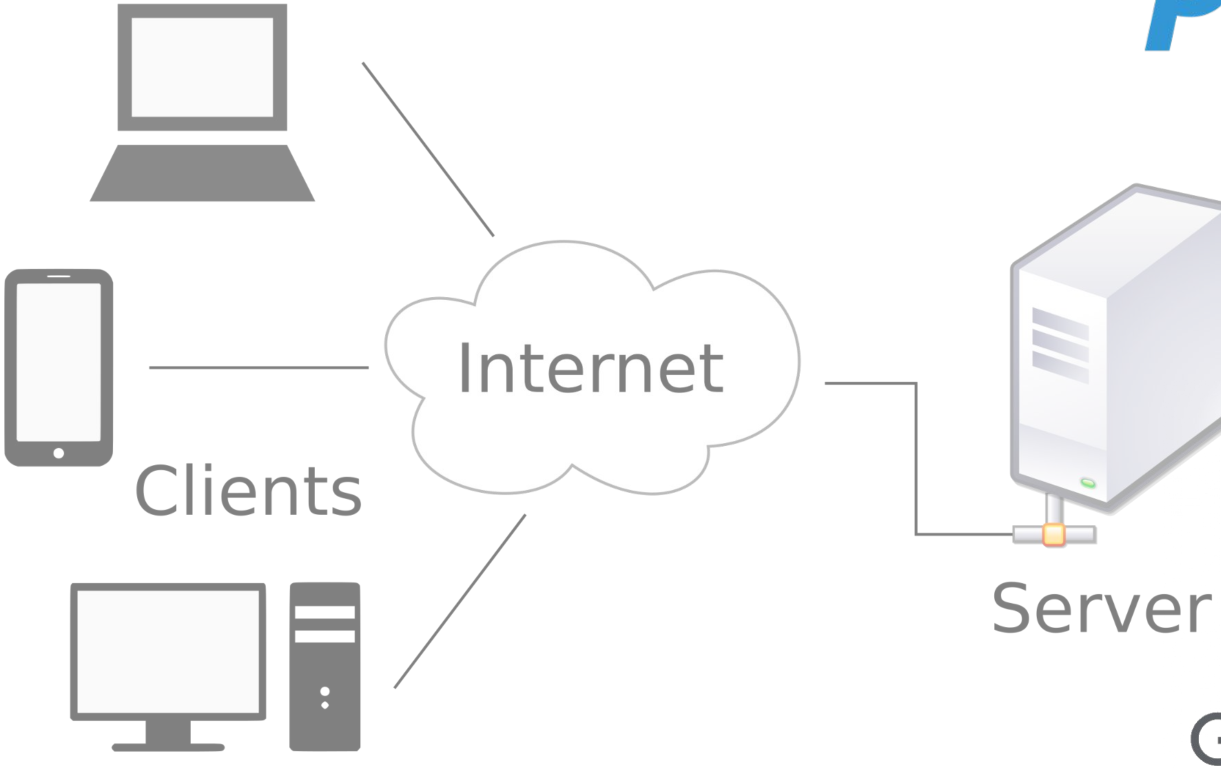
3 Programming Languages

PLC Simulator



7 (4+3) Physics Simulations

PLC3000.com Server Side

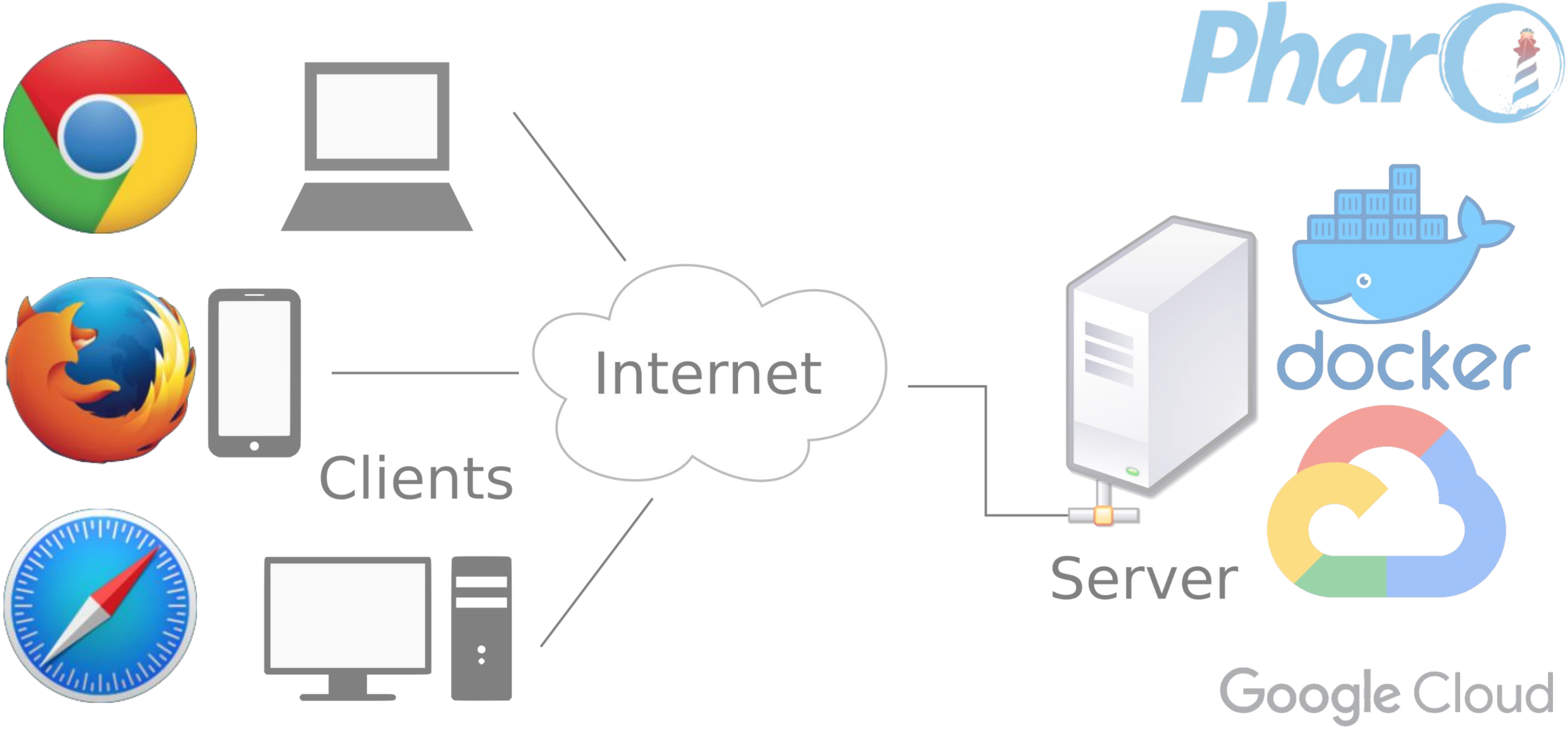


Pharo

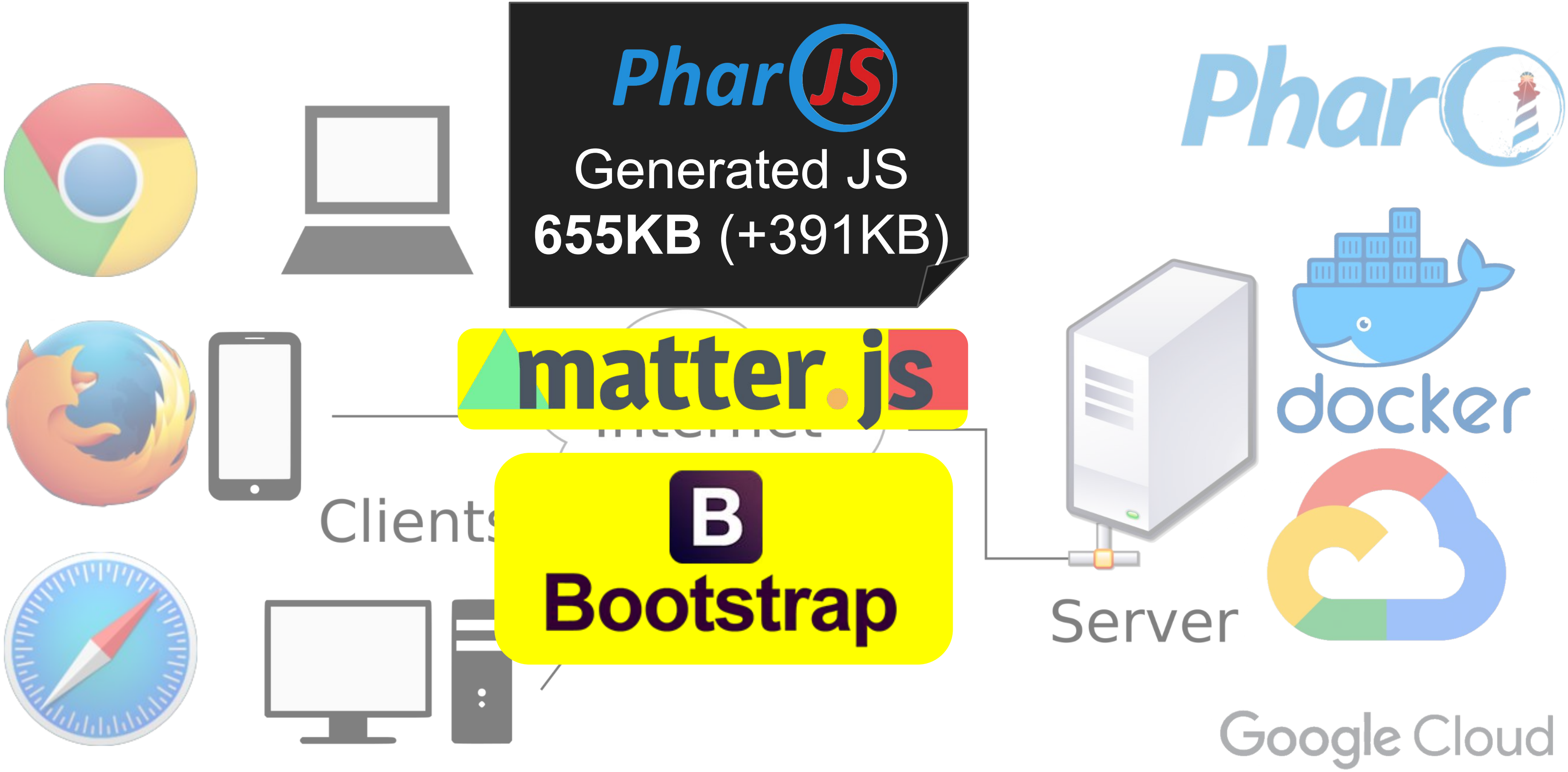
docker

Google Cloud

PLC3000.com Clients Run in Web Browsers



PLC3000.com Client JS Code



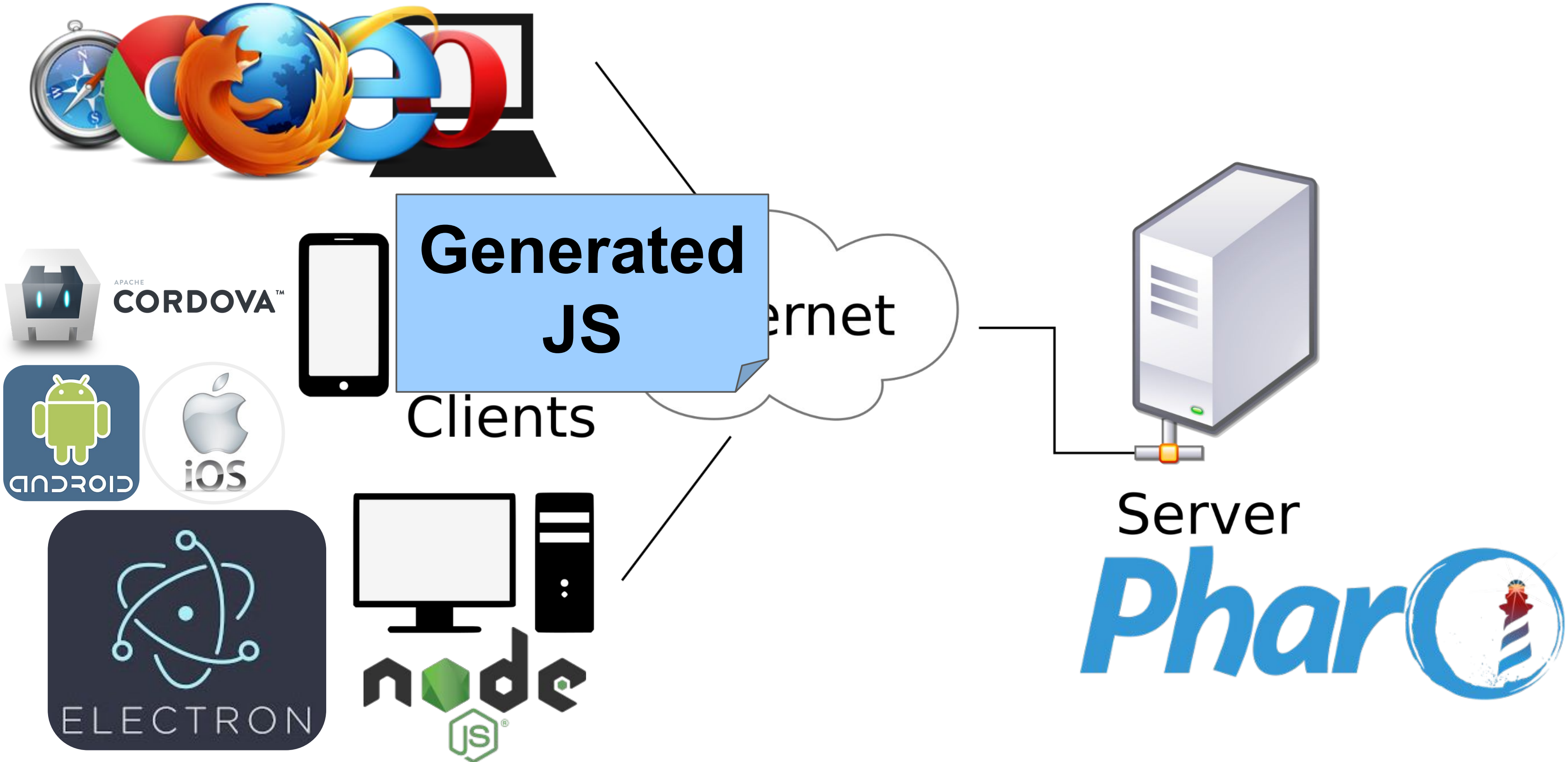
Generated JS
655KB (+391KB)

Summary

PharJS Supports Real World Applications

- Write 100% Pharo Code
- Reuse JS Libraries
- Tests + debugging in Pharo
  - Pharo talks to JavaScript
- Different Architectures are Possible

Pharo on the Client Side = PharJS



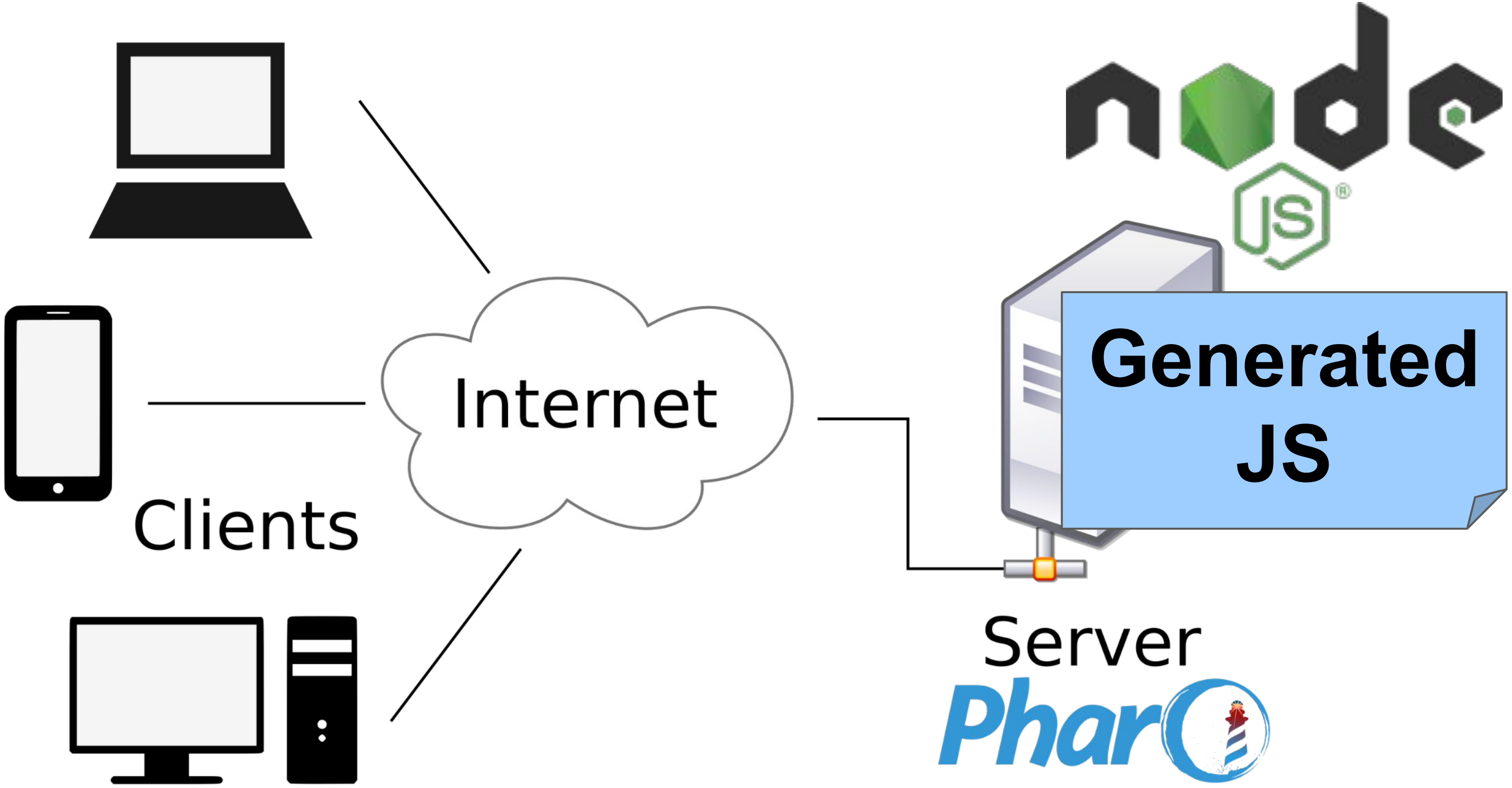
Generated JS

ernet

Clients

Server

PharJS is for Server Side Too!



Internet

Clients

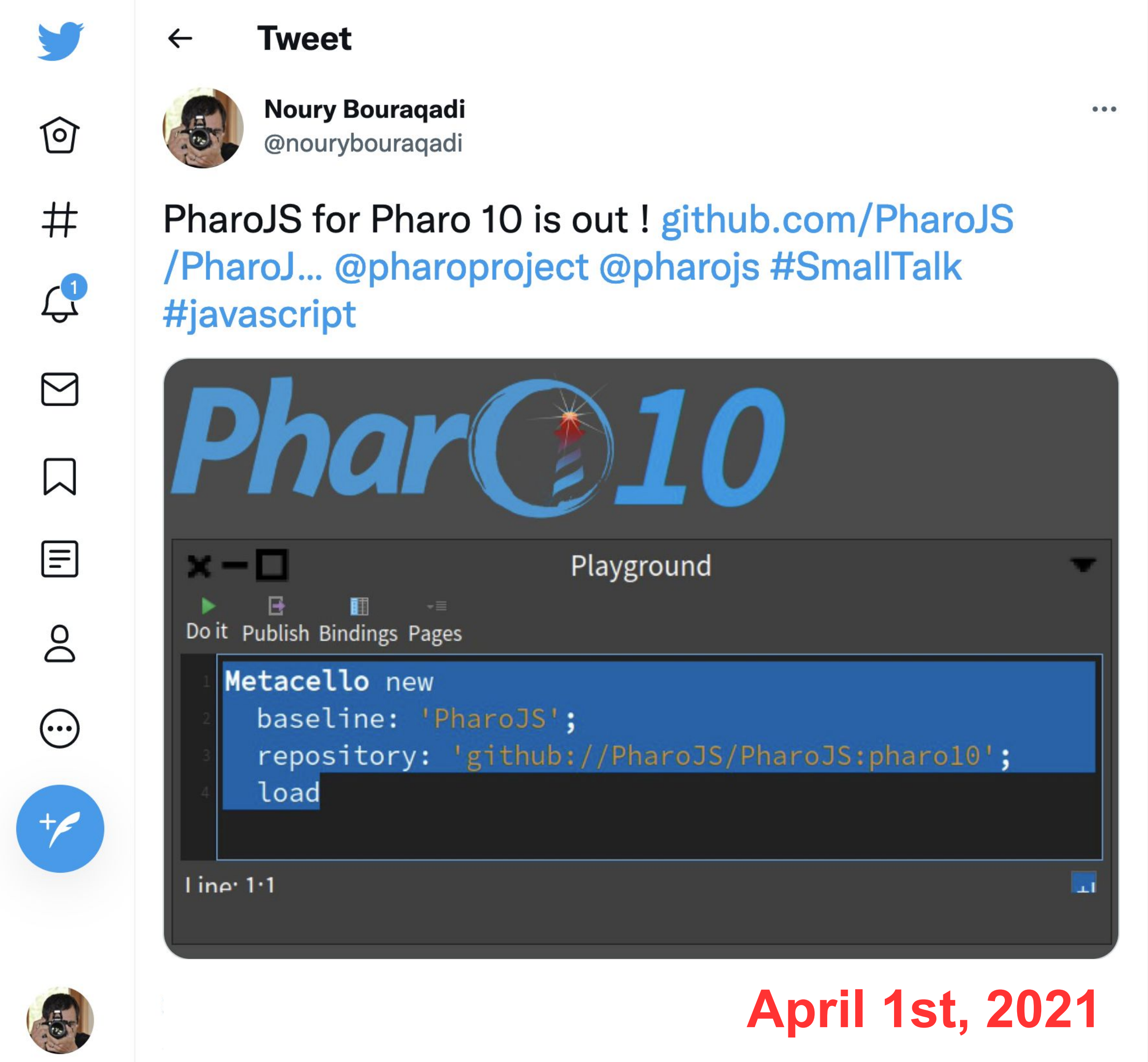
Generated JS

Server

PharJS Supports Different Workflows

🔴 Run-Time vs 🔵 Development-Time

- HTML, CSS
  - Handwritten Files 🔵
  - Generated 🔴 🔵
  - DOM Elements Creation & Setup 🔴
  - Reuse Third-Party Libraries 🔵
- Javascript
  - Generated 🔴 🔵
  - Reuse Third-Party Libraries 🔵



Noury Bouraqadi
@nourybouraqadi

PharoJS for Pharo 10 is out ! github.com/PharoJS/PharoJ... @pharoproject @pharojs #SmallTalk #javascript

```
Metacello new
  baseline: 'PharoJS';
  repository: 'github://PharoJS/PharoJS:pharo10';
  load
```

April 1st, 2021

PharoJS

for
Pharo X
is now Beta ;-)

PharJS Future Development

- Improved Middleware
  - Framework for Client-Server Apps
- Support latest JS constructs to reuse JS Frameworks
- Support more Pharo concepts (threads, slots, …)
- Extended Support for Live/Interactive Programming
  - Hot code update : easy
  - Debugging generated JS code : complex

Develop in Pharo, Run on JavaScript

PharoJS.org

Kindly supported by

EUROPEAN SMALLTALK USER GROUP
esug
www.esug.org

GitHub

Thanks to all the contributors