

Pharo Days, 3-4 March 2022, Lille, France

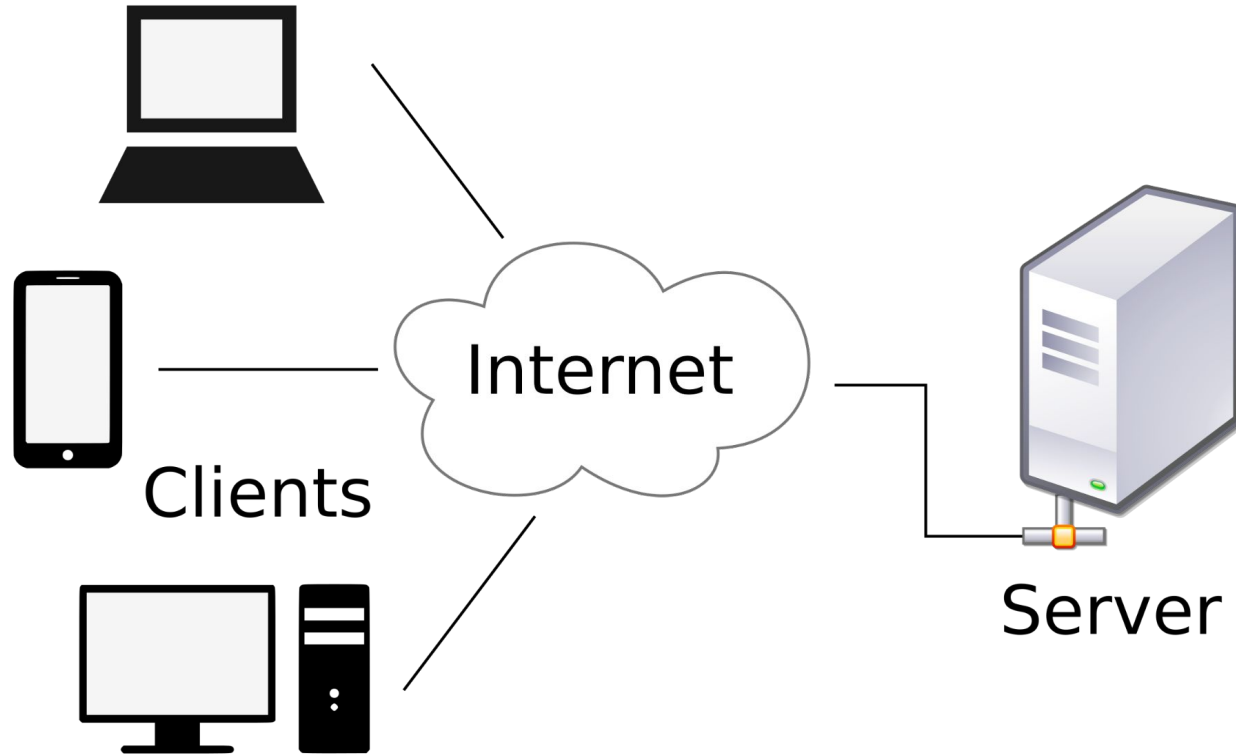
# Client-Server Development with



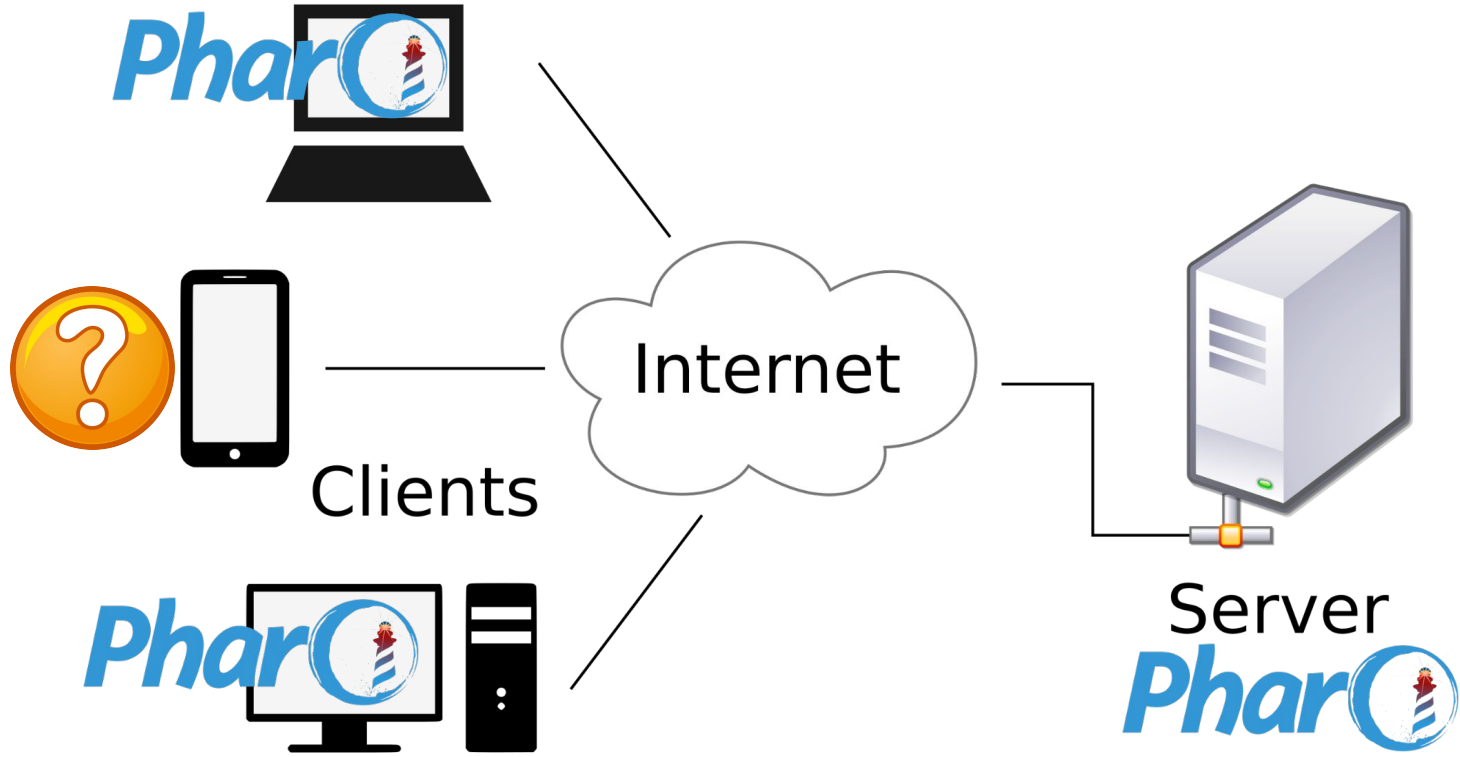
Noury Bouraqadi & Dave Mason



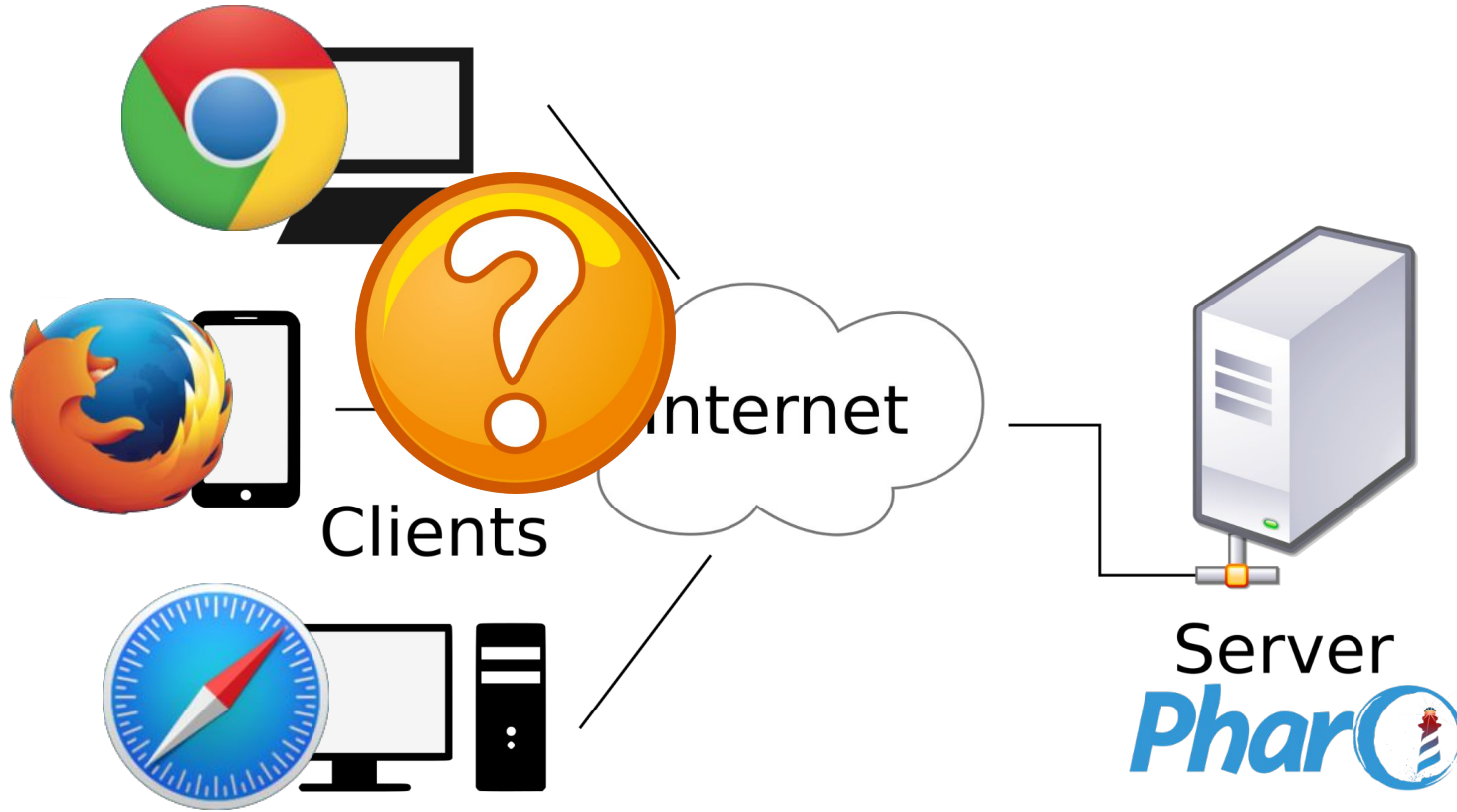
# Client-Server Applications



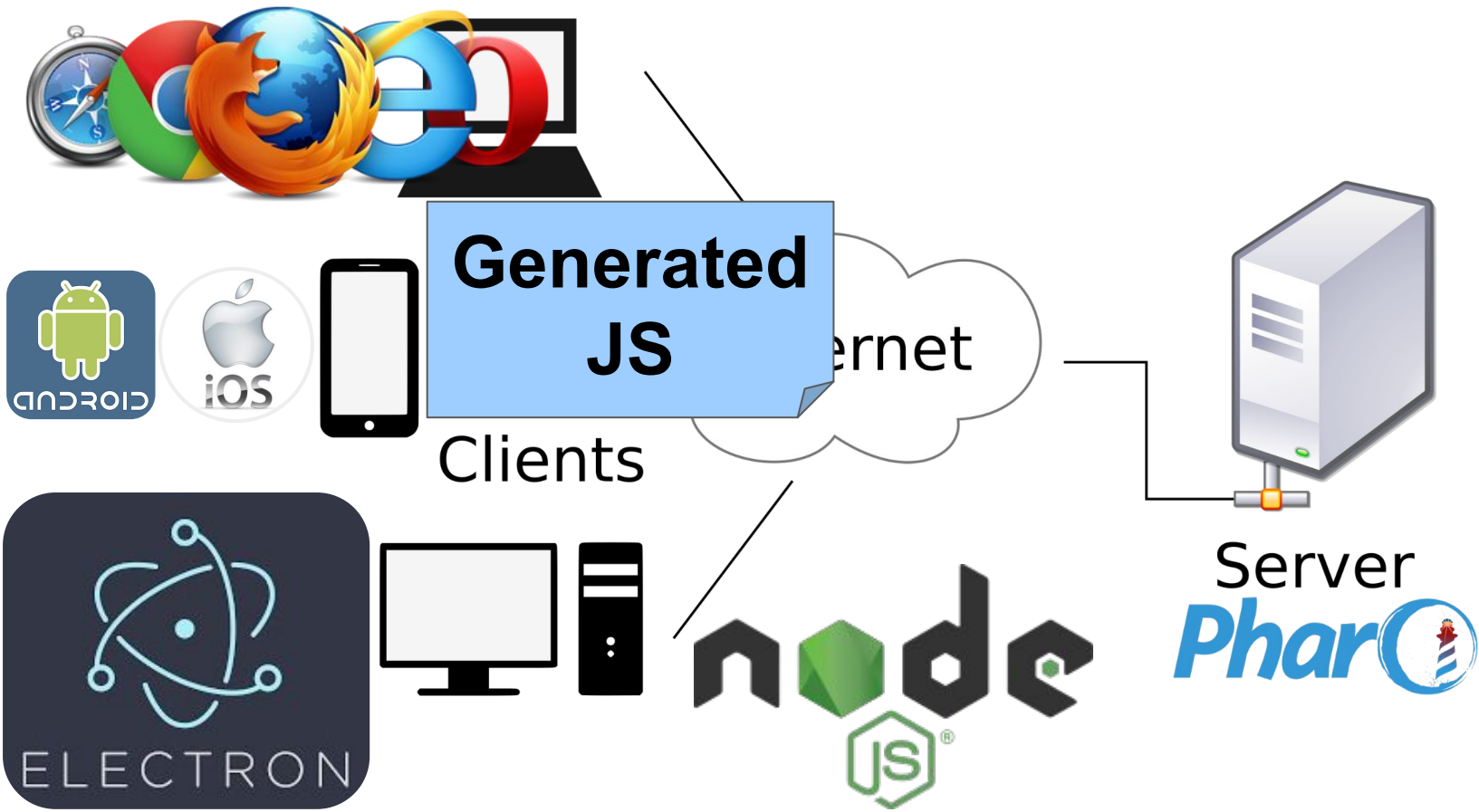
# Client-Server Applications with Pharo



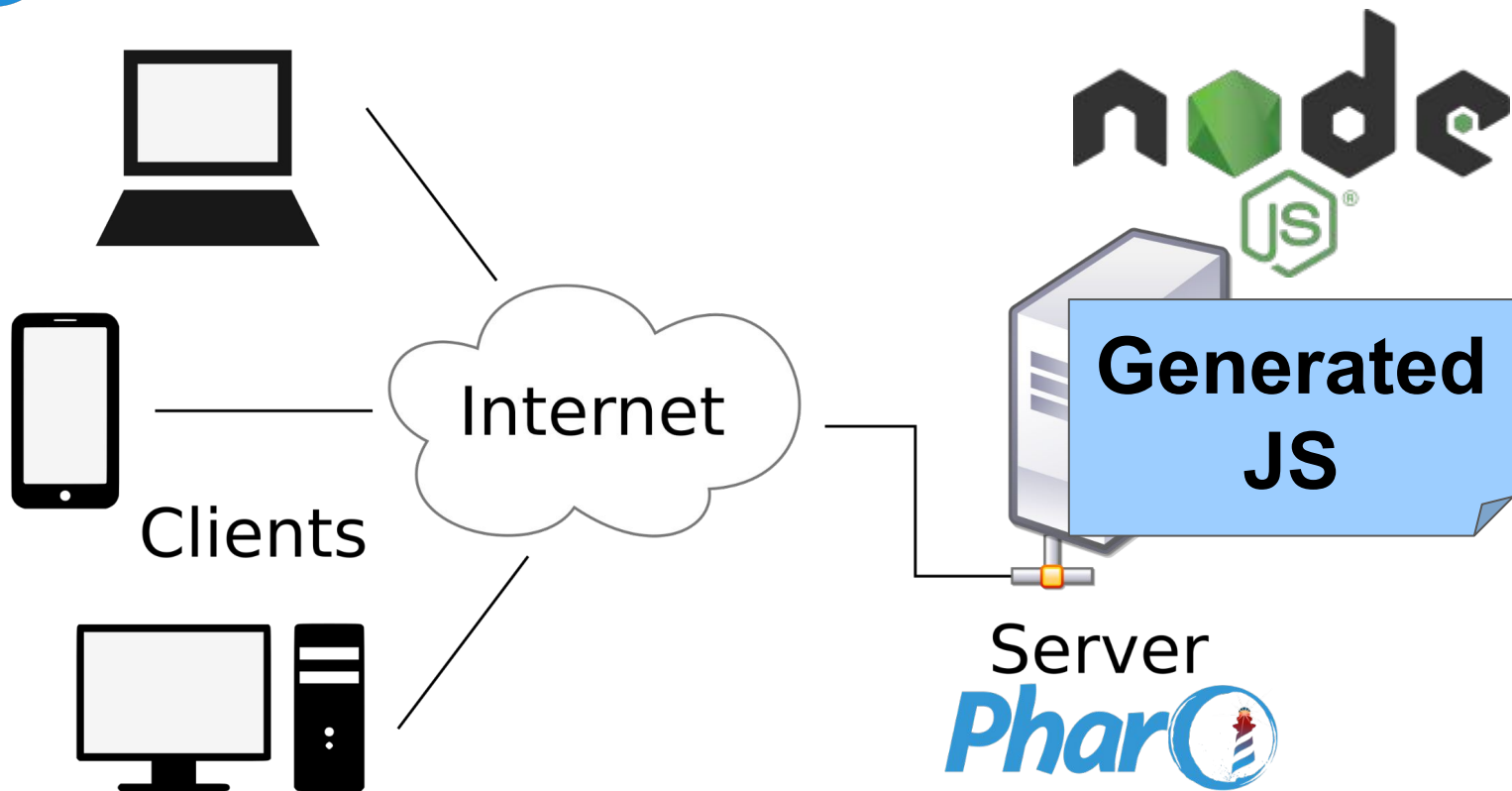
# Pharo on the Client Side?



# Pharo on the Client Side with *PharJS*



# PharJS for Server Side Too!





The image shows a screenshot of the Pharo IDE interface. At the top center, the Pharo logo is displayed in a white rounded rectangle. Below it, the text "1) Develop in Pharo" is written in large white font. In the center, "2) Generate JS" is written in large white font. Below that, the PharoJS logo is shown in a white rounded rectangle. The background of the screenshot includes a "Playground" window with code, a "Variables" window, and a "Bytecode" window.

Pharo

1) Develop in Pharo

2) Generate JS

PharoJS



The image shows a row of logos for various platforms and technologies. From left to right: the Android logo, the iOS logo, a compass icon, the Electron logo, and the Node.js logo. The text "3) Run on JS" is written in large white font across the center of these logos.

3) Run on JS

ANDROID

iOS

ELECTRON

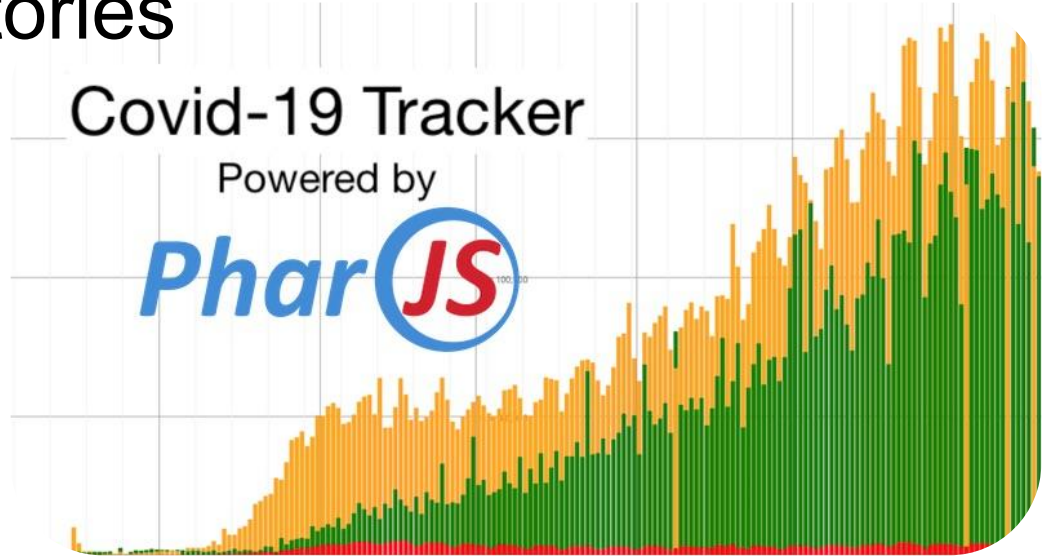
node JS

# PharJS Success Stories



Covid-19 Tracker

Powered by

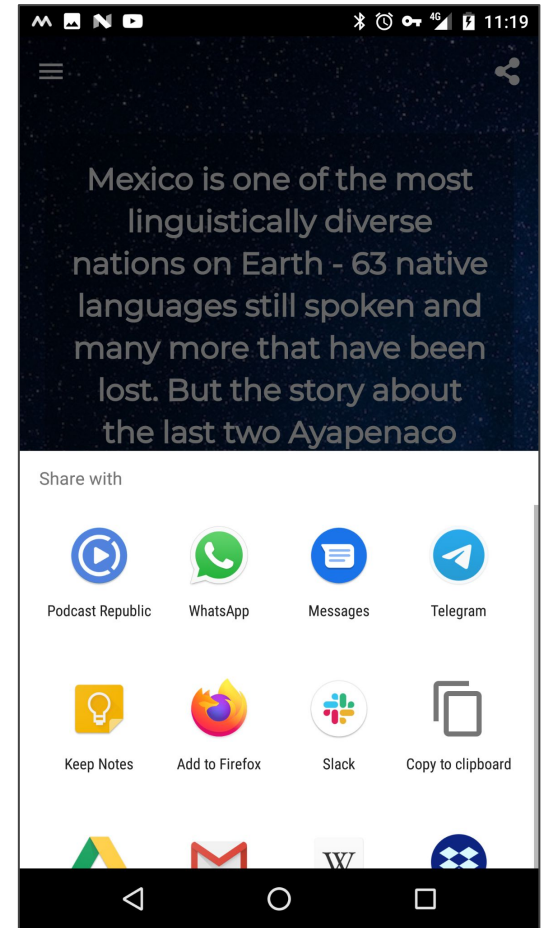
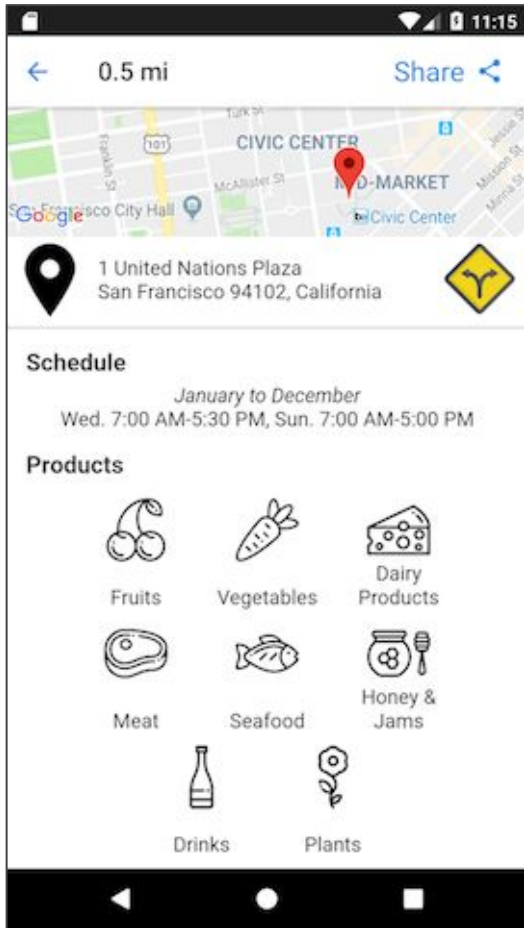


**PharJS**  
*Smalltalk REPL*

**PLC3000**  
Teaching PLC Automation  
Made Easy



# PharJS Success Stories



# Phar*JS* in a Nutshell

- **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

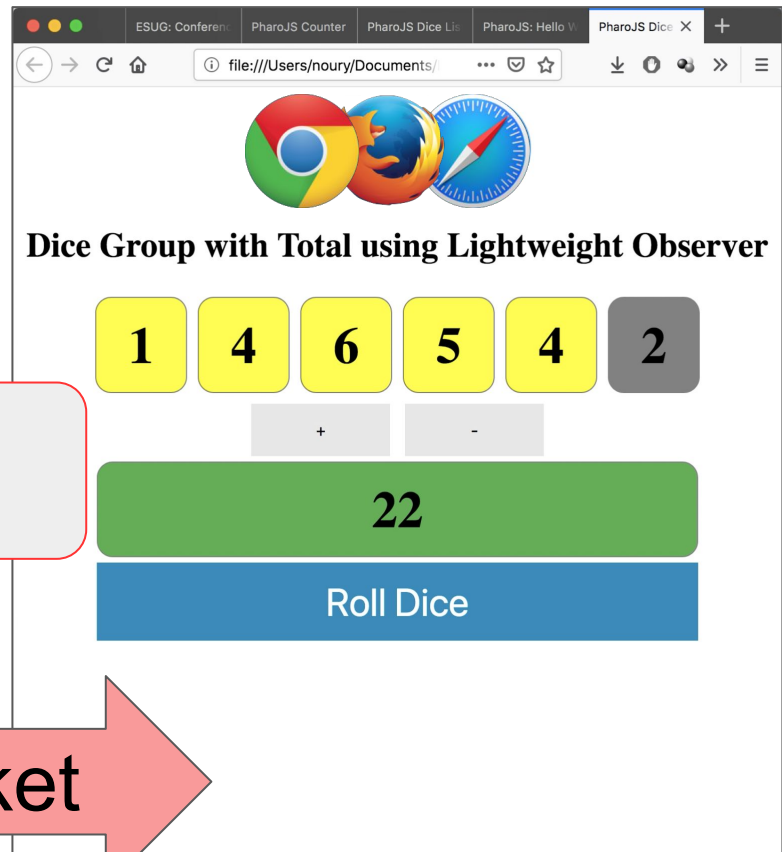
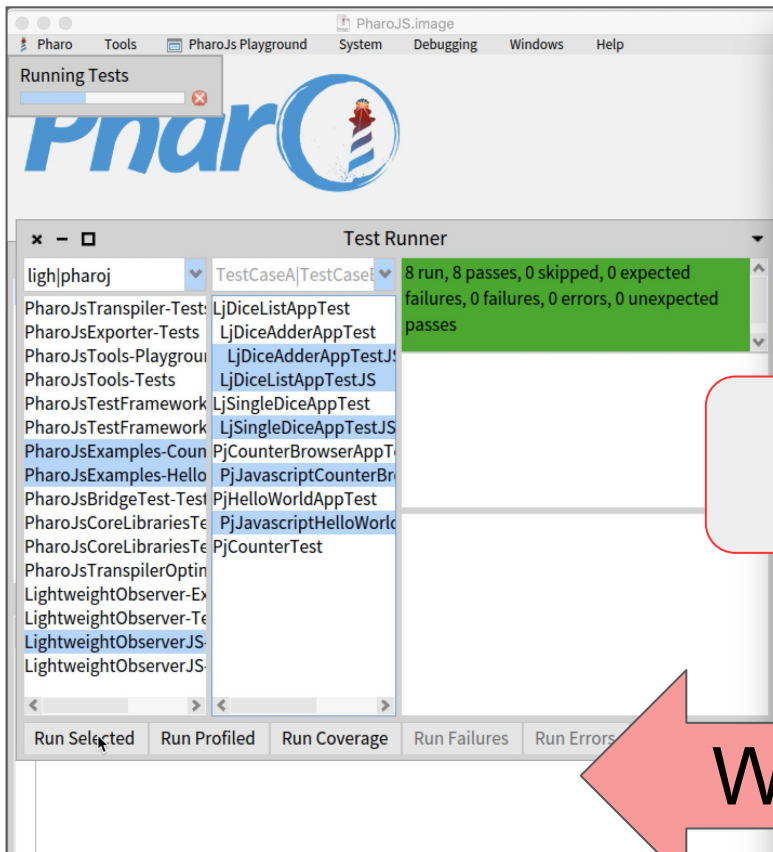
# PharJS in a Nutshell

- **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



Client-Server

# PharoJS Tests Talk to Web Browsers



Video

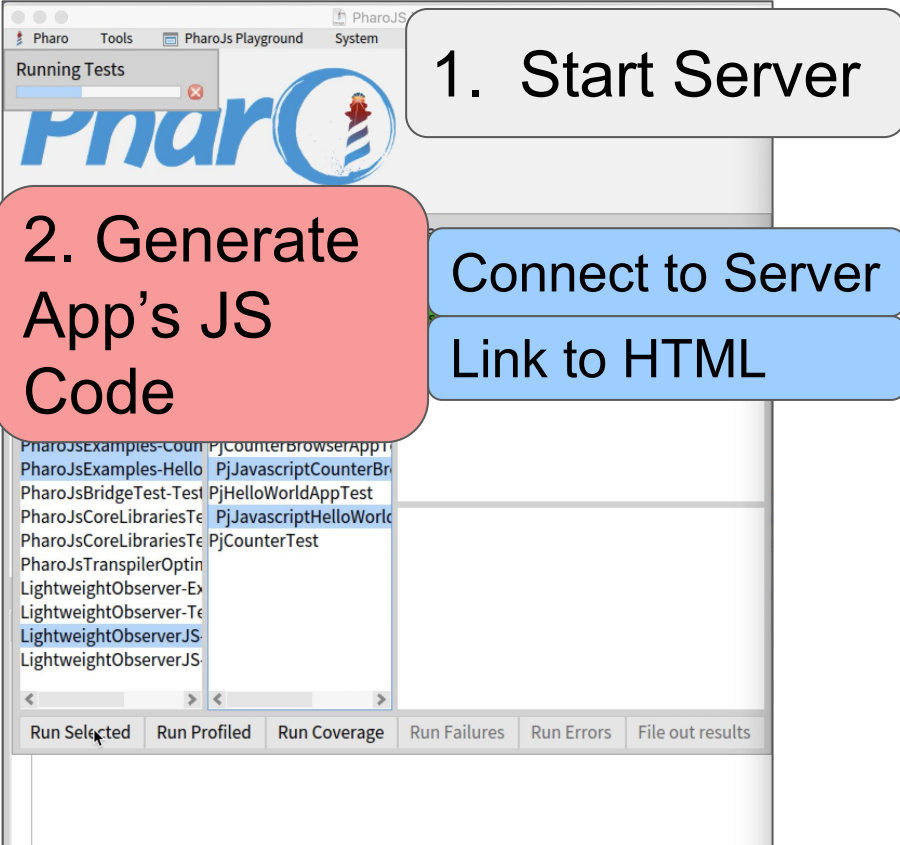
Web Socket

## 1. Start Server

The screenshot shows the PharoJS Playground interface. At the top, there's a 'Running Tests' progress bar. Below it is the PharoJS logo. The main area is a 'Test Runner' window. It has a list of test cases on the left, including 'LjDiceListAppTest', 'LjDiceAdderAppTest', and 'LjSingleDiceAppTest'. The right side of the window shows the results for the selected test: '8 run, 8 passes, 0 skipped, 0 expected failures, 0 failures, 0 errors, 0 unexpected passes'. At the bottom of the Test Runner window, there are buttons for 'Run Selected', 'Run Profiled', 'Run Coverage', 'Run Failures', 'Run Errors', and 'File out results'.

The screenshot shows a web browser window displaying a dice game interface. The browser's address bar shows the file path: 'file:///Users/noury/Documents/'. The page features a title 'Dice Group with Total using Lightweight Observer'. Below the title, there are six dice represented by colored squares: 1 (yellow), 4 (yellow), 6 (yellow), 5 (yellow), 4 (yellow), and 2 (grey). Below the dice are '+' and '-' buttons. A large green box displays the total value '22'. At the bottom, there is a blue button labeled 'Roll Dice'.

# PharoJS Tests Talk to Web Browsers



The screenshot shows the PharoJS Playground interface. A 'Running Tests' window is open at the top left. The main area displays the PharoJS logo and a list of test classes. A red callout box highlights the step '2. Generate App's JS Code'. A grey callout box at the top right indicates '1. Start Server'. Two blue callout boxes below it indicate 'Connect to Server' and 'Link to HTML'.

1. Start Server

2. Generate App's JS Code

Connect to Server

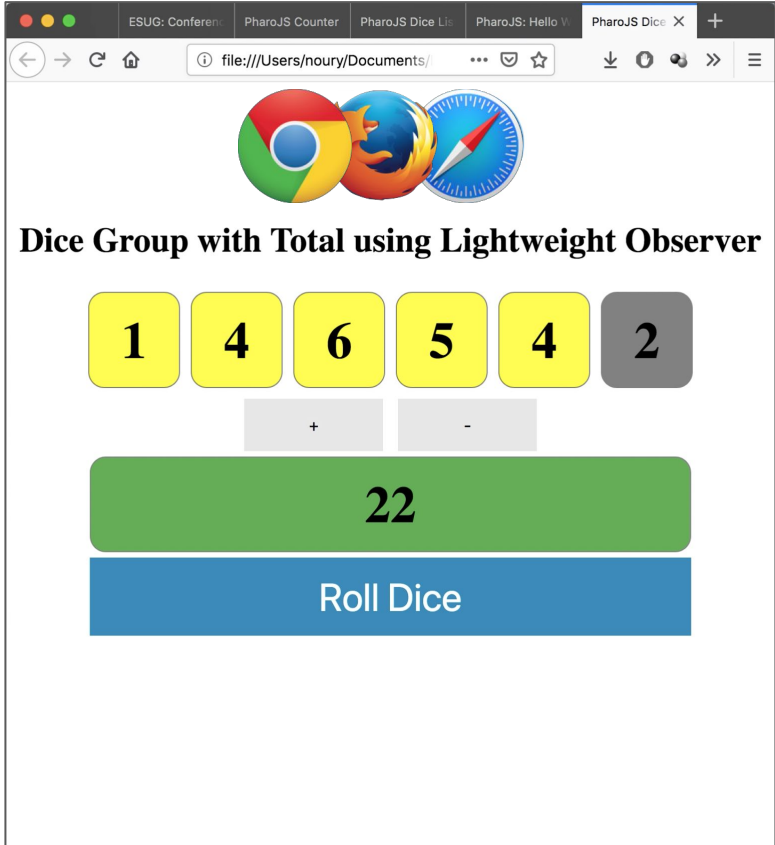
Link to HTML

Running Tests

PharoJS

PharoJSexamples-CounterBrowserAppTest  
PharoJSexamples-HelloWorldAppTest  
PharoJSBridgeTest-Test  
PharoJSCoreLibrariesTest  
PharoJSCoreLibrariesTest  
PharoJSTranspilerOptimizationsTest  
LightweightObserver-ExampleTest  
LightweightObserver-Test  
LightweightObserverJS-Test  
LightweightObserverJS-Test

Run Selected Run Profiled Run Coverage Run Failures Run Errors File out results



The screenshot shows a web browser window displaying a dice game interface. The title is 'Dice Group with Total using Lightweight Observer'. The interface shows six dice with faces 1, 4, 6, 5, 4, and 2. Below the dice are '+' and '-' buttons. A green bar displays the total '22'. A blue button labeled 'Roll Dice' is at the bottom.

file:///Users/noury/Documents/

Dice Group with Total using Lightweight Observer

1 4 6 5 4 2

+

-

22

Roll Dice

# PharoJS Tests Talk to Web Browsers

The screenshot shows the PharoJS Playground interface. A 'Running Tests' window is open at the top left. The main area displays the PharoJS logo and a list of test examples: 'PharoJSexamples-CounterBrowserAppTest', 'PharoJSexamples-HelloPjJavascriptCounterBrowserAppTest', and 'PharoJSBridgeTest-TestPjHelloWorldAppTest'. A 'Run Selected' button is visible at the bottom.

**1. Start Server**

**2. Generate App's JS Code**

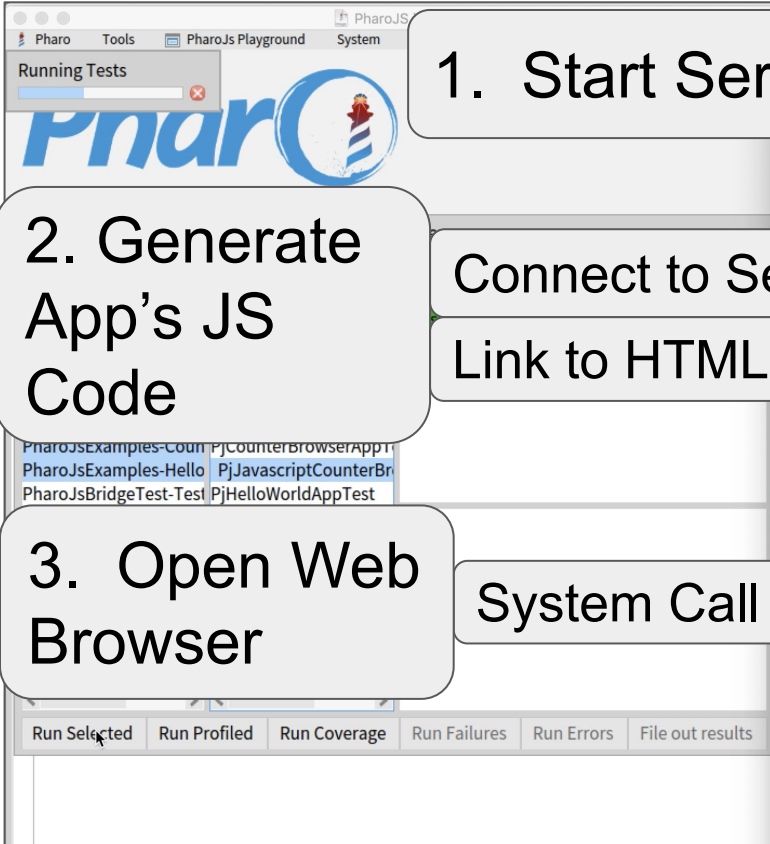
**3. Open Web Browser**

Callouts for step 1 include: 'Connect to Server' and 'Link to HTML'.

Callout for step 3 includes: 'System Call'.

The screenshot shows a web browser window displaying a dice game interface. The title bar shows the file path: 'file:///Users/noury/Documents/'. The page content includes the PharoJS logo and the text: 'Dice Group with Total using Lightweight Observer'. Below this, there are six dice faces showing the numbers 1, 4, 6, 5, 4, and 2. A green bar displays the total value '22'. A blue button labeled 'Roll Dice' is at the bottom.

# PharoJS Tests Talk to Web Browsers



1. Start Server

2. Generate App's JS Code

3. Open Web Browser

Connect to Server

Link to HTML

System Call

Running Tests

PharoJS Playground

System

PharoJSexamples-CounterBrowserAppTest

PharoJSexamples-HelloWorldAppTest

PharoJSexamples-HelloWorldAppTest

Run Selected

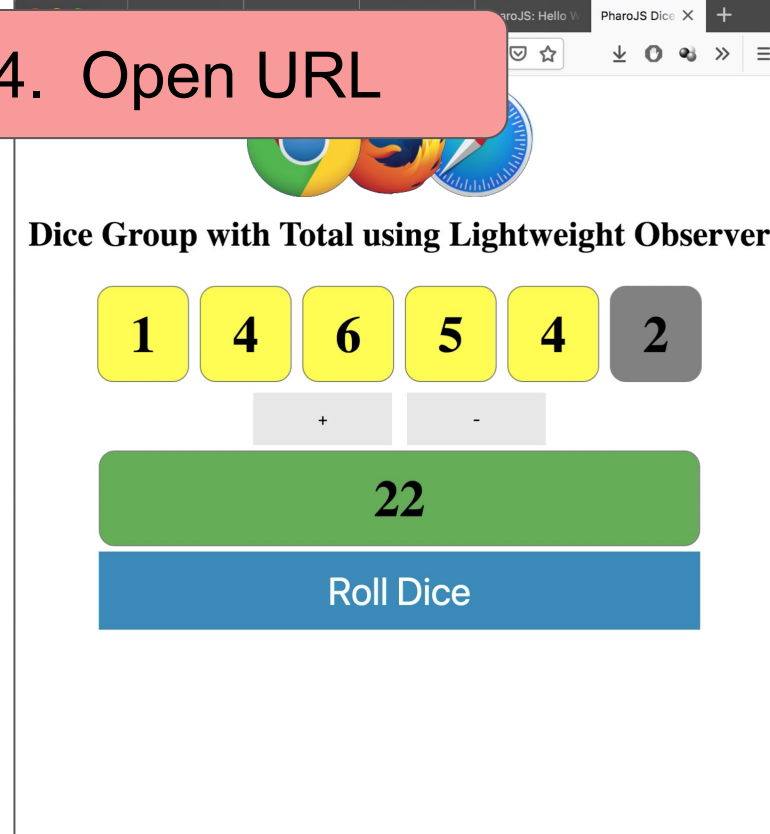
Run Profiled

Run Coverage

Run Failures

Run Errors

File out results



4. Open URL

Dice Group with Total using Lightweight Observer

1 4 6 5 4 2

+

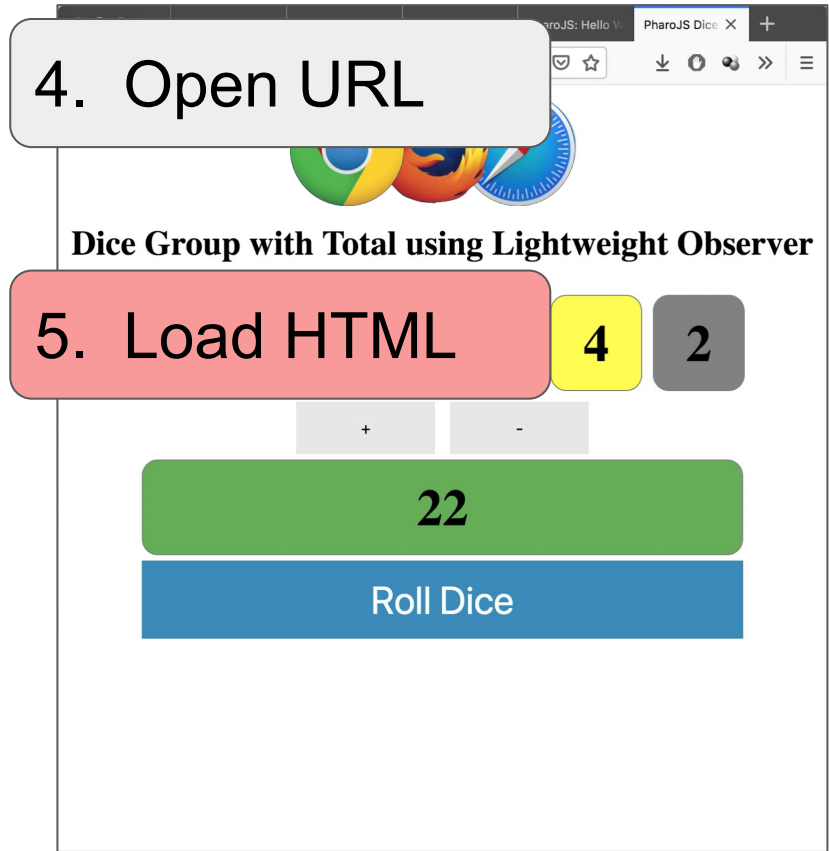
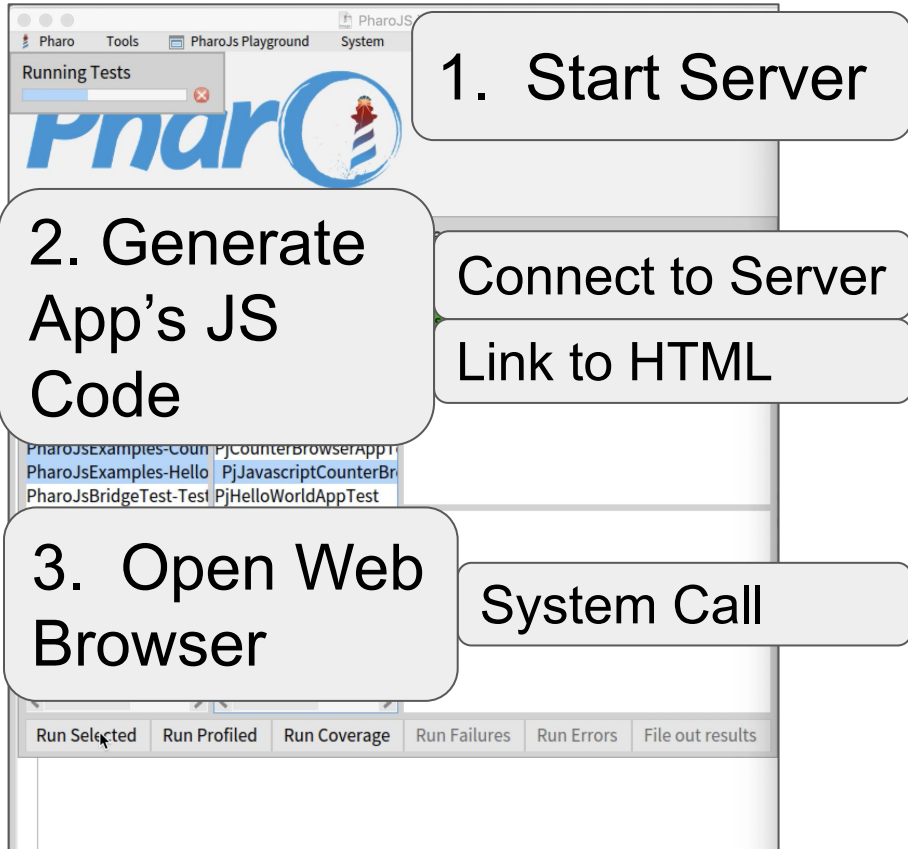
-

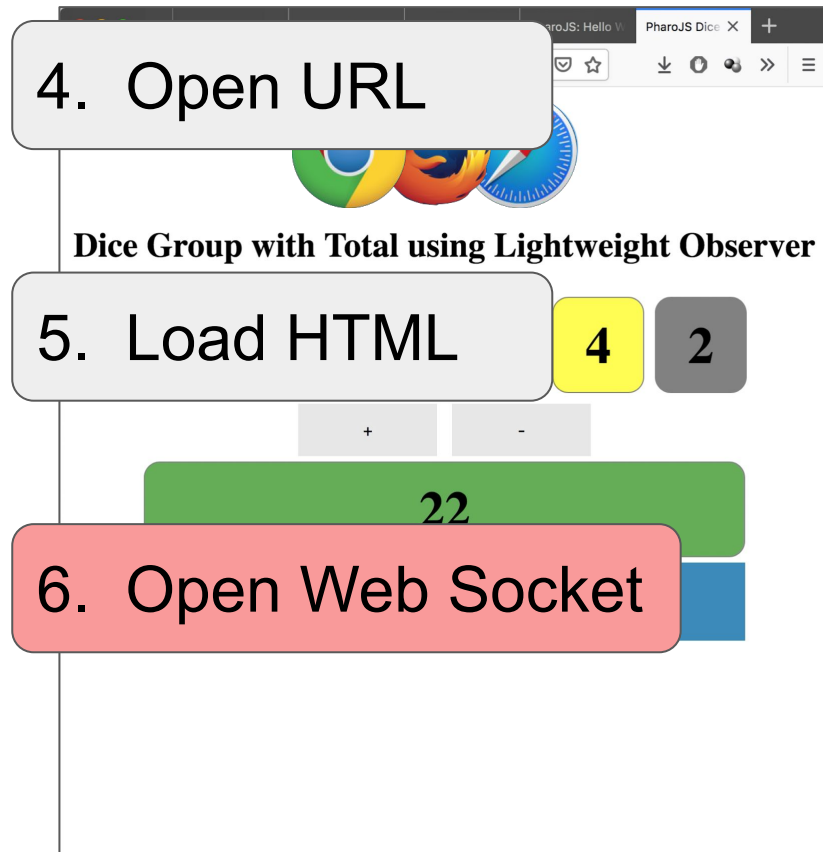
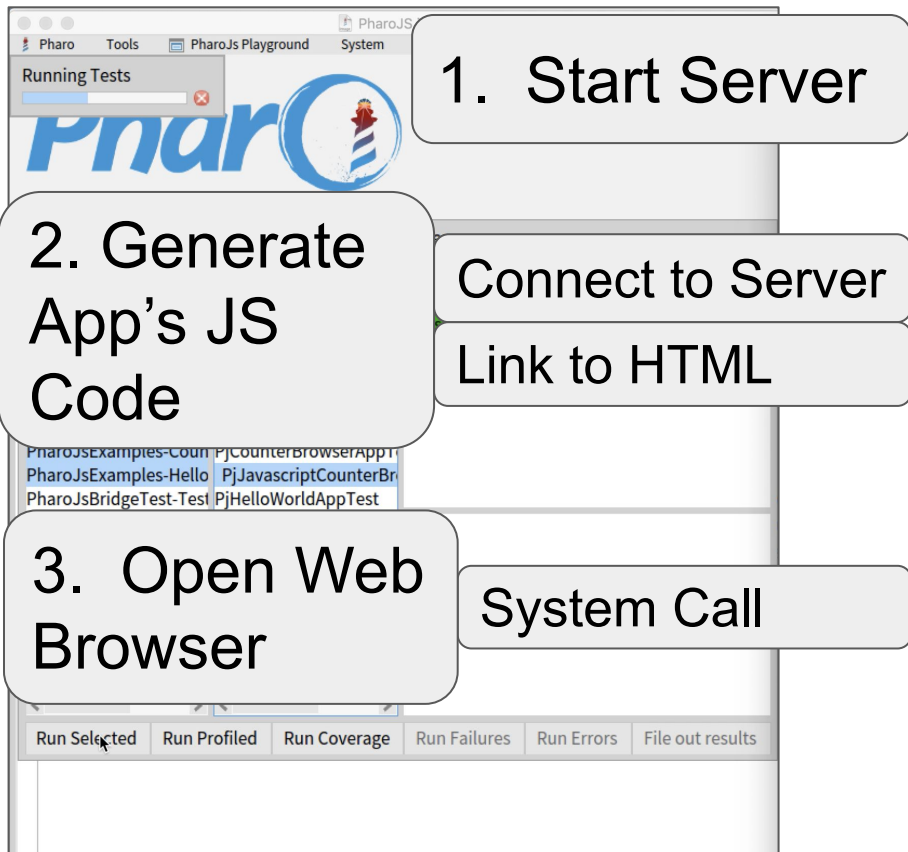
22

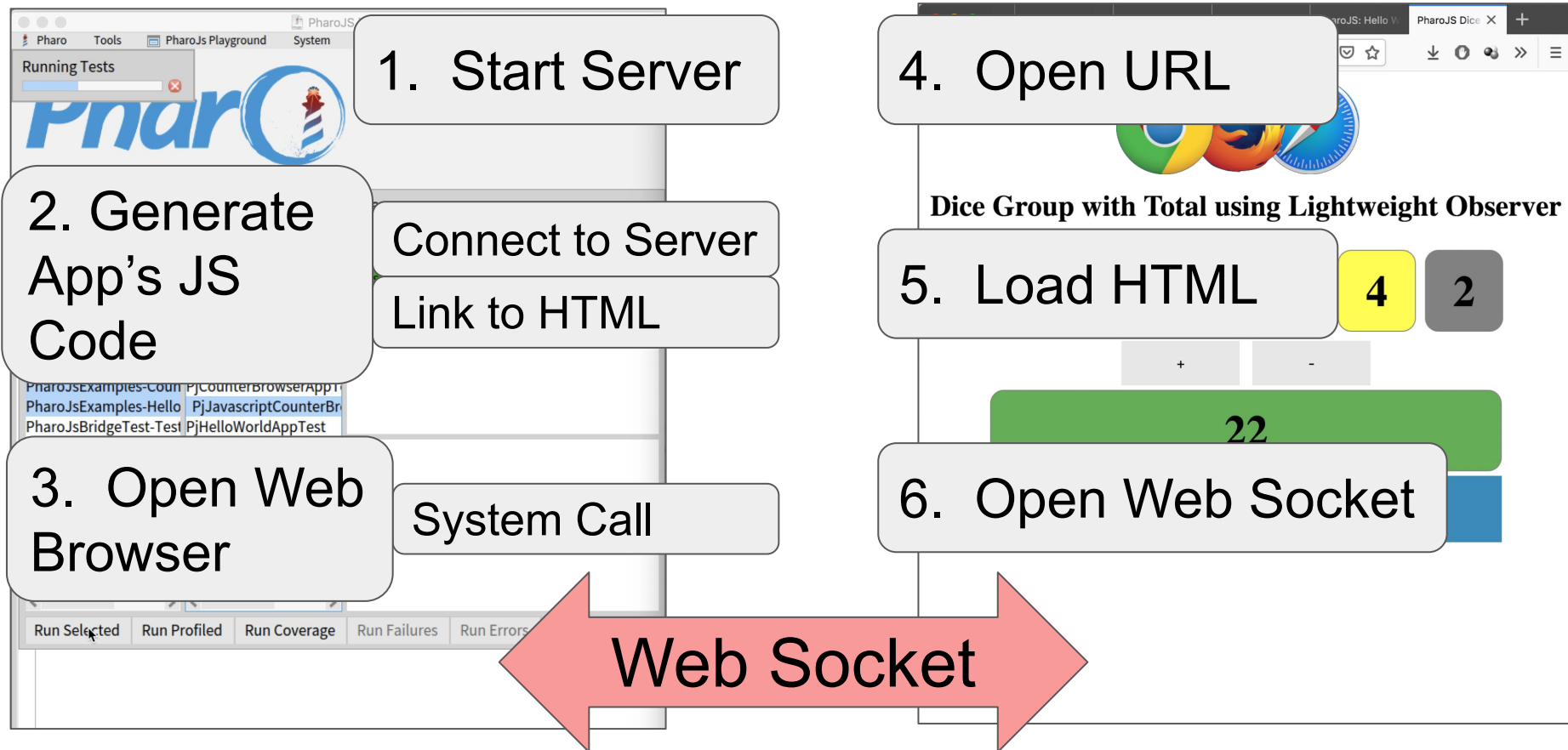
Roll Dice



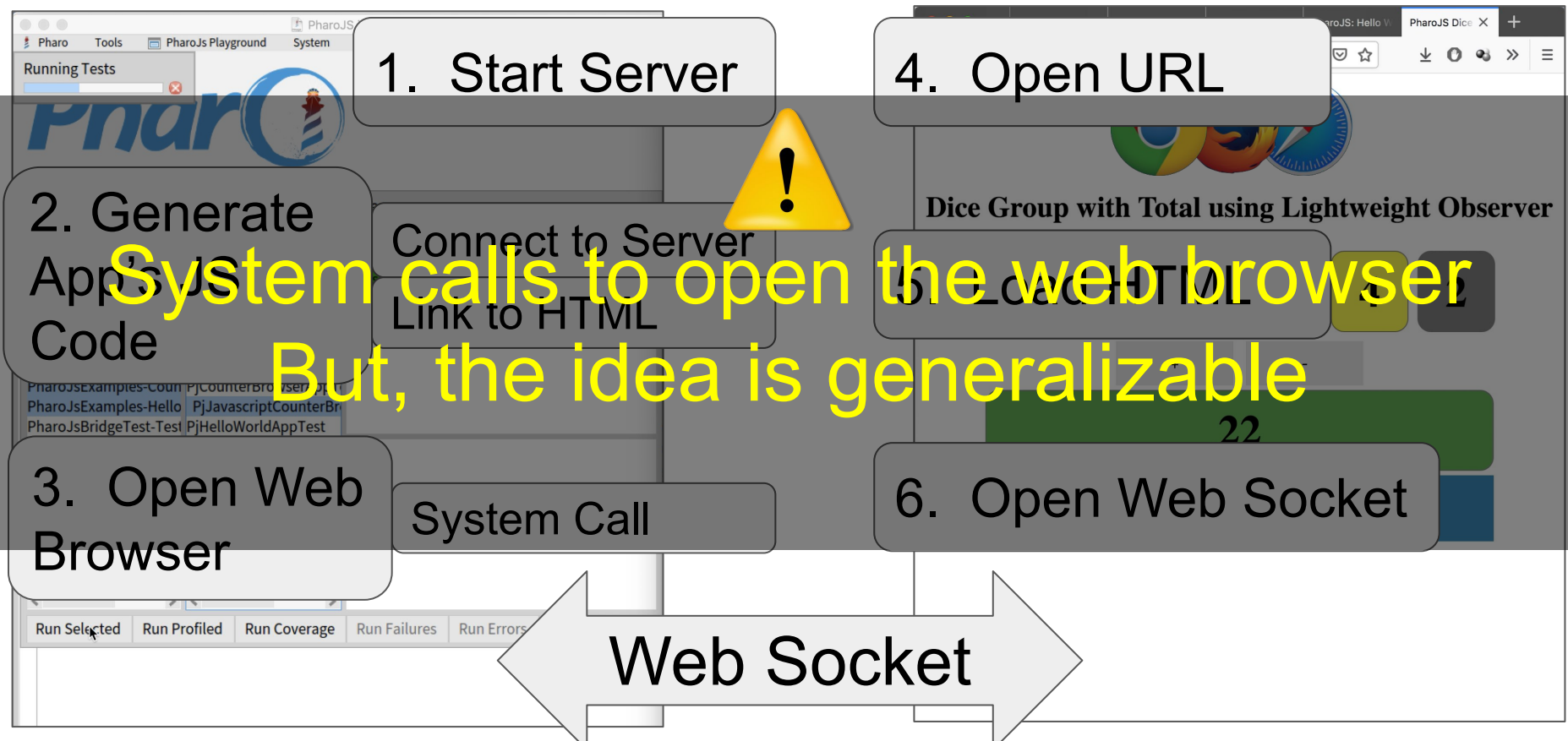
# PharoJS Tests Talk to Web Browsers







# PharoJS Tests Talk to Web Browsers





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

Transcript

```
I love Pharo!
```

Eval

Video

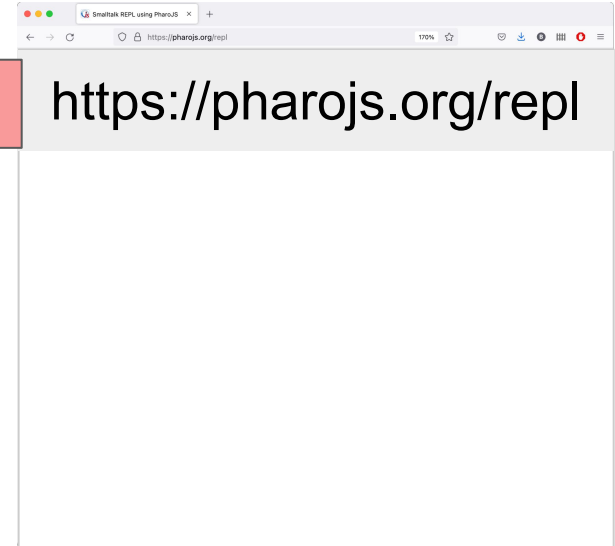
# HTML is a String in the image



Zinc HTTP  
Server

GET

HTML  
String



# HTML is a String in the image



Zinc HTTP  
Server

GET

HTML  
String



```
<!DOCTYPE html>
<html>
...

</script>
</body>
</html>
```

# Browser Processes the HTML



Zinc HTTP  
Server



```
<!DOCTYPE html>
```

```
<html>
```

```
...
```

```

```

```
</script>
```

```
</body>
```

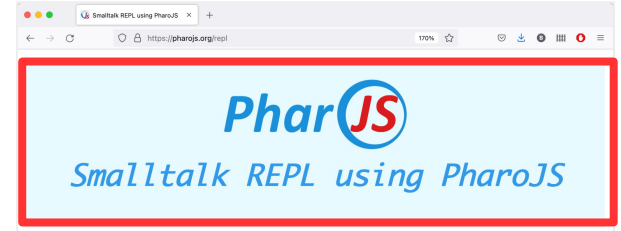
```
</html>
```



# Browser Loads Resources



Zinc HTTP  
Server



```
<!DOCTYPE html>
<html>
...

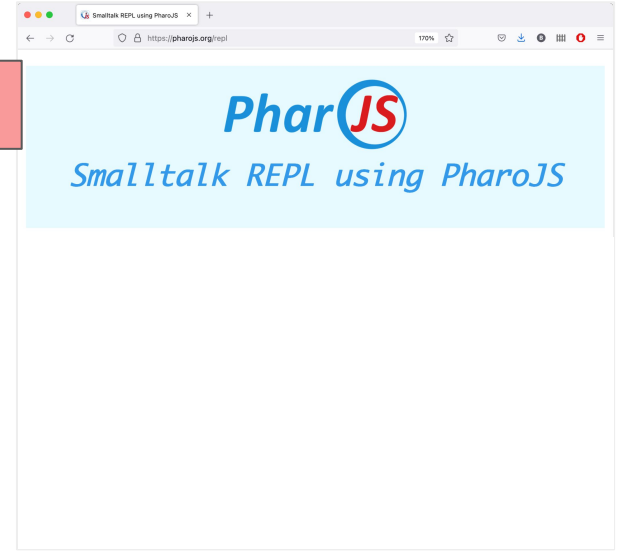
</script>
</body>
</html>
```

# Browser Requests JavaScript Code



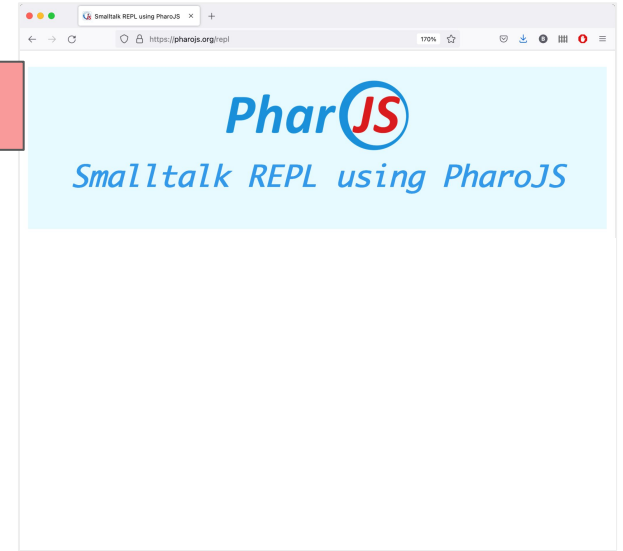
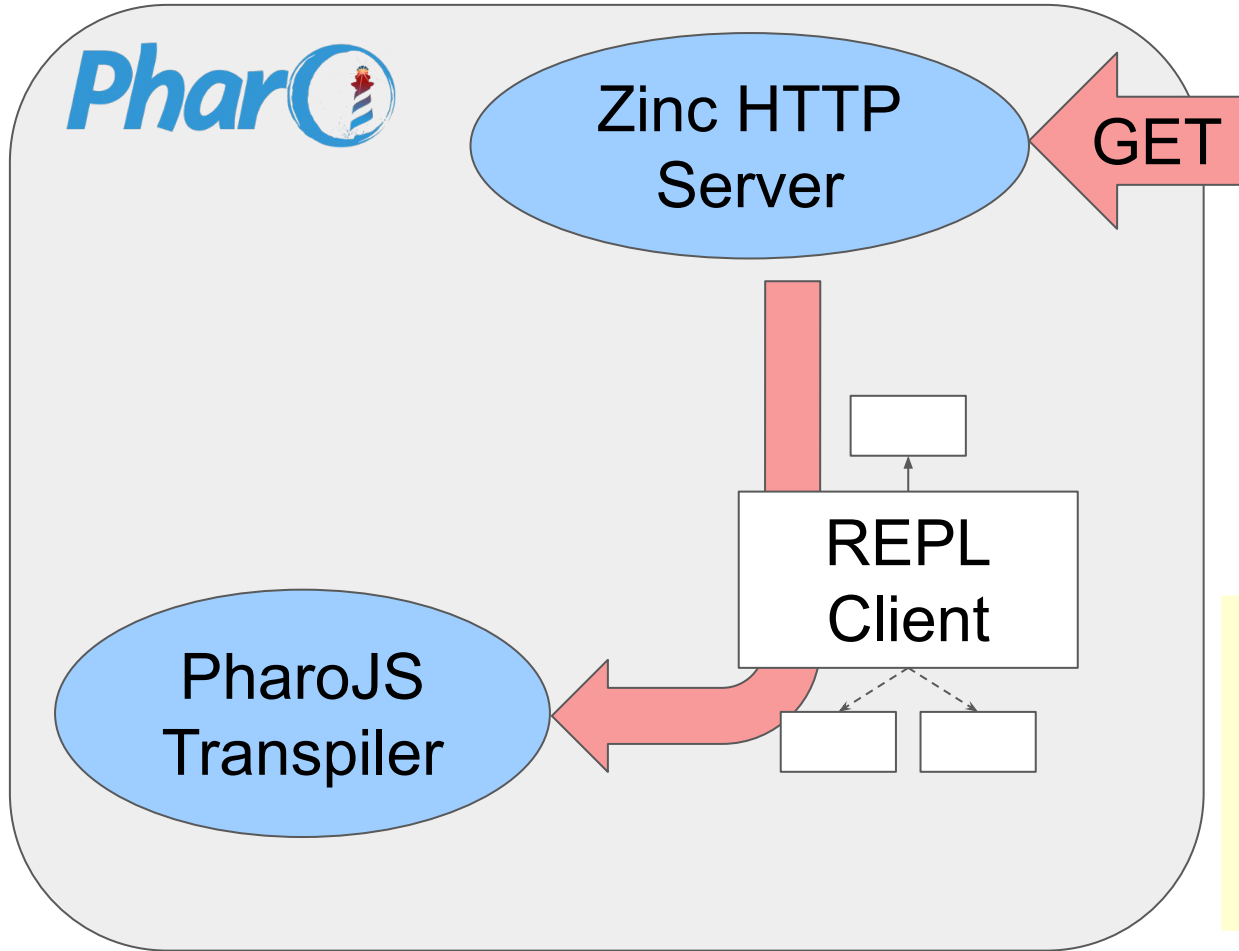
Zinc HTTP  
Server

GET



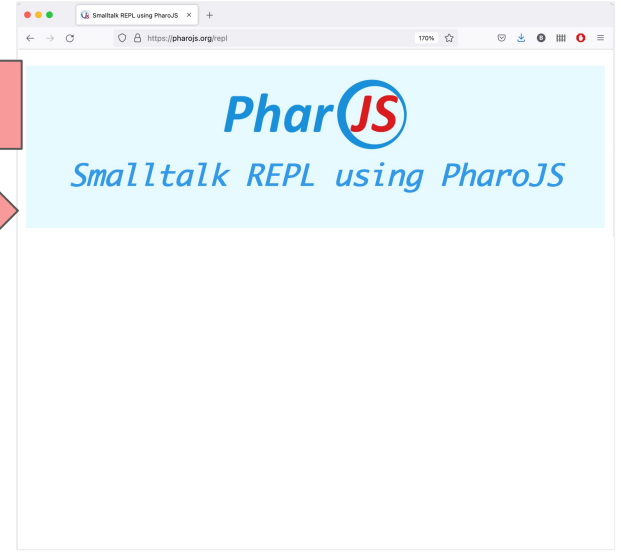
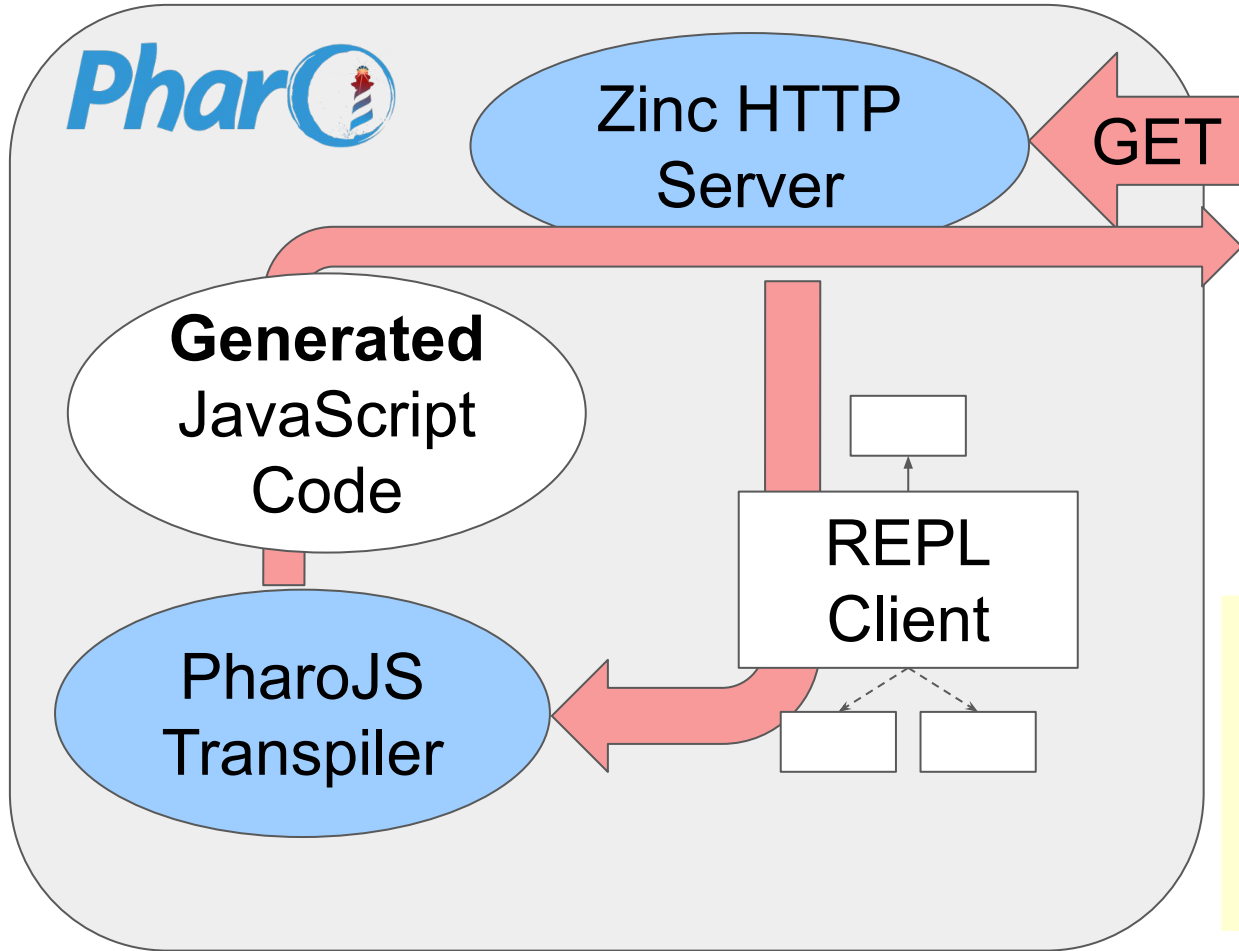
```
<script src="repl/index.js">  
</script>  
</body>  
</html>
```

# REPL Client JS Code is Generated



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

# REPL Client JS Code is Generated

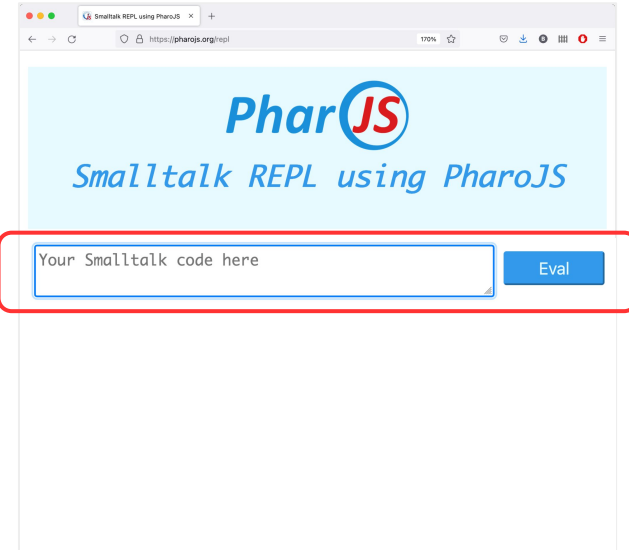


```
<script src="repl/index.js">  
</script>  
</body>  
</html>
```

# Client Creates and Links DOM Elements



Zinc HTTP  
Server



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

# Client Sends ST Code Snippet



Zinc HTTP  
Server

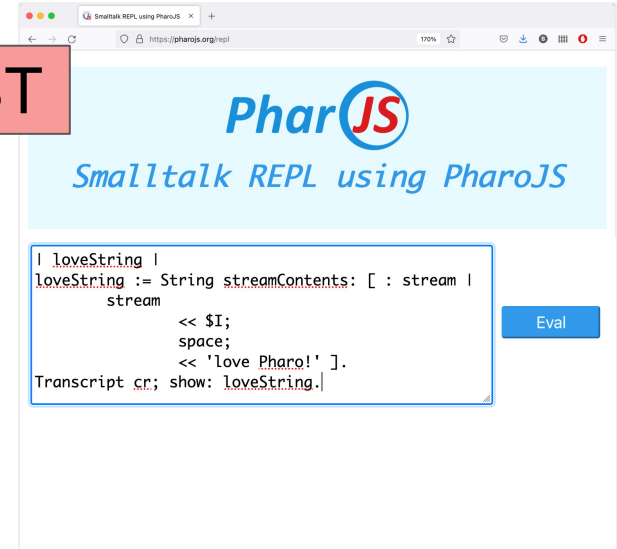
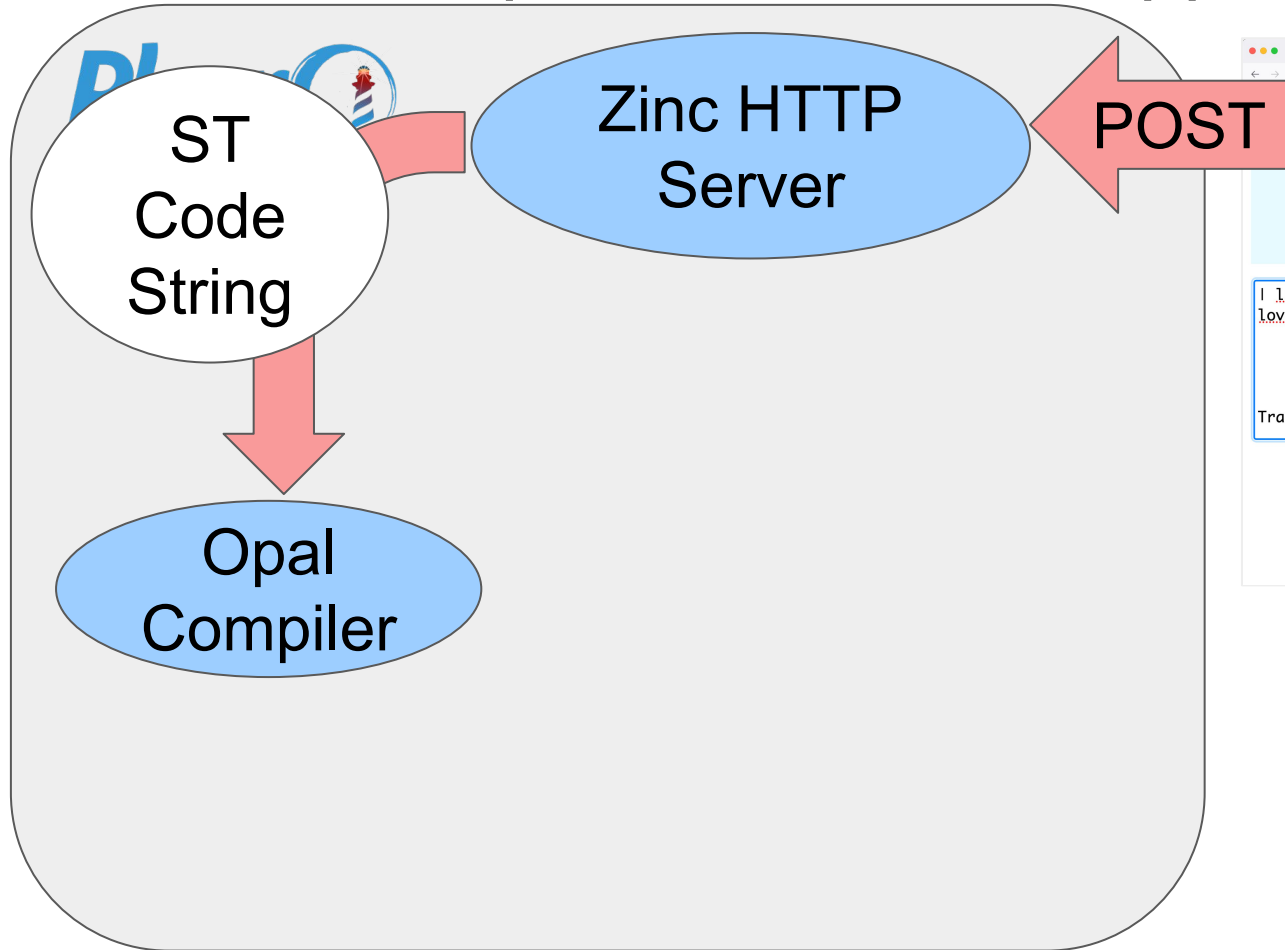
POST

A screenshot of a web browser window displaying the PharoJS REPL interface. The browser's address bar shows "https://pharojs.org/repl". The page header includes the PharoJS logo and the text "Smalltalk REPL using PharoJS". A code editor area is highlighted with a red border, containing the following Smalltalk code snippet:

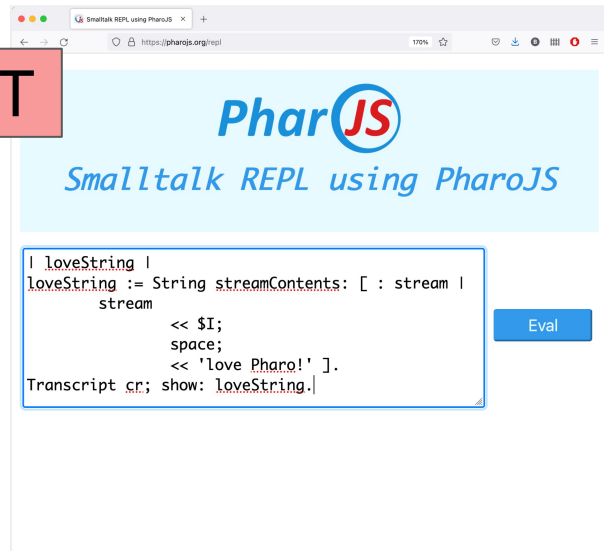
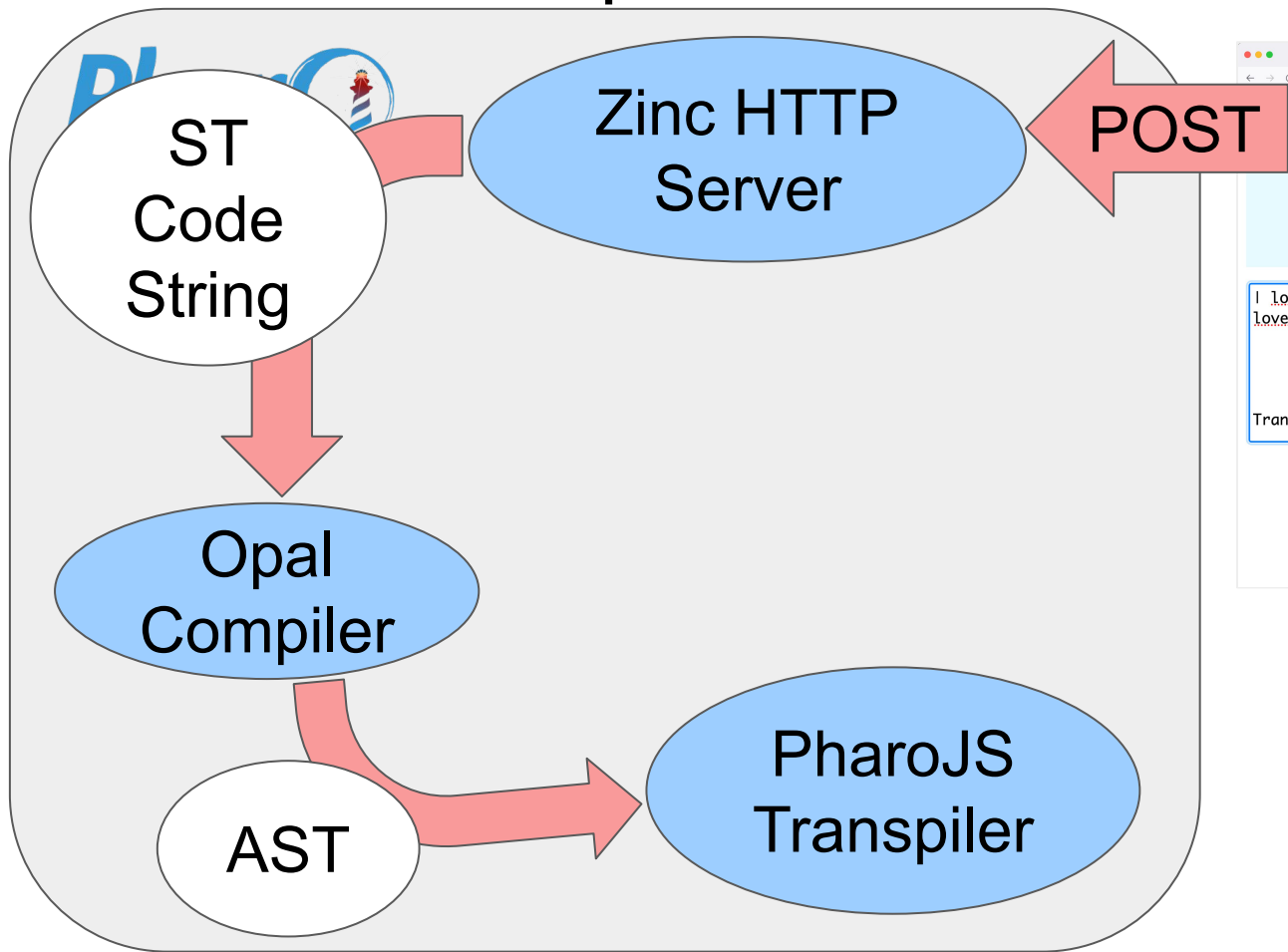
```
| loveString |
loveString := String streamContents: [ : stream |
    stream
        << $I;
        space;
        << 'love Pharo! ' ].
Transcript cr; show: loveString.
```

To the right of the code editor is a blue button labeled "Eval".

# Server Compiles ST Code Snippet

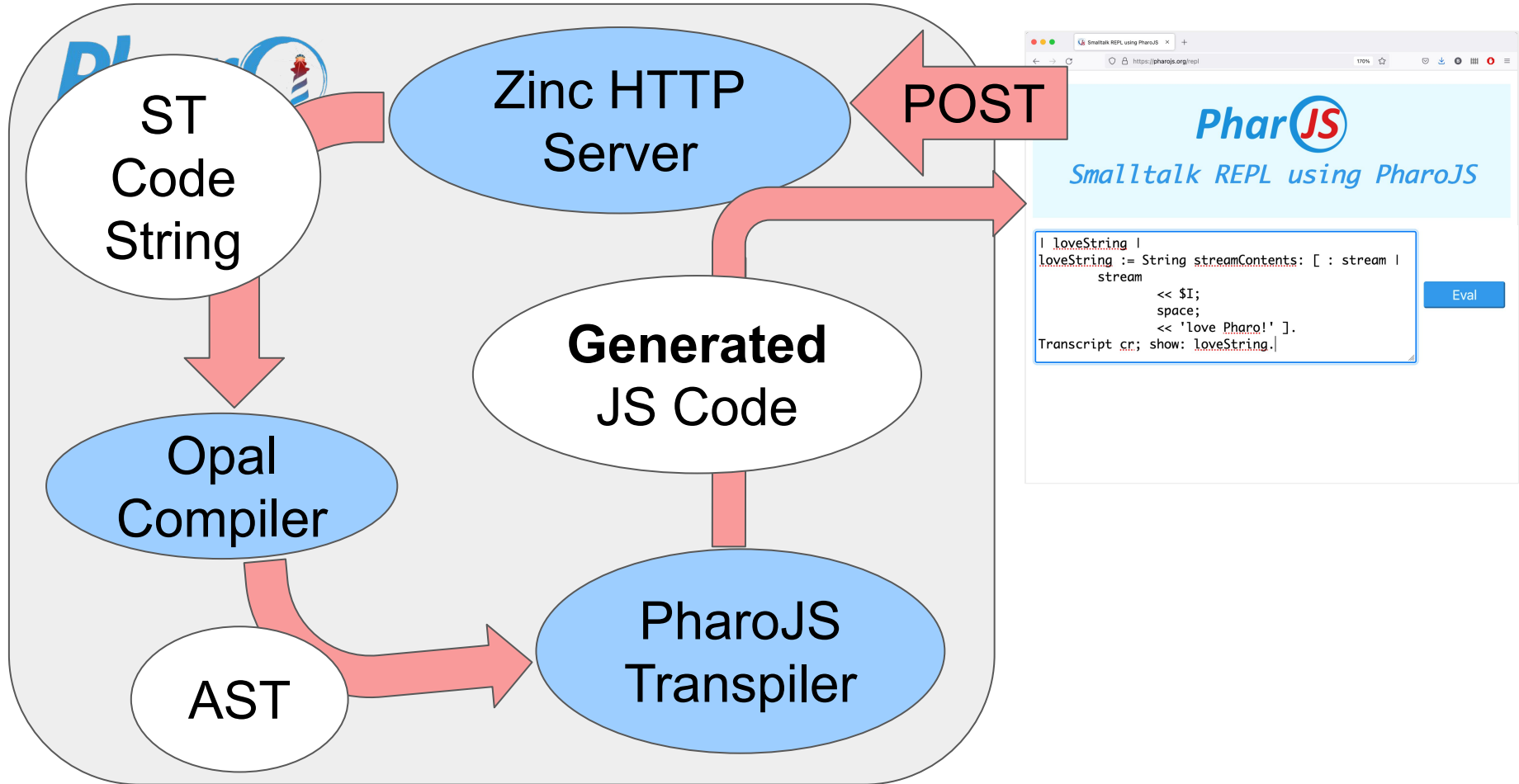


# PharoJS Transpiles AST





# Server Sends Generated JS



# Client Executes Generated JS Code



Zinc HTTP  
Server

A screenshot of a web browser window showing the PharoJS REPL interface. The browser's address bar shows the URL "https://pharojs.org/repl". The page has a light blue header with the "PharoJS" logo and the text "Smalltalk REPL using PharoJS". Below the header, there is a code editor containing Smalltalk code:

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

The code is followed by a "Transcript" section that displays the output "I love Pharo!". Below the transcript is a text input field with the placeholder text "Your Smalltalk code here" and a blue "Eval" button. A red rectangular box highlights the entire bottom section of the interface, including the transcript and the input field.

# Small App

Smalltalk REPL using PharoJS

PharoJS

Smalltalk REPL using PharoJS

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

Transcript

I love Pharo!

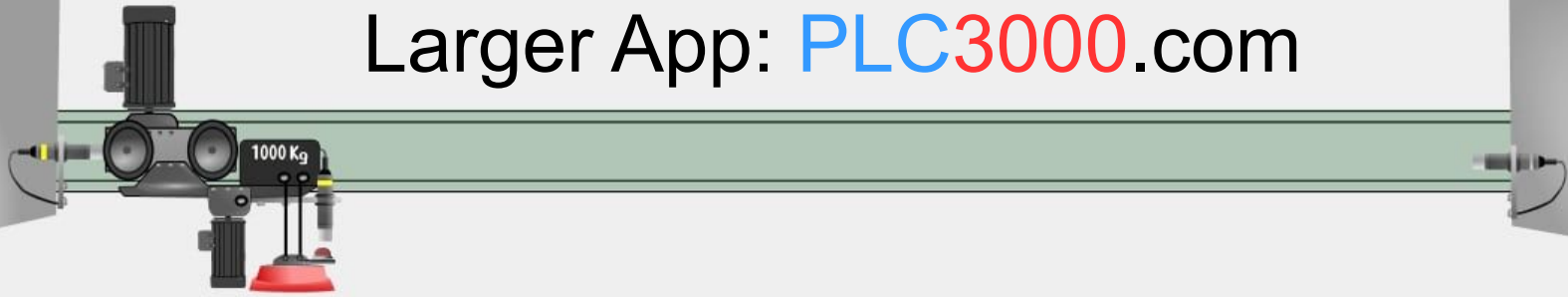
Your Smalltalk code here

Eval

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

Generated JS  
**267KB (+3KB)**

Larger App: [PLC3000.com](http://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

39

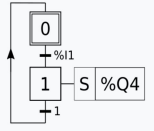
## AUTOMATION OF AN EL IN CONTROL OF A CROSSROAD - TIMER, COUNTER

### 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 Grafset with 2 steps:



According to this Grafset, 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 Grafset in IL with the method Transition/Transition without *Crossing Bits* is:

### Code

```
Ladder ST IL Run Export Load
```

```
LD S2 %M1
OR %M2
AND NOT %M3
LD %M4
LD %M1
OR %M2
AND %M3
LD %M4
OR %M1
AND %M2
LD %M4
OR %M1
AND %M2
LD %M4
OR %M1
AND %M2
```

### PLC

Program Running

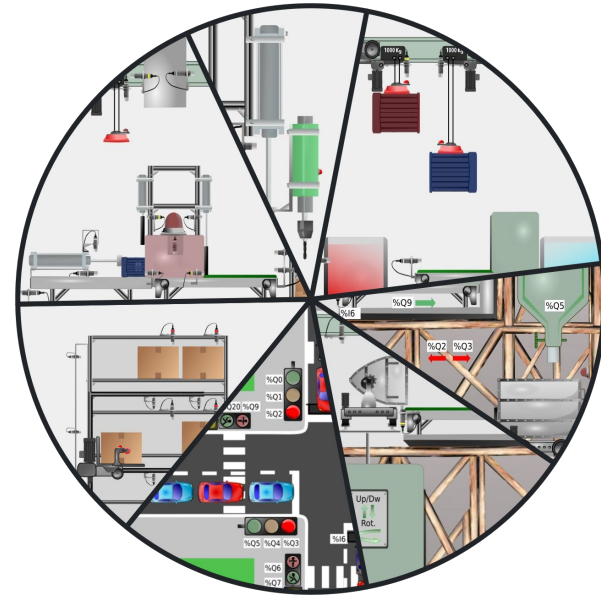
System Bits: S0 S1 S2

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



27 (13 + 14)  
Exercises  
& Tutorials

3  
Programming  
Languages

PLC  
Simulator

7 (4+3)  
Physics  
Simulations

# PLC3000.com = Educational Software + Contents

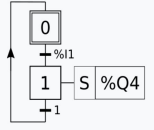
**AUTOMATION OF AN**  
**EL** INSTRUCTIONS  
**CONTROL OF A CROSSROAD - TIMER, COUNTER**

**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 Grafset with 2 steps:



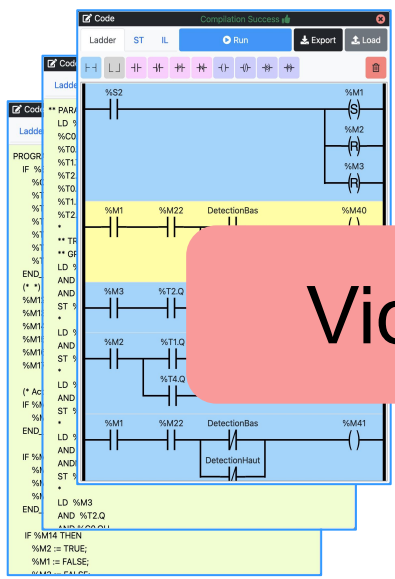
According to this Grafset, as soon as %I1 is true, the step 1 has to be activated causing the forcing of %Q4 to 1.

**Wrong Code**

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

**Code** Compilation Subjects

Ladder ST IL Run Export Load



**PLC** Program Running

System Bits: S0 S1 S2

Memory: M0 M1 M2 M3 M4 M5 M6 M7 M8

IS I6 I7

Q6 Q7 Q8 Q9  
Q16 Q17 Q18 Q19  
Q23

Counters:

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

Time:

**Video**



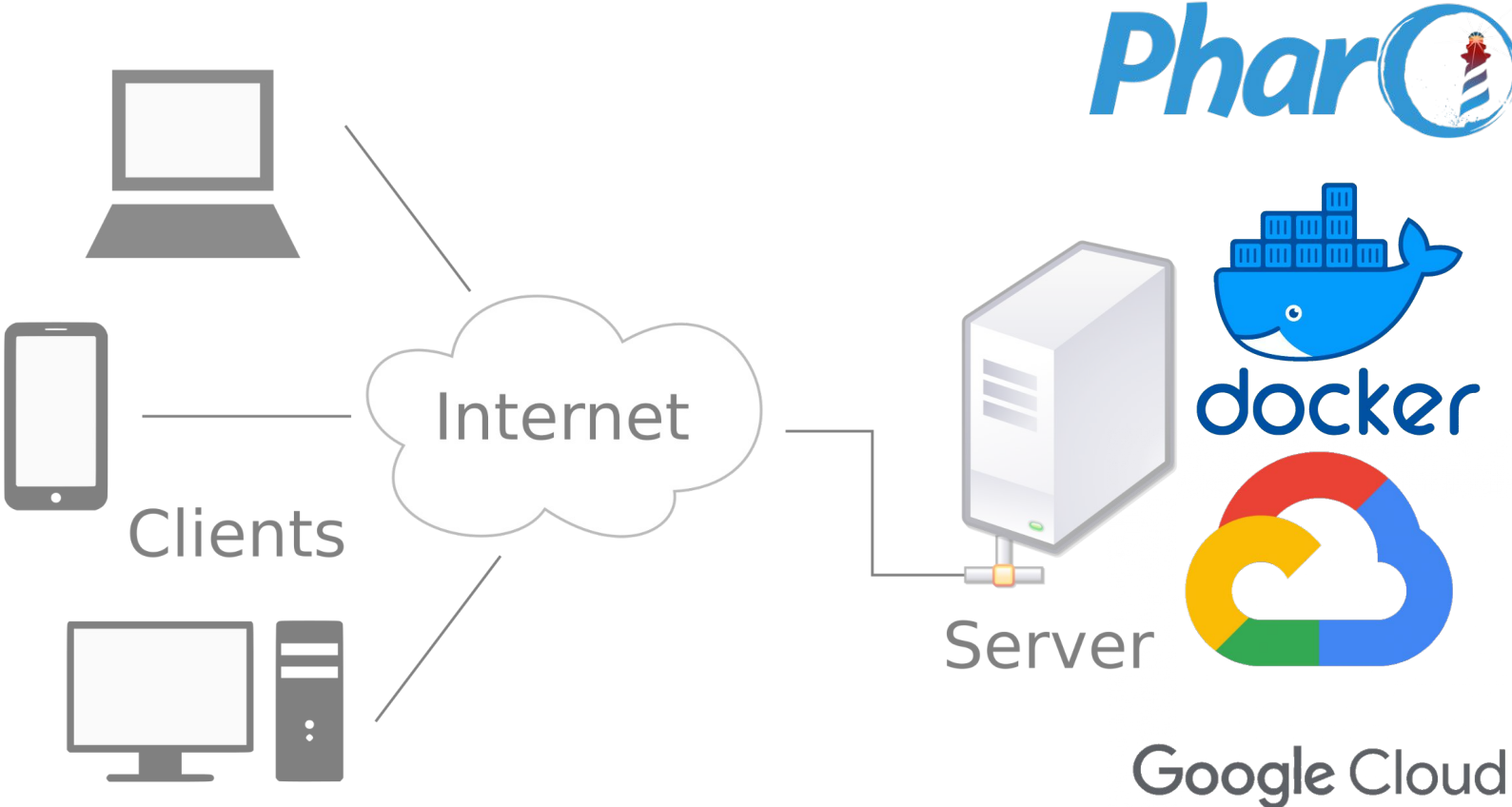
**27 (13 + 14)  
Exercises  
& Tutorials**

**3  
Programming  
Languages**

**PLC  
Simulator**

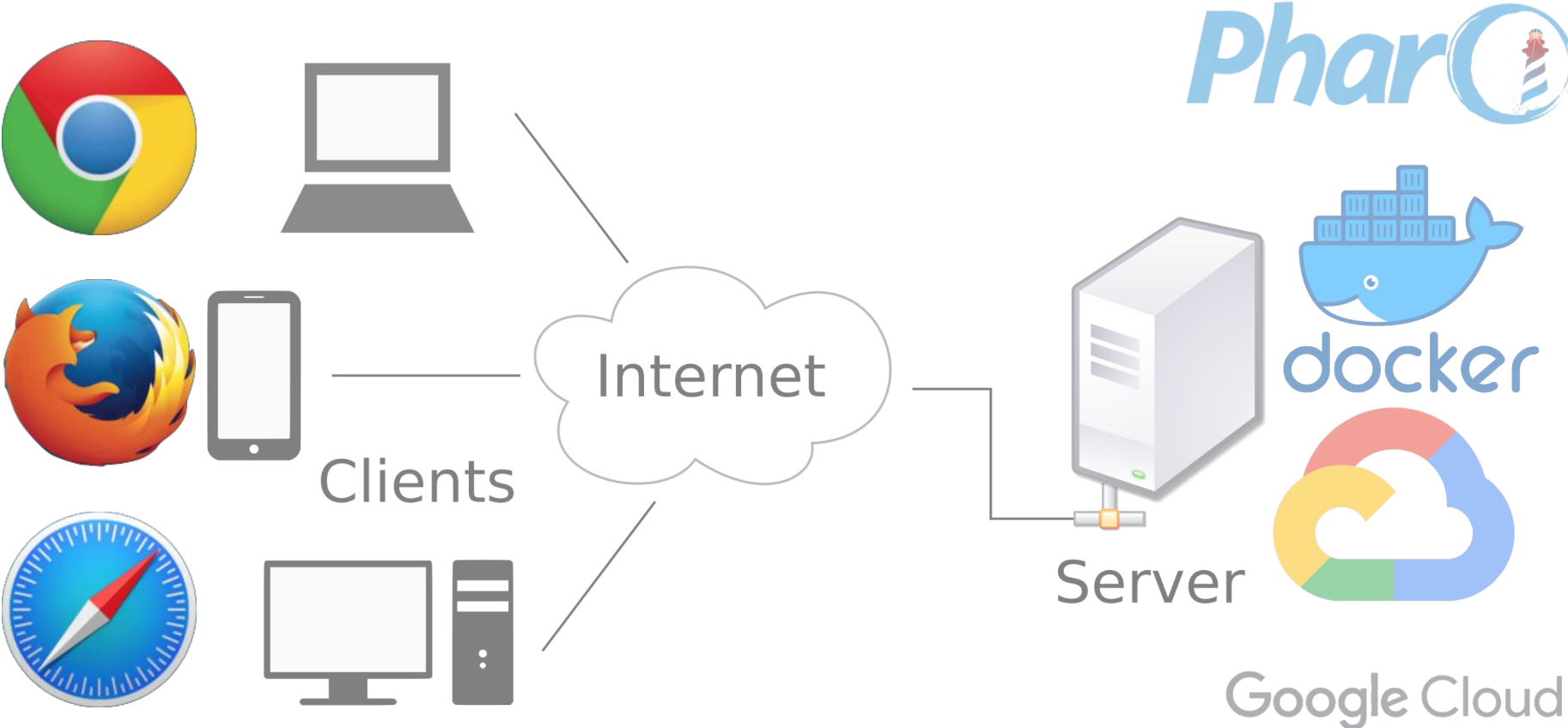
**7 (4+3)  
Physics  
Simulations**

# PLC3000.com Server Side

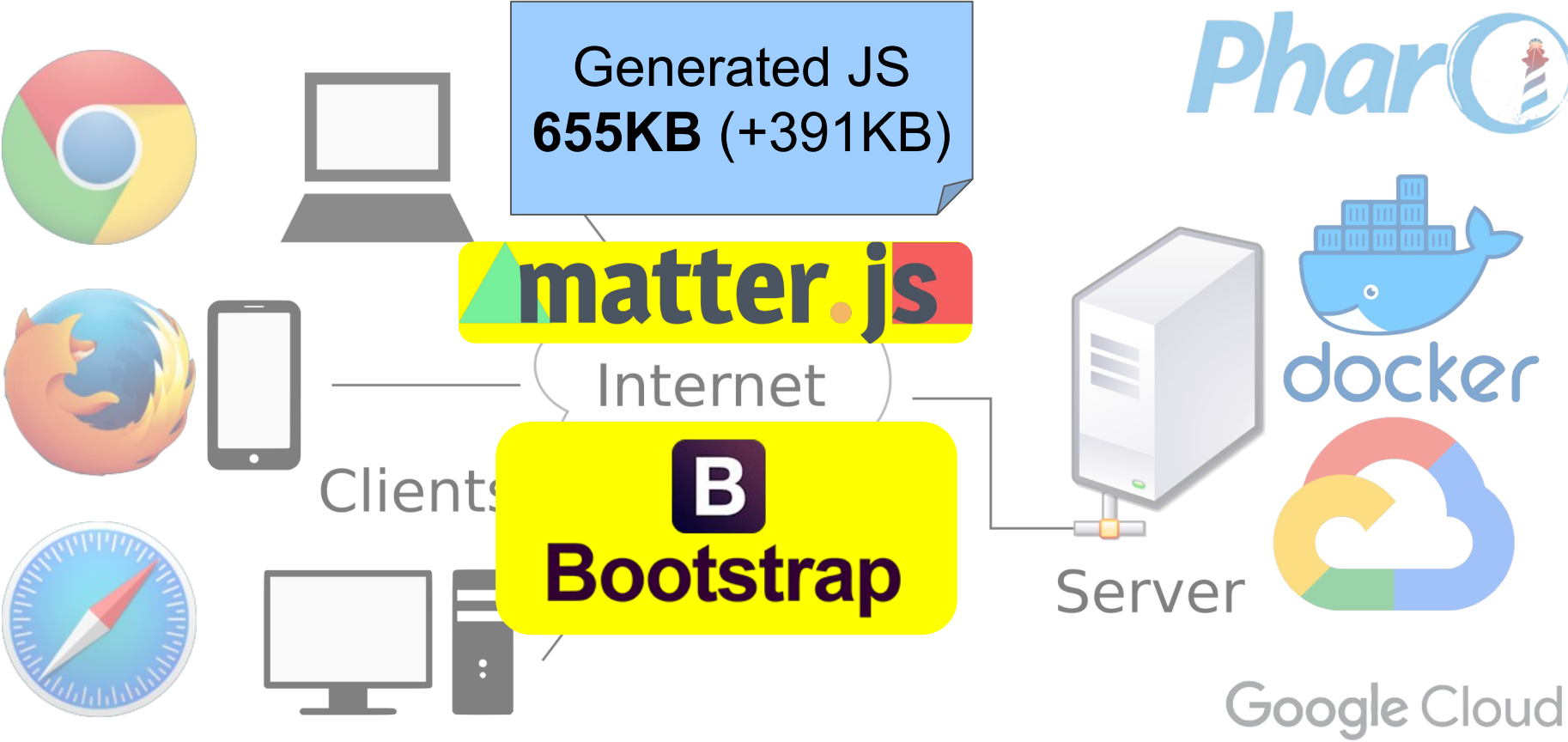




# PLC3000.com Clients Run in Web Browsers



# PLC3000.com Client JS Code

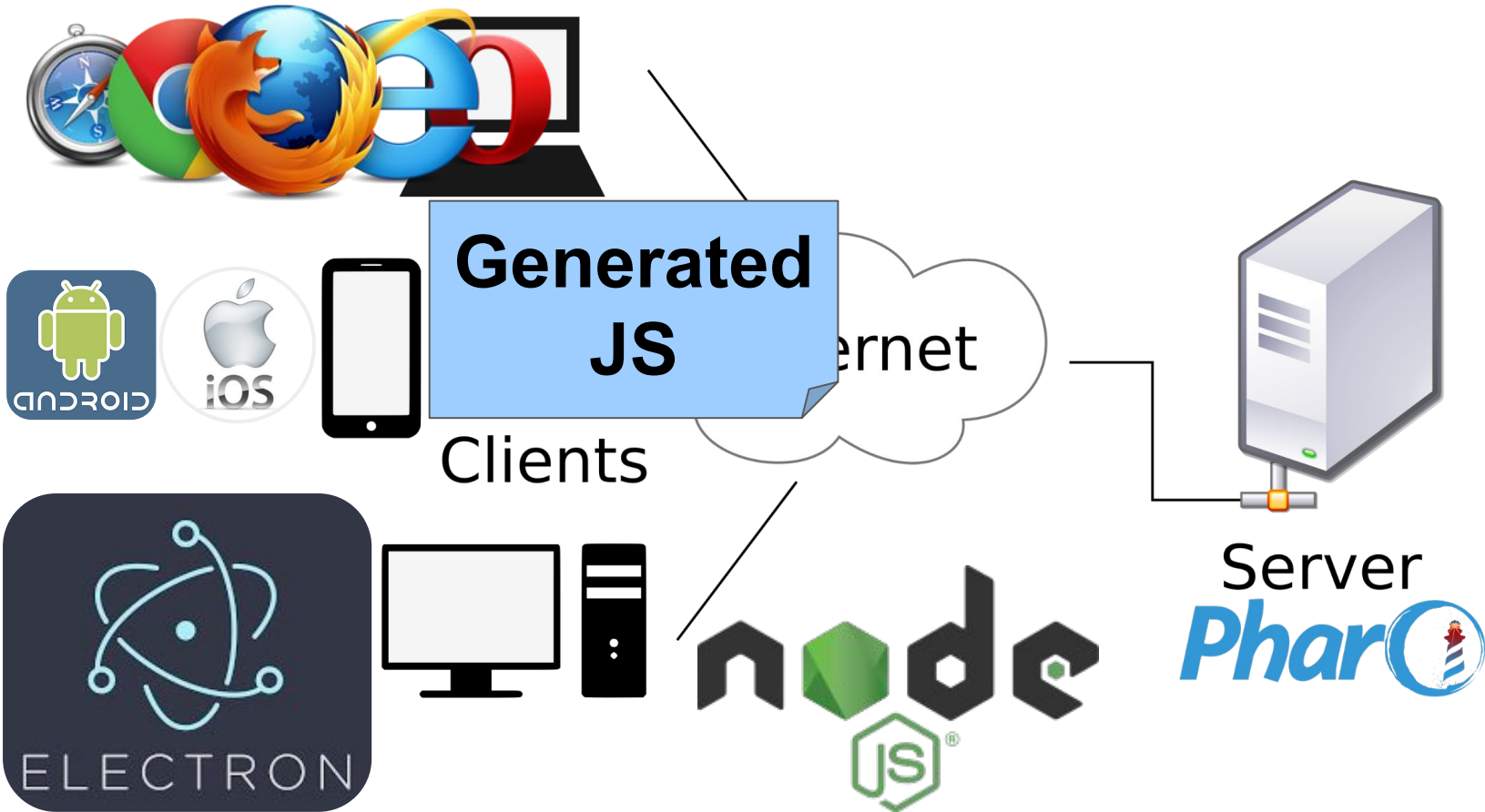


# Summary

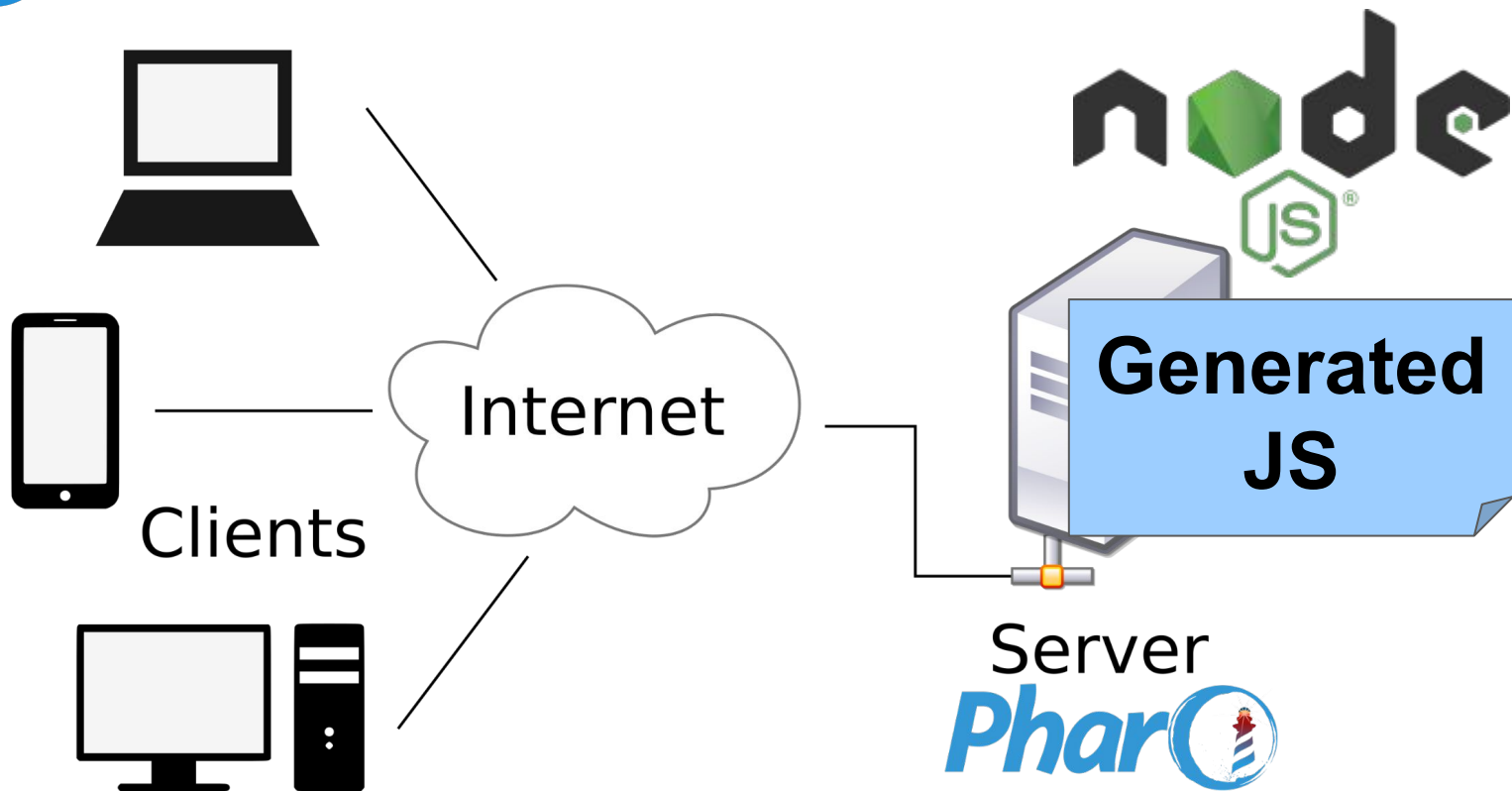
## *Phar* Supports Client-Server Apps

- 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 = *Phar*JS



# PharJS is for Server Side Too!



# PharJS Supports Different Workflows

Run-Time vs Development-Time

- HTML, CSS

- Handwritten Files ●
- Generated ● ●
- DOM Elements Creation & Setup ●
- Reuse Third-Party Libraries ●

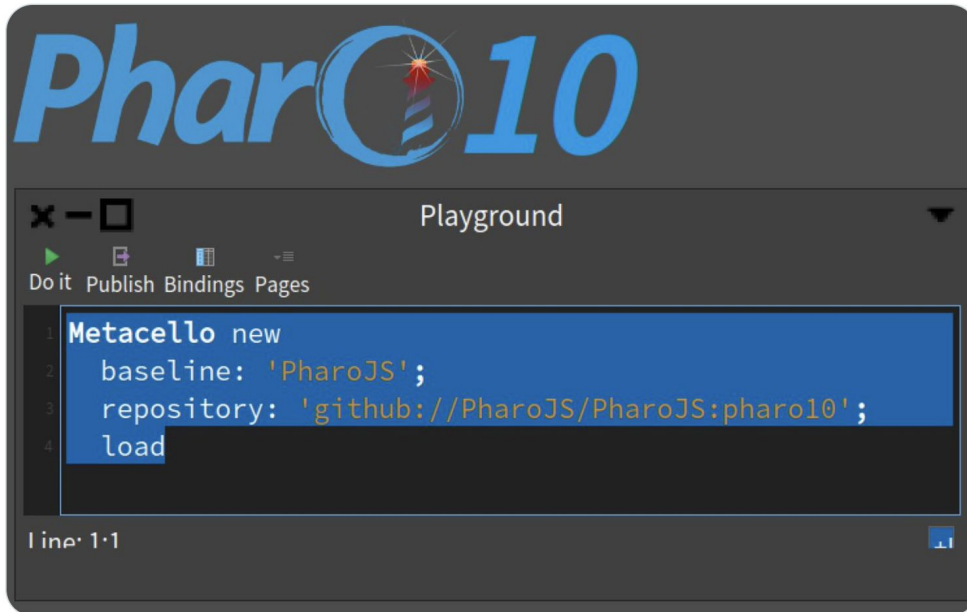
- Javascript

- Generated ● ●
- Reuse Third-Party Libraries ●

 Tweet

**Noury Bouraqadi**  
@nourybouraqadi

PharoJS for Pharo 10 is out ! [github.com/PharoJS/PharoJ...](https://github.com/PharoJS/PharoJS) @pharoproject @pharojs #SmallTalk #javascript



**Phar**  
for  
Pharo X  
since  
Spring 2021 ;-)

**April 1st, 2021**

# PharoJS Future Development

- Pharo X Support
- 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



Thanks to all the contributors

