Compiling the world to WebAssembly

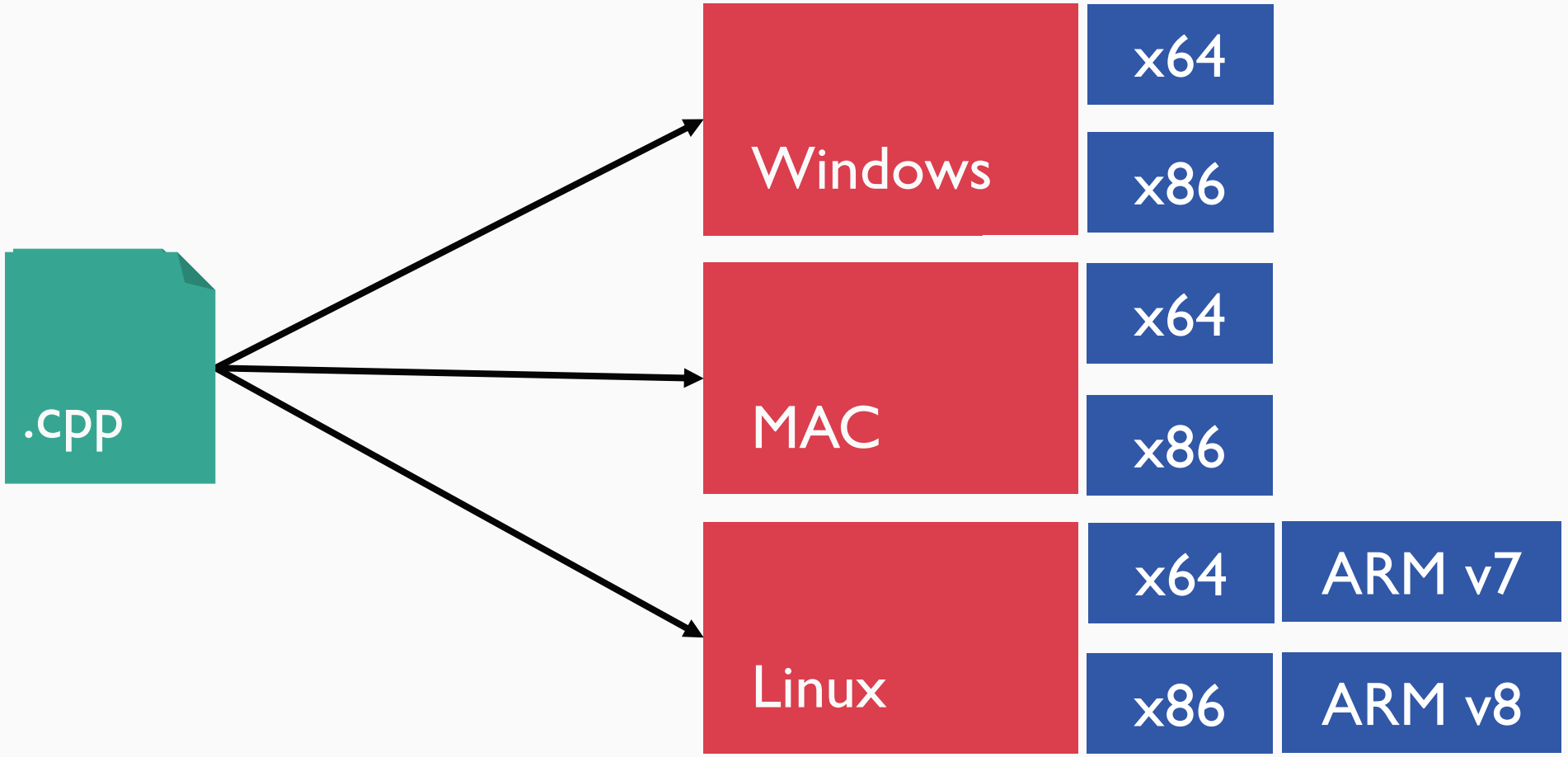
Boyan Mihaylov
Software architect and developer

@boyanio
https://boyan.io

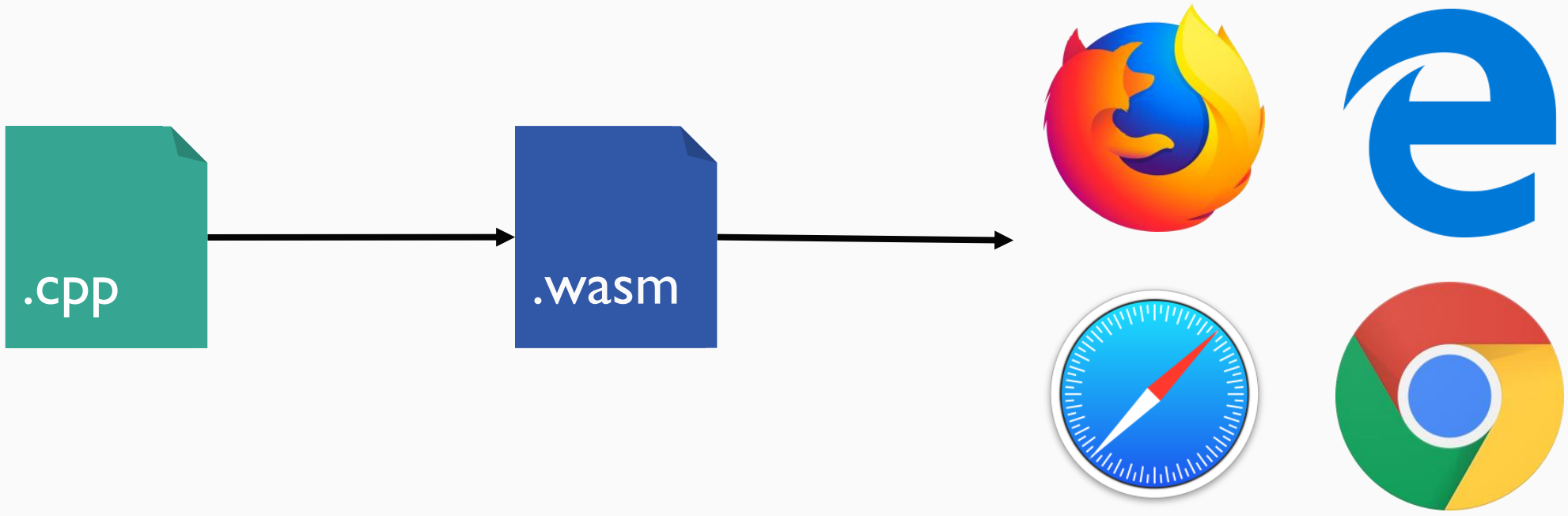
WebAssembly (*WASM*) is compiler target for programs on the web

```
C:\wasm>type index.c
#include <stdio.h>

int main(void) {
        printf("Hello, cool people!\n");
        return 0;
}
C:\wasm>clang index.c

C:\wasm>a.exe
Hello, cool people!

C:\wasm>emcc -o a.js index.c

C:\wasm>node a.js
Hello, cool people!
```

Traditional multi-target compilation



Multi-target compilation with WebAssembly



```
function add(a, b) {
  return a + b;
}
```

```
> add(2, 3)
< 5
> add("a", 5)
< "a5"
> add("a", null)
< "anull"
> add(5, {})
< "5[object Object]"
> add({}, "a")
< "[object Object]a"
> add("a")
< "aundefined"
```

```
> '7' - 3
< 4
```

weak typing, implicit conversion

```
> '7' + 3
< "73"
```

…not really consistent

```
> '7' - '3'
< 4
```

string – string = number ?

```
> 7 + '3'
< "73"
```

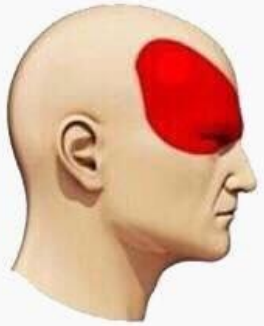
“+” is for concatenation

```
> 7 + + '3'
< 10
```

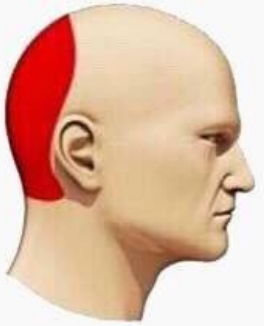
“+ +” is for addition ?

Types of Headaches

Migraine



Hypertension



Stress

JavaScript

@boyanio

WebAssembly is a typed language

It supports 32 and 64-bit integers (i32, i64)
and floating points (f32, f64)

Binary representation (.wasm)

```
0061 736d 0100 0000 0187 8080 8000 0160
027f 7f01 7f03 8280 8080 0001 0004 8480
8080 0001 7000 0005 8380 8080 0001 0001
0681 8080 8000 0007 9080 8080 0002 066d
656d 6f72 7902 0003 6164 6400 000a 8d80
8080 0001 8780 8080 0000 2001 2000 6a0b
```

Textual representation (.wat)

```
(module
  (memory $0 1)
  (export "add" (func $add))
  (func $add (param $0 i32) (param $1 i32) (result i32)
    (i32.add
      (get_local $1)
      (get_local $0)
    )
  )
)
```

WebAssembly provides consistent and predictable performance

16 Anim (2-214)

System WebAssembly

NumRunners 15

Close Controls

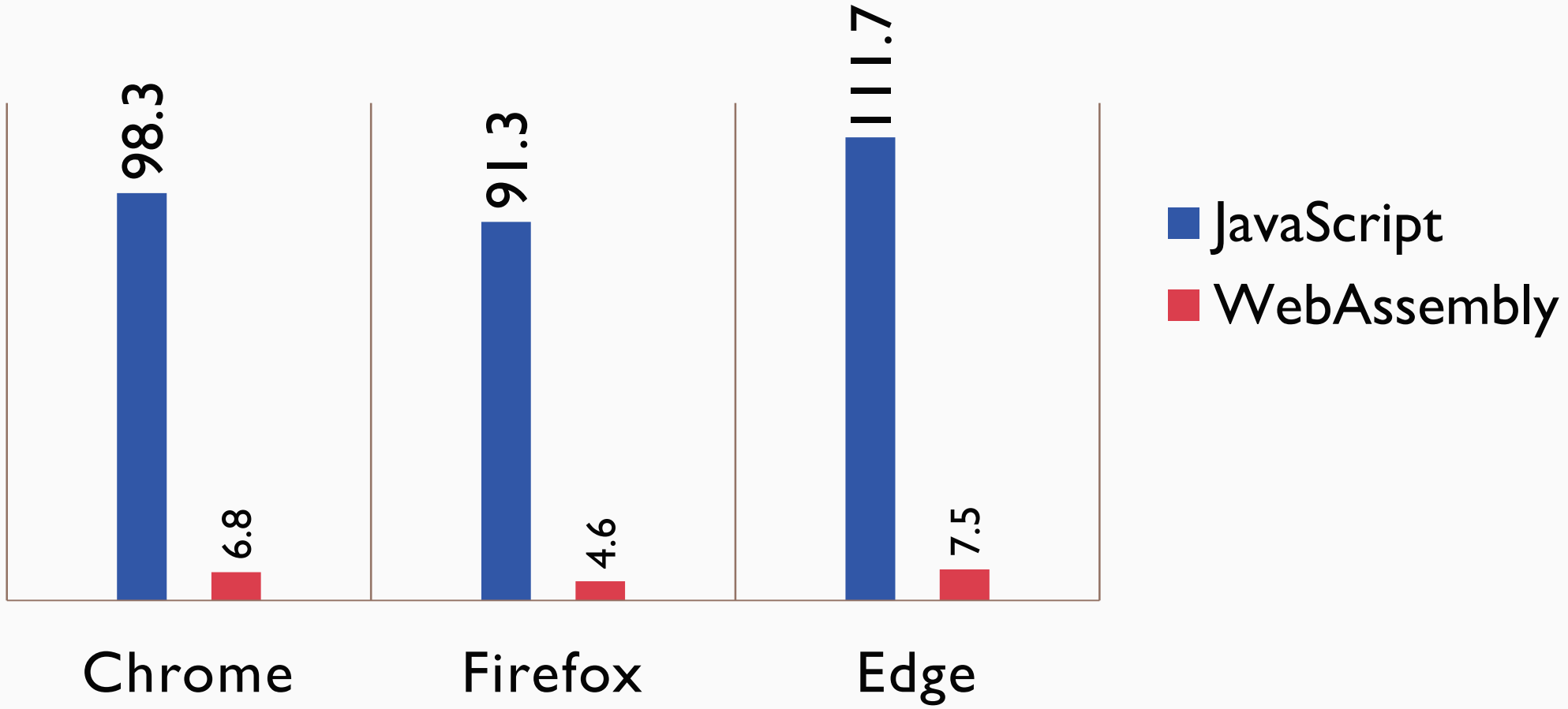
3D animation performance

https://github.com/sessamekesh/wasm-3d-animation-demo

@boyanio

Performance comparison

average animation time (ms)



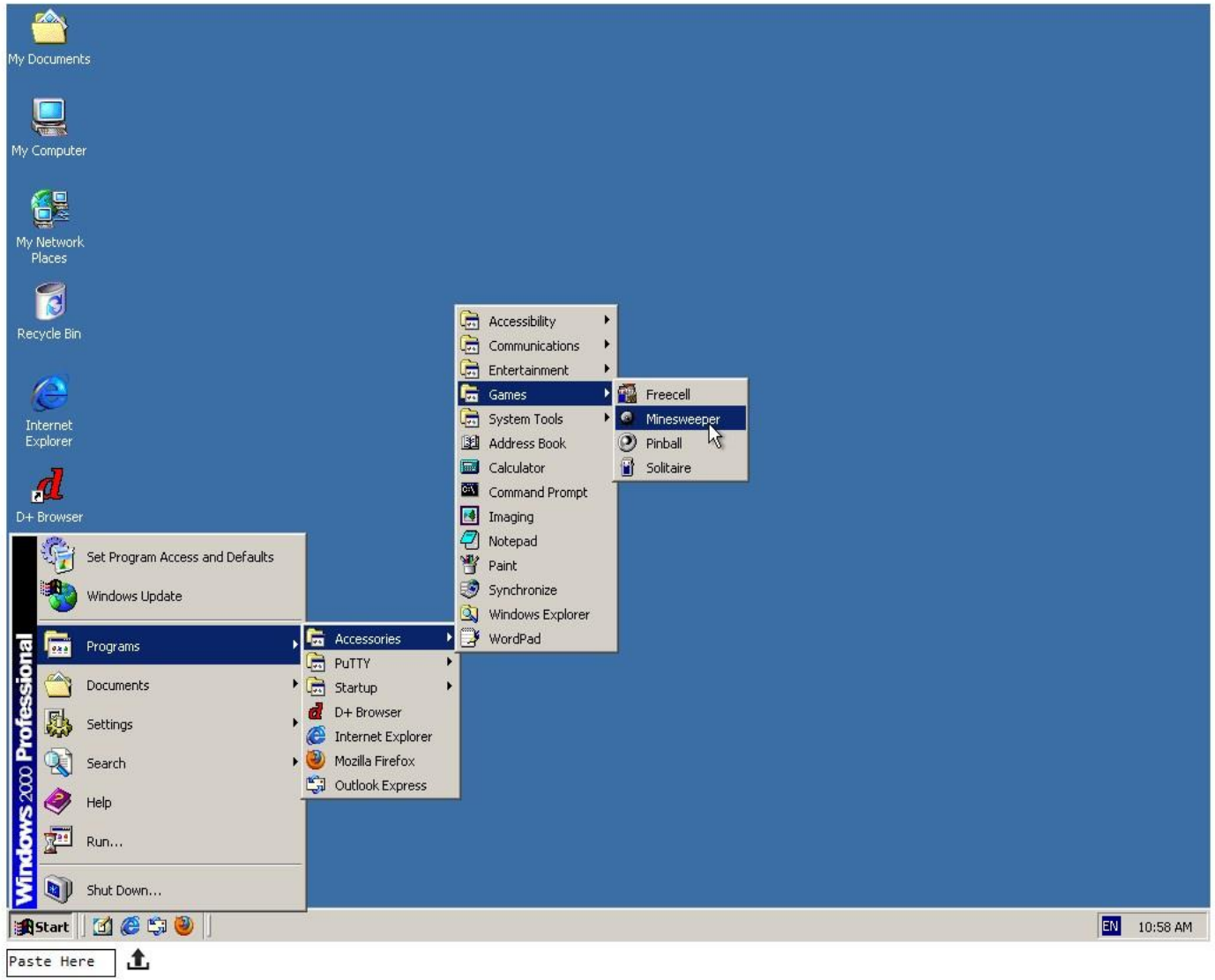
https://www.lucidchart.com/techblog/2017/05/16/webassembly-overview-so-fast-so-fun-sorta-difficult/

The Adobe Flash plugin has crashed.
Send crash report

Reusing code on the web

@boyanio

What can we do with WebAssembly?



Windows 2000 in the browser

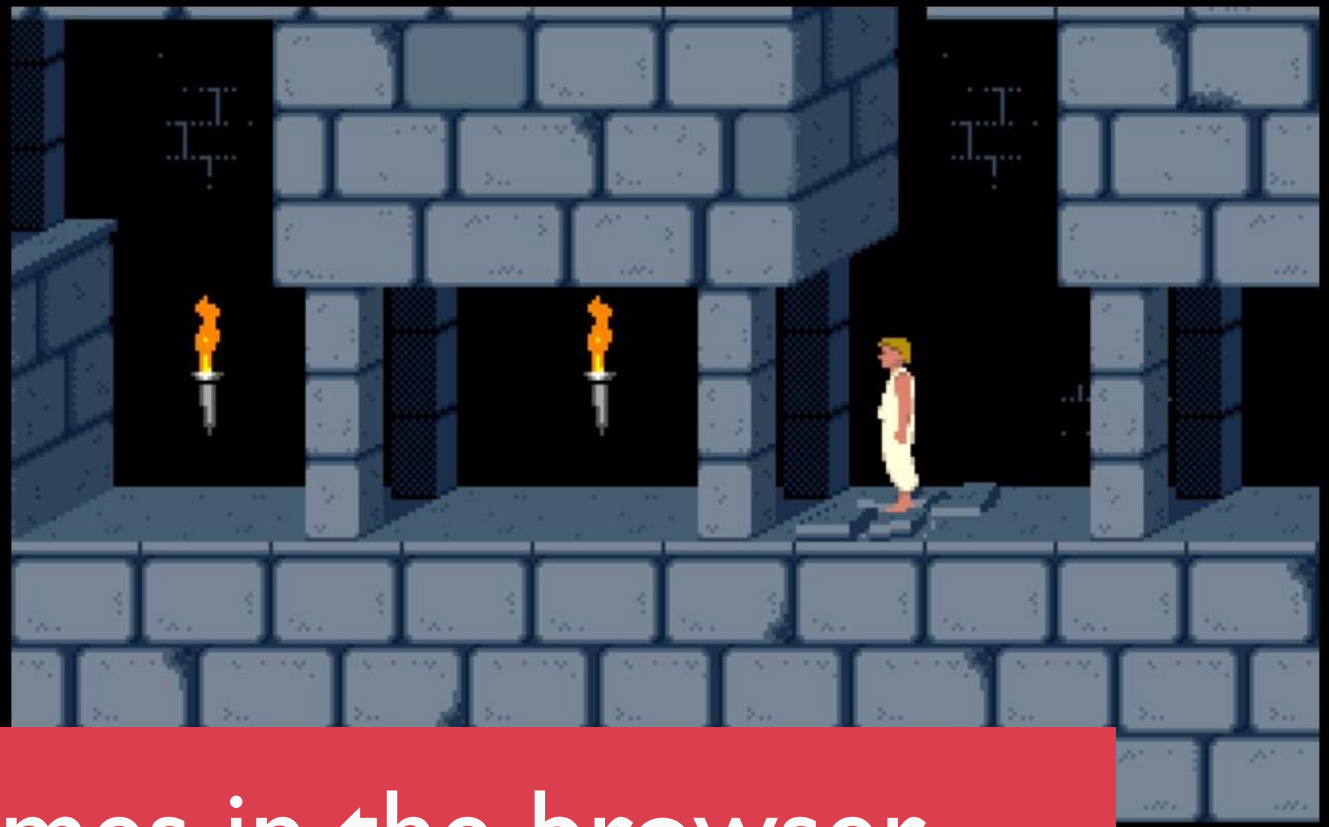
https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/win2k.cfg&mem=192&graphic=1&w=1024&h=768

DOSBox SVN, CPU spee

https://archive.org/details/msdos_Prince_of_Persia_1990#

SIGN IN

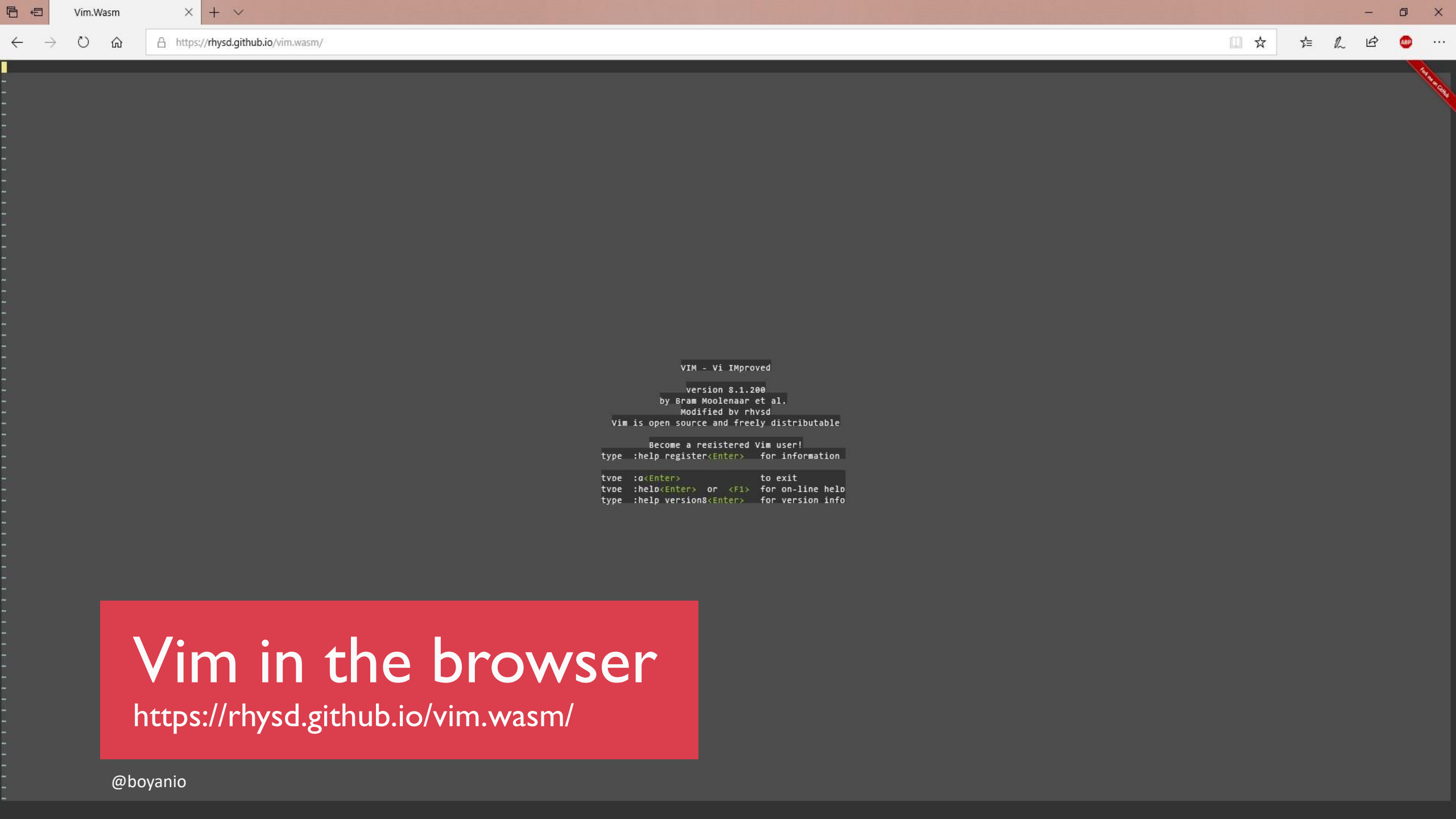
ABOUT CONTACT BLOG PROJECTS HELP DONATE JOBS VOLUNTEER PEOPLE



DOS games in the browser

https://archive.org/details/msdos_Prince_of_Persia_1990

by Jul 29, 2014


Vim.Wasm
https://rhysd.github.io/vim.wasm/
VIM - Vi IMproved
version 8.1.200
by Bram Moolenaar et al.
Modified by rhysd
Vim is open source and freely distributable
Become a registered Vim user!
type :help register<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info
Vim in the browser
https://rhysd.github.io/vim.wasm/
@boyanio

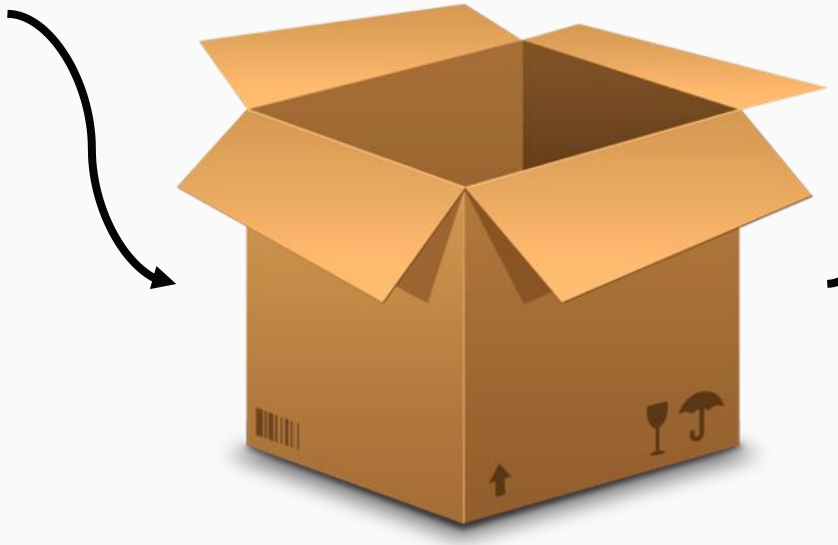
How to get started with WebAssembly


WA
https://wasdk.github.io/WasmFiddle/?
C
Build
Run
JS
int add(int a, int b) {
return a + b;
}
var wasmModule = new WebAssembly.Module(wasmCode);
var wasmInstance = new WebAssembly.Instance(wasmModule, wasmImports);
log(wasmInstance.exports.main());
Text Format
Wast
Wasm
Output
Canvas
Clear
(module
(table 0 anyfunc)
(memory $0 1)
(export "memory" (memory $0))
(export "add" (func $add))
(func $add (; 0 ;) (param $0 i32) (param $1 i32) (result i32)
(i32.add
(get_local
(get_local
)
)
)
Wasm Fiddle
https://wasdk.github.io/WasmFiddle/
@boyanio

Module imports & exports

```
const imports = {
  "name": {
    "first": "Anna",
    "last": "Nanna"
  },
  "print": what => {
    console.log(what);
  }
};
```

```
exports.add(1, 4);
exports.print();
```



Open Source LLVM to JavaScript compiler

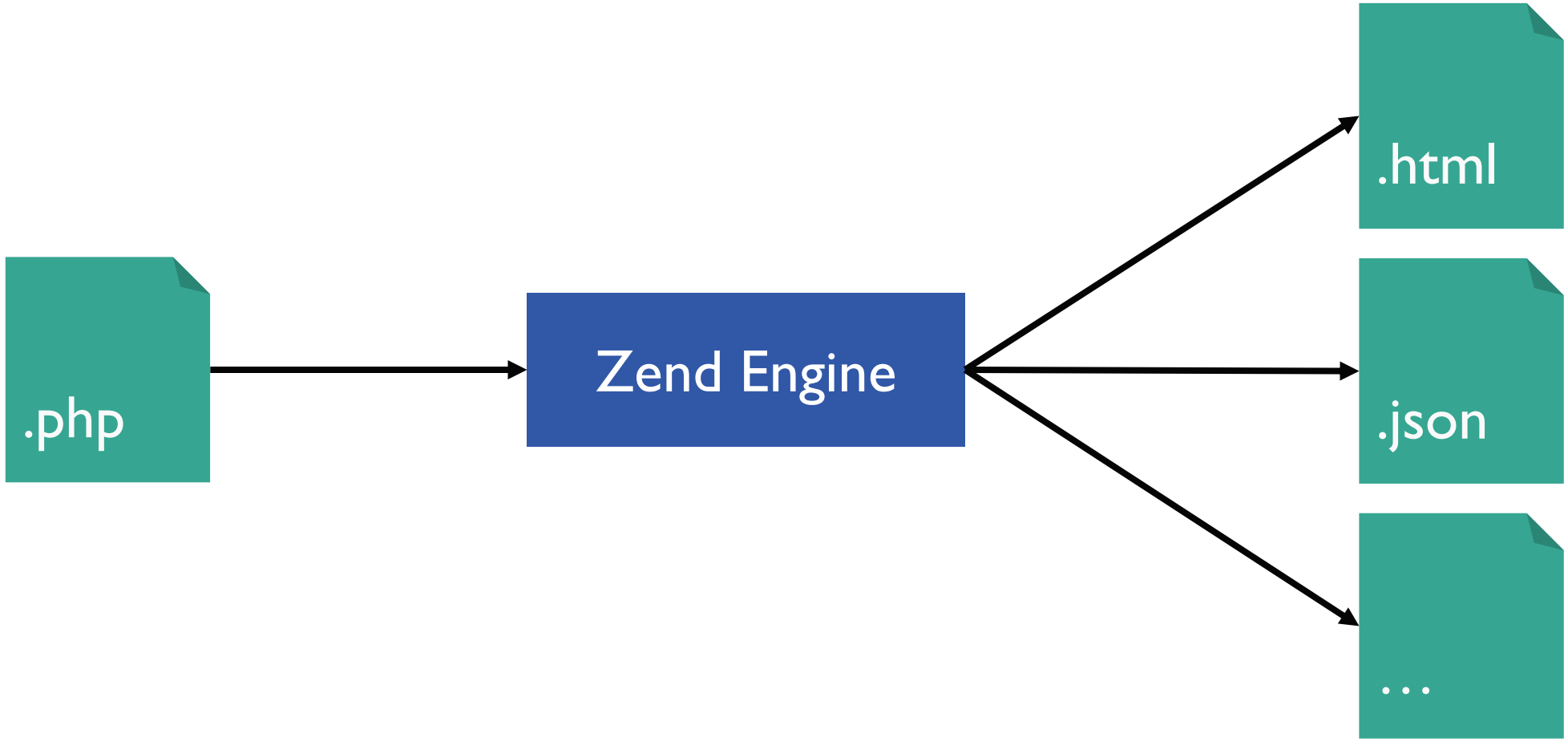
PHP and WebAssembly?

Compile PHP interpreter to WebAssembly

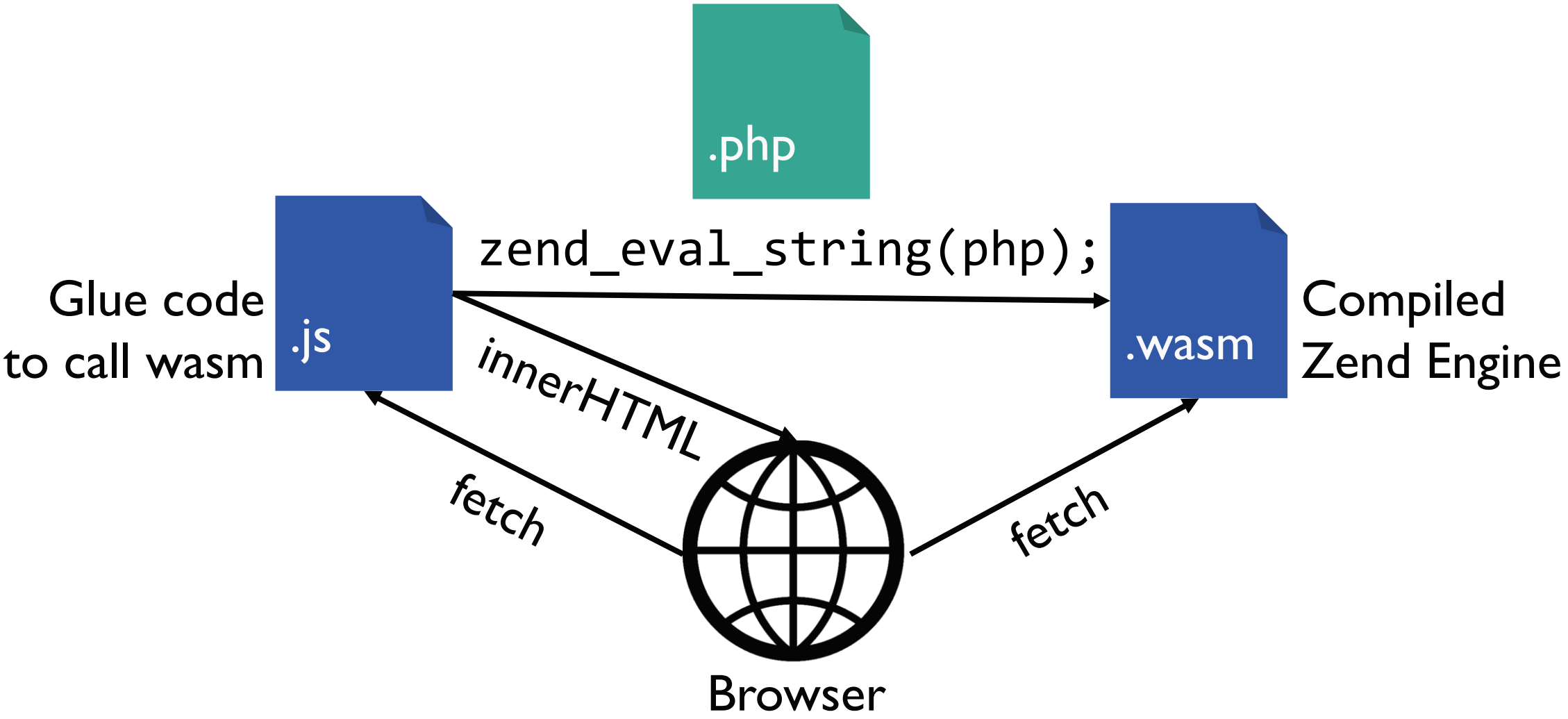
1/3

@boyanio

PHP → Zend inside a web server



PHP → Zend compiled to WebAssembly



RUN

Fork me on GitHub

```php
<?php

phpinfo();
```

PHP Version 7.4.0-dev

System	Emscripten emscripten 1.0 #1 x86-JS
Build Date	Sep 16 2019 18:12:58
Configure Command	'./configure' '--disable-all' '--disable-cgi' '--disable-cli' '--disable-rpath' '--disable-ph without-pear' '--without-pcre-jit' '--with-layout=GNU' '--enable-embed=static' '--en enable-ctype' '--enable-mbstring' '--disable-mbregex' '--enable-tokenizer' 'PKG_C 'PKG_CONFIG_LIBDIR=/home/lyp/fun/emsdk/emscripten/incoming/system/loca /emscripten/incoming/system/lib/pkgconfig'
Server API	PHP Embedded Library
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
d	API20190902,NTS
	no
	disabled
ng	enabled
ger	disabled
port	provided by mbstring
IPv6 Support	enabled

PHP in the browser

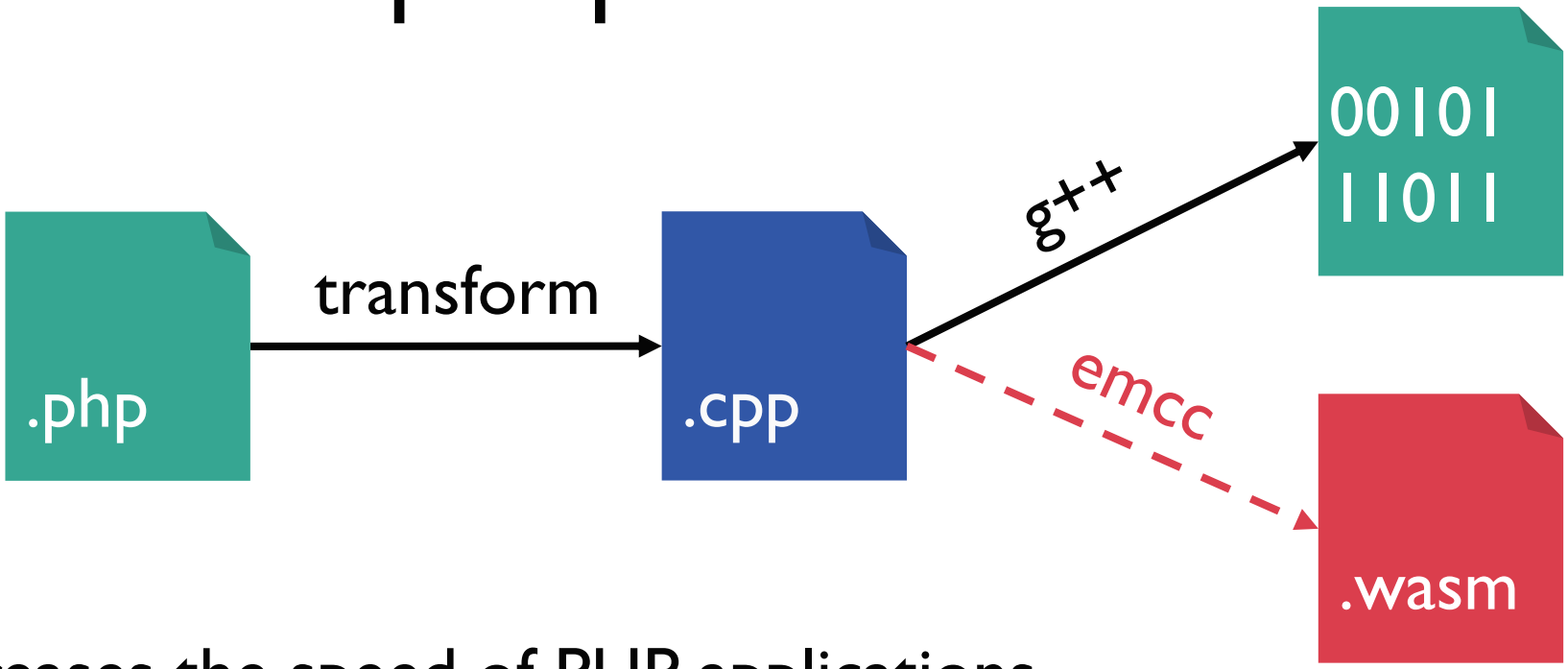
https://oraoto.github.io/pib/

@boyanio

Compile PHP to WebAssembly

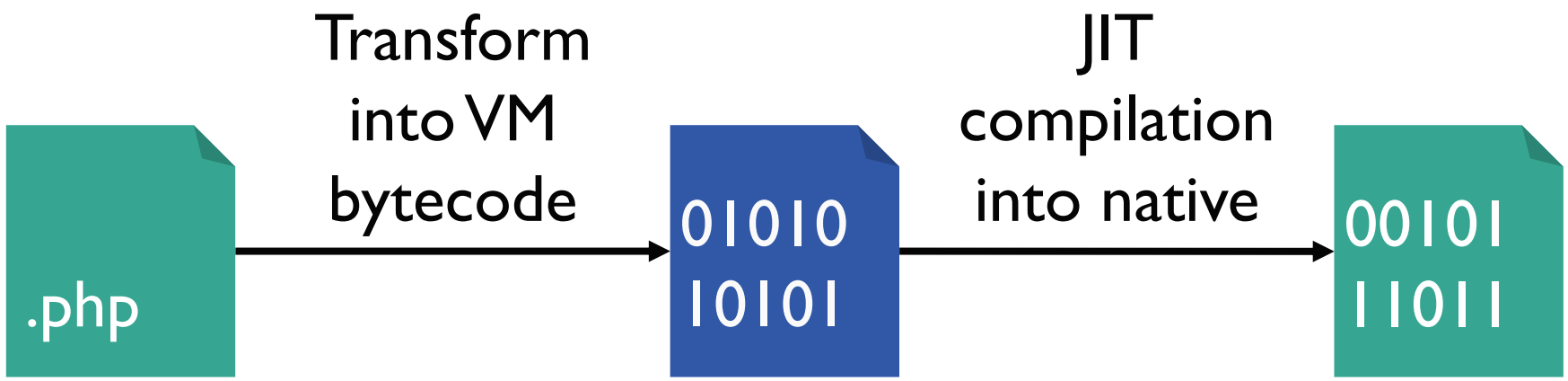
2/3

Facebook: HipHop for PHP



- Increases the speed of PHP applications
- Uses type inference to find the best variable type
- No support for `eval()` and `create_function()`
- Discontinued in 2013

Facebook: HipHop Virtual Machine (HHVM)



- Virtual machine
- Successor of HipHop for Facebook
- Uses just-in-time compilation (JIT)
- Should be used together with a web server

Other (discontinued) PHP compilers

PeachPie
PHP → .NET

Roadsend
PHP → native

Phalanger
PHP → .NET

BinaryPHP
PHP → C++

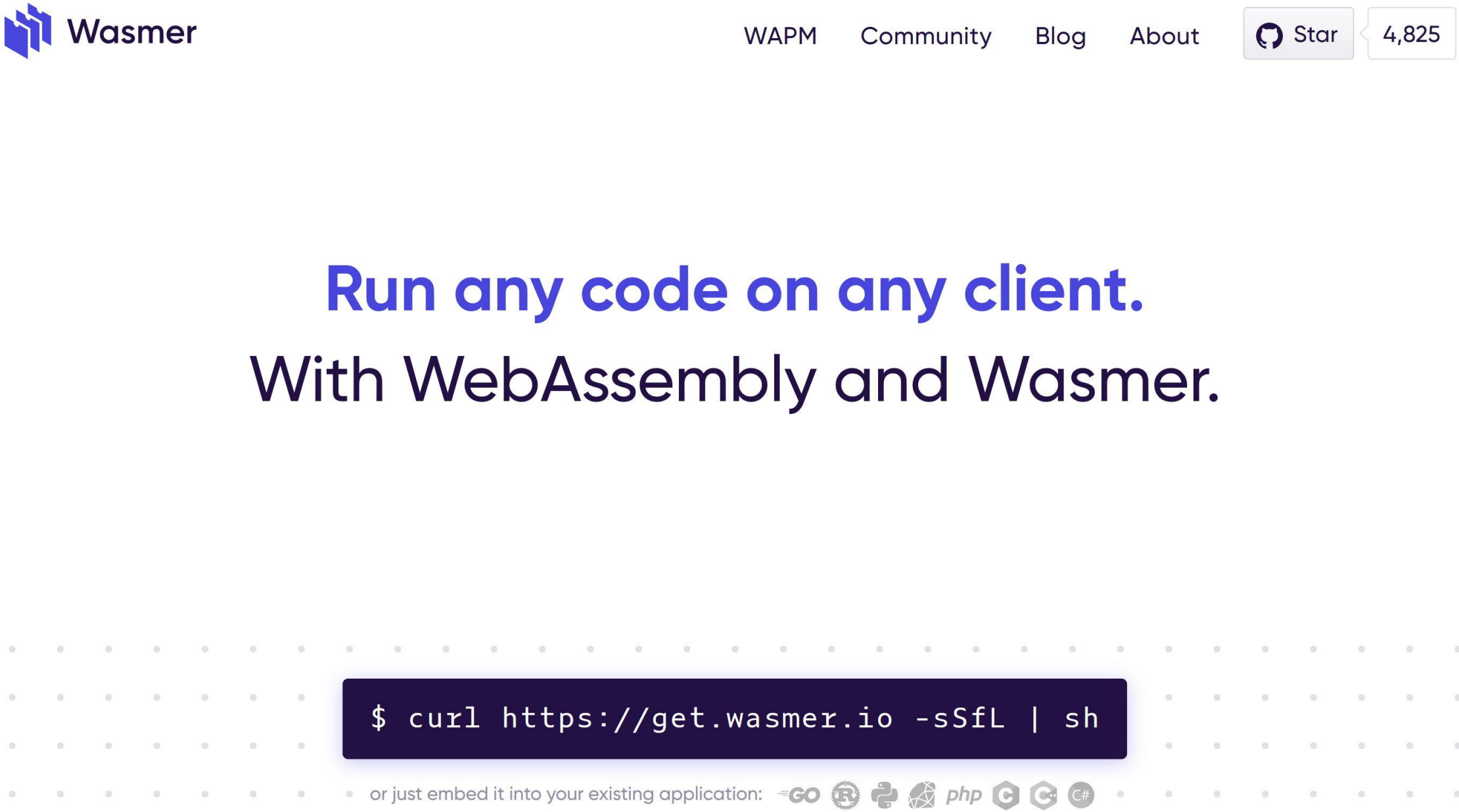
phc
PHP → native

Project Zero
PHP → Java bytecode

https://stackoverflow.com/questions/1408417/can-you-compile-php-code-and-upload-a-binary-ish-file-which-will-just-be-run

Consume WebAssembly in PHP

3/3


Wasmer
WAPM
Community
Blog
About
Star
4,825
Run any code on any client.
With WebAssembly and Wasmer.
$ curl https://get.wasmer.io -sSfL | sh
or just embed it into your existing application:
GO
php
C#

@boyanio

Consume C in PHP via WebAssembly

```
// 1) add.c
int add(int a, int b) {
  return a + b;
}

// 2) Compile add.c to add.wasm

// 3) add.php
$module = new Wasm\Module(__DIR__ . 'add.wasm');
$instance = $module->instantiate();
var_dump($instance->add(2, 3)); // int(5)
```

Sudoku Battle

Solving sudoku puzzles using pure PHP and WebAssembly inside PHP.

PHP

Solve

WebAssembly

Solve

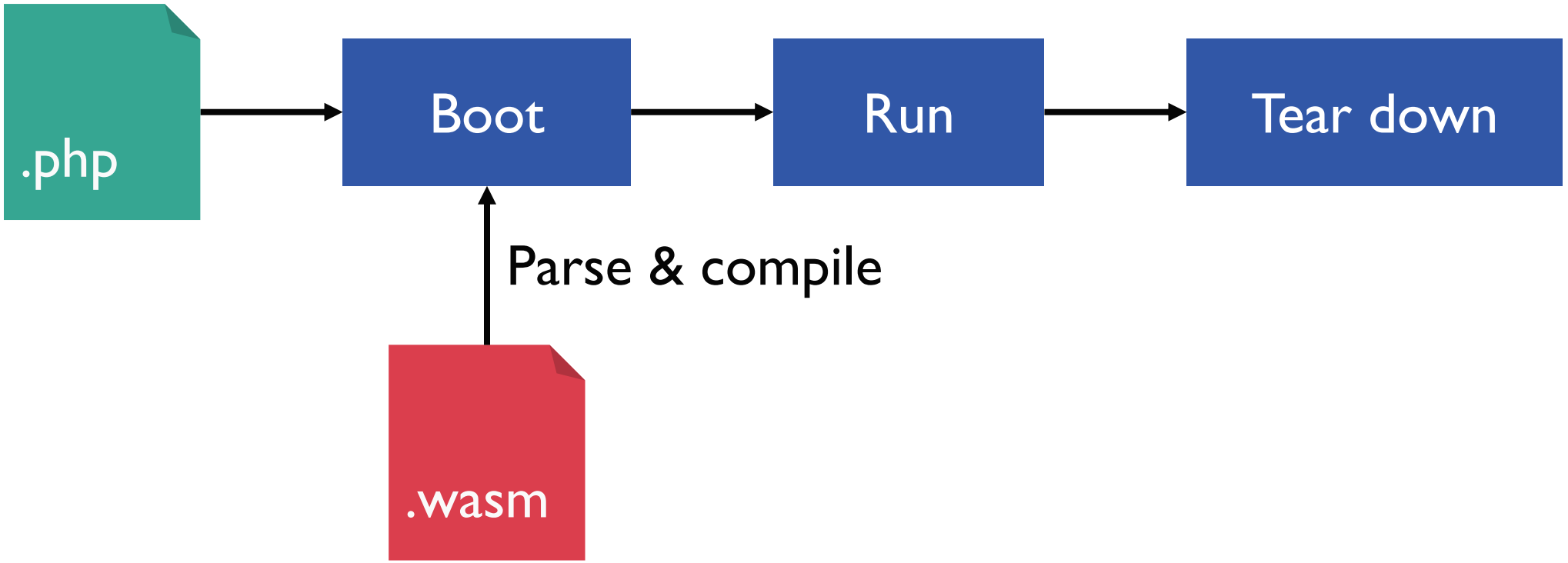
				4	8			
2	6		9					
					6			
8		2	4					
			2			3		
						5		
6	2							
						7	6	
9		7		1		2		

WebAssembly inside PHP

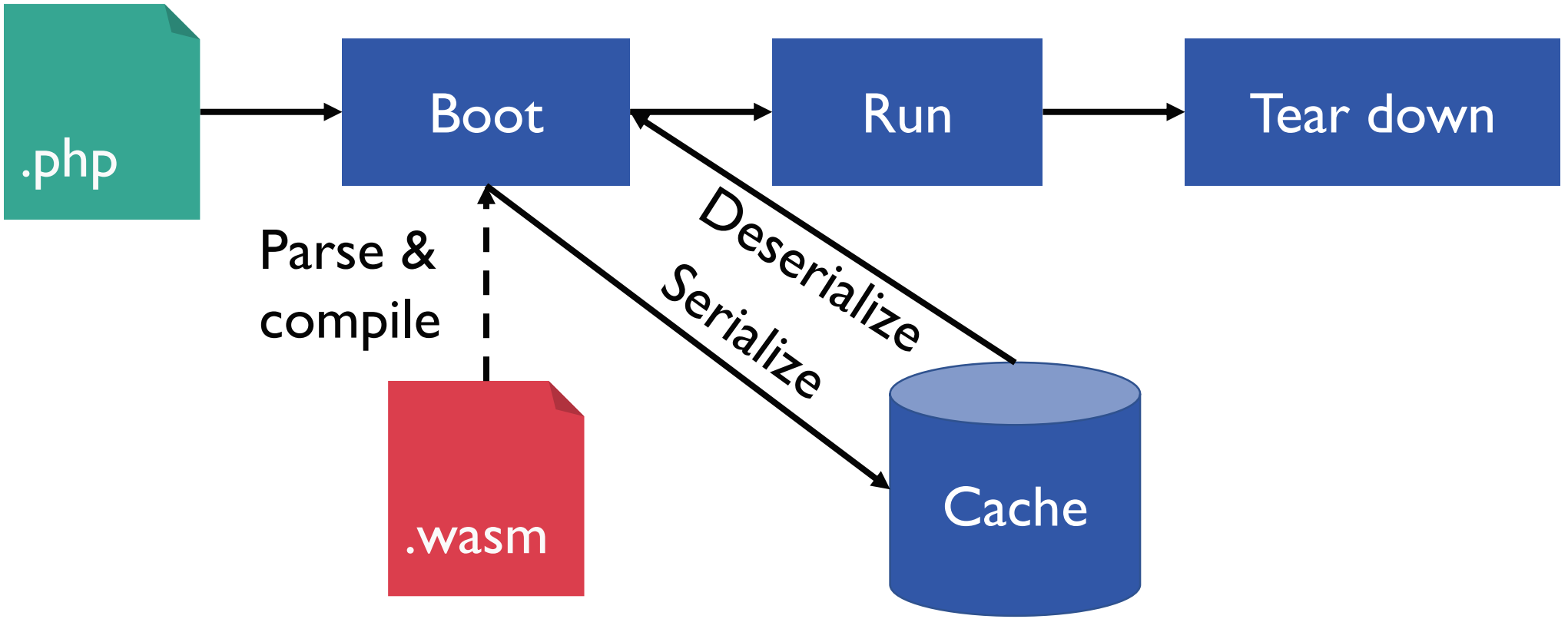
https://github.com/boyanio/php-wasm

@boyanio

Booting cost for WebAssembly modules

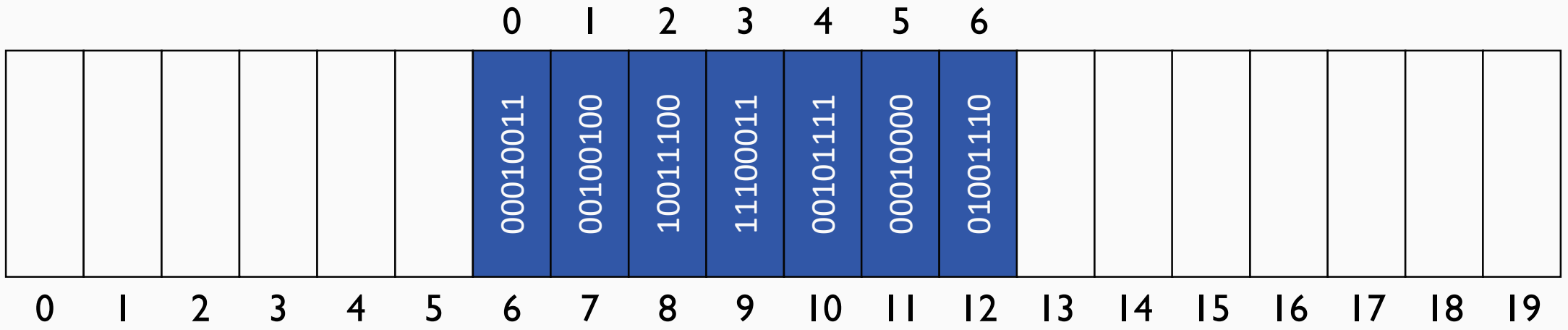


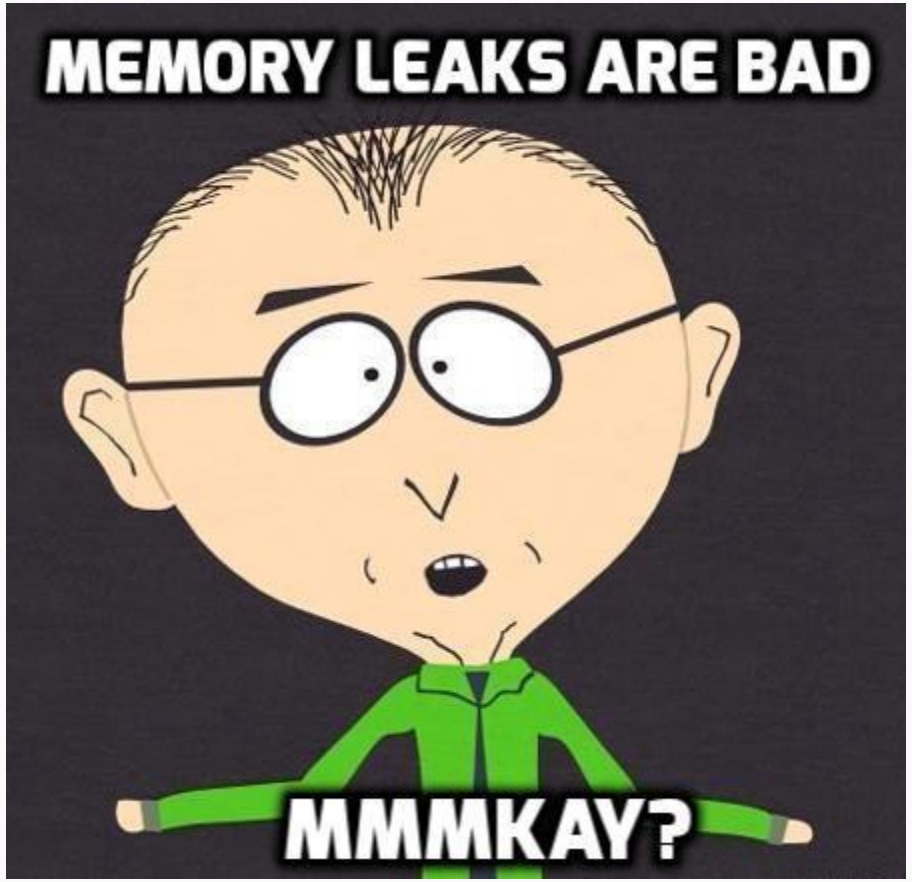
Cache WebAssembly modules



Linear memory

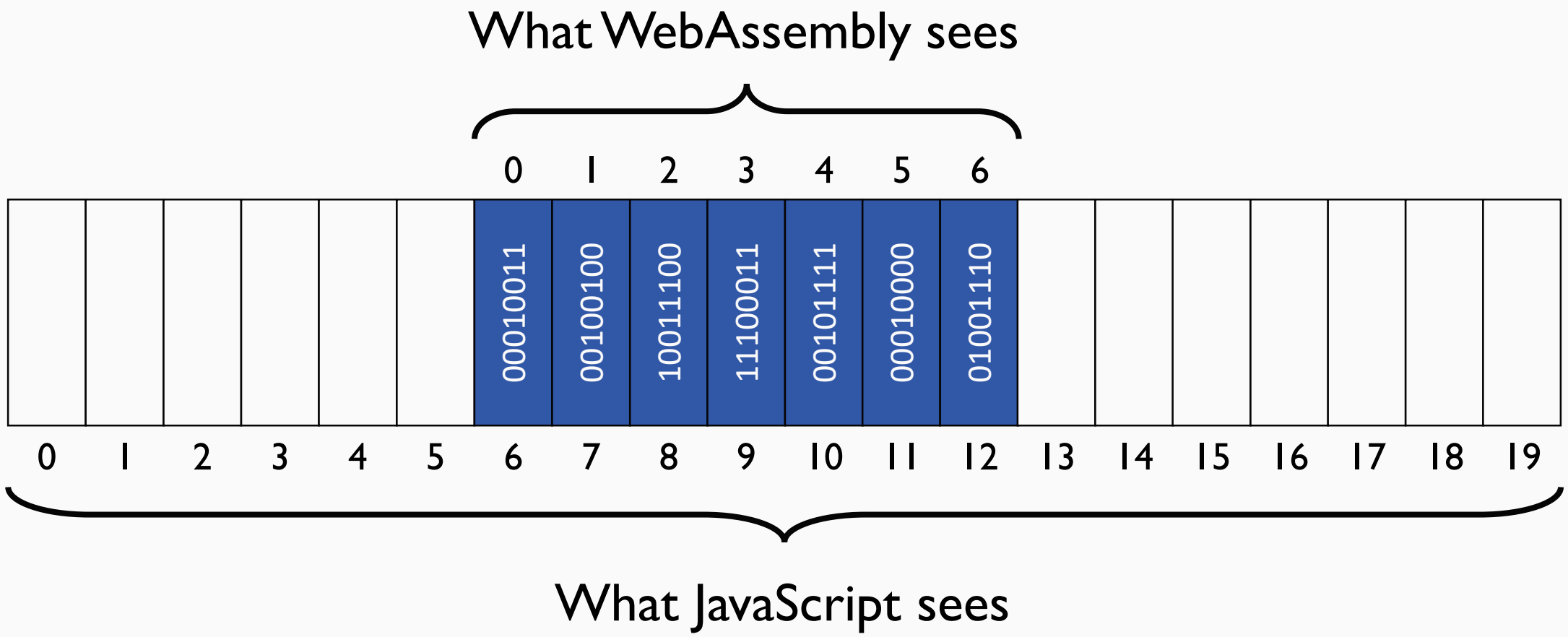
```
const memory = new WebAssembly.Memory({
  initial: 10, // initial 10 pages of 64KiB each
  maximum: 100 // max 100 pages of 64KiB each
});
```





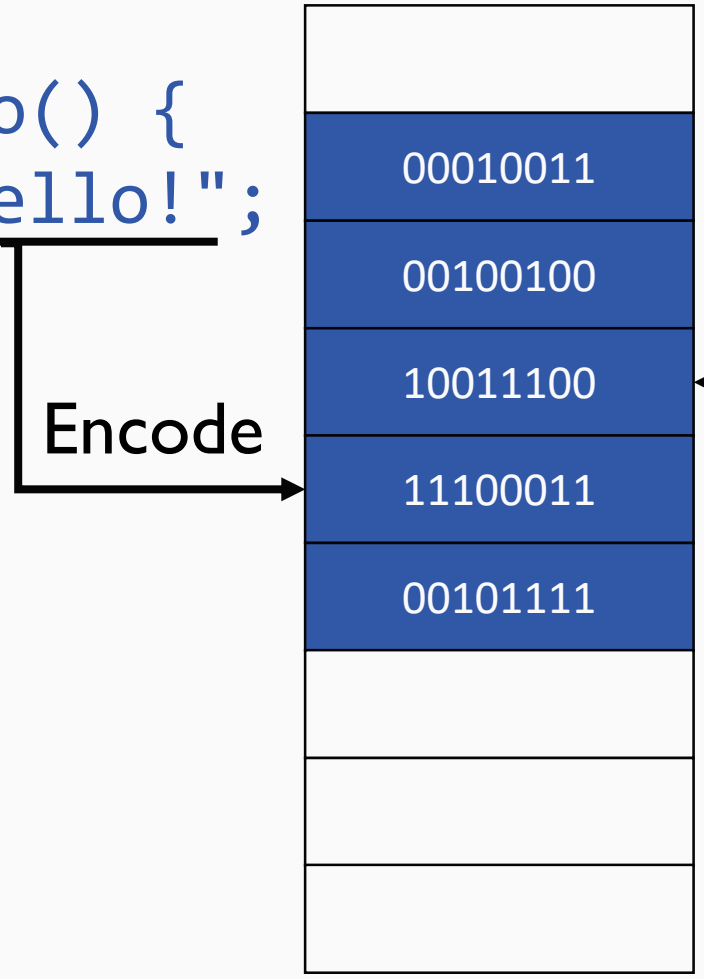
@boyanio

Linear memory



Working with strings

```
// app.c
char * hello() {
  return "Hello!";
}
```




```
// index.js
const pt = exports.hello();
const res = decode(pt);
```

Decode

```
console.log(res);
// Hello!
```

How secure is WebAssembly?

WebAssembly runs in a memory-safe sandboxed environment

The Wheel of WebAssembly

This project shows the power of WebAssembly. Each part of the wheel represents a programming language that can compile to WebAssembly. Every time you spin the wheel, a program in the language it lands on generates a random number between 1

The Wheel of WebAssembly

https://boyan.io/wasm-wheel/

@boyanio

The future of web belongs to those, who compile

Boyan Mihaylov
@boyanio
https://boyan.io