

WebAssembly is here. What does it mean for other Web frameworks?

Boyan Mihaylov
@boyanio
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!
```

All I want from
WebAssembly is...

A close-up, slightly blurred photograph of a car's instrument cluster. The primary focus is a speedometer with a black face, white markings, and a red needle. The scale ranges from 0 to 260 km/h, with major increments every 20 units. A red needle is positioned at approximately 180 km/h. Below the speedometer, there is a fuel gauge with a red needle and a 'Diesel 1/2' label. To the right of the speedometer, a red digital display shows the number '220'. The overall lighting is dim, with the red elements of the gauges providing a strong contrast.

Performance

WebAssembly provides consistent,
predictable performance

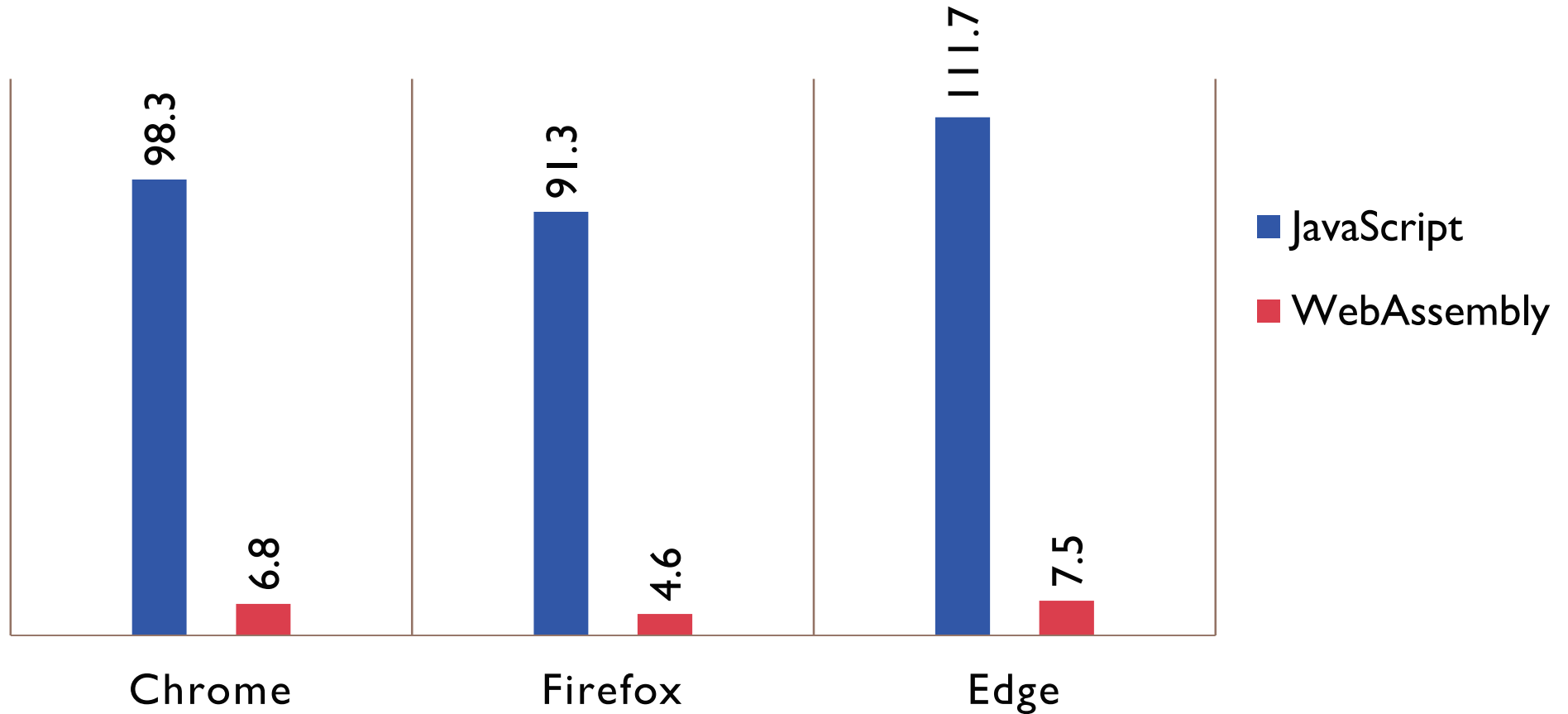


3D animation performance

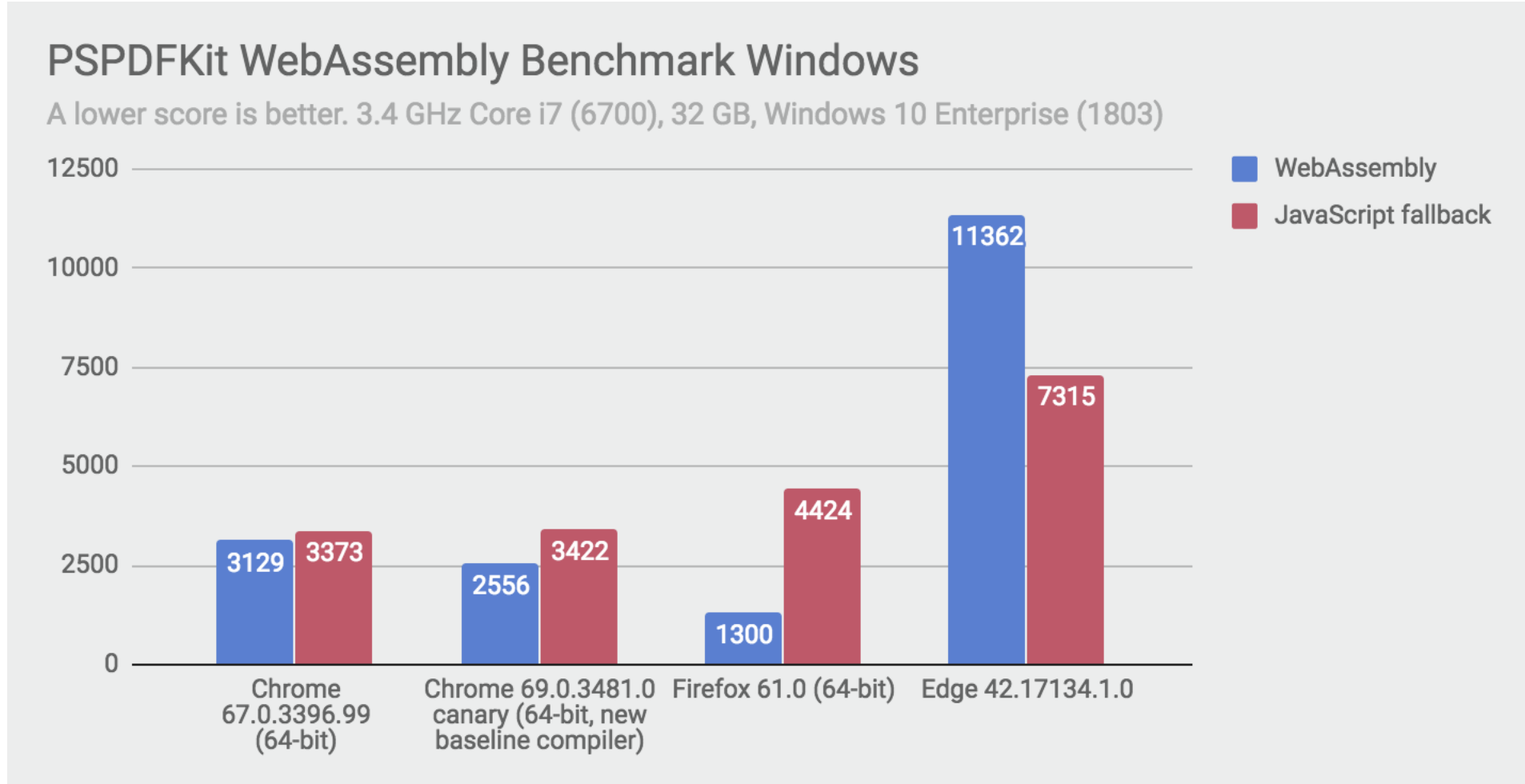
<https://github.com/sesamekesh/wasm-3d-animation-demo>

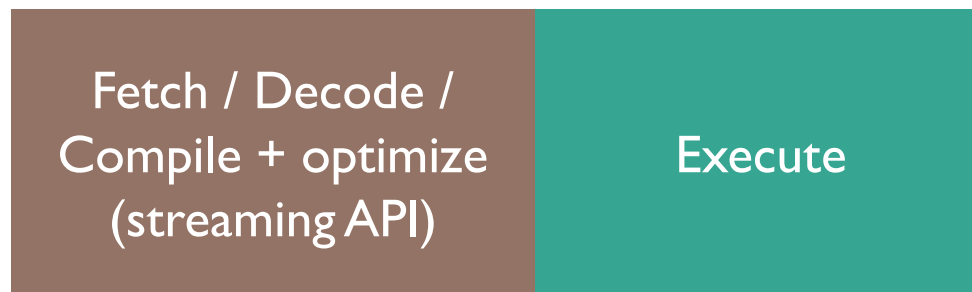
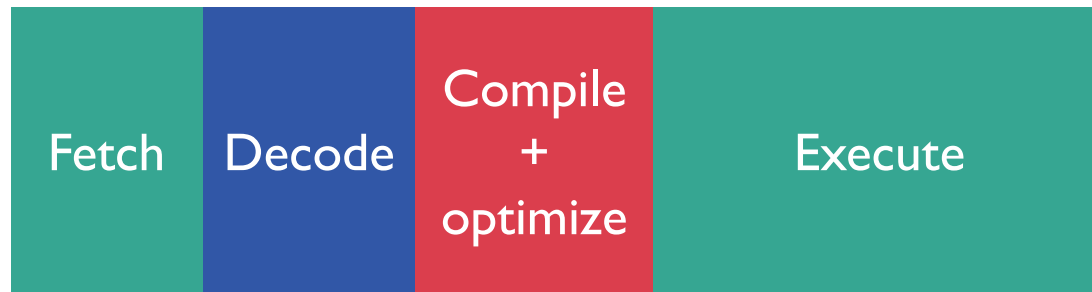
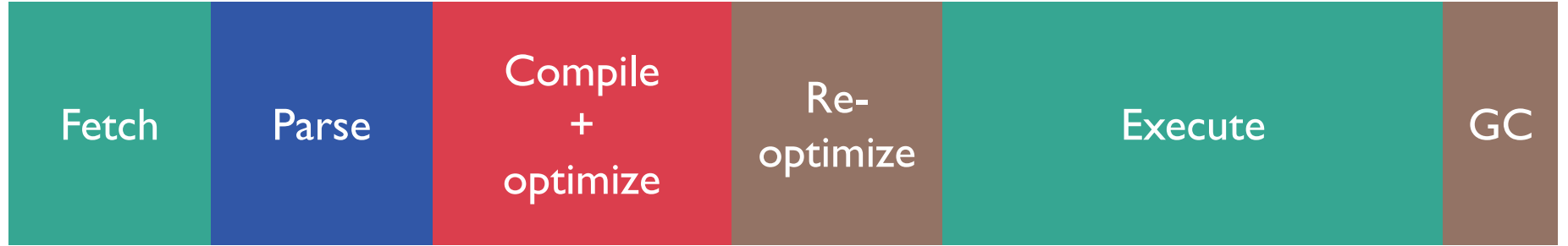
Performance comparison

Average animation time (ms)



A real-world WebAssembly benchmark







Code reusability

@boyanio

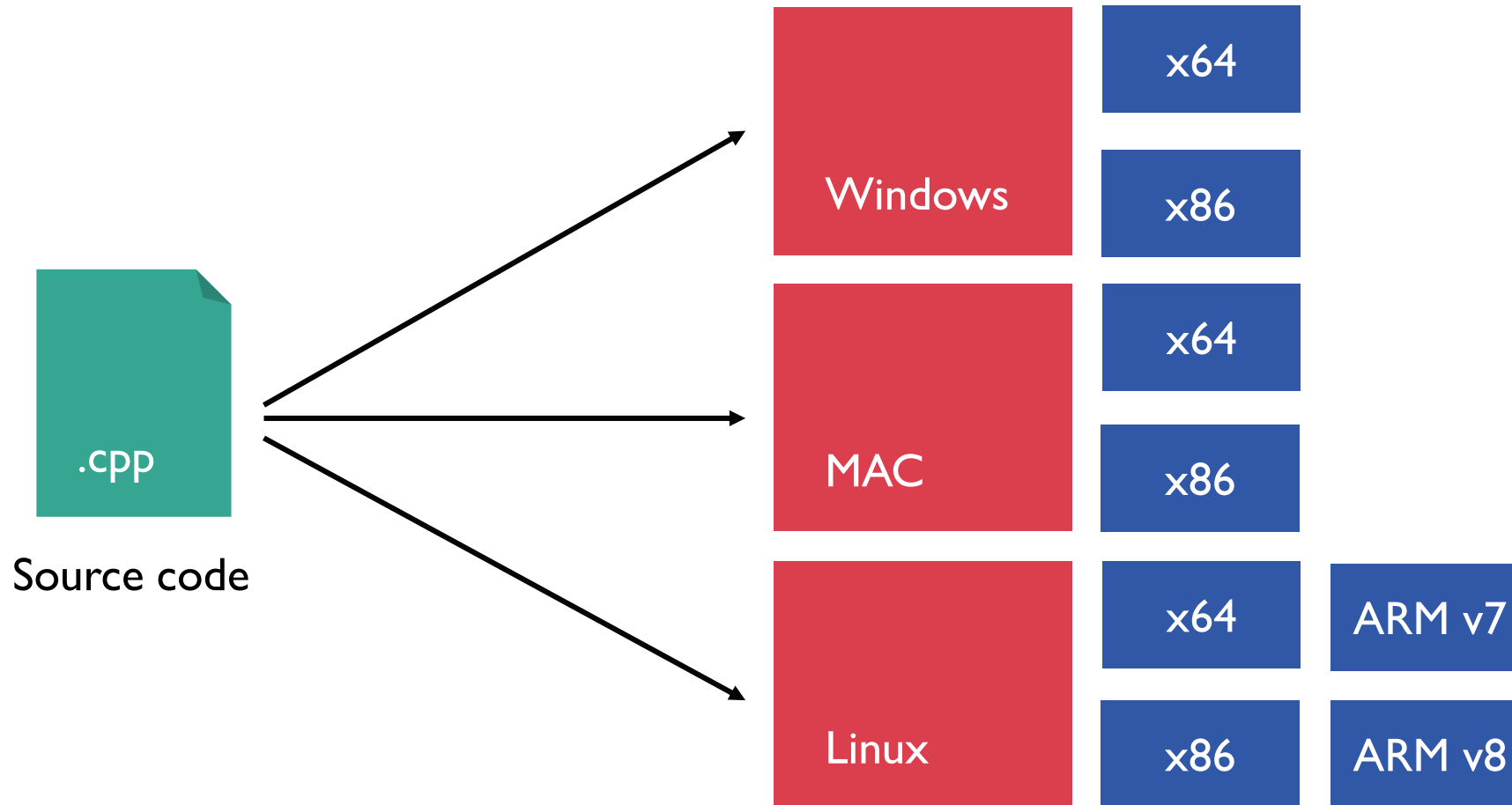
<https://techcrunch.com/2016/07/05/lzlabs-launches-product-to-move-mainframe-cobol-code-to-linux-cloud/>



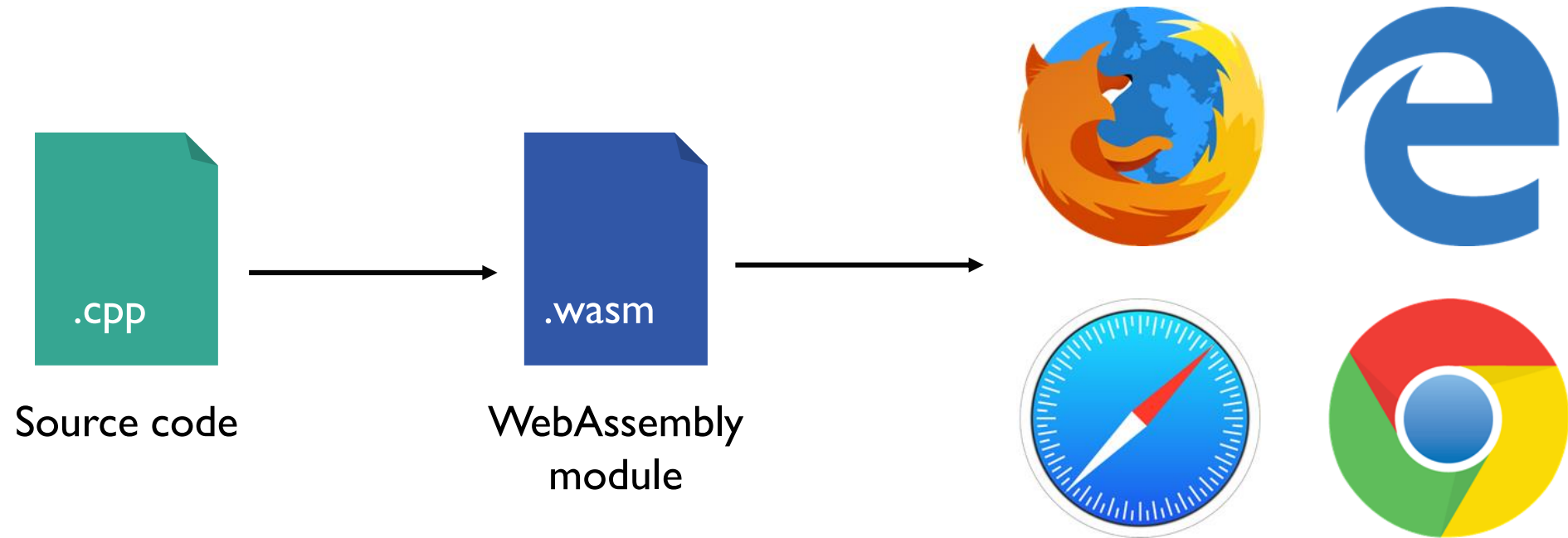
The Adobe Flash plugin has crashed.
[Send crash report](#)

Reusing code on the Web

Traditional multi-target compilation



Multi-target compilation with WebAssembly



LIVE

BREAKING NEWS

WASM REPLACING JAVASCRIPT?

22:57

WILL WEBASSEMBLY OVERTAKE JAVASCRIPT IN WEB APPLICATION CODING NEEDS?

@boyanio

<https://www.washingtonexaminer.com/cnn-nyt-reporters-aggressively-miss-the-point-with-nikki-haleys-reaction-to-the-grammys-stupid-fire-and-fury-reading>

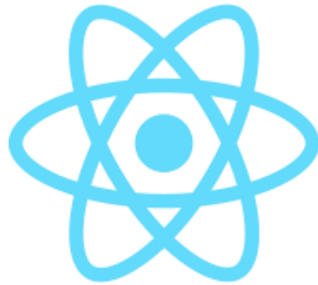


“WebAssembly fills in the gaps that would be awkward to fill with JavaScript.”

Eric Elliott

A close-up photograph of a Monopoly board. A red rectangular text box is superimposed over the center of the board. The board shows various property names and prices, including 'OLD KENT ROAD £60', 'MAYFAIR £400', 'PARK LANE £350', and 'SUPER TAX PAY £100'. Three game pieces (yellow, red, and white) are visible on the board near the 'GO' space. A 'CHANCE' square is also visible in the upper left.

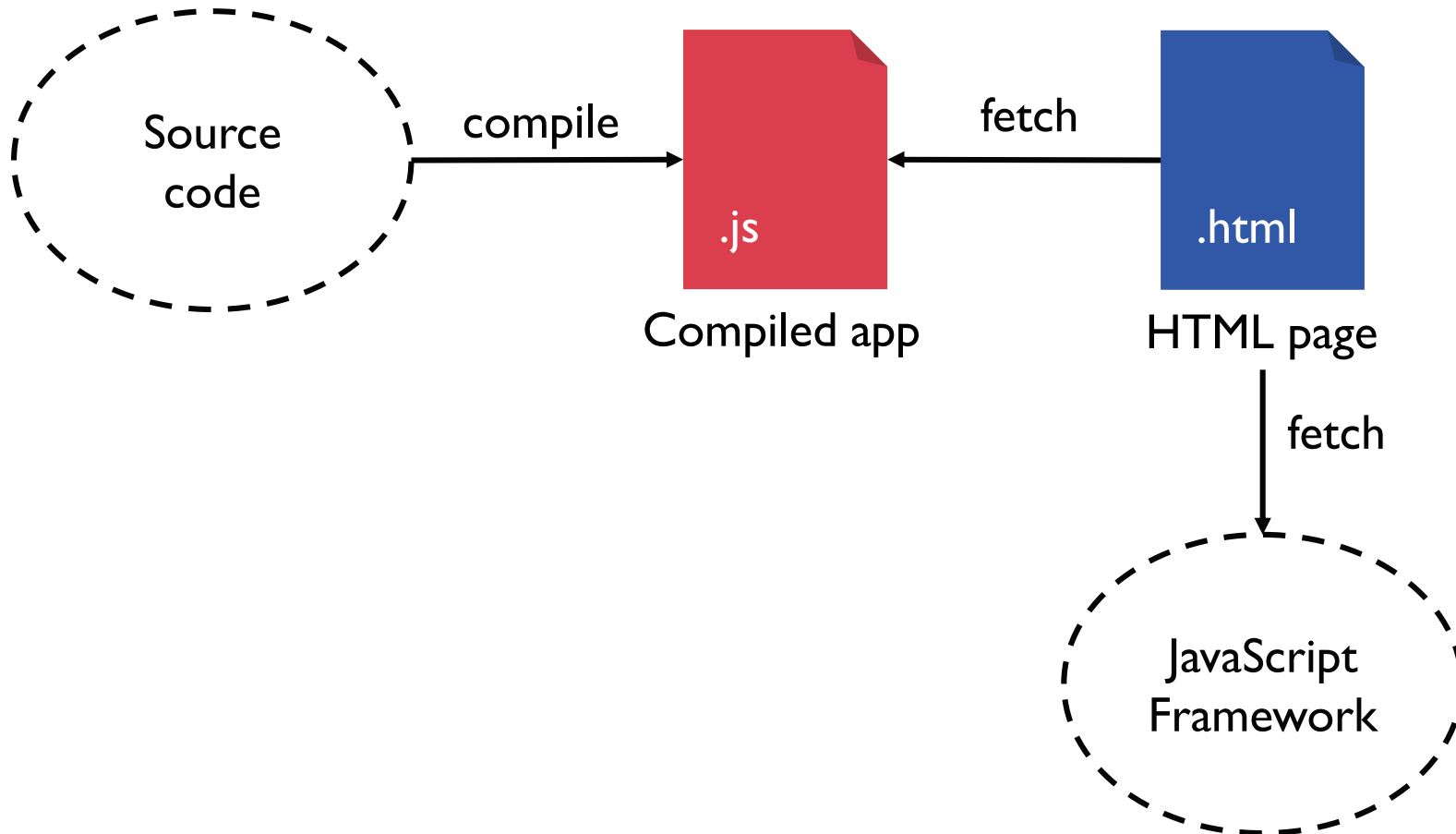
WebAssembly will break
the JavaScript monopoly



The Web of JavaScript frameworks



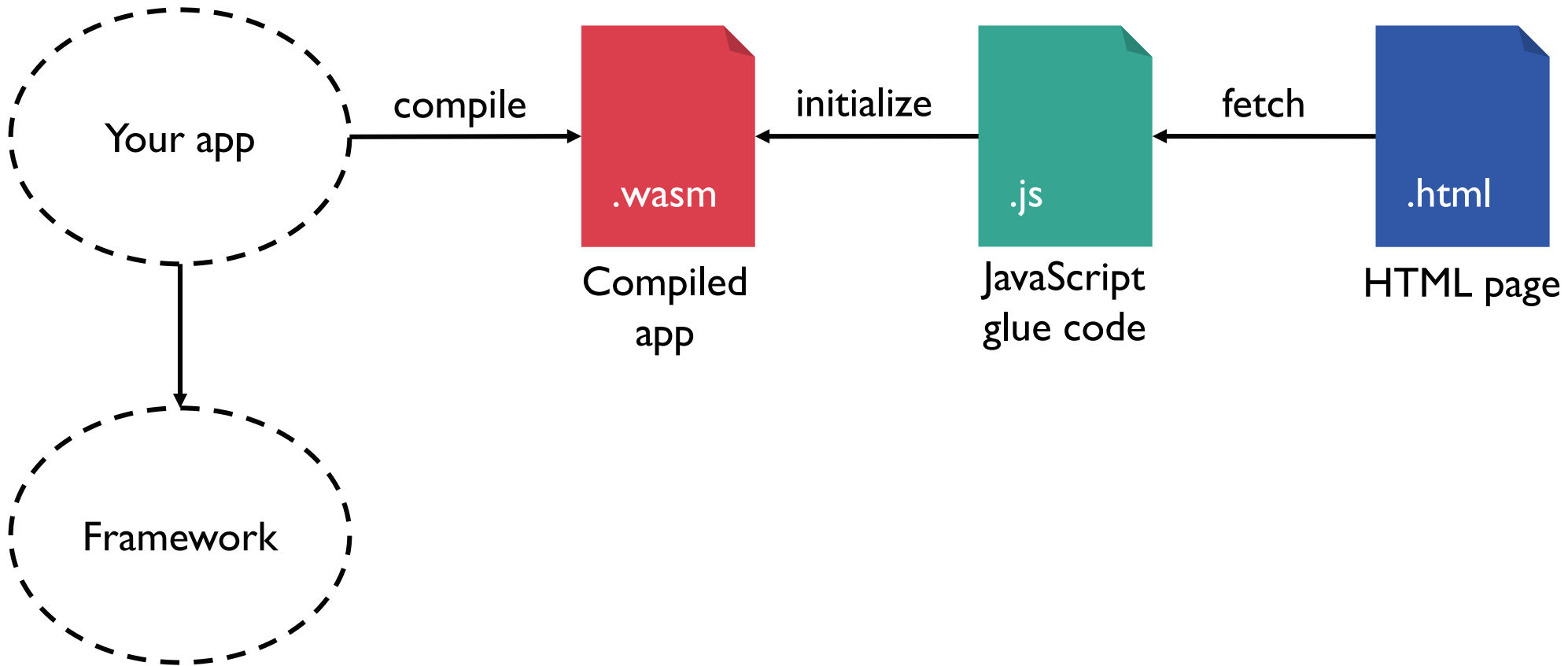
JavaScript frameworks architecture



Rewriting existing JavaScript
frameworks into a language
that can be compiled to
WebAssembly



WebAssembly-compiled frameworks



C++/Python → JavaScript → C++/Python

A graphic for a JavaScript coding camp. It features a cartoon character with purple hair, wearing a brown flight suit and goggles, pointing upwards. The background is blue with clouds and a large white moon. In the top left corner, there is an orange square with the letters 'JS' in white. Below the character, there are several interlocking gears of different sizes.

JavaScript Coding

Half Day Camp | Grades 7-8

Learn JavaScript and build your own games for web and mobile platforms

No direct DOM access

index.c

```
extern void createElement(void);

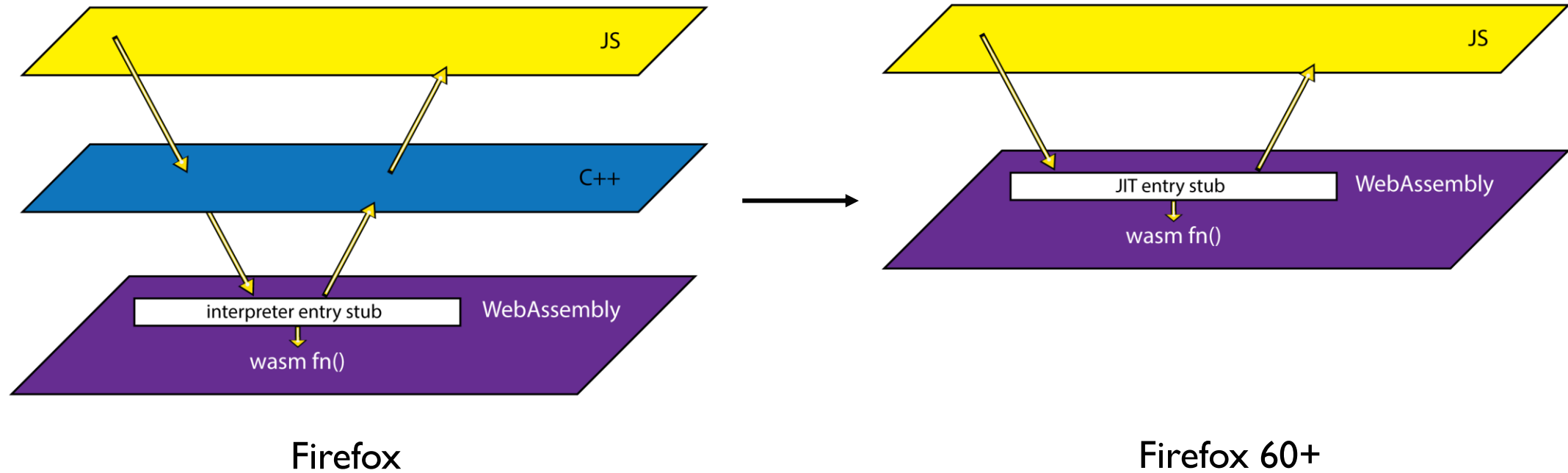
int main(void)
{
    createElement();
    createElement();
    ...
    return 0;
}
```

main.js

```
const imports = {
    createElement: () => {
        document.createElement('div');
    }
};

WebAssembly.instantiate(..., imports);
```

JavaScript → WebAssembly overhead



Easier to create fast native mobile apps



Rewriting *parts* of existing
JavaScript frameworks into
a language that can be compiled
to WebAssembly



2



Languages

- Java: [RxJava](#)
- JavaScript: [RxJS](#)
- C#: [Rx.NET](#)
- C#(Unity): [UniRx](#)
- Scala: [RxScala](#)
- Clojure: [RxClojure](#)
- C++: [RxCpp](#)
- Lua: [RxLua](#)
- Ruby: [Rx.rb](#)
- Python: [RxPY](#)
- Go: [RxGo](#)
- Groovy: [RxGroovy](#)
- JRuby: [RxJRuby](#)
- Kotlin: [RxKotlin](#)
- Swift: [RxSwift](#)
- PHP: [RxPHP](#)
- Elixir: [reaxive](#)
- Dart: [RxDart](#)

LibSass - Sass compiler written in C++

Currently maintained by Marcel Greter (@mgreter) and Michael Mifsud (@xzyfer)

Originally created by Aaron Leung (@akhleung) and Hampton Catlin (@hcatlin)

build passing build passing coverage 86% open issues 4% issue resolution 2 d bountysource \$130 in 0 bounties slack 2/146

[LibSass](#) is just a library! If you want to use LibSass to compile Sass, you need an implementer. Some implementations are only bindings into other programming languages. But most also ship with a command line interface (CLI) you can use directly. There is also [SassC](#), which is the official lightweight CLI tool built by the same people as LibSass.

Excerpt of "sanctioned" implementations:

- <https://github.com/sass/node-sass> (Node.js)
- <https://github.com/sass/perl-libsass> (Perl)
- <https://github.com/sass/libsass-python> (Python)
- <https://github.com/wellington/go-libsass> (Go)
- <https://github.com/sass/sassc-ruby> (Ruby)
- <https://github.com/sass/libsass-net> (C#)
- <https://github.com/medialize/sass.js> (JS)
- <https://github.com/bit3/jsass> (Java)

This list does not say anything about the quality of either the listed or not listed [implementations](#)!

The authors of the listed projects above are just known to work regularly together with LibSass developers.

Webassembly integration. Split the core into two parts. #8193

<https://github.com/vuejs/vue/issues/8193>

Initial stab at porting `asm/stack.ts` to Rust #752

<https://github.com/glimmerjs/glimmer-vm/pull/752>

Writing custom components
in a language that can be
compiled to WebAssembly



3

Angular & WebAssembly

A collection of examples of how WebAssembly can be used with Angular



[Home](#) [GitHub](#) [Twitter](#)

Fibonacci battlefield

Console logger

Text to ASCII art converter

Bitmap to ASCII art converter

3D cube

Proof of work

Angular & WebAssembly

<https://boyan.io/angular-wasm/>

The rise of non-JavaScript Web frameworks

[Docs](#)[API](#)[Community](#)[Blog](#)

Blazor

Full-stack web development with C# and WebAssembly

[↗ Get Started](#)

Build a Web UI with C#

Blazor is an experimental .NET web framework using C# and HTML that runs in the browser.

[What is Blazor?](#)

Blazor
<https://blazor.net>

Full-stack .NET

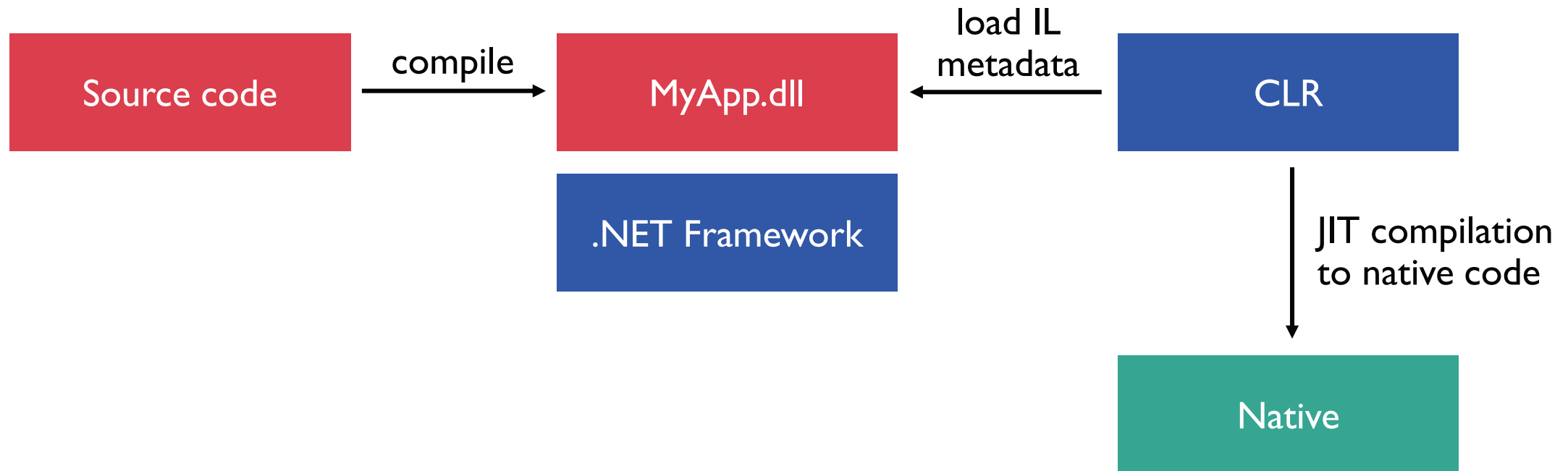
Do full-stack .NET development using stable and consistent tools, languages, and APIs both in the browser and on the server.

[Learn more about the .NET platform](#)

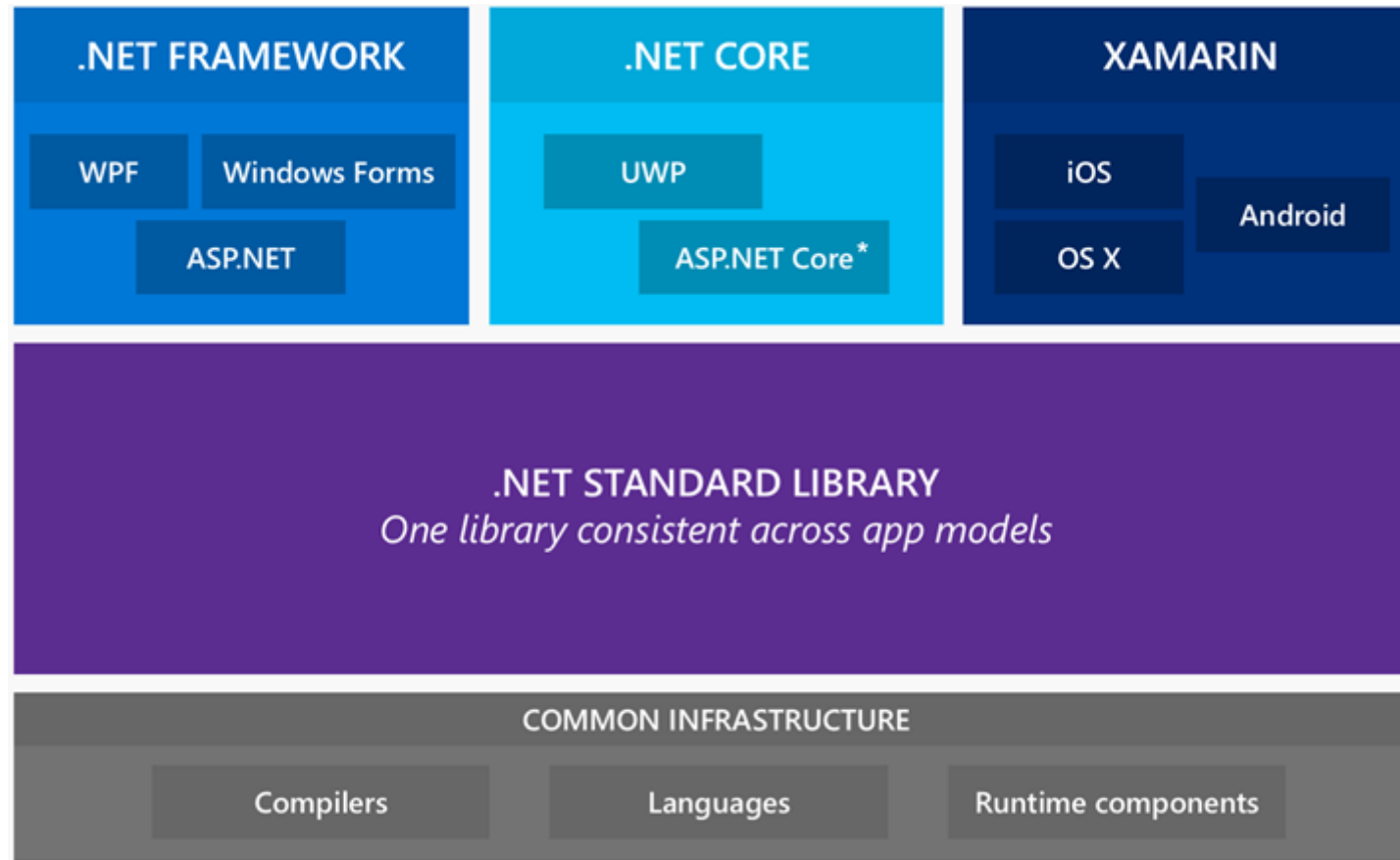


@boyanio

Traditional .NET architecture



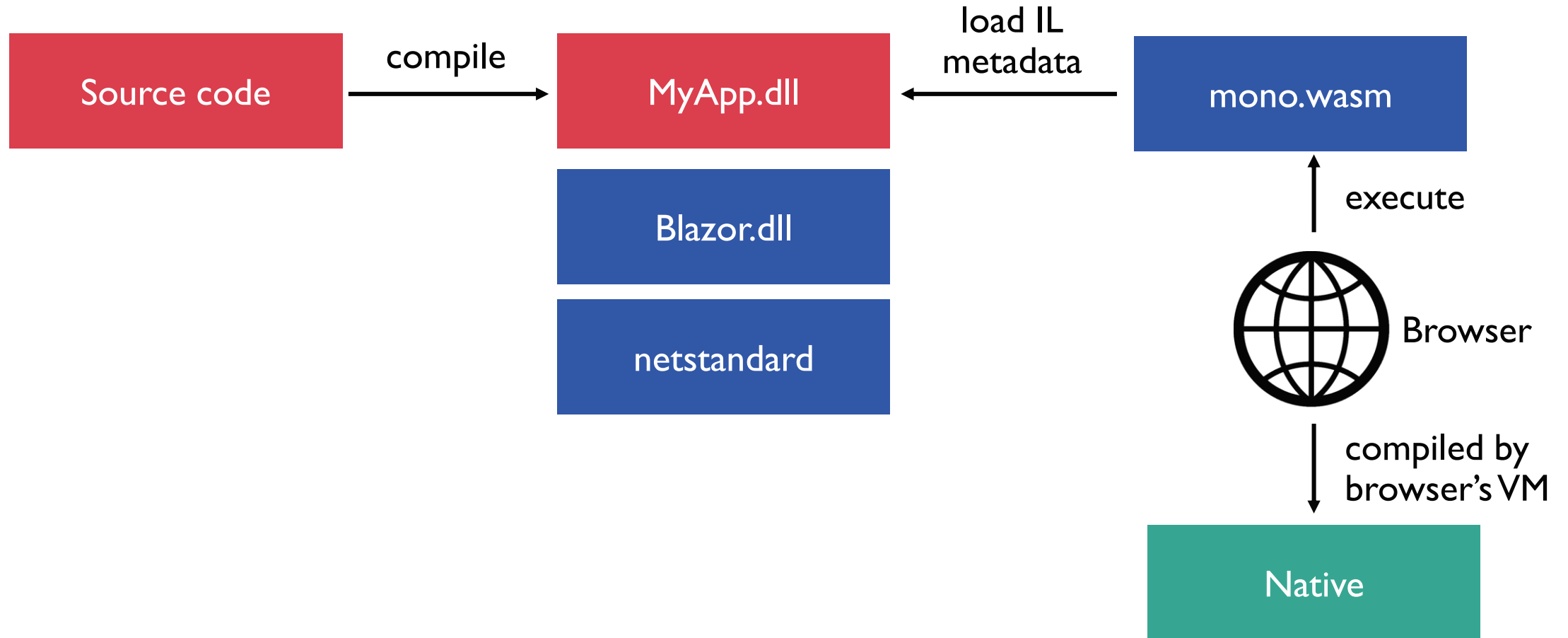
.NET Core architecture



Mono is an open source implementation of Microsoft's .NET Framework



Blazor architecture



React vs. Blazor

This demo shows how React apps can live together with Blazor apps, which are basically C# apps running in the browser with the help of WebAssembly.

React chat Blazor chat

Send

It's coming

B

22 seconds ago



Not loading fast yet...

28 seconds ago

Hi! I am cool, a?!

B

54 seconds ago

Send

It's coming

B

22 seconds ago



Not loading fast yet...

28 seconds ago

Hi! I am cool, a?!

B

54 seconds ago

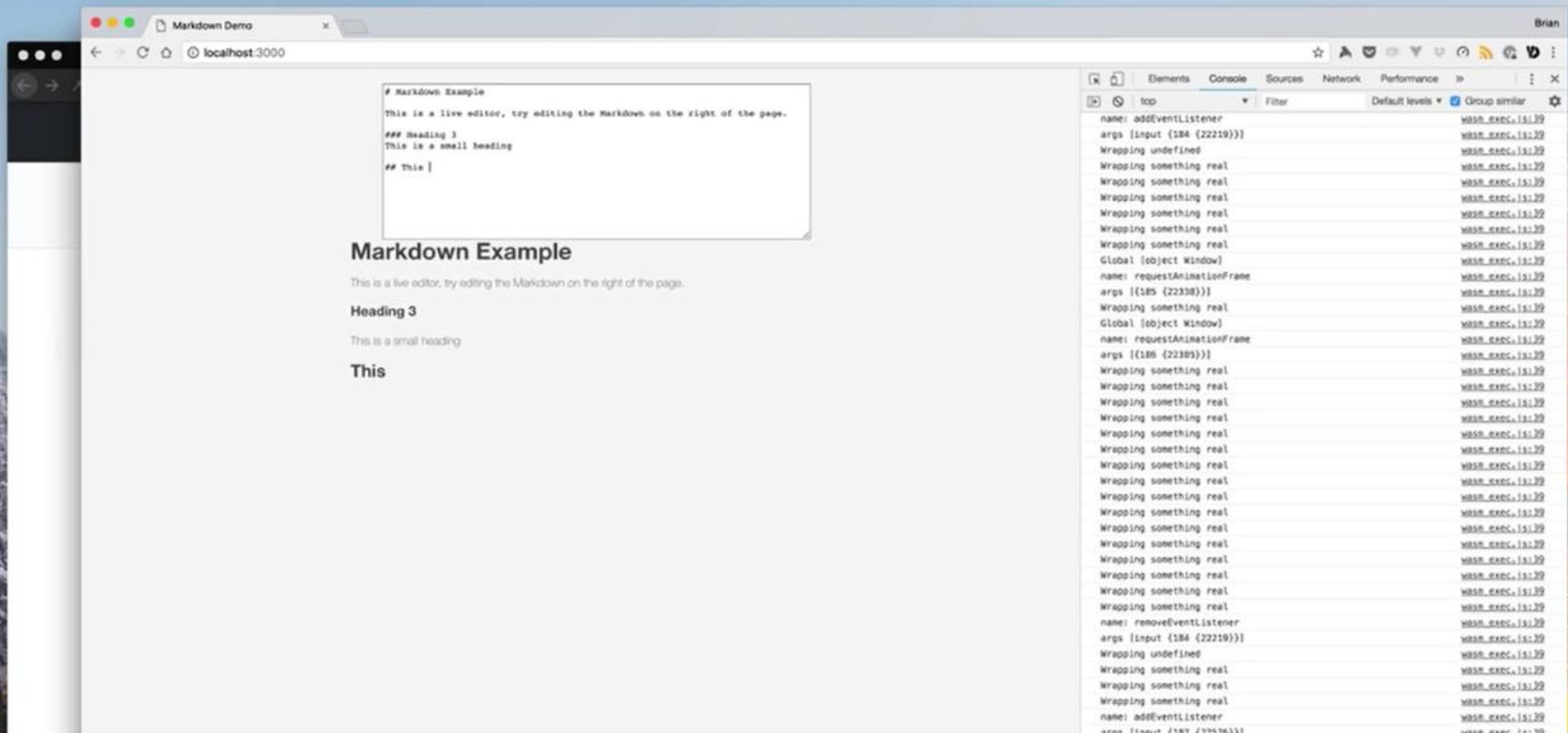


Hello!

1 minute ago

React vs. Blazor

<https://boyan.io/react-blazor/>



Blazor-inspired Web framework in Go

<https://github.com/bketelsen/wasmplay>

WebAssembly enables different languages to work together on the Web

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



How secure is WebAssembly?





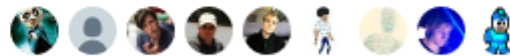
Alon Zakai

@kripen

WebAssembly is not at risk, but multithreading in WebAssembly is in the same state as SharedArrayBuffer in JavaScript (not going to be enabled until the security issues are handled)

5:29 PM - 5 Jan 2018

10 Retweets 19 Likes



↺ 10



19



WebAssembly runs in a memory-safe
sandboxed environment

The future of Web
belongs to those, who compile

Boyan Mihaylov / @boyanio / boyan.io