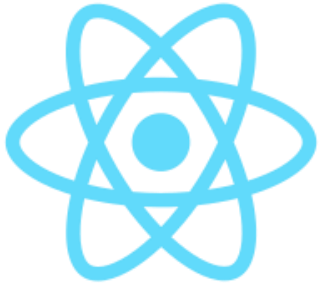
The rise of non-JavaScript frameworks using WebAssembly

Boyan Mihaylov
@boyanio
boyan.io

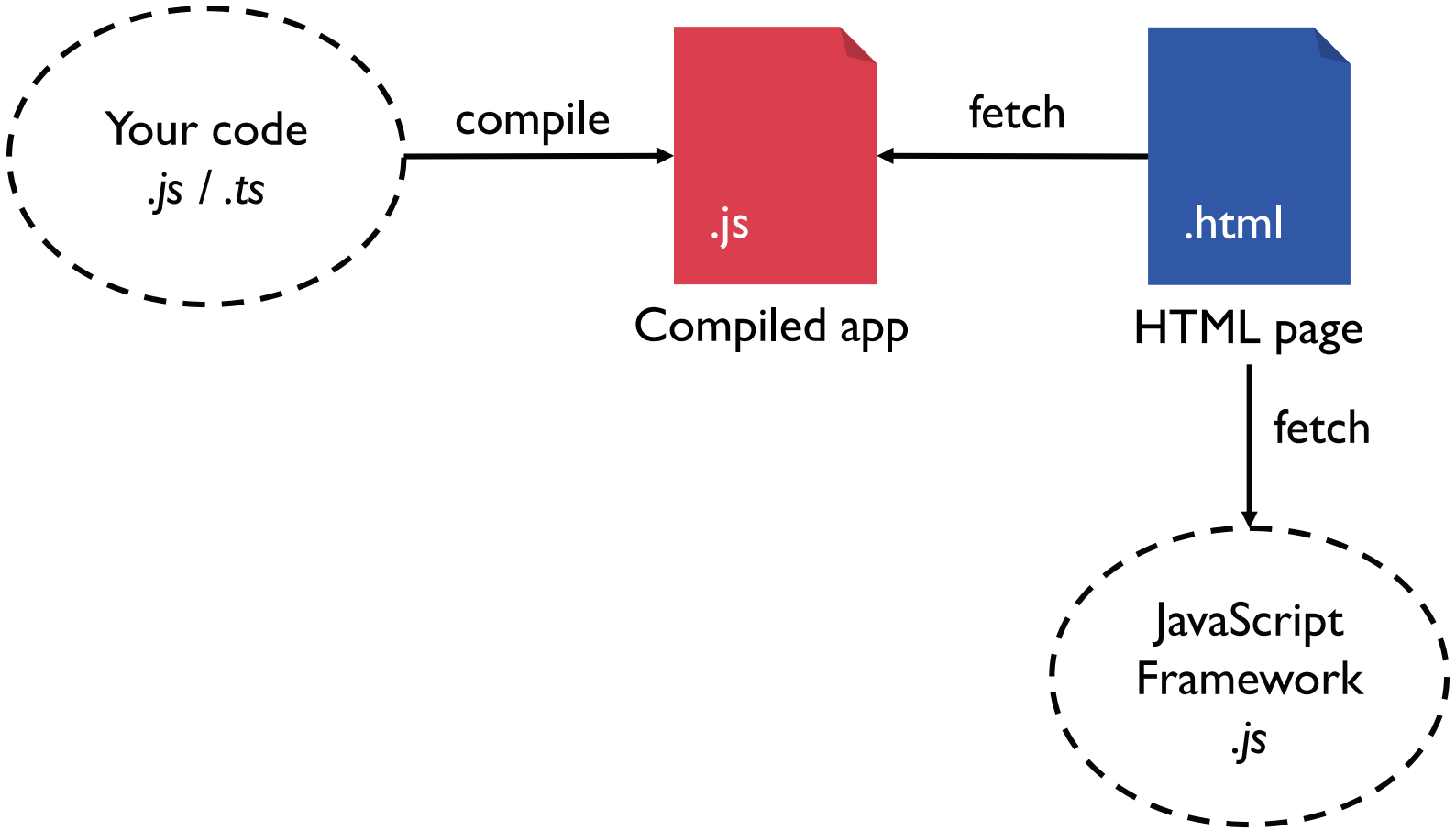
WebAssembly (*WASM*) is compiler target for programs on the Web

```
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!
```


The Web of JavaScript frameworks

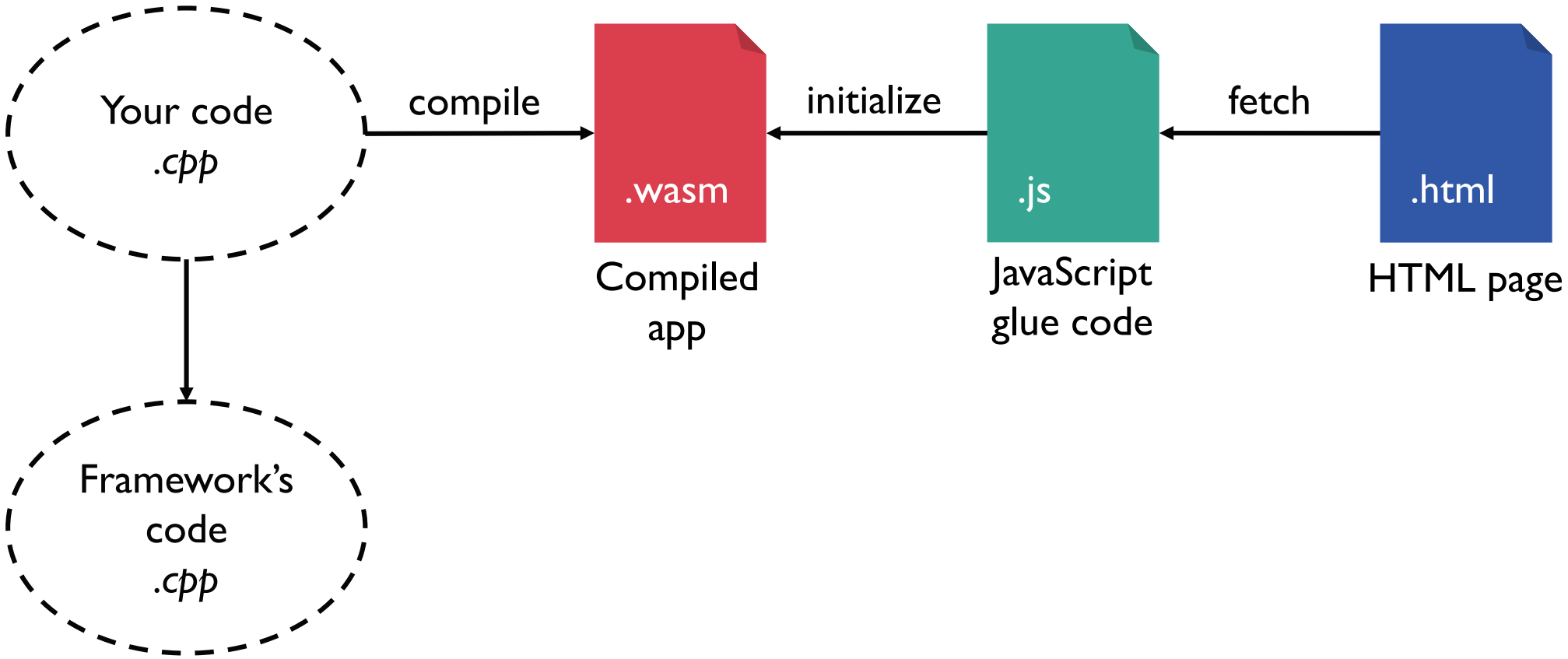

JavaScript frameworks architecture



How would WebAssembly influence the way we do Web development today?

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

WebAssembly-compiled frameworks



C++/Python → JavaScript → C++/Python

No direct DOM access

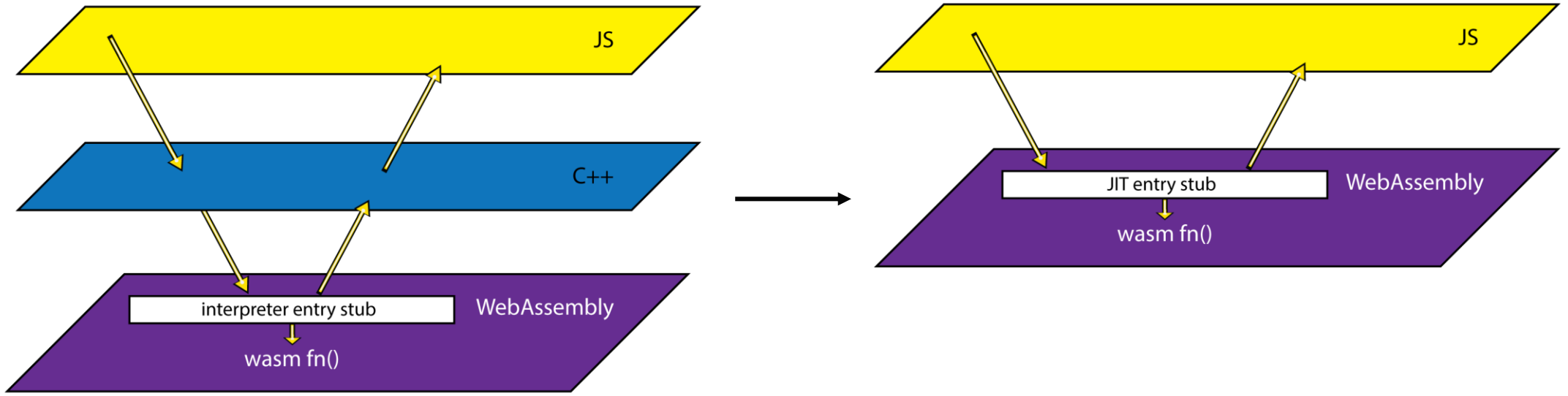
index.c

```
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



Firefox

Firefox 60+

https://blog.benj.me/2018/07/04/mozilla-2018-faster-calls-and-anyref/

Easier to create fast native mobile apps

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

2

Readme.md

LibSass - Sass compiler written in C++

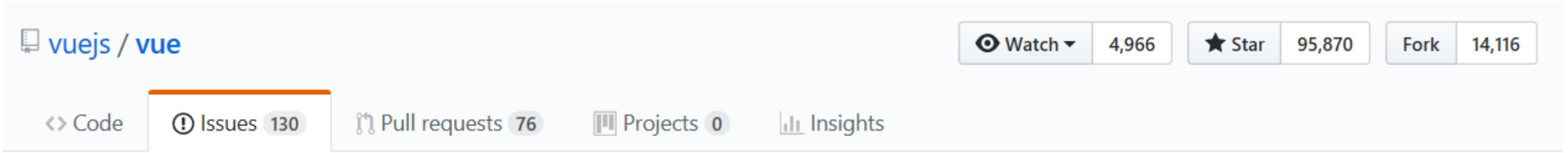
Currently maintained by Marcel Greter (@mgreter) and Michael Mifsud (@xzyfer)
Originally created by Aaron Leung (@akhleung) and Hampton Catlin (@hcatlin)

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:

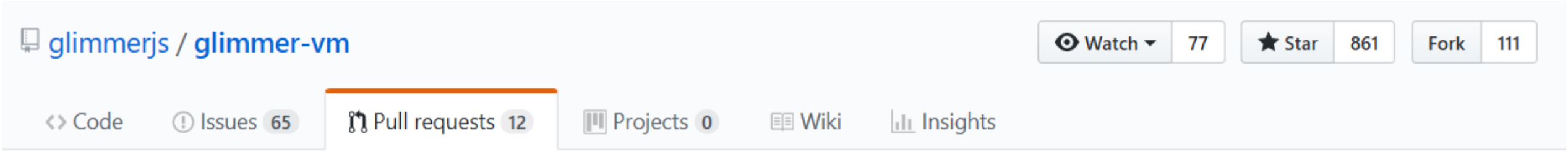
- https://github.com/sass/node-sass (Node.js)
- https://github.com/sass/perl-libsass (Perl)
- https://github.com/sass/libsass-python (Python)
- https://github.com/wellington/go-libsass (Go)
- https://github.com/sass/sassc-ruby (Ruby)
- https://github.com/sass/libsass-net (C#)
- https://github.com/medialize/sass.js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

Angular & WebAssembly

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

@boyanio

Emergence of non-JavaScript frameworks using WebAssembly

4

Blazor

Build client web apps with C#

Get Started Documentation

Supported on Windows, Linux, and macOS

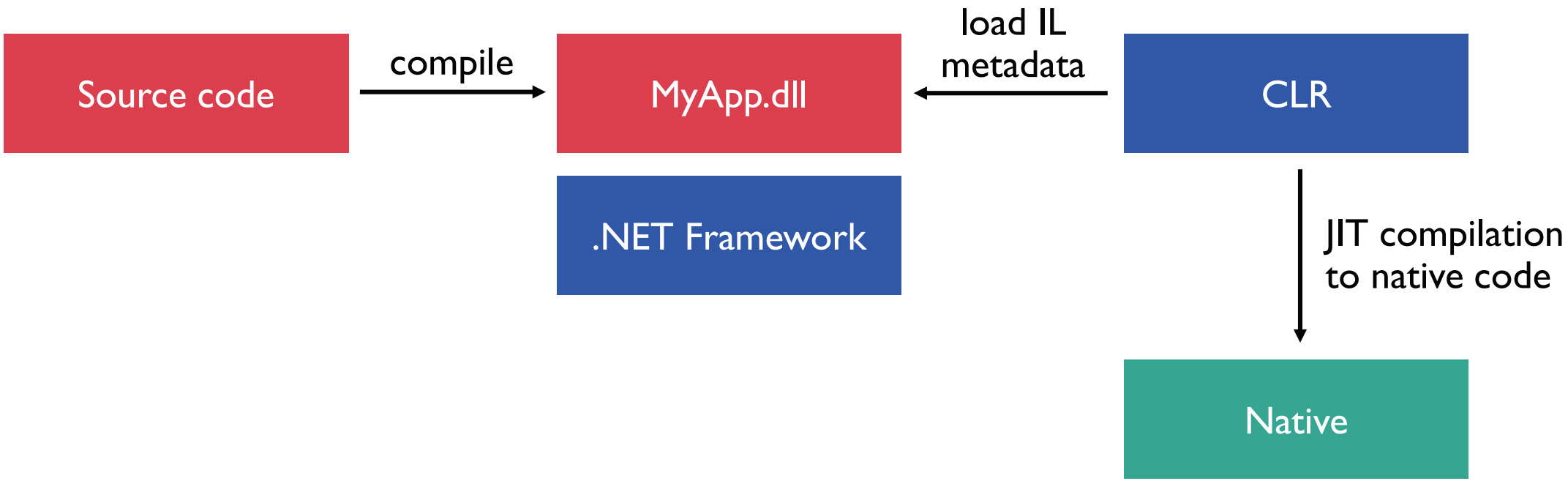
Blazor
https://blazor.net

@boyanio

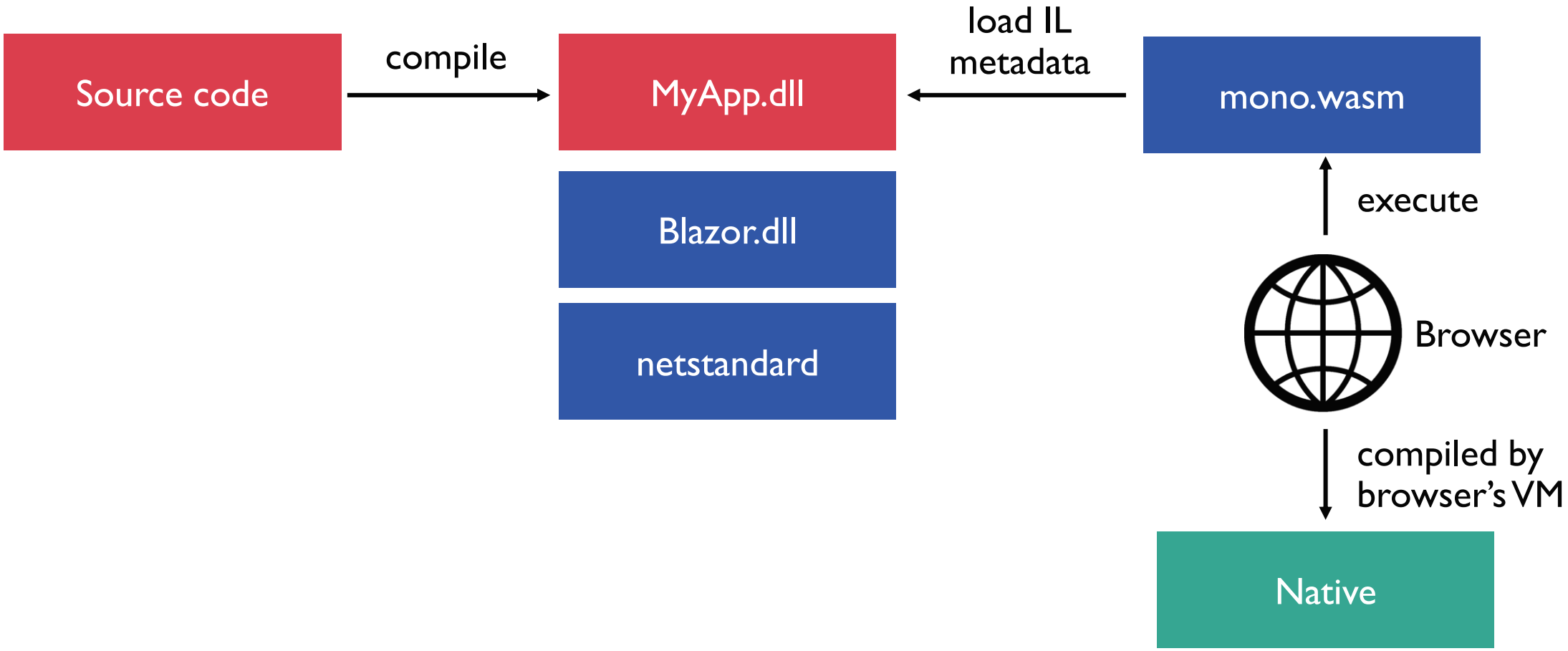
Interactive web UI with C#

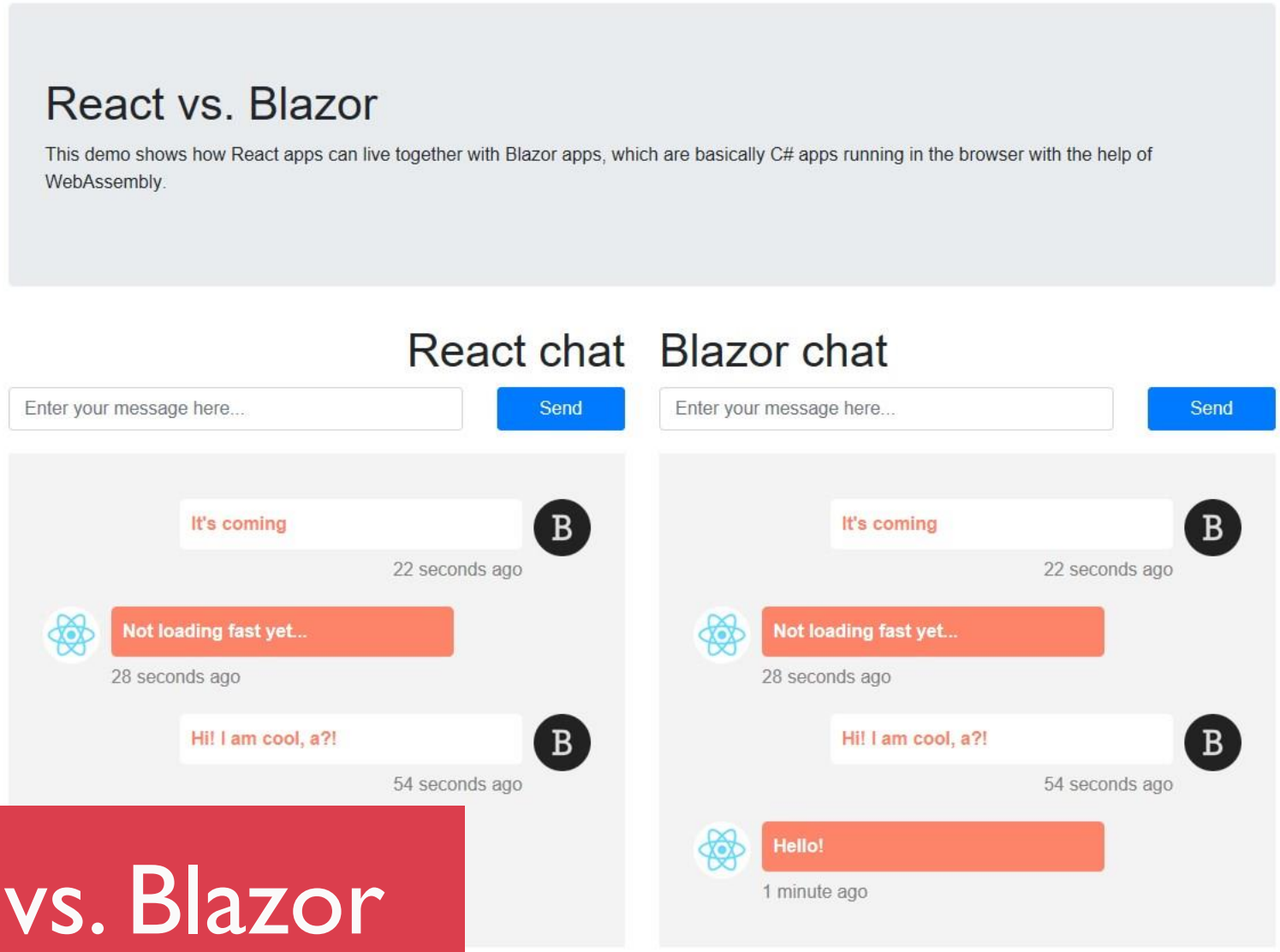
Blazor lets you build interactive web UIs using C# instead of JavaScript. Blazor apps are composed of

Traditional .NET architecture



Blazor architecture





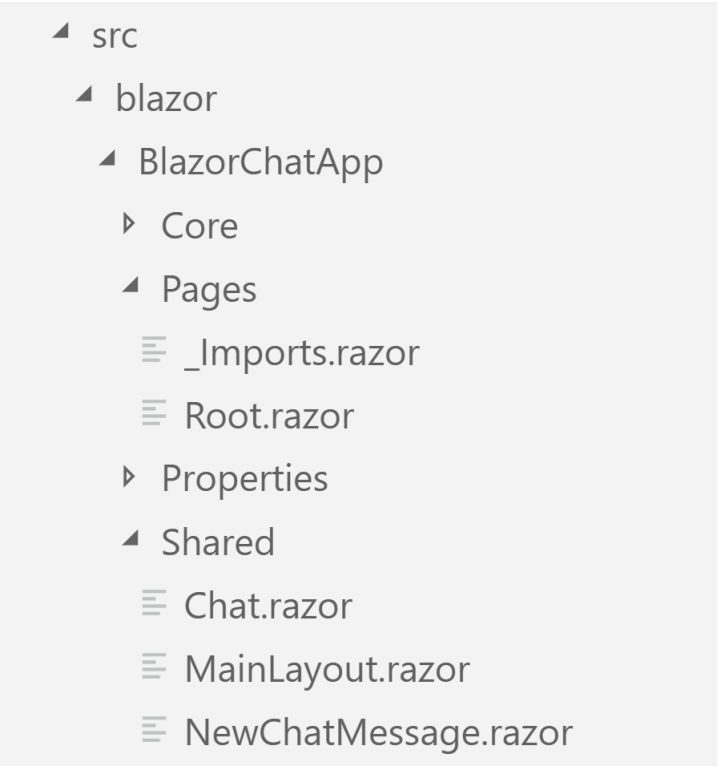
React vs. Blazor

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

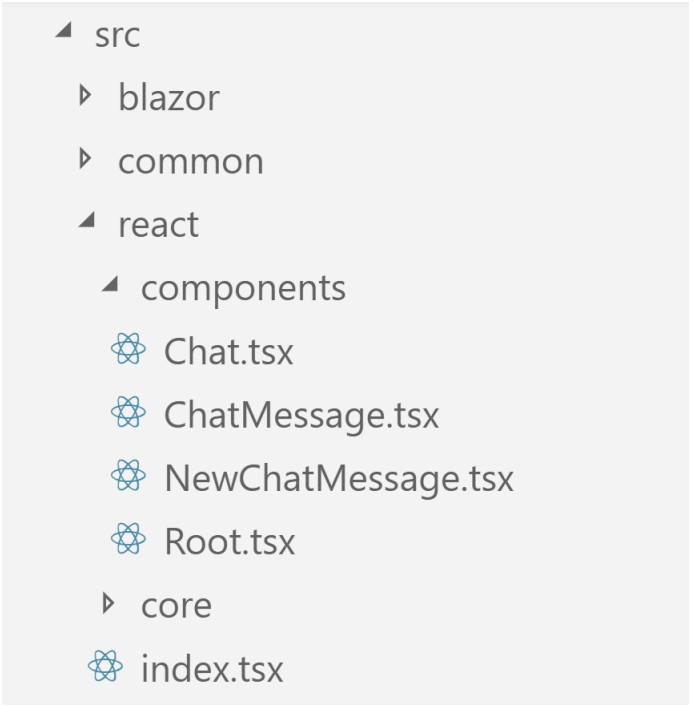
@boyanio

Application structure

Blazor



React



Application structure

Blazor (Root.razor)

```
@page “/”

<div>
  <h1>Blazor chat</h1>
  <NewChatMessage />
  <Chat … />
</div>
```

React (Root.jsx)

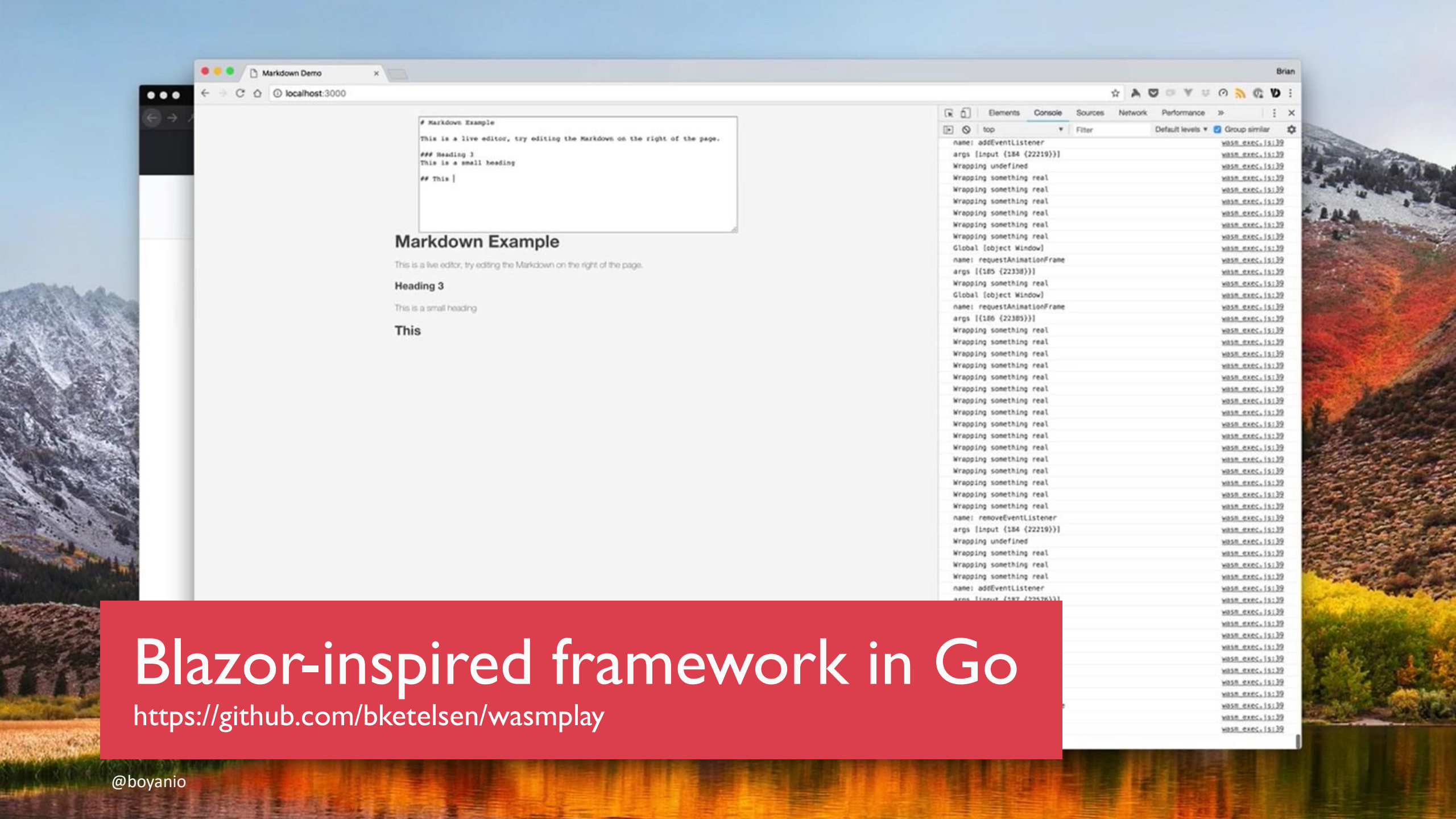
```
…
  render() {
    return (
      <div>
        <h1>React chat</h1>
        <NewChatMessage />
        <Chat … />
      </div>
    );
  }
}
```

Blazor now in official preview!

Daniel

April 18th, 2019

With this newest Blazor release we're pleased to announce that **Blazor is now in official preview!** Blazor is no longer experimental and we are committing to ship it as a supported web UI framework including support for running client-side in the browser on WebAssembly.


Markdown Demo
Brian
localhost:3000
Markdown Example
This is a live editor, try editing the Markdown on the right of the page.
Heading 3
This is a small heading
This
Markdown Example
This is a live editor, try editing the Markdown on the right of the page.
Heading 3
This is a small heading
This
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
Group similar
Blazor-inspired framework in Go
https://github.com/bketelsen/wasmplay
@boyanio

README.md

build passing gitter join chat rustc 1.30+

Yew

Yew (pronounced `/juː/`, the same way as "you") is a modern Rust framework inspired by Elm and ReactJS for creating multi-threaded frontend apps with WebAssembly.

The framework supports ***multi-threading & concurrency*** out of the box. It uses Web Workers API to spawn actors (agents) in separate threads and uses a local scheduler attached to a thread for concurrent tasks.

Become a sponsor on Patreon

Cutting Edge technologies

Rust to WASM compilation

This framework is designed to be compiled into modern browsers' runtimes: wasm, asm.js, emscripten.

React-inspired framework in Rust

https://github.com/DenisKolodin/yew

`src/main.rs`

WebAssembly enables different languages to work together on the Web

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

The future of Web belongs to those, who compile

Boyan Mihaylov / @boyanio / boyan.io