

Compiling JavaScript ahead-of-time

Oliver Medhurst

Chrome



V8



Node
Deno

Safari



JavaScriptCore



Bun

Firefox
Servo



SpiderMonkey

Interpretation

JavaScript
source code



QuickJS

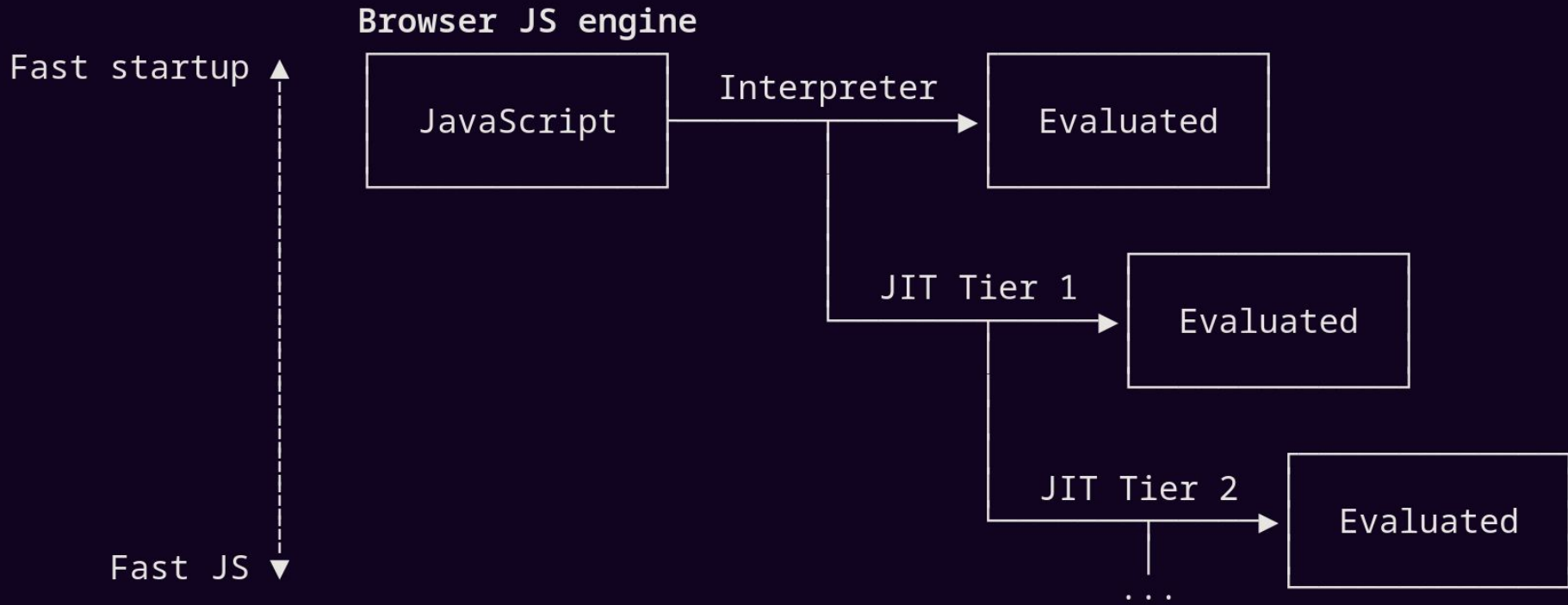


Executed

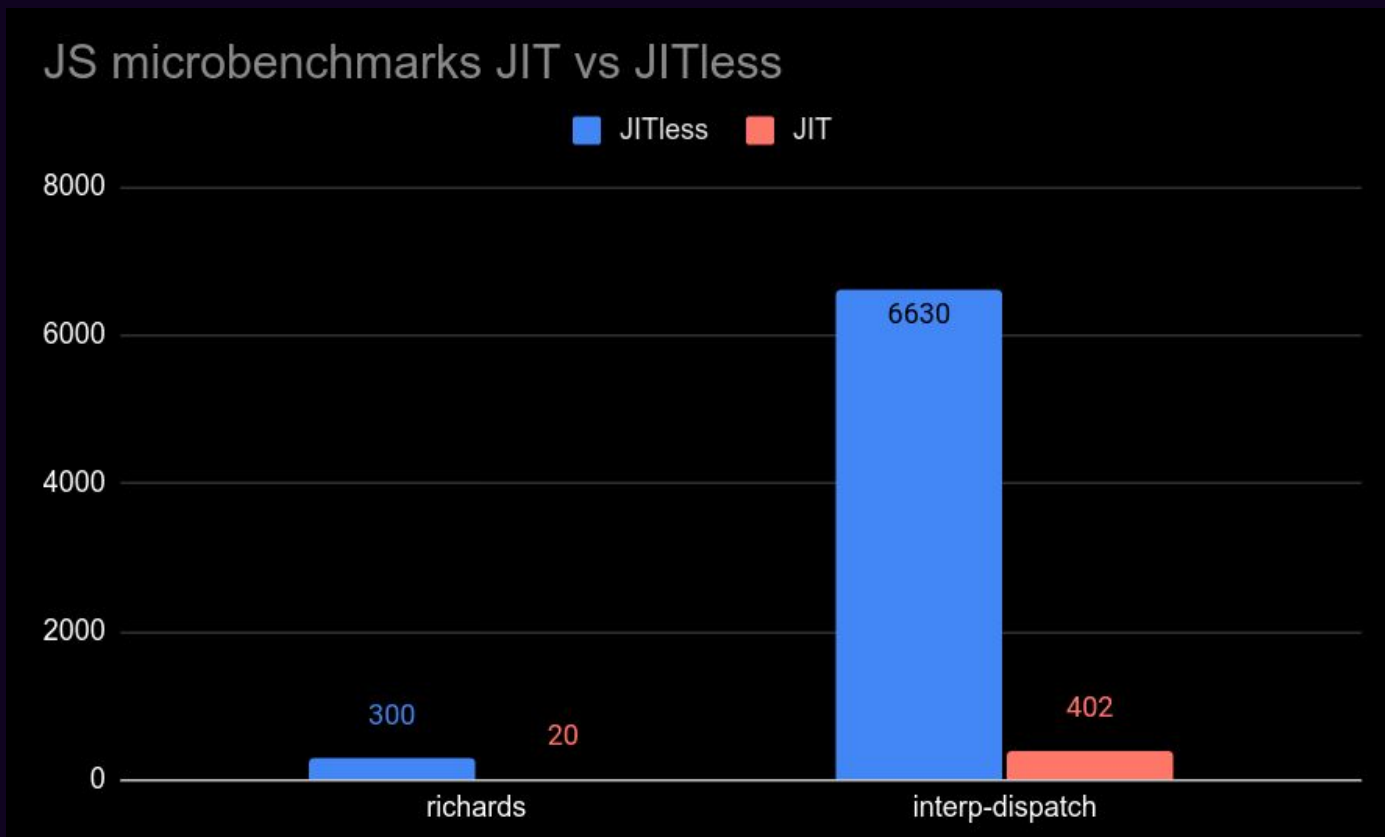
Just-in-time Compilation



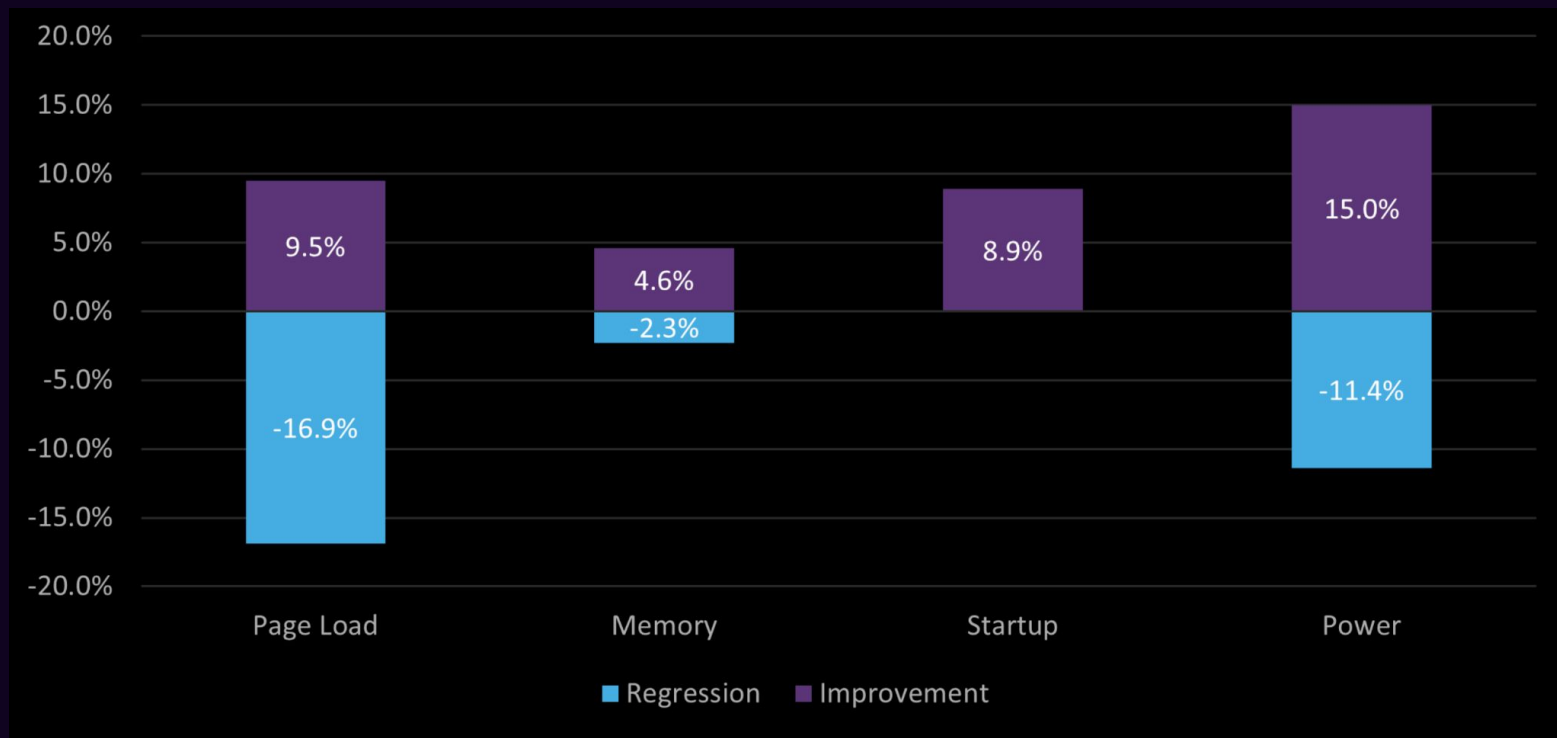
Modern JS engines



JIT pro: performance!

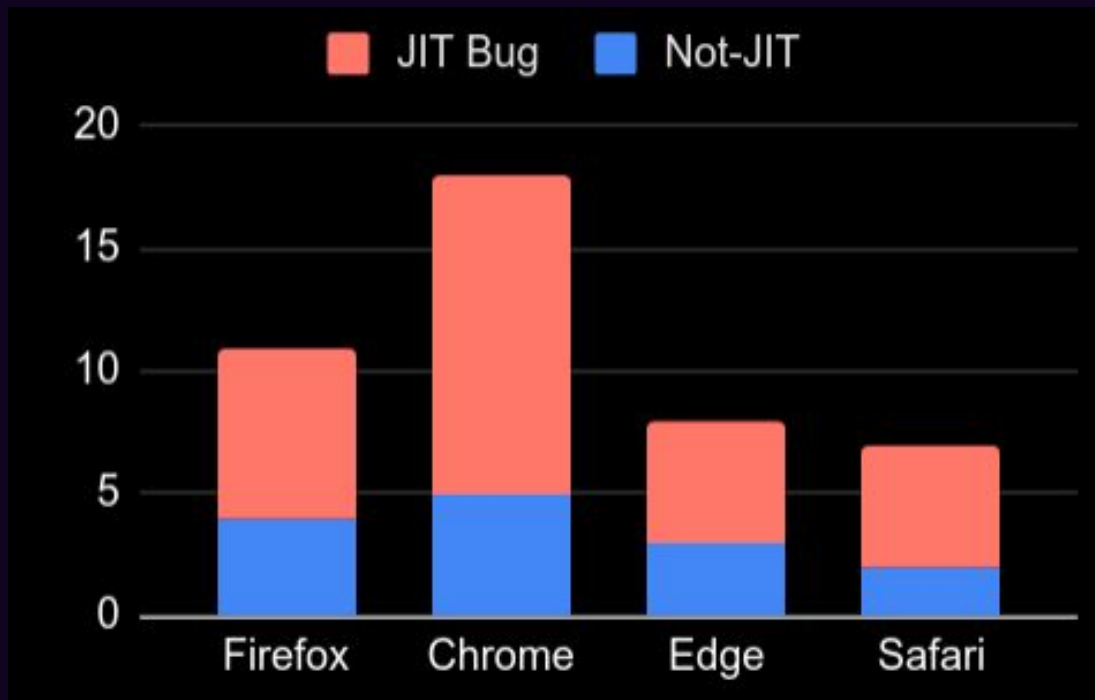


JIT pro: performance?



<https://microsoftedge.github.io/edgevr/posts/Super-Duper-Secure-Mode/>

JIT con: security





JIT con: overhead

Benchmark 1 (77 runs): node bench/richards.js

measurement	mean ± σ	min ... max	outliers	delta
wall_time	65.3ms ± 4.45ms	57.8ms ... 80.1ms	3 (4%)	0%
peak_rss	63.8MB ± 718KB	60.7MB ... 64.9MB	2 (3%)	0%

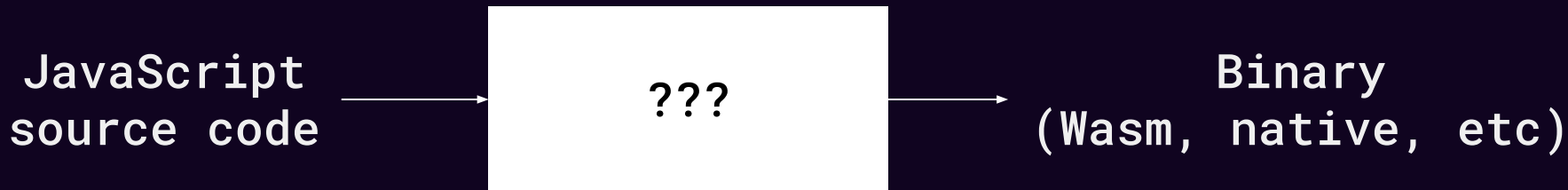
Benchmark 2 (14 runs): node --jitless bench/richards.js

measurement	mean ± σ	min ... max	outliers	delta
wall_time	365ms ± 8.31ms	356ms ... 391ms	1 (7%)	 +458.8% ± 4.6%
peak_rss	50.7MB ± 179KB	50.3MB ... 50.9MB	0 (0%)	 - 20.5% ± 0.6%

Ahead-of-time Compilation



Ahead-of-time Compilation



Porffor

JavaScript
source code



Porffor



Binary
(Wasm, native, etc)

Porffor

- Started in October 2023
- Full-time (with funding!) since August 2024

Live Demo!

Porffor's Architecture

(over-simplified)



External

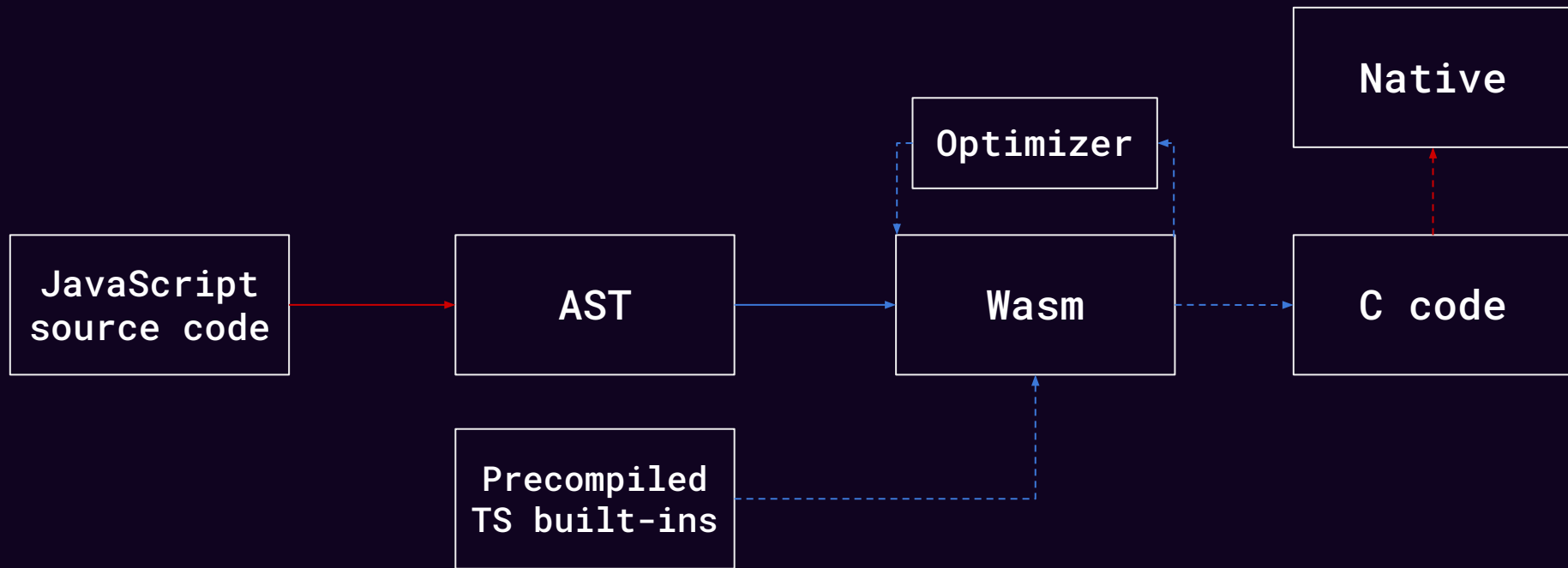


Porffor



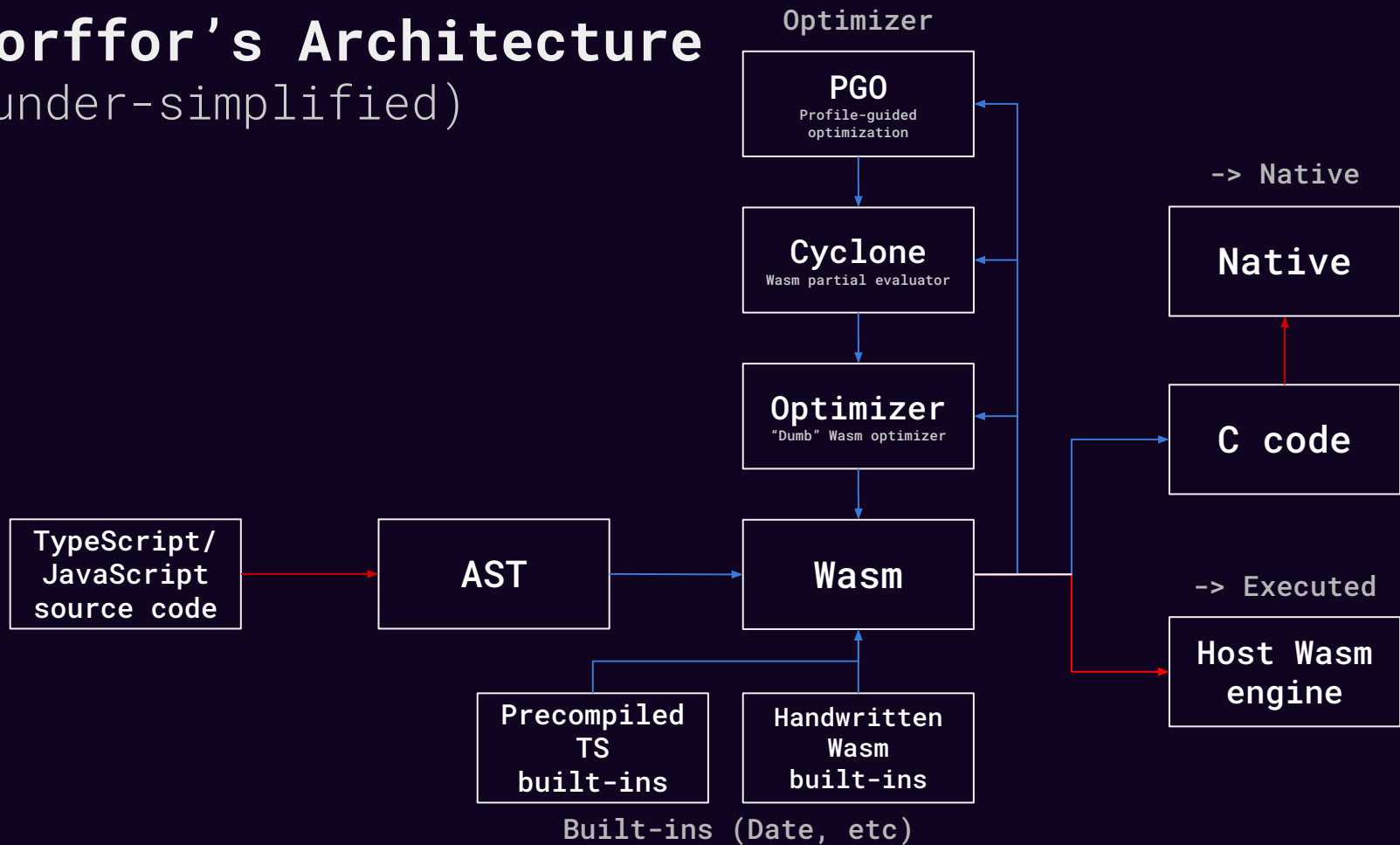
Porffor's Architecture

(simplified)



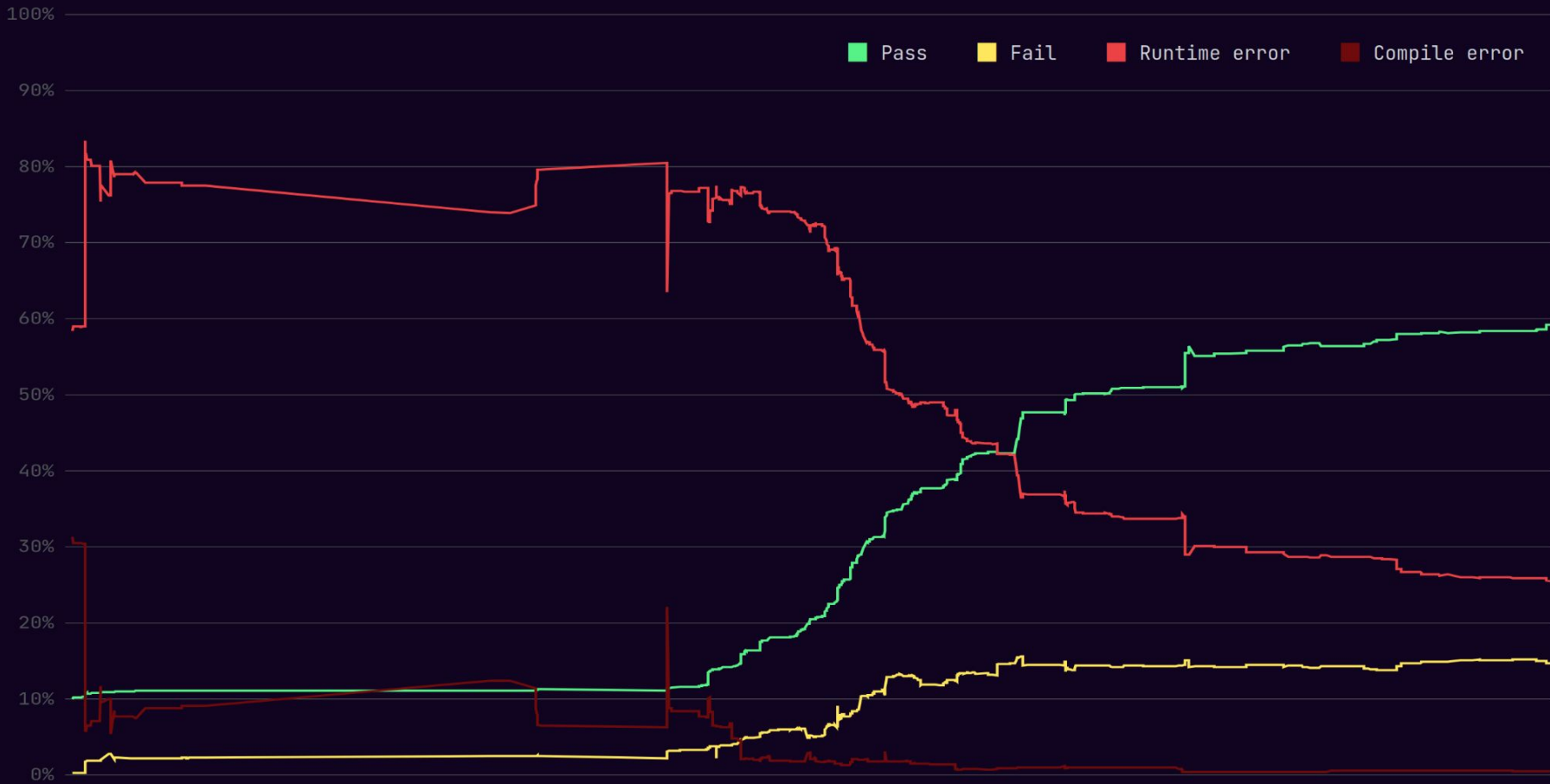
Porffor's Architecture

(under-simplified)





Test262 59.20%


Porffor is ran against Test262, the official ECMAScript conformance test suite, every commit to track conformance progress.




Vaporware?

Vaporware?




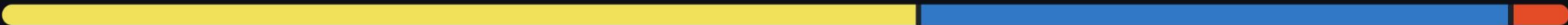
 CanadaHonk / **porffor** ☆ 3.4k ✓

[Code](#) [Issues 102](#) [Pull requests 14](#) [Discussions](#) [Actions](#) [Projects](#) [Wiki](#) [Settings](#)

 **porffor** Public Unpin Unwatch 39

main 1 Branch 0 Tags Q t + <>

 **CanadaHonk** compiler: rewrite wasm imports to be dynamic, user-defin... ... ✓ e58ec6b · 20 hours ago ⌚ 2,607 Commits



● JavaScript 58.7% ● TypeScript 37.7% ● HTML 3.6%

```
~/porffor$ token compiler --exclude builtins_precompiled.js
```

Language	Files	Lines	Code	Comments	Blanks
JavaScript	27	14598	11266	915	2417
TypeScript	33	12287	8184	1593	2510
Total	60	26885	19450	2508	4927

- promise.ts
- reflect.ts
- set.ts
- string.ts
- string_f64.ts
- stringtonumber.ts
- symbol.ts
- typedarray.js
- weakmap.ts
- weakref.ts
- weakset.ts
- z_ecma262.ts
- 2c.js
- allocator.js
- assemble.js
- builtins.js
- builtins_objects.js
- builtins_precompiled.js
- codegen.js
- cyclone.js
- disassemble.js

```
export const __Array_prototype_some = (_this: any[], callbackFn: any, thisArg: any) => {
    const len: i32 = _this.length;
    let i: i32 = 0;
    while (i < len) {
        if (!!callbackFn.call(thisArg, _this[i], i++, _this)) return true;
    }

    return false;
};

// 21.4.1.5 DaysInYear (y)
// https://tc39.es/ecma262/multipage/numbers-and-dates.html#sec-daysinyear
export const __ecma262_DaysInYear = (y: number): number => {
    // 1. Let ry be  $\mathbb{R}(y)$ .
    // 2. If (ry modulo 400) = 0, return 366F.
    if (y % 400 == 0) return 366;

    // 3. If (ry modulo 100) = 0, return 365F.
    if (y % 100 == 0) return 365;

    // 4. If (ry modulo 4) = 0, return 366F.
    if (y % 4 == 0) return 366;

    // 5. Return 365F.
    return 365;
};
```

Thanks! Questions?

Please ask anything (including JS code for a REPL)
and say hi later! Follow progress:



porffor.dev



[canadahonk/porffor](https://github.com/canadahonk/porffor)



[@canadahonk](https://twitter.com/canadahonk)



[@goose.icu](https://twitter.com/goose.icu)