WebAssembly - The Next Big Platform

Sven Sauleau

2019

JavaScript, what happened?

Loading time

1. Fetching.
2. Parsing source.
3. Compiling + optimizing $\xrightarrow{\infty}$ reoptimizing.

Performance

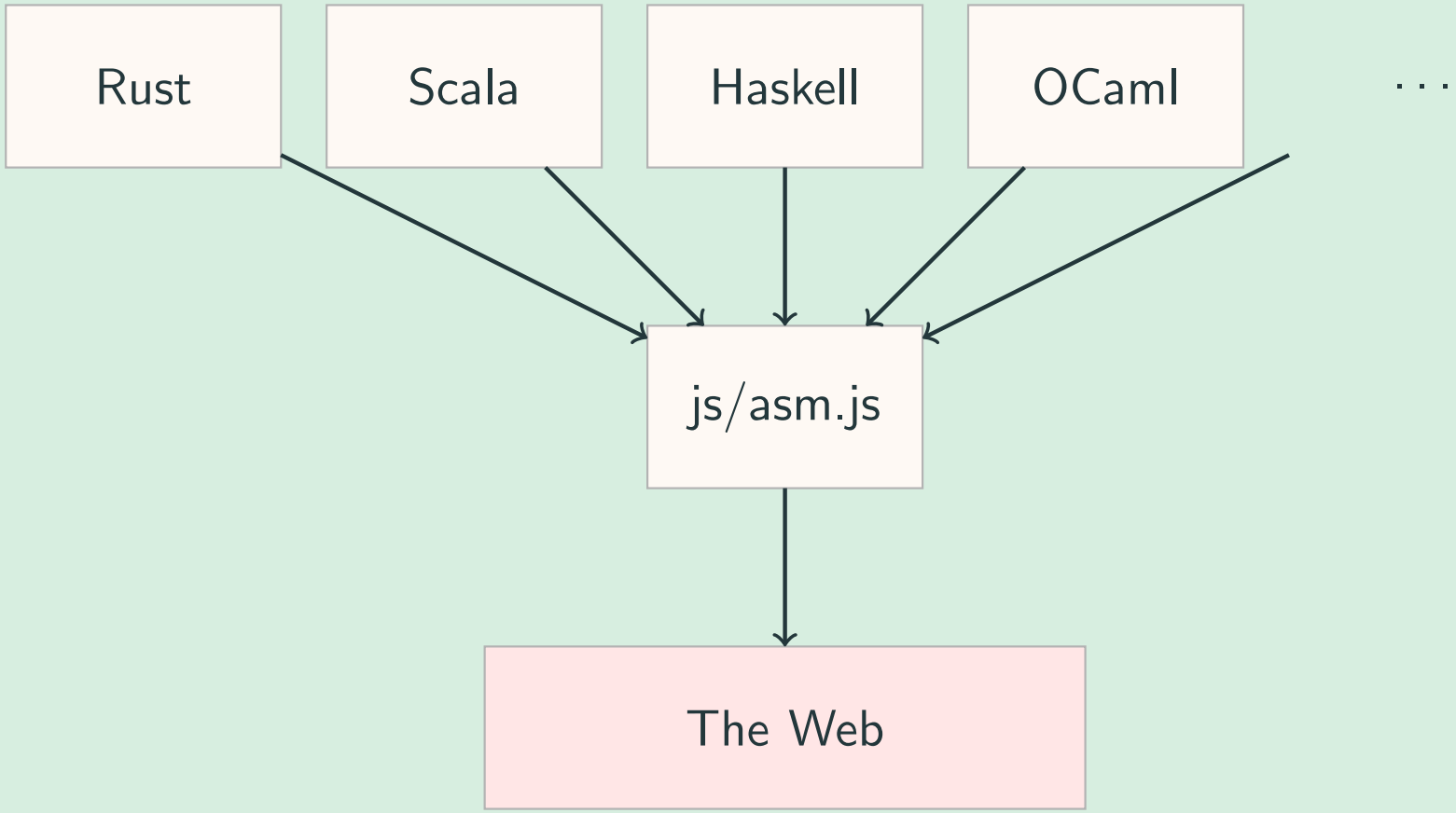
Dynamic and untyped.

Complex runtime.

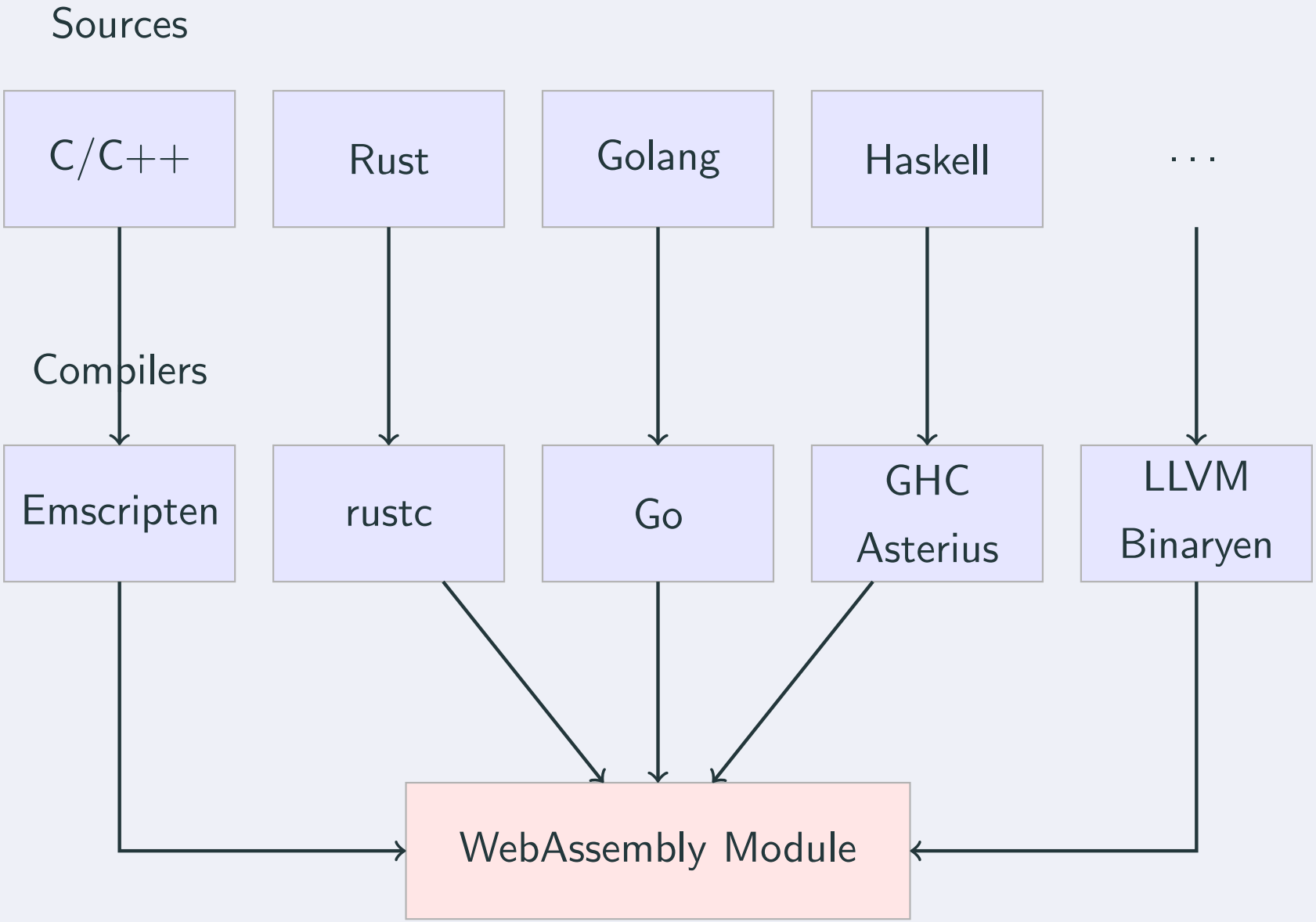
Managed memory.

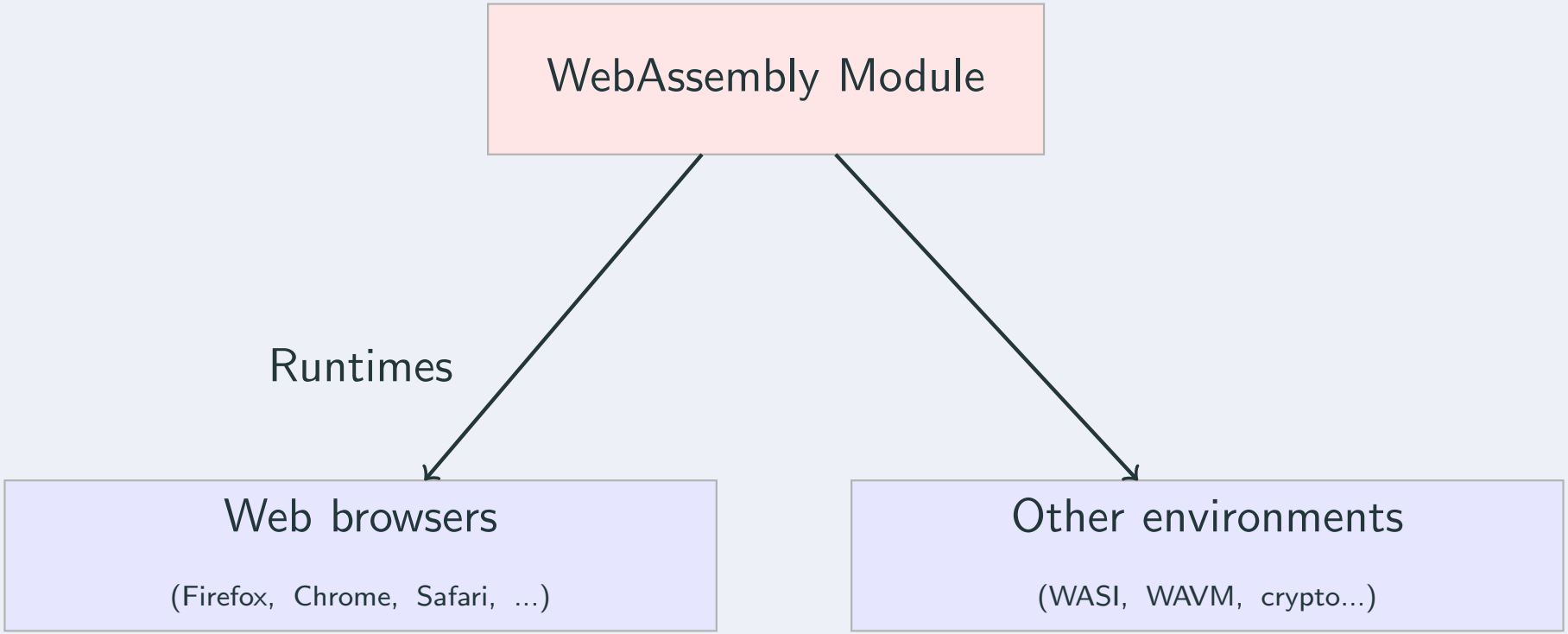
Became a compilation target

cargo build --target asmjs-unknown-emscripten


Rust
Scala
Haskell
OCaml
· · ·
js/asm.js
The Web

WebAssembly, at the rescue?


Sources
C/C++
Rust
Golang
Haskell
· · ·
Compilers
Emscripten
rustc
Go
GHC
Asterius
LLVM
Binaryen
WebAssembly Module


WebAssembly Module
Runtimes
Web browsers
(Firefox, Chrome, Safari, ...)
Other environments
(WASI, WAVM, crypto...)

What's even WebAssembly?

Safe to execute

Language/hardware/platform independent

Deterministic and easy to reason about

WebAssembly is fast

Compiled to machine code.

Static analysis.

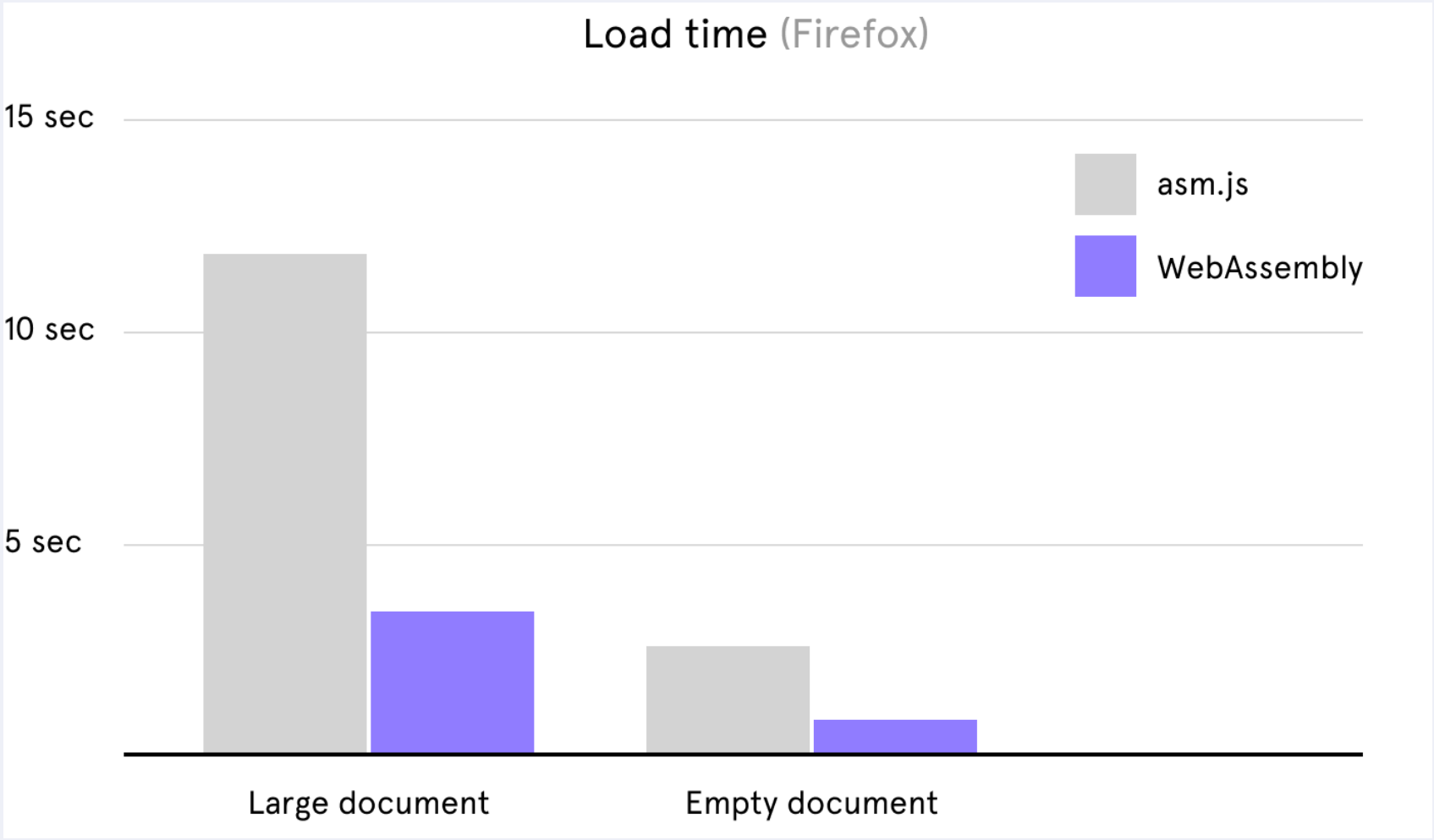
Optimized Ahead Of Time.

Crossing the boundary[1]

Can require a conversion.

Can require checks.

[1] https://hacks.mozilla.org/2018/10/calls-between-javascript-and-webassembly-are-finally-fast-%F0%9F%8E%89/



"Our load time improved by more than 3x [...]"

— Figma, medium

WebAssembly is a

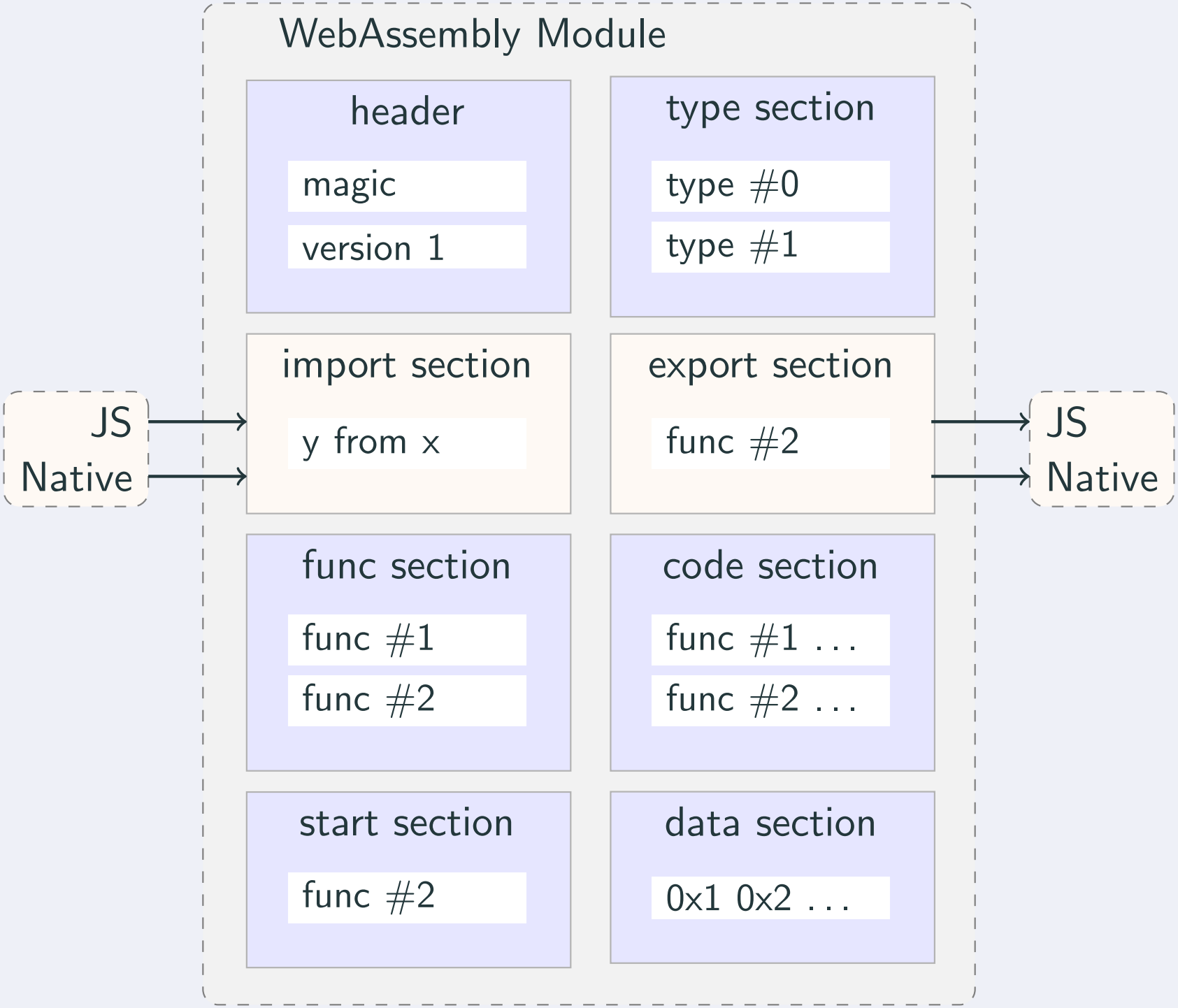
portable,

low-level,

safe

format.

Inside the black box


WebAssembly Module
header
magic
version 1
type section
type #0
type #1
import section
y from x
export section
func #2
JS
Native
JS
Native
func section
func #1
func #2
code section
func #1 ...
func #2 ...
start section
func #2
data section
0x1 0x2 ...

```
cargo build --target wasm32-unknown-unknown
```

$$
\begin{aligned}
\text{value type} &::= \text{i32} \mid \text{i64} \mid \text{f32} \mid \text{f64} \\
\text{function type} &::= [\mathit{vec}(\text{value type})] \to [\mathit{vec}(\text{value type})]
\end{aligned}
$$

Rust and WebAssembly

github.com/rustwasm/wasm-bindgen

demo!

github.com/cloudflare/wrangler

wrangler

crates.io v1.0.0 | Azure Pipelines succeeded

Get started with Cloudflare Workers and Wrangler by reading the documentation.

Installation

You have many options to install wrangler!

Using `npm`

```
npm i @cloudflare/wrangler -g
```

Using `cargo`

```
cargo install wrangler
```

github.com/xtuc/ics-parser-demo-app

Thanks!