PandelisZ

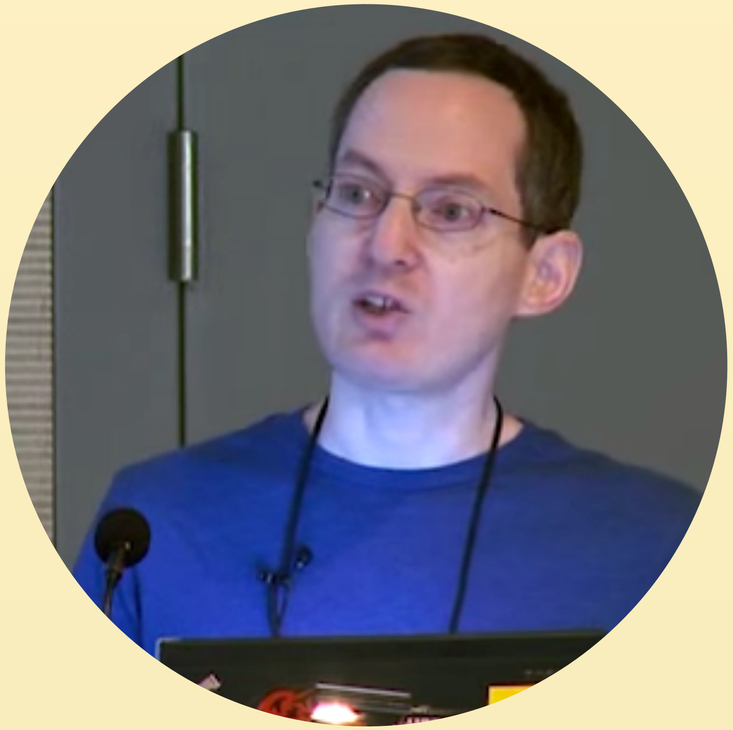
Developer at :D4 Software

Committee HaCS BCU

GitHub Campus Expert 🚩

Web Assembly

What?



Alon Zakai

moz://a

Creator of **asm.js**

Low Level Optimisations

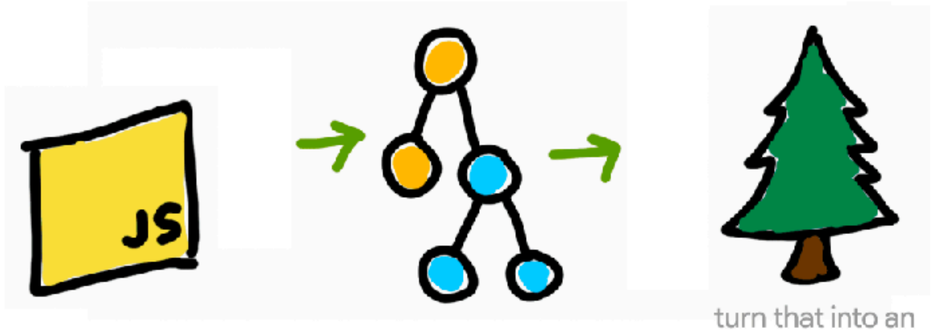
MAKE THE

interpreter's

JOB EASIER

How

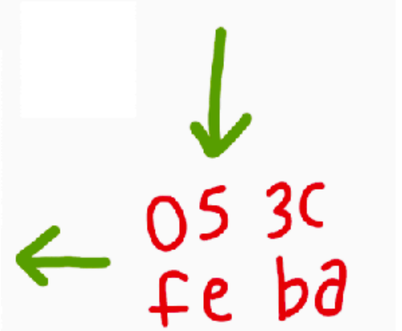
Works



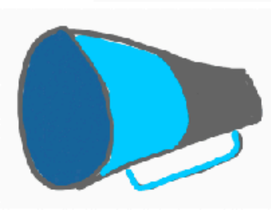
We take your JS

Parse it

turn that into an Abstract Syntax Tree



Generate bytecode

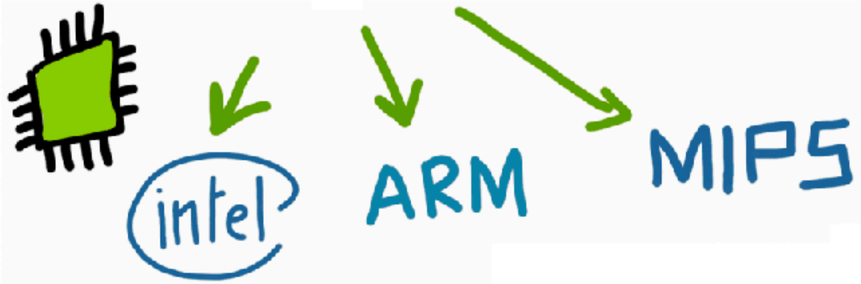


Get feedback

for speculative optimisations

TURBOFAN

Optimize & Compile it



then run your optimized code!

By @addyosmani

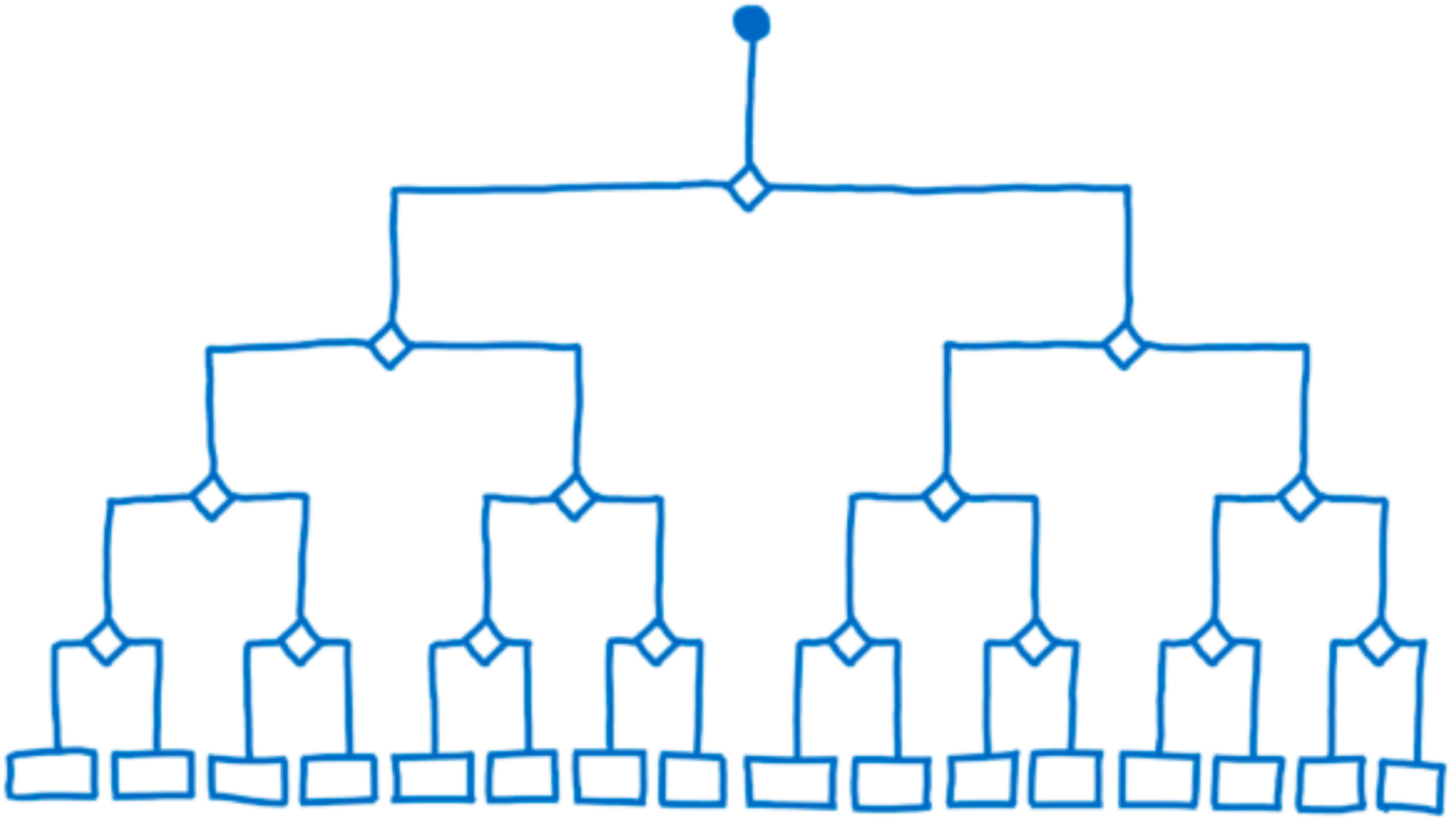
sum += arr[i]

is sum an int?

is arr an array?

is i an int?

is arr[i] an int?




Total
JavaScript
Parse
Compile
GC
Parse-Background
Average
sina.com.cn
adwords.google.com
bcc.co.uk-amp
bing.com
cnn.com
discourse.org
ebay.fr
facebook.com
google.de
inbox.google.com
instagram.com
linkedin.com
maps.google.co.jp
msn.com
qq.com
reddit.com
reddit.musicplayer.io
baidu.com
speedometer-angular
speedometer-backbone
speedometer-ember
speedometer-flight
speedometer-jquery
speedometer-vanilla
taobao.com
twitter.com
weibo.com
wikipedia.org
wikipedia.org-visual-e…
wikiwand.com
yahoo.co.jp
yandex.ru
youtube.com
youtube.com-polymer…
0%
5%
10%
15%
20%
25%
30%
35%
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%

```
Boolean foo = true;
```

Statically-typed language ✅

Compiled C/C++ is just 2X slower than native. ✅

Fast forward

Very large asm.js files. 

Needs to be parsed like JavaScript 

Its's a massive **hack**. 

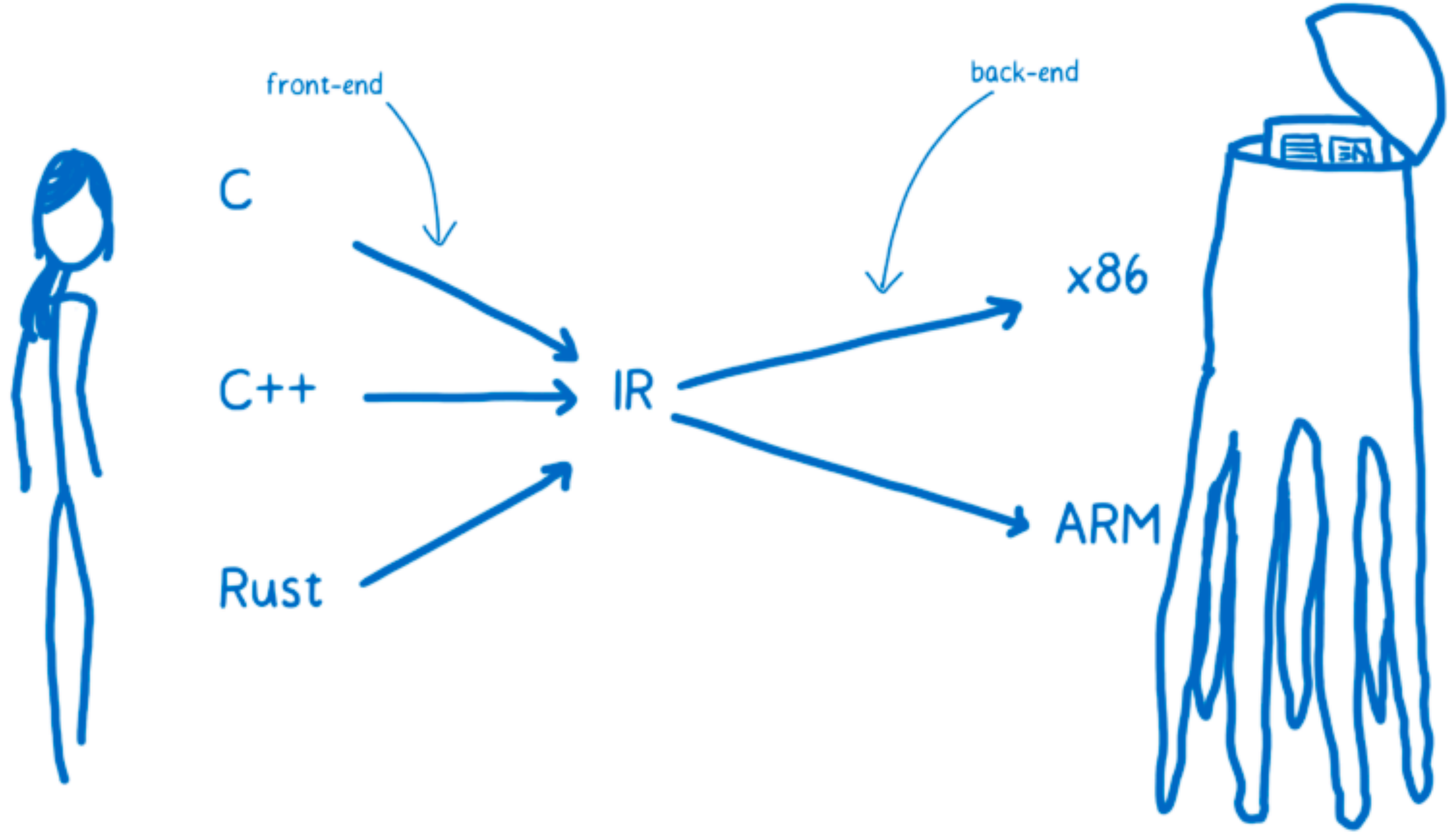
Your boys the browser vendors

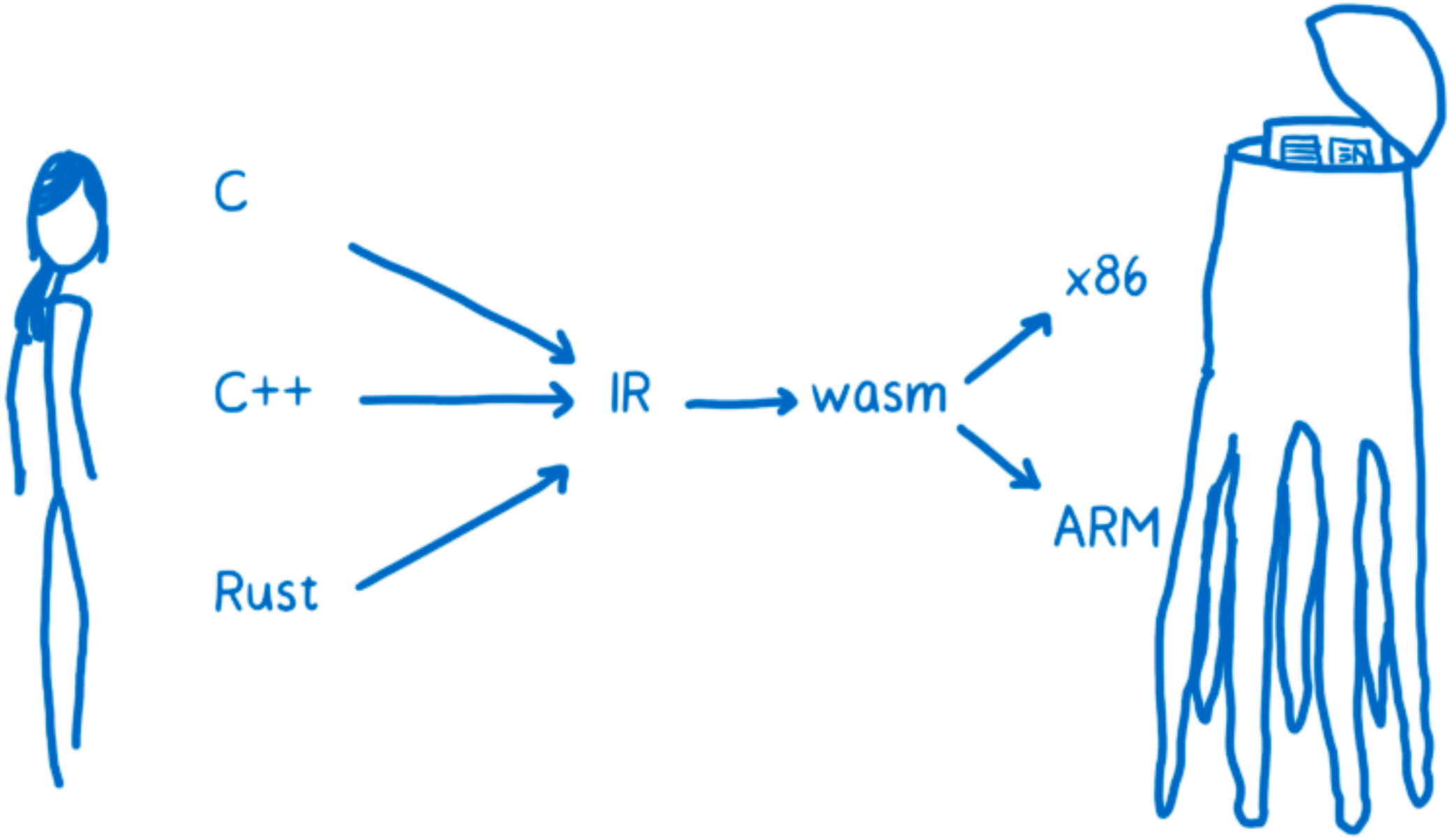
Didn’t We go through this?

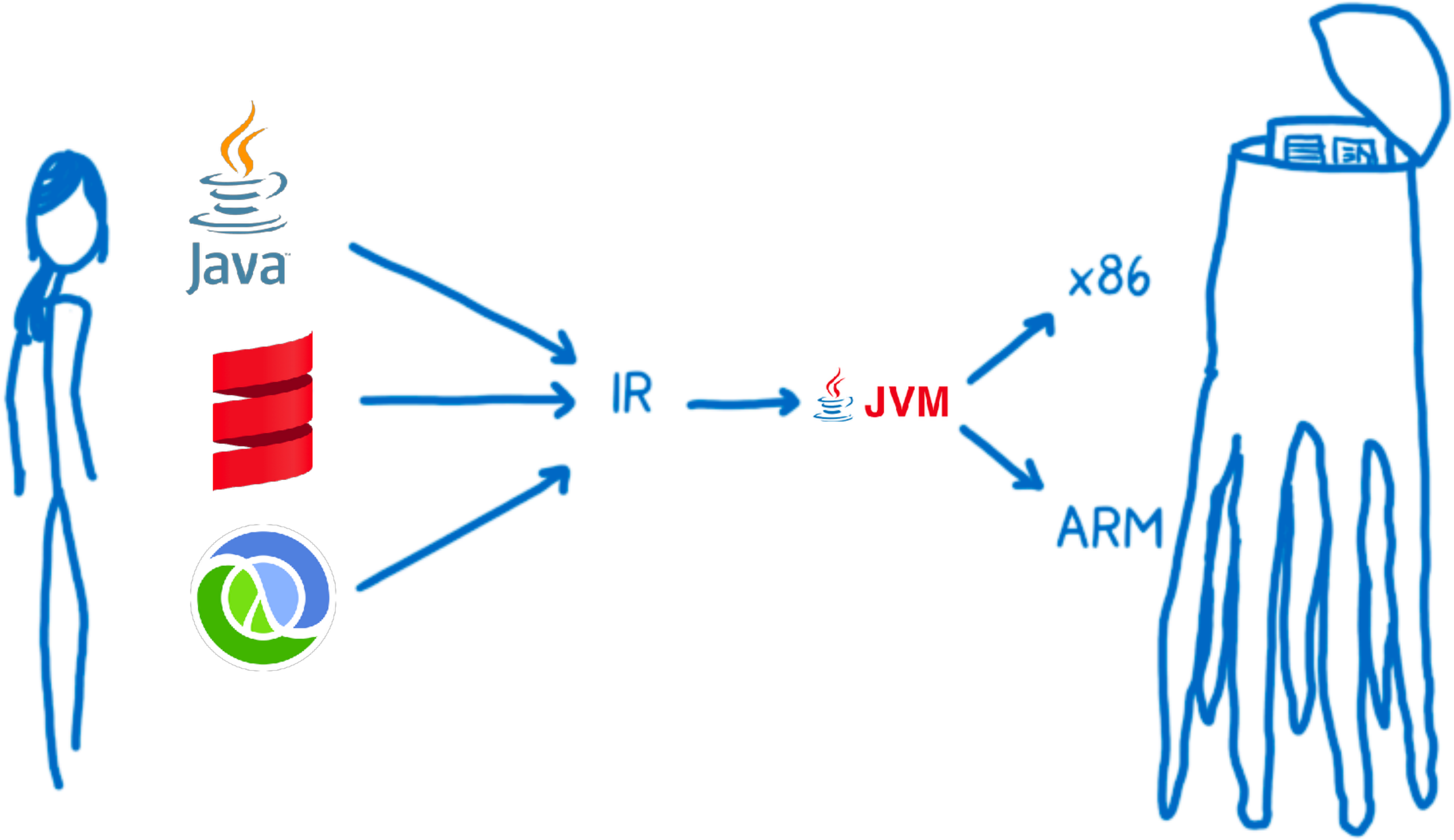

Java

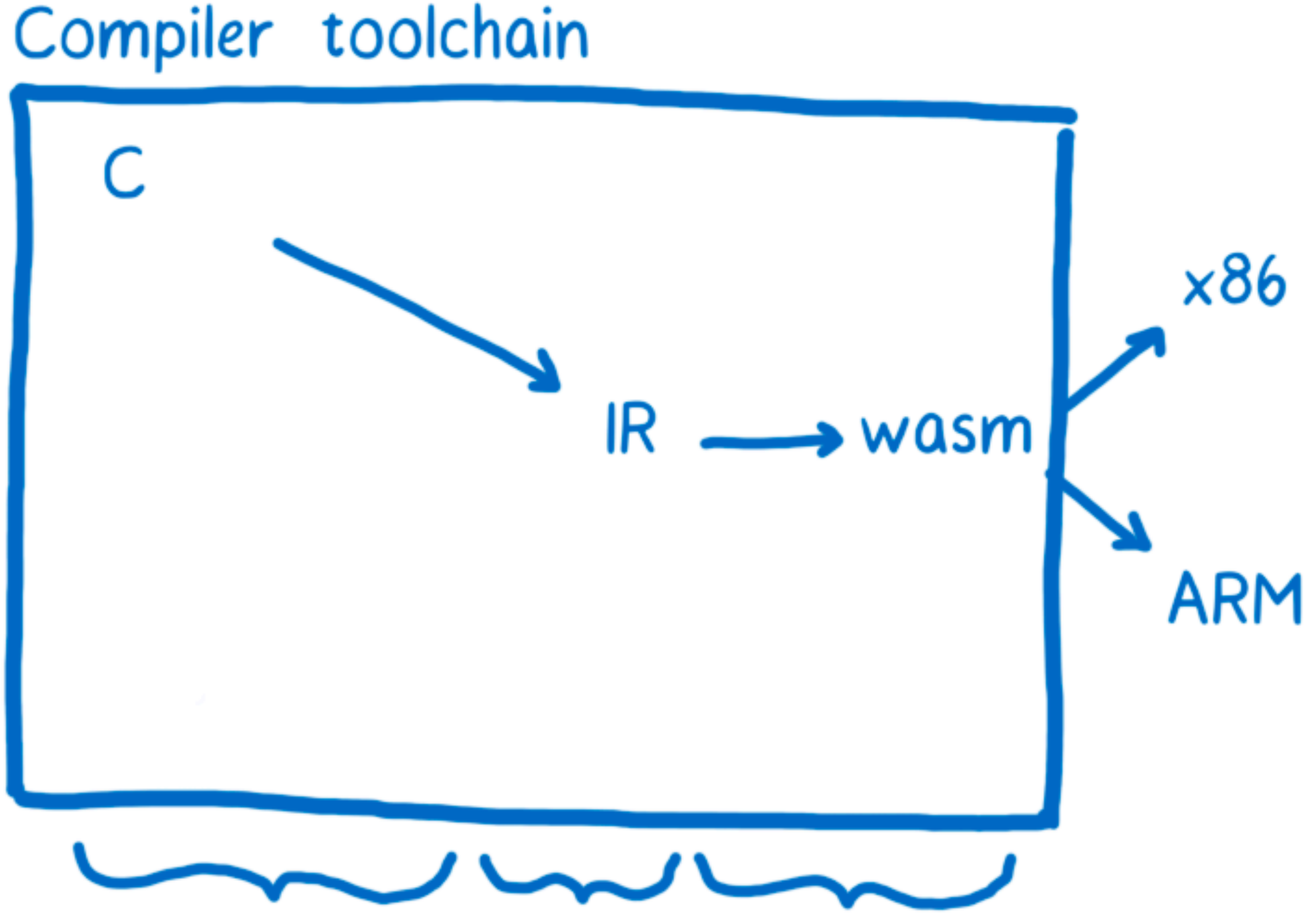
How?

Strap in for some lingo

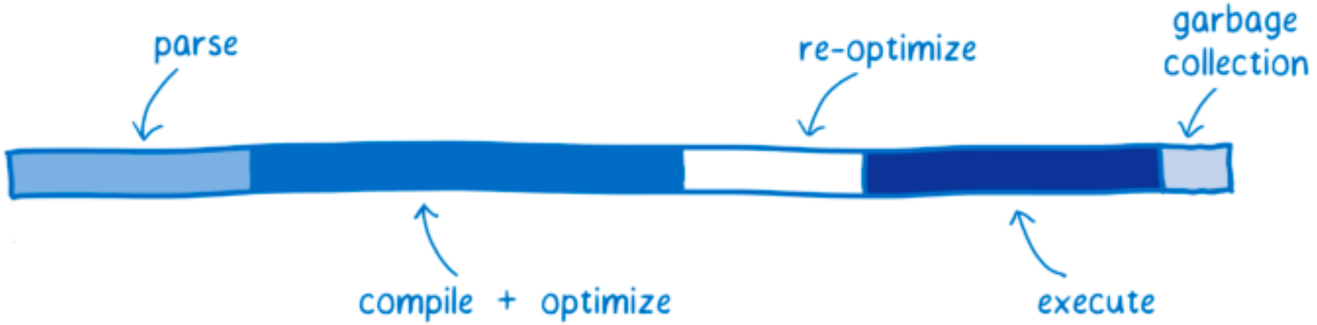

front-end
back-end
C
C++
Rust
IR
x86
ARM


C
C++
Rust
IR
wasm
x86
ARM

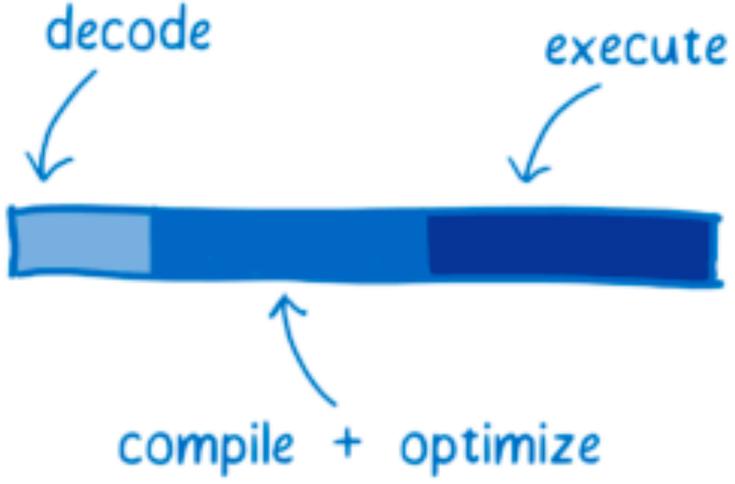

Java
IR
JVM
x86
ARM


Compiler toolchain
C
IR
wasm
x86
ARM
front-end: clang
optimizer: LLVM
back-end: LLVM WASM backend
or
Emscripten using asm2wasm

JS


parse
re-optimize
garbage collection
compile + optimize
execute

WA


decode
execute
compile + optimize

Why?

Developers want to
build stuff.
And we want to make
the **rules.**

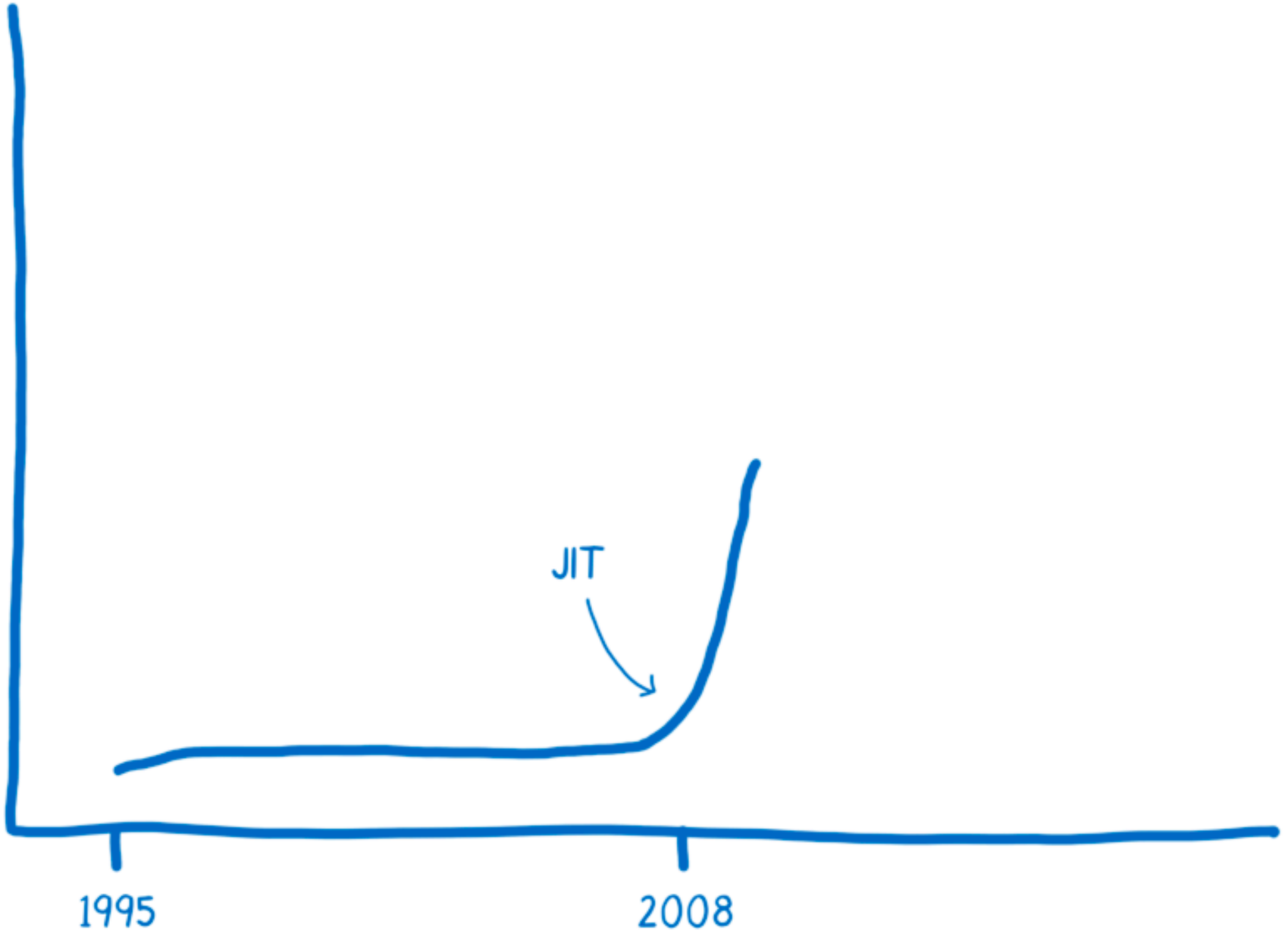
(And then break them)

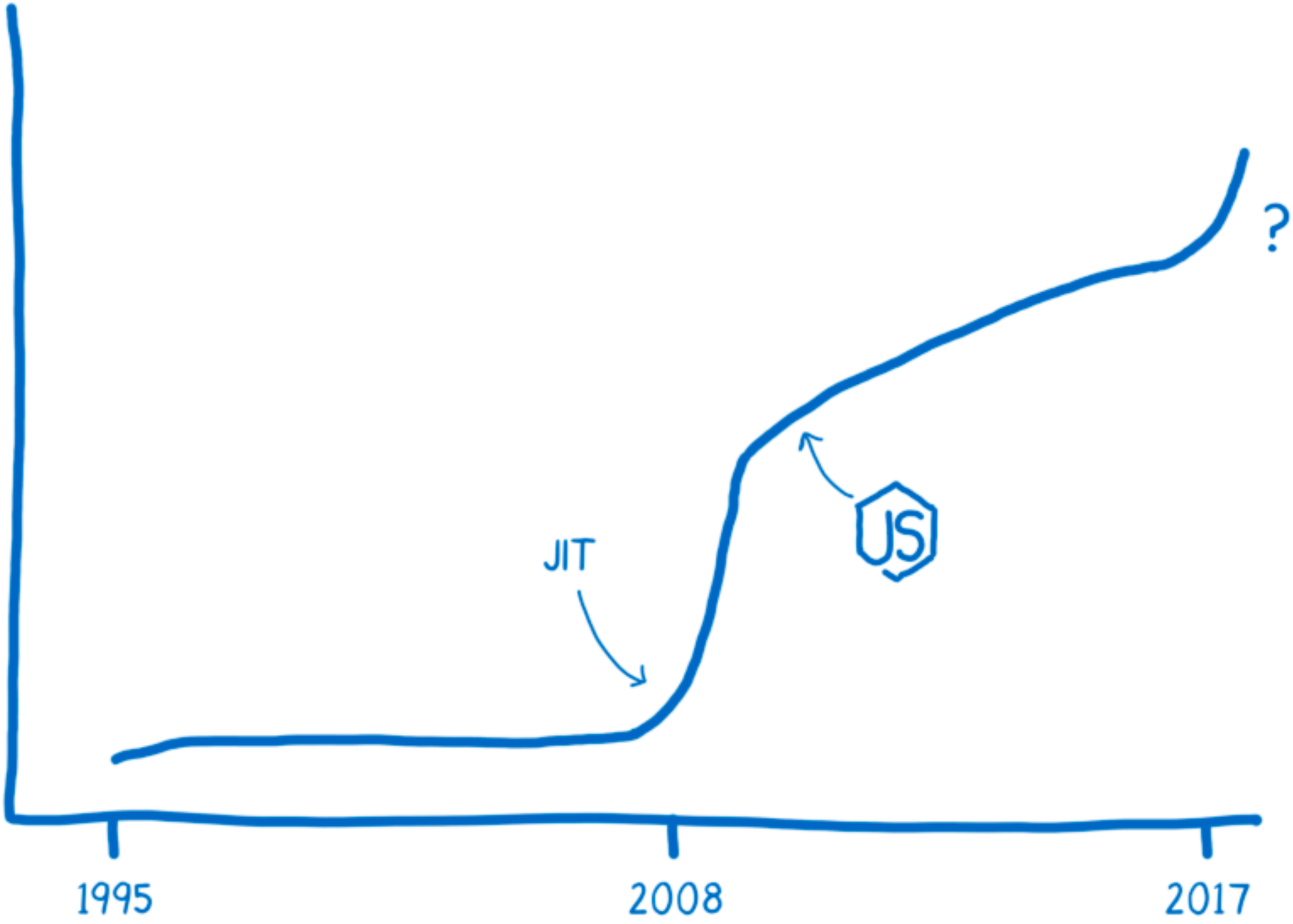
Build frameworks.


Build lots of frameworks.

React is quite **big**

Faster. ⚡


JIT
1995
2008


?
JIT
JS
1995
2008
2017

Typing Instructions

$$\boxed{C \vdash e^* : tf}$$

$$\frac{}{C \vdash t.\mathbf{const}\ c : \epsilon \to t} \qquad \frac{}{C \vdash t.unop : t \to t} \qquad \frac{}{C \vdash t.binop : t\ t \to t} \qquad \frac{}{C \vdash t.testop : t \to \mathsf{i32}} \qquad \frac{}{C \vdash t.relop : t\ t \to \mathsf{i32}}$$

$$\frac{t_1 \neq t_2 \qquad sx^? = \epsilon \Leftrightarrow (t_1 = \mathbf{in} \wedge t_2 = \mathbf{in}' \wedge |t_1| < |t_2|) \vee (t_1 = \mathbf{fn} \wedge t_2 = \mathbf{fn}')}{C \vdash t_1.\mathbf{convert}\ t_2\_sx^? : t_2 \to t_1} \qquad \frac{t_1 \neq t_2 \qquad |t_1| = |t_2|}{C \vdash t_1.\mathbf{reinterpret}\ t_2 : t_2 \to t_1}$$

$$\frac{}{C \vdash \mathbf{unreachable} : t_1^* \to t_2^*} \qquad \frac{}{C \vdash \mathbf{nop} : \epsilon \to \epsilon} \qquad \frac{}{C \vdash \mathbf{drop} : t \to \epsilon} \qquad \frac{}{C \vdash \mathbf{select} : t\ t\ \mathsf{i32} \to t}$$

$$\frac{tf = t_1^n \to t_2^m \qquad C, \mathsf{label}\ (t_2^m) \vdash e^* : tf}{C \vdash \mathbf{block}\ tf\ e^*\ \mathbf{end} : tf} \qquad \frac{tf = t_1^n \to t_2^m \qquad C, \mathsf{label}\ (t_1^n) \vdash e^* : tf}{C \vdash \mathbf{loop}\ tf\ e^*\ \mathbf{end} : tf}$$

$$\frac{tf = t_1^n \to t_2^m \qquad C, \mathsf{label}\ (t_2^m) \vdash e_1^* : tf \qquad C, \mathsf{label}\ (t_2^m) \vdash e_2^* : tf}{C \vdash \mathbf{if}\ tf\ e_1^*\ \mathbf{else}\ e_2^*\ \mathbf{end} : t_1^n\ \mathsf{i32} \to t_2^m}$$

$$\frac{C_{\mathsf{label}}(i) = t^*}{C \vdash \mathbf{br}\ i : t_1^*\ t^* \to t_2^*} \qquad \frac{C_{\mathsf{label}}(i) = t^*}{C \vdash \mathbf{br\_if}\ i : t^*\ \mathsf{i32} \to t^*} \qquad \frac{(C_{\mathsf{label}}(i) = t^*)^+}{C \vdash \mathbf{br\_table}\ i^+ : t_1^*\ t^*\ \mathsf{i32} \to t_2^*}$$

$$\frac{C_{\mathsf{return}} = t^*}{C \vdash \mathbf{return} : t_1^*\ t^* \to t_2^*} \qquad \frac{C_{\mathsf{func}}(i) = tf}{C \vdash \mathbf{call}\ i : tf} \qquad \frac{tf = t_1^* \to t_2^* \qquad C_{\mathsf{table}} = n}{C \vdash \mathbf{call\_indirect}\ tf : t_1^*\ \mathsf{i32} \to t_2^*}$$

$$\frac{C_{\mathsf{local}}(i) = t}{C \vdash \mathbf{get\_local}\ i : \epsilon \to t} \qquad \frac{C_{\mathsf{local}}(i) = t}{C \vdash \mathbf{set\_local}\ i : t \to \epsilon} \qquad \frac{C_{\mathsf{local}}(i) = t}{C \vdash \mathbf{tee\_local}\ i : t \to t} \qquad \frac{C_{\mathsf{global}}(i) = \mathsf{mut}^?\ t}{C \vdash \mathbf{get\_global}\ i : \epsilon \to t} \qquad \frac{C_{\mathsf{global}}(i) = \mathsf{mut}\ t}{C \vdash \mathbf{set\_global}\ i : t \to \epsilon}$$

$$\frac{C_{\mathsf{memory}} = n \qquad 2^a \leq (|tp| <)^?|t| \qquad (tp\_sz)^? = \epsilon \vee t = \mathbf{im}}{C \vdash t.\mathbf{load}\ (tp\_sz)^?\ a\ o : \mathsf{i32} \to t} \qquad \frac{C_{\mathsf{memory}} = n \qquad 2^a \leq (|tp| <)^?|t| \qquad tp^? = \epsilon \vee t = \mathbf{im}}{C \vdash t.\mathbf{store}\ tp^?\ a\ o : \mathsf{i32}\ t \to \epsilon}$$

$$\frac{C_{\mathsf{memory}} = n}{C \vdash \mathbf{current\_memory} : \epsilon \to \mathsf{i32}} \qquad \frac{C_{\mathsf{memory}} = n}{C \vdash \mathbf{grow\_memory} : \mathsf{i32} \to \mathsf{i32}}$$

$$\frac{}{C \vdash \epsilon : \epsilon \to \epsilon} \qquad \frac{C \vdash e_1^* : t_1^* \to t_2^* \qquad C \vdash e_2 : t_2^* \to t_3^*}{C \vdash e_1^*\ e_2 : t_1^* \to t_3^*} \qquad \frac{C \vdash e^* : t_1^* \to t_2^*}{C \vdash e^* : t^*\ t_1^* \to t^*\ t_2^*}$$

Typing Modules

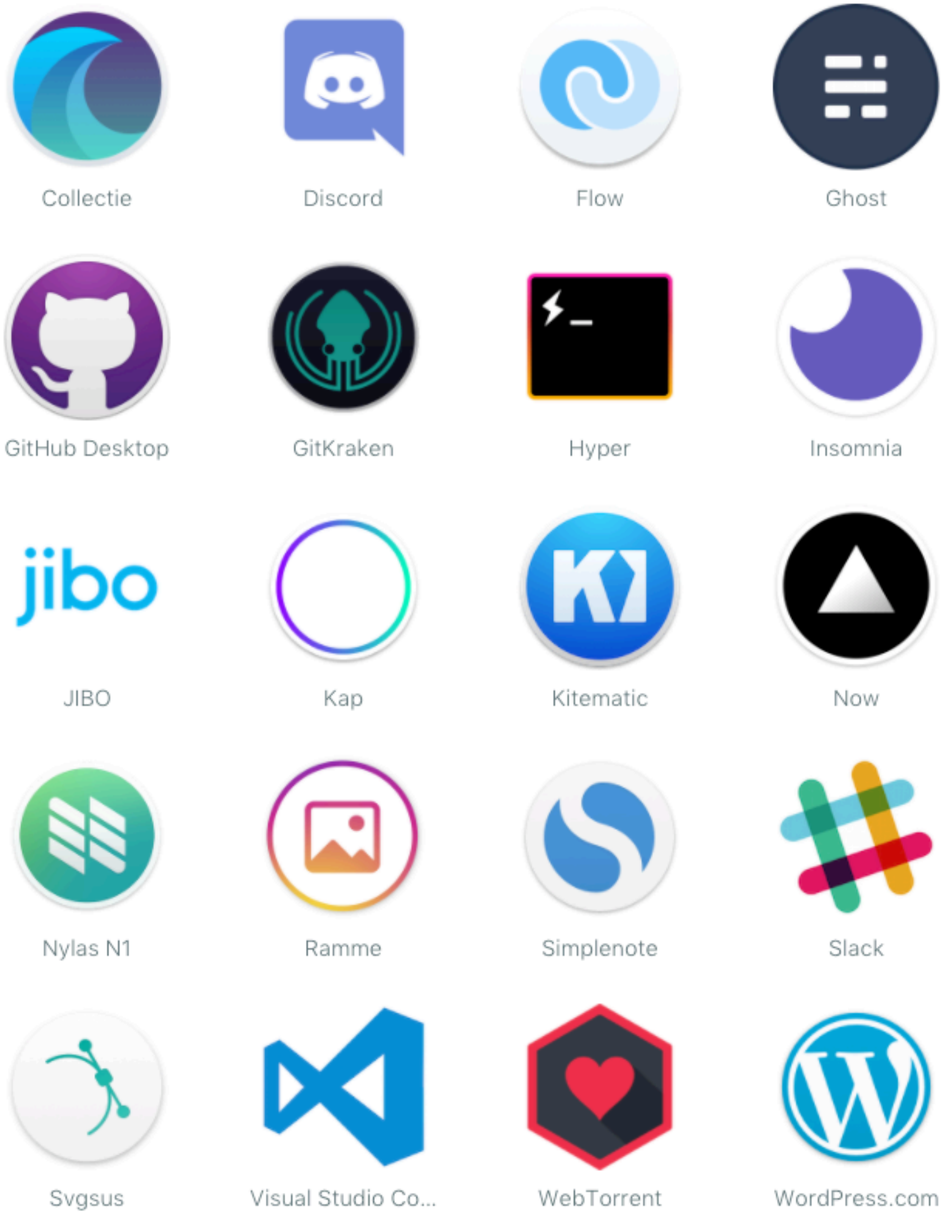
$$\frac{tf = t_1^* \to t_2^* \qquad C, \mathsf{local}\ t_1^*\ t^*, \mathsf{label}\ (t_2^*), \mathsf{return}\ (t_2^*) \vdash e^* : \epsilon \to t_2^*}{C \vdash ex^*\ \mathbf{func}\ tf\ \mathbf{local}\ t^*\ e^* : ex^*\ tf} \qquad \frac{tg = \mathsf{mut}^?\ t \qquad C \vdash e^* : \epsilon \to t \qquad ex^* = \epsilon \vee tg = t}{C \vdash ex^*\ \mathbf{global}\ tg\ e^* : ex^*\ tg}$$

$$\frac{(C_{\mathsf{func}}(i) = tf)^n}{C \vdash ex^*\ \mathbf{table}\ n\ i^n : ex^*\ n} \qquad \frac{}{C \vdash ex^*\ \mathbf{memory}\ n : ex^*\ n}$$

$$\frac{}{C \vdash ex^*\ \mathbf{func}\ tf\ im : ex^*\ tf} \qquad \frac{tg = t}{C \vdash ex^*\ \mathbf{global}\ tg\ im : ex^*\ tg} \qquad \frac{}{C \vdash ex^*\ \mathbf{table}\ n\ im : ex^*\ n} \qquad \frac{}{C \vdash ex^*\ \mathbf{memory}\ n\ im : ex^*\ n}$$

$$\frac{\begin{array}{c}(C \vdash f : ex_{\mathrm{f}}^*\ tf)^* \qquad (C_i \vdash glob_i : ex_{\mathrm{g}}^*\ tg_i)_i^* \qquad (C \vdash tab : ex_{\mathrm{t}}^*\ n)^? \qquad (C \vdash mem : ex_{\mathrm{m}}^*\ n)^? \\ (C_i = \{\mathsf{global}\ tg^{i-1}\})_i^* \qquad C = \{\mathsf{func}\ tf^*, \mathsf{global}\ tg^*, \mathsf{table}\ n^?, \mathsf{memory}\ n^?\} \qquad ex_{\mathrm{f}}^{**}\ ex_{\mathrm{g}}^{**}\ ex_{\mathrm{t}}^{*?}\ ex_{\mathrm{m}}^{*?}\ \mathrm{distinct}\end{array}}{\vdash \mathbf{module}\ f^*\ glob^*\ tab^?\ mem^?}$$

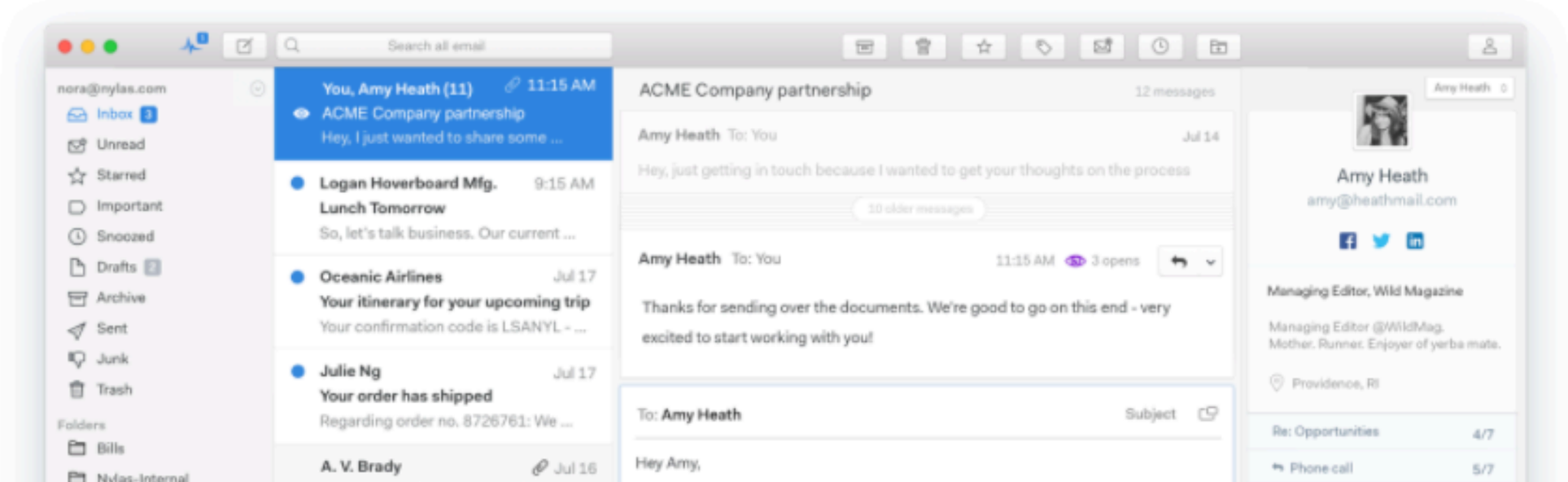
The desktop is still moving to the web


Collectie
Discord
Flow
Ghost
GitHub Desktop
GitKraken
Hyper
Insomnia
jibo
JIBO
Kap
Kitematic
Now
Nylas N1
Ramme
Simplenote
Slack
Svgsus
Visual Studio Co...
WebTorrent
WordPress.com

Mailspring

Leaving Nylas Mail? Mailspring is a new version by one of the original authors. It's faster, leaner, and shipping today! Mailspring replaces the JavaScript mailsync code in Nylas Mail with a new C++ sync engine based on Mailcore2. It uses roughly half the RAM and CPU of Nylas Mail and idles with almost zero "CPU Wakes", which translates to great battery life. A major overhaul of the package manager and dependency tree mean it launches faster too. You might not even notice it's an Electron app!

Mailspring is built on the modern web with Electron, React, and Flux. It is designed to be extensible, so it's easy to create new experiences and workflows around email. Want to learn more? Check out the full documentation.



new ***C++*** sync engine based on Mailcore2. It uses roughly ***half the RAM and CPU*** of Nylas Mail and idles with almost zero "CPU Wakes"

A major overhaul of the package manager and dependency tree mean it launches faster too. ***You might not even notice it's an Electron app!***

Facebook

iOS · Android

Using React Native in the Facebook App

Facebook Ads Manager

iOS · Android

How We Built the First Cross-Platform React Native App

Instagram

iOS · Android

React Native at Instagram

F8

iOS · Android

Tutorial: Building the F8 conference app

Airbnb

iOS · Android

Hybrid React Native Apps at Airbnb

Skype

iOS · Android

Tesla

iOS · Android

Walmart

iOS · Android

React Native at Walmart Labs

Artsy

iOS

React Native at Artsy

Baidu Mobile (手机百度)

iOS · Android

Baidu Mobile is a search engine used by over 600 million people in China

Bloomberg

iOS · Android

How Bloomberg Used React Native to Develop its new Consumer App

CBS Sports Franchise Football

Android

Award winning Fantasy Football league manager built with React Native

Service worker

Background sync

Web VR

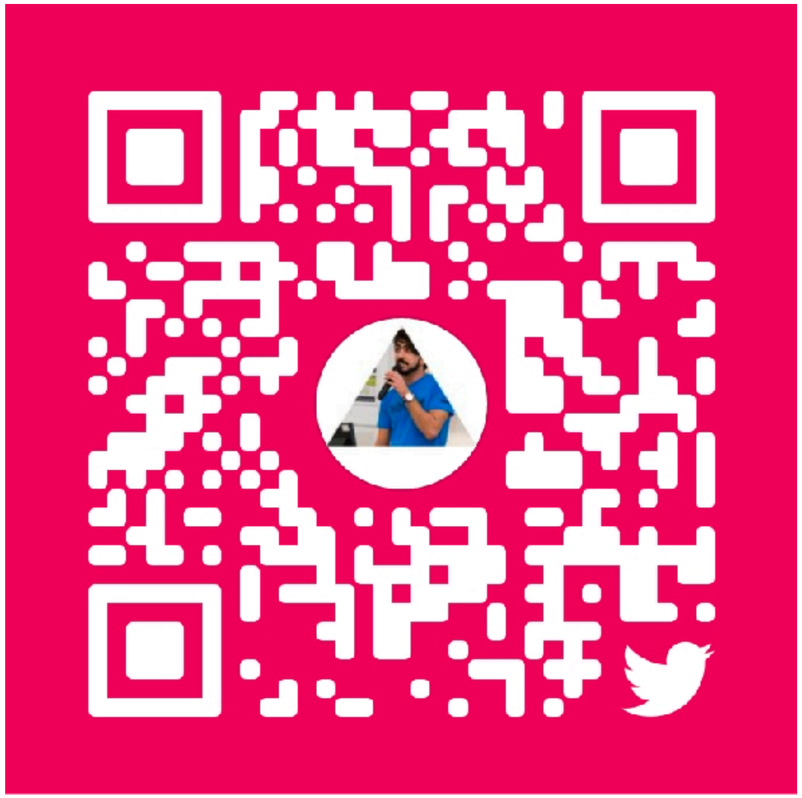
Async Iteration ESNext	In Development
CSS Font Display	In Development
File and Directory Entries API	In Development
Geometry Interfaces	In Development
Intersection Observer	In Development
Navigation Timing Level 2	In Development
Object rest/spread ESNext	In Development
Payment Request	In Development
Readable Streams Streams	In Development
Service Workers	In Development
Variation Fonts CSS Fonts Level 4	In Development
Web Audio	In Development
WebVR	In Development

Conic Gradients CSS Image Values and Replaced Content Module Level 4	Under Consideration
CSS Painting API Level 1	Under Consideration
CSS Properties and Values API Level 1	Under Consideration
CSS Rhythmic Sizing	Under Consideration
CSS Text Decoration Level 4	Under Consideration
ImageBitmap HTML5	Under Consideration
requestIdleCallback	Under Consideration
Scroll Anchoring	Under Consideration
SVG in OpenType Fonts	Under Consideration
ViewportAPI	Under Consideration
Web Animations	Under Consideration
Web App Manifest	Under Consideration
Web Authentication	Under Consideration
Web Share	Under Consideration

JS Can’t keep up forever

Let’s C what

It can do



PandelisZ

github.com/pandelisz/slides

github.com/mbasso/awesome-wasm

Sketches: Lin Clark

How v8 works: @AddyOsmani