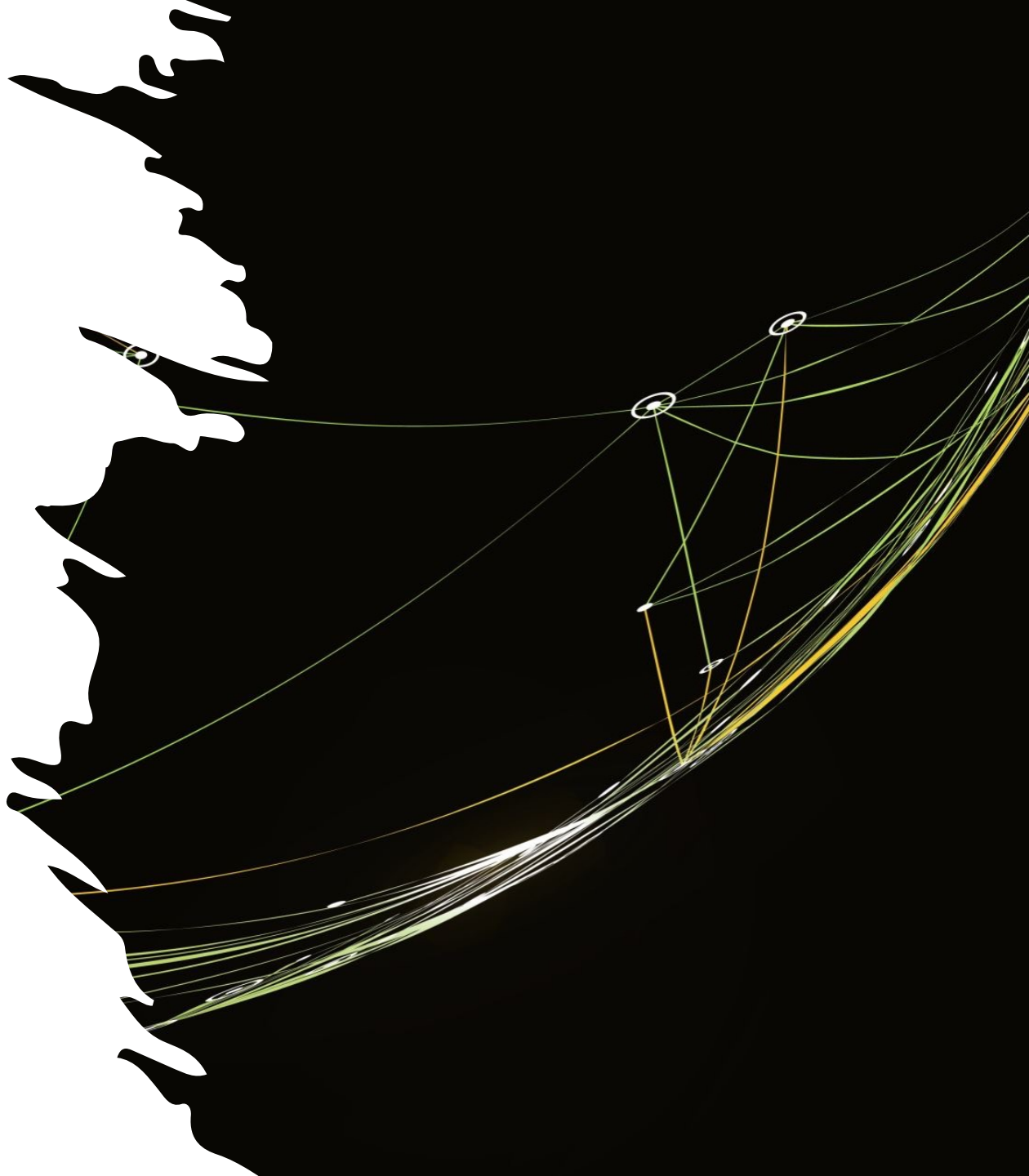
Beyond the browser

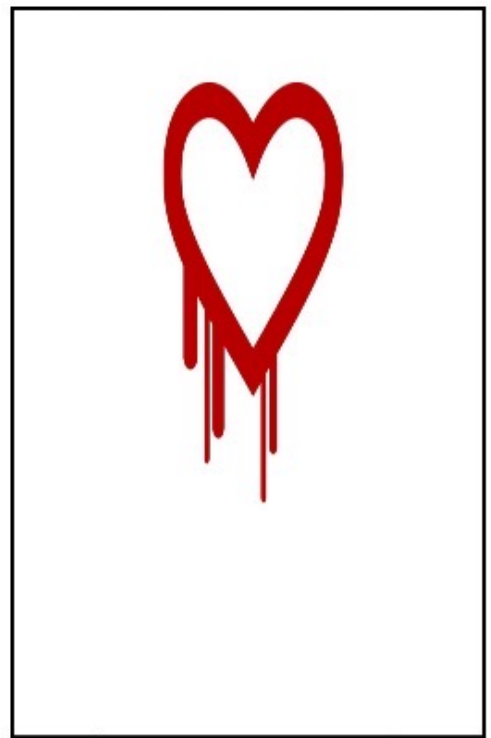
Toward a more declarative future with WebAssembly

Hi there 👋👋

- Senior Technical Evangelist @ SUSE
- CNCF Ambassador
- Co-creator of the KCNA certification
- Cloud Native Wasm Day co-chair
- Documentation maintainer for Kubernetes

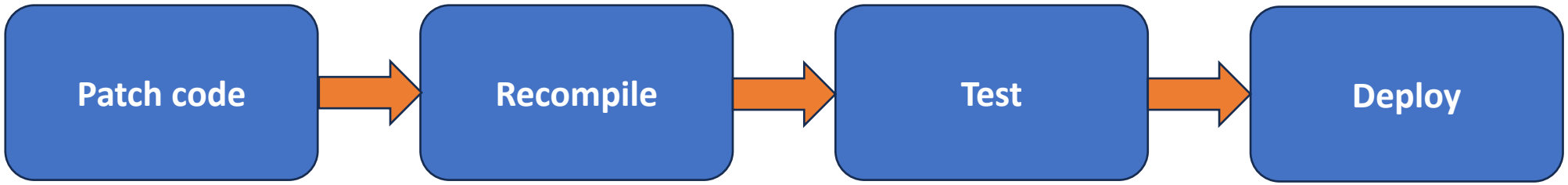
How we build software is broken



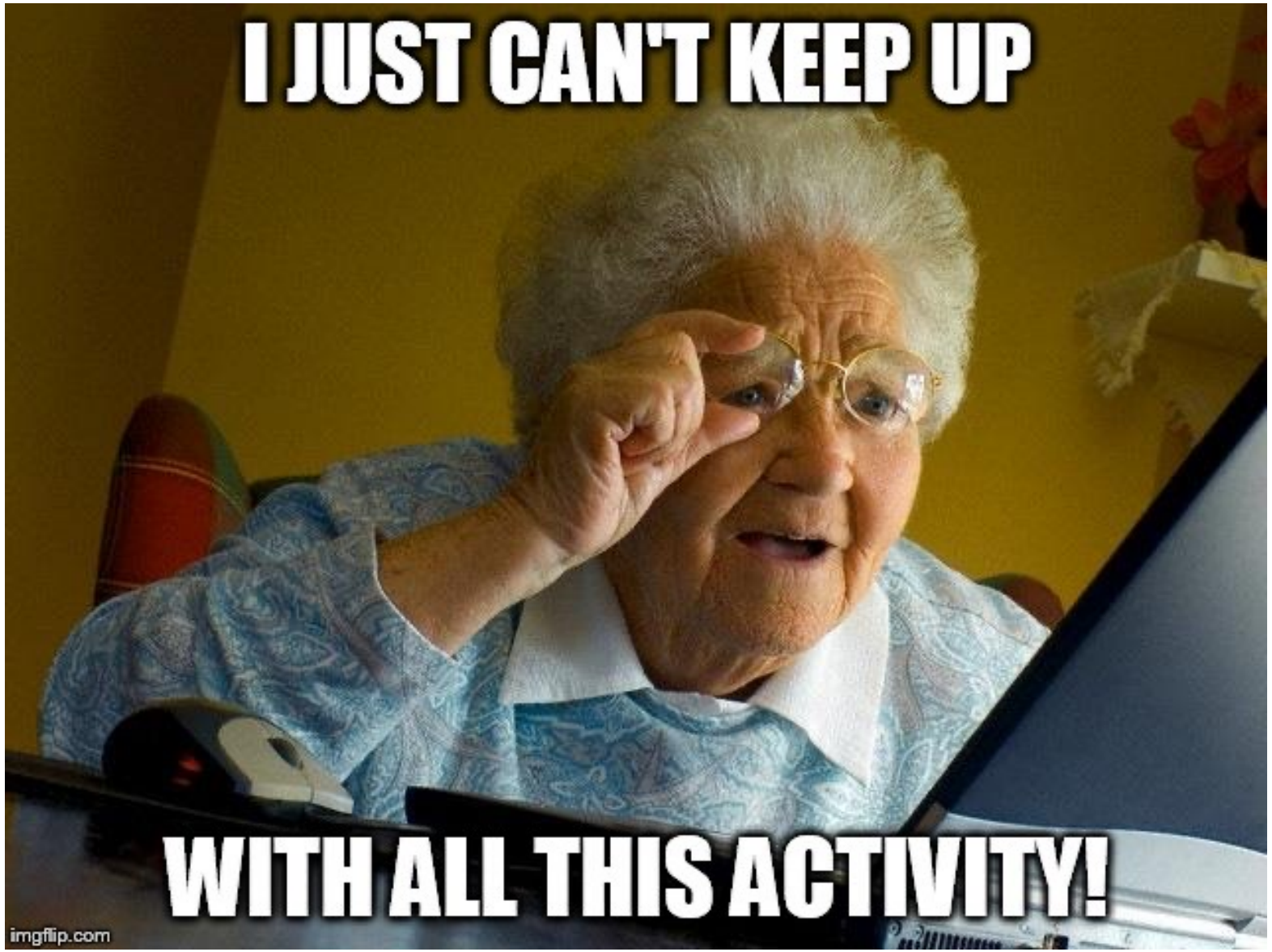


who are you?

im you but stronger

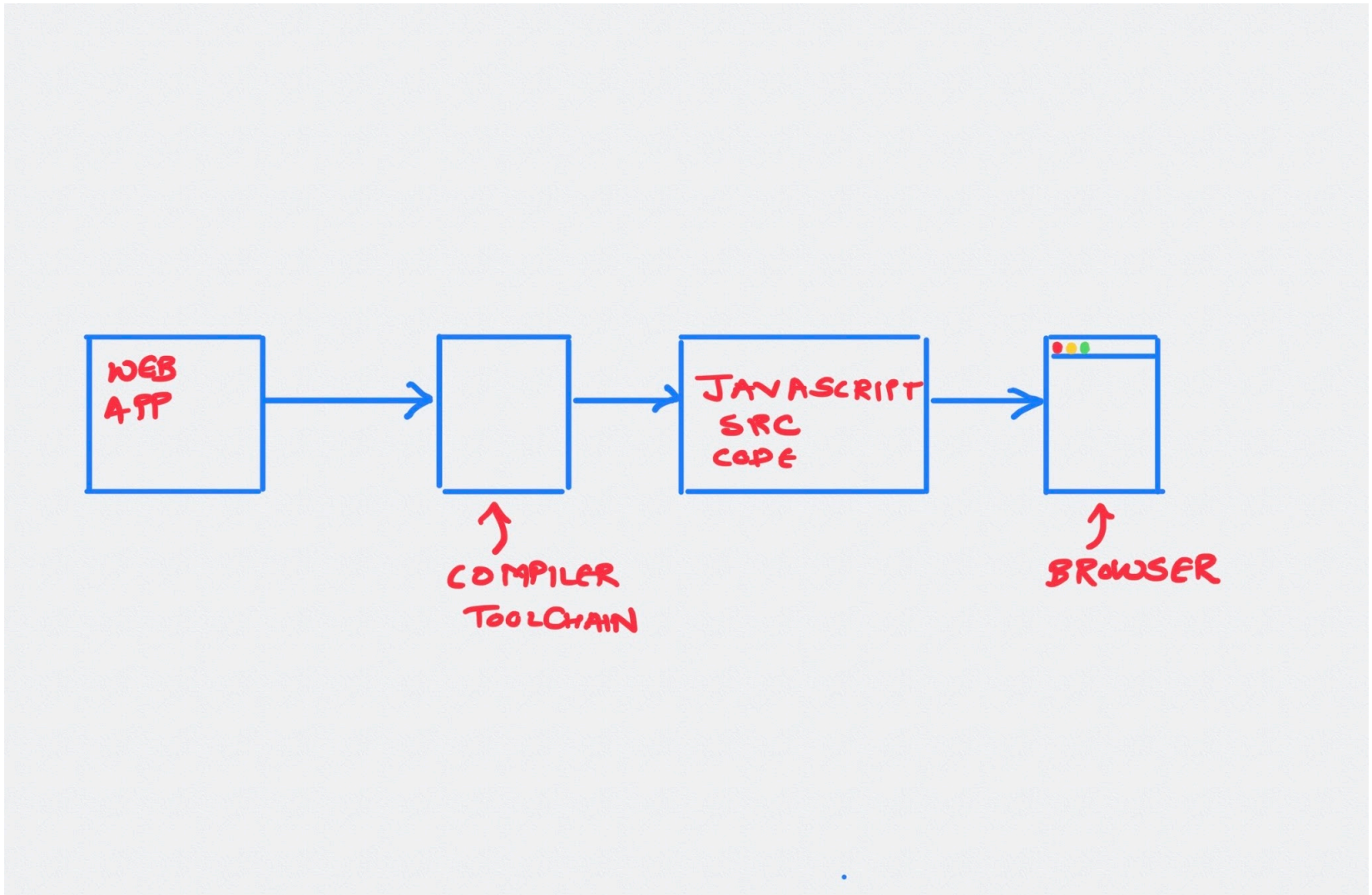

Patch code
Recompile
Test
Deploy

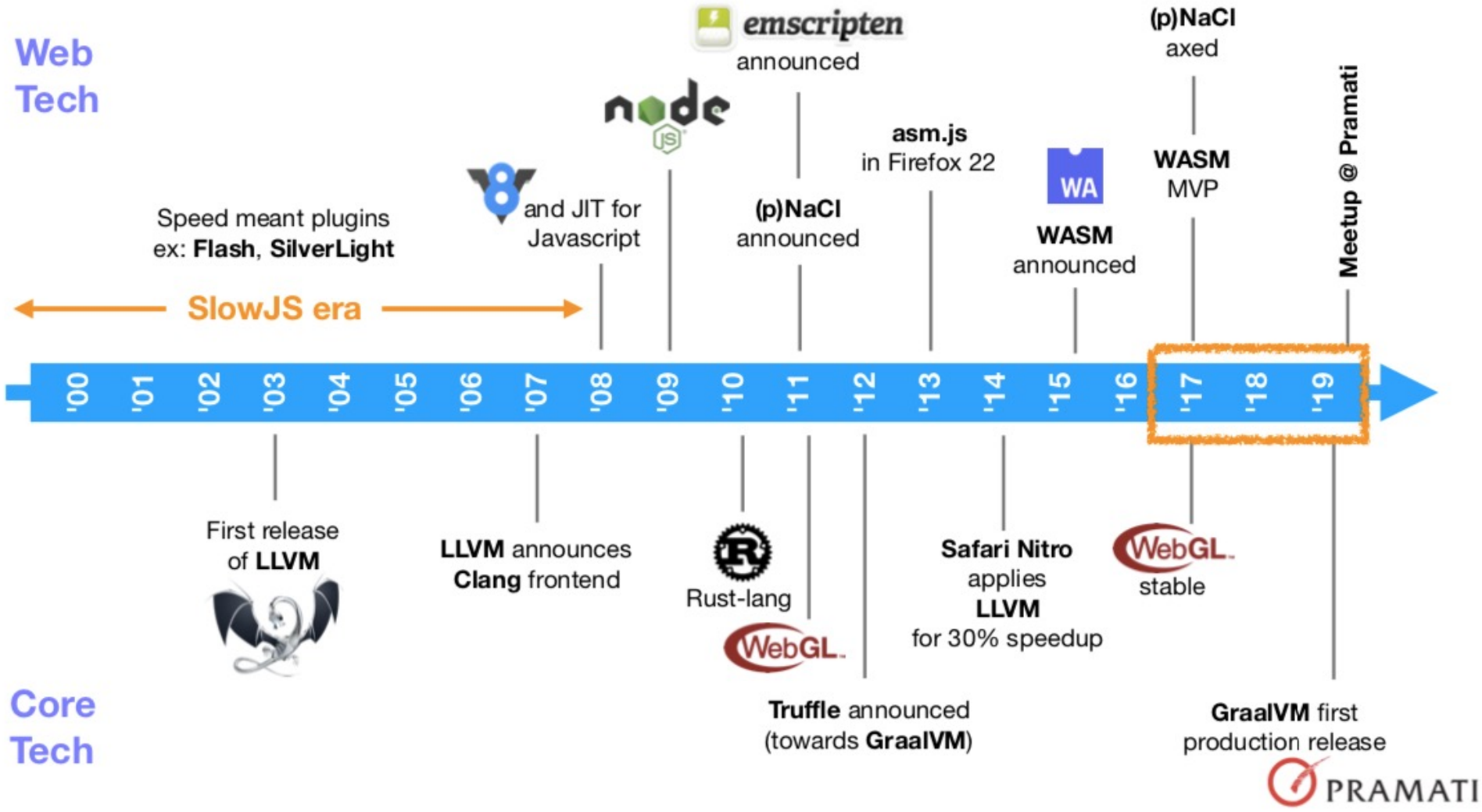
. . .


I JUST CAN'T KEEP UP
WITH ALL THIS ACTIVITY!
imgflip.com

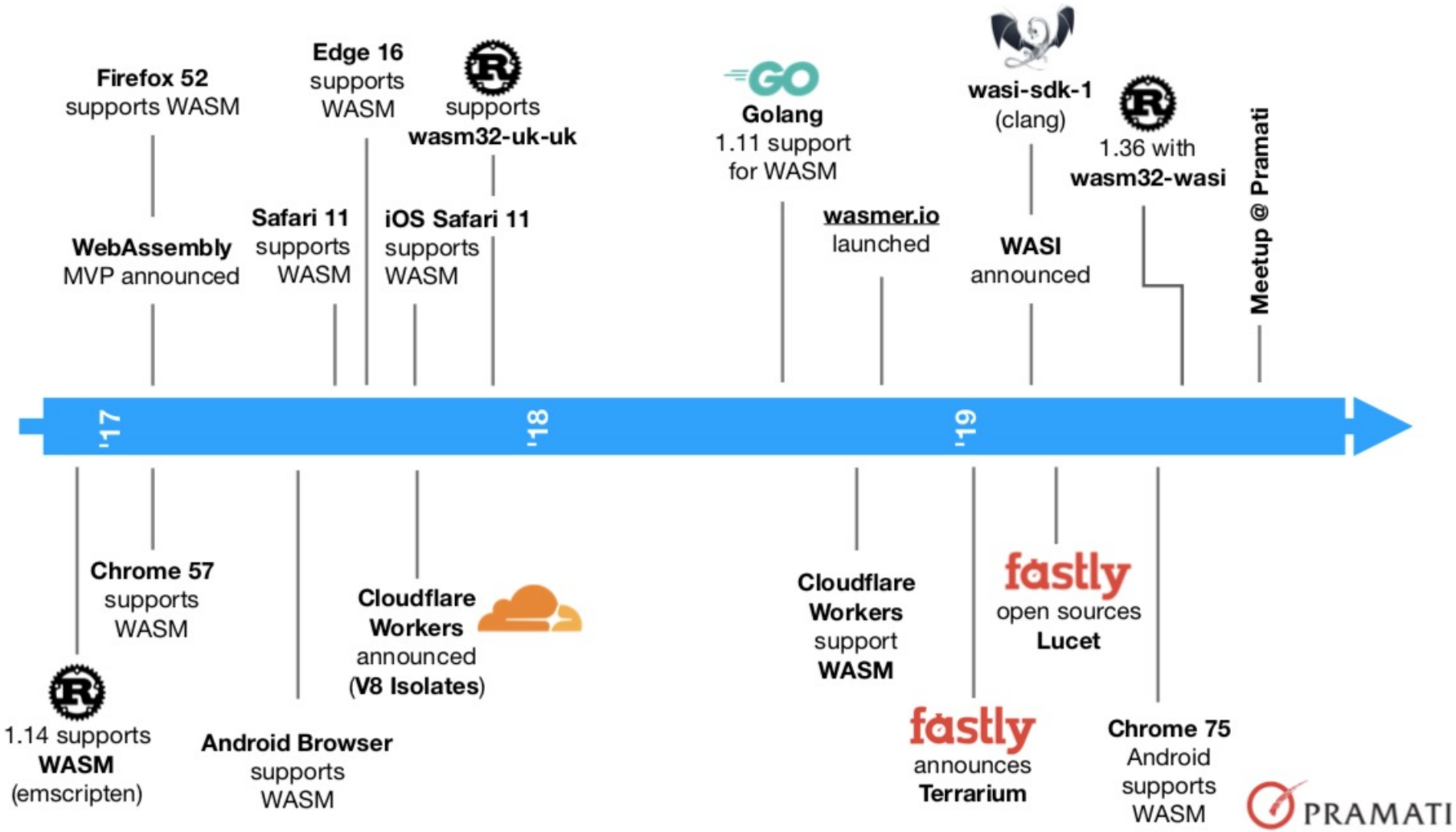

NOT ONLY
This is the way
imgflip.com

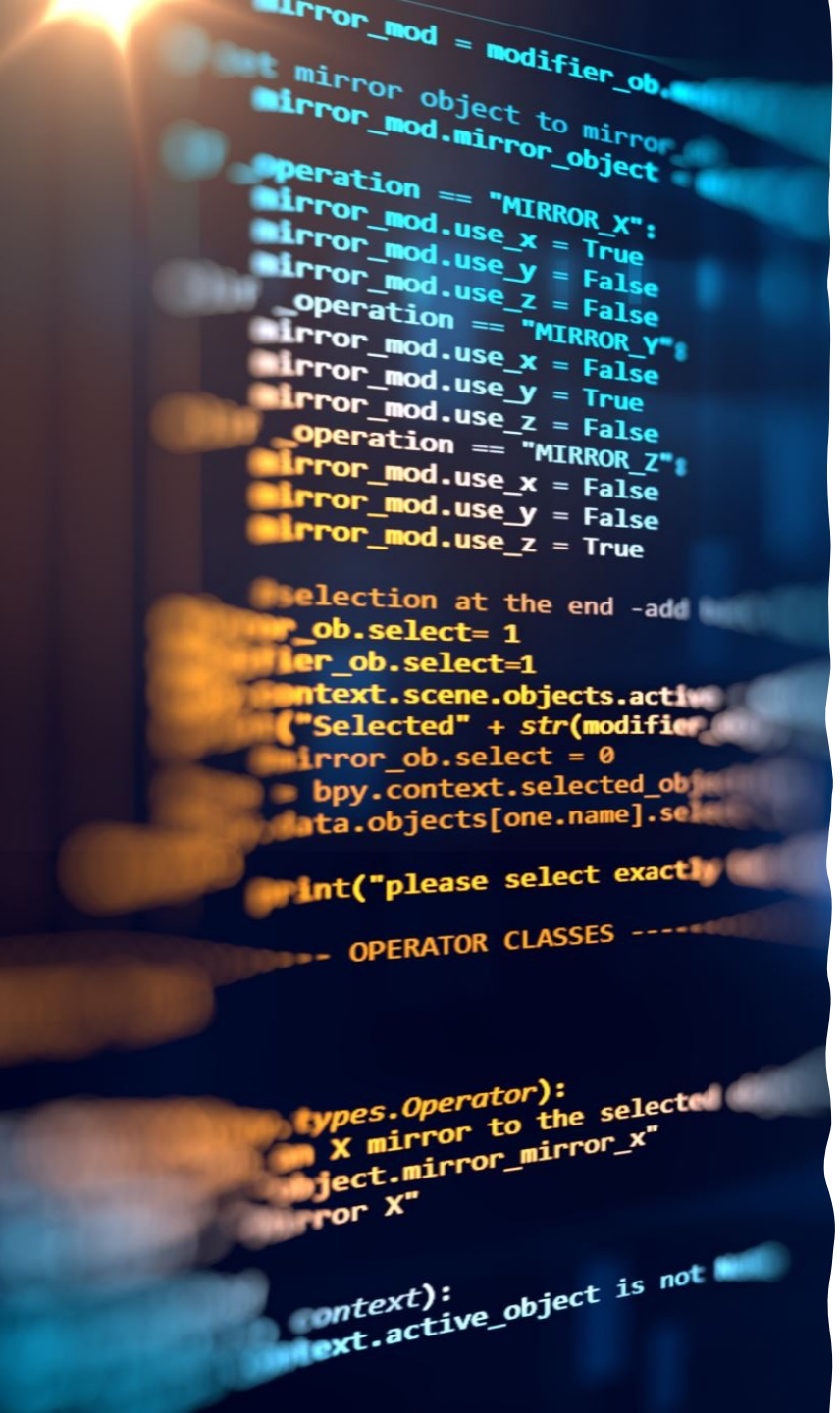

DO YOU HAVE TIME TO TALK ABOUT
WEBASSEMBLY
imgflip.com


WEB
APP
JAVASCRIPT
SRC
CODE
COMPILER
TOOLCHAIN
BROWSER



Source: http://sriku.org/blog/2019/08/24/talk-the-nuts-and-bolts-of-webassembly/


Firefox 52
supports WASM
WebAssembly
MVP announced
Edge 16
supports
WASM
Safari 11
supports
WASM
supports
wasm32-uk-uk
iOS Safari 11
supports
WASM
Golang
1.11 support
for WASM
wasmer.io
launched
wasi-sdk-1
(clang)
WASI
announced
1.36 with
wasm32-wasi
Meetup @ Pramati
'17
'18
'19
1.14 supports
WASM
(emscripten)
Chrome 57
supports
WASM
Android Browser
supports
WASM
Cloudflare
Workers
announced
(V8 Isolates)
Cloudflare
Workers
support
WASM
fastly
announces
Terrarium
fastly
open sources
Lucet
Chrome 75
Android
supports
WASM
PRAMATI



“WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.”

Source: https://webassembly.org


What does a Wasm program look like?

```
1 (module)
```

```
          00 01 02 03 04 05 06 07

00000000  00 61 73 6D 01 00 00 00
```

Wasm Modules

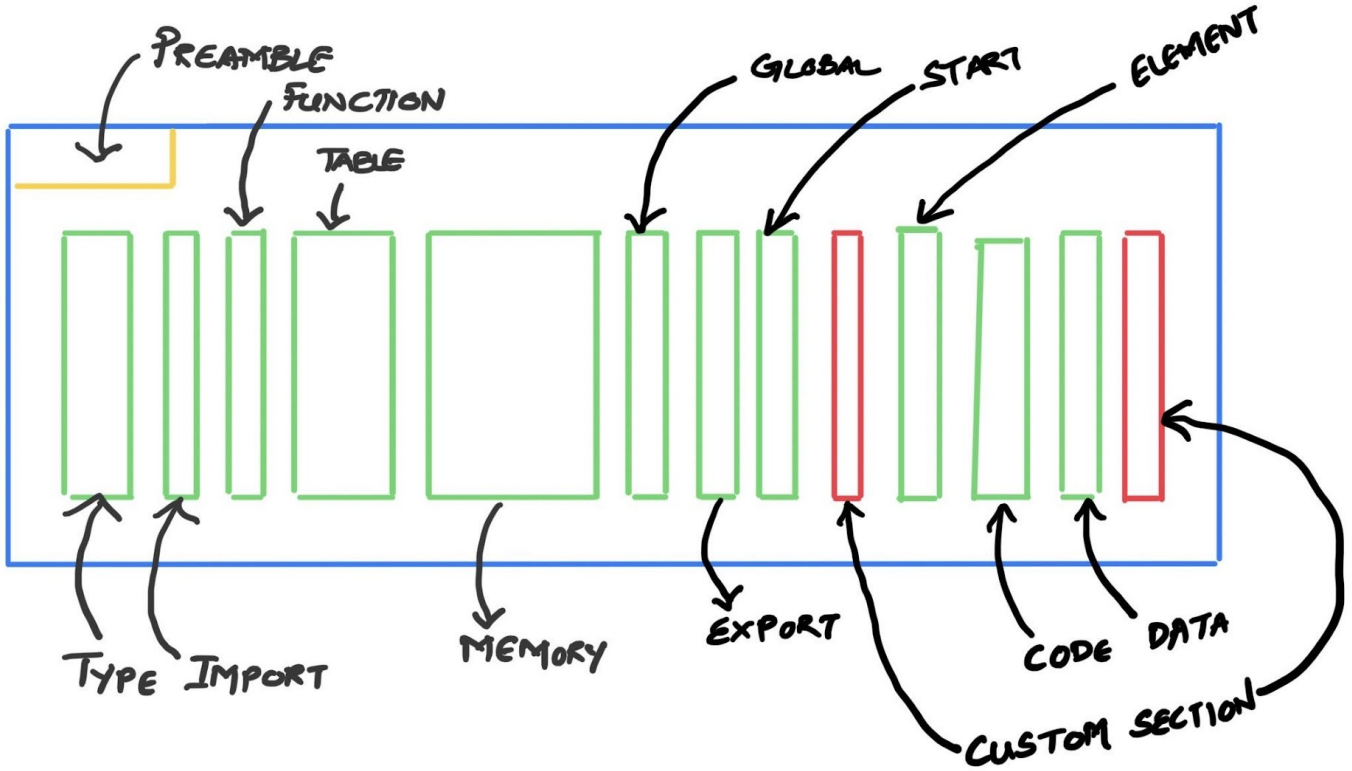
- Compilation target that many languages support
- deny-by-default but really no function to call by default

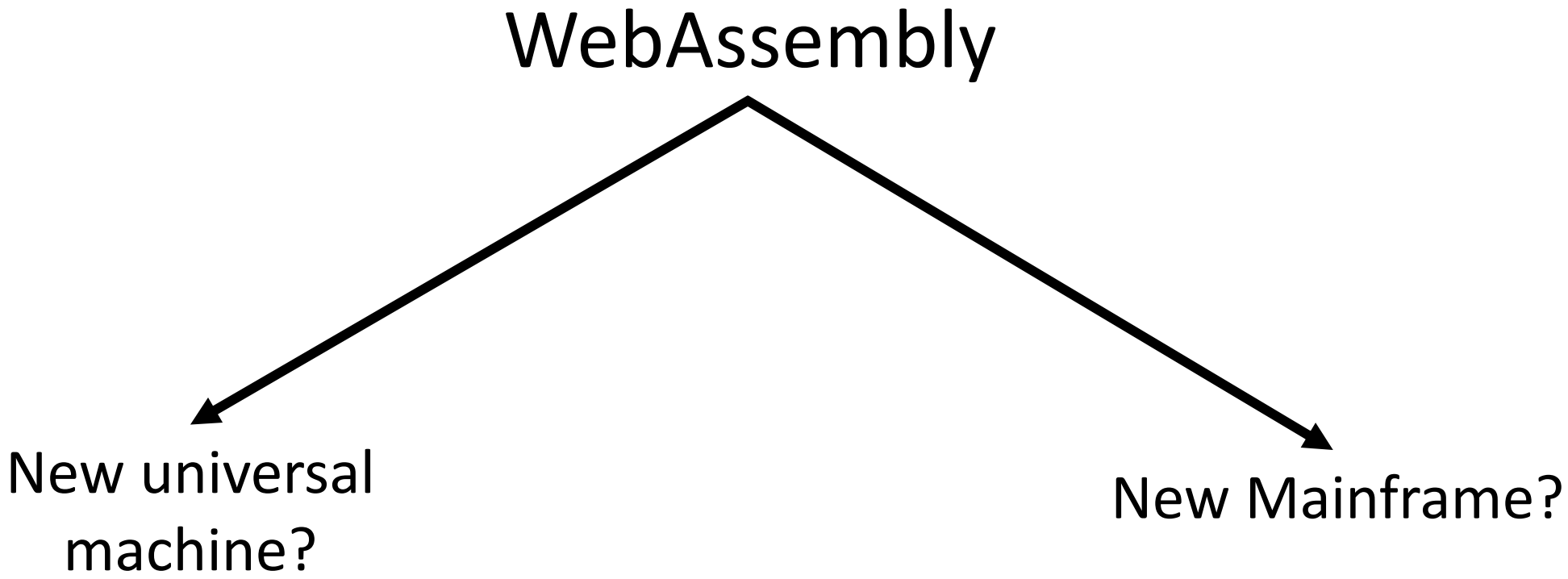
CLOUD NATIVE
Wasm DAY

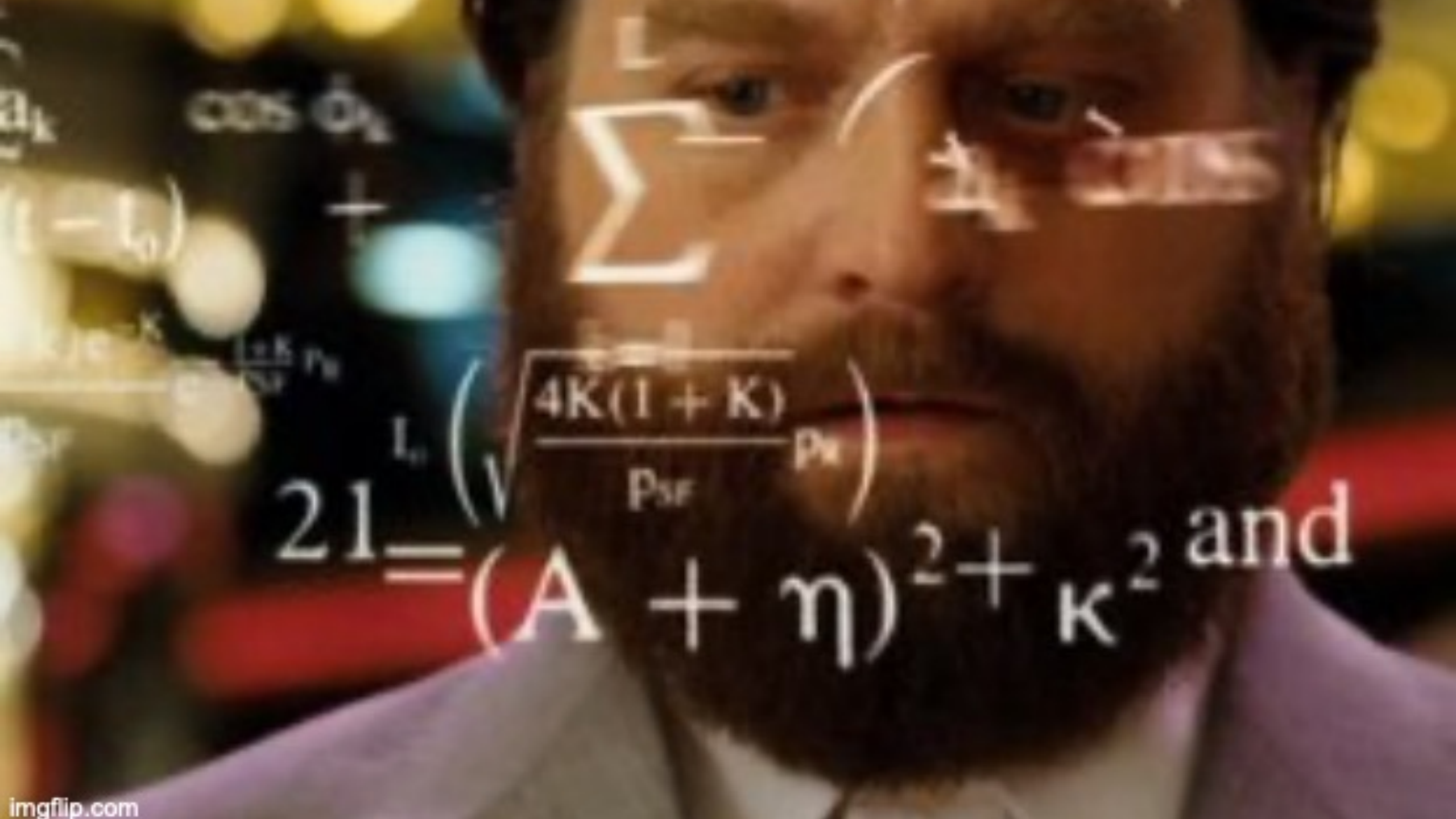
CLOUD NATIVE
Wasm DAY
EUROPE

Source: Evolution of Wasm: Past, Present, Future by Bailey Hayes
https://youtu.be/6_BRLqxiZPU

Anatomy of a Wasm module




WebAssembly
New universal machine?
New Mainframe?



Wasm Modules

- Compilation target that many languages support
- deny-by-default but really no function to call by default

CLOUD NATIVE
Wasm DAY

CLOUD NATIVE
Wasm DAY
EUROPE

Source: Evolution of Wasm: Past, Present, Future by Bailey Hayes
https://youtu.be/6_BRLqxiZPU

Specifications

- Core specification: defines the semantics of WebAssembly modules independent from a concrete embedding. The WebAssembly core is specified in a single document.
- Embedding interfaces:
  - JavaScript API: defines JavaScript classes and objects for accessing WebAssembly from within JavaScript, including methods for validation, compilation, instantiation, and classes for representing and manipulating imports and exports as JavaScript objects.
  - Web API: defines extensions to the JavaScript API made available specifically in web browsers, in particular, an interface for streaming compilation and instantiation from origin-bound `Response` types.
  - WASI API: defines a modular system interface to run WebAssembly outside the web, providing access to things like files, network connections, clocks, and random numbers.
- Tool conventions: repository describing non-standard conventions useful for coordinating interoperability between tools working with WebAssembly. This includes conventions for linking schemes, debugging information, language ABIs and more.
- Original design documents: documents describing the design, goals and high-level overview of WebAssembly. Some of these documents are outdated by now.

BYTECODE ALLIANCE

Security | Mission & Values | Announcements | Articles | Projects | Membership

BYTECODE ALLIANCE

About the Bytecode Alliance

The Bytecode Alliance is a nonprofit organization dedicated to creating secure new software foundations, building on standards such as WebAssembly and WebAssembly System Interface (WASI).

CNCF Cloud Native Interactive Landscape

CLOUD NATIVE COMPUTING FOUNDATION

The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2023-05-29T13:35:28.

You are viewing 27 cards with a total of 140,375 stars, market cap of $21.5T and funding of $53.4B.

Landscape | Card Mode | Members | Serverless | Wasm

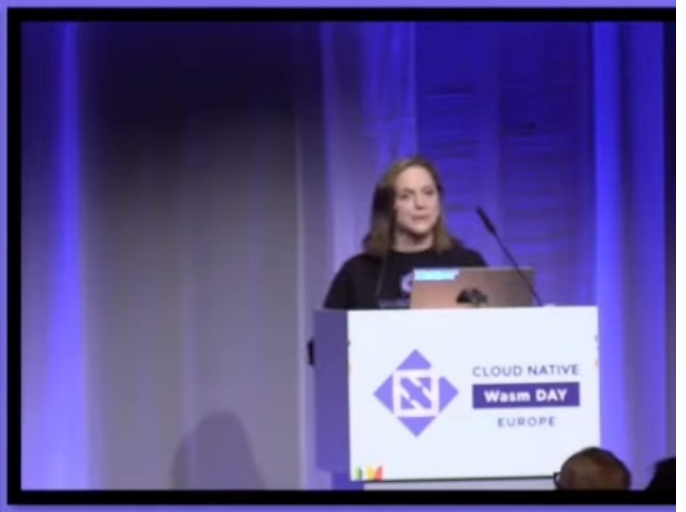
120%

The Component Model


SUGAR

What is a component?

CLOUD NATIVE
Wasm DAY



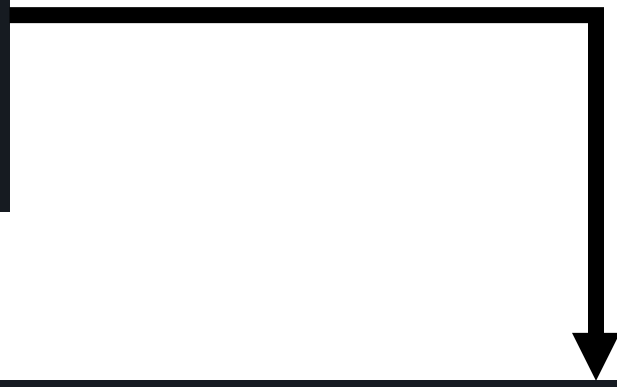
CLOUD NAT
Wasm DA
EUROPE

Hello 

- Describes imports & exports in WIT
- MAY represent the execution environment of a component.

Hello 🌏

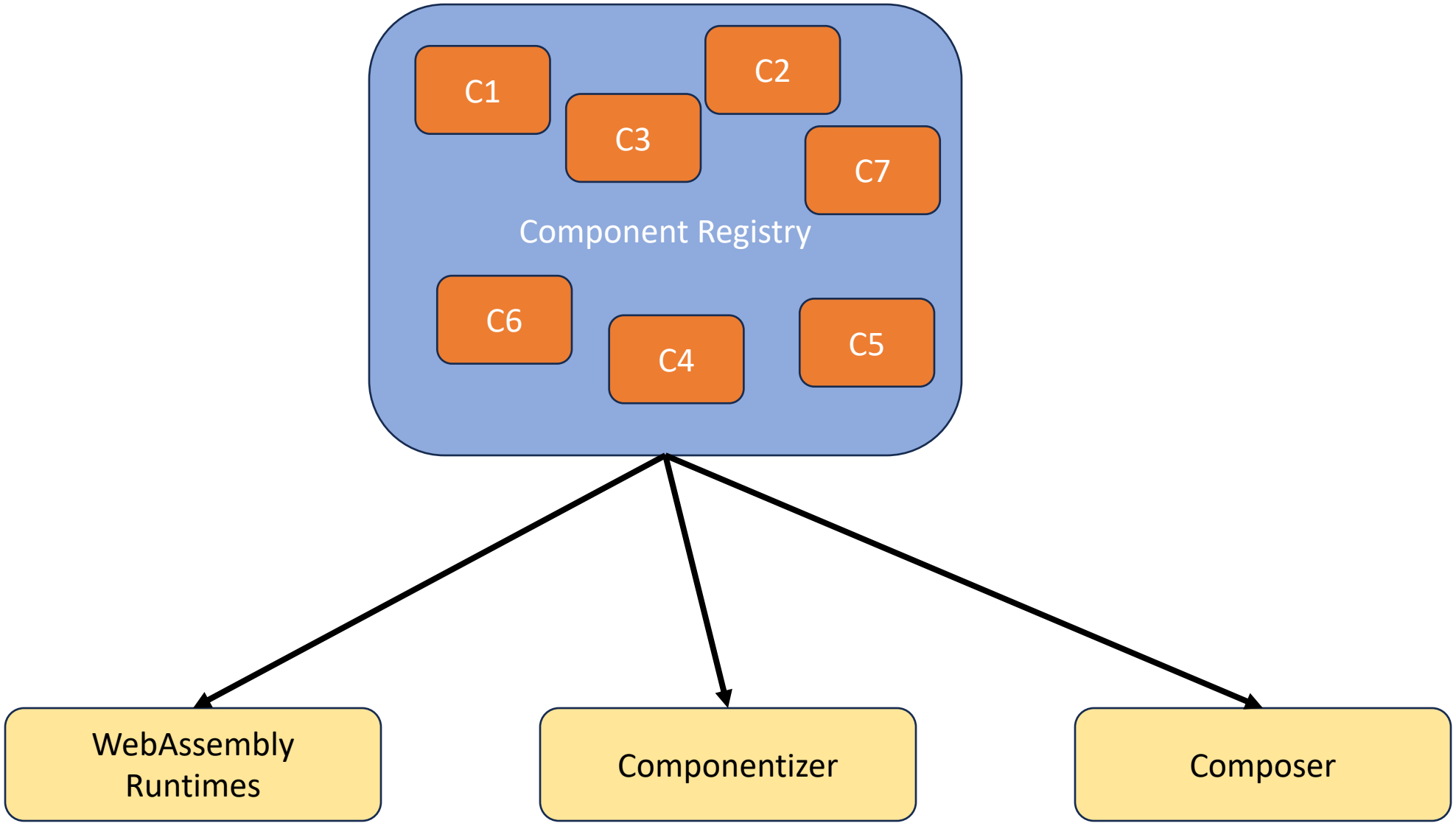
```
world my-world {
  import host: interface {
    log: func(param: string)
  }

  export run: func()
}
```



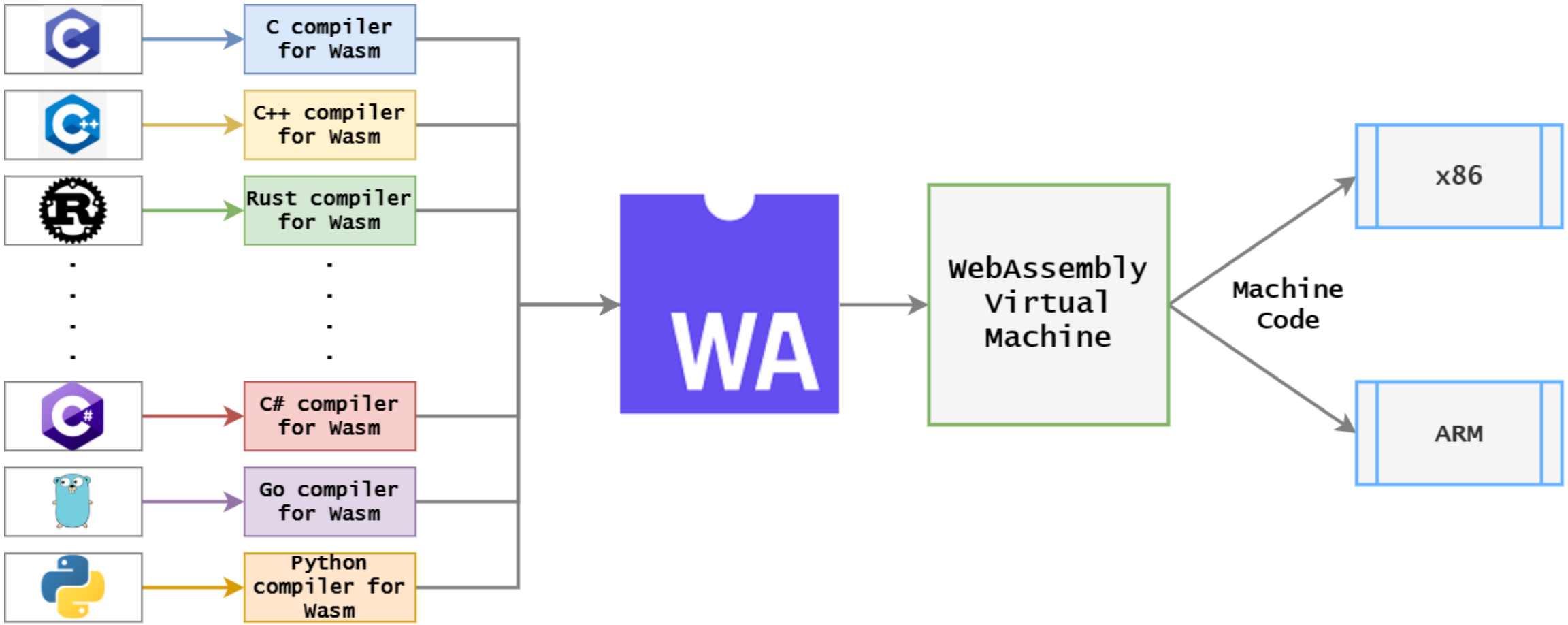
```
(type $my-world (component
  (import "host" (instance
    (export "log" (func (param "param" string)))
  ))
  (export "run" (func))
))
```

Component Registry

- Federated namespace of Wasm components
- Implements package transparency to enhance security
  - Cryptographic verifiable log of anything that has ever happened to a component


C1
C2
C3
C7
Component Registry
C6
C4
C5
WebAssembly Runtimes
Componentizer
Composer

Benefits?


C compiler for Wasm
C++ compiler for Wasm
Rust compiler for Wasm
C# compiler for Wasm
Go compiler for Wasm
Python compiler for Wasm
WA
WebAssembly Virtual Machine
Machine Code
x86
ARM

Demo time!

Resources

- Bytecode Alliance: https://bytecodealliance.org
- The Component Model:

  https://github.com/WebAssembly/component-model/tree/main