

WebAssembly

Saiyam Pathak

WebAssembly



Replace Javascript?

Replace Docker?

A new JAVA?

New programming language?

Going down the memory lane....

Mozilla, 2015

“WebAssembly... defines a portable, size- and load-time-efficient format and execution model specifically designed to serve as a compilation target for the web.”

Luke Wagner

- Java Applets
- VBScript
- Adobe Flash

JOINT EFFORT TO MAKE IT A SUCCESS

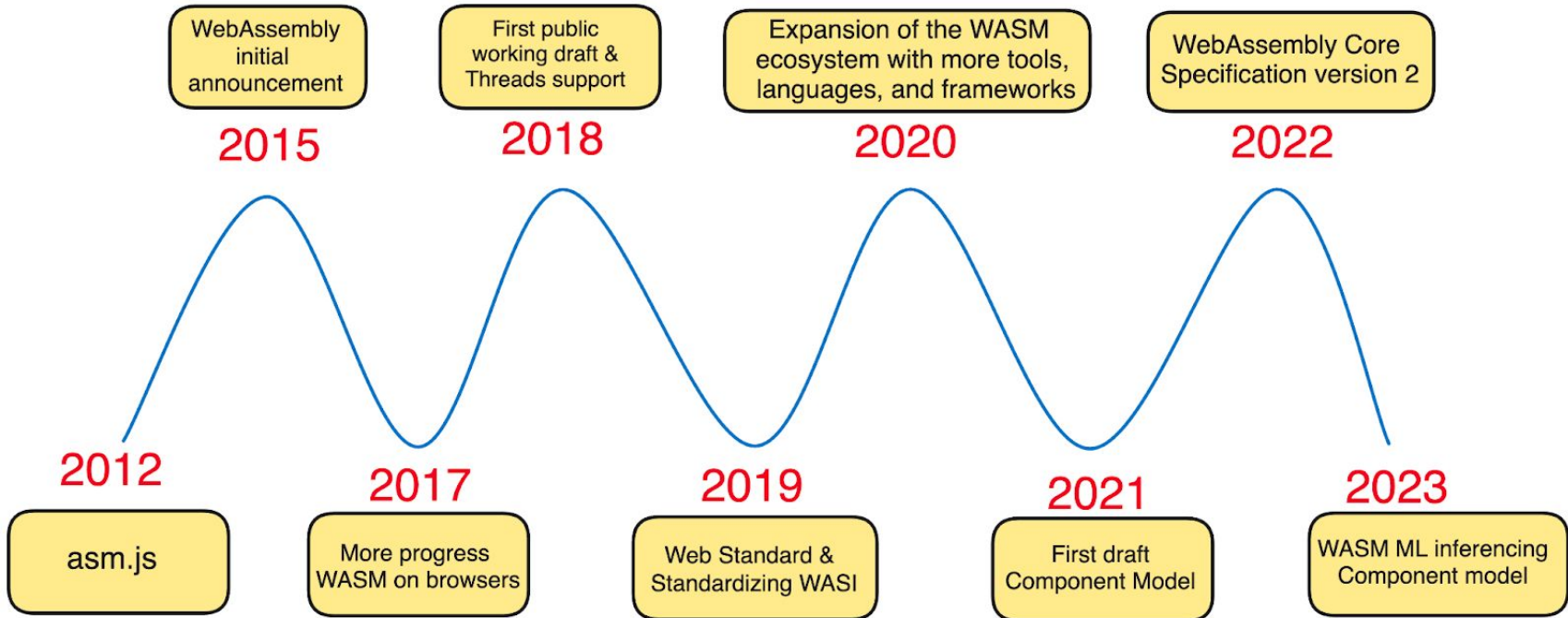
W3C

INSTEAD OF LANGUAGE FOCUS ON NEW BINARY

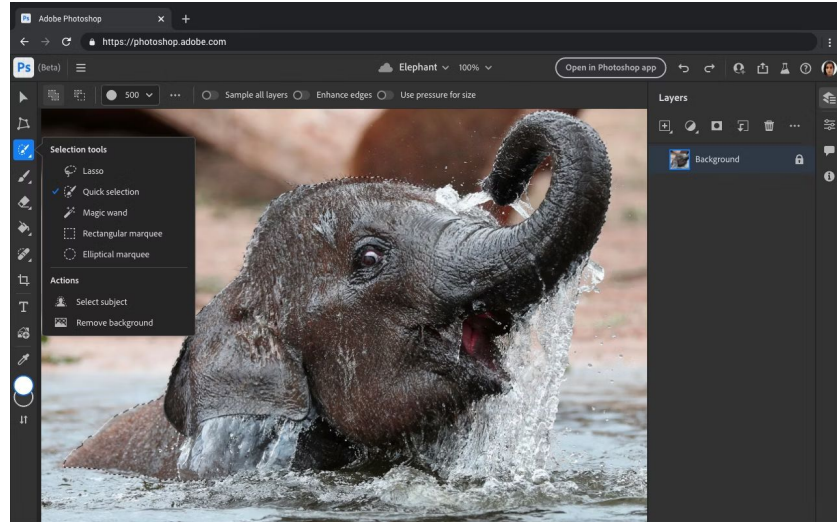
FORMAT

WASM vs JVM

Native languages languages
Strict sandboxing
w3c standard



Webassembly - Success in browser?



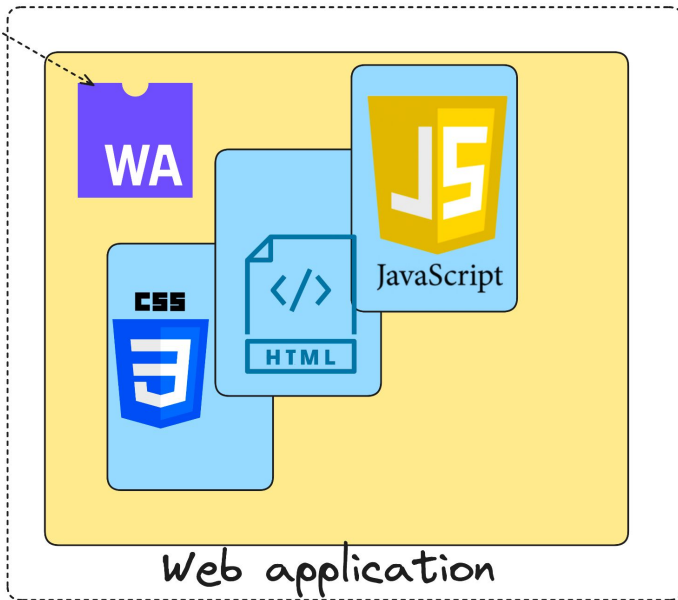
Emscripten

Webassembly

Write once run anywhere

.wasm

C, c++
Rust
Go



Secure sandboxing
Near native speeds
Polyglot
Small Binaries

Webassembly - Only for Browsers?

Isolation and security

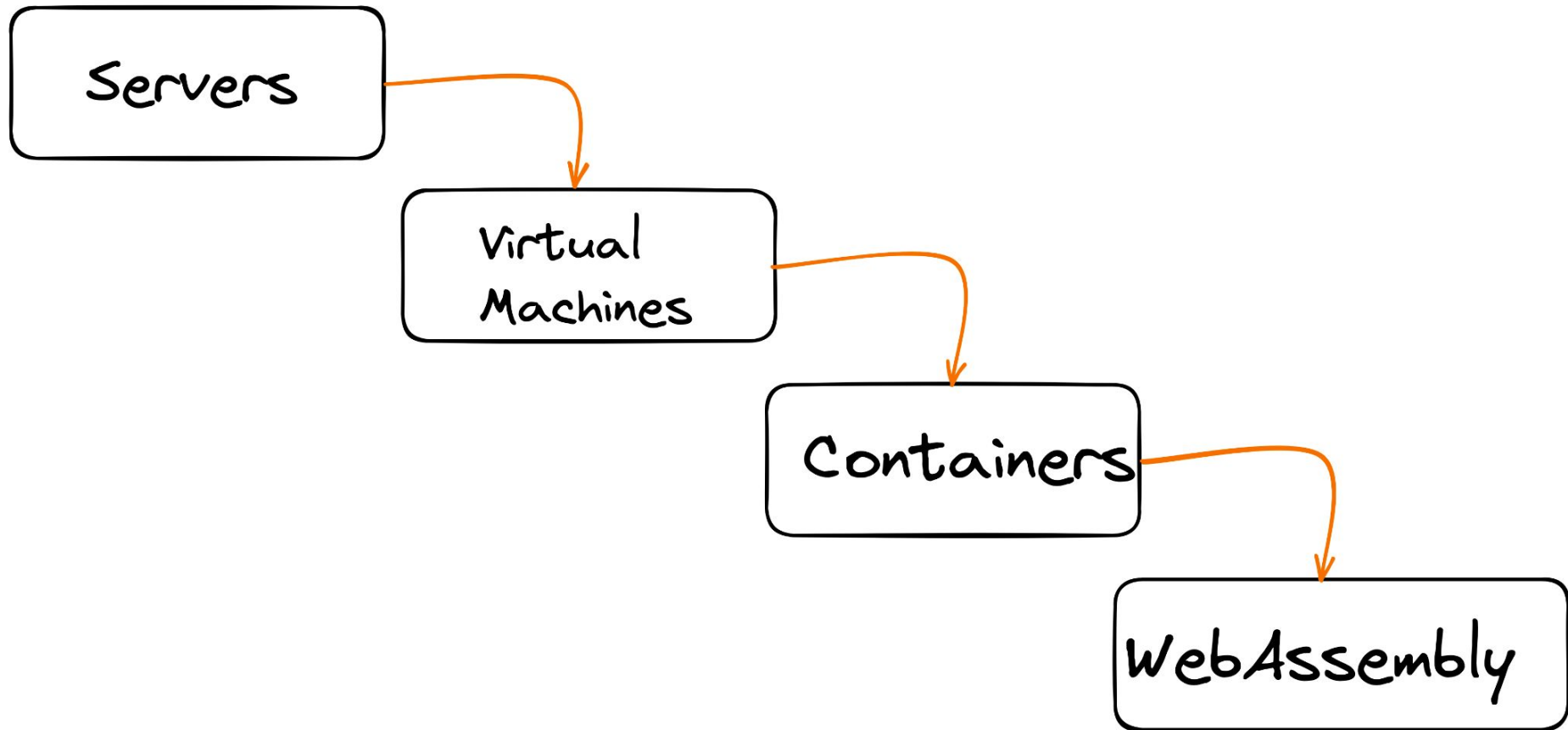
Multi platform

Small Binaries

Instant startup times

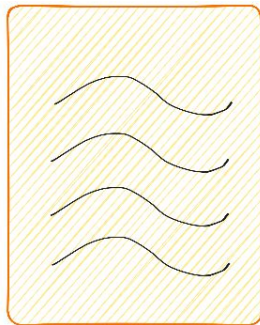


Evolution waves



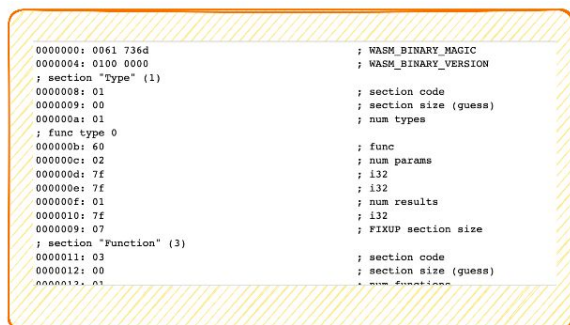


Code



Rust (hello)

cargo build --target wasm32-wasi



hello.wasm

```
{module
  (func (export "addTwo") (param i32 i32) (result i32)
    local.get 0
    local.get 1
    i32.add))
```

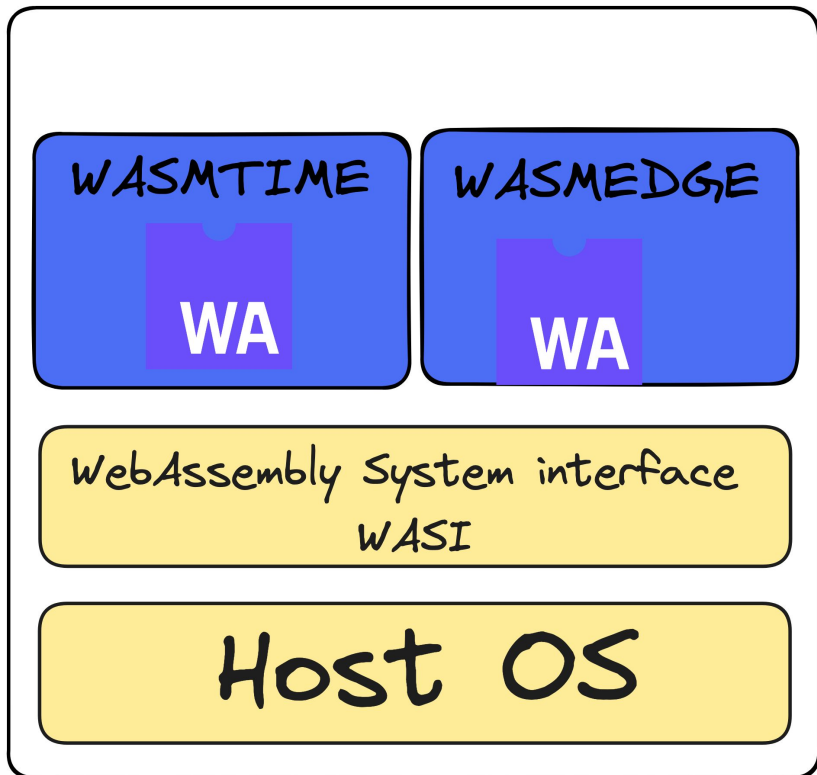
```
const wasmInstance =
  new WebAssembly.Instance(wasmModule, {});
const { addTwo } = wasmInstance.exports;
for (let i = 0; i < 10; i++) {
  console.log(addTwo(i, i));
}
```

x64

x86

ARM

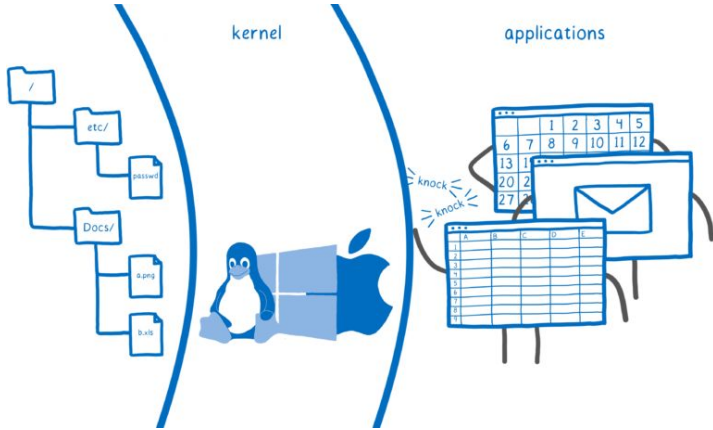
What about system calls? Wasm runtimes



Files, directories, environment variables?

WASI and Component model

WASM module does not compile to specific OS



Source-<https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/>

WASI - WebAssembly System interface

