Go Meets Wasm: Harnessing the Power of Wasm Components with Go

Jiaxiao Zhou, Microsoft

周佳孝 (Mossaka)

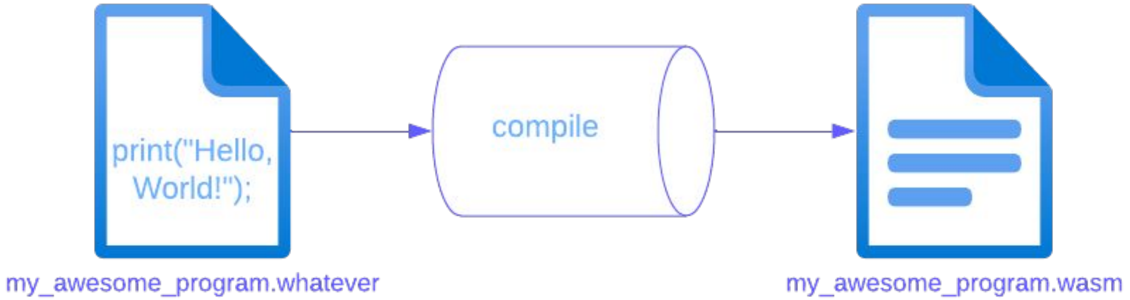
- Building open-source software for developers at Microsoft’s DeisLabs
- Programming language enthusiastic
- Organizer of SPLVM meetup

I am on X as @jiaxiao_zhou

What is Wasm?

A bytecode format for a compilation target

What is Wasm?


print("Hello, World!");
my_awesome_program.whatever
compile
my_awesome_program.wasm

“Write Once, Run Everywhere”

[Write once, run anywhere - Wikipedia]

Wasm is Awesome

- **Language agnostic**
- **Sandboxing**
- **Memory management**
- **Zero capability**

Suitable for isolating code in **clouds**, **edge networks**, embedded devices, browsers, and network proxies.

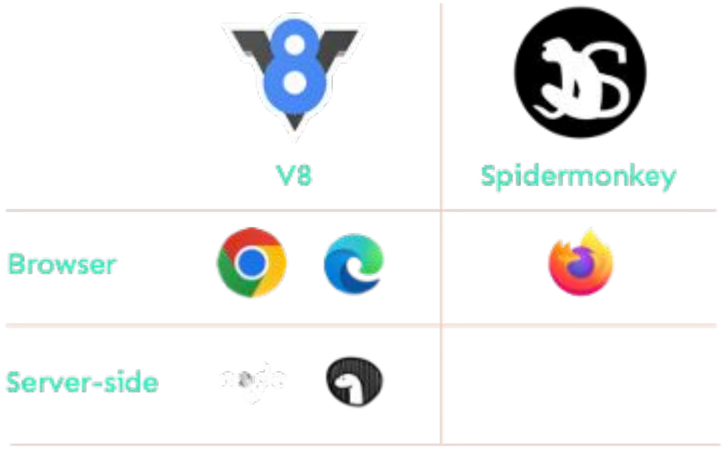
WASI

WASI

- 2019: Bytecode Alliance was founded to standardize WASI
- 2019: WASI Preview 1
  - Posix APIs
  - No network support
  - No fork / exec
- 2023: WASI Preview 2

Runtimes

JavaScript Runtimes


V8
Spidermonkey
Browser
Server-side

WASI runtimes

Wasmtime

Wasmer

WasmEdge

Go 1.21 + WASI Preview 1

- Build program with `GOOS=wasip1 GOARCH=wasm`
- `go:wasmimport` compiler directive
- The runtime binary was embedded into the compiled Wasm bytecode
- Limitations
  - Any host function call will block all goroutines due to single threaded architecture of Wasm
  - No network sockets API in wasip1
- wasi-go project
  - provides all the socket capabilities which is ABI-compatible with WasmEdge runtime

Demo

Motivation

- Not easy to **compose** with other Wasm modules
- Not easy to **share memory** with other Wasm modules

Wasm Component Model

1. ABI and IDL (WIT) for composing Wasm modules
2. Higher level types: String, Record, etc.
3. The “World”: grouping of imports and exports
    a. Host
    b. Guest
4. `wasi:http/proxy` world

WASI Preview 2

- The next major iteration of WASI
- Complete rebase to Wasm Component Model
- Wasi-cli
- Wasi-http
- Wasi-cloud-core

Go + WASI Preview 2

- Support compile to Wasm component directly ❌
- Support WASI Preview 2 APIs ❌

Missing Features

1. Wasm component model canonical ABI defines a `realloc` function

   Go memory management in the WebAssembly Component Model

2. `go:wasmexport`

   proposal: cmd/compile: go:wasmexport directive · Issue #42372 · golang/go · GitHub

But, there is hope…

- TinyGo
- \`Wasm-tools\`
- \`Wit-bindgen-go\`
- \`WASI preview 1 adapter\`

Demo

The Future

- TinyGo being the experimental ground for WASI Preview 2 APIs
- Go Wasm32 proposal
- Go `go:wasmexport`
- Go `realloc` implementation
- WIT bindings checking in to Go upstream
- Go emits Wasm components directly from runtime

Getting Involved!

Join Bytecode Alliance SIG-Guest-Language Go Subgroup:

Agenda: BA SIG-Guest-Languages Go Subgroup - Agenda

When: every other Tuesday at 10am PST (1pm EST)

Reference

- WebAssembly for the Java Geek - JVM Advent
- Whence-Wasm
- WASI: a New Kind of System Interface
- No Ghosts! · sunfishcode's blog
- What is a World? · sunfishcode's blog
- A fork() in the road
- GitHub - WebAssembly/wasi-cli: Command-Line Interface (CLI) World for WASI
- proxy.md - WebAssembly/wasi-http · GitHub
- WebAssembly/wasi-cloud-core