FERMYON

Build a Serverless WebAssembly App in Python using Spin

Sohan Maheshwar

Lead Developer Advocate @ Fermyon

The next wave of cloud compute will be powered by WebAssembly

FERMYON


What is it?
How and when to use it?
How will this change everything?

FERMYON

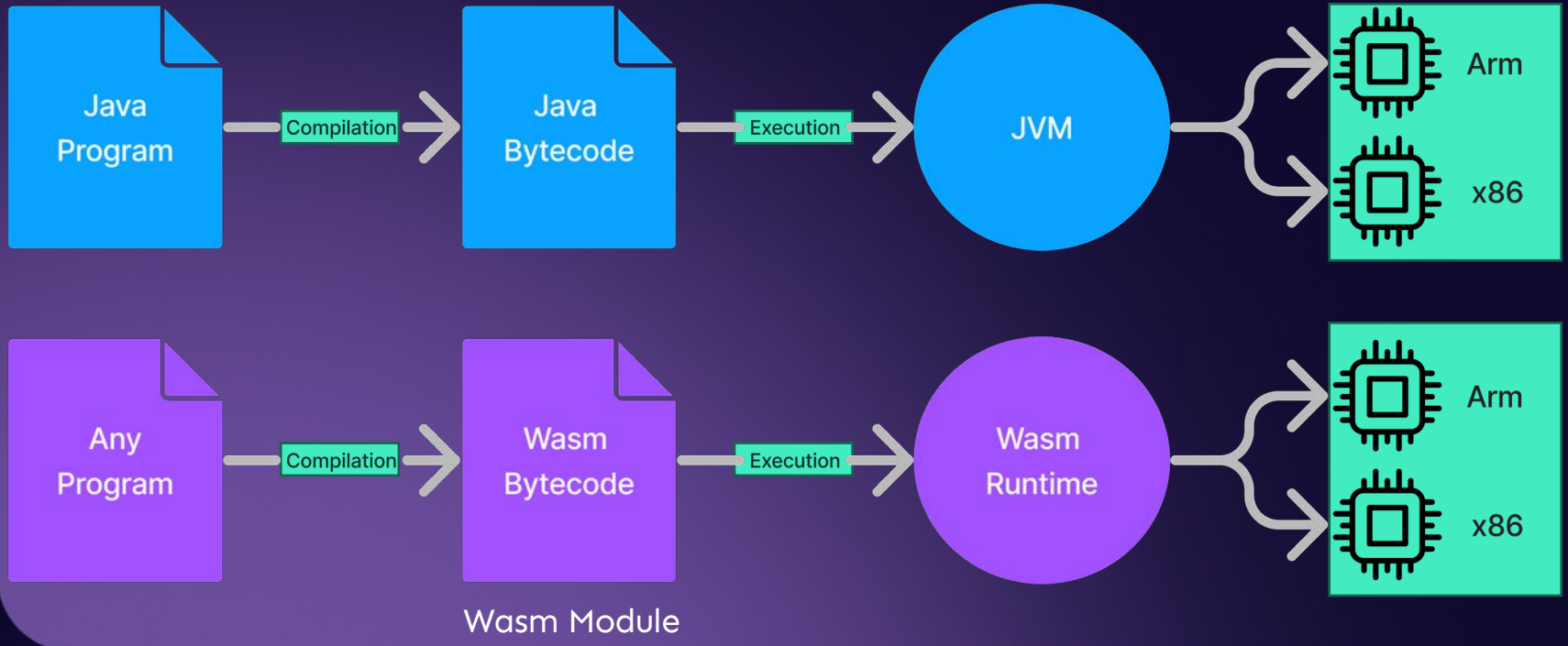
What Is WebAssembly?

The boring answer: It's just another bytecode format

A few things to know about WebAssembly

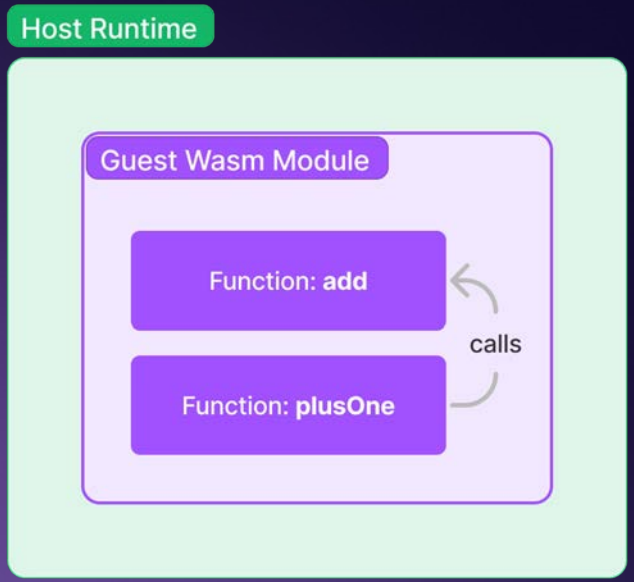
- Wasm is just another name for it
- Designed as a portable compilation target

Wasm is another bytecode format

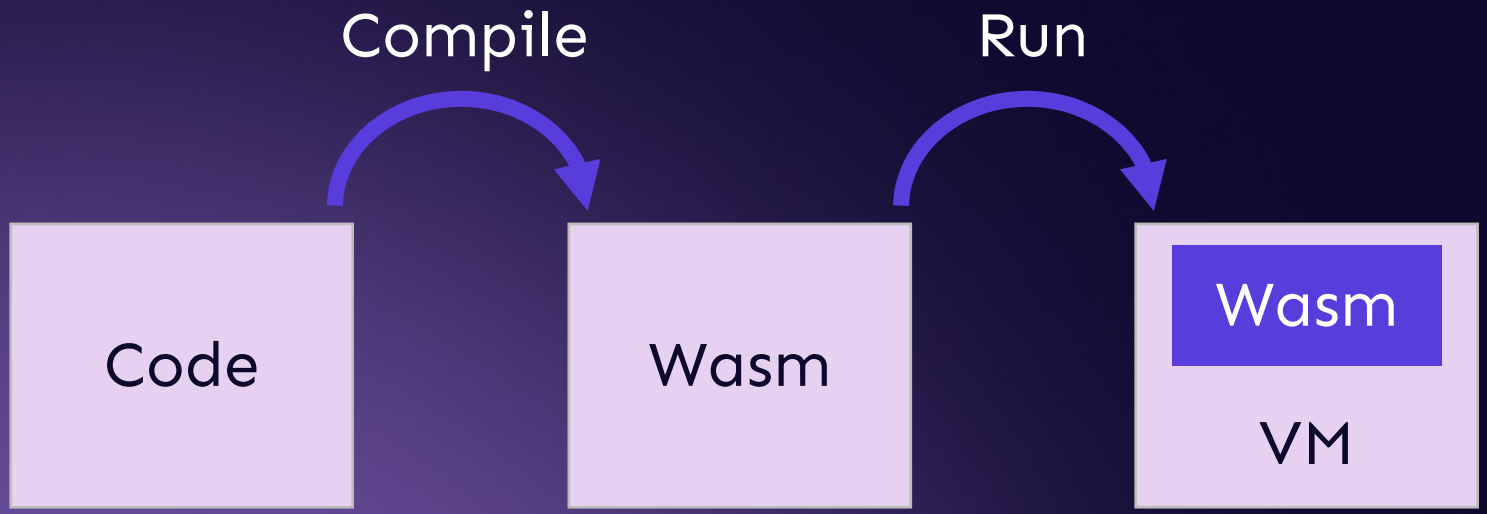


A few things to know about WebAssembly

- Wasm is just another name for it
- Designed as a portable compilation target
- Originates from the browser, now also available outside
- "compile once" and then run that code on any number of targets


Host Runtime
Guest Wasm Module
Function: add
Function: plusOne
calls

Compile and Run



WASI: A new kind of System Interface

TL;DR: it allows you to run WebAssembly outside of the browser

- Access to several operating-system-like features, including files and filesystems, clocks, and random numbers
- Independent of browsers, so it doesn't depend on Web APIs or JS
- It extends Wasm's sandboxing to include I/O.

Compilation and Language Support

WebAssembly Support in Top 20 Languages

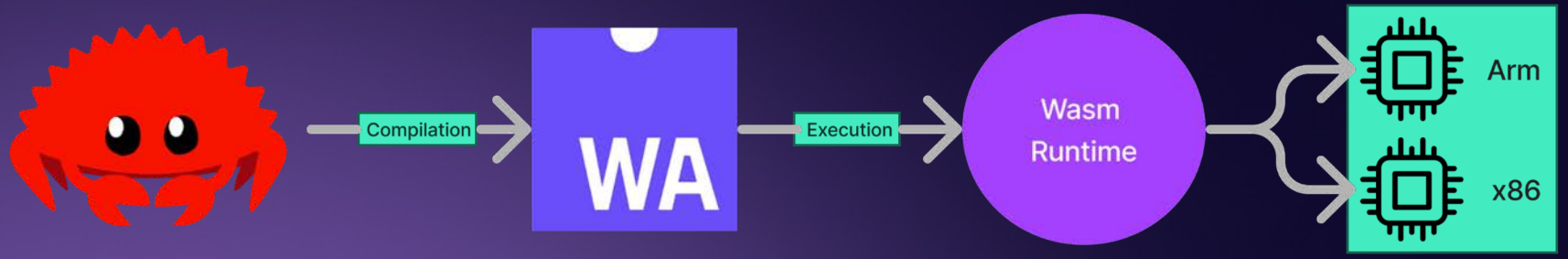
This reports on the top 20 languages from RedMonk's ranking. Some languages, like CSS, PowerShell, and "Shell", don't really have a meaningful expression in Wasm. However, we have left them here for completeness.

Language	Core	Browser	WASI	Spin SDK
JavaScript	✅	✅	⏳	✅
Python	✅	⏳	✅	✅
Java	✅	✅	✅	⏳
PHP	✅	✅	✅	❌
CSS	N/A	N/A	N/A	N/A
C# and .NET	✅	✅	✅	✅
C++	✅	✅	✅	❌
TypeScript	✅	⏳	❌	✅
Ruby	✅	✅	✅	❌

https://www.fermyon.com/wasm-languages/webassembly-language-support/

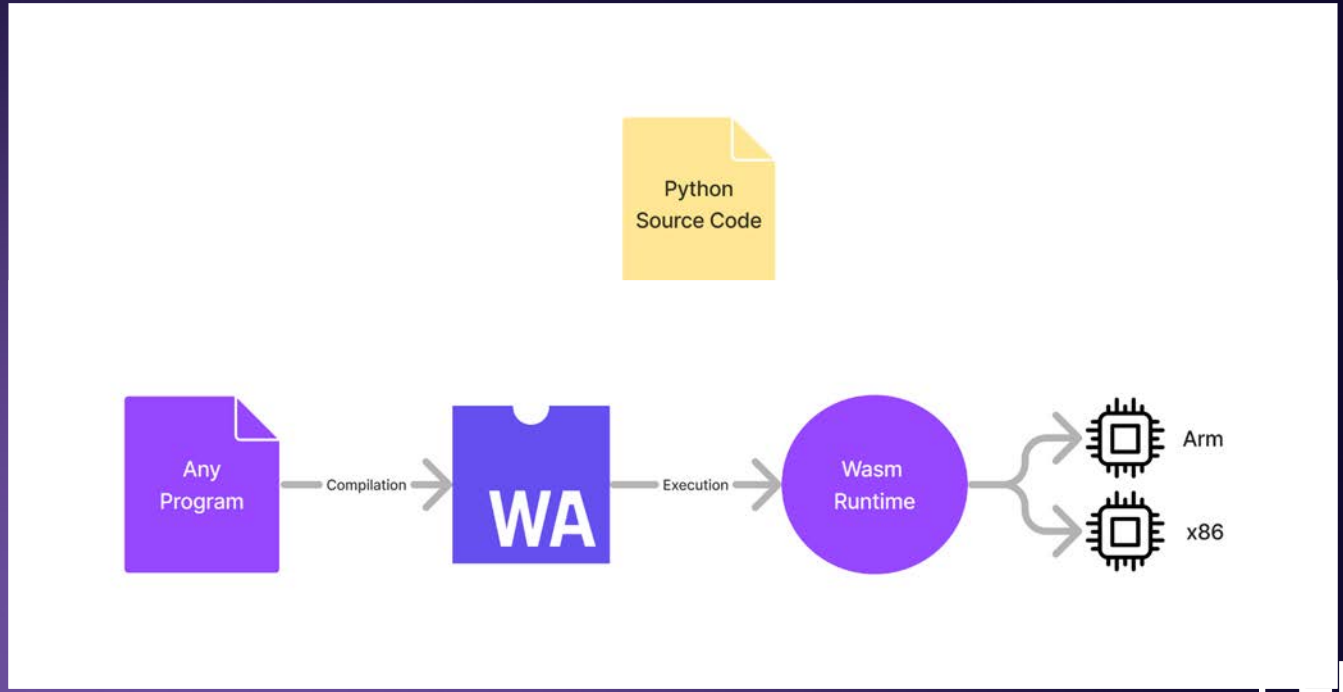
How do I compile my code to Wasm?

Rust has great Wasm support



Interpreted languages are a little more tricky

You need to compile the interpreter to Wasm



FERMYON

Getting started with Wasm

FERMYON

Introducing Spin

- The open-source tool for building WebAssembly serverless apps
- Create a new serverless with just a few commands.

https://developer.fermyon.com/spin

FERMYON

The framework to compose serverless WebAssembly apps.

OPEN SOURCE

15+ LANGUAGES

4.6K GITHUB ★

SIMPLE CLI

github.com/fermyon/spin

SPIN

Spin is an open source project, built with open standards like WASI, Wagi and the WebAssembly Component Model.

AT A GLANCE:

Serverless AI NEW

Quickly test and run inferencing workloads with LLMs.

Key/Value Store

Easily persist data in your apps with a built-in KV store.

Powerful CLI

Easy to create, run and deploy projects - in as little as 66 seconds.

NoOps SQL Database

Add SQLite data to your app with an always-available SQLlite DB.

COMPOSING APPS:

- HTTP & Redis Triggers
- Relational Database Support
- Variables & Secrets Rotation

DEV EXPERIENCE:

- Supports almost any programming language
- Easy to debug with included helper commands

FERMYON

DEMO

developer.fermyon.com/spin

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

*https://00f.net/2023/01/04/webassembly-benchmark-2023/

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

Startup Time

Startup times comparable with natively compiled code

Only 2.3x slower than native*

*https://00f.net/2023/01/04/webassembly-benchmark-2023/

FERMYON

4 things making WebAssembly great

Binary Size

Rust hello-world ~2MB

AOT compiled ~300KB

Basic Spin http api
~2.3MB JIT
~1.1MB AOT

Startup Time

Startup times comparable with natively compiled code

Only 2.3x slower than native*

Portability

Build once, run anywhere!

Same build (JIT) works across OS and platform arc

Security

Sandboxed execution

Capability based security model

*https://00f.net/2023/01/04/webassembly-benchmark-2023/

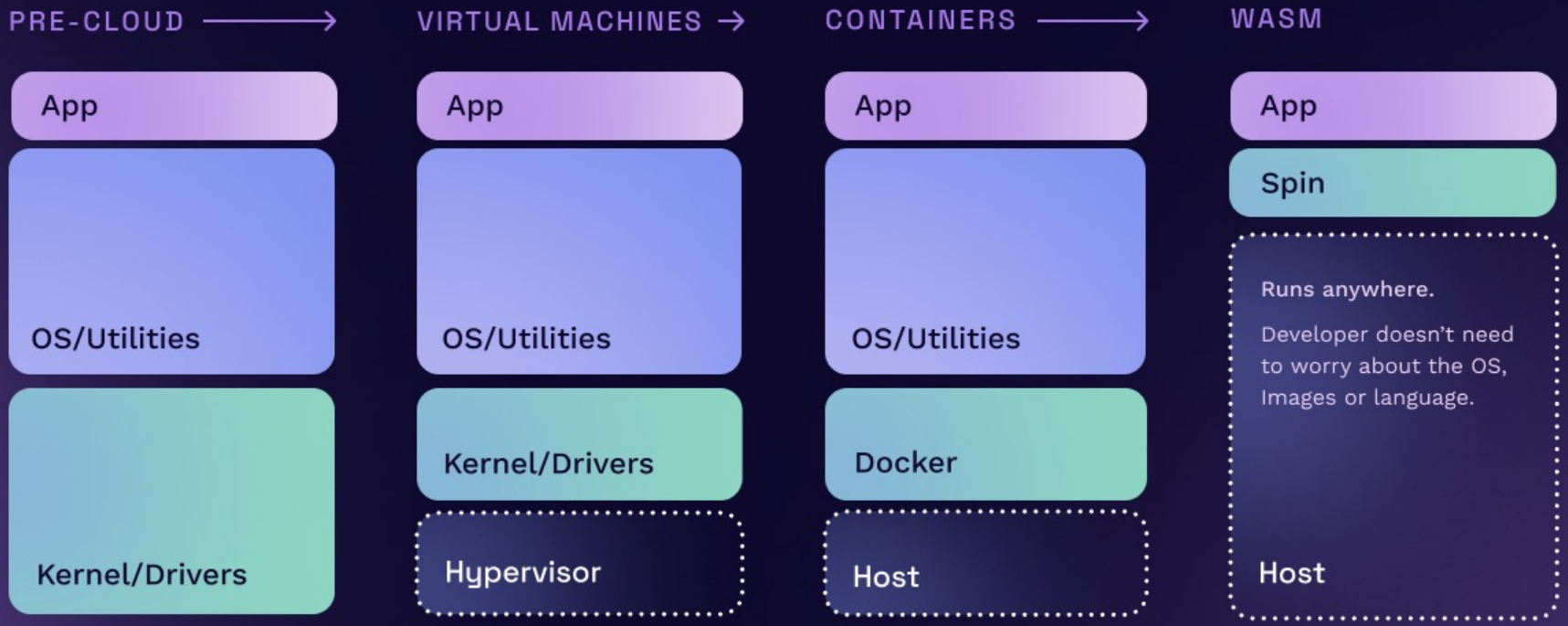
FERMYON

FERMYON

How this will change cloud computing

Gradually, then suddenly

How cloud computing has evolved:



FERMYON

No more clunky containers

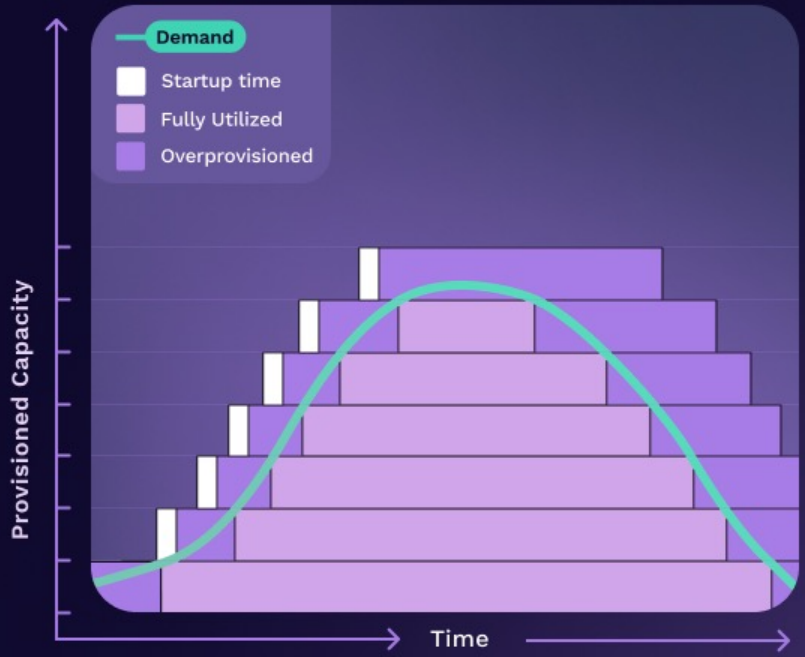
FERMYON

PAIN POINT

Containers are too expensive, over-consuming resources.

[A]cross 50 of the top public software companies currently utilizing cloud infrastructure, an estimated **$100B of market value is being lost** among them due to cloud impact on margins [...]

Source: https://a16z.com/the-cost-of-cloud-a-trillion-dollar-paradox/



FERMYON

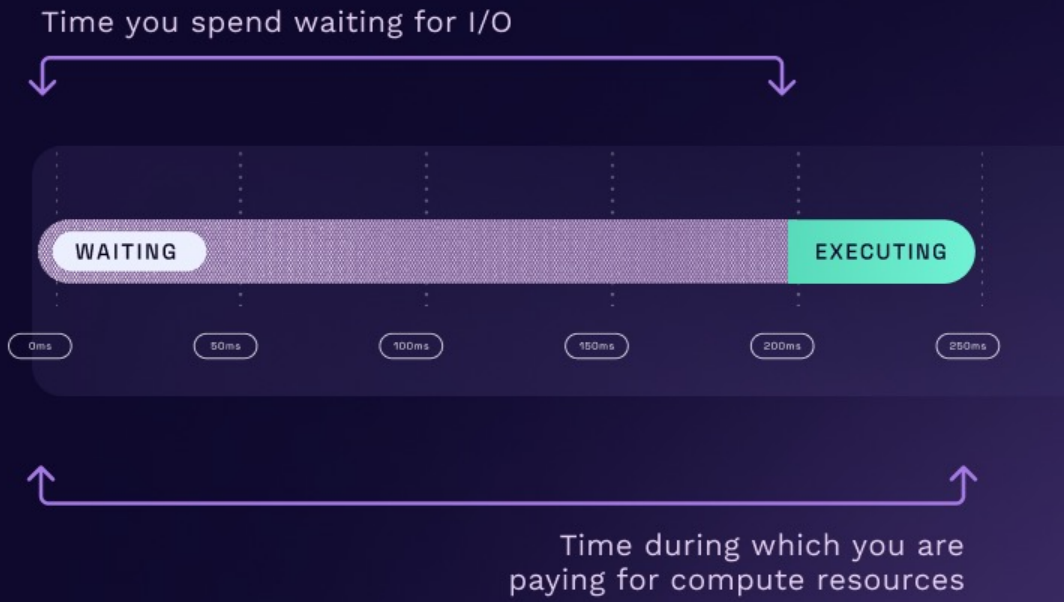
Fix serverless

FERMYON

PAIN POINT

Serverless has a Cold Start problem.

Solutions like AWS Lambda frequently take 2-3 seconds to wake up and start executing. Often the startup delay is orders of magnitude longer than the execution time itself, requiring complex and expensive workarounds.

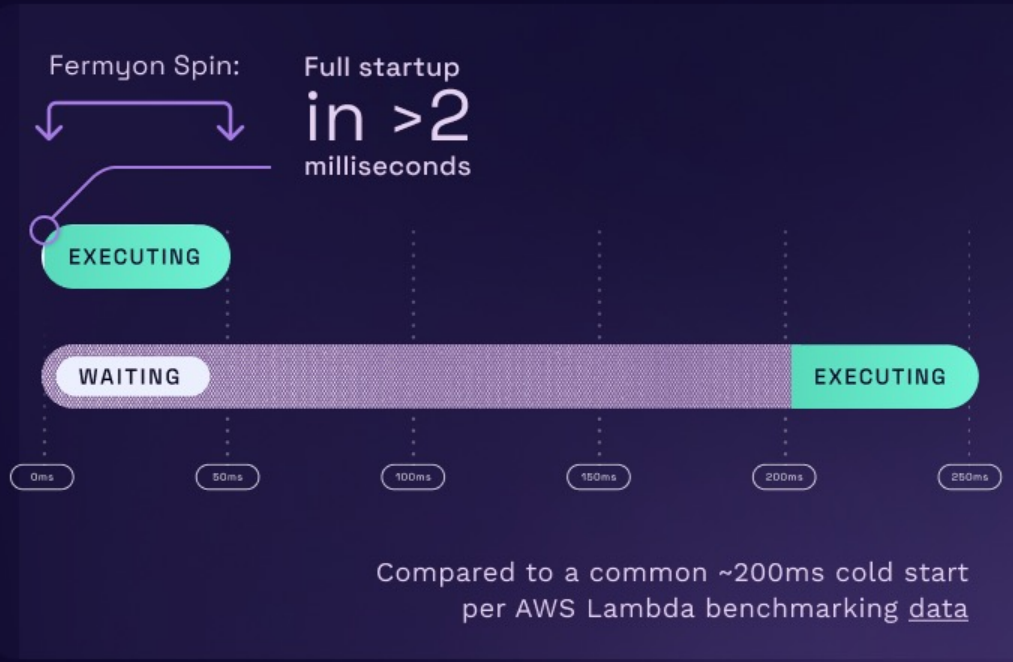


FERMYON

WHY FERMYON

Cold Starts become a thing of the past.

With WebAssembly's small size and fast start-up time, we're able to cold-start a module on every execution (within a millisecond in Spin), so it is not a problem to completely startup, execute and shutdown within the span of every request.



Compared to a common ~200ms cold start per AWS Lambda benchmarking data

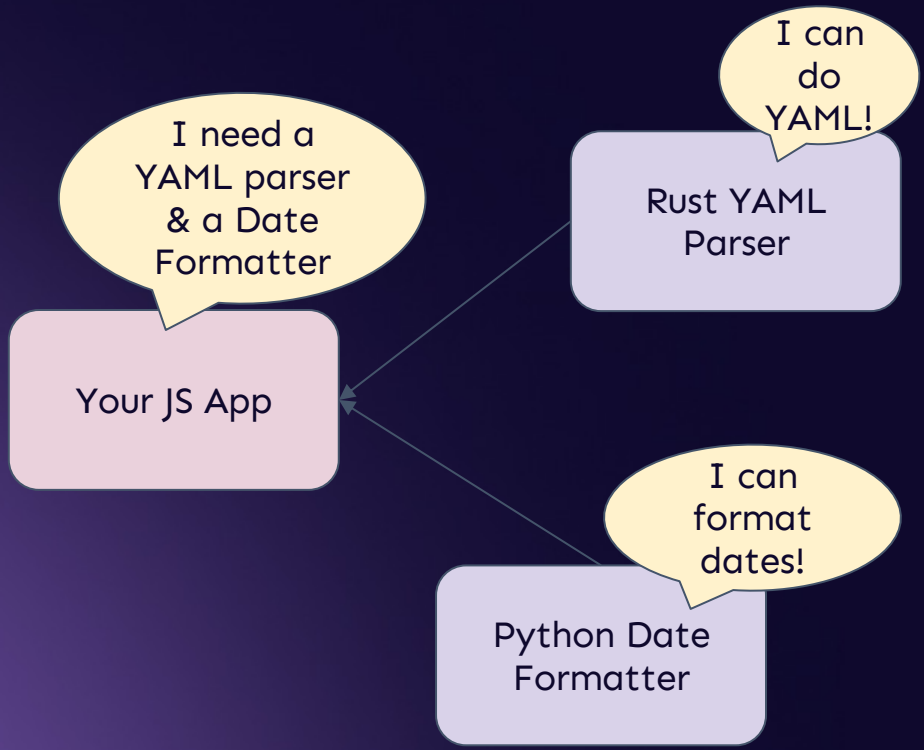
DEMO

fermyon.com/serverless-ai

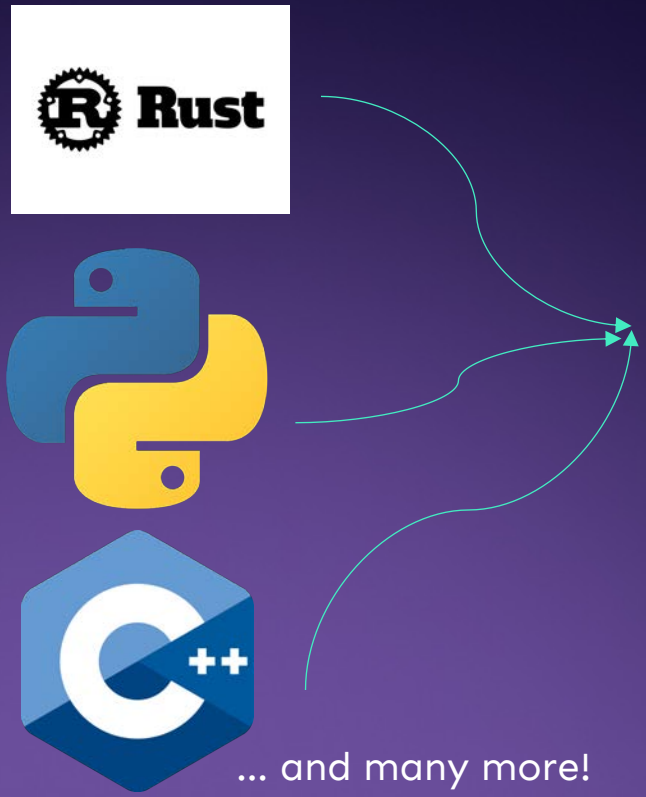
FERMYON

Component Model

- Write code in whatever language you want
- Use libraries from all over without even knowing what language they were written in!
- Wasm will take care of the rest.



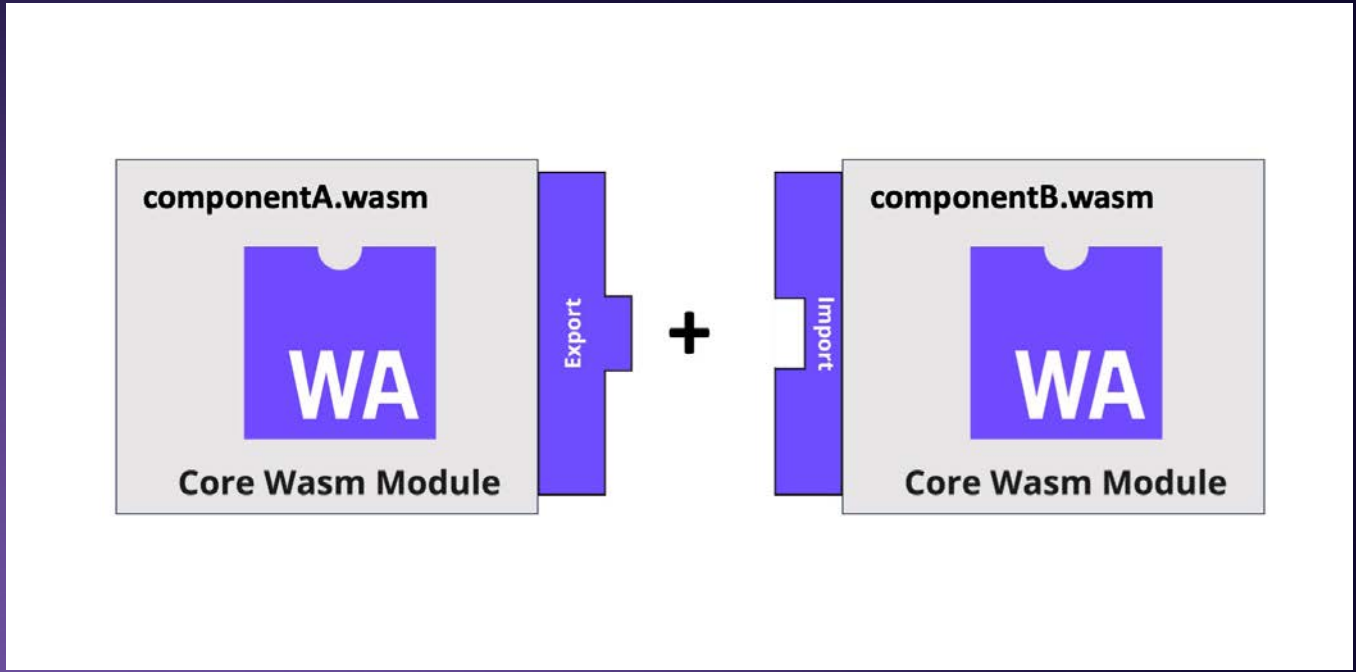
Wasm Module



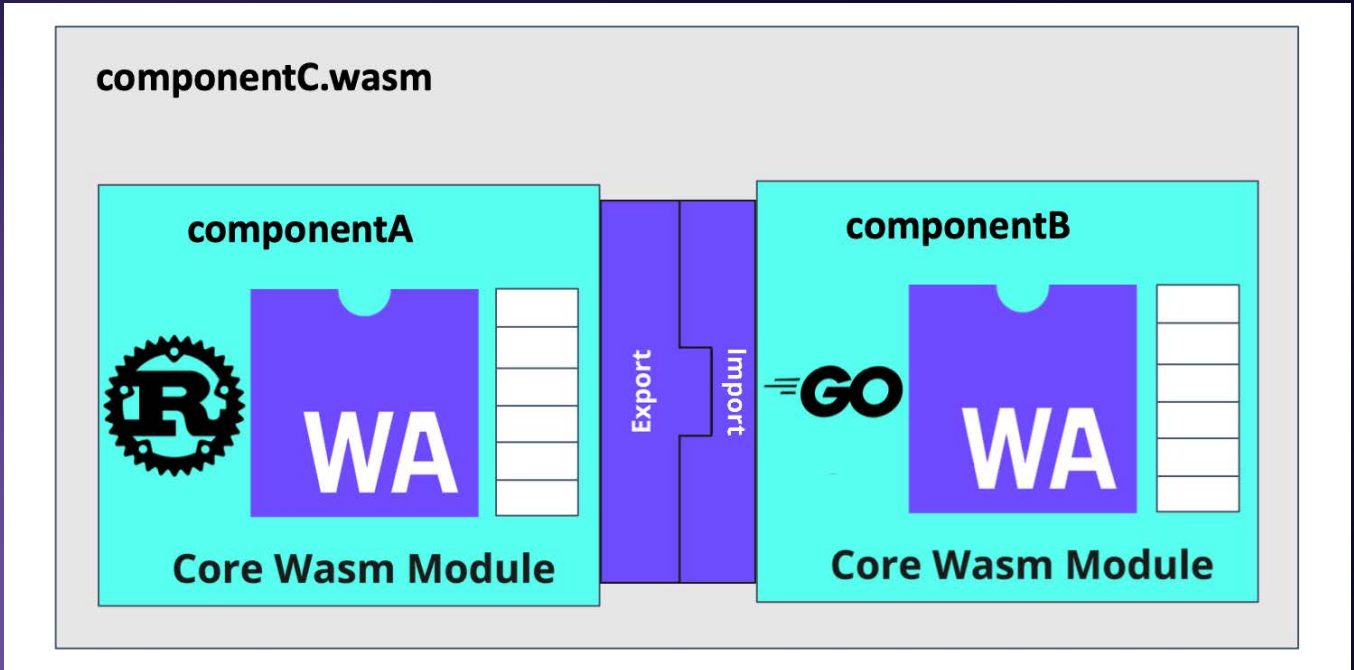
Core Wasm Module

- Portable
- Small .wasm
- Isolated
- Fast Startup
- Open Standard
- Polyglot

Composing Applications



Composing Applications



componentize-py

- Tool to convert a Python application to a WebAssembly component.
- It takes the following as input:
  - a WIT file or directory
  - the name of a WIT world defined in the above file or directory
  - the name of a Python module which targets said world
  - a list of directories in which to find the Python module and its dependencies

github.com/bytecodealliance/componentize-py

Join our Discord server!

Check out Spin!

Thank You!

linkedin.com/in/sohanmaheshwar/

developer.fermyon.com

FERMYON