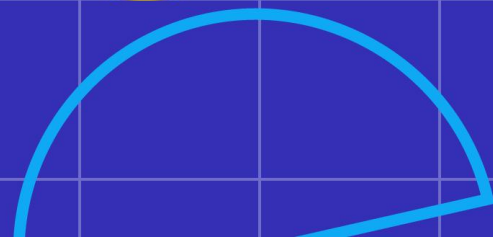
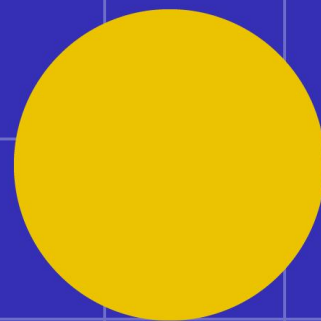
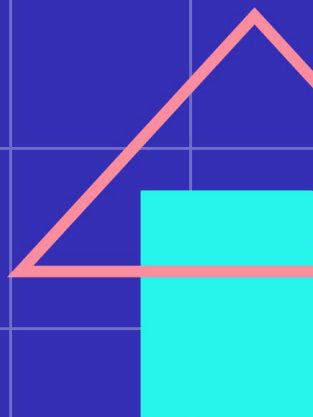


MoonBit 编程语言 (WASM 技术) 服务端应用展望 以及对Kubernetes生态的影响

沙渺 (MoonBit 语言社区开发者)





敬请注意

本分享包含大量目前尚处在早期开发阶段甚至概念阶段，尚未获得广泛应用的技术。

仅为前景展望，不推荐用于当前立项开发的实际工程。



内容

- WASM 技术栈现状和 WASM 后端应用的构想
- MoonBit 语言介绍
- MoonBit 方案对 WASM 技术栈的作用
- 对 Kubernetes 提出的挑战



分享者

- 沙渺，大连本地独立开发者
- Rust 语言社区早期参与者
The Rust Programming Language（中文纸质版）译者3
- MoonBit 语言社区开发者
官方标准库代码、教程资料编写者



WebAssembly (WASM)



WEBASSEMBLY

```
(block $block/118 (result i32) (i32.const 0)
(loop $loop/117 (param i32) (result i32) (local.set $i/82)
  (local.get $i/82)
  (if (result i32)
    (then (local.get $i/82) (call $$builtin.println|Bool|.fn/23) (drop)
      (local.get $i/82) (i32.eqz) (br $loop/117))
    (else (i32.const 0))))))
(drop)
(array.new_data $Array<UnsafeMaybeUninit<Int>> $moonbit.const_data
  (i32.const 300) (i32.const 5))
(i32.const 5) (struct.new @$moonbitlang/core/builtin.Array<Int>)
(local.tee $p/83) (i32.const 0) (call $Array::op_get|Int|.fn/17)
(call $$builtin.println|Int|.fn/24) (drop) (local.get $p/83) (i32.const 1)
(call $Array::op_get|Int|.fn/17) (call $$builtin.println|Int|.fn/24)
(drop) (local.get $p/83) (i32.const 4) (i32.const 11)
(call $Array::op_set|Int|.fn/29) (drop) (local.get $p/83) (i32.const 4)
(call $Array::op_get|Int|.fn/17) (call $$builtin.println|Int|.fn/24)
(drop)
(block $block/116 (result i32) (i32.const 0)
(loop $loop/115 (param i32) (result i32) (local.set $i/84)
  (local.get $i/84) (local.get $p/83) (call $Array::length|Int|.fn/16)
  (i32.lt_s)
  (if (result i32)
```



WASM 现状

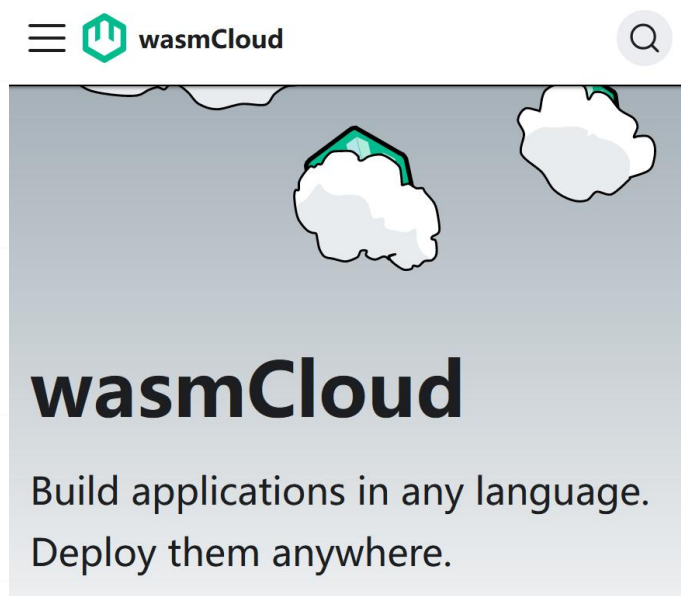
WebAssembly 1.0 has shipped in 4 major browser engines.



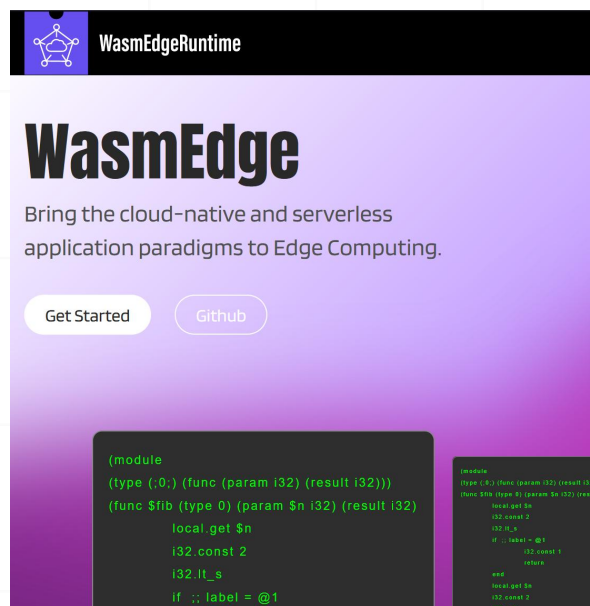
- 1.0 MVP 已稳定（2017）已被 Web 前端应用实际接受
- 有后端应用潜力，但更多具体疑问仍待解答
- 问题集中在扩展特性（features）、外部语言接口（FFI）、操作系统界面（system interfaces 或 WASI）上



WASM 后端应用先驱



wasmCloud
(CNCF-Sandbox)



WasmEdge
(CNCF-Sandbox)



fermyon/spin



WASM 扩展特性

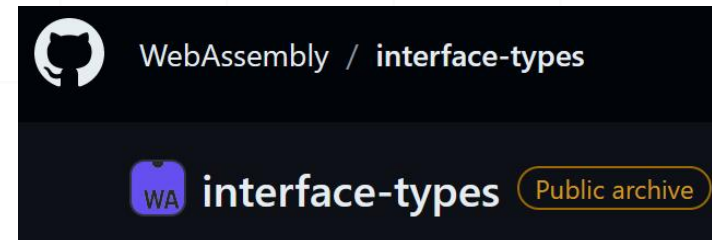
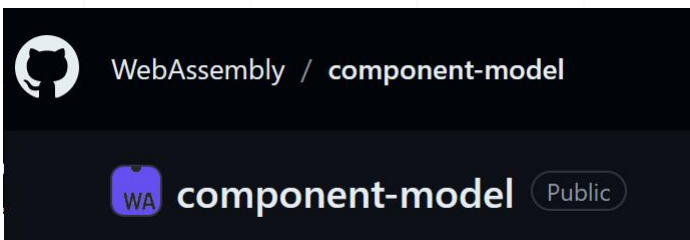
- 多数特性已标准化
- 一些特性对后端应用不可或缺
- 但各个运行时的实现，成熟度不一
- 使用扩展特性，基本需要限定运行时

	Your browser	Chrome	Firefox	Safari	Wasmtime	Wasmer	Node.js	Deno	wasm2c
Standardized features									
JS BigInt to Wasm i64 integration	✓	85	78	15 ^[e]	N/A	N/A	15.0	1.1.2	N/A
Bulk memory operations	✓	75	79	15	0.20	1.0	12.5	0.4	1.0.30
Extended constant expressions	✓	114	112	17.4	✗	✗	✗ ^[i]	1.33	✗ ^[s]
Garbage collection	✓	119	120	✗	✗	✗	22.0	1.38	✗
Multiple memories	✓	120	125	✗	15	✗	✗ ^[i]	1.38	✗ ^[u]
Multi-value	✓	85	78	13.1	0.17	1.0	15.0	1.3.2	1.0.24
Mutable globals	✓	74	61	12	✓	0.7	12.0	0.1	1.0.1
Reference types	✓	96	79	15	0.20	2.0	17.2	1.16	1.0.31
Relaxed SIMD	✗	114	✗ ^[d]	✗	15	✗	✗ ^[m]	1.33	✗
Non-trapping float-to-int conversions	✓	75	64	15	✓	✓	12.5	0.4	1.0.24
Sign-extension operations	✓	74	62	14.1 ^[f]	✓	✓	12.0	0.1	1.0.24
Fixed-width SIMD	✓	91	89	16.4	0.33	2.0	16.4	1.9	1.0.33
Tail calls	✓	112	121	✗	22 ^[h]	✗	20.0	1.32	✗ ^[v]
Threads and atomics	✓	74	79	14.1 ^[f]	15	✗	16.4	1.9	✗
In-progress proposals									
Exception handling	✓	95	100	15.2	✗	✗	17.0	1.16	✗ ^[r]
JS Promise Integration	✗	✗ ^[a]	✗ ^[c]	✗	N/A	N/A	✗ ^[l]	✗ ^[o]	N/A
Memory64	✗	✗ ^[b]	✗ ^[d]	✗	✗ ^[g]	✗	✗ ^[k]	✗ ^[p]	✗ ^[t]
Type reflection	✗	✗ ^[b]	✗ ^[d]	✗	✗	2.0	✗ ^[n]	✗ ^[q]	✗



WASM 外部语言接口 (FFI)

- 基本接口已在 WASM 1.0 标准化
- 但只能交换简单数据类型
- 交换缓冲区和高级数据结构的方法各有不一





WASM 系统接口 (WASI)

- WASI (WebAssembly System Interface)
- 用于允许 WASM 代码调用操作系统的能力 (stdout、socket这些)
- 目前处在 0.2 预览版
- 功能缺口和争议都很大





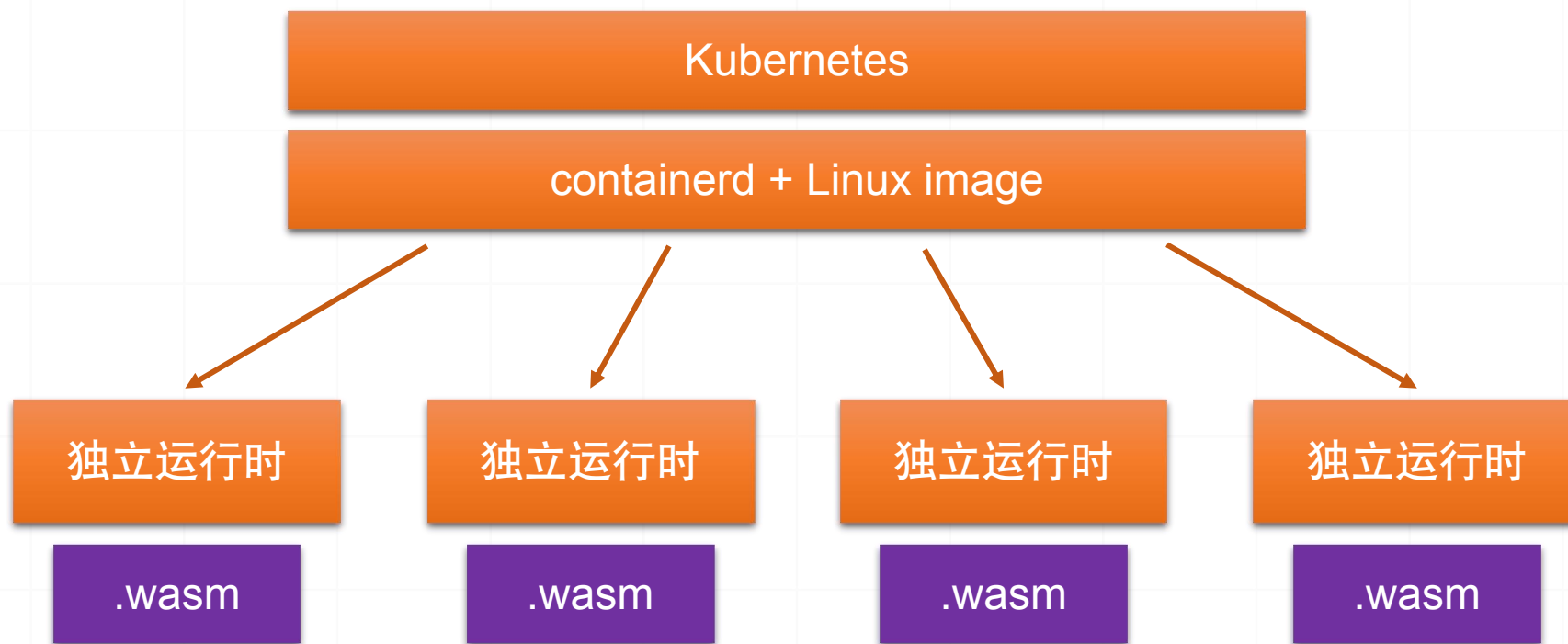
WASM 应用于后端的构想初步存在

但具体的挑战非常巨大



Kubernetes + WASM

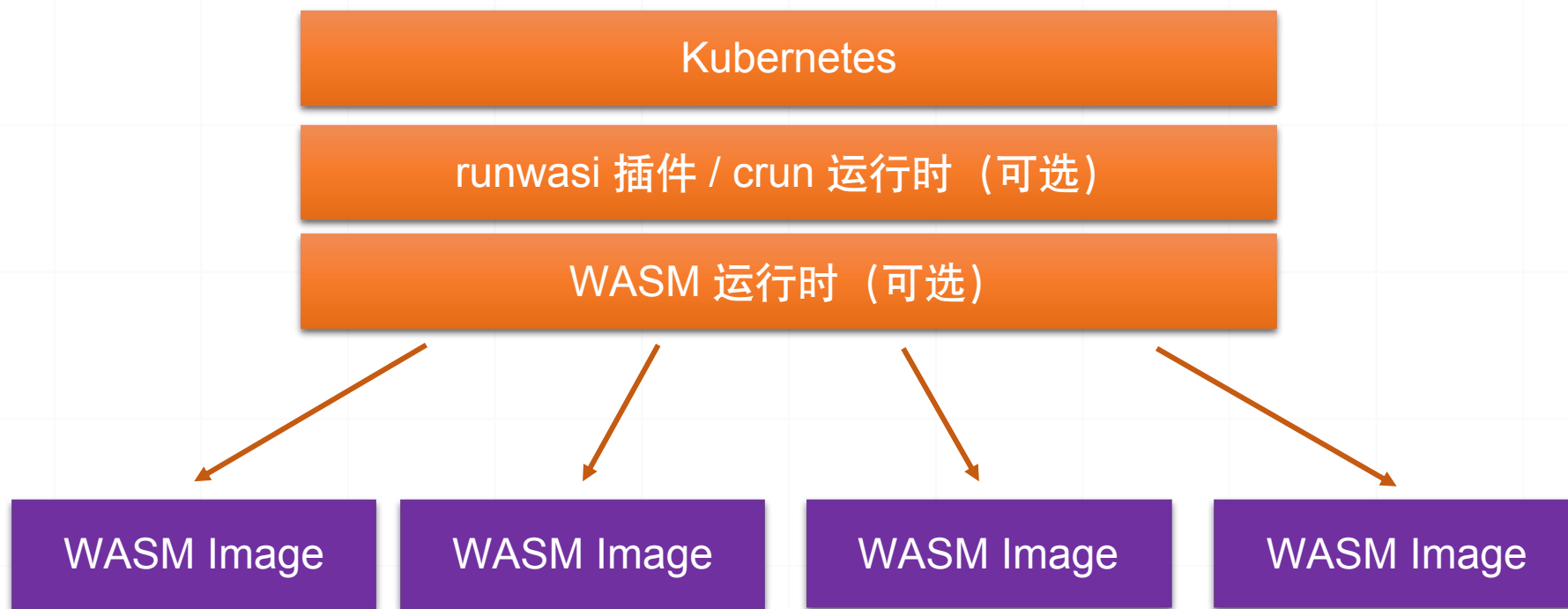
- 方法（1）将 WASM 独立运行时程序看作工作负载





Kubernetes + WASM

- 方法（2）直接将 WASM 看作镜像的可选架构





Search Docker Hub

ctrl+K



Sign In

Sign up

Explore / wasmedge/example-wasi-http



wasmedge/example-wasi-http

Sponsored OSS ☆1

↓ Pulls 10K+

By [WasmEdge](#) · Updated about 1 year ago

A OCI compatible image provides Wasm-Wasi HTTP application demonstration.

IMAGE

Overview

Tags

Sort by

Newest ▼

Filter Tags



TAG

[aot-ubuntu2004-x86-64](#)

Last pushed a year ago by [hydai](#)

```
docker pull wasmedge/example-wasi-http:aot-ubuntu2004-x86-64
```

Copy

Digest

[dbfac3ae5d22](#)

OS/ARCH

wasi/wasm

Compressed Size ⓘ

621.79 KB



Kubernetes 用于 WASM 生态的价值

- 复用当前生态
- 大多数概念可以复用
- 个别概念对 WASM 镜像更加有用（例如节点亲和性）



对 Kubernetes 的挑战

- 层数过度设计
- 抽象粒度与 WASM 惯例的差异
- 容器间交互模型与 WASM 惯例的差异
- WASM 的细粒度观测和管理不是 Kubernetes 层的专长



Kubernetes + WASM 后端应用

架构和概念有共识，但很多问题仍待解答



Intelligent developer platform for **Cloud** and **Edge** using WASM.

Compact

◀ Tiny wasm output, start up quickly and less attack surface ▶



[Get Started](#)

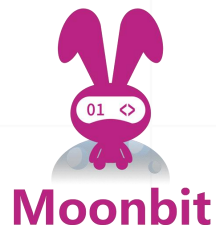
[Learn More](#)

```
> moon new hello
> cd hello
> cat main/main.mbt
fn main {
  println(@lib.hello())
}
> moon run main
Hello, world!
```



MoonBit 程序设计语言

- 全新语言（2023 -），专业面向 WASM 目标
- 具有类似于 Rust 的现代化语言特性，带 GC
- 当前阶段：社区共建和Pre-beta测试阶段
- 时间表：24/08实现core稳定，25 年 1.0



(台上有资料可领取)



MoonBit 与其他语言

- MoonBit vs Rust
- MoonBit vs Golang
- MoonBit vs JavaScript
- MoonBit vs PHP, Lua, Python



云原生社区
Cloud Native Community

云原生社区 MEETUP



CLOUD NATIVE
LANDSCAPE

EXPLORE

GUIDE

STATS

Search items

Filters

GROUP:

Projects and products

Members

Certified partners and providers

Serverless

Wasm

CNAI

VIEW MODE:

Grid

Wasm

Languages



MoonBit



WASM focused languages

<https://landscape.cncf.io/?group=wasm>



MoonBit 作为 WASM 原生语言的作用

- 降低入门门槛（不必非得学习 Rust）
- 平衡代码尺寸和性能
- 按照 WASM 基础规范设计的 FFI 外部语言接口（支持对外调用和外部回调）
- 直接对接 WASM 扩展（例：wasm-gc target）
- 直接对接 WASI 系统接口标准



新语言对 WASM 后端技术栈的意义

- 封装，或者说“过顶”开发思路 (OTT, over the top)
- 有限使用 WASM 特性，基本只当作 ISA (指令集)
- 绕过 WASM 低级概念，转而使用语言的高级概念
- 牺牲语言互换性，换取 WASM 下立刻应用高级特性

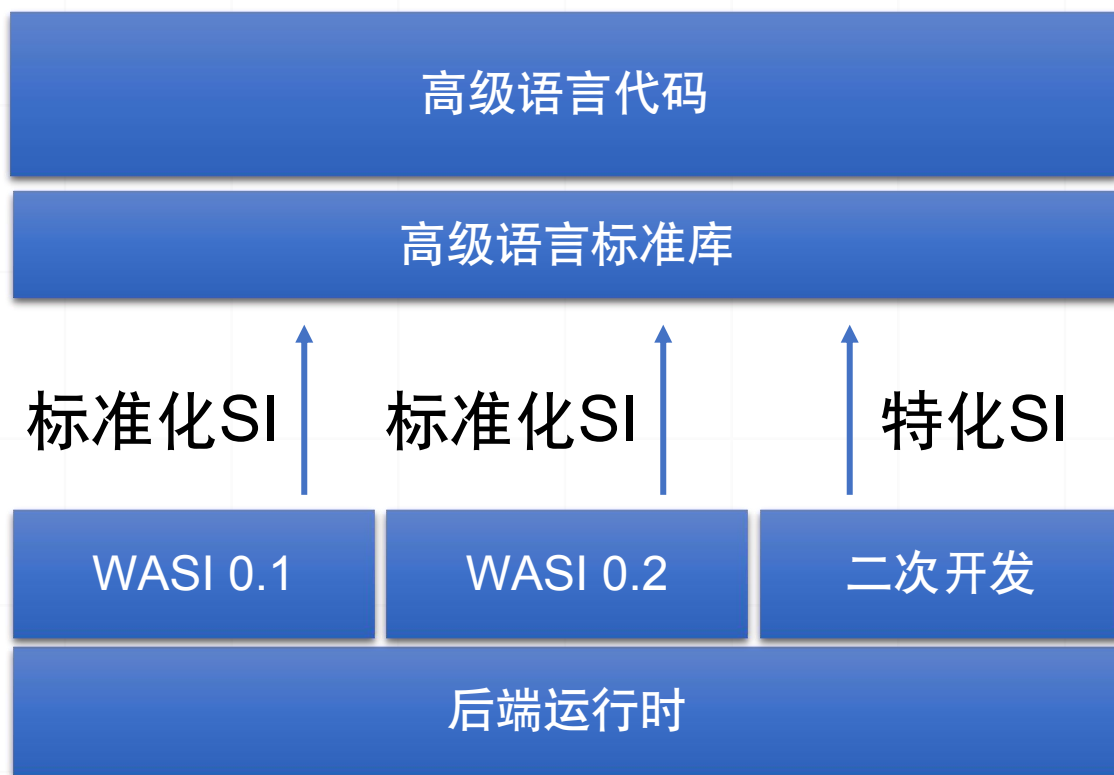


关注点分离 (1)





关注点分离 (2)



wasi v0.16.1

Author

peter-jerry-ye

License

Apache-2.0

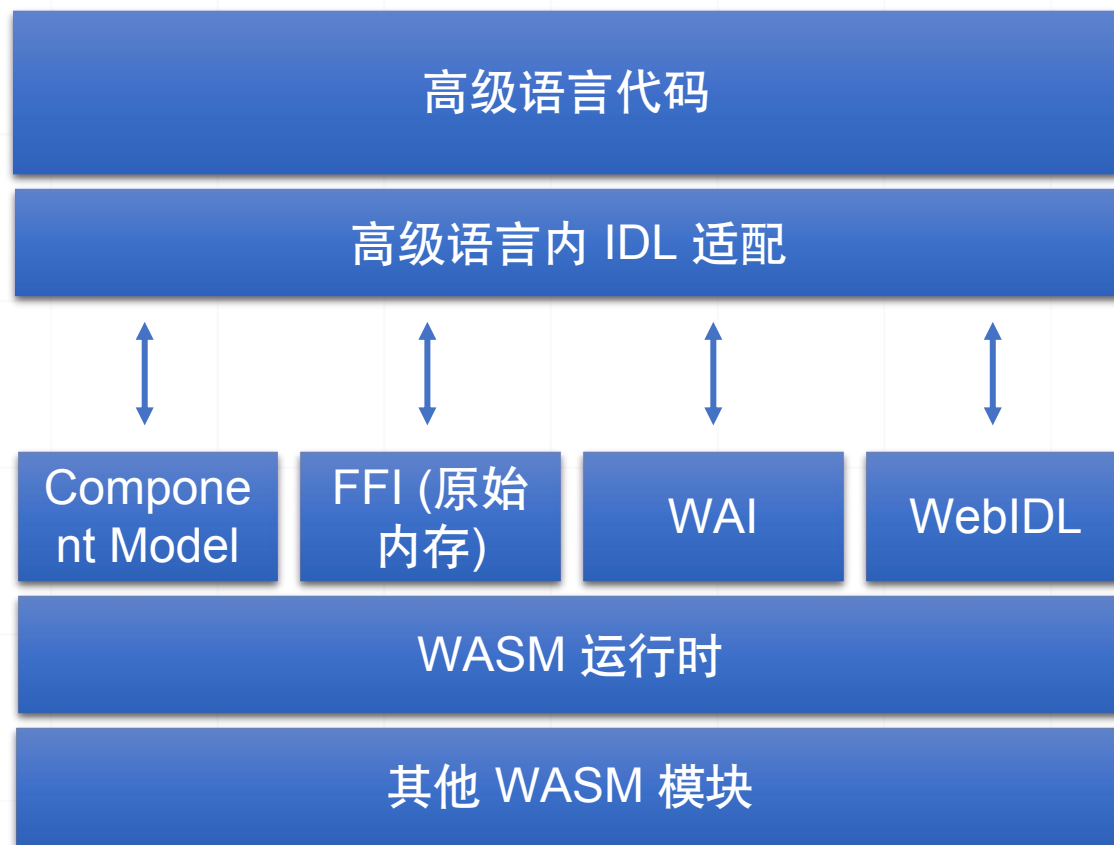
WASI

WASI Preview 1 implementation for MoonBit (GC ver)

月兔的WASI Preview 1实现



关注点分离 (3)





MoonBit 语言运用封装 / OTT 思路

在 WASM 层之上力求解决 WASM 的挑战



https://landscape.cncf.io/?group=wasm



EXPLORE

GUIDE

STATS

Search items



Filters

GROUP:

Projects and products

Members

Certified partners and providers

Serverless

Wasm

CNAI

VIEW MODE:

Grid

Card

ZOOM:

-

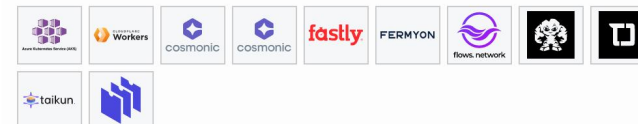
+

Wasm

Languages



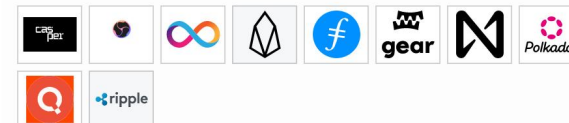
Hosted Platforms



Embedded Functions



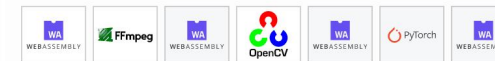
Decentralized Platforms



Application Frameworks



AI/Machine Learning



Orchestration & Management



Edge/Bare metal



Runtimes



Packaging, Registries & Application Delivery

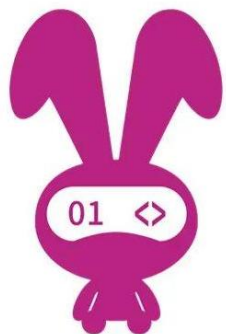


Tooling



Debugging & Observability





MoonBit

【MoonBit 官网】 www.moonbitlang.cn

【MoonBit 论坛】 taolun.moonbitlang.cn

【MoonBit 代码】 GitHub: moonbitlang

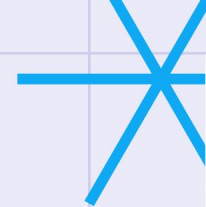
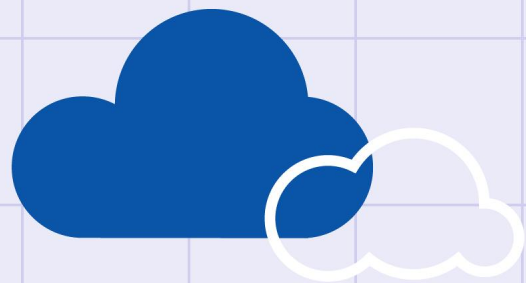
【本次分享者】 GitHub: shamiao



← 扫这个加 MoonBit 官方群
(微信号: moonbit_helper)

加我交流具体应用研究→
(微信: shamiao)





感谢观看



云原生社区
Cloud Native Community

(本分享的资料后续上传到 GitHub 的 [shamiao/kcd24](#))

