WebAssembly Micro Runtime：云原生时代的超轻量级新型沙箱

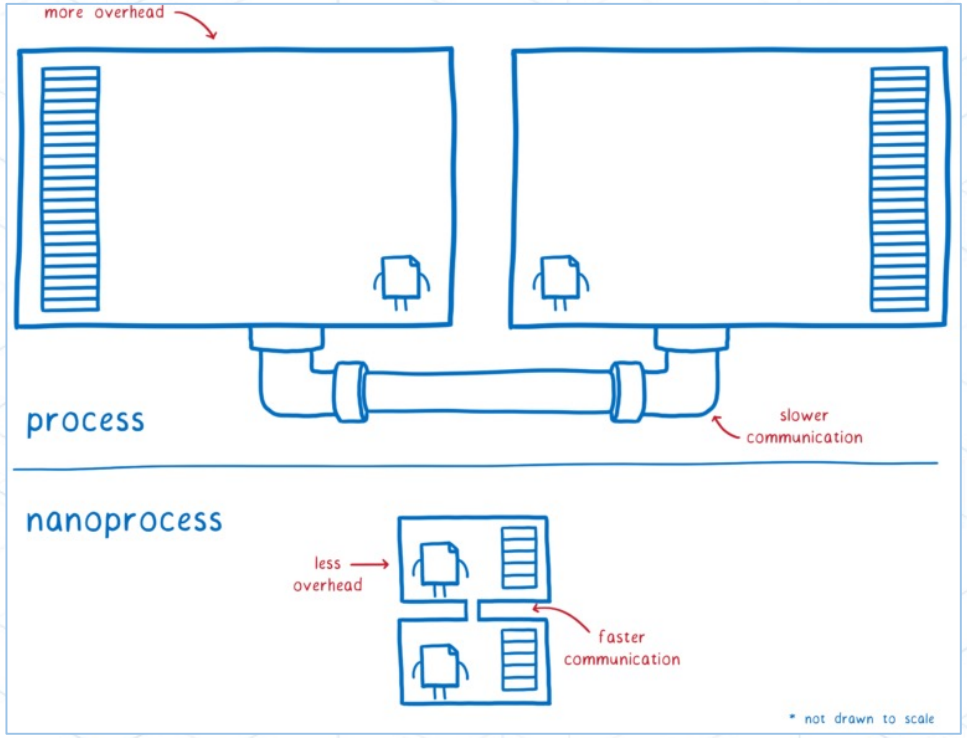
王鑫， 英特尔

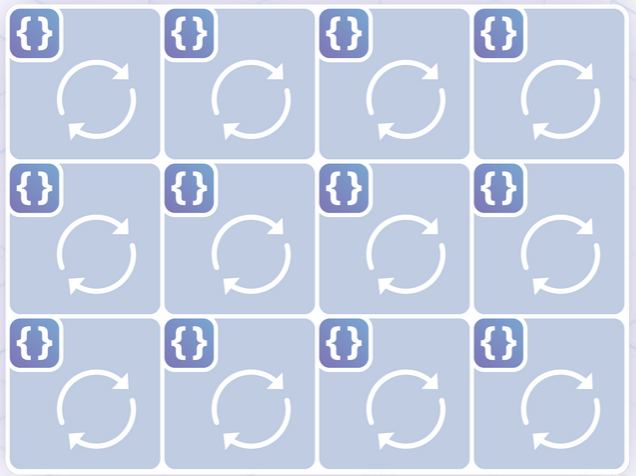
2021-05-22

目 录
contents

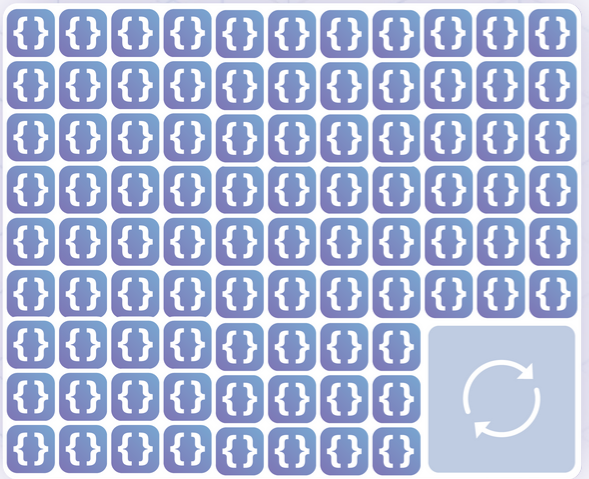
一、 WebAssembly 的内存安全特性与云原生

WebAssembly Nano-Process 带来更细的隔离粒度





Virtual machine



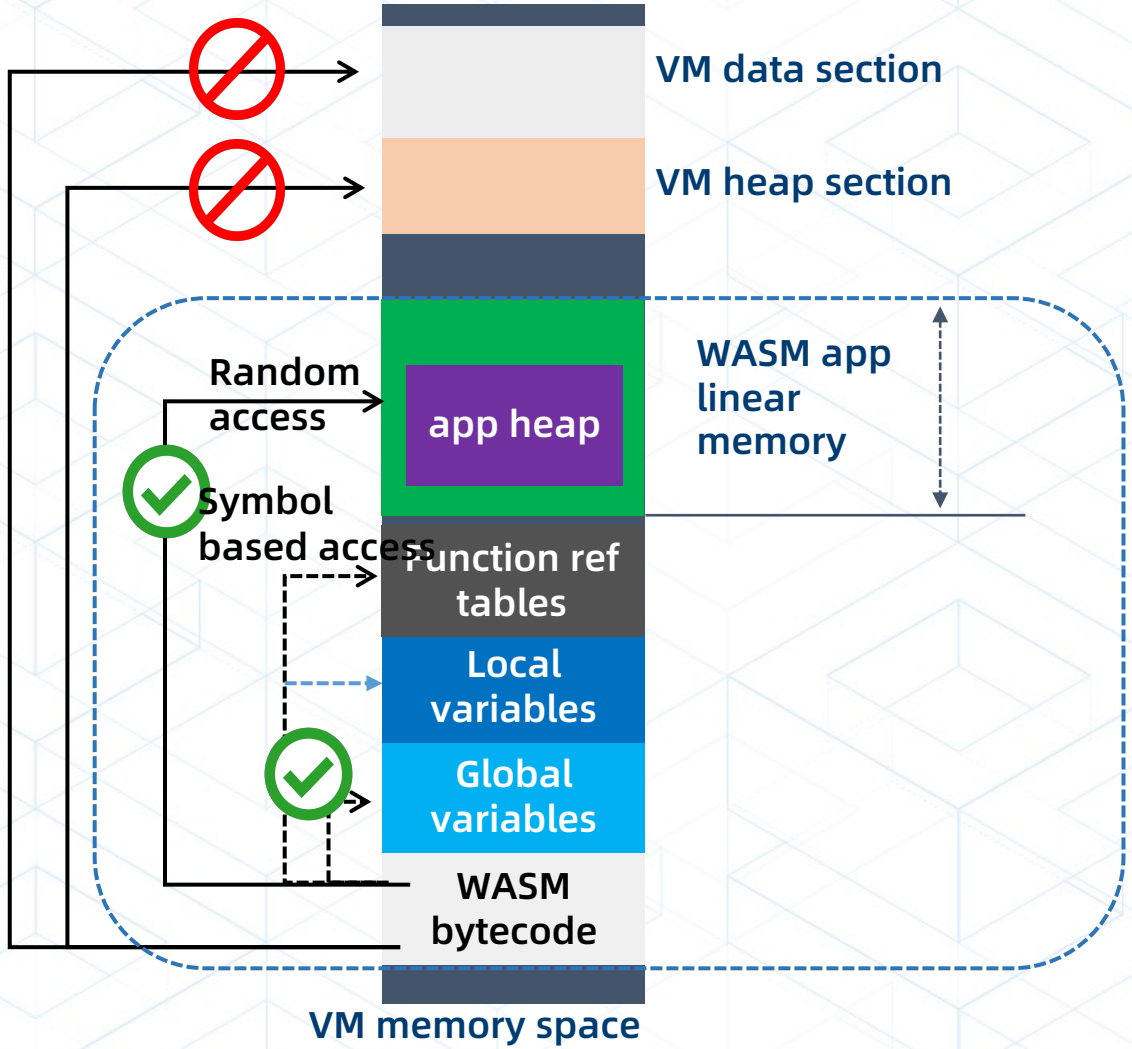
Isolate model

User code

Process overhead

Source:
https://blog.cloudflare.com/cloud-computing-without-containers/
https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

WASM 应用沙箱内存访问机制



- WASM 规范定义两个基本操作码 iload/istore 来允许 APP 随机访问线性空间(Linear memory)
- WASM APP 通过符号 (symbol) 访问函数、全局变量、局部变量
- 符号访问通过 WASM 字节码中静态索引操作数和文件结构中引用表来完成
- LLVM 的 WASM 编译后端使用了线性空间来访问所有需要地址访问的操作，所以局部变量如果有指针引用也在线性空间分配地址

5

托管语言内存安全的共性与Wasm的特性

Memory safe languages discussed here: Wasm, Java, JS, Python

- 栈内数据带类型标记
- 保证应用程序不能访问栈
- 托管对象与垃圾回收机制
- 受管的函数调用栈
- 分支保护
- 函数地址与调用
- 沙箱与随机地址访问
- 第三方模块引入保护
- 原生函数的调用机制

二、WebAssembly Micro Runtime(WAMR) 项目背景

WAMR项目发展历史

□ 英特尔于2019年5月开源 WAMR 项目 (解释器)

- 团队: 在 managed runtime 领域超过 15 年历史。从零开发轻量级 Java 语言运行时与 WAMR 项目。PWA/NodeJS/V8 优化。
- 动机: 跨越从嵌入式到云端的托管代码运行环境

□转入字节码联盟 (Bytecode Alliance)，2019年11月

- WasmTime 和 WAMR 成为 BA 下两个主要的 Wasm 运行引擎开源项目
- 2020年完成 JIT 和 AoT 支持，以及绝大部分 Post MVP 特性支持
- 2021年字节码联盟正式成为非营利组织，新增微软、谷歌和 Arm 等成员

三、WAMR 技术架构与优异性能

WAMR 使命与设计目标

- 高度模块化与可定制化 - 满足从嵌入式到云端的容器引擎需求的碎片化
- 最好的性能、最小的内存消耗
- 支持解释器、Ahead-of-Time(AoT)编译、Just-in-Time(JIT)编译多种模式
- 多CPU架构、多平台支持
- 自主实现 AoT loader，任何平台上都能支持 AoT 模式
- 多实例支持与管理
- 提供可扩展的 WASM 微服务应用编程框架（异步模式）
- 远程 WASM 应用管理

面向云端的优异的执行性能

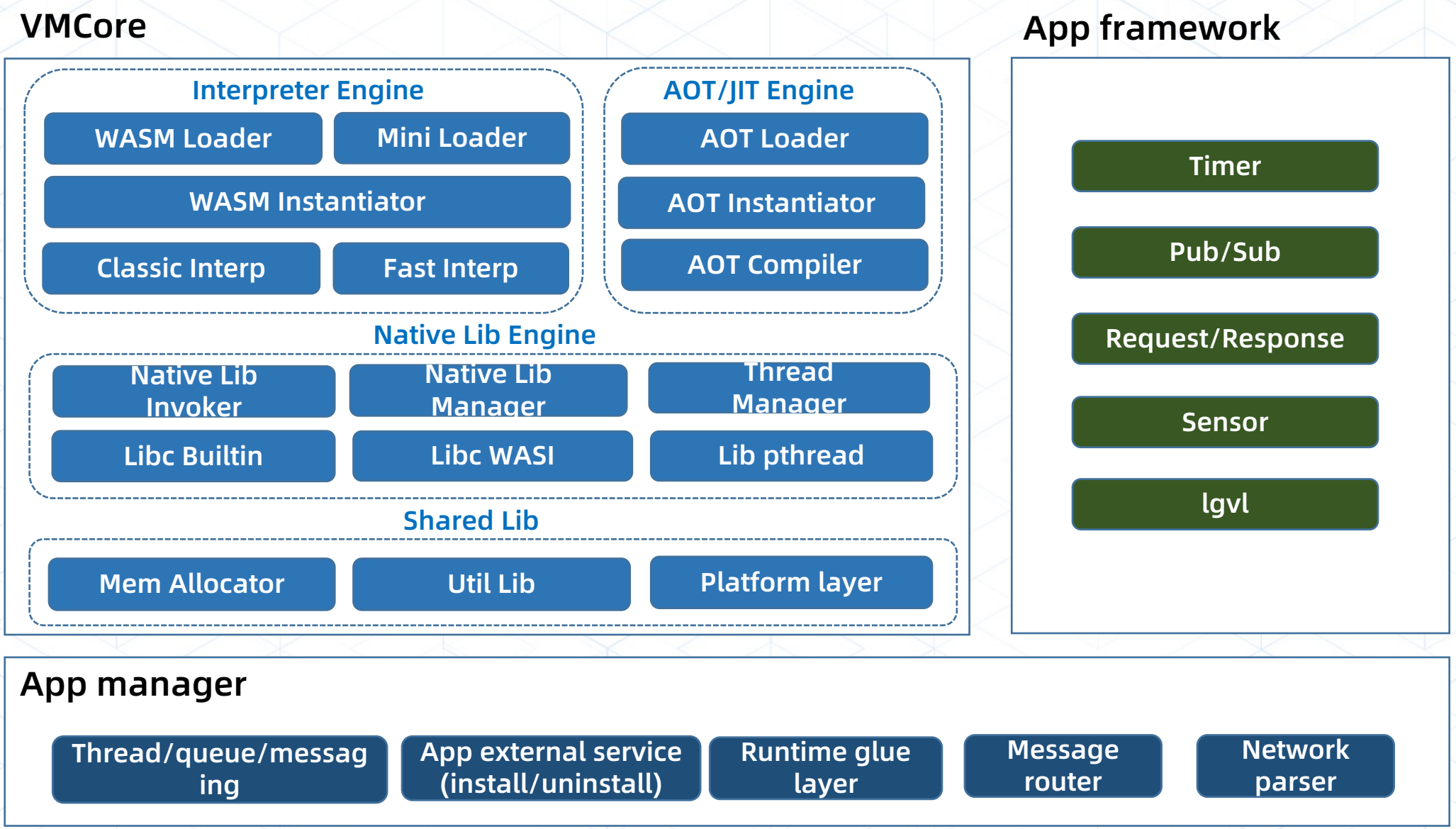
workload	native	wamr-aot	wavm	wasmer	wasmtime	nodejs
base64	0.2	0.2	0.2	0.33	0.3	0.29
fib2	0.53	0.74	0.79	0.99	1.36	1.18
gimli	0.16	0.17	0.16	0.12	0.23	0.18
heapsort	0.71	0.67	0.7	0.79	1.05	0.99
matrix	1.19	1.56	1.69	1.75	2.75	2.15
memmove	2.74	7.97	9.3	9.31	16.85	10.83
nestedloop	4.89	4.89	4.89	4.9	6.48	5
nestedloop2	3.3	3.3	3.3	3.33	3.97	3.38
nestedloop3	3.36	3.36	3.36	3.37	4.4	3.5
random	0.42	0.3	0.3	0.3	0.31	0.32
seqhash	2.33	3.16	3.18	3.28	4.06	3.32
sieve	0.26	0.33	0.32	0.35	0.47	0.54
strchr	0.37	0.25	0.37	0.25	0.49	0.39
switch2	0.23	0.46	0.2	0.44	0.33	0.29
coremark(score)	31660	25330	25059	22323	15740	16892

i7-7700 CPU @ 3.60GHz
Ubuntu 18.04 64-bit
gcc-7.5.0
wasi-sdk-12, emsdk-2.0.6

Unit: second except coremark
Smaller is better except coremark

WASM VM	language	codegen
wamr	C	LLVM
wavm	C++	LLVM
wasmer	Rust	LLVM
wasmtime	Rust	CraneLift
nodejs	C++	Liftoff/Turbofan

WAMR 技术架构


VMCore
Interpreter Engine
WASM Loader
Mini Loader
WASM Instantiator
Classic Interp
Fast Interp
AOT/JIT Engine
AOT Loader
AOT Instantiator
AOT Compiler
Native Lib Engine
Native Lib Invoker
Native Lib Manager
Thread Manager
Libc Builtin
Libc WASI
Lib pthread
Shared Lib
Mem Allocator
Util Lib
Platform layer
App framework
Timer
Pub/Sub
Request/Response
Sensor
lgvl
App manager
Thread/queue/messaging
App external service (install/uninstall)
Runtime glue layer
Message router
Network parser

WAMR 技术参数

- 实现语言：C
- 解释器：fast and classic 分别面向速度优先和内存优先
- 编译后端：LLVM
- 目标文件大小
  - VMCore - 60K for AoT, 90K for interpreter
  - 150K for the whole runtime
- AoT 支持平台：Linux, SGX, MCU
- SIMD 支持
- SGX/TDX 一等公民支持
- Wasm 多线程特性与 pthread 支持
- Reference type, Multi-modules

- CPU Arch support:
  - X86-64, X86-32
  - ARM, THUMB, AARCH64
  - MIPS
  - XTENSA
- Platform support:
  - Linux，SGX (Linux)
  - Windows
  - MacOS
  - Android
  - Zephyr, AliOS Things
  - Vxworks
  - Nuttx, RT-thread
  - iOS*

Workloads:

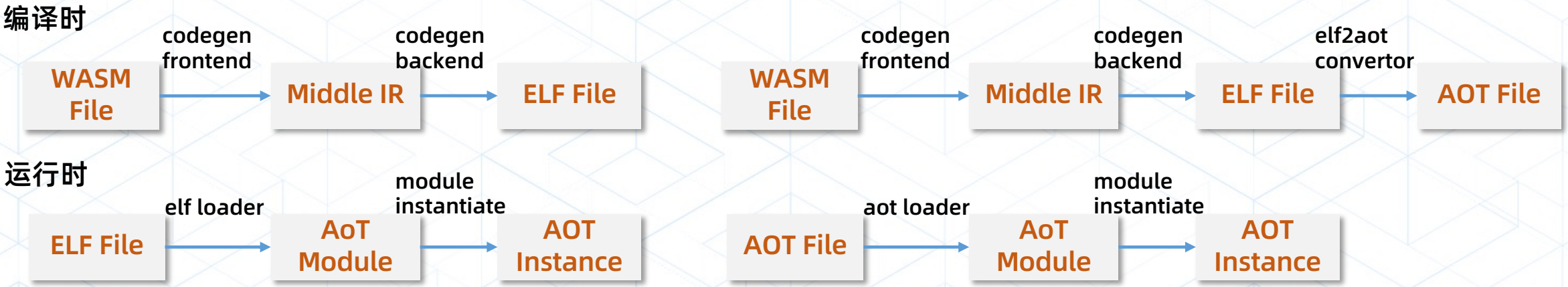
- Tensorflow lite
- XNNPack
- wasm-av1
- bwa
- meshoptimizer

https://github.com/bytecodealliance/wasm-micro-runtime/wiki/Performance

WAMR AoT 目标格式设计选择

自定义的 AoT 文件格式，自主实现的 AoT 模块加载器

- 不依赖 Linux 的 elf loader，更好适应多平台如 Windows, Intel SGX, MCU等
- WAMR来负责加载和执行 AoT 文件
- 进一步减少文件大小



选择一：采用 elf 文件格式:

选择二：自定义的 AoT 文件格式:

Adopted

四、WAMR 的云端应用场景

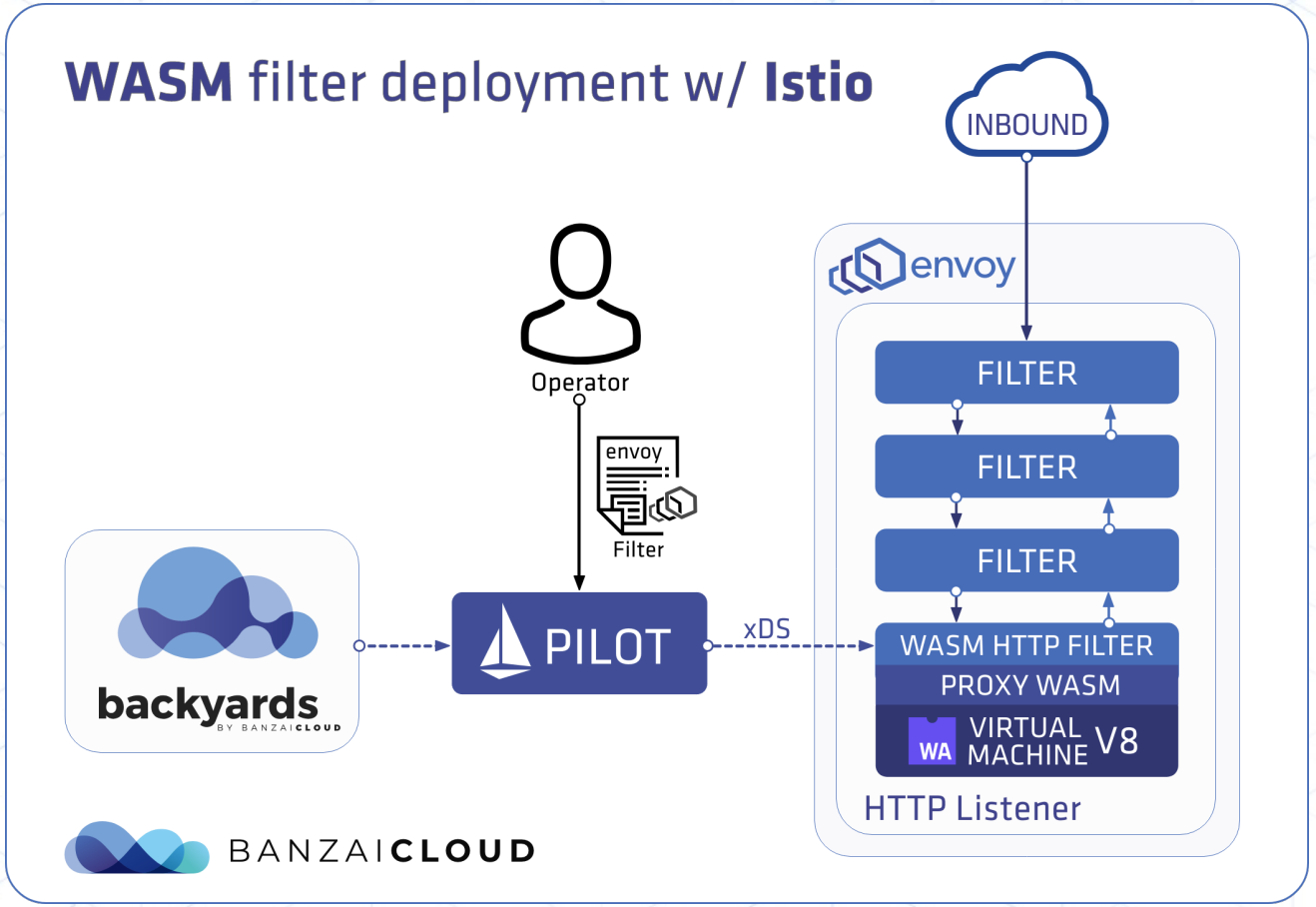
WAMR 正成为 Envoy 的又一个 Wasm 引擎

增加 WMAR 到 Envoy 正在进行中：

- [proxy-wasm-cpp-host PR142](): 已提交
- [EnvoyProxy PR16057](): 正在 **Review** 中

期望：

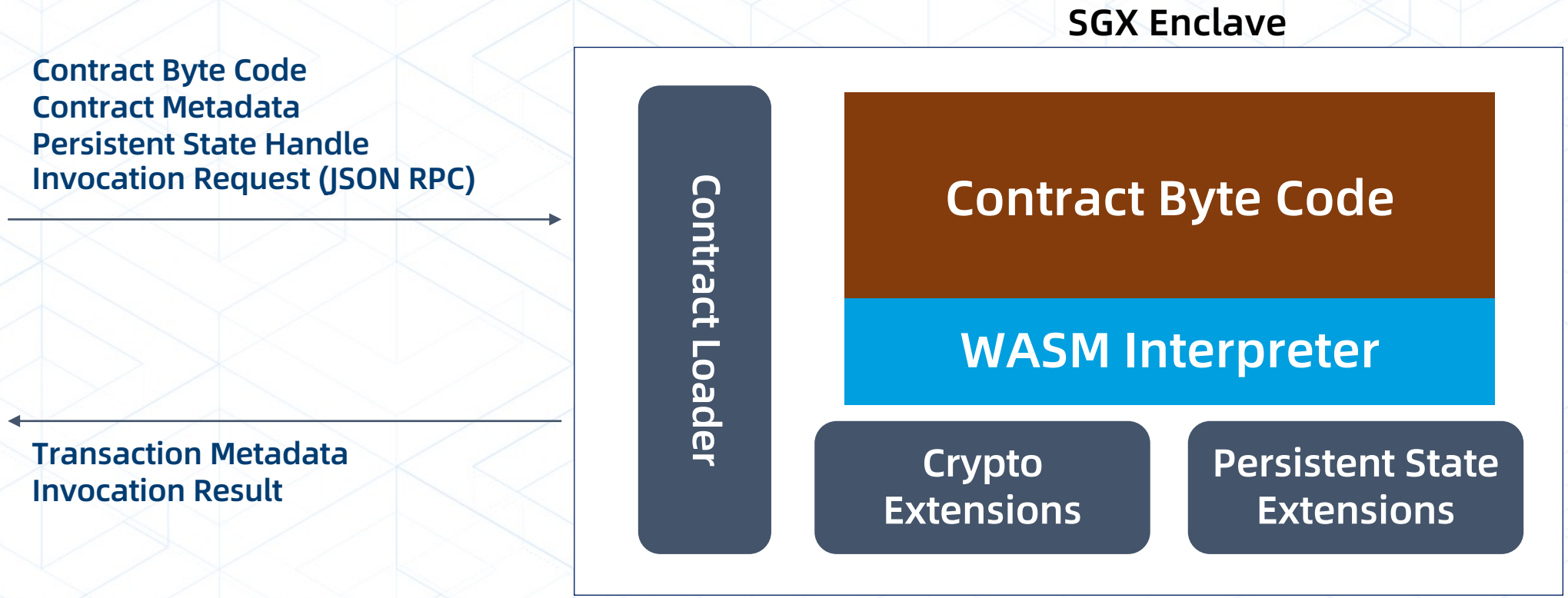
- 带来更好执行性能，更小的内存消耗
- 持续改进
- 成为 CSP 个性化的基础



Source: https://banzaicloud.com/blog/envoy-wasm-filter/

Private data objects: Smart Contracts for SGX

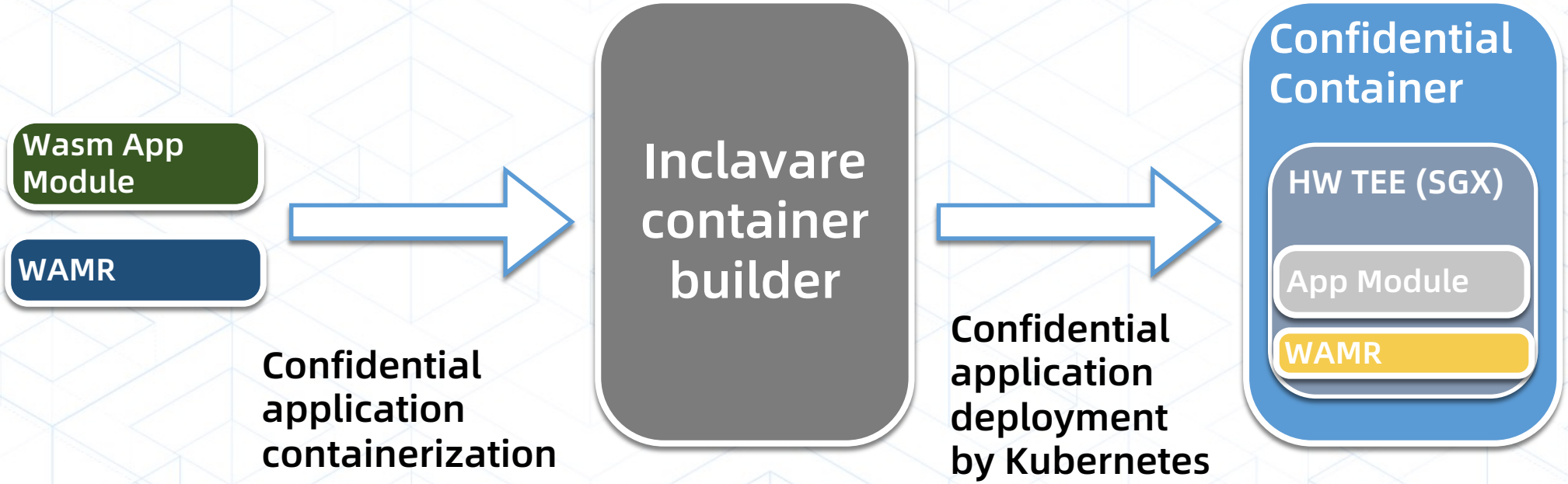
Wawaka Interpreter For Private Data Objects



Byte code - provided by the contract writer
WASM interpreter - provided by WAMR
Loader and Extensions - provided by PDO

https://github.com/hyperledger-labs/private-data-objects

Inclavare Container project: Build confidential containers with Wasm for SGX



An infra that builds confidential containers for TEE and extend Kubernetes cross the TEEs

- Build confidential containers cross supporting multiple hardware-based TEEs
- Run unmodified application modules in hardware-based TEE
- Provide confidentiality, integrity, and attestation for the application modules

https://github.com/alibaba/inclavare-containers

五、社区情况与发展规划

WAMR 开放治理规划中

WAMR 社区发展非常迅速，在中国尤其好

蚂蚁团队积极加入共建，秦晓康成为英特尔外的第一个 code owner

开放治理模式正在规划中，期望很快官宣。尚有委员会席位，快来！

非常期待大家参加贡献，对于未来我们有很多的想法

谢谢

👉想了解更多，微信扫码，关注公众号👈