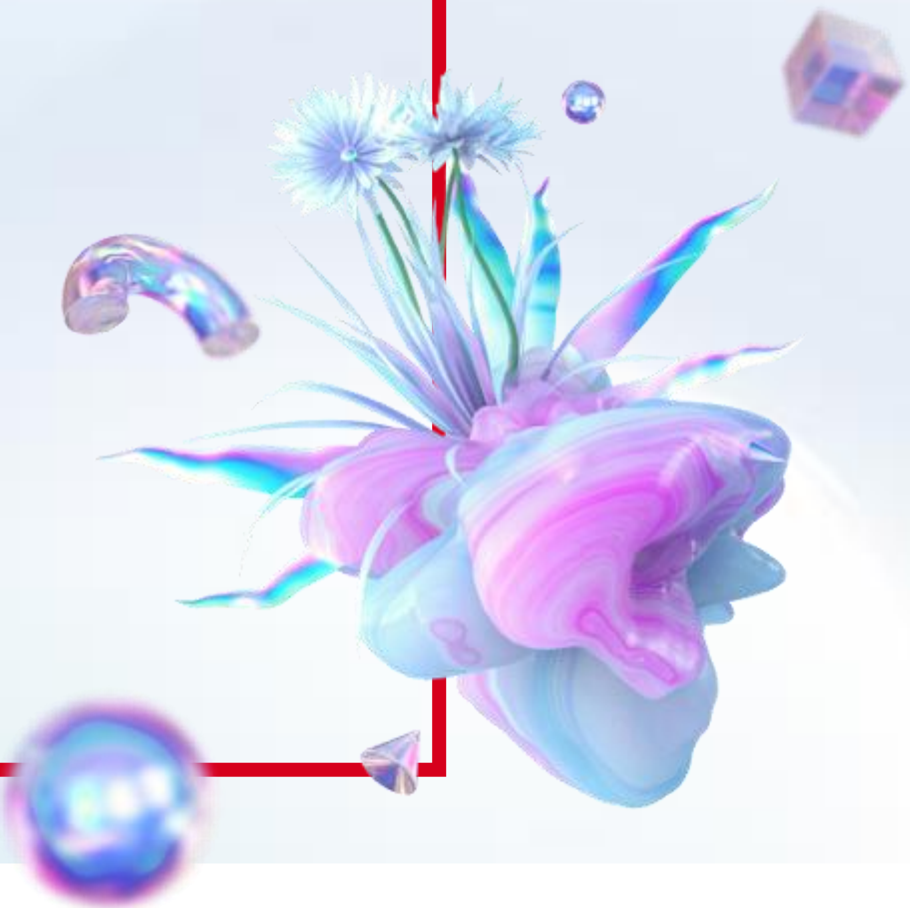
WebAssembly/WASI, Docker container, Dapr, and Kubernetes better together



Thang Chung – NashTech

Nov 2023

Thang Chung

Technical Manager, NashTech VN
Microsoft Azure MVP

- Creator of Vietnam Microservices Group on Facebook (>15k members).
  - https://www.facebook.com/groups/645391349250568
- Experience: >16 years in software consult, design, development, and deployment software for outsourcing, product, and startup companies.
- Expertise in cloud computing, cloud-native platform, serverless, and WebAssembly/WASI.
- Blog: https://dev.to/thangchung
- GitHub: https://github.com/thangchung
- LinkedIn: https://www.linkedin.com/in/thang-chung-2b475614/
- Twitter: @thangchung

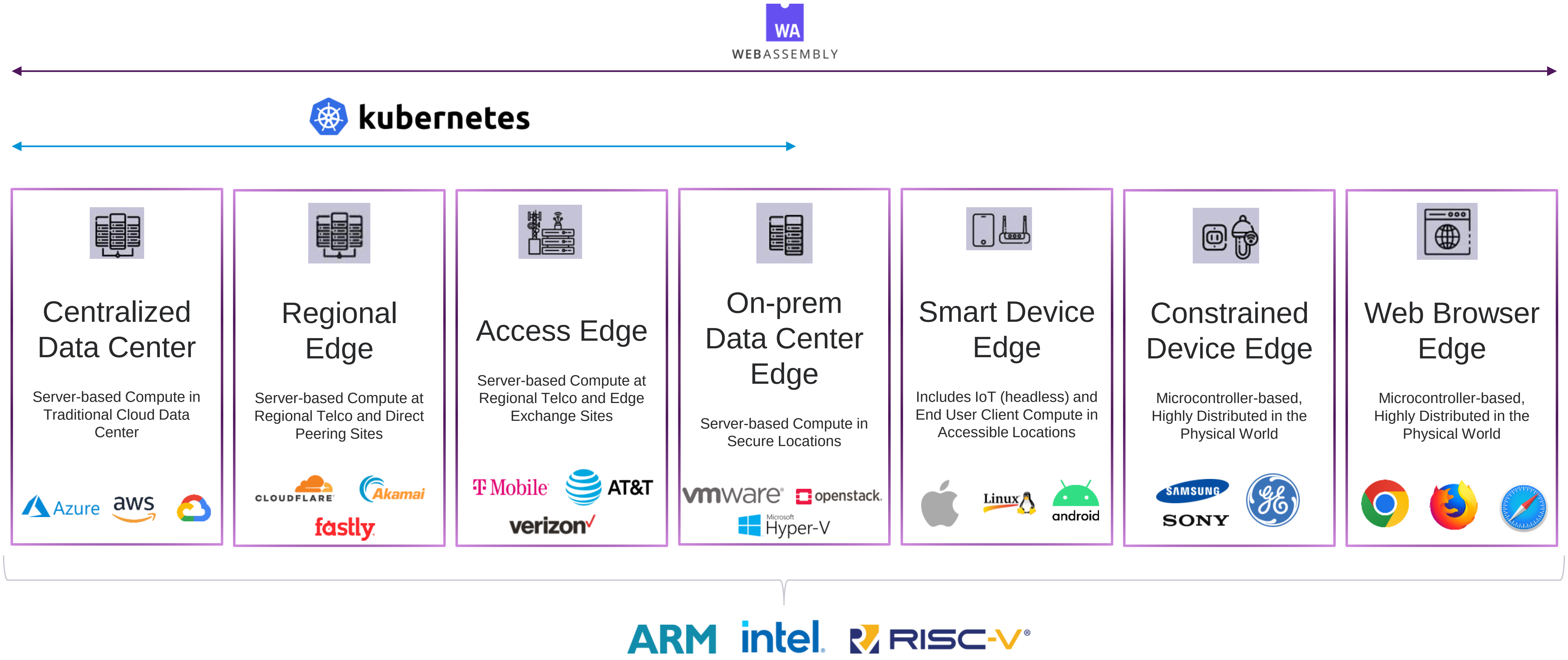
Agenda

1. **WebAssembly (WASM) / WebAssembly System Interface (WASI): What / Why?**
2. **WASM / WASI on Kubernetes (k8s)**
   - containerd-wasm-shims (runwasi)
   - Add more capabilities with CNCF other components
3. **Demo: Dapr on k8s and WASM/WASI**
4. **WASM/WASI roadmap**
5. **Q&A**

01

WASM / WASI

Modern Computing – The Status Quo



WEBASSEMBLY

kubernetes

Centralized Data Center

Server-based Compute in Traditional Cloud Data Center

Regional Edge

Server-based Compute at Regional Telco and Direct Peering Sites

Access Edge

Server-based Compute at Regional Telco and Edge Exchange Sites

On-prem Data Center Edge

Server-based Compute in Secure Locations

Smart Device Edge

Includes IoT (headless) and End User Client Compute in Accessible Locations

Constrained Device Edge

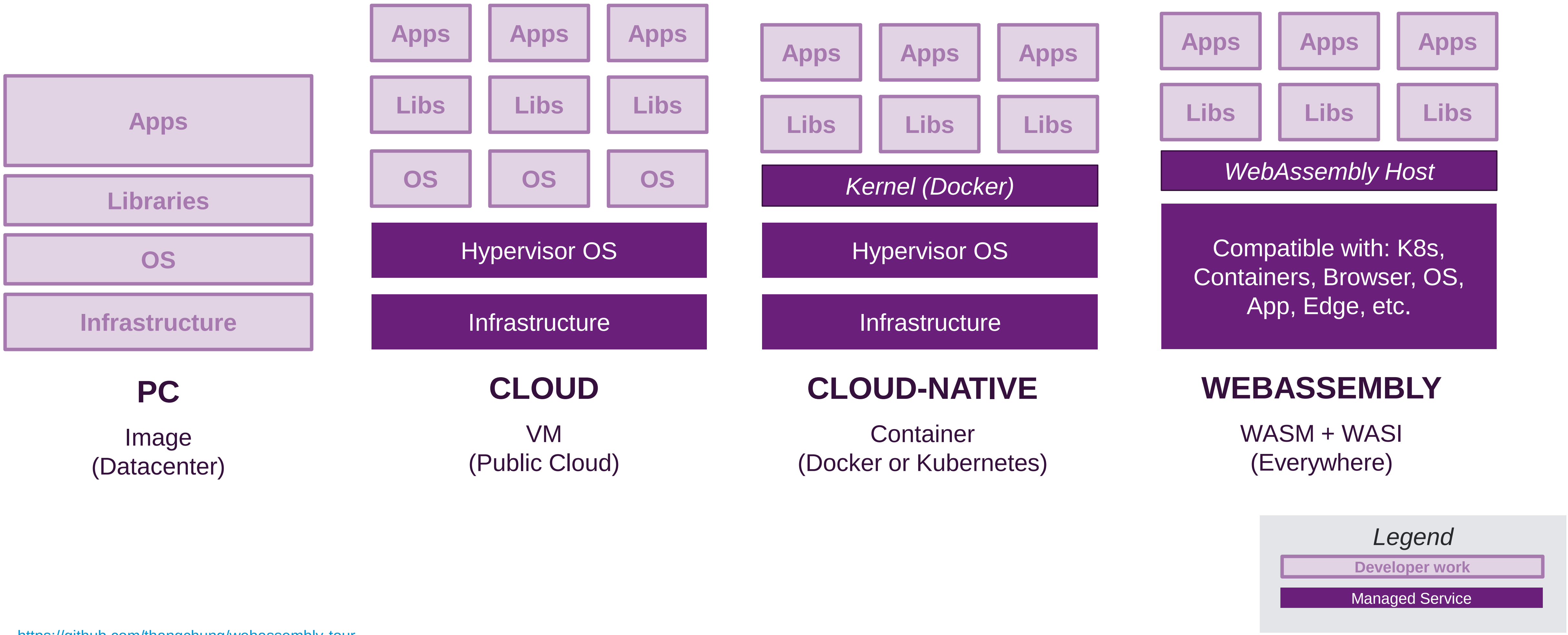
Microcontroller-based, Highly Distributed in the Physical World

Web Browser Edge

Microcontroller-based, Highly Distributed in the Physical World

https://cosmonic.com/blog/industry/webassembly-at-the-edge

Modern Computing - A Path to Abstraction



Apps

Libraries

OS

Infrastructure

PC

Image
(Datacenter)

Apps | Apps | Apps

Libs | Libs | Libs

OS | OS | OS

Hypervisor OS

Infrastructure

CLOUD

VM
(Public Cloud)

Apps | Apps | Apps

Libs | Libs | Libs

Kernel (Docker)

Hypervisor OS

Infrastructure

CLOUD-NATIVE

Container
(Docker or Kubernetes)

Apps | Apps | Apps

Libs | Libs | Libs

WebAssembly Host

Compatible with: K8s, Containers, Browser, OS, App, Edge, etc.

WEBASSEMBLY

WASM + WASI
(Everywhere)

Legend

Developer work

Managed Service

https://github.com/thangchung/webassembly-tour

WebAssembly (WASM)

- WebAssembly (today): it’s **neither web**, **not assembly**.
- It is a specification of **a binary instruction format**, designed as a **portable compilation target**.

```
0061 736d                    ; WASM_BINARY_MAGIC
0100 0000                    ; WASM_BINARY_VERSION
01                        ; section code
00                        ; section size
01                        ; num types
60                        ; func
02                        ; num params
7f                       ; i32
7f                       ; i32
01                        ; num results
7f                       ; i32
07                        ; FIXUP section size
03                        ; section code
00                        ; section size (guess)
01                        ; num functions
00                        ; function 0 signature index
02                        ; FIXUP section size
07                        ; section code
00                        ; section size (guess)
01                        ; num exports
03                        ; string length
6164 64                     ; export name "add"
00                        ; export kind
00                        ; export func index
07                        ; FIXUP section size
0a                        ; section code
00                        ; section size
01                        ; num functions
00                        ; func body size
00                        ; local decl count
20                        ; local.get
00                        ; local index
20                        ; local.get
01                        ; local index
6a                        ; i32.add
0b                        ; end
07                        ; FIXUP func body size
09                        ; FIXUP section size
```

WebAssembly Binary Format (.wasm)*

```
(module
  (func $add (param $lhs i32) (param $rhs i32) (result i32)
    local.get $lhs
    local.get $rhs
    i32.add)
  (export "add" (func $add))
)
```

WebAssembly Text Format (.wat)*

WebAssembly System Interface (WASI)

- A WASM native API ecosystem
- POSIX like interface to enable existing applications to target a conceptual OS
- Capability-based, e.g., files, sockets, clocks, random numbers, and more
- cargo build --target wasm32-wasi

Portable

Independent of OS and processor architecture

Secure

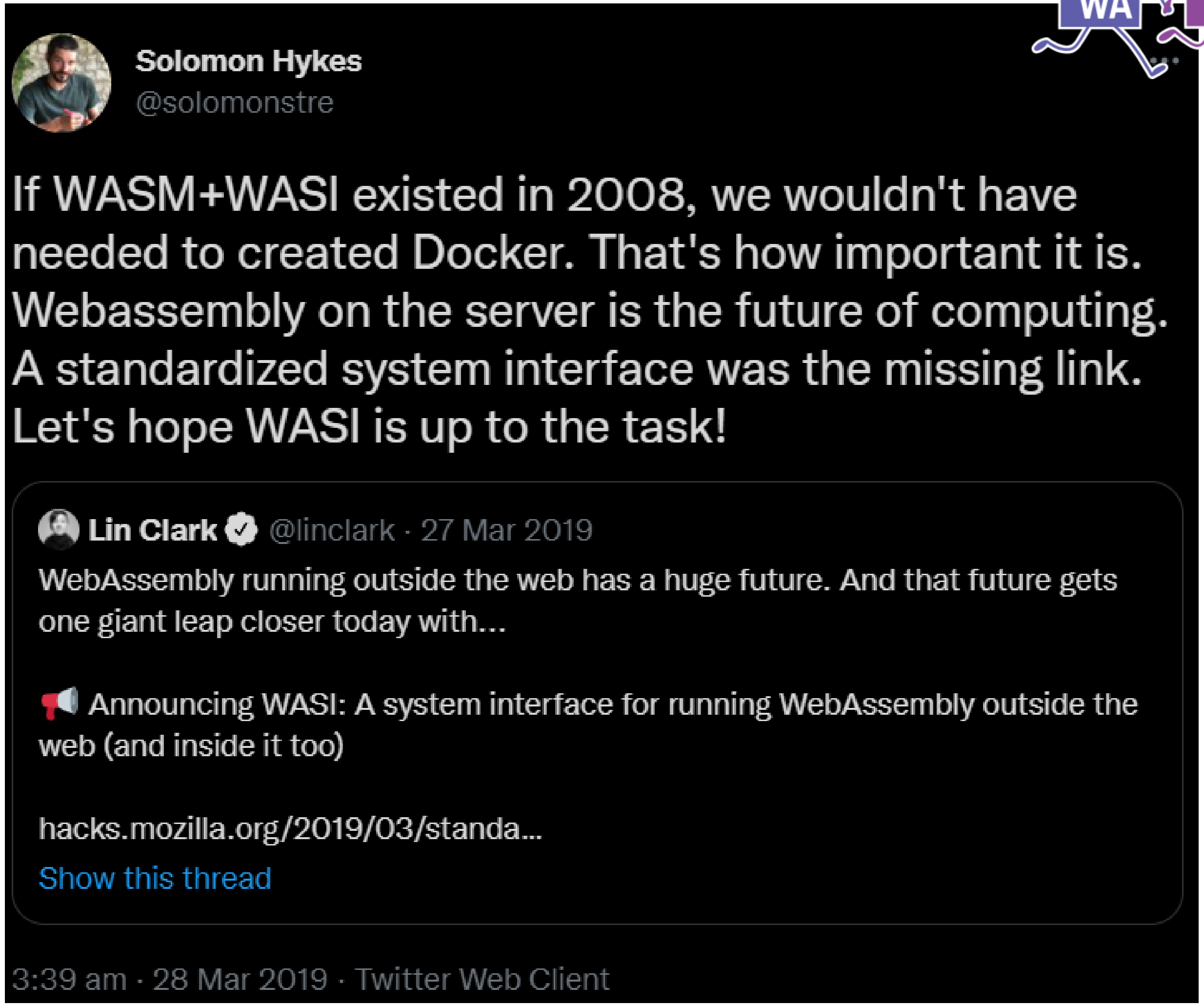
Preserve in-browser security model through WASI's capability-based security

Small

Binaries should be small and quick to transfer

Quick

Startup times comparable with natively compiled code



Solomon Hykes
@solomonstre

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

Lin Clark @linclark · 27 Mar 2019

WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

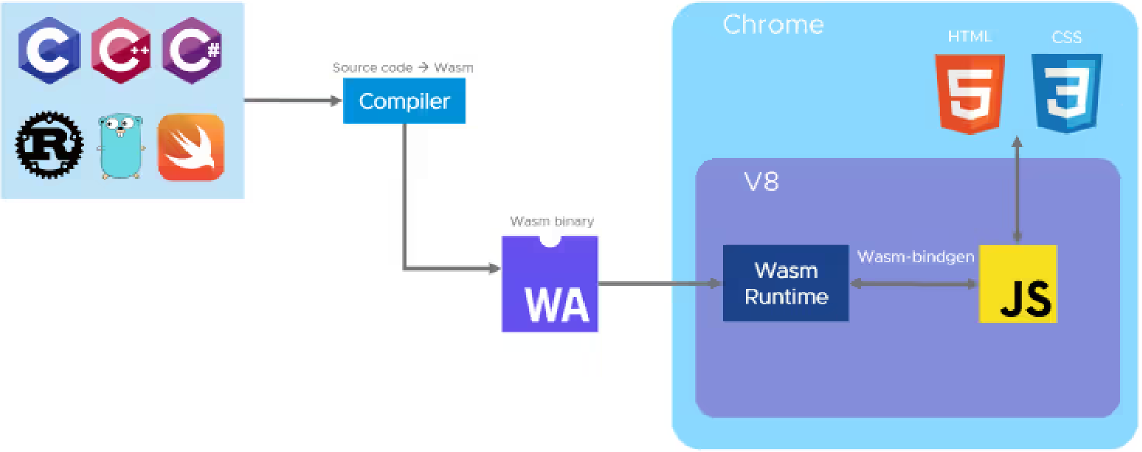
📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

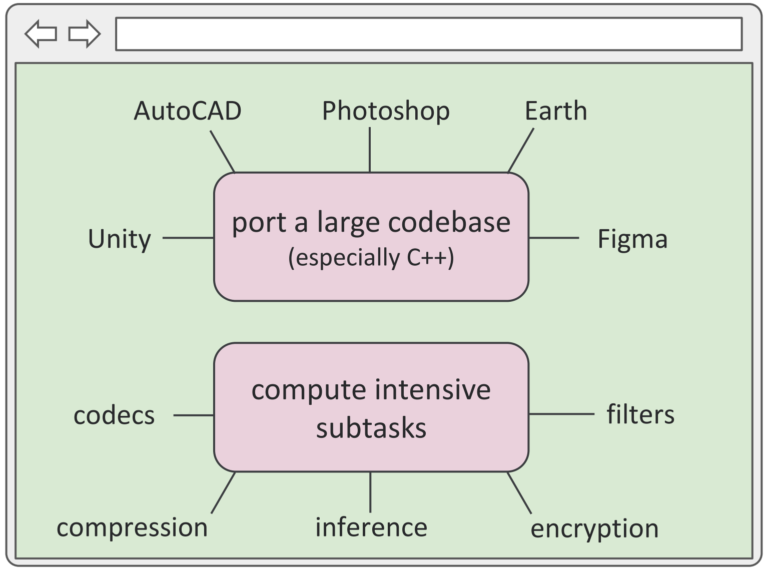
hacks.mozilla.org/2019/03/standa...

Show this thread

3:39 am · 28 Mar 2019 · Twitter Web Client

How does it work on the browser?





AutoCAD

Photoshop

Earth

Unity

port a large codebase
(especially C++)

Figma

codecs

compute intensive subtasks

filters

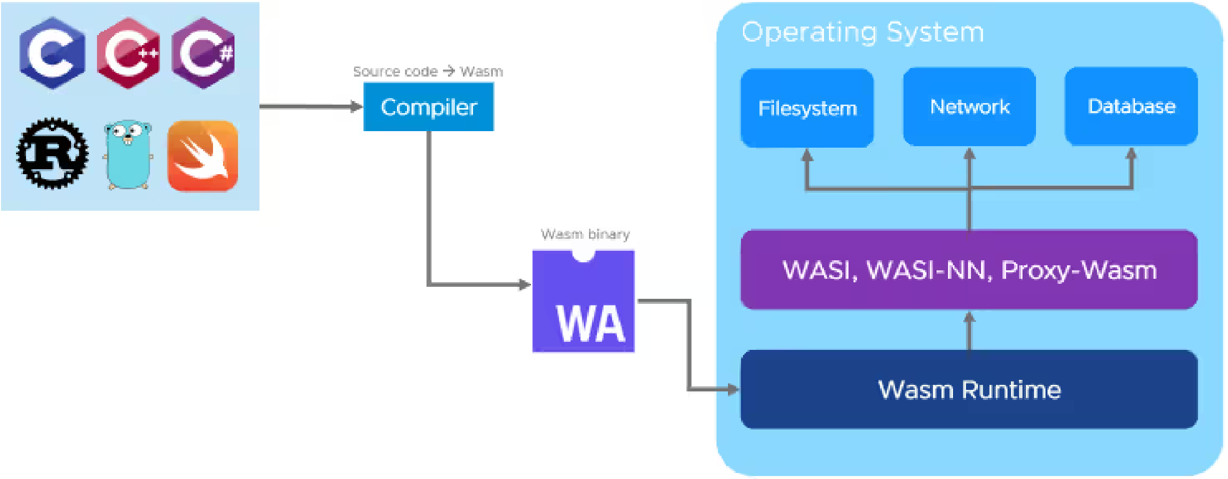
compression

inference

encryption

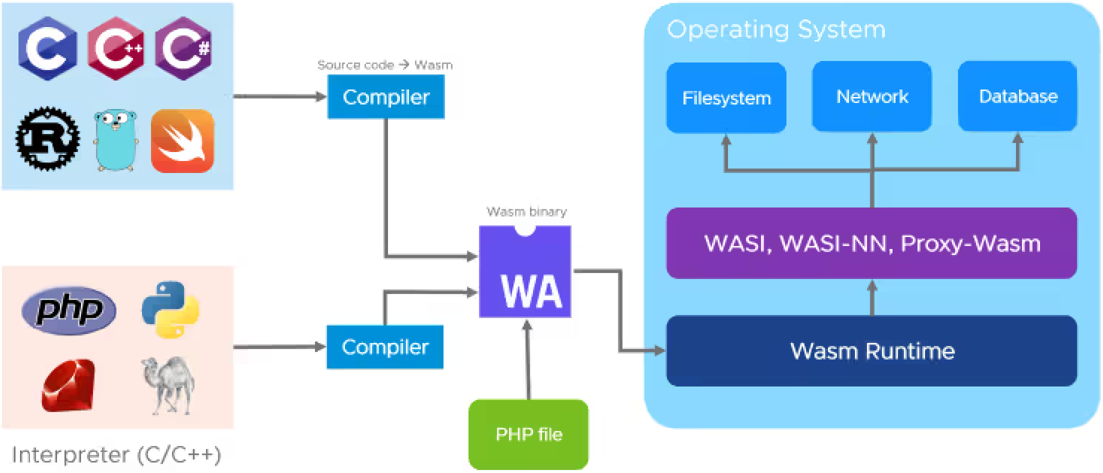
https://wasmlabs.dev/articles/docker-without-containers/

How does it work on the server?



https://wasmlabs.dev/articles/docker-without-containers/

How does it work on the server (interpreted languages)?



Source code → Wasm

Compiler

Compiler

Interpreter (C/C++)

Wasm binary

WA

PHP file

Operating System

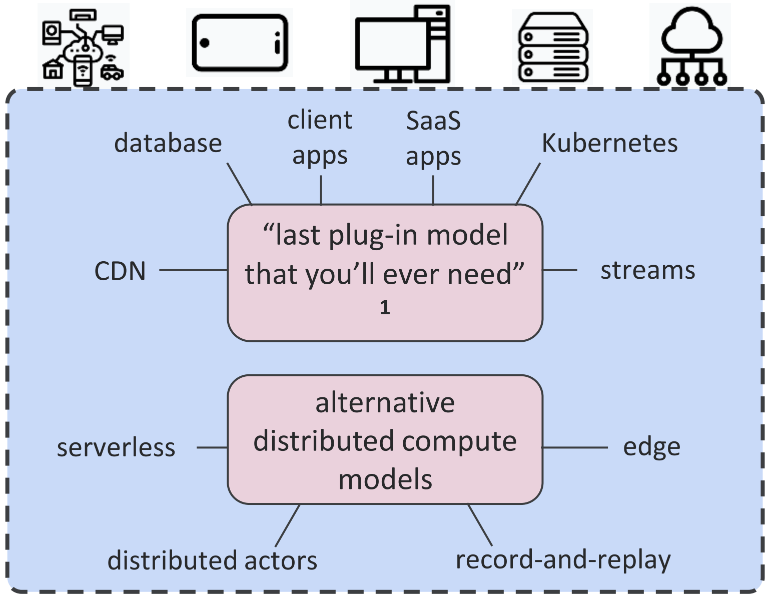
Filesystem

Network

Database

WASI, WASI-NN, Proxy-Wasm

Wasm Runtime



database

client apps

SaaS apps

Kubernetes

CDN

“last plug-in model that you’ll ever need”[1]

streams

serverless

alternative distributed compute models

edge

distributed actors

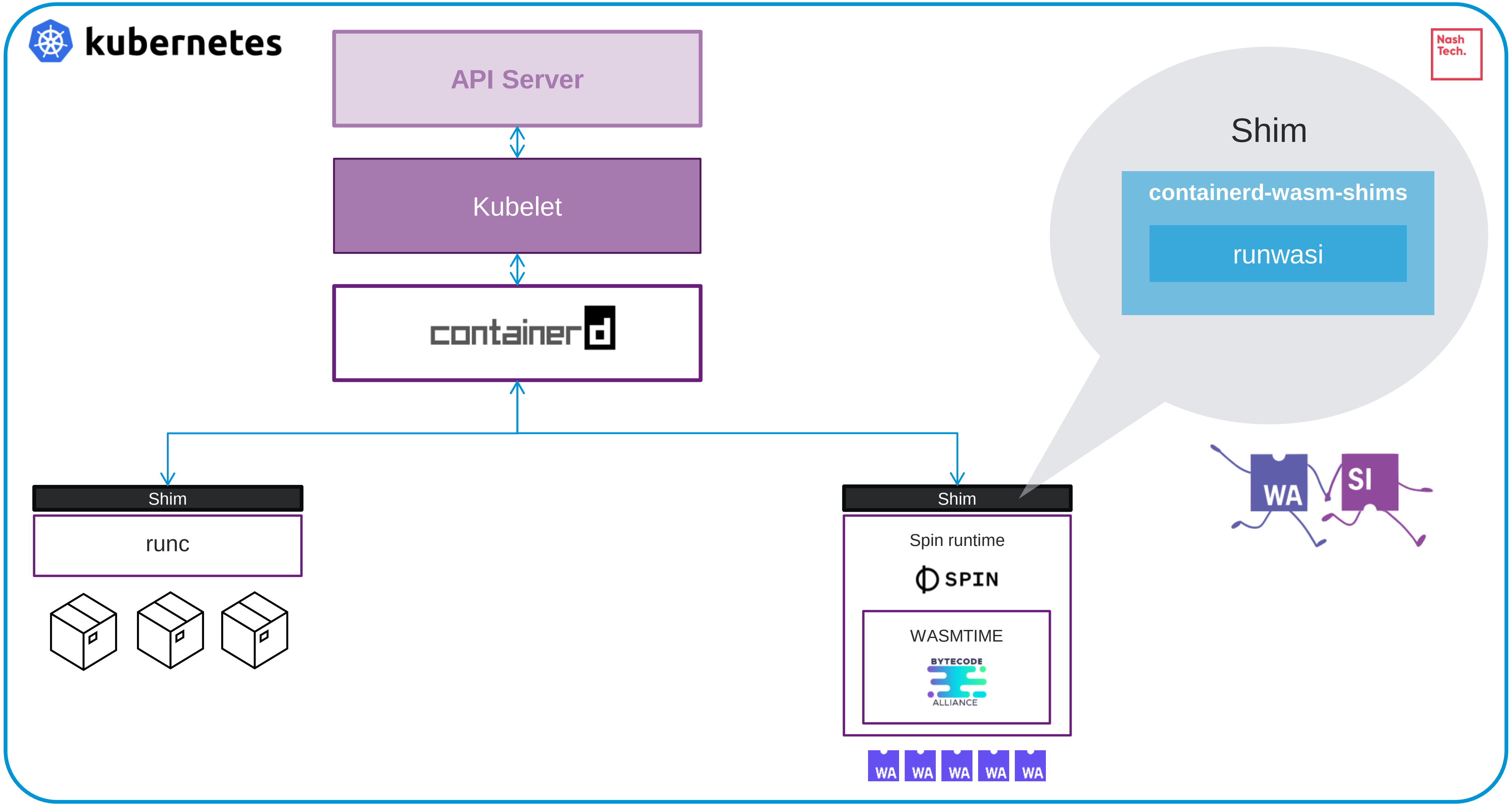
record-and-replay

https://wasmlabs.dev/articles/docker-without-containers/

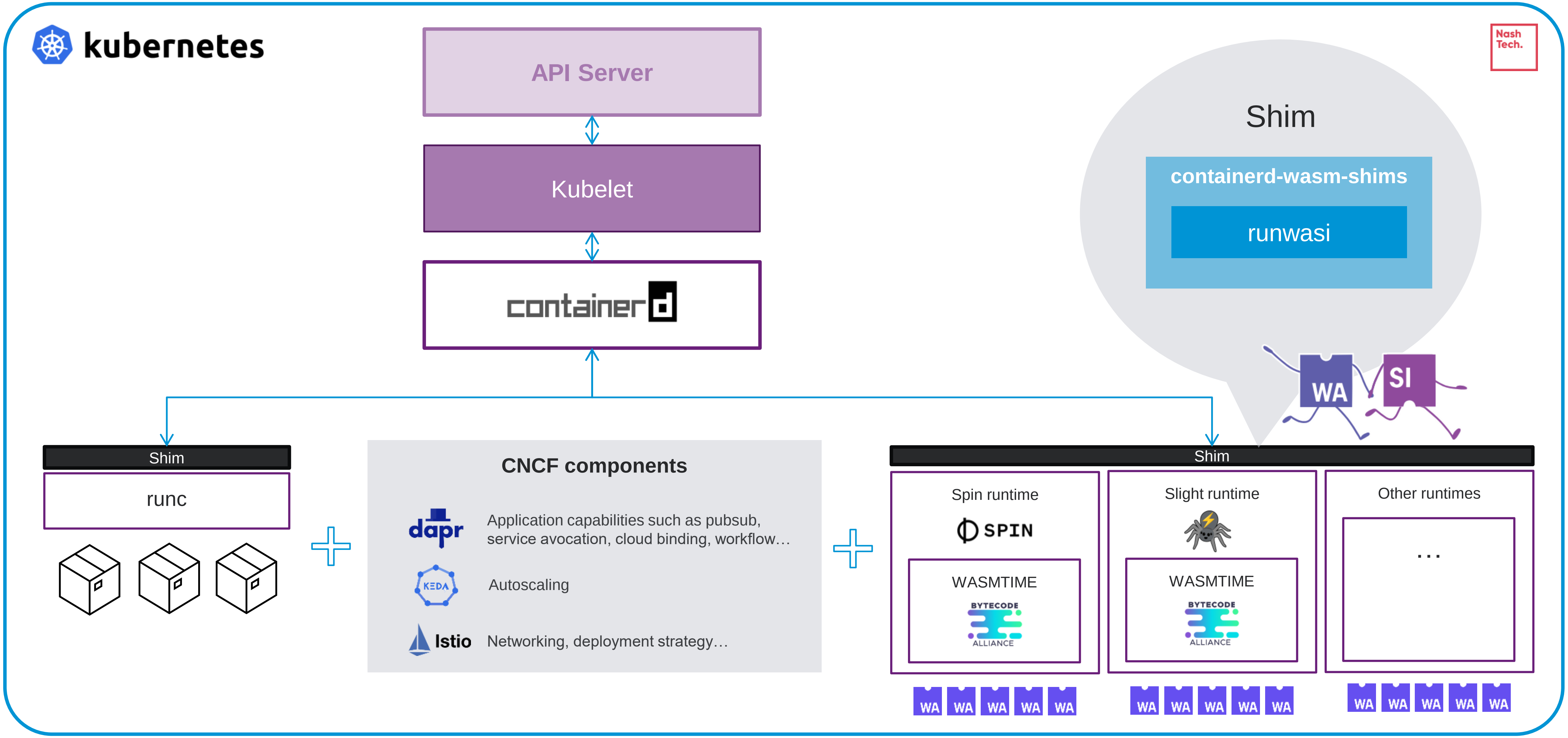
02

WASM/WASI on Kubernetes

How does it work with current containerd?



How can we leverage CNCF other components?



kubernetes

API Server

Kubelet

containerd

Shim

containerd-wasm-shims

runwasi

Shim

runc

CNCF components

dapr — Application capabilities such as pubsub, service avocation, cloud binding, workflow…

KEDA — Autoscaling

Istio — Networking, deployment strategy…

Shim

Spin runtime

SPIN

WASMTIME

BYTECODE ALLIANCE

Slight runtime

WASMTIME

BYTECODE ALLIANCE

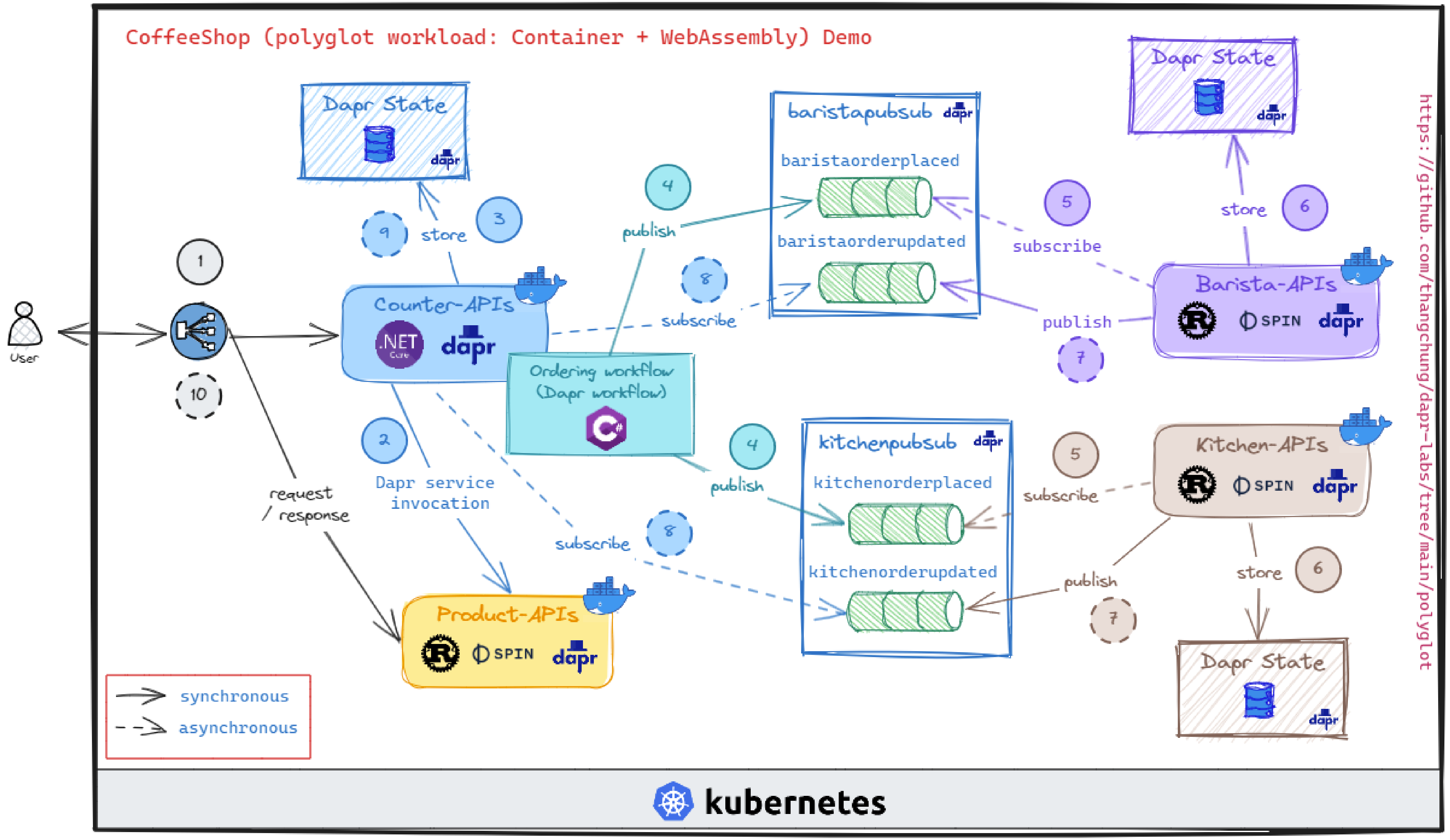
Other runtimes

…

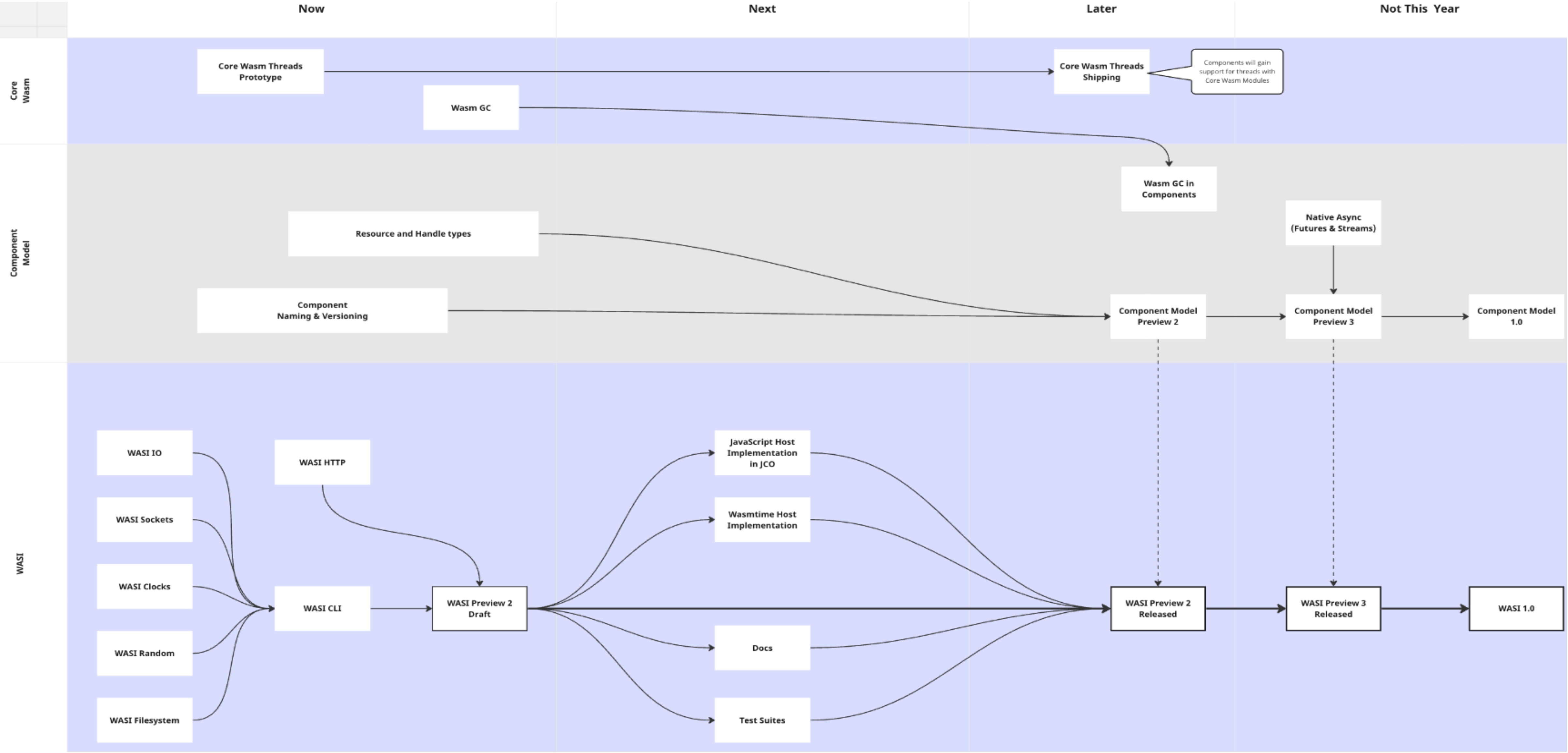
03

DEMO: Dapr on k8s and WASM/WASI

CoffeeShop (polyglot workload: Container + WebAssembly) Demo



Appendix: WASM / WASI Roadmap



Now

Next

Later

Not This Year

Core Wasm

Core Wasm Threads Prototype

Wasm GC

Core Wasm Threads Shipping

Components will gain support for threads with Core Wasm Modules

Component Model

Wasm GC in Components

Resource and Handle types

Native Async (Futures & Streams)

Component Naming & Versioning

Component Model Preview 2

Component Model Preview 3

Component Model 1.0

WASI

WASI IO

WASI Sockets

WASI Clocks

WASI Random

WASI Filesystem

WASI HTTP

WASI CLI

WASI Preview 2 Draft

JavaScript Host Implementation in JCO

Wasmtime Host Implementation

Docs

Test Suites

WASI Preview 2 Released

WASI Preview 3 Released

WASI 1.0

https://bytecodealliance.org/articles/webassembly-the-updated-roadmap-for-developers

References

- https://github.com/bytecodealliance
- https://www.fermyon.com/blog/spin-in-docker
- https://github.com/containerd/runwasi
- https://github.com/deislabs/containerd-wasm-shims
- https://wasmlabs.dev/articles/docker-without-containers/
- https://bytecodealliance.org/articles/webassembly-the-updated-roadmap-for-developers
- https://cosmonic.com/blog/industry/webassembly-at-the-edge
- https://www.webassembly.guide
- https://github.com/thangchung/webassembly-tour
- DEMO: https://github.com/thangchung/dapr-labs/tree/main/polyglot

Thank you

Nash
Tech.