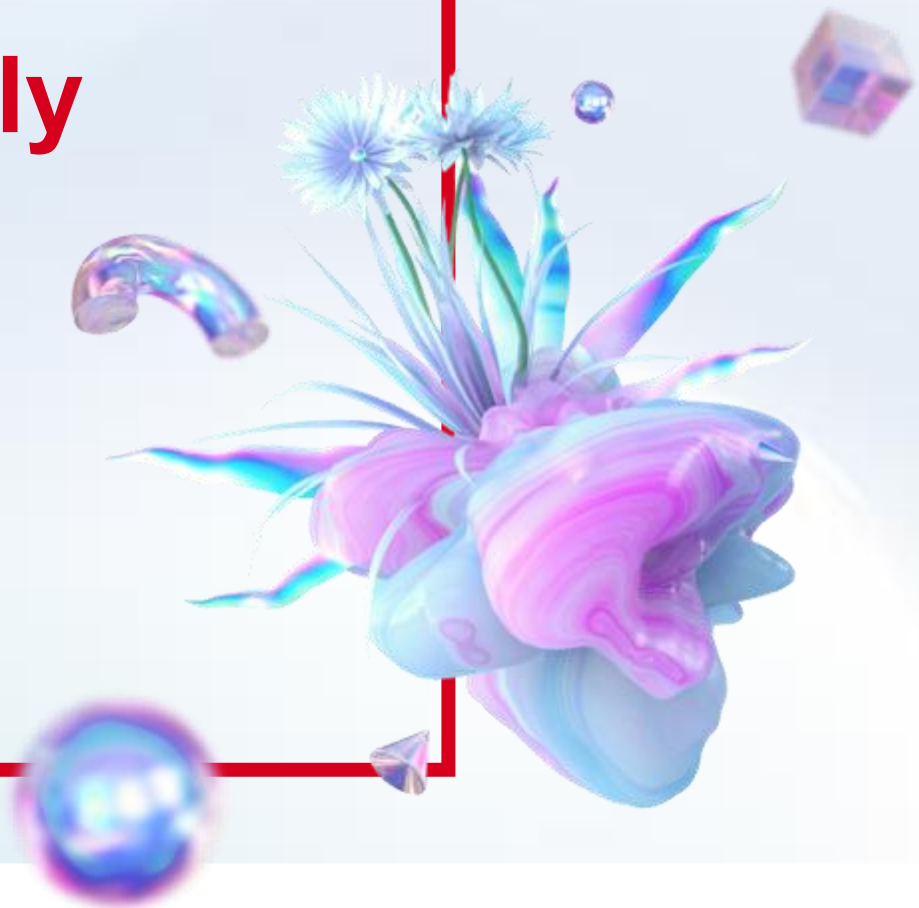
Building WebAssembly Application on Azure Kubernetes Services (AKS)



Thang Chung – NashTech

April 2024

Thang Chung

Technical Manager, NashTech VN
Microsoft Azure MVP

- Creator of Vietnam Microservices Group on Facebook (>19k members).
  - https://www.facebook.com/groups/645391349250568
- Experience: >17 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/
- X (former Twitter): @thangchung

https://wasm.ee/

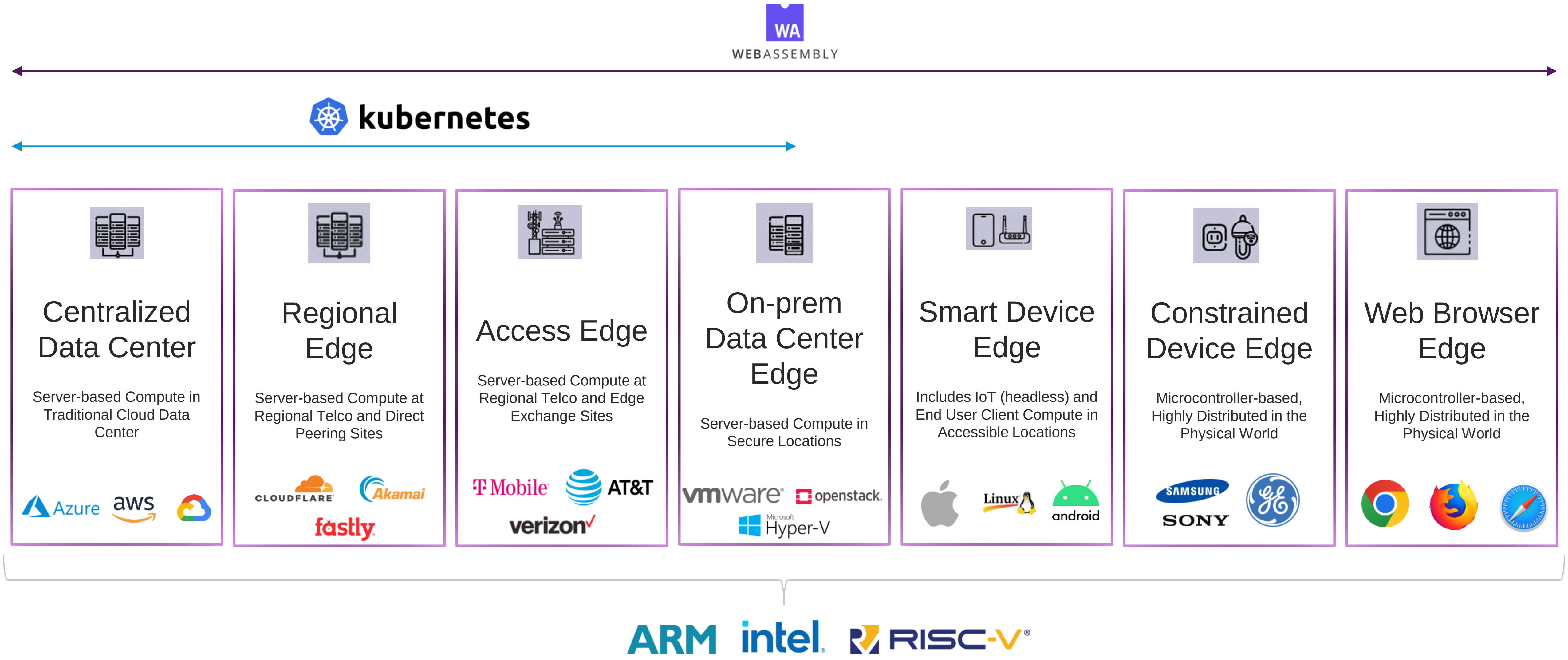
Agenda

1. **WebAssembly (WASM) / WebAssembly System Interface (WASI): Why / What?**
2. **WASM / WASI on Kubernetes (k8s)**
   - containerd-wasm-shims (runwasi)
   - kwasm
   - Add more capabilities with CNCF other components
3. **Demo: Build and Run WebAssembly App on AKS with SpinKube**

WASM / WASI

01

Modern Computing – The Status Quo



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

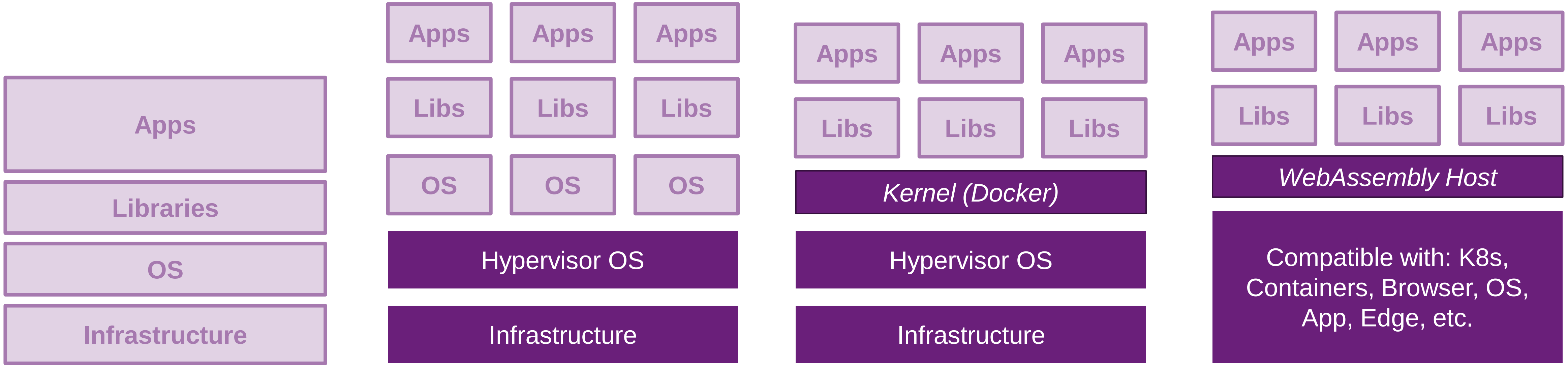
Microcontroller-based, Highly Distributed in the Physical World

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

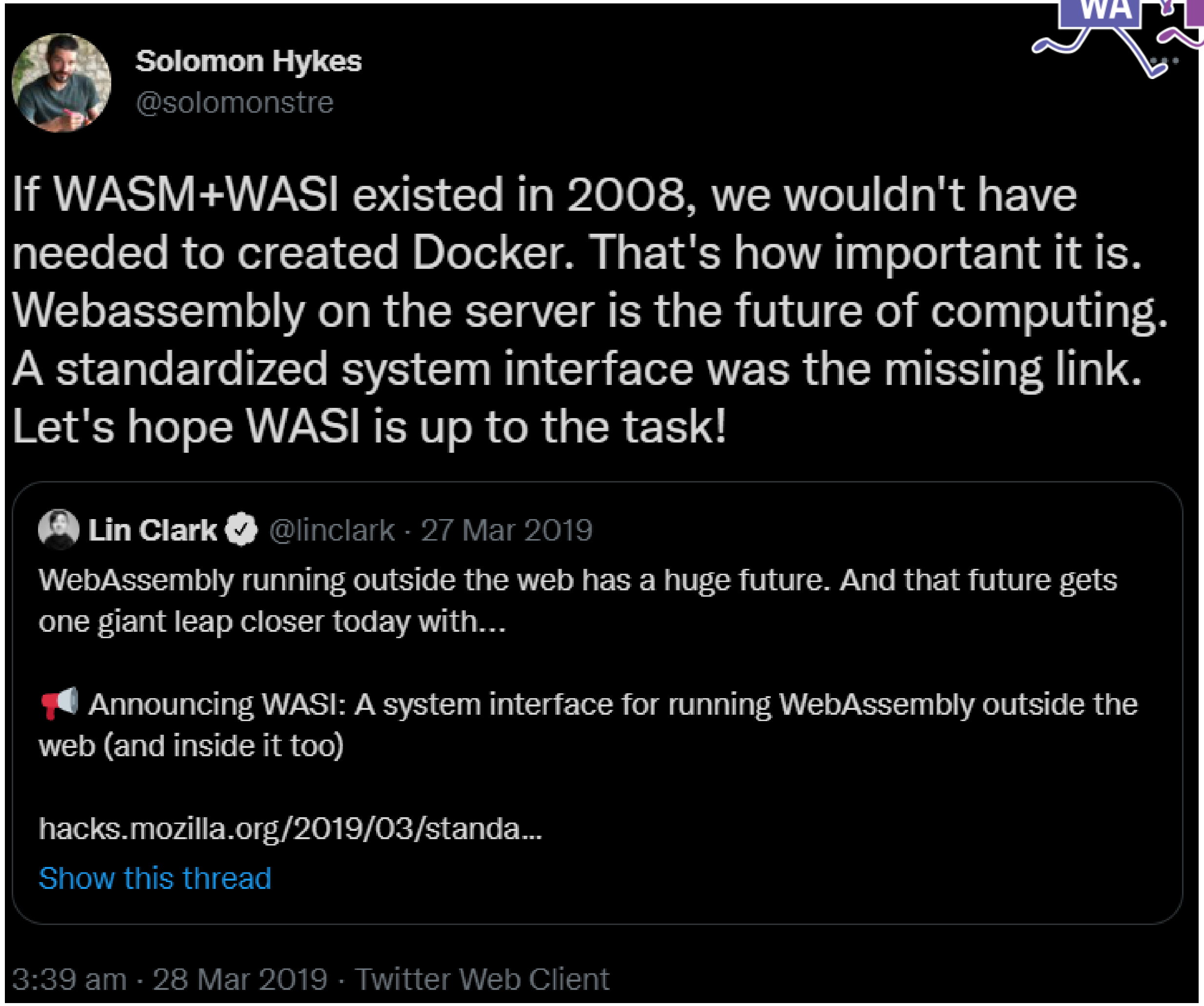
Preserve in-browser security model through WASI's capability-based security

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...

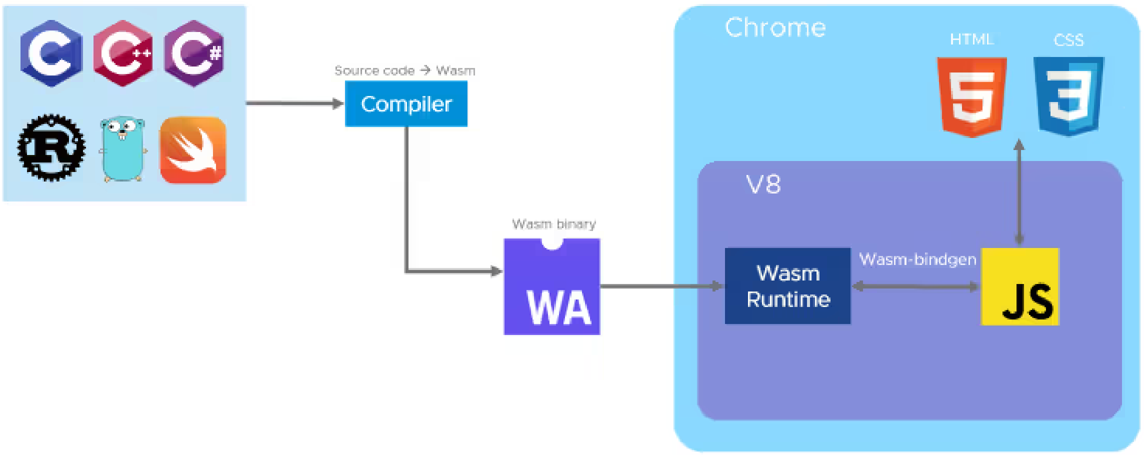
📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

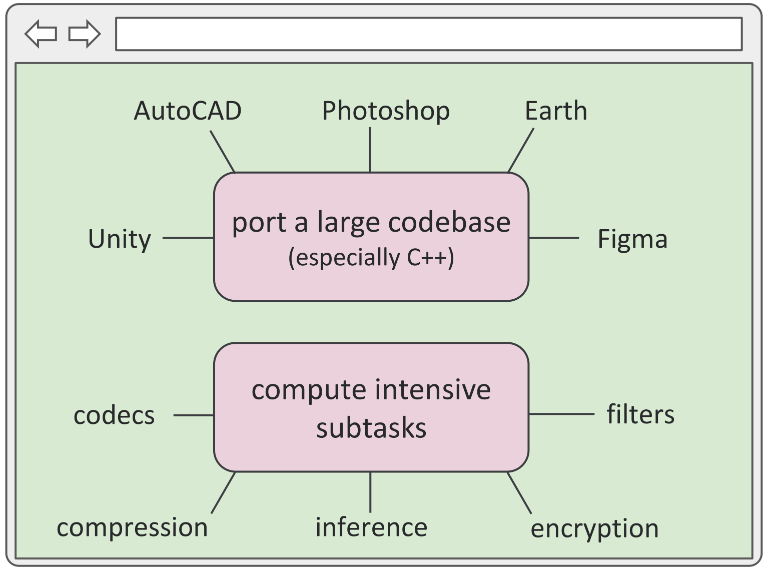
hacks.mozilla.org/2019/03/standa...

Show this thread

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

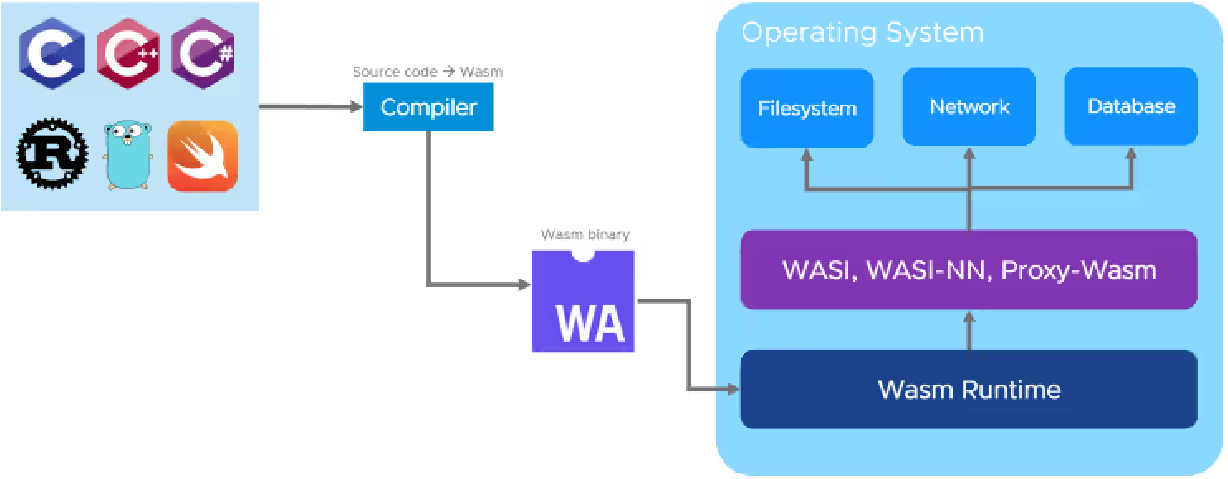
How does it work on the BROWSER?




AutoCAD
Photoshop
Earth
Unity
port a large codebase
(especially C++)
Figma
codecs
compute intensive subtasks
filters
compression
inference
encryption

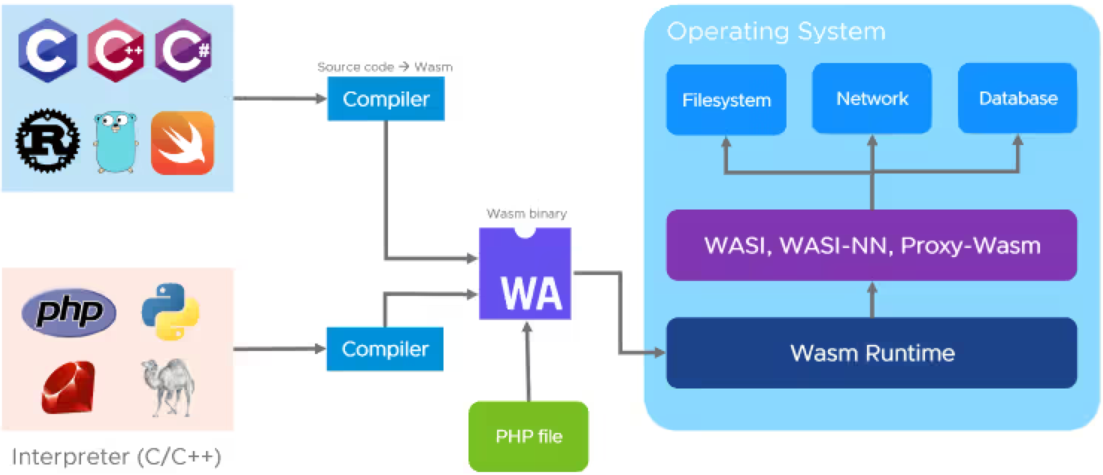
https://wasmlabs.dev/articles/docker-without-containers/

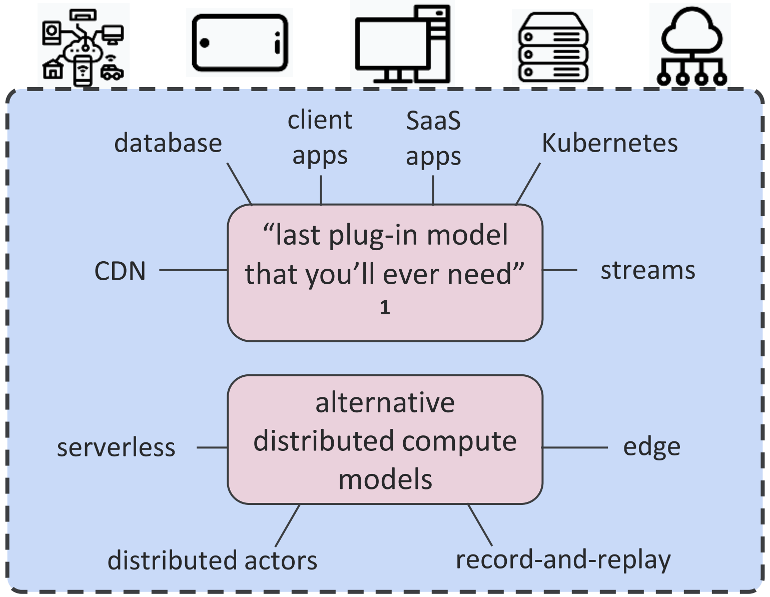
How does it work on the SERVER? (1/2)



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

How does it work on the SERVER (interpreted languages)? (2/2)



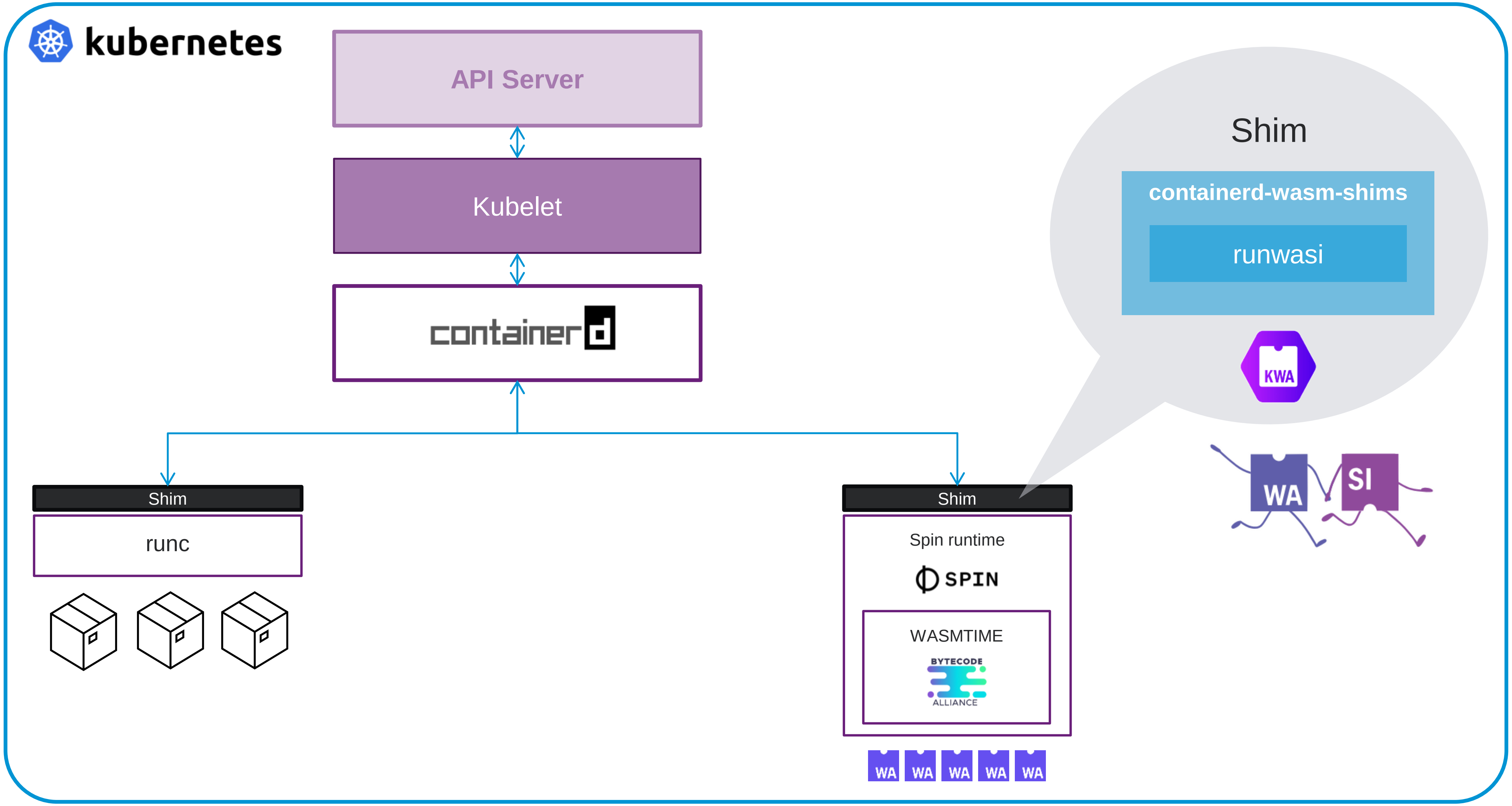


https://wasmlabs.dev/articles/docker-without-containers/

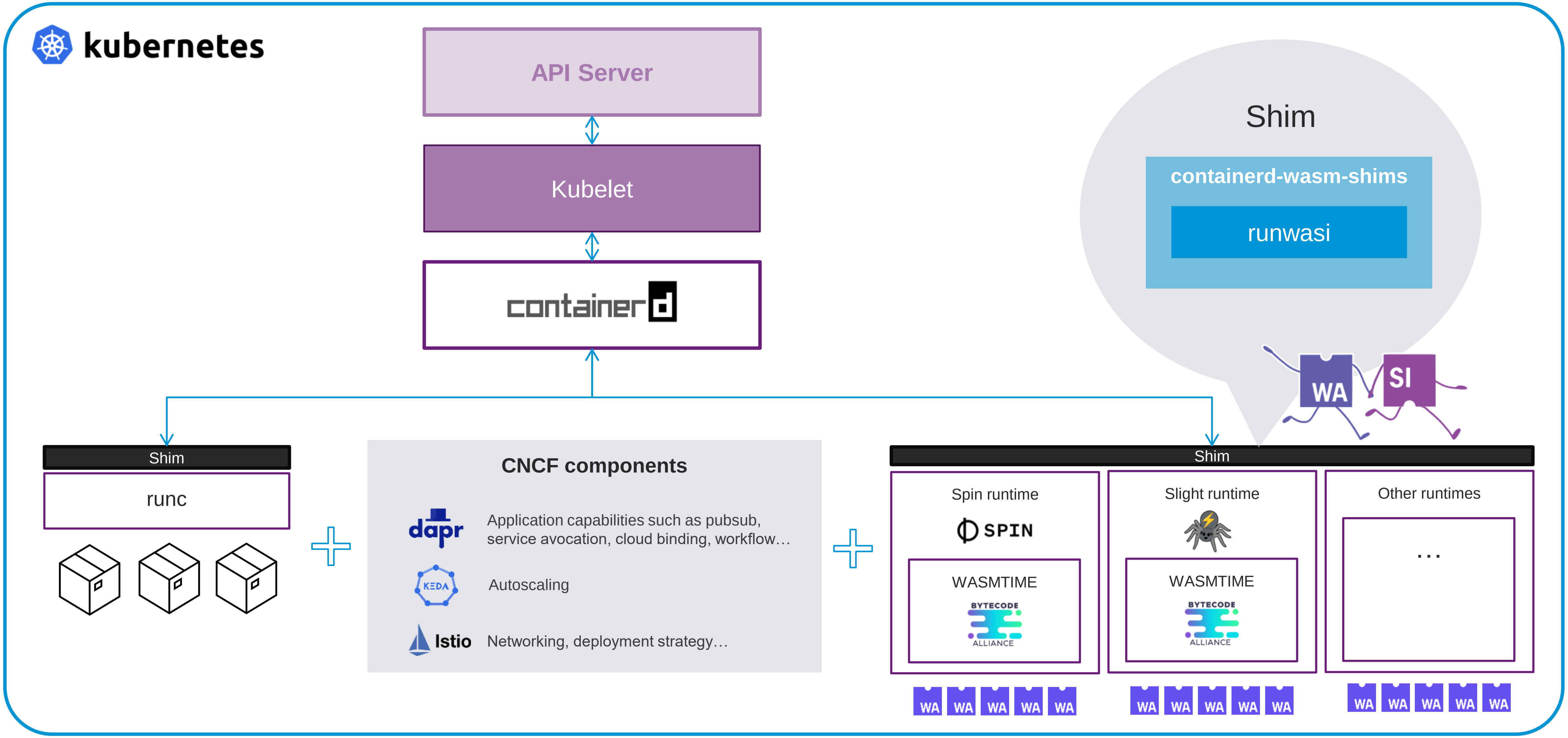
02

WASM/WASI on Kubernetes

How does it work with current Containerd?



How can we leverage CNCF other components?



API Server

Kubelet

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

WASMTIME

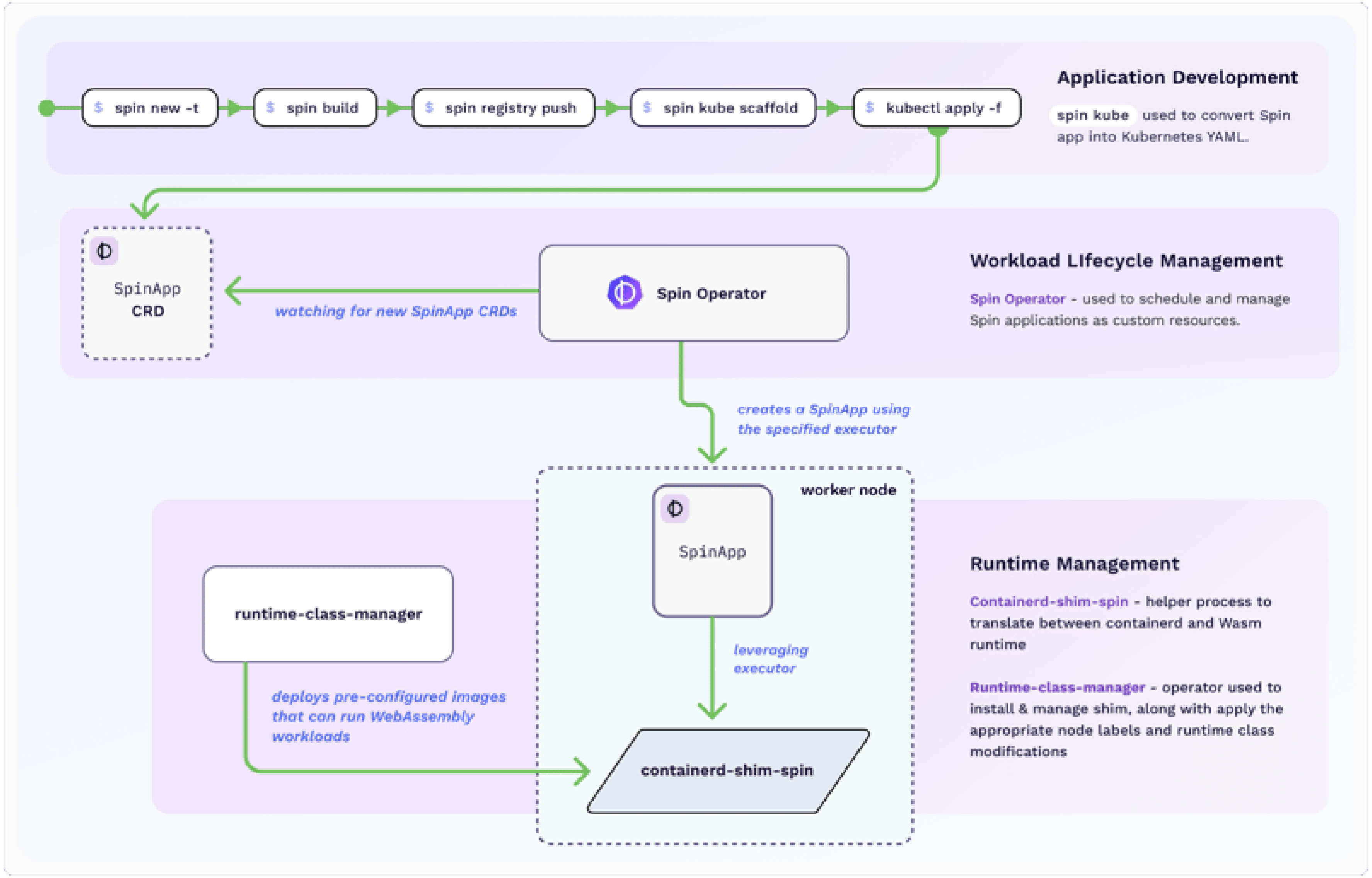
Slight runtime

WASMTIME

Other runtimes

…

SpinKube Architecture



```
$ spin new -t
$ spin build
$ spin registry push
$ spin kube scaffold
$ kubectl apply -f
```

Application Development

spin kube used to convert Spin app into Kubernetes YAML.

Workload LIfecycle Management

Spin Operator - used to schedule and manage Spin applications as custom resources.

Runtime Management

Containerd-shim-spin - helper process to translate between containerd and Wasm runtime

Runtime-class-manager - operator used to install & manage shim, along with apply the appropriate node labels and runtime class modifications

SpinApp
CRD
watching for new SpinApp CRDs
Spin Operator
creates a SpinApp using the specified executor
worker node
SpinApp
leveraging executor
runtime-class-manager
deploys pre-configured images that can run WebAssembly workloads
containerd-shim-spin

Comparing Serverless Units

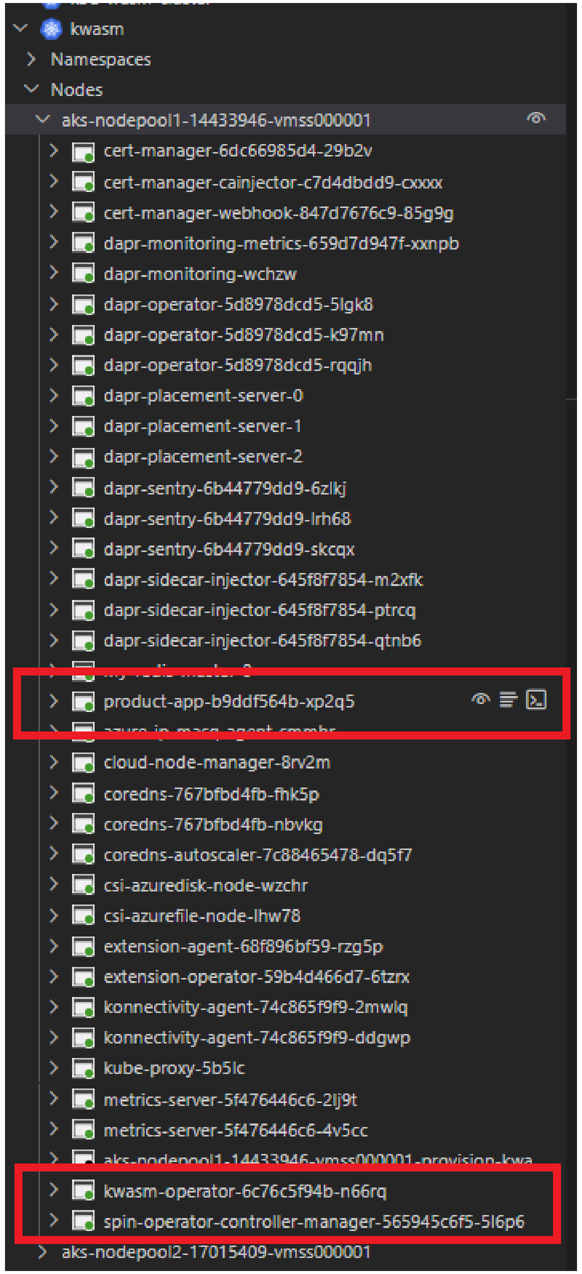
Characteristic	MicroVM	WebAssembly
Isolation	Sandboxed (via Firecracker VMM + KVM)	Sandboxed (via runtime with capabilities-based security and linear memory)
Overhead and Density	1000s per node (48 core, 382 GB RAM, 3360 GB disk)	1000s per node (8 core, 32 GB RAM, 100 GB disk)
Performance	Near native	Near native
Fast Switching	125 ms (startup to clean up)	< 1ms (startup to cleanup)
Soft Allocation	Run in production with oversubscription ratios as high as 10x	Untested
Compatibility	Linux + KVM only. Most software is compatible unless specific hardware requirements	OS and platform-agnostic bytecode. Only supports libraries that can compile to Wasm.

Firecracker: Lightweight Virtualization for Serverless Applications https://www.usenix.org/system/files/nsdi20-paper-agache.pdf

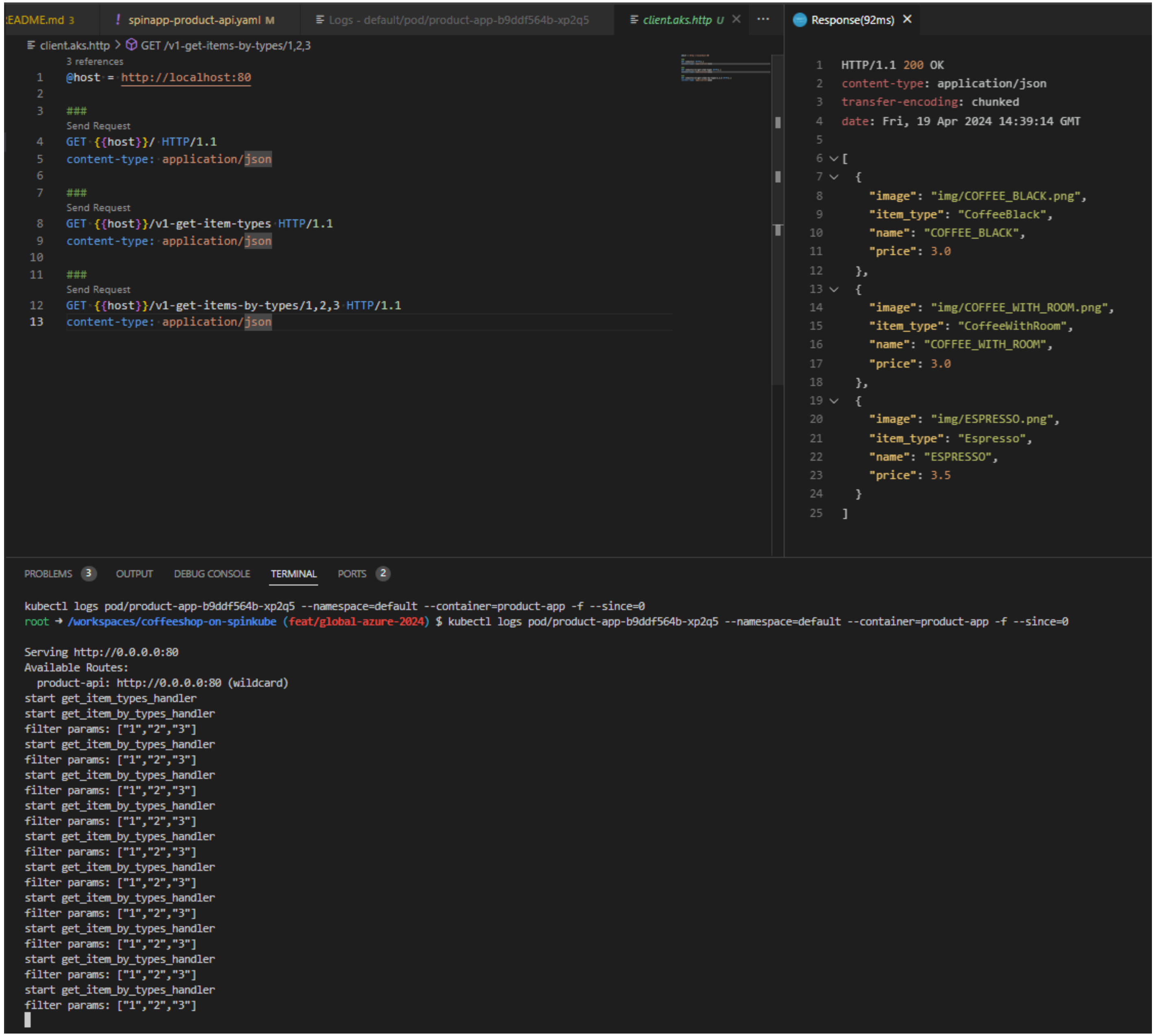
03

DEMO: SpinKube on AKS

https://dev.to/thangchung/how-to-build-and-run-spinapp-on-azure-kubernetes-services-aks-with-spinkube-in-3-steps-1212



```
kwasm
  Namespaces
  Nodes
    aks-nodepool1-14433946-vmss000001
      cert-manager-6dc66985d4-29b2v
      cert-manager-cainjector-c7d4dbdd9-cxxxx
      cert-manager-webhook-847d7676c9-85g9g
      dapr-monitoring-metrics-659d7d947f-xxnpb
      dapr-monitoring-wchzw
      dapr-operator-5d8978dcd5-5lgk8
      dapr-operator-5d8978dcd5-k97mn
      dapr-operator-5d8978dcd5-rqqjh
      dapr-placement-server-0
      dapr-placement-server-1
      dapr-placement-server-2
      dapr-sentry-6b44779dd9-6zlkj
      dapr-sentry-6b44779dd9-lrh68
      dapr-sentry-6b44779dd9-skcqx
      dapr-sidecar-injector-645f8f7854-m2xfk
      dapr-sidecar-injector-645f8f7854-ptrcq
      dapr-sidecar-injector-645f8f7854-qtnb6
      my-redis-master-0
      product-app-b9ddf564b-xp2q5
      azure-ip-masq-agent-cmmhr
      cloud-node-manager-8rv2m
      coredns-767bfbd4fb-fhk5p
      coredns-767bfbd4fb-nbvkg
      coredns-autoscaler-7c88465478-dq5f7
      csi-azuredisk-node-wzchr
      csi-azurefile-node-lhw78
      extension-agent-68f896bf59-rzg5p
      extension-operator-59b4d466d7-6tzrx
      konnectivity-agent-74c865f9f9-2mwlq
      konnectivity-agent-74c865f9f9-ddgwp
      kube-proxy-5b5lc
      metrics-server-5f476446c6-2lj9t
      metrics-server-5f476446c6-4v5cc
      aks-nodepool1-14433946-vmss000001-provision-kwa
      kwasm-operator-6c76c5f94b-n66rq
      spin-operator-controller-manager-565945c6f5-5l6p6
    aks-nodepool2-17015409-vmss000001
```



EADME.md 3
spinapp-product-api.yaml M
Logs - default/pod/product-app-b9ddf564b-xp2q5
client.aks.http U
Response(92ms)
client.aks.http > GET /v1-get-items-by-types/1,2,3

```
3 references
@host = http://localhost:80

###
Send Request
GET {{host}}/ HTTP/1.1
content-type: application/json

###
Send Request
GET {{host}}/v1-get-item-types HTTP/1.1
content-type: application/json

###
Send Request
GET {{host}}/v1-get-items-by-types/1,2,3 HTTP/1.1
content-type: application/json
```

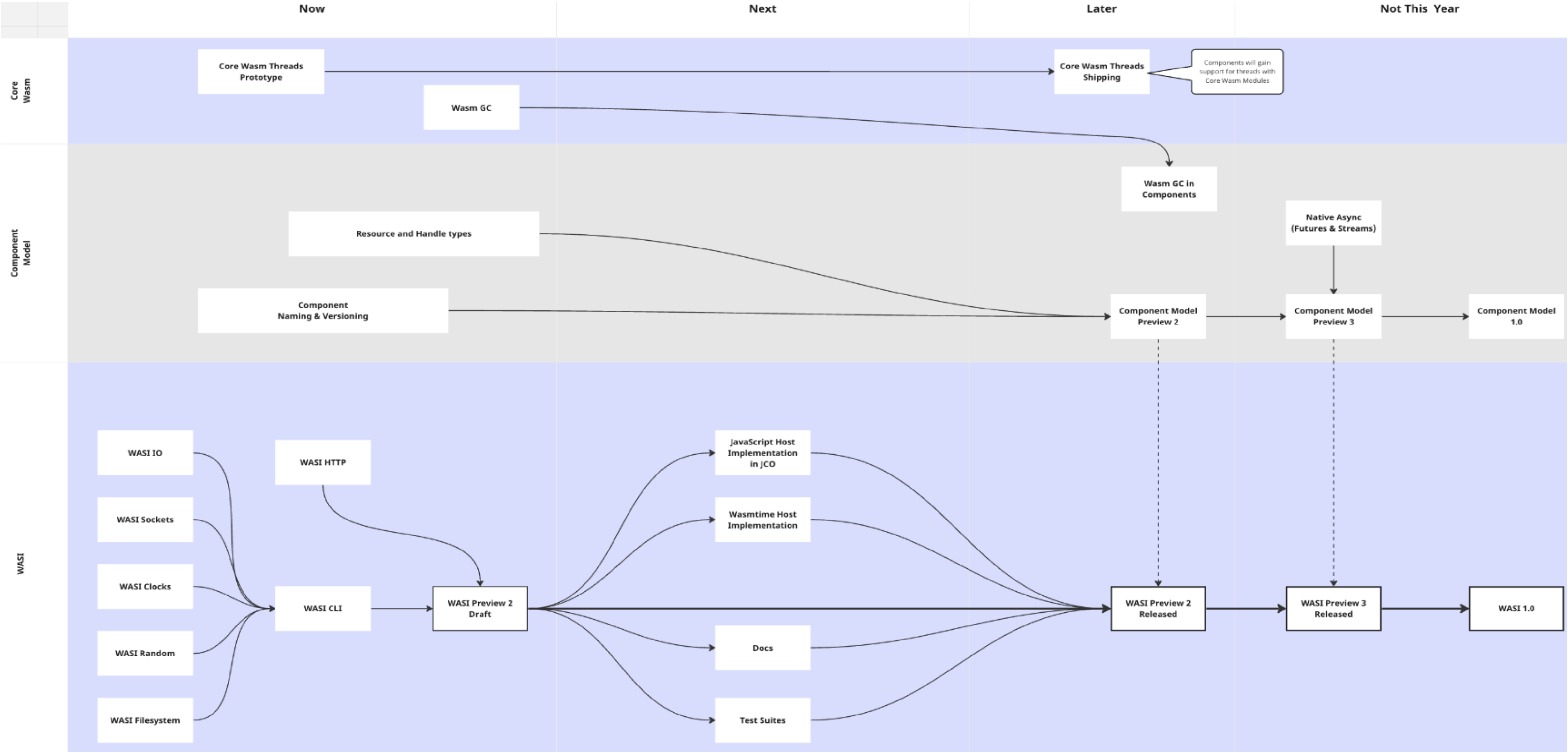
```
HTTP/1.1 200 OK
content-type: application/json
transfer-encoding: chunked
date: Fri, 19 Apr 2024 14:39:14 GMT

[
  {
    "image": "img/COFFEE_BLACK.png",
    "item_type": "CoffeeBlack",
    "name": "COFFEE_BLACK",
    "price": 3.0
  },
  {
    "image": "img/COFFEE_WITH_ROOM.png",
    "item_type": "CoffeeWithRoom",
    "name": "COFFEE_WITH_ROOM",
    "price": 3.0
  },
  {
    "image": "img/ESPRESSO.png",
    "item_type": "Espresso",
    "name": "ESPRESSO",
    "price": 3.5
  }
]
```

```
kubectl logs pod/product-app-b9ddf564b-xp2q5 --namespace=default --container=product-app -f --since=0
root ➜ /workspaces/coffeeshop-on-spinkube (feat/global-azure-2024) $ kubectl logs pod/product-app-b9ddf564b-xp2q5 --namespace=default --container=product-app -f --since=0

Serving http://0.0.0.0:80
Available Routes:
  product-api: http://0.0.0.0:80 (wildcard)
start get_item_types_handler
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
start get_item_by_types_handler
filter params: ["1","2","3"]
```

Appendix: WASM / WASI Roadmap



Now

Next

Later

Not This Year

Core Wasm

Core Wasm Threads Prototype

Core Wasm Threads Shipping

Components will gain support for threads with Core Wasm Modules

Wasm GC

Component Model

Wasm GC in Components

Native Async (Futures & Streams)

Resource and Handle types

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/spinkube
- https://github.com/bytecodealliance
- https://github.com/fermyon
- https://github.com/containerd/runwasi
- https://github.com/deislabs/containerd-wasm-shims
- https://kwasm.sh/
- https://dev.to/thangchung/spinkube-the-first-look-at-webassemblywasi-application-spinapp-on-kubernetes-36jd
- https://dev.to/thangchung/how-to-build-and-run-spinapp-on-azure-kubernetes-services-aks-with-spinkube-in-3-steps-1212
- https://dev.to/thangchung/series/24617
- https://github.com/thangchung/webassembly-tour
- 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
- DEMO: https://github.com/thangchung/coffeeshop-on-spinkube

Thank you

Nash
Tech.