KubeCon | CloudNativeCon

North America 2023

KubeCon | CloudNativeCon
North America 2023

Tutorial: Building Cloud-Native Applications Using WebAssembly and Containers

Melissa Klein, Fermyon
Mikkel Mørk Hegnhøj, Fermyon
Ralph Squillace, Microsoft

Objective and Agenda

First-hand experience with server-side WebAssembly and Kubernetes.

- 10 min. introduction to server-side WebAssembly
- Follow along the tutorial

GitHub repository with all the content you need to complete the tutorial:

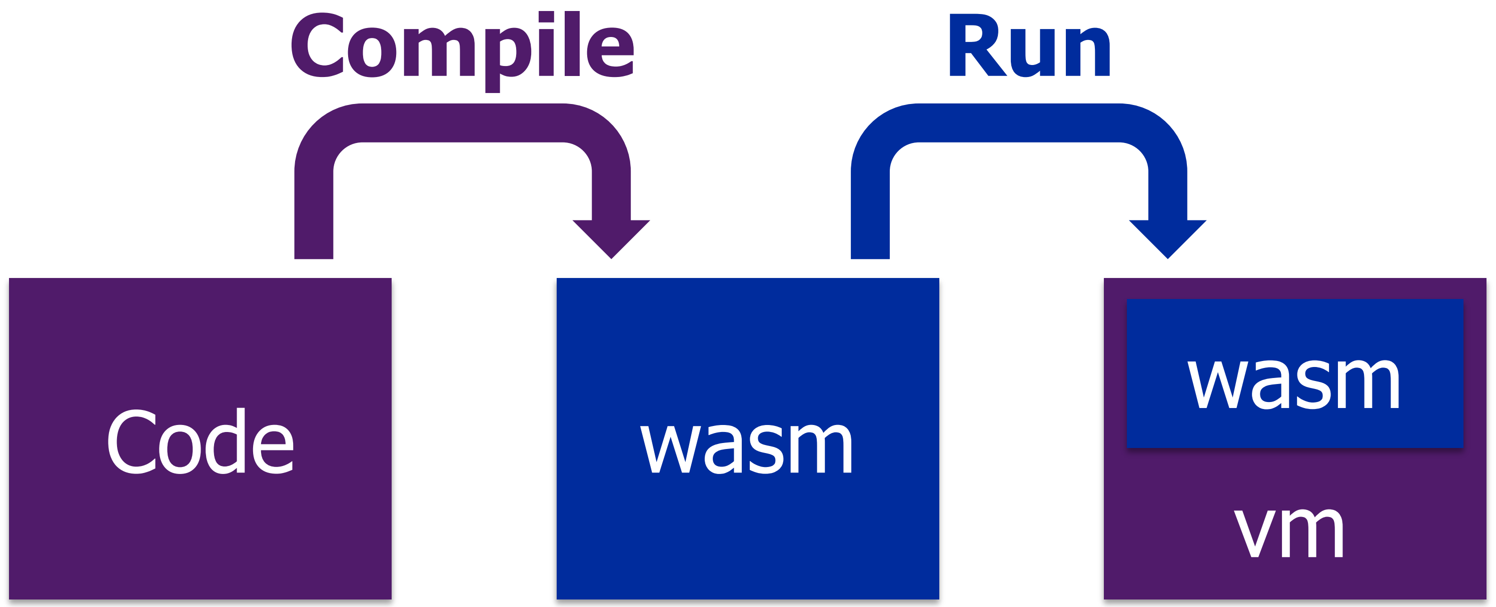
https://github.com/fermyon/workshops/blob/main/wasm-and-containers/

What is WebAssembly?

- It is a specification of a binary instruction format, designed as a portable compilation target
- Originates from the browser, now also available outside
- Language support is emerging and stabilizing
- Wasm is just another name for it

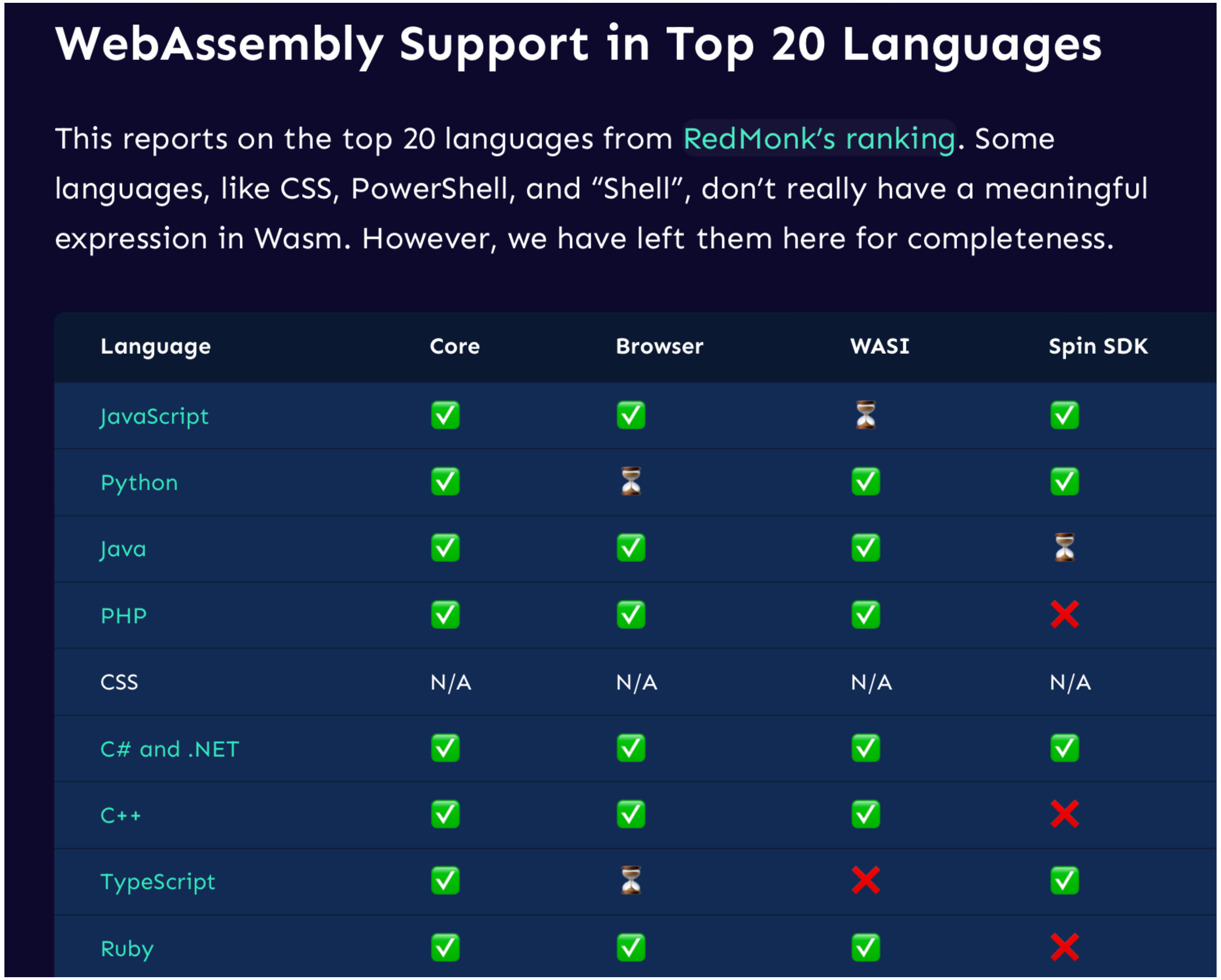
GitHub Repo

Compile and Run



GitHub Repo

WebAssembly 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	✅	✅	✅	❌

GitHub Repo

Runtimes

JavaScript runtimes

Designed to complement and run alongside JavaScript

- V8 (Chromium browsers)
- SpiderMonkey (Firefox)
- Nitro (WebKit)

WASI runtimes

Designed to be independent of browsers

- Wasmtime
- WasmEdge
- Node.js and Bun (experimental)

GitHub Repo

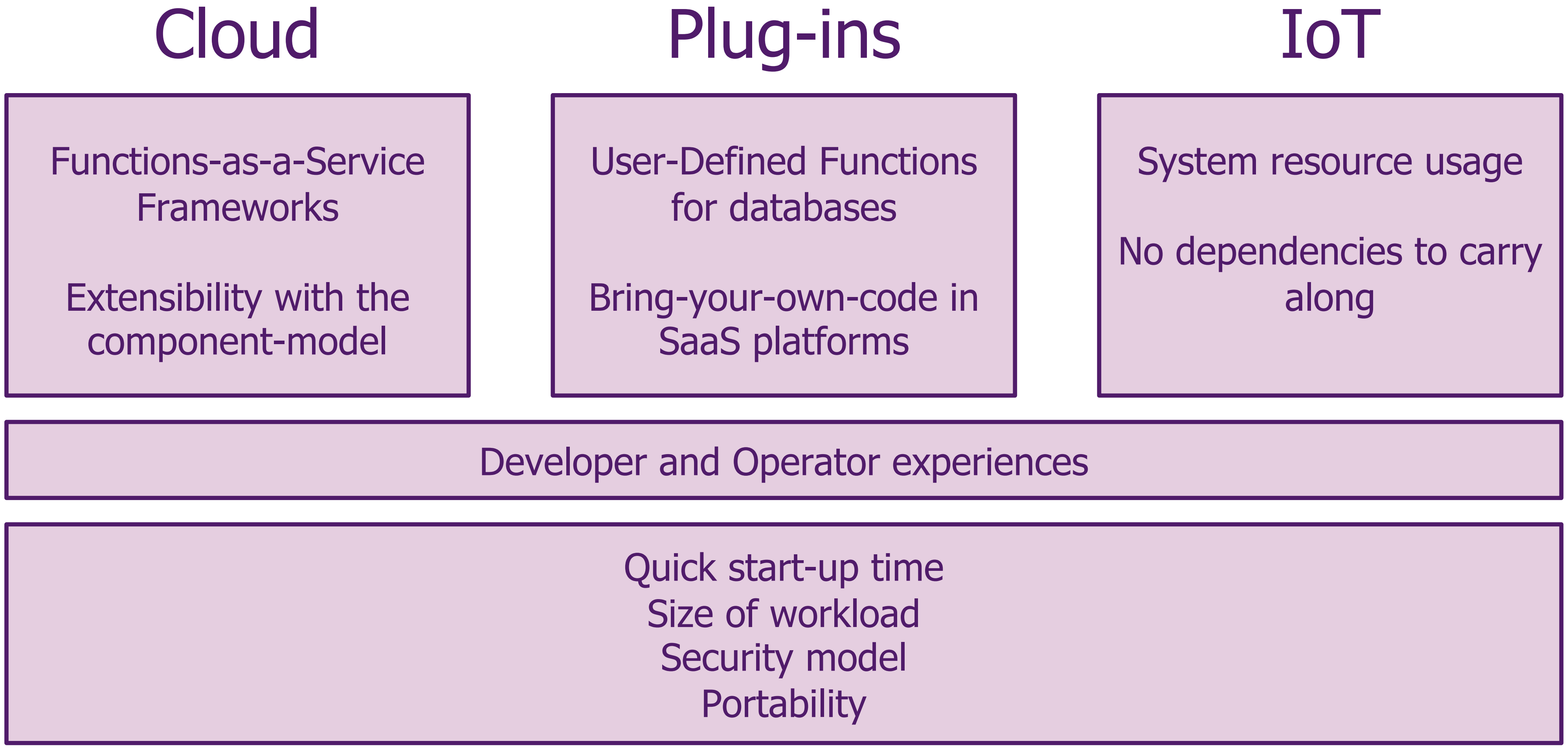
4 things making WebAssembly great

Binary Size	Startup Time	Portability	Security
Rust hello-world ~2MB AOT compiled ~300KB Basic Spin http api ~2.3MB JIT ~1.1MB AOT	Startup times comparable with natively compiled code Only 2.3x slower than native*	Build once, run anywhere! Same build (JIT) works across OS and platform arc	Sandboxed execution Capability based security model

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

GitHub Repo

What are good use-case for WASI?



Cloud

Functions-as-a-Service Frameworks

Extensibility with the component-model

Plug-ins

User-Defined Functions for databases

Bring-your-own-code in SaaS platforms

IoT

System resource usage

No dependencies to carry along

Developer and Operator experiences

Quick start-up time
Size of workload
Security model
Portability

GitHub Repo

3 easy options for running WebAssembly

Use a runtime

```
> cargo build --target wasm32-wasi --release
> wasmtime target/wasm32-wasi/release/my_app.wasm
```

Use a framework

```
> spin build –f my_app/spin.toml
> spin up –f my_app/spin.toml
```

Use runwasi with Kubernetes

```
> docker build --platform wasi/wasm –t my_app .
> docker push ghcr.io/my_name/my_app
> kubectl apply –f ./runtimeclass.yaml
> kubectl apply –f ./my_app.yaml
```

GitHub Repo

Dockerfiles

KubeCon | CloudNativeCon
North America 2023

~/spin_webassembly/Dockerfile

```
FROM scratch

COPY spin.toml .
COPY target/wasm32-wasi/release/hello_world.wasm
target/wasm32-wasi/release/hello_world.wasm
```

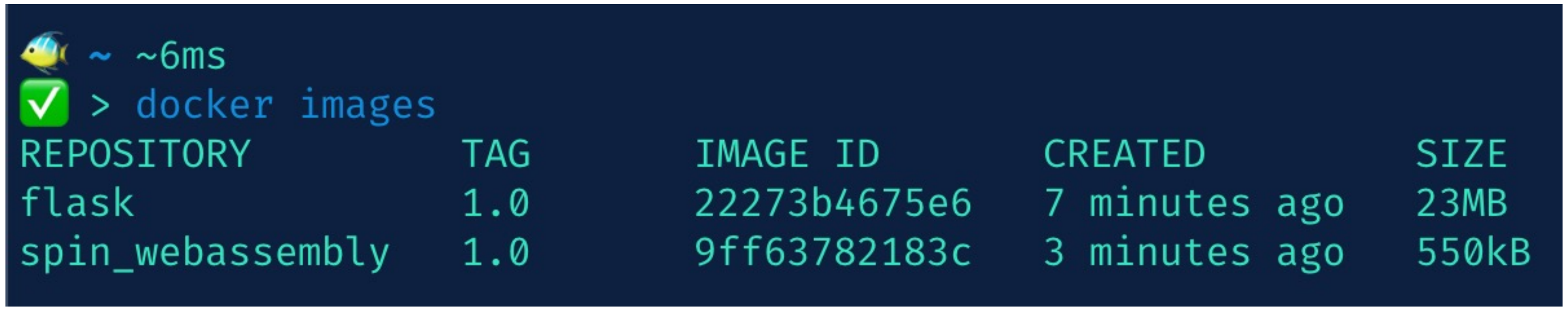
~/python_flask/Dockerfile

```
FROM python:3.10-alpine
WORKDIR /app

COPY requirements.txt /app

RUN --mount=type=cache,target=/root/.cache/pip \ pip3 install -r
requirements.txt
COPY . /app

ENTRYPOINT ["python3"]
CMD ["app.py"]
```



```
~ ~6ms
✅ > docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
flask             1.0       22273b4675e6   7 minutes ago   23MB
spin_webassembly  1.0       9ff63782183c   3 minutes ago   550kB
```

GitHub Repo

Modules

→ Part 1: Build an WebAssembly application using Spin

Part 2: Run your Spin app in a container

Part 3: Deploy to Kubernetes

Part 4: Using Azure Kubernetes Service

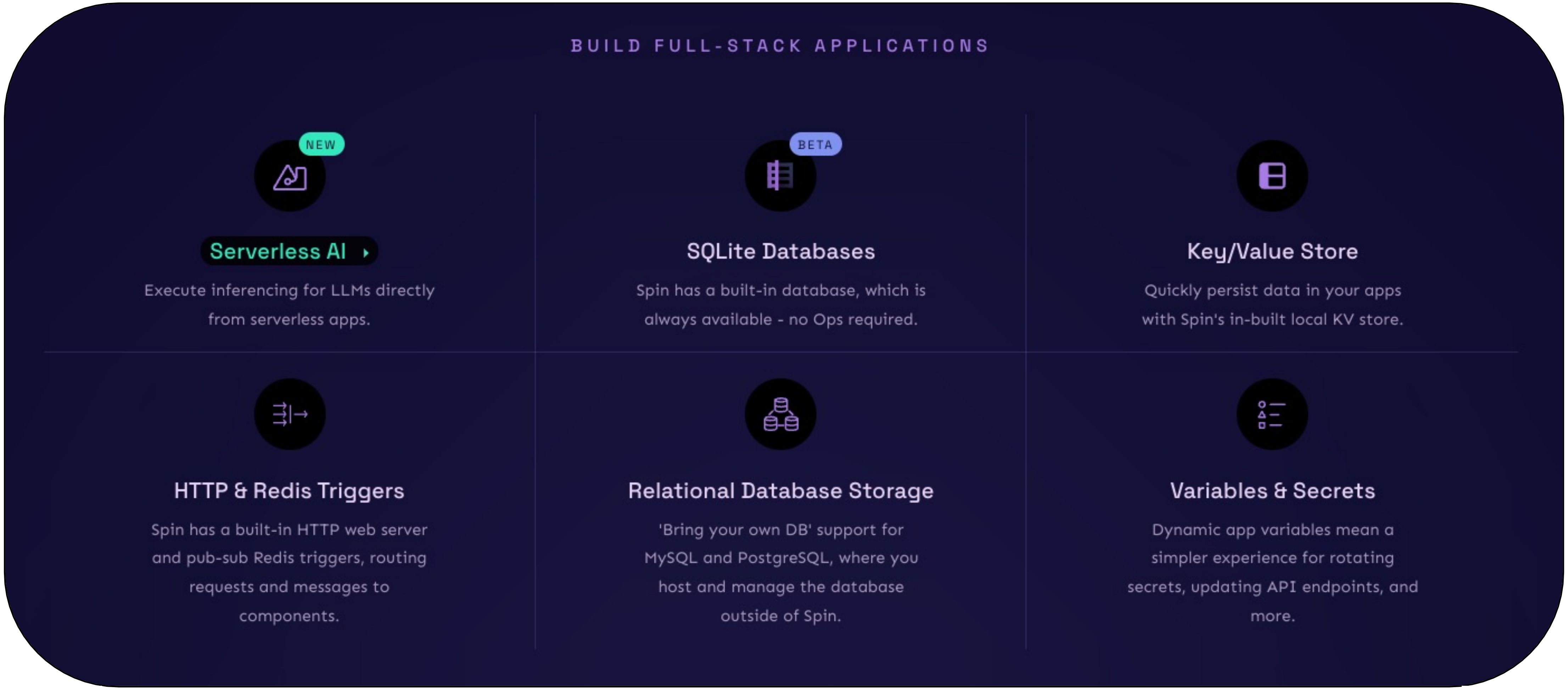
GitHub Repo

SPIN

The developer tool for building serverless WebAssembly apps with Spin

Spin

GitHub Repo



BUILD FULL-STACK APPLICATIONS

NEW

Serverless AI ›

Execute inferencing for LLMs directly from serverless apps.

BETA

SQLite Databases

Spin has a built-in database, which is always available - no Ops required.

Key/Value Store

Quickly persist data in your apps with Spin's in-built local KV store.

HTTP & Redis Triggers

Spin has a built-in HTTP web server and pub-sub Redis triggers, routing requests and messages to components.

Relational Database Storage

'Bring your own DB' support for MySQL and PostgreSQL, where you host and manage the database outside of Spin.

Variables & Secrets

Dynamic app variables mean a simpler experience for rotating secrets, updating API endpoints, and more.

Spin

GitHub Repo

Modules

KubeCon | CloudNativeCon
North America 2023

~~Part 1: Build an WebAssembly application using Spin~~

→ Part 2: Run your Spin app in a container

Part 3: Deploy to Kubernetes

Part 4: Using Azure Kubernetes Service

GitHub Repo

Modules

~~Part 1: Build an WebAssembly application using Spin~~

~~Part 2: Run your Spin app in a container~~

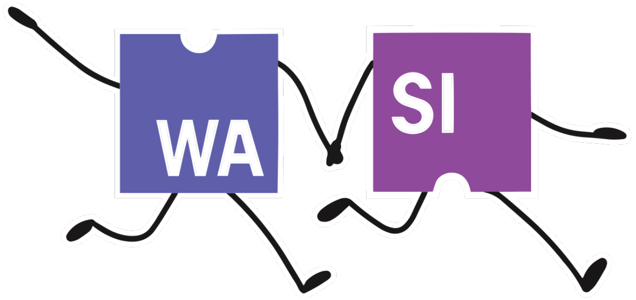
→ Part 3: Deploy to Kubernetes

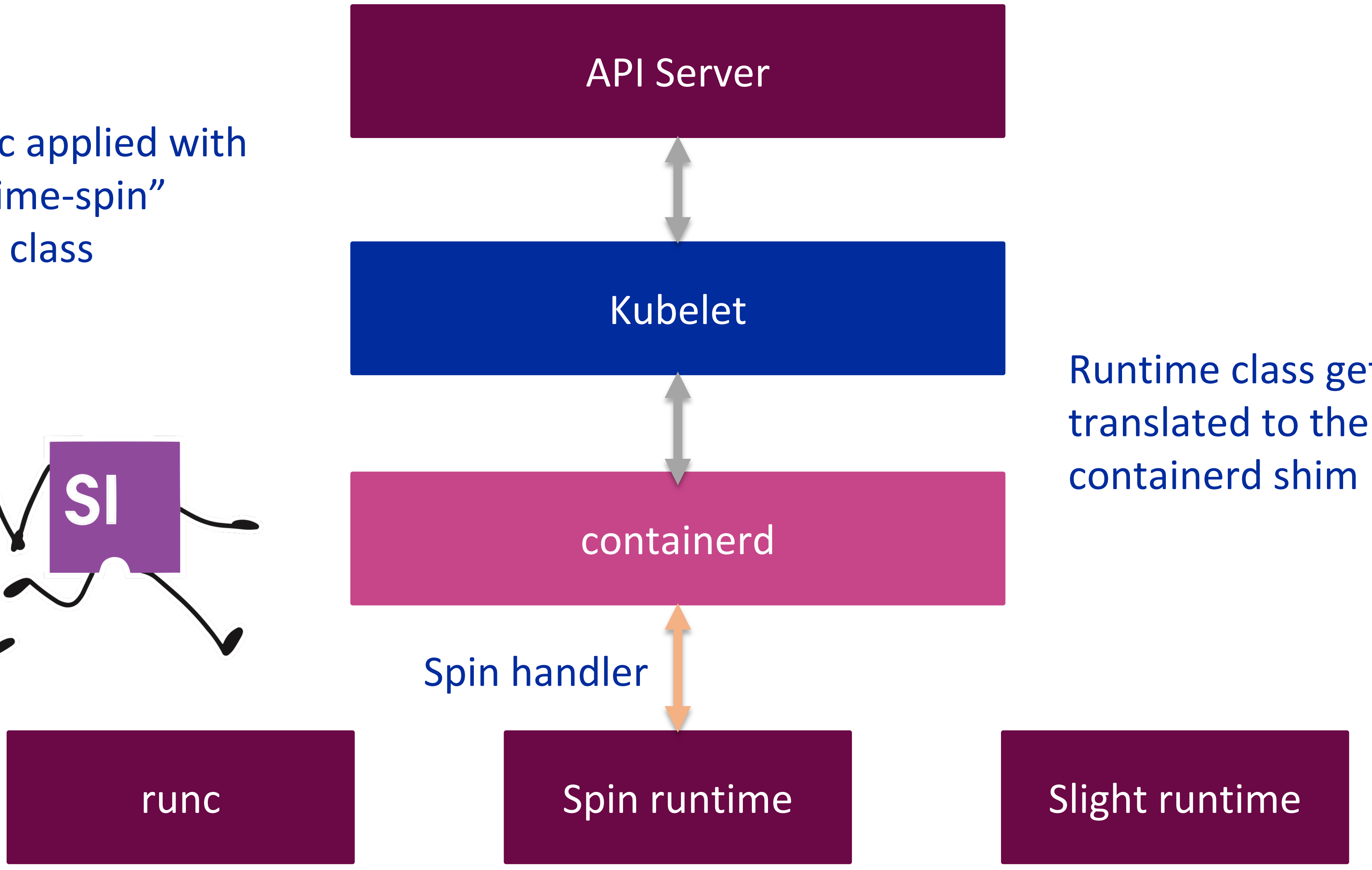
Part 4: Using Azure Kubernetes Service

GitHub Repo

Running Wasm in Kubernetes

Pod spec applied with "wasmtime-spin" runtime class





Runtime class gets translated to the containerd shim handler

GitHub Repo

These containerd shims are just binaries on the $PATH

Running Wasm in Kubernetes

KubeCon CloudNativeCon North America 2023

Runtime Class

```yaml
apiVersion: node.k8s.io/v1
kind: RuntimeClass
metadata:
  name: wasmtime-spin
handler: spin
scheduling:
  nodeSelector:
    spin-enabled: "true"
```

Spin pod deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wasm-spin
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wasm-spin
  template:
    metadata:
      labels:
        app: wasm-spin
    spec:
      runtimeClassName: wasmtime-spin
      containers:
        - name: spin-hello
          image: ghcr.io/deislabs/containerd-wasm-shims/examples/spin-rust-hello:v0.5.1
          command: ["/"]
```

GitHub Repo

Modules

~~Part 1: Build an WebAssembly application using Spin~~

~~Part 2: Run your Spin app in a container~~

~~Part 3: Deploy to Kubernetes~~

→ Part 4: Using Azure Kubernetes Service

GitHub Repo

Learn more / Get involved

Workshop
https://github.com/fermyon/workshops/tree/main/wasm-and-containers

Spin
https://github.com/fermyon/spin

Runwasi
https://github.com/fermyon/spin

CNCF runwasi - Slack
https://cloud-native.slack.com/archives/C04LTPB6Z0V

Spin Discord
https://discord.com/invite/AAFNfS7NGf

Please scan the QR Code above
to leave feedback on this session

PromCon
North America 2021