



**KubeCon**



**CloudNativeCon**

**Europe 2023**





KubeCon



CloudNativeCon

Europe 2023

# Hands on with WebAssembly Microservices and Kubernetes

*Jiaxiao Zhou (Microsoft), Kate Goldenring (Fermyon), Radu Matei (Fermyon), David Justice (Microsoft), Mikkel Mork Hegnhøj (Fermyon)*

# Agenda and Objective

First-hand experience with server-side WebAssembly using Spin, runwasi, and Kubernetes.

We have set up a GitHub repository with all the content you need.

You will have two options:

1. Go ahead and have fun, we have people in the room to help with questions
2. Listen to a 10-min introduction of WebAssembly, then either goto 1, or follow along as we will do the exercises from stage

<https://github.com/deislabs/kc-eu-2023-k8s-wasm-microservices>



# What is WebAssembly?

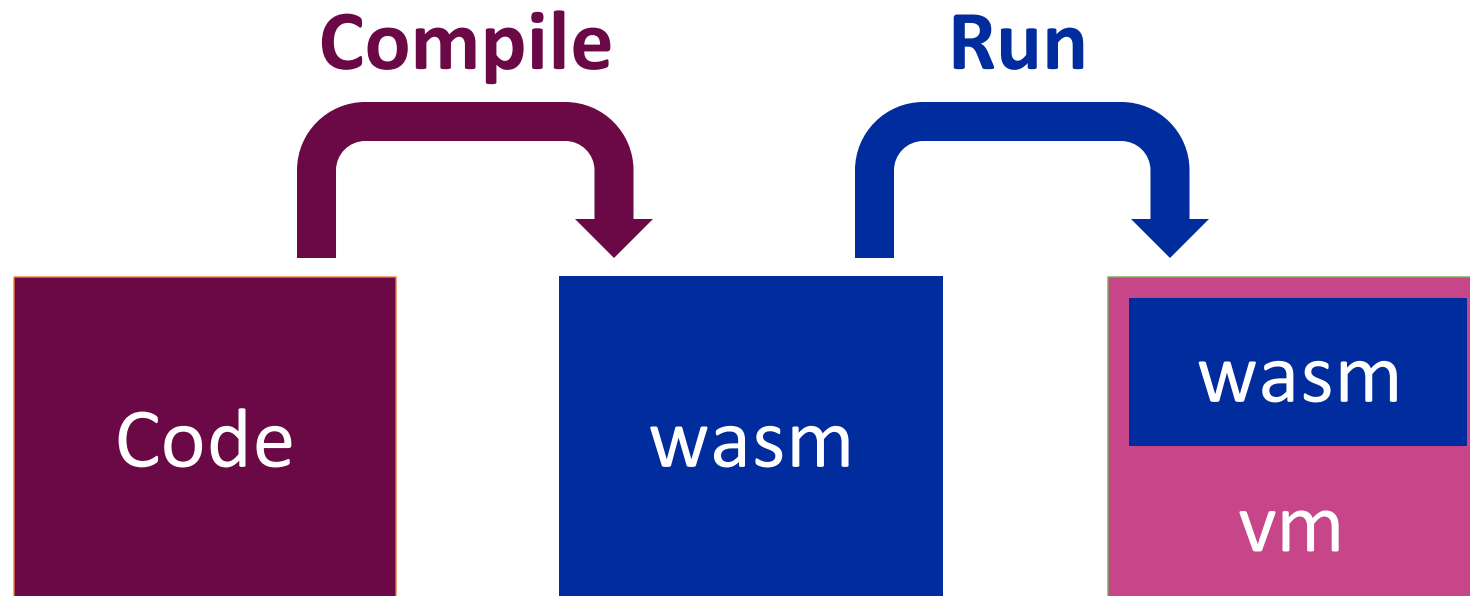
**It is a specification of a binary instruction format, designed as a portable compilation target**

**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	Browser	Other	WASI	Notes
JavaScript	🕒	🕒	🕒	
Python	🕒	✅	✅	
Java	✅	✅	❌	
PHP	✅	✅	✅	
CSS	N/A	N/A	N/A	
C# and .NET	✅	✅	✅	Covers .NET as well
C++	✅	✅	✅	
TypeScript	❌	🕒	❌	Consider <a href="#">AssemblyScript</a>
Ruby	✅	✅	✅	

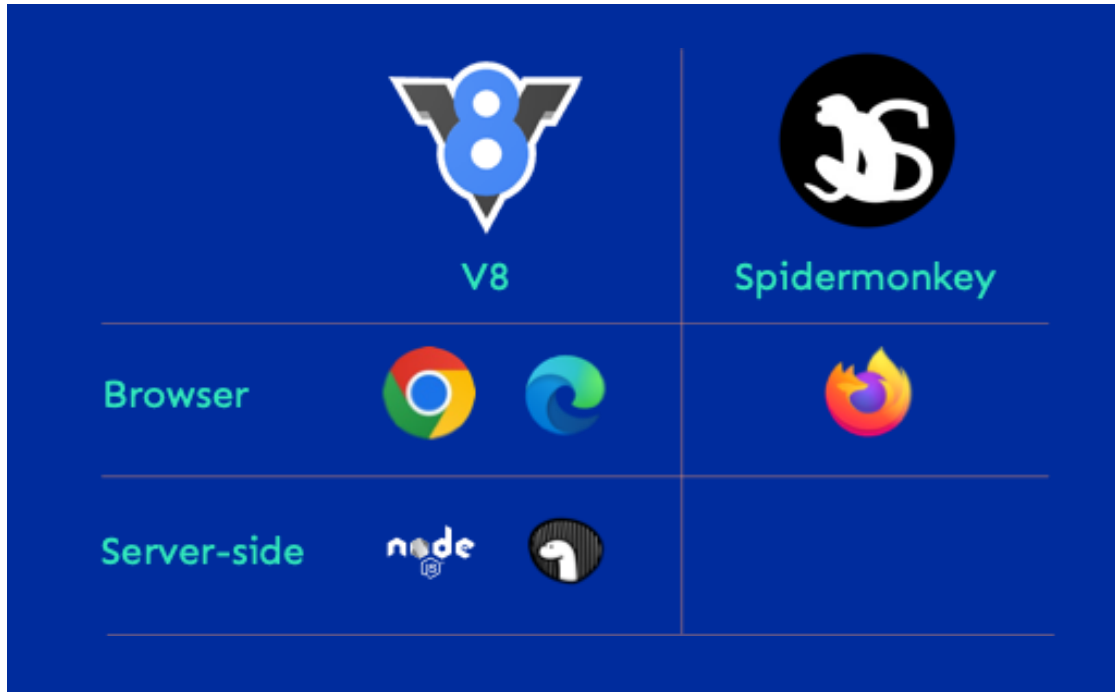


GitHub Repo



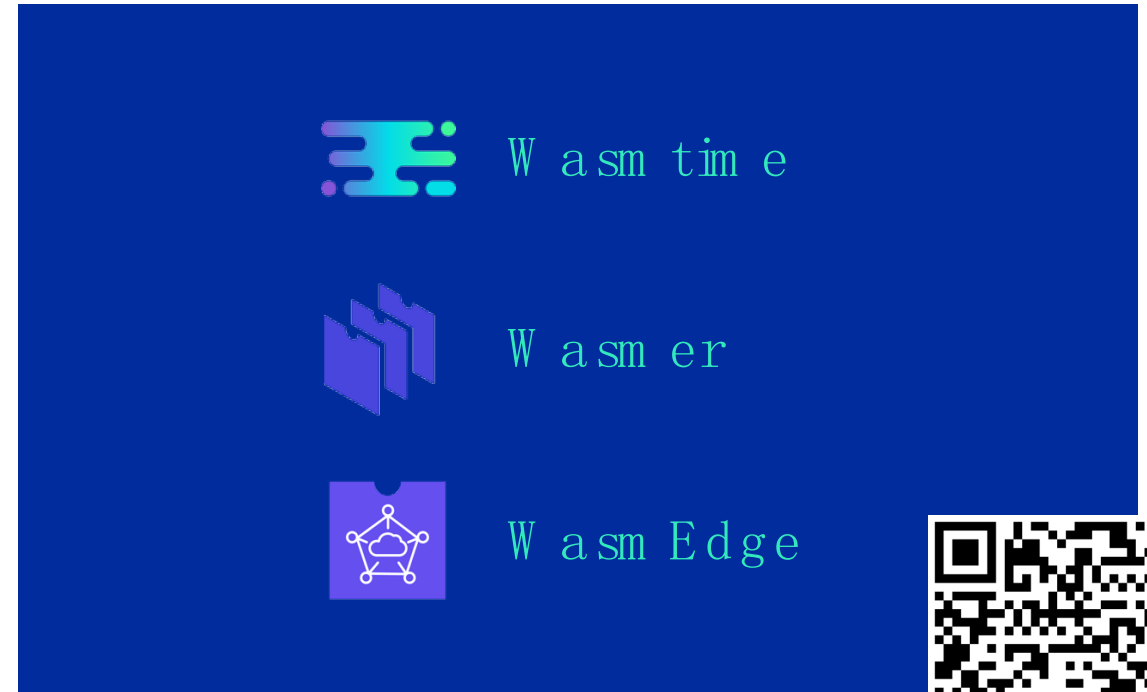
## JavaScript runtimes

Designed to complement and run alongside JS to share functionality between JS and Wasm



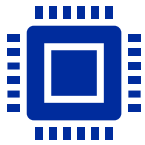
## WASI runtimes

Designed to be independent of browsers, so it doesn't depend on Web APIs or JS, and isn't limited by the need to be compatible with JS



# WASI – The WebAssembly Systems Interface

**POSIX like interface to enable existing applications to target a conceptual OS**  
(e.g., files, sockets, clocks, random numbers, and more)



## 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

**GitHub Repo**





# 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 . -t my_app:latest  
> docker push ghcr.io/my_name/my_app:latest  
> kubectl apply -f ./runtimeclass.yaml  
> kubectl apply -f ./my_app.yaml
```

GitHub Repo



# Dockerfile

## ~/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



# Part 0 and 1

→ **Part 0: Setup**

→ **Part 1: Getting started with Spin**

**Part 2: Run your first Spin application on k3d**

**Part 3: Deploy your Spin applications to Kubernetes**

**Part 4: Magic k8s ball on Azure Kubernetes Service**

**GitHub Repo**



# Ø SPIN

The developer tool for building  
WebAssembly microservices and web applications

GitHub Repo



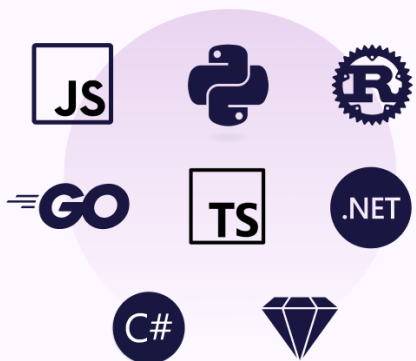
<https://github.com/fermyon/spin>





## The Developer Tool for Serverless WebAssembly

### Bring Your Code



Supports many of the most popular programming languages

`spin new`

### Serverless Made Simple



Works with developer tools & registries.  
Adds internal & external storage

`spin build`

`spin up`

### Deploy & Run



Kubernetes Cluster  
with Containerd

**FERMYON CLOUD**

`spin cloud deploy`

<http://localhost:3000>

<https://github.com/fermyon/spin>

GitHub Repo

# Part 2

~~Part 0: Setup~~

~~Part 1: Getting started with Spin~~

→ **Part 2: Run your first Spin application on k3d**

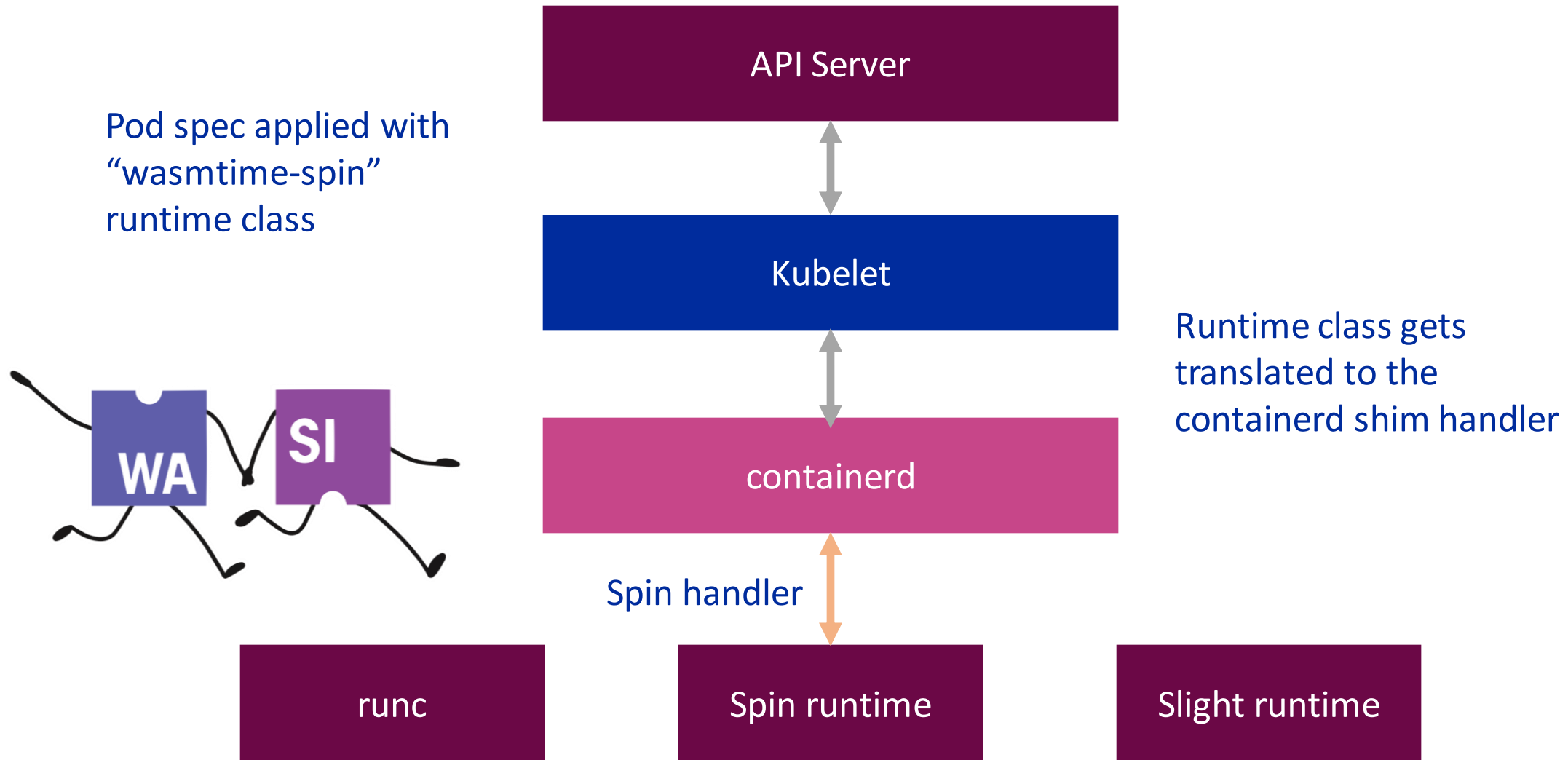
**Part 3: Deploy your Spin applications to Kubernetes**

**Part 4: Magic k8s ball on Azure Kubernetes Service**

GitHub Repo



# Running Wasm in Kubernetes



These containerd shims are just binaries on the \$PATH

GitHub Repo



# Running Wasm in Kubernetes

## Runtime Class

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

## Spin pod deployment

```
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





# Part 3

~~Part 0: Setup~~

~~Part 1: Getting started with Spin~~

~~Part 2: Run your first Spin application on k3d~~

→ **Part 3: Deploy your Spin applications to Kubernetes**

**Part 4: Magic k8s ball on Azure Kubernetes Service**

GitHub Repo



# Part 4

~~Part 0: Setup~~

~~Part 1: Getting started with Spin~~

~~Part 2: Run your first Spin application on k3d~~

~~Part 3: Deploy your Spin applications to Kubernetes~~

➔ **Part 4: Magic k8s ball on Azure Kubernetes Service**

GitHub Repo



# Spinning a Magic 8 Ball

## A magic 8 ball

- Returns a random response to a question

## A Magic k8s ball

- Remembers the responses to questions



GitHub Repo



# Learn more / Get involved



Workshop:

<https://github.com/deislabs/kc-eu-2023-k8s-wasm-microservices>



Runwasi

<https://github.com/fermyon/spin>

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



Spin Discord

<https://discord.com/invite/AAFNfS7NGf>

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





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

# Learn more / Get involved

- TODO

Session QR Codes will be  
sent via email before the event

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