A woman wearing a yellow hard hat and a yellow jacket is looking upwards and to the right. She is holding a black laptop. The background shows an industrial site with orange metal structures under a blue sky with some clouds.

(Serverseitiges)
WebAssembly
professionell
betreiben

ContainerConf

Sprecherintro: Tobias Fenster

Business

- Geschäftsführer und Co-Gründer 4PS by Hilti Deutschland



Community

- Microsoft Regional Director und MVP für Azure (cloud native) und Business Applications (Business Central)
- Docker Captain
- Blog und Podcast "Window on Technology" via tobiasfenster.io

Agenda

Kurze WebAssembly- und SpinKube-Intro

Bereitstellung von SpinKube im (Azure) Kubernetes Cluster

Bereitstellung von Anwendungen

Konfiguration von Anwendungen

Zugriff für Clients

Monitoring von Anwendungen

Nutzung eines Key-Value-Store

WebAssembly (Wasm) - Intro

Sicheres, portables, low-level Codeformat

- Annähernd native Performance in einer sicheren Sandbox
- Hardware-, platform- und sprachunabhängig

Kleine, effizient ausführbare Dateien

- Modulares, binäres Format
- Stream- und parallelisierbar

Offener Standard: <https://webassembly.github.io/spec/core/>

Zunächst v.a. im Browser genutzt: Sicher, portabel, kleine, effizient, streambar, ...

Braucht eine Ausführungsumgebung sowie ein „System Interface“ (WASI) für Zugriff auf z.B. Dateien, Netzwerk etc.

SpinKube - Intro

„Hyper-efficient serverless on Kubernetes, powered by WebAssembly“

Ermöglicht die Ausführung von Spin-Anwendungen (Wasm) in Kubernetes

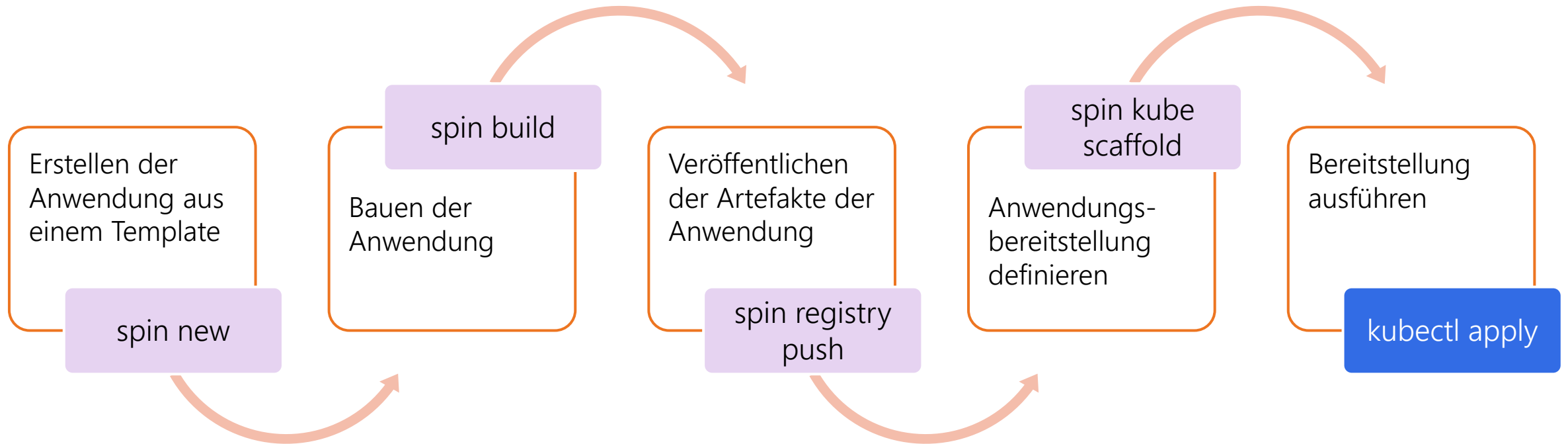
Beinhaltet Spin operator, containerd Spin shim und runtime class manager

Wesentlich entwickelt von Fermion, Microsoft, SuSE und Reply

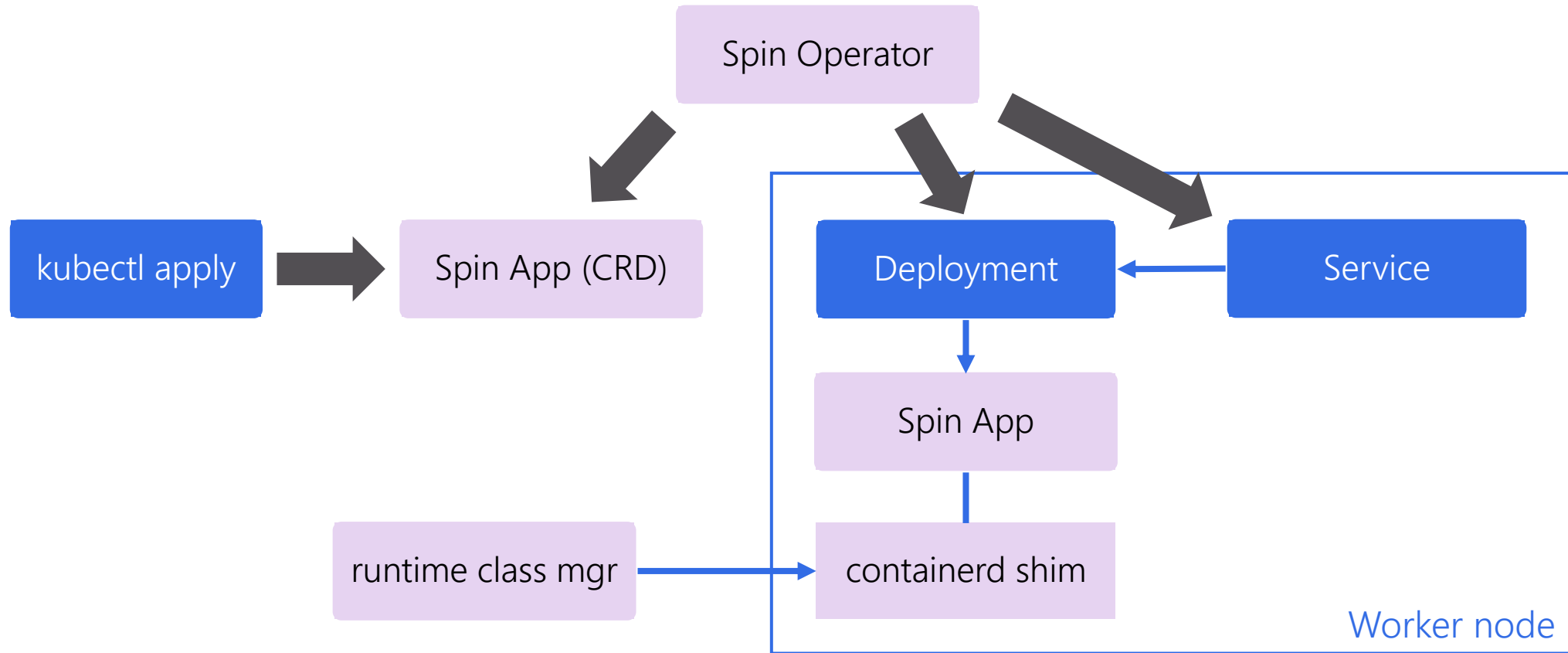
OpenSource-Projekt, aktuell in der CNCF Sandbox



SpinKube - Intro: Entwicklung und Bereitstellung



SpinKube - Intro: Lifecycle und Runtime Management



SpinKube – Bereitstellung im (Azure) Kubernetes Cluster

CRDs und runtime class

```
kubectl apply -f https://github.com/.../spin-operator.crd.yaml
```

```
kubectl apply -f https://github.com/.../spin-operator.runtime-class.yaml
```

Operator und executor

```
helm install kwasm-operator ... --set kwasmOperator.installerImage=...
```

```
helm install spin-operator ... oci://ghcr.io/spinkube/charts/spin-operator
```

```
kubectl apply -f https://github.com/.../spin-operator.shim-executor.yaml
```

Außerdem für die folgenden Demos

- Azure Container Registry (Artifacts), verbunden mit Azure Kubernetes Services Cluster
- Azure Key Vault (Konfiguration/Secrets), Azure Cosmos DB (Key/Value-Store)
- Azure Web App Routing (gemanagter Nginx Ingress für Client-Zugriff)
- OpenTelemetry collector, Jaeger (Monitoring)

Bereitstellung einer Spin Wasm Anwendung

Vorbereitung: Spin Anwendung entwickeln

Anwendungspaket

- als OCI Artifact erstellen
- in OCI Registry bereitstellen (Docker Hub, GitHub Container Registry, Azure Container Registry, ...)

Kubernetes Manifest generieren

Anwendung in Kubernetes Cluster bereitstellen



Konfiguration einer Spin Wasm Anwendung

Voraussetzung: Azure Key Vault erstellt, Zugriff in Runtime Config hinterlegt
Variablen (Konfiguration)

- in Spin-Manifest (spin.toml) definieren
- im Anwendungscode referenzieren
- lokal und im Kubernetes-Manifest festlegen (direkt, ConfigMap, Secret)
- oder aus Azure Key Vault lesen



Zugriff von Clients auf eine Spin Wasm Anwendung

Voraussetzung: Gemanagter Ingress in Azure Kubernetes Service vorhanden

Gemanagten Ingress konfigurieren

Auf Anwendung zugreifen



Monitoring einer Spin Wasm Anwendung

Voraussetzung: OpenTelemetry Collector und Jaeger vorhanden, Spin executor mit OpenTelemetry

Executor umstellen

Calls ausführen und in Jaeger einsehen



Nutzung eines Key/Value-Stores in einer Spin Anwendung

Voraussetzung: Azure Cosmos DB Account, Datenbank und Container erstellt,
Zugriffsinformationen in Runtime Config hinterlegt

Key/Value-Store in Spin Manifest konfigurieren

Runtime Config in Kubernetes Manifest integrieren

Anwendung anpassen: GET / POST / DELETE / HEAD für Store



Rückblick

Kurze WebAssembly- und SpinKube-Intro

Bereitstellung von SpinKube im (Azure) Kubernetes Cluster

Bereitstellung von Anwendungen

Konfiguration von Anwendungen

Zugriff für Clients

Monitoring von Anwendungen

Nutzung eines Key-Value-Store

Fragen? Danke für die Aufmerksamkeit

Ich freue mich über Feedback und weiteren Austausch, direkt
oder über Social Media