

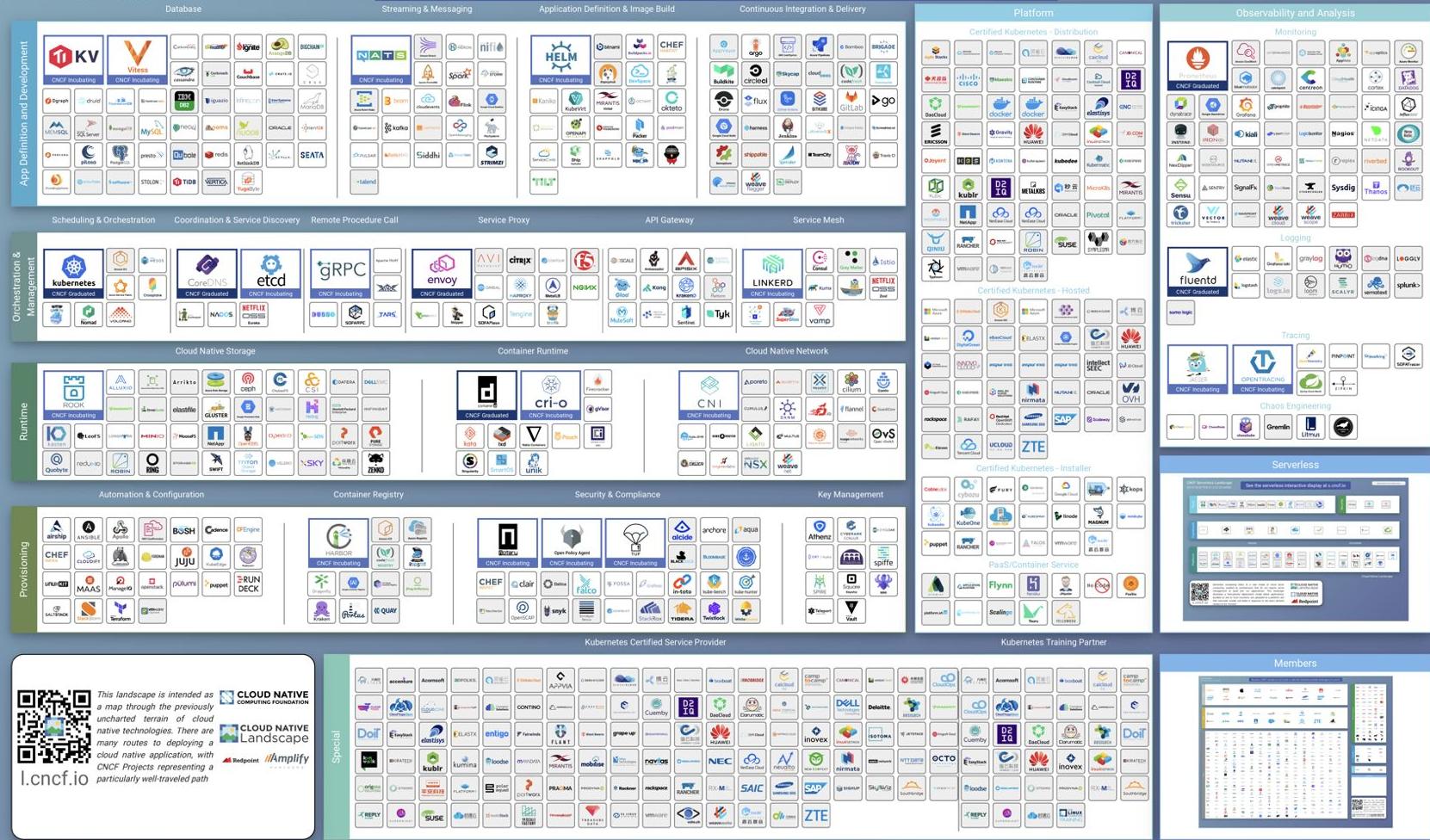
Serverless



Serverless

is the next logical evolution in
Cloud Native Software Development





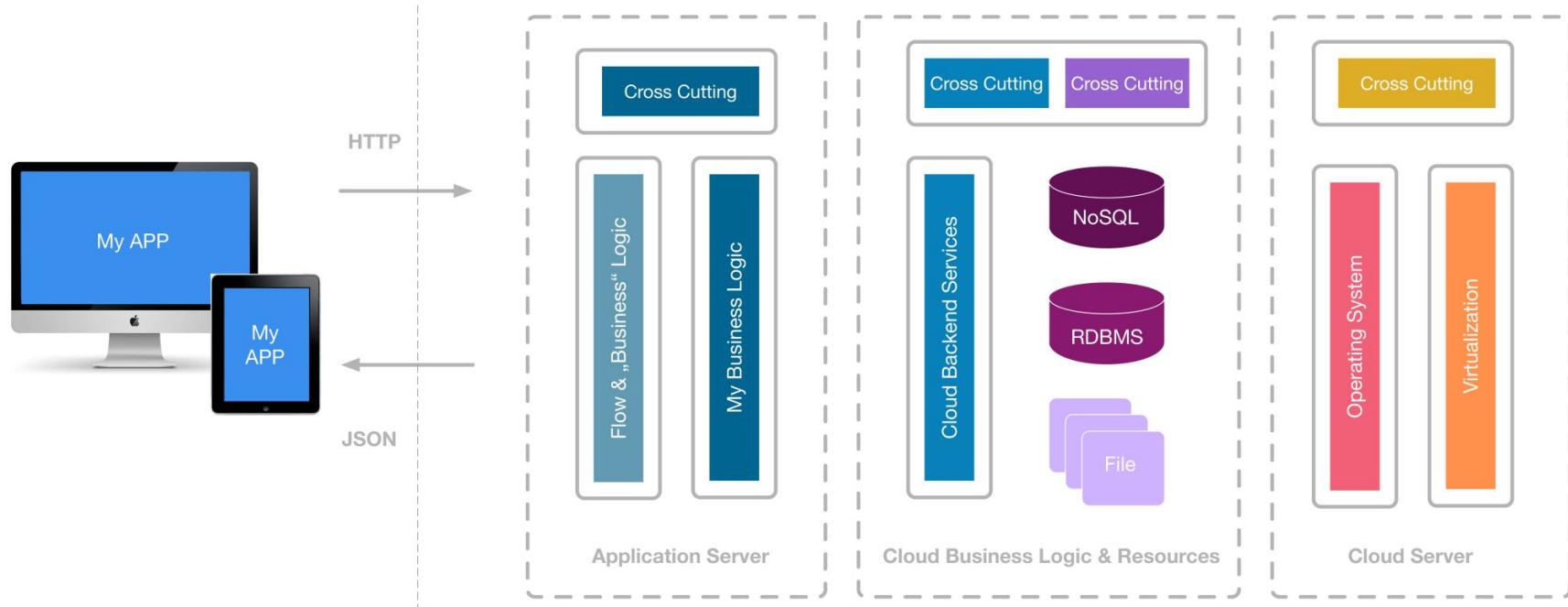
The background of the slide features a complex, abstract network graph composed of numerous small, semi-transparent white dots connected by thin white lines, forming a web-like structure against a dark blue background.

CLOUD NATIVE SOFTWARE DEVELOPMENT IS

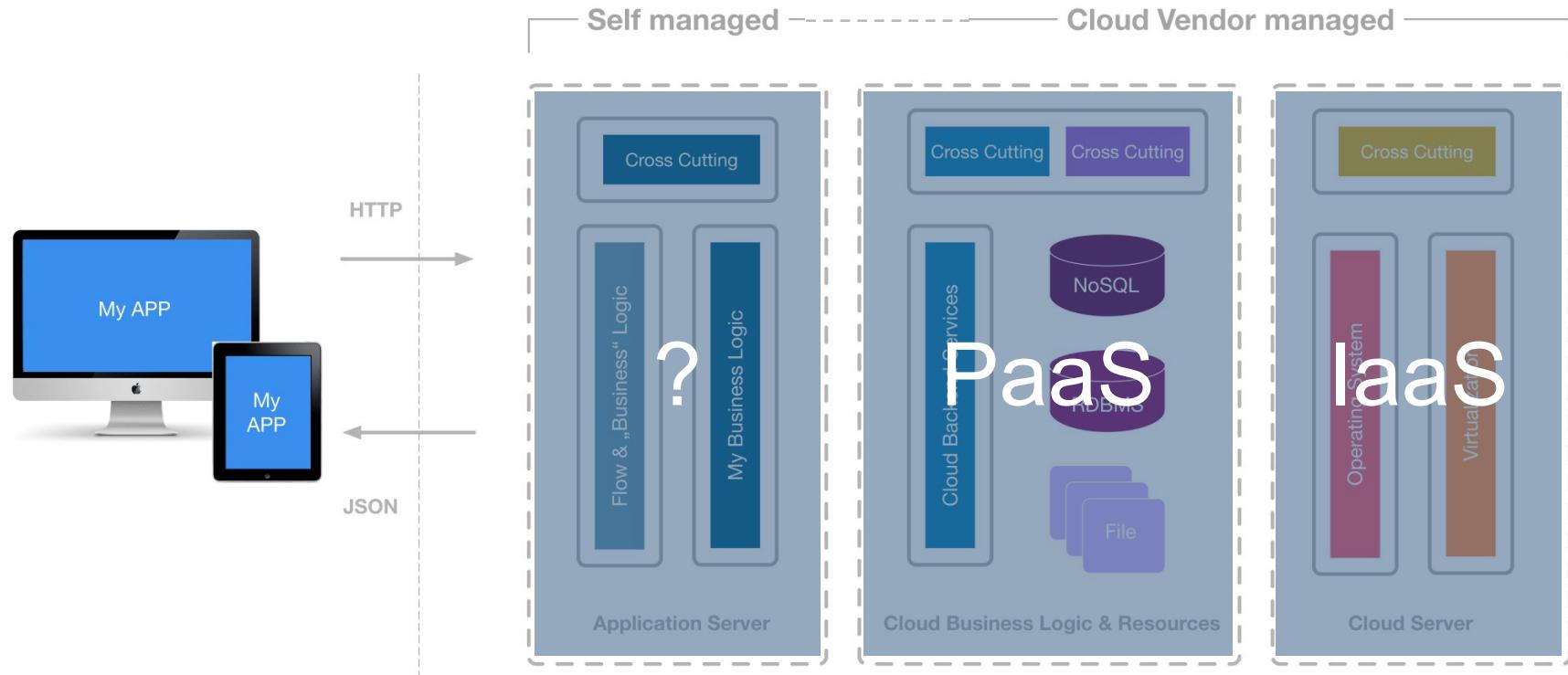
COMPLEX.

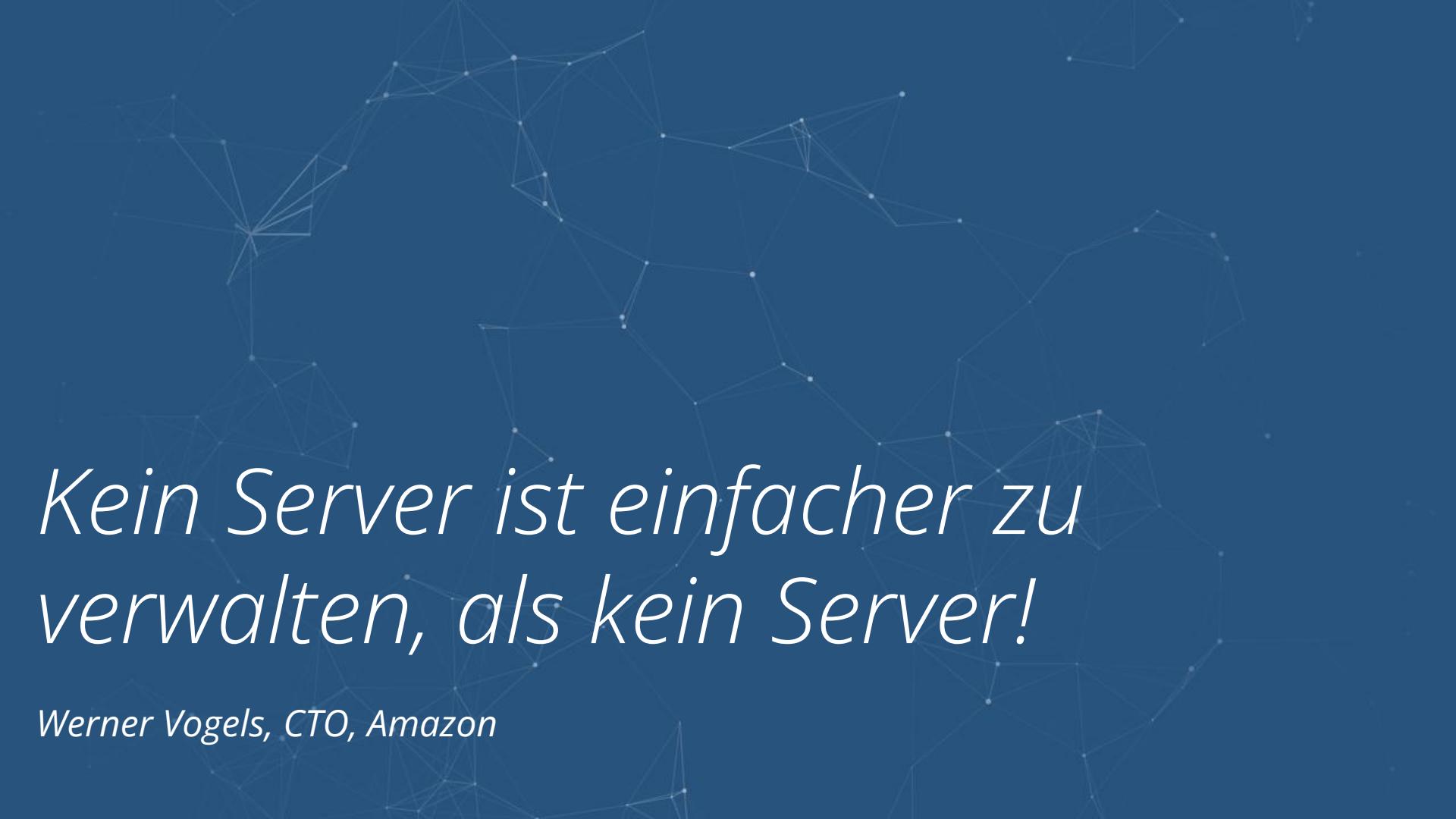
DOCKER, YAML, MICROSERVICES, KUBERNETES, ET.AL.

Traditionelle Cloud-basierte Anwendungsarchitektur



Traditionelle Cloud-basierte Anwendungsarchitektur



The background of the slide features a complex, abstract network graph composed of numerous small, semi-transparent white dots connected by thin gray lines. This pattern creates a sense of depth and connectivity, resembling a cloud or a complex system of data nodes.

*Kein Server ist einfacher zu
verwalten, als kein Server!*

Werner Vogels, CTO, Amazon



Bild: pavlinec – gettyimages.de

Serverless computing refers to a new model of cloud native computing,

Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications.

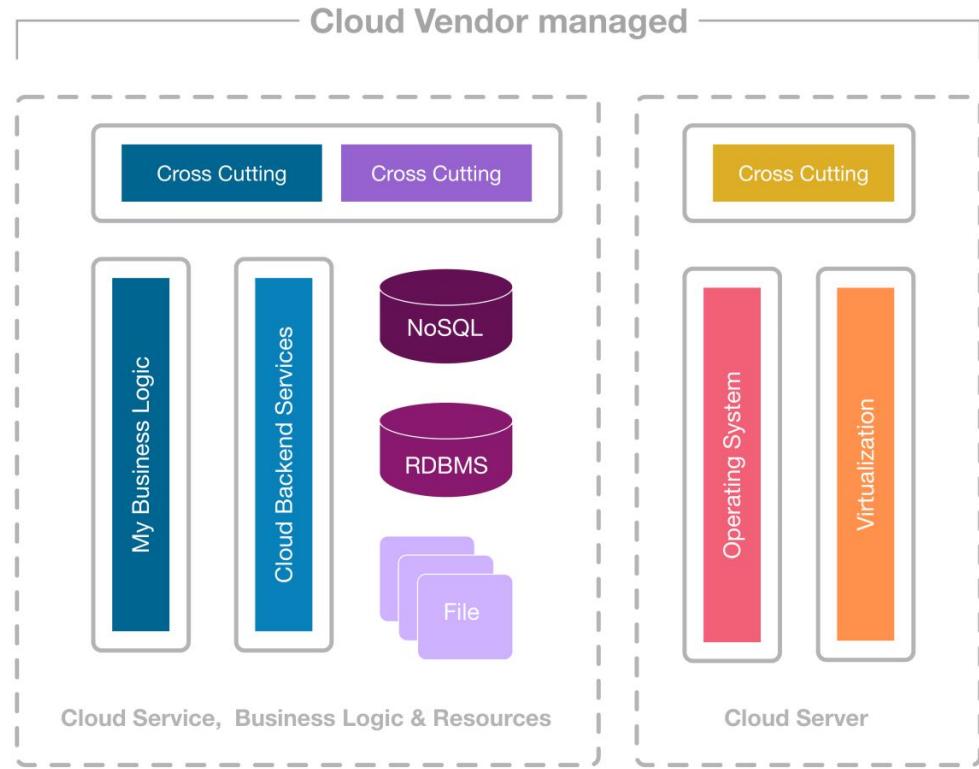
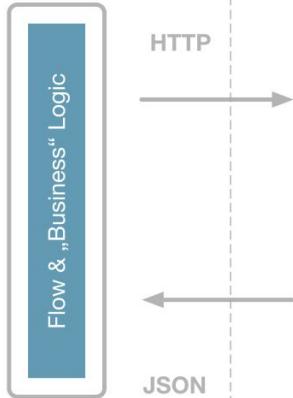
Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. It leverages a finer-grained deployment model

Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. It leverages a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform

Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. It leverages a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment.

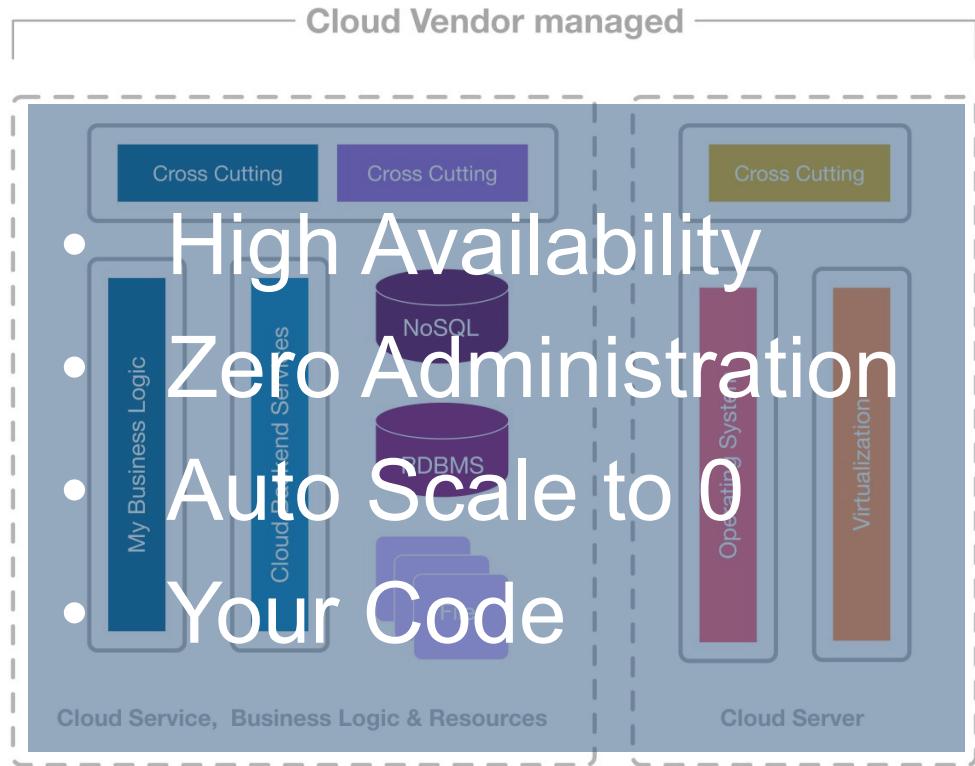
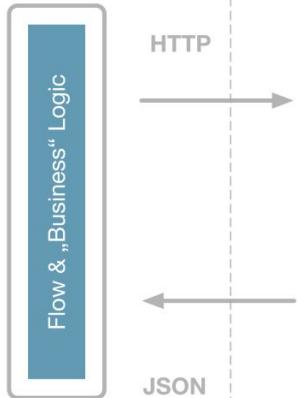
Serverless Anwendungsarchitektur

Run Code, not Servers!

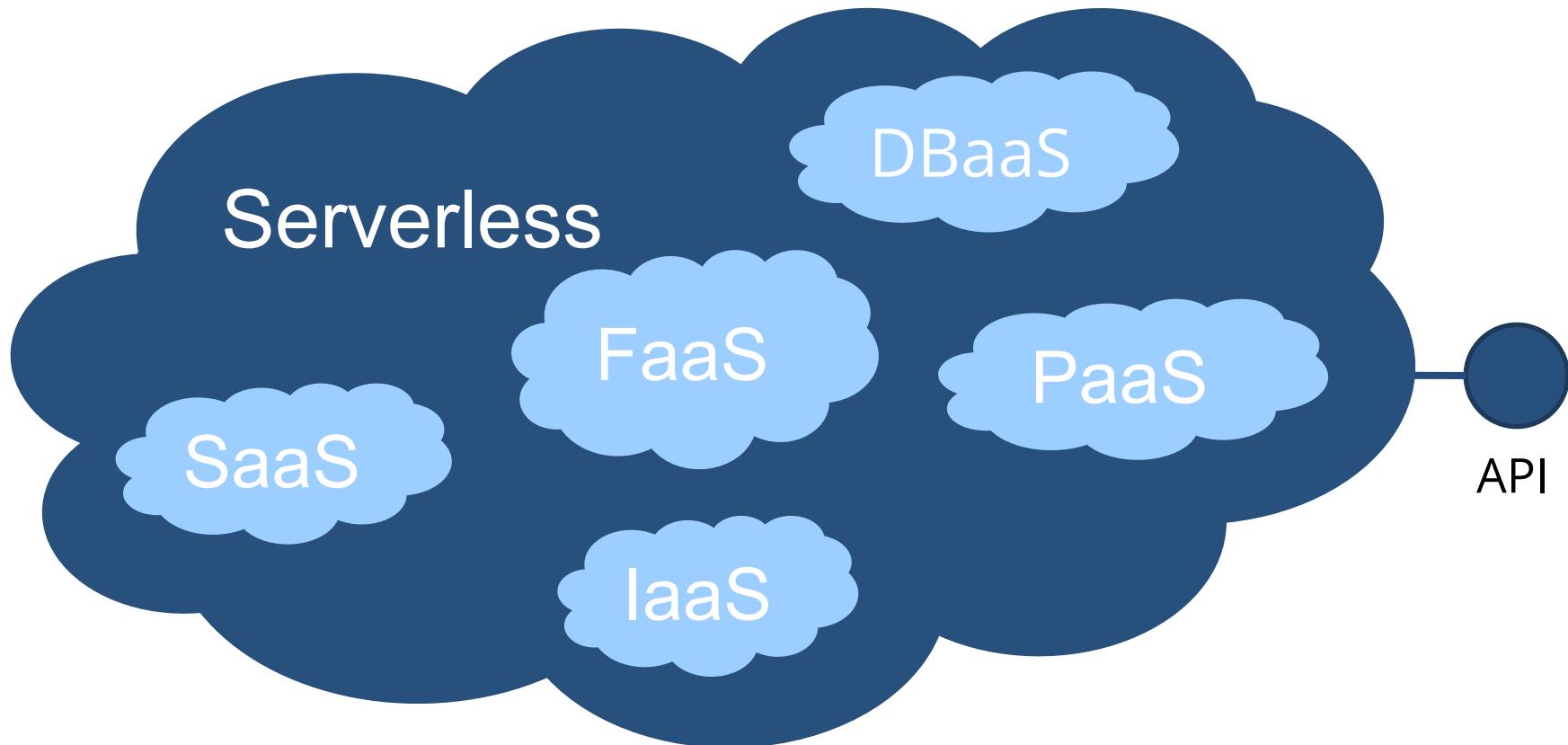


Serverless Anwendungsarchitektur

Run Code, not Servers!



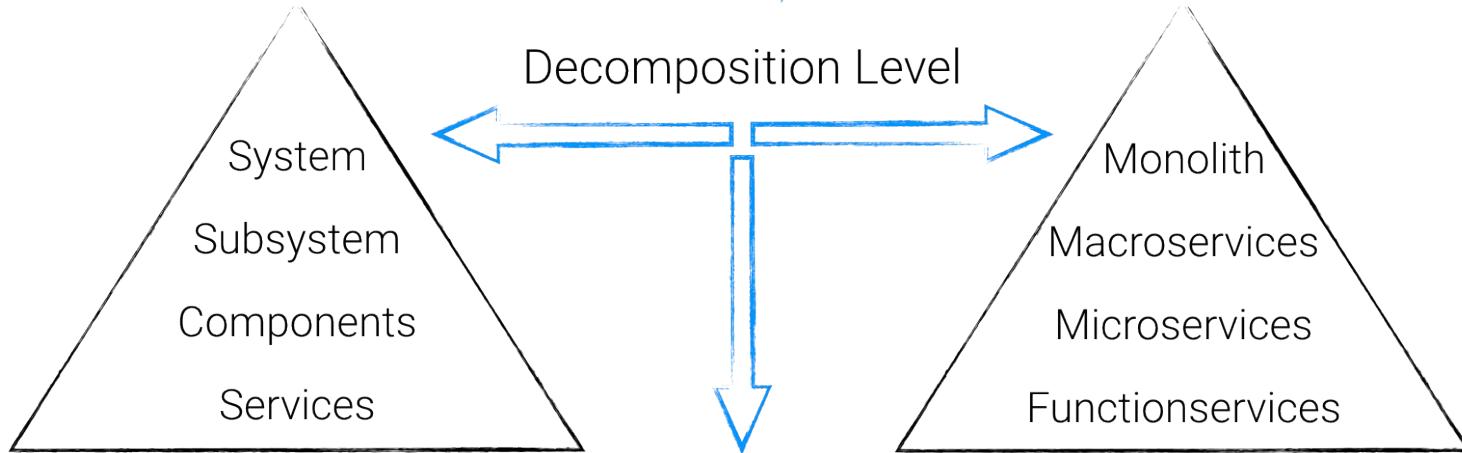
Out-of the Box Self-scaling Fully Managed Backend



Dev Components



Ops Components

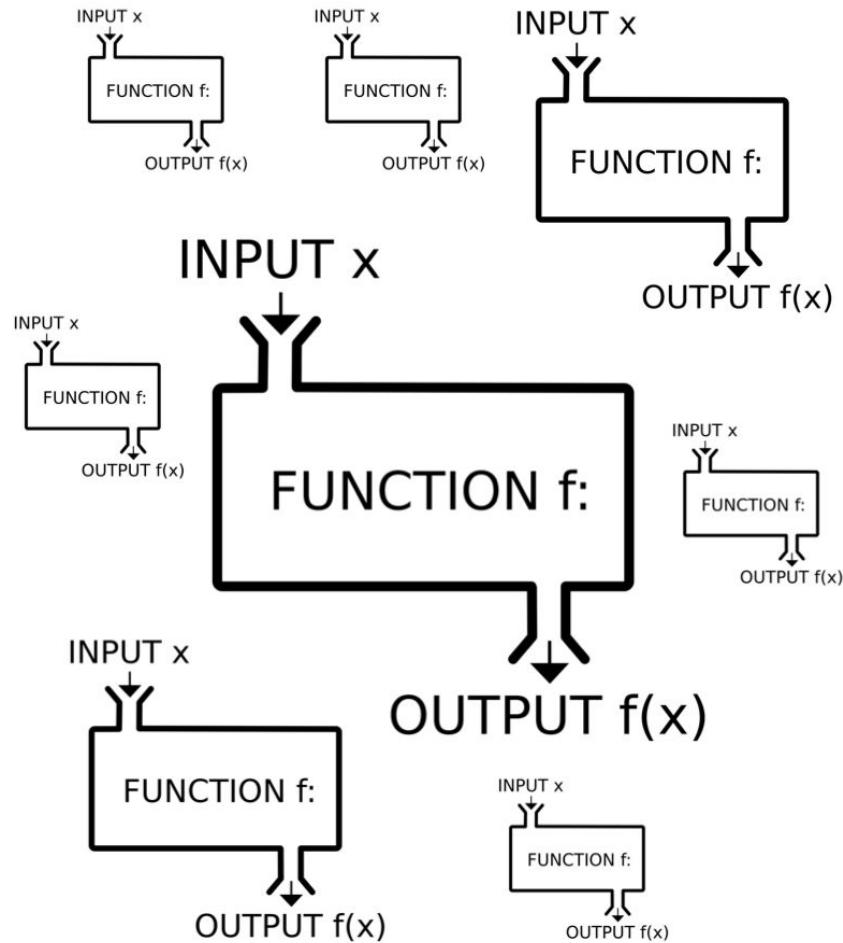


Decomposition Trade-Offs

- | | |
|--|---|
| <ul style="list-style-type: none">+ More flexible to scale+ Runtime isolation (crash, slow-down, ...)+ Independent releases, deployments, teams+ Higher resources utilisation | <ul style="list-style-type: none">- Distribution debt: Latency, Consistency- Increased infrastructure complexity- Increased troubleshooting complexity- Increased integration complexity |
|--|---|

Functions

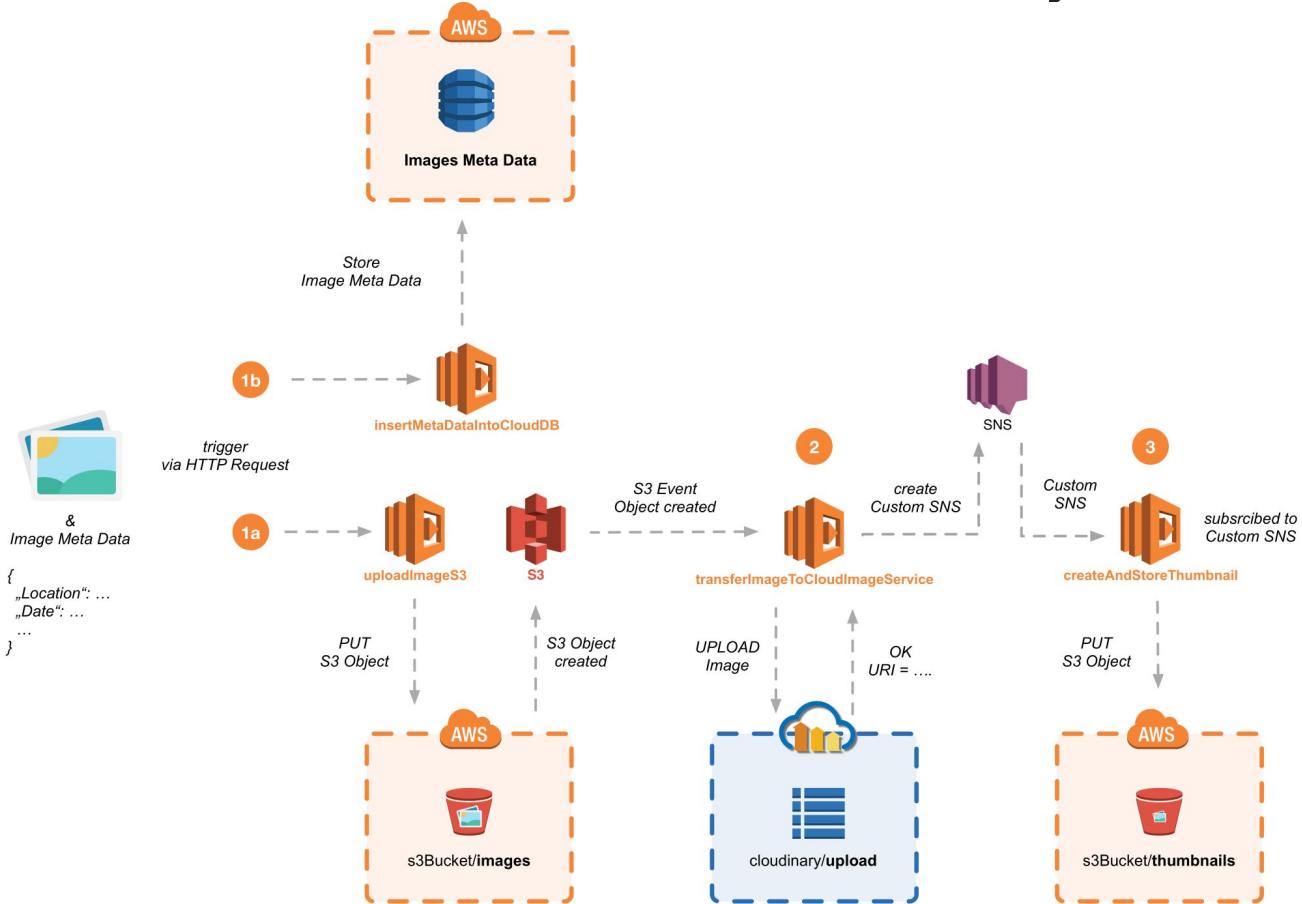
as preferred Serverless Application
Programming Model



EVENT-DRIVEN ARCHITECTURE

*enables loosely coupled reactive
software components and services.*

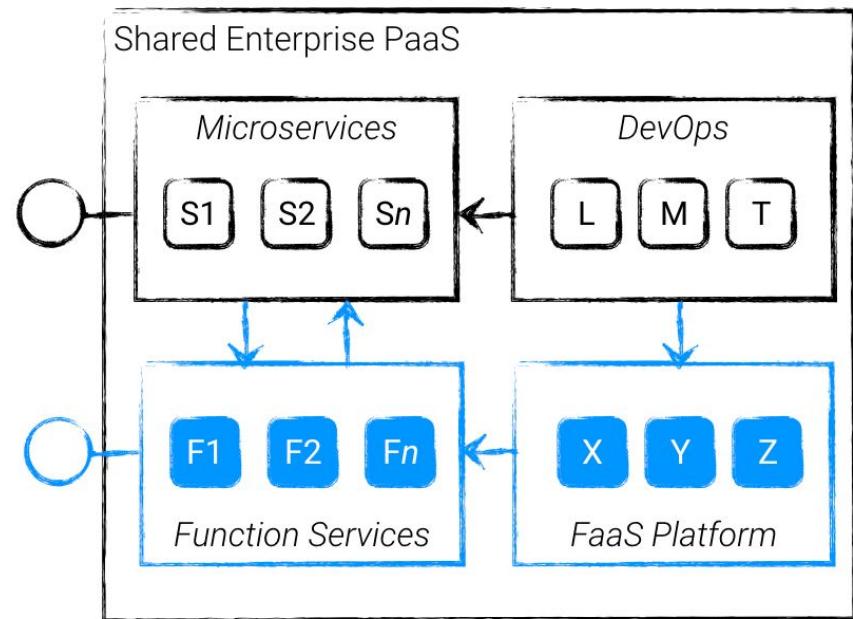
Create Thumbnails the AWS Lambda Way



Use Case 1

Hybrid Architectures

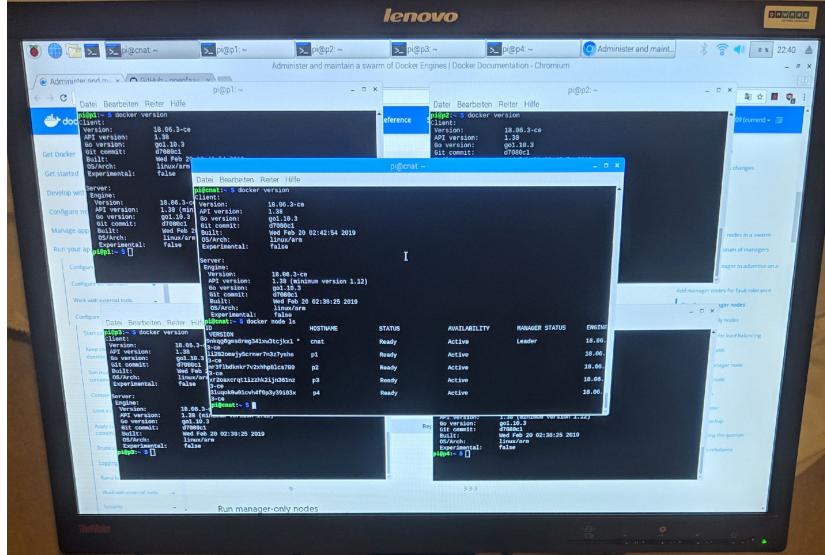
- Kombination von Microservice Architektur mit EDA
- Nutzung von Function Services für Event-getriebene Use Cases
- Reduzierter Ressourcen-Verbrauch per Scale-to-Zero
- Integration in bestehende Enterprise PaaS Umgebung



Use Case 2

Edge und Fog Computing

- Anbindung unserer LoRaWan Raum-Sensoren mittels Serverless Backend
- Couch Projekt: Nutzung von FaaS auf Low Power Devices
- Unterstützung von leichtgewichtigen Cluster Scheduler wie Docker Swarm

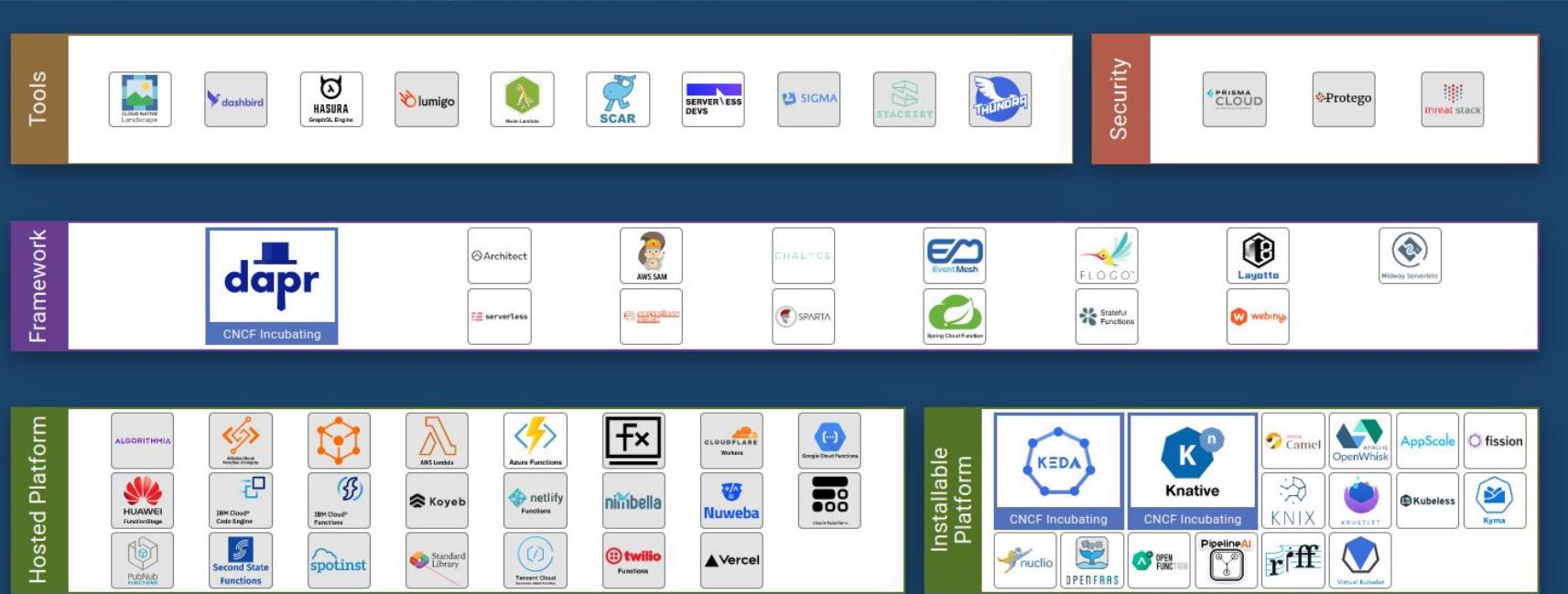


<https://github.com/lreimer/raspi-swarm-box>

Self-hosted Serverless

- *Scale to zero*
- *React to events*

CNCF Serverless Landscape



Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment





OPENFAAS



fission



Kubeless



nuclio



Kyma

siehe auch <https://bit.ly/2Mh1kxj>

Die Kandidaten

- OpenFaas
<https://www.openfaas.com>
- Fission
<https://fission.io>
- ~~Kubeless~~
<https://kubeless.io>
- Nuclio
<https://nuclio.io>
- Knative
<https://knative.dev/>
- Kyma
<https://kyma-project.io>

LANGUAGE	USE CASES	GENERATION	PLATFORMS	RUNTIMES	TRIGGERS
FISSION	GO	ENTERPRISE	2ND	K8S	GO, PYTHON, NODEJS, JAVA/JVM CRON, HTTP, NATS, AZURE QUEUE STORAGE, KAFKA, KUBEWATCH
KUBELESS	GO	ENTERPRISE	2ND	K8S	NODEJS, JAVA, GO, JVM, PYTHON, PHP, RUBY, .NET CORE, BALLERINA, VERTX CRON, HTTP, NATS, KINESIS, KAFKA
OPENFAAS	GO	ENTERPRISE, IOT	1ST	K8S, DOCKER	GO, C#, JAVA8, DOCKERFILE, NODEJS, PHP, PYTHON, RUBY HTTP, CRON, KAFKA, AWS SNS, S3, CLOUDEVENTS, IFTTT, REDIS, MQTT, NATS
NUCIO	GO	ENTERPRISE, IOT	2ND	DOCKER, K8S, AWS, GCP	NET CORE, GO, JAVA, NODEJS, PYTHON, SHELL CRON, EVENTHUB, HTTP, KAFKA, KINESIS, NATS, RABBITMQ, MQTT
OPENWHISK	SCALA	ENTERPRISE, HOSTED?	2ND	K8S, MESOS, DOCKER, OPENSHIFT	NODEJS, SWIFT, JAVA, GO, CLOUDANT, RSS, KAFKA, SCALA, PYTHON, PHP, RUBY, NET CORE, BALLERINA JIRA, BLUEMIX PUSH, SLACK, GITHUB
FN PROJECT	GO	ENTERPRISE, HOSTED?	1ST	DOCKER, K8S	JAVA, GO, NODEJS, PYTHON, RUBY HTTP

IT DEPENDS ON YOUR USE CASE.

- › FISSION IS A PRETTY COMPLETE PLATFORM.
- › OPENFAAS IS VERY POPULAR WITH AN ACTIVE COMMUNITY. CURRENTLY THE ONLY ONE WITH SUPPORT FOR ARM DEVICES.
- › NUCLIO IS FAST, LIGHTWEIGHT AND HAS SUPPORT FOR MANY TRIGGERS. PROMISING ROADMAP.
- › KUBELESS ~~IS~~ ^{was} LIGHTWEIGHT AND SIMPLE.

Evaluation der Lehrveranstaltung (15 min)

<https://evasys.cc.hm.edu/evasys/online.php?pswd=C2GA3>

