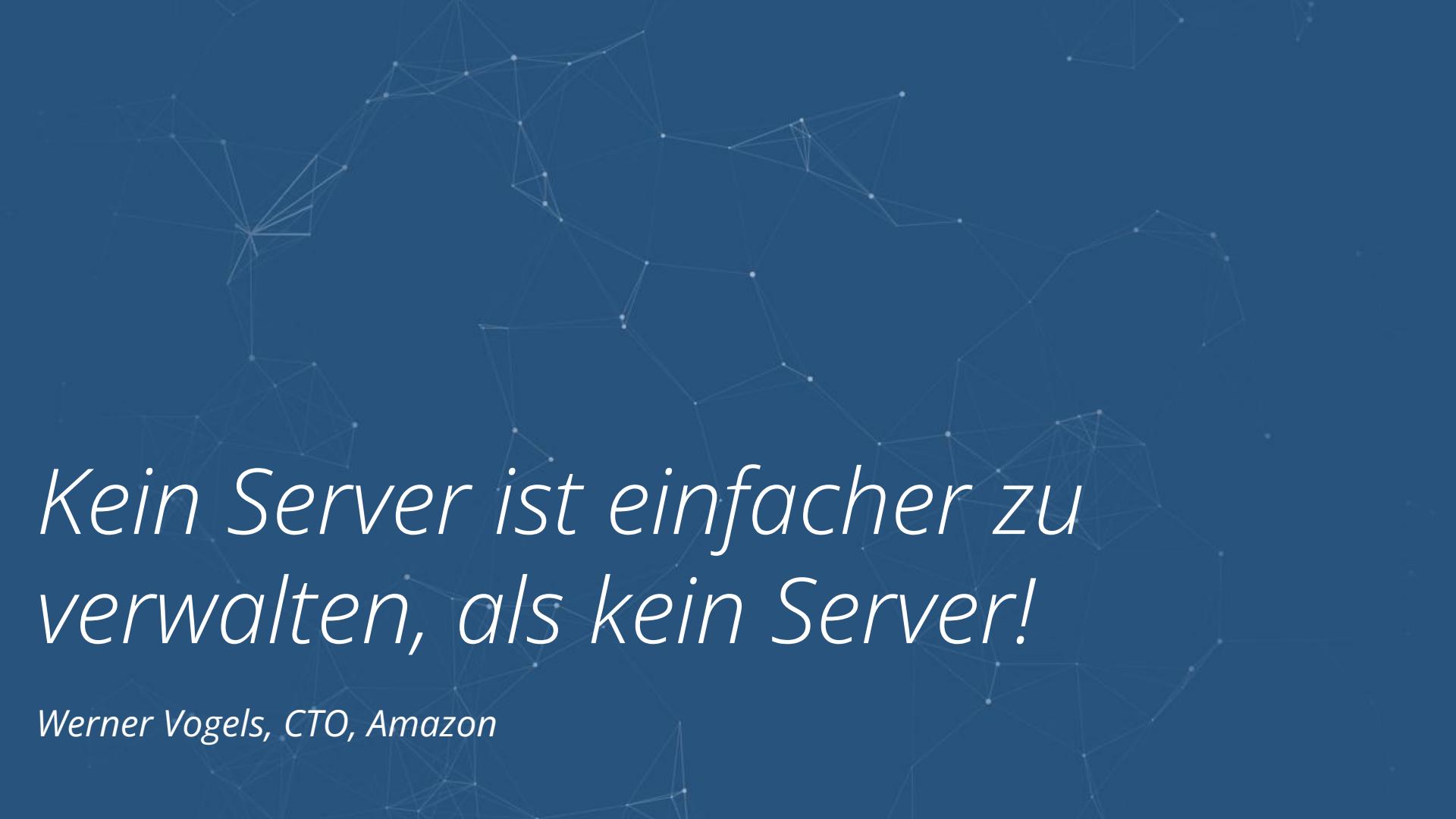


Cloud Computing

Serverless

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 points.

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

Werner Vogels, CTO, Amazon



Bild: pavlinec – gettyimages.de

The background of the slide features a complex, abstract network graph composed of numerous small, semi-transparent white dots connected by thin gray lines. These dots are scattered across the dark blue background, creating a sense of depth and connectivity. The overall aesthetic is modern and technological.

Ist Cloud Native nicht immer Serverless?

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape ([lcnf.io](#)) has a large number of options. This Cloud Native Trail Map is a recommended process for navigating the complex cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer ([cncf.io/training](#))

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider ([cncf.io/ksp](#))

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally ([cncf.io/enduser](#))

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

[lcnf.io](#)

v20200507

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices



3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cnf.io/kick
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



5. SERVICE PROXY, DISCOVERY, & MESH

- Consul is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rock is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Setting up the "Train" of Kubernetes etcd provides a reliable distributed database as a cluster of machines. TiKV is a high performance distributed transactional key-value store written in Rust.



9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.



2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for defining, managing, and running workflow definitions, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops



4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general-purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.



8. STREAMING & MESSAGING

When you need higher performance than JSON/REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.



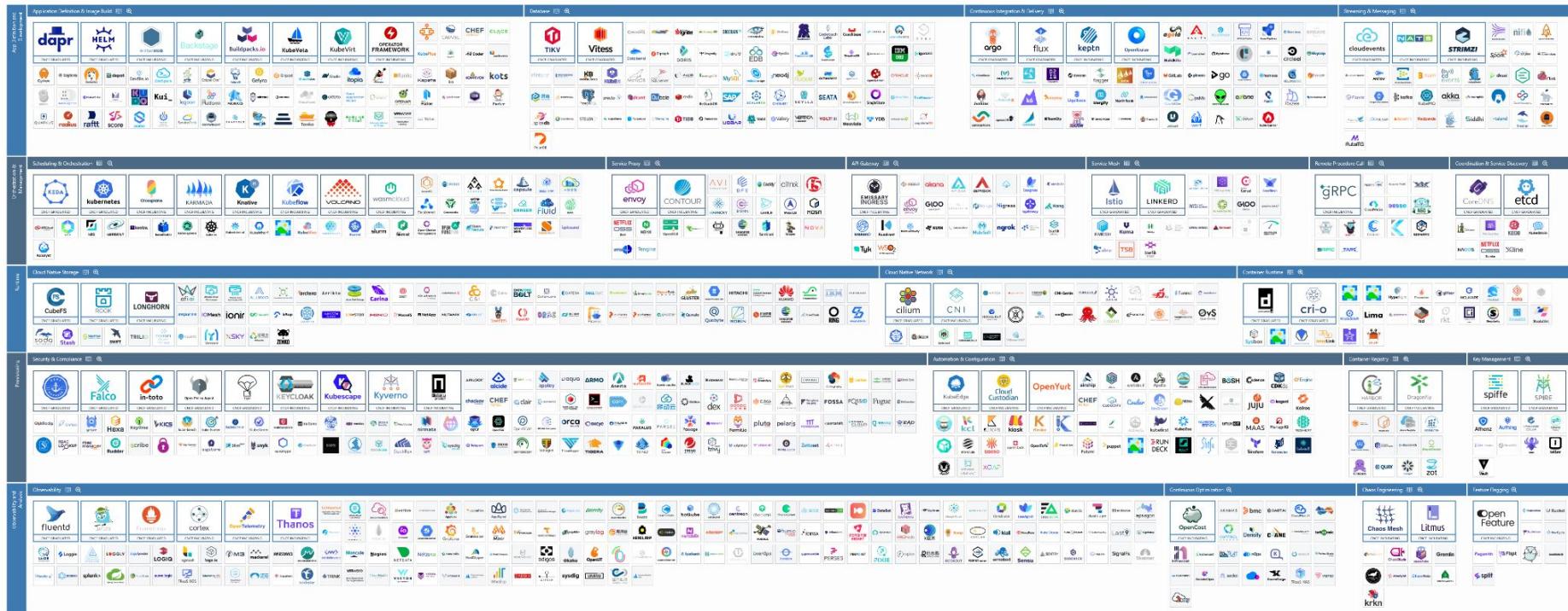
10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



Filters: GLOBE Projects and products Members Certified partners and providers Services Wasm CNCF XRAY MTS Card 790K+ - +

Type... to search items



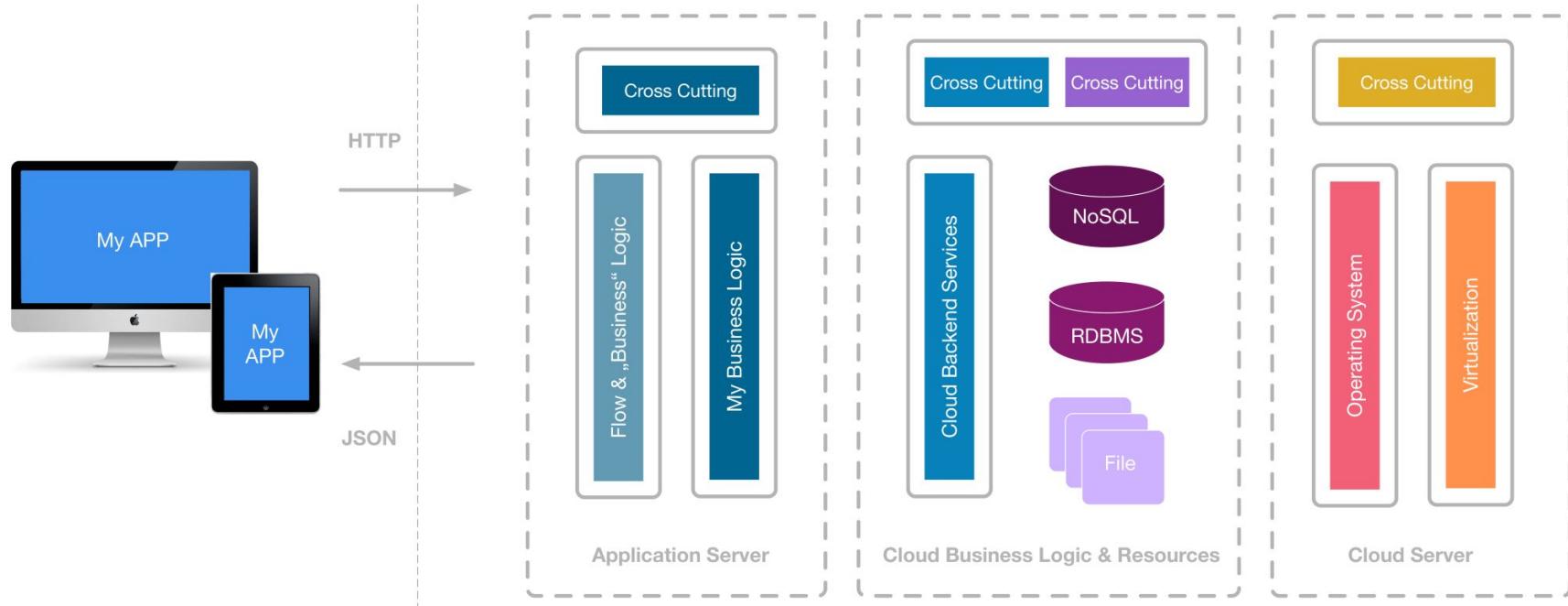
The background of the slide features a complex, abstract network graph composed of numerous small, semi-transparent white dots connected by thin gray lines, creating a sense of a vast, interconnected system.

Cloud Native ist

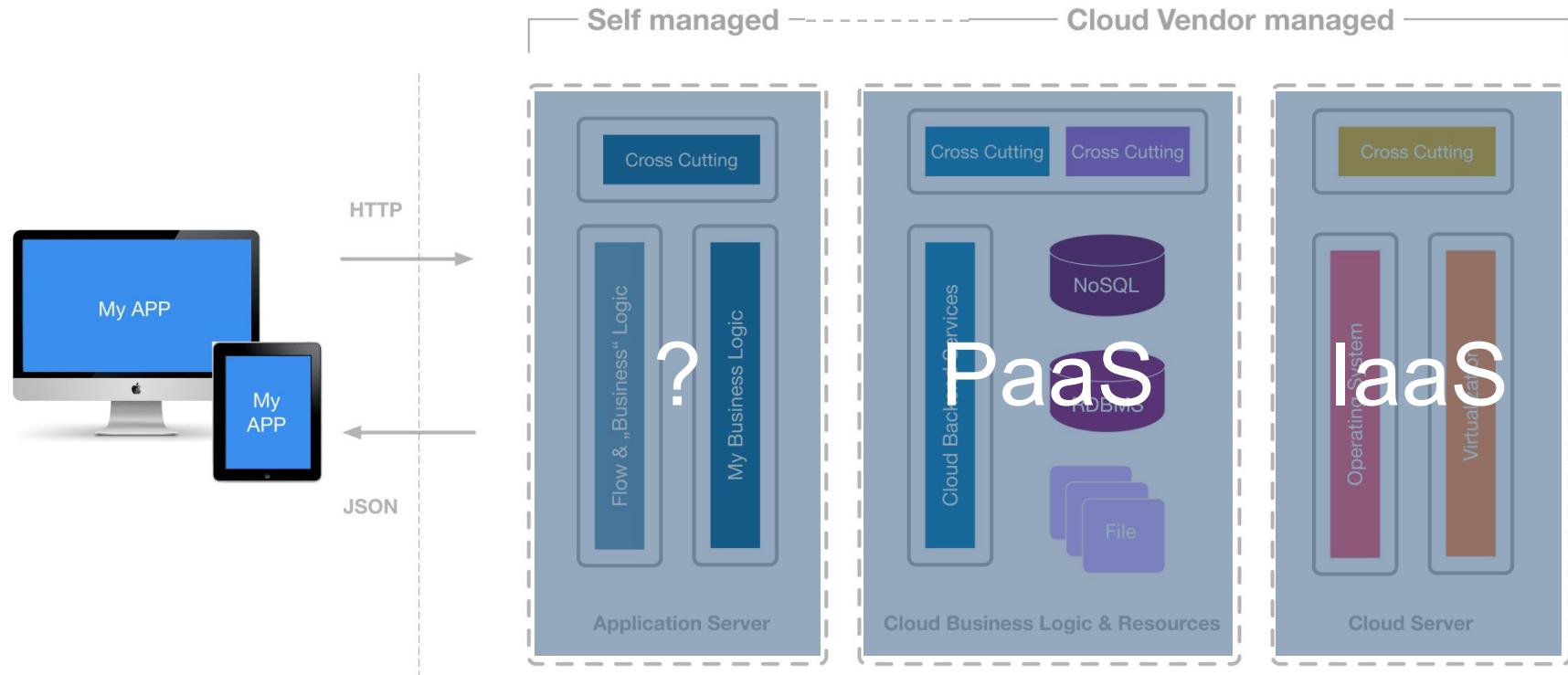
komplex!

Kubernetes, Docker, Microservices, YAML, ...

Traditionelle Cloud-basierte Anwendungsarchitektur



Traditionelle Cloud-basierte Anwendungsarchitektur



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.

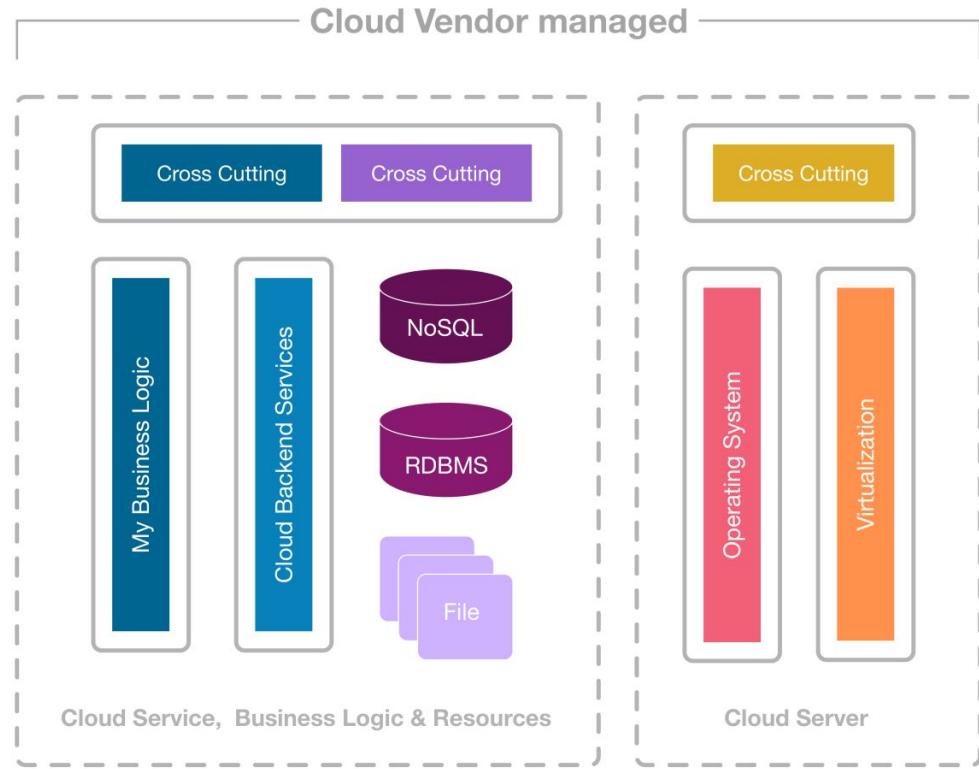
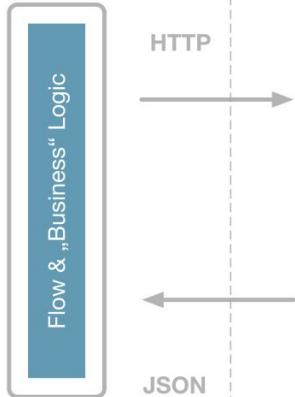
Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. It leverages a finer-grained deployment model

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.

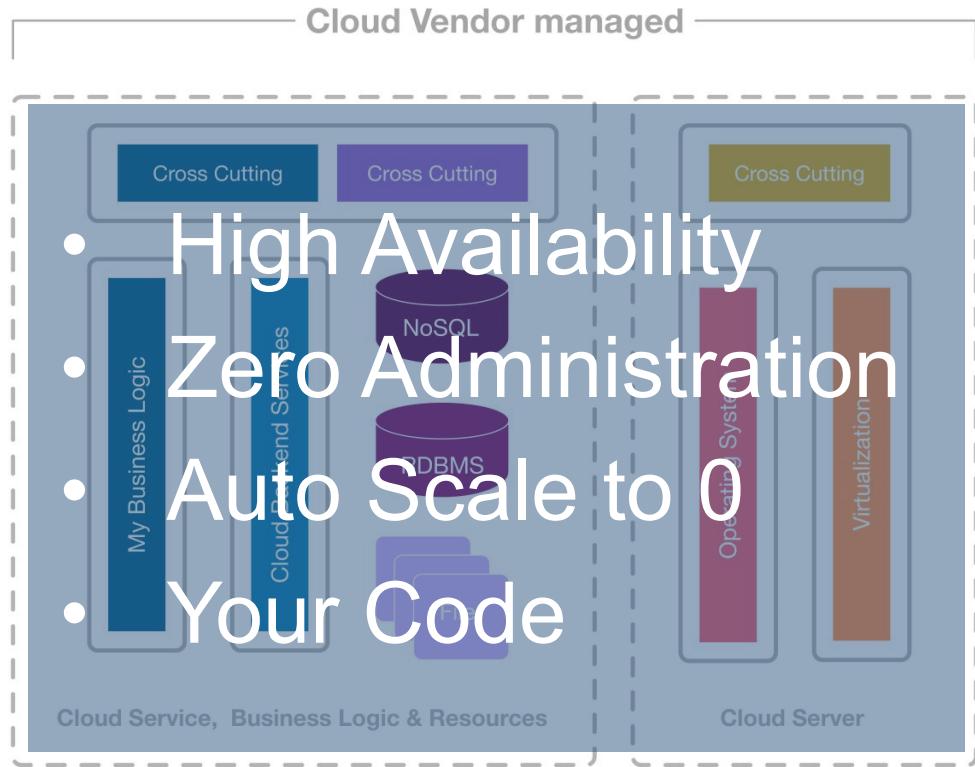
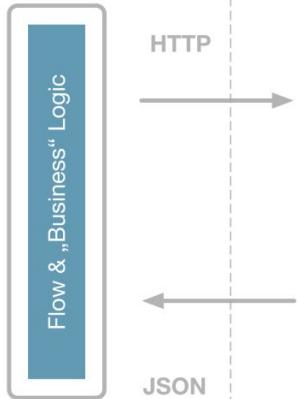
Serverless – Anwendungsarchitektur

Run Code, not Servers!

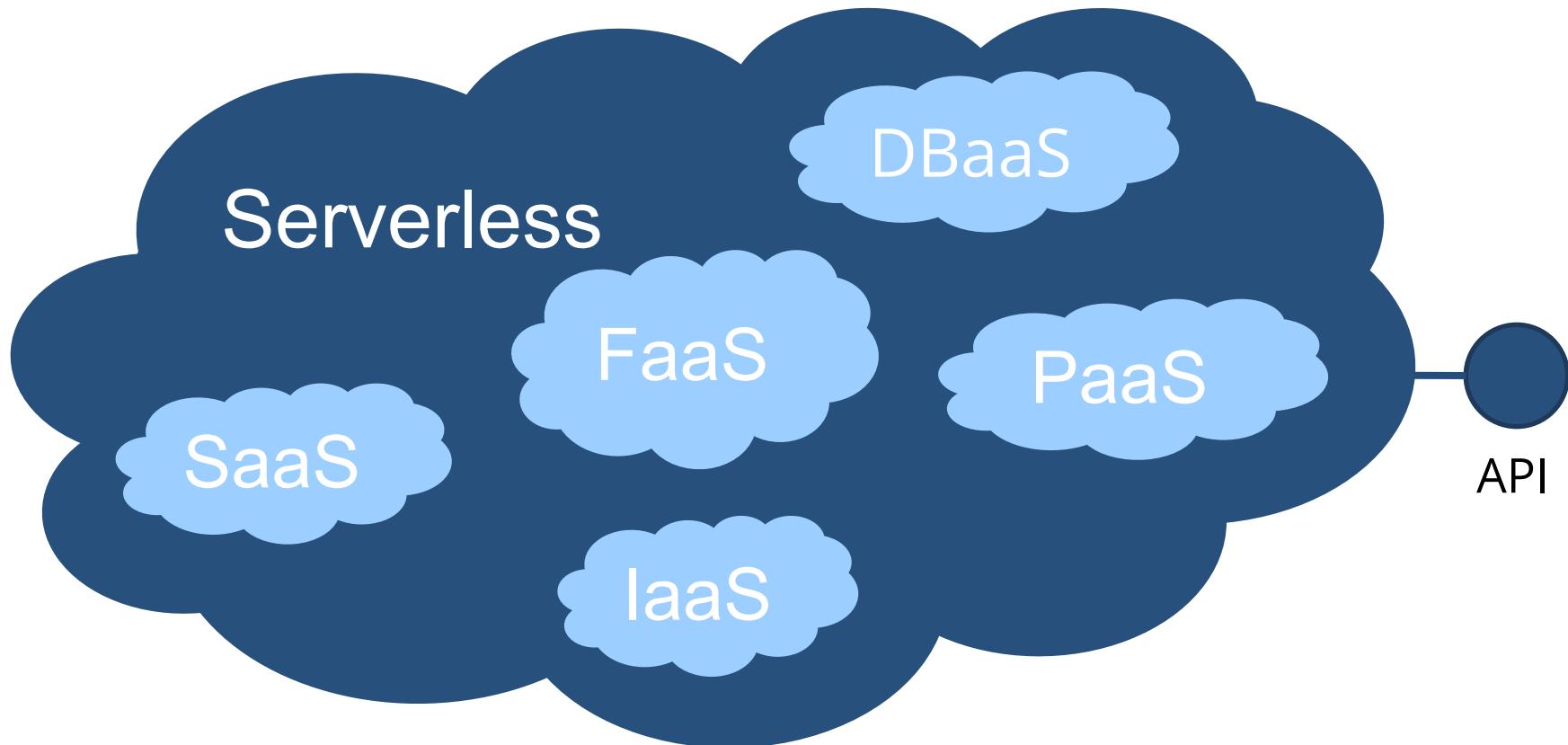


Serverless – Anwendungsarchitektur

Run Code, not Servers!



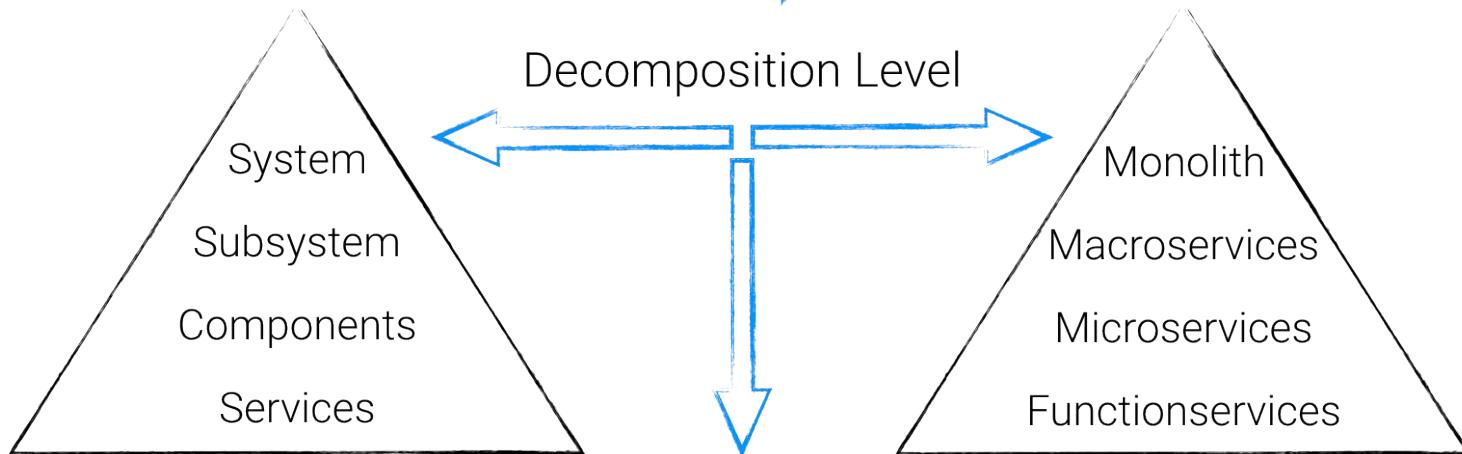
Out-of the Box Self-scaling Fully Managed Backend



Dev Components



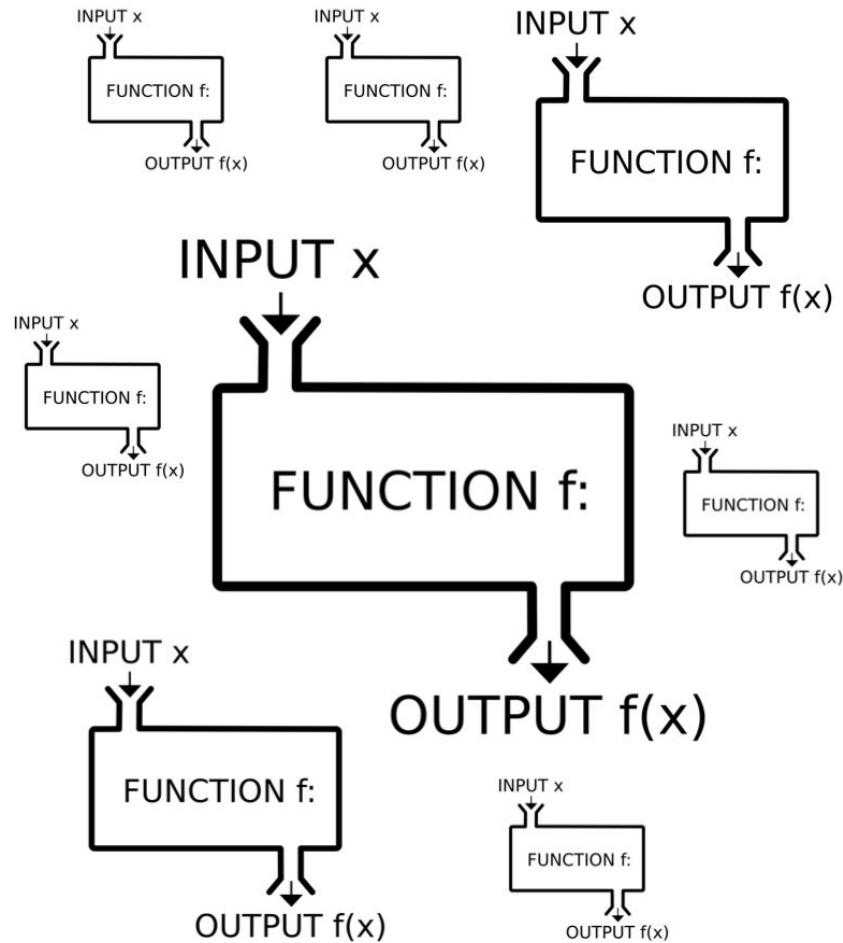
Ops Components



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 |
|--|---|

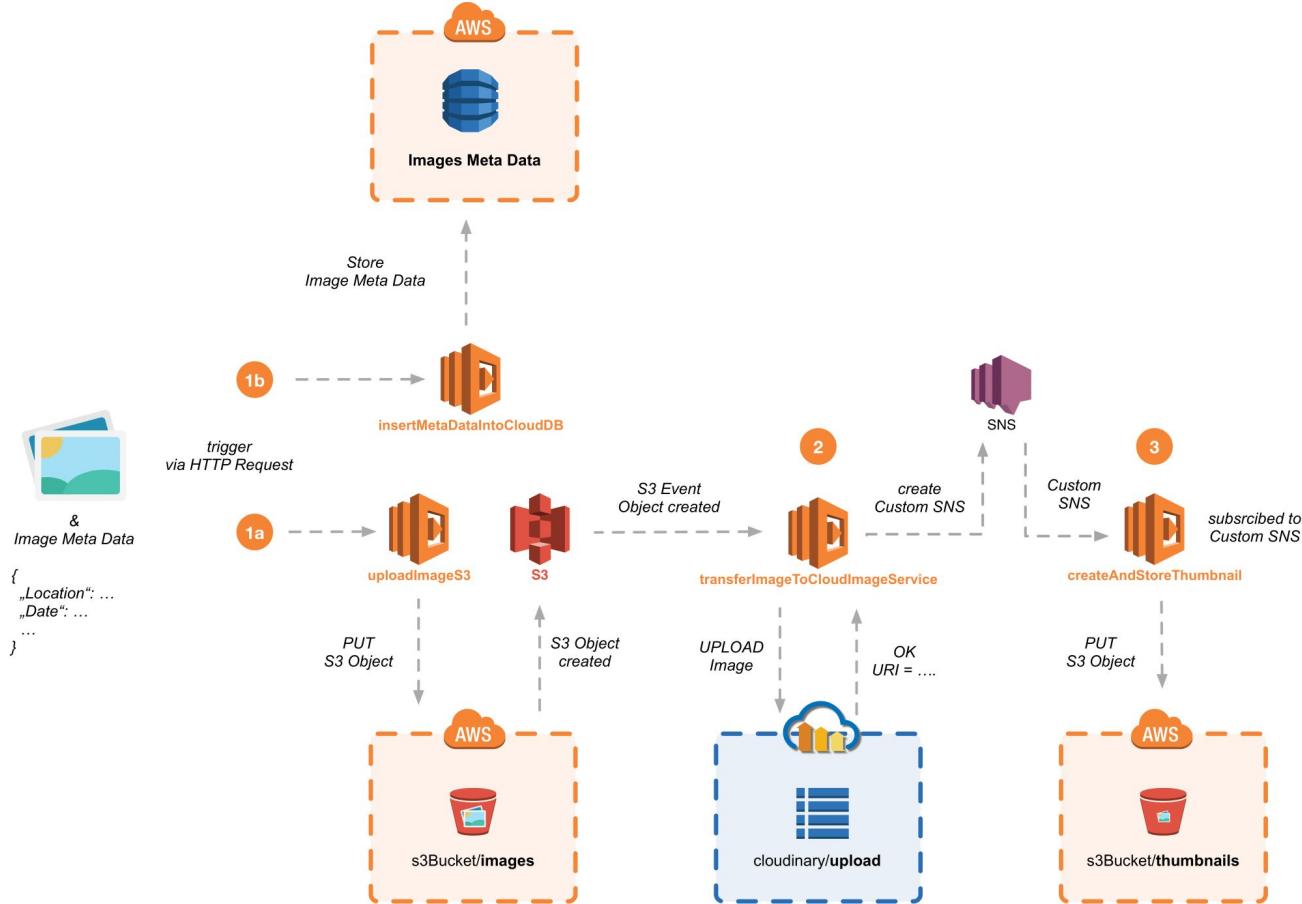
Serverless-Programmiermodell: Functions (as a Service) = FaaS



The background of the slide features a complex, abstract network graph. It consists of numerous small, semi-transparent white dots connected by thin, light gray lines, forming a dense web of triangles and polygons. This pattern repeats across the entire slide, creating a sense of digital connectivity and complexity.

FaaS ist perfekt für Event-getriebene Architekturen -
es ermöglicht
lose gekoppelte, reaktive
Komponenten oder Services.

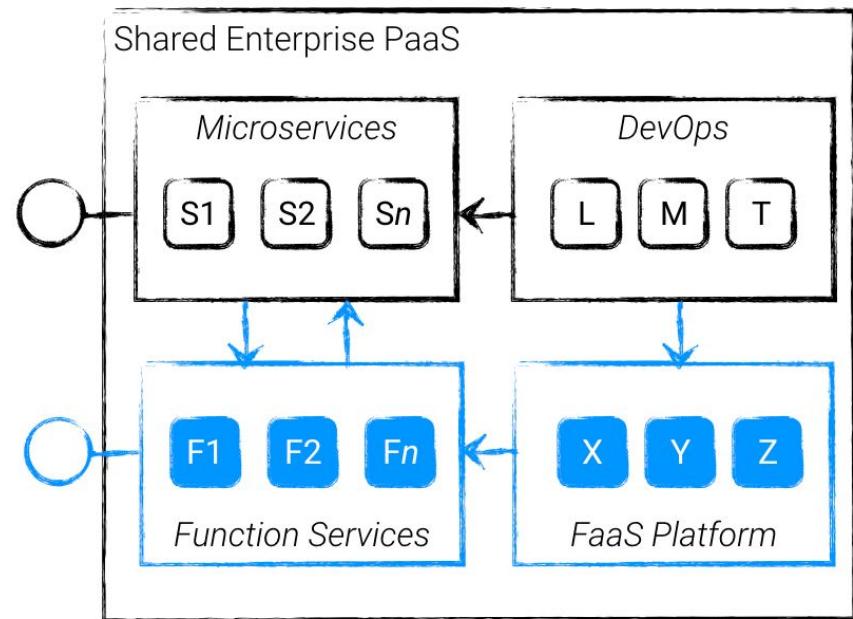
Thumbnails erstellen à la AWS Lambda



Use Case 1

Hybrid Architectures

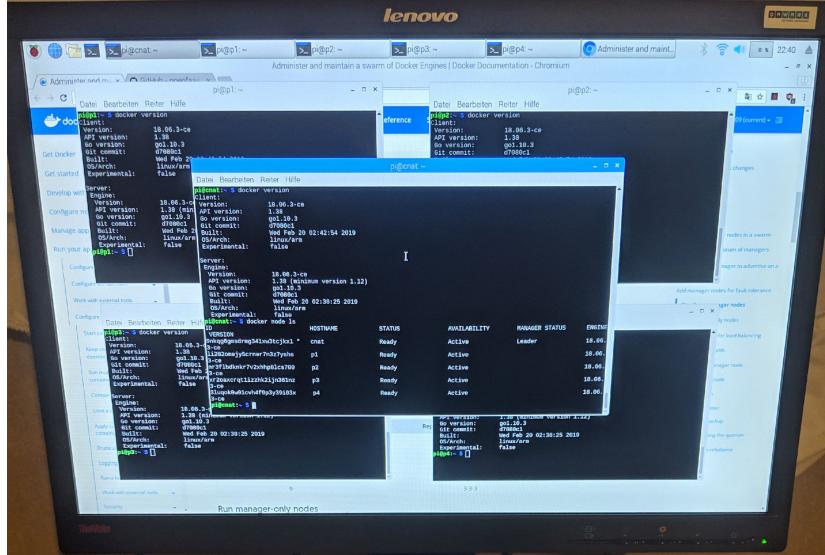
- Kombination von Microservice Architektur mit EDA
- Nutzung von Function Services für Event-getriebene Use Cases
- Reduzierter Ressourcen-Verbrauch per Scale-to-Zero
- Integration in bestehende Enterprise PaaS Umgebung



Use Case 2

Edge und Fog Computing

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



Self-hosted Serverless

Scale to Zero - still react to Events!

Filters

GROUP: Projects and products | Members | Certified partners and providers

Serverless | Wasm | CNAI

VIEW MODE

Grid

Card

ZOOM: - +

Serverless

Installable Platform



Tools



Installable Platforms



Hosted Platform



Embedded Functions



Framework



Security

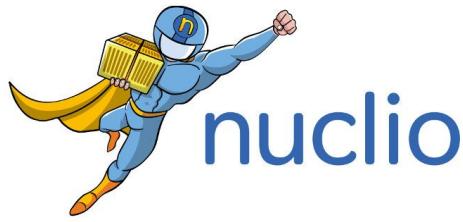




OPENFAAS



fission



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

Die Kandidaten

- OpenWhisk
<https://openwhisk.apache.org/>
- OpenFaaS
<https://www.openfaas.com>
- Fission
<https://fission.io>
- Nuclio
<https://nuclio.io>
- Knative
<https://knative.dev/>
- Keda
<https://keda.sh/>

LANGUAGE	USE CASES	GENERATION	PLATFORMS	RUNTIMES	TRIGGERS
FISSION	GO	ENTERPRISE	2ND	K8S	GO, PYTHON, NODEJS, JAVA/JVM [REDACTED]
OPENFAAS	GO	ENTERPRISE, IOT	1ST	K8S, DOCKER	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

Knative



Knative is a platform-agnostic solution for running serverless deployments.

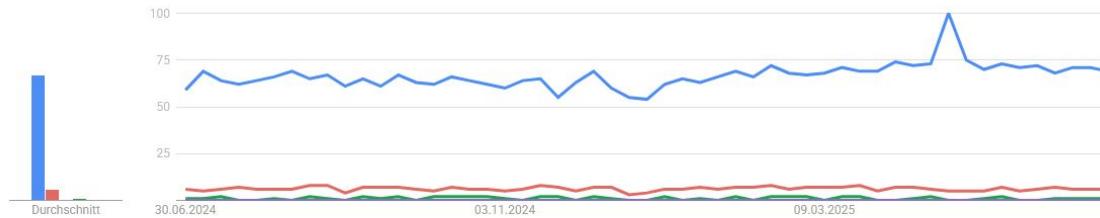
- FaaS: Knative Functions
 - Eine Vielzahl von Programmiersprachen wird hier durch Language Packs unterstützt:
<https://github.com/knative/func/blob/main/docs/language-packs/language-pack-contract.md#built-in-language-packs>
- Auto Scaling: Knative Serving
- EDA: Knative Eventing

KEDA

- Kubernetes Autoscaling/Event Platform
- Kann für EDAs verwendet werden
- Scaling/Erzeugen von Containern ist mit einer Vielzahl von Triggern möglich: Prometheus Metriken, Queues (SQS, Kafka, ...), Databases (Postgres, MySQL, ...)



Interesse im zeitlichen Verlauf ?



Aufschlüsselung nach Region



- Keda
- Knative
- Apache OpenWhisk
- openfaas
- Fission Serverless

Sortieren: Interesse an "Keda" ▾



1	Honduras	
2	Guatemala	
3	Nicaragua	
4	Malaysia	
5	Kuba	

Die Farbintensität gibt die Prozentzahl der Suchanfragen an [WEITERE INFORMATIONEN](#)

◀ 1–5 von 83 Regionen werden angezeigt ▶

Regionen mit geringem Suchvolumen einschließen