



DIE GROSSE CLOUD-NATIVE FAAS HITPARADE

// #CONTAINERCONF 2019 // DIE GROßE CLOUD-NATIVE FAAS HITPARADE -> { CREATED WITH ❤️ AND ☕ BY @LEANDERREIMER #CLOUDNATIVERD @QAWARE }



MARIO-LEANDER REIMER
PRINCIPAL SOFTWARE ARCHITECT
#CLOUDNATIVERD
QAWARE GMBH

[HTTP://GITHUB.COM/LREIMER](http://github.com/lreimer)
[HTTP://SPEAKERDECK.COM/LREIMER](http://speakerdeck.com/lreimer)

DISCLAIMER

THIS PRESENTATION IS MEANT TO PROVIDE AN OVERVIEW OF THE CURRENT **CLOUD NATIVE FAAS PLATFORM** ECOSYSTEM.

IT IS NOT MY INTENTION TO DISCREDIT THE GREAT WORK OF THE PEOPLE AND DEVELOPERS BEHIND THESE PLATFORMS.

CLOUD NATIVE SOFTWARE DEVELOPMENT IS
COMPLICATED.

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

CLOUD NATIVE DEVEX

IMPROVE YOUR INNER DEVELOPMENT LOOP

SERVERLESS IS THE NEXT
LOGICAL **EVOLUTION**
IN CLOUD NATIVE SOFTWARE DEVELOPMENT.

A photograph of a server room with blue metal racks. The racks are filled with various electronic components, including power strips and bundles of blue and white cables. The lighting is bright, and the overall color scheme is dominated by the blue of the racks and the white of the cables.

NO SERVER IS EASIER TO MANAGE THAN NO SERVER.³

³ WERNER VOGELS, CTO, AMAZON

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 WHERE 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 WHERE APPLICATIONS, BUNDLED AS **ONE OR MORE FUNCTIONS**, ARE UPLOADED TO A **PLATFORM**

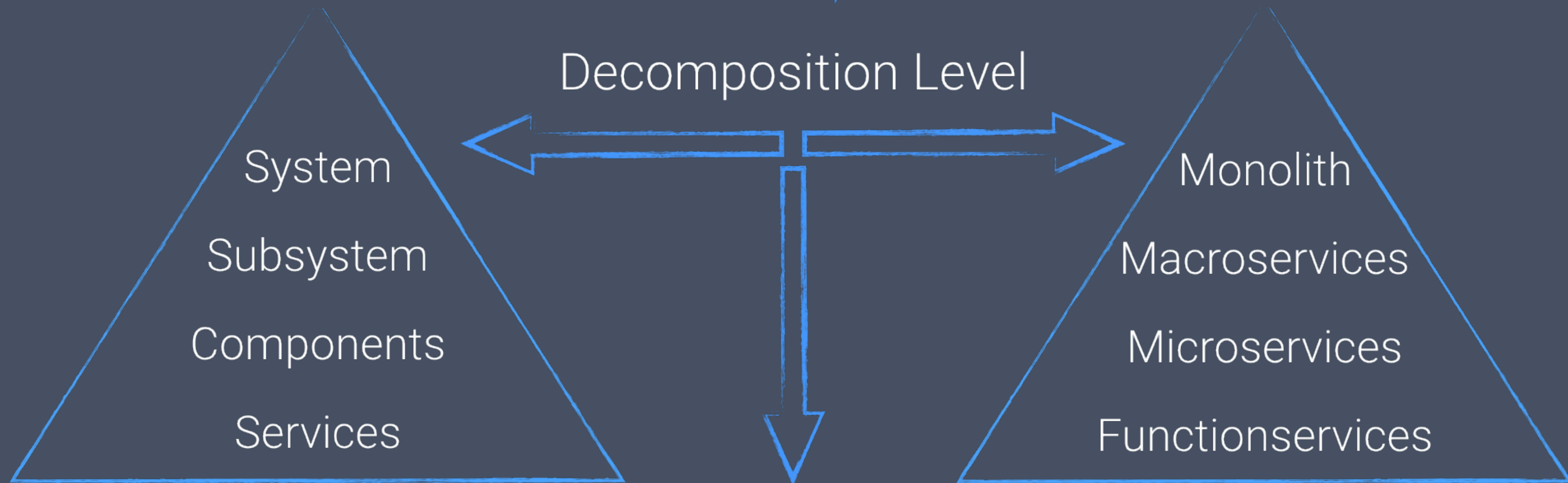
SERVERLESS COMPUTING REFERS TO A **NEW MODEL** OF CLOUD NATIVE COMPUTING, ENABLED BY **ARCHITECTURES** THAT DO **NOT REQUIRE SERVER MANAGEMENT** TO BUILD AND RUN APPLICATIONS. IT LEVERAGES A **FINER-GRAINED DEPLOYMENT** MODEL WHERE APPLICATIONS, BUNDLED AS **ONE OR MORE FUNCTIONS**, ARE UPLOADED TO A **PLATFORM** AND THEN **EXECUTED, SCALED, AND BILLED** IN RESPONSE TO THE **EXACT DEMAND** NEEDED AT THE MOMENT. ¹

¹[HTTPS://LANDSCAPE.CNCF.IO/](https://landscape.cncf.io/)

Dev Components



Ops Components



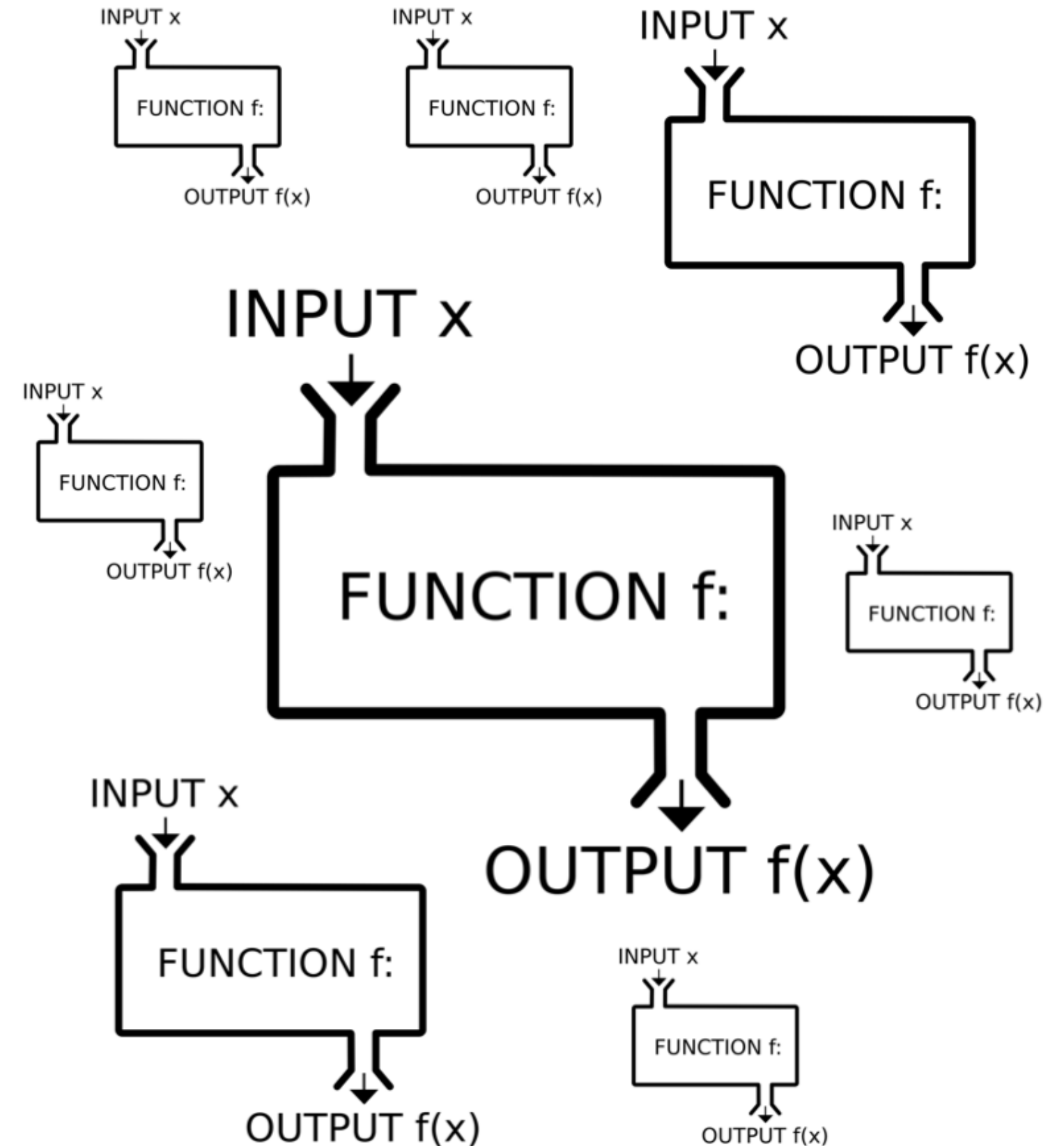
Decomposition Trade-Offs

- + More flexible to scale
- + Runtime isolation (crash, slow-down, ...)
- + Independent releases, deployments, teams
- + Higher resources utilisation

- Distribution debt: Latency, Consistency
- Increased infrastructure complexity
- Increased troubleshooting complexity
- Increased integration complexity

FUNCTIONS

AS PREFERRED SERVERLESS APPLICATION
PROGRAMMING MODEL



EVENT-DRIVEN ARCHITECTURES

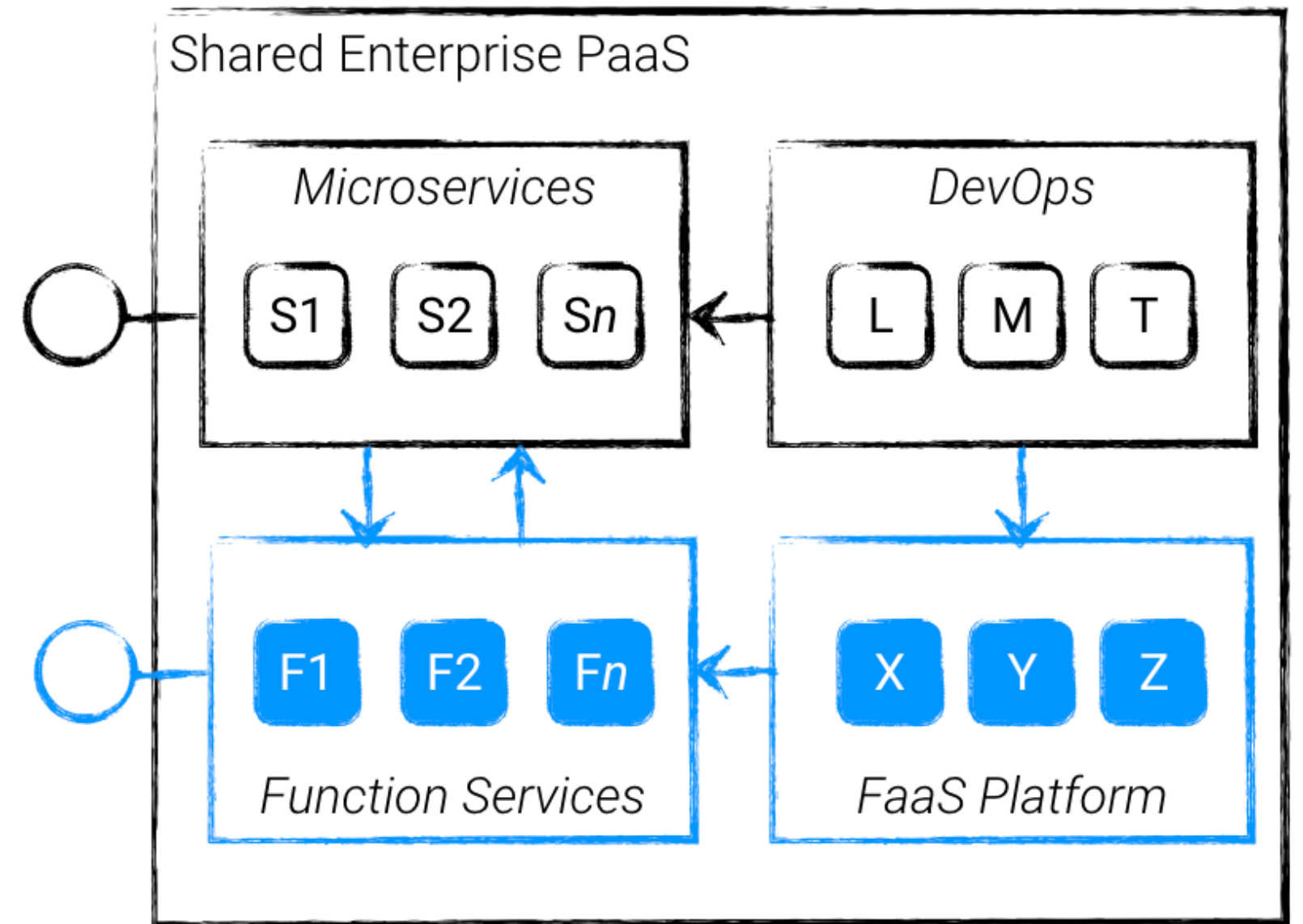
ENABLE LOOSELY COUPLED REACTIVE SOFTWARE COMPONENTS AND SERVICES.²

² [HTTPS://CLOUDEVENTS.IO](https://cloudevents.io)

MY USE CASE 1

HYBRID ARCHITECTURES

- > COMBINE MICROSERVICE ARCHITECTURE WITH EDA
- > LEVERAGE FUNCTION SERVICES FOR EVENT-DRIVEN USE CASES
- > REDUCE RESOURCE CONSUMPTION AND COSTS VIA SCALE 2 ZERO
- > INTEGRATE INTO EXISTING ENTERPRISE PAAS ENVIRONMENT



MY USE CASE 2

EDGE AND FOG COMPUTING

- COUCH PROJECT: EVALUATE USAGE OF FAAS FOR LOW POWER DEVICES
- SUPPORT FOR LIGHTWEIGHT SCHEDULERS LIKE DOCKER SWARM
- SUPPORT FOR COMPILED LANGUAGE RUNTIMES LIKE GO, ET.AL.



MY CRITERIA TO CHOOSE THE IDEAL FAAS PLATFORM

- › **OPEN SOURCE** – NO INITIAL \$\$\$ AND NO VENDOR LOCK-IN PLEASE
- › **MATURITY** – GOOD AND ACTIVE COMMUNITY, LITTLE ISSUES, FREQUENT RELEASES
- › **USE CASES** – GENERAL PURPOSE, ENTERPRISE, BIG DATA, AI, EDGE COMPUTING
- › **APPROACHABLE** – QUICK START AND SUFFICIENT DOCUMENTATION
- › **LANGUAGE RUNTIMES** – PLEASE MORE THAN JAVASCRIPT!
- › **DEVELOPER FRIENDLY** – TOOLS & FRAMEWORKS, LOCAL RUNTIME, TESTING, IDE SUPPORT
- › **OPERABILITY** – EASY SETUP, SUPPORTED PLATFORMS, TECHNOLOGY FOOTPRINT
- › **INTEGRATION** – SUPPORTED TRIGGERS, INFRASTRUCTURE, PLUGINS, STANDARDS
- › **PERFORMANCE** – GOOD COLD STARTUP PERFORMANCE AND THROUGHPUT

Tools



Security



Framework



Hosted

Installable

Platform



Cloud Native Landscape

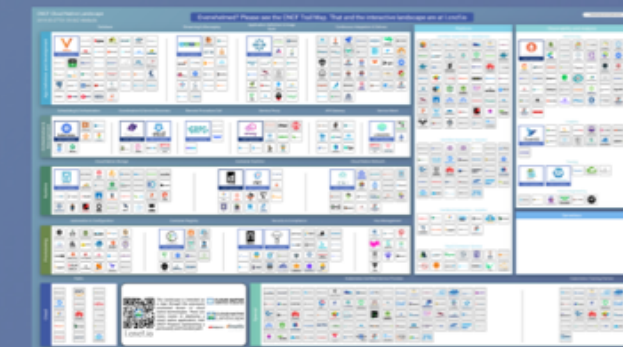


Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform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

CLOUD NATIVE
Landscape

CLOUD NATIVE
COMPUTING FOUNDATION

Redpoint



Tools



Security



Framework



Hosted

Installable

Platform



Cloud Native 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



Tools



Security



Framework



Hosted

Installable

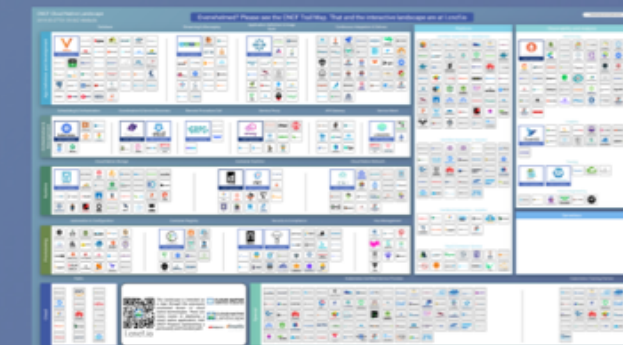
Platform



Cloud Native 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



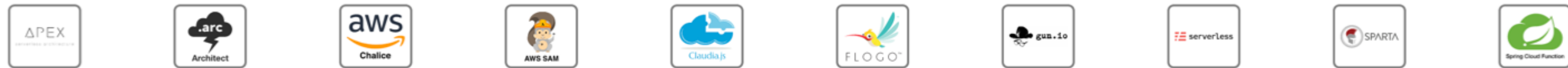
Tools



Security



Framework



Hosted

Installable

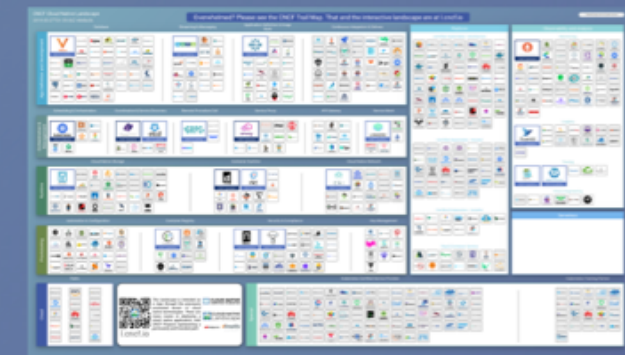
Platform



Cloud Native 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





MY **MAIN** CONTENDERS

- > **FISSION**
[HTTPS://FISSION.IO](https://fission.io)
- > **KNATIVE**
[HTTP://KNATIVE.DEV](http://knative.dev)
- > **KUBELESS**
[HTTPS://KUBELESS.IO](https://kubeless.io)
- > **NUCLIO**
[HTTPS://NUCLIO.IO](https://nuclio.io)
- > **OPENFAAS**
[HTTPS://WWW.OPENFAAS.COM](https://www.openfaas.com)
- > **KYMA**
[HTTP://KYMA-PROJECT.IO](http://kyma-project.io)

	LICENSE	USE CASES	PLATFORMS	RUNTIMES	TRIGGERS
FISSION	APACHE-2.0	ENTERPRISE	K8S	GO, PYTHON, NODEJS, JAVA/JVM	CRON, HTTP, NATS, AZURE QUEUE STORAGE, KAFKA, KUBEWATCH
KUBELESS	APACHE-2.0	ENTERPRISE	K8S	NODEJS, JAVA, GO, JVM, PYTHON, PHP, RUBY, .NET CORE, BALLERINA, VERT.X	CRON, HTTP, NATS, KINESIS, KAFKA
OPENFAAS OPENFAAS CLOUD	MIT	ENTERPRISE, IOT	K8S, DOCKER	GO, C#, JAVA8, JAVA12, DOCKERFILE, NODEJS 8.X, NODEJS 12.X, PHP7, RUBY, PYTHON 3	HTTP, CRON, KAFKA, AWS SNS, S3, CLOUDEVENTS, IFTTT, REDIS, MQTT, NATS
NUCLIO	APACHE-2.0	ENTERPRISE, AI	DOCKER, K8S, AWS, GCP	.NET CORE, GO, JAVA, NODEJS, PYTHON, SHELL	CRON, EVENTHUB, HTTP, KAFKA, KINESIS, NATS, RABBITMQ, MQTT
KNATIVE	APACHE-2.0	ENTERPRISE	K8S	LANGUAGE AGNOSTIC	CRONJOB, API SERVER, GITHUB, GCP PUB/SUB, KAFKA, AWS SQS, HTTP
KYMA	APACHE-2.0	ENTERPRISE	K8S	JAVA, GO, NODEJS, PYTHON, RUBY	HTTP, NATS, KNATIVE EVENTING



CLOUD NATIVE **FAAS** DEMOS



WHICH ONE IS BEST?

IT DEPENDS
ON YOUR USE CASE!

FORK ME

[HTTPS://GITHUB.COM/LREIMER/THE-BIG-FAAS-LEBOWSKI](https://github.com/lreimer/the-big-faas-lebowski)

[HTTPS://GITHUB.COM/LREIMER/RASPI-SWARM-BOX](https://github.com/lreimer/raspi-swarm-box)

THANK YOU!