



Innovate Summit 2017

CNCF & State of Serverless

Lee Calcote & Girish Ranganathan

innovate.solarwinds.io



Lee Calcote



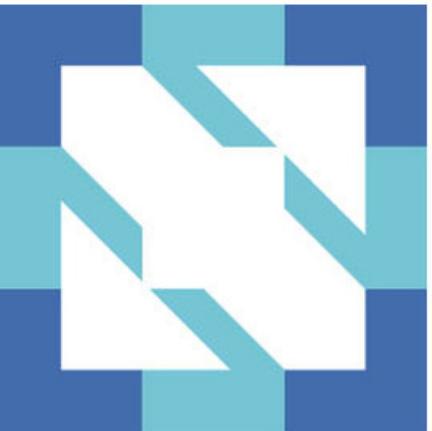
@lcalcote



Girish Ranganathan



Dr.



**CLOUD NATIVE
COMPUTING FOUNDATION**



What is the CNCF?

- Foster growth and evolution of ecosystem
- Promote underlying technologies
- Provide stewardship for projects
- Make technologies accessible and reliable

CNCF Members



CNCF Projects



Kubernetes
Orchestration



Prometheus
Monitoring



OpenTracing
Distributed Tracing API



Fluentd
Logging



linkerd
Service Mesh



gRPC
Remote Procedure Call



CoreDNS
Service Discovery



containererd
Container Runtime



rkt
Container Runtime



CNI
Networking API



Envoy
Service Mesh



Jaeger
Distributed Tracing



Notary
Security

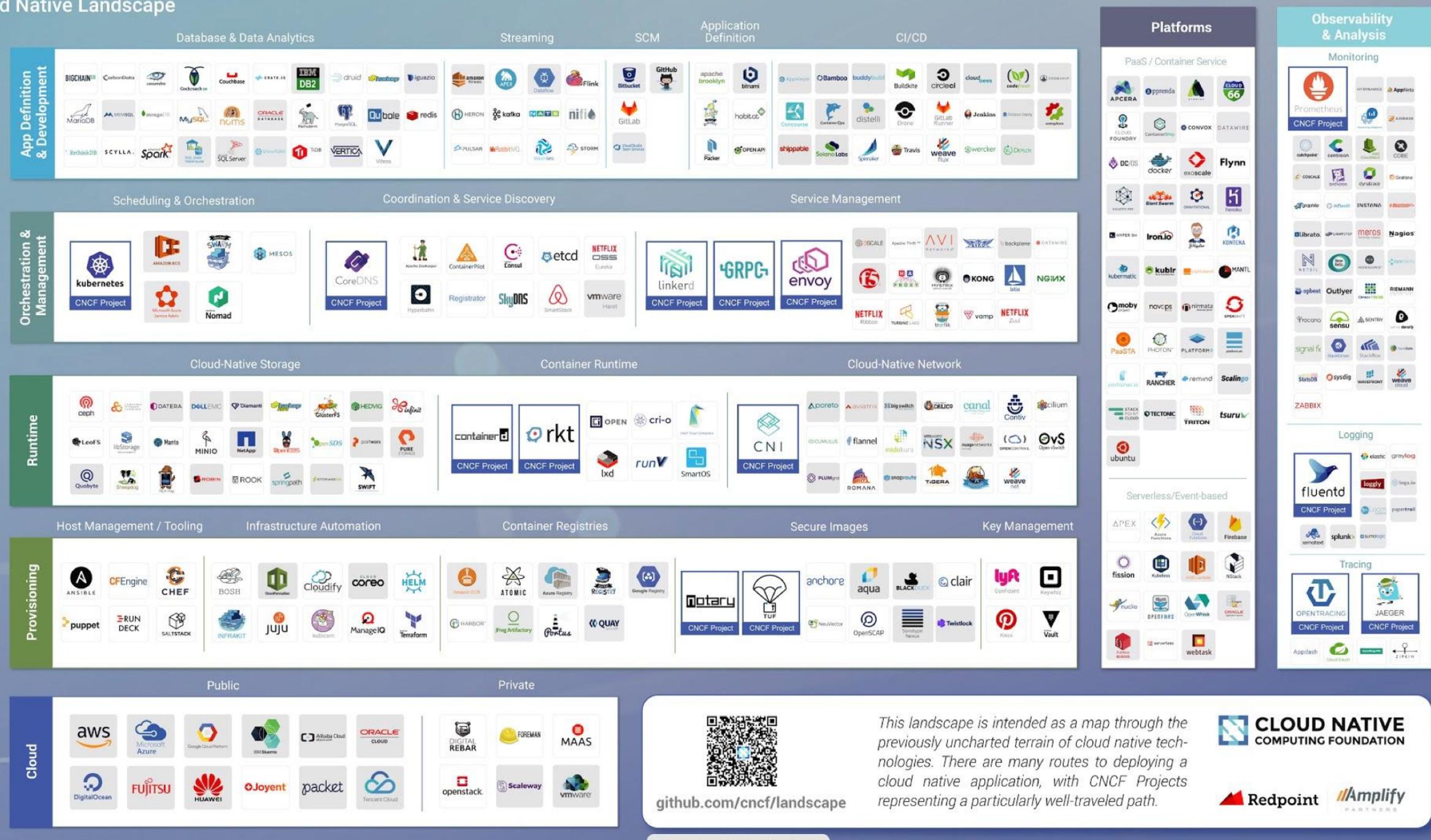


TUF
Software Update Spec



CNCF Landscape

Cloud Native Landscape



CNCF Working Groups

Continuous Integration

Provides infrastructure to hosted projects.

Looks to offer integration testing between projects.

Networking

Providing a Container Networking Interface (**CNI**) specification.

Aims for connectivity and portability in cloud native application networking.

Storage

Providing a Container Storage Interface (**CSI**) specification.

Aims for portability across cloud orchestration systems.

Serverless

Educate cloud native developers on serverless architectures.

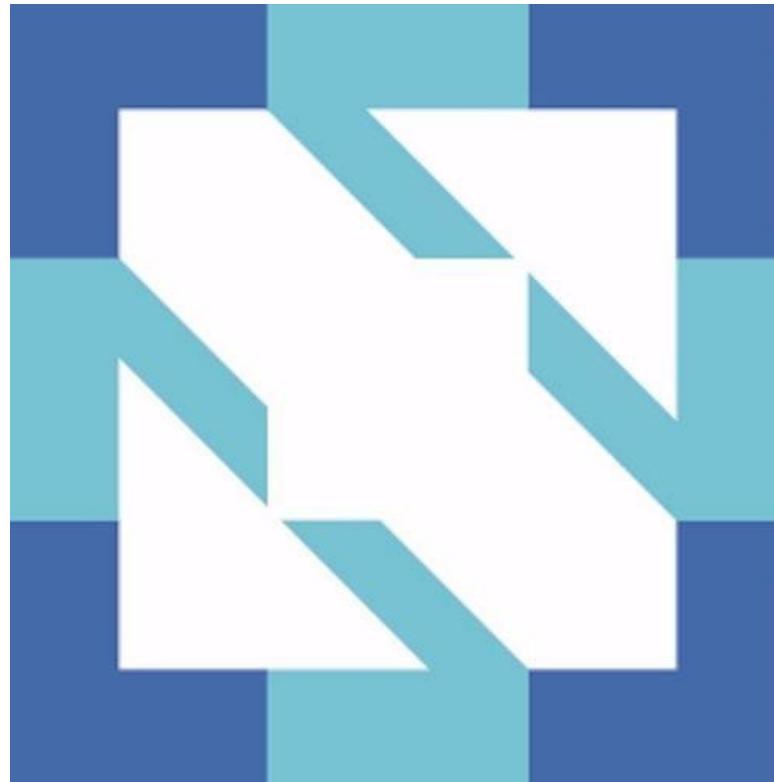
Determine what the CNCF should do in this space.

Recommend involvement in specifications and projects.

Solarwinds Participation

- Our products included in Cloud Native Landscape.
- Our works highlighted in CNI project adoption.
- We are on the OpenTracing Specification Council.
- SolarWinds working within Serverless WG.
 - Published Serverless white paper. Creating OpenEvents.
- Founding the OpenMetrics working group.
- Integrating with Istio and OpenCensus.





Serverless WG

Serverless WG

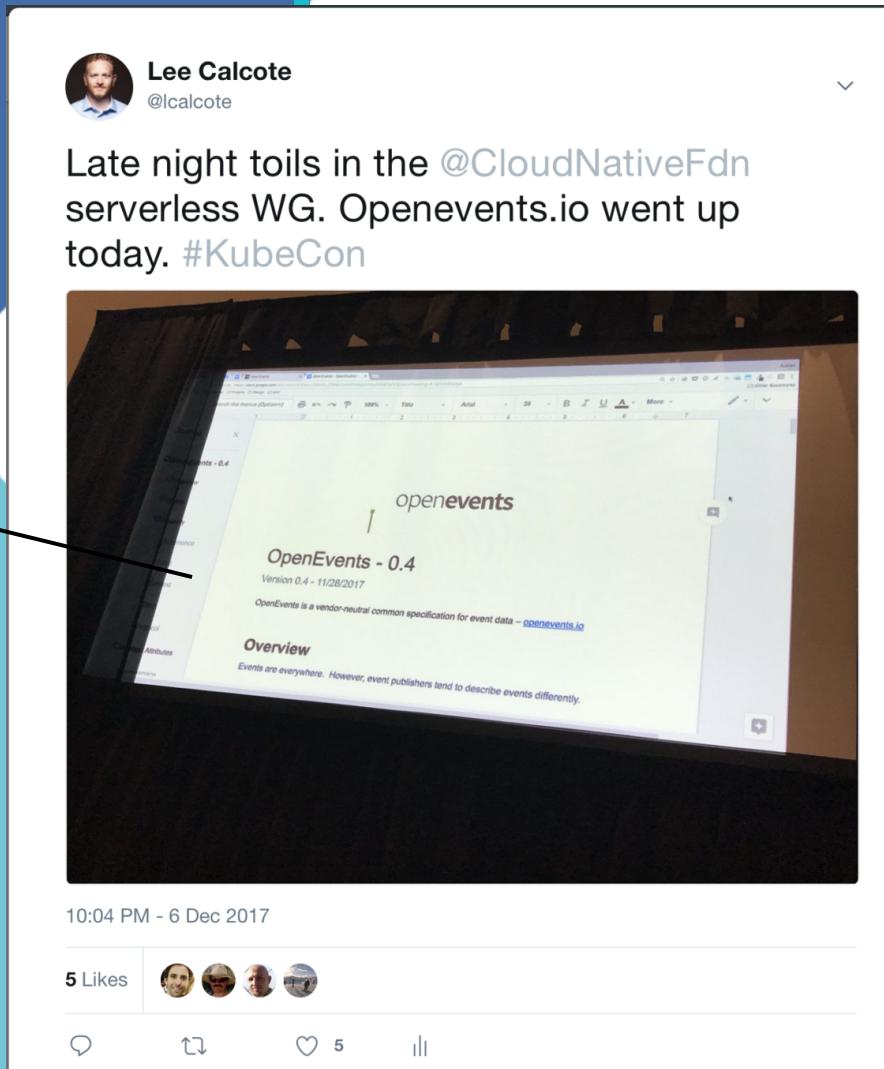
- Formed in June 2017 at the request of CNCF Technical Oversight Committee (TOC)
- Asked for state of tech/community & recommendations for possible involvement
- Most key Serverless players involved
- IBM, VMWare, Google, Red Hat, Huawei, Microsoft, **SolarWinds**, Docker, iguazio, Amazon, MasterCard, Pivotal, Serverless Inc., Clay Labs, The New Stack, A Cloud Guru, Platform9, Bitnami, Auth0, and Hyper

Serverless WG: White Paper

- Describes & defined Serverless
- Highlights promising use cases and areas where they've already proven value
- Differentiates Serverless from PaaS and Container Orchestration
- Describes the mechanics of a generic Serverless system
- Identifies potential future "harmonization" the WG could look at

Serverless WG: Event Specification

OpenEvents.io



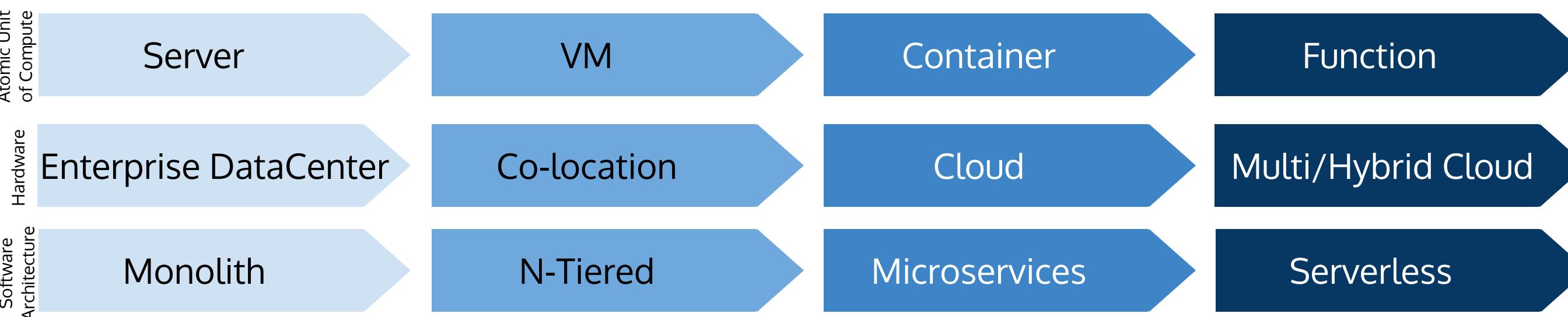
event data, similar to CNI and
at between a few proposals:
ing (CNEM) – (iguazio)
s, Inc.)
eration – (IBM, DMTF)

Serverless WG: Other Areas of focus

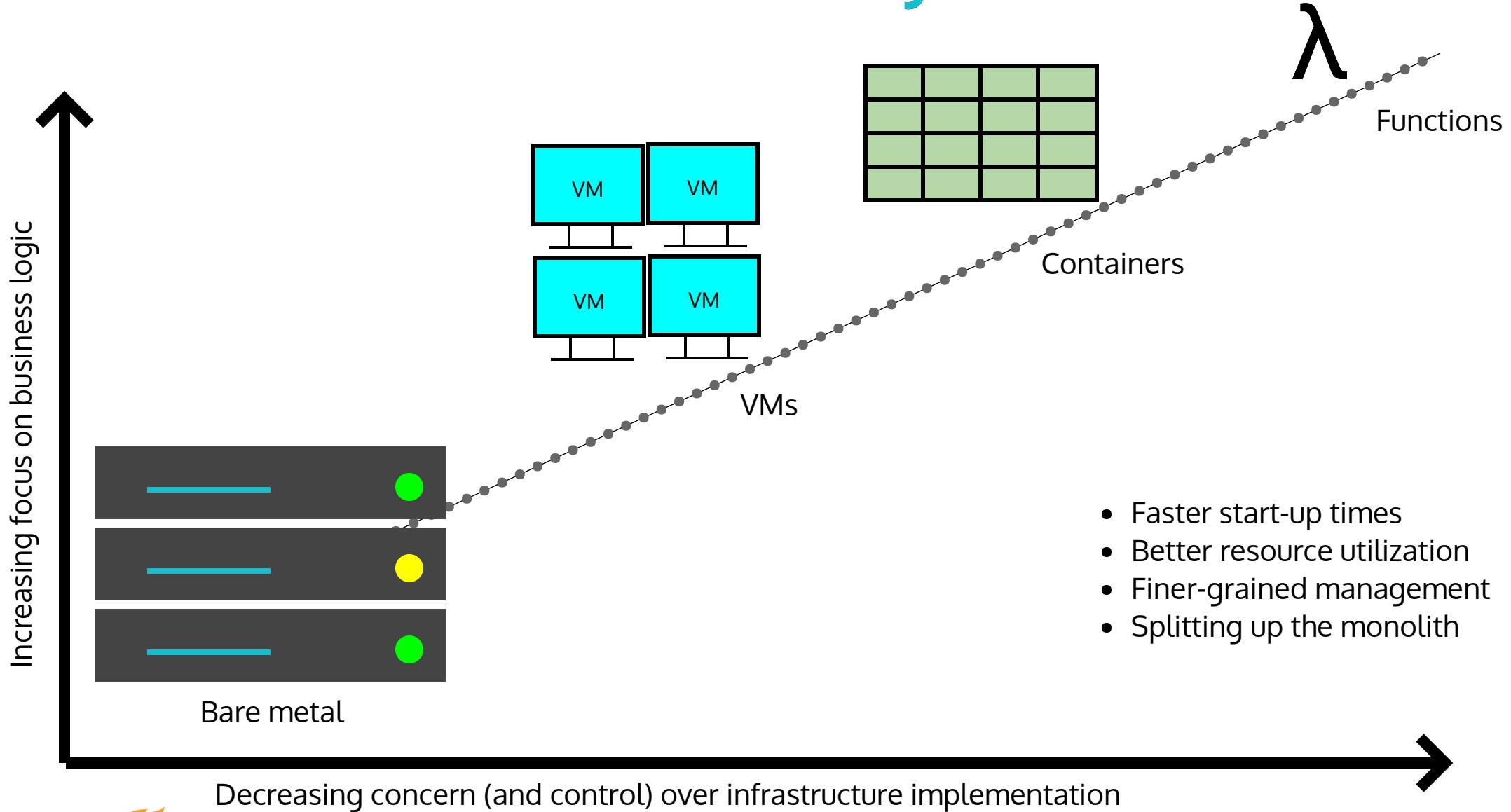
- Maintain a matrix and landscape of serverless implementations and features.
- Produce additional documents and samples that educate users on serverless apps.
- Document integration with other CNCF projects, such as how to monitor and observe.
- Potential collaboration on a declarative packaging format specification.

Functions as a Service

Transitional phases to Serverless



But why?



The Promised Land

No compute cost when idle.

Flexible and precise scaling.

No provisioning, updating, and managing server infrastructure.

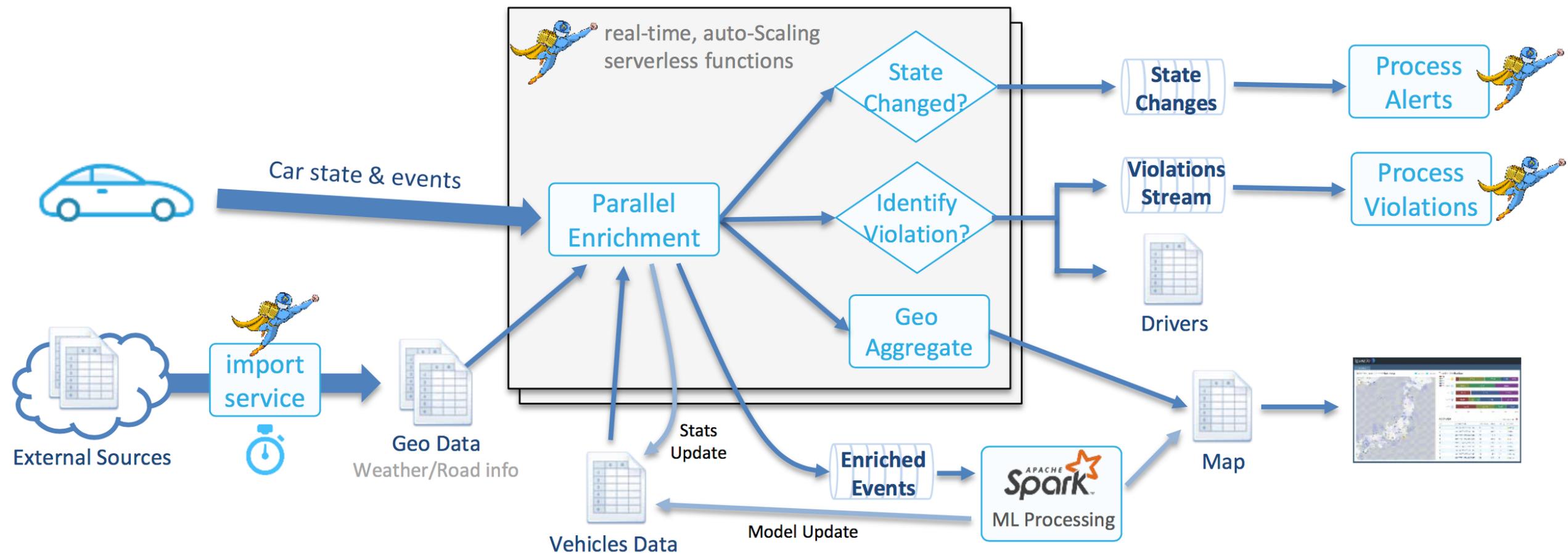
When to use...

consider serverless when your workload is:

- Asynchronous, concurrent, easy to parallelize into independent units of work
- Infrequent or has sporadic demand, with large, unpredictable variance in scaling requirements
- Stateless, ephemeral, without a major need for instantaneous cold start time
- Highly dynamic in terms of changing business requirements that drive a need for accelerated developer velocity

IoT & Stream Analysis

Common Use Cases

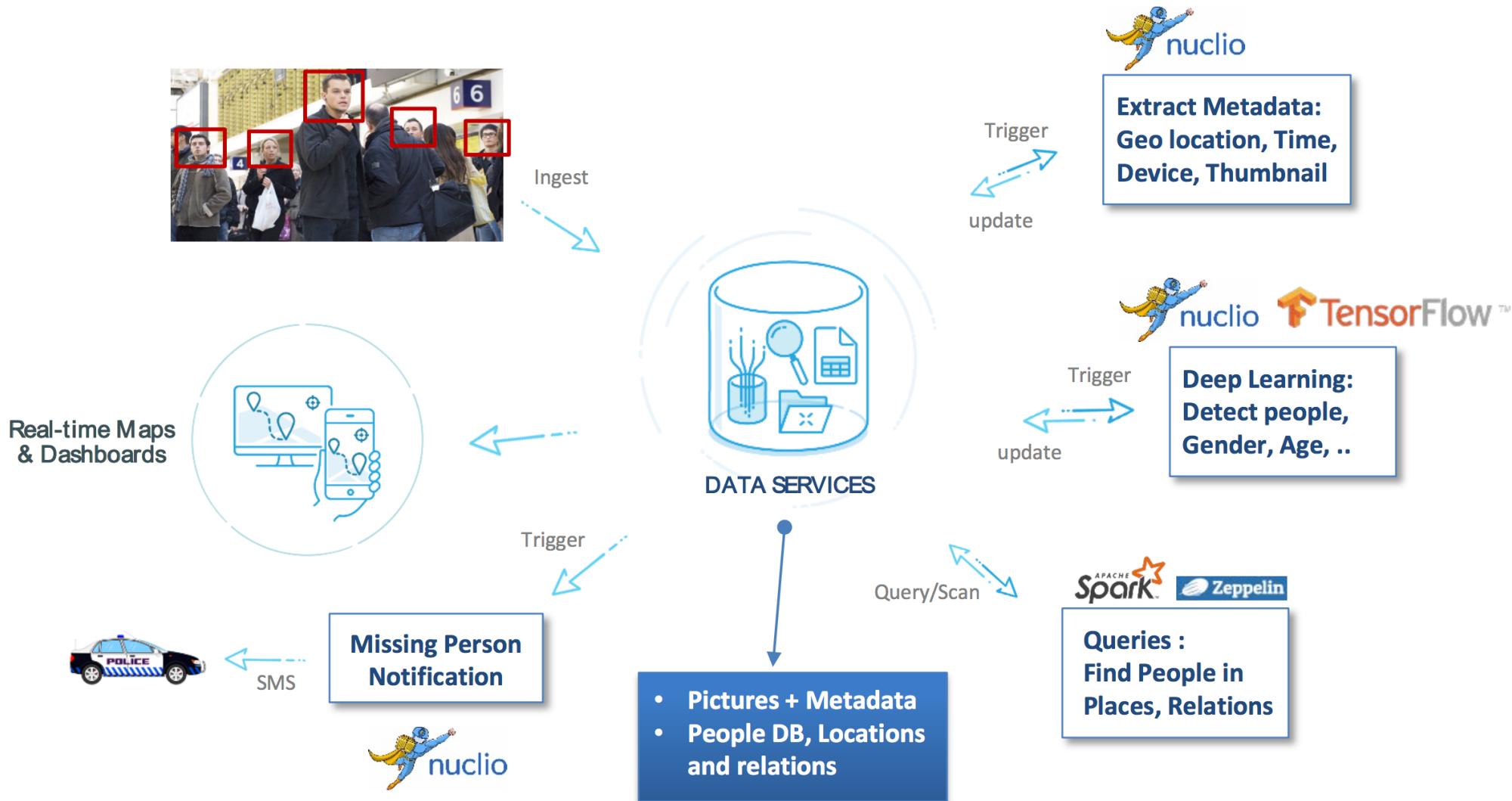


Example workloads

some of these include:

- Executing logic in response to **database changes** (insert, update, trigger, delete)
- Performing analytics on **IoT sensor** input messages (such as MQTT messages)
- Handling **stream processing** (analyzing or modifying data in motion)
- Managing single time extract, transform, and load jobs that require a lot of processing for a short time (**ETL jobs**)
- Providing cognitive computing via a **chat bot** interface (asynchronous, but correlated)
- Scheduling **tasks performed for a short time**, such as cron or batch style invocations
- Serving **machine learning** and **AI models** (retrieving one or more data elements such as tables, NLP, or images and matching against a pre-learned data model to identify text, faces, anomalies, etc.)
- **Continuous integration pipelines** that provision resources for build jobs on-demand, instead of keeping a pool of build slave hosts waiting for jobs to be dispatched

Surveillance - Common Use Cases



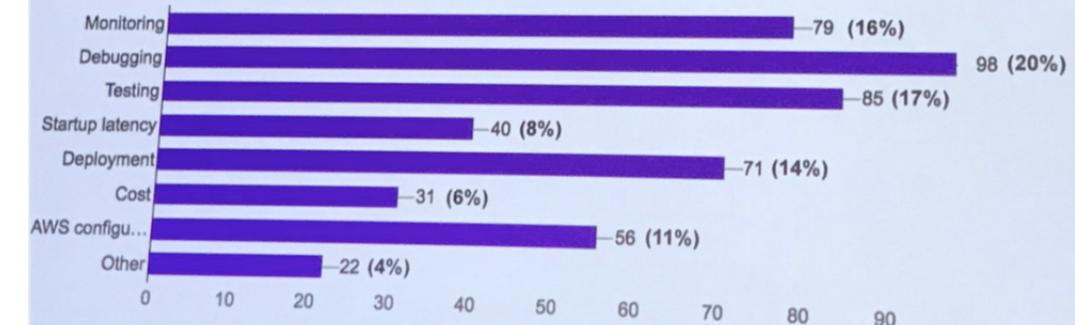
Serverless Pain Points



Lee Calcote
@lcalcote

Current pain points experienced in developing #serverless architectures.
@ServerlessConf #ServerlessConf

Which of the following are serious pain points for you in developing serverless architectures?
(175 responses)



9:48 AM - 27 Apr 2017 from Austin, TX

42 Retweets 36 Likes



1



42



36



Primary Personas

1. **Developer:** writes code for, and benefits from, the serverless platform which provides them the point of view that there are no servers nor that their code is always running.
2. **Provider:** deploys the serverless platform for an external or internal customer.

Serverless = No Ops?

Servers are still required to run a serverless platform.

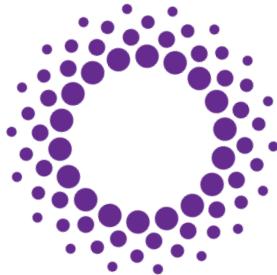
The **provider** will need to manage servers.

And will have some cost for running the platform, even when idle.

A self-hosted system can still be considered serverless:

typically one team acts as the **provider** and another as the **developer**.

FaaS Landscape



fission



APACHE
OpenWhisk™



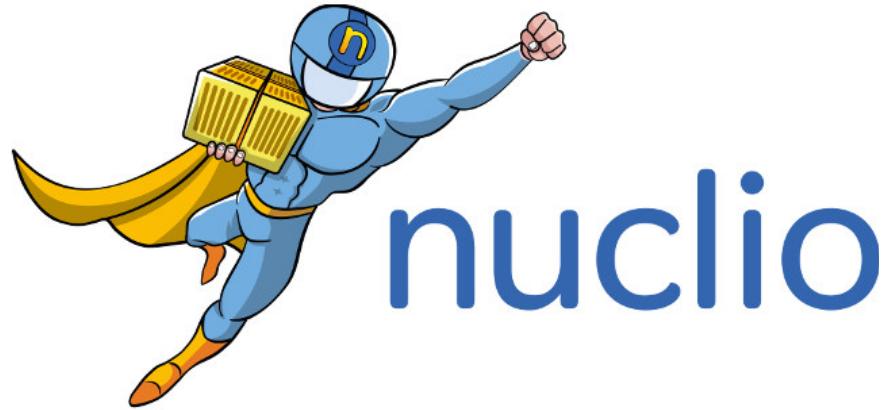
Kubeless





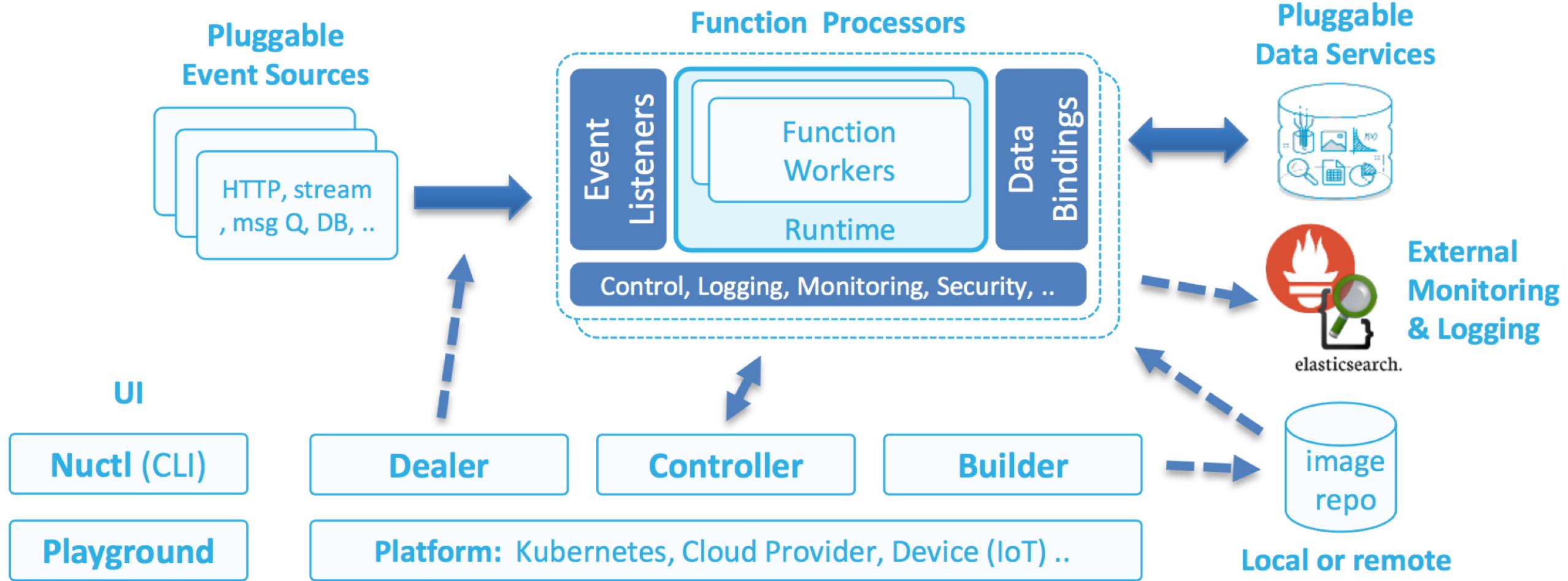
nuclio

What is Nuclio



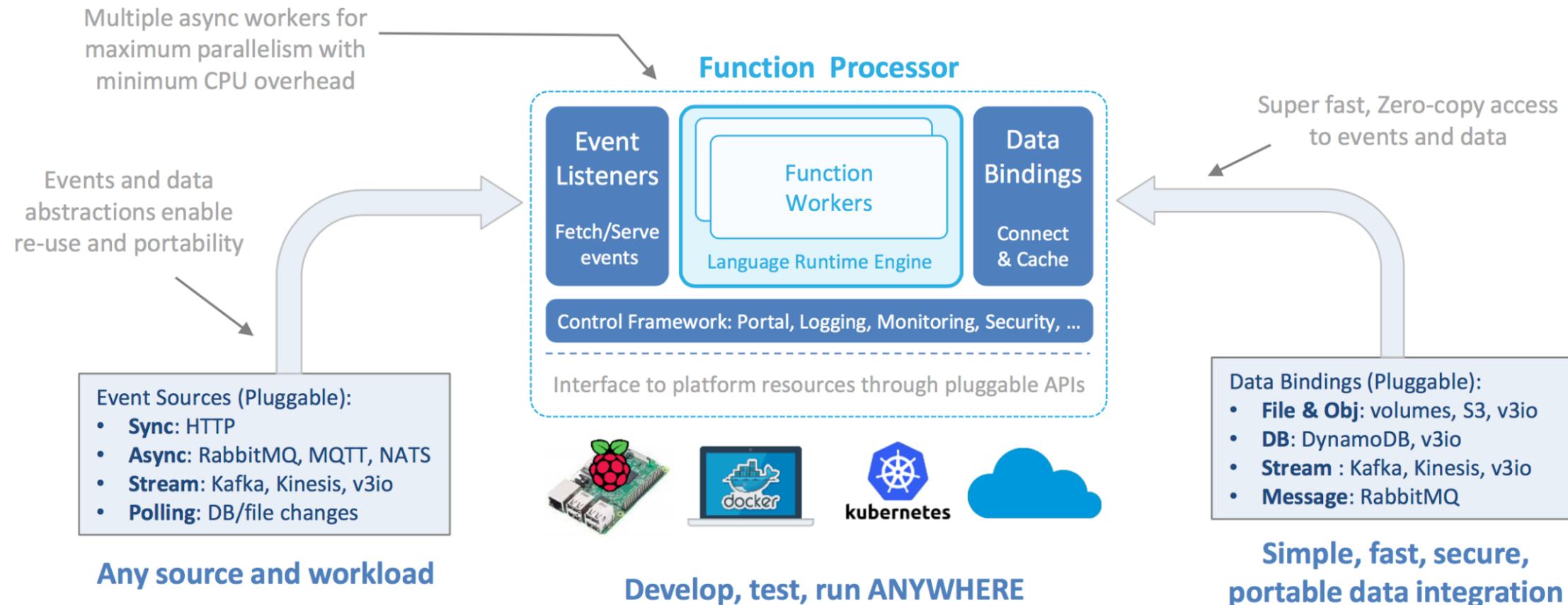
- Serverless for Real-Time Events and Data Processing
- Comprehensive, Open, Portable and Super Fast
“Serverless” by Iguazio
- Packaging
 - Used as a standalone binary (for example, for IoT devices),
 - Docker container
- Runs on
 - Raspberry Pi
 - Docker
 - Kubernetes
 - GCE

Architecture



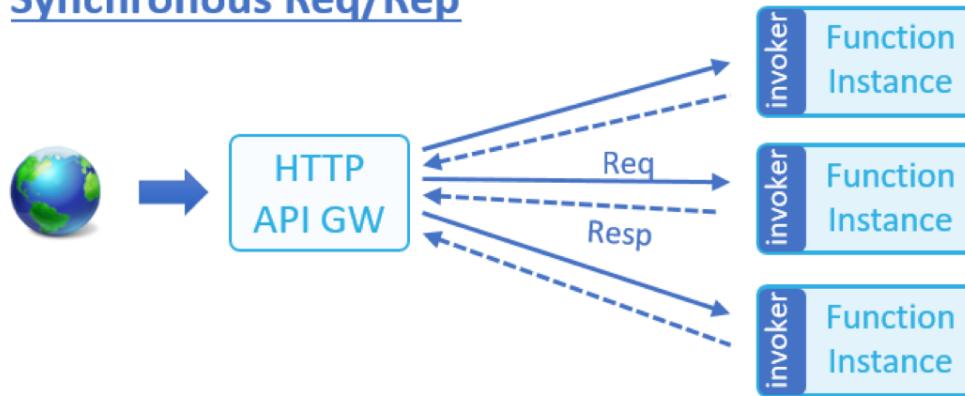
Function Processor

400K events/sec per process (100x faster than leading implementations)

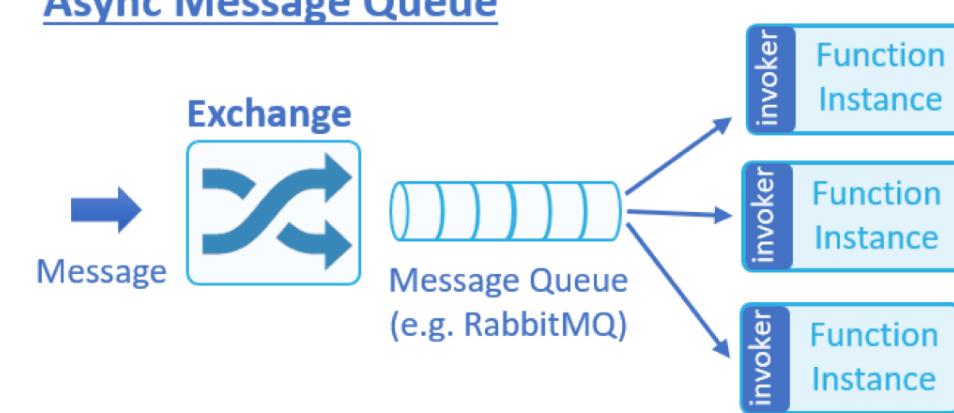


Event Sources

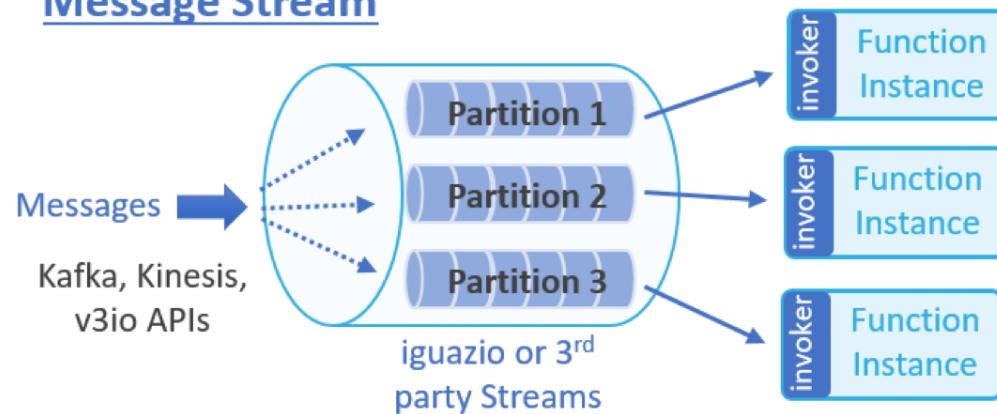
Synchronous Req/Rep



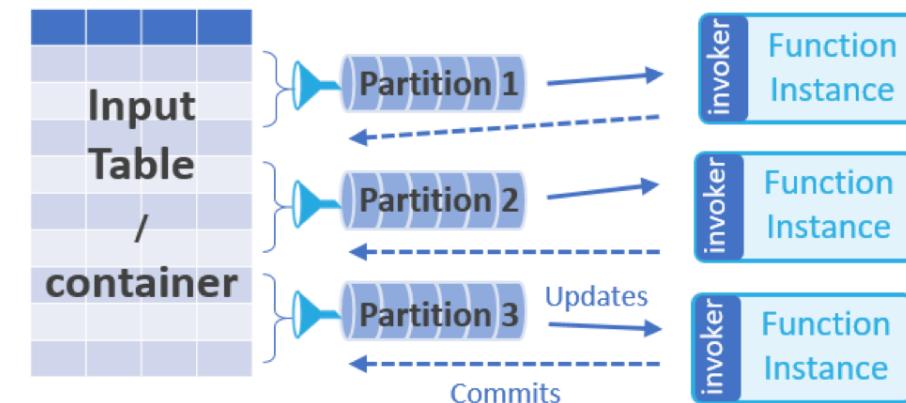
Async Message Queue



Message Stream



Record Polling/Stream





nuclio

Lets do it. . .



**Thank you.
Questions?**

yes, we're hiring

solarwinds 

The SolarWinds logo, featuring the company name in a bold, sans-serif font with a stylized orange flame or wind icon to the right.

The SolarWinds and SolarWinds Cloud trademarks are the exclusive property of SolarWinds Worldwide, LLC and its affiliates and may be registered or pending registration with the U.S. Patent and Trademark Office and in other countries. All other SolarWinds and SolarWinds Cloud trademarks, service marks, and logos may be common law marks or are registered or pending registration. All other trademarks mentioned herein are used for identification purposes only and are trademarks (and may be registered trademarks) of their respective companies.