



Serverless? Not so FaaS!

Matt Stine (@mstine)

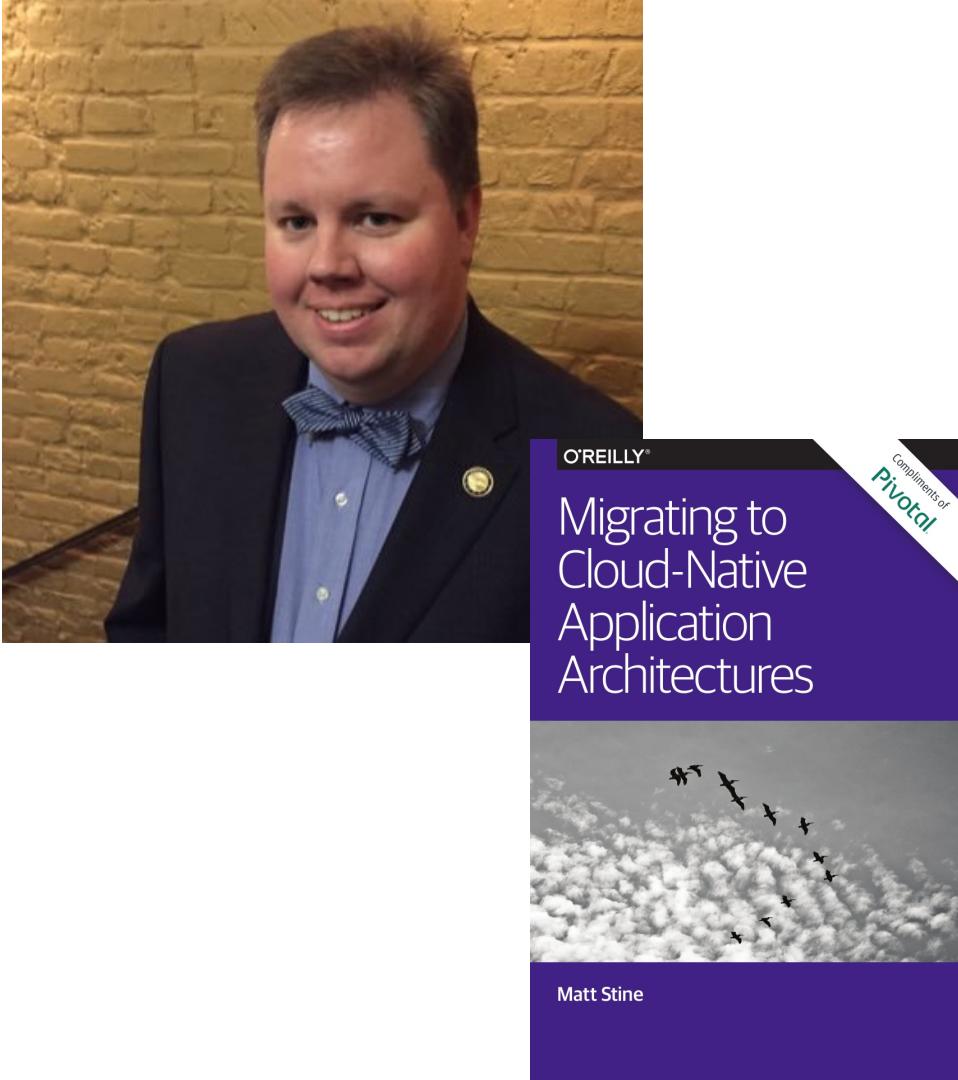
Global Field CTO / Chief Architect

mstine@pivotal.io

<http://mattstine.com>

Who is this guy?

- 18 year Enterprise IT veteran developer / architect
- 6 years focused on Cloud Native Platforms and Architecture
- Frequent conference circuit speaker
- Host of Software Architecture Radio
<http://softwarearchitecturerad.io>
- I wrote a little cloud book...

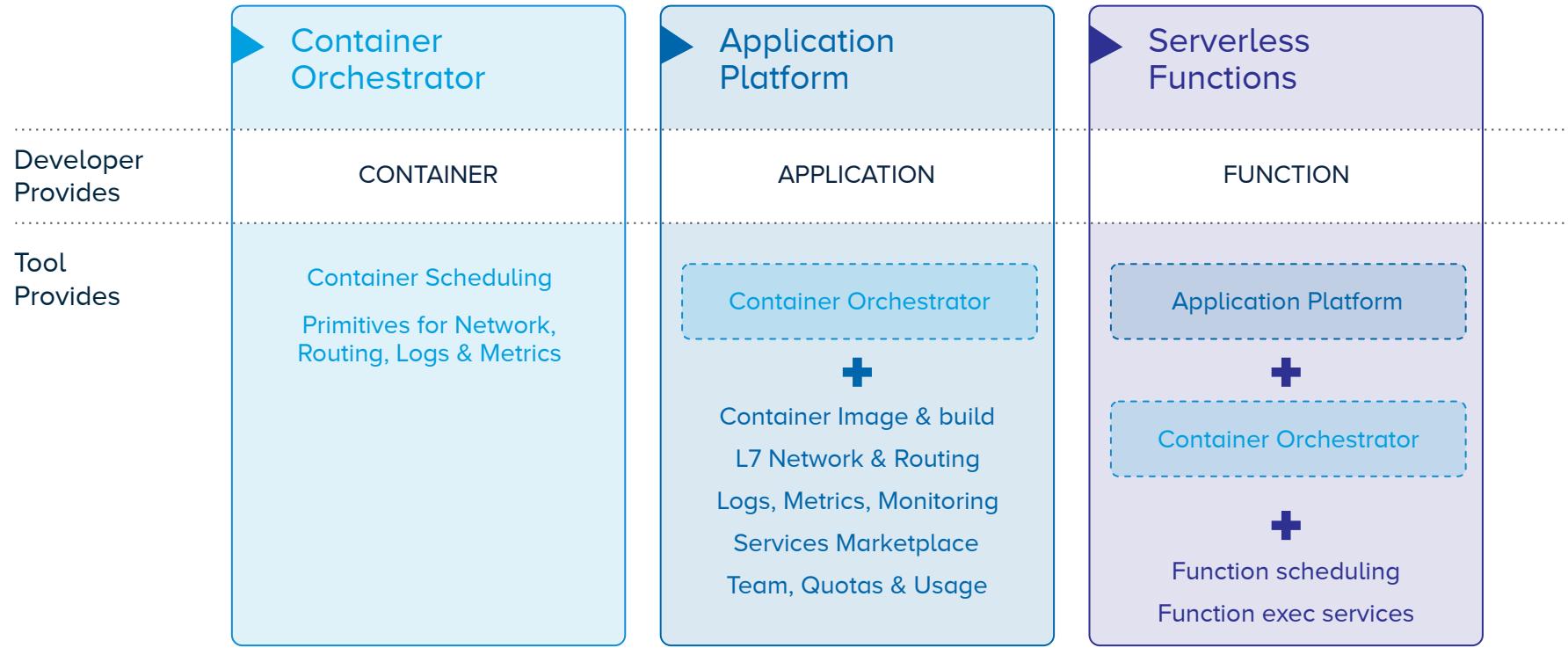


Agenda

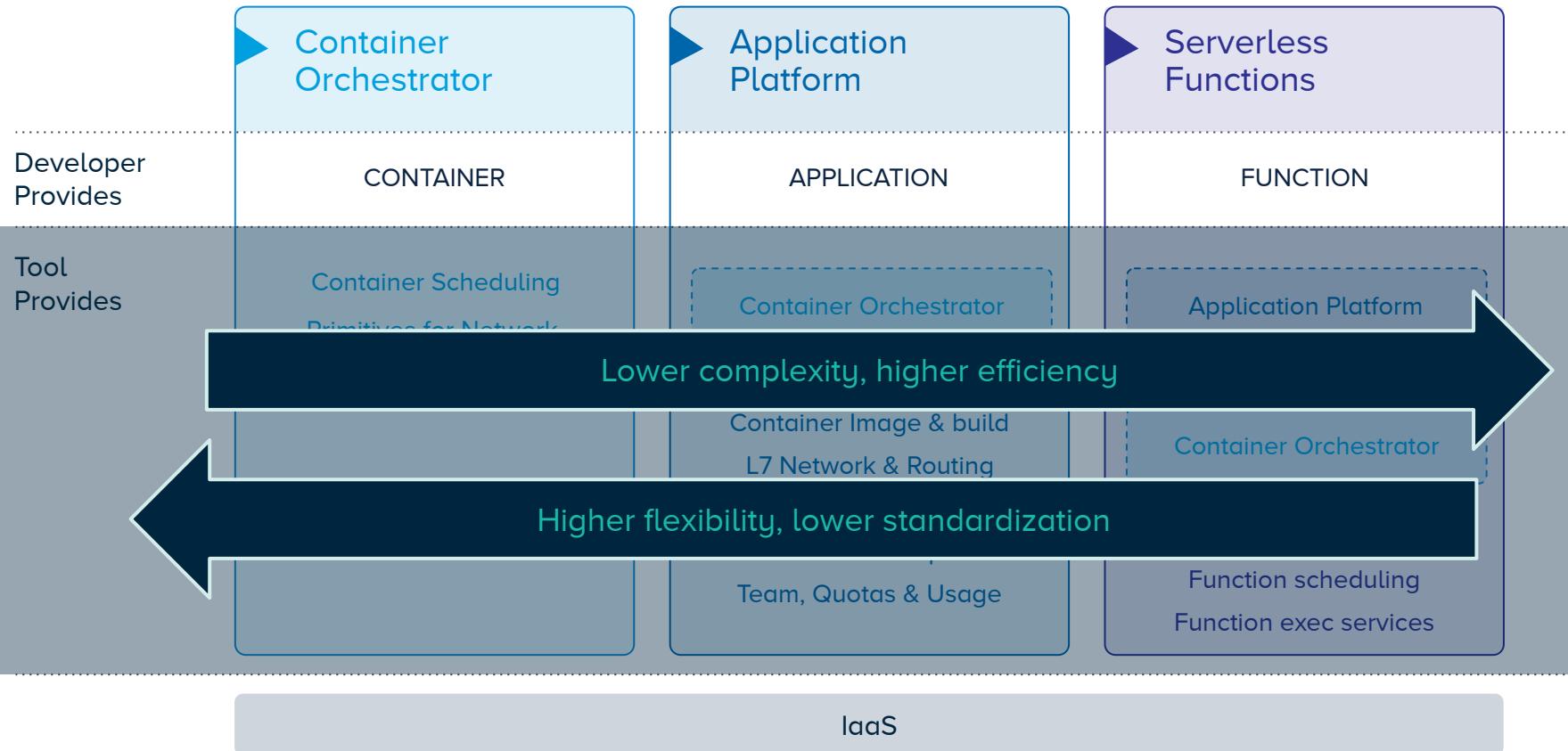
- The Four Abstractions
- FaaS vs. PaaS
- Use Cases
- Tradeoffs
- Provider Updates
- Rise of Kubernetes
- Java / Spring
- Smattering of Demos

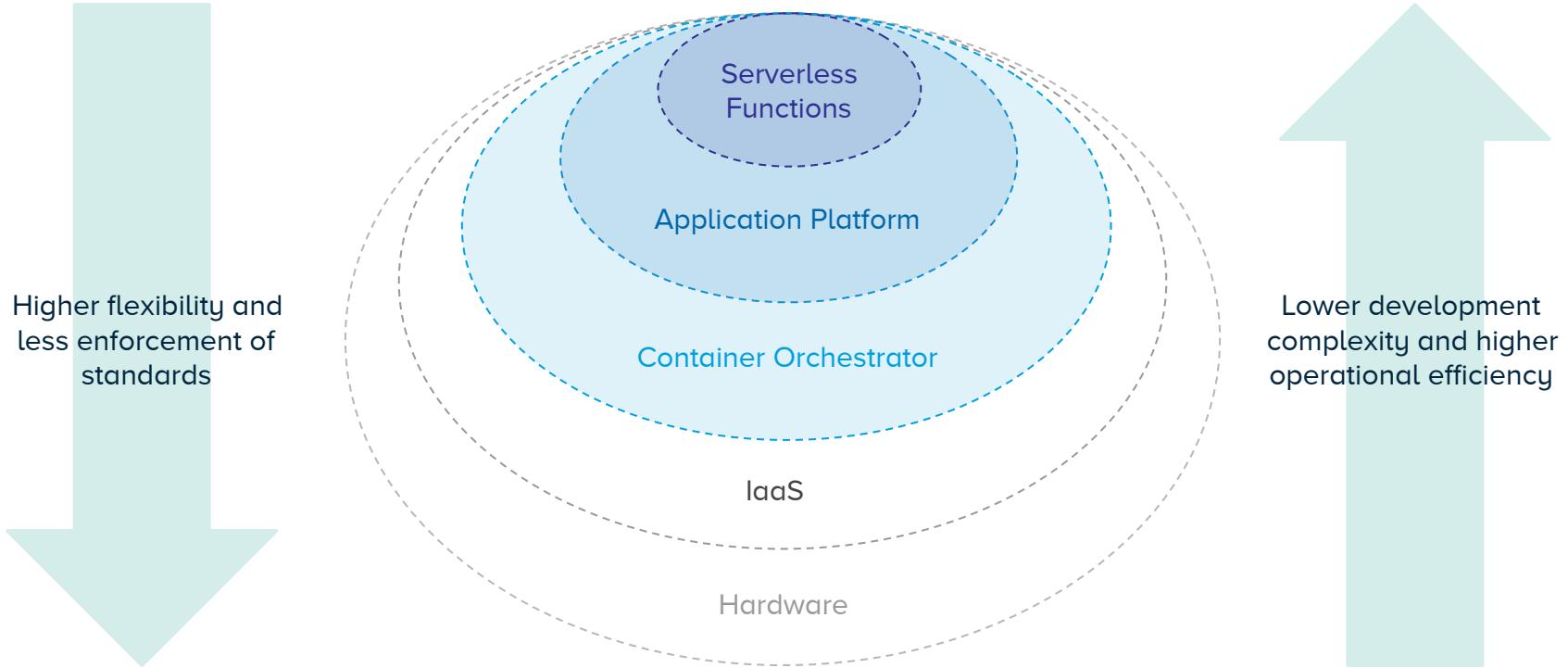


Choose the right tool for the job



Choose the right tool for the job





Strategic goal: Push as many workloads as technically feasible to the top of the platform hierarchy



Kelsey Hightower

@kelseyhightower

Following



I'm starting prototypes using a pure serverless approach then migrating "functions" to containers when I need lower latency or constant throughput for long periods of time.

8:59 AM - 28 Feb 2018

25 Retweets 112 Likes



What is FaaS?

Developer / Operator experience

PaaS	FaaS
runs apps	runs functions
push app to deploy server	register function and bind it to a trigger
server runs and waits for requests	function doesn't run until triggered
server listens to network	platform deploys and invokes functions
server handles lots of requests	functions handle events and then go away
scale out manually or by policy	auto-scale based on concurrent event load
pay per instance	pay per use – time & memory



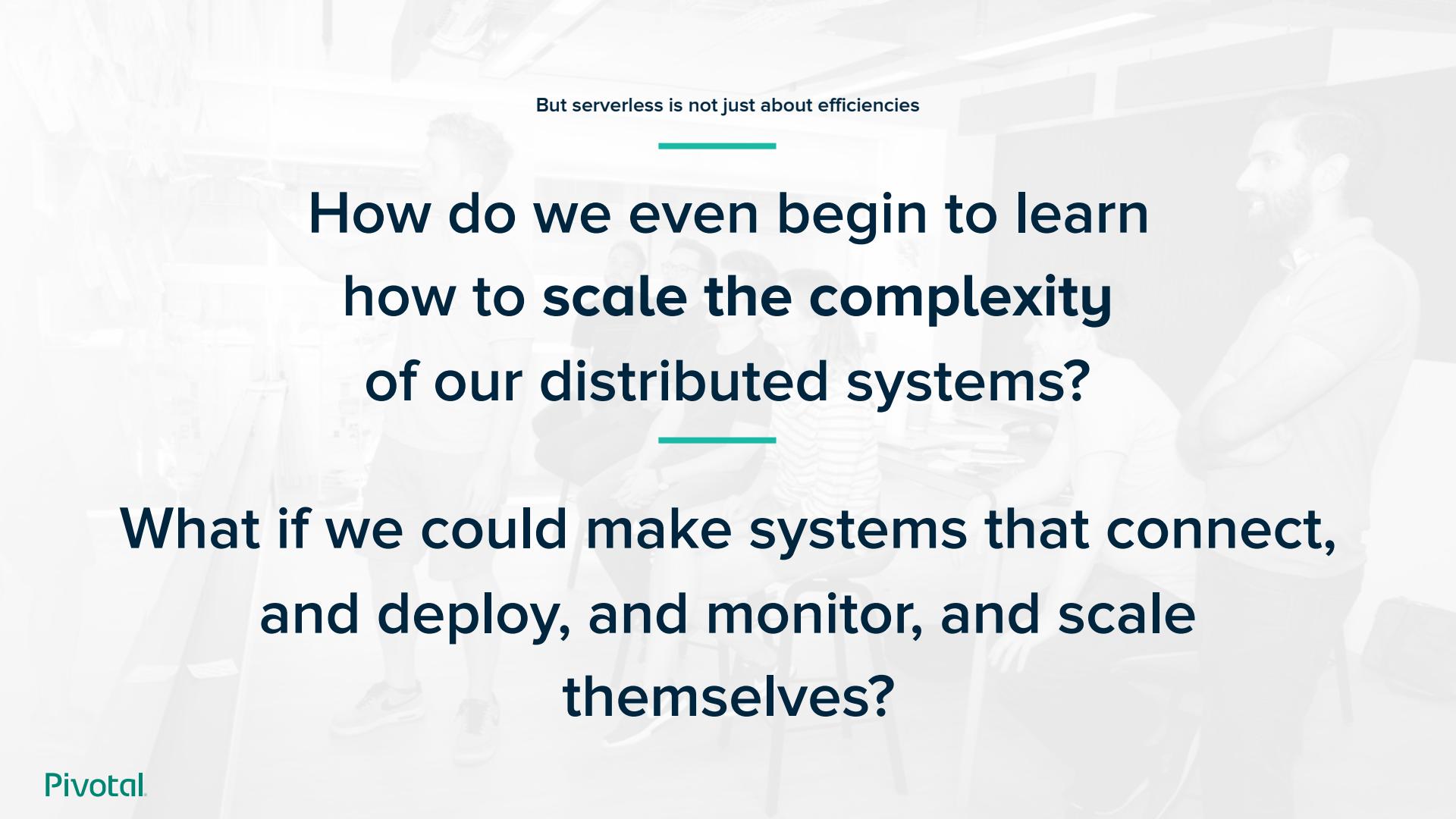
Serverless is telling us
“there’s even more you
don’t care about”.

Dave Syer
Serverless Spring
S1P 2017

Three types of efficiencies enabled by Serverless

- Narrowly-scoped units of code, and built-in event integration, contribute to **software development efficiencies**.
- Functions which don't consume resources when idle can provide significant **resource efficiencies**.
- Applying serverless to distributed computing brings **operational efficiencies** based on automated event-based scheduling and self-scaling.





But serverless is not just about efficiencies

**How do we even begin to learn
how to scale the complexity
of our distributed systems?**

**What if we could make systems that connect,
and deploy, and monitor, and scale
themselves?**

Three Categories of Use Cases

Web Events

- Website back-end services like form post handlers
- Back-end data services for mobile and web apps e.g GraphQL
- Webhook handlers
- CI/CD automation
- Chat integrations
- Digital assistant services e.g. Alexa skills

Event-based Integration

- Scheduled tasks, ETL
- File processing e.g. images and videos
- Complex Event Processing and Change Data Capture
- Monitoring, notifications and alerting
- Custom auth e.g. via API Gateway

Large Scale Data Processing

- e.g. pyWren map/reduce
- IoT streams
- Log ingestion
- Machine Learning
- Stateful Stream Processing

Tradeoffs

2017

- Vendor Control
- Vendor Lock-In
- Stateless!
- Maturity
- Developer UX

2018

- Rise of Open Source
- Rise of Abstraction Layers
- Rumors of Statefulness
- Starting to Grow Up
- Developer UX Improving!

AWS Lambda

<https://aws.amazon.com/lambda/>

- First FaaS, introduced in Nov 2014.
- Entry tier pricing attractive for low-volume Web handlers eg bots.
- Built-in triggers facilitate integration with many other AWS services.
- Auto-scales with concurrent events.
- Runs code written for Node.js, Python, Java 8, .NET Core, (and recently) Go.
- SAM (for declarative deployments) and SAM Local (for local simulation).

AWS Lambda

Run code without thinking about servers. Pay only for the compute time you consume.

Get started with AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running.

With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.



What is AWS Lambda?

Benefits

NO SERVERS TO MANAGE

AWS Lambda automatically runs your code without requiring you to provision or manage servers. Just write the code and upload it to Lambda.

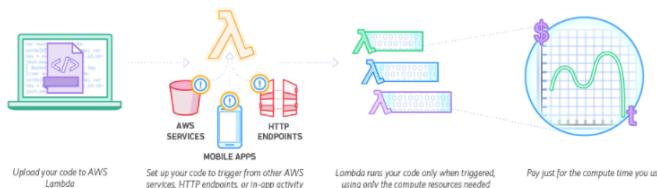
CONTINUOUS SCALING

AWS Lambda automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.

SUBSECOND METERING

With AWS Lambda, you are charged for every 100ms your code executes and the number of times your code is triggered. You don't pay anything when your code isn't running.

How it works



A photograph of a group of people in an office or workshop environment. On the left, a man in a grey t-shirt and shorts stands writing on a whiteboard with a marker. In the center, three people are seated on a black metal stool; two are looking towards the right, and one is looking towards the left. On the right, a man in a light blue polo shirt and jeans stands with his arms crossed, looking towards the left. The background shows office equipment and a window.

DEMO

```
# mstine at eolian.local in ~/work [22:35:27]
→ mvn archetype:generate -DarchetypeGroupId=com.mattstine.aws -DarchetypeArtifactId=aws-lambda-archetype -DarchetypeVersion=1.0-SNAPSHOT
```

Azure Functions

<https://azure.microsoft.com/services/functions/>

- Available since Nov 2016 with support for .NET and JavaScript based on WebJobs C# OSS runtime.
- New v2 (.NET core) runtime in preview also supports Java and Linux or Windows containers.
- Developers can use Core-Tools to build and run functions locally.
- Included with Azure Stack.

Microsoft Azure

SALES 1-800-867-1389 | MY ACCOUNT | PORTAL | Search | FREE ACCOUNT >

Functions

Build apps faster with a serverless architecture

Accelerate your development with an event-driven, serverless compute experience. Scale on demand and pay only for the resources you consume.

[Start free >](#)

[Sign in to your account >](#)

Explore Azure Functions: [Pricing details](#) [Documentation](#) [Roadmap](#) [Community](#) [Serverless computing](#)



Take advantage of serverless compute with Functions

Easily build the apps you need using simple, serverless functions that scale to meet demand. Use the programming language of your choice, and don't worry about servers or infrastructure.



Manage your apps instead of infrastructure

Focus on building great apps. Don't worry about provisioning and maintaining servers, especially when your workload grows. Azure Functions provides a fully managed compute platform with high reliability and security. With scale on demand, you get the resources you need—when you need them.



 Develop your way

Create Functions in the language you choose, such as JavaScript, C#, and Java, and scripting options such as Python, PHP, Bash, Batch, and PowerShell. Write code in an easy-to-use web-based interface, or upload precompiled code built with your favorite development tool. Take advantage of continuous deployment to integrate with your DevOps pipeline and use monitoring tools to troubleshoot.

Bind into services

Build rich, [serverless](#) scenarios by capitalizing on a range of Azure and external services. Easily interact with Azure Cosmos DB, Storage, and more, and external services including Twilio, SendGrid, and others—by adding input/output bindings to Functions. Save costs with less code to maintain.



A photograph of a group of people in an office or workshop environment. On the left, a man in a grey t-shirt and shorts stands writing on a whiteboard with a marker. In the center, three people are seated on a black metal stool; two are looking towards the right, and one is looking towards the left. On the right, a man in a light blue polo shirt and jeans stands with his arms crossed, looking towards the left. The background shows office equipment and a window.

DEMO

```
# mstine at eolian.local in ~/work [23:56:37]
→ mvn archetype:generate \
  -DarchetypeGroupId=com.microsoft.azure \
  -DarchetypeArtifactId=azure-functions-archetype
```

Google Cloud Functions

<https://cloud.google.com/functions/>

- Still in Beta
- Supports node.js
runs in a docker container
performs npm install build step
can deploy from git repo
- Blocks per request and scales using concurrency (like Lambda)
- Integrates with events from HTTP, Pub/Sub, Storage, and Firebase.

Google Cloud Platform

Why Google Products Solutions Launcher Pricing Customers Documentation Support Partners TRY IT FREE CONTACT SALES

CLOUD FUNCTIONS BETA

A serverless environment to build and connect cloud services

(...) TRY IT FREE VIEW DOCUMENTATION

Serverless Applications on Google's Infrastructure

Cloud computing has made possible fully serverless models of computing where logic can be spun up on-demand in response to events originating from anywhere. Construct applications from bite-sized business logic billed to the nearest 100 milliseconds, only while your code is running. Serve users from zero to planet-scale, all without managing any infrastructure.



Microservices Over Monoliths

Developer agility comes from building systems composed of small, independent units of functionality focused on doing one thing well. Cloud Functions lets you build and deploy services at the level of a single function, not at the level of entire applications, containers, or VMs.



Connect & Extend Cloud Services

Cloud Functions provides a connective layer of logic that lets you write code to connect and extend cloud services. Listen and respond to events such as a file upload to Cloud Storage, an incoming message on a Cloud Pub/Sub topic, a log change in Stackdriver Logging, or a mobile-related event from Firebase. Cloud Functions augments existing cloud services and allows you to address an increasing number of use cases with event-driven code.



Serverless Economics

Cloud Functions are ephemeral, spinning up on-demand and back down in response to events in the environment. Pay only while your function is executing, metered to the nearest 100 milliseconds, and pay nothing after your function finishes.



The Rise of Kubernetes

Kubernetes is an open-source platform designed to automate deploying, scaling, and operating **application containers**.



With Kubernetes, you are able to quickly and efficiently respond to customer demand:

- Deploy your applications quickly and predictably.
- Scale your applications on the fly.
- Roll out new features seamlessly.
- Limit hardware usage to required resources only.

A large cargo ship, likely an container ship, is shown from a low angle, filling most of the frame. The ship's hull is dark, and it is covered in numerous shipping containers stacked high. The sky above is overcast and grey.

Kubernetes has become the de facto
standard for container orchestration.

Kubernetes FTW

- Amazon Elastic Container Service for Kubernetes (EKS)
- Azure Container Service (AKS)
- Google Kubernetes Engine (GKE)



FaaS Kubernetes Landscape

Public Cloud

EKS + FaaS

AWS Lambda

Azure Functions

Google Cloud Functions

AKS + FaaS

GKS + FaaS

Pivotal

Kubernetes Native FaaS



Fission

Kubeless

Other OSS FaaS (Will Run on K8S)

Apache OpenWhisk

OpenFaaS

fn Project

nuclio

Other OSS FaaS / Will Run on K8S

- **Apache OpenWhisk** (<https://openwhisk.apache.org>)

(packs the container for you, but not K8S native)

- **OpenFaaS** (<https://www.openfaas.com>)

(uses Docker Swarm or K8S as backend runtime - you pack the container)

- **fn Project** (<https://fnproject.io>)

(orchestrator agnostic - you pack the container)

- **nuclio** (<https://nuclio.io>)

(runs on k8s, packs container for you, manually configured ingress)

Kubernetes Native FaaS

- **Fission** (<https://openwhisk.apache.org>)

(Abstracts Docker/K8S, provides runtimes and triggers)

- **Kubeless** (<http://kubeless.io>)

(K8S CRDs, Deployments/Pods, Config Map, Services)

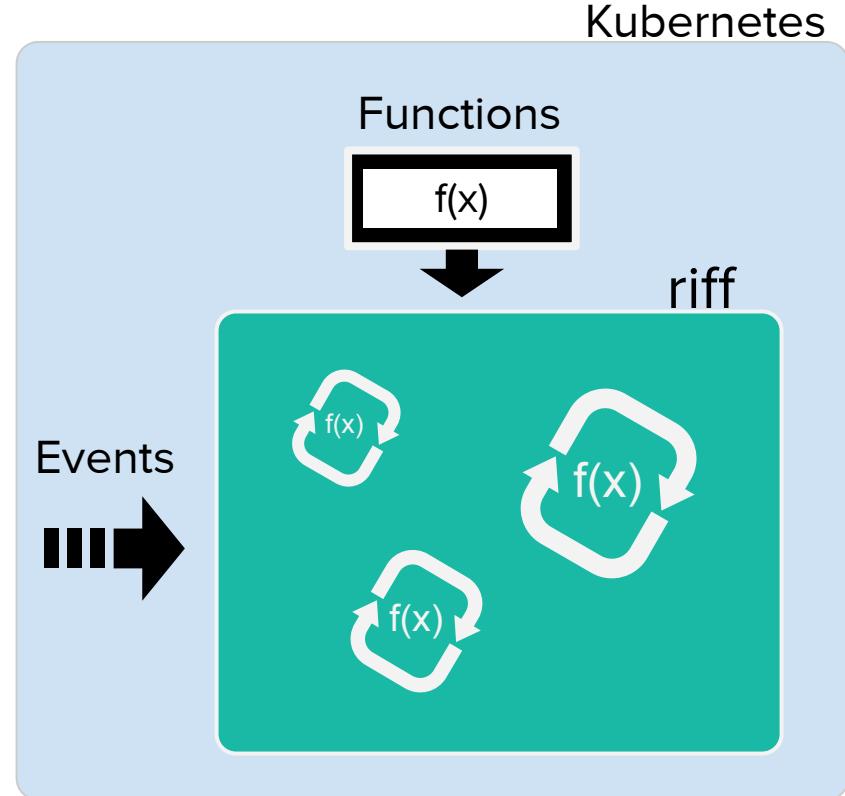
- **Riff** (<https://projectriff.io>)

(Let's go deeper...)

What is riff ?

- **riff is for functions!**
- riff provides developers with a FaaS for executing Functions triggered by Events.
- riff is a new open source project
- sponsored by Pivotal
- built by a team of engineers from Spring and PCF.

<https://github.com/projectriff>



A photograph of a group of people in an office or workshop environment. On the left, a man in a grey t-shirt and shorts stands writing on a whiteboard with a marker. In the center, three people are seated on a black metal stool; two are looking towards the right, and one is looking towards the left. On the right, a man in a light-colored polo shirt and jeans stands with his arms crossed, looking towards the left. The background shows shelves with papers and a window.

DEMO

```
# mstine at eolian.local in ~/work/riff [0:43:05]
```

```
→
```

```
Every 1.0s: kubectl get po,deploy --namespace riff-system
```

```
eolian.local: Fri Apr 27 00:44:07 2018
```

NAME	READY	STATUS	RESTARTS	AGE
po/projectriff-kafka-855897b9b6-mgld6	1/1	Running	4	2d
po/projectriff-riff-function-controller-6cd454579b-lp5rp	1/1	Running	6	2d
po/projectriff-riff-http-gateway-bc46b8d8f-k5ssq	1/1	Running	6	2d
po/projectriff-riff-topic-controller-65f56c4ff-htlsg	1/1	Running	0	2d
po/projectriff-zookeeper-5f898c6869-j2ghg	1/1	Running	0	2d

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/projectriff-kafka	1	1	1	1	2d
deploy/projectriff-riff-function-controller	1	1	1	1	2d
deploy/projectriff-riff-http-gateway	1	1	1	1	2d
deploy/projectriff-riff-topic-controller	1	1	1	1	2d
deploy/projectriff-zookeeper	1	1	1	1	2d

Developers write functions

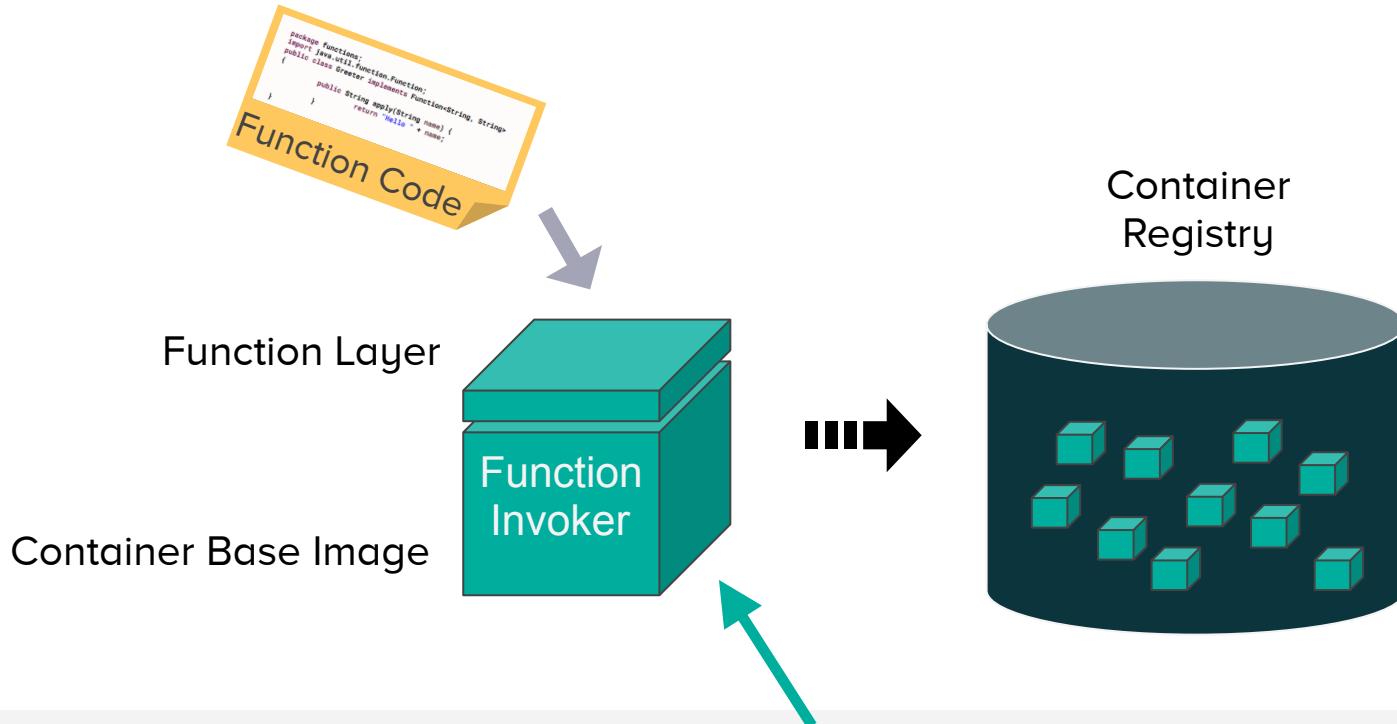
greeter.sh

```
echo "hello $1"
```

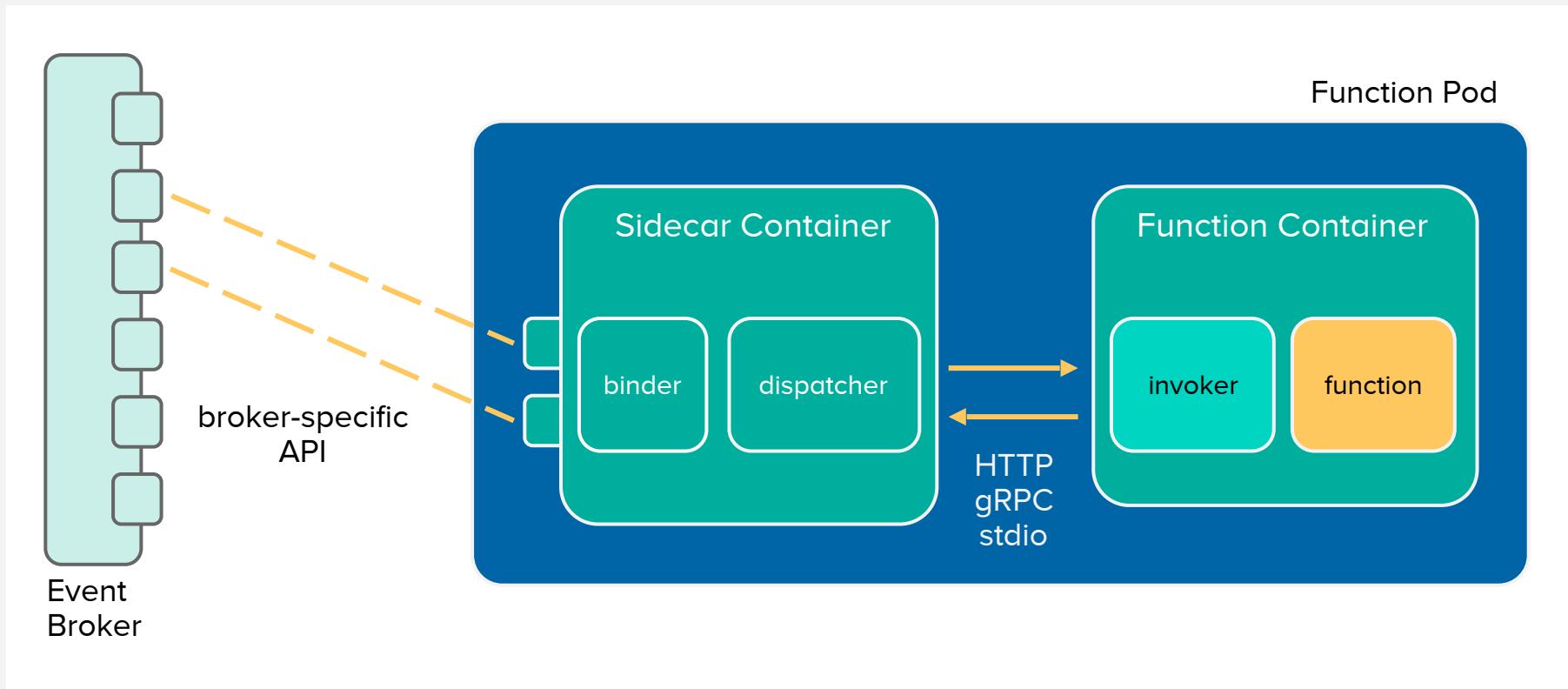
square.js

```
module.exports = (x) => x ** 2
```

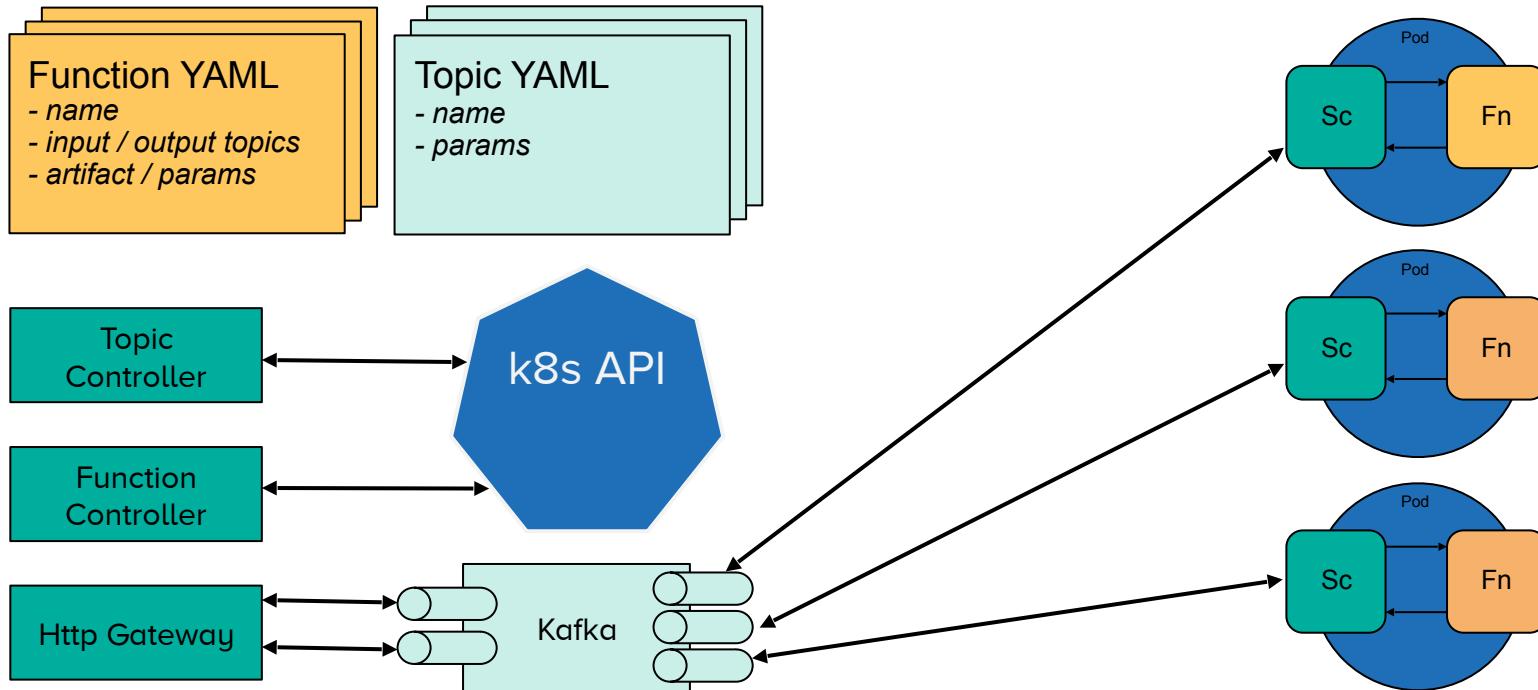
Functions are packaged as containers



Sidecars connect functions with event brokers



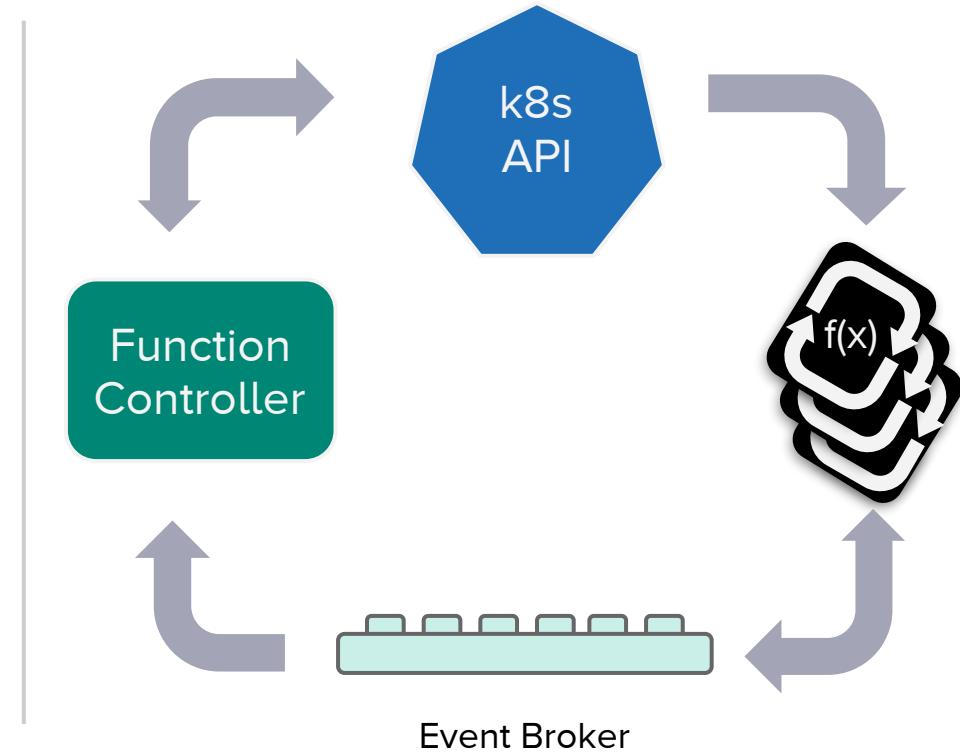
Functions and topics are Kubernetes resources



Functions scale with events

riff function controller

- ★ interacts with k8s API
- ★ monitors event-lag
- ★ scales functions 0-1 and 1-N



PROJECTS : SPRING CLOUD

Spring Cloud Function



Spring Cloud Function is a project with the following high-level goals:

- Promote the implementation of business logic via functions.
- Decouple the development lifecycle of business logic from any specific runtime target so that the same code can run as a web endpoint, a stream processor, or a task.
- Support a uniform programming model across serverless providers, as well as the ability to run standalone (locally or in a PaaS).
- Enable Spring Boot features (auto-configuration, dependency injection, metrics) on serverless providers.

It abstracts away all of the transport details and infrastructure, allowing the developer to keep all the familiar tools and processes, and focus firmly on business logic.

[QUICK START](#)

<https://cloud.spring.io/spring-cloud-function/>

Spring Cloud Functions:

- Plain old `java.util.function` Functions, Consumers, Suppliers
- Registered as Beans via `@FunctionScan`
- Can leverage Reactor's Flux (Reactive Streams Publisher)
- Adapters for Java FaaS providers (AWS, Azure, OpenWhisk)

Plain Old Java Functions

```
@SpringBootApplication
public class Application {

    @Bean
    public Function<Flux<String>, Flux<String>> uppercase() {
        return flux -> flux.map(value -> value.toUpperCase());
    }

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

A photograph of a group of people in an office or workshop environment. On the left, a man in a grey t-shirt and shorts stands writing on a whiteboard with a marker. In the center, three people are seated on a black metal stool; two are looking towards the right, and one is looking towards the left. On the right, a man in a light blue polo shirt and jeans stands with his arms crossed, looking towards the left. The background shows office equipment and a window.

DEMO

```
# mstine at eolian.local in ~/workspace/spring-cloud-function/spring-cloud-function-samples/function-sample-pojo on git  
:72efd58 * [1:55:43]  
→
```

{

```
# mstine at eolian.local in ~/workspace/spring-cloud-function/spring-cloud-function-samples/function-sample-pojo on git  
:72efd58 * [1:55:40]  
→
```



Transforming How The World Builds Software

Matt Stine (@mstine)

Global Field CTO / Chief Architect

mstine@pivotal.io

<http://mattstine.com>

GREAT INDIAN **DEVELOPER** SUMMIT



2019™

Conference : April 23-26, Bangalore



Register early and get the best discounts!



www.developersummit.com



@greatindiandev



bit.ly/gidslinkedin



facebook.com/gids19



bit.ly/saltmarchyoutube



flickr.com/photos/saltmarch/