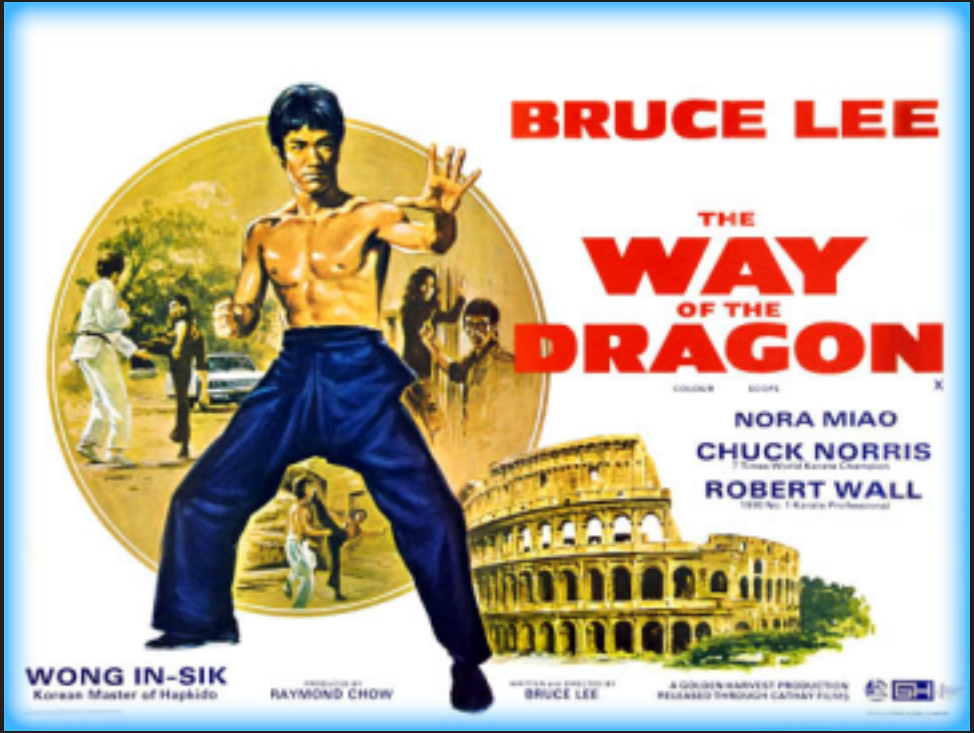
RED HAT DEVELOPER

# Kubernetes Serverless With Knative

Burr Sutter (burrsutter.com)

http://bit.ly/knative-tutorial



http://gonewiththetwins.com/new/way-dragon-return-dragon-1972/

http://www.fanpop.com/clubs/bruce-lee/images/27605335/title/way-dragon-photo

# Exercise Setup

https://redhat-developer-demos.github.io/knative-tutorial/knative-tutorial/v1.0.0/setup.html

Testing/Demo Scripts
https://github.com/burrsutter/scripts-knative
https://github.com/burrsutter/sidebyside

# Knative Tutorial Exercises (bit.ly/knative-tutorial)

- Setup
- Deploy Knative Service & Revisions
- Configurations & Routes
- Auto-scaling
- Build
- Eventing

# Agenda

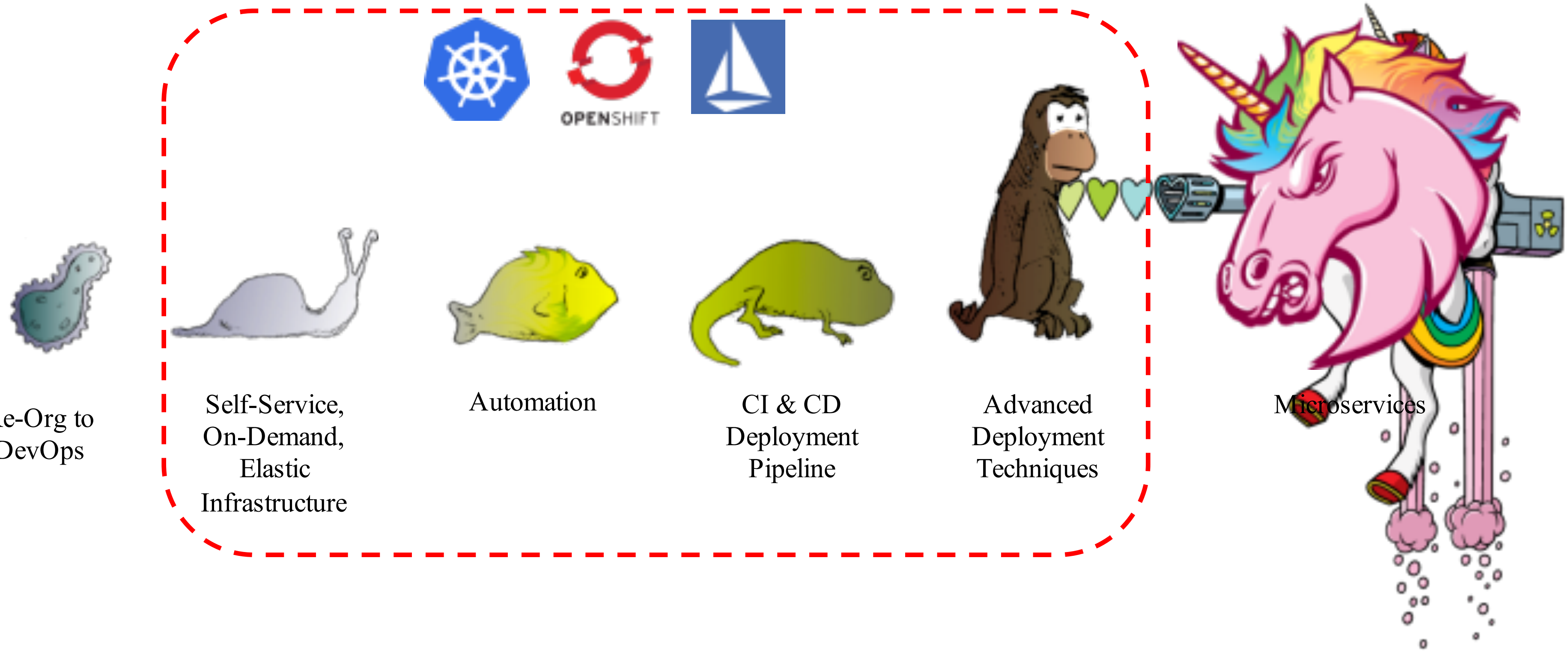
- What is Serverless
- Serverless vs FaaS
- Serverless via BaaS & SaaS
- FaaS Introduction
- Knative Serving
- Knative Build
- Knative Eventing

Are custom software Apps/APIs a key strategic advantage for your organization?

OR

Do you regard IT as a cost center that must be better streamlined?

# Your Journey to Awesomeness



Re-Org to DevOps

Self-Service, On-Demand, Elastic Infrastructure

Automation

CI & CD Deployment Pipeline

Advanced Deployment Techniques

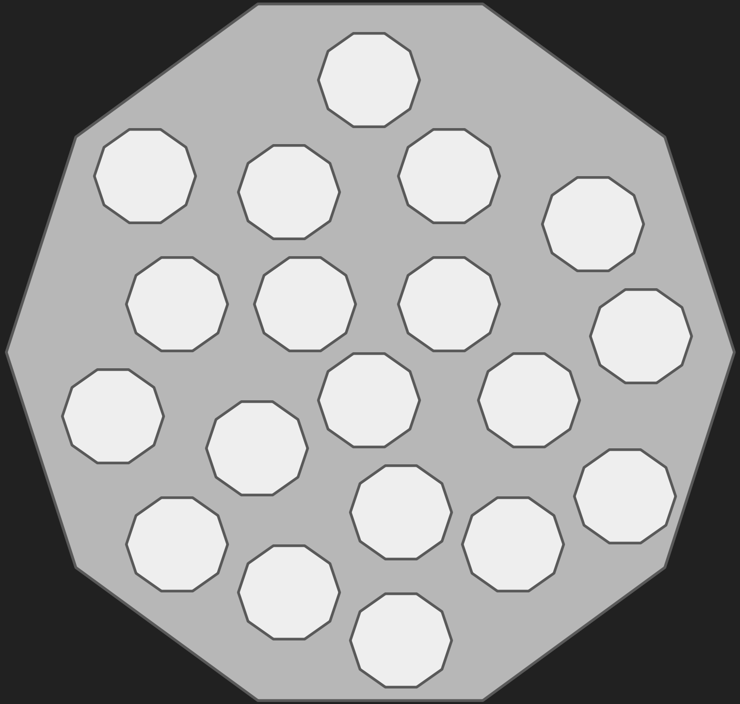
Microservices

We cannot solve our problems with the same thinking we used when we created them.

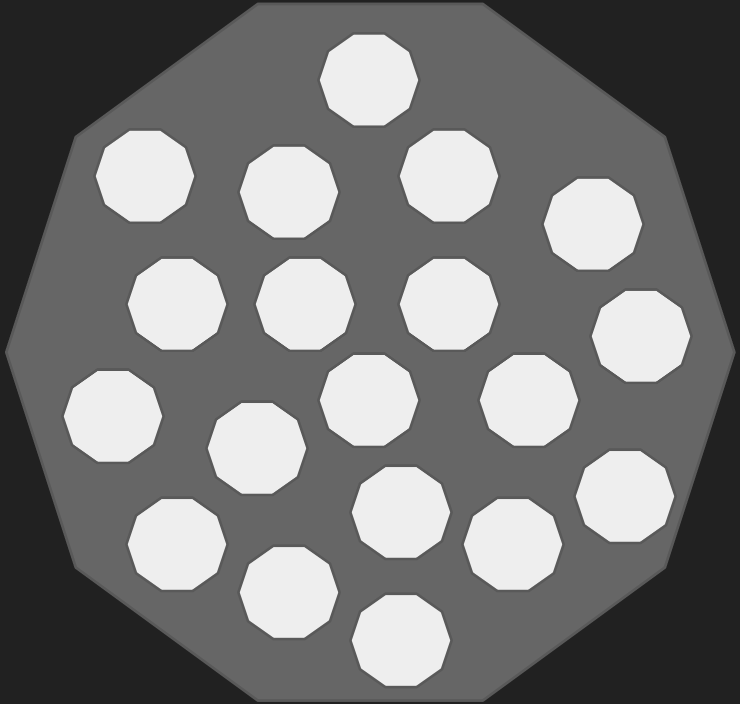
Albert Einstein
(Theoretical Physicist)

# Monolith

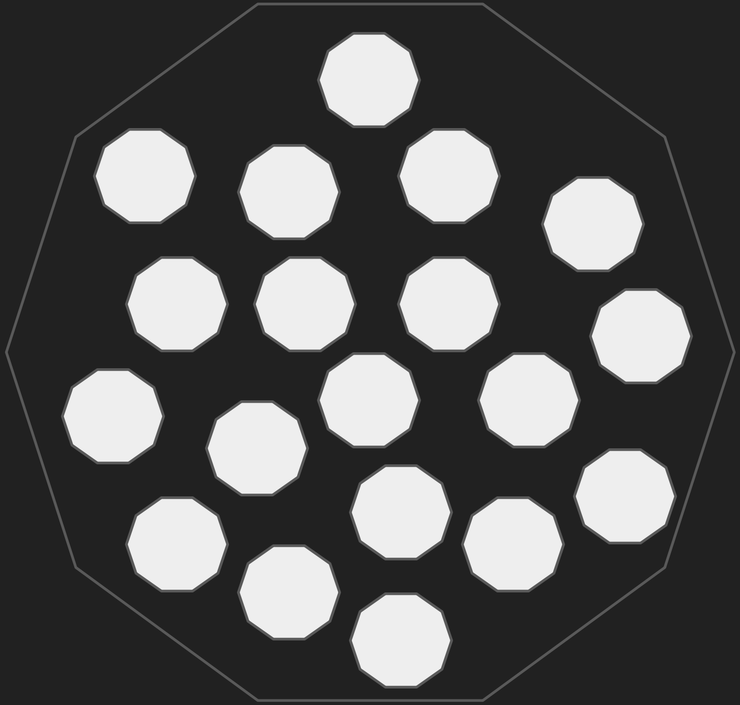
# The Application



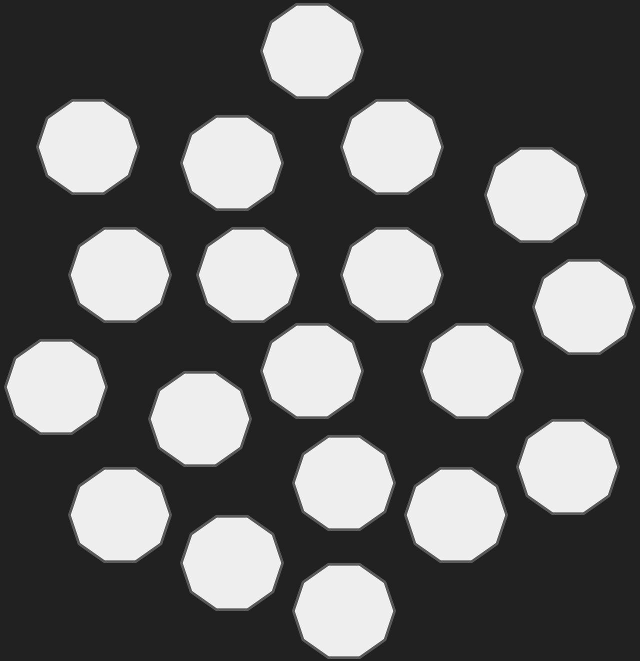
# Modules



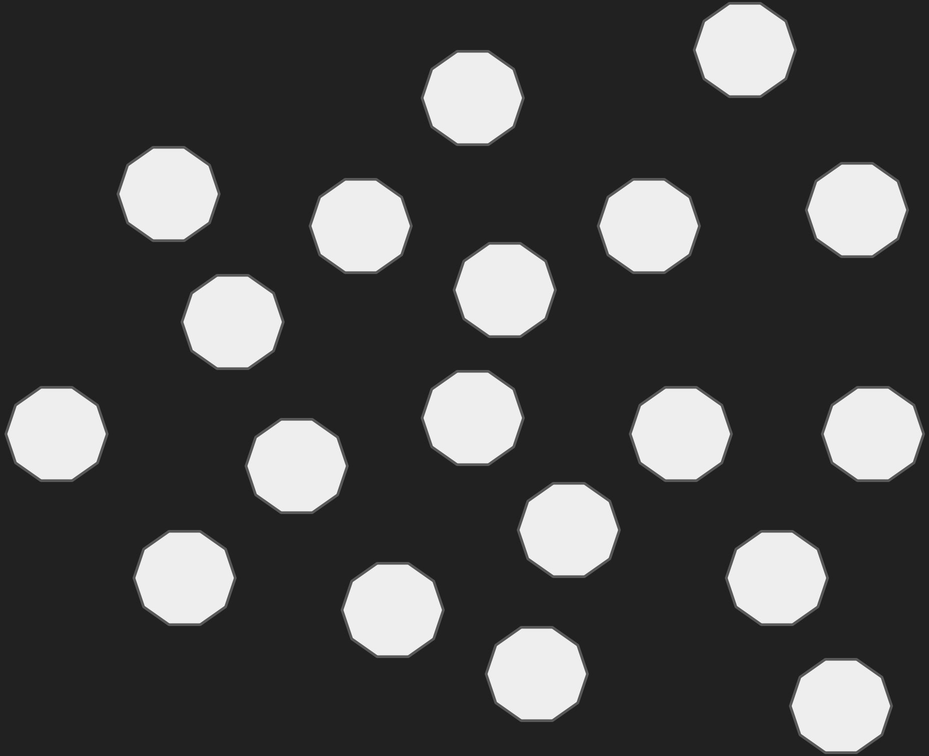
# Microservices



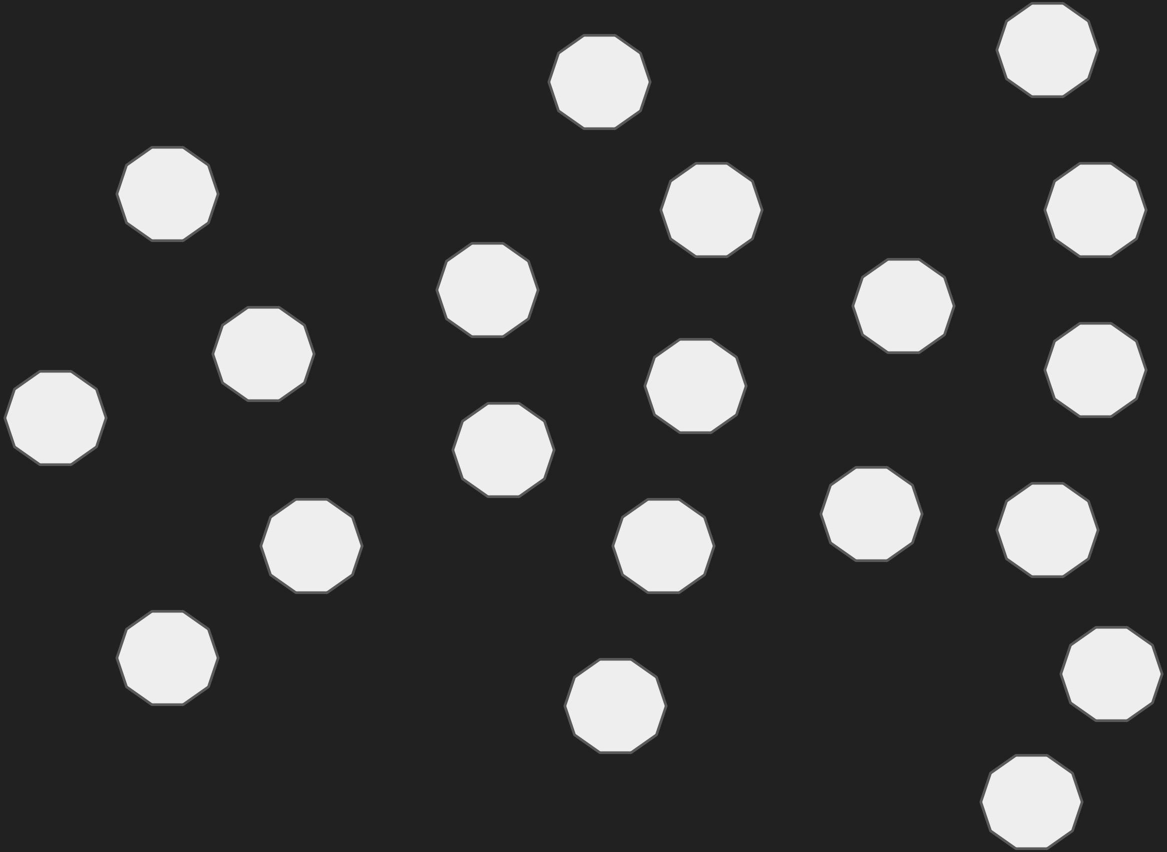
# Microservices



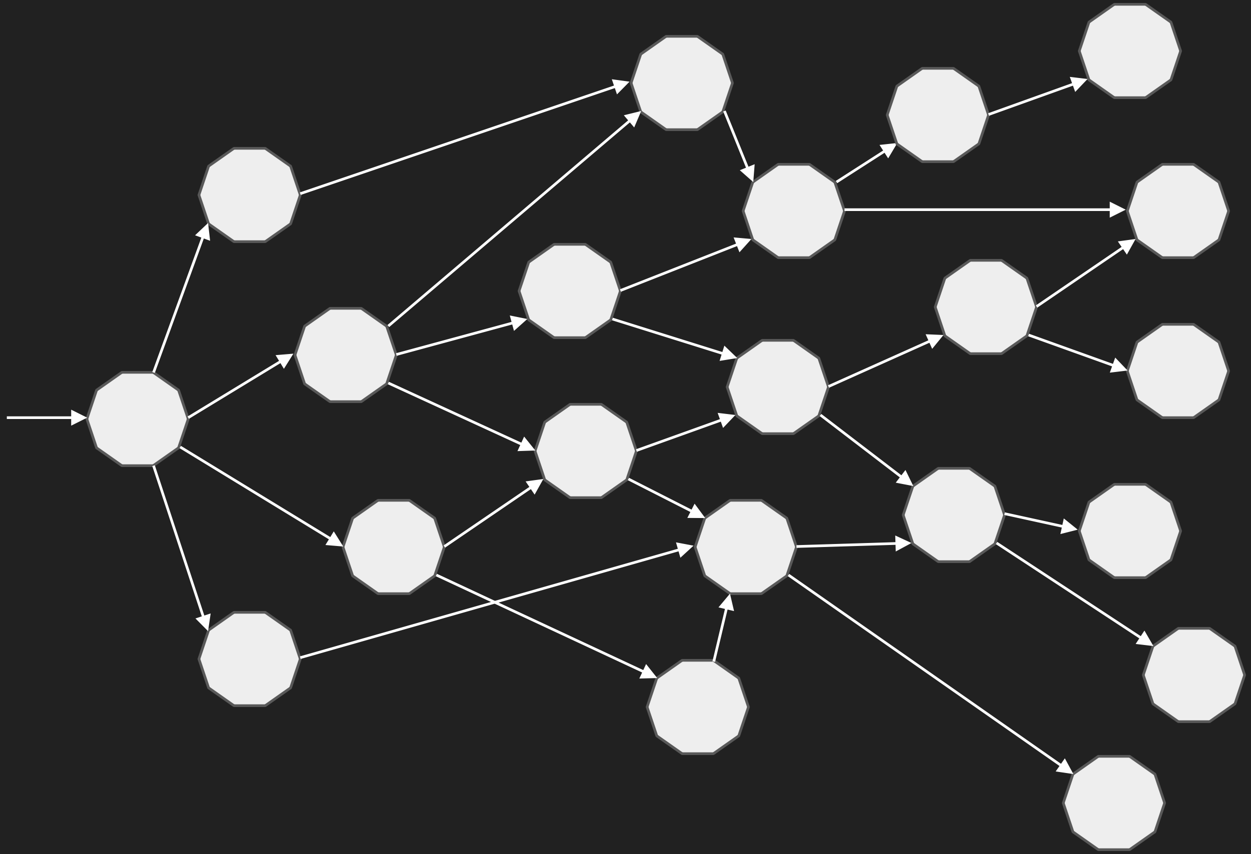
# Microservices



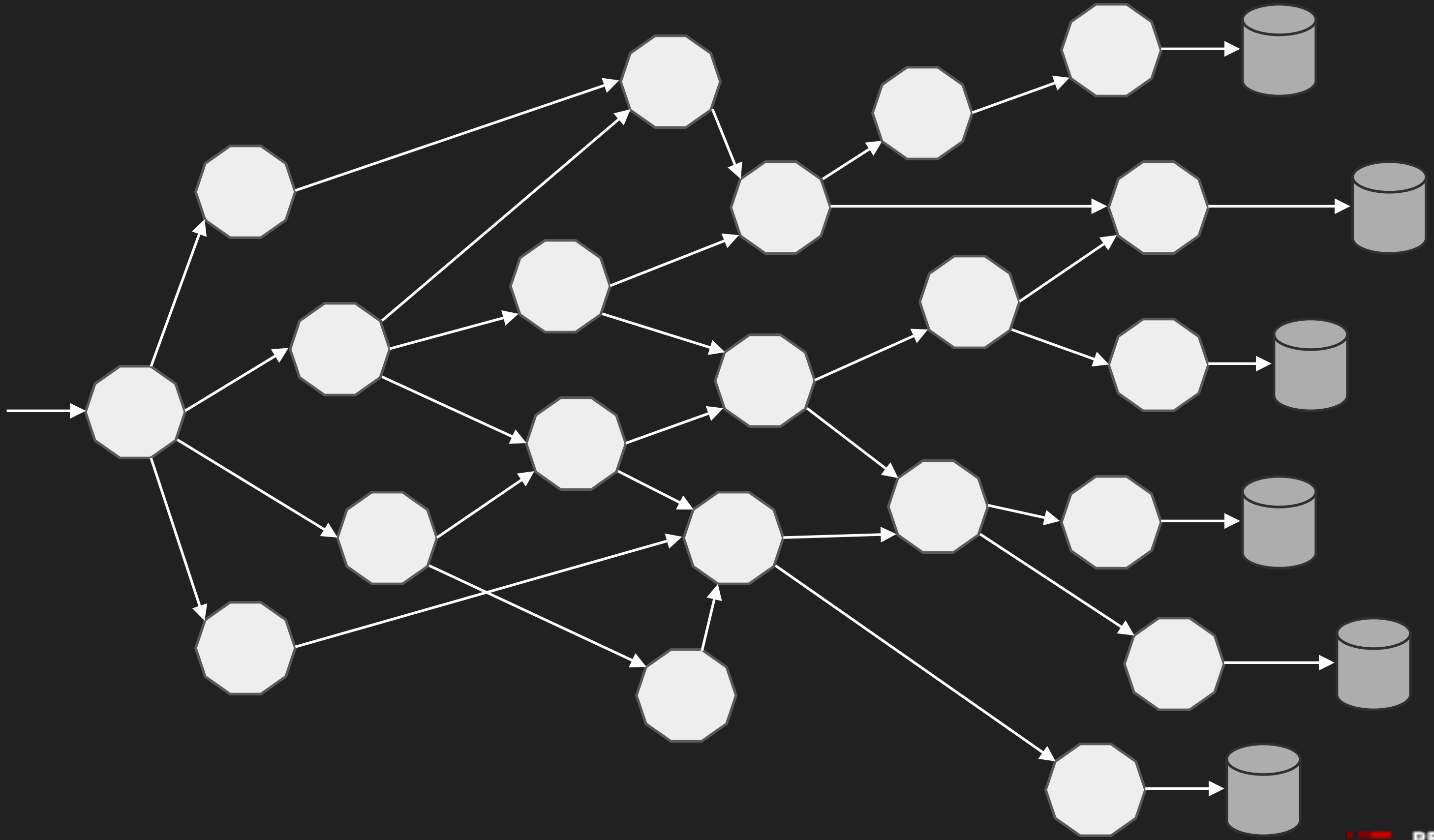
# Microservices



# Network of Services

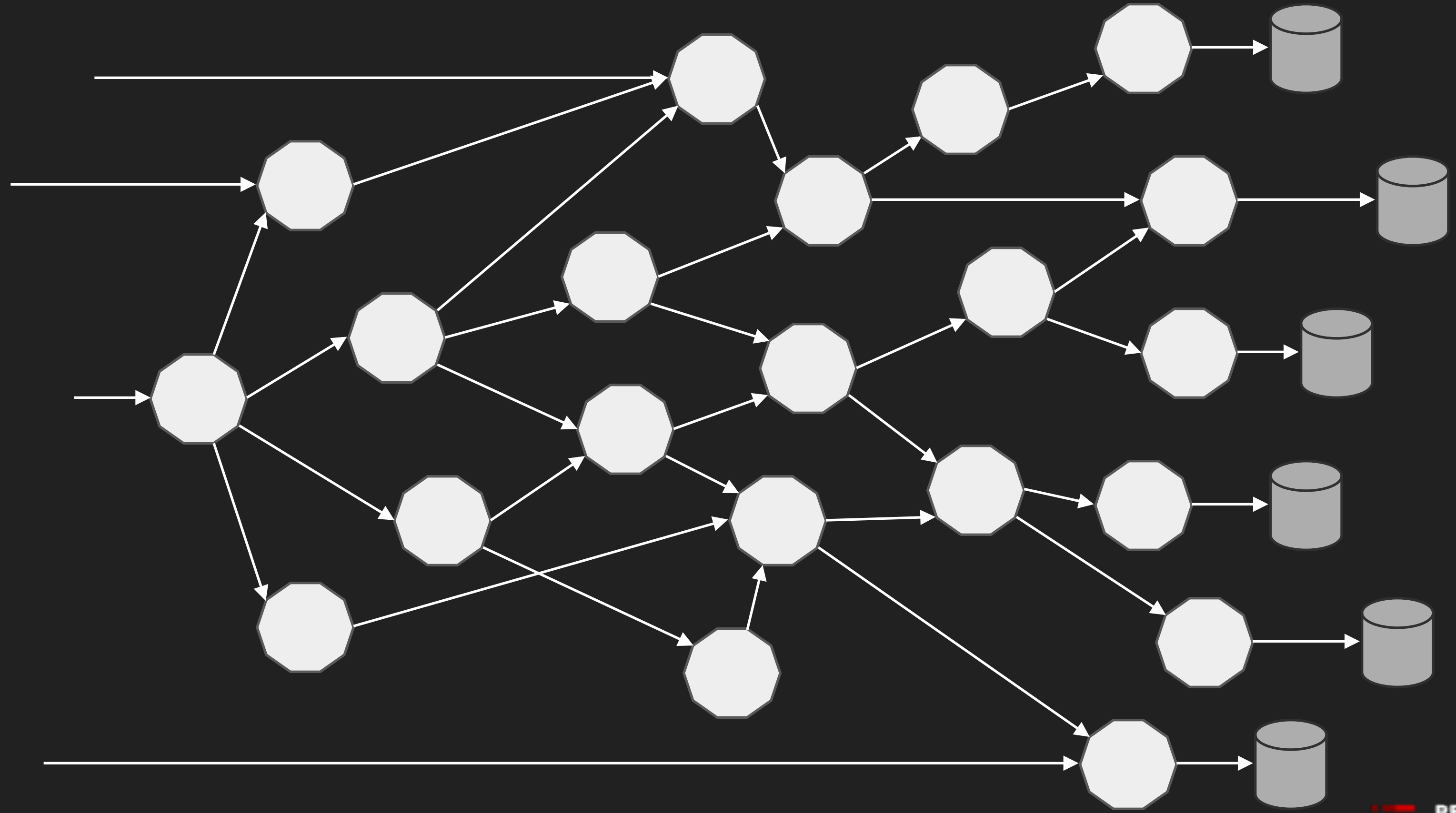


# Microservices own their Data



@burrsutter - bit.ly/serverlesskube

# Multiple Points of Entry



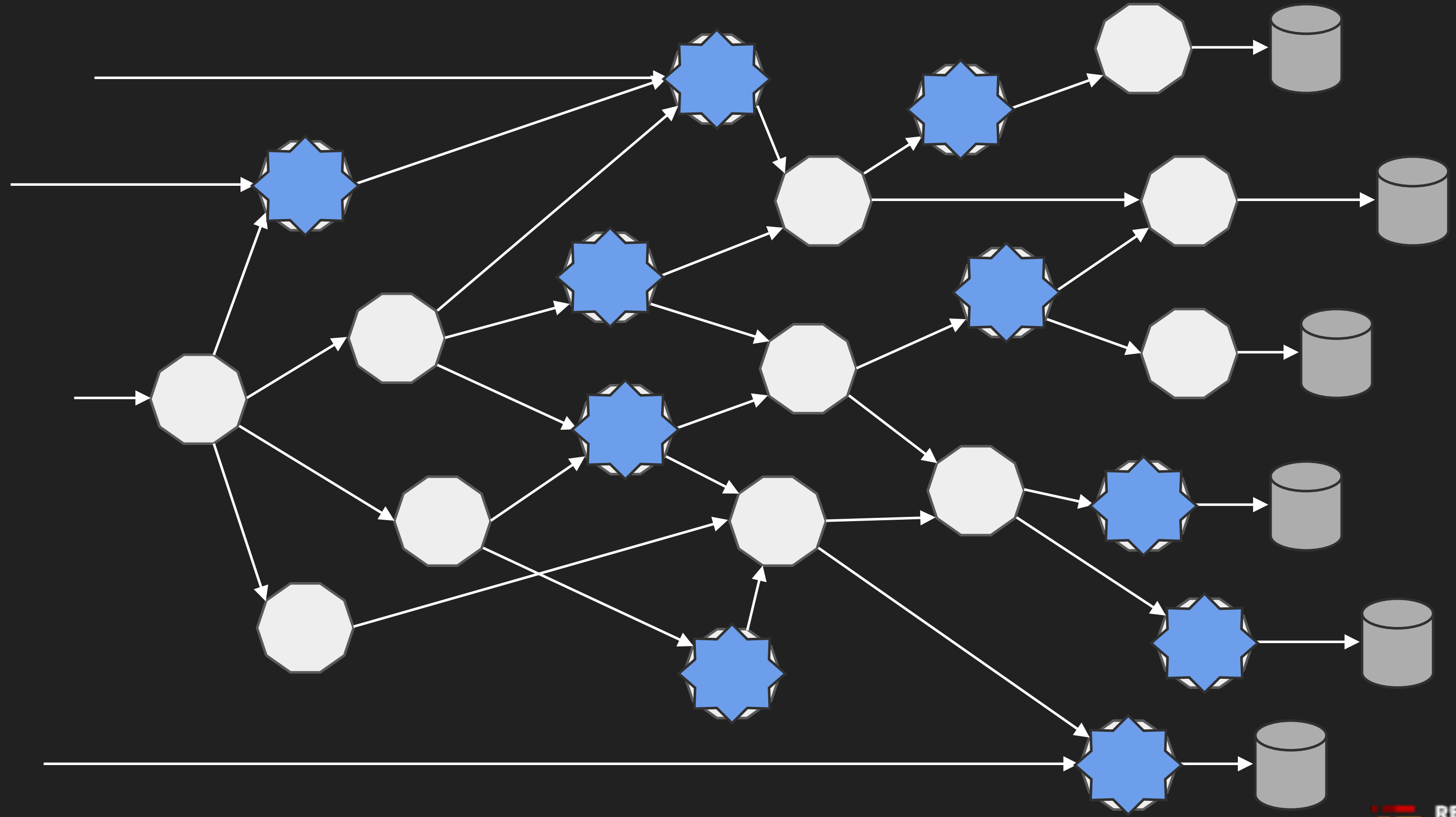
@burrsutter - bit.ly/serverlesskube

RED HAT DEVELOPER

“Change is the essential process of all of existence.”

—Spock

# Let there be Functions?



@burrsutter - bit.ly/serverlesskube

# Serverless

https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.pdf

# Cloud Native Computing Foundation

"Serverless computing refers to the concept of building and running applications that do not require server management. It describes 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."

https://www.cncf.io/blog/2018/02/14/cncf-takes-first-step-towards-serverless-computing/

# Serverless vs FaaS

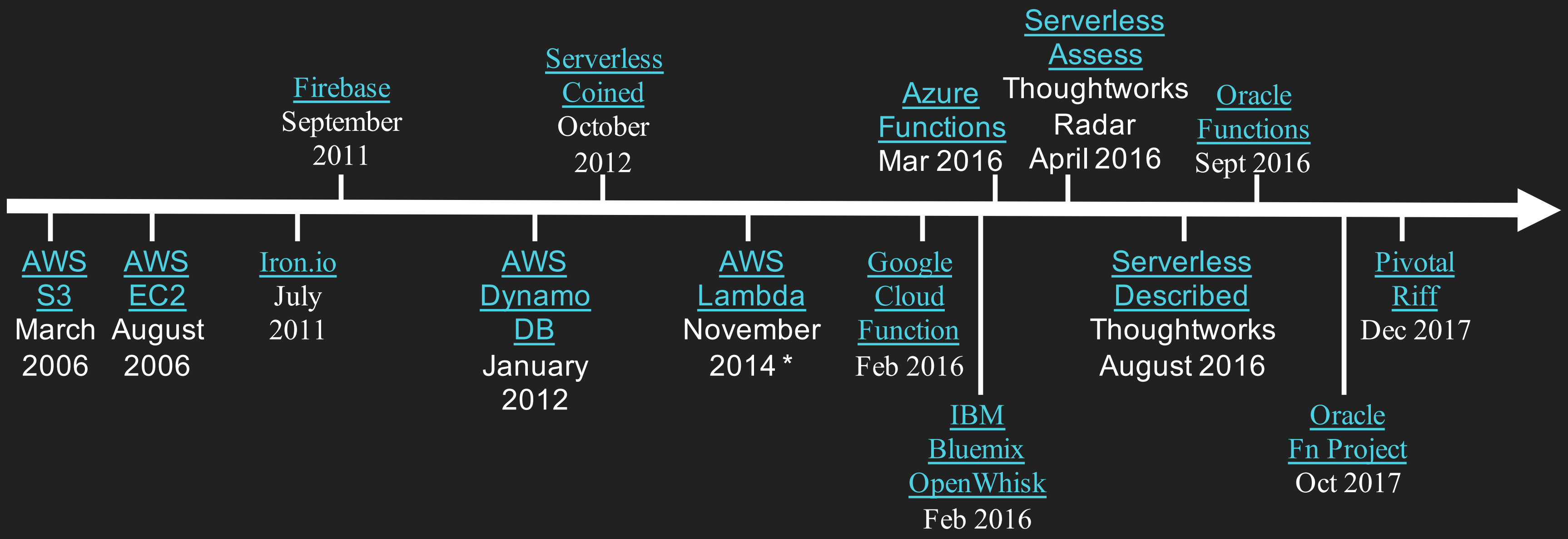
'...are application designs that incorporate third-party “Backend as a Service” (BaaS) services, and/or that include custom code run in managed, ephemeral containers on a “Functions as a Service” (FaaS) platform'

https://martinfowler.com/articles/serverless.html

'The survey defined FaaS as typically providing event-driven computing where developers run and manage application code with functions that are triggered by events'

https://thenewstack.io/add-it-up-serverless-faas/

# Short History of Serverless



AWS S3
March 2006

AWS EC2
August 2006

Iron.io
July 2011

Firebase
September 2011

AWS Dynamo DB
January 2012

Serverless Coined
October 2012

AWS Lambda
November 2014 *

Google Cloud Function
Feb 2016

IBM Bluemix OpenWhisk
Feb 2016

Azure Functions
Mar 2016

Serverless Assess
Thoughtworks Radar
April 2016

Serverless Described
Thoughtworks
August 2016

Oracle Functions
Sept 2016

Oracle Fn Project
Oct 2017

Pivotal Riff
Dec 2017

* Only supports JavaScript
Only for stateless, short-lived, simple applications

The first question is “is there a suitable service I can consume?” before “is there something I can buy and set up using a cloud provider?”.

[Thoughtworks Nov 29 2017]

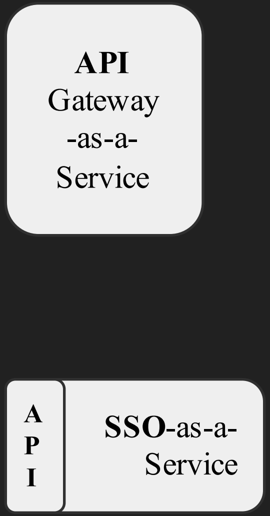
# It is Serverless because of BaaS/SaaS (managed by another party services).

# It is all about the Services

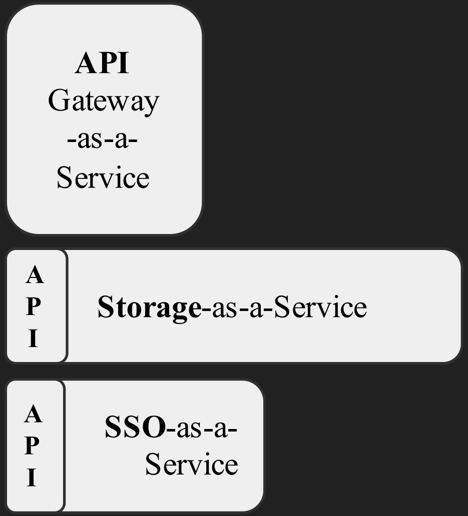
# HTTP Input/Output Service

**API**
Gateway
-as-a-
Service

# Authentication Service



# File Storage Service



**API** Gateway -as-a- Service

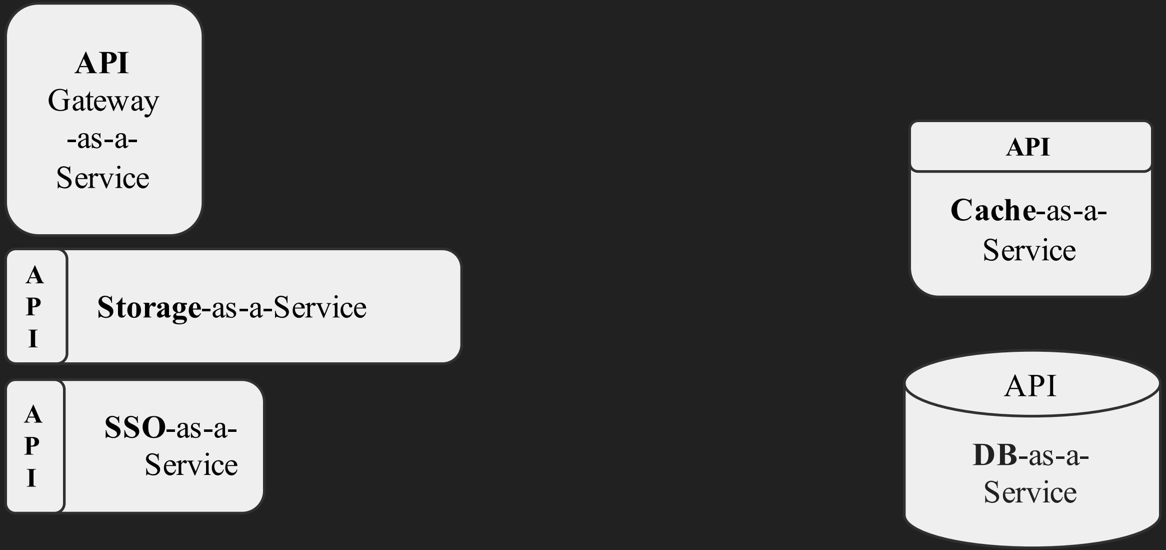
**A P I**

**Storage**-as-a-Service

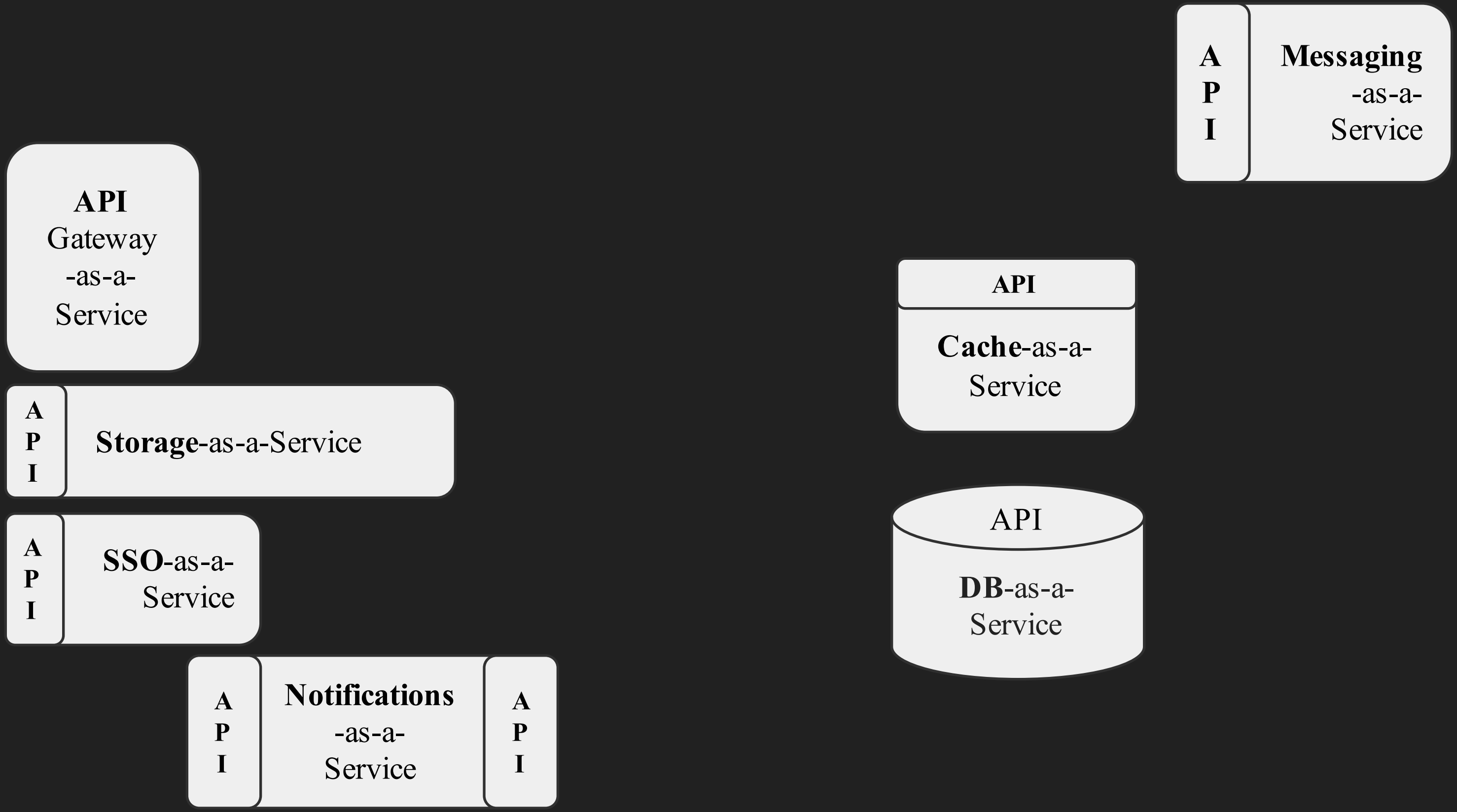
**A P I**

**SSO**-as-a- Service

# Data Services

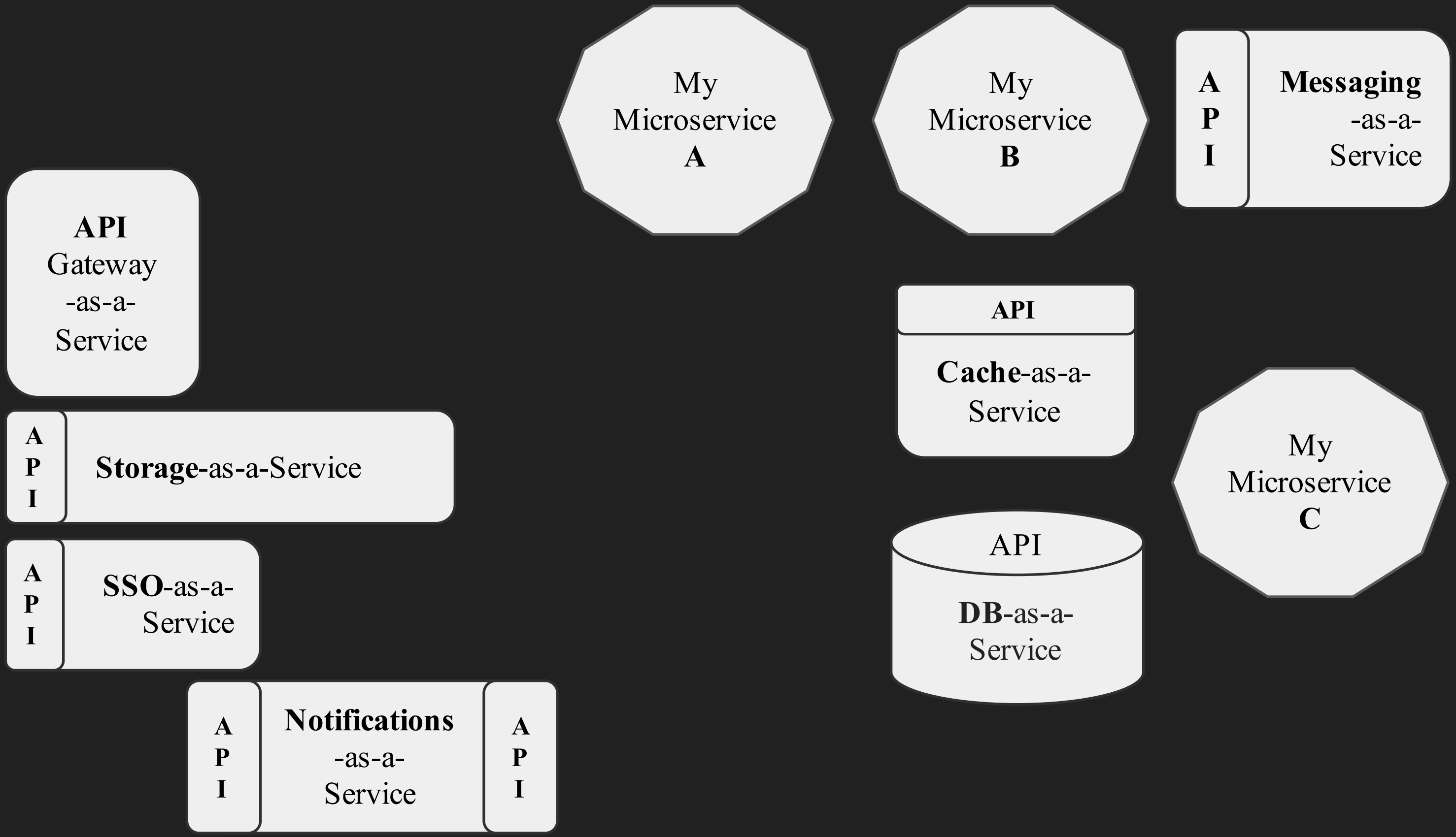


# Connectivity Services

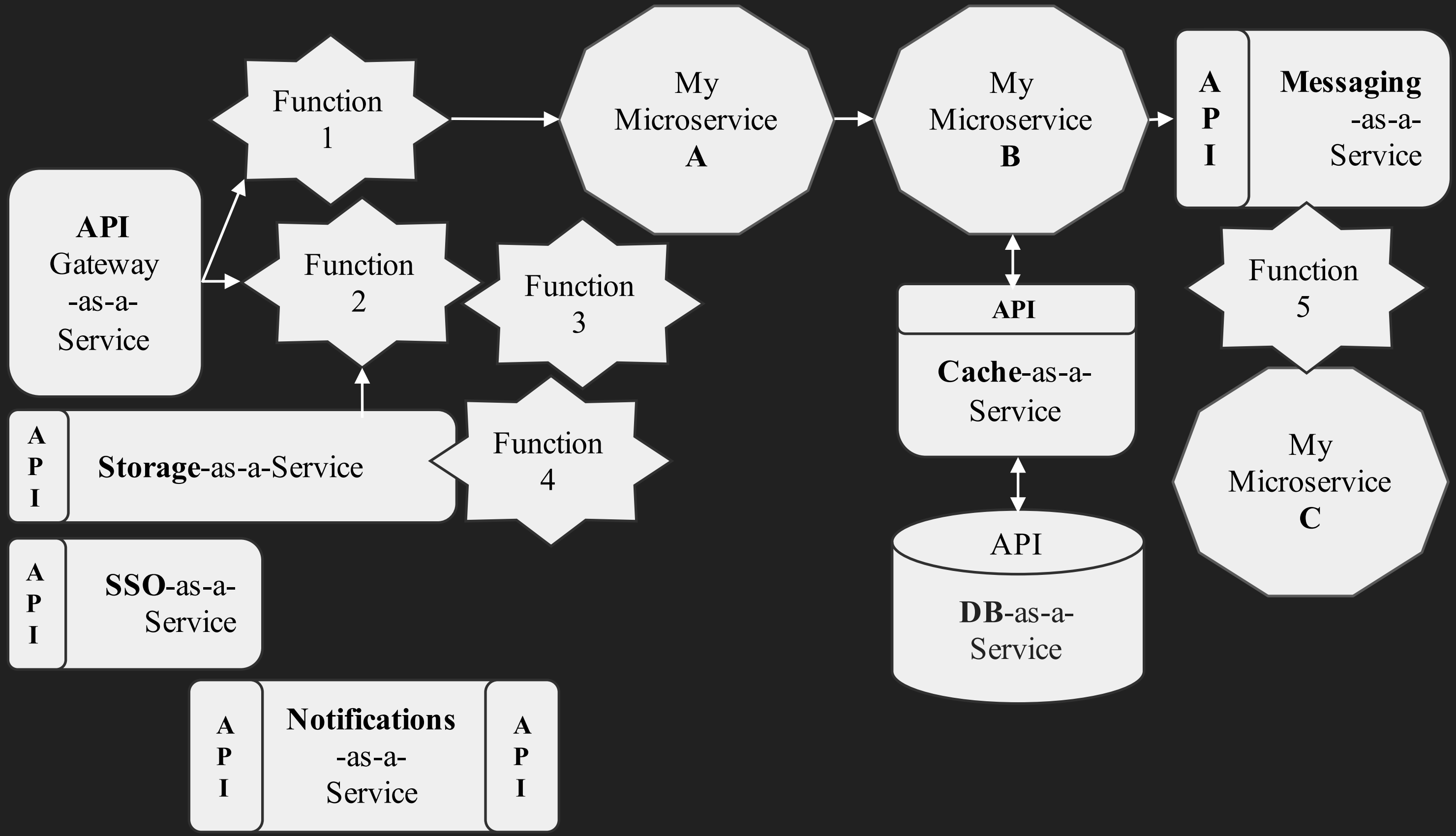


@burrsutter - bit.ly/serverlesskube

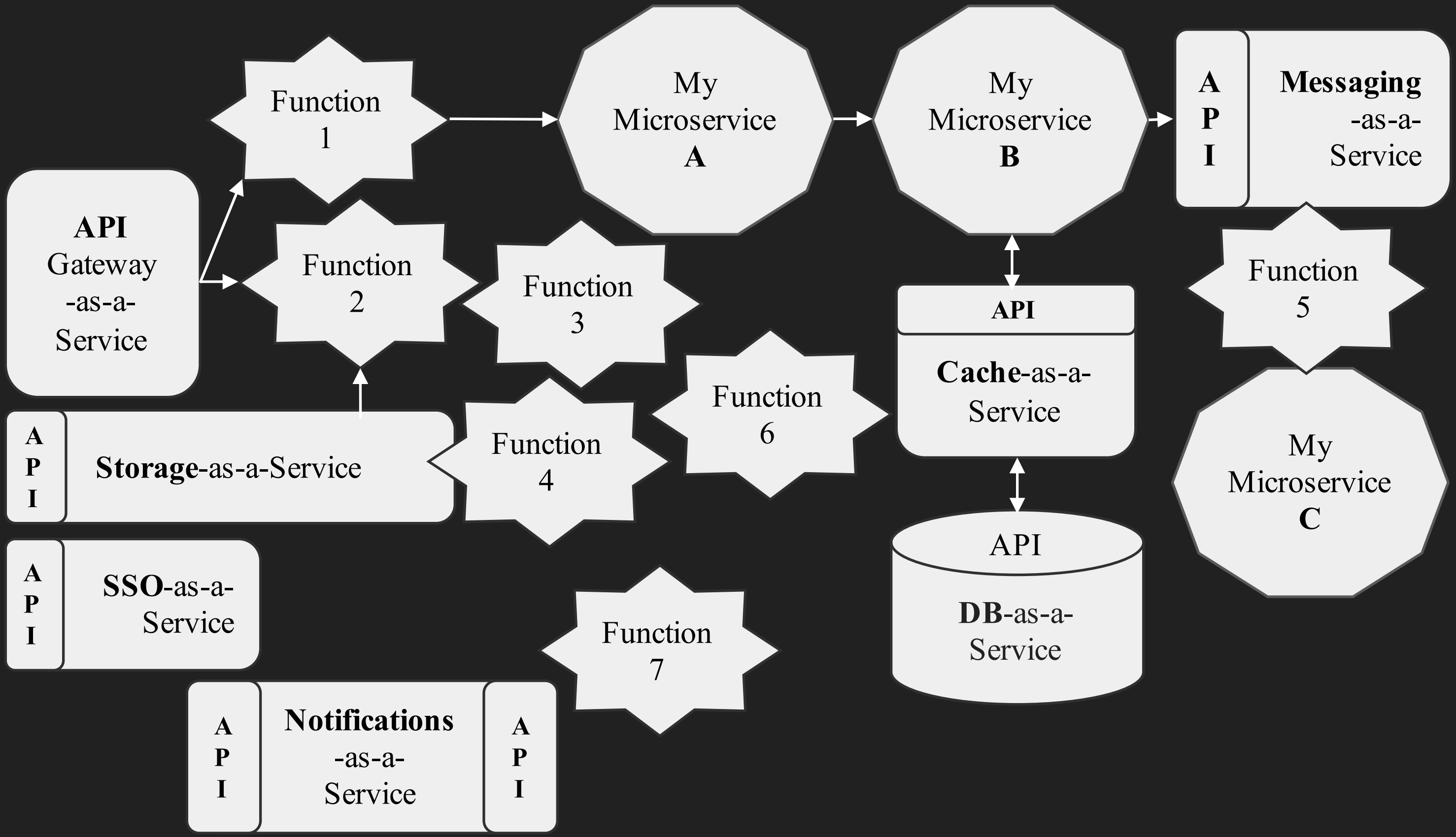
# Your Containerized Services



# Event-Driven Input

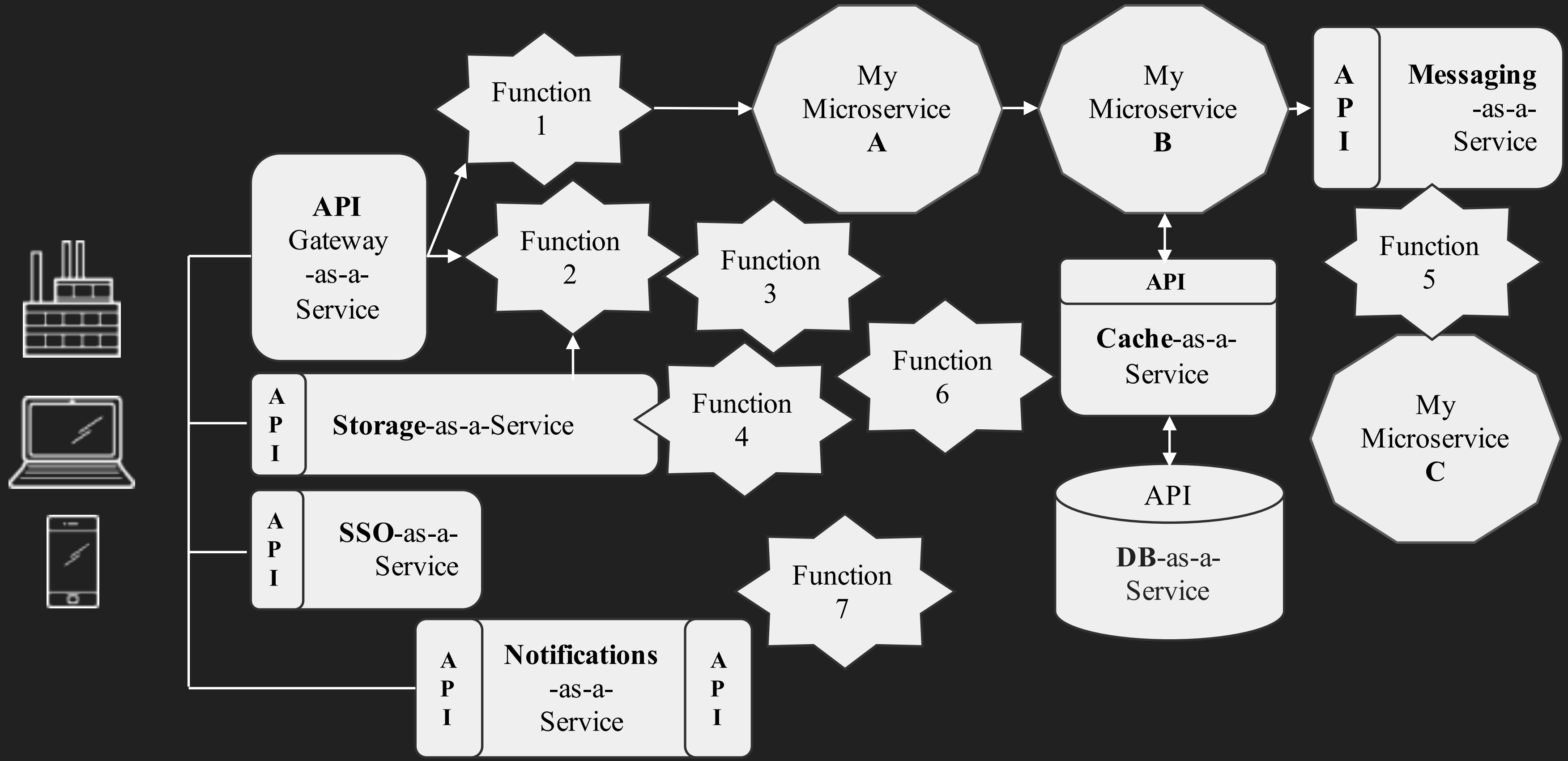


# Event-Driven Output



@burrsutter - bit.ly/serverlesskube

# Synergy



Function 1
My Microservice A
My Microservice B
API
Messaging -as-a- Service
API Gateway -as-a- Service
Function 2
Function 3
API
Cache-as-a-Service
Function 5
Function 6
API
Storage-as-a-Service
Function 4
My Microservice C
API
DB-as-a-Service
API
SSO-as-a-Service
Function 7
API
Notifications -as-a- Service
API

## Microservices

Your Control
Long-Lived Processes
Known Programming Model
Often Sync Request-Response

Mature:
IDE Integration
Debuggers
Tracers
Monitoring
CI/CD

## Serverless Functions

Cloud Control
Short-Lived Processes
New Programming Model
Event-Driven Async

Immature:
?

# Good and Bad about Serverless

☺️

- Automatic Scalability
- Automatic Cost Reduction
- Quicker and Easier Development
- Better Capacity Utilization
- Delivery speed

😞

- Debugging
- Deployment and Architectural complexity
- Learning curve
- Vendor Lock-in
- Monitoring

https://hackernoon.com/serverless-survey-77-delivery-speed-4-dev-workdays-mo-saved-26-aws-monthly-bill-d99174f70663

# FaaS

# FaaS Kubernetes Players

fn

Kubeless

DISPATCH

APACHE OpenWhisk™

riff

fission

OPENFAAS

nuclio

# Kubernetes/OpenShift Review

```
mvn package
docker build
kubectl apply -f deploy.yml
kubectl apply -f service.yml
```

# Knative

https://github.com/knative

# Knative Announced July 24

FaaS Kubernetes Players

fn

Kubeless

DISPATCH

APACHE OpenWhisk™

riff

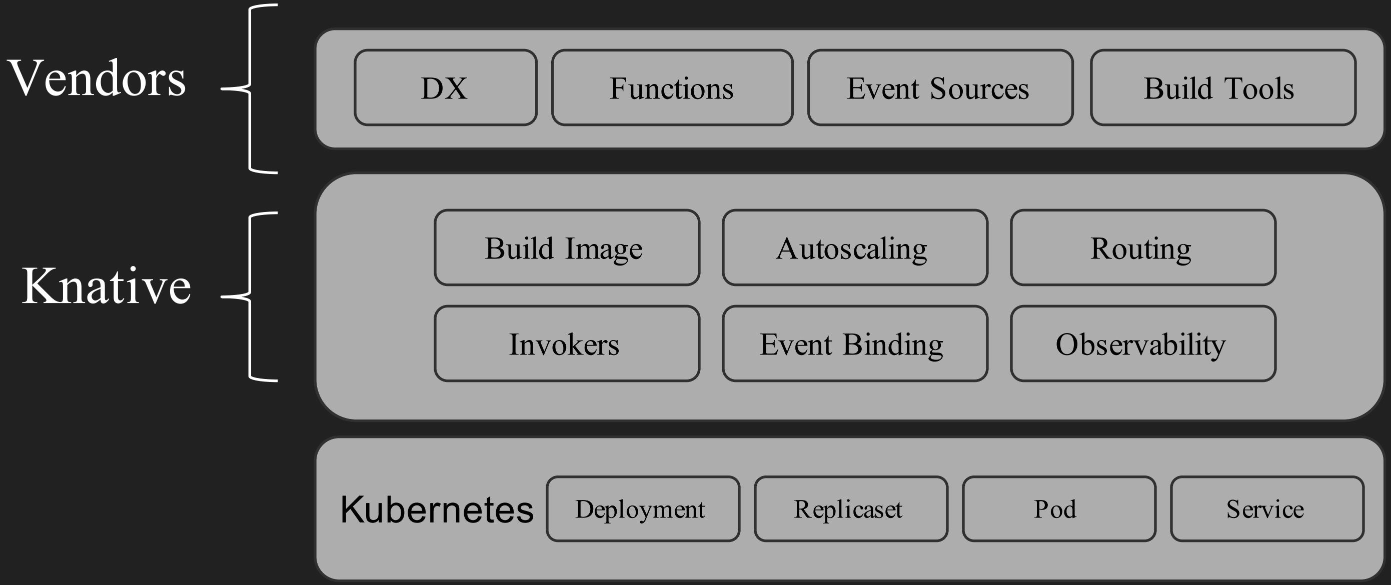
fission

OPENFAAS

nuclio

# Why?

# Primitives


Vendors
DX
Functions
Event Sources
Build Tools
Knative
Build Image
Autoscaling
Routing
Invokers
Event Binding
Observability
Kubernetes
Deployment
Replicaset
Pod
Service

# What is Knative?

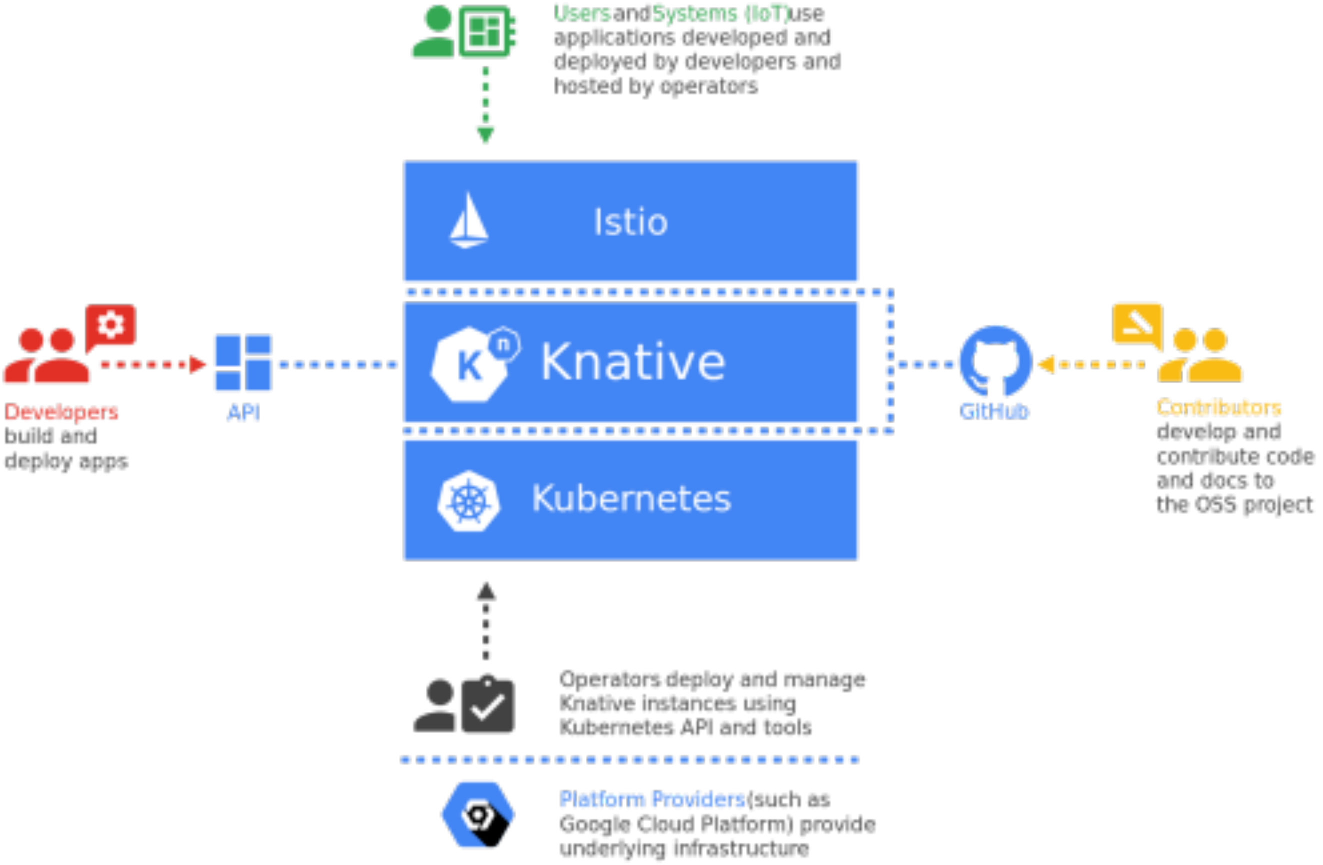
"Kubernetes-based platform to build, deploy, and manage modern serverless workloads."

"Essential base primitives for all"

"Knative provides a set of middleware components that are essential to build modern, source-centric, and container-based applications that can run anywhere: on premises, in the cloud, or even in a third-party data center"

# Interesting Capabilities

- Scale-to-zero: No pod == no memory
- Scale-from-zero: Traffic spike starts N pods
- Configurations & Revisions - built-in Blue/Green
- In-Cluster Image Building
- Traffic splitting
- Eventing System



Users and Systems (IoT) use applications developed and deployed by developers and hosted by operators
Istio
Knative
Kubernetes
Developers build and deploy apps
API
GitHub
Contributors develop and contribute code and docs to the OSS project
Operators deploy and manage Knative instances using Kubernetes API and tools
Platform Providers (such as Google Cloud Platform) provide underlying infrastructure

# Knative's Primary Components

Serving

Build

Eventing

# Knative Serving

```
kubectl get crd | grep serving
 configurations.serving.knative.dev
 revisions.serving.knative.dev
 routes.serving.knative.dev
 services.serving.knative.dev
```

# Knative Serving Autoscaler

```
kubectl -n knative-serving edit
configmap config-autoscaler
```

```
container-concurrency-target-default: "1"

scale-to-zero-grace-period: 30s

stable-window: 30s
```

# Knative Build

```
kubectl get crd | grep build

 builds.build.knative.dev

 buildtemplates.build.knative.dev

 clusterbuildtemplates.build.knative.dev
```

# Knative Eventing

```
kubectl get crd | grep eventing

 channels.eventing.knative.dev
 cronjobsources.sources.eventing.knative.dev
 githubsources.sources.eventing.knative.dev
 kuberneteseventsources.sources.eventing.knative.dev
 containersources.sources.eventing.knative.dev
 subscriptions.eventing.knative.dev
```

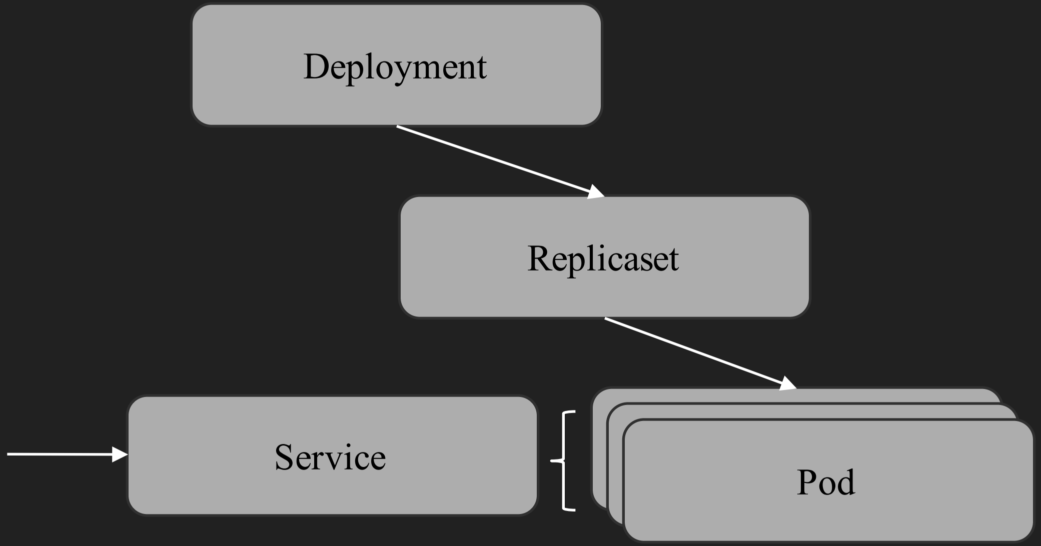
# Exercises

bit.ly/knative-tutorial

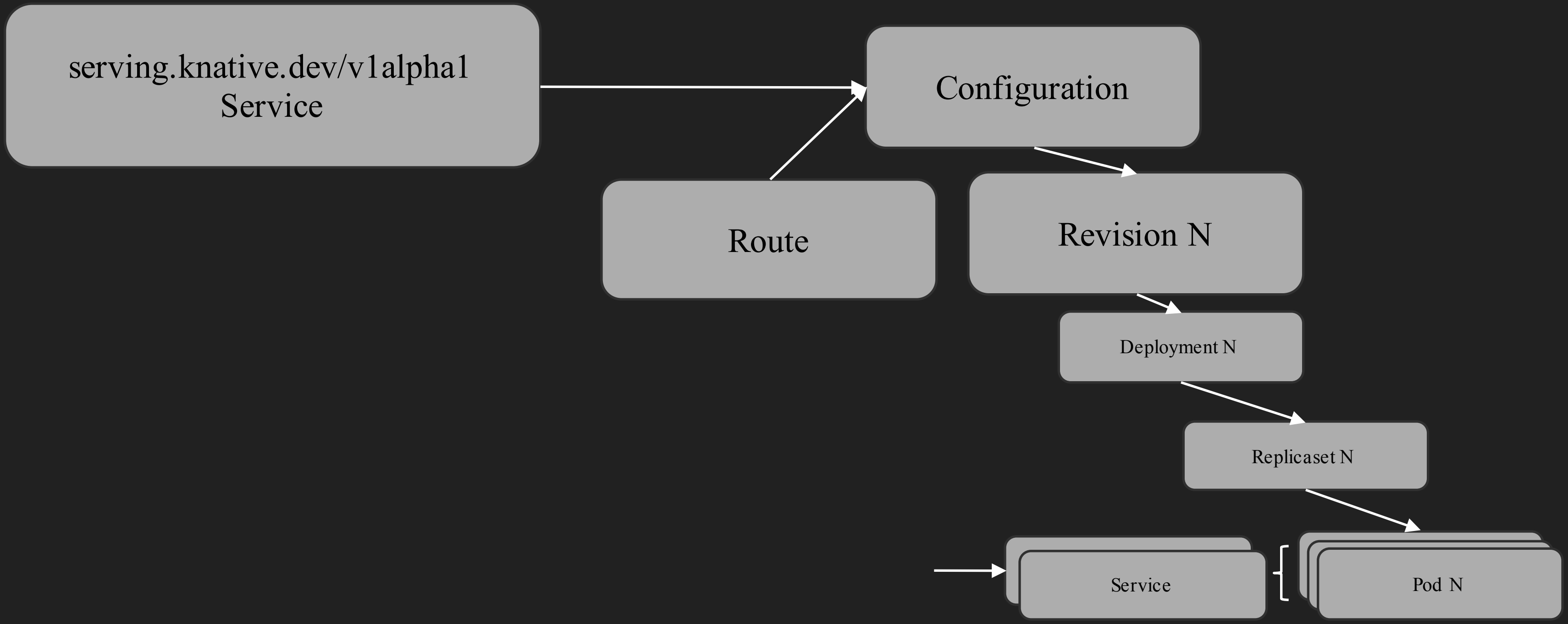
https://github.com/burrsutter/scripts-knative

# Knative Serving

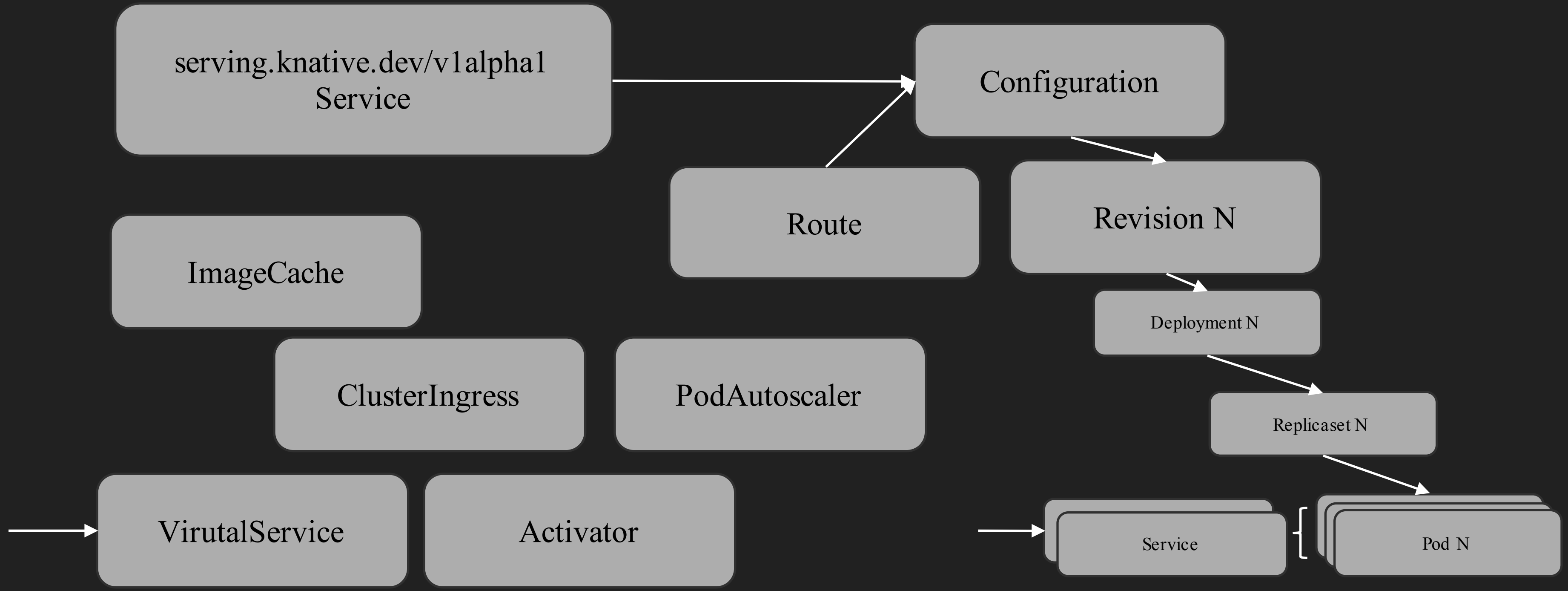
# kubectl apply -f Deployment.yaml



# kubectl apply -f ksvc.yaml



# kubectl apply -f ksvc.yaml



serving.knative.dev/v1alpha1
Service

Configuration

Route

Revision N

ImageCache

Deployment N

ClusterIngress

PodAutoscaler

Replicaset N

VirutalService

Activator

Service

Pod N

# Resources

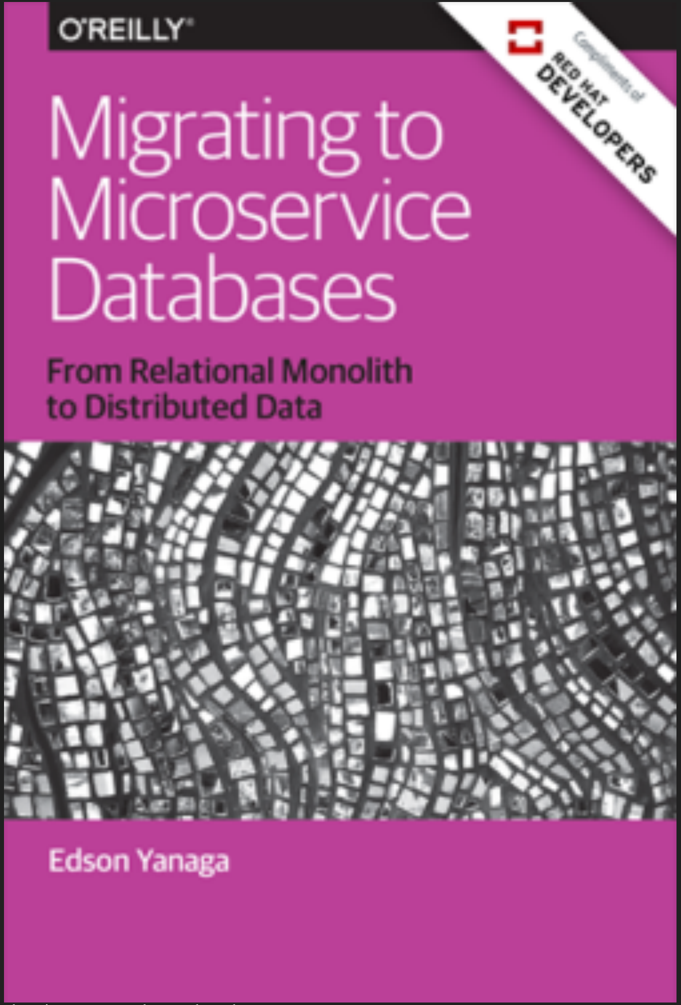
https://blog.openshift.com/knative-serving-your-serverless-services/

https://blog.openshift.com/knative-building-your-serverless-service/

https://blog.openshift.com/knative-serving-your-serverless-services/

# Cloud events

- Define specification for Cloud Events
    - Effort via CNCF's Serverless Working Group
- https://cloudevents.io/



O'REILLY

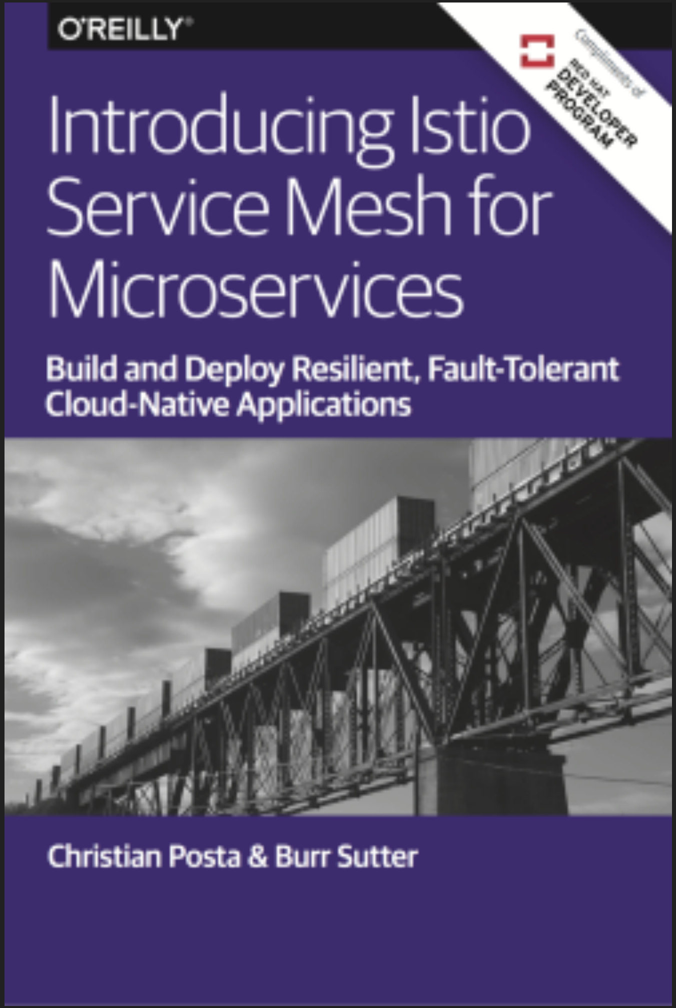
Compliments of RED HAT DEVELOPERS

# Migrating to Microservice Databases

From Relational Monolith to Distributed Data

Edson Yanaga

bit.ly/mono2microdb



O'REILLY®

Compliments of RED HAT DEVELOPER PROGRAM

Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant Cloud-Native Applications

Christian Posta & Burr Sutter

bit.ly/istio-book

bit.ly/javamicroservicesbook

O'REILLY
Compliments of RED HAT DEVELOPERS
Microservices for Java Developers
A Hands-On Introduction to Frameworks & Containers
Christian Posta

**Free** eBooks from developers.redhat.com

Microservices Introductory Materials

Demo: bit.ly/msa-instructions

Slides: bit.ly/microservicesdeepdive

Video Training: bit.ly/microservicesvideo

Kubernetes for Java Developers

9 Steps to Awesome with Kubernetes

bit.ly/reactivemicroservicesbook

O'REILLY
Compliments of RED HAT DEVELOPER PROGRAM
Building Reactive Microservices in Java
Asynchronous and Event-Based Application Design
Clement Escoffier

Advanced Materials

bit.ly/istio-tutorial

bit.ly/knative-tutorial

learn.openshift.com/servicemesh

bit.ly/serverlesskube

# Raffle Rules (applicable in the real)

1. Follow: @burrsutter
2. With picture of the session
3. Mention @burrsutter
4. With hashtag #oredev