

12 Ways of the Cloud Native Warrior

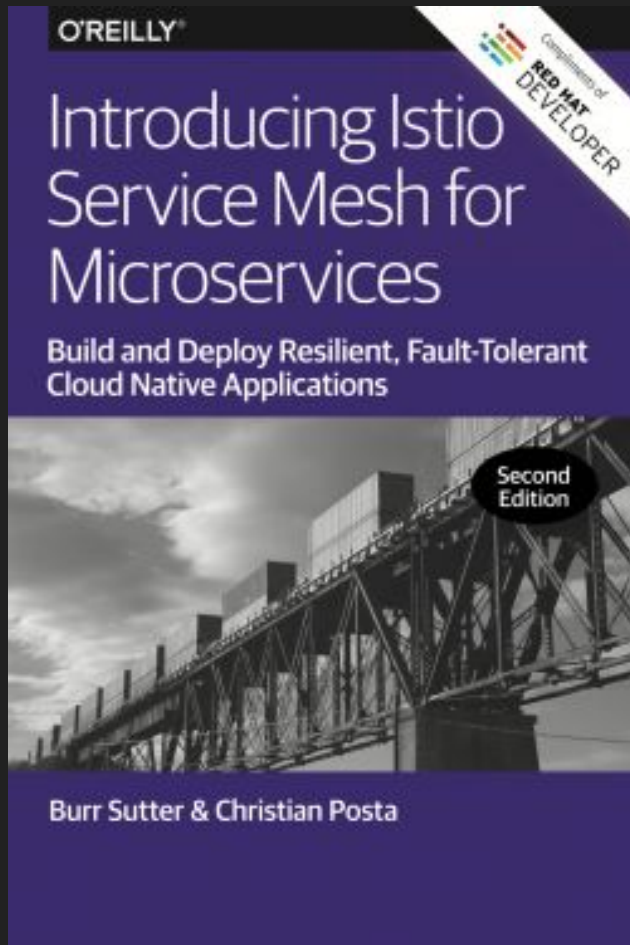
Burr Sutter (burrsutter.com)



@burrsutter - bit.ly/12wayscloud



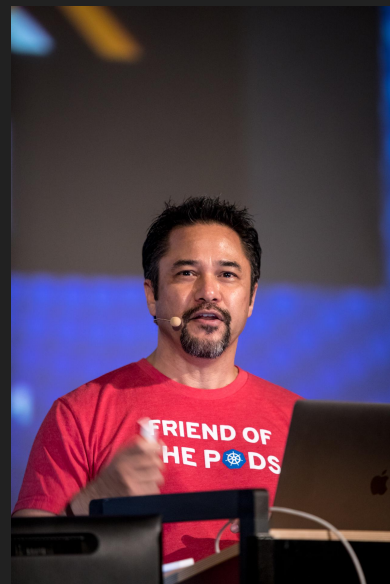
RED HAT
DEVELOPER



bit.ly/istiobook

Recording from Devovx BE 2018:

https://www.youtube.com/watch?v=ZpbXSdzp_vo



"excellent stuff.. worth 3 hours spending to watch this video.. Really loved it. Will give a complete picture on Kubernetes and how it works" - Pradeep

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?

Journey to Awesomeness

DevOps



Self-Service,
On-Demand,
Elastic
Infrastructure



Automation



CI & CD
Deployment
Pipeline



Advanced
Deployment
Techniques



Microservices



@burrsutter

Force Multipliers & Superpowers

Wonder Woman Style

Boomerang
Tiara



Re-Org to
DevOps

Indestructible
Bracelets



Self-Service,
On-Demand,
Elastic
Infrastructure

Lasso of Truth



Automation
Puppet, Chef,
Ansible,
Kubernetes

Amazonian
Shield



CI & CD
Deployment
Pipeline

Hand-to-Hand
Combat



Advanced
Deployment
Techniques

Superhuman
Strength



Microservices



12 Ways

1. DevOps & Agile
2. Skill Portability
3. On-Demand - Never Wait & Always Automate
4. Pipeline Throughput, Deployment Interval, Optimize for Learning
5. Deployment Techniques
6. Application Architecture vs Application Infrastructure
7. Linux Containers
8. Kubernetes/OpenShift & Hybrid Cloud
9. Service Mesh: Observability, Security, Traffic Control (Istio)
10. Serverless Serving (Knative)
11. Build In Cluster (Tekton)
12. Beyond HTTP (AMQP, Kafka, Knative Eventing, Camel-K)



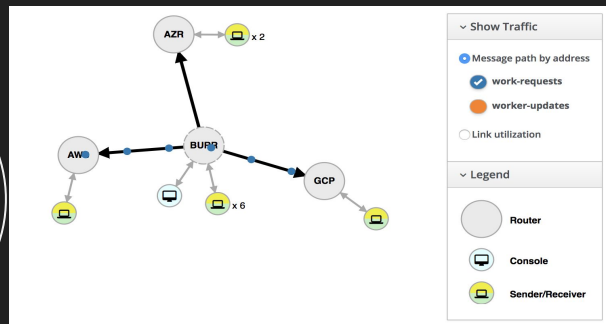
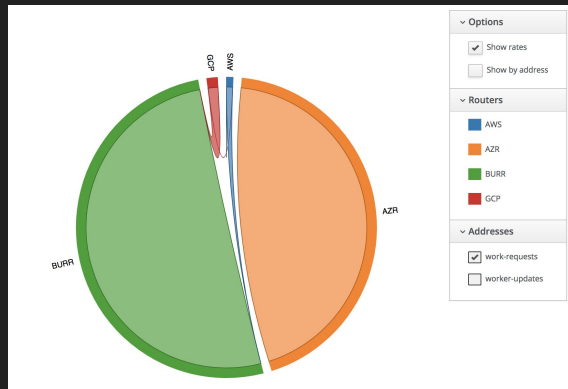
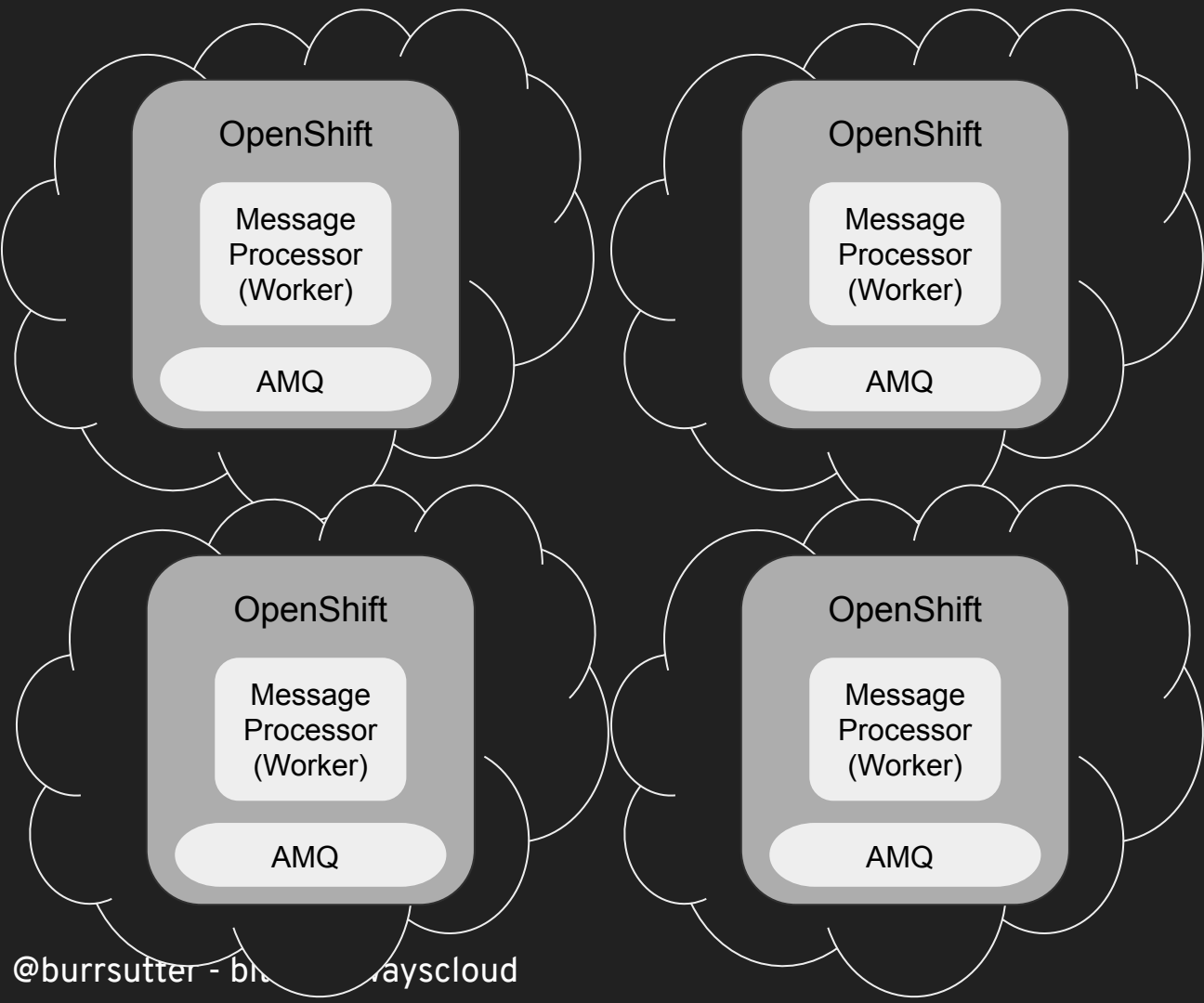
*“ Adapt what is useful,
reject what is useless,
and add what is
specifically
your own.*

— Bruce Lee

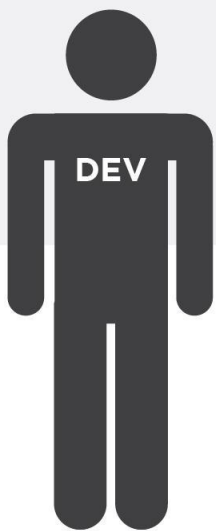
@burrsutter



RED HAT
DEVELOPER







WALL OF CONFUSION





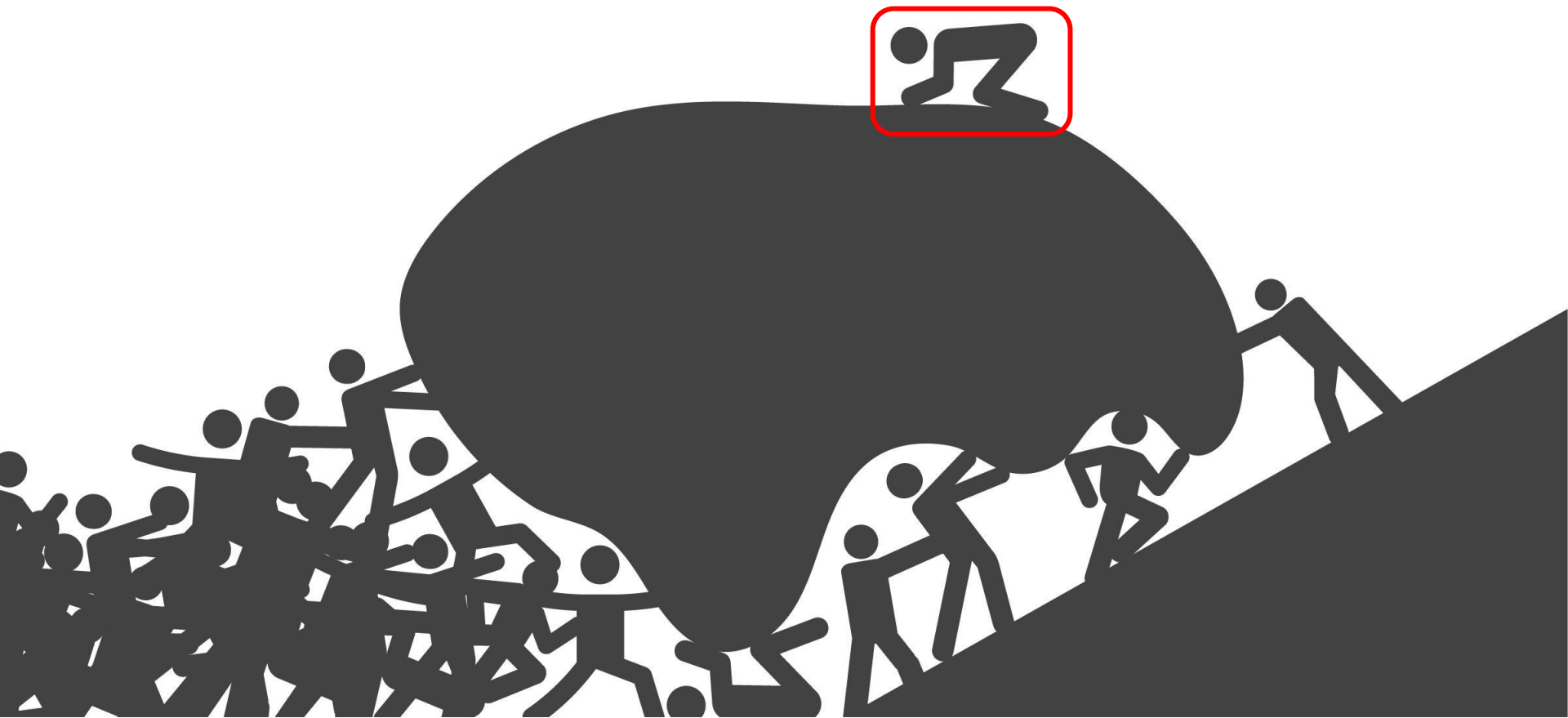
WALL OF CONFUSION

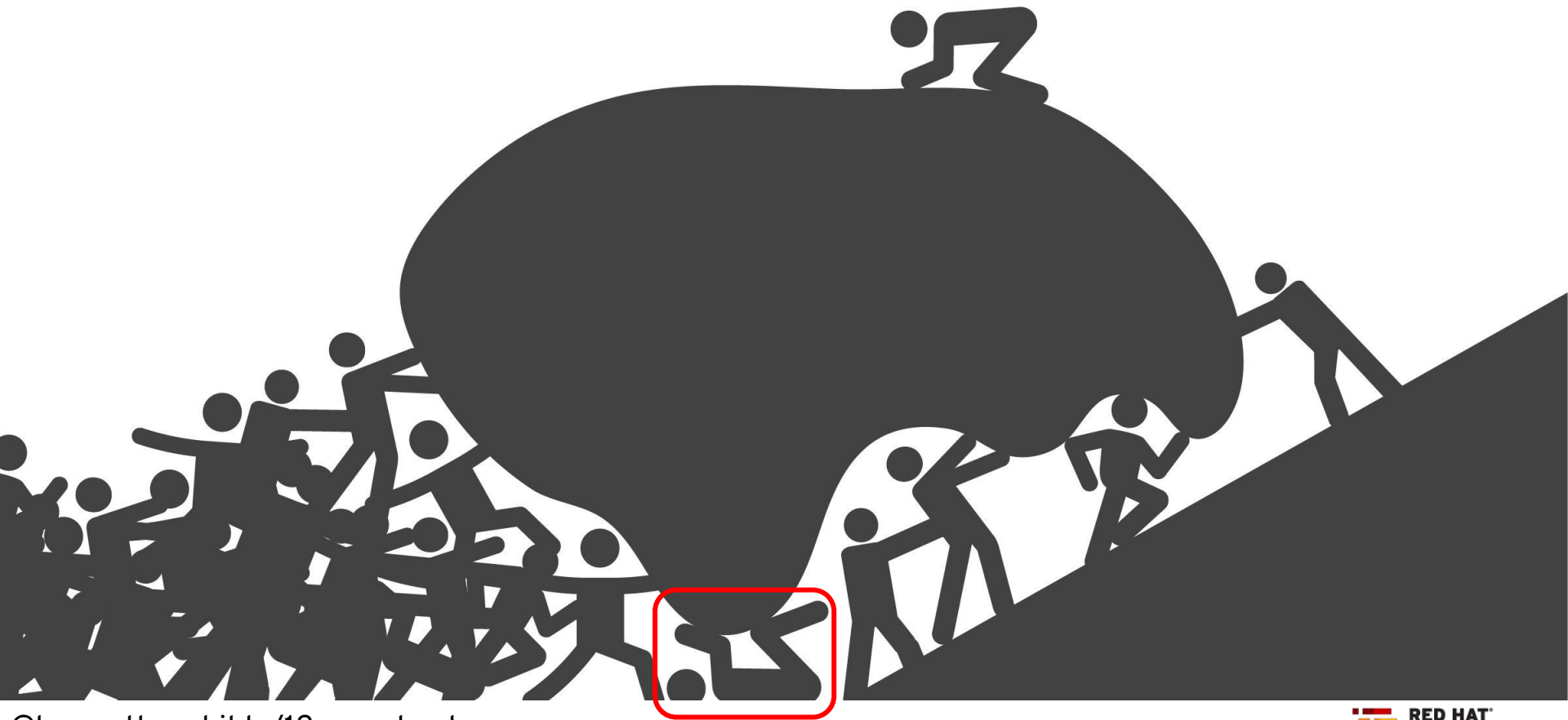




@burrsutter - bit.ly/12wayscloud



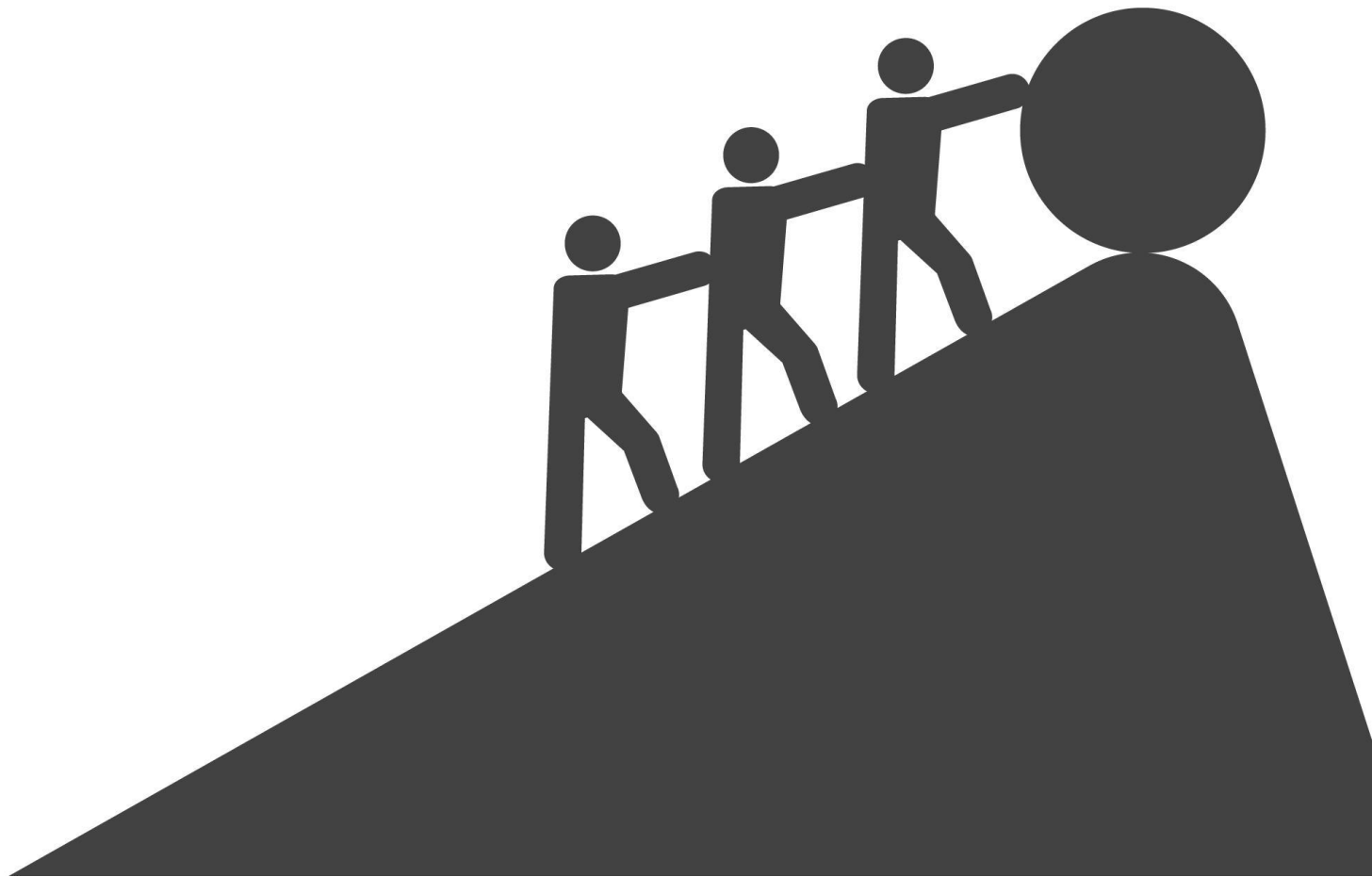




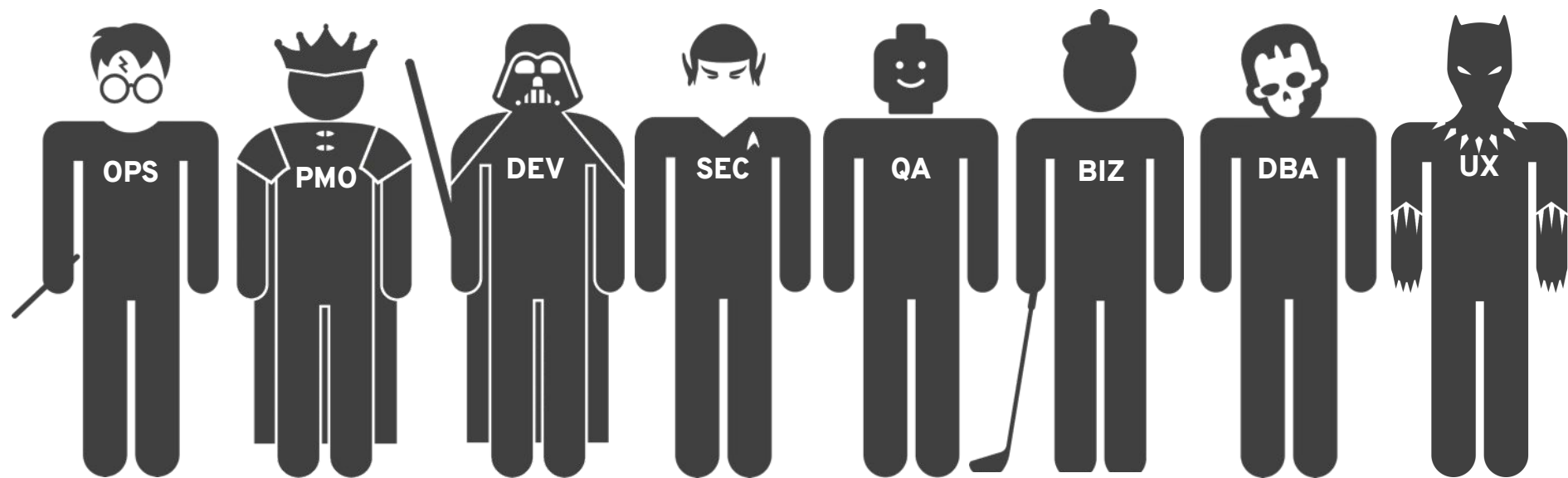
@burrsutter - bit.ly/12wayscloud

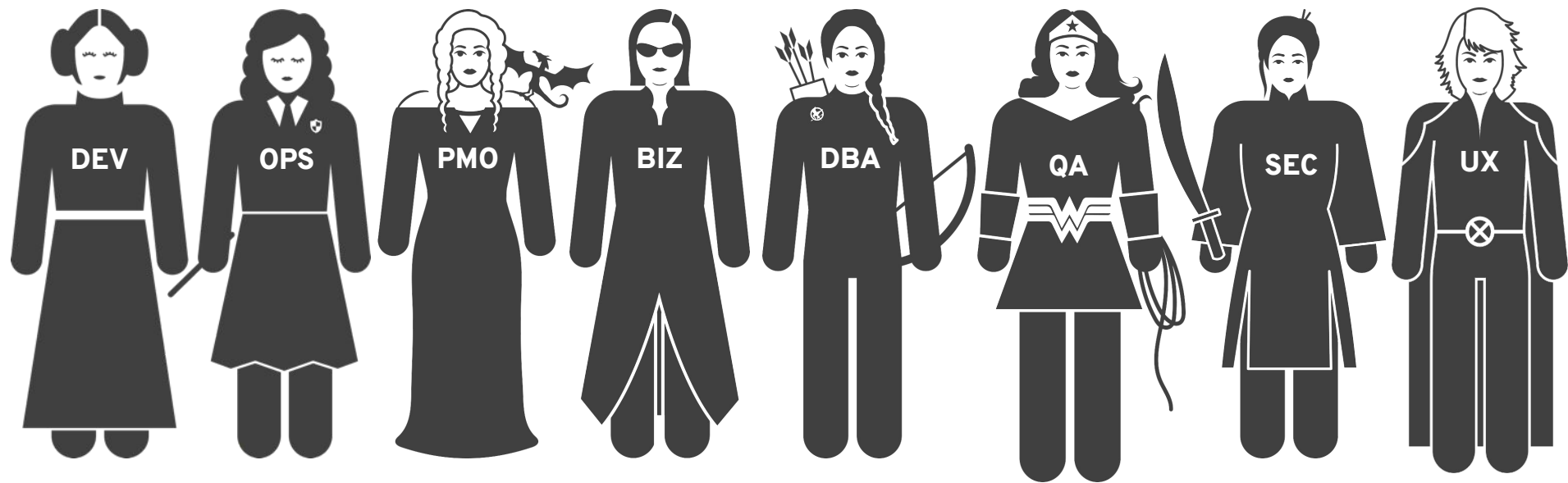


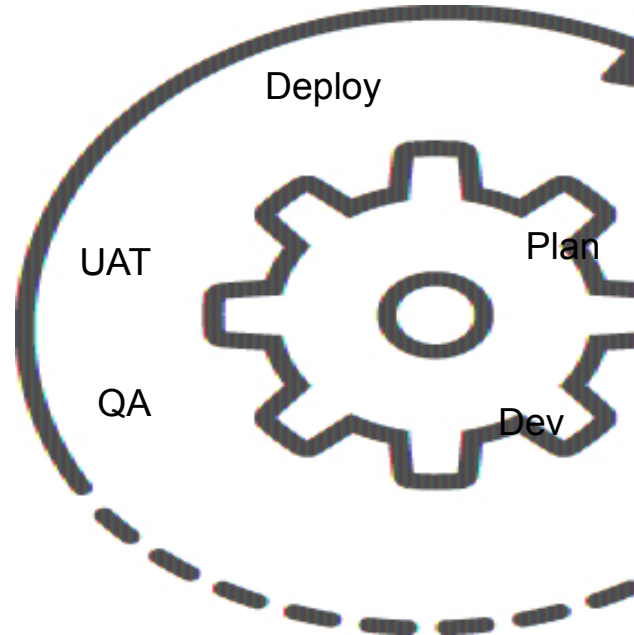
@burrsutter - bit.ly/12wayscloud



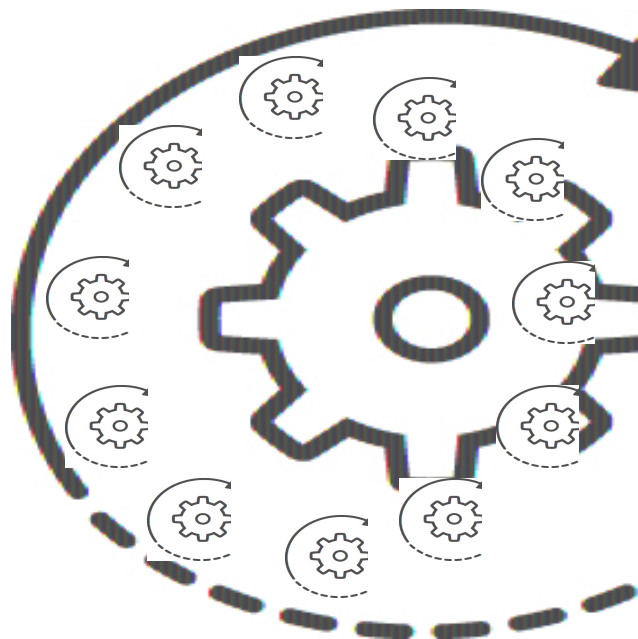




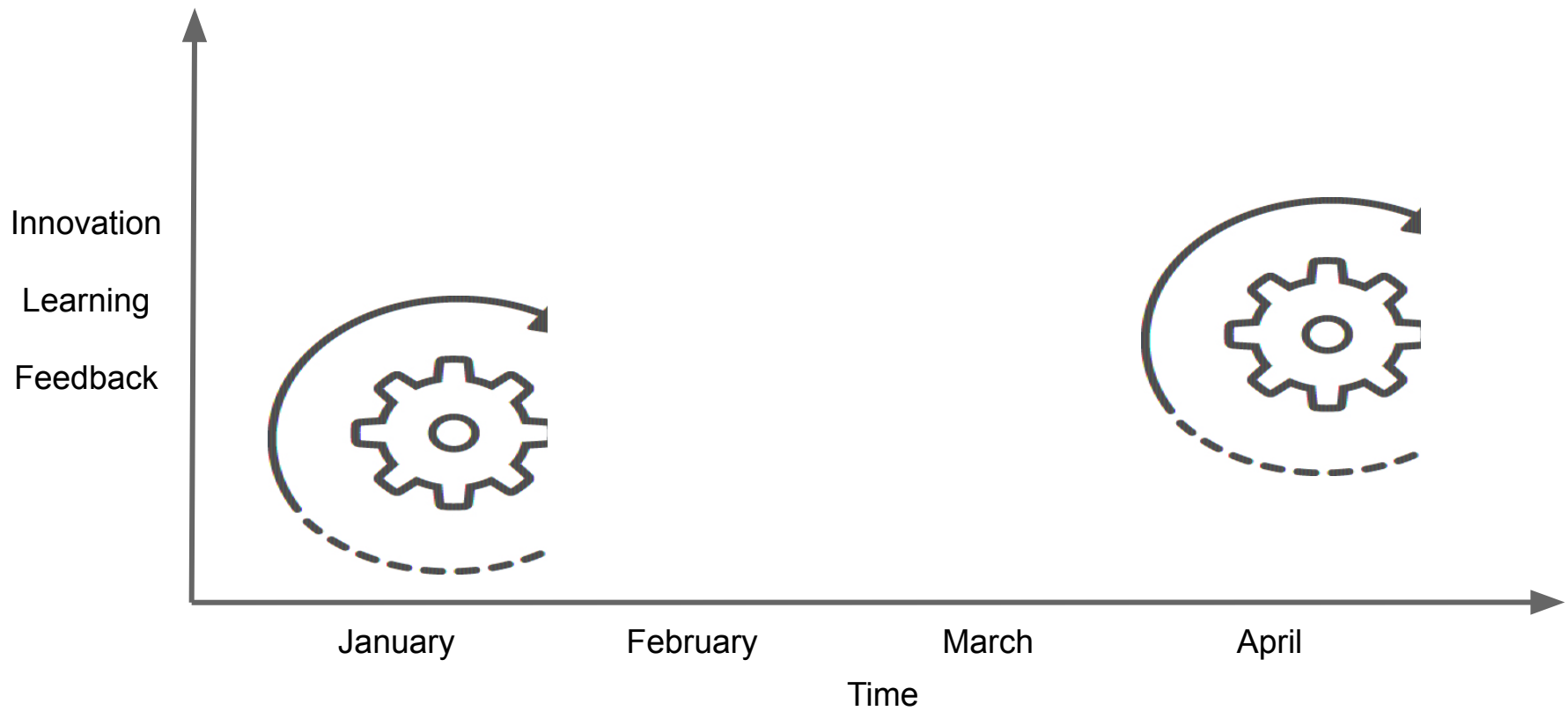


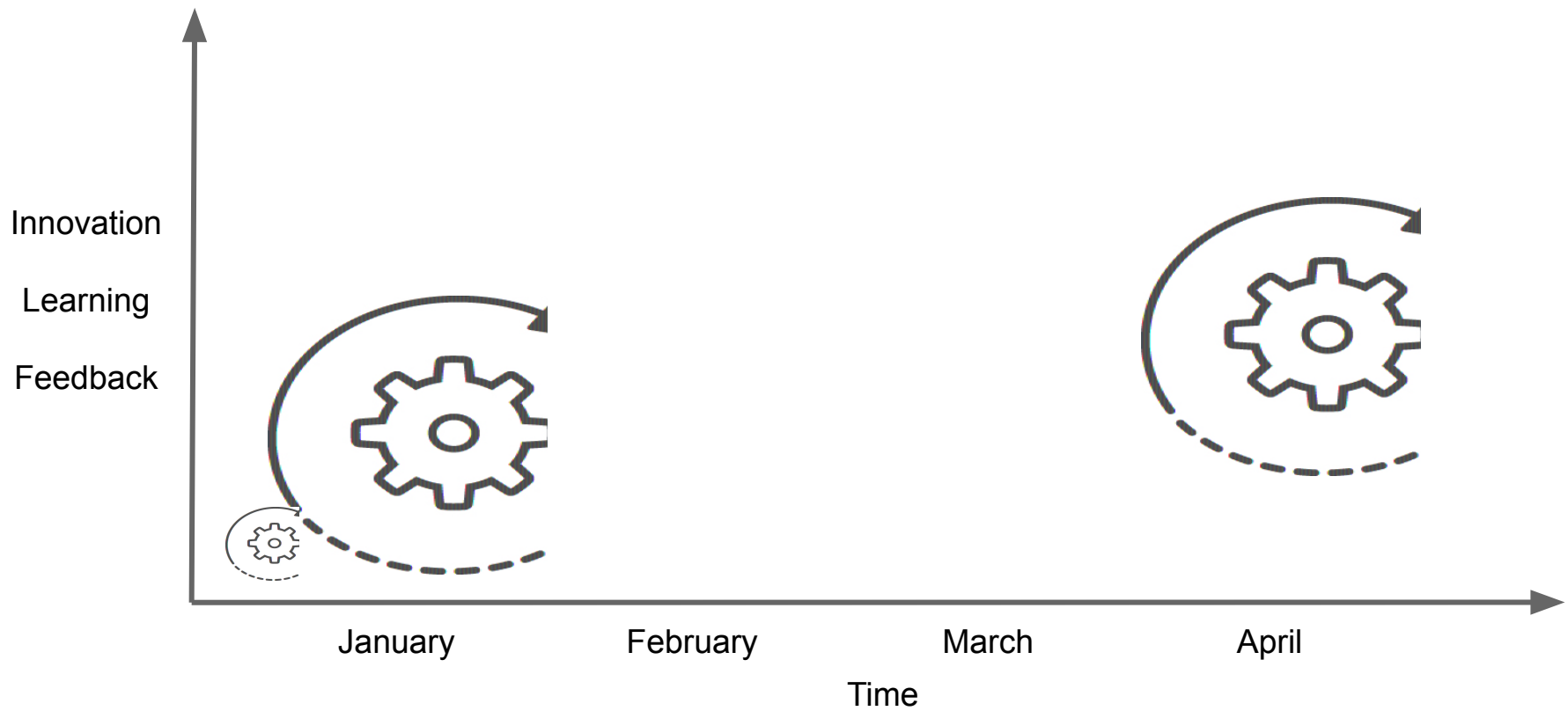


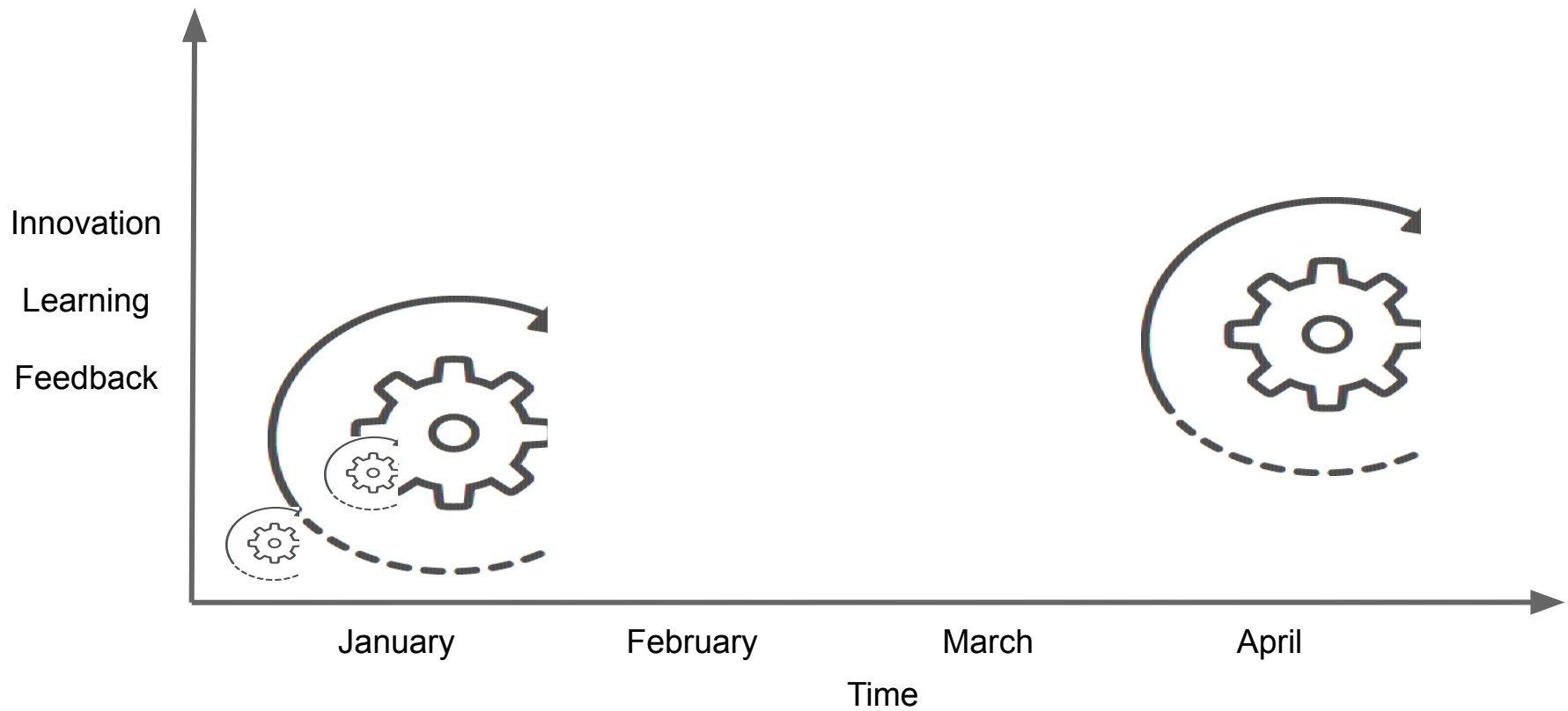
3 Month
Deployment Cycle
3 Months BEFORE you gain Feedback and Learn

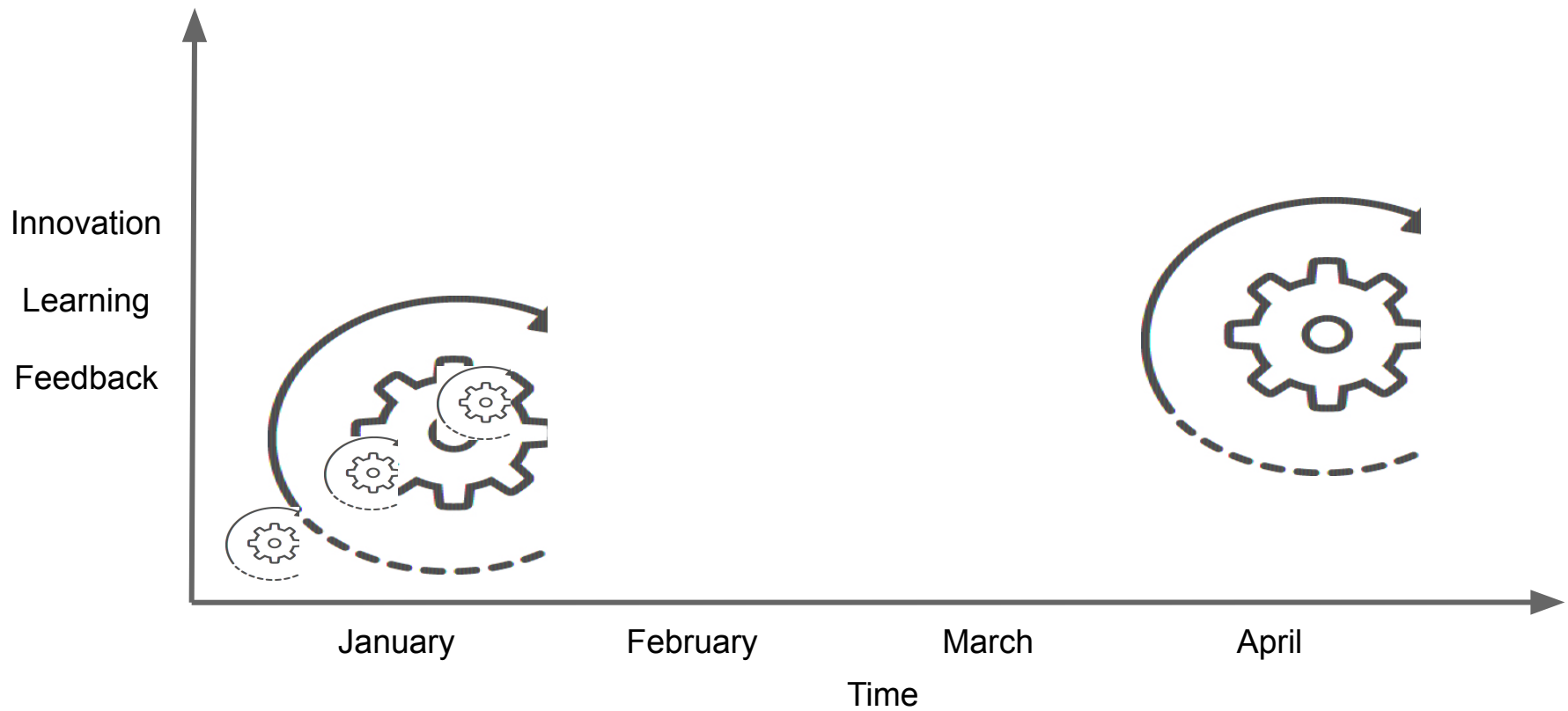


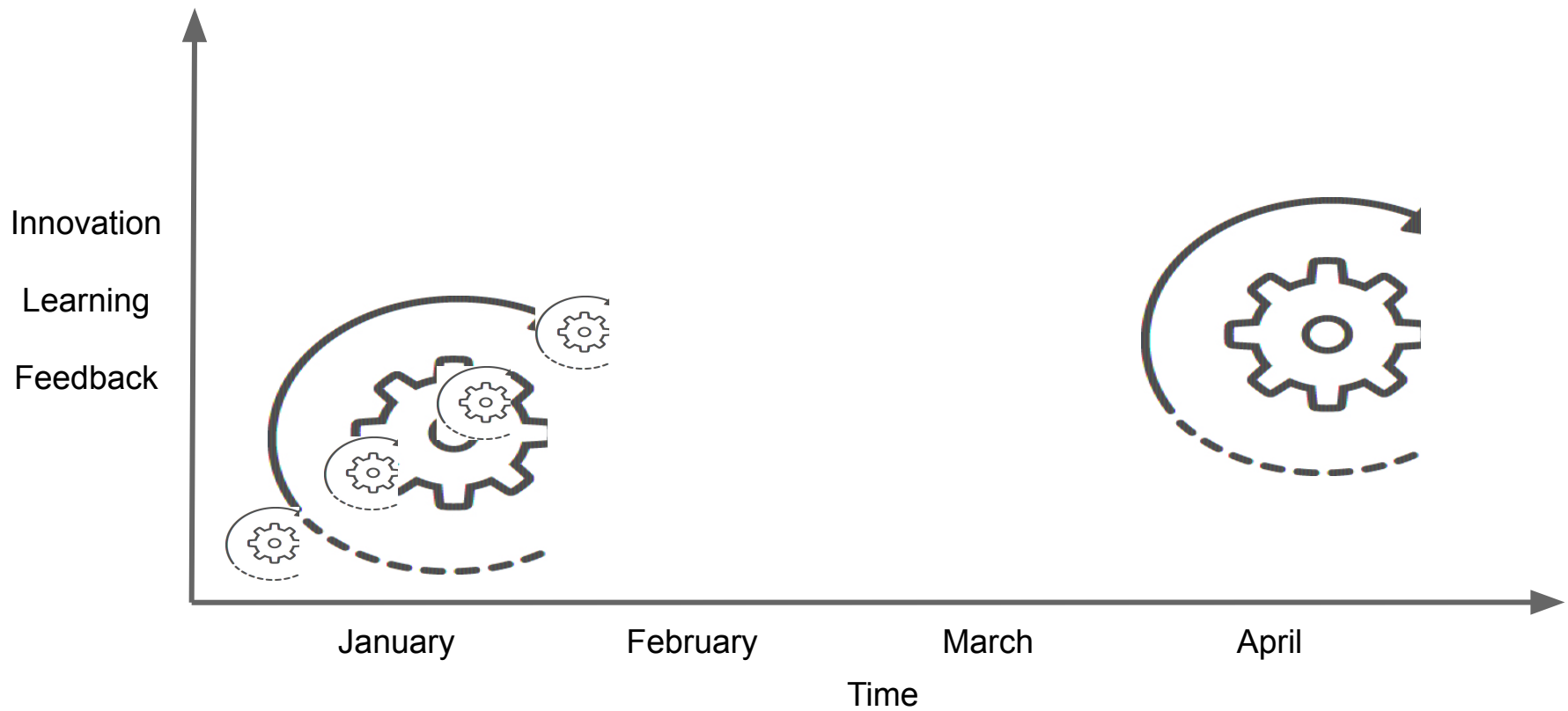
3 Months
Vs
1 Week

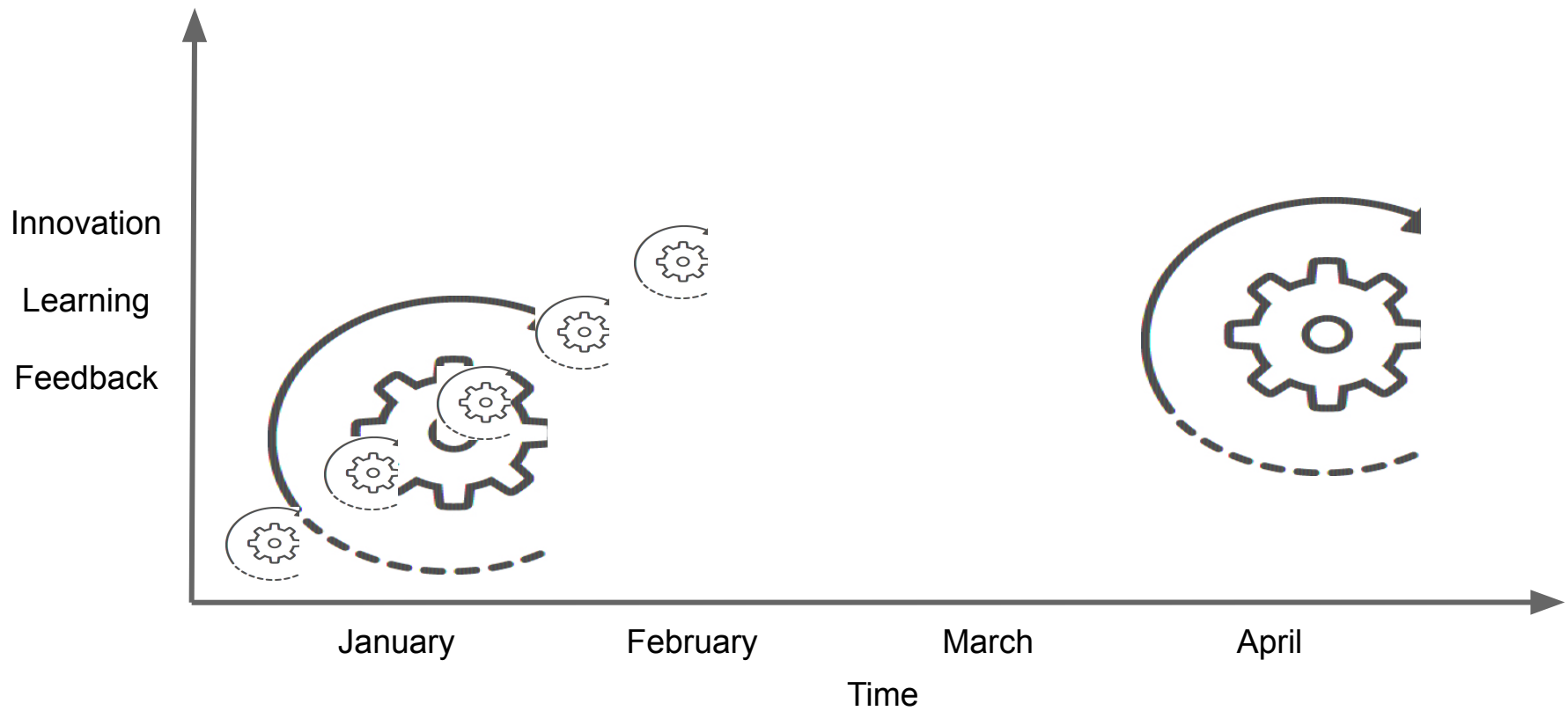


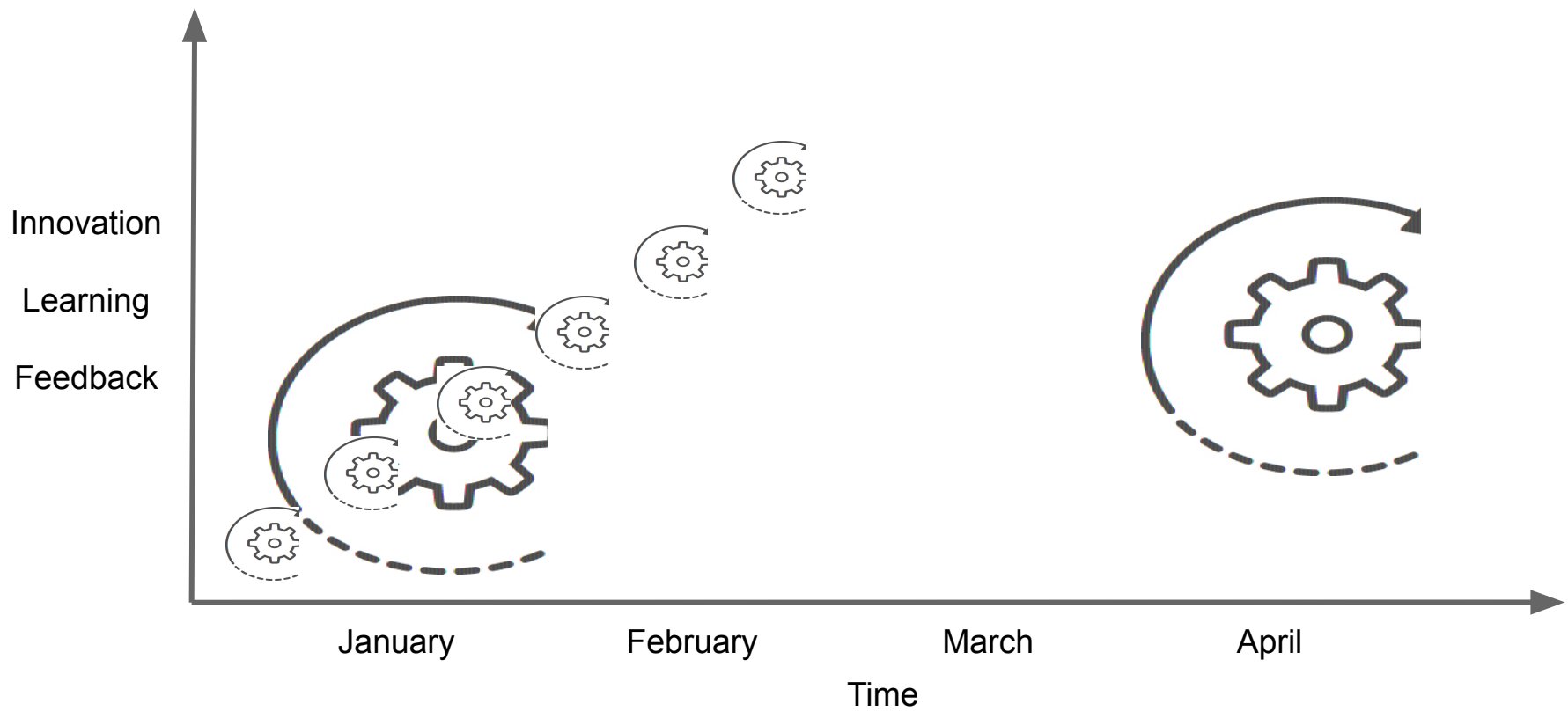


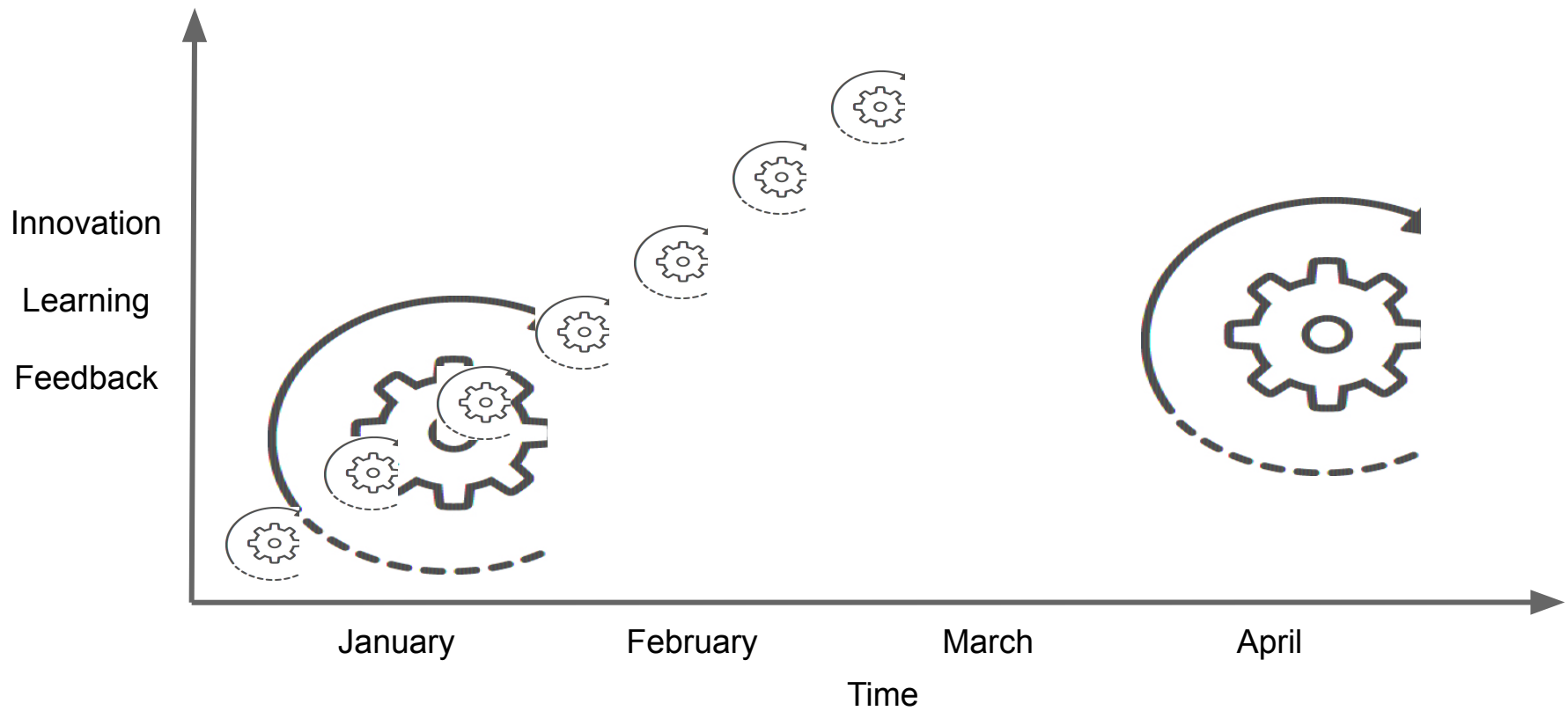


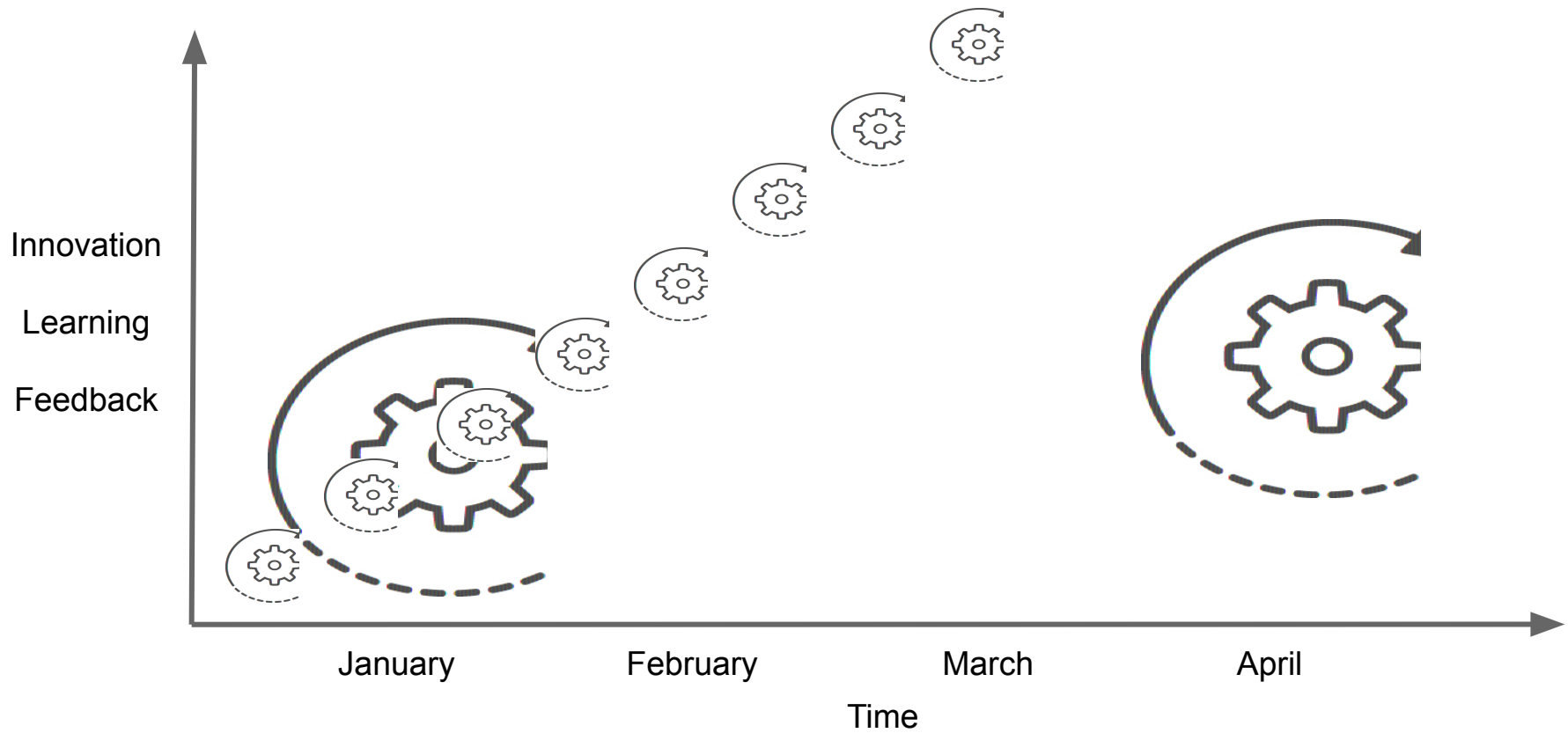


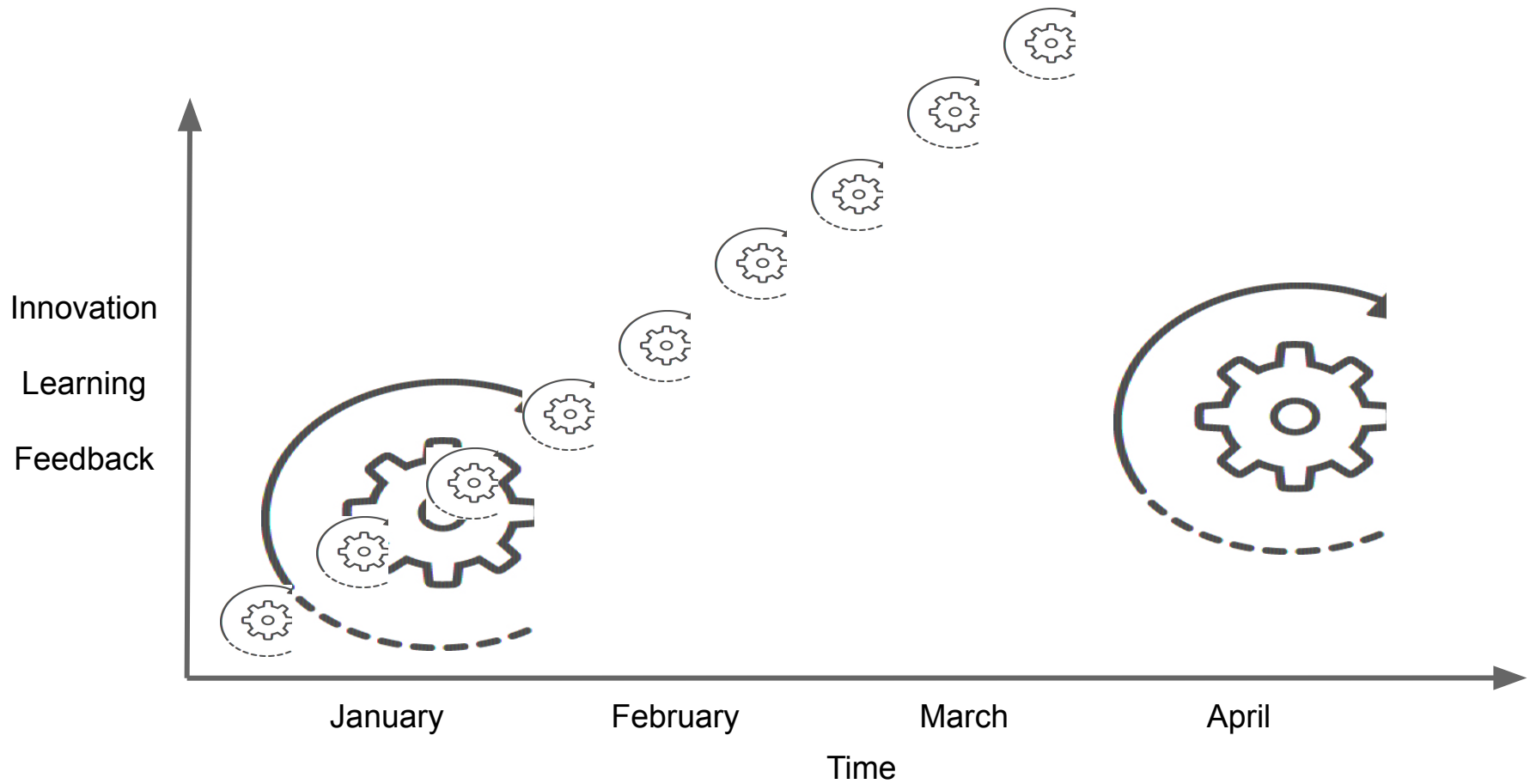


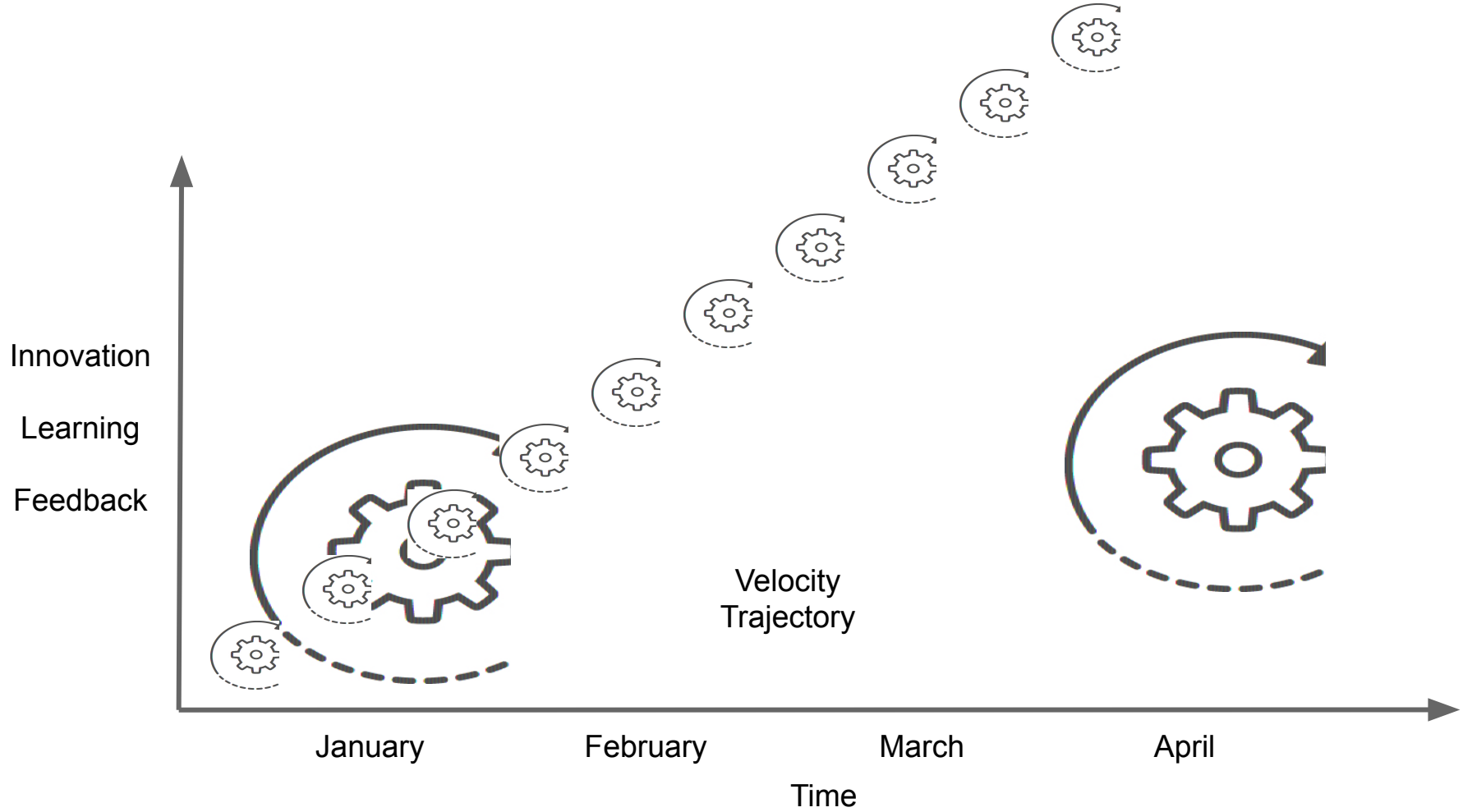












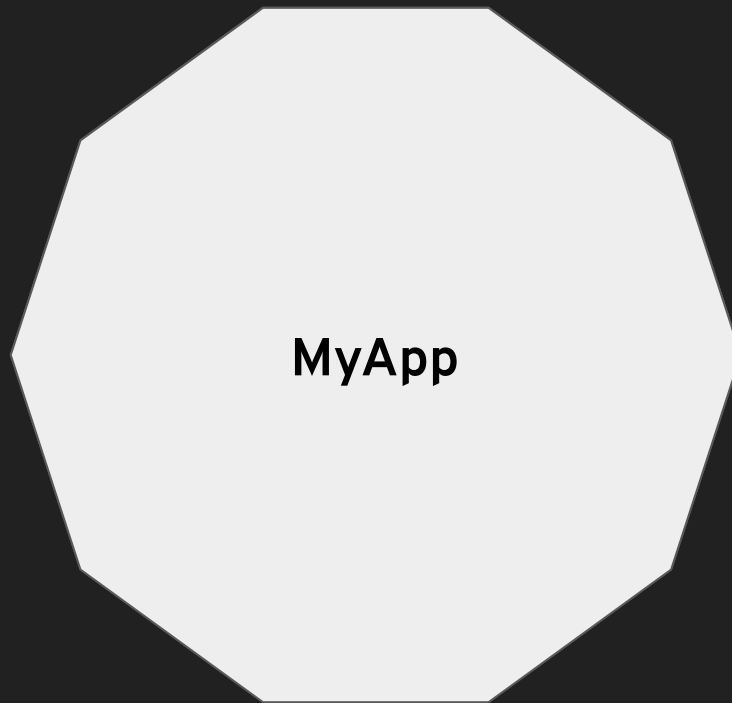


“

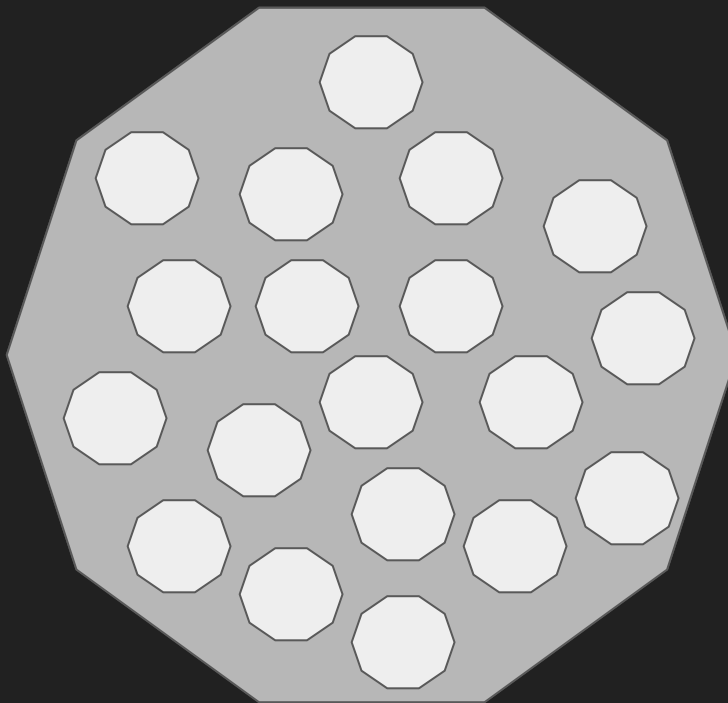
*Those who are unaware
they are walking in
darkness will never
seek the light.*

— Bruce Lee

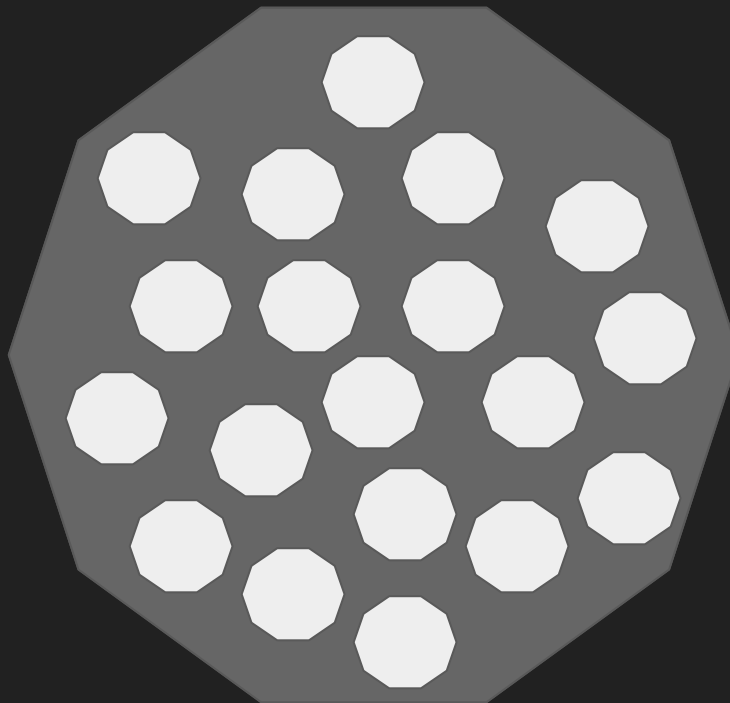
Monolith



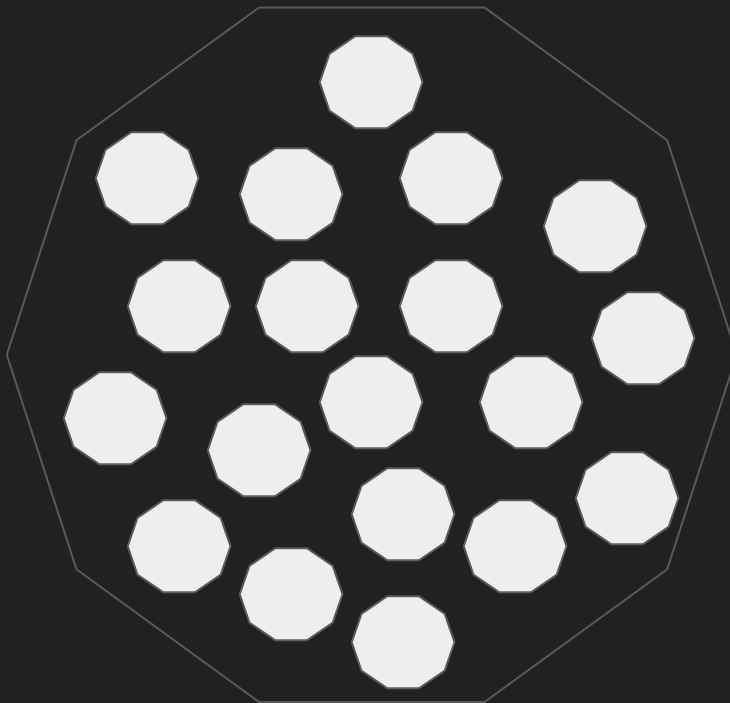
The Application



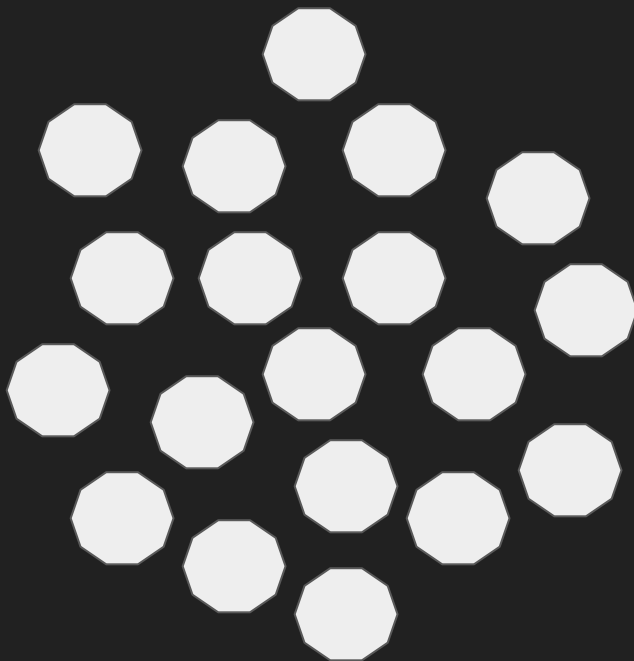
Modules



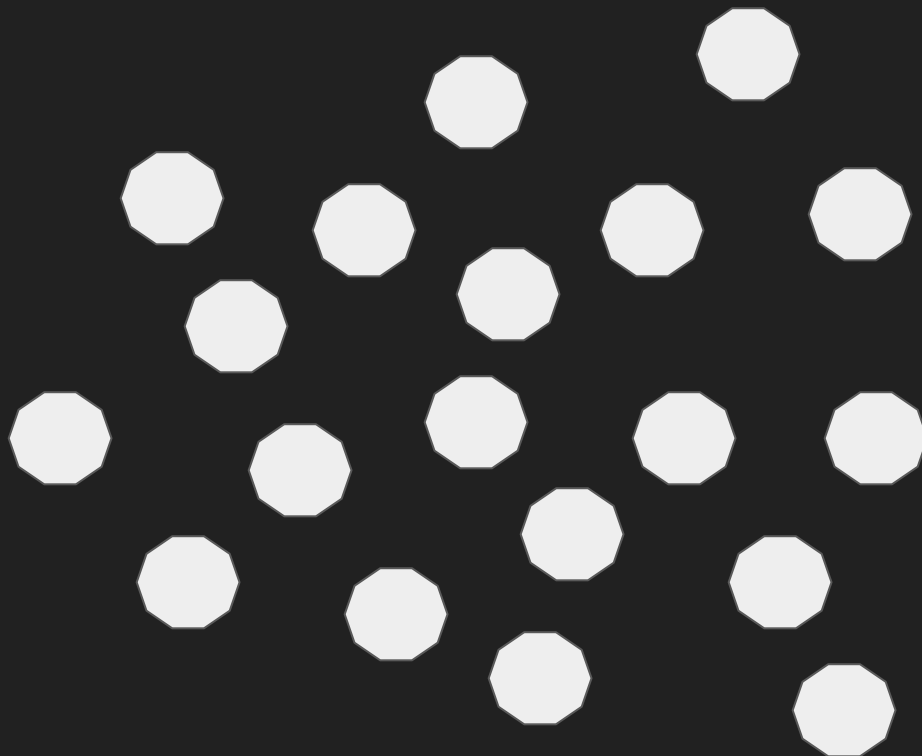
Microservices



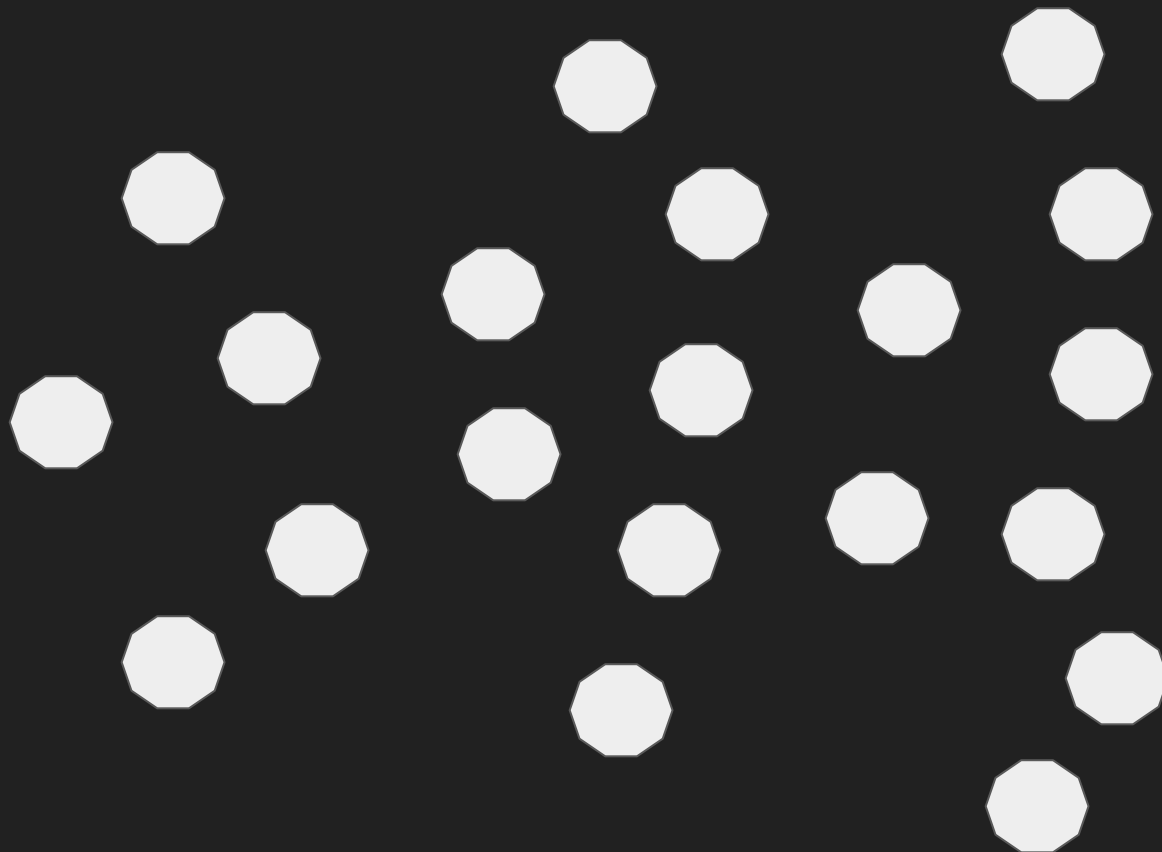
Microservices



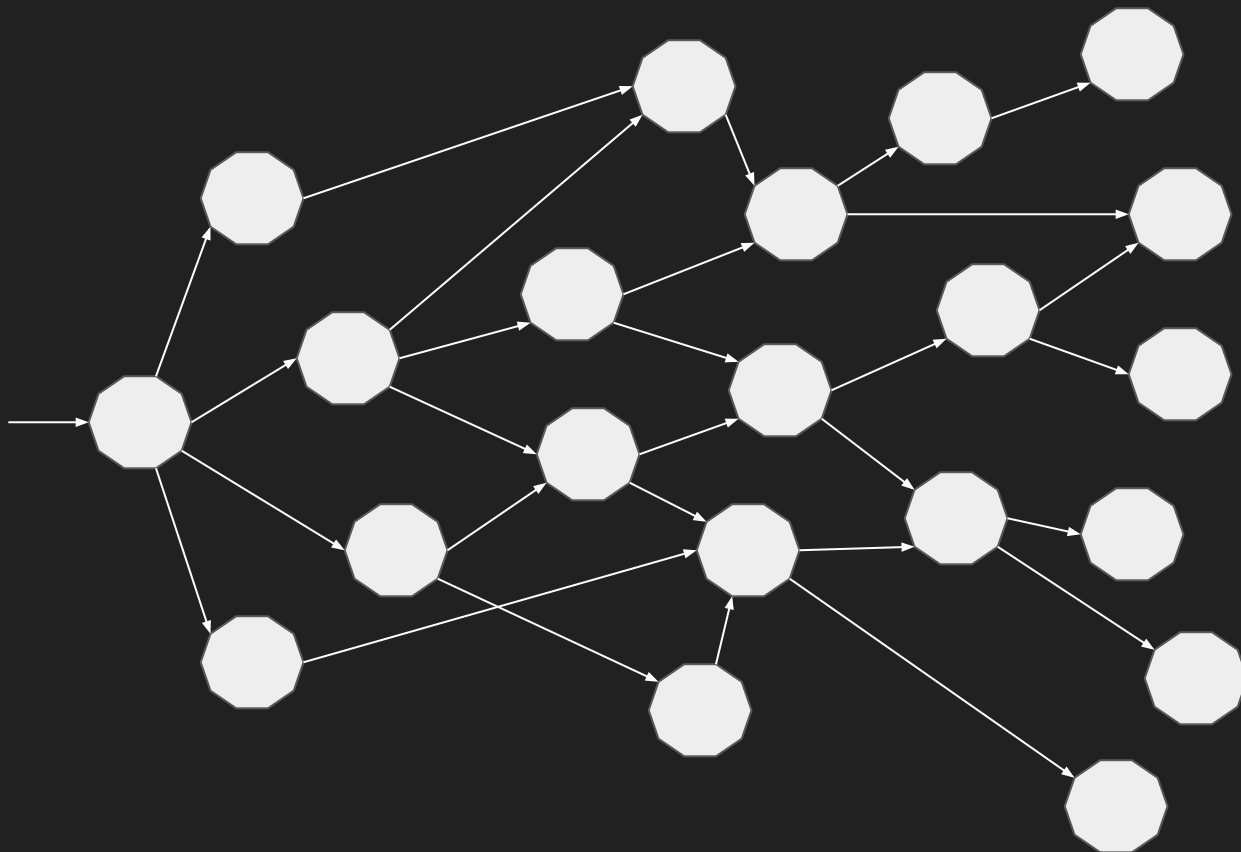
Microservices



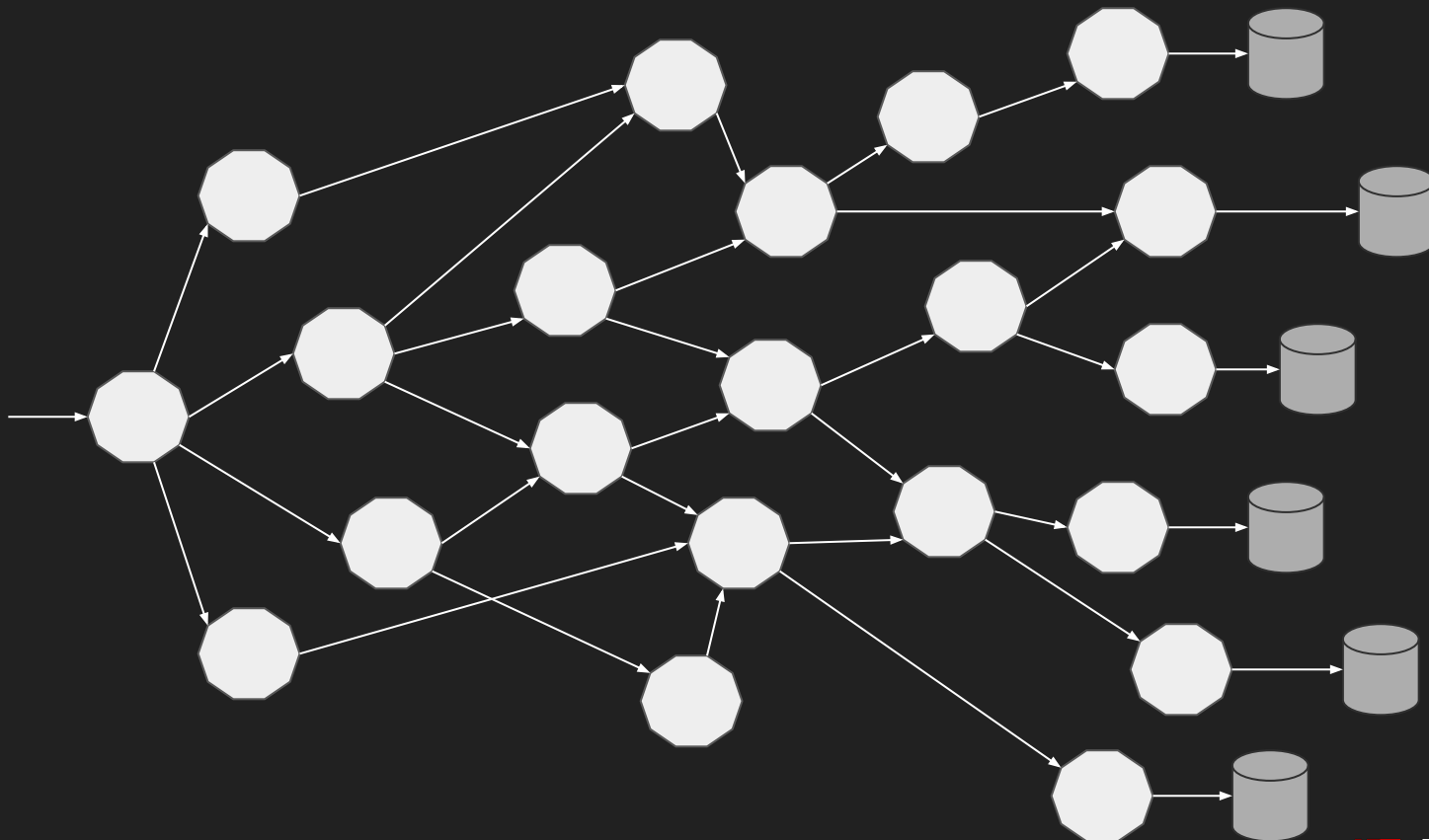
Microservices



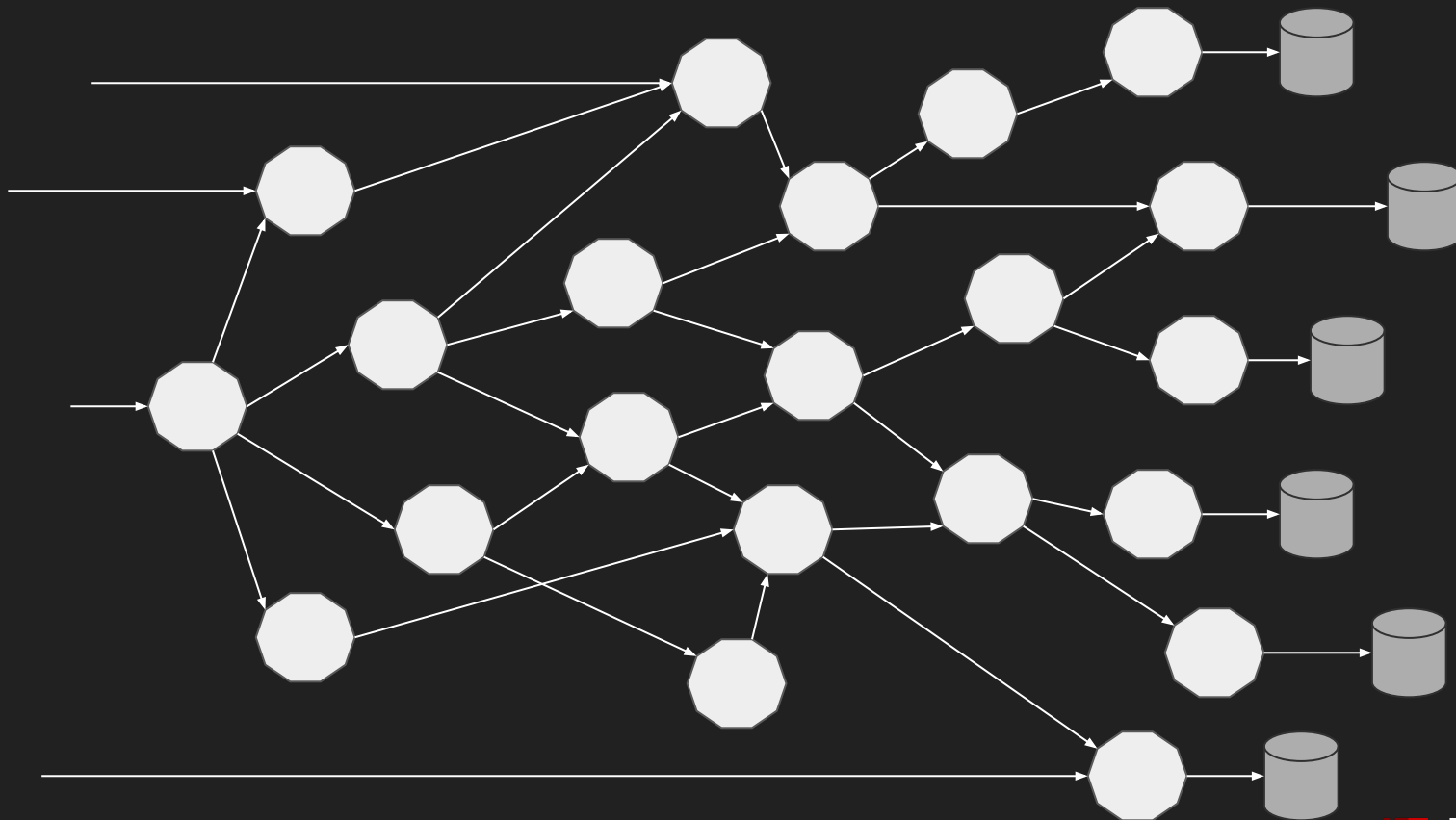
Network of Services



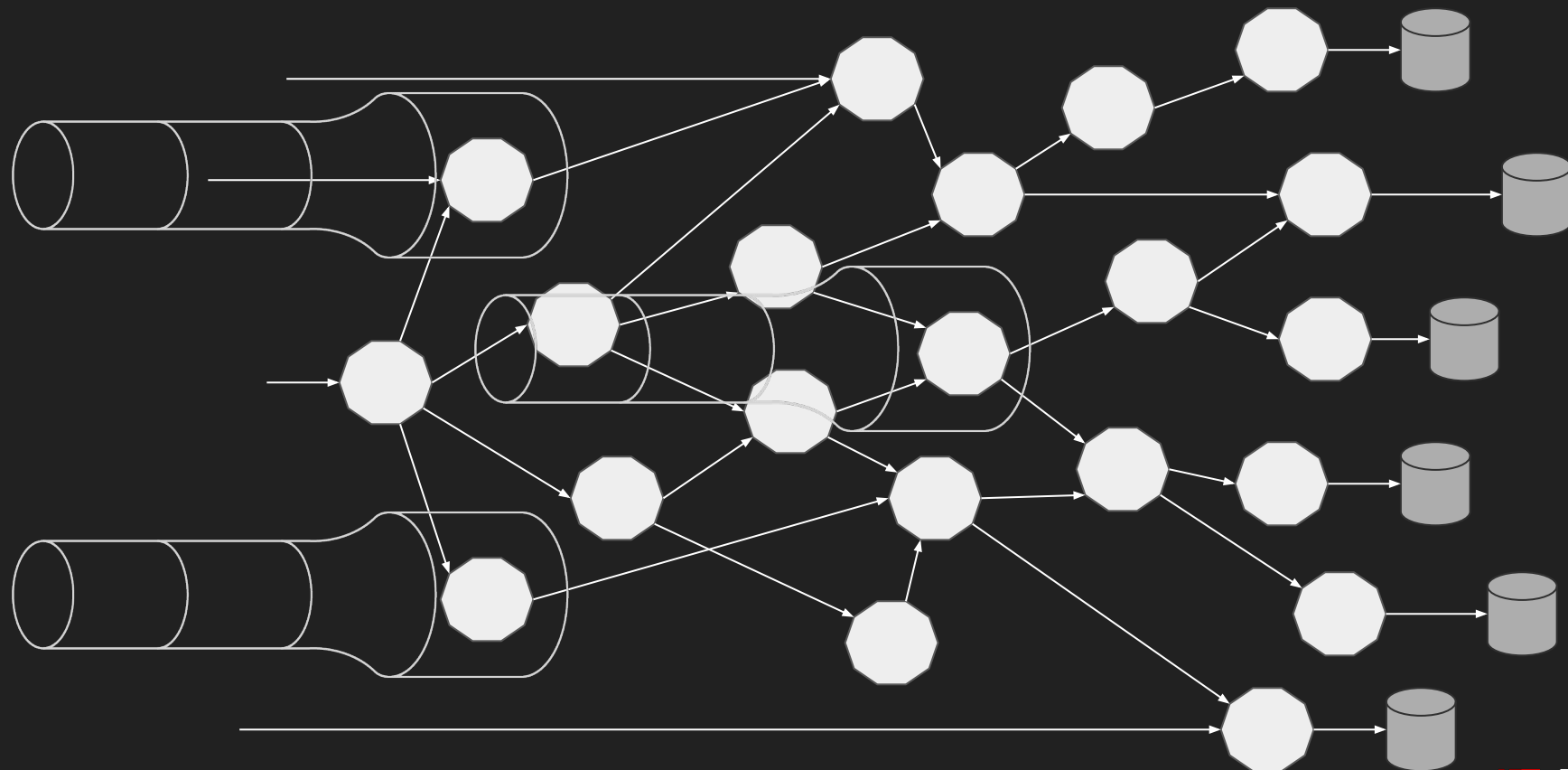
Microservices own their Data



Multiple Points of Entry

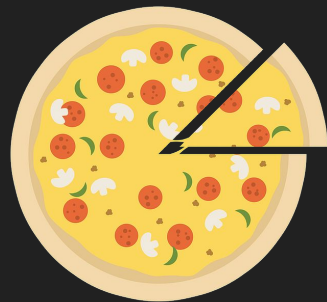


Multiple Teams, Multiple Pipelines



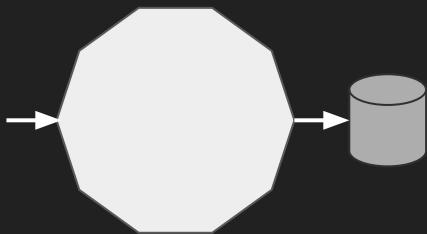
Microservices Principles

1. Deployment **Independence** - updates to an individual microservice have no negative impact to any other component of the system. Optimized for **Replacement**
2. Organized around **business** capabilities
3. **Products** not Projects
4. **API** Focused
5. **Smart** endpoints and dumb pipes
6. Decentralized Governance
7. Decentralized Data Management
8. Infrastructure Automation (infrastructure as code)
9. Design for failure
10. Evolutionary Design



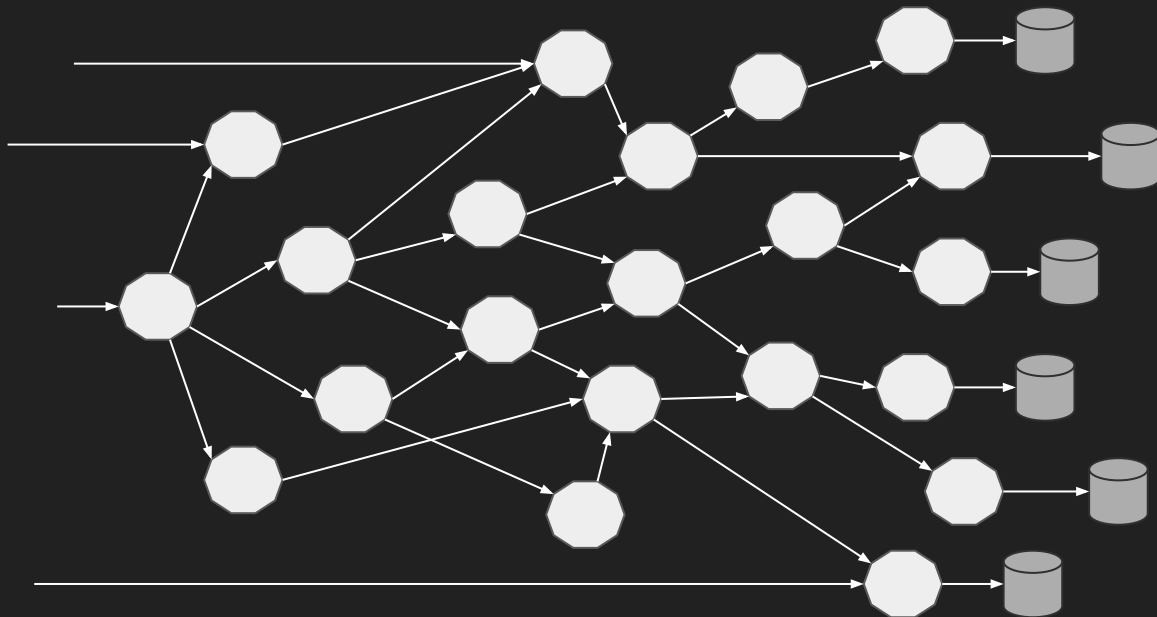
2 Pizza Team

Old School

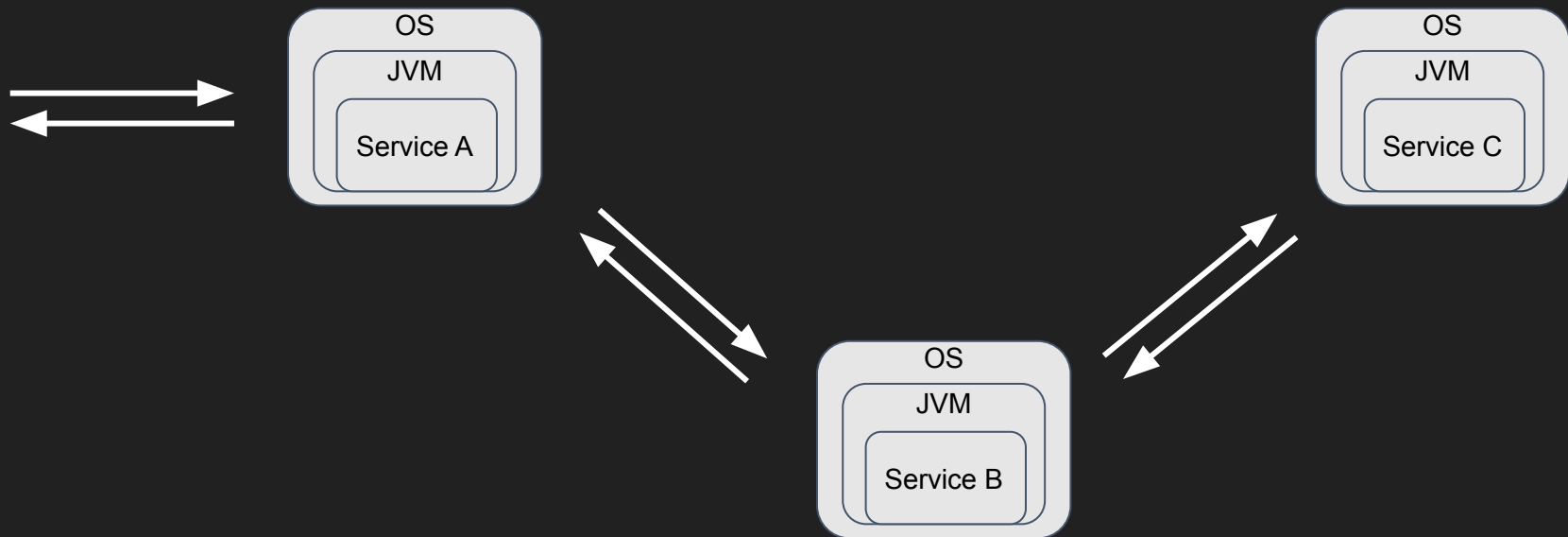


Love Thy Mono

New School



Microservices == Distributed Computing

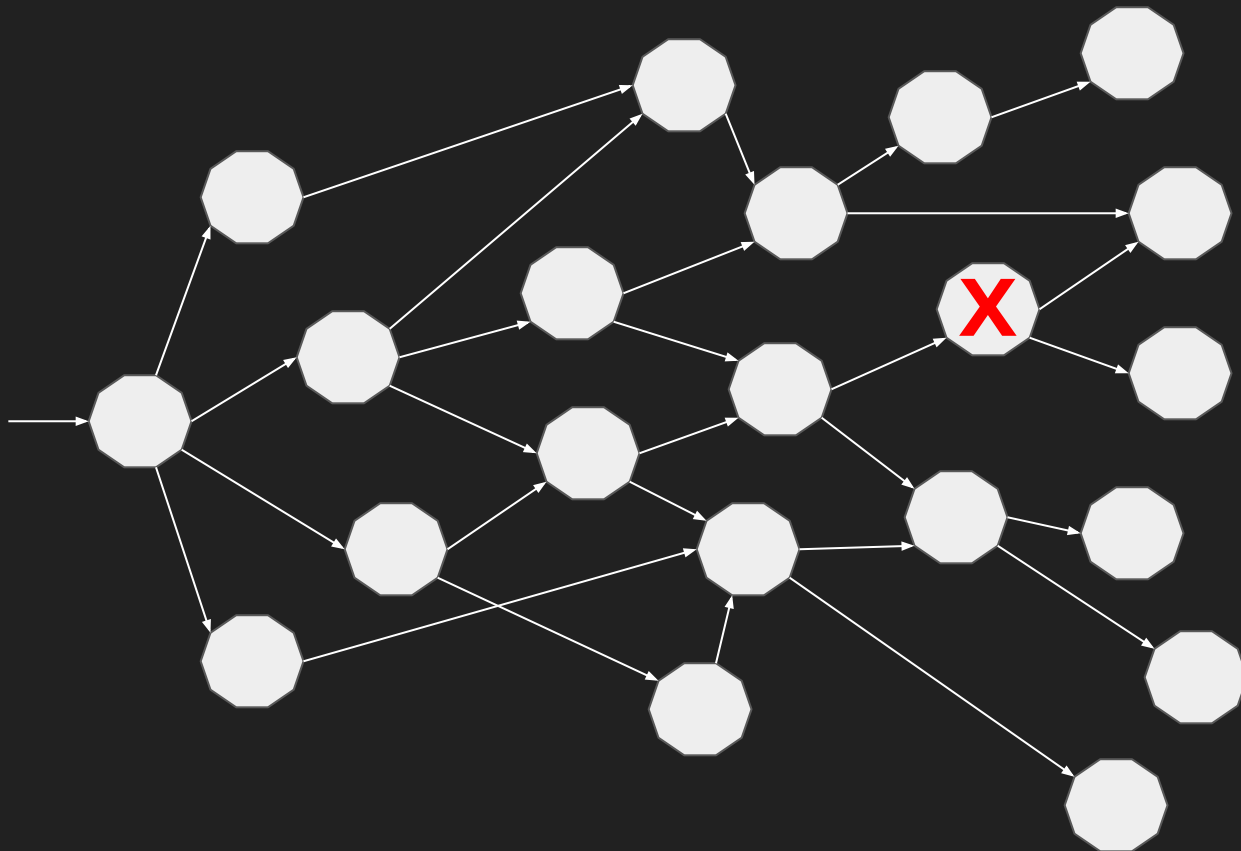


Fallacies of Distributed Computing

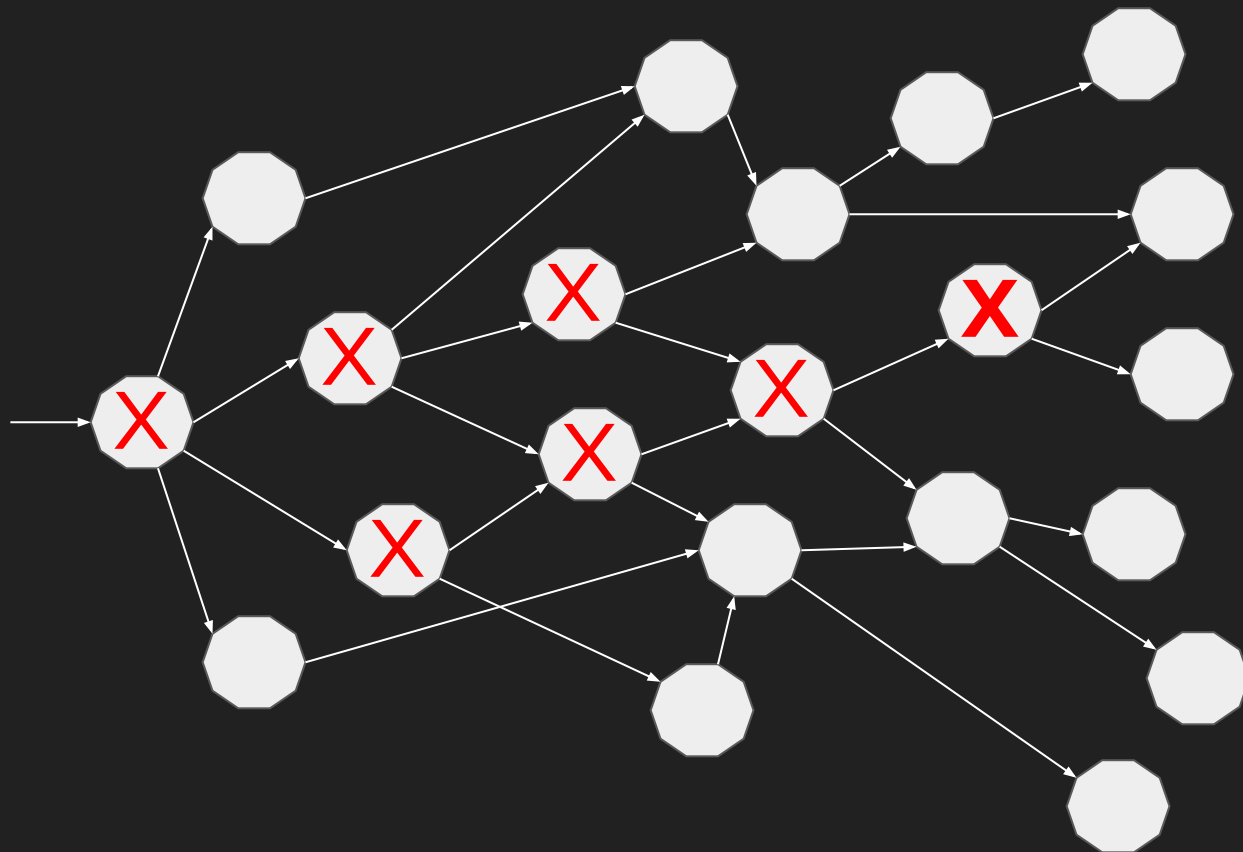
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

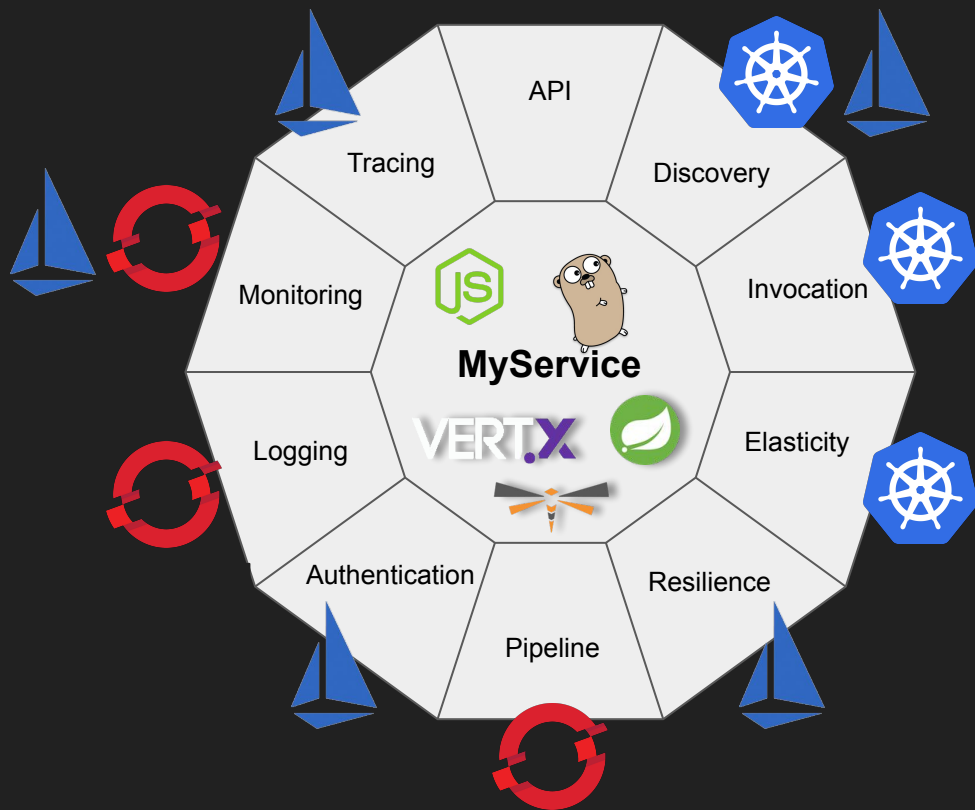
Failure of a Service



Cascading Failure



Microservices'ilities + Istio

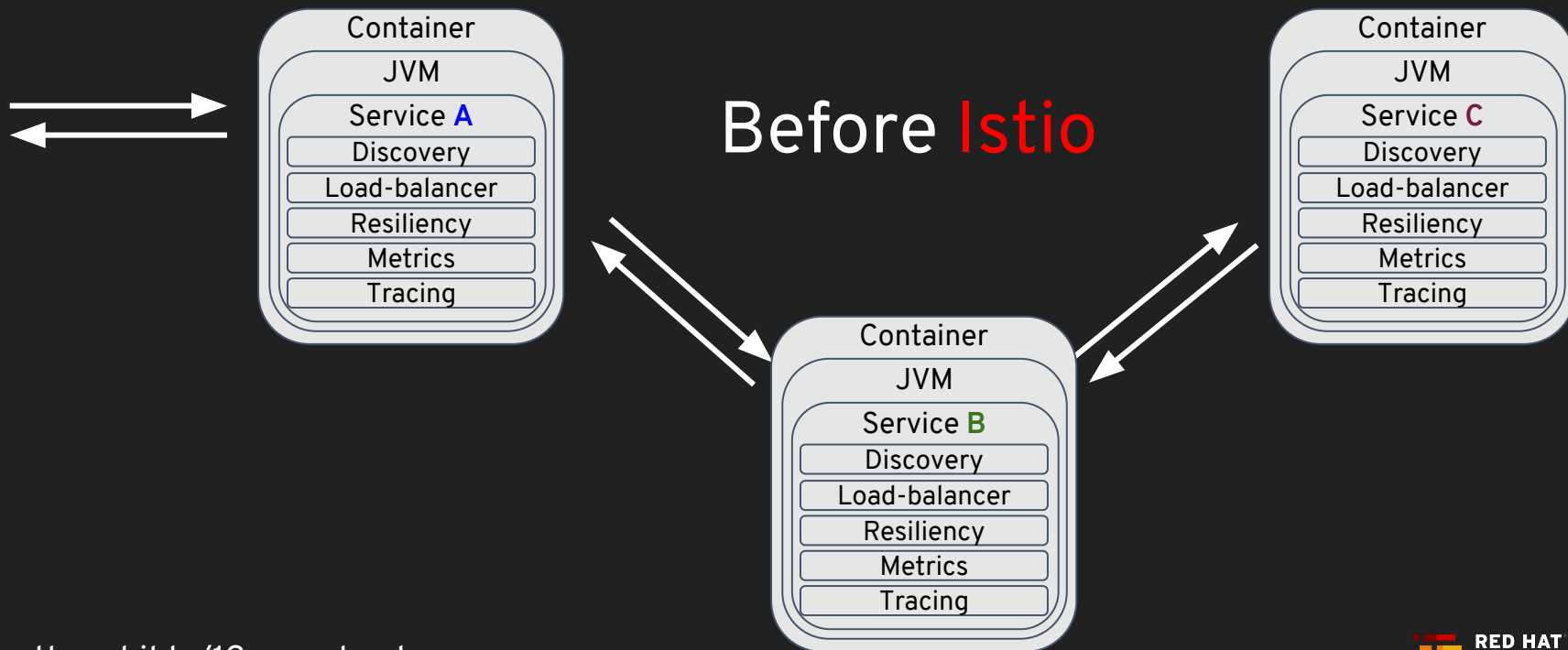


Istio

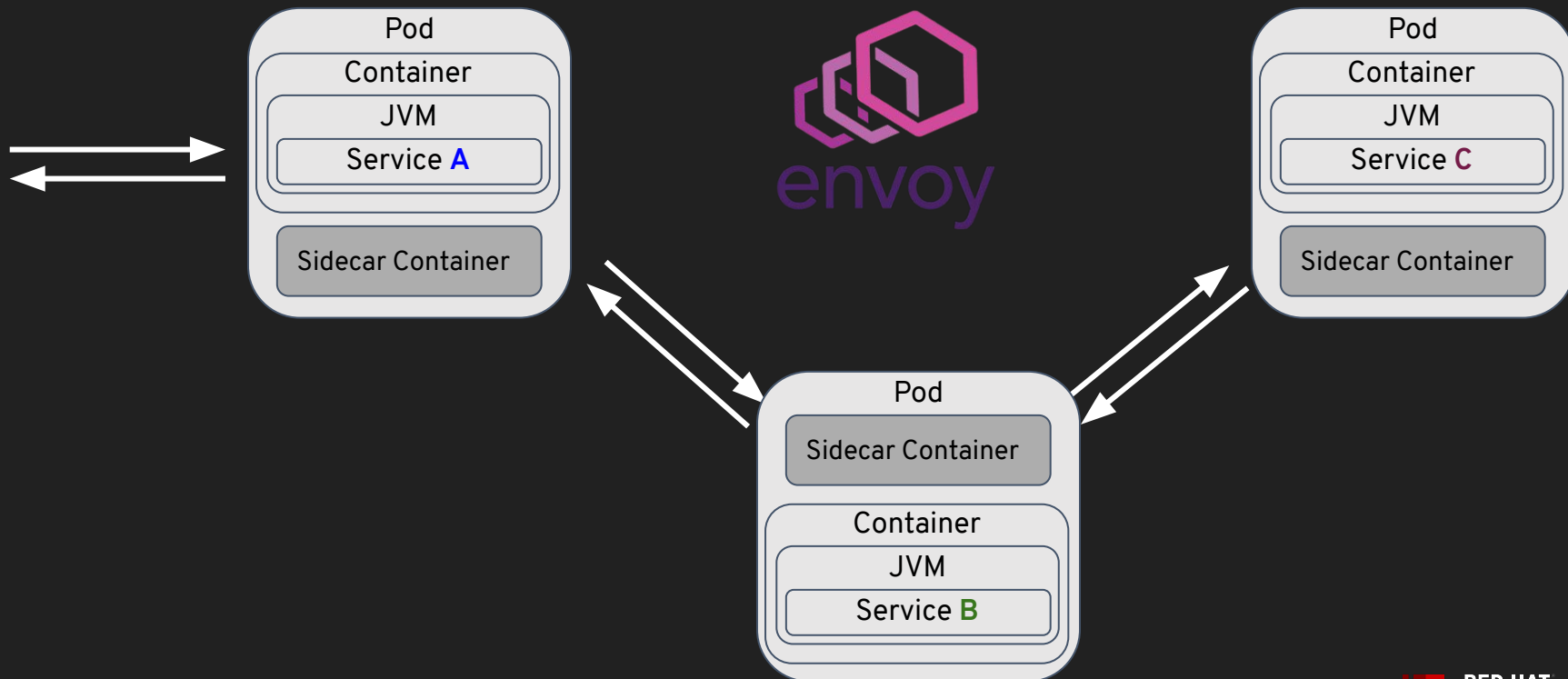


@burrsutter - bit.ly/12wayscloud

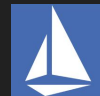
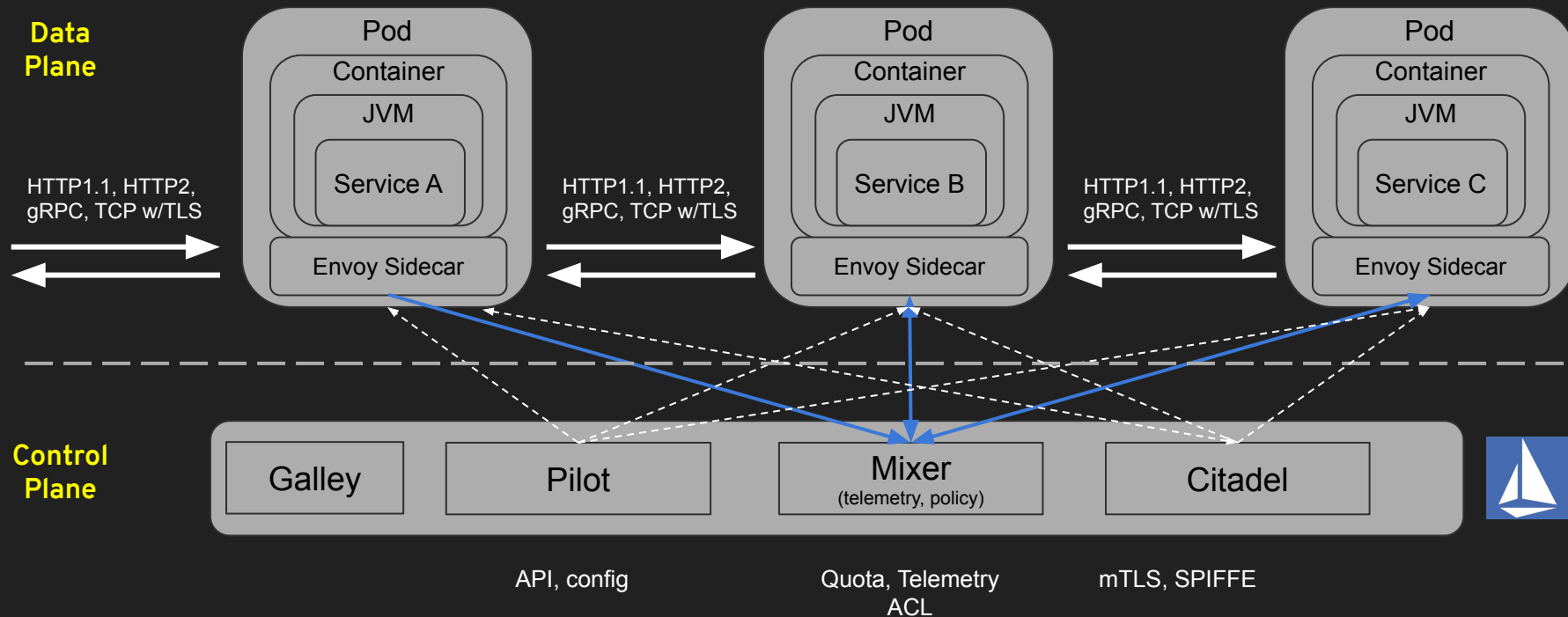
Microservices embedding Capabilities



Envoy is the current sidecar



Istio Data Plane vs Control Plane



Kiali Console

Not Secure | <https://kiali-istio-system.apps.at.burr.run/console/graph/namespaces/?edges=h...>

Kiali

Overview

Graph

Applications

Workloads

Services

Istio Config

Distributed Tracing

Graph

Namespace: tutorial

Display

Edge Labels

Graph Type

Versioned app

Fetching: Last min

Every 15 sec

Namespace: tutorial

- 4 apps
- 3 services
- 9 links

HTTP Traffic (requests per second):

Total	%Success	%Error
3.20	100.00	0.00

0 25 50 75 100 %

OK 3xx 4xx 5xx

HTTP - Total Request Traffic min / max:
RPS: 1.60 / 1.60, %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
ⓘ Not enough traffic to generate chart.

unknown

customer

preference

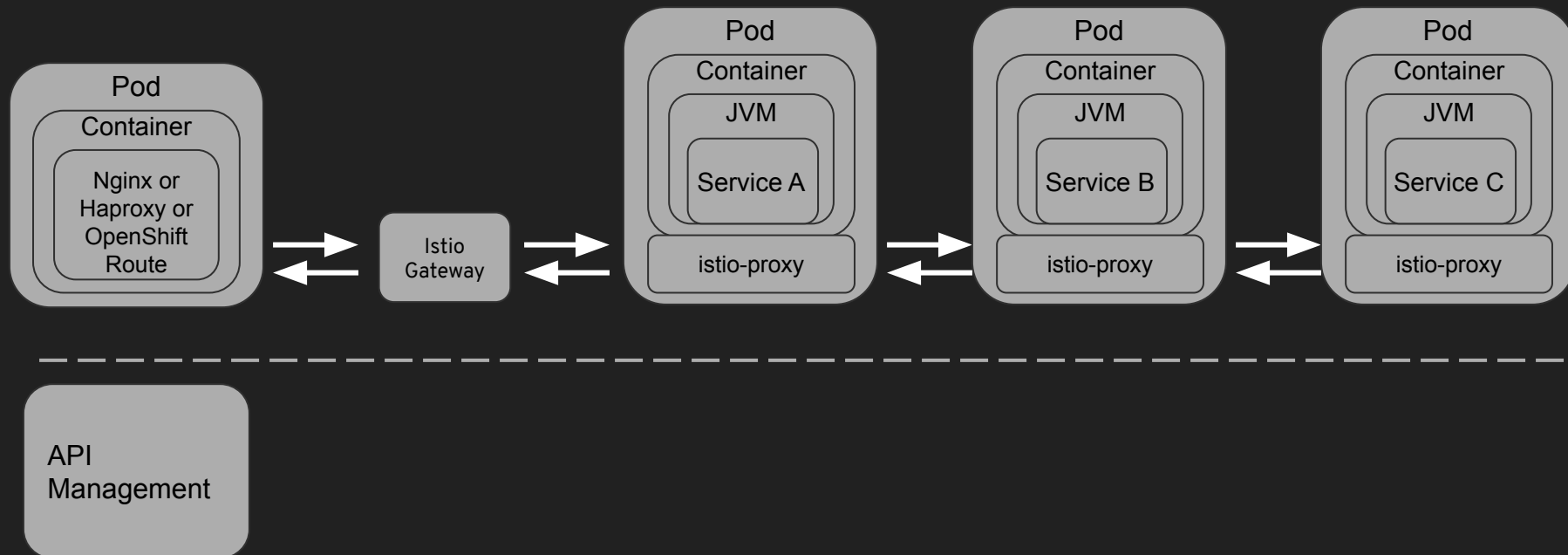
recommendation

v1

v2

Legend

API Gateways





“

*Showing off is the
fool's idea of glory.*

— Bruce Lee

@burrsutter



RED HAT
DEVELOPER

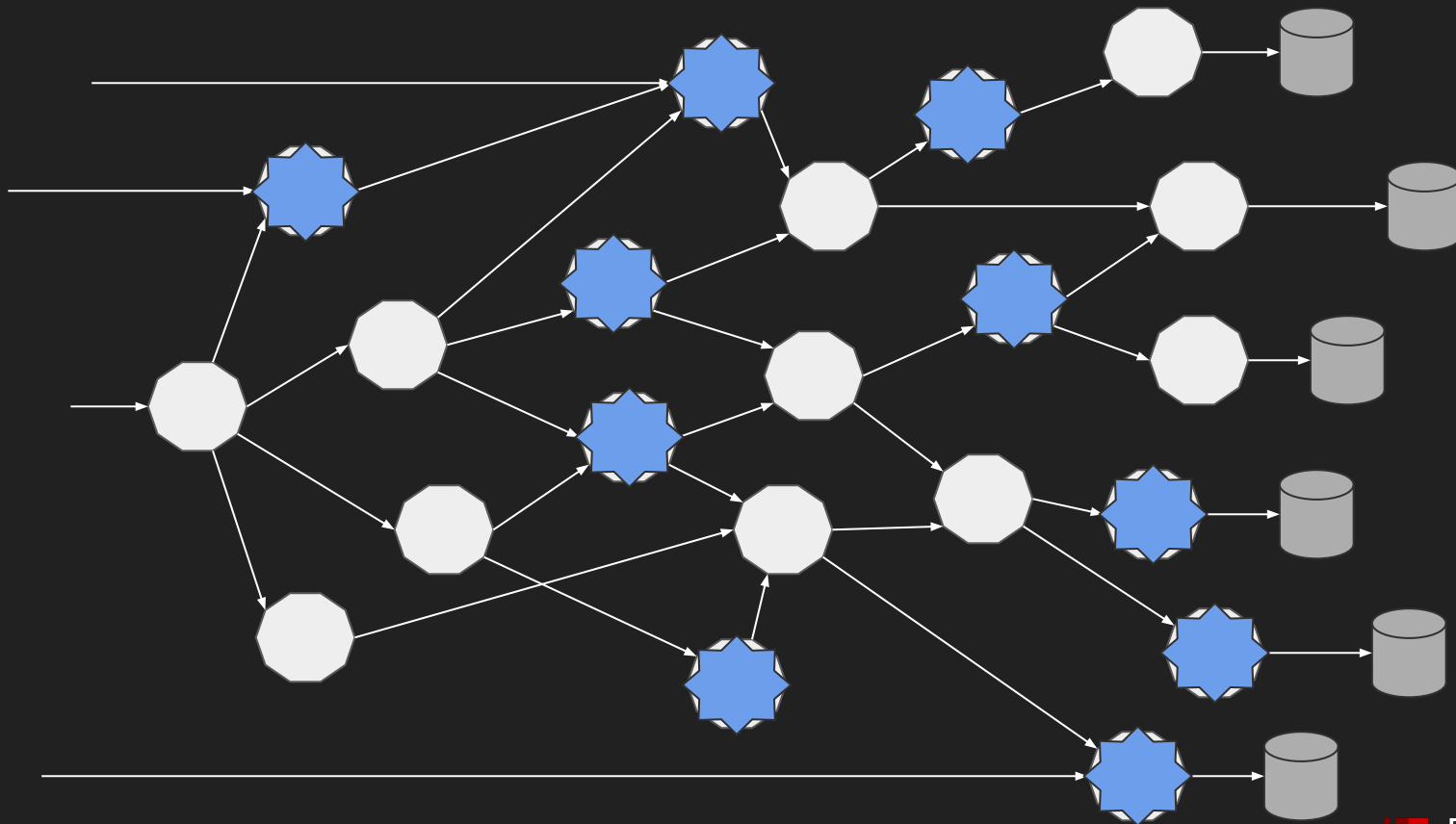




“ *Be happy, but
never satisfied.*

— *Bruce Lee*

Let there be Functions?



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

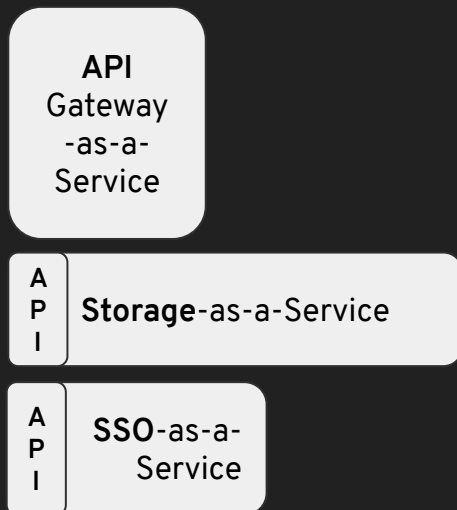
API
Gateway
-as-a-
Service

Authentication Service

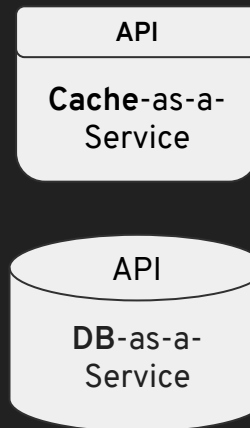
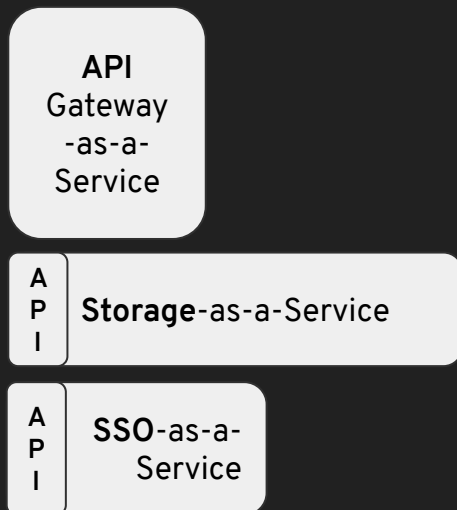
API
Gateway
-as-a-
Service

**A
P
I** **SSO-as-a-
Service**

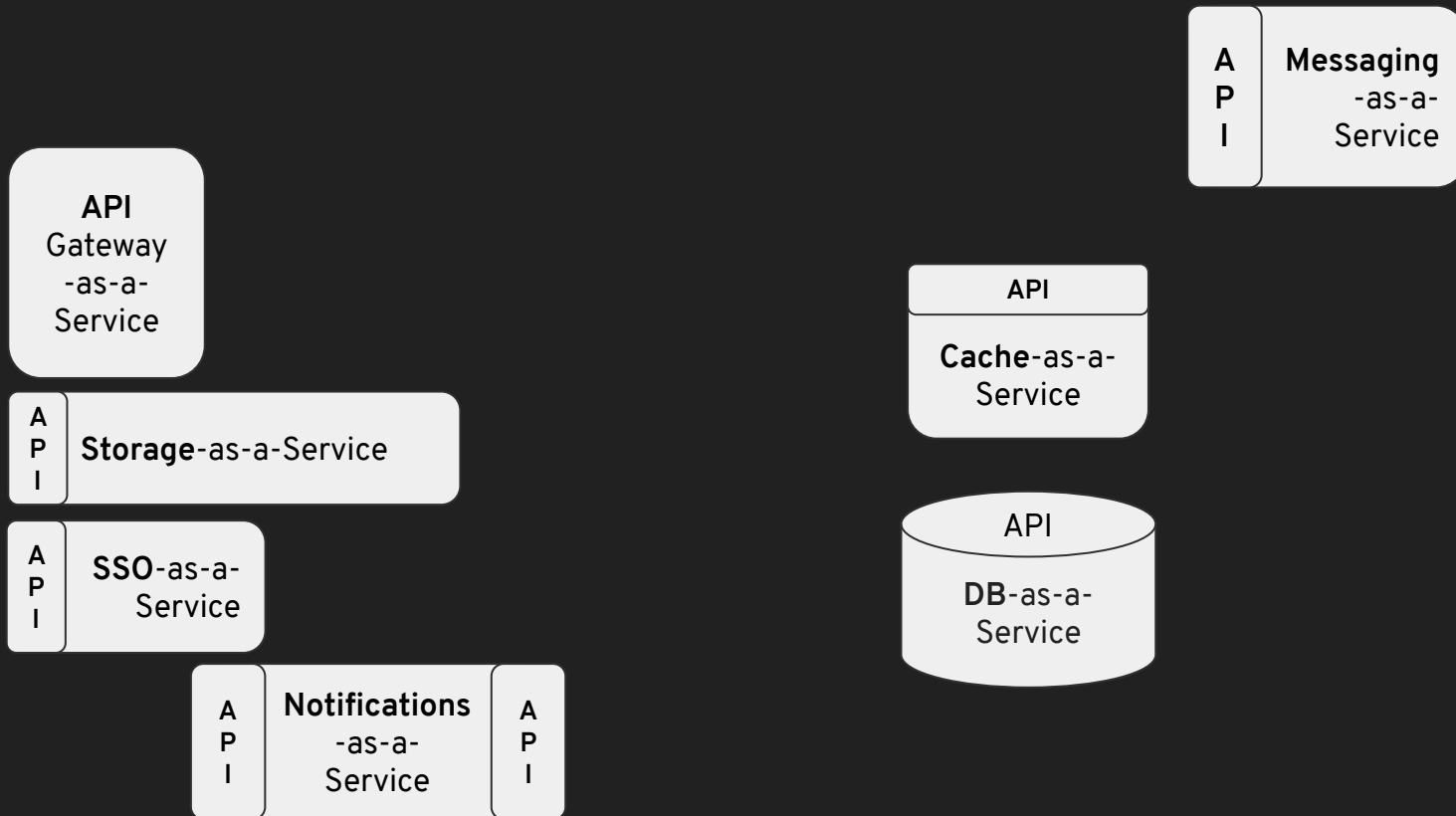
File Storage Service



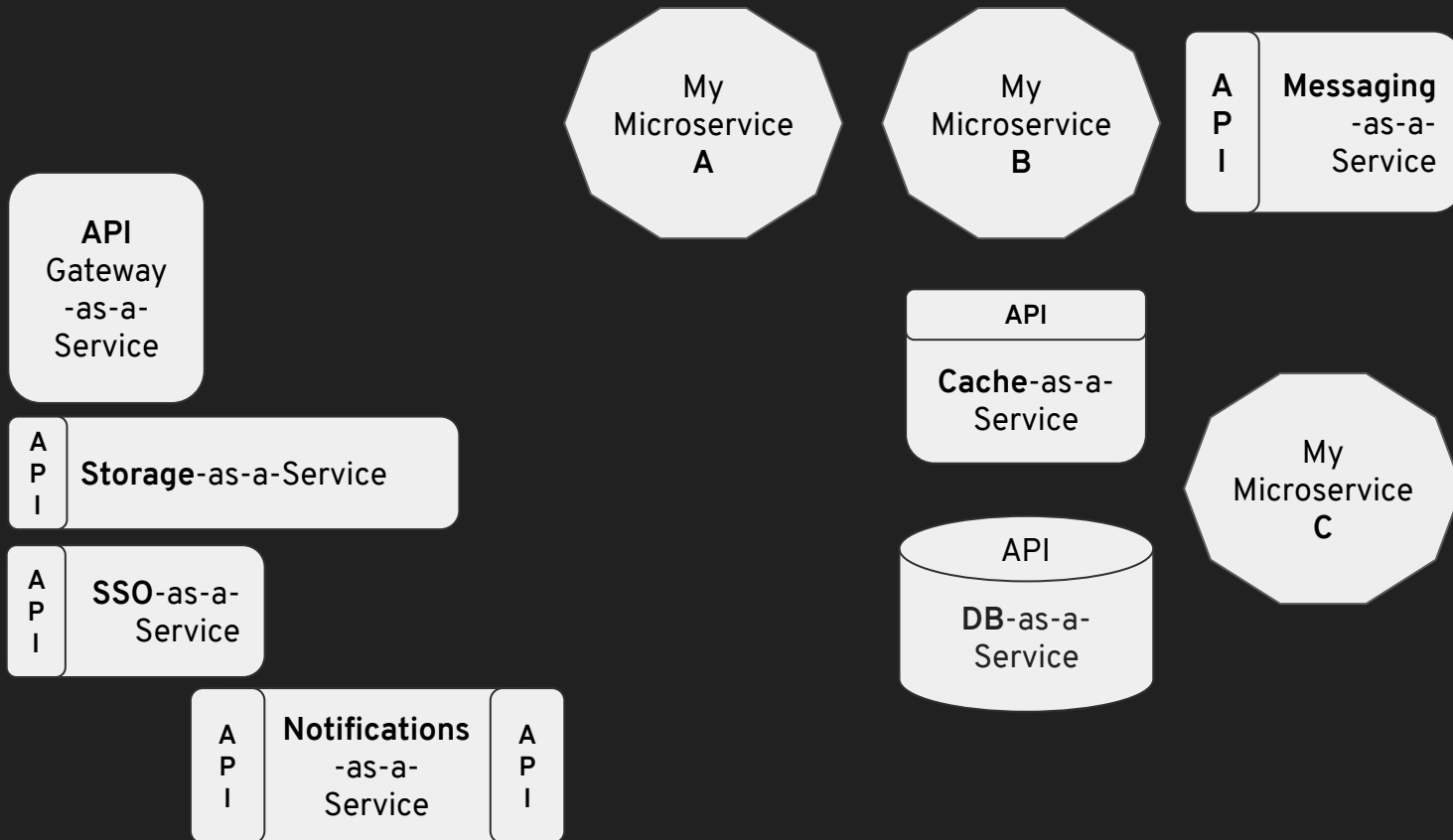
Data Services



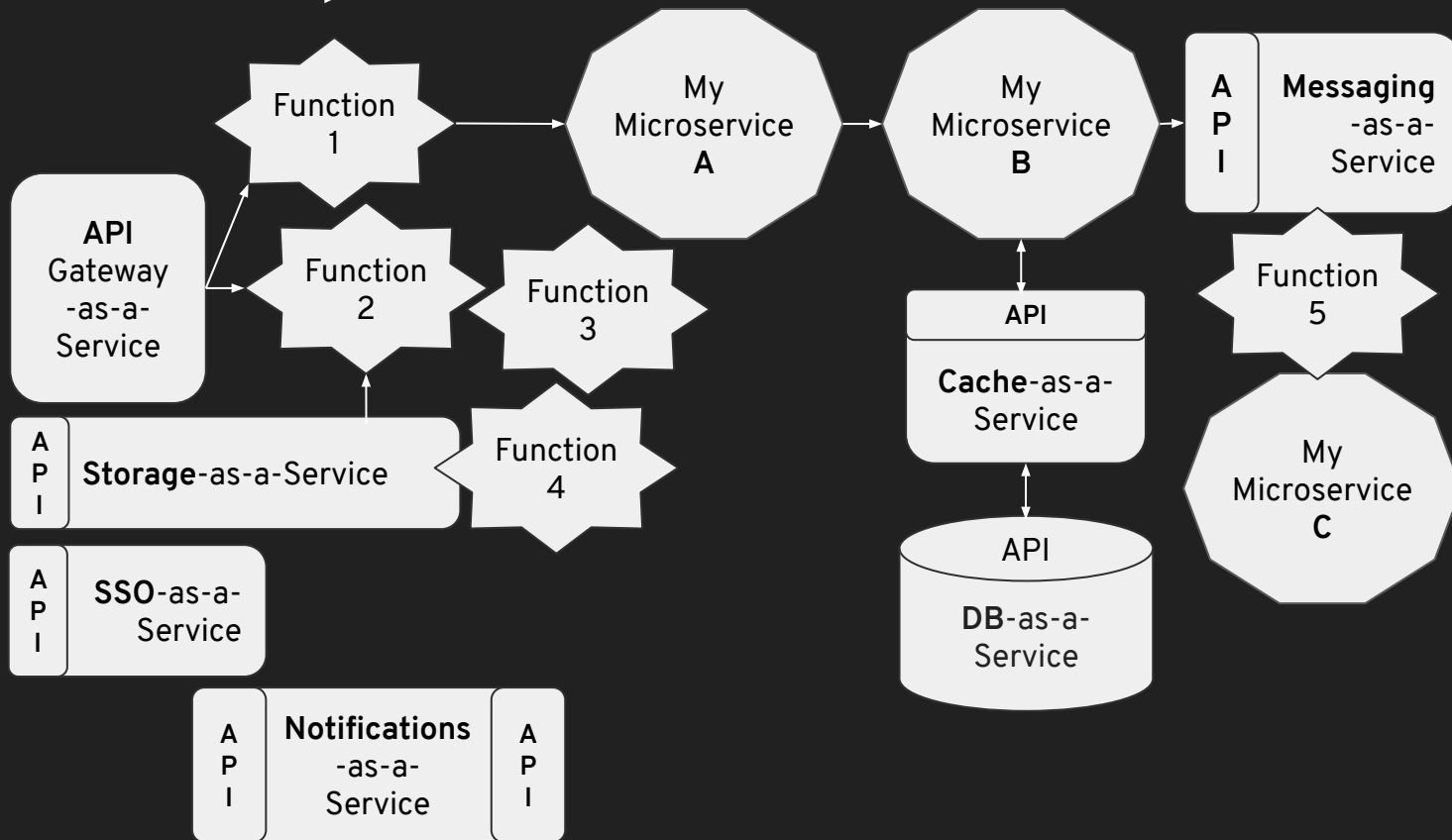
Connectivity Services



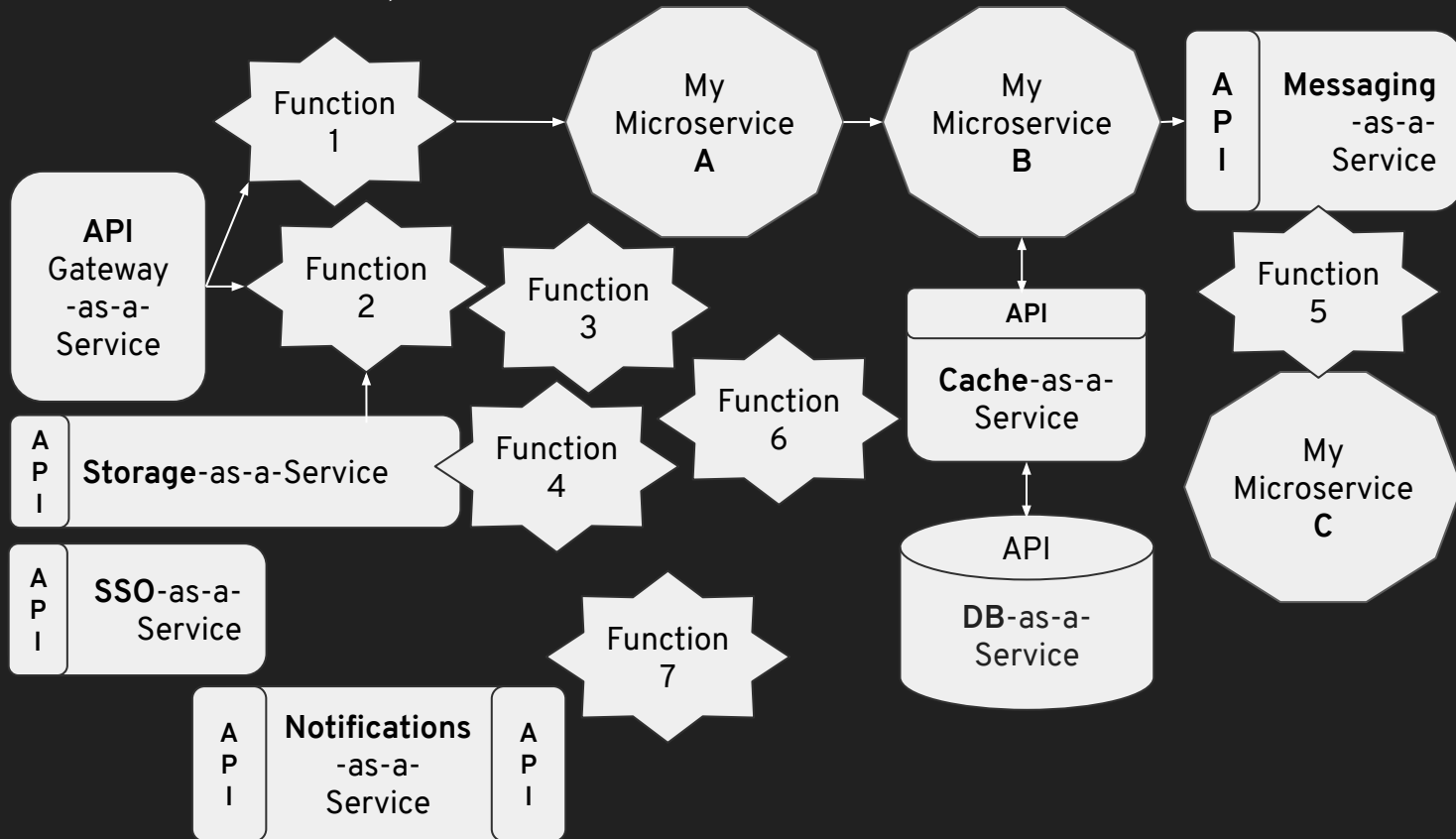
Your Containerized Services



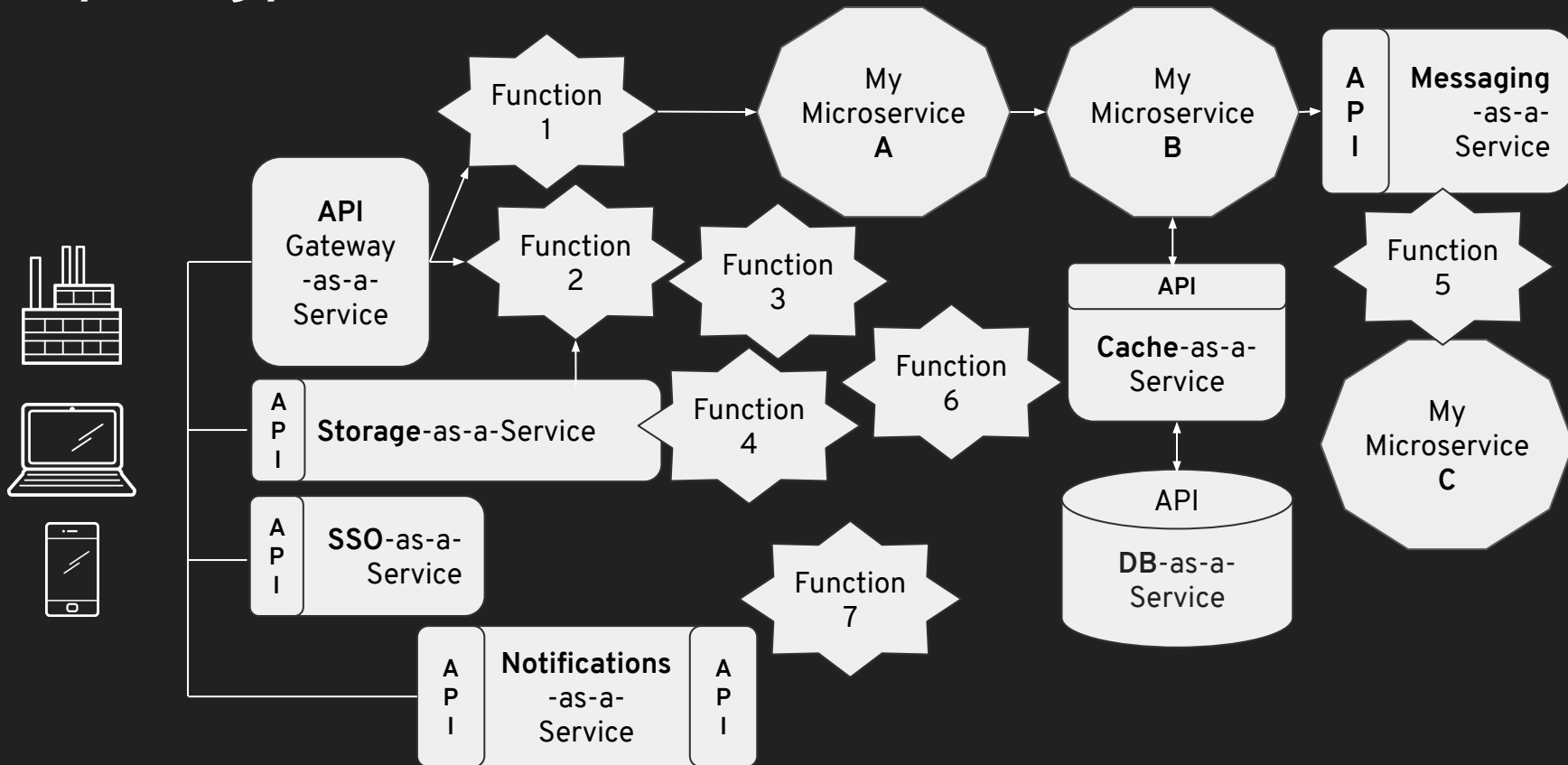
Event-Driven Input



Event-Driven Output

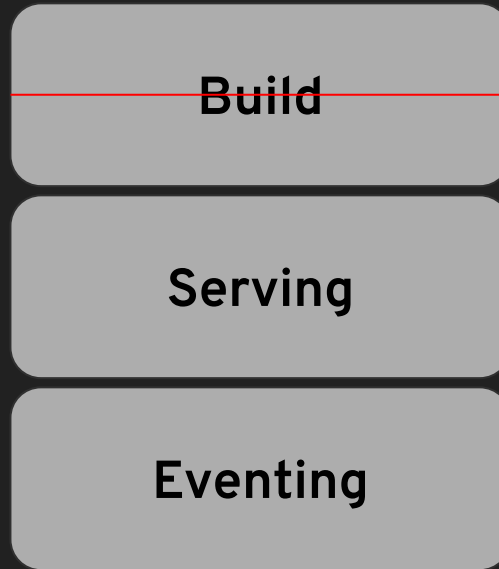


Synergy



Knative

Knative's Primary Components



FaaS Kubernetes Players



FaaS Kubernetes Players



APACHE
OpenWhisk™



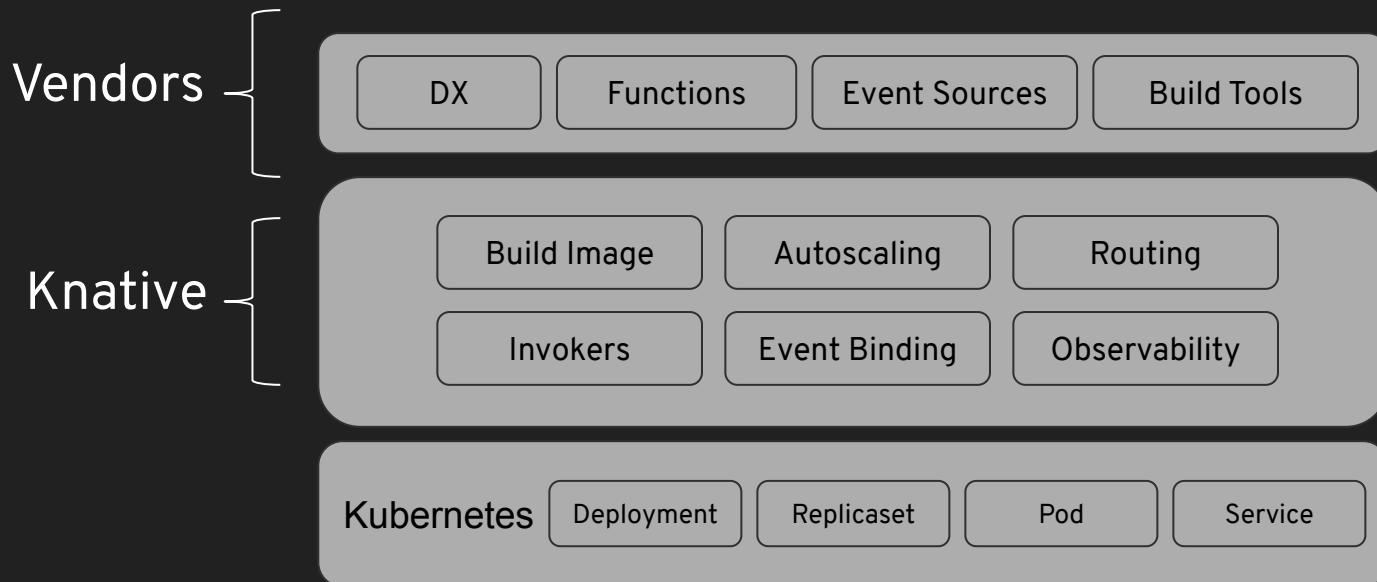
Why?



OPENFAAS

nuclio

Primitives



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 (Beyond HTTP)

Knative Serving

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

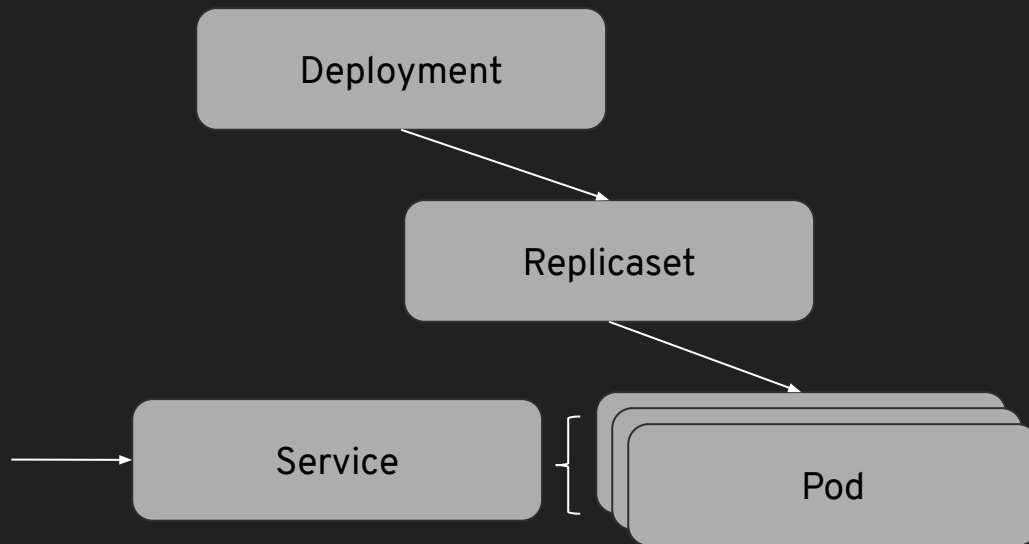
```
kubectl -n knative-serving edit  
configmap config-autoscaler
```

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

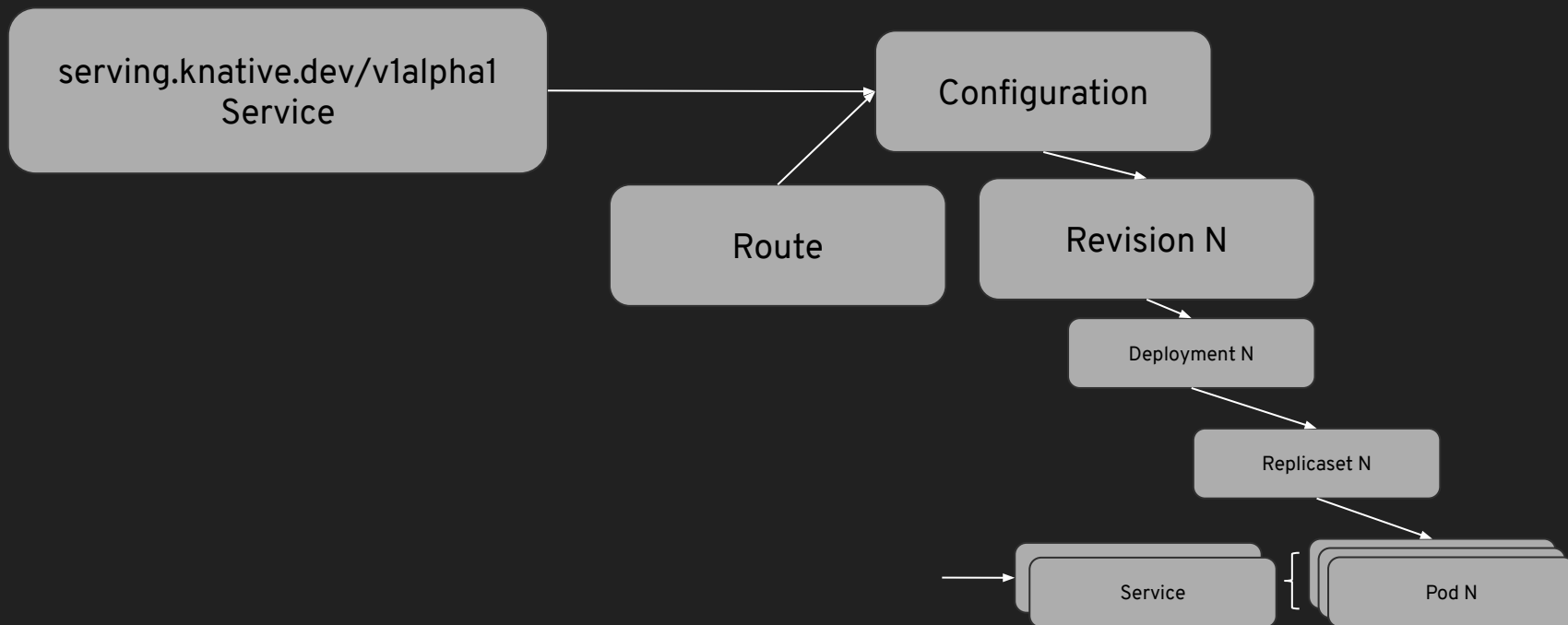
```
scale-to-zero-grace-period: 30s
```

```
stable-window: 30s
```

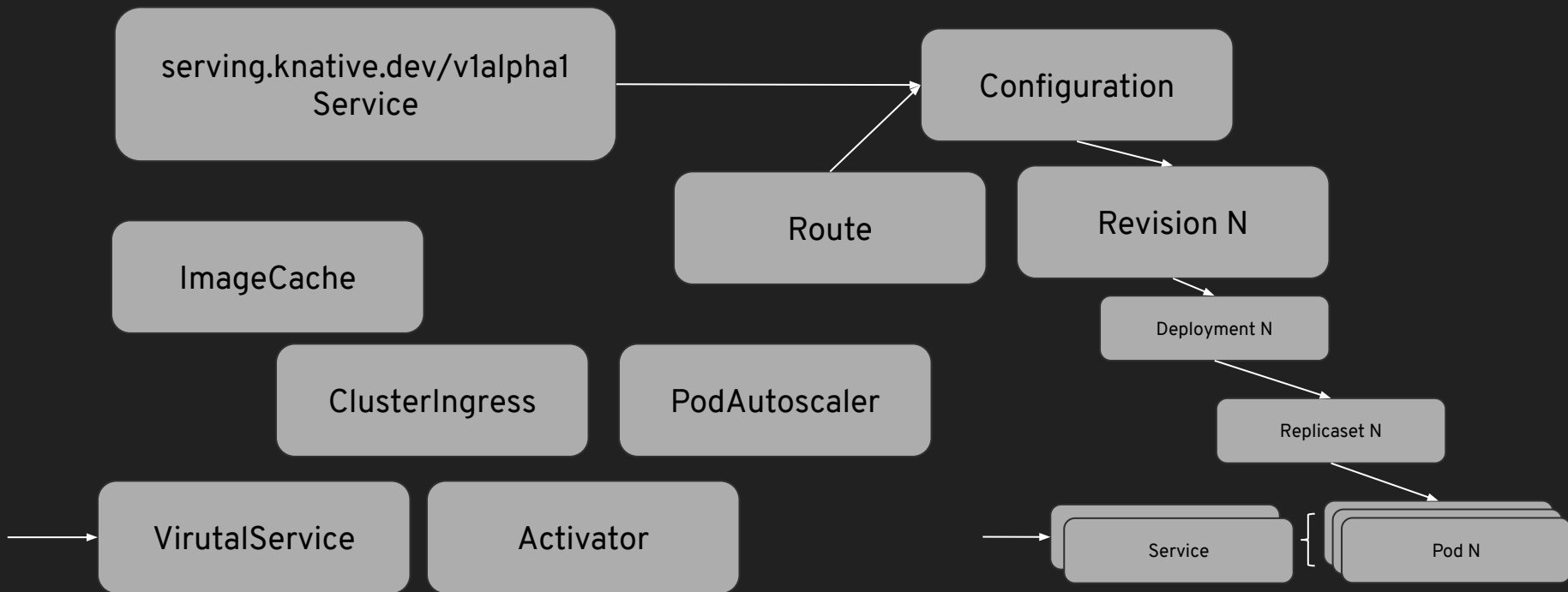
kubectl apply -f Deployment.yaml



kubectl apply -f ksvc.yaml



kubectl apply -f ksvc.yaml



Knative Build

Knative Build

```
kubectl get crd | grep build
```

```
builds.build.knative.dev
```

```
buildtemplates.build.knative.dev
```

```
clusterbuildtemplates.build.knative.dev
```

Knative Build Resources

1. Secret

User ID/Password for remote registry

2. ServiceAccount

Associated with the Secret

3. PVC: Maven

Caches Maven downloads

4. PVC: Image Cache

Caches Linux image downloads

5. Build

The Steps

6. Service

Knative Serving Service



TEKTON

Knative Eventing

Knative Eventing Sources

Github

Websockets

AWS SQS

GCP PubSub

Apache Kafka

Apache Camel

<https://github.com/knative/eventing-sources/tree/master/contrib>

<https://github.com/knative/eventing-sources>

Knative Eventing

```
kubectl get crd | grep eventing
```

```
channels.eventing.knative.dev
```

```
subscriptions.eventing.knative.dev
```

```
cronjobsources.sources.eventing.knative.dev
```

```
githubsources.sources.eventing.knative.dev
```

```
kuberneteseventsources.sources.eventing.knative.dev
```

```
containersources.sources.eventing.knative.dev
```

Resources

<https://learning.oreilly.com/live-training/> (search for "burr")

bit.ly/9stepsawesome - Introduction to Kubernetes ([recording](#))

bit.ly/istio-tutorial - Istio: Enter the Service Mesh (bit.ly/istio-intro)

bit.ly/knative-tutorial

bit.ly/quarkus-tutorial

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

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

<https://blog.openshift.com/knative-configurations-routes-and-revisions/>

O'REILLY®



Compliments of
RED HAT
DEVELOPERS

Migrating to Microservice Databases

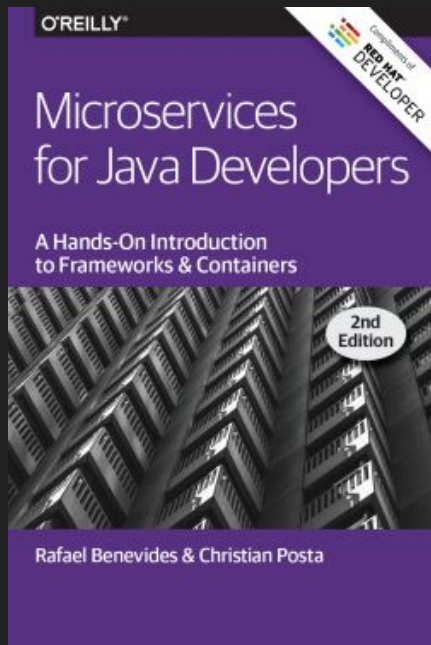
From Relational Monolith
to Distributed Data



Edson Yanaga

bit.ly/mono2microdb

bit.ly/javamicroservicesbook



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](#)

Advanced Materials

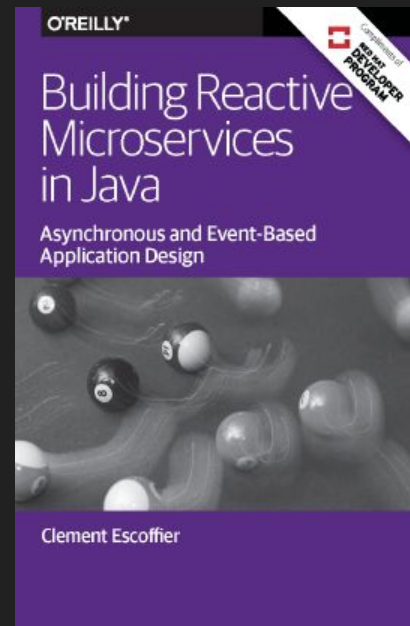
bit.ly/istio-tutorial

learn.openshift.com/servicemesh

bit.ly/istio-intro

bit.ly/serverlesskube

bit.ly/reactivemicroservicesbook



bit.ly/knative-tutorial



*“ Don't fear failure.
Not failure, but low aim,
is the crime.
In great
attempts it is
glorious even to fail*

— Bruce Lee

@burrsutter