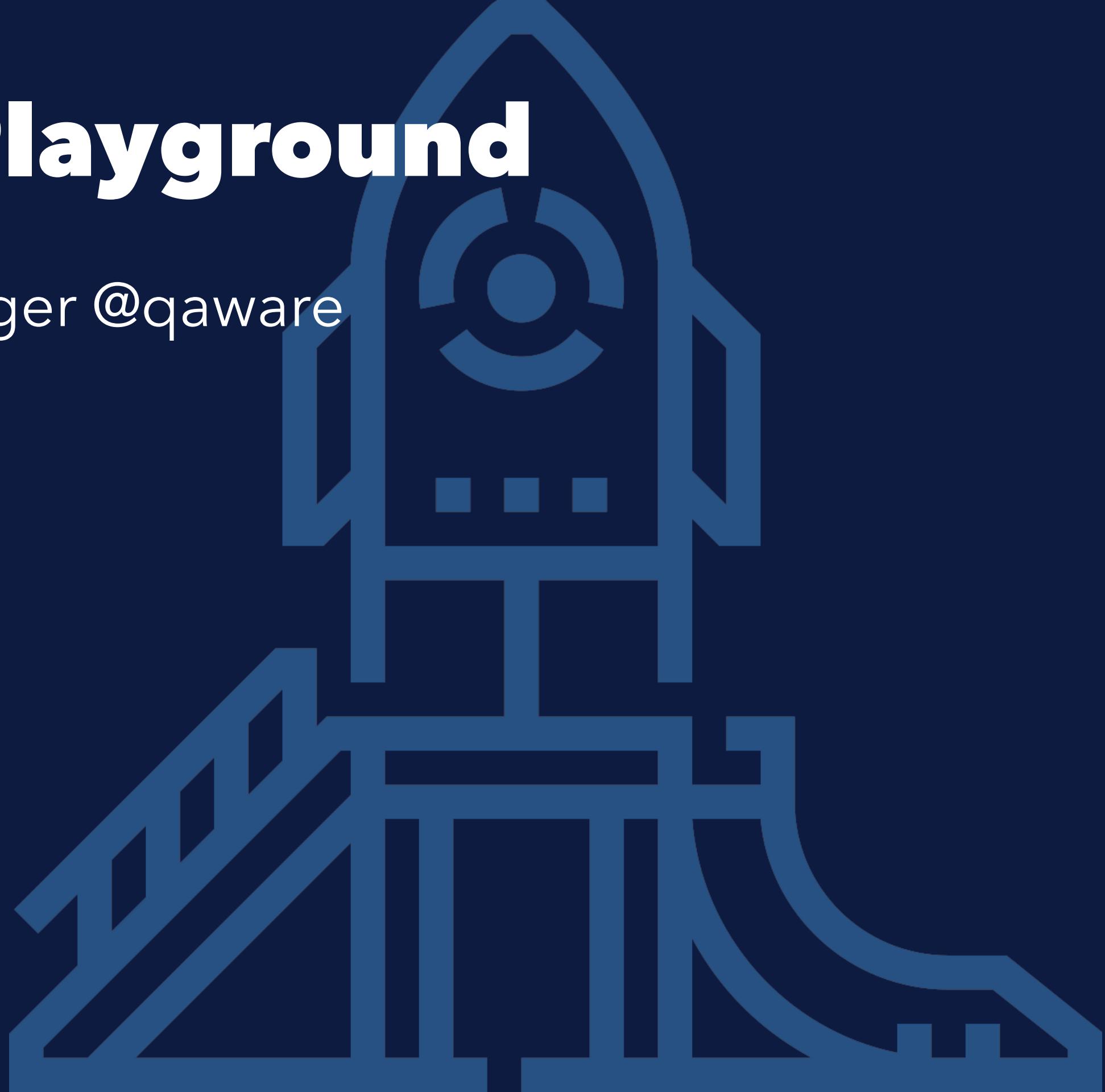


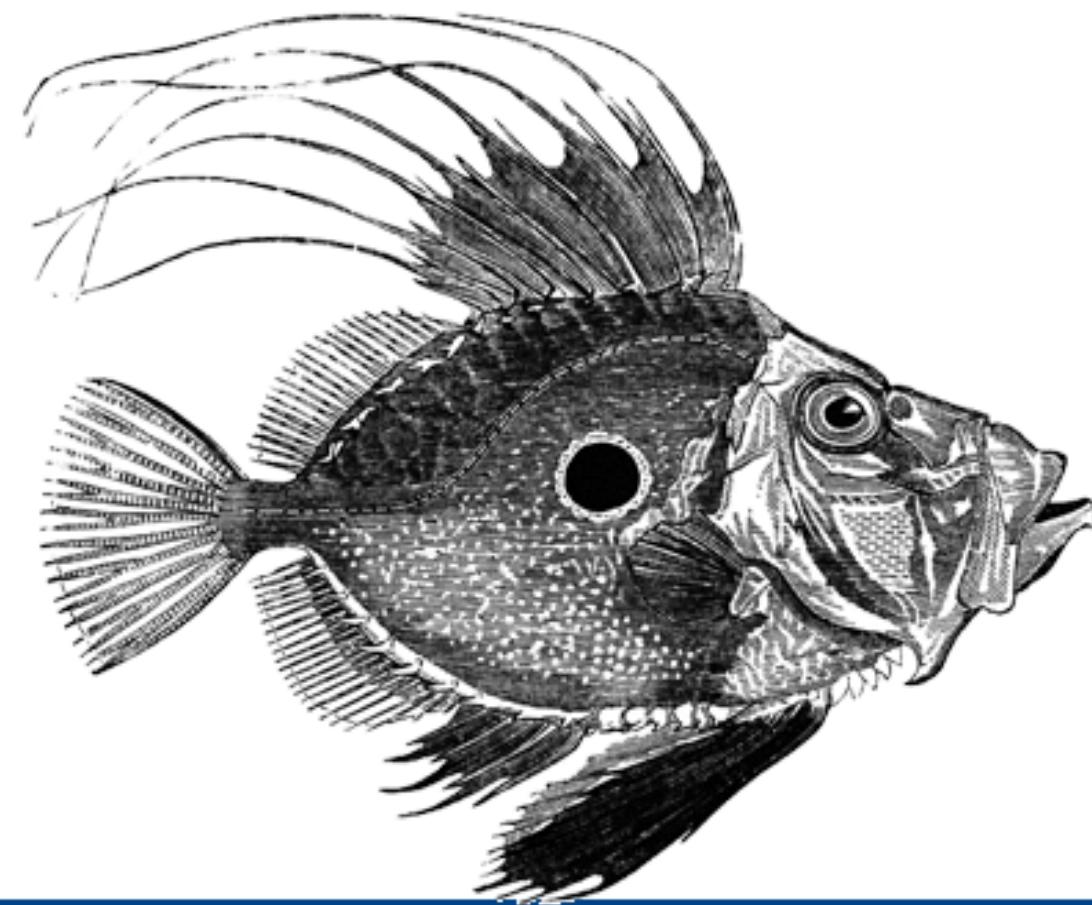
# Istio Playground

@adersberger @qaware



# Why?

*Fight Microservice Obesity*



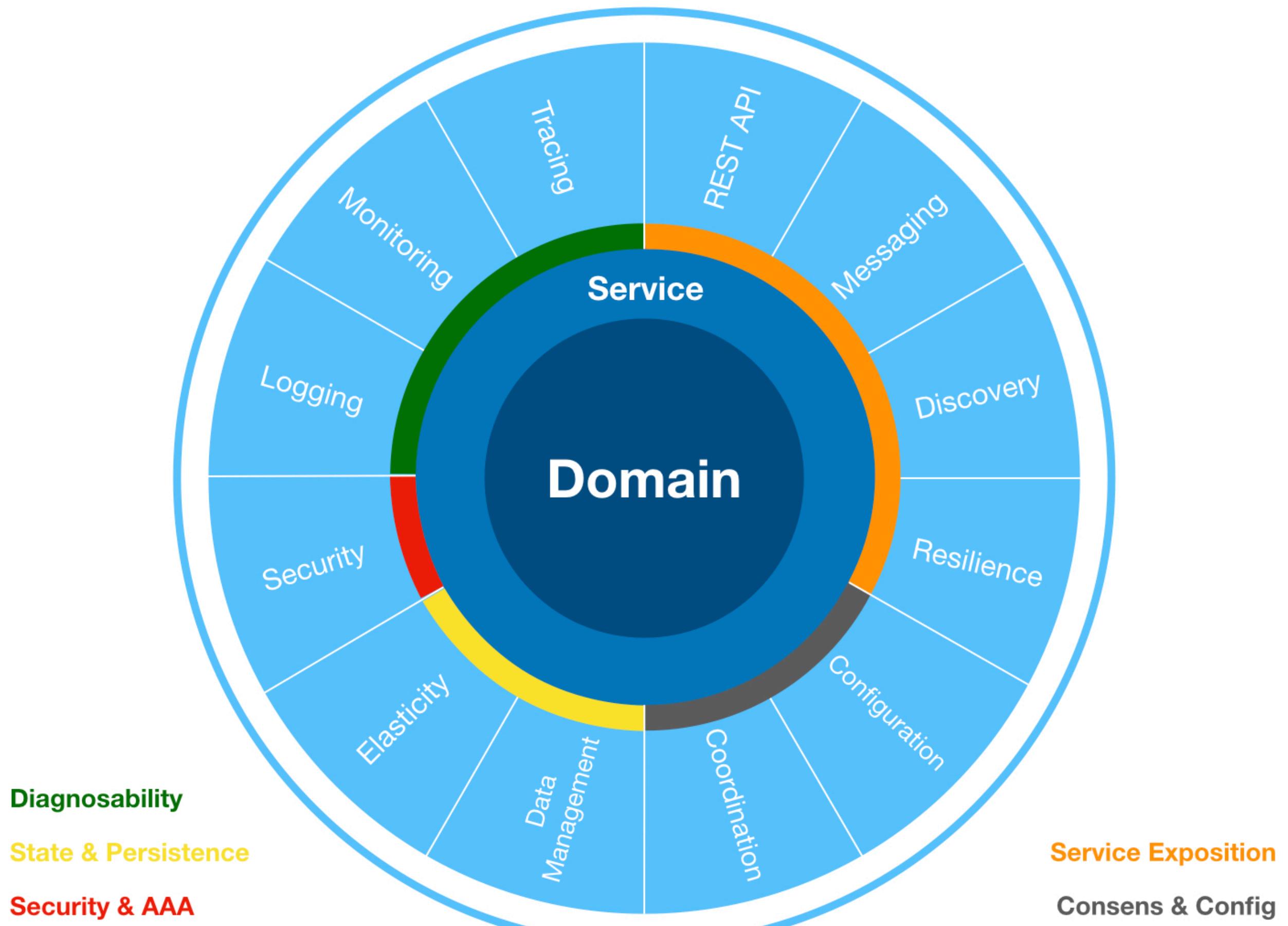
# Hype-Driven Software Development

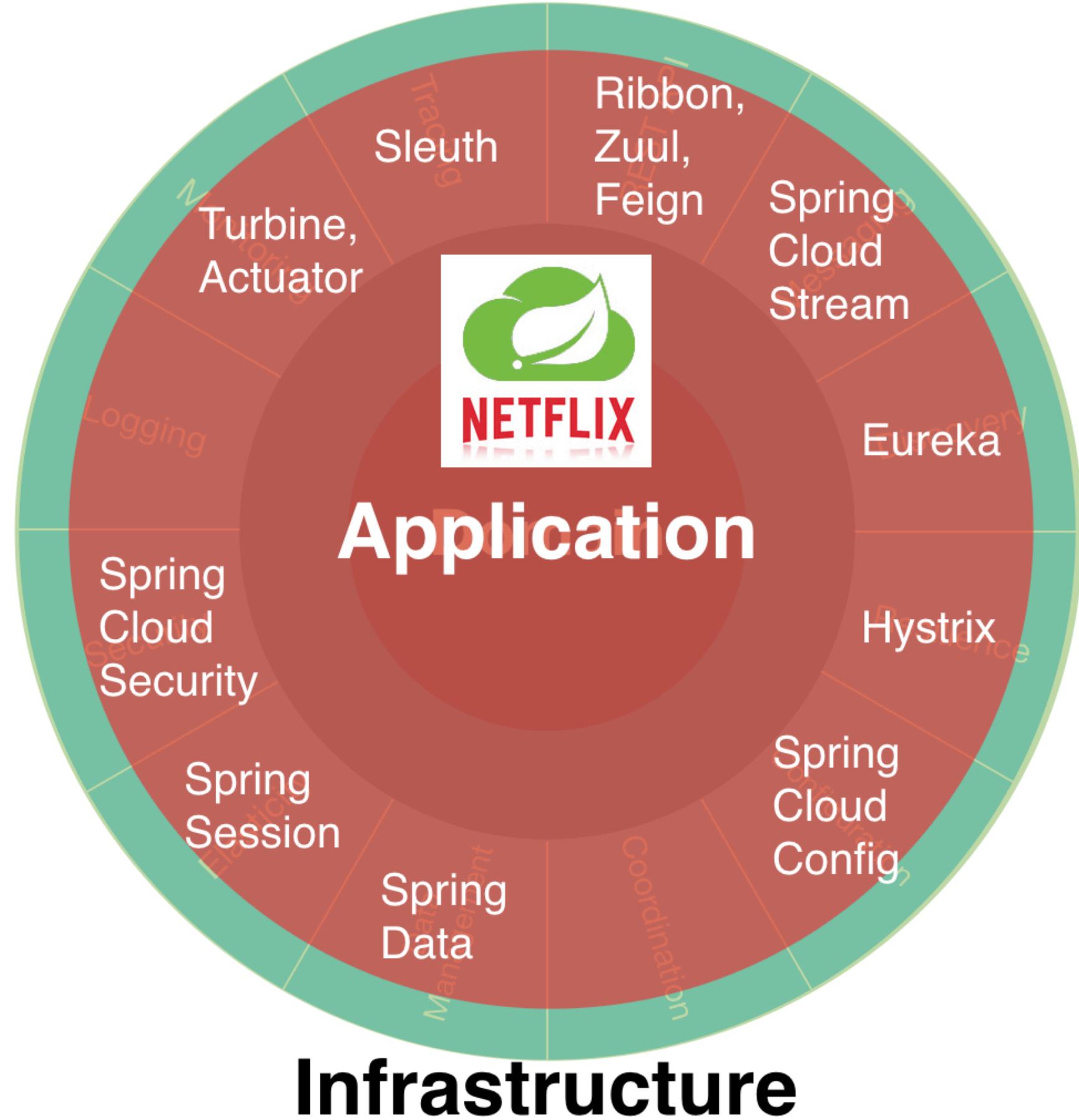
*Istio Edition*

O RLY<sup>?</sup>

*Josef Adersberger*

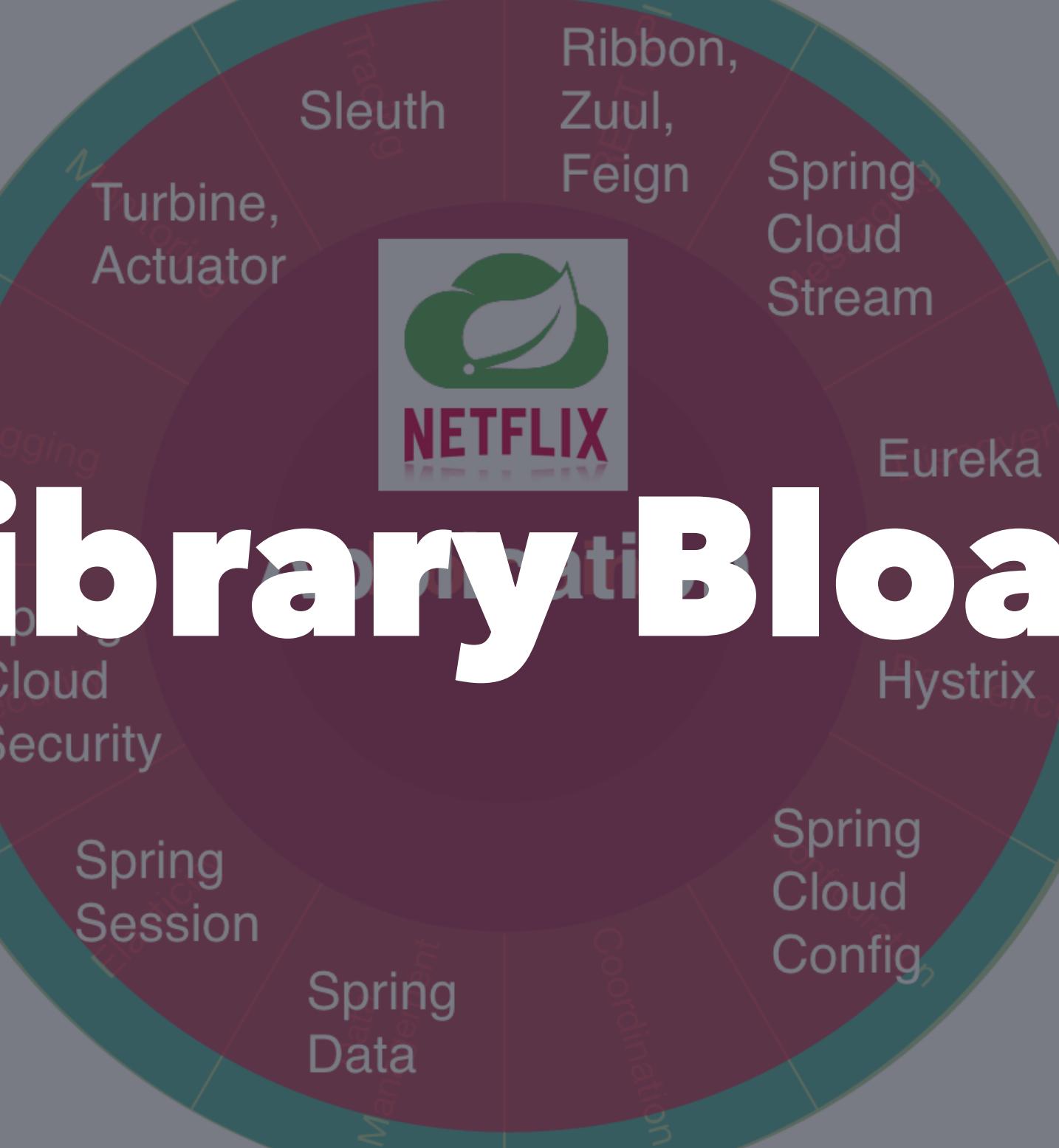
# Atomic Architecture





# library Bloat

Infrastructure



# Can we evolve existing enterprise applications into the cloud with reasonable effort?



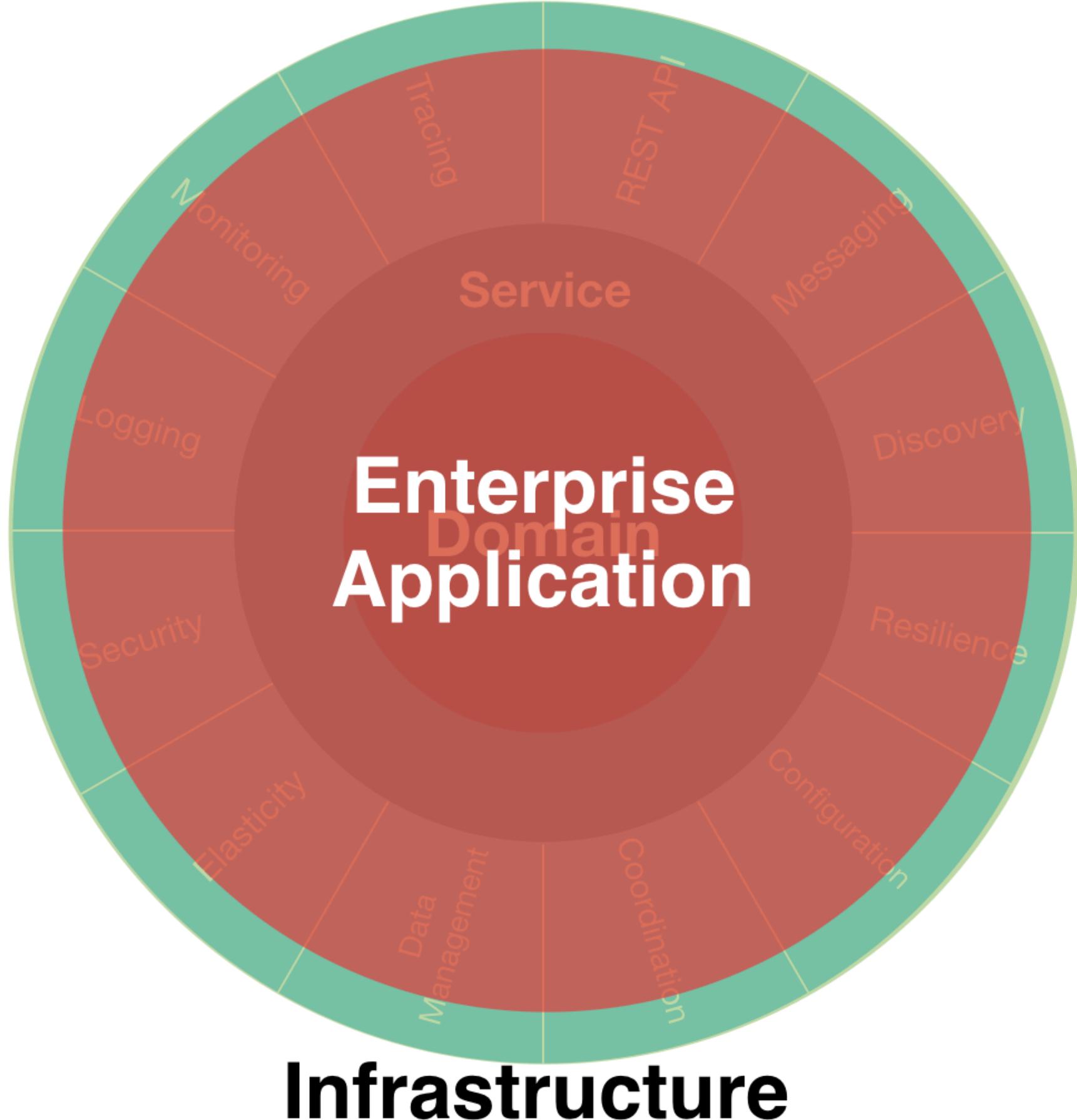
- Monolithic Deployment
- Traditional Infrastructure

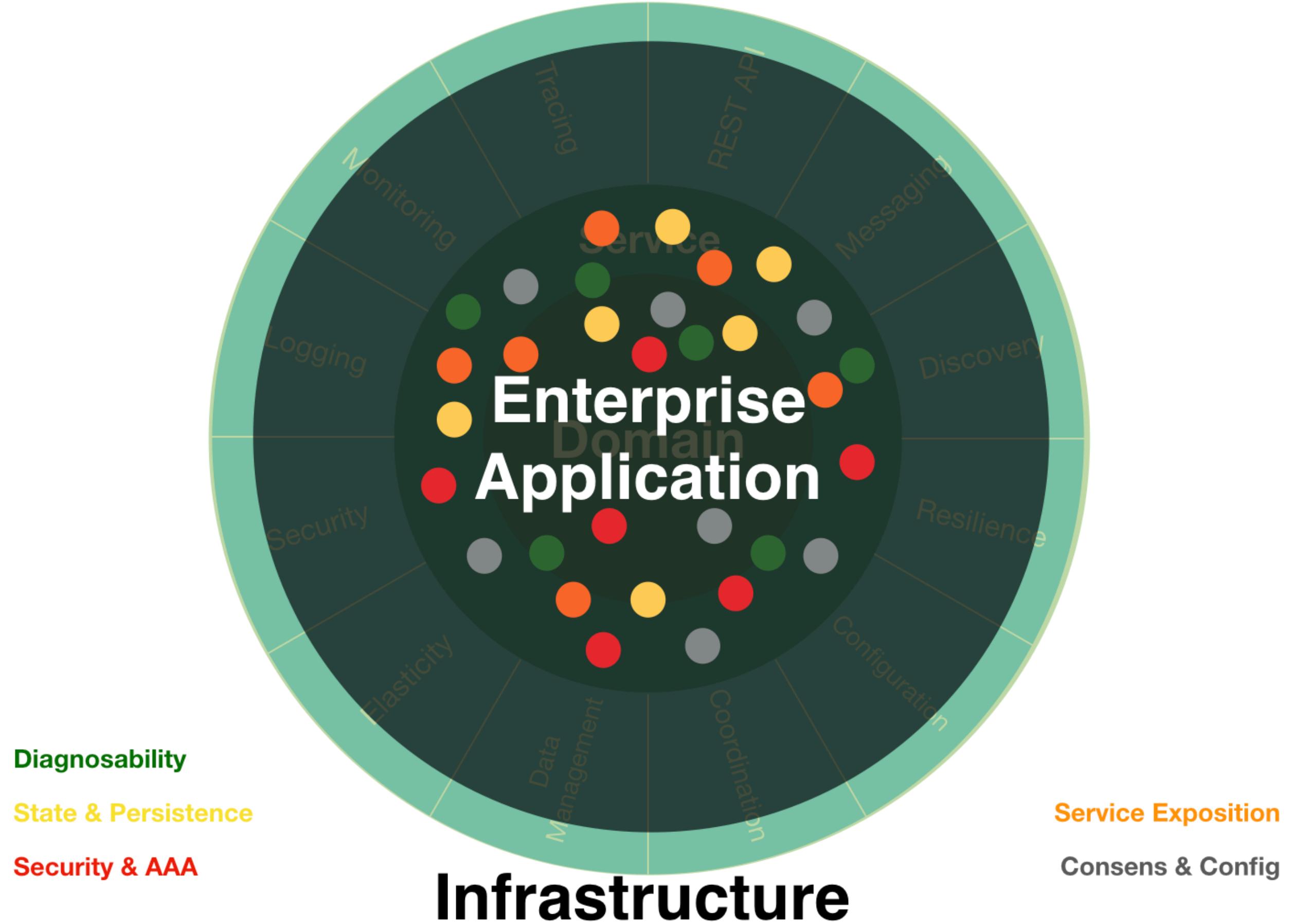


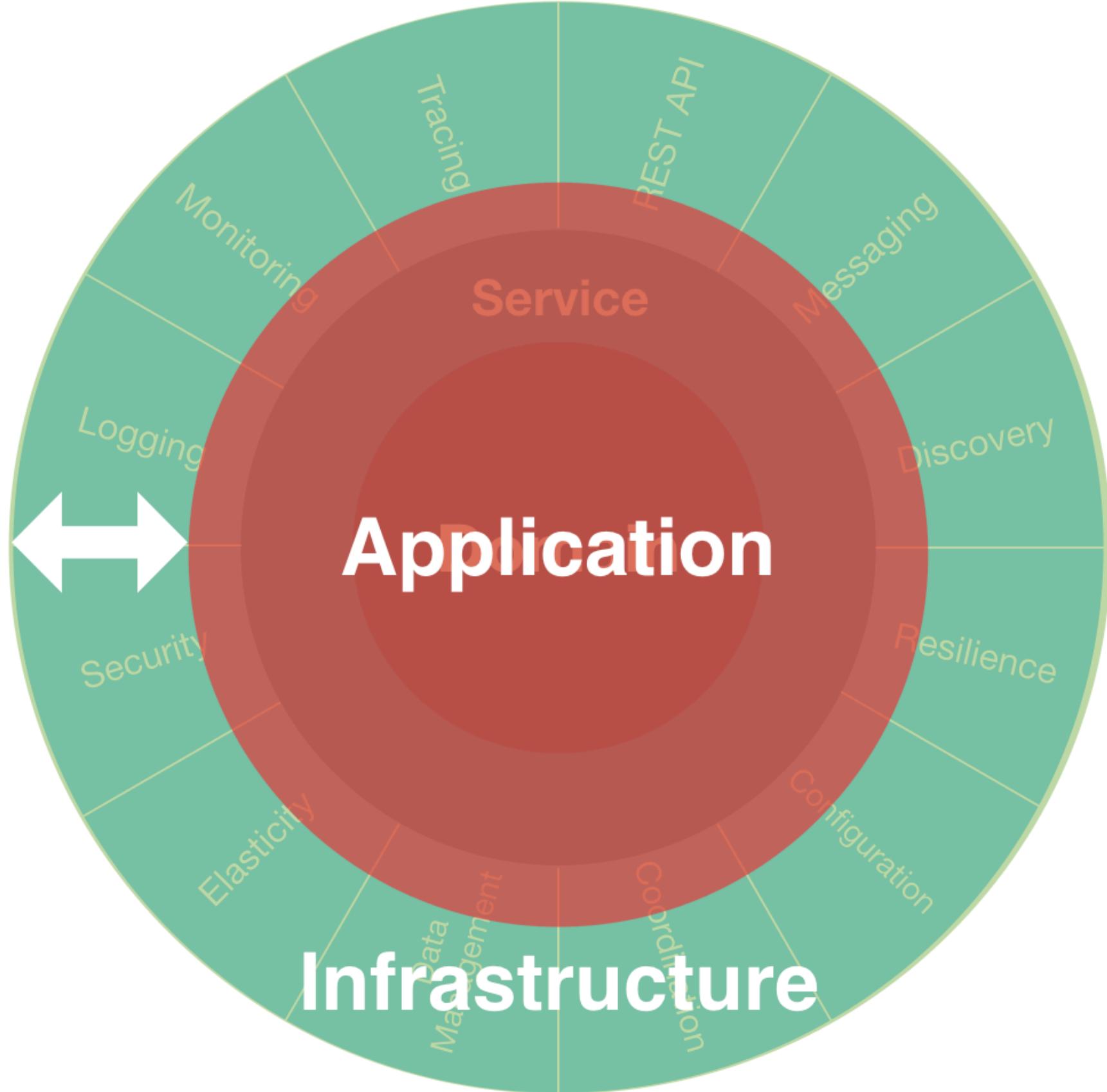
- Containerization
- 12-Factor App Principles

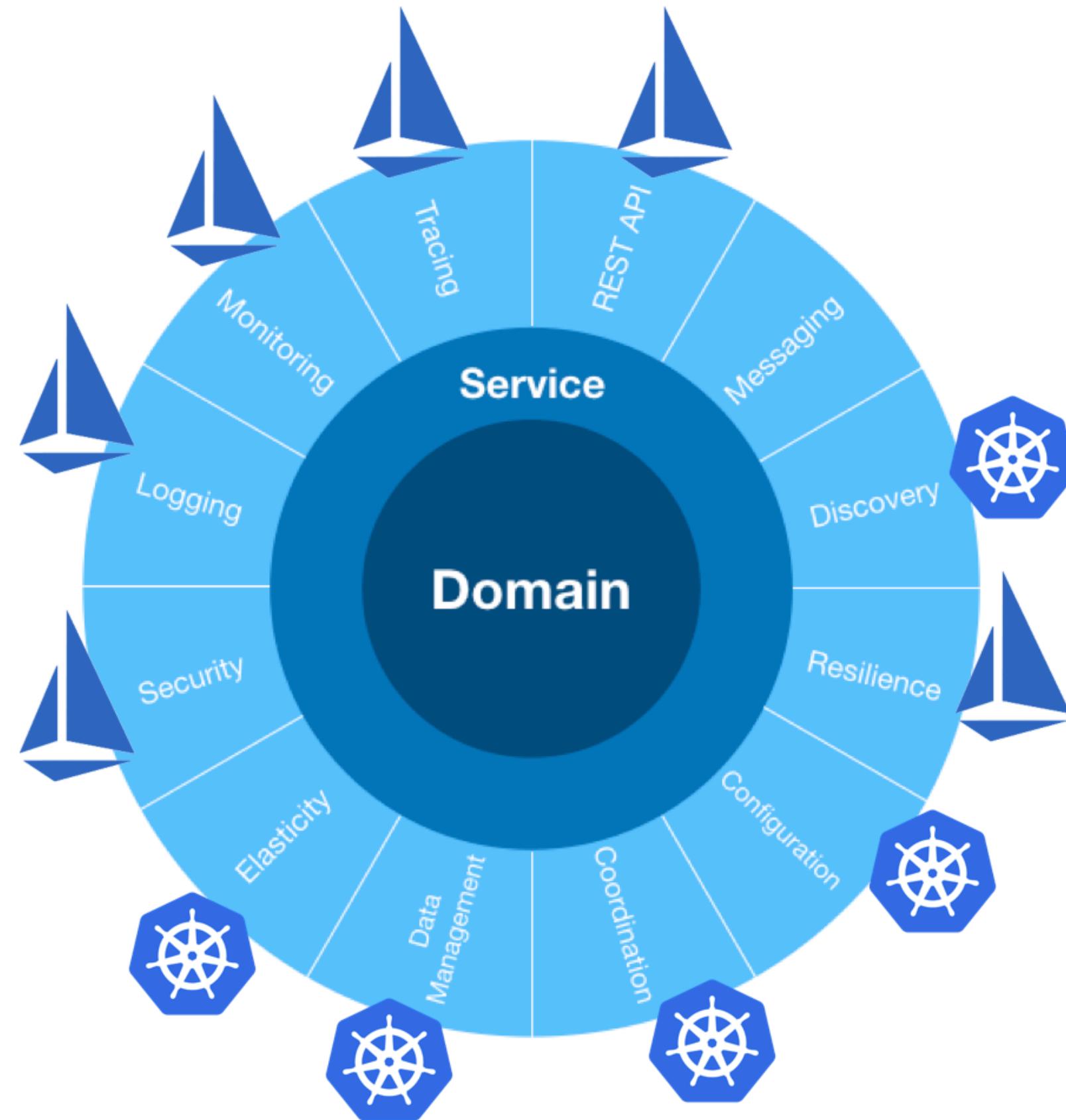


- Microservices
- Cloud-native Apps



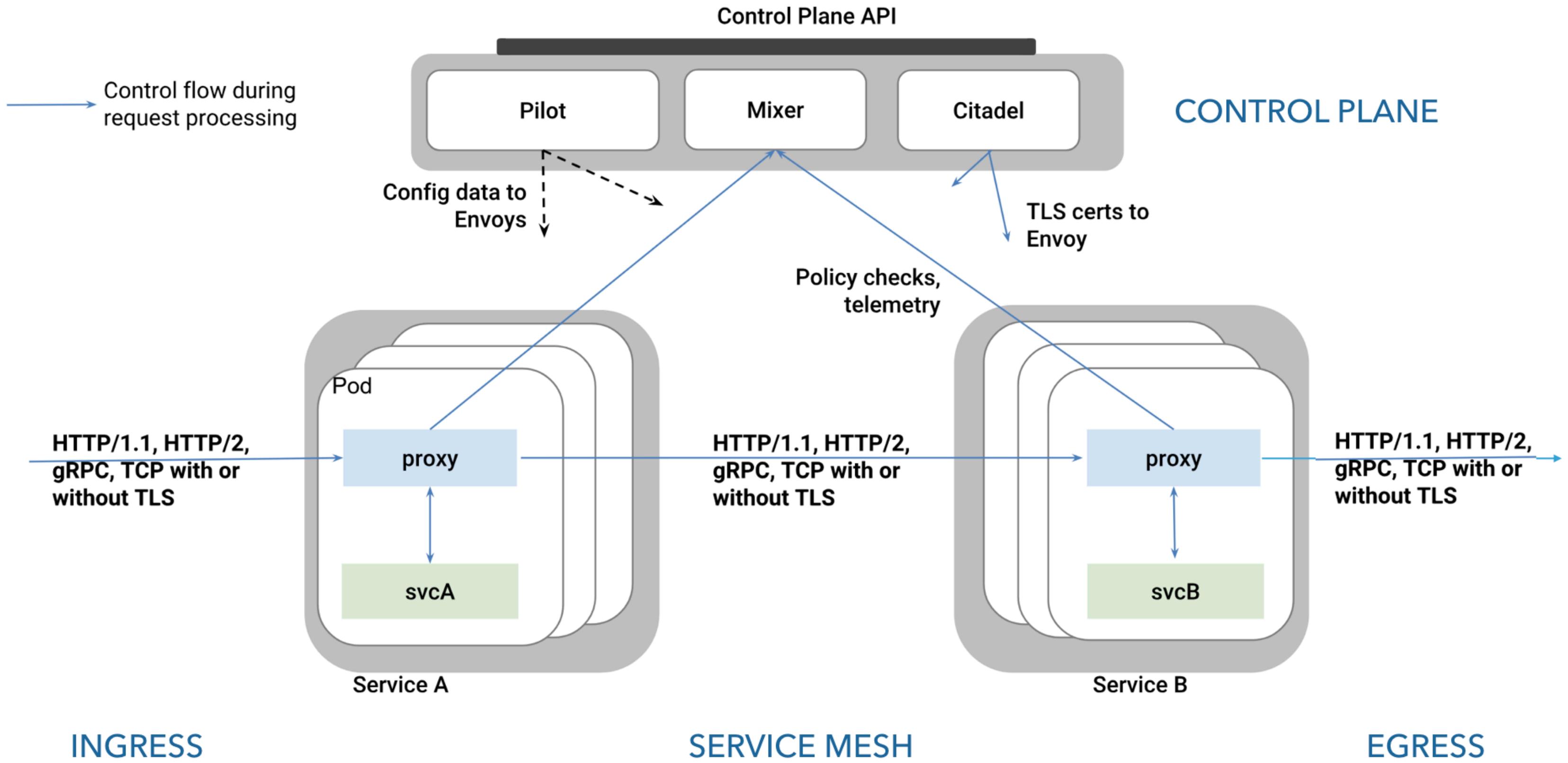




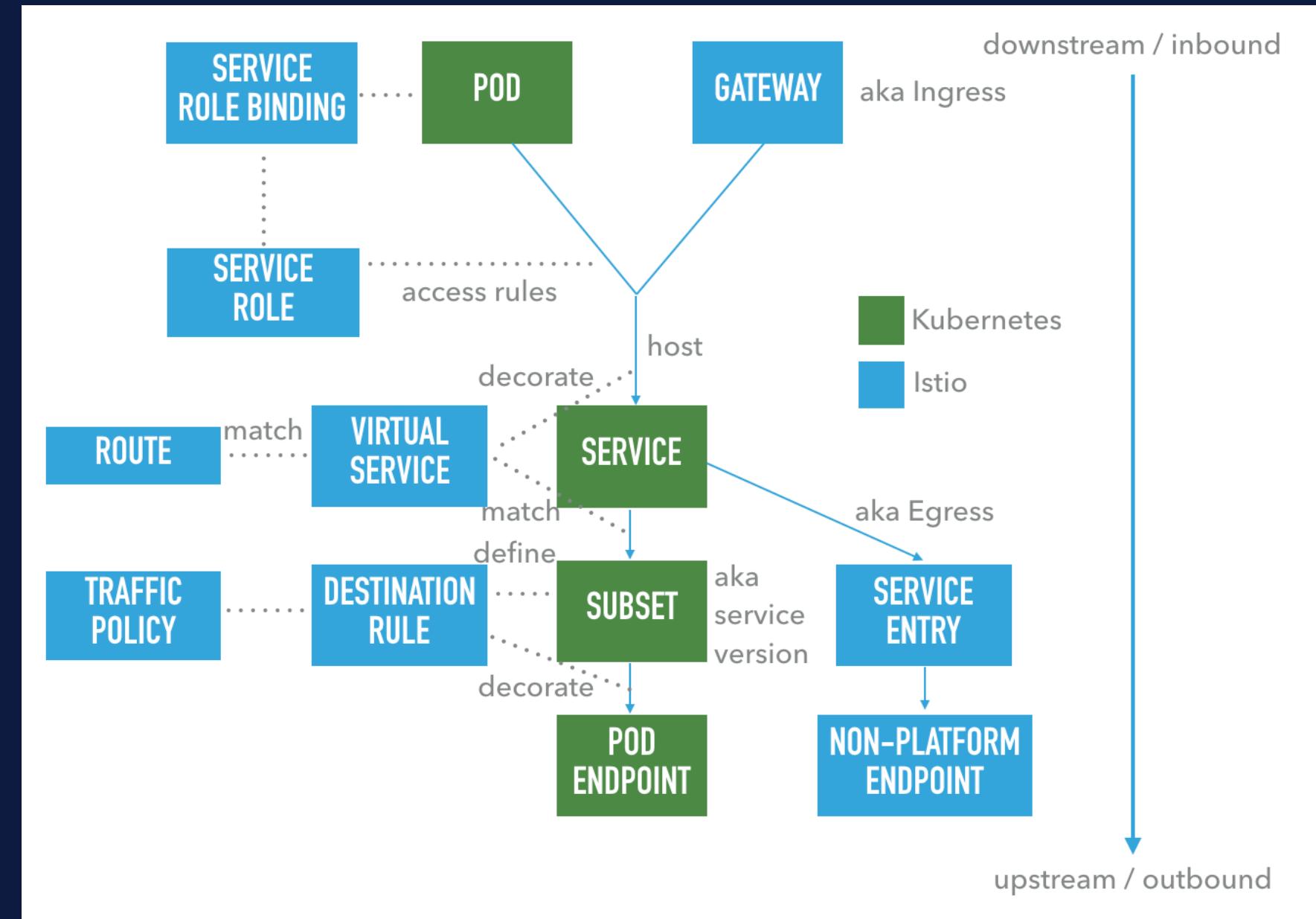


# Setting the Sails with Istio 0.8





# Istio Abstractions



# Workshop Prerequisites

- Bash
- git Client
- Text editor (like VS.Code)

# Baby Step: Grab the Code

```
git clone https://github.com/adersberger/istio-playground
```

```
cd istio-playground/code
```

# Baby Step: Install a (local) Kubernetes Cluster

Version 18.04.0-ce-mac62 (23965)

<https://www.docker.com/community-edition>

e0a85f64c1

- Preferences: enable Kubernetes
- Preferences: increase resource usage to 3 cores



Engine: 18.04.0-ce

Notary Notary: 0.6.0



Compose: 1.21.0



Credential Helper: 0.6.0

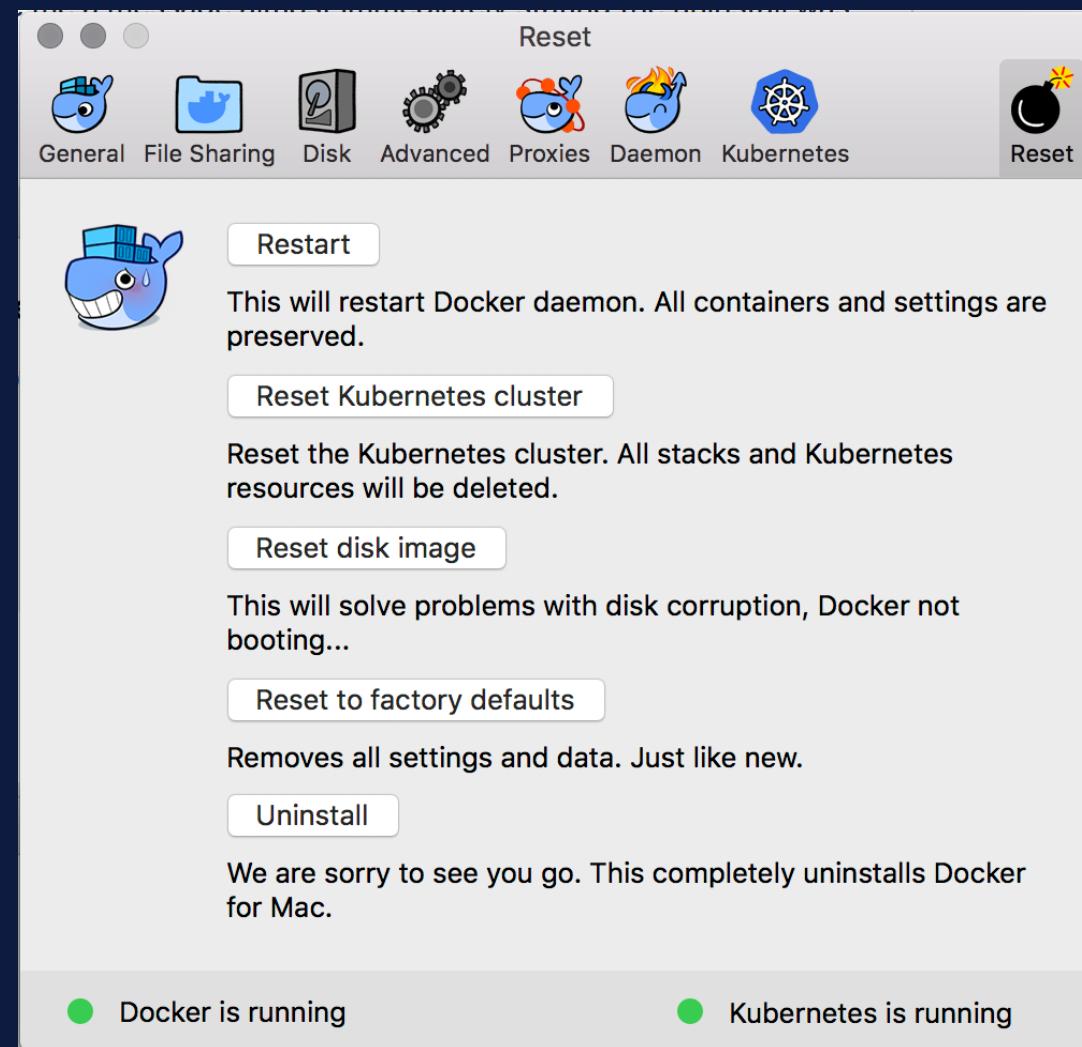


Machine: 0.14.0



Kubernetes: v1.9.6

# The Ultimate Guide to Fix Strange Kubernetes Behavior



# Setup Kubernetes Environment

```
# Switch k8s context
kubectl config use-context docker-for-desktop
# Deploy k8s dashboard
kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
# Extract id of default service account token (referred as TOKENID)
kubectl describe serviceaccount default
# Grab token and insert it into k8s Dashboard UI auth dialog
kubectl describe secret TOKENID
# Start local proxy
kubectl proxy --port=8001 &
# Open k8s Dashboard
open http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login
```

# Deploy Istio

```
curl -L https://git.io/getLatestIstio | sh -
cd istio-0.8.0
export PATH=$PWD/bin:$PATH
istioctl
```

```
# deploy Istio
# (demo setting, default deployment is via Helm)
kubectl apply -f install/kubernetes/istio-demo.yaml
kubectl get pods -n istio-system
```

```
# label default namespace to be auto-sidecarred
kubectl label namespace default istio-injection=enabled
kubectl get namespace -L istio-injection
```

# Deploy Sample Application (BookInfo)

```
kubectl apply -f samples/bookinfo/kube/bookinfo.yaml  
istioctl create -f samples/bookinfo/routing/bookinfo-gateway.yaml  
open http://localhost/productpage
```

# Deploy Sample Application (BookInfo)

```
kubectl apply -f samples/bookinfo/kube/bookinfo.yaml  
istioctl create -f samples/bookinfo/routing/bookinfo-gateway.yaml  
open http://localhost/productpage
```

**Hint: Since Istio release 0.8 you can substitute istioctl with kubectl. We're still using istioctl for clarity purposes.**

## Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

## Namespace

default ▾

## Overview

## Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

## Discovery and Load Balancing

Ingresses

Services

## Config and Storage

Config Maps

Persistent Volume Claims

Secrets

## Details

Name: productpage-v1-7bbdd59459-5bb28

Namespace: default

Labels: app: productpage pod-template-hash: 3668815015 version: v1

Annotations: sidecar.istio.io/status: {"version":"55c9e544b52e1d4e45d18a58d0b34ba4b72531e45fb6d1572c77191422556ffc","initContainers":["istio-init"],"containers":["istio-proxy"],"volumes":["istio-envoy","istio-c..."]}

Creation Time: 2018-06-11T14:26 UTC

Status: Running

QoS Class: Burstable

## Network

Node: docker-for-desktop

IP: 10.1.1.178

## Containers

## productpage

Image: istio/examples-bookinfo-productpage-v1:1.5.0

Environment variables: -

Commands: -

Args: -

## istio-proxy

Image: docker.io/istio/proxyv2:0.8.0

Environment variables: POD\_NAME: productpage-v1-7bbdd59459-5bb28

POD\_NAMESPACE: default

INSTANCE\_IP:

ISTIO\_META\_POD\_NAME: productpage-v1-7bbdd59459-5bb28

ISTIO\_META\_INTERCEPTION\_MODE: REDIRECT

Commands: -

Args: -

## Init Containers

## istio-init

Image: docker.io/istio/proxy\_init:0.8.0

Environment variables: -

Commands: -

Args: -

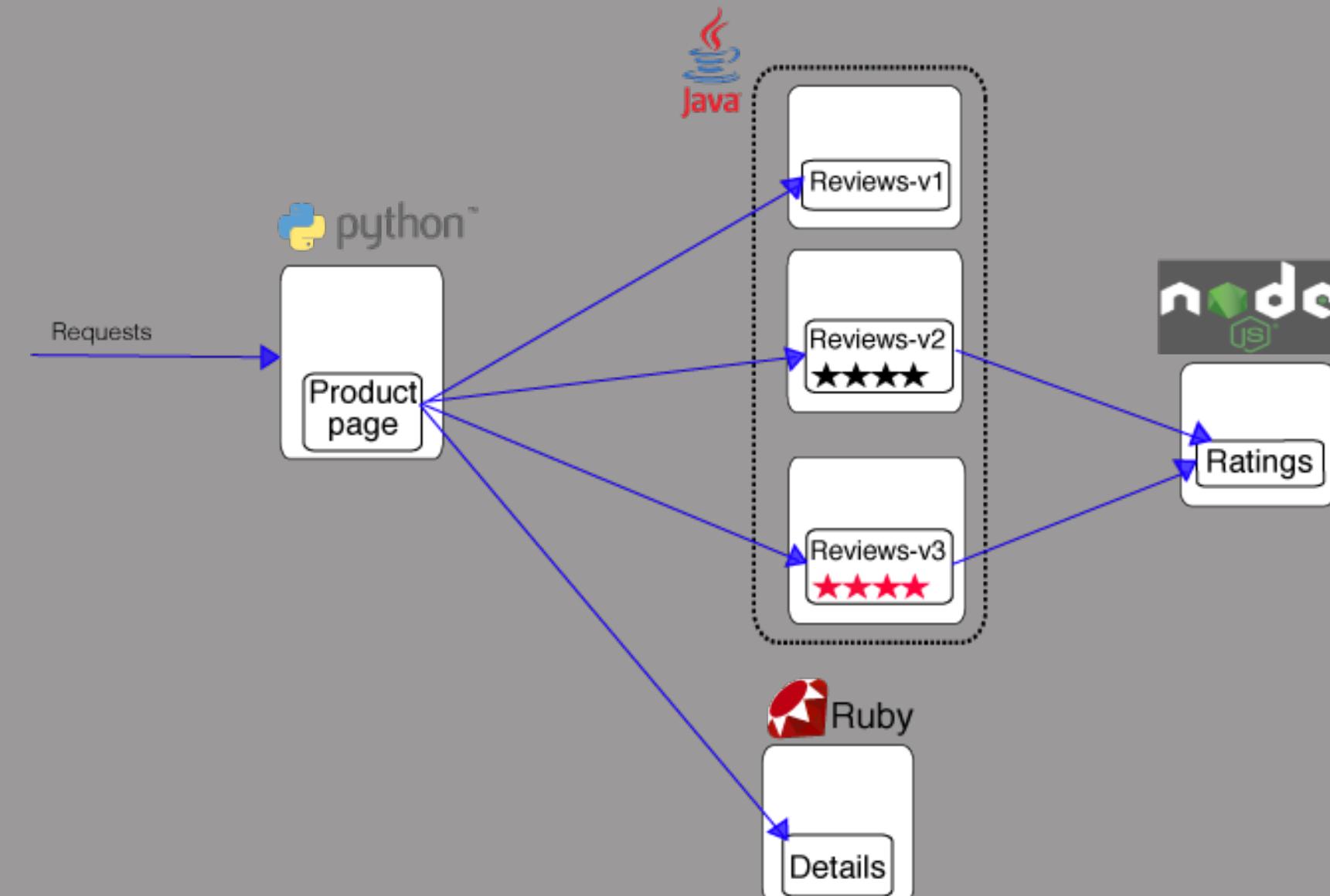
# bookinfo-gateway.yaml (1/2)

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: bookinfo-gateway
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
  hosts:
  - "*"
```

# bookinfo-gateway.yaml (2/2)

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo
spec:
  hosts:
    - "*"
  gateways:
    - bookinfo-gateway
  http:
    - match:
        - uri:
            exact: /productpage
        - uri:
            exact: /login
        - uri:
            exact: /logout
        - uri:
            prefix: /api/v1/products
  route:
    - destination:
        host: productpage
        port:
          number: 9080
```

# Sample Application: BookInfo<sup>1</sup>



<sup>1</sup> Istio BookInfo Sample (<https://istio.io/docs/guides/bookinfo.html>)

# Expose Istio Observability Tools

```
#Metrics: Prometheus
kubectl expose deployment prometheus --name=prometheus-expose
  --port=9090 --target-port=9090 --type=LoadBalancer -n=istio-system
open http://localhost:9090/graph?g0.expr=istio_request_count
```

```
#Metrics: Grafana
kubectl expose deployment grafana --name=grafana-expose
  --port=3000 --target-port=3000 --type=LoadBalancer -n=istio-system
open http://localhost:3000/d/1/istio-dashboard
```

```
#Tracing: Jaeger
kubectl expose deployment istio-tracing --name=tracing-expose
  --port=16686 --target-port=16686 --type=LoadBalancer -n=istio-system
open http://localhost:16686
```

```
#Tracing: ServiceGraph
kubectl expose service servicegraph --name=servicegraph-expose
  --port=8088 --target-port=8088 --type=LoadBalancer -n=istio-system
open http://localhost:8088/force/forcegraph.html
open http://localhost:8088/dotviz
```

# Deploy Missing Observability Feature: Log Analysis (EFK)

```
cd ..  
kubectl apply -f logging-stack.yaml  
kubectl expose deployment kibana --name=kibana-expose  
  --port=5601 --target-port=5601 --type=LoadBalancer -n=logging  
istioctl create -f fluentd-istio.yaml  
open http://localhost:5601/app/kibana
```

- Perform some requests to the BookInfo application
- Use \* as the index pattern
- Select @timestamp as the time filter field name

# fluentd-istio.yaml (1/3)

```
# Configuration for logentry instances
apiVersion: "config.istio.io/v1alpha2"
kind: logentry
metadata:
  name: newlog
  namespace: istio-system
spec:
  severity: '"info"'
  timestamp: request.time
  variables:
    source: source.labels["app"] | source.service | "unknown"
    user: source.user | "unknown"
    destination: destination.labels["app"] | destination.service | "unknown"
    responseCode: response.code | 0
    responseSize: response.size | 0
    latency: response.duration | "0ms"
  monitored_resource_type: '"UNSPECIFIED"'
```

# fluentd-istio.yaml (2/3)

```
# Configuration for a fluentd handler
apiVersion: "config.istio.io/v1alpha2"
kind: fluentd
metadata:
  name: handler
  namespace: istio-system
spec:
  address: "fluentd-es.logging:24224"
```

# fluentd-istio.yaml (3/3)

```
# Rule to send logentry instances to the fluentd handler
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: newlogtofluentd
  namespace: istio-system
spec:
  match: "true" # match for all requests
  actions:
    - handler: handler.fluentd
      instances:
        - newlog.logentry
```

# Stimulate!

```
slapper -rate 4 -targets ./target -workers 2 -maxY 15s
```

Download from: <https://github.com/adersberger/slapper/releases/tag/0.1>

# Slapper<sup>2</sup> in action

```
sent: 773    in-flight: 0   rate:    4/4 RPS responses: [200]: 7
72  [503]: 1  0/  0]
<1.0 ms: [  0/  0]
<1.0 ms: [  0/  0]
1.0-1.3 ms: [  0/  0]
1.3-1.7 ms: [  0/  0]
1.7-2.1 ms: [  0/  0]
2.1-2.8 ms: [  0/  0]
2.8-3.5 ms: [  0/  0]
3.5-4.6 ms: [  0/  0]
4.6-5.9 ms: [  0/  0]
5.9-7.6 ms: [  0/  0]
7.6-9.8 ms: [  0/  0]
10- 13 ms: [  0/  0]
13- 16 ms: [  0/  0]
16- 21 ms: [  0/  0]
21- 27 ms: [  2/  0] ****
27- 35 ms: [ 11/  0] ****
35- 45 ms: [ 10/  0] ****
45- 57 ms: [  9/  0] ****
57- 74 ms: [  7/  0] ****
74- 95 ms: [  0/  0]
95-122 ms: [  0/  0]
```

---

<sup>2</sup> Key bindings:

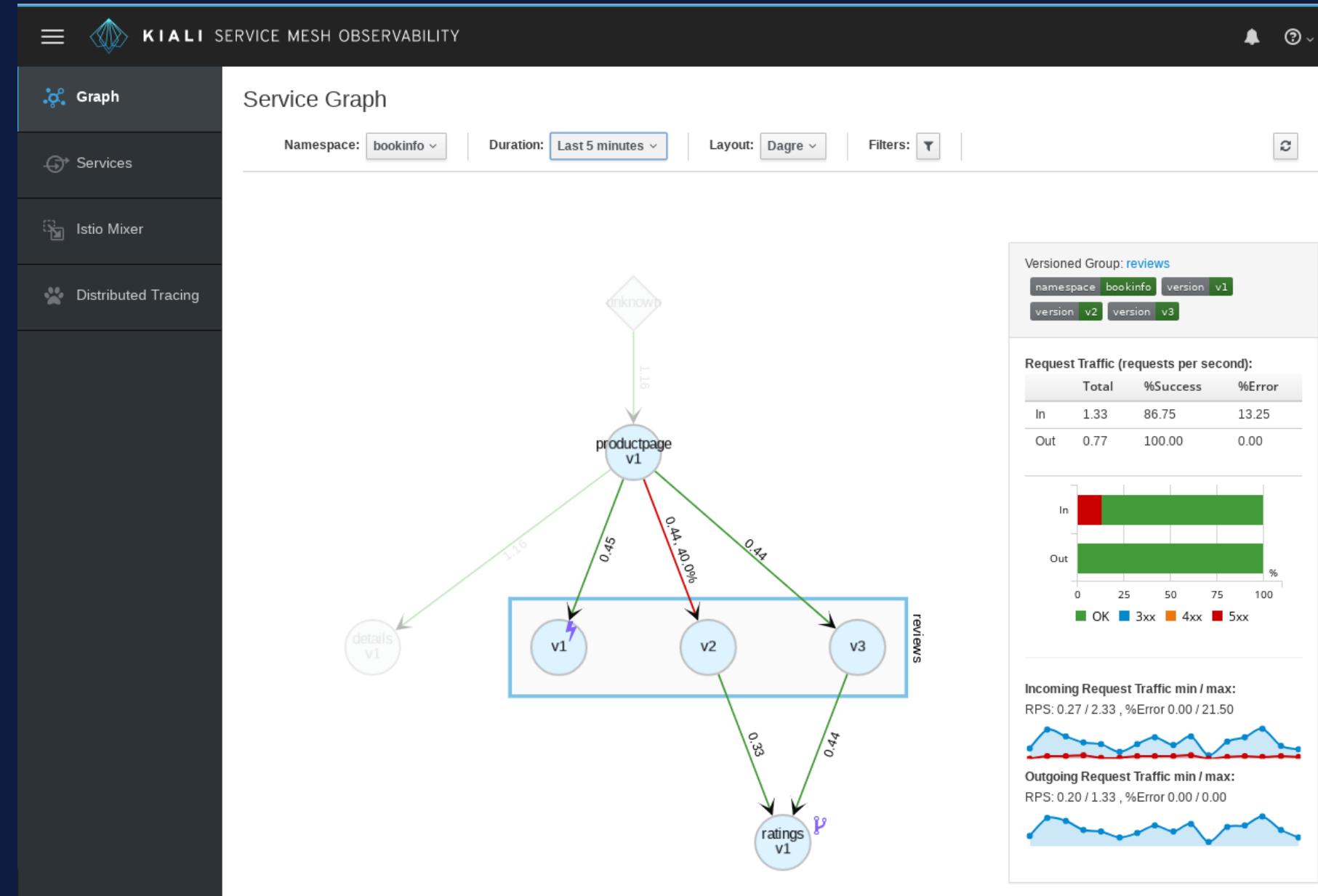
q, ctrl-c - quit

r - reset stats

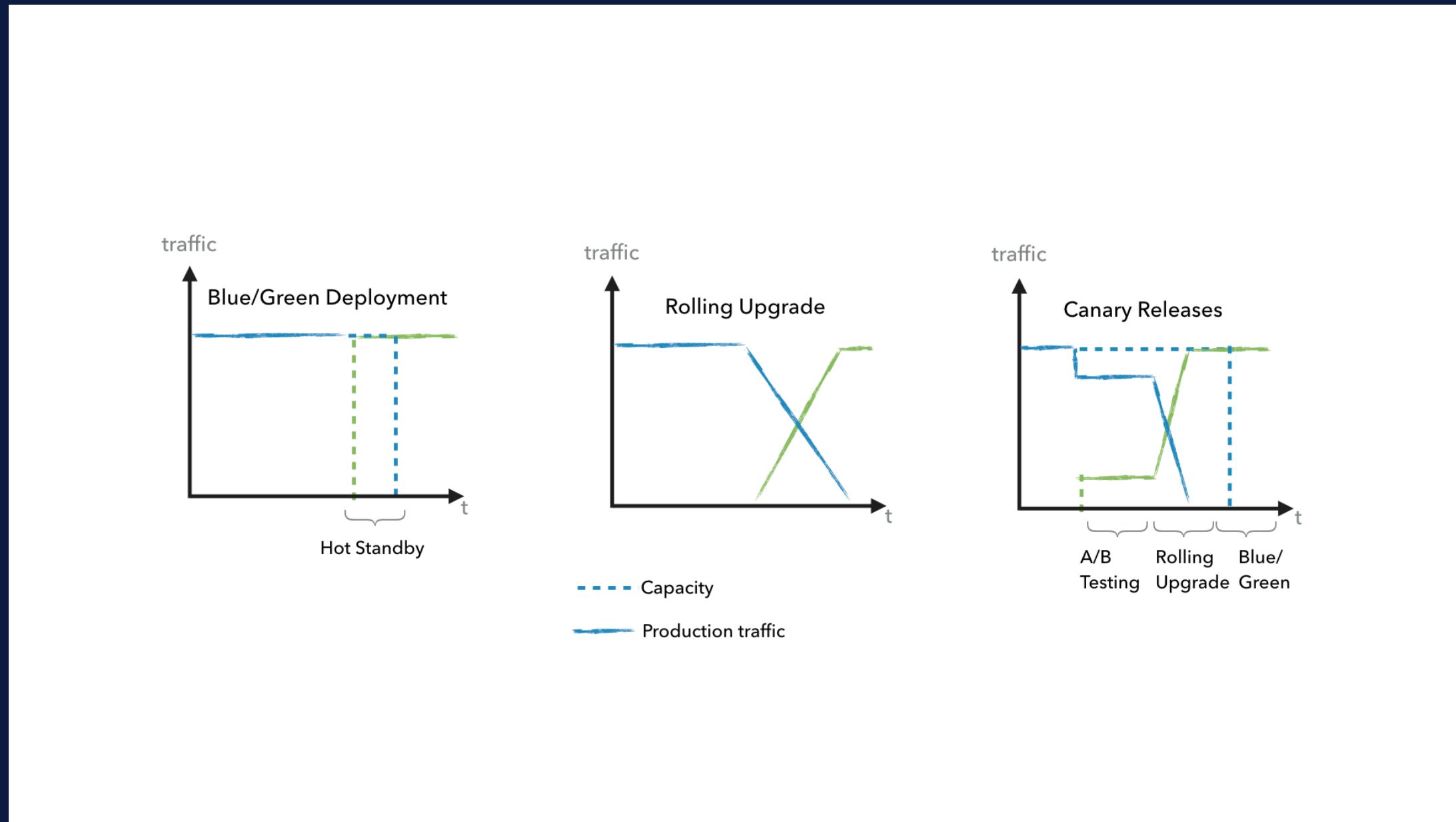
k - increase rate by 100 RPS

j - decrease rate by 100 RPS

# Observability Outlook: Kiali



# Release Patterns



# Canary Releases: A/B Testing

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
    - match:
      - headers:
          cookie:
            regex: "^(.*?;)?(user=jason)(;.*)?$$"
    route:
      - destination:
          host: reviews
          subset: v2
      - route:
          - destination:
              host: reviews
              subset: v1
```

# Canary Releases: A/B Testing

```
istioctl create -f samples/bookinfo/routing/route-rule-reviews-test-v2.yaml
```

```
open http://localhost/productpage
```

- login as "jason" / "jason" leads to v2 (black stars)
- anonymous user leads to v1 (no stars)

# Canary Releases: Rolling Upgrade

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
    - route:
        - destination:
            host: reviews
            subset: v1
            weight: 50
        - destination:
            host: reviews
            subset: v3
            weight: 50
```

```
istioctl replace -f samples/bookinfo/routing/route-rule-rev
```

# Canary Releases: Blue/Green

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
    - route:
        - destination:
            host: reviews
            subset: v3
```

```
istioctl replace -f samples/bookinfo/routing/route-rule-reviews-v3.yaml
```

# Time to Play!

Traffic Management	Resiliency	Security	Observability
Request Routing	Timeouts	mTLS	Metrics
Load Balancing	Circuit Breaker	Role-Based Access Control	Logs
Traffic Shifting	Health Checks (active, passive)	Workload Identity	Traces
Traffic Mirroring	Retries	Authentication Policies	
Service Discovery	Rate Limiting	CORS Handling	
Ingress, Egress	Delay & Fault Injection		
API Specification	Connection Pooling		

<https://istio.io/docs/tasks>

# Thank you!



[josef.adersberger@qaware.de](mailto:josef.adersberger@qaware.de)



@adersberger

[TWITTER.COM/QAWARE](http://TWITTER.COM/QAWARE) - [SLIDEShare.NET/QAWARE](http://SLIDEShare.NET/QAWARE)