



KubeCon



CloudNativeCon

Europe 2018



# Istio by Example

Josef Adersberger, CTO QAware

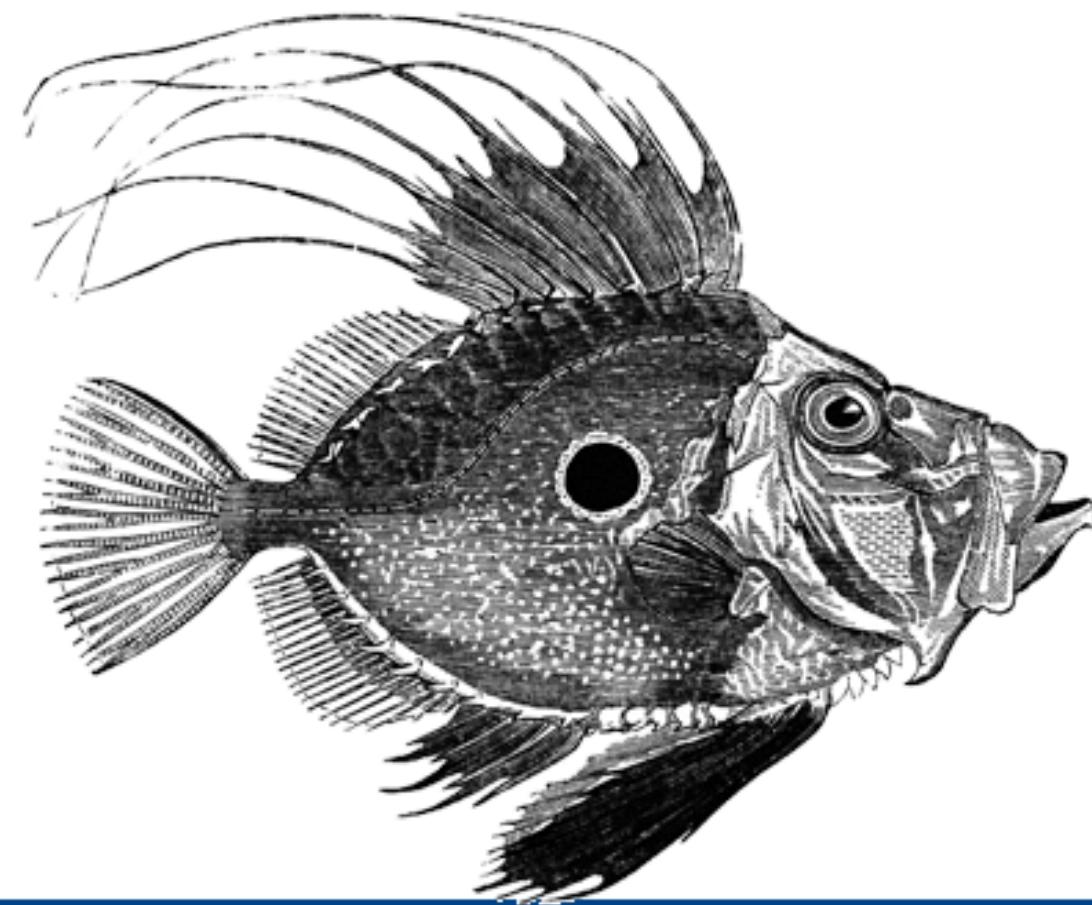


@adersberger



# Why?

*Fight Microservice Obesity*



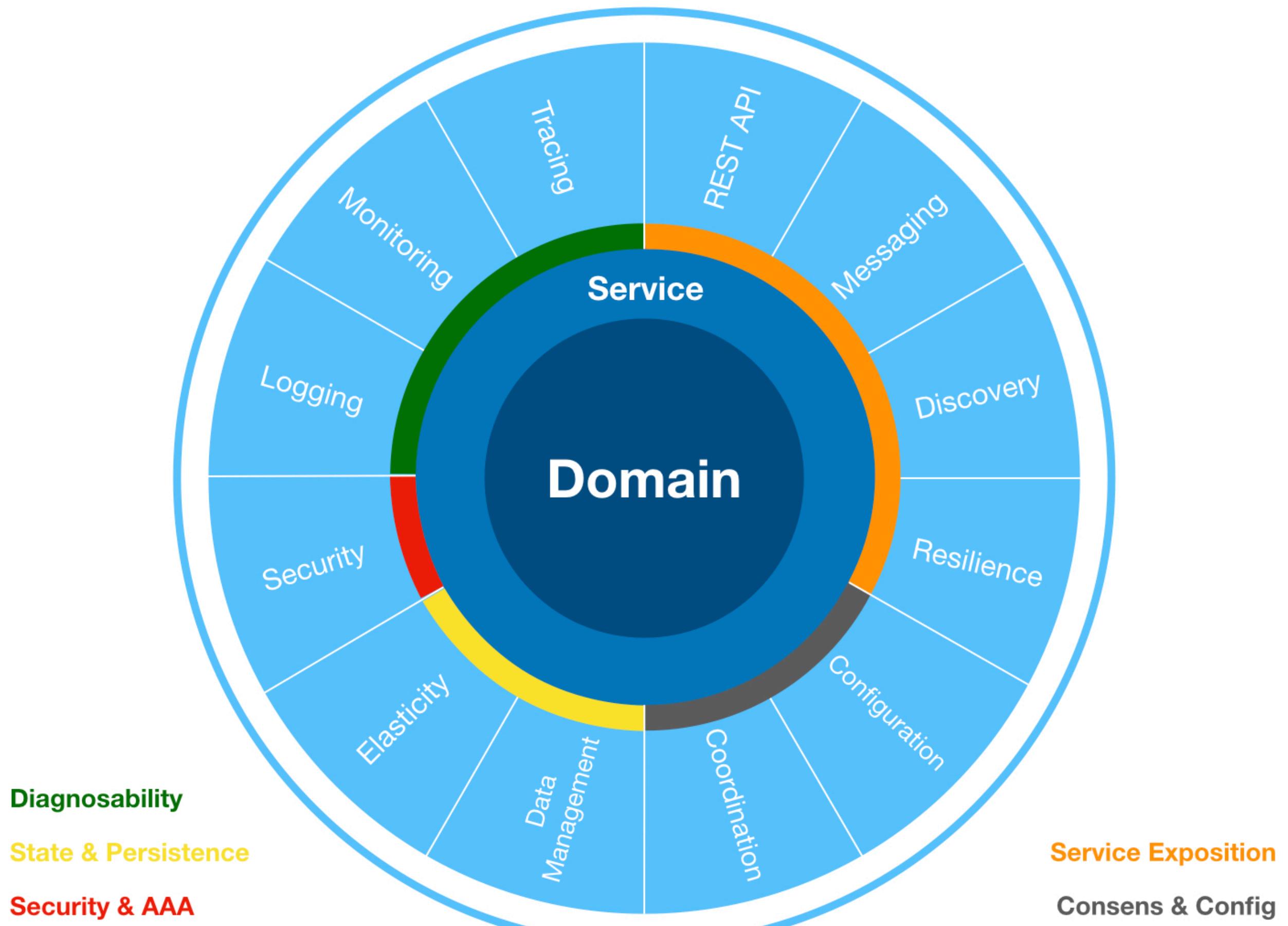
# Hype-Driven Software Development

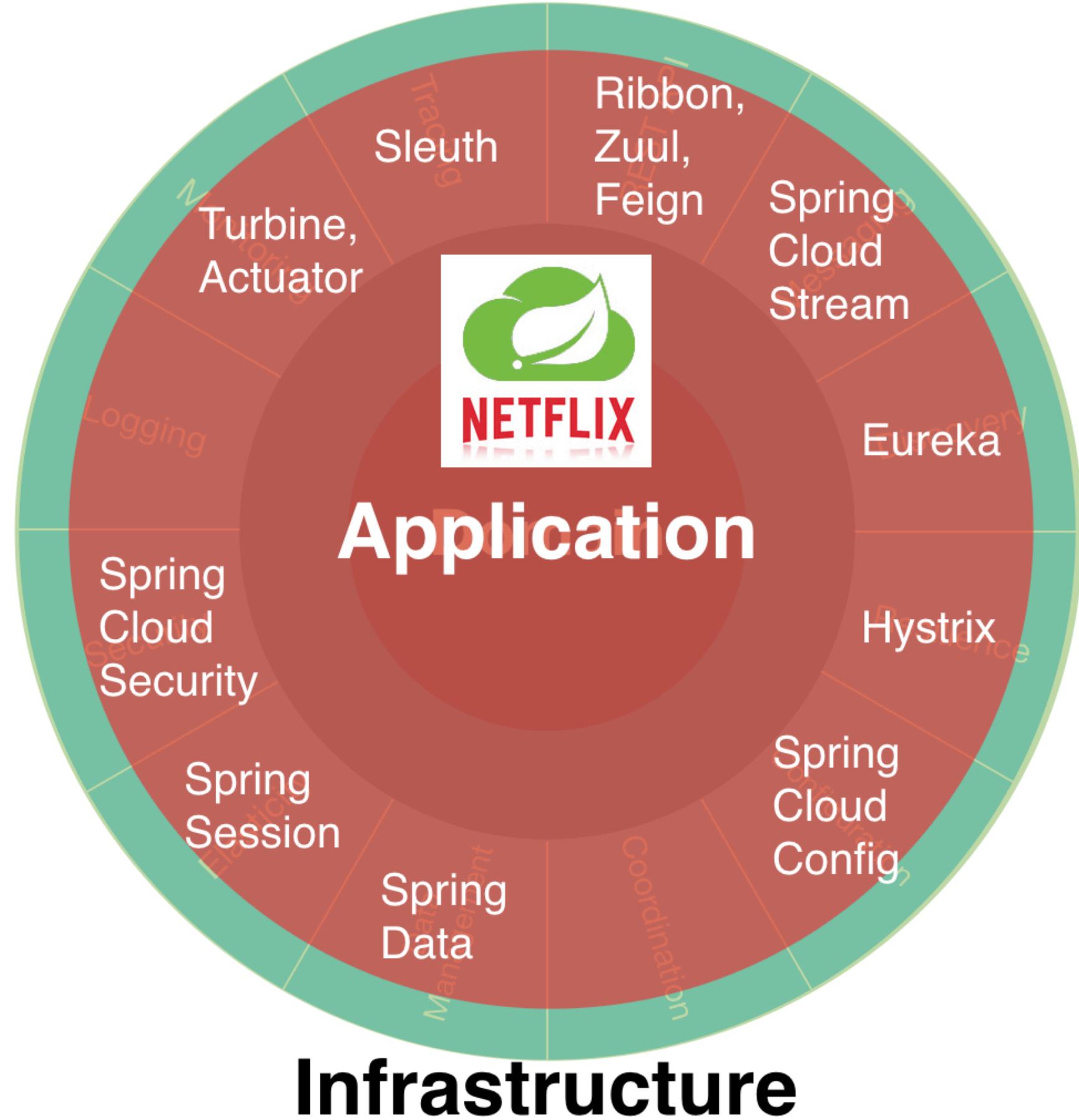
*Istio Edition*

O RLY<sup>?</sup>

*Josef Adersberger*

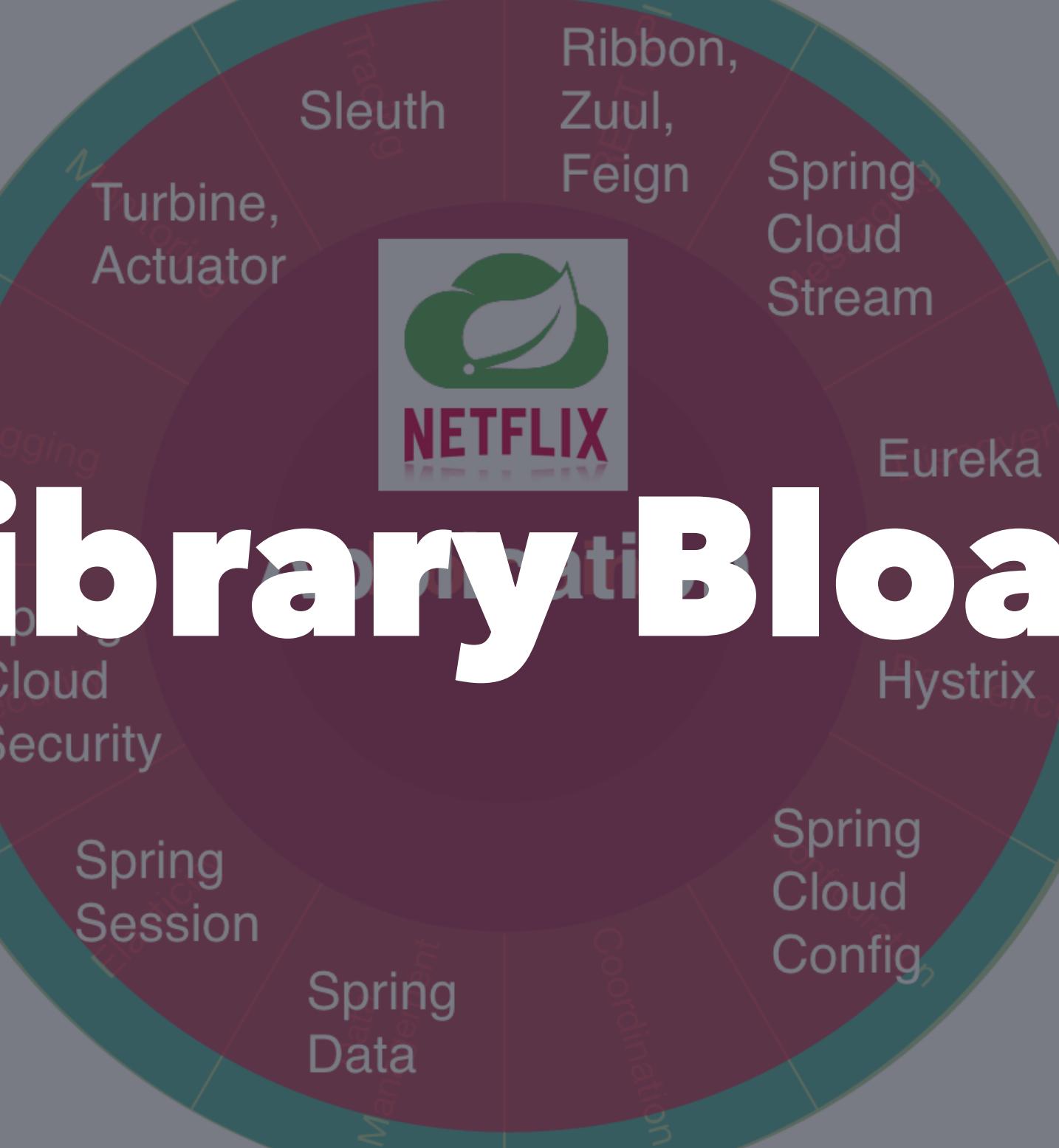
# Atomic Architecture

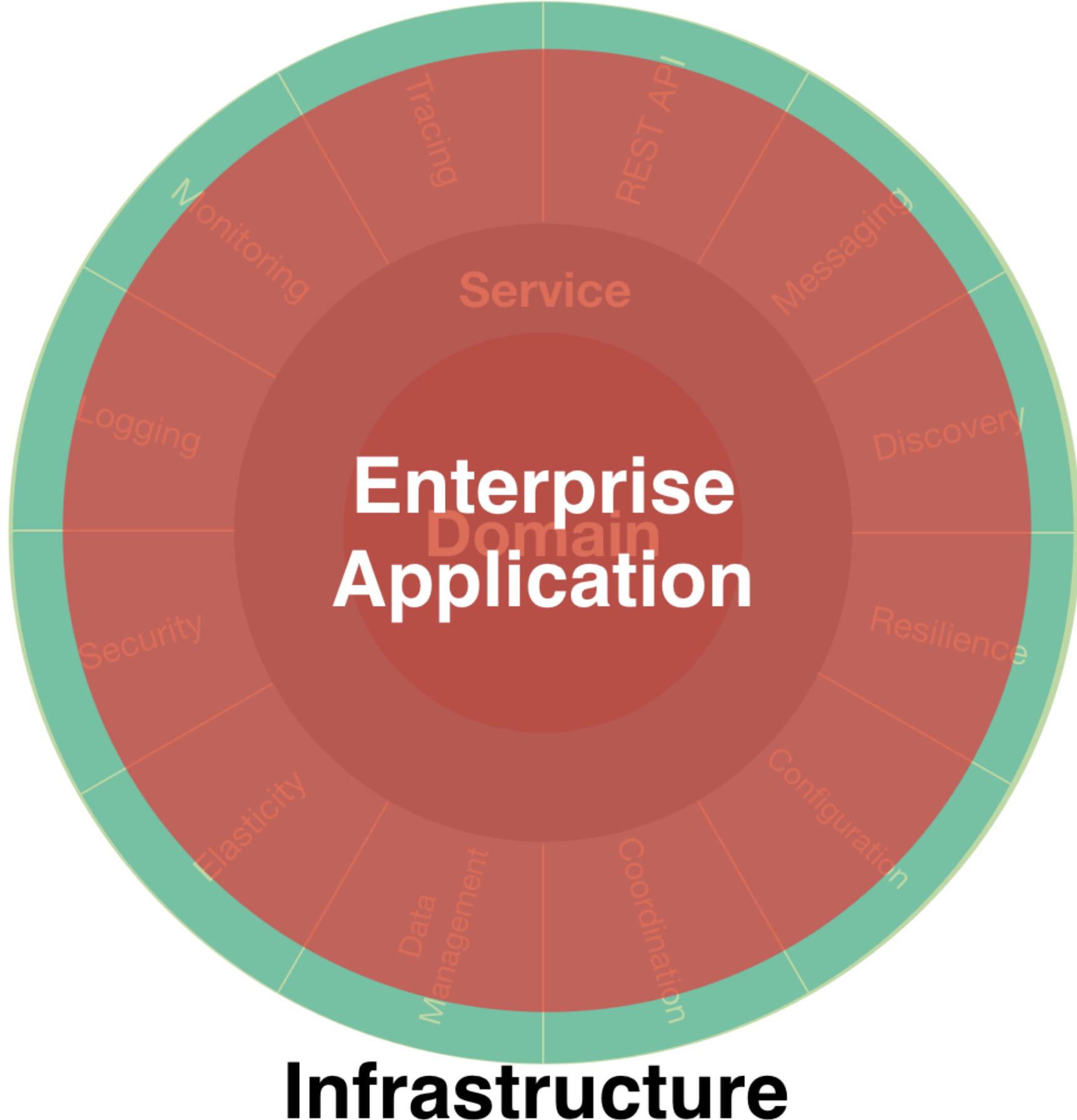


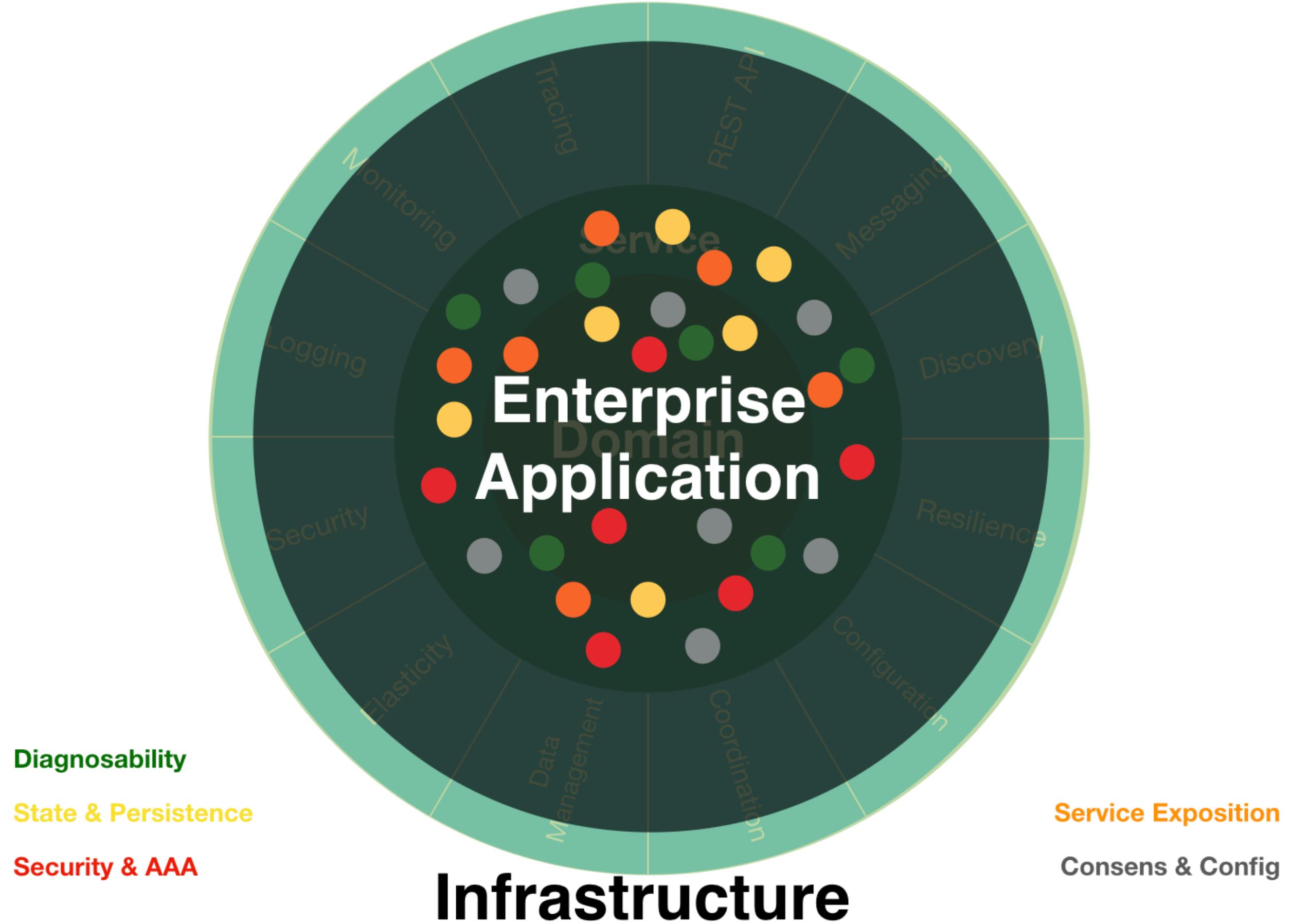


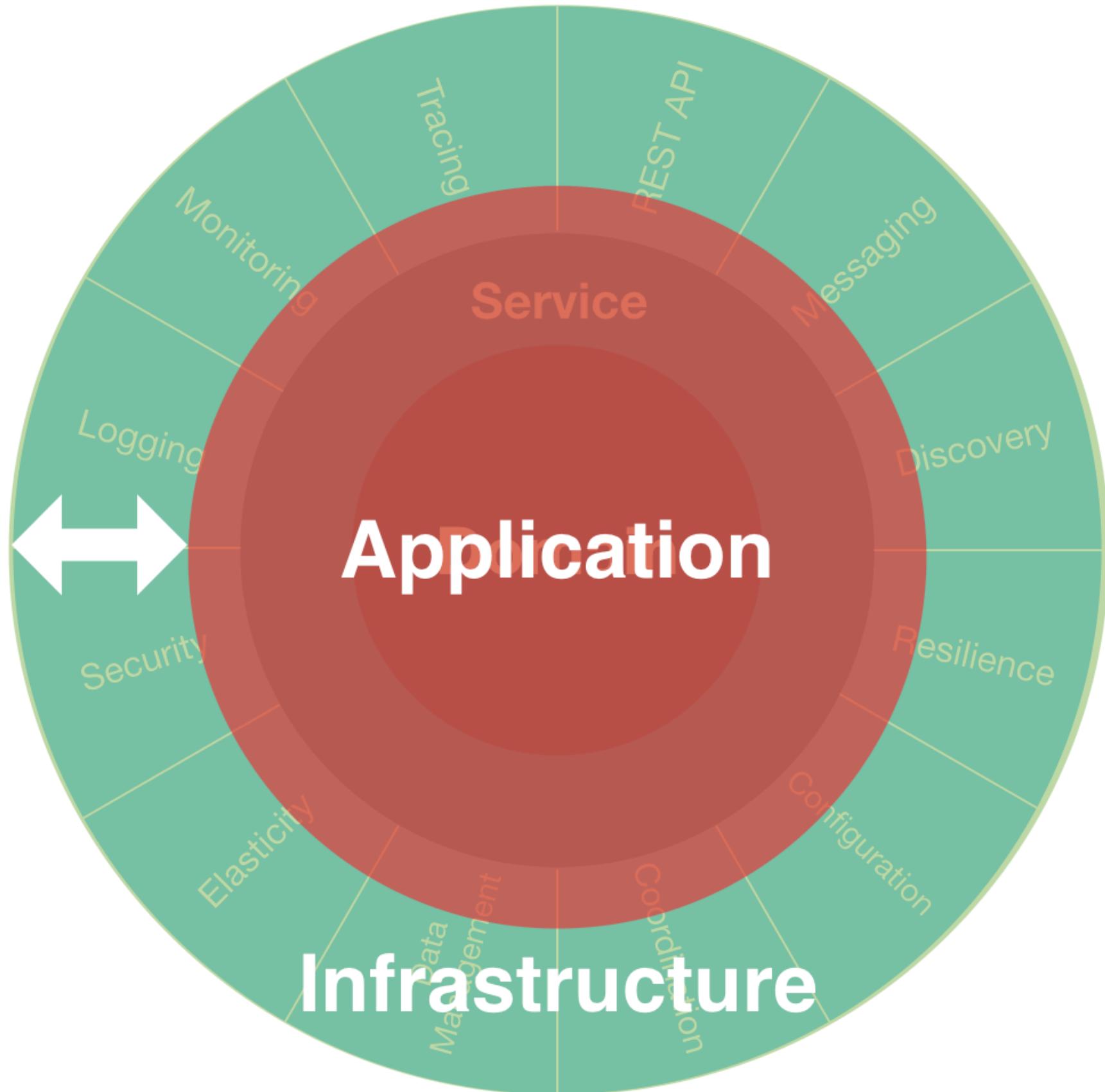
# library Bloat

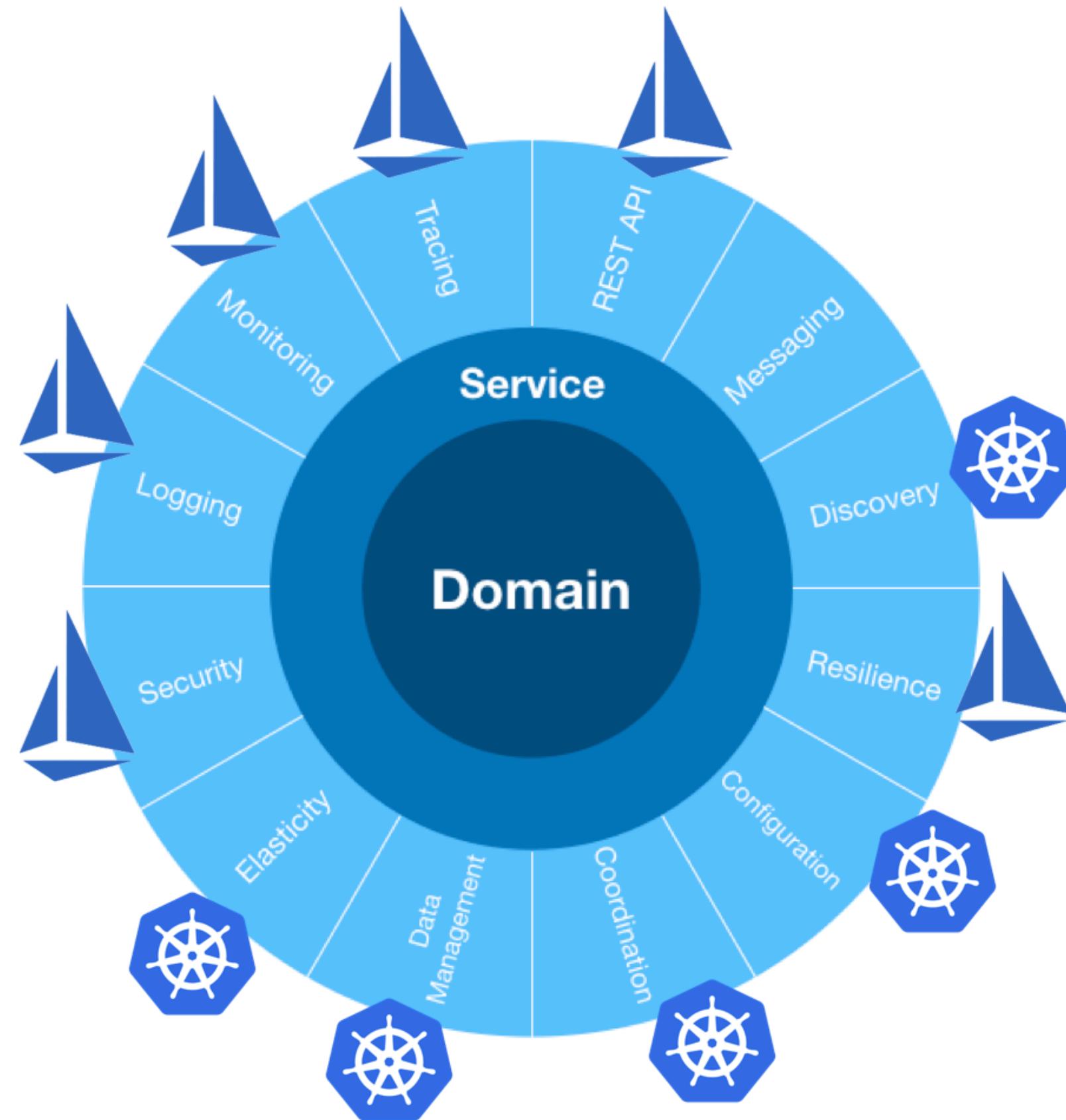
Infrastructure









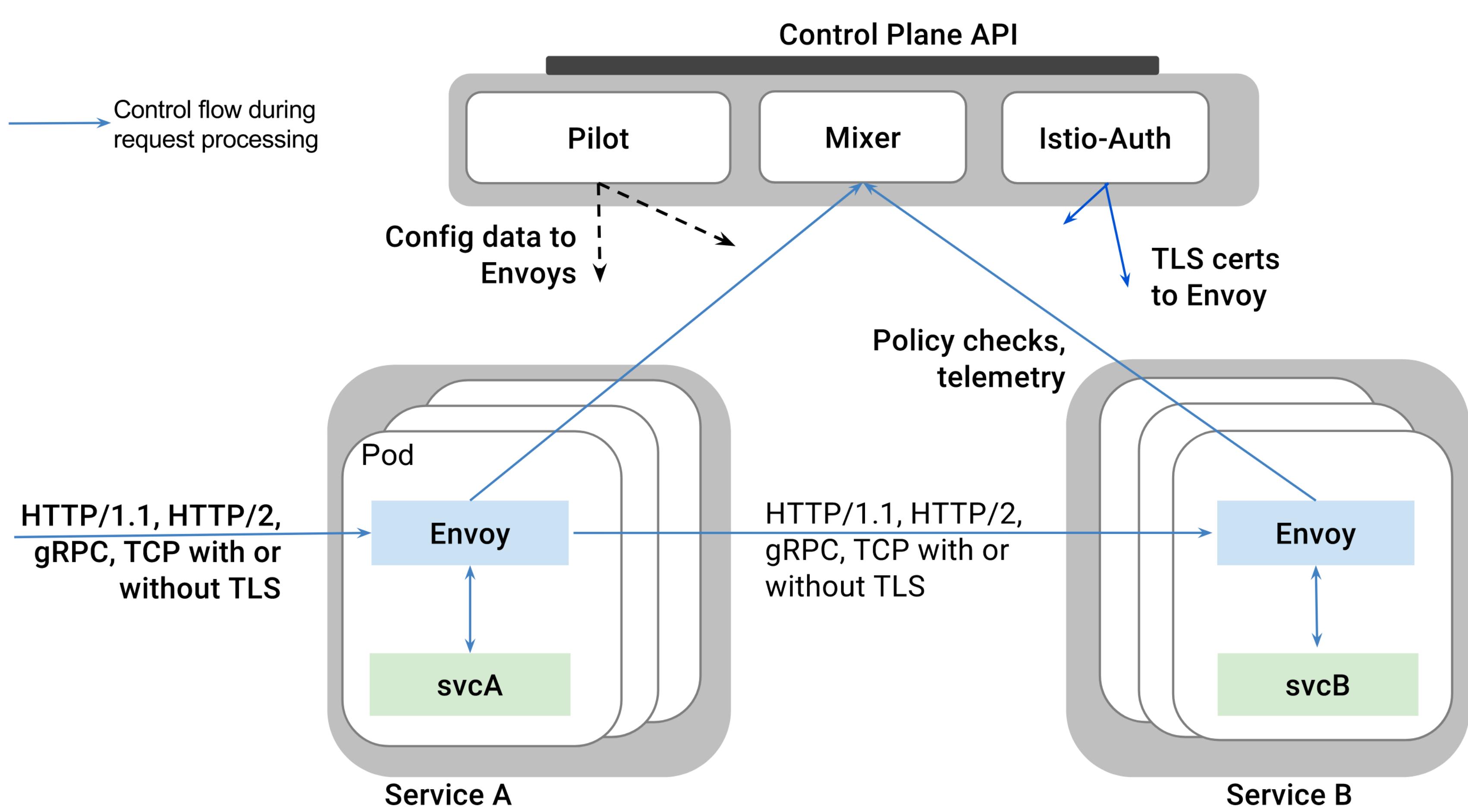


# Setting the sails with Istio



# Features

Traffic Management	Resiliency	Security	Observability
Request Routing	Timeouts	mTLS	Metrics
Load Balancing	Circuit Breaker	Access Control	Logs
Traffic Shifting	Health Checks (active, passive)	Workload Identity	Traces
Traffic Mirroring	Retries	RBAC	
Service Discovery	Rate Limiting		
Ingress, Egress	Delay & Fault Injection		



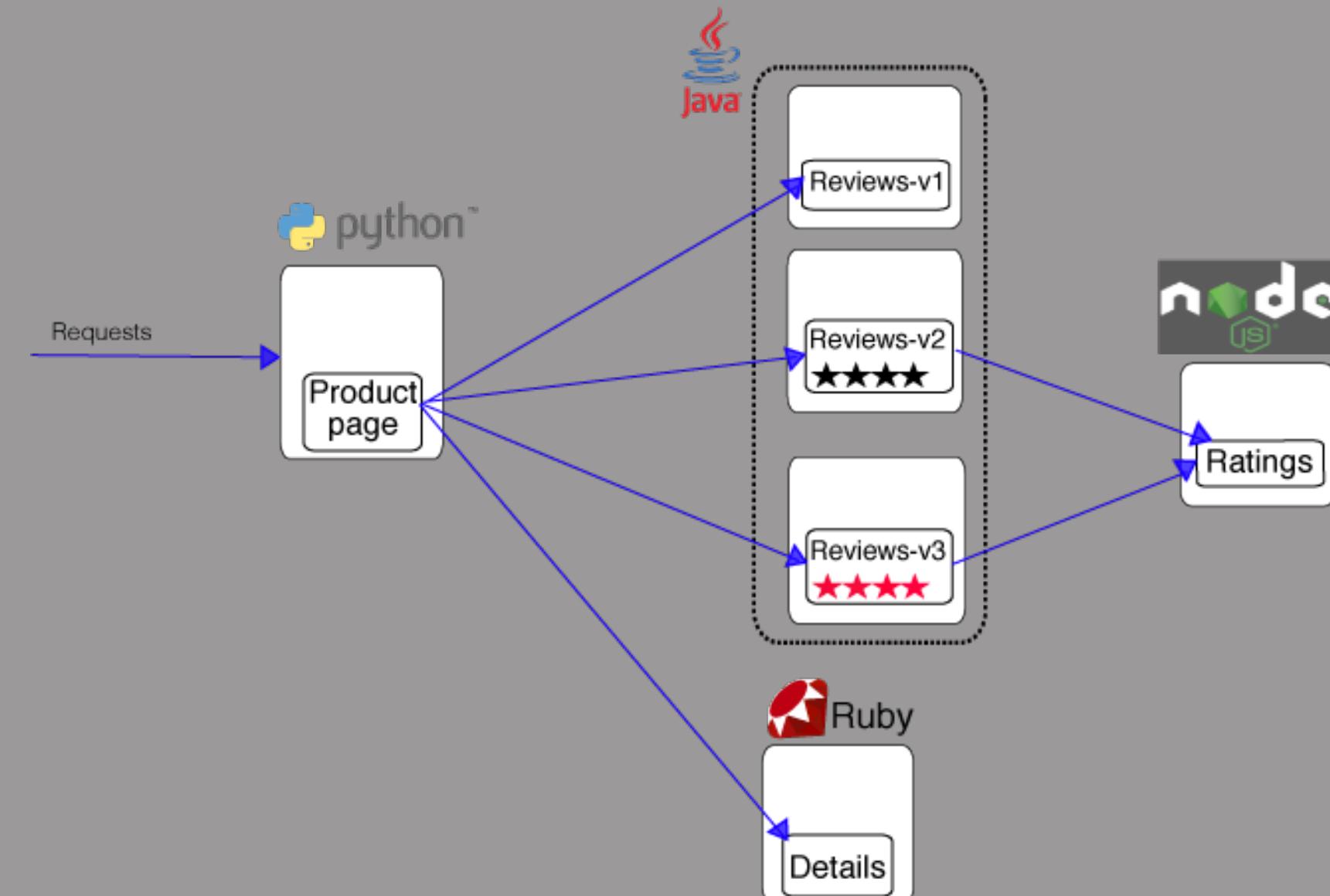
# Deploy Istio & Sample App

```
curl -L https://git.io/getLatestIstio | sh -
cd istio-
export PATH=$PWD/bin:$PATH
# deploy istio with mTLS enabled by default
kubectl apply -f install/kubernetes/istio-auth.yaml
# ... lengthy copy & paste code to deploy sidecar auto-deployment
# label default namespace to be auto-sidecarred
kubectl label namespace default istio-injection=enabled
# deploy and open sample application
kubectl apply -f istio-*/samples/bookinfo/kube/bookinfo.yaml
open http://localhost/productpage
```

# Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: gateway
  annotations:
    kubernetes.io/ingress.class: "istio"
spec:
  rules:
  - http:
    paths:
    - path: /productpage
      backend:
        serviceName: productpage
        servicePort: 9080
    - path: /login
      backend:
        serviceName: productpage
        servicePort: 9080
    - path: /logout
      backend:
        serviceName: productpage
        servicePort: 9080
    - path: /api/v1/products.*
      backend:
        serviceName: productpage
        servicePort: 9080
```

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



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

# Deploy Observability Add-Ons

```
#Prometheus
```

```
kubectl apply -f istio-*/install/kubernetes/addons/prometheus.yaml
kubectl expose deployment prometheus --name=prometheus-expose
--port=9090 --target-port=9090 --type=LoadBalancer -n=istio-system
```

```
#Grafana
```

```
kubectl apply -f istio-*/install/kubernetes/addons/grafana.yaml
kubectl expose deployment grafana --name=grafana-expose
--port=3000 --target-port=3000 --type=LoadBalancer -n=istio-system
```

```
#Jaeger
```

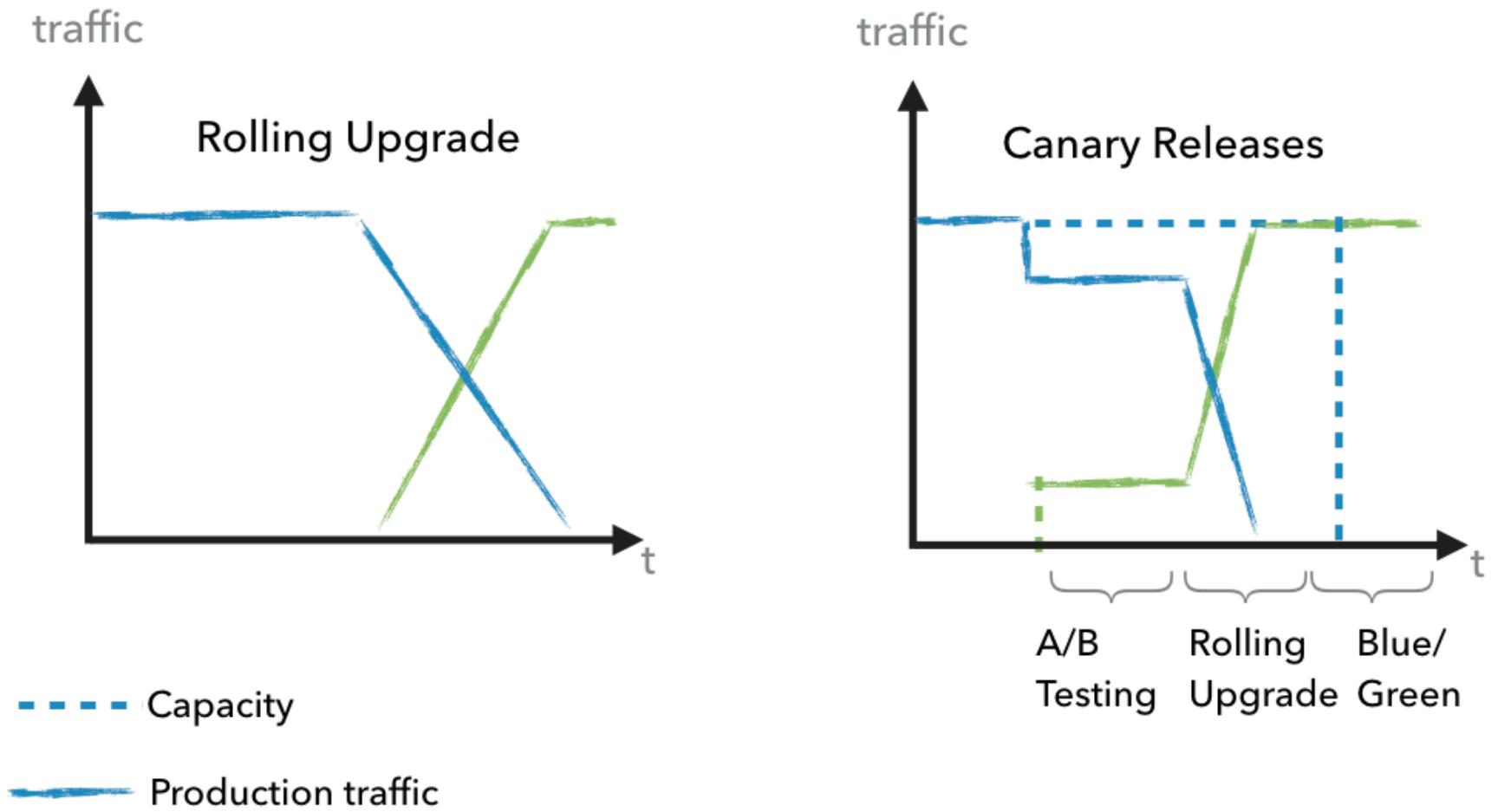
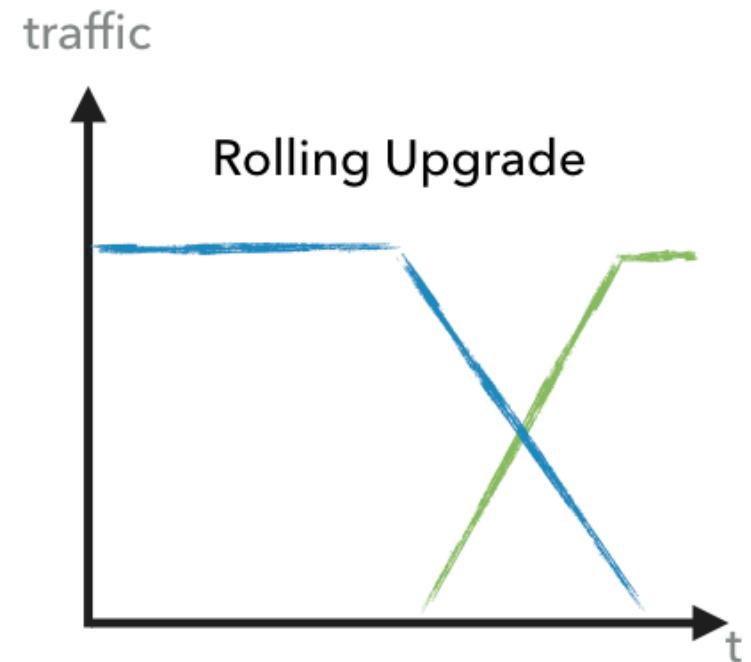
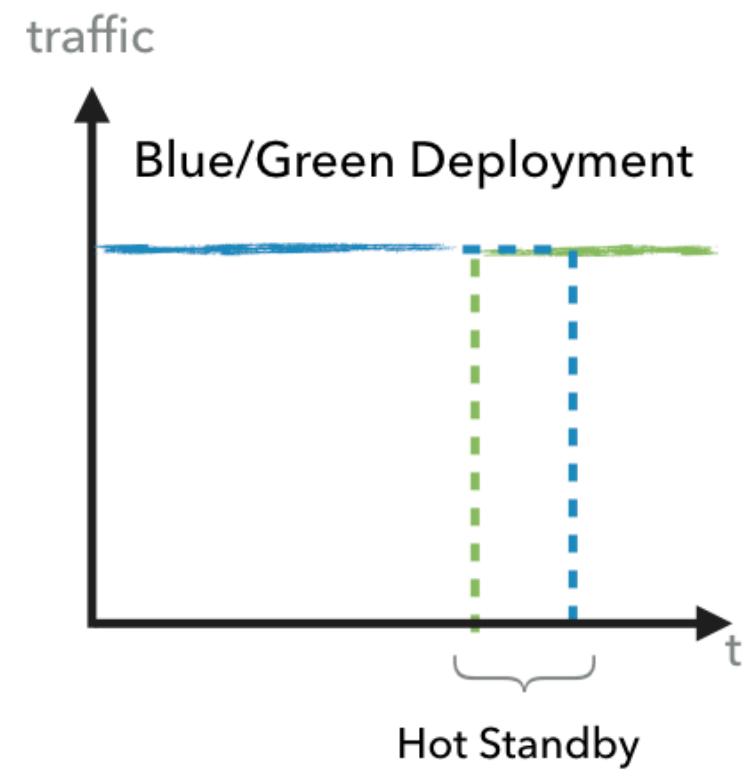
```
kubectl apply -n istio-system -f
https://raw.githubusercontent.com/jaegertracing/jaeger-kubernetes/
master/all-in-one/jaeger-all-in-one-template.yaml
kubectl expose deployment jaeger-deployment --name=jaeger-expose
--port=16686 --target-port=16686 --type=LoadBalancer -n=istio-system
```

```
#EFK
```

```
kubectl apply -f logging-stack.yaml
kubectl expose deployment kibana --name=kibana-expose
--port=5601 --target-port=5601 --type=LoadBalancer -n=logging
```

# Observe Services

```
# Logs  
istioctl create -f fluentd-istio.yaml  
# Metrics  
istioctl create -f telemetry.yaml
```



# Canary Releases: A/B Testing

```
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-test-v2
spec:
  destination:
    name: reviews
  precedence: 2
  match:
    request:
      headers:
        cookie:
          regex: "^(.*?; )?(user=jason)(; .*?)?$"
  route:
  - labels:
      version: v2
```

```
istioctl create -f route-rule-reviews-test-v2.yaml
```

# Canary Releases: Rolling Upgrade

```
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-default
spec:
  destination:
    name: reviews
  precedence: 1
  route:
    - labels:
        version: v1
        weight: 50
    - labels:
        version: v3
        weight: 50
```

```
istioctl create -f route-rule-reviews-50-v3.yaml
```

# Canary Releases: Blue/Green

```
apiVersion: config.istio.io/v1alpha2
kind: RouteRule
metadata:
  name: reviews-default
spec:
  destination:
    name: reviews
  precedence: 1
  route:
  - labels:
      version: v3
    weight: 100
```

```
istioctl replace -f route-rule-reviews-v3.yaml
```

# Security: Access Control

```
apiVersion: "config.istio.io/v1alpha2"
kind: denier
metadata:
  name: denyreviewsv3handler
spec:
  status:
    code: 7
    message: Not allowed
---
apiVersion: "config.istio.io/v1alpha2"
kind: checknothing
metadata:
  name: denyreviewsv3request
spec:
---
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: denyreviewsv3
spec:
  match: source.labels["layer"]=="inner" && destination.labels["layer"] == "outer"
  actions:
  - handler: denyreviewsv3handler.denier
    instances: [ denyreviewsv3request.checknothing ]
```

# Security: Egress

```
apiVersion: networking.istio.io/v1alpha3
kind: ExternalService
metadata:
  name: google-ext
spec:
  hosts:
  - www.google.com
  ports:
  - number: 443
    name: https
    protocol: http
---
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: google-ext
spec:
  name: www.google.com
  trafficPolicy:
    tls:
      mode: SIMPLE # initiates HTTPS when talking to www.google.com
```

# Resiliency: Circuit Breaker

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: httpbin
spec:
  name: httpbin
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
      http:
        http1MaxPendingRequests: 1
        maxRequestsPerConnection: 1
    outlierDetection:
      http:
        consecutiveErrors: 1
        interval: 1s
        baseEjectionTime: 3m
        maxEjectionPercent: 100
```

# 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)