@evanwijk

DOTNEXT

How to get a grip on your microservices system using a service-mesh

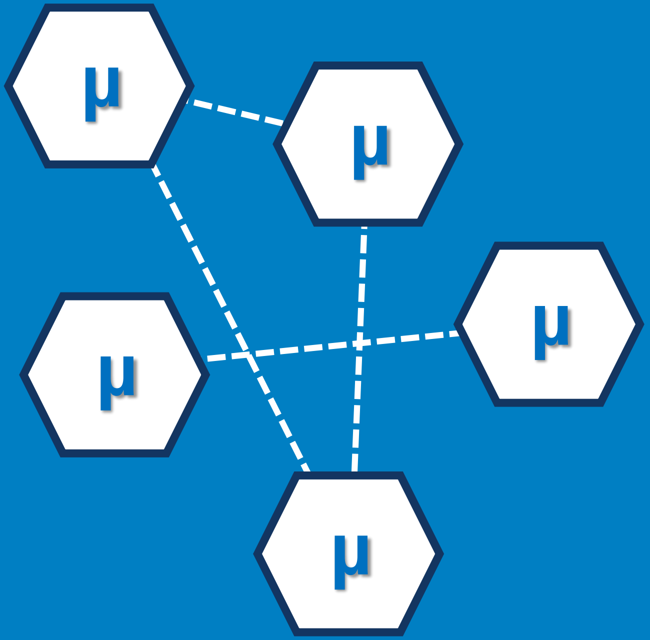
Edwin van Wijk
Principal Architect
infoSupport Solid Innovator

MVP
Microsoft® Most Valuable Professional

Contents

- Docker & Kubernetes
- Service mesh concept
- Service mesh implementation using Istio
- Istio features
  - Monitoring
  - Traffic routing (dark- / canary-releasing)
  - Testing resiliency through "chaos engineering"

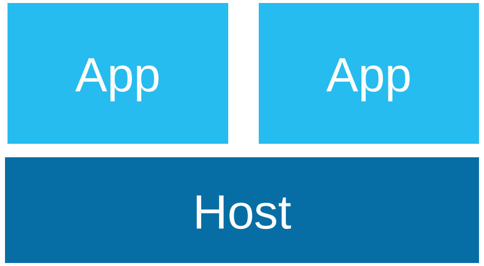
Introduction



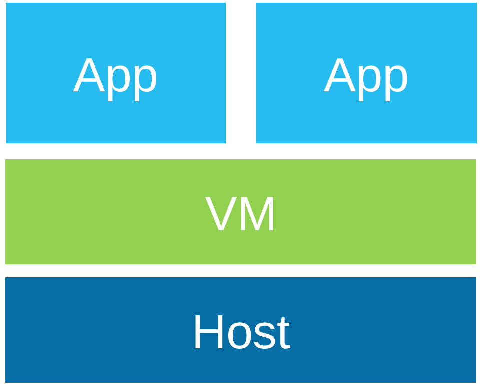
Microservices

- The high-level architecture style of choice nowadays
- Offers lots of advantages but …
- Adds complexity
  - More running parts (100+ !!), very distributed, polyglot, how to trace / monitor?
- We need a controlled way to run these services
  - DevOps, infra as code, containers

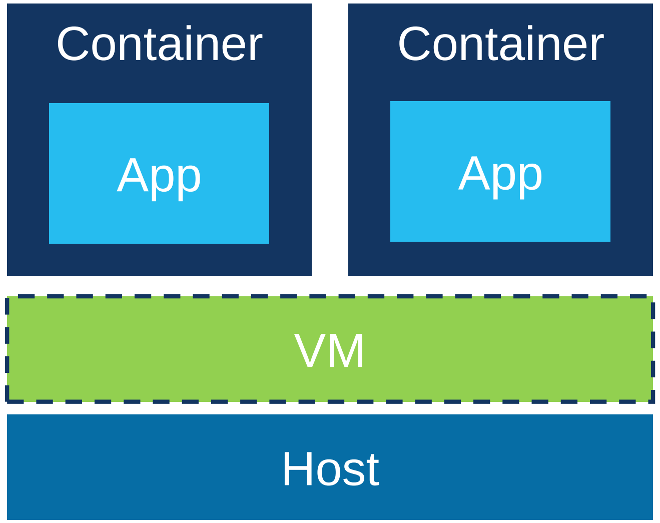
Evolution



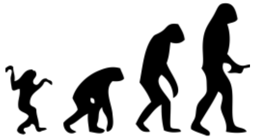
Evolution

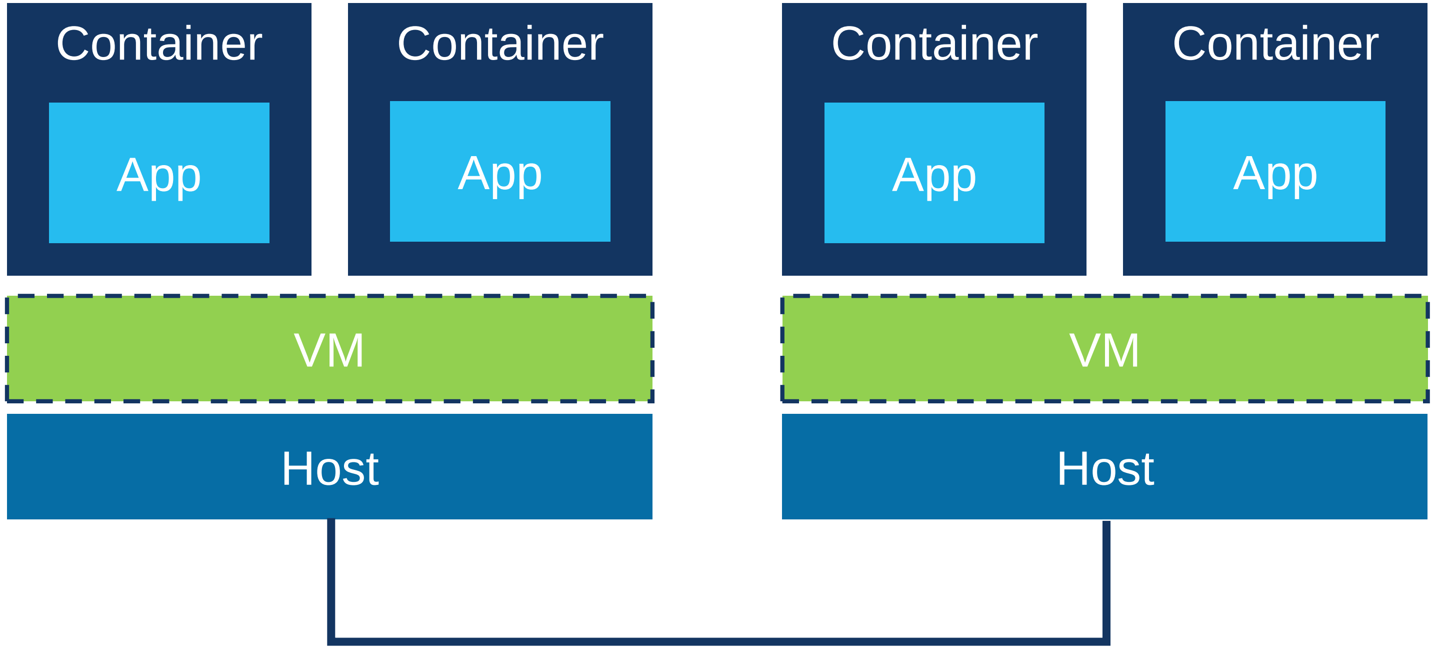


Evolution

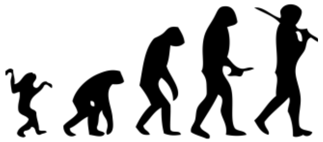


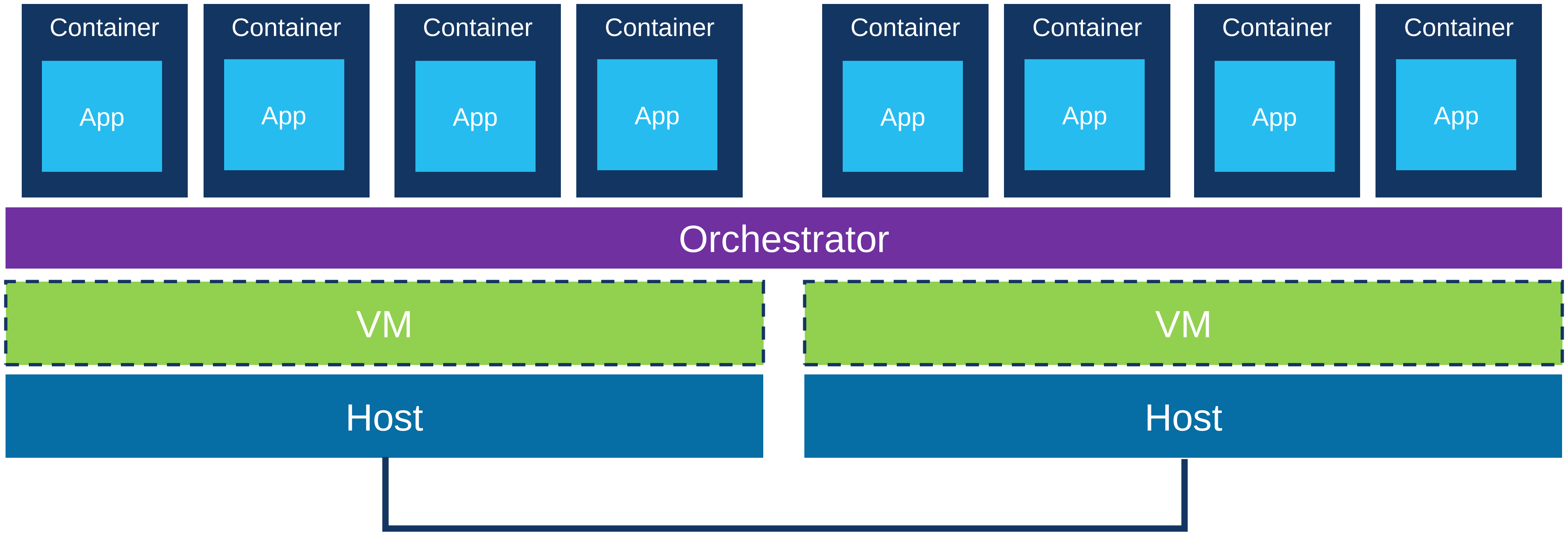
Evolution




Container
App
Container
App
Container
App
Container
App
VM
VM
Host
Host

Evolution





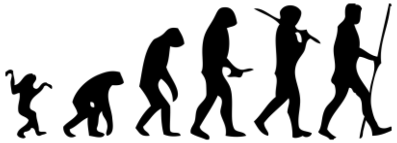
Container App | Container App | Container App | Container App | Container App | Container App | Container App | Container App

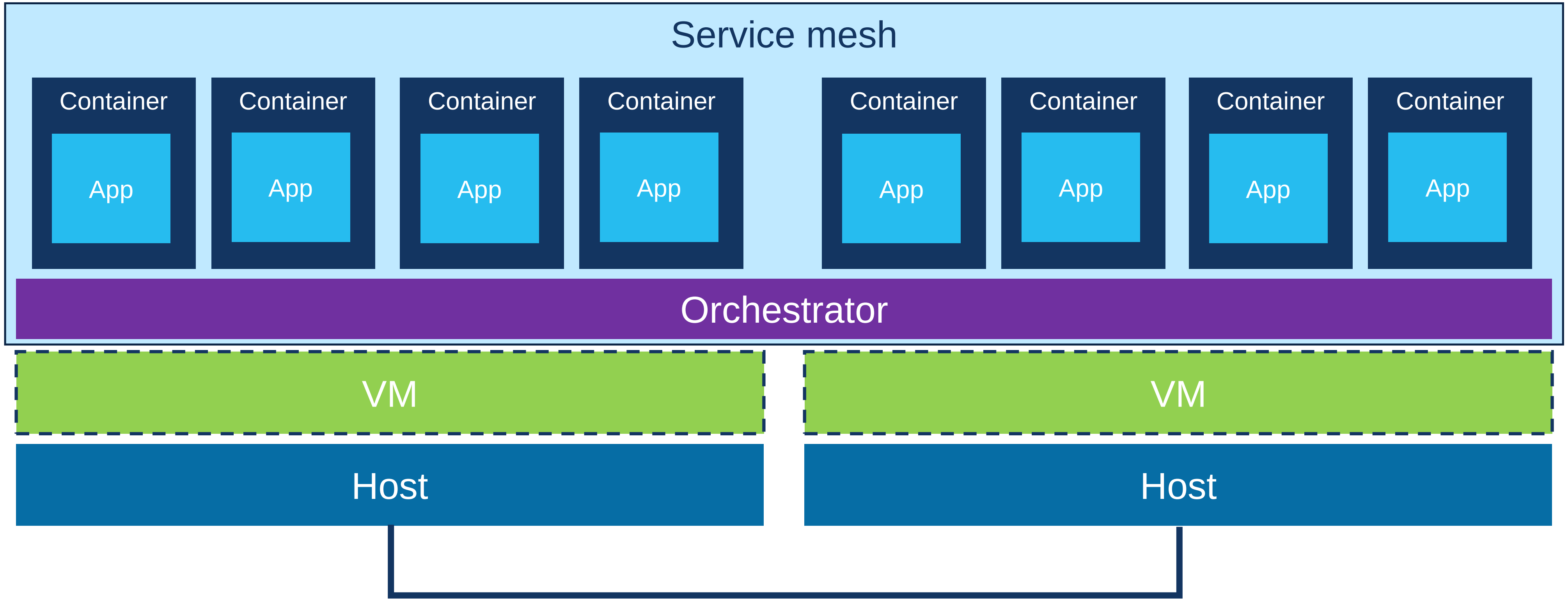
Orchestrator

VM | VM

Host | Host

Evolution

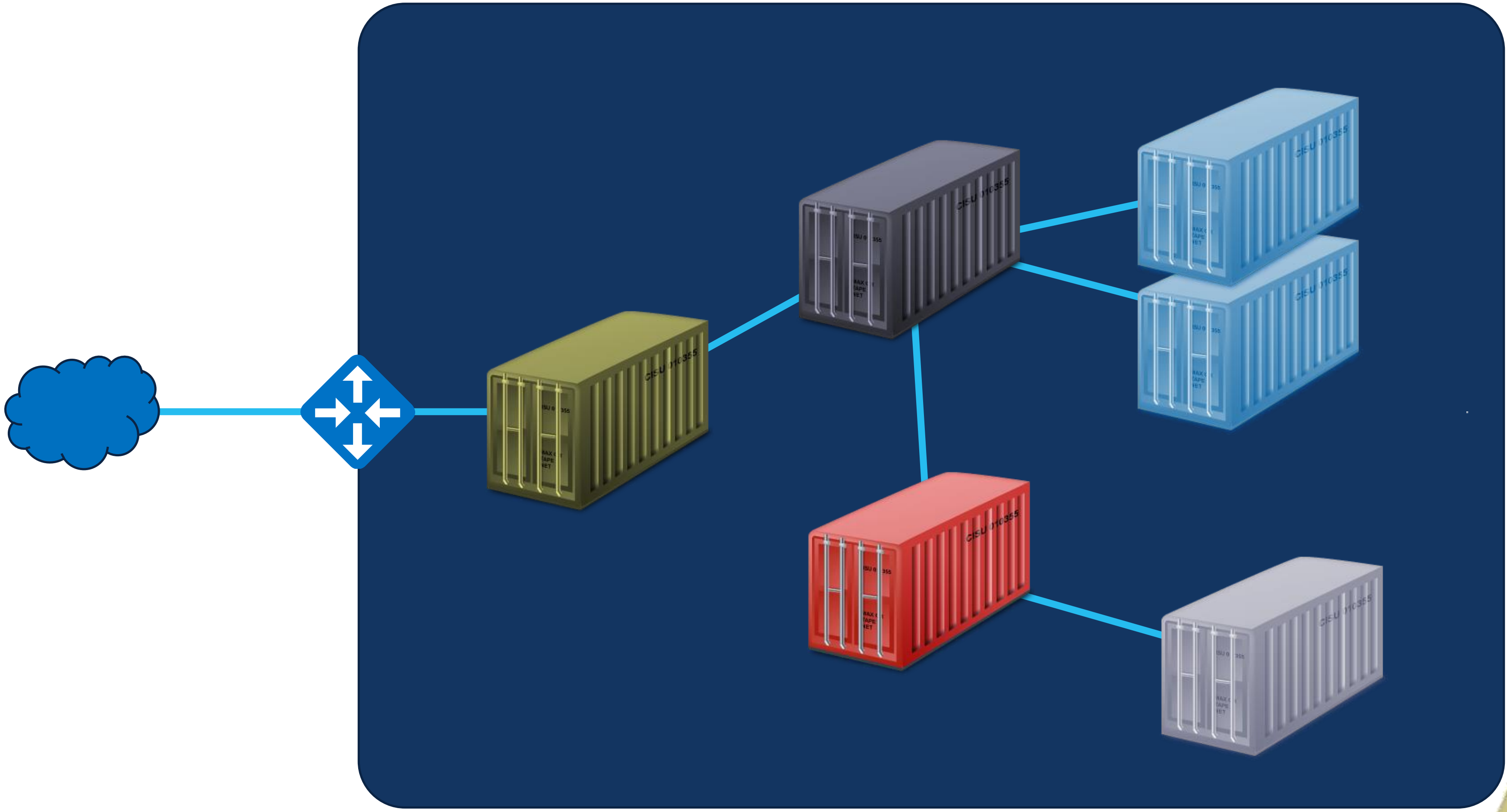



Service mesh
Container
App
Container
App
Container
App
Container
App
Container
App
Container
App
Container
App
Container
App
Orchestrator
VM
VM
Host
Host

Docker

Standardized software delivery

Docker



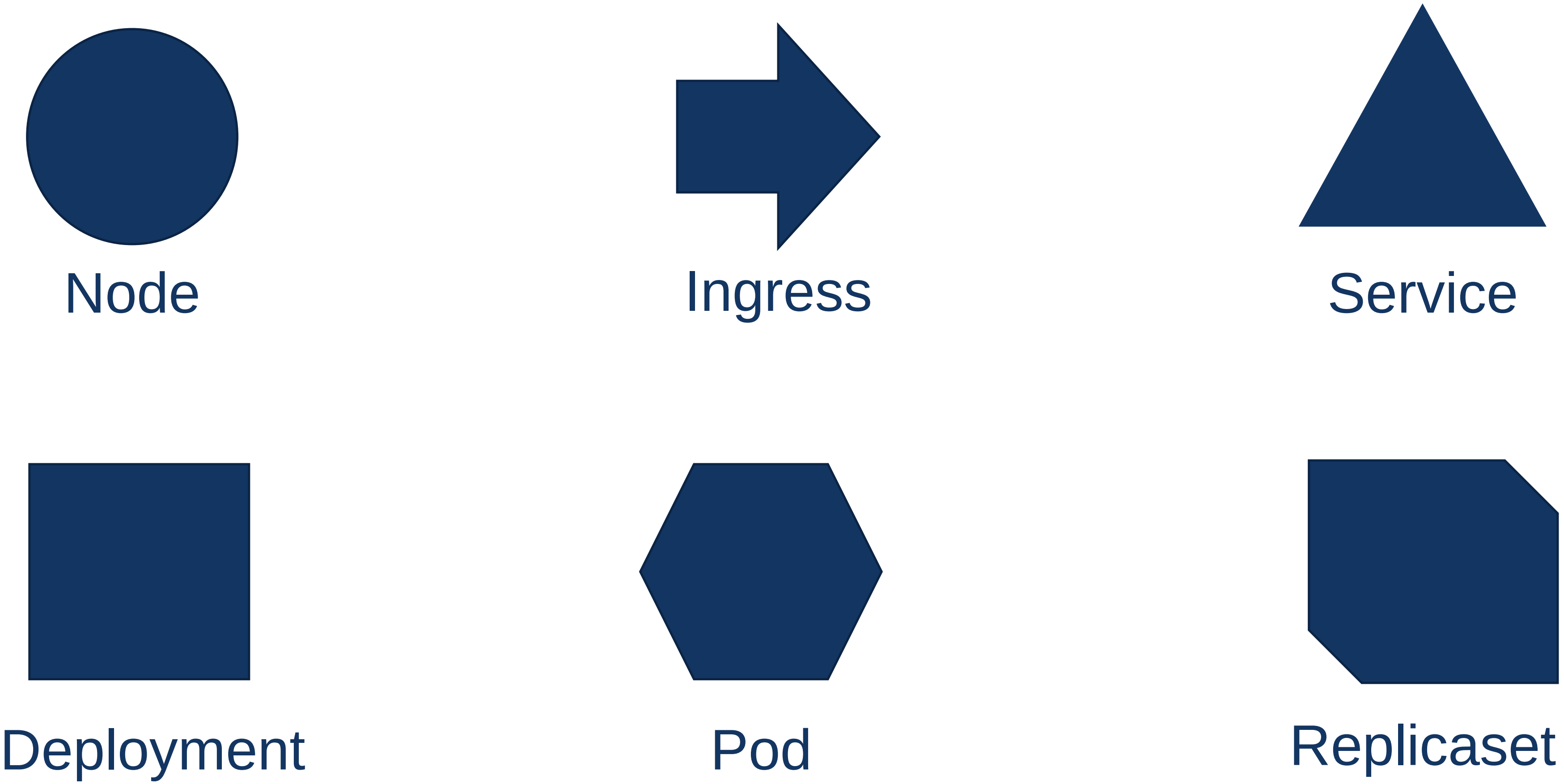
Kubernetes*

* a.k.a. K8S

Kubernetes

- Open-source orchestration tool for containers by Google
  - Automates deployment, scaling, and management
  - Handles computing, networking, and storage
- Runs just about anywhere
  - On prem, Cloud, SAAS (e.g. AKS)
- Developer version available in Docker Desktop
- Adopted as go to orchestrator by the community

Kubernetes objects*



Node

Ingress

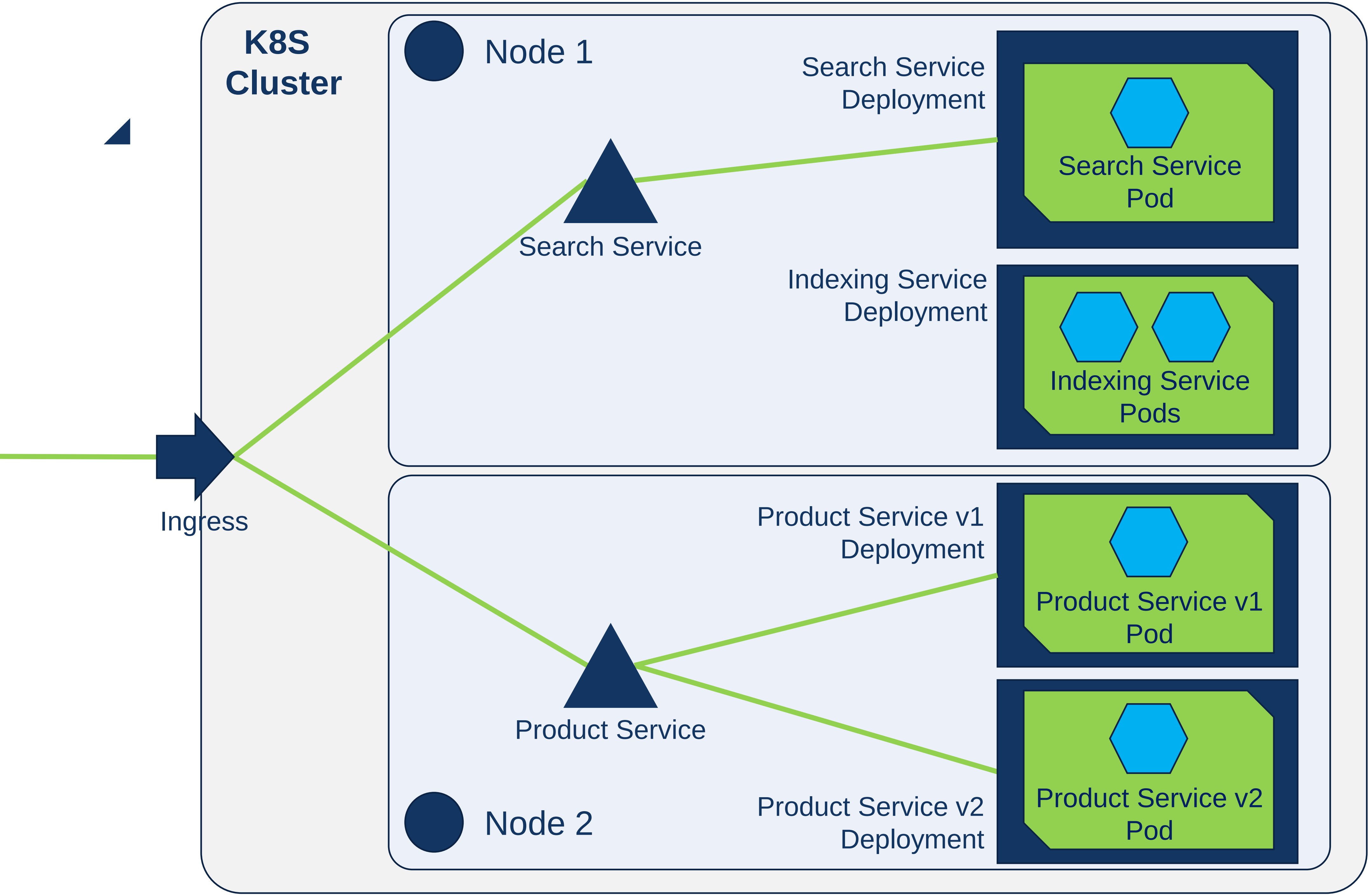
Service

Deployment

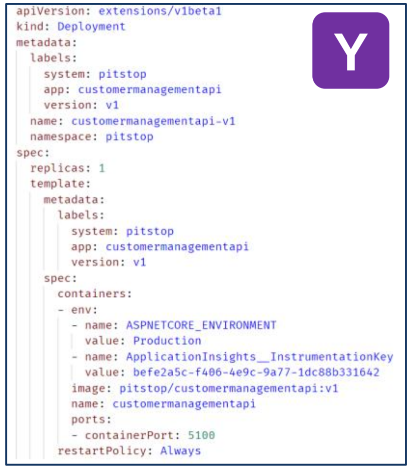
Pod

Replicaset

* Not complete, only relevant for the session


K8S Cluster
Node 1
Search Service Deployment
Search Service Pod
Search Service
Indexing Service Deployment
Indexing Service Pods
Ingress
Product Service v1 Deployment
Product Service v1 Pod
Product Service
Product Service v2 Pod
Node 2
Product Service v2 Deployment

Kubernetes orchestration

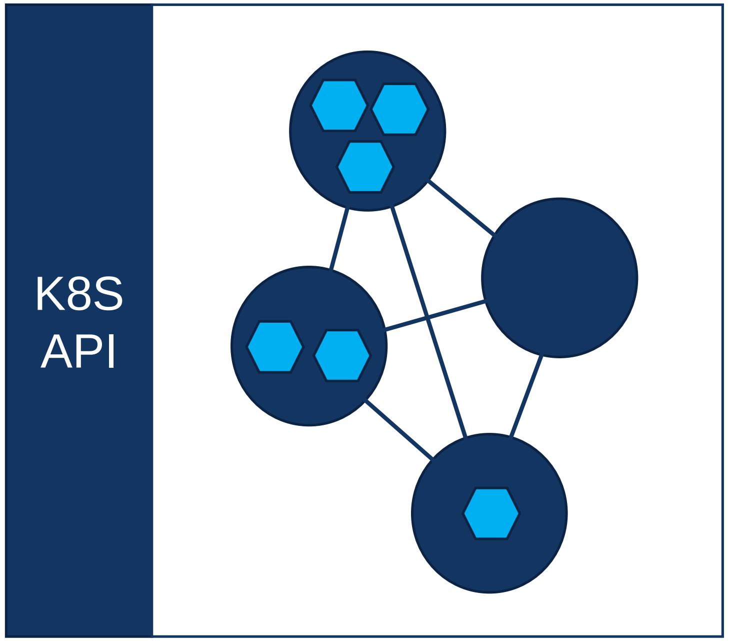


```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    system: pitstop
    app: customermanagementapi
    version: v1
  name: customermanagementapi-v1
  namespace: pitstop
spec:
  replicas: 1
  template:
    metadata:
      labels:
        system: pitstop
        app: customermanagementapi
        version: v1
    spec:
      containers:
      - env:
        - name: ASPNETCORE_ENVIRONMENT
          value: Production
        - name: ApplicationInsights__InstrumentationKey
          value: befe2a5c-f406-4e9c-9a77-1dc88b331642
        image: pitstop/customermanagementapi:v1
        name: customermanagementapi
        ports:
        - containerPort: 5100
      restartPolicy: Always
```

Configuration File(s)
(expected state)

Kubectl apply

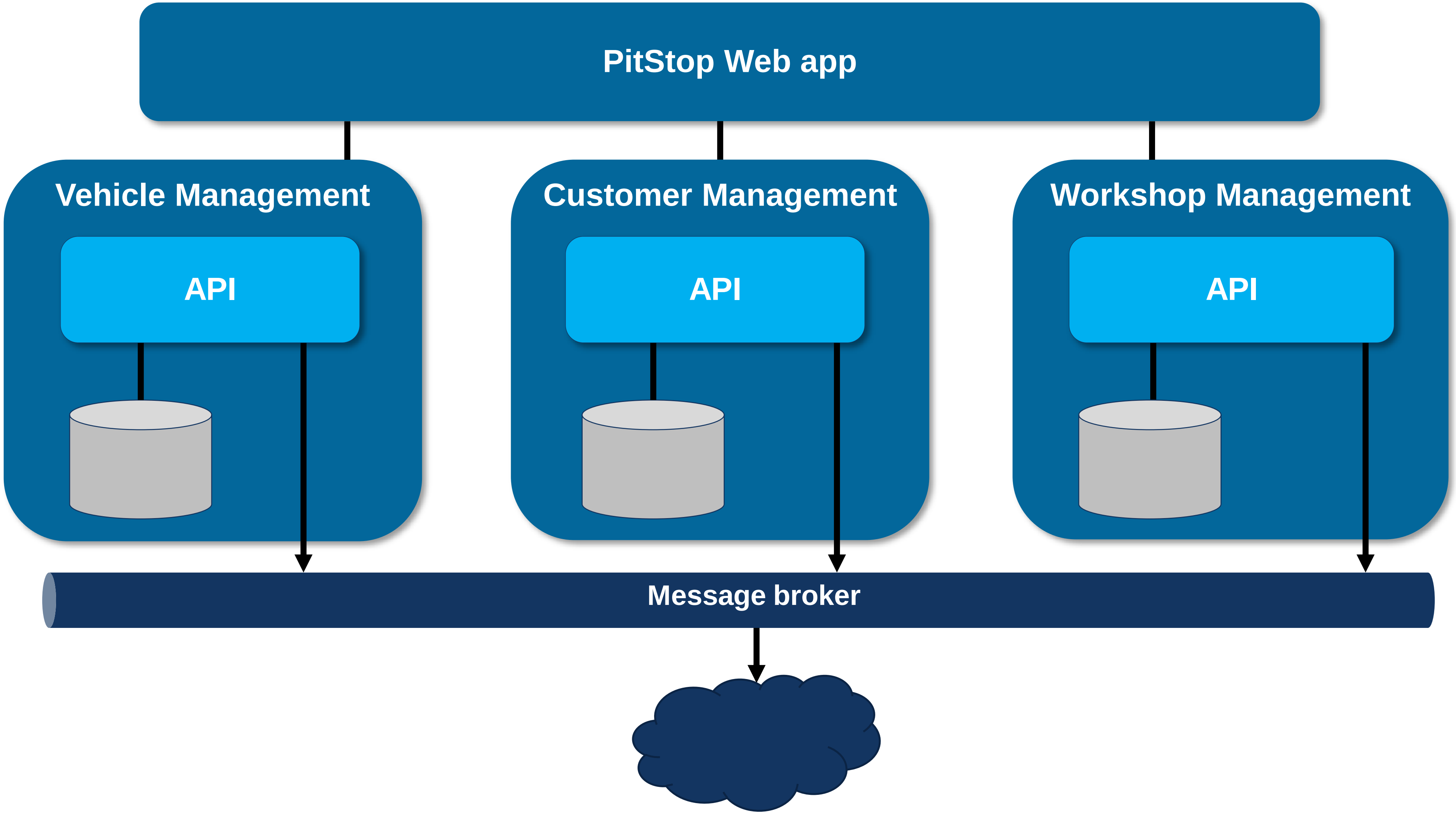
Kubernetes CLI



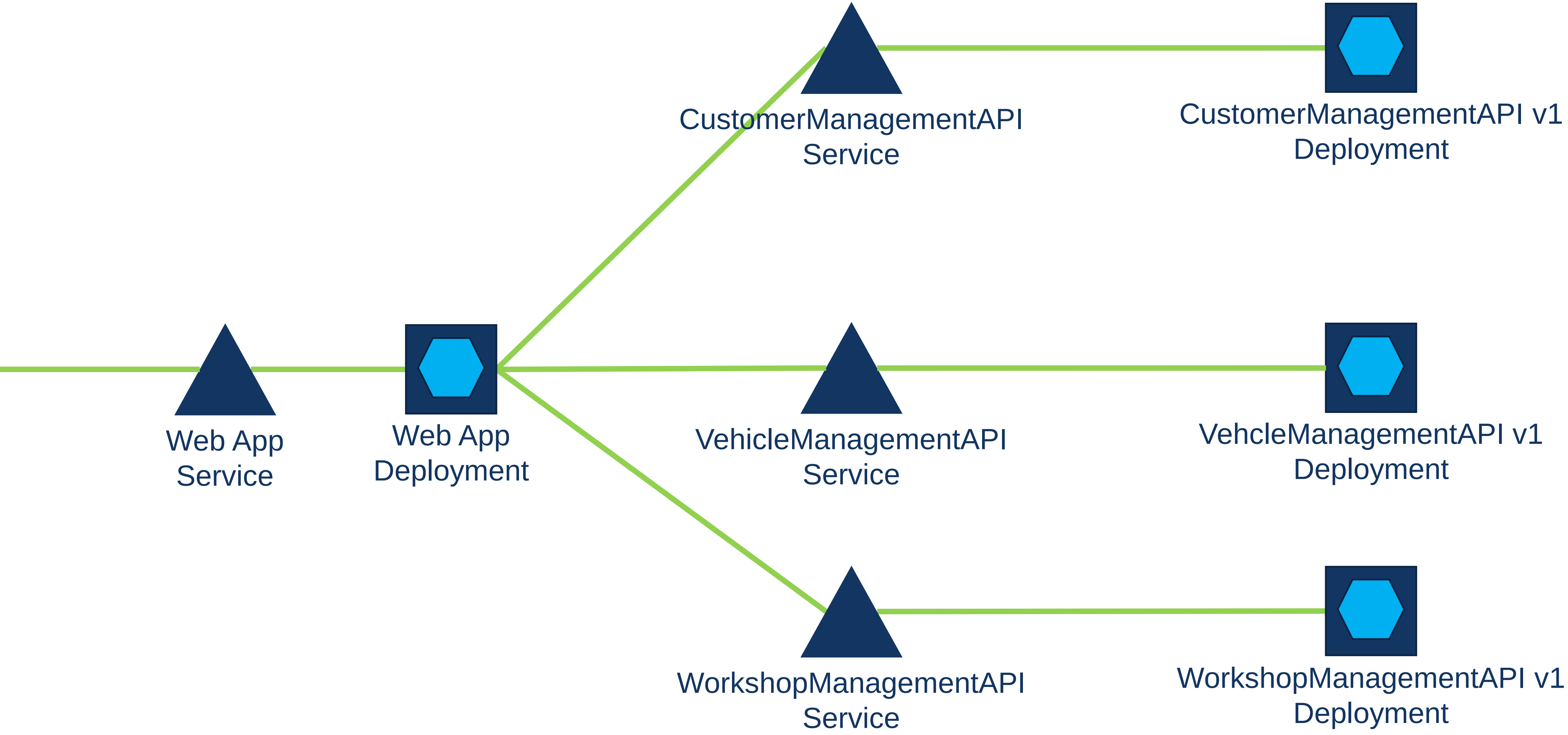
Kubernetes Cluster

Sample application

Solution-architecture overview



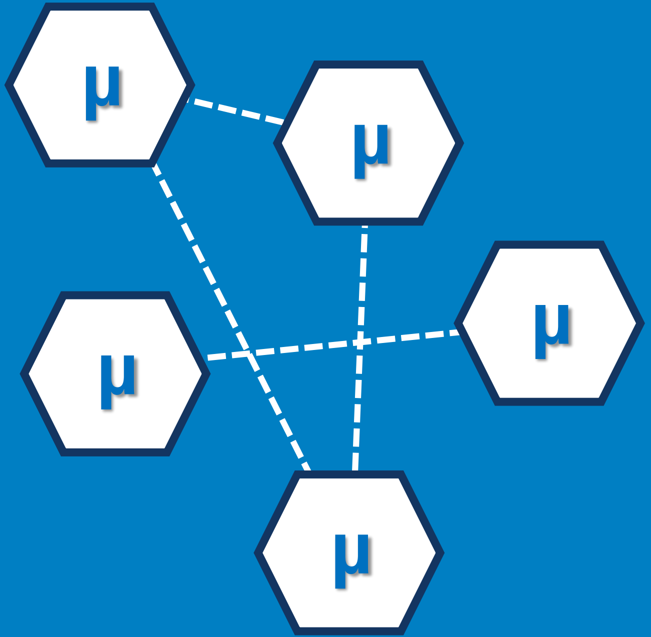
Kubernetes setup



F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
! 1
" 2
£ 3
$ 4 €
% 5
^ 6
& 7 7
* 8 8
(9 9
) 0 /
– -
Q
W
U 4
I 5
O 6
P *
{ [
A
S
D
F
G
H
K
@
B
N
<
,
Alt

demo

Service mes~~s~~h



Service mesh

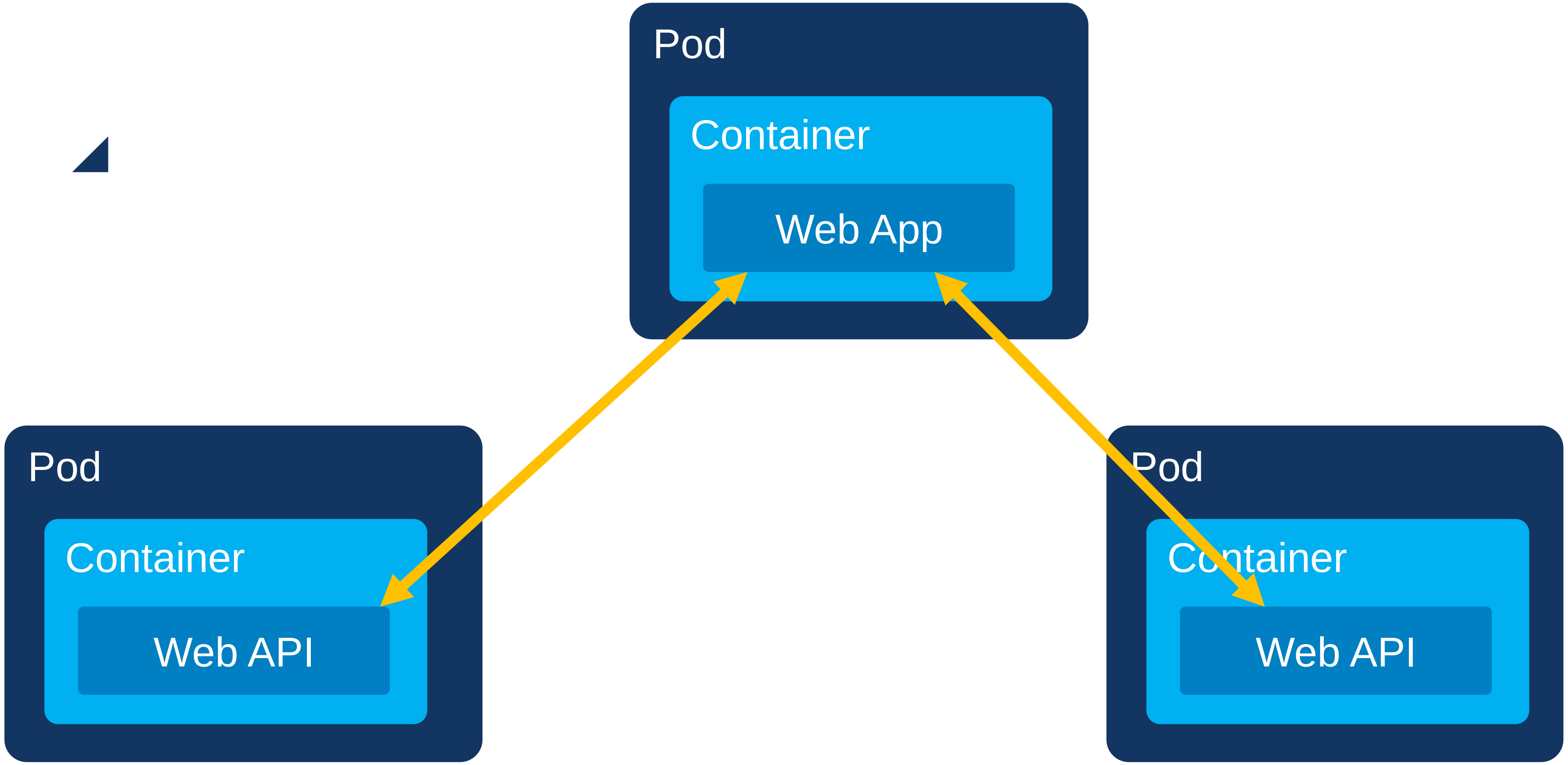
Provides a transparent and language-independent way to automate cross-cutting concerns

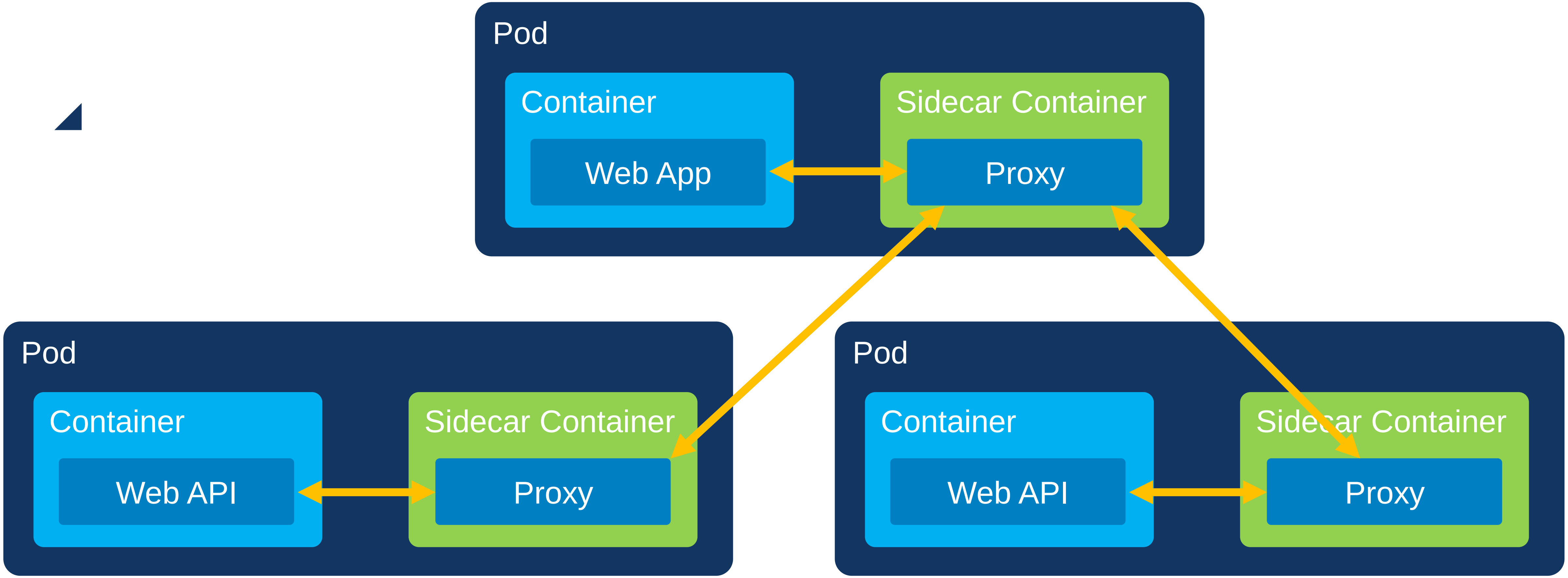
Intelligent routing, Monitoring, A/B Testing, Canary Releasing, Retries / Circuit-breakers, Security, …

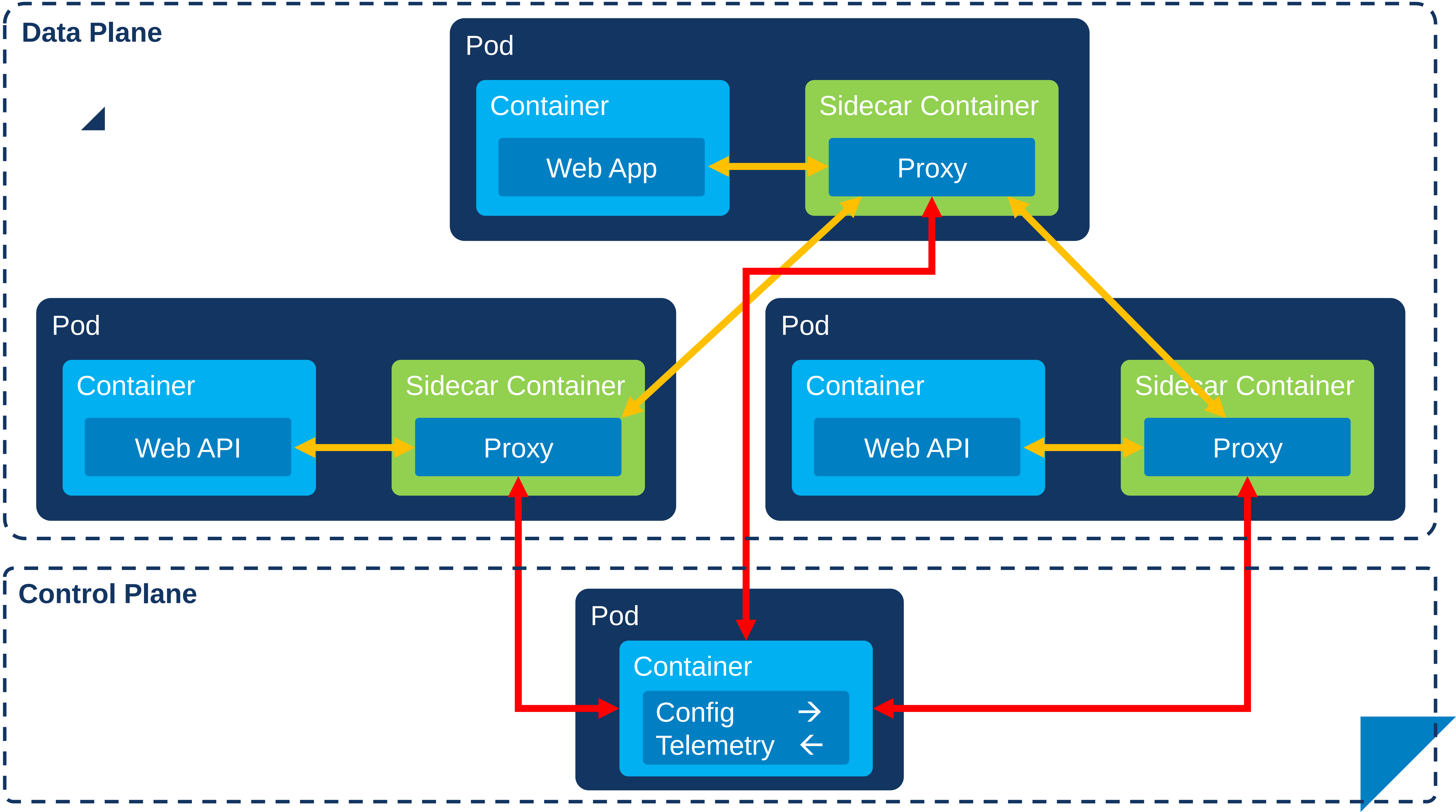
Service mesh

"Ambassador" Pattern

Implemented using
the "sidecar" pattern


Pod
Container
Web App
Pod
Container
Web API
Pod
Container
Web API


Pod
Container
Web App
Sidecar Container
Proxy
Pod
Container
Web API
Sidecar Container
Proxy
Pod
Container
Web API
Sidecar Container
Proxy



Data Plane

Pod

Container

Web App

Sidecar Container

Proxy

Pod

Container

Web API

Sidecar Container

Proxy

Pod

Container

Web API

Sidecar Container

Proxy

Control Plane

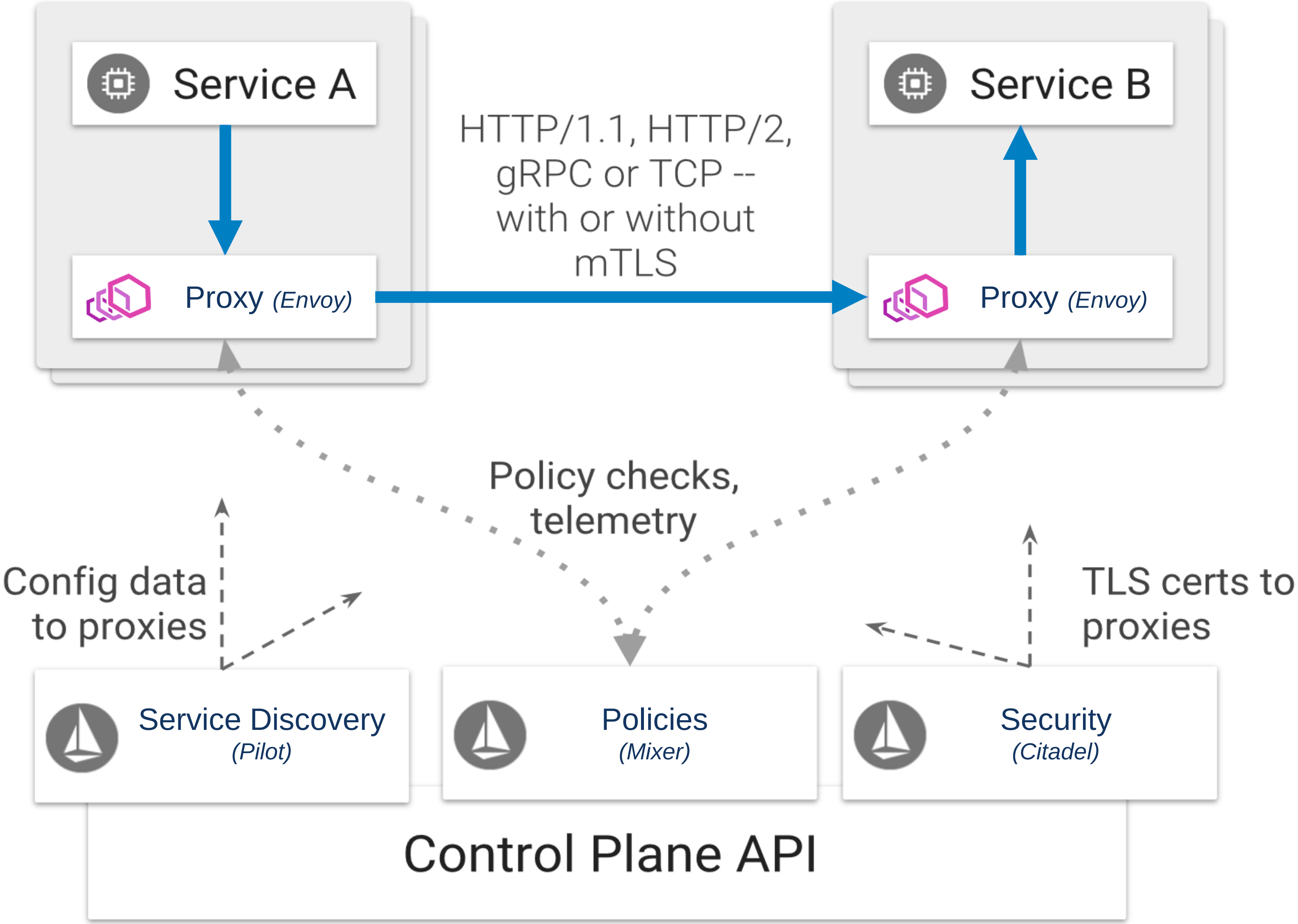
Pod

Container

Config →

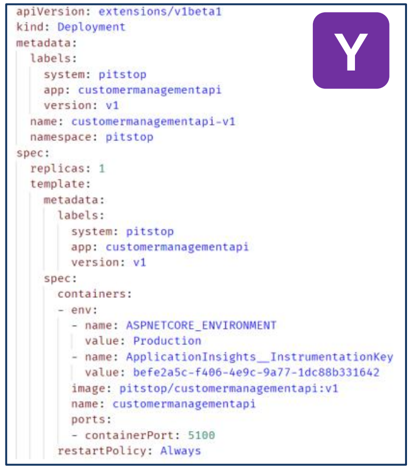
Telemetry ←

Istio



Service A
Service B
HTTP/1.1, HTTP/2, gRPC or TCP -- with or without mTLS
Proxy (Envoy)
Proxy (Envoy)
Policy checks, telemetry
Config data to proxies
TLS certs to proxies
Service Discovery (Pilot)
Policies (Mixer)
Security (Citadel)
Control Plane API

Istio proxy injection



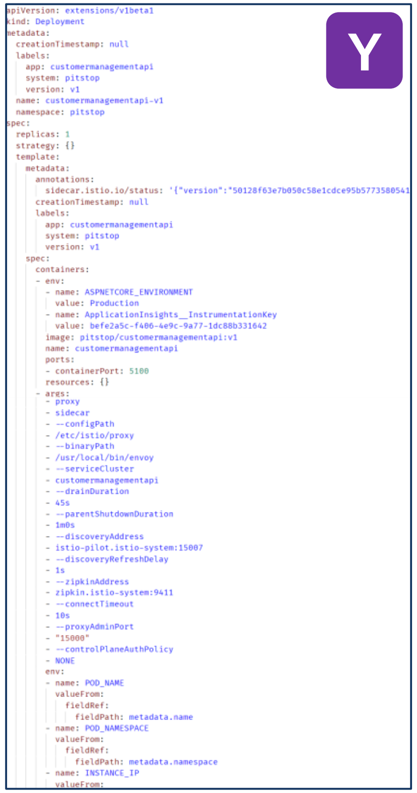
```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    system: pitstop
    app: customermanagementapi
    version: v1
  name: customermanagementapi-v1
  namespace: pitstop
spec:
  replicas: 1
  template:
    metadata:
      labels:
        system: pitstop
        app: customermanagementapi
        version: v1
    spec:
      containers:
      - env:
        - name: ASPNETCORE_ENVIRONMENT
          value: Production
        - name: ApplicationInsights__InstrumentationKey
          value: befe2a5c-f406-4e9c-9a77-1dc88b331642
        image: pitstop/customermanagementapi:v1
        name: customermanagementapi
        ports:
        - containerPort: 5100
      restartPolicy: Always
```

Configuration File

(expected state)

istioctl kube-inject

Kubernetes CLI



```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: customermanagementapi
    system: pitstop
    version: v1
  name: customermanagementapi-v1
  namespace: pitstop
spec:
  replicas: 1
  strategy: {}
  template:
    metadata:
      annotations:
        sidecar.istio.io/status: '{"version":"50128f63e7b050c58e1cdce95b5773580541
      creationTimestamp: null
      labels:
        app: customermanagementapi
        system: pitstop
        version: v1
    spec:
      containers:
      - env:
        - name: ASPNETCORE_ENVIRONMENT
          value: Production
        - name: ApplicationInsights__InstrumentationKey
          value: befe2a5c-f406-4e9c-9a77-1dc88b331642
        image: pitstop/customermanagementapi:v1
        name: customermanagementapi
        ports:
        - containerPort: 5100
        resources: {}
      - args:
        - proxy
        - sidecar
        - --configPath
        - /etc/istio/proxy
        - --binaryPath
        - /usr/local/bin/envoy
        - --serviceCluster
        - customermanagementapi
        - --drainDuration
        - 45s
        - --parentShutdownDuration
        - 1m0s
        - --discoveryAddress
        - istio-pilot.istio-system:15007
        - --discoveryRefreshDelay
        - 1s
        - --zipkinAddress
        - zipkin.istio-system:9411
        - --connectTimeout
        - 10s
        - --proxyAdminPort
        - "15000"
        - --controlPlaneAuthPolicy
        - NONE
        env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: INSTANCE_IP
          valueFrom:
```

Configuration File

(expected state)

Capabilities

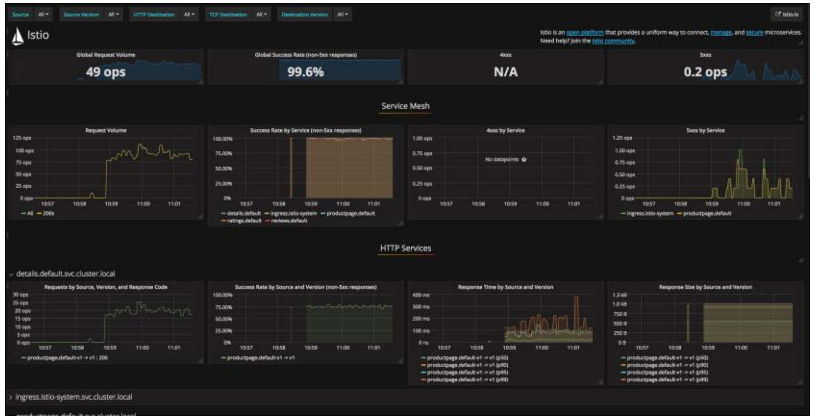
Observability

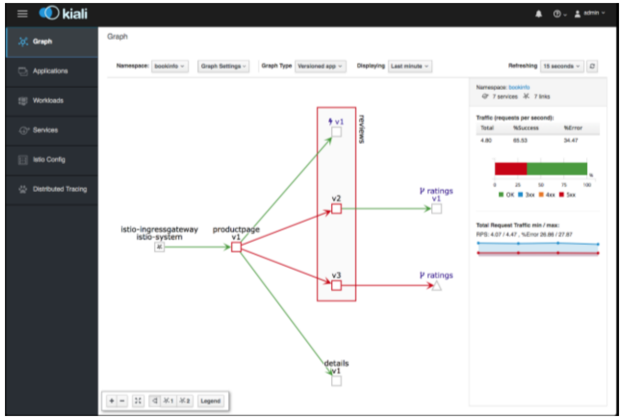
Logging / Tracing / Monitoring

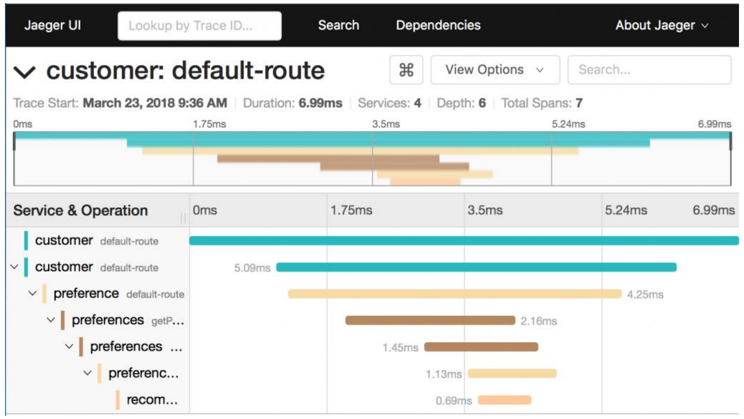
Grafana

kiali

JAEGER


Source All
Source Version All
HTTP Destination All
TCP Destination All
Destination Version All
istio.io
Istio
Istio is an open platform that provides a uniform way to connect, manage, and secure microservices.
Need help? Join the Istio community.
Global Request Volume
49 ops
Global Success Rate (non-5xx responses)
99.6%
4xxs
N/A
5xxs
0.2 ops
Service Mesh
Request Volume
Success Rate by Service (non-5xx responses)
4xxs by Service
No datapoints
5xxs by Service
All
200s
details.default
ingress.istio-system
productpage.default
ratings.default
reviews.default
HTTP Services
details.default.svc.cluster.local
Requests by Source, Version, and Response Code
Success Rate by Source and Version (non-5xx responses)
Response Time by Source and Version
Response Size by Source and Version
productpage.default-v1 -> v1 : 200
productpage.default-v1 -> v1
productpage.default-v1 -> v1 (p50)
productpage.default-v1 -> v1 (p90)
productpage.default-v1 -> v1 (p95)
productpage.default-v1 -> v1 (p99)
ingress.istio-system.svc.cluster.local

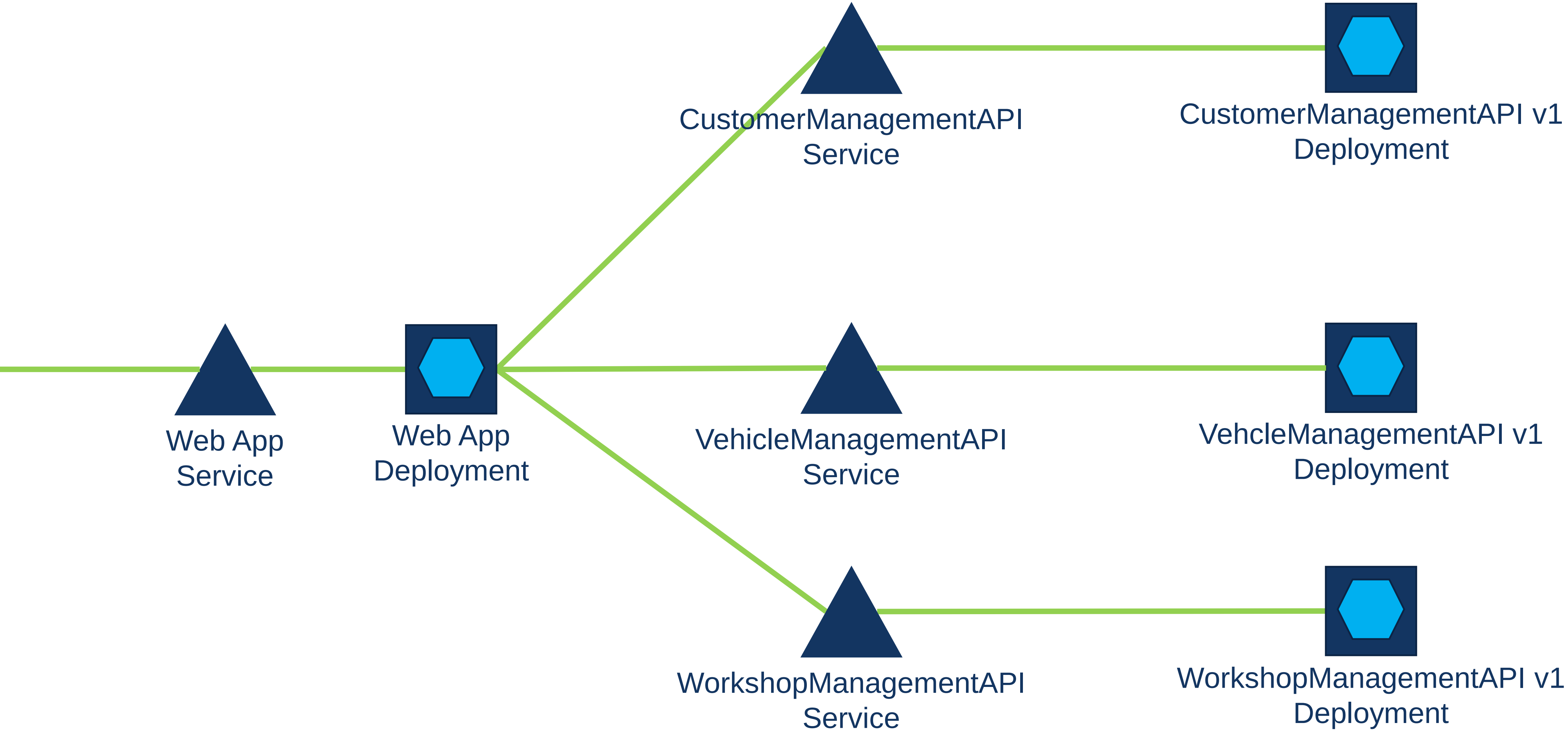




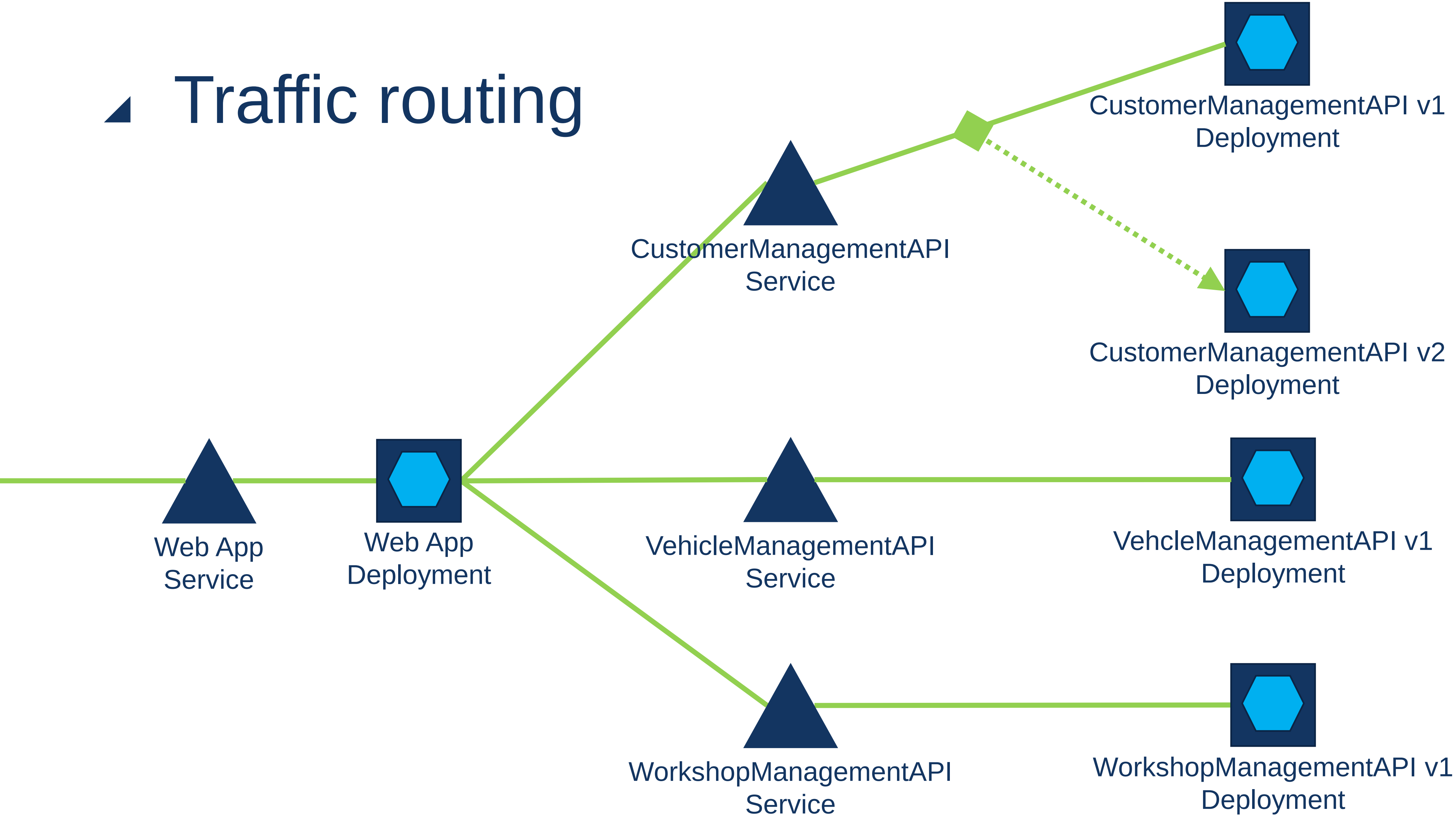
Prometheus

Traffic Routing

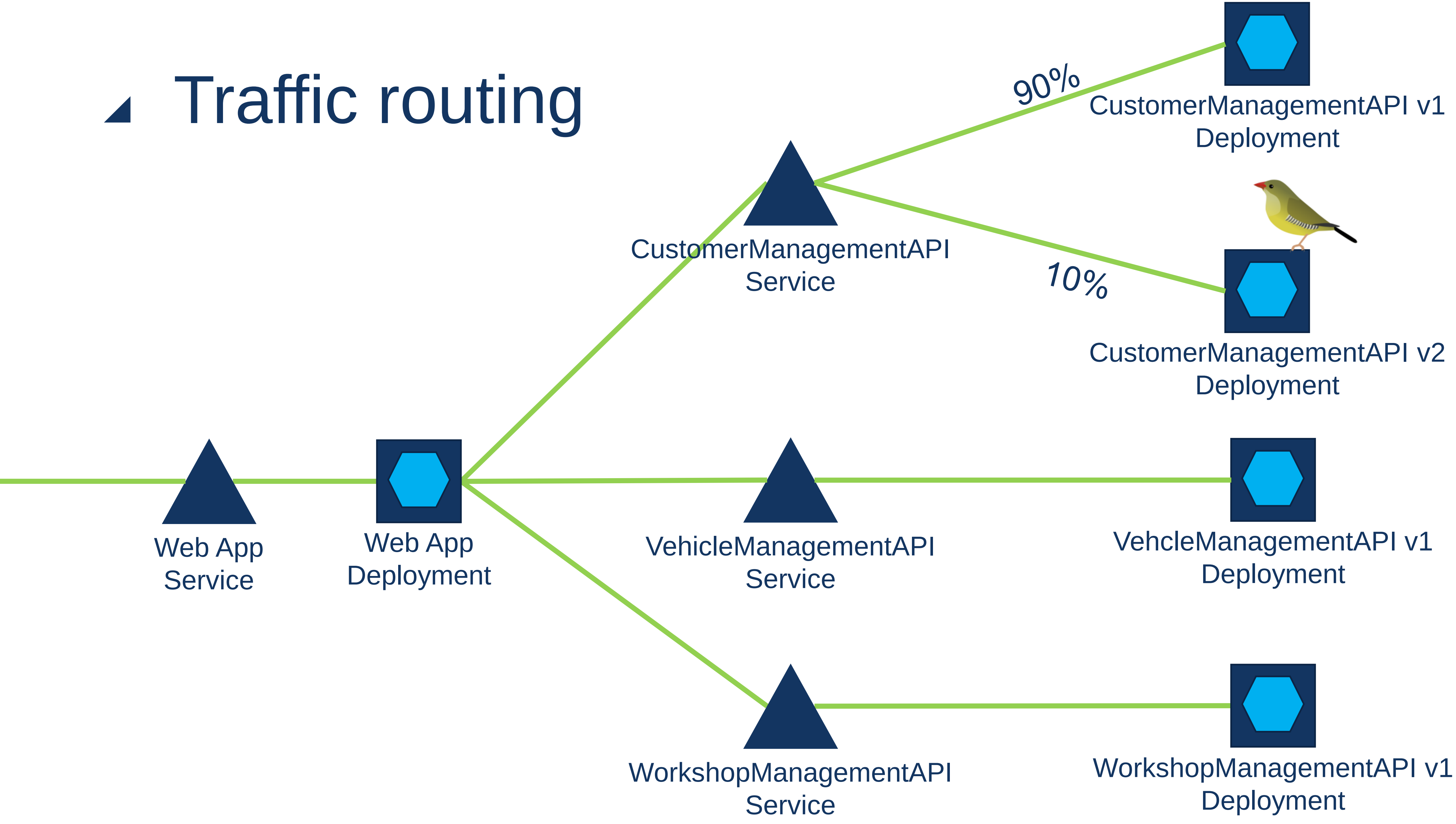
Traffic routing



Traffic routing

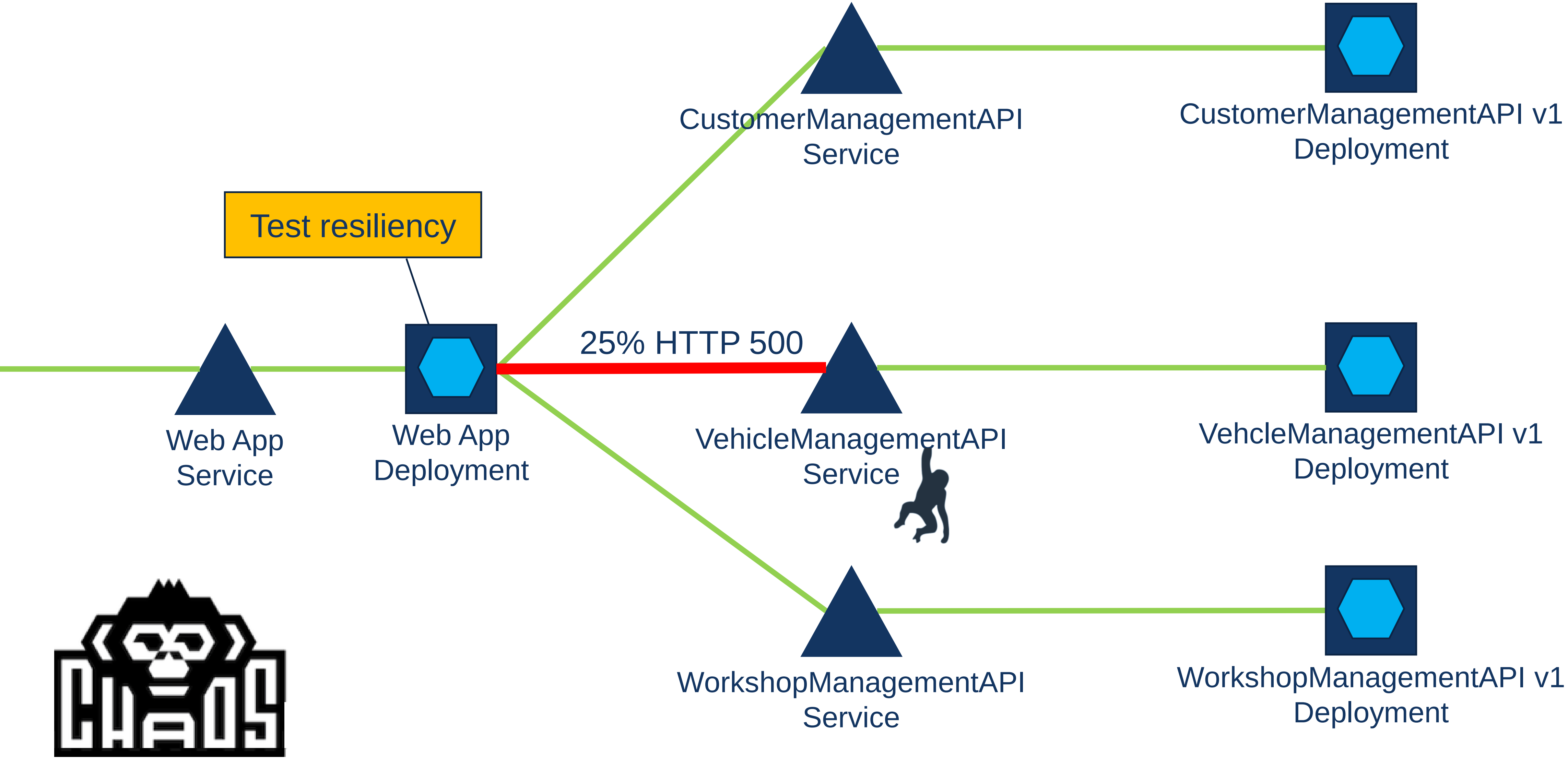


Traffic routing



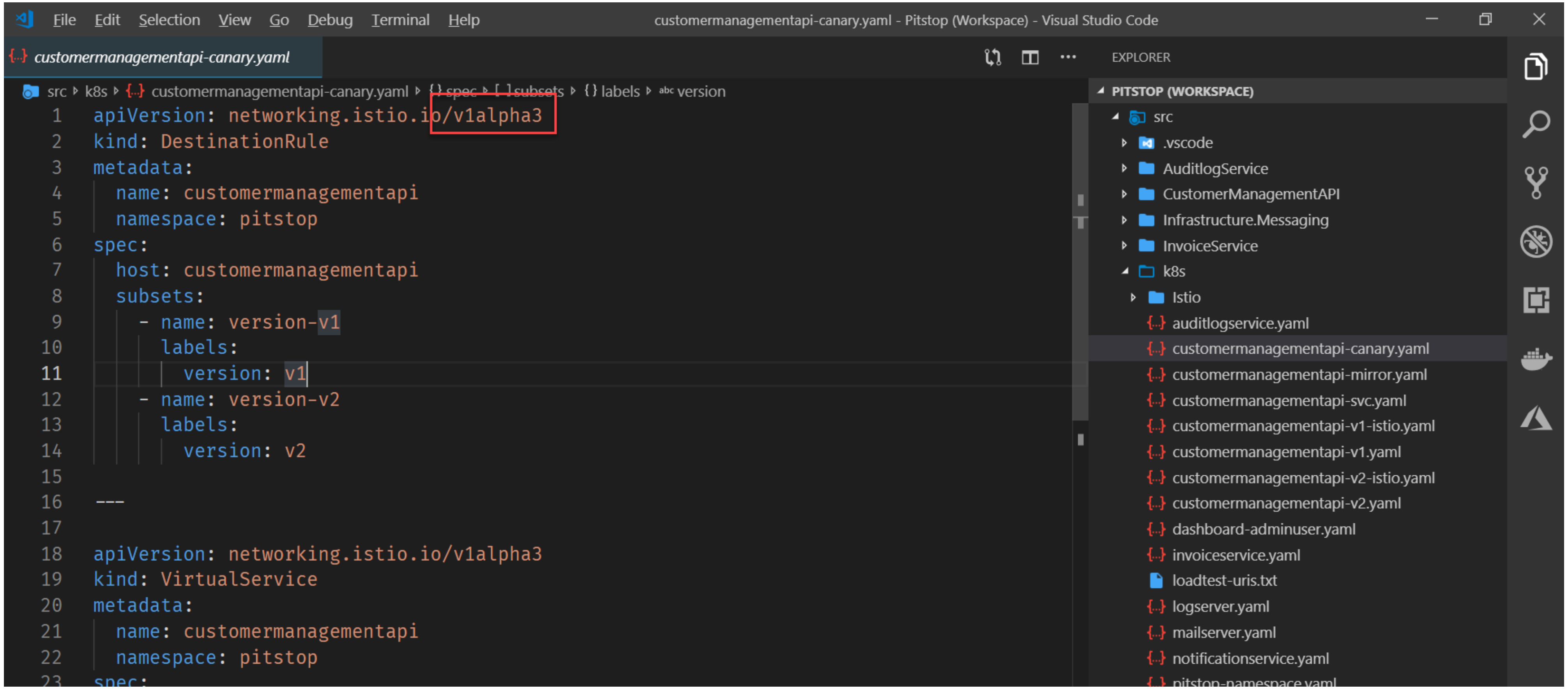
Chaos engineering

Chaos engineering


CustomerManagementAPI Service
CustomerManagementAPI v1 Deployment
Test resiliency
25% HTTP 500
Web App Service
Web App Deployment
VehicleManagementAPI Service
VehcleManagementAPI v1 Deployment
WorkshopManagementAPI Service
WorkshopManagementAPI v1 Deployment
CHAOS

Be aware!

- Istio is still in active development



File
Edit
Selection
View
Go
Debug
Terminal
Help
customermanagementapi-canary.yaml - Pitstop (Workspace) - Visual Studio Code
customermanagementapi-canary.yaml
src ▸ k8s ▸ customermanagementapi-canary.yaml ▸ { } spec ▸ [ ] subsets ▸ { } labels ▸ abc version
EXPLORER
PITSTOP (WORKSPACE)
src
.vscode
AuditlogService
CustomerManagementAPI
Infrastructure.Messaging
InvoiceService
k8s
Istio
auditlogservice.yaml
customermanagementapi-canary.yaml
customermanagementapi-mirror.yaml
customermanagementapi-svc.yaml
customermanagementapi-v1-istio.yaml
customermanagementapi-v1.yaml
customermanagementapi-v2-istio.yaml
customermanagementapi-v2.yaml
dashboard-adminuser.yaml
invoiceservice.yaml
loadtest-uris.txt
logserver.yaml
mailserver.yaml
notificationservice.yaml
pitstop-namespace.yaml

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: customermanagementapi
  namespace: pitstop
spec:
  host: customermanagementapi
  subsets:
    - name: version-v1
      labels:
        version: v1
    - name: version-v2
      labels:
        version: v2

---

apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: customermanagementapi
  namespace: pitstop
spec:
```

Resources

https://kubernetes.io

https://istio.io

https://github.com/edwinvw/pitstop

Thank you!

Edwin van Wijk
Principal Architect

Info Support
Solid Innovator

MVP
Microsoft®
Most Valuable
Professional