You make **possible**

Service Mesh for Network Engineers

What does what where and when

Roger Dickinson
@DCgubbins

DEVNET1697

Barcelona | January 27-31, 2020

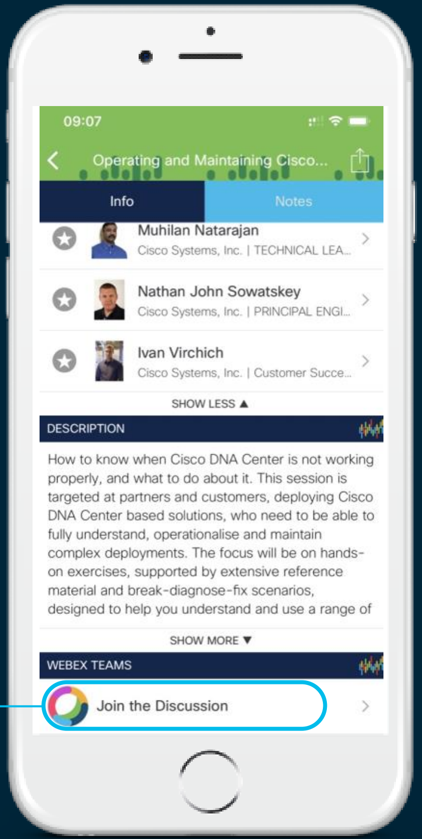
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat
with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



09:07

Operating and Maintaining Cisco...

Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...

Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...

Ivan Virchich
Cisco Systems, Inc. | Customer Succe...

DESCRIPTION

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

WEBEX TEAMS

About Me

- Developer Advocate
- 20 years at Cisco mainly doing DC and Cloud
- Draws cartoons
- Graduate of the school of hard knocks of routing and switching

Service Mesh for Network Engineers

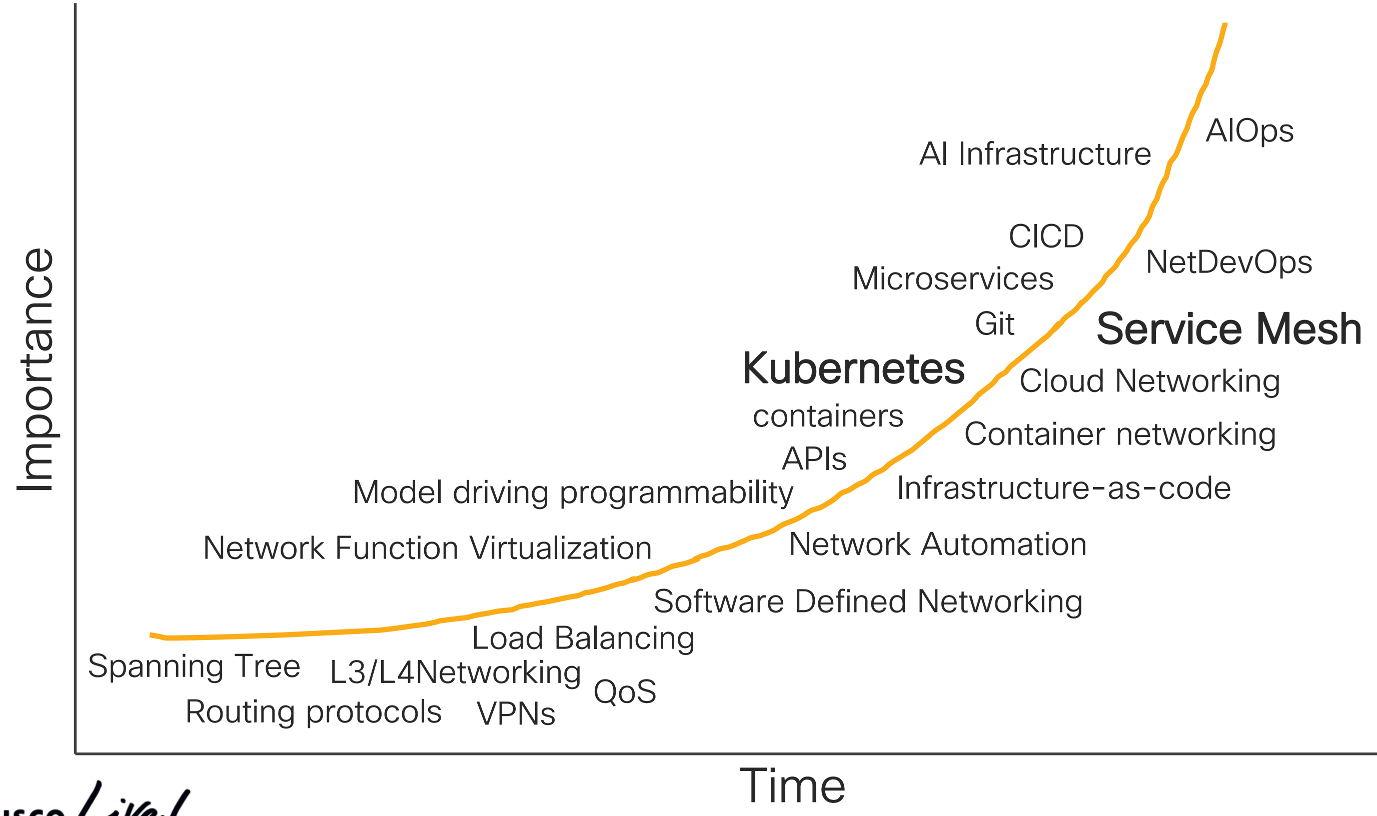
DevNet-1697

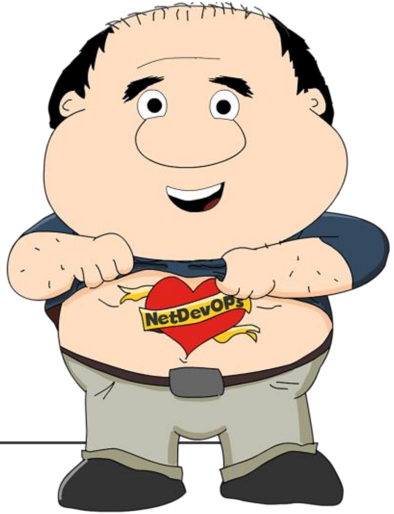
Agenda

- Cloud Native for the Network Engineer
- A somewhat brief Kubernetes review
- Introduction to Service Mesh
- Ok, so how does this affect my day job?

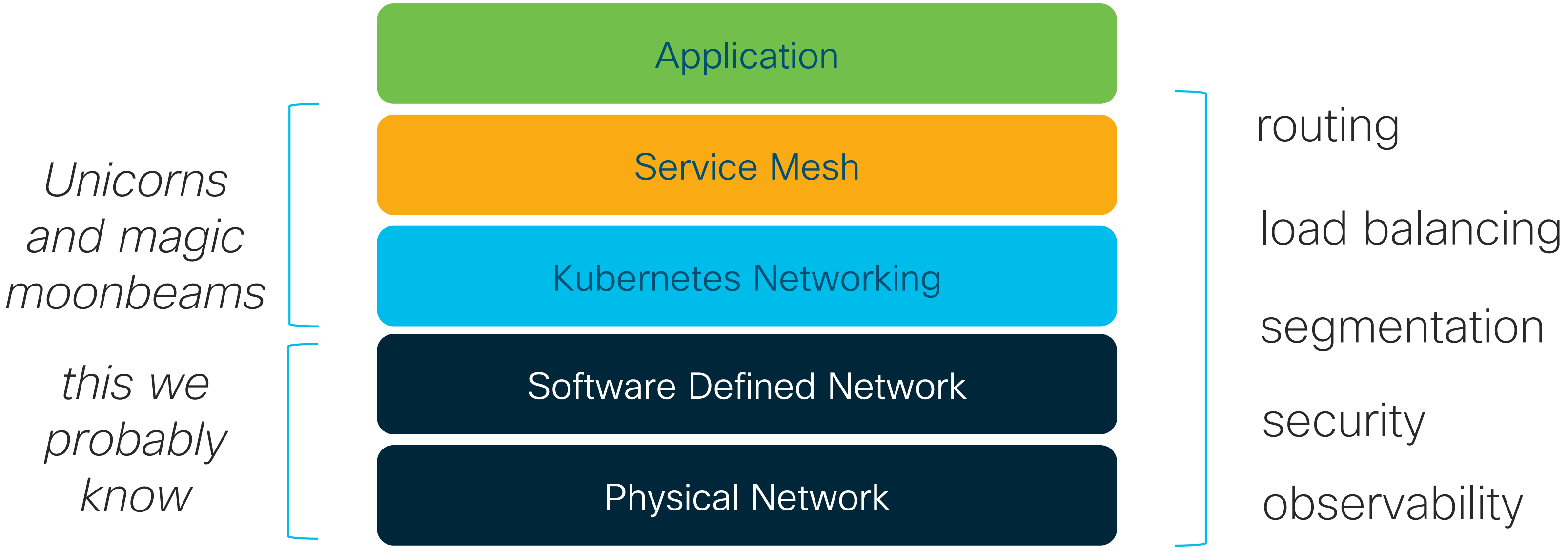
Cloud Native for Network Engineers

A Not Very Scientific Graph of what Network Engineers need to know in 2020





Cloud Native Abstractions



Application

Service Mesh

Kubernetes Networking

Software Defined Network

Physical Network

Unicorns and magic moonbeams

this we probably know

routing

load balancing

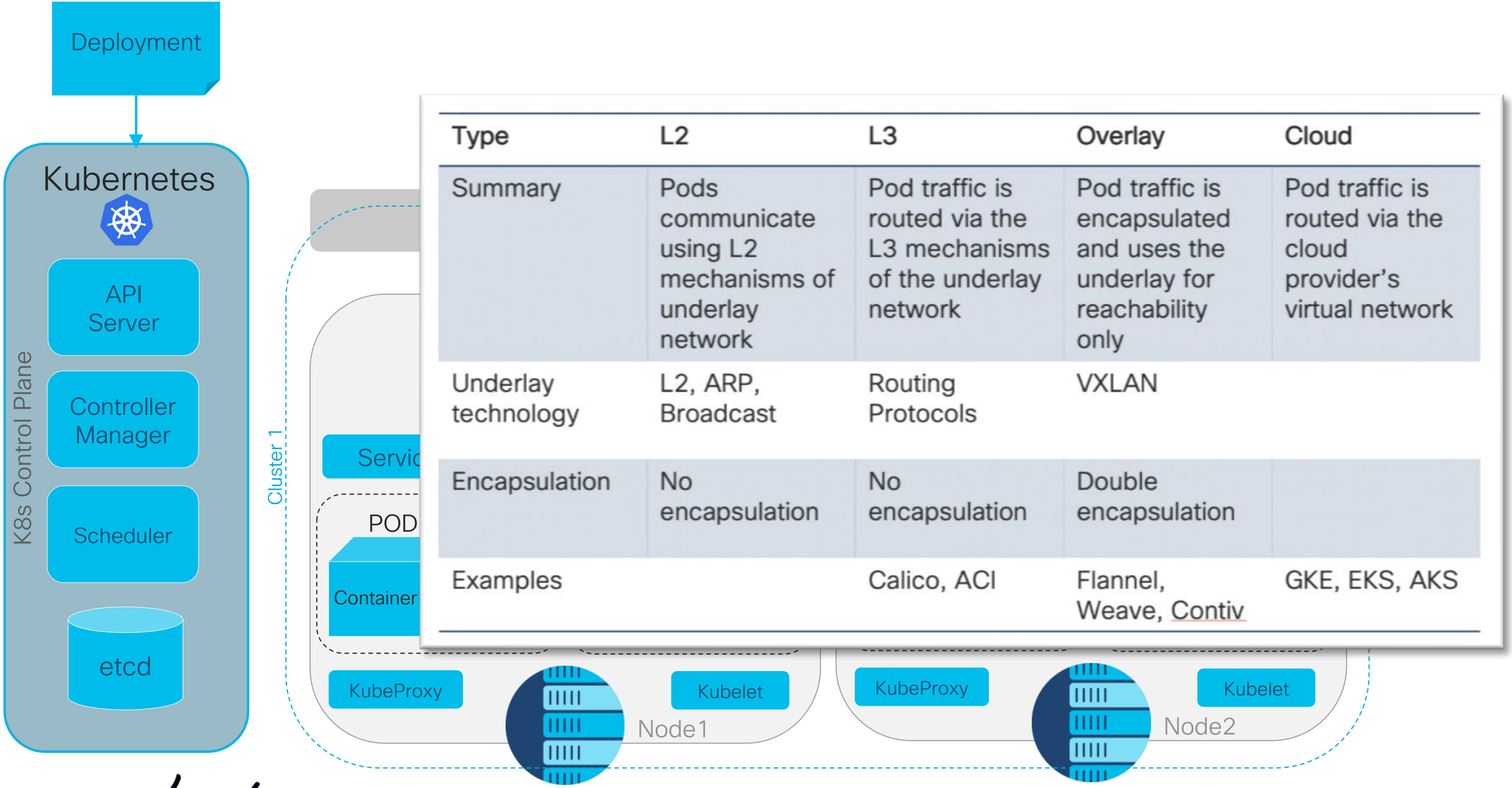
segmentation

security

observability

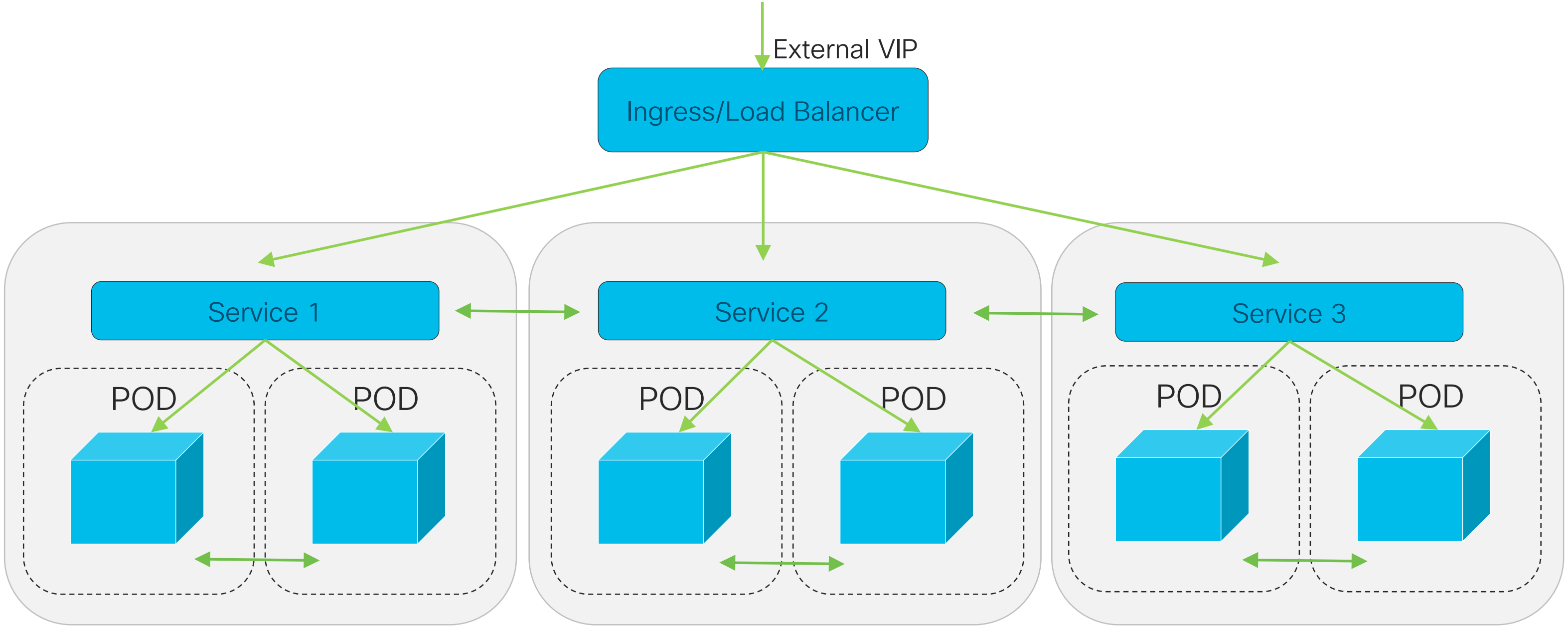
A somewhat brief Kubernetes Review

Kubernetes Components



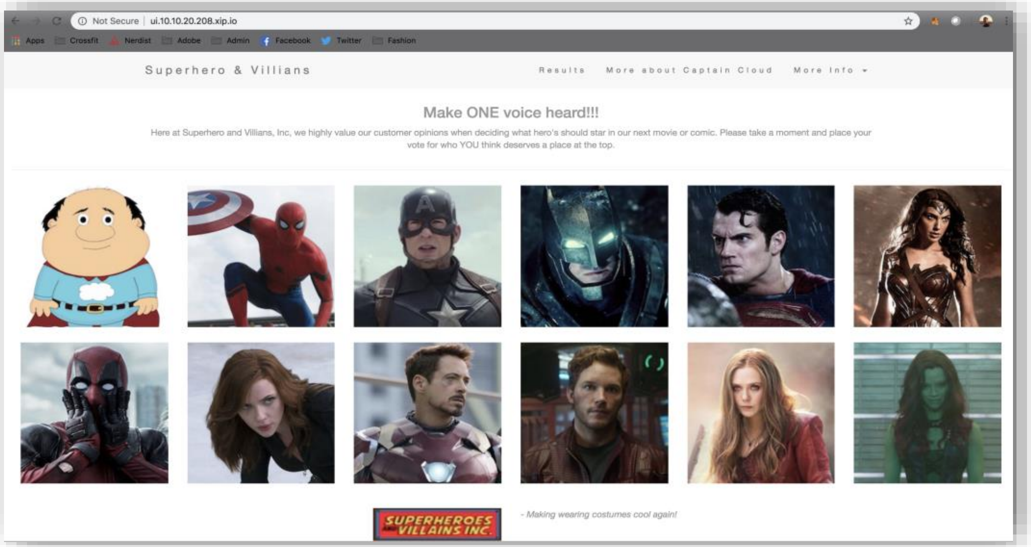
Type	L2	L3	Overlay	Cloud
Summary	Pods communicate using L2 mechanisms of underlay network	Pod traffic is routed via the L3 mechanisms of the underlay network	Pod traffic is encapsulated and uses the underlay for reachability only	Pod traffic is routed via the cloud provider's virtual network
Underlay technology	L2, ARP, Broadcast	Routing Protocols	VXLAN	
Encapsulation	No encapsulation	No encapsulation	Double encapsulation	
Examples		Calico, ACI	Flannel, Weave, Contiv	GKE, EKS, AKS

Kubernetes Logical Traffic Flow



https://medium.com/@zhaohuabing/which-one-is-the-right-choice-for-the-ingress-gateway-of-your-service-mesh-21a280d4a29c

Let’s look at a real Microservice – MyHero



Make ONE voice heard!!!

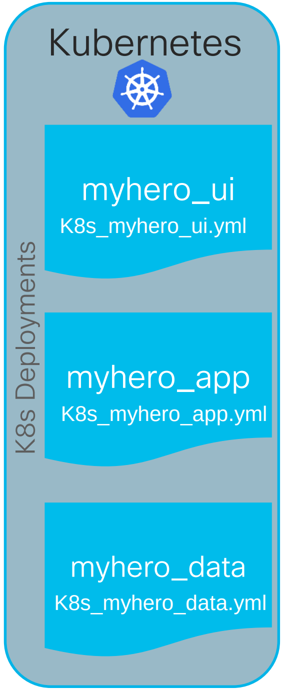
Here at Superhero and Villians, Inc, we highly value our customer opinions when deciding what hero's should star in our next movie or comic. Please take a moment and place your vote for who YOU think deserves a place at the top.

- Making wearing costumes cool again!

https://github.com/hpreston/myhero_demo

- In order to demonstrate the power of Istio we will install a microservices applications called MyHero.
- MyHero is a simple app that allows you to vote for your favourite superhero
- We will look at how Istio can be used to manage traffic to and from different microservices

MyHero Architecture



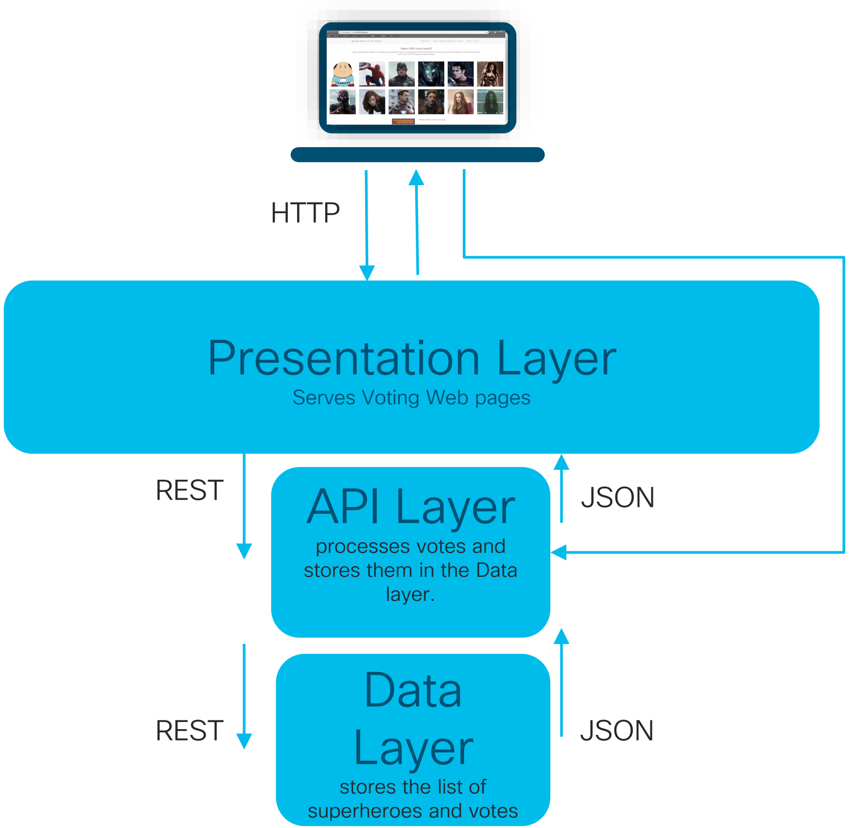
Kubernetes

K8s Deployments

myhero_ui
K8s_myhero_ui.yml

myhero_app
K8s_myhero_app.yml

myhero_data
K8s_myhero_data.yml



HTTP

Presentation Layer
Serves Voting Web pages

REST

API Layer
processes votes and stores them in the Data layer.

JSON

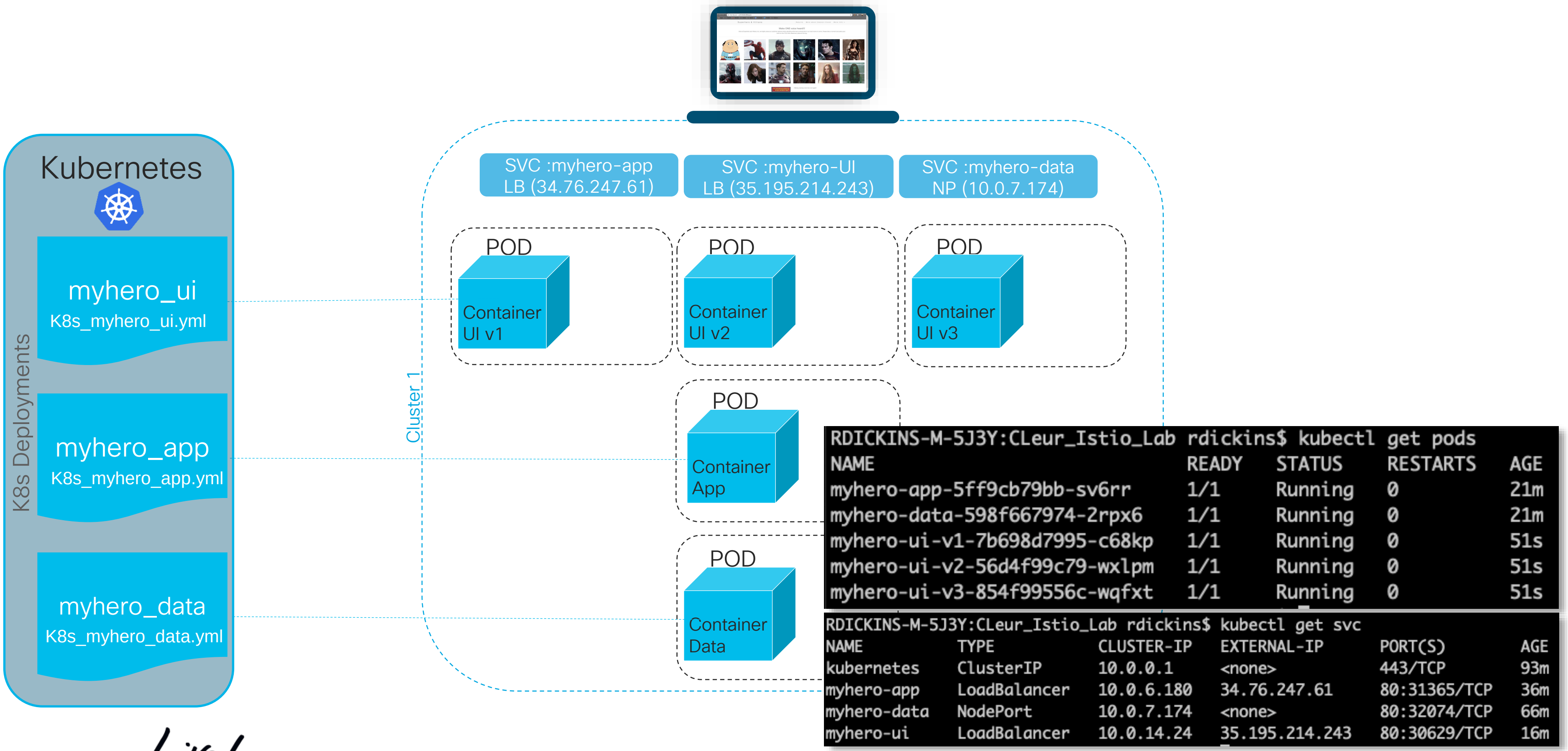
REST

Data Layer
stores the list of superheroes and votes

JSON

MyHero comprises of three layers. The presentation layer externally exposed via HTTP while all three layers loosely coupled by REST APIs

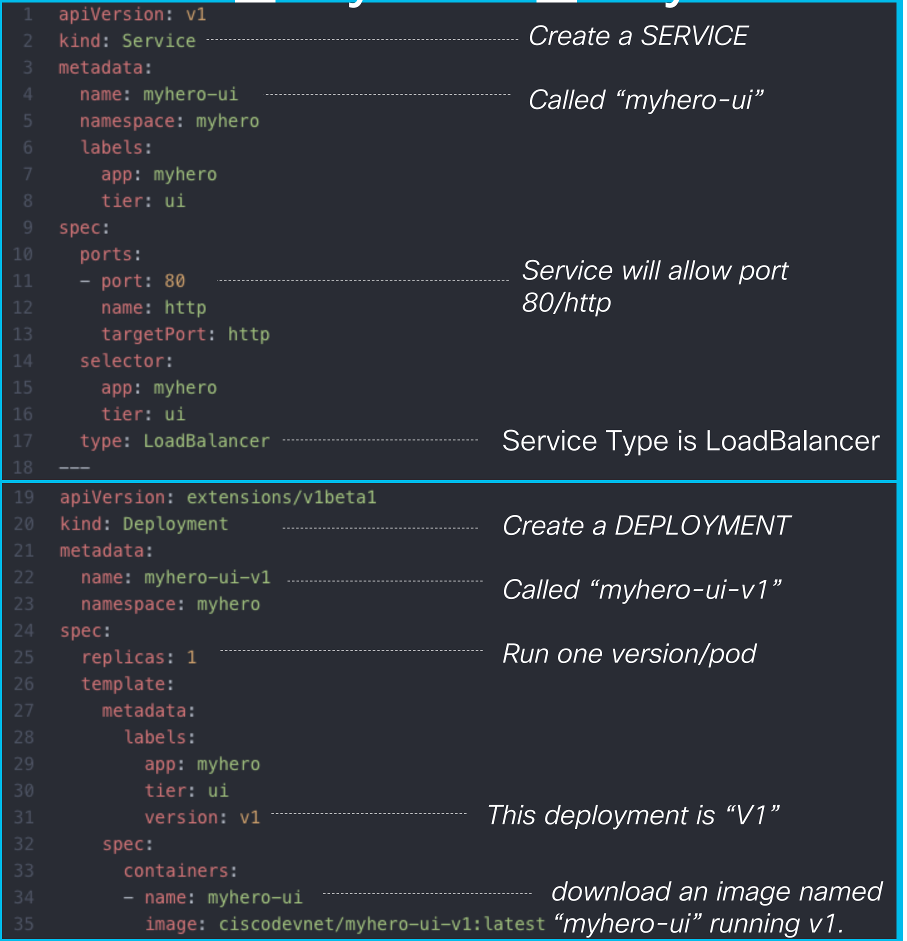
MyHero Pods



```
RDICKINS-M-5J3Y:CLeur_Istio_Lab rdickins$ kubectl get pods
NAME                            READY   STATUS    RESTARTS   AGE
myhero-app-5ff9cb79bb-sv6rr     1/1     Running   0          21m
myhero-data-598f667974-2rpx6    1/1     Running   0          21m
myhero-ui-v1-7b698d7995-c68kp   1/1     Running   0          51s
myhero-ui-v2-56d4f99c79-wxlpm   1/1     Running   0          51s
myhero-ui-v3-854f99556c-wqfxt   1/1     Running   0          51s
```

```
RDICKINS-M-5J3Y:CLeur_Istio_Lab rdickins$ kubectl get svc
NAME          TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)        AGE
kubernetes    ClusterIP      10.0.0.1      <none>           443/TCP        93m
myhero-app    LoadBalancer   10.0.6.180    34.76.247.61     80:31365/TCP   36m
myhero-data   NodePort       10.0.7.174    <none>           80:32074/TCP   66m
myhero-ui     LoadBalancer   10.0.14.24    35.195.214.243   80:30629/TCP   16m
```

k8s_myhero_ui.yml



```yaml
apiVersion: v1
kind: Service
metadata:
  name: myhero-ui
  namespace: myhero
  labels:
    app: myhero
    tier: ui
spec:
  ports:
  - port: 80
    name: http
    targetPort: http
  selector:
    app: myhero
    tier: ui
  type: LoadBalancer
---
```

Create a SERVICE

Called "myhero-ui"

Service will allow port 80/http

Service Type is LoadBalancer

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: myhero-ui-v1
  namespace: myhero
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: myhero
        tier: ui
        version: v1
    spec:
      containers:
      - name: myhero-ui
        image: ciscodevnet/myhero-ui-v1:latest
```

Create a DEPLOYMENT

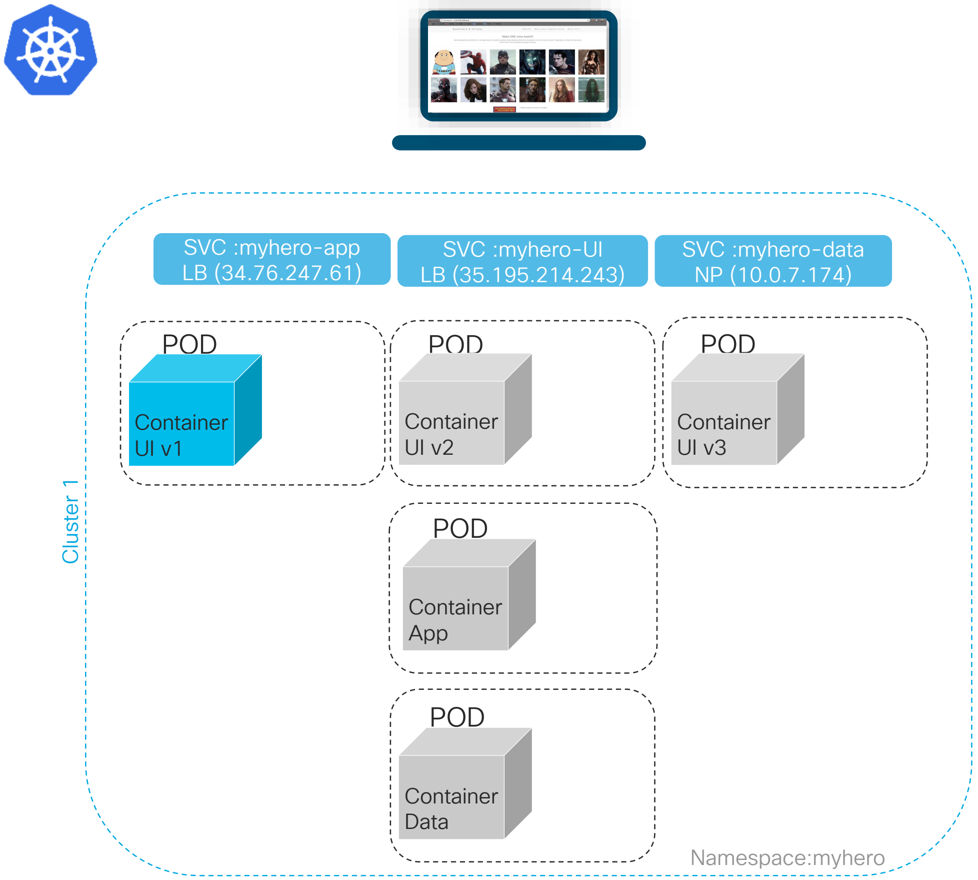
Called "myhero-ui-v1"

Run one version/pod

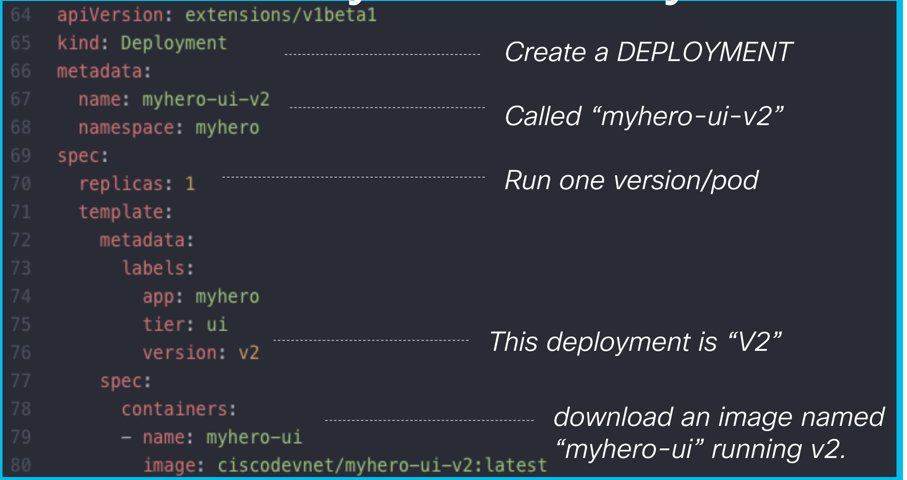
This deployment is "V1"

download an image named "myhero-ui" running v1.

https://github.com/dcgubbins/GKE-K8s-Istio-lab



k8s_myhero_ui.yml



```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: myhero-ui-v2
  namespace: myhero
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: myhero
        tier: ui
        version: v2
    spec:
      containers:
      - name: myhero-ui
        image: ciscodevnet/myhero-ui-v2:latest
```

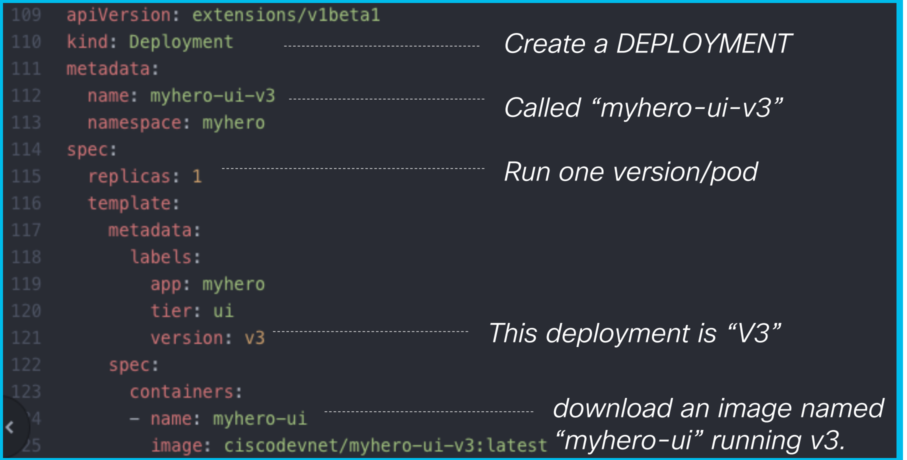
Create a DEPLOYMENT

Called “myhero-ui-v2”

Run one version/pod

This deployment is “V2”

download an image named “myhero-ui” running v2.



```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: myhero-ui-v3
  namespace: myhero
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: myhero
        tier: ui
        version: v3
    spec:
      containers:
      - name: myhero-ui
        image: ciscodevnet/myhero-ui-v3:latest
```

Create a DEPLOYMENT

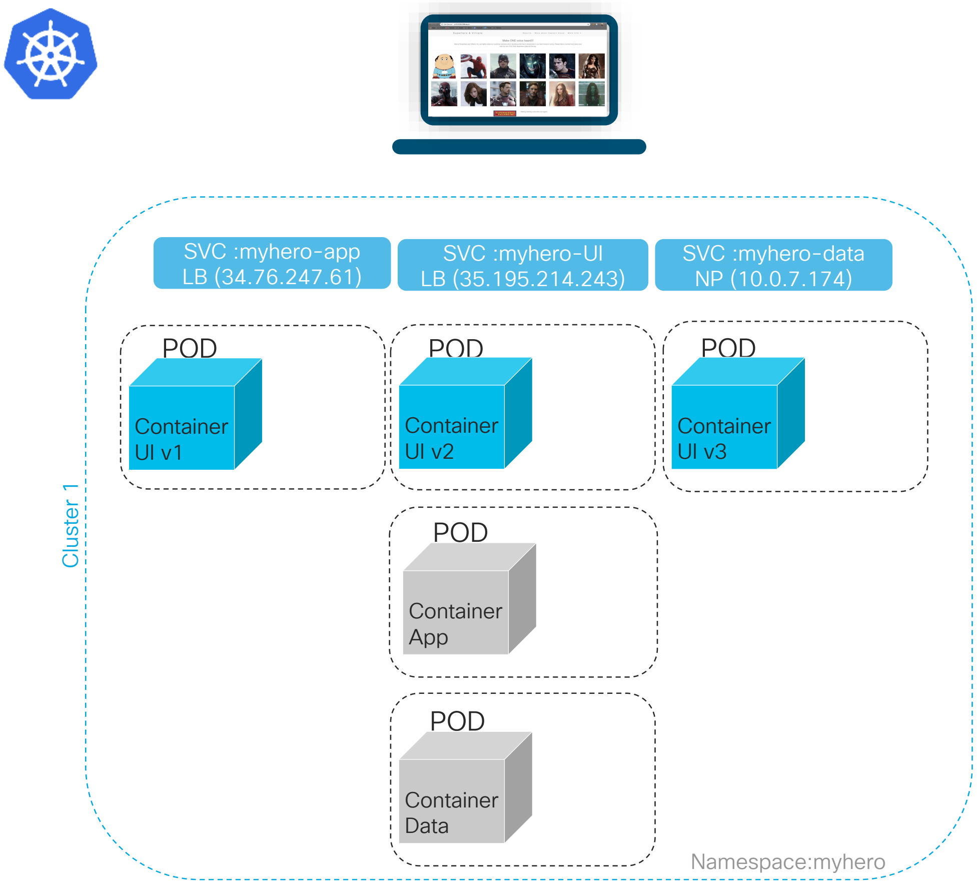
Called “myhero-ui-v3”

Run one version/pod

This deployment is “V3”

download an image named “myhero-ui” running v3.

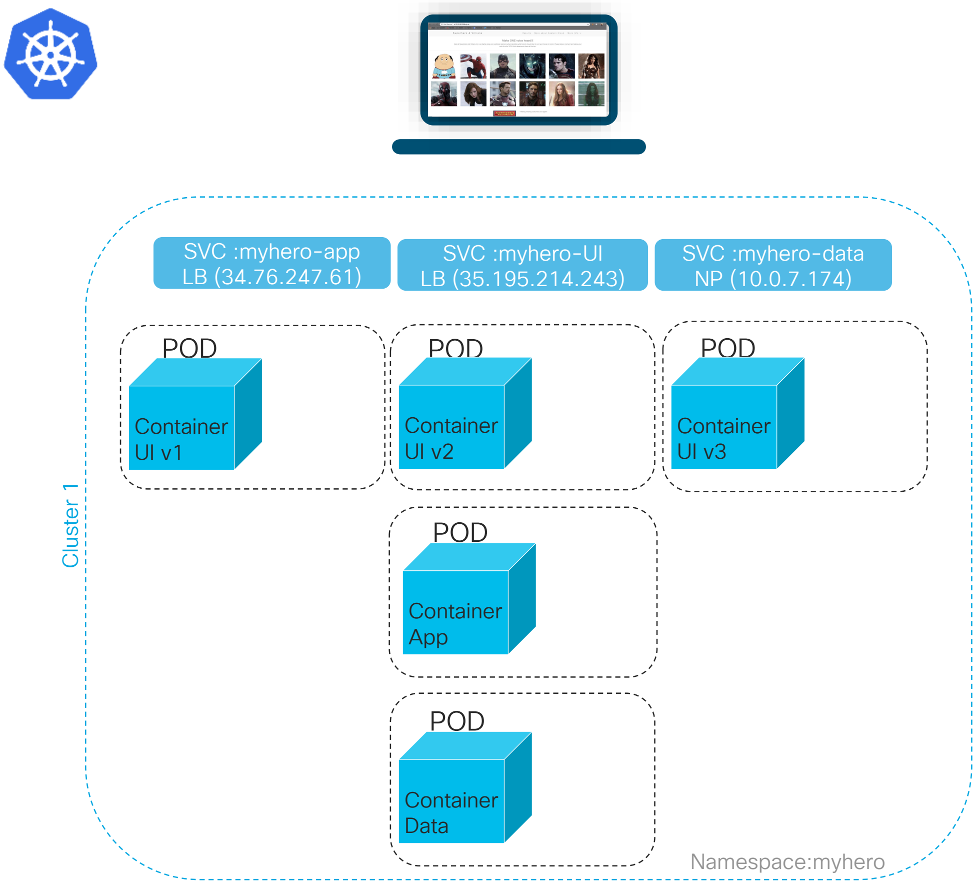
https://github.com/dcgubbins/GKE-K8s-Istio-lab



K8s is a container orchestration tool that automates:

- containers at scale
- container deployment
- container-to-container networking
- load balancing

But more is needed….



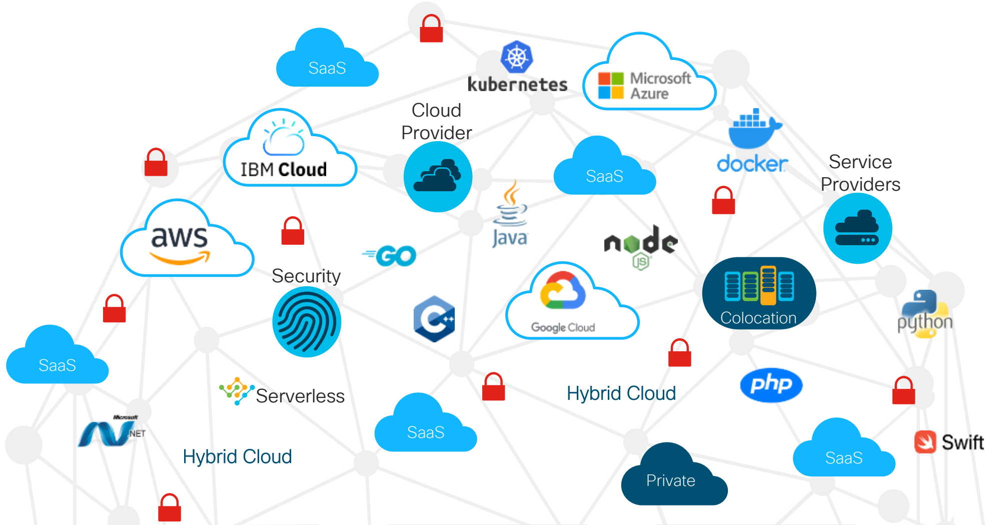
Introduction to Service Mesh

Service Mesh

A service mesh is a dedicated infrastructure layer for handling reliable service-to-service communication

The service mesh is a networking model that sits at a layer of abstraction above TCP/IP. It assumes that the underlying L3/L4 network is present and capable of delivering bytes from point to point.

https://buoyant.io/2017/04/25/whats-a-service-mesh-and-why-do-i-need-one/



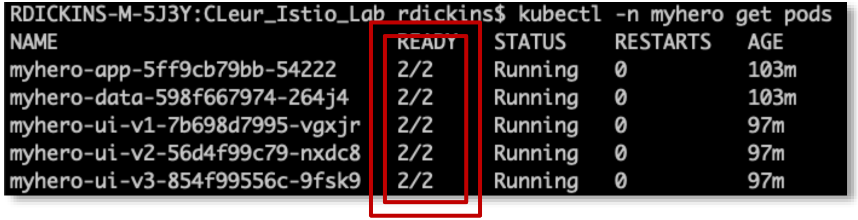
Service Mesh Functions:

- Service discovery
- Load balancing
- Failure recovery
- Metrics and monitoring

- A/B testing
- Canary rollouts
- Rate limiting
- Access control
- End-to-end authentication
- Behavioural insights

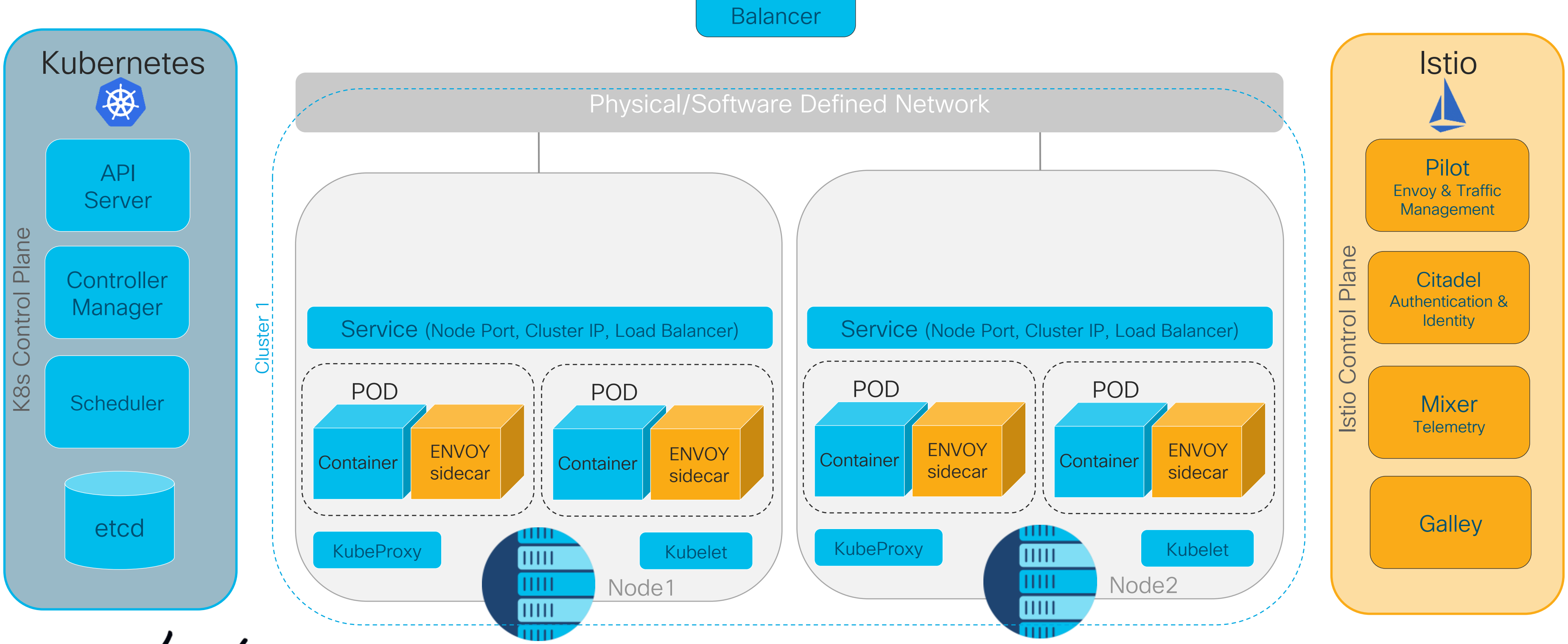
envoy CONDUIT LINKERD Istio

K8s + Istio



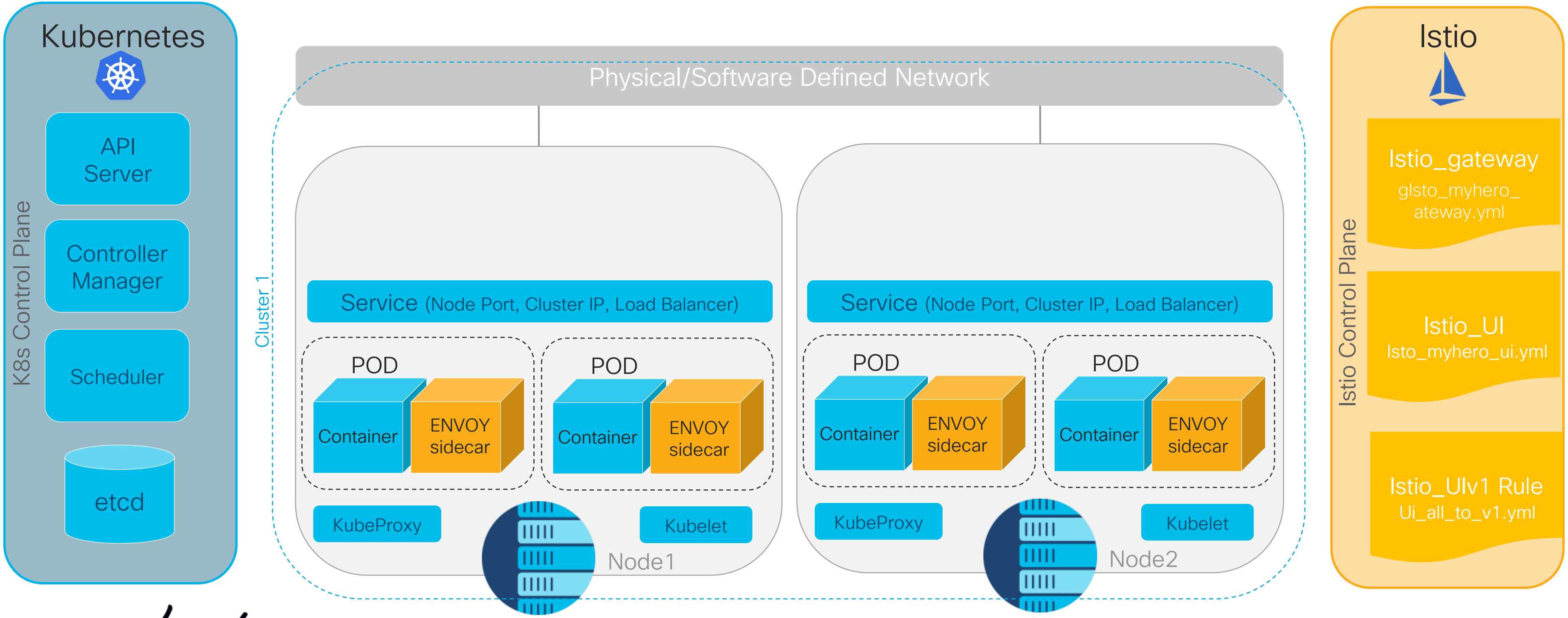
```
RDICKINS-M-5J3Y:CLeur_Istio_Lab rdickins$ kubectl -n myhero get pods
NAME                             READY   STATUS    RESTARTS   AGE
myhero-app-5ff9cb79bb-54222      2/2     Running   0          103m
myhero-data-598f667974-264j4     2/2     Running   0          103m
myhero-ui-v1-7b698d7995-vgxjr    2/2     Running   0          97m
myhero-ui-v2-56d4f99c79-nxdc8    2/2     Running   0          97m
myhero-ui-v3-854f99556c-9fsk9    2/2     Running   0          97m
```

Load Balancer



Istio

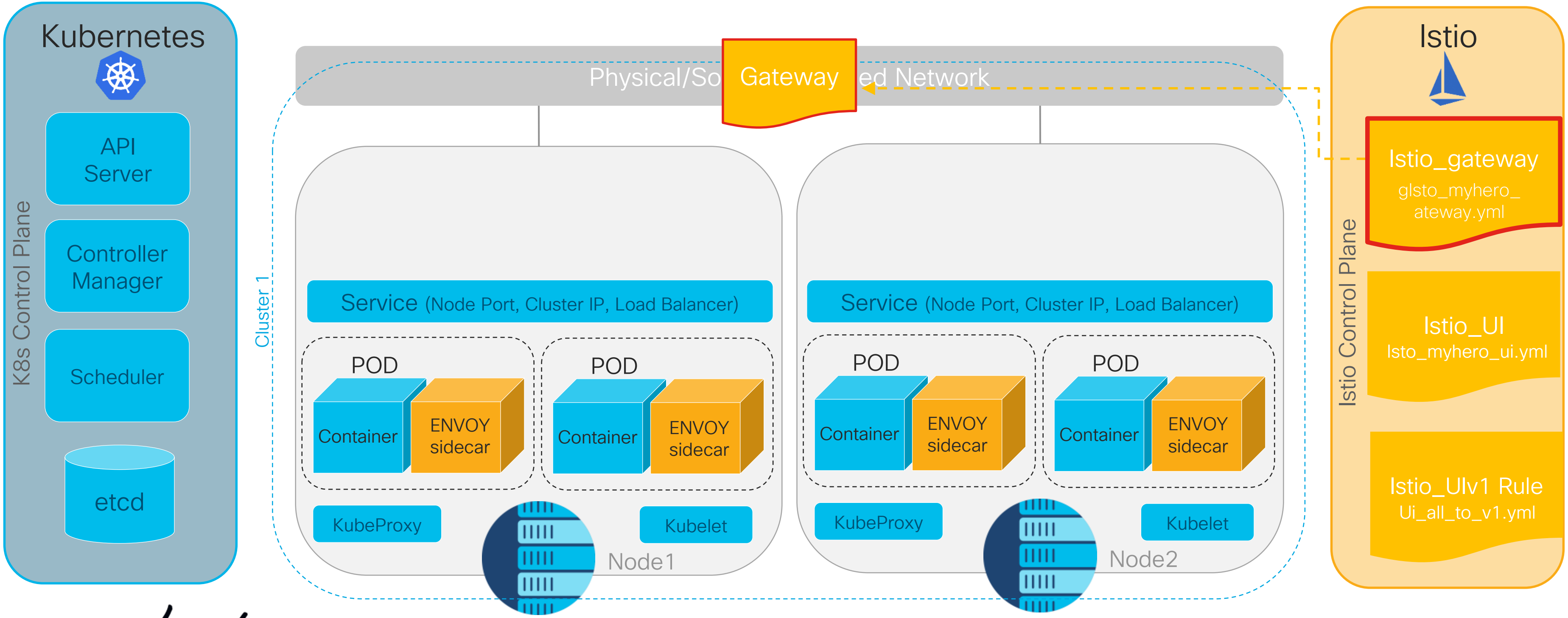
Traffic Management Example



Istio

Traffic Management Example

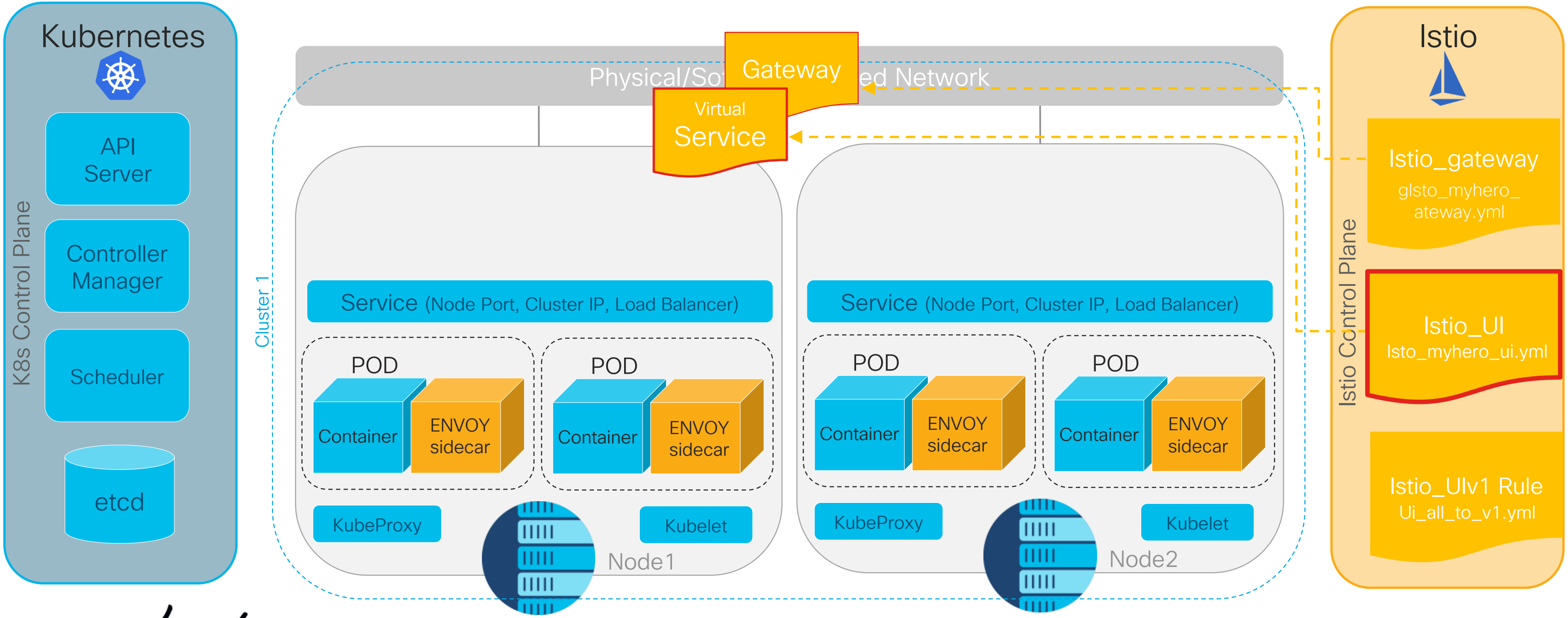
A Gateway configures a load balancer for HTTP/TCP traffic, most commonly operating at the edge of the mesh to enable ingress traffic for an application.



Istio

Traffic Management Example

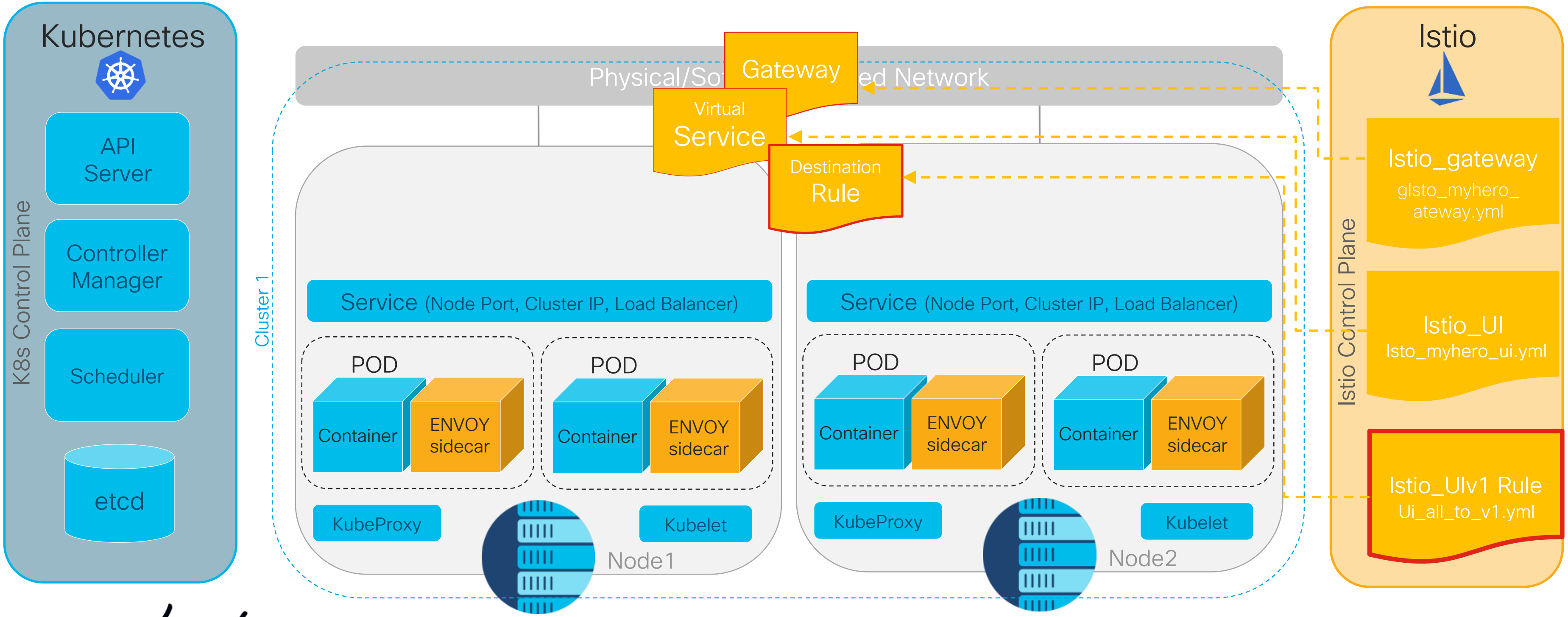
A VirtualService defines the rules that control how requests for a service are routed within an Istio service mesh.



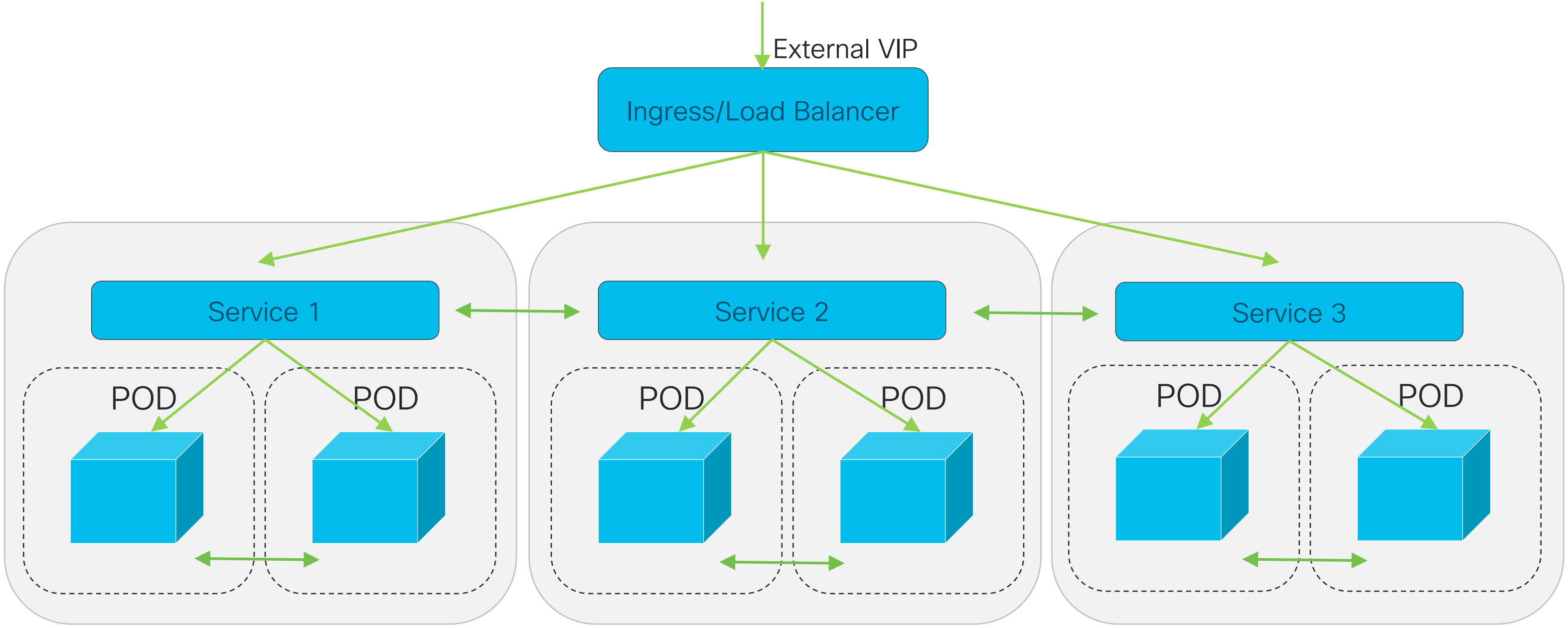
Istio Components

Traffic Management Example

A DestinationRule configures the set of policies to be applied to a request after VirtualService routing has occurred.

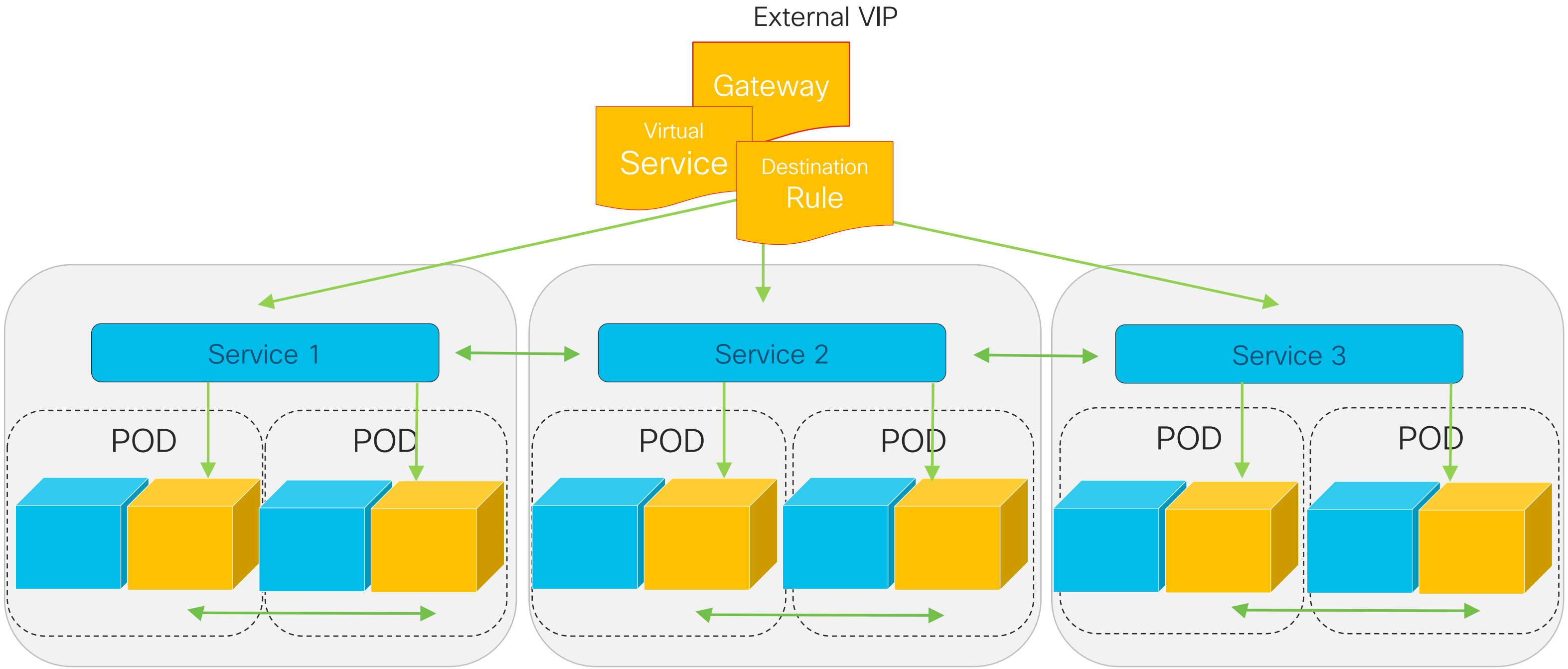


Kubernetes Logical Traffic Flow



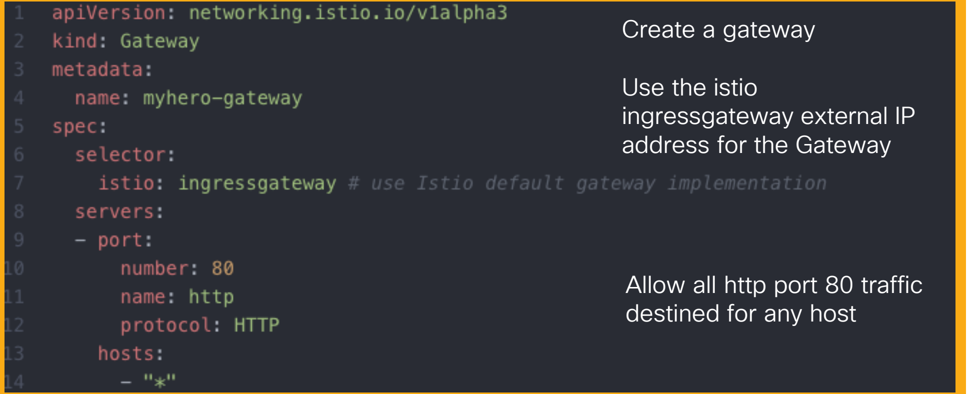
https://medium.com/@zhaohuabing/which-one-is-the-right-choice-for-the-ingress-gateway-of-your-service-mesh-21a280d4a29c

Istio Logical Traffic Flow



https://medium.com/@zhaohuabing/which-one-is-the-right-choice-for-the-ingress-gateway-of-your-service-mesh-21a280d4a29c

Create an Ingress Gateway



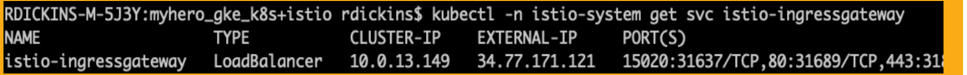
```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: myhero-gateway
spec:
  selector:
    istio: ingressgateway # use Istio default gateway implementation
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
      - "*"
```

Create a gateway

Use the istio ingressgateway external IP address for the Gateway

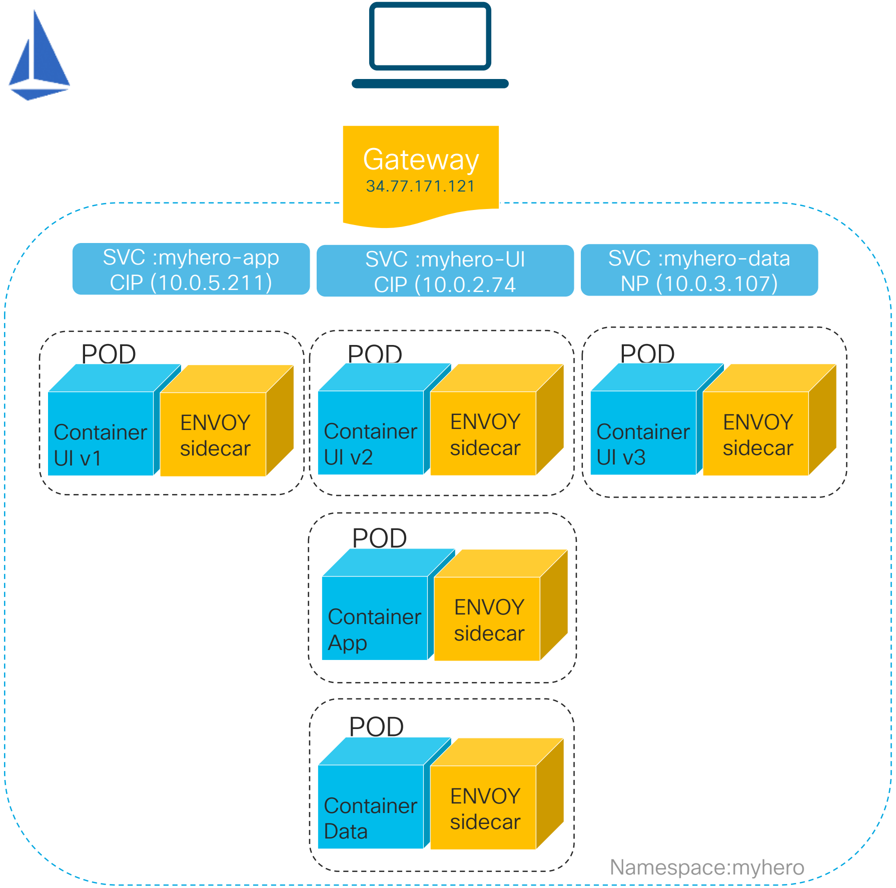
Allow all http port 80 traffic destined for any host

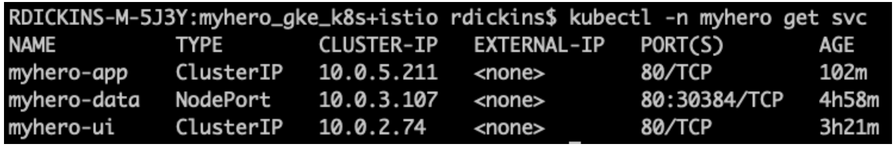
K8s UI and App Service Types are changed to ClusterP (no need for Load Balancer now we have Istio)



```
RDICKINS-M-5J3Y:myhero_gke_k8s+istio rdickins$ kubectl -n istio-system get svc istio-ingressgateway
NAME                   TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)
istio-ingressgateway   LoadBalancer   10.0.13.149   34.77.171.121   15020:31637/TCP,80:31689/TCP,443:31
```

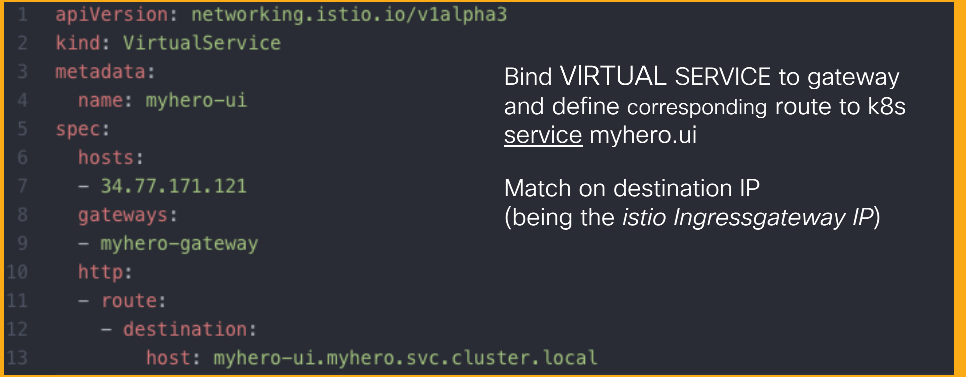
https://github.com/dcgubbins/GKE-K8s-Istio-lab





```
RDICKINS-M-5J3Y:myhero_gke_k8s+istio rdickins$ kubectl -n myhero get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
myhero-app    ClusterIP   10.0.5.211   <none>        80/TCP         102m
myhero-data   NodePort    10.0.3.107   <none>        80:30384/TCP   4h58m
myhero-ui     ClusterIP   10.0.2.74    <none>        80/TCP         3h21m
```

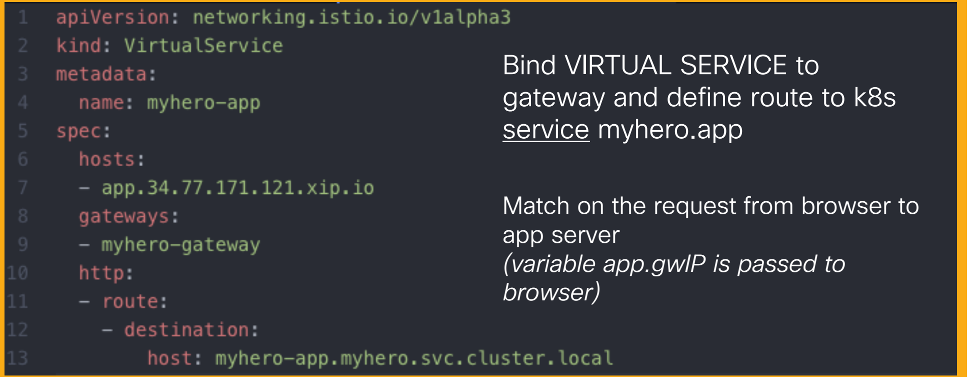
Create a Virtual Service



```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: myhero-ui
spec:
  hosts:
  - 34.77.171.121
  gateways:
  - myhero-gateway
  http:
  - route:
    - destination:
        host: myhero-ui.myhero.svc.cluster.local
```

Bind VIRTUAL SERVICE to gateway and define corresponding route to k8s service myhero.ui

Match on destination IP (being the *istio Ingressgateway IP*)

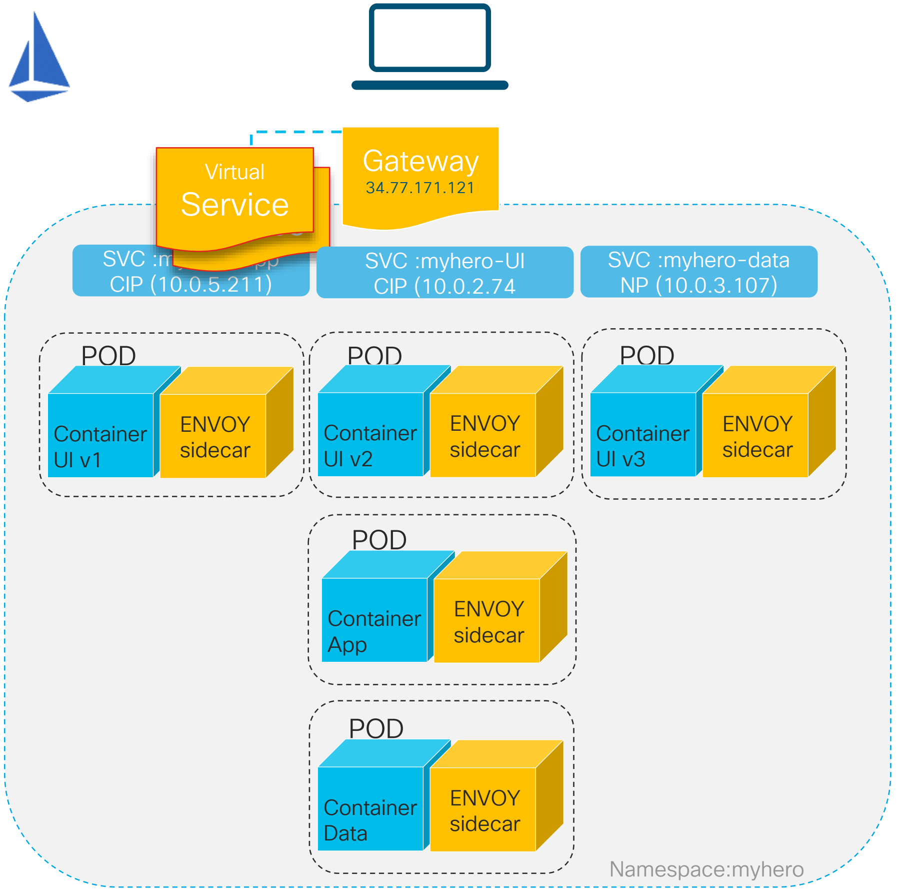


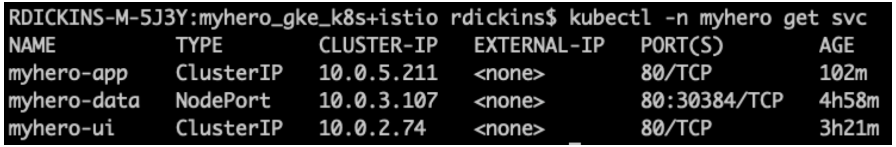
```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: myhero-app
spec:
  hosts:
  - app.34.77.171.121.xip.io
  gateways:
  - myhero-gateway
  http:
  - route:
    - destination:
        host: myhero-app.myhero.svc.cluster.local
```

Bind VIRTUAL SERVICE to gateway and define route to k8s service myhero.app

Match on the request from browser to app server *(variable app.gwIP is passed to browser)*

https://github.com/dcgubbins/GKE-K8s-Istio-lab

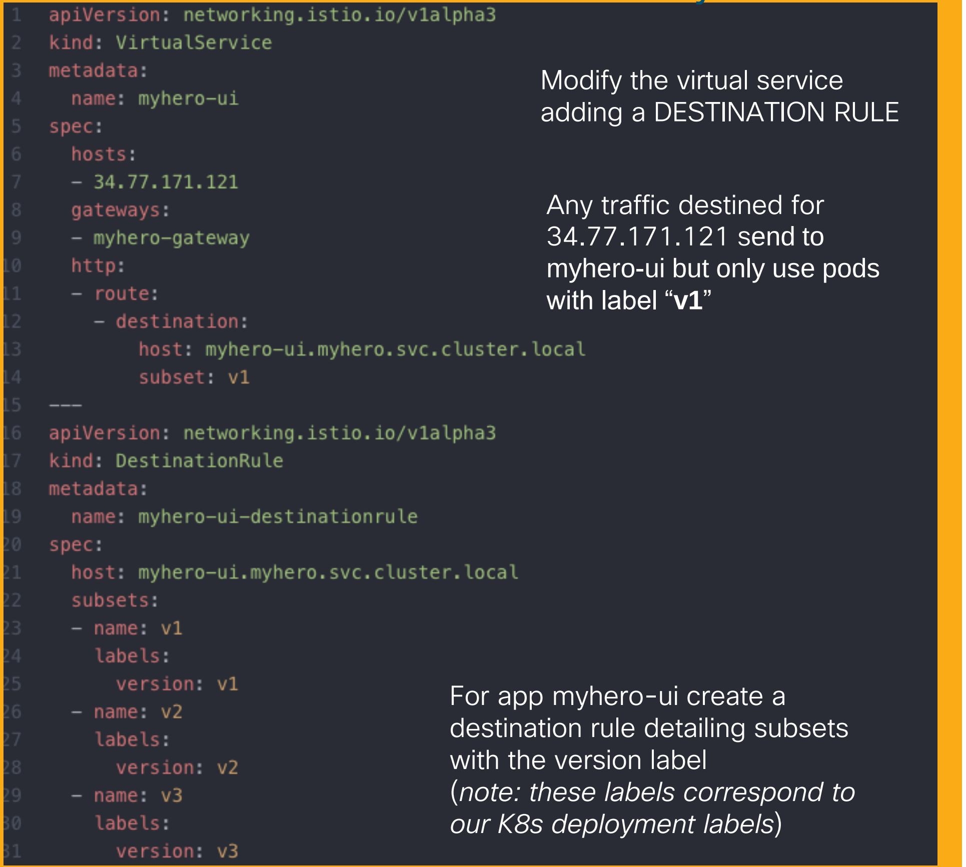




```
RDICKINS-M-5J3Y:myhero_gke_k8s+istio rdickins$ kubectl -n myhero get svc
NAME          TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
myhero-app    ClusterIP   10.0.5.211    <none>        80/TCP         102m
myhero-data   NodePort    10.0.3.107    <none>        80:30384/TCP   4h58m
myhero-ui     ClusterIP   10.0.2.74     <none>        80/TCP         3h21m
```

Create a Destination Rule

Send All Traffic to UI v1 only



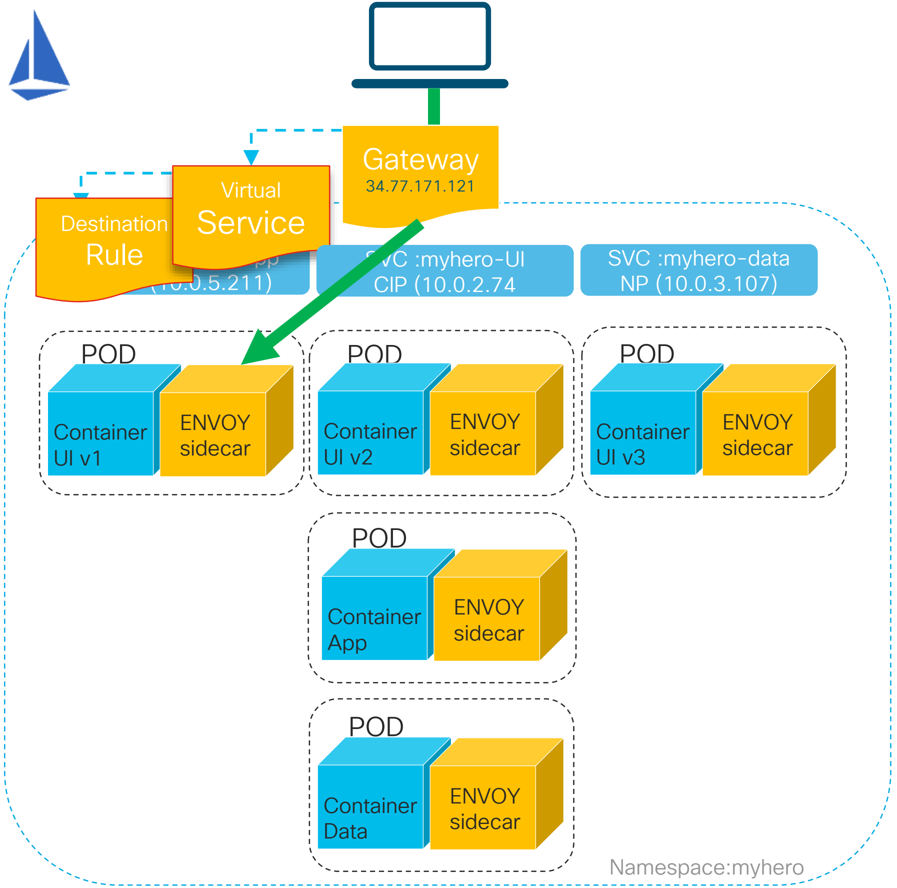
```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: myhero-ui
spec:
  hosts:
  - 34.77.171.121
  gateways:
  - myhero-gateway
  http:
  - route:
    - destination:
        host: myhero-ui.myhero.svc.cluster.local
        subset: v1
---
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: myhero-ui-destinationrule
spec:
  host: myhero-ui.myhero.svc.cluster.local
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
  - name: v3
    labels:
      version: v3
```

Modify the virtual service adding a DESTINATION RULE

Any traffic destined for 34.77.171.121 send to myhero-ui but only use pods with label “**v1**”

For app myhero-ui create a destination rule detailing subsets with the version label (*note: these labels correspond to our K8s deployment labels*)

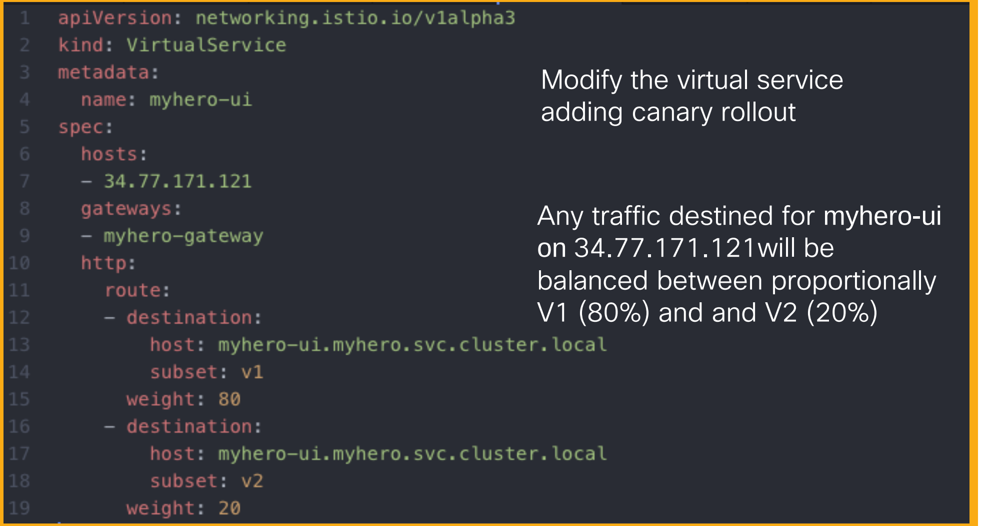
https://github.com/dcgubbins/GKE-K8s-Istio-lab



Destinations Rules are Applied AFTER Virtual Service

Modify the Virtual Service

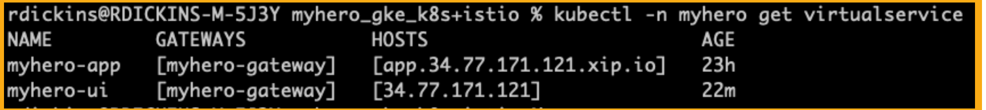
Canary Rollout



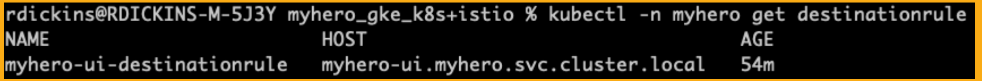
```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: myhero-ui
spec:
  hosts:
  - 34.77.171.121
  gateways:
  - myhero-gateway
  http:
    route:
    - destination:
        host: myhero-ui.myhero.svc.cluster.local
        subset: v1
      weight: 80
    - destination:
        host: myhero-ui.myhero.svc.cluster.local
        subset: v2
      weight: 20
```

Modify the virtual service adding canary rollout

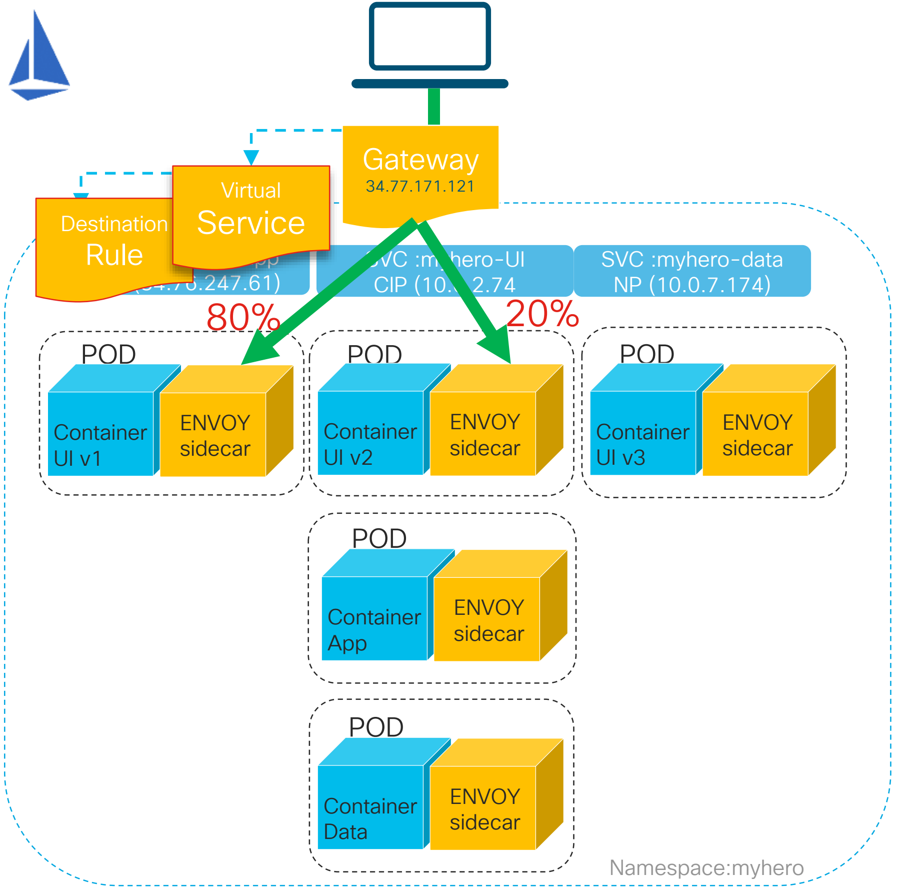
Any traffic destined for myhero-ui on 34.77.171.121will be balanced between proportionally V1 (80%) and and V2 (20%)



```
rdickins@RDICKINS-M-5J3Y myhero_gke_k8s+istio % kubectl -n myhero get virtualservice
NAME          GATEWAYS           HOSTS                        AGE
myhero-app    [myhero-gateway]   [app.34.77.171.121.xip.io]   23h
myhero-ui     [myhero-gateway]   [34.77.171.121]              22m
```



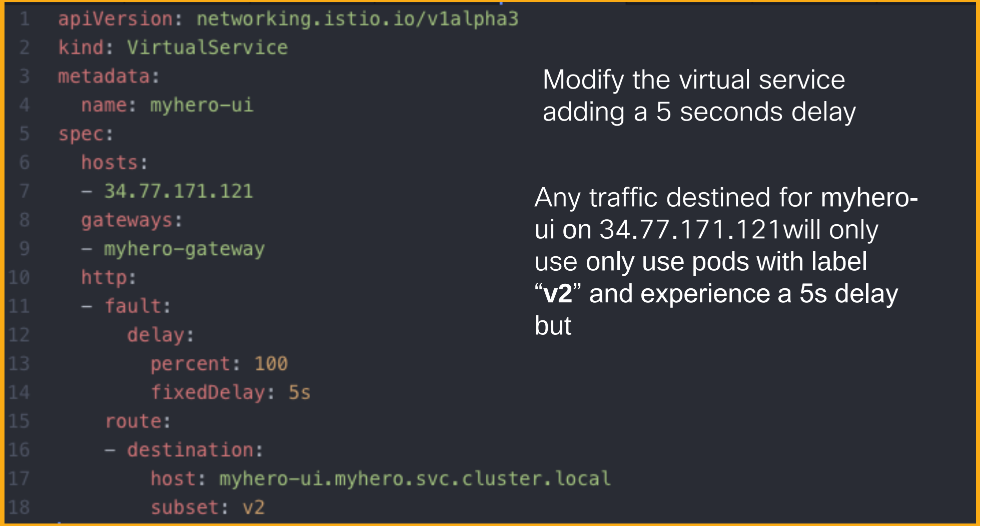
```
rdickins@RDICKINS-M-5J3Y myhero_gke_k8s+istio % kubectl -n myhero get destinationrule
NAME                         HOST                                 AGE
myhero-ui-destinationrule    myhero-ui.myhero.svc.cluster.local   54m
```



https://github.com/dcgubbins/GKE-K8s-Istio-lab

Modify the Virtual Service

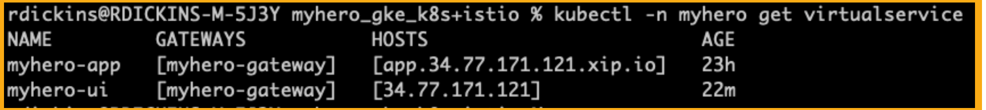
Inject a 5s Delay



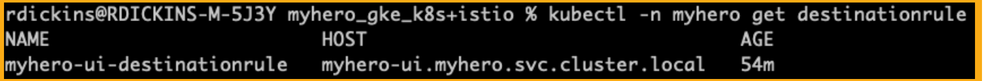
```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: myhero-ui
spec:
  hosts:
  - 34.77.171.121
  gateways:
  - myhero-gateway
  http:
  - fault:
      delay:
        percent: 100
        fixedDelay: 5s
    route:
    - destination:
        host: myhero-ui.myhero.svc.cluster.local
        subset: v2
```

Modify the virtual service adding a 5 seconds delay

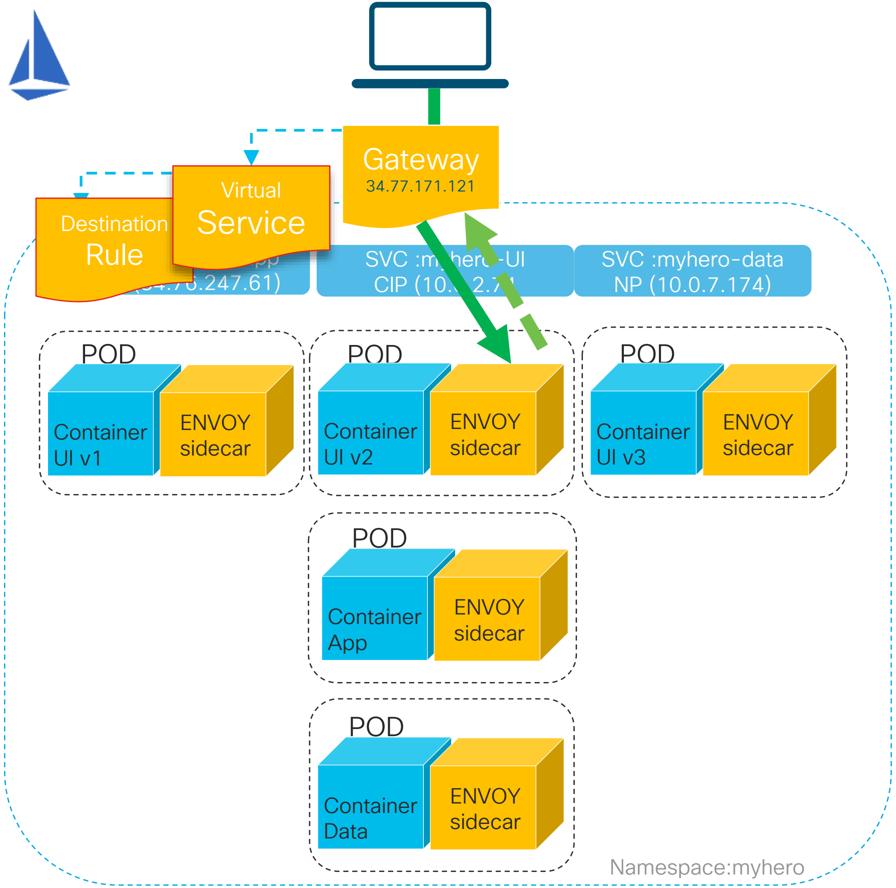
Any traffic destined for myhero-ui on 34.77.171.121will only use only use pods with label “**v2**” and experience a 5s delay but



```
rdickins@RDICKINS-M-5J3Y myhero_gke_k8s+istio % kubectl -n myhero get virtualservice
NAME          GATEWAYS           HOSTS                      AGE
myhero-app    [myhero-gateway]   [app.34.77.171.121.xip.io]   23h
myhero-ui     [myhero-gateway]   [34.77.171.121]            22m
```



```
rdickins@RDICKINS-M-5J3Y myhero_gke_k8s+istio % kubectl -n myhero get destinationrule
NAME                         HOST                                  AGE
myhero-ui-destinationrule    myhero-ui.myhero.svc.cluster.local    54m
```



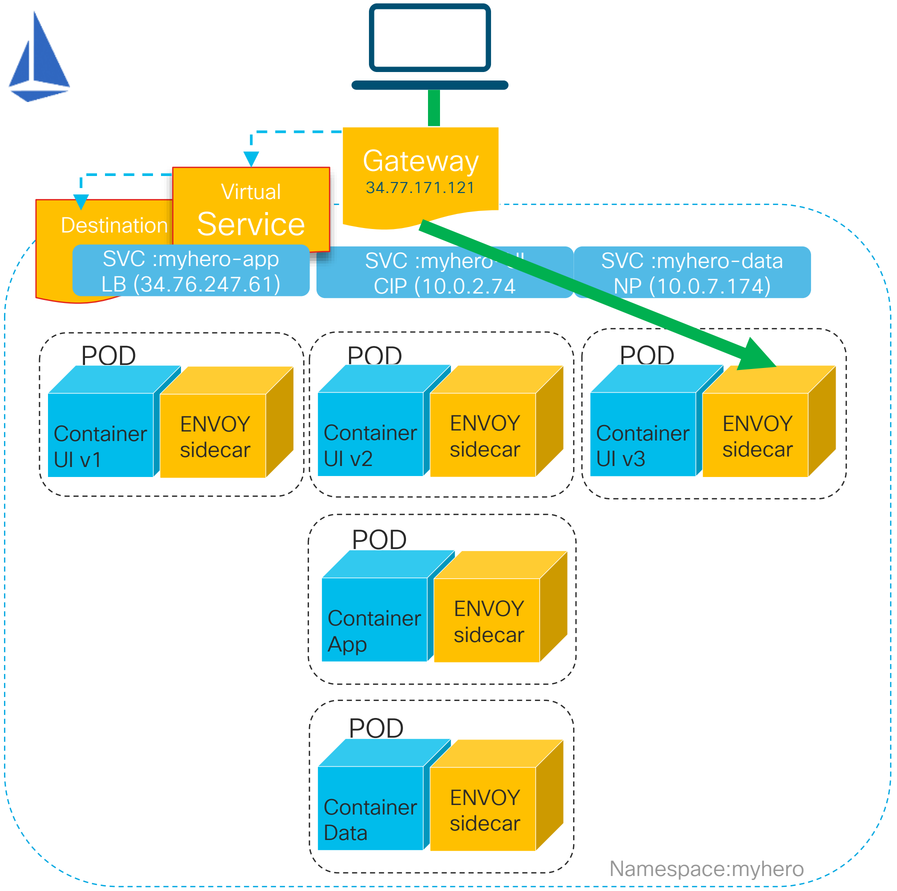
https://github.com/dcgubbins/GKE-K8s-Istio-lab

So Much More

Service Mesh Functions:

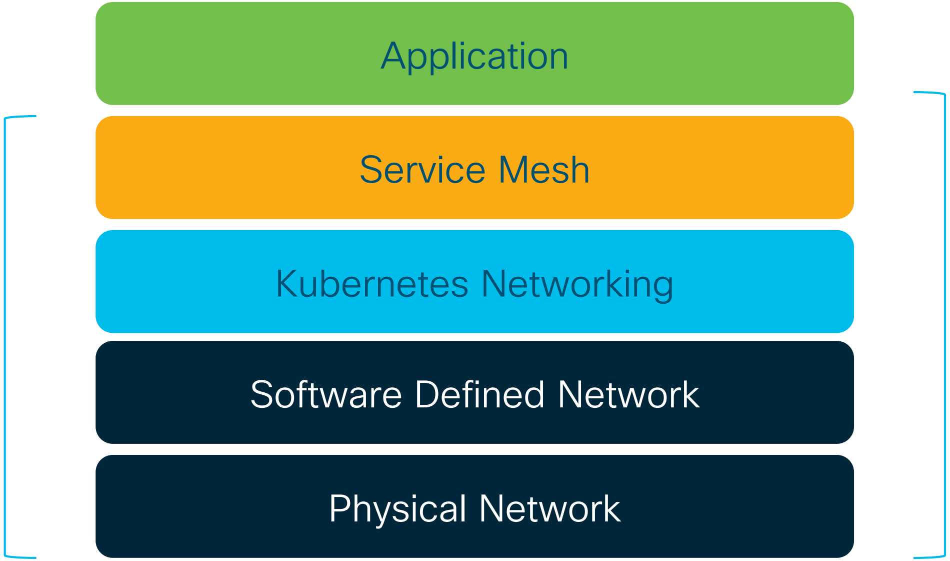
- Service discovery
- Load balancing
- Failure recovery
- Metrics and monitoring

- A/B testing
- Canary rollouts
- Rate limiting
- Access control
- End-to-end authentication
- Behavioural insights



OK, so how does this affect my day job?

Cloud Native Abstractions



Now we know ALL this (kinda)

routing

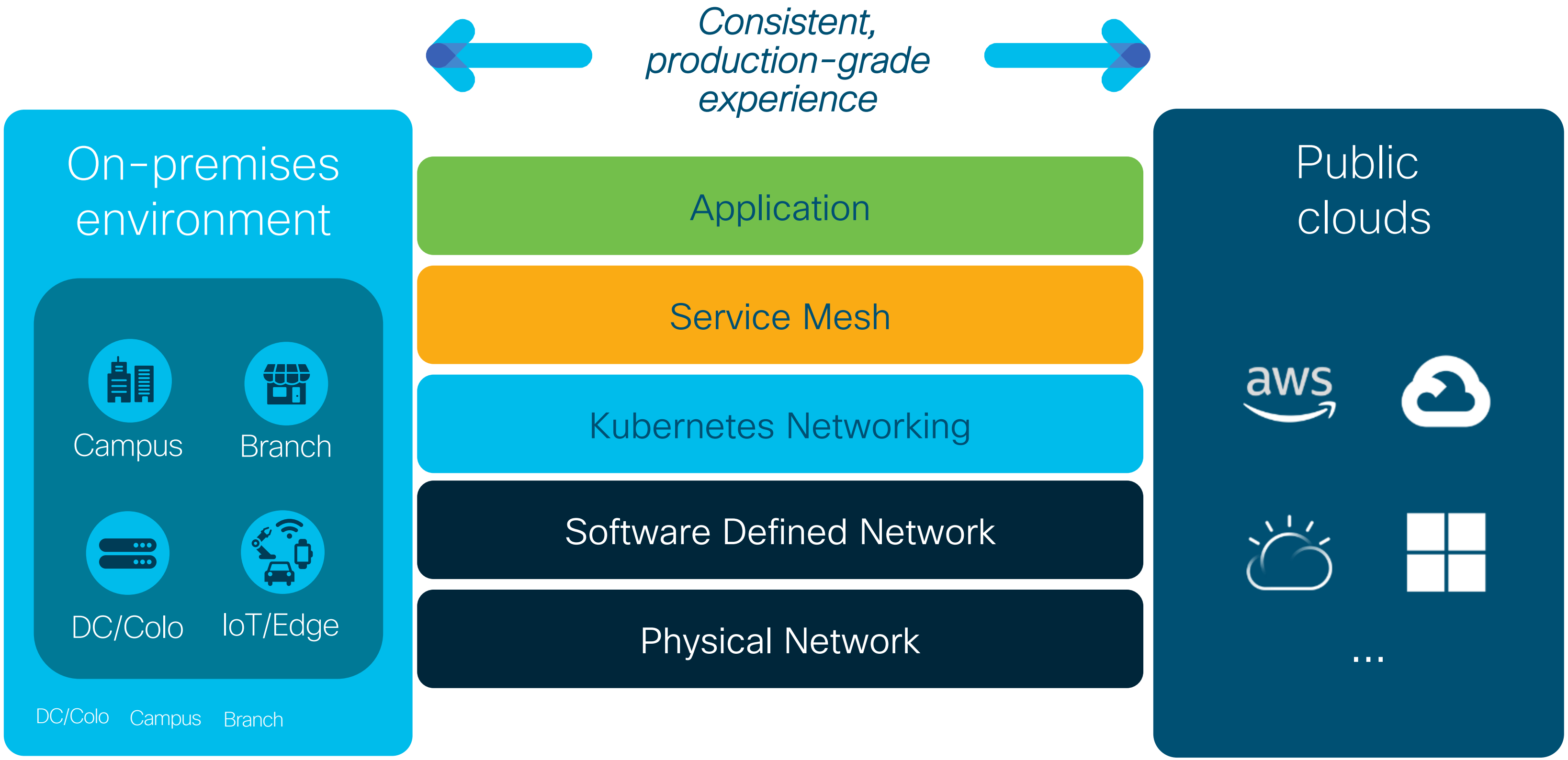
load balancing

segmentation

security

observability

Delivering MyHero Everywhere



Consistent, production-grade experience

On-premises environment

Campus

Branch

DC/Colo

IoT/Edge

DC/Colo Campus Branch

Application

Service Mesh

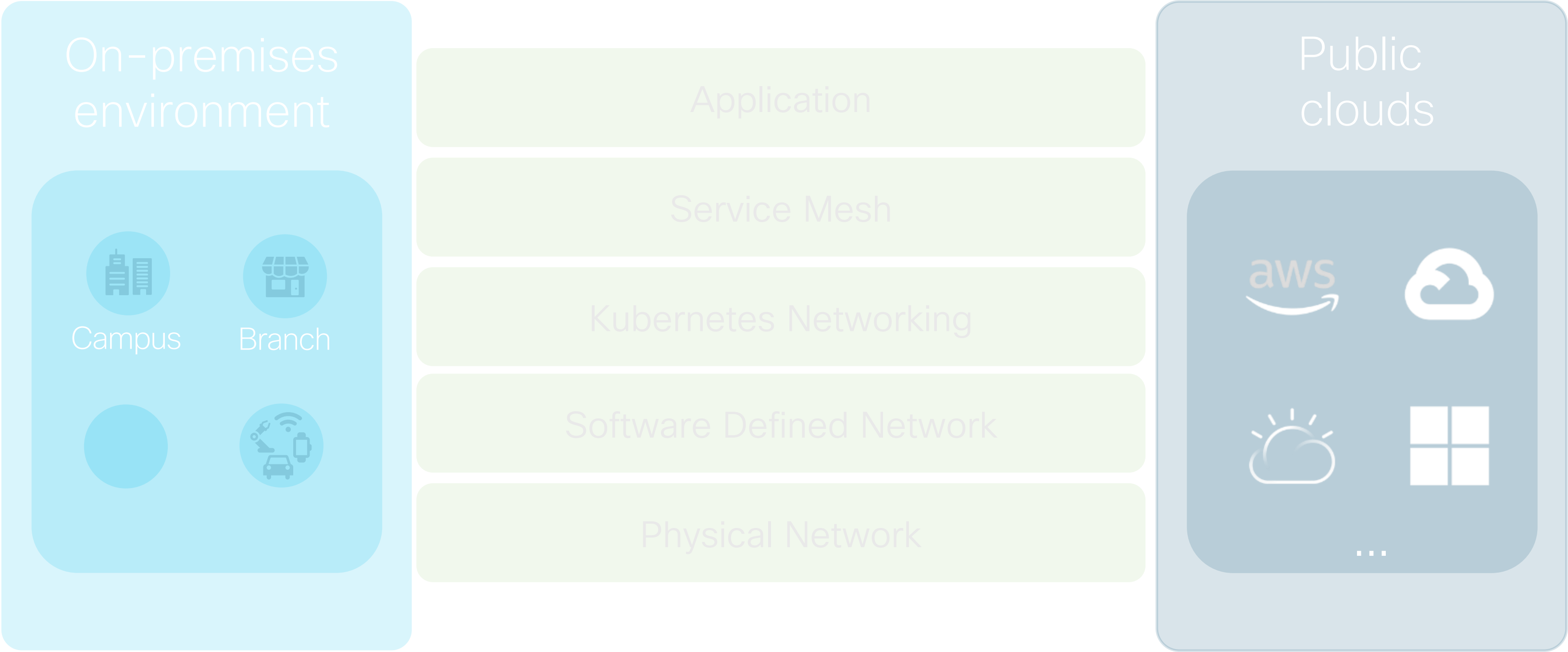
Kubernetes Networking

Software Defined Network

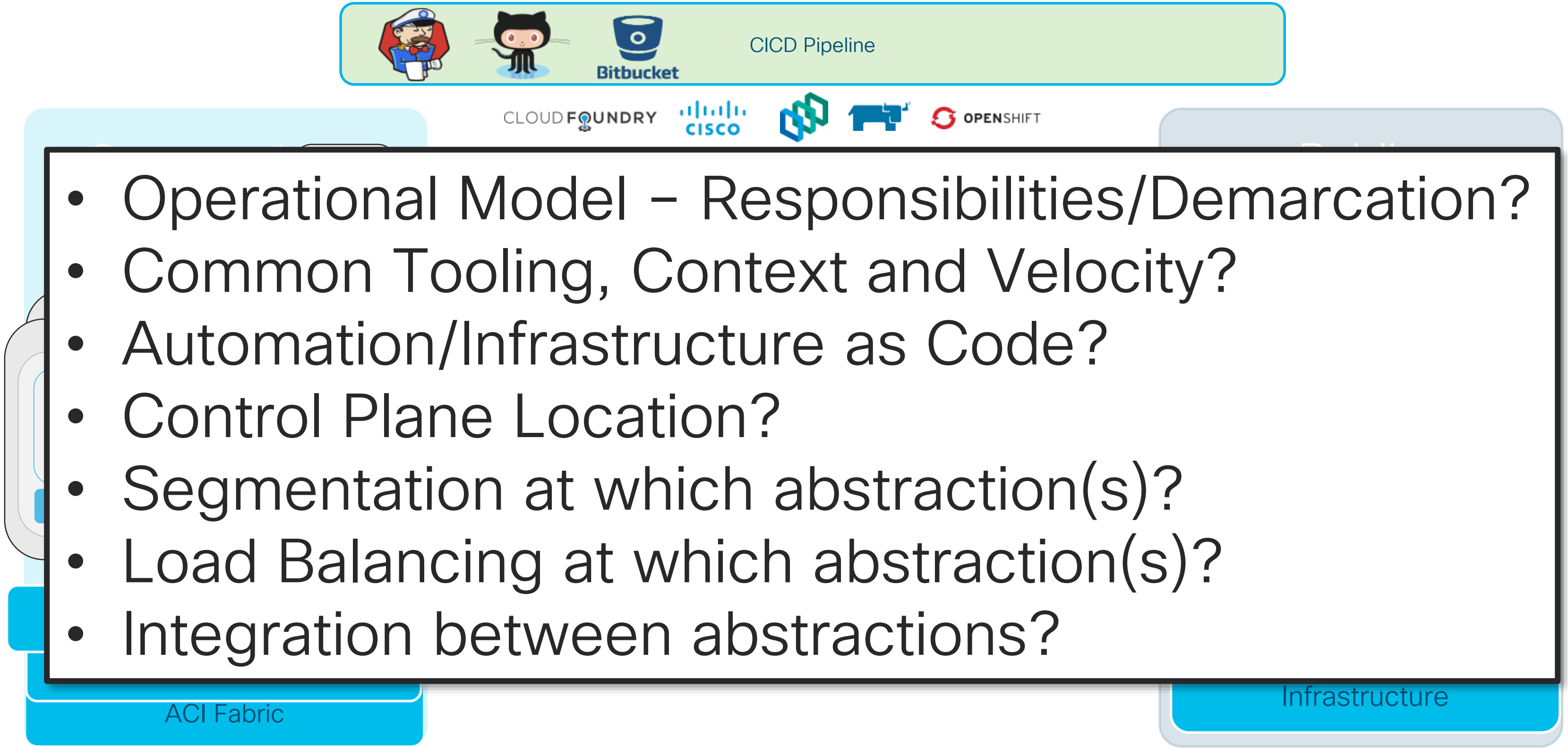
Physical Network

Public clouds

Delivering MyHero Everywhere



Delivering MyHero Everywhere



- Operational Model – Responsibilities/Demarcation?
- Common Tooling, Context and Velocity?
- Automation/Infrastructure as Code?
- Control Plane Location?
- Segmentation at which abstraction(s)?
- Load Balancing at which abstraction(s)?
- Integration between abstractions?

Conclusion

- Service Mesh will affect your day job
- You need to understand where it fits and integrates into your domain(s)
- Service Mesh may become your domain!?
- Embrace Kubernetes and Service Mesh in 2020

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco Showcase

Walk-In Labs

Meet the Engineer 1:1 meetings

Related sessions

Thank you

You make **possible**