

Traffic Routing with Istio on Oracle Cloud

Slides: <https://slides.peterj.dev>

Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Introduction

- I am Peter (@pjausovec)
- Software Engineer at Oracle
- Working on "cloud-native" stuff
- Books:
 - [Cloud Native: Using Containers, Functions, and Data to Build Next-Gen Apps](#)
 - SharePoint Development
 - VSTO For Dummies
- Courses:
 - Kubernetes Course (<https://startkubernetes.com>)
 - Istio Service Mesh Course (<https://learnistio.com>)

Docker?

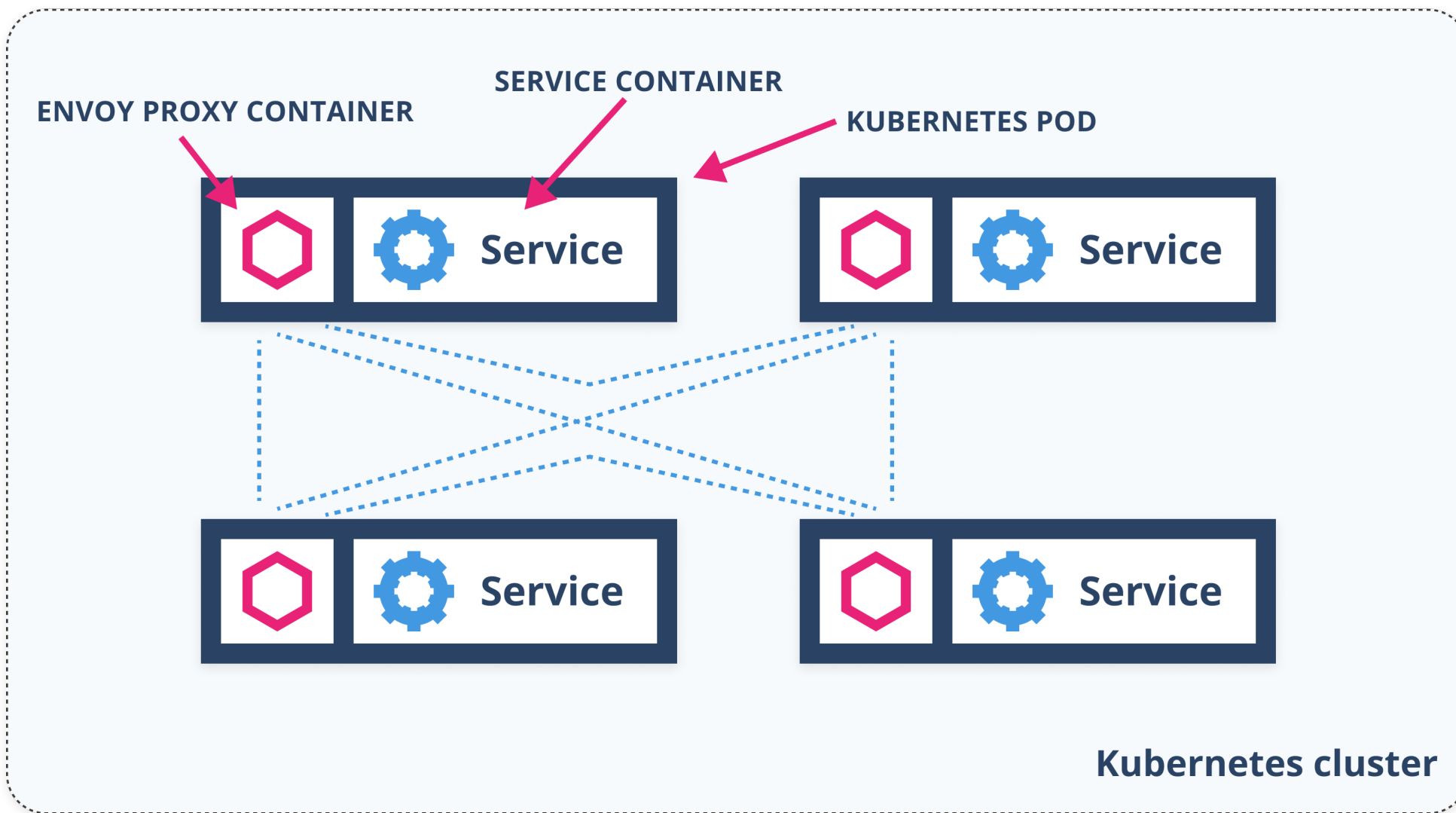
Kubernetes?

Service Mesh

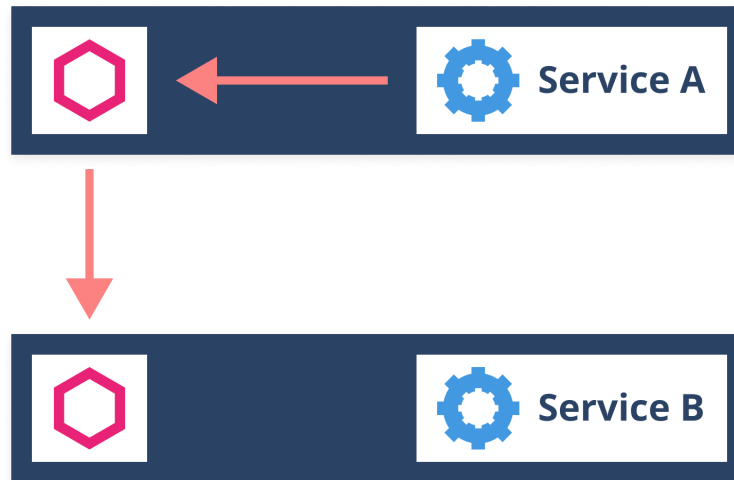
Dedicated infrastructure layer to connect, manage, and secure workloads by managing the communication between them

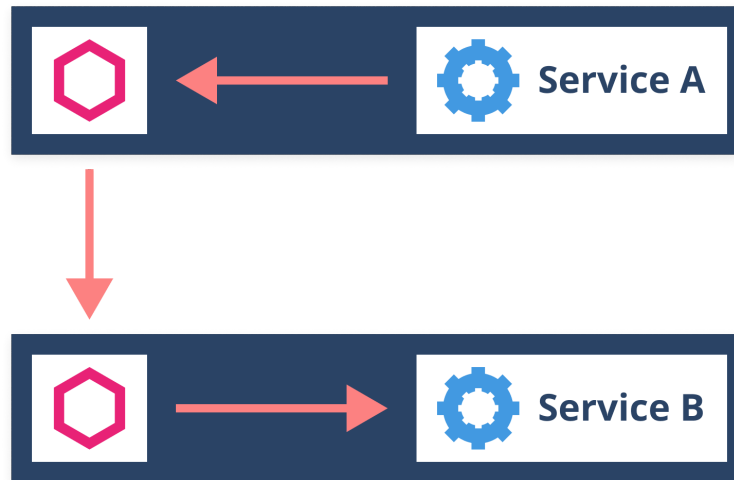
Istio service mesh

- Open source service mesh
 - Google, IBM, Lyft
- Well-defined API
- Can be deployed on-premise, in the cloud
 - Kubernetes
 - Mesos











Service Mesh - Architecture

Data plane (proxies)

- Run next to each service instance (or one per host)
 - Istio uses Envoy proxy
- Intercept all incoming/outgoing requests (iptables)
- Configure on how to handle traffic
- Emits metric

Control plane

- Validates rules
- Translates high-level rules to proxy configuration
- Updates the proxies/configuration
- Collects metrics from proxies



Service Mesh - Features

Connect

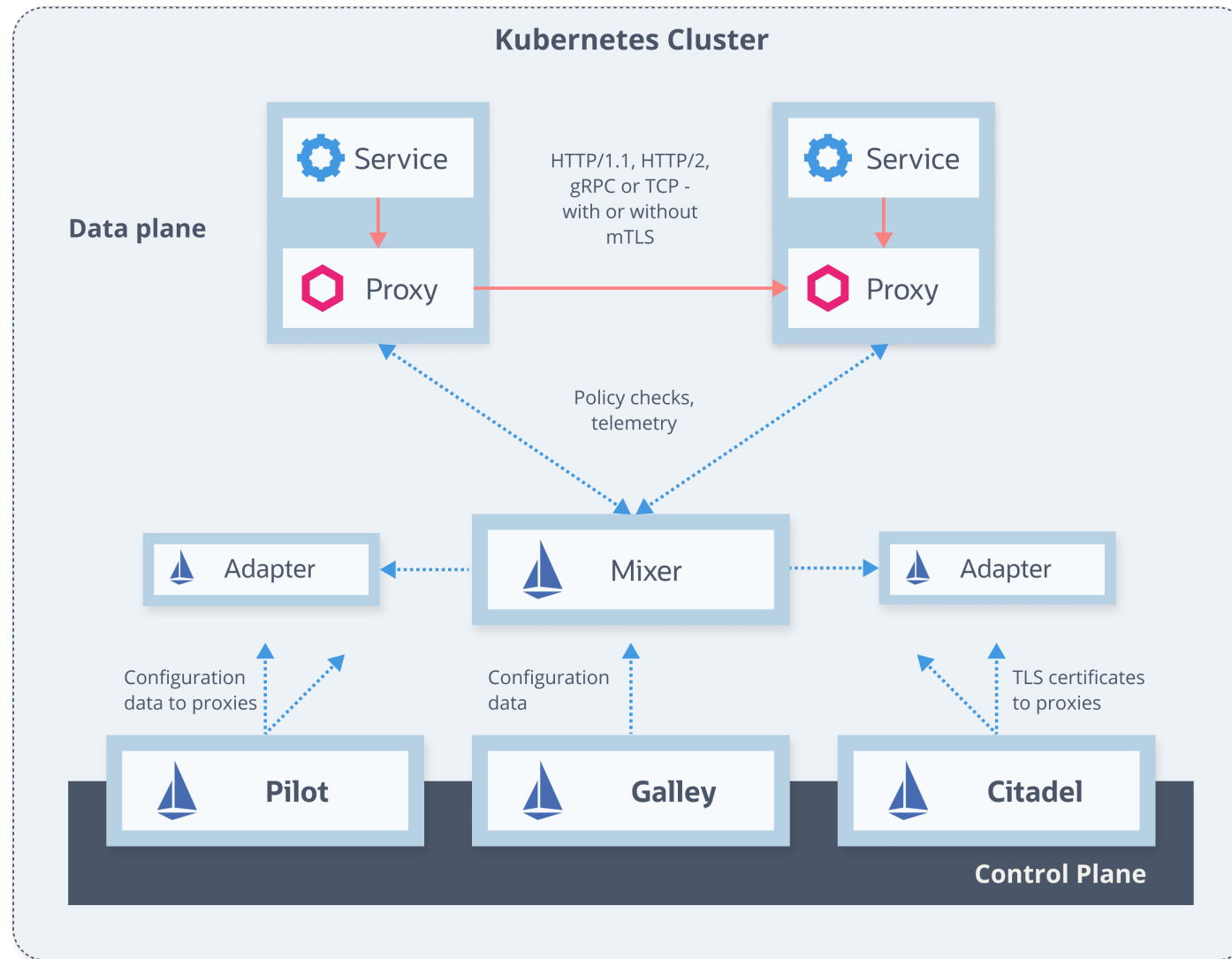
- Layer 7 routing and traffic management
 - %-based traffic split (URIs, header, scheme, method, ...)
 - Circuit breakers, timeouts and retries

Manage

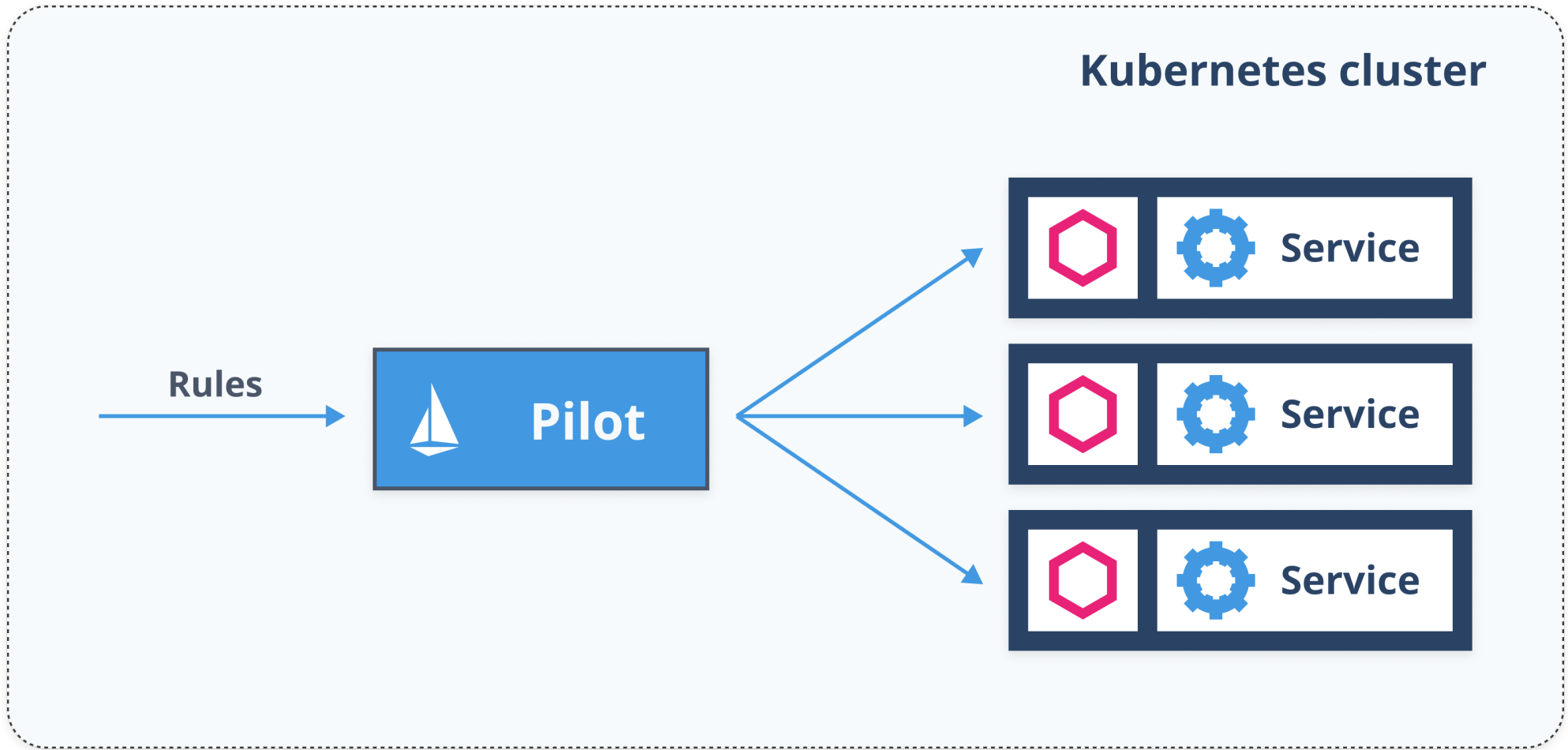
- Telemetry (proxies collect metrics automatically -> tools: Grafana, Jaeger, Kiali)
- Visibility into service communication without code changes

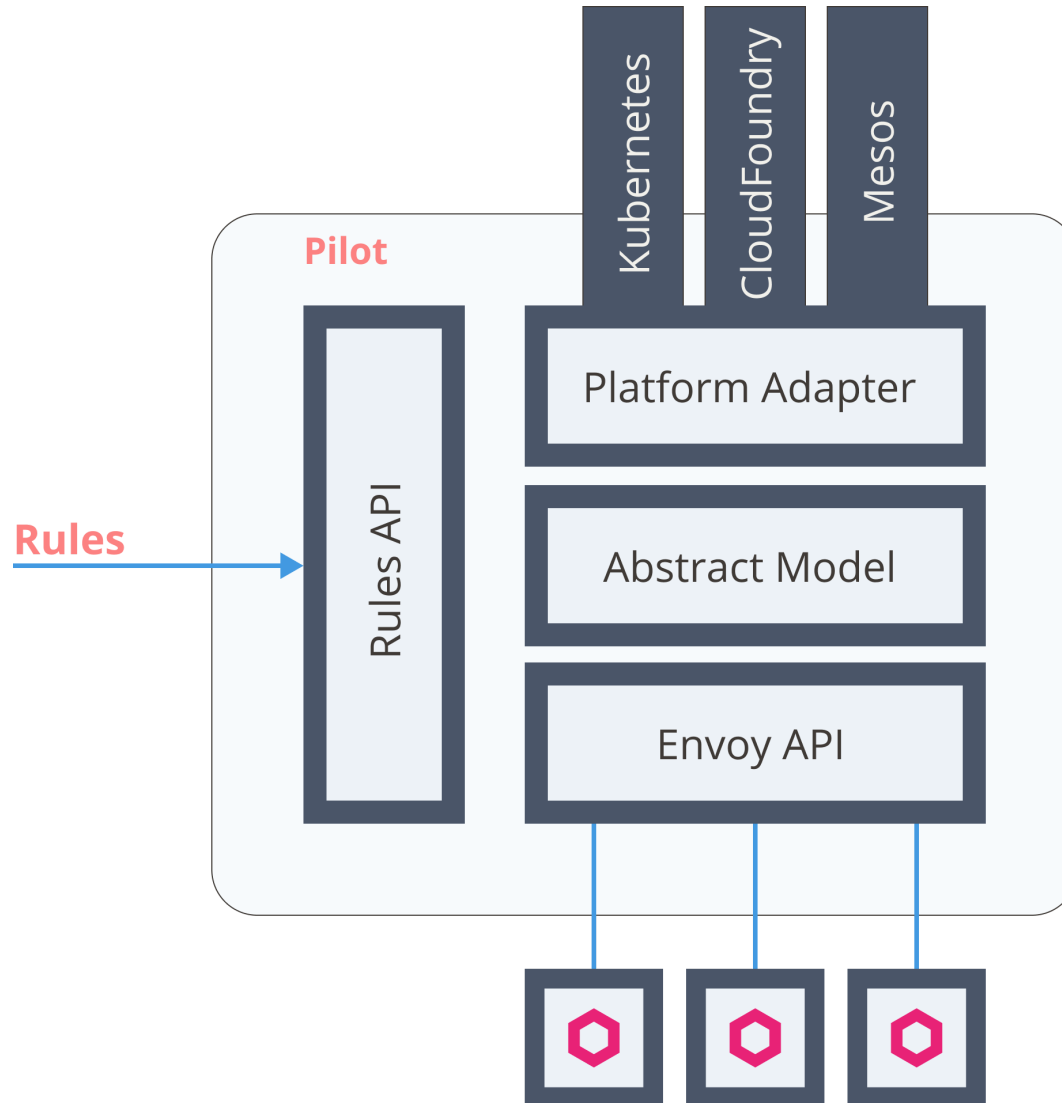
Secure

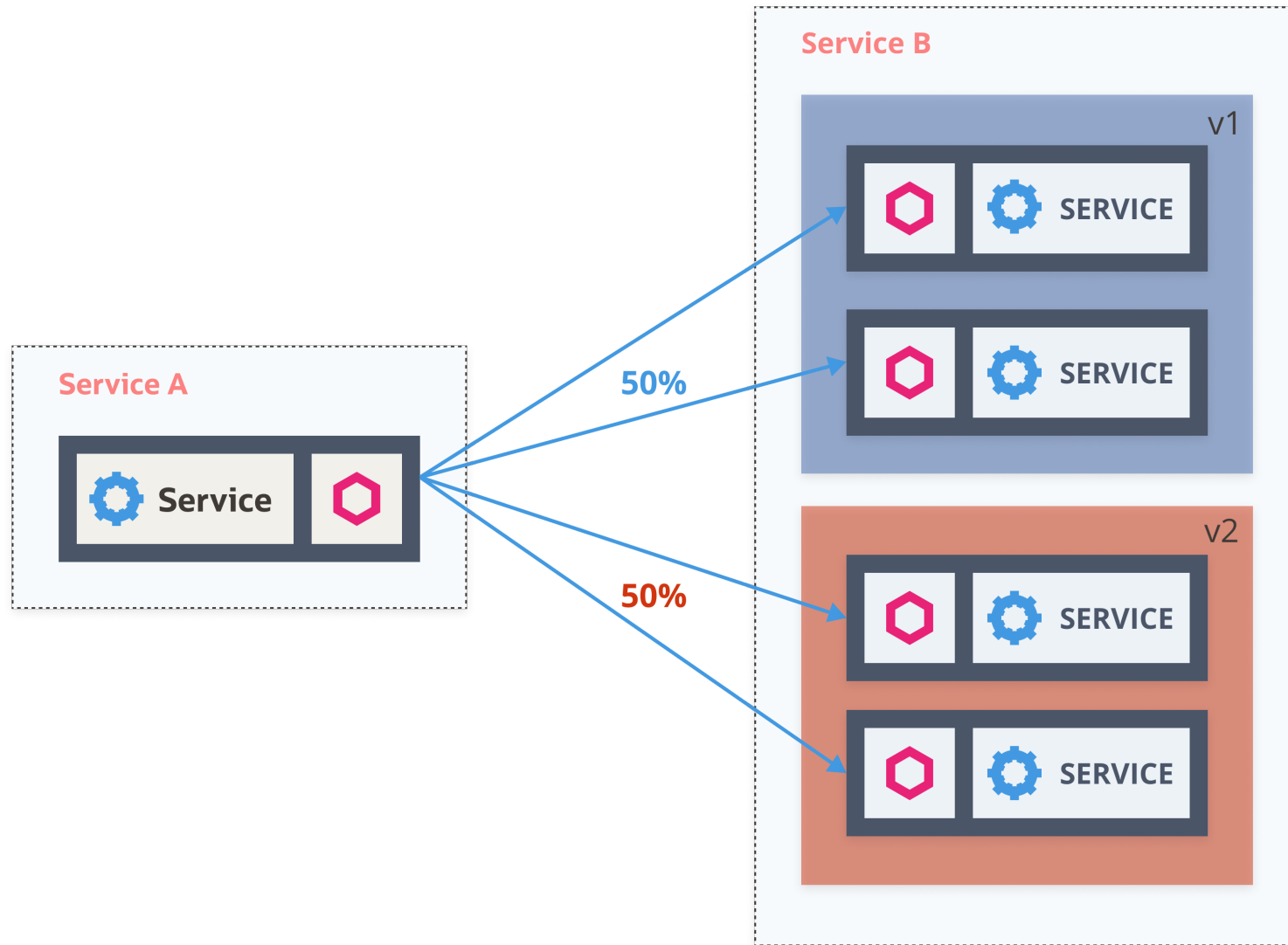
- Secure communication between services (mutual TLS)
- Identity + cert for each service



Traffic Management







Traffic Management Scenarios

Canary

- "5% of the traffic should go to **v2** and 95% of the traffic to **v1**"

A/B tests

- "Is User-Agent header == Firefox, direct traffic to v2 version"
- "Any requests to URL /api/user should be redirected to /api/v1/user"
- "All GET requests to /api/v2 all POST requests to /api/v1"
- ...

Traffic Management Scenarios

Ingress Gateway

- Route domains/subdomains to different services
 - <https://hello.example.com> → hello-svc
 - <https://blog.example.com> → blog-svc
 - <https://other-domain.com> → some-service

Egress Gateway/Service Entry

- Control outgoing traffic/external services

Mirroring/Shadowing

- Duplicate incoming requests and send them to different service

Service Mesh - Istio

Traffic Management Resources

- Gateway
- VirtualService
- DestinationRule
- ServiceEntry
- Sidecar

Service Mesh - Virtual Service

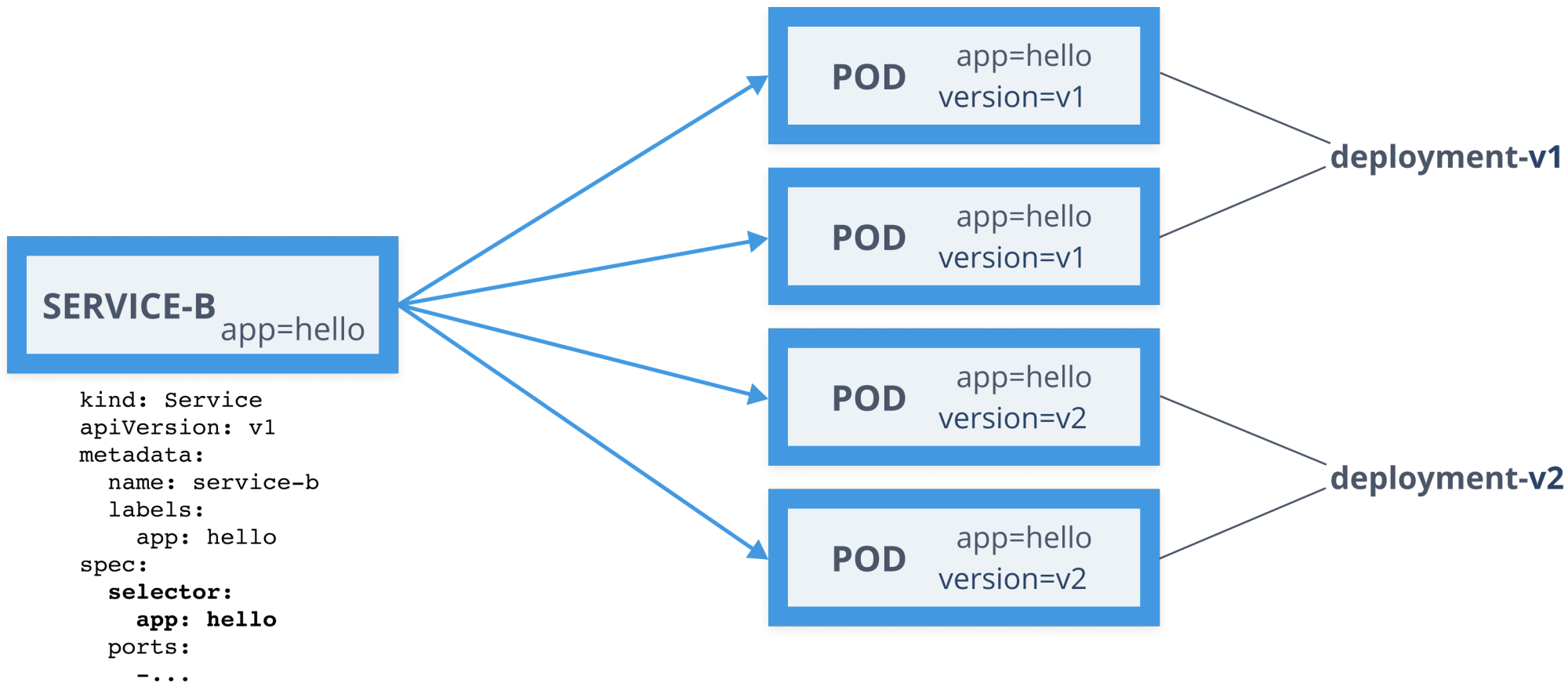
```
1 apiVersion: networking.istio.io/v1alpha3
2 kind: VirtualService
3 metadata:
4   name: serviceb-vs
5 spec:
6   hosts:
7     - service-b.default.svc.cluster.local
8   http:
9     - route:
10       - destination:
11         host: service-b.default.svc.cluster.local
12         subset: v1
13         weight: 98
14       - destination:
15         host: service-b.default.svc.cluster.local
16         subset: v2
17         weight: 2
```

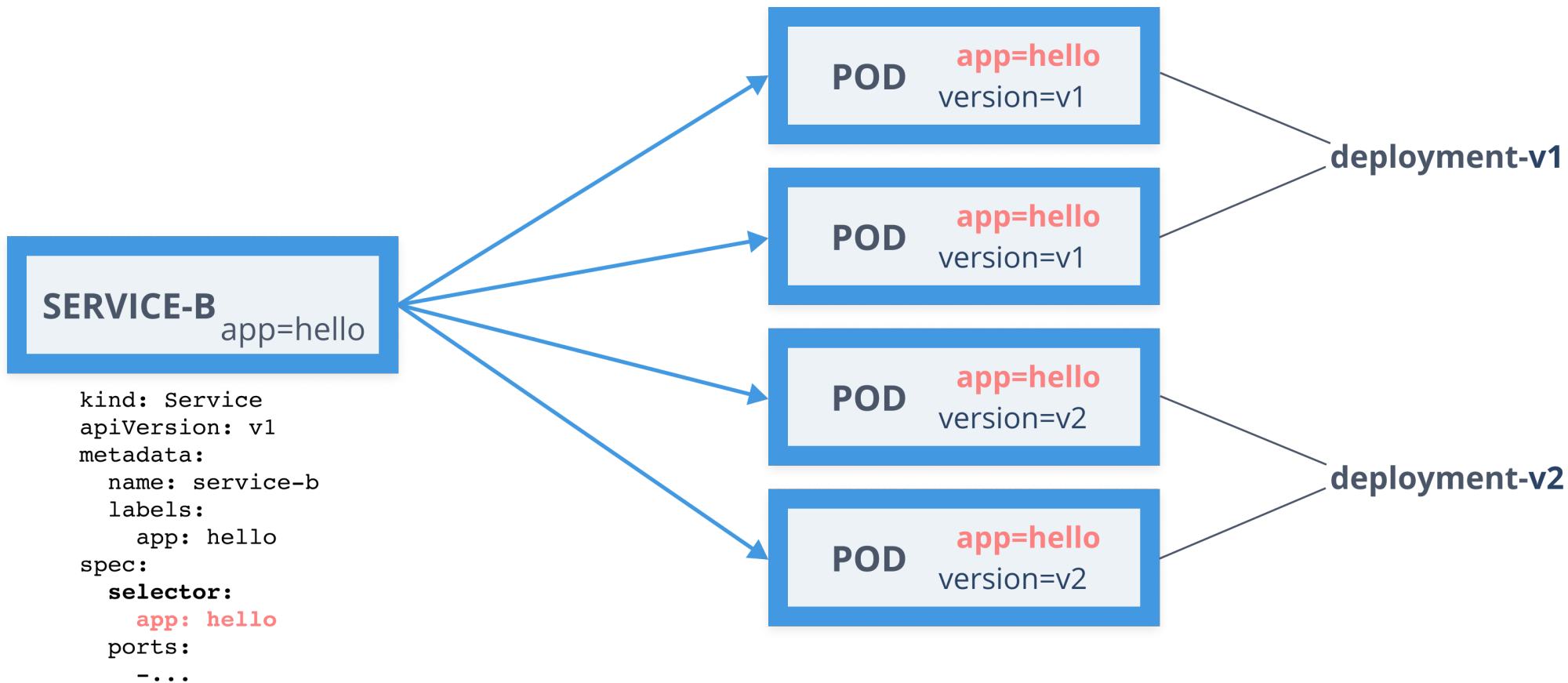


Service Mesh - Destination Rule

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: serviceb-dr
spec:
  host: service-b.default.svc.cluster.local
  subsets:
    - name: v1
      labels:
        version: v1
    - name: v2
      labels:
        version: v2
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```







Destination rule

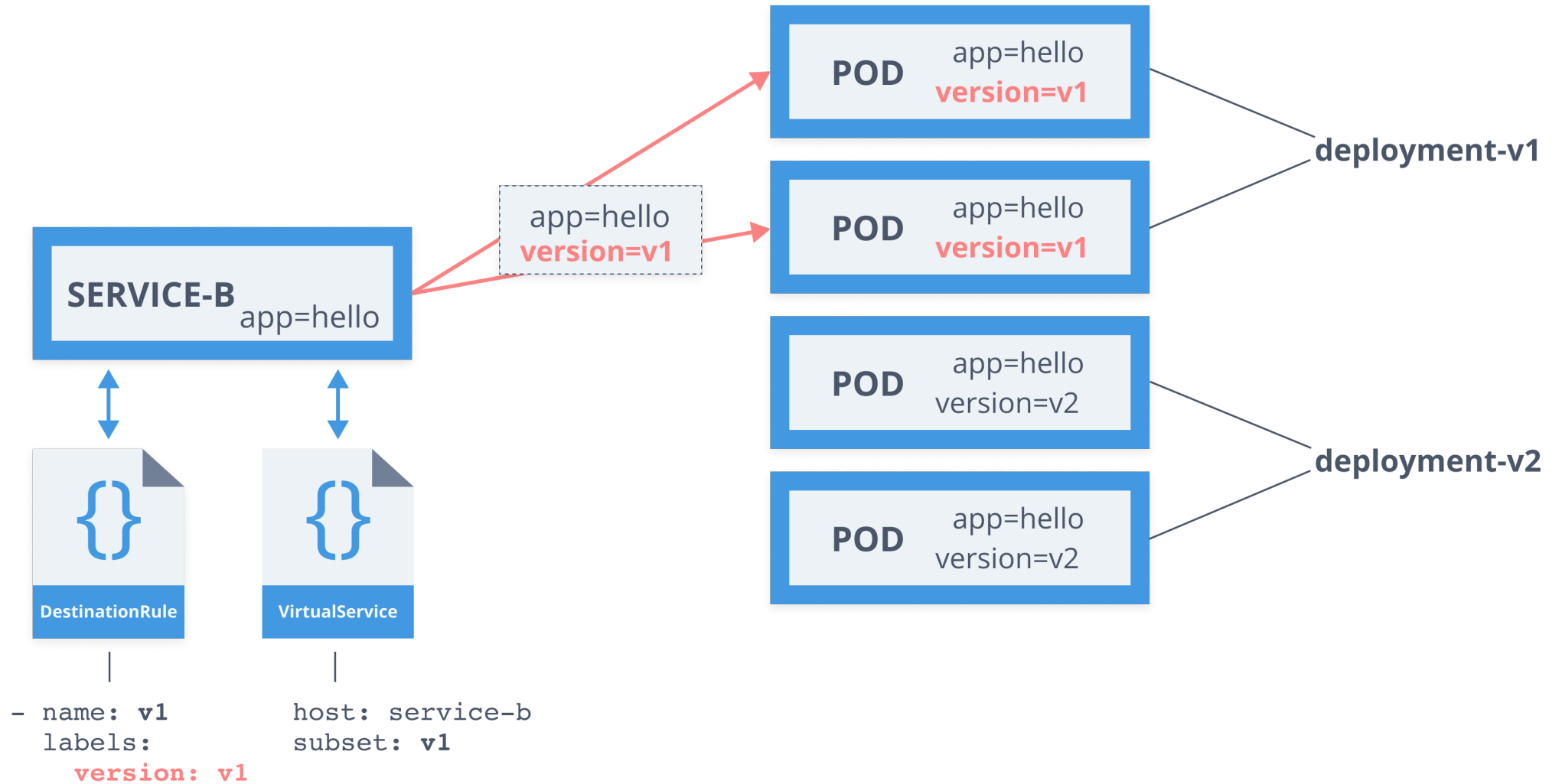
```
1 apiVersion: networking.istio.io/v1alpha3
2 kind: DestinationRule
3 metadata:
4   name: serviceb-dr
5 spec:
6   host: service-b.default.svc.cluster.local
7   subsets:
8     - name: v1
9       labels:
10        version: v1
11     - name: v2
12       labels:
13        version: v2
```

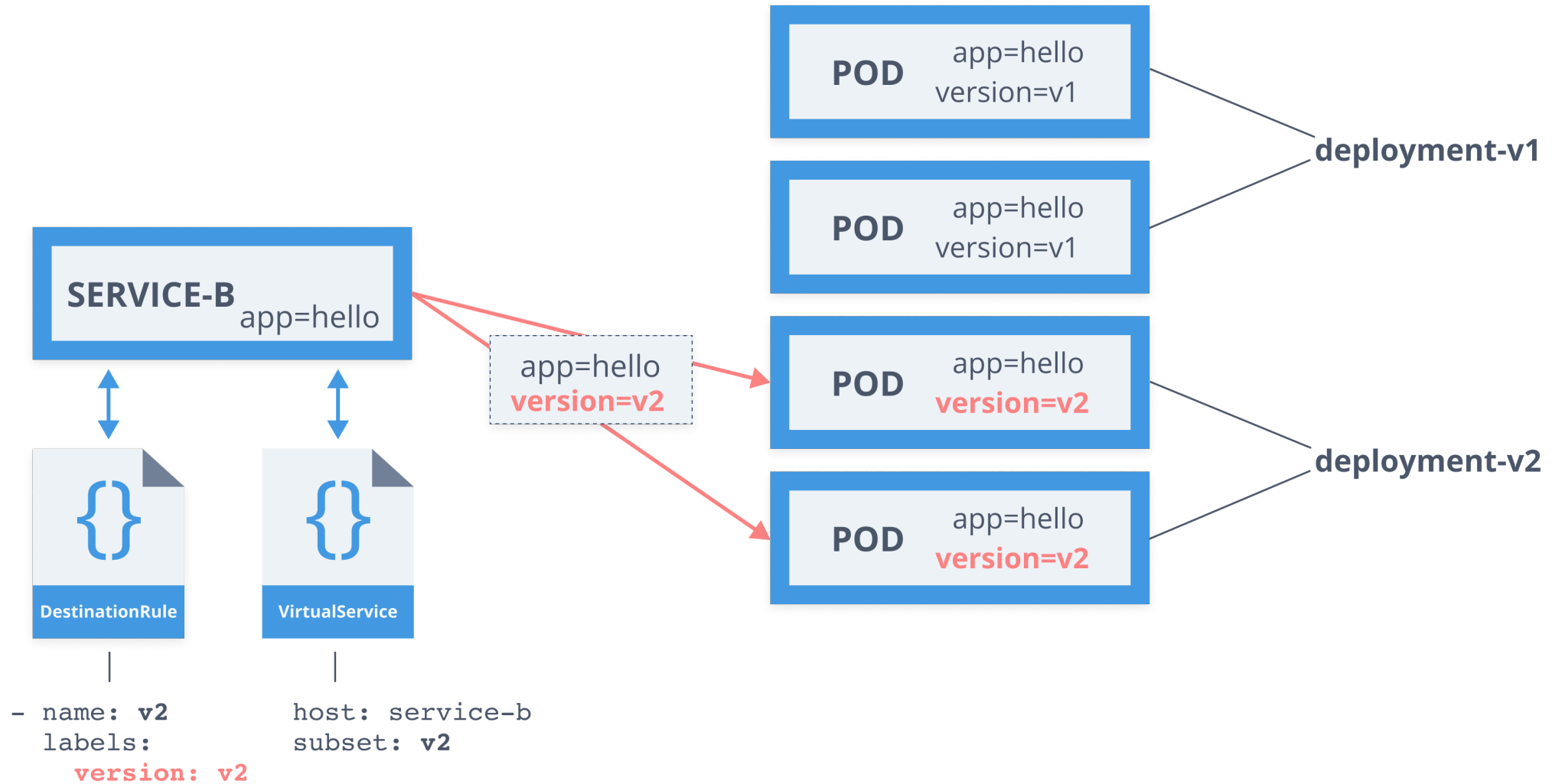


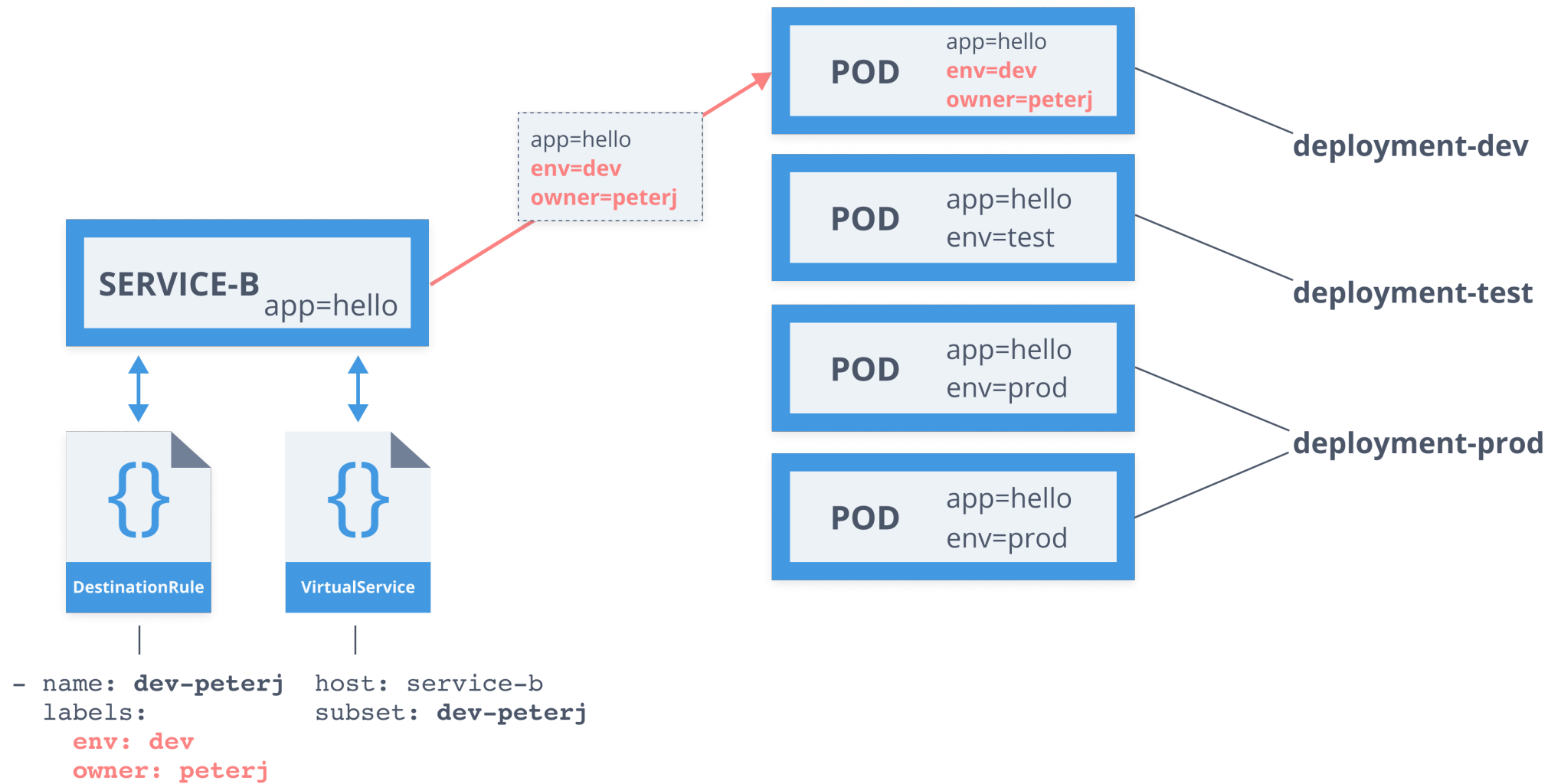
Virtual service

```
1 | ...  
2 | http:  
3 |   - route:  
4 |     - destination:  
5 |       host: service-b.default.svc.cluster.local  
6 |       subset: v1  
7 |       weight: 30
```









Service Mesh - Service Entry

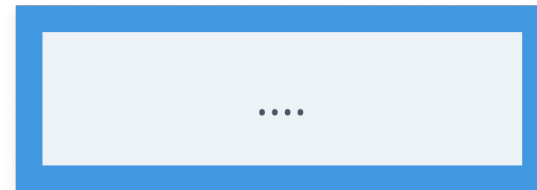
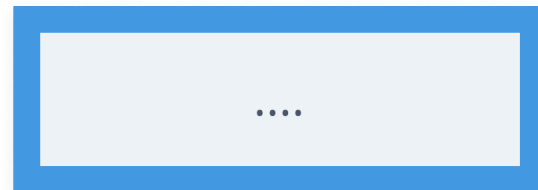
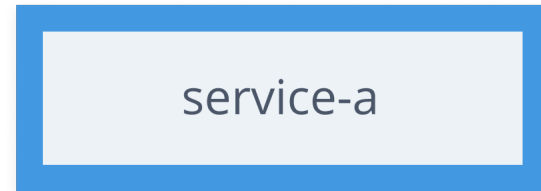
```
1 apiVersion: networking.istio.io/v1alpha3
2 kind: ServiceEntry
3 metadata:
4   name: movie-db
5 spec:
6   hosts:
7     - api.themoviedb.org
8   ports:
9     - number: 443
10      name: https
11      protocol: HTTPS
12   resolution: DNS
13   location: MESH_EXTERNAL
```

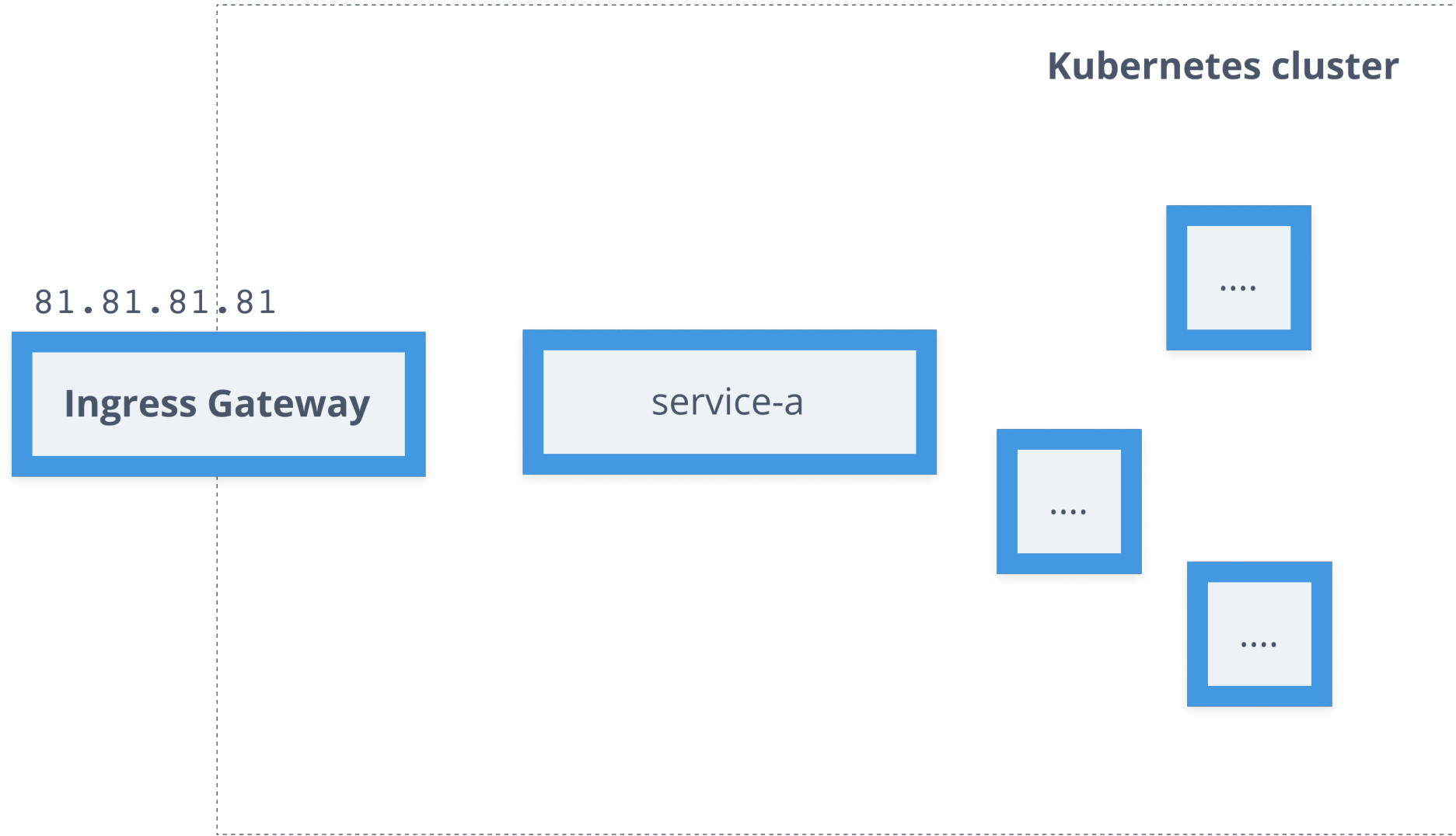
Service Mesh - Gateway

```
1 apiVersion: networking.istio.io/v1alpha3
2 kind: Gateway
3 metadata:
4   name: gateway
5 spec:
6   selector:
7     istio: ingressgateway
8   servers:
9     - port:
10       number: 80
11       name: http
12       protocol: HTTP
13     hosts:
14       - "hello.example.com"
```



Kubernetes cluster





81.81.81.81

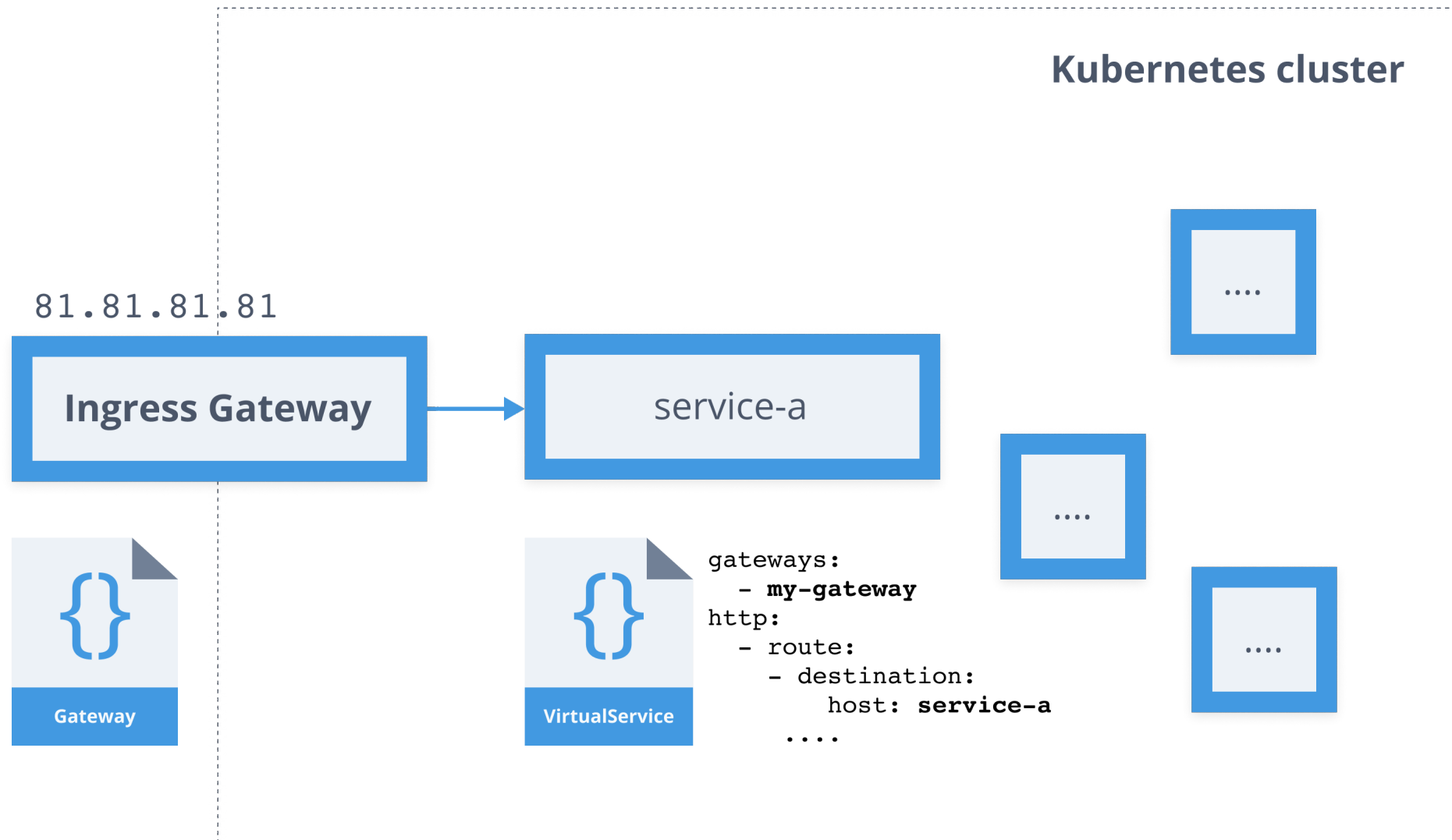
Ingress Gateway

Domain Registrar - DNS settings

CNAME: **www.service-a.com**

A: **81.81.81.81**

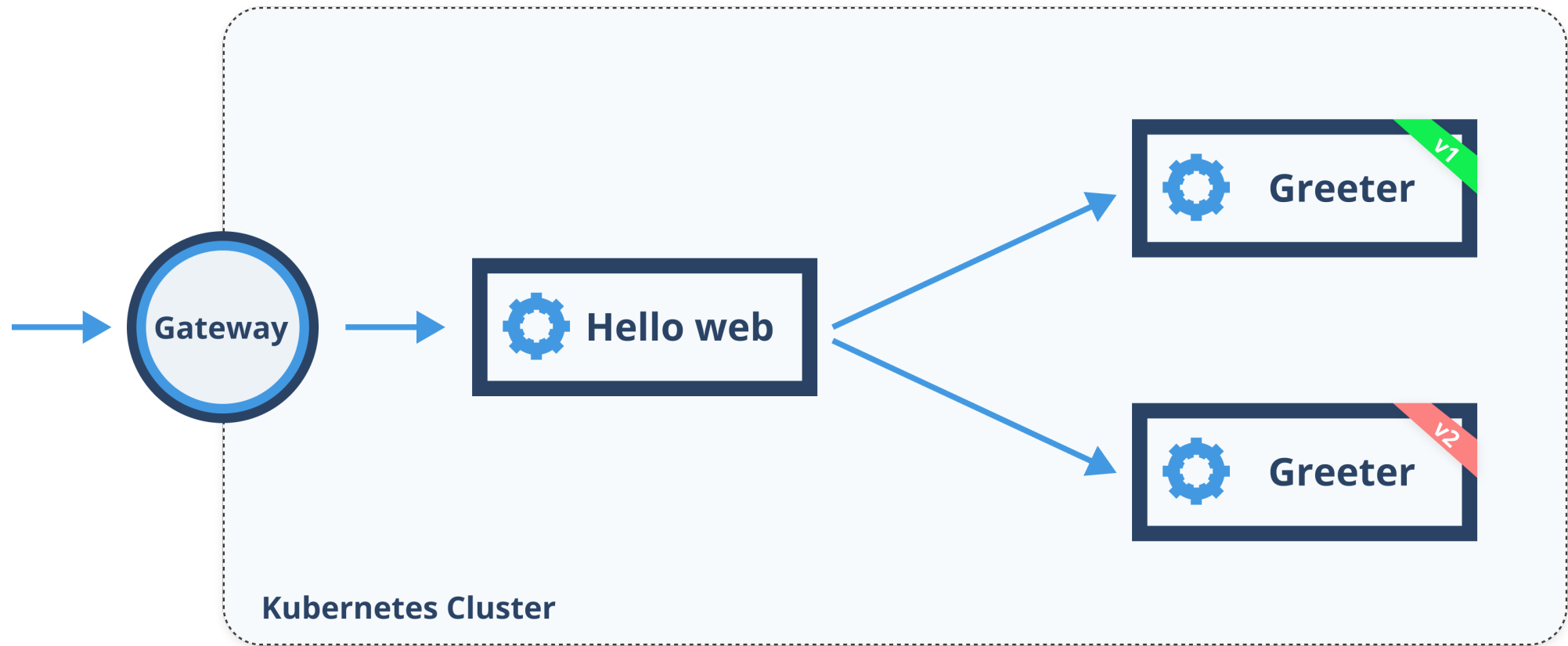
```
servers:  
  - port:  
      number: 80  
      name: http  
      protocol: HTTP  
hosts:  
  - "www.service-a.com"
```



Service Mesh - Sidecar

```
1 apiVersion: networking.istio.io/v1alpha3
2 kind: Sidecar
3 metadata:
4   name: default
5   namespace: prod-us-west-1
6 spec:
7   egress:
8     - hosts:
9       - 'prod-us-west-1/*'
10      - 'prod-apis/*'
11      - 'istio-system/*'
```





Resources

- Kubernetes on Oracle Cloud (OKE) - (<https://cloud.oracle.com>)
- Kubernetes - (<https://kubernetes.io>)
- Istio - (<https://istio.io>)

Oracle Microservices Example

- MuShop - <https://github.com/oracle-quickstart/oci-cloudnative>



Thank you

Slides: <https://slides.peterj.dev>

Contact

- [@pjausovec](#)
- <https://peterj.dev>

Table of Contents

- [Introduction](#)
- [Service Mesh](#)
- [Traffic Management](#)
- [Resources](#)