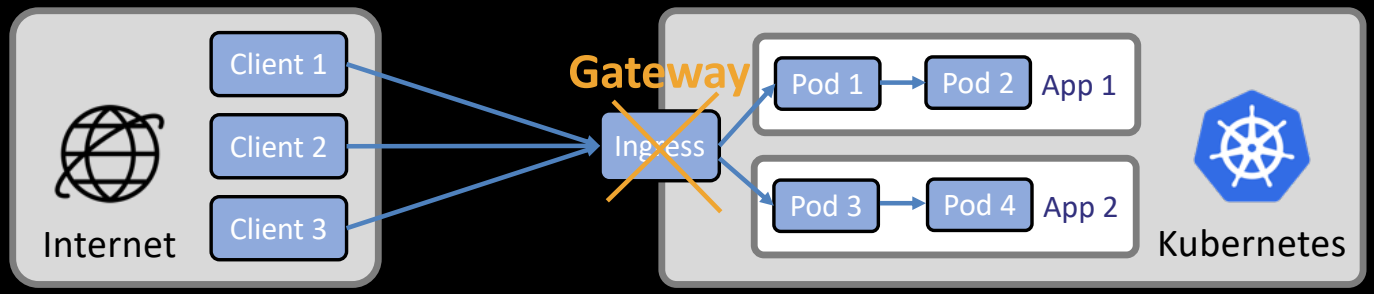
KUBERNETES GATEWAY API

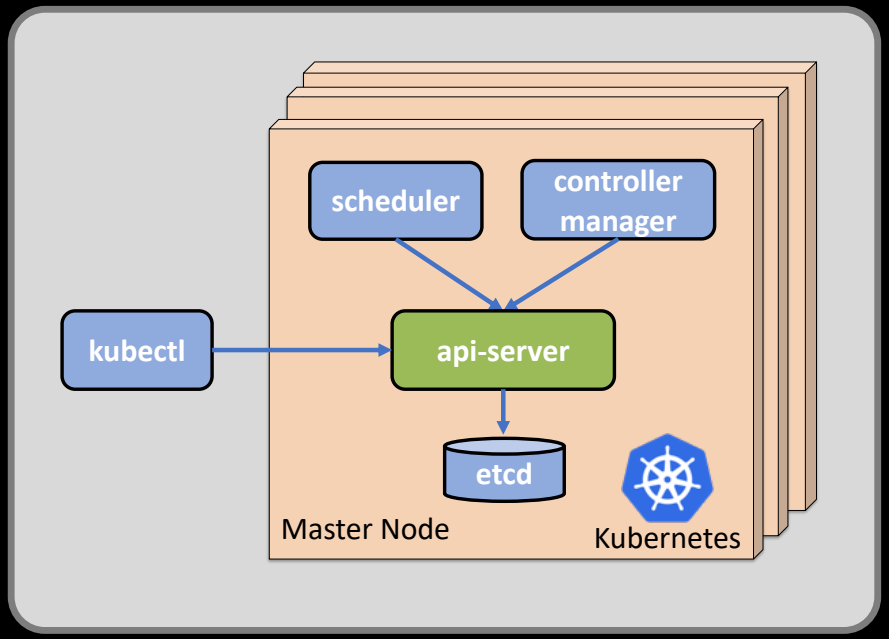


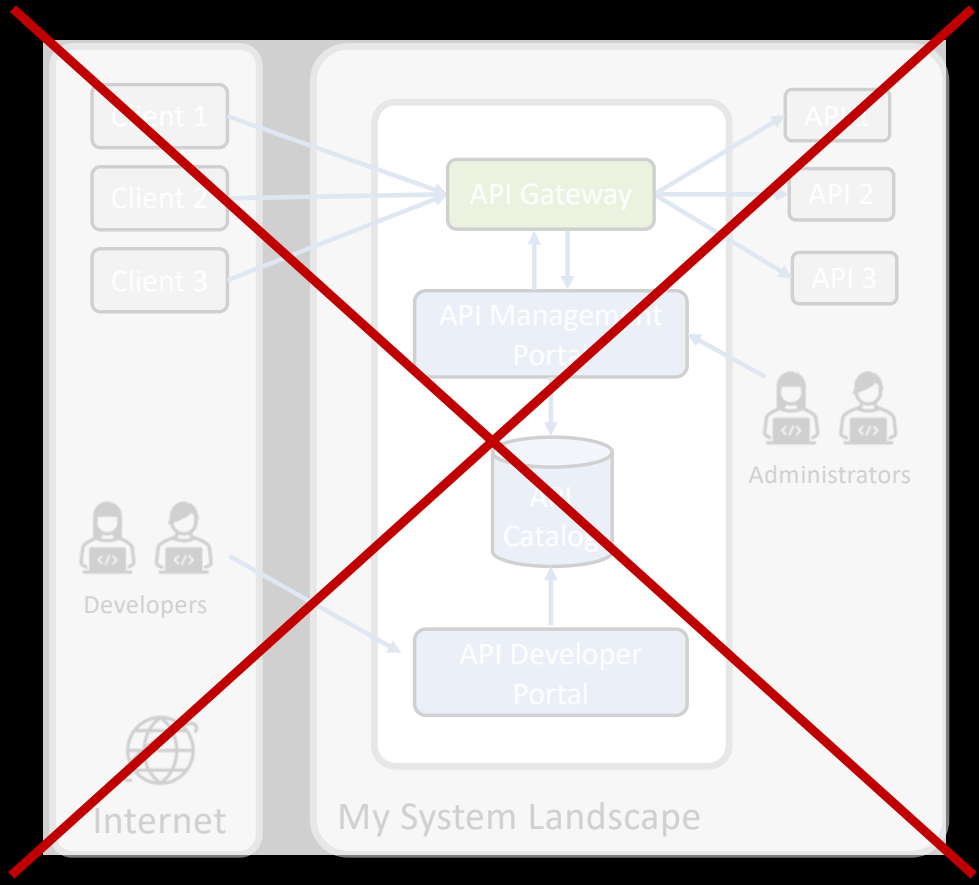
MAGNUS LARSSON

CADEC 2024.01.18 & 2024.01.24 | CALLISTAENTERPRISE.SE

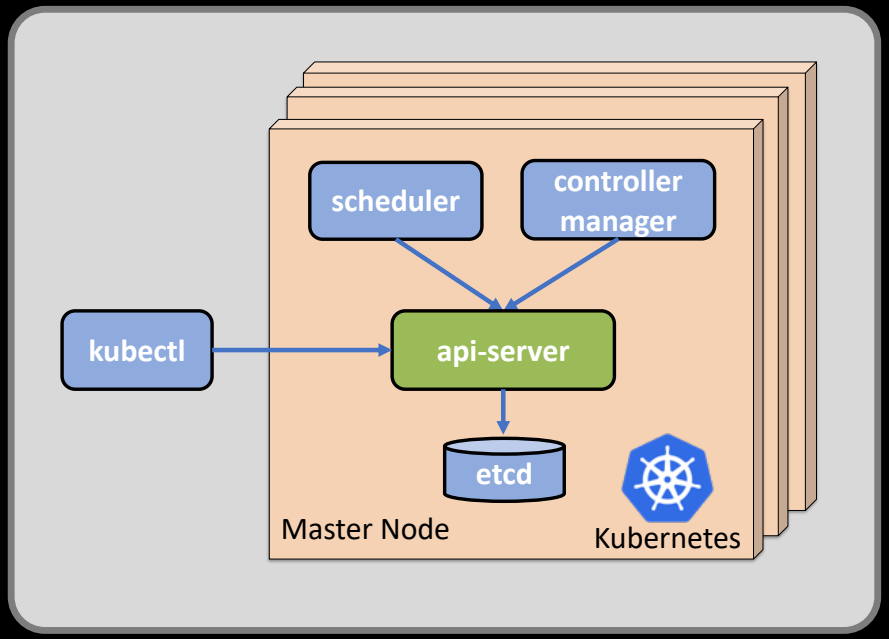
CALLISTA

Gateway API != API Gateway





Gateway API != API Gateway

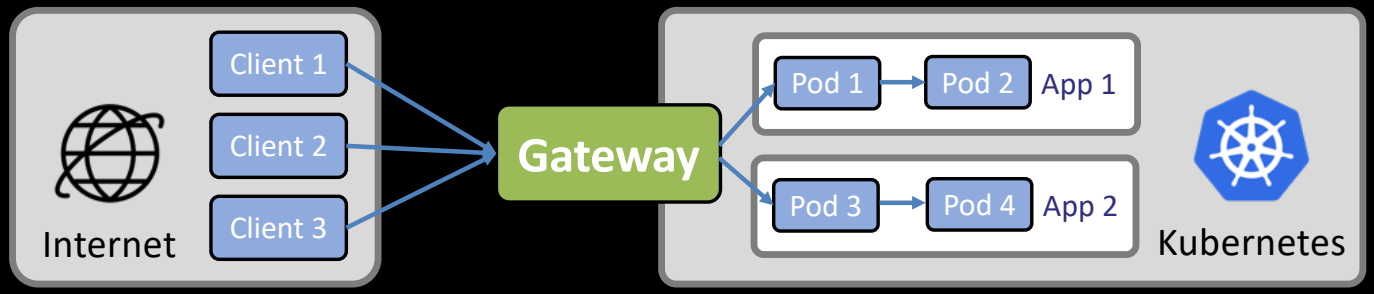



```yaml
apiVersion: apps/v1
kind: Deployment
```

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
```

Gateway API != API Gateway

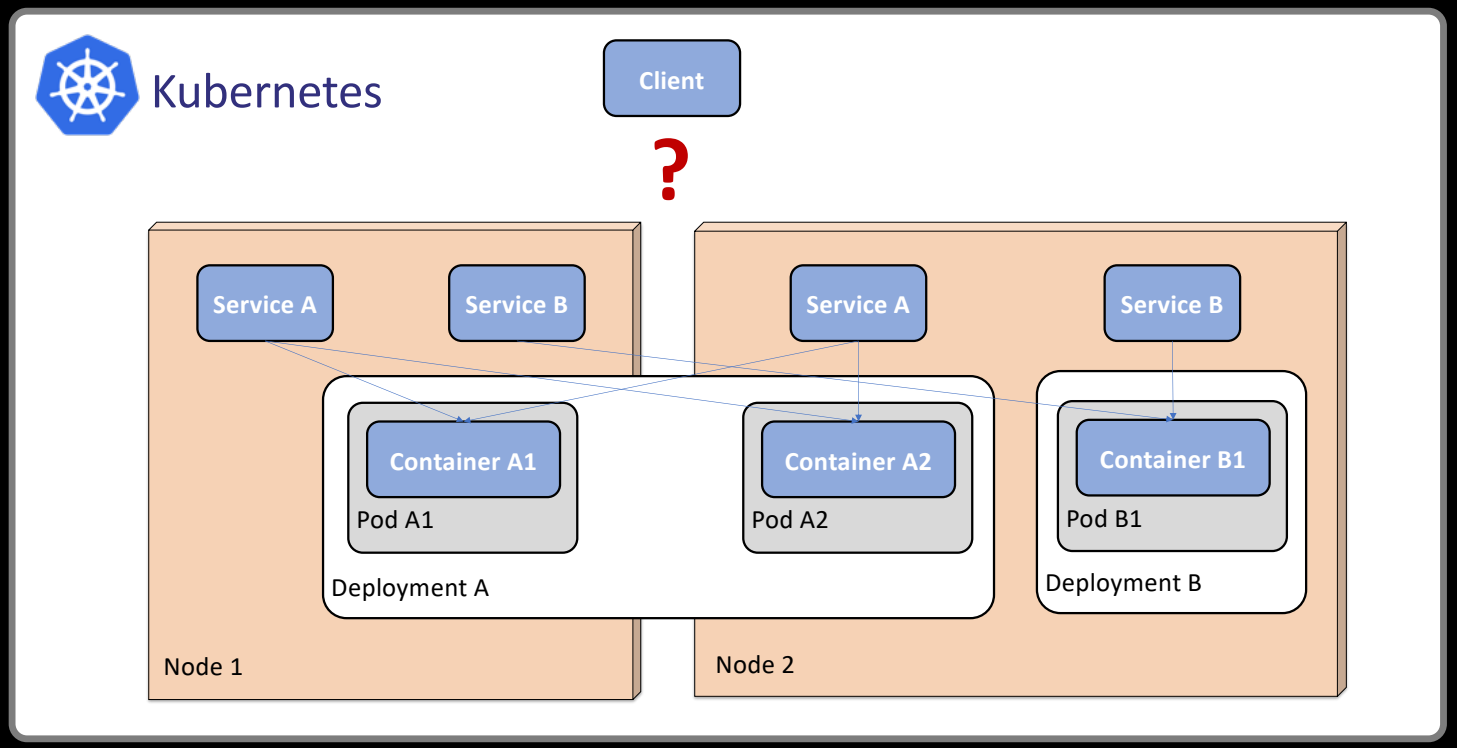


AGENDA

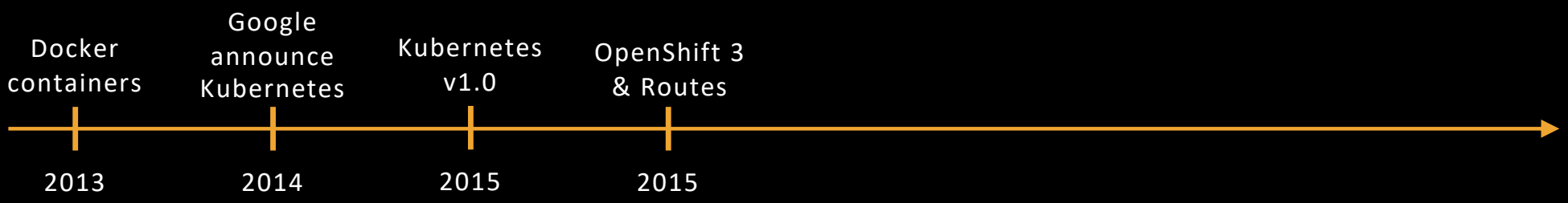
- Recap on Kubernetes Ingress
  - …and its limitations
- Introducing the Gateway API
- Trying it out
- Ready for production?
- Summary

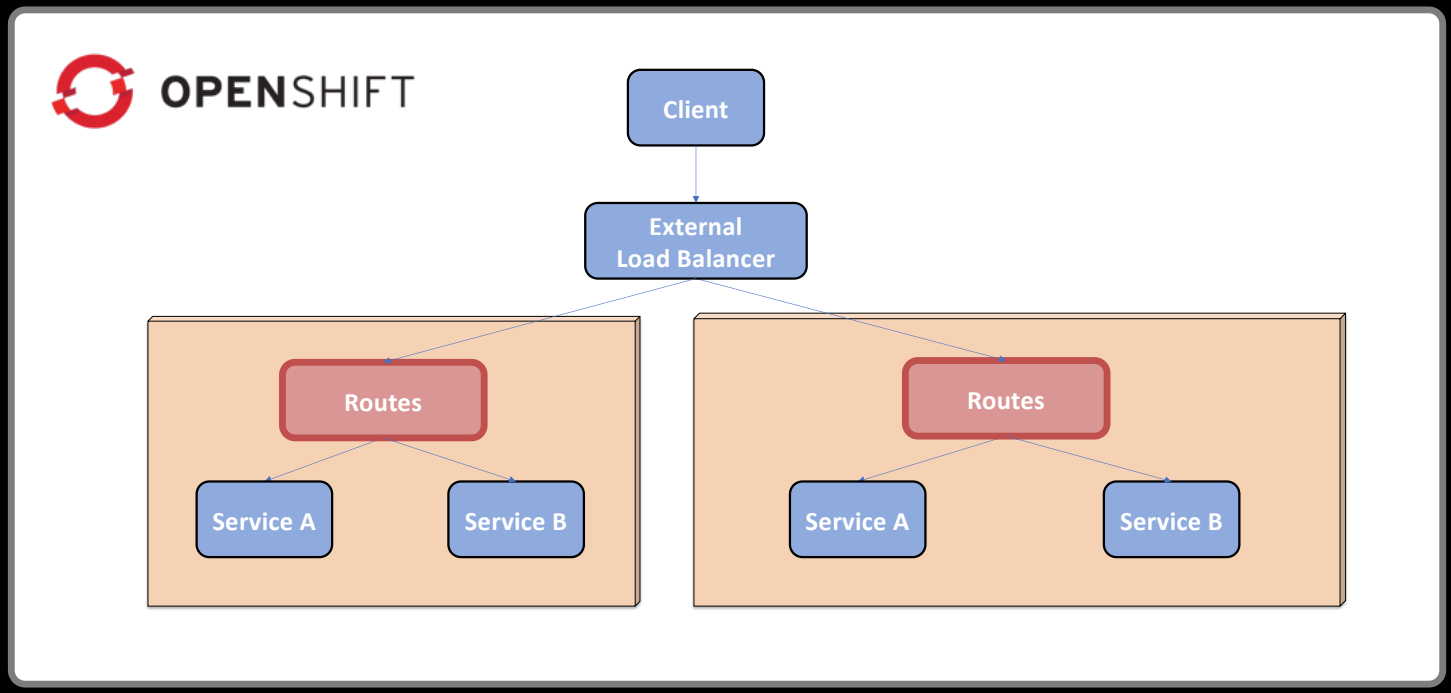
TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES

TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES

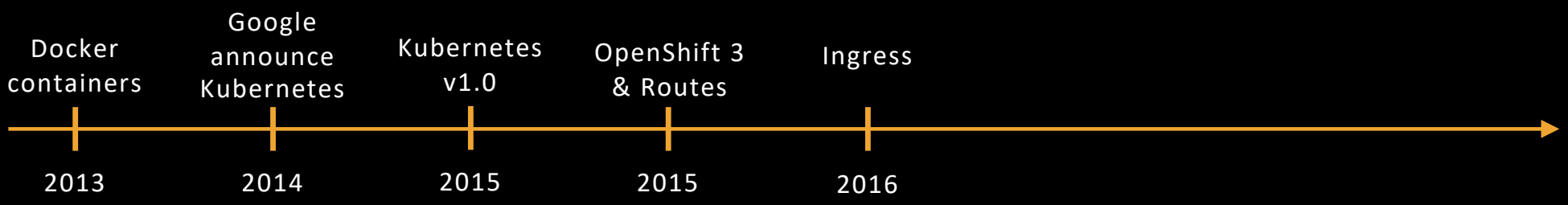


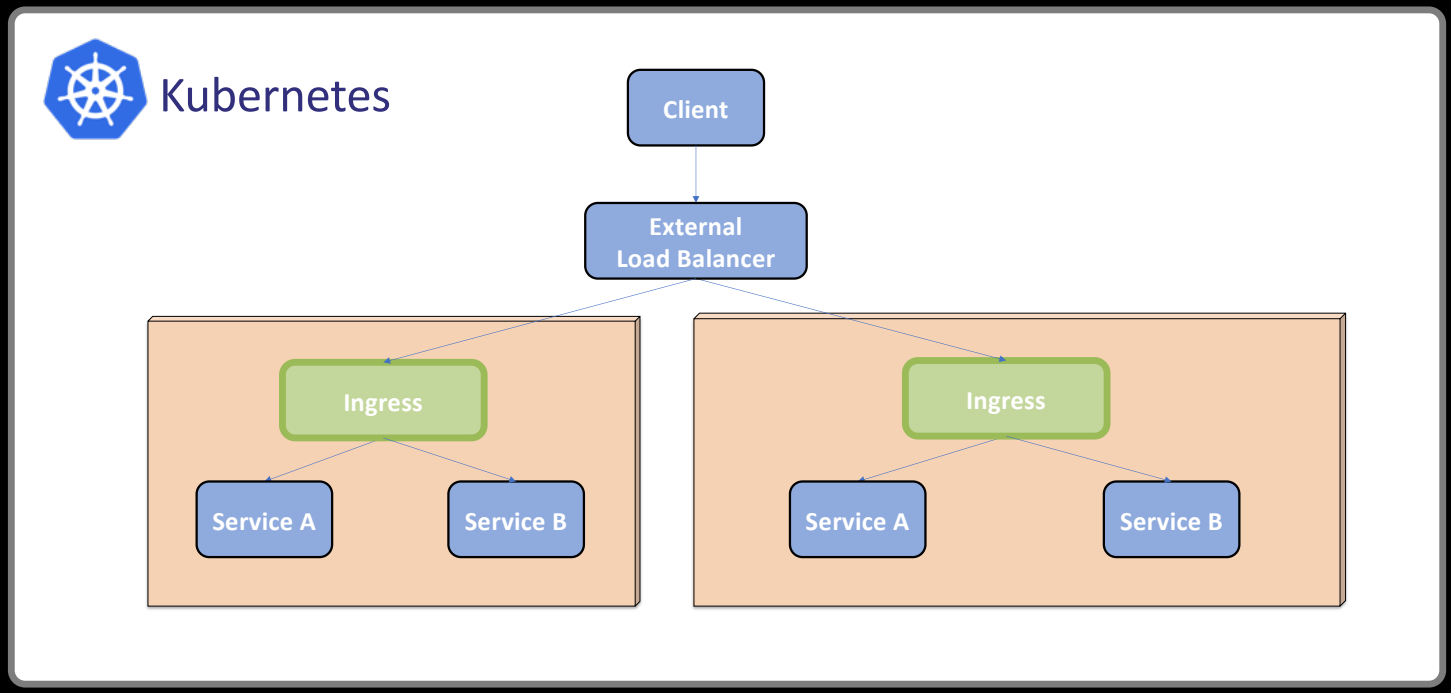
TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES



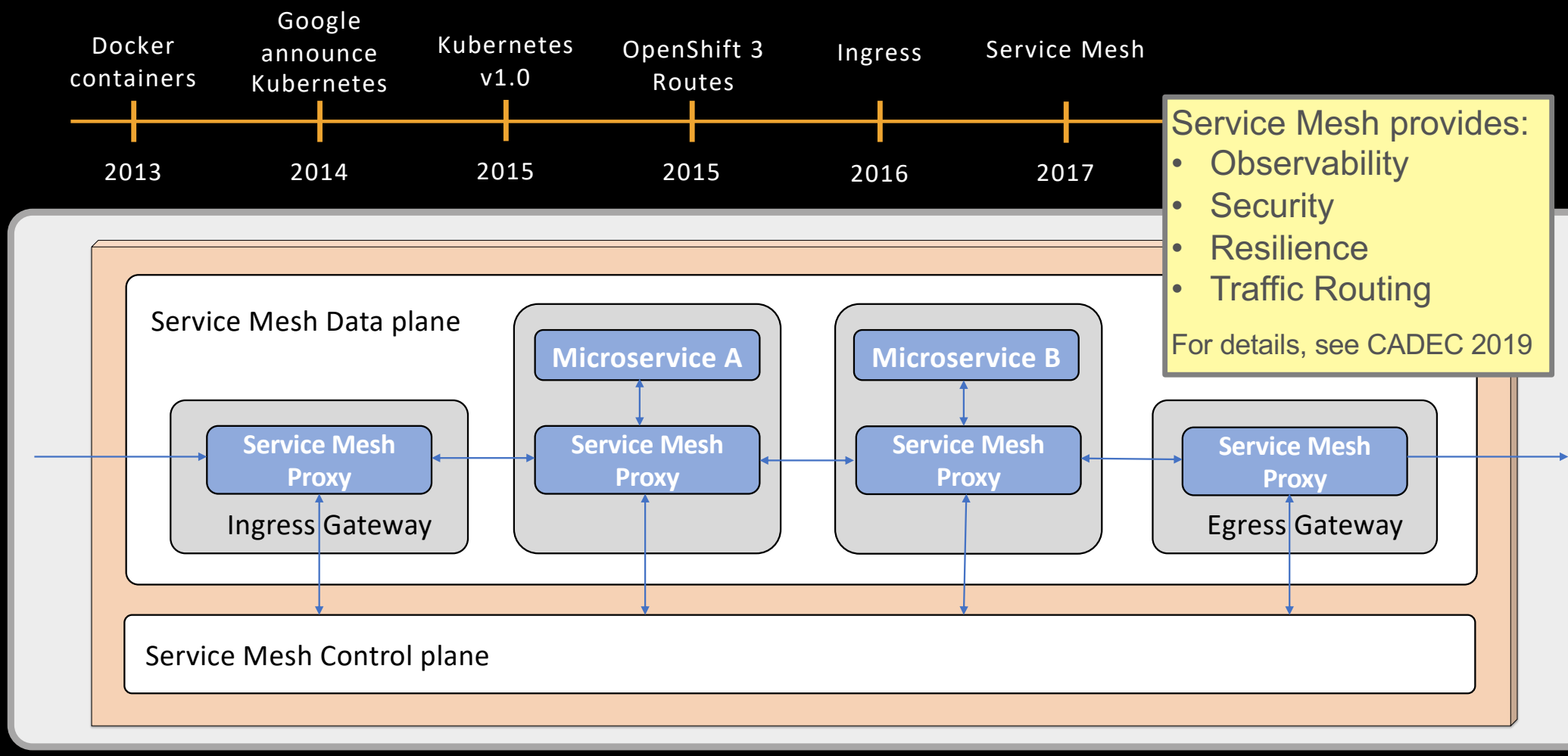


TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES

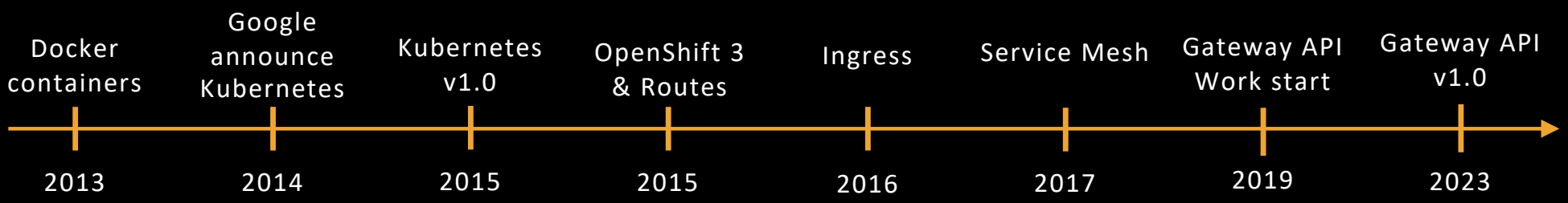




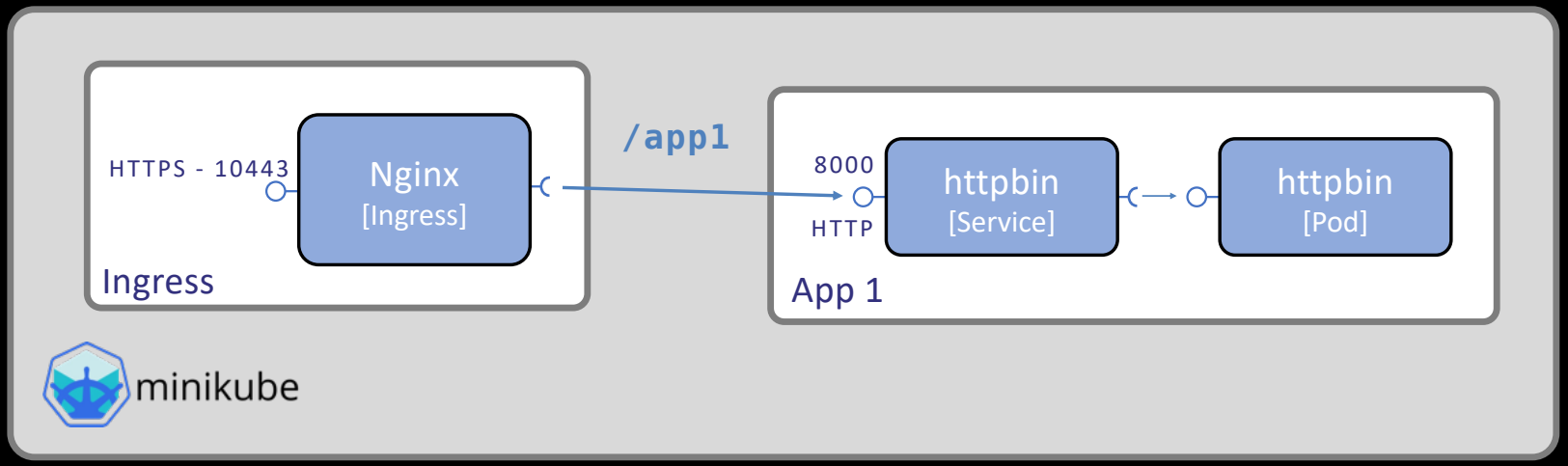
TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES

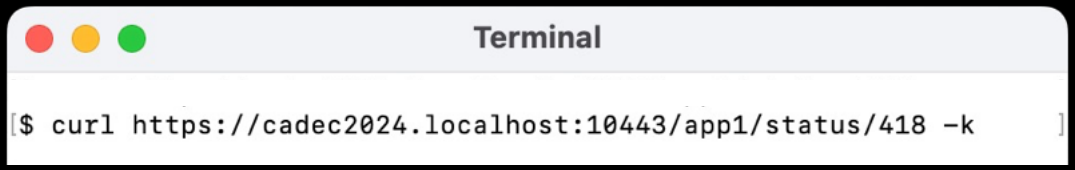


TIMELINE – EVOLUTION OF INGRESS CAPABILITIES IN KUBERNETES



INGRESS – AN EXAMPLE





INGRESS – AN EXAMPLE

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

INGRESS LIMITATIONS

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

INGRESS LIMITATIONS

1. Only supports HTTP and HTTPS

   E.g. TCP requires vendor-specific extension

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

INGRESS LIMITATIONS

2. Vendor specific annotation to support URL rewrite

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

INGRESS LIMITATIONS

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

App Dev/Ops

3.Separation of concerns

INGRESS LIMITATIONS

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

Cluster Op

App Dev/Ops

Cluster Op

3.Separation of concerns

INGRESS LIMITATIONS

4. Lack of support for a Service Mesh

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

INGRESS LIMITATIONS

4. Lack of support for a Service Mesh

2. Vendor specific annotation to support URL rewrite

1. Only supports HTTP and HTTPS

   E.g. TCP requires vendor-specific extension

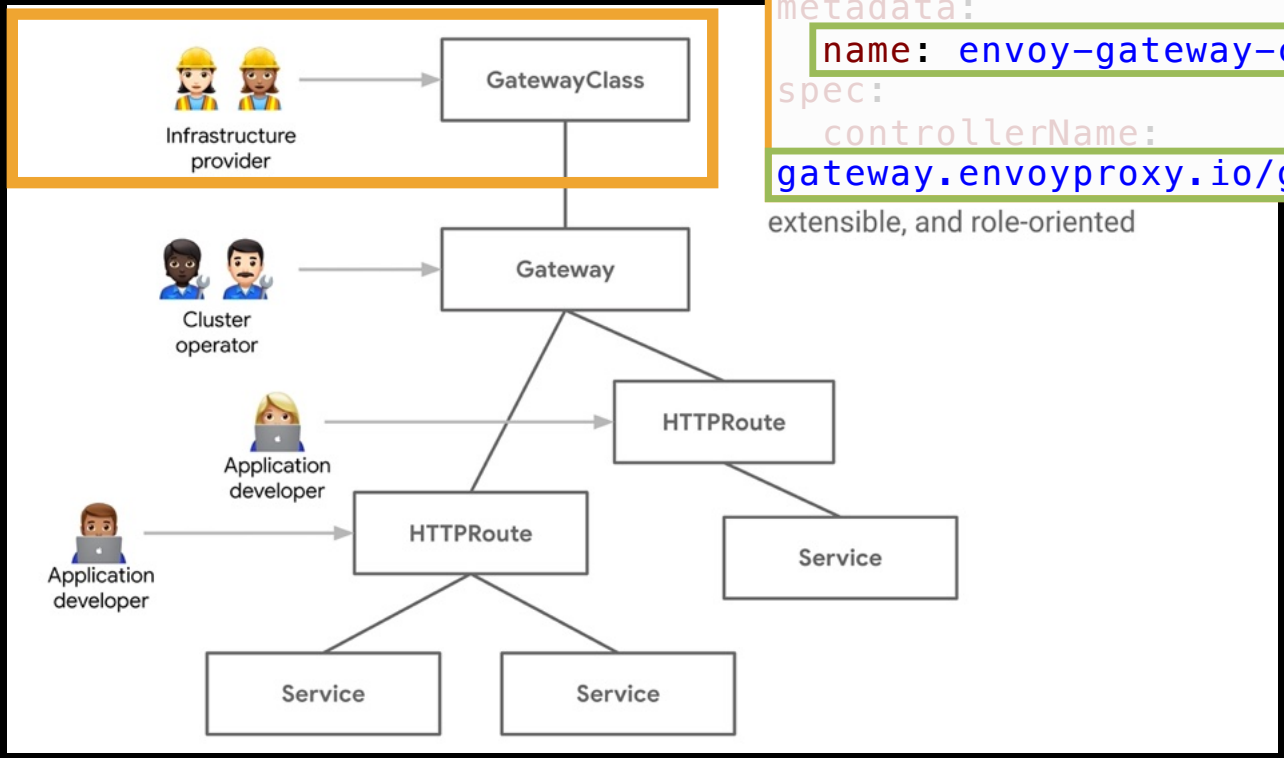
3. Separation of concerns

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$2
    cert-manager.io/issuer: selfsigned
  labels:
  name: cadec2024-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: cadec2024.localhost
    http:
      paths:
      - path: /app1(/|$)(.*)
        pathType: Prefix
        backend:
          service:
            name: httpbin
            port:
              name: http
  tls:
  - hosts:
    - cadec2024.localhost
    secretName: tls-certificate
```

WHERE ARE WE?

- Recap on Kubernetes Ingress
- Introducing the Gateway API
- Trying it out
- Ready for production?
- Summary

GATEWAY API – AN OVERVIEW

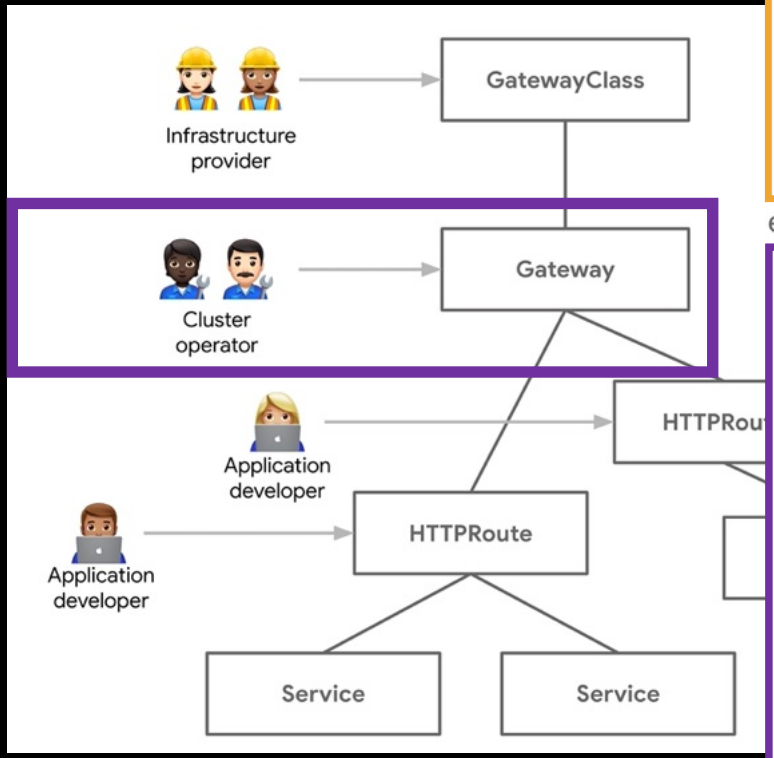



```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: envoy-gateway-class
spec:
  controllerName:
gateway.envoyproxy.io/gatewayclass-controller
```

extensible, and role-oriented

Source: https://gateway-api.sigs.k8s.io

GATEWAY API – AN OVERVIEW



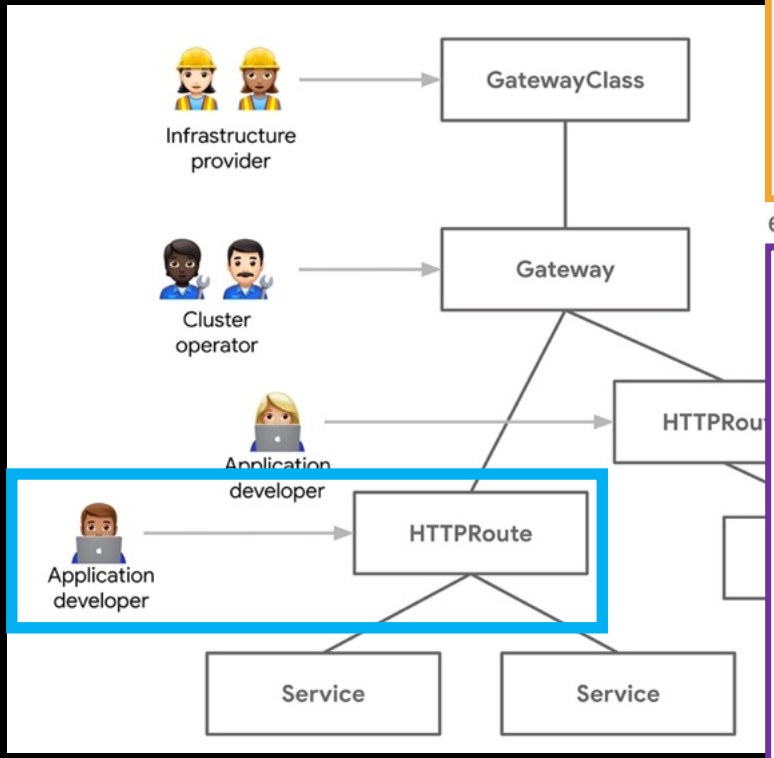
Source: https://gateway-api.

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: envoy-gateway-class
spec:
  controllerName:
gateway.envoyproxy.io/gatewayclass-controller
```

extensible, and role-oriented

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: envoy-http-ports
  annotations:
    cert-manager.io/cluster-issuer: selfsigned
spec:
  gatewayClassName: envoy-gateway-class
  listeners:
  - name: https
    protocol: HTTPS
    hostname: cadec2024.localhost
    port: 20443
    tls:
      mode: Terminate
      certificateRefs:
      - name: gw-cadec2024
```

GATEWAY API – AN OVERVIEW



Source: https://gateway-api.

```yaml
apiVersion: gateway.netwo
kind: GatewayClass
metadata:
  name: envoy-gateway-cla
spec:
  controllerName:
gateway.envoyproxy.io/gat
```

extensible, and role-oriented

```yaml
apiVersion: gateway.netwo
kind: Gateway
metadata:
  name: envoy-http-ports
  annotations:
    cert-manager.io/clust
spec:
  gatewayClassName: envoy
  listeners:
  - name: https
    hostname: cadec2024.l
    protocol: HTTPS
    port: 20443
    tls:
      mode: Terminate
      certificateRefs:
      - name: gw-cadec2024
```

```yaml
apiVersion: gateway...
kind: HTTPRoute
metadata:
  name: gw-http-routes
spec:
  parentRefs:
  - namespace: gateways
    name: envoy-http-ports
  hostnames:
  - cadec2024.localhost
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /app1
    filters:
    - type: URLRewrite
      urlRewrite:
        path:
          type: ReplacePrefixMatch
          replacePrefixMatch: /
    backendRefs:
    - kind: Service
      name: httpbin
      port: 8000
```

GATEWAY API – THERE IS MORE THAN JUST HTTP ROUTES

Object	OSI Layer	Routing Discriminator	TLS Support	Purpose
HTTPRoute	Layer 7	Anything in	Terminated	HTTP and HTTPS Routing
TLSRoute	Somewhere between layer 4 and 7			
TCPRoute	Layer 4			
UDPRoute	Layer 4			
GRPCRoute	Layer 7			

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: envoy-tcp-mongodb
  namespace: gateways
spec:
  gatewayClassName: envoy-gateway-class
  listeners:
  - name: tcp-mongodb
    port: 20017
    protocol: TCP
      allowedRoutes:
        kinds:
        - group: gateway.networking.k8s.io
          kind: TCPRoute
        namespaces:
          from: All
```

Source: https://gateway-api.sigs.k8s.io/concepts/api-overview/#route-summary-table

GATEWAY API – THERE IS MORE THAN JUST HTTP ROUTES

Object	OSI Layer	Routing Discriminator	TLS Support	Purpose
HTTPRoute	Layer 7	Anything in	Terminated	HTTP and HTTPS Routing
TLSRoute	Somewhere between layer 4 and 7			
TCPRoute	Layer 4			
UDPRoute	Layer 4			
GRPCRoute	Layer 7			

```yaml
apiVersion: gateway.networking.
kind: Gateway
metadata:
  name: envoy-tcp-mongodb
  namespace: gateways
spec:
  gatewayClassName: envoy-gatew
  listeners:
  - name: tcp-mongodb
    port: 20017
    protocol: TCP
      allowedRoutes:
        kinds:
        - group: gateway.networking.k8s.io
          kind: TCPRoute
        namespaces:
          from: All
```

```yaml
apiVersion: gateway.networking.k8s.io/v1alpha2
kind: TCPRoute
metadata:
  name: envoy-gw-tcp-mongodb-route
spec:
  parentRefs:
  - namespace: gateways
    name: envoy-tcp-mongodb
  rules:
  - backendRefs:
    - kind: Service
      name: mongodb
      port: 27017
```

Source: https://gateway-api.sigs.k8s.io/concepts/api-overview/#route-summary-table

GATEWAY API –WHAT'S IN V1.0?

- Resources in Gateway API v1.0 - `gateway.networking.k8s.io`

- v1
  - GatewayClass
  - Gateway
  - HTTPRoute

- v1beta1
  - ReferenceGrant

- v1alpha2
  - GRPCRoute
  - TCPRoute
  - TLSRoute
  - UDPRoute
  - BackendTLSPolicy

GATEWAY API – COVERED IN THIS PRESENTATION

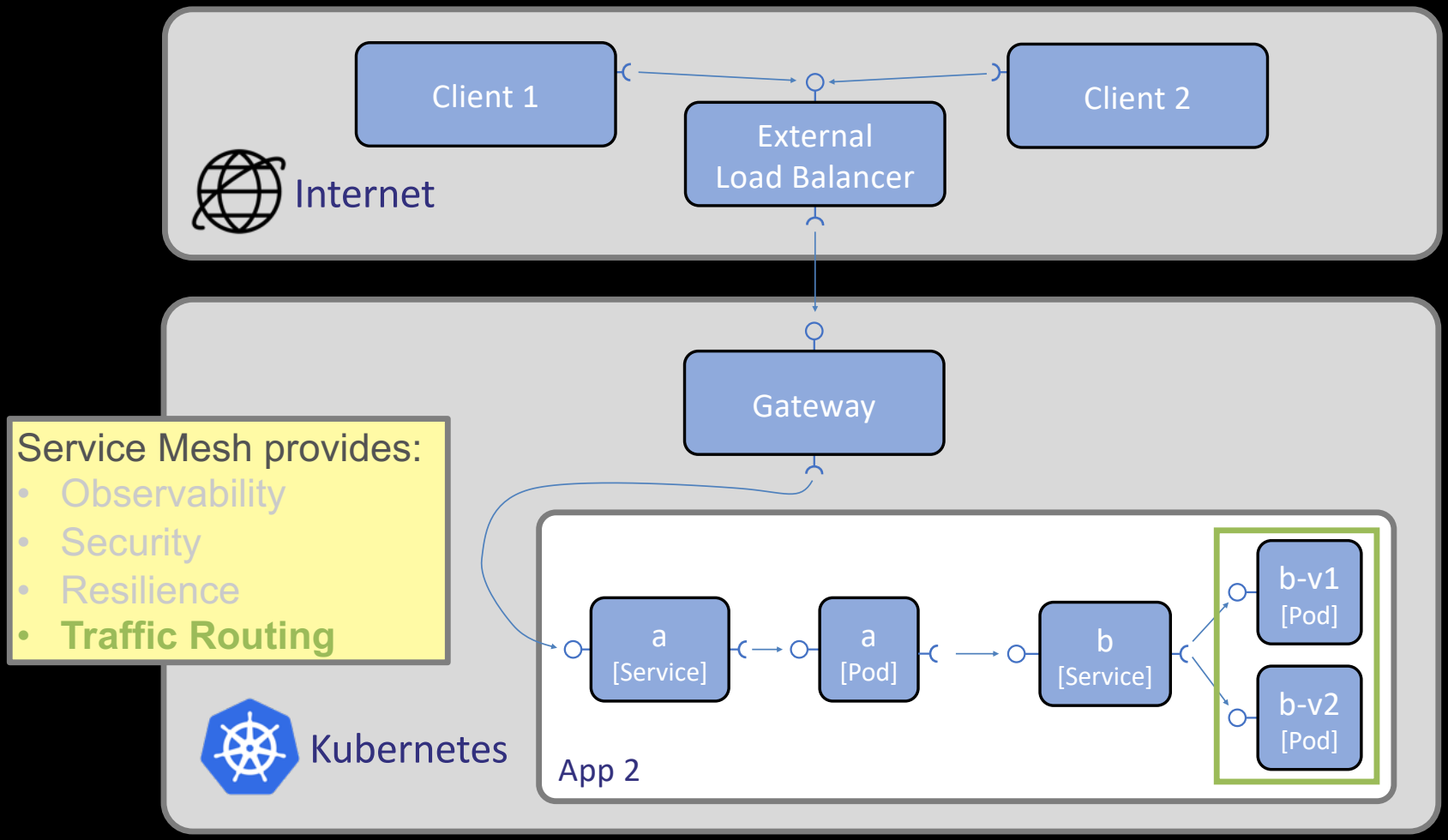
- Resources in Gateway API v1.0 - `gateway.networking.k8s.io`

- v1
  - GatewayClass
  - Gateway
  - HTTPRoute

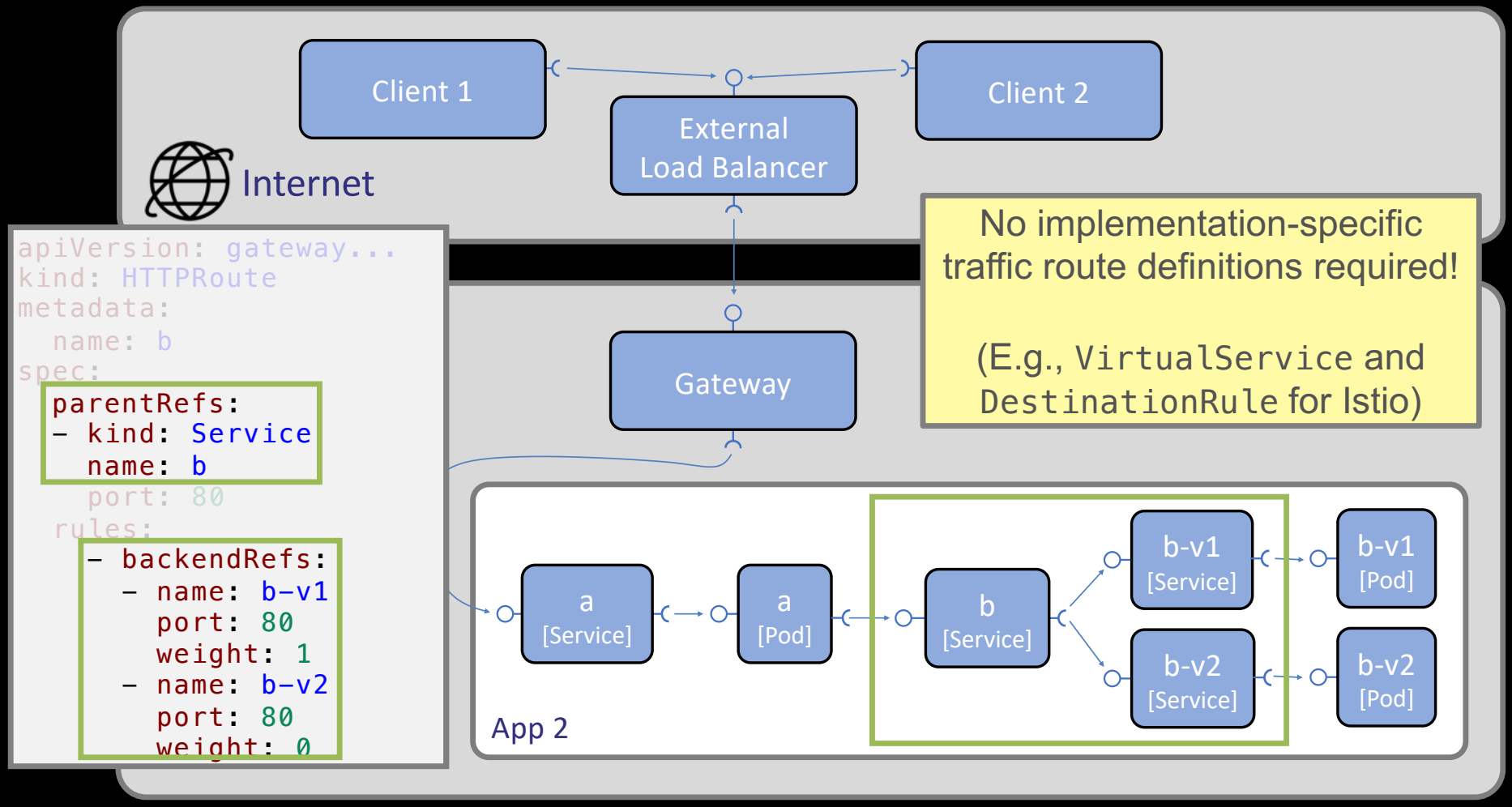
- v1beta1
  - ReferenceGrant

- v1alpha2
  - GRPCRoute
  - TCPRoute
  - TLSRoute
  - UDPRoute
  - BackendTLSPolicy

GAMMA - GATEWAY API FOR MESH MANAGEMENT AND ADMINISTRATION



GAMMA - GATEWAY API FOR MESH MANAGEMENT AND ADMINISTRATION



KUBERNETES GATEWAY API IMPLEMENTATIONS

- See https://gateway-api.sigs.k8s.io/implementations/
- 25 implementations, only a few in GA state
- 3 service mesh implementations, all in an experimental state

Gateway API support in Istio

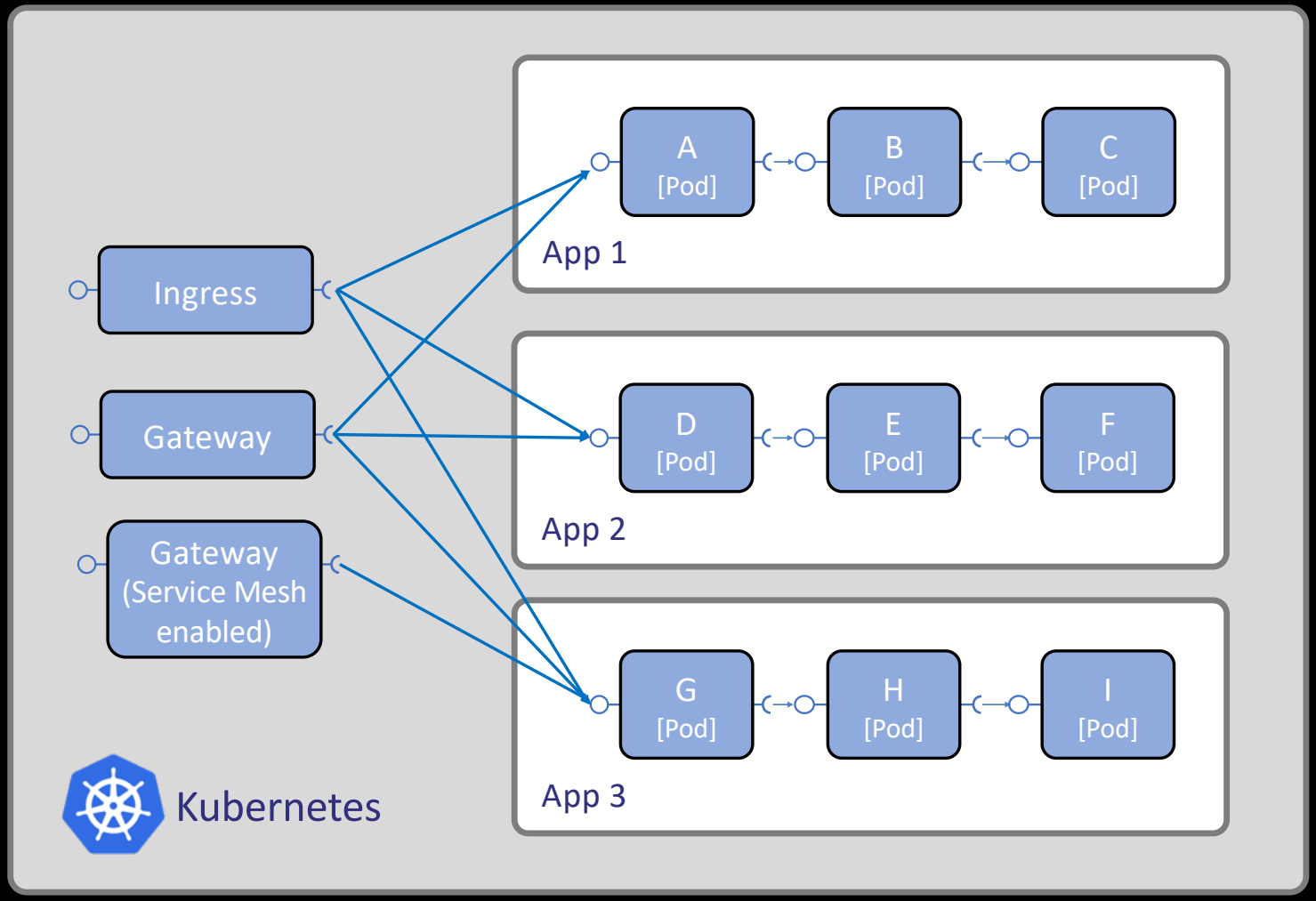
In addition to its own traffic management API,
Istio includes beta support for the Kubernetes Gateway API
and ***intends to make it the default API*** for traffic management
in the future.

https://istio.io/latest/docs/tasks/traffic-management/ingress/gateway-api/

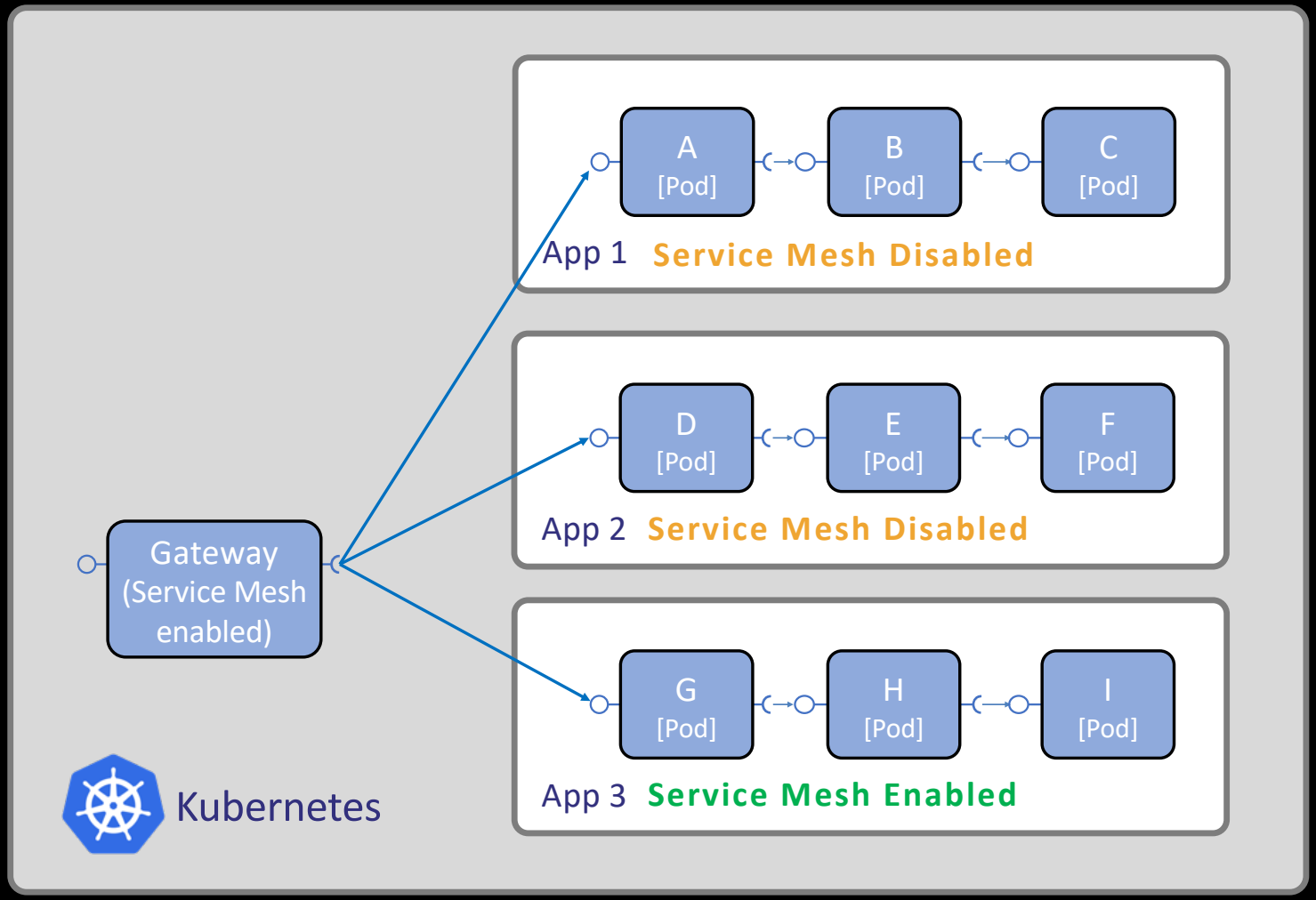
WHERE ARE WE?

- Recap on Kubernetes Ingress
  - …and its limitations
- Introducing the Gateway API
- Trying it out
- Ready for production?
- Summary

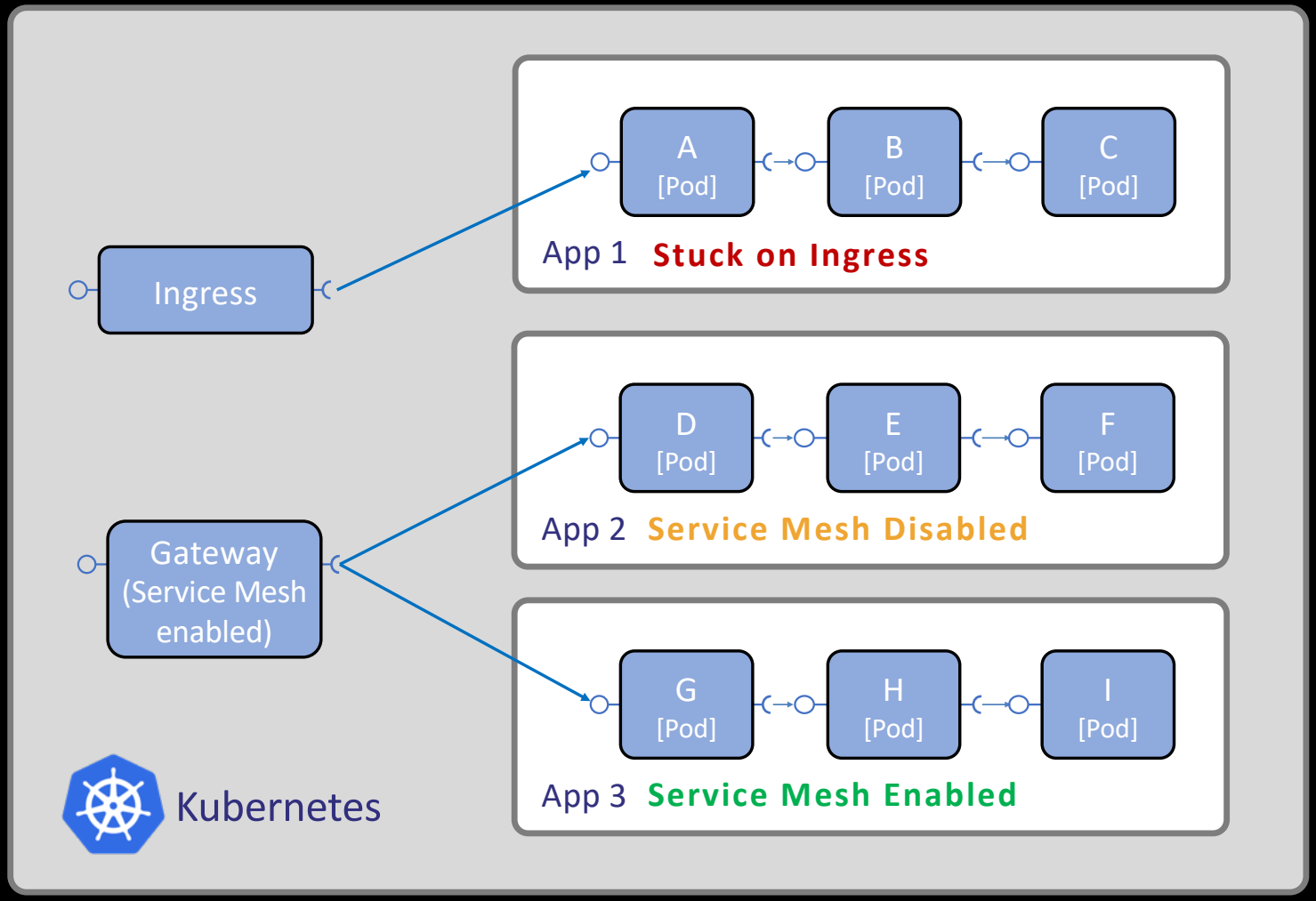
TEST TARGET– REPLACING THE INGRESS IN A SHARED ENVIRONMENT



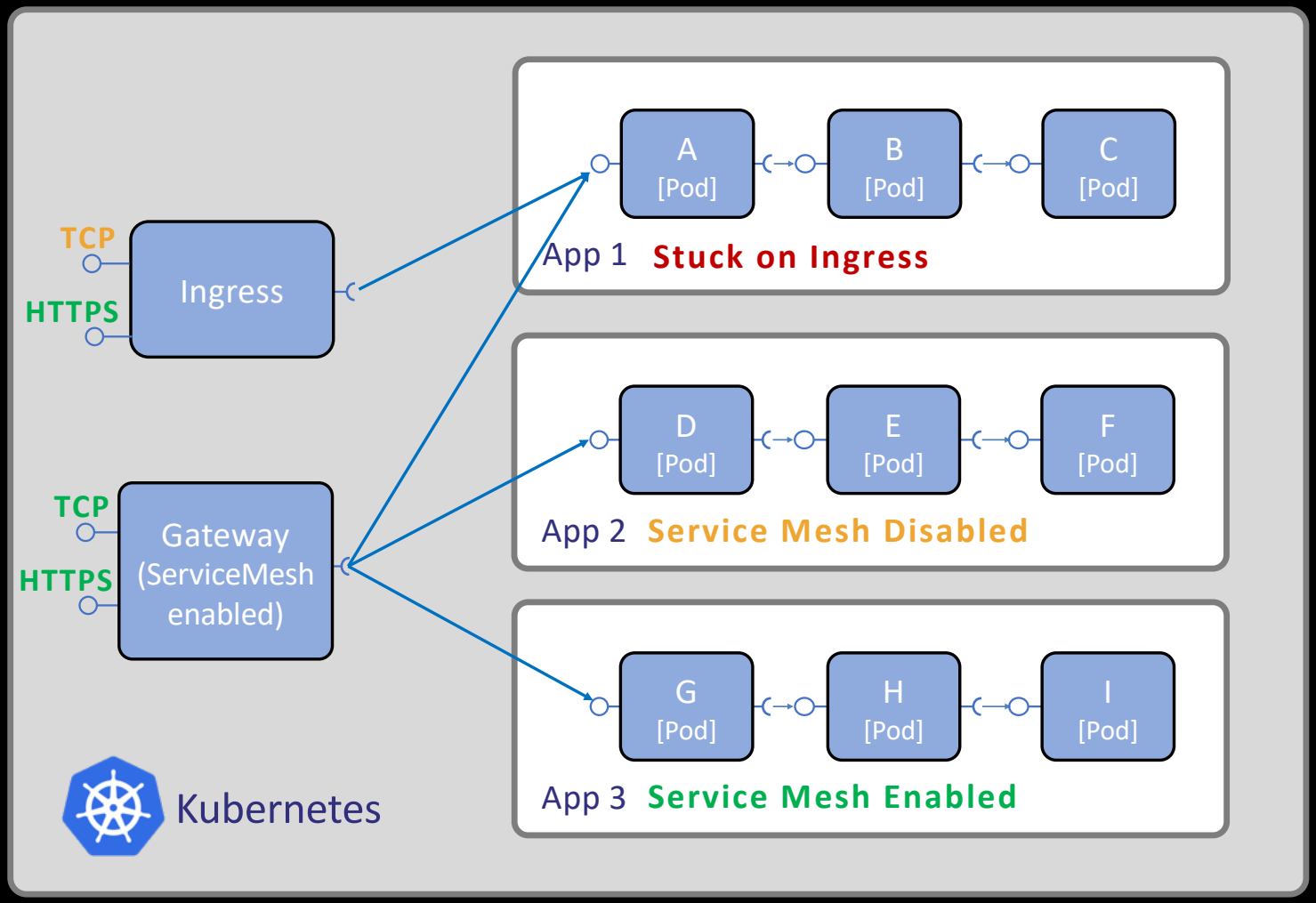
TEST TARGET– REPLACING THE INGRESS IN A SHARED ENVIRONMENT



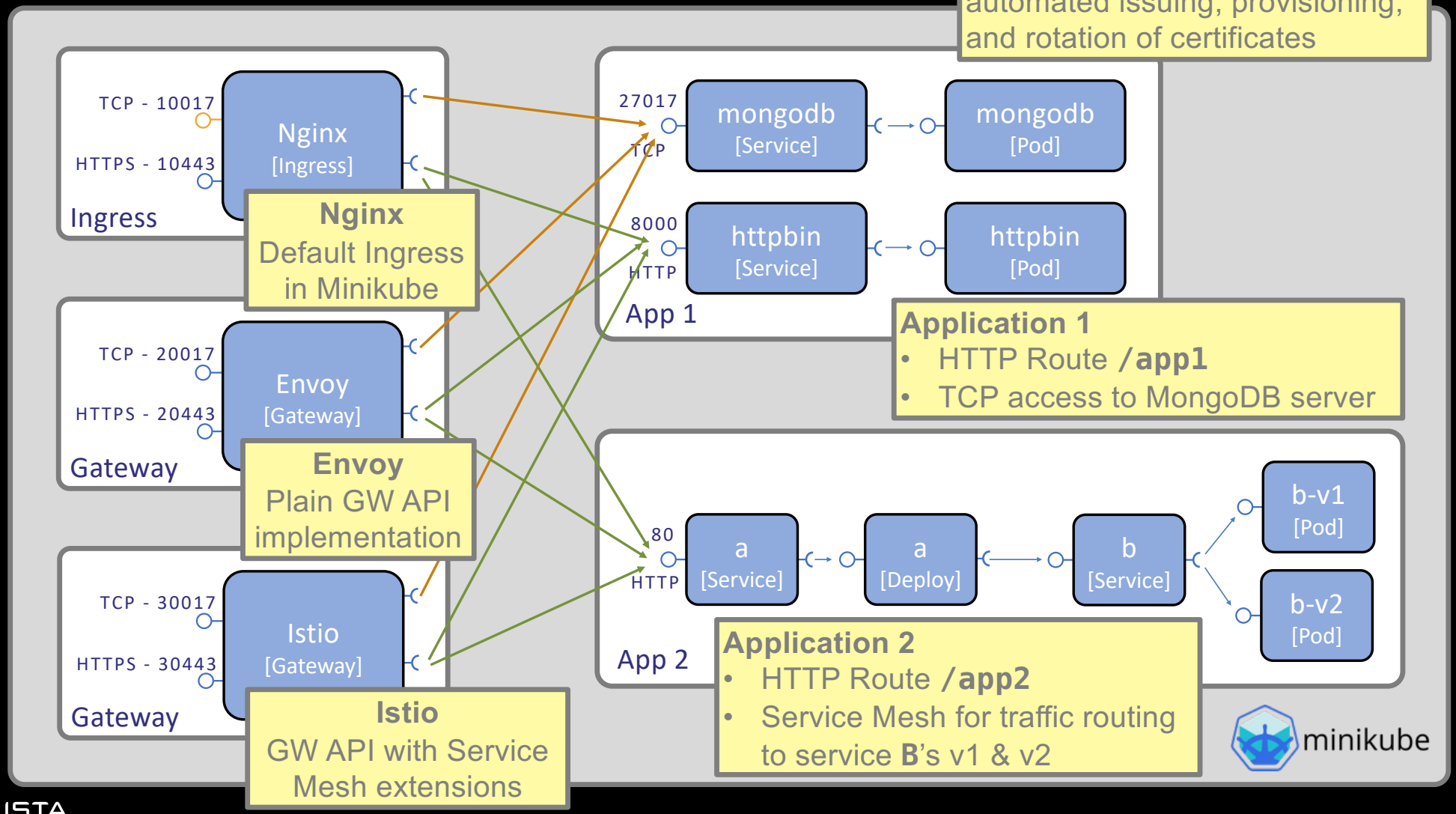
TEST TARGET– REPLACING THE INGRESS IN A SHARED ENVIRONMENT



TEST TARGET– REPLACING THE INGRESS IN A SHARED ENVIRONMENT



TEST ENVIRONMENT



TEST ENVIRONMENT – MOVING PARTS IN THE KUBERNETES CLUSTER

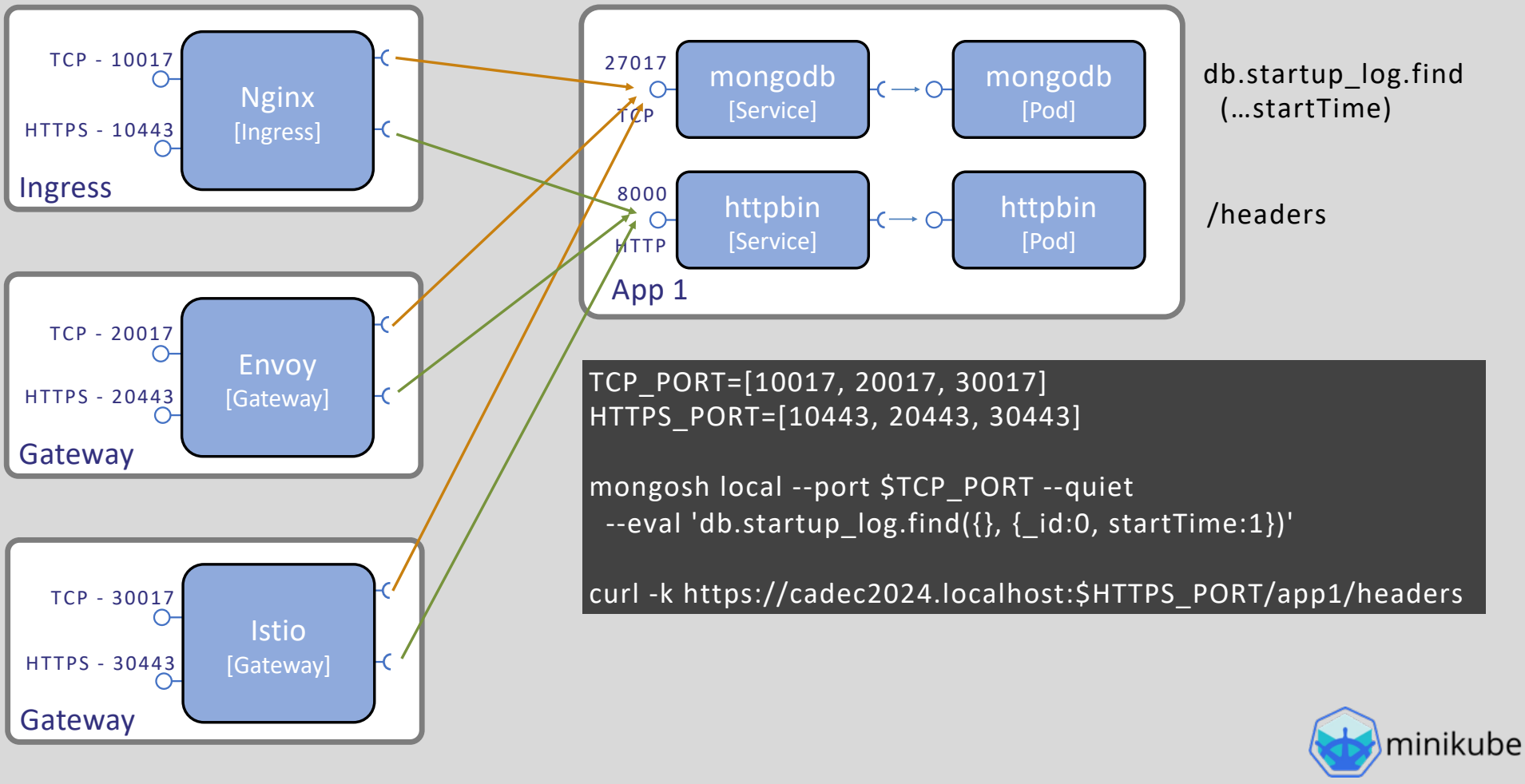
Terminal

```
$ kubectl get pods -A
NAMESPACE              NAME                                                      READY   STATUS      RESTARTS        AGE
app1                   httpbin-86b8ffc5ff-q67w4                                  1/1     Running     2 (16h ago)     6d19h
app1                   mongodb-875f5d8b7-jnngz                                   1/1     Running     2 (16h ago)     6d19h
app2                   a-58c4d55d8b-vrvqx                                        2/2     Running     8 (5h9m ago)    6d19h
app2                   b-v1-6c886dbd6-xm44j                                      2/2     Running     4 (16h ago)     6d19h
app2                   b-v2-7d4f6c997b-h45l7                                     2/2     Running     4 (16h ago)     6d19h
cert-manager           cert-manager-cainjector-7b5b5d4786-vnz2m                  1/1     Running     3 (16h ago)     6d19h
cert-manager           cert-manager-f6559db96-qz8ws                              1/1     Running     3 (16h ago)     6d19h
cert-manager           cert-manager-webhook-55fb5c9c88-llvbf                     1/1     Running     3 (16h ago)     6d19h
envoy-gateway-system   envoy-gateway-96964c865-m5bjq                             1/1     Running     2 (16h ago)     6d19h
envoy-gateway-system   envoy-gateways-envoy-http-ports-350bb3ee-5f5f45c77b-qh72w  1/1     Running     2 (16h ago)     6d19h
envoy-gateway-system   envoy-gateways-envoy-tcp-mongodb-022ab9d3-5d6dcd9bd7-hnnns 1/1     Running     2 (16h ago)     6d19h
gateways               istio-http-ports-istio-b75855964-f48fs                    1/1     Running     4 (5h8m ago)    6d19h
gateways               istio-tcp-mongodb-istio-7bf87498bc-ph5vn                  1/1     Running     4 (5h8m ago)    6d19h
ingress-nginx          ingress-nginx-admission-create-bcw24                      0/1     Completed   0               6d19h
ingress-nginx          ingress-nginx-admission-patch-pslx7                       0/1     Completed   1               6d19h
ingress-nginx          ingress-nginx-controller-75c685b784-n8h6r                 1/1     Running     2 (16h ago)     6d19h
istio-system           grafana-b8bbdc84d-m8nd2                                   1/1     Running     2 (16h ago)     6d19h
istio-system           istiod-79b769cf65-dmkhp                                   1/1     Running     2 (16h ago)     6d19h
istio-system           jaeger-7d7d59b9d-p4bd7                                    1/1     Running     2 (16h ago)     6d19h
istio-system           kiali-545878ddbb-hrt5r                                    1/1     Running     2 (16h ago)     6d19h
istio-system           prometheus-db8b4588f-vbxr8                                2/2     Running     4 (16h ago)     6d19h
kube-system            coredns-5dd5756b68-xlxp9                                  1/1     Running     2 (16h ago)     6d19h
kube-system            etcd-cadec2024-gw                                         1/1     Running     2 (16h ago)     6d19h
kube-system            kube-apiserver-cadec2024-gw                               1/1     Running     2 (16h ago)     6d19h
kube-system            kube-controller-manager-cadec2024-gw                      1/1     Running     2 (16h ago)     6d19h
kube-system            kube-proxy-crf4h                                          1/1     Running     2 (16h ago)     6d19h
kube-system            kube-scheduler-cadec2024-gw                               1/1     Running     2 (16h ago)     6d19h
kube-system            storage-provisioner                                       1/1     Running     6 (5h9m ago)    6d19h
$
```

WHERE ARE WE?

- Recap on Kubernetes Ingress
- Introducing the Gateway API
- Trying it out
  - Replacing Kubernetes Ingress
- Ready for production?
- Summary

DEMO #1, REPLACING KUBERNETES INGRESS



DEMO #1, REPLACING KUBERNETES INGRESS

- Test commands

  - TCP – MongoDb shell

```
mongosh local --port 10017 --quiet --eval 'db.startup_log.find({}, {_id:0, startTime:1})'
mongosh local --port 20017 --quiet --eval 'db.startup_log.find({}, {_id:0, startTime:1})'
```

  - HTTPS – httpbin return incoming headers

```
curl -k https://cadec2024.localhost:10443/app1/headers
curl -k https://cadec2024.localhost:30443/app1/headers
```

DEMO #1 - APP1, EXPECTED RESPONSES

- MongoDB

```
[ { startTime: ISODate('2023-12-06T07:04:56.000Z') } ]
```

- HTTPS

```
{
  "headers": {
    "Accept": "*/*",
    "Host": "cadec2024.localhost:10443",
    "User-Agent": "curl/8.1.2",
    "X-Forwarded-Host": "cadec2024.localhost:10443",
    "X-Forwarded-Scheme": "https",
    "X-Scheme": "https"
  }
}
```

- Note extra headers from the Istio GW!

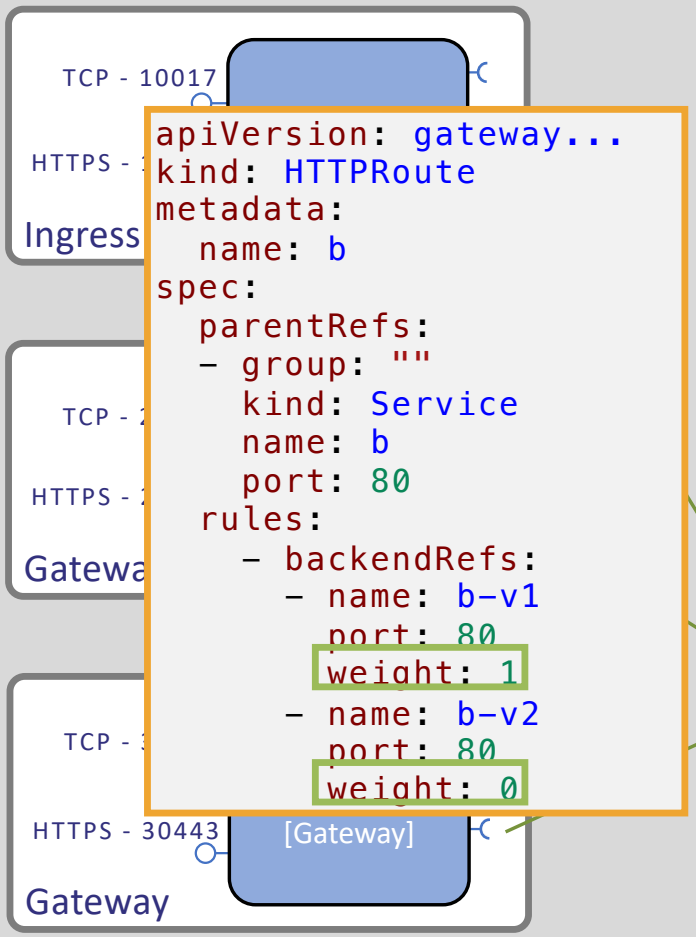
```
"X-B3-Sampled": "1",
"X-B3-Spanid": "fca8bbd692731d69",
"X-B3-Traceid": "e3ecc1201966e0d3fca8bbd692731d69",
```

Someone added Distributed Tracing headers!

WHERE ARE WE?

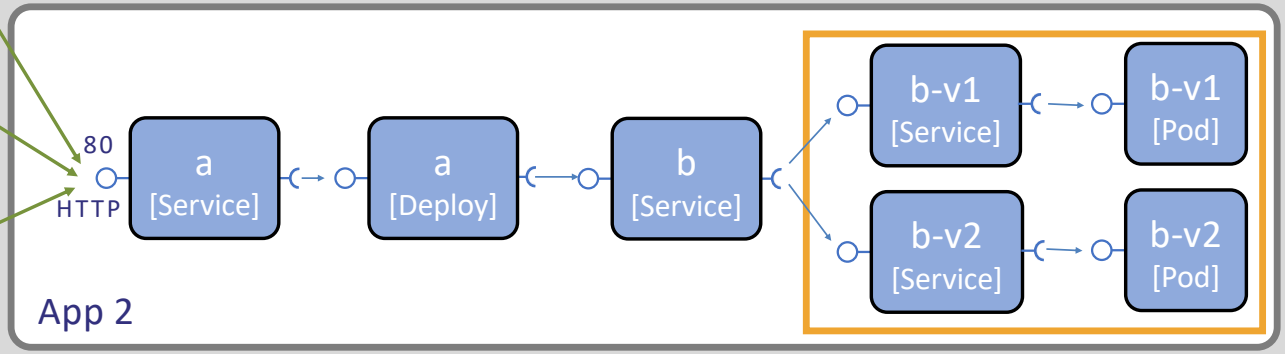
- Recap on Kubernetes Ingress
- Introducing the Gateway API
- Trying it out
  - Replacing Kubernetes Ingress
  - Using Service Mesh extensions
- Ready for production?
- Summary

DEMO #2 – SERVICE MESH TESTS



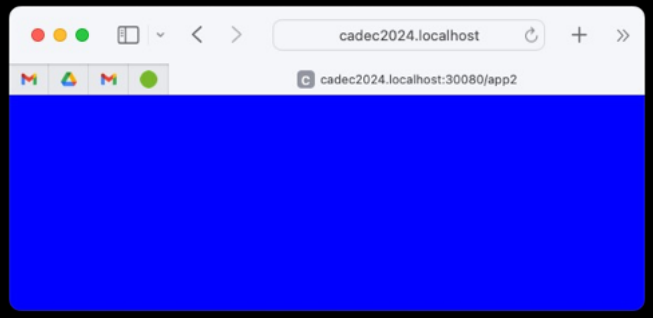

```yaml
apiVersion: gateway...
kind: HTTPRoute
metadata:
  name: b
spec:
  parentRefs:
  - group: ""
    kind: Service
    name: b
    port: 80
  rules:
    - backendRefs:
      - name: b-v1
        port: 80
        weight: 1
      - name: b-v2
        port: 80
        weight: 0
```

```
# Configure traffic to service B v1
./scripts/app2-service-b-split-traffic.bash 1 0
open http://cadec2024.localhost:30080/app2

# Configure traffic to service B v2
./scripts/app2-service-b-split-traffic.bash 0 1
open http://cadec2024.localhost:30080/app2

# Configure traffic to both service B v1 and v2, e.g. 40% and 60%
./scripts/app2-service-b-split-traffic.bash 2 3
open http://cadec2024.localhost:30080/app2
```

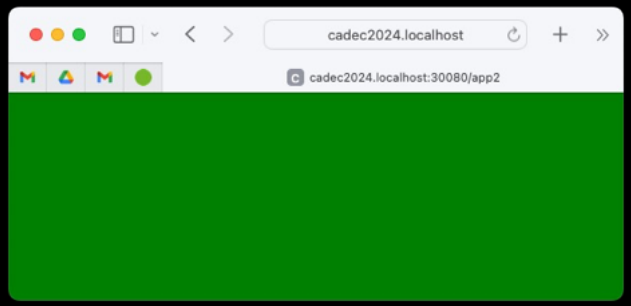


DEMO #2 – APP2, EXPECTED RESPONSES

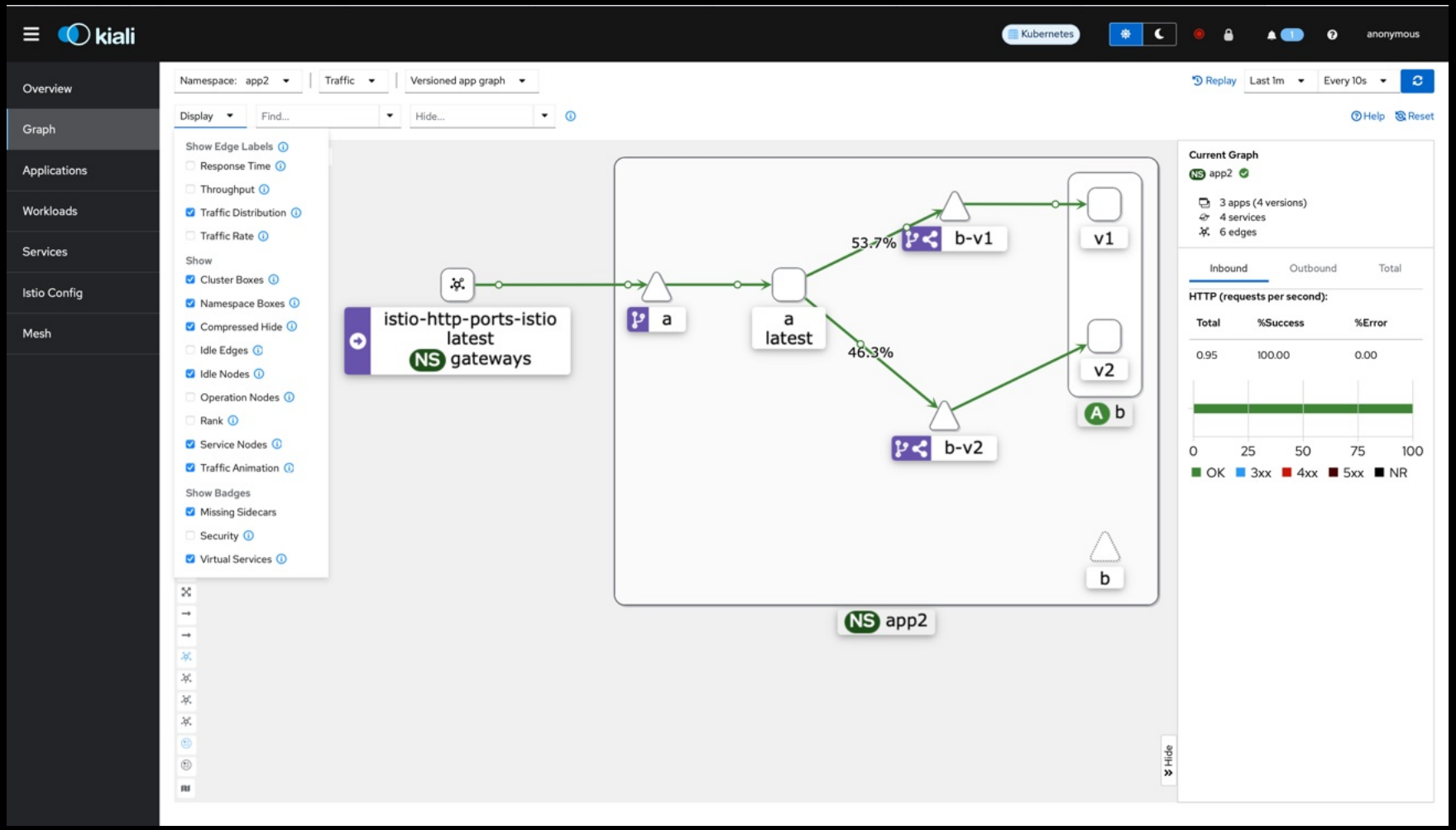
- Service B, v1



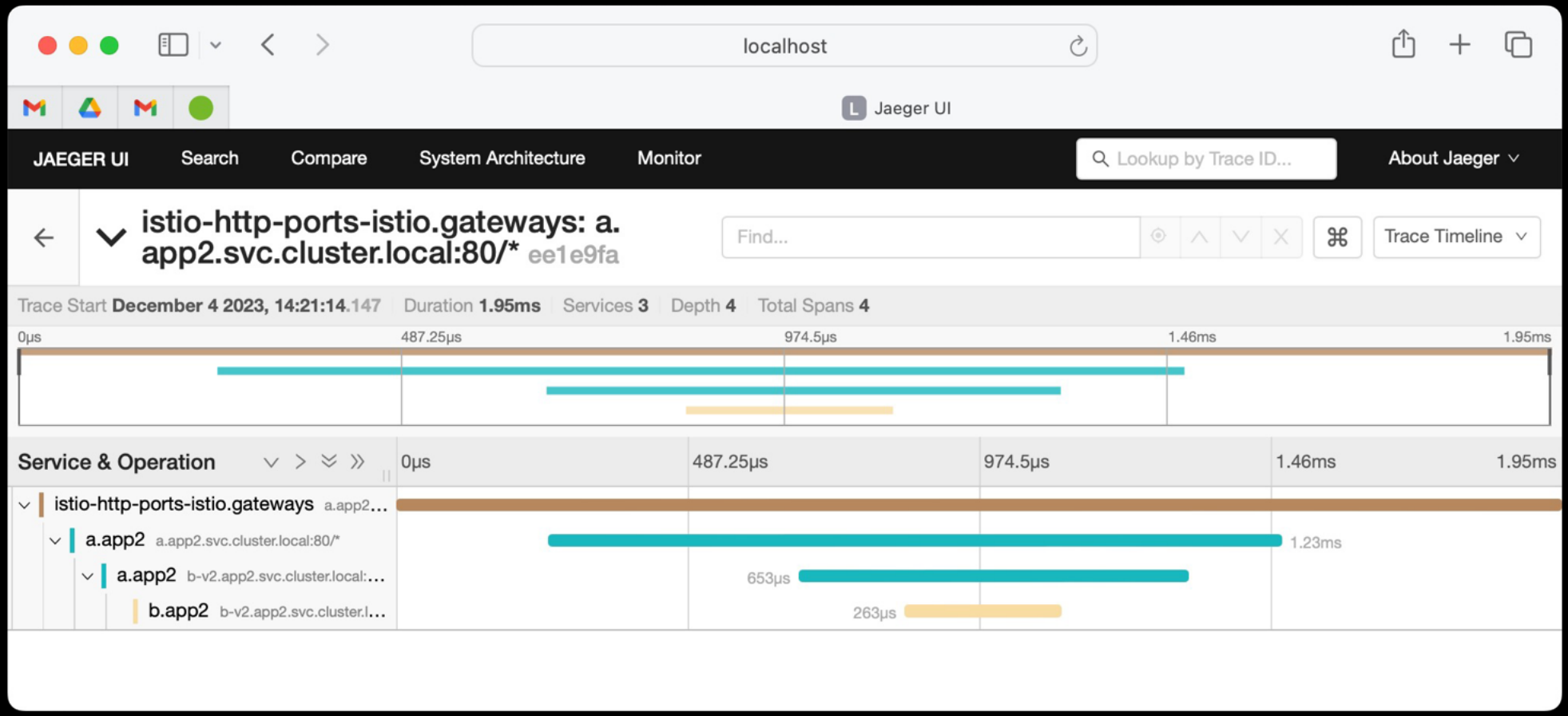
- Service B, v2



APP 2 - SERVICE MESH OBSERVABILITY USING KIALI



APP 2 - SERVICE MESH DISTRIBUTED TRACING USING JAEGER



GATEWAY API PORTABILITY

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: envoy-http-ports
  annotations:
    cert-manager.io/cluster-issuer: selfsigned
spec:
  gatewayClassName: envoy
  listeners:
  - name: https
    hostname: cadec2024.localhost
    protocol: HTTPS
    port: 20443
    tls:
      mode: Terminate
      certificateRefs:
      - name: gw-cadec2024
```

GATEWAY API PORTABILITY

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: envoy-http-ports
  annotations:
    cert-manager.io/cluster-issuer: selfsigned
spec:
  gatewayClassName: envoy ➔ istio
  listeners:
  - name: https
    hostname: cadec2024.localhost
    protocol: HTTPS
    port: 20443
    tls:
      mode: Terminate
      certificateRefs:
      - name: gw-cadec2024
```

Change Gateway API implementation:

- Change reference to the Gateway Class

Use cases:

- Introduce a Service Mesh
- Developers can use their favorite; whatever used in production

WHERE ARE WE?

- Recap on Kubernetes Ingress
- Introducing the Gateway API
- Trying it out
- Ready for production?
- Summary

GATEWAY API AND MANAGED KUBERNETES

- It depends on what packaged or managed version of Kubernetes is used…

AWS EKS	https://www.gateway-api-controller.eks.aws.dev
Azure AKS	https://azure.microsoft.com/en-gb/updates/public-preview-application-gateway-for-containers/
Google GKE	https://cloud.google.com/kubernetes-engine/docs/concepts/gateway-api
VMware Tanzu	https://docs.vmware.com/en/VMware-NSX-Advanced-Load-Balancer/1.11/Avi-Kubernetes-Operator-Guide/GUID-84BD68AB-B96F-425C-8323-3A249D6AC8B2.html
RedHat OpenShift	https://cloud.redhat.com/blog/gateway-api-with-openshift-networking-developer-preview-update

WHERE ARE WE?

- Recap on Kubernetes Ingress
- Introducing the Gateway API
- Trying it out
- Ready for production?
- Summary

SUMMARY

- Ingress served us well…
  - But has its limitations
- Addressed by the Gateway API
  - Role-based design
  - Supports
    - » Routes HTTP(S), TCP, UDP, and gRPC based traffic
    - » Service Mesh traffic routing
  - Feature-rich
    - » Minimize the need for vendor-specific extension
- Easy to migrate from Ingress
  - Coexist to support stepwise migration
- Easy to introduce a Service Mesh
  - Just switch Gateway API implementation
  - Relies on Gateway API portability

SUMMARY

- When and how to migrate?

1. Start learning now

2. Wait with production until your Kubernetes platform supports it!

3. Start by replacing Ingress with HTTP Routes

4. Move on based on business needs with
   - » Routes for other protocols
   - » Introducing a Service Mesh

QUESTIONS?