



Introduction to service mesh with Istio and Kiali

Alissa Bonas



Evolution of application architecture

...

How did we get to service mesh?

Monolith application

**Single unit of
executable**

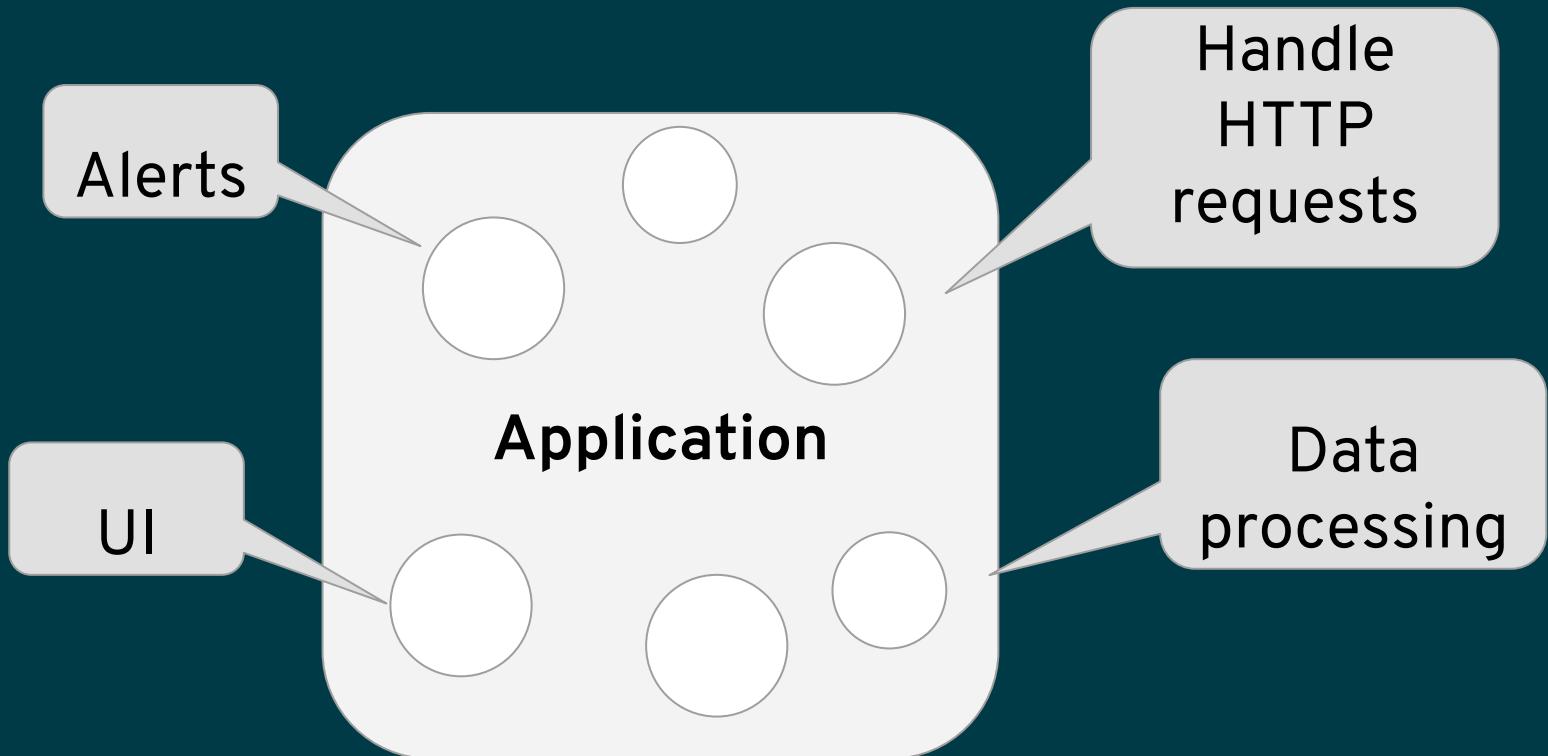
=

Application

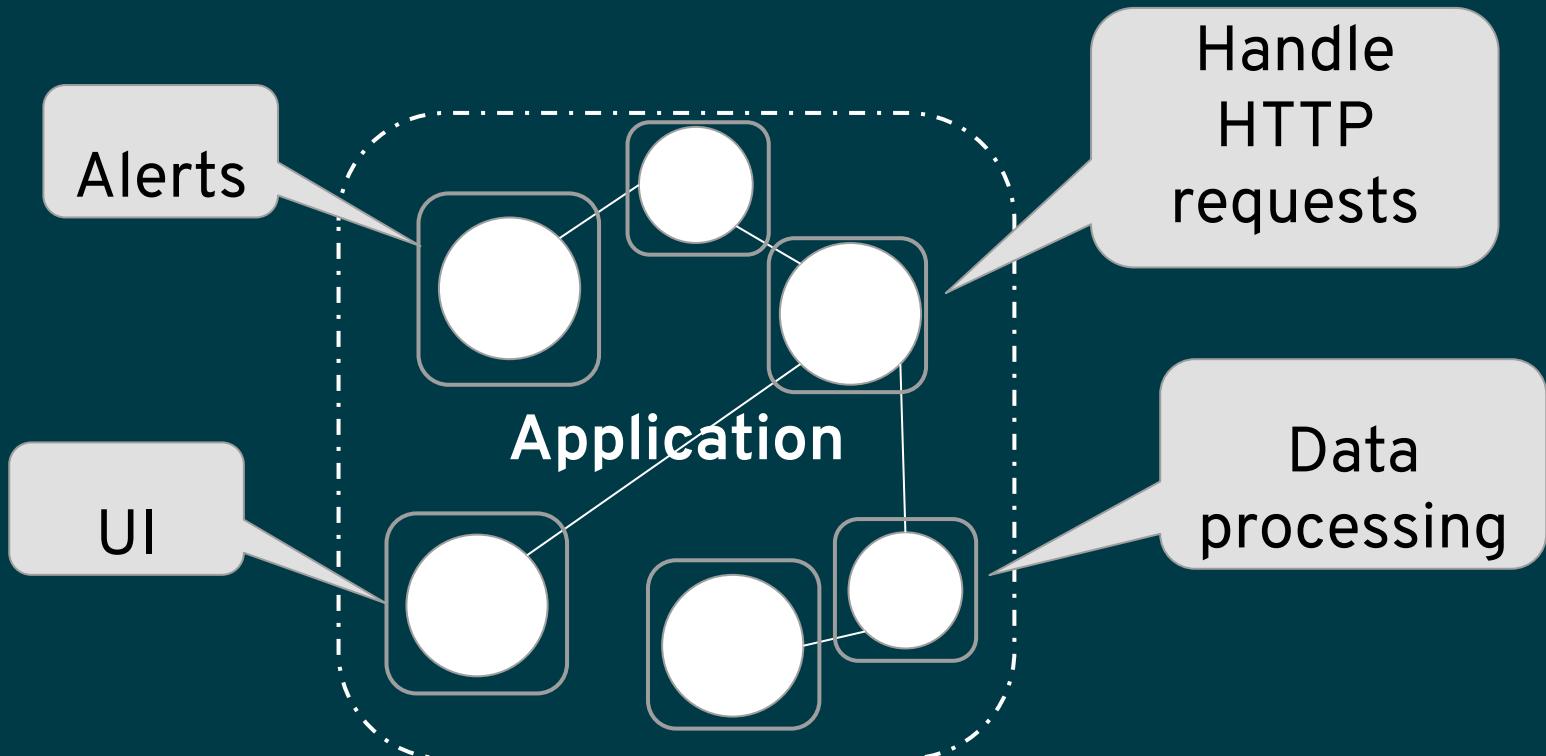
=

Single process

Application modules



Multiple processes

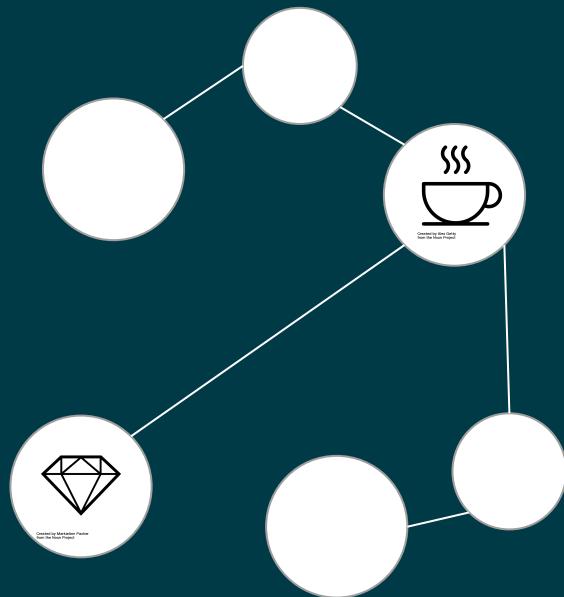


Microservices

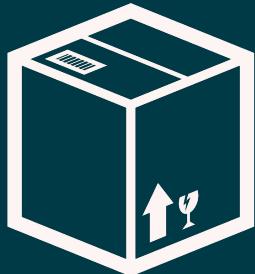
Language agnostic

Scaled separately

Upgraded separately



A shift in Application Packaging and Runtime





Containerizing an app



Run multiple containers



Orchestrate containers

- Run many containers on multiple hosts
- Scale - manage several instances (replicas) of the same container
- Manage a container based environment



Container orchestration platforms



Kubernetes

okd

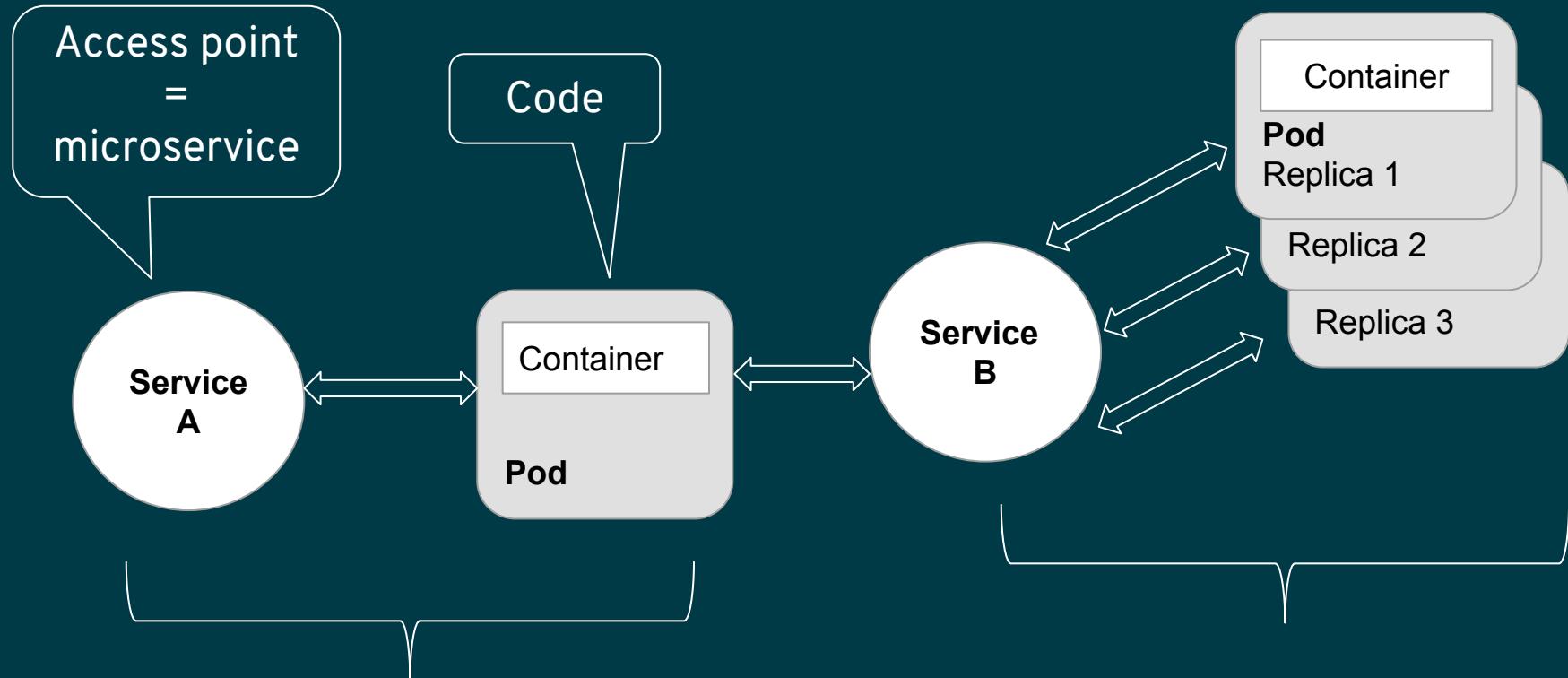
OKD
(OpenShift)



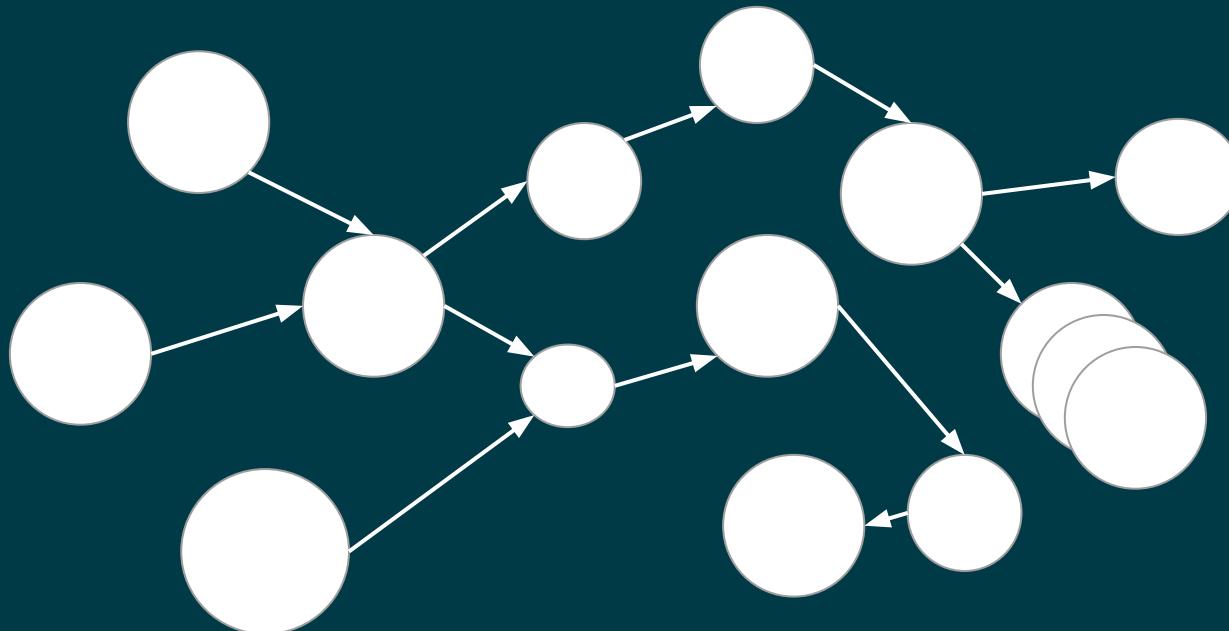
Kubernetes building blocks (some...)

- Pod - a group of one or more containers, with shared storage/network
- Deployment - manages pod definition and defines replicas of pods
- Service - an abstraction, an access point to a set of Pods
 - Sometimes called a **microservice**

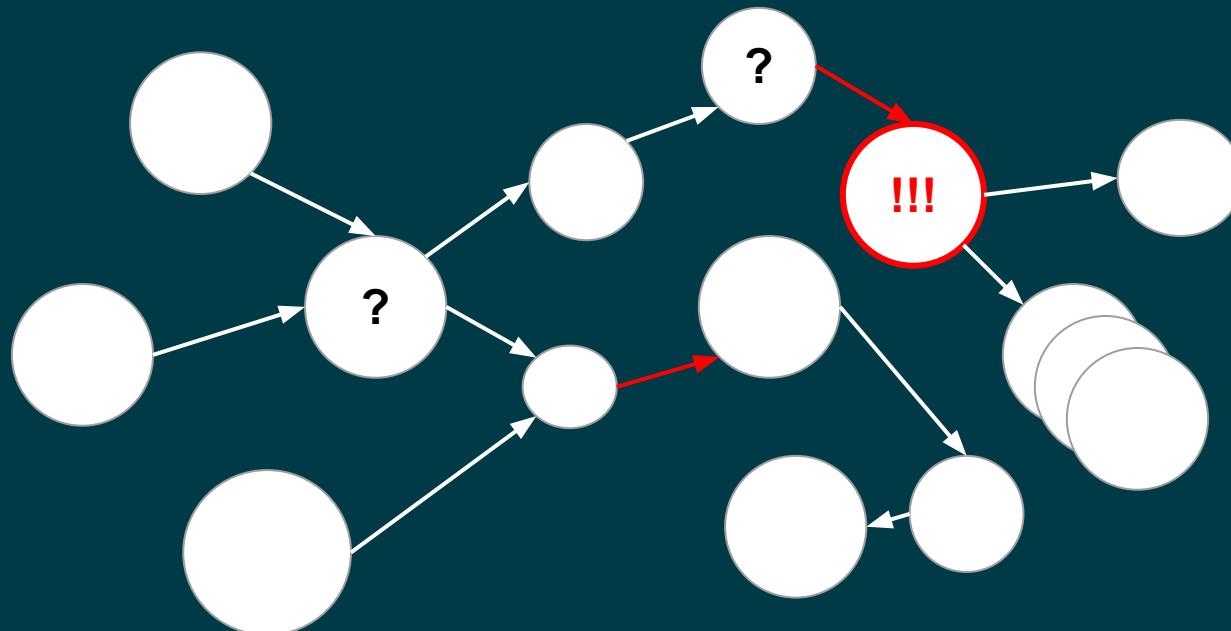
Microservices - the Kubernetes way



High Complexity



Multiple points of failure



Challenges

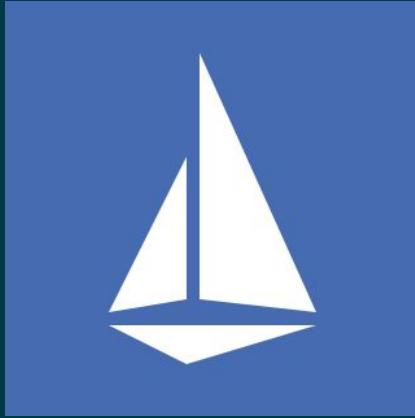
- How are the requests routed between services?
- How do I detect failures and downtime?
- How to upgrade and test new versions of a service?
- Securing the communication



Service mesh to the rescue

What is a service mesh

- Infrastructure/framework that handles communication between services
- Often implemented as network proxies deployed alongside the microservices



Istio - Ιστίο

...

Open source service mesh



The dry facts

- Started in May 2017
- Means “sail” in Greek
- Developed in Go



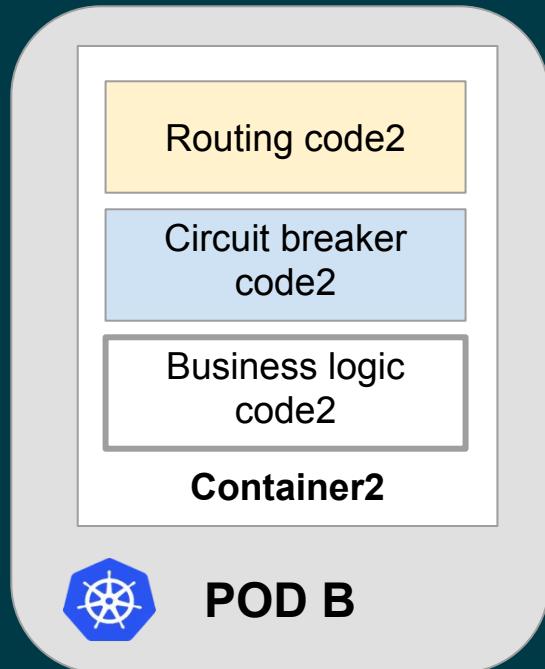
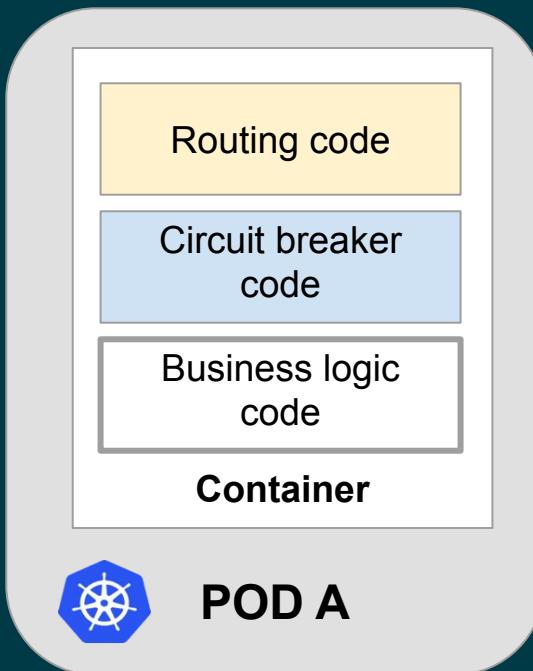
Istio features

- Load balancing (HTTP, gRPC, TCP...)
- Traffic control (routing rules, retries, timeouts, fault injection, mirroring)
- Secure service-to-service communication
- Access controls (authorization)
- Metrics and traces for traffic

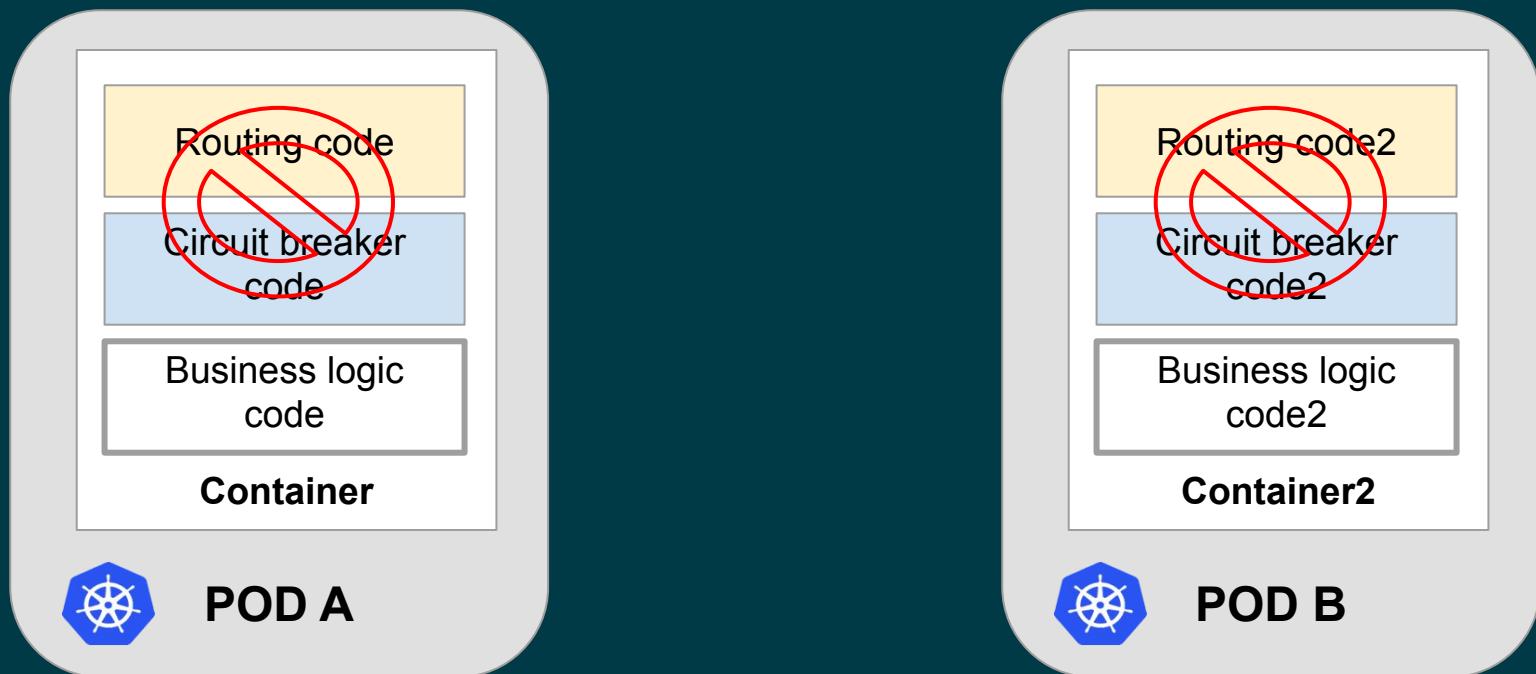
Important Terminology

- Workload - anything owning/controlling pods (like a Deployment) or the pods themselves
- Service - a **microservice**
- Application - *label* “app” on a pod/service
- Version - *label* “version” on a pod/service

Before Istio



Istio



Sidecar Proxy

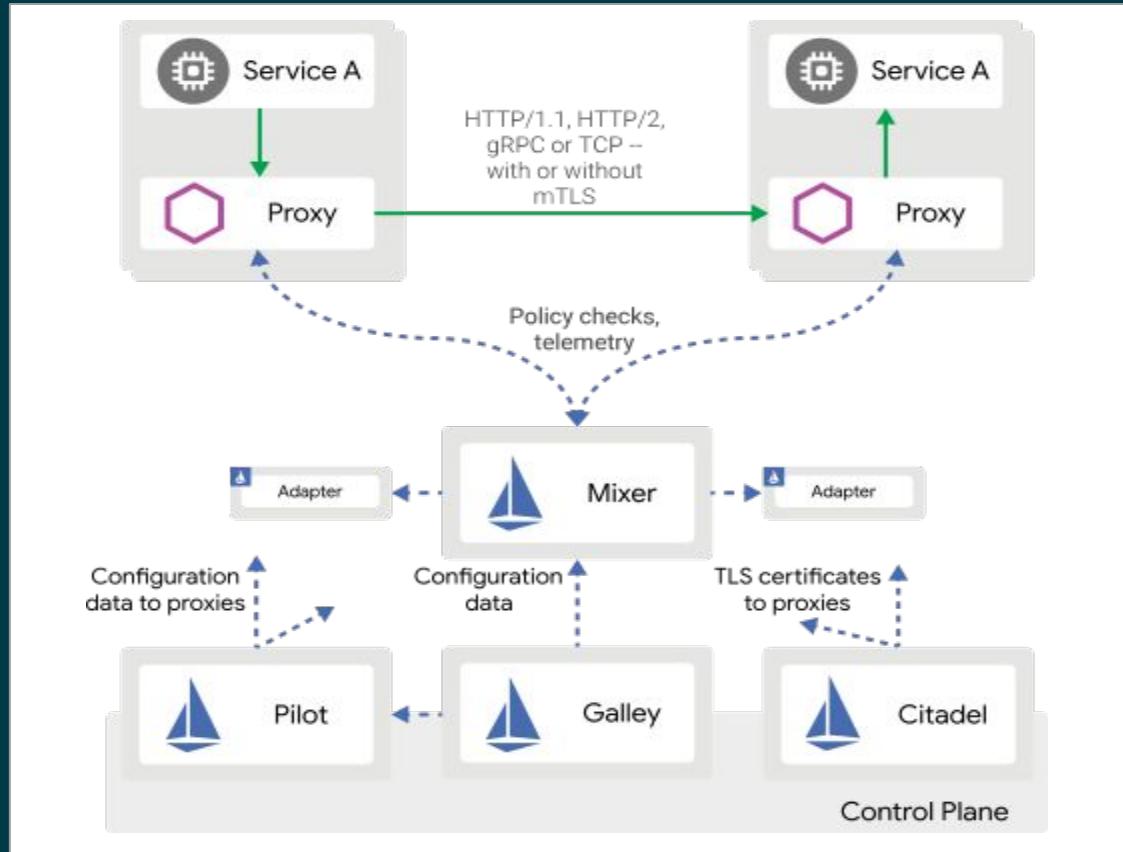
- A proxy is deployed in a container next to each instance of microservice (inside a pod)
- Container name: istio-proxy
- It is **transparent** to application code
- Envoy open source proxy is currently used



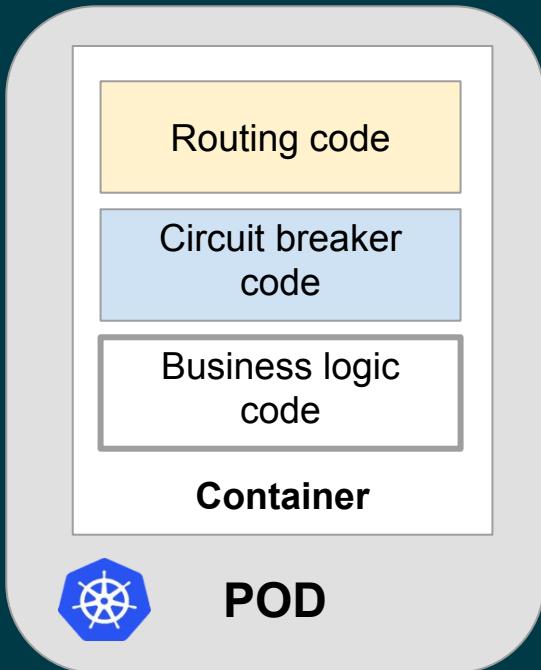
How is the sidecar injected?

- Manually
- Automatically injected to pod on creation
 - *kubectl label namespace default istio-injection=enabled*
 - Mutating Admission Webhook is used for sidecar injection
 - Actually... 2 containers are injected: `istio-init` and `istio-proxy`

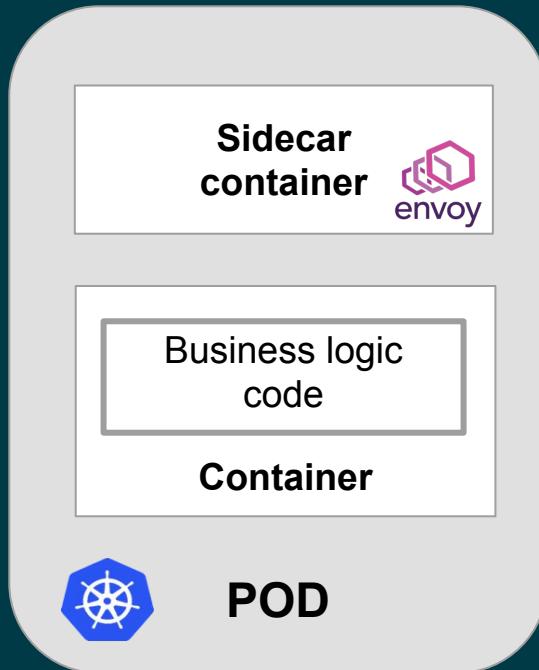
Istio architecture



Sidecar Proxy in Istio and Kubernetes

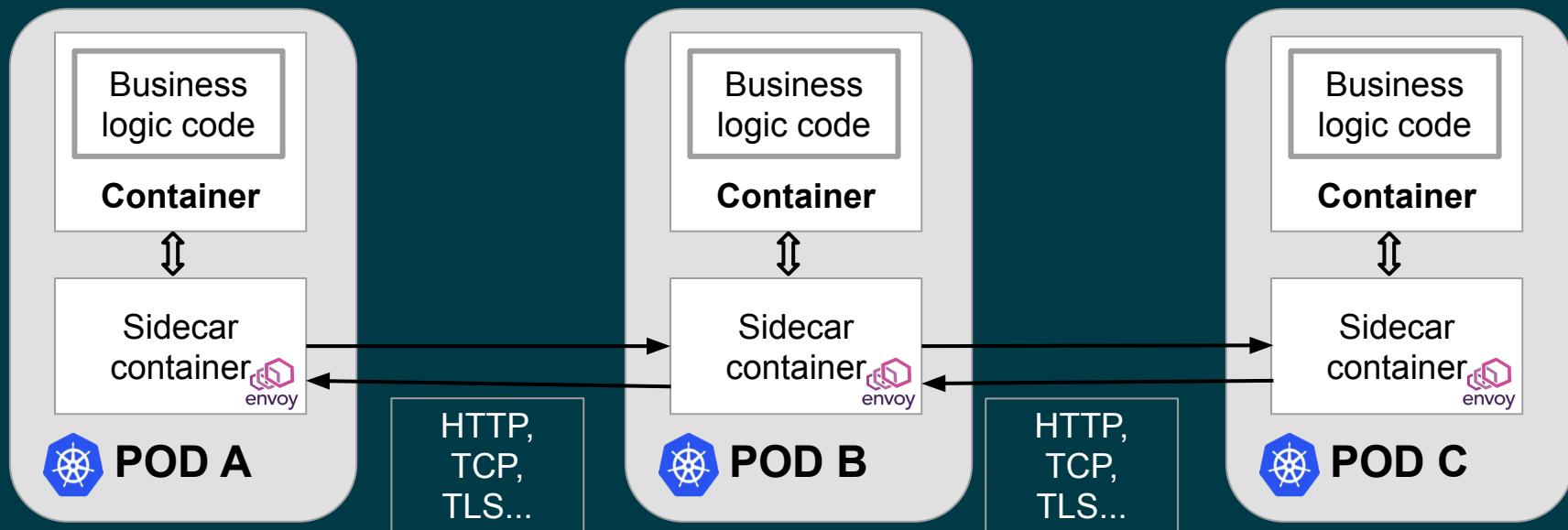


Before Istio, no sidecar



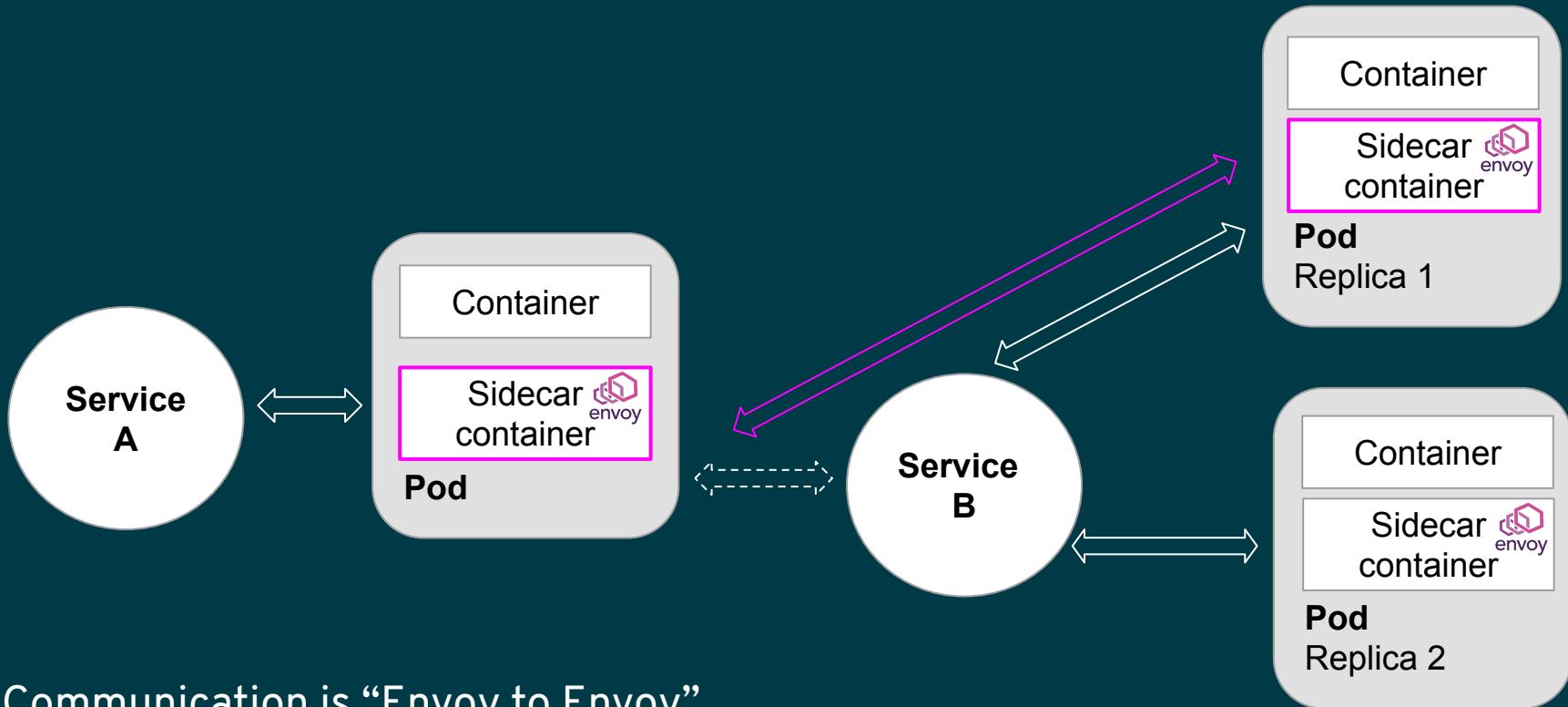
With sidecar

With Istio - sidecar intercepts all traffic



Configuration is transparent to the services and not part of the code

Istio routing in Kubernetes

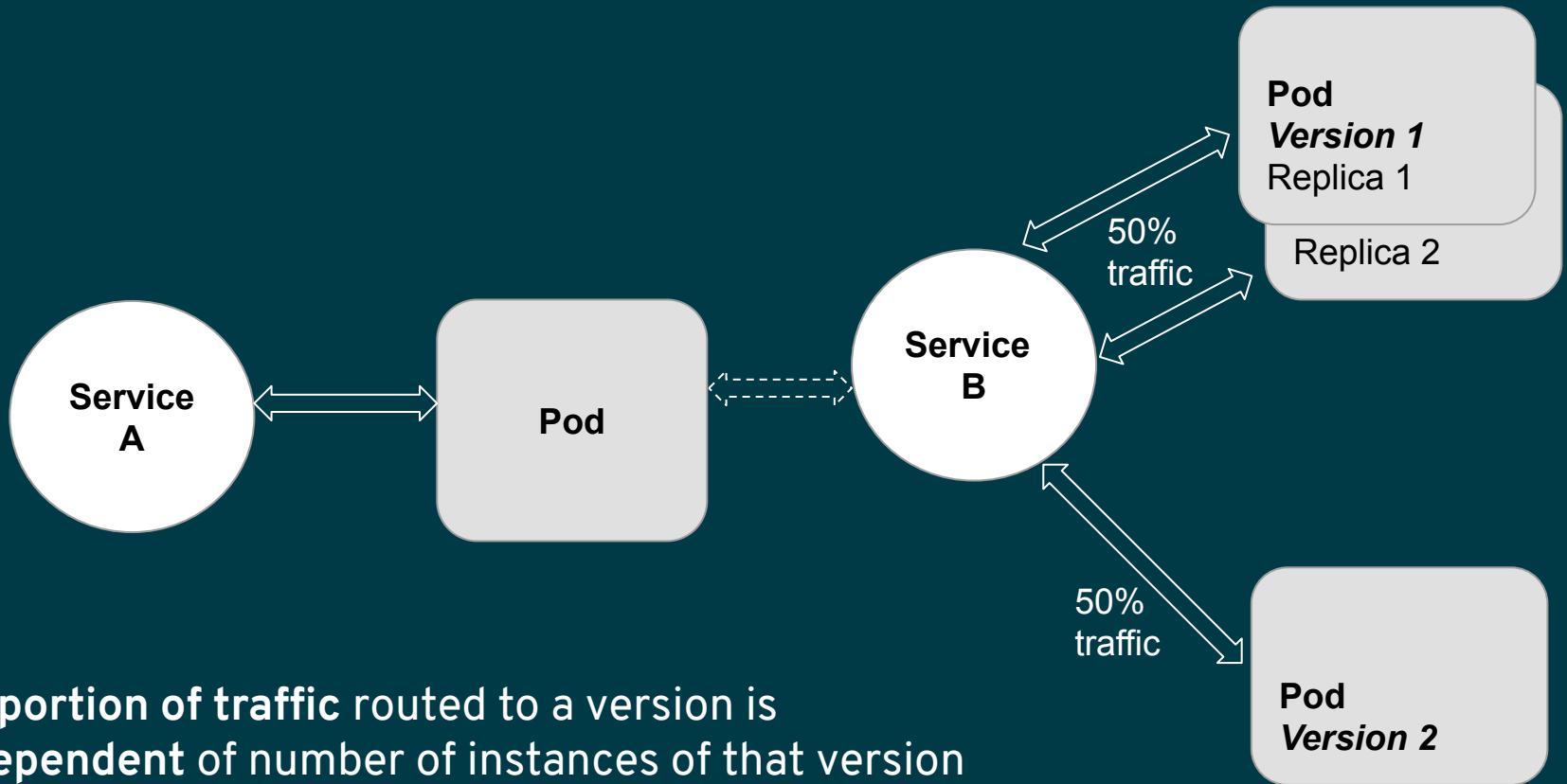


Communication is “Envoy to Envoy”
bypassing the Kubernetes Service

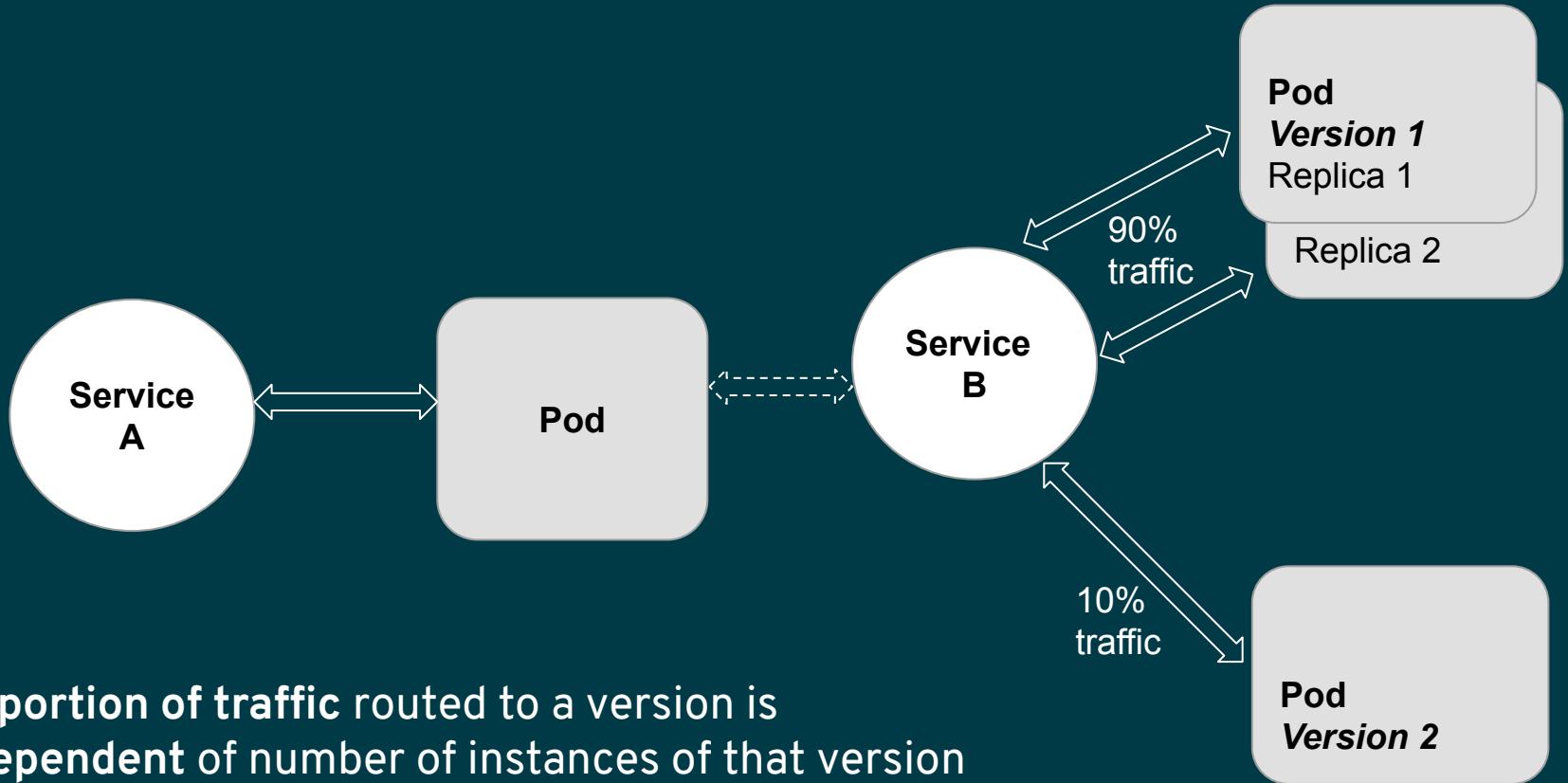
Different routing scenarios

- A/B testing
- Traffic shifting
 - Canary deployment (an example of traffic shifting)
- Mirroring traffic

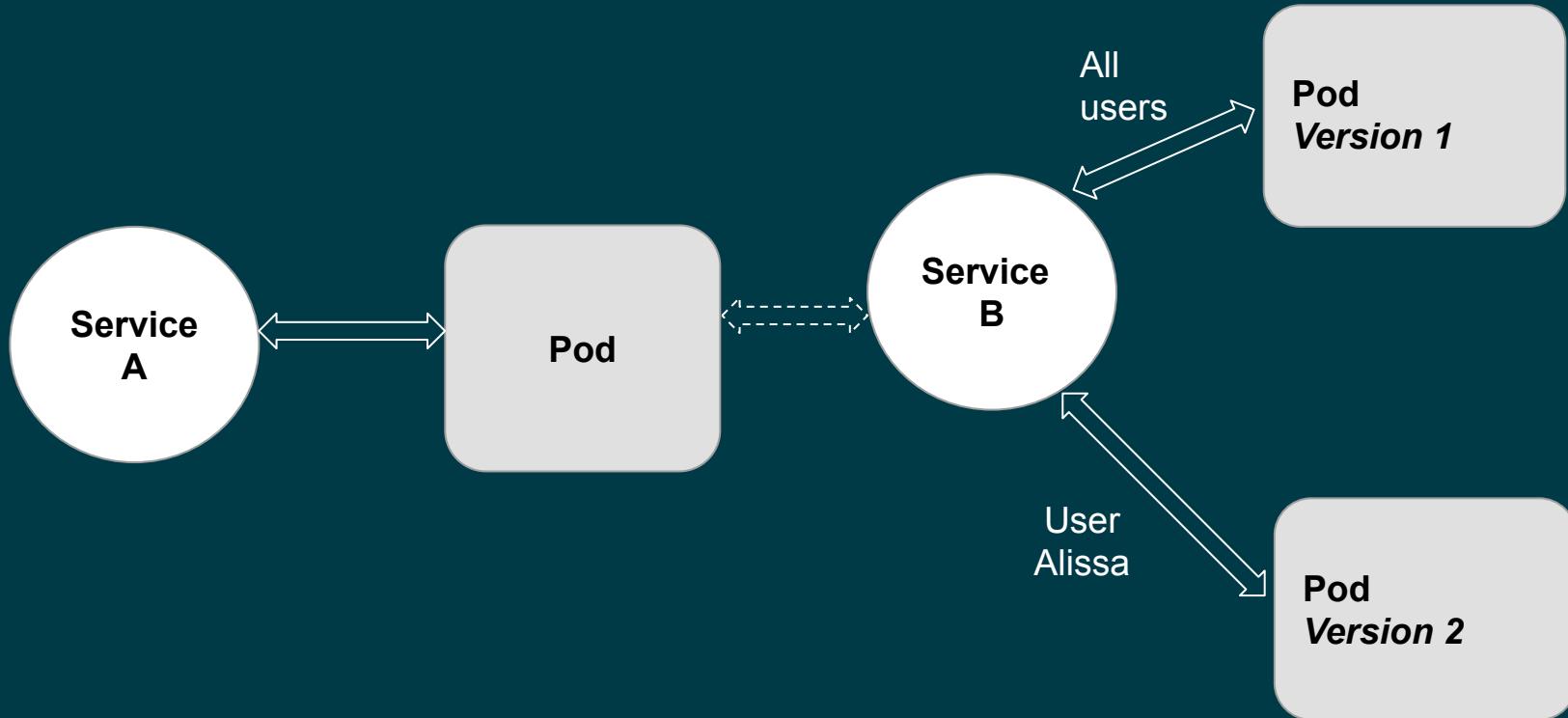
Weighted Routing with Istio - A/B



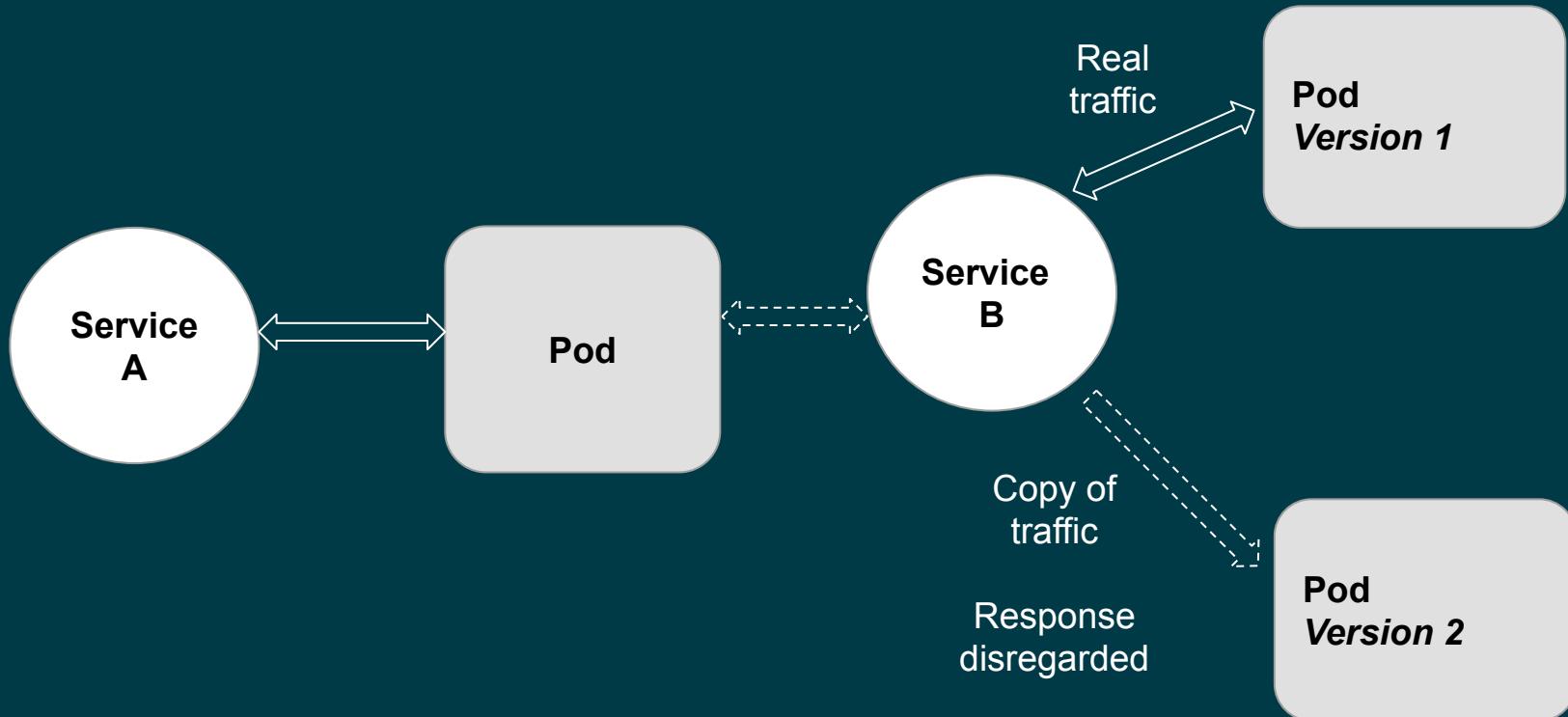
Weighted Routing - Canary



Matching Routing with Istio

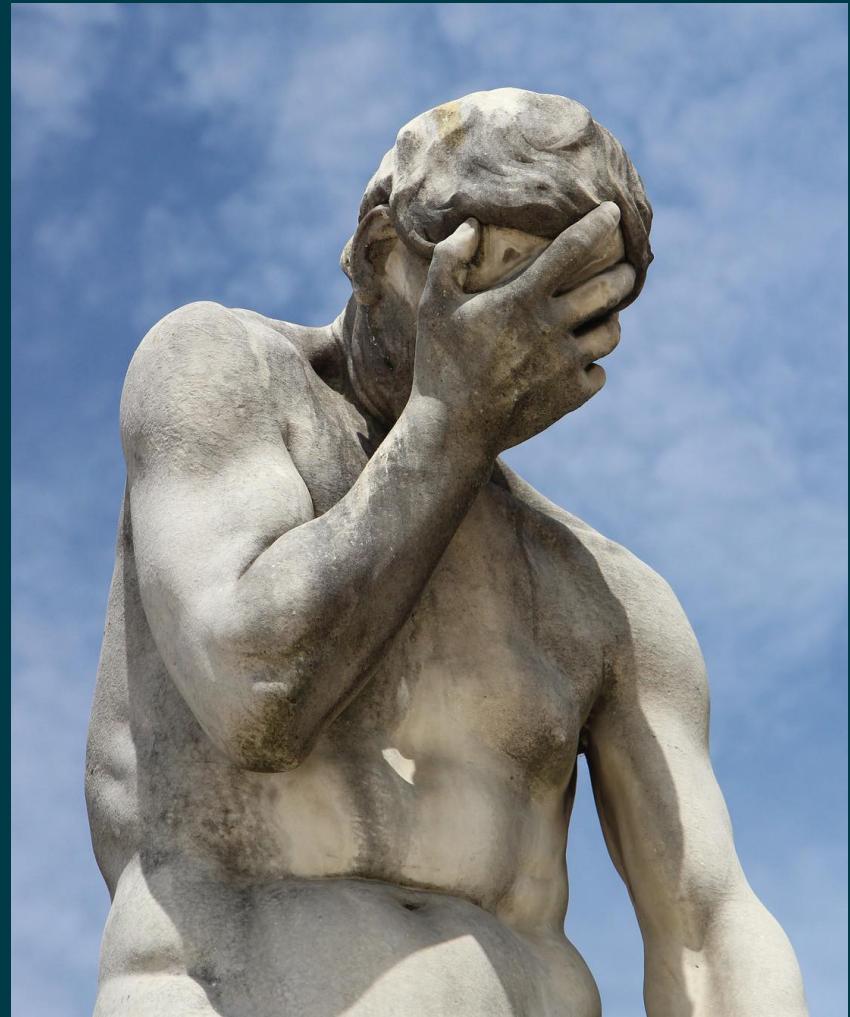


Mirroring traffic



"Anything that
can go wrong
will go wrong"

(Murphy's law)



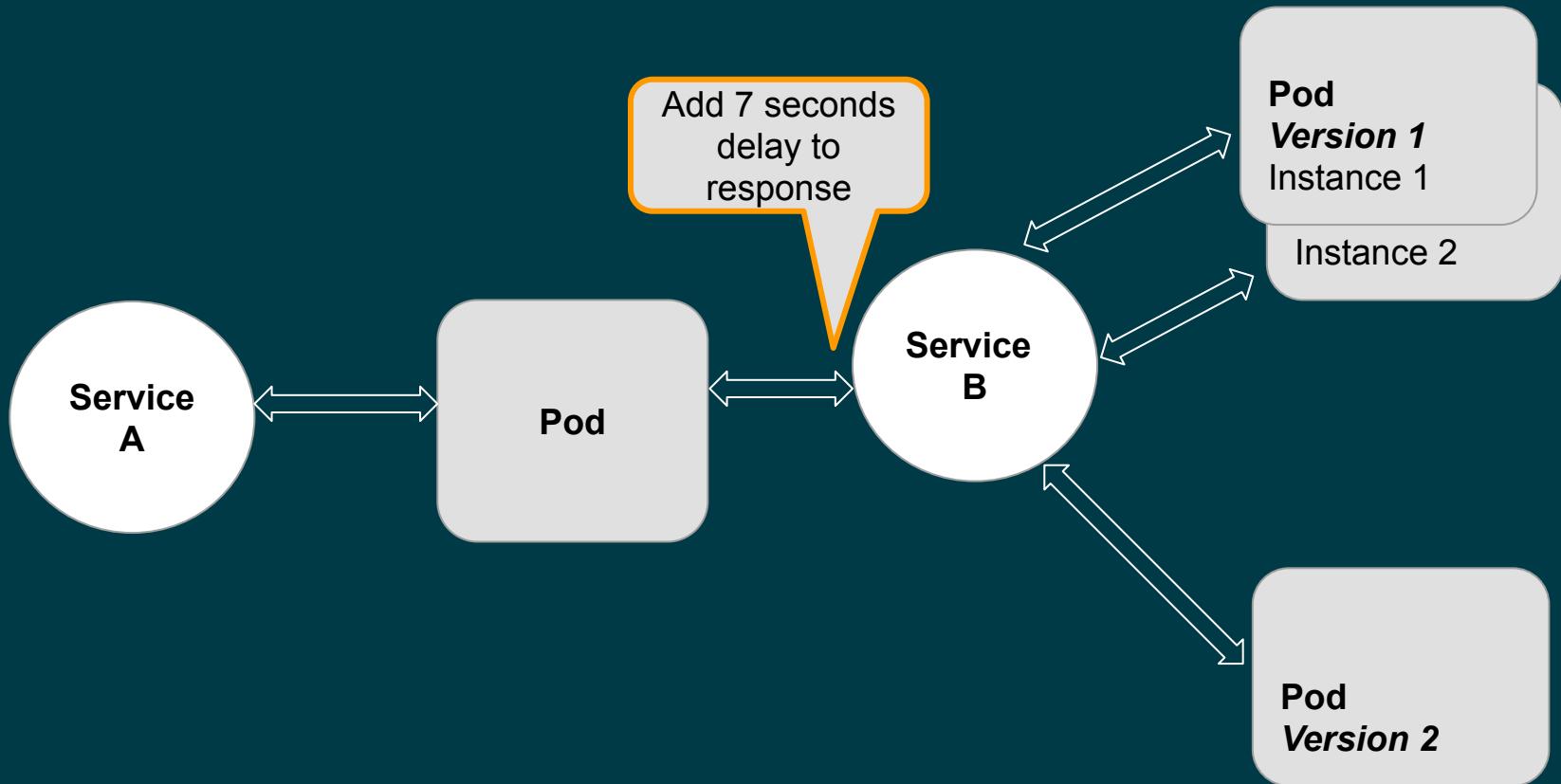


KEEP
CALM
AND
DO
CHAOS
ENGINEERING

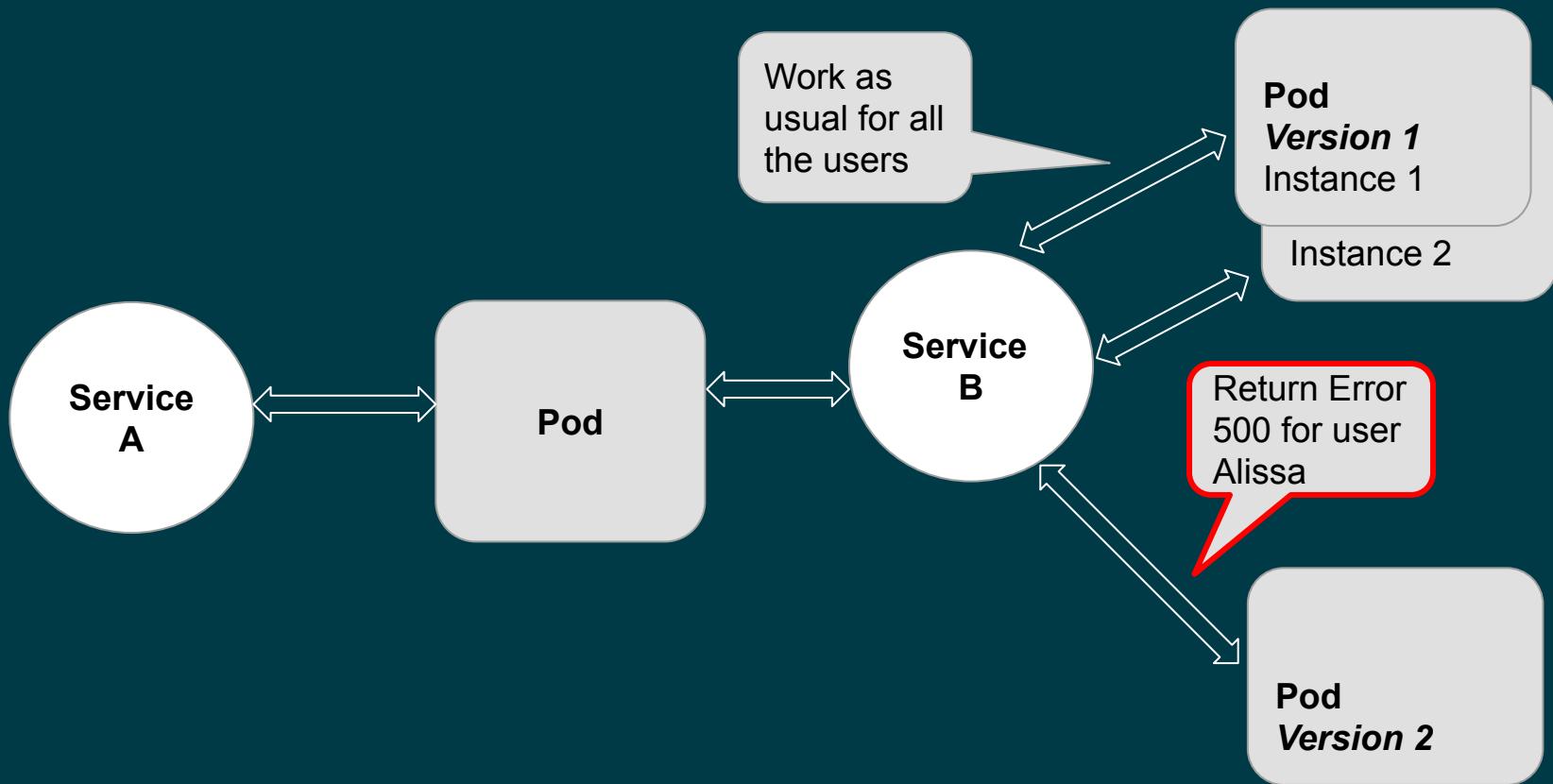
Chaos engineering with Istio

- Inject delays
 - Simulate network latency
 - Simulate an overloaded service
- Define aborts (Inject Errors)
 - Simulate failure in a service (return a predefined HTTP Error)
 - A good alternative for a manual shutdown or “scale to zero”

Inject delay



Inject Error



Circuit breaker

- Set a connection pool to limit connections and requests
- Example: “Set a connection pool of 100 connections with no more than 10 req/connection to service A”

Outlier detection

- Classify instances as healthy/unhealthy
- Eject unhealthy instances for a defined timeframe which can be increased over time
- Example: “Scan all pods every 5 mins, any instance that fails 7 consecutive times with 5XX error code will be ejected for 15 minutes.”

Authorization and Authentication

- Authentication
 - End user authentication (JSON Web Token (JWT))
 - Service to service authentication (mutual TLS)
 - Permissive mode is possible for flexible migration
- Authorization
 - Can service <A> send <this request> to service ?
 - Roles are visible across namespaces
 - ServiceRole and ServiceRoleBinding

Security

- Defining a Gateway ingress/egress to enable traffic in/out of mesh
- Citadel monitors service accounts creation and creates a certificate for them
 - Certificates only in memory, sent to Envoy via SDS API
- mTLS can be defined on multiple levels
 - Client and server exchange certificates, 2 way
 - All mesh, specific service, etc.



Configuration objects

- VirtualService != Kubernetes service
 - Rules for how requests to a service are routed within service mesh
 - Routing logic, load weighting, chaos injection
- DestinationRule
 - Configures policies to be applied to a request **after** VirtualService routing has occurred
 - Load balancer, circuit breaker
- MeshPolicy, Gateway, ServiceEntry and more...

Configuration Yaml example

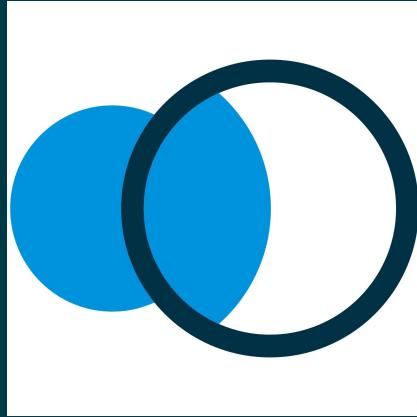
All Istio objects are
CRD
(CustomResource
Definition)

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
    - route:
        - destination:
            host: reviews
            subset: v1
            weight: 50
        - destination:
            host: reviews
            subset: v2
            weight: 25
        - destination:
            host: reviews
            subset: v3
            weight: 25
```

New set of challenges

- How many versions exist for service A?
- Is there any traffic **now**?
- Is **routing configured** for service B?
- Is my configuration **valid**?
- Is security **on**?
- Is the app **healthy**?





Kiali - Κιάλι

...

Open source
Istio service mesh observability



Dry facts

- Started in January 2018
- Means “spyglass” or “monocular” in Greek
- Developed in Go and React



Kiali Features

- Visualize mesh connections and traffic
- Service and application health
- Configure routing via UI
- Validate Istio configurations
- View metrics, traces and logs
- Visualize security configuration

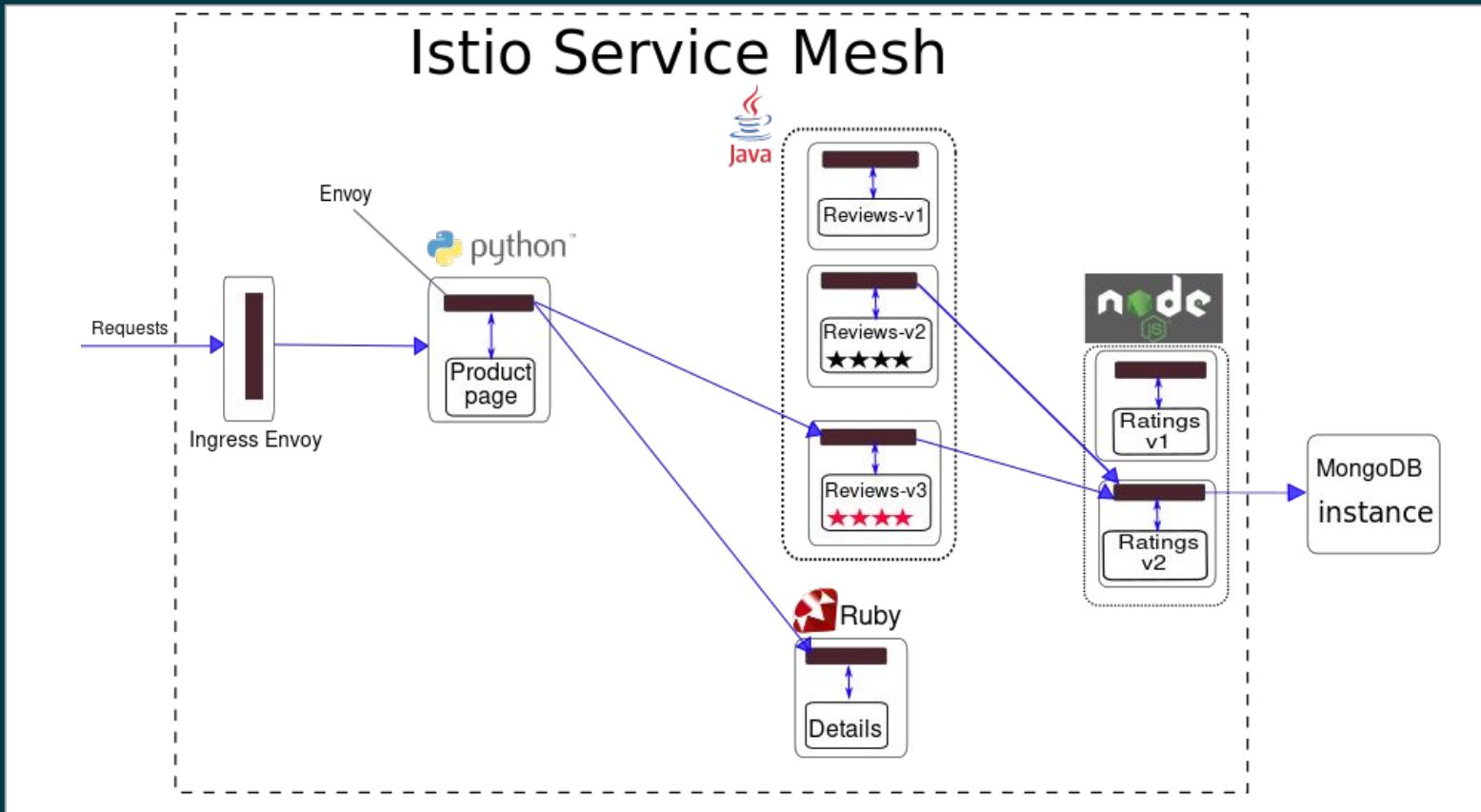
A picture is worth a thousand yaml's

Demos based on Bookinfo example

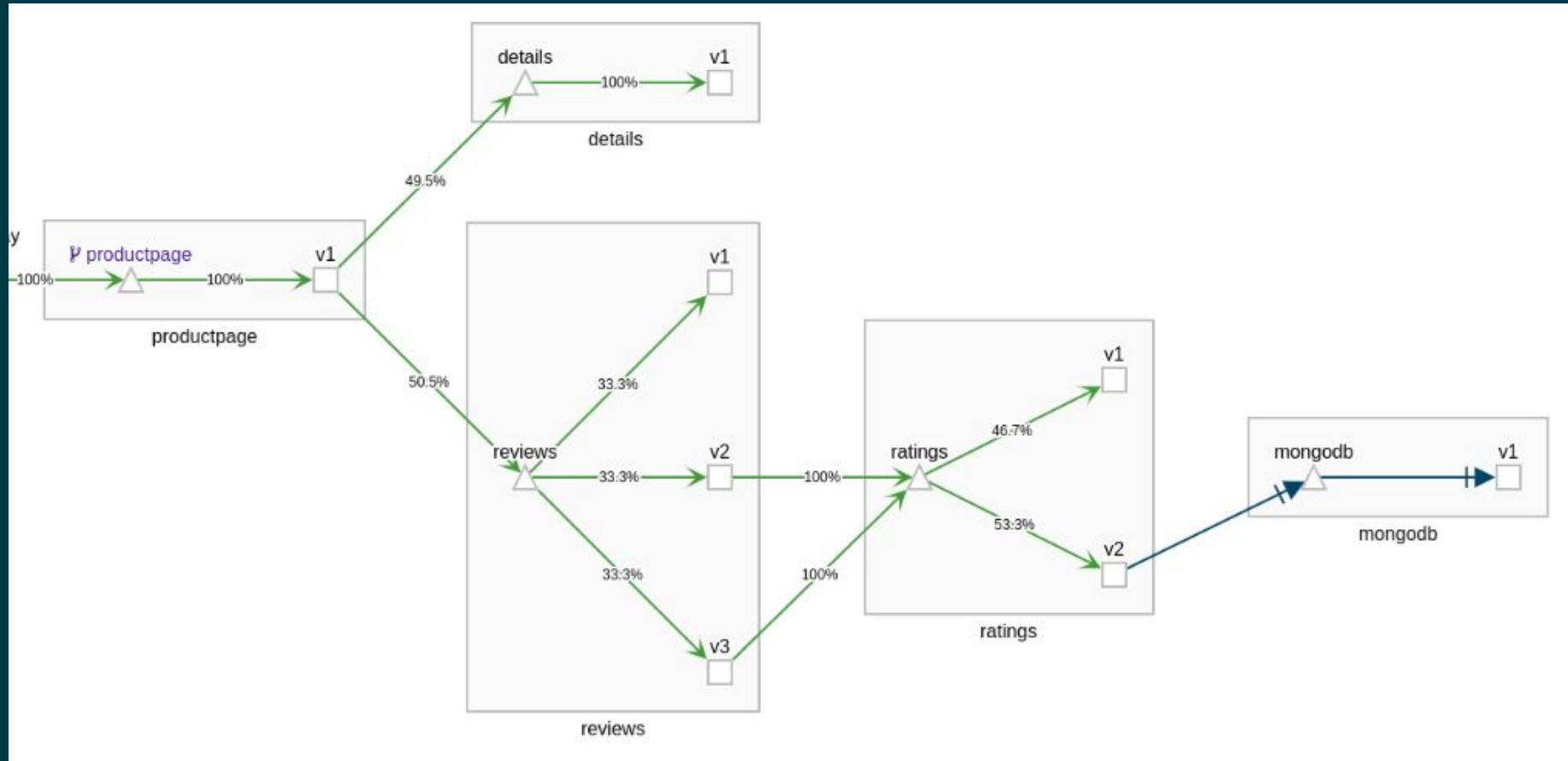
Let's see Kiali in action

- Mesh visualization
- Fault Injection
- Configuration Validation
- Configure routing rules
- Tracing
- Traffic stats

Bookinfo example



Bookinfo on Kiali



Kiali Features

Overview page

The screenshot shows the Kiali Overview page with the following details:

- Namespaces:** A list of namespaces with their respective application counts and traffic status.
- bookinfo:** 6 Applications, Traffic, last 1m. Status: 6 (green checkmark).
- default:** 0 Applications, N/A, No traffic.
- istio-system:** 14 Applications, Traffic, last 1m. Status: 14 (green checkmark).
- myproject:** 0 Applications, N/A, No traffic.

Filtering and Metrics: The top navigation bar includes filters for "Name" (dropdown), "Show health for" (dropdown set to "Apps"), "Compact" and "Expand" buttons, and time intervals "Last 1m" and "Every 15s".

Left Sidebar: Navigation menu with links: Overview, Graph, Applications, Workloads, Services, Istio Config, and Distributed Tracing.

Mesh Topology Graph

Namespace: bookinfo

Graph [?](#)

Versioned app graph Requests percentage Display Find... Hide... [?](#)

Last 1m Every 15s [?](#)

Overview Graph Applications Workloads Services Istio Config Distributed Tracing

Namespace: bookinfo applications, services, workloads

Current Graph:

- 9 apps
- 5 services
- 14 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
6.84	100.00	0.00

HTTP - Total Request Traffic min / max:
RPS: 3.07 / 12.40 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
Sent: 200.20 / 686.40 B/s
Received: 161.93 / 555.20 B/s

The diagram illustrates a mesh topology graph for the bookinfo namespace. It shows the flow of requests between various services, including productpage, reviews, ratings, and mongodb. The graph includes versioned app graphs (v1, v2, v3) and highlights specific metrics such as success rates and error percentages. A legend at the bottom provides icons for different service types and edge properties.

Hide and Seek

Namespace: bookinfo

May 22, 23:38:12 ... May 22, 23:39:12

Last 1m ▾ Every 15s ▾

Graph Overview Applications Workloads Services Istio Config Distributed Tracing

Versioned app graph Response time Display response time > 130 Hide... ?

Namespace: bookinfo applications, services, workloads

Current Graph:

- 9 apps
- 5 services
- 14 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
3.69	100.00	0.00

OK 3xx 4xx 5xx

HTTP - Total Request Traffic min / max:
RPS: 3.67 / 3.73 , %Error 0.00 / 0.00

TCP - Total Traffic - min / max:
Sent: 143.00 / 171.60 B/s
Received: 115.67 / 138.80 B/s

istio-ingressgateway (istio-system)

productpage

reviews

ratings

mongodb

v1 v2 v3

details

202ms 21ms 138ms 8ms 80ms 22ms

Legend: + - X X1 X2

Details Page

Services > Namespace: bookinfo > Service: details

⌚ details (Show on graph)

Overview Traffic Inbound Metrics Traces

Last 1m ▾ Actions ▾

Labels

app details service details

Type ClusterIP
IP 172.30.23.185
Created at 5/23/2019, 11:33:02 AM
Resource Version 44630

Ports

TCP http (9080)

Endpoints

172.17.0.21 : details-v1-74c4f8c9bf-rt68c

Health

🕒 Healthy
🕒 Error Rate over last 1m: 0.00%

Workloads (1) Virtual Services (0) Destination Rules (0)

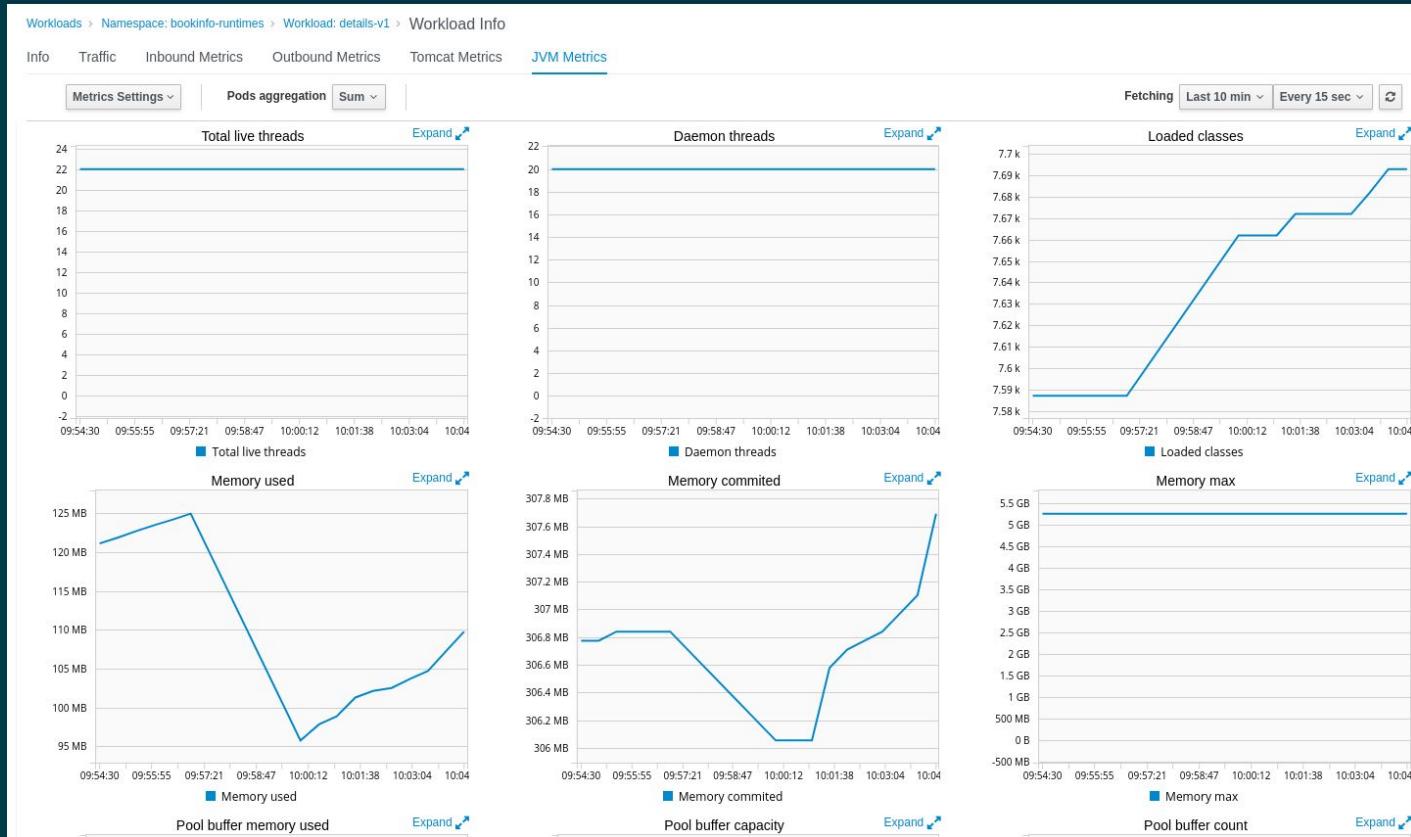
Name	Type	Labels	Created at	Resource version
details-v1	Deployment	app details version v1	5/23/2019, 11:33:02 AM	81259

Viewing Logs

The screenshot shows the Kiali interface for viewing logs. On the left, a sidebar navigation includes 'Overview', 'Graph', 'Applications', 'Workloads' (which is selected), 'Services', 'Istio Config', and 'Distributed Tracing'. The main content area shows a breadcrumb path: 'Workloads > Namespace: bookinfo > Workload: details-v1'. Below this, there's a button to 'Show on graph'. The tab bar at the top of the main content area includes 'Overview', 'Traffic', 'Logs' (which is selected), 'Inbound Metrics', and 'Outbound Metrics'. A sub-menu for 'Pod' shows 'details-v1-74c4f8c9bf-rt68c' and 'Container' with a dropdown menu. On the right, there are buttons for 'Tail' (set to 500 lines), 'Last 10m', and a refresh icon. The main pane displays a large list of log entries from the selected pod. Each entry consists of a timestamp, a log message, and a status code. The log messages indicate various HTTP requests to the '/details/0' endpoint.

Timestamp	Log Message	Status Code
2019-05-23T14:36:42.740174638Z	[23/May/2019:14:36:42 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:42.740209764Z	- - /details/0	
2019-05-23T14:36:42.805878473Z	[23/May/2019:14:36:42 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:42.80591068Z	- - /details/0	
2019-05-23T14:36:43.783127706Z	[23/May/2019:14:36:43 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:43.783158683Z	- - /details/0	
2019-05-23T14:36:43.834505658Z	[23/May/2019:14:36:43 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:43.834526797Z	- - /details/0	
2019-05-23T14:36:44.809577254Z	[23/May/2019:14:36:44 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:44.80959804Z	- - /details/0	
2019-05-23T14:36:44.863175939Z	[23/May/2019:14:36:44 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:44.863195834Z	- - /details/0	
2019-05-23T14:36:45.838997307Z	[23/May/2019:14:36:45 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:45.839028738Z	- - /details/0	
2019-05-23T14:36:45.890481856Z	[23/May/2019:14:36:45 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:45.89051259Z	- - /details/0	
2019-05-23T14:36:46.87859919Z	[23/May/2019:14:36:46 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:46.87863823Z	- - /details/0	
2019-05-23T14:36:46.919680618Z	[23/May/2019:14:36:46 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:46.919694638Z	- - /details/0	
2019-05-23T14:36:47.916498391Z	[23/May/2019:14:36:47 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:47.916529494Z	- - /details/0	
2019-05-23T14:36:47.949807626Z	[23/May/2019:14:36:47 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:47.949825834Z	- - /details/0	
2019-05-23T14:36:48.96120209Z	[23/May/2019:14:36:48 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:48.9613433Z	- - /details/0	
2019-05-23T14:36:49.027269864Z	[23/May/2019:14:36:49 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:49.027289795Z	- - /details/0	
2019-05-23T14:36:49.993673632Z	[23/May/2019:14:36:49 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:49.99368792Z	- - /details/0	
2019-05-23T14:36:50.058061113Z	[23/May/2019:14:36:50 UTC] "GET /details/0 HTTP/1.1"	200
2019-05-23T14:36:50.058084242Z	- - /details/0	

Runtime metric dashboards



Weighted Routing

Kiali dashboard showing the creation of a Weighted Routing rule for the bookinfo namespace.

Create Weighted Routing

WORKLOAD

- reviews-v1 (Traffic weight: 0%)
- reviews-v2 (Traffic weight: 37%)
- reviews-v3 (Traffic weight: 63%)

TRAFFIC WEIGHT

Labels

- app reviews service reviews

Type ClusterIP
IP 172.30.118.128
Created at 5/23/2019, 1:37:13 PM
Resource Version 7088

Evenly distribute traffic

Advanced Options

- TLS**: ISTIO_MUTUAL (selected)
- LoadBalancer**: ROUND_ROBIN
- Workloads (3)**: reviews-v1, reviews-v2, reviews-v3
- Virtual Services**

Resource version

7433
7436
7374

Actions

Health: Healthy
Error Rate over last 10m: 0.00%

Configuration validations

The screenshot shows the Kiali interface for validating an Istio configuration. The left sidebar includes links for Overview, Graph, Applications, Workloads, Services, Istio Config, and Distributed Tracing. The main area displays the path: Istio Config > Namespace: bookinfo > Istio Object Type: destinationrules > Istio Object: reviews. The 'YAML' tab is selected, showing the following YAML code:

```
1 kind: DestinationRule
2 apiVersion: networking.istio.io/v1alpha3
3 metadata:
4   name: reviews
5   namespace: bookinfo
6   selfLink: >
7     /apis/networking.istio.io/v1alpha3/namespaces/bookinfo/destinationrules/reviews
8   uid: fb5ae97b-7cd6-11e9-93a2-50b7b9deb8f30
9   resourceVersion: '23688'
10  generation: 1
11  creationTimestamp: '2019-05-22T21:17:14Z'
12  labels:
13    kiali_wizard: weighted_routing
14  spec:
15    host: reviews
16    trafficPolicy:
17      tls:
18        mode: ISTIO_MUTUAL
19    subsets:
20      - labels:
21        version: v1
22      This subset's labels are not found in any matching host.
23        - version: v2
24        - name: v2
25      - labels:
26        - version: v3
27        - name: v3
28
29
```

A red horizontal bar highlights the validation message for the first subset: "This subset's labels are not found in any matching host." At the bottom of the screen, there are buttons for Save, Reload, and Cancel.

Tracing (integration with Jaeger)

kiali

Services > Namespace: bookinfo > Service: reviews > Service Traces

Info Traffic Inbound Metrics Error Traces (14)

Lookback * Last lh Search Traces

Show Advanced Options

Duration

Time

14 Traces

Sort: Most Recent

istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage cbe9ac2 437.59ms

8 Spans 2 Errors

details.bookinfo (1) istio-ingressgateway (1) productpage.bookinfo (3) ratings.bookinfo (1) reviews.bookinfo (2)

Today | 1:38:01 pm
9 minutes ago

istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage 35b601b 414.79ms

8 Spans 2 Errors

details.bookinfo (1) istio-ingressgateway (1) productpage.bookinfo (3) ratings.bookinfo (1) reviews.bookinfo (2)

Today | 1:38:00 pm
9 minutes ago

istio-ingressgateway: productpage.bookinfo.svc.cluster.local:9080/productpage c624906 2.04s

This screenshot shows the Kiali tracing interface for the reviews service within the bookinfo namespace. The main dashboard displays a timeline of trace spans as colored dots, with a zoomed-in view of a specific error trace below. The trace details show 14 spans across various services like details, reviews, and ratings, with two errors identified. The interface includes a sidebar with navigation links for Overview, Graph, Applications, Workloads, Services, Istio Config, and Distributed Tracing.

Visualizing security

Screenshot of the Kiali dashboard showing namespaces and their security status.

Namespaces:

- bookinfo**: 6 Applications, Traffic, last 1m. Status: ✓ 6. Icons: 🔒, 📡, 🚧, ⚙️, 🌐, 📈.
- default**: 0 Applications, N/A. Status: N/A. Icons: 🔒, 📡, 🚧, ⚙️, 🌐, 📈.
- istio-system**: 14 Applications, Traffic, last 1m. Status: ✓ 14. Icons: 🔒, 📡, 🚧, ⚙️, 🌐, 📈.
- myproject**: 0 Applications, N/A. Status: N/A. Icons: 🔒, 📡, 🚧, ⚙️, 🌐, 📈.

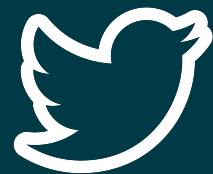
Top right corner: Mesh-wide mTLS is enabled, admin, Last 1m, Every 15s.

Left sidebar navigation:

- Overview (selected)
- Graph
- Applications
- Workloads
- Services
- Istio Config
- Distributed Tracing

Connect with the community

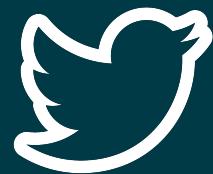
Kiali.io



KialiProject

github.com/kiali

Istio.io



IstioMesh

github.com/istio

Icon credits

- Twitter by Lubos Volkov, the Noun Project
- [Light Bulb](#) by artworkbean, the Noun Project
- [Magnifying Glass](#) by Musket from the Noun Project
- [Questions](#) by Rediffusion from the Noun Project
- [Mug](#) by Alex Getty from the Noun Project
- [Diamond](#) by MarkieAnn Packer from the Noun Project
- [Box](#) by Cornelius Danger from the Noun Project



THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

JAPAN

Thank you!



mikeyteva

Introduction to service mesh with Istio and Kiali

...

Alissa Bonas



mikeyteva