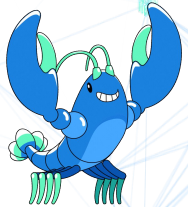
BUOYANT

Creators of LINKERD

Federated Services

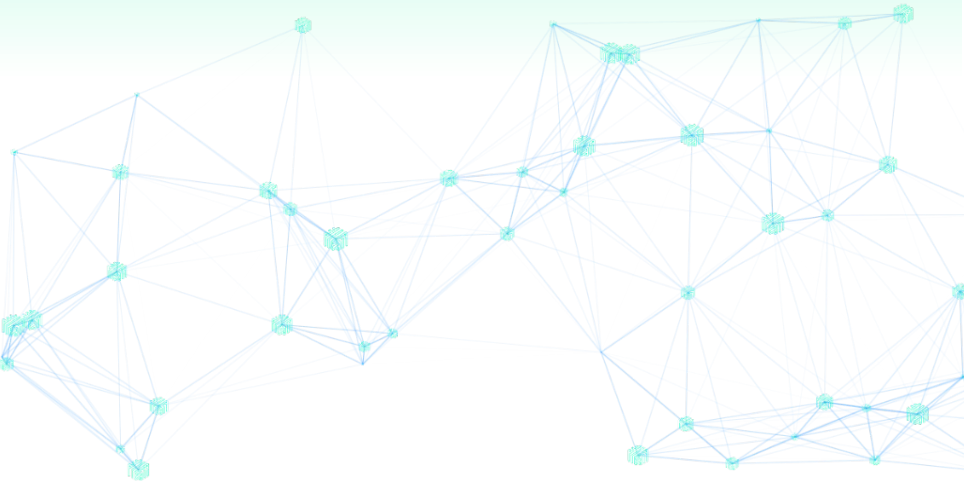
Flynn, Technical Evangelist for Linkerd



@BuoyantIO

buoyant.io

What's on the agenda?



- **Kubernetes Services**
- **Federated Services**
  - and why should you care 🙂
- **DEMO!**
- **Gotchas**

BUOYANT

How do you follow along?

- https://github.com/BuoyantIO/service-mesh-academy/tree/main/federated-services
- **For this demo, I'll be using need Buoyant Enterprise for Linkerd 2.18.2!**
  - Linkerd edge-25.4.4 or later will work too
- I'll be using a pair of `kind` clusters, but the main requirement here is direct pod-to-pod networking between your clusters.

BUOYANT

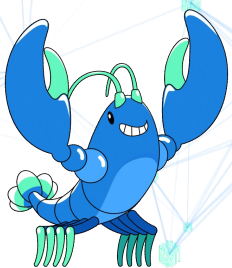
How do you follow along?

- **kind**
  https://kind.sigs.k8s.io/
- **kubectl**
  https://kubernetes.io/docs/tasks/tools/
- **linkerd CLI**
  https://linkerd.io/2/getting-started
- **bat**
  https://github.com/sharkdp/bat
- **jq**
  https://github.com/jqlang/jq
- **yq**
  https://github.com/mikefarah/yq

BUOYANT

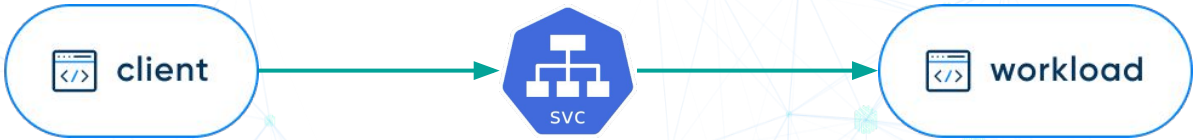
BUOYANT
SERVICE MESH
ACADEMY

Kubernetes Services



What is a Service?

A Kubernetes Service is a rendezvous point between a workload that offers functionality and clients that want to use that functionality.



What is a Service?

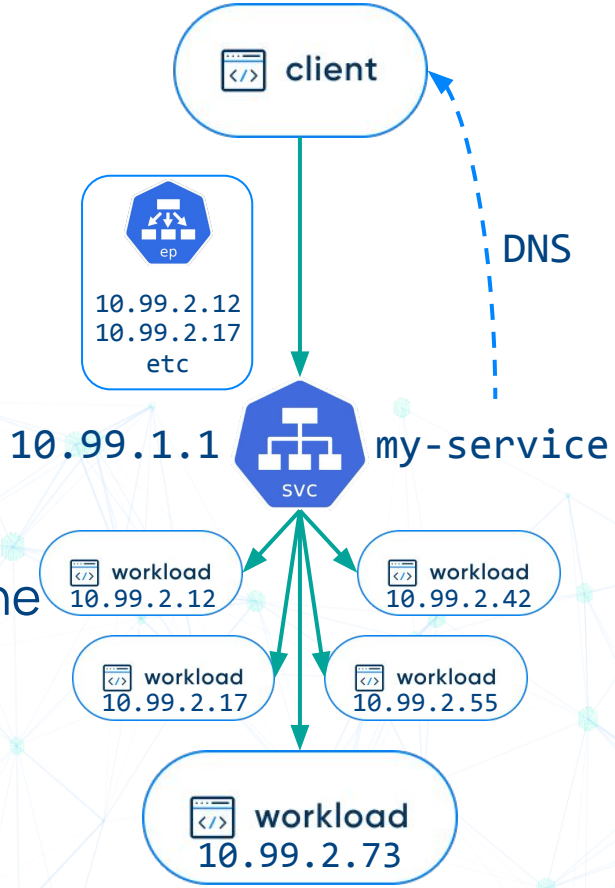
A Kubernetes Service is a rendezvous point between a workload that offers functionality and clients that want to use that functionality.

Service is also arguably the worst, most overloaded, most confusing API in Kubernetes. 😞

What's so bad about Service?

Services have at least *five* different functions.

- They give us a name to refer to our rendezvous point.
- The name of the Service appears in the cluster's DNS.
- The Service itself gets an IP address.
- The Service collects all the IP addresses of the workload replicas (*endpoints*).
- The Service IP and its endpoints are used to configure `kube-proxy`.



What's so bad about Service?

These functions span many different areas.

- They give us a name to refer to our rendezvous point. ⇐ Normal resource stuff
- The name of the Service appears in the cluster's DNS. ⇐ DNS
- The Service itself gets an IP address. ⇐ IPAM
- The Service collects all the IP addresses of the workload replicas (*endpoints*). ⇐ IPAM / resource stuff
- The Service IP and its endpoints are used to configure `kube-proxy`. ⇐ CNI

What is a Service?

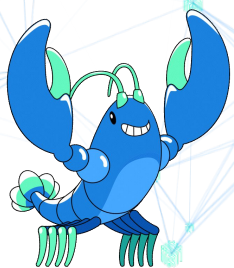
In other words, our “simple” rendezvous concept is anything but simple to implement!

Fortunately, application developers *usually* don’t have to worry about all of this.

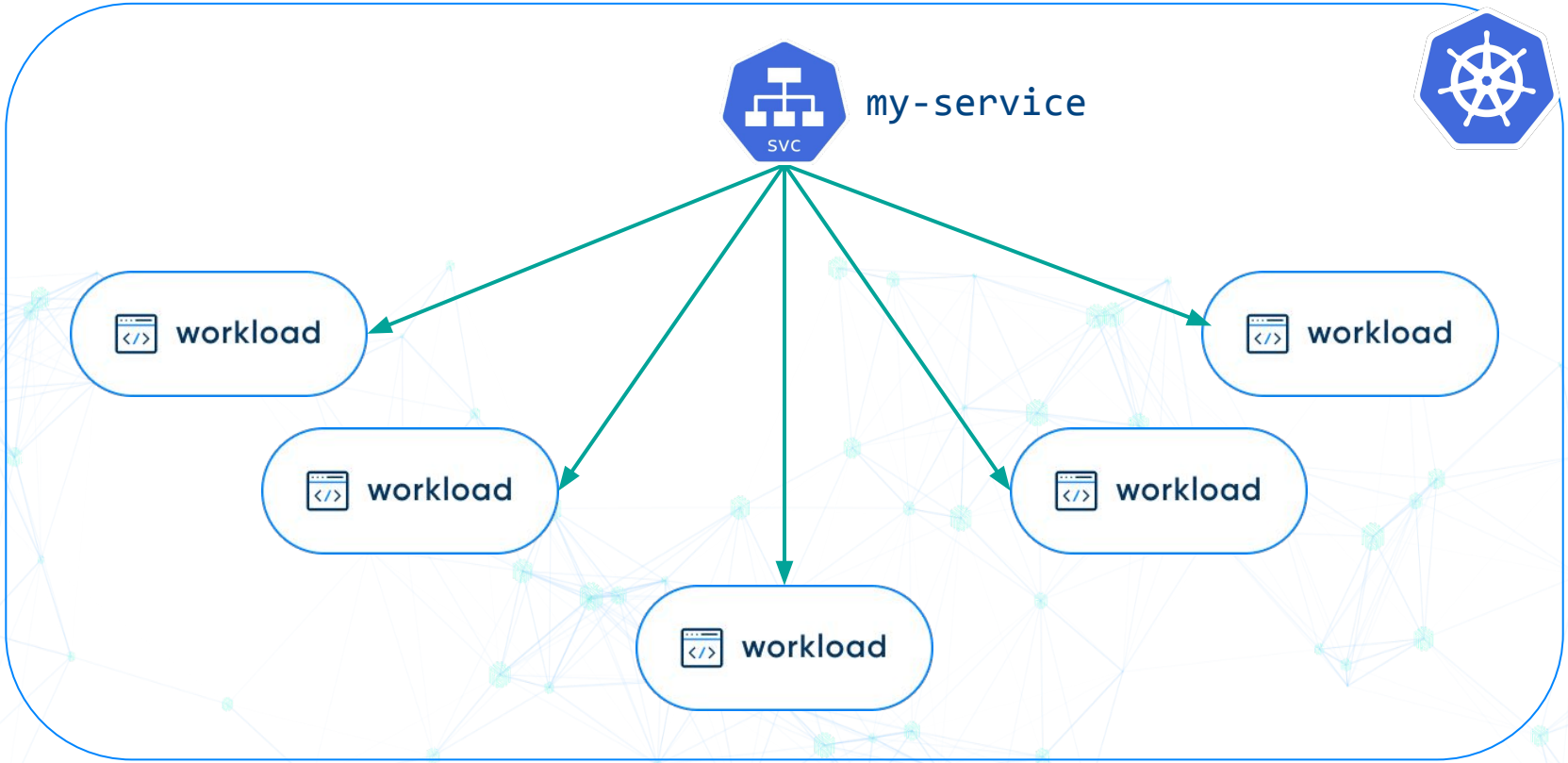
- Basically everyone implementing Kubernetes networking stuff does, though.
  - Meshes
  - Gateway API
  - Ingress controllers
  - etc.

BUOYANT
SERVICE MESH
ACADEMY

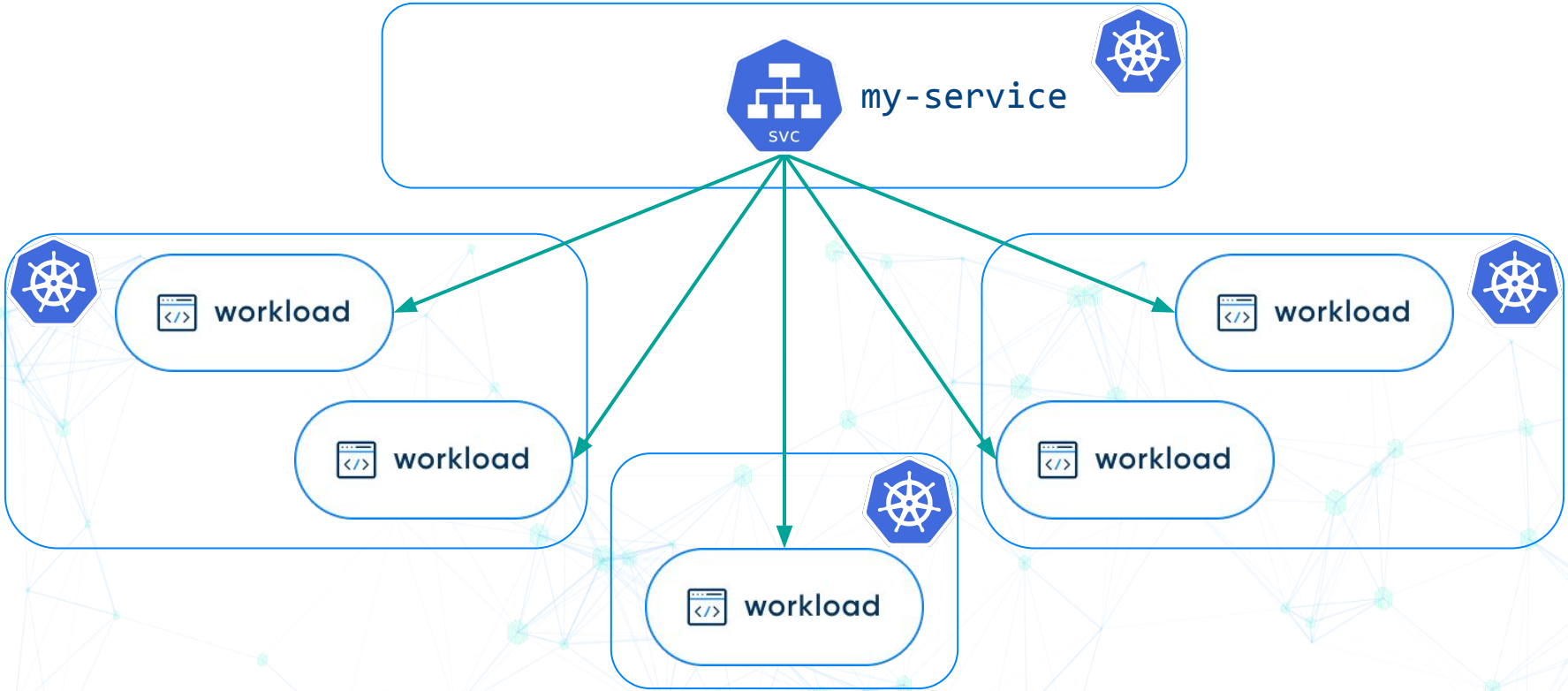
Federated Services



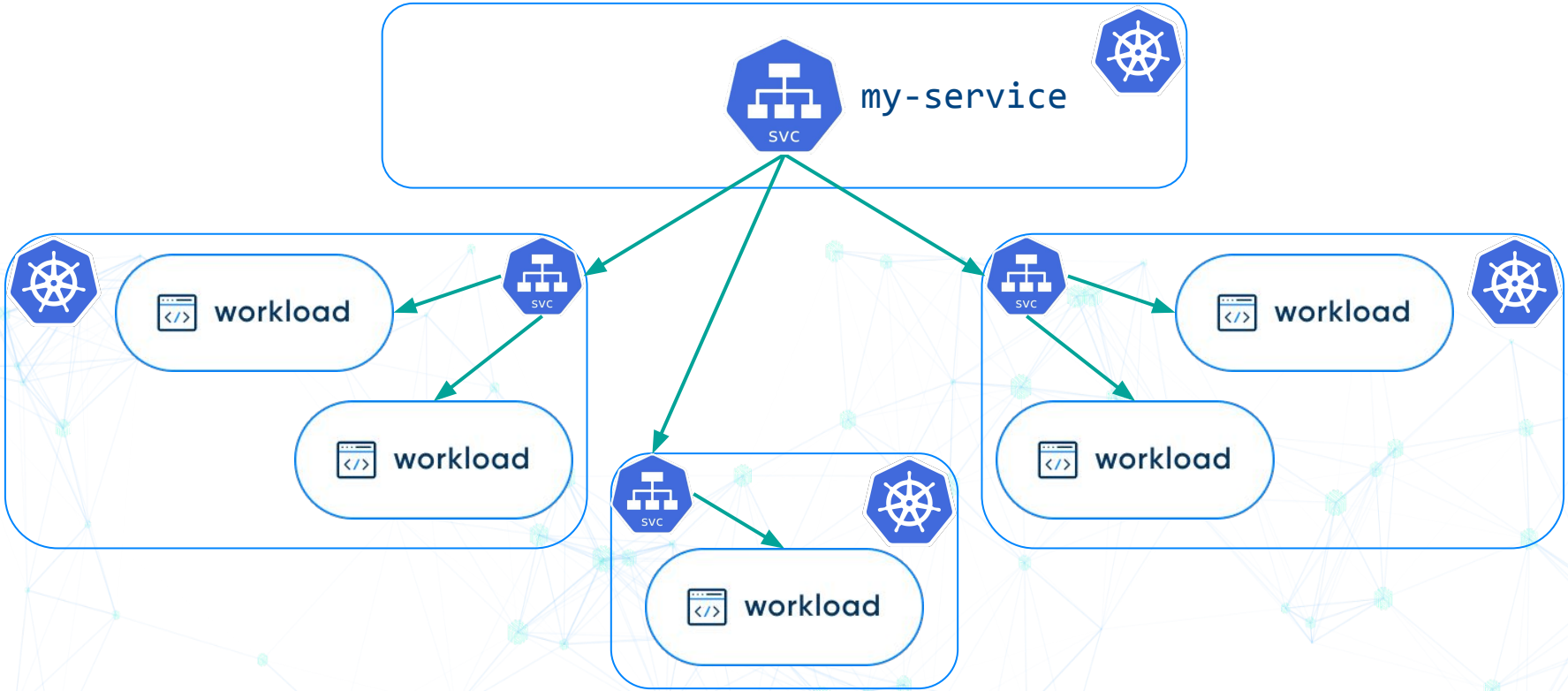
Federated Services



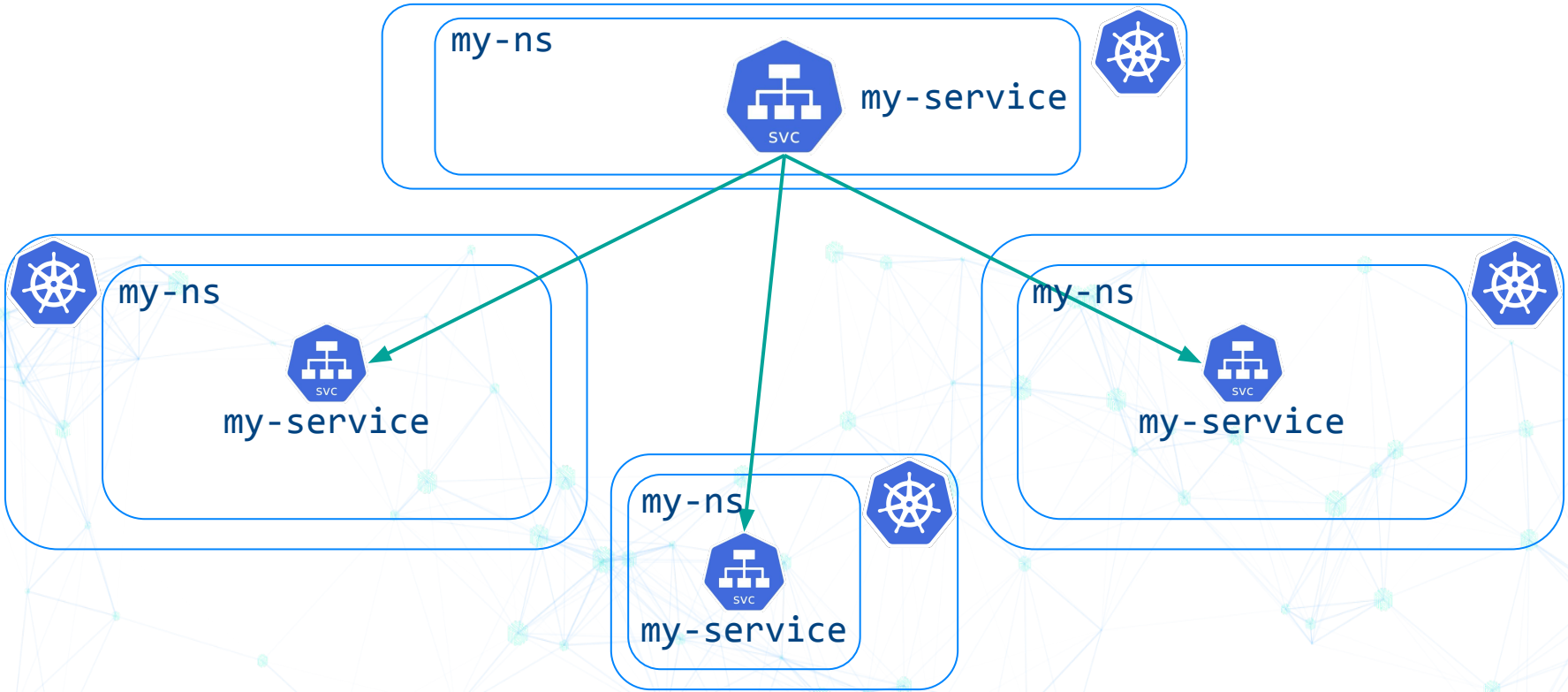
Federated Services



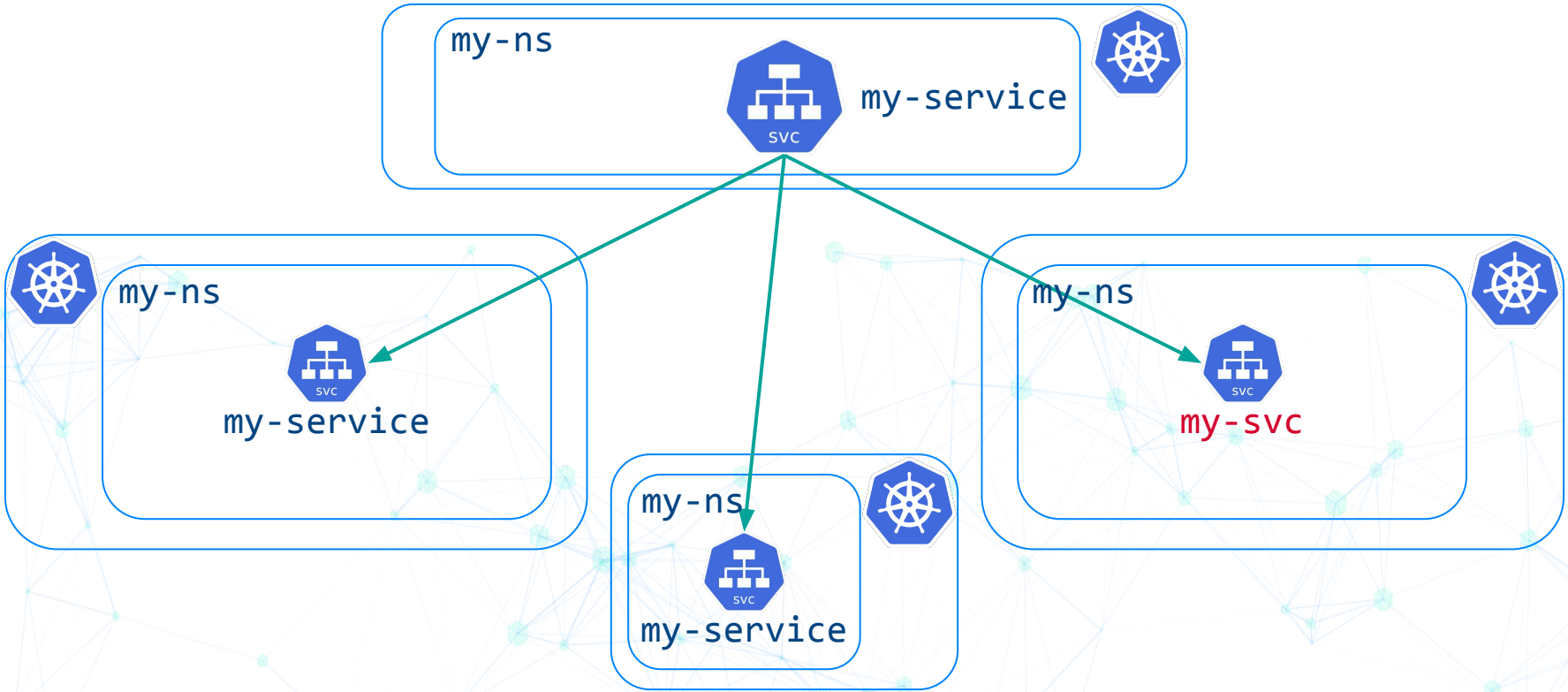
Federated Services: Discovery



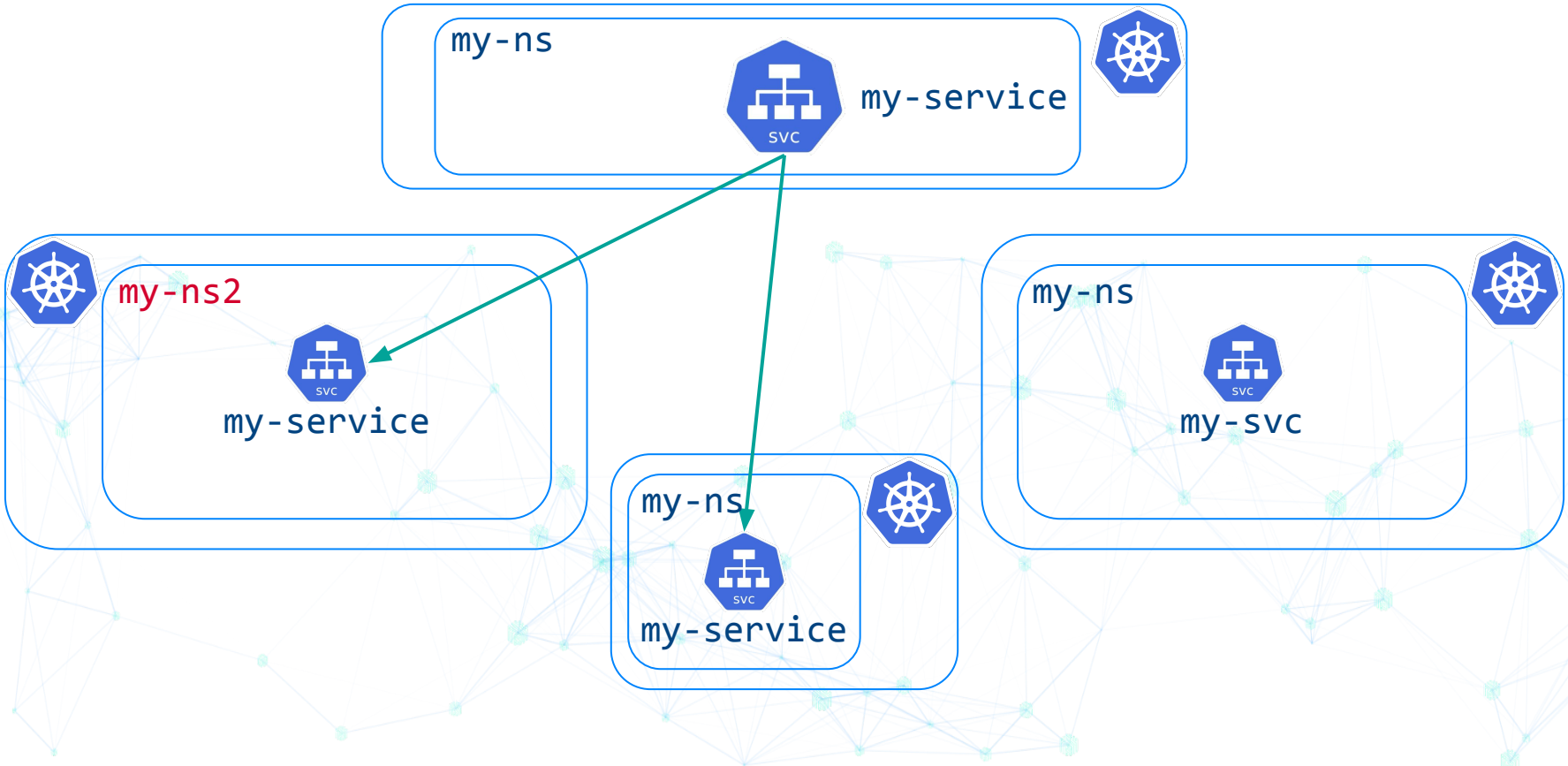
Federated Services: Namespace Sameness



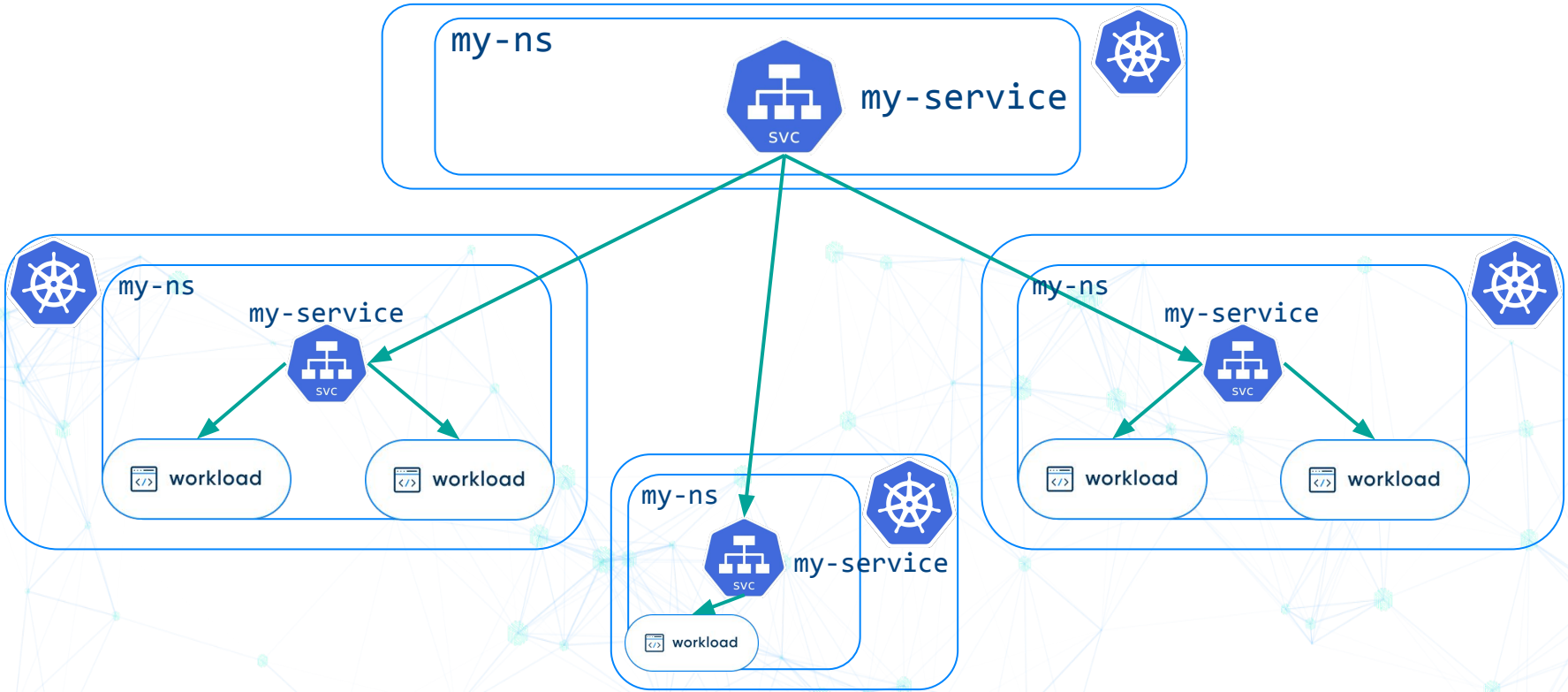
Federated Services: Namespace Sameness



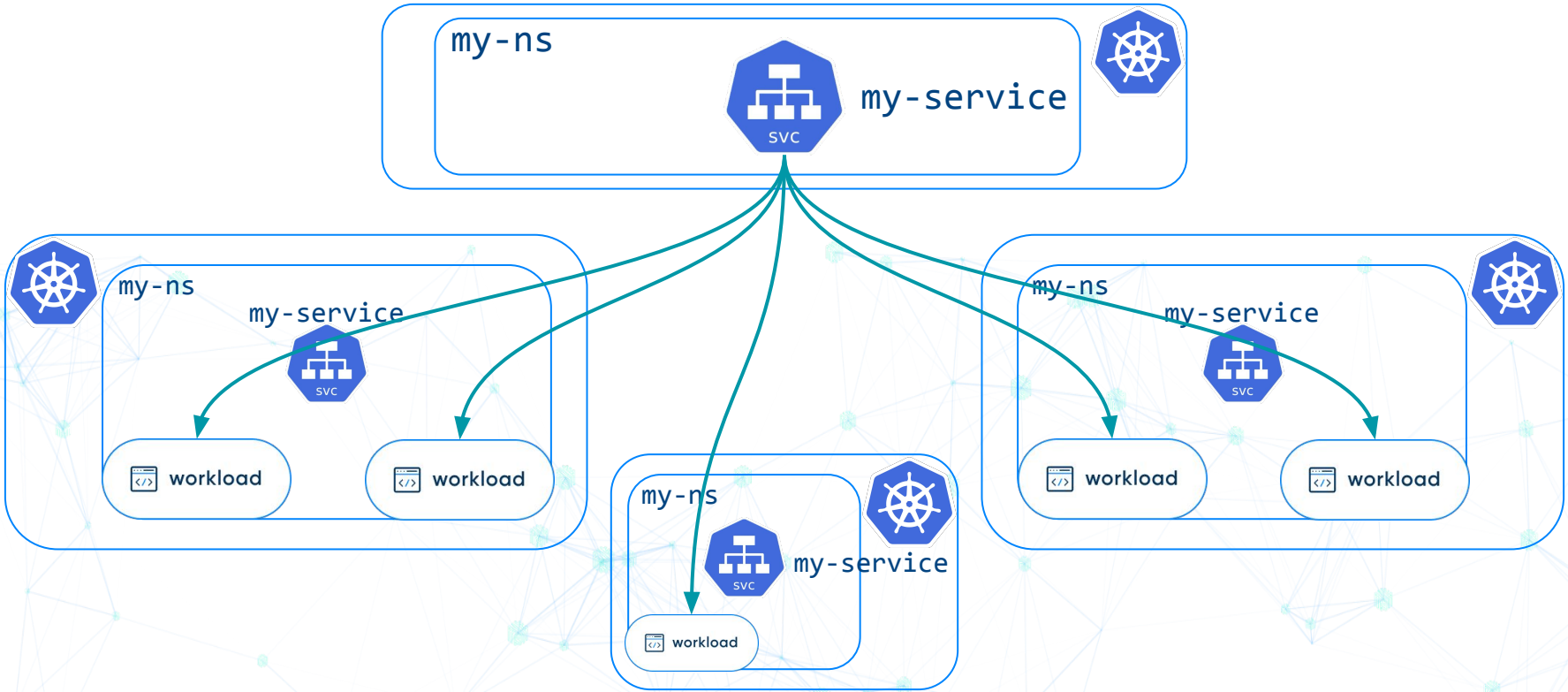
Federated Services: Namespace Sameness



Federated Services: Discovery vs Routing

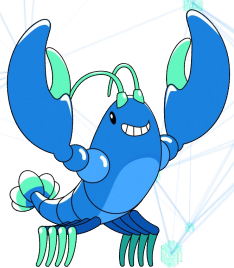


Federated Services: Discovery vs Routing

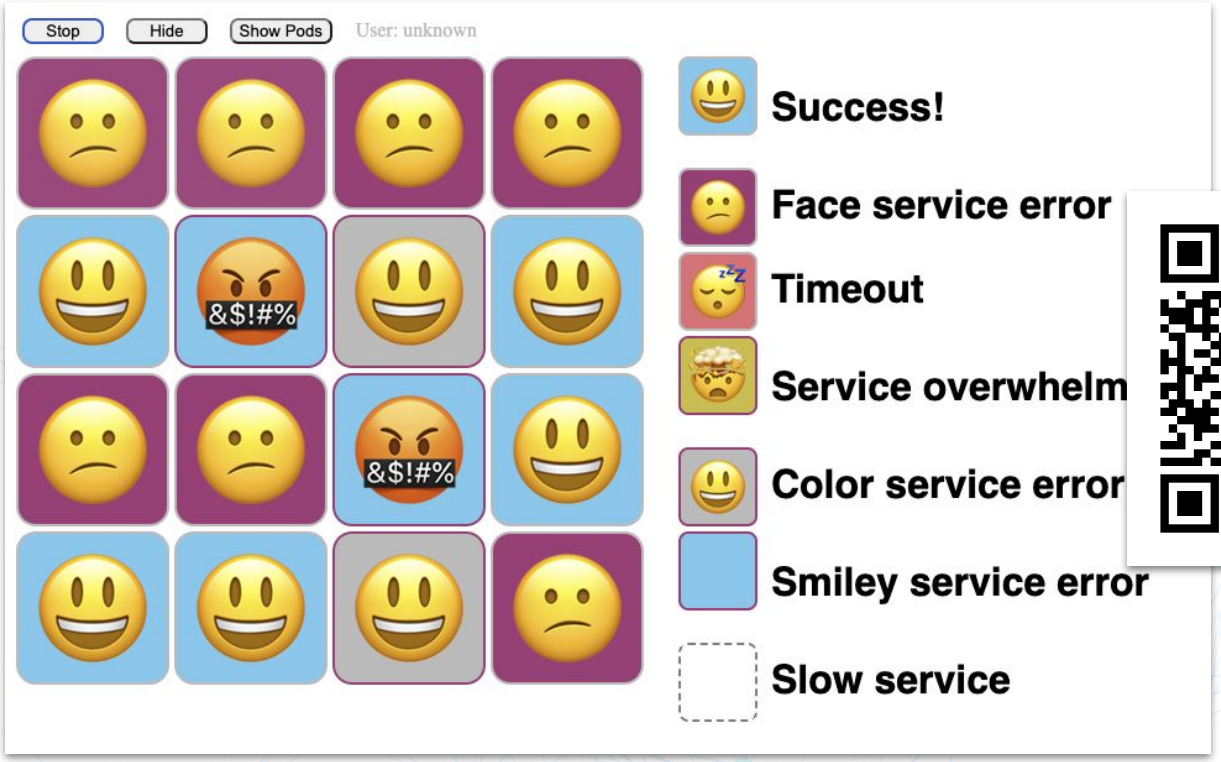


BUOYANT
SERVICE MESH
ACADEMY

Demo Architecture

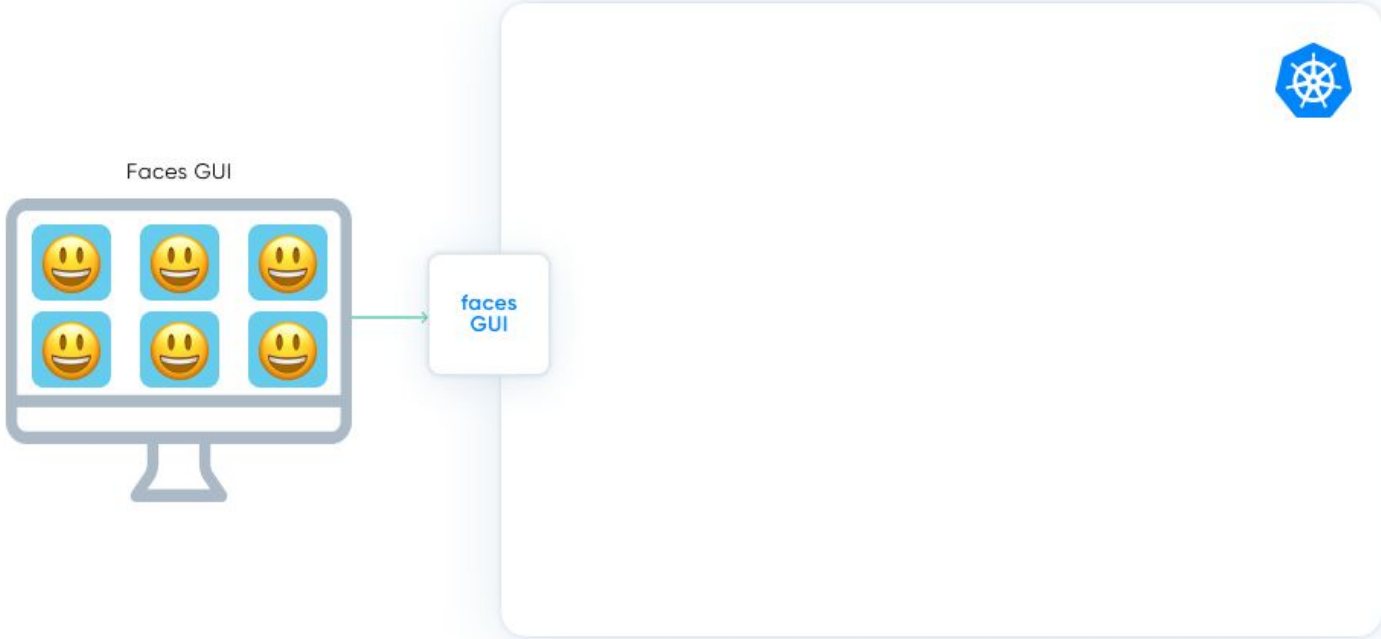


Faces (http://github.com/BuoyantIO/faces-demo)

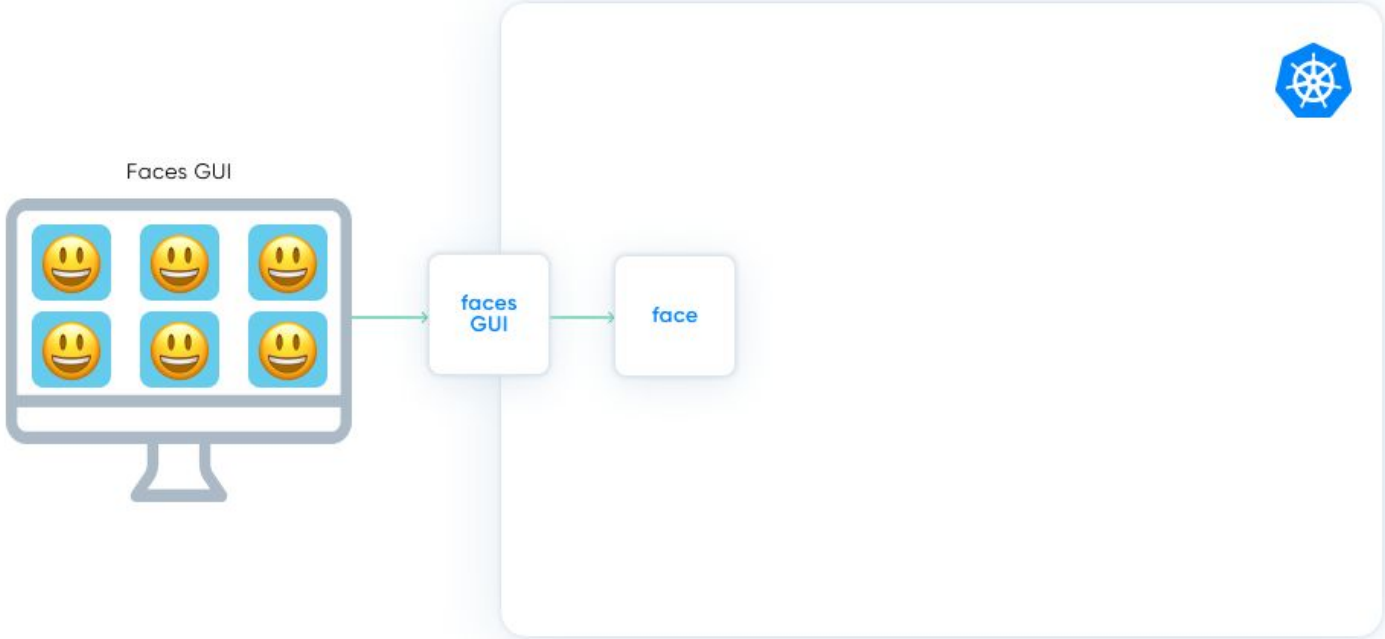


Faces (http://github.com/BuoyantIO/faces-demo)

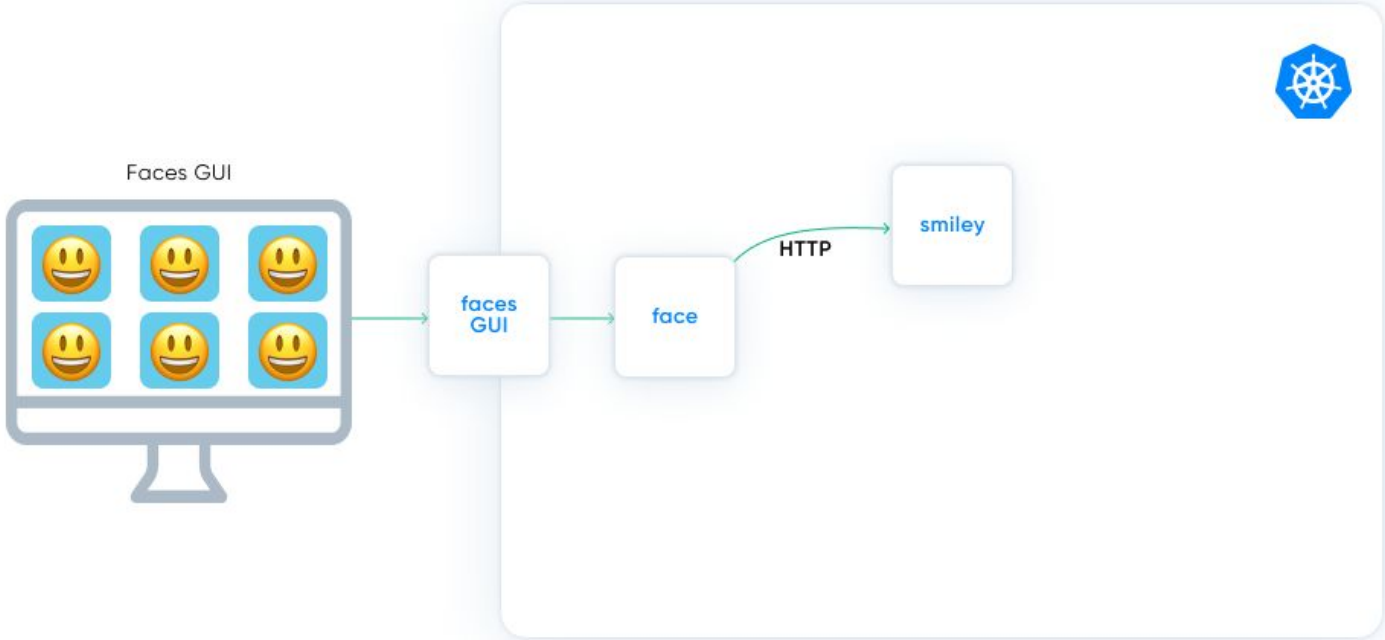
Faces (http://github.com/BuoyantIO/faces-demo)



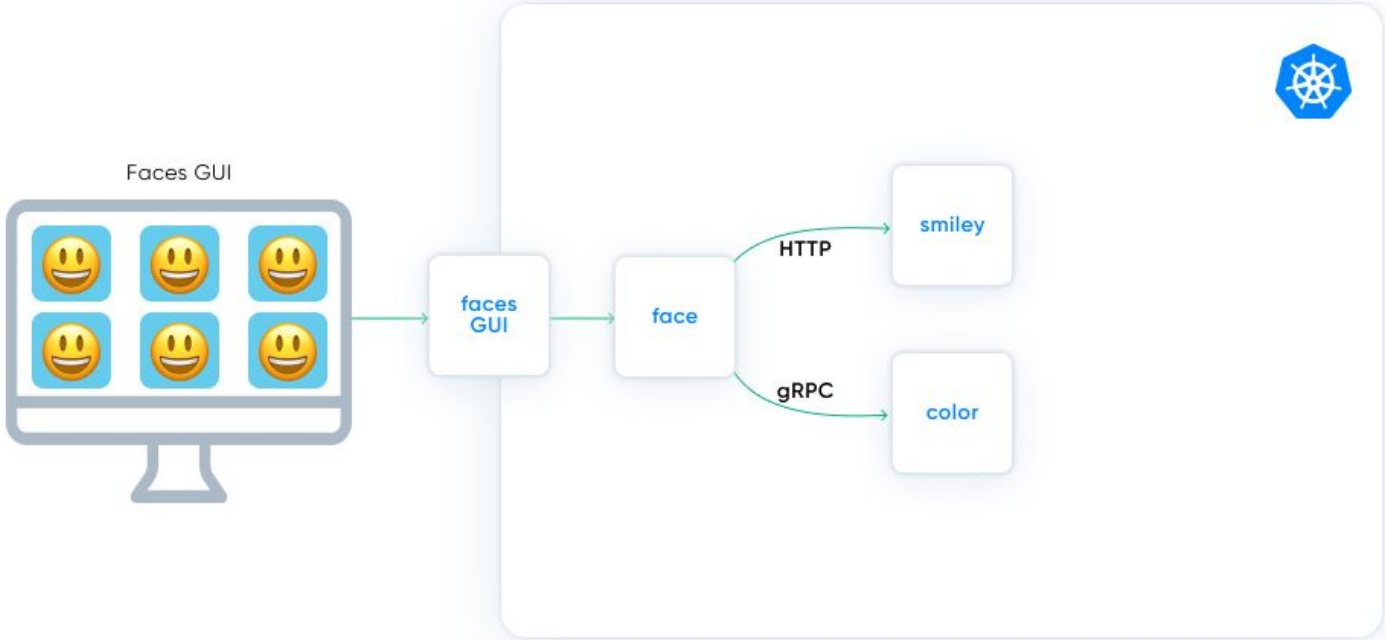
Faces (http://github.com/BuoyantIO/faces-demo)



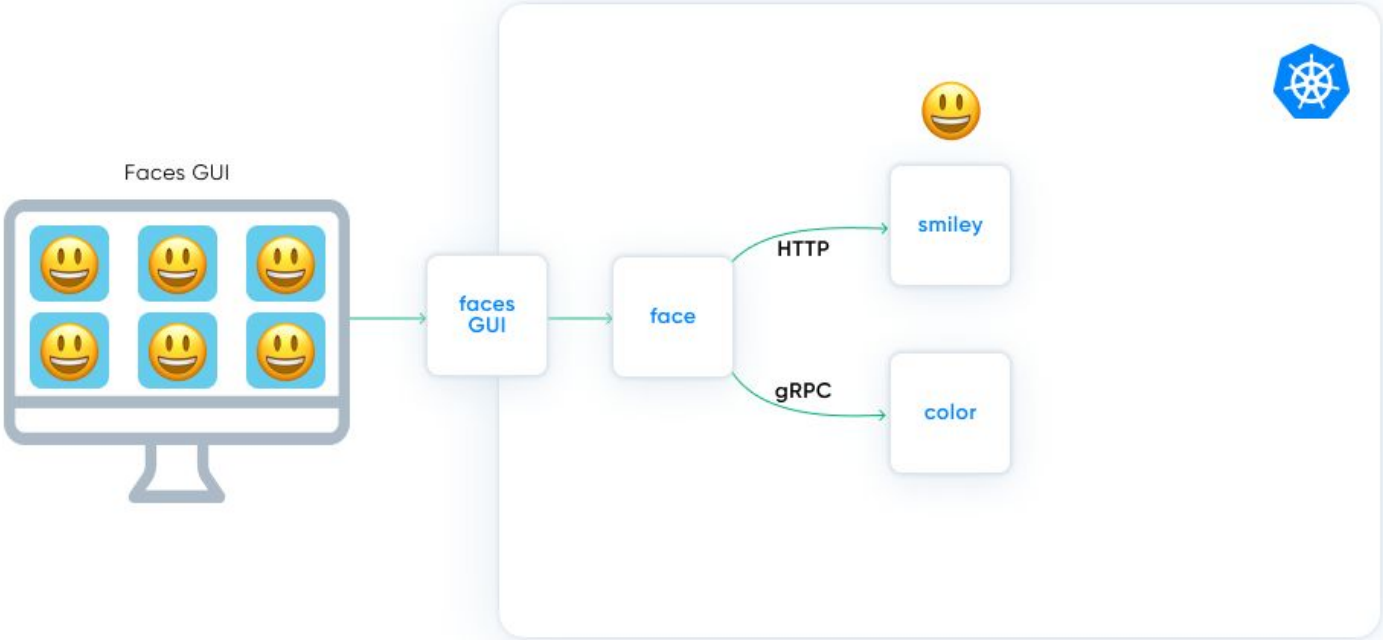
Faces (http://github.com/BuoyantIO/faces-demo)



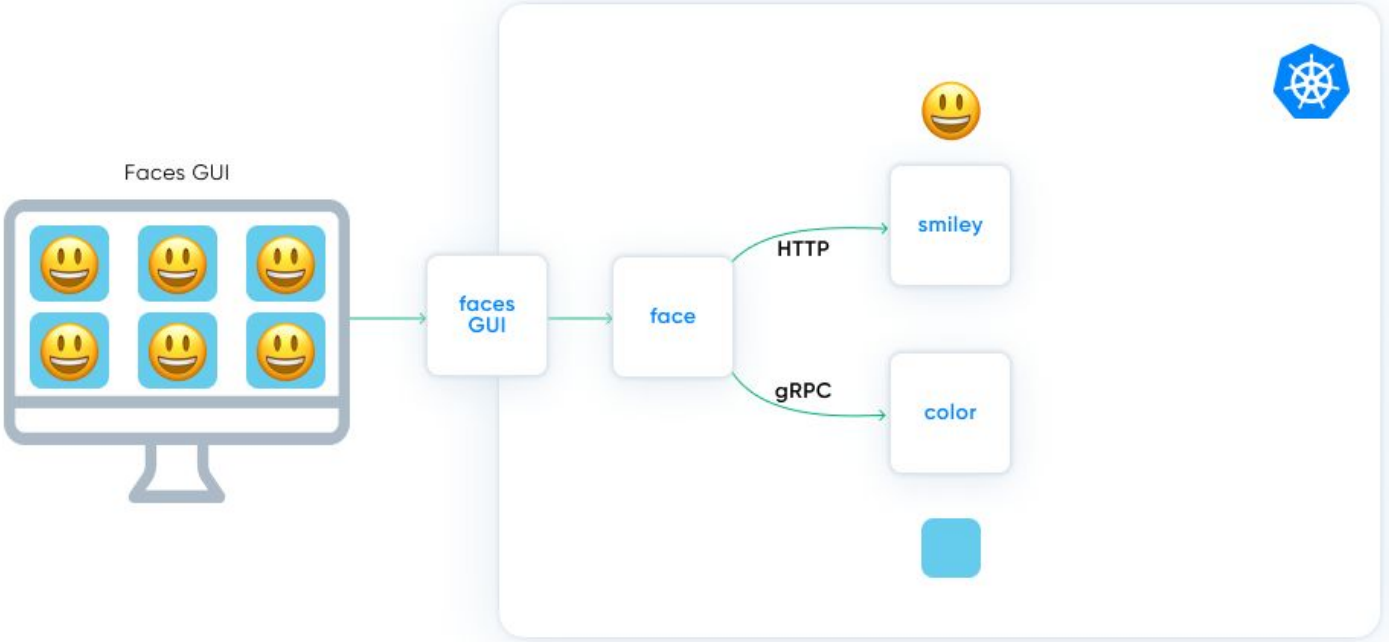
Faces (http://github.com/BuoyantIO/faces-demo)



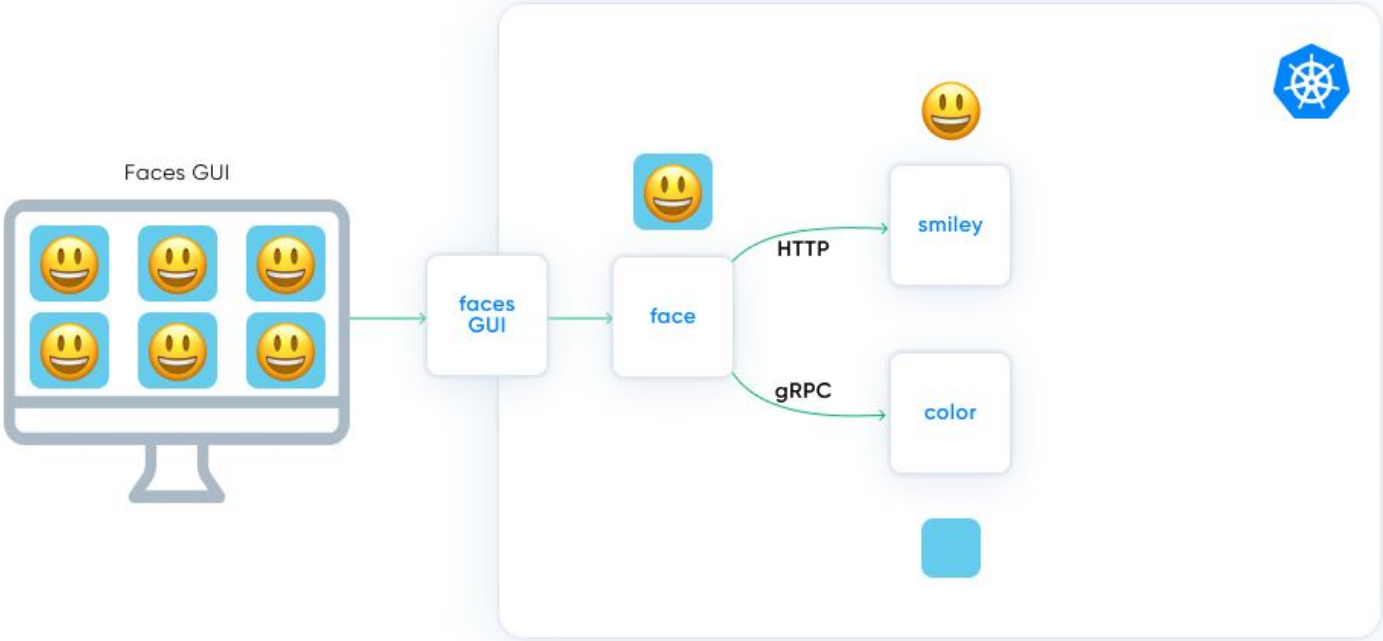
Faces (http://github.com/BuoyantIO/faces-demo)



Faces (http://github.com/BuoyantIO/faces-demo)

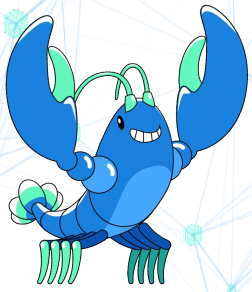


Faces (http://github.com/BuoyantIO/faces-demo)



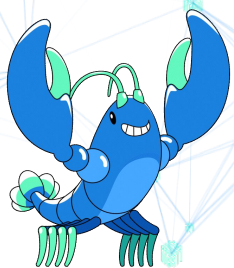
BUOYANT
SERVICE MESH
ACADEMY

DEMO 🤞



BUOYANT
SERVICE MESH
ACADEMY

Gotchas



Gotchas

- Routing happens **where the request is made**
  - Exactly one routing decision happens for the whole request
  - Example of *outbound policy* being made by the *outbound proxy*
- Authorization happens **where the request is received**
  - Example of *inbound policy* being made by the *inbound proxy*

Tell us how we can improve!

Your feedback matters!

(We promise it won't take more than a few minutes, and it will help us tremendously — thank you! 🙂)

BUOYANT
Creators of LINKERD

Buoyant Enterprise for LINKERD

Rust-based network security and reliability for modern applications. Built on open source and designed for the enterprise.

- ➔ Zero-trust security and compliance across your entire network
- ➔ Global traffic management and control
- ➔ Full L7 application observability
- ➔ Built for the enterprise

Learn more & try it for free at buoyant.io/**enterprise-linkerd**

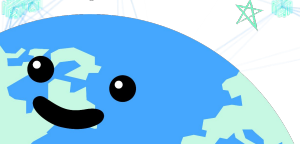
BUOYANT
Creators of LINKERD

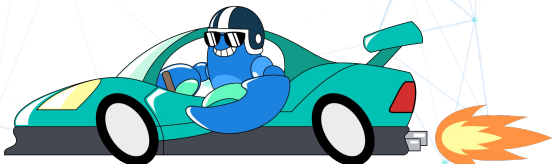
Updated Courses!

BUOYANT
Creators of LINKERD

Get Certified!

With hands-on LINKERD self-paced courses





SERVICE MESH ACADEMY

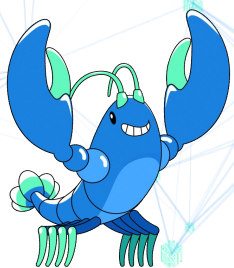
Up Next on August 14

How to Keep Your Cloud Spend at Bay with HAZL

SIGN UP TODAY!
buoyant.io/sma

BUOYANT
SERVICE MESH
ACADEMY

Q&A



BUOYANT
SERVICE MESH
ACADEMY

Thanks much!

flynn@buoyant.io
@flynn on slack.linkerd.io