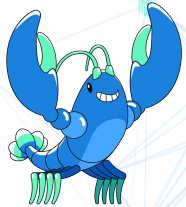
BUOYANT

Creators of LINKERD

Reducing Costs with HAZL

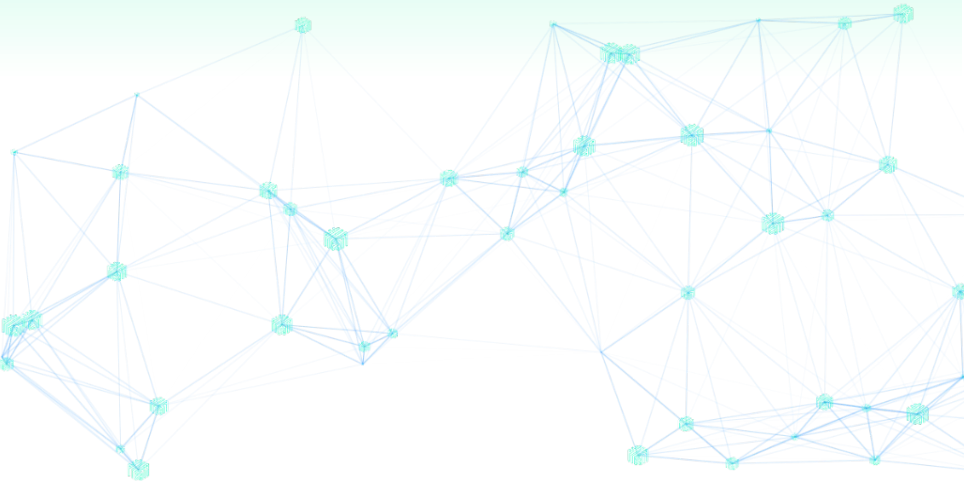
Flynn, Technical Evangelist for Linkerd



@BuoyantIO

buoyant.io

What's on the agenda?



- **What is Linkerd?**
- **What is HAZL?**
  - and why should you care 🙂
  - including an intro to zones, regions, etc.
- **DEMO!**
- **Gotchas**

BUOYANT

How do you follow along?

- https://github.com/BuoyantIO/service-mesh-academy/tree/main/reduce-costs-with-hazl
- **For this demo, you will need Buoyant Enterprise for Linkerd 2.18+!**
  - I'll be using 2.18.3
- I'll be using a multizone `k3d` clusters, but the real requirement here is multiple zones
  - I'll show how to set this up

BUOYANT

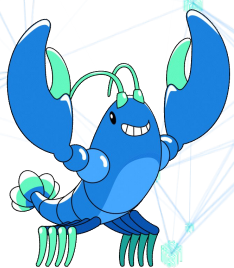
How do you follow along?

- **k3d**
  https://k3d.io/
- **kubectl**
  https://kubernetes.io/docs/tasks/tools/
- **linkerd CLI**
  https://linkerd.io/2/getting-started
- **bat**
  https://github.com/sharkdp/bat

BUOYANT

BUOYANT
SERVICE MESH
ACADEMY

What is Linkerd?



What is Linkerd?

Linkerd is a **service mesh**.

***service mesh**, n:*

- An infrastructure layer providing security, reliability, and observability at the platform level, uniformly, across an entire application.

What is Linkerd?

Linkerd is a **service mesh**.

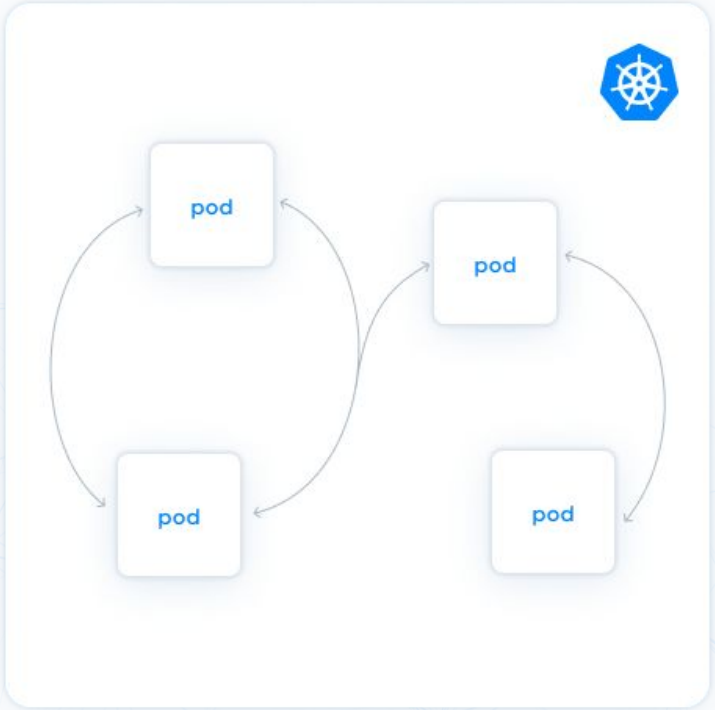
***service mesh**, n:*

- An infrastructure layer providing security, reliability, and observability at the platform level, uniformly, across an entire application.

The Microservices Architecture

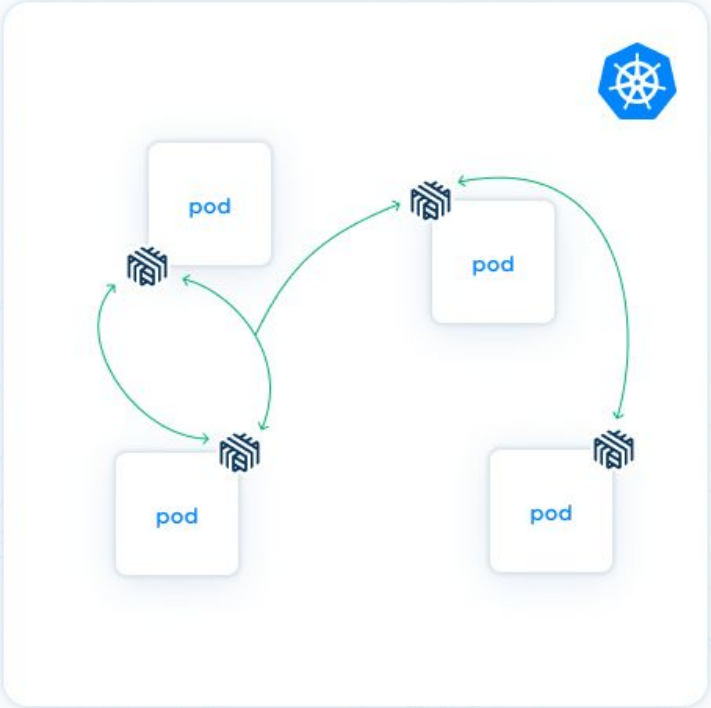
- Microservices communicate over an insecure, unreliable network.
- These are *fundamental characteristics* of the way real networking is built; they cannot be changed.

- Service meshes like Linkerd exist to make this situation better.



Microservices and the Mesh

- Like most other meshes, Linkerd works by adding a proxy (a *sidecar)* next to each application pod.
- *Unlike* any other mesh, Linkerd uses a purpose-built, lightweight, ultrafast Rust microproxy.
- These microproxies mediate and measure all communications in the mesh, which allows for all the mesh's functionality.



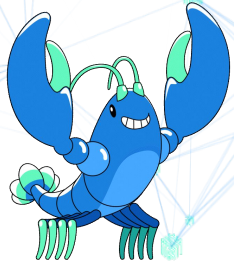
Why is this important?

Security, reliability, and observability are not optional.

- You can get them from a mesh.
- You can get them by writing a lot of application code.
- You can't do without them.

BUOYANT
SERVICE MESH
ACADEMY

What is HAZL?



HAZL: High Availability Zonal Loadbalancer

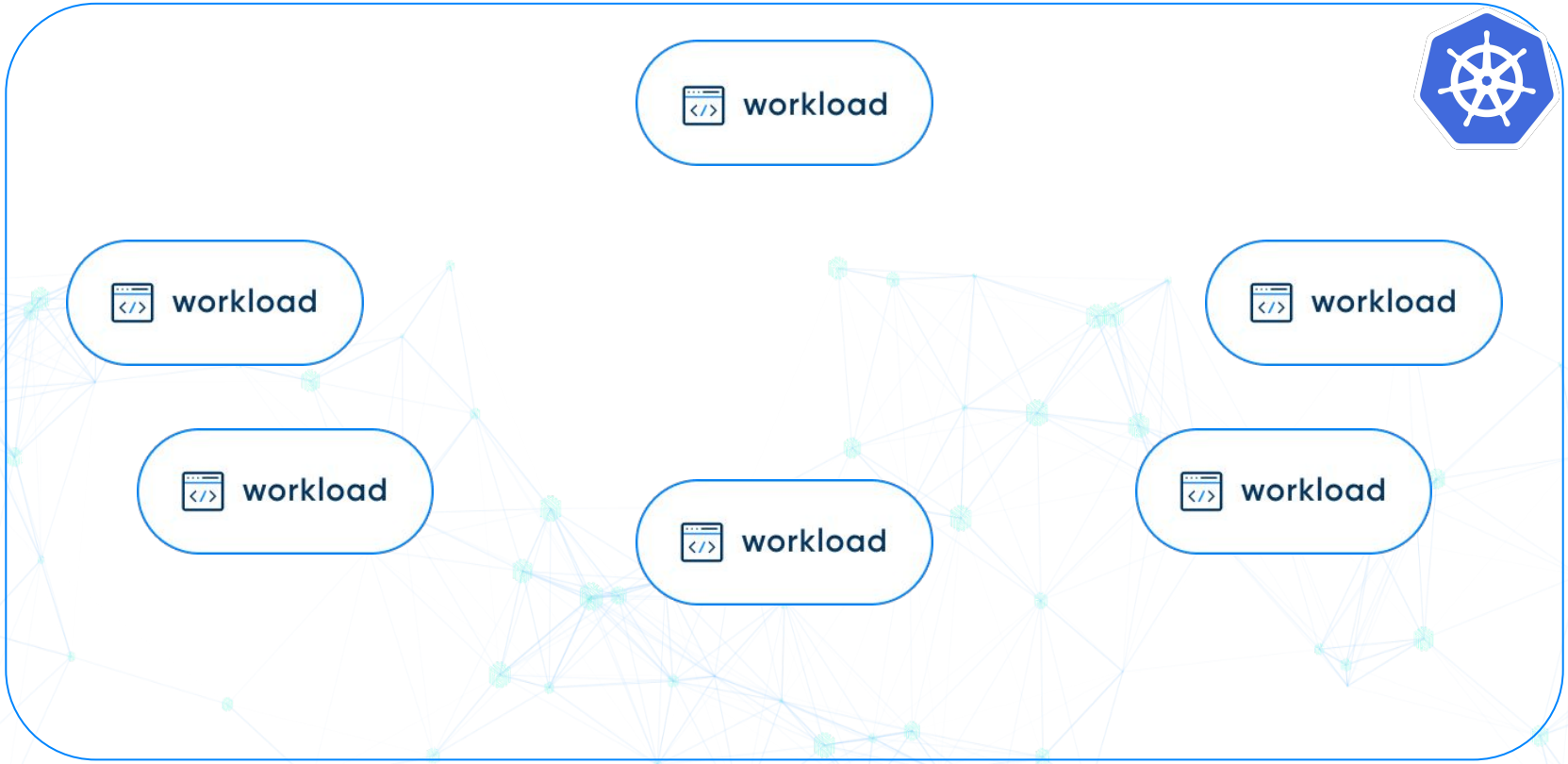
Ultimately, HAZL is a way to keep costs and latency down while preserving reliability.

HAZL: High Availability Zonal Loadbalancer

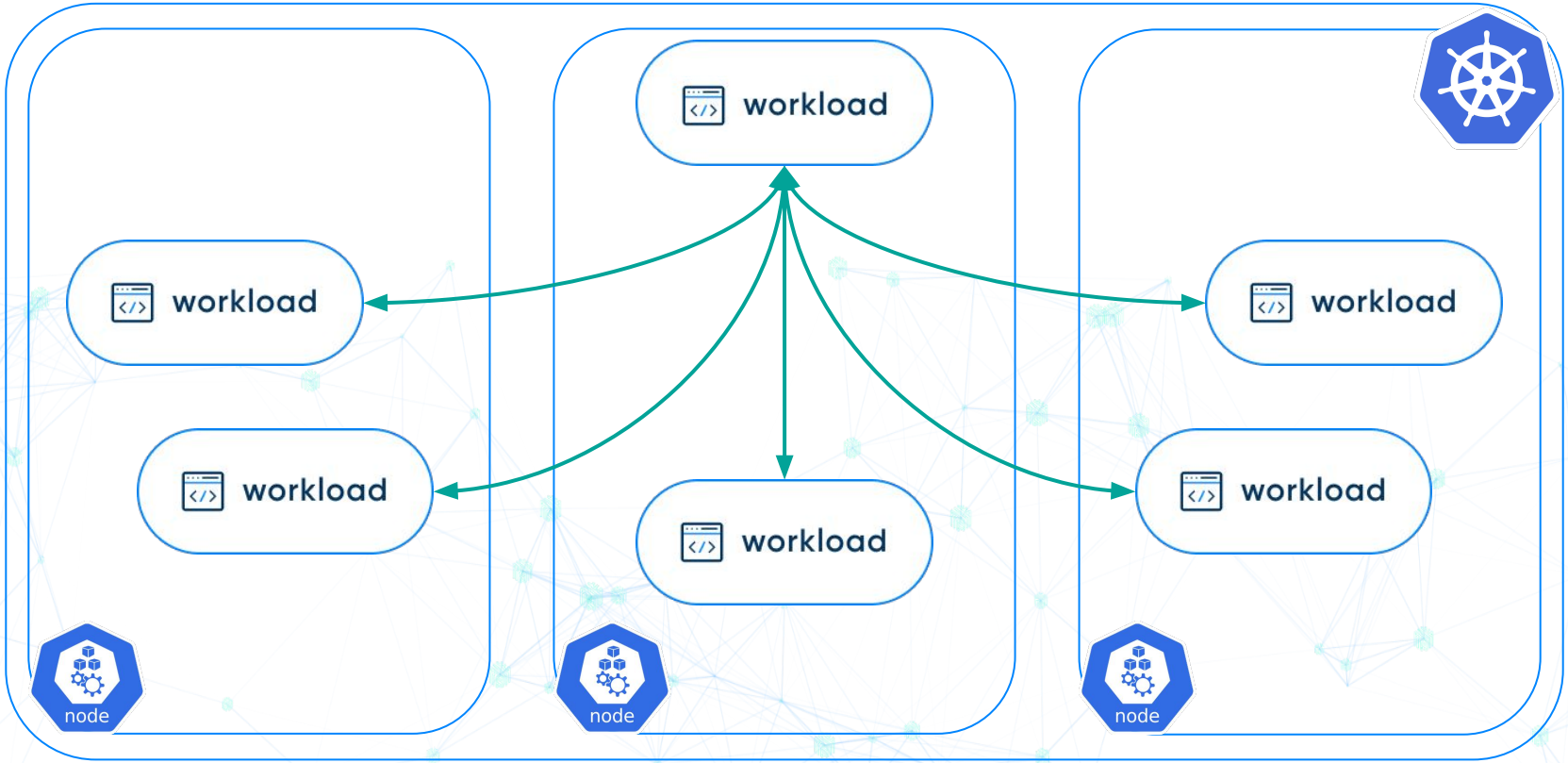
Ultimately, HAZL is a way to keep costs and latency down while preserving reliability.

OK, great, what's this "zone" business?

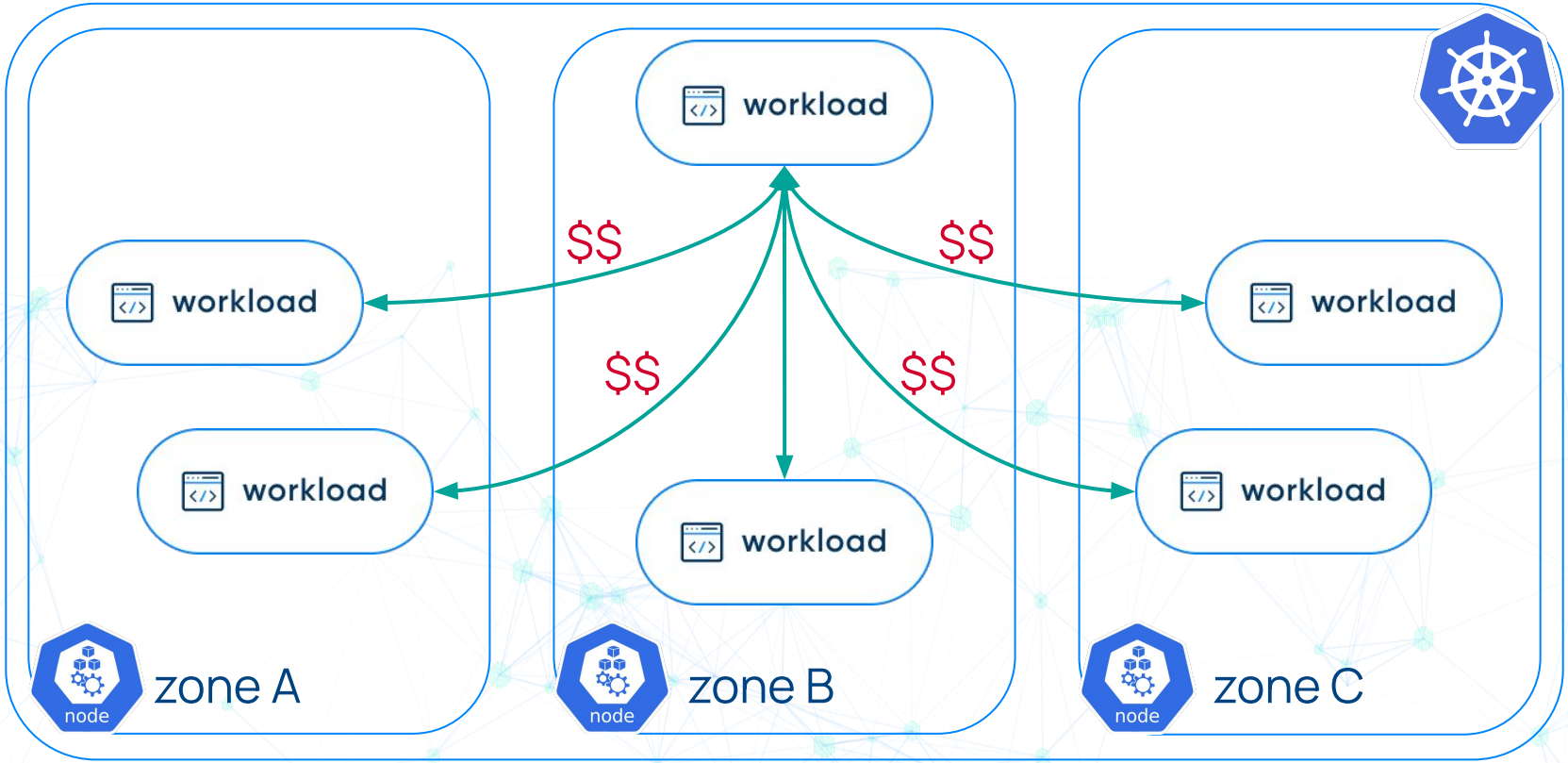
Single-Zone Cluster



Single-Zone Cluster

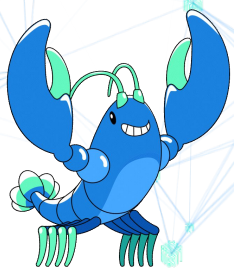


Multi-Zone Cluster

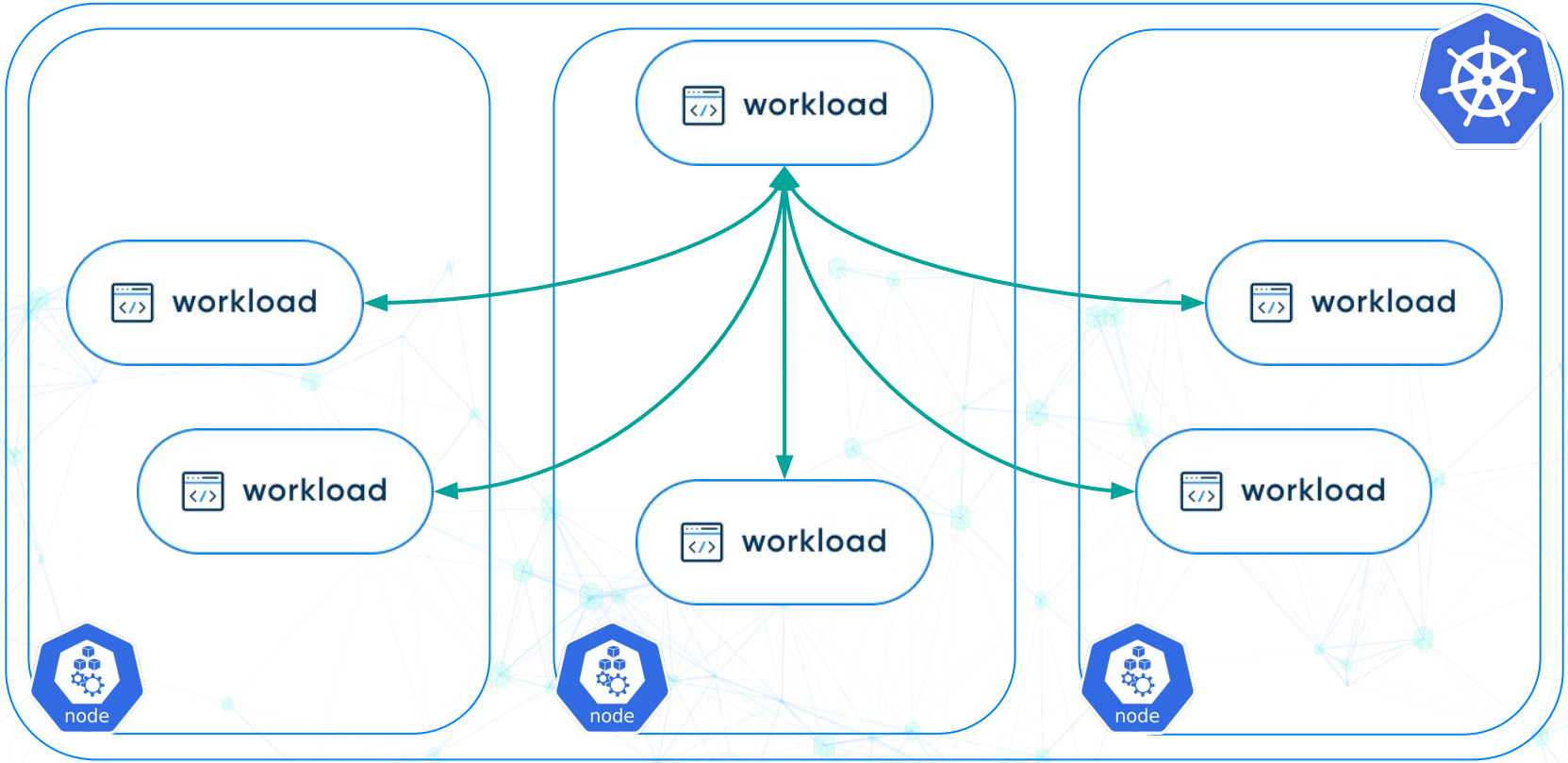


BUOYANT
SERVICE MESH
ACADEMY

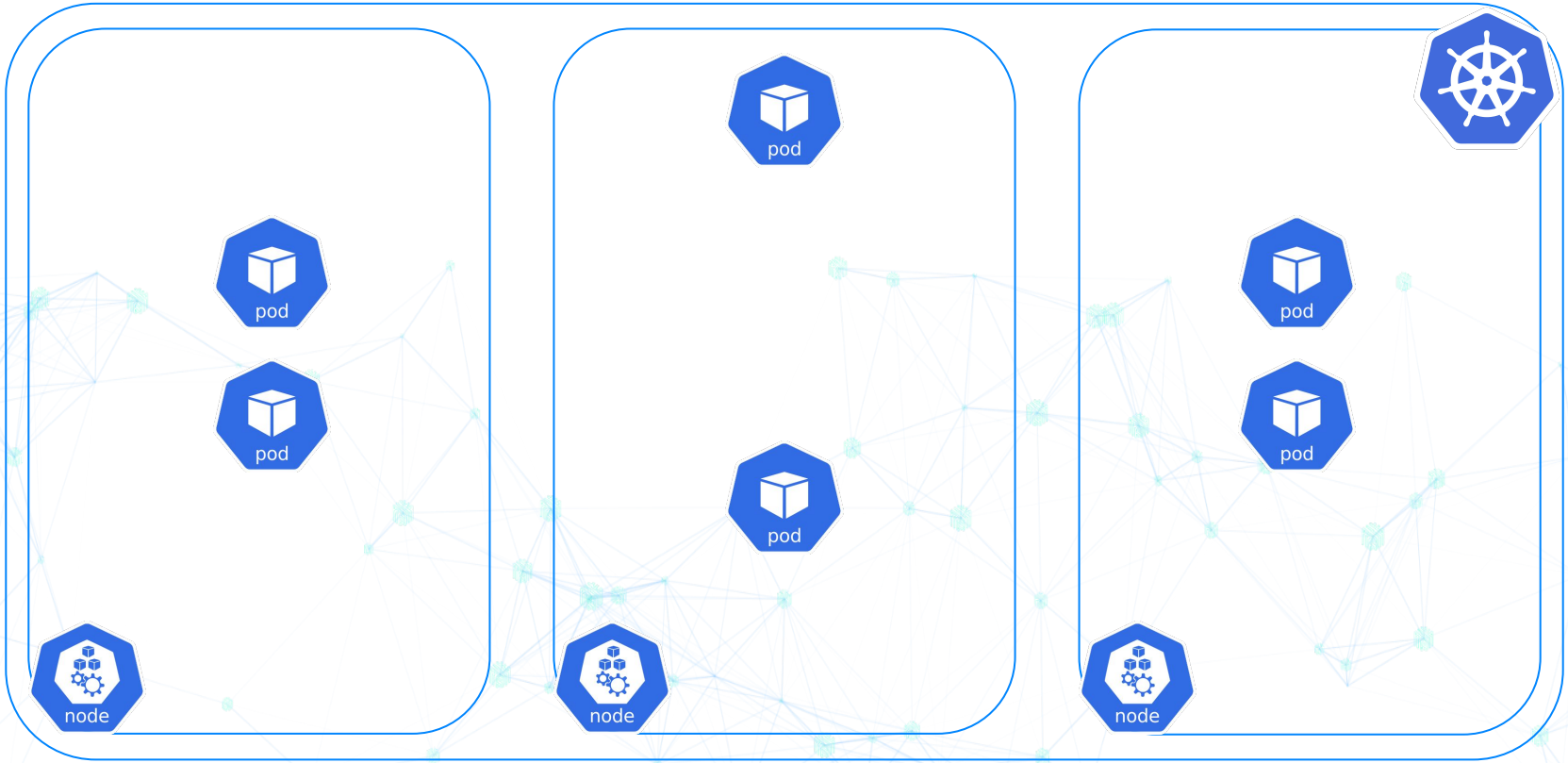
Why is cross-zone traffic expensive?



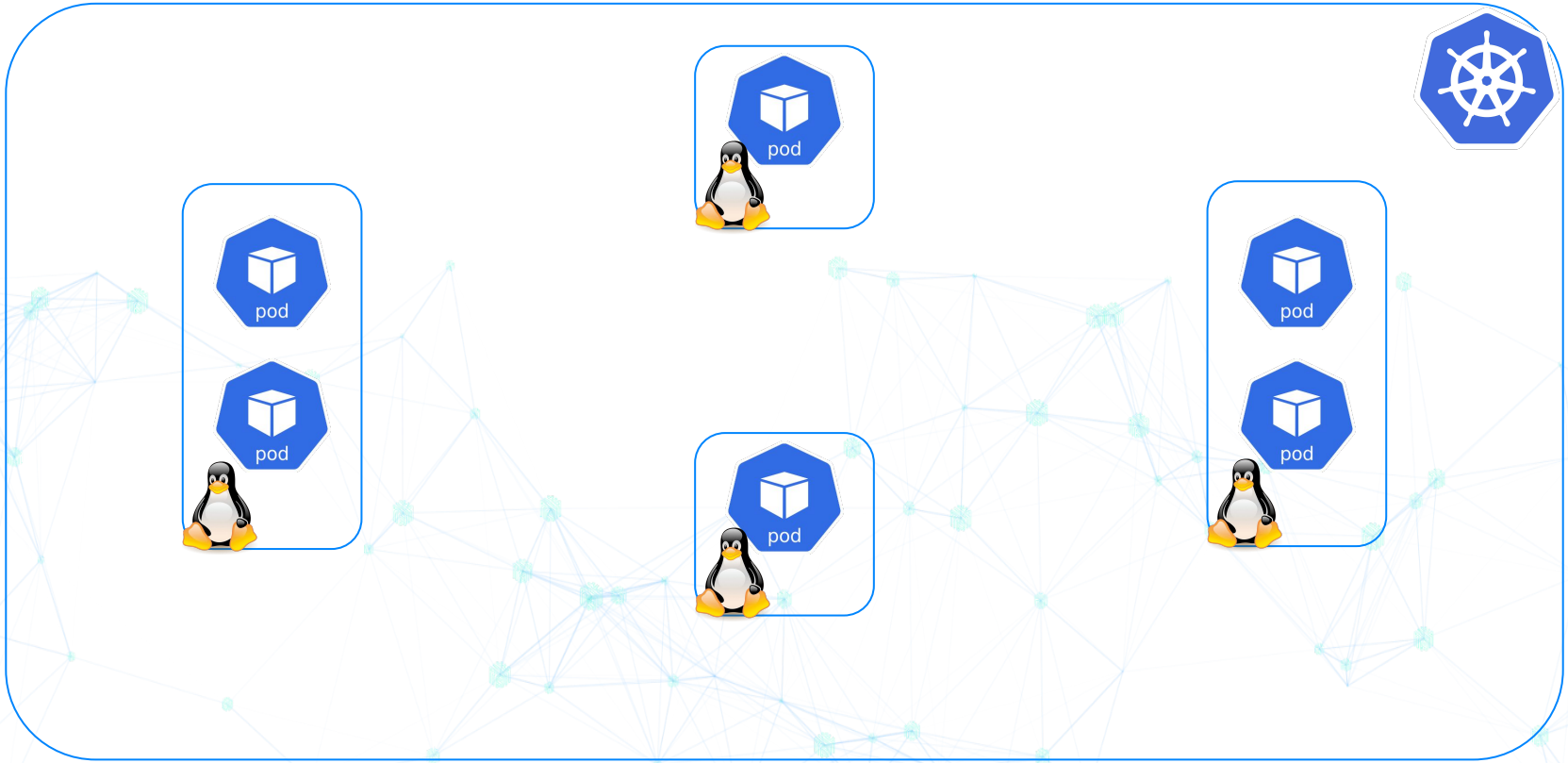
Multi-Zone Cluster



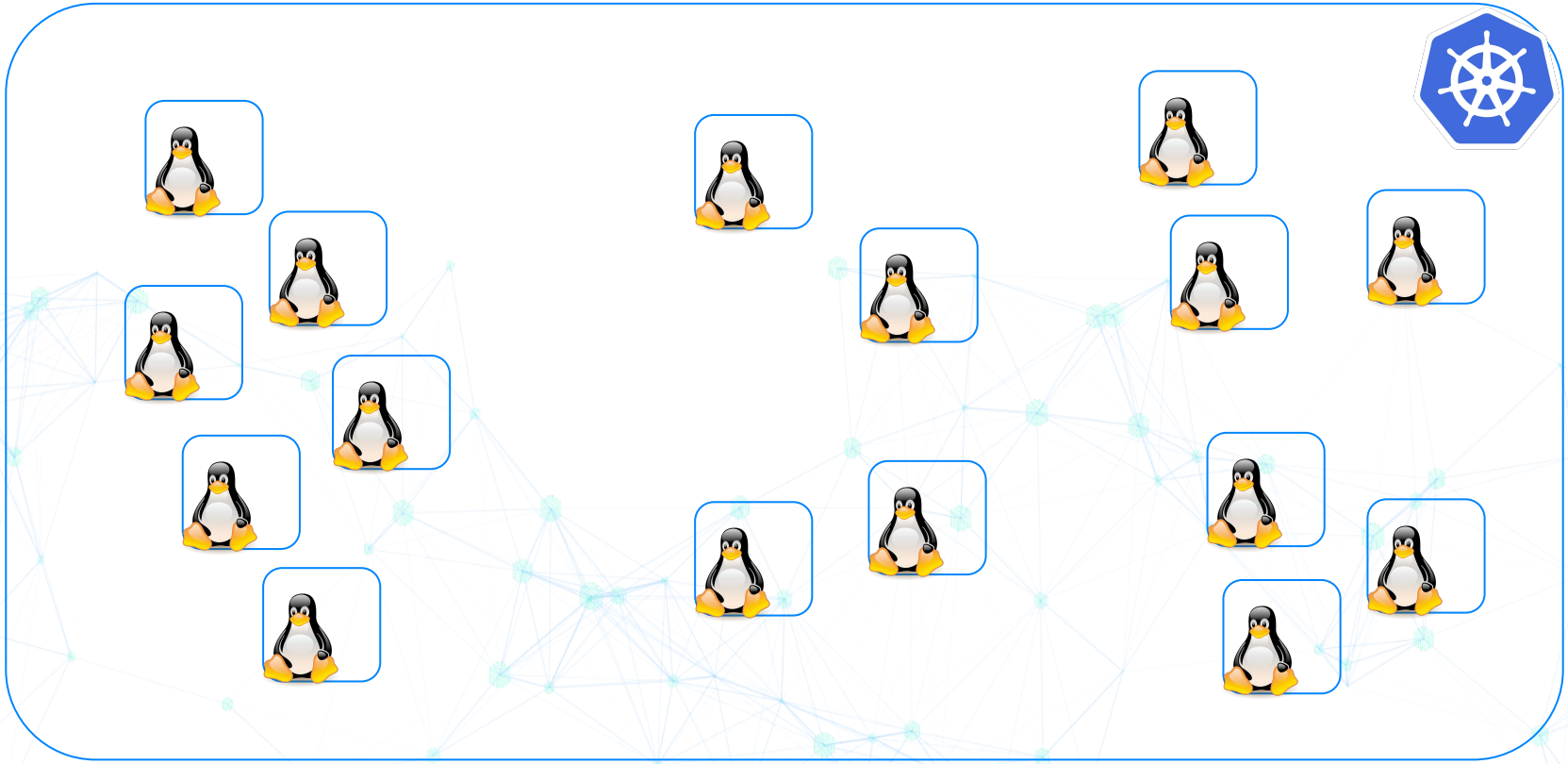
Multi-Zone Cluster



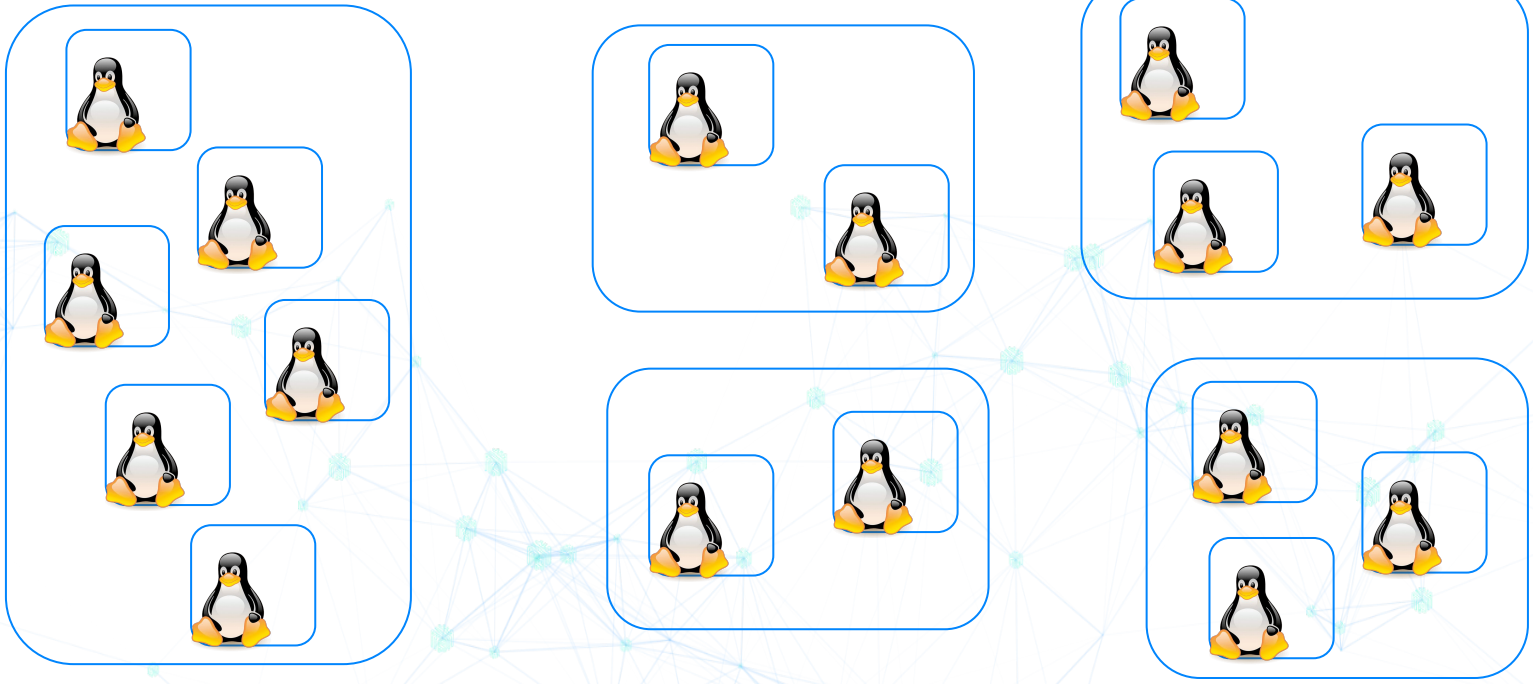
Node Hardware



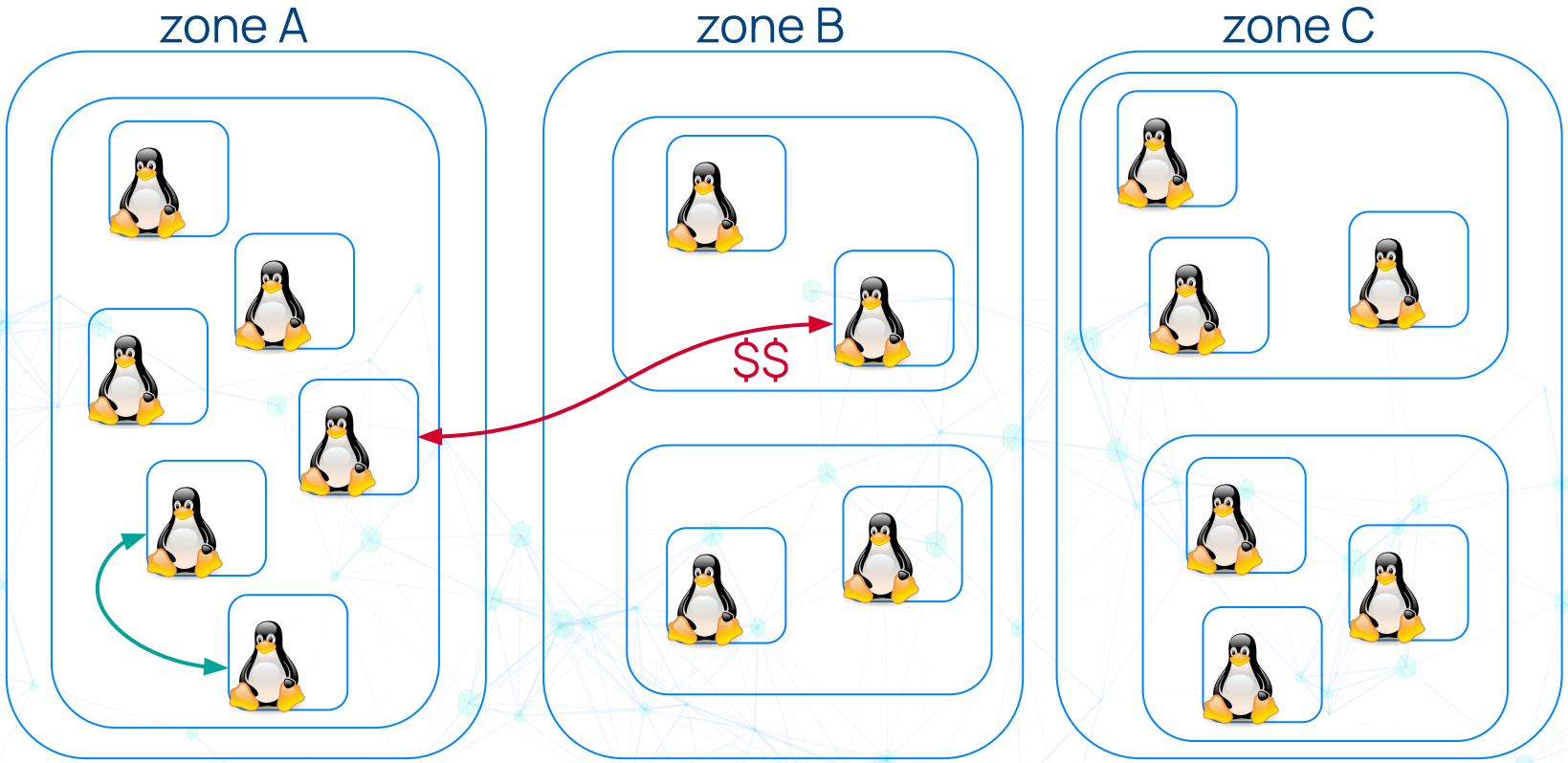
Node Hardware



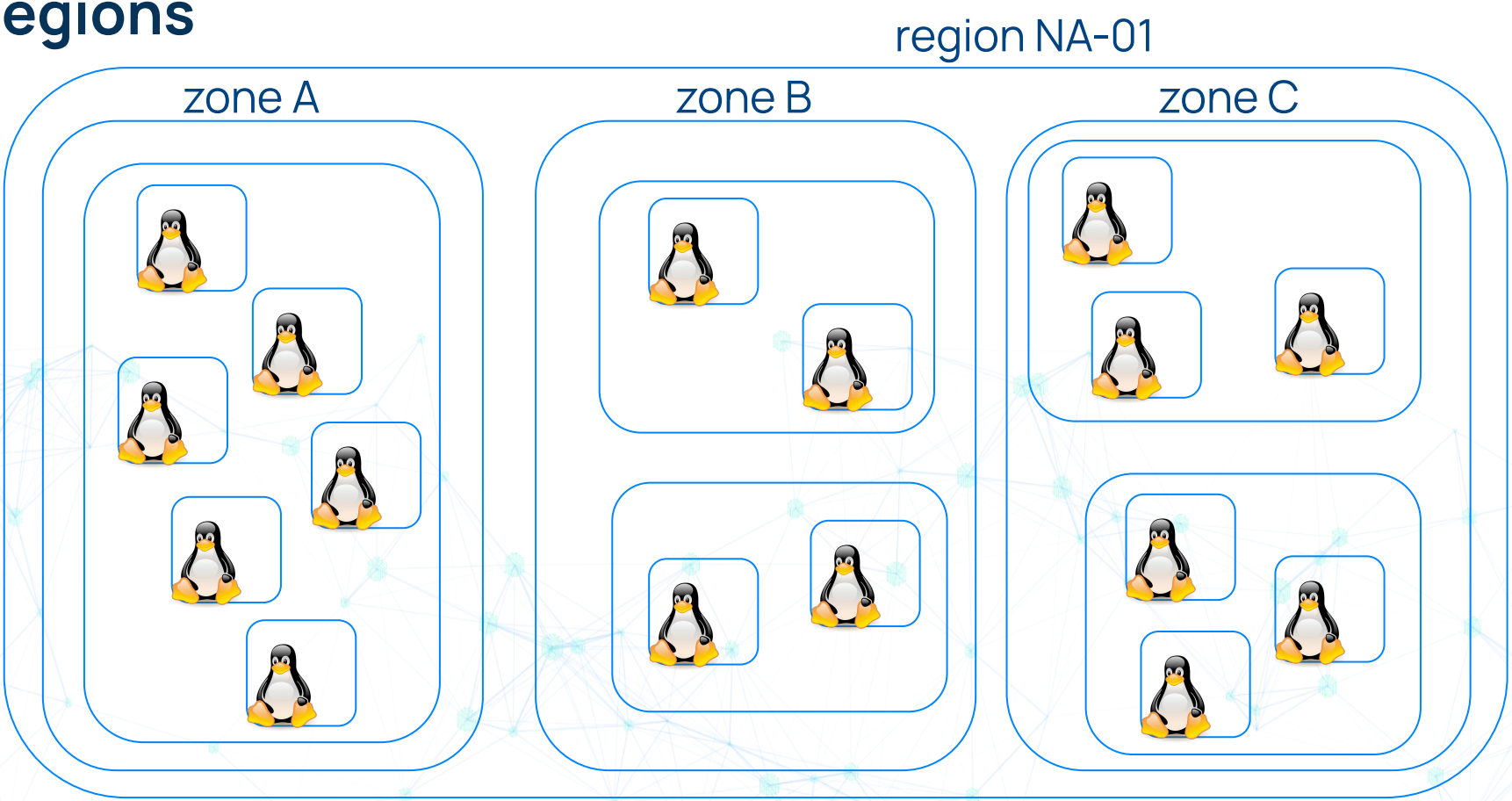
Data Centers



Zones



Regions



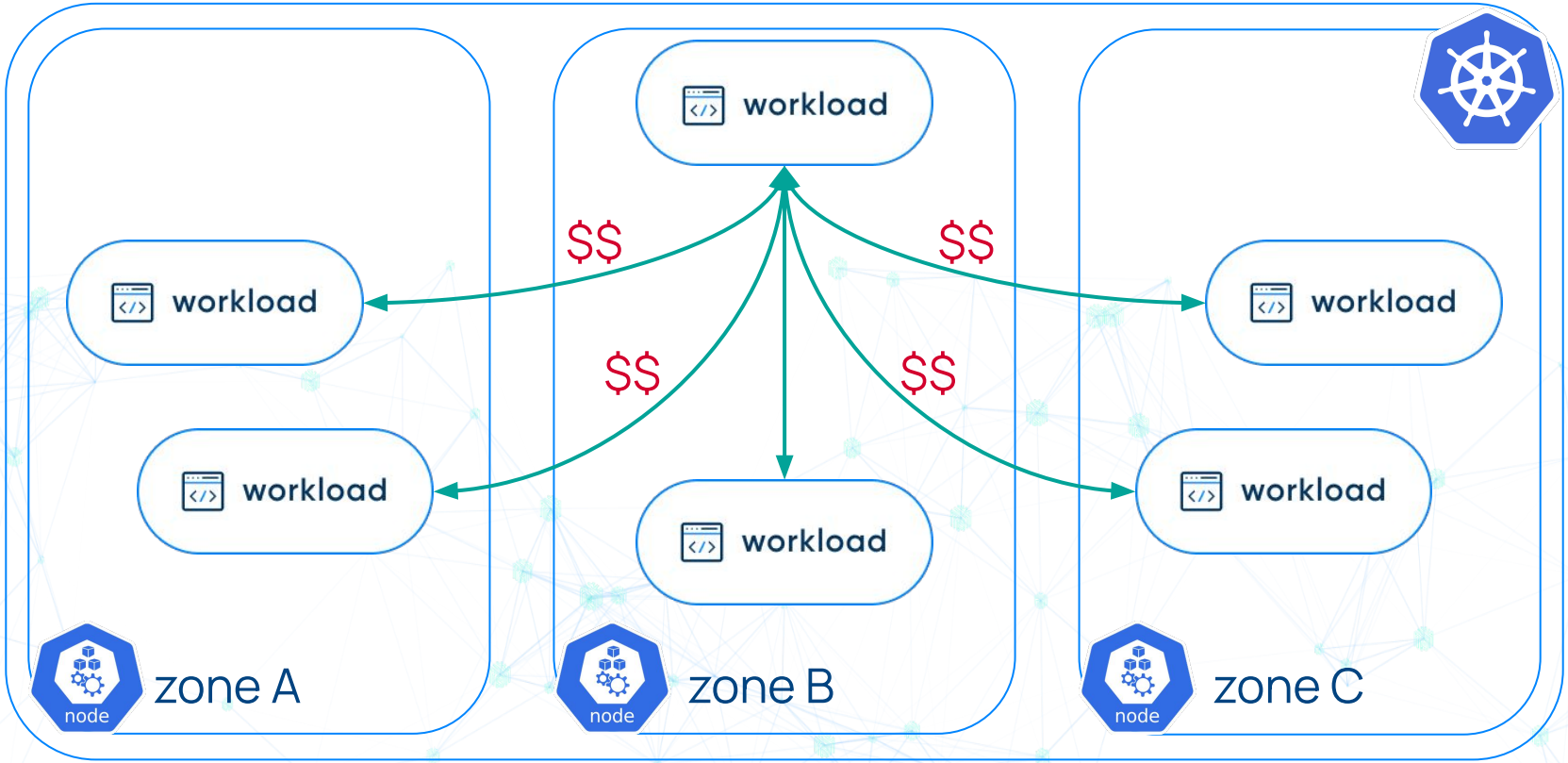
Cheat Sheet

- **Node failure**
  - One of the Linux boxes has crashed
- **A whole lot of Nodes fail**
  - Maybe a server rack lost power… maybe a whole data center caught on fire
- **A Zone goes off the air**
  - Hollywood version: Canada is invading Boston
  - Realistic version: Somebody screwed up BGP
- **A Region goes down**
  - Hollywood version: Major meteor strike
  - Realistic version: Somebody screwed up BGP 😂

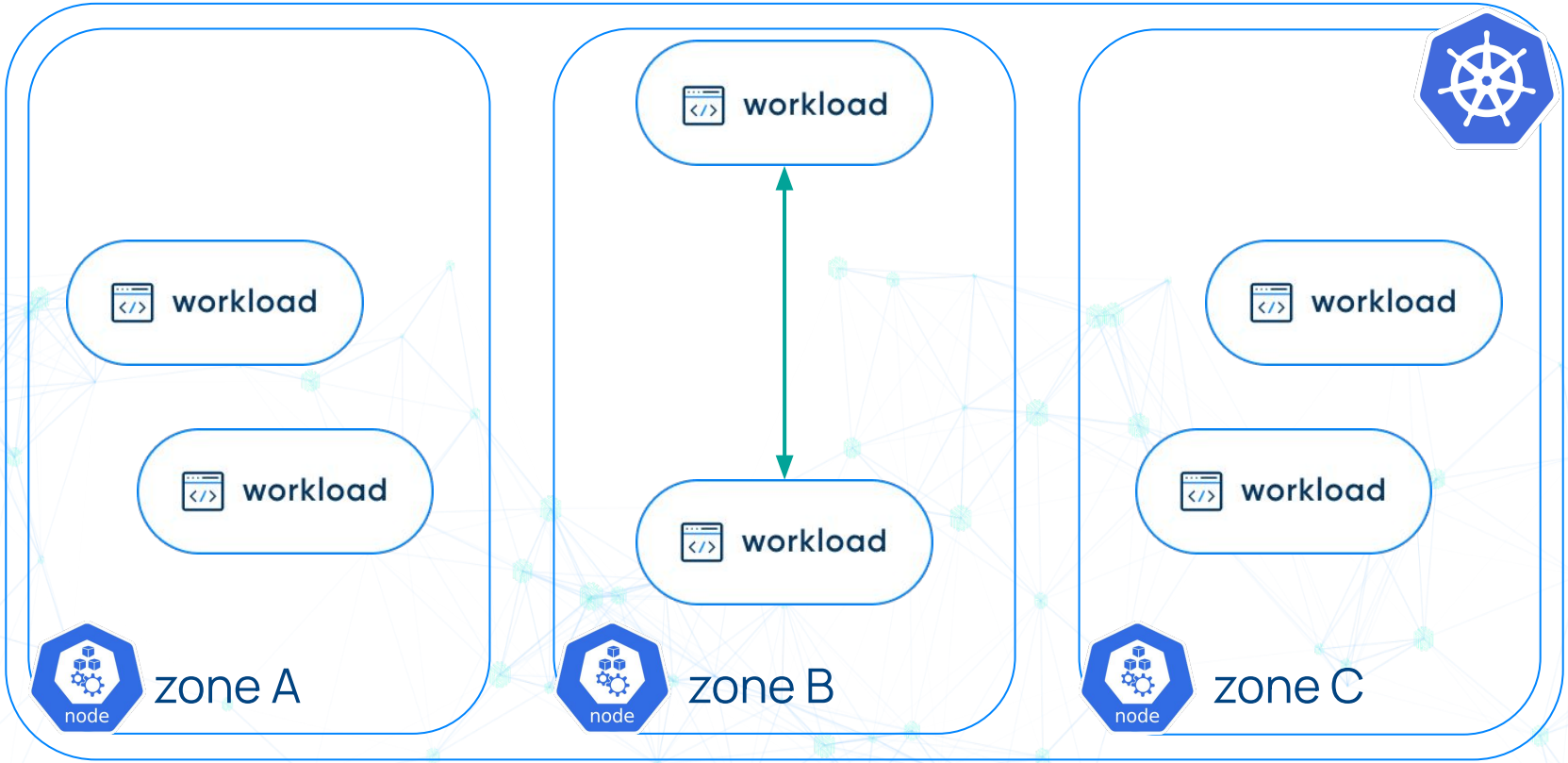
Cheat Sheet

- **Multizone clusters exist to help with a Zone failing**
  - This is **not common**
  - On the other hand, it's also hard to fix and slow to recover from
- **Multizone clusters get expensive**
  - Traffic across zones costs money because bandwidth between zones is more limited

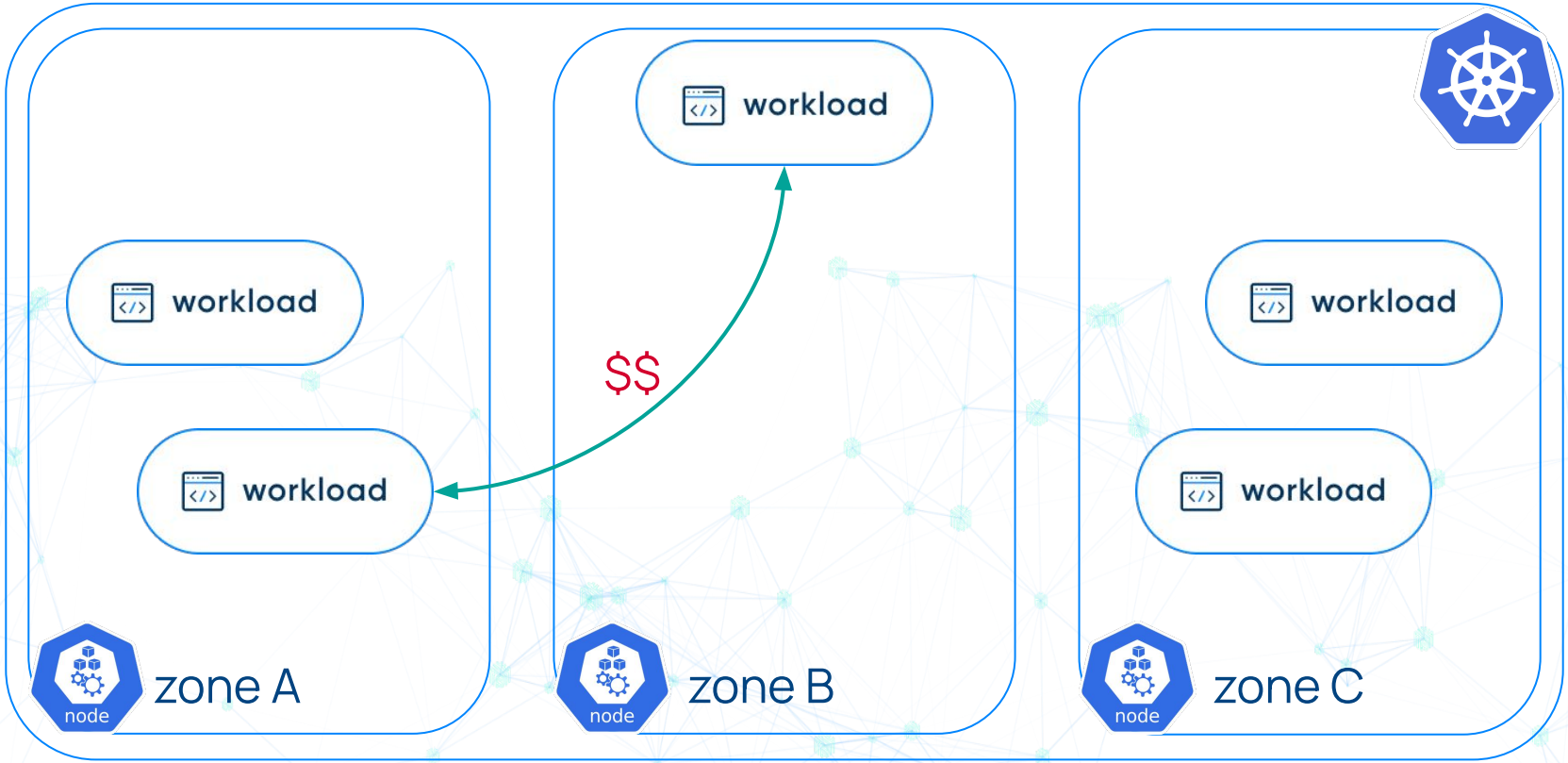
Multi-Zone Cluster



Multi-Zone Cluster



Multi-Zone Cluster



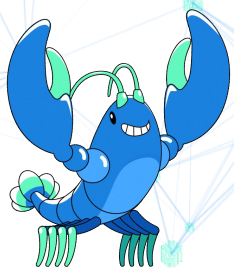
HAZL: High Availability Zonal Loadbalancer

HAZL does what that last slide shows.

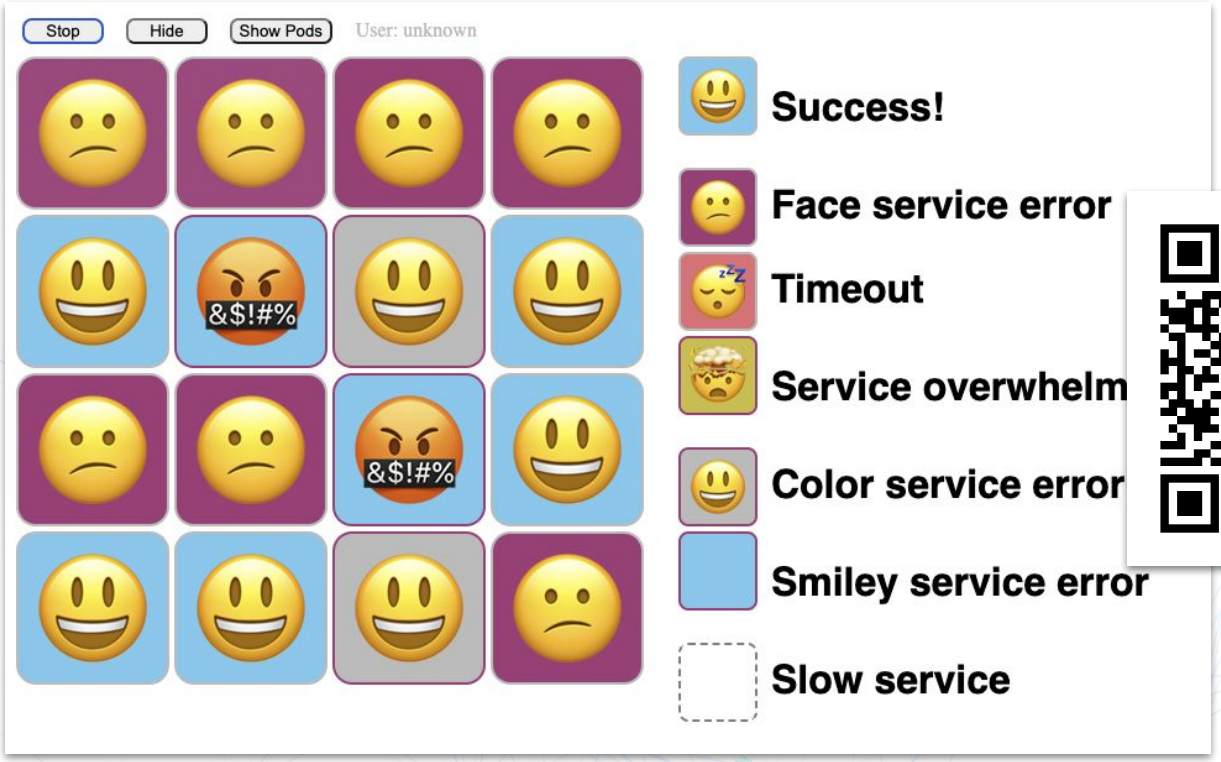
- Keep traffic in-zone if possible
- Allow traffic to go out-of-zone if needed

BUOYANT
SERVICE MESH
ACADEMY

Demo Architecture

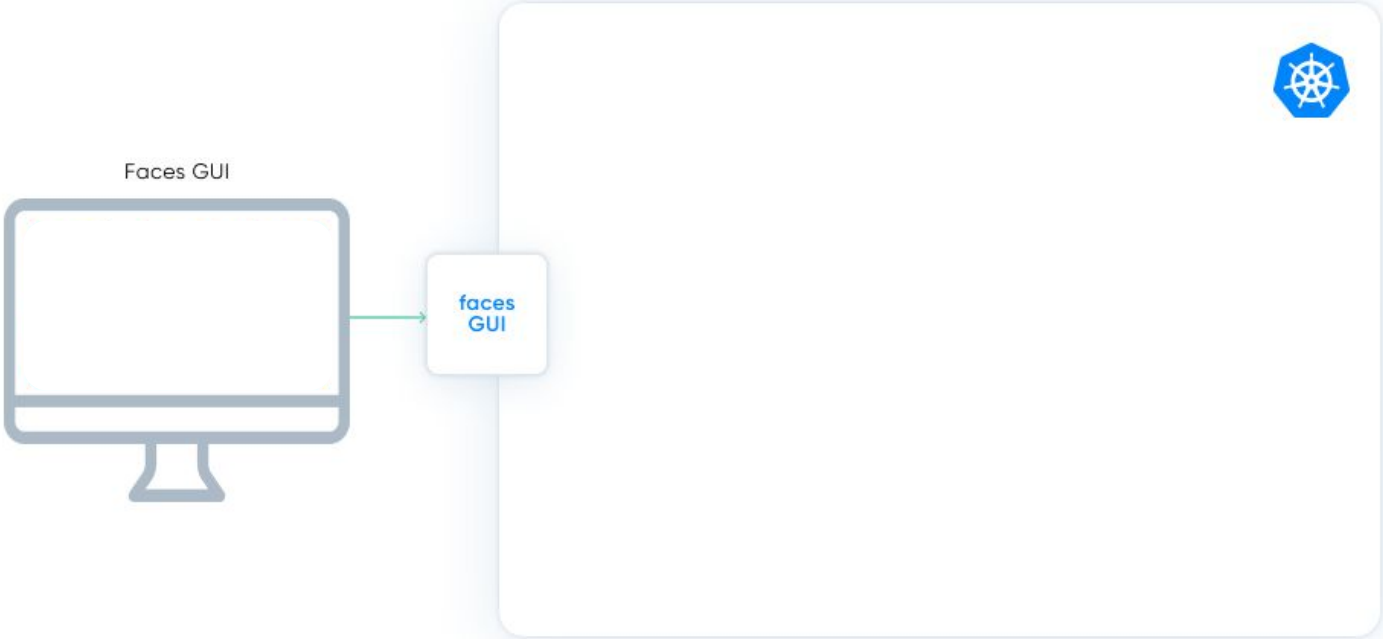


Faces (http://github.com/BuoyantIO/faces-demo)

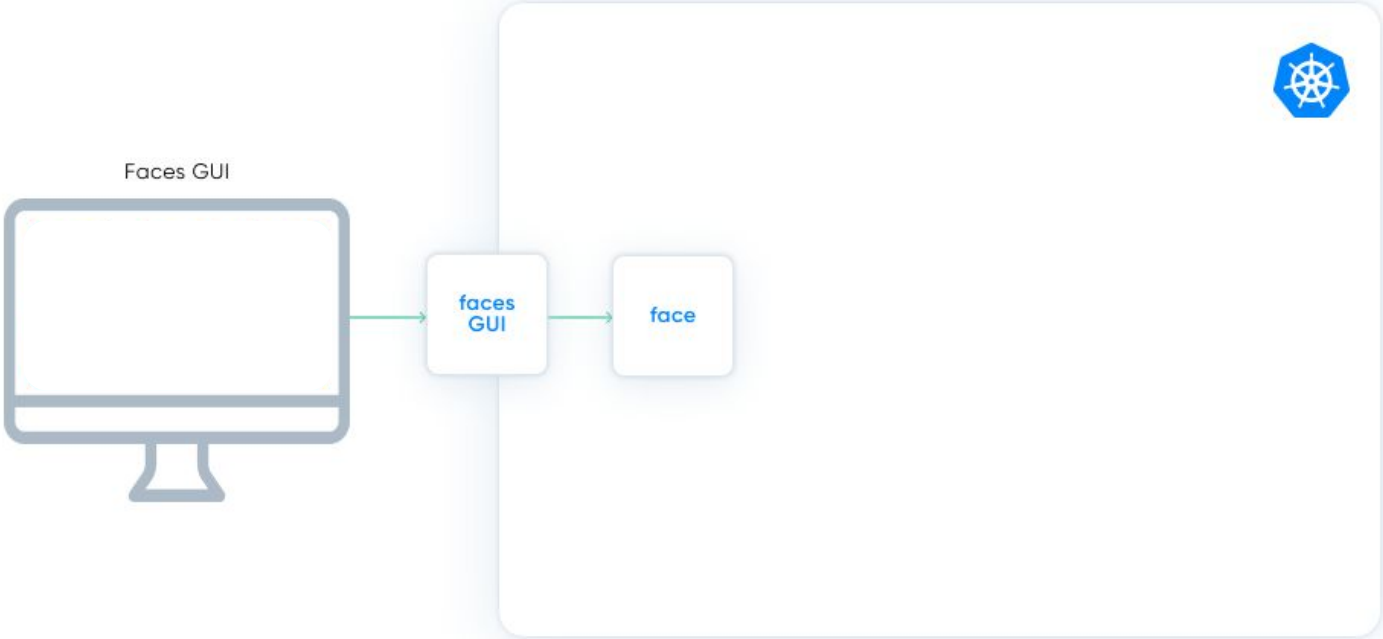


Faces (http://github.com/BuoyantIO/faces-demo)

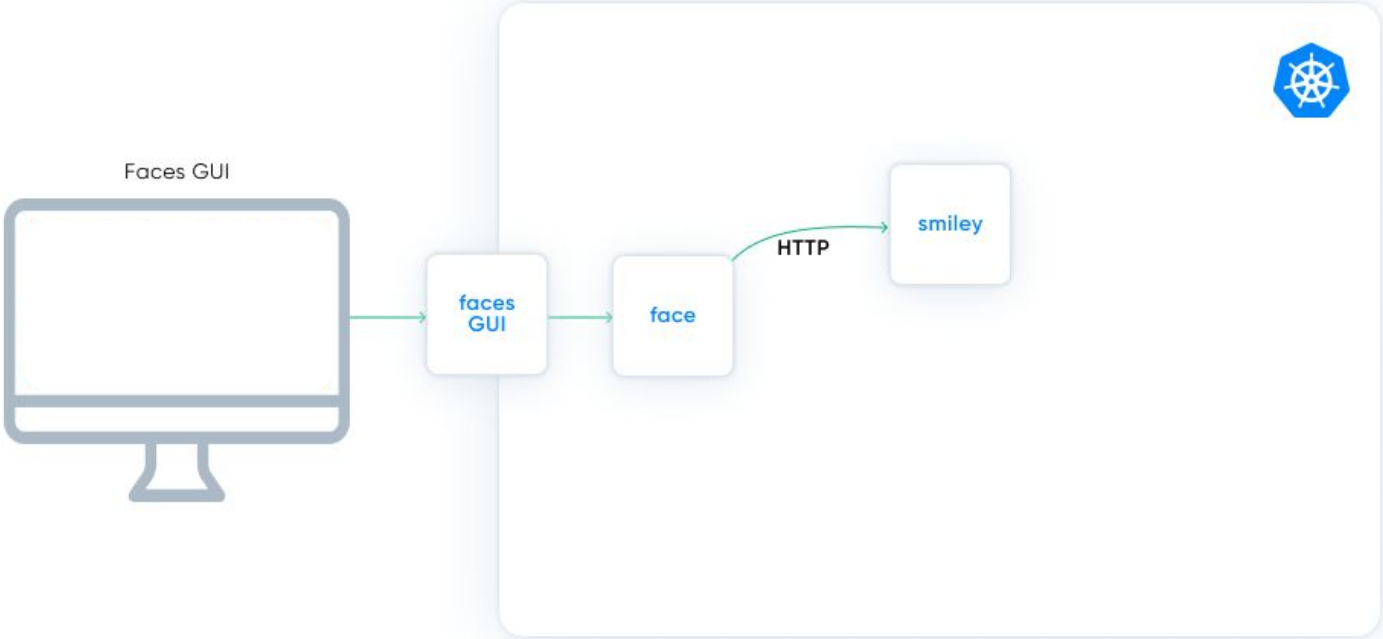
Faces (http://github.com/BuoyantIO/faces-demo)



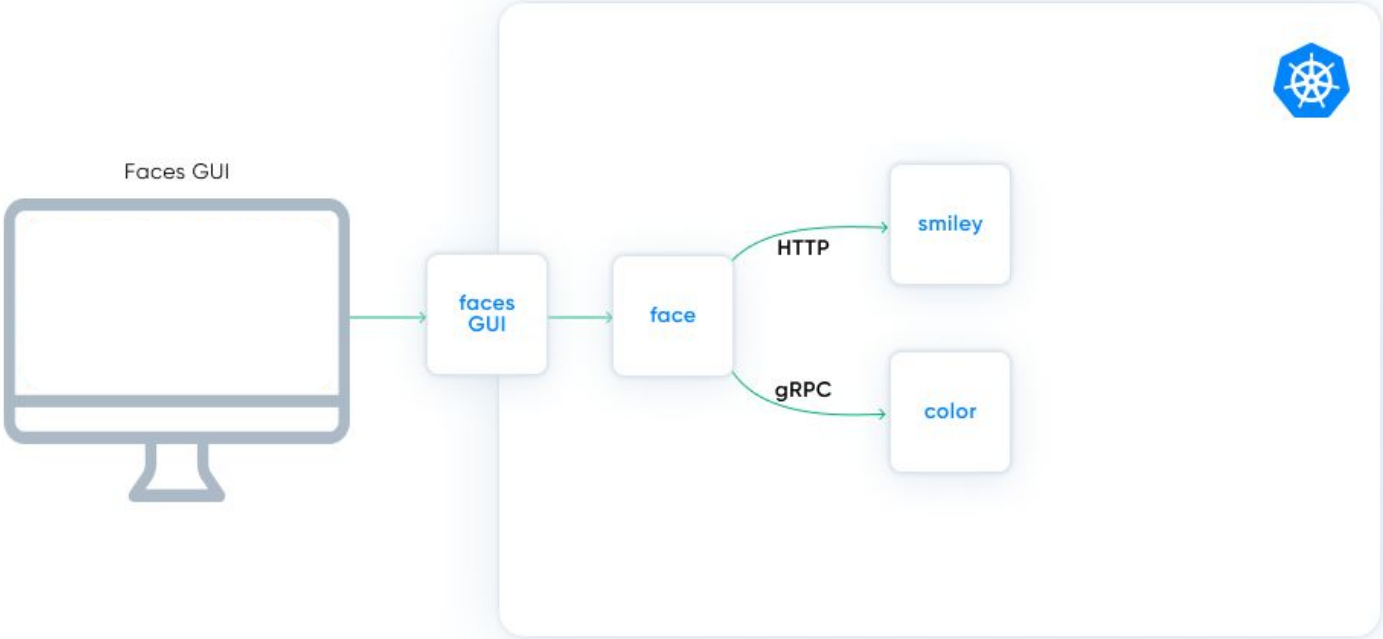
Faces (http://github.com/BuoyantIO/faces-demo)



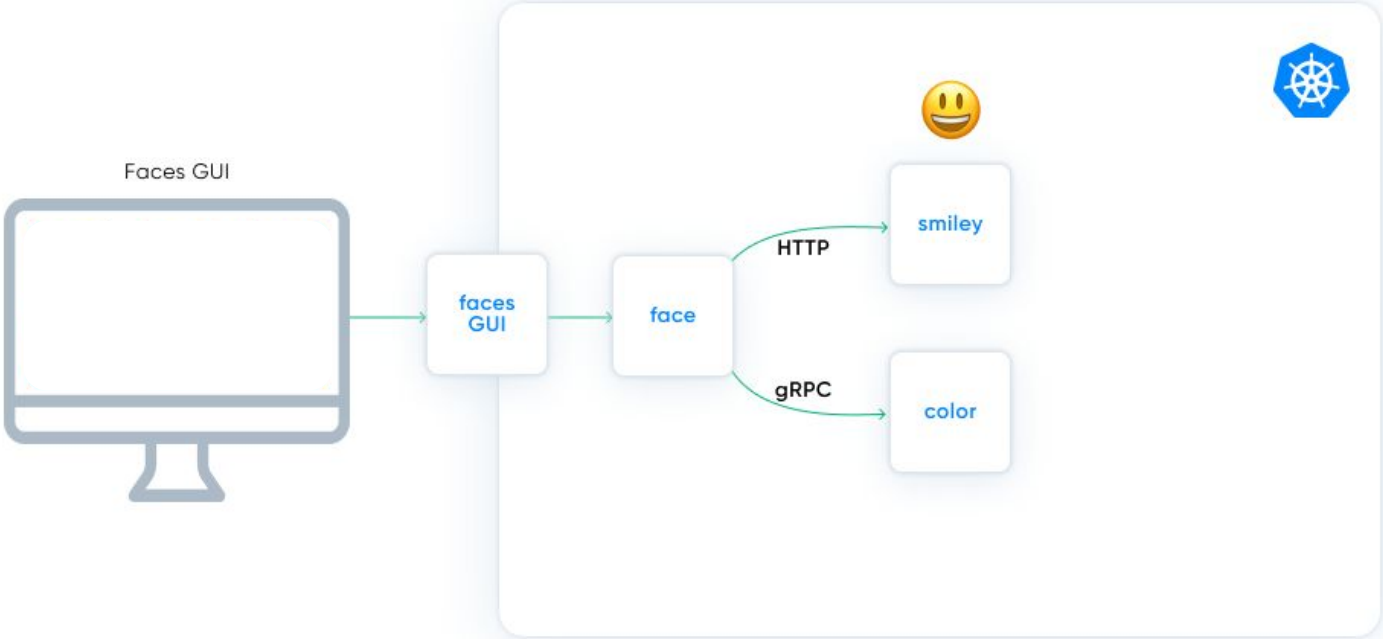
Faces (http://github.com/BuoyantIO/faces-demo)



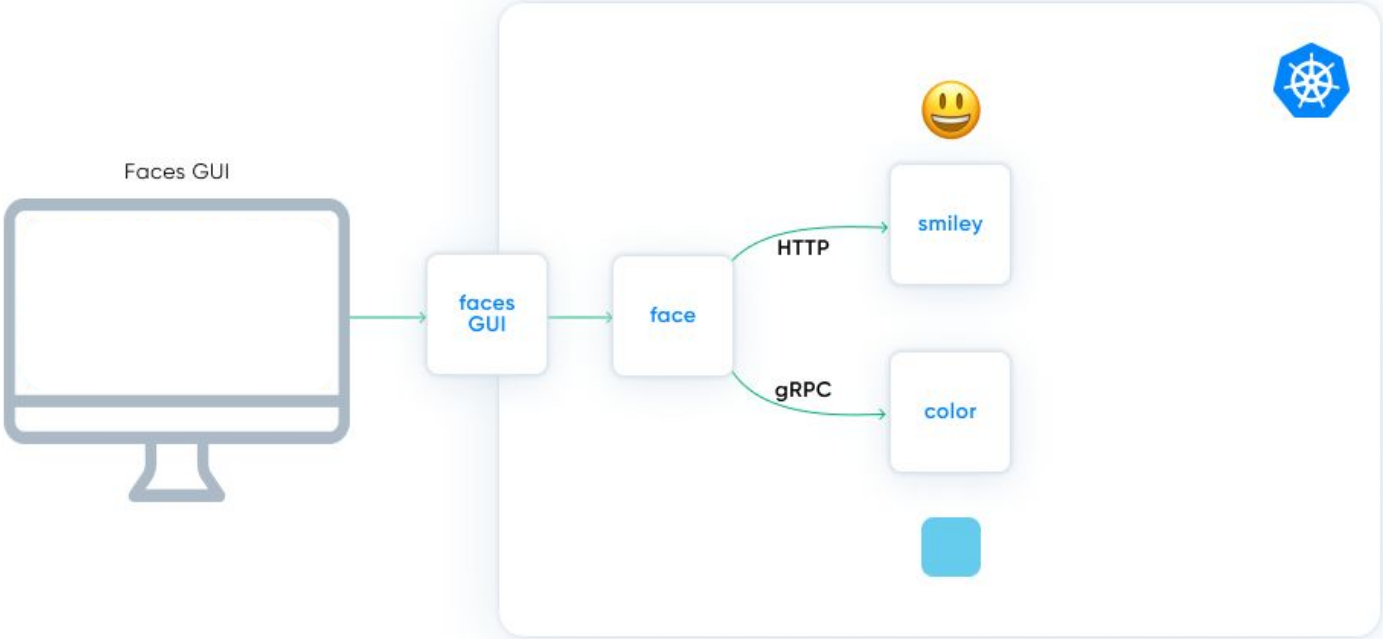
Faces (http://github.com/BuoyantIO/faces-demo)



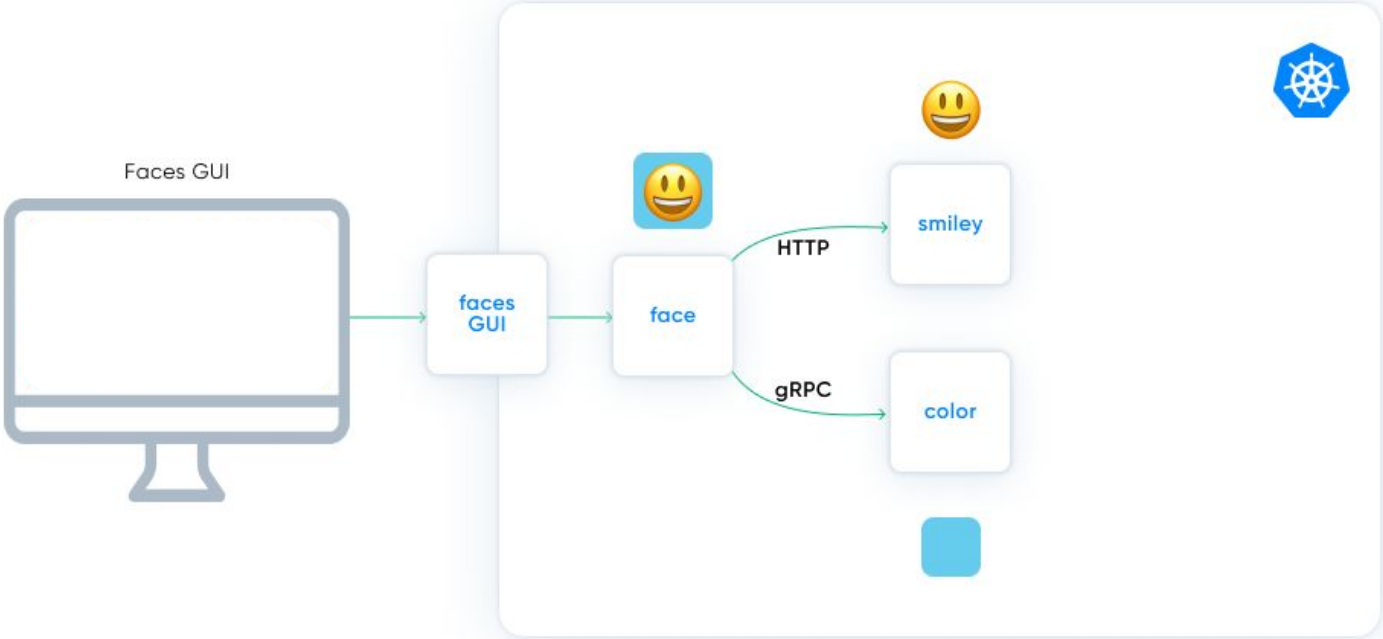
Faces (http://github.com/BuoyantIO/faces-demo)



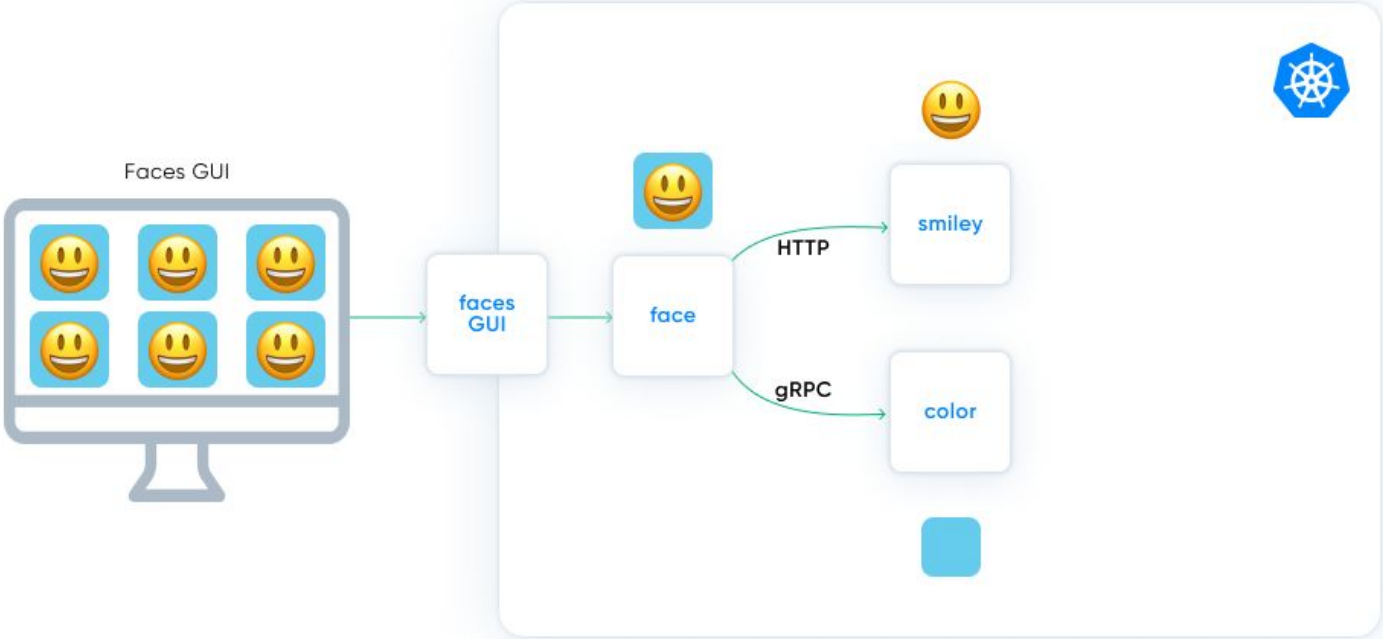
Faces (http://github.com/BuoyantIO/faces-demo)



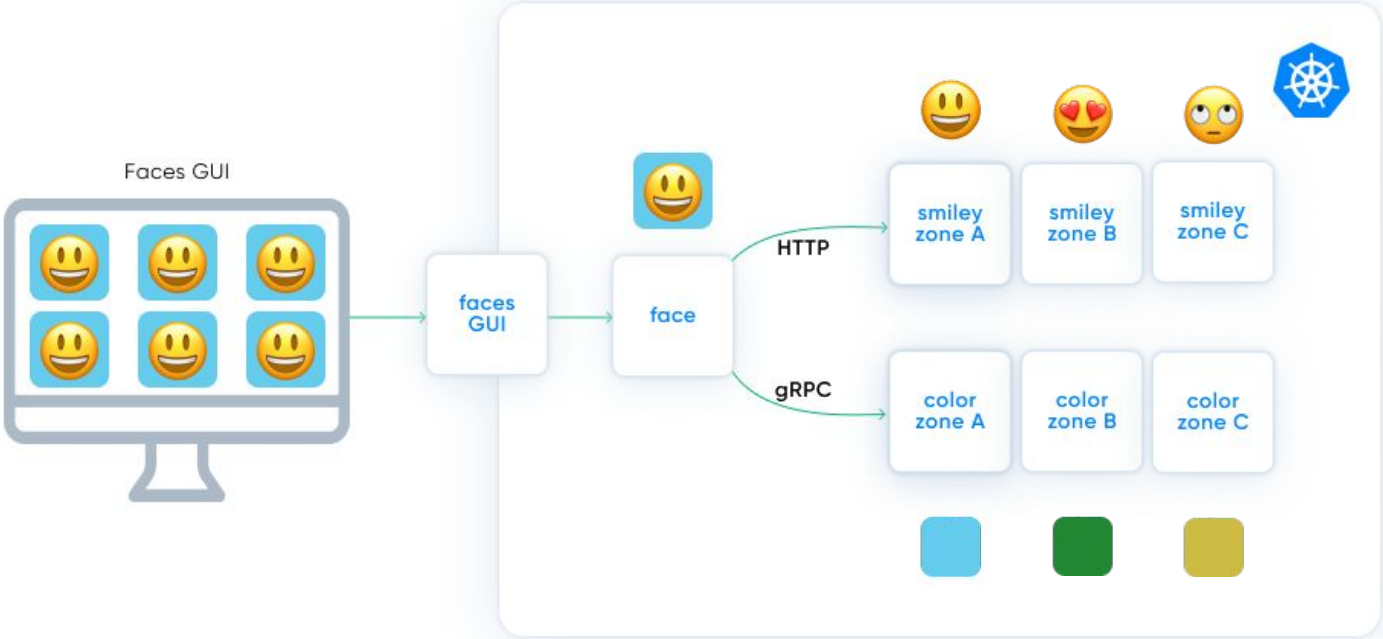
Faces (http://github.com/BuoyantIO/faces-demo)



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

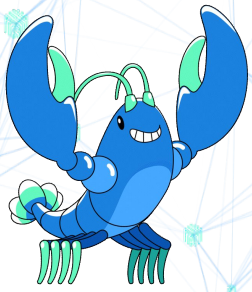


Faces (http://github.com/BuoyantIO/faces-demo)



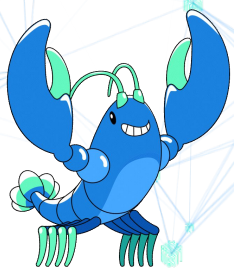
BUOYANT
SERVICE MESH
ACADEMY

DEMO 🤞



BUOYANT
SERVICE MESH
ACADEMY

Gotchas



Gotchas

- Routing happens **where the request is made**
  - This means that HAZL affects *outbound* traffic, not inbound traffic
  - Think of it like retries or circuit breaking
- Don't forget that you configure HAZL with *per-endpoint* numbers!
  - …even though the metrics report scaled numbers
- You can't currently configure HAZL per target: the HAZL load bands affect outbound traffic from a given pod.
  - This is a roadmap item.

https://docs.buoyant.io/buoyant-enterprise-linkerd/latest/reference/hazl/

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

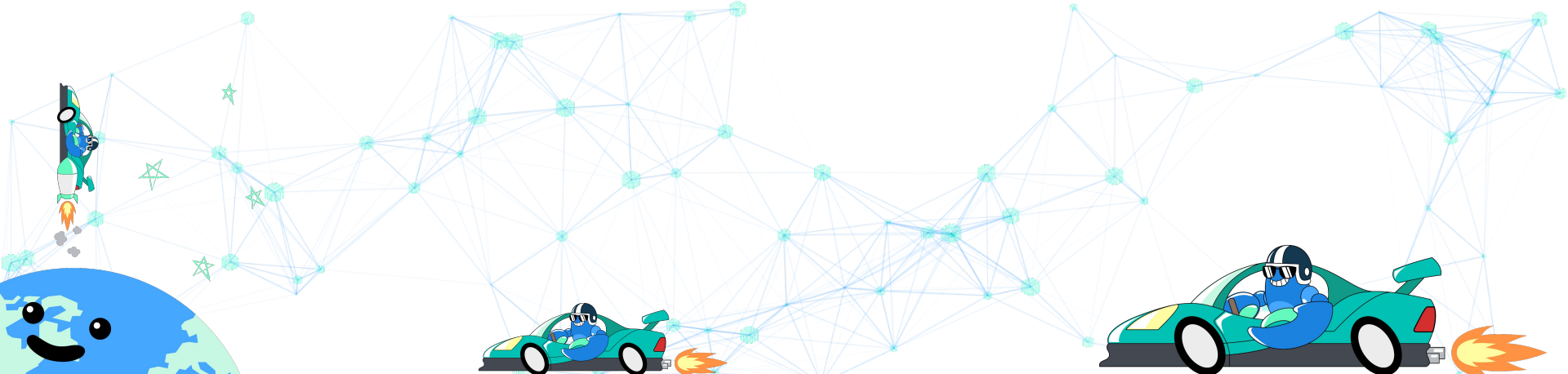
Learn more & try it for free at buoyant.io/**enterprise-linkerd**

BUOYANT
Creators of LINKERD

SERVICE MESH ACADEMY

Get Service Mesh Certified!

With hands-on LINKERD self-paced courses

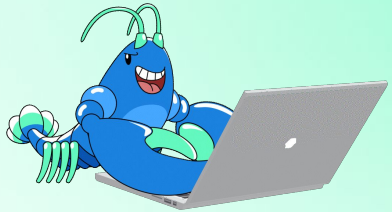


SERVICEMESH ACADEMY | BUOYANT Creators of LINKERD

Kyverno 101... with Linkerd!

Cortney Nickerson
Head of Community, Nirmata

Flynn
Linkerd Evangelist



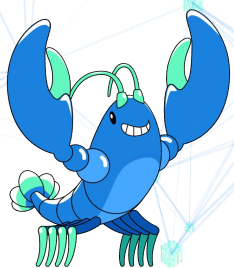
SIGN UP TODAY!
buoyant.io/sma

Thursday, September 18, 2025

9 am PT | 12 pm ET | 6 pm CET

BUOYANT
SERVICE MESH
ACADEMY

Q&A



BUOYANT
SERVICE MESH
ACADEMY

Thanks much!

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