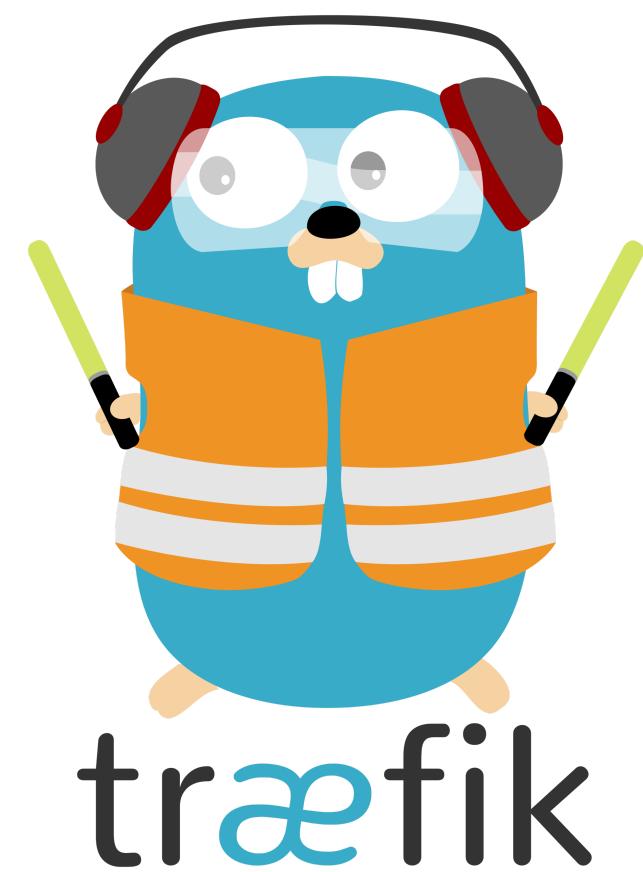


# How To Easily Deploy And Manage Your Microservices With Traefik And Maesh

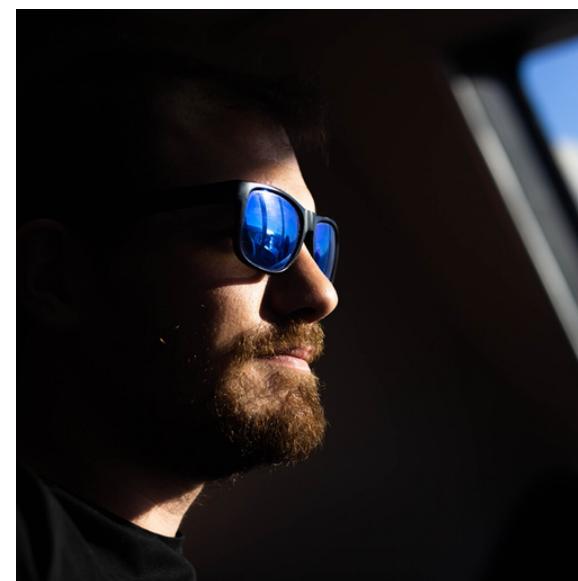


Scale 18x - March 8th 2020

# Whoami

Julien Levesy

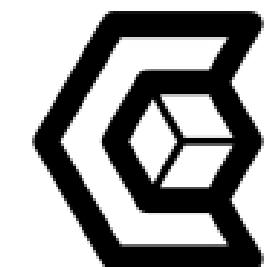
- Software Engineer @ Containous
- Works on Traefik Enterprise Edition
-  jlevesy
-  jlevesy



# Containous

<https://containo.us>

- We Deliver Traefik, Maesh, Yaegi and Traefik Enterprise Edition
- We Believe in Open Source
- 30-ish (and counting !) people from everywhere



**CONTAINOUS**

# Agenda

- Why Traefik ?
- What is Traefik ?
- Core concepts
- Demo
- A word about service mesh
- Doing more with Traefik

# Why Traefik



Why, Mr Anderson?

# Evolution Of Software Design

Mainframe → 3-tiers → Microservices

# The Promise Of Microservices...

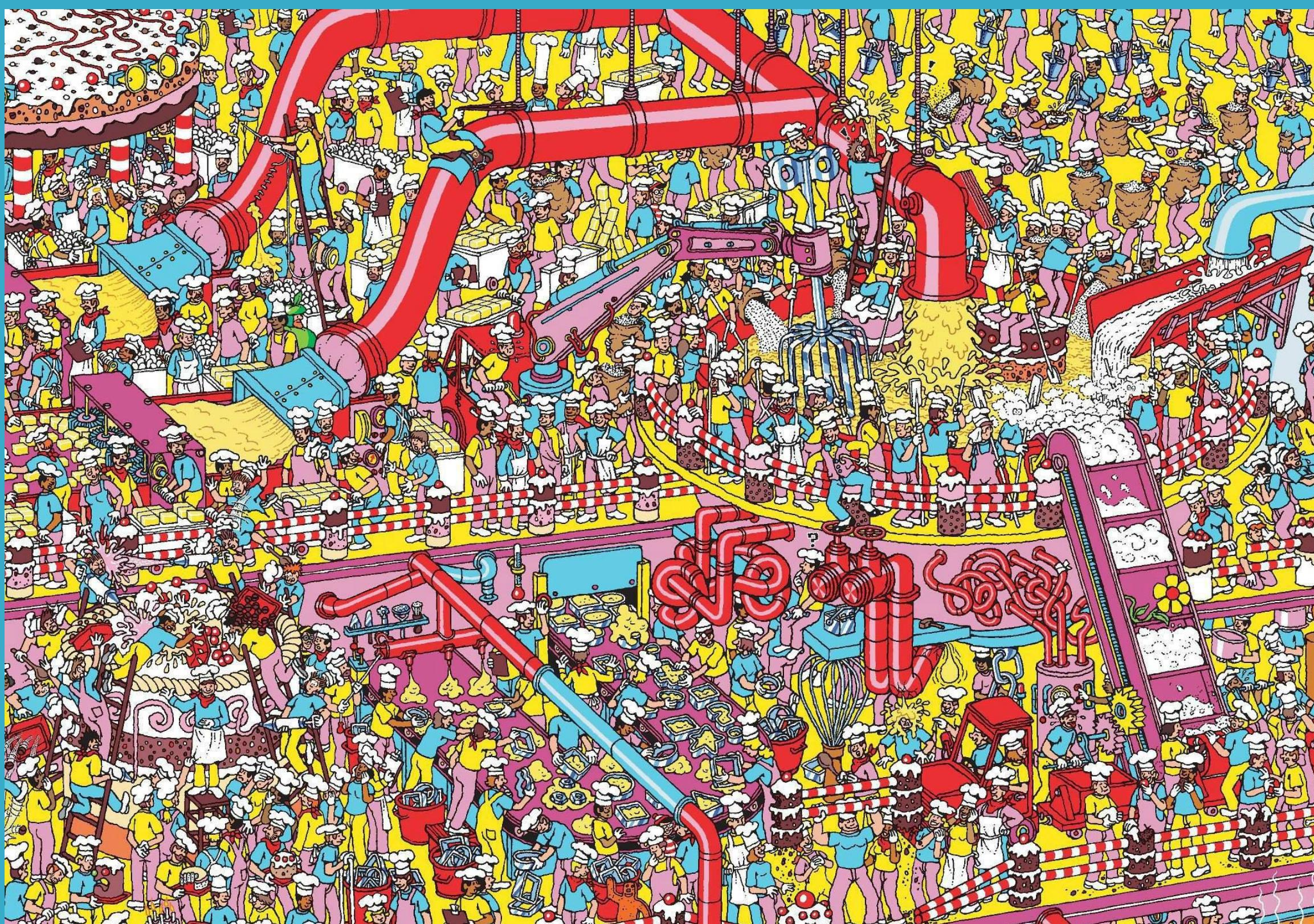


...And What Happens !

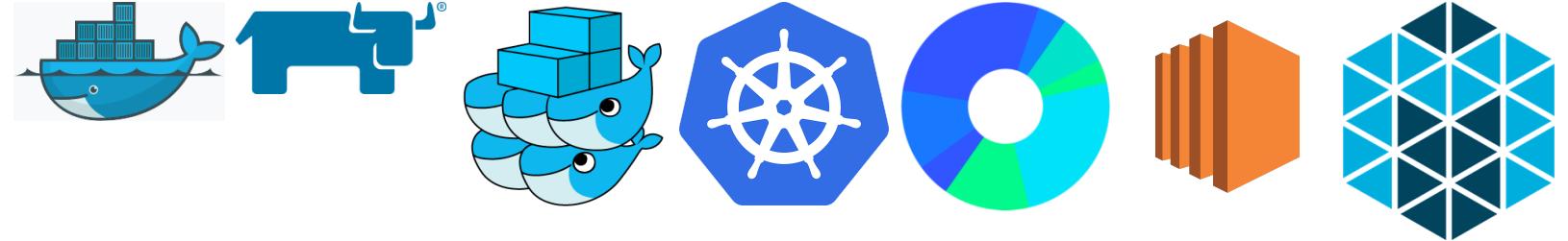


# How Do I Talk To My Service ?

(from outside)



# Tools Of The Trade



# Configuring Proxies With Microservices IS Hard

(Imagine the configuration file)

# What If I Told You?



That You Don't Have to Write This Configuration File

# Here Comes Traefik!



# What Is Traefik ?

- TCP, UDP (incoming!) and HTTP Reverse Proxy
- Fully Dynamic
- Expressive Routing Rule Syntax
- Middlewares
- Advanced Loadbalancing
- Versatile

# Traefik Project



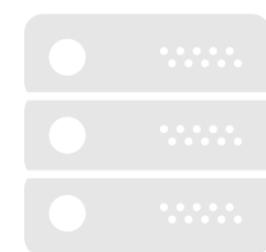
**27,000+**

stars on GitHub



**450+**

contributors



**100K +**  
living  
instances



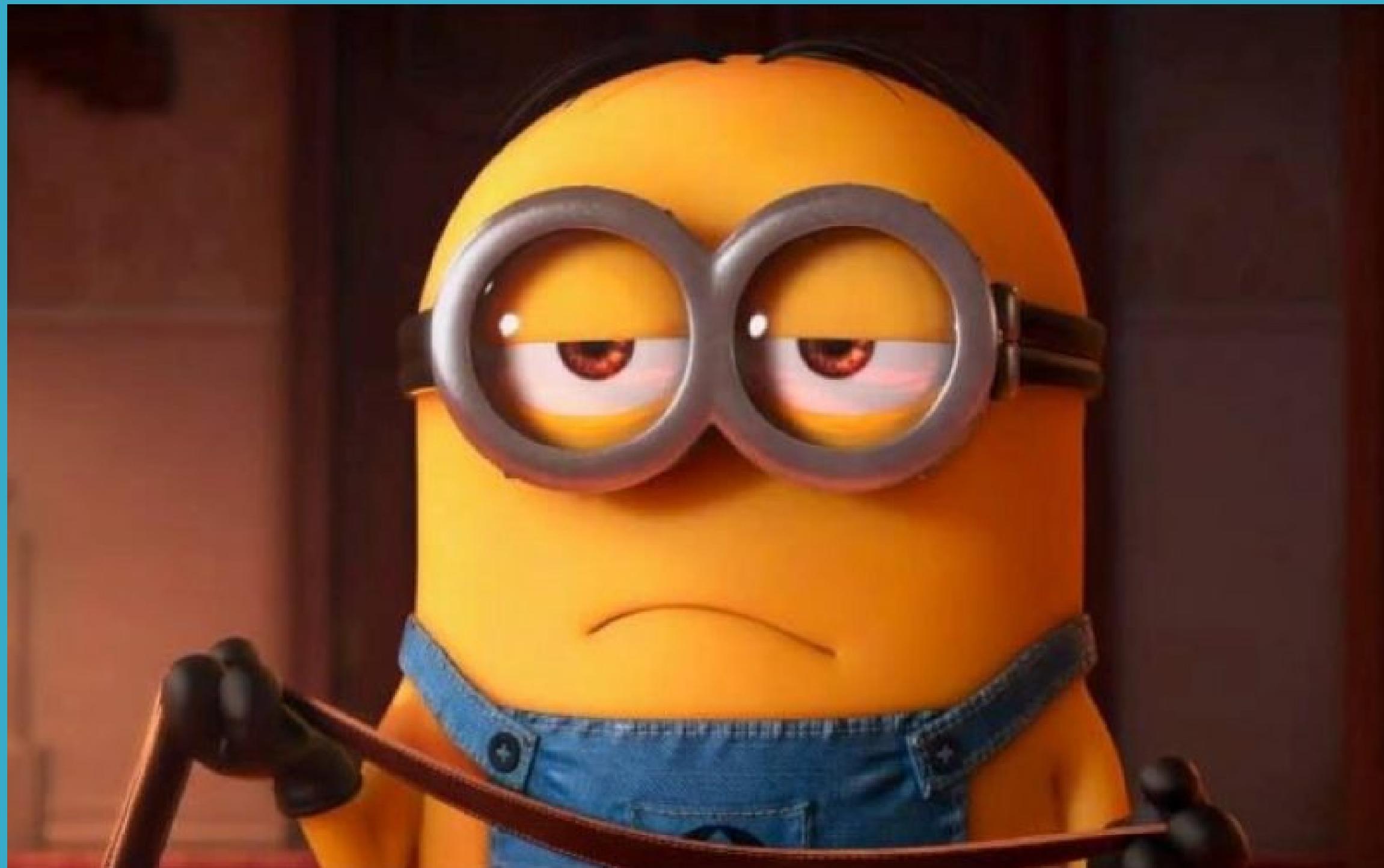
**1.4 B+**

downloads

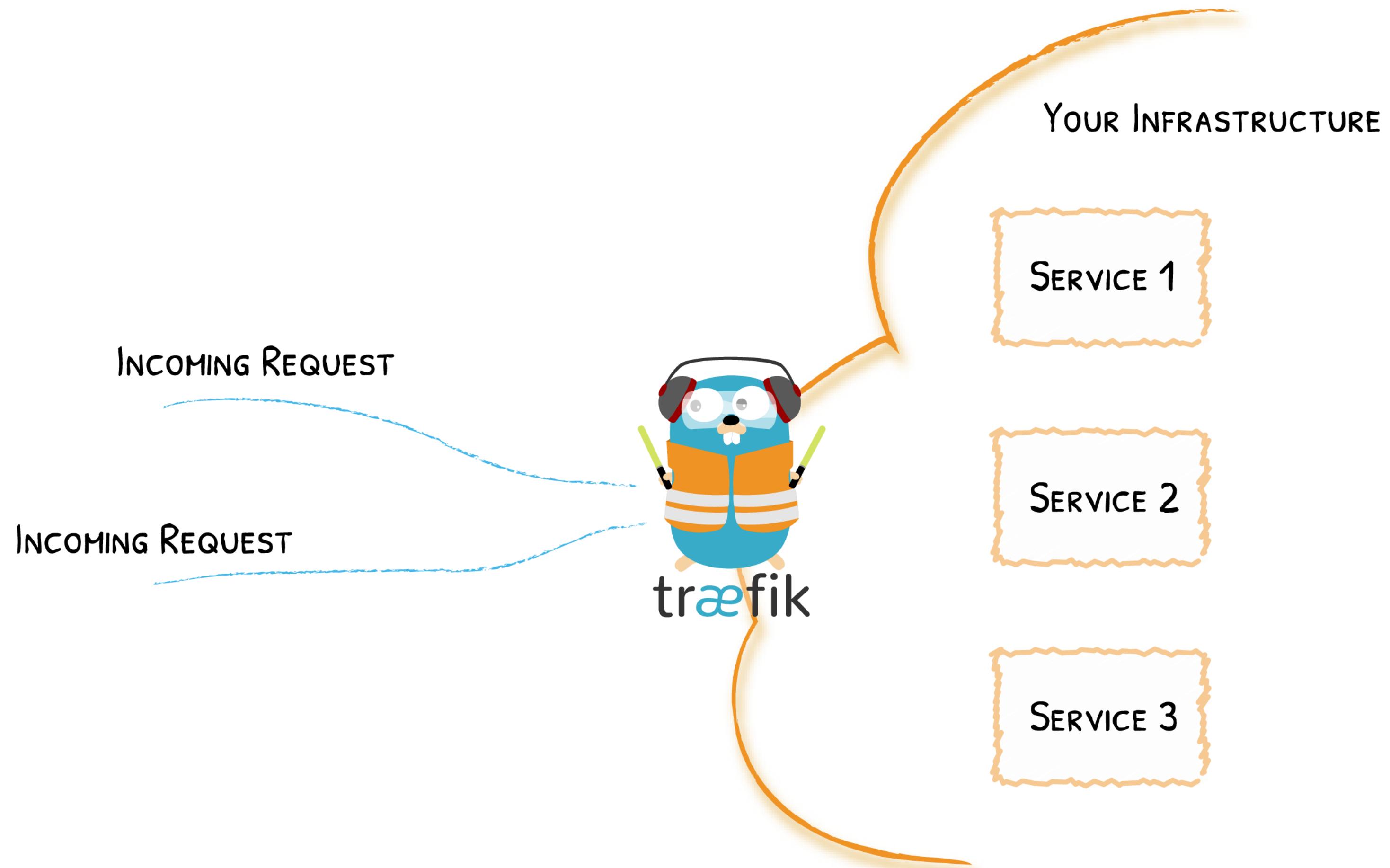


**TOP 15**  
golang project

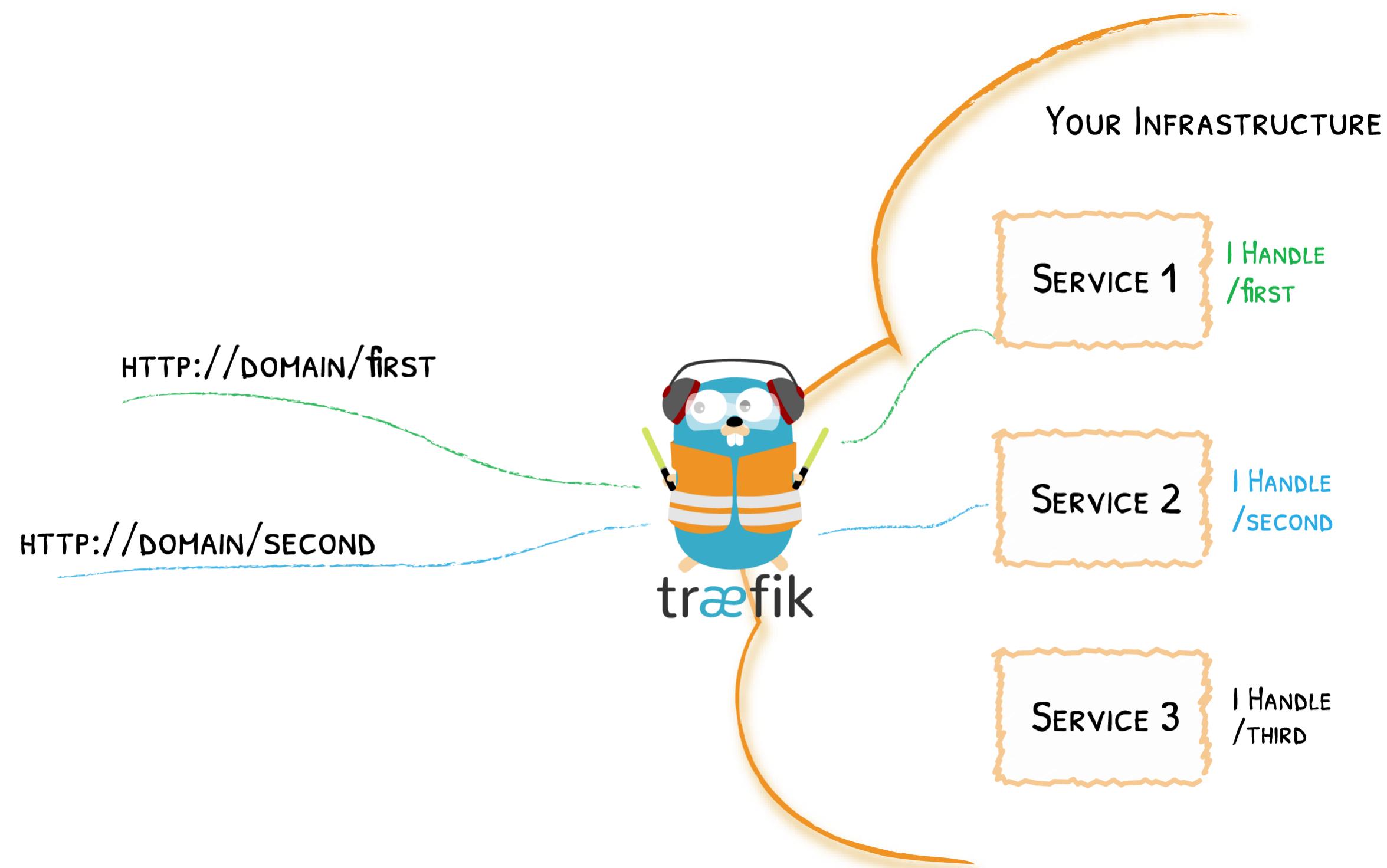
# Traefik Core Concepts



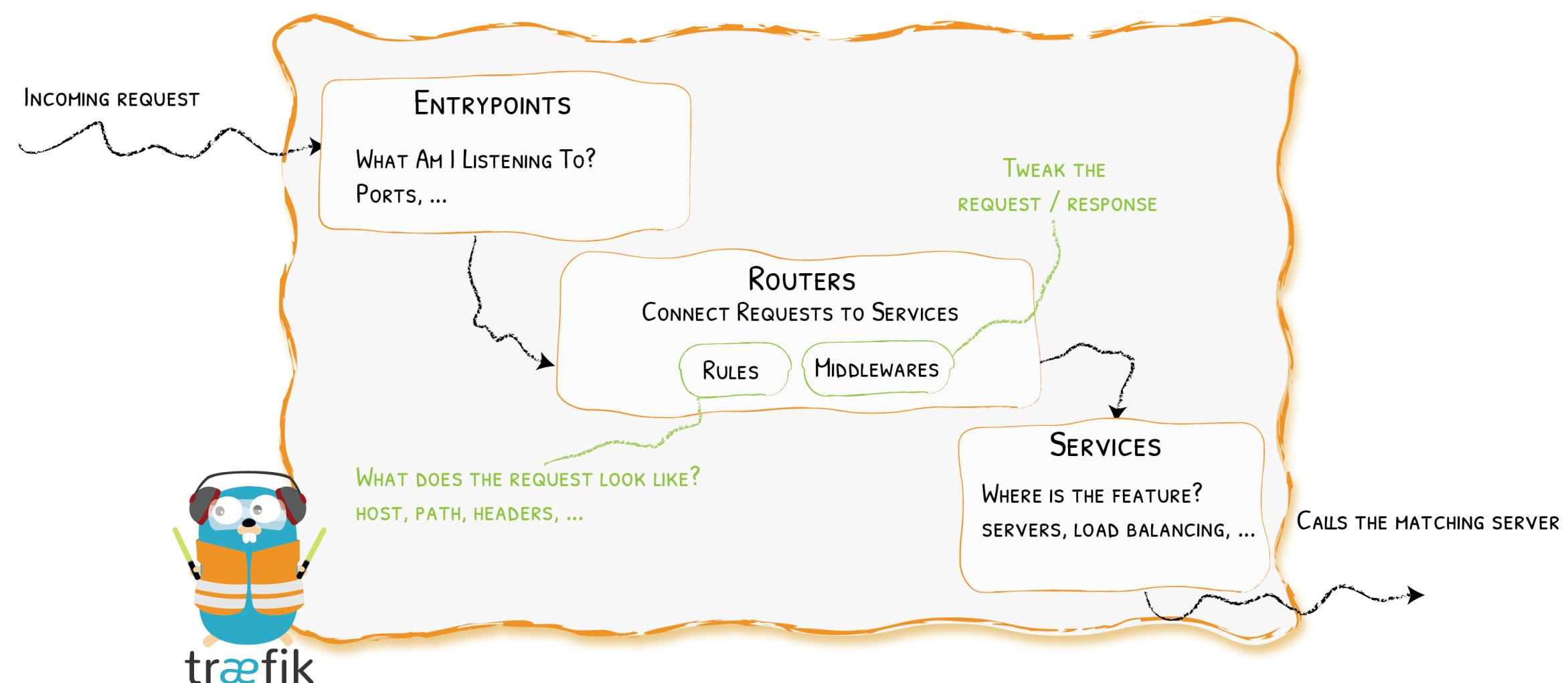
# Traefik Is An Edge Router



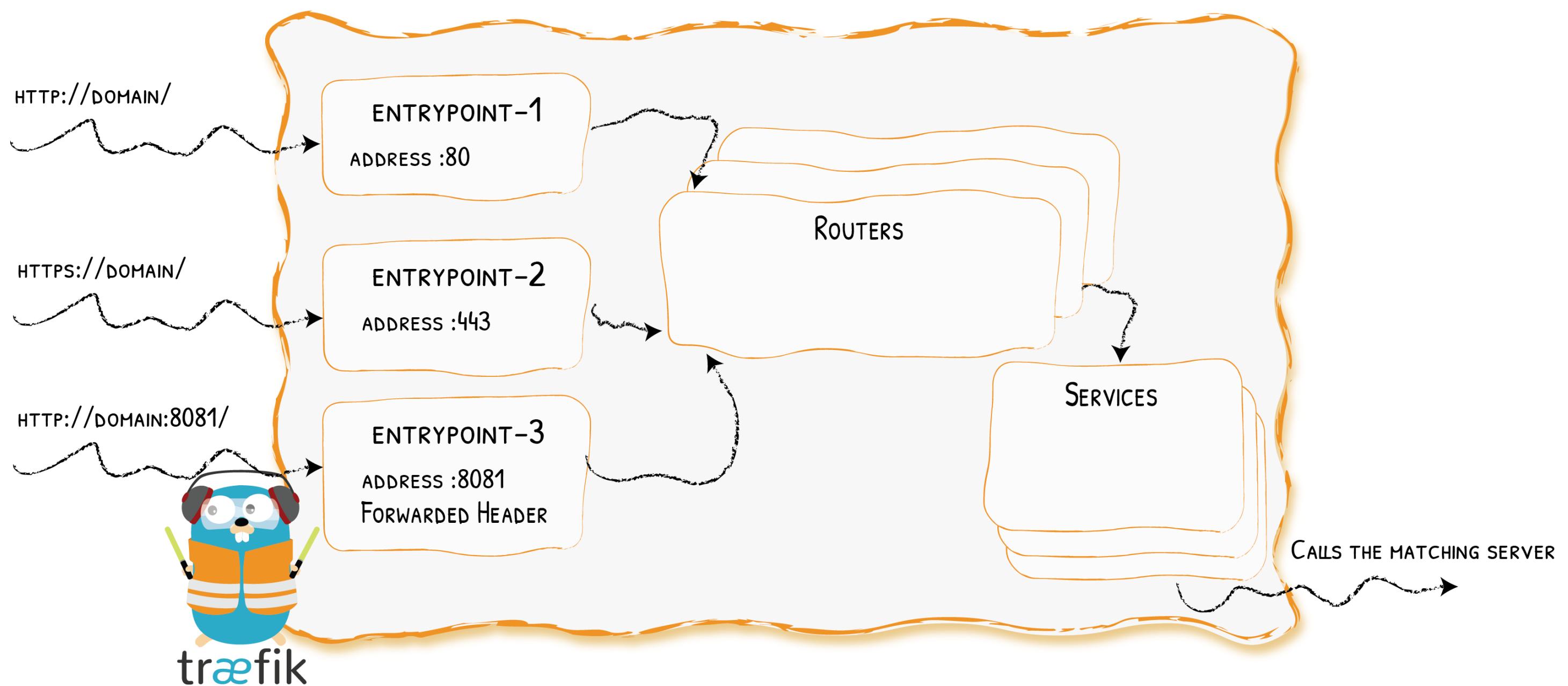
# Dynamically Discovers Services



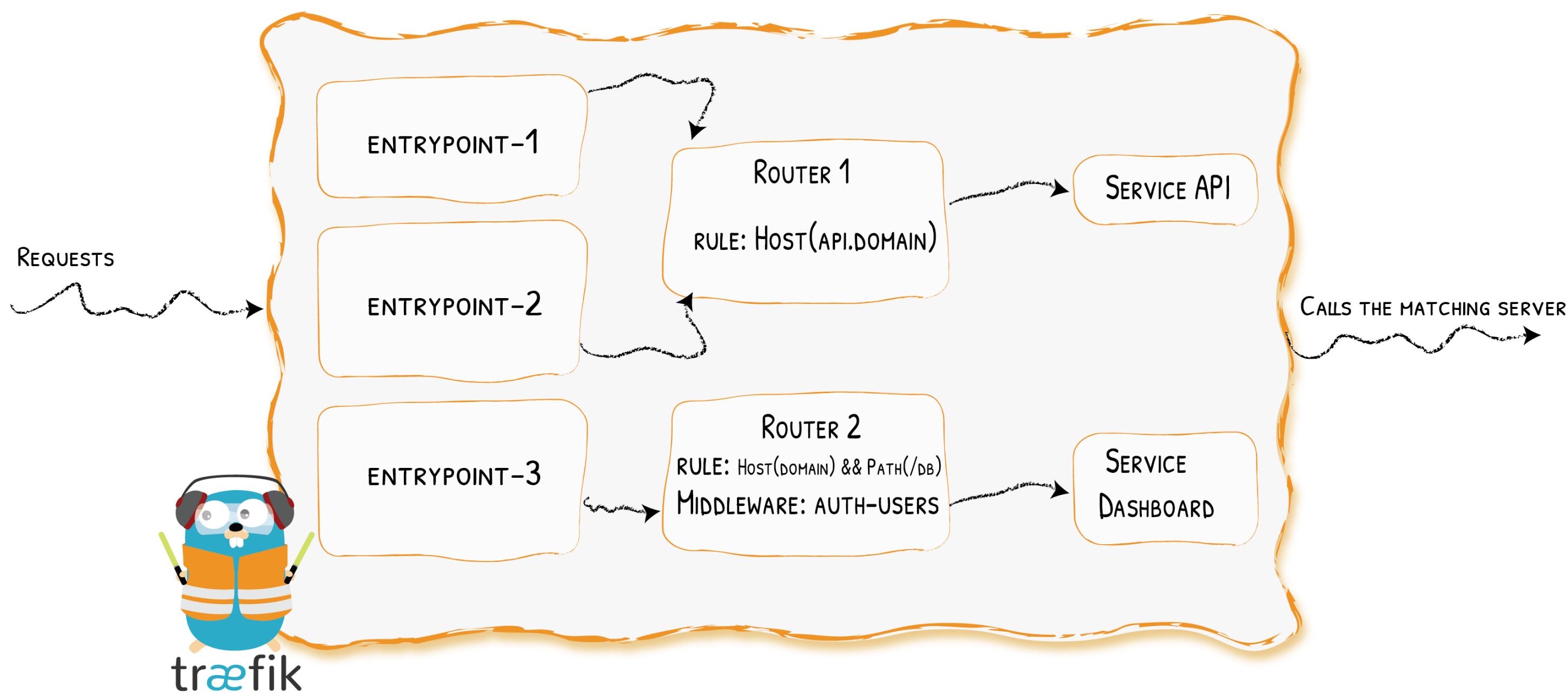
# Architecture At A Glance



# Entrypoints



# Routers



# Routing Rules

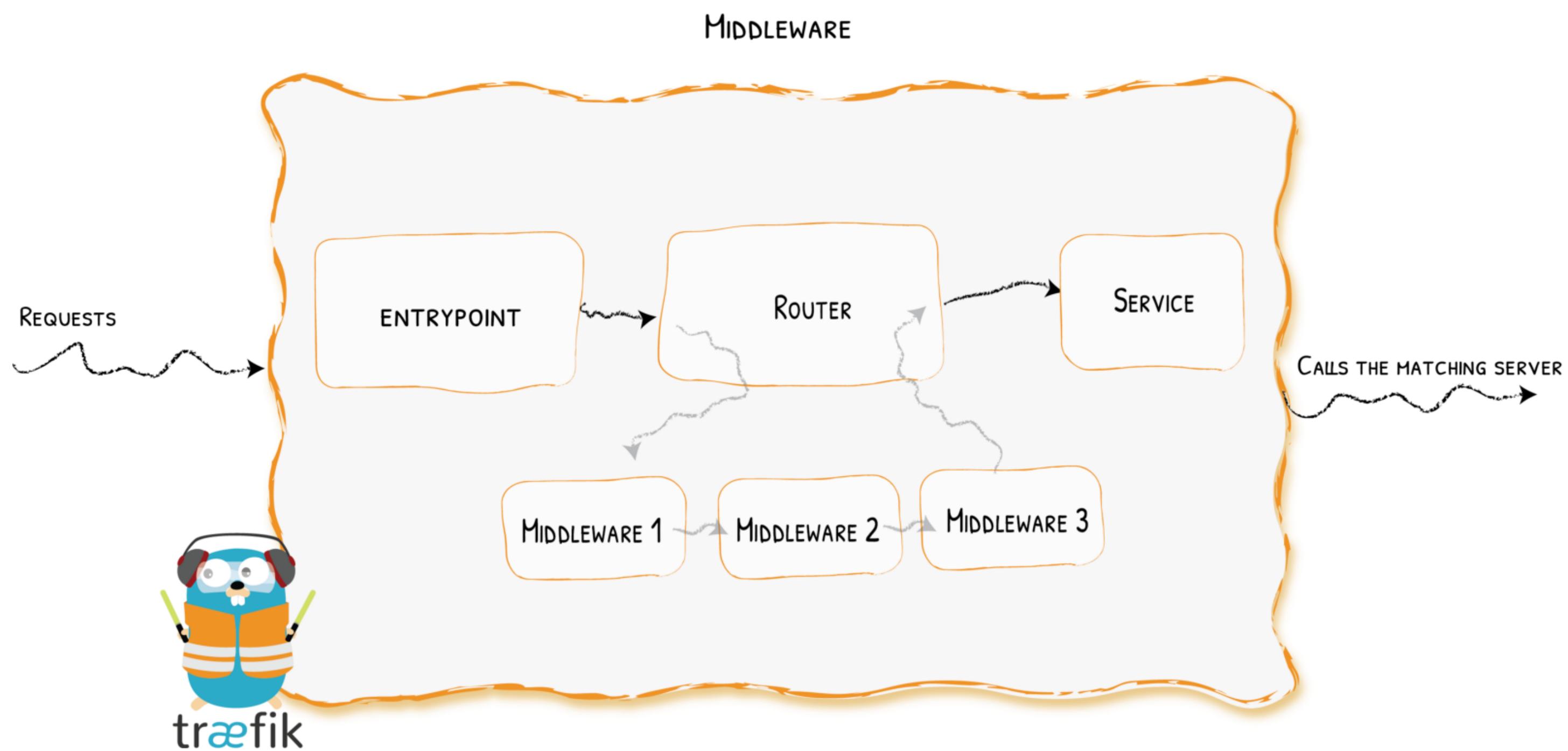
## HTTP

```
# The host is `traefik.io` OR the host is `containo.us` and the path this `/traefik`
Host(`traefik.io`) || (Host(`containo.us`) && Path(`/traefik`))
```

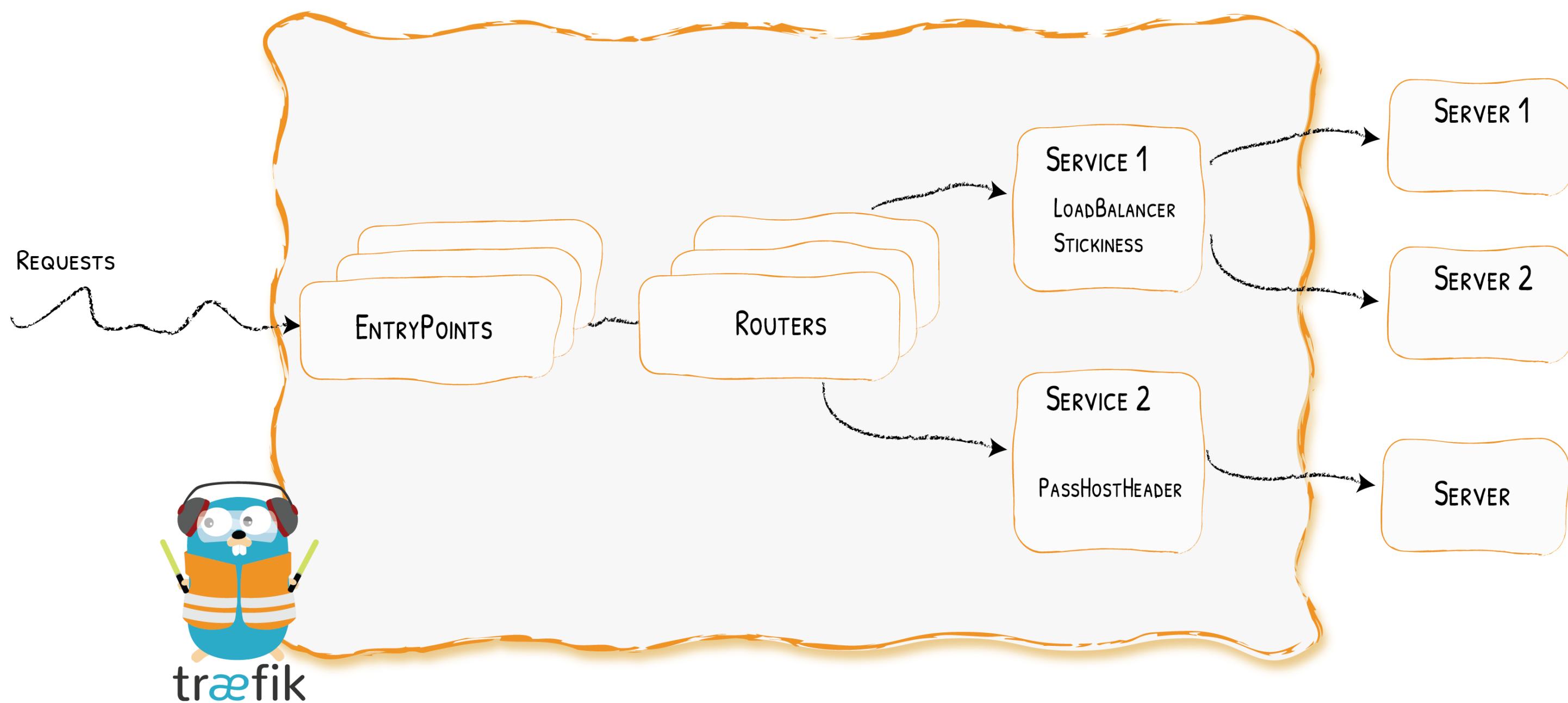
## TCP

```
# If the incoming connection is TLS and has SNI enabled.
HostSNI(`traefik.io`)
```

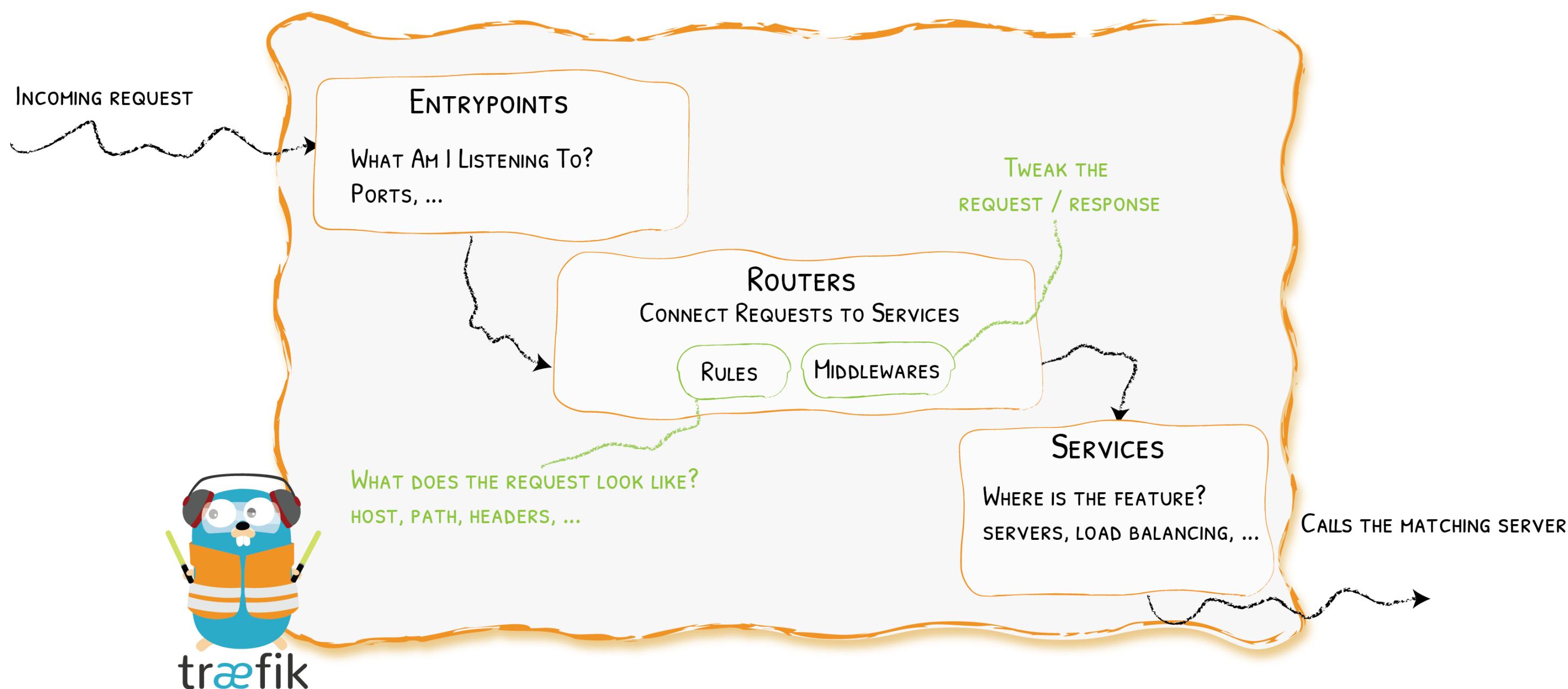
# Middlewares



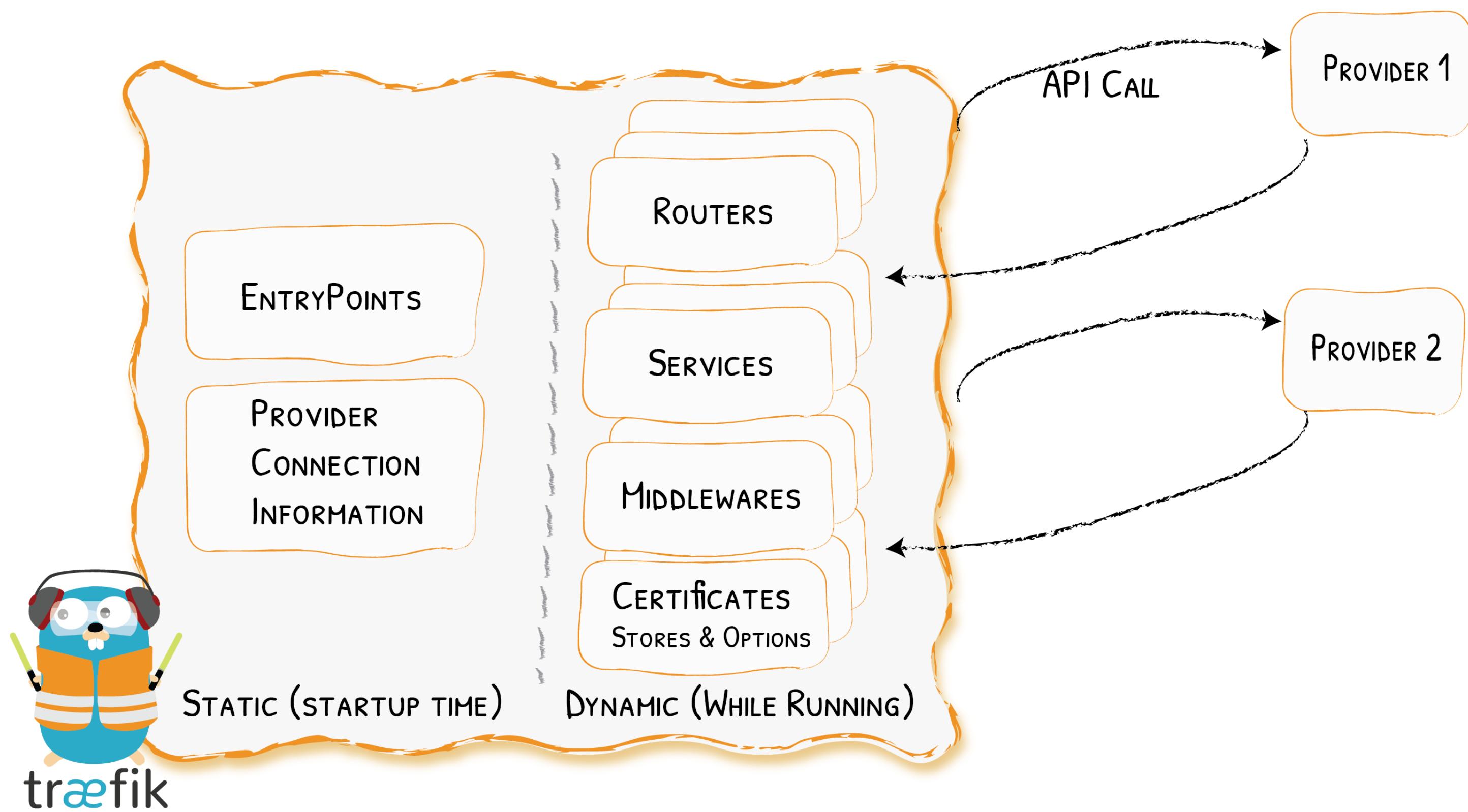
# Services



# Architecture (Again) At A Glance



# Static Vs Dynamic



# Traefik With ⚓

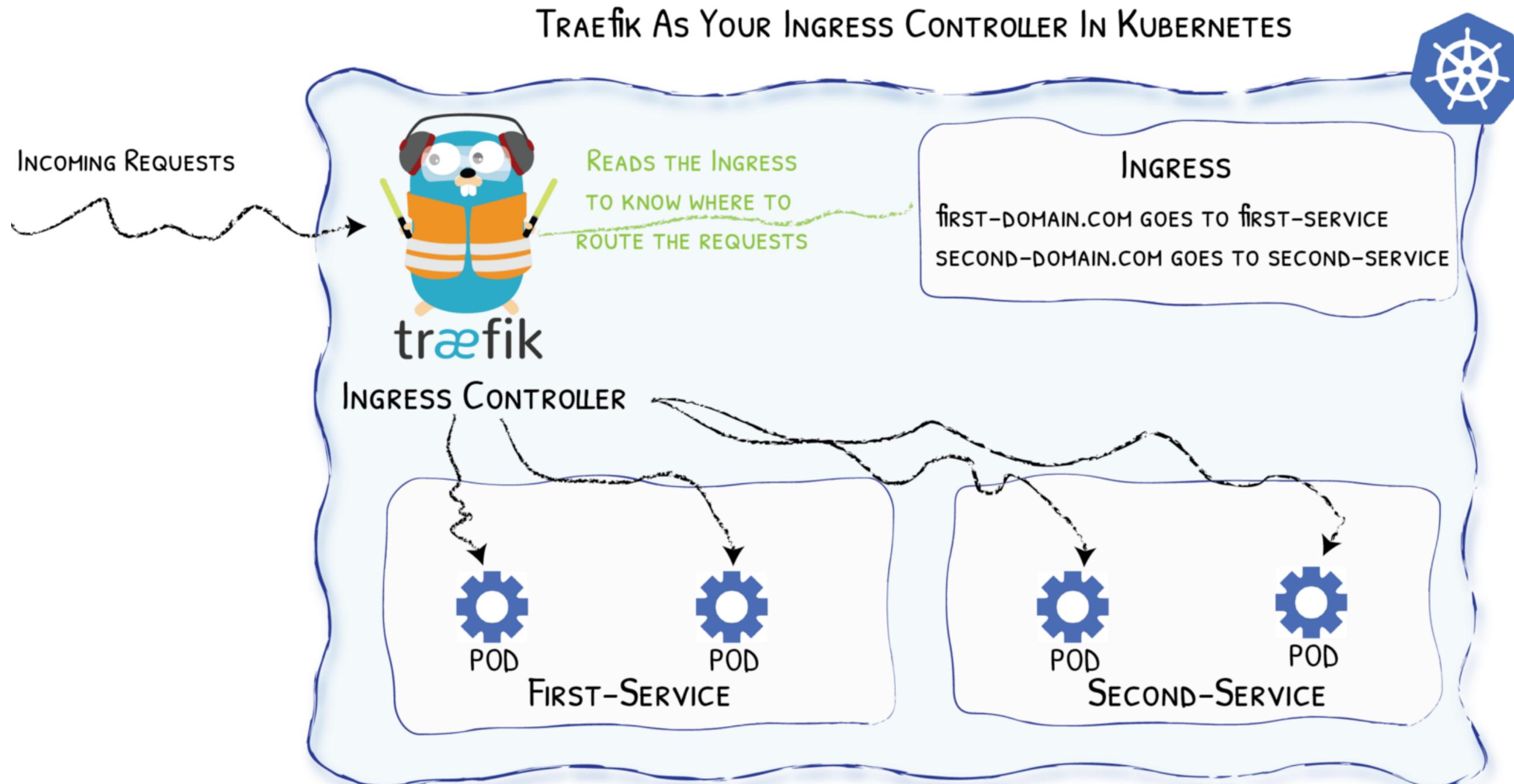


Diagram from <https://medium.com/@geraldcroes>

# ✳️ Ingress

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: corporate-webapp
  annotations:
    kubernetes.io/ingress.class: 'traefik'
spec:
  rules:
  - host: localhost
    http:
      paths:
      - backend:
          serviceName: corporate-webapp
          servicePort: 80
```

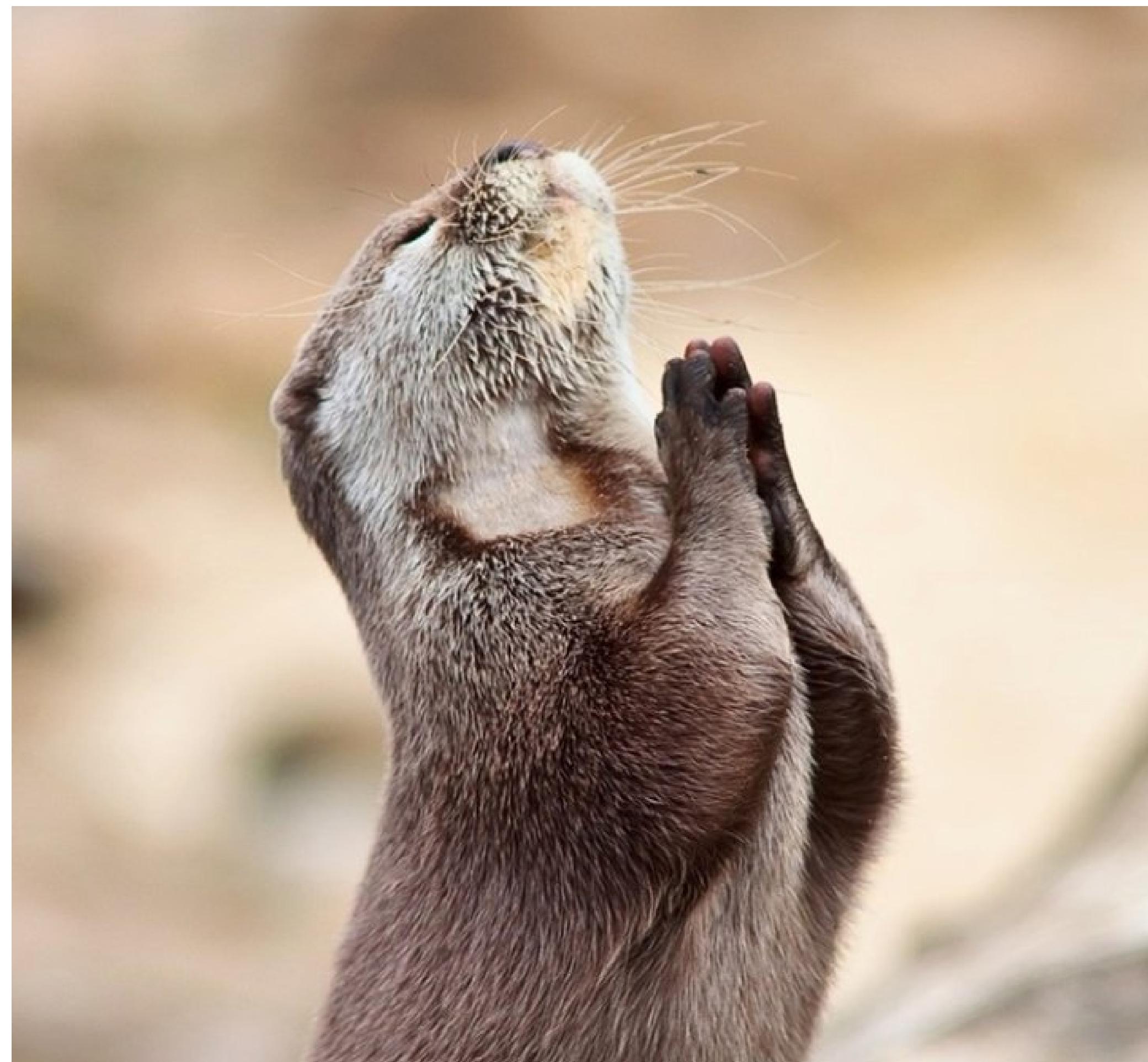
# ✳️ Ingress Route (HTTP)

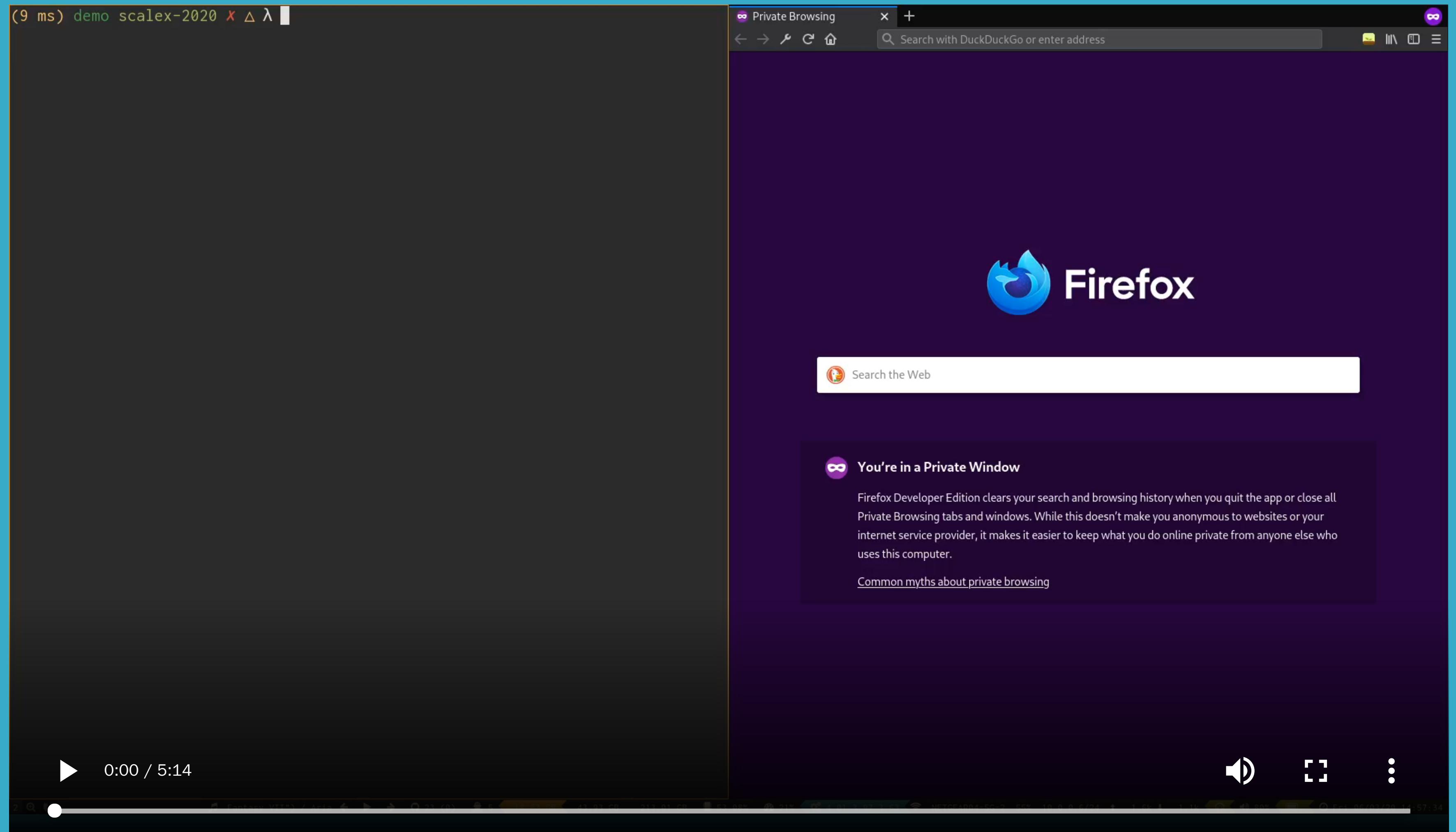
```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: myapp
spec:
  entryPoints:
    - web
  routes:
    - match: Host(`localhost`) && PathPrefix(`/whoami`)
      kind: Rule
      services:
        - name: webapp
          port: 80
```

# 🌐 Ingress Route (TCP)

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRouteTCP
metadata:
  name: ingressroutetcpmongo.crd
spec:
  entryPoints:
    - mongotcp
  routes:
    - match: HostSNI(`mongo-prod`)
      services:
        - name: mongo-prod
          port: 27017
```

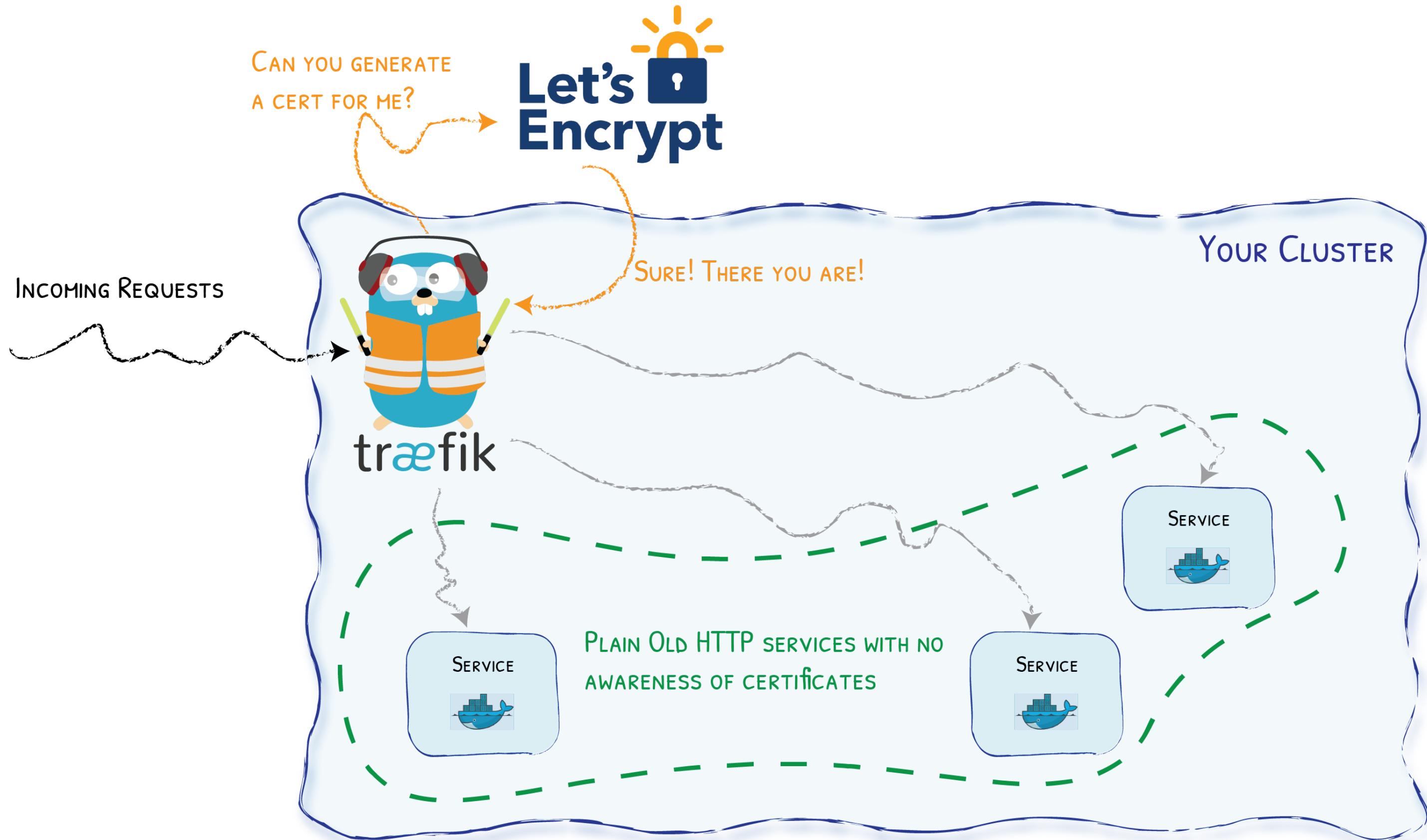
# Demo



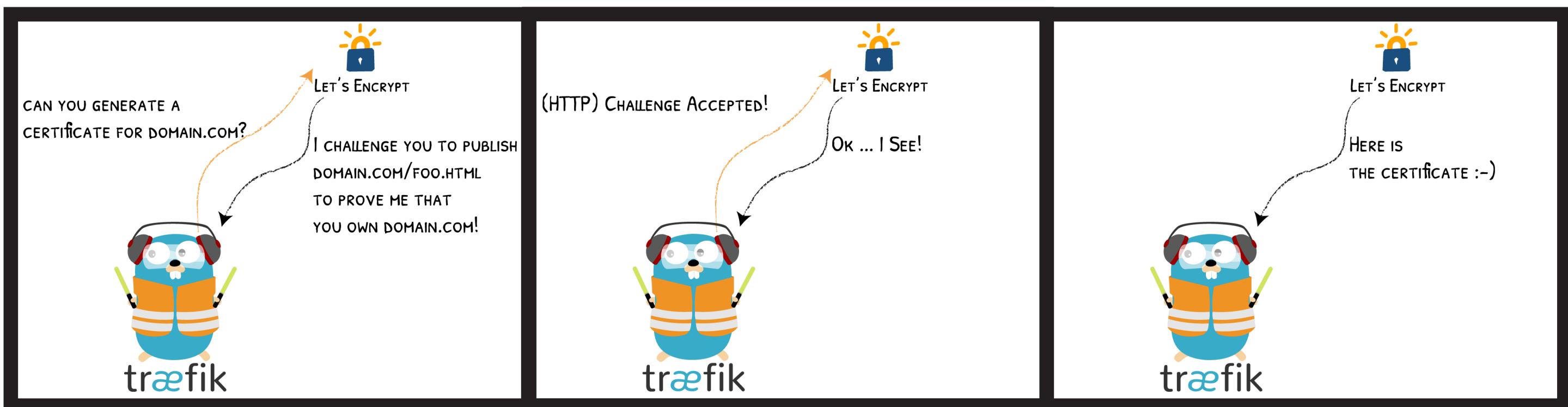


# Traefik And Let's Encrypt

# HTTPS & Let's Encrypt



# Let's Encrypt HTTP Challenge



# HTTP-01 Pro And Cons

- ✓ Easy to understand
- ✗ Requires port 80 to be exposed publicly
- ✗ No wildcard certificates

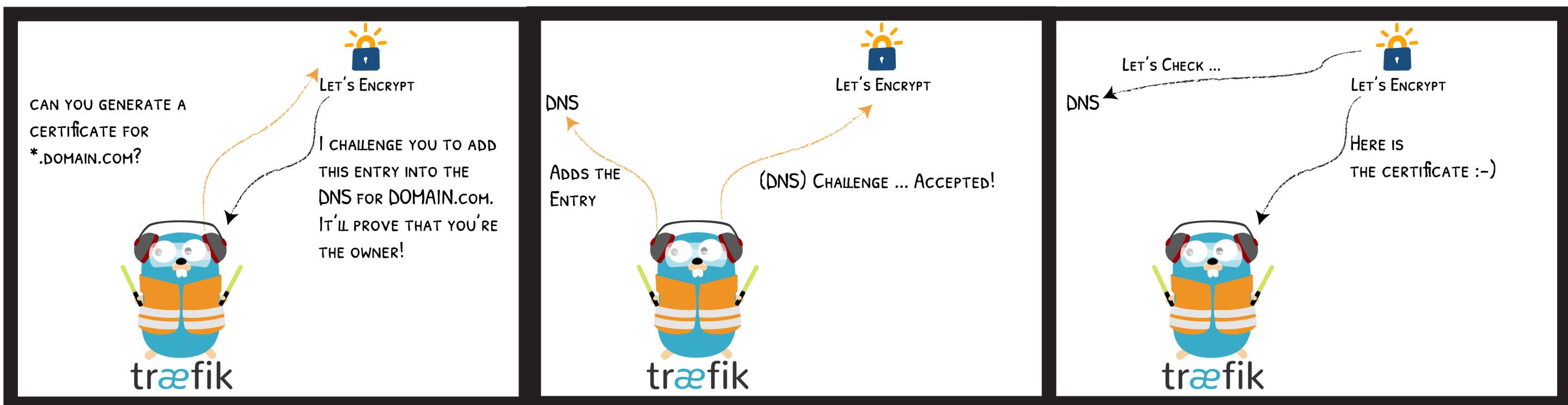
# Let's Encrypt TLS-ALPN Challenge



# TLS-ALPN-01 Pro And Cons

- ✓ More Secure
- ✗ No wildcard certificates

# Let's Encrypt DNS Challenge



# DNS-01 Challenge Pro And Cons

- ✓ Supports wildcard certificates
- ✓ Works for non exposed instances
- ✗ Needs DNS provider with an API

# Certificate Resolvers

## static-config.toml

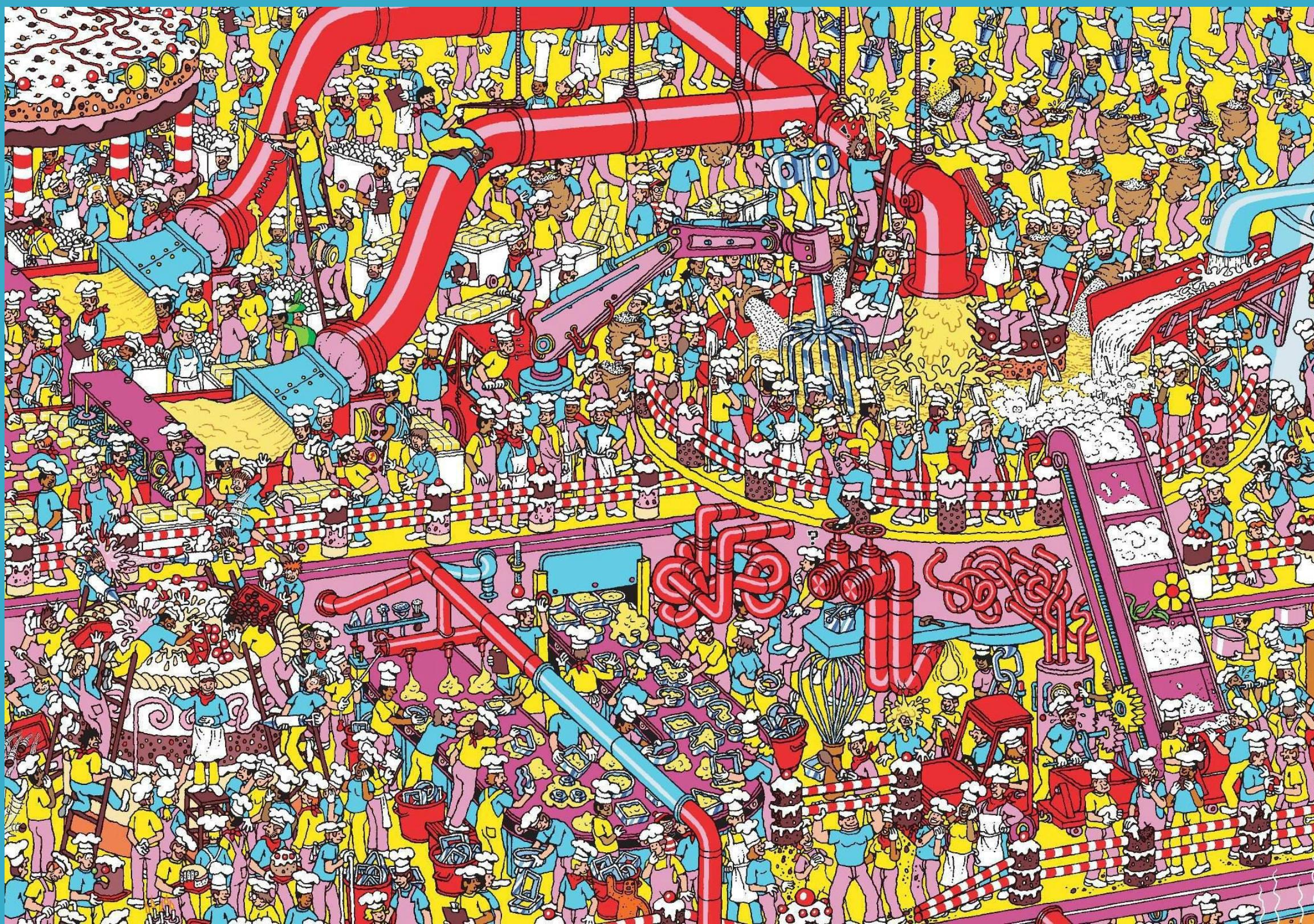
```
# [...]
[certificatesResolvers.tls.acme]
email = "your-email@your-domain.org"
[certificatesResolvers.tls.acme.tlsChallenge]
```

## app-route.yaml

```
----
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: blogtls
spec:
  # [...]
  routes:
  - match: Host(`company.com`) && Path(`/blog`)
  # [...]
  tls:
    certresolver: tls
```

# How Do I Talk To My Service ?

(from inside)



# Modern Infrastructure Needs

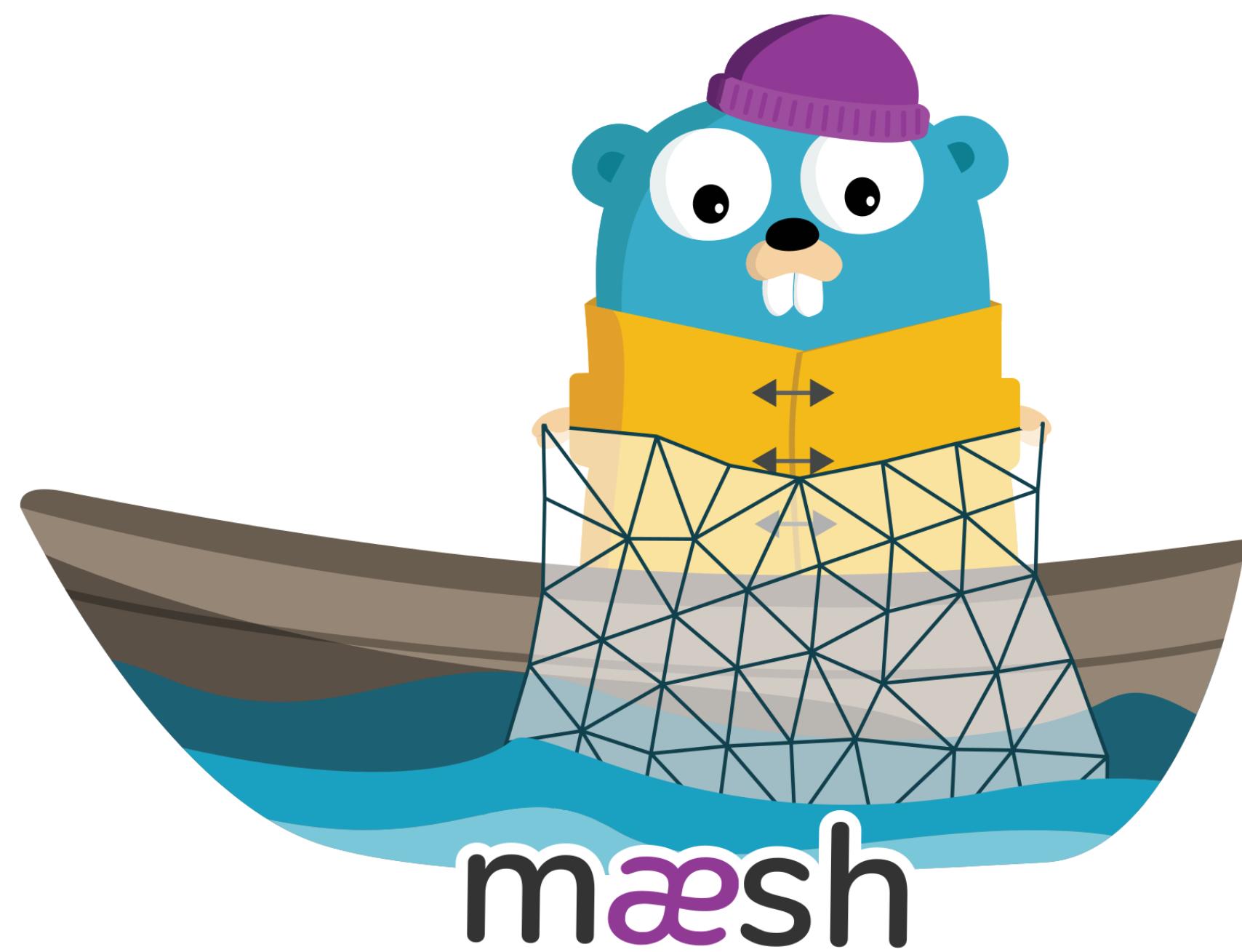
- Observability
- Access control
- Routing

# Instrumenting Applications At Scale Is Costly

(Developers should focus on business logic)

# Lets Instrument The Network!

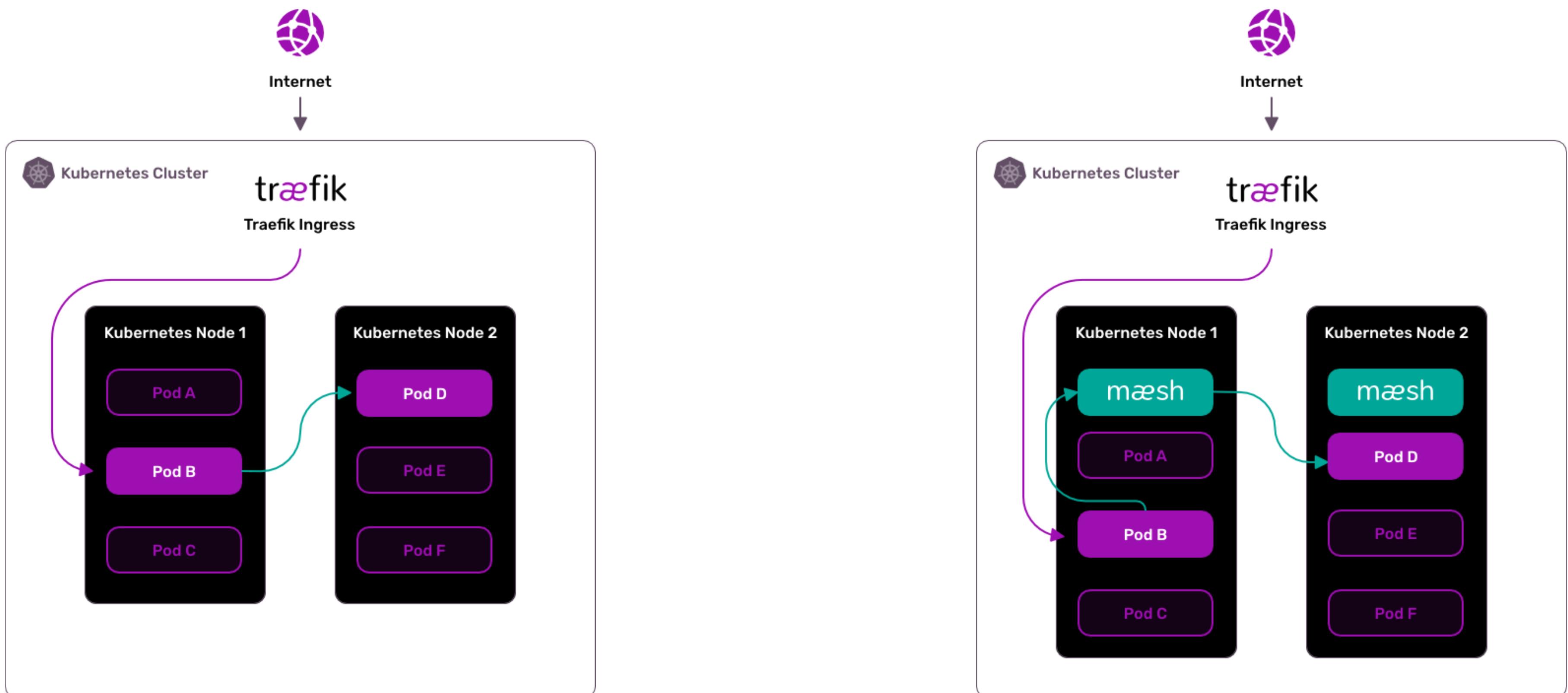
# Say Hello To Maesh



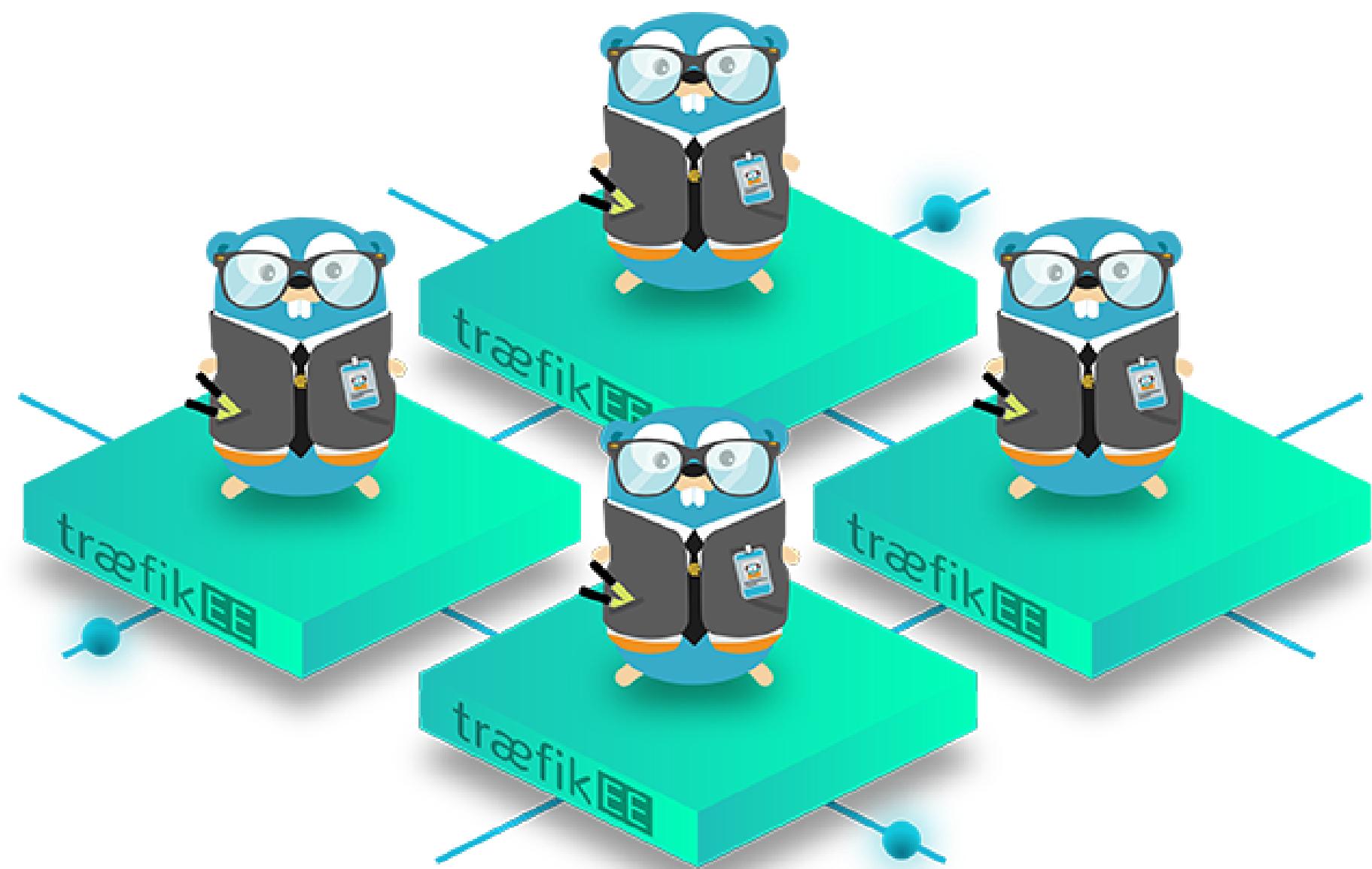
# What Is Maesh?

- Lightweight service mesh
- Provides routing, access control and observability
- Fully based on Traefik
- Non-invasive and opt-in
- SMI compliant

# Maesh Architecture



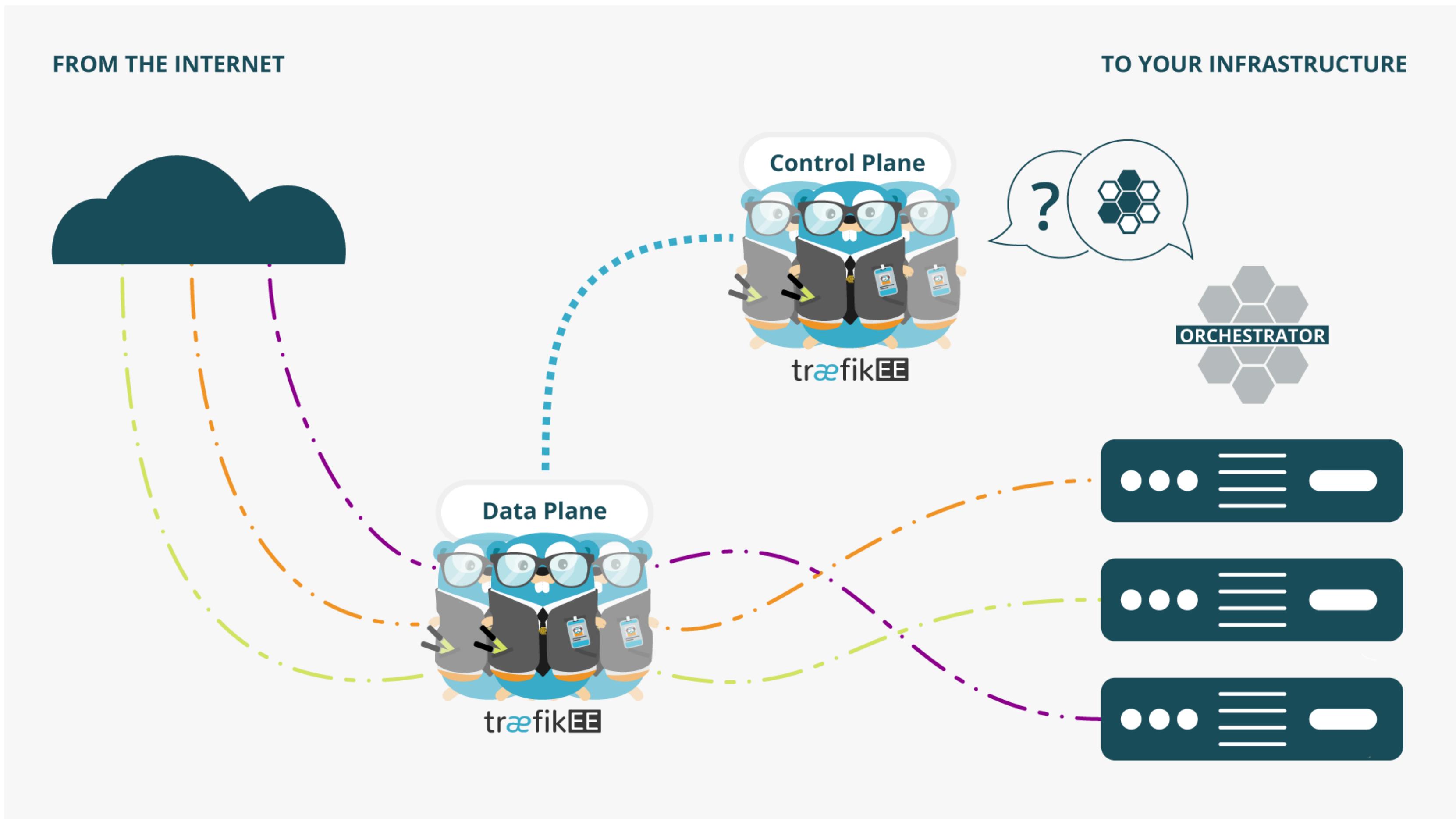
# Doing More With Traefik



# What Is TraefikEE ?

- Makes multiple instances of Traefik acts like one
- Highly Available and Secure
- Distributed features
- 24/7 support

# TraefikEE Architecture



# Free Trial

<https://containo.us/traefikee>

# That's All Folks!



We Have  
Stickers!

traefik

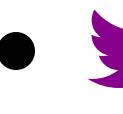
# We Are Hiring!

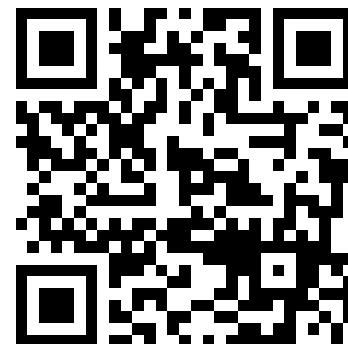


**CONTAINOUS**

```
docker run -it containous/jobs
```

# Thank You!

-  @jlevesy
-  jlevesy



- Slides (HTML): <https://containous.github.io/slides/scalex-2020>
- Slides (PDF): <https://containous.github.io/slides/scalex-2020/slides.pdf>
- Source on : <https://github.com/containous/slides/tree/scalex-2020>