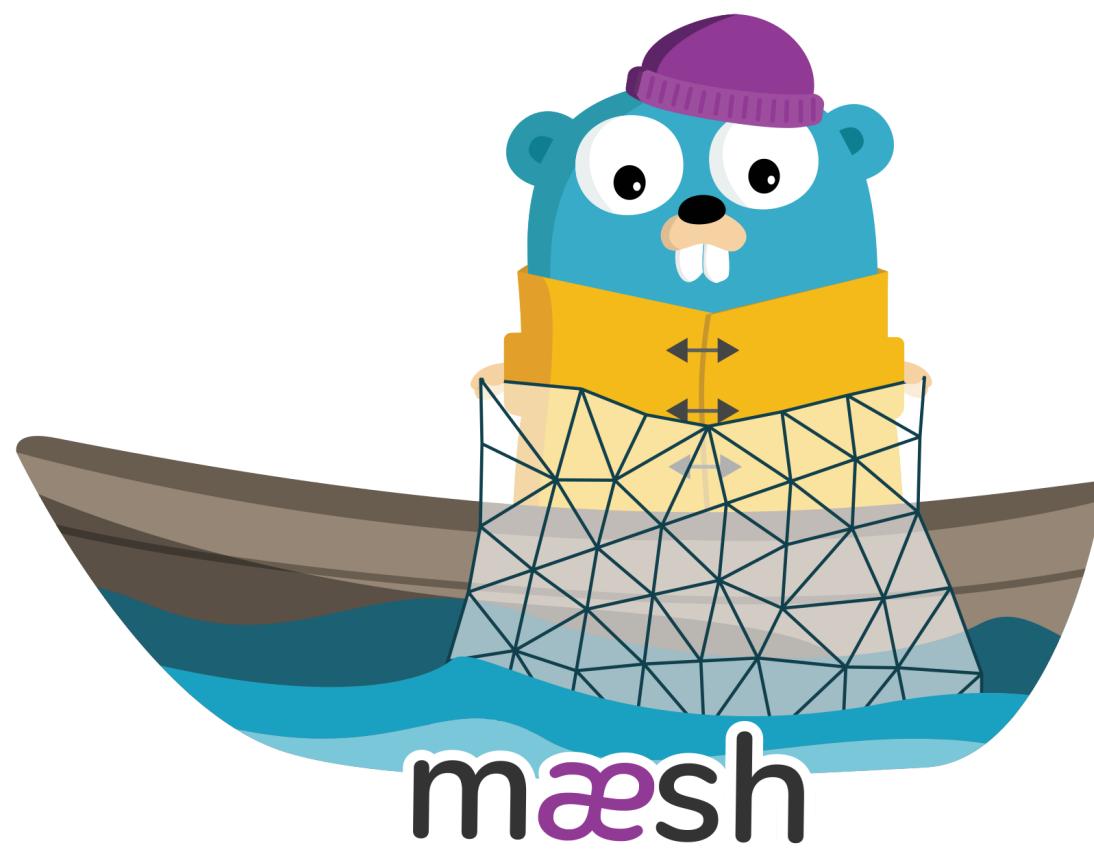


# Maesh Online Meetup - October 2018



Maesh Online Meetup - October 2018

# How To Use These Slides?

- **Browse the slides:** Use the arrows
  - Change chapter: Left/Right arrows
  - Next or previous slide: Top and bottom arrows
- **Overview of the slides:** keyboard's shortcut "o"
- **Speaker mode (and notes):** keyboard's shortcut "s"

# Whoami 1/4

- Michael Matur
  - Devops & Developer @ Containous
- Blacksmith on Traefik
-  @michaelmatur
-  mmatur



# Whoami 2/4

- Gerald Croes:
  - Træfik's VP of Engineering ▯ Advocate @ Containous
-  @geraldcroes
-  geraldcroes



# Whoami 3/4

- Daniel Tomcej:
  - Træfik's Developer  Advocate @ Containous
-  @daniel\_tomcej
-  dtomcej



# Whoami 4/4

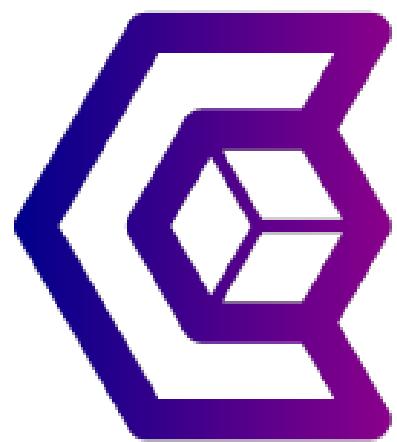
- Manuel Zapf:
  - Træfik's Solution Architect  @ Containous
-  @mZapfDE
-  SantoDE



# Containous

<https://containo.us>

- We Believe in Open Source
- We Deliver Traefik and Traefik Enterprise Edition
- Commercial Support
- 30 people distributed, 90% tech



# What The Hell Is A Service Mesh?

*A service mesh is a configurable network layer inside your cluster to handle network-based communication among your applications. It ensures that this communication is fast, reliable and secure.*

# Benefits

- Provides extended functionality (Service Discovery, Load Balancing...)
- Centralized Place instead of having to reimplement across Services
- ...

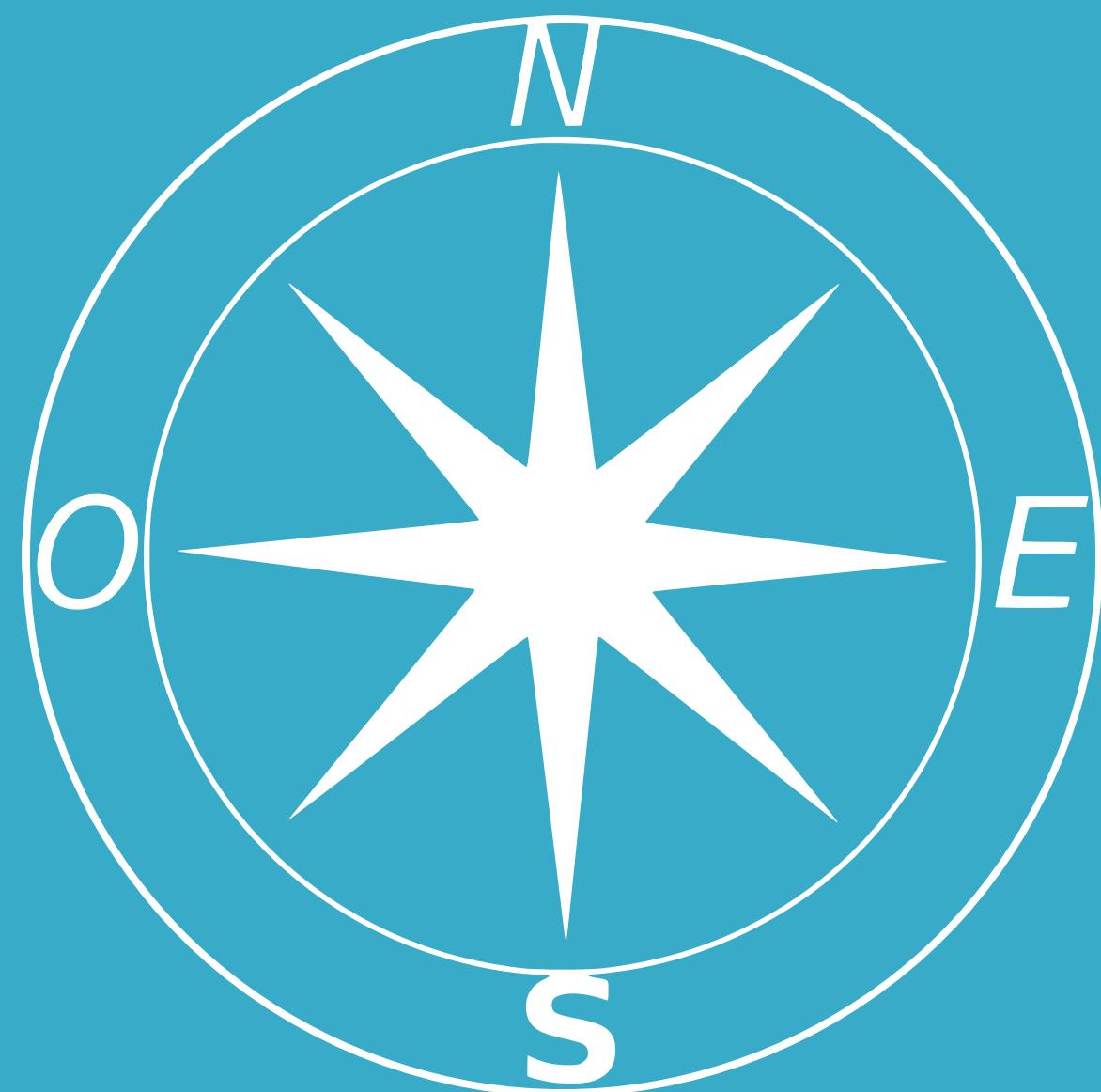
# *Challenges*

- Current solutions can be complex to understand
- Ressource Hungry
- ...

# Say Hello To Maesh



# Traefik And West/East Traffic



# What Is Maesh?

*Maesh is a lightweight, easy to configure, and non-invasive service mesh that allows visibility and management of the traffic flows inside any Kubernetes cluster.*

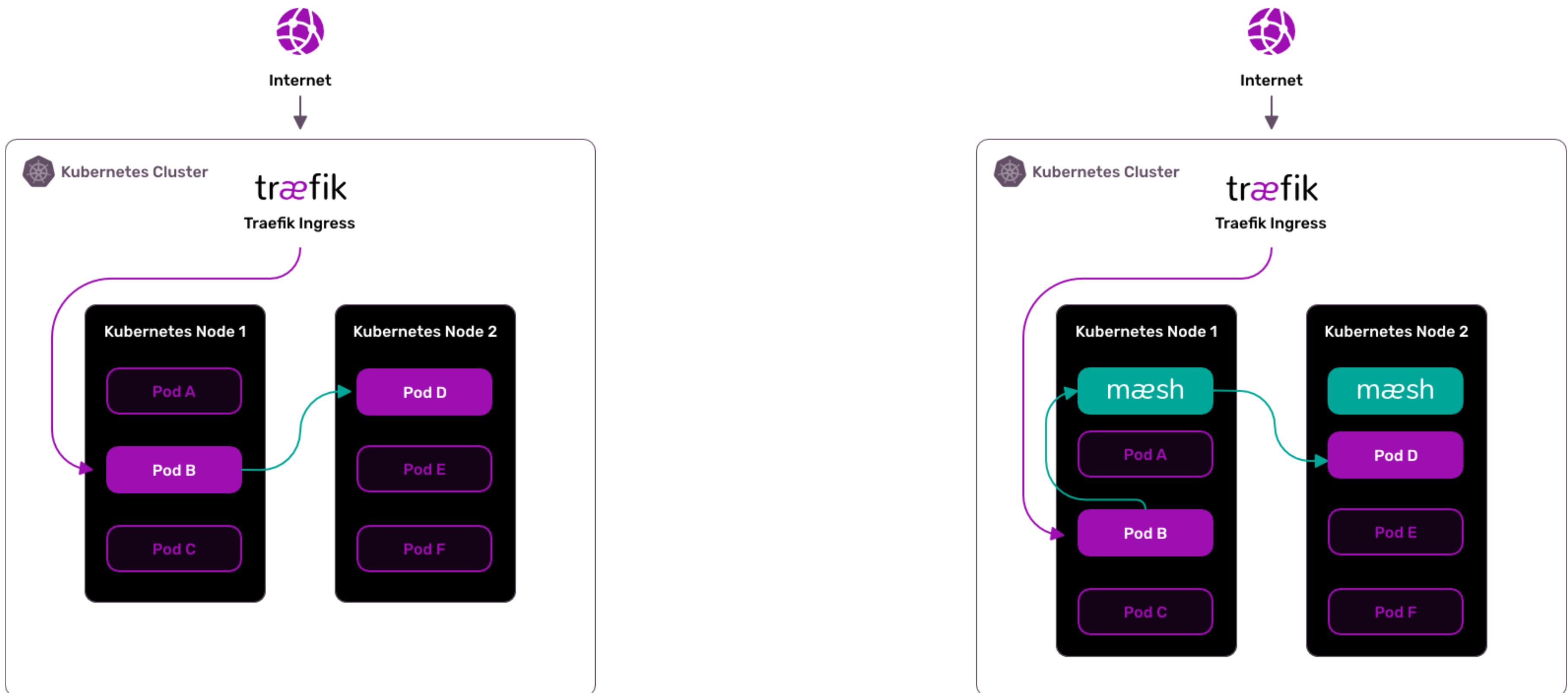
# *More On Maesh*

- Built on top of Traefik,
- SMI (Service Mesh Interface specification) compliant,
- Opt-in by default.

# Features

- OpenTracing / Metrics
- Retries & failovers
- Circuit breakers & Rate limits
- Access controls
- Easy installation

# Maesh Architecture



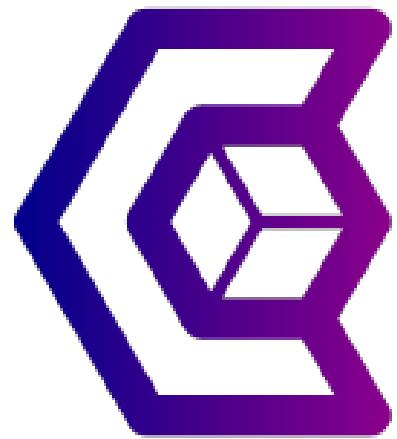
# Demo



That's All Folks!



# We Are Hiring!



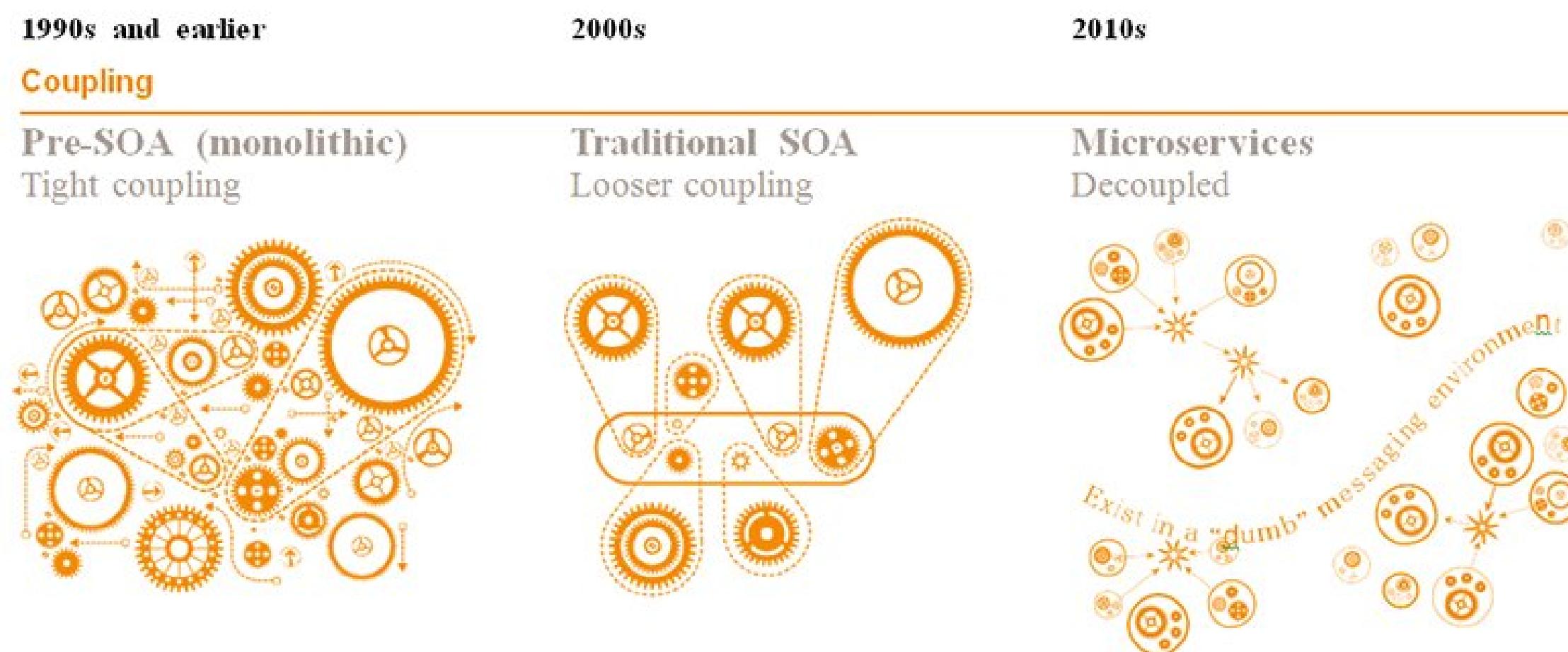
```
docker run -it containous/jobs
```

# Why Traefik?



Why, Mr Anderson?

# Evolution Of Software Design



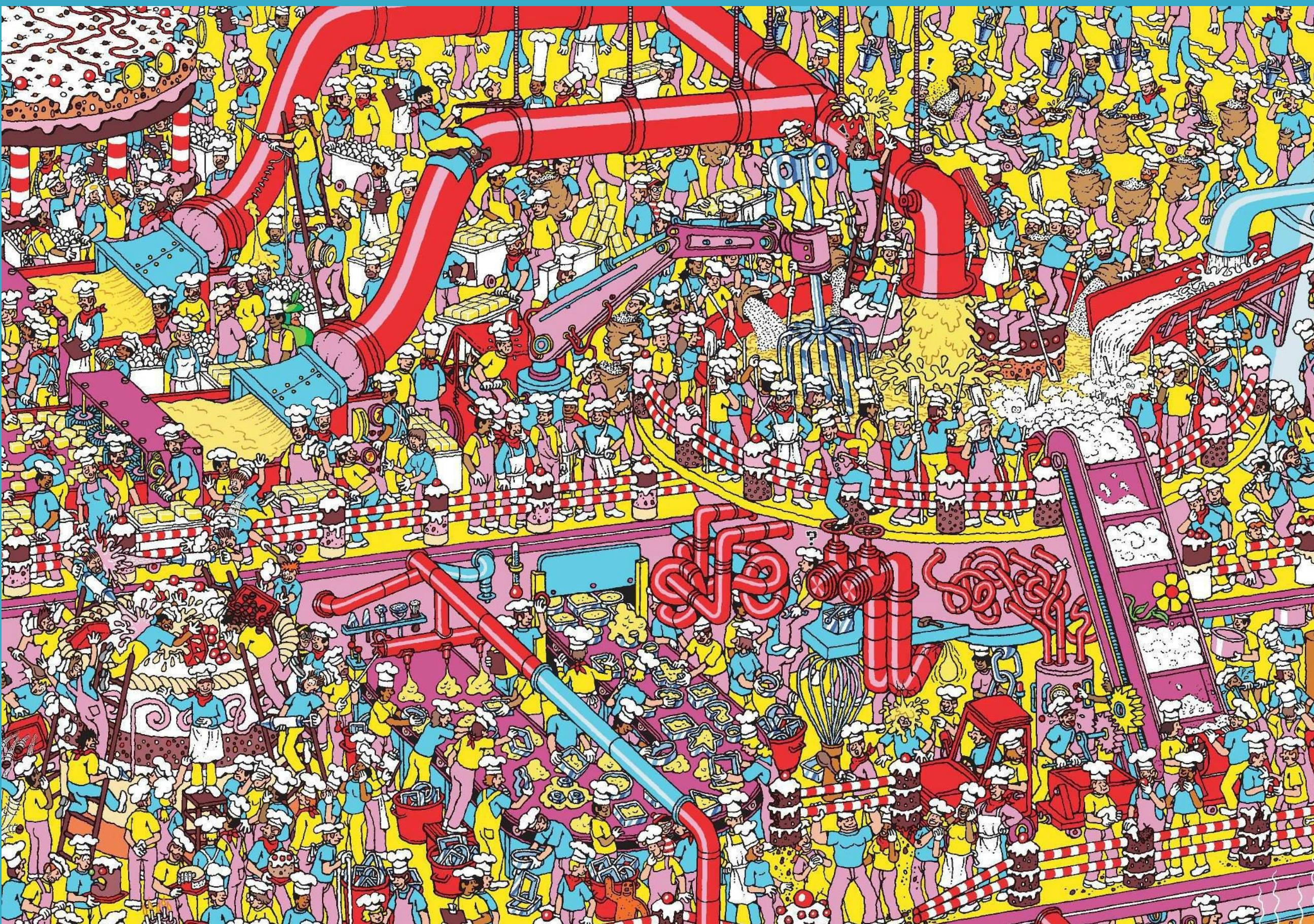
# The Premise Of Microservices...



*...And What Happens*

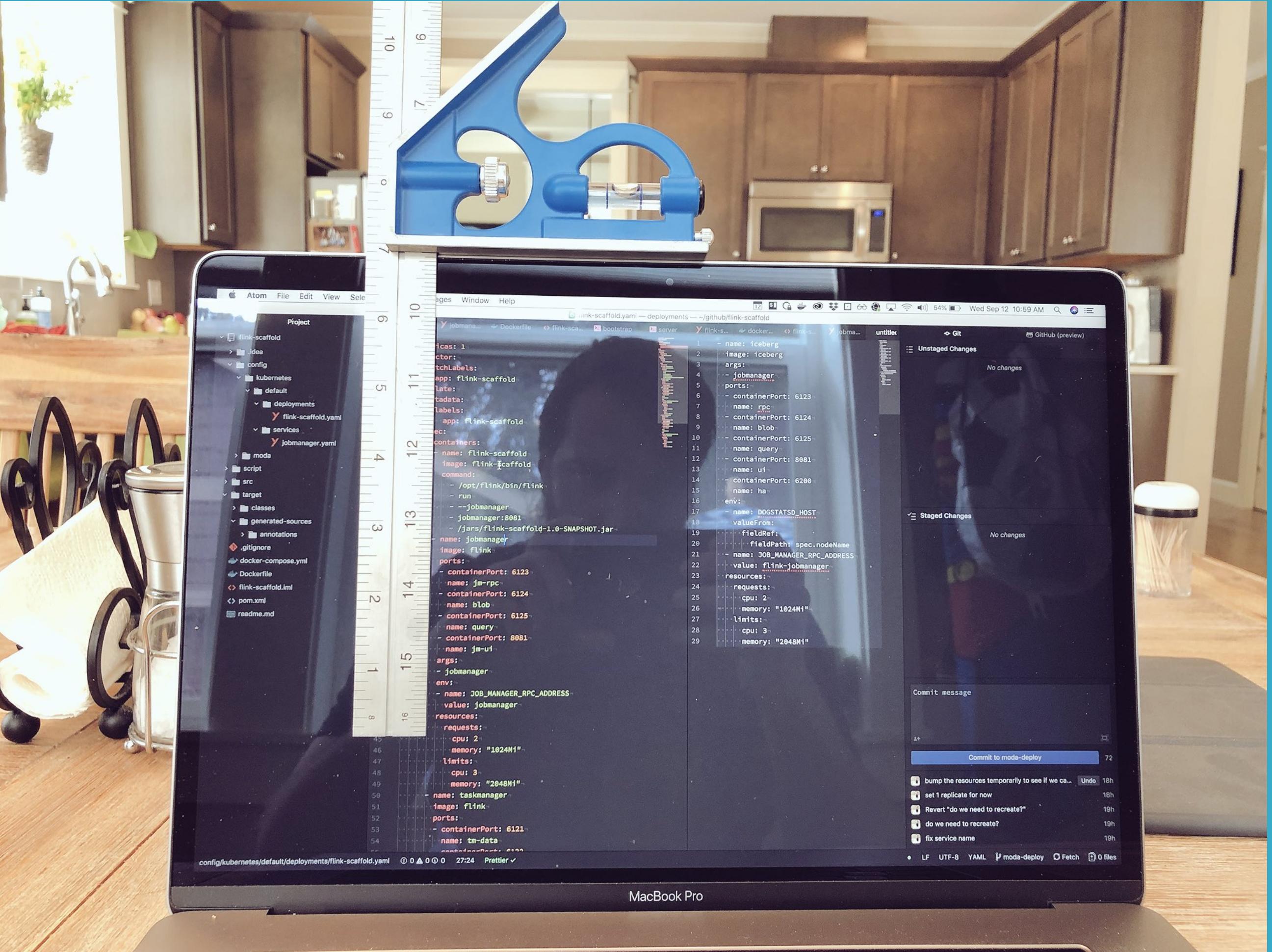


# Where's My Service?



# Tools Of The Trade





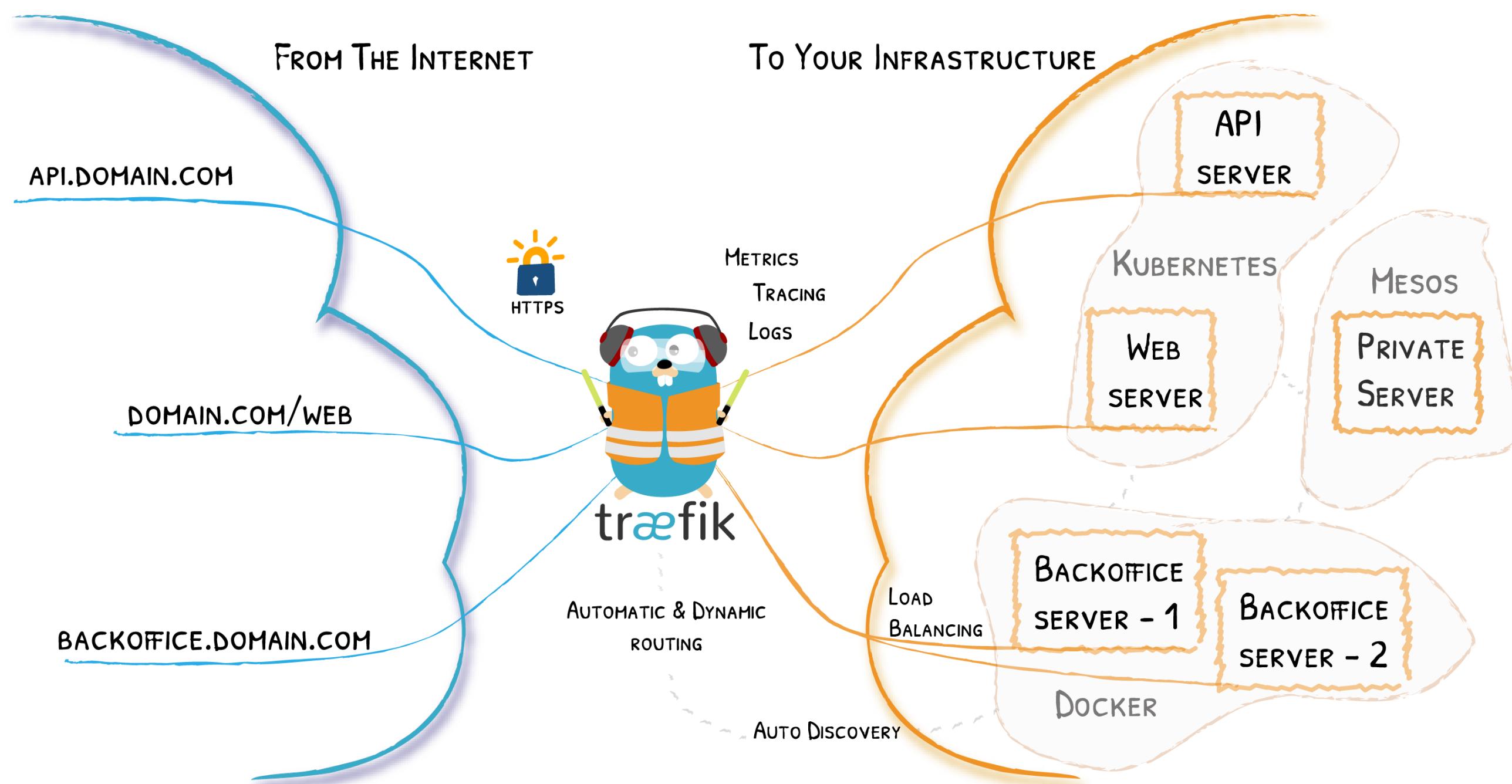
Source: <https://twitter.com/Caged/status/1039937162769096704>

# What If I Told You?



That You Don't Have to Write This Configuration File...?

# Here Comes Traefik!



# Traefik Project

-  <https://github.com/containous/traefik>
- MIT License
- Written in Go
- 24,000+  900M+  350+ 
- Created in 2015
- Current stable branch: v1 . 7

# BACK toTRAEFIK 2.0

Part →

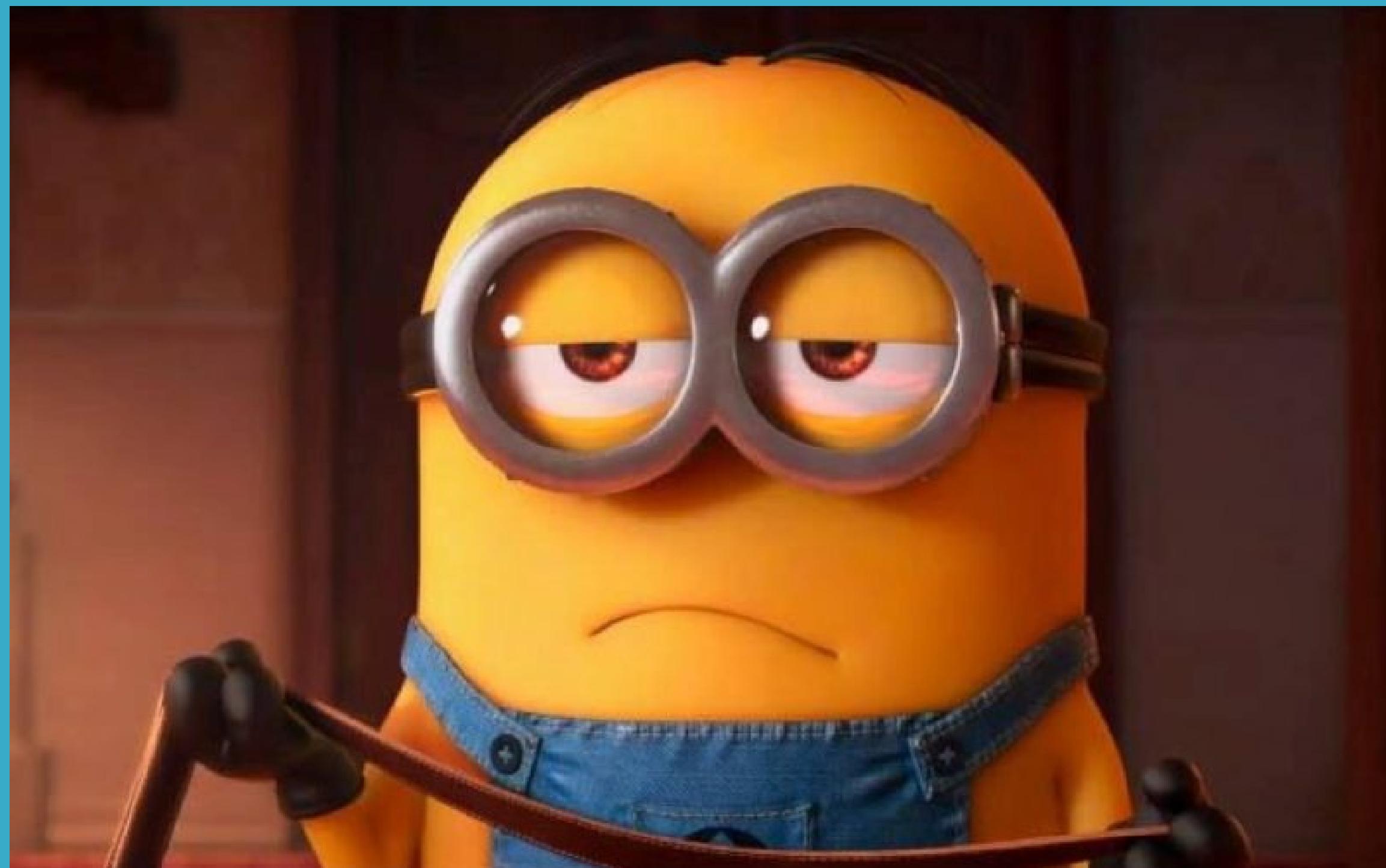


# Traefik 2.0 Quick Overview

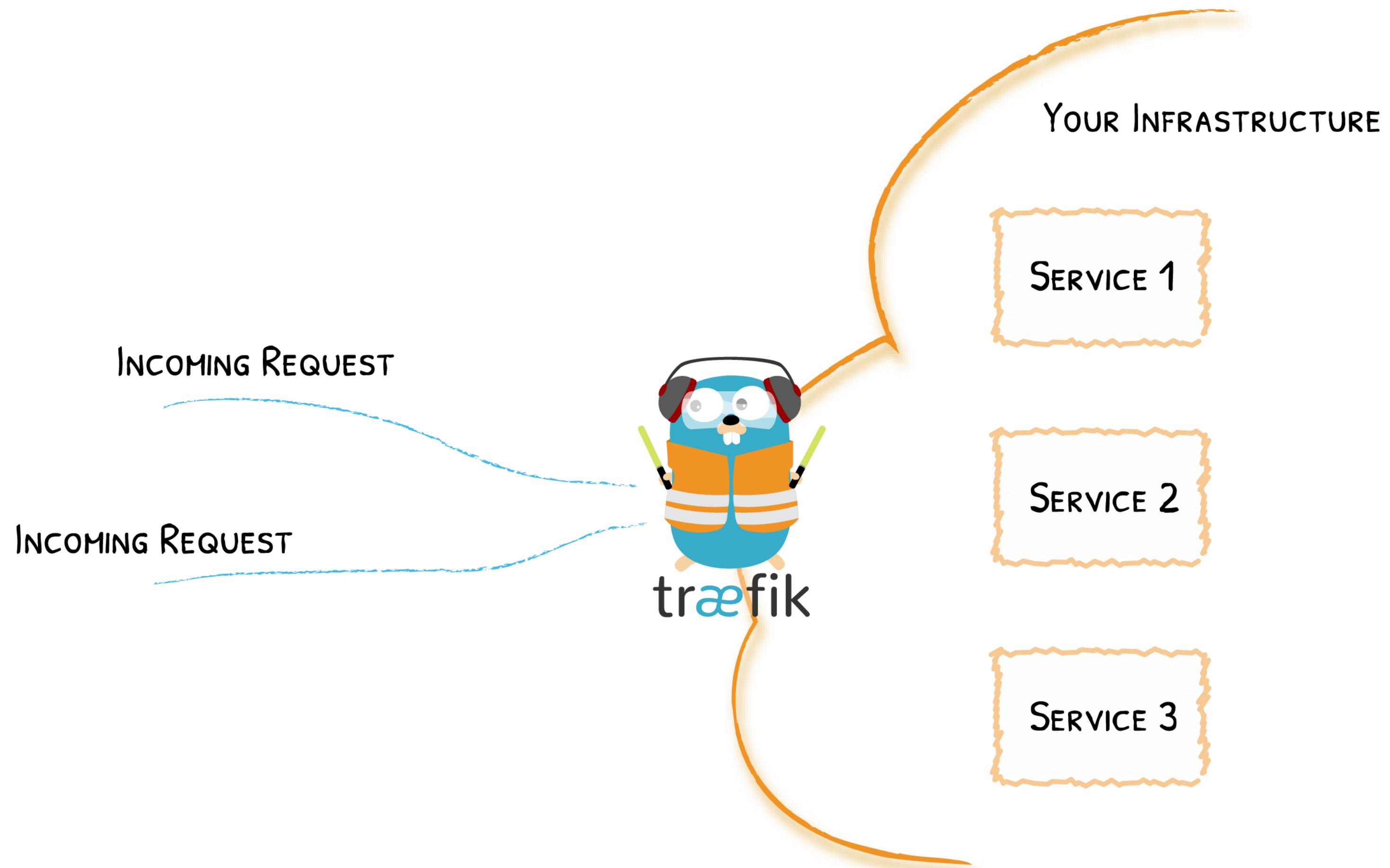
- Revamped Documentation
- Clarified Concepts
- Expressive Routing Rule Syntax
- Middlewares
- TCP Support
- TLS stores & options
- And so Much More...

Learn more on the blog post

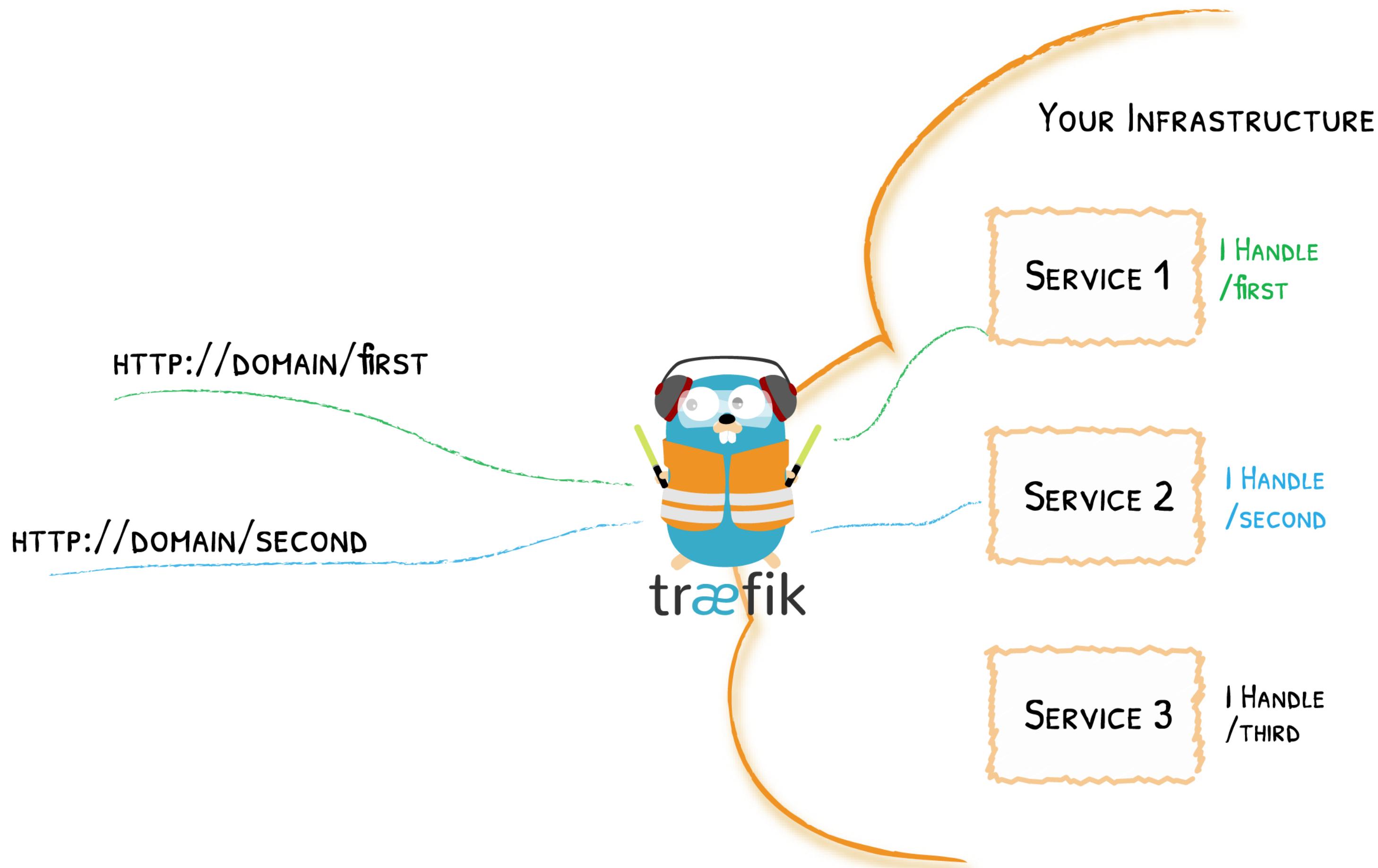
# Traefik (V2.0) Core Concepts



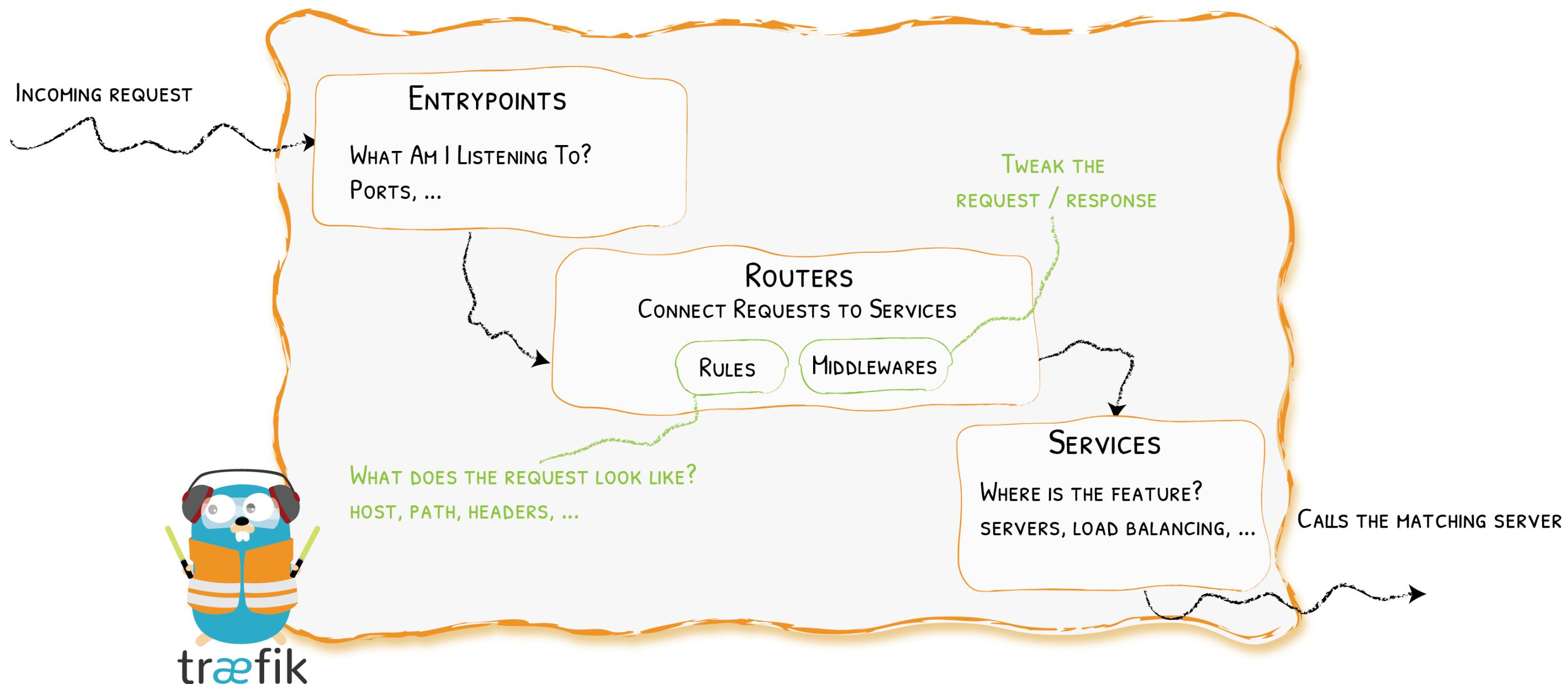
# Traefik Is An Edge Router



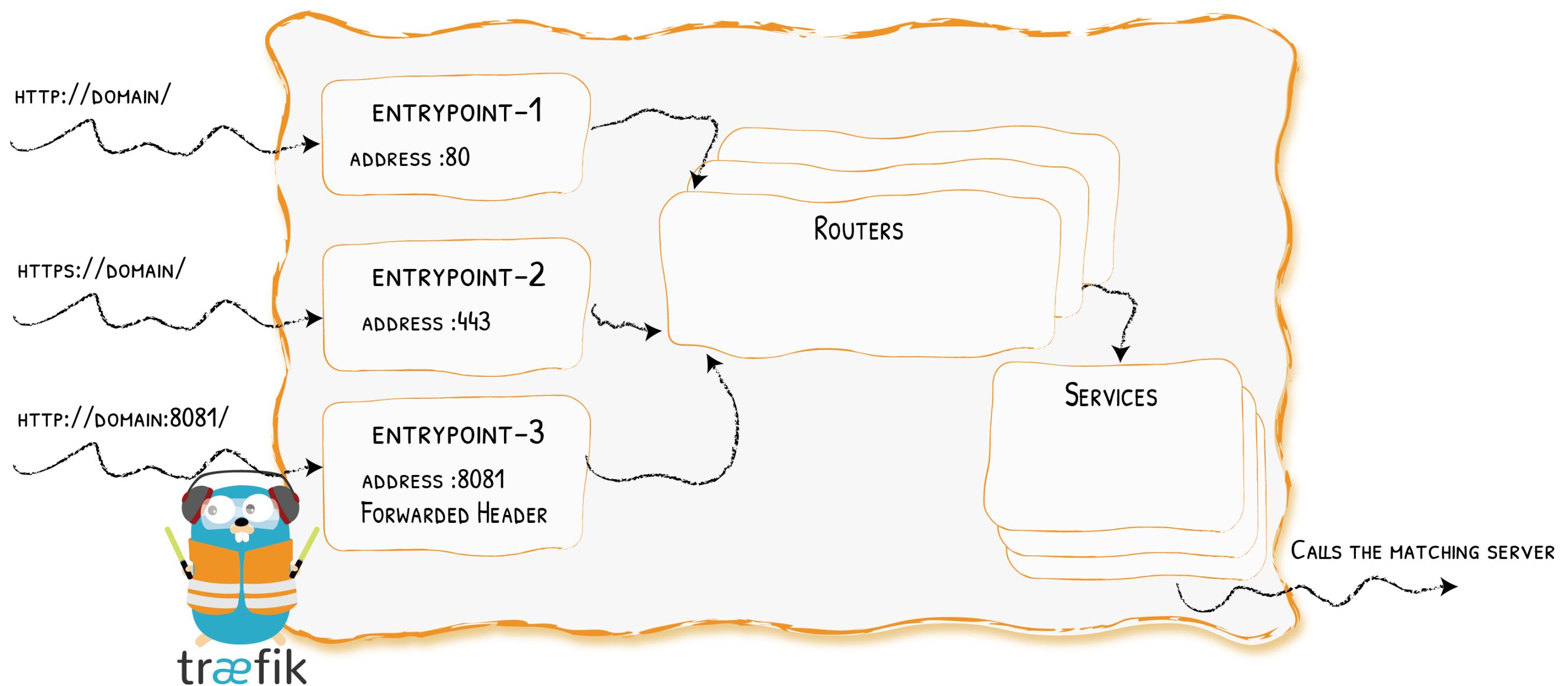
# Traefik Dynamically Discovers Services



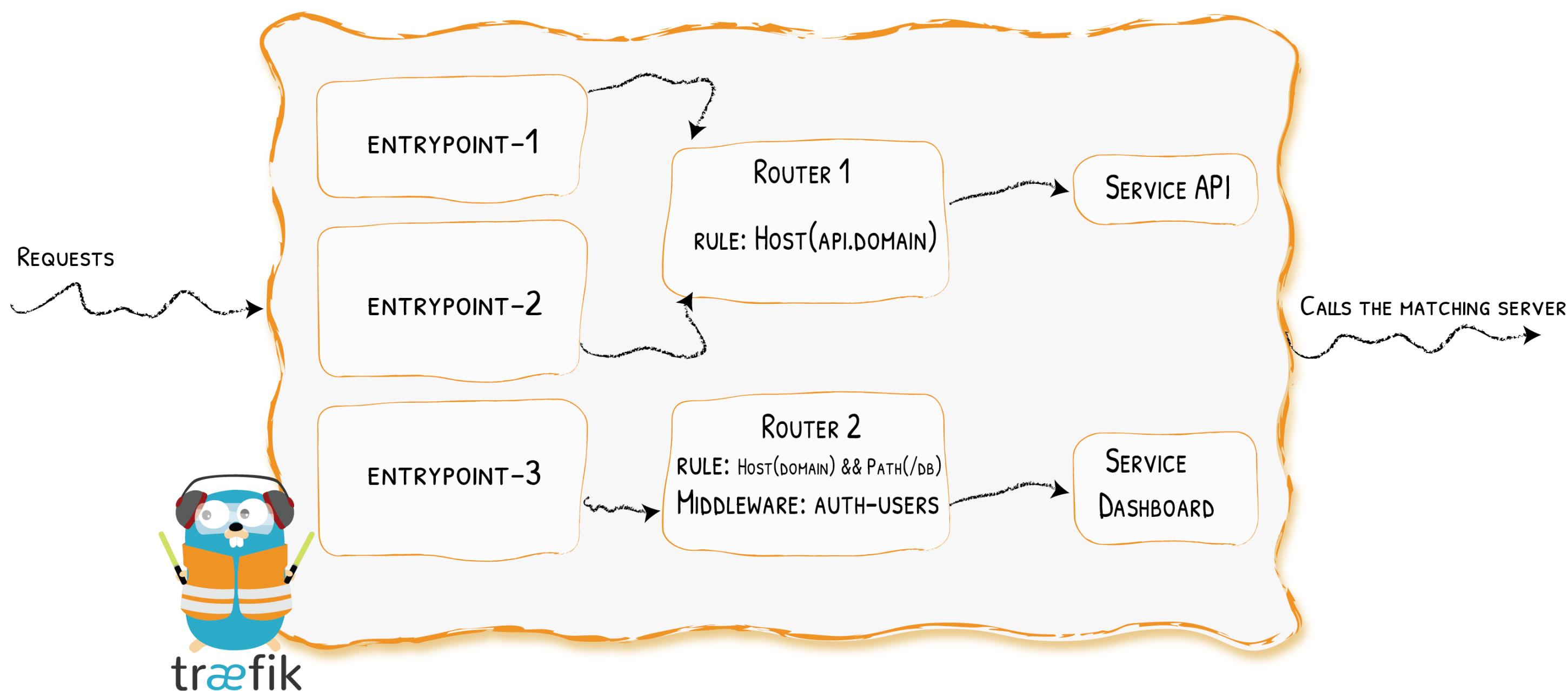
# Architecture (V2.0) At A Glance



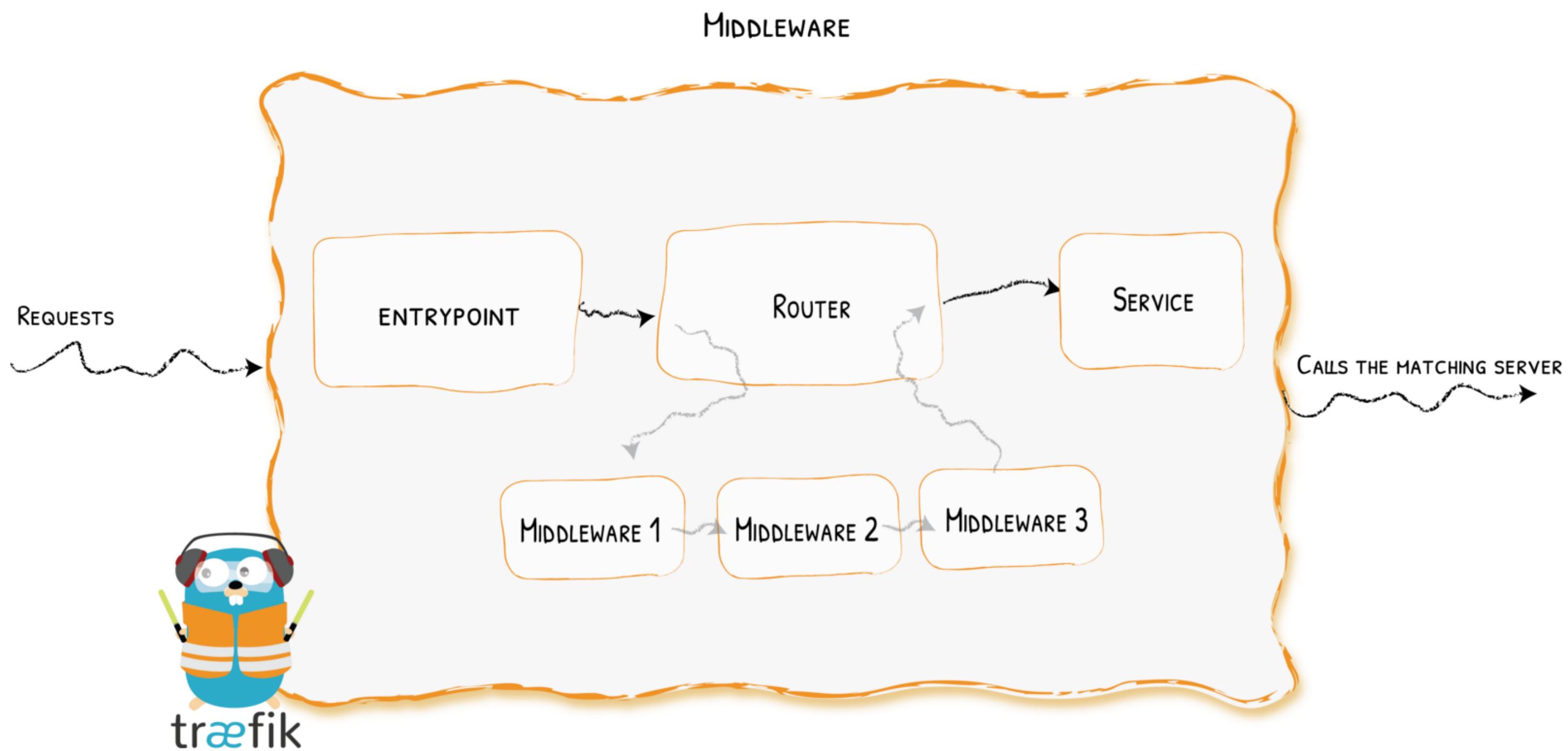
# Entrypoints



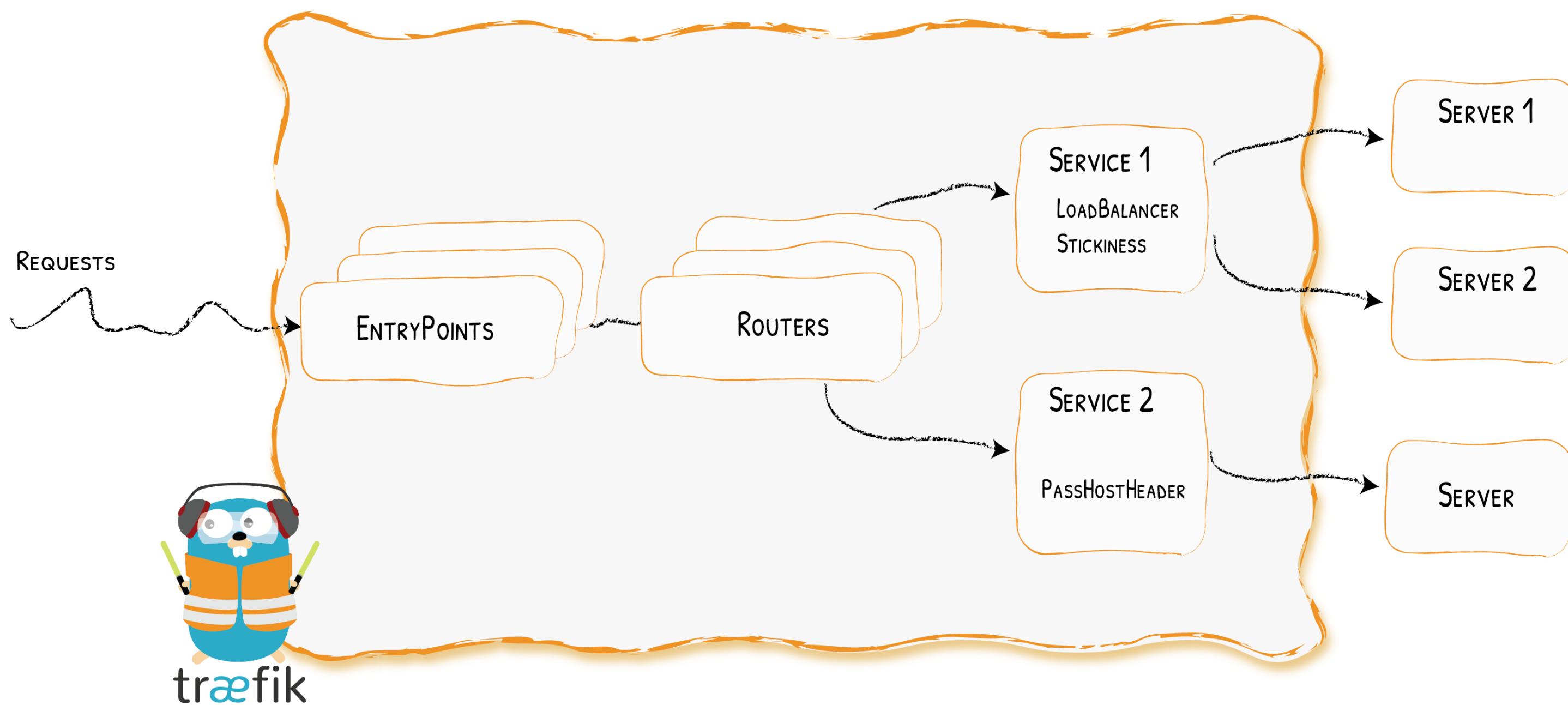
# Routers



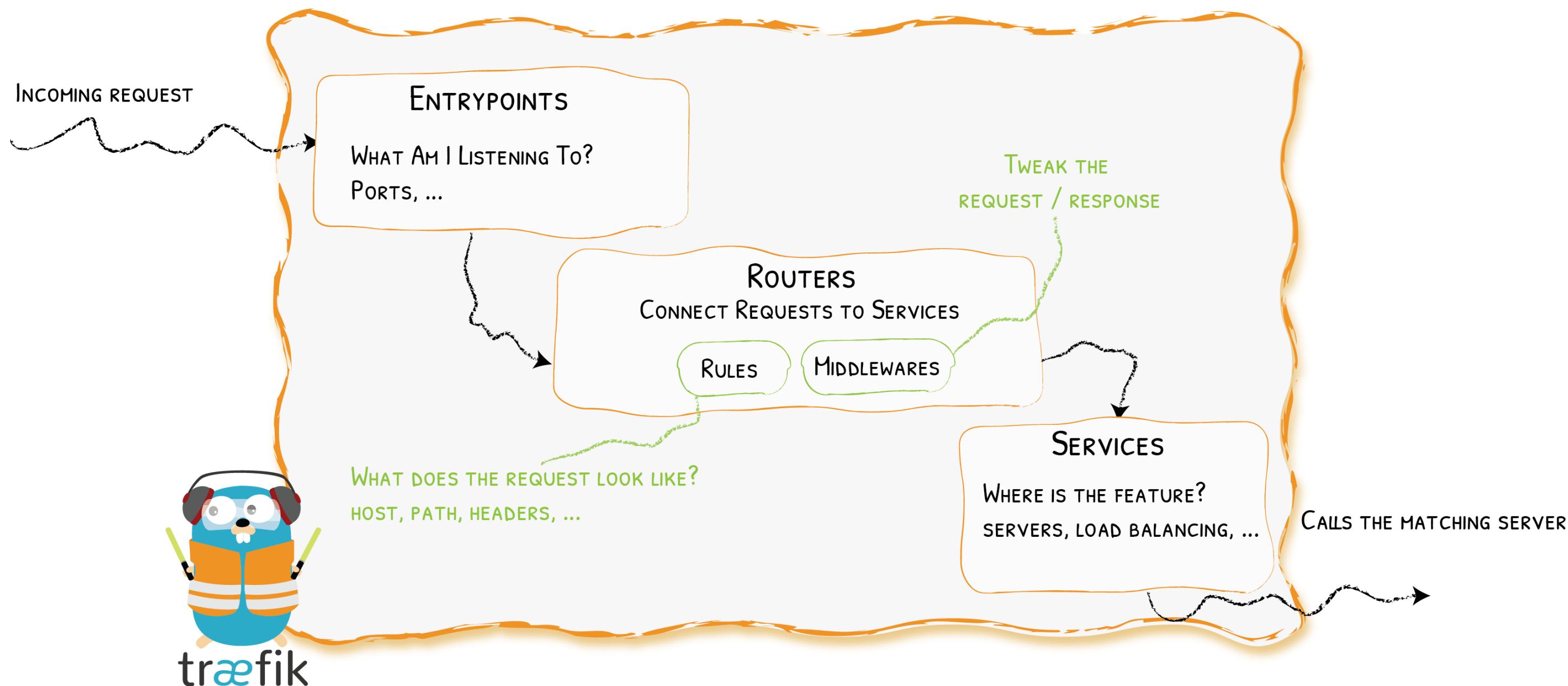
# Middlewares



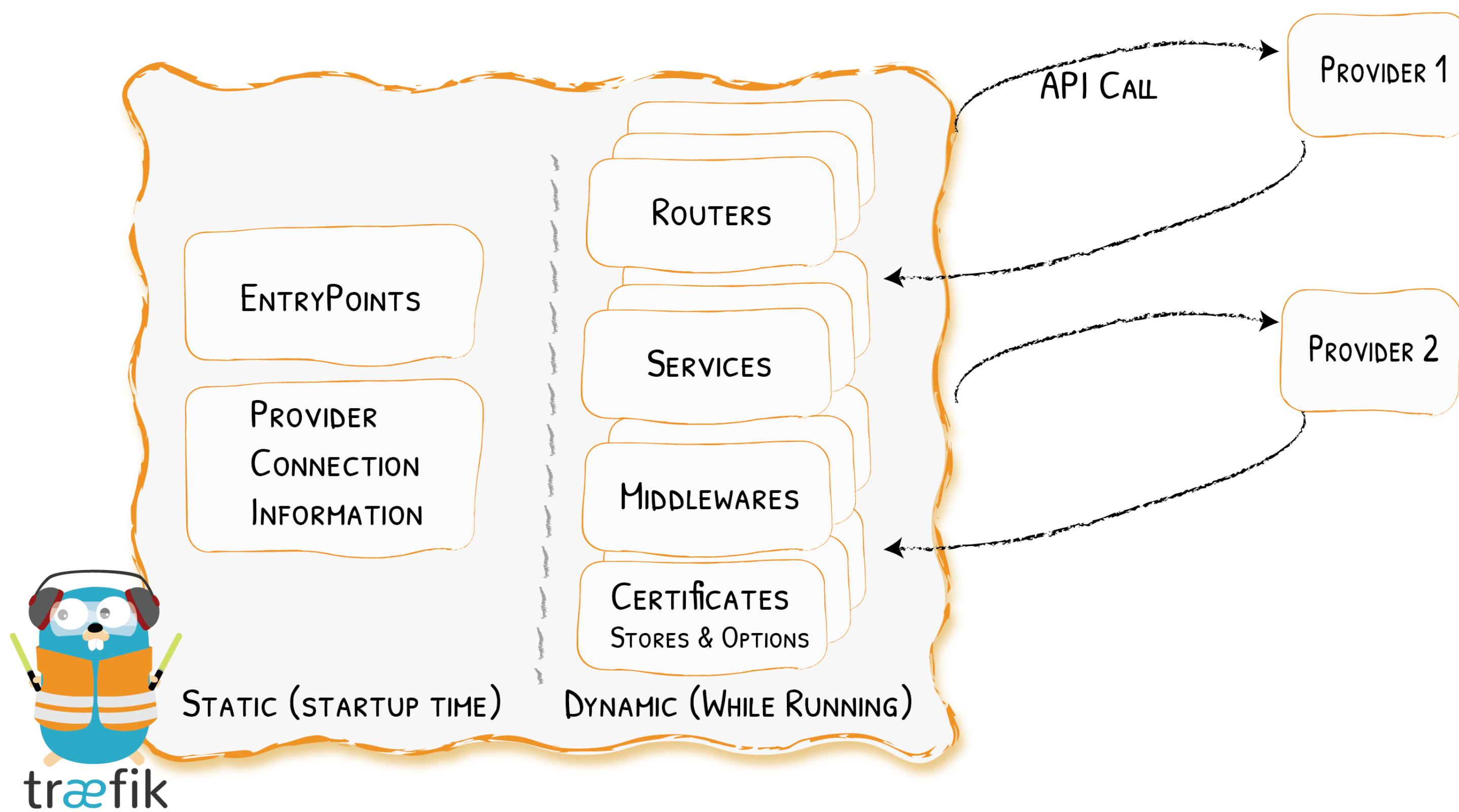
# Services



# Architecture (Again) At A Glance

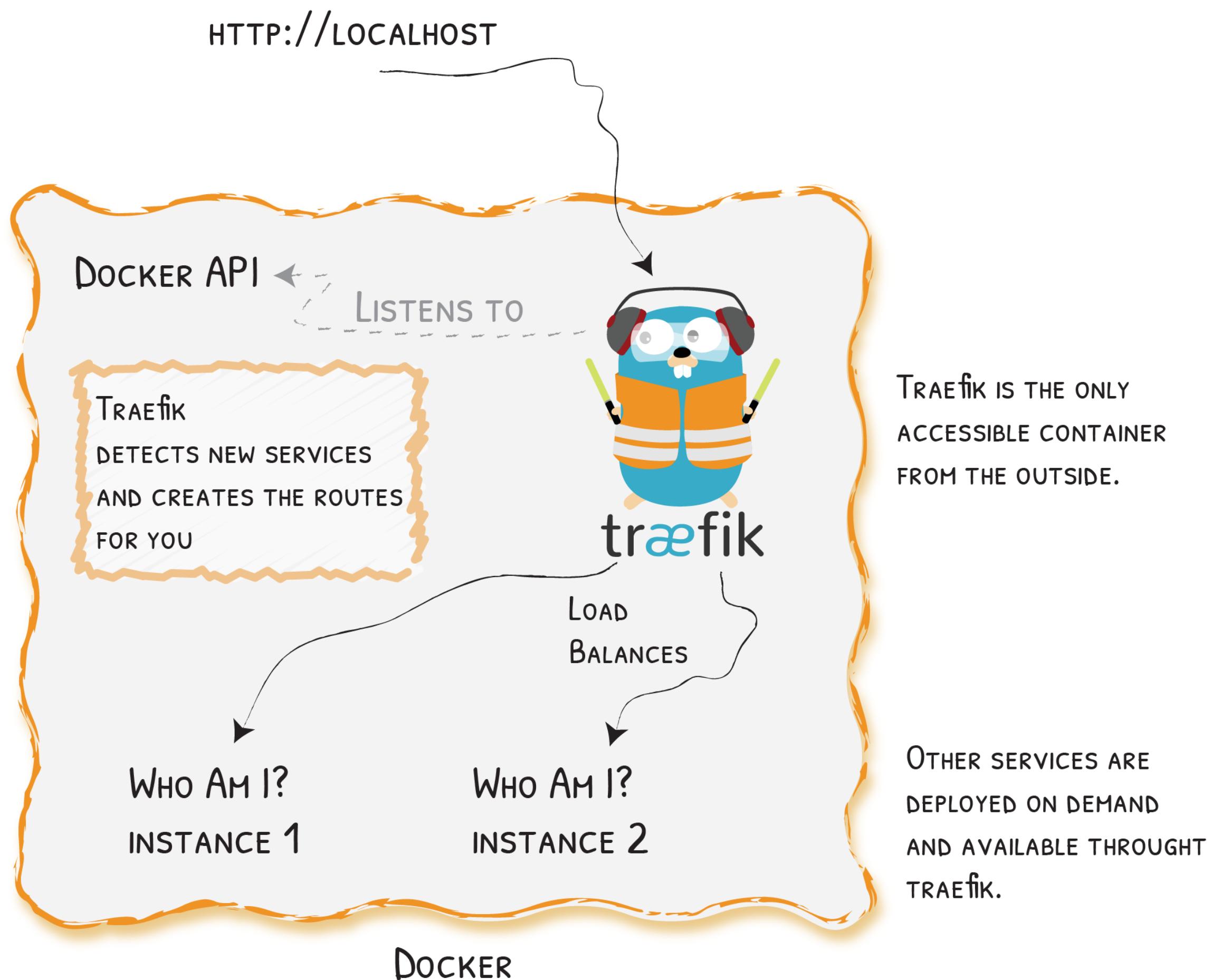


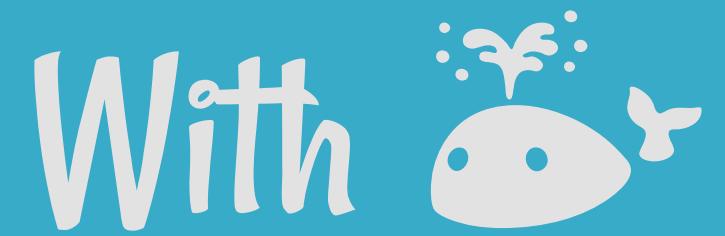
# Static & Dynamic Configuration



Show Me The Configuration!

# Simple Example With





- With Docker Compose:

```
version: '3'

services:
  reverse-proxy:
    image: traefik:v2.0
    command: --providers.docker
    ports:
      - "80:80"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock

  webapp:
    image: containous/whoami
    labels:
      - "traefik.http.routers.webapp.rule=Host(`localhost`)"
```

# With Context

```
# https://mycompany.org/jenkins -> http://jenkins:8080/jenkins
jenkins:
  image: jenkins/jenkins:lts
  environment:
    - JENKINS_OPTS=--prefix=/jenkins
  labels:
    - "traefik.http.services.jenkins.LoadBalancer.server.Port=8080" # Because 50000 is also exposed
    - "traefik.http.routers.jenkins.rule=Host(`mycompany.org`) && PathPrefix(`/jenkins`)"
    - "traefik.http.routers.jenkins.service=jenkins"
```

# With 🐠: Rewrites

```
# https://mycompany.org/gitserver -> http://gitserver:3000/
gitserver:
  image: gitea/gitea
  labels:
    - "traefik.http.routers.gitserver.rule=Host(`mycompany.org`) && PathPrefix(`/gitserv`)"
    - "traefik.http.middlewares.gitserver-stripPrefix.stripPrefix.prefixes=/gitserv"
    - "traefik.http.routers.gitserver.middlewares=gitserver-stripPrefix"
```

# With 🐳: Websockets

```
# https://webterminal.mycompany.org -> http://webterminal/
webterminal:
  image: ts10922/ttyd
  labels:
    - "traefik.http.routers.devbox.rule=Host(`webterminal.mycompany.org`)"
```

# Traefik With ⚓

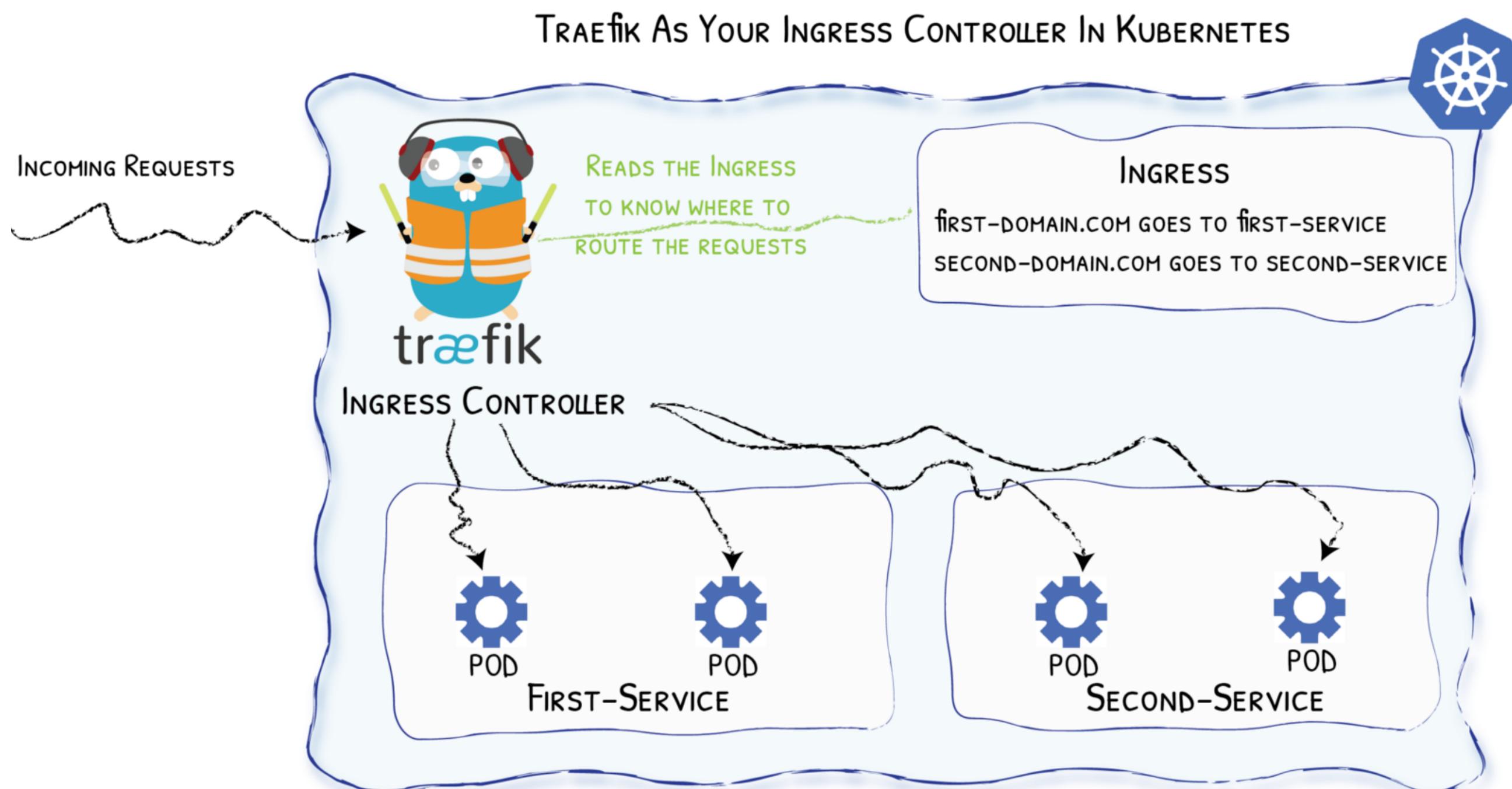


Diagram from <https://medium.com/@geraldcroes>

# Example Code With ⚓

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: 'traefik'
spec:
  rules:
  - host: localhost
    http:
      paths:
      - path: "/whoami"
        backend:
          serviceName: webapp
          servicePort: 80
```

# ✳️ CRD - Custom Resources Definition

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

```
$ kubectl apply -f webapp.yaml
$ kubectl get ingressroute
```

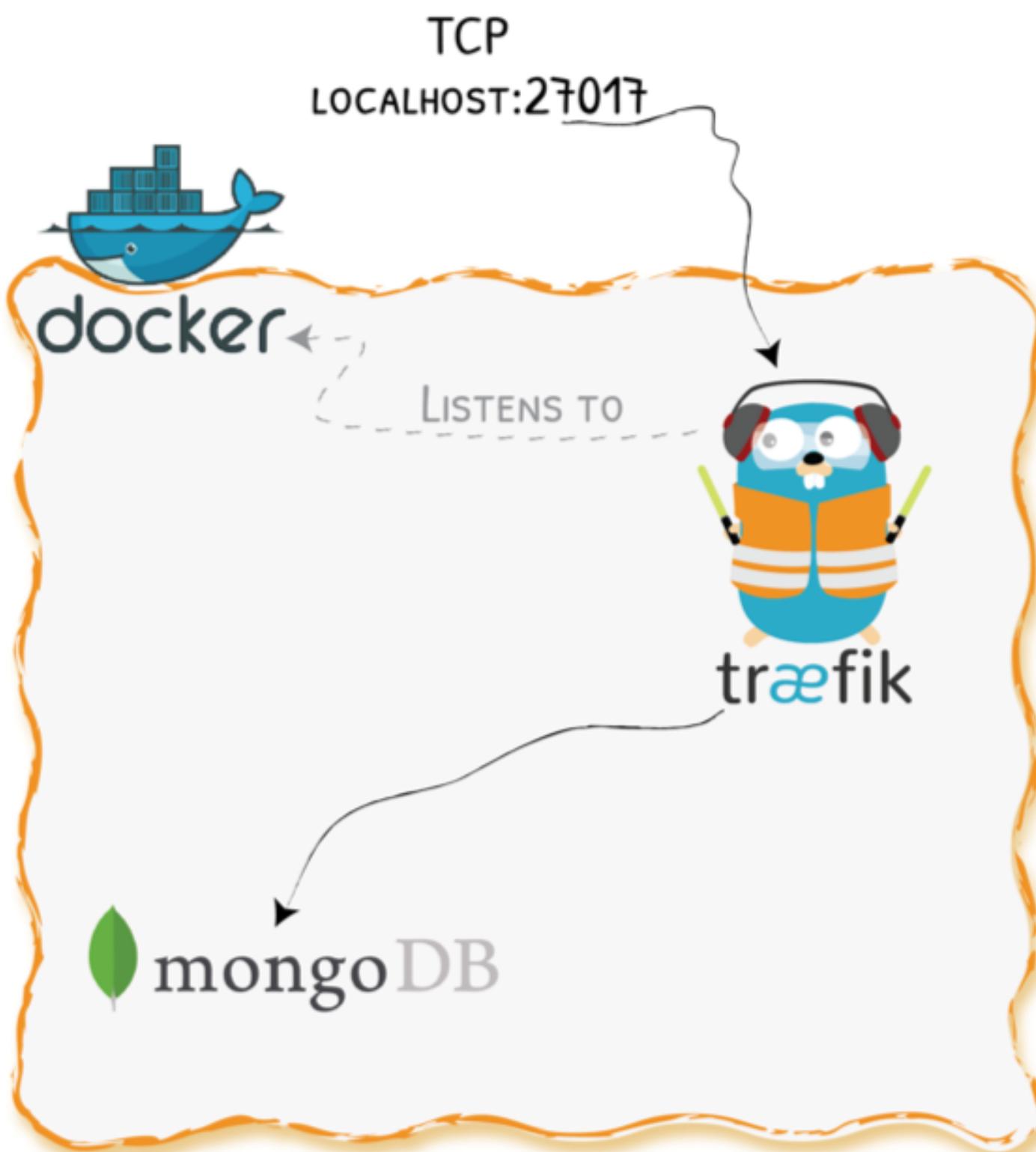
# 🌐 & TCP (With CRD)

```
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
```



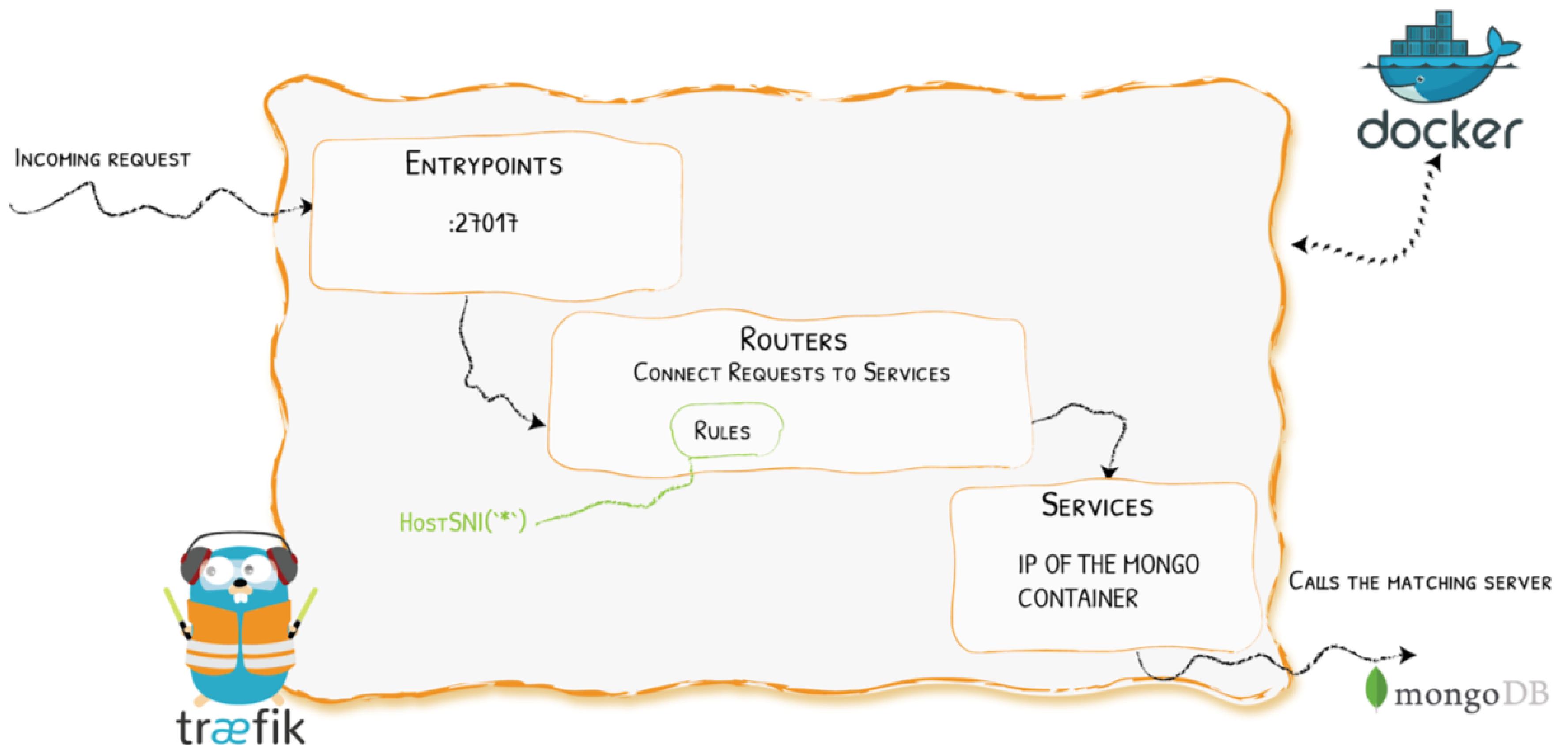
HTTP  
&  
TCP

# Demo 1 - Straightforward TCP Routing

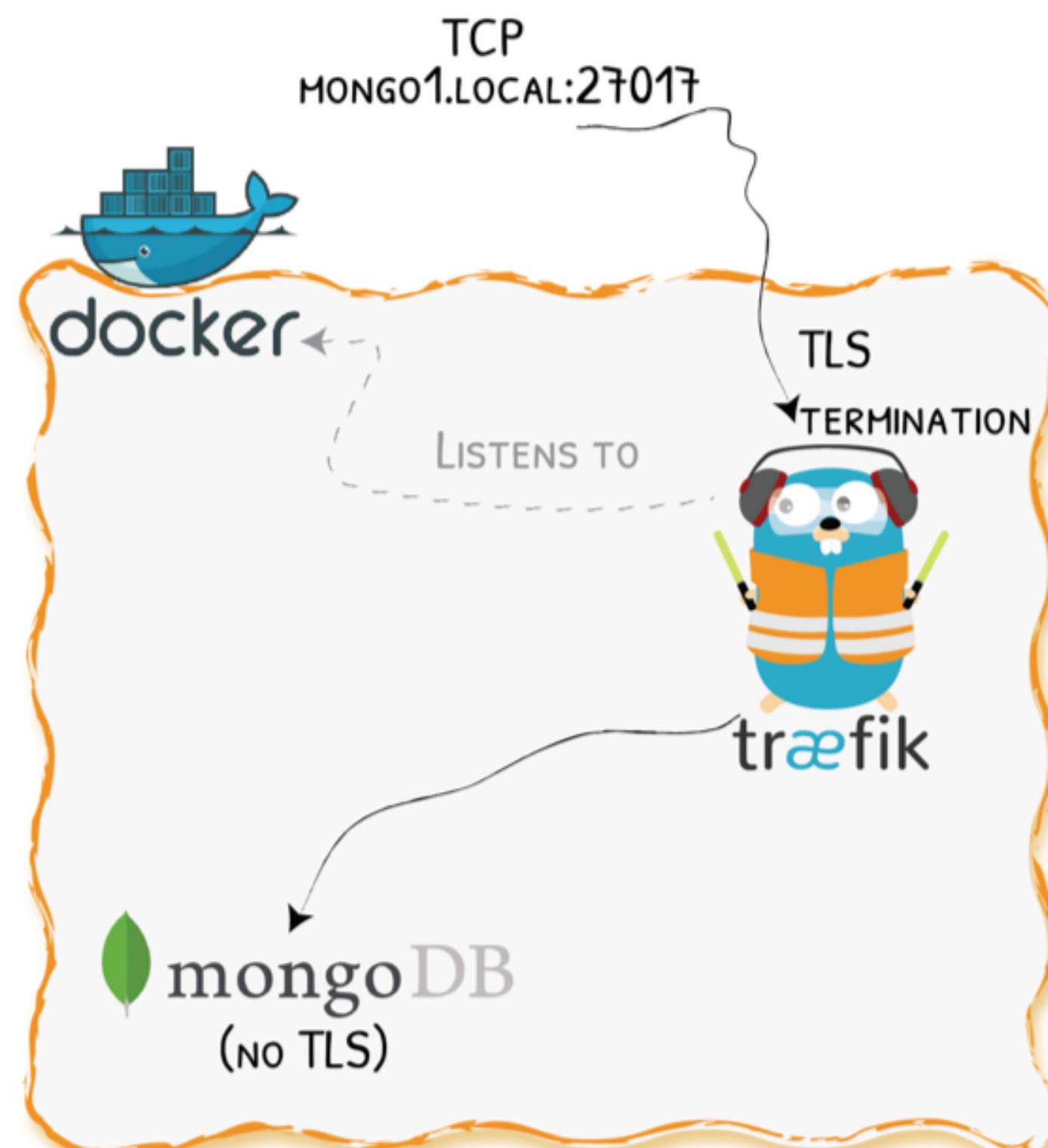


Demo Code on [GitHub](#)

# Demo 1 - Configuration

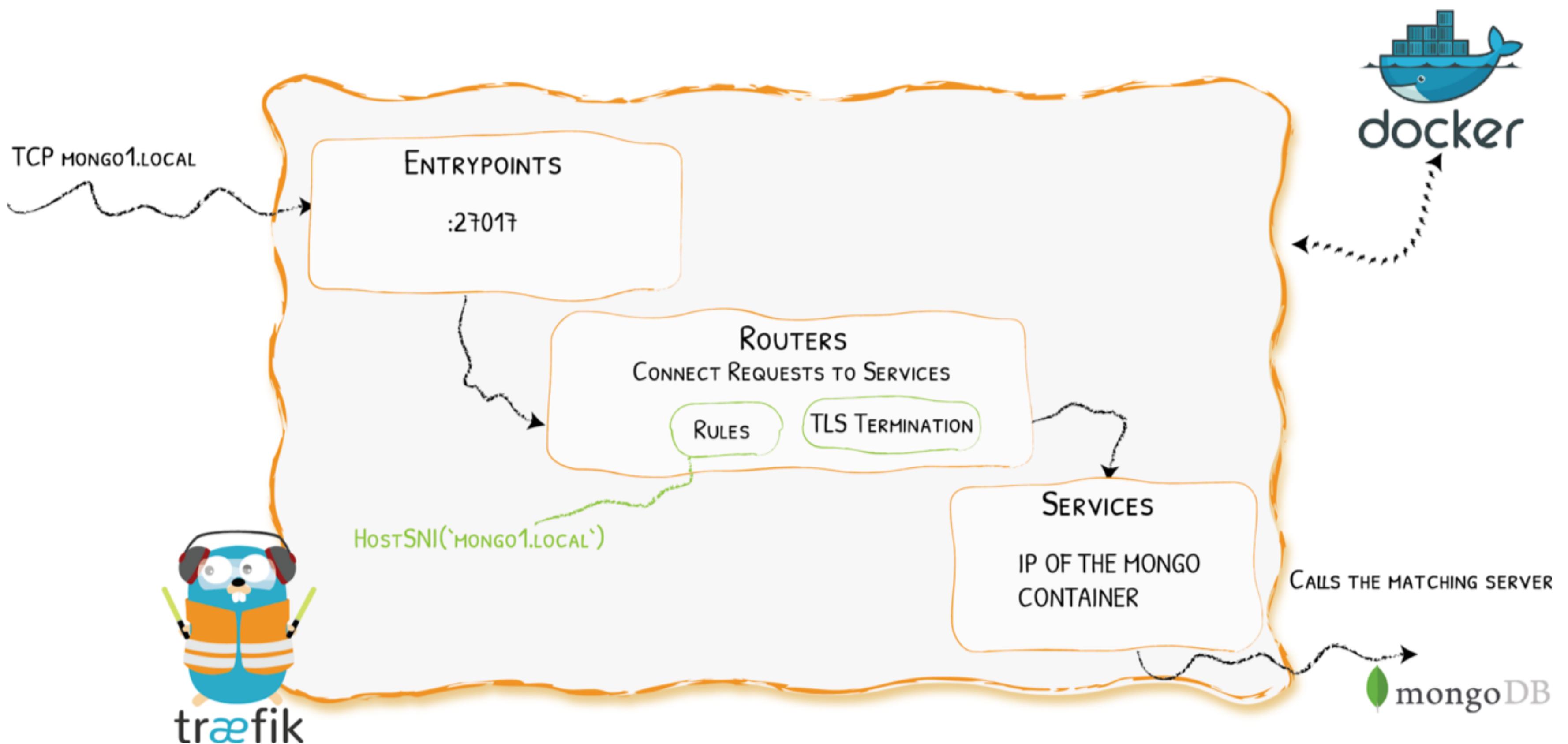


# Demo 2 - Let's Add TLS To TCP With Traefik

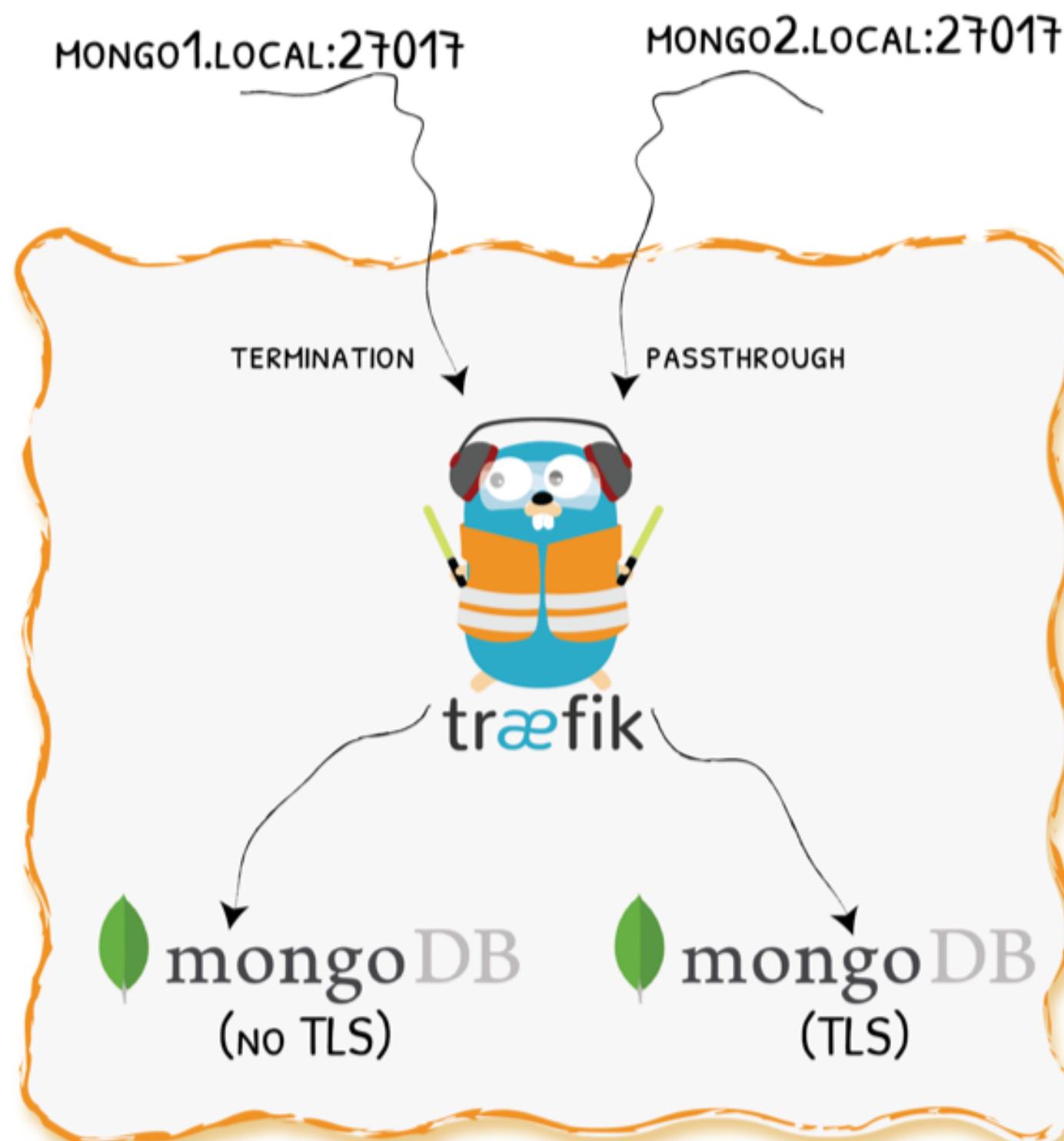


Demo Code on [GitHub](#)

# Demo 2 - Configuration

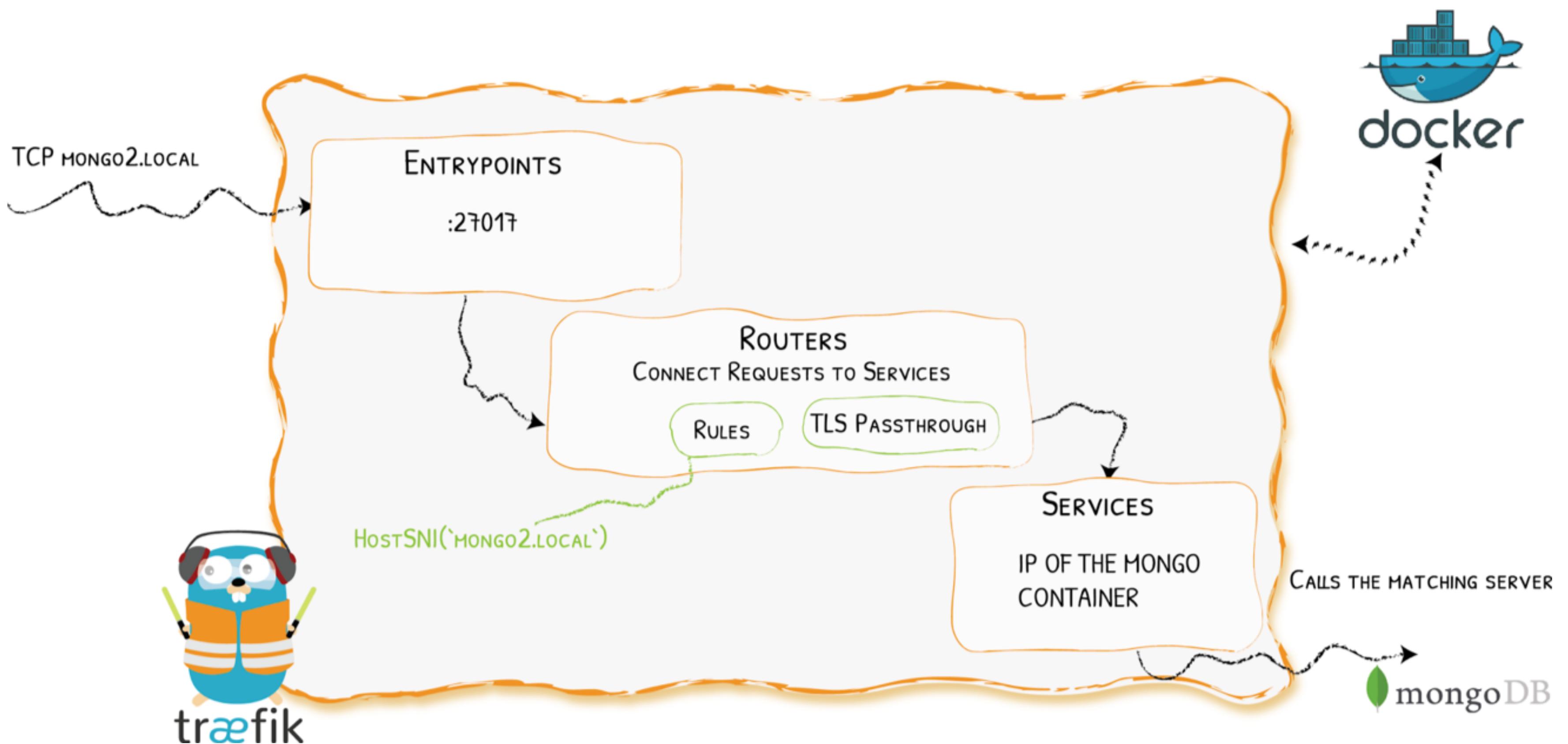


# Demo 3 - SNI Routing + TLS Passthrough

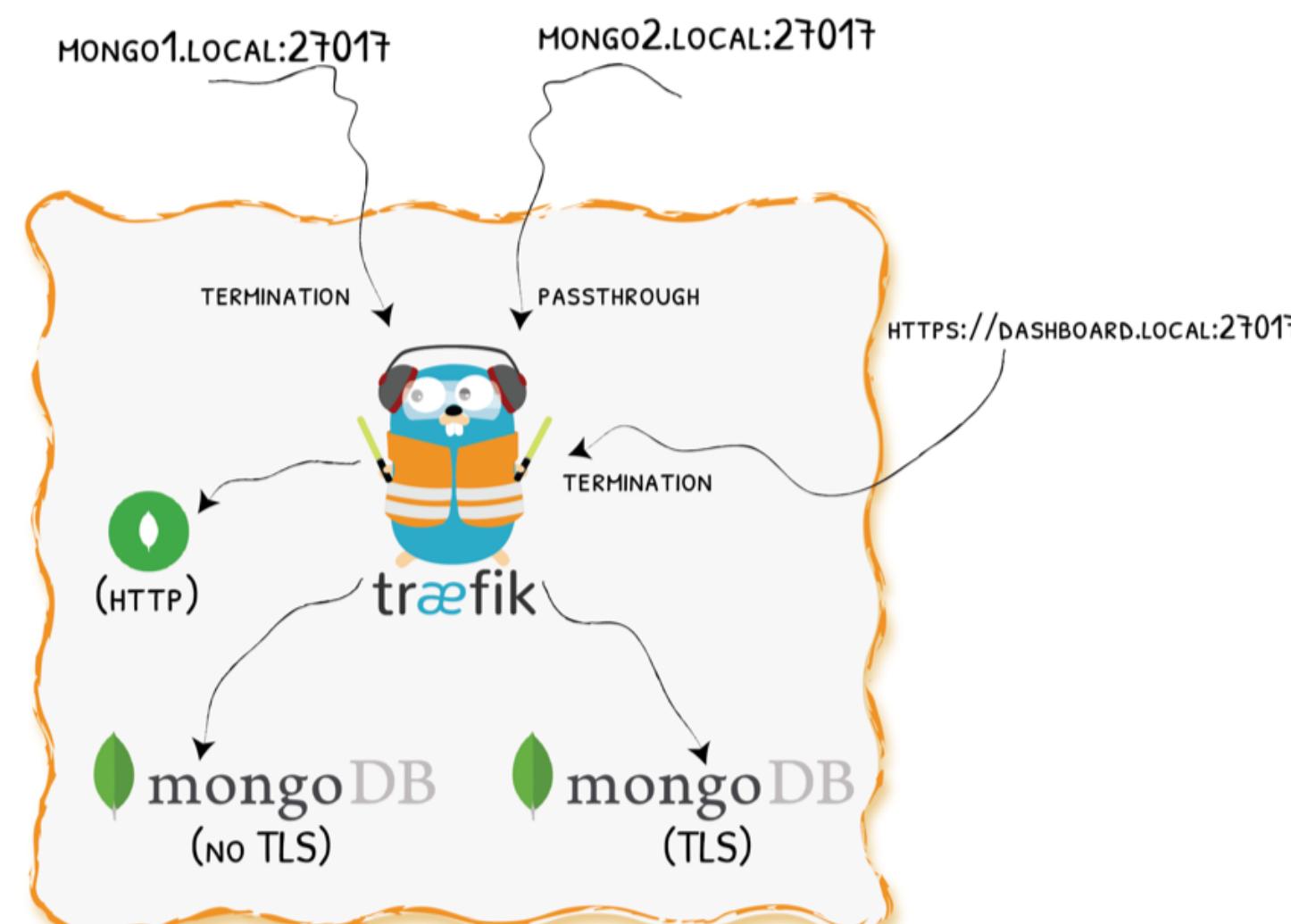


Demo Code on [GitHub](#)

# Demo 3 - Configuration



# Demo 4 - Muxing HTTPS And TCP On The Same Port



Demo Code on [Gist](#)

# We Also Missed Talking About...

A cloud of various technology terms and concepts, including:

- MESOS
- ZIPKIN
- LIMITING
- KUBERNETES
- Dynamic Metrics
- HTTP ERROR
- CERTIFICATE
- TLS Reverse-Proxy
- HEADERS
- GRPC
- DYNAMIC/WILDCARD
- Security Configurations
- Tracing PROXY
- SECRETS
- PROMETHEUS
- JAEGER
- WEBSOCKETS
- SSL
- FORWARDING
- REDIRECTS
- DOCKER
- PROTOCOL
- CHECKS
- CLUSTER AUTH
- HSTS
- RATE
- CONSUL
- SWARM MODE
- SWARM
- MODE

## *More With V2.0*

- New WebUI
- New metrics
- YAML (or TOML or CLI)
- Advanced Load-Balancing: WRR, Canary, Mirroring
- (Not done yet) UDP

# More Info

[bit.ly/traefik-v2](https://bit.ly/traefik-v2)

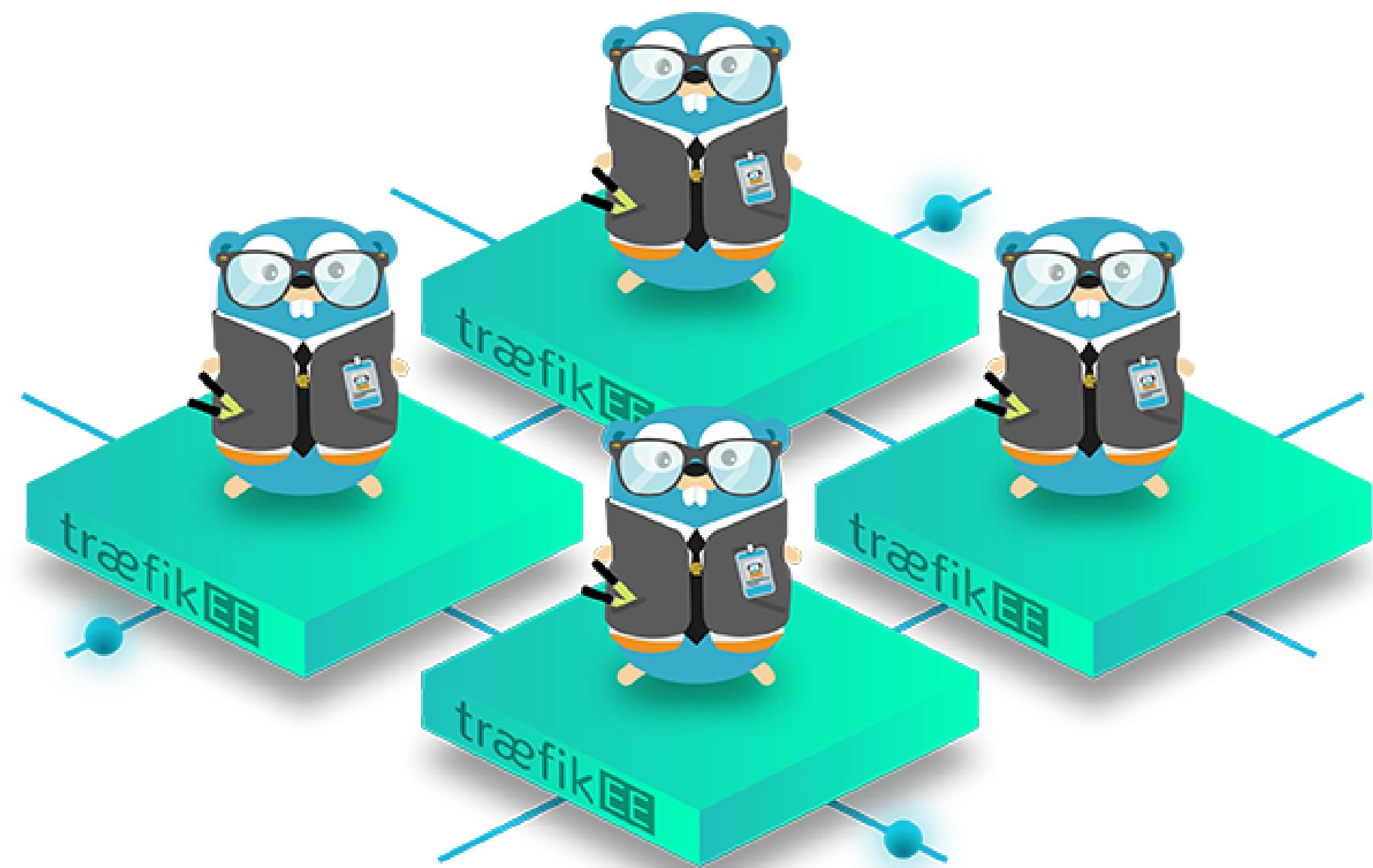
**TO BE  
CONTINUED...**

# The Herd



You came to the wrong neighbour

# Traefik Comes In Herd



High Availability

# HIGH AVAILABILITY

traefik ENTERPRISE EDITION

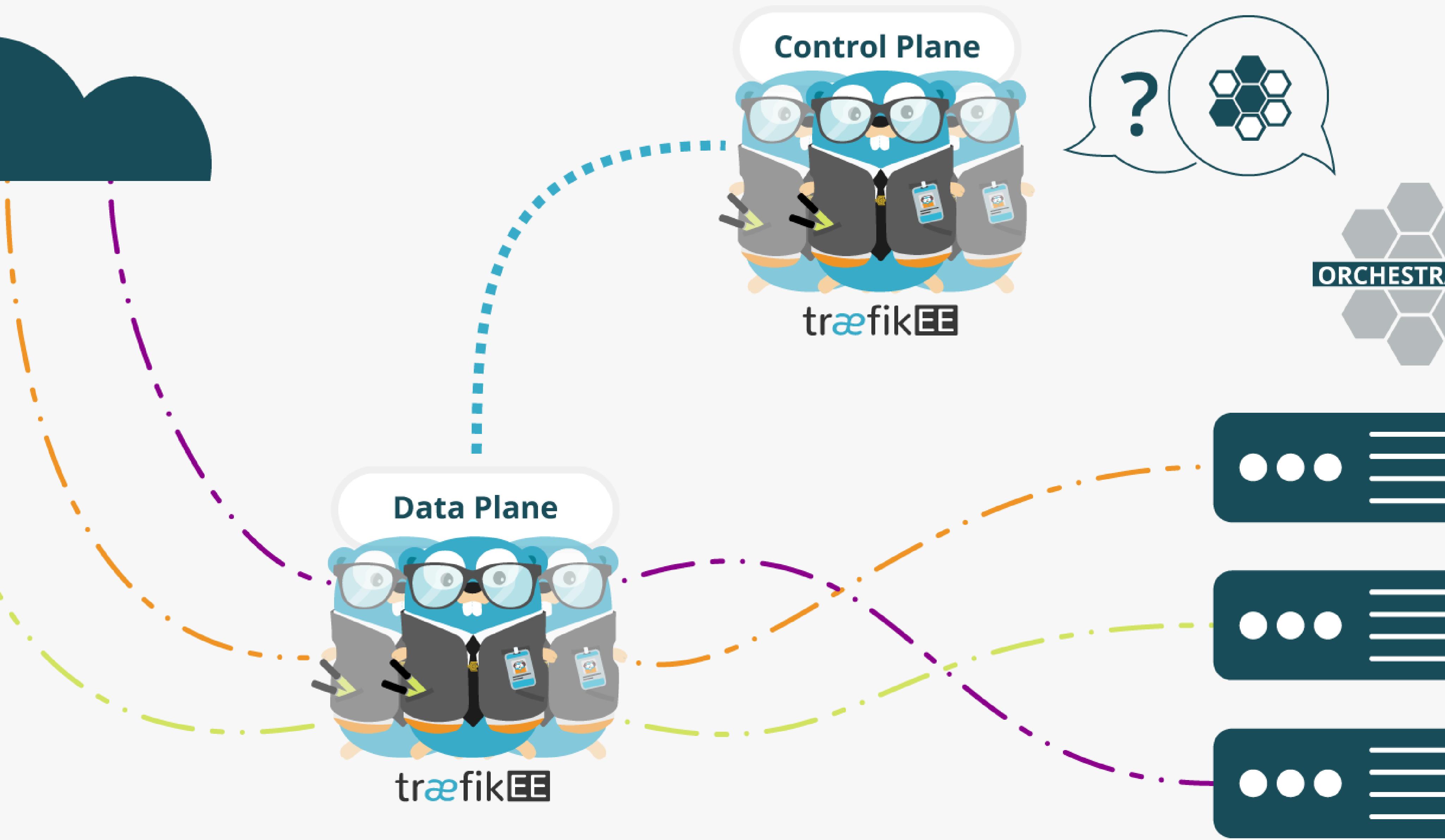
# SECURITY

traefik ENTERPRISE EDITION

Scalability

# SCALABILITY

traefik ENTERPRISE EDITION



# As Simple As Traefik

- Install it:

```
# Cluster Installation
traefikeectl install \
--licensekey="SuperSecretLicence" \
--dashboard \
--kubernetes # Or --swarm
```

- Configure it:

```
# Routing Configuration, same as Traefik's
traefikeectl deploy \
--acme.email=ssl-admin@mycompany.org
--acme.tlsChallenge
...
```

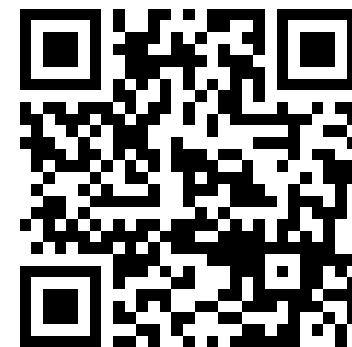
# Free Trial

<https://containo.us/traefikee>

# Thank You!

 @DamienDuportal

 dduportal



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