

# Traefik V2.0 & Maesh: Look Ma', Routing HTTP And TCP With No Hands!



Belgium Kubernetes Meetup - October 2019 @ Louvain

# 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

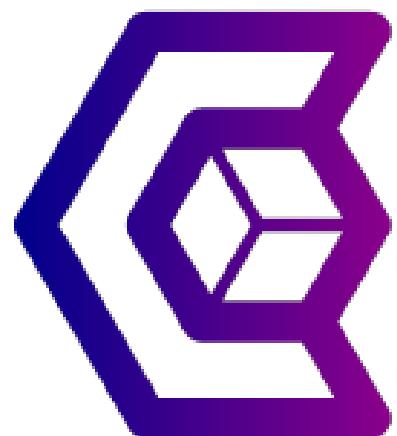
- Damien DUPORTAL:
  - Træfik's Developer  Advocate @ Containous
  -  @DamienDuportal
  -  dduortal



# Containous

<https://containo.us>

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

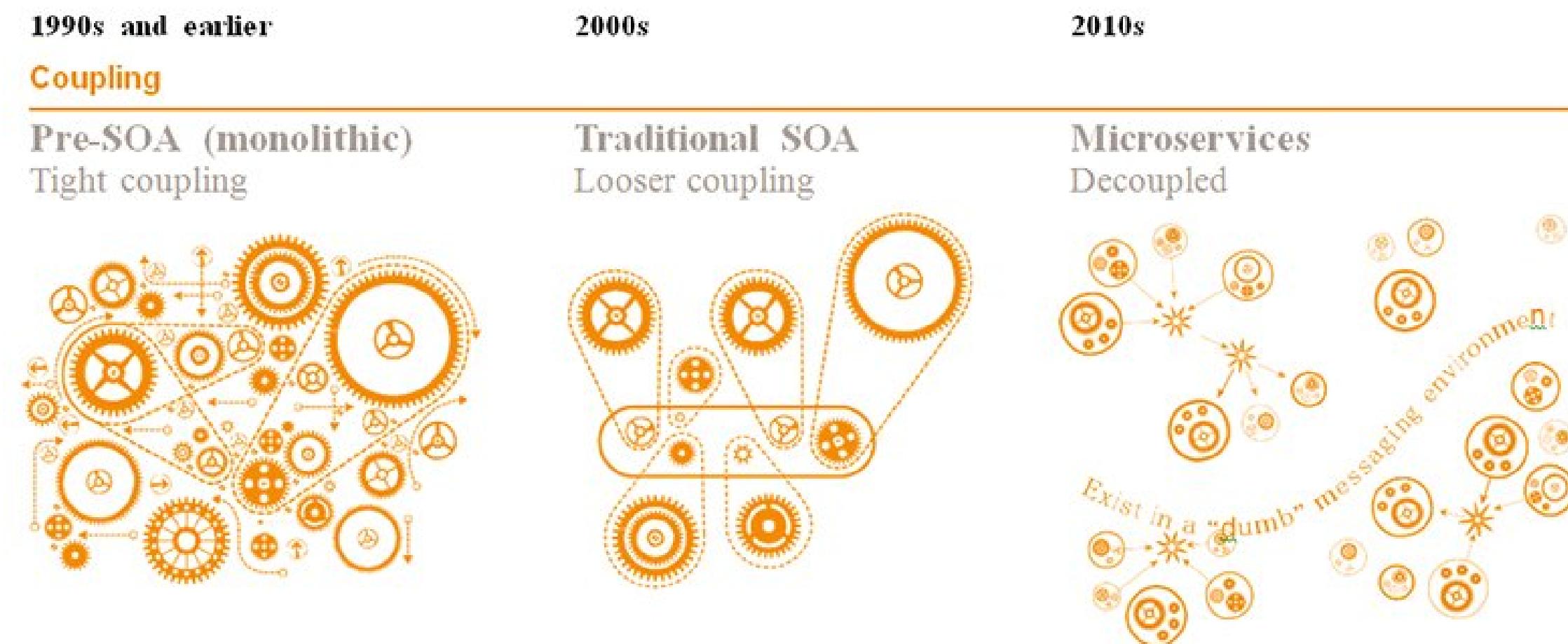


# Why Traefik?



Why, Mr Anderson?

# Evolution Of Software Design



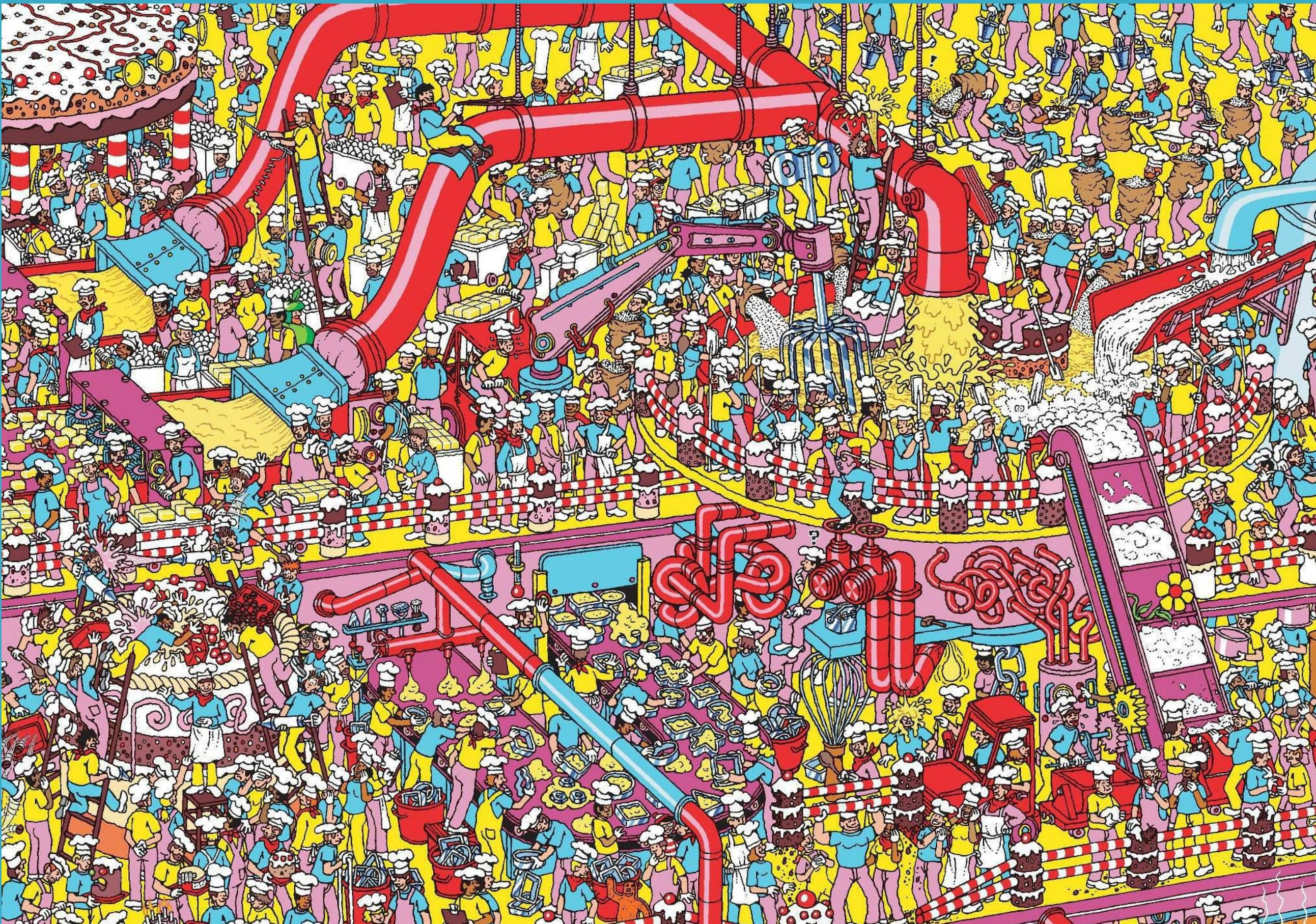
# The Premise Of Microservices...



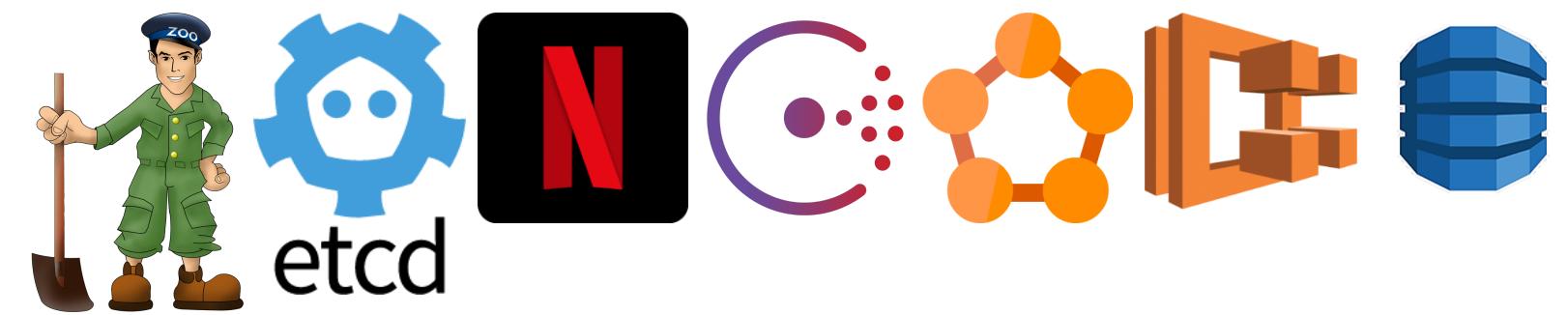
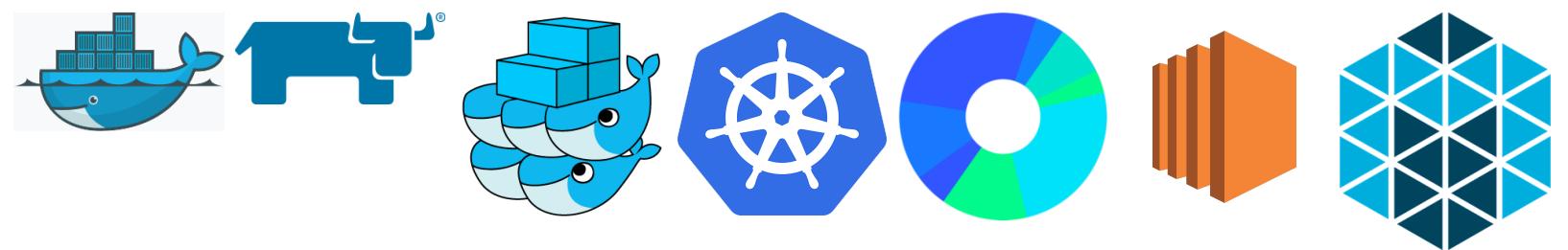
...And What Happens

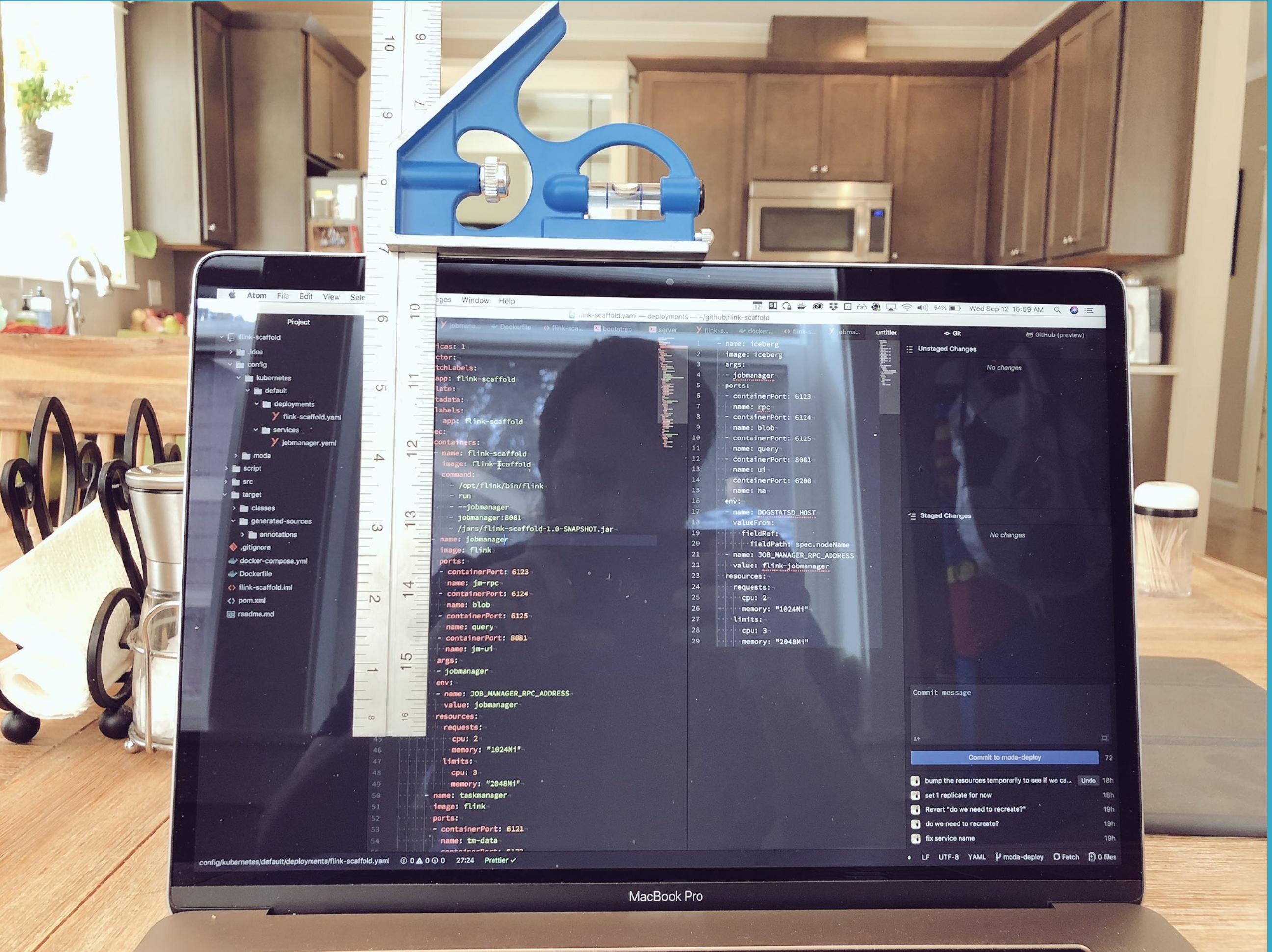


# Where's My Service?



# Tools Of The Trade





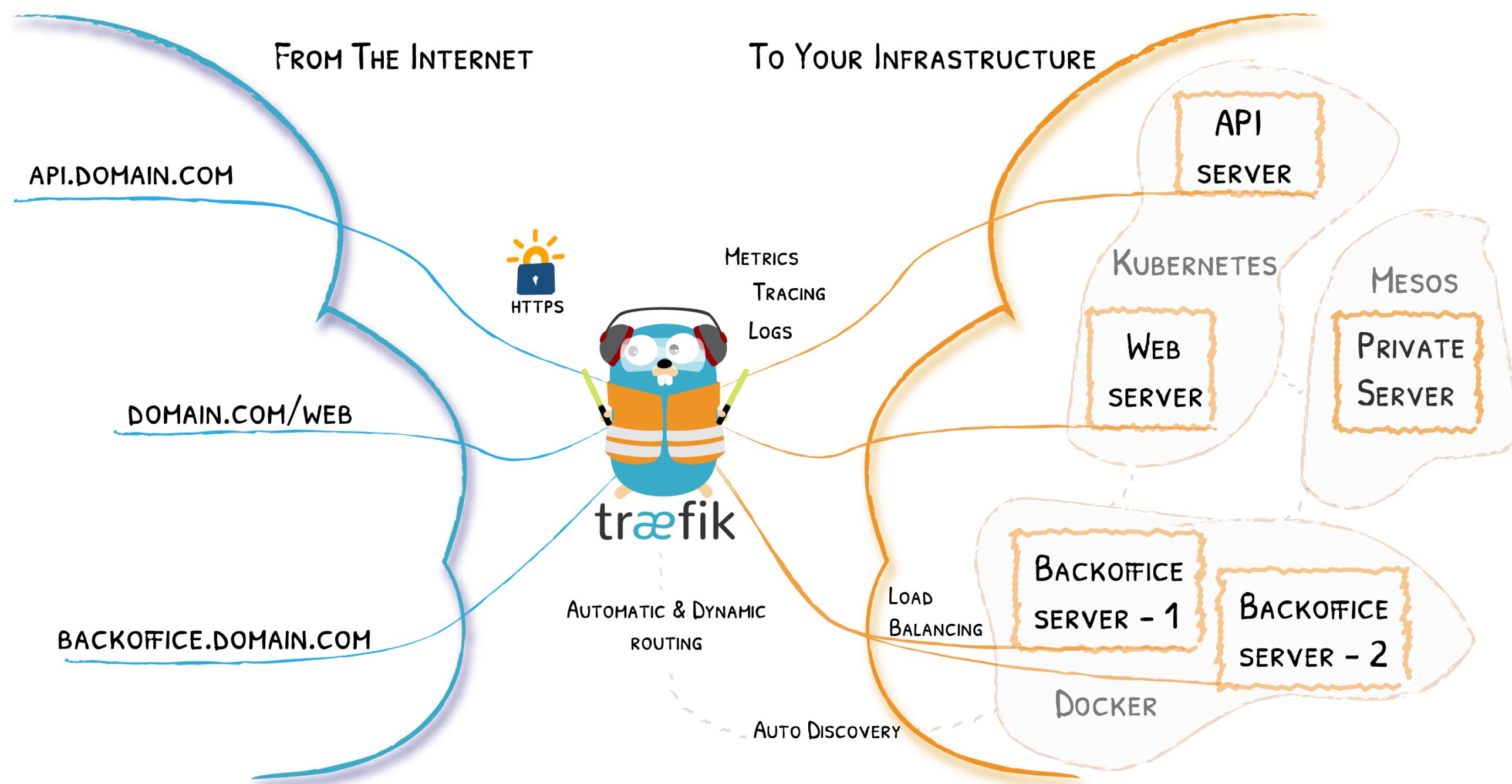
Source: <https://twitter.com/Caged/status/1039937162769096704>

# 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+ ⭐ 1B+ ↓ 400+ 
- Created in 2015, 4Y 
- Current stable branch: v2 . 0

# BACK toTRAEFIK 2.0

Part →

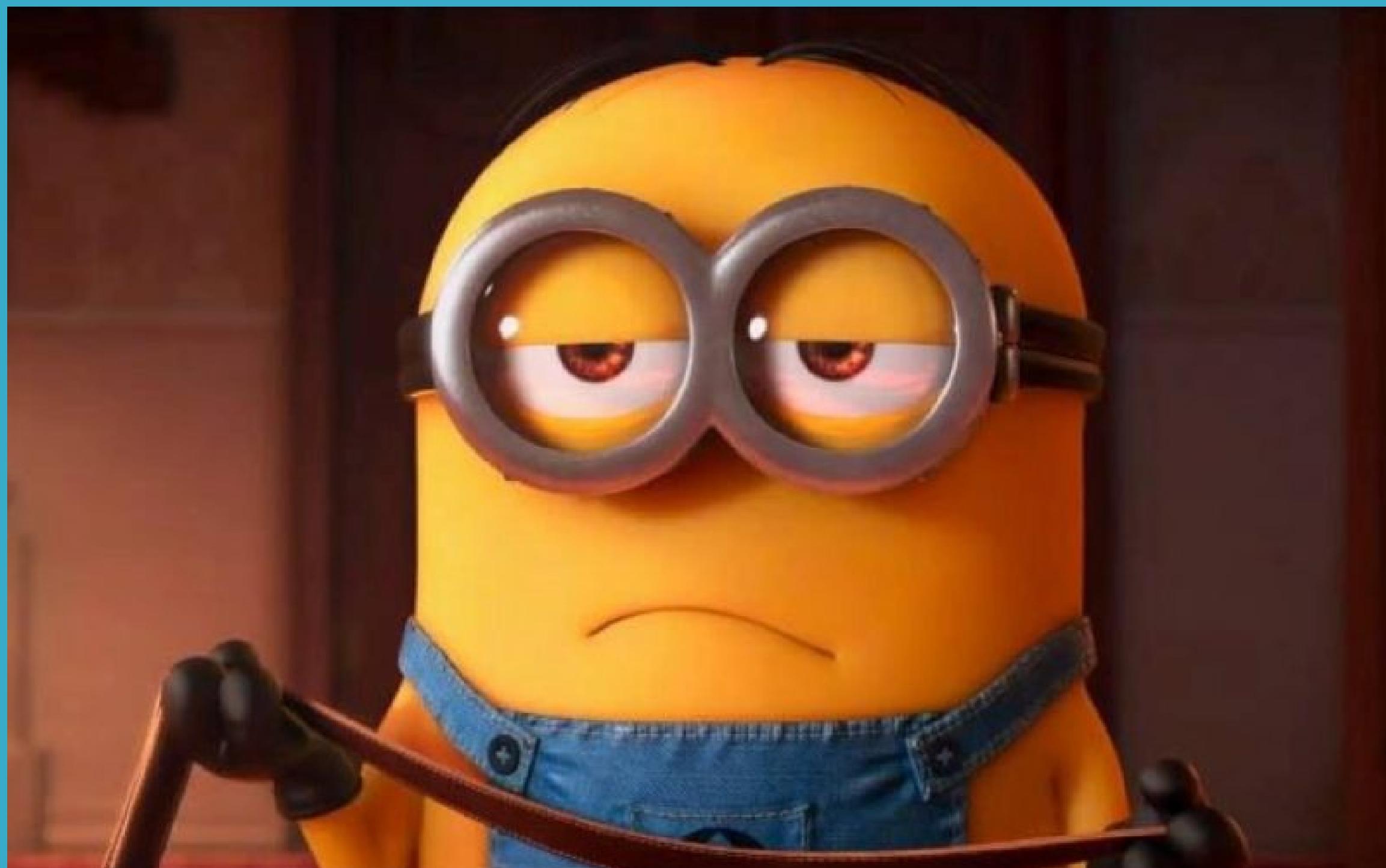


# Traefik 2.0 Quick Overview

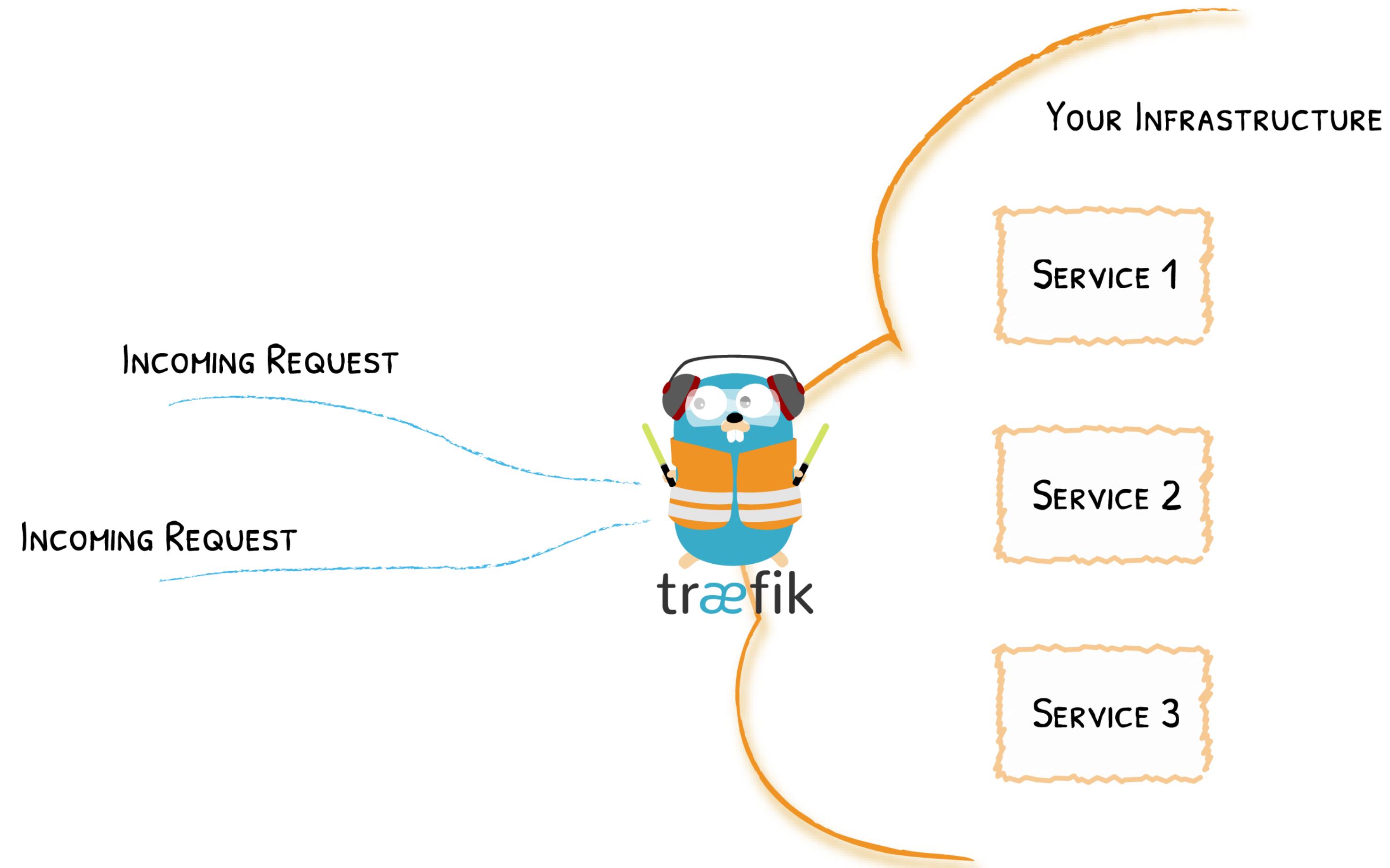
- Revamped Documentation
- Clarified Concepts
- Expressive Routing Rule Syntax
- Middlewares
- TCP Support
- Canary / Mirroring
- And so Much More…

[Learn more on the blog post](#)

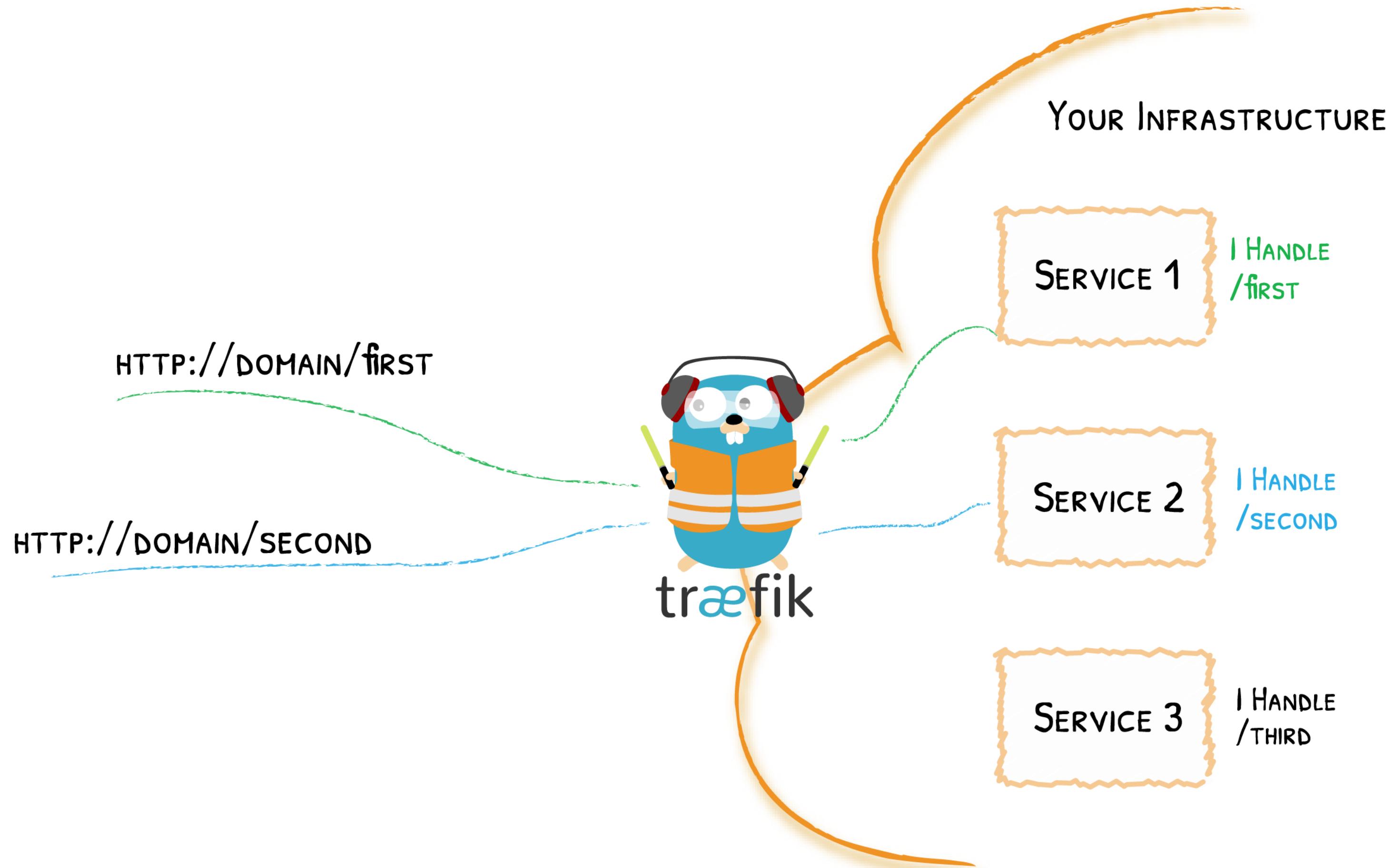
# Traefik (V2.0) Core Concepts



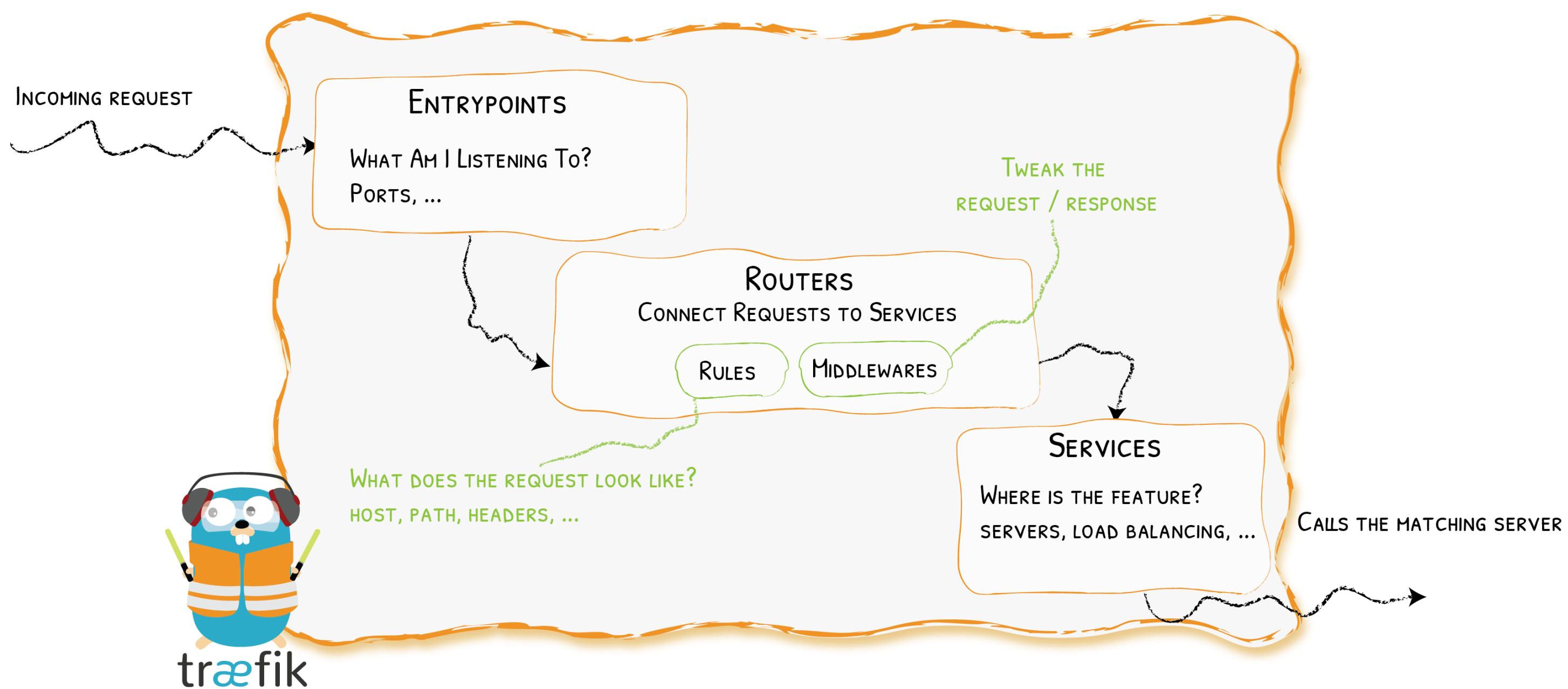
# Traefik Is An Edge Router



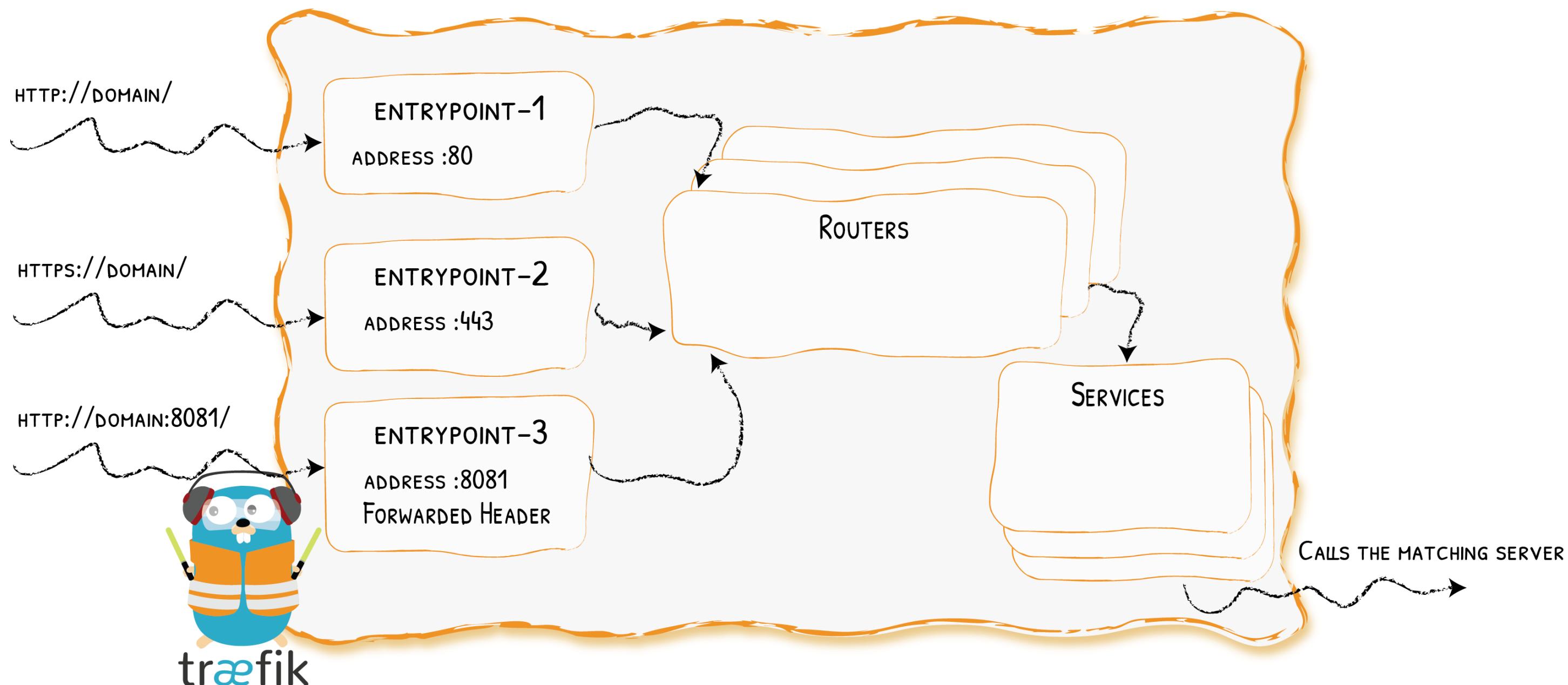
# Dynamically Discovers Services



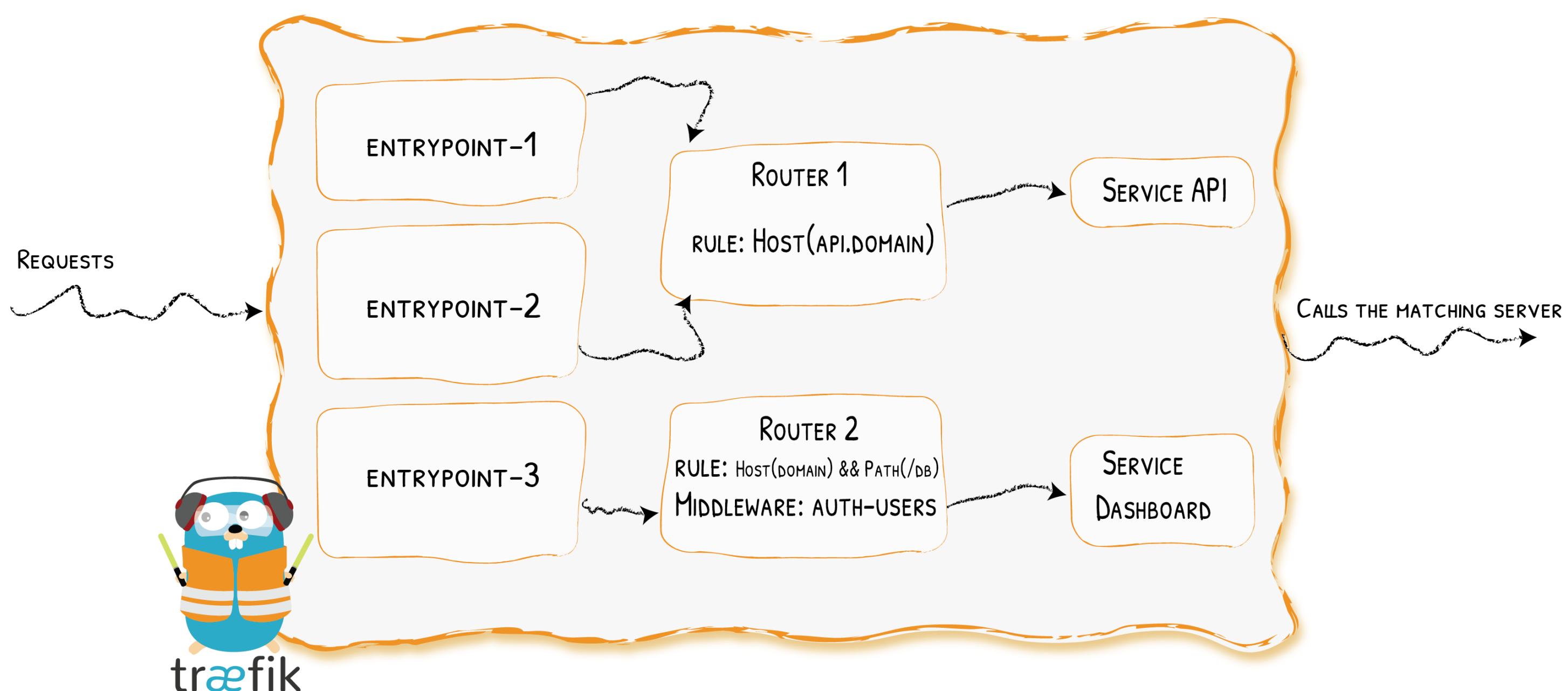
# Architecture (V2.0) At A Glance



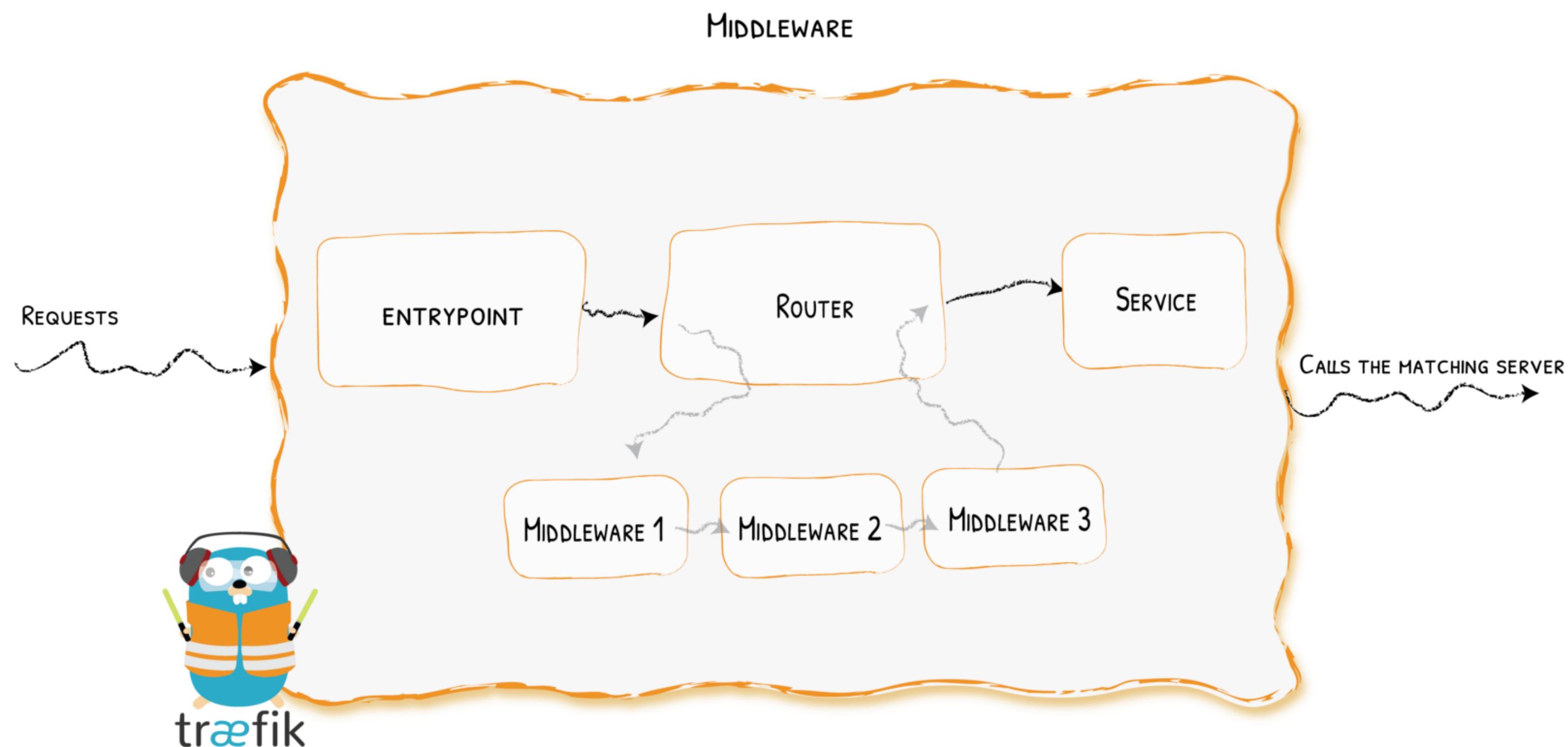
# Entrypoints



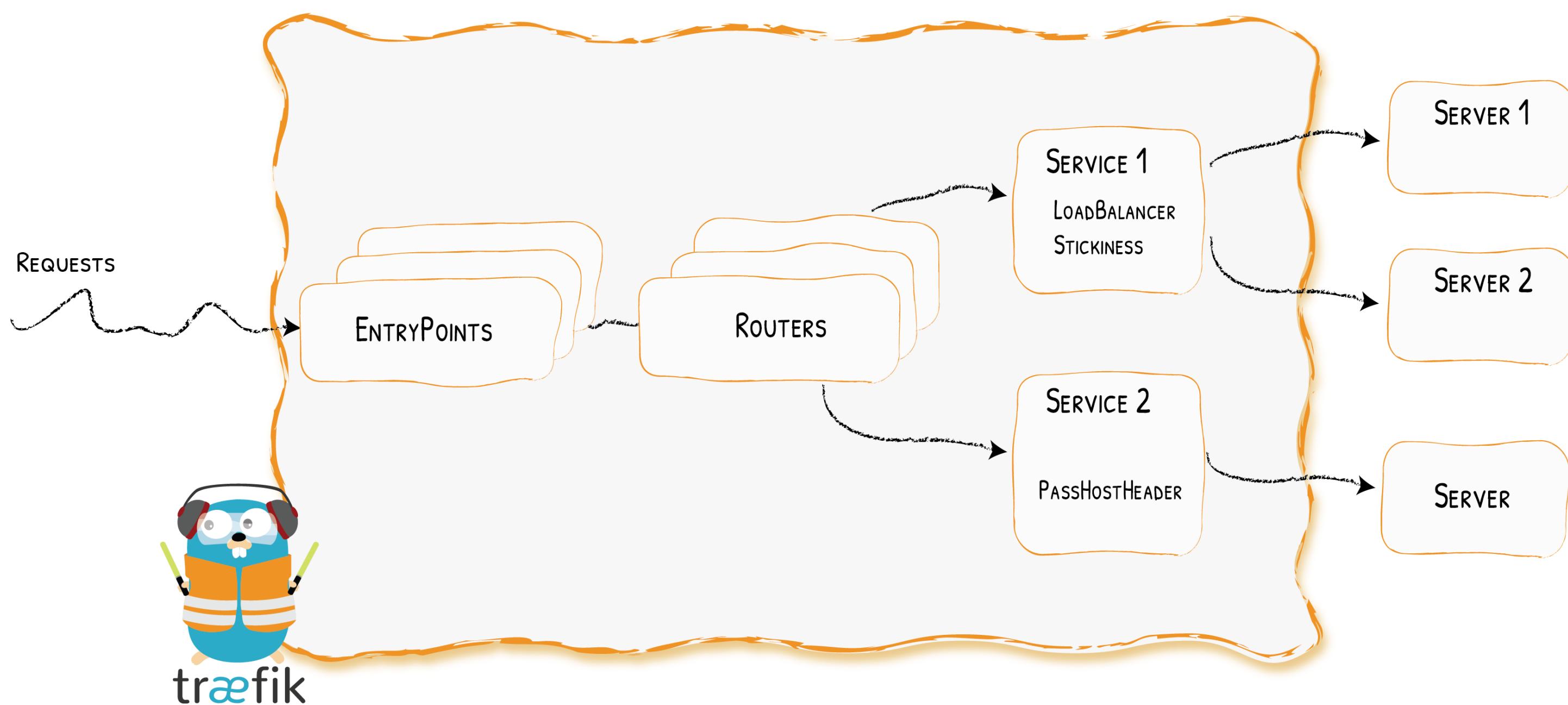
# Routers



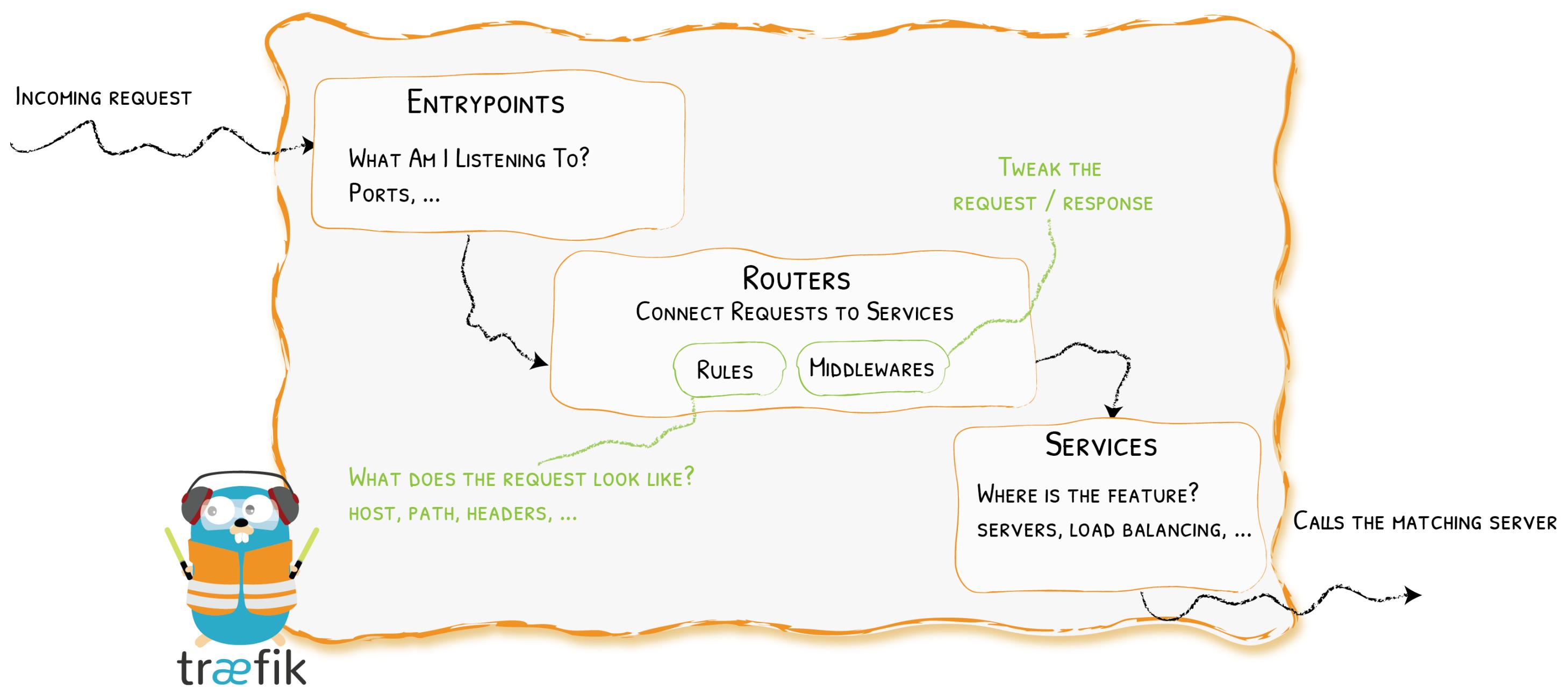
# Middlewares



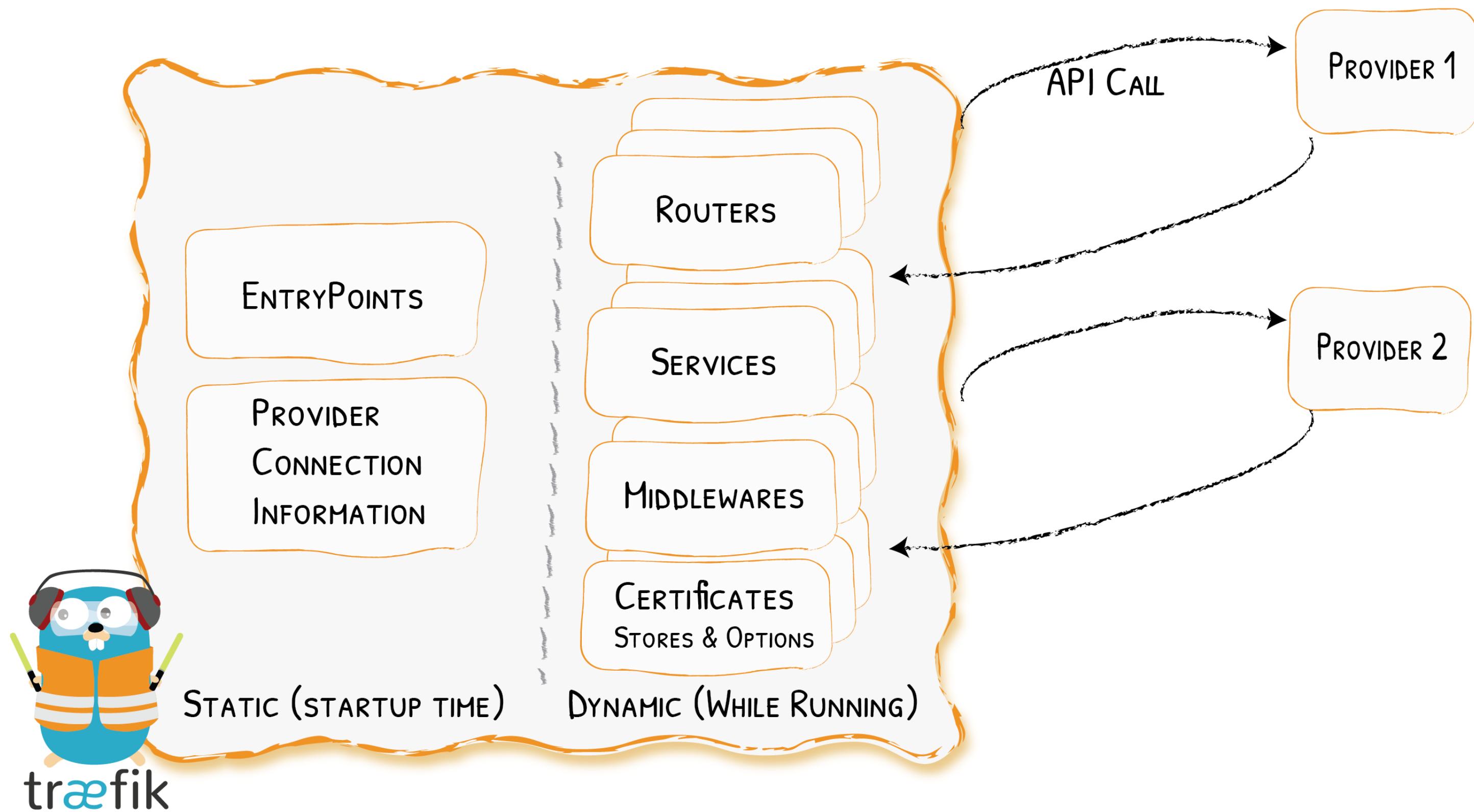
# Services



# Architecture (Again) At A Glance

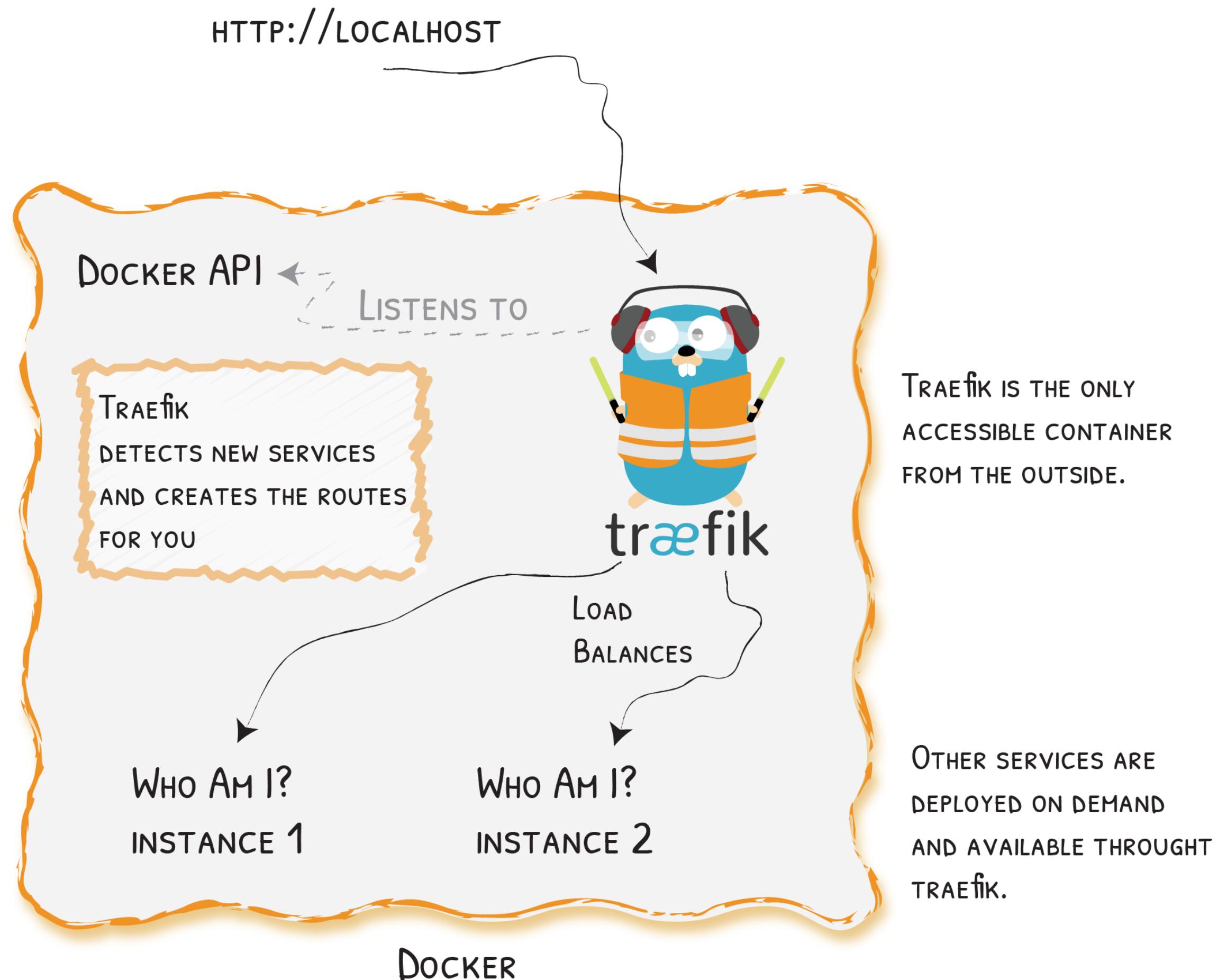


# Static & Dynamic Configuration



# Show Me The Configuration!

# Traefik With



# Example With

```
version: '3'

services:
  reverse-proxy:
    image: traefik:v2.0
    command: --providers.docker.endpoint="tcp://proxy-docker.svc.local:2376"
    ports:
      - "80:80"

  corporate-webapp:
    image: company/corporate-webapp:1.2.3
    labels:
      - "traefik.http.routers.webapp.rule=Host(`company.com`)"

  admin-webapp:
    image: company/admin-webapp:15.2.2
    labels:
      - "traefik.http.routers.admin-webapp.rule=Host(`company.com`) && PathPrefix(`/admin`)"
      - "traefik.http.routers.admin-webapp.service=admin-svc"
      - "traefik.http.services.admin-svc.LoadBalancer.server.Port=9999"
```

# Traefik With ⚓

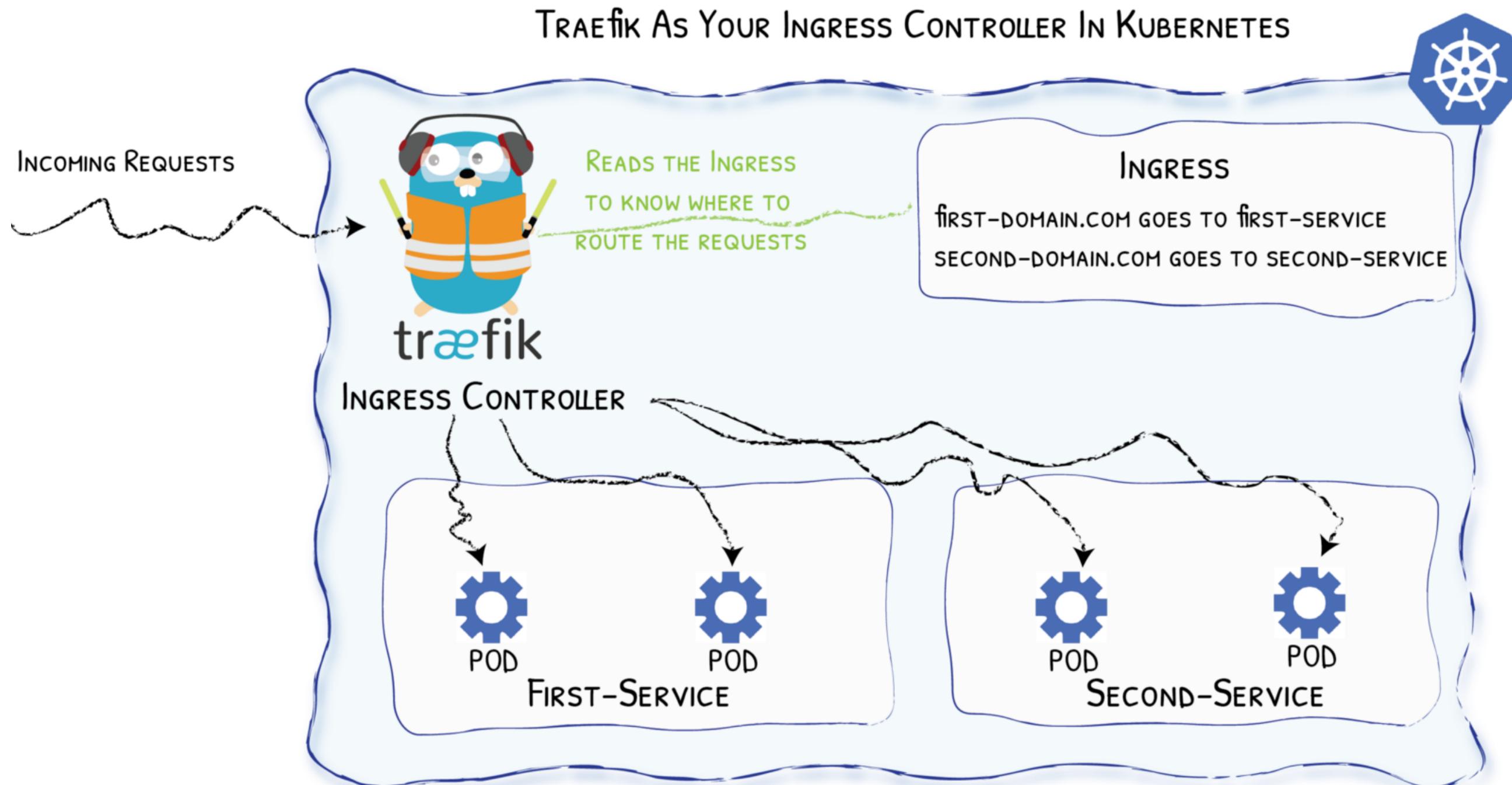


Diagram from <https://medium.com/@geraldcroes>

# Ingress Example With ⚙

```
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
```

# But...

## Annotations

### General annotations

The following general annotations are applicable on the Ingress object:

Annotation	Description
traefik.ingress.kubernetes.io/app-root: "/index.html"	Redirects all requests for / to the defined path. (1)
traefik.ingress.kubernetes.io/error-pages: <YML>	See <a href="#">custom error pages</a> section. (2)
traefik.ingress.kubernetes.io/frontend-entry-points: http,https	Override the default frontend endpoints.
traefik.ingress.kubernetes.io/pass-client-tls-cert: <YML>	Forward the client certificate following the configuration in YAML. (3)
traefik.ingress.kubernetes.io/pass-tls-cert: "true"	Override the default frontend PassTLSCert value. Default: false. (DEPRECATED)
traefik.ingress.kubernetes.io/preserve-host: "true"	Forward client Host header to the backend.
traefik.ingress.kubernetes.io/priority: "3"	Override the default frontend rule priority.
traefik.ingress.kubernetes.io/rate-limit: <YML>	See <a href="#">rate limiting</a> section. (4)
traefik.ingress.kubernetes.io/redirect-entry-point: https	Enables Redirect to another entryPoint for that frontend (e.g. HTTPS).
traefik.ingress.kubernetes.io/redirect-permanent: "true"	Return 301 instead of 302.
traefik.ingress.kubernetes.io/redirect-regex: ^http://localhost/(.*)	Redirect to another URL for that frontend. Must be set with traefik.ingress.kubernetes.io/redirect-replacement.
traefik.ingress.kubernetes.io/redirect-replacement: http://mydomain/\$1	Redirect to another URL for that frontend. Must be set with traefik.ingress.kubernetes.io/redirect-regex.
traefik.ingress.kubernetes.io/request-modifier: AddPrefix: /users	Adds a <a href="#">request modifier</a> to the backend request.
traefik.ingress.kubernetes.io/rewrite-target: /users	Replaces each matched Ingress path with the specified one, and adds the old path to the X-Replaced-Path header.

## Annotations

You can add these Kubernetes annotations to specific Ingress objects to customize their behavior.

Tip	
Annotation keys and values can only be strings. Other types, such as boolean or numeric values must be quoted, i.e. "true", "false", "100".	
Note	
The annotation prefix can be changed using the --annotations-prefix command line argument, but the default is nginx.ingress.kubernetes.io, as described in the table below.	
Name	type
nginx.ingress.kubernetes.io/app-root	string
nginx.ingress.kubernetes.io/affinity	cookie
nginx.ingress.kubernetes.io/affinity-mode	"balanced" or "persistent"
nginx.ingress.kubernetes.io/auth-realm	string
nginx.ingress.kubernetes.io/auth-secret	string
nginx.ingress.kubernetes.io/auth-secret-type	string
nginx.ingress.kubernetes.io/auth-type	basic or digest
nginx.ingress.kubernetes.io/auth-tls-secret	string
nginx.ingress.kubernetes.io/auth-tls-verify-depth	number
nginx.ingress.kubernetes.io/auth-tls-verify-client	string
nginx.ingress.kubernetes.io/auth-tls-error-page	string
nginx.ingress.kubernetes.io/auth-tls-pass-certificate-to-upstream	"true" or "false"
nginx.ingress.kubernetes.io/auth-url	string
nginx.ingress.kubernetes.io/auth-cache-key	string

# ✳️ 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
```

# Demo



# Demo Menu

[Click here for demo's instructions](#)

1. Install Traefik v2 in Kubernetes cluster
2. Deploy and expose an HTTP application
3. Deploy and expose a TCP application

# Install Traefik In Kubernetes

[Click here for demo's instructions](#)

1.  Custom Resource Definitions
2.  Configure RBAC
  - To allow watching the Kubernetes API
3.  Install Traefik as a Deployment
4.  Expose and  publish Traefik with the right Service type
  - LoadBalancer, NodePort?
5.  Configure Traefik and its features

# Example: HTTP Application

[Click here for demo's instructions](#)

1.  Install the web application as a Deployment
2.  Expose the web application with a Service of type ClusterIP
  - Internal access only
3.  Publish the web application with an IngressRoute

# Example: TCP Application

[Click here for demo's instructions](#)

Example with MongoDB:

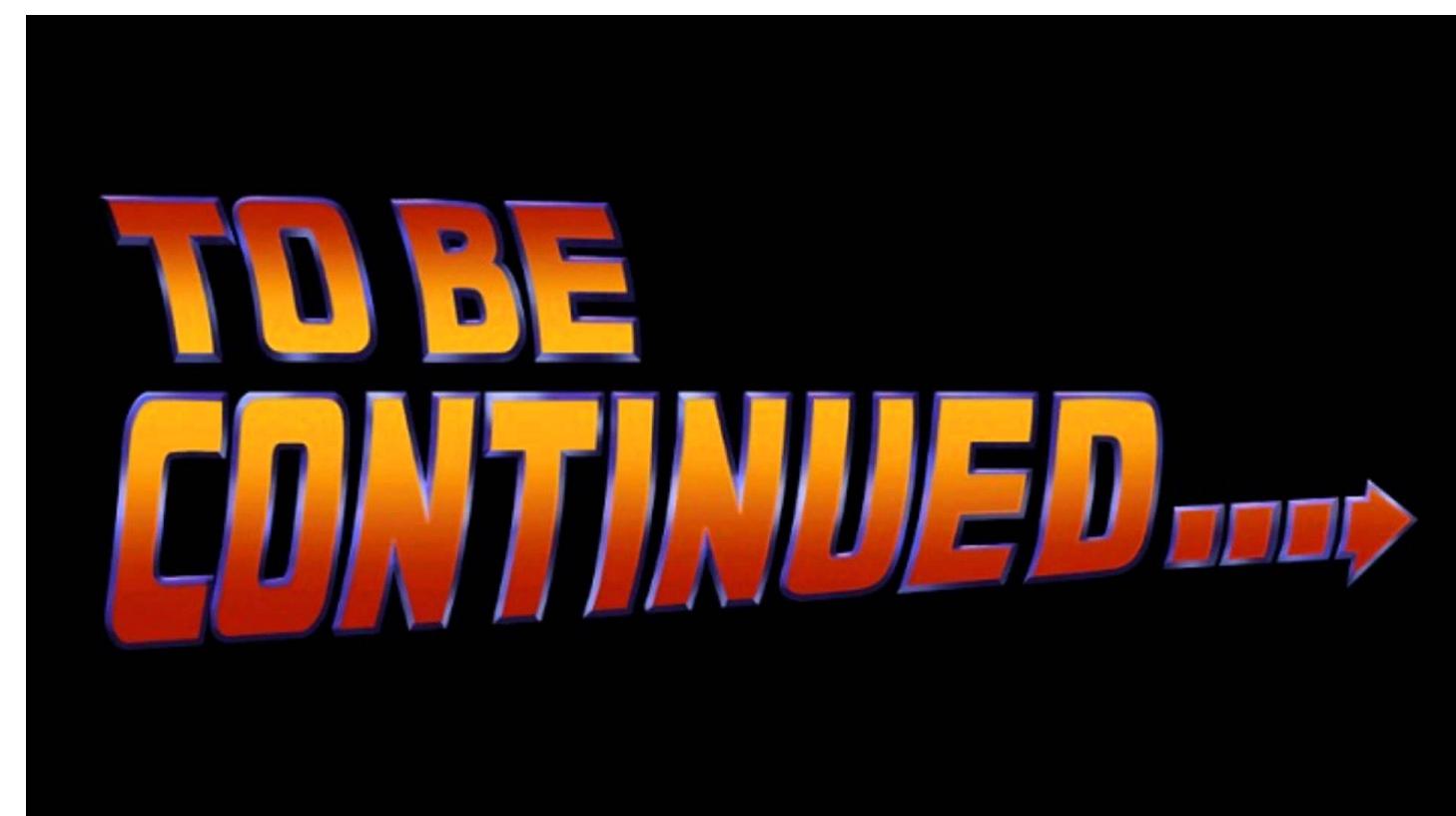
1. Install MongoDB as a Deployment
2. Expose MongoDB with a Service of type ClusterIP
  - Internal access only
3. Publish MongoDB with an IngressRouteTCP

# More To Come

- New Helm Chart for v2.0
- Advanced Load-Balancing with CRDs: Canary, Mirroring, StickySession, etc.
  - Available today with File and Docker providers
- Example and Guides
- UDP?

# More Info

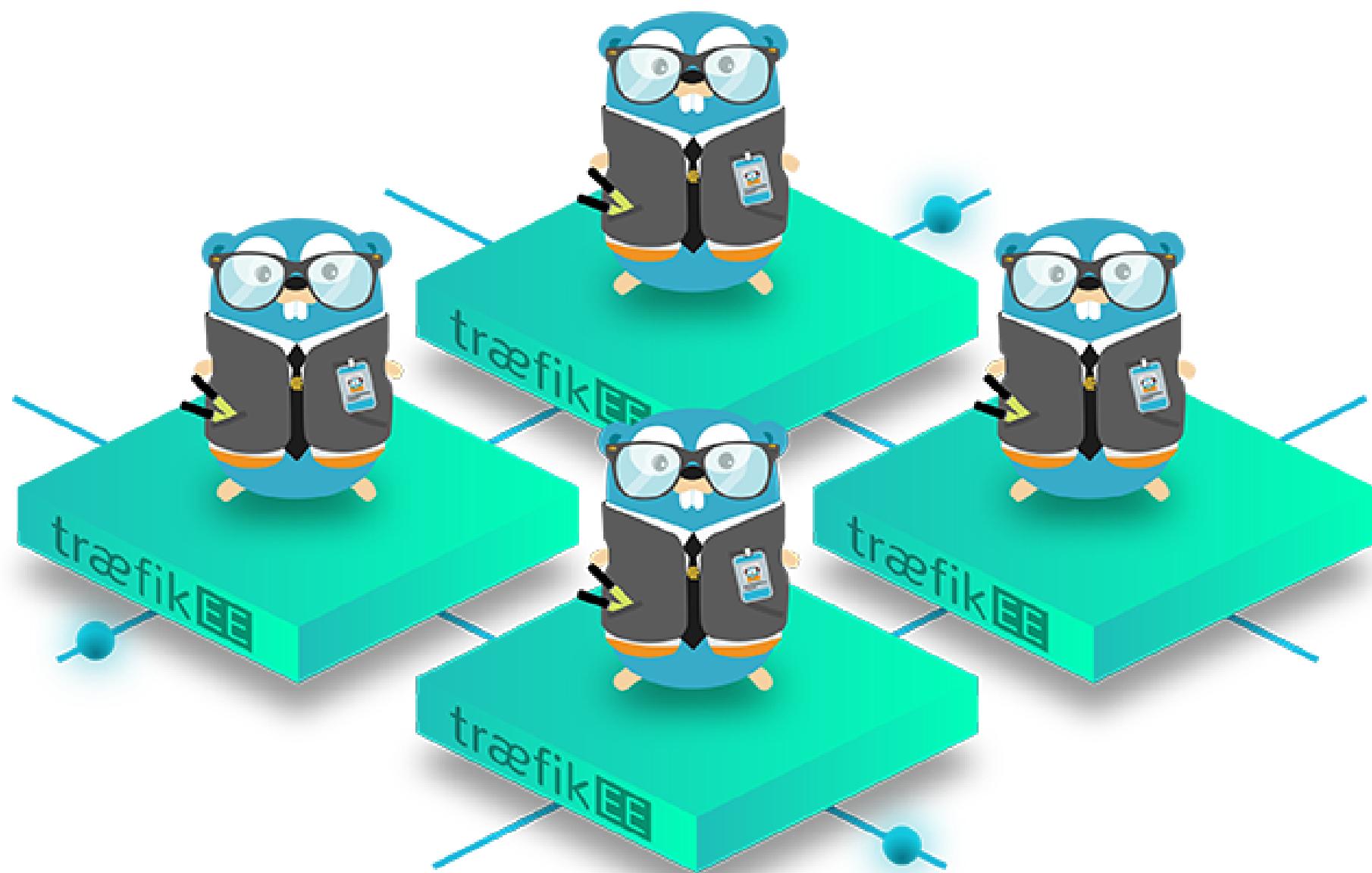
- Documentation
- Traefik 2.0 Blog Post

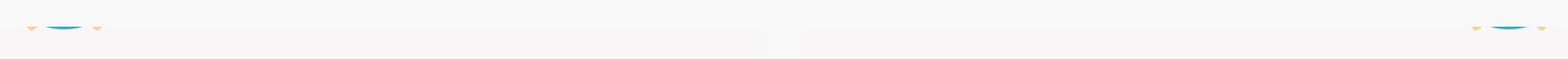


# We Also Missed Talking About ...

A circular word cloud centered around the term "CIRCUIT-BREAKERS". The words are arranged in a circle and include: mesos, ZIPKIN, LIMITING, KUBERNETES, Metrics, CERTIFICATE, TLS, Reverse-Proxy, HEADERS, DYNAMIC/WILDCARD, Security, Configurations, Tracing, PROXY, PROMETHEUS, JAEGER, SECRETS, WEBSOCKETS, SSL, FORWARD, REDIRECTS, DOCKER, CHECKS, PROTOCOL, HEALTH, HSTS, CLUSTER, AUTH, RATE, CONSUL, SWARM, MODE.

# Traefik Also Comes In Herd





HIGH AVAILABILITY

traefik ENTERPRISE EDITION



SECURITY

traefik ENTERPRISE EDITION

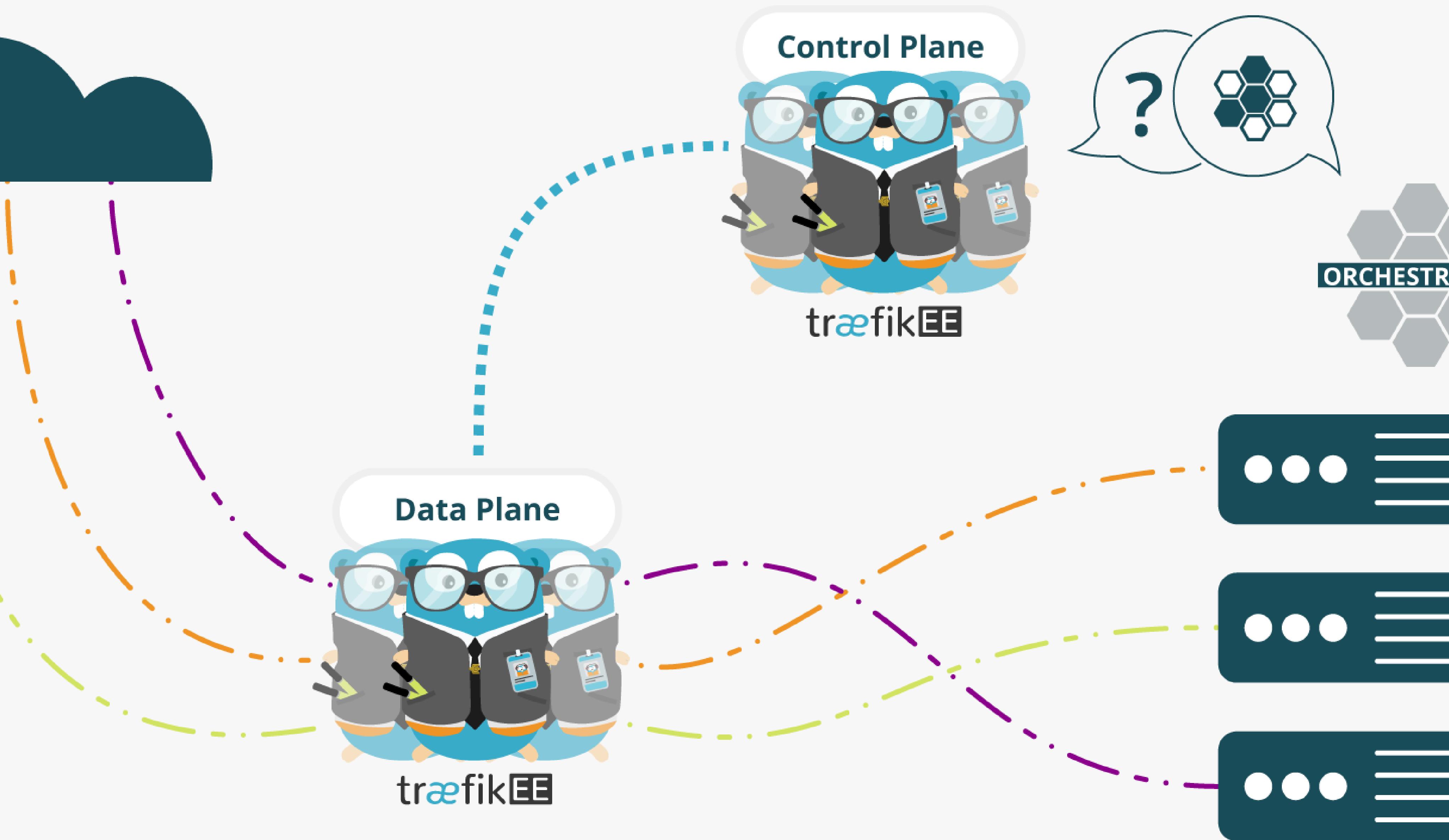


SCALABILITY

traefik ENTERPRISE EDITION

INTERNET

TO YOUR INFRA



# As Simple As Traefik

- Install it:

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

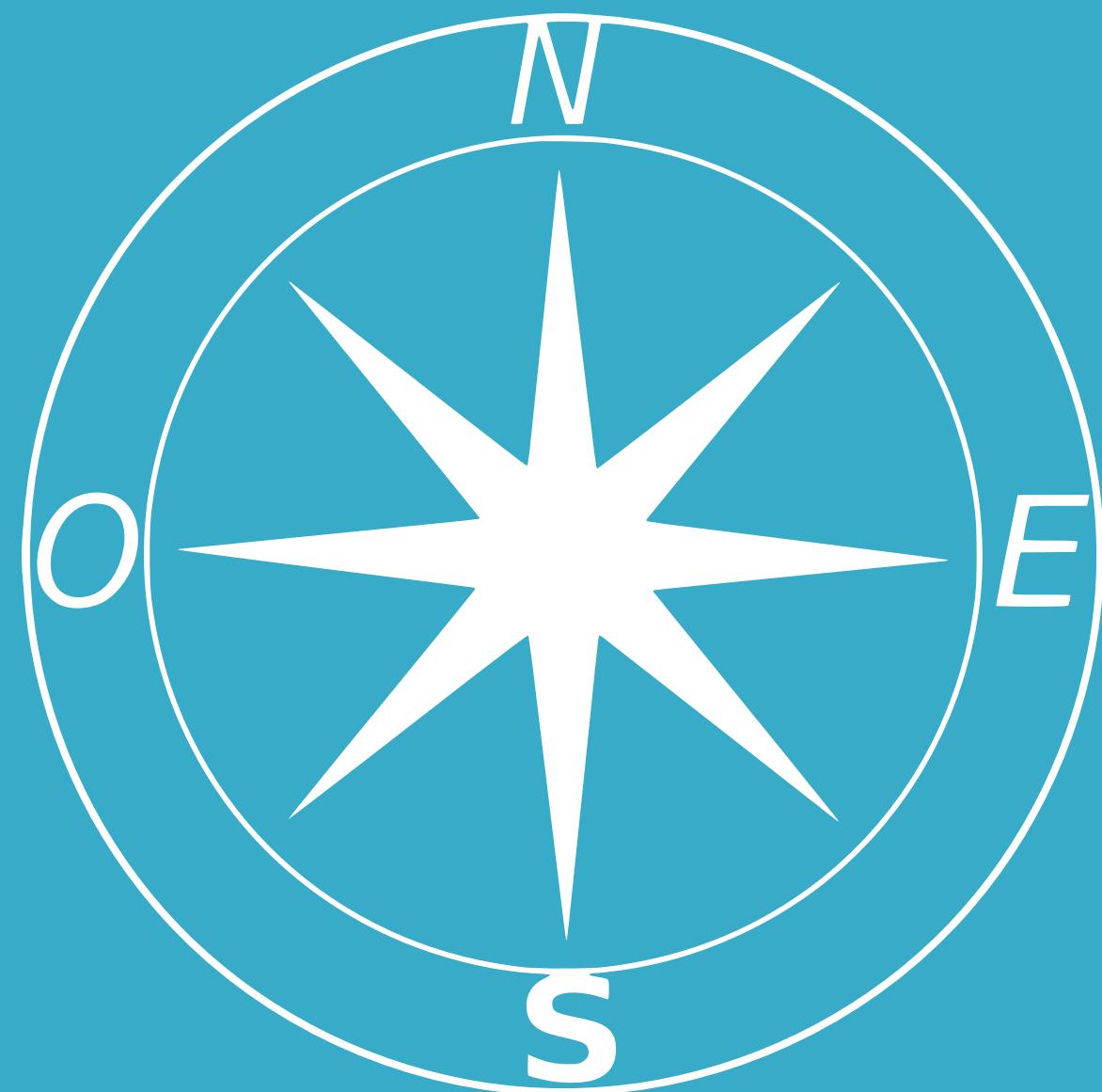
- Configure it:

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

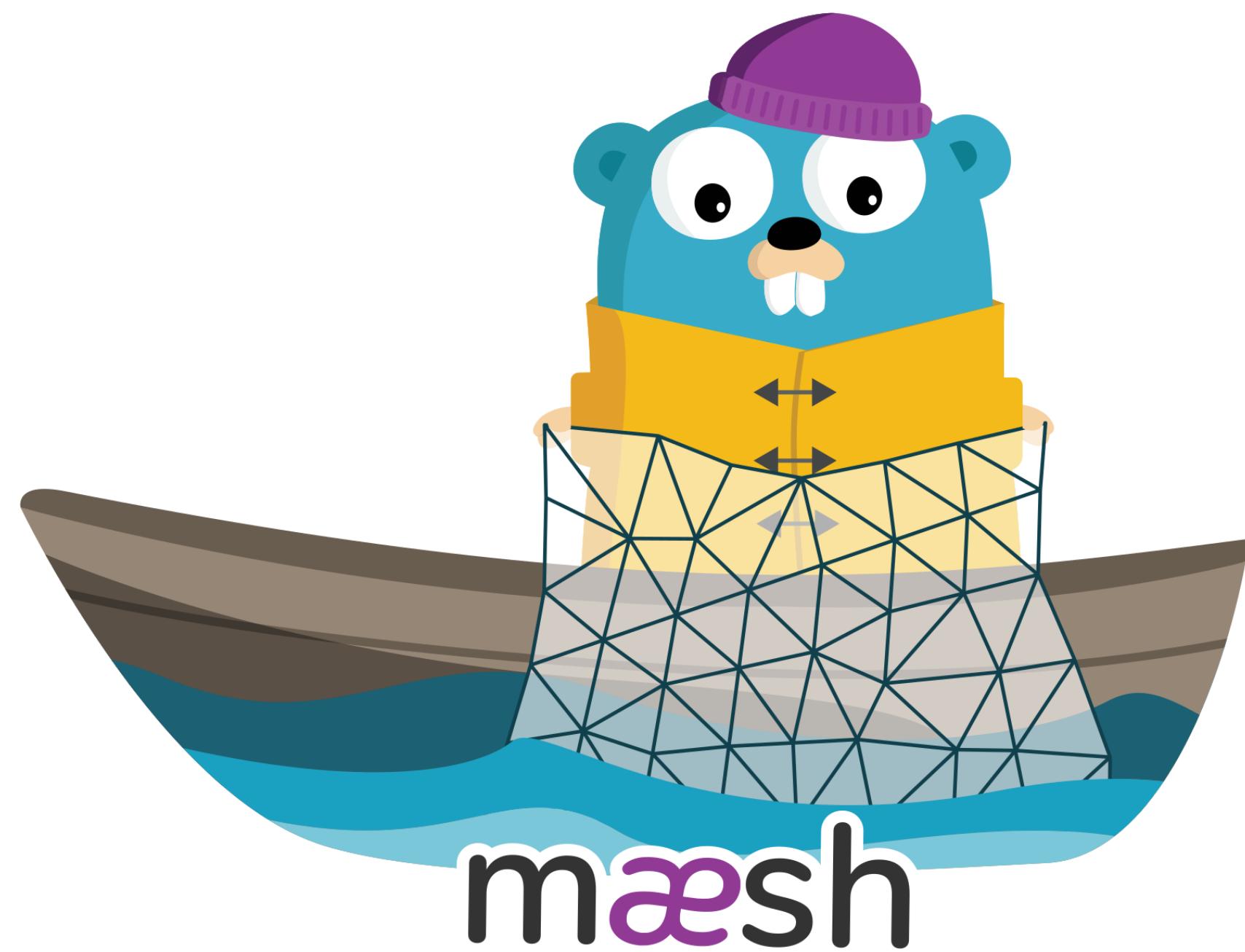
# Free Trial

<https://containo.us/traefikee>

# East / West Traefik



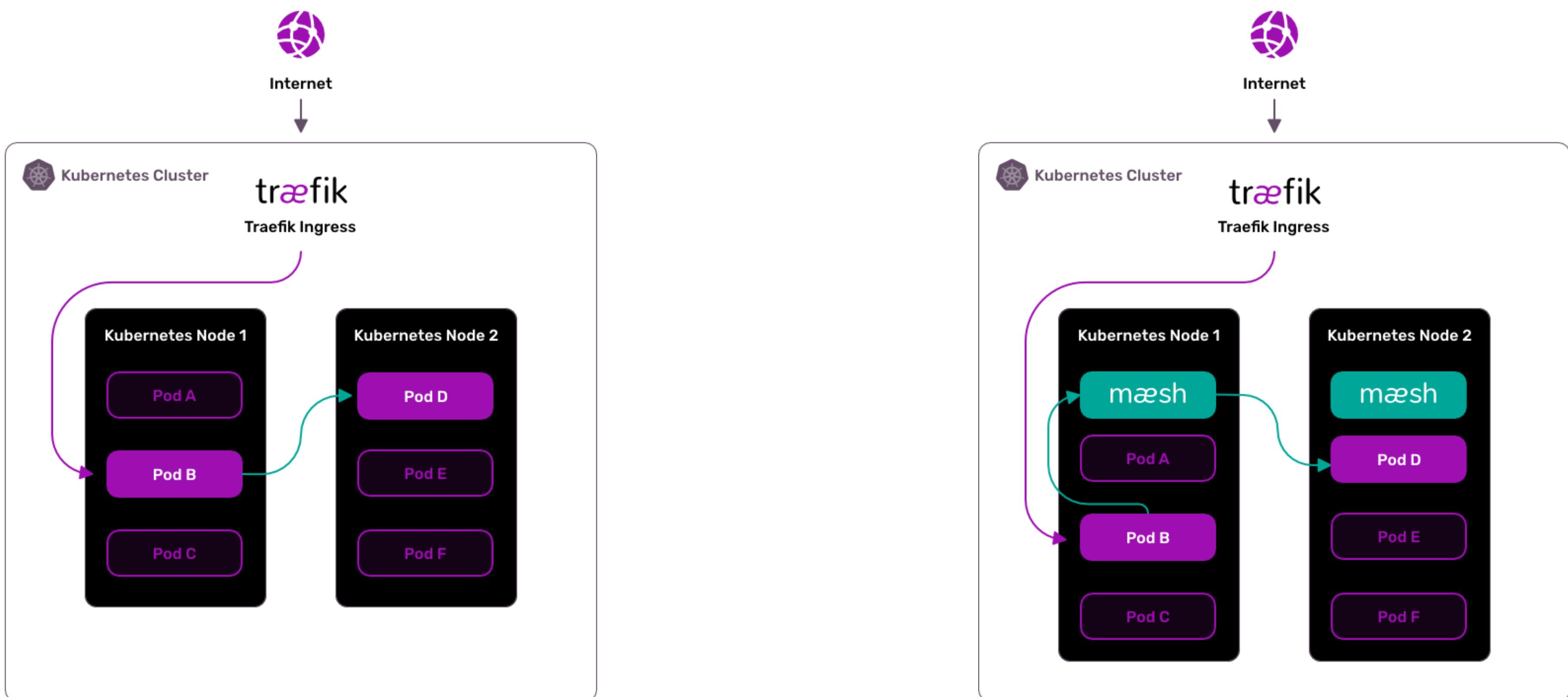
# Say Hello To Maesh



# 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.*

# Maesh Architecture



# More On Maesh

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

Maesh Website

# Show Me The Code!

- Install Maesh (Helm Chart):

```
helm repo add maesh https://containous.github.io/maesh/charts
helm repo update
helm install --name=maesh --namespace=maesh maesh/maesh --values=./maesh/values.yaml
```

- Deploy Applications:

```
kubectl apply -f apps/0-namespace.yaml
kubectl apply -f apps/1-svc-accounts.yaml
kubectl apply -f apps/2-apps-client.yaml
kubectl apply -f apps/3-apps-servers.yaml
kubectl apply -f apps/4-ingressroutes.yaml
```

- Deploy SMI Objects to allow traffic in the mesh:

```
kubectl apply -f apps/5-smi-http-route-groups.yaml
kubectl apply -f apps/6-smi-traffic-targets.yaml
```

# A Closer Look To SMI Objects

```
apiVersion: specs.smi-spec.io/v1alpha1
kind: HTTPRouteGroup
metadata:
  name: app-routes
  namespace: apps
matches:
- name: all
  pathRegex: "/"
  methods: [ "*" ]
---
apiVersion: access.smi-spec.io/v1alpha1
kind: TrafficTarget
metadata:
  name: client-apps
  namespace: apps
destination:
  kind: ServiceAccount
  name: apps-server
  namespace: apps
specs:
- kind: HTTPRouteGroup
  name: app-routes
  matches:
  - all
sources:
- kind: ServiceAccount
  name: apps-client
  namespace: apps
```

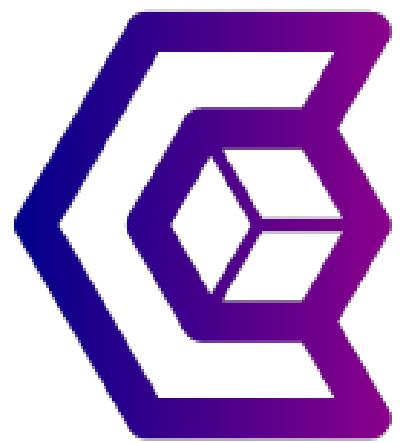
That's All Folks!



We Have  
Stickers!

traefik

# We Are Hiring!



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

# Thank You!

 @DamienDuportal

 dduportal



- Slides (HTML): <https://containous.github.io/slides/meetup-louvain-kube-2019>
- Slides (PDF): <https://containous.github.io/slides/meetup-louvain-kube-2019/slides.pdf>
- Source on : <https://github.com/containous/slides/tree/meetup-louvain-kube-2019>