

(📖) Quoi De Neuf Avec Traefik 2.0, Le Edge Router À 1 Milliard De Téléchargements?



Toulouse Devops Meetup - 2019

Whoami

Jean-Baptiste Doumenjou

- 📖 Developer
- Maintainer of Træfik
- 🐦 @jbdoumenjou
- 🐙 jbdoumenjou



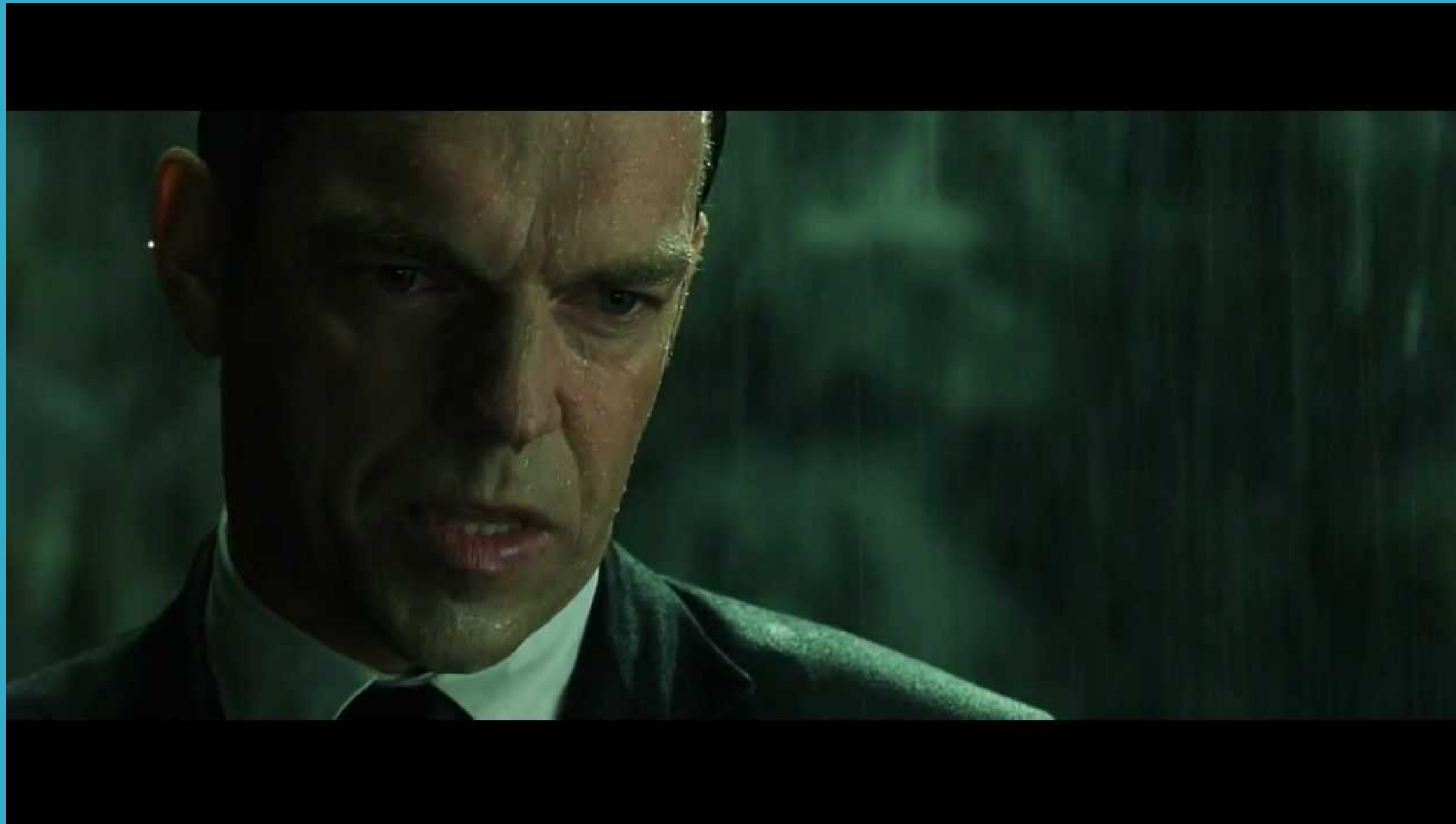
Containous

<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?

THE EVOLUTION OF SOFTWARE ARCHITECTURE

1990's

SPAGHETTI-ORIENTED
ARCHITECTURE
(aka Copy & Paste)



2000's

LASAGNA-ORIENTED
ARCHITECTURE
(aka Layered Monolith)



2010's

RAVIOLI-ORIENTED
ARCHITECTURE
(aka Microservices)



WHAT'S NEXT?

PROBABLY PIZZA-ORIENTED ARCHITECTURE

By @benorama

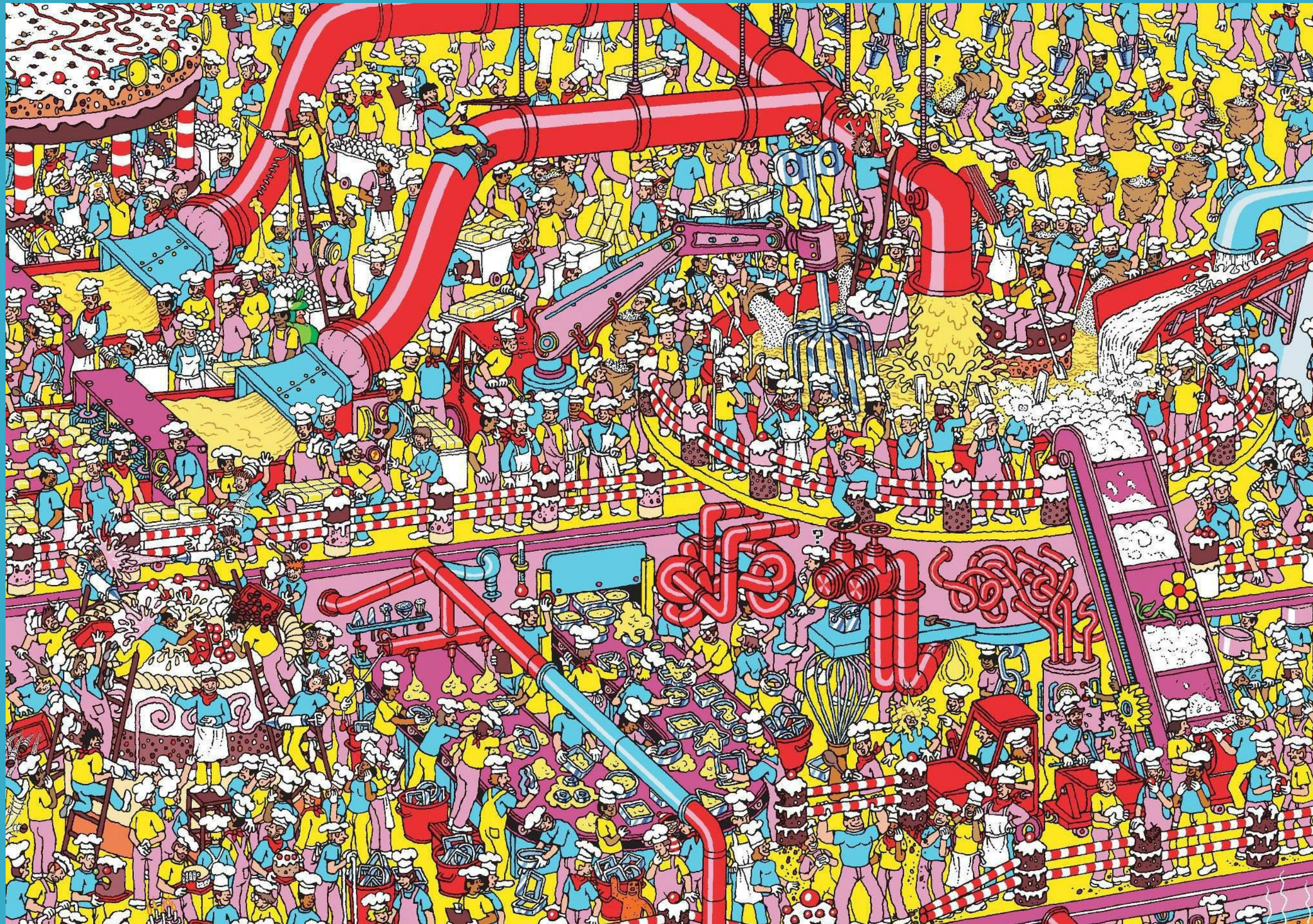
The Premise Of Microservices...



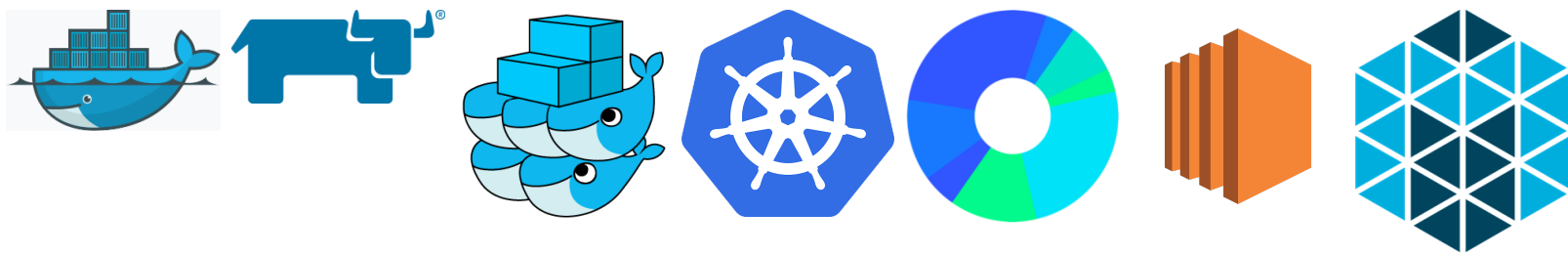
...And What Happens

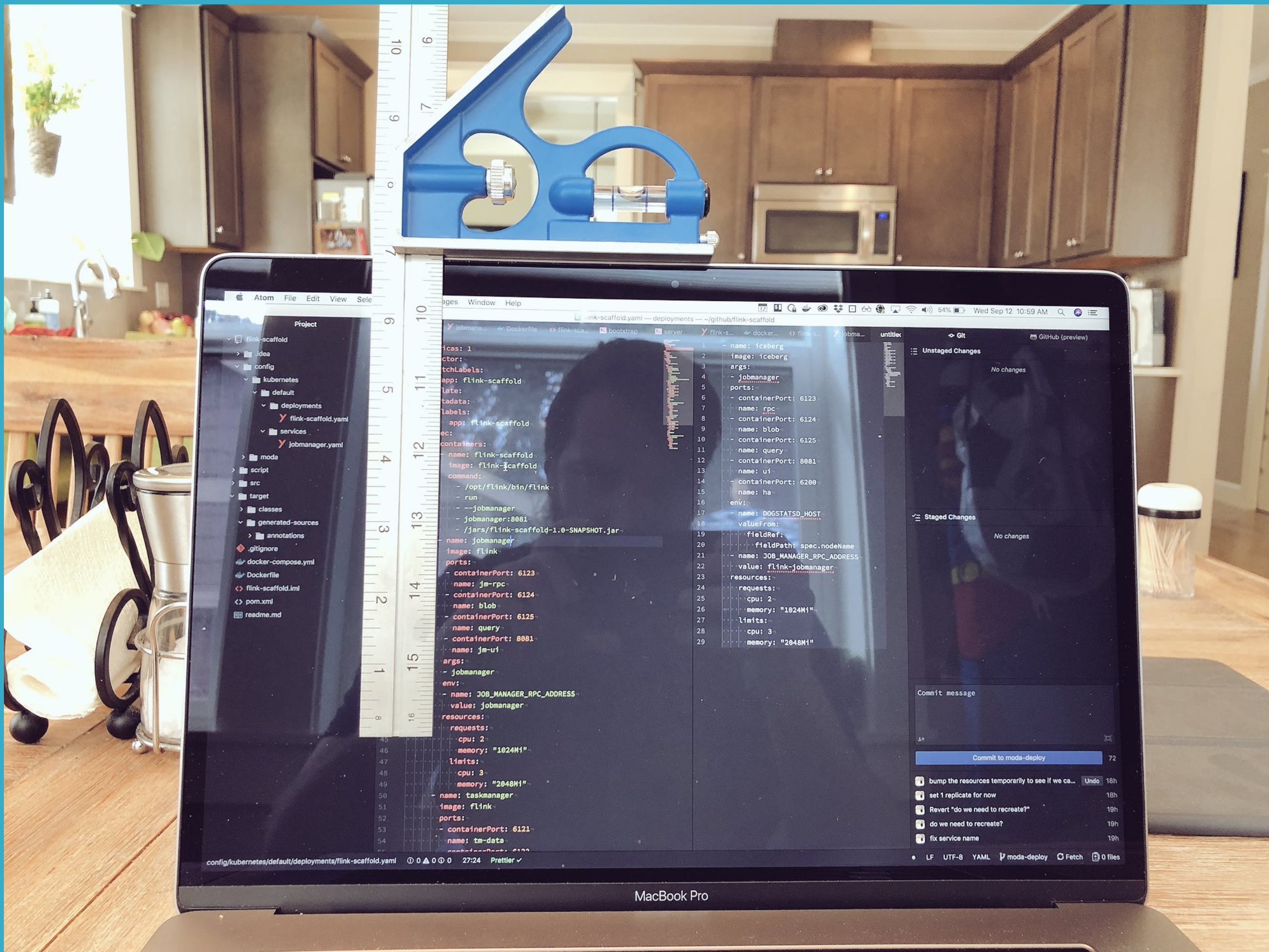


Where's My Service?



Tools Of The Trade





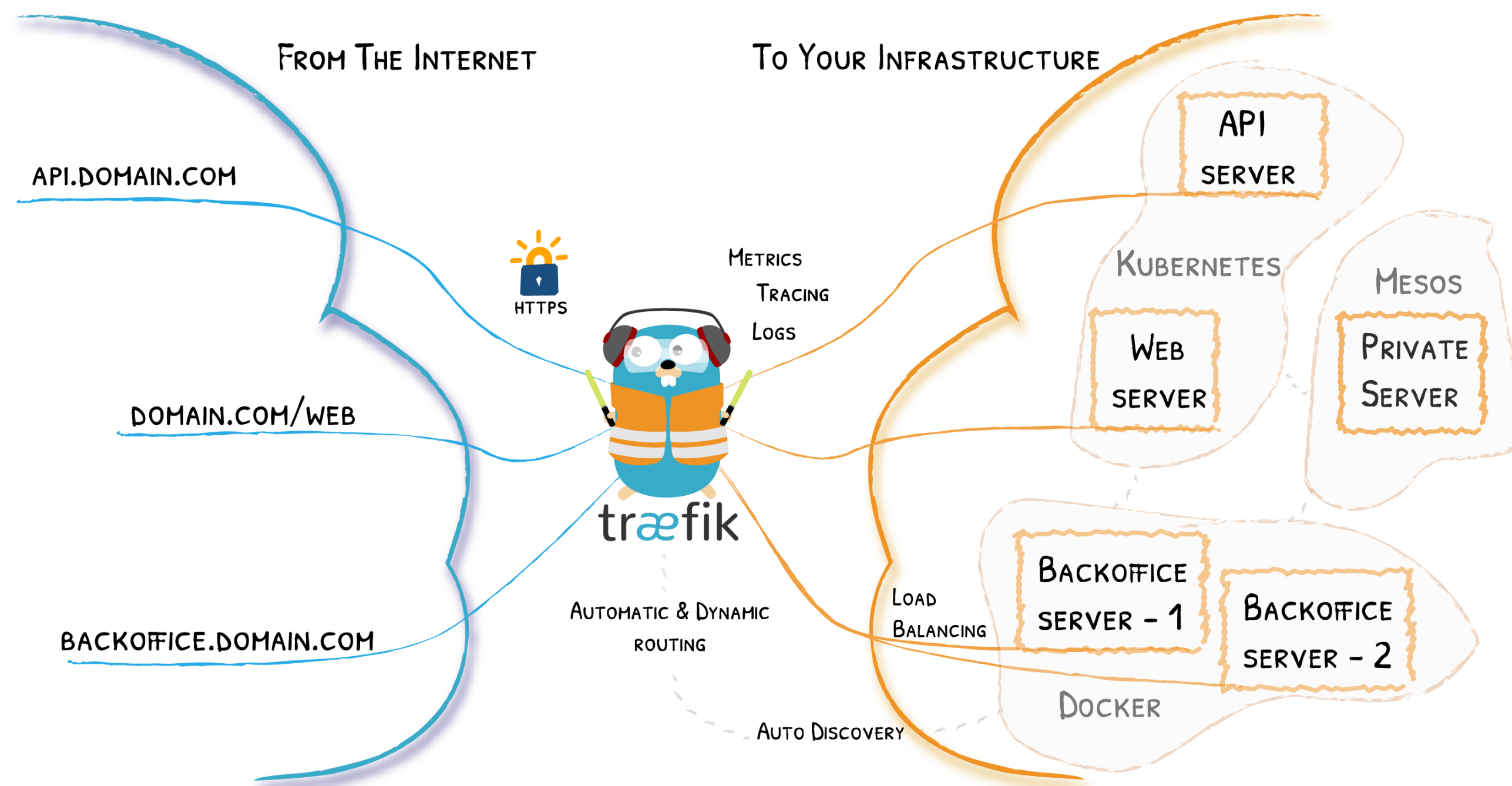
Source: <https://twitter.com/Caged/status/1039937162769096704>

What If I Told You?



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

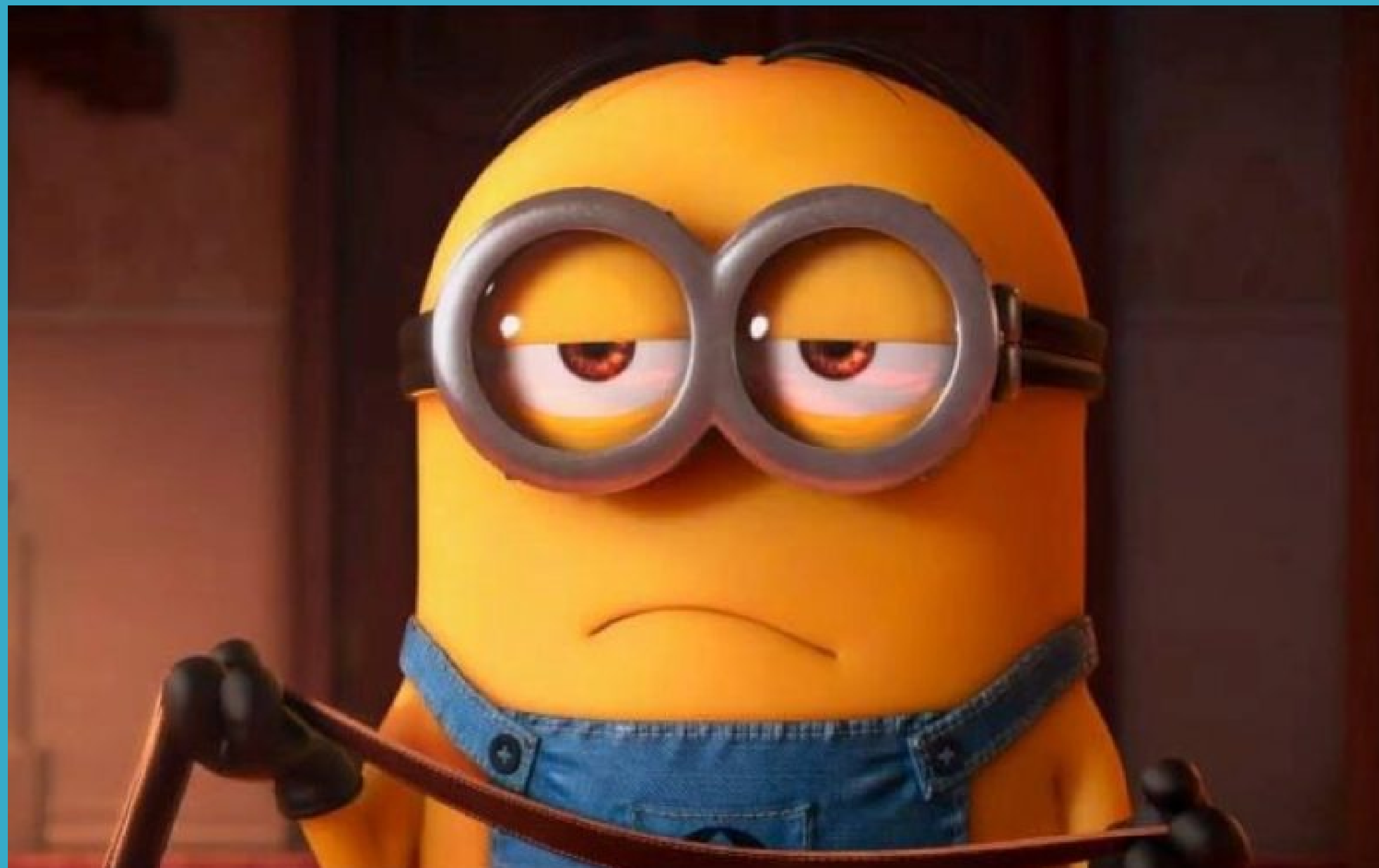
Here Comes Traefik!



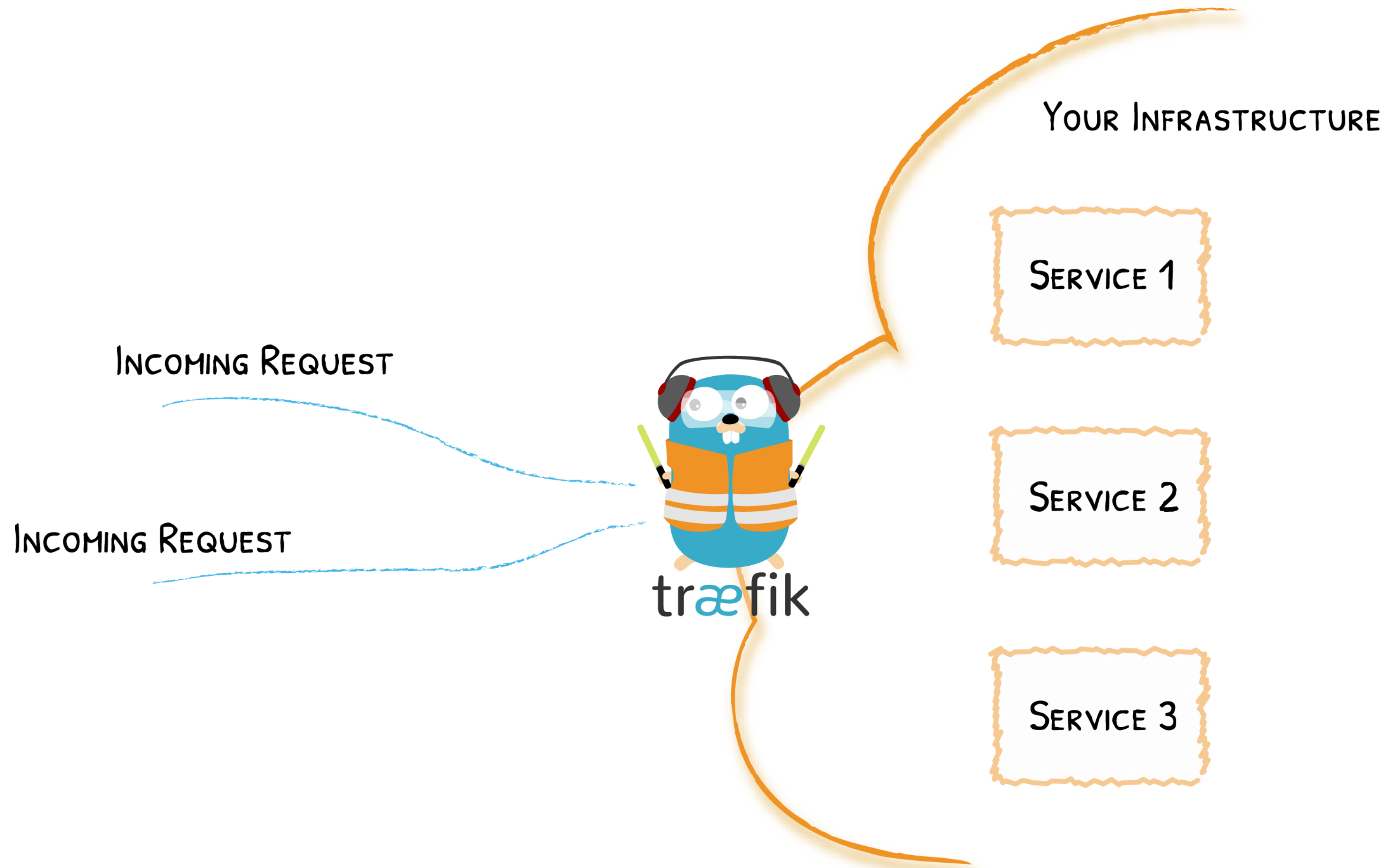
Traefik Project

-  <https://github.com/containous/traefik>
- MIT License
- Written in Go
- 25,000+  1B+  400+ 
- Created in 2015, 4Y 
- Current stable branch: v2.0

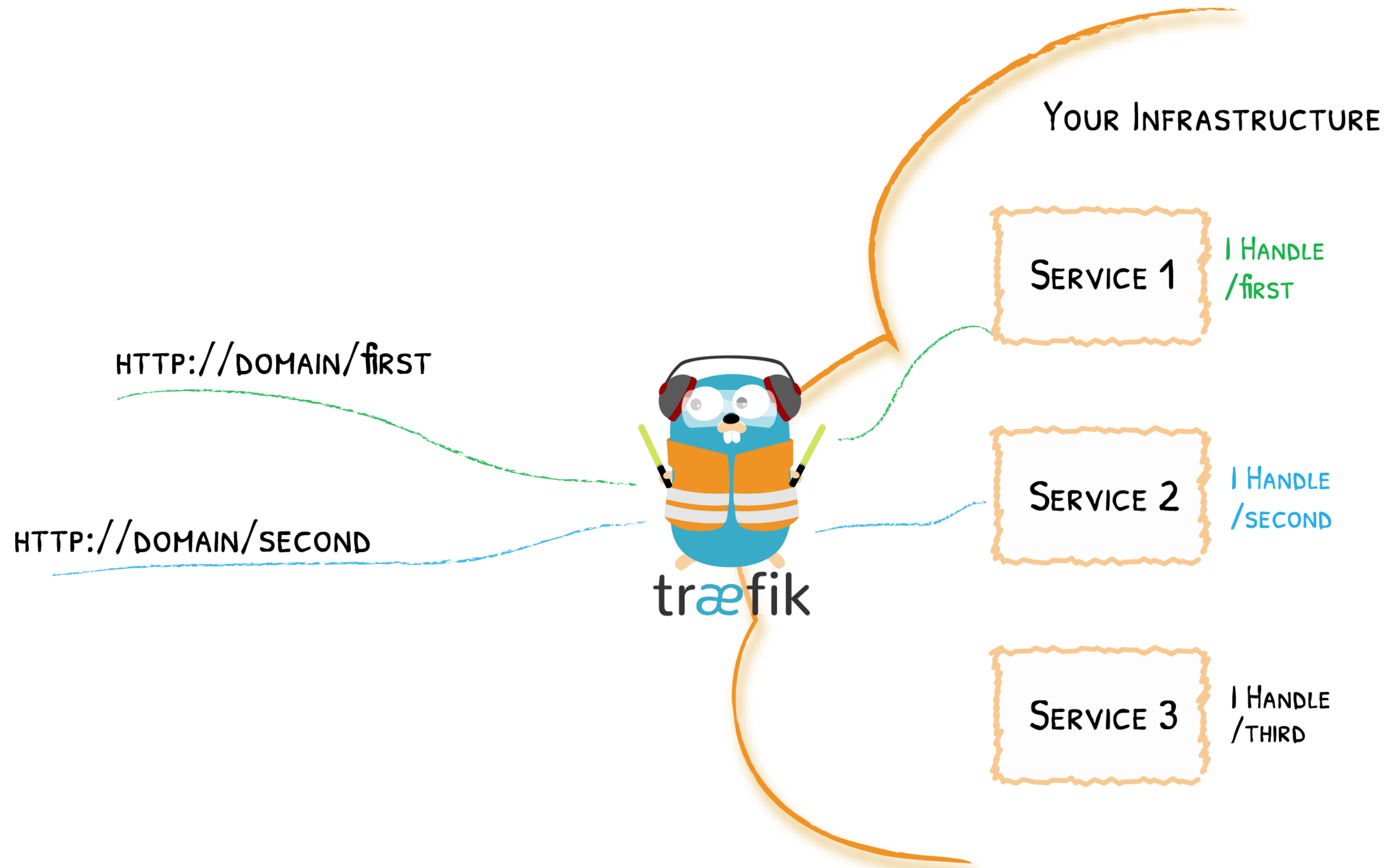
Traefik (V2.0) Core Concepts



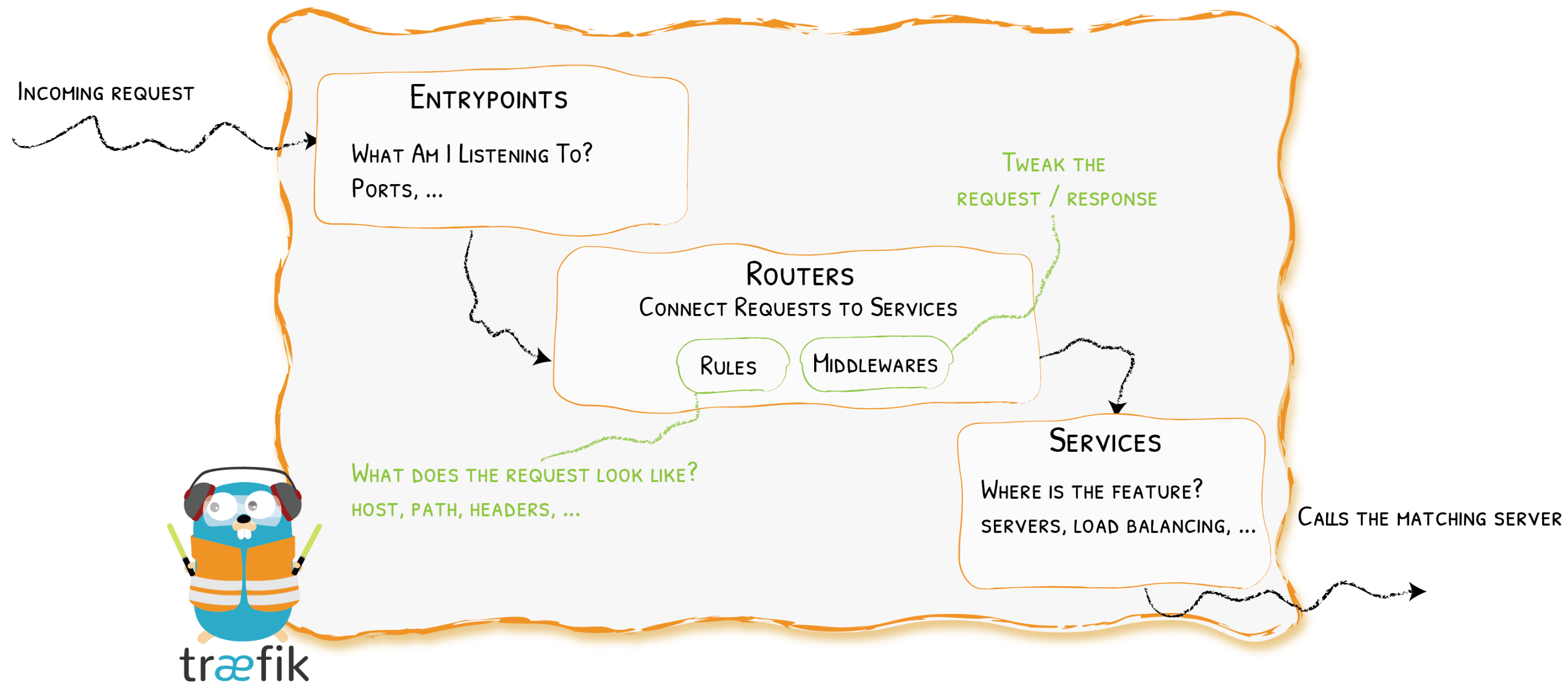
Traefik Is An Edge Router



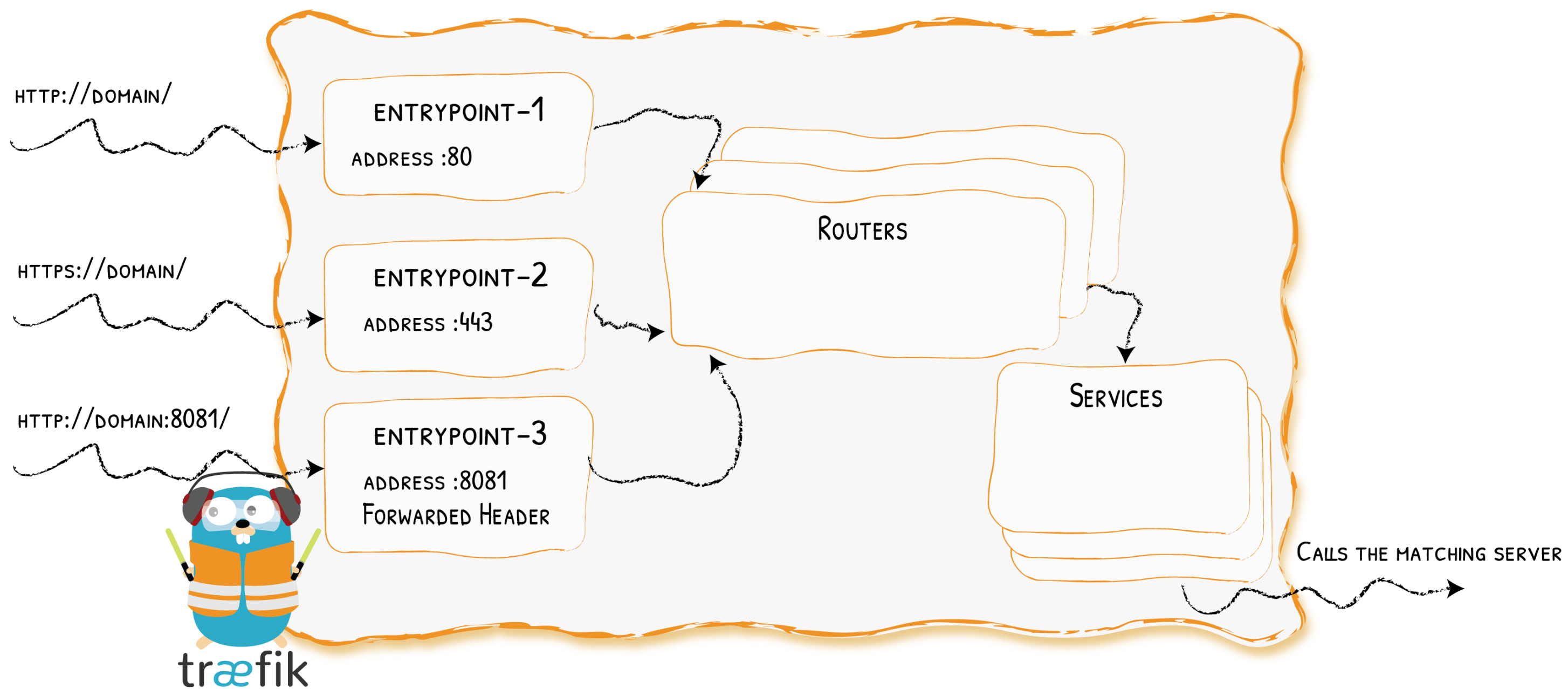
Dynamically Discovers Services



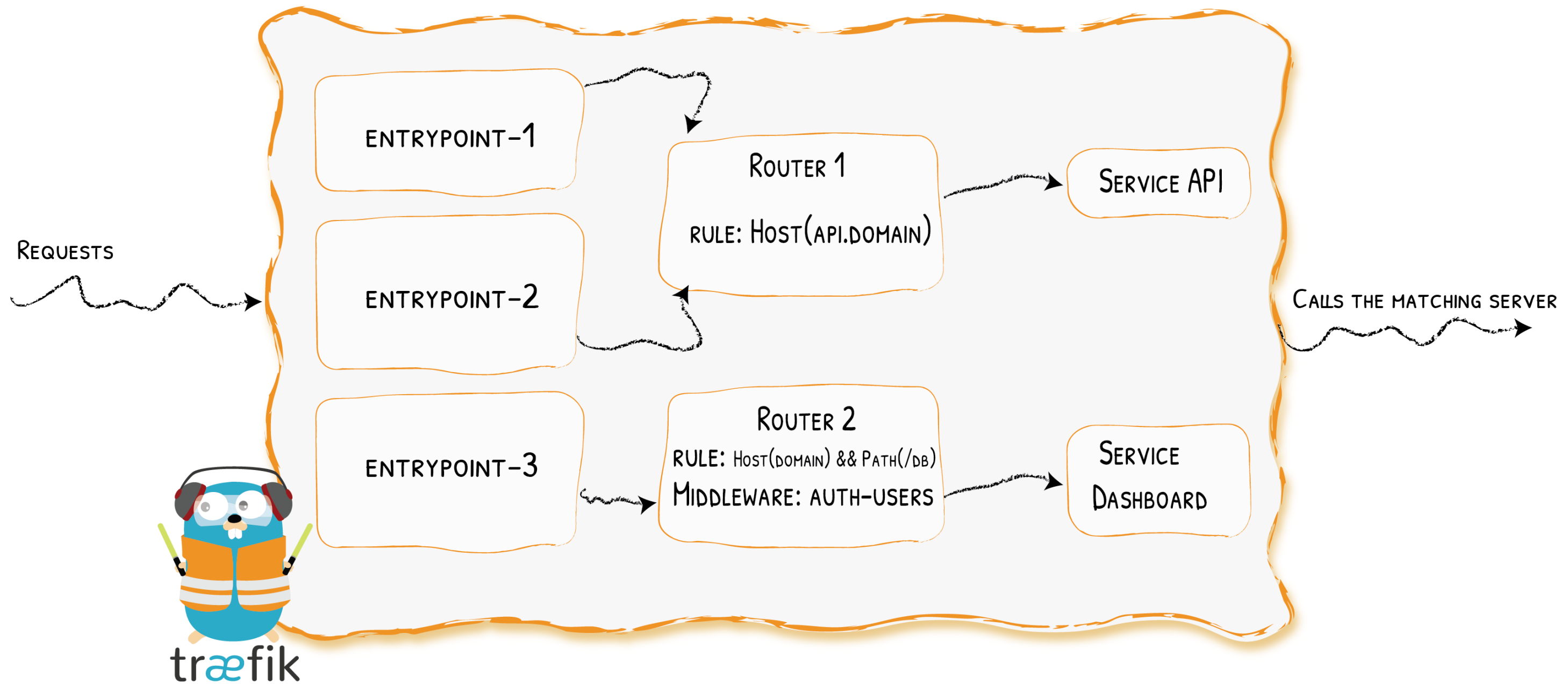
Architecture (V2.0) At A Glance



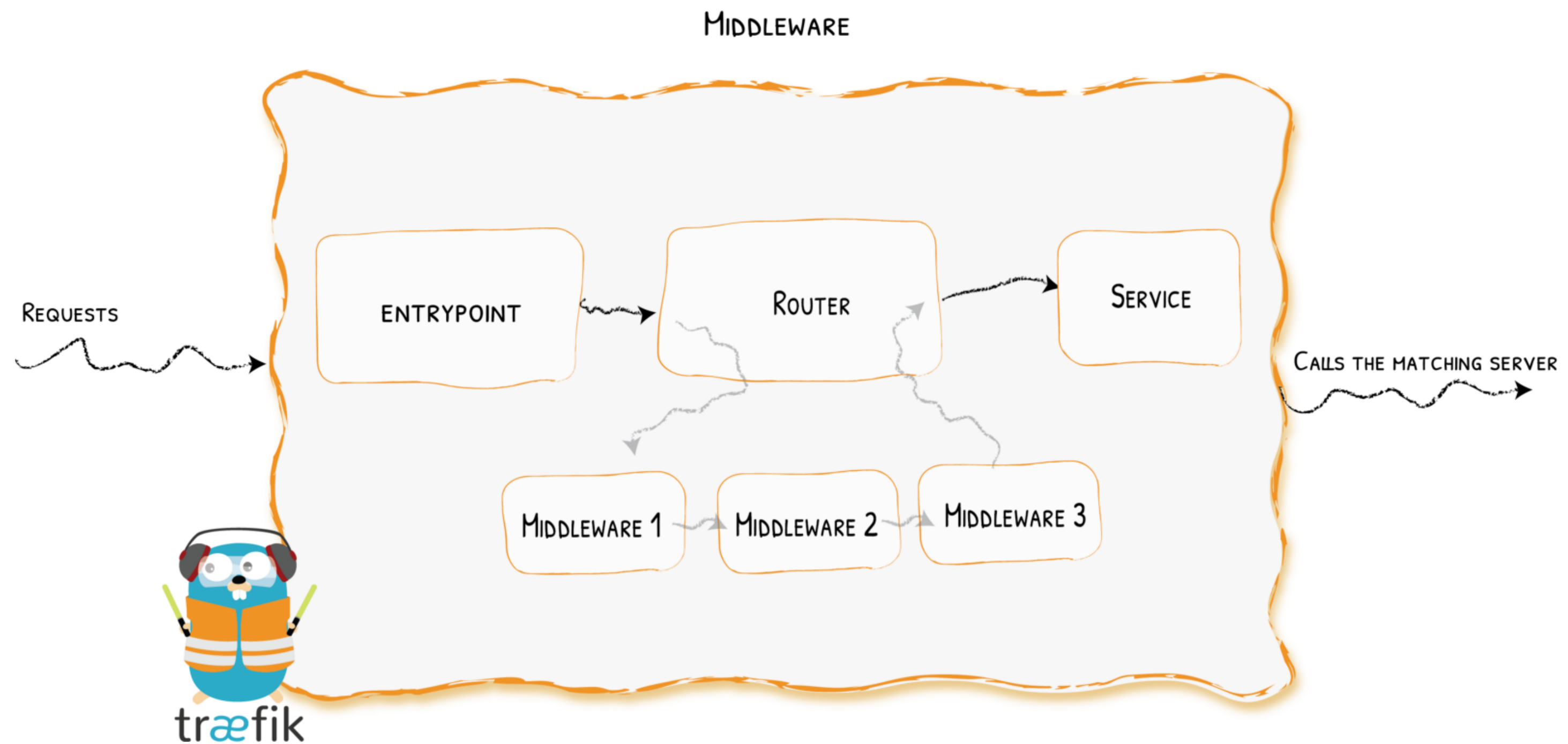
Entrypoints



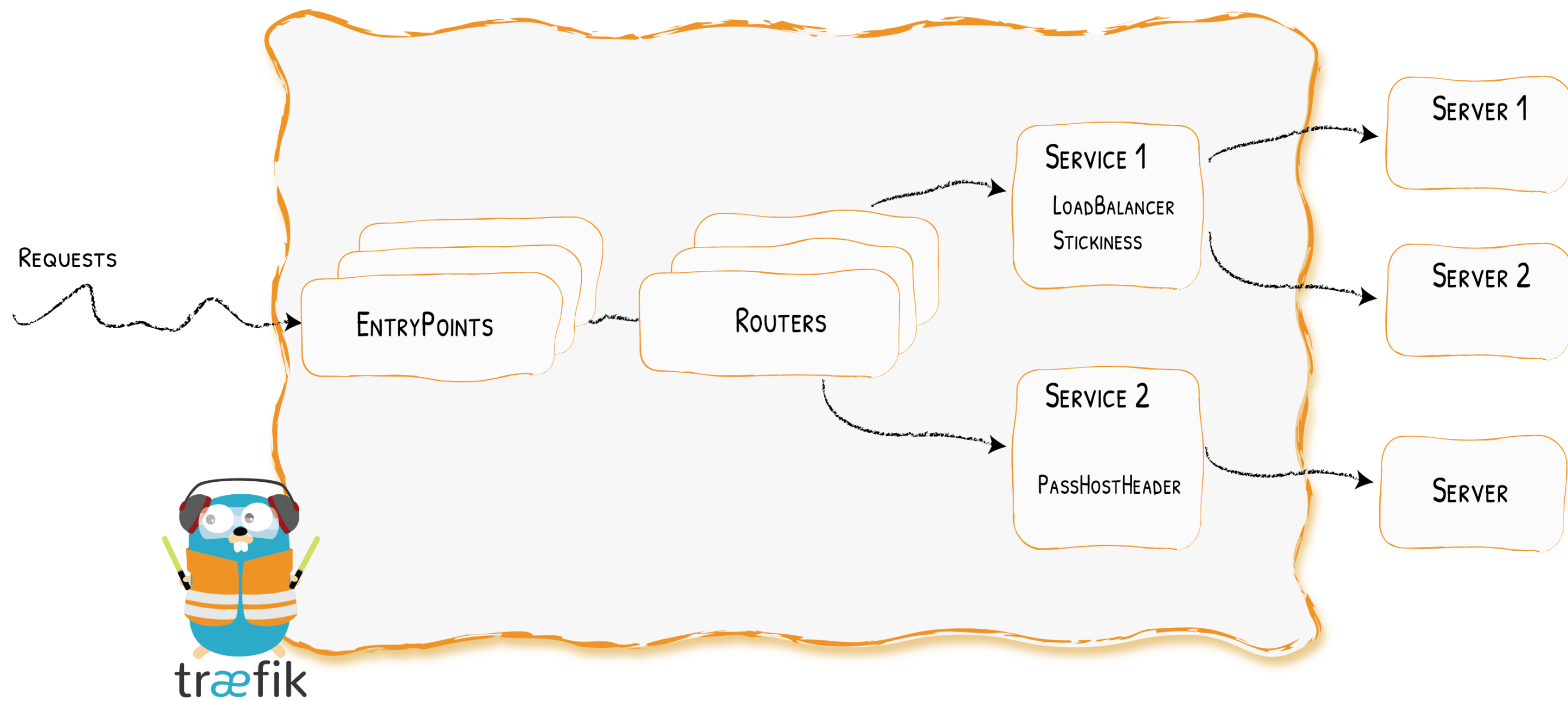
Routers



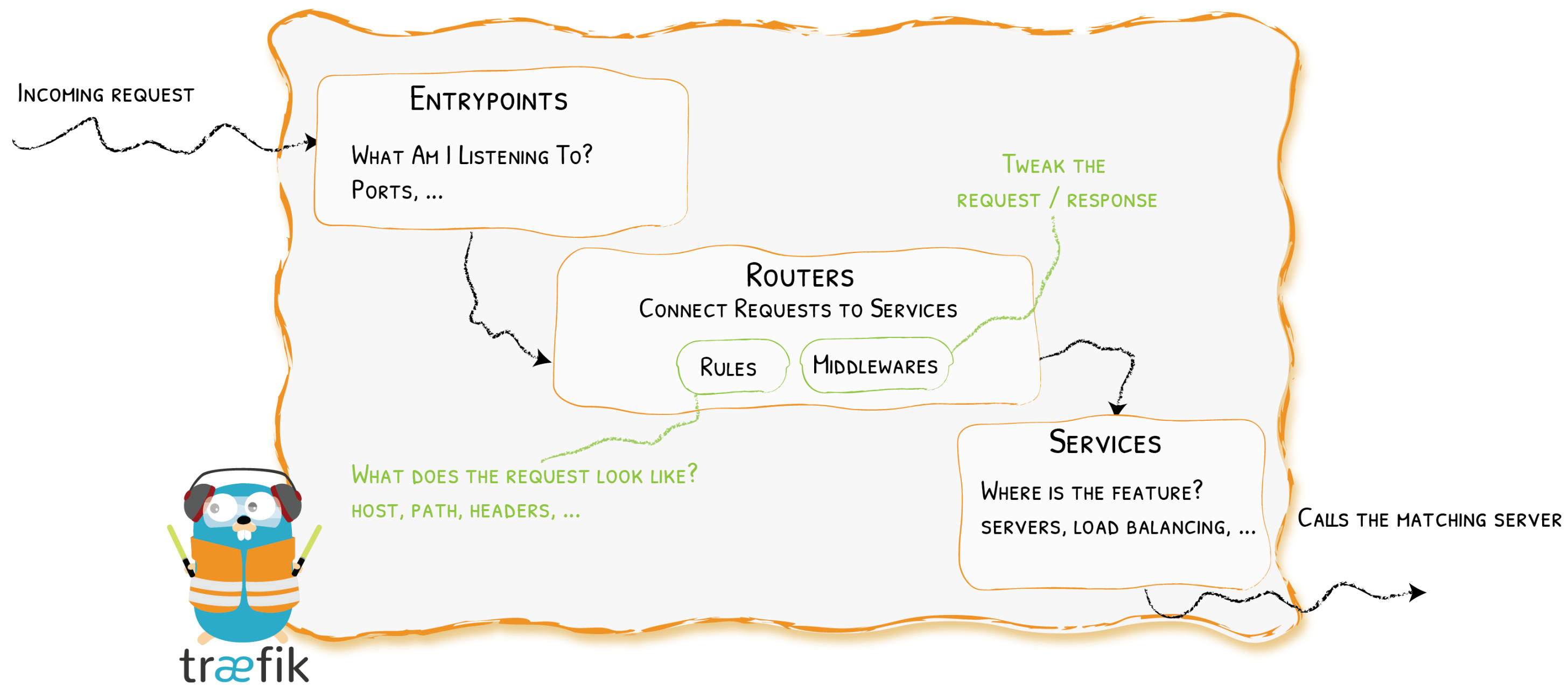
Middlewares



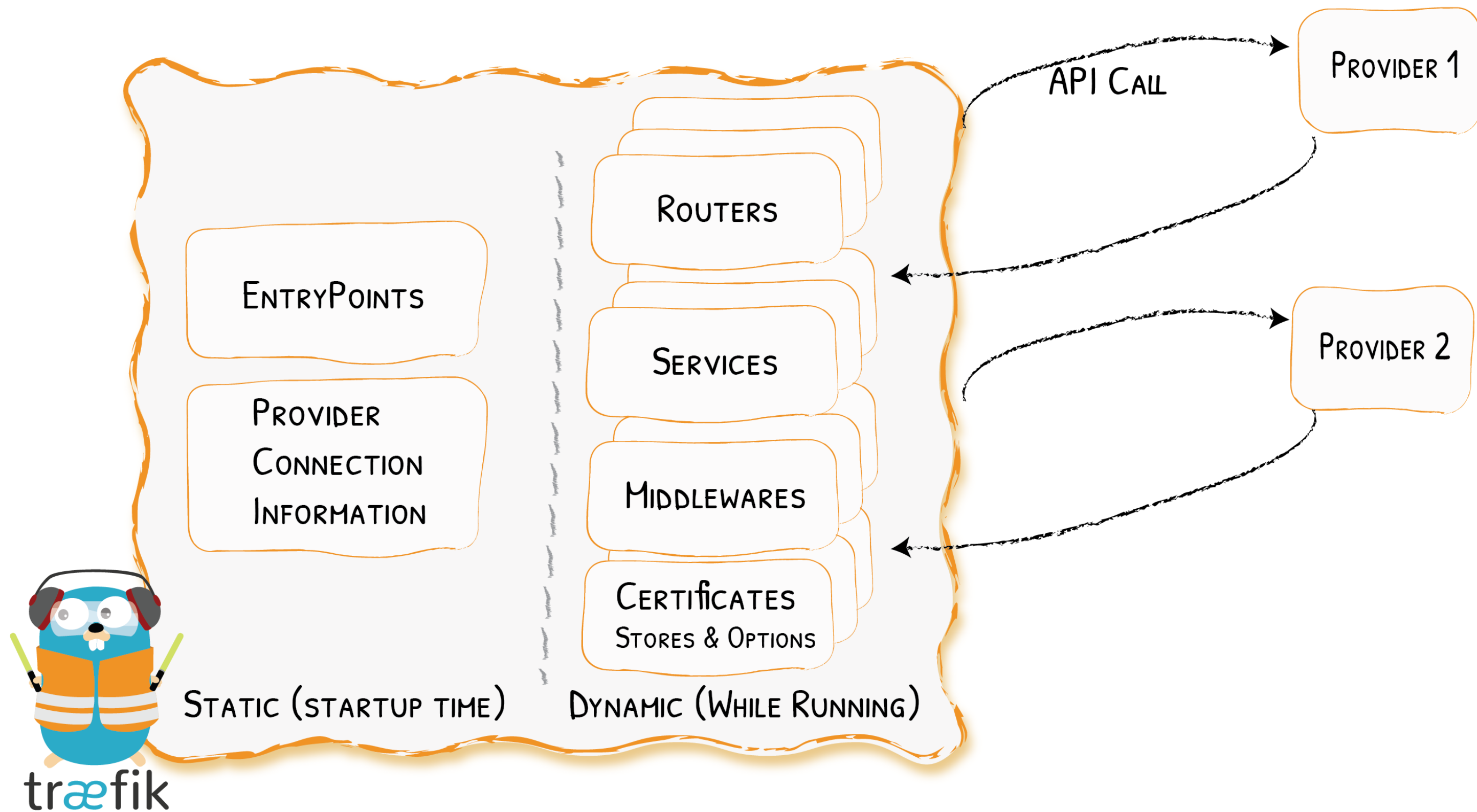
Services



Architecture (Again) At A Glance

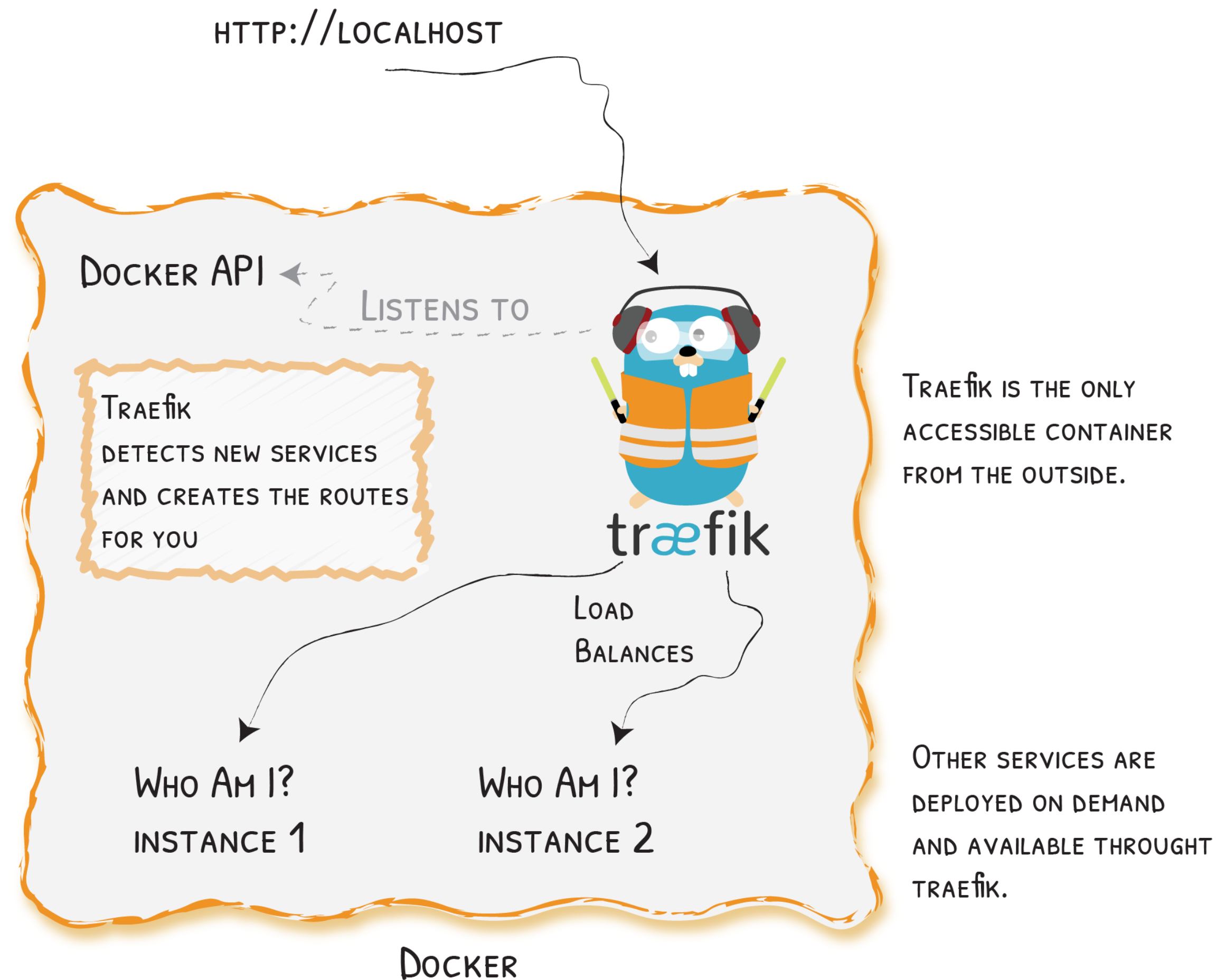


Static & Dynamic Configuration



Show Me The Configuration!

Simple Example With



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

```
# http://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

```
# http://mycompany.org/gitserver -> http://[container IP]:3000
gitserver:
  image: gitea/gitea
  labels:
    - "traefik.http.routers.gitserver.rule=Host(`mycompany.org`) && PathPrefix(`/gitserver`)"
    - "traefik.http.middlewares.gitserver-stripprefix.stripprefix.prefixes=/gitserver"
    - "traefik.http.routers.gitserver.middlewares=gitserver-stripprefix"
```

With File Configuration

Canary Releases

```
http:
  services:
    canary:
      weighted:
        services:
          - name: appv1
            weight: 3 # 75%
          - name: appv2
            weight: 1 # 25%
    appv1:
      loadBalancer:
        servers:
          - url: "http://private-ip-server-1/"
    appv2:
      loadBalancer:
        servers:
          - url: "http://private-ip-server-2/"
```

Traefik With

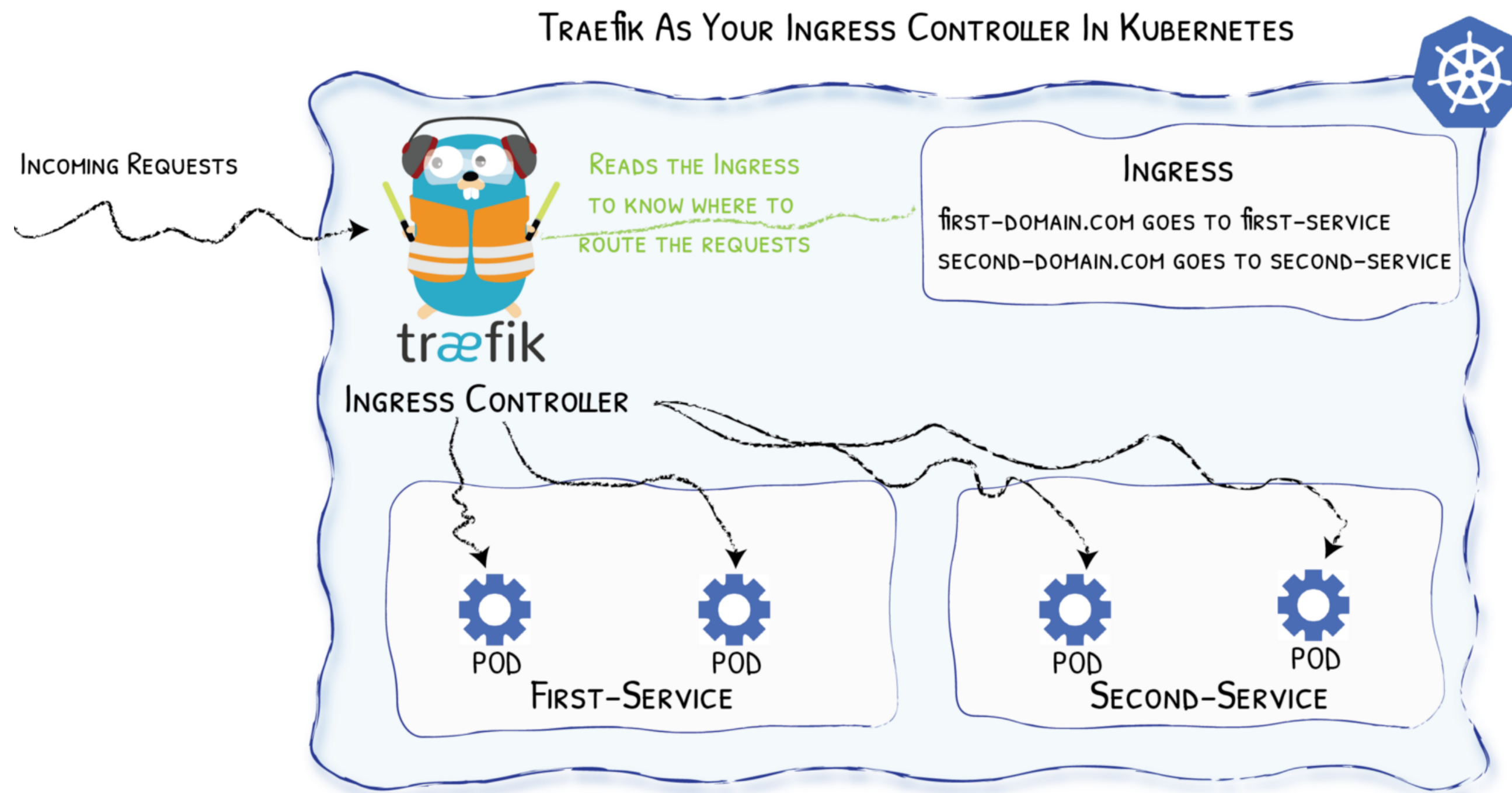


Diagram from <https://medium.com/@geraldcroes>

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
```

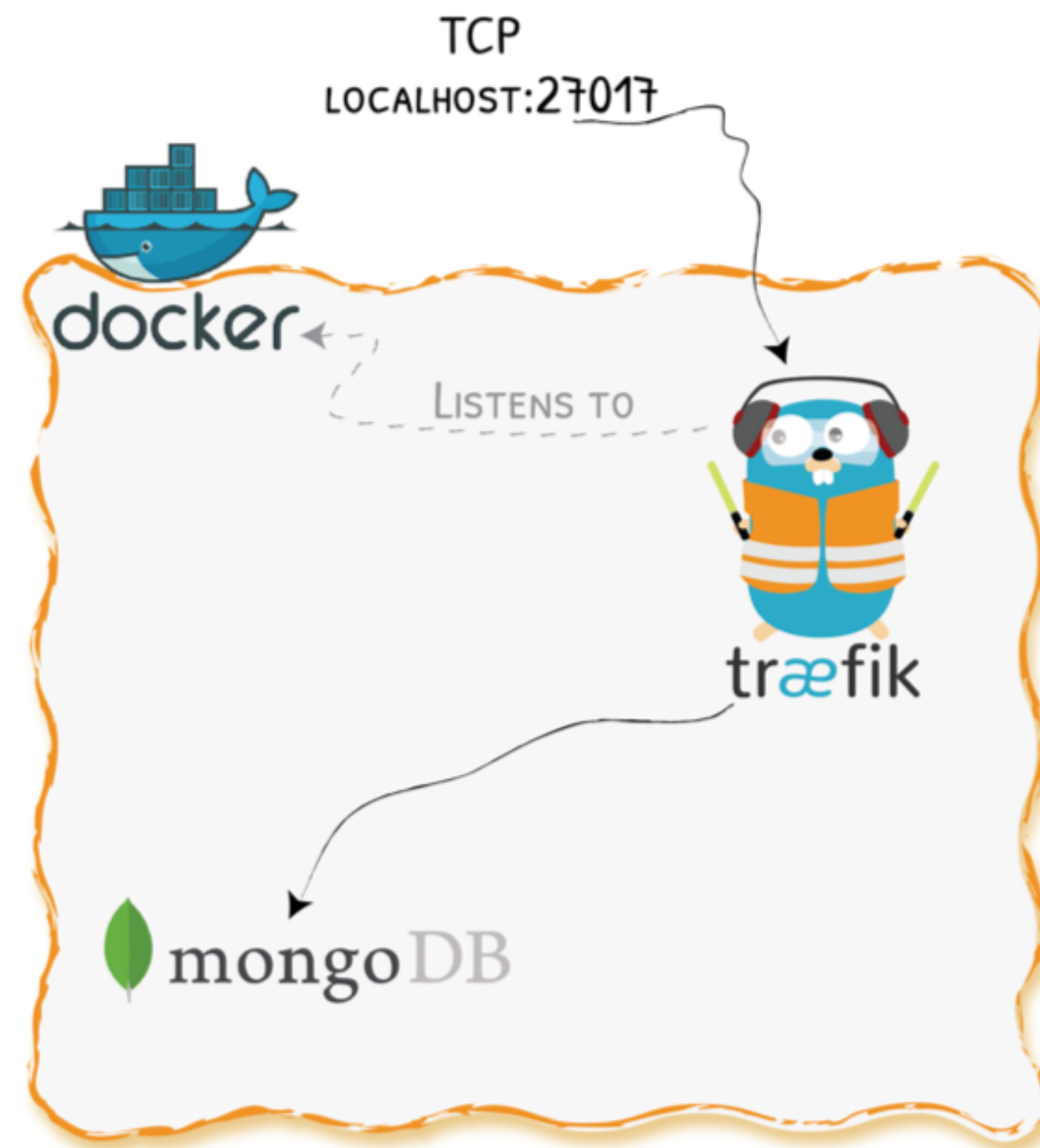
& TCP (With CRD)

```
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
```

Demo

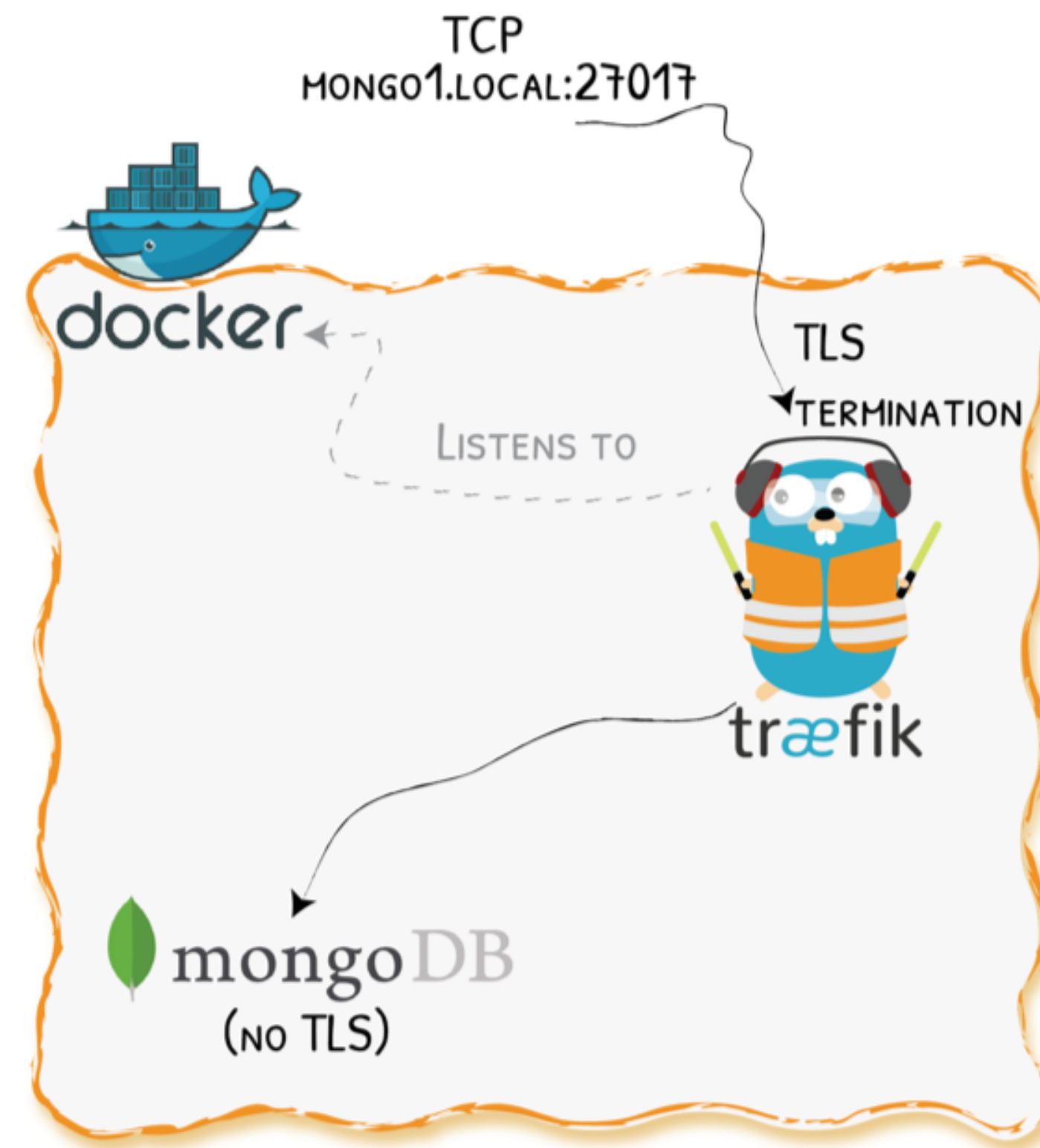


Demo 1 - TCP



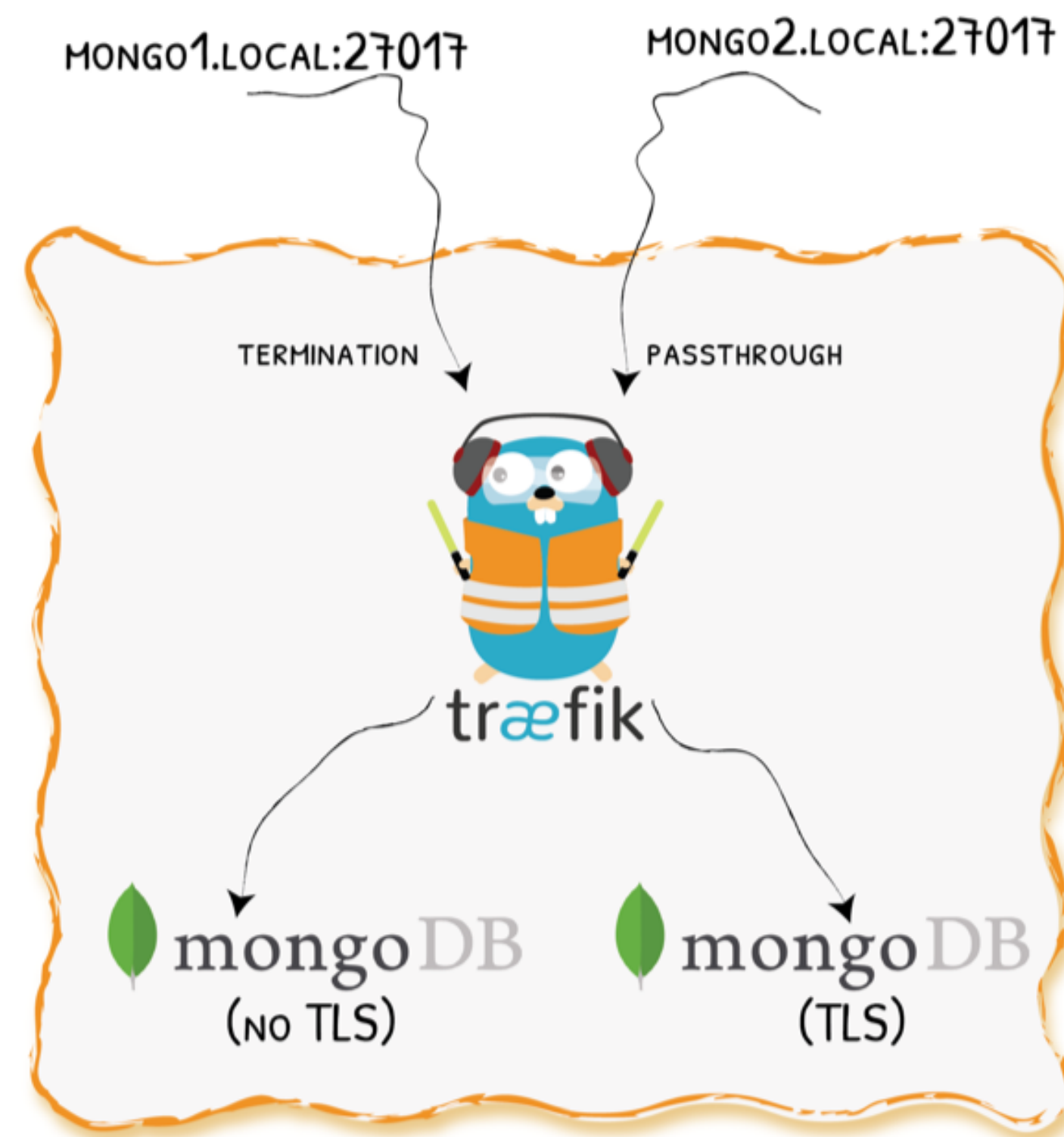
Demo Code on [GitHub](#)

Demo 2 - TCP + TLS



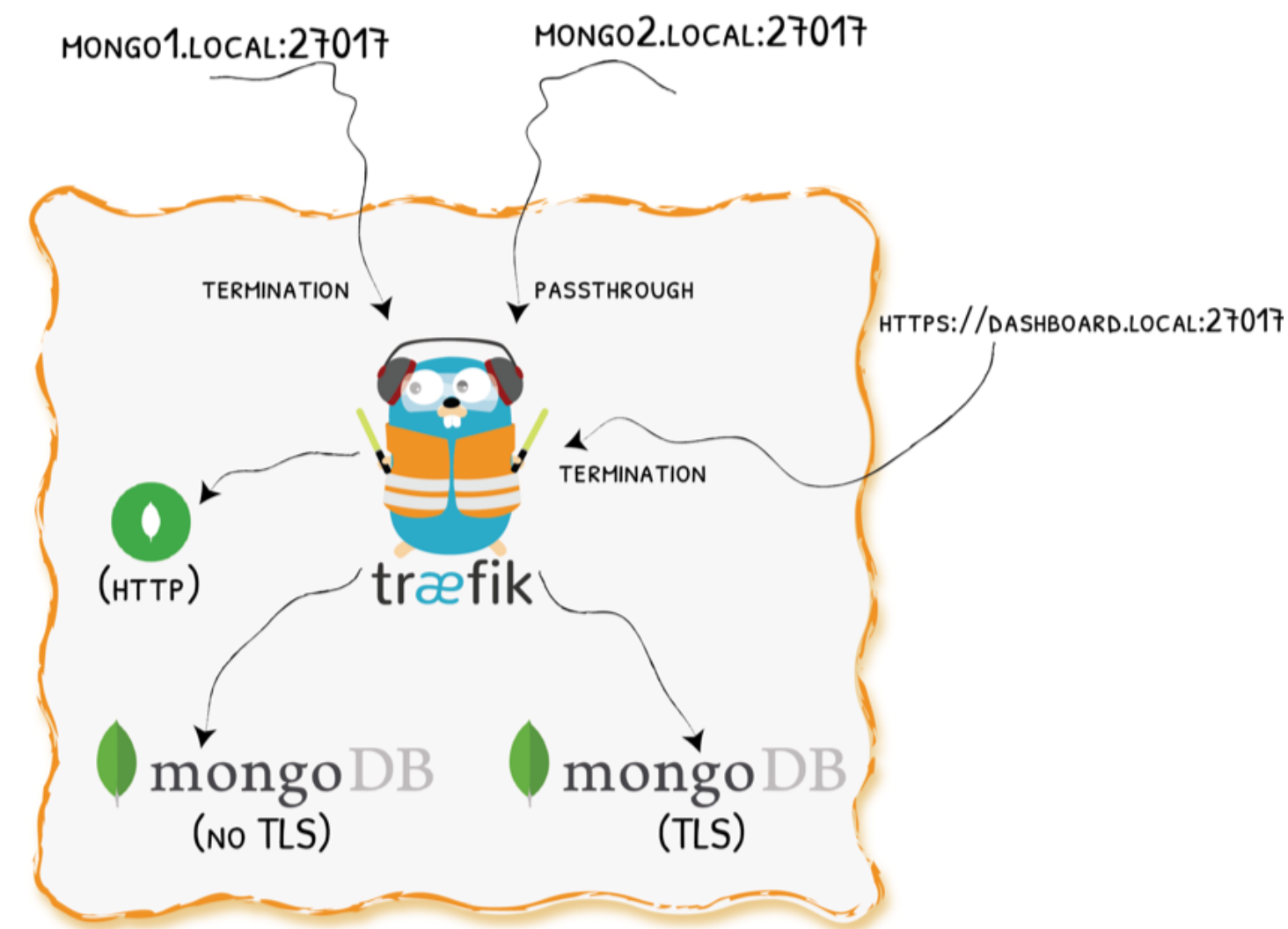
Demo Code on [GitHub](#)

Demo 3 - SNI Routing + TLS Passthrough For TCP



Demo Code on [GitHub](#)

Demo 4 - Muxing HTTPS And TCP On The Same Port



Demo Code on [GitHub](#)

Demo 5 - Canary Release Of A WebApp



Demo Code on [GitHub](#)

What Next

- KV Store
- UDP
- Canary and Mirroring With K8s CRD
- And More...

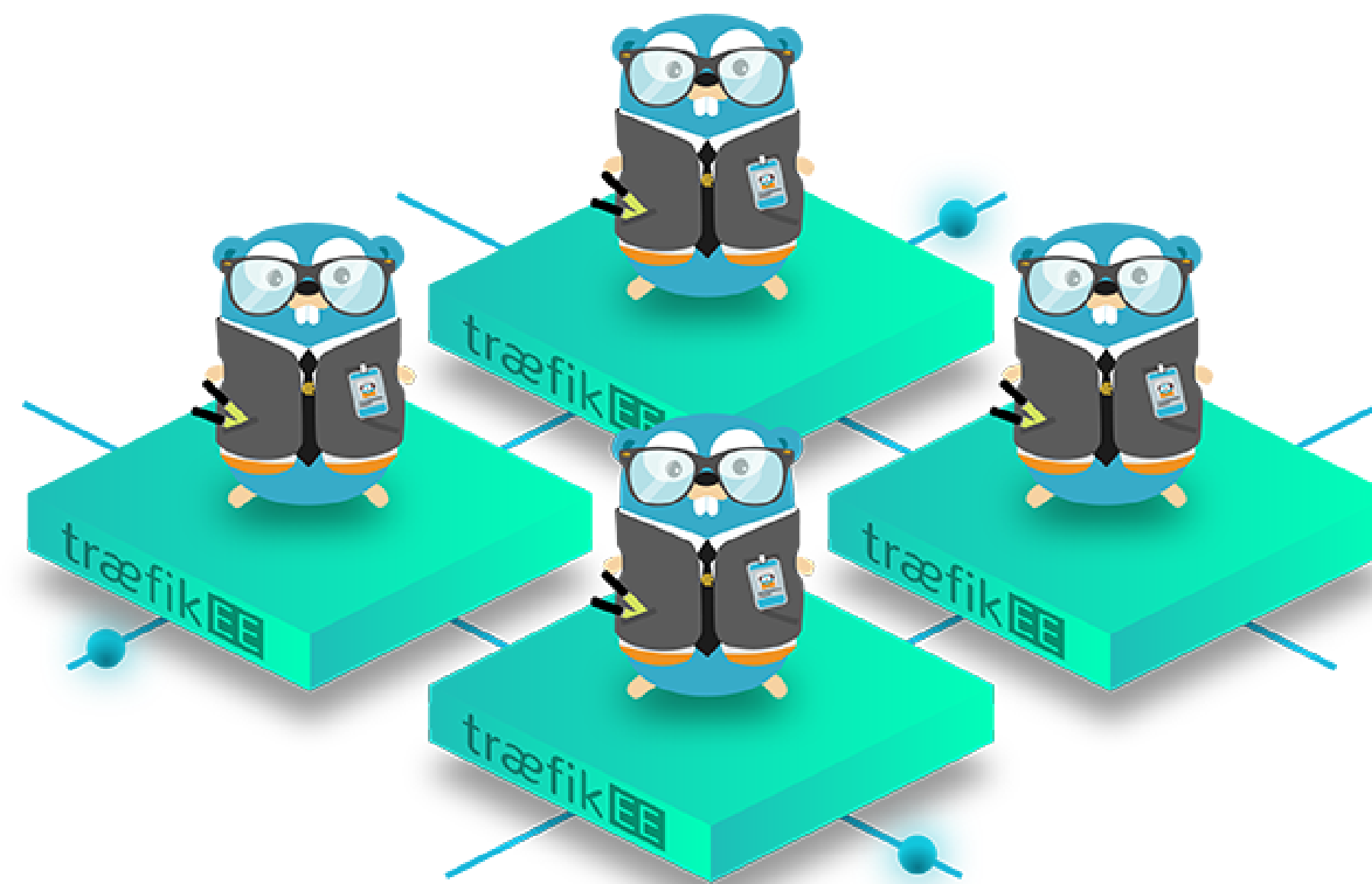
We Also Missed Talking About ...



More Info

- [Back to Traefik 2.0](#)
- [Community Forum](#)
- [Official Documentation](#)

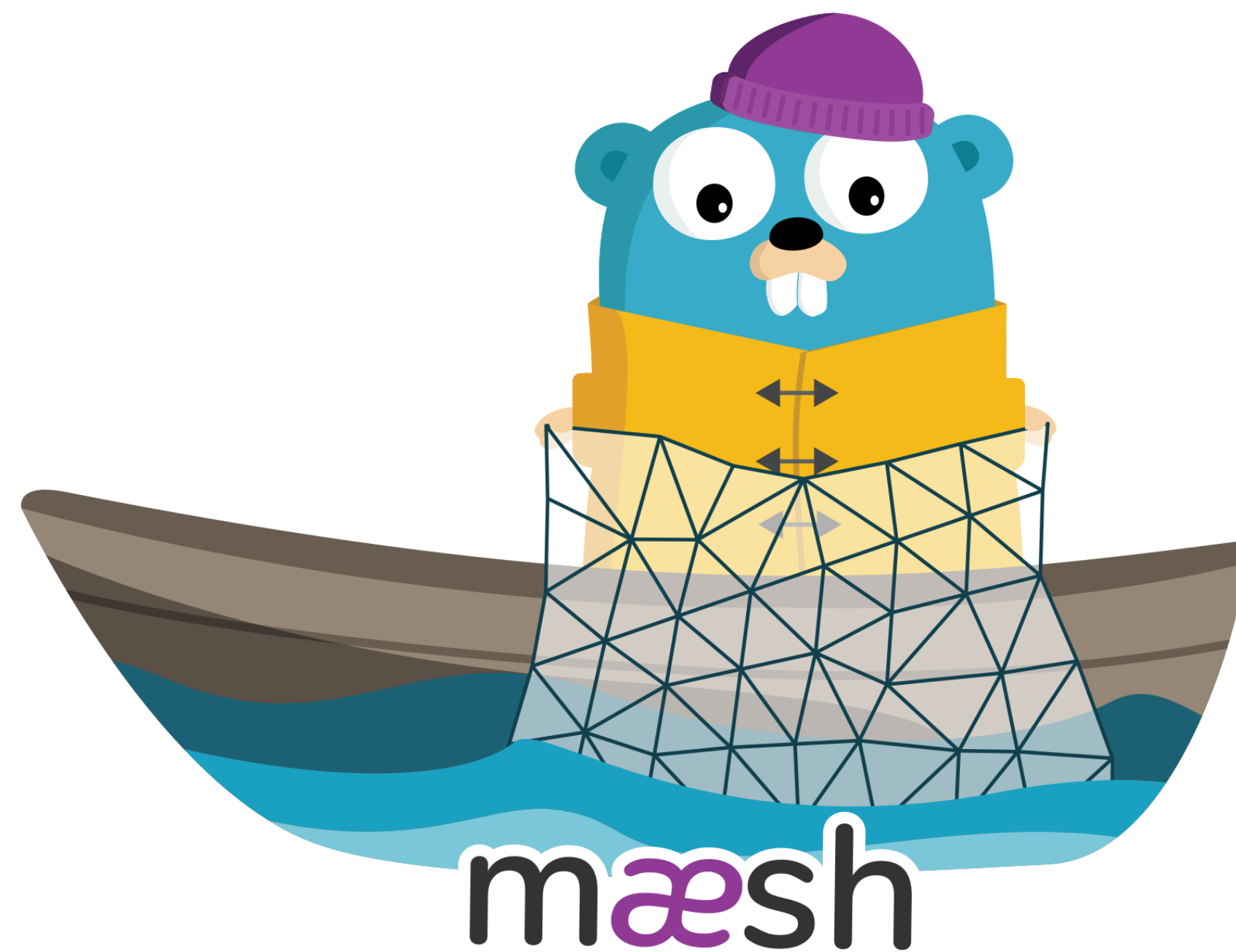
Traefik Comes In Herd



Free Trial

<https://containo.us/traefikee>

Say Hello To Maesh



More On Maesh

<https://mae.sh>

That's All Folks!



We Have
Stickers!

We Are Hiring!



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

Thank You!



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