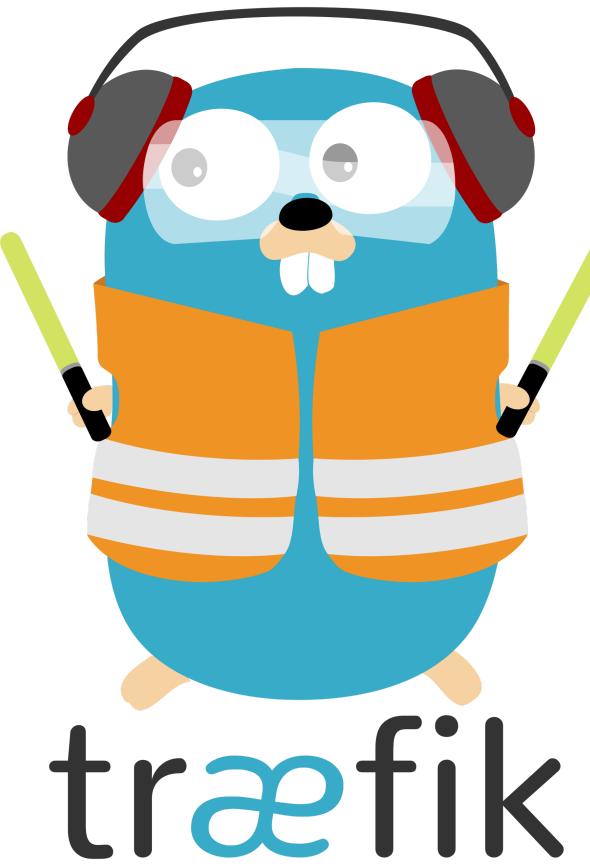


Smart Routing Et HTTPS Pour Tous

Traefik En Action !



<https://containous.github.io/slides>

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

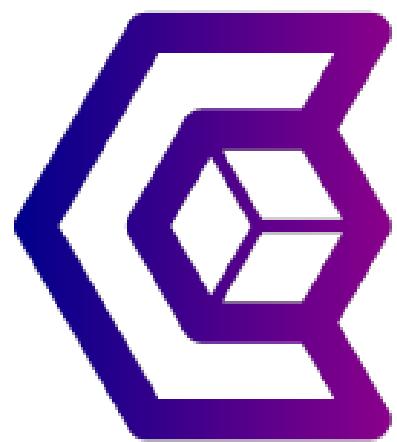
- Damien DUPORTAL:
 - Træfik's Developer  Advocate @ Containous
-  @DamienDuportal
-  dduortal



Containous

<https://containo.us>

- We Believe in Open Source
- We Deliver Traefik
- Commercial Support for Traefik
- 20 people, 90% tech

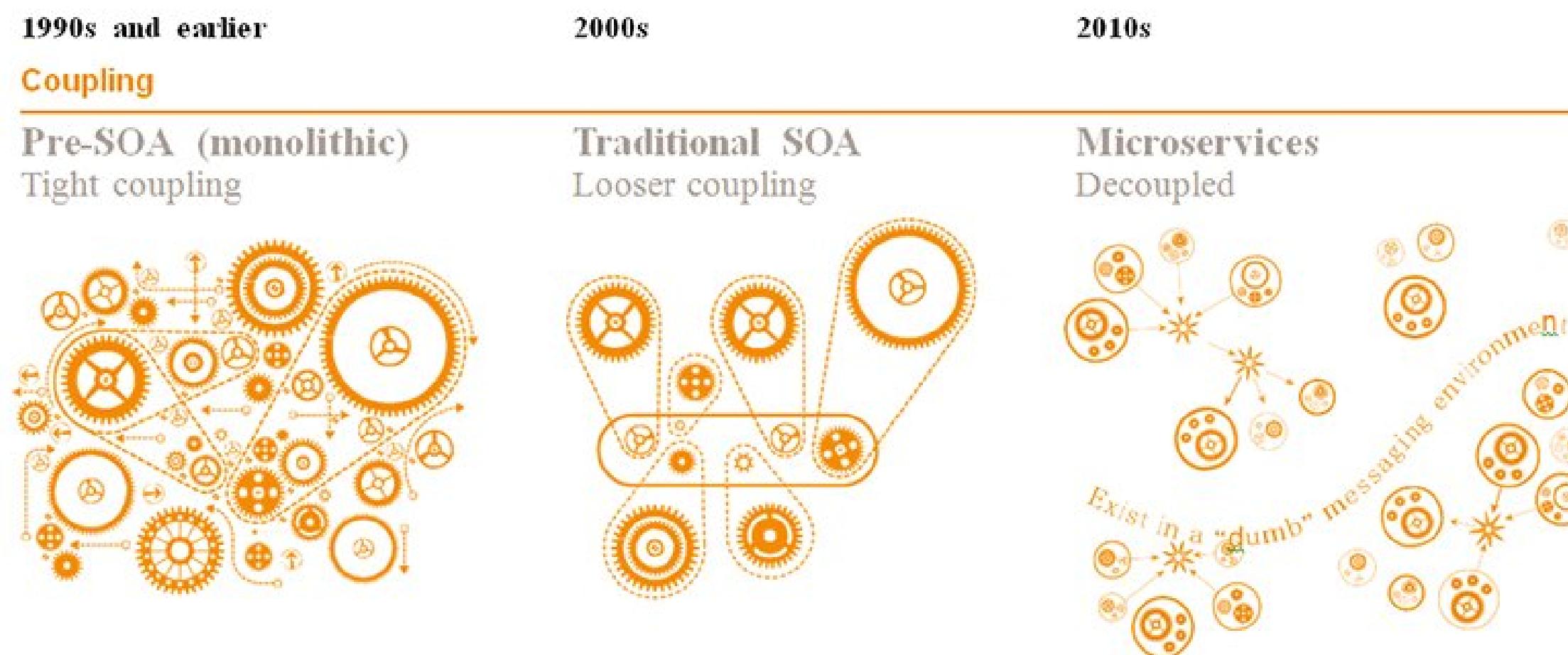


Why Traefik?



Why, Mr Anderson?

Evolution Of Software Design



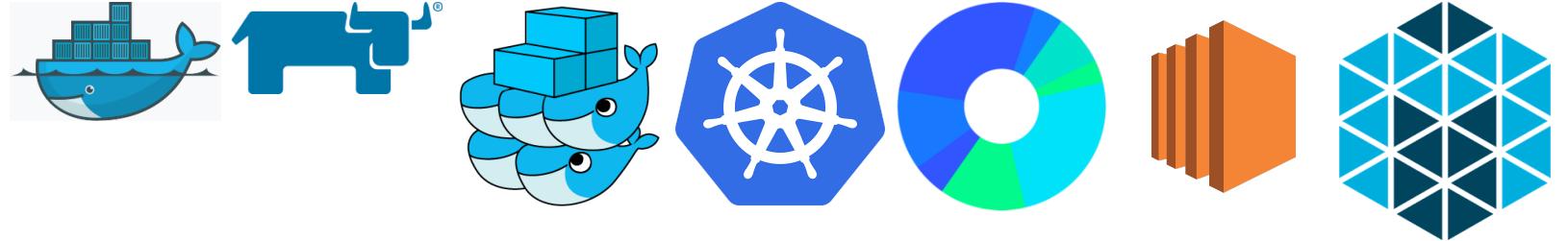
The Premise Of Microservices...



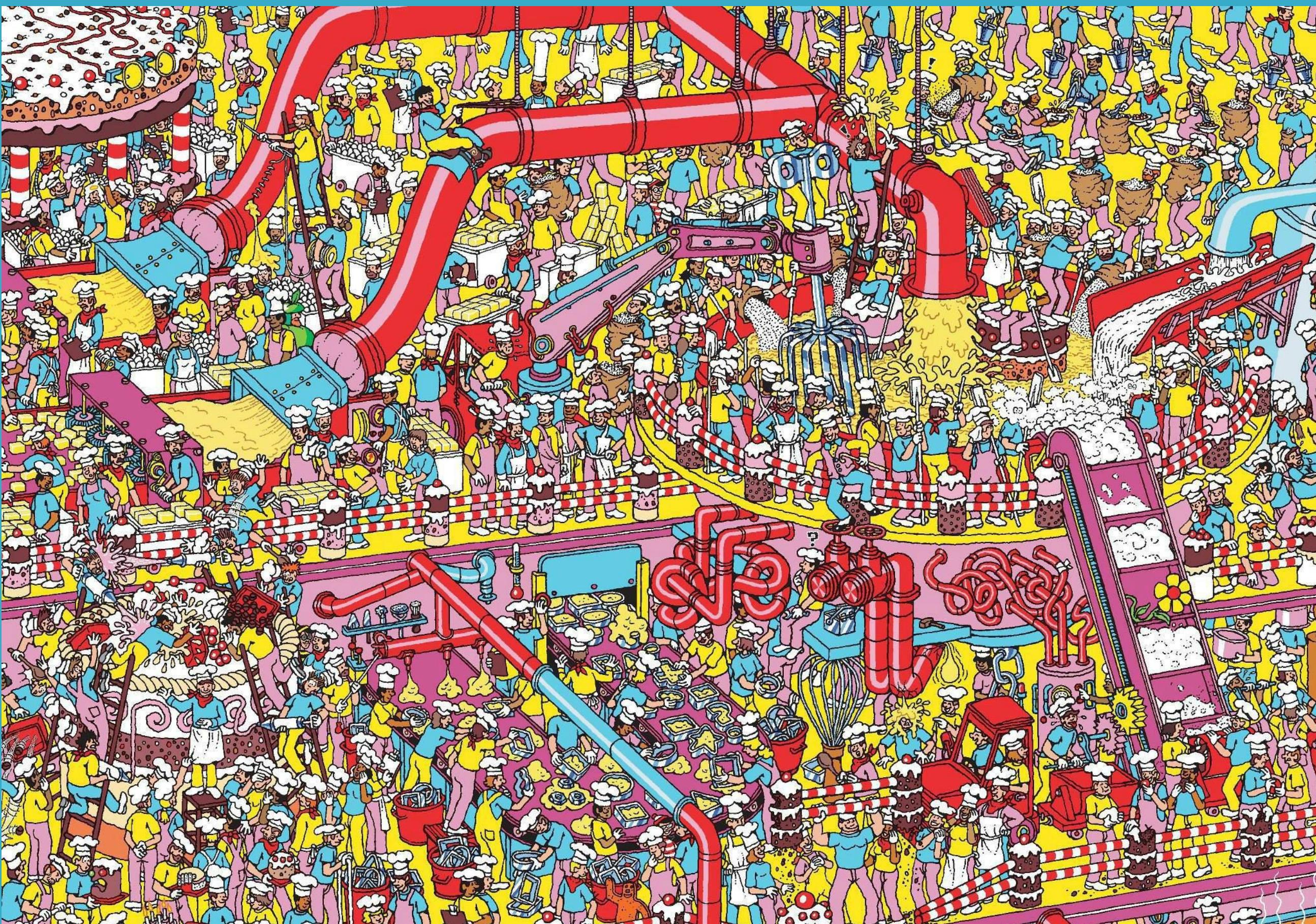
...And What Happens

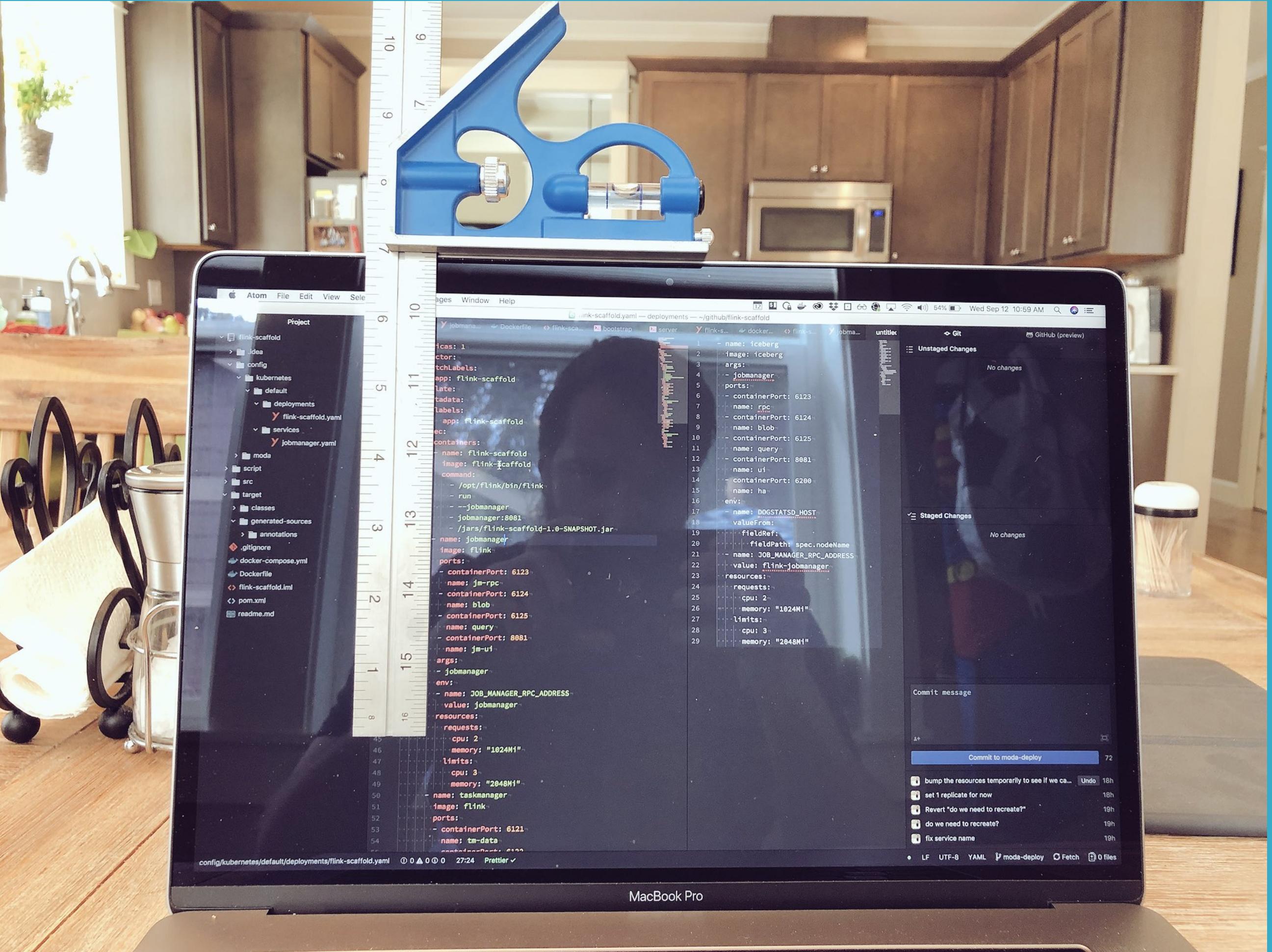


Tools Of The Trade



Where's My Service?





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

What If I Told You?



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

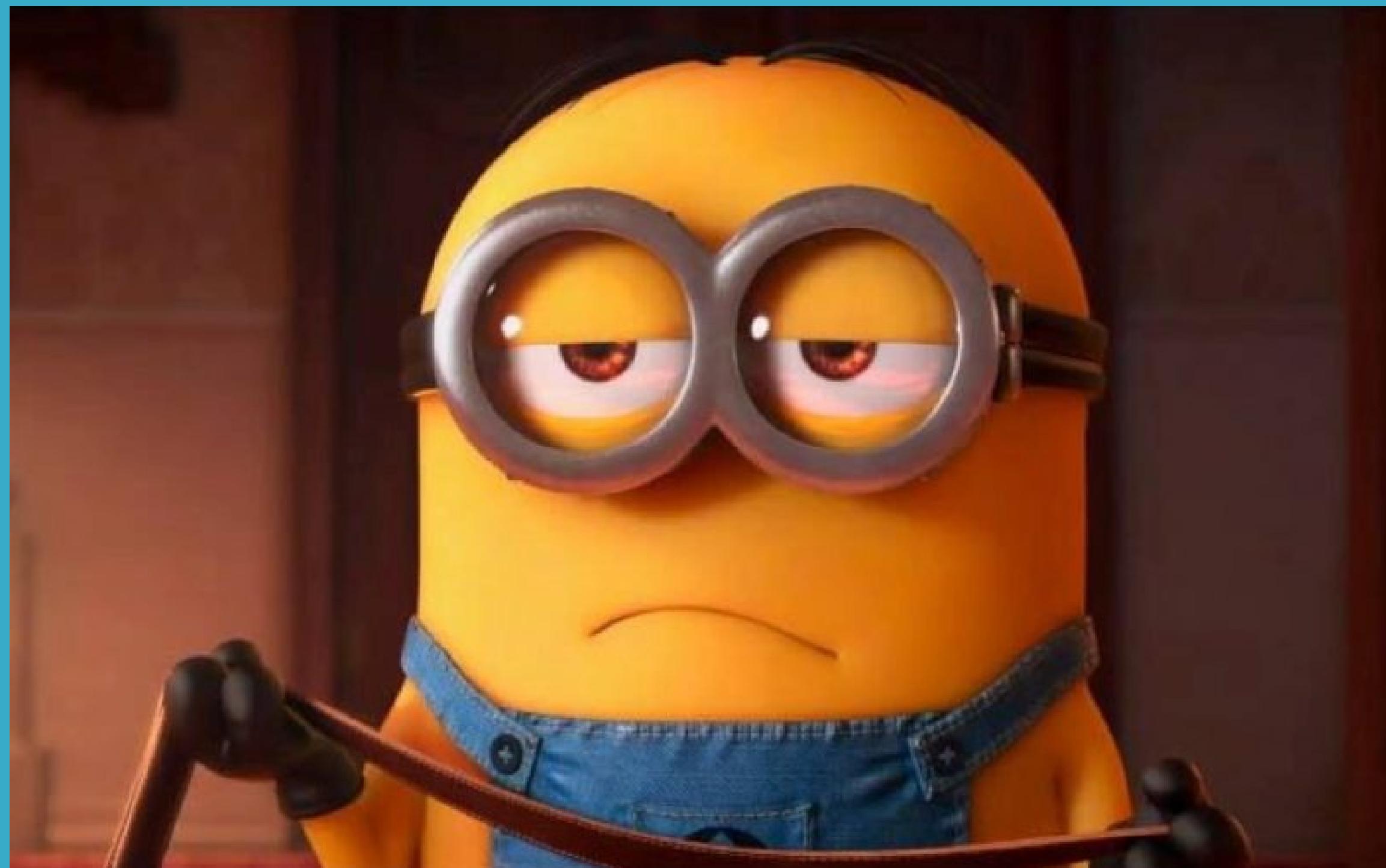
Here Comes Traefik!



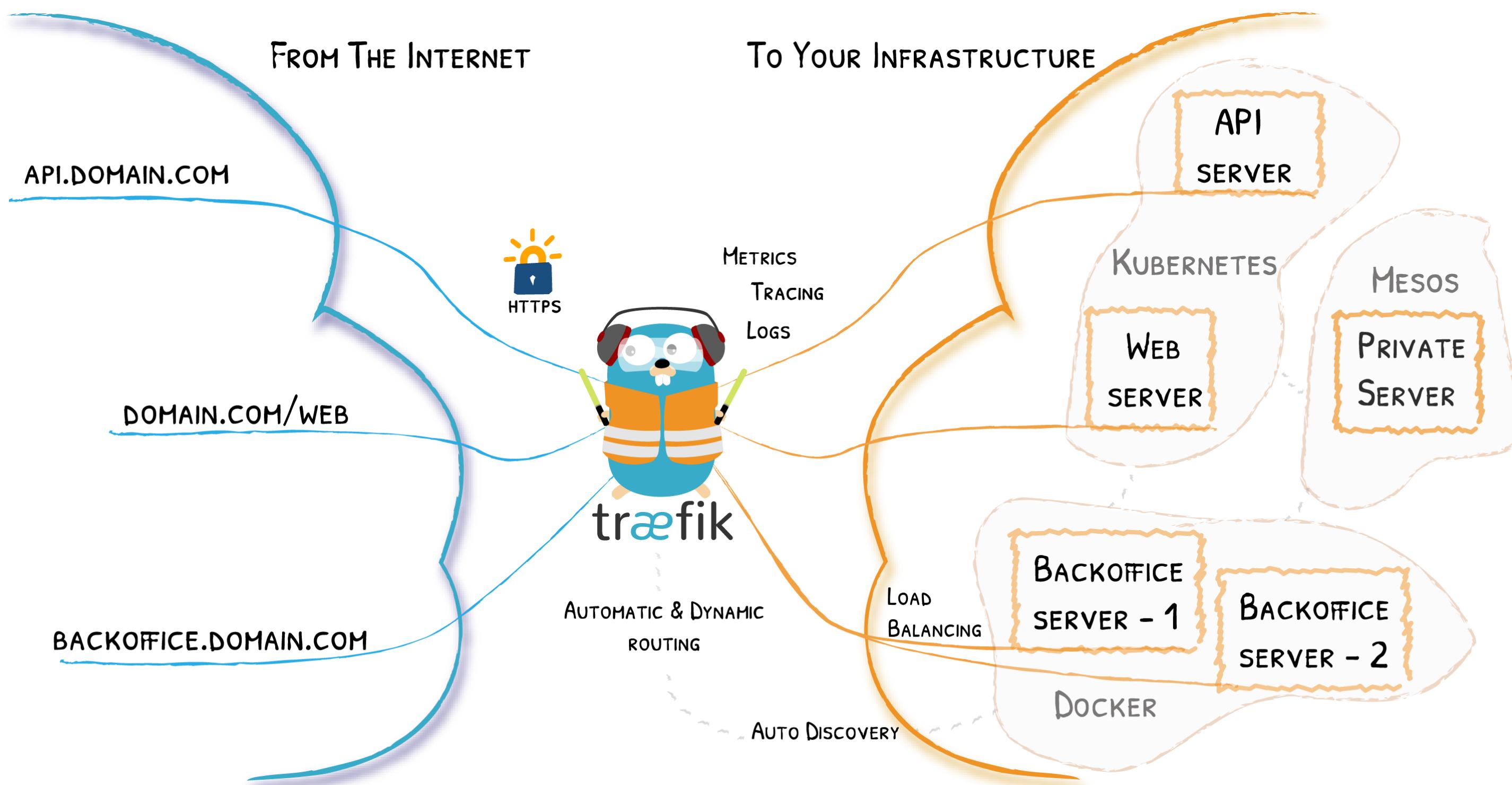
Traefik Project

-  <https://github.com/containous/traefik>
- MIT License
- Written in Go
- 21,000+ 
- 600M+ 
- 350+ 

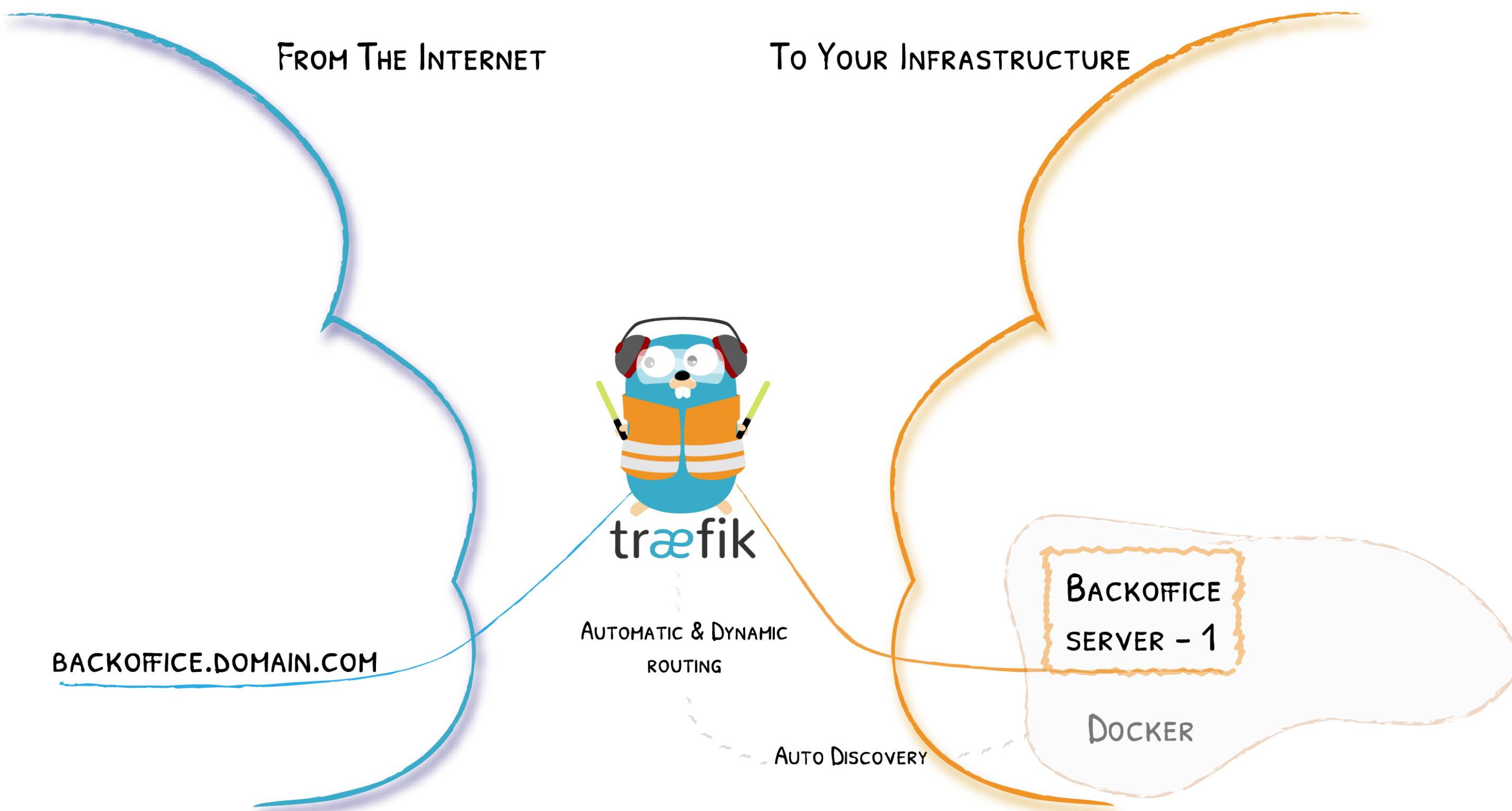
Traefik Core Concepts



Remember The Diagram?



Let's Simplify



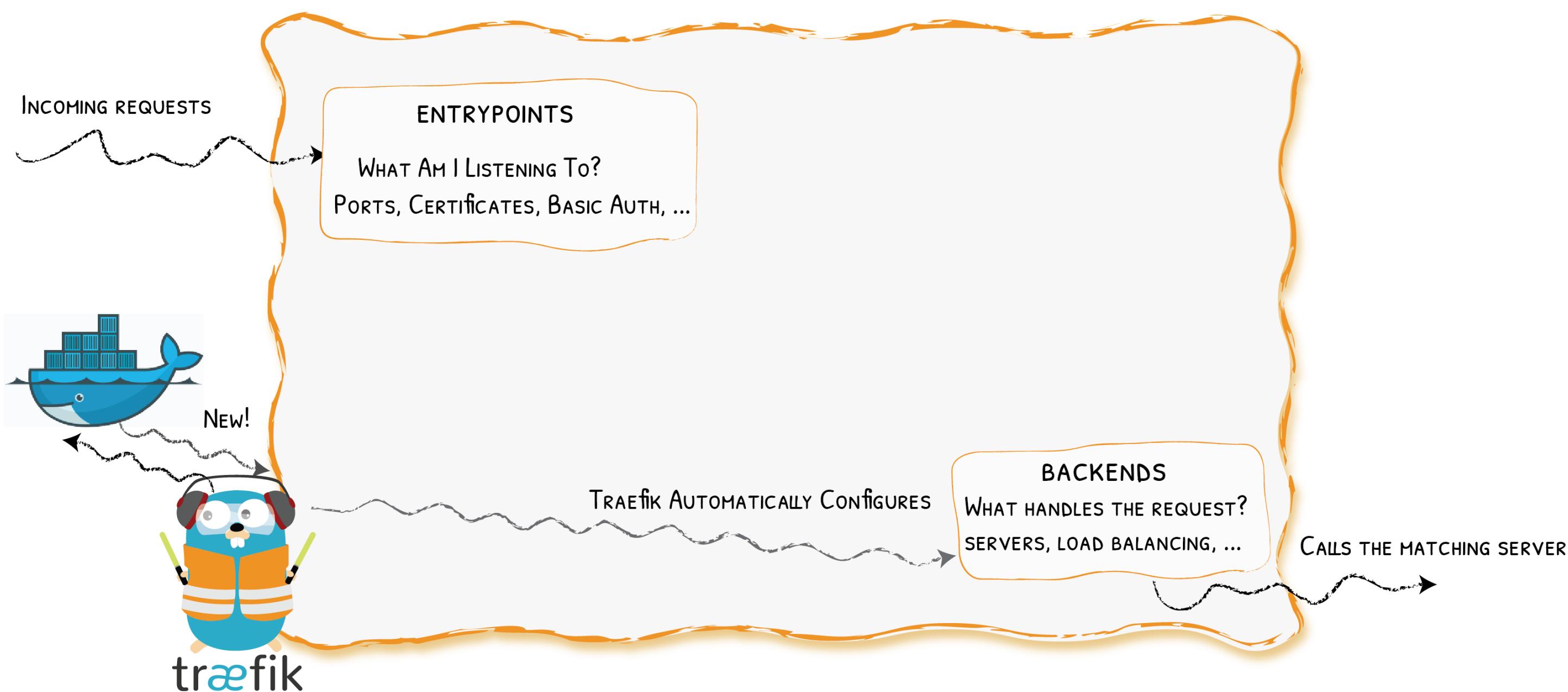
Providers



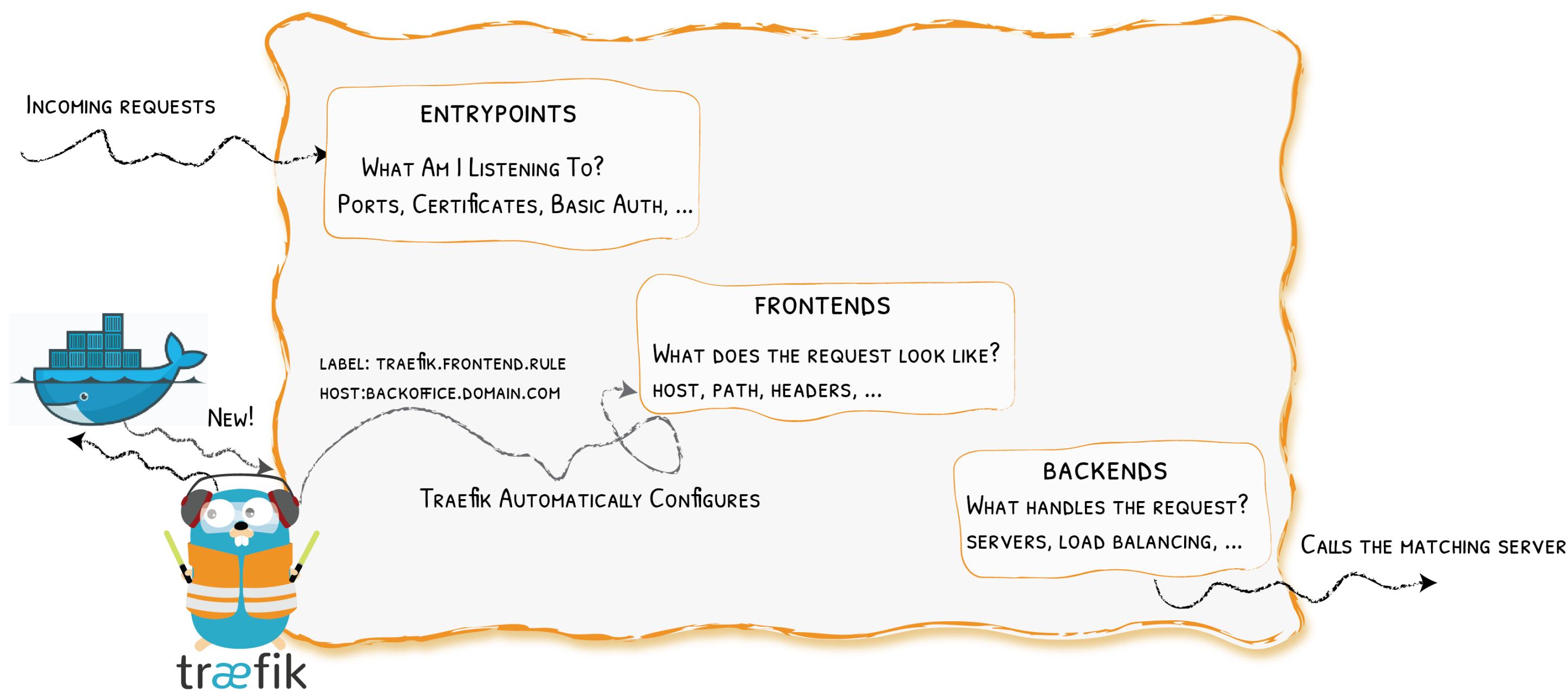
Entrypoints



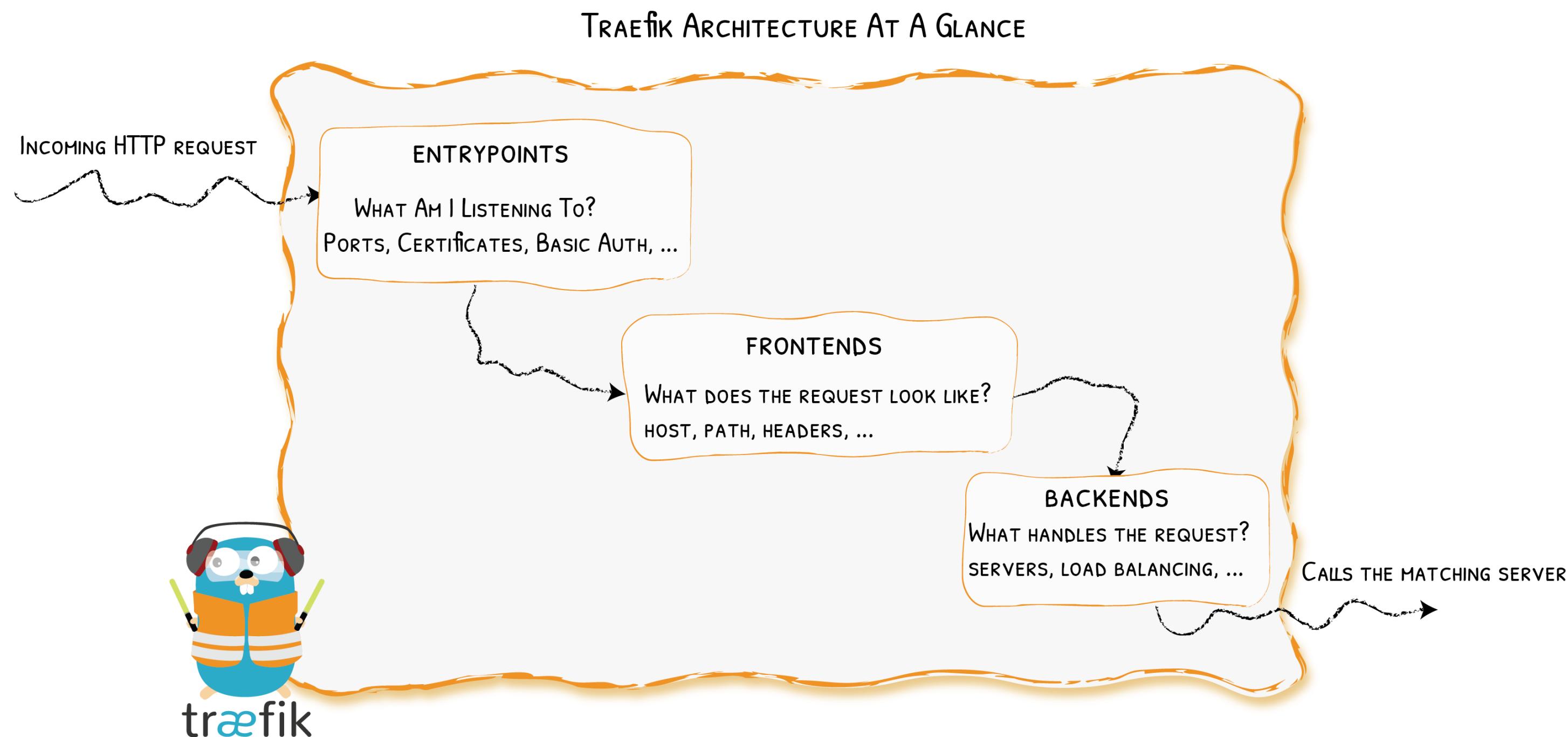
Backends



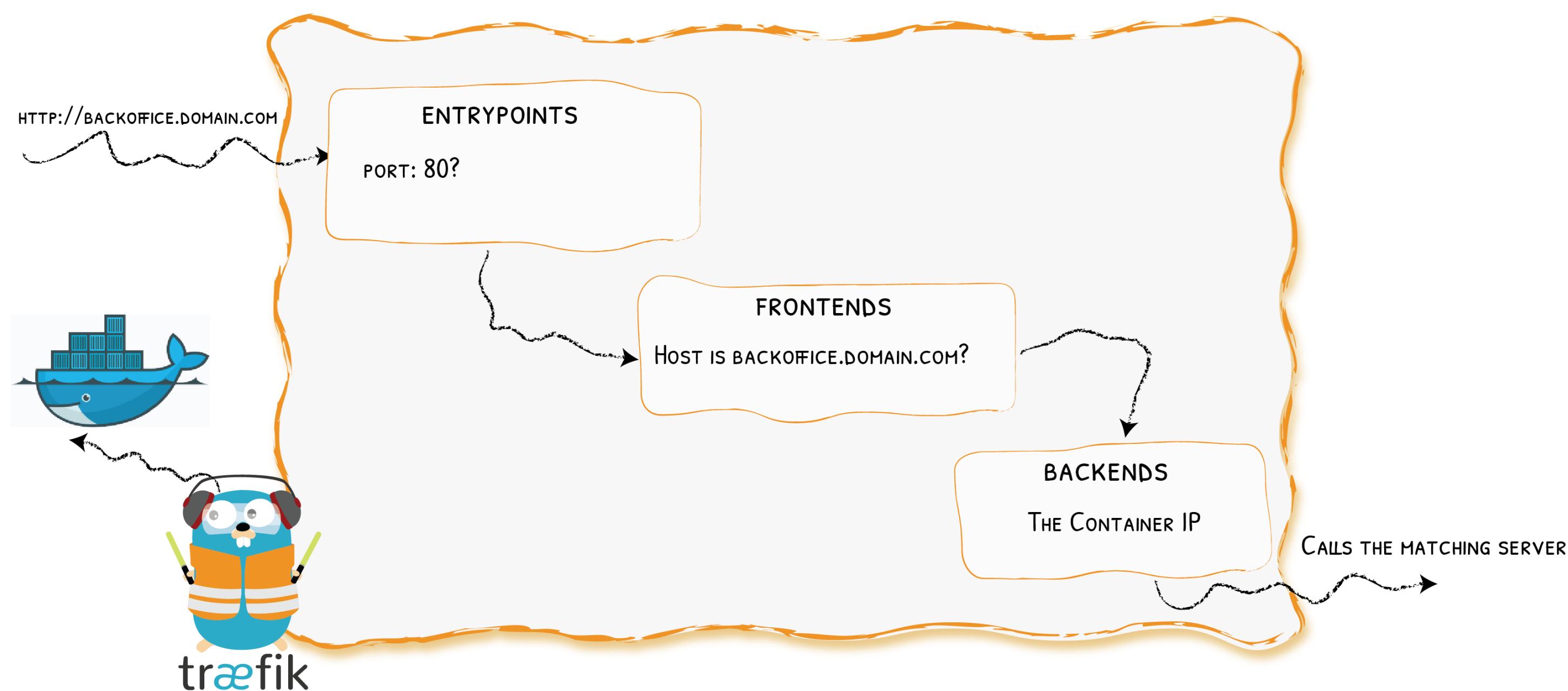
Frontends



At A Glance



In Practice



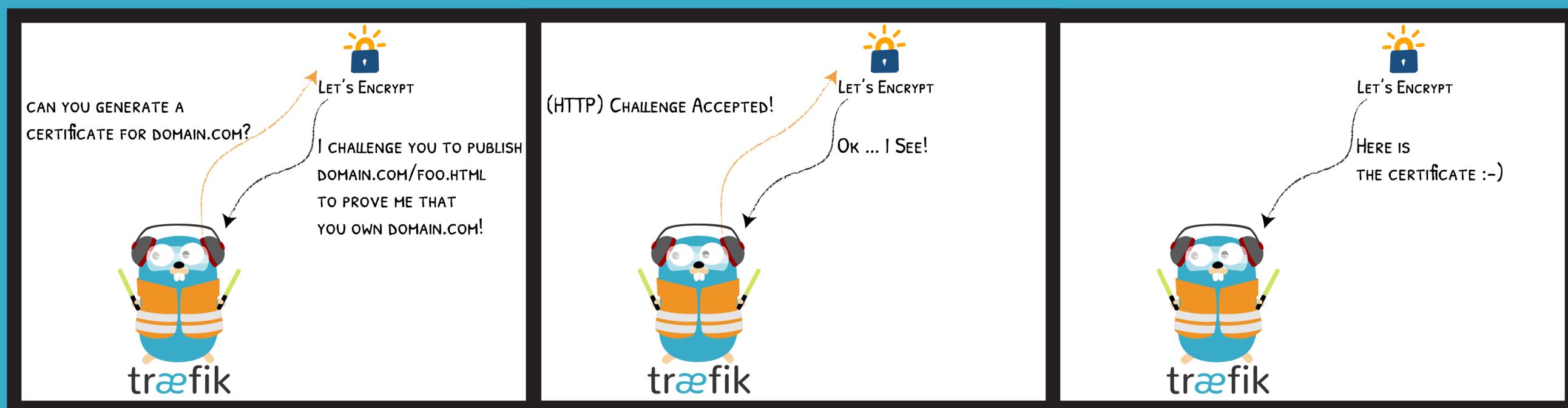
Show Me The Configuration!

Keep It Simple

- With 🐳:

```
entrypoint:  
  image: traefik:v1.7  
  command:  
    - "--docker"  
    - "--docker.domain=mycompany.org"  
    - "--acme.email=ssl-admin@mycompany.org"  
    - "--acme.httpChallenge.entryPoint=http"  
  # Or you could use a TOML file with "--configFile=/etc/traefik/traefik.toml  
volumes:  
  - /var/run/docker.sock:/var/run/docker.sock
```

HTTPS For Everyone With Let's Encrypt



- TLS, DNS and HTTP challenges supported

With 🐳: Simple Backend

```
# https://www.mycompany.org -> http://webserver:80/
webserver:
  image: nginx:alpine
  labels:
    - "traefik.frontend.rule=Host:www.mycompany.org"
```

With Context

```
# https://mycompany.org/jenkins -> http://jenkins:8080/jenkins
jenkins:
  image: jenkins/jenkins:lts
  labels:
    - "traefik.frontend.rule=PathPrefix:/jenkins"
    - "traefik.port=8080" # Because 50000 is also exposed
  environment:
    - JENKINS_OPTS=--prefix=/jenkins
```

With 🐳: Rewrites

```
# https://mycompany.org/gitserver -> http://gitserver:3000/
gitserver:
  image: gitea/gitea:1.5
  labels:
    - "traefik.frontend.rule=PathPrefixStrip:/gitserver"
    - "traefik.port=3000" # Because 22 is also exposed
```

With Websockets

```
# https://mycompany.org/webterminal -> http://webterminal:7681/
webterminal:
  image: ts10922/ttyd
  labels:
    - "traefik.frontend.rule=PathPrefixStrip:/webterminal"
  expose:
    - "7681"
```

Traefik With ⚓

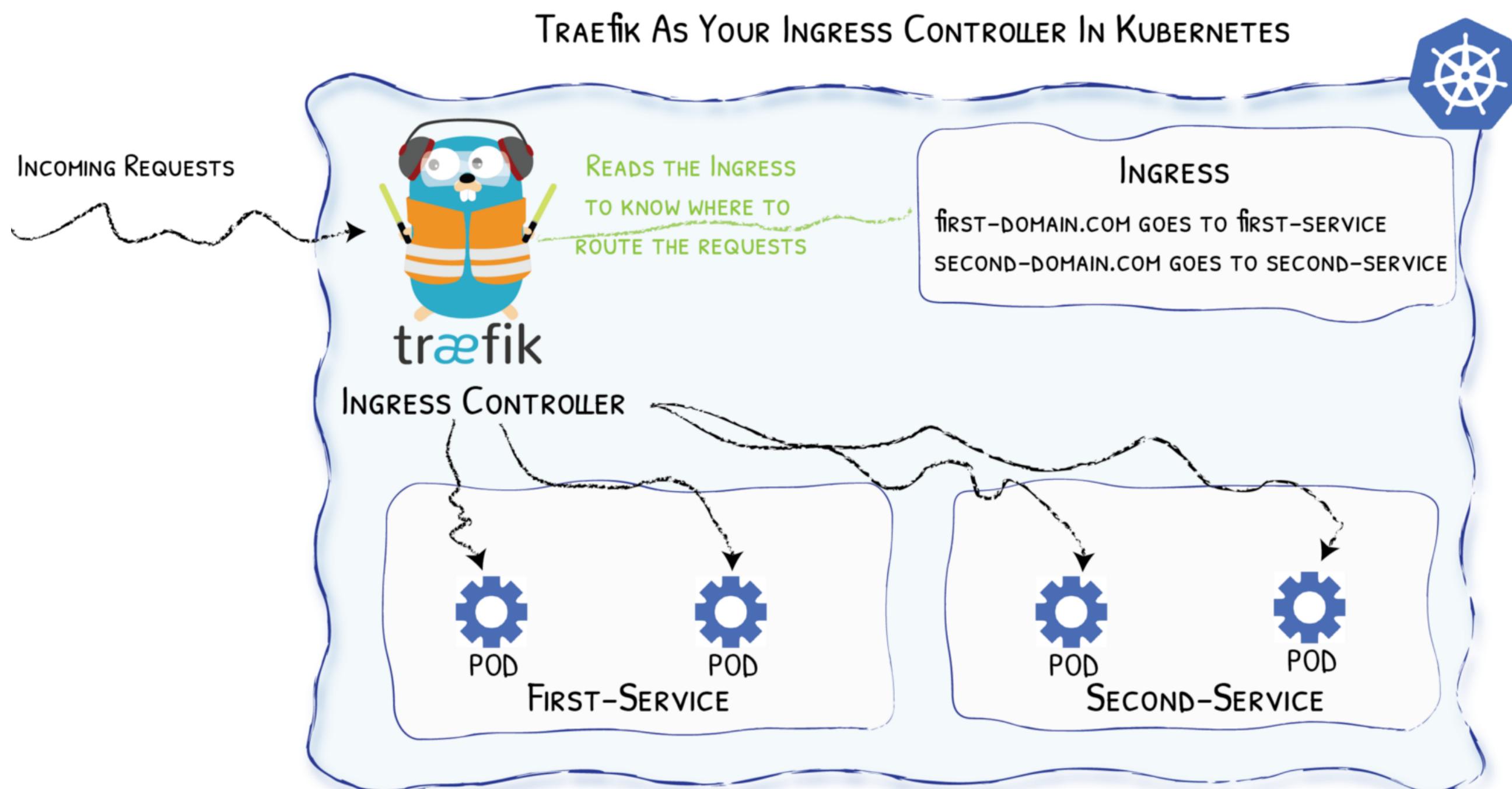


Diagram from <https://medium.com/@geraldcroes>

Did You Say YAML?

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

We Missed Talking About...

A cloud of technology terms in various colors, including:

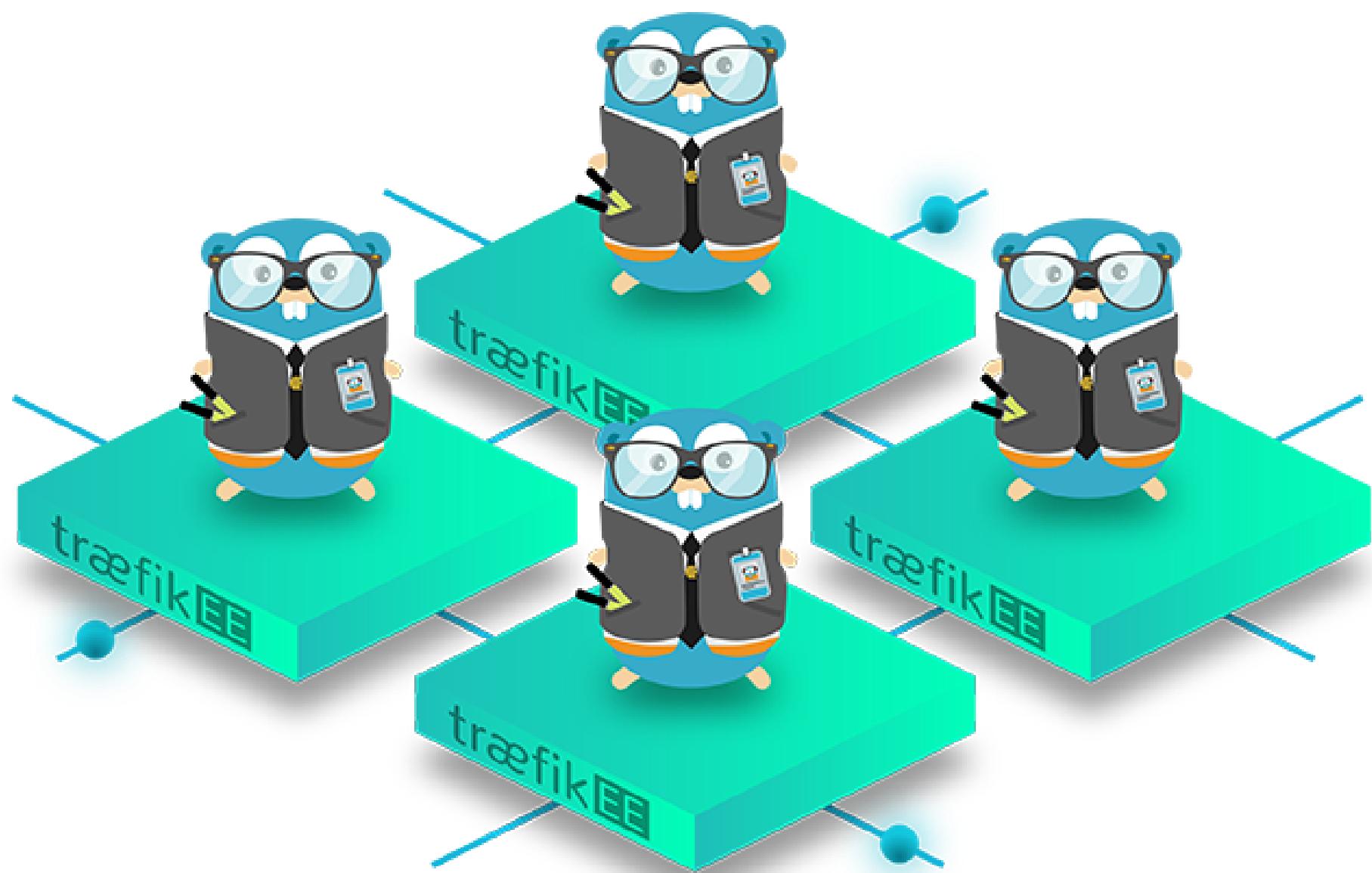
- MESOS
- ZIPKIN
- LIMITING
- KUBERNETES
- Dynamic Metrics
- HTTP
- CERTIFICATE
- ERROR
- TLS Reverse-Proxy
- HEADERS
- GRPC
- DYNAMIC/WILDCARD
- Security Configurations
- Tracing PROXY
- SECRETS
- PROMETHEUS
- JAEGER
- WEBSOCKETS
- SSL
- FORWARD REDIRECTS
- DOCKER
- PROTOCOL
- CHECKS
- CLUSTER AUTH
- HSTS
- RATE
- CONSUL
- SWARM MODE
- S3
- BREAKERS
- CIRCUIT

The Herd



You came to the wrong neighbour

Traefik Comes In Herd



High Availability



HIGH AVAILABILITY

traefik ENTERPRISE EDITION

SECURITY

traefik ENTERPRISE EDITION

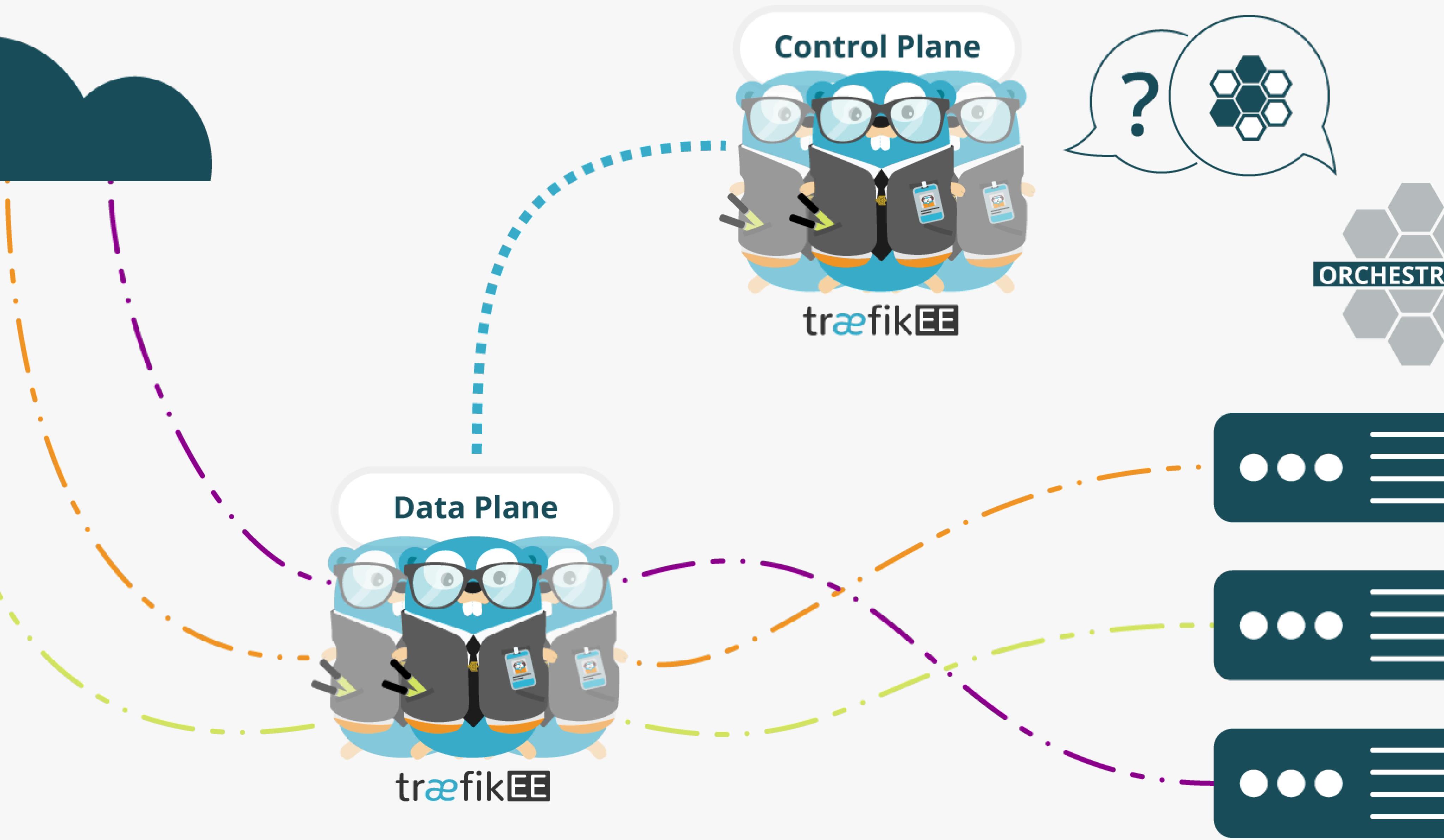
Scalability

SCALABILITY

traefik ENTERPRISE EDITION

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
  ...
```

Early (Free) Access

<https://containo.us/traefikee>

But
What About Open Source?

BACK toTRAEFIK 2.8

PART →



Revamped Documentation

The screenshot shows the Traefik documentation website. The header features a search bar, a GitHub link with 21k stars and 2.1k forks, and a logo. The left sidebar contains a navigation menu with links like Welcome, Getting Started, Configuration Discovery, Routing & Load Balancing, HTTPS & TLS, Middlewares, Operations, Observability, Contributing, and Glossary. The main content area has a large title "Welcome" and a central diagram illustrating Traefik's role as an "Edge Router". The diagram shows requests from the internet (API.DOMAIN.COM, DOMAIN.COM/WEB, BACKOFFICE.DOMAIN.COM) being received by Traefik, which then routes them to various infrastructure components like Kubernetes, Mesos, Docker, and Backoffice servers. Traefik is also shown receiving Metrics, Tracing, and Logs. The bottom text describes Traefik as an open-source Edge Router.

Welcome

FROM THE INTERNET

API.DOMAIN.COM

DOMAIN.COM/WEB

BACKOFFICE.DOMAIN.COM

HTTPS

træfik

To Your Infrastructure

KUBERNETES

MESOS

PRIVATE SERVER

BACKOFFICE SERVER - 1

BACKOFFICE SERVER - 2

DOCKER

Metrics
Tracing
Logs

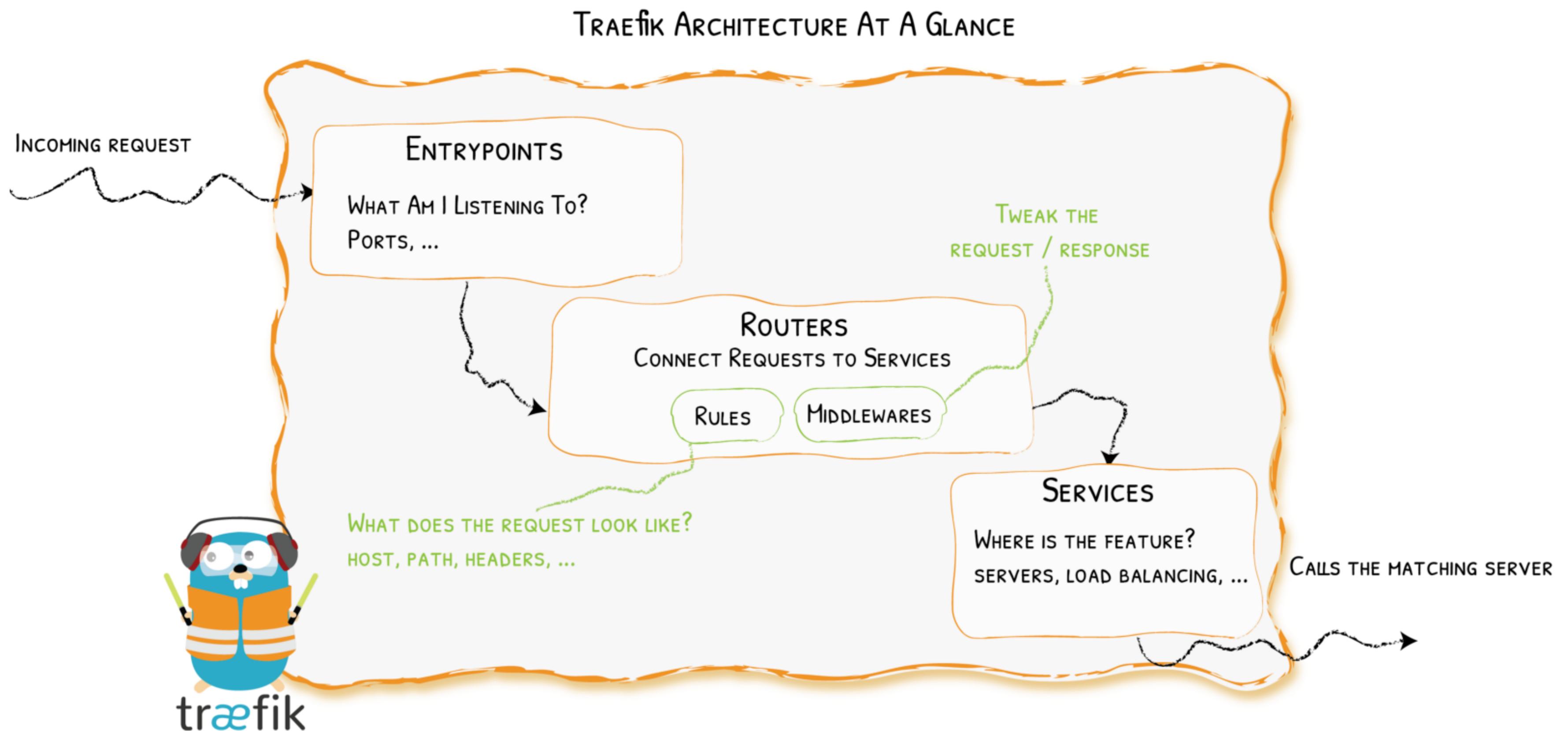
AUTOMATIC & DYNAMIC ROUTING

Load Balancing

Auto Discovery

Traefik is an [open-source](#) *Edge Router* that makes publishing your services a fun and easy experience. It receives requests on behalf of your system and finds out which components are responsible for handling them.

Clarified Concepts

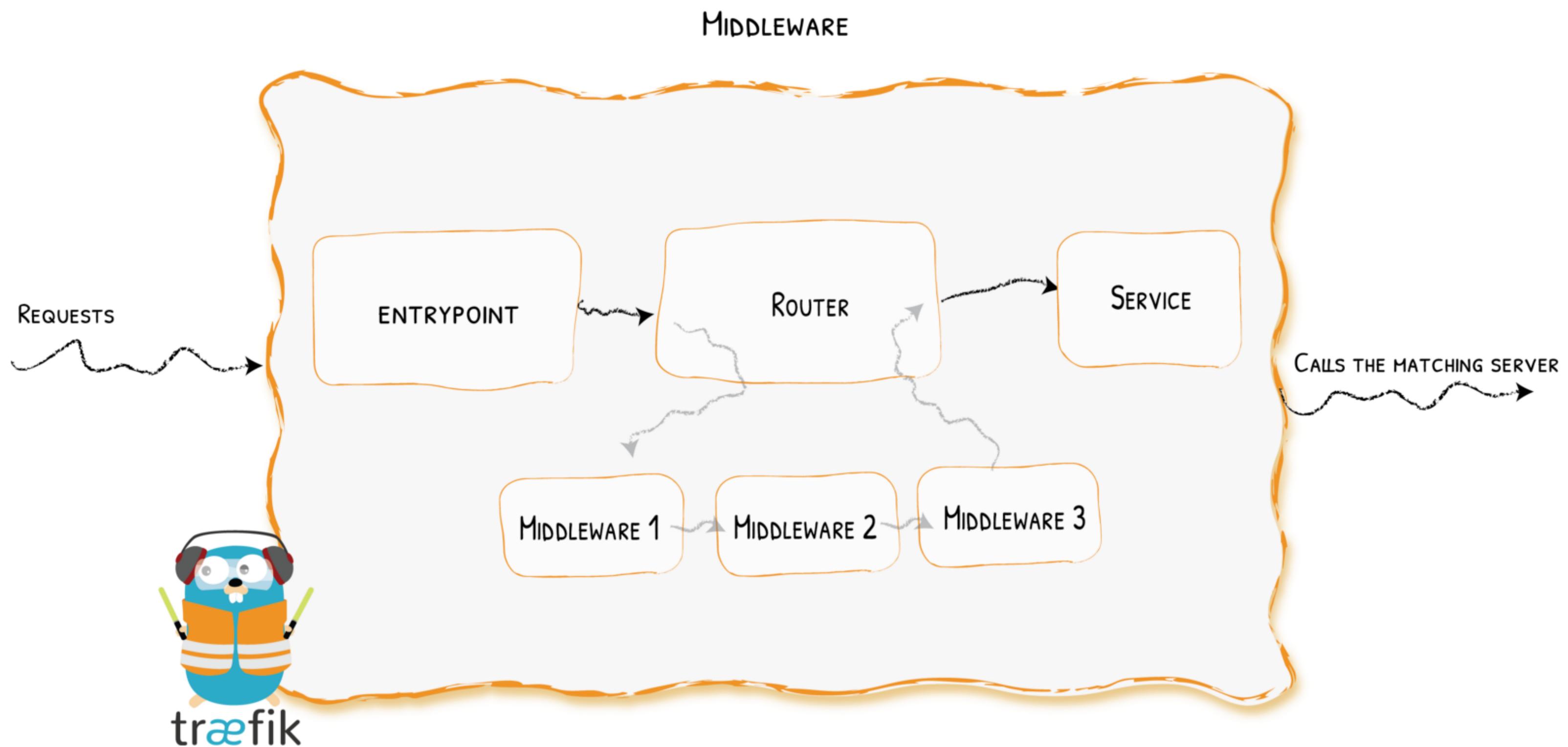


Expressive Routing Rule Syntax



```
# Send both requests to backend service:  
#   https://api.mycompany.com/v2  
#   https://api-v2.mycompany.com  
  
rule=(Host('api.mycompany.com') && PathPrefix('/v2')) || Host('api-v2.mycompany.com')
```

Middlewares





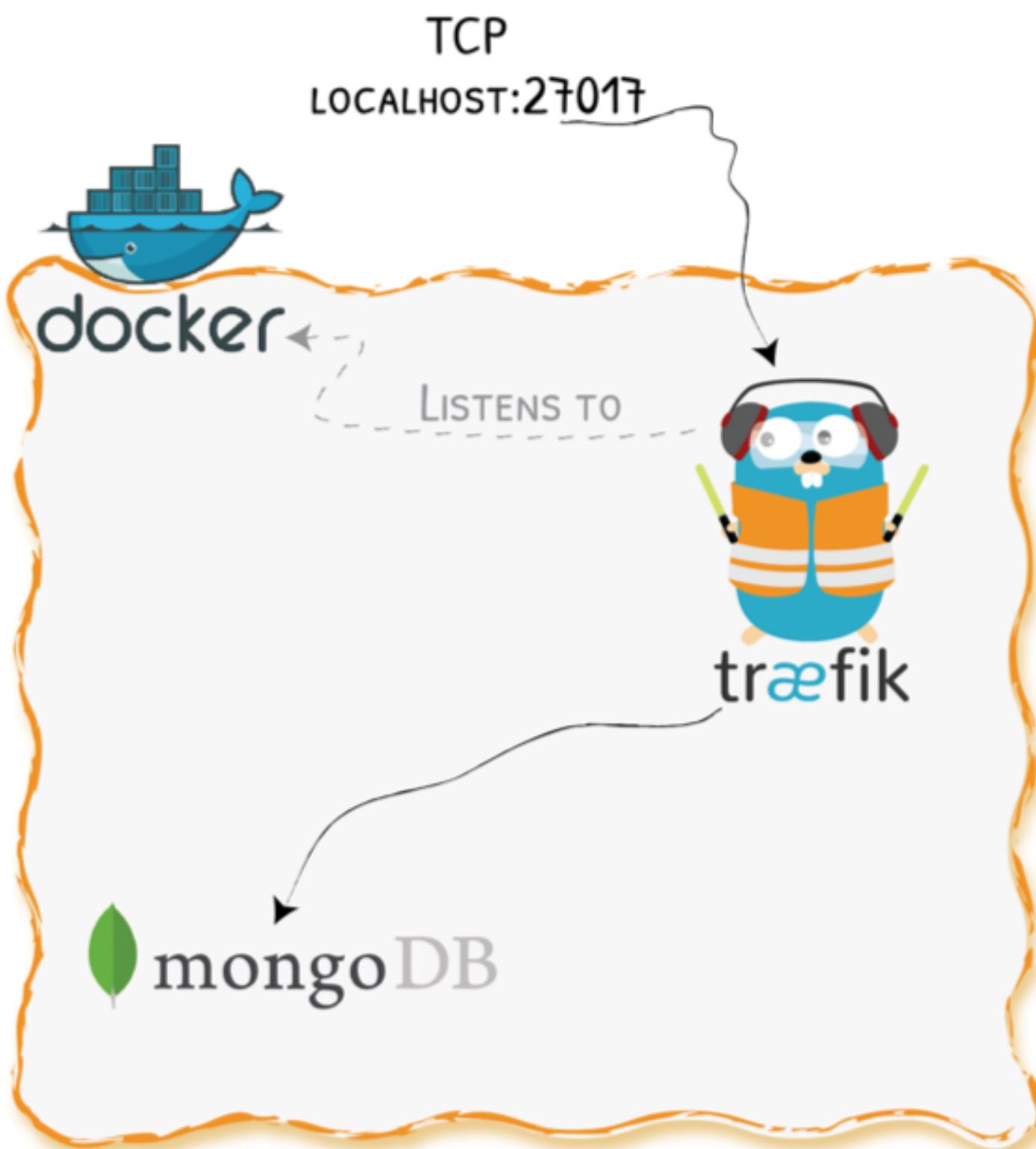
CRD - Custom Resources Definition

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
spec:
  entrypoints:
    - web
    - web-secure
  routes:
    - match: Host(`traefik.io`) && PathPrefix(`/foo`)
      kind: Rule
      services:
        - name: whoami1
          port: 80
          strategy: RoundRobin
      middlewares:
        - name: striprefix
    - match: Host(`containo.us`) && Method(`POST`)
      kind: Rule
      services:
        - name: whoami2
          port: 80
  tls:
    secretName: supersecret
```



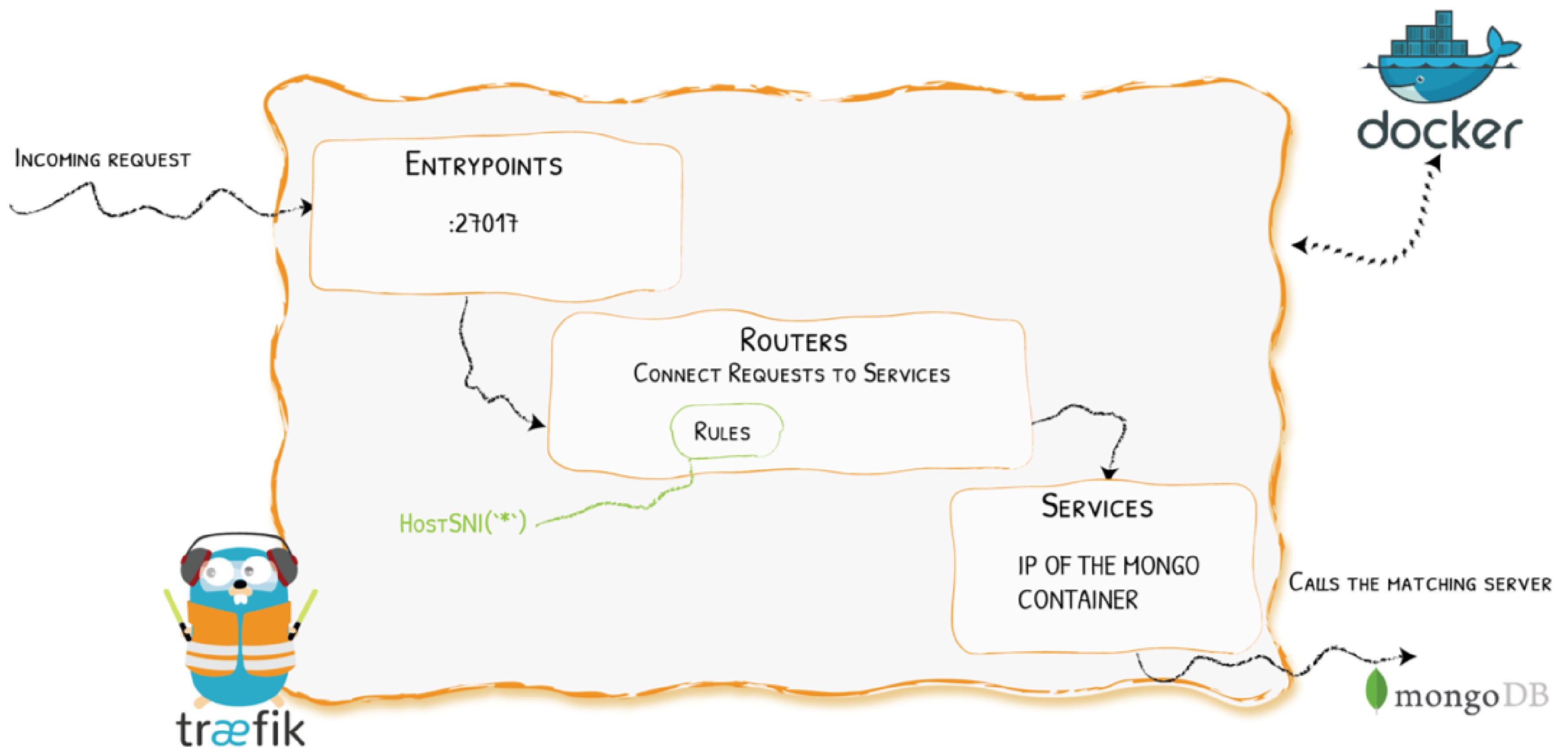
HTTP
&
TCP

Demo 1 - Straightforward TCP Routing

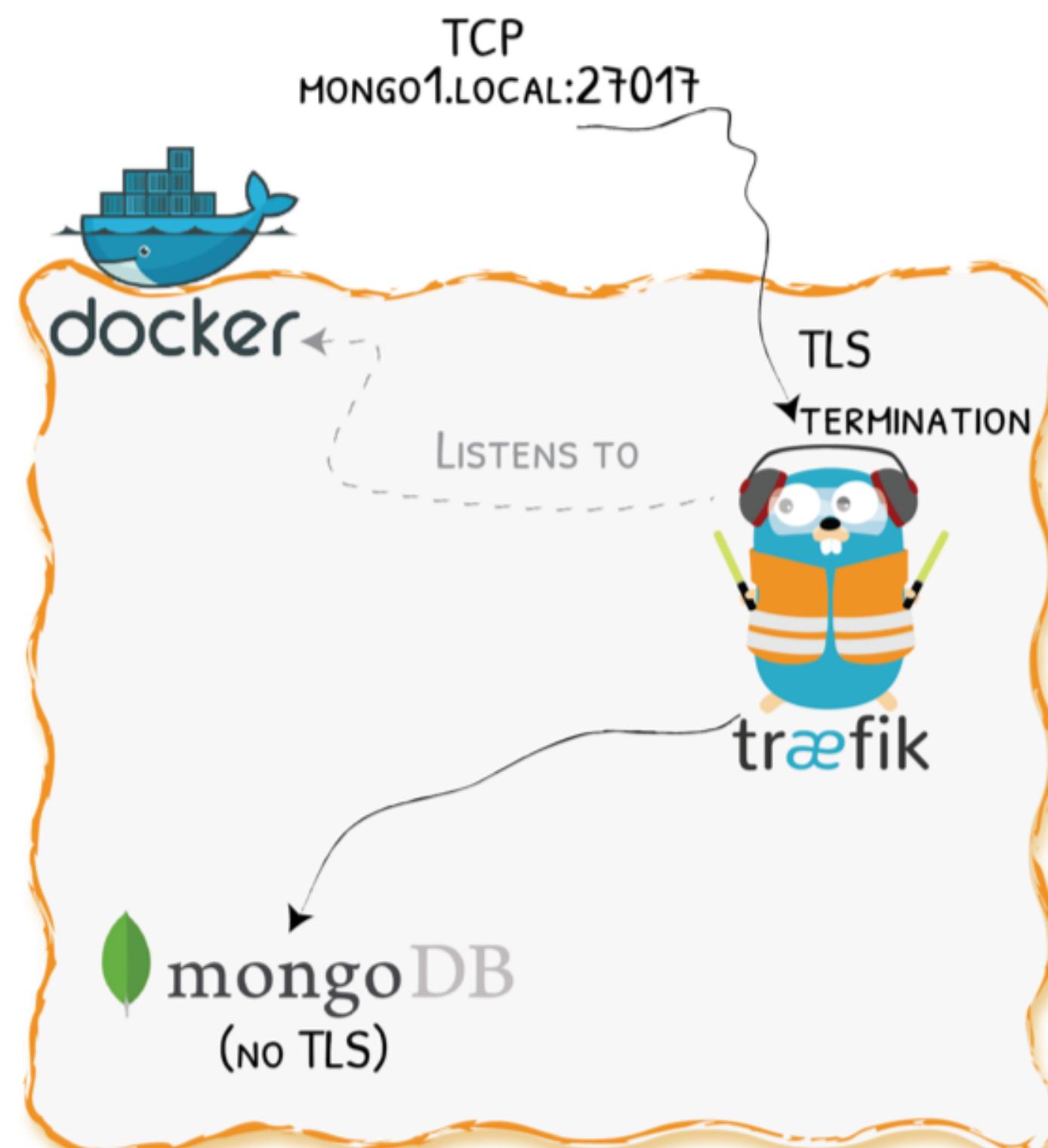


Demo Code on [GitHub](#)

Demo 1 - Configuration

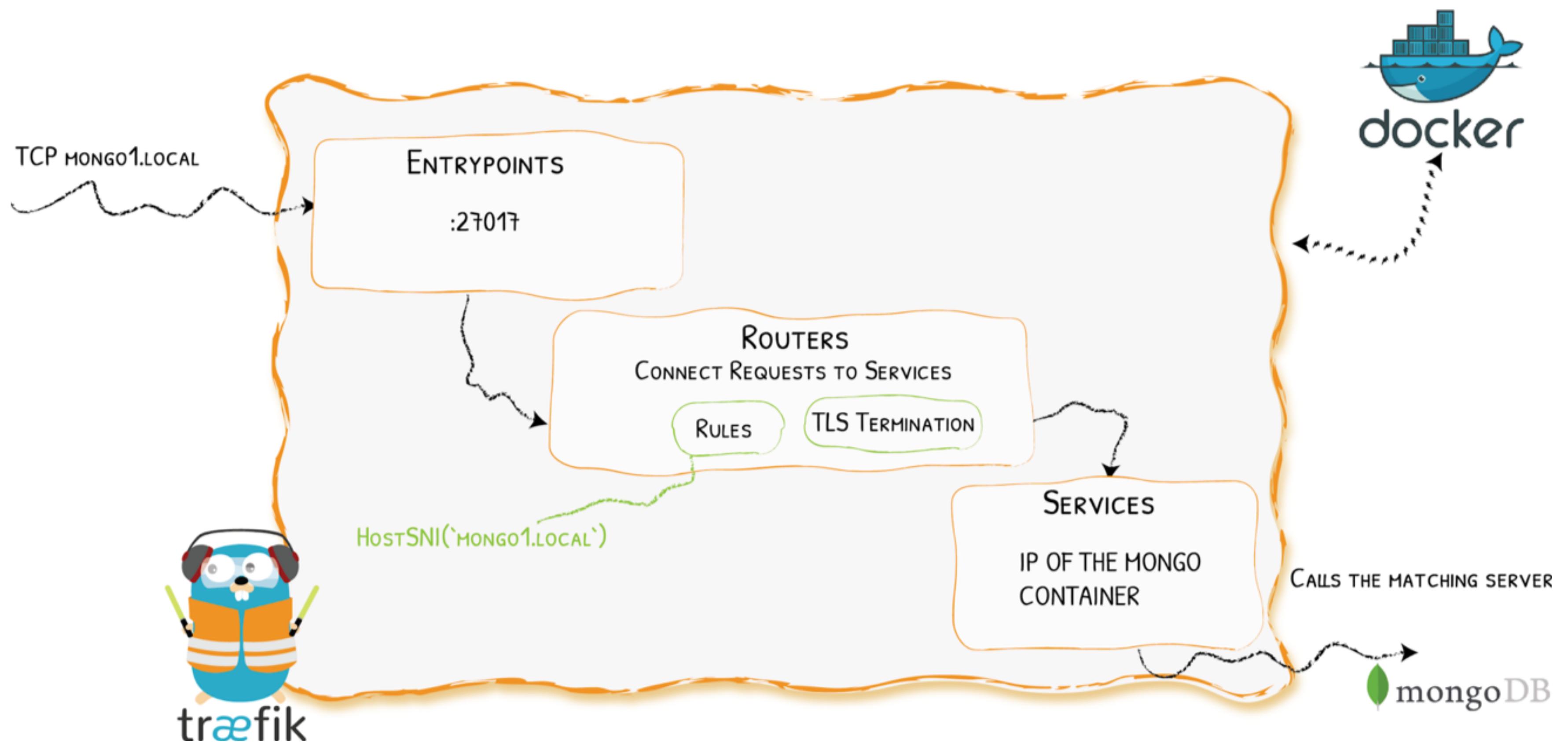


Demo 2 - Let's Add TLS To TCP With Traefik

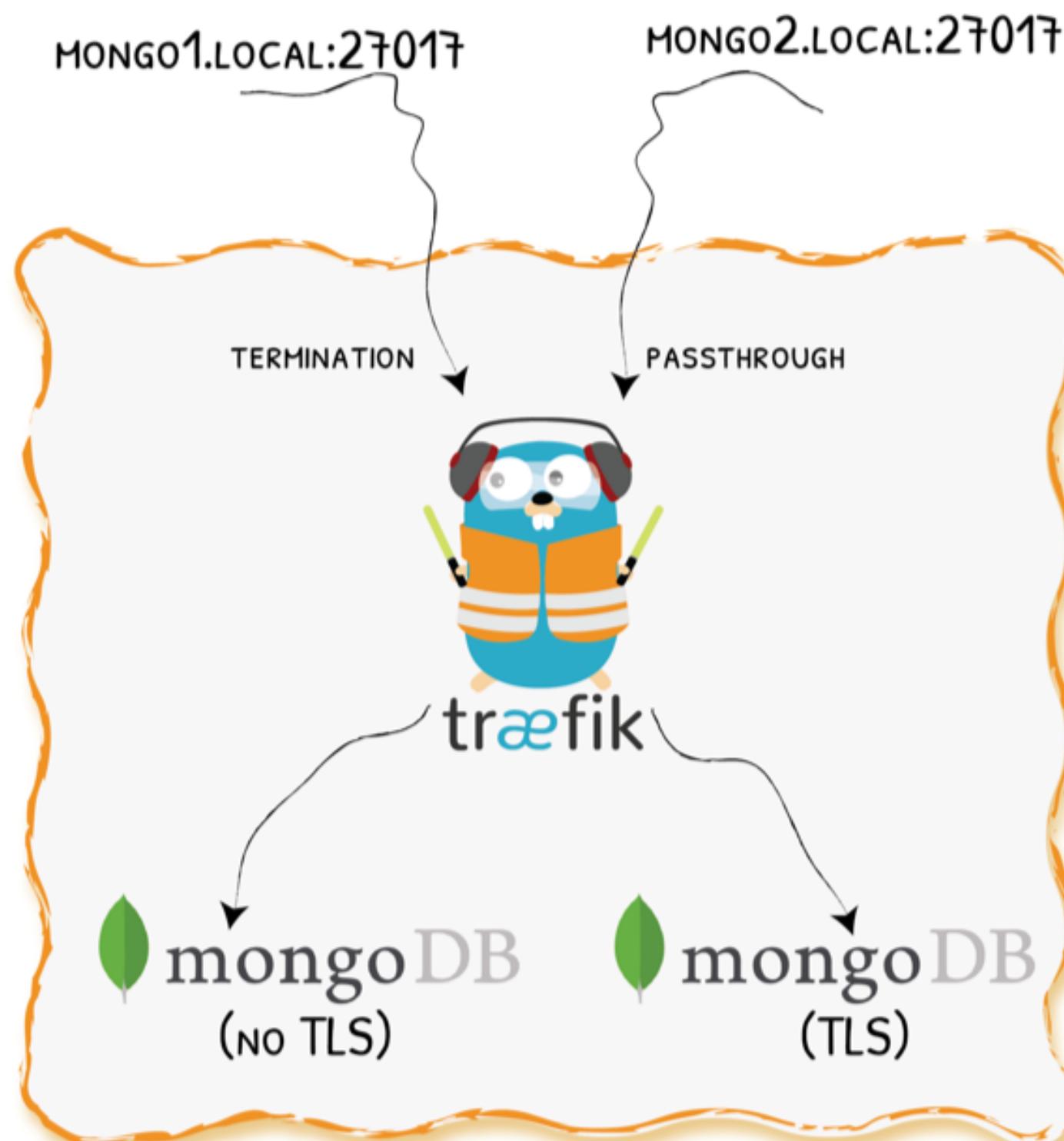


Demo Code on [GitHub](#)

Demo 2 - Configuration

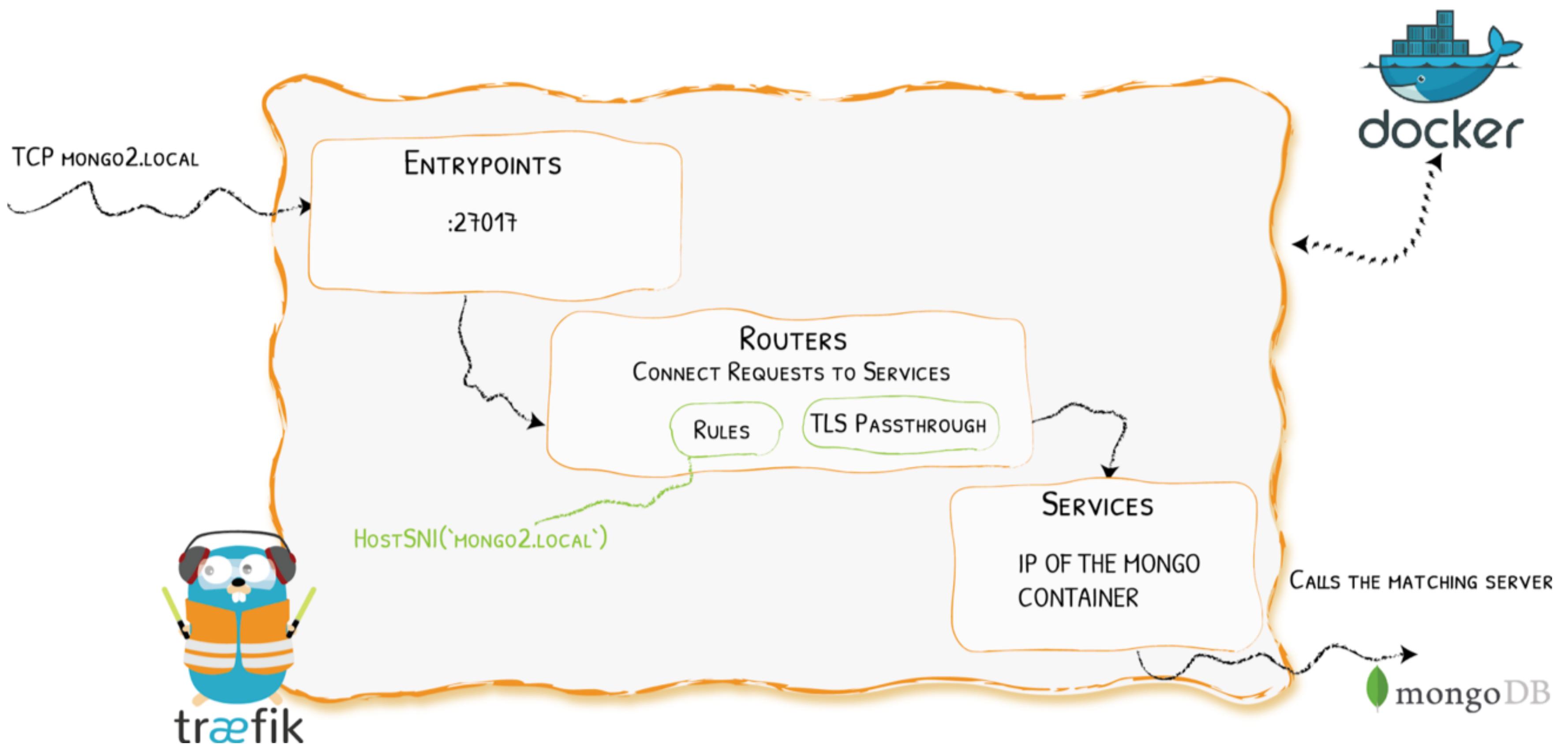


Demo 3 - SNI Routing + TLS Passthrough

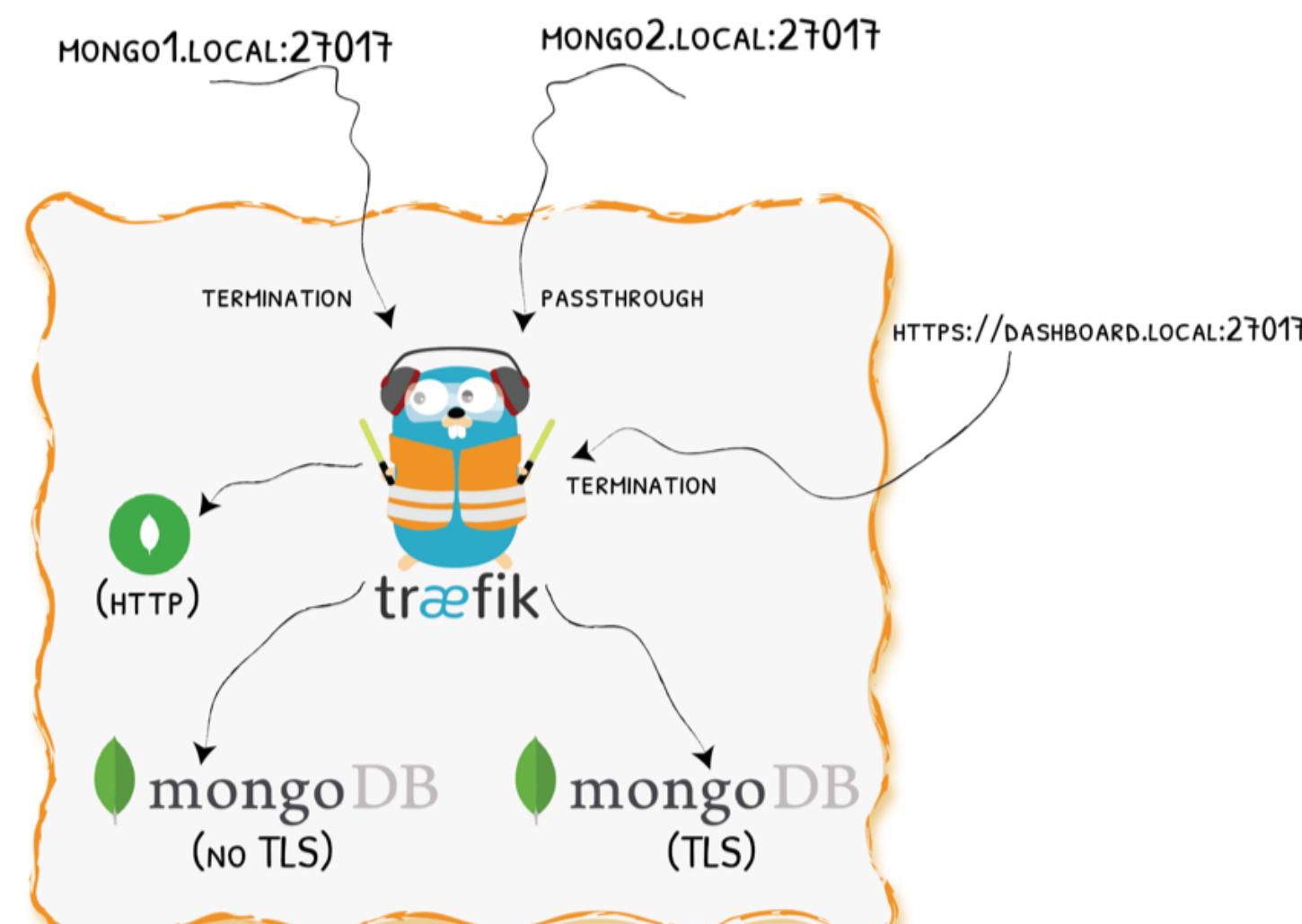


Demo Code on [GitHub](#)

Demo 3 - Configuration



Demo 4 - Muxing HTTPS And TCP On The Same Port



Demo Code on

More To Come

- New WebUI
- Newmetrics
- UDP
- YAML
- TLS stores & options
- Canary

More Info

bit.ly/traefik-v2

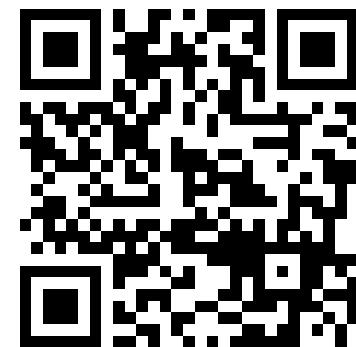
**TO BE
CONTINUED...**



Thank You!

 @DamienDuportal

 dduportal



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