INDONESIA
OpenInfra Days

02.11.2019 | Surabaya, Indonesia

Introduction to Traefik

Open Source Cloud Native Edge Router

Ahmad Saufi Maulana – System Engineer
upi@klim.or.id

Biznet GioCloud   Biznet   Mellanox TECHNOLOGIES   BANK BRI   OSF OpenStack Foundation

Traefik Overview

An open-source reverse proxy and load balancer for HTTP and TCP-based applications that is easy, dynamic, automatic, fast, full-featured, production proven, provides metrics, and integrates with every major cluster technology.

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.

Natively compliant with every major cluster technology

Such as Kubernetes, Docker, Docker Swarm, AWS, Mesos, Marathon, Rancher, Azure Service Fabric, Consul, Etcd, and Amazon DynamoDB; and can handle many at the same time.

Reverse Proxy

Reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client, appearing as if they originated from the proxy server itself.

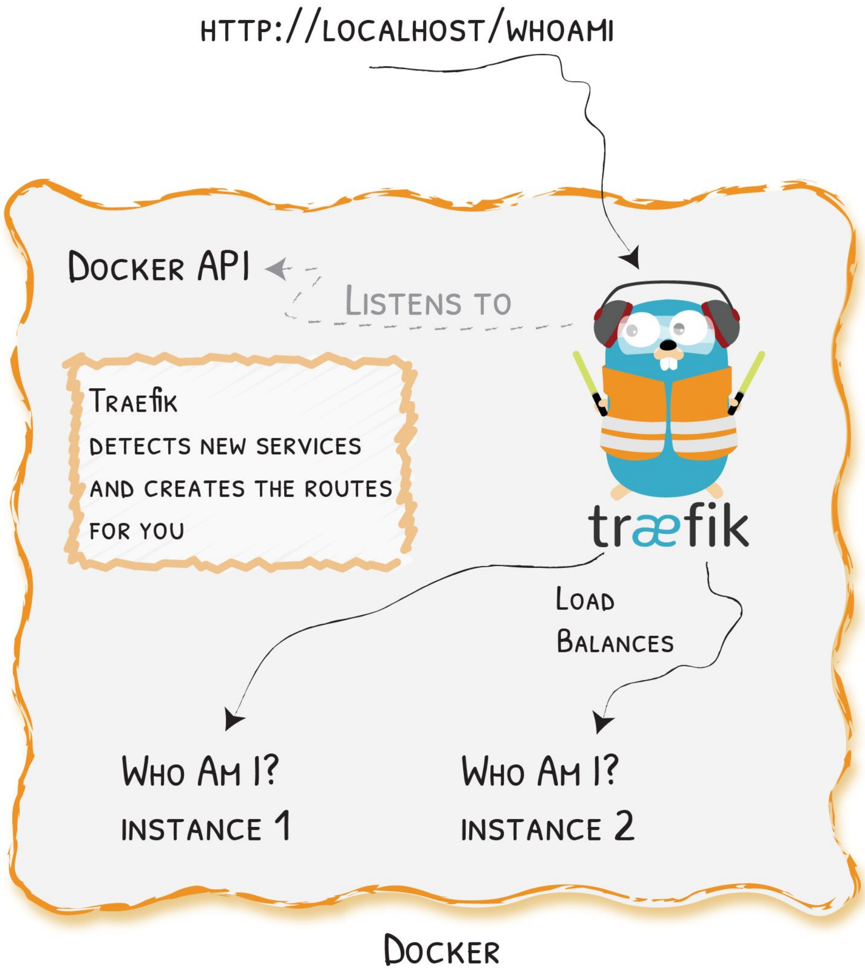
Load Balancing

Load balancing improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives.

Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.

Demo 1 - Simple Use Case Using Docker

It’s just a simple demo, don’t panic and get confused.



HTTP://LOCALHOST/WHOAMI

Docker API

Listens to

Traefik
detects new services
and creates the routes
for you

træfik

Load
Balances

Who Am I?
instance 1

Who Am I?
instance 2

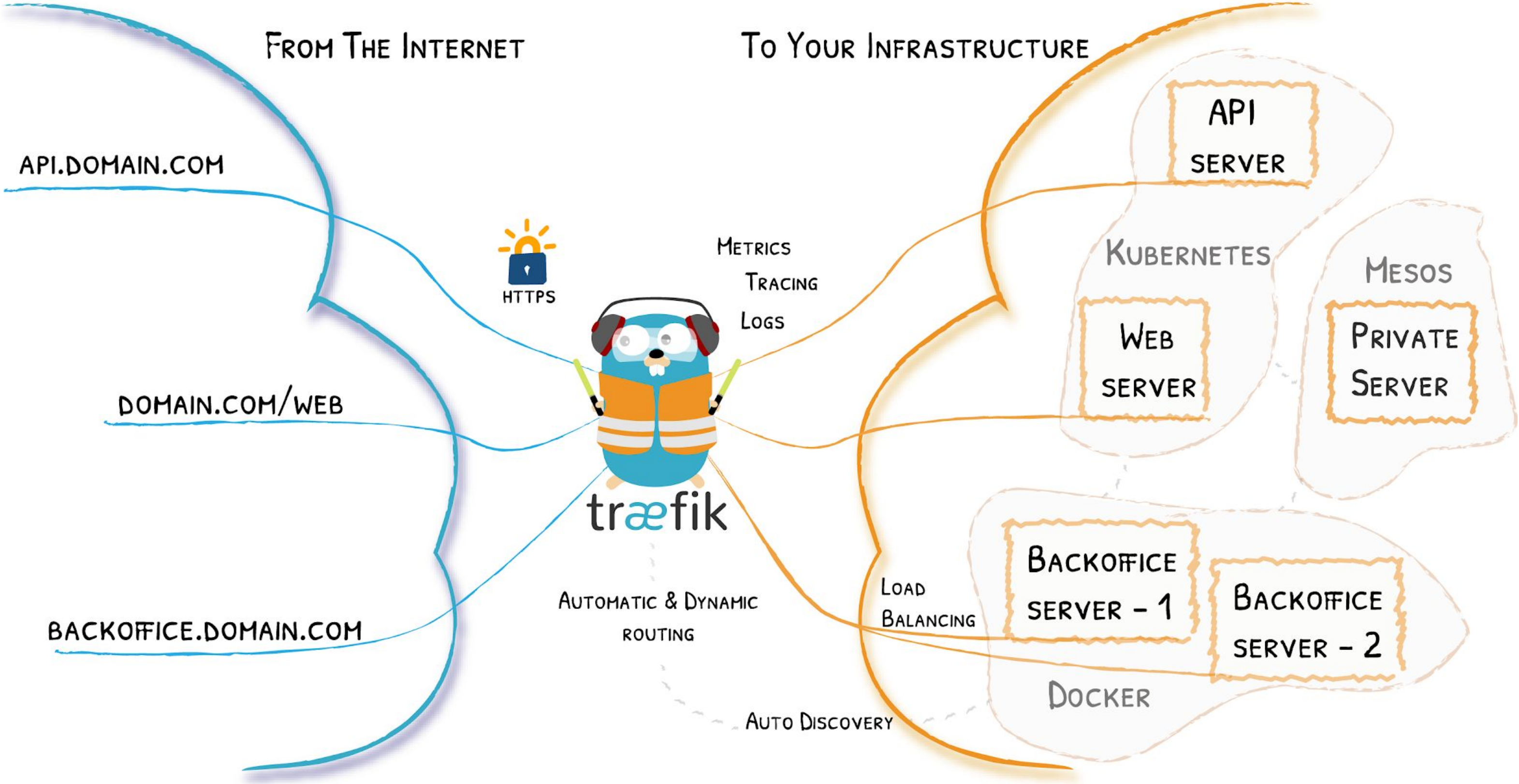
Docker

Traefik is the only accessible container from the outside.

Other services are deployed on demand and available throught traefik.

Traefik Concepts
everything you need to know

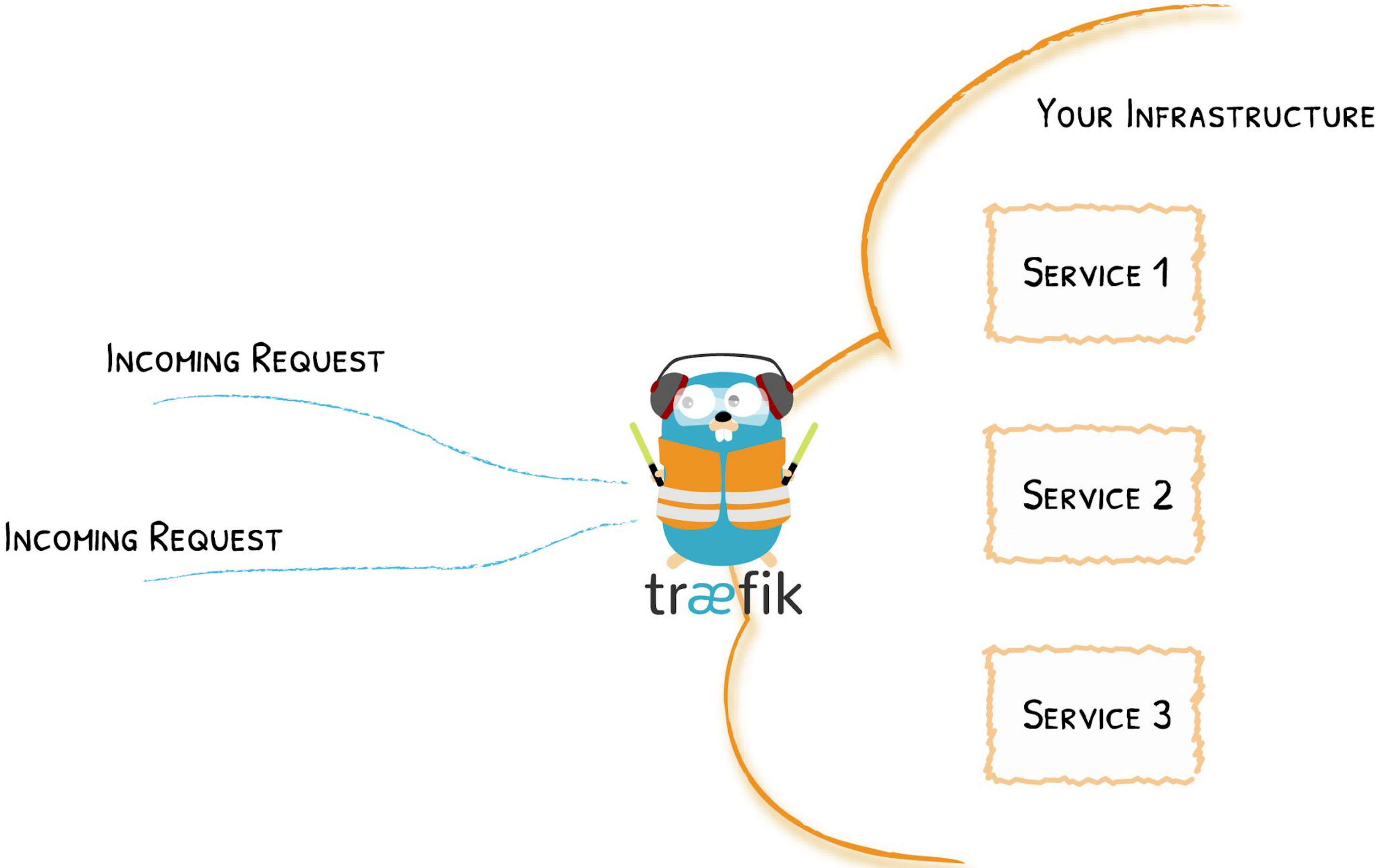
- Edge Router - the door to your platform
- Auto Service Discovery - no longer need to create and synchronize configuration files cluttered with IP addresses or other rules.


From The Internet
To Your Infrastructure
API.DOMAIN.COM
DOMAIN.COM/WEB
BACKOFFICE.DOMAIN.COM
HTTPS
Metrics
Tracing
Logs
træfik
Automatic & Dynamic
routing
Auto Discovery
Load
Balancing
API
SERVER
Kubernetes
Web
SERVER
Mesos
Private
Server
Backoffice
server - 1
Backoffice
server - 2
Docker

Edge Router

Edge router is a specialized router residing at the edge or boundary of a network. This router ensures the connectivity of its network with external networks, a wide area network or the Internet.

Traefik is an Edge Router, it means that it's the door to your platform, and that it intercepts and routes every incoming request.



Your Infrastructure

Service 1

Service 2

Service 3

Incoming Request

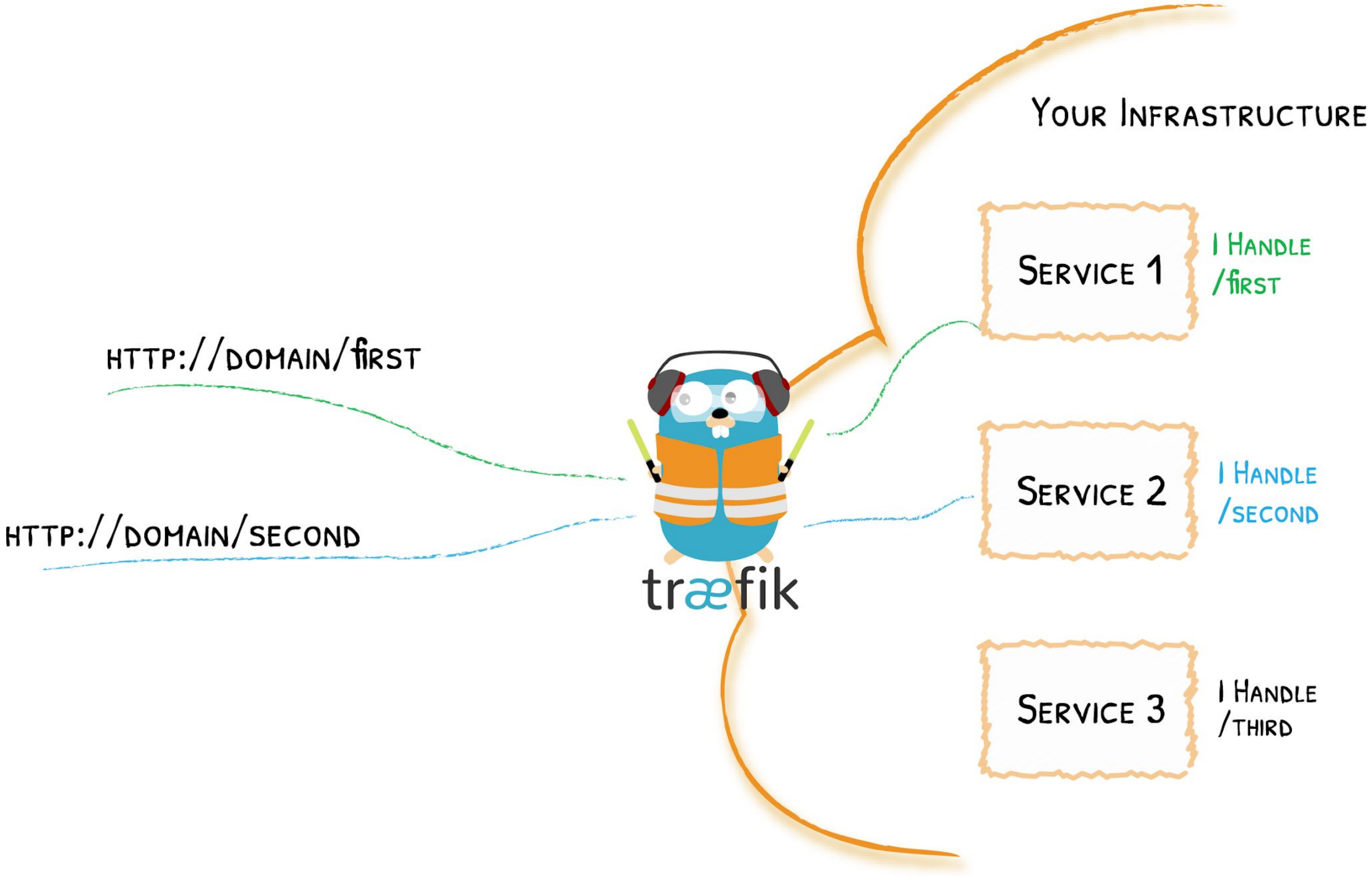
Incoming Request

træfik

Auto Service Discovery

Where traditionally edge routers (or reverse proxies) need a configuration file that contains every possible route to your services, Traefik gets them from the services themselves.

Deploying your services, you attach information that tells Traefik the characteristics of the requests the services can handle.



Your Infrastructure

Service 1

I Handle /first

Service 2

I Handle /second

Service 3

I Handle /third

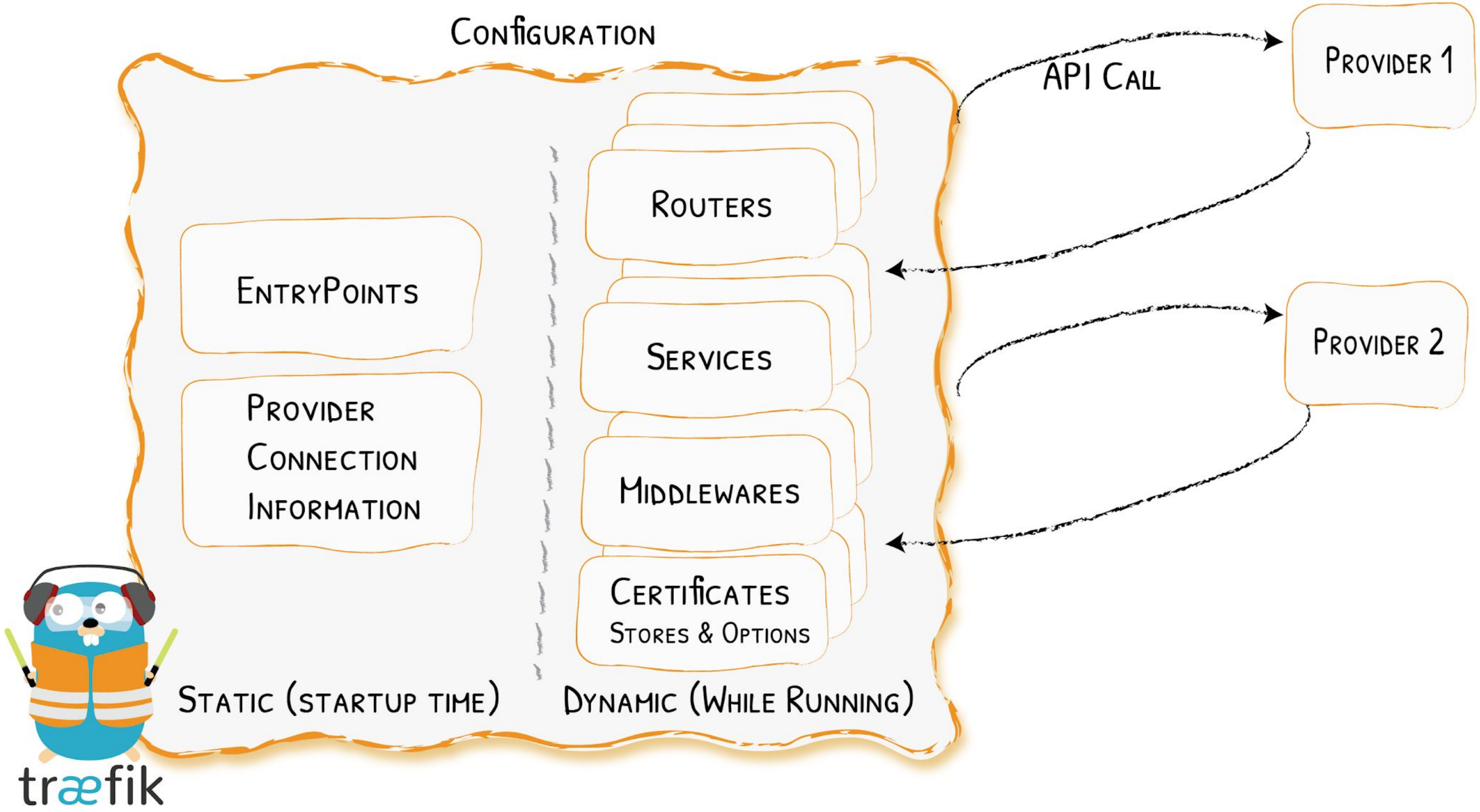
http://domain/first

http://domain/second

Configuration Introduction
how the magic happens?

Configuration in Traefik can refer to two different things:

- The fully dynamic routing configuration (referred to as the dynamic configuration)
- The startup configuration (referred to as the static configuration)


Configuration
EntryPoints
Provider Connection Information
Routers
Services
Middlewares
Certificates
Stores & Options
Static (startup time)
Dynamic (While Running)
API Call
Provider 1
Provider 2
træfik

Dynamic Configuration

The dynamic configuration contains everything that defines how the requests are handled by your system. This configuration can change and is seamlessly hot-reloaded, without any request interruption or connection loss.

Traefik gets its dynamic configuration from providers: whether an orchestrator, a service registry, or a plain old configuration file.

In the Demo 1, the dynamic configuration comes from docker in the form of labels attached to your containers.

Static Configuration

Elements in the static configuration set up connections to providers and define the entrypoints Traefik will listen to (these elements don't change often).

There are three different, mutually exclusive (e.g. you can use only one at the same time), ways to define static configuration options in Traefik:

- In a configuration file
- In the command-line arguments
- As environment variables

These ways are evaluated in the order listed above.

If no value was provided for a given option, a default value applies. Moreover, if an option has sub-options, and any of these sub-options is not specified, a default value will apply as well.

The last
but not least

- HTTPS & TLS - Literally https & tls, what else do you think?
- Middlewares - Tweaking the Request!

HTTPS & TLS

Traefik supports HTTPS & TLS, which concerns roughly two parts of the configuration: routers, and the TLS connection (and its underlying certificates).

You can configure Traefik to use an ACME provider (like Let's Encrypt) for automatic certificate generation.

Traefik automatically tracks the expiry date of ACME certificates it generates.
If there are less than 30 days remaining before the certificate expires, Traefik will attempt to renew it automatically.

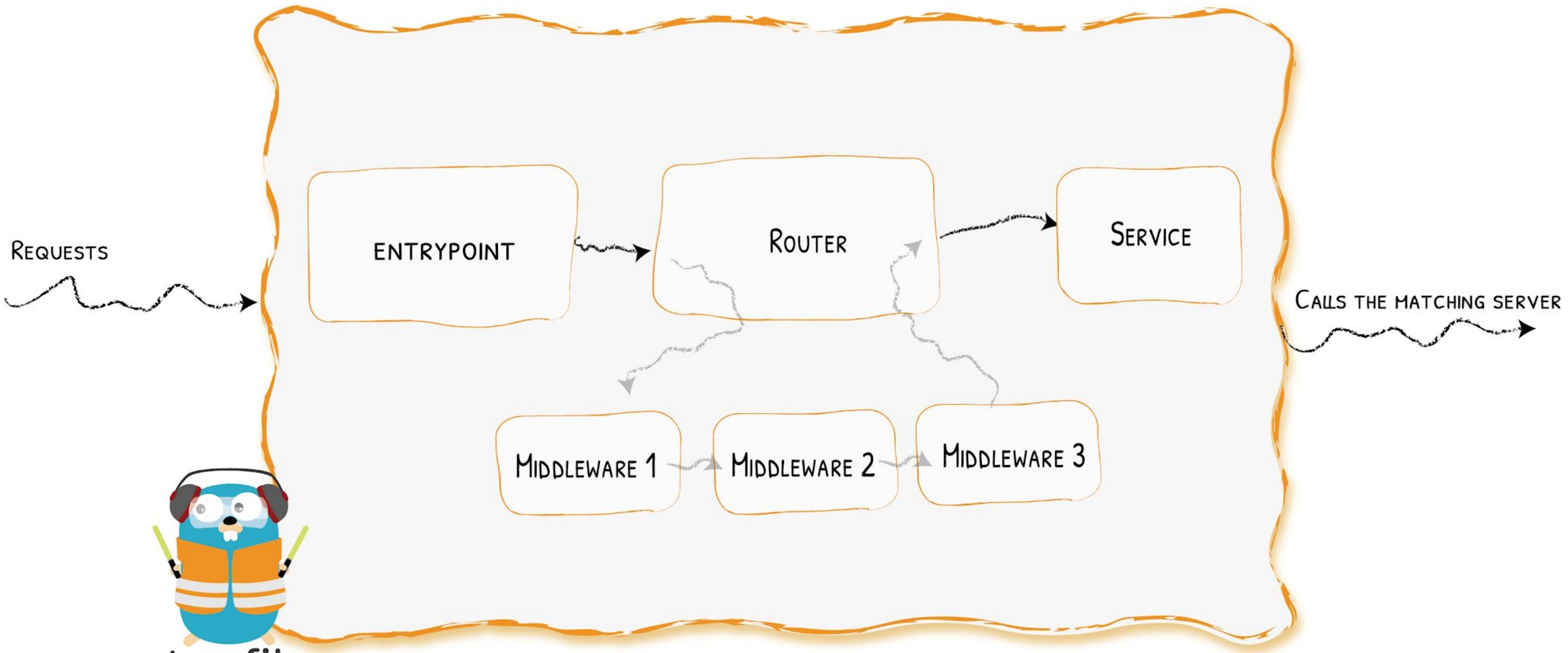
Middlewares

Attached to the routers, pieces of middleware are a mean of tweaking the requests before they are sent to your service (or before the answer from the services are sent to the clients).

There are many different available middlewares in Traefik, some can modify the request, the headers, some are in charge of redirections, some add authentication, and so on.

Pieces of middleware can be combined in chains to fit every scenario.

Middleware



Requests

Entrypoint

Router

Service

Middleware 1

Middleware 2

Middleware 3

Calls the matching server

træfik

Demo 2 - Traefik with Let’s Encrypt

This demo is to demonstrate how to create a certificate with the Let's Encrypt TLS challenge to use https on a simple service exposed with Traefik.

Don’t get too confused, give yourself a little bit of challenge for fun!

Thank you!