



agenda »

Intro to containers

Azure VM deployment

Portainer

Traefik

USING TRAEFIK AS REVERSE PROXY FOR WINDOWS CONTAINER BASED ENVIRONMENTS

Tobias Fenster, April 16 2021





Tobias Fenster

CTO COSMO CONSULT Group

Dual Microsoft MVP for Business
Applications and Azure

Microsoft Regional Director

Ambassador traefik and portainer

 @tobiasfenster

 tobiasfenster.io

 tobiasfenster



ASSUMPTION:

You know how containers and especially networking with containers work





Overview



EDGE ROUTER

INCOMING REQUEST
INCOMING REQUEST



YOUR INFRASTRUCTURE

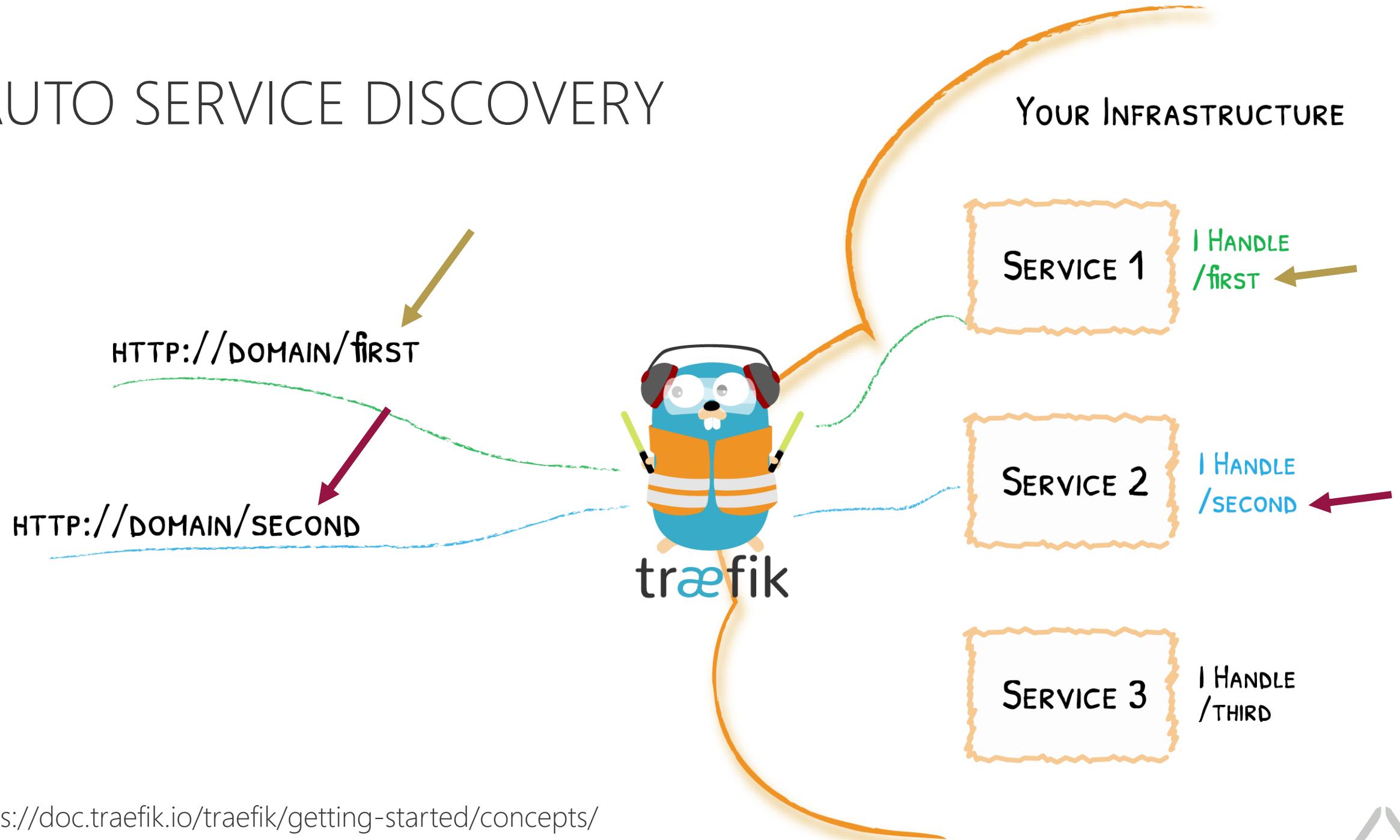
SERVICE 1

SERVICE 2

SERVICE 3

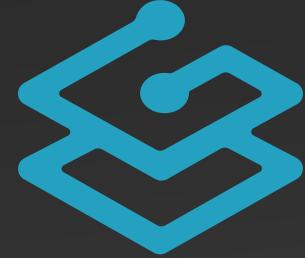


AUTO SERVICE DISCOVERY



<https://doc.traefik.io/traefik/getting-started/concepts/>





traefik
proxy

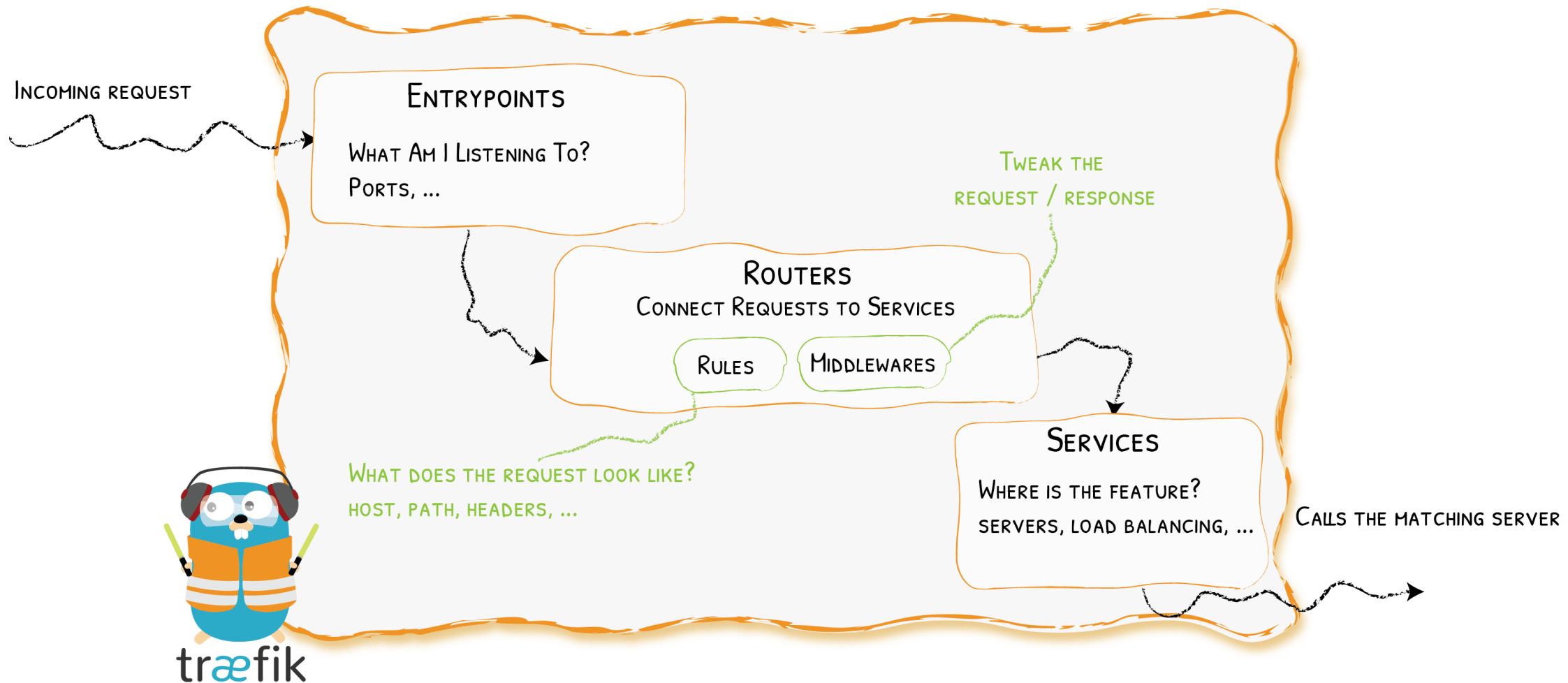




Configuration



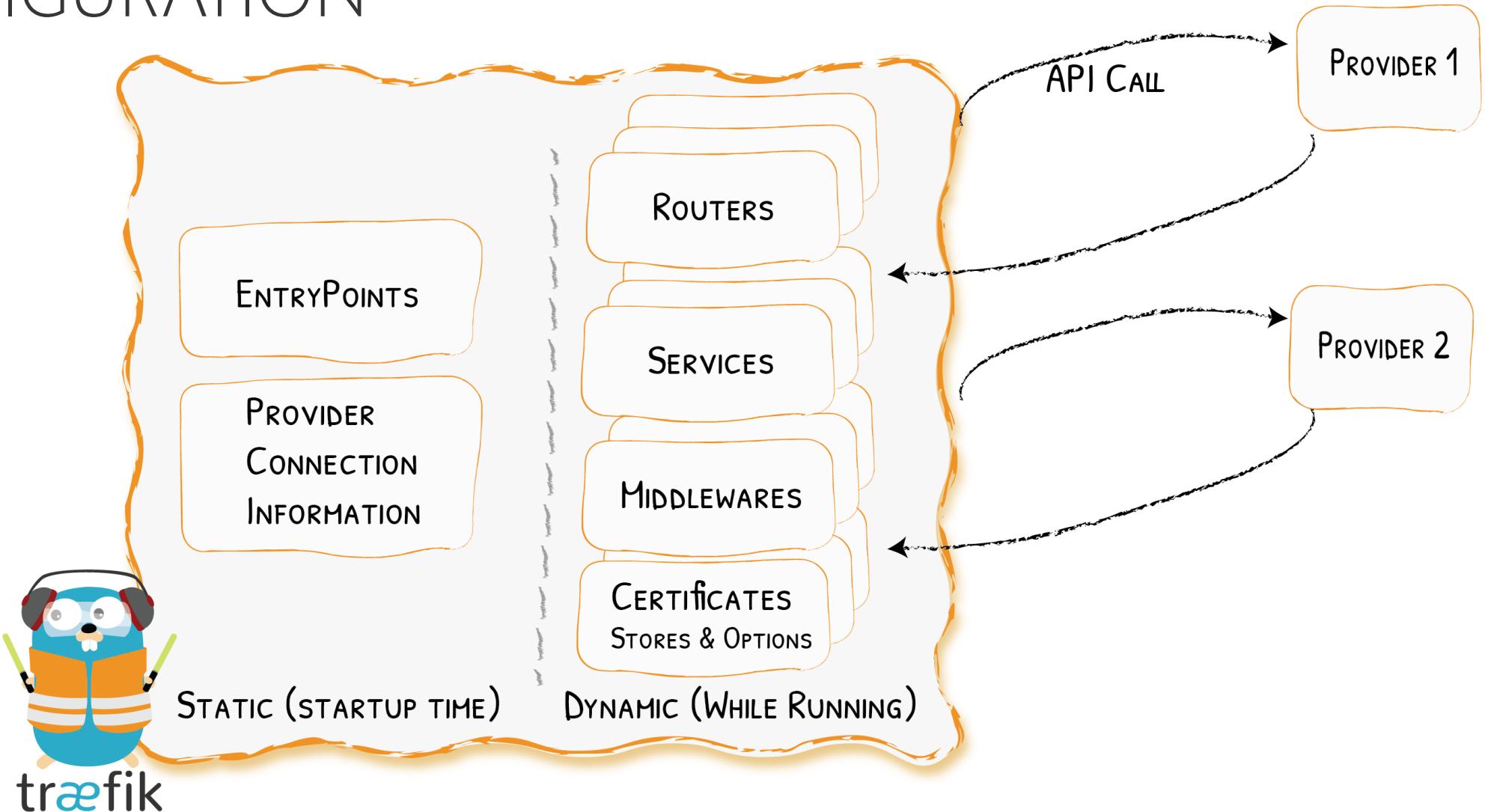
OVERALL ARCHITECTURE



<https://doc.traefik.io/traefik/routing/overview/>



CONFIGURATION



<https://doc.traefik.io/traefik/getting-started/configuration-overview/>



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



CONFIGURATION

Dynamics configuration as labels on the containers

services:

sysinfo4:

image: tobiasfenster/container-sysinfo:4.0.0

labels:

- "traefik.enable=true"
- "traefik.http.services.sysinfo4.loadBalancer.server.port=80"
- "traefik.http.services.sysinfo4.loadBalancer.server.scheme=http"
- "traefik.http.routers.sysinfo4.rule=PathPrefix(`/sysinfo4/`)"
- "traefik.http.routers.sysinfo4.tls.certresolver=myresolver"

...





Let's
encrypt



CONFIGURATION

Static configuration as part of the traefik container cmd

```
--providers.docker=true  
  --providers.docker.exposedbydefault=false  
  --providers.docker.endpoint=npipe:///./pipe/docker_engine  
--entrypoints.websecure.address=:443  
--certificatesresolvers.myresolver.acme.tlschallenge=true  
  --certificatesresolvers.myresolver.acme.email=tobias.fenster@cosmoconsult.com  
  --certificatesresolvers.myresolver.acme.storage=c:/le/acme.json  
--serversTransport.insecureSkipVerify=true  
--api.dashboard=true
```



LET'S ENCRYPT

Which domain? How does renewal work?

- Looking at the Host / HostRegexp router rules to find out automatically if certResolver is defined
 - "traefik.http.routers.sysinfo5.rule=Host(`docker-4s6koalhokb7w.westeurope.cloudapp.azure.com`) && PathPrefix(`/sysinfo5/`)"
 - "traefik.http.routers.sysinfo5.tls.certresolver=myresolver"
- Alternatively explicitly defined as tls.domains on a router
 - "traefik.http.routers.sysinfo5.tls.domains[0].main=docker-4s6koalhokb7w.westeurope.cloudapp.azure.com,"
- Automatically renewed if certification has < 30 days until expiry
- Expiry notice sent to given email

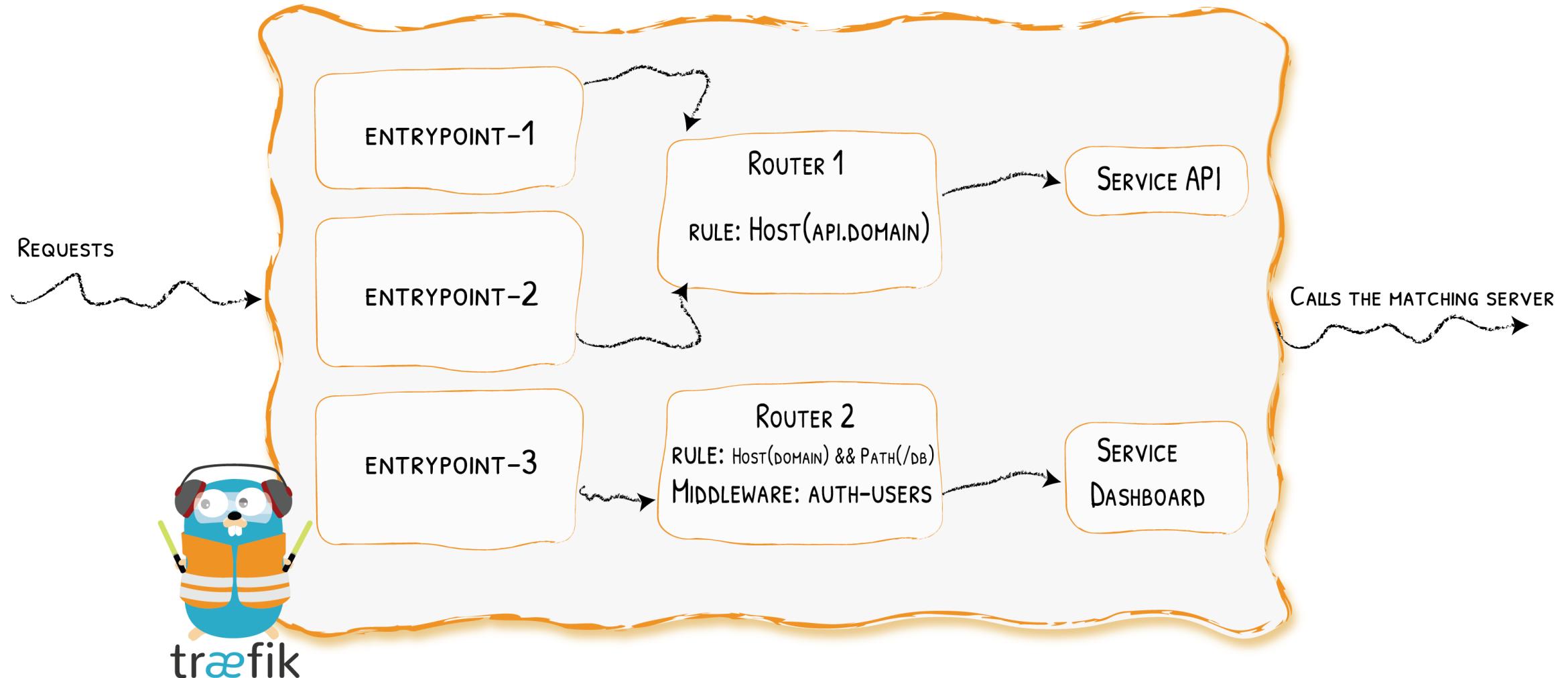


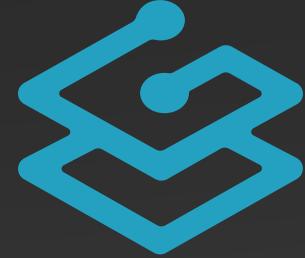


Routers



ROUTERS





traefik
proxy



ROUTER RULES

Who gets which request?

- `Headers(`key` , `value`)` or `HeadersRegexp(`key` , `regexp`)`
Routing based on headers
- `Host(`example.com` , ...)` or `HostRegexp(`example.com` , `{subdomain:[a-z]+}.example.com` , ...)`
Routing based on requested host
- `Method(`GET` , ...)`
Routing based on used HTTP method
- `Path(`/path` , `/articles/{cat:[a-z]+}/{id:[0-9]}+` , ...)`
Routing based on exact matches of the path
- `PathPrefix(`/products` , `/articles/{cat:[a-z]+}/{id:[0-9]}+`)`
Routing based on path prefix matches
- `Query(`foo=bar` , `bar=baz`)`
Routing based on query strings



ROUTER RULES

Who gets which request?

- Can be combined with && and ||, () can be used
- Sorted by default in descending order using rule length (longest has highest priority) unless overruled with explicit priority
- Sample:

```
rule1=HostRegexp(`.*\.westeurope\.cloudapp\.azure\.com`) && Path(`/sysinfo5/`)  
rule2=HostRegexp(`.*\.westeurope\.cloudapp\.azure\.com`) || Query(`ver=4`)  
rule3=Host(`docker-4s6koalhokb7w.westeurope.cloudapp.azure.com`)
```



Request: <https://docker-4s6koalhokb7w.westeurope.cloudapp.azure.com/sysinfo4/>



ROUTER RULES

Who gets which request?

- Can be combined with && and ||, () can be used
- Sorted by default in descending order using rule length (longest has highest priority) unless overruled with explicit priority
- Sample:

```
rule1=HostRegexp(`.*\.westeurope\.cloudapp\.azure\.com`) && Path(`/sysinfo5/`)  
rule2=HostRegexp(`.*\.westeurope\.cloudapp\.azure\.com`) || Query(`ver=4`)  
rule3=Host(`docker-4s6koalhokb7w.westeurope.cloudapp.azure.com`)
```



Request: <https://docker-4s6koalhokb7w.westeurope.cloudapp.azure.com/sysinfo4/>

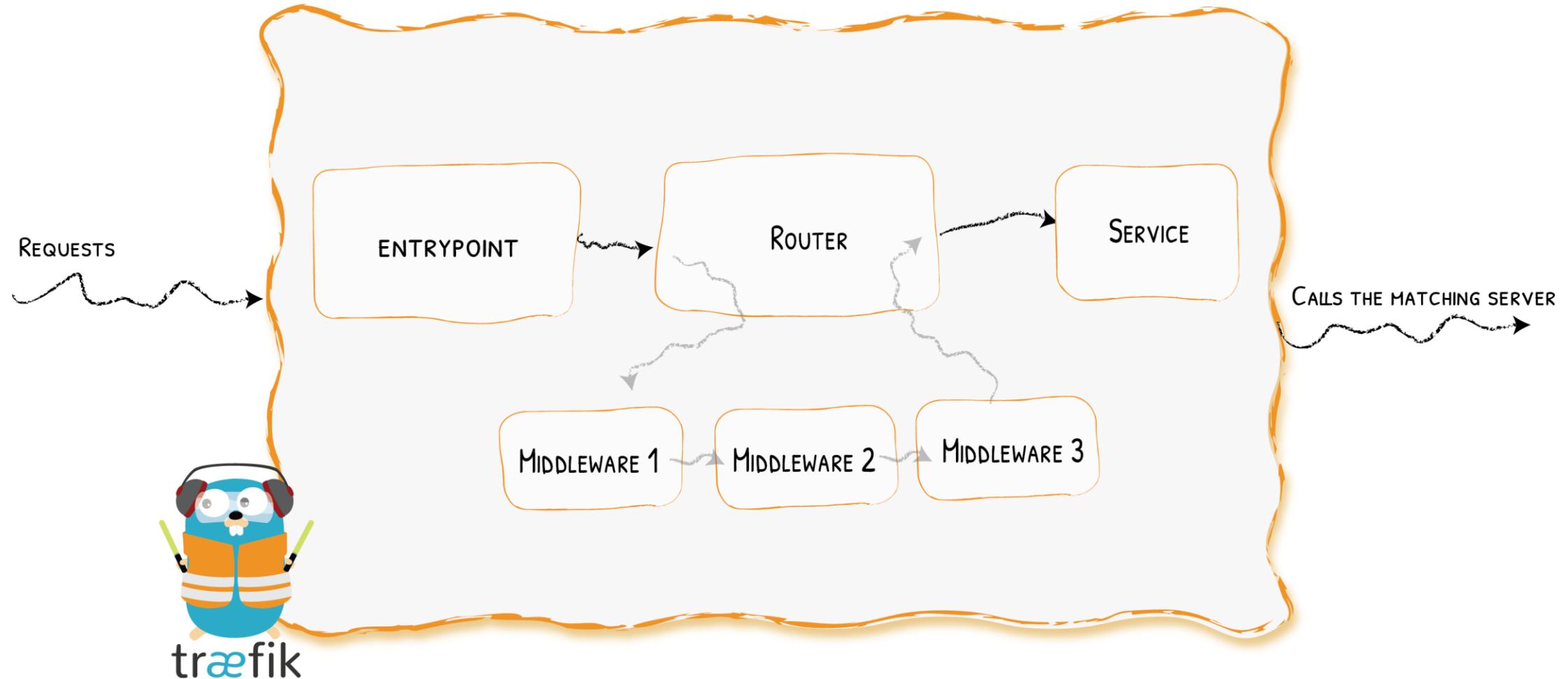




Middle-
wares

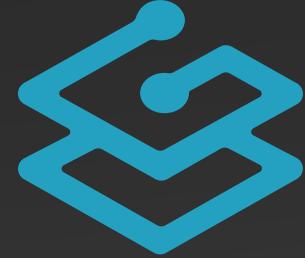


MIDDLEWARE



<https://doc.traefik.io/traefik/middlewares/overview/>





traefik
proxy



MIDDLEWARE

Available middlewares

- AddPrefix: Add a Path Prefix
- BasicAuth: Basic auth mechanism
- Buffering: Buffers the request/response
- Chain: Combine multiple pieces of middleware
- CircuitBreaker: Stop calling unhealthy services
- Compress: Compress the response
- DigestAuth: Adds Digest Authentication
- Errors Define: custom error pages
- ForwardAuth: Authentication delegation
- Headers: Add / Update headers
- IPWhiteList: Limit the allowed client IPs
- InFlightReq: Limit the number of simultaneous connections
- PassTLSClientCert: Adding Client Certificates in a Header
- RateLimit: Limit the call frequency
- RedirectScheme: Redirect easily the client elsewhere
- RedirectRegex: Redirect the client elsewhere
- ReplacePath: Change the path of the request
- ReplacePathRegex: Change the path of the request
- Retry: Automatically retry the request in case of errors
- StripPrefix: Change the path of the request
- StripPrefixRegex: Change the path of the request



Any
Questions?





Keep learning with
#GlobalAzure!

