Modern HTTP Routing

LISA 2018

2018-10-31

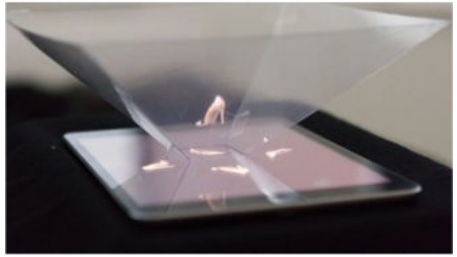
SANDOR SZÜCS

@sszuecs

teapot engineer

WE ARE CONSTANTLY INNOVATING TECHNOLOGY

HOME-BREWED, CUTTING-EDGE & SCALABLE
technology solutions



help our brand to
WIN ONLINE

~ 2,000
employees from

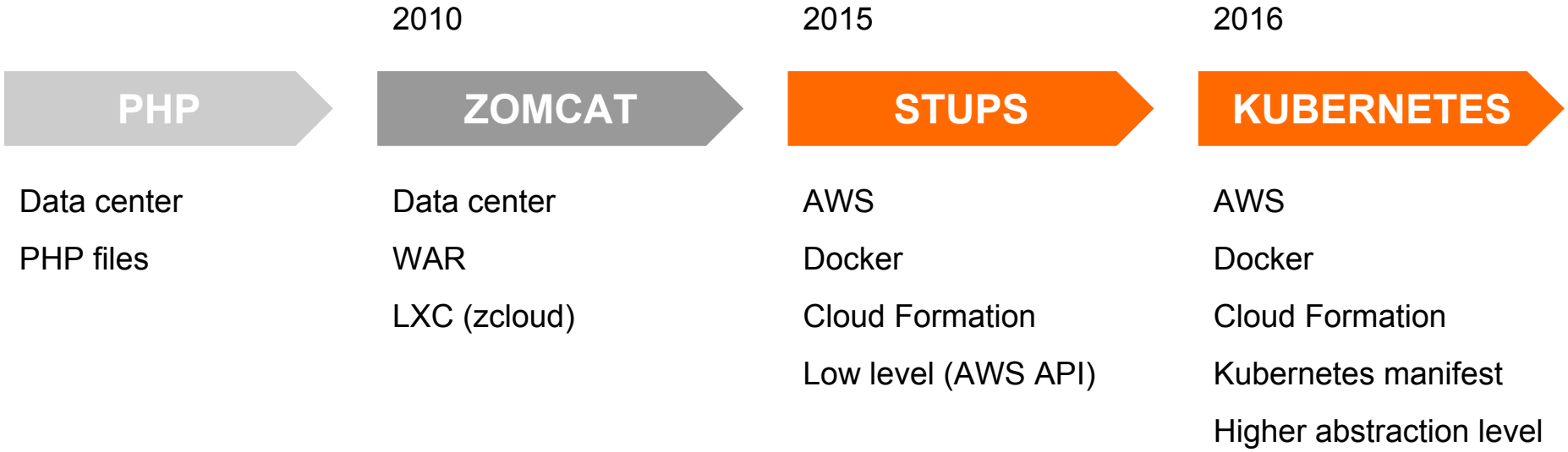
77
nations

7 tech locations
(HQs in Berlin)

zalando

ZALANDO TECH'S INFRASTRUCTURE

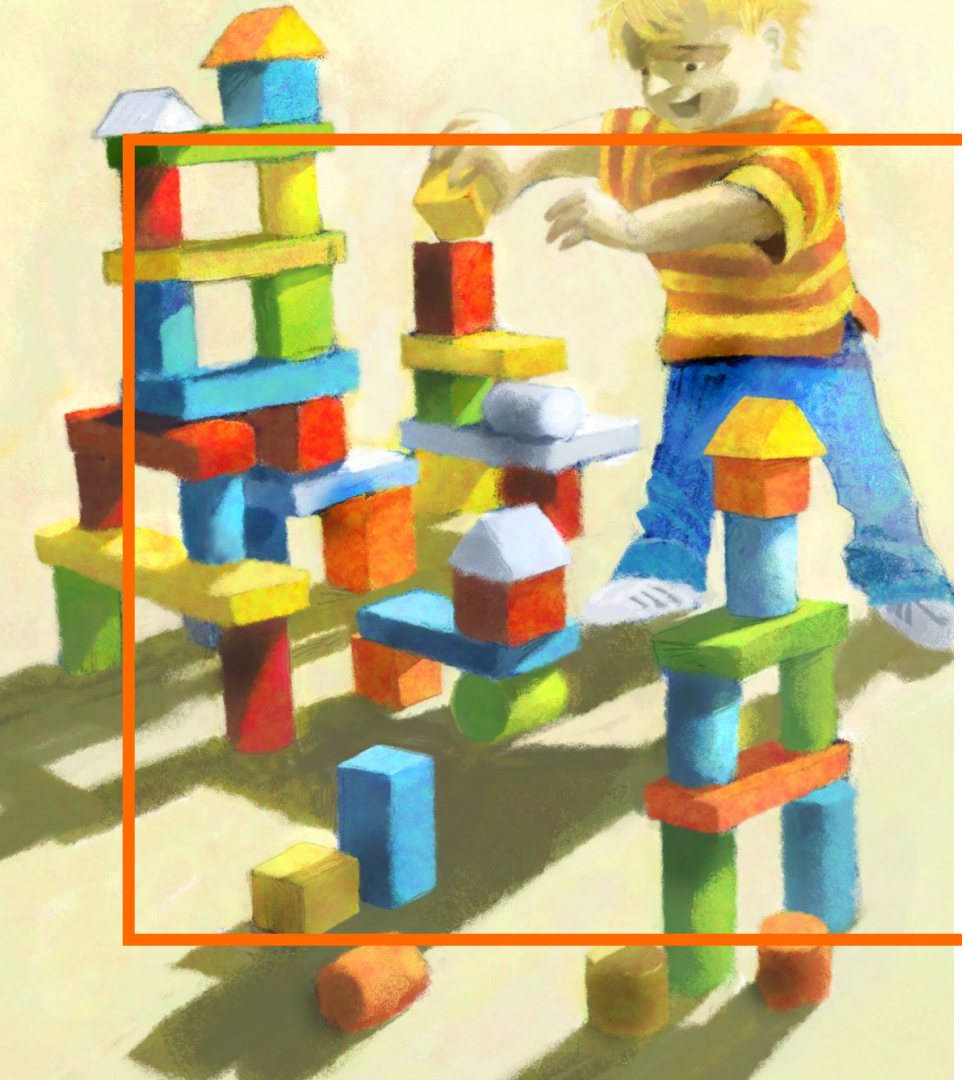
FOUR ERAS AT ZALANDO TECH



zalando

LARGE SCALE?

WORKER NODES	TEAMS	APPLICATIONS	PODS
719	134	1448	15339



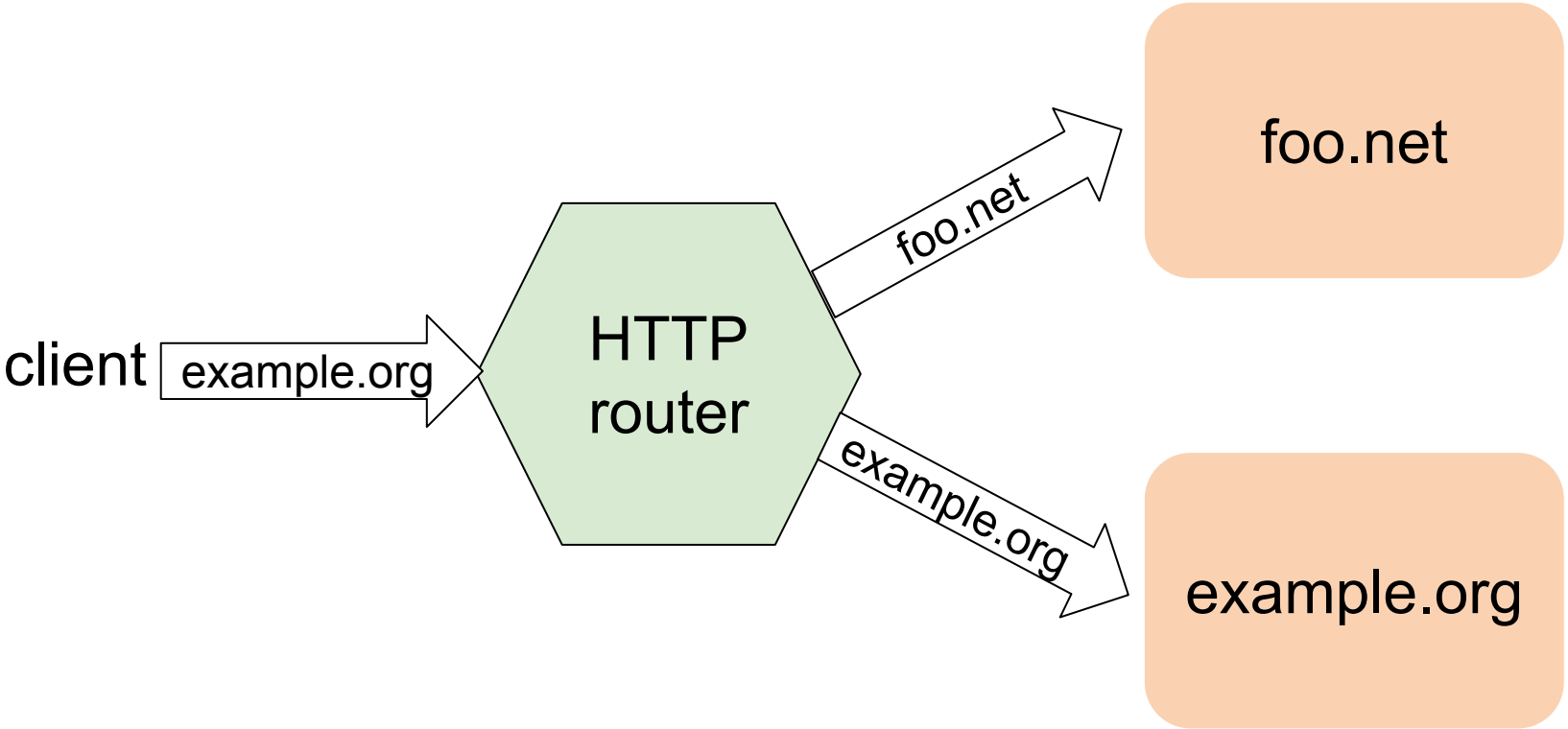
HTTP Routing

HTTP Routing

- `% curl https://example.org/a/path?q=value`
- HTTP headers

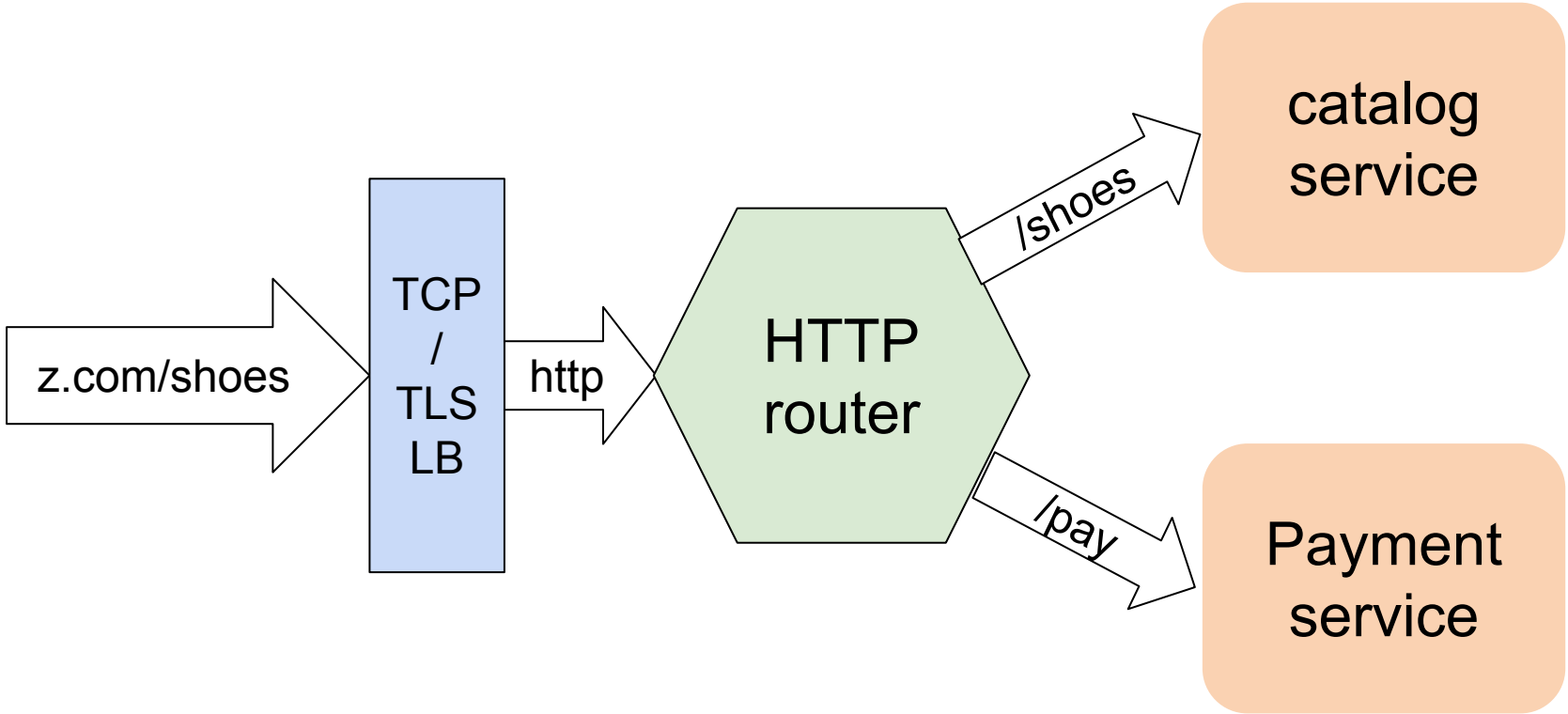
```
GET /a/path?q=value HTTP/1.1
Host: example.org
User-Agent: curl/7.49.0
Accept: */*
Cookie: Zm9vCg==
Authorization: Bearer <token>
```

HTTP Routing by Host header



zalando

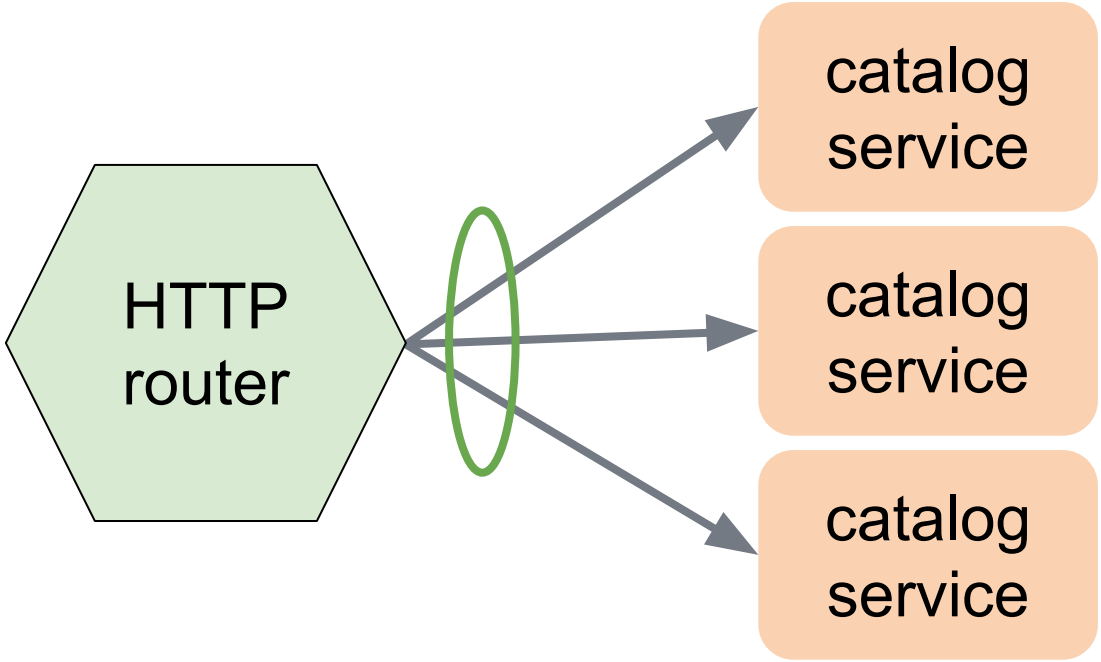
HTTP Routing by path



zalando

HTTP Router basic features

- Healthchecks
- Metrics
- Access logs
- opentracing



Modern HTTP routing

zalando

Modern HTTP Routing

- Many possibilities
  - Visibility (logs, metrics, tracing)
  - Change requests and responses
  - Resiliency (ratelimits, circuitbreaker)
  - blue / green deployments
  - Shadow traffic (clone)
  - A/B tests
  - Authnz
  - API Gateway
  - Kubernetes

Skipper a modern HTTP router

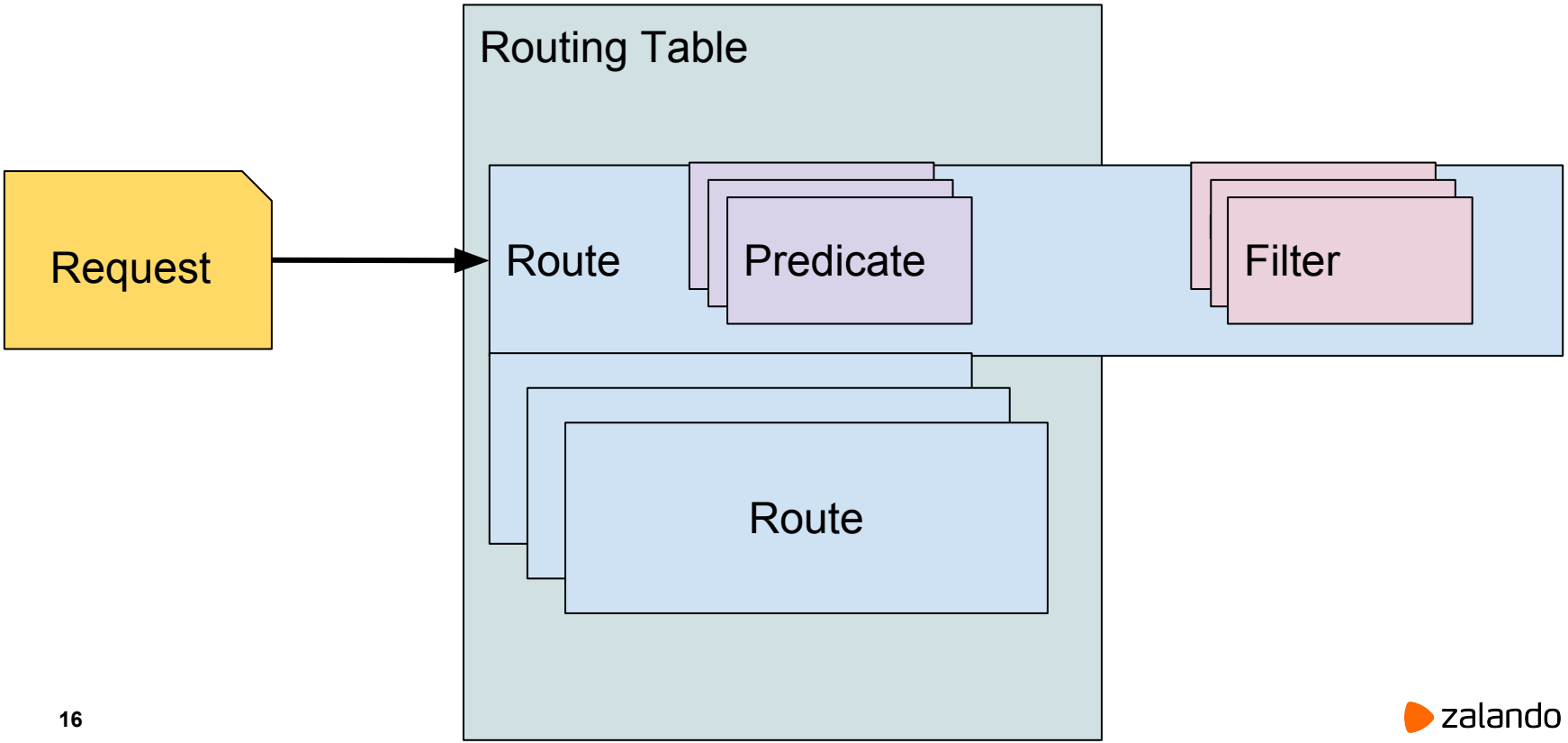
zalando

https://github.com/zalando/skipper

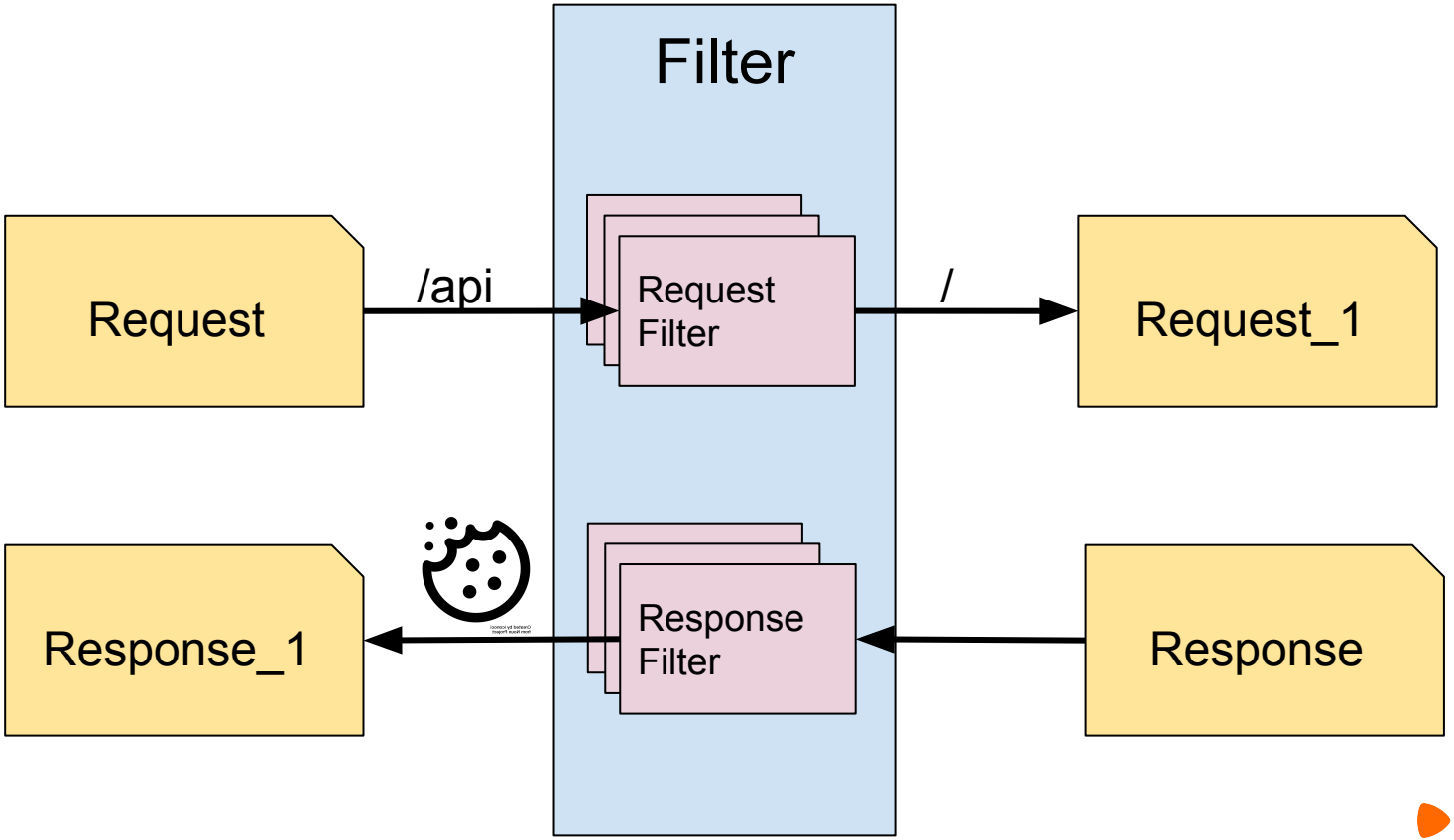
Extending Skipper

- HTTP router binary
- 1st class library
- Go Plugins
- Lua script support

Skipper: Predicate



Skipper: Filter



zalando

Cookie pictogram created by iconoci from the Noun Project

Skipper Route

Eskip Syntax:

```
RouteID: Predicate1 && Predicate2
      -> Filter1 -> Filter2
      -> “http://backend/url”
```

Skipper Route

Eskip Syntax:

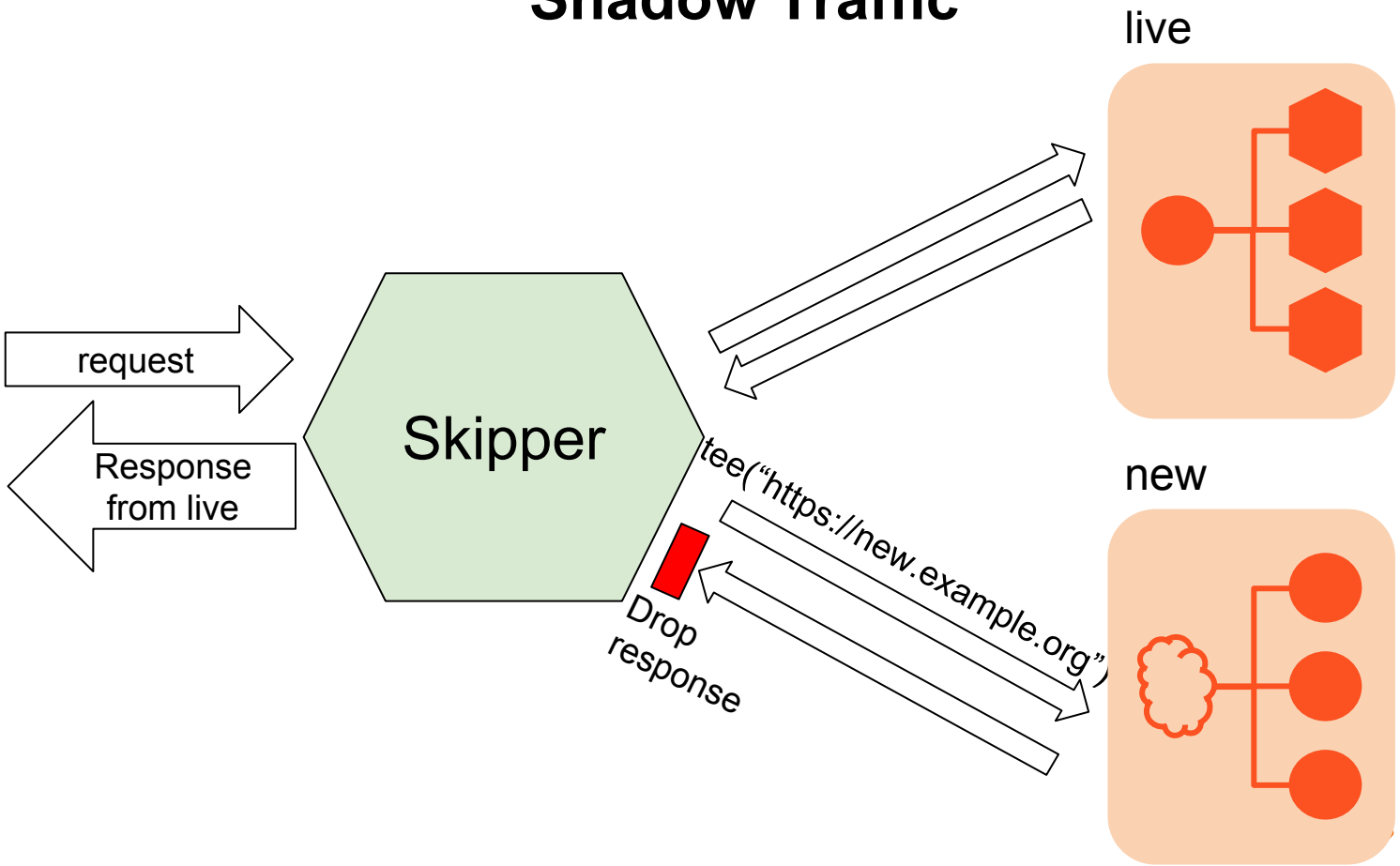
```
    RouteID: Predicate1 && Predicate2
          -> Filter1 -> Filter2
          -> "http://backend/url"

R1: Host("example.org") && Path("/api")
    ->  modPath("/api", "/")
    -> "http://backend.example.org/"
```

Ship to production - deployment patterns

- Skipper high level patterns
  - Shadow traffic
  - Blue-green deployments
  - A/B tests

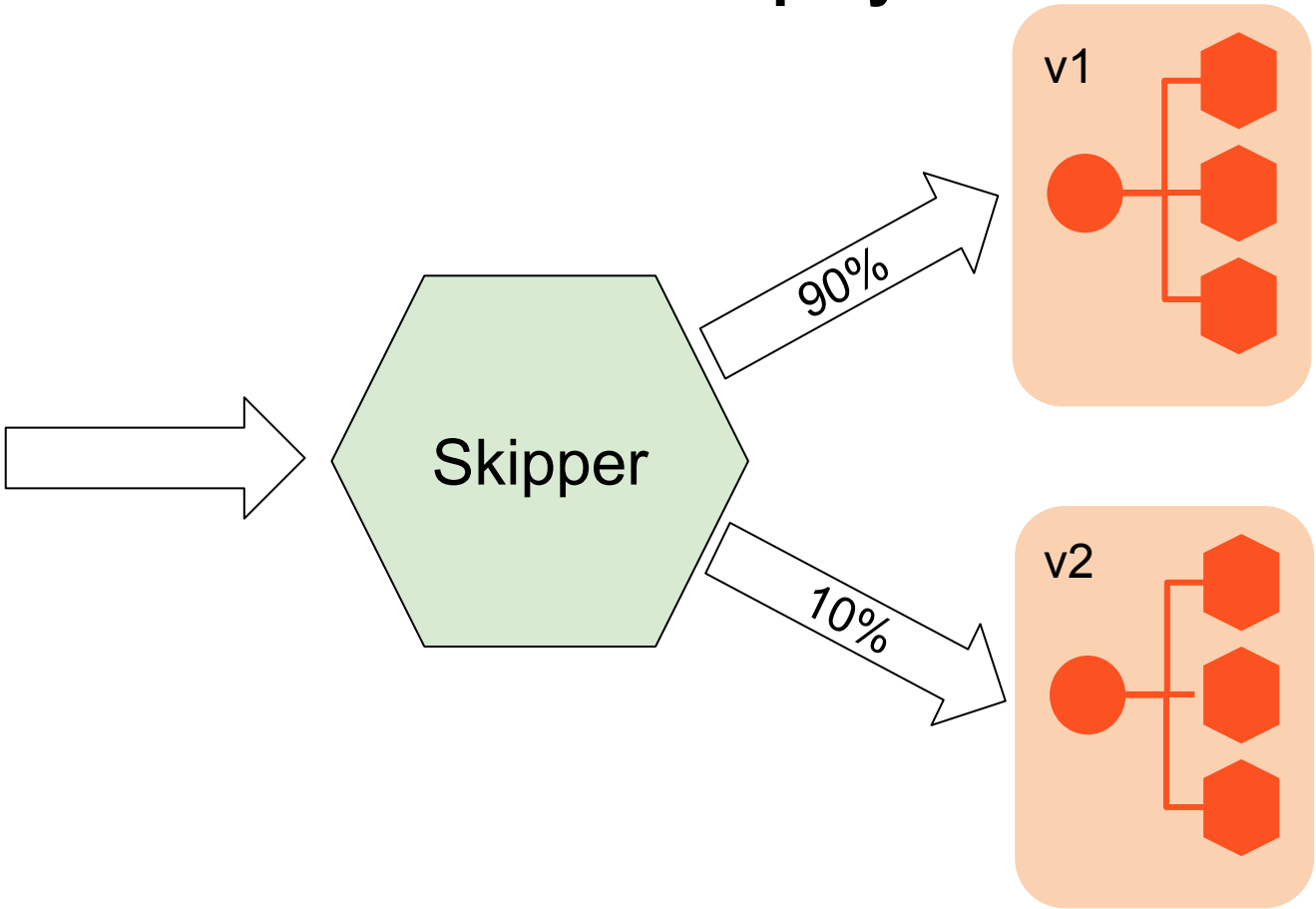
Shadow Traffic



Shadow Traffic - Route

```
R1: Host("example.org")
    -> tee("https://new.example.org")
    -> "https://backend.example.org/"
```

Blue-Green deployment



zalando

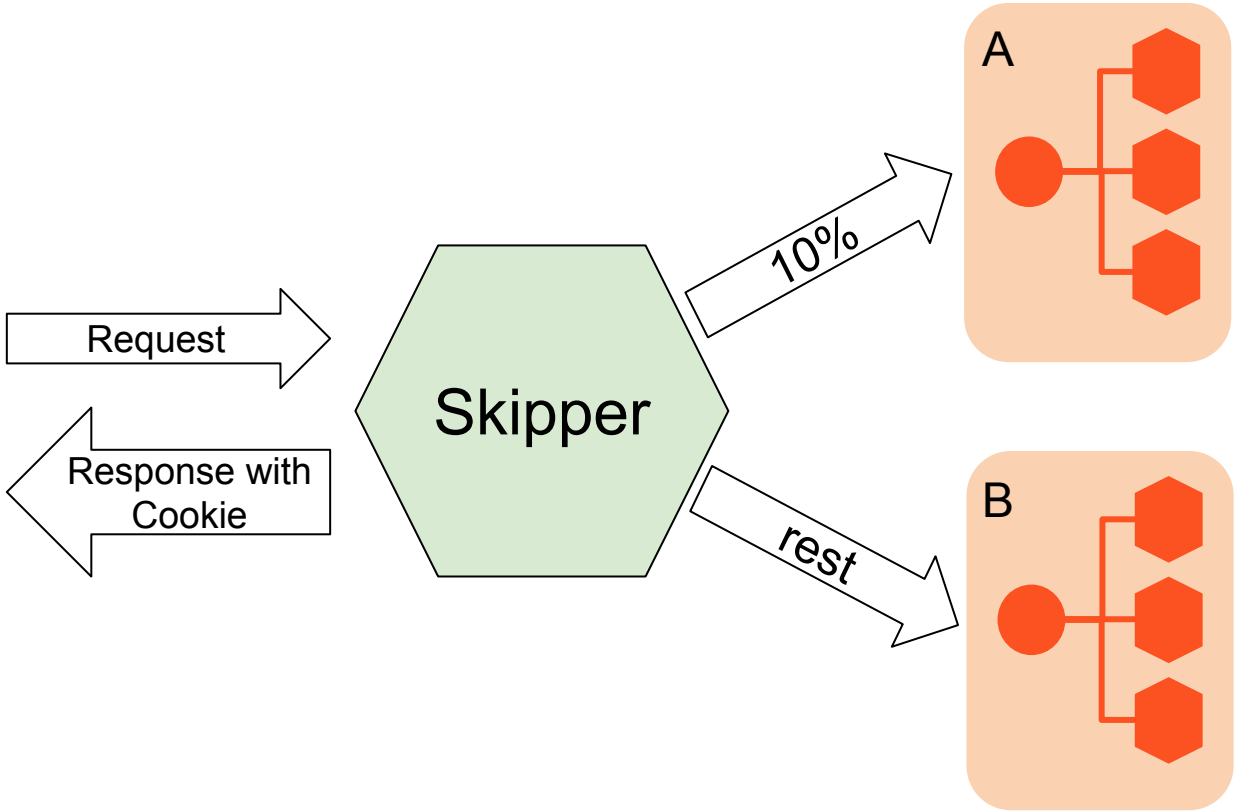
Blue-Green deployment

```
R1: Host("example.org") &&
    Traffic(0.1)
    -> "http://v2.example.org/"
R2: Host("example.org")
    -> "http://v1.example.org/"
```

Blue-Green deployment automation

https://github.com/zalando-incubator/stackset-controller

A/B tests part 1



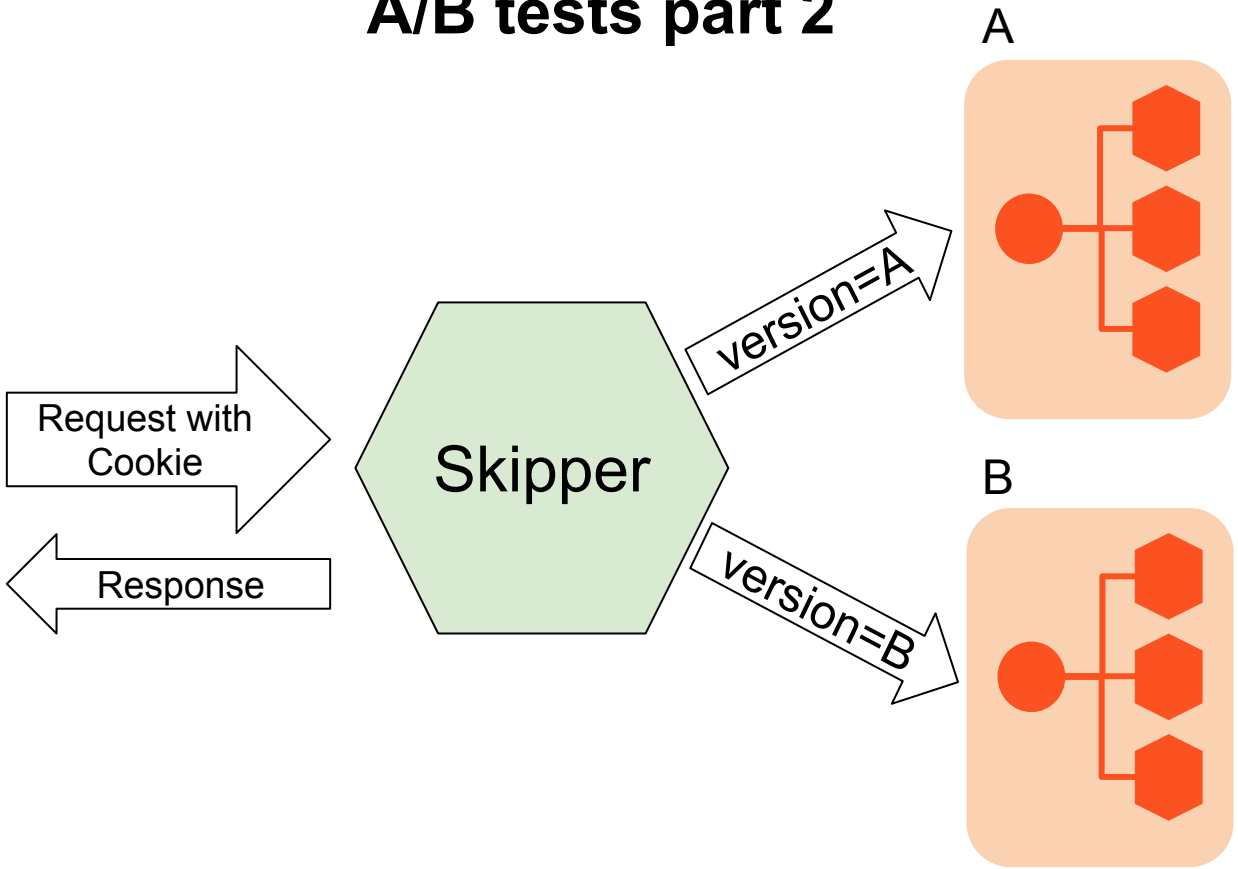
https://github.com/zalando/skipper

A/B test

2 initial routes required which set a Cookie

```
R1: Traffic(0.1)
    -> responseCookie("version", "A")
    -> "http://serviceA/"
R2: *
    -> responseCookie("version", "B")
    -> "http://serviceB/"
```

A/B tests part 2



https://github.com/zalando/skipper

A/B test

2 routes with Cookie predicate

```
R3: Cookie("version", "A")
    -> "http://serviceA/"
R4: Cookie("version", "B")
    -> "http://serviceB/"
```

Visibility

- Metrics
- Opentracing

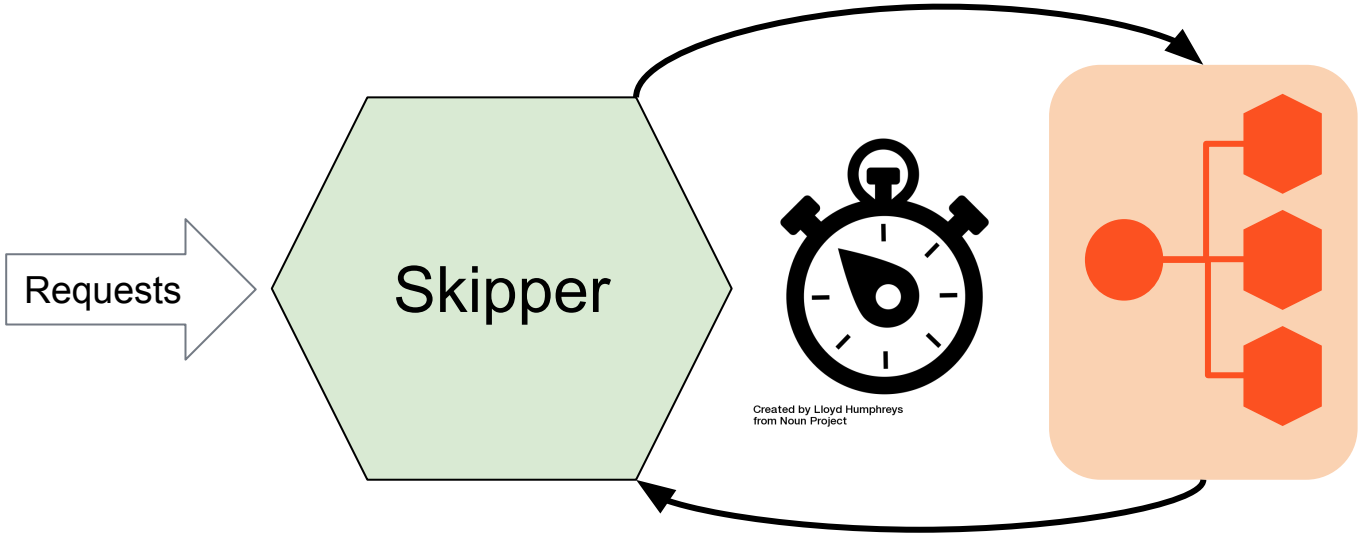
Created by Andrey Vasiliev
from Noun Project

Metrics

Backend response slow? Yes / No

zalando

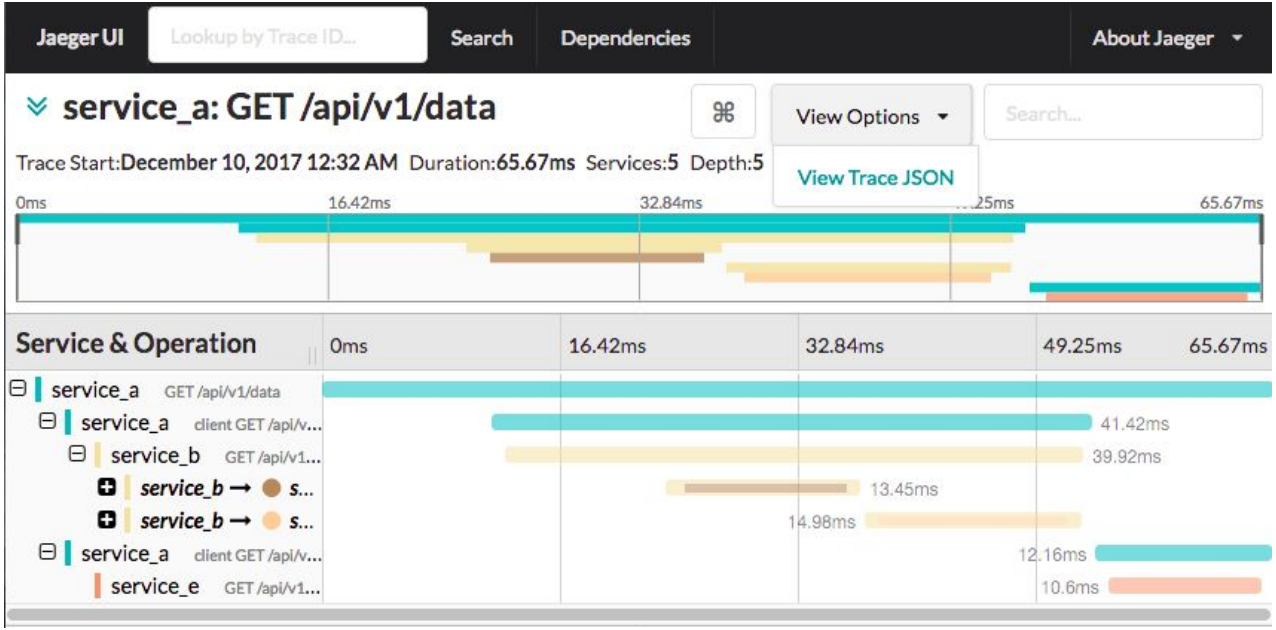
Metrics



 Thanks to Lloyd Humphreys from Noun Project creating the pictogram

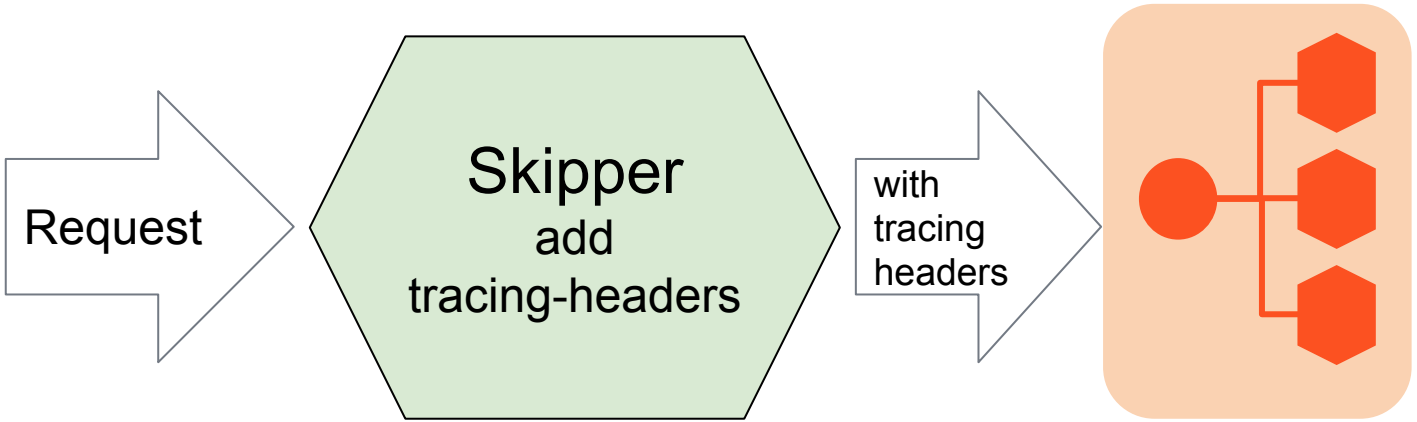
Opentracing

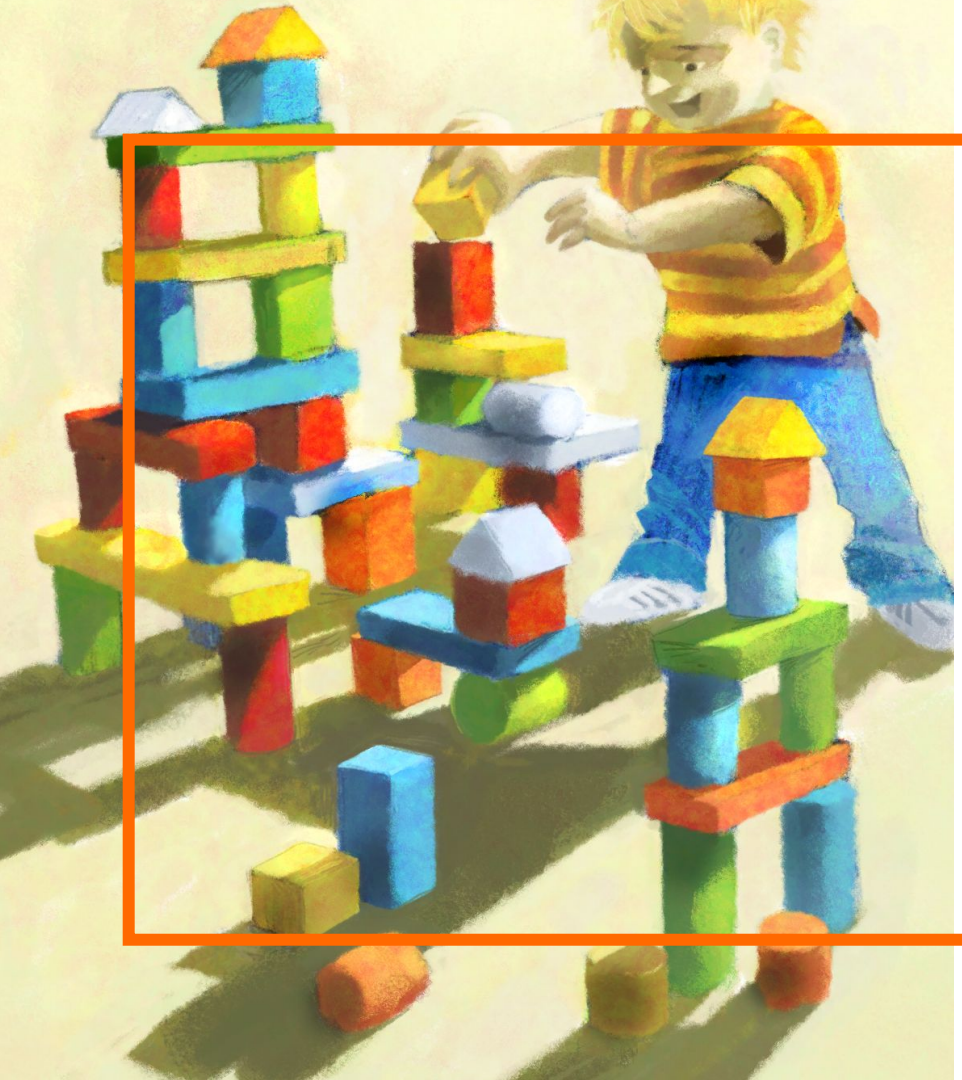
which service is slow?



zalando

Opentracing

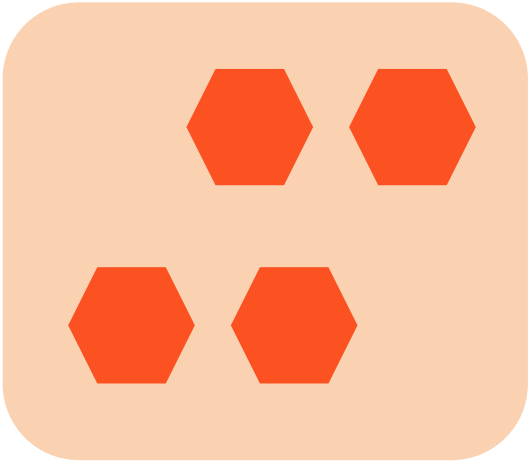




KUBERNETES

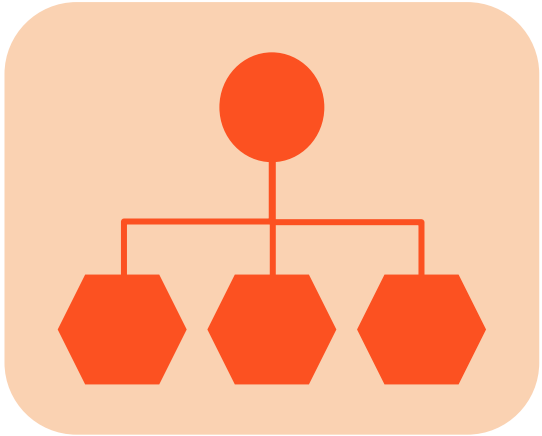
DEPLOYMENT - POD

A deployment creates
a set of Pods



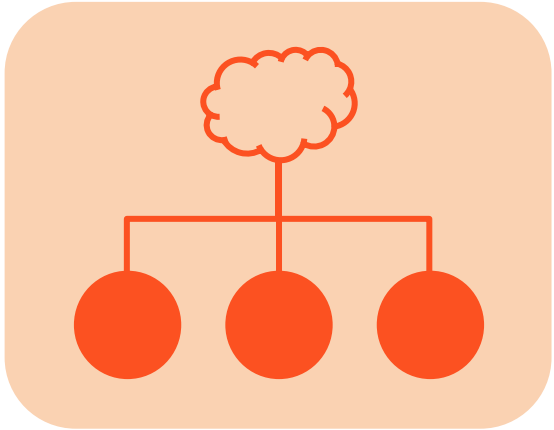
SERVICE

A service is an cluster internal TCP load balancer to Pods

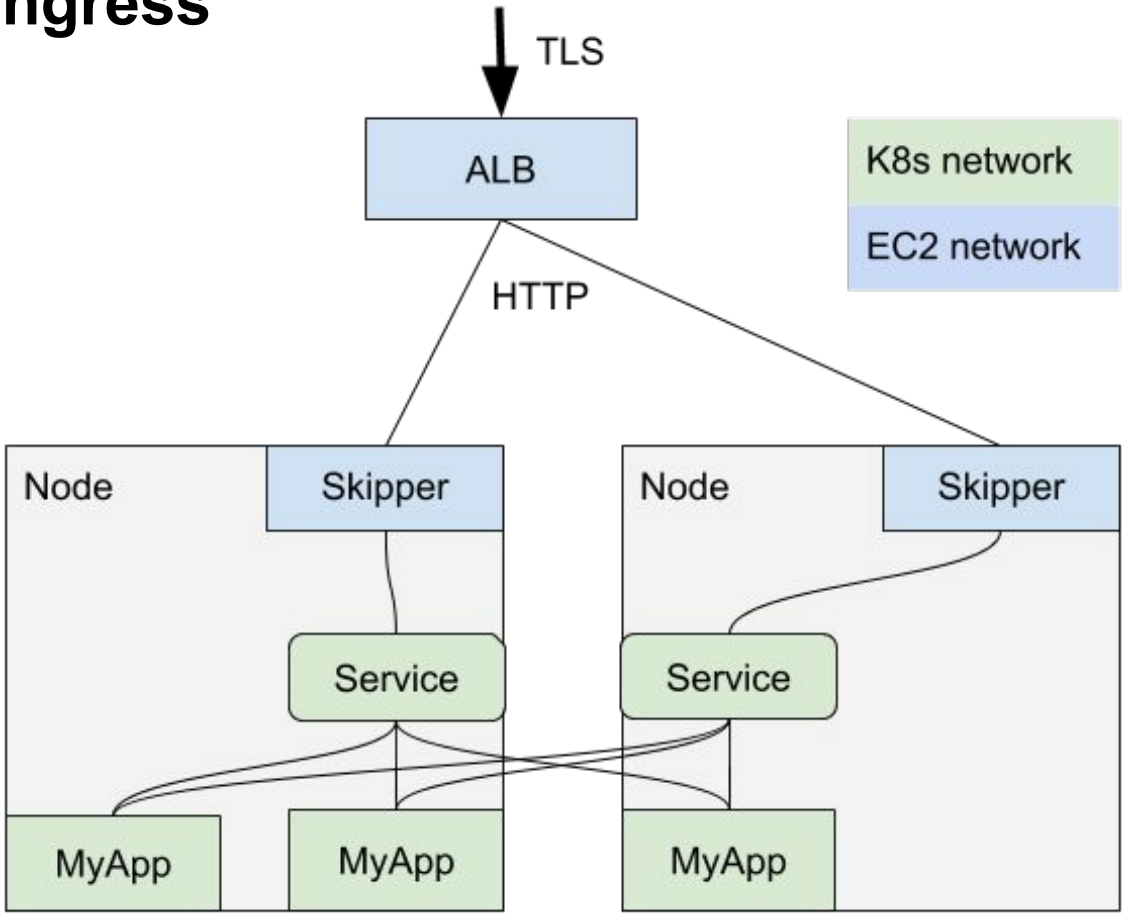


INGRESS

AN EXTERNAL ACCESS POINT
TO SERVICES
-
Configures http router

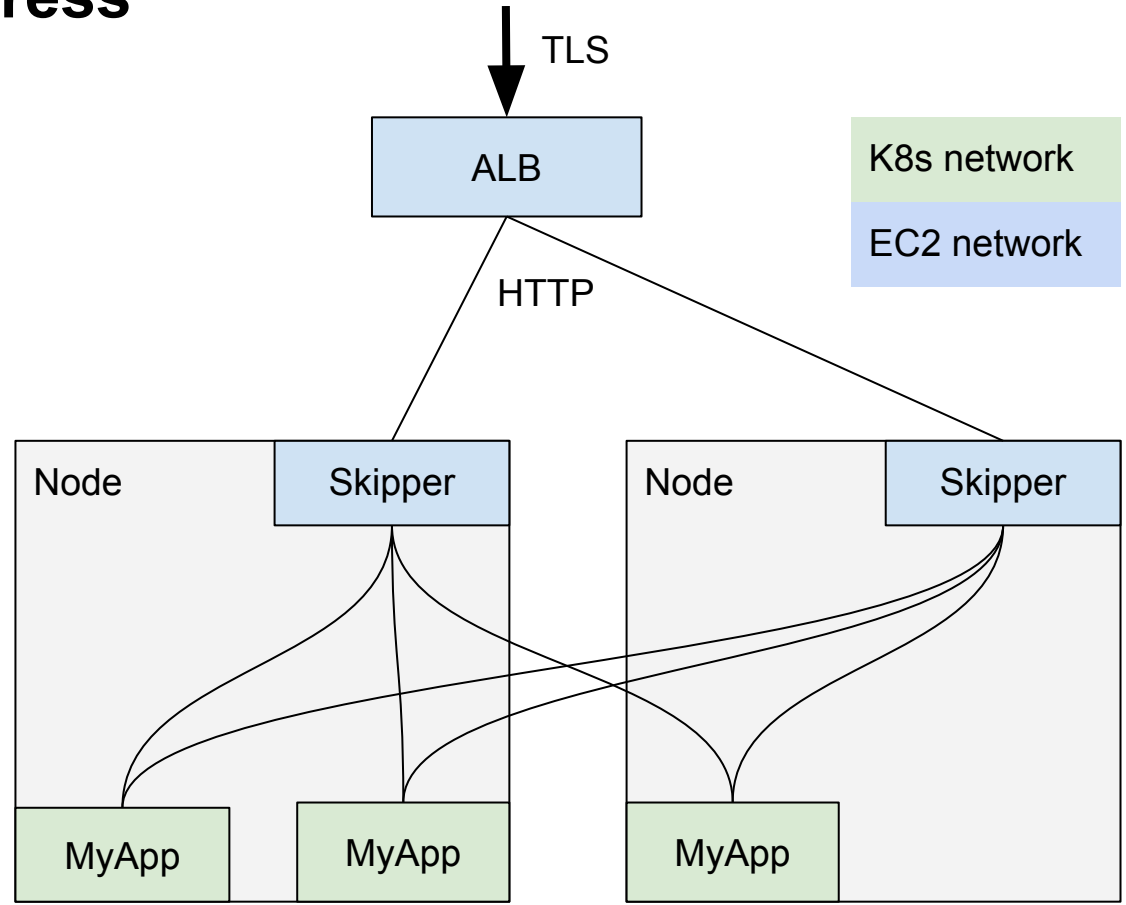


Logical Ingress



zalando

Real Ingress



zalando

CHALLENGES

-

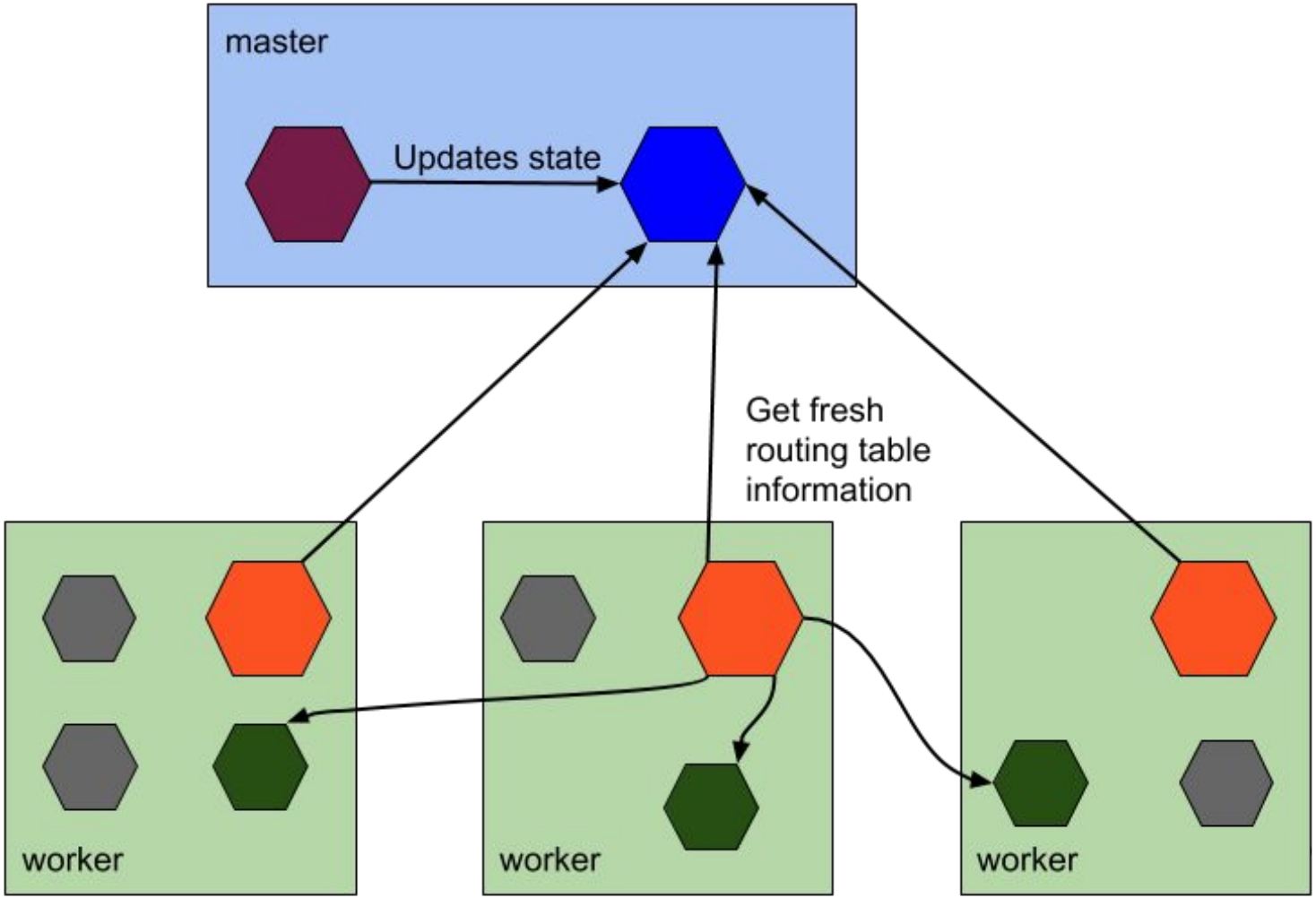
Kubernetes Ingress

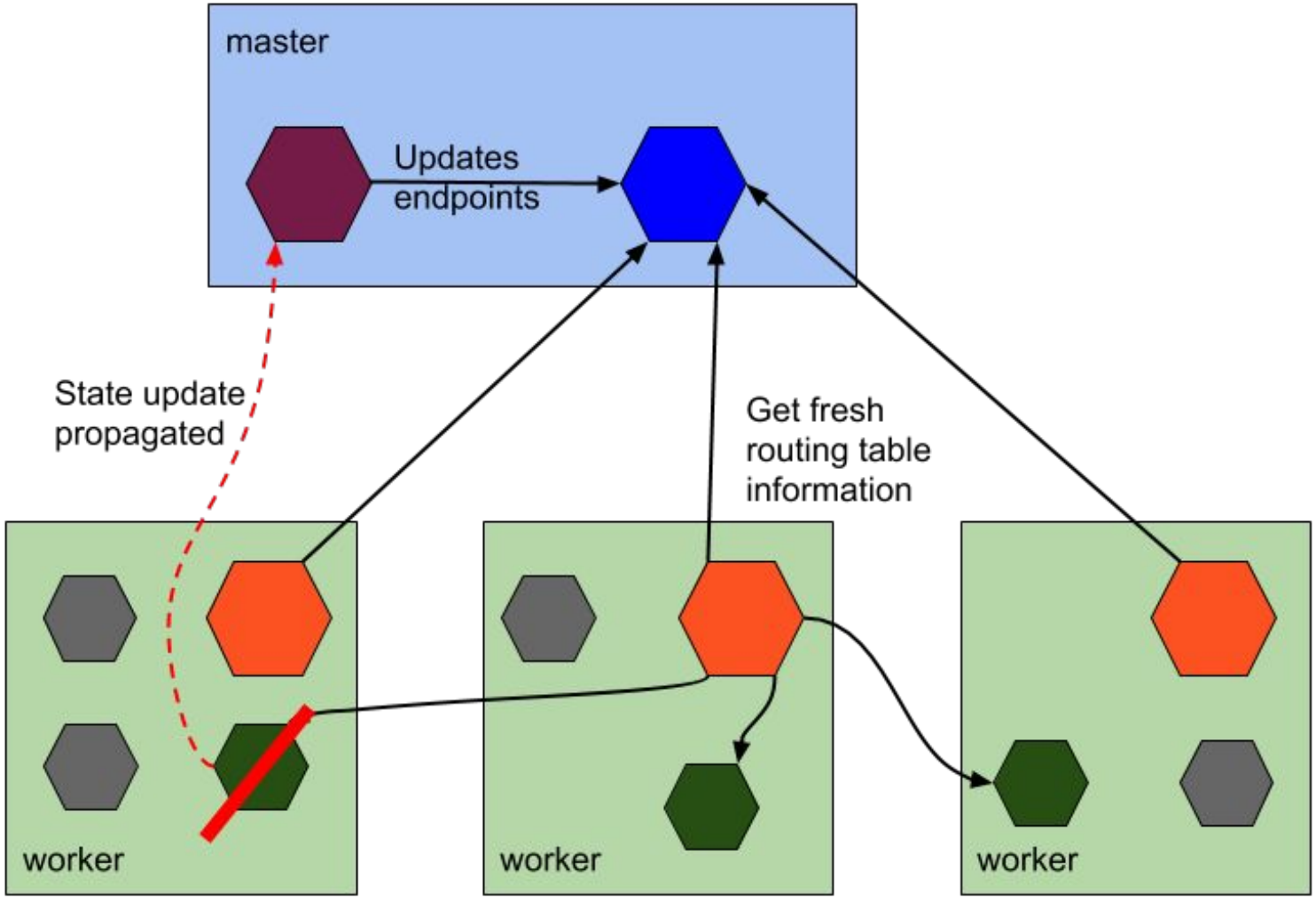
Kubernetes - challenges for ingress controllers

Planned cases

- Deployment
- Pod Autoscaling
- Cluster Autoscaling
  - Node shutdown
  - Node creation
- Pod to Node rebalancing

--> require update of pool members or routing table


master
Updates state
Get fresh
routing table
information
worker
worker
worker


master
Updates
endpoints
State update
propagated
Get fresh
routing table
information
worker
worker
worker

Kubernetes - challenges for ingress controllers

Timeouts to apiserver

- Most controllers can’t detect hanging kube-apiserver calls, because of client-go
- https://github.com/kubernetes/client-go/issues/374

Kubernetes - challenges for ingress controllers

Documentation only

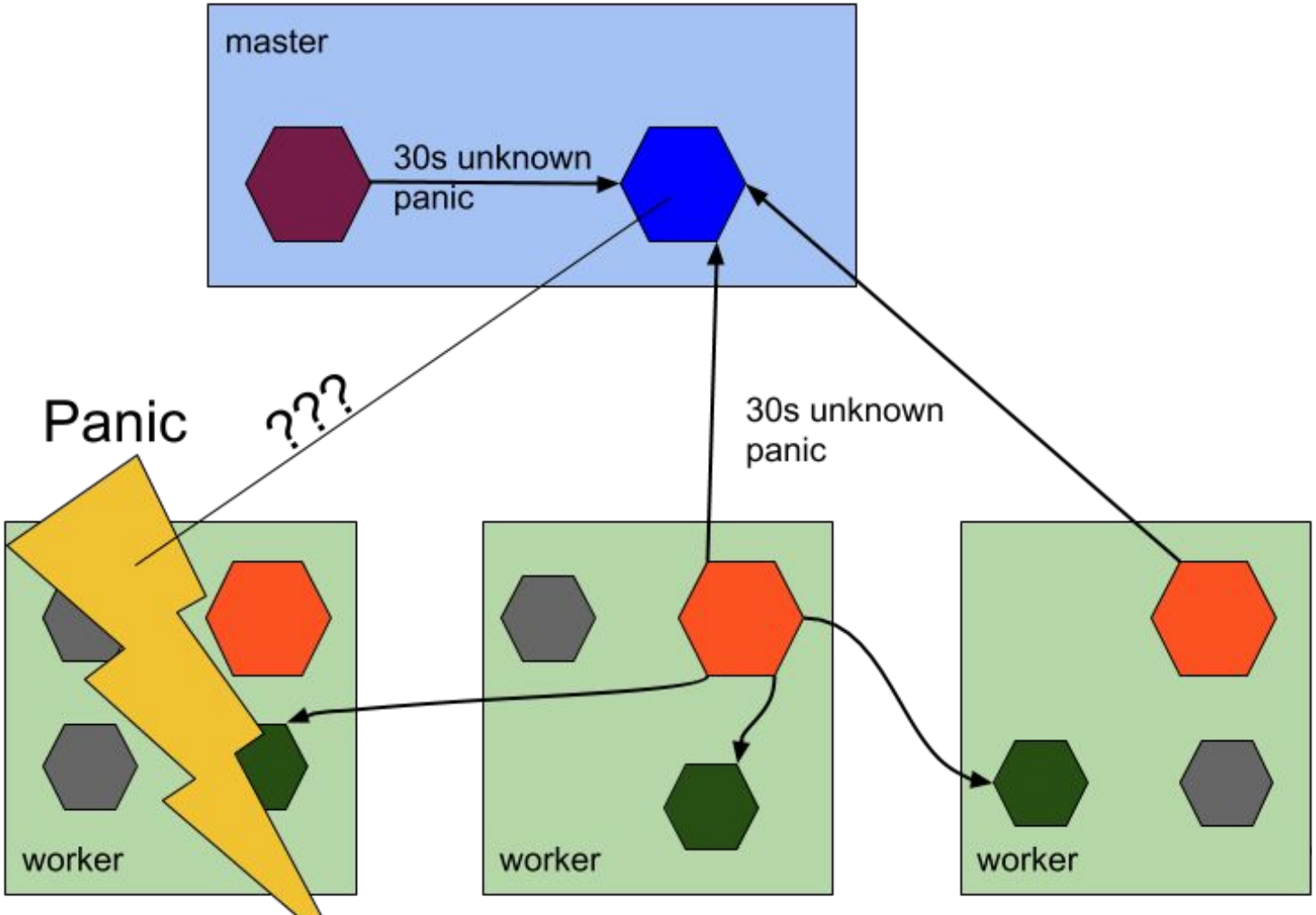
Race Conditions to populate changes

- start/stop Pod → update Endpoints
- https://opensource.zalando.com/skipper/kubernetes/ingress-backends
- change Kubernetes Service implementation on all nodes (iptables/ipvs config on each node)

Kubernetes - challenges

Unplanned

- Hanging calls to Kubernetes apiserver
- kernel panics → workers can **not** update master
- Cloud provider node terminations


master
30s unknown
panic
Panic
???
30s unknown
panic
worker
worker
worker

Kubernetes - solutions for ingress controllers

Faster http routing table updates

- Autoscaling
- Deployments are online

Kubernetes - solutions for ingress controllers

Faster http routing table updates

- Autoscaling
- Deployments are online

Observe connections

- Broken endpoint detection → mark as dead
- Retry connection

@sszuecs | Open Source

Skipper HTTP Ingress Router
https://github.com/zalando/skipper

Skipper documentation

https://zalando.github.io/skipper

Kubectl plugin skipper
https://github.com/szuecs/kubectl-plugins

Kube AWS Ingress Controller
https://github.com/zalando-incubator/kube-ingress-aws-controller

External DNS
https://github.com/kubernetes-incubator/external-dns

Zalando Cluster Configuration
https://github.com/zalando-incubator/kubernetes-on-aws

zalando

QUESTIONS?

SANDOR SZÜCS

TECH INFRASTRUCTURE

TEAPOT ENGINEER

sandor.szuecs@zalando.de

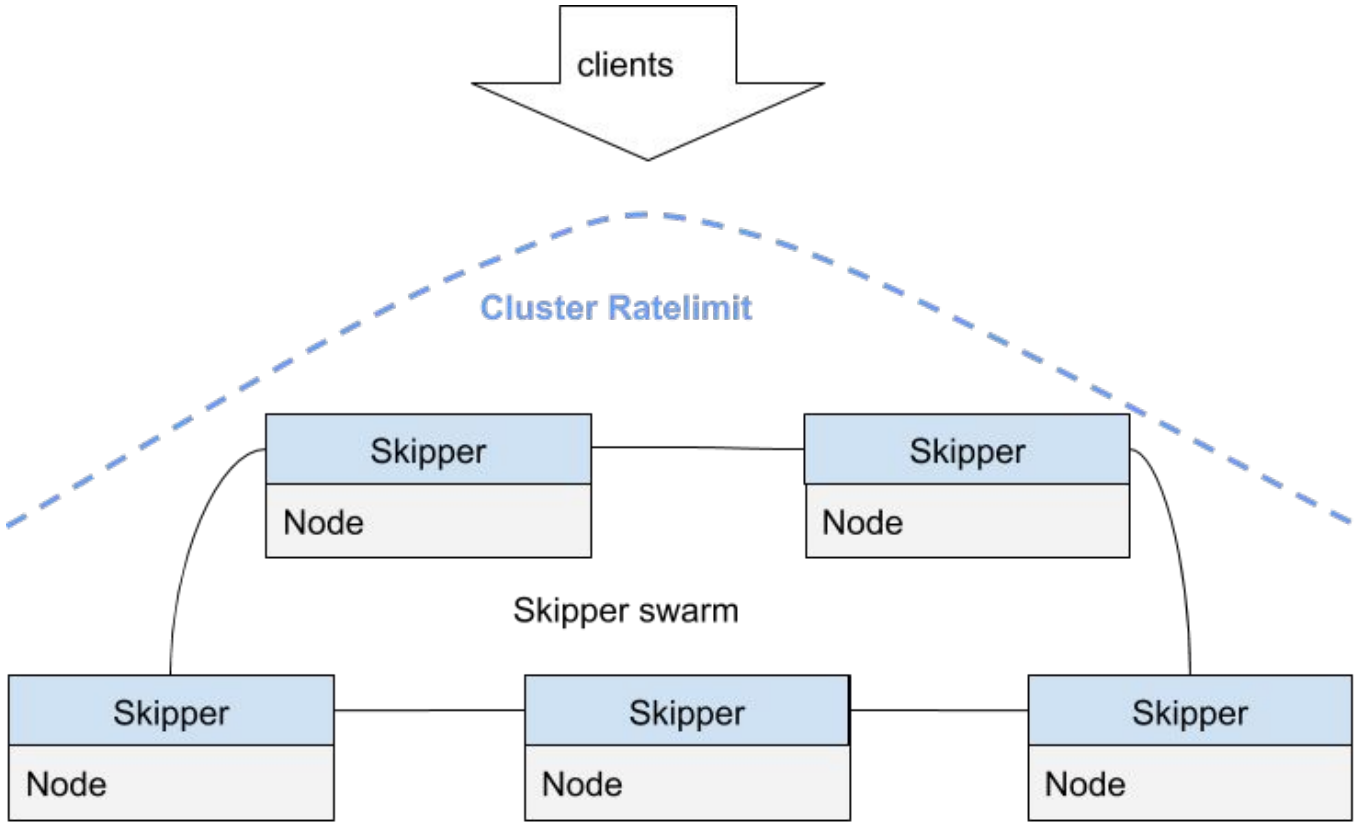
@sszuecs

Illustrations by @01k

One last thing

Cluster Ratelimits

Cluster Ratelimit

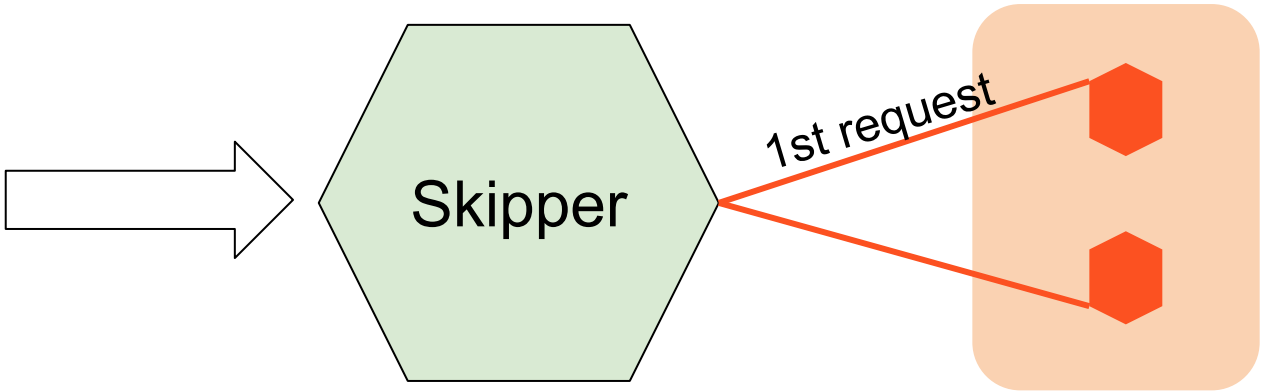


zalando

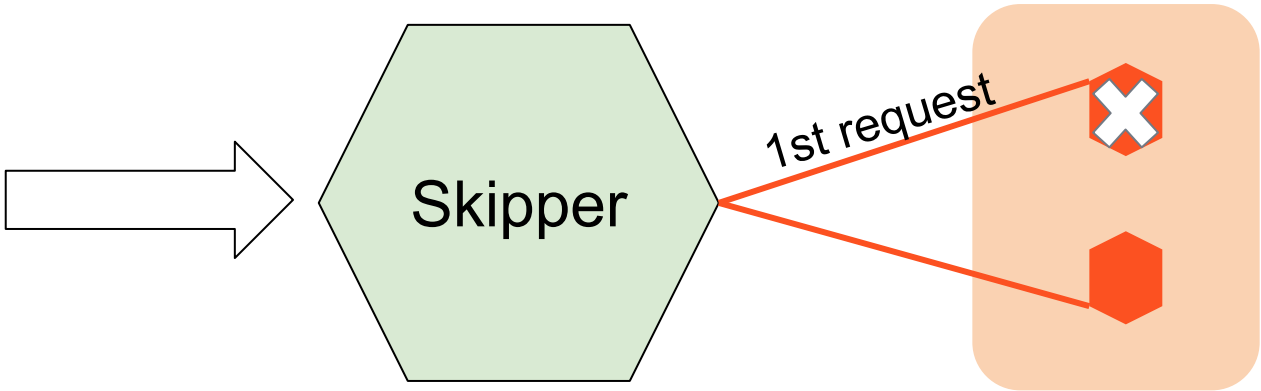
Kubernetes - solutions

Retries

Retry



Retry



Retry