Zuul - Netflix API Gateway that Handles 2M Requests Per Second

Susheel Aroskar
Cloud Gateway - Netflix

NETFLIX

Amazon AWS S3 outage is breaking things for a lot of websites and apps

Darrell Etherington @etherington / Feb 28, 2017

Comment

OPINION

Why Netflix didn't sink when Amazon S3 went down

When a typo caused an outage in part of Amazon Simple Storage Service's (S3) cloud infrastructure on February 28, many of their customers' websites went down but Netflix survived unblemished.

How??

How??

Zuul shifted all the traffic from the affected us-east-1 region to other healthy AWS regions

https://github.com/Netflix/zuul

Susheel Aroskar

Senior Software Engineer
Cloud Gateway
saroskar@netflix.com

github.com/raksoras
@susheelaroskar

Agenda

Agenda

- What is Zuul?

Agenda

- What is Zuul?
- What Zuul can do for you?

Agenda

- What is Zuul?
- What Zuul can do for you?
- How Zuul does it?

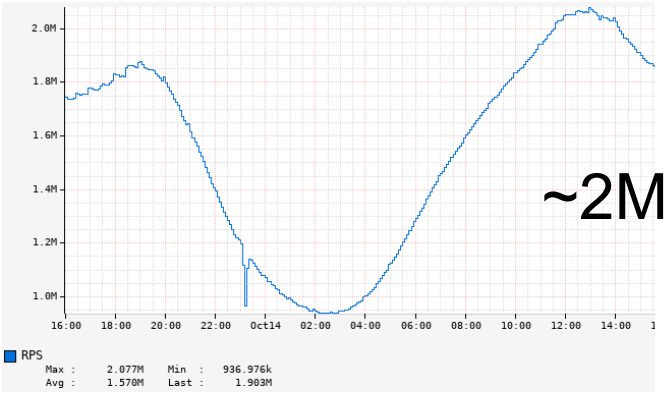
Agenda

- What is Zuul?
- What Zuul can do for you?
- How Zuul does it?
- Future roadmap

What Is Zuul?

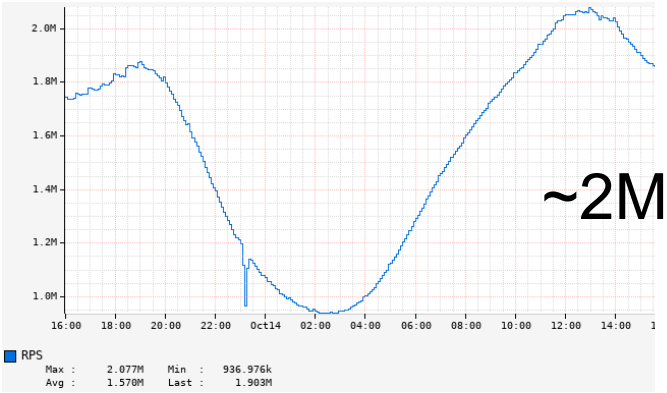
Zuul Introduction

Zuul is Netflix’s API Gateway

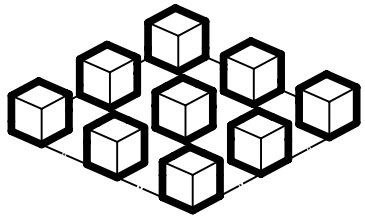


~2M RPS

Zuul is Netflix’s API Gateway

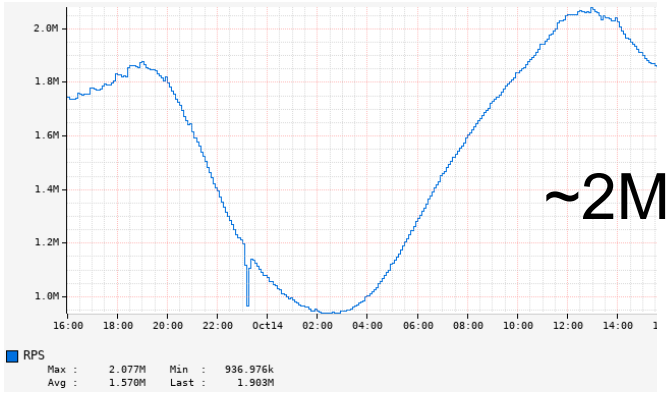


~2M RPS

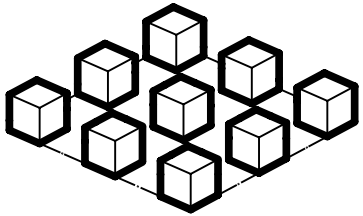


72 Clusters

Zuul is Netflix’s API Gateway

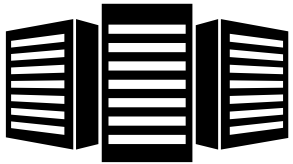


~2M RPS

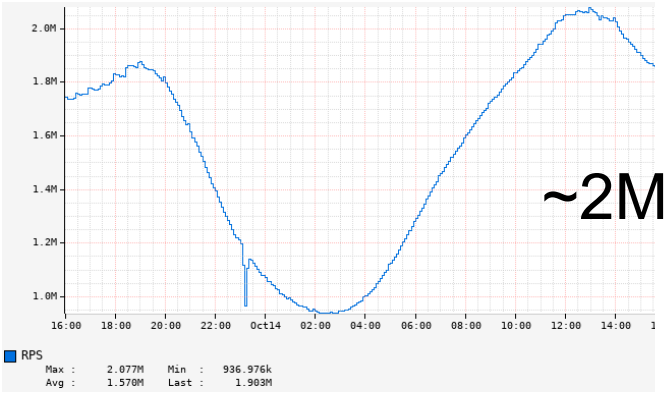


72 Clusters

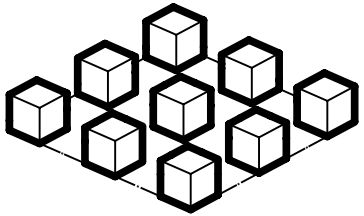
Zuul is Netflix's API Gateway



6000 servers

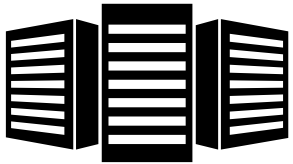


~2M RPS

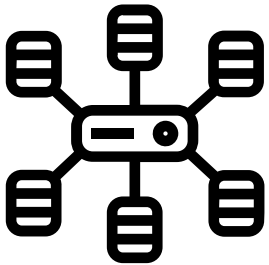


72 Clusters

Zuul is Netflix's API Gateway



6000 servers



150 Backend Services

What Can Zuul Do for you?

Zuul Features

Core

Responsibilities

Route Traffic

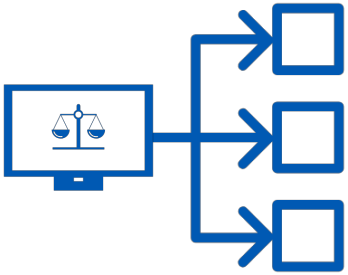
Core

Responsibilities

Route Traffic

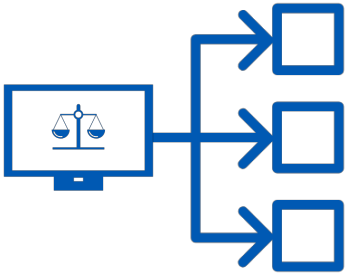
Core

Responsibilities



Load Balance

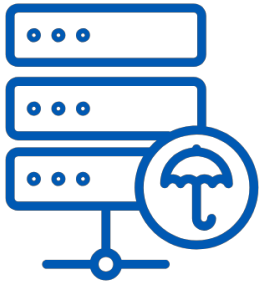
Route Traffic



Load Balance

Core

Responsibilities

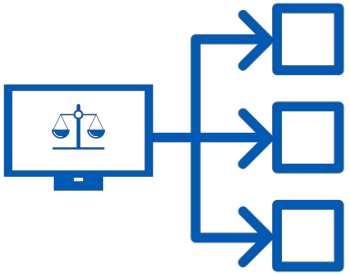


Protect Origins

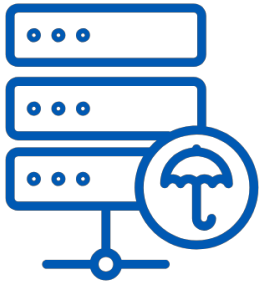
Route Traffic

Core

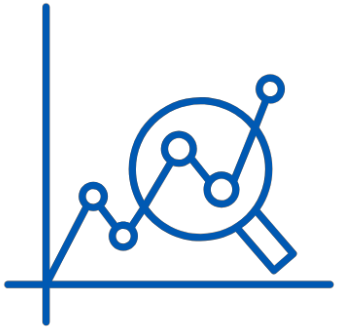
Responsibilities



Load Balance



Protect Origins



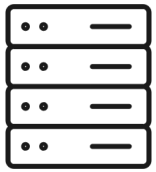
Metrics and Monitoring

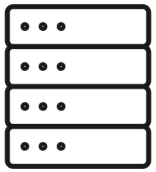
1. Route Traffic

CORE
RESPONSIBILITIES

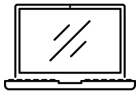
NETFLIX

Login Service

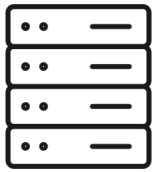


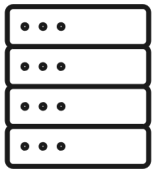


Playback Service



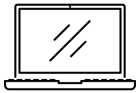
Login Service

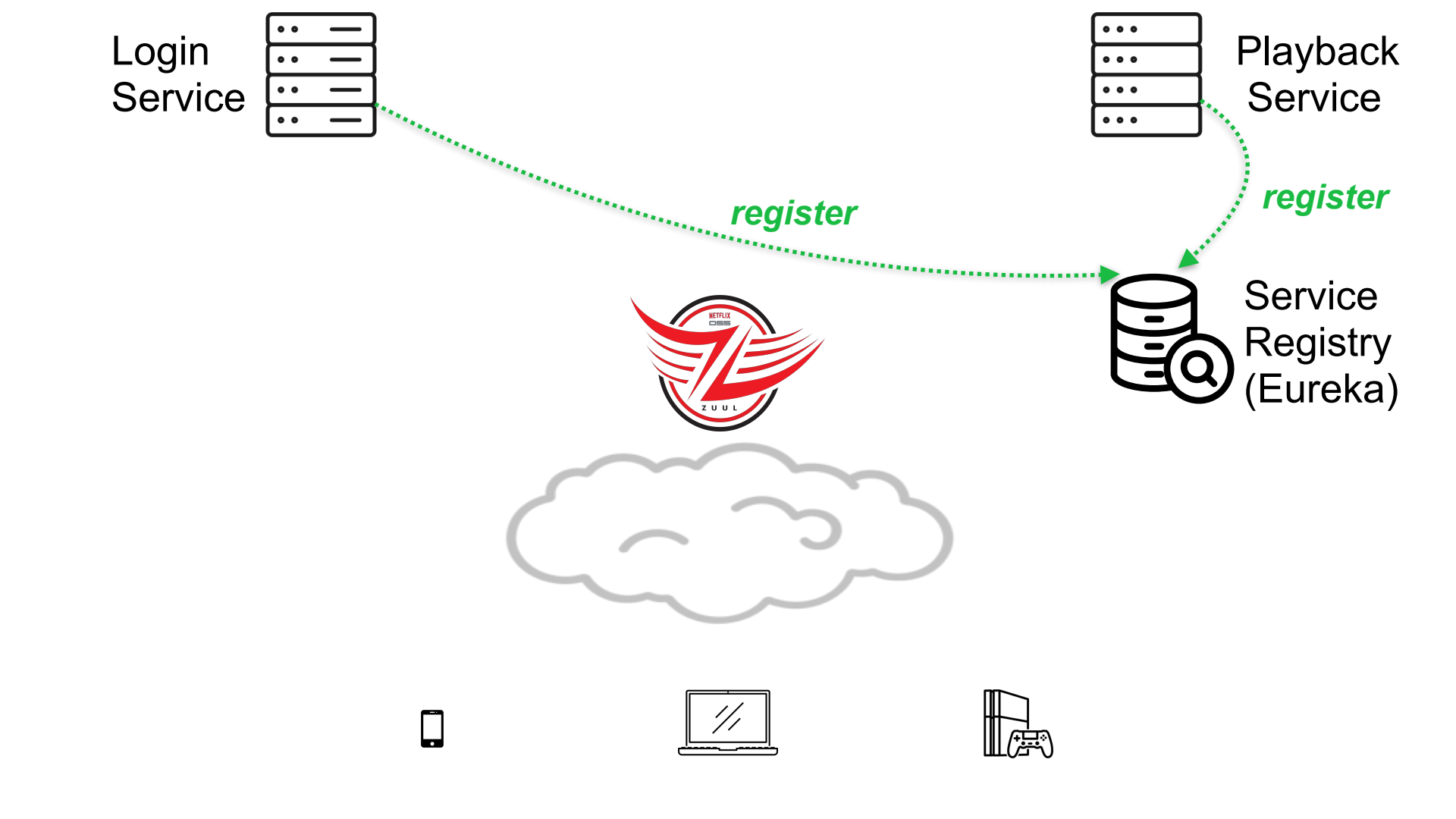


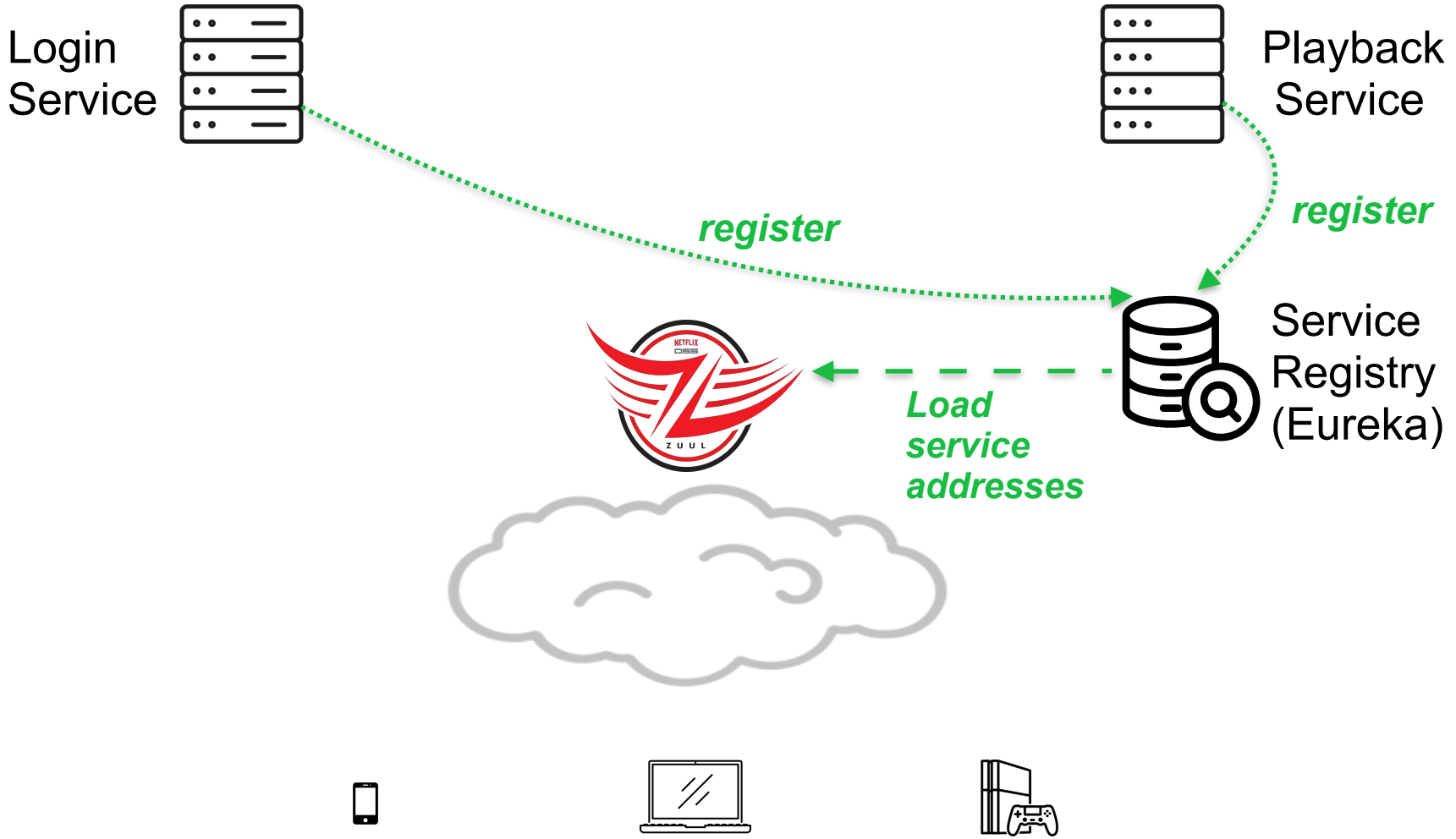


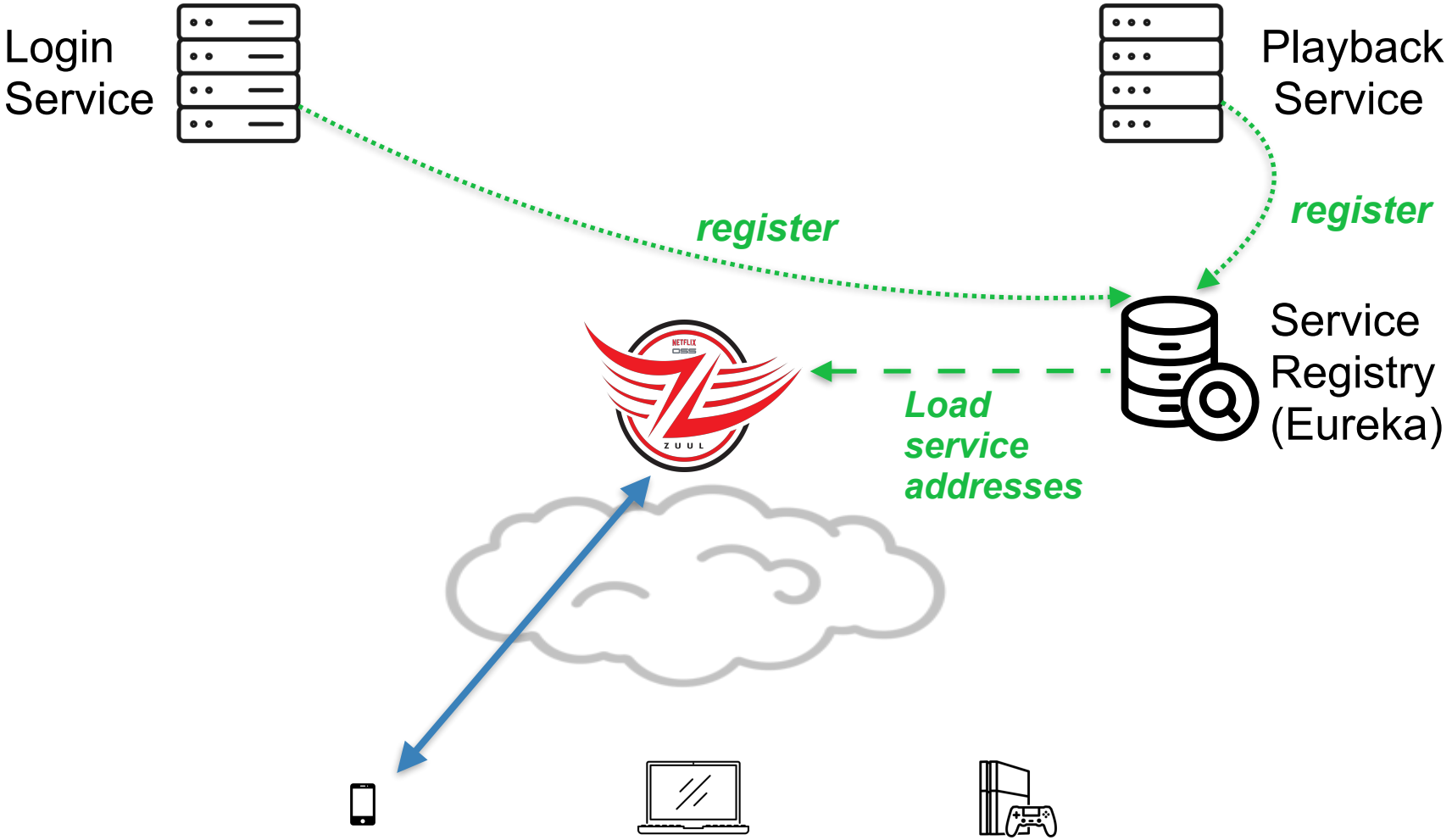
Playback Service

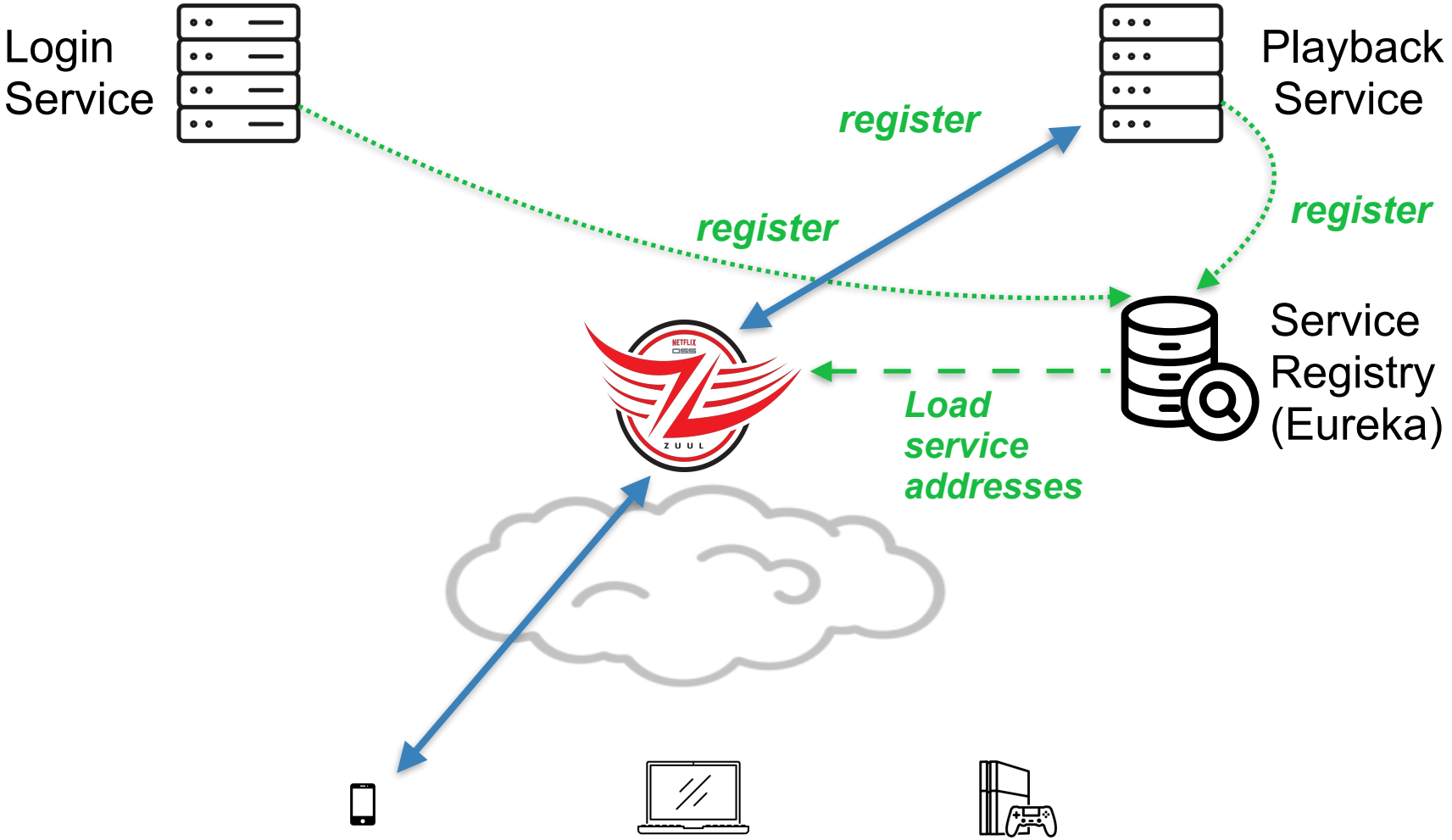
Service Registry (Eureka)

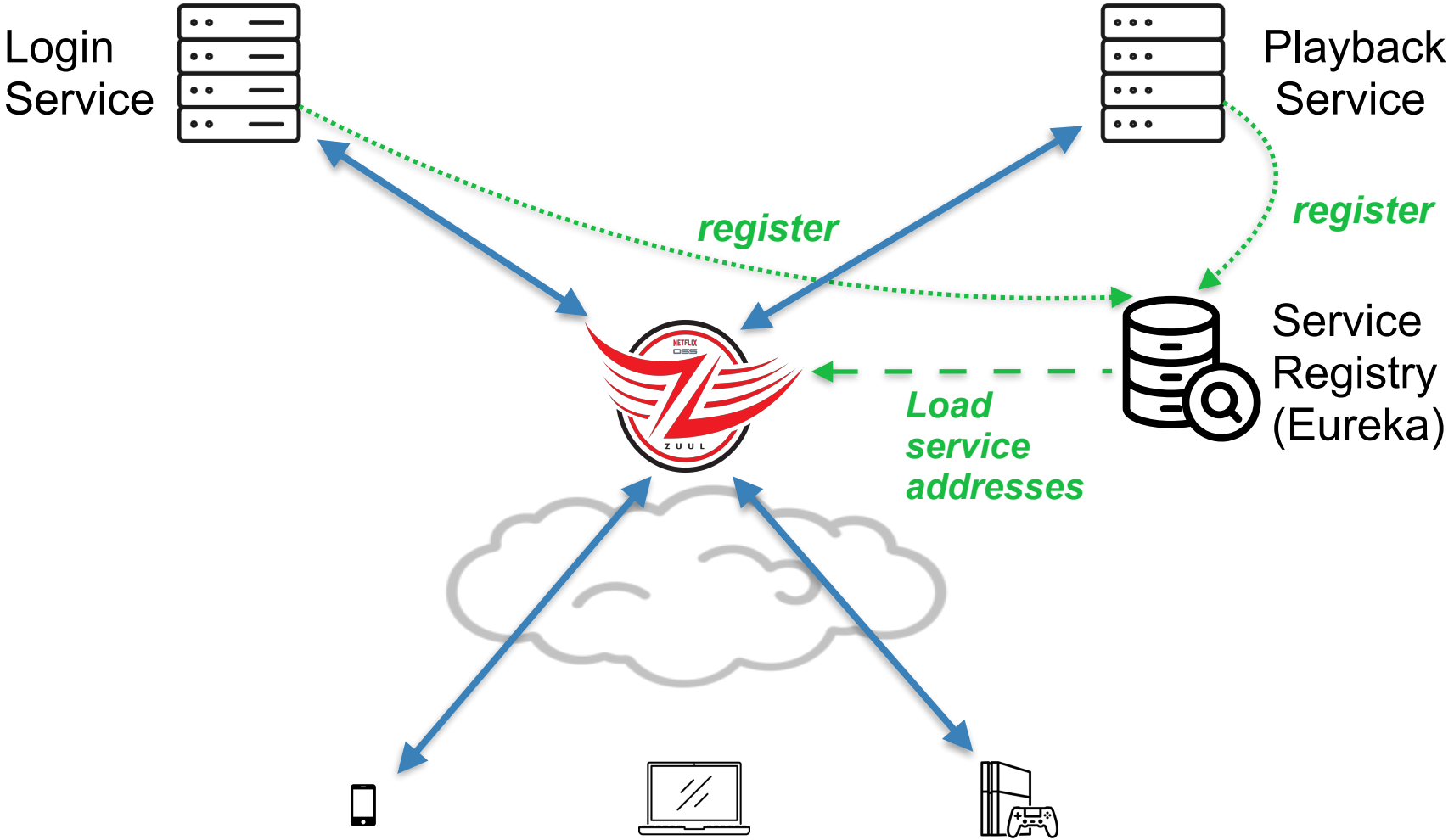


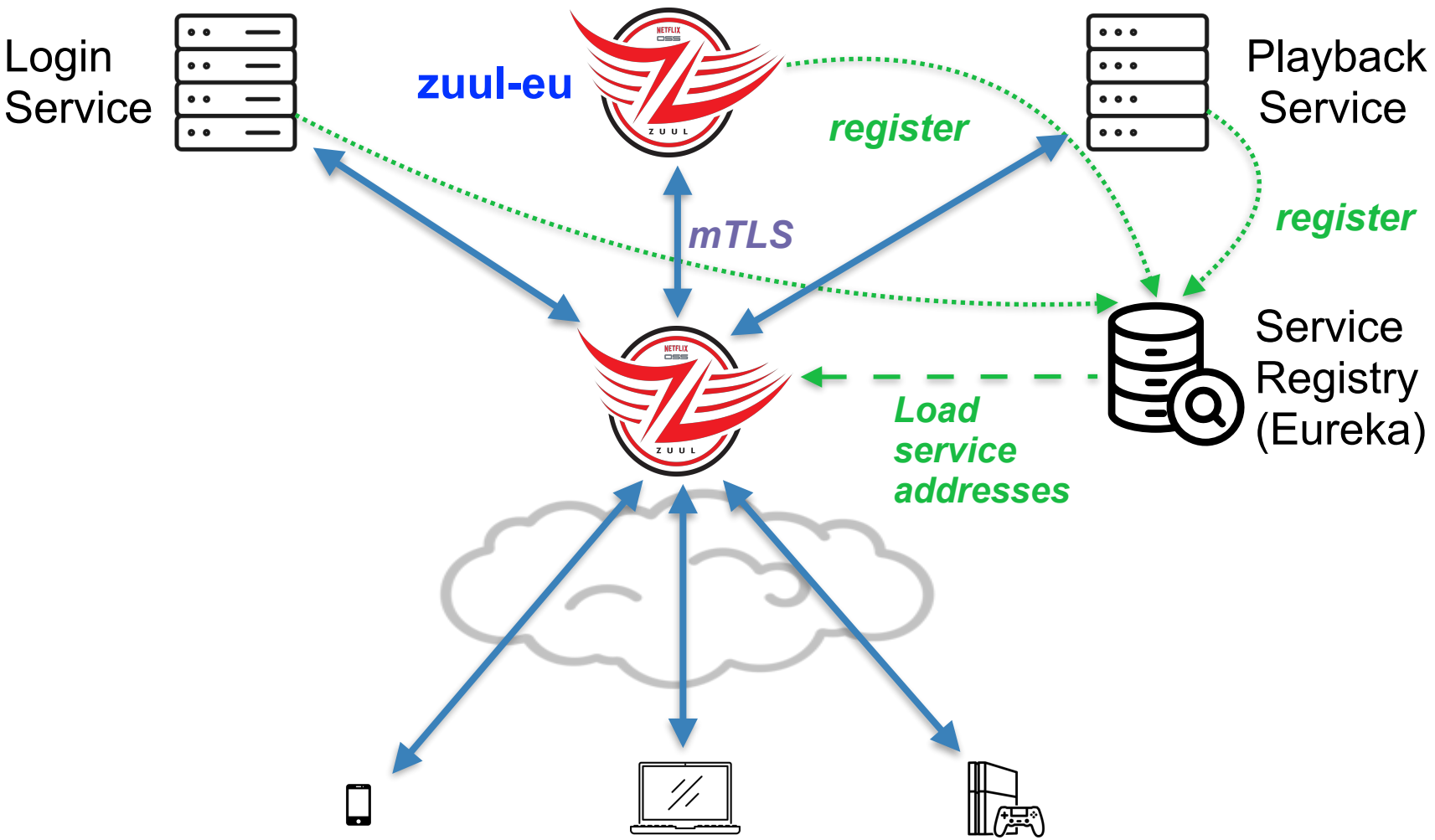

Login
Service
Playback
Service
register
register
Service
Registry
(Eureka)
NETFLIX
OSS
ZUUL


Login
Service
Playback
Service
register
register
Service
Registry
(Eureka)
Load
service
addresses
NETFLIX
OSS
ZUUL


Login
Service
Playback
Service
register
register
Service
Registry
(Eureka)
Load
service
addresses
NETFLIX
OSS
ZUUL


Login
Service
Playback
Service
register
register
register
Service
Registry
(Eureka)
Load
service
addresses
NETFLIX
OSS
ZUUL


Login
Service
Playback
Service
register
register
Service
Registry
(Eureka)
Load
service
addresses
NETFLIX
OSS
ZUUL


Login
Service
zuul-eu
register
Playback
Service
register
mTLS
Service
Registry
(Eureka)
Load
service
addresses

You can change routes dynamically at runtime

You can change routes dynamically at runtime

Key	Operator	value
path	regexp	^(/api)?/(users|logs)(/?).*
region	==	all

Routing Action - Override VIP

vipOverride

Baseline App or VIP	Target
api-prod.netflix.net:7001	api-prod-rest.netflix.net:7001

You can change routes dynamically at runtime

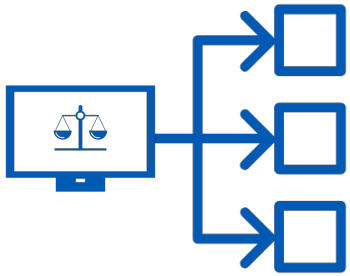
Key	Operator	value
path	regexp	^(/api)?/(users|logs)(/?).*
region	==	all

Routing Action - Override VIP

vipOverride

Baseline App or VIP	Target
api-prod.netflix.net:7001	api-prod-rest.netflix.net:7001

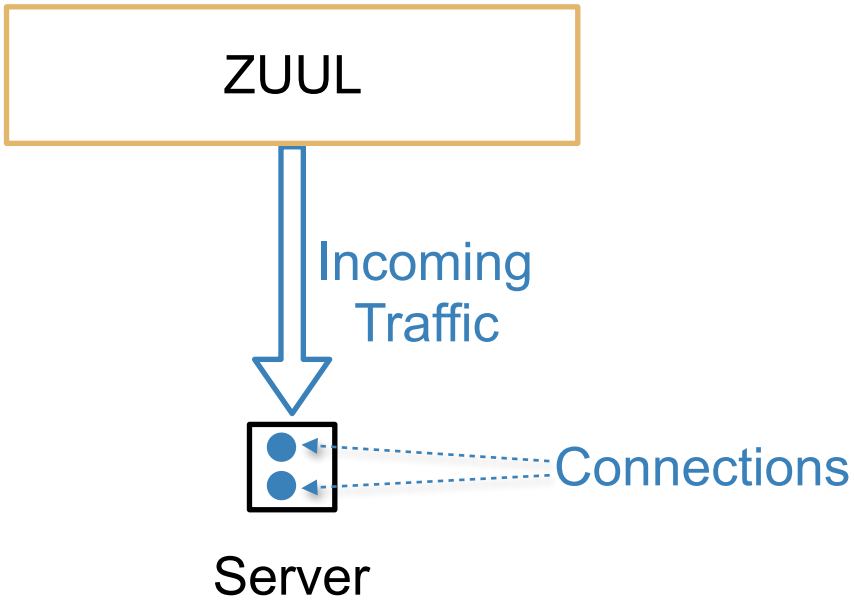
2. Load Balance



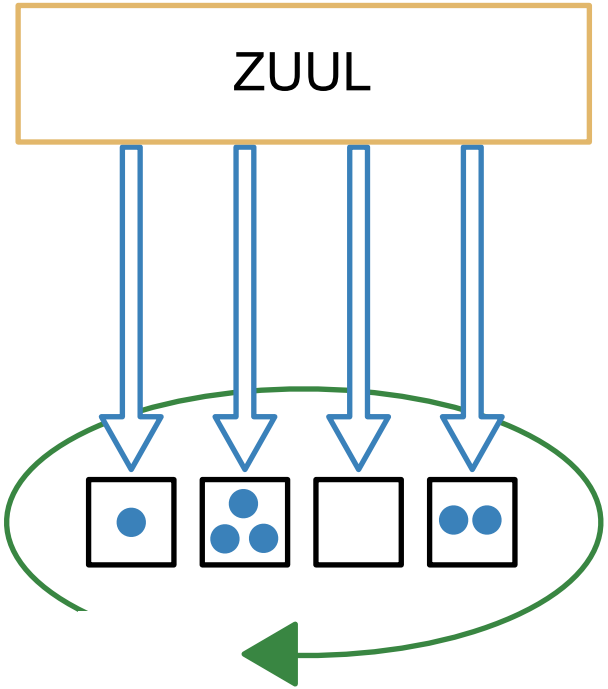
CORE
RESPONSIBILITIES

NETFLIX

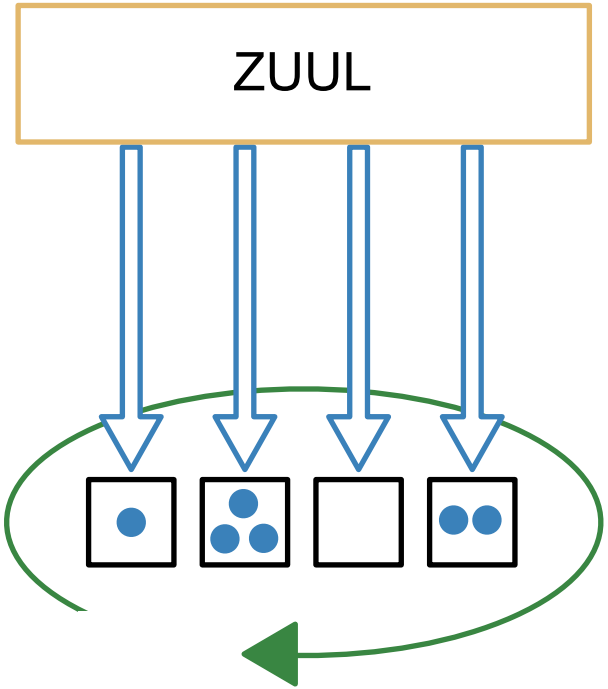
Load balancing notation



Round Robin

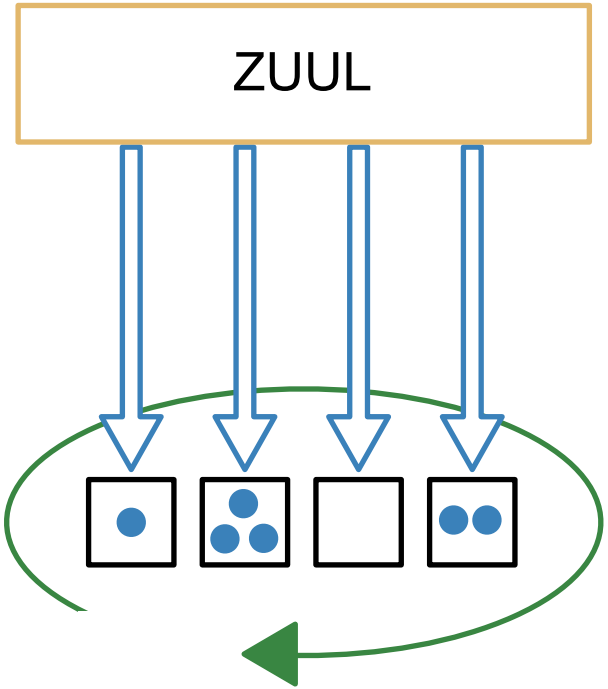


Round Robin



Easiest to implement,
No locking/ Synchronization

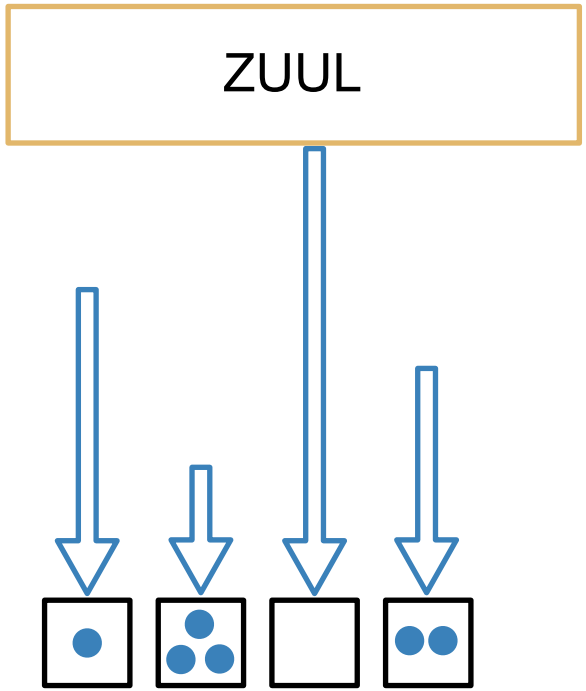
Round Robin



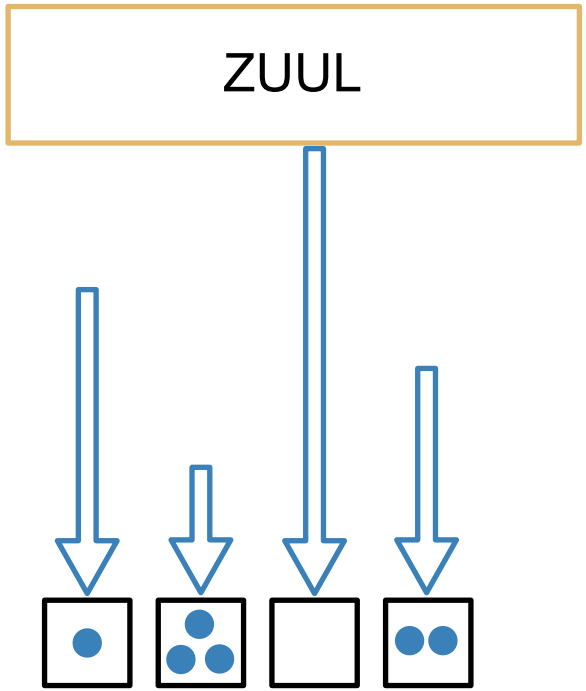
👍 Easiest to implement,
No locking/ Synchronization

👎 Unequal resource utilization

Least Connections

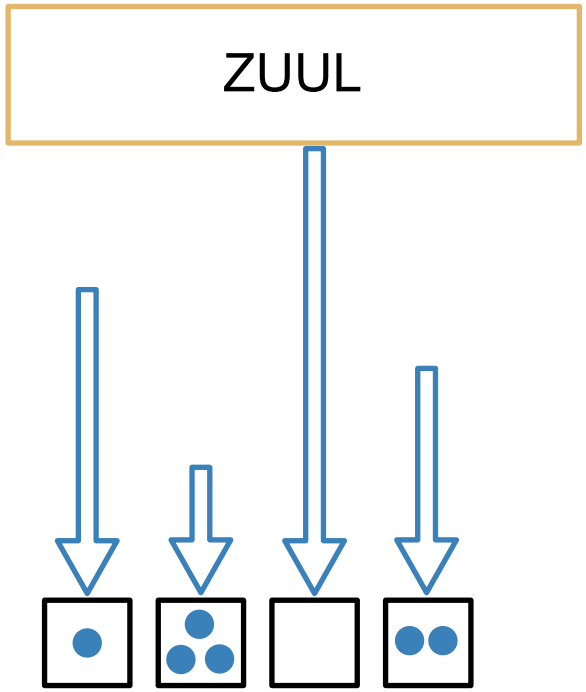


Least Connections



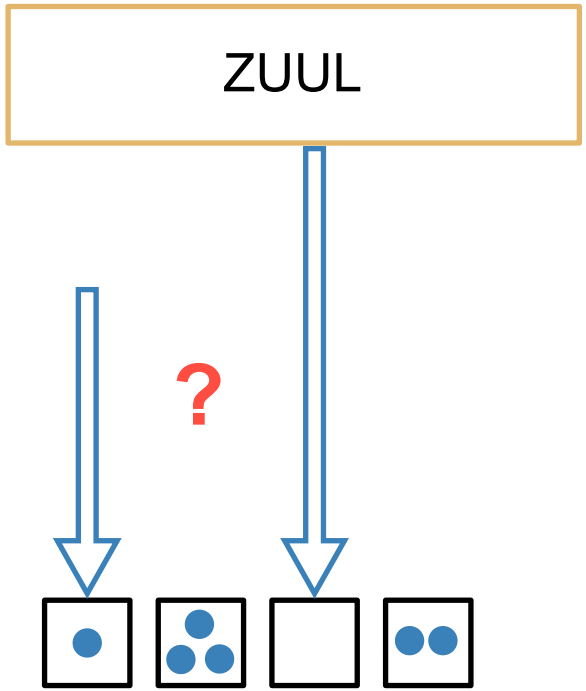
👍 More balanced resource utilization

Least Connections

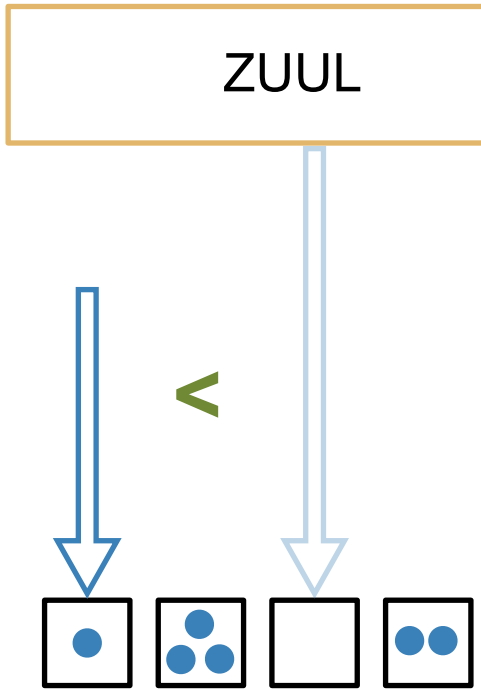


- More balanced resource utilization
- Selecting next server is expensive, O(N) operation.

Power of 2

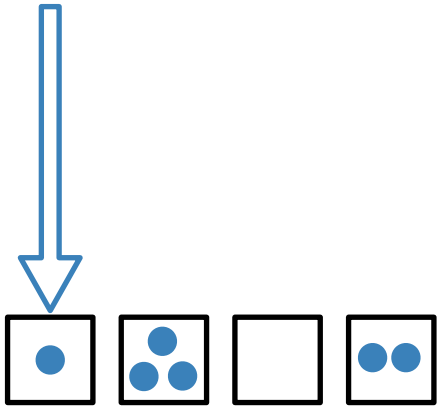


Power of 2



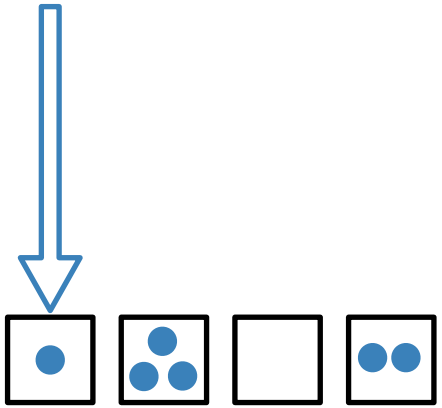
Power of 2

ZUUL



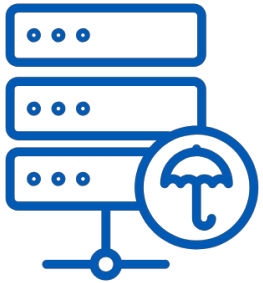
Power of 2

ZUUL



- More balanced resource utilization
- Inexpensive next server selection

3. Protect Origins



CORE
RESPONSIBILITIES

NETFLIX

Protection from slow clients - timeouts

Default = 100 ms

Protection from slow clients - timeouts

Default = 100 ms

Origin *Billing* = 150 ms

Protection from slow clients - timeouts

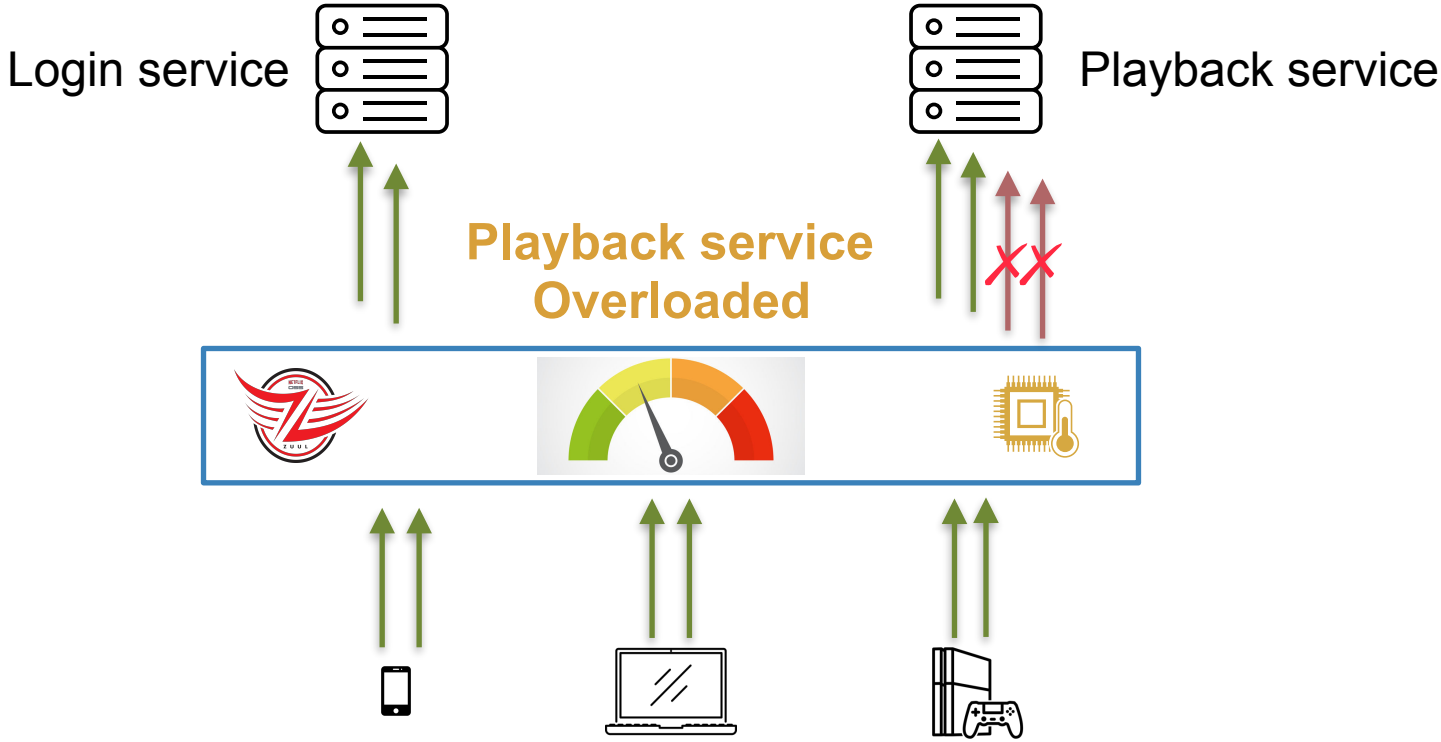
Default = 100 ms

- Origin *Billing* = 150 ms
  - Path */payment* = 200 ms

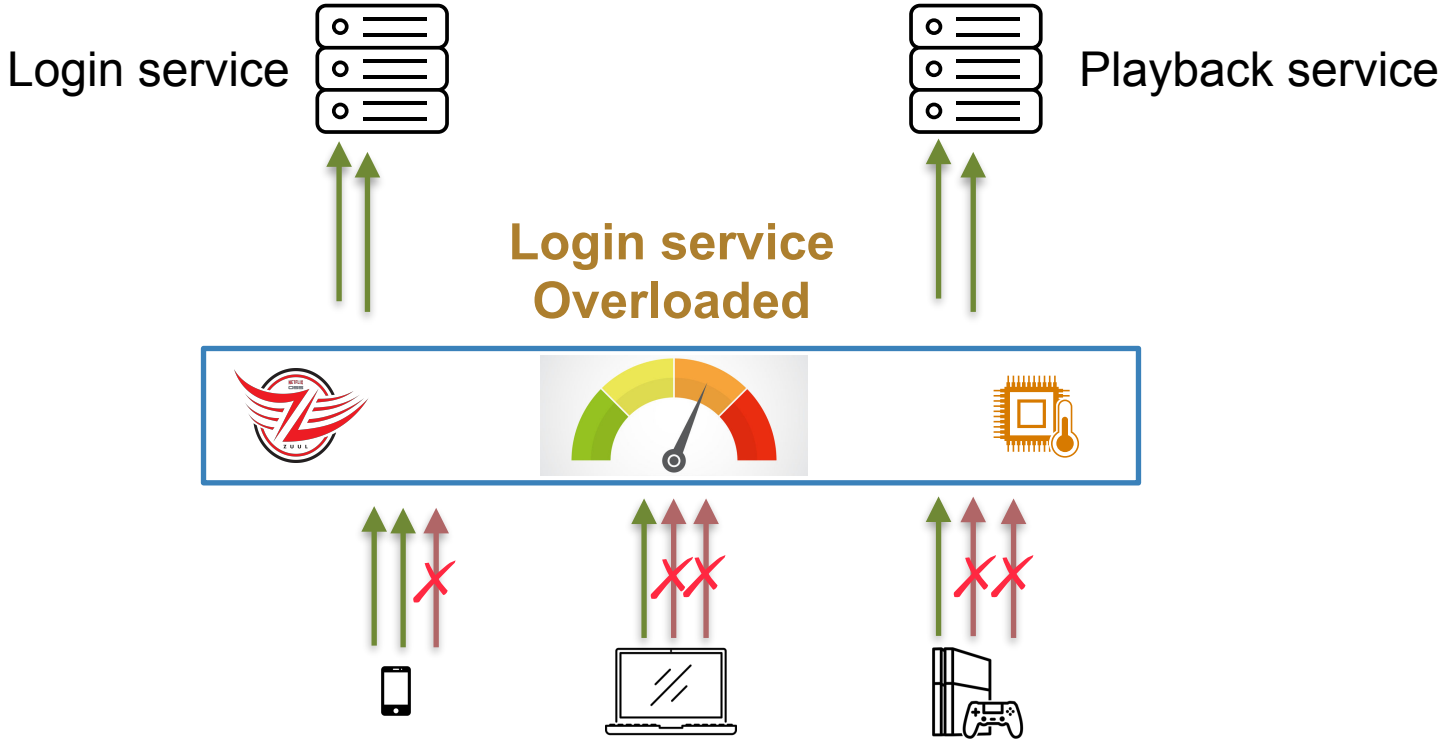
Surviving sudden surge in traffic - throttling and load shedding

Surviving sudden surge in traffic - throttling and load shedding

Per origin concurrency throttling



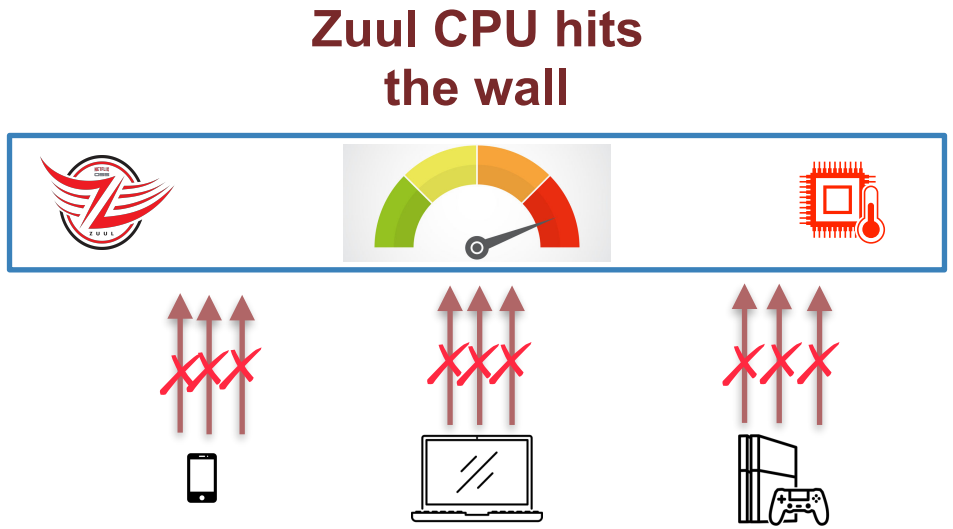
Global concurrency throttling at Zuul frontend



Zuul CPU load shedding

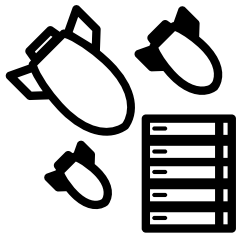
Login service

Playback service



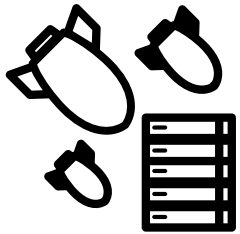
Other side of protection - security

Other side of protection - security



DDoS protection

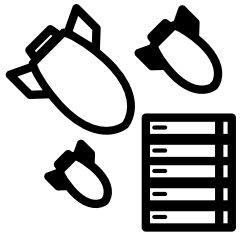
Other side of protection - security



DDoS protection

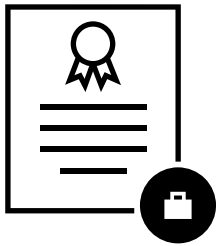
Single sign-on

Other side of protection - security



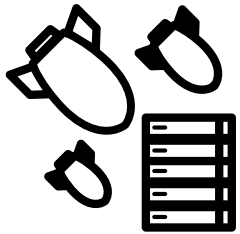
DDoS protection

Single sign-on



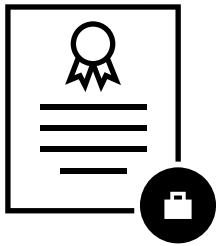
Client auth certificates

Other side of protection - security

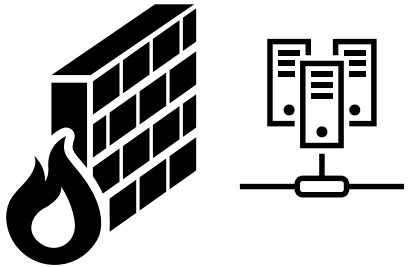


DDoS protection

Single sign-on

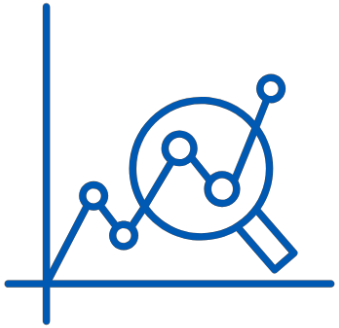


Client auth certificates



Whitelisted internal CORP access

4. Metrics and Monitoring



CORE
RESPONSIBILITIES

NETFLIX

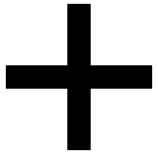
Historic and near real time aggregate metrics - Atlas

Metrics

Total request counts

Latency

Geo location



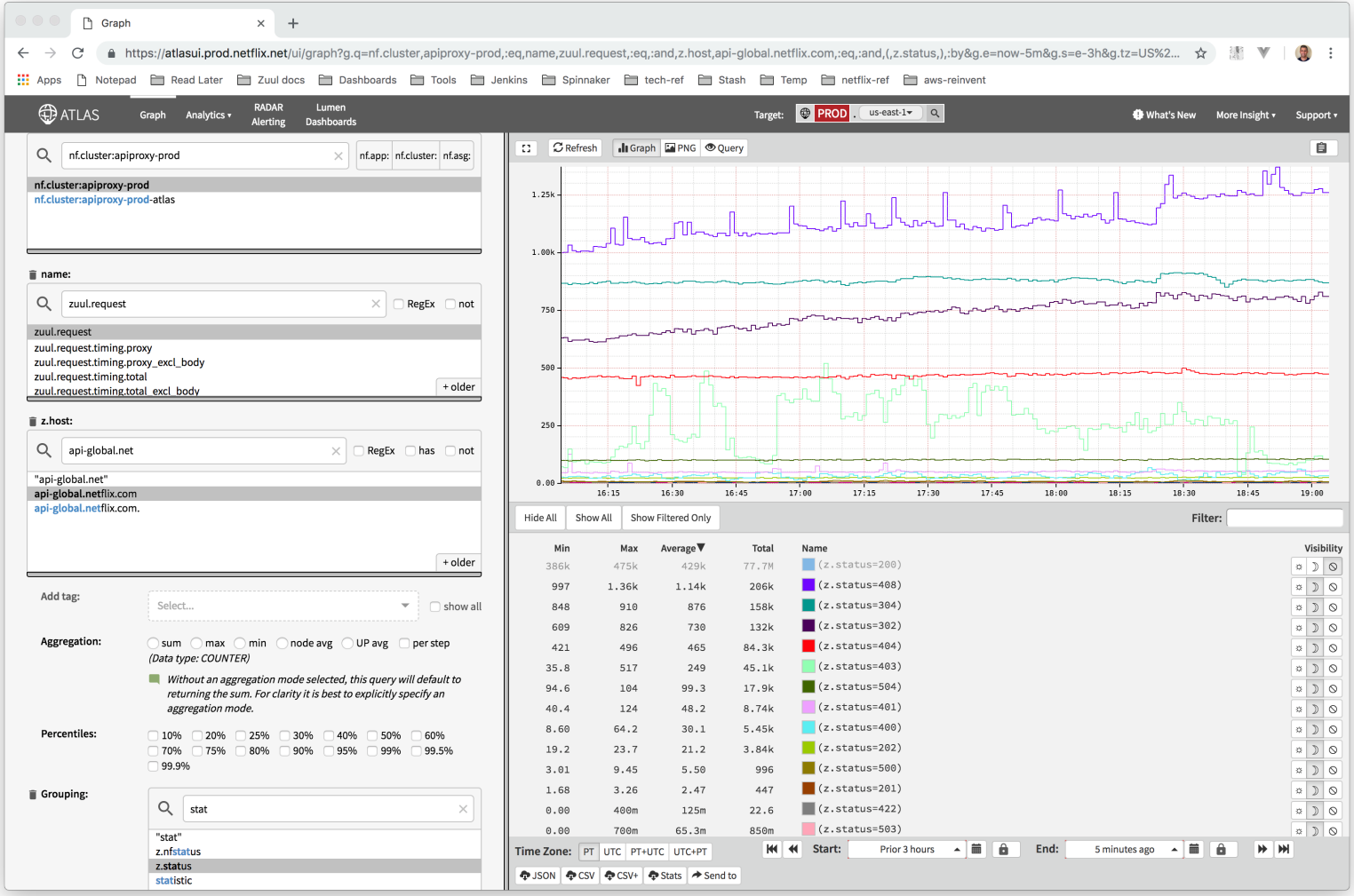
Atlas

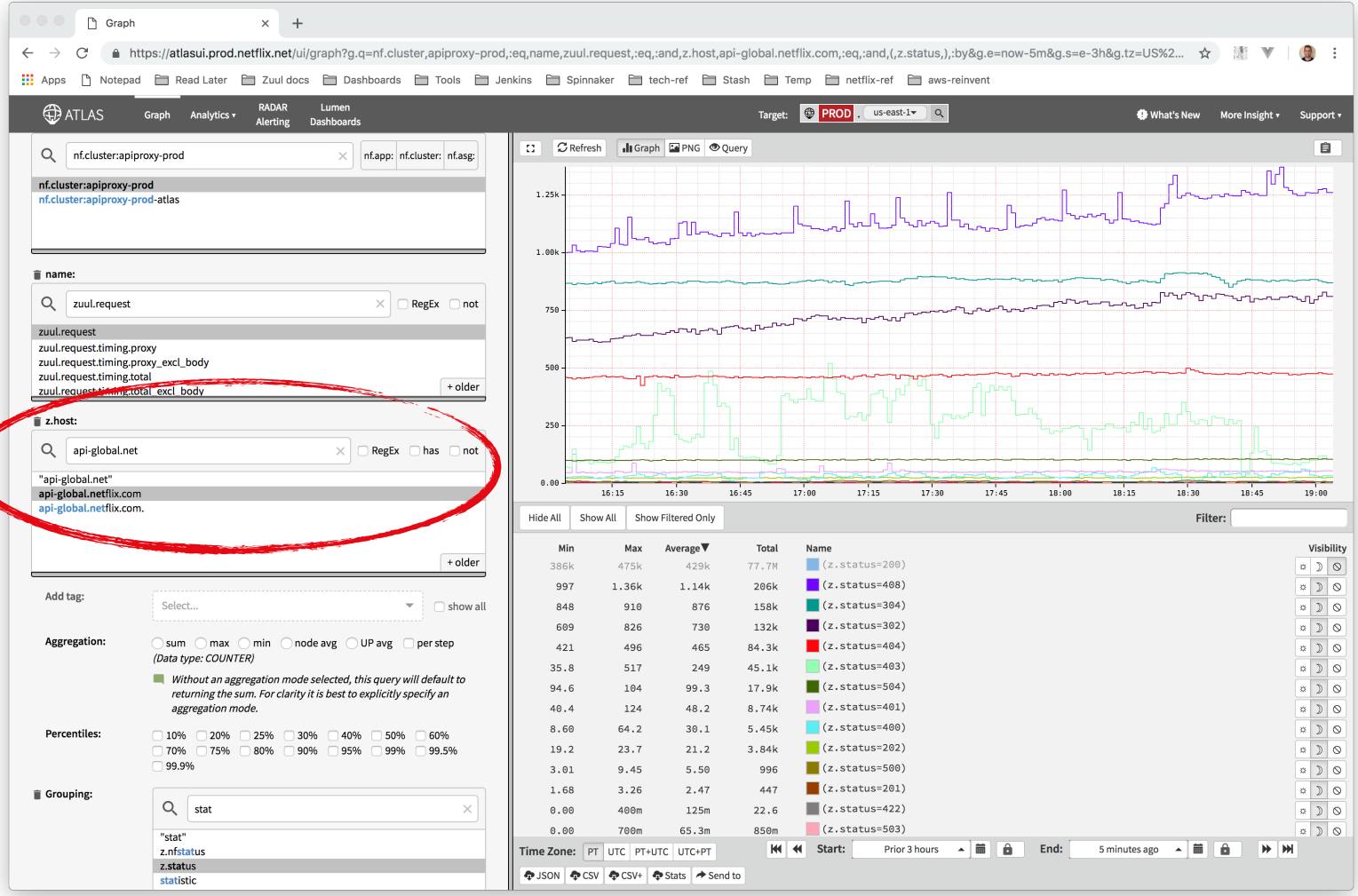
Tags

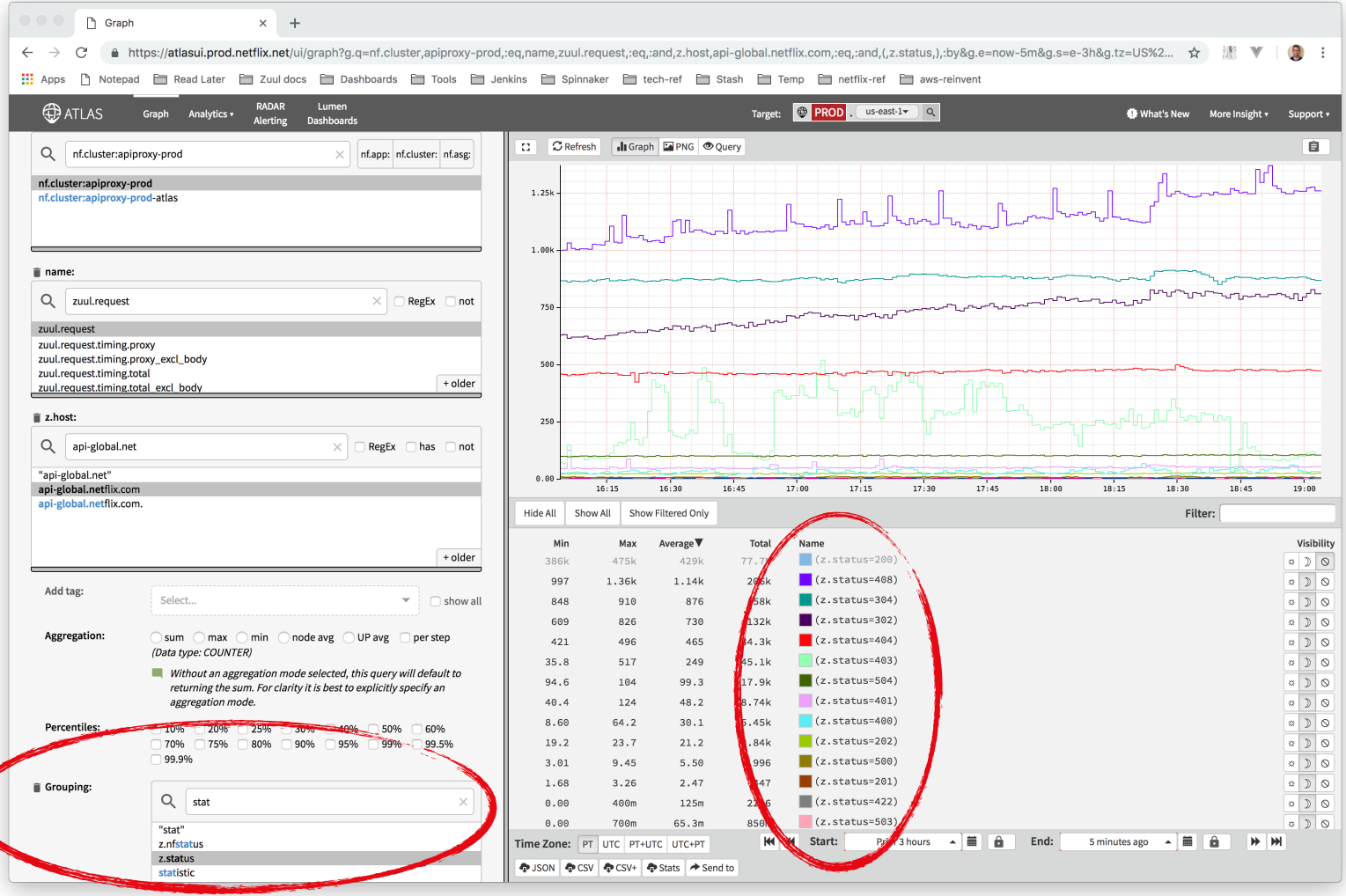
Status

Host name

Device type

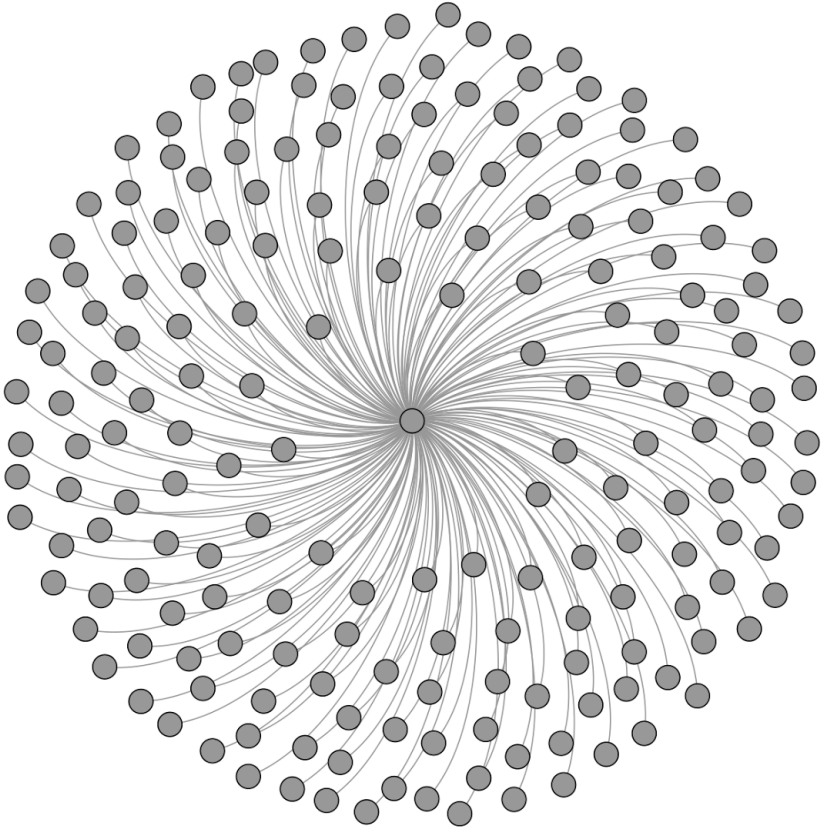

Graph
https://atlasui.prod.netflix.net/ui/graph?g.q=nf.cluster,apiproxy-prod,:eq,name,zuul.request,:eq,:and,z.host,api-global.netflix.com,:eq,:and,(,z.status,),:by&g.e=now-5m&g.s=e-3h&g.tz=US%2...
Apps
Notepad
Read Later
Zuul docs
Dashboards
Tools
Jenkins
Spinnaker
tech-ref
Stash
Temp
netflix-ref
aws-reinvent
ATLAS
Graph
Analytics
RADAR Alerting
Lumen Dashboards
Target:
PROD
us-east-1
What's New
More Insight
Support
nf.cluster:apiproxy-prod
nf.app:
nf.cluster:
nf.asg:
nf.cluster:apiproxy-prod
nf.cluster:apiproxy-prod-atlas
name:
zuul.request
RegEx
not
zuul.request
zuul.request.timing.proxy
zuul.request.timing.proxy_excl_body
zuul.request.timing.total
zuul.request.timing.total_excl_body
+ older
z.host:
api-global.net
RegEx
has
not
"api-global.net"
api-global.netflix.com
api-global.netflix.com.
+ older
Add tag:
Select...
show all
Aggregation:
sum
max
min
node avg
UP avg
per step
(Data type: COUNTER)
Without an aggregation mode selected, this query will default to returning the sum. For clarity it is best to explicitly specify an aggregation mode.
Percentiles:
10%
20%
25%
30%
40%
50%
60%
70%
75%
80%
90%
95%
99%
99.5%
99.9%
Grouping:
stat
"stat"
z.nfstatus
z.status
statistic
Refresh
Graph
PNG
Query
1.25k
1.00k
750
500
250
0.00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
Hide All
Show All
Show Filtered Only
Filter:
Min
Max
Average
Total
Name
Visibility
386k 475k 429k 77.7M (z.status=200)
997 1.36k 1.14k 206k (z.status=408)
848 910 876 158k (z.status=304)
609 826 730 132k (z.status=302)
421 496 465 84.3k (z.status=404)
35.8 517 249 45.1k (z.status=403)
94.6 104 99.3 17.9k (z.status=504)
40.4 124 48.2 8.74k (z.status=401)
8.60 64.2 30.1 5.45k (z.status=400)
19.2 23.7 21.2 3.84k (z.status=202)
3.01 9.45 5.50 996 (z.status=500)
1.68 3.26 2.47 447 (z.status=201)
0.00 400m 125m 22.6 (z.status=422)
0.00 700m 65.3m 850m (z.status=503)
Time Zone:
PT
UTC
PT+UTC
UTC+PT
Start:
Prior 3 hours
End:
5 minutes ago
JSON
CSV
CSV+
Stats
Send to


Graph
https://atlasui.prod.netflix.net/ui/graph?g.q=nf.cluster,apiproxy-prod,:eq,name,zuul.request,:eq,:and,z.host,api-global.netflix.com,:eq,:and,(,z.status,),:by&g.e=now-5m&g.s=e-3h&g.tz=US%2...
Apps
Notepad
Read Later
Zuul docs
Dashboards
Tools
Jenkins
Spinnaker
tech-ref
Stash
Temp
netflix-ref
aws-reinvent
ATLAS
Graph
Analytics
RADAR Alerting
Lumen Dashboards
Target:
PROD
us-east-1
What's New
More Insight
Support
nf.cluster:apiproxy-prod
nf.app:
nf.cluster:
nf.asg:
nf.cluster:apiproxy-prod
nf.cluster:apiproxy-prod-atlas
name:
zuul.request
RegEx
not
zuul.request
zuul.request.timing.proxy
zuul.request.timing.proxy_excl_body
zuul.request.timing.total
zuul.request.timing.total_excl_body
+ older
z.host:
api-global.net
RegEx
has
not
"api-global.net"
api-global.netflix.com
api-global.netflix.com.
+ older
Add tag:
Select...
show all
Aggregation:
sum
max
min
node avg
UP avg
per step
(Data type: COUNTER)
Without an aggregation mode selected, this query will default to returning the sum. For clarity it is best to explicitly specify an aggregation mode.
Percentiles:
10%
20%
25%
30%
40%
50%
60%
70%
75%
80%
90%
95%
99%
99.5%
99.9%
Grouping:
stat
"stat"
z.nfstatus
z.status
statistic
Refresh
Graph
PNG
Query
1.25k
1.00k
750
500
250
0.00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
Hide All
Show All
Show Filtered Only
Filter:
Min
Max
Average
Total
Name
Visibility
386k 475k 429k 77.7M (z.status=200)
997 1.36k 1.14k 206k (z.status=408)
848 910 876 158k (z.status=304)
609 826 730 132k (z.status=302)
421 496 465 84.3k (z.status=404)
35.8 517 249 45.1k (z.status=403)
94.6 104 99.3 17.9k (z.status=504)
40.4 124 48.2 8.74k (z.status=401)
8.60 64.2 30.1 5.45k (z.status=400)
19.2 23.7 21.2 3.84k (z.status=202)
3.01 9.45 5.50 996 (z.status=500)
1.68 3.26 2.47 447 (z.status=201)
0.00 400m 125m 22.6 (z.status=422)
0.00 700m 65.3m 850m (z.status=503)
Time Zone:
PT
UTC
PT+UTC
UTC+PT
Start:
Prior 3 hours
End:
5 minutes ago
JSON
CSV
CSV+
Stats
Send to


Graph
https://atlasui.prod.netflix.net/ui/graph?g.q=nf.cluster,apiproxy-prod,:eq,name,zuul.request,:eq,:and,z.host,api-global.netflix.com,:eq,:and,(,z.status,),:by&g.e=now-5m&g.s=e-3h&g.tz=US%2...
Apps
Notepad
Read Later
Zuul docs
Dashboards
Tools
Jenkins
Spinnaker
tech-ref
Stash
Temp
netflix-ref
aws-reinvent
ATLAS
Graph
Analytics
RADAR Alerting
Lumen Dashboards
Target:
PROD
us-east-1
What's New
More Insight
Support
nf.cluster:apiproxy-prod
nf.app:
nf.cluster:
nf.asg:
nf.cluster:apiproxy-prod
nf.cluster:apiproxy-prod-atlas
name:
zuul.request
RegEx
not
zuul.request
zuul.request.timing.proxy
zuul.request.timing.proxy_excl_body
zuul.request.timing.total
zuul.request.timing.total_excl_body
+ older
z.host:
api-global.net
RegEx
has
not
"api-global.net"
api-global.netflix.com
api-global.netflix.com.
+ older
Add tag:
Select...
show all
Aggregation:
sum
max
min
node avg
UP avg
per step
(Data type: COUNTER)
Without an aggregation mode selected, this query will default to returning the sum. For clarity it is best to explicitly specify an aggregation mode.
Percentiles:
10%
20%
25%
30%
40%
50%
60%
70%
75%
80%
90%
95%
99%
99.5%
99.9%
Grouping:
stat
"stat"
z.nfstatus
z.status
statistic
Refresh
Graph
PNG
Query
1.25k
1.00k
750
500
250
0.00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
Hide All
Show All
Show Filtered Only
Filter:
Min
Max
Average
Total
Name
Visibility
386k 475k 429k (z.status=200)
997 1.36k 1.14k (z.status=408)
848 910 876 (z.status=304)
609 826 730 (z.status=302)
421 496 465 (z.status=404)
35.8 517 249 (z.status=403)
94.6 104 99.3 (z.status=504)
40.4 124 48.2 (z.status=401)
8.60 64.2 30.1 (z.status=400)
19.2 23.7 21.2 (z.status=202)
3.01 9.45 5.50 (z.status=500)
1.68 3.26 2.47 (z.status=201)
0.00 400m 125m (z.status=422)
0.00 700m 65.3m (z.status=503)
Time Zone:
PT
UTC
PT+UTC
UTC+PT
Start:
Prior 3 hours
End:
5 minutes ago
JSON
CSV
CSV+
Stats
Send to

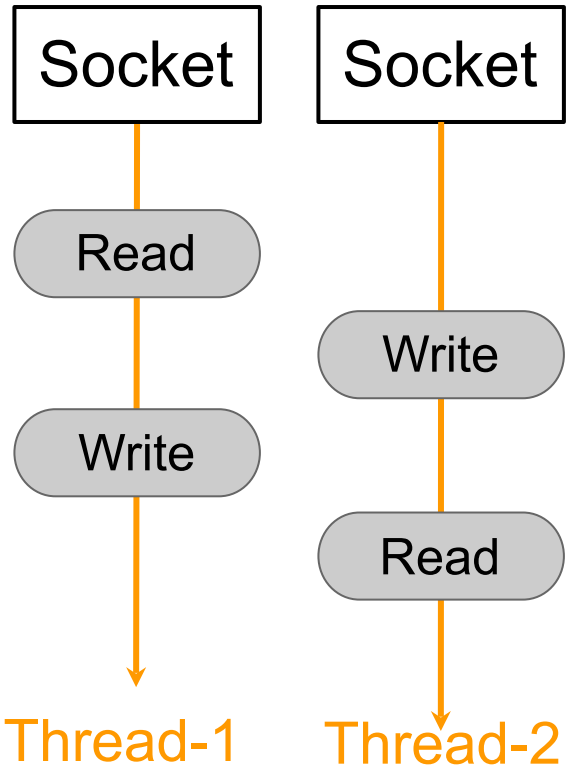
How Zuul does it?

Zuul Architecture

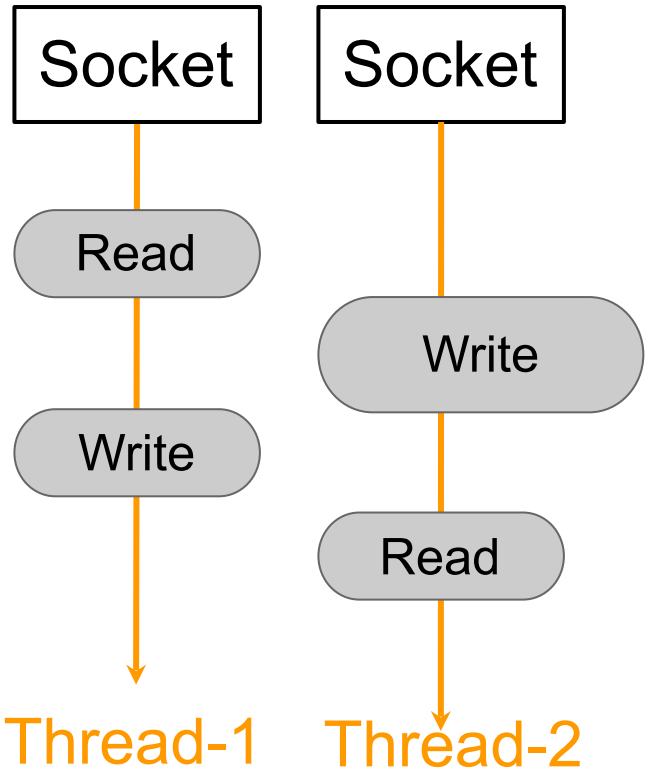


C10K challenge

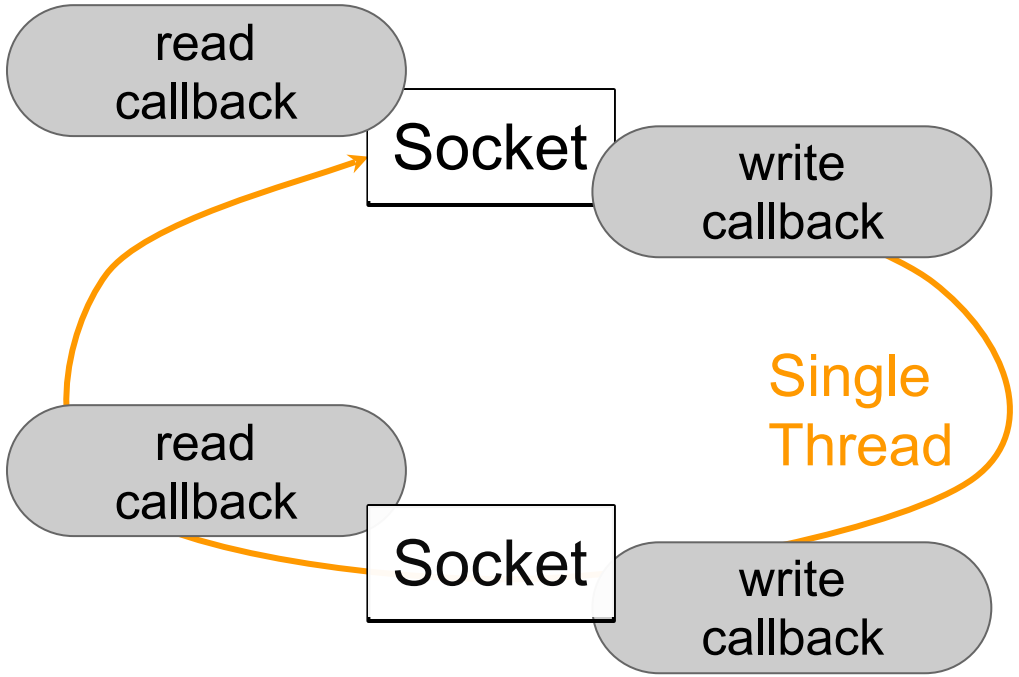
Thread per Connection



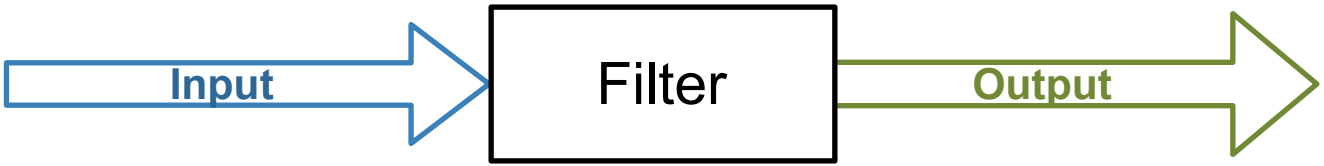
Thread per Connection



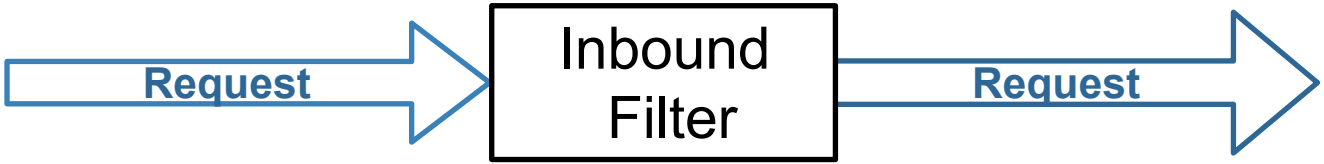
Async I/O



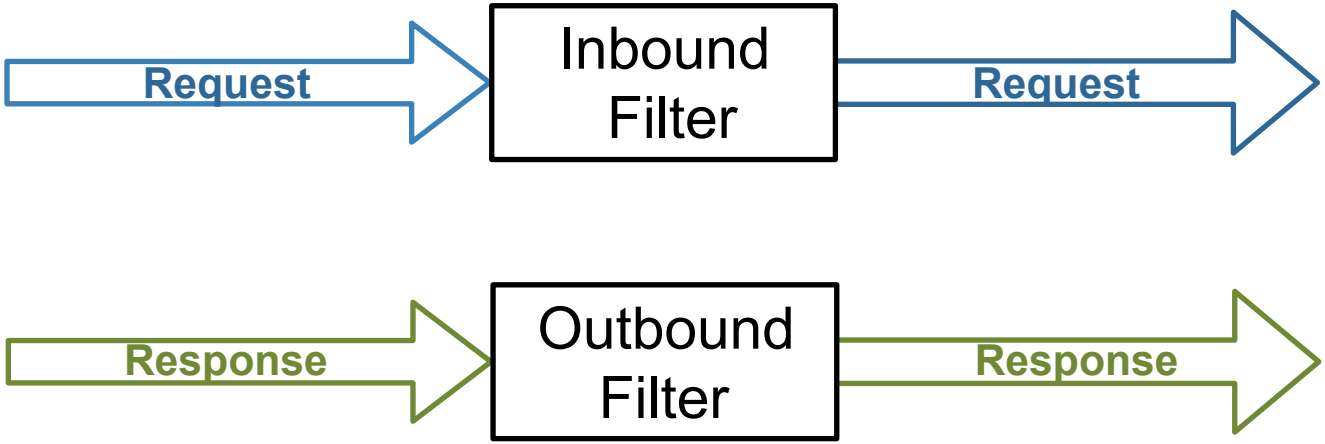
Core abstraction at the heart of the Zuul - Filter



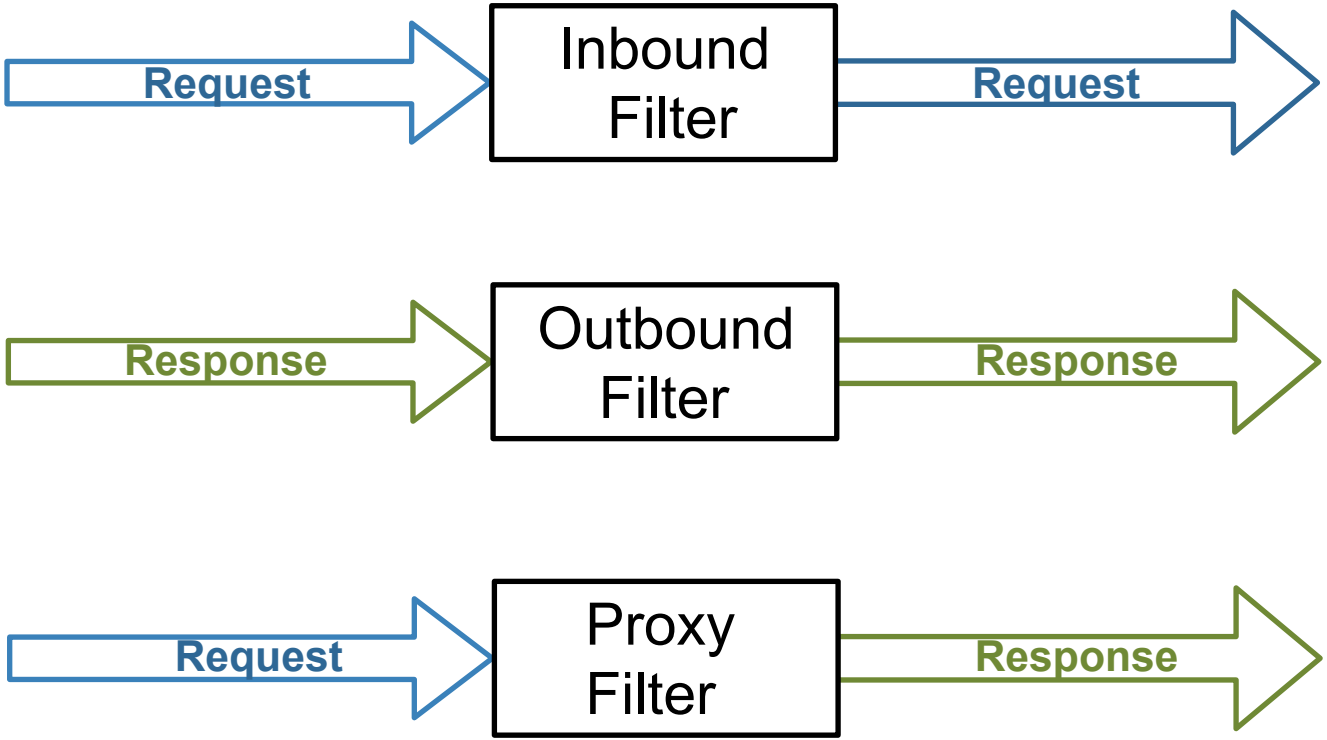
Zuul is made up of different types of filters



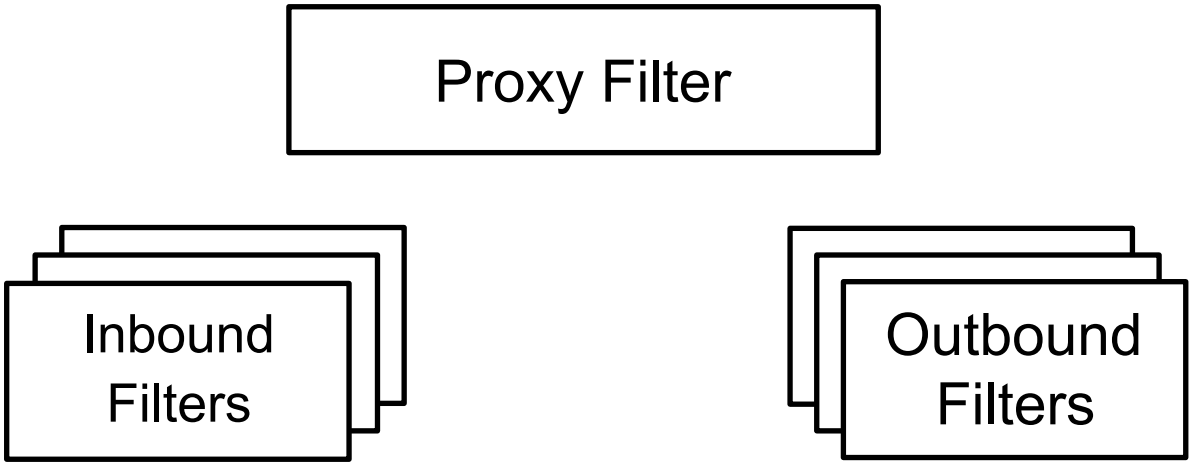
Zuul is made up of different types of filters



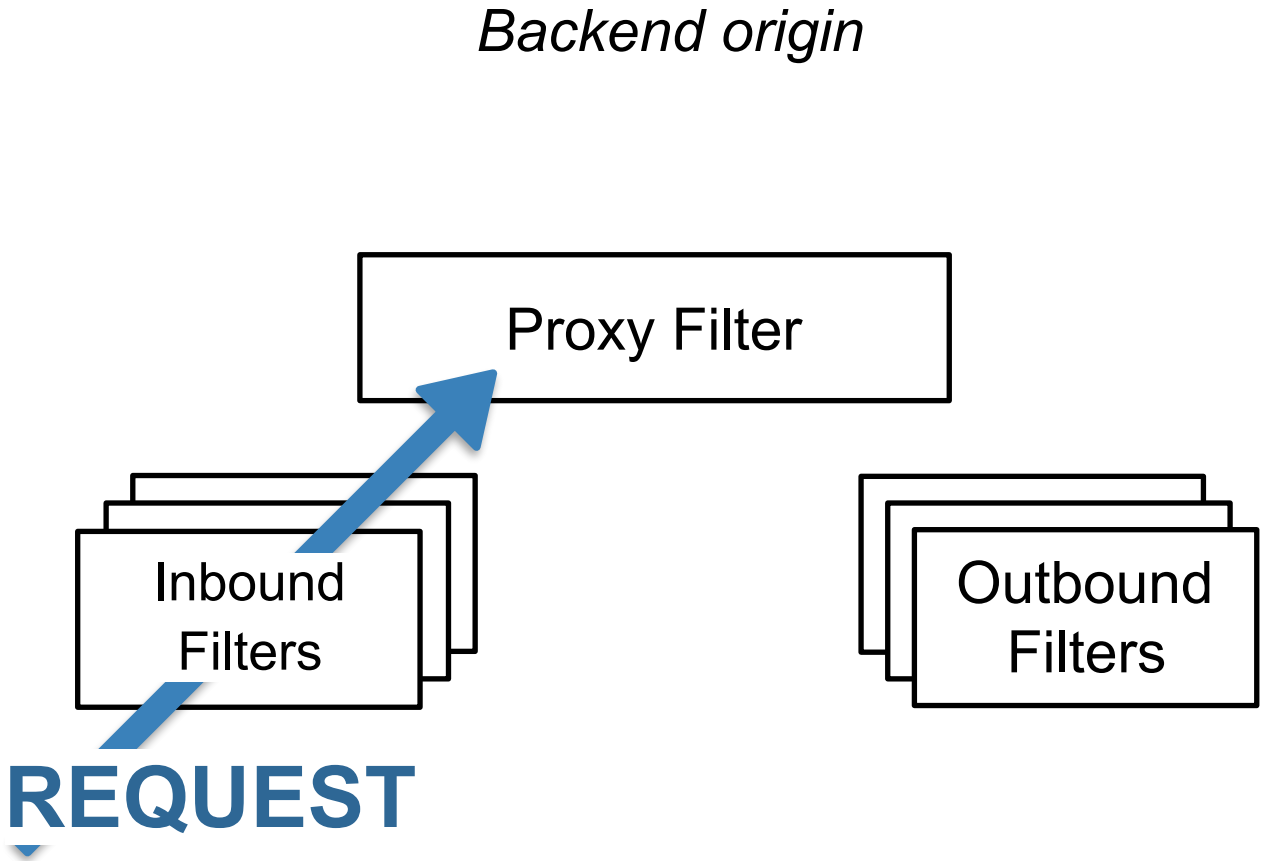
Zuul is made up of different types of filters



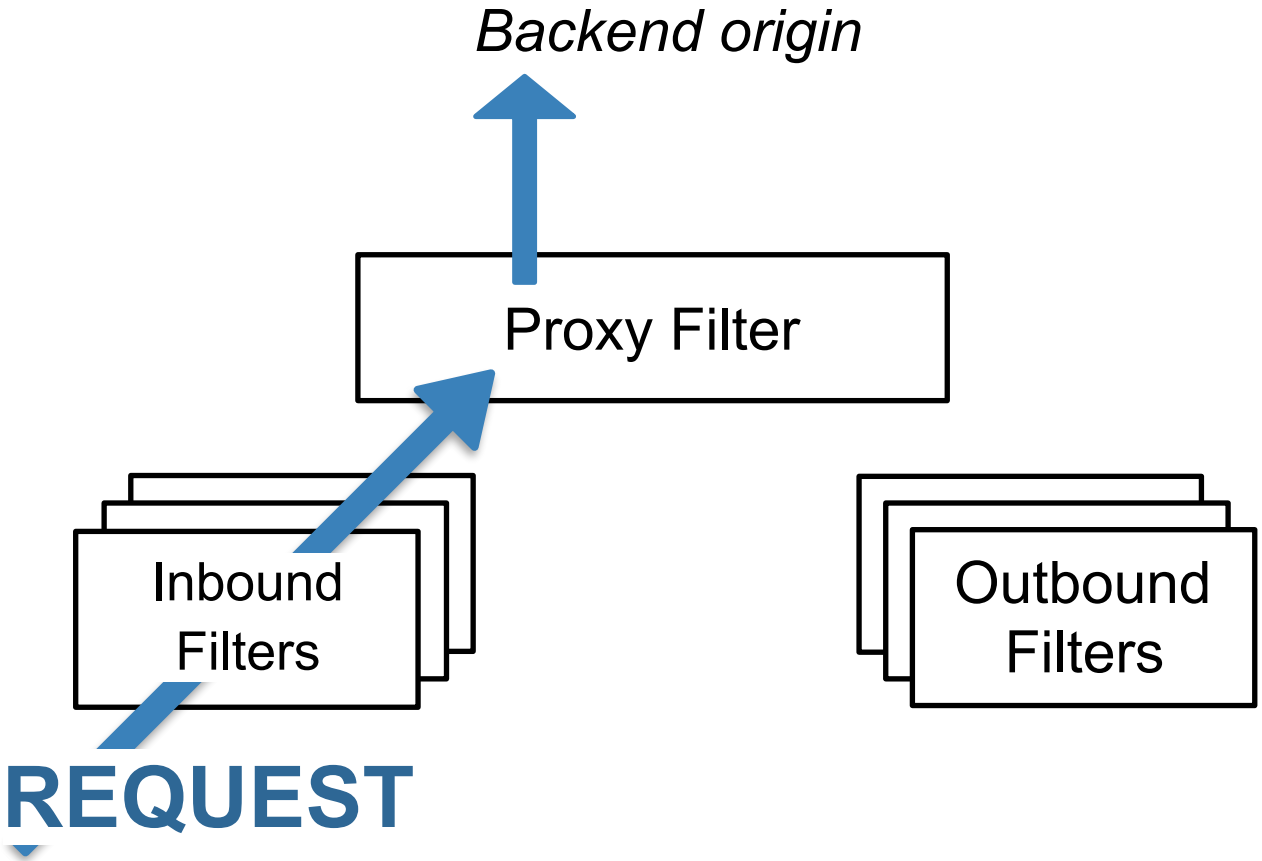
Simple, extensible Zuul core architecture



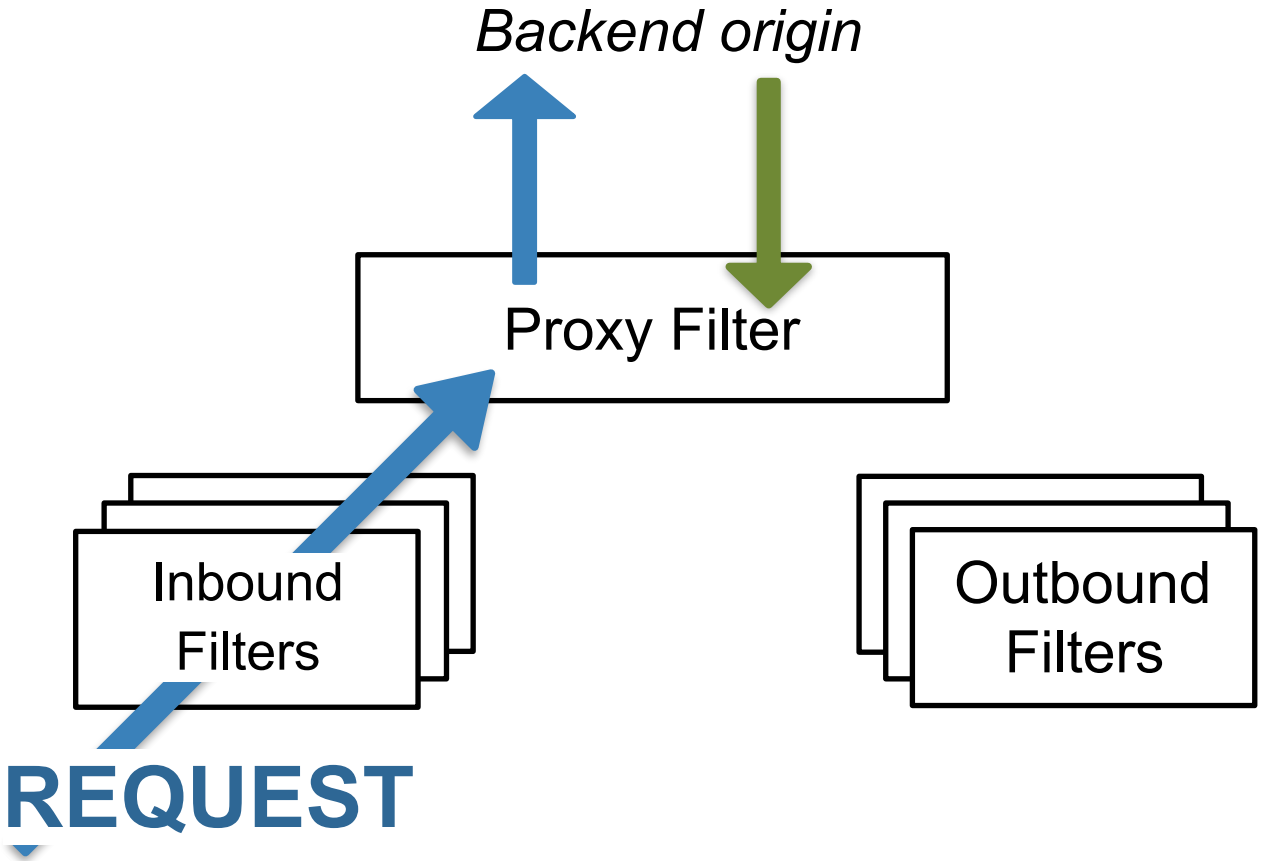
Simple, extensible Zuul core architecture



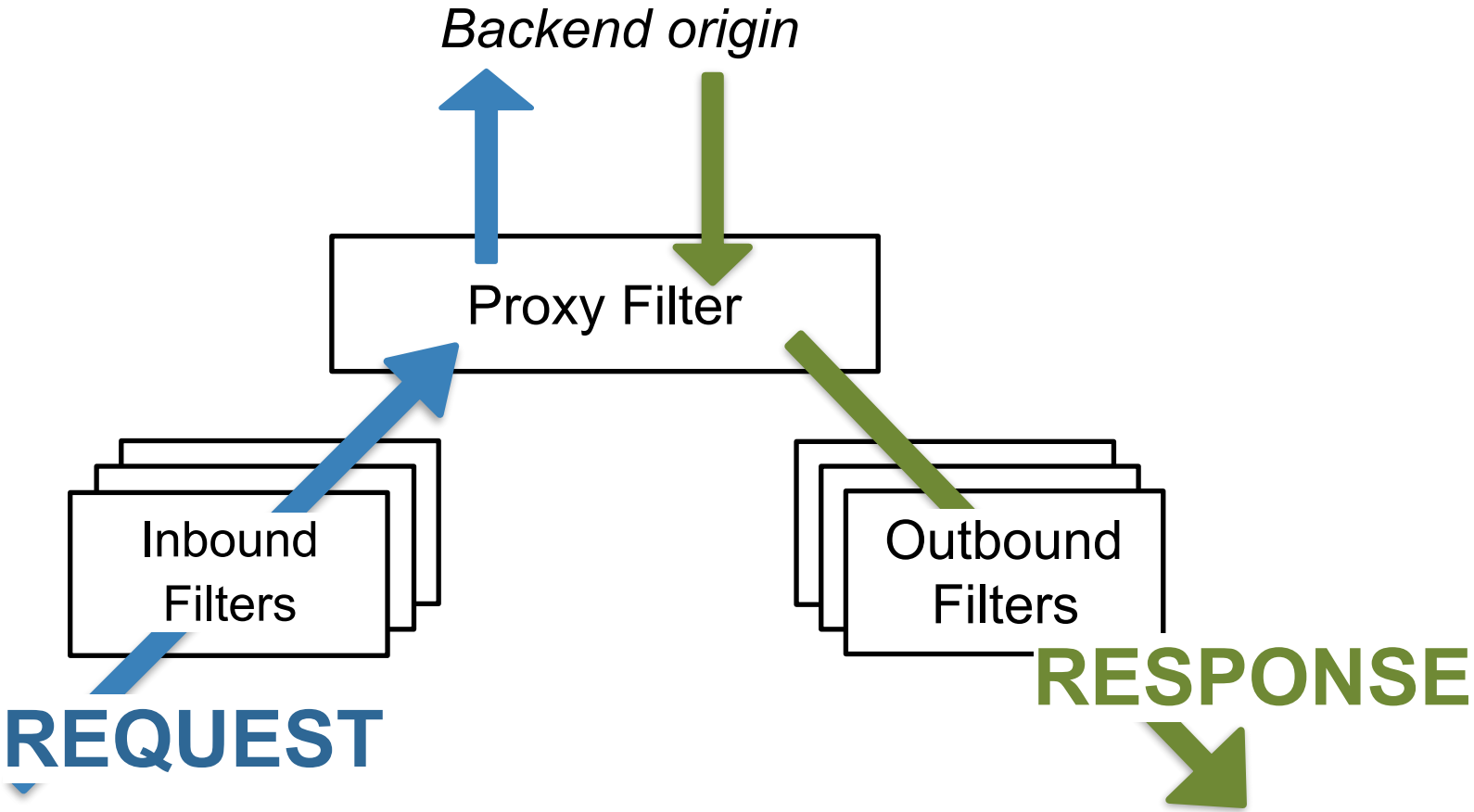
Simple, extensible Zuul core architecture



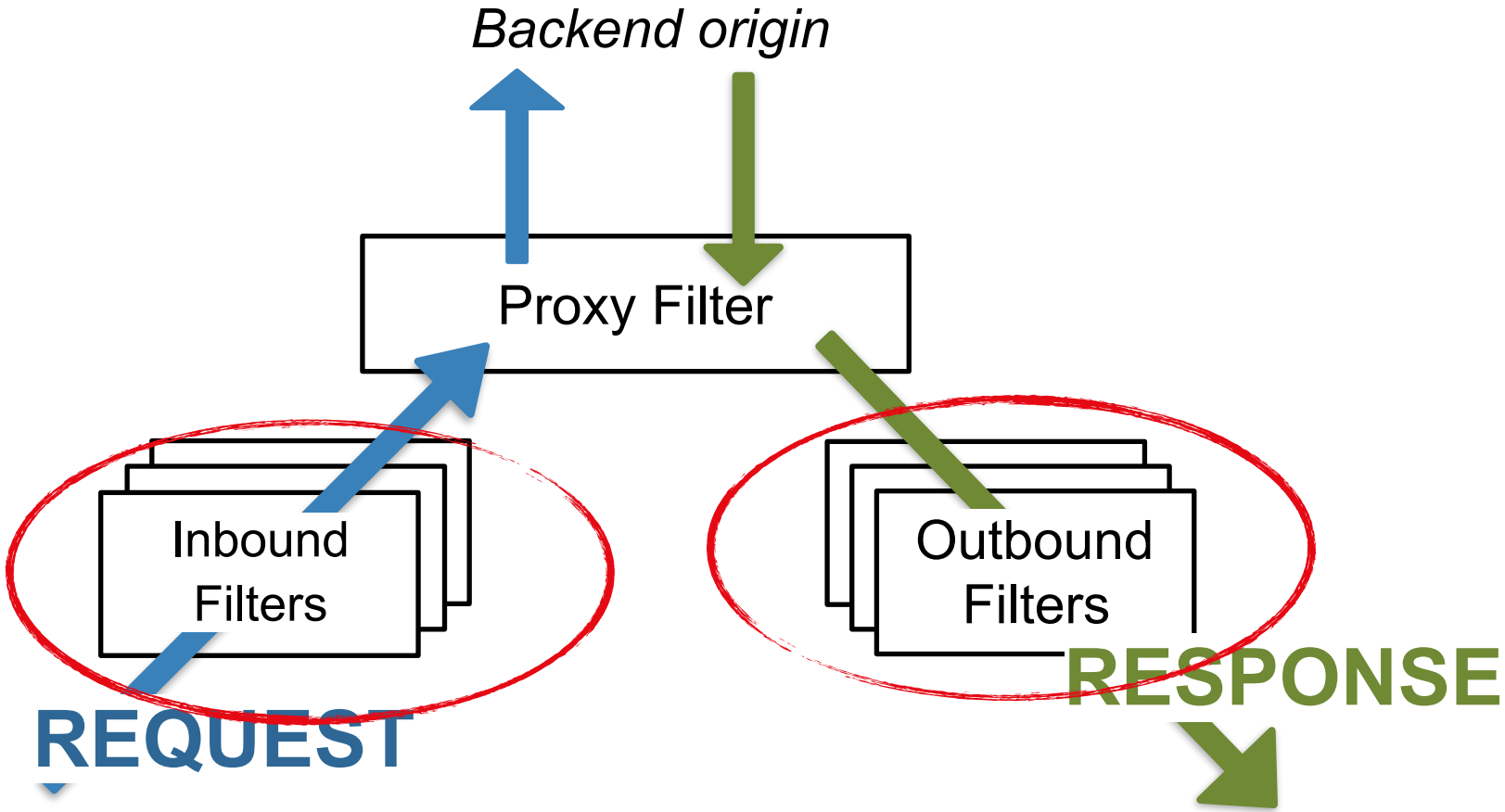
Simple, extensible Zuul core architecture



Simple, extensible Zuul core architecture



Simple, extensible Zuul core architecture



Future roadmap

Plans for future

Planned features

- End to end HTTP/2 proxying

Planned features

- End to end HTTP/2 proxying
- WebSocket proxying

Planned features

- End to end HTTP/2 proxying
- WebSocket proxying
- gRPC proxying

Planned features

- End to end HTTP/2 proxying
- WebSocket proxying
- gRPC proxying
- Edge caching

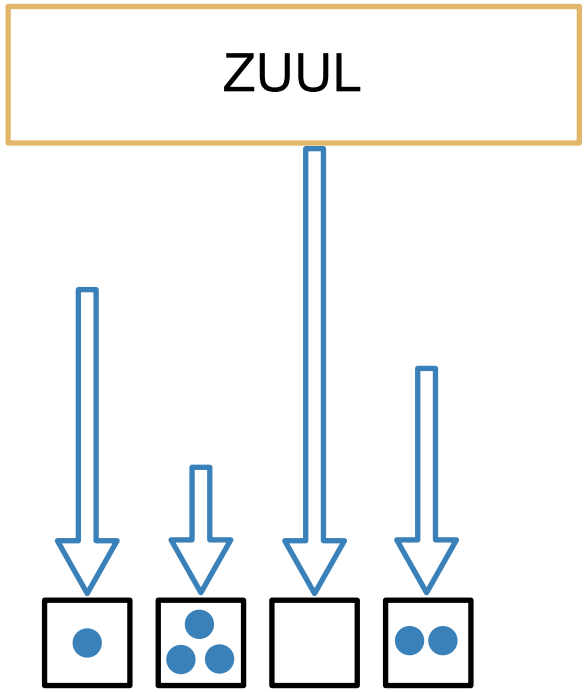
In conclusion

You can deploy Zuul at the edge of your network to gain higher **availability** and better **visibility** with unparalleled **flexibility** in handling your traffic.

Thank you.

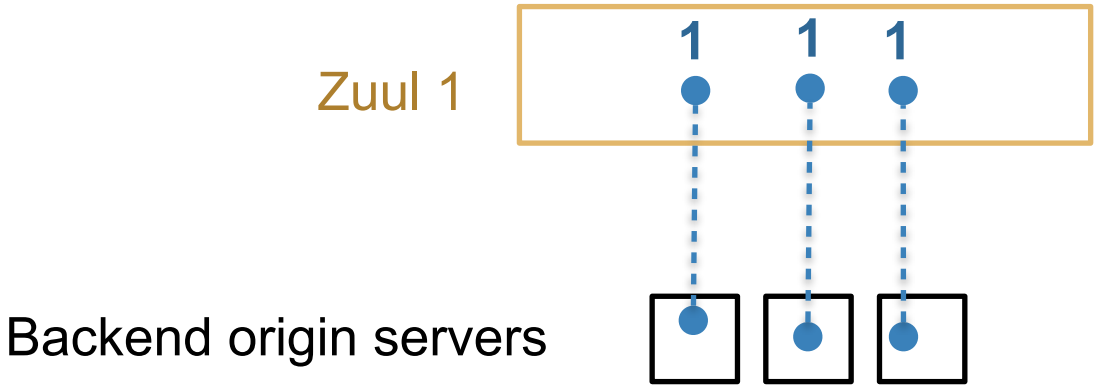
NETFLIX

Least Connections

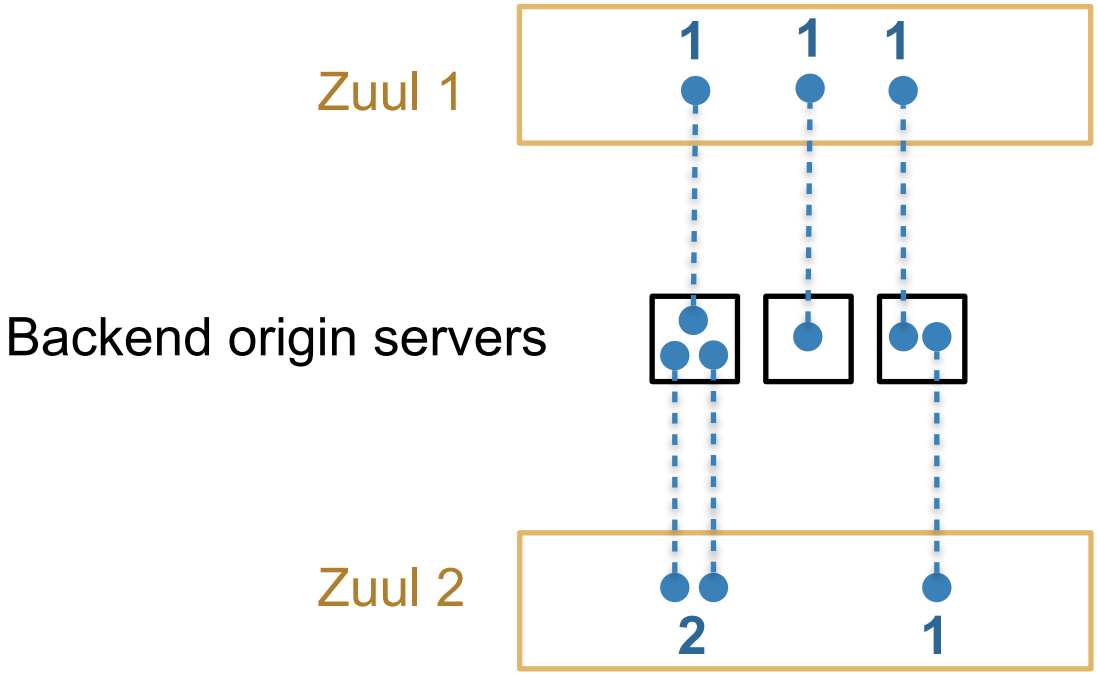


- 👍 More balanced resource utilization
- 👎 Tricky to track number of connections when multiple load balancing nodes are involved

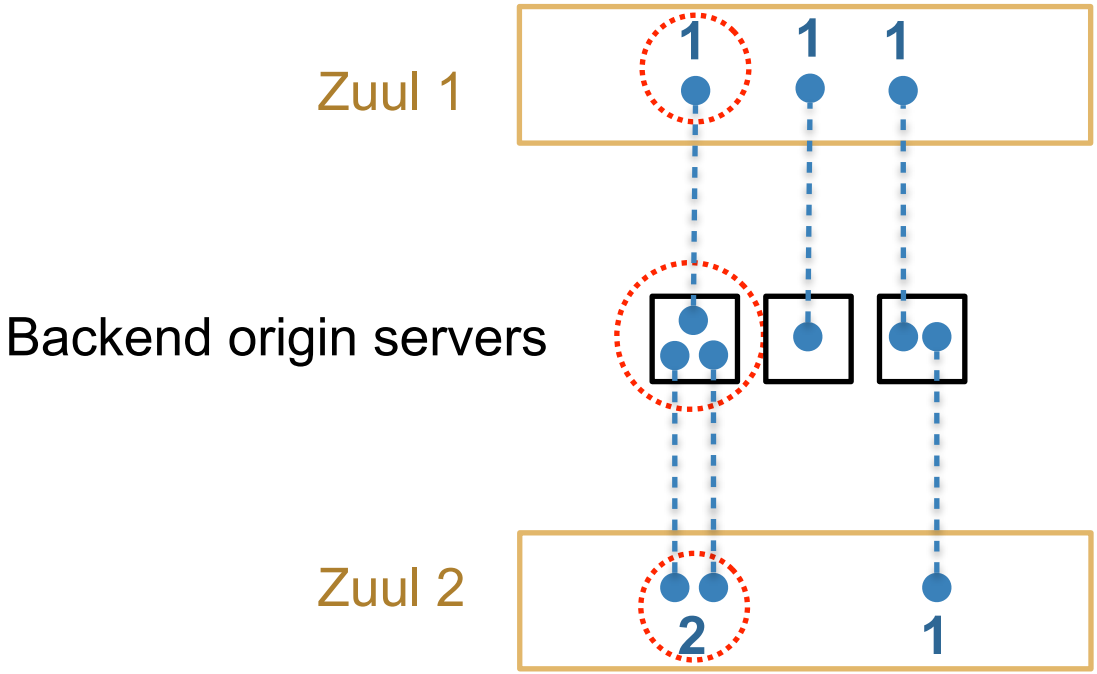
Determining number of connections with more than one LB



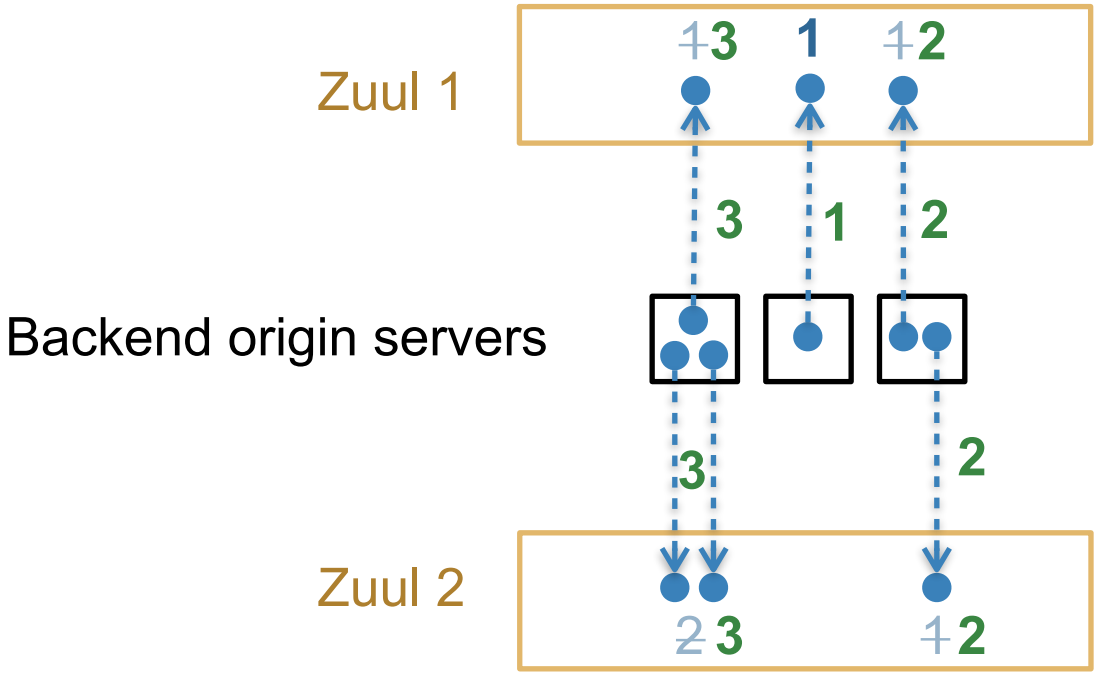
Determining number of connections with more than one LB



Determining number of connections with more than one LB



Zuul's improvement to least connections algorithm



Zuul's improvement