Observability and control in the age of the service mesh: present and future

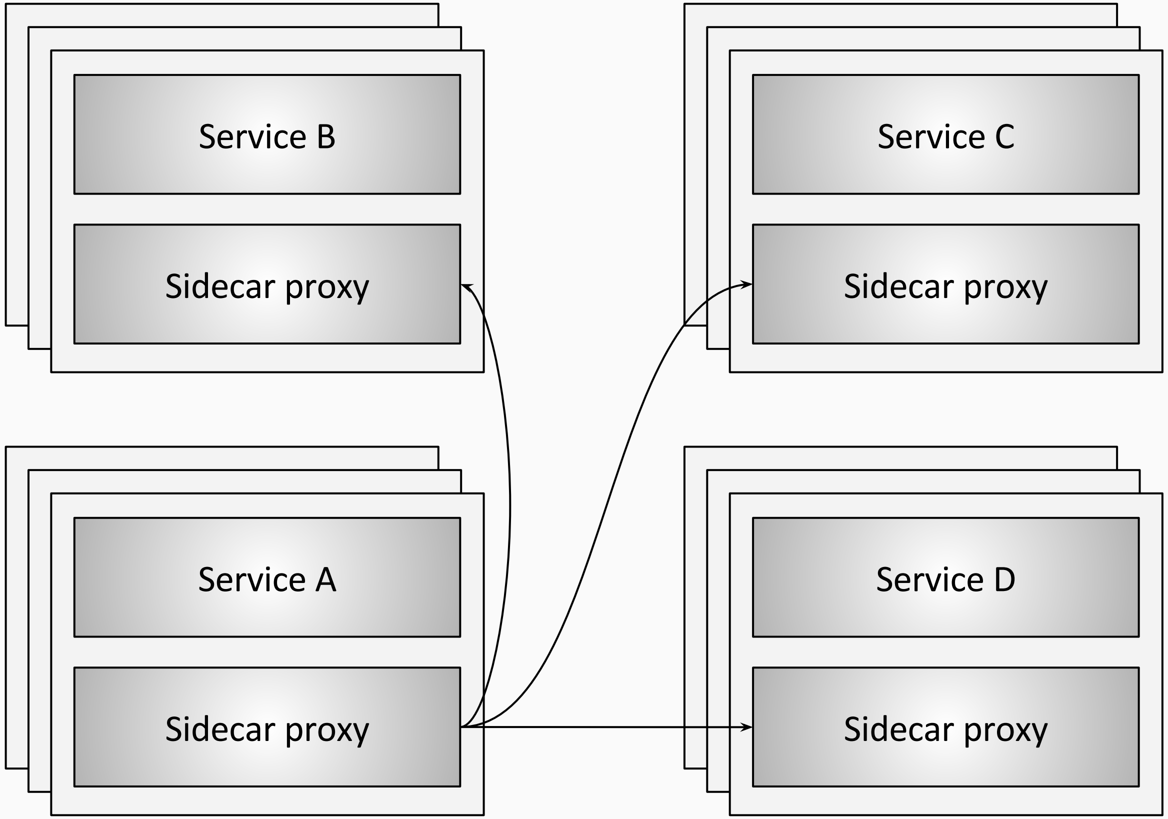
Matt Klein / @mattklein123, Software Engineer @Lyft

What is Envoy and the service mesh?

The network should be transparent to applications. When network and application problems do occur it should be easy to determine the source of the problem.

envoy

Service mesh refresher



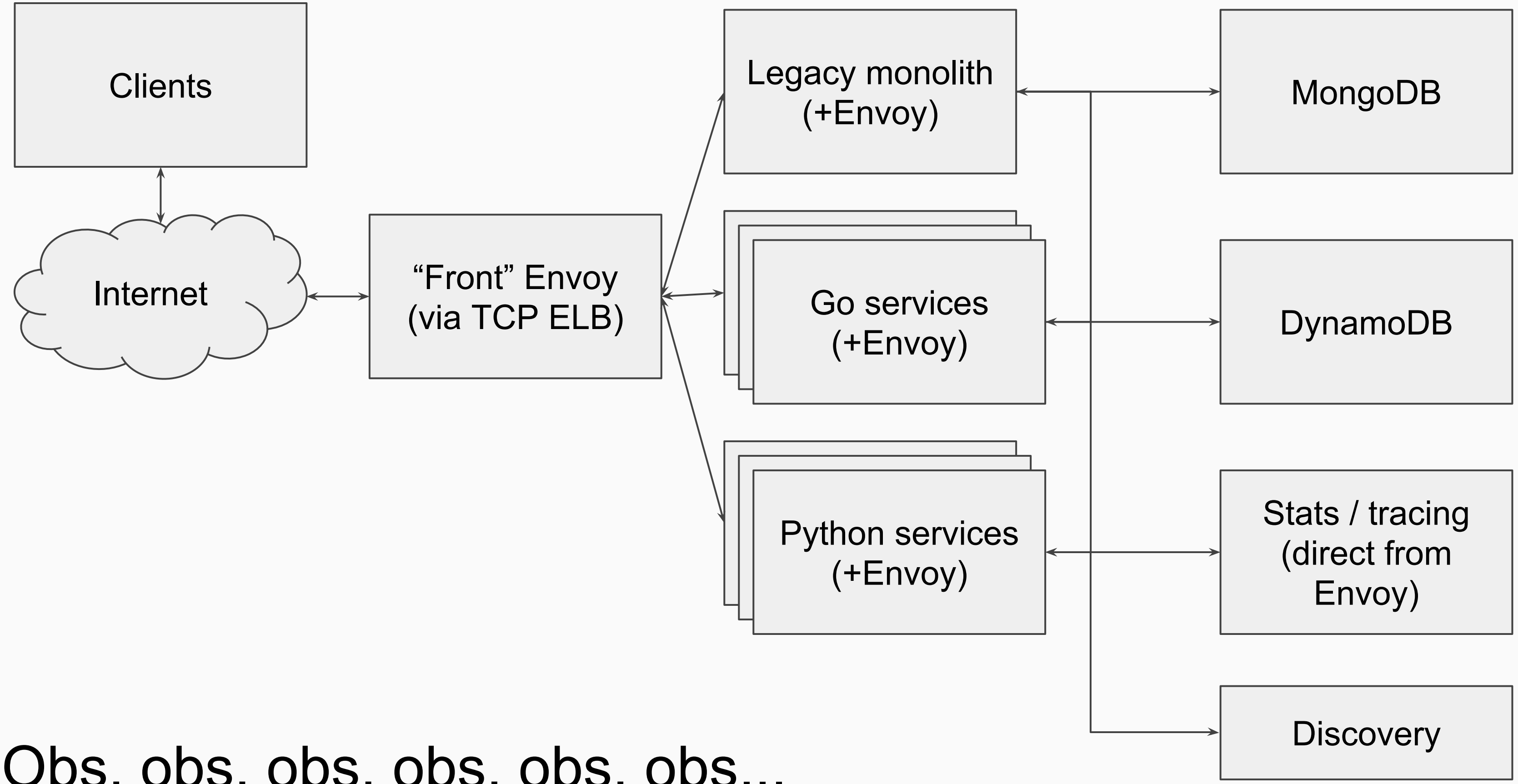
Envoy refresher

- **Out of process architecture**
- **Modern C++11 code base**
- **L3/L4 filter architecture**
- **HTTP L7 filter architecture**
- **HTTP/2** first
- **Service discovery** and **active/passive health checking**
- **Advanced load balancing**
- Best in class **observability** (stats, logging, and tracing)
- **Edge proxy**

Observability

- **Observability** is by far the most important thing that Envoy provides.
- Having all SoA traffic transit through Envoy gives us a single place where we can:
  - Produce consistent **statistics** for every hop
  - Create and propagate a stable **request ID / tracing context**
  - Consistent **logging**
  - Distributed **tracing**

Lyft today

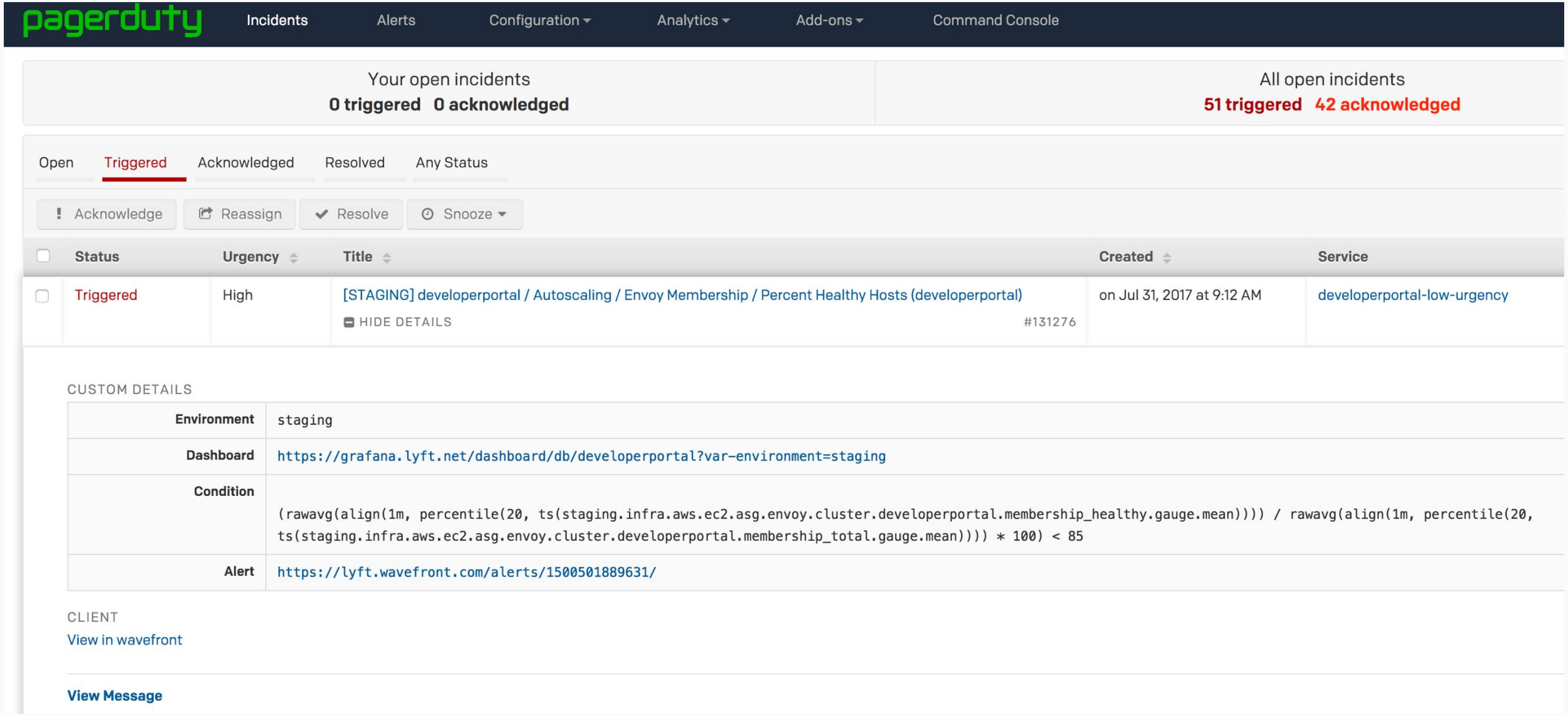


Obs, obs, obs, obs, obs, obs...

State of incident handling @lyft: something breaks

The page goes out (hopefully). What is the best case scenario of what follows?

State of incident handling @lyft: the page



pagerduty | Incidents | Alerts | Configuration | Analytics | Add-ons | Command Console

Your open incidents
0 triggered 0 acknowledged

All open incidents
51 triggered 42 acknowledged

Open | Triggered | Acknowledged | Resolved | Any Status

Acknowledge | Reassign | Resolve | Snooze

Status	Urgency	Title	Created	Service
Triggered	High	[STAGING] developerportal / Autoscaling / Envoy Membership / Percent Healthy Hosts (developerportal) HIDE DETAILS #131276	on Jul 31, 2017 at 9:12 AM	developerportal-low-urgency

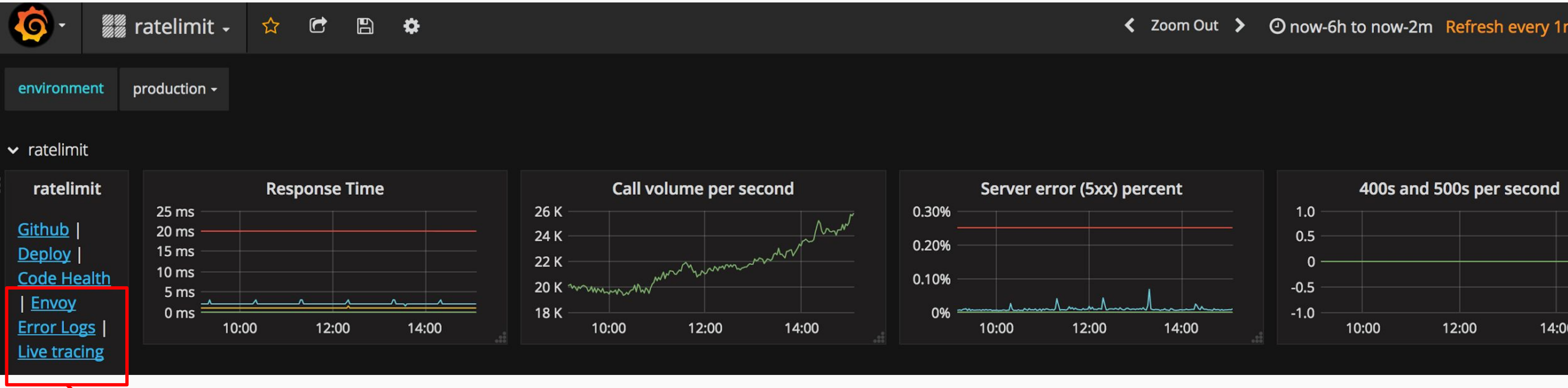
CUSTOM DETAILS

Environment	`staging`
Dashboard	`https://grafana.lyft.net/dashboard/db/developerportal?var-environment=staging`
Condition	`(rawavg(align(1m, percentile(20, ts(staging.infra.aws.ec2.asg.envoy.cluster.developerportal.membership_healthy.gauge.mean)))) / rawavg(align(1m, percentile(20, ts(staging.infra.aws.ec2.asg.envoy.cluster.developerportal.membership_total.gauge.mean)))) * 100) < 85`
Alert	`https://lyft.wavefront.com/alerts/1500501889631/`

CLIENT
View in wavefront

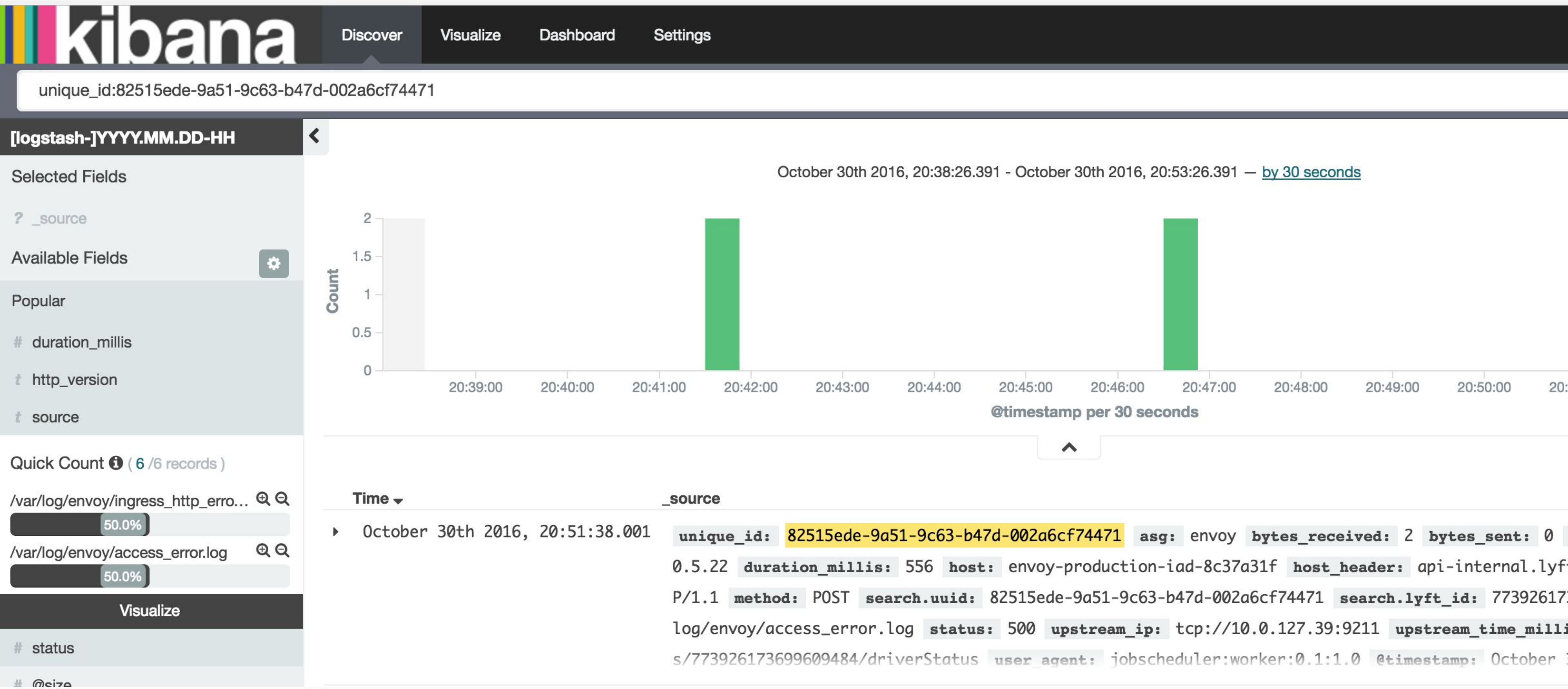
View Message

State of incident handling @lyft: per service auto-generated panel



Links to logging and tracing

State of incident handling @lyft: logging



unique_id:82515ede-9a51-9c63-b47d-002a6cf74471

[logstash-]YYYY.MM.DD-HH

Selected Fields

? _source

Available Fields

Popular

duration_millis

http_version

source

Quick Count (6 /6 records)

/var/log/envoy/ingress_http_erro… 50.0%

/var/log/envoy/access_error.log 50.0%

Visualize

status

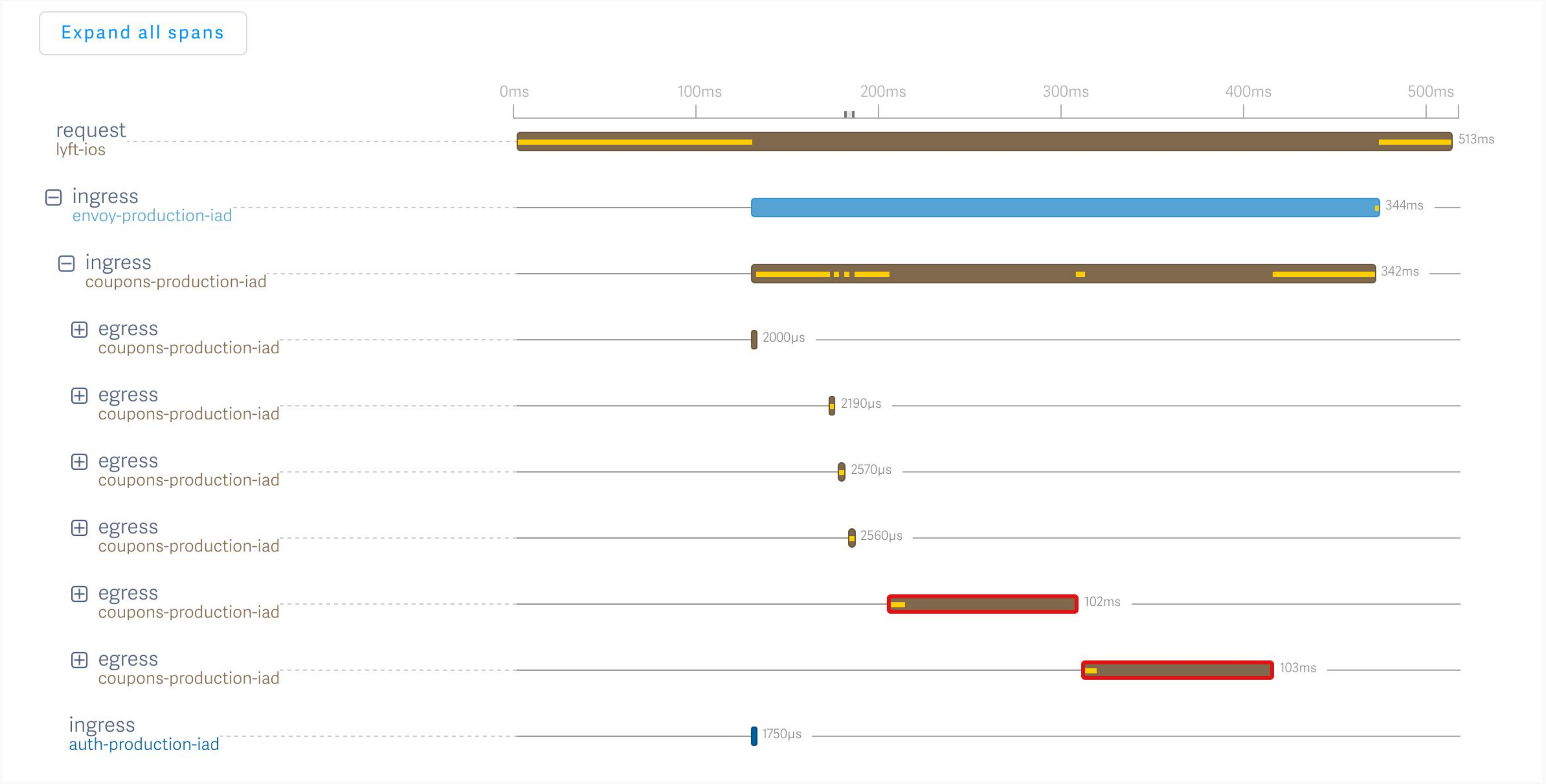
@size

October 30th 2016, 20:38:26.391 - October 30th 2016, 20:53:26.391 — by 30 seconds

Time	_source
October 30th 2016, 20:51:38.001	

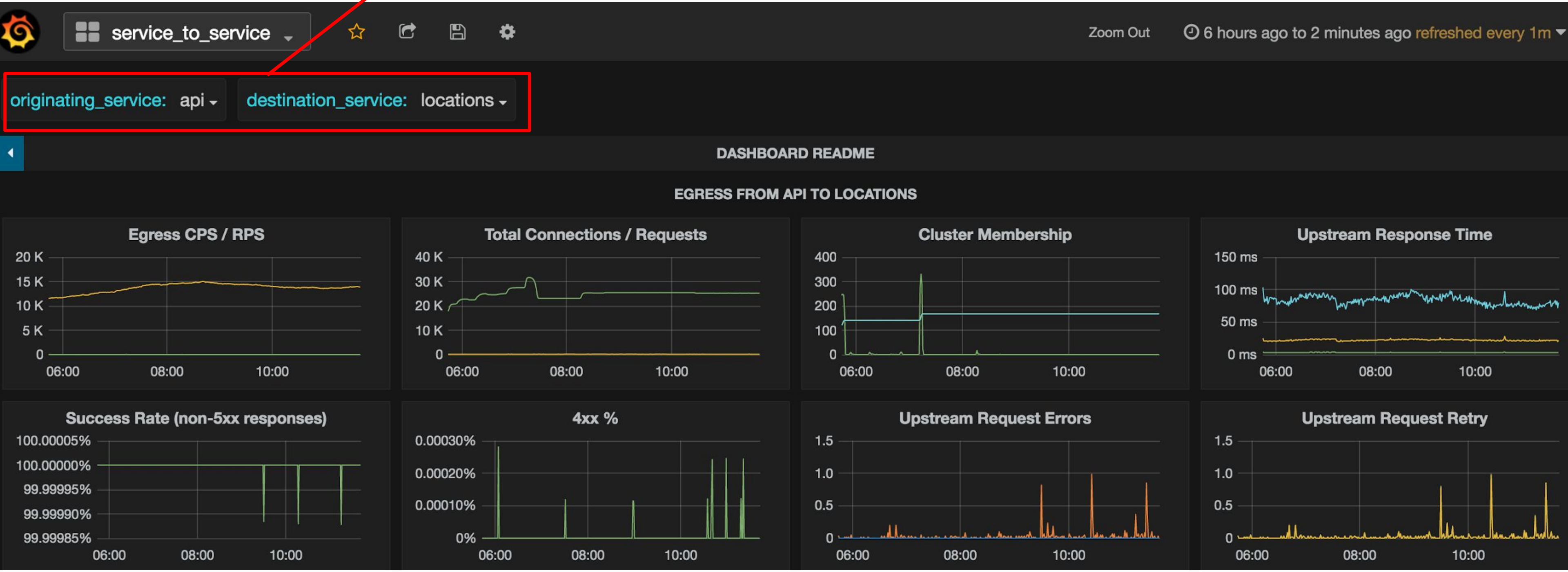
```
unique_id: 82515ede-9a51-9c63-b47d-002a6cf74471 asg: envoy bytes_received: 2 bytes_sent: 0
0.5.22 duration_millis: 556 host: envoy-production-iad-8c37a31f host_header: api-internal.lyf
P/1.1 method: POST search.uuid: 82515ede-9a51-9c63-b47d-002a6cf74471 search.lyft_id: 77392617
log/envoy/access_error.log status: 500 upstream_ip: tcp://10.0.127.39:9211 upstream_time_milli
s/773926173699609484/driverStatus user_agent: jobscheduler:worker:0.1:1.0 @timestamp: October
```

State of incident handling @lyft: distributed tracing

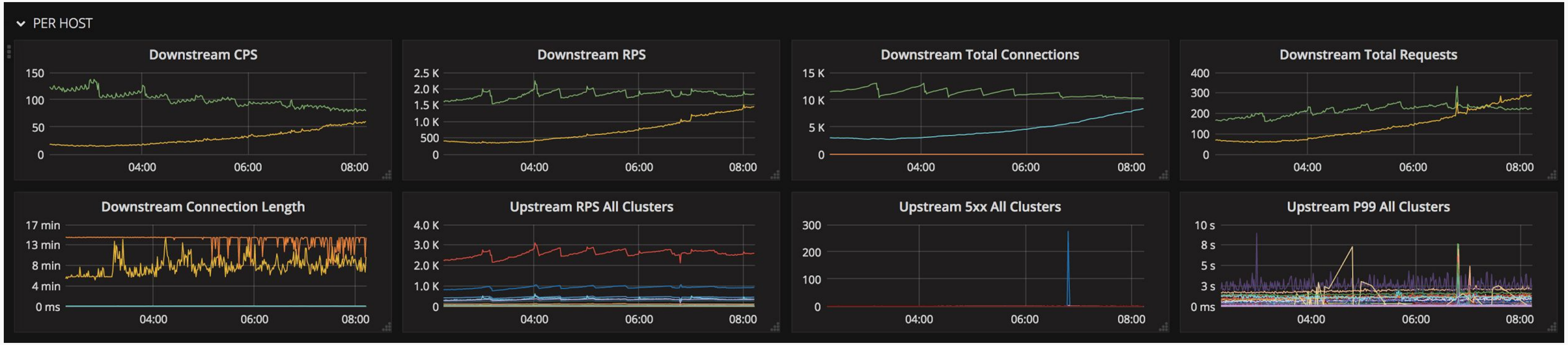


State of incident handling @lyft: service to service template dashboard

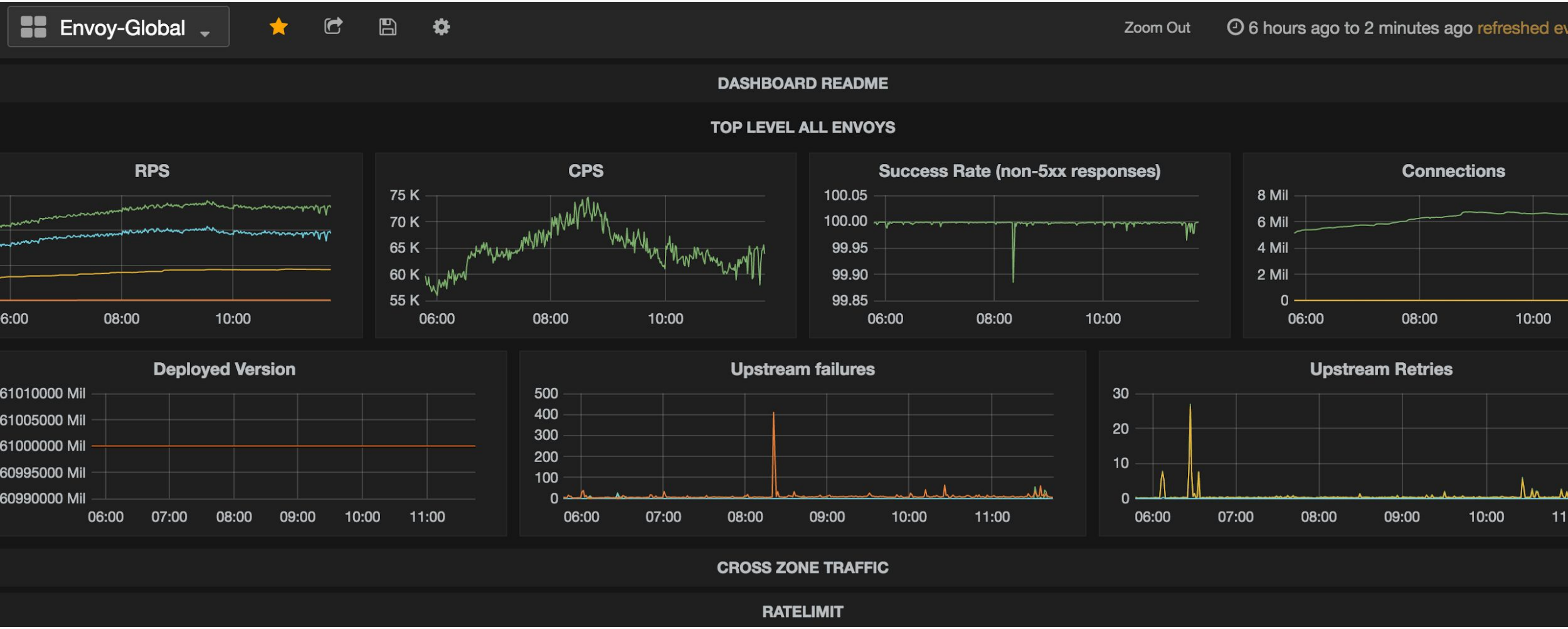
Template with drop down for every service



State of incident handling @lyft: edge proxy


PER HOST
Downstream CPS
Downstream RPS
Downstream Total Connections
Downstream Total Requests
Downstream Connection Length
Upstream RPS All Clusters
Upstream 5xx All Clusters
Upstream P99 All Clusters
04:00
06:00
08:00

State of incident handling @lyft: global health dashboard

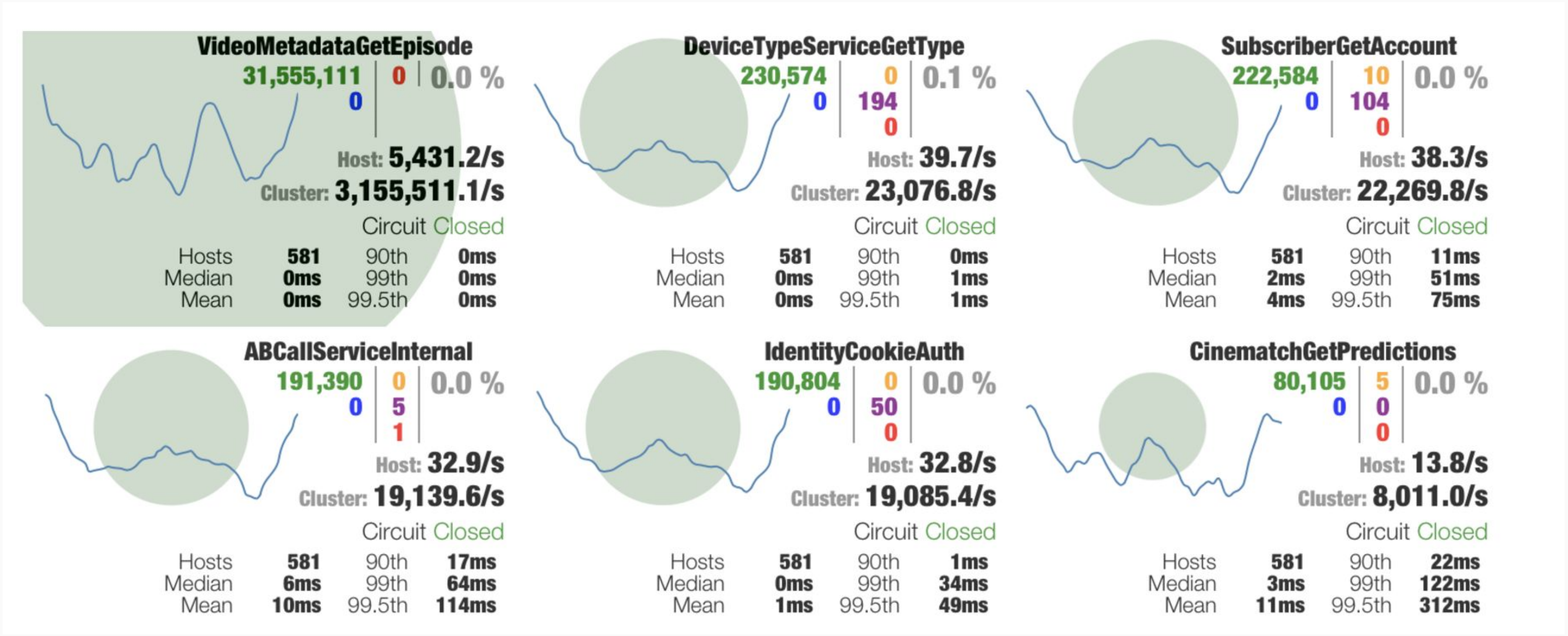


Future of microservice observability: problems

- Dev/Ops have too many data sources that are **not linked**.
- **Cognitive load** of different data sources make issue investigation with traditional stats, logging, and tracing is VERY high
- **Service mesh** yields an observability base that allows us to do *incredible things by default*.

How can we reimagine observability and operations in the age of the service mesh?

State of incident handling: Hystrix



VideoMetadataGetEpisode
31,555,111 | 0 | 0.0 %
0
Host: 5,431.2/s
Cluster: 3,155,511.1/s
Circuit Closed
Hosts 581 90th 0ms
Median 0ms 99th 0ms
Mean 0ms 99.5th 0ms

DeviceTypeServiceGetType
230,574 | 0 | 0.1 %
0 | 194
0
Host: 39.7/s
Cluster: 23,076.8/s
Circuit Closed
Hosts 581 90th 0ms
Median 0ms 99th 1ms
Mean 0ms 99.5th 1ms

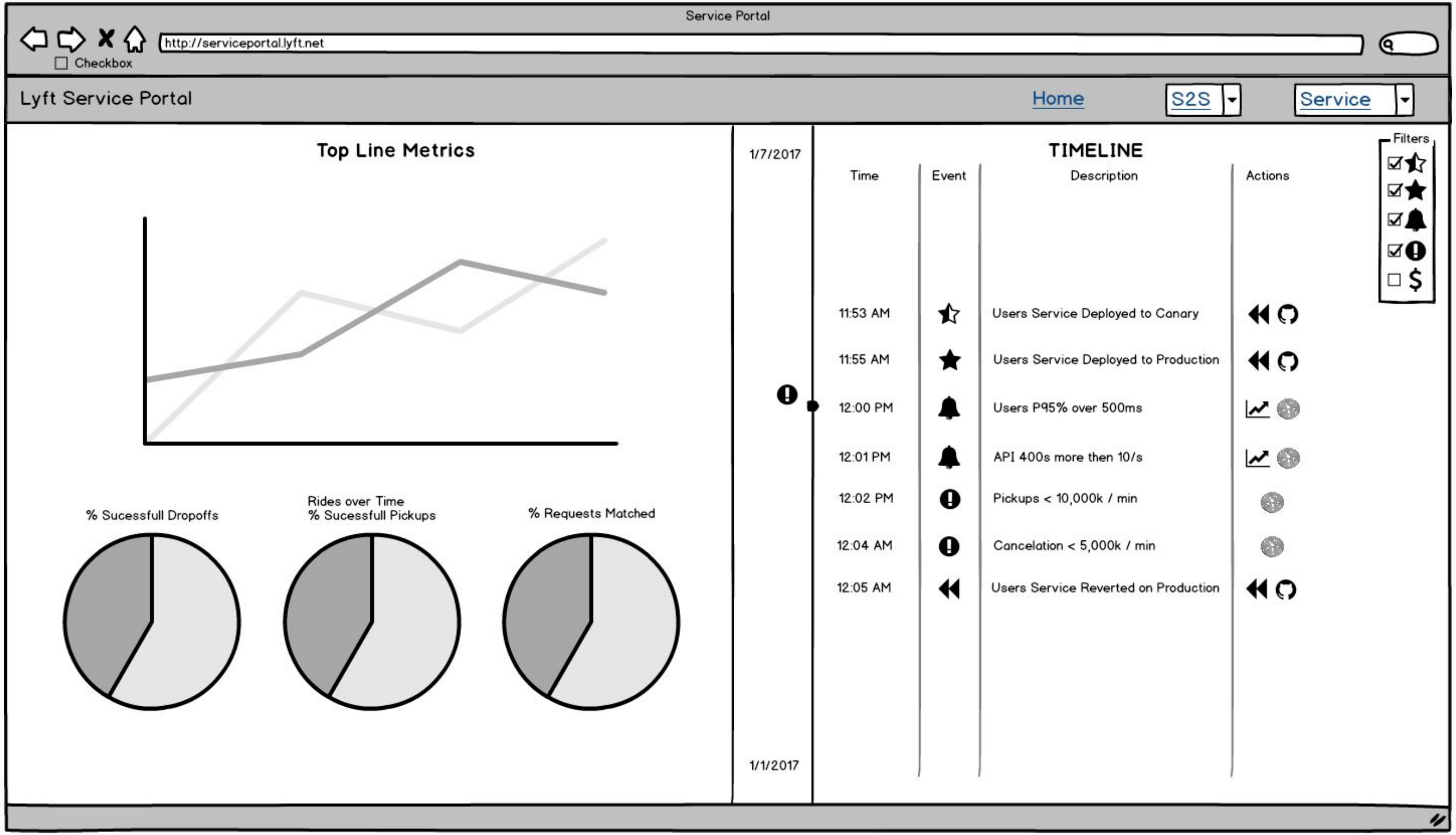
SubscriberGetAccount
222,584 | 10 | 0.0 %
0 | 104
0
Host: 38.3/s
Cluster: 22,269.8/s
Circuit Closed
Hosts 581 90th 11ms
Median 2ms 99th 51ms
Mean 4ms 99.5th 75ms

ABCallServiceInternal
191,390 | 0 | 0.0 %
0 | 5
1
Host: 32.9/s
Cluster: 19,139.6/s
Circuit Closed
Hosts 581 90th 17ms
Median 6ms 99th 64ms
Mean 10ms 99.5th 114ms

IdentityCookieAuth
190,804 | 0 | 0.0 %
0 | 50
0
Host: 32.8/s
Cluster: 19,085.4/s
Circuit Closed
Hosts 581 90th 1ms
Median 0ms 99th 34ms
Mean 1ms 99.5th 49ms

CinematchGetPredictions
80,105 | 5 | 0.0 %
0 | 0
0
Host: 13.8/s
Cluster: 8,011.0/s
Circuit Closed
Hosts 581 90th 22ms
Median 3ms 99th 122ms
Mean 11ms 99.5th 312ms

Service portal sketch: landing



Service Portal

http://serviceportal.lyft.net

☐ Checkbox

Lyft Service Portal — Home — S2S ▾ — Service ▾

Top Line Metrics

% Sucessfull Dropoffs

Rides over Time
% Sucessfull Pickups

% Requests Matched

1/7/2017

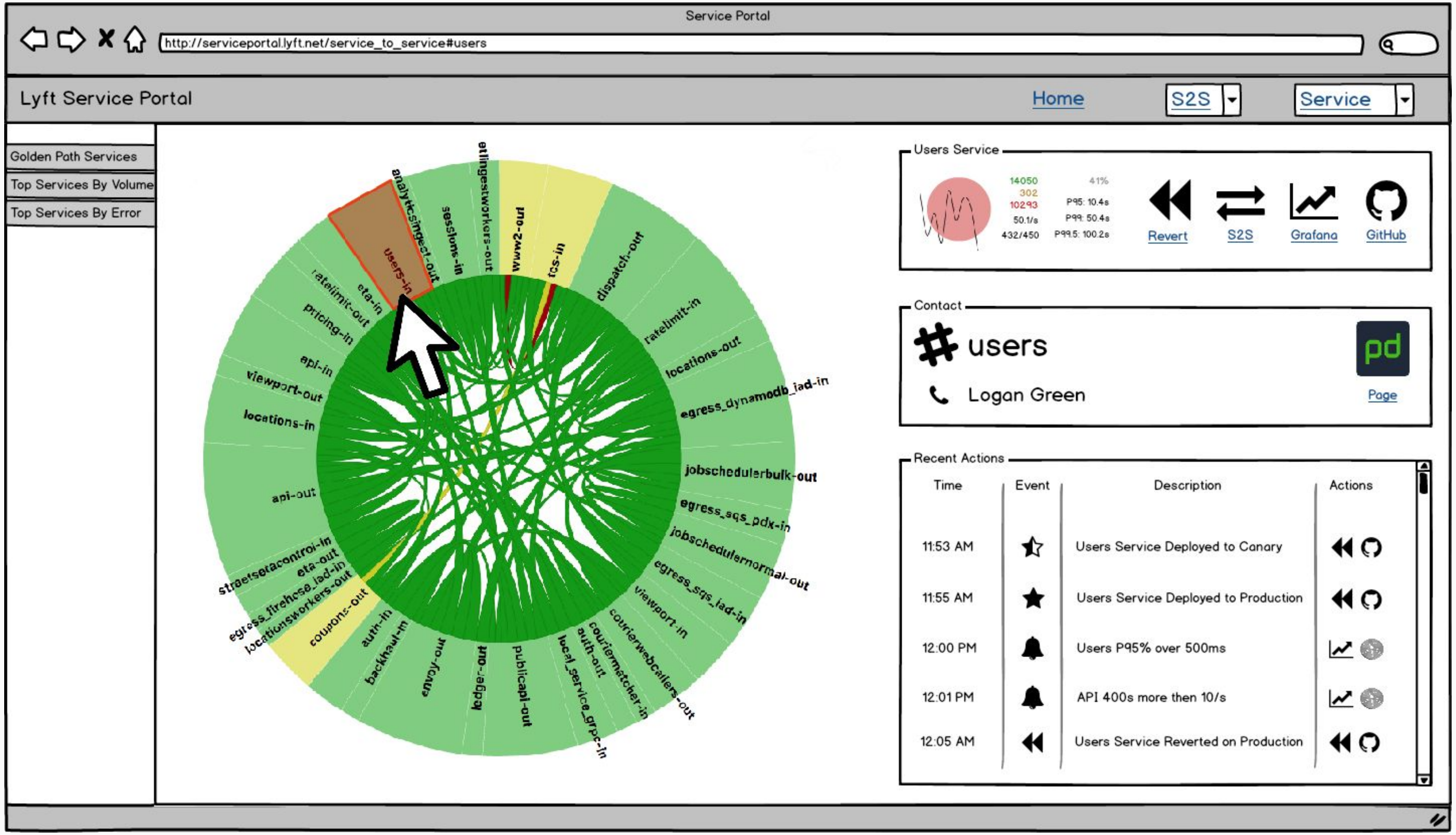
1/1/2017

TIMELINE

Time	Event	Description	Actions
11:53 AM	☆	Users Service Deployed to Canary	⏪
11:55 AM	★	Users Service Deployed to Production	⏪
12:00 PM	🔔	Users P95% over 500ms	📈
12:01 PM	🔔	API 400s more then 10/s	📈
12:02 PM	❗	Pickups < 10,000k / min	
12:04 AM	❗	Cancelation < 5,000k / min	
12:05 AM	⏪	Users Service Reverted on Production	⏪

Filters
☑ ☆
☑ ★
☑ 🔔
☑ ❗
☐ $

Service portal sketch: service detail



Service Portal

http://serviceportal.lyft.net/service_to_service#users

Lyft Service Portal

Home | S2S | Service

- Golden Path Services
- Top Services By Volume
- Top Services By Error

Users Service

14050
302
10293
50.1/s
432/450

41%
P95: 10.4s
P99: 50.4s
P99.5: 100.2s

Revert | S2S | Grafana | GitHub

Contact

users

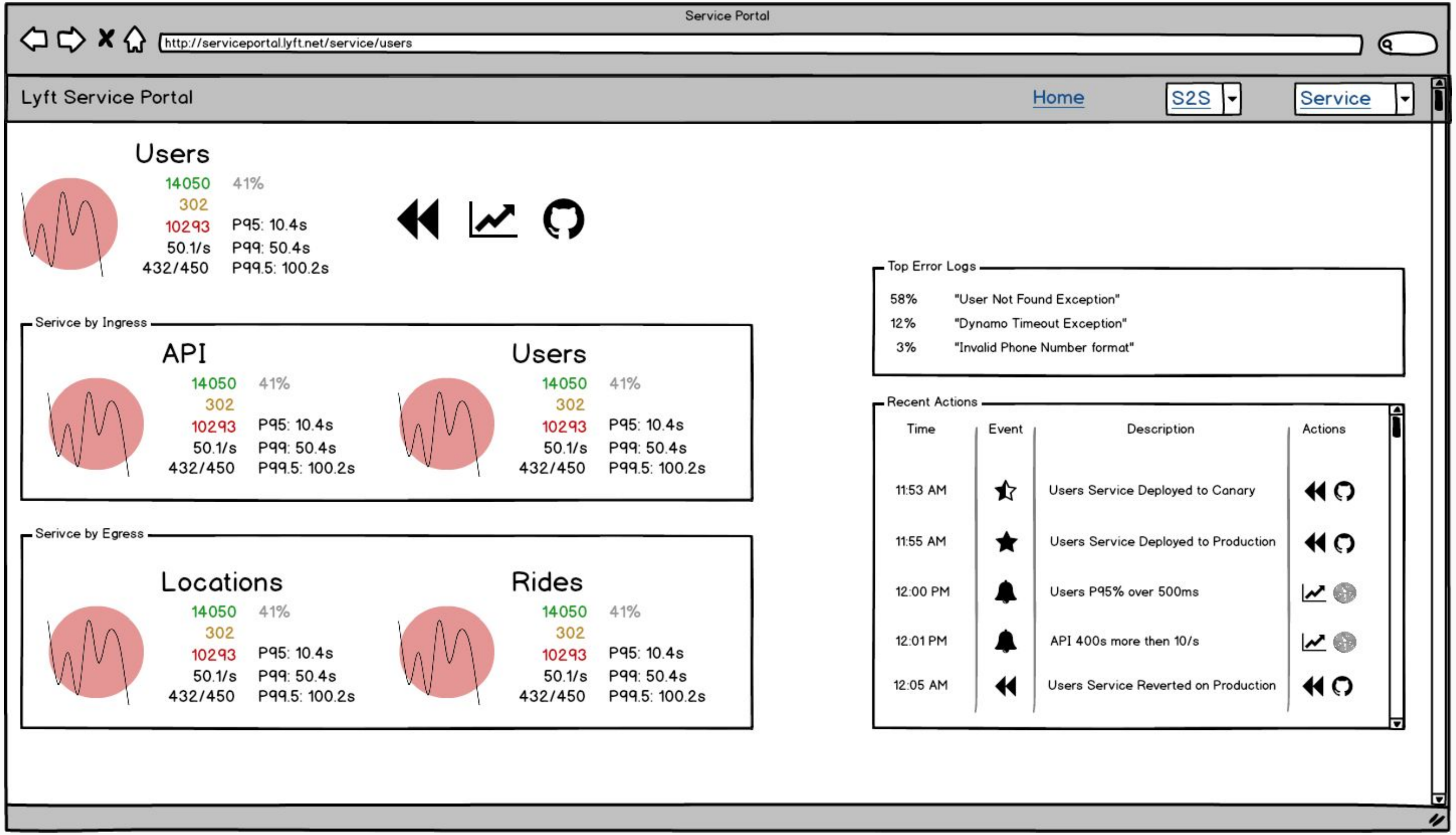
Logan Green

Page

Recent Actions

Time	Event	Description	Actions
11:53 AM	☆	Users Service Deployed to Canary	
11:55 AM	★	Users Service Deployed to Production	
12:00 PM	🔔	Users P95% over 500ms	
12:01 PM	🔔	API 400s more then 10/s	
12:05 AM	⏪	Users Service Reverted on Production	

Service portal sketch: service detail alternate



Service Portal

http://serviceportal.lyft.net/service/users

Lyft Service Portal

Home · S2S · Service

Users

14050 41%
302
10293 P95: 10.4s
50.1/s P99: 50.4s
432/450 P99.5: 100.2s

Serivce by Ingress

API

14050 41%
302
10293 P95: 10.4s
50.1/s P99: 50.4s
432/450 P99.5: 100.2s

Users

14050 41%
302
10293 P95: 10.4s
50.1/s P99: 50.4s
432/450 P99.5: 100.2s

Serivce by Egress

Locations

14050 41%
302
10293 P95: 10.4s
50.1/s P99: 50.4s
432/450 P99.5: 100.2s

Rides

14050 41%
302
10293 P95: 10.4s
50.1/s P99: 50.4s
432/450 P99.5: 100.2s

Top Error Logs

58%	"User Not Found Exception"
12%	"Dynamo Timeout Exception"
3%	"Invalid Phone Number format"

Recent Actions

Time	Event	Description	Actions
11:53 AM	☆	Users Service Deployed to Canary	
11:55 AM	★	Users Service Deployed to Production	
12:00 PM		Users P95% over 500ms	
12:01 PM		API 400s more then 10/s	
12:05 AM		Users Service Reverted on Production	

Service portal sketch: service detail

Optimal visualization of high level state

Actions relevant to mitigation

Machine learning to identify problems

RBAC and versioning

How do we get there?

- A **universal data plane** like Envoy provides unified APIs for control as well as consistent observability output.
- Allows us to build **more feature-rich full service mesh solutions** such as Istio.
- When we assume the existence of the service mesh, we can **focus on an incredible UI/UX** instead of constantly trying to keep every application up to date.
- Assume that service mesh is the future… **All data is available**.
- We need to start building the UI/UX/ML of the future for distributed system command control. **Need to start now!**

Q&A

- Thanks for coming! Questions welcome on Twitter: **@mattklein123**
- We are super excited about building a **community** around Envoy. Talk to us if you need help getting started.
- **Lyft is hiring!**