Three Pillars with Zero Answers

A New Observability Scorecard

December 6, 2018

LIGHTSTEP

First, a Critique

The Conventional Wisdom

Observing microservices is hard

Google and Facebook solved this (right???)

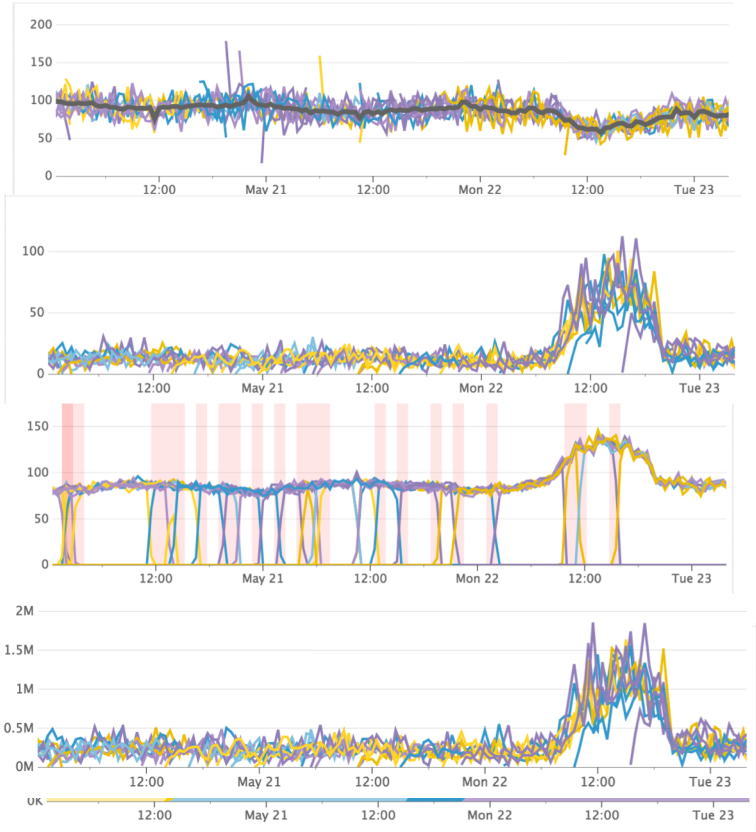
They used **Metrics, Logging, and Distributed Tracing**…

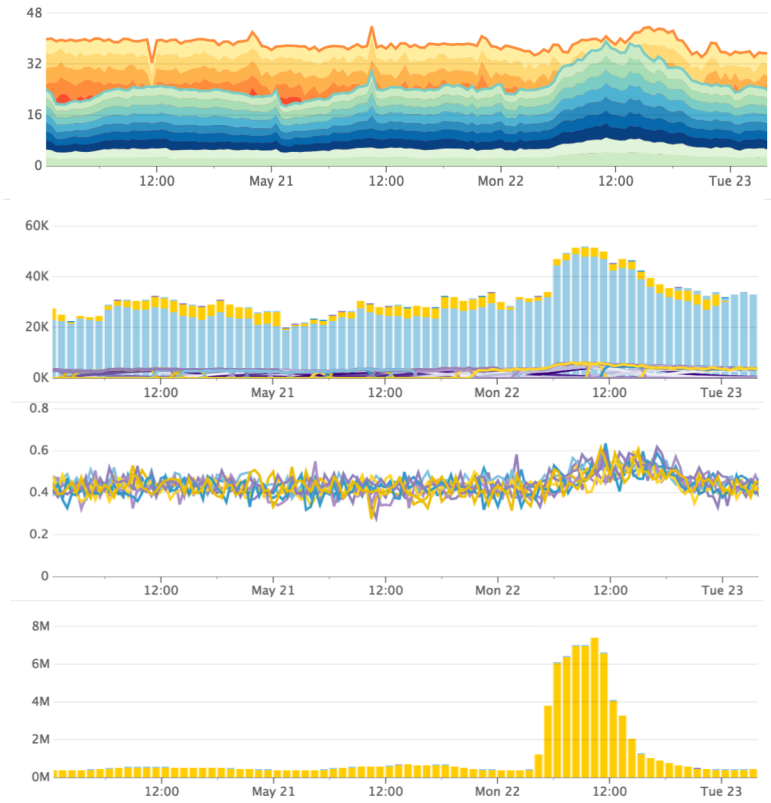
So we should, too.

The Three Pillars of Observability

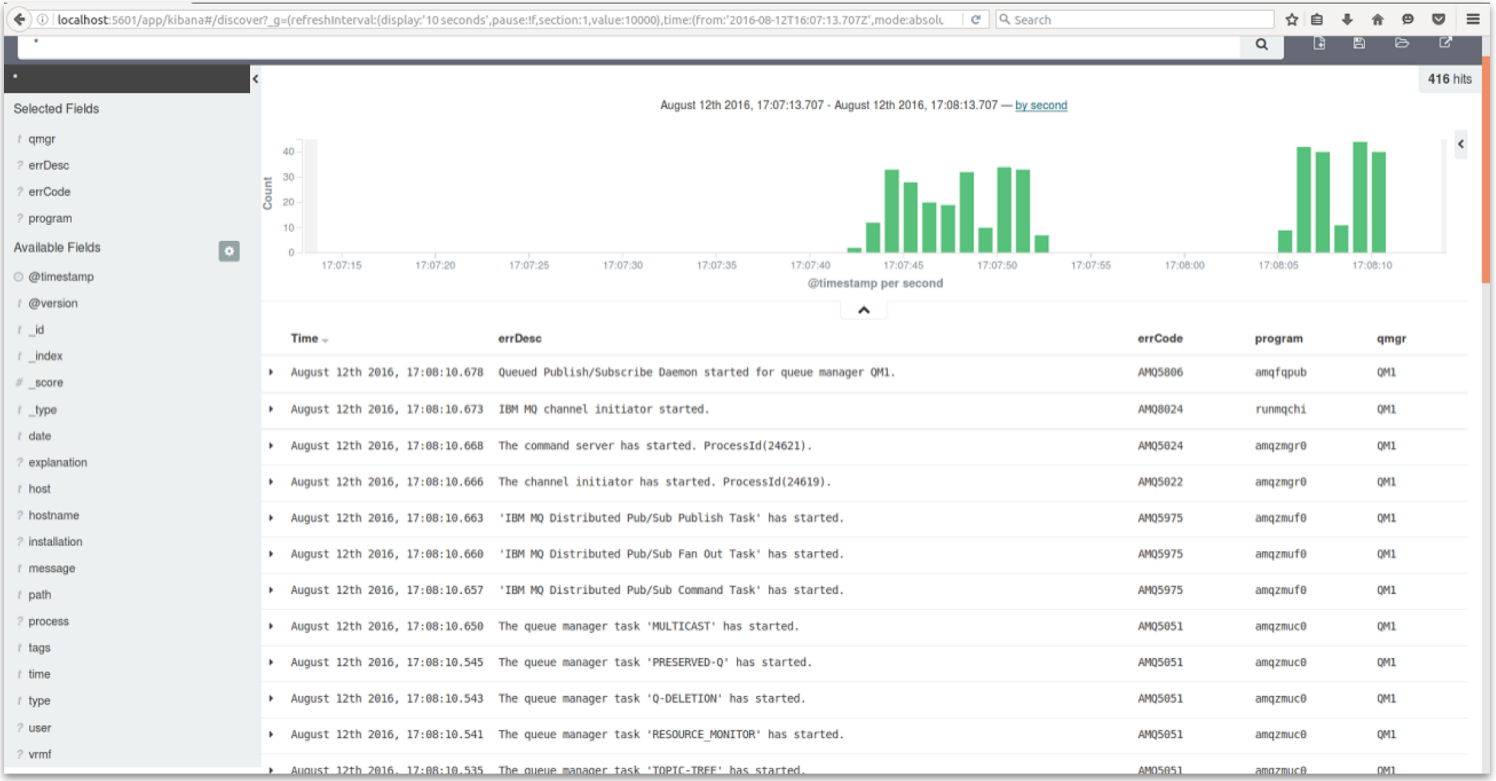
- Metrics
- Logging
- Distributed Tracing

Metrics!

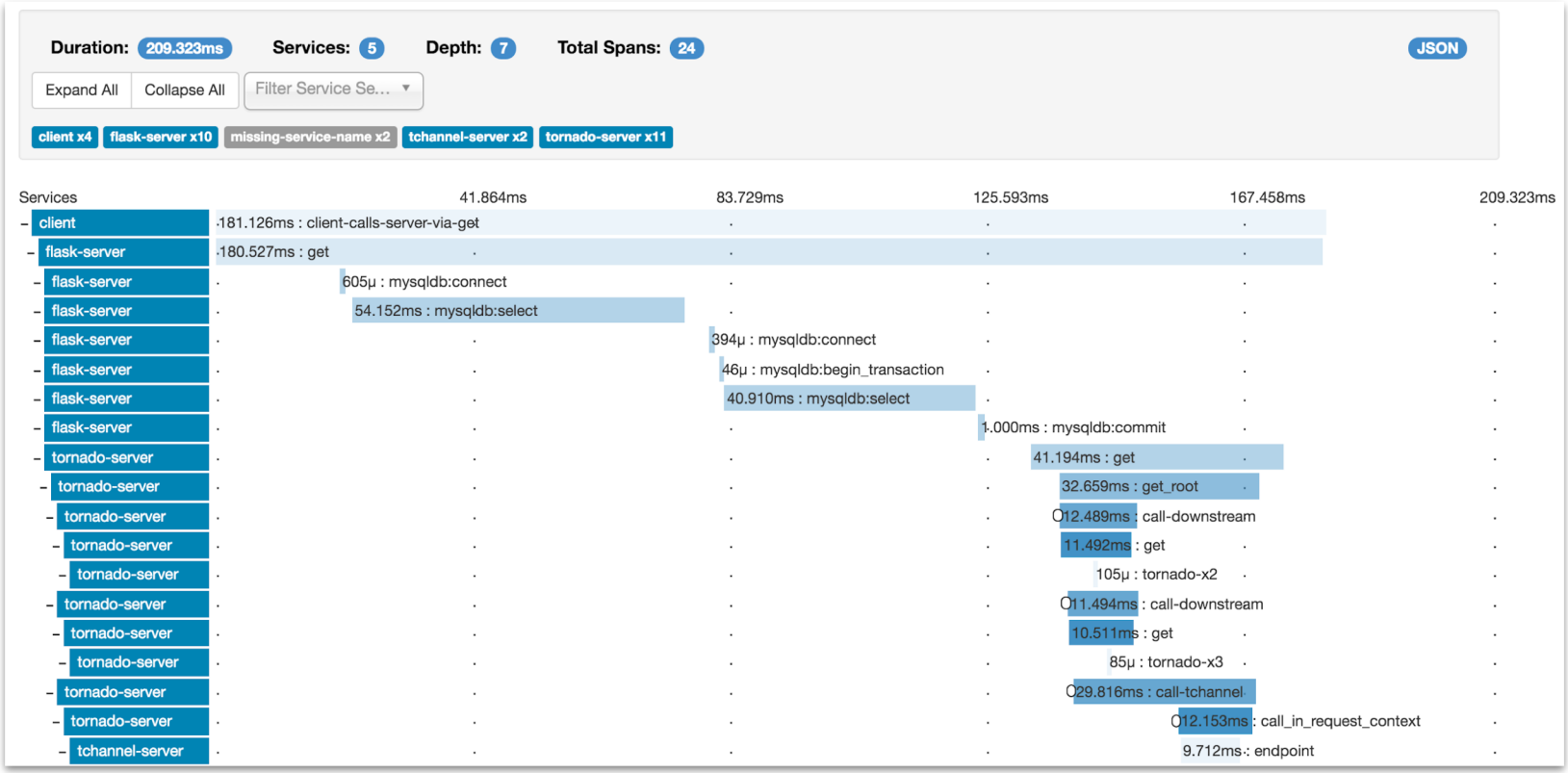

200
150
100
50
0
12:00
May 21
12:00
Mon 22
12:00
Tue 23
100
50
0
150
100
50
0
2M
1.5M
1M
0.5M
0M
0K


48
32
16
0
60K
40K
20K
0K
0.8
0.6
0.4
0.2
0
8M
6M
4M
2M
0M
12:00
May 21
12:00
Mon 22
12:00
Tue 23

Logging!

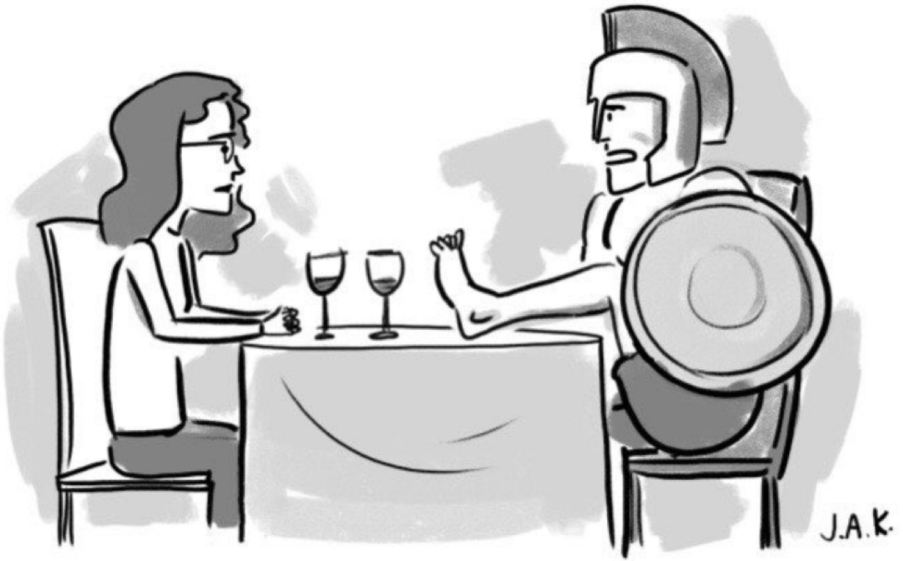

localhost:5601/app/kibana#/discover?_g=(refreshInterval:(display:'10 seconds',pause:!f,section:1,value:10000),time:(from:'2016-08-12T16:07:13.707Z',mode:absolu
Search
Selected Fields
t qmgr
? errDesc
? errCode
? program
Available Fields
@timestamp
t @version
t _id
t _index
_score
t _type
t date
? explanation
t host
? hostname
? installation
t message
t path
? process
t tags
t time
t type
? user
? vrmf
416 hits
August 12th 2016, 17:07:13.707 - August 12th 2016, 17:08:13.707 — by second
Count
40
30
20
10
0
17:07:15
17:07:20
17:07:25
17:07:30
17:07:35
17:07:40
17:07:45
17:07:50
17:07:55
17:08:00
17:08:05
17:08:10
@timestamp per second
Time
errDesc
errCode
program
qmgr
August 12th 2016, 17:08:10.678 Queued Publish/Subscribe Daemon started for queue manager QM1. AMQ5806 amqfqpub QM1
August 12th 2016, 17:08:10.673 IBM MQ channel initiator started. AMQ8024 runmqchi QM1
August 12th 2016, 17:08:10.668 The command server has started. ProcessId(24621). AMQ5024 amqzmgr0 QM1
August 12th 2016, 17:08:10.666 The channel initiator has started. ProcessId(24619). AMQ5022 amqzmgr0 QM1
August 12th 2016, 17:08:10.663 'IBM MQ Distributed Pub/Sub Publish Task' has started. AMQ5975 amqzmuf0 QM1
August 12th 2016, 17:08:10.660 'IBM MQ Distributed Pub/Sub Fan Out Task' has started. AMQ5975 amqzmuf0 QM1
August 12th 2016, 17:08:10.657 'IBM MQ Distributed Pub/Sub Command Task' has started. AMQ5975 amqzmuf0 QM1
August 12th 2016, 17:08:10.650 The queue manager task 'MULTICAST' has started. AMQ5051 amqzmuc0 QM1
August 12th 2016, 17:08:10.545 The queue manager task 'PRESERVED-Q' has started. AMQ5051 amqzmuc0 QM1
August 12th 2016, 17:08:10.543 The queue manager task 'Q-DELETION' has started. AMQ5051 amqzmuc0 QM1
August 12th 2016, 17:08:10.541 The queue manager task 'RESOURCE_MONITOR' has started. AMQ5051 amqzmuc0 QM1
August 12th 2016, 17:08:10.535 The queue manager task 'TOPIC-TREE' has started. AMQ5051 amqzmuc0 QM1

Tracing!


Duration: 209.323ms
Services: 5
Depth: 7
Total Spans: 24
JSON
Expand All
Collapse All
Filter Service Se...
client x4
flask-server x10
missing-service-name x2
tchannel-server x2
tornado-server x11
Services
41.864ms
83.729ms
125.593ms
167.458ms
209.323ms
client
flask-server
flask-server
flask-server
flask-server
flask-server
flask-server
flask-server
tornado-server
tornado-server
tornado-server
tornado-server
tornado-server
tornado-server
tornado-server
tornado-server
tornado-server
tchannel-server
181.126ms : client-calls-server-via-get
180.527ms : get
605µ : mysqldb:connect
54.152ms : mysqldb:select
394µ : mysqldb:connect
46µ : mysqldb:begin_transaction
40.910ms : mysqldb:select
1.000ms : mysqldb:commit
41.194ms : get
32.659ms : get_root
12.489ms : call-downstream
11.492ms : get
105µ : tornado-x2
11.494ms : call-downstream
10.511ms : get
85µ : tornado-x3
29.816ms : call-tchannel
12.153ms : call_in_request_context
9.712ms : endpoint


SNAP
CRACKLE!
POP
CRACKLE!
POP

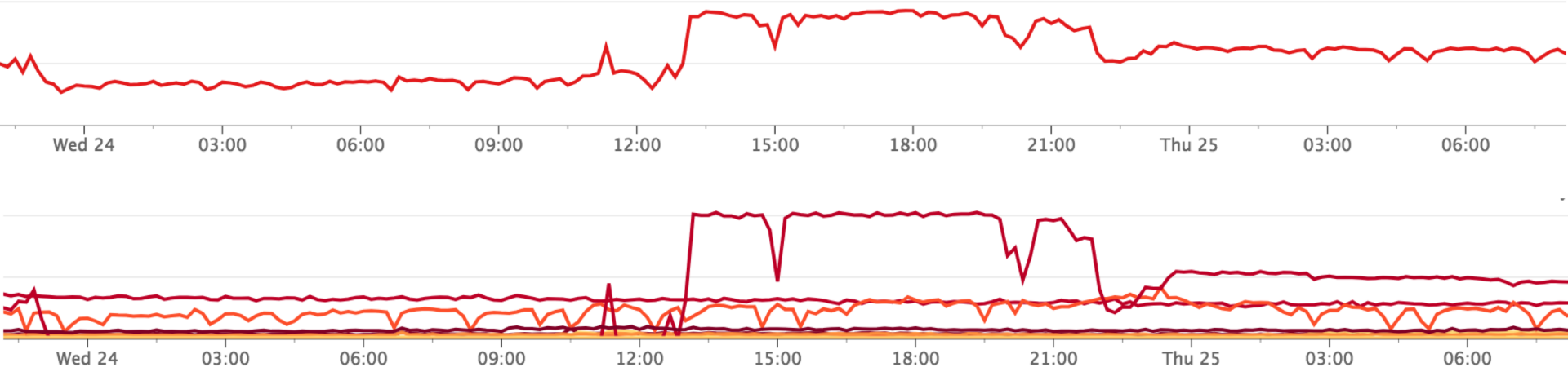
Fatal Flaws



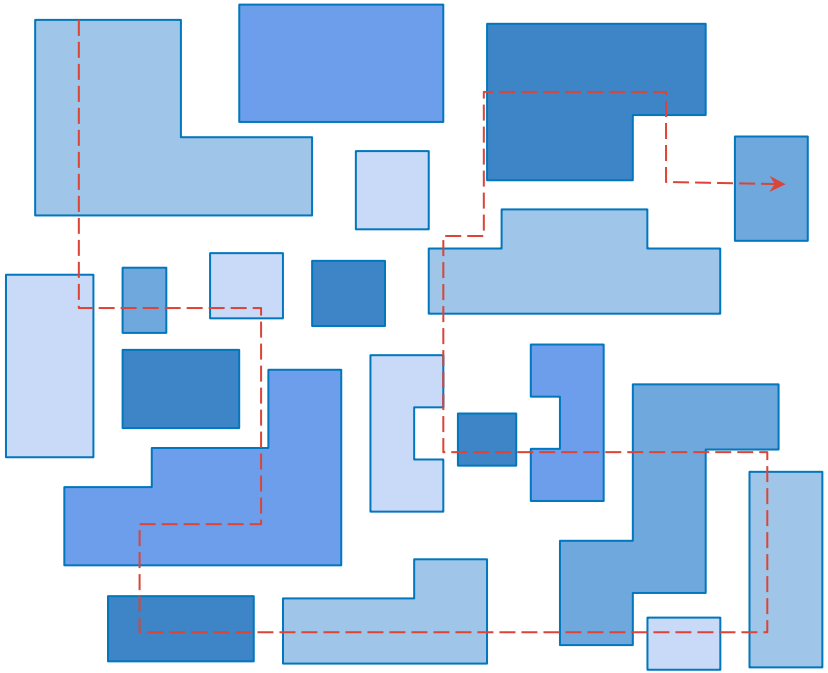
A word nobody knew in 2015…

Dimensions (aka “tags”) can explain variance in timeseries data (aka “metrics”) …

… but **cardinality**



Logging Data Volume: a reality check



```
          transaction rate
x      all microservices
x   cost of net+storage
x    weeks of retention
-----------------------
        way too much $$$$
```

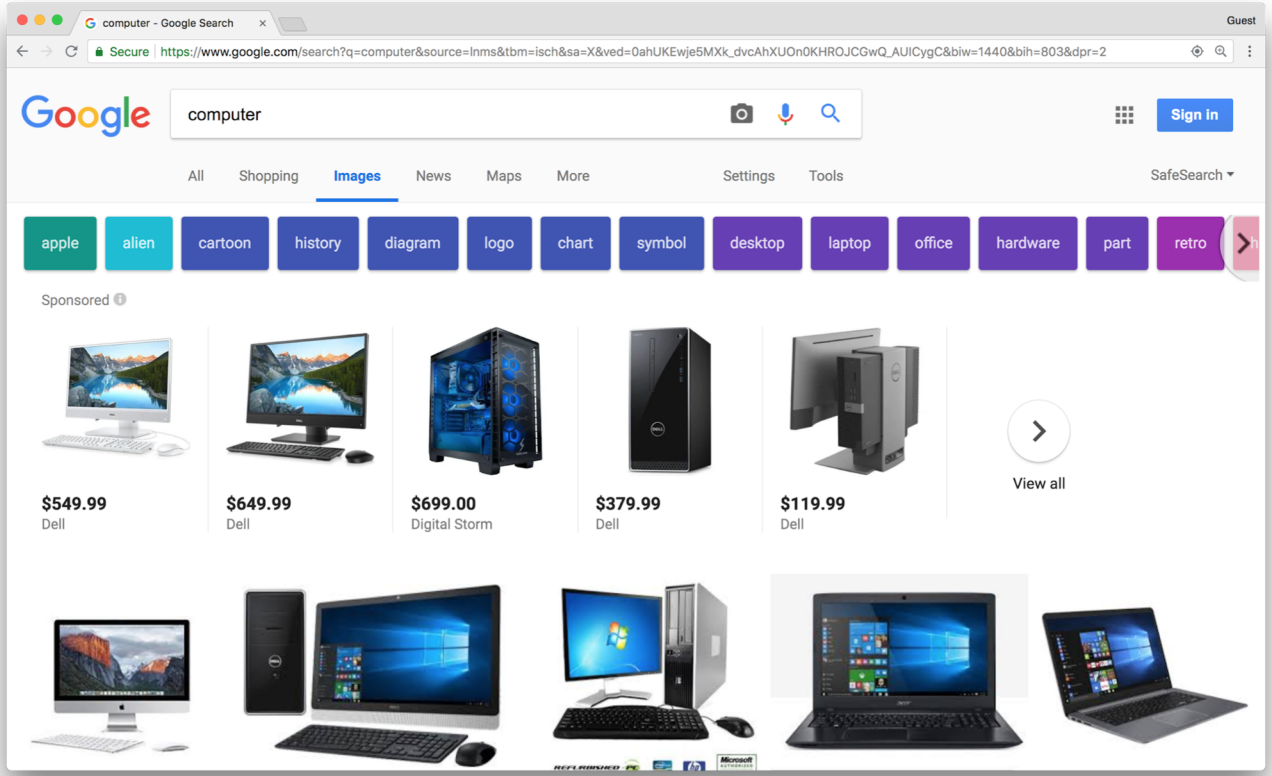
The Life of Transaction Data: Dapper

Stage	Overhead affects…	Retained
Instrumentation Executed	App	100.00%
Buffered within app process	App	000.10%
Flushed out of process	App	000.10%
Centralized regionally	Regional network + storage	000.10%
Centralized globally	WAN + storage	000.01%

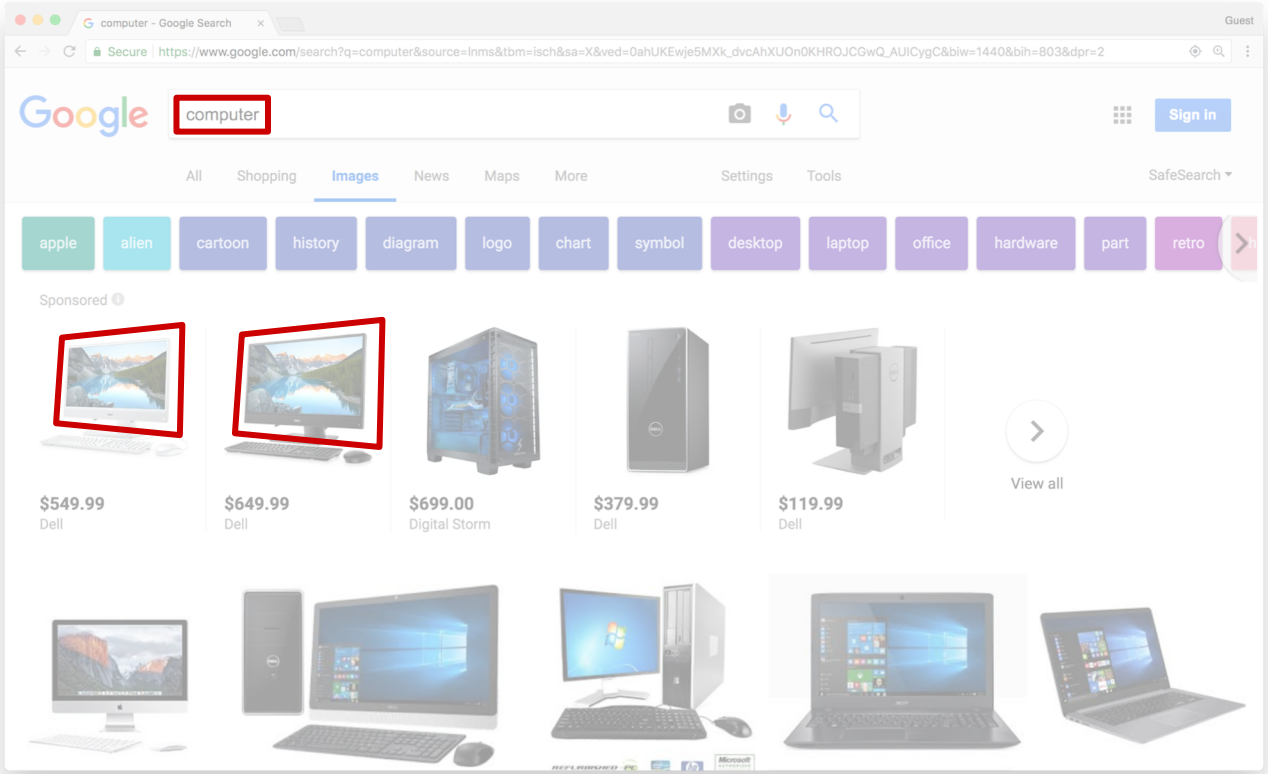
Fatal Flaws: A Review

	Logs	Metrics	Dist. Traces
TCO scales gracefully	—	✓	✓
Accounts for all data (i.e., unsampled)	✓	✓	—
Immune to cardinality	✓	—	✓

Data vs UI


computer - Google Search
Guest
Secure https://www.google.com/search?q=computer&source=lnms&tbm=isch&sa=X&ved=0ahUKEwje5MXk_dvcAhXUOn0KHROJCGwQ_AUICygC&biw=1440&bih=803&dpr=2
Google
computer
Sign in
All
Shopping
Images
News
Maps
More
Settings
Tools
SafeSearch
apple
alien
cartoon
history
diagram
logo
chart
symbol
desktop
laptop
office
hardware
part
retro
Sponsored
$549.99
Dell
$649.99
Dell
$699.00
Digital Storm
$379.99
Dell
$119.99
Dell
View all

Data vs UI

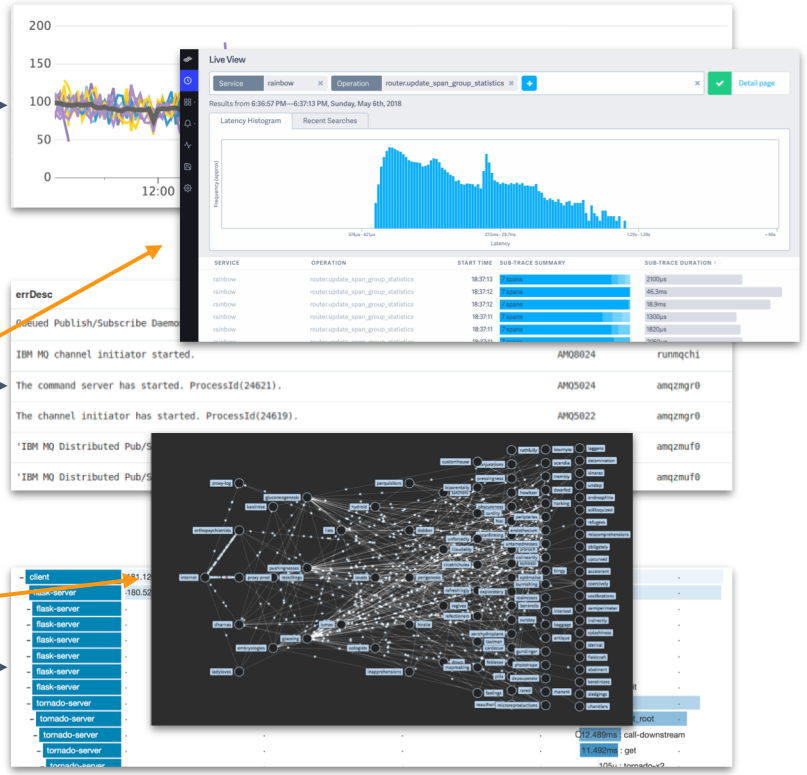

computer - Google Search
Guest
Secure https://www.google.com/search?q=computer&source=lnms&tbm=isch&sa=X&ved=0ahUKEwje5MXk_dvcAhXUOn0KHROJCGwQ_AUICygC&biw=1440&bih=803&dpr=2
Google
computer
Sign in
All
Shopping
Images
News
Maps
More
Settings
Tools
SafeSearch
apple
alien
cartoon
history
diagram
logo
chart
symbol
desktop
laptop
office
hardware
part
retro
Sponsored
$549.99
Dell
$649.99
Dell
$699.00
Digital Storm
$379.99
Dell
$119.99
Dell
View all

Data vs UI

Metrics

Logs

Traces



Metrics, Logs, and Traces are **Just Data**,

… not a feature or use case.

A New Scorecard for Observability

Observability: Quick Vocab Refresher

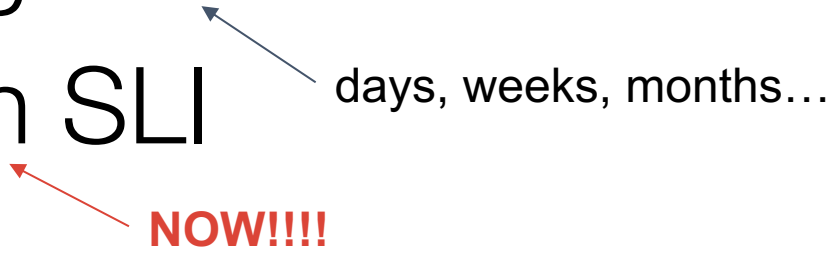
"SLI" = "Service Level Indicator"

TL;DR: An SLI is **an indicator of health** that a service's **consumers** would care about.

… *not* an indicator of its inner workings

Observability: Two Fundamental **Goals**

- Gradually improving an SLI
- Rapidly restoring an SLI

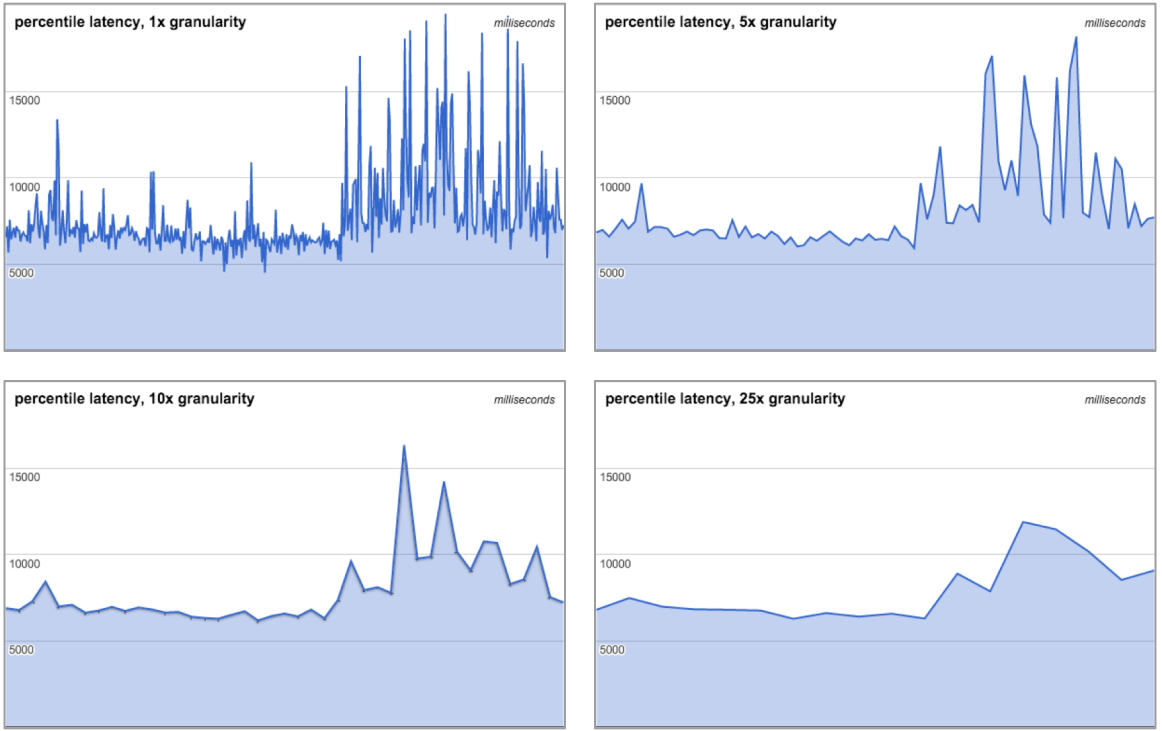


Reminder: “SLI” = “Service Level Indicator”

Observability: Two Fundamental **Activities**

1. Detection: perfect SLI capture
2. Refinement: reduce the search space

An interlude about stats frequency



Scorecard >> **Detection**

Specificity:

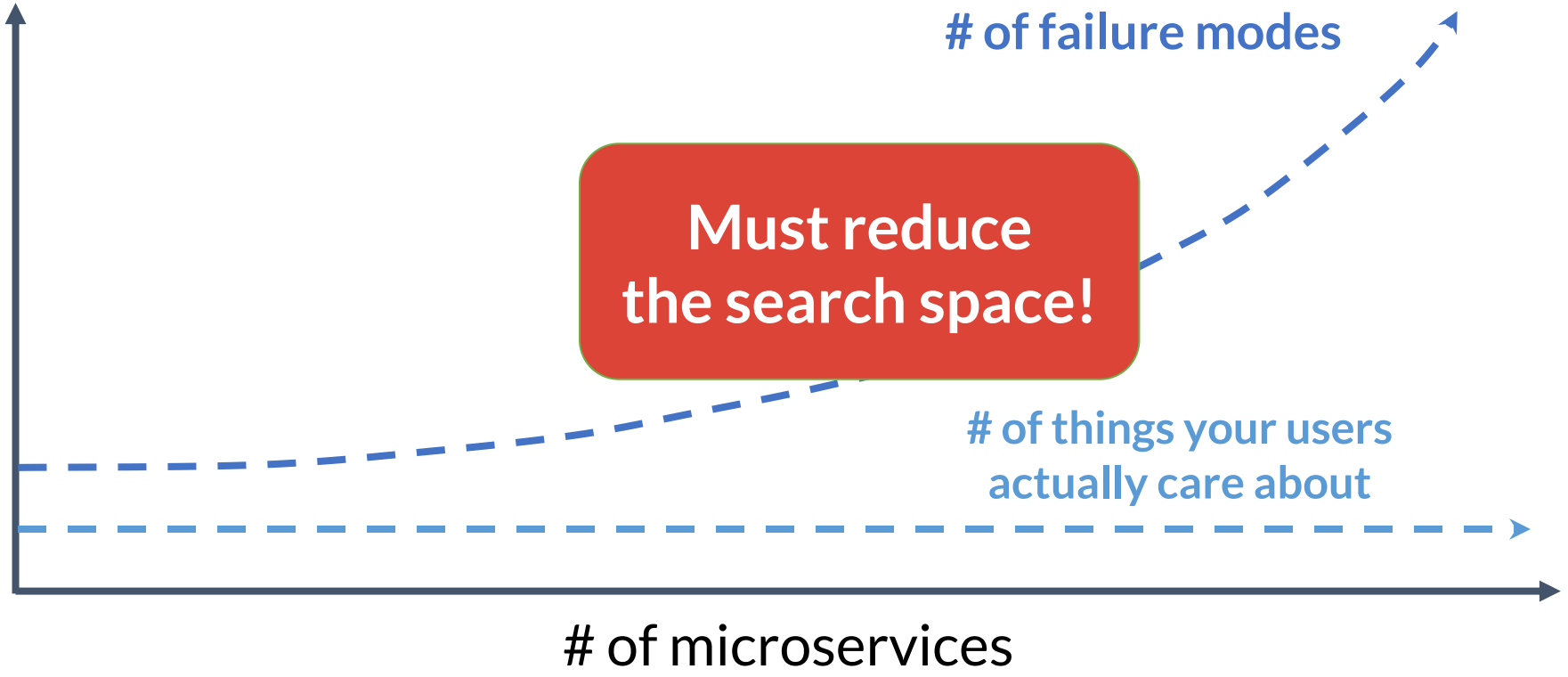
- Arbitrary dimensionality and cardinality
- Any layer of the stack, including mobile+web!

Fidelity:

- Correct stats!!!
- High stats frequency (i.e., “beware smoothing”!)

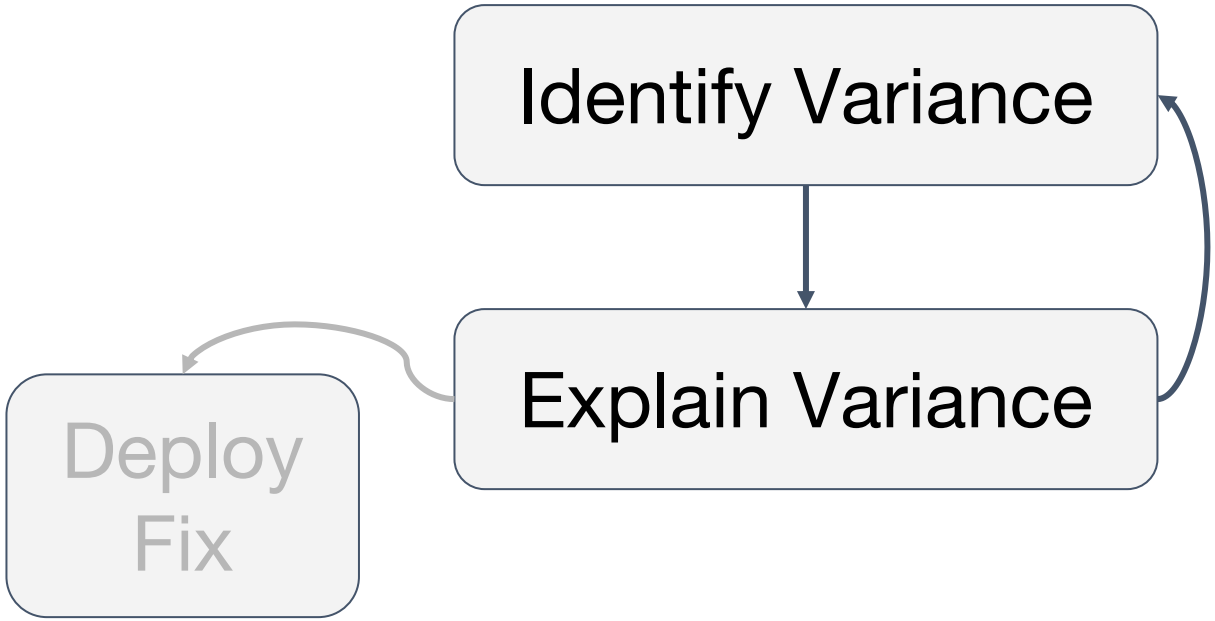
Freshness: ≤ 5 second lag

Scorecard >> **Refinement**

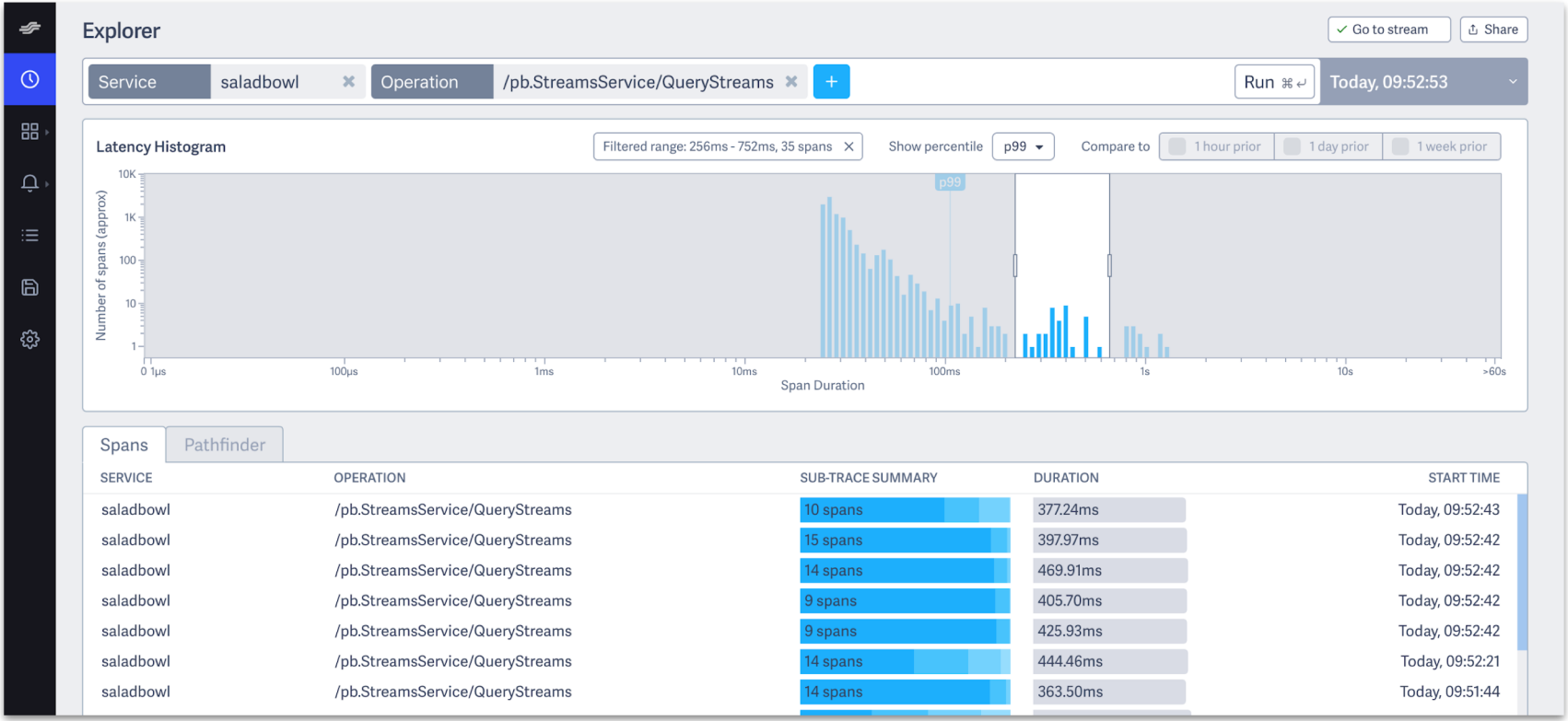


LIGHTSTEP

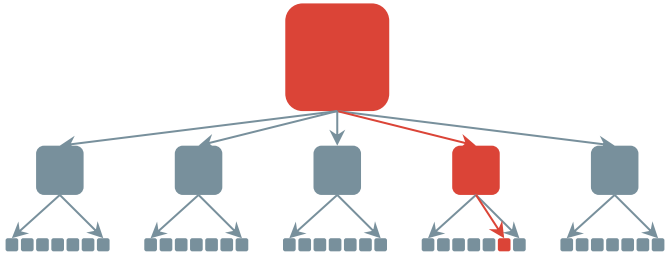
Scorecard >> **Refinement**



An interlude about variance and “p99”



Scorecard >> **Refinement**



Identifying Variance:

- Cardinality: understand which tag changed
- Robust stats: *histograms* (see prev slide)
- Data retention: always “Know What’s Normal”

Explaining variance:

- Correct stats!!!
- “Suppress the messengers” of microservice failures

Wrapping up…

(first, a hint at my perspective)

(Review)

The Life of Transaction Data: Dapper

Stage	Overhead affects…	Retained
Instrumentation Executed	App	100.00%
Buffered within app process	App	000.10%
Flushed out of process	App	000.10%
Centralized regionally	Regional network + storage	000.10%
Centralized globally	WAN + storage	000.01%

The Life of Transaction Data: ~~Dapper~~ LightStep

Stage	Overhead affects…	Retained
Instrumentation Executed	App	100.00%
Buffered within app process	App	100.00%
Flushed out of process	App	100.00%
Centralized regionally	Regional network + storage	100.00%
Centralized globally	WAN + storage	on-demand

An Observability Scorecard

Detection

- Specificity: unlimited cardinality, across the entire stack
- Fidelity: correct stats, high stats frequency
- Freshness: ≤ 5 seconds

Refinement

- Identifying variance: unlimited cardinality, hi-fi histograms, data retention
- “Suppress the messengers”

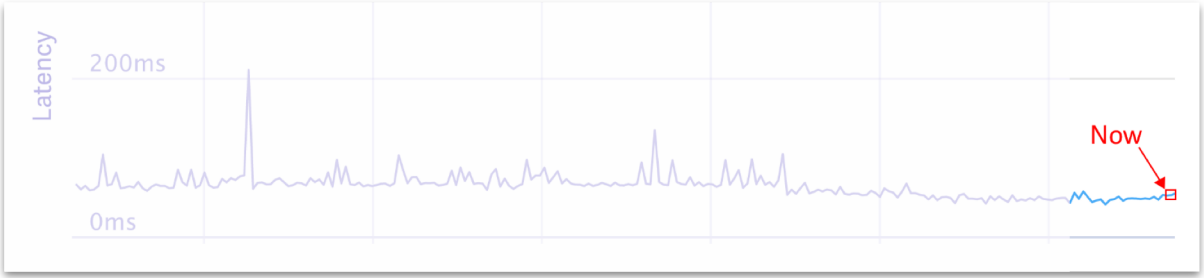
Thank you!

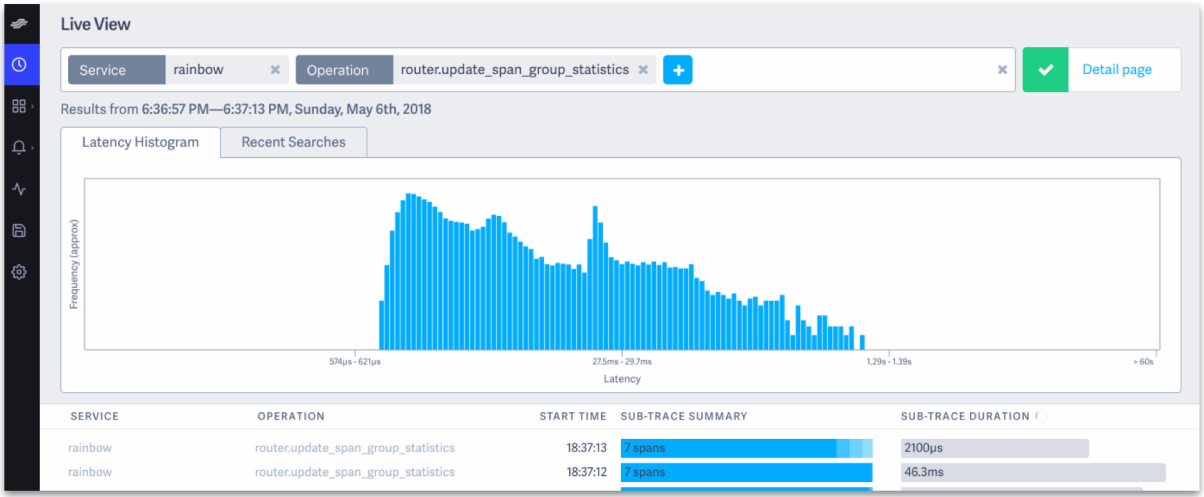
Ben Sigelman, Co-founder and CEO
twitter: @el_bhs
email: bhs@lightstep.com

LIGHTSTEP

Extra slides

Ideal Measurement: **Robust**





LIGHTSTEP

Ideal Measurement: High-Dimensional

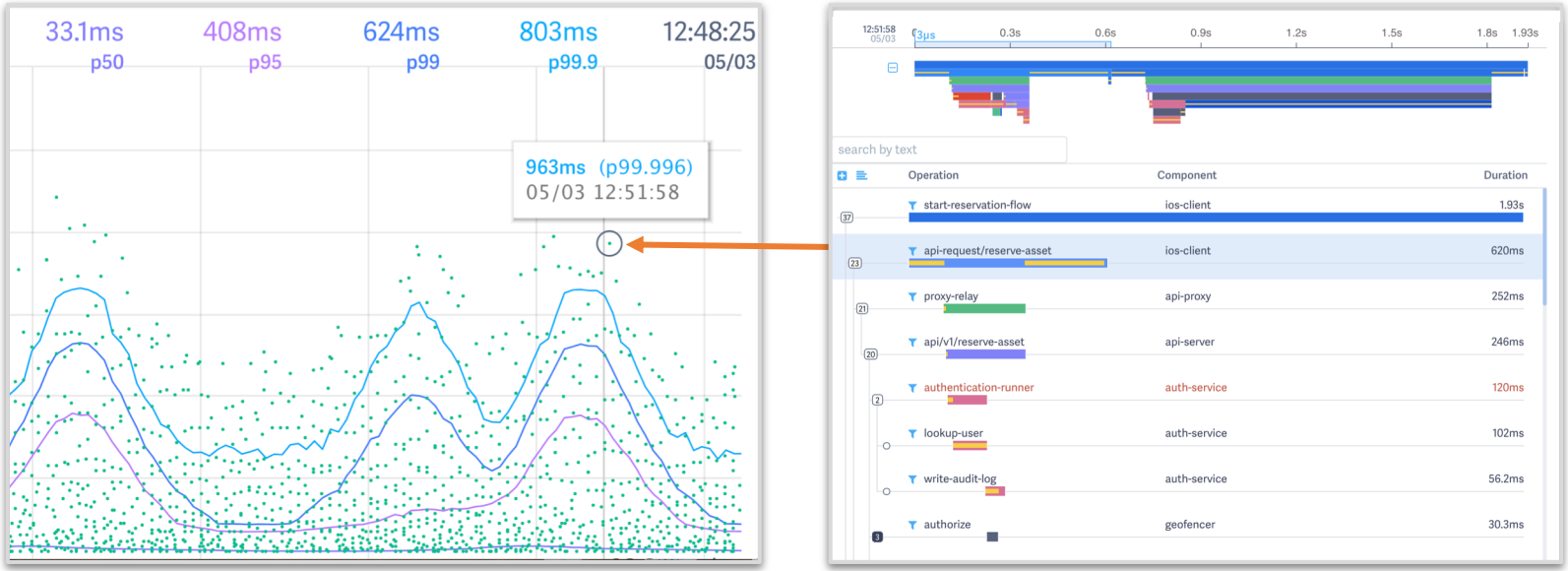
LIGHTSTEP

Ideal Refinement: **Real-time**

Must be able to **test and eliminate hypotheses** quickly

- Actual data must be ≤10s fresh
- UI / API latency must be very low

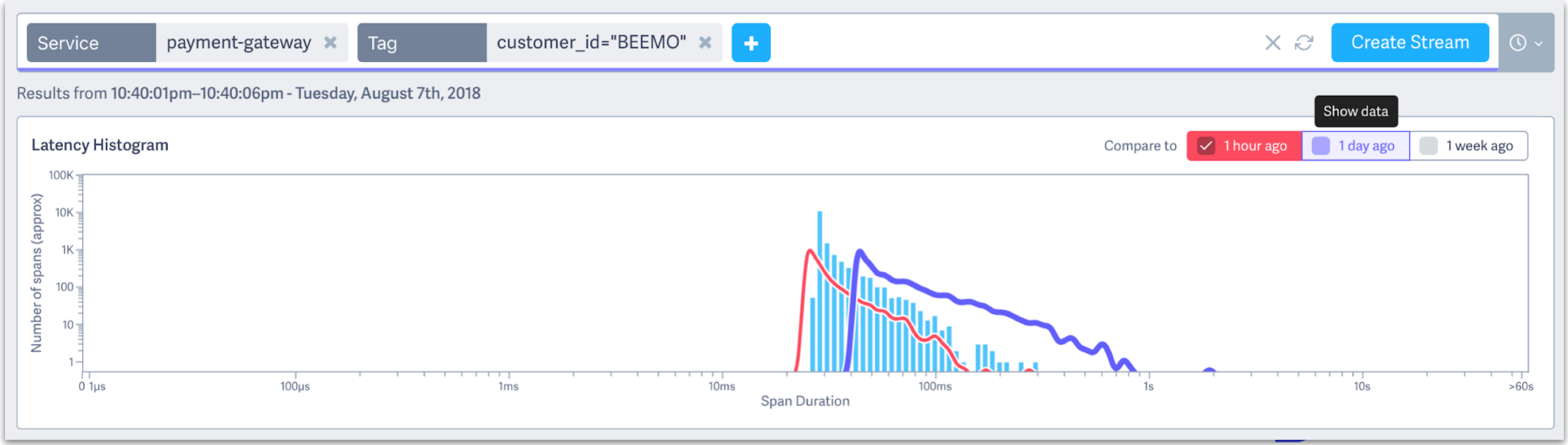
Ideal Refinement: **Global**



LIGHTSTEP

Ideal Refinement: Context-Rich

We can’t expect humans to **know what’s normal**



Thank you / Q&A