

Traces Are the Fuel, Not the Car

Making Distributed Tracing Valuable

March 4, 2019

Ben Sigelman, CEO and Co-founder, LightStep

Part I

Observability Dogma: 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.



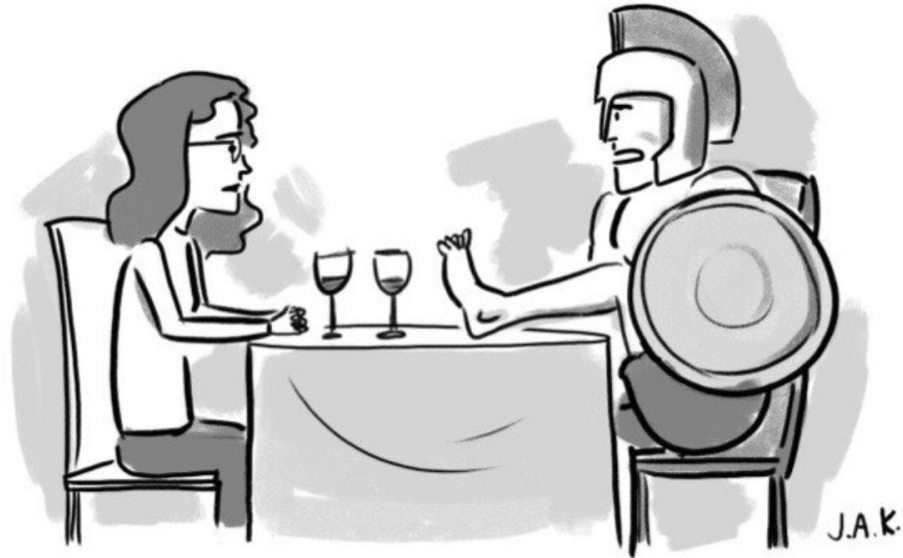
Metrics!

Traces!

Logs!

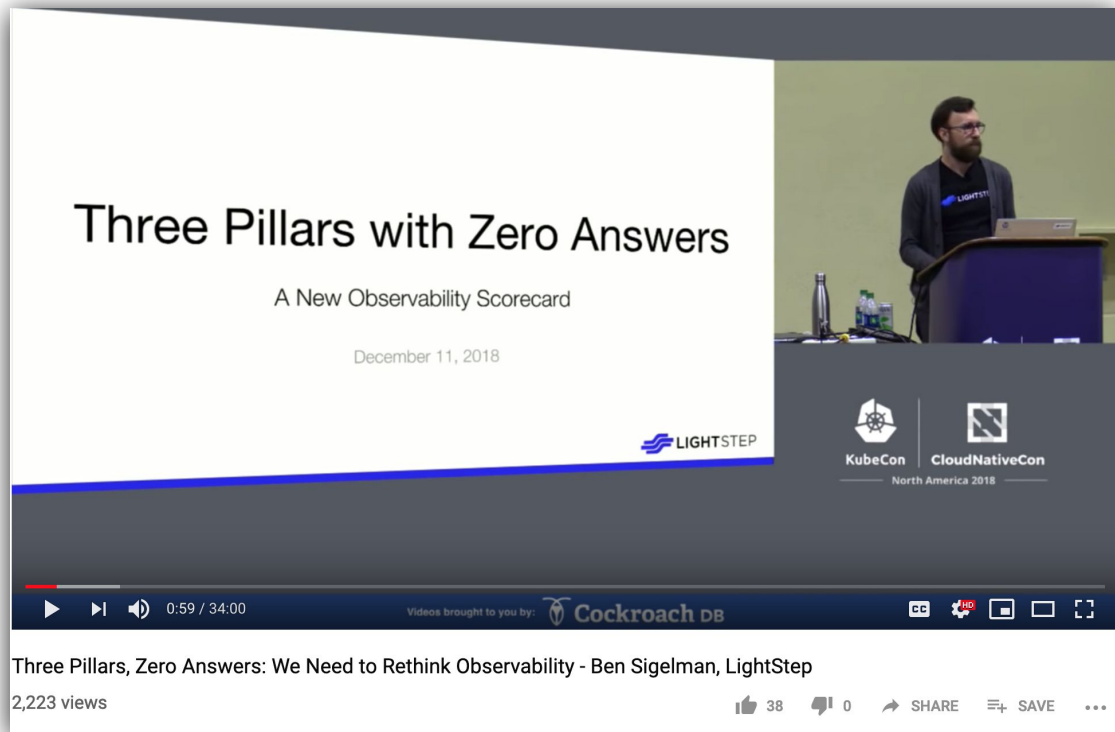
The Three Pillars of Observability

Fatal Flaws



"I'm ready to be vulnerable."

So Many Flaws, So Little Time...



The video player shows a presentation slide with the title "Three Pillars with Zero Answers" and subtitle "A New Observability Scorecard". The date "December 11, 2018" is displayed. The slide also features the LightStep logo. The video is from a conference, with logos for KubeCon and CloudNativeCon North America 2018 visible. The video player interface includes a progress bar at 0:59 / 34:00, a video description, and engagement metrics (38 likes, 0 dislikes).

Three Pillars with Zero Answers

A New Observability Scorecard

December 11, 2018

LIGHTSTEP

KubeCon CloudNativeCon
North America 2018

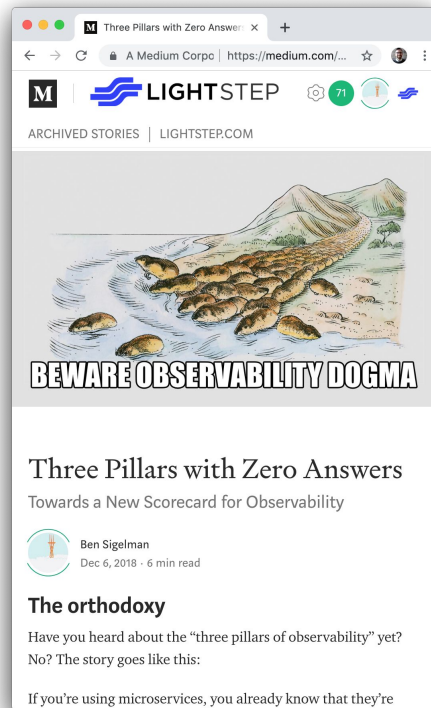
0:59 / 34:00

Videos brought to you by: Cockroach DB

Three Pillars, Zero Answers: We Need to Rethink Observability - Ben Sigelman, LightStep

2,223 views

38 0 SHARE SAVE



The Medium article page features a meme image of a dog on a riverbank with the text "BEWARE OBSERVABILITY DOGMA". The article title is "Three Pillars with Zero Answers" with the subtitle "Towards a New Scorecard for Observability". The author is Ben Sigelman, and the article was published on Dec 6, 2018. The first paragraph is titled "The orthodoxy" and discusses the "three pillars of observability".

Three Pillars with Zero Answers

Towards a New Scorecard for Observability

Ben Sigelman
Dec 6, 2018 · 6 min read

The orthodoxy

Have you heard about the "three pillars of observability" yet? No? The story goes like this:

If you're using microservices, you already know that they're

Fatal Flaws: “TL;DR” edition

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

A fun game!

Design your own (positive-ROI) observability system:

- ☐ High-throughput
- ☐ High-cardinality
- ☐ Unsampled
- ☐ Lengthy retention window

Choose three.

Metrics, Logs, and Traces are
Just Data,

... not a feature or use case.



Metrics!

Logs!

Traces!

The Three Pillars of Observability

The image shows three large, horizontal industrial pipes with flanged ends, arranged diagonally from the top-left to the bottom-right. The pipes are made of metal and show signs of rust and wear. The top-left pipe is labeled 'Metrics!', the top-right pipe is labeled 'Logs!', and the bottom pipe is labeled 'Traces!'. The background is a dark, industrial setting with some structural elements visible.

Metrics!

Logs!

Traces!

The Three Pillars Pipes of Observability

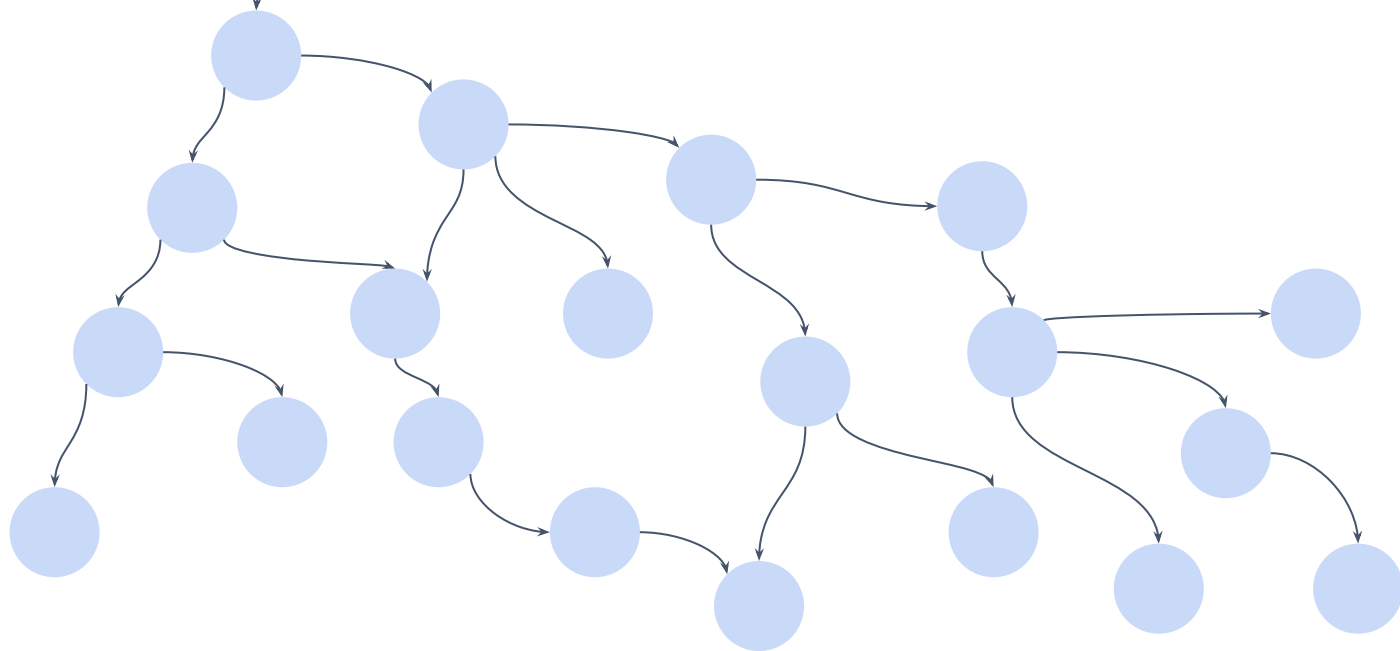
Part II

Service-Centric Observability

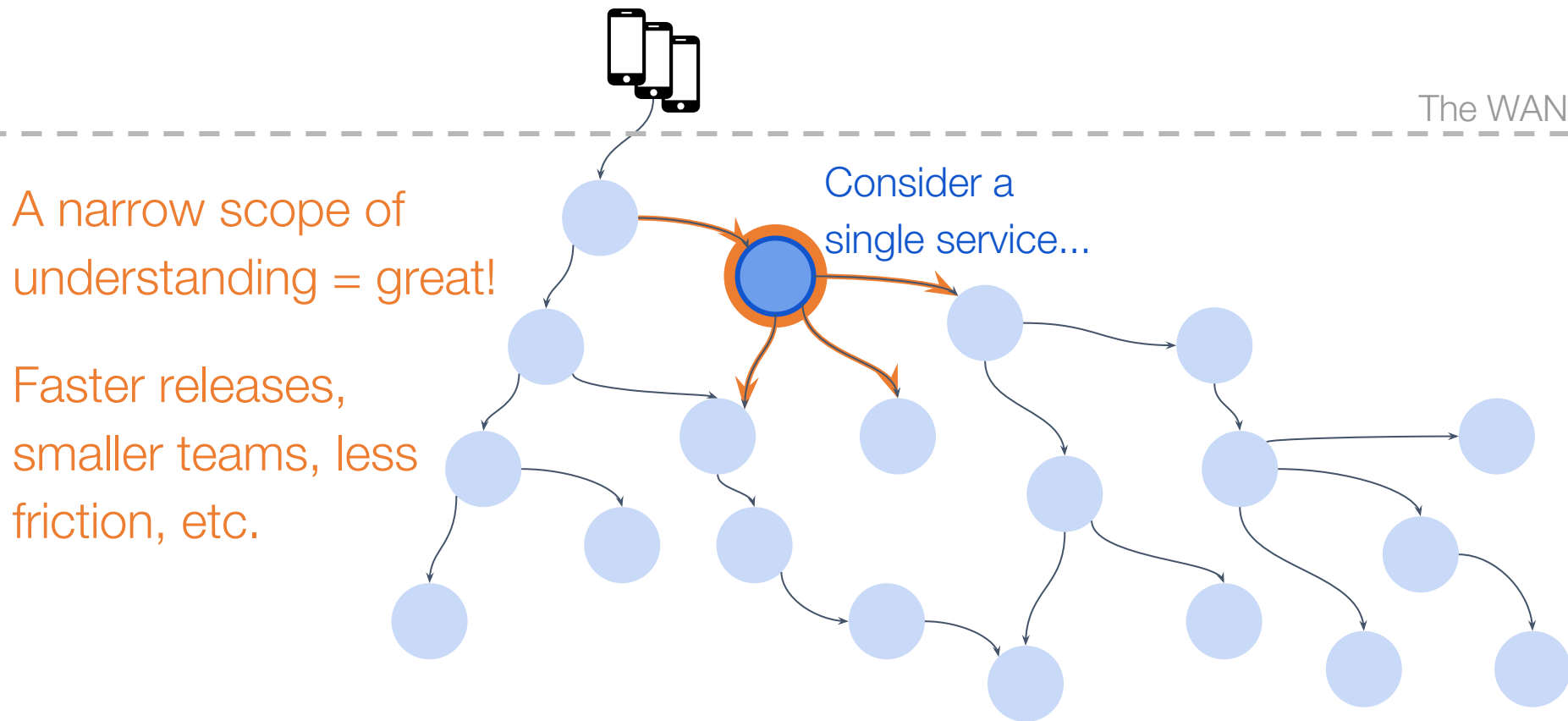
A microservices architecture



The WAN



A microservices architecture



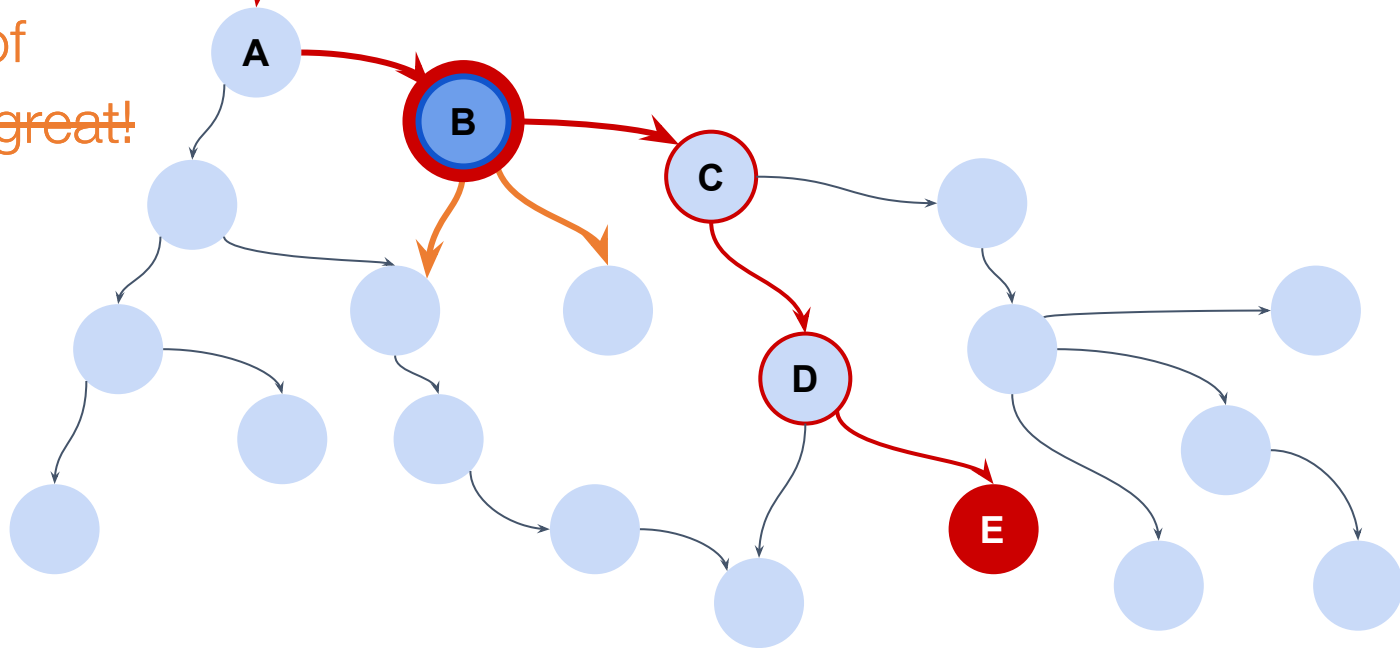
A microservices architecture (with a slowdown)



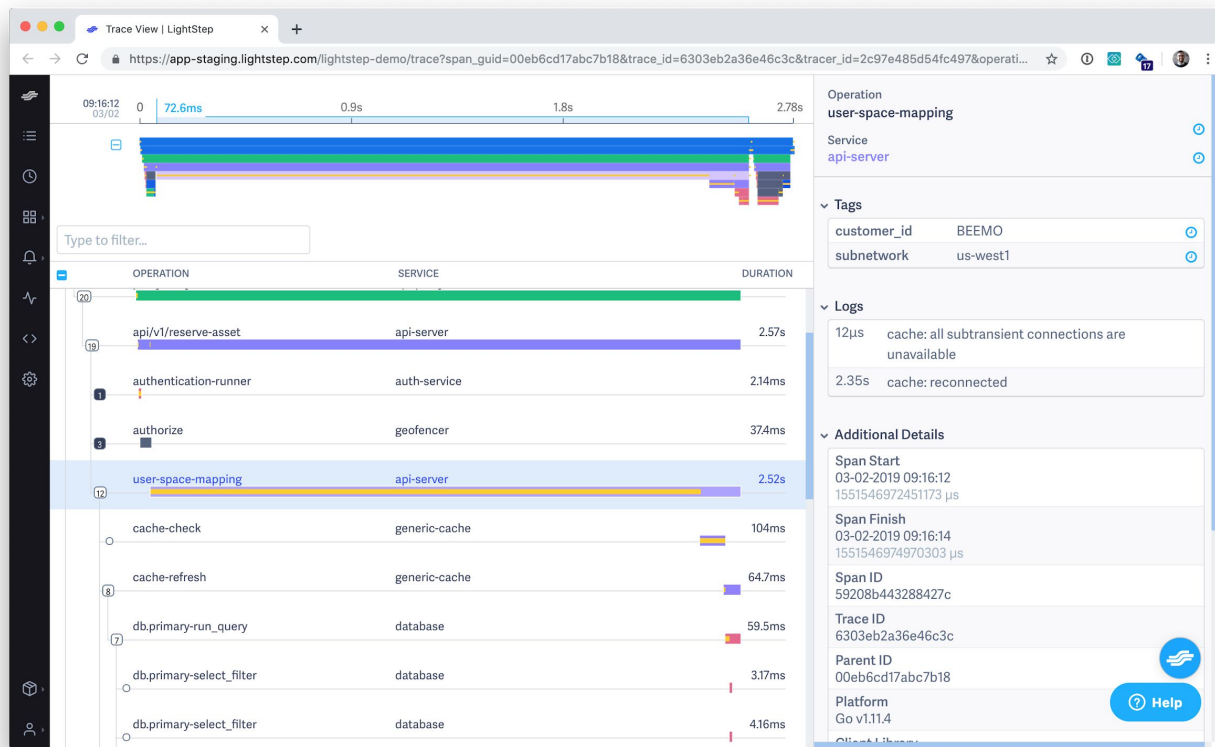
The WAN

A narrow scope of
understanding = great!
... problematic.

Decoupling cuts
both ways.



Hands-on with a single distributed trace



Distributed traces, in summary

- One distributed trace per transaction
- Crosses microservice boundaries
- They are **necessary** if we want to understand the relationships between distant actors in our architecture ...

... and yet:

- too numerous to centralize in “standard” ways
- too data-dense for our brains to process without help

“Distributed Tracing” != “Distributed Traces”

Distributed *traces*: basically just structs

Distributed *tracing*: **the art and science of making distributed traces valuable**

So... how *do* we make
distributed traces valuable?

Quick Vocab Refresher: **SLIs**

“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

Two Fundamental **Goals**

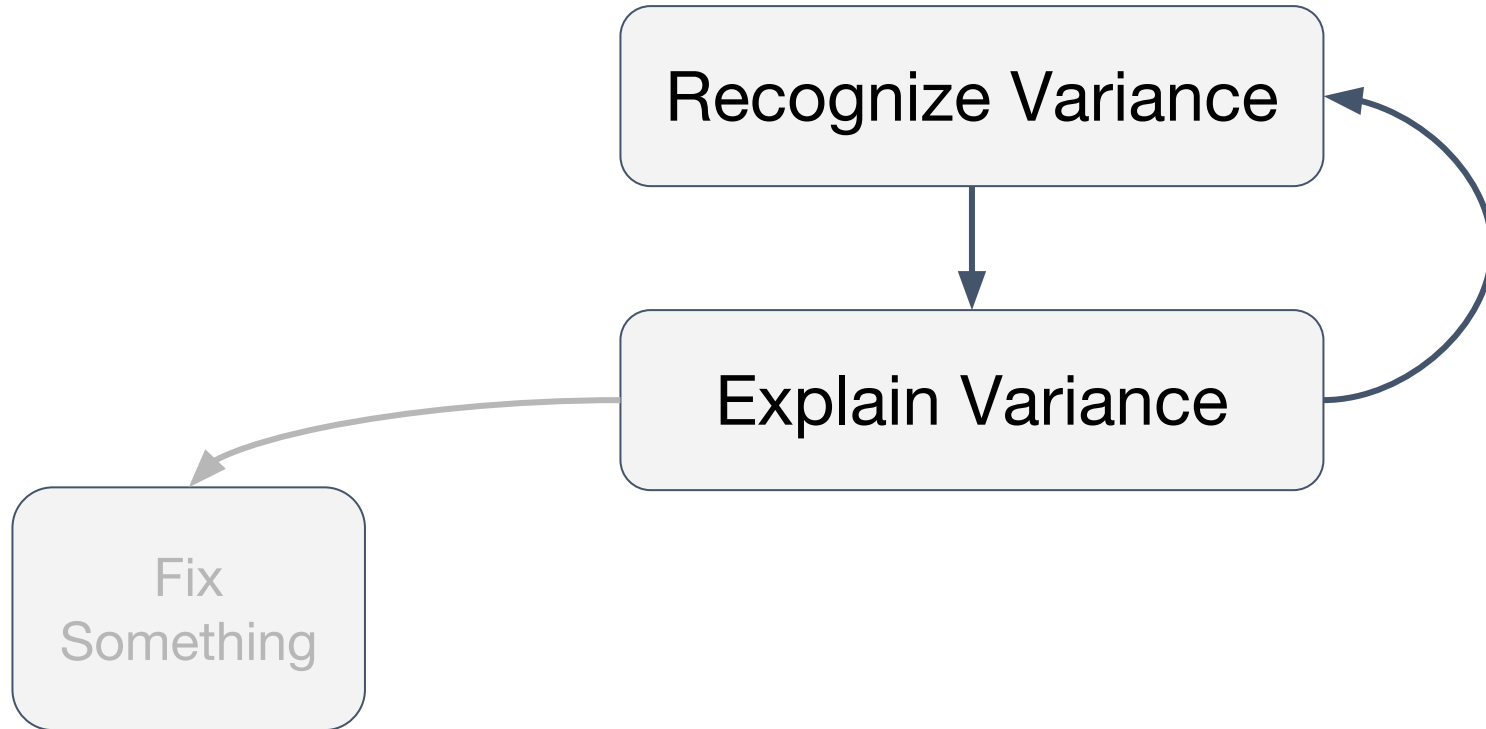
- Gradually **improving** an SLI
 - Rapidly **restoring** an SLI
- 
- days, weeks, months...
- NOW!!!!**

Reminder: “SLI” = “Service Level Indicator”

Two Fundamental **Activities**

1. **Detection:** measuring SLIs precisely
2. **Explaining variance:** recognizing and explaining variance, often iteratively

The Refinement Process



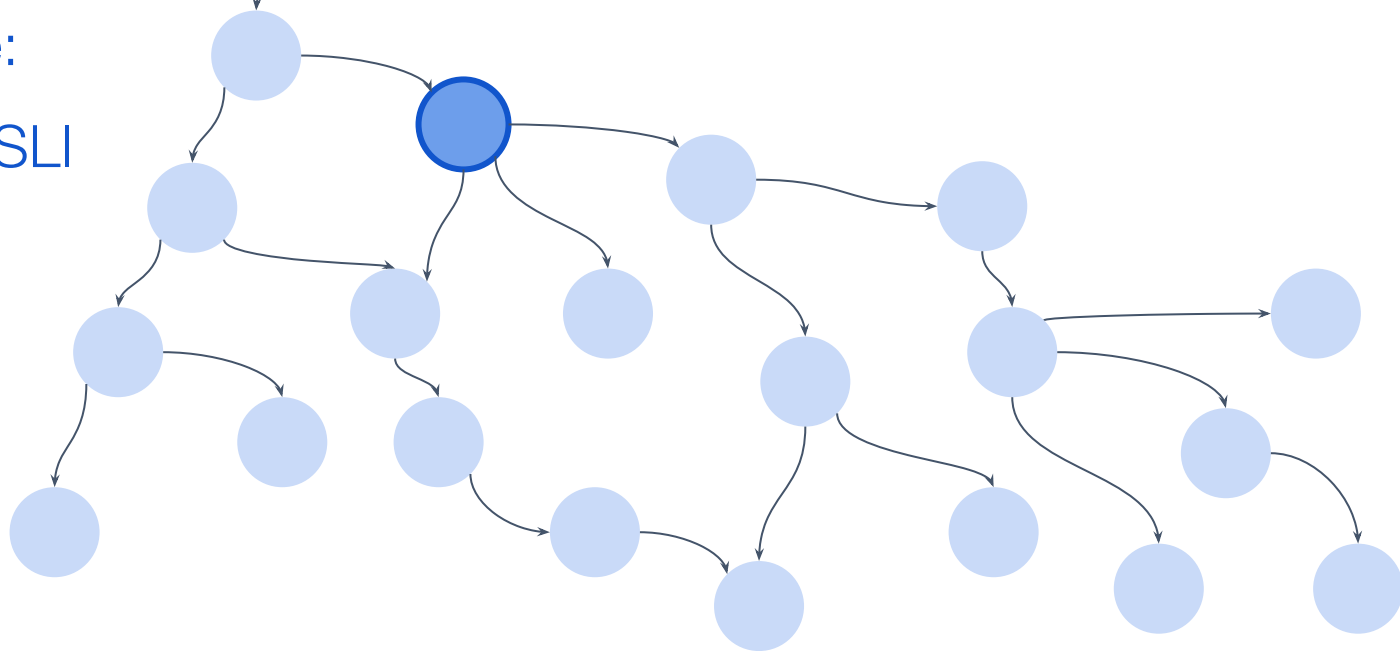
A Service-Centric Approach



The WAN

Given any service:

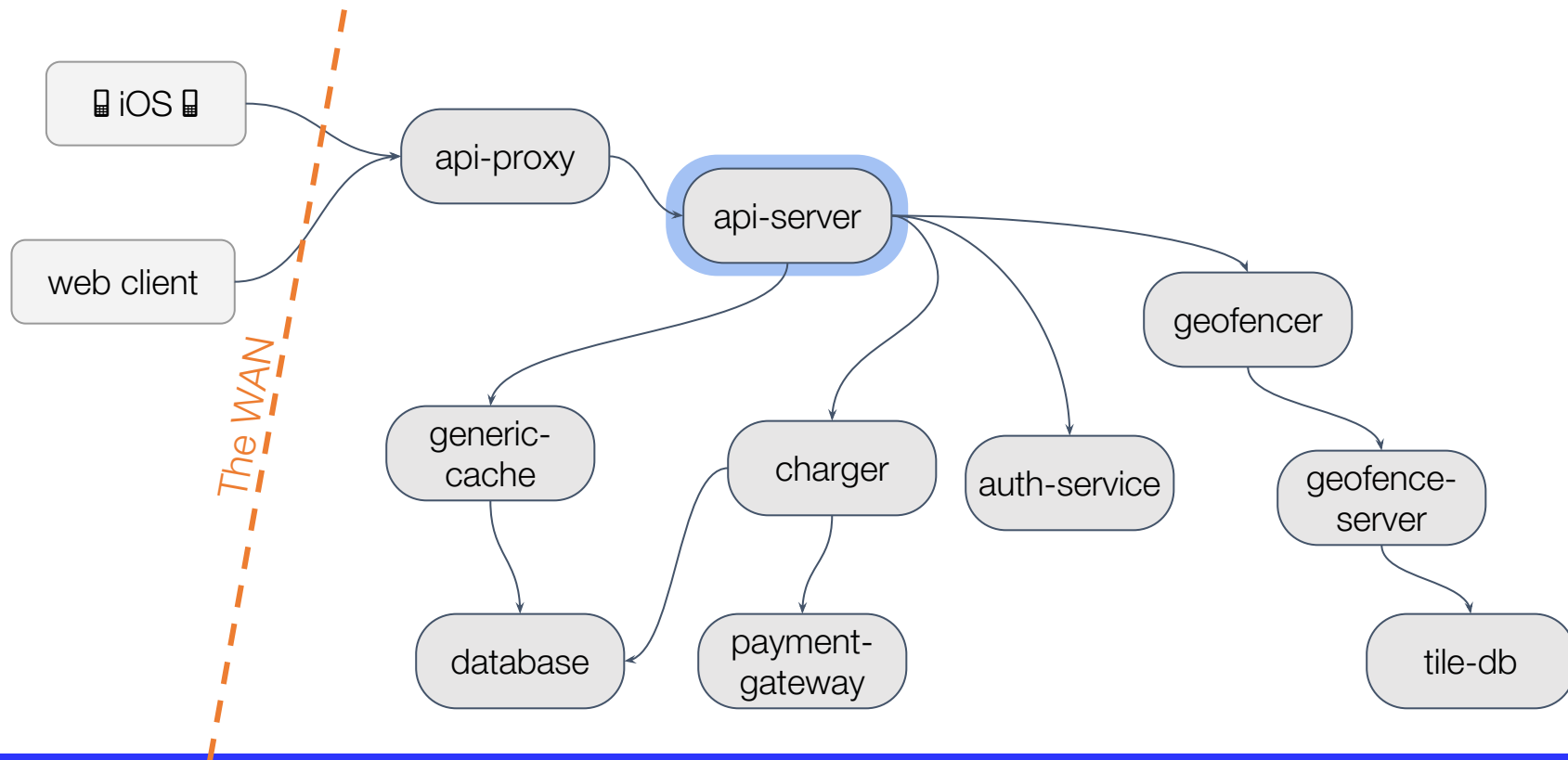
1. Start with an SLI
2. Find variance
3. Explain it



Part III

“Show & Tell”

A simple microservices architecture

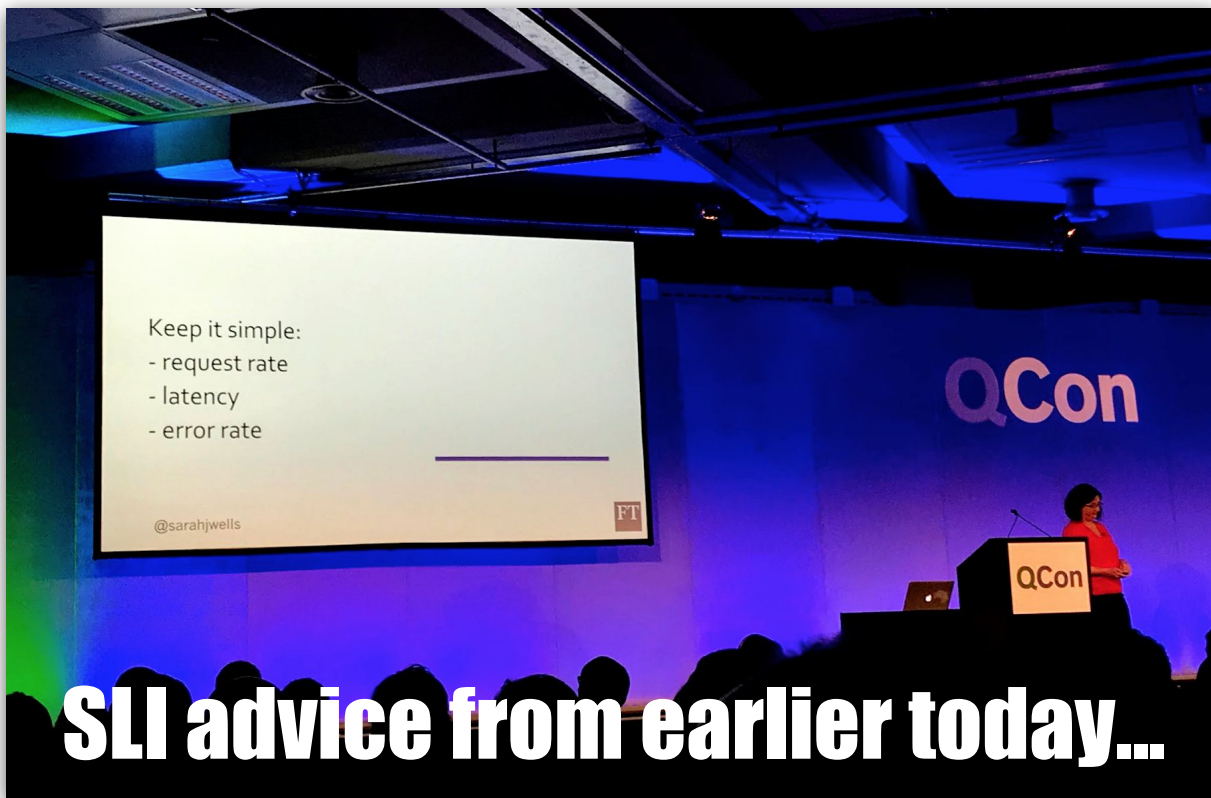


Recognizing Variance

1. Discovering SLIs (slide)
2. High-percentile latency measurement
3. “Performance is a shape” (and knowing what’s normal)
4. Examining individual traces

(link)

A blast from the past...

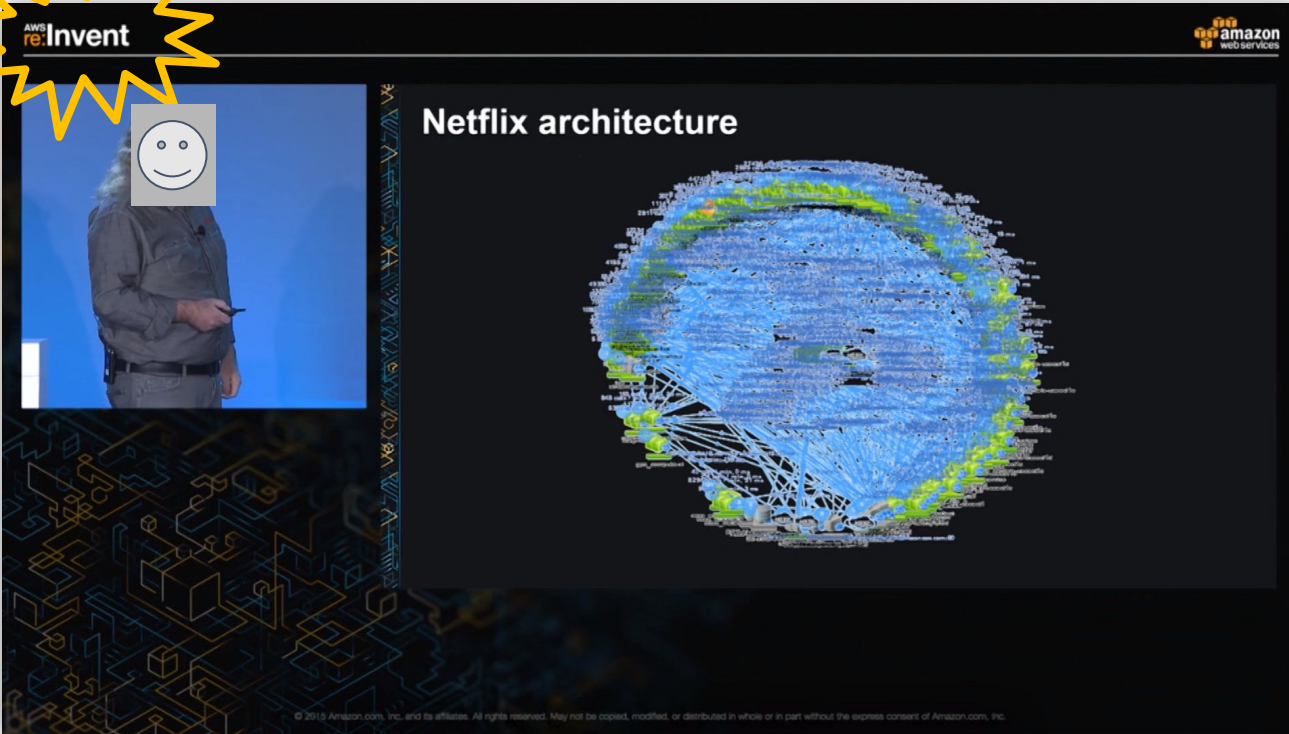


Service Diagrams

1. “Where’s Waldo” antipatterns (next slide)
2. Finding the common-case bottleneck
3. Finding the latency-outlier bottleneck

(link)

Service Diagrams and “Actionability”



The image is a screenshot of a presentation slide from AWS re:Invent. The slide features a dark background with a blue and yellow circuit-like pattern at the bottom. On the left, there is a video frame showing a person in a grey jacket, with a smiley face icon overlaid on their face. Above the video frame is a yellow starburst graphic. The top left corner of the slide has the 'AWS re:Invent' logo, and the top right corner has the 'amazon web services' logo. The main title of the slide is 'Netflix architecture'. Below the title is a large, complex network diagram representing the Netflix architecture, with numerous nodes and connections. At the bottom of the slide, there is a small copyright notice: '© 2015 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.'

aws re:Invent

amazon web services

Netflix architecture

© 2015 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.

Explaining Variance With Many Dimensions

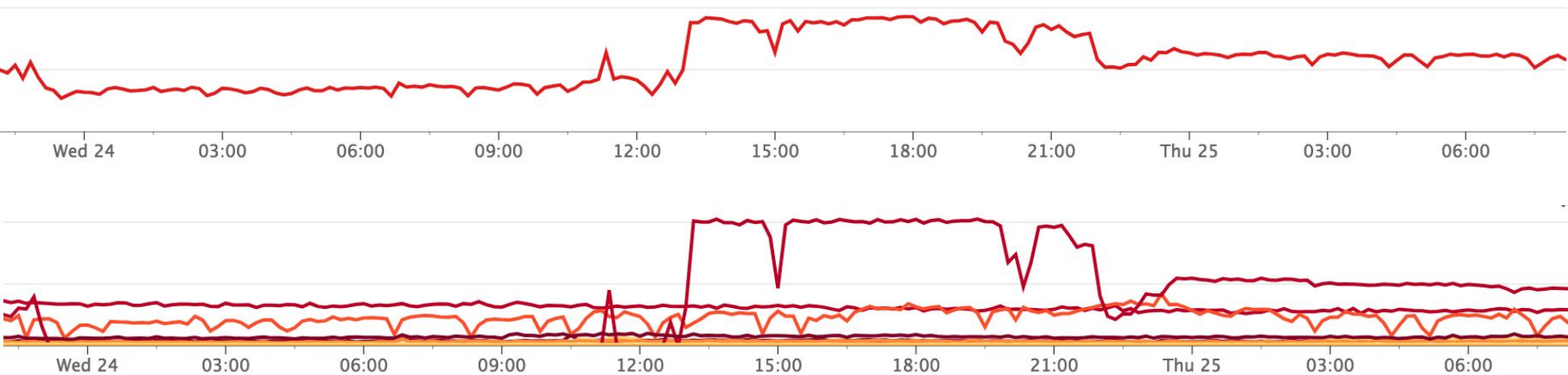
1. A “cardinality refresher” (next slide)
2. Exploring data with no cardinality limits
3. Explaining variance across the stack

(link)

A word nobody knew in 2015...

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

... but **cardinality**



Wrapping up...

What we've learned

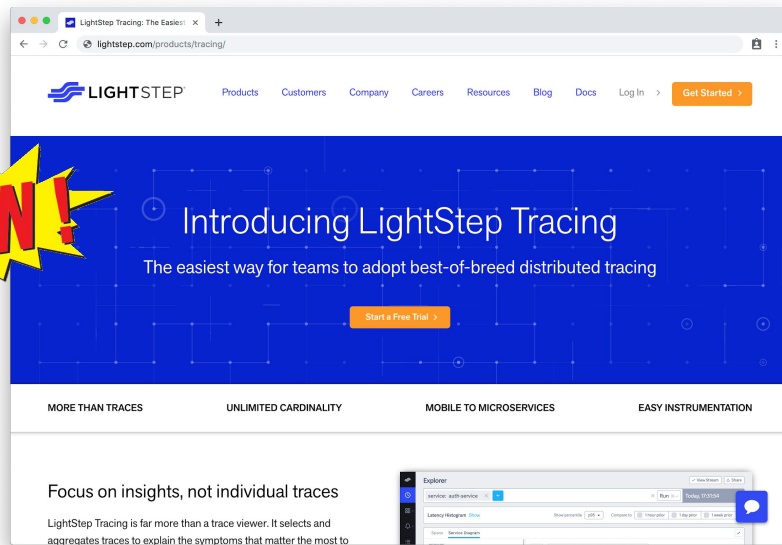
- Microservices helped us reduce human comms overhead
 - ... and that created huge problems for observability
- Distributed traces are *necessary but not sufficient*
- Distributed tracing is much more than distributed traces
- A service-centric approach with a modern, sophisticated distributed tracing system can do amazing things

Thank you!

Ben Sigelman, Co-founder and CEO
twitter: [@el_bhs](#)
email: bhs@lightstep.com

PS: LightStep announced something cool today! Stop by [Booth #3](#) to learn more.

I am friendly and would love to chat... please say hello, I don't make it to Europe often!



Extra slides