Driving Service Ownership with Distributed Tracing

Daniel “Spoons” Spoonhower, CTO and Co-founder

Lightstep

Who am I?

spoons (aka Daniel Spoonhower)

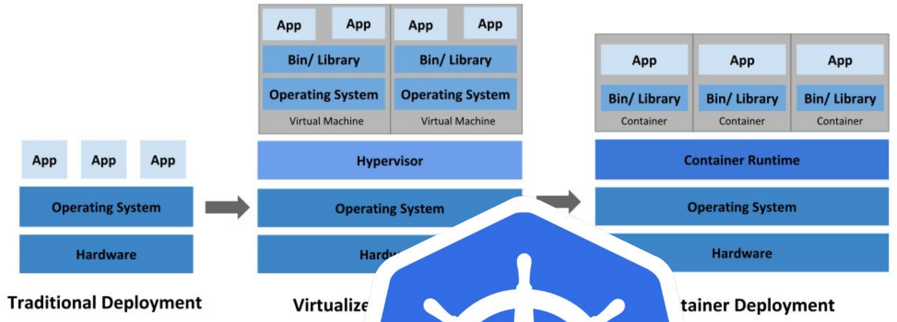
CTO and Co-founder, Lightstep

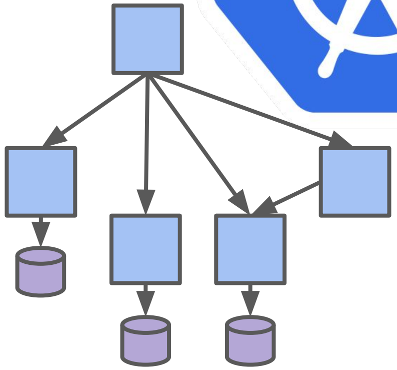
@save_spoons

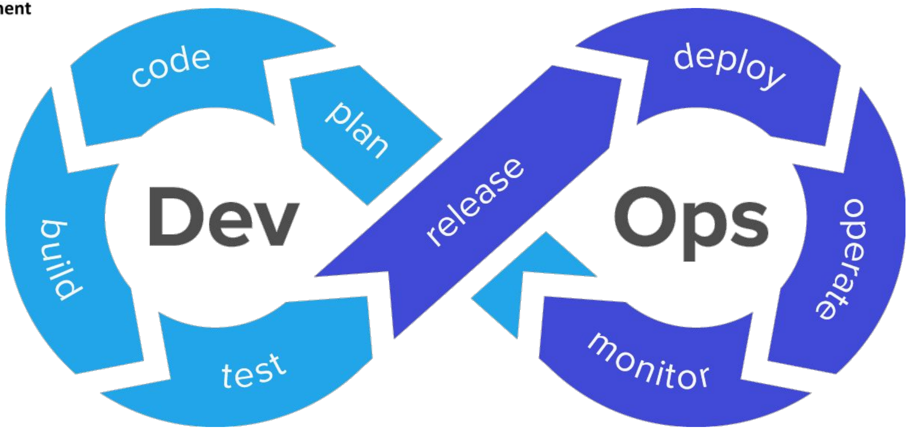
spoons@lightstep.com

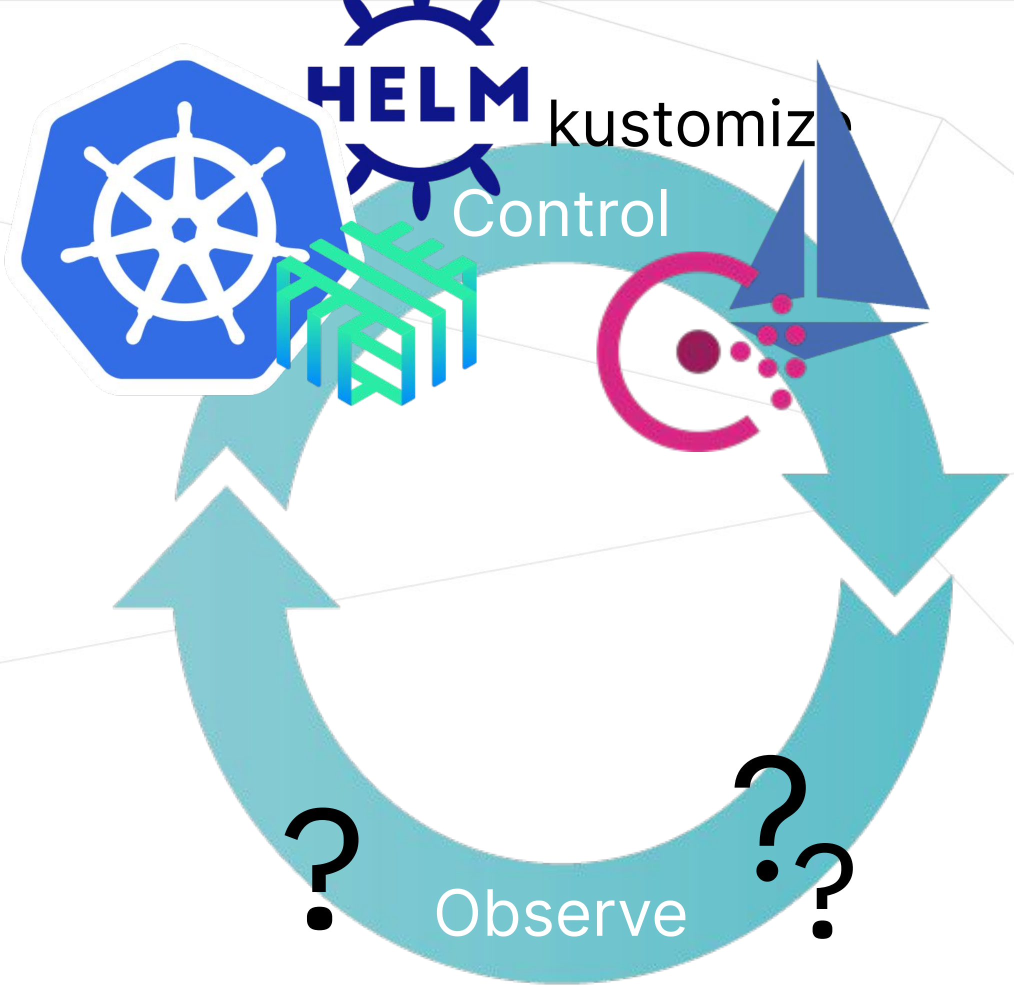
What Changed?

Cloud Native, Microservices & DevOps

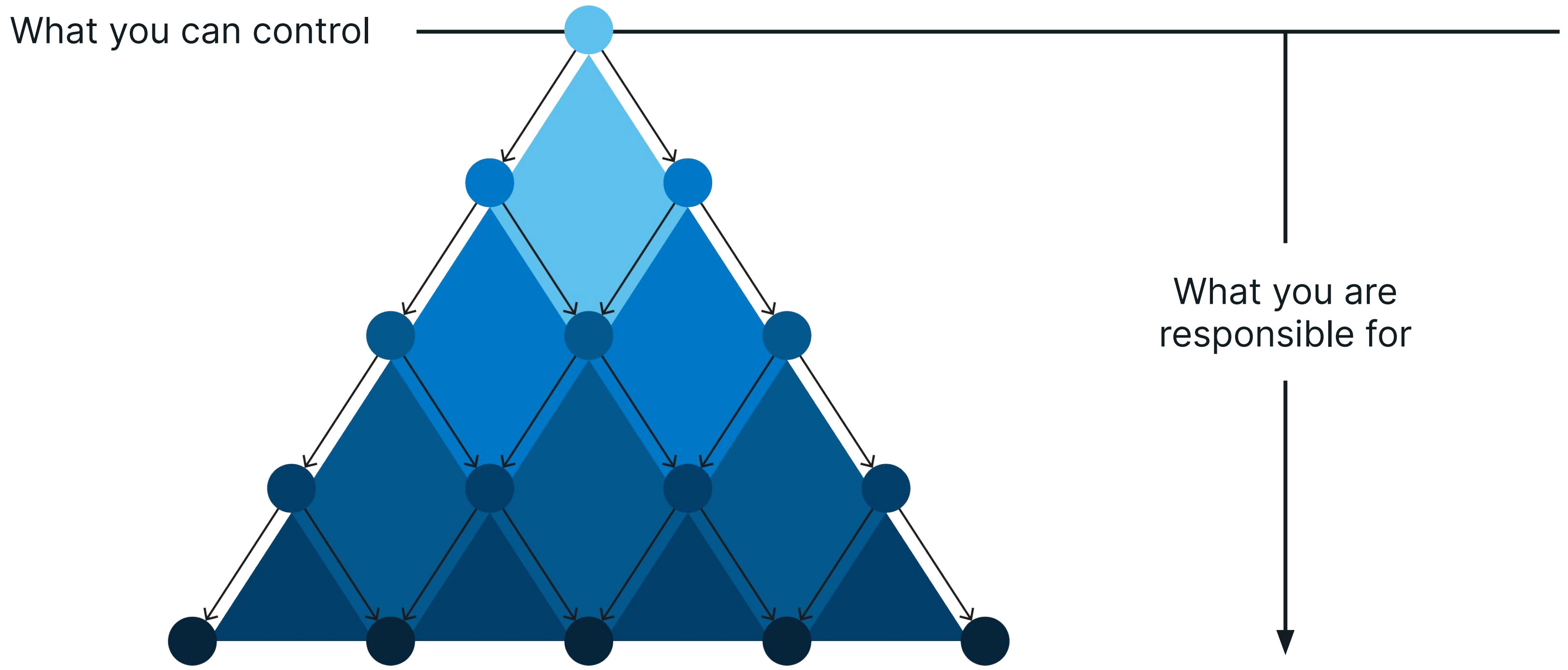

App
App
App
Operating System
Hardware
Traditional Deployment
App
App
App
App
Bin/ Library
Bin/ Library
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor
Operating System
Hard
Virtualize
App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware
tainer Deployment




code
plan
build
Dev
test
release
deploy
Ops
operate
monitor


HELM
kustomize
Control
?
Observe
?
?

Stress (n): responsibility without control



Closing the gap between control and responsibility

Responsibility for delivering **performance** and **reliability**

Like many problems, the solution requires:

- Having the right data
- Setting the right goals
- Giving teams ownership

Distributed tracing is essential to closing this gap

Ownership means…

Accountability

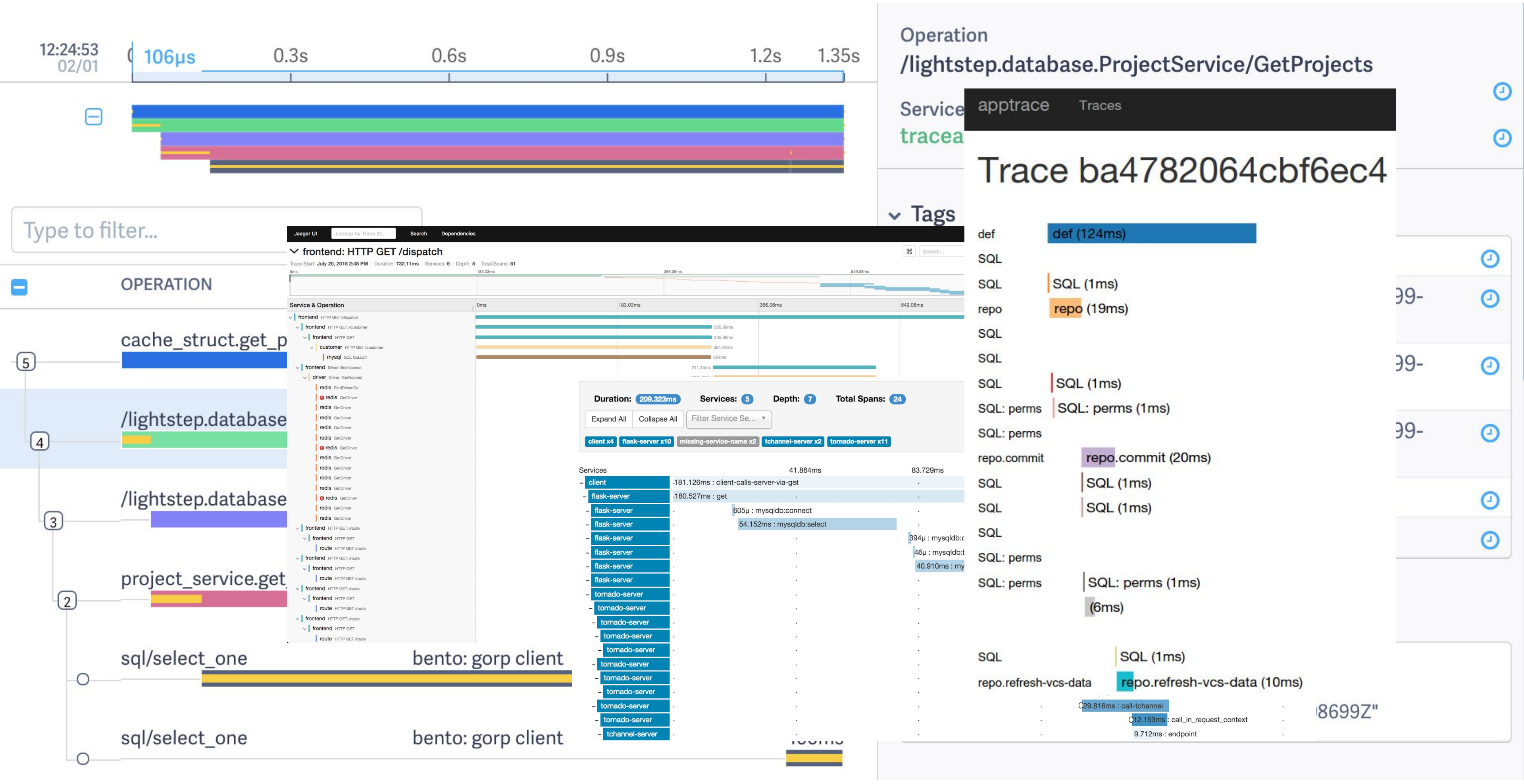
+

Agency

Loosely coupled services require…

Distributed Tracing

Distributed Tracing



12:24:53
02/01

106μs 0.3s 0.6s 0.9s 1.2s 1.35s

Type to filter...

OPERATION

5 cache_struct.get_p

4 /lightstep.database

3 /lightstep.database

2 project_service.get

sql/select_one bento: gorp client

sql/select_one bento: gorp client

Operation
/lightstep.database.ProjectService/GetProjects

Service
tracea

Tags

99-
99-
99-
8699Z"

Jaeger UI | Lookup by Trace ID... | Search | Dependencies

frontend: HTTP GET /dispatch

Trace Start: **July 20, 2018 2:48 PM** | Duration: **732.11ms** | Services: **6** | Depth: **5** | Total Spans: **51**

0ms 183.03ms 366.06ms 549.08ms

Service & Operation	
frontend HTTP GET /dispatch	
frontend HTTP GET: /customer	305.89ms
frontend HTTP GET	305.88ms
customer HTTP GET /customer	304.46ms
mysql SQL SELECT	304ms
frontend Driver::findNearest	211.15ms
driver Driver::findNearest	
redis FindDriverIDs	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
redis GetDriver	
frontend HTTP GET: /route	
frontend HTTP GET	
route HTTP GET /route	
frontend HTTP GET: /route	
frontend HTTP GET	
route HTTP GET /route	
frontend HTTP GET: /route	
frontend HTTP GET	
route HTTP GET /route	
frontend HTTP GET: /route	
frontend HTTP GET	
route HTTP GET /route	

Duration: 209.323ms **Services:** 5 **Depth:** 7 **Total Spans:** 24

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
client	181.126ms : client-calls-server-via-get	
flask-server	180.527ms : get	
flask-server	605μ : mysqldb:connect	
flask-server	54.152ms : mysqldb:select	
flask-server		394μ : mysqldb:c
flask-server		46μ : mysqldb:t
flask-server		40.910ms : my
flask-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tornado-server		
tchannel-server		

29.816ms : call-tchannel
12.153ms : call_in_request_context
9.712ms : endpoint

apptrace Traces

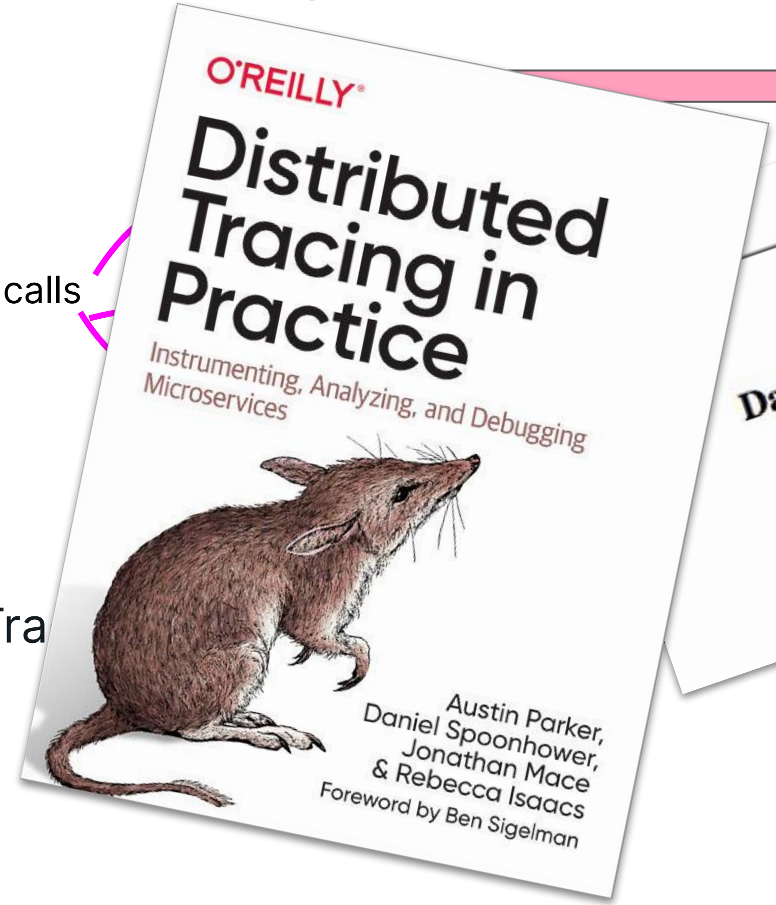
Trace ba4782064cbf6ec4

def	def (124ms)
SQL	
SQL	SQL (1ms)
repo	repo (19ms)
SQL	
SQL	
SQL	SQL (1ms)
SQL: perms	SQL: perms (1ms)
SQL: perms	
repo.commit	repo.commit (20ms)
SQL	SQL (1ms)
SQL	SQL (1ms)
SQL	
SQL: perms	
SQL: perms	SQL: perms (1ms)
	(6ms)
SQL	SQL (1ms)
repo.refresh-vcs-data	repo.refresh-vcs-data (10ms)

Relationships matter

calls

Tra ween callers and callees

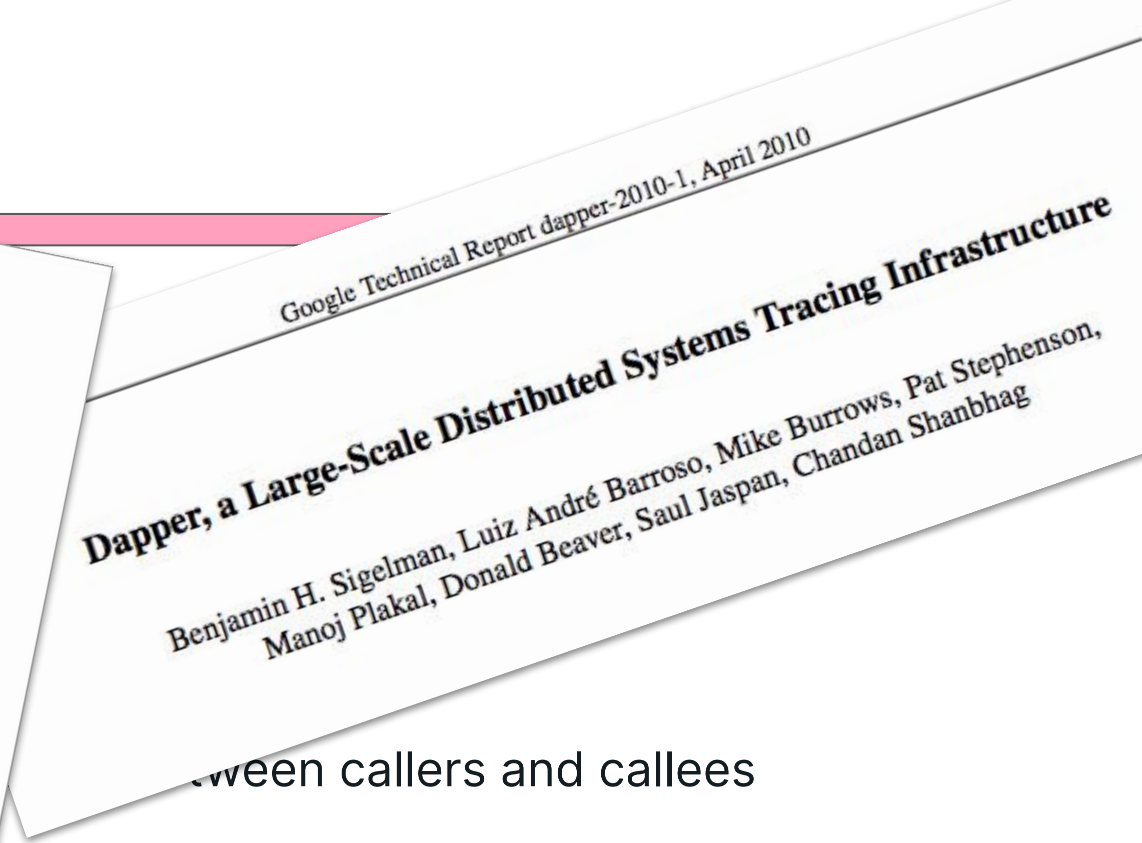

O'REILLY

Distributed Tracing in Practice

Instrumenting, Analyzing, and Debugging Microservices

Austin Parker, Daniel Spoonhower, Jonathan Mace & Rebecca Isaacs

Foreword by Ben Sigelman


Google Technical Report dapper-2010-1, April 2010

Dapper, a Large-Scale Distributed Systems Tracing Infrastructure

Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, Chandan Shanbhag

Traces are the raw material, not the finished product

Distributed *traces* – basically just structs

Distributed *tracing* – **the art and science of deriving value from traces**

Service Ownership

Benefits & Risks

Service ownership, defined

Ownership means teams are responsible for delivery of their software and services

Includes responsibilities like:

- Incident response
- Cost management
- Fixing bugs

Compare to DevOps...

- As a engineering culture
- As a set of tools
- As a feedback loop between developers and their users

Benefits

Team are more independent ⟹ higher developer velocity

Engineering team performance tied to real business metrics

- Application developers
- Platform engineers

Risks

Teams are more independent ⟹ more frameworks & tools

- Higher vendor costs
- More training new team members & internal transfers
- Harder to get a “big picture” view of your application

Managing trade-offs around benefits and risks

Allowing for independence

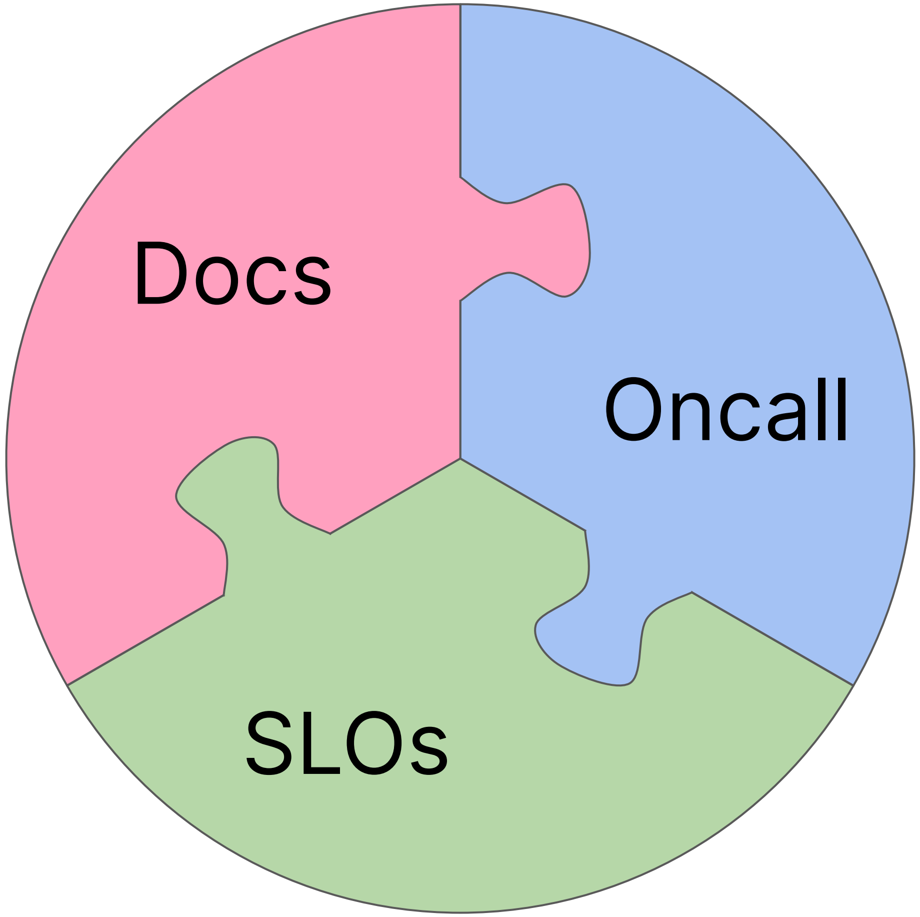
Define clear responsibilities and goals

...while ensuring consistency

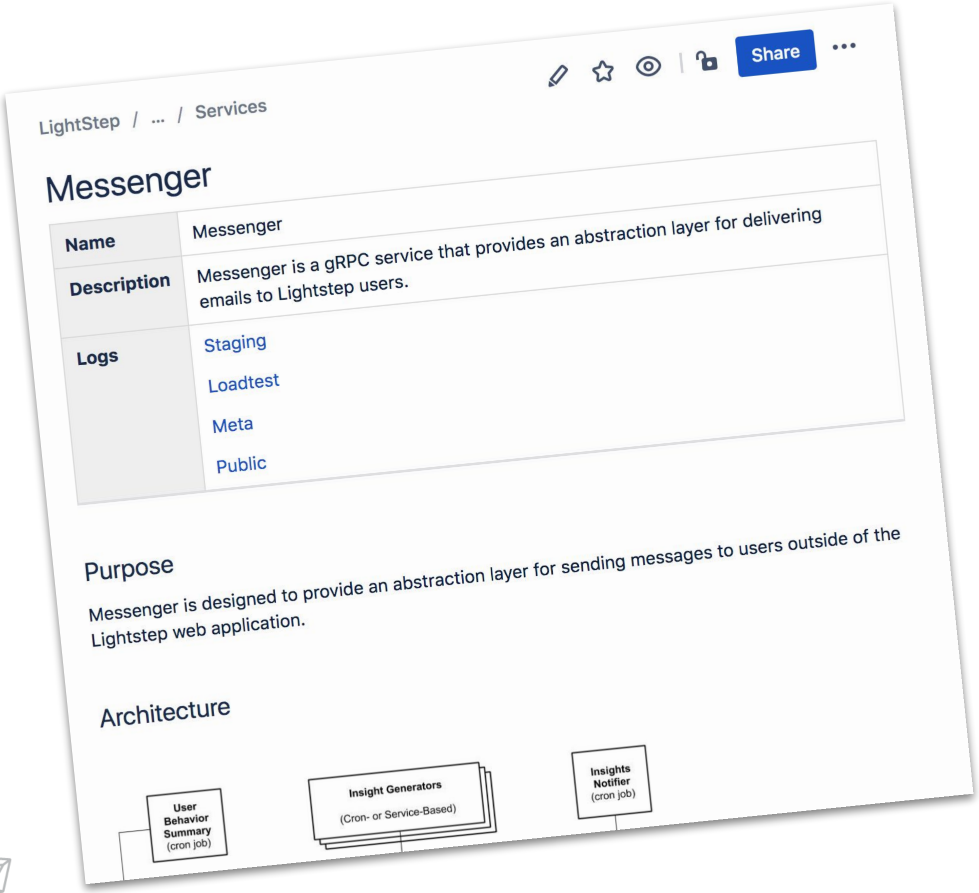
Measure progress toward goals and hold teams accountable

Driving Toward Ownership

3 pieces to the puzzle



Centralized documentation



LightStep / ... / Services

Messenger

Name	Messenger
Description	Messenger is a gRPC service that provides an abstraction layer for delivering emails to Lightstep users.
Logs	Staging Loadtest Meta Public

Purpose

Messenger is designed to provide an abstraction layer for sending messages to users outside of the Lightstep web application.

Architecture

Share
User Behavior Summary (cron job)
Insight Generators
(Cron- or Service-Based)
Insights Notifier (cron job)

Start with *expertise* then *ownership*

Make it easy to find related...

- Telemetry & dashboards
- Alert definitions
- Playbooks

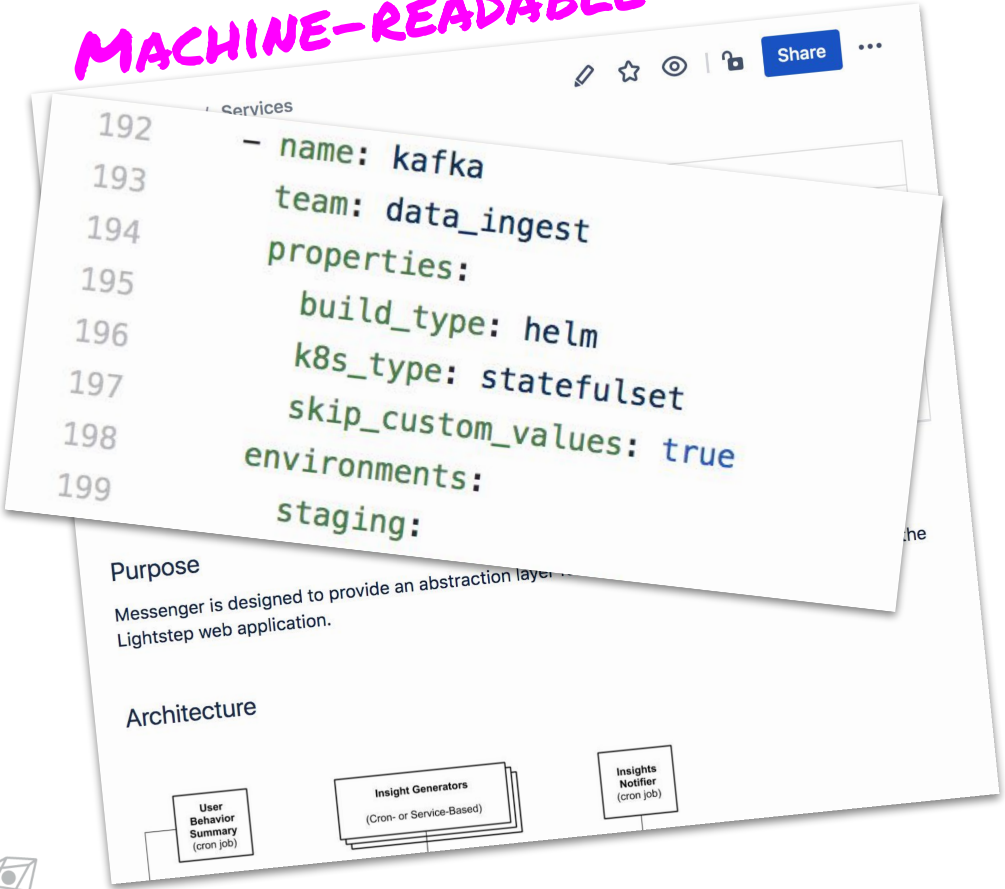
Use a template!

Track last-modified dates

- Require periodic audits & updates

~~Centralized~~ documentation

MACHINE-READABLE


Share
Services
Purpose
Messenger is designed to provide an abstraction layer
Lightstep web application.
Architecture
User Behavior Summary (cron job)
Insight Generators (Cron- or Service-Based)
Insights Notifier (cron job)

```yaml
- name: kafka
  team: data_ingest
  properties:
    build_type: helm
    k8s_type: statefulset
    skip_custom_values: true
  environments:
    staging:
```

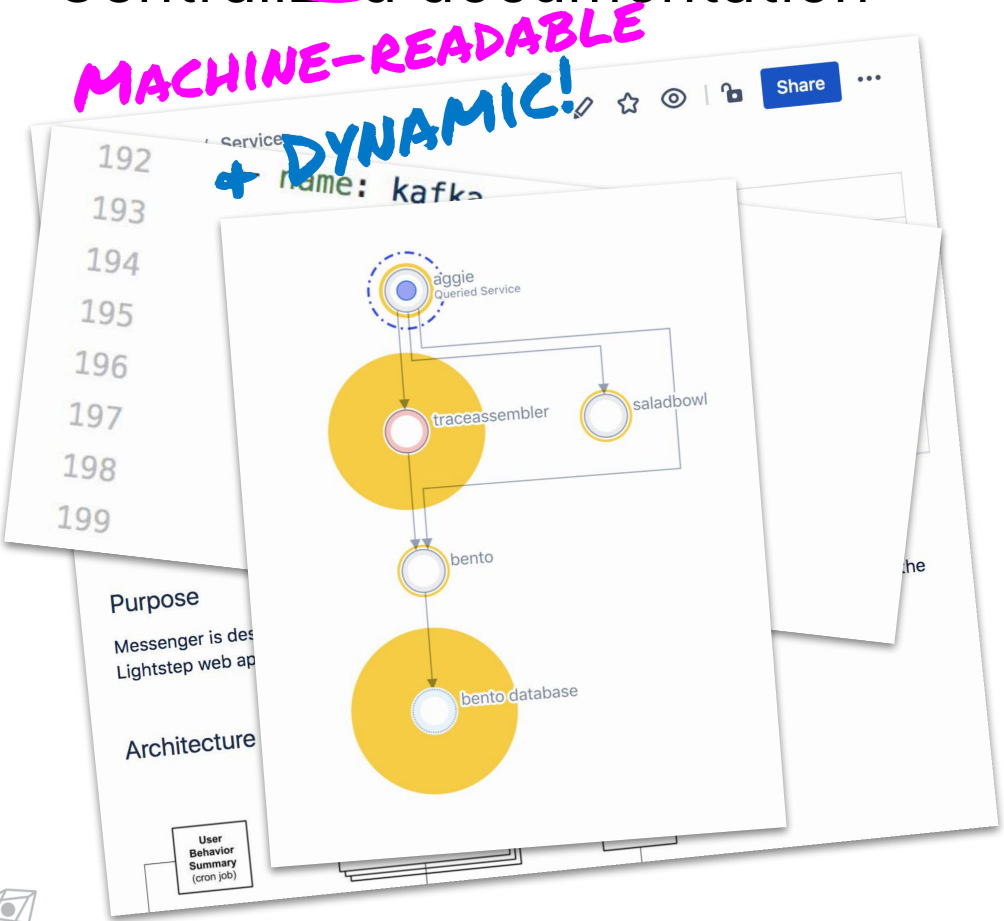
Make documentation *machine-readable*

Use it to generate:

- Dashboard config
- Escalation policy config
- Deployment pipeline config
- …

Make documentation *necessary* for day-to-day work

~~Centralized~~ documentation



It’s hard to keep service dependencies up to date *manually*…

So don’t!

- Use telemetry from the application

Why is documentation important?

Record who is accountable

Automate many mundane tasks

Train new team members

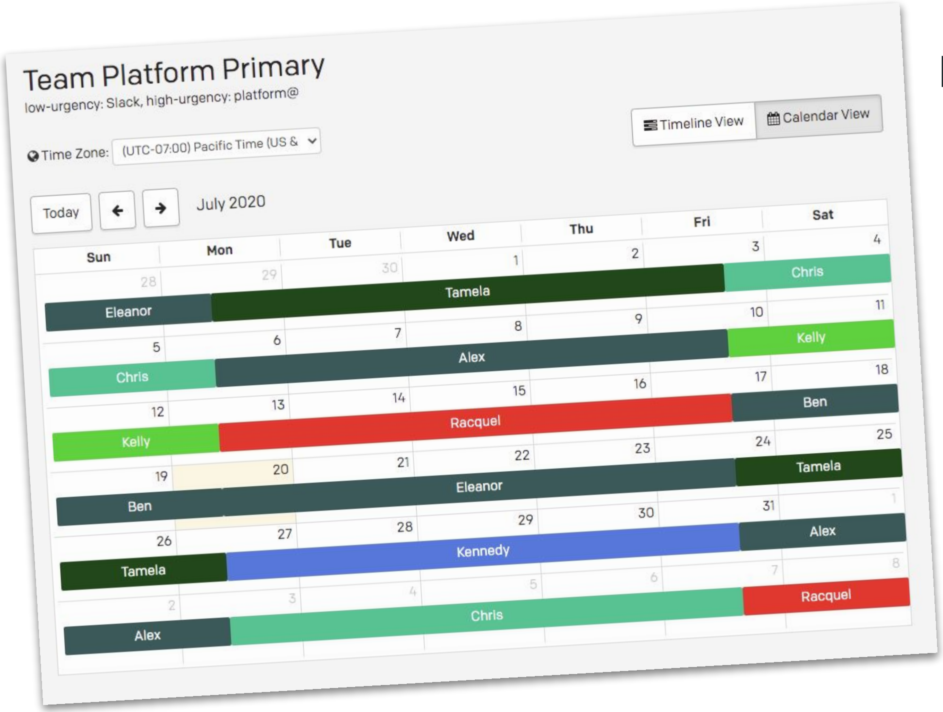
Build confidence

Oncall

Oncall is (often) responsible for...

- Incident response
- Communicating status internally & externally
- Production change management
  - Deploying new code
  - Pushing infrastructure changes
- Monitoring dashboards
- Low-urgency alert triage
- Customer requests
  - And other interrupt driven work
- Shift handoffs
- Writing postmortems

Handling alerts



Team Platform Primary

low-urgency: Slack, high-urgency: platform@

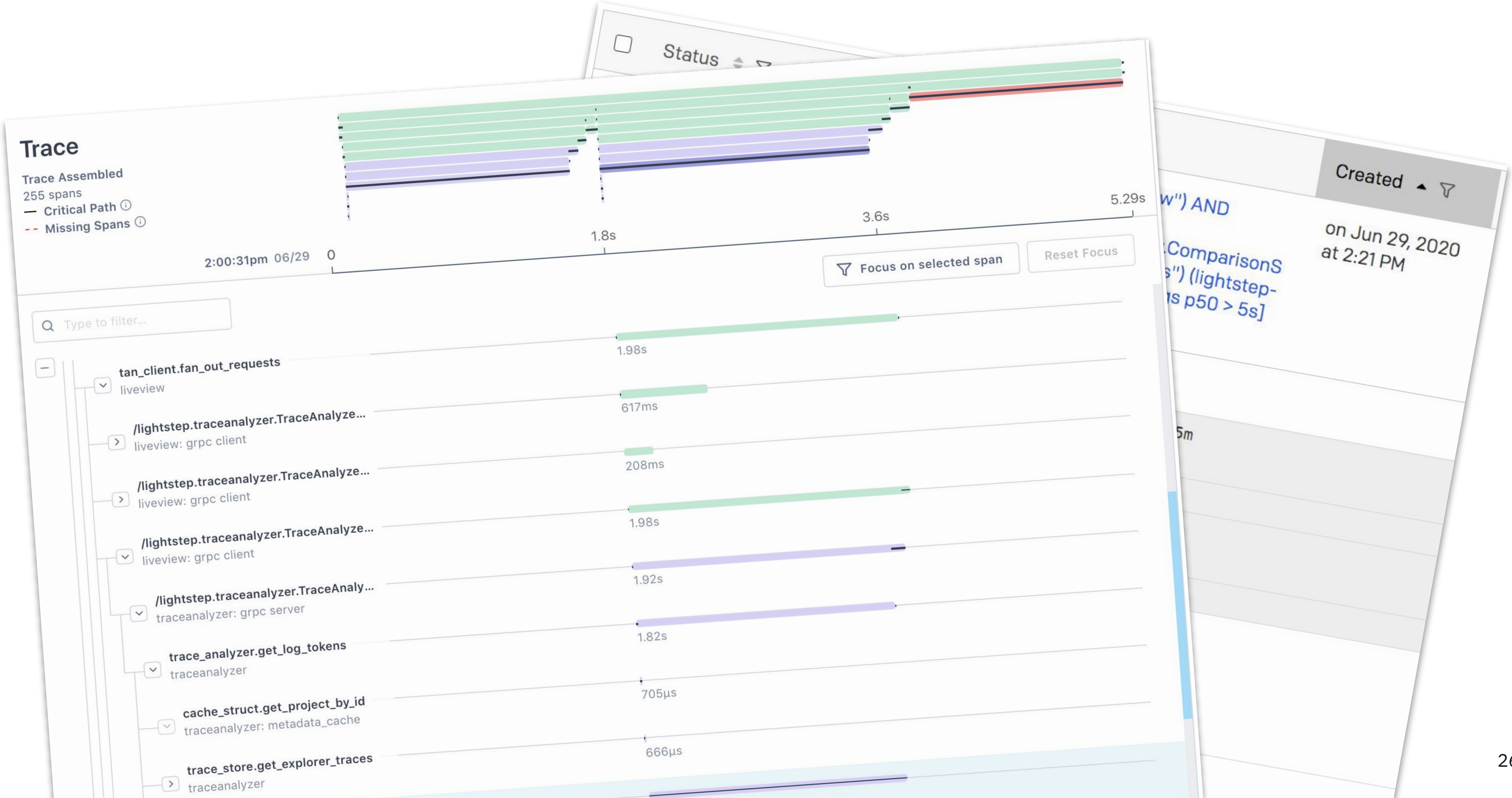
Time Zone: (UTC-07:00) Pacific Time (US &

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
Eleanor			Tamela			Chris
5	6	7	8	9	10	11
Chris			Alex			Kelly
12	13	14	15	16	17	18
Kelly			Racquel			Ben
19	20	21	22	23	24	25
Ben			Eleanor			Tamela
26	27	28	29	30	31	1
Tamela			Kennedy			Alex
2	3	4	5	6	7	8
Alex			Chris			Racquel

How to improve incident response:

- Make pages more actionable
- Deliver pages to the right teams
- Reduce the number of pages

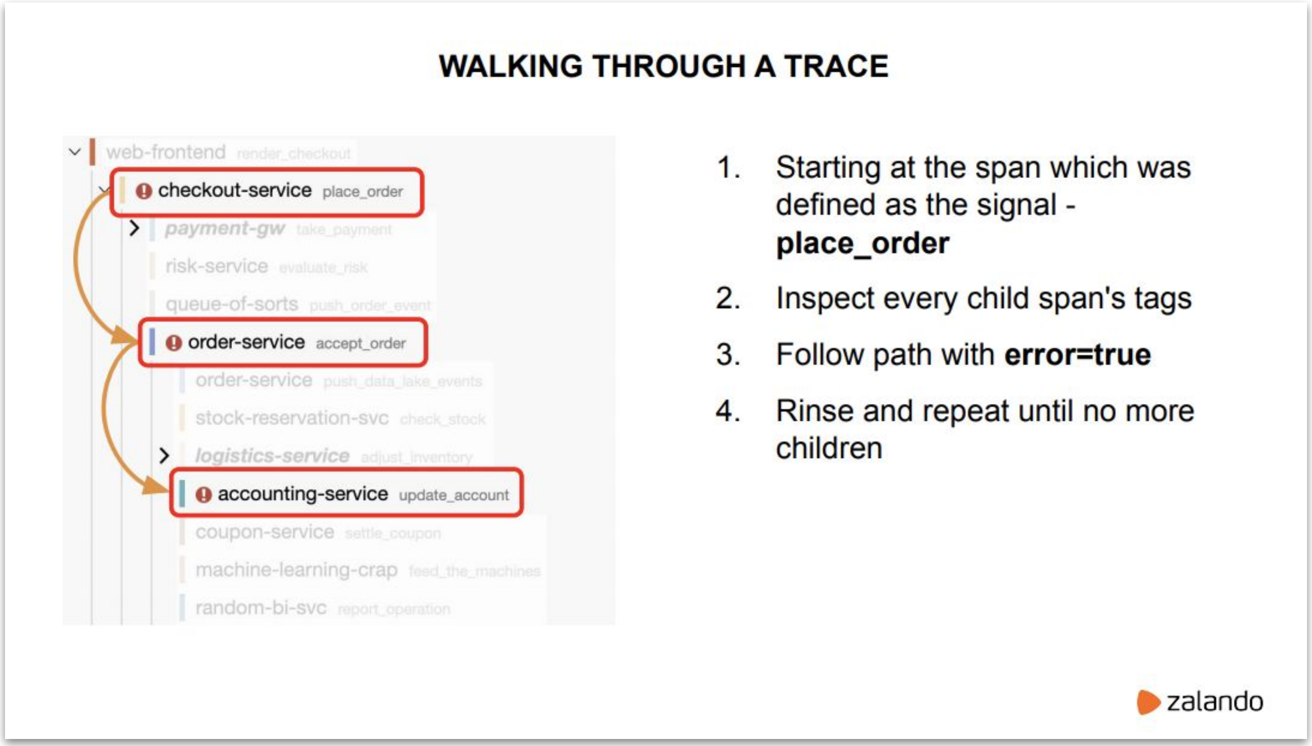
More context means faster root cause analysis



tan_client.fan_out_requests — liveview — 1.98s
/lightstep.traceanalyzer.TraceAnalyze... — liveview: grpc client — 617ms
/lightstep.traceanalyzer.TraceAnalyze... — liveview: grpc client — 208ms
/lightstep.traceanalyzer.TraceAnalyze... — liveview: grpc client — 1.98s
/lightstep.traceanalyzer.TraceAnaly... — traceanalyzer: grpc server — 1.92s
trace_analyzer.get_log_tokens — traceanalyzer — 1.82s
cache_struct.get_project_by_id — traceanalyzer: metadata_cache — 705µs
trace_store.get_explorer_traces — traceanalyzer — 666µs

Dynamic alert delivery

Send alerts directly to the teams that are responsible for taking action!


web-frontend render_checkout
checkout-service place_order
payment-gw take_payment
risk-service evaluate_risk
queue-of-sorts push_order_event
order-service accept_order
order-service push_data_lake_events
stock-reservation-svc check_stock
logistics-service adjust_inventory
accounting-service update_account
coupon-service settle_coupon
machine-learning-crap feed_the_machines
random-bi-svc report_operation
zalando

WALKING THROUGH A TRACE

1. Starting at the span which was defined as the signal - **place_order**
2. Inspect every child span's tags
3. Follow path with **error=true**
4. Rinse and repeat until no more children

Are We All on the Same Page? Let’s Fix That
Luis Mineiro @voidmaze SRE @ Zalando, SREcon EMEA 2019
https://www.usenix.org/conference/srecon19emea/presentation/mineiro

Improving postmortems

"How will we do better next time?"

- Make sure root causes are fixed
- Improve responses for novel issues

Make sure postmortems are blameless

- By establishing what happened *objectively* using traces

Why is improving oncall important?

Direct impact on customer experience (revenue, reputation, etc.)

Time spent handing pages, writing postmortems, handling interrupts is... time **not** spent building new features, proactive optimization

Stress of oncall has major impact on job satisfaction

SLOs

Service Level Objectives

- Promises service owners make to their customers
  - Both external and internal customers
- Stated in a way that can be **measured** on short time scales
  - Days, hours, or even minutes

Common indicators

- Latency (p50, p99, etc.)
- Error rate
- Availability

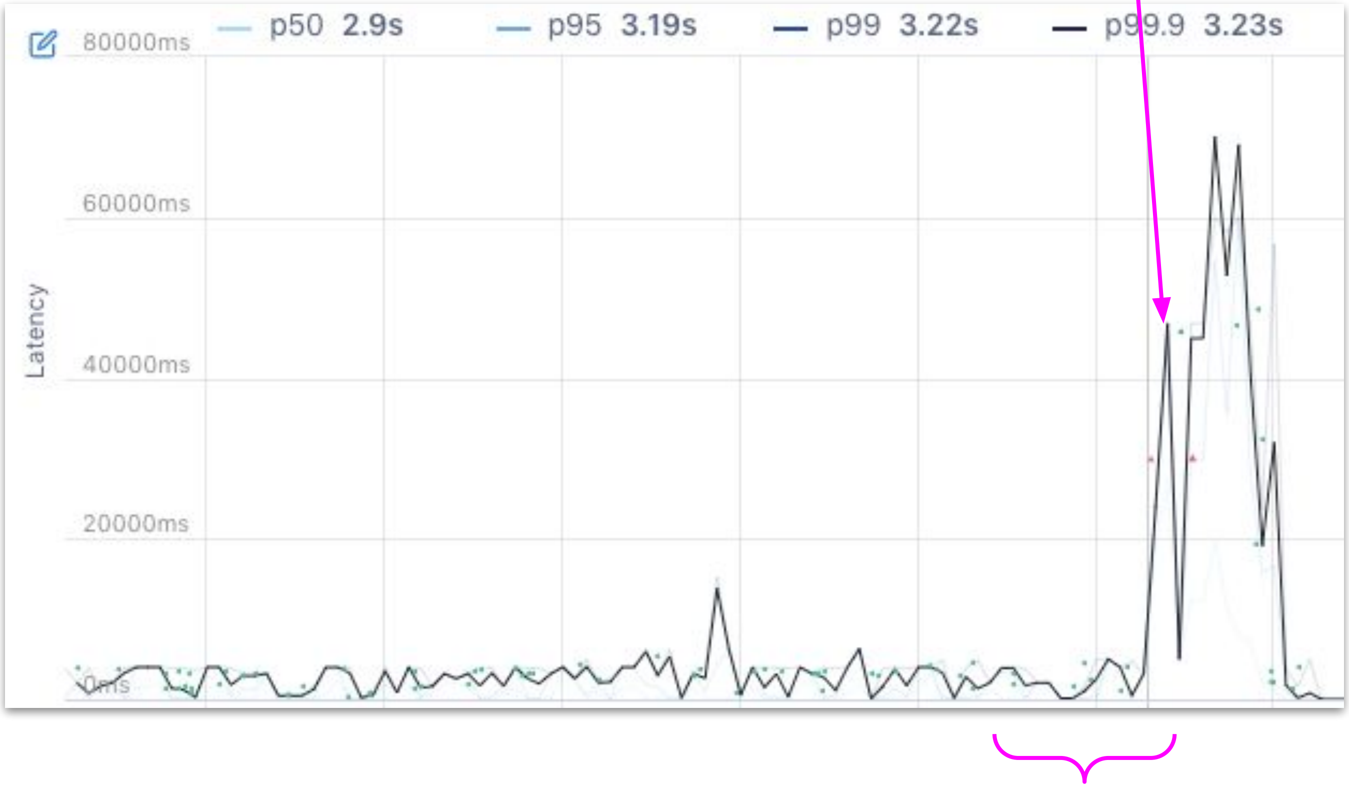
threshold

`p99 latency < 5s` `over the last` **`5 minutes`**

service level indicator (SLI)

measurement window

“instantaneous” latency



latency over 5 minute window

Determining SLOs

Ask:

- What do your customers expect?
- What can you provide today?
- How do you expect that to change?

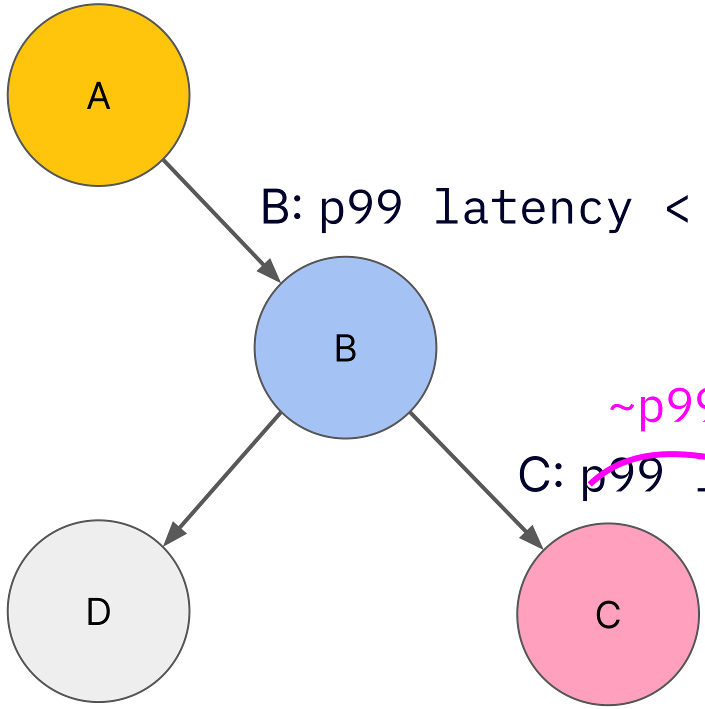
Derive internal SLOs using tracing

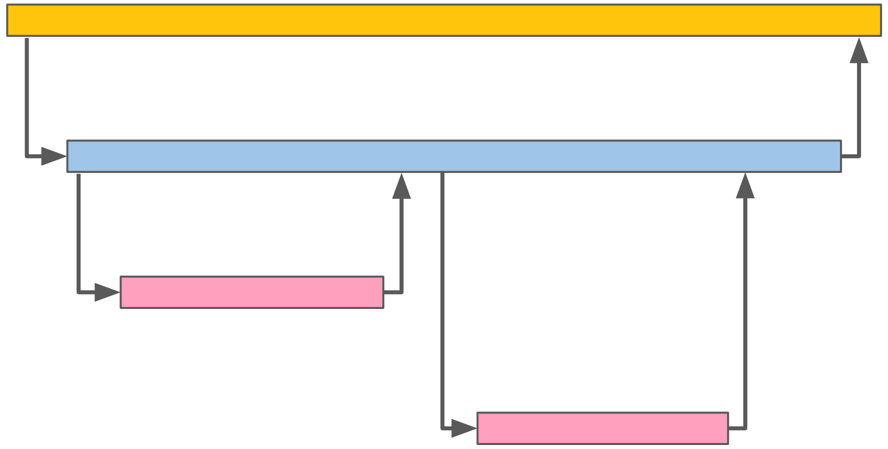
A: p99 latency < 5s

B: p99 latency < 5s

~p99.5 latency < 2.5s

C: ~~p99 latency < 2.5s~~





Why are SLOs important?

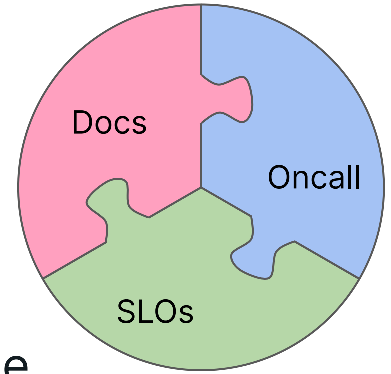
They measure success in delivering service

Teams use them as a guide to prioritize work

Consistency across your org

- Hold teams accountable in a consistent way
- Do it transparently

3-piece puzzle review



Documentation

- Establishes ownership: who you will hold accountable
- Also provides critical information for building confidence

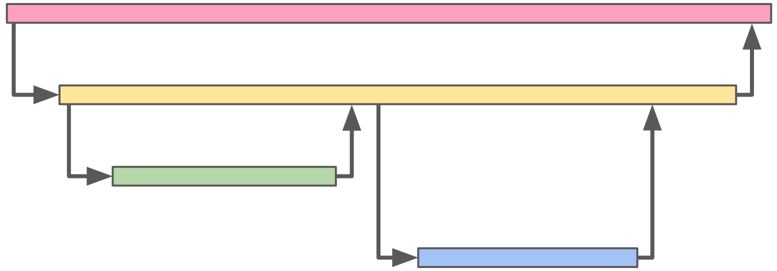
Oncall

- Oncall (and service ownership) are more than incident response

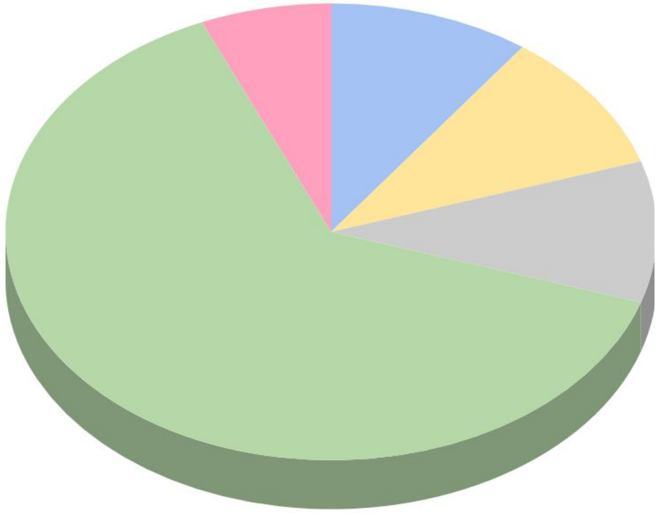
SLOs

- How you measure success!

Tracing provides key information for each



Budgeting for ownership



Giving teams agency to improve reliability

- Will help them hit their goals
- Lower their stress

But having agency requires

- The right information
- The time to act on it

Ownership doesn’t come for free!

Next steps...

Making changes

Rolling out new processes/tools in a DevOps org is hard

- Process/tools must provide **value to app dev teams**
- Ideally, they are **necessary** parts of their day-to-day work

To establish and maintain service ownership

- Use a combination of docs, oncall process, and SLOs
- *Manufacture* a need for those process/tools where necessary

Ownership = Accountability + Agency

Accountability

- Set deliverables and goals for service owners
- Judge their performance based on those deliverables and goals

Agency

- Offer the information, confidence, and budget to improve

Thank you

@save_spoons

Daniel “Spoons” Spoonhower, CTO and Co-founder

lightstep.com

Lightstep