Distributed Tracing and Traffic Splitting

Charles Pretzer

@charpretz

cpretzer

@charles on https://slack.linkerd.io

charles@buoyant.io

Agenda

Service Mesh Overview

Deploying the Linkerd Service Mesh

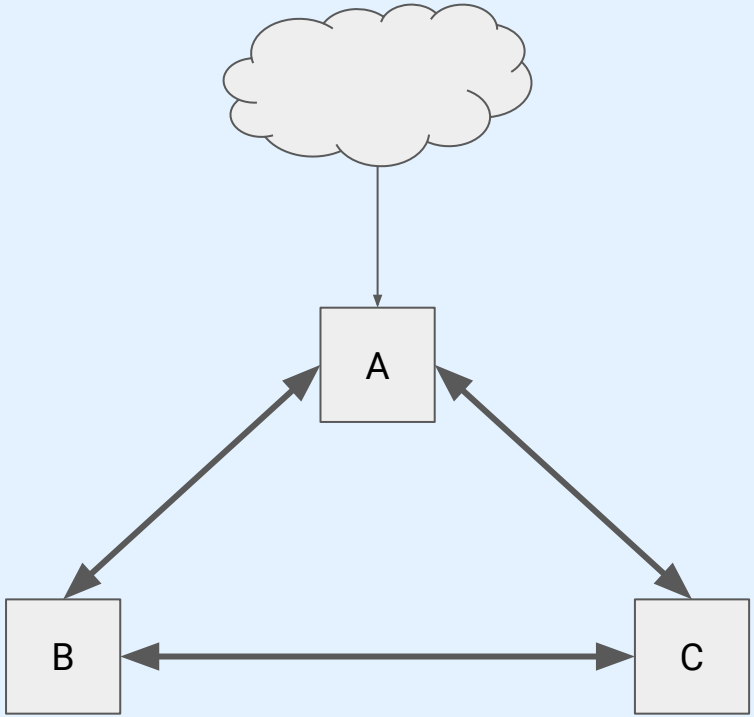
Distributed Tracing

Splitting Traffic

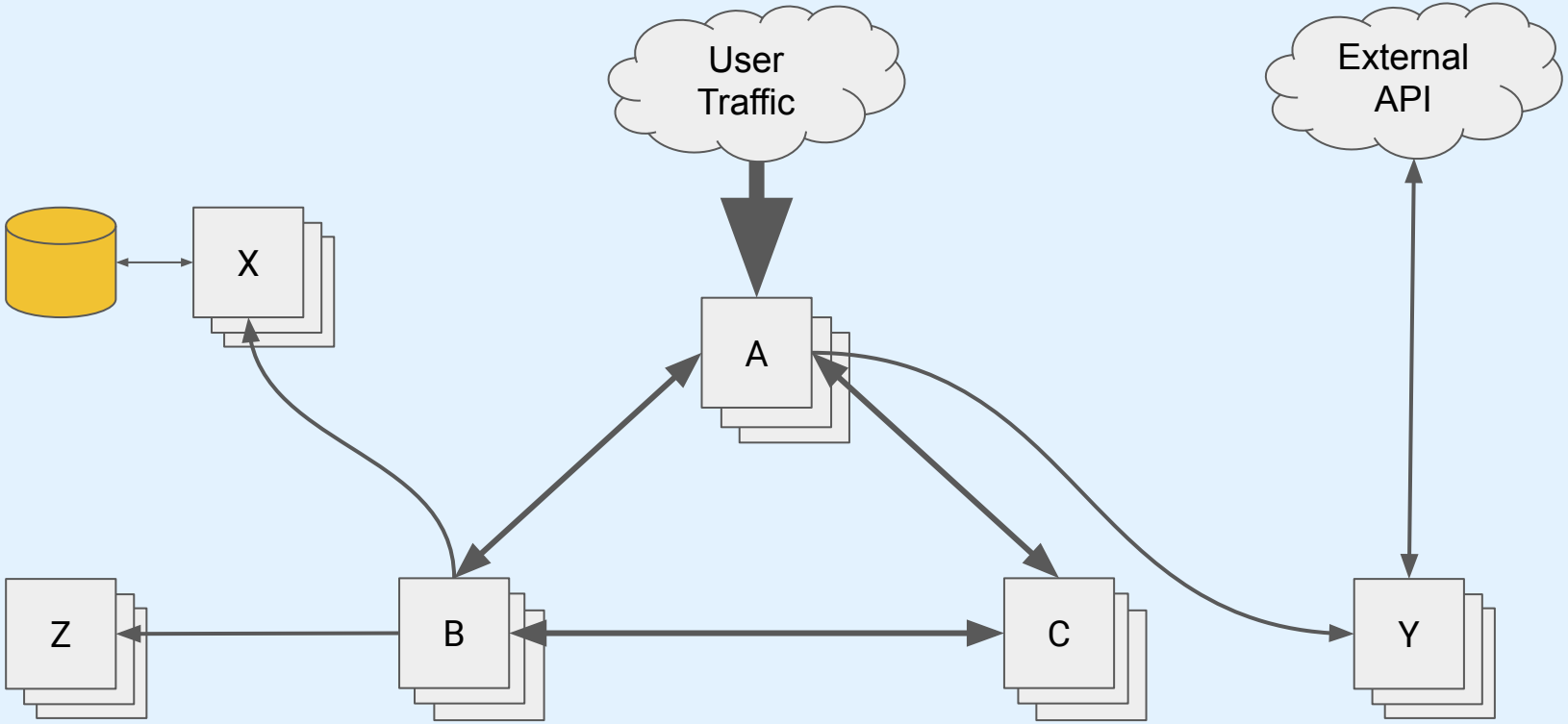
Questions/Discussion

Service Mesh Overview

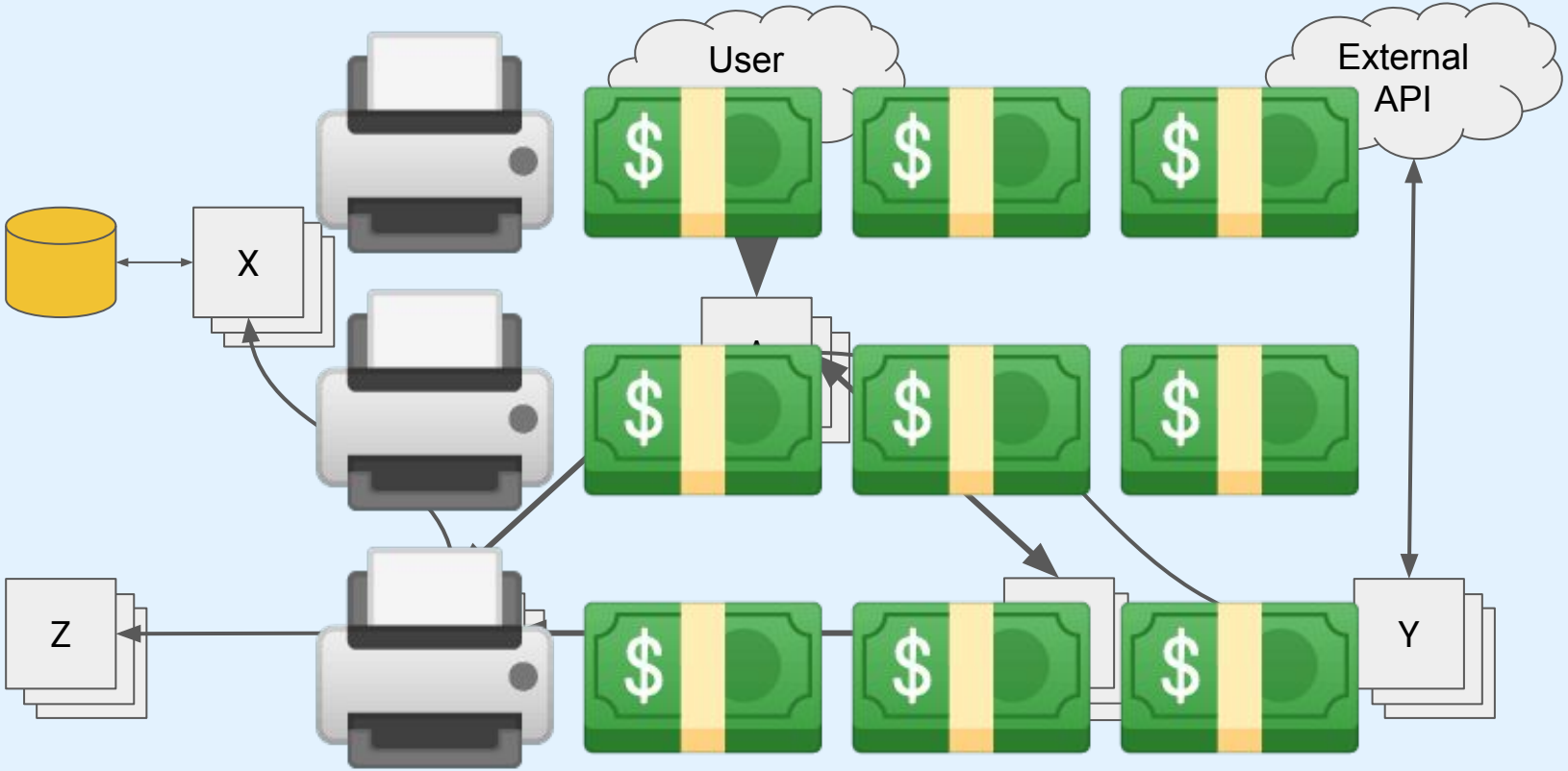
What's a service mesh?



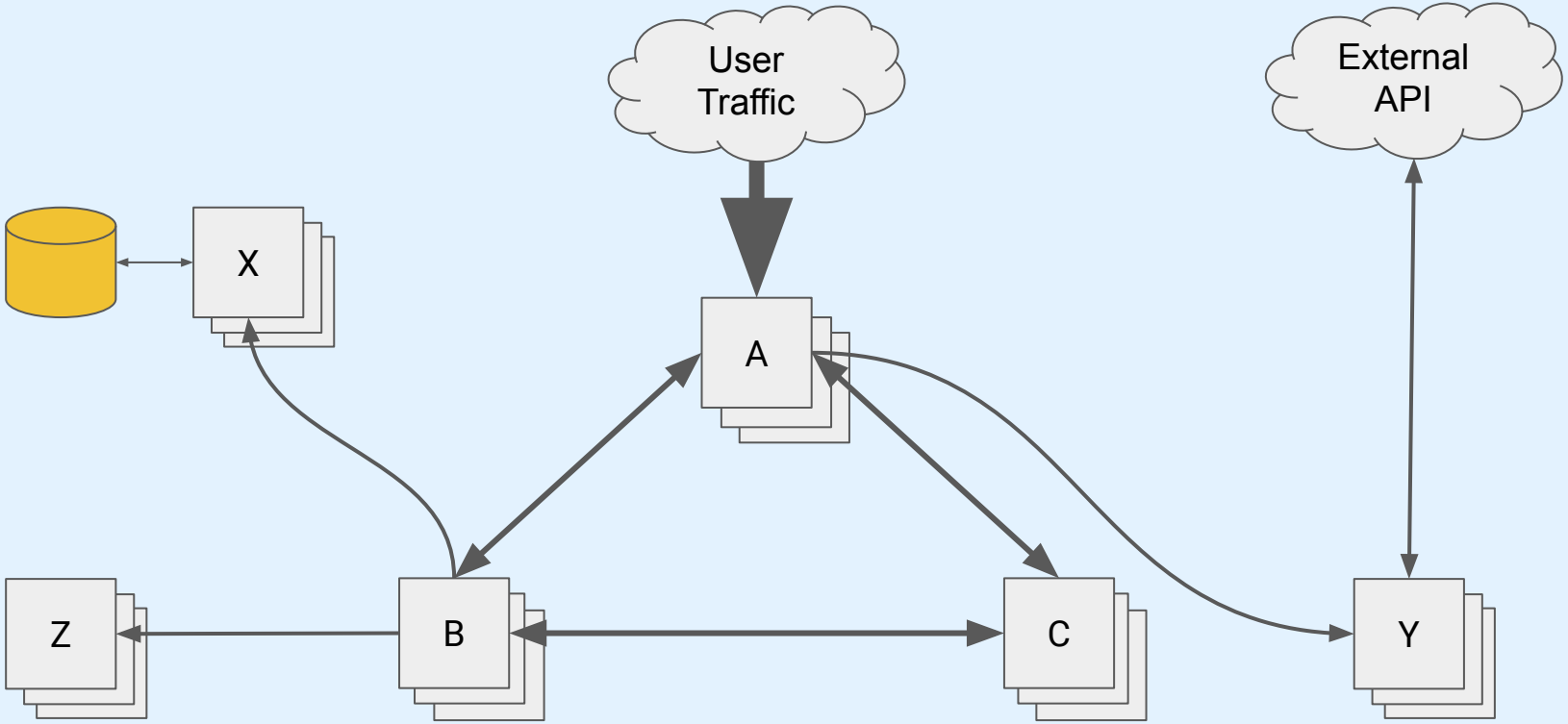
Service Mesh: The basics



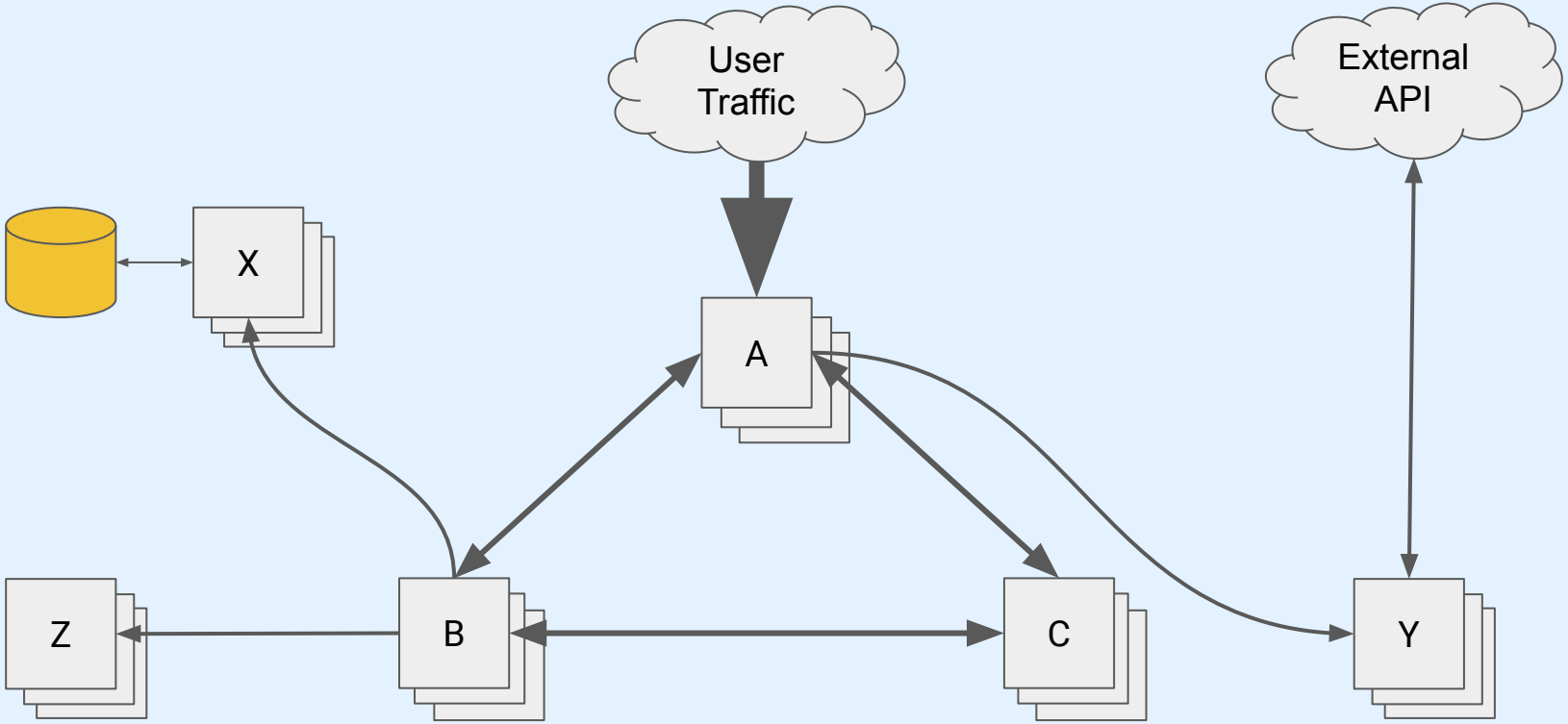
Service Mesh: The basics



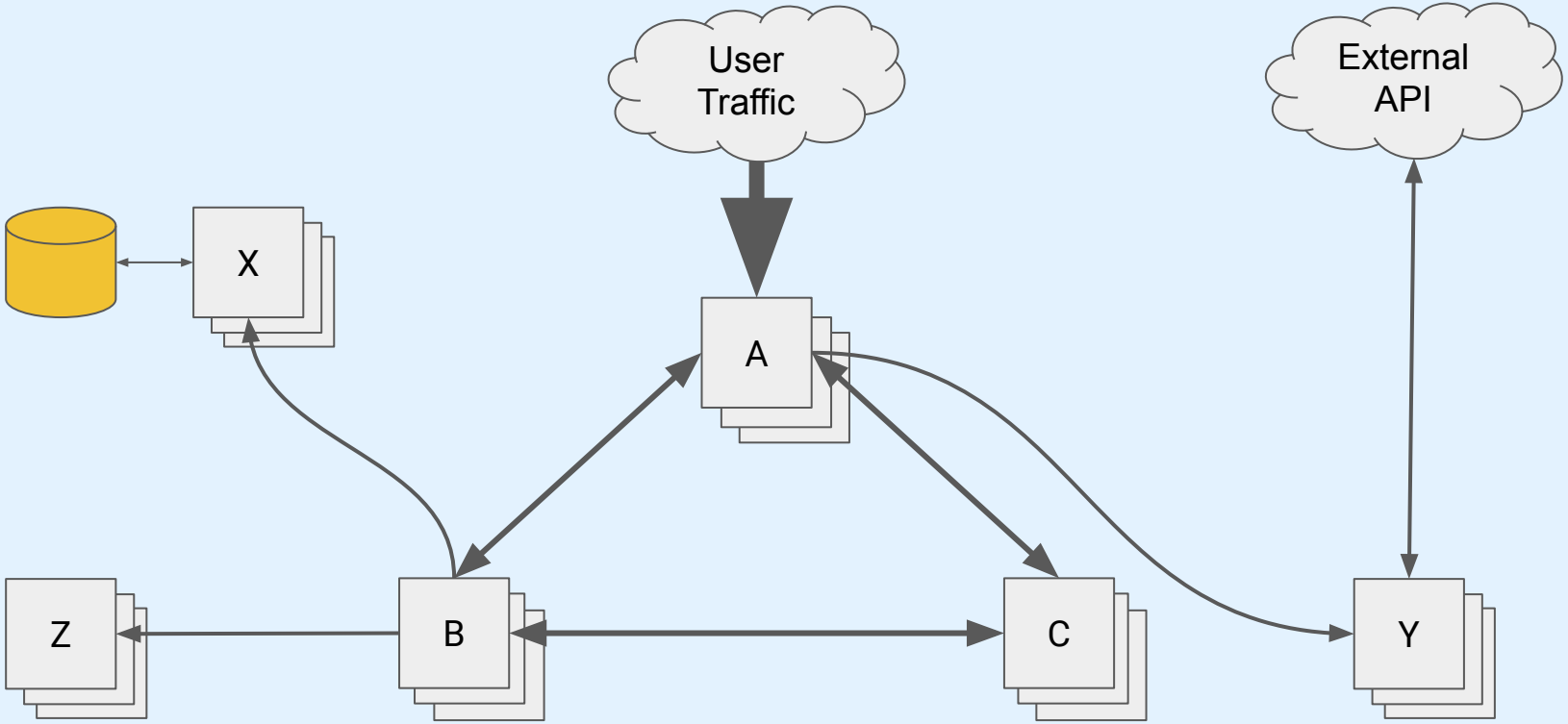
Service Mesh: The basics



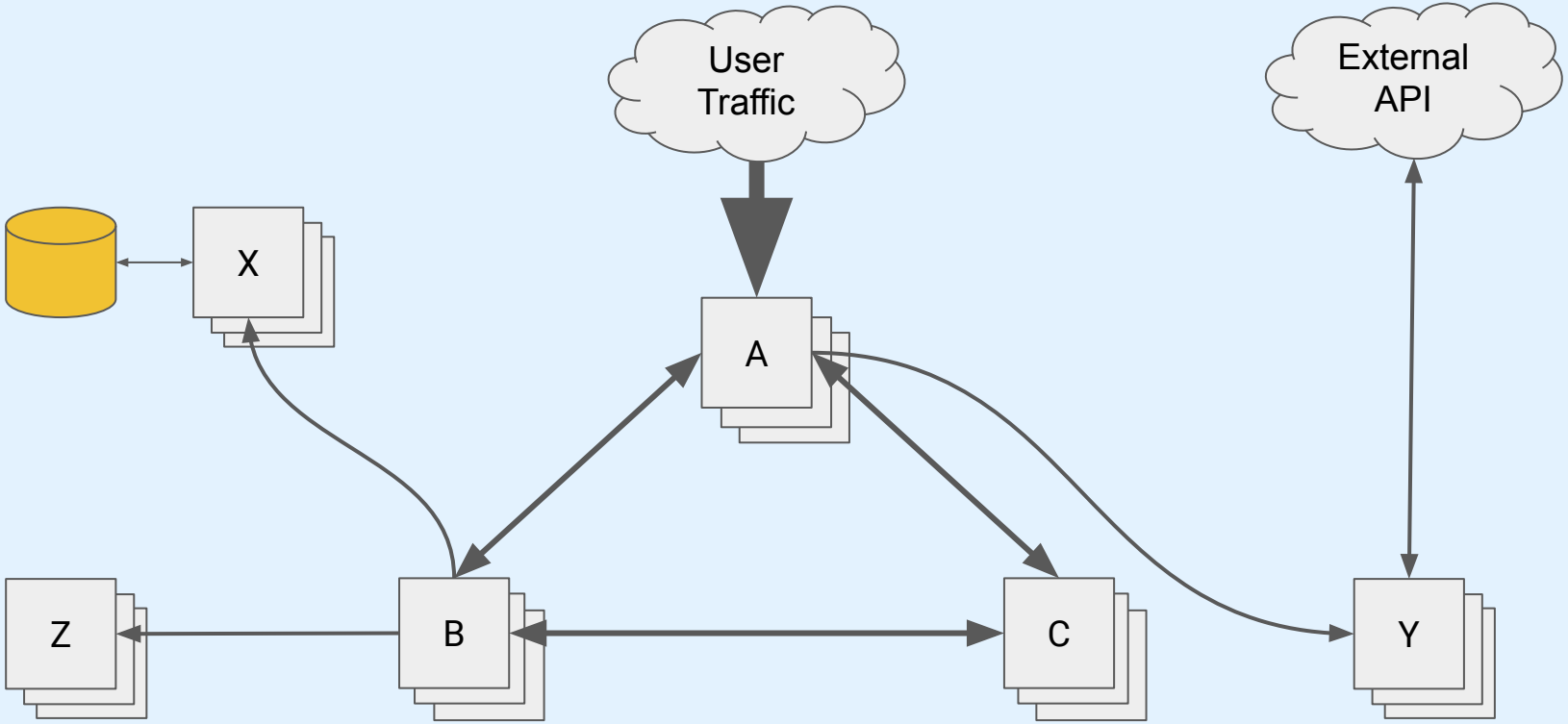
Service Mesh: The basics



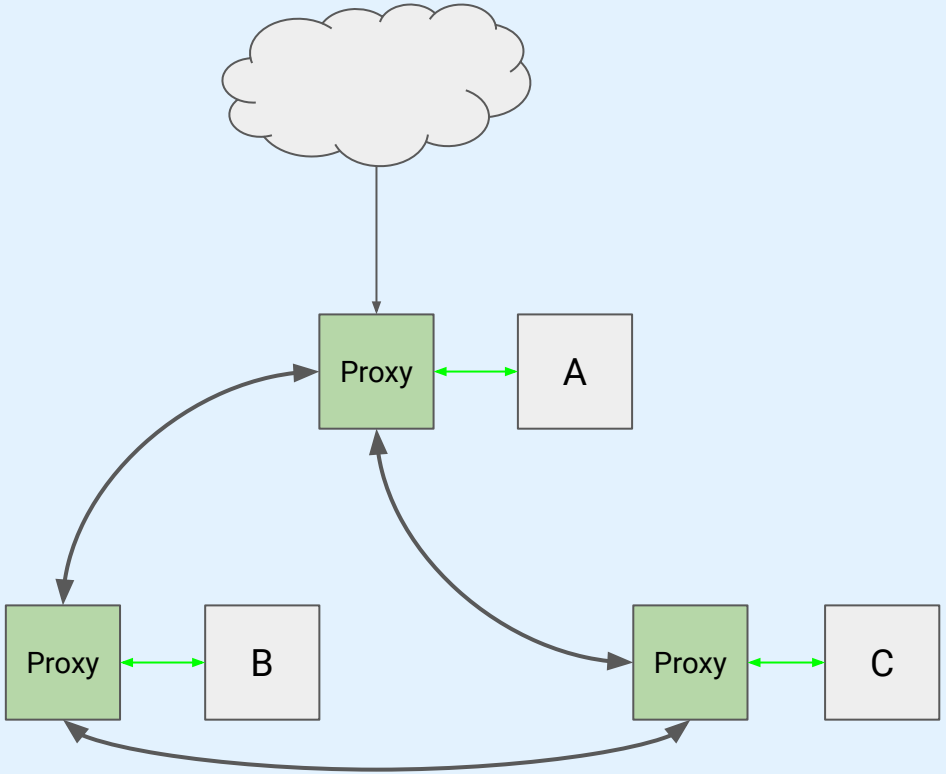
Service Mesh: The basics



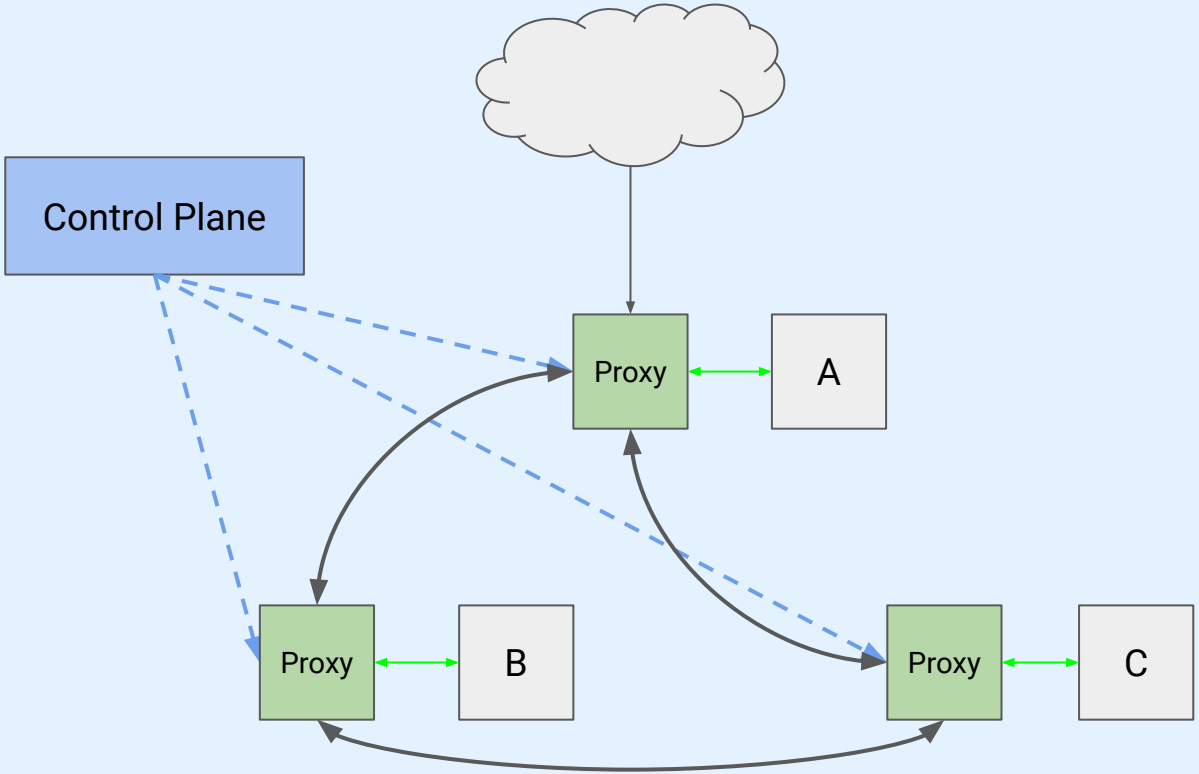
Service Mesh: The basics



Service Mesh: Data Plane



Service Mesh: Control Plane



Linkerd

- Observability: Collecting actionable traffic metrics
- Security: Encrypting traffic between services
- Reliability: Ensuring services are available
- Traffic Management: Routing traffic to services

Linkerd: Observability

- Rich traffic metrics: "Golden Metrics"
  - Request rate, Success rate, latency
  - Across many dimensions
- Request inspection
- Distributed Tracing

Linkerd: Security

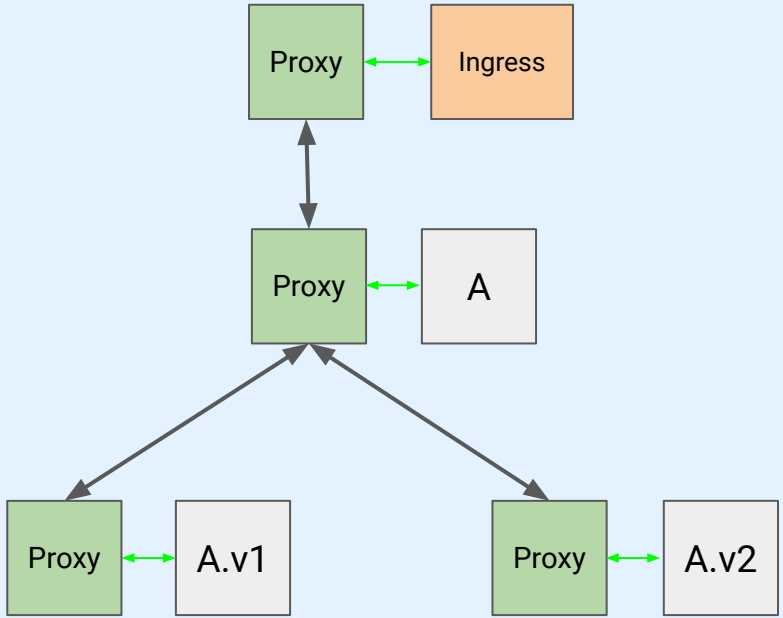
- Cryptographic identity
  - mTLS between services
  - On by default
  - Transparent

Linkerd: Reliability

- Latency aware load balancing
- Retries
- Timeouts

Linkerd: Traffic Management

- Traffic Split
  - Introduced in 2.4.0
  - Enables canary and blue/green deployments



Distributed Tracing

Distributed Tracing

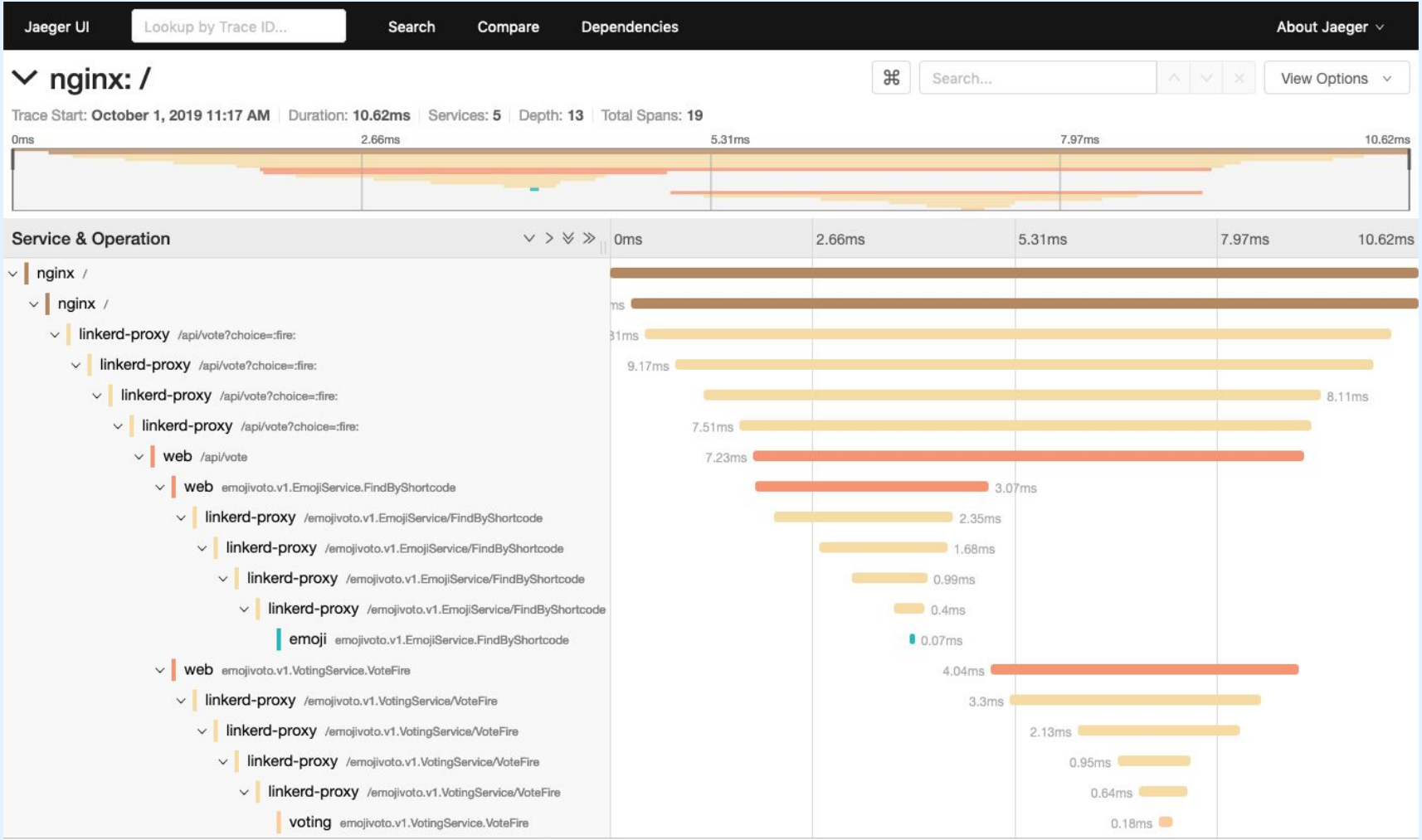
Overview

Myths

Demo

Distributed Tracing

Distributed Tracing


Jaeger UI
Lookup by Trace ID...
Search
Compare
Dependencies
About Jaeger
nginx: /
Search...
View Options
Trace Start: October 1, 2019 11:17 AM | Duration: 10.62ms | Services: 5 | Depth: 13 | Total Spans: 19
0ms
2.66ms
5.31ms
7.97ms
10.62ms
Service & Operation
nginx /
nginx /
linkerd-proxy /api/vote?choice=:fire:
linkerd-proxy /api/vote?choice=:fire:
linkerd-proxy /api/vote?choice=:fire:
linkerd-proxy /api/vote?choice=:fire:
web /api/vote
web emojivoto.v1.EmojiService.FindByShortcode
linkerd-proxy /emojivoto.v1.EmojiService/FindByShortcode
linkerd-proxy /emojivoto.v1.EmojiService/FindByShortcode
linkerd-proxy /emojivoto.v1.EmojiService/FindByShortcode
linkerd-proxy /emojivoto.v1.EmojiService/FindByShortcode
emoji emojivoto.v1.EmojiService.FindByShortcode
web emojivoto.v1.VotingService.VoteFire
linkerd-proxy /emojivoto.v1.VotingService/VoteFire
linkerd-proxy /emojivoto.v1.VotingService/VoteFire
linkerd-proxy /emojivoto.v1.VotingService/VoteFire
linkerd-proxy /emojivoto.v1.VotingService/VoteFire
voting emojivoto.v1.VotingService.VoteFire
9.17ms
8.11ms
7.51ms
7.23ms
3.07ms
2.35ms
1.68ms
0.99ms
0.4ms
0.07ms
4.04ms
3.3ms
2.13ms
0.95ms
0.64ms
0.18ms

Distributed Tracing

- Visualize dependencies for an individual request
- Visualize request timings and orderings
  - How much time is spent in each service?
  - How much time is spent in the network?
  - Which tasks are executed in parallel?

Four Myths about Distributed Tracing

Myth 1: I need distributed tracing to measure the health / latency / throughput of my services

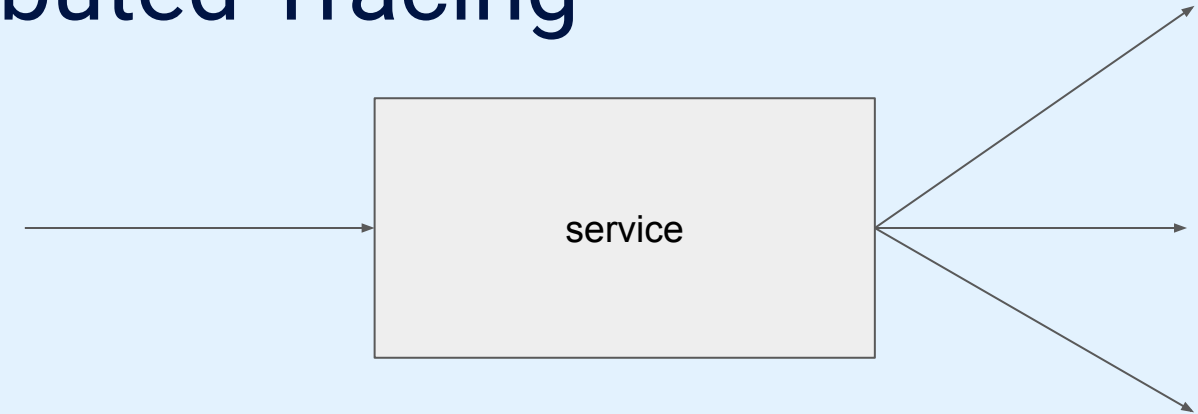
Myth 2: I need distributed tracing to see which services talk to which other services

Myth 3: I can “get distributed tracing” from the service mesh without having to make any changes to my application

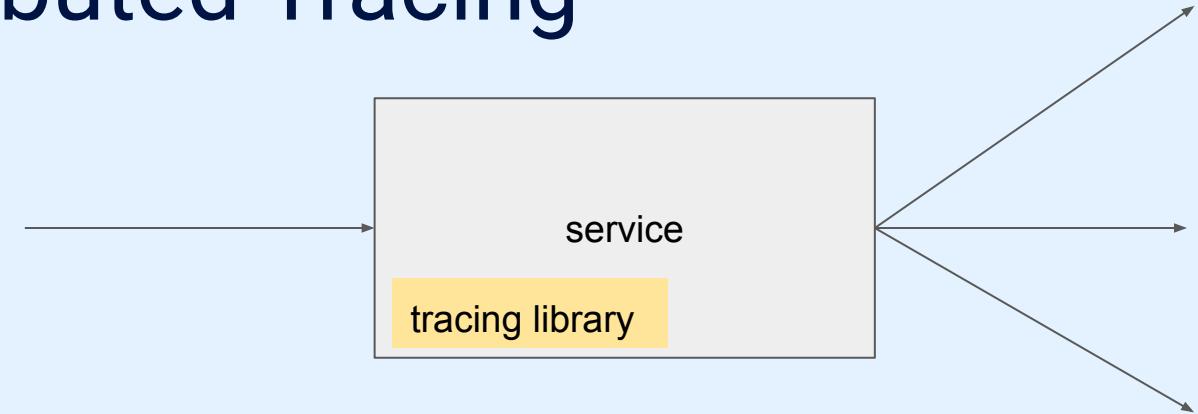
Myth 4: I can “get distributed tracing” from a service mesh by instrumenting my application with a distributed tracing library

Distributed Tracing Architecture

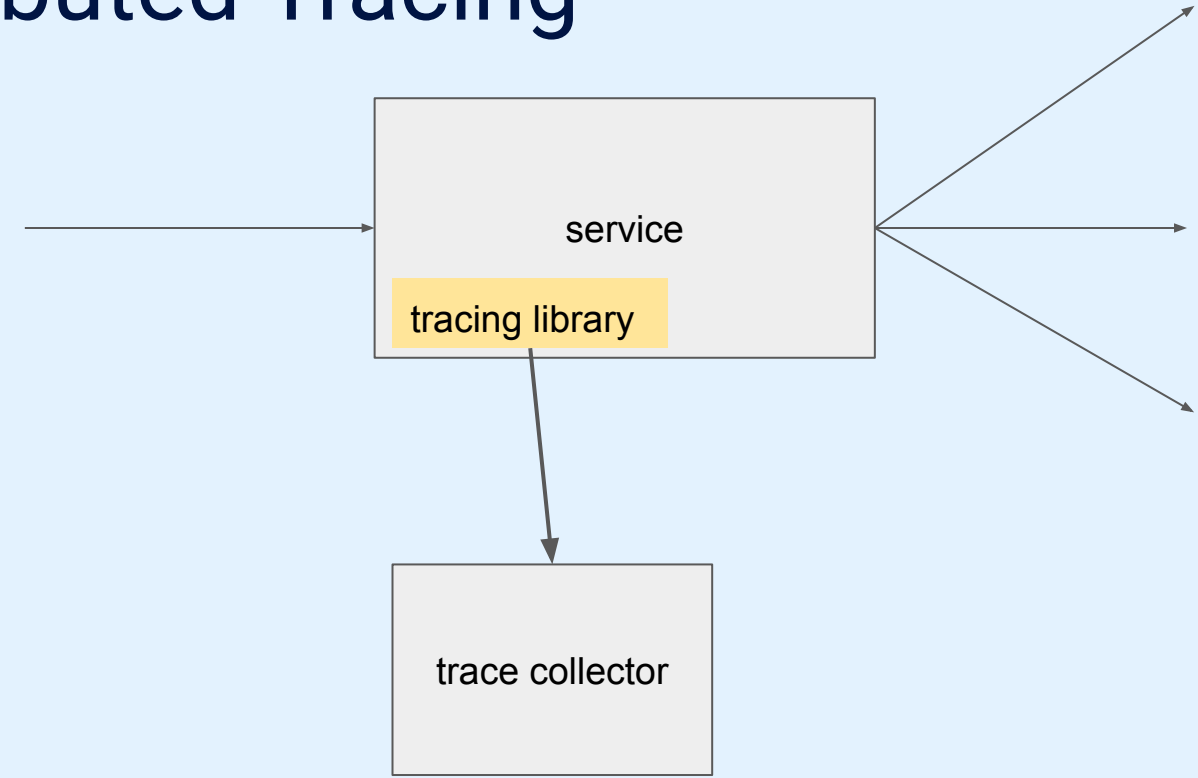
Distributed Tracing



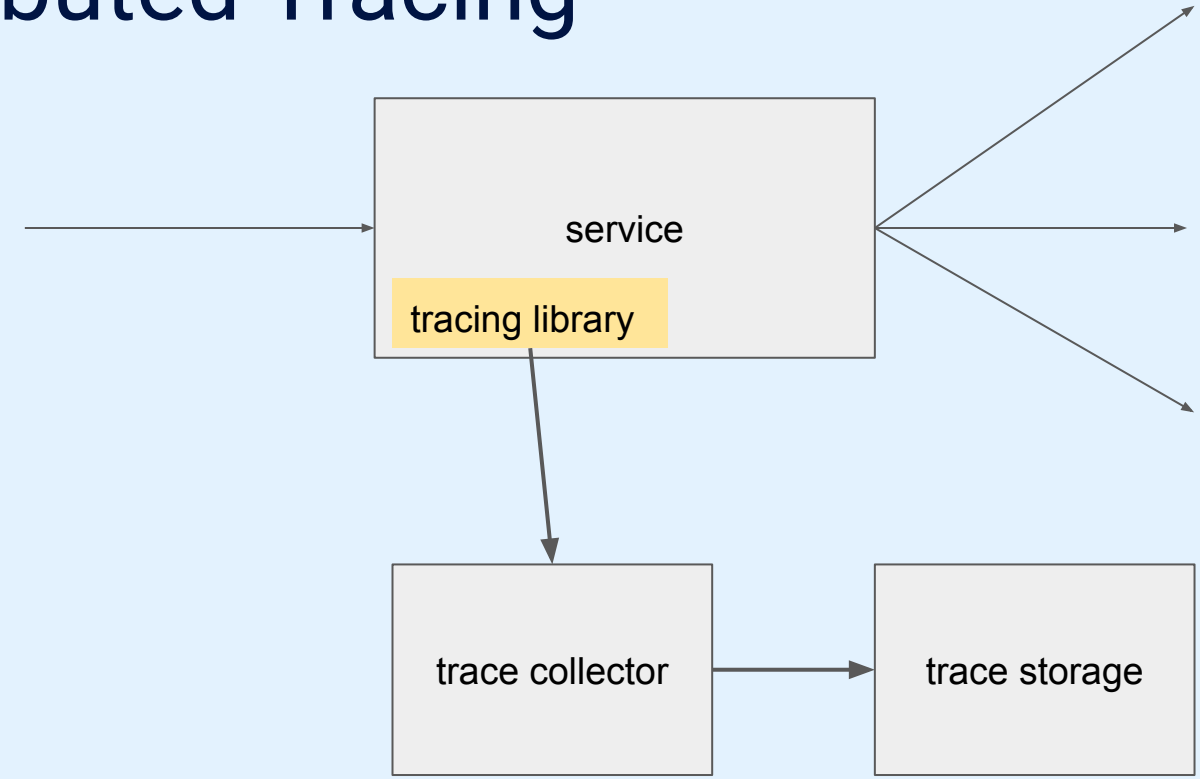
Distributed Tracing



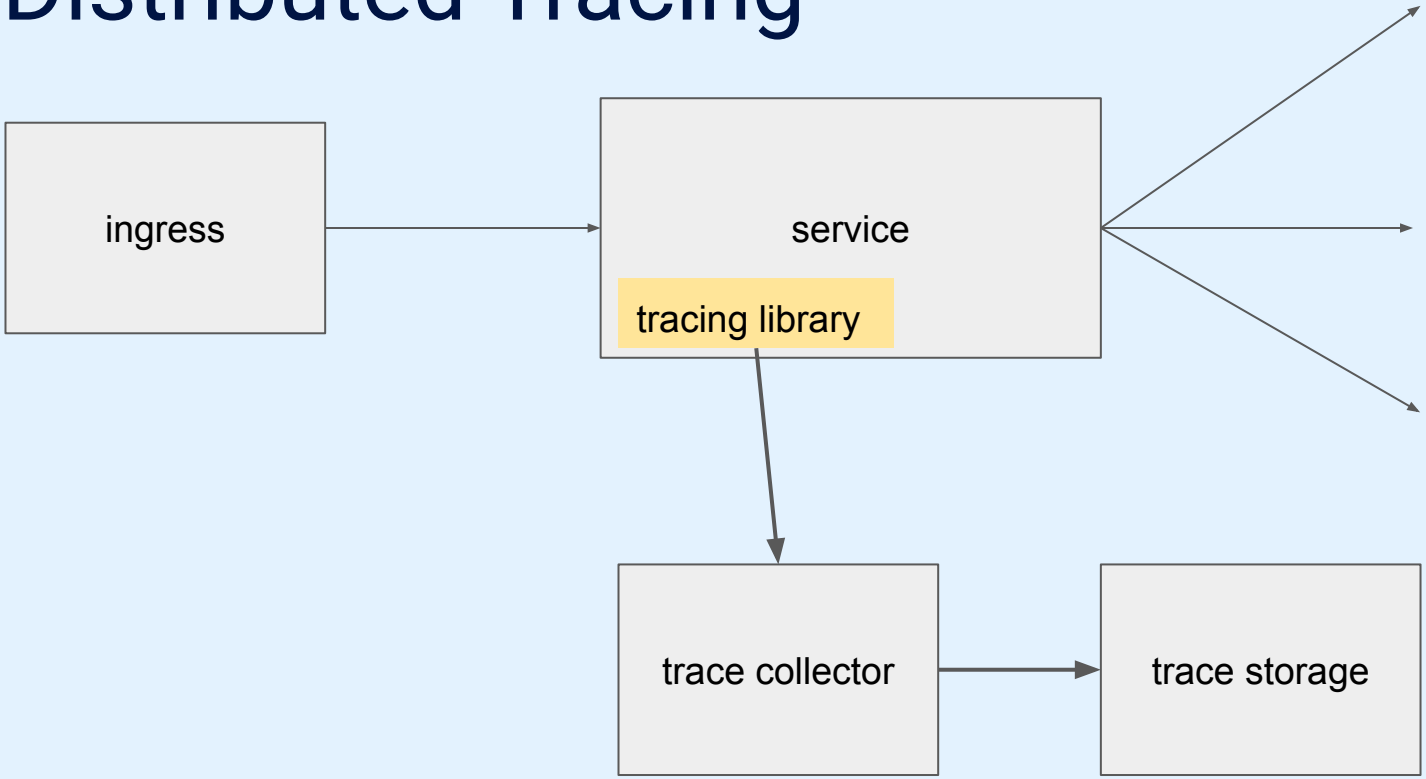
Distributed Tracing



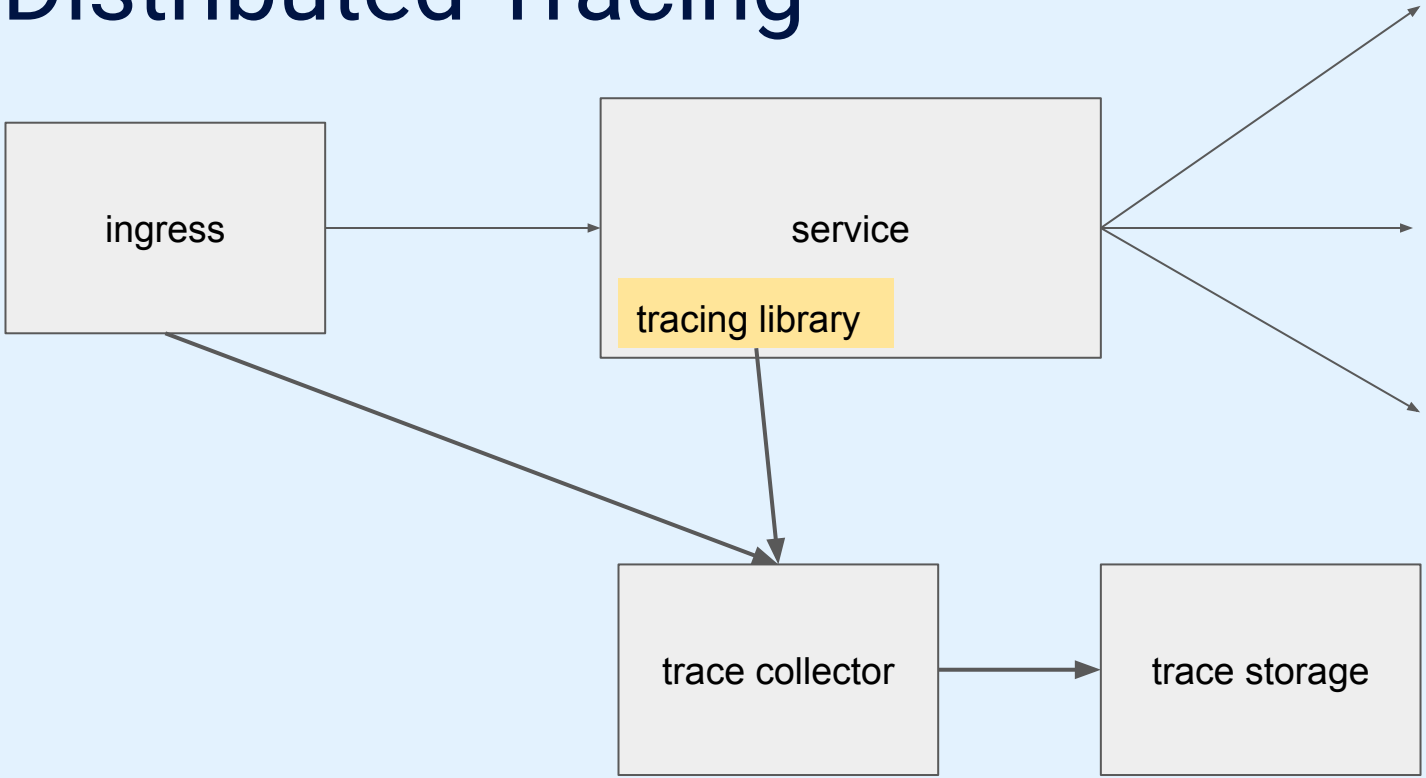
Distributed Tracing



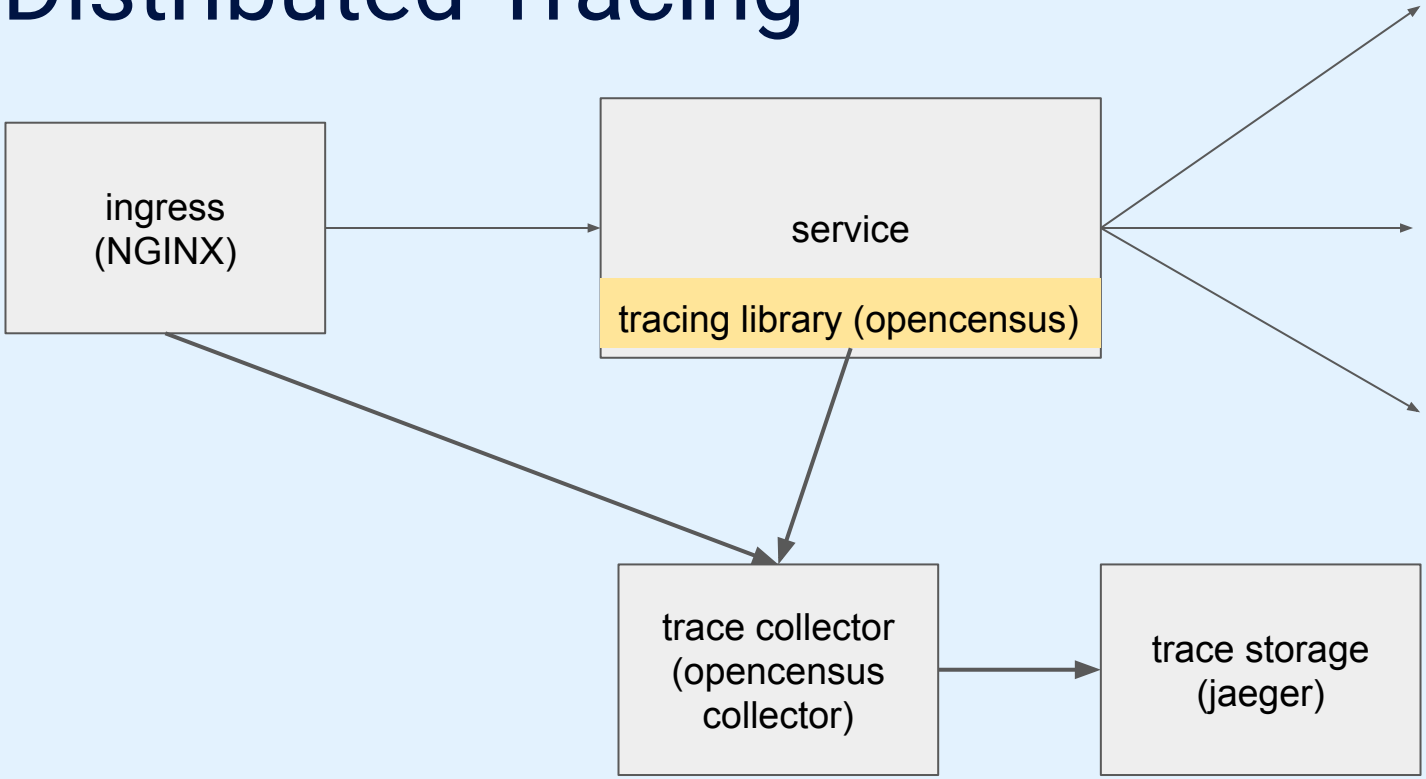
Distributed Tracing



Distributed Tracing



Distributed Tracing

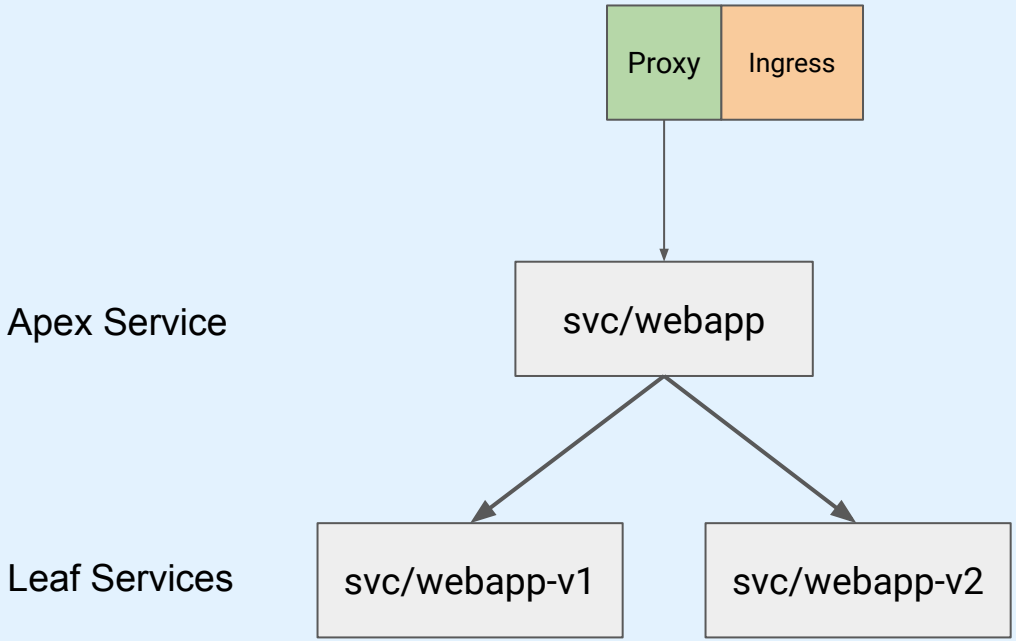


Traffic Split with SMI and Linkerd

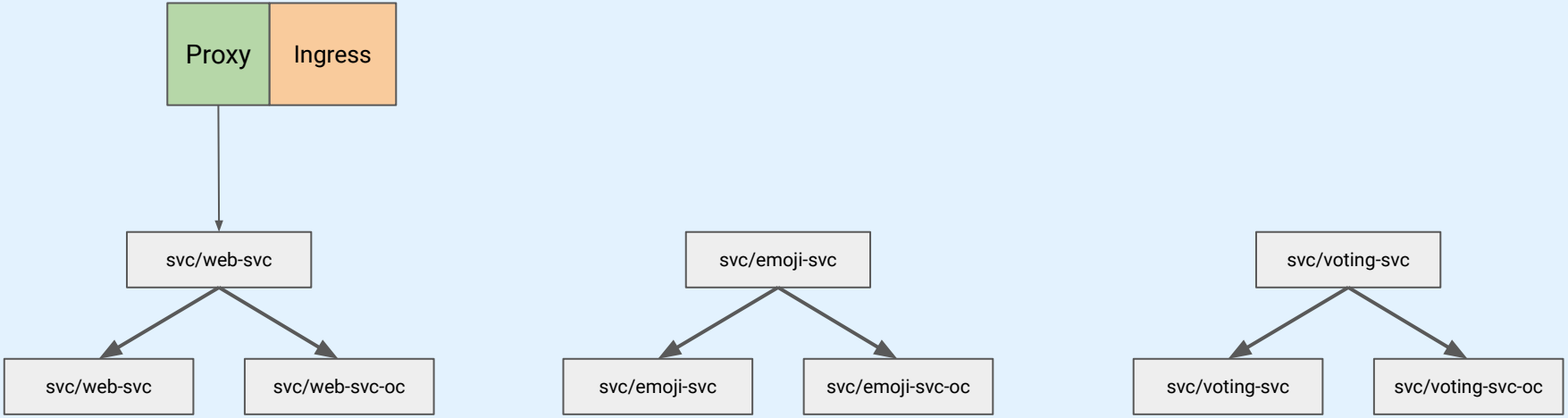
Traffic Split

- Part of the Service Mesh Interface (SMI)
- TrafficSplit is a Custom Resource Definition (CRD)
- Tells the service mesh to redirect traffic
- Traffic sent to the **apex** is redirected to a **leaf**
- Supports probabilistic weighting
- Traffic splitting always happens client-side

Traffic Split



Traffic Split



Join our community!

github.com/linkerd

slack.linkerd.io

@linkerd

FROM YOUR FRIENDS AT