Instrumenting Lambda with OpenTelemetry

Meetup SRE / OpenTelemetry France – 23th March – v1
Dimitris Finas

Agenda

- What is OpenTelemetry?

- Why use it for Lambda?

- How to do it?

What is OpenTelemetry?

OpenTelemetry is an open source project. It is a set of APIs, SDKs, tooling and integrations that are designed for the creation and management of *telemetry data,* such as traces, metrics, and logs.

OpenTelemetry's Mission is to enable effective observability by making high-quality, portable telemetry ubiquitous and vendor-agnostic.

Traces

Metrics

Logs
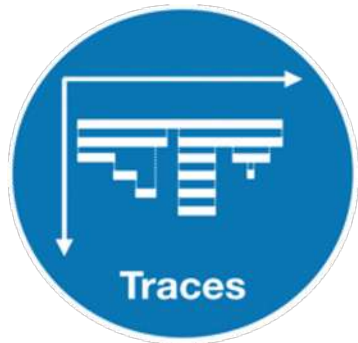
A trace represents a single user's journey across multiple applications and systems (usually microservices).
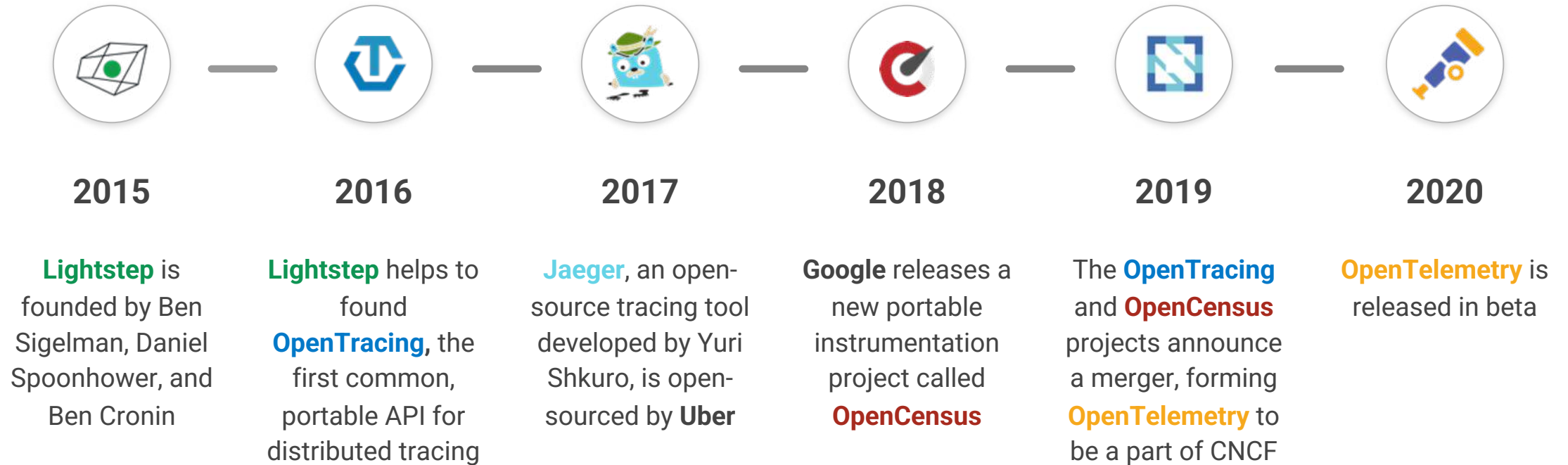
Numeric data measured at various time intervals (time series data); SLI's (request rate, error rate, duration, CPU%, etc.)
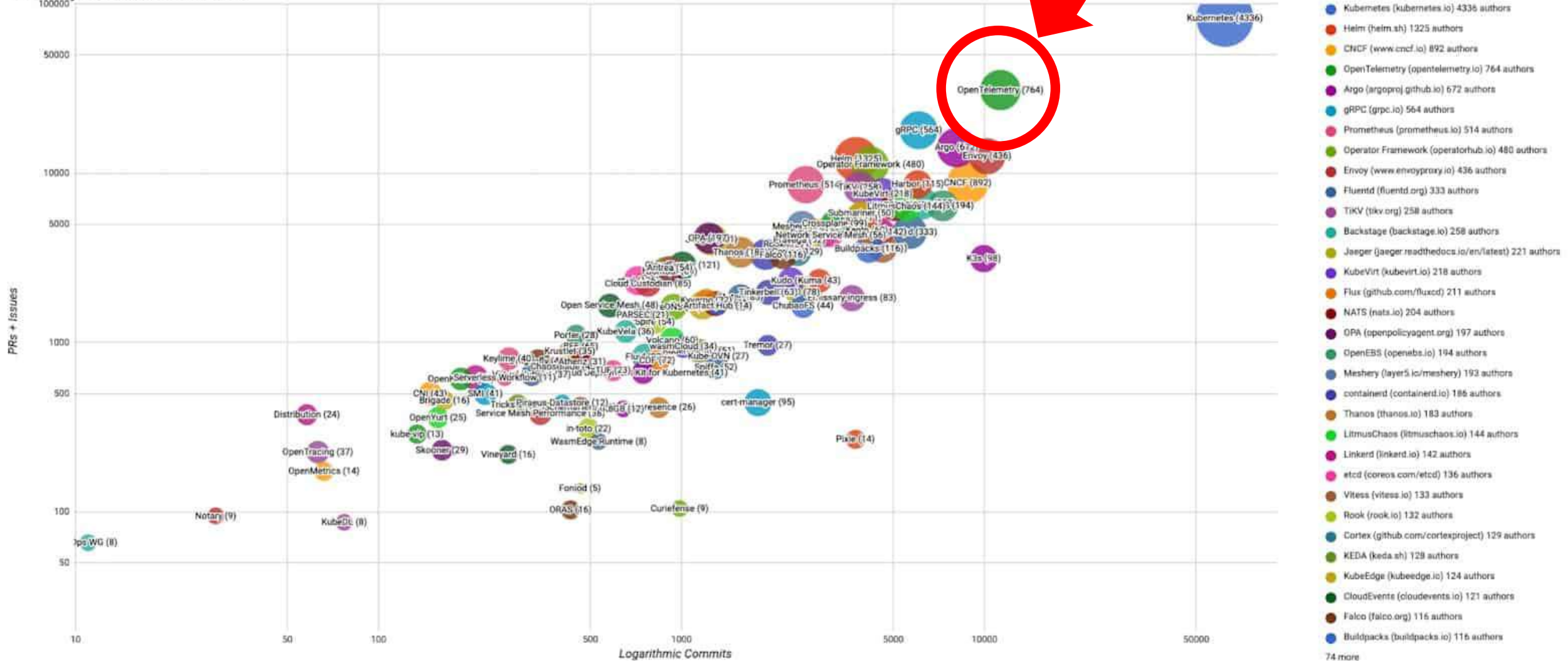
Timestamped records of discrete events that happened within an application or system, such as a failure, an error, or a state transformation

You want more? See Otel vision in https://github.com/open-telemetry/community/blob/main/mission-vision-values.md#otel-mission-vision-and-values
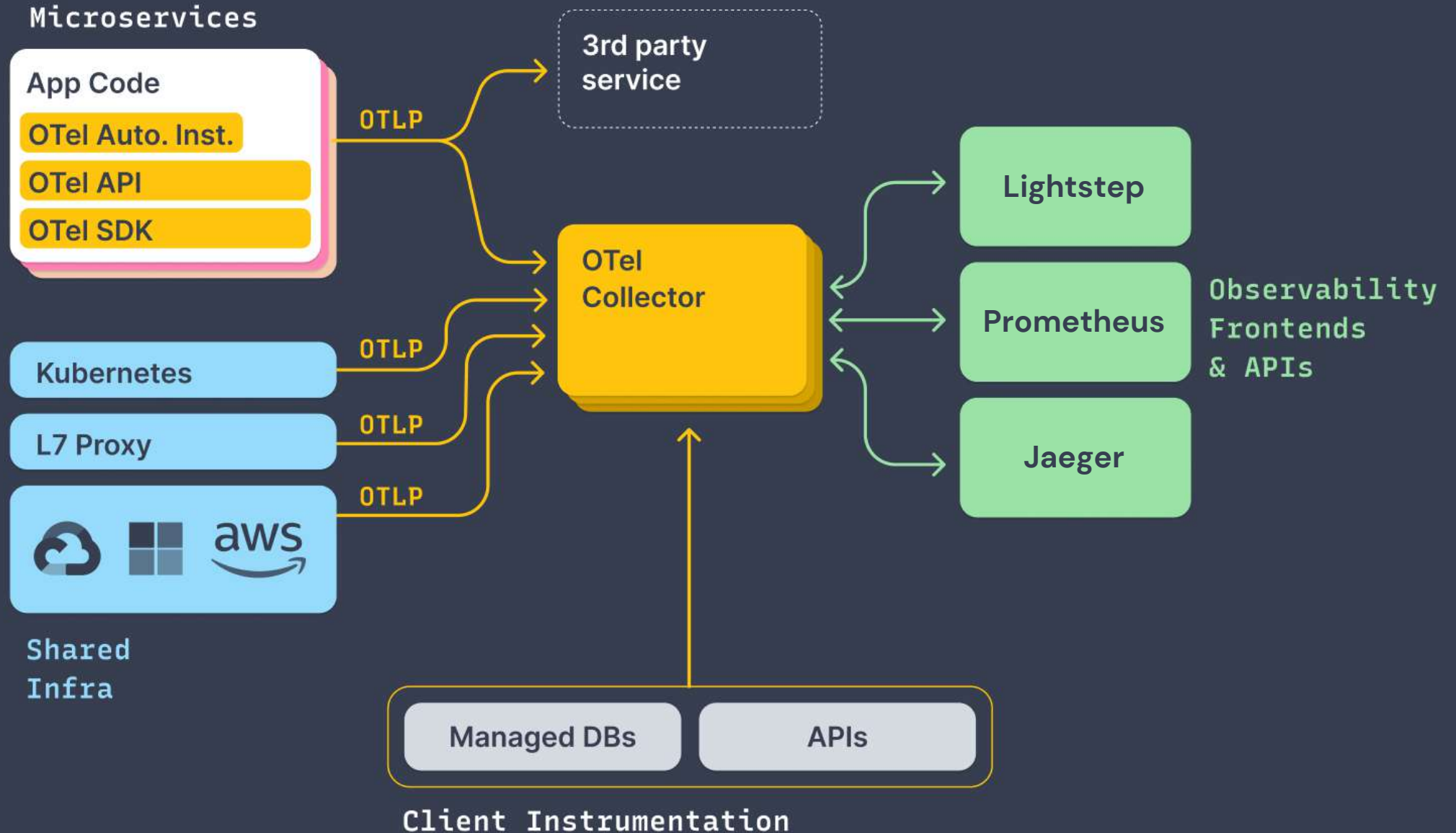
OpenTelemetry - A brief history

2015 — **Lightstep** is founded by Ben Sigelman, Daniel Spoonhower, and Ben Cronin

2016 — **Lightstep** helps to found **OpenTracing,** the first common, portable API for distributed tracing

2017 — **Jaeger**, an open-source tracing tool developed by Yuri Shkuro, is open-sourced by **Uber**

2018 — **Google** releases a new portable instrumentation project called **OpenCensus**

2019 — The **OpenTracing** and **OpenCensus** projects announce a merger, forming **OpenTelemetry** to be a part of CNCF

2020 — **OpenTelemetry** is released in beta

2nd most active project in CNCF

CNCF Projects 1/1/2020 - 1/1/2021

Lightstep

5

A performant solution, agnostic from any vendor

Benefits of using OpenTelemetry with lambdas

- Choose my observability backend (x-ray limitations)

- Store history of all calls

- Distributed traces for faster RCA

- Measure and improve performance

- Measure Golden Signals ⇔ RED metrics ⇔ Rate / Error / Delays

- Alerts on golden signals (SLI) / SLO

- Monitor my own technical/business attributes

Lightstep

How to do it?

- Enable X-Ray

- Add AWS Layer

- Add environment variable

- Add collector config

- Add environment variable again

- Update your code (optional)

Git Example

Step by Step Video

https://github.com/dimitrisfinas/opentelemetry-aws/tree/main/lambda

https://www.youtube.com/watch?v=iycLNwmpcOQ

Further readings

- OpenTelemetry documentation https://opentelemetry.io/docs/ and https://opentelemetry.lightstep.com/

- OpenTelemetry registry to get up-to-date list of supported technologies and projects: https://opentelemetry.io/registry/

- How to choose your Observability solution: https://medium.com/dzerolabs/unpacking-observability-how-to-choose-an-observability-vendor-aa0e6d80b71d