
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

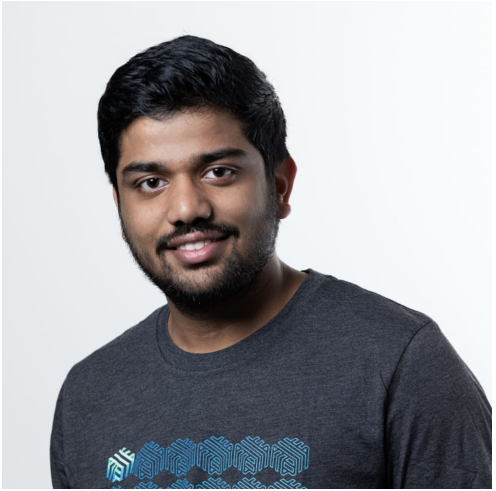
Tips and Tricks for migrating from Jaeger to OpenTelemetry

Vineeth Pothulapati

Agenda

- Pre-requisites
- Why migrate?
- Jaeger & OpenTelemetry architecture
- Levels of migration
- Jaeger and OpenTelemetry boundaries

About me



Vineeth Pothulapati

Product Manager @Timescale
(*focused on Promscale & Tobs*)

Maintainer of OpenTelemetry Operator

@vineeetth

https://github.com/timescale/promscale
https://github.com/timescale/tobs

Pre-requisites

- OpenTelemetry aka **OTel**
- This talk is focused on **traces**
- This talk isn’t intended to push the migration 🙂
- Understanding the components involved (next slide)

Tracing Components

-	Instrumentation API	Instrumentation SDK	Agent/ Collector	Storage	Visualisation
Jaeger	Opentracing	Jaeger client libraries	Jaeger	native storage options	Jaeger Query
OTel	OTel	OTel	OTel	-	-

Why migrate?

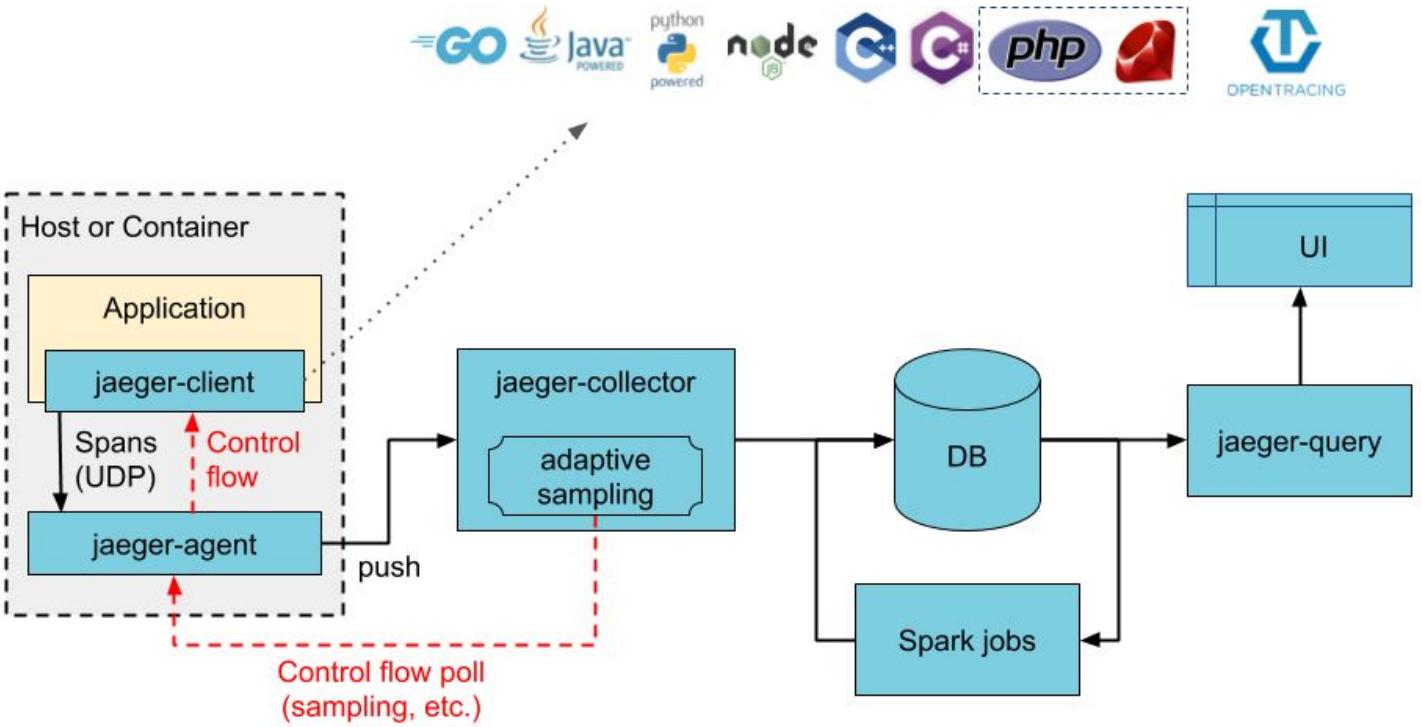
- Jaeger client libraries are deprecated in favour of OTel
- OpenTelemetry is the new standard for instrumentation, and collection of observability signals.
- Support for various processors to process the data in the collector and wide-range of integrations to receive and export the data
- Support for auto-instrumentation

Levels of migration

- Instrumentation layer
- Collector layer

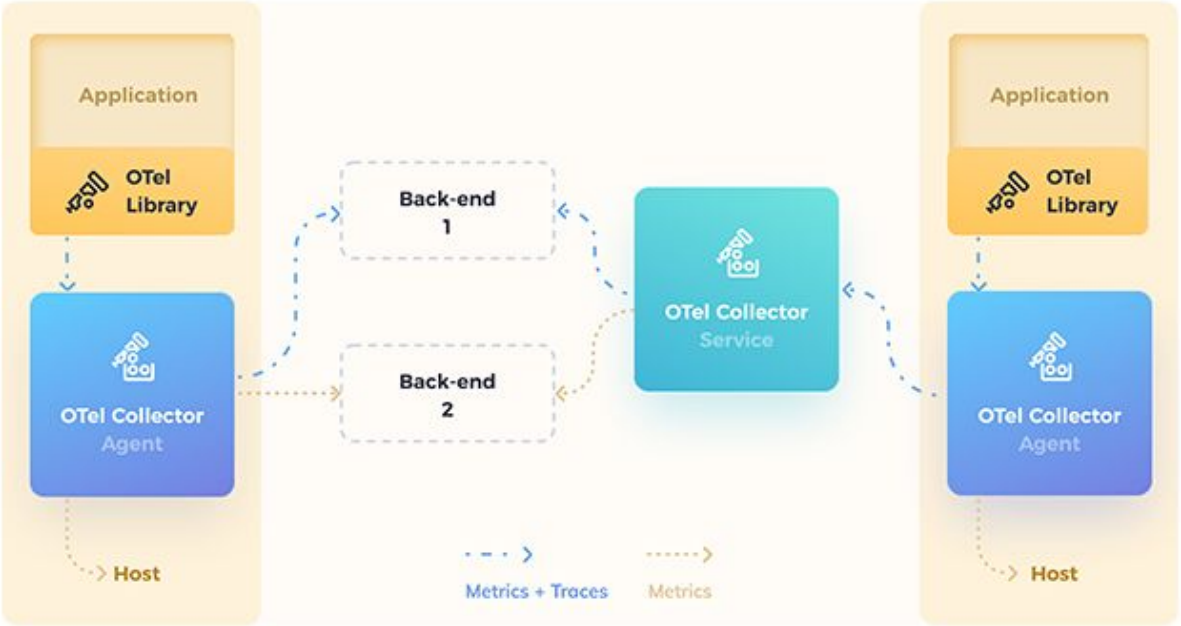
Jaeger & OpenTelemetry architectures

Jaeger Architecture

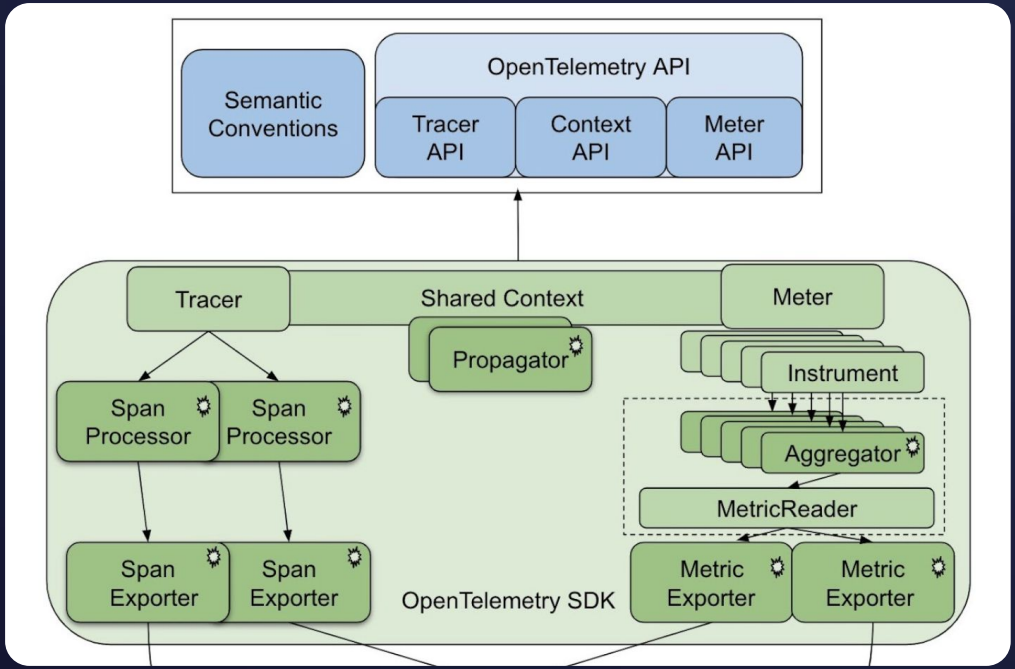


source: Architecture diagram from Jaeger docs

OpenTelemetry Architecture



source: Architecture diagram from OpenTelemetry docs



source: New Relic blog post

Instrumentation layer

Instrumentation

Migration in instrumentation layer can be done in two ways:

- Using OpenTelemetry shim
- Complete re-instrumentation

Using OpenTelemetry shim

- shim: consists of a set of classes which implement the OpenTracing API while using OpenTelemetry constructs behind the scenes

- Its purpose is to allow applications which are already instrumented using OpenTracing to start using OpenTelemetry with a minimal effort, without having to rewrite large portions of the codebase.

https://medium.com/jaegertracing/migrating-from-jaeger-client-to-opentelemetry-sdk-bd337d796759

Complete re-instrumentation

- Complete instrumentation offers complete OTel as a package
- Supports metrics and logs
- Improved and performant API & SDK
- Easy integration with auto-instrumented applications

KubeCon | CloudNativeCon
North America 2022

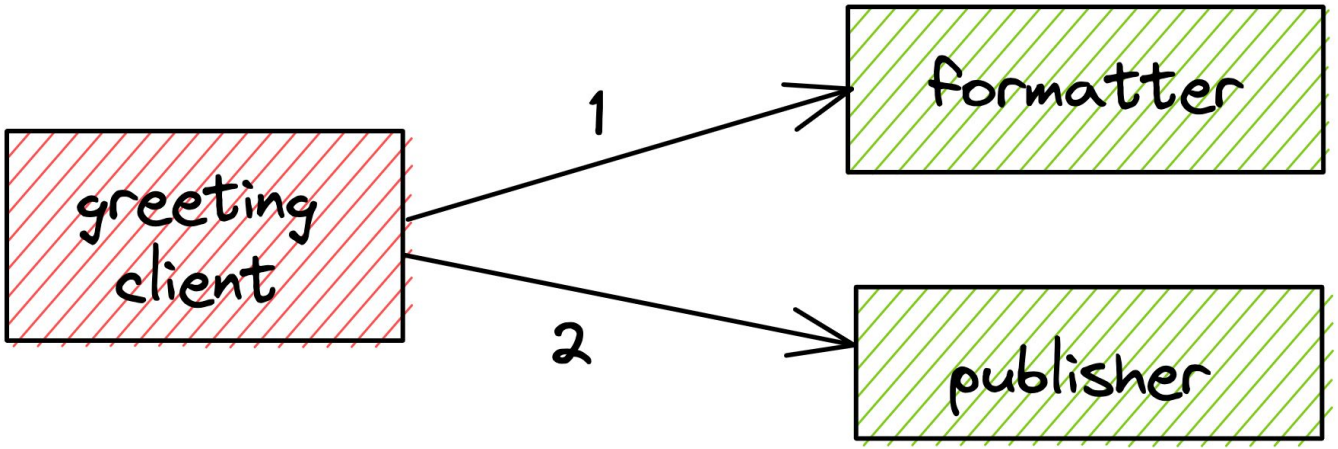
BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Demo App

Migrating from Jaeger to OpenTelemetry

(using re-instrumentation)



https://github.com/VineethReddy02/tracing-instrumentation-demo

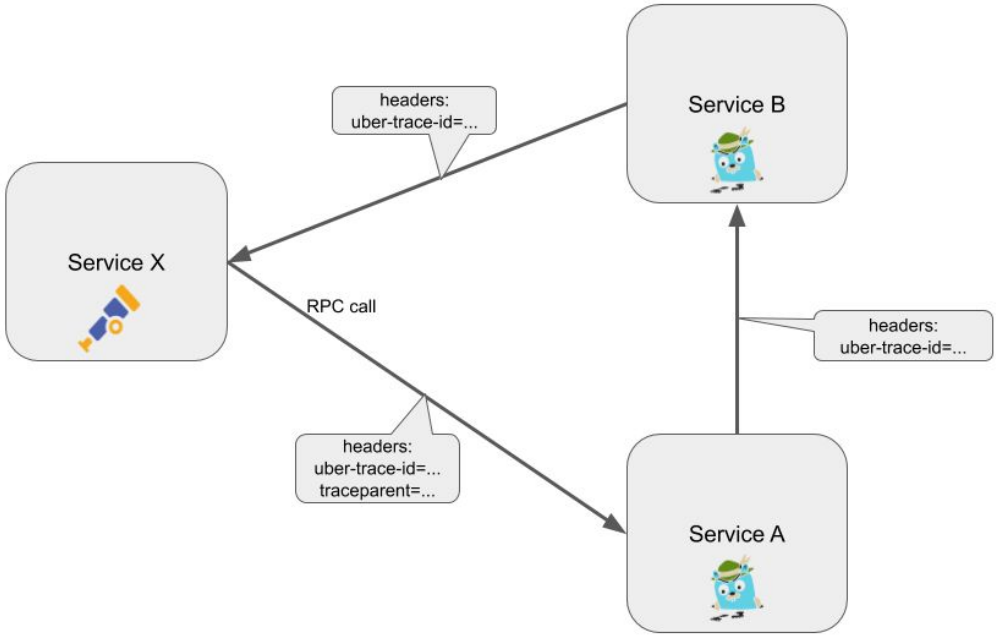
Context Propagation

Jaeger

- Jaeger
- B3
- W3C

OpenTelemetry

- Jaeger
- B3
- W3C



https://medium.com/jaegertracing/jaeger-clients-and-w3c-trace-context-c2ce1b9dc390

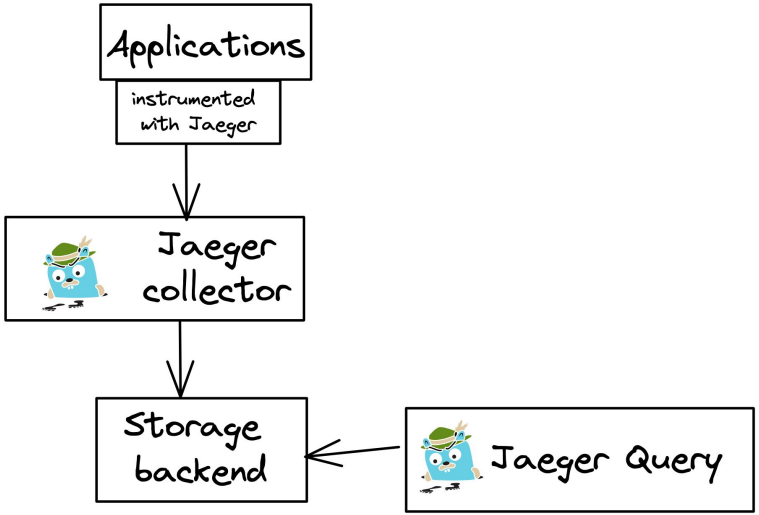
Impact

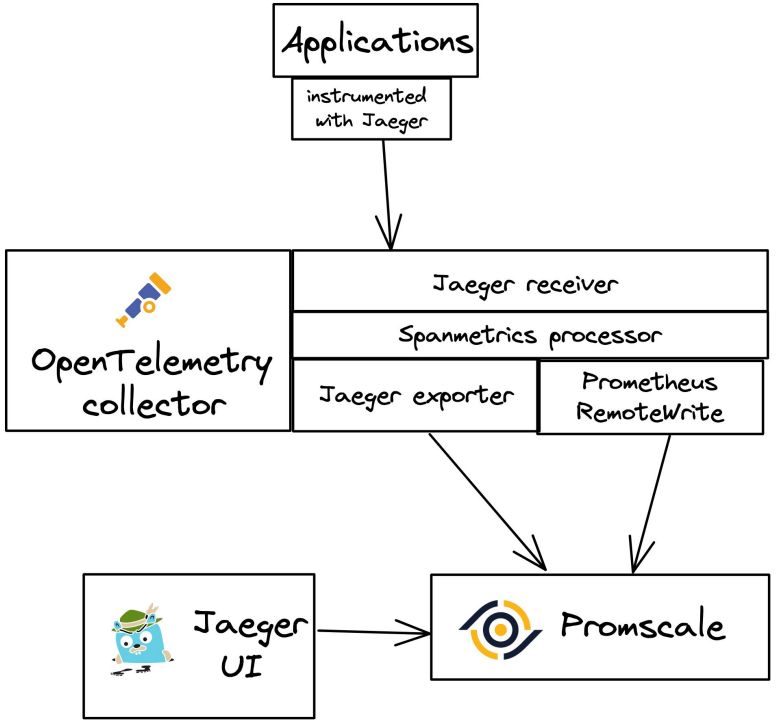
- Improved tracer implementation
- Switch to the OpenTelemetry SDK while continuing to use your existing OpenTracing instrumentation
- Improved performance
- Access to OpenTelemetry's framework plugins

Migrating Collector layer

Collector

You can replace Jaeger collector with OpenTelemetry collector





Difference btw Jaeger & Otel collectors

Jaeger Collector

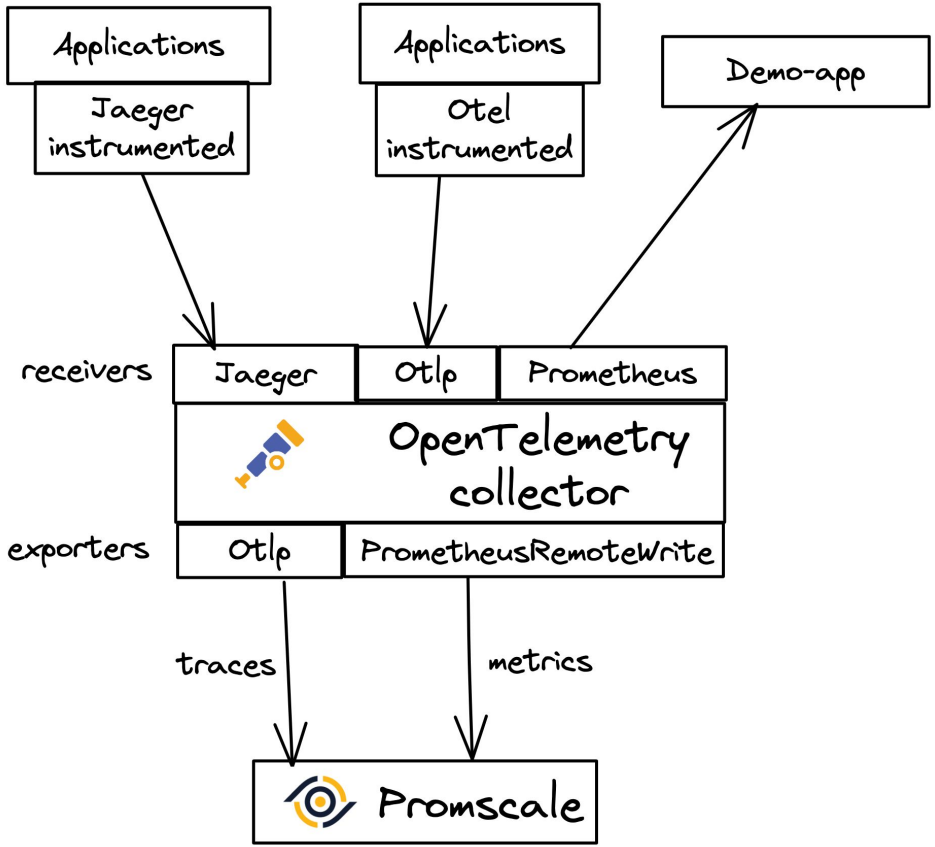
- Flexible sampling
- Matured storage options (native support for In-memory, Cassandra, ES, etc...)

OpenTelemetry Collector

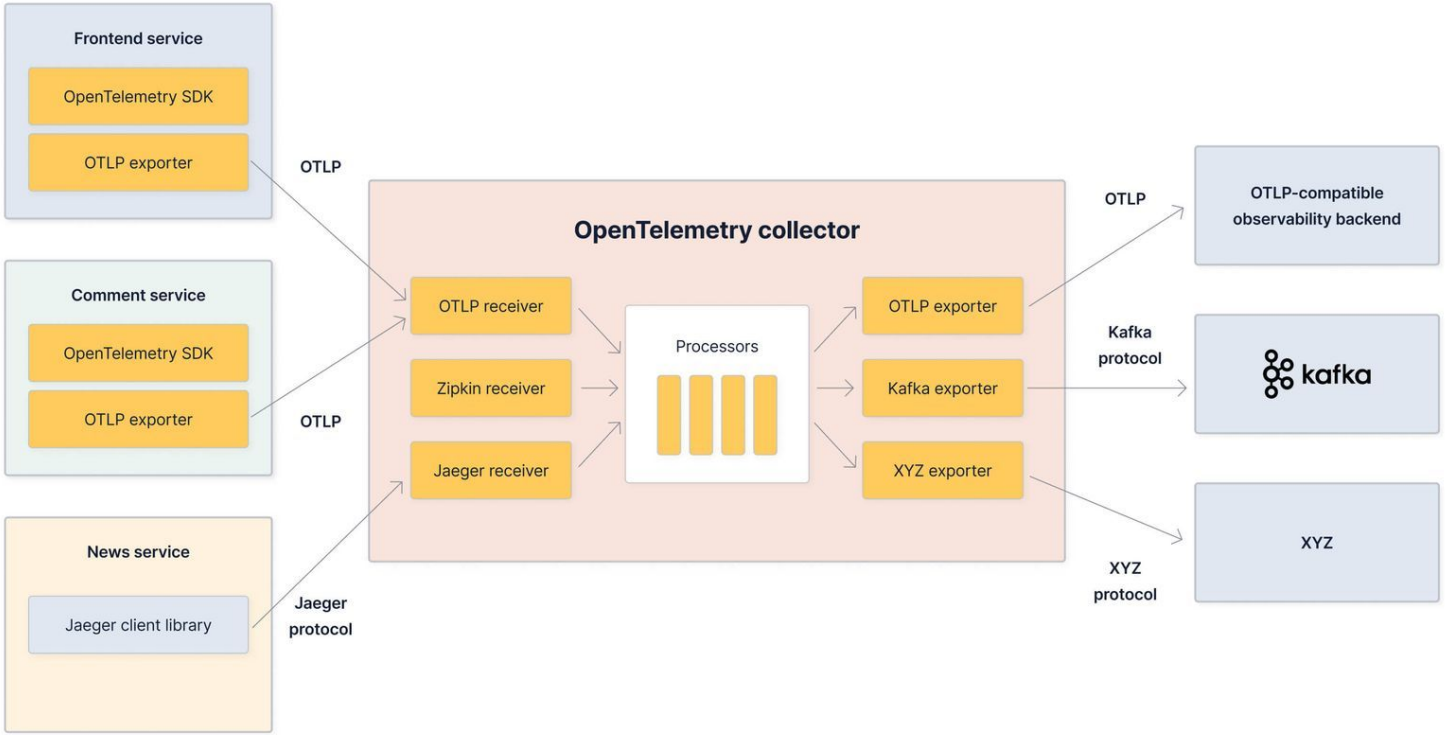
- Receivers and Exporters
- Processors
- Support for metrics and logs
- Extensions

OTel Collector Config


```yaml
receivers:
  jaeger:
    protocols:
      grpc:
        endpoint: 0.0.0.0:14250
      thrift_http:
        endpoint: 0.0.0.0:14268
  otlp:
    protocols:
      grpc:
        endpoint: 0.0.0.0:4317
  prometheus:
    config:
      scrape_configs:
      - job_name: opentelemetry-collector
        scrape_interval: 10s
  zipkin:
    endpoint: 0.0.0.0:9411
exporters:
  otlp:
    endpoint: "tobs-promscale.default.svc:9202"
  prometheusremotewrite:
    endpoint: "http://tobs-promscale.default.svc:9201/write"
service:
  pipelines:
    traces:
      receivers:
      - otlp
      - jaeger
      - zipkin
      exporters:
      - otlp
    metrics:
      receivers:
      - prometheus
      exporters:
      - prometheusremotewrite
```

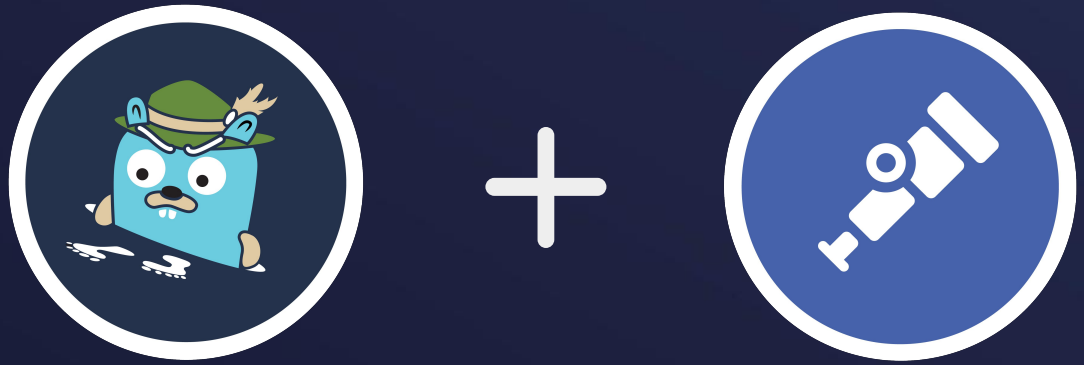


Anatomy of OTel collector



Impact

- Migrating the collector moves the complete data processing and storage backend away from Jaeger
- Configure pipelines to receive and send data from multiple sources to destinations
- Vendor neutral processing system, you can seamlessly migrate from one vendor to another by changing the collector config

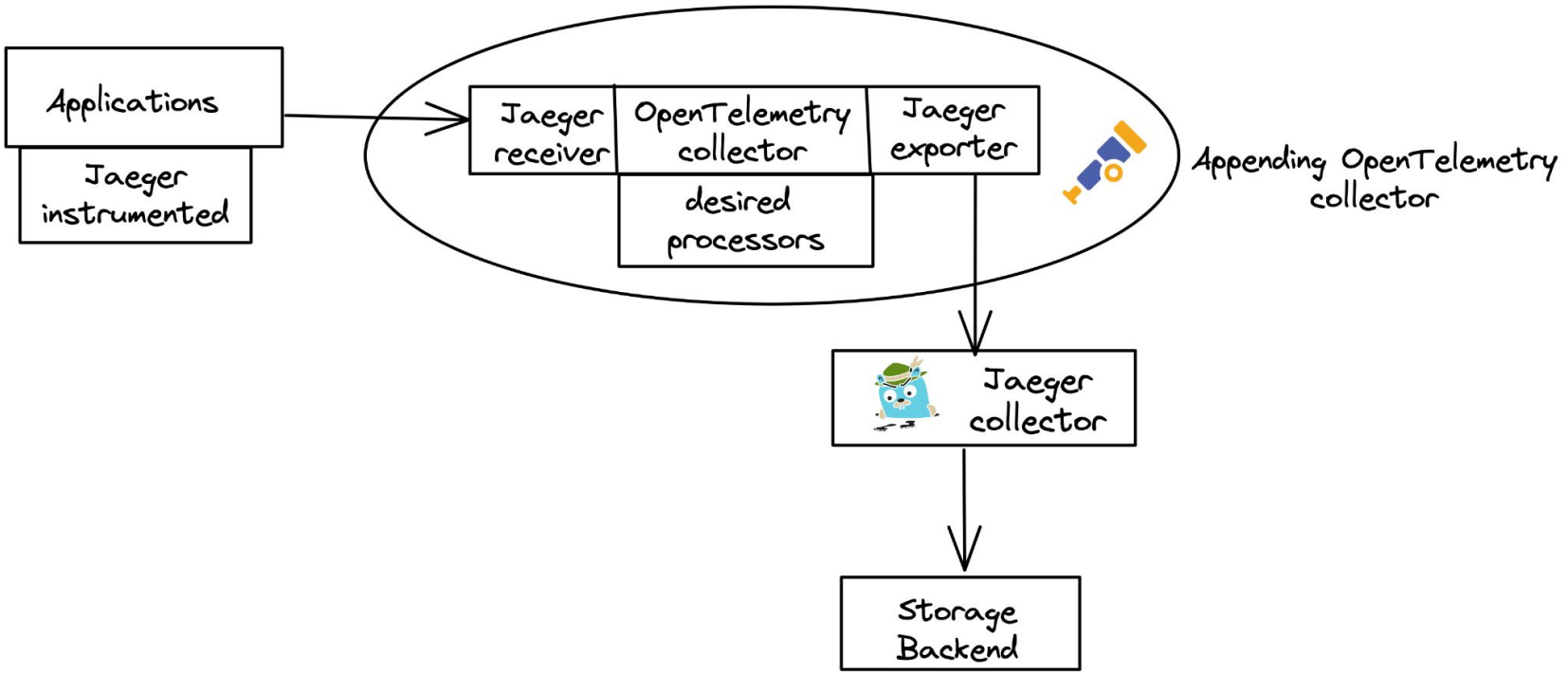


Add OpenTelemetry collector to Jaeger deployment

Why OTel In Jaeger?

- Receive data from multiple sources
- Process data using wide range of data processing tools available in collector
- Export data to multiple backends/vendors
- Support for metrics and logs alongside traces to receive, process and export the observability signals

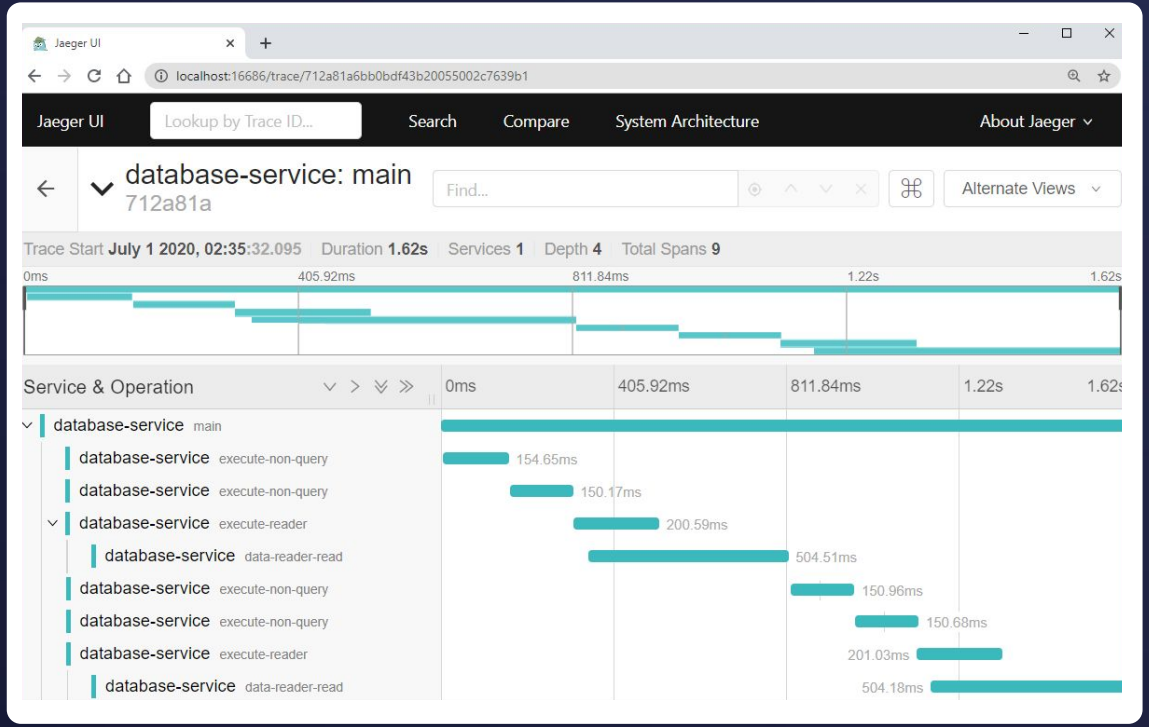
Otel collector in Jaeger ingestion flow



Impact

- Leverage best in both worlds by using both OpenTelemetry and Jaeger collectors
- Jaeger as a project is becoming more like an tracing platform that offers storage, querying and visualising of traces
- Using Jaeger offers native support for Cassandra, Elastic search, Badger and In-memory storage systems.
- Jaeger exposes an gRPC based remote write integration, this allows you to plug desired backend to store traces.
  example: Promscale

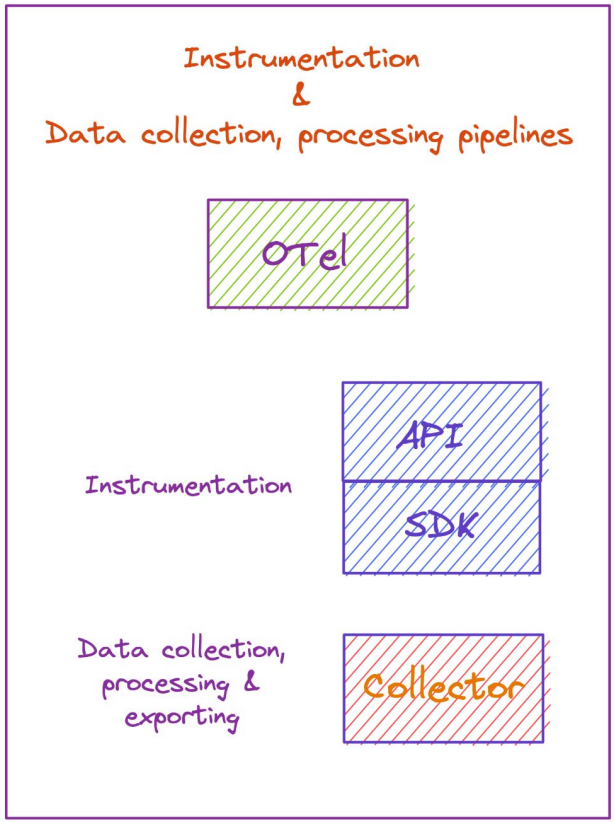
Querying & Visualizing Traces

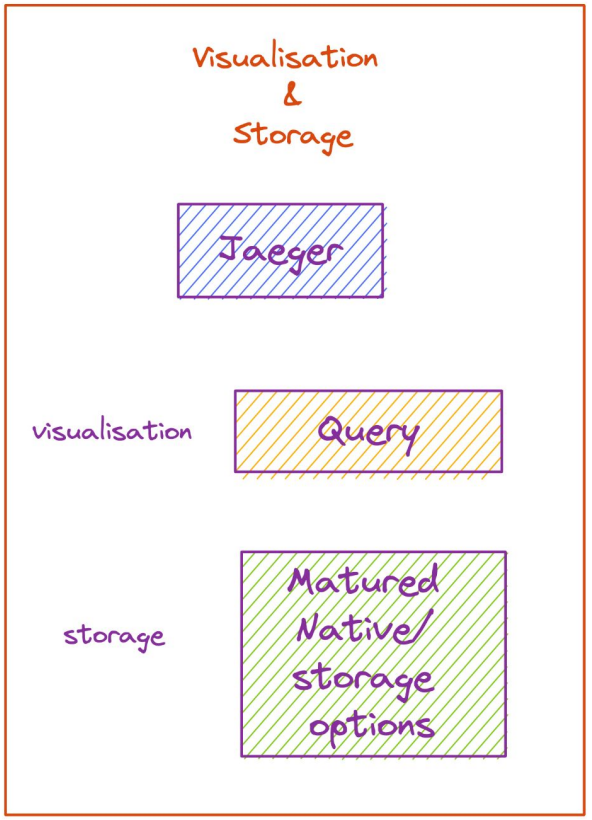


Querying

- With Jaeger you can query the traces using Jaeger UI
- With OpenTelemetry there is no native querying support, this is left to the storage offerings.
- You can use Jaeger UI/Query component with the compatible storage backends.
  example: Promscale & Elasticsearch

OTel-Jaeger boundaries





Conclusion: Start with OTel in some capacity

OTel Instrumentation

Migrating the instrumentation layer is definitely an ideal option. However, you should instrument new apps using OTel.

Introduce OTel collector

Migrate the data processing layer into OTel, to leverage receivers, processors, exporters and extensions. Helps in unifying the o11y data. You can also use Jaeger collector with OTel

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Thank you!

Questions?