



Philipp Krenn

@xeraa

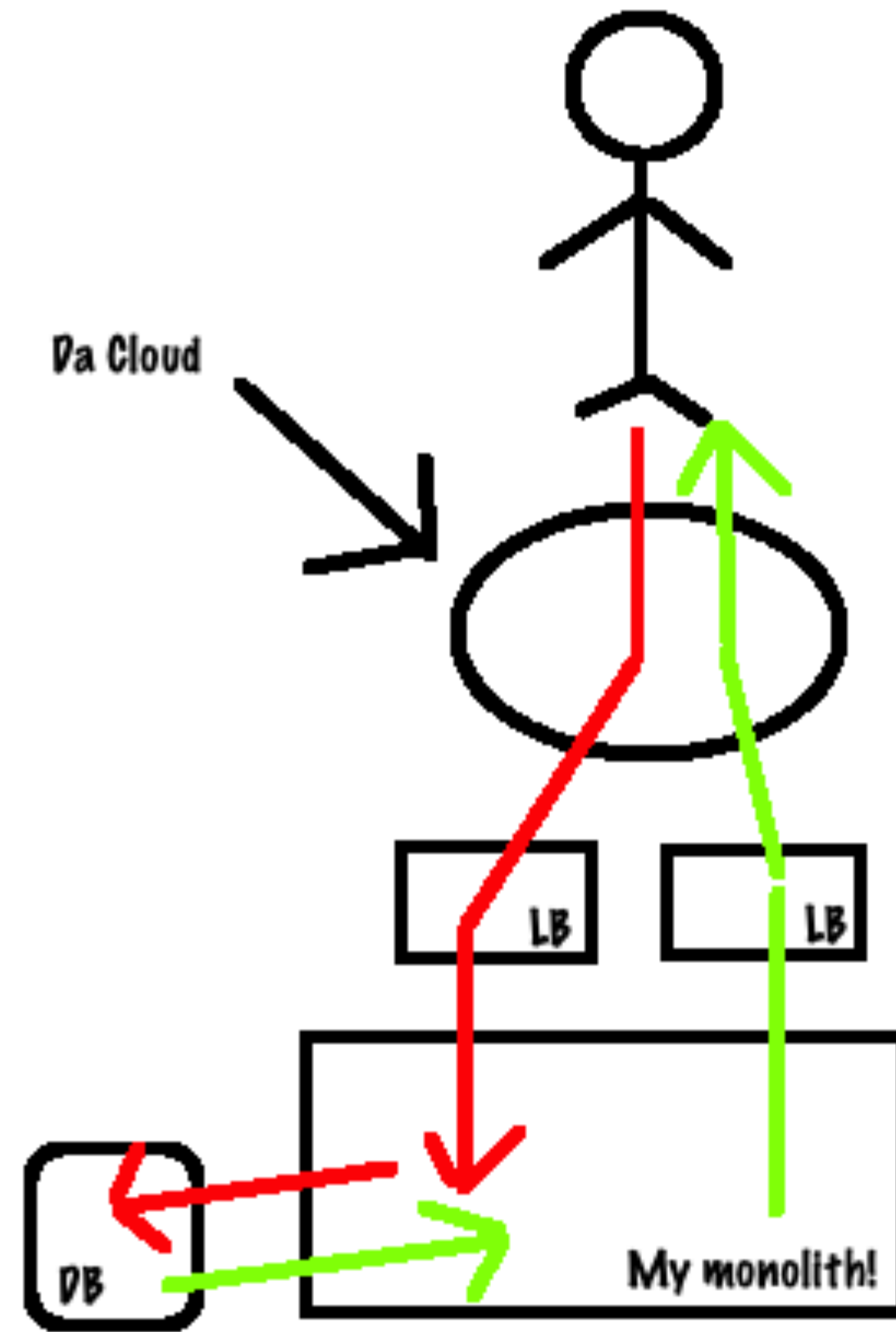
Who is **using** OpenTelemetry?

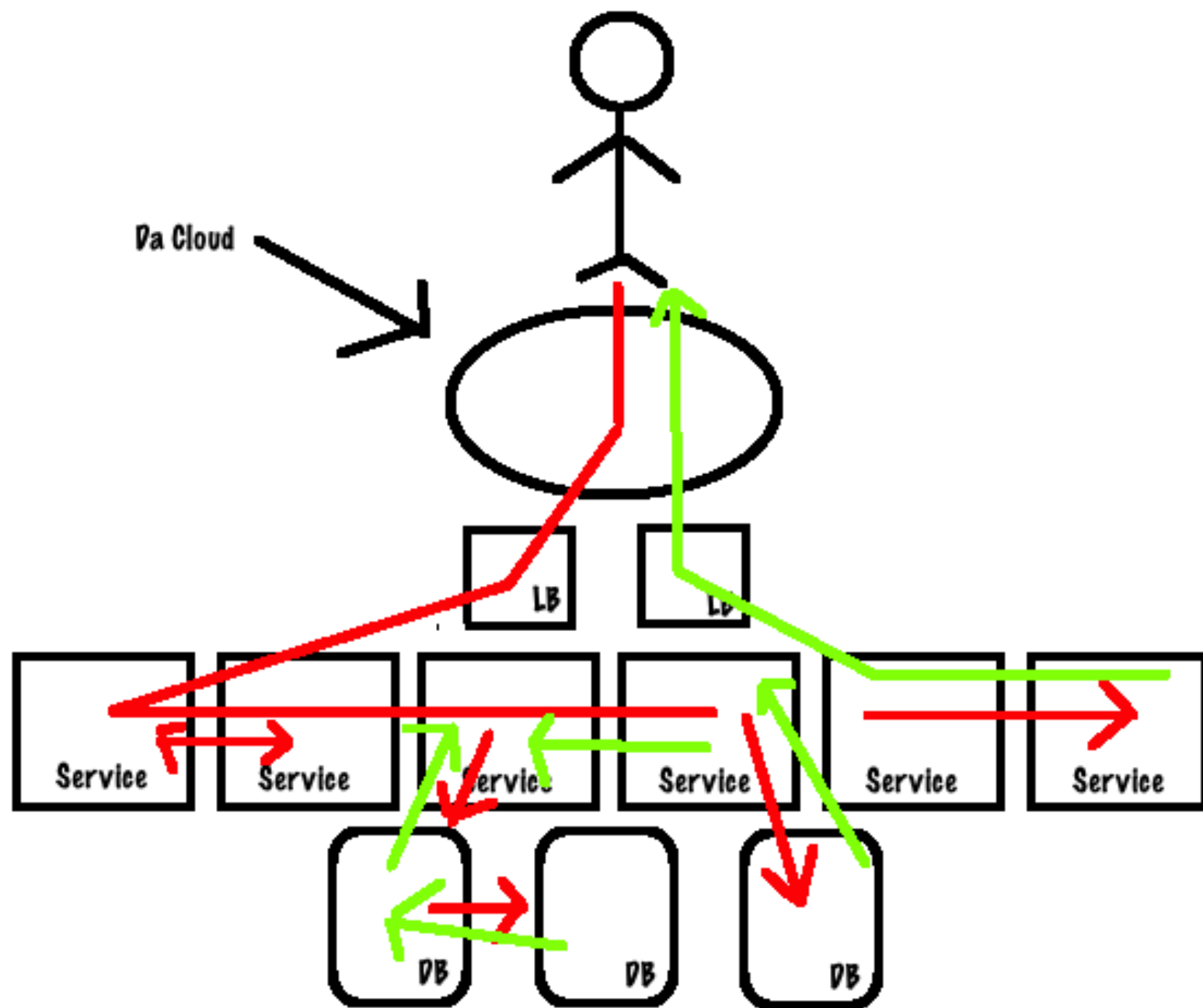
Who is using OpenTelemetry **in** **production?**

Why do we even need it?

Simpler Times

<https://www.kartar.net/2019/07/intro-to-distributed-tracing/>





Better Times for Vendors

<https://www.kartar.net/2019/07/intro-to-distributed-tracing/>

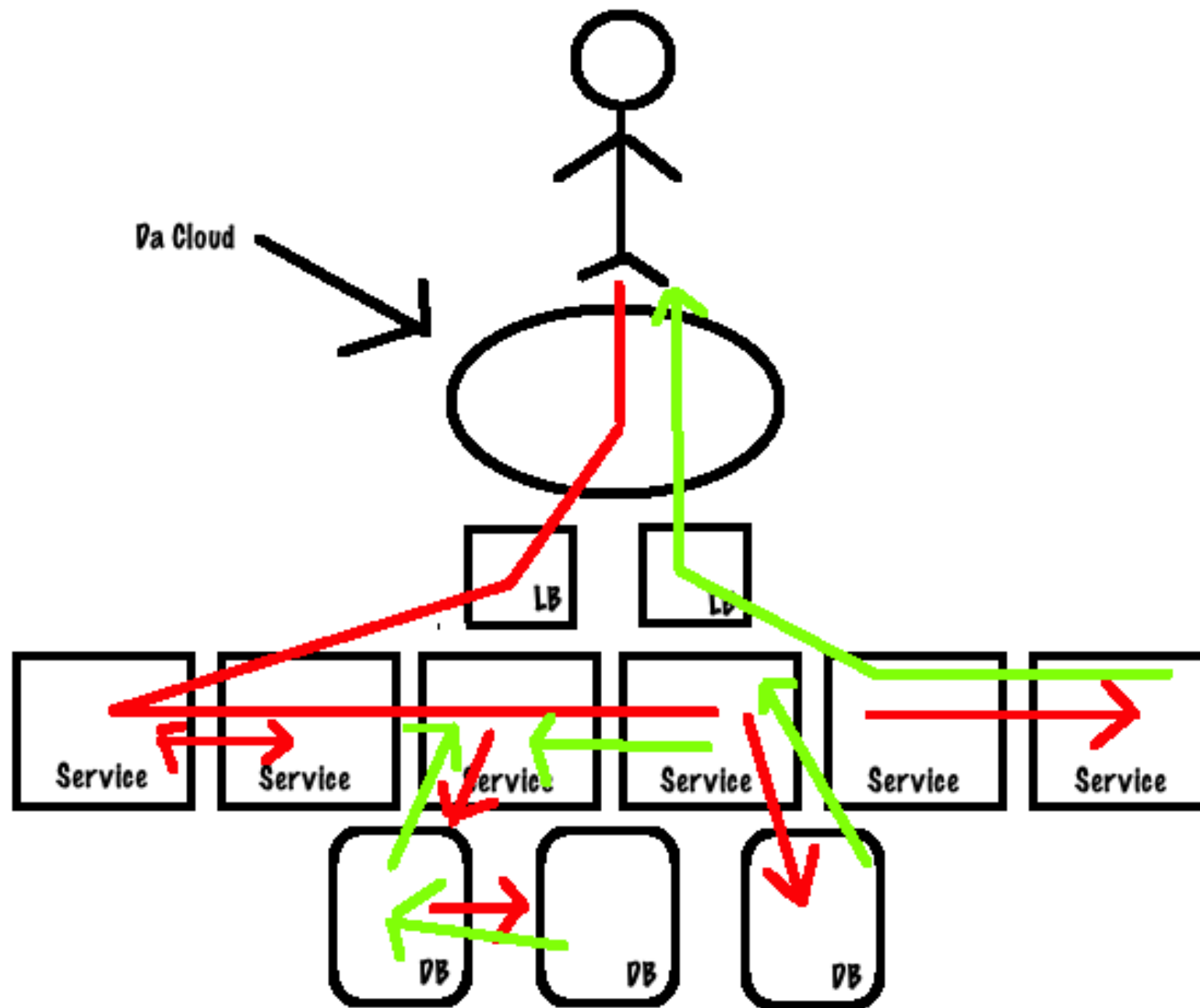


elastic

Developer



Better Times ~~for Vendors~~ with Standards



Agenda

**What is OpenTelemetry?
How do I make it bootiful?**

***What is
OTel?***

OpenTelemetry is a collection of tools, APIs, and SDKs. Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software's performance and behavior.

– **<https://opentelemetry.io>**

Metrics

Statistical information in aggregate

Logs

Timestamped record of (structured) events

Traces

Request scoped context

Open source – Apache 2.0

CNCF incubating

**Merger of OpenTracing &
OpenCensus**

OpenTracing Solution to

uber-trace-id:

118c6c15301b9b3b3 : 56e66177e6e55a91 :

18c6c15301b9b3b3 : 1

elastic-apm-traceparent:

00-f109f092a7d869fb4615784bacefcfd7-

5bf936f4fcde3af0-01

OpenCensus

👍 **SDK**

👎 **Google**

OpenTelemetry

Vendor neutral with wide support:
<https://opentelemetry.io/vendors/>

Company	Distribution	Native OTLP	Learn more
AppDynamics	Yes	Yes	https://docs.appdynamics.com/latest/en/application-monitoring/appdynamics-for-opentelemetry
Aspecto	Yes	Yes	https://www.aspecto.io
AWS	Yes	No	https://aws-otel.github.io
Azure	Yes	No	https://docs.microsoft.com/azure/azure-monitor/app/opentelemetry-overview
Datadog	Yes	Yes	https://docs.datadoghq.com/tracing/setup_overview/open_standards
Dynatrace	Yes	Yes	https://www.dynatrace.com/support/help/how-to-use-dynatrace/transactions-and-services/service-monitoring-settings/opentelemetry/
Elastic	Yes	Yes	https://www.elastic.co/guide/en/apm/get-started/current/open-telemetry-elastic.html
F5	No	Yes	https://github.com/open-telemetry/opentelemetry-collector-contrib/tree/main/exporter/f5cloudexporter
Grafana Labs	No	Yes	https://grafana.com/oss/opentelemetry/
Honeycomb	Yes	Yes	https://docs.honeycomb.io/getting-data-in/
Instana	No	Yes	https://www.ibm.com/docs/en/obi/current?topic=apis-opentelemetry
Lightstep	Yes	Yes	https://github.com/lightstep?q=launcher
LogicMonitor	Yes	Yes	https://www.logicmonitor.com/support/tracing/getting-started-with-tracing
Logz.io	Yes	No	https://docs.logz.io/shipping/tracing-sources/opentelemetry.html#overview

Three Ways to **OpenTelemetry Integration**

Vendor integration on the OpenTelemetry application agent

Observability Vendor

Vendor Protocol

Vendor Protocol

My Application

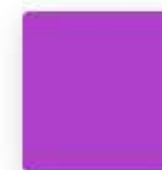


Agent

Vendor
Exporter



Legend

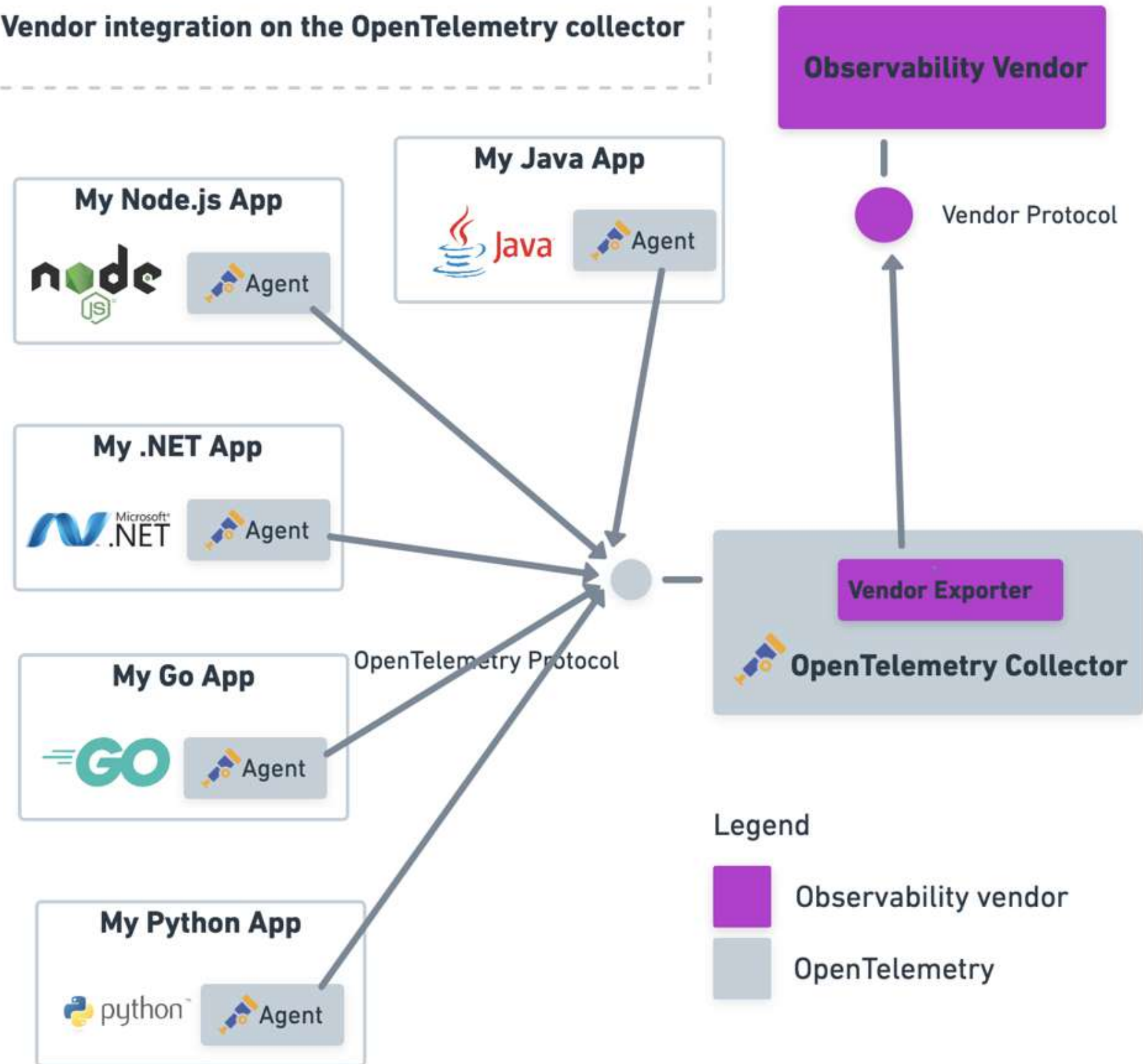


Observability vendor

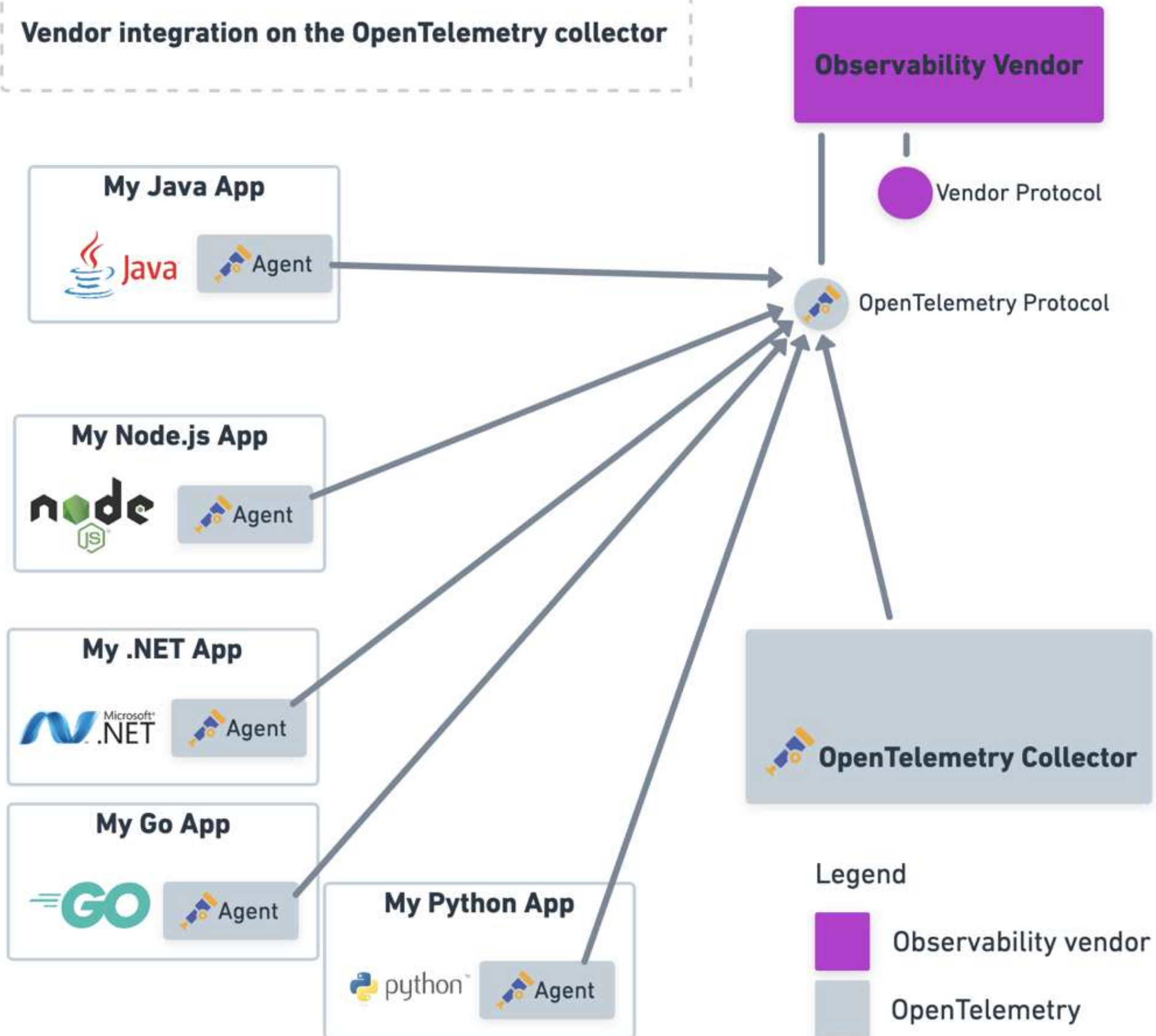


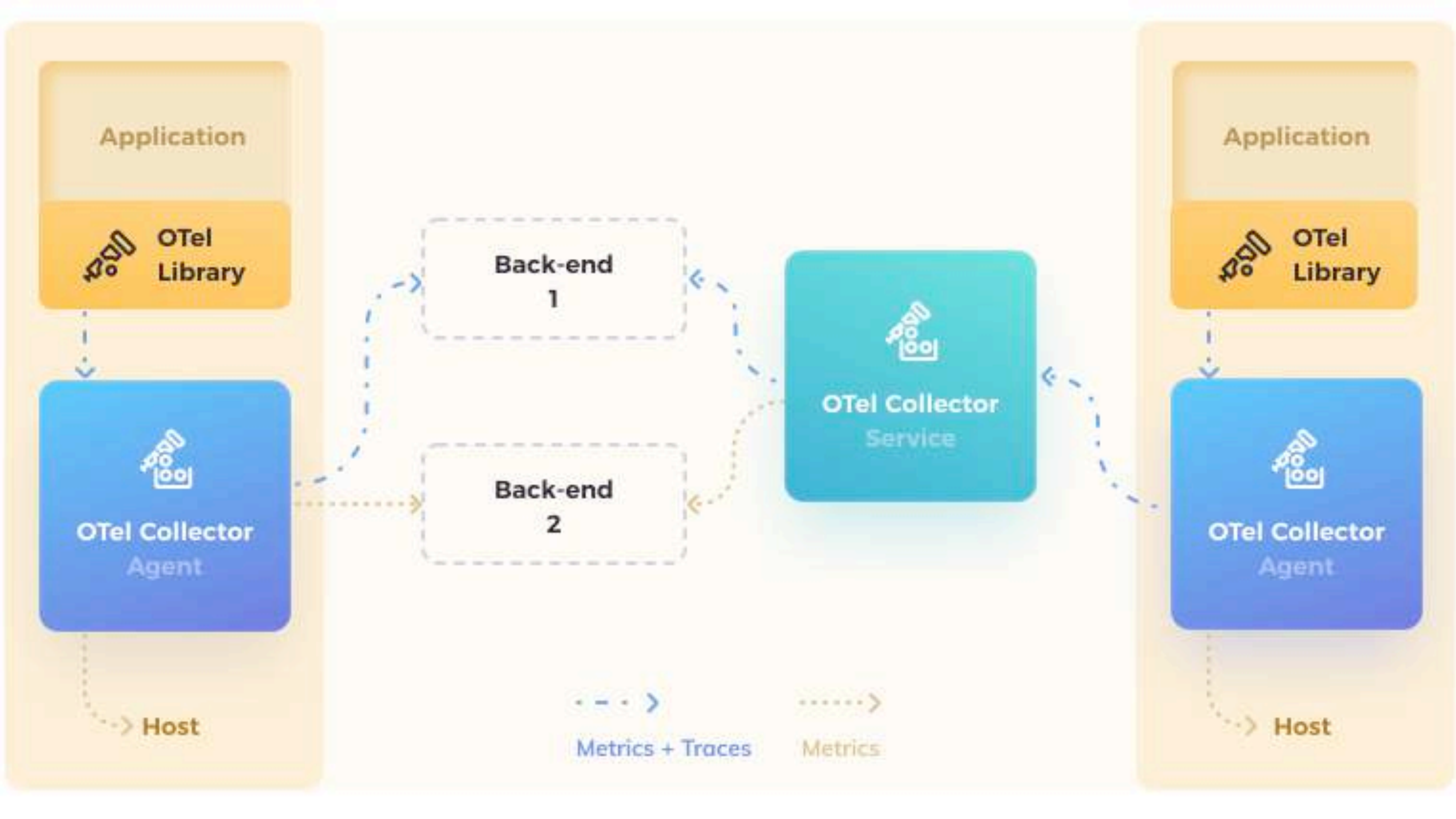
OpenTelemetry

Vendor integration on the OpenTelemetry collector



Vendor integration on the OpenTelemetry collector





Status

Tracing: stable

Metrics: stable / prototyping

Logs: draft

<https://opentelemetry.io/status/>

Metrics Terminology

counter measure observer

Tracing Terminology

Trace: record of a request, DAG

Span: named & timed single operation,
optionally nested

Tracing Terminology

Span Context: IDing a request (span + trace)

Context Propagation: Attaching IDs

Questions?

***Make It
Bootiful***

Auto Instrumentation

Live in / attach to app process

```
public static void main  
    premain  
    agentmain
```

<https://www.javaadvent.com/2019/12/a-beginners-guide-to-java-agents.html>

Java Agent Setup

```
export OTEL_RESOURCE_ATTRIBUTES=service.name=frontend,  
                                           service.version=1.0  
                                           deployment.environment=production
```

```
export OTEL_EXPORTER_OTLP_ENDPOINT=https://otlp_url  
export OTEL_EXPORTER_OTLP_HEADERS="Authorization=Bearer <secret_token>"  
java -javaagent:/path/to/opentelemetry-javaagent-all.jar \  
     -classpath lib/*:classes/ \  
     com.mycompany.checkout.CheckoutServiceServer
```

Auto Instrumentation Support

**Spring Batch, Data, Integration, Scheduling,
Web MVC, Web Services, WebFlux,...**

<https://github.com/open-telemetry/opentelemetry-java-instrumentation/blob/main/docs/supported-libraries.md>

Manual Instrumentation

```
Span span = tracer.spanBuilder("my span").startSpan();  
// Put the span into the current Context  
try (Scope scope = span.makeCurrent()) {  
    // Do something  
} catch (Throwable t) {  
    span.setStatus(StatusCode.ERROR, "Something bad happened!");  
    throw t;  
} finally {  
    span.end(); // Cannot set a span after this call  
}
```

Manual Instrumentation

```
Span span = tracer.spanBuilder("/resource/path").setSpanKind(SpanKind.CLIENT).startSpan();  
span.setAttribute("http.method", "GET");  
span.setAttribute("http.url", url.toString());
```

Auto vs Manual Instrumentation

Auto: Code unchanged, handled by runtime,
less execution visibility

Manual: Code changed, handled by
application, more execution visibility

A medium shot of actor Ryan Reynolds. He is wearing a dark grey suit, a white shirt, and a dark tie. He has a serious expression and is looking slightly to his left. The background is a vibrant purple with a pattern of vertical lines and scattered small, colorful dots in shades of blue, pink, and white. Overlaid at the bottom of the image is the text "WHY NOT BOTH?!?" in a large, white, bold, sans-serif font.

WHY NOT BOTH?!?

Apps instrumented using
OpenTelemetry agents



OpenTelemetry Collectors



*Elastic
Protocol*

*OpenTelemetry
Protocol*

 Elastic Observability



Kibana Observability Apps



APM Server



Elasticsearch

Demo

**[https://github.com/open-telemetry/
opentelemetry-java-instrumentation](https://github.com/open-telemetry/opentelemetry-java-instrumentation)**

**[https://github.com/cyrille-leclerc/
my-shopping-cart](https://github.com/cyrille-leclerc/my-shopping-cart)**



Observability

APM

Services

Service map

Settings

Anomaly detection

Alerts and rules

Add data

Service Map

Environment

All



☒ Comparison

Day before



Last 24 hours

Refresh

frontend

Latency (avg.) 254 ms

Throughput (avg.) 31.0 tpm

Failed transaction rate (avg.) 1.7%

Service Details

Focus map



frontend



checkout-service



anti-fraud-service



postgresql



redis



unknown

OTel Collector Config 1/2

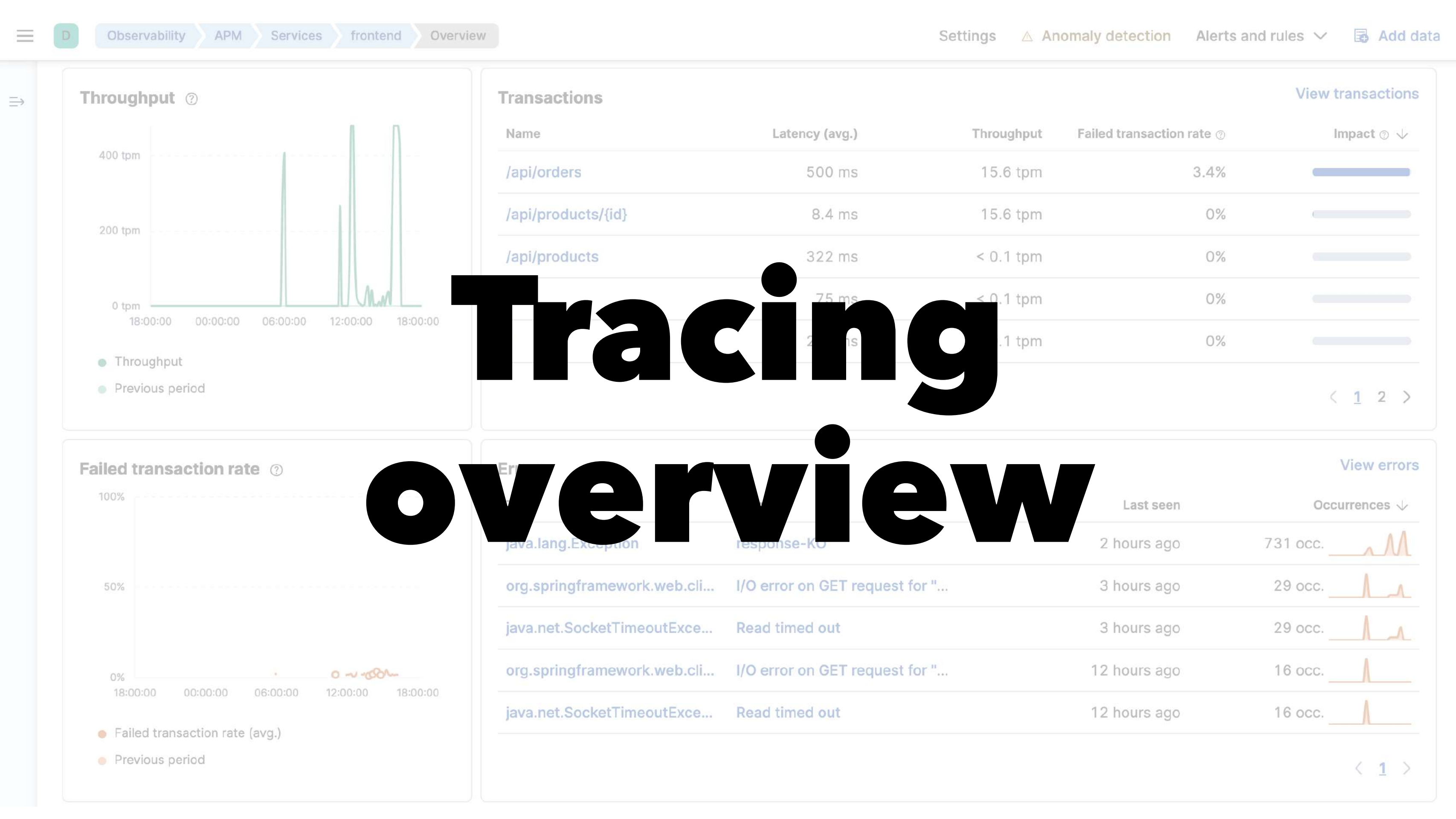
```
# Push or pull based
receivers:
  otlp:
    protocols:
      grpc:
        endpoint: '127.0.0.1:4317'

processors:
  batch: ~

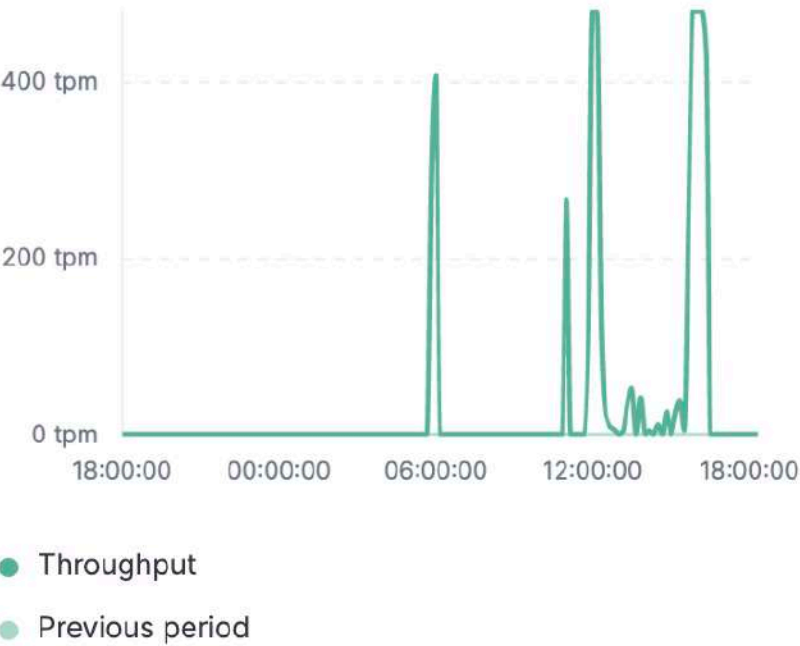
# Push or pull based
exporters:
  logging:
  otlp/elastic:
    endpoint: "..."
    headers:
      Authorization: "Bearer ..."
```

OTel Collector Config 2/2

```
service:  
  pipelines:  
    metrics:  
      receivers:  
        - otlp  
      exporters:  
        - otlp/elastic  
    traces:  
      receivers:  
        - otlp  
      processors:  
        - batch  
      exporters:  
        - otlp/elastic
```



Throughput ?



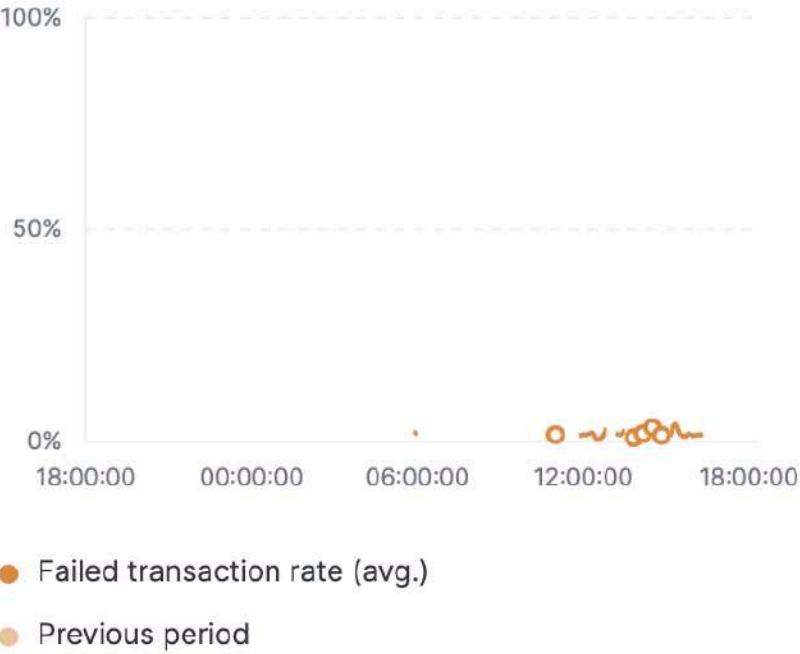
Transactions

[View transactions](#)

Name	Latency (avg.)	Throughput	Failed transaction rate ?	Impact ? ↓
/api/orders	500 ms	15.6 tpm	3.4%	
/api/products/{id}	8.4 ms	15.6 tpm	0%	
/api/products	322 ms	< 0.1 tpm	0%	
/**	75 ms	< 0.1 tpm	0%	
/	211 ms	< 0.1 tpm	0%	

< 1 2 >

Failed transaction rate ?



Errors

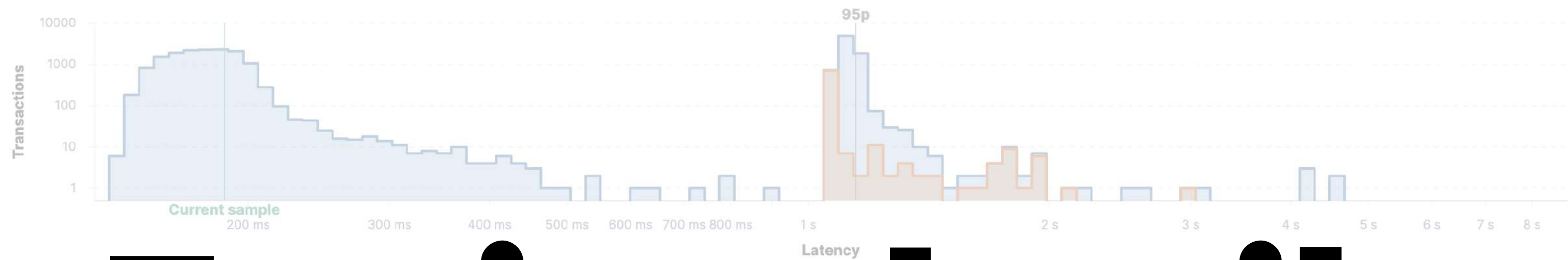
[View errors](#)

Type	Name	Last seen	Occurrences ↓
java.lang.Exception	response-KO	2 hours ago	731 occ.
org.springframework.web.cli...	I/O error on GET request for "...	3 hours ago	29 occ.
java.net.SocketTimeoutExce...	Read timed out	3 hours ago	29 occ.
org.springframework.web.cli...	I/O error on GET request for "...	12 hours ago	16 occ.
java.net.SocketTimeoutExce...	Read timed out	12 hours ago	16 occ.

< 1 >

Latency distribution ⓘ

ⓘ Click and drag to select a range



Trace sample

2 hours ago 187 ms (100% of trace)

POST http://localhost:8080/api/orders 200

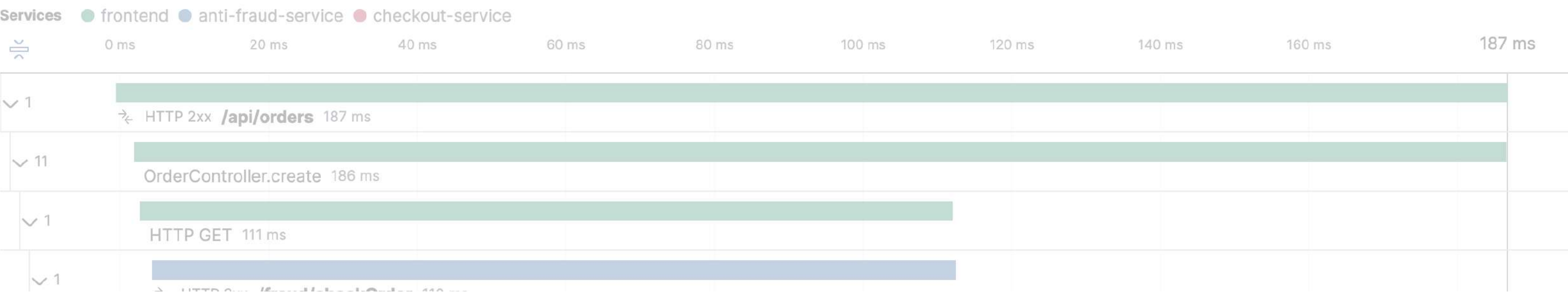
Java (0.3.)

View full trace

Timeline

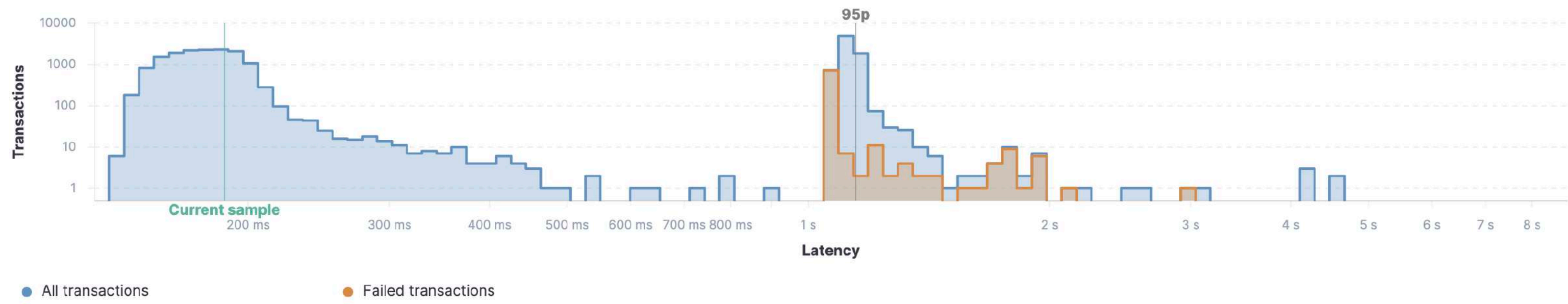
Metadata

Logs



Latency distribution

Click and drag to select a range



Trace sample

1 of 500

Investigate

View full trace

2 hours ago

187 ms (100% of trace)

POST http://localhost:8080/api/orders

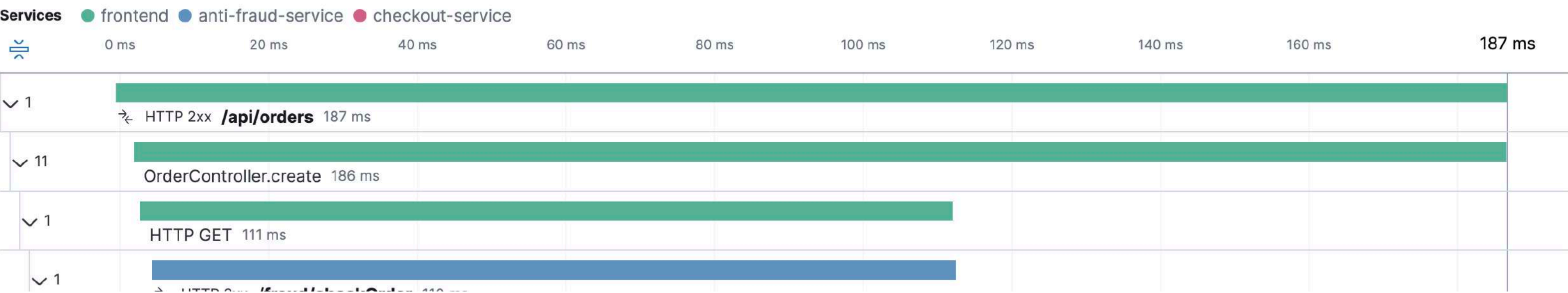
201 Created

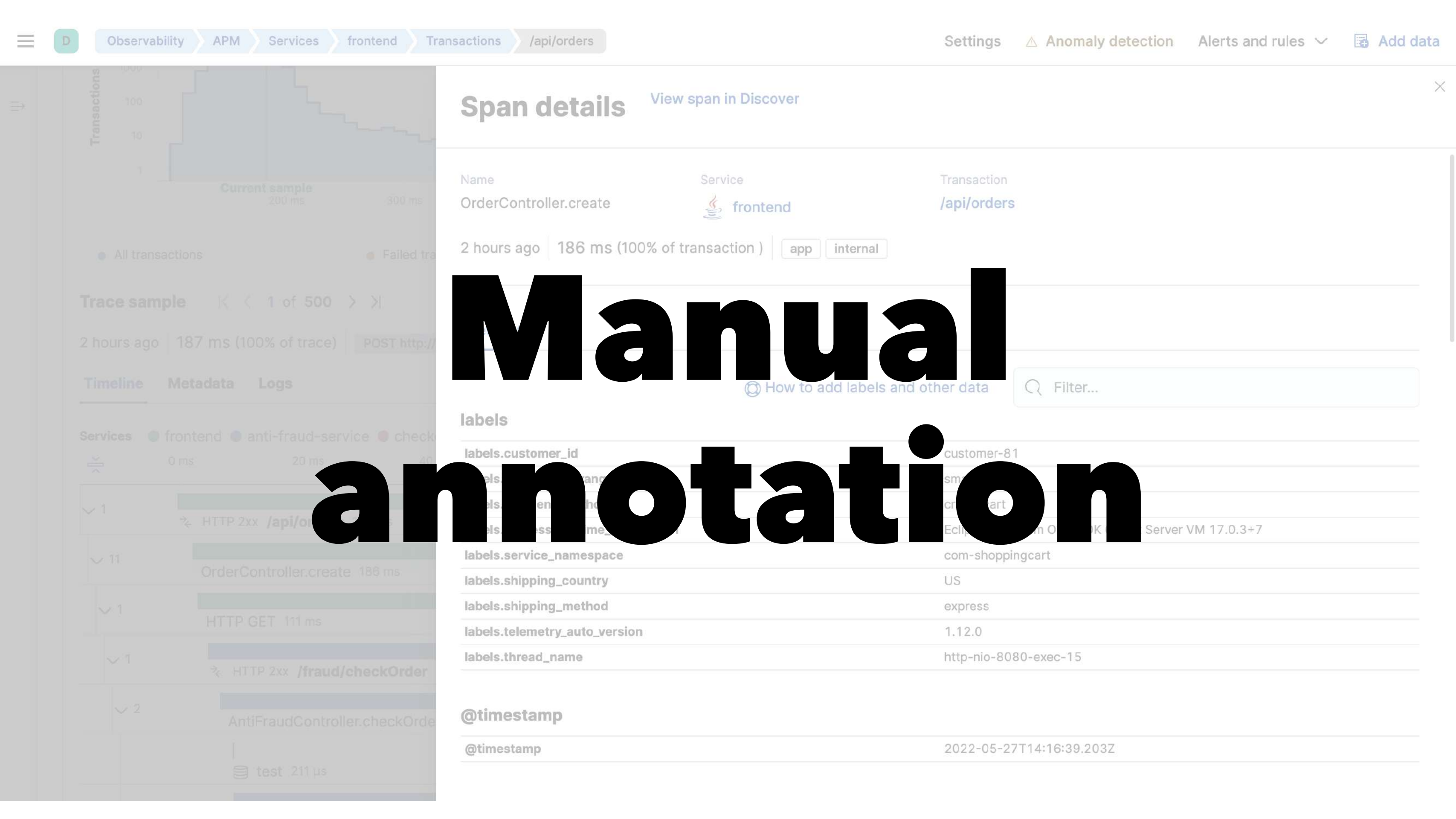
Java (0.3.)

Timeline

Metadata

Logs





D

Observability

APM

Services

frontend

Transactions

/api/orders

Settings

Anomaly detection

Alerts and rules

Add data

Transactions

1000

100

10

1

Current sample

200 ms

300 ms

All transactions

Failed tra

Trace sample

1 of 500

2 hours ago

187 ms (100% of trace)

POST http://

Timeline

Metadata

Logs

Services

frontend

anti-fraud-service

check

0 ms

20 ms

40

1

HTTP 2xx

/api/orders

187 ms

11

OrderController.create

186 ms

1

HTTP GET

111 ms

1

HTTP 2xx

/fraud/checkOrder

2

AntiFraudController.checkOrde

test

211 μs

Span details

View span in Discover

Name

Service

Transaction

OrderController.create

frontend

/api/orders

2 hours ago

186 ms (100% of transaction)

app

internal

Metadata

How to add labels and other data

Filter...

labels

labels.customer_id	customer-81
labels.order_price_range	small
labels.payment_method	credit_card
labels.process_runtime_description	Eclipse Adoptium OpenJDK 64-Bit Server VM 17.0.3+7
labels.service_namespace	com-shoppingcart
labels.shipping_country	US
labels.shipping_method	express
labels.telemetry_auto_version	1.12.0
labels.thread_name	http-nio-8080-exec-15

@timestamp

@timestamp	2022-05-27T14:16:39.203Z
------------	--------------------------

+ Add filter

metrics*

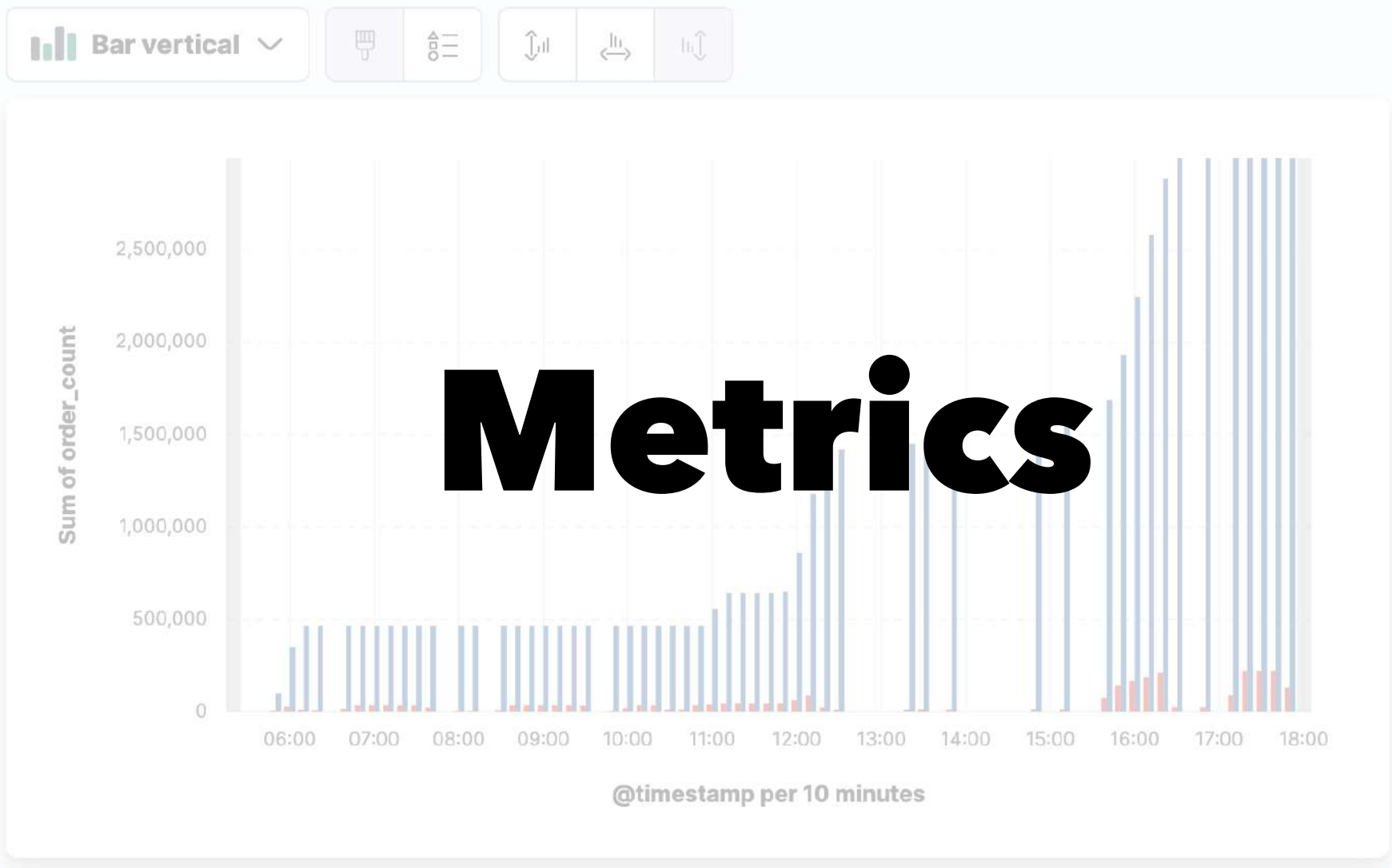
order

Filter by type 0

- # antifraud_rejected_order
- # antifraud_rejected_order_count
- # antifraud_rejected_order_value
- # order
- # order_count
- # order_sum
- # order_value_with_tags_counter
- # order_with_tags

Empty fields 0

Meta fields 0



Suggestions

Current visualization

2,813,626

1,063,443.5!

Bar vertical

metrics*

Horizontal axisOptional

@timestamp

Vertical axis

Sum of order_count

Average of order_sum

+ Add or drag-and-drop a field

Break down byOptional

+ Add or drag-and-drop a field

Add layer

+ Add filter

metrics*

order

Filter by type 0

antifraud_rejected_order

antifraud_rejected_order_count

antifraud_rejected_order_value

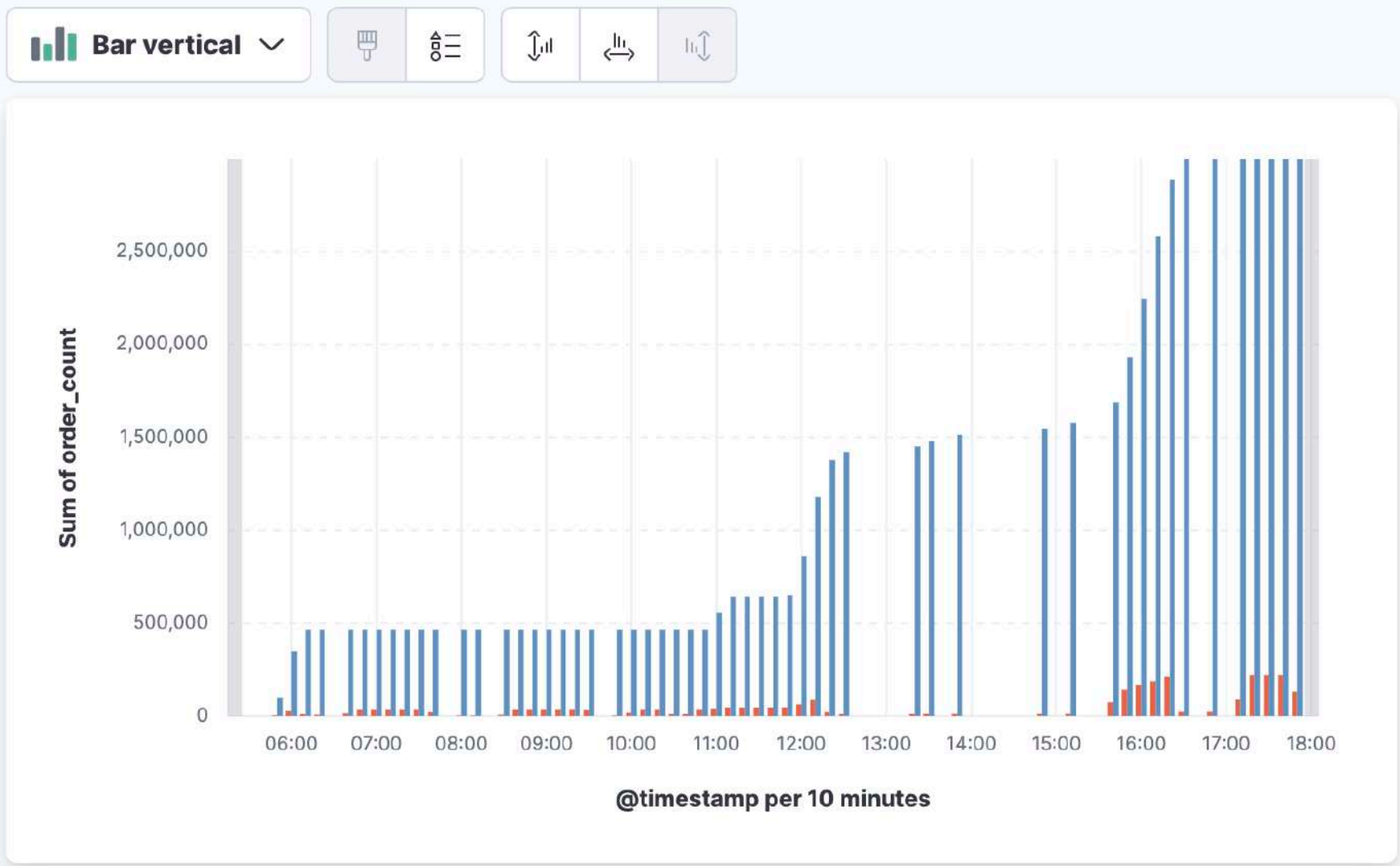
order

order_count

order_sum

order_value_with_tags_counter

order_with_tags



Suggestions

Current visualization

2,813,626

1,063,443.5!

Bar vertical

metrics*

Horizontal axis Optional

@timestamp

Vertical axis

Sum of order_count

Average of order_sum

+ Add or drag-and-drop a field

Break down by Optional

+ Add or drag-and-drop a field

Add layer

Spring 6 & Spring Boot 3

Micrometer as wrapper
for OTel, Brave, Zipkin, Wavefront

Think SLF4J

**[https://github.com/micrometer-metrics/
tracing](https://github.com/micrometer-metrics/tracing)**

OTel vs Vendor Agents

No feature parity (yet)

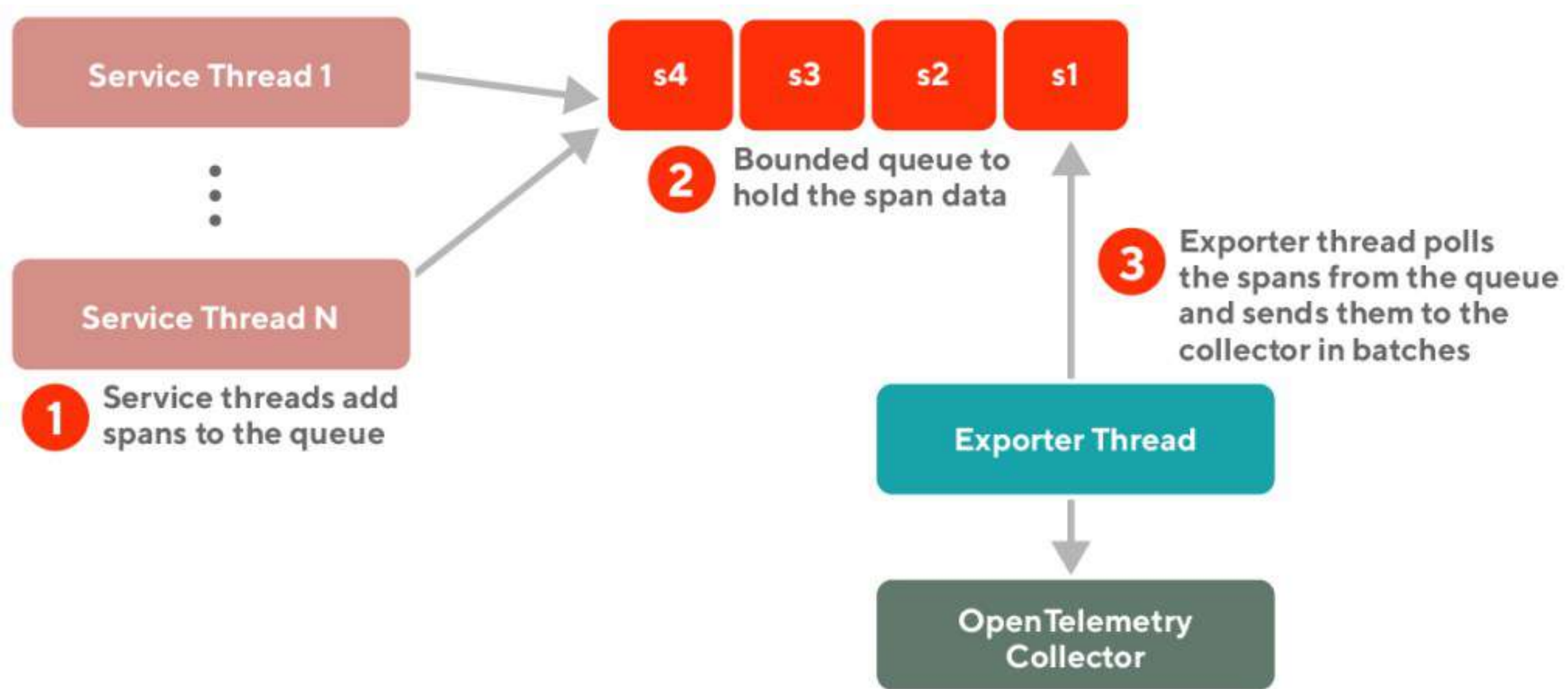
Strong OTel momentum; especially in tracing

Overhead Overview

In the app & on the OTel backend

Agent config – features, sampling, etc:
<https://opentelemetry.io/docs/instrumentation/java/automatic/agent-config/>

Overhead Agent



<https://doordash.engineering/2021/04/07/optimizing-opentelemetrys-span-processor/>

Overhead Benchmark

CPU Network Request latency Throughput Startup time

The following table shows the average CPU load for the user running the test application:

Condition	Value
No instrumentation	32%
Splunk OpenTelemetry Java agent	39%
Splunk OpenTelemetry Java agent with AlwaysOn Profiling	46%

- Tested on March 16, 2022 using Splunk Java OTel agent version 1.9.0. For a description of the test environment, see [Configuration of the test environment](#).

<https://docs.splunk.com/observability/gdi/get-data-in/application/java/performance.html>

Spring Native

Not yet

Questions?

Conclusion



Why?



What?



Who?



How?



Philipp Krenn

@xeraa