The Art of Event-Driven Observability with OpenTelemetry

Henrik Rexed
Cloud Native Advocate | Dynatrace

Henrik Rexed

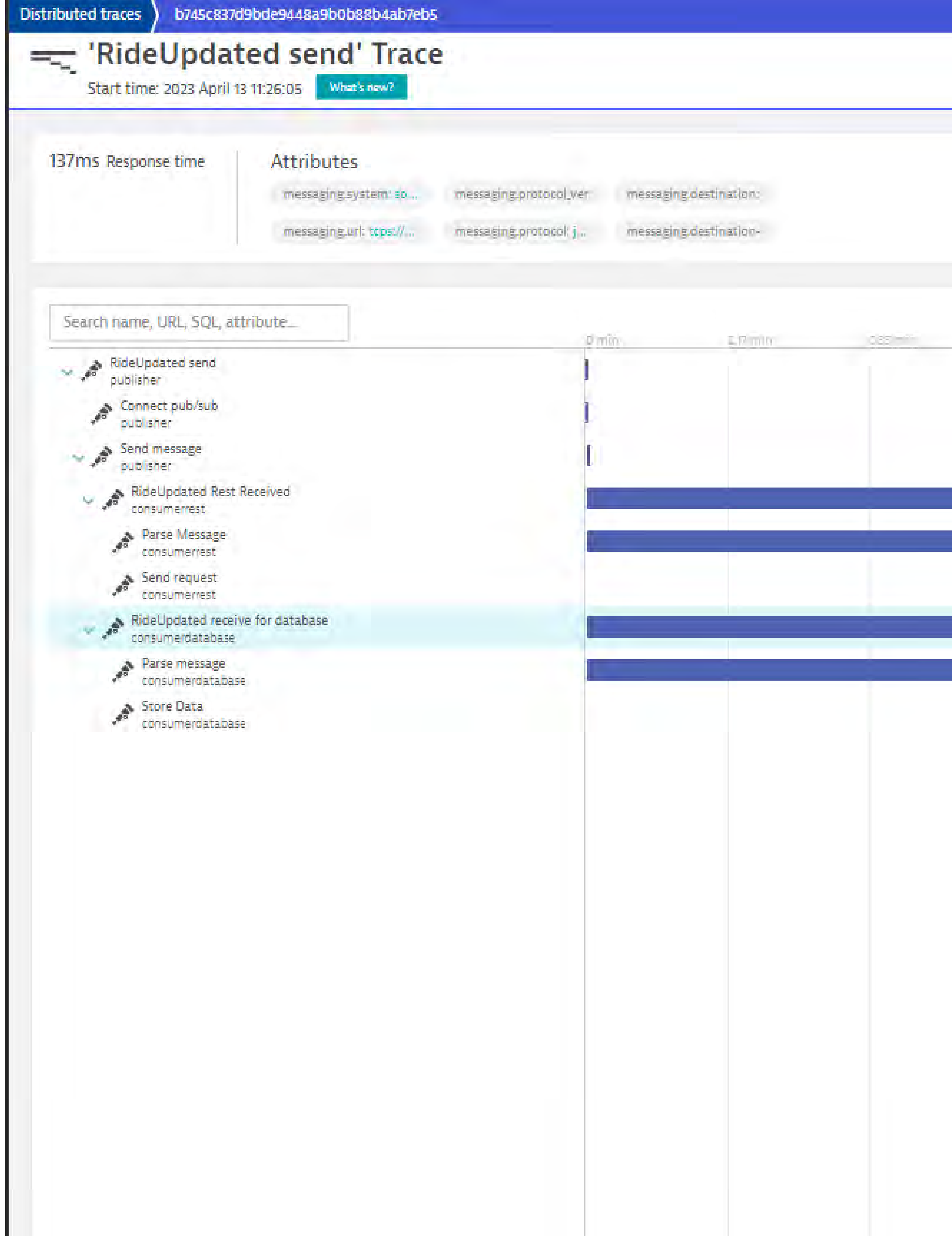
Cloud Native Advocate
15+ years of Performance engineering

Owner of

Producer of

If you stay with me you will ...

- Gentle reminder on OpenTelemetry
  - The various components
  - How to produce traces
- The various way of instrumenting EDA architecture
- The value of Span links



Distributed traces > b745c837d9bde9448a9b0b88b4ab7eb5

'RideUpdated send' Trace

Start time: 2023 April 13 11:26:05

137ms Response time

Attributes

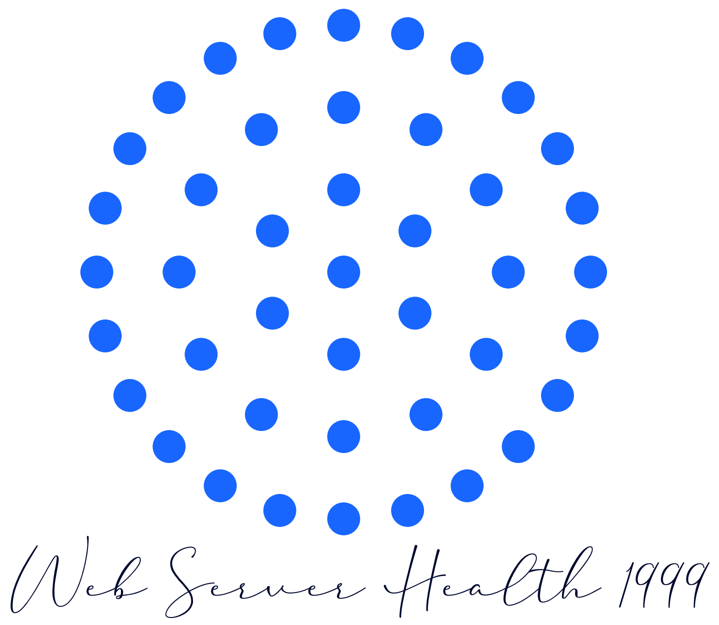
messaging.system: so... messaging.protocol_ver messaging.destination:
messaging.url: tcps://... messaging.protocol: j... messaging.destination-

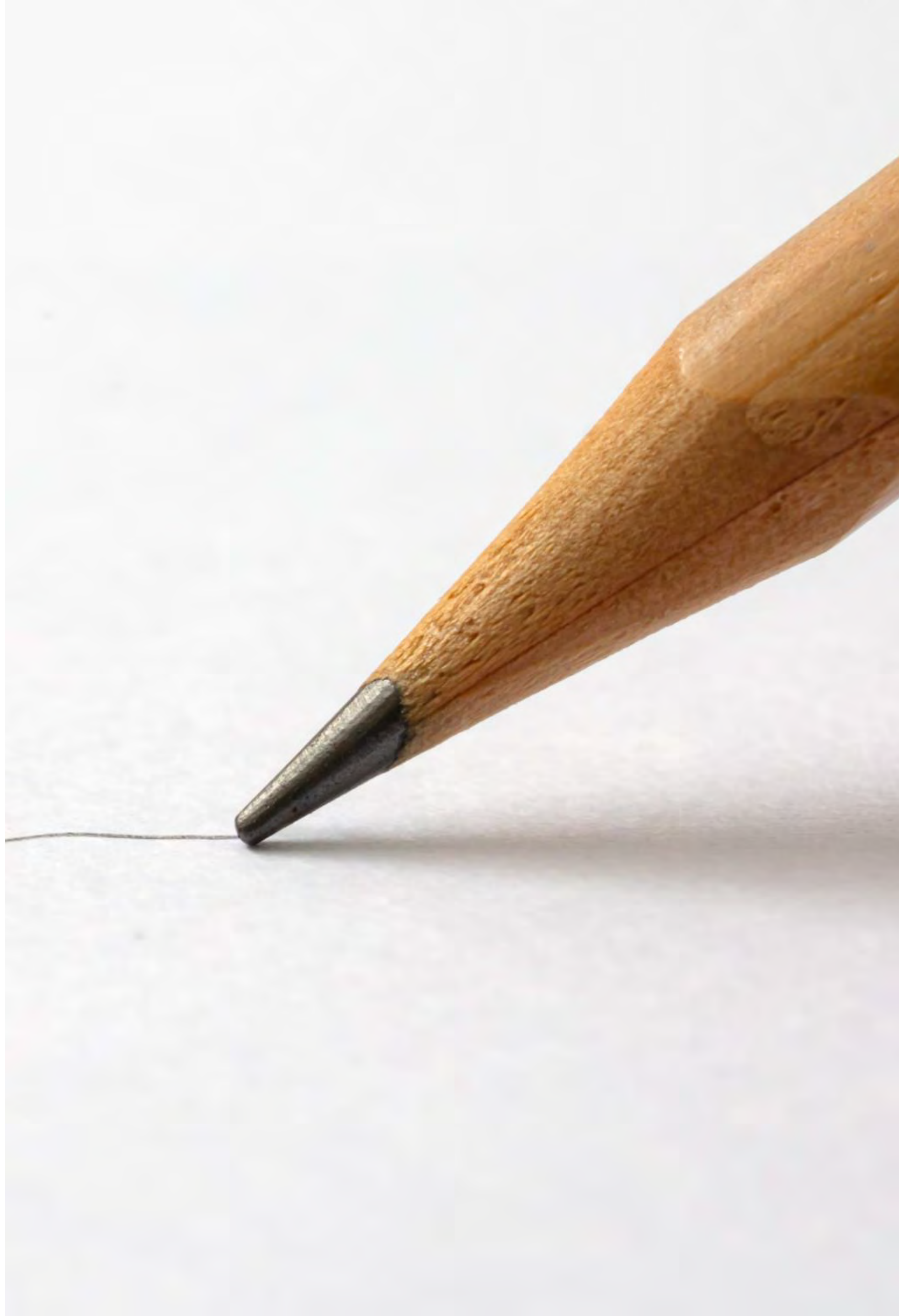
RideUpdated send publisher
Connect pub/sub publisher
Send message publisher
RideUpdated Rest Received consumerrest
Parse Message consumerrest
Send request consumerrest
RideUpdated receive for database consumerdatabase
Parse message consumerdatabase
Store Data consumerdatabase

Once Upon the time

24 years ago …

- My Manager taught me how to use tools to get system health (perfmon, top, nmon…etc).





20 years ago…

- With solution having history and providing metadata from our environment helped me to design a better visualization:

Web Server Health 2004

13 years ago…

- With the usage of APM solutions, distributed traces, metrics , it was easier to represent the situation:

Web Server Health 2010

10 years ago…

- By looking at the logs produced by our application and servers, the situation was ….a bit ….clearer

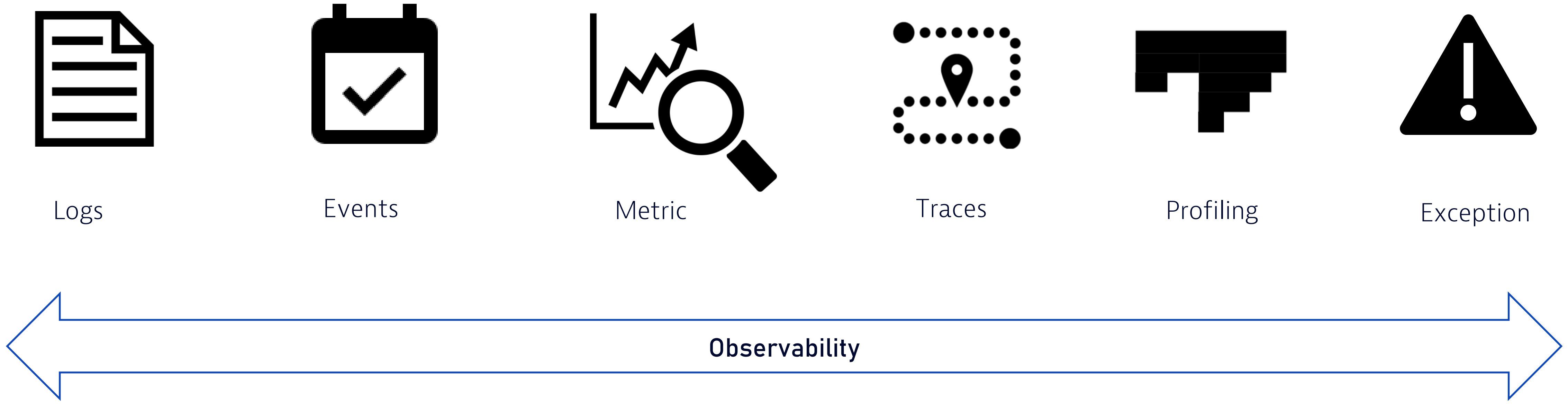
Web Server Health 2014

What I wanted to represent

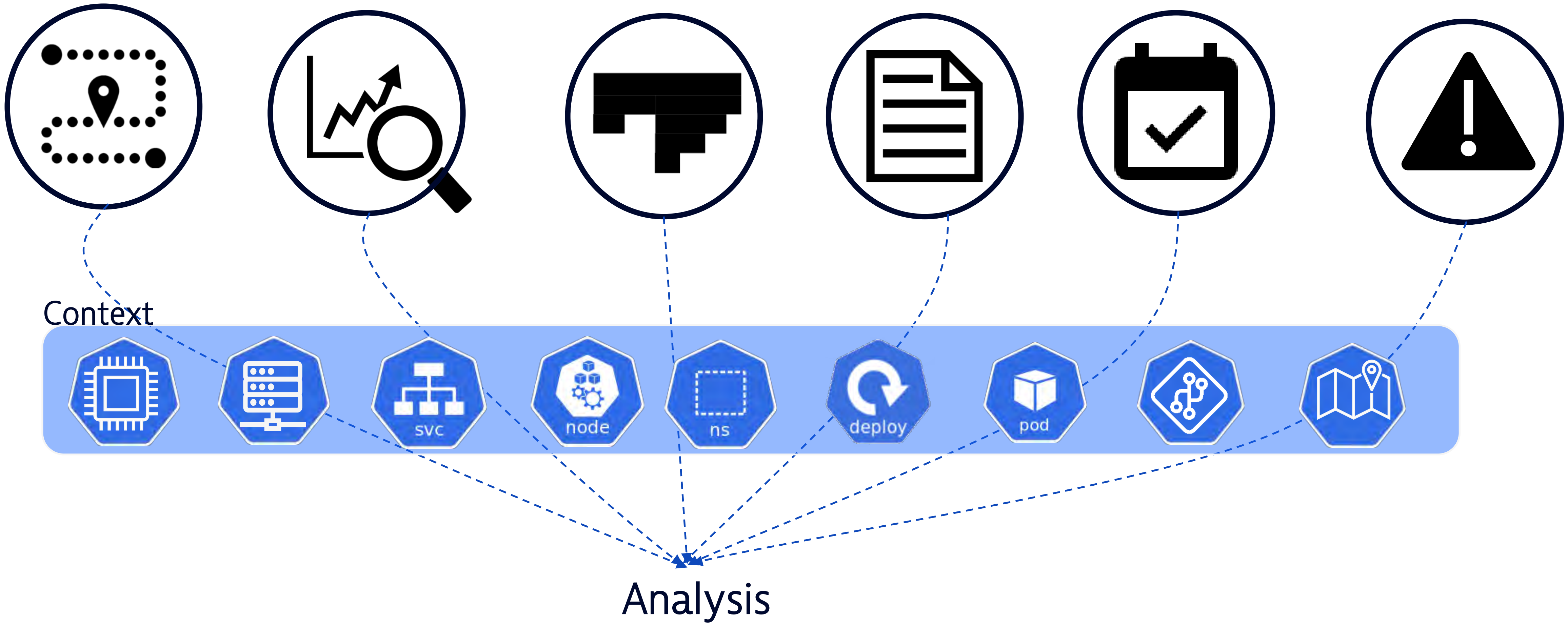
What I wanted to represent

Our artistict tools

Observability pillars



Why do we need several signals?

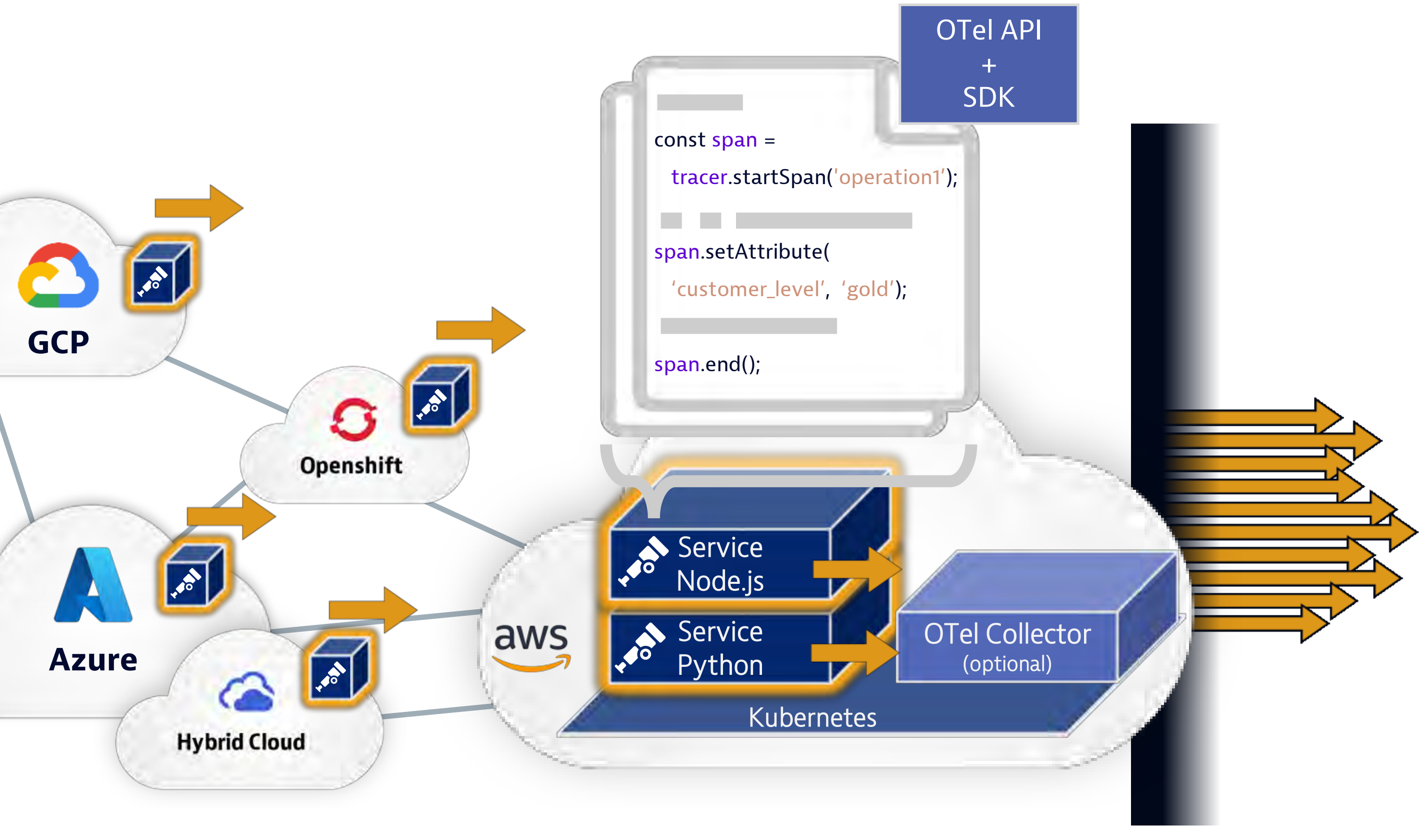


We isolate our signals

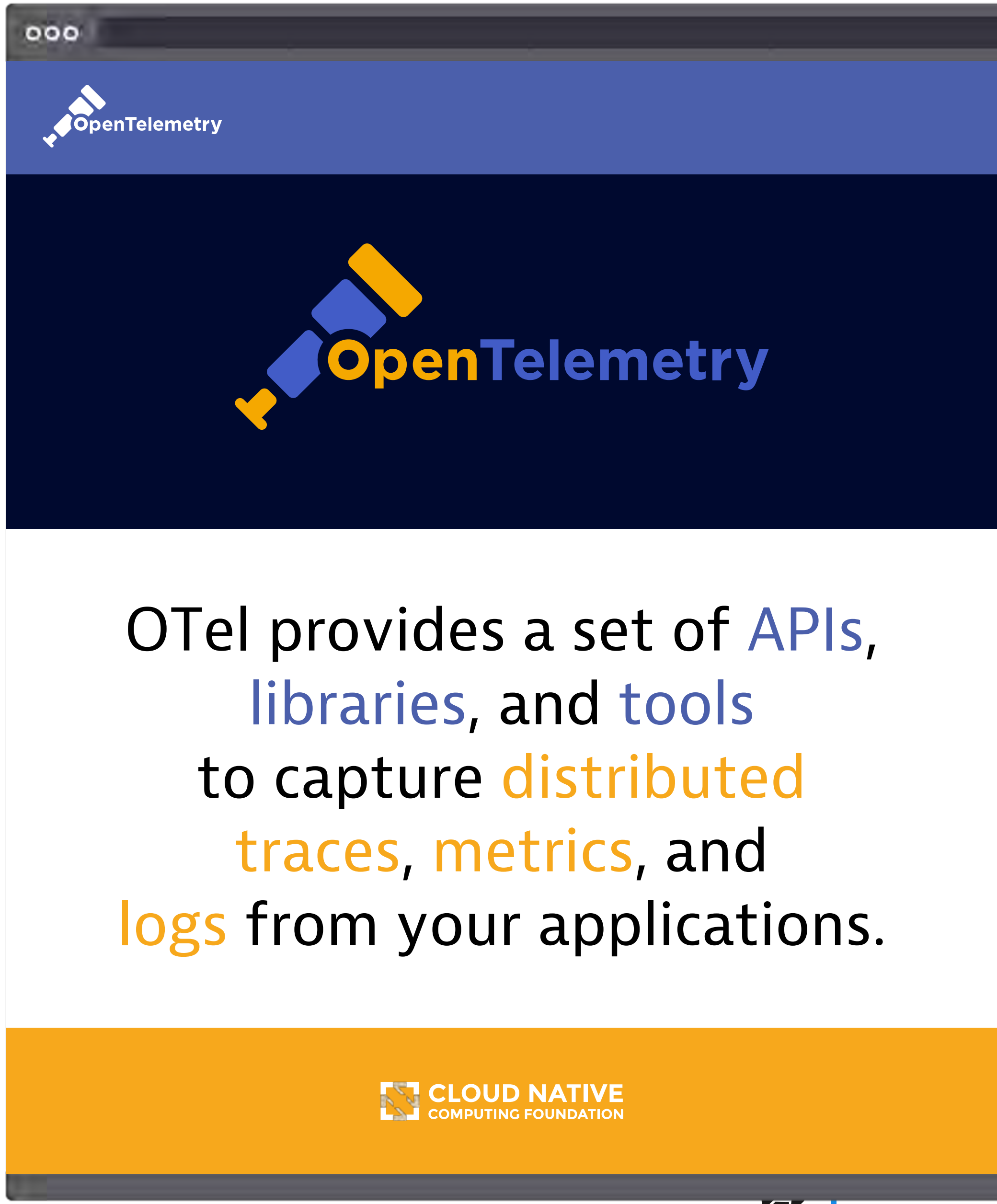
- Most of the Organizations tends to seperate the usage and storage of :
  - Logs
  - Traces
  - Profiling
  - Metrics

OpenTelemetry

What is OpenTelemetry (OTel)?

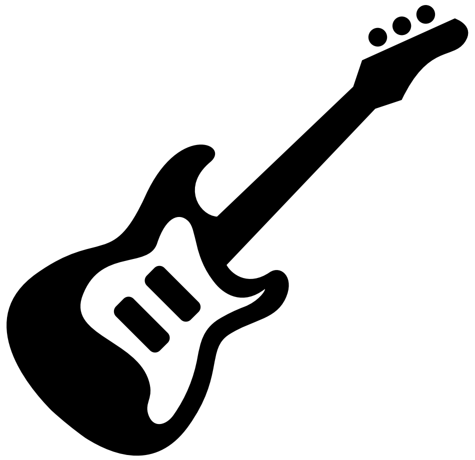


```
const span =
  tracer.startSpan('operation1');
span.setAttribute(
  'customer_level', 'gold');
span.end();
```



OTel provides a set of APIs, libraries, and tools to capture distributed traces, metrics, and logs from your applications.

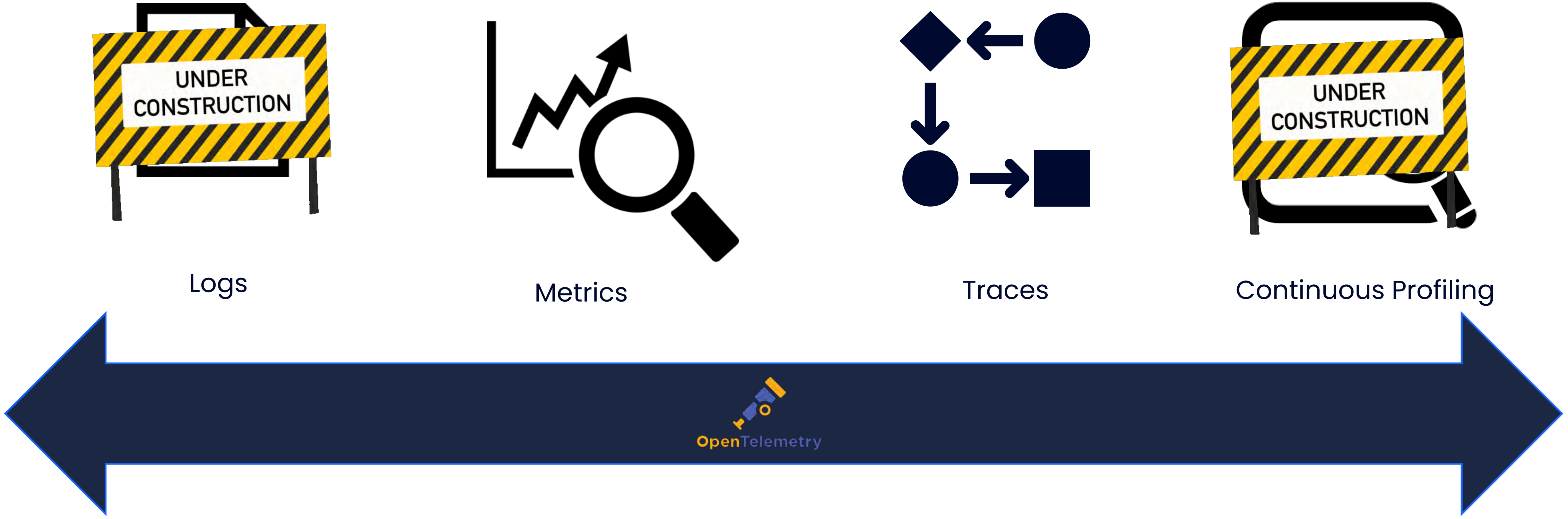
The main Component of OpenTelemetry



Instrument

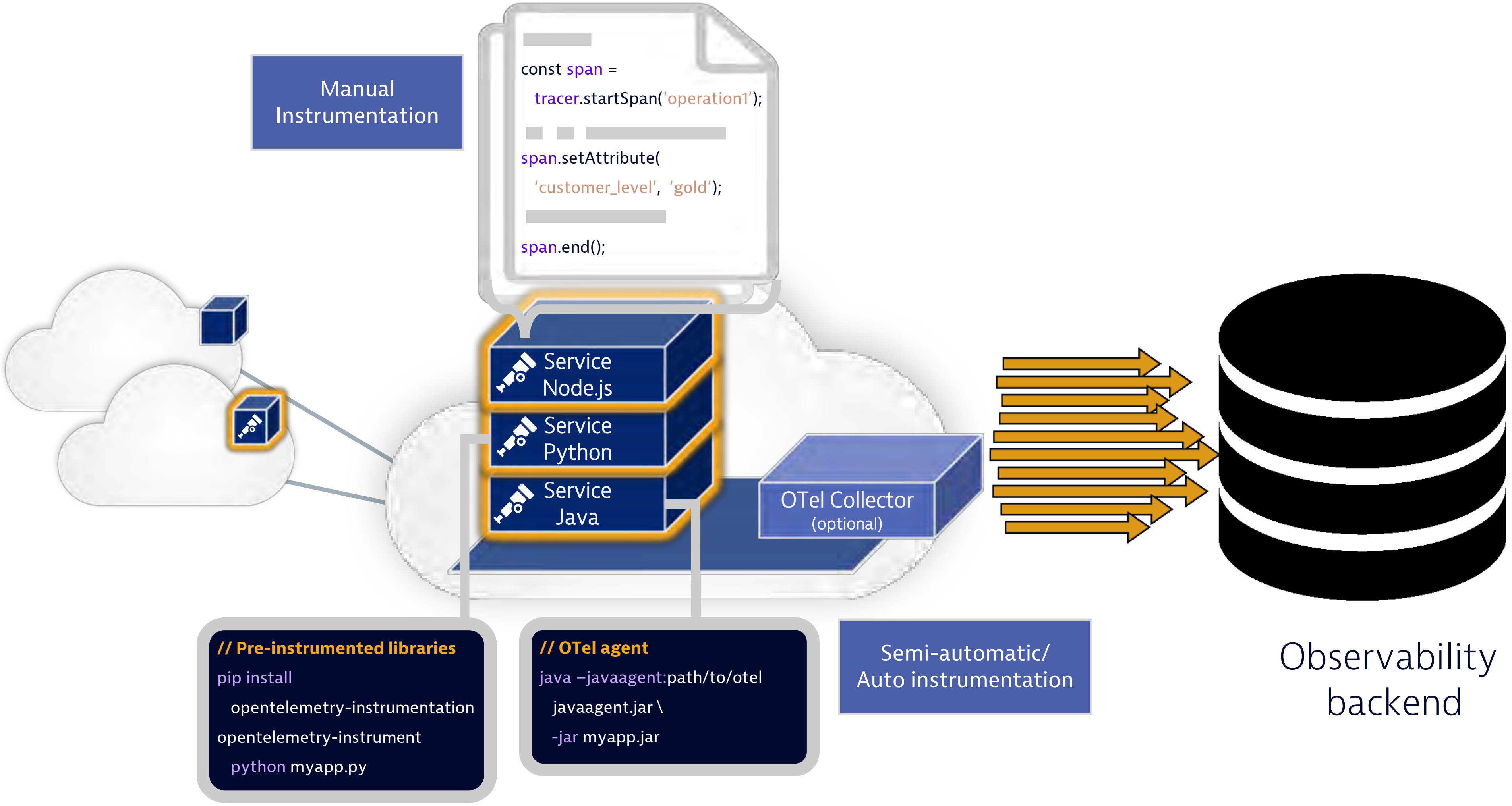
Collector

OpenTelemetry the Standard for Observability



How to produce traces?

How to add OTel to your applications?



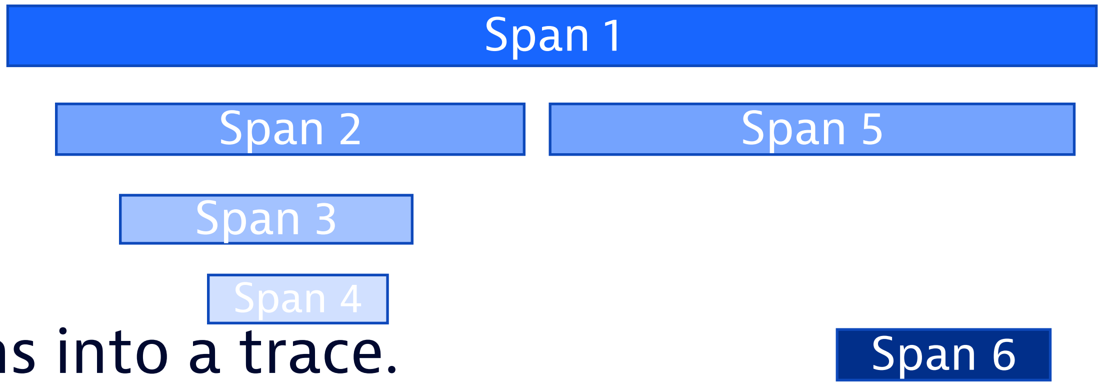
```
const span =
  tracer.startSpan('operation1');

span.setAttribute(
  'customer_level', 'gold');

span.end();
```

```
// Pre-instrumented libraries
pip install
  opentelemetry-instrumentation
opentelemetry-instrument
  python myapp.py
```

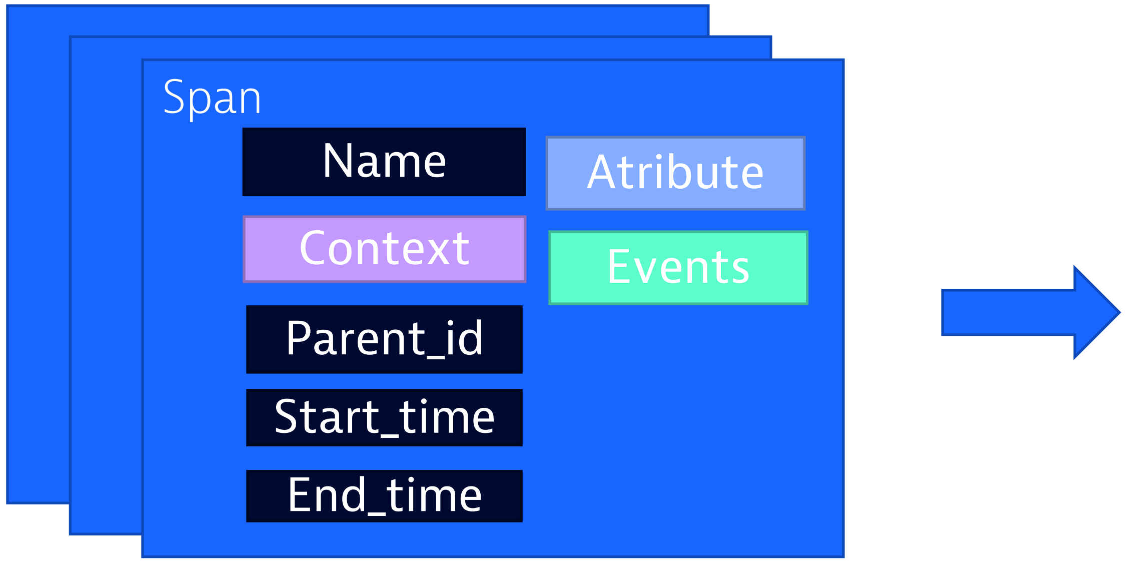
```
// OTel agent
java –javaagent:path/to/otel
  javaagent.jar \
  -jar myapp.jar
```

What is a trace?

- A trace is made of Spans.

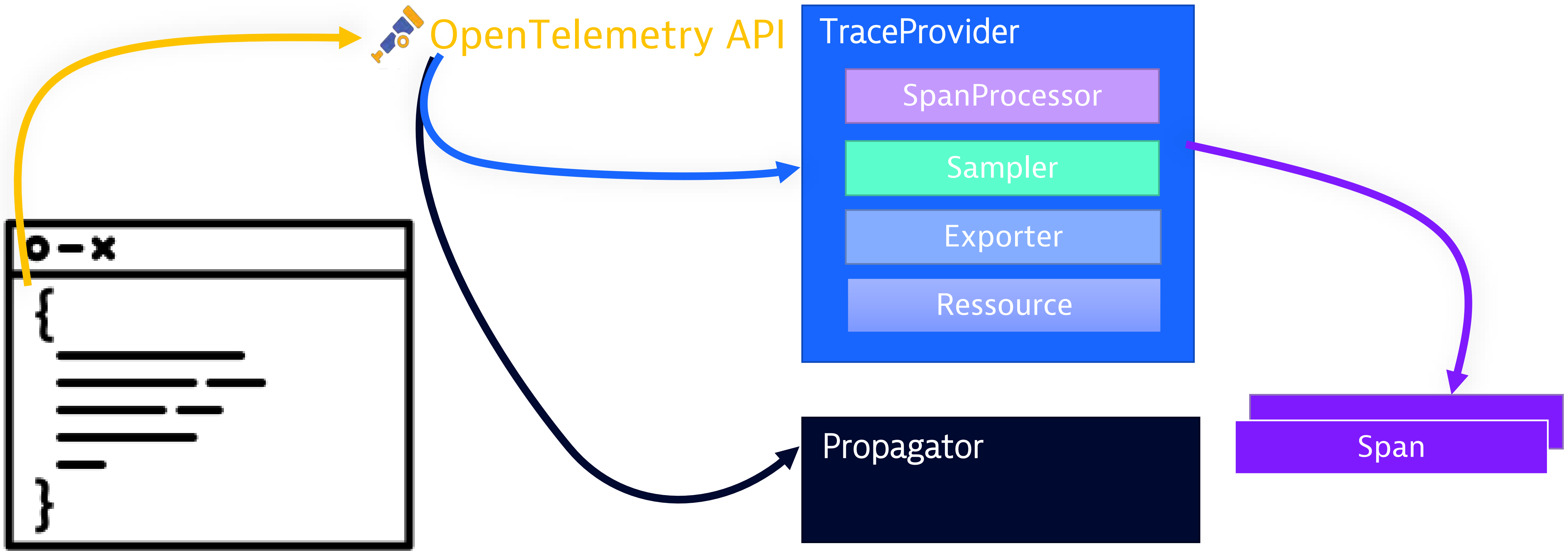


- The trace Context glue all the various spans into a trace.



```
{
  "name": "Hello-Greetings",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x5fb397be34d26b51",
  },
  "parent_id": "0x051581bf3cb55c13",
   'status_code' "STATUS_CODE_OK"
  "start_time": "2022-04-29T18:52:58.114304Z",
  "end_time": "2022-04-29T18:52:58.114435Z",
  "attributes": {
    "http.route": "some_route1"
  },
  "events": [
    {
      "name": "hey there!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    },
    {
      "name": "bye now!",
      "timestamp": "2022-04-29T22:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ],
}
```

Tracing Instrumentation



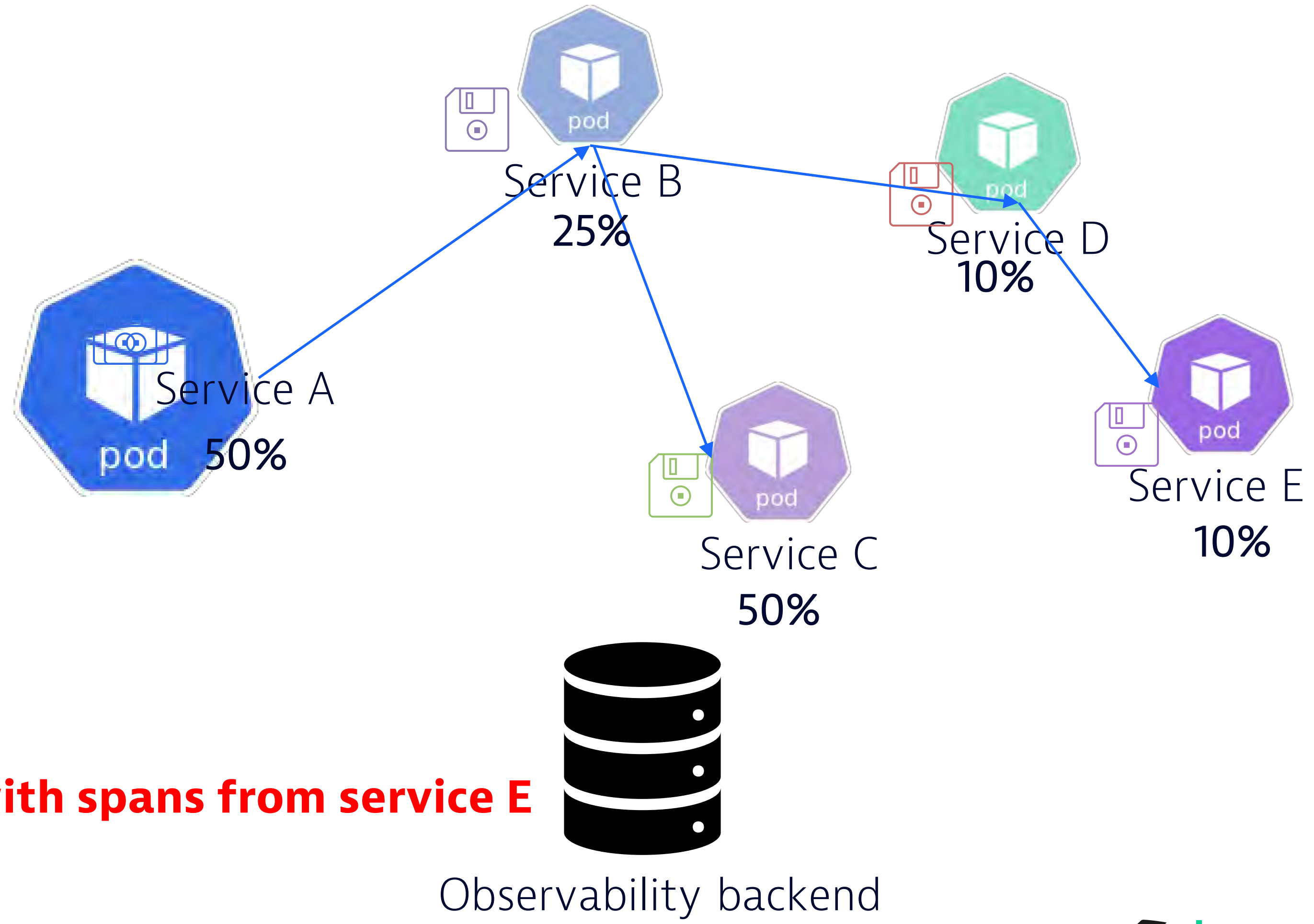
Resource

- Resource is the identity of a process production Telemetry
- The Resource is key to name telemetry components in the backends.
- There are standard attributes to define a resource :

Attribute	Type	Description	Required?
Service.name	String	Name of the service	Yes
Service.namespace	String	Namespace of the service.name	No
Service.instance.id	String		No
Service.version	String	Version number of the service	No

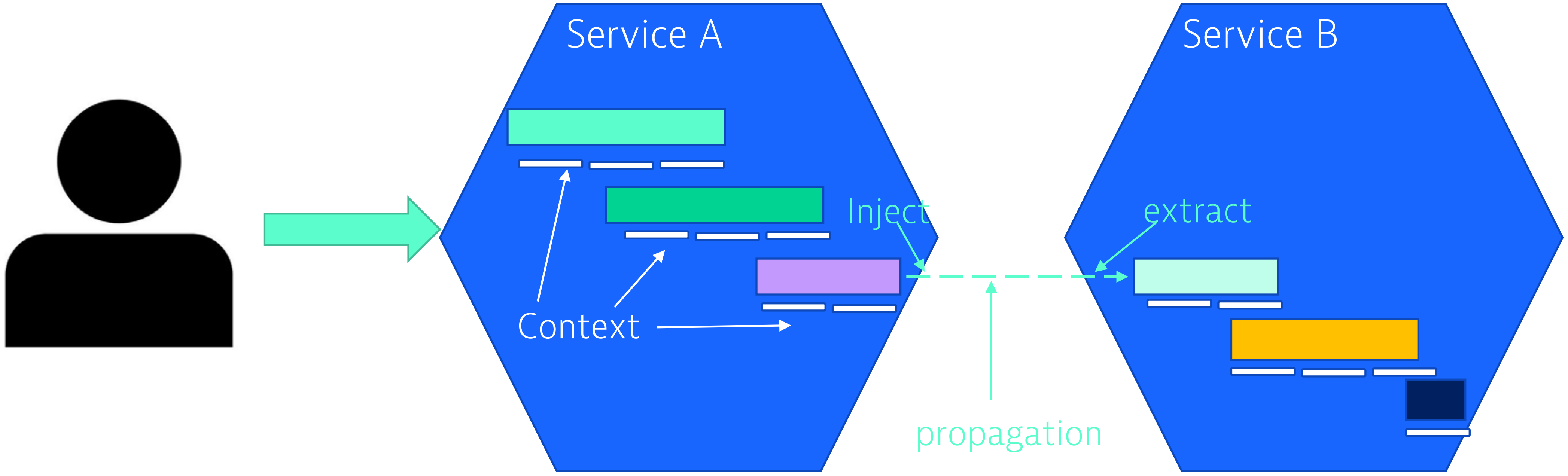
Tracing Sampler

- AlwaysON
- AlwaysOff
- ParentBased
- TraceIdratioBased
- parentbased_always_on
- parentbased_traceidratio
- parentbased_always_off



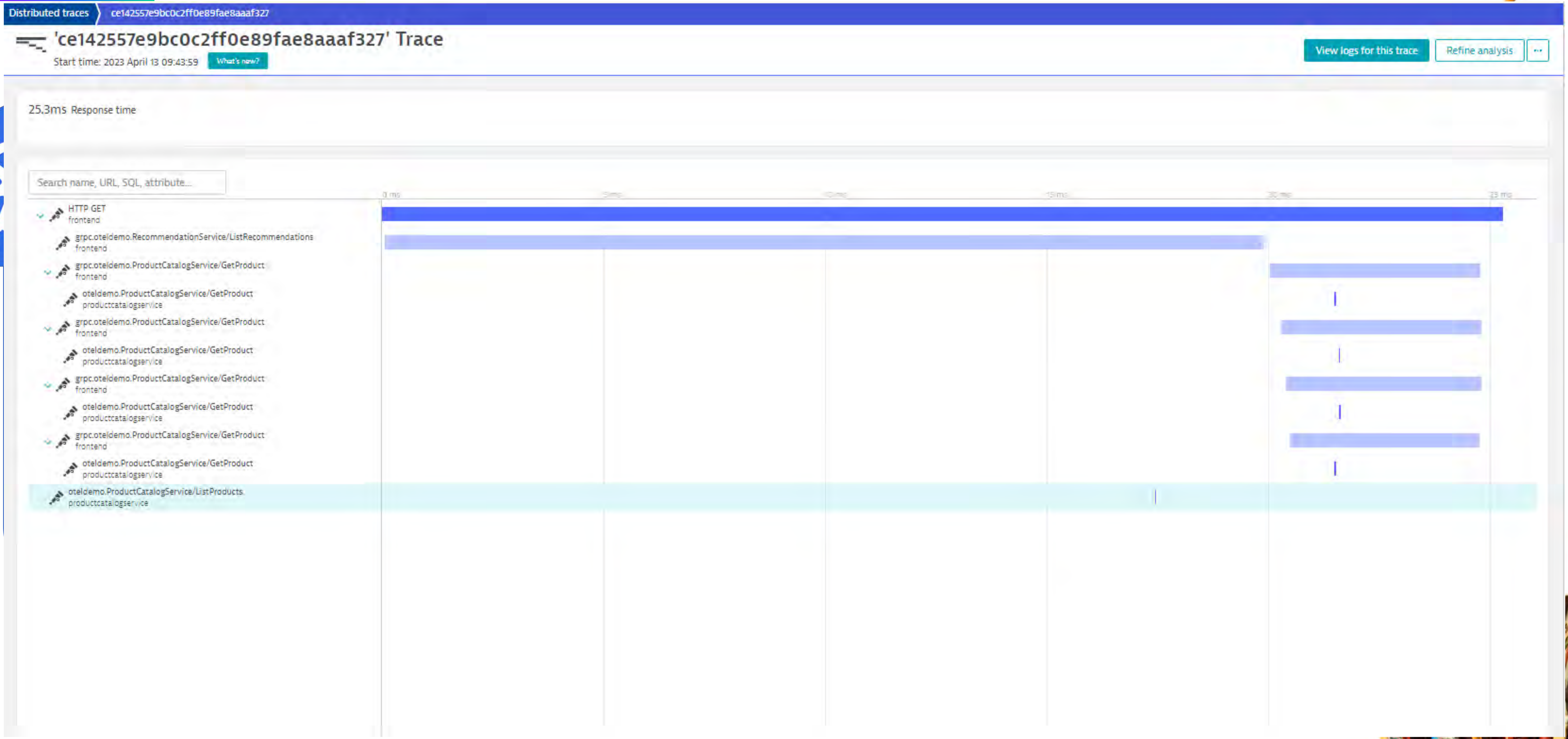
1000 requests = 1 request with End2End trace with spans from service E

What is propagation?



In a traditional micro service architecture

Distributed tracing in normal architecture



Distributed traces 〉 ce142557e9bc0c2ff0e89fae8aaaf327

'ce142557e9bc0c2ff0e89fae8aaaf327' Trace

Start time: 2023 April 13 09:43:59

25.3ms Response time

HTTP GET
frontend

grpc.oteldemo.RecommendationService/ListRecommendations
frontend

grpc.oteldemo.ProductCatalogService/GetProduct
frontend

oteldemo.ProductCatalogService/GetProduct
productcatalogservice

grpc.oteldemo.ProductCatalogService/GetProduct
frontend

oteldemo.ProductCatalogService/GetProduct
productcatalogservice

grpc.oteldemo.ProductCatalogService/GetProduct
frontend

oteldemo.ProductCatalogService/GetProduct
productcatalogservice

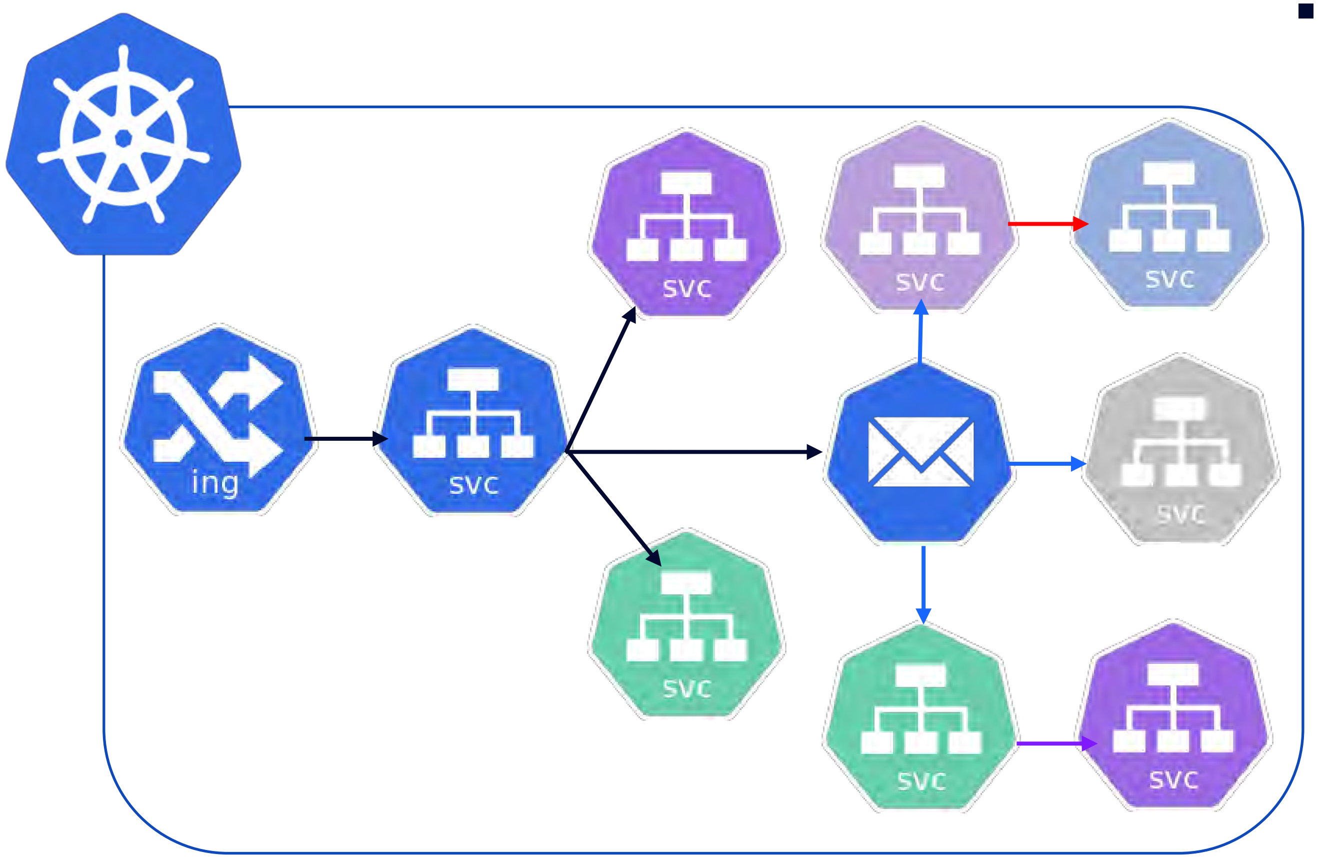
grpc.oteldemo.ProductCatalogService/GetProduct
frontend

oteldemo.ProductCatalogService/GetProduct
productcatalogservice

oteldemo.ProductCatalogService/ListProducts
productcatalogservice

YEAH, SO?

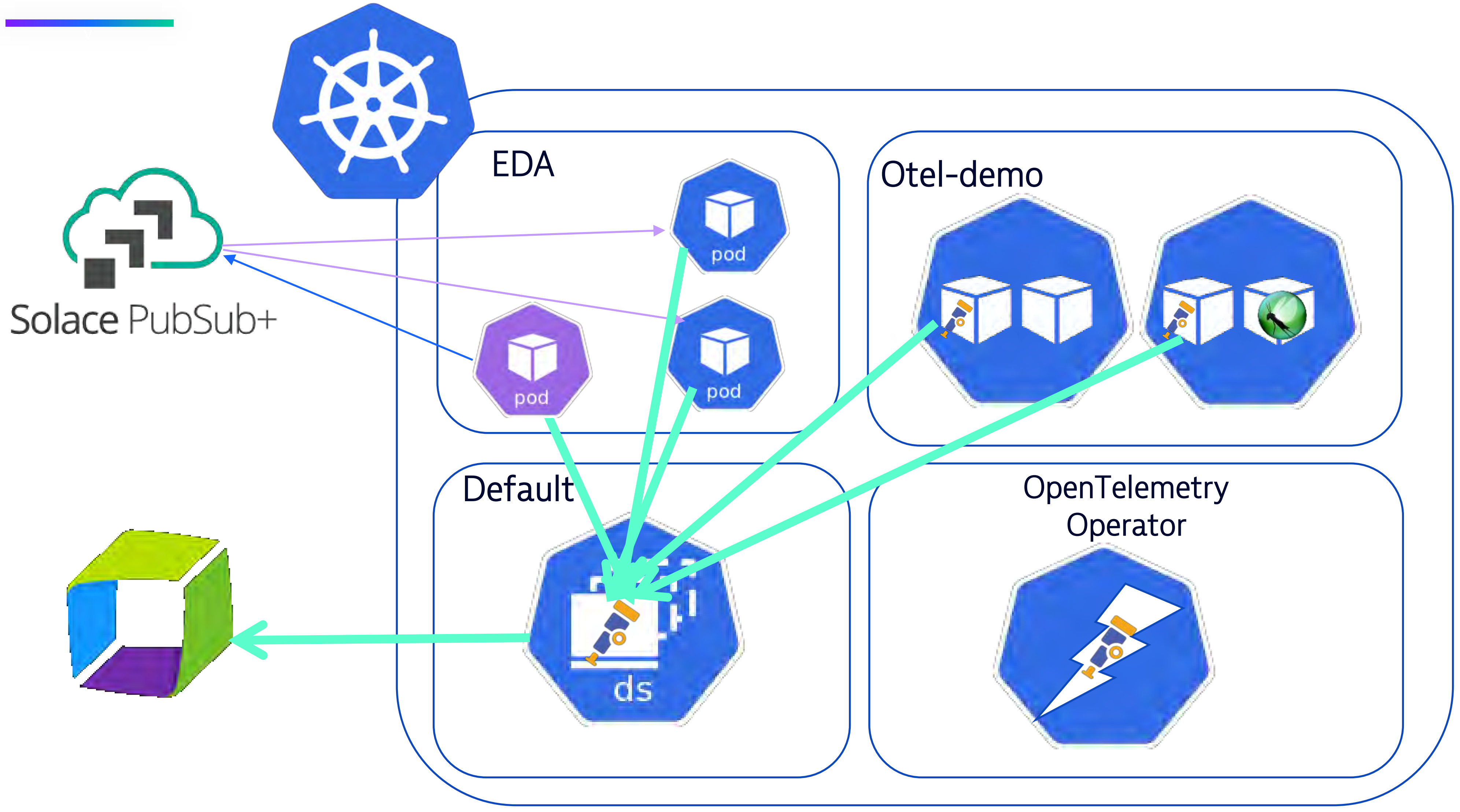
Distributed tracing in EDA



- With OpenTelemetry, I can represent my transactions in various way :
  - One Big Traces
  - Separate my transactions in sub transactions.

Example 1 – End2End trace

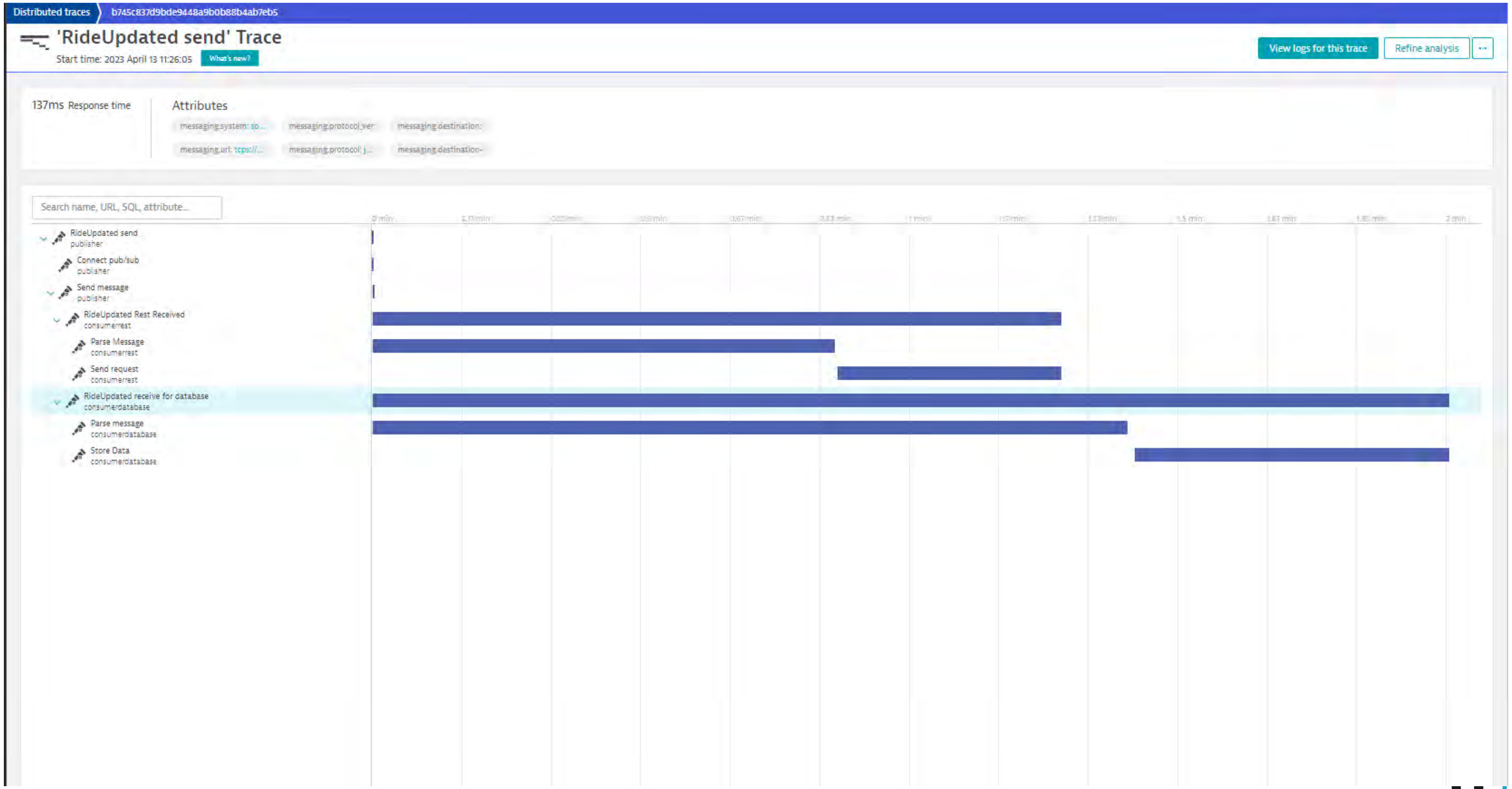
Our environnent



https://github.com/isItObservable/tracing_eda

SHH!

Is generated distributed traces useful?



Distributed traces 〉 b745c837d9bde9448a9b0b88b4ab7eb5

'RideUpdated send' Trace

Start time: 2023 April 13 11:26:05

137ms Response time

Attributes

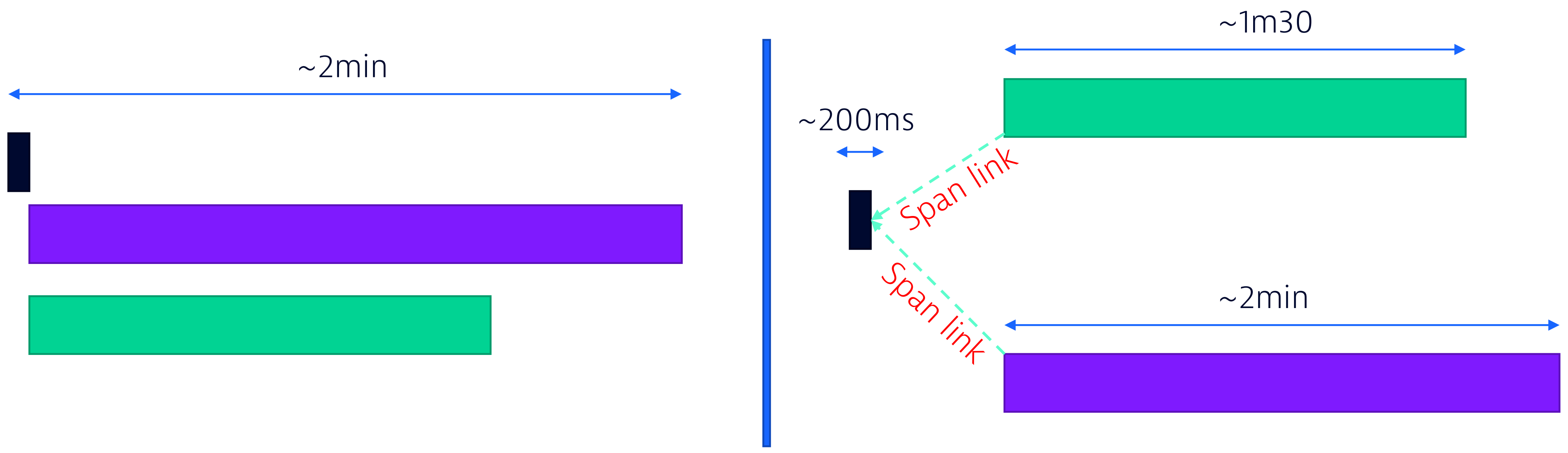
messaging.system: so...
messaging.protocol_ver
messaging.destination:
messaging.url: tcps://...
messaging.protocol: j...
messaging.destination-

RideUpdated send
publisher
Connect pub/sub
publisher
Send message
publisher
RideUpdated Rest Received
consumerrest
Parse Message
consumerrest
Send request
consumerrest
RideUpdated receive for database
consumerdatabase
Parse message
consumerdatabase
Store Data
consumerdatabase

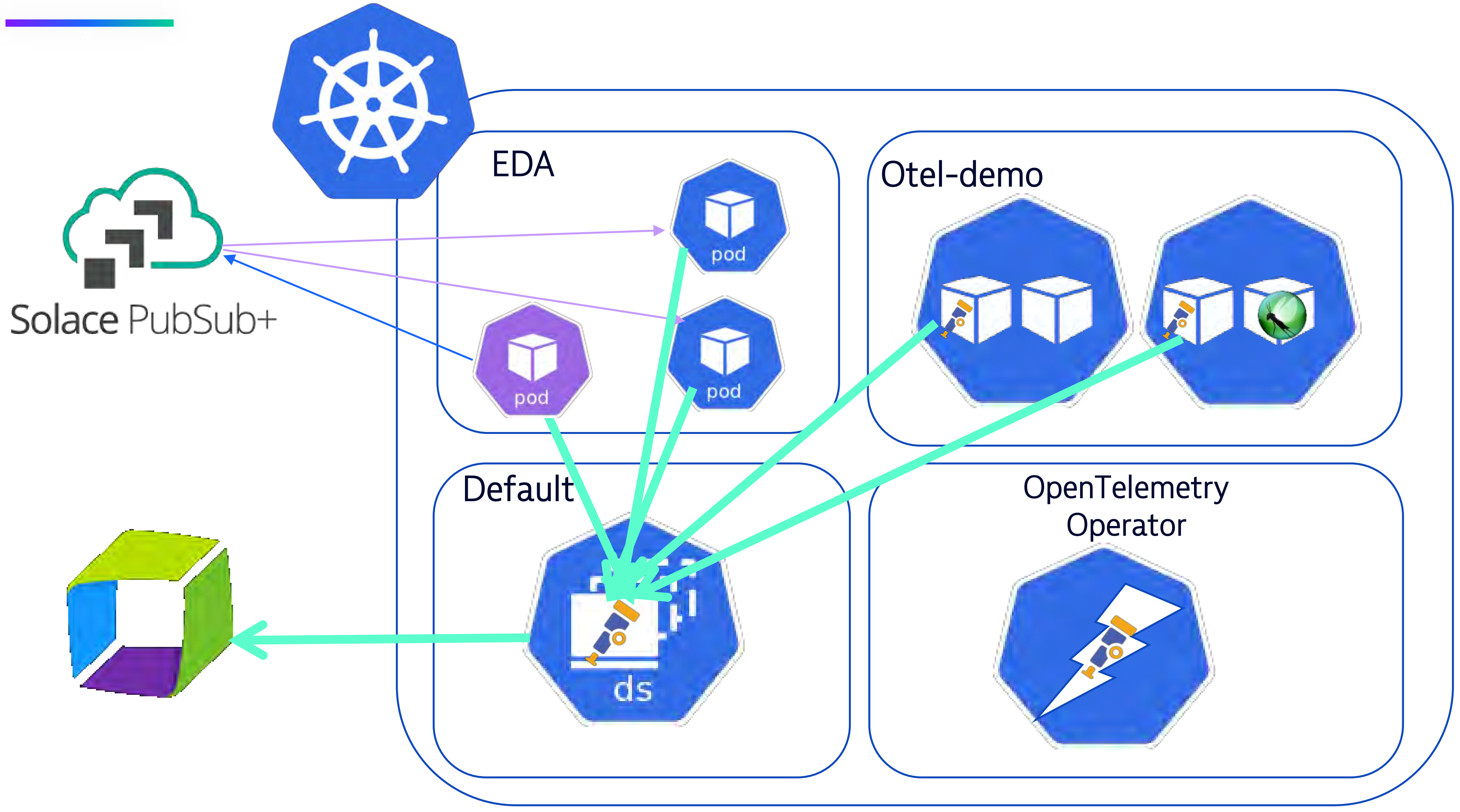
Example 1 – The usage of Span Links

Span Links

- Span link is feature allowing to create a releationship between seperated spans.



Our environnent



https://github.com/isItObservable/tracing_eda

4

Conclusion

Pros/Cons

Pros	Cons
Keep track on the consumers	Relation between Producer & Consumer
Visualize traces	Sampling decision

Take Away

Instrumentation

- Make sure your code is agnostic using no Vendor library and exporter

Observability

- Make sure the metrics produced has enough dimensions
- Produce logs with contextual information
- Add Span Attributes to your spans

Creativity

- Understand your system
- Design the right Observability

Is it observable

- Looking for educational content on Observability , Checkout the Youtube Channel :

Is It Observable

Is it Observable ?

dynatrace

CLOUD DONE RIGHT