

solace.

PubSub+

Distributed

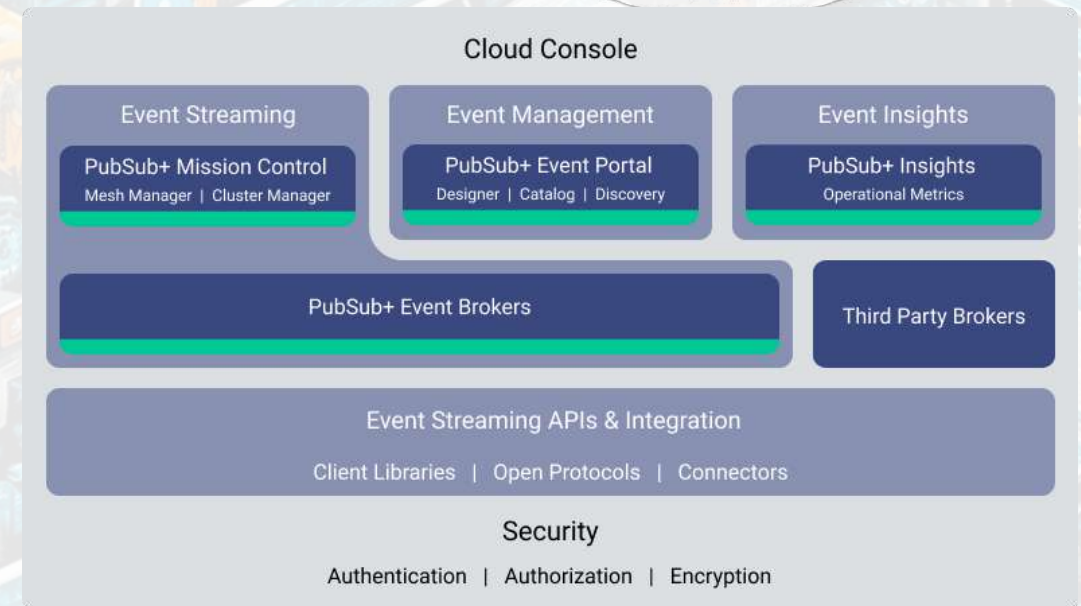
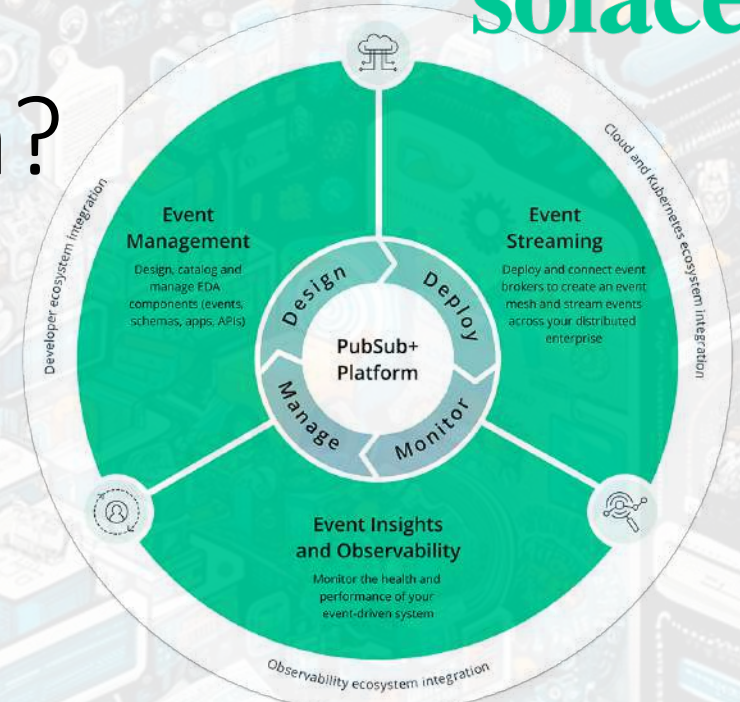
Tracing

*Demo 101 for dummies, experts
and everyone in between*



What is Solace PubSub+ Platform?

Solace PubSub+ Platform is a complete event streaming and management platform for the real-time enterprise. The Platform helps enterprises design, deploy, and manage event-driven architectures (EDAs) across hybrid cloud, multi-cloud, and IoT environments, so enterprises can be more integrated and event-driven.



What is Distributed Tracing?

*Distributed tracing is **a method of observing requests as they propagate through distributed cloud environments**. It follows an interaction (business process) and tags it with a unique identifier. This identifier stays with the transaction as it interacts with microservices, containers, and infrastructure. In turn, this identifier offers real-time visibility into user experience, from the top of the stack to the application layer and the infrastructure beneath.*

In production distributed tracing provides a real-time 360 degrees on the business, enabling quick response whenever needed with relevant root cause analysis data.

Developers can use distributed tracing to troubleshoot requests that exhibit high latency or errors.

Solace PubSub+ Distributed Tracing adds end-to-end observability of events streaming through your enterprise.

Related: Monitoring, APM, Observability, Log analysis

Why Distributed Tracing for Solace PubSub+ ?

Real-time insights in core business processes is critical for organizations deploying complex microservices over an event driven architecture or striving to satisfy regulatory and audit compliance.

Solace PubSub+ Distributed Tracing uses OpenTelemetry to provide granular real-time data about the status and delivery of messages and events to the observability tools of your choice.

Note that I deliberately left out the ‘business value’. Your homework to come up with that 😊 (I do have answers).

OpenTelemetry

Open Source and vendor Neutral, 100% free to use, de facto standard for Distributed Tracing throughout the industry.

OpenTelemetry is adopted and supported by industry leaders, backed by CNCF.

For more see <https://opentelemetry.io/>



OpenTelemetry integrates with leading application performance monitoring and observability tooling



DATADOG



dynatrace



elastic apm



JAEGER



Lightstep

splunk>



Google Cloud
Operations Suite



Amazon
CloudWatch

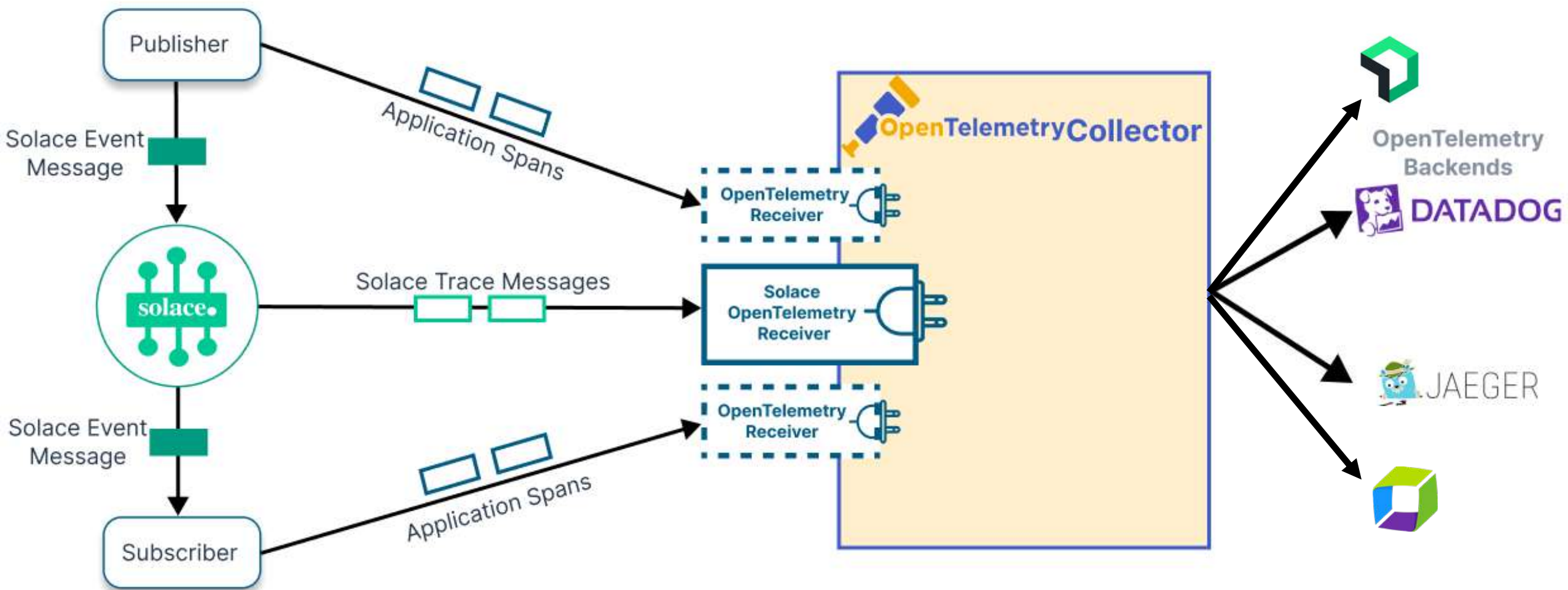


logz.io

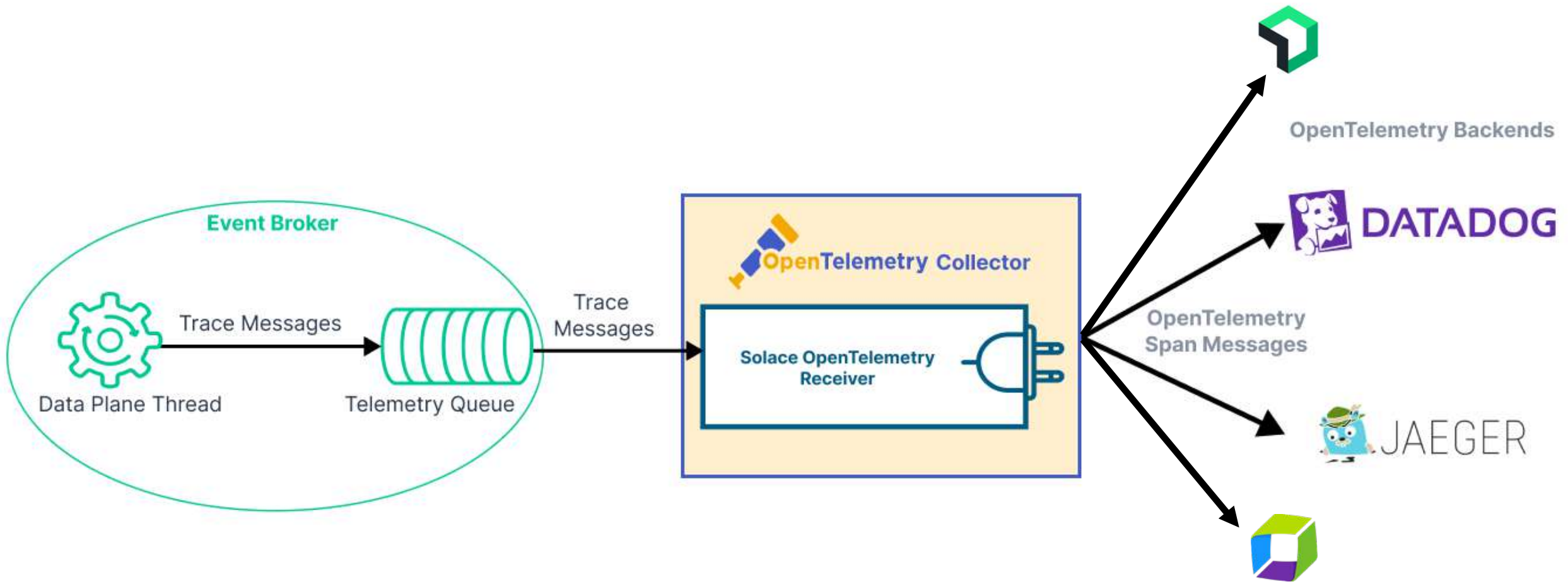


ZIPKIN

Solace PubSub+ Distributed Tracing - Setup



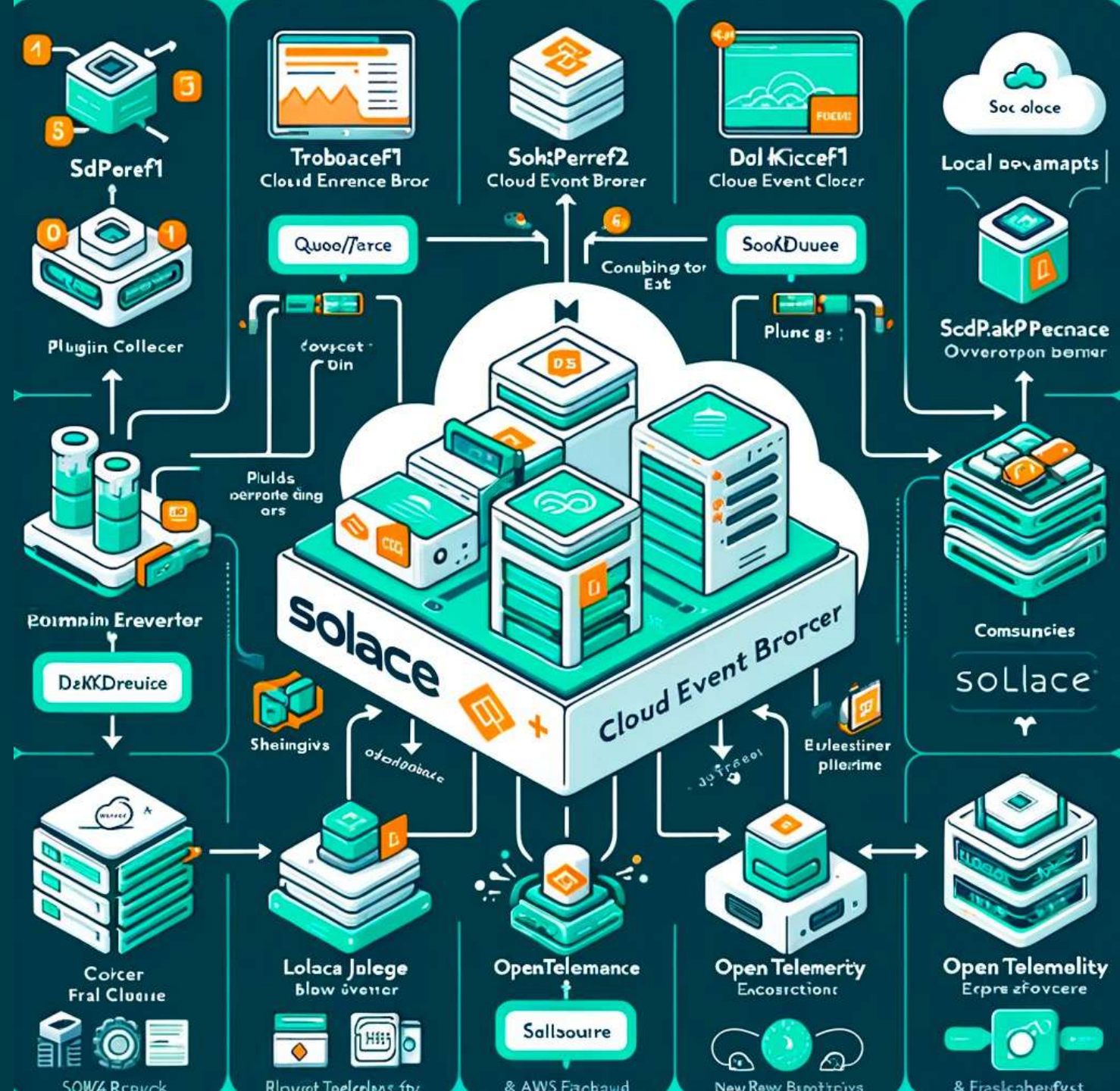
Solace PubSub+ Distributed Tracing - Broker



Demo setup

- SDKPerf **Publisher** *local*
- Solace PubSub+ Cloud broker *AWS Frankfurt*
- SDKPerf **Consumer** *local*
- OTEL Collector *local*
- Jaeger *local*
- New Relic *Cloud*
- Dynatrace *Cloud*
- DataDog *Cloud*

MacBook Pro M2



OTEL Collector with Solace Receiver plugin

```
× -zsh
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [core] [Channel #1] Closing the name resolver {
"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [core] [Channel #1] ccBalancerWrapper: closing{
"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [core] [Channel #1 SubChannel #2] Subchannel Co
nnectivity change to SHUTDOWN {"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [core] [Channel #1 SubChannel #2] Subchannel de
leted {"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [transport] [client-transport 0x14001ffd8c0] Cl
osing: rpc error: code = Canceled desc = grpc: the client connection is closing {"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [transport] [client-transport 0x14001ffd8c0] lo
opyWriter exiting with error: transport closed by client {"grpc_log": true}
2024-01-24T21:21:47.861+0100    info    zapgrpc/zapgrpc.go:178    [core] [Channel #1] Channel deleted {"grpc_
log": true}
2024-01-24T21:21:47.861+0100    info    extensions/extensions.go:59    Stopping extensions...
2024-01-24T21:21:47.861+0100    info    service@v0.92.0/service.go:205    Shutdown complete.
emilzegers@emilzegers otelcol-contrib_0.92.0 % ./otelcol-contrib --config=../otel-collector-config-single.yaml
```


OTEL Collector configuration

```
! my.yaml
1 processors:
2   batch:
3
4 exporters:
5   logging:
6     loglevel: "debug"
7
8   otlp/jaeger:
9     endpoint: localhost:4317
10    tls:
11      insecure: true
12
13   otlphttp/newrelic:
14     endpoint: ${OTEL_EXPORTER_OTLP_ENDPOINT}
15     tls:
16       insecure: false
17     headers:
18       api-key: ${NEW_RELIC_LICENSE_KEY}
19
20   otlphttp/dynatrace:
21     endpoint: "https://xgo98208.live.dynatrace.com/api/v2/otlp"
22     headers:
23       Authorization: "Api-Token ${API_TOKEN}"
24
25 receivers:
26   otlp:
27     protocols:
28       grpc: # Default port 4317
29         endpoint: 0.0.0.0:4319
30       http: # Default port 4318
```

```
30     http: # Default port 4318
31       endpoint: 0.0.0.0:4320
32       include_metadata: true
33       cors:
34         allowed_origins:
35           - http://*
36           - https://*
37         allowed_headers:
38           - "*"
39
40   solace/broker1:
41     broker: [ez-dt.messaging.solace.cloud]
42     max_unacknowledged: 500
43     auth:
44       sasl_plain:
45         username: trace
46         password: trace123
47     queue: queue://#telemetry-tp1
48     tls:
49       insecure: false
50       insecure_skip_verify: true
51
52 service:
53   telemetry:
54     logs:
55       level: "debug"
56   pipelines:
57     traces:
58       receivers: [solace/broker1, otlp]
59       processors: [batch]
60       exporters: [otlp/jaeger, logging, otlphttp/newrelic, otlphttp/dynatrace]
```


SDKPerf Publisher and Consumer

✕ -zsh

CPU usage = 0%

AD Pub ACK/NACK stats:

Total ACK Events = 1

Total NACK Events = 0

==> Garbage collection information

====> Collection count 4

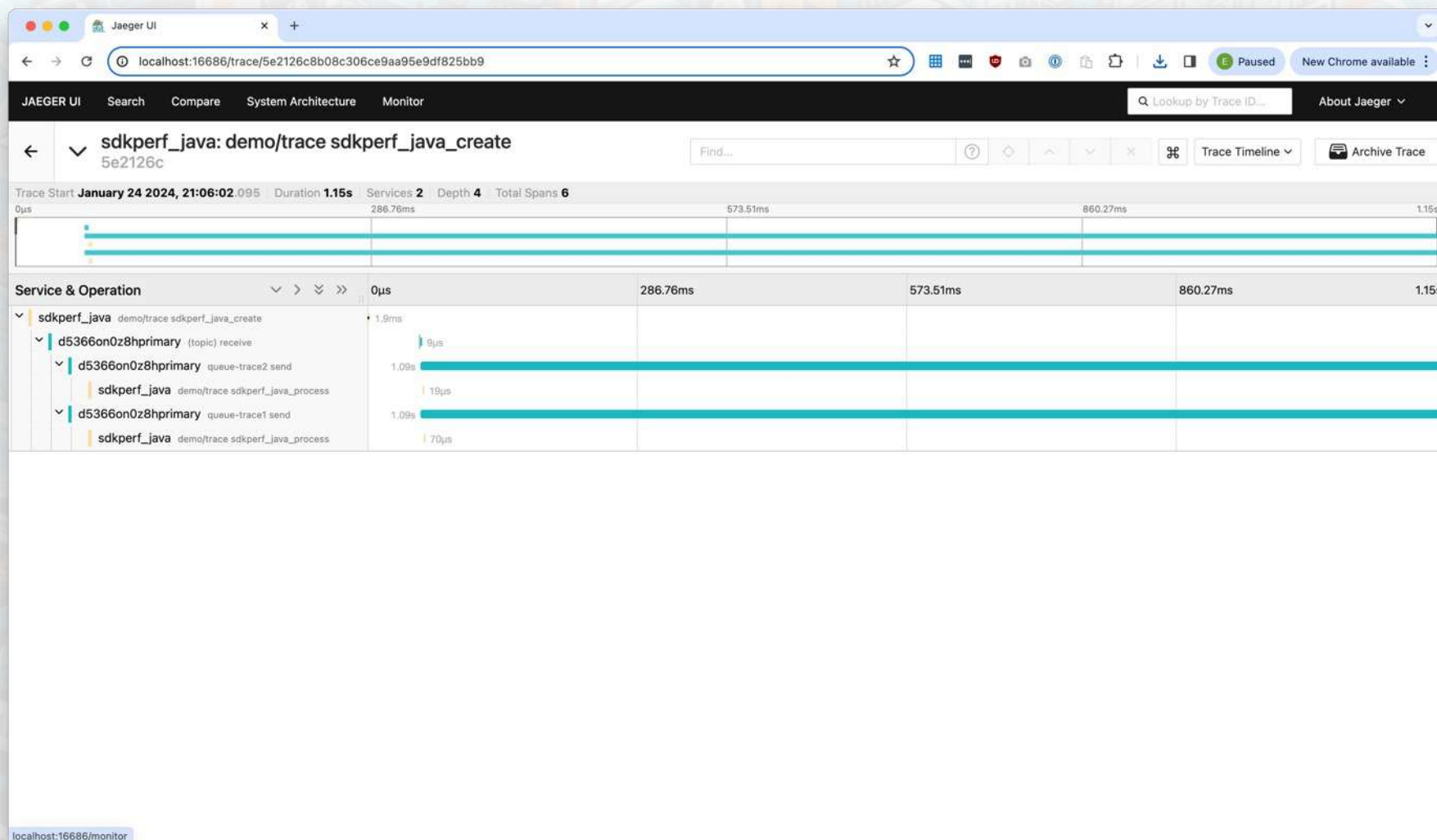
====> Collection time 4 ms.

> Exiting

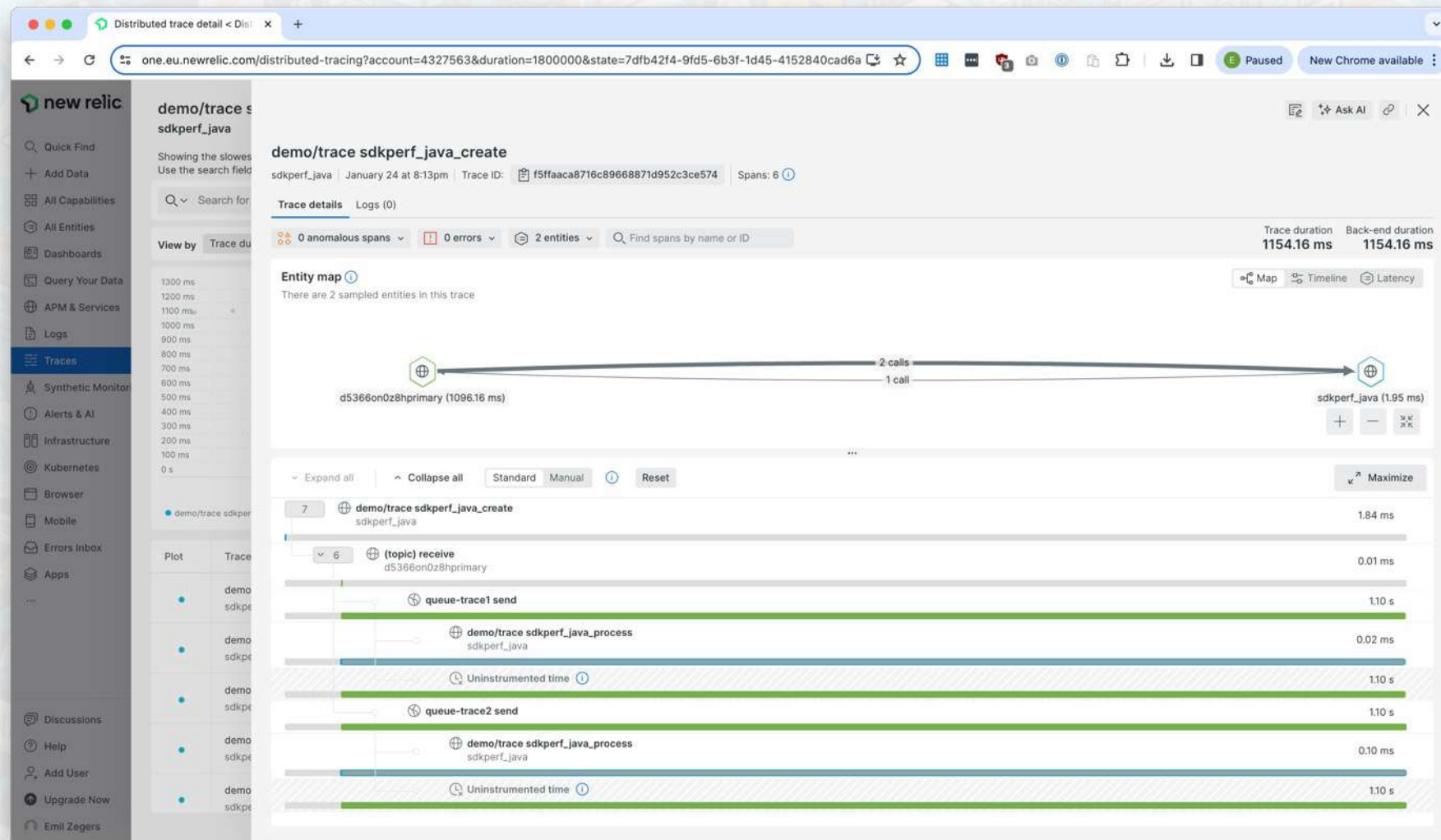
^C%

```
emilzegers@emilzegers sdkperf-jcsmp-8.4.14.10 % while true; do ./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -cu=solace-cloud-client@ez-aws-fra -cp= -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q -tcc -tcrc -tecip="http://localhost:4319"; sleep 10; done
```

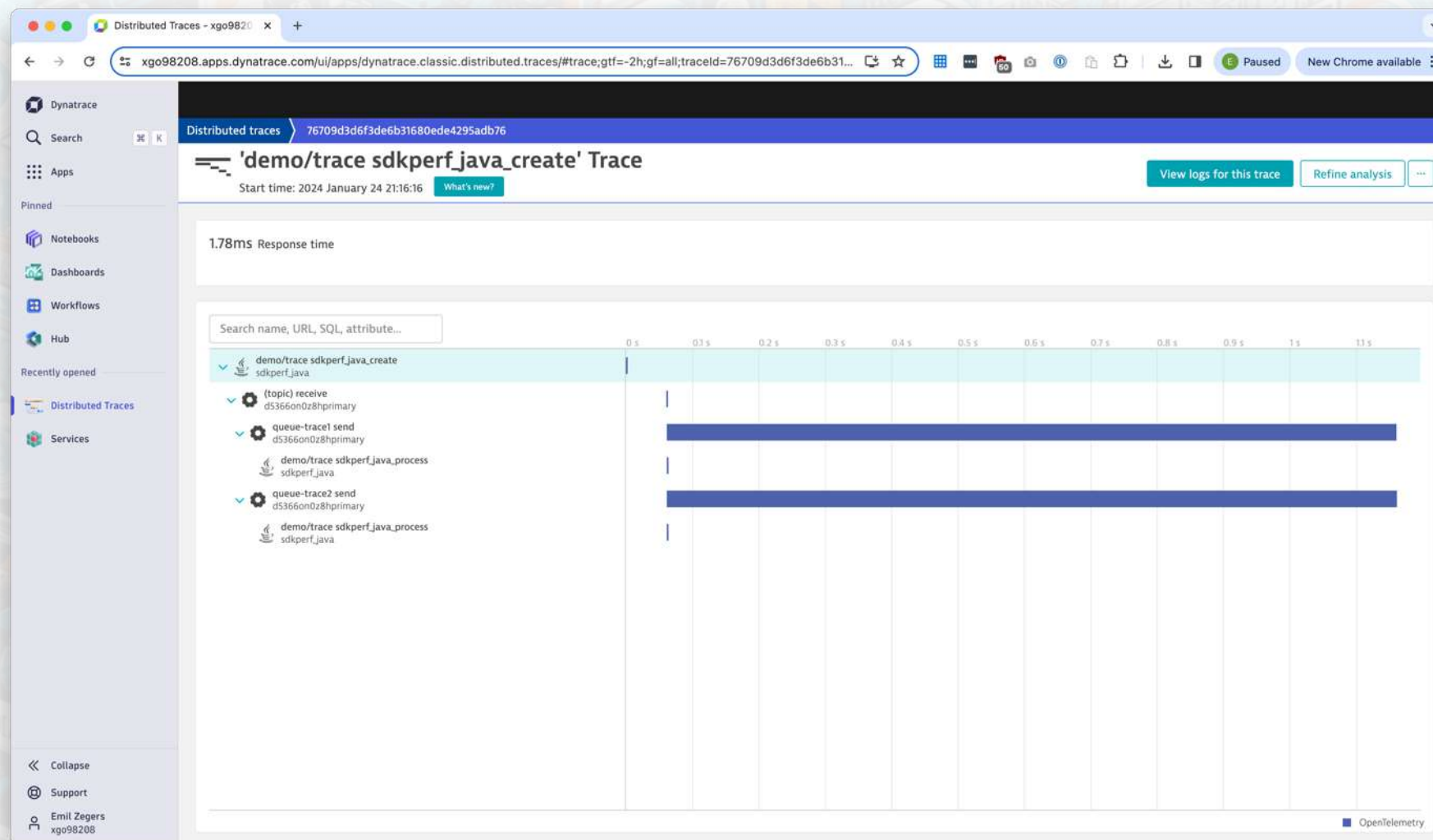

Jaeger



New Relic



Dynatrace



DataDog

To Do...

Splunk

To Do...

Remarks

Open Telemetry is *supported* by all observability solution providers nowadays.

The *challenge* still is if all parts of an application chain support traceid propagation/span creation.



Related capabilities



Solace PubSub+ Insights

Cloud services, in-console view of key performance metrics with advanced dashboards for deeper analysis of capacity, message flow, queues, and more.

[Learn More ›](#)



Solace PubSub+ Monitor

Monitor and alerts for Solace PubSub+ appliances and software to enable operations teams to keep their fingers on the pulse of mission-critical applications.

[Learn More ›](#)



PubSub+ Event Broker Syslog

Syslog is as important as ever for pulling data into your dashboards or maintaining log files for debugging or situation analysis.

[Learn More ›](#)

Links

Repository

<https://github.com/taatuut/clear-agnostic>

Solace PubSub+ Platform & Distributed Tracing

<https://solace.com/blog/monitoring-vs-observability-in-event-driven-systems/>

<https://solace.com/products/event-broker/distributed-tracing/>

<https://docs.solace.com/Features/Distributed-Tracing/Distributed-Tracing-Overview.htm>

<https://solace.com/products/platform/>

OpenTelemetry

<https://opentelemetry.io/ecosystem/registry/?s=solace>

<https://opentelemetry.io/docs/collector/custom-collector/>

solace.

PubSub+ Distributed Tracing

Emil Zegers

Senior Sales Engineer

emil.zegers@solace.com

+31 (0)6 19929703

