Distributed tracing in ABAP with OpenTelemetry

June 2024

Booking.com

Why are we here?

TO SHARE

TO COLLECT FEEDBACK

WE WANT THIS OUT OF THE BOX

Agenda

- What is telemetry?
- Open Telemetry
- Open Telemetry at SAP
- Telemetry in ABAP
- Implementation pattern
- Further actions
- Questions?

01

What is telemetry?

Booking.com

What is telemetry?

Traces, logs, and metrics are essential components of observability, providing valuable insights into the behavior and performance of systems.

Traces

Detail view how request goes through multiple components

Logs

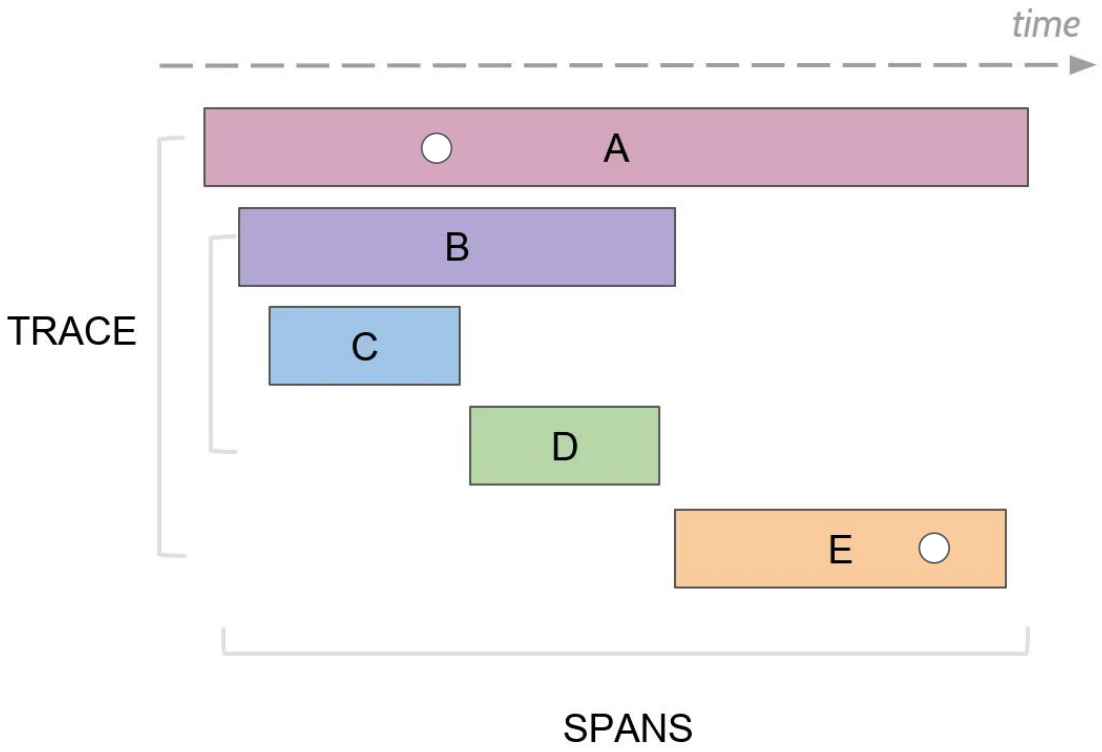
Events or messages recorded by a system

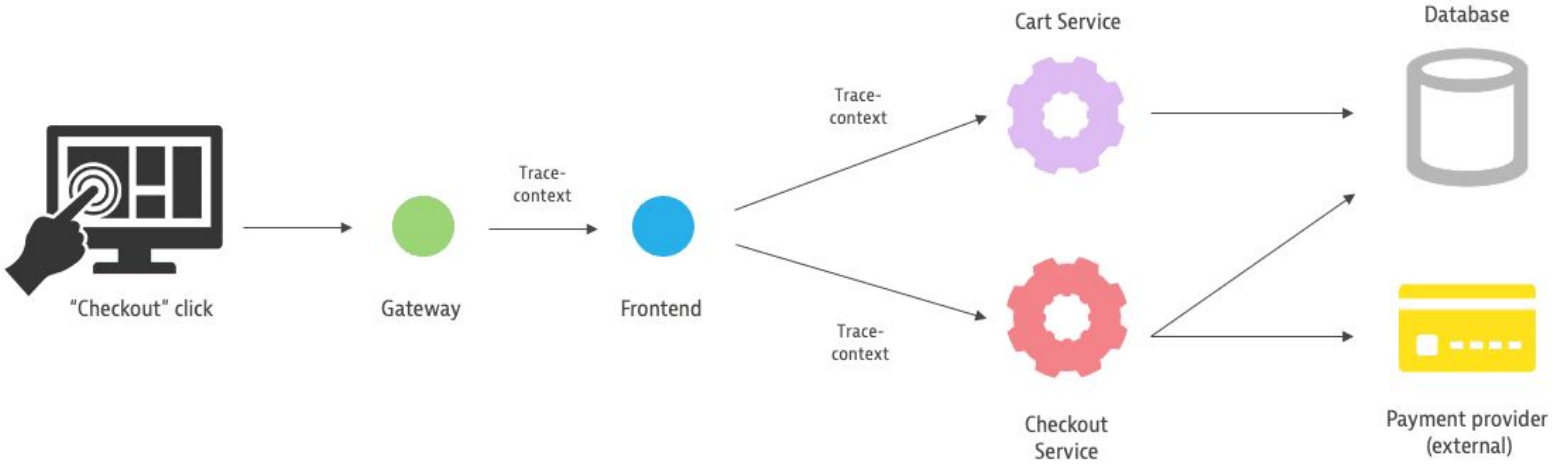
Metrics

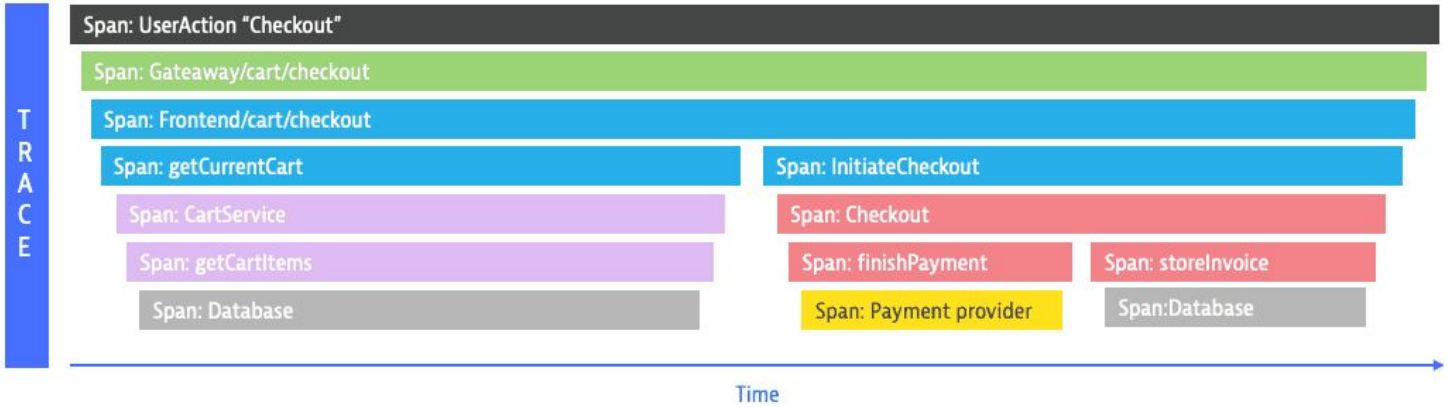
Quantitative measurements that provide insight into the performance and behavior of a system over time

Traces

- Spans
- Traces
- Events
- Links







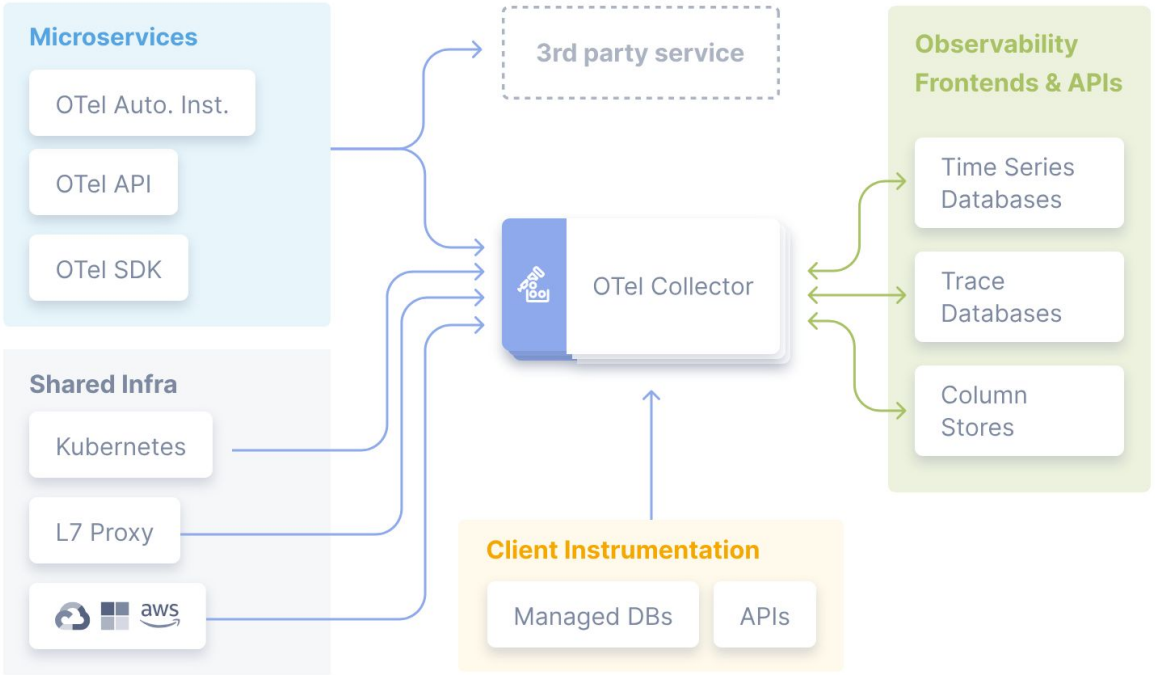
https://docs.dynatrace.com/docs/observe-and-explore/distributed-traces/concepts

02

Open Telemetry

Booking.com

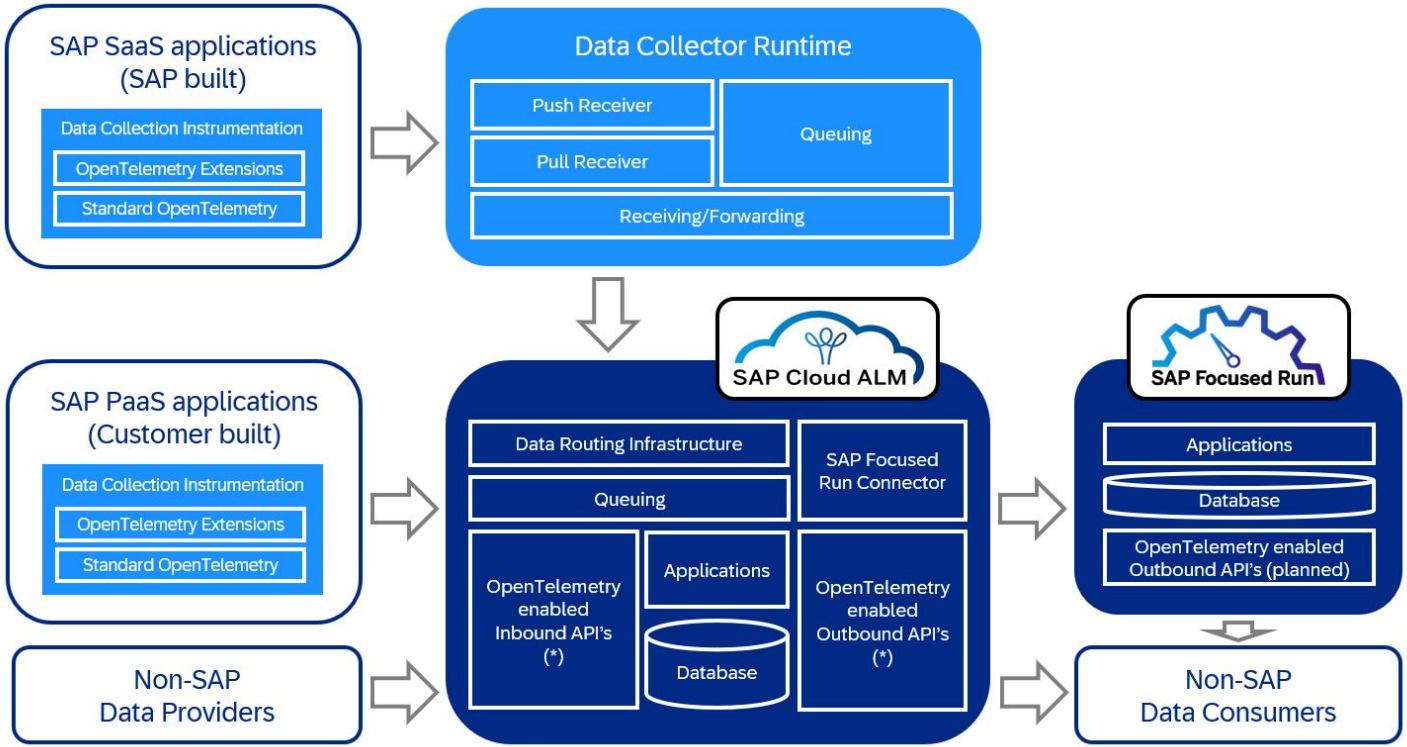
Open Telemetry



- Industry standard
- >40 vendors
- SDK in 12 languages
- Highly extendable
- Various integrations

Open Telemetry: collector

Open Telemetry @ SAP

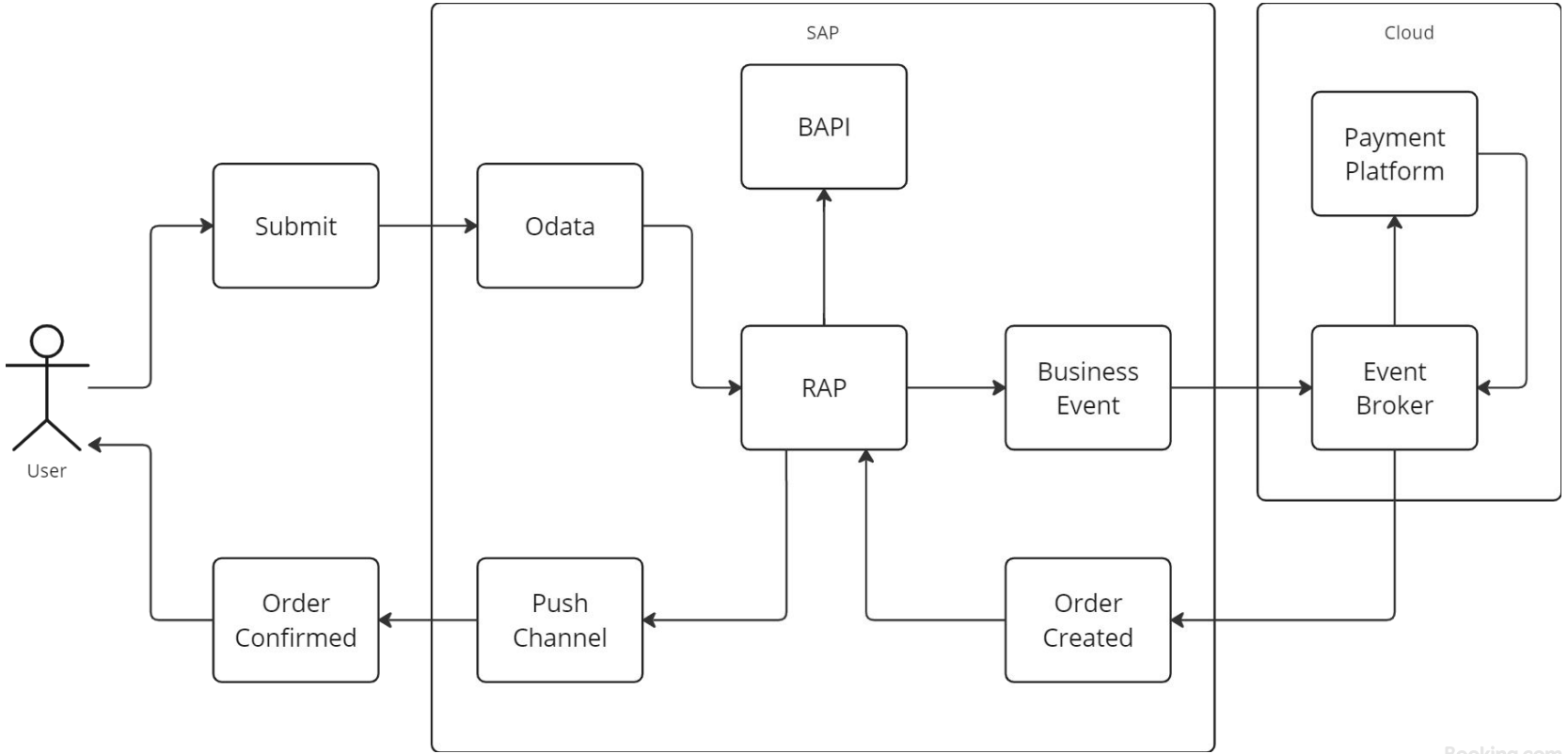


03

Distributed tracing in ABAP

Booking.com

Sample Process Flow



How do you trace ABAP apps?

Booking.com

Open Telemetry in ABAP?

System capabilities

- Native JSON/XML identity serialisation
- HTTP client
- WebSocket
- TCP
- MQTT/AMQP clients
- FS / DB capabilities

Community projects

- abap-opentelemetry by Lars Hvam
- abap-protobuf by Lars Hvam

```
CLASS ltcl_test IMPLEMENTATION.

  METHOD zif_otlp_span_processor~on_end.
* todo
    RETURN.
  ENDMETHOD.

  METHOD trace.

    DATA(lo_cut) = NEW zcl_otlp_tracer_provider( ).
    lo_cut->add_span_processor( me ).

    DATA(lo_span) = lo_cut->get_tracer( 'tracerName' )->start_span(
      iv_name = 'spanName'
      iv_kind = zif_otlp_model_trace=>gc_span_kind-producer ).

    lo_span->end( ).

  ENDMETHOD.

ENDCLASS.
```

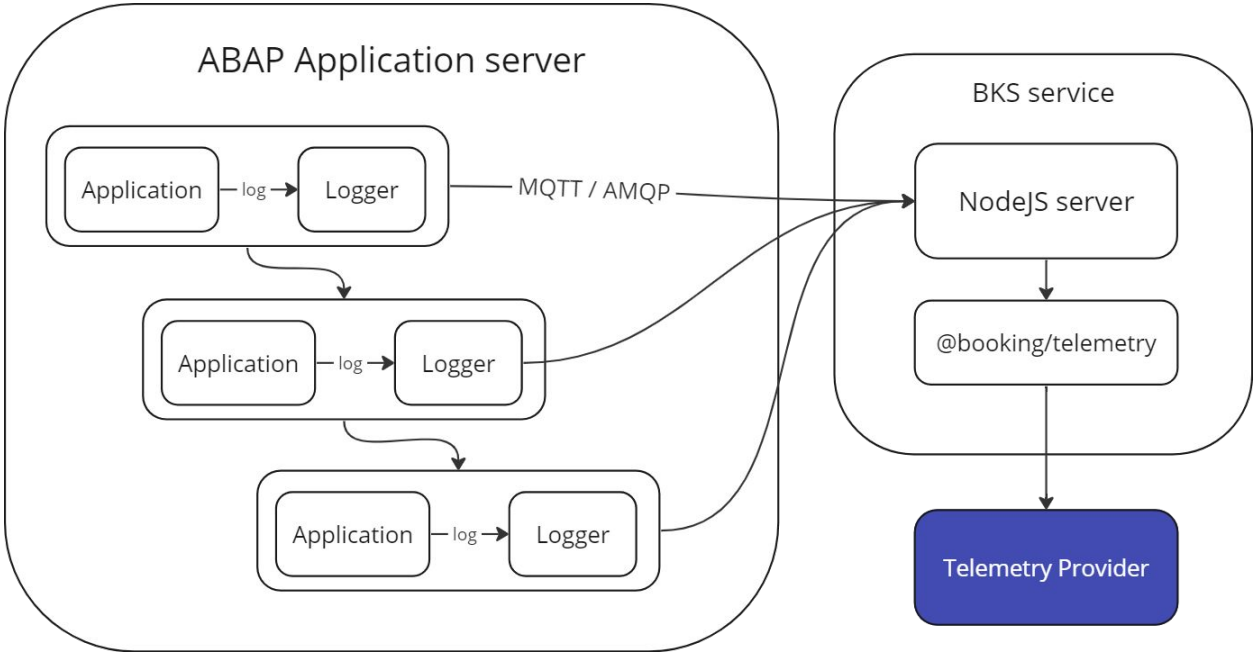
OTel @ Booking.com: fundamentals

- Minimal time spent on serialization
- Native SAP clients, no custom protocols
- HTTP calls are too expensive, because they are synchronous
- Real-time analytics
- Ended span must appear even in case of short dump or rollback
- Extendable

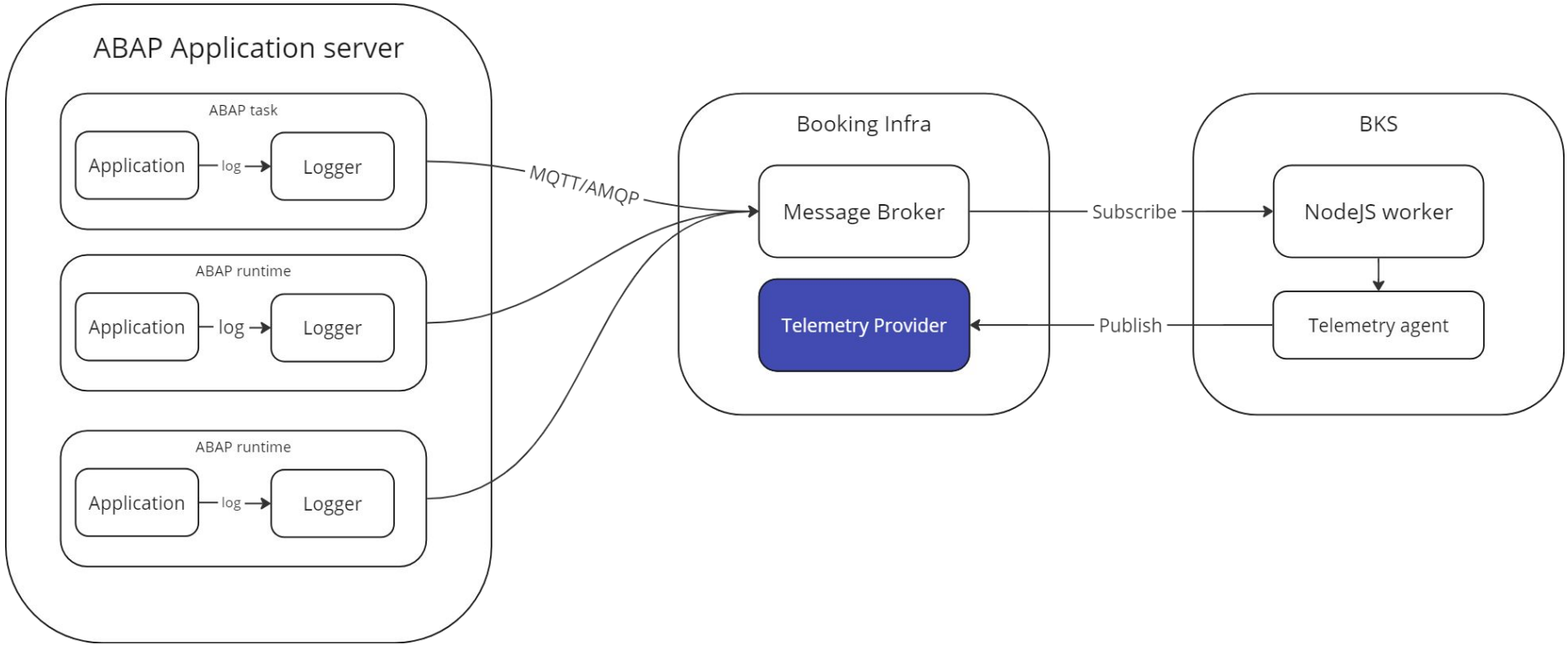
Demo

Booking.com

OTel @ Booking.com: proxy service



OTel @ Booking.com: using message broker



Abap2otel: own way to the same goal

```
if x_mqtt eq abap_true.
  trace->use( )->mqtt( version = version )->destination(
    destination = conv #( m_dest )
    handler = cond #( when out is bound then new lcl_mqtt_handler( ) ) ).
endif.

if x_http eq abap_true.
  trace->use( )->http( version = version )->destination( destination = conv #( h_dest ) ).
endif.

data(root) = trace->start_span( 'ABAP telemetry: performance test' ).
data(root_context) = root->span_context( ).

DO p_times TIMES.

  DATA(span) = trace->start_span(
    name = |Stress test { sy-index }|
    context = root_context
    ).

  data span_prev like span.
  if span_prev is bound.
      span->link(
        exporting
          context    = span_prev->span_context( )
          attributes = value #( ( key = 'link.type' value = 'previous' ) )
      ).
  endif.
  span_prev = span.



  DATA(child_span) = trace->start_span(
                      context = span->span_context( )
                       name    = 'Child span'
                     ).
  child_span->end( ).

  span->log( |Stress test { sy-index } message|  ).
  span->log( name = |Test event with a custom attribute| attributes = value #( ( key = 'foo' value = 'bar' ) )  ).
```


```
{
  "SPANS": [
    {
      "NAME": "Stress test 1",
      "SPAN_ID": "7112504E91E38BDC",
      "TRACE_ID": "0A7425B8447B1EEEBCA0B930058F88EB",
      "PARENT_SPAN_ID": "441DBE0D0E6A0635",
      "START_TIME": "2024-04-02T15:29:26.72699Z",
      "END_TIME": "2024-04-02T15:29:31.711294Z",
      "STATUS": "UNSET",
      "LINKS": [
        {
          "TRACE_ID": "0A7425B8447B1EEEBCA0B930058F88EE",
          "SPAN_ID": "7112504E91E38BDD",
          "ATTRS": [{ "KEY": "attr-name", "VALUE": "attr-value" }]
        }
      ],
      "ATTRS": [
        {
          "KEY": "sap.transaction_id",
          "VALUE": "E2D0204E085E0170E0066030CEA1B42C"
        },
        {
          "KEY": "sap.connection_id",
          "VALUE": "0A7425B8447B1EEEBCA0B3646246A8EB"
        },
        {
          "KEY": "sap.context_id",
```

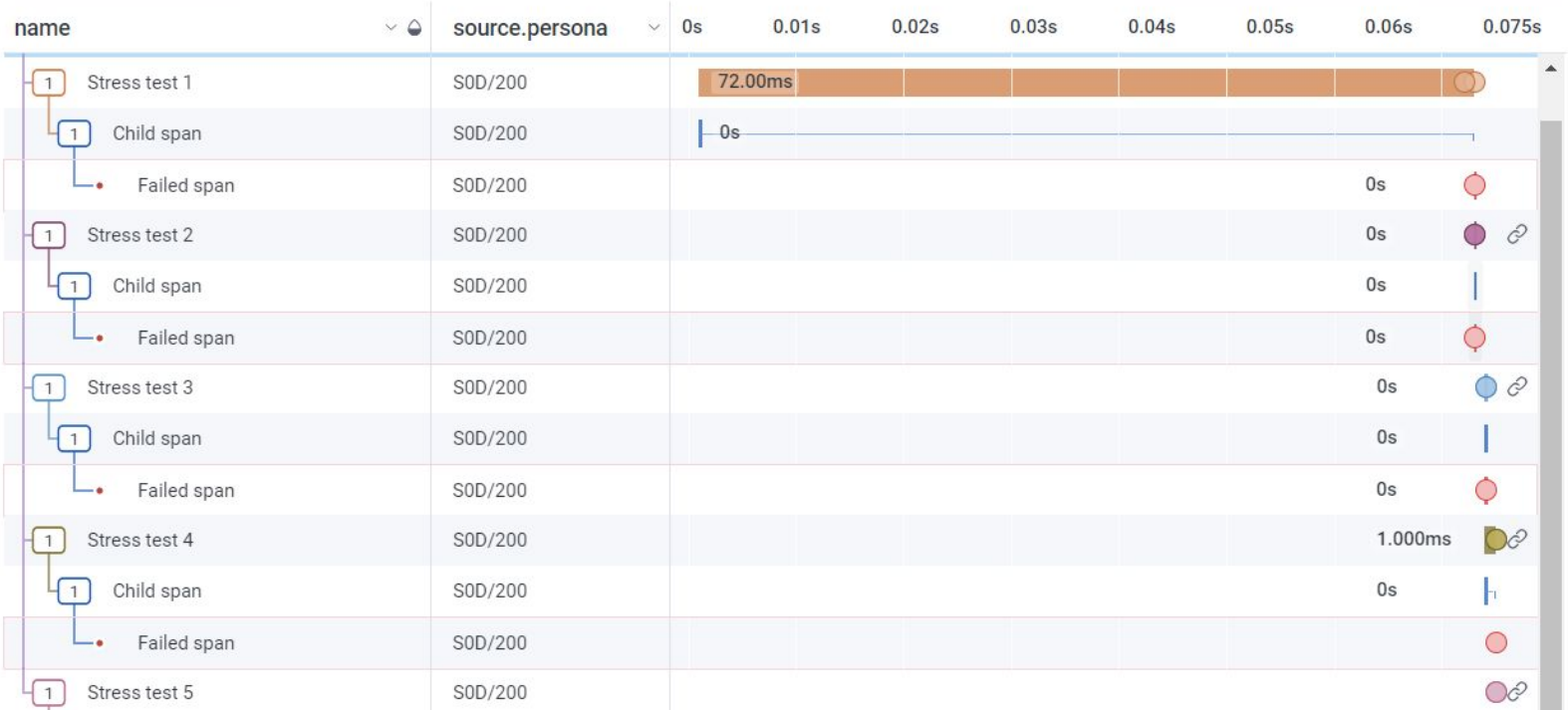
Trace 0a7425b8447b1eeebe9bb3378f7f88eb

Trace summary 16 spans at Apr 12 2024 17:01:55 UTC+02:00 (75.00ms)

Search spans

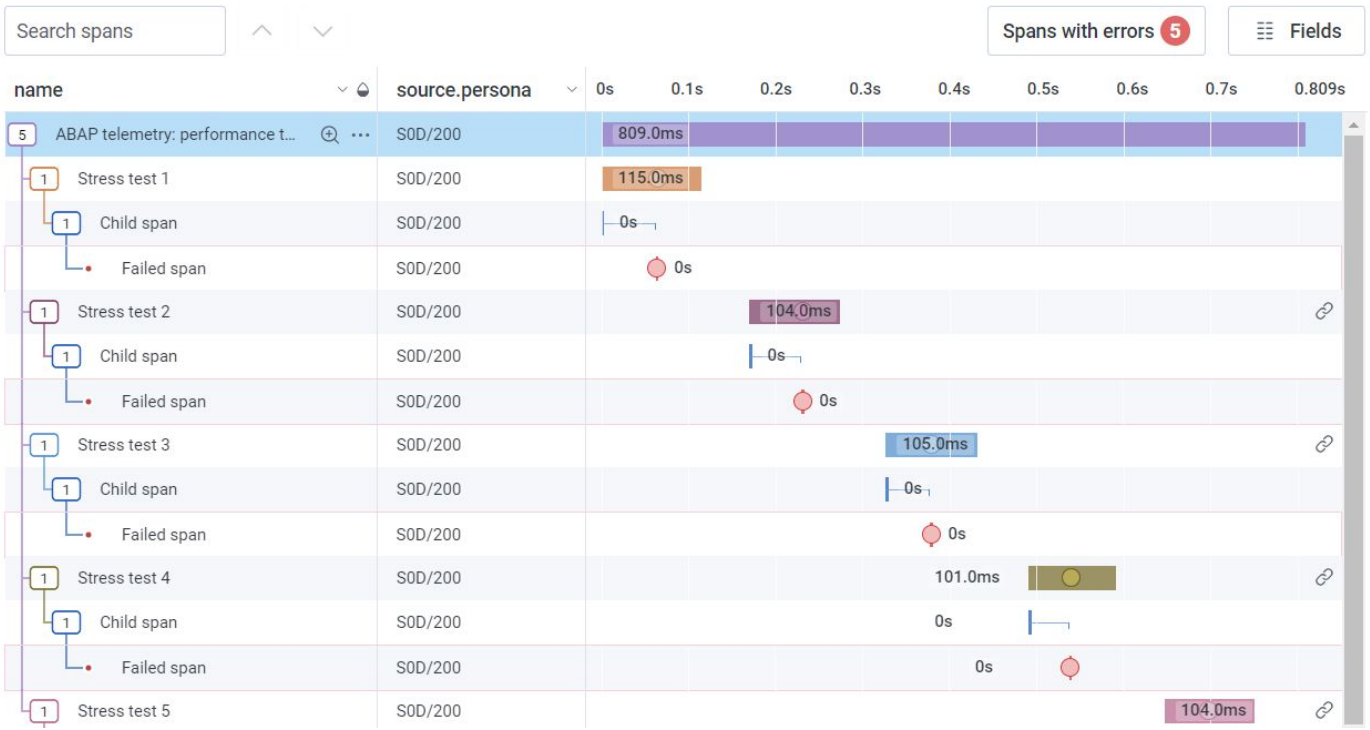
Spans with errors 5

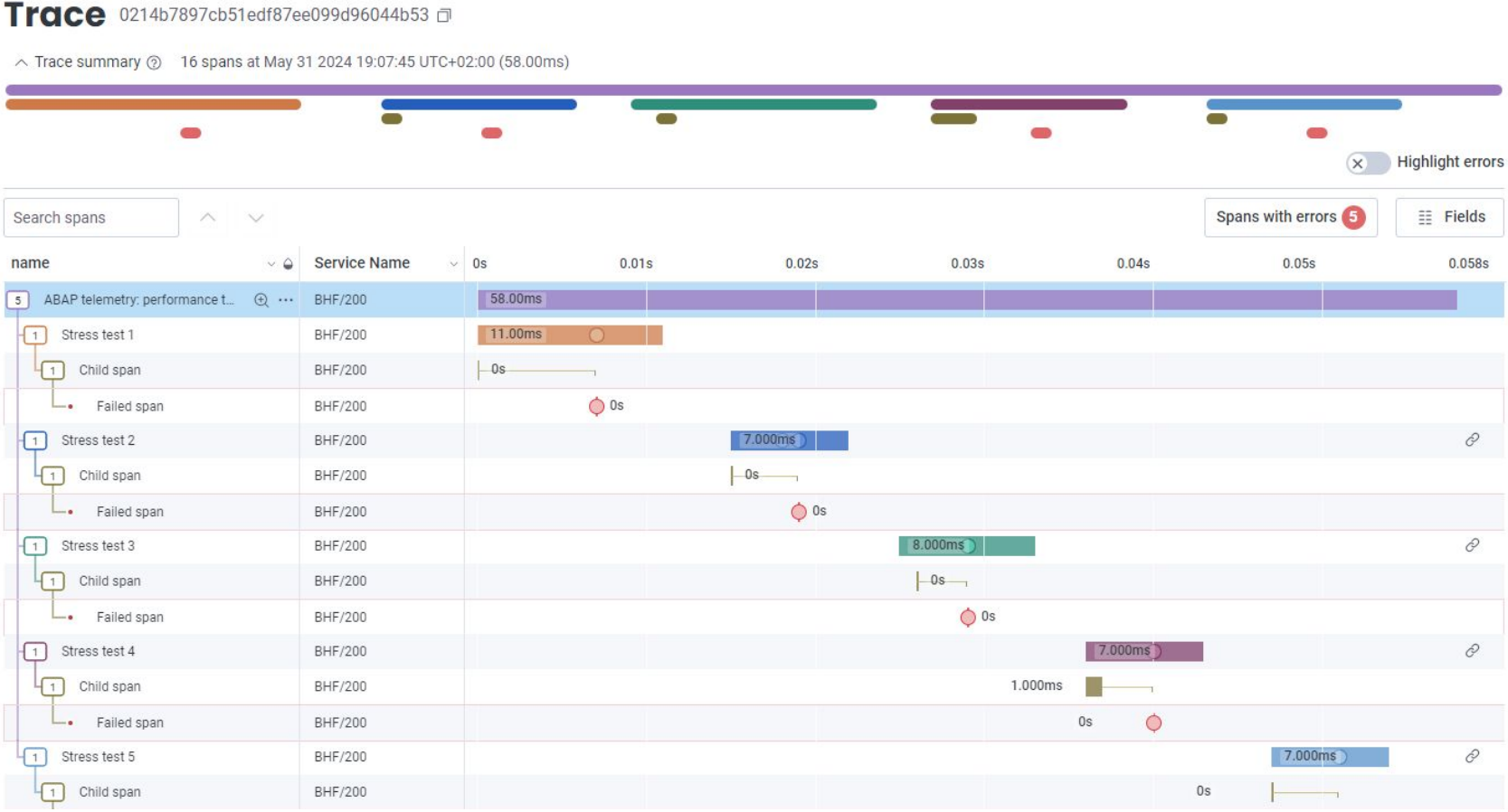
Fields



Trace 06b02025cd671edebe9b01858887866d

Trace summary 16 spans at Apr 12 2024 16:22:10 UTC+02:00 (809.0ms)




Trace
0214b7897cb51edf87ee099d96044b53
Trace summary
16 spans at May 31 2024 19:07:45 UTC+02:00 (58.00ms)
Highlight errors
Search spans
Spans with errors 5
Fields
name
Service Name
0s
0.01s
0.02s
0.03s
0.04s
0.05s
0.058s
ABAP telemetry: performance t...
BHF/200
58.00ms
Stress test 1
11.00ms
Child span
0s
Failed span
Stress test 2
7.000ms
Stress test 3
8.000ms
Stress test 4
7.000ms
1.000ms
Stress test 5
7.000ms

How did we achieve it?

- Custom telemetry SDK
- Custom telemetry protocol (asJSON)
- WebSocket based connectivity
- Custom NodeJS proxy middleware

What do we miss?

- Truly Asynchronous code
- OpenTelemetry SDK
- Protobuf
- gRPC
- Kafka

Thank you!

Booking.com