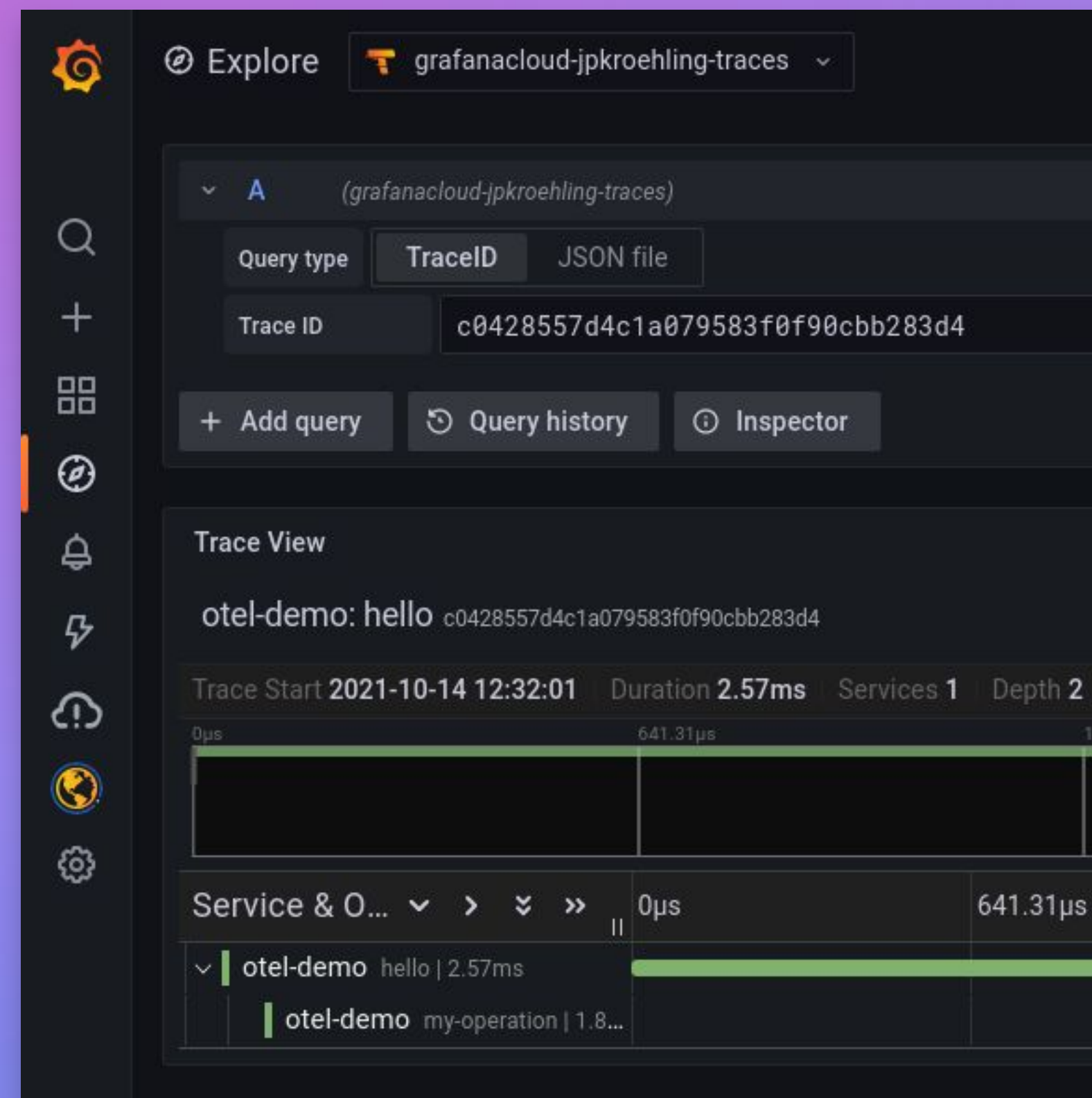




Tudo o que você PRECISA saber sobre OpenTelemetry



Juraci Paixão Kröhling
Engenheiro de software
@jpkrohling



Quem sou eu



Juraci Paixão Kröhling
Engenheiro de software
@jpkrohling

Programa

- **Origens do projeto OpenTelemetry**
- **Padrões e especificações**
- **Instrumentação**
- **Conduíte (pipeline)**
- **Futuro**
- **Perguntas e respostas**



OpenTracing

Objetivo: servir de padrão para rastreamento distribuído

- Especificação: o que é um rastreador, rastro, trecho, ...
- Convenções semânticas: quais atributos usar em quais casos
- API de instrumentação para diversas linguagens
- Bibliotecas de instrumentação



OpenCensus

Objetivo: ser “a” biblioteca para captura de telemetria

- SDKs para diversas linguagens: API e rastreador fazem parte da mesma biblioteca
- Conduíte para dados de telemetria: opencensus-service
- Suporte além de rastreamento: métricas disponível, suporte a logs viria em seguida
- Bibliotecas de instrumentação



OpenTelemetry

OpenTracing + OpenCensus

- Especificação
- Convenções semânticas
- API de instrumentação para diversas linguagens
- API e SDK em módulos separados
- Bibliotecas de instrumentação
- Conduíte para processamento de dados de telemetria:
`opentelemetry-collector`



Padrões e especificações

Especificação da API

- ✓ Como a API de instrumentação deve ser, independente do SDK usado

Especificação do SDK

- ✓ Como os SDKs devem se comportar, independente da linguagem e implementação

Especificação de dados

- ✓ Interface description language (IDL), especificando como os dados devem ser formatados e transmitidos ou recebidos



Especificação de dados

OTLP

OpenTelemetry Line Protocol, um conjunto de arquivos proto, definindo tanto o transporte quanto o conteúdo da mensagem

XKCD #927

Aposto que você pensou no XKCD #927 em algum momento desta apresentação



HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.



Especificação de dados

OTLP

OpenTelemetry Line Protocol, um conjunto de arquivos proto, definindo tanto o transporte quanto o conteúdo da mensagem

XKCD #927

Aposto que você pensou no XKCD #927 em algum momento desta apresentação



Instrumentação

Instrumentação manual

- ✓ Utilizando diretamente a API de instrumentação

Bibliotecas de instrumentação

- ✓ Utilizando bibliotecas que integram com frameworks populares, como Spring, Servlets, JAX-RS, ...

Instrumentação automática

- ✓ Geralmente por meios de manipulação de bytecode



Demonstração: instrumentação

The screenshot displays the Visual Studio Code interface with the file `GreetingResource.java` open. The file contains the following code:

```
9 import io.opentelemetry.api.trace.Span;
10 import io.opentelemetry.api.trace.Tracer;
11
12 @Path("/hello")
13 public class GreetingResource {
14
15     @Inject
16     Tracer tracer;
17
18     @GET
19     @Produces(MediaType.TEXT_PLAIN)
20     public String hello() {
21         Span span = tracer.spanBuilder("my-operation").startSpan();
22         // do something interesting...
23         span.end();
24         return "Hello RESTEasy";
25     }
26 }
```

The Explorer sidebar on the left shows the project structure, including the `src/main/java/com/grafana` directory where `GreetingResource.java` is located. The terminal at the bottom shows the output of a `curl` command:

```
jpkroehling@caju ~/Projects/src/github.com/jpkrohling/otel-demo $ curl localhost:8080/hello
Hello RESTEasyjpkroehling@caju ~/Projects/src/github.com/jpkrohling/otel-demo $
```

The Problems panel on the left shows a single error message:

```
End time      : 2021-10-14 10:32:01.877598666 +0000 UTC
Status code   : STATUS_CODE_UNSET
Status message :
Attributes:
-> http.client_ip: STRING(127.0.0.1)
-> http.status_code: INT(200)
-> http.scheme: STRING(http)
-> http.host: STRING(localhost:8080)
-> http.method: STRING(GET)
-> http.route: STRING(/hello)
-> http.flavor: STRING(1.1)
-> http.user_agent: STRING(curl/7.76.1)
-> http.target: STRING(/hello)
```

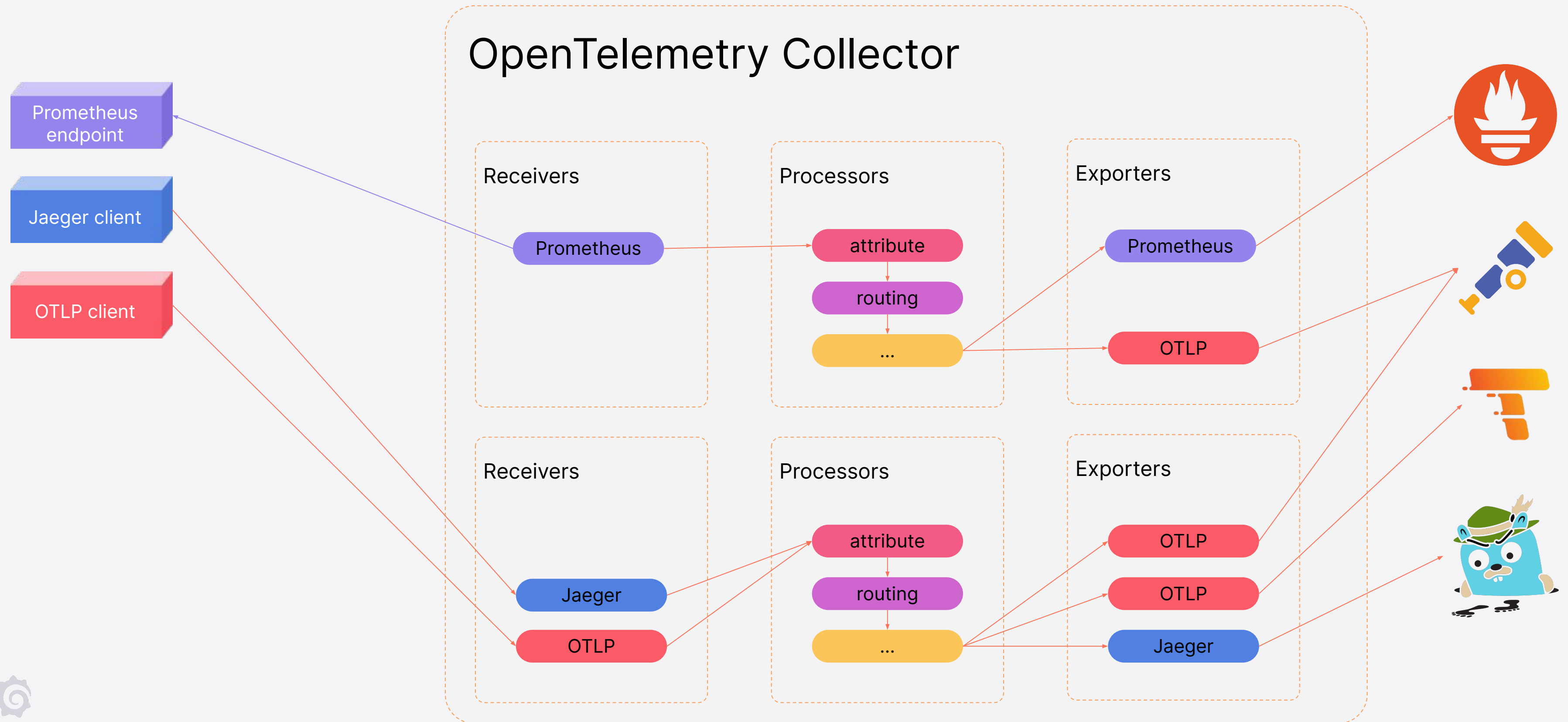


Collector: conduíte
para dados de
telemetria



Middleware

→ Escrita
→ Leitura



Existe uma
palestra dedicada
a modos de
deployment na
KubeCon NA 2021



Componentes

Receivers

awscontainerinsightreceiver
awsecscontainermetricsreceiver
awsxrayreceiver
carbonreceiver
cloudfoundryreceiver
collectdreceiver
dockerstatsreceiver
dotnetdiagnosticsreceiver
filelogreceiver
fluentforwardreceiver
googlecloudpubsubreceiver
googlecloudspannerreceiver
hostmetricsreceiver
httpdreceiver
influxdbreceiver
jaegerreceiver
jmxreceiver
k8sclusterreceiver
kafkametricsreceiver
kafkareceiver
kubletstatsreceiver
memcachedreceiver
mongodbatlasreceiver
mysqlreceiver
nginxreceiver
opencensusreceiver
otlpreceiver
podmanreceiver
prometheusexecreceiver
prometheusreceiver
receivercreator
redisreceiver
sapmreceiver
scraperhelper
signalfxreceiver
simpleprometheusreceiver
splunkhecreceiver
statsdreceiver
syslogreceiver
tcplogreceiver
udplogreceiver
wavefrontreceiver
windowsperfcountersreceiver
zipkinreceiver
zookeeperreceiver

Processors

attributesprocessor
cumulativetodeltaprocessor
deltatorateprocessor
filterprocessor
groupbyattrsprocessor
groupbytraceprocessor
k8sattributesprocessor
metricsgenerationprocessor
metricstransformprocessor
probabilisticsamplerprocessor
resourcedetectionprocessor
resourceprocessor
routingprocessor
spanmetricsprocessor
spanprocessor
tailsamplingprocessor

Exporters

alibabacloudlogserviceexporter
awscloudwatchlogsexporter
awsemfexporter
awskinesisexporter
awsprometheusremotewriteexporter
awsxrayexporter
azuremonitorexporter
carbonexporter
datadogexporter
dynatraceexporter
elasticexporter
elasticsearchexporter
f5cloudexporter
fileexporter
googlecloudexporter
googlecloudpubsubexporter
honeycombexporter
humioexporter
influxdbexporter
jaegerexporter
kafkaexporter
loadbalancingexporter
logzioexporter
lokiexporter
newrelicexporter
observiqexporter
opencensusexporter
otlpexporter
prometheusexporter
prometheusremotewriteexporter
sapmexporter
sentryexporter
signalfxexporter
skywalkingexporter
splunkhecexporter
stackdriverexporter
sumologicexporter
tanzuobservabilityexporter
zipkinexporter



Futuro

Métricas!

Logs!

eBPF?



Obrigado pela atenção!

Onde me encontrar:

[jpkrohling](#) @ twitter and github

[jpkroehling](#) @ Slack

Faça parte do projeto:



#otel-collector (CNCF)
#observabilidade