



All you **NEED** to know about OpenTelemetry

Cloud Native Computing Linz Meetup
2021-04-27

Juraci Paixão Kröhling
Software Engineer
@jpkrohling



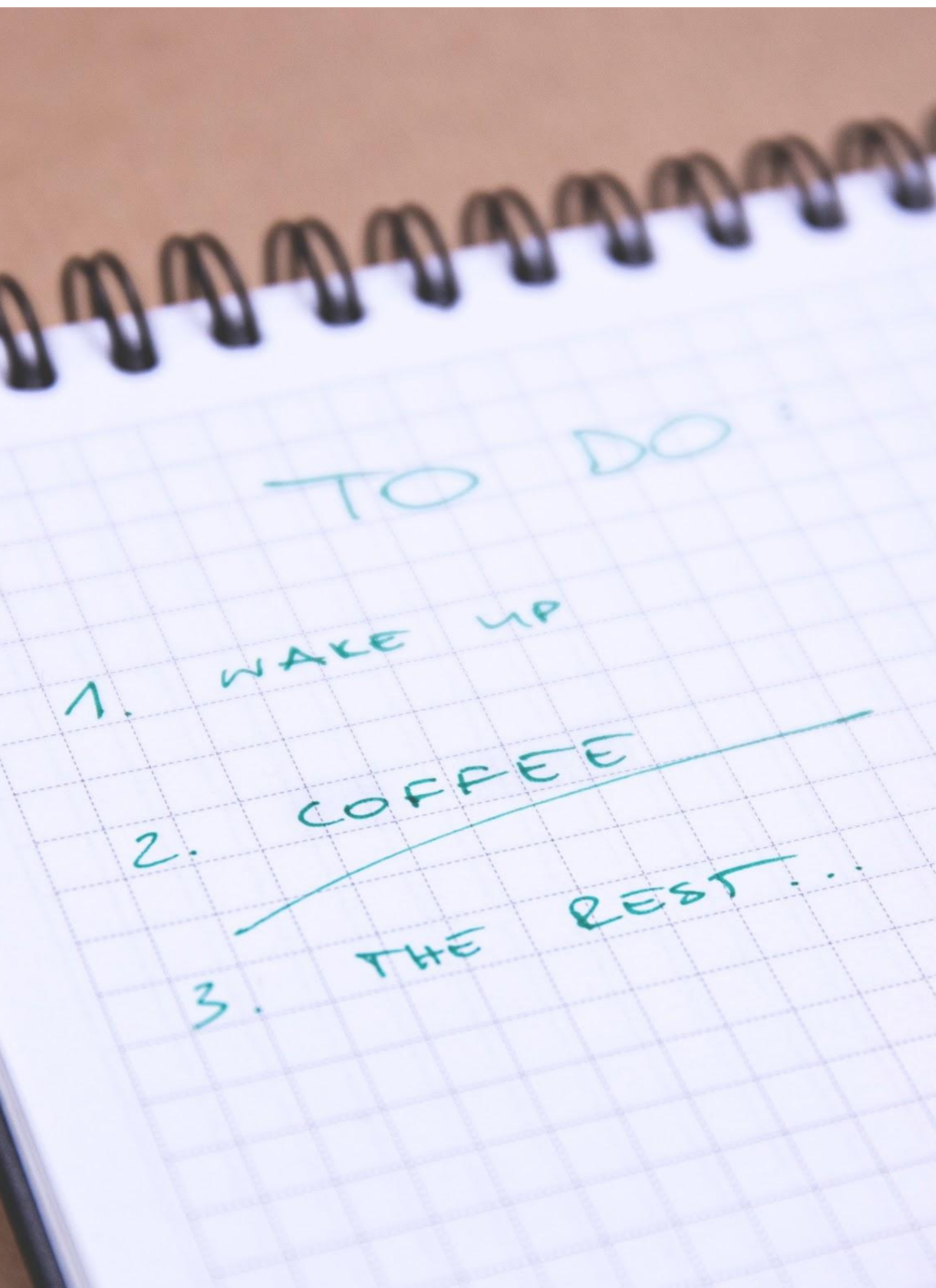


Hi 🖐️, I'm Juraci!

Software Engineer @ Red Hat, distributed
tracing team

Maintainer on the Jaeger project

Member of the OpenTelemetry project



Agenda

OpenTelemetry origins

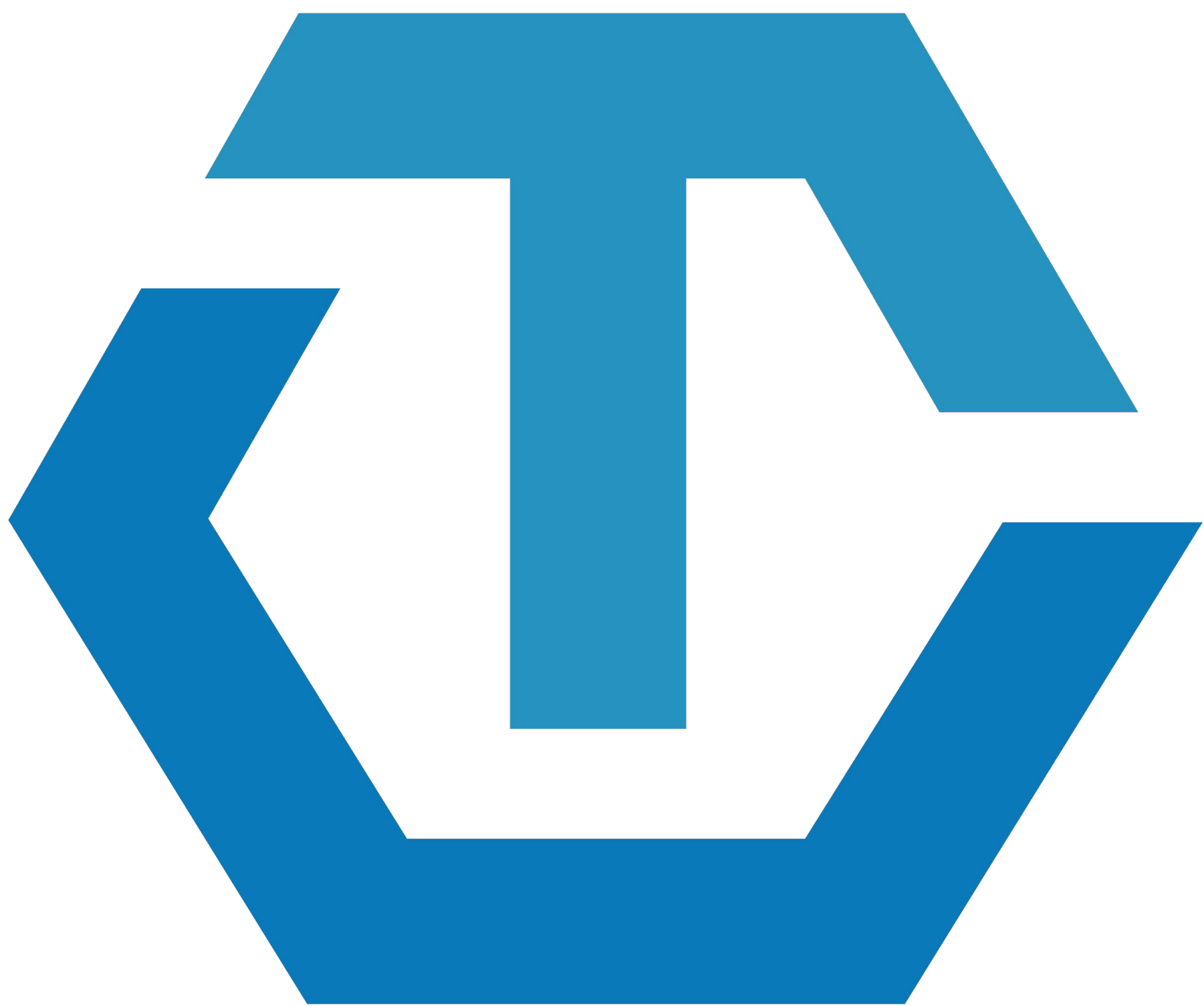
Specification and standards

Instrumentation

Backend

Future

Q&A



OpenTracing

Aimed to be the standard for distributed tracing

Specification (what's a trace, tracer, span, ...)

Semantic conventions

API (Java, Go, Python, ...)

Instrumentation libraries

OpenCensus

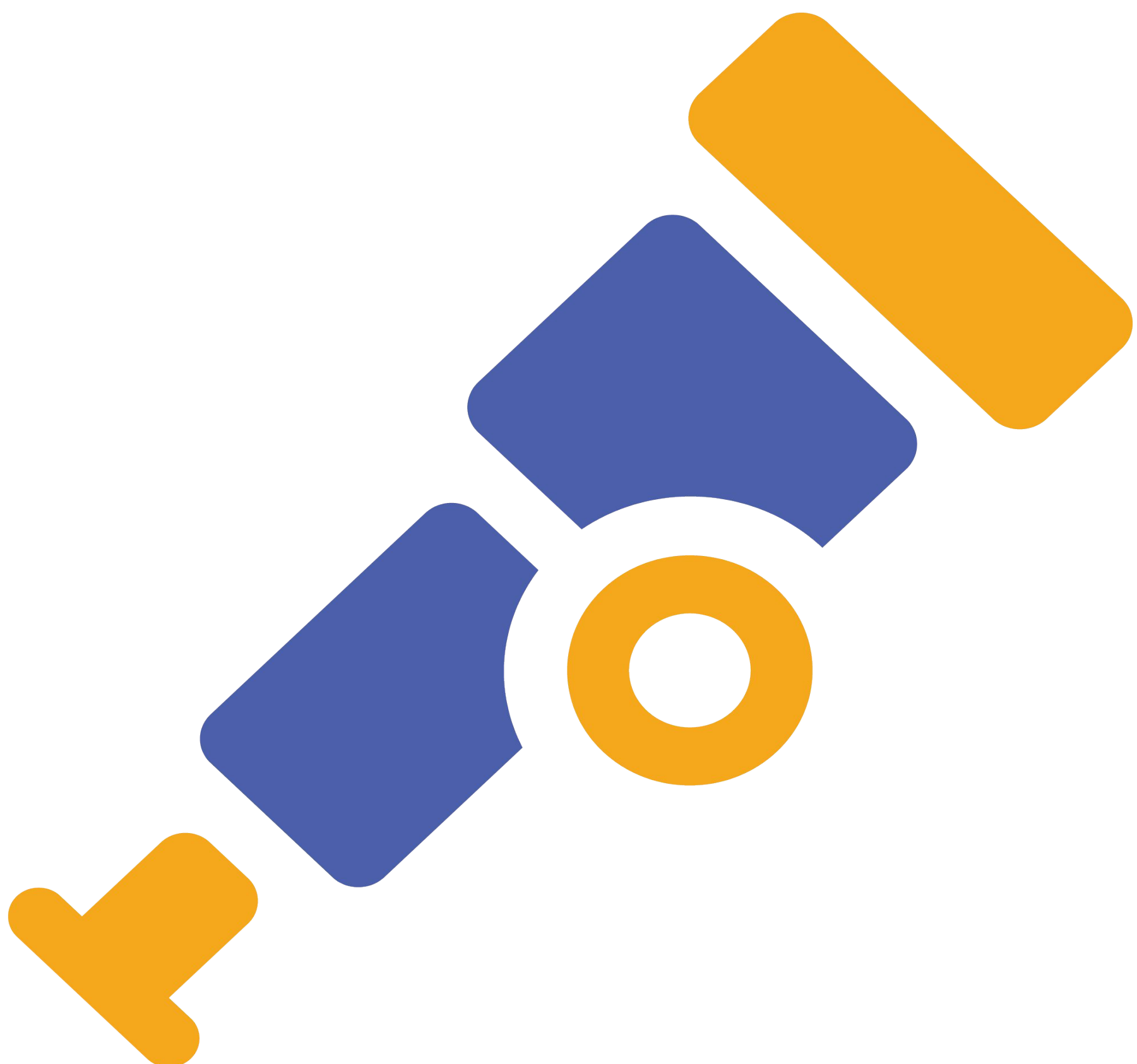
Aimed to be the single library for telemetry

Language specific SDKs

Backend “service”

Not only for tracing

Instrumentation libraries



OpenTelemetry

OpenTracing + OpenCensus

Specification (what's a trace, tracer, span, ...)

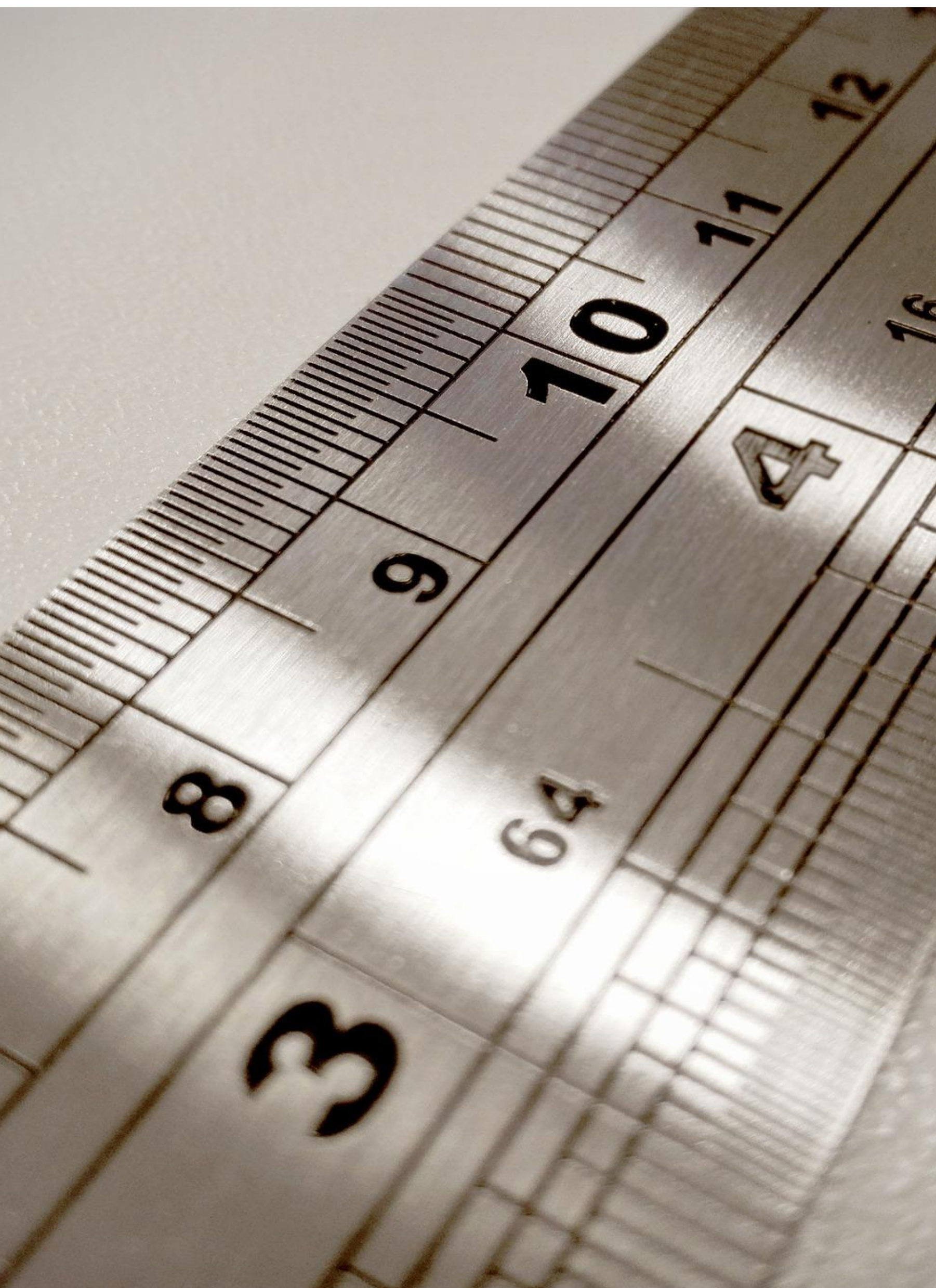
Semantic conventions

API (Java, Go, Python, ...)

Language specific SDKs

Backend “service”, now known as “collector”

Not only for tracing

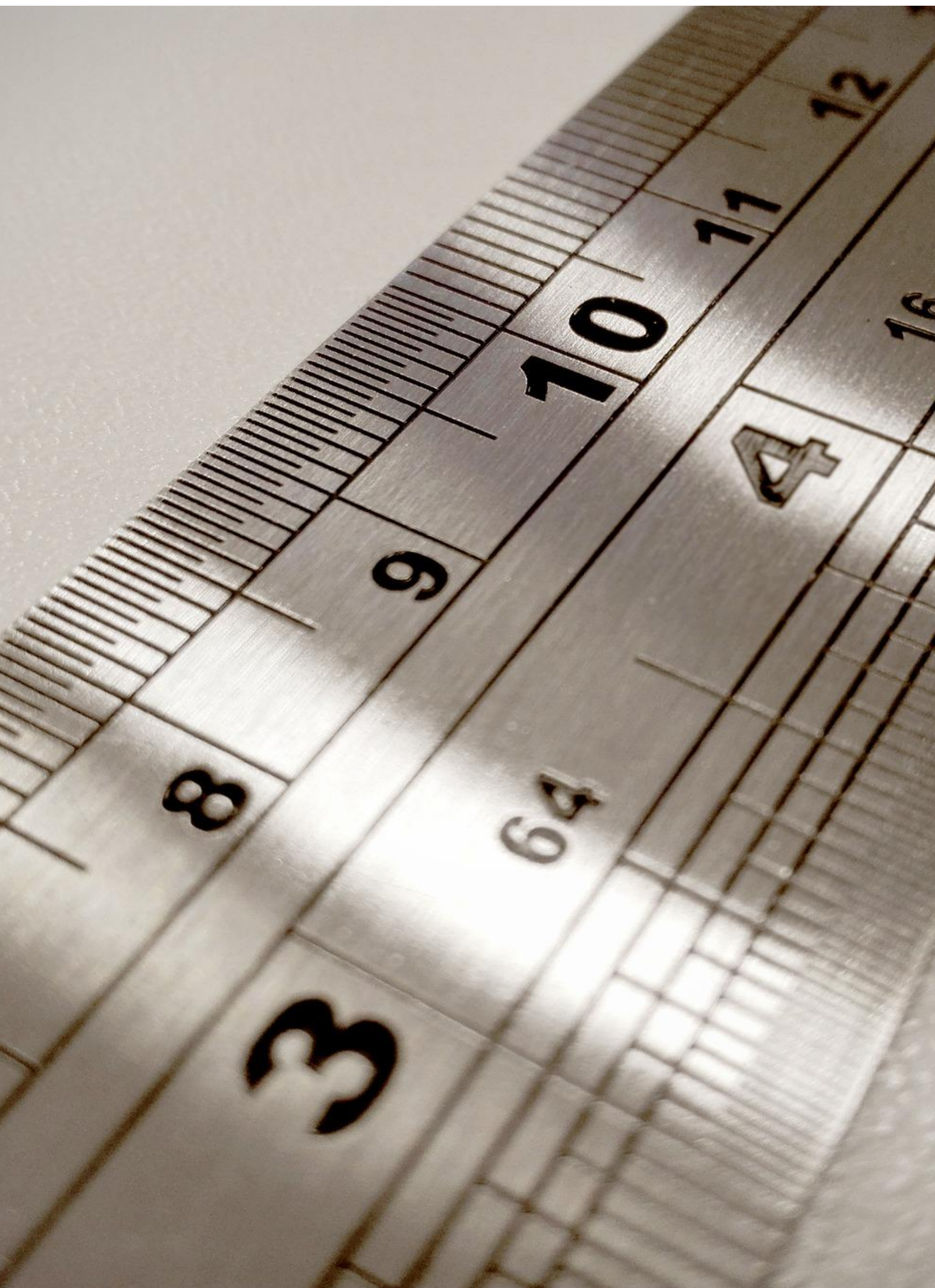


Specification and standards

API specification

SDK specification

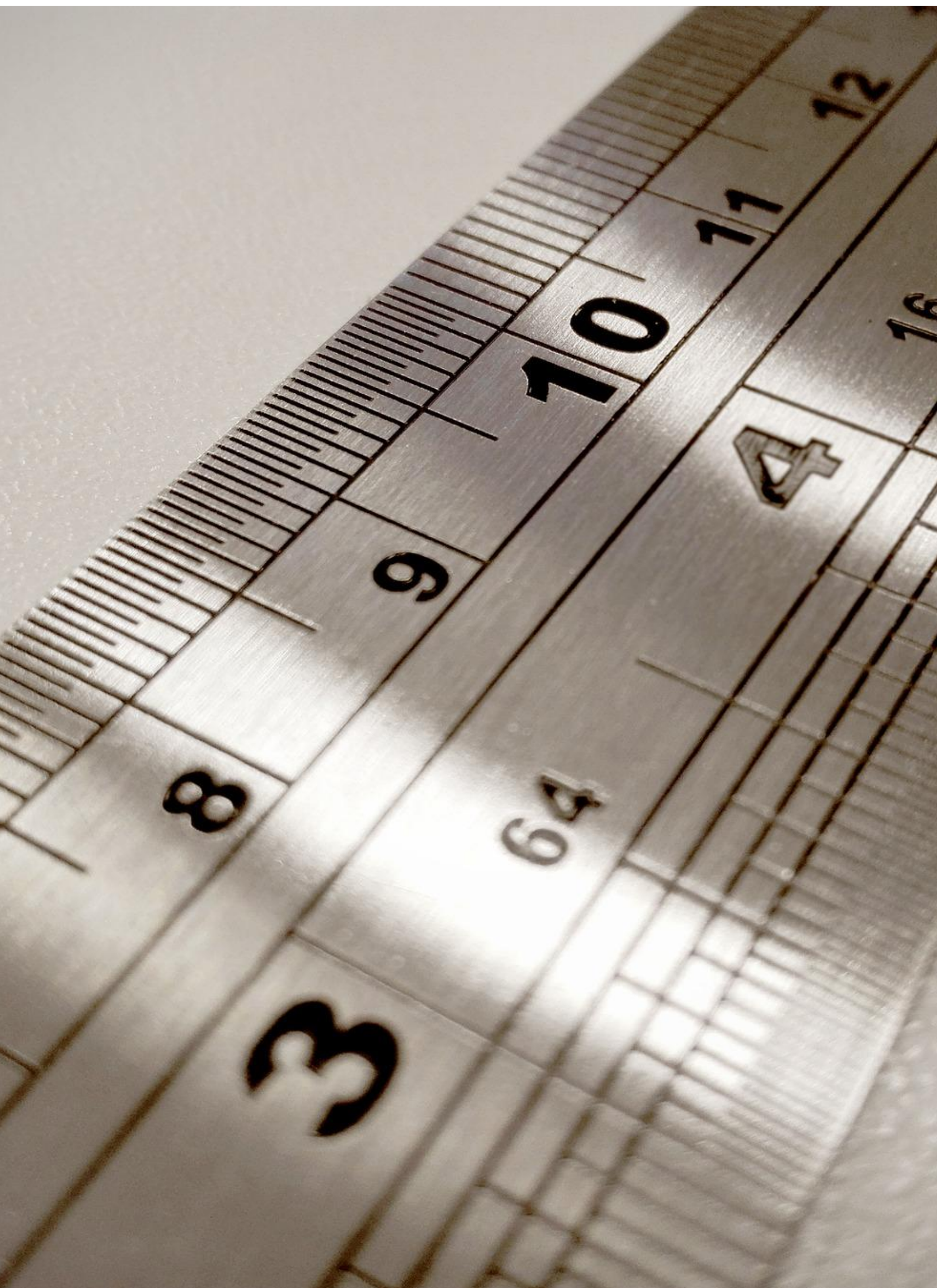
Data specification



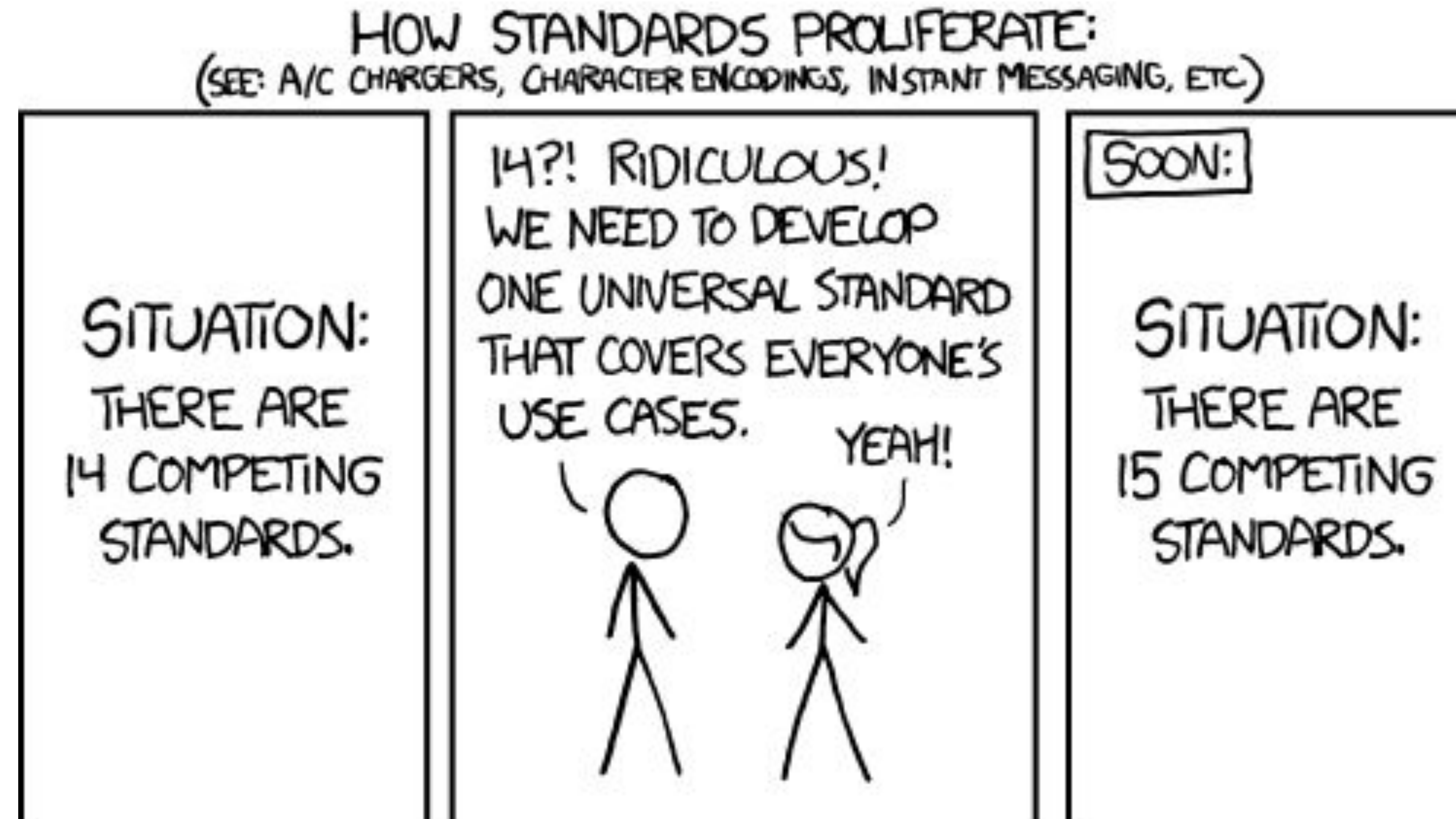
Data specification

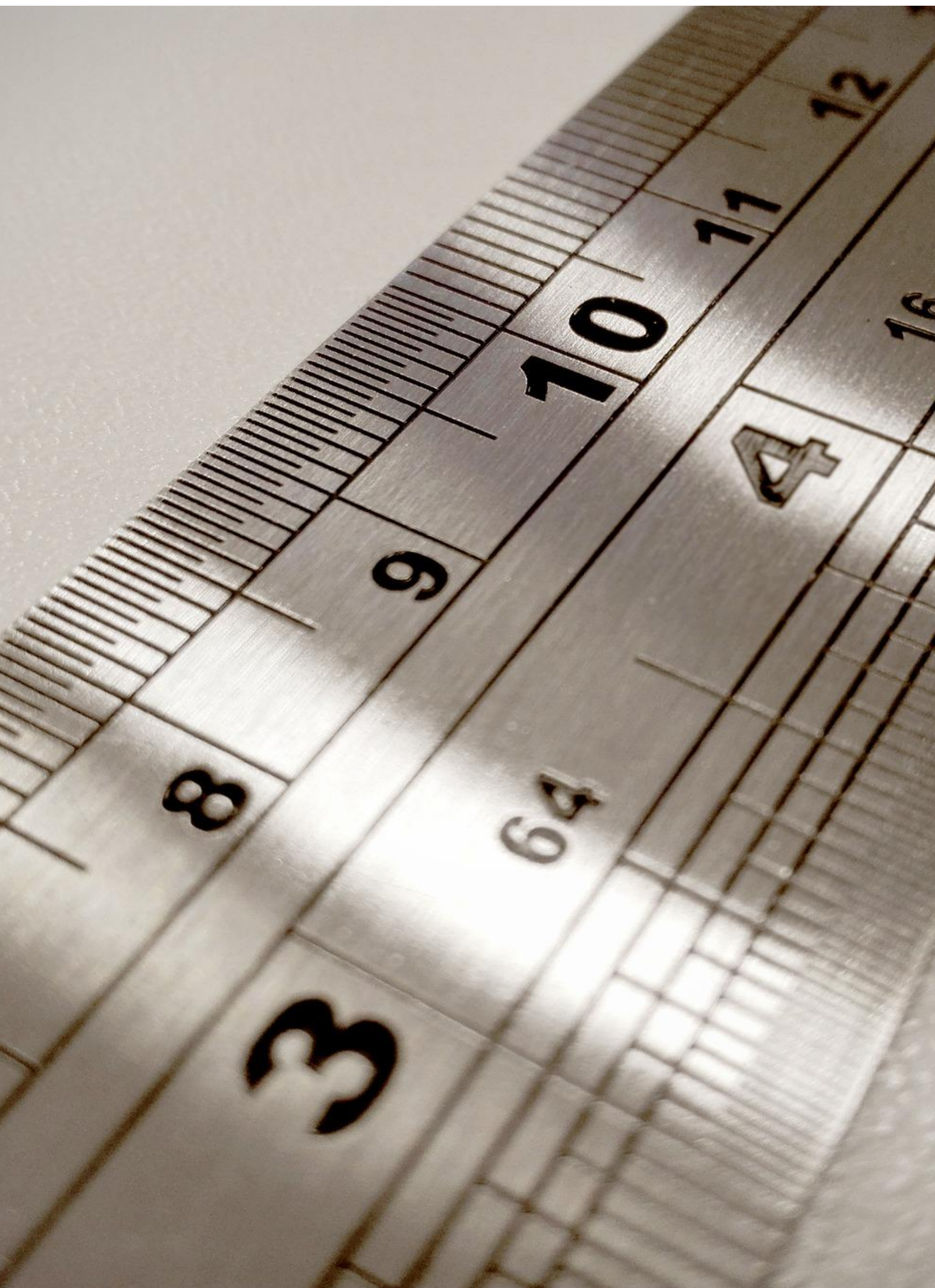
Obligatory XKCD [#927](#) reference

OpenTelemetry Line Protocol (OTLP)



Data specification

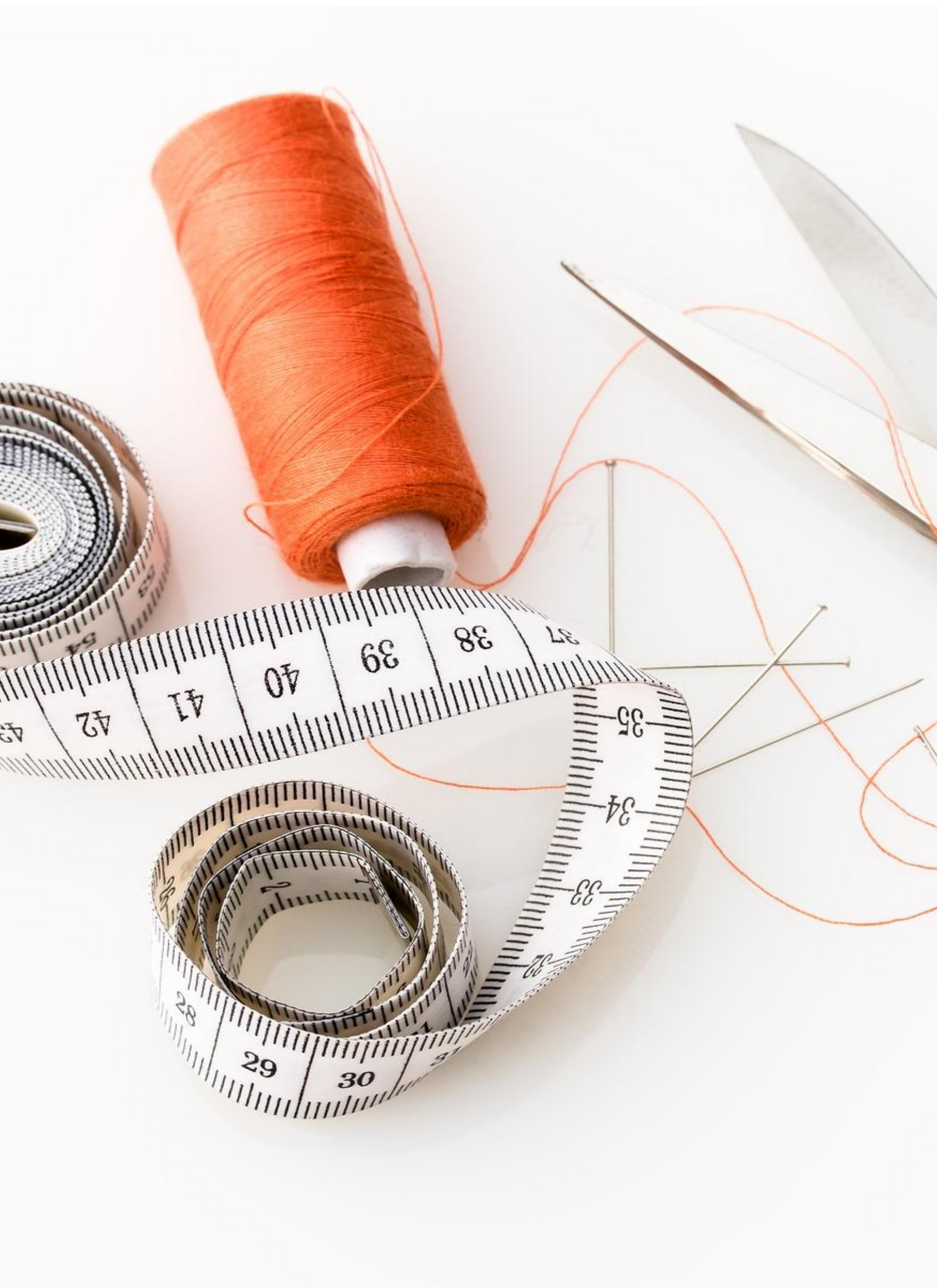




Data specification

Obligatory XKCD [#927](#) reference

OpenTelemetry Line Protocol (OTLP)

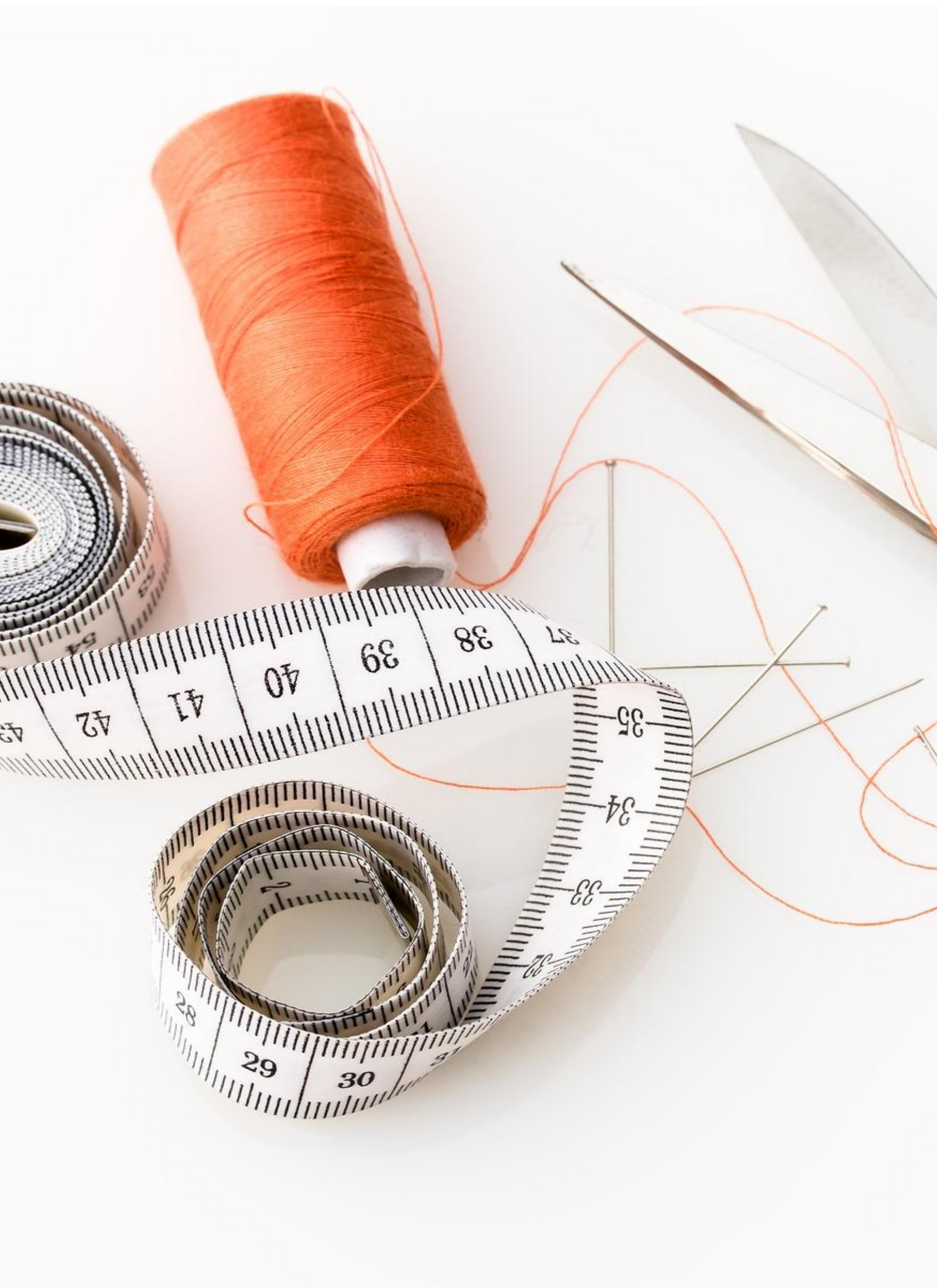


Instrumentation

Manual instrumentation

Instrumentation libraries

Automatic instrumentation



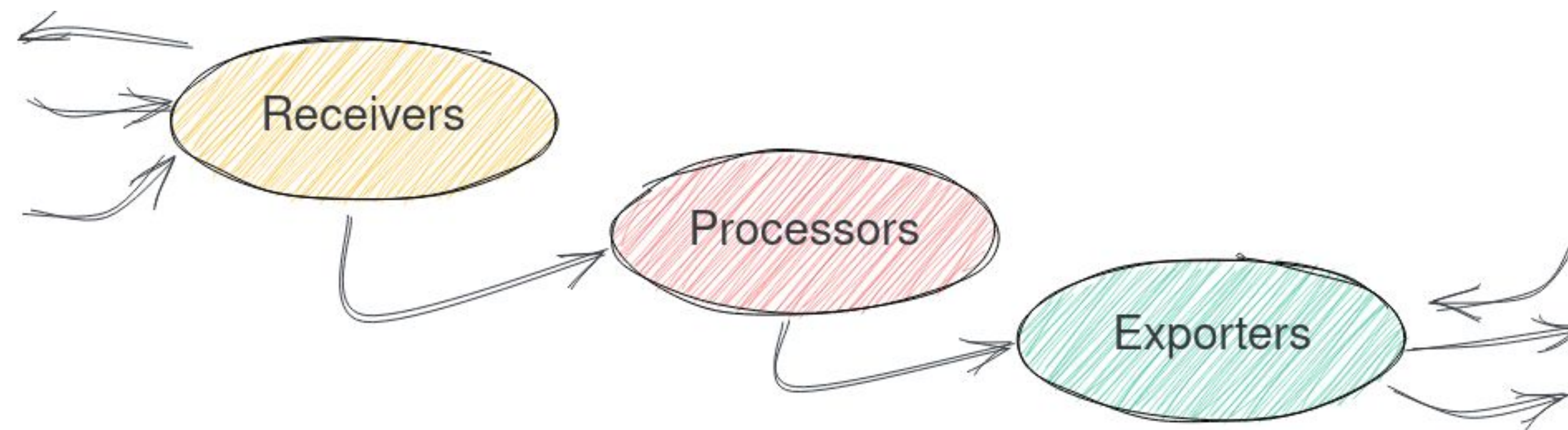
Instrumentation - Demo

Making your project more observable



Backend - OpenTelemetry Collector

Pipeline for telemetry data





OpenTelemetry Collector

Receivers - Core

fluentforwardreceiver
hostmetricsreceiver
jaegerreceiver
kafkareceiver

opencensusreceiver
otlpreceiver
prometheusreceiver
zipkinreceiver



OpenTelemetry Collector

Receivers - Contrib

awsecscontainermetricsreceiver
awsxrayreceiver
carbonreceiver
collectdreceiver
dockerstatsreceiver
dotnetdiagnosticsreceiver
filelogreceiver
jmxreceiver
k8sclusterreceiver
kafkametricsreceiver
kubeletstatsreceiver
memcachedreceiver

nginxreceiver
prometheusexecreceiver
receivercreator
redisreceiver
sapmreceiver
signalfxreceiver
simpleprometheusreceiver
splunkhecreceiver
statsdreceiver
syslogreceiver
wavefrontreceiver
windowsperfcountersreceiver
zookeeperreceiver



OpenTelemetry Collector

Exporters - Core

fileexporter
jaegerexporter
kafkaexporter
loggingexporter
opencensusexporter

otlpexporter
otlphttpexporter
prometheusexporter
prometheusremotewriteexporter
zipkinexporter



OpenTelemetry Collector

Exporters - Contrib

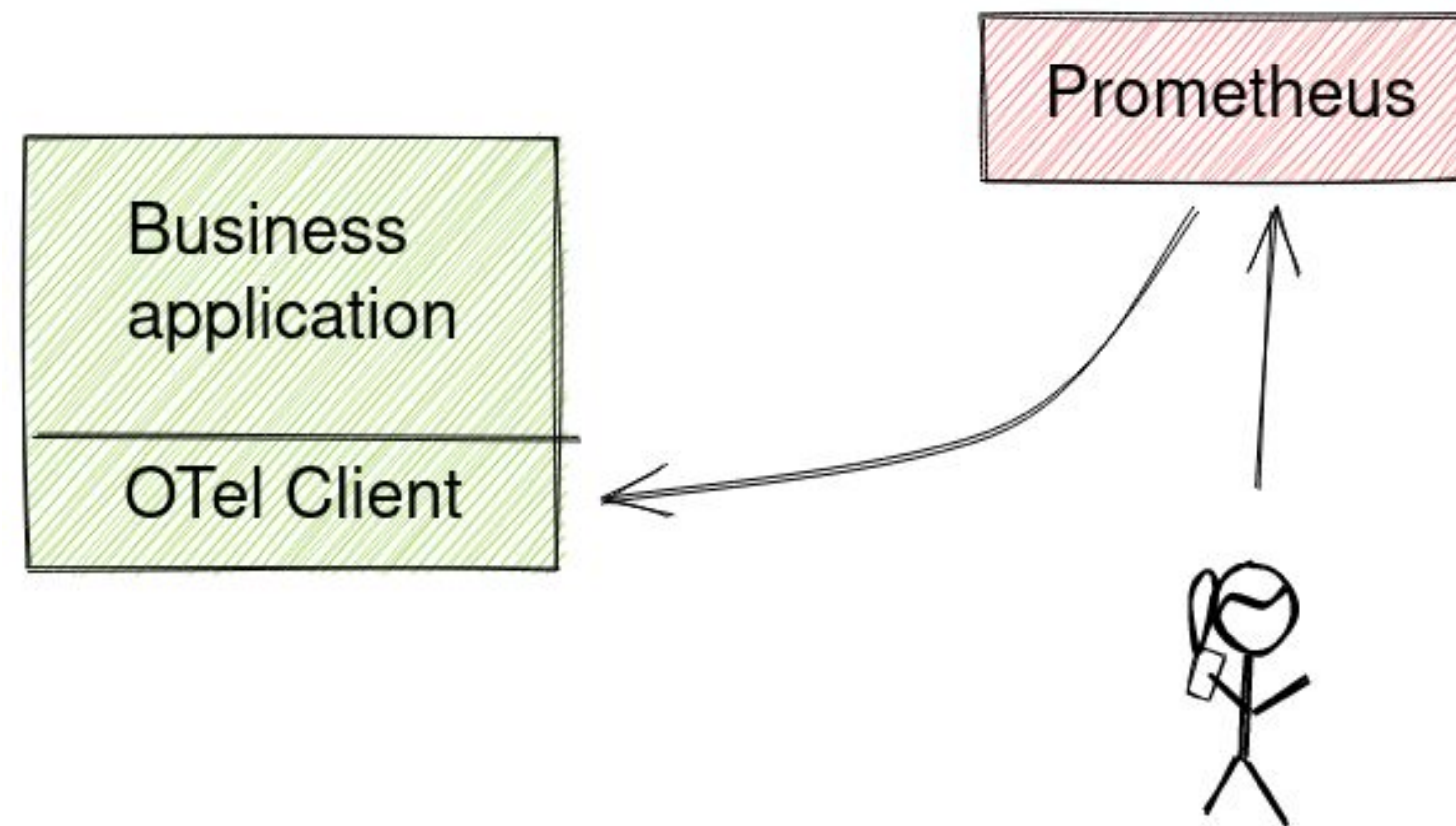
alibabacloudlogserviceexporter
awsemfexporter
awskinesisexporter
awsprometheusremotewriteexporter
awsxrayexporter
azuremonitorexporter
carbonexporter
datadogexporter
dynatraceexporter
elasticexporter
elasticsearchexporter
f5cloudexporter
googlecloudexporter

honeycombexporter
jaegerthrifthttpexporter
loadbalancingexporter
logzioexporter
lokiexporter
newrelicexporter
sapmexporter
sentryexporter
signalfxexporter
splunkhecexporter
stackdriverexporter
sumologicexporter
uptraceexporter



Reference architecture

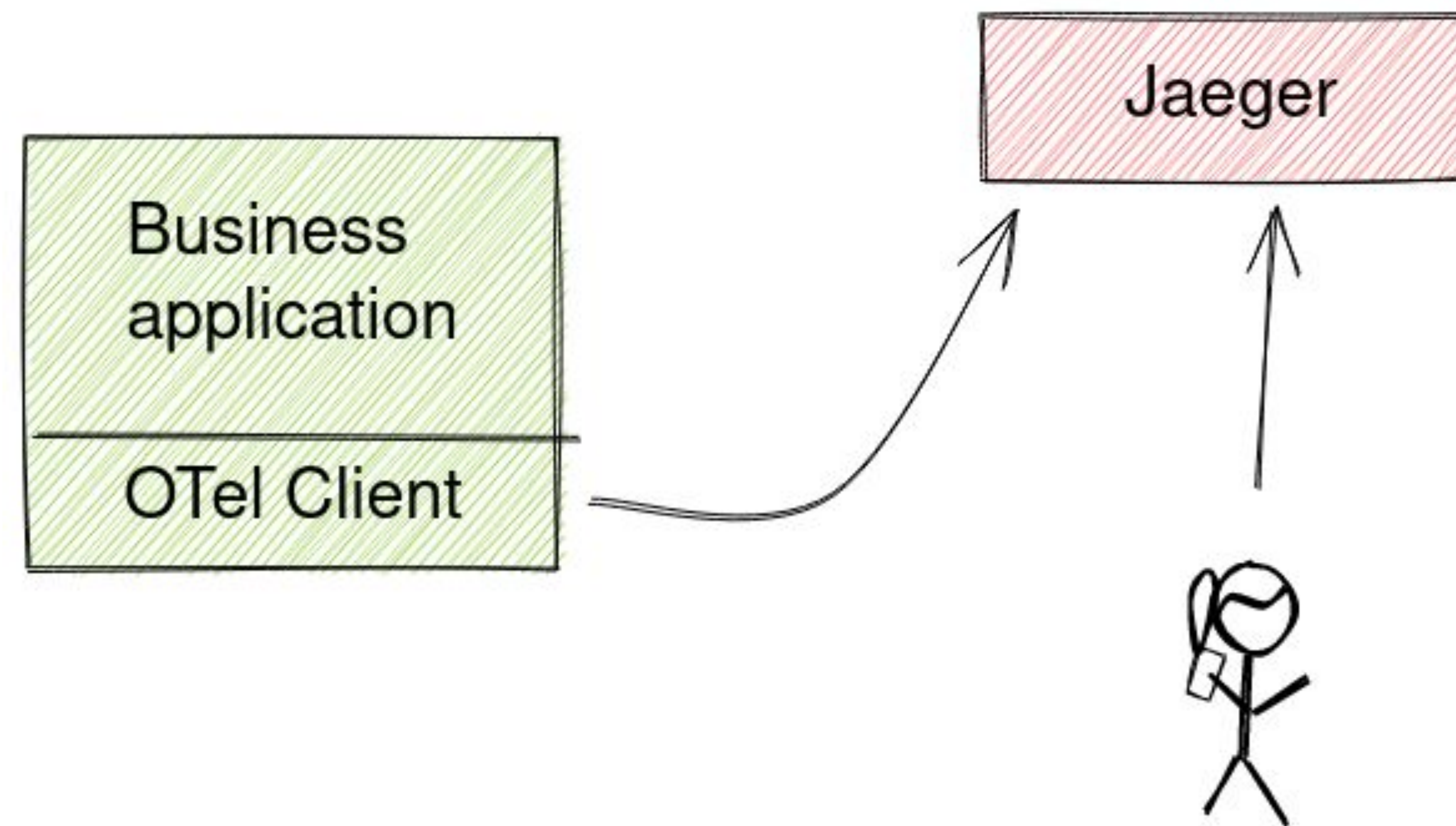
No collector





Reference architecture

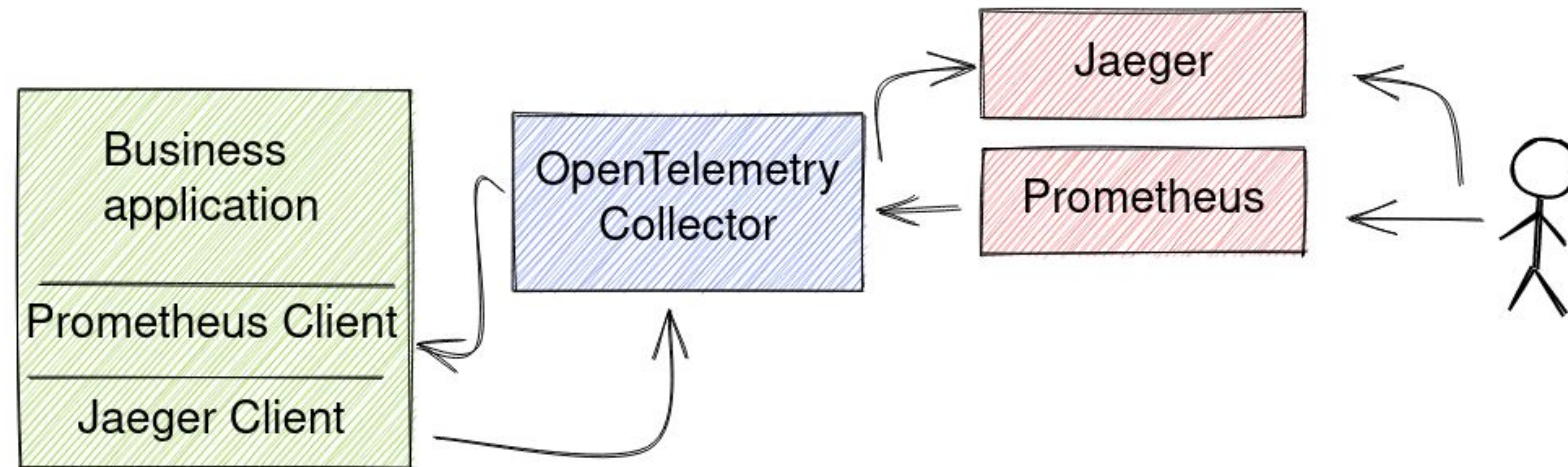
No collector





Reference architecture

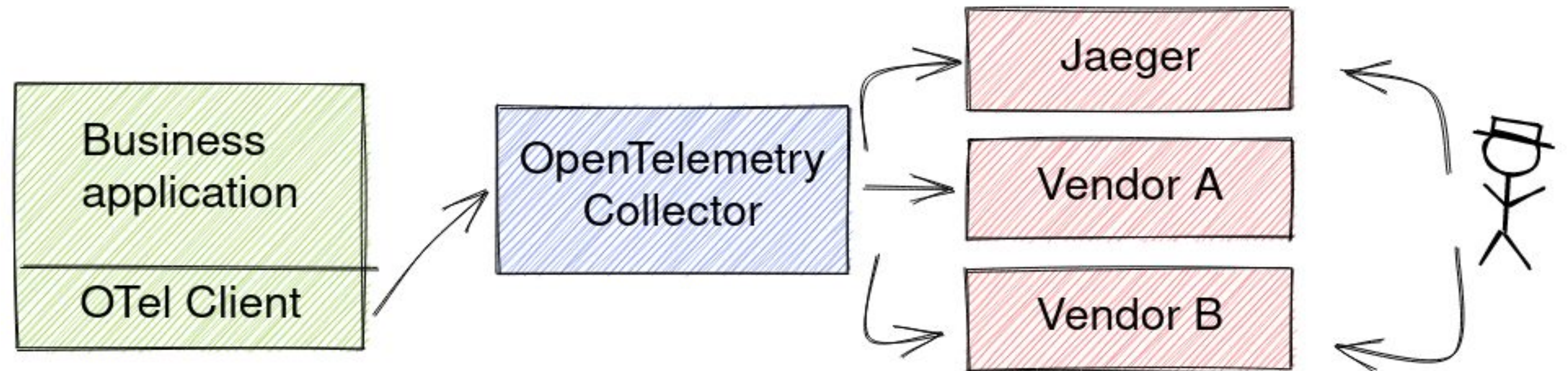
Collector as equalizer





Reference architecture

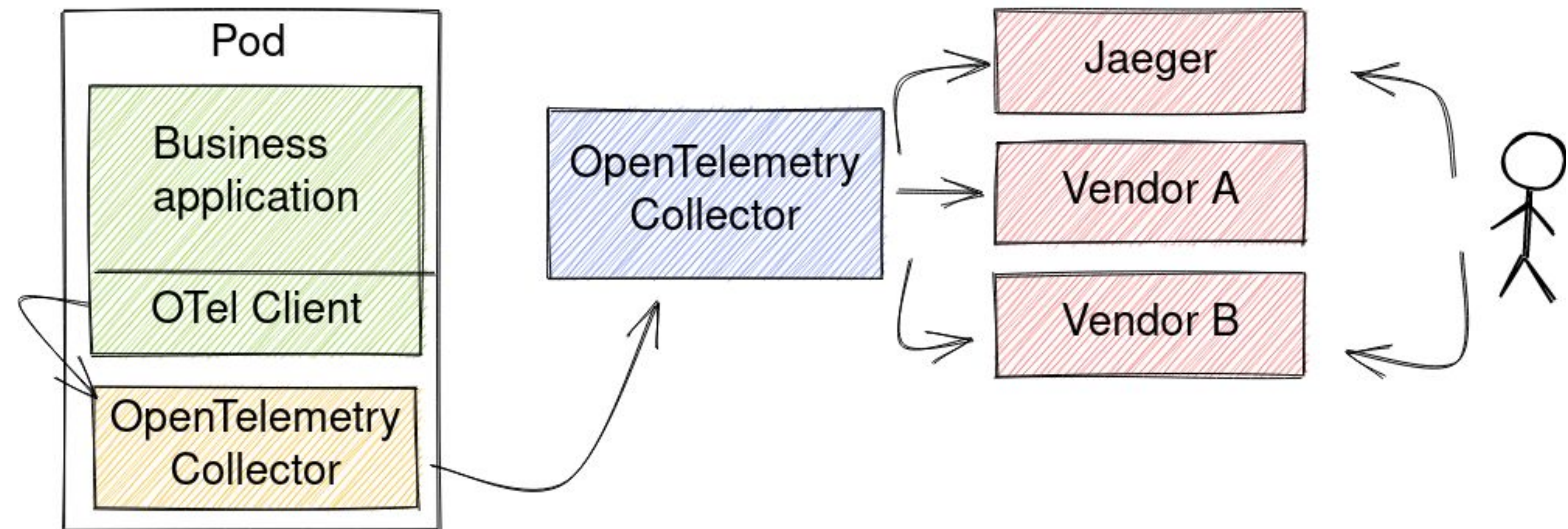
Collector as proxy





Reference architecture

Sidecar





The future

Distributed tracing parts are GA, metrics to follow, logs later

Some SDKs are GA already, others to follow

Kubernetes Operator in development

Jaeger Agent/Collector to be based on OpenTelemetry Collector

Jaeger clients will eventually be deprecated



Key takeaways

- OpenTelemetry is a collection of smaller, independent projects
- OTLP is the data (format, wire) specification
- As a developer, you probably care most about the instrumentation
- As an SRE, you care most about instrumentation and collector
- As a vendor, you care most about OTLP
- Tracing is the most mature part, metrics coming next, logs after that



Q&A



Thank you

Recommended resources:

[OpenTelemetry](#)

Photos from Pixabay and Pexels:

[Agenda](#), [Standards](#), [Instrumentation](#), [Collector](#), [Future](#), [Key takeaways](#)