



Metrics e Tracing com OpenTelemetry



25 de Março Date, 2021 - TDC



Hello!

Francisco Ribeiro

Lead Solutions Engineer

francisco@wso2.com

<https://www.linkedin.com/in/fribeiro/>

<http://www.xicojunior.com>

Observability



Se refere à capacidade de entender o desempenho de uma aplicação com base em dados de saída, ou telemetria.



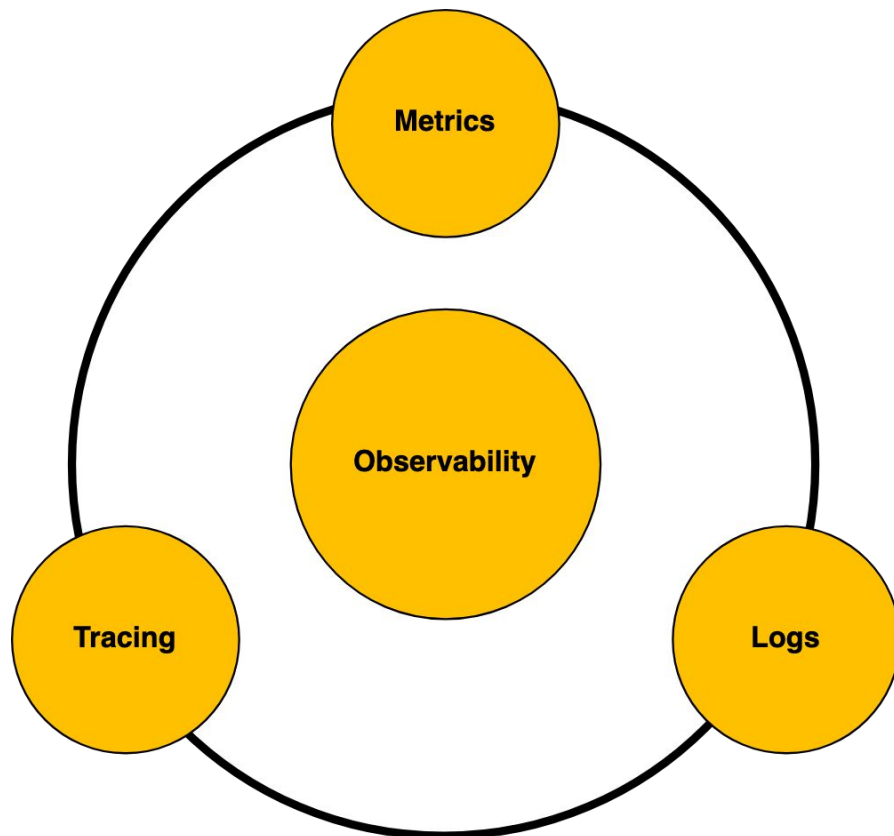


<https://br.pinterest.com/pin/518054763360854994/>



<https://www.sportskeeda.com/f1/formula-1-technical-importance-of-a-f1-pit-wall>

Observability



- Métricas
- Logs
- Tracing Distribuído



OpenTelemetry

OpenTelemetry(OTEL)

Um framework de Observability para cloud-native software.

OpenTelemetry é uma coleção de ferramentas, APIs e SDKs. Estas são usadas para instrumentar, gerar, coletar e exportar dados de telemetria(métricas, logs e tracing) para análise para entender o comportamento e performance das aplicações.



CLOUD NATIVE
COMPUTING FOUNDATION

<https://opentelemetry.io/>

**The ultimate goal for OpenTelemetry is to ensure
that this telemetry
data is a built-in feature of cloud-native
software.**

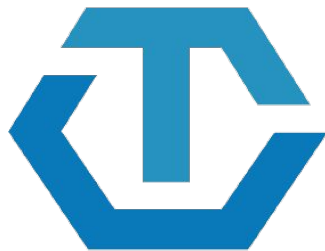


OpenTelemetry



OpenCensus

+

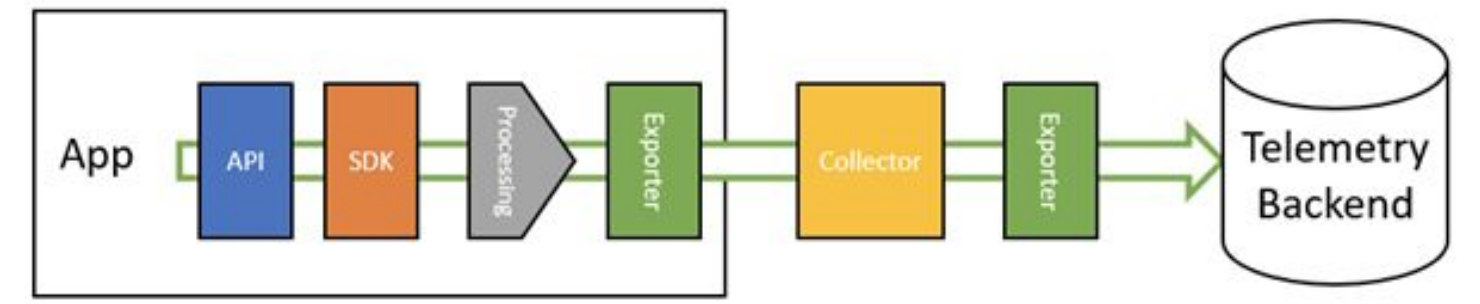


OpenTracing

=



Componentes



<https://laptrinhx.com/what-is-opentelemetry-everything-you-wanted-to-know-3895678607/>

Linguagens Suportadas e Exporters

Exporters

- Jaeger
- Prometheus
- Kafka
- OTLP
- Zipkin
- LightStep
- Dynatrace
- Honeycomb
- ...

Linguagens Suportadas

- Java
- Python
- C#
- Go
- Javascript
- Rust
- Erlang/Elixir



OpenTelemetry Tracing

```
private static final Tracer tracer =  
    GlobalOpenTelemetry.getTracerProvider().get("com.xicojunior.otel.servle  
t.OtelServlet");  
  
Span childSpan = tracer.spanBuilder("OtelServlet.doGet").startSpan();  
try(Scope scope = childSpan.makeCurrent()) {  
    Span span = Span.current();  
    span.setAttribute("http.route", "otel");  
    span.setAttribute("userId", userId);  
    span.setAttribute("http.method", "GET");  
} finally {  
    childSpan.end();  
}
```



OpenTelemetry - Metrics

```
private static Meter meter = GlobalMetricsProvider.get().get("backend");
```

```
//Create the Counter
```

```
LongUpDownCounter requestCounter =  
    meter.longUpDownCounterBuilder("request_counter")  
        .build();
```

```
//Increase the counter
```

```
requestCounter.add(1);
```



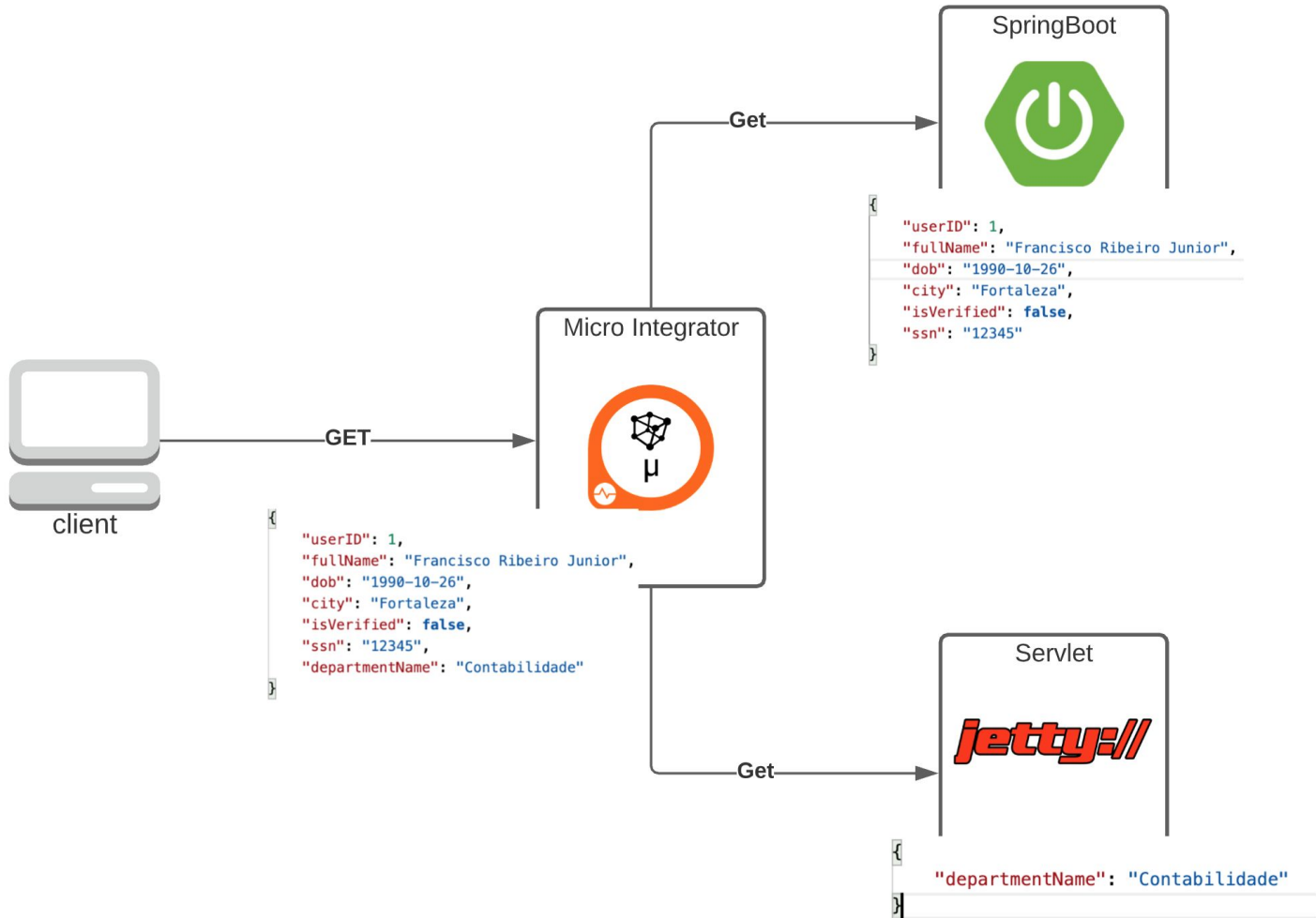
OpenTelemetry Versão Atual

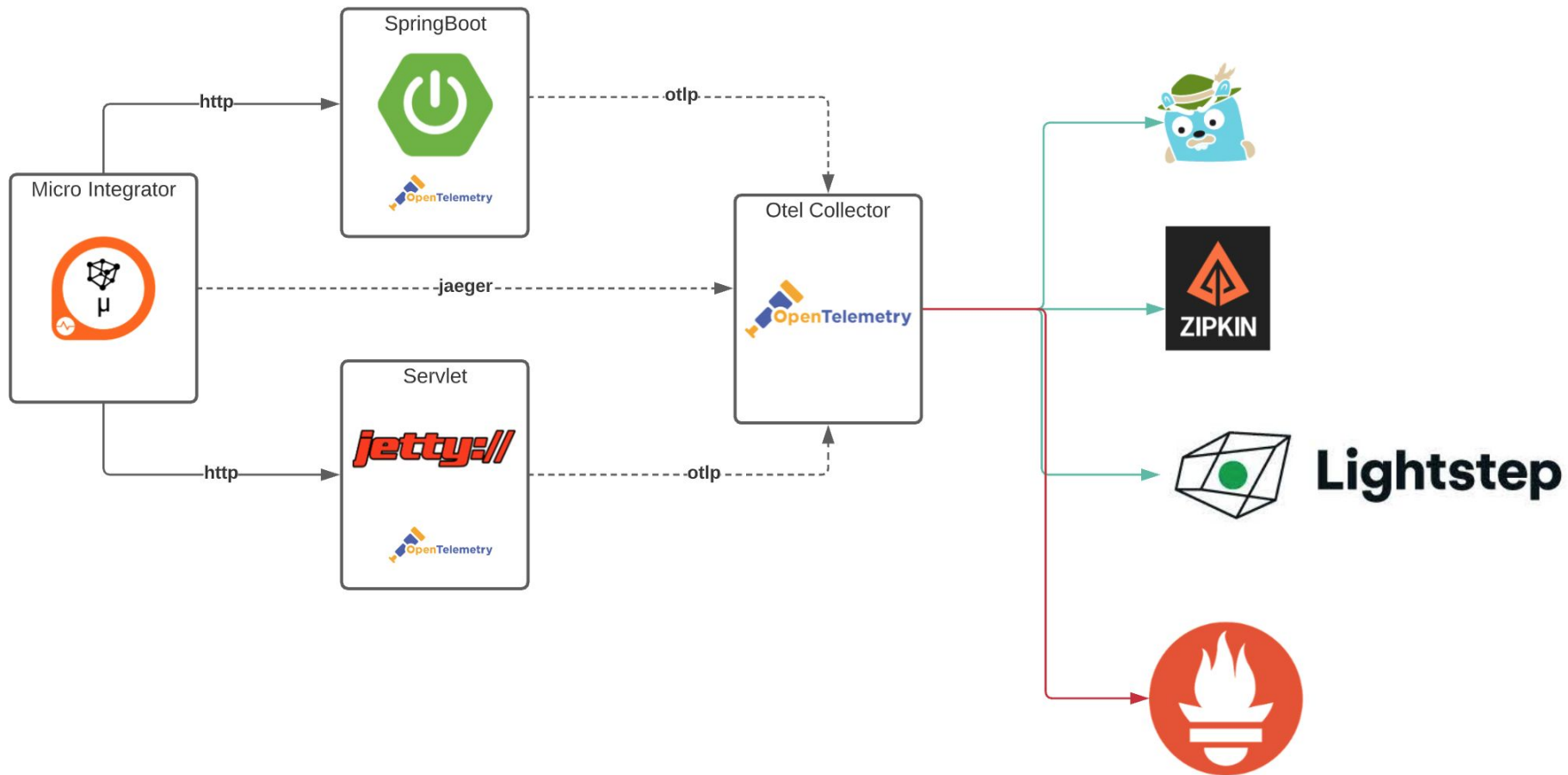
- Especificação na v1.1.0 - Tracing API Stable
- Métricas está em Alpha - Roadmap 30/11 Metrics API estável



Demo







OpenTelemetry Agents

```
java -javaagent:/opentelemetry-javaagent-all.jar \  
    -Dotel.resource.attributes=service.name=otel-servlet \  
    -Dcustom-span=false \  
    -Dotel.propagators=jaeger \  
    -jar target/otel-servlet-jar-with-dependencies.jar
```

Question Time!



Thanks!



wso2.com



in

