

# Analisando o comportamento de aplicações:



Os 3! pilares usados pelas maiores empresas do mundo para fazer observabilidade

George Tavares



## Golden Signals

- Rate
- Errors
- Duration
- Saturation

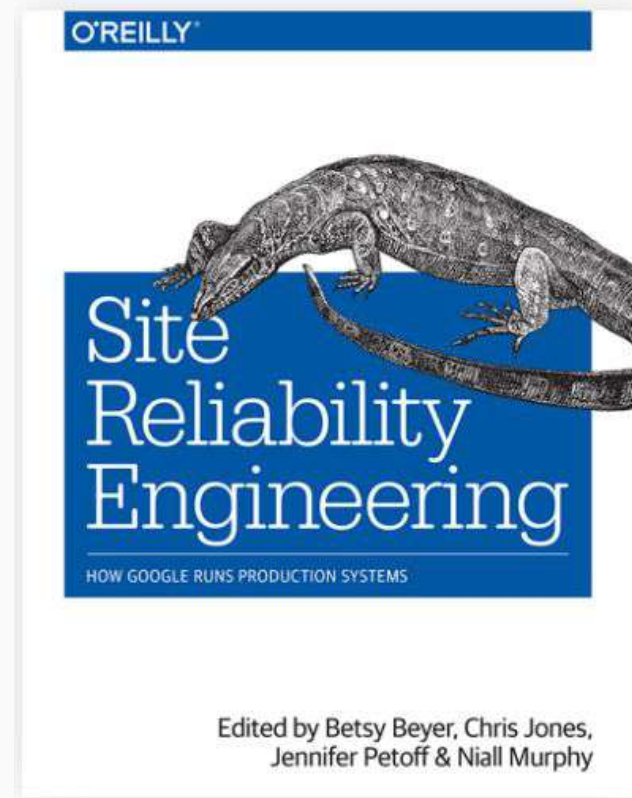
- SLI
- SLO
- SLA

## Metodologia USE - Recursos

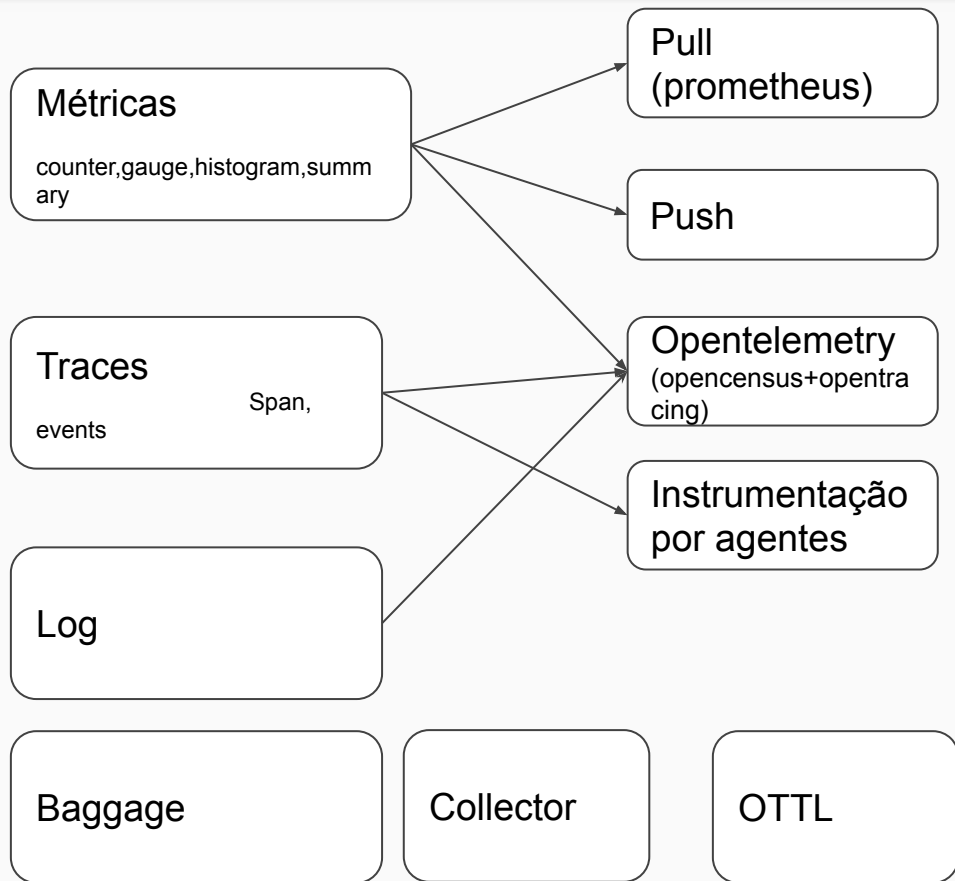
- Utilization
- Saturation
- Error

## Metodologia RED - Serviços

- Rate
- Error
- Duration



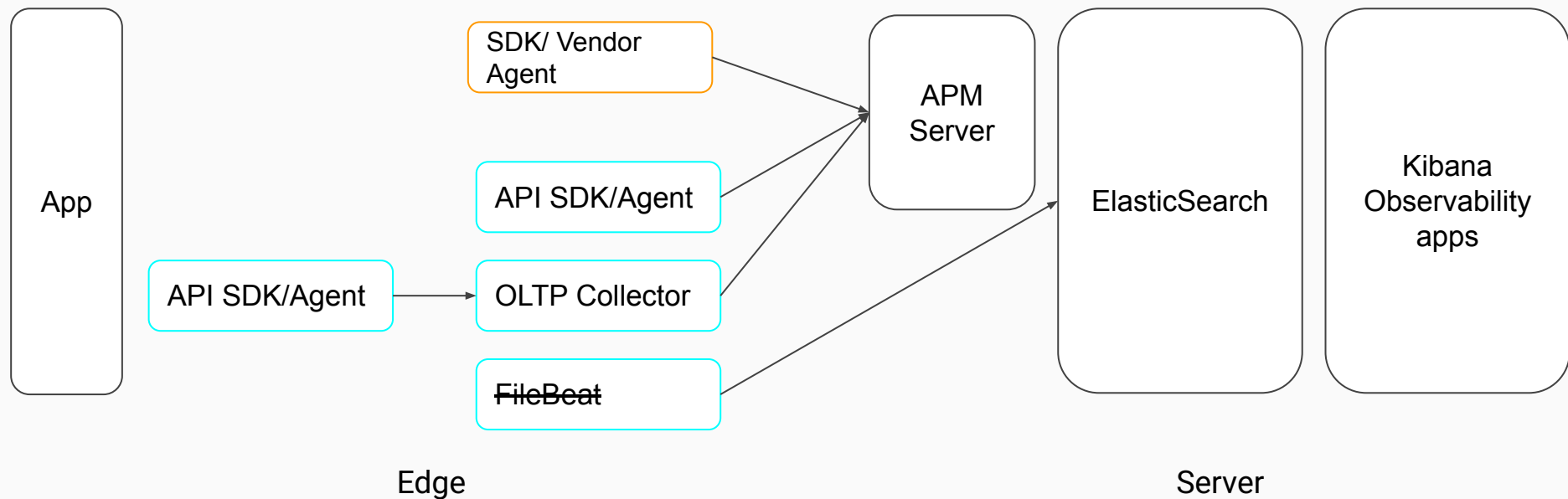
# Observabilidade Opentelemetry



Alertas??

Profiling

# Opentelemetry Demo Setup



# Semantic Convention

- Define nomes comuns aos nomes dos atributos, métricas propriedades
- Cada provider tinha seu próprio nome
- Prometheus teve que mudar bastante para suportar isso

## HTTP server #

Metric: `http.server.request.duration`

This metric is required.

When this metric is reported alongside an HTTP server span, the metric value SHOULD be the same as the HTTP server span duration.

This metric SHOULD be specified with [ExplicitBucketBoundaries](#) of [ 0.005, 0.01, 0.025, 0.025, 0.5, 0.75, 1, 2.5, 5, 7.5, 10 ] .

## Naming conventions

Meter names should use dots to separate segments, `a.name.like.this`. Micrometer applies naming conventions to convert registered meter names to match the expectations of backend monitoring systems.

Given the following declaration of a timer: `registry.timer("http.server.requests")`, applied naming conventions will emit the following metrics for different monitoring systems:

- Prometheus: `http_server_requests_duration_seconds`
- Atlas: `httpServerRequests`
- Graphite: `http.server.requests`
- InfluxDB: `http_server_requests`

\* Services:

\*\* code1

Quarkus, opentelemetry, metric otlp exporter

- hello: call wait

- error: call wait, throw exception

- remote: call wait, remote call code2 -> hello

\*\* code2

Quarkus, opentelemetry, metric otlp exporter, jdbc addon

- hello: call wait ( request to postgres pg\_sleep)

```
1 create user postgres with password 'postgres';
2
3 create role postgres with superuser password 'postgres'
4   inherits (postgres) noinherit noencryption noreplicate noadmin noall;
5
6 create database postgres with owner postgres;
7 create database otel with owner postgres;
8
9 create database postgres template postgres;
10 create database otel template postgres;
11
12
13
14
15
16
17
18
19
20
21
```

```
2025-06-20 10:01:18.438 UTC [1]: database system is ready to accept connections
2025-06-20 10:01:18.438 UTC [2]: server started by postgres (pid 120). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [3]: server started by postgres (pid 121). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [4]: server started by postgres (pid 122). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [5]: server started by postgres (pid 123). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [6]: server started by postgres (pid 124). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [7]: server started by postgres (pid 125). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [8]: server started by postgres (pid 126). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [9]: server started by postgres (pid 127). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [10]: server started by postgres (pid 128). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [11]: server started by postgres (pid 129). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [12]: server started by postgres (pid 130). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [13]: server started by postgres (pid 131). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [14]: server started by postgres (pid 132). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [15]: server started by postgres (pid 133). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [16]: server started by postgres (pid 134). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [17]: server started by postgres (pid 135). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [18]: server started by postgres (pid 136). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [19]: server started by postgres (pid 137). Starting up user bootstrap... [1]
2025-06-20 10:01:18.438 UTC [20]: server started by postgres (pid 138). Starting up user bootstrap... [1]
```

Iniziando postgres e otel

- Summary
- Network
- File
- Log
- Content
- Audio
- Video
- JavaScript
- Performance
- Timeline
- Debugger
- Console
- Profiler
- Web Inspector
- Accessibility
- Device
- Remote Explorer
- Extensions

GCP traces



Performance



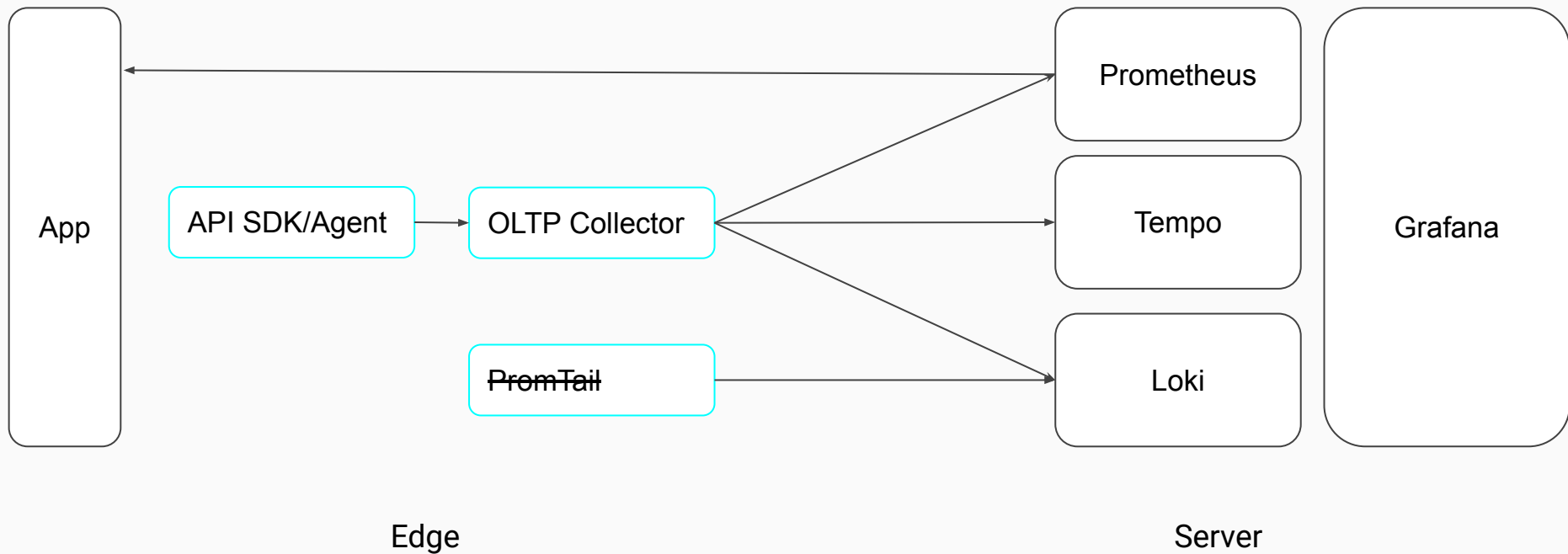
Endpoints/traces

Labels: [Endpoints](#) [Traces](#)

Legend: [Endpoints](#) [Traces](#)



# Grafana Tempo Loki Prometheus Demo Setup



- Home
- Dashboard
- Tools
- Services
- Hosts
- Ports
- Network
- Security
- Monitoring
- Alerts
- Configuration
- Settings
- Help
- Logout

# Logs no dashboard



Log Entries

Time	Message	Source
10:00:00	INFO: [root@kali:~]#	root@kali:~
10:00:01	INFO: [root@kali:~]#	root@kali:~
10:00:02	INFO: [root@kali:~]#	root@kali:~
10:00:03	INFO: [root@kali:~]#	root@kali:~
10:00:04	INFO: [root@kali:~]#	root@kali:~
10:00:05	INFO: [root@kali:~]#	root@kali:~
10:00:06	INFO: [root@kali:~]#	root@kali:~
10:00:07	INFO: [root@kali:~]#	root@kali:~
10:00:08	INFO: [root@kali:~]#	root@kali:~
10:00:09	INFO: [root@kali:~]#	root@kali:~
10:00:10	INFO: [root@kali:~]#	root@kali:~



## Continuos Profiling

- Grafana comprou o Pyroscope
- Elastic comprou Optimyze e está doando para o opentelemetry
- Permite identificar os hot spots de gargalo de performance
- Não é uma visão de timeline, e sim uma visão de agregados de tempo de execução
- eBPF



Trace distribuído

- Desenvolvedor na Noson
- Entusiasta de tecnologia, adoro estudar infraestrutura e devops
- Tiro certificações nas horas vagas: 4xAWS, 4xAZURE, CKA, CKS, ~~Elastic Engineering, Elastic Observability~~ ...
- Vencedor do 1o K8S ULC da Linuxtips



<https://github.com/tanquetav/lab-observabilidade>

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



Perguntas ?