

**DOSE DE TELEMETRIA**

# **O FANTÁSTICO MUNDO DA OBSERVABILIDADE**



# PRAZER

## JURACI PAIXÃO KRÖHLING

- Webmaster, 2000
- Engenheiro de software
- Focado em sistemas de telemetria desde 2016
- Ex-mantenedor do Jaeger e OpenTracing
- Mantenedor de módulos do OpenTelemetry Collector
- Criador do Jaeger Operator, OpenTelemetry Operator e OpenTelemetry Collector Builder
- Membro do comitê de governança do OpenTelemetry
- Palestrante
  - KubeCon EU e NA, Devoxx, JavaLand, OpenSource Summit EU e NA, FOSDEM, ...
- Membro do comitê de programação de algumas KubeCon's, como EU e NA 2023
- Criador do Dose de Telemetria





# PRAZER

# JURACI PAIXÃO KRÖHLING

- Pai



# RESUMO DE TÓPICOS

## O QUE VAMOS DISCUTIR

---

- Movimento Agile
- Microserviços
- DevOps
- Engenharia de confiabilidade
- Telemetria vs. Monitoramento vs. Observabilidade
- Os sinais da observabilidade
- O que esperar do futuro



GRATION



AGILE

N

SPR

RETROSPECTIVE

RATION

DEVELO

TEAMWORK

# MOVIMENTO AGILE

## MELHORANDO O TIME TO MARKET

- Demorar meses pra entregar uma nova funcionalidade não faz mais sentido
- O lema agora é: pequenas iterações, rápidas evoluções
- Terminou, manda pra produção!
- Não deu certo, arruma (ou reverte)
- Equipes do tamanho de duas pizzas
- Cada equipe focada em um aspecto do negócio





# MICROSERVIÇOS

## RESULTADO DO AGILE

- Cada serviço resolve um problema de negócio
- Cada serviço tem sua equipe
- Equipes gerenciam mais de um serviço ao mesmo tempo
- Computação distribuída ao extremo!





# DEVOPS

## TRAZENDO AGILIDADE DE ENTREGA

- Desenvolvimento ágil
- Integração e entrega contínuas
- Rápido feedback
- Evolução e mudança são constantes
- Exige salvaguardas





# ENGENHARIA DE CONFIABILIDADE SALVAGUARDAS!

- Entrega as salvaguardas exigidas por DevOps
- Resiliência
- Automação
- Expectativas (SLOs), promessas (SLAs)
- Telemetria
  - SLIs
  - Monitoramento
  - Observabilidade
- Plano de lançamento



## TELEMETRIA:

Dados gerados pela nossa  
aplicação, informando o estado  
em tempo de execução.



## MONITORAMENTO:

**Utilização de dados de telemetria  
pra responder rapidamente  
perguntas frequentes.**



## OBSERVABILIDADE:

**Capacidade de se utilizar dados  
de telemetria pra responder  
perguntas que ainda não temos.**



# TIPOS DE DADOS DE TELEMETRIA

## TRÊS PILARES

### LOGS

Eventos relacionados ao ciclo de vida do serviço.

### MÉTRICAS

Valores numéricos criados com base em eventos do serviço.

### RASTROS

Eventos relacionados a uma transação de negócio, potencialmente afetando múltiplos serviços.





# TIPOS DE DADOS DE TELEMETRIA MAIS UTILIZADOS

## LOGS

Eventos relacionados ao ciclo de vida do serviço.

## MÉTRICAS

Valores numéricos criados com base em eventos do serviço.

## RASTROS

Eventos relacionados a uma transação de negócio, potencialmente afetando múltiplos serviços.



# TIPOS DE DADOS DE TELEMETRIA COM POTENCIAL FUTURO

## ERROS

Stacktraces de erros que acontecem em um serviço.

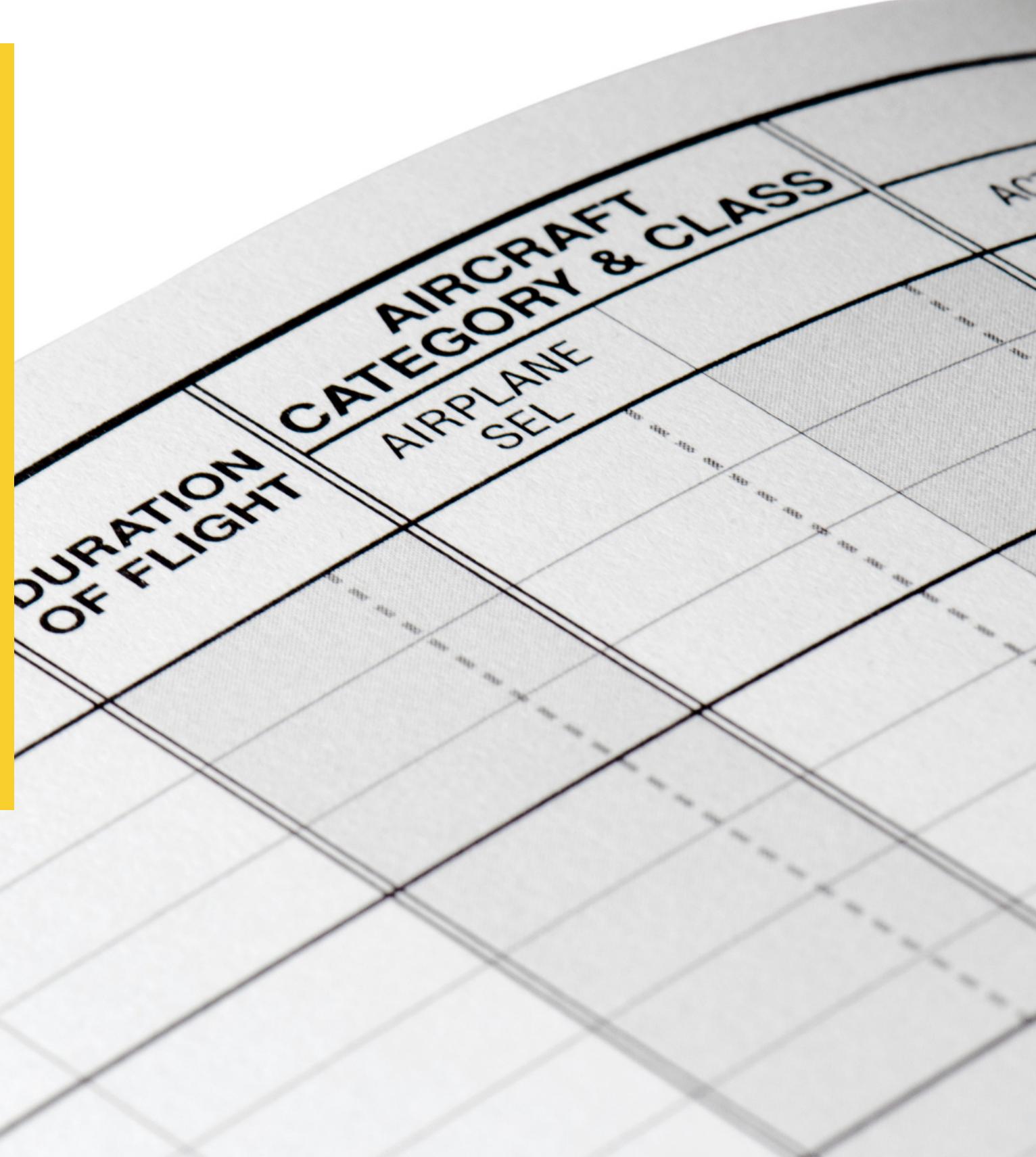
## PERFIS

Estado da memória e CPU em momentos específicos.

## EVENTOS

Algo que tenha acontecido fora do serviço, mas que pode ter um efeito nele.





# LOGS

## DIÁRIO DE BORDO

Registro de eventos importantes para a aplicação, onde cada evento geralmente contém um horário e a descrição.



# Logs como eu aprendi

## TEXTO PURO

O evento é representado como uma linha de texto sem estrutura definida, mas talvez seguindo um padrão.

## ARMAZENAMENTO LOCAL

Os eventos são escritos pela aplicação utilizando uma biblioteca de instrumentação, que geralmente escrevia direto para um arquivo de texto em disco local.

## SEM COLETA CENTRAL

Antes de termos microsserviços, logs não eram enviados para lugares centrais: quando precisávamos, entrávamos na máquina e fazíamos análise lá mesmo. No máximo, copiávamos os arquivos de logs para a nossa máquina de desenvolvimento.

## BUSCA VIA GREP

Não existiam ferramentas de análise ou de busca de eventos específicos.



# Logs - estado da arte

## MENSAGEM ESTRUTURADA

O evento é representado como uma mensagem estruturada, com campos e atributos fáceis de serem trabalhados e analisados.

## ARMAZENAMENTO?

Os eventos são escritos pela aplicação utilizando as mesmas bibliotecas de instrumentação de antes, mas se espera que a aplicação escreva os eventos para stdout e stderr. Escrever em arquivo local não é mais essencial (ou recomendado).

## COLETA CENTRAL

Ferramentas são utilizadas para armazenas logs de forma centralizada, possibilitando uma melhor governança do que é armazenado. Ferramentas mais modernas têm ainda bancos de dados feitos especificamente para armazenamento de logs.

## ANÁLISE PROFISSIONAL

Análise dos logs é feita com ferramentas específicas para o propósito.



# Logs - diferentes audiências!

## DEBUG

Pessoa desenvolvedora que está tentando entender o comportamento, provavelmente durante o desenvolvimento da aplicação.

## INFO

Usuário rodando a aplicação. A mensagem tem que vir partindo do pressuposto de que a pessoa não sabe o que está acontecendo.

## WARN, ERROR, PANIC

Alguém às 02:00 tentando achar o motivo de um problema. Deve incluir informações sobre o que está de errado e possíveis causas ou soluções.



---

## LOGS:

**Se seu evento pode ser analizado  
de forma isolada, é um log.**

---



# EXEMPLOS DE LOGS BOAS PRÁTICAS

## CICLO DE VIDA

Mensagem na hora da invocação do serviço com logs contendo versões, configurações chave, ...

## EVENTOS CHAVE

Servidor HTTP inicializado, conexão com banco de dados caiu ou voltou, nova topologia de rede detectada,..

## ALGO INESPERADO

Talvez um erro crítico que não faça parte de uma requisição, um estado inesperado de uma dependência, ...



# ISSO NÃO É LOG ANTI-PATTERNS

## TRANSAÇÕES

Coisas que acontecem no escopo de uma requisição HTTP.

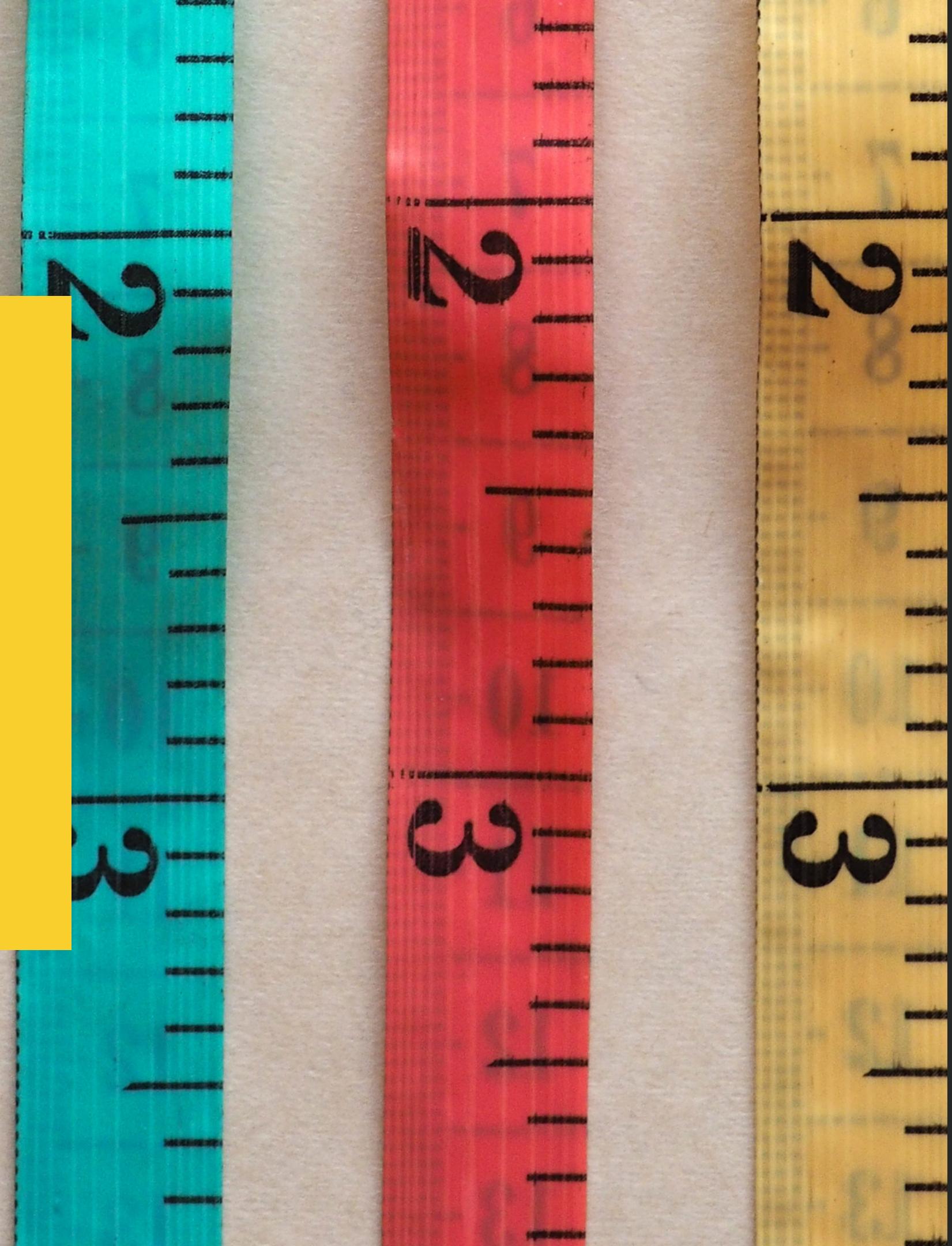
## EVENTO ESPERADO

Eventos que não são chave nem esperados e que acontecem com frequência, como atualização ou invalidação de cache.

## ESTADO INTERNO

Estado da memória, do tamanho das filas de processamento, latência de uma operação.





# MÉTRICAS

## NÚMEROS POR TRÁS DO SERVIÇO

Registro da representação numérica de um evento,  
segregando por dimensões arbitrárias.



# Tipos de métricas

## CONTADORES

Contador de eventos, podendo ser monotônico (somente sobe), ou sobe-desce. Por exemplo, número de visualizações de um vídeo (monotônico), ou número de pessoas visualizando o vídeo neste momento (sobe-desce).

## TEMPORIZADORES

Medição de tempo entre dois eventos. Por exemplo, medição de quanto tempo levou para um serviço retornar uma chamada ao cliente, ou quanto tempo um algoritmo levou para finalizar uma execução.

## VALORES FIXOS

Armazenam um valor fixo arbitrário. Por exemplo, a temperatura atual no local, quanto da memória está sendo utilizada no momento, ou o tamanho de uma fila de execução.



## MÉTRICAS:

**Se seu evento pode ser resumido  
a um número, ou analisado de  
forma agregada, use métricas.**



# EXEMPLOS DE MÉTRICAS BOAS PRÁTICAS

## POUCAS DIMENSÕES

Os melhores bancos de dados de métricas da atualidade sofrem quando a cardinalidade é alta.

## MEÇA LATÊNCIAS

Meça quanto tempo as operações chave estão levando, incluindo as chamadas a serviços dependentes.

## CONTADORES

Ao invés de criar um evento de log toda vez que algo específico acontecer, incremente um contador.



# SINAIS DOURADOS

## OS QUATRO SINAIS DOURADOS

### LATÊNCIA

Quanto tempo determinada operação levou.

### TRÁFEGO

Requisições por segundo que a aplicações está tendo.

### ERROS

Taxa de erros retornados pela aplicação.

### SATURAÇÃO

Percentual mostrando o quanto do seu serviço ainda está livre (utilização em relação à capacidade).



# O QUE MEDIR

## RED, ESPECIAL PARA APLICAÇÕES

### REQUISIÇÕES

O número de requisições do seu serviço. Pense em colocar o status como uma dimensão da métrica.

### ERROS

Quantas das requisições foram erros? Armazene em uma métrica separada, para ficar mais fácil de calcular percentuais de sucesso.

### DURAÇÃO

Meça quanto tempo seu serviço demorou para responder para seu usuário.



# O QUE MEDIR USE, ESPECIAL PARA INFRA

## UTILIZAÇÃO

Quanto do seu recurso foi utilizado.

## SATURAÇÃO

Percentual mostrando o quanto do seu serviço ainda está livre (utilização em relação à capacidade).

## ERROS

Taxa de erros retornada ao seu usuário.





# RASTROS

## OS PASSOS DE UMA TRANSAÇÃO

Registro de todos os passos tomados durante a execução de uma transação de negócio, potencialmente passando por diversos serviços em diferentes etapas da transação.



# Rastreamento distribuído

## PROPAGAÇÃO DE CONTEXTO

A cada chamada, seja local ou remota, o contexto é propagado. Contexto é composto do ID do rastro, ID do trecho de origem, e mais alguns dados extras.

## SUPER LOGS

Trechos são eventos, semelhantes à logs. O que os diferencia é que trechos tem uma duração e informações de causalidade (ID do rastro, ID do trecho de origem). Tudo além disso é "extra".

## RASTROS NÃO EXISTEM

Um rastro é uma coleção de trechos (spans). Na maioria das vezes, não existe uma estrutura de dados que represente o rastro, apenas trechos individuais que compartilham o mesmo ID do rastro.

## BAGAGENS, AMOSTRAGEM, ...

Existem outros conceitos e funcionalidades dentro da disciplina de rastreamento distribuído, mas que não são relevantes neste workshop.



---

## RASTROS:

**Se seu evento pertence a uma  
transação de negócio, ele é um  
trecho (span) em um rastro  
(trace).**



# EXEMPLOS DE TRECHOS BOAS PRÁTICAS

## FRONTEIRAS

Crie trechos que representem as bordas da sua aplicação: serviços HTTP, ou conexões ao banco de dados são bons candidatos.

## MEÇA O DIFERENTE

Crie trechos que representem os desvios do comportamento padrão da sua aplicação.

## MEÇA O IMPORTANTE

Crie trechos ao redor de algoritmos críticos, problemáticos ou importantes para sua aplicação.



# EXEMPLOS DE TRECHOS ANTI-PATTERNS

## TODOS OS MÉTODOS

Resista a tentação de criar trechos para todos os métodos da sua aplicação. Menos é mais.

## TRECHOS REPETIDOS

Se tiver a mesma operação acontecendo diversas vezes dentro de um trecho, faça um trecho e armazene o número de execuções como atributo.

## RASTROS GIGANTES

Rastros representam transações de negócio, mas às vezes faz sentido quebrar um rastro em vários, com "span links" entre eles.



# Obrigado pela atenção!

---



Dose de  
Telemetria

