

Observabilidade de Sistemas de Software

...

\$ whoami

Engenharia de Computadores e Telemática - UA - 2008

Desenvolver serviços de backend desde 2008

- Telco
- E-Commerce
- Turismo
- Empresas grandes e pequenas

SRE desde 2018

- Operações de sistemas de software
- Observabilidade, Incidentes, Performance, Escala

3

20

36

Qual é o propósito desta apresentação?

Dar uma introdução a uma **prática** fundamental na **operação, manutenção, e evolução** de sistemas de software.

Isto **não** é uma apresentação de uma carreira alternativa em Engenharia de Software.

Agenda

Porque precisamos de monitorização?

Isto consegue ser complexo, e caro. Que valor é que traz?

O que devemos monitorizar?

100 fontes vão sugerir 1.000 coisas diferentes para monitorizar. Por onde começar?

Como obter os sinais de monitorização?

Os sinais não surgem do nada. Vamos ter de os gerar a partir dos nossos sistemas.

Como interpretar esses sinais?

É maioritariamente números. Não é só ver qual é o maior, ou menor?

Que uso dar aos sinais de monitorização?

Agora que temos todos estes dados, que é que fazemos com eles?

Definições

Monitorização

Coletar, processar, agregar e exibir dados quantitativos em tempo real sobre um sistema, como contagens e tipos de pedidos, contagens e tipos de erros, tempos de processamento e tempos de vida do servidor.

Alertas

Uma notificação destinada a ser lida por um humano e enviada para um sistema, como um tracker de bugs ou tickets, um e-mail, ou um pager.

- *Site Reliability Engineering, cap 6, Monitoring Distributed Systems*

Definições

Telemetria

Telemetria é a coleta in situ de medições ou outros dados em pontos remotos e sua transmissão automática para equipamentos receptores (telecomunicações) para monitoramento.

- *Wikipedia, Telemetry*

Observabilidade

Uma medida de quão bem se consegue entender e explicar qualquer estado em que seu sistema possa entrar, não importa quão novo ou bizarro [...] sem a necessidade de desenvolver novo código.

- *Observability Engineering, cap 1, What Is Observability?*

Porquê?

“Os clientes da nossa loja online
não conseguem iniciar o
checkout!!”



Como identificar a origem do
problema?

Uh oh!



Looks like something is wrong on our side.

If the problem persists, **contact our Support Team.**

“Precisamos de definir o orçamento para o ano que vem. Quanto é que temos de alocar para a infraestrutura de IT?”



Que dados precisamos para responder a esta pergunta?



“Este processo está a ficar cada vez mais lento. Temos de melhorar a performance.”



Como é que identificamos o que está a tornar o processo lento?



“Li um artigo sobre encoding binário versus texto, e acho que seria interessante para nós. Que devo fazer para perceber se vale a pena?”



Que dados é que podem influenciar esta decisão?



O que monitorizar?

Os Quatro Sinais Dourados

Tráfego

Latência

Erros

Saturação

Perspectivas diferentes, sinais diferentes

USE

“Para cada recurso, medir:

- Utilização (% tempo que o recurso esteve ocupado)
- Saturação (quantidade de trabalho que o recurso tem para fazer, tipicamente o comprimento de uma fila)
- Erros (contagem de erros)”

RED

“Para cada recurso [serviço] medir:

- Rácio (número de pedidos por segundo)
- Erros (quantos desses pedidos estão a falhar)
- Duração (quantidade de tempo que esses pedidos demoram)”

Perspectivas diferentes, sinais diferentes

USE

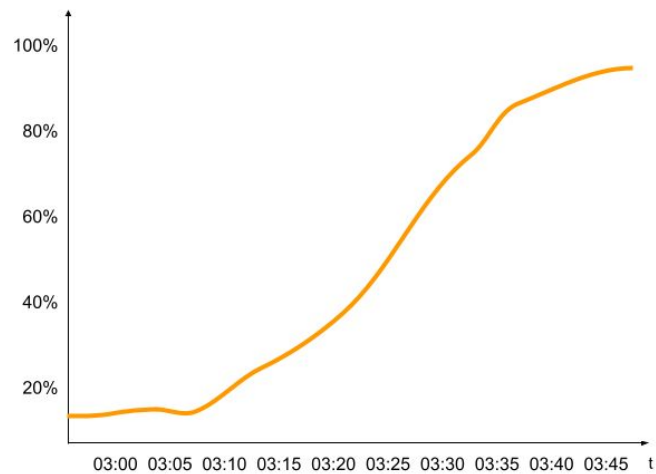
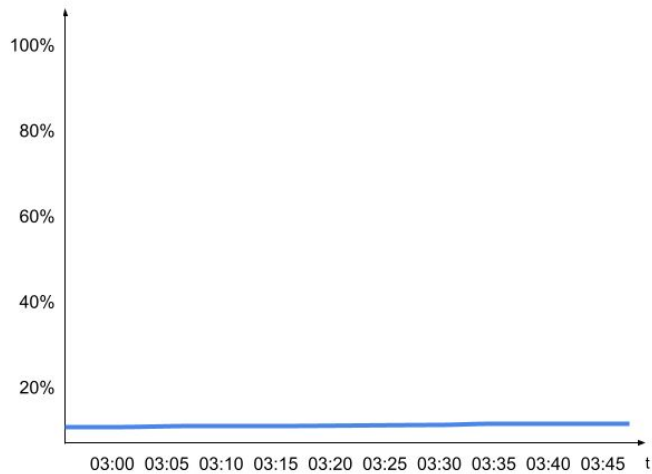
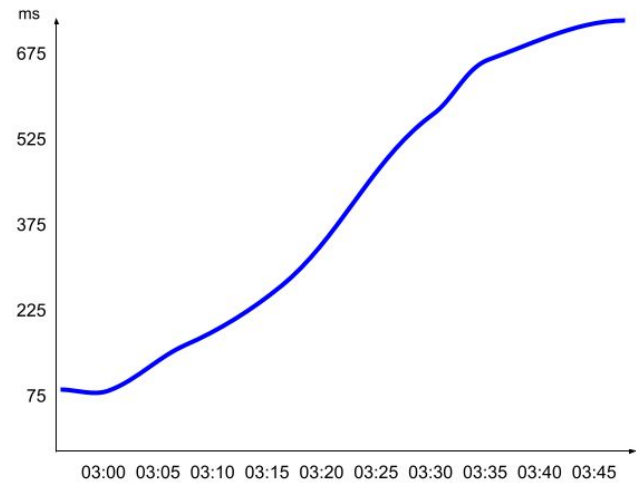
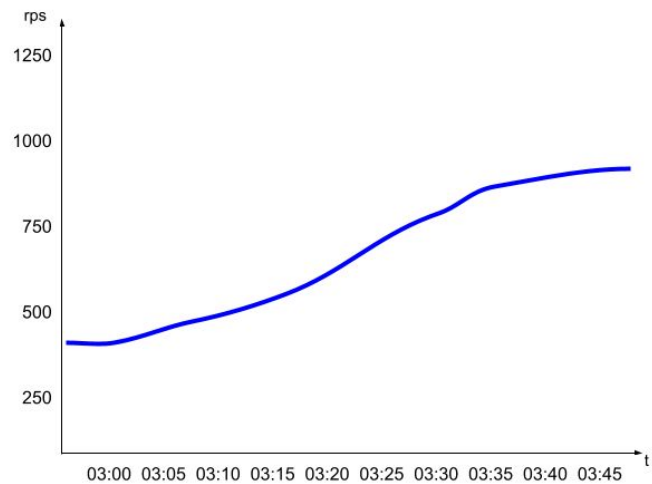
“Para cada recurso, medir:

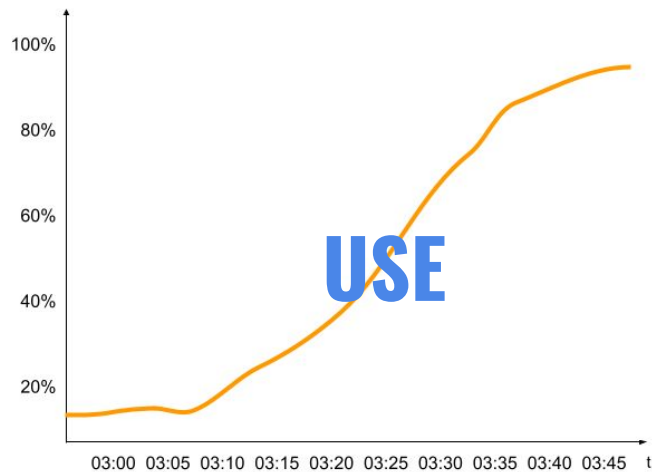
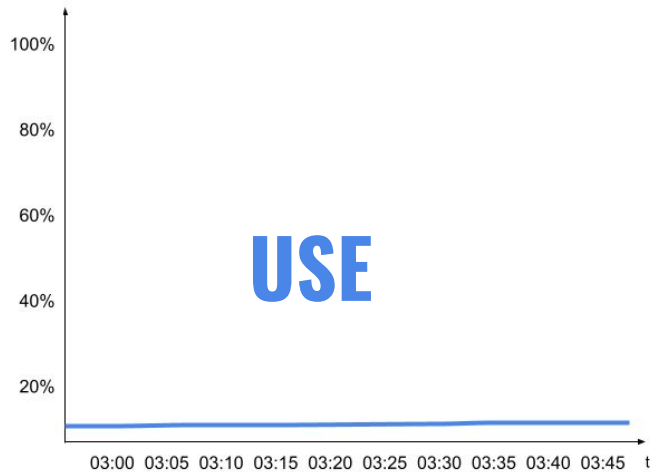
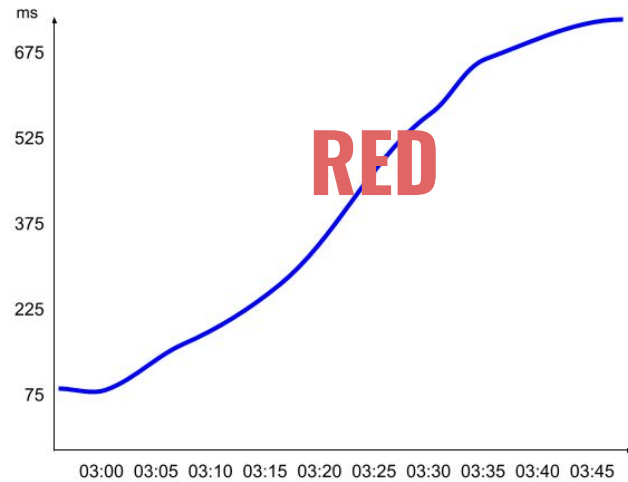
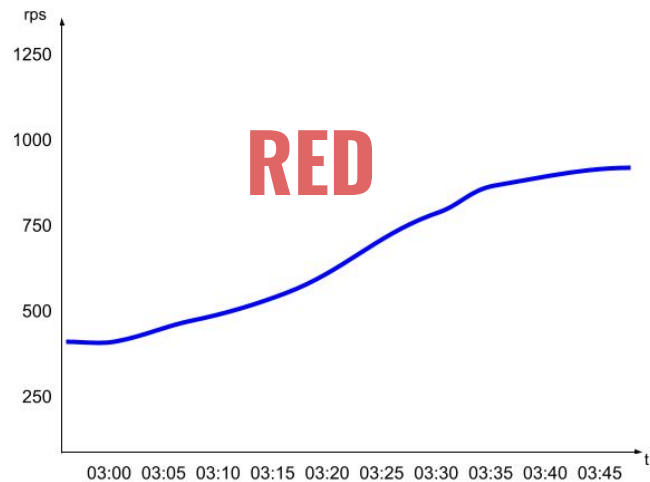
- **Utilização** (% tempo que o recurso esteve ocupado)
- **Saturação** (quantidade de trabalho que o recurso tem para fazer, tipicamente o comprimento de uma fila)
- **Erros** (contagem de erros)”

RED

“Para cada recurso [serviço] medir:

- **Rácio** (número de pedidos por segundo)
- **Erros** (quantos desses pedidos estão a falhar)
- **Duração** (quantidade de tempo que esses pedidos demoram)”





Arquitecturas diferentes, sinais diferentes

	RPC	Eventos/Mensagens	Frontend/Cliente
Tráfego	Número de pedidos	Número de Mensagens (backlog)	Número de pedidos, número de utilizadores concorrentes
Latência	Duração de um pedido	Tempo para processar uma mensagem	Tempo de renderização, duração dos pedidos
Erros	Quantos pedidos terminaram em erro	Mensagens com erro (Dead Letter Queue)	Erros de renderização, erros de pedidos
Saturação	CPU, Memória, Rede, Discos	CPU, Memória, Rede, Discos	Tamanho dos payloads

Arquitecturas diferentes, sinais diferentes

	RPC	Eventos/Mensagens	Pipelines
Tráfego	Número de pedidos	Número de Mensagens (backlog)	???
Latência	Duração de um pedido	Tempo para processar uma mensagem	???
Erros	Quantos pedidos terminaram em erro	Mensagens com erro (Dead Letter Queue)	???
Saturação	CPU, Memória, Rede, Discos	CPU, Memória, Rede, Discos	???

Checkpoint: Que sinais estão em falta?

Erro no checkout

Capacity Planning

Problema de latência

Mudar para encoding binário

Monitorização não é só para conceitos técnicos

Sinais de alto nível

Tempo Real + Dentro da aplicação = perspectiva única de código e negócio!

Como? → Trazer as abstrações que definimos no código para a nossa telemetria

Sinais de alto nível

Exemplo 1: Medir tipos de erros

27 Excepções

– Hmmmm... OK 🤔

27 Excepções

– Muito melhor!! 🙌

- 17 por falha de autenticação
- 1 por falha de pagamento
- 9 por artigos fora de stock

Sinais de alto nível

Exemplo 2: Performance de operações VS Performance de serviços

1 Serviço → 10 operações → 70% erros

“Que operação é que está a falhar?”

- “Adicionar ao carrinho” - 0,5% erros 👍
- “Adicionar à wishlist” - 0,1% erros 👍
- “Completar pagamento” - 95% erros 😱

Sinais de alto nível

Exemplo 3: Abstração do mecanismo de transporte

API REST + Graceful degradation => “200 OK” + erro na aplicação

90% Erros → 100% “200 OK”

Soluções possíveis:

- Métrica com erros silenciosos
- Tracing com Span Status = Erro

Sinais de alto nível

Conceitos de alto nível são os **melhores sinais para usar com alertas.**

Alertas com base em sintomas

(Symptom Based Alerting)

Tarefa: Quais são as operações críticas de uma loja online?

Como obter os sinais?

Instrumentação – a prática de adicionar a uma aplicação
(de forma manual ou automática)
comandos que geram outputs sobre o estado
de uma aplicação (a sua performance, erros,
execuções, etc)



A diagram of a classical temple with three columns. The pediment (roof) contains the word 'OBSERVABILIDADE'. Each column contains a word: 'MÉTRICAS' on the left, 'LOGS' in the center, and 'TRACES' on the right. The temple sits on a multi-tiered base.

OBSERVABILIDADE

MÉTRICAS

LOGS

TRACES



OBSERVABILIDADE

MÉTRICAS

LOGS

TRACES

PROFILING

Métricas

“Uma métrica é uma medida de um serviço capturado em tempo de execução. O momento de captura de uma medida é conhecido como evento métrico, que consiste não apenas na medida em si, mas também no momento [timestamp] em que ela foi capturada e nos metadados associados.”

Métricas

Ideal para:

- Dados numéricos (tipicamente agregados) visualizados ao longo do tempo
- Identificar tendências

Métricas

nome	atributos/dimensões	tipo	unidade	timestamp	valor
requests	Host-ID: 123jhjh66kj Target: /api/v1/foo Status-Code: 200	Integer	requests	1517423321	3

Métricas

Simple graph

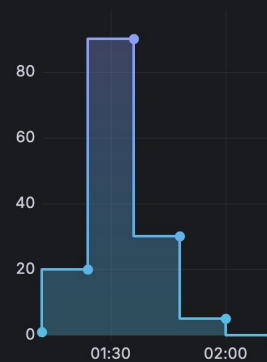


Interpolation modes

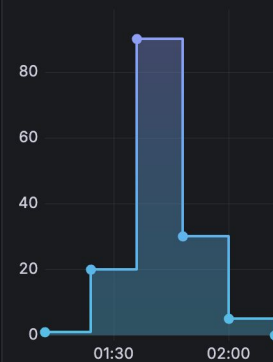
Interpolation mode: smooth



Interpolation mode: Step before



Interpolation mode: Step after



Métricas - Instrumentos

Contador

Um valor que vai acumulando ao longo do tempo - será algo como o conta quilómetros dum carro; o valor só aumenta.

Medições com um contador são normalmente síncronas.

Tipicamente tem sub-tipos, ou variações.

Medidor (Gauge)

Mede um valor atual no momento em que é lido. Um exemplo seria o medidor de combustível de um carro. Os medidores são assíncronos.

Histograma

Agregação de valores em classes (buckets). Esses valores são estatisticamente significativos. Exemplos de uso de histograma são: latência de pedidos, ou tamanho (em Bytes) de respostas.

Distributed Tracing

“Os rastreamentos [traces] dão uma visão geral do que acontece quando um pedido é feito a uma aplicação. Quer a aplicação seja um monolito com uma única base de dados ou um conjunto sofisticado de serviços, os rastreamentos são essenciais para entender o ‘caminho’ completo que um pedido percorre na aplicação.”

Distributed Tracing

Ideal para:

- Acompanhar uma transação (pedido) ao longo de vários serviços
- Identificar a origem de erros ou fraca performance

Fun fact! → Também permite gerar métricas RED para uma determinada operação (implicitamente).

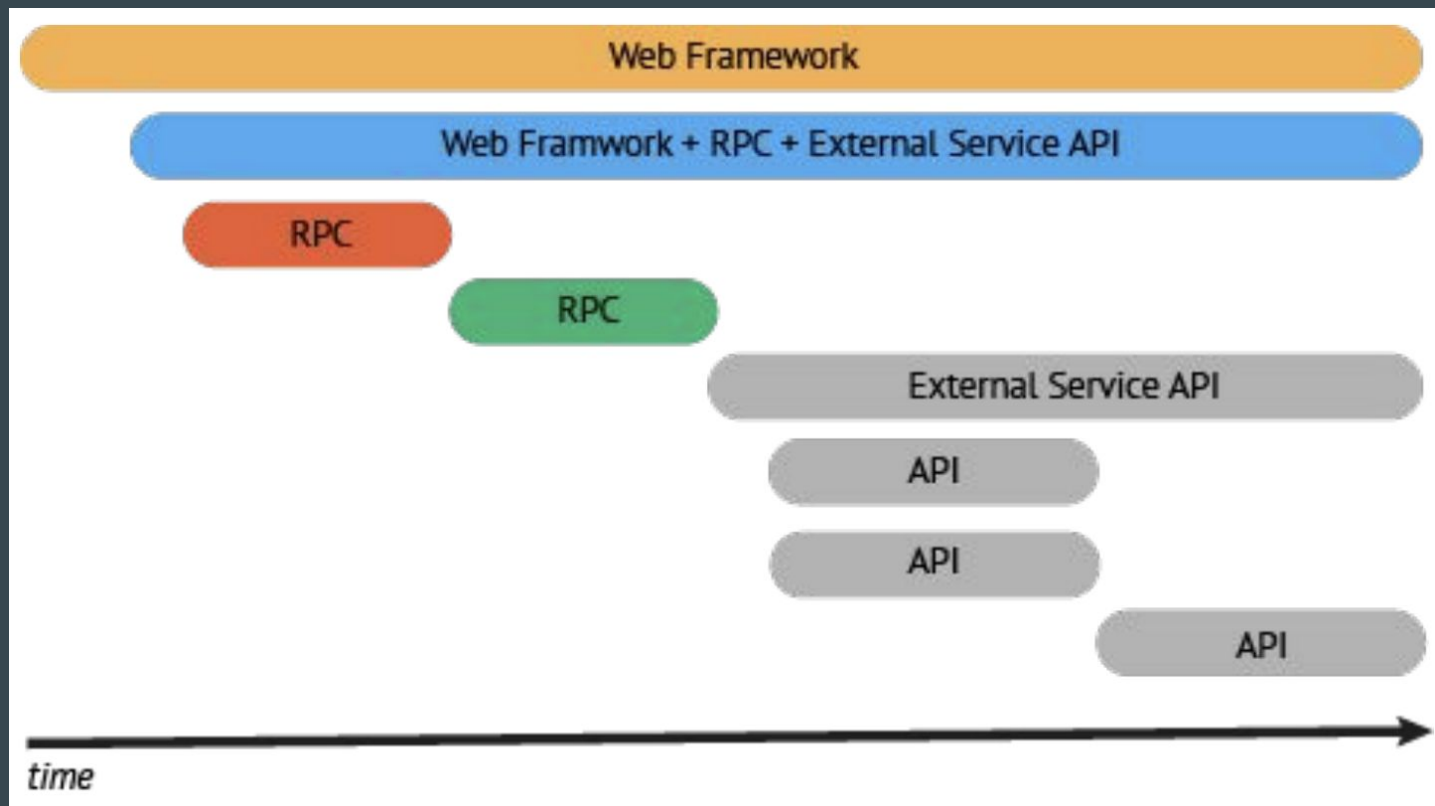
Distributed Tracing

name	context	parent_id	start_time	end_time	attributes	events
------	---------	-----------	------------	----------	------------	--------

Distributed Tracing

```
{
  "name": "hello",
  "context": {
    "trace_id": "0x5b8aa5a2d2c872e8321cf37308d69df2",
    "span_id": "0x051581bf3cb55c13"
  },
  "parent_id": null,
  "start_time": "2022-04-29T18:52:58.114201Z",
  "end_time": "2022-04-29T18:52:58.114687Z",
  "attributes": {
    "http.route": "some_route1"
  },
  "events": [
    {
      "name": "Guten Tag!",
      "timestamp": "2022-04-29T18:52:58.114561Z",
      "attributes": {
        "event_attributes": 1
      }
    }
  ]
}
```


Distributed Tracing



Logs

“Um log é um texto com timestamp, estruturado (recomendado) ou não estruturado, com metadados. (...) Os logs geralmente contêm informações detalhadas de debugging/diagnóstico, como inputs de uma operação, o resultado da operação e quaisquer metadados de suporte para essa operação.”

Logs

Ideal para:

- Registrar eventos isolados de uma aplicação (startup, shutdown, etc.)
- Obter telemetria de forma simplificada

Logs

timestamp	atributos	mensagem
1517423321	Host-ID: 123jhjh66kj Application: ServiceFoo	“lorem ipsum dolor sit amet”

Logs

Search for log entries... (e.g. host.name:host-1)			Customize	Highlights	Last 1 day	Stop streaming
Oct 8, 2020	event.dataset	Message				
		t.elastic.dev", referrer: "https://www.elastic.co/cn/downloads/hadoop"				10 AM
09:41:36.000	nginx.access	[nginx][access] 115.23.241.49 undefined "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0				11 AM
09:41:36.000	nginx.access	[nginx][access] 10.52.3.152 undefined "POST /intake/v2/events HTTP/1.1" 400 388706				12 PM
09:41:36.000	nginx.access	[nginx][access] 119.4.208.128 undefined "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0				01 PM
09:41:36.000	nginx.error	[nginx][warn] a client request body is buffered to a temporary file /tmp/client-body/0000065586, client: 10.52.3.152, server: apm.3.5.241.204.195.ip.es.io, request: "POST /intake/v2/events HTTP/1.1", host: "apm.35.241.204.195.ip.es.io"				02 PM
		10.52.3.152 -- [08/Oct/2020:08:41:36 +0000] "POST /intake/v2/events HTTP/1.1" 400 388706 "-" "elasticapm-python/5.9.0" 452587 0.032 [default-apm-apm-server-8200] [] 10.52.3.144:8200 387943 0.032 400 5a2035017a852f61cebabd15f18c39ee				03 PM
		119.4.208.128 -- [08/Oct/2020:08:41:36 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 652 0.001 [default-apm-apm-server-8200] [] 10.52.6.126:8200 0.001 200 5b43dbc01de84498ccbcfc95c5e528cfc				04 PM
		82.197.49.139 -- [08/Oct/2020:08:41:37 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/blog/customizing-your-document-routing" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 645 0.001 [default-apm-apm-server-8200] [] 10.52.3.144:8200 0.001 200 cd7de0fa90e7d8e8b442a4631e463a14				05 PM
		115.23.241.49 -- [08/Oct/2020:08:41:37 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/kr/pricing/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 639 0.001 [default-apm-apm-server-8200] [] 10.52.6.126:8200 0.001 200 b6cf0075e370e39dc4640b2426943f29				06 PM
		192.109.246.130 -- [08/Oct/2020:08:41:37 +0000] "POST /intake/v2/rum/events HTTP/1.1" 202 0 "https://www.elastic.co/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36" 4293 0.005 [default-apm-apm-server-8200] [] 10.52.3.144:8200 0.006 202 73909003a3d8323437c7797f3cabae44				07 PM
		86.98.44.36 -- [08/Oct/2020:08:41:37 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36" 729 0.001 [default-apm-apm-server-8200] [] 10.52.3.144:8200 0.001 200 385f4cf14a89e33820e77e89ae4c6be				08 PM
		203.251.156.66 -- [08/Oct/2020:08:41:37 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36" 757 0.001 [default-apm-apm-server-8200] [] 10.52.6.126:8200 0.001 200 d23f84b9c620cb53b503914ecbdb47b4				09 PM
		115.23.241.49 -- [08/Oct/2020:08:41:37 +0000] "POST /intake/v2/rum/events HTTP/1.1" 202 0 "https://www.elastic.co/kr/pricing/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 2414 0.001 [default-apm-apm-server-8200] [] 10.52.6.126:8200 0.001 202 c457491b031acef856a4bbb395974fb4				10 PM
		91.126.218.188 -- [08/Oct/2020:08:41:38 +0000] "OPTIONS /intake/v2/rum/events HTTP/1.1" 200 0 "https://www.elastic.co/siem" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:81.0) Gecko/20100101 Firefox/81.0" 620 0.001 [default-apm-apm-server-8200] [] 10.52.3.144:8200 0.001 200 2c6cf2692ca0960f767acf1df745516b				11 PM
						Thu 08
						01 AM
						02 AM
						03 AM
						04 AM
						05 AM
						06 AM
						07 AM
						08 AM

Last update 7 seconds ago

OpenTelemetry

*“OpenTelemetry, também conhecido como OTel, é uma **framework de observabilidade open source, neutra em termos de fornecedor, para instrumentação, geração, coleta e exportação** de dados de telemetria, como tracing, métricas e logs .”*

*“**Como especificação da indústria** , OpenTelemetry é suportado por mais de 40 fornecedores de observabilidade, integrado por muitas bibliotecas, serviços e aplicações, e adotado por vários utilizadores finais.”*

OpenTelemetry Spec

API

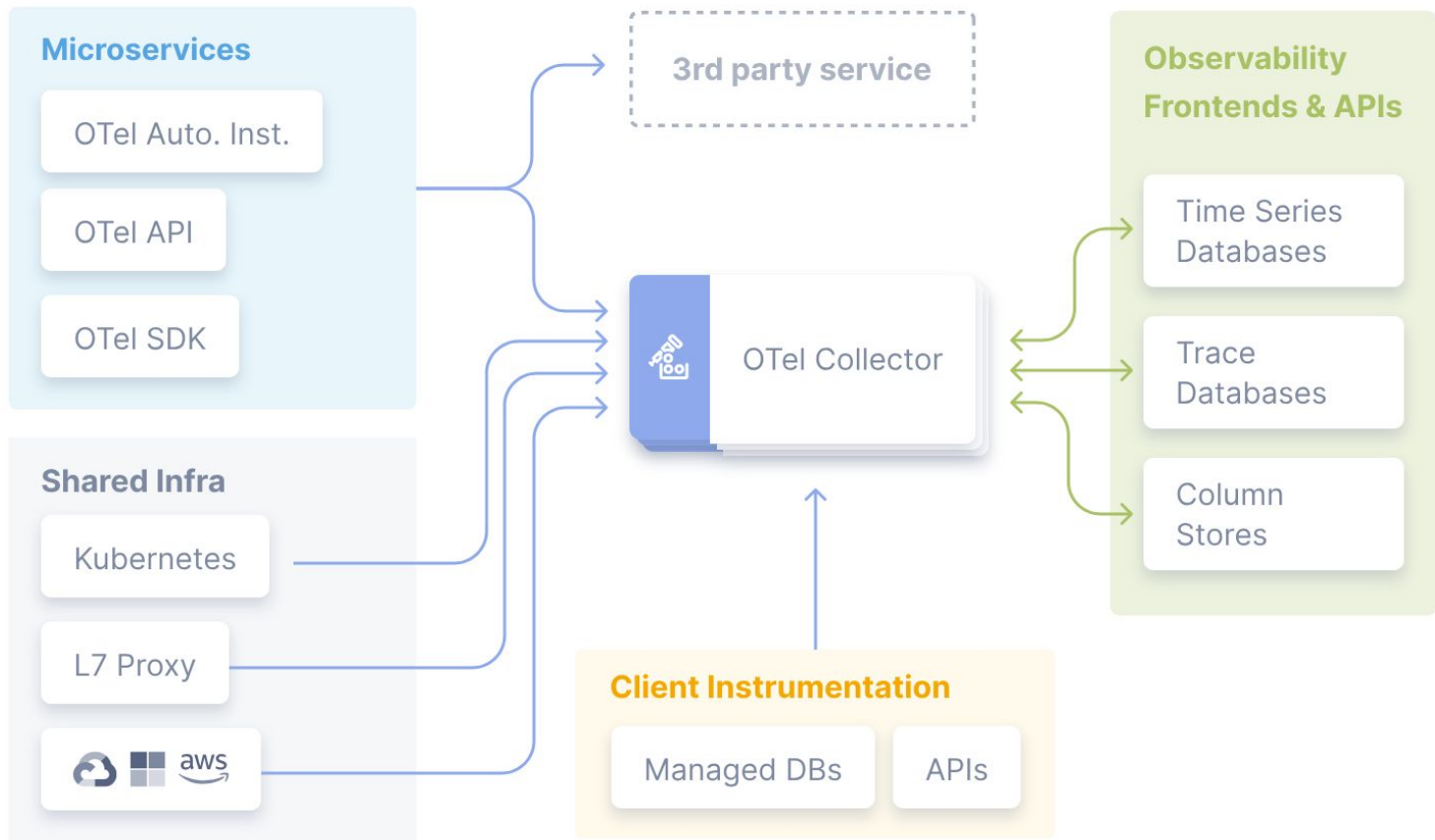
- Instrumentação
- Collector

OpenTelemetry Protocol (OTLP)

- Como os dados são codificados e transportados

Semantic Conventions

- Boas práticas sobre instrumentação
- Definição de conceitos comuns, e como os representar em telemetria



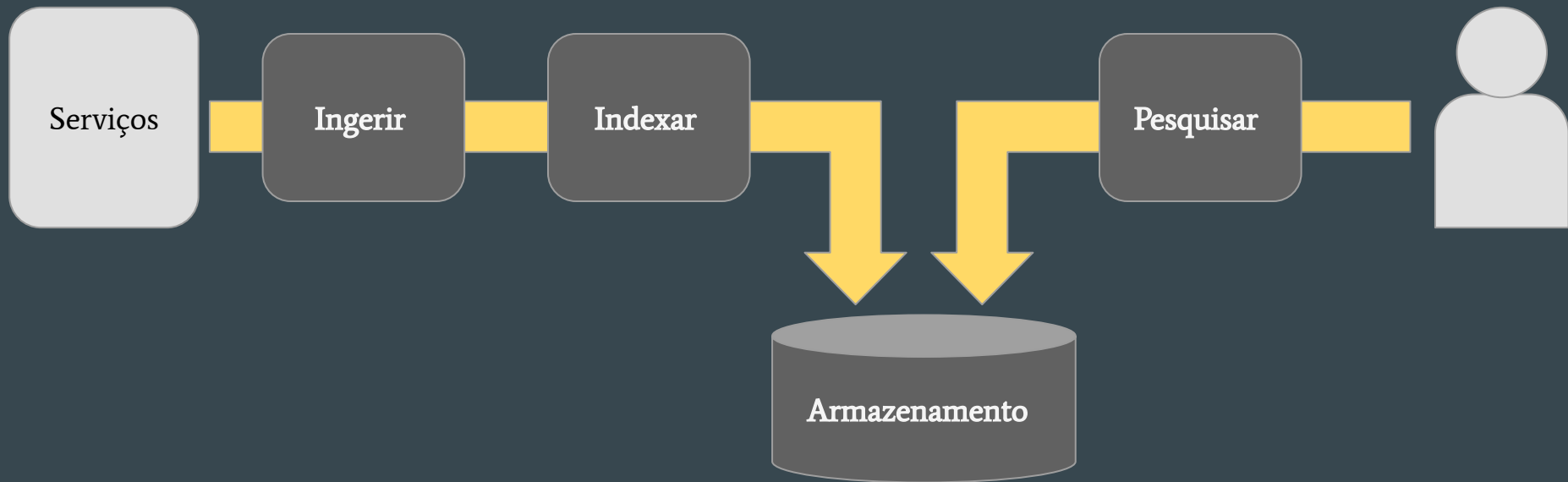
Uma palavra sobre open standards

Open standards maximizam **interoperabilidade**

Ferramentas proprietárias podem apresentar limitações.



Backend de Telemetria



Tarefa: Obter contagem de pedidos com Logs e com Métricas

Outras coisas a explorar

Analytics

- Não é real time, mas é excelente para detetar padrões, anomalias, e juntar dados de fontes diferentes.

Real User Metrics (RUM) + Web Vitals

- Métricas chave para Frontend e que influenciam a performance de uma página nas pesquisas orgânicas em motores de busca (Search Engine Optimization)

Checkpoint: Que pilar usar para os sinais que identificamos antes?

Erro no checkout

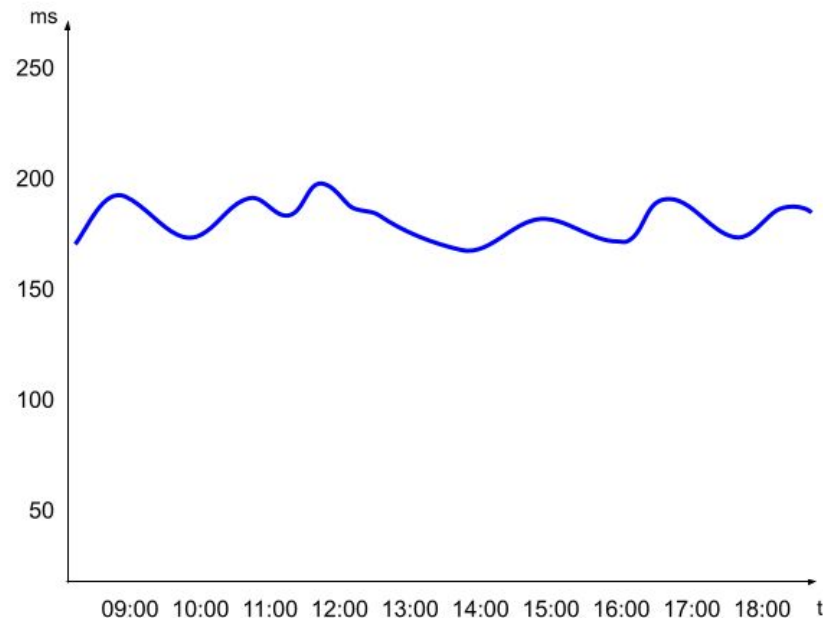
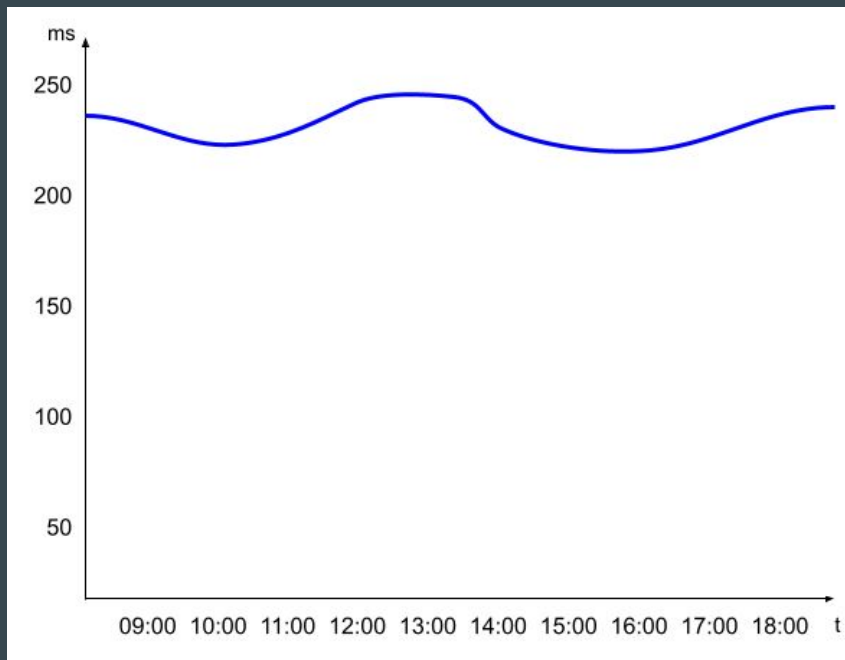
Capacity Planning

Problema de latência

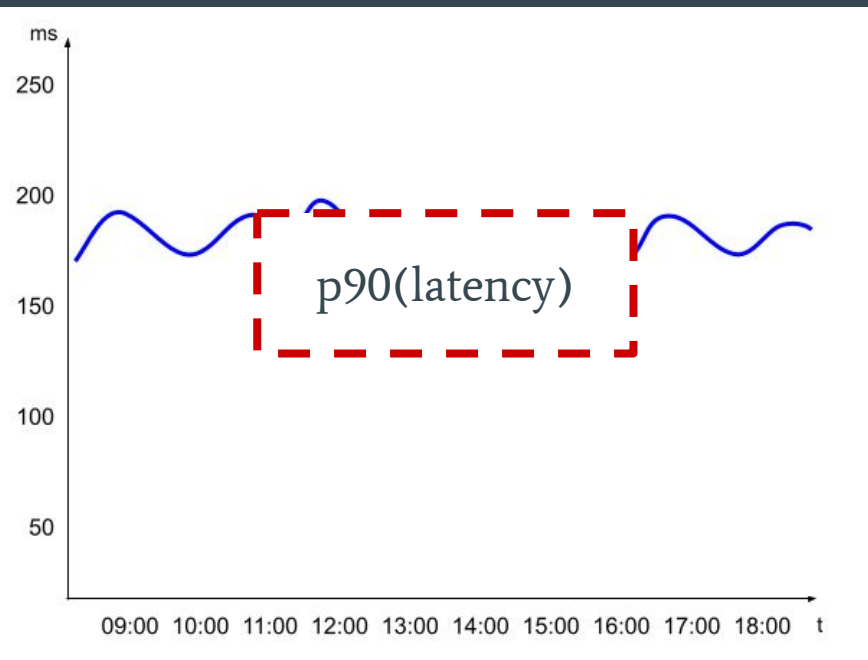
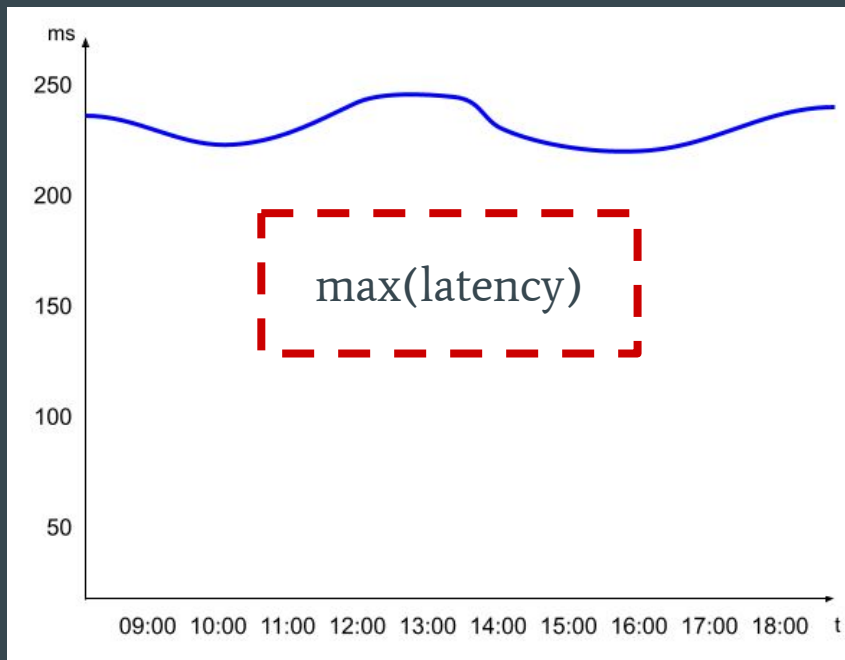
Mudar para encoding binário

Como interpretar os sinais?

Qual é o sistema mais lento?



Qual é o sistema mais lento?



Vamos precisar de matemática

Tarefa: Como agregar valores para obtermos a latência global dum serviço?

Serviço Foo

Utilizador X

Latência = 150 ms

Utilizador Y

Latência = 150 ms

Utilizador Z

Latência = 150 ms

Utilizador K

Latência = 30 ms

Serviço Foo

Utilizador X

Latência = 150 ms

Utilizador Y

Latência = 150 ms

Utilizador Z

Latência = 150 ms

Utilizador K

Latência = 30 ms

Média da latência = 120ms

Serviço Foo :: Latência deve estar abaixo de 120ms

Utilizador X

Utilizador Y

Utilizador Z

Utilizador K

Latência = 150 ms

Latência = 150 ms

Latência = 150 ms

Latência = 30 ms

Média da latência = 120ms 

Serviço Foo :: Latência deve estar abaixo de 120ms

Utilizador X

Utilizador Y

Utilizador Z

Utilizador K

Latência = 150 ms

Latência = 150 ms

Latência = 150 ms

Latência = 30 ms

Média da latência = 120ms 👍

Mas... 75 % dos utilizadores acima da latência alvo 😱

Vamos precisar de matemática

Latência → Percentis, ou
Histogramas

Tráfego, Erros → Rates

Saturação → Sem agregação

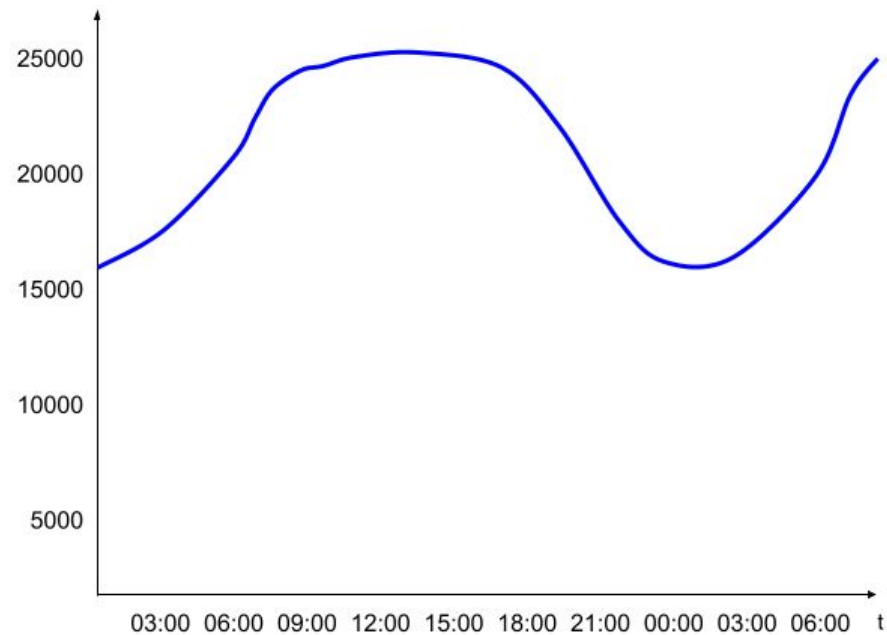
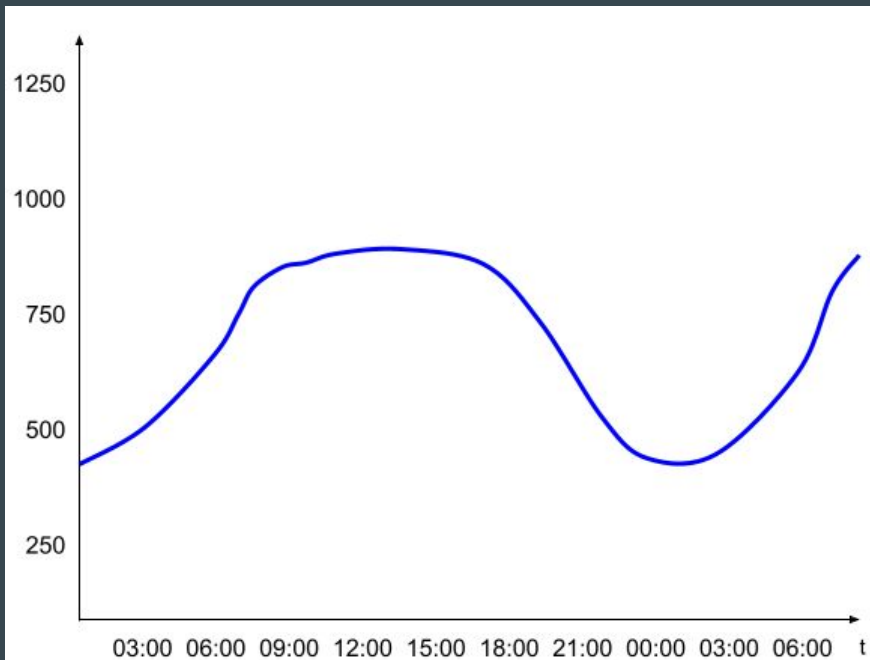
Min, Max, Mediana, etc...

Valores absolutos VS Rácios

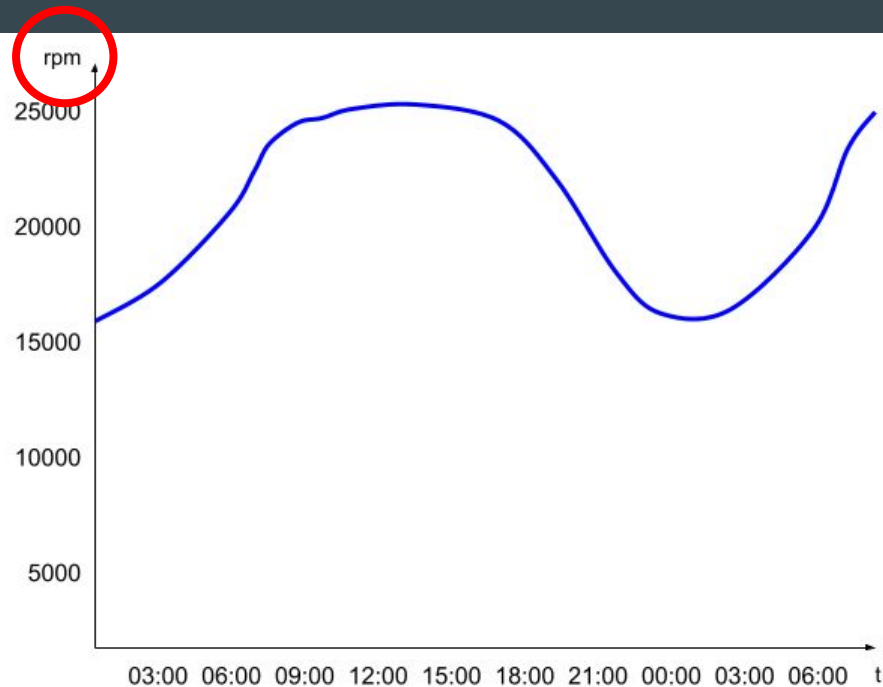
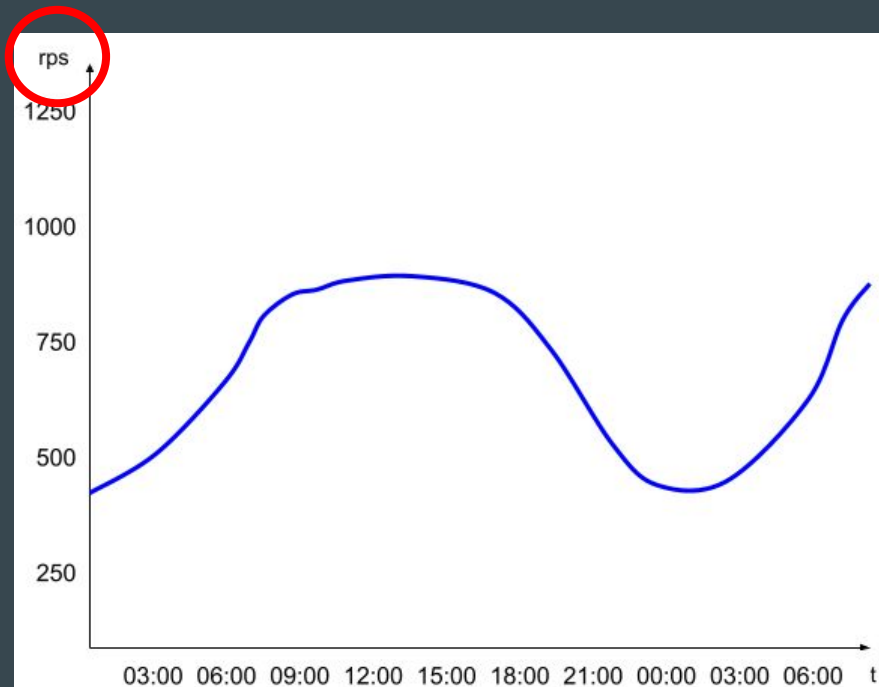
- 10 Erros
 - Em 12 operações 🤯
 - Em 10.000 operações 🙄

Escala Logarítmica

Qual é o sistema com mais tráfego?



Qual é o sistema com mais tráfego?



Granularidade

Granularidade com que medimos, armazenamos, e consumimos

Reduzir Granularidade 

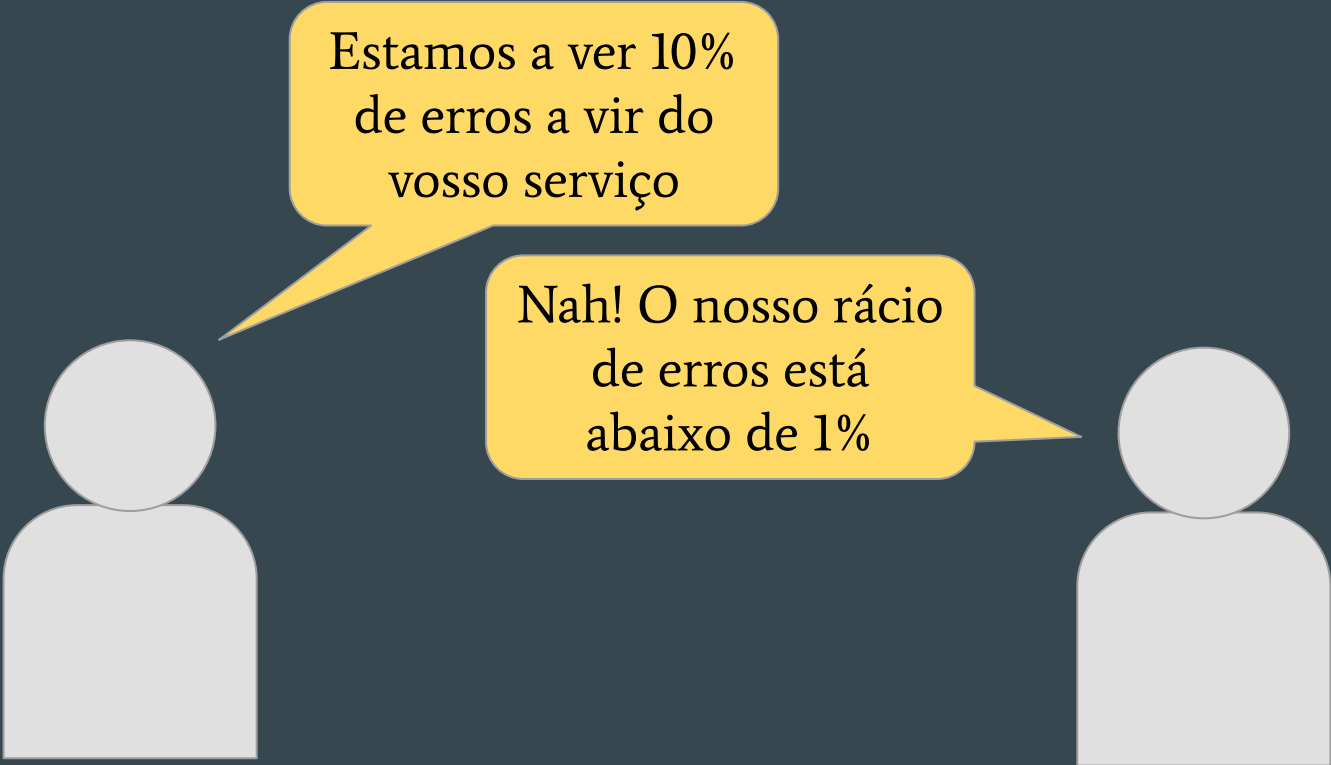
- $B \rightarrow MB \rightarrow GB$
- $s \rightarrow min \rightarrow h$

Aumentar Granularidade... 

“Numa hora transmitimos 10GB”

“E quanto é que fomos transmitindo por minuto?”

Quem é que tem razão?

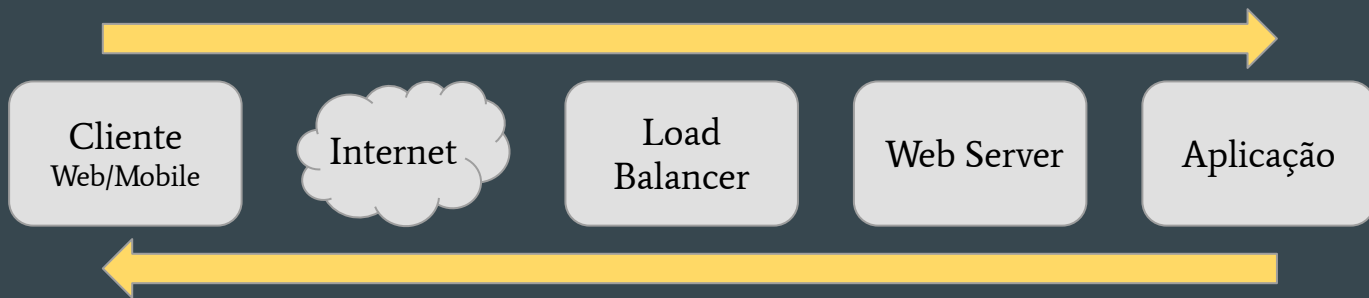


Estamos a ver 10%
de erros a vir do
vosso serviço

The diagram shows two stylized human figures, one on the left and one on the right, represented by white circles for heads and white rounded rectangles for bodies. They are positioned against a dark blue background. A yellow speech bubble originates from the left figure and points towards the center. Another yellow speech bubble originates from the right figure and points towards the center, overlapping the first one. The text inside the speech bubbles is in a black serif font.

Nah! O nosso rácio
de erros está
abaixo de 1%

A perspectiva conta



Outros conceitos

Logs estruturados

- Antes: 1746652595 INFO 435 123-43 Something happened
- Depois: timestamp: 746652595 level: INFO user_id: 435 server_id: 123-43 message: "Something happened"

Dimensões / Atributos

- Descrever os dados de telemetria
- 115 pedidos total
 - 40 pedidos /home
 - 30 pedidos /login
 - 45 pedidos /terms

Checkpoint: Que unidades e funções usar para os nossos sinais?

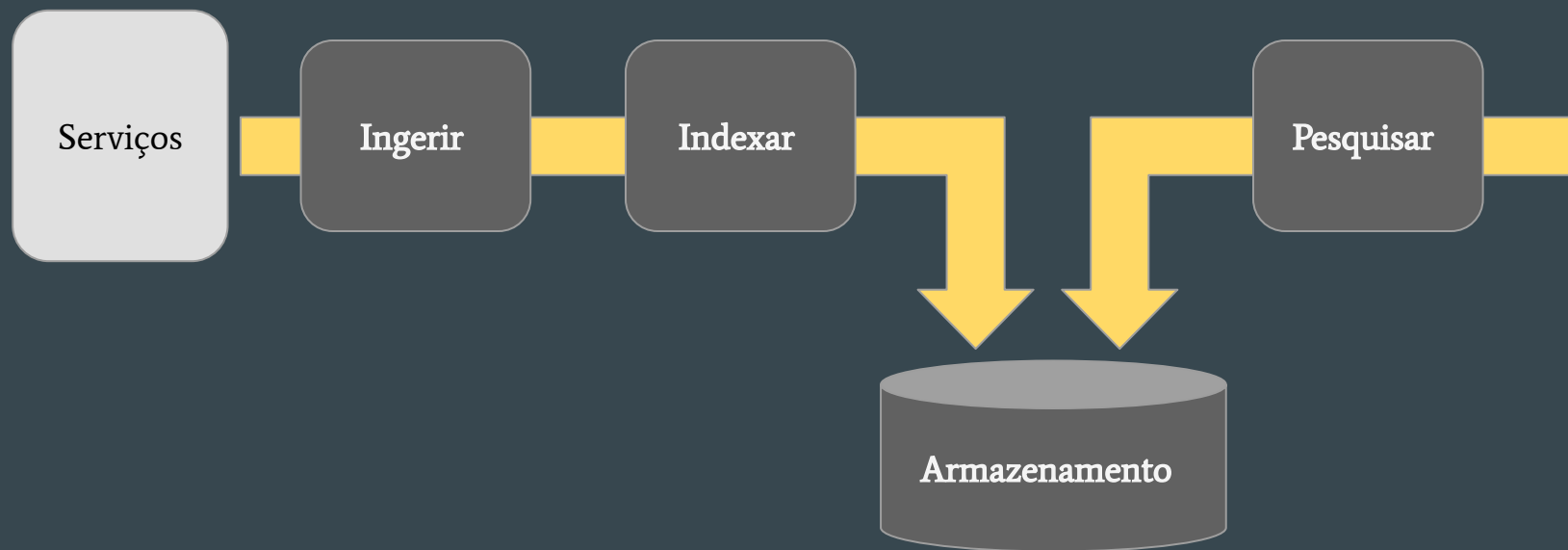
Erro no checkout

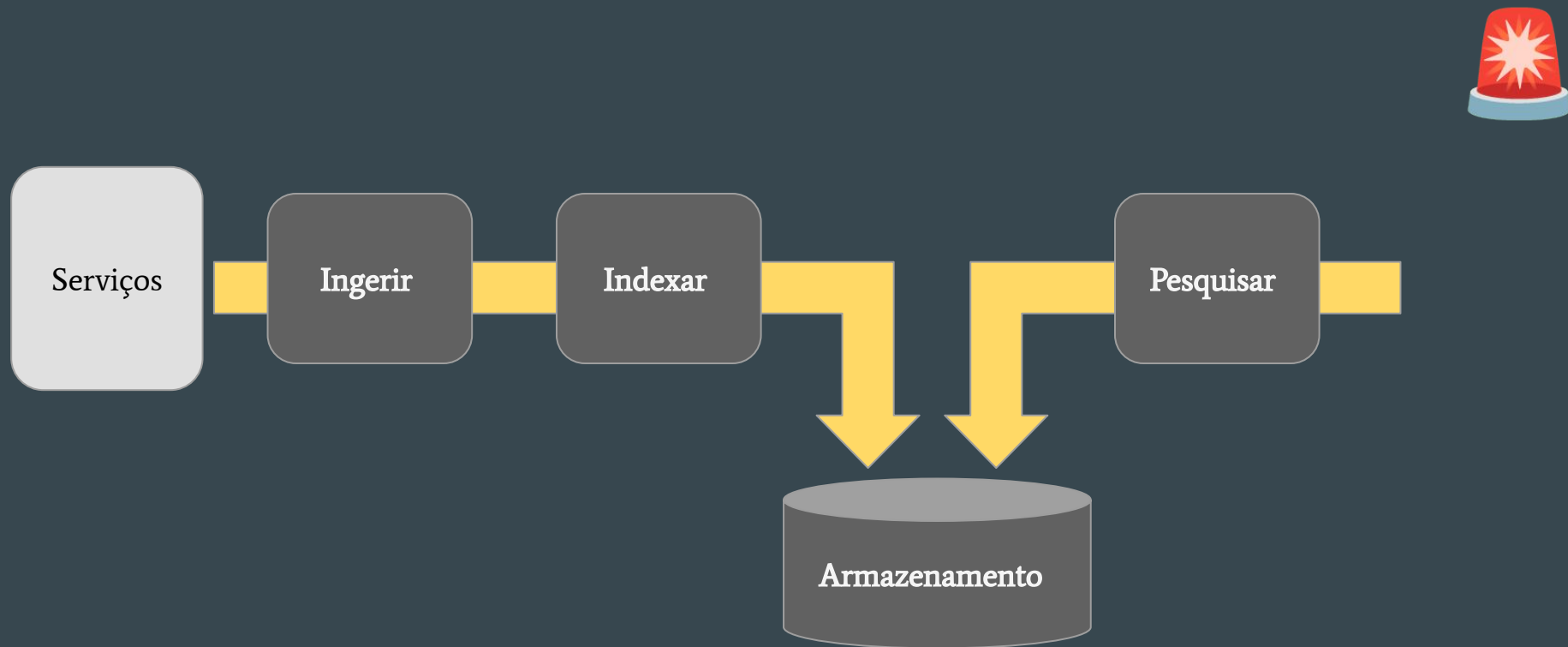
Capacity Planning

Problema de latência

Mudar para encoding binário

Que uso dar os sinais gerados?





Auto Scaling → “Se CPU > x%, aumentar em 1 instância”



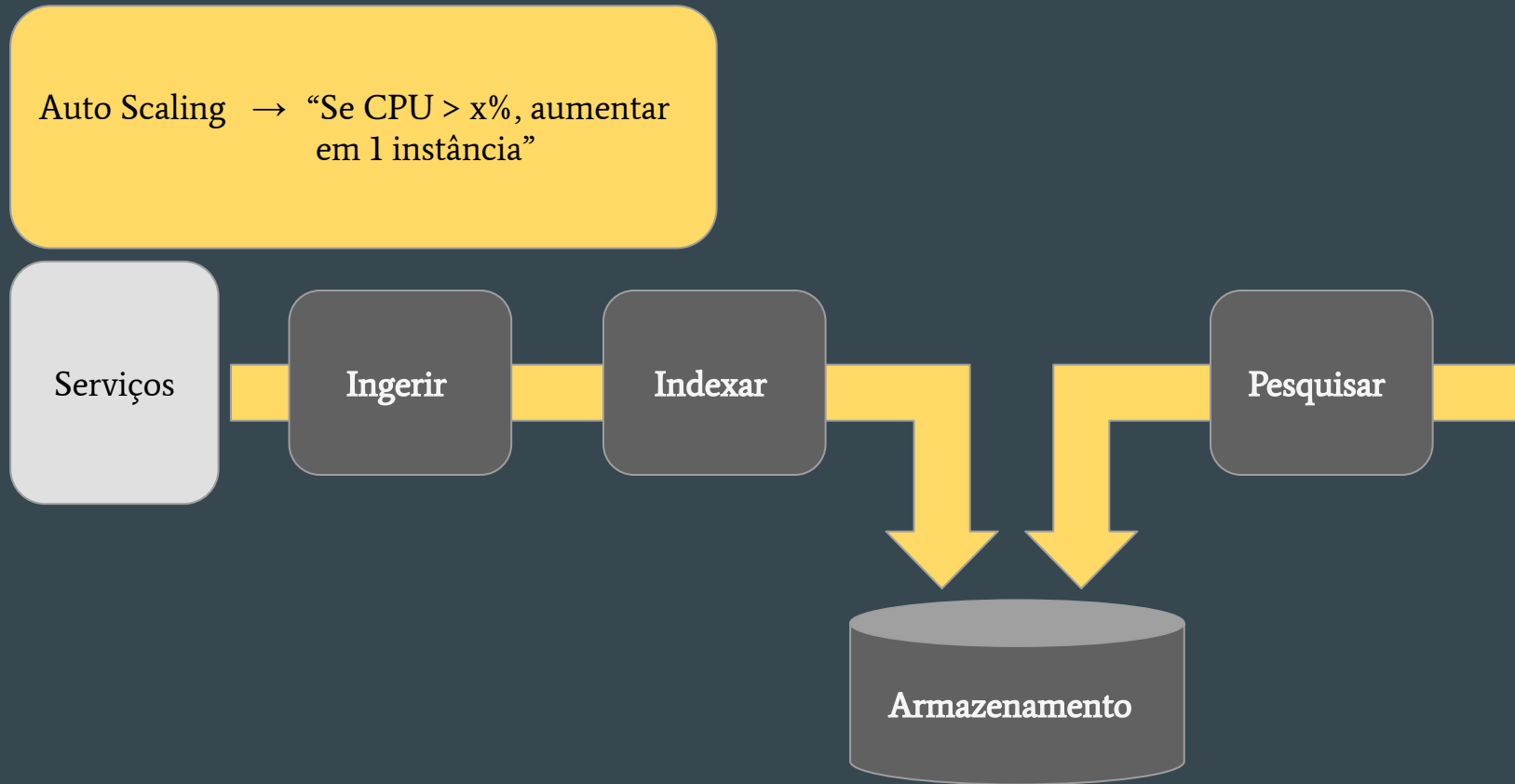
Serviços

Ingerir

Indexar

Pesquisar

Armazenamento



Circuit breakers → “Abrir circuito se rácio de erros
>x% durante y segundos”

Auto
Scaling

Serviços

Ingerir

Indexar

Pesquisar

Armazenamento



Resumindo...

Telemetria tem muitos usos

Encontrar erros | Planear capacidade | Mudanças de infraestrutura | Medir performance

4 Sinais Dourados

Tráfego | Erros | Latência | Saturação

Ajustar os sinais à arquitetura

RPCs | Mensagens | Frontend | ...

3 Pilares da Observabilidade

Métricas | Tracing | Logs

OpenTelemetry

Saber como interpretar os sinais

Como são agregados? Onde são medidos?

Sinais de alto nível

Abstrações de código trazidas para a telemetria

Aplicar os sinais onde fazem mais sentido

Alertas | Dashboards | Automatismos

Questões?

Links

Livros:

[Site Reliability Engineering](#)

[Distributed Systems Observability](#)

[Observability Engineering](#)

DIY Observabilidade:

[OpenTelemetry](#)

[Jaeger \(Ferramenta de Tracing\)](#)

[Prometheus \(Backend de Métricas\)](#)

[Grafana \(Dashboards\)](#)