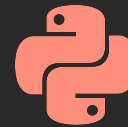




Introdução a observabilidade

Live de Python # 261



1. Um passo atrás

Teoria de controle

2. Trazendo para o nosso mundo

Aplicações python

3. Telemetria

A arte de sinalizar o que acontece enquanto acontece

4. Observabilidade

Entender o que se passou pelos sinais de telemetria

Título	Data
Observabilidade - Uma introdução	Agora!
Observabilidade - Métricas (Opentelemetry / Prometheus)	25/03
Observabilidade - Rastreamento / Tracing (Opentelemetry / Tempo / Jeager)	08/04
Observabilidade - Logs (Opentelemetry / Loki)	15/04



Isso é uma série!

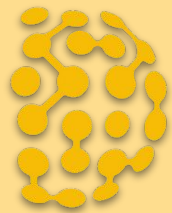


- Não sou especialista em observabilidade
 - Quero compartilhar as coisas que aprendi nos últimos meses estudando
- Não falaremos sobre eBPF
- Não falaremos sobre profiling
 - Temos live sobre em Python puro
- Não falaremos sobre monitoramento máquinas
 - clusters, vms, containers, ...
 - O Jeff tem VÁRIOS vídeos sobre isso, valem a pena! #VAI

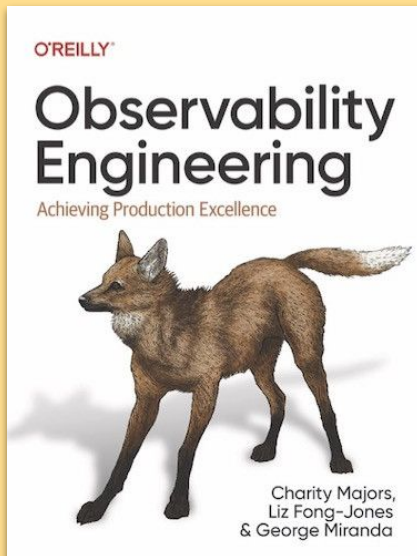


Disclaimers





Dose de Telemetria



cassiobotaro/explorando-observabilidade



✖ Explorando e tomando notas sobre o mundo de observabilidade com Open Telemetry.

1
Contributor

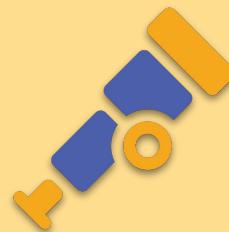
0
Issues

14
Stars

1
Fork



<https://oilly.news/>



OpenTelemetry



Fontes legais pra estudar comigo <3



Um
passo
atrás

Teoria de controle

Uma volta para 1960



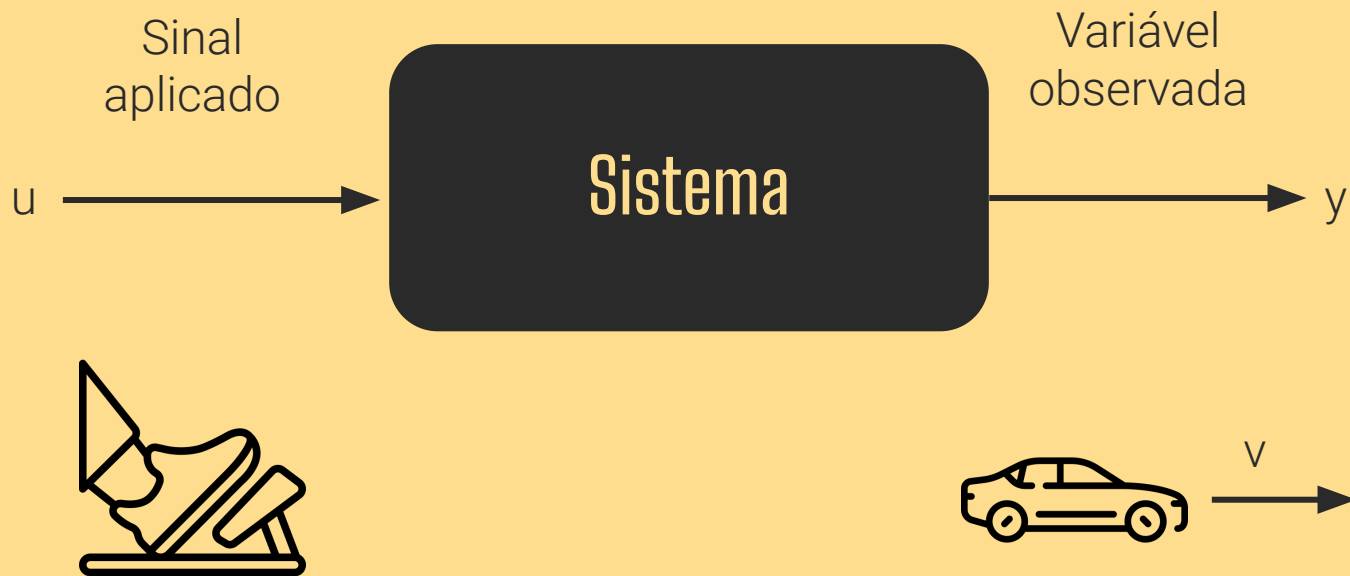
Na matemática/engenharia existe uma grande área chamada **teoria de controle** introduzida por R. E. Kalman.



O exemplo clássico



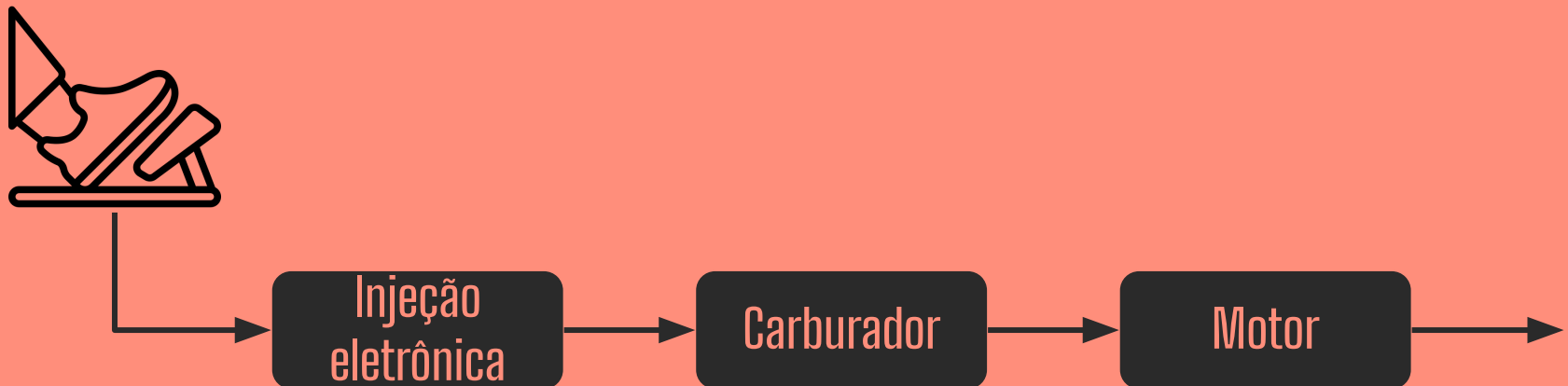
Um carro sendo dirigido. Existe um sinal aplicado quando você aperta o acelerador. Por fim, a velocidade do carro é controlada por esse sinal.



Abrindo a caixa preta



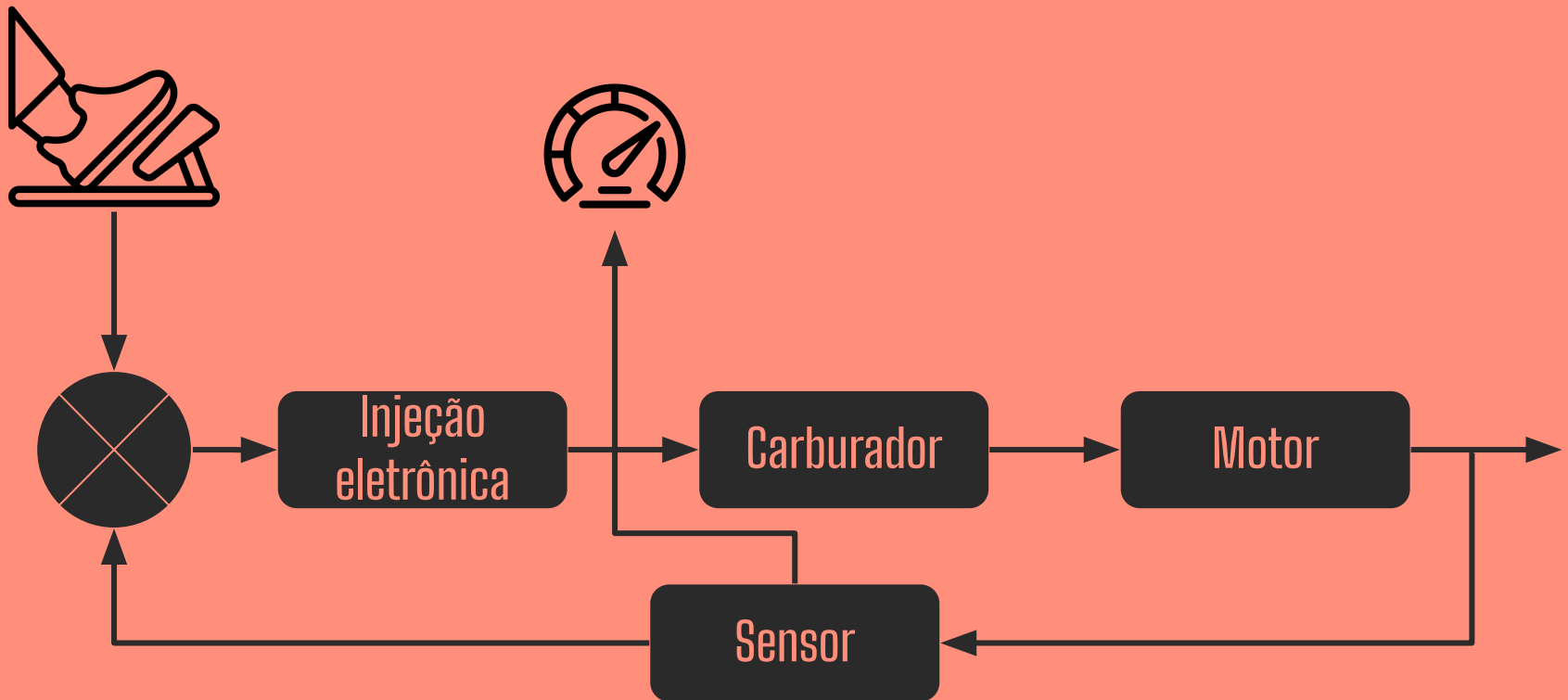
São múltiplos componentes e a observação deve ser capaz de entender o que cada um deles está fazendo no momento. **Pelo resultado, temos que saber quais os valores de entrada/saída em cada um dos componentes.**



Abrindo a caixa preta



Para isso, inserimos um **sensor** no sistema. Que **observa a variação no tempo em relação a cada entrada**



Visualização dos sensores principais



Aplica
ções

Trazendo isso para o
nosso mundo

A vida na programação



Quando vamos desenvolver uma aplicação, fazemos várias escolhas

- **Qual linguagem:** Python

A vida na programação



Quando vamos desenvolver uma aplicação, fazemos várias escolhas

- **Qual linguagem:** Python
- **Qual framework:** Flask, Django, FastAPI, Tornado, Sanic, ...

A vida na programação



Quando vamos desenvolver uma aplicação, fazemos várias escolhas

- **Qual linguagem:** Python
- **Qual framework:** Flask, Django, FastAPI, Tornado, Sanic, ...
- **Qual banco de dados:** Postgres, MariaDB, Mongo, ...

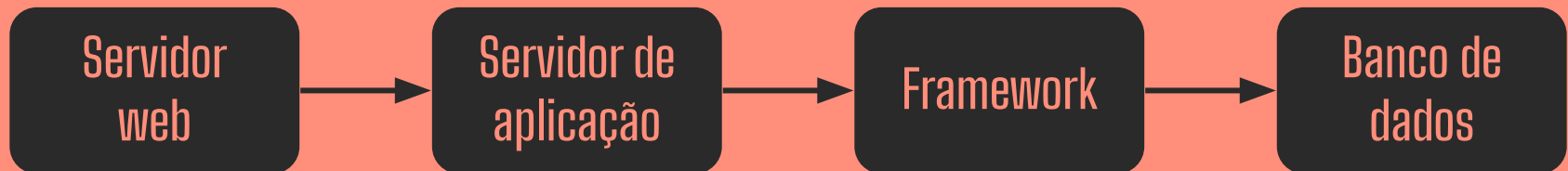
A vida na programação



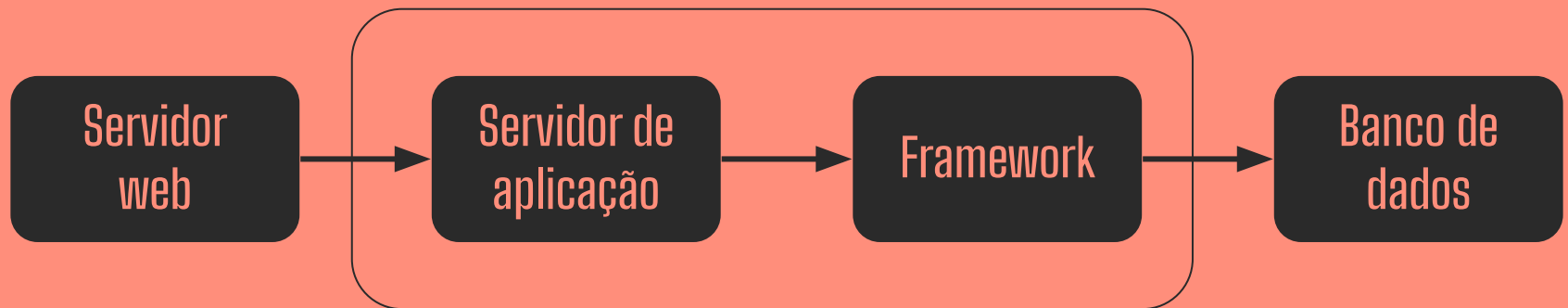
Quando vamos desenvolver uma aplicação, fazemos várias escolhas

- **Qual linguagem:** Python
- **Qual framework:** Flask, Django, FastAPI, Tornado, Sanic, ...
- **Qual banco de dados:** Postgres, MariaDB, Mongo, ...
- **Qual servidor de aplicação:** Gunicorn, Uvicorn, Hypercorn, ...
- **Qual servidor web:** Apache, Nginx, Traefik, ...
- **Qual a tecnologia de containers:** Docker, k8s, Podman, ...
- **Qual plataforma de deploy:** AWS, DO, Heroku, Fly, ...

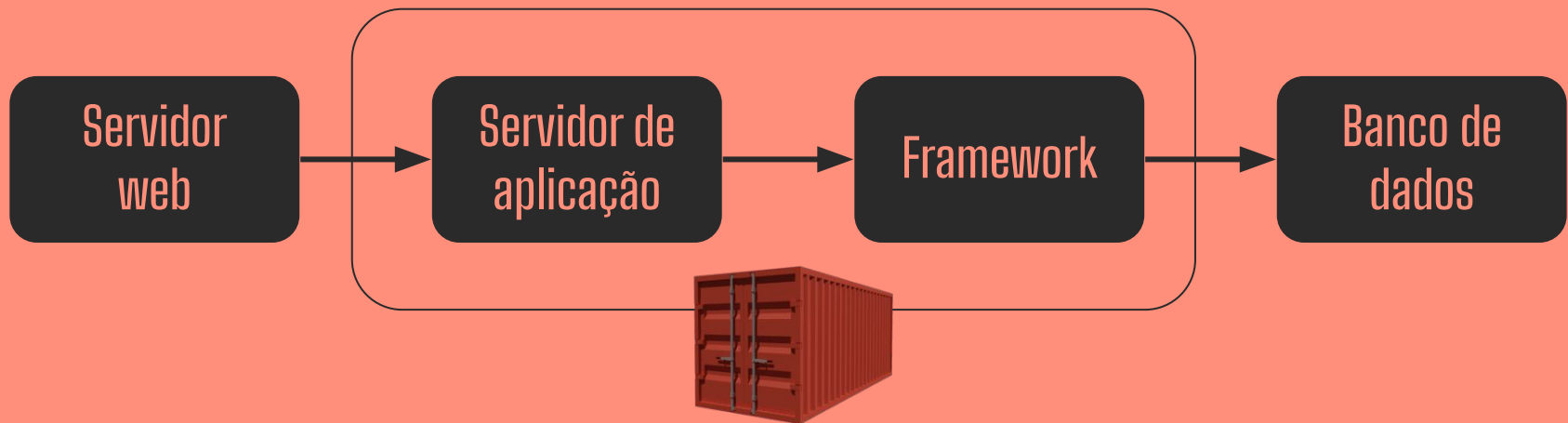
Montamos uma caixa complexa do sistema



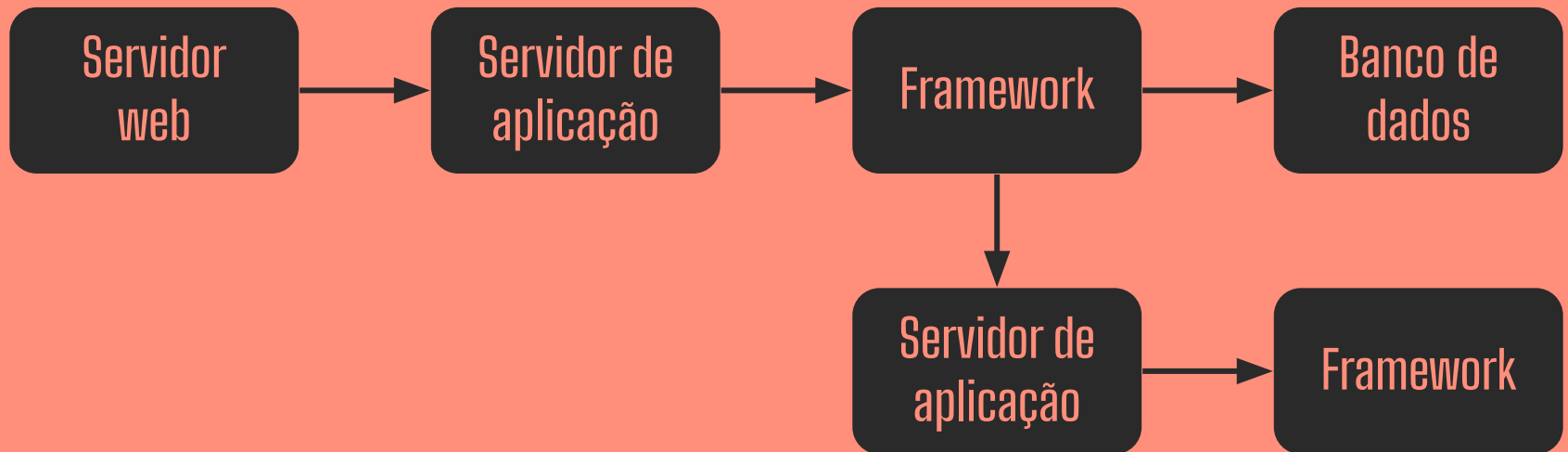
Montamos uma caixa complexa do sistema



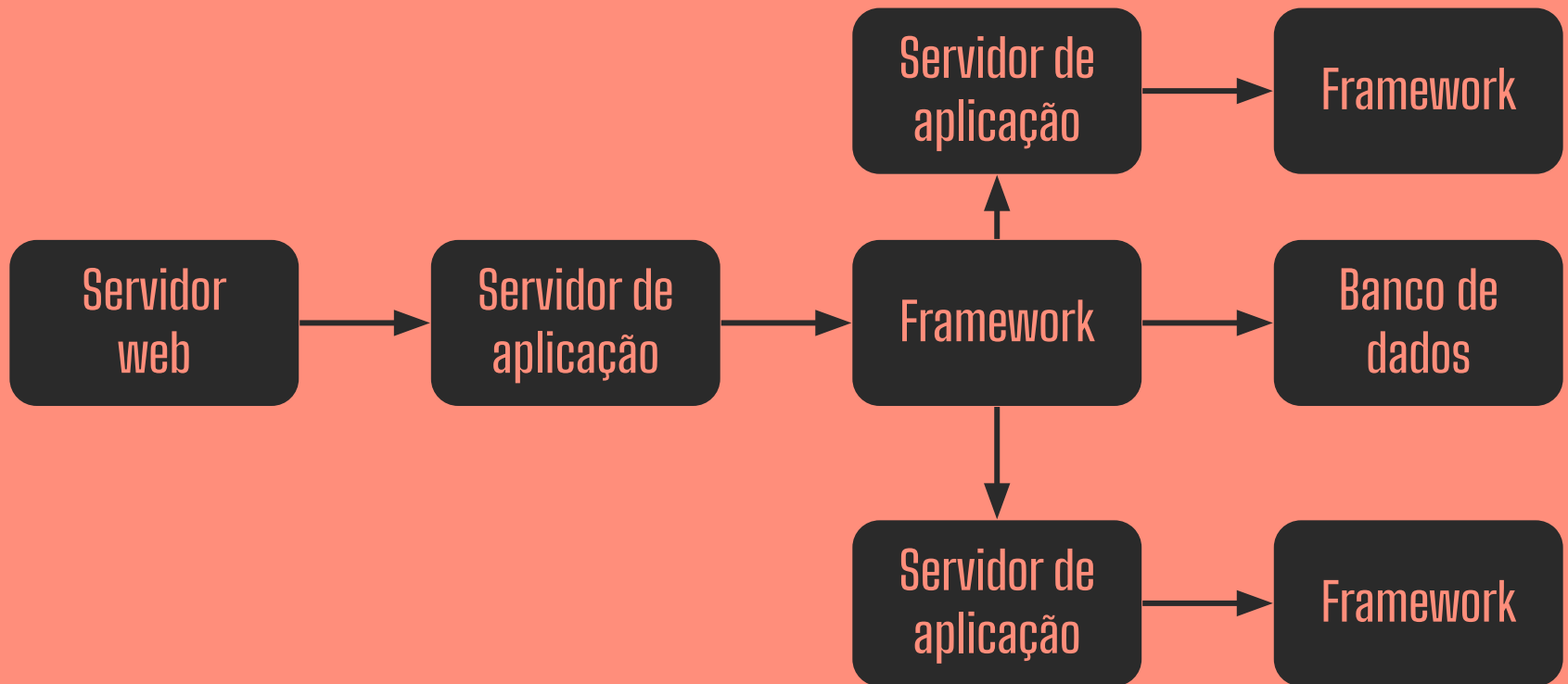
Montamos uma caixa complexa do sistema



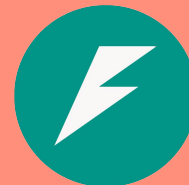
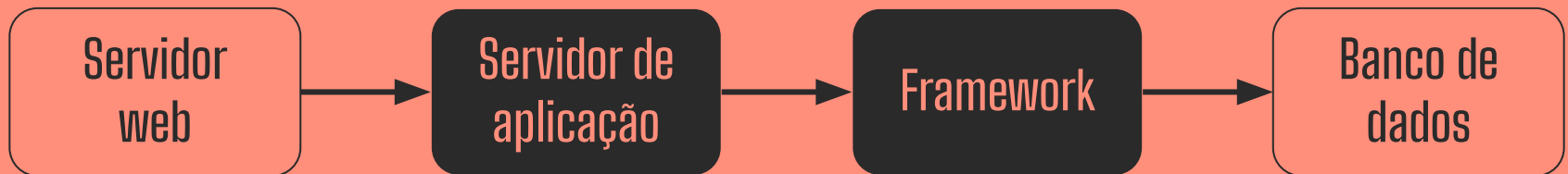
Montamos uma caixa complexa do sistema



Montamos uma caixa complexa do sistema



Exemplificando com a prática!



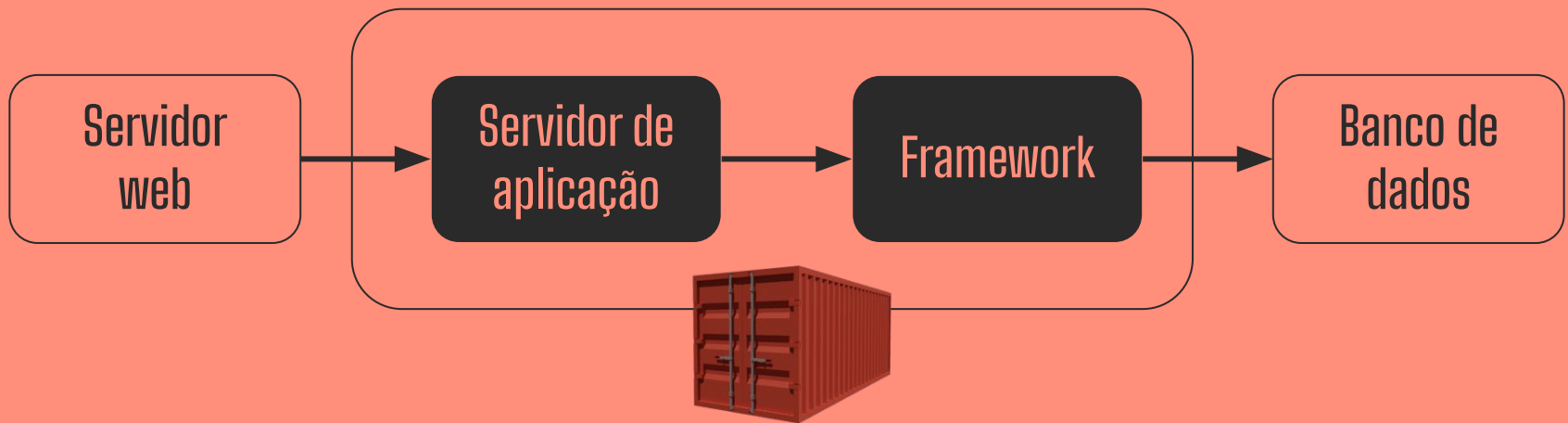
Um olá mundo!

```
pip install fastapi uvicorn
```

```
1  from fastapi import FastAPI
2
3  app = FastAPI()
4
5  @app.get('/')
6  def home():
7      return {'message': 'Olá mundo!'}
```

```
uvicorn app.app:app --root 0.0.0.0 --port 8000
```

Montamos uma caixa complexa do sistema



Um ambiente passível de reprodução

```
FROM python:3.12.2-slim
ENV POETRY_VIRTUALENVS_CREATE=false
```

```
WORKDIR app/
```

```
RUN pip install poetry
```

```
COPY pyproject.toml poetry.lock .
```

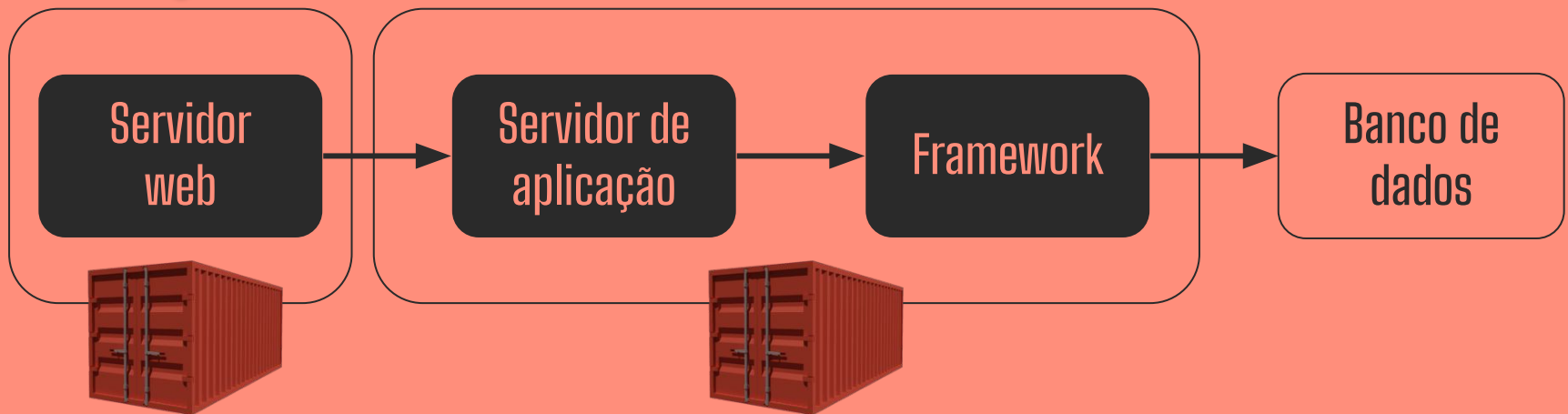
```
RUN poetry install
```

```
COPY . .
```

```
CMD uvicorn spam.app:app \
  --port 8000 \
  --host 0.0.0.0
```

```
services:
  spam:
    build: "spam/."
    ports:
      - 8000:8000
```

Montamos uma caixa complexa do sistema





```
1  worker_processes 4;
2
3  events { worker_connections 512; }
4
5  http {
6      server {
7          listen 80;
8          listen [::]:80;
9          location / {
10             proxy_pass http://spam:8001;
11         }
12     }
13 }
```

```
1  worker_processes 4;
2
3  events { worker_connections 512; }
4
5  http {
6      server {
7          listen 80;
8          listen [::]:80;
9          location / {
10             proxy_pass
11         }
12     }
13 }
```

```
1  services:
2      spam:
3          build: "spam/."
4          ports:
5              - 8001:8001
6
7      nginx:
8          image: nginx
9          volumes:
10             - './etc/nginx.conf:/etc/nginx/nginx.conf'
11          ports:
12             - 80:80
13          depends_on:
14             - spam
```

Deploy!
take 1

Com isso algumas perguntas surgem



- A aplicação está funcionando?
- Quantas requisições a aplicação recebeu hoje? Ontem? Na última semana?
- Qual a média de pessoas usando a aplicação por vez?
- Qual o horário em que as pessoas mais usam?

Algumas outras são críticas



- A aplicação de erro em algum momento?
- Existe uma latência muito grande na aplicação?
- Quanto tempo demora em média para fazer X?
- Existe algum usuário que consome mais recursos?

Domingo às 8 da
noite!



Até que então!



< Chats

Pessoa Amada

online

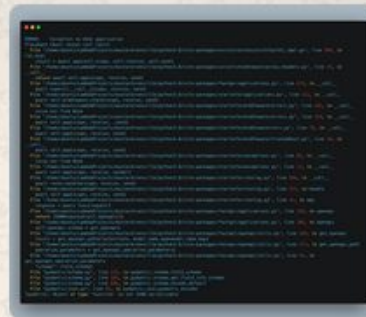


Opa, precisamos de uma ajuda, está por aí?

Domingo, 10 Mar 2024 20:53 ✓✓

O que aconteceu?

Domingo, 10 Mar 2024 21:15



Temos esse erro aqui em produção

Domingo, 10 Mar 2024 21:15 ✓✓

Pode entrar em uma chamada de vídeo agora?

Domingo, 10 Mar 2024 21:15

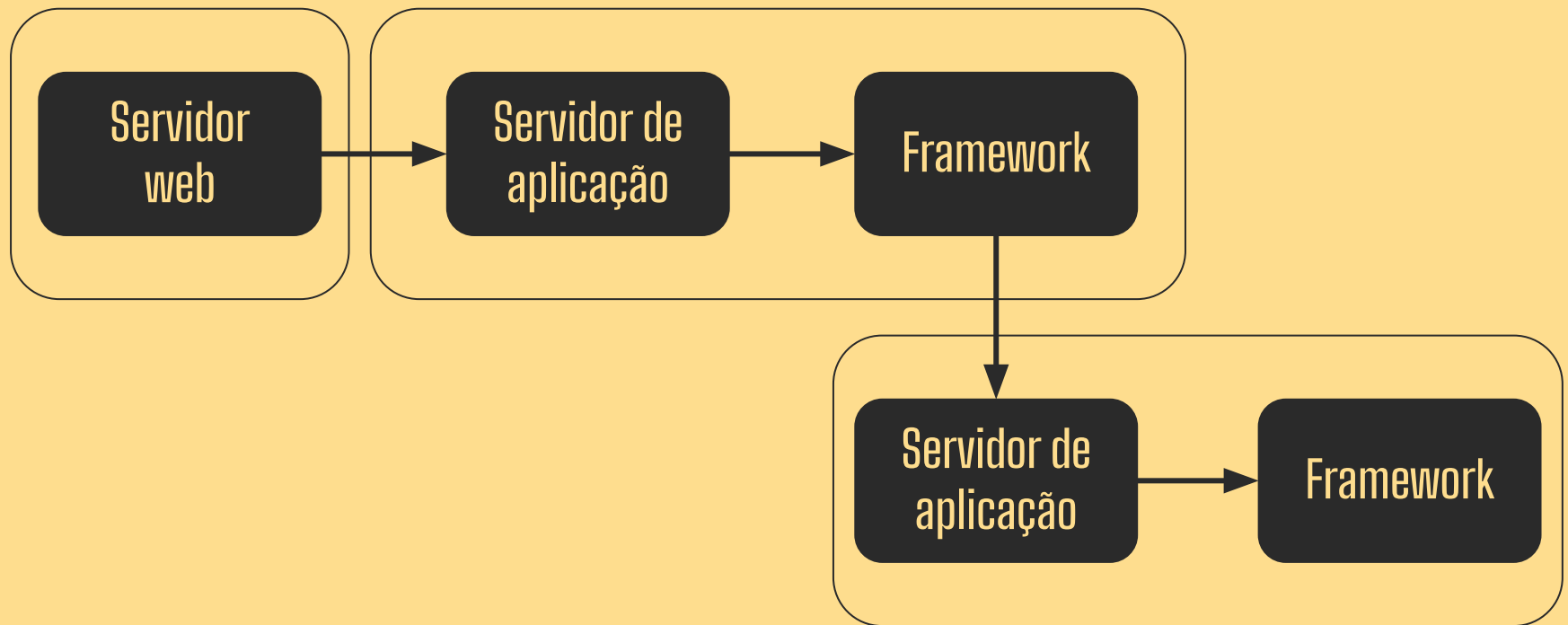


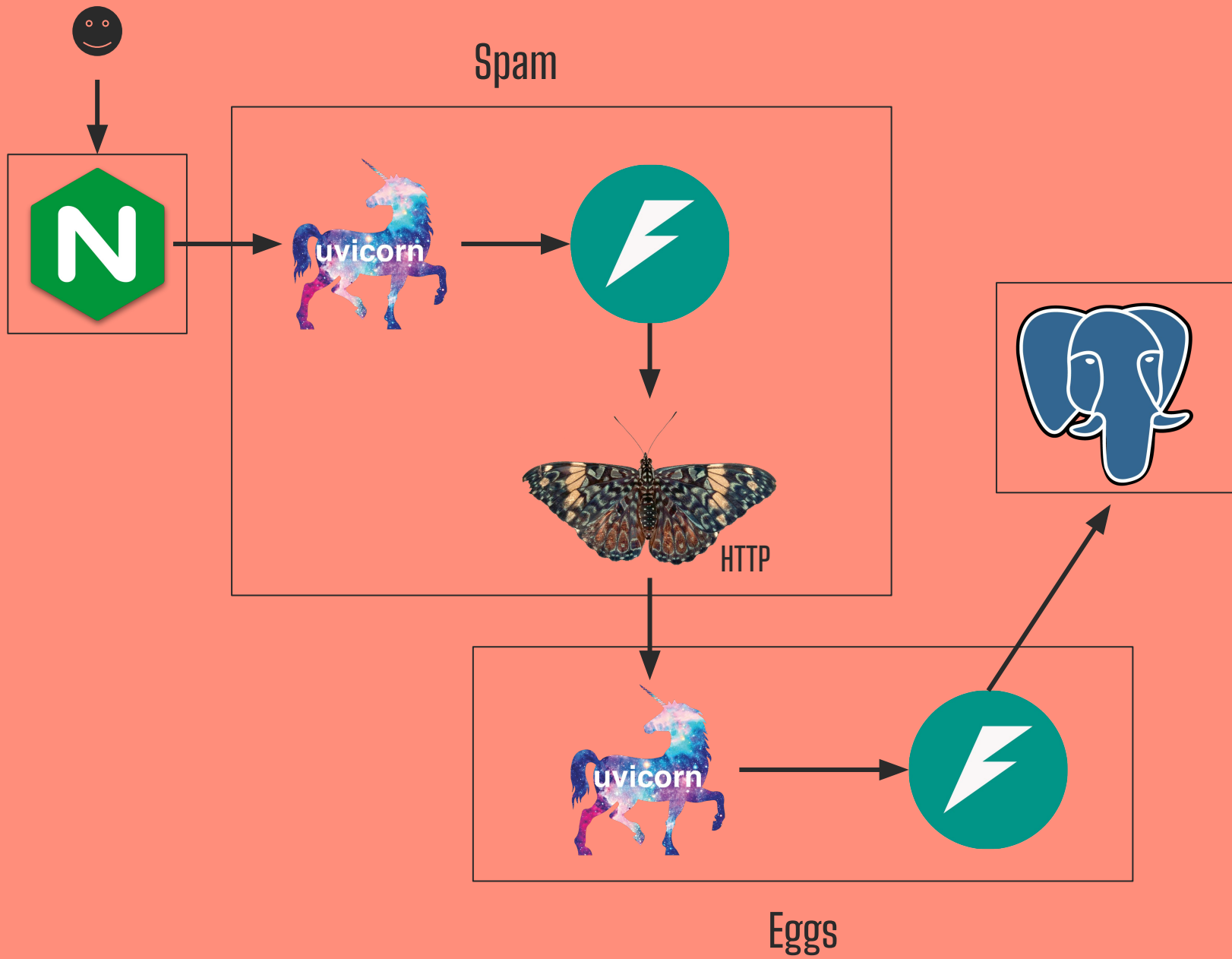
Message





Exponenciando o problema!





Mostrar o código!
take 2

Deploy!
take 2

Agora as perguntas dobram



- O quanto a comunicação entre os serviços falha?
- O quanto demora **a mais** por conta da dupla latência de rede?
- Qual das aplicações falha mais?
- Existe uma medida de contingência se *eggs* não responder?
- Quantas requisições de *spam* de fato chegam em *eggs*?
- Quantas falhas acontecem em *spam*?
 - Quais e quantas delas são relacionadas a *eggs*?
- Existe algo que evidencie quando o erro é referente a *eggs* ou a *spam*?

Como posso entender o que acontece na minha aplicação enquanto isso acontece?



A pergunta que fica



A arte de sinalizar o
que acontece
enquanto acontece

Telem
etria

Telemetria



Telemetria é uma forma de gerar **sinais locais da aplicação em tempo de execução** e enviar esses sinais para um lugar centralizado em formato de **eventos** da aplicação.

Ou seja,

A aplicação deve ser feita de uma forma em que se permita ser observada por meio de dados de telemetria.

Eventos e sinais



Os **sinais** são extraídos da aplicação em tempo de execução. Como se estivéssemos "tirando uma foto" do momento atual da aplicação. Uma coisa que explicará seu comportamento no tempo.

Um **evento** pode ser qualquer ação específica da aplicação:

- Spam requisitou Eggs
- Algo falhou
- Uma busca no banco de dados

Um evento gera um sinal de telemetria.

Tipos de sinais



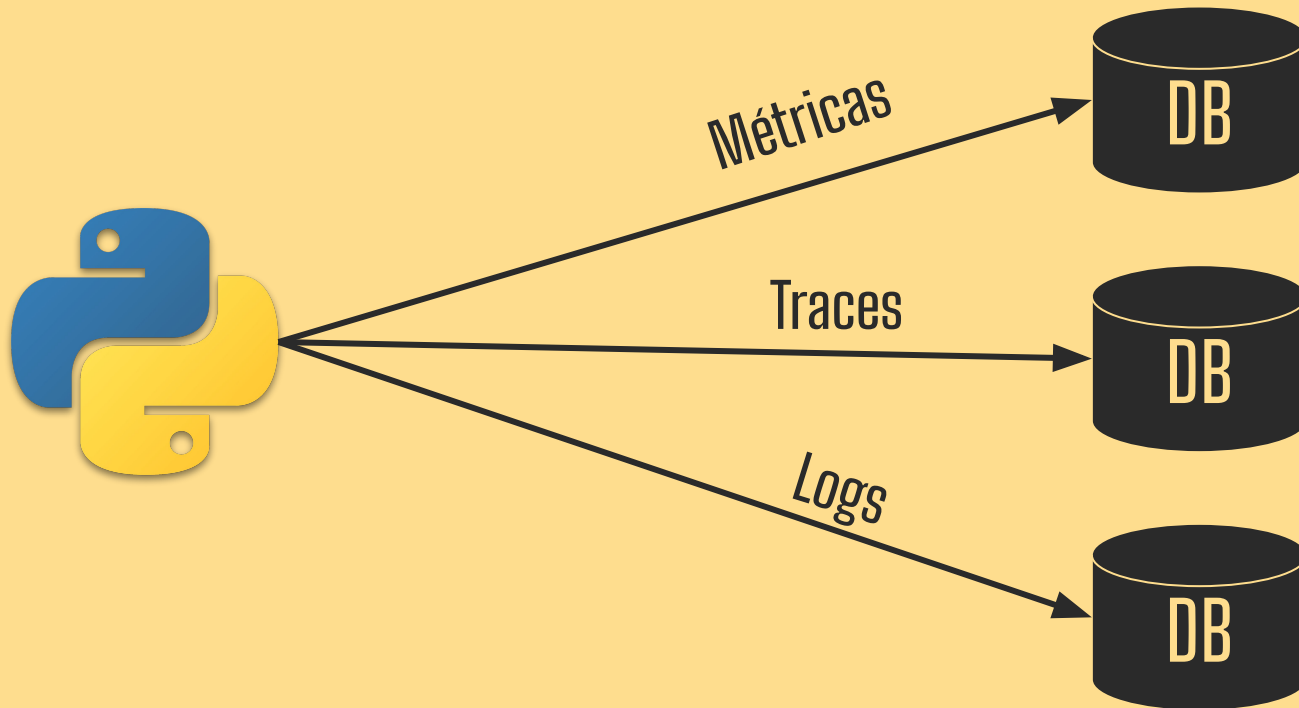
Os sinais mais comuns são:

- **Logs:** Um registro de uma operação
 - Um erro aconteceu no arquivo X, as Y horas devido a Z
- **Rastros [Traces]:** O caminho que uma requisição percorre dentro da aplicação e seu tempo:
 - Nginx -> Endpoint / -> Função XPTO -> Requisição para *Eggs* -> Que faz uma busca X no banco de dados
- **Métricas:** Medidas capturadas em tempo de execução
 - Quantas requisições recebi hoje?
 - Quantidade de usuários ativos agora
 - Quantos erros 500 geramos?

Armazenamento de sinais



Os sinais gerados pelas aplicações devem ser armazenados para consulta posterior, para ser feita uma análise detalhada posteriormente.



Serviços pagos



Caso te preocupe a sensação de subir e manter mais 3 bancos de dados, existem alternativas pagas para instrumentar e armazenar os dados



Instrumentação



Quando queremos que a nossa aplicação gere sinais de telemetria, precisamos **instrumentar** a aplicação. Fornecer instrumentos para que os sinais sejam gerados e enviados para os bancos de dados.



Bancos de dados de Telemetria



O problema é que existe uma abundância de bancos de dados de telemetria

Logs

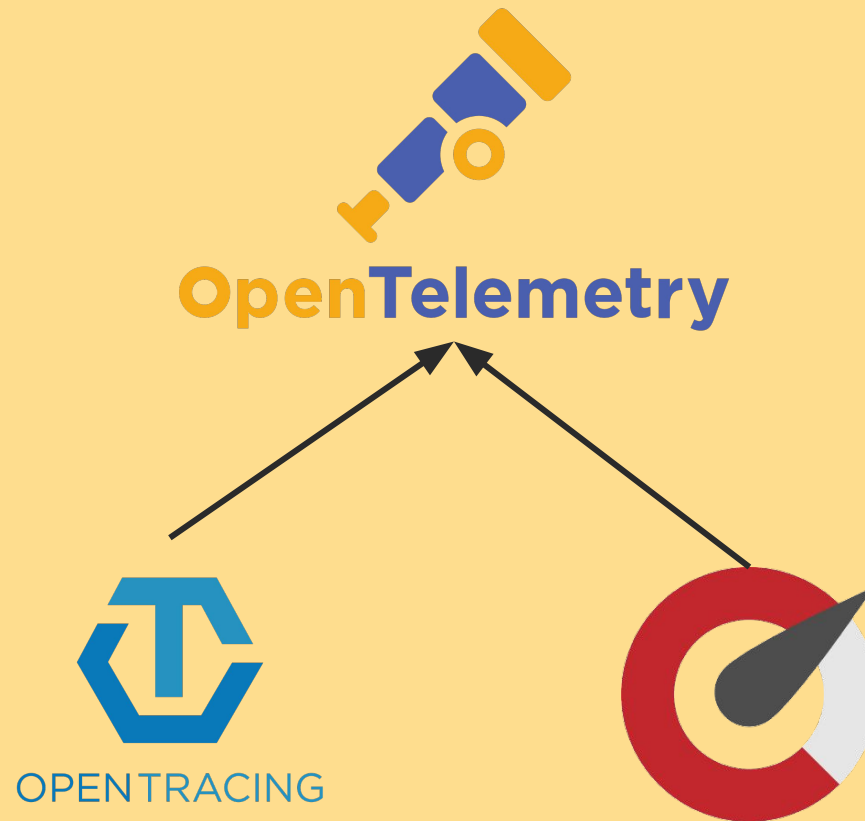


Métricas



Tracing





A ideia por trás do OTel



- **API:** Especificação universal para telemetria
- **SDK:** Implementação para a API gerenciada pelo APP
- **Exporter:** Uma forma de exportar sinais pelo aplicativo pela API
- **Sinais:**
 - **Tracer**
 - Propagação de contexto
 - **Meter**
 - **Logs**
 - **Profiling:** Em breve
- **Collector:** Uma aplicação a parte para coletar telemetria das aplicações. Um centralizador dos dados de telemetria e um exportador para diversos bancos de dados.

Language SDKs

For the development status, or maturity level, of a [language SDK](#), see the following table:

Language	Traces	Metrics	Logs
C++	Stable	Stable	Stable
C#/.NET	Stable	Stable	Stable
Erlang/Elixir	Stable	Experimental	Experimental
Go	Stable	Stable	In development
Java	Stable	Stable	Stable
JavaScript	Stable	Stable	Experimental
PHP	Stable	Stable	Stable
Python	Stable	Stable	Experimental
Ruby	Stable	In development	In development
Rust	Beta	Alpha	Alpha
Swift	Stable	Experimental	In development

<https://opentelemetry.io/status/>

Deploy!
take 3
(com telemetria)

0 motivo de
estarmos aqui!

Obser
vabilidad
ade

Observabilidade



Observabilidade é a arte de **espremer dados de telemetria**, **somar** mais alguns **dados externos** (CI, Updates, Negócio, Configuração, ...) e **responder novas perguntas** sobre a aplicação **que ainda não foram feitas**.

Poder olhar para os bancos de dados e enxergar raízes de problemas, priorizar novas tarefas, saber sobre os problemas em tempo real, estudar os dados e tomar decisões.



apoia.se/livedepython



pix.dunossauro@gmail.com



patreon.com/dunossauro



Ajude o projeto <3



Ademar Peixoto, Adilson Herculano, Alemao, Alexandre Harano, Alexandre Lima, Alexandre Takahashi, Alexandre Villares, Alfredo Braga, Alisson Souza, Alysson Oliveira, Andre Azevedo, Andre Mesquita, Andre Paula, Aniltair Filho, Antônio Filho, Arnaldo Turque, Aslay Clevisson, Aurelio Costa, Bárbara Grillo, Ber_dev_2099, Bernardo Fontes, Bruno Almeida, Bruno Barcellos, Bruno Batista, Bruno Freitas, Bruno Ramos, Caio Nascimento, Carlos Ramos, Cristian Firmino, Daniel Bianchi, Daniel Freitas, Daniel Wojcickoski, Danilo Boas, Danilo Silva, David Couto, David Kwast, Davi Souza, Dead Milkman, Denis Bernardo, Dgeison Peixoto, Diego Guimarães, Dino, Edgar, Edilson Ribeiro, Emerson Rafael, Ennio Ferreira, Erick Andrade, Érico Andrei, Everton Silva, Fabio Barros, Fábio Barros, Fabio Valente, Fabricio Biazotto, Felipe Augusto, Felipe Rodrigues, Fernanda Prado, Fernando Celmer, Flávio Meira, Francisco Silvério, Frederico Damian, Gabriel Espindola, Gabriel Mizuno, Gabriel Moreira, Gabriel Paiva, Gabriel Ramos, Gabriel Simonetto, Giovanna Teodoro, Giuliano Silva, Guilherme Beira, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Guilherme Silva, Gustavo Suto, Haelmo Almeida, Harold Gautschi, Heitor Fernandes, Helvio Rezende, Henrique Andrade, Henrique Sebastião, Hiago Couto, Higor Monteiro, Igor Taconi, Janael Pinheiro, Jean Victor, Jefferson Antunes, Joelson Sartori, Jonatas Leon, Jônatas Oliveira, Jônatas Silva, Jorge Silva, Jose Barroso, Jose Edmario, José Gomes, Joseíto Júnior, Jose Mazolini, Josir Gomes, Juan Felipe, Juliana Machado, Julio Batista-silva, Julio Franco, Júlio Gazeta, Júlio Sanchez, Kaio Peixoto, Leandro Silva, Leandro Vieira, Lengo, Leonan Ferreira, Leonardo Mello, Leonardo Nazareth, Lucas Carderelli, Lucas Lattari, Lucas Mello, Lucas Mendes, Lucas Nascimento, Lucas Schneider, Lucas Simon, Luciano Filho, Luciano Ratamero, Luciano Teixeira, Luiz Carlos, Luiz Duarte, Luiz Lima, Luiz Paula, Mackilem Laan, Marcelo Araujo, Marcio Silva, Marco Mello, Marcos Gomes, Marina Passos, Mateus Lisboa, Matheus Silva, Matheus Vian, Mirian Batista, Mlevi Lsantos, Murilo Carvalho, Murilo Viana, Nathan Branco, Ocimar Zolin, Otávio Carneiro, Pedro Henrique, Pedro Pereira, Peterson Santos, Philipe Vasconcellos, Pytonyc, Rafael Araújo, Rafael Faccio, Rafael Romão, Raimundo Ramos, Ramayana Menezes, Renato José, Renato Moraes, Rene Pessoto, Renne Rocha, Ricardo Combat, Ricardo Silva, Rinaldo Magalhaes, Riverfount, Rjribeiro, Rodrigo Barretos, Rodrigo Oliveira, Rodrigo Quiles, Rodrigo Santana, Rodrigo Vaccari, Rodrigo Vieira, Rogério Nogueira, Rui Jr, Samanta Cicilia, Santhiago Cristiano, Selmison Miranda, Téó Calvo, Thiago Araujo, Thiago Borges, Tiago Henrique, Tony Dias, Tyrone Damasceno, Valdir, Valdir Tegon, Varlei Menconi, Vcwild, Vinícius Costa, Vinicius Stein, Vladimir Lemos, Washington Teixeira, Willian Lopes, Willik Araujo, Wilson Duarte, Zeca Figueiredo



Obrigado você



Chequem os links de
referências na descrição



Por favor!

