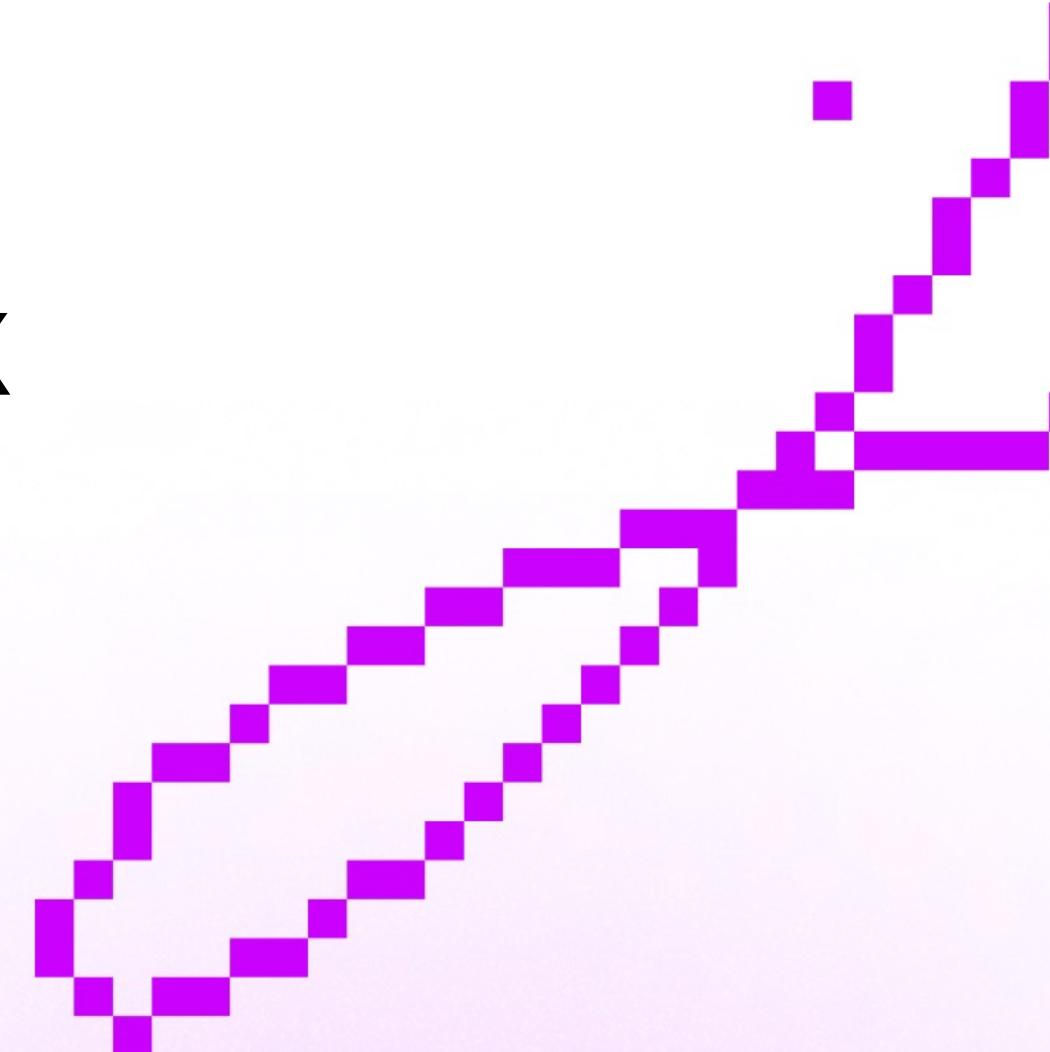


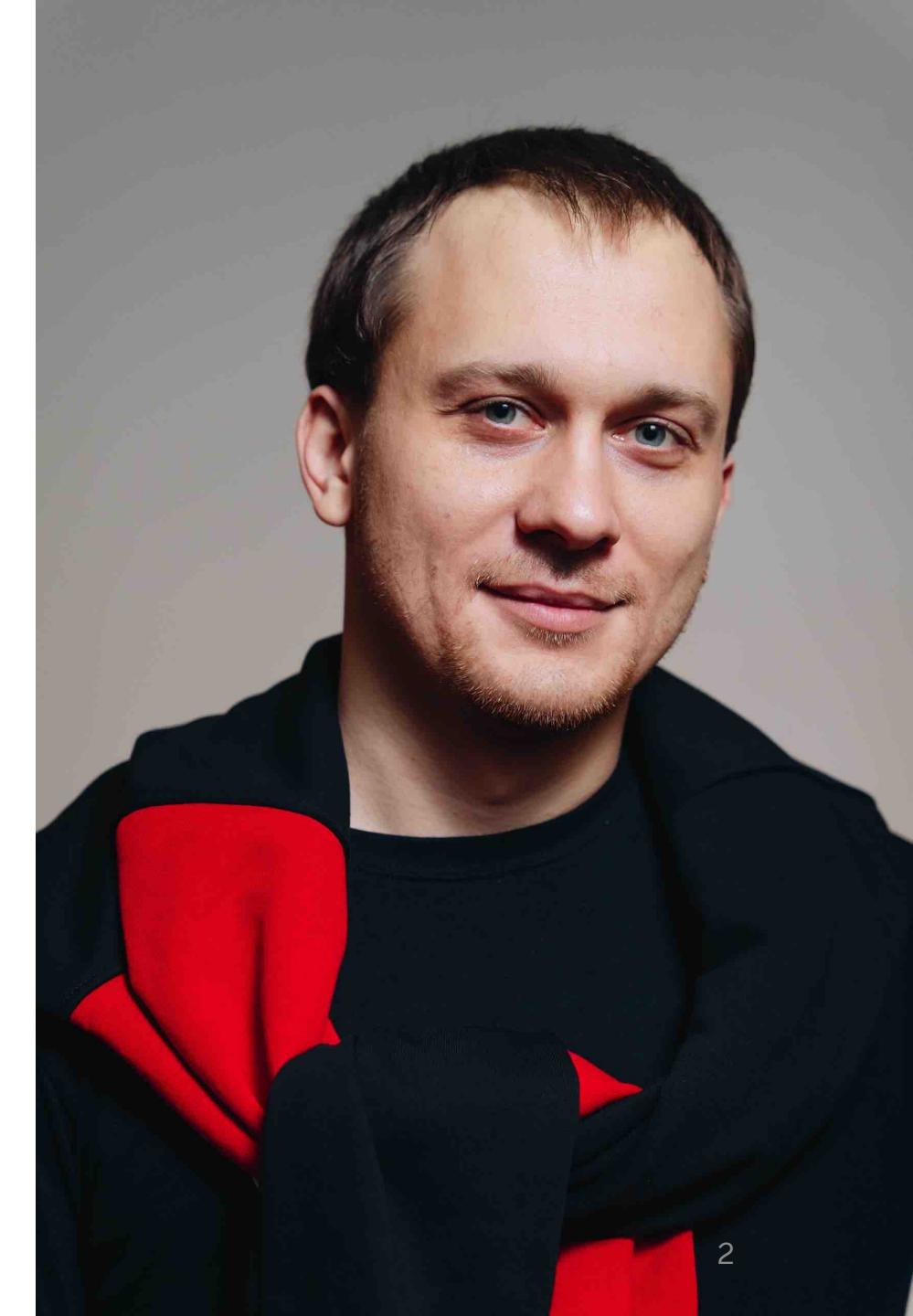
# **Observability распределённых систем**



**Контур**

Григорий Кошелев

# Григорий Кошелев

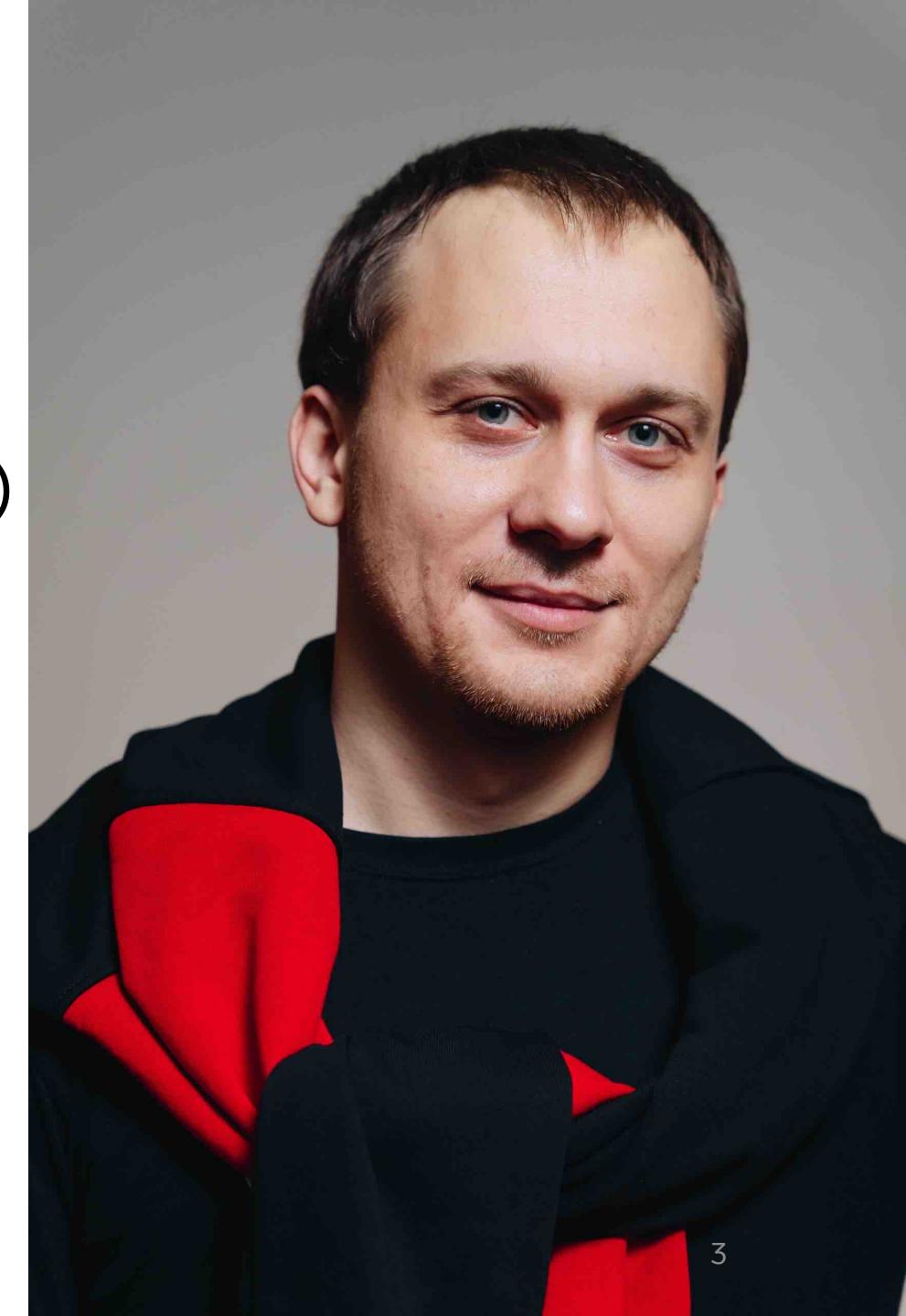


# Григорий Кошелев

— Веду Telegram-канал про технологии  
(~10 постов с тегом #телеометрия, #sre)



[https://t.me/chnl\\_GregoryKoshelev](https://t.me/chnl_GregoryKoshelev)

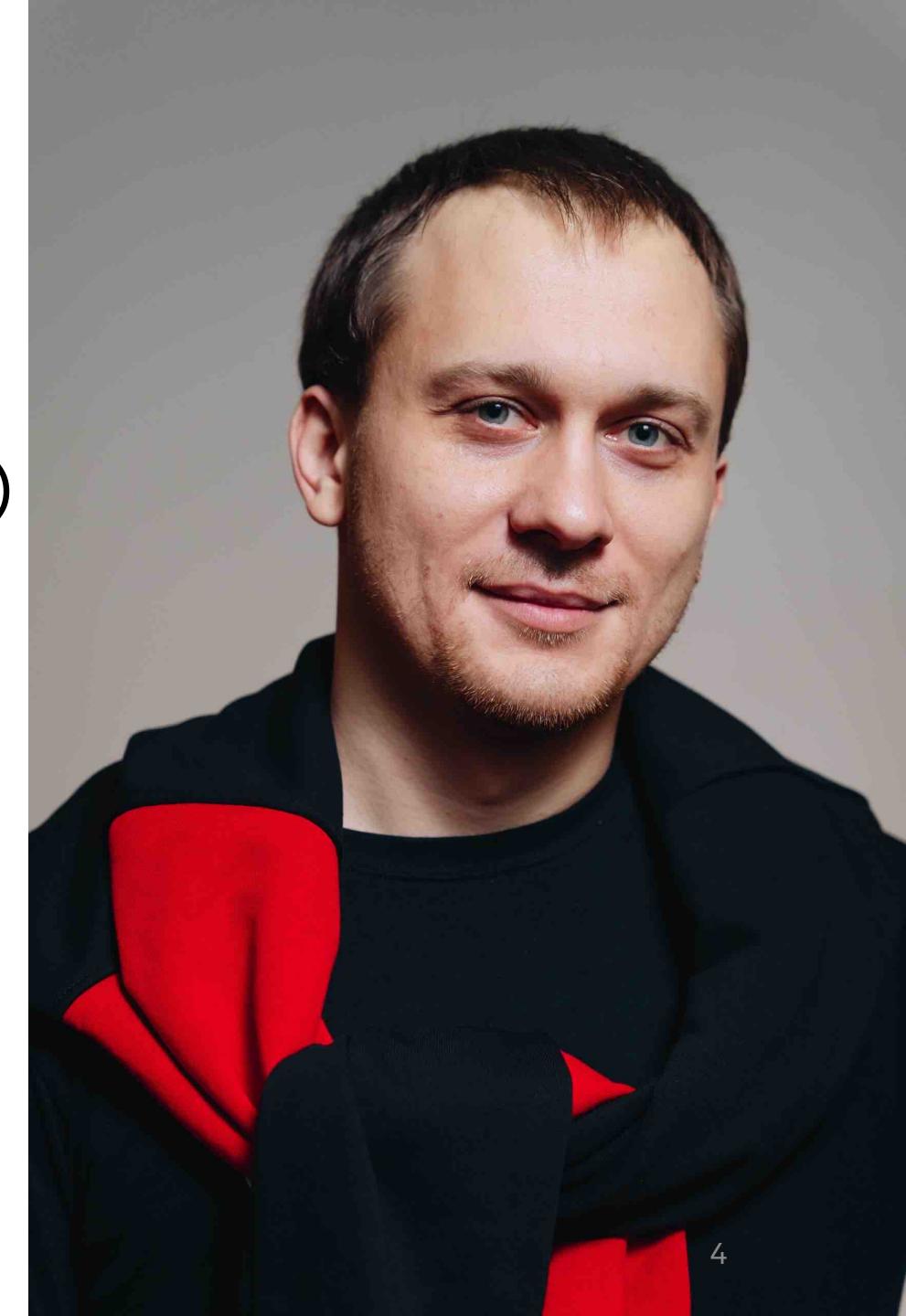


# Григорий Кошелев

- Веду Telegram-канал про технологии (~10 постов с тегом #телеометрия, #sre)
- 4 года руководил разработкой платформы для обработки, хранения и передачи данных телеметрии



[https://t.me/chnl\\_GregoryKoshelev](https://t.me/chnl_GregoryKoshelev)

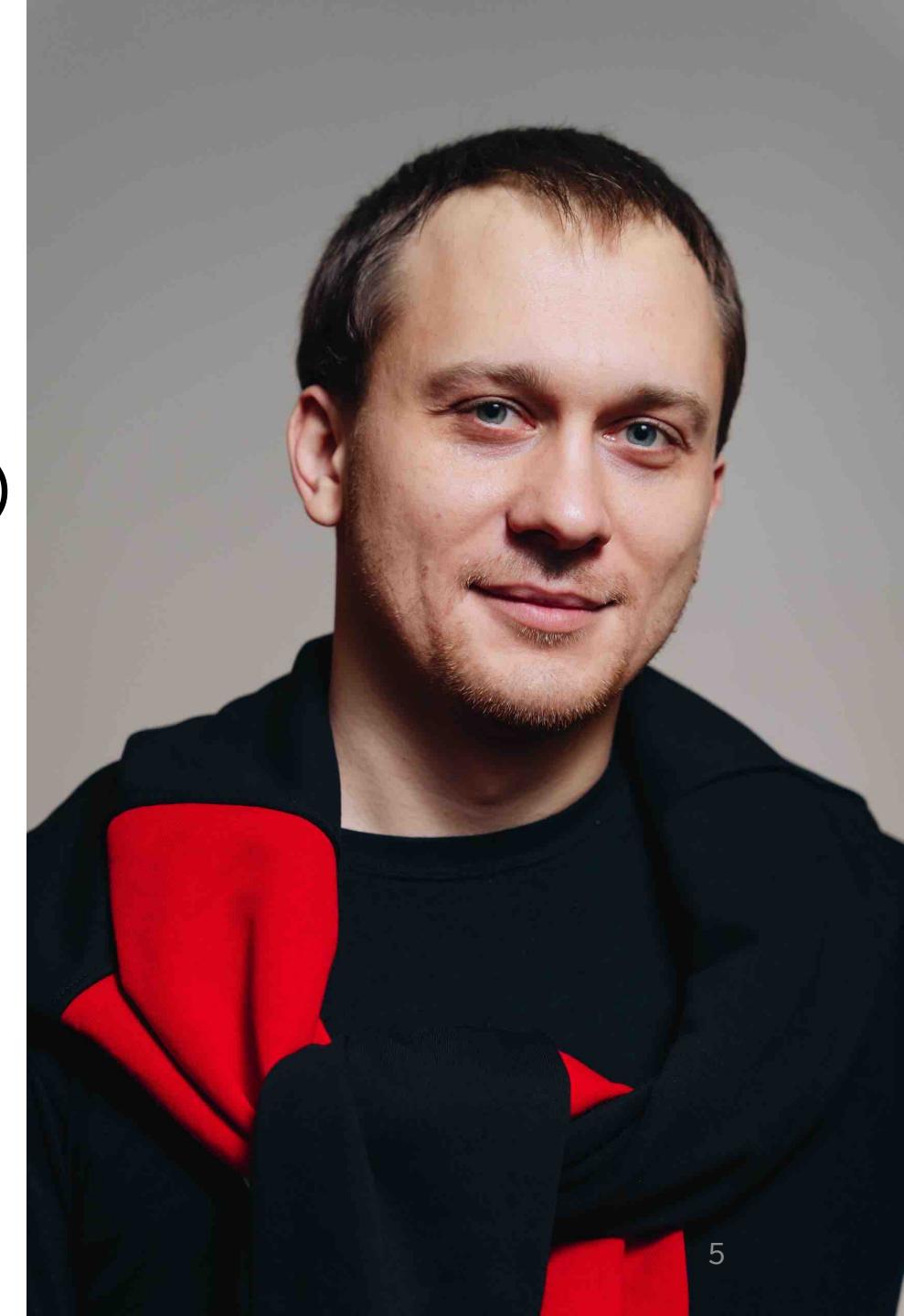


# Григорий Кошелев

- Веду Telegram-канал про технологии (~10 постов с тегом #телеометрия, #sre)
- 4 года руководил разработкой платформы для обработки, хранения и передачи данных телеметрии
  - Много докладов по теме Observability



[https://t.me/chnl\\_GregoryKoshelev](https://t.me/chnl_GregoryKoshelev)



# Observability распределённых систем

- Телеметрия и её виды
- Распределённые трассировки
- OpenTelemetry
- Создание метрик из трассировок
- Корреляция логов и метрик с трассировками
- Проектирование телеметрии
- Поиск root cause

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

# Распределённая система

# Распределённая система

- РС состоит из **нескольких** приложений

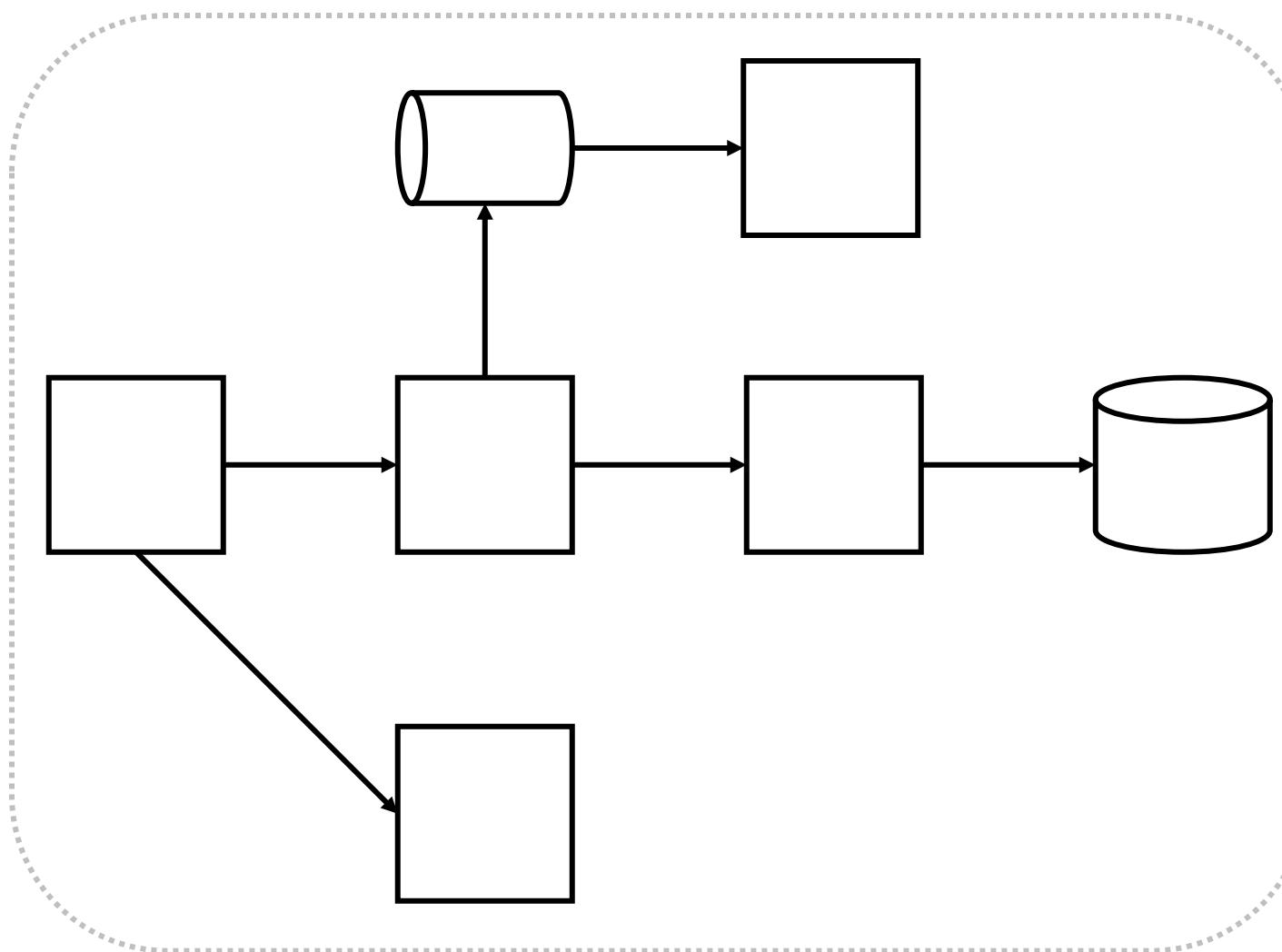
# Распределённая система

- РС состоит из **нескольких** приложений
- Приложения обмениваются **сообщениями**

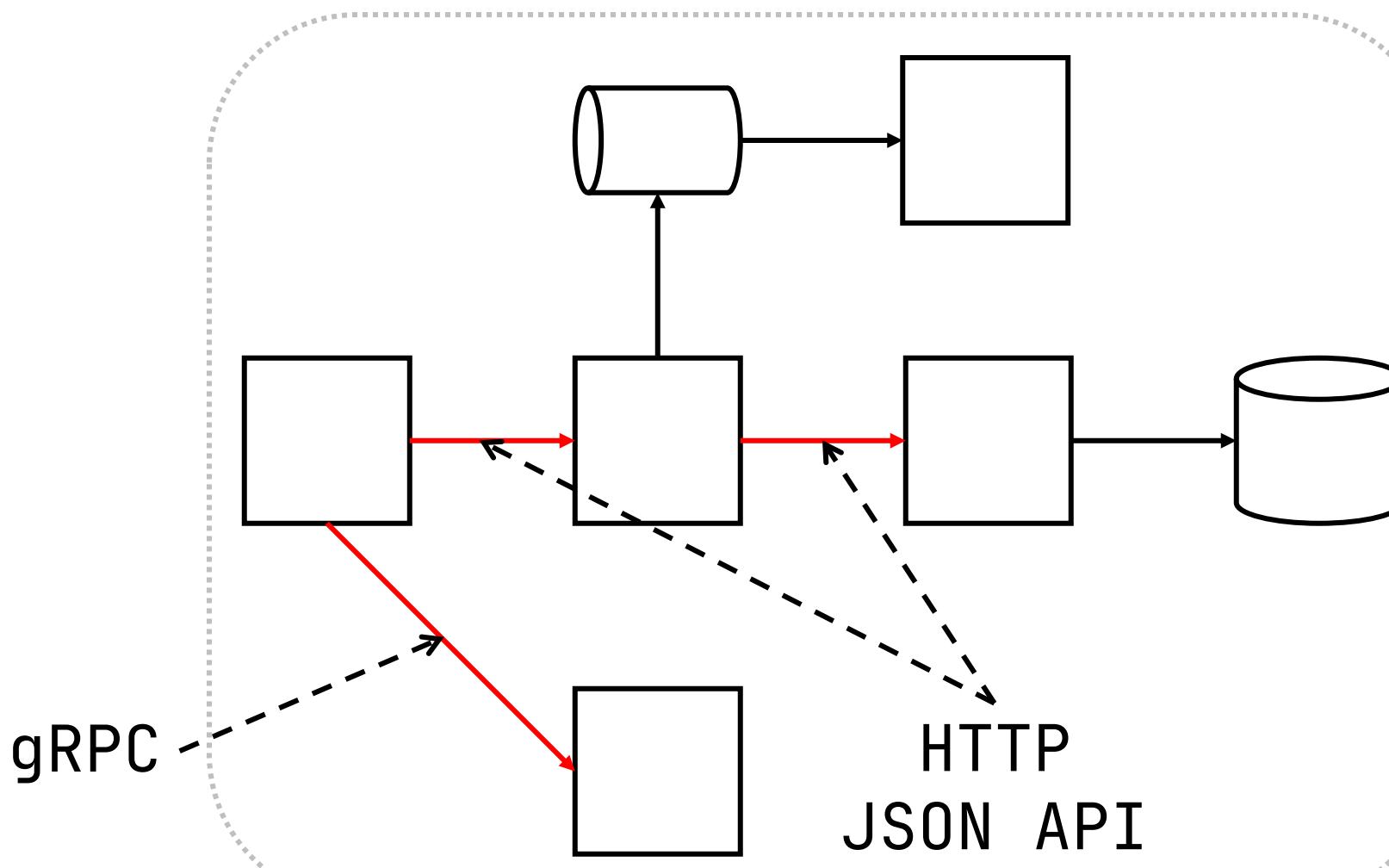
# Распределённая система

- РС состоит из **нескольких** приложений
- Приложения обмениваются **сообщениями**
- Приложения **могут** находиться на разных узлах

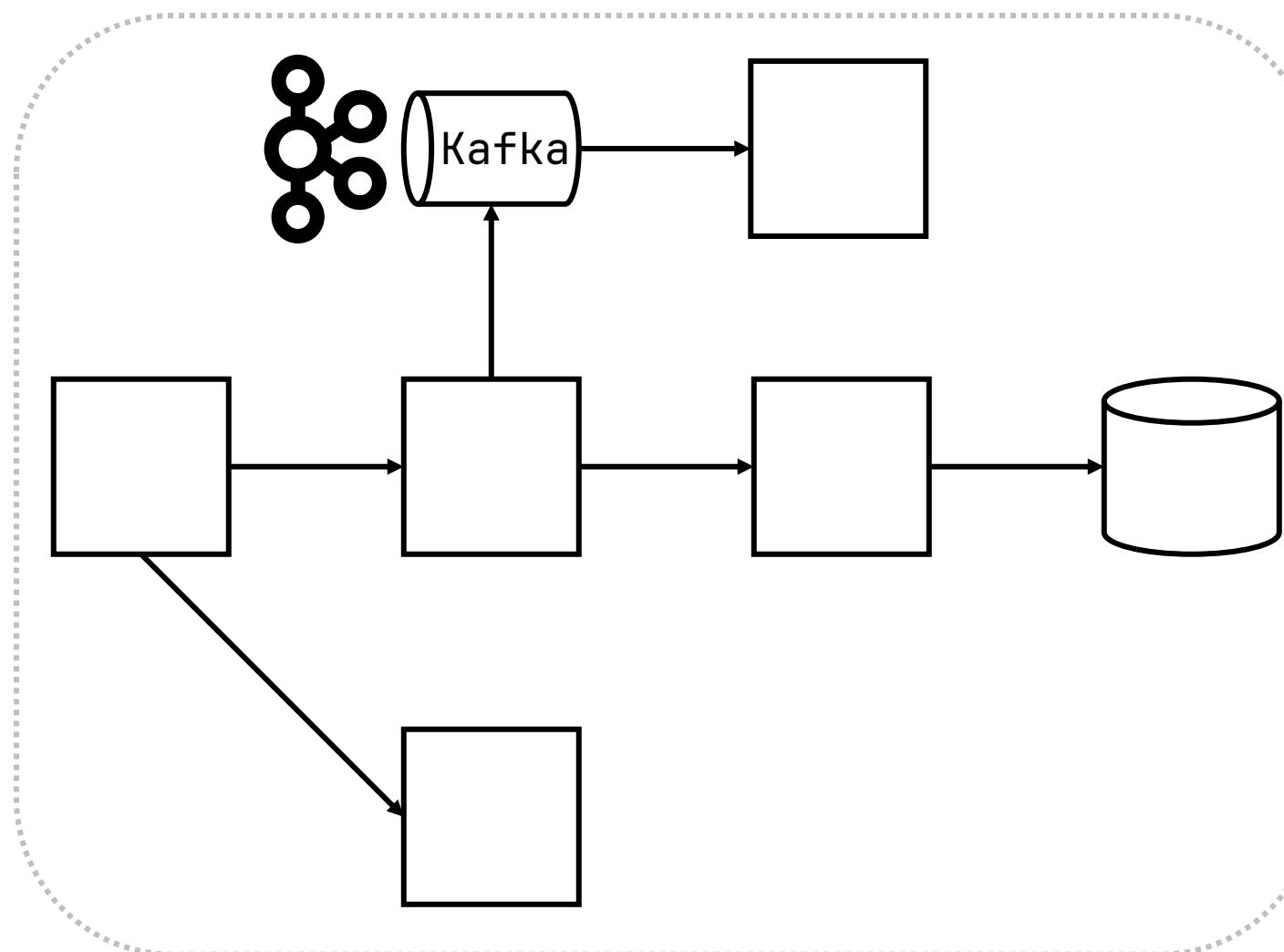
# Распределённая система



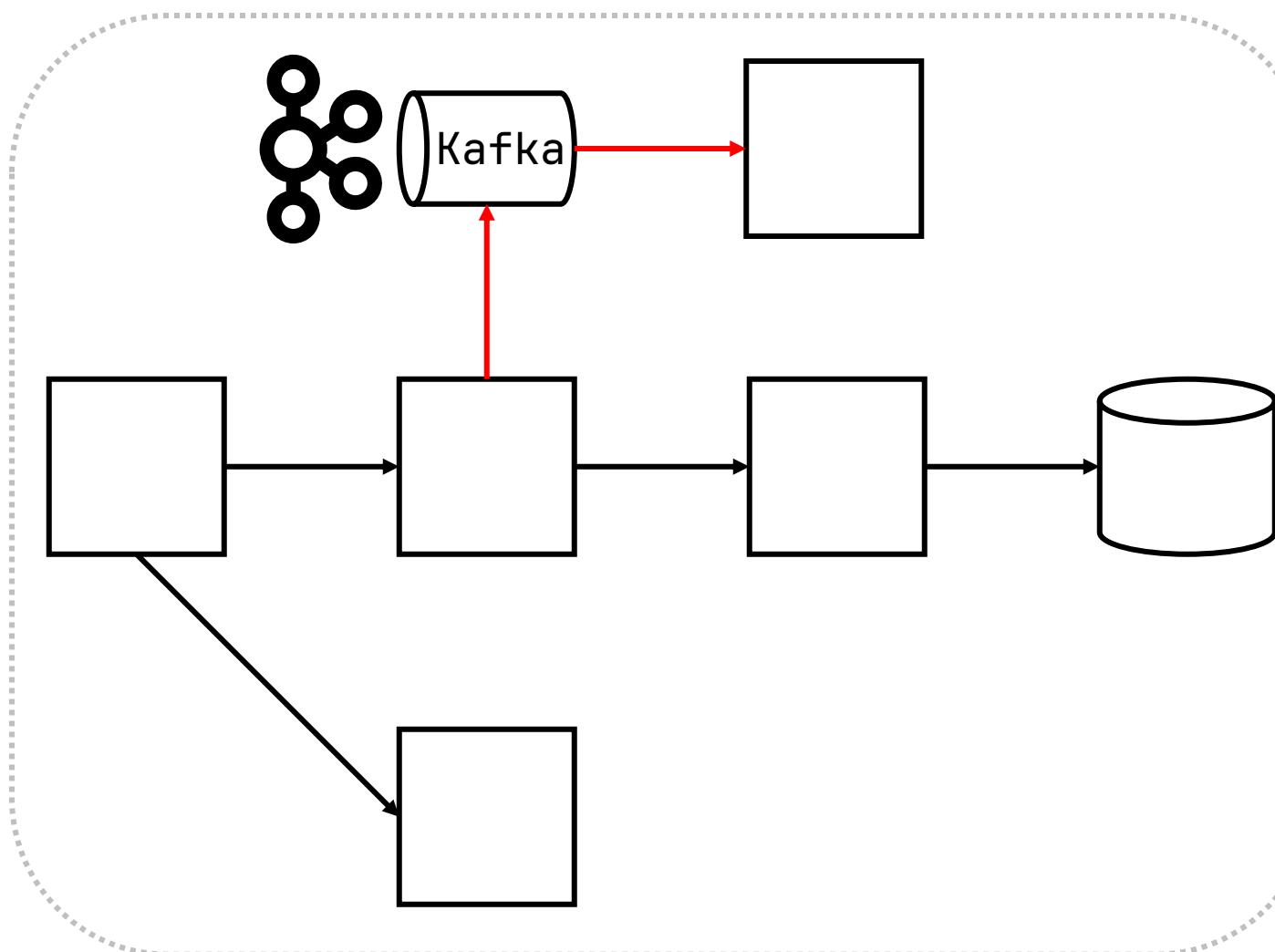
# Распределённая система



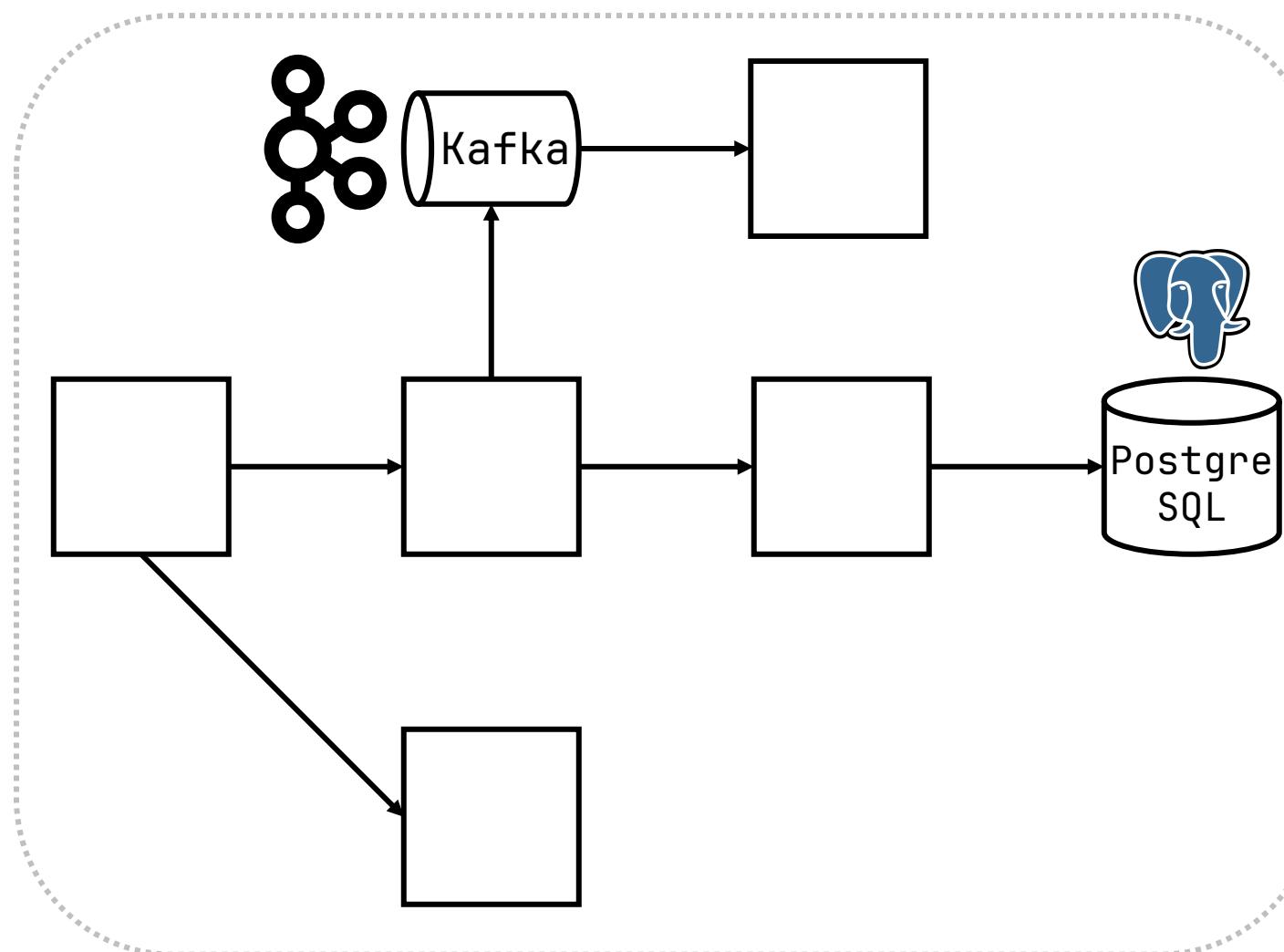
# Распределённая система



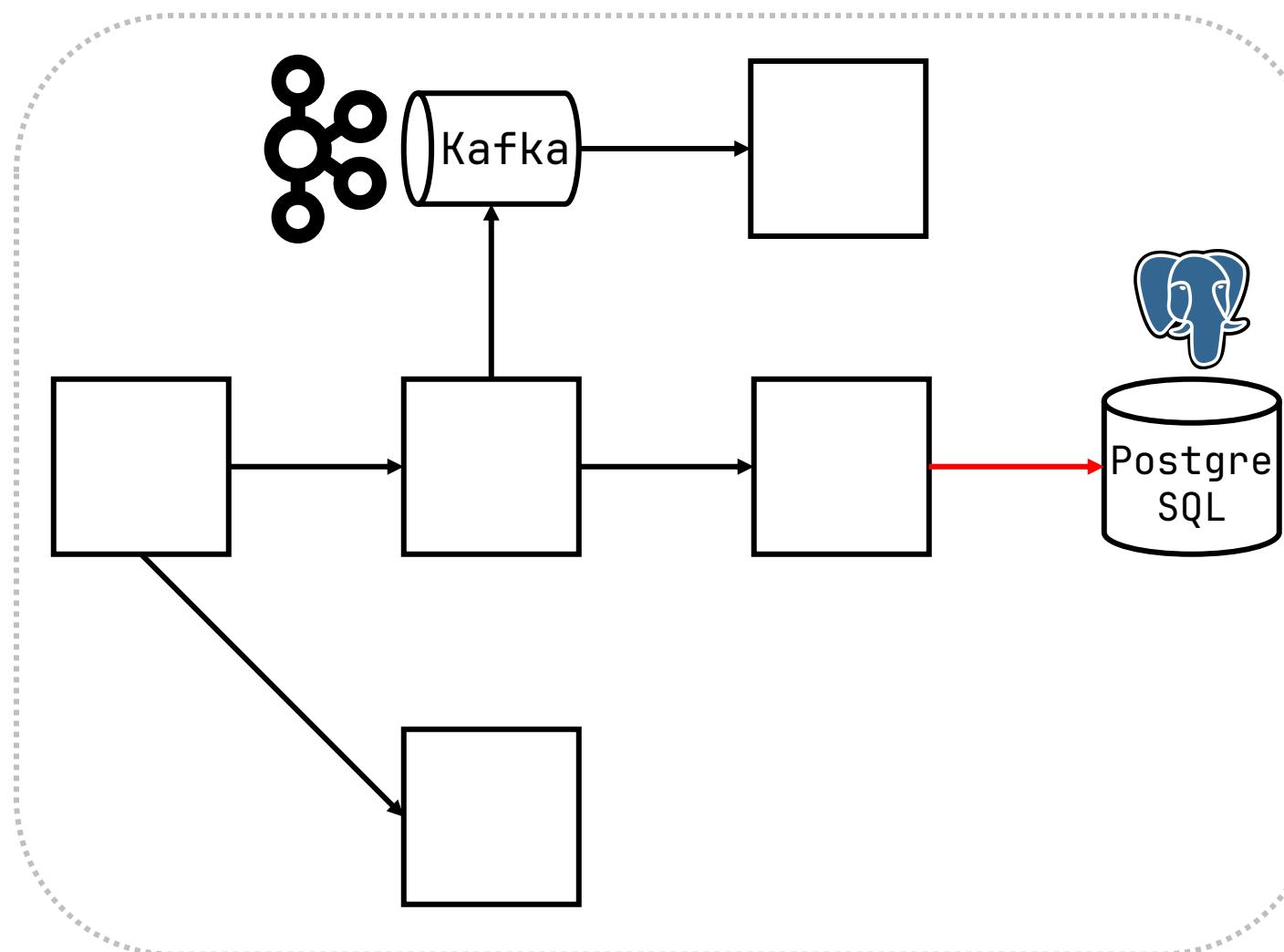
# Распределённая система



# Распределённая система



# Распределённая система



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

# Observability

# **Observability**

Lightning Talk

«В поисках определения Observability,  
или «Почему» и «зачем» не одно и то же»

# Observability

«Observability is the ability to measure  
a system's current state based on the data it generates,  
recorded as **logs**, **metrics**, and **traces**»

Jay Livens, Dynatrace

# Observability

«Observability is the ability to measure  
a system's current state based on the data it generates,  
recorded as **logs**, **metrics**, and **traces**»

Jay Livens, Dynatrace

Observability — это способность измерять  
текущее состояние системы по данным **телеметрии**

# Телеметрия

# Телеметрия

— Результат **наблюдения** за системой

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

## Событие

- тип
- метка времени
- идентификатор (*опционально*)
- данные о наблюдении

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

## Событие

- **тип**
- метка времени
- идентификатор (*опционально*)
- данные о наблюдении

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

## Событие

- тип
- **метка времени**
- идентификатор (*опционально*)
- данные о наблюдении

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

## Событие

- тип
- метка времени
- **идентификатор** (*опционально*)
- данные о наблюдении

# Телеметрия

- Результат **наблюдения** за системой
- Выражается последовательностью **событий**

## Событие

- тип
- метка времени
- идентификатор (*опционально*)
- **данные о наблюдении**

# Телеметрия

Сколько видов телеметрии?

# Телеметрия

Сколько видов телеметрии?

— 3?

# Телеметрия

Сколько видов телеметрии?

- 3?
- до 5?

# Телеметрия

Сколько видов телеметрии?

- 3?
- до 5?
- до 10?

# Телеметрия

Сколько видов телеметрии?

- 3?
- до 5?
- до 10?
- более 10?

# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Телеметрия

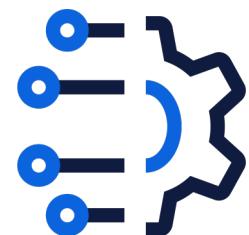
Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Телеметрия

Базовая телеметрия

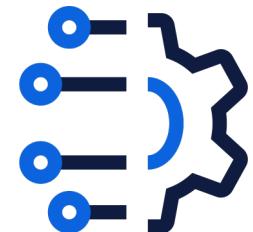
- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

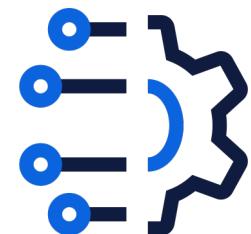
- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- **Метрики**
- Распределённые  
трассировки

# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые трассировки



# Телеметрия

Базовая телеметрия

- Логи
- **Метрики**
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Распределённые трассировки

 **Honest Update**  
@honest\_update ...

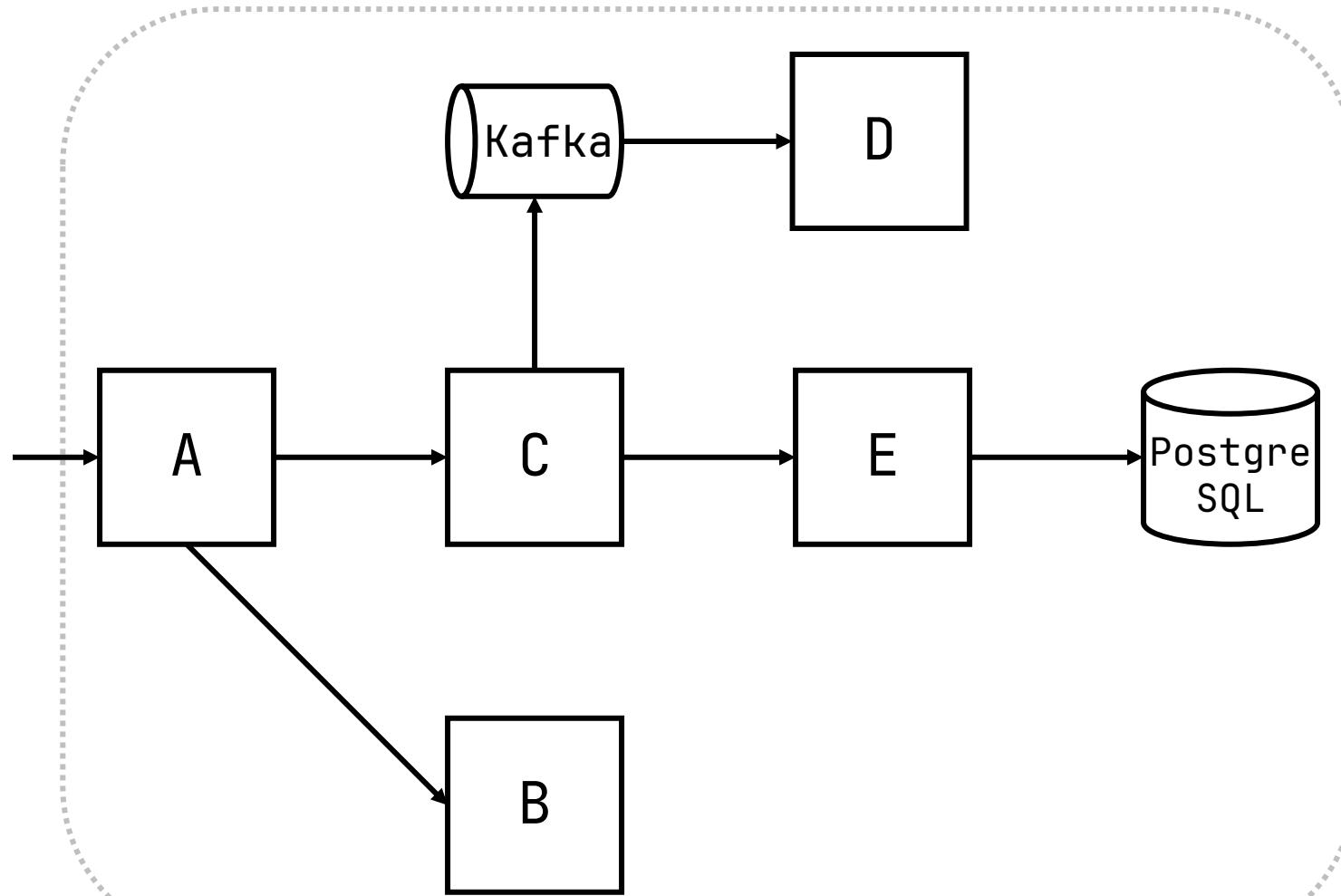
We replaced our monolith with micro services so that every outage could be more like a murder mystery.

4:10 AM · 8 окт. 2015 г. · Buffer

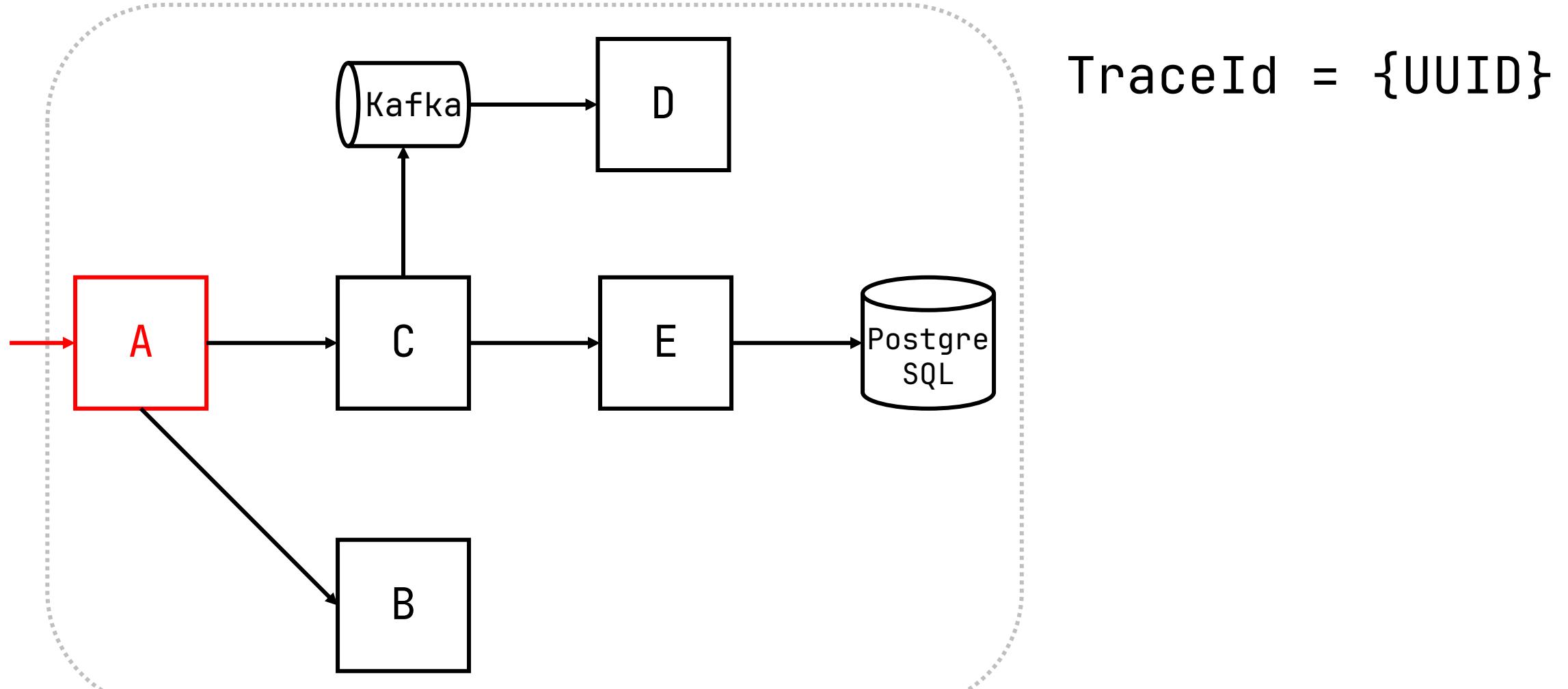
---

**2 900** ретвитов   **73** твита с цитатами   **2 709** отметок «Нравится»

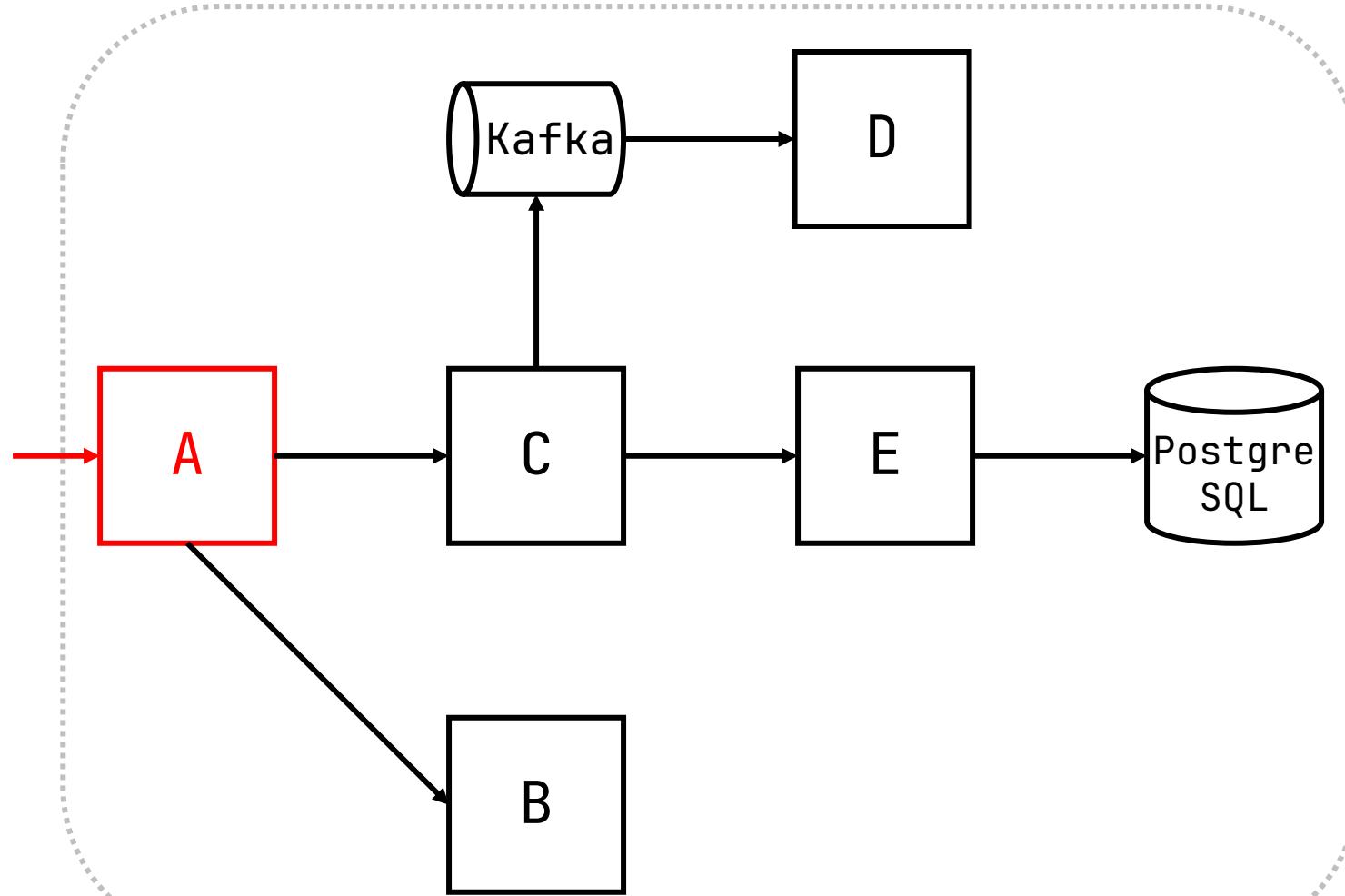
# Распределённые трассировки



# Распределённые трассировки

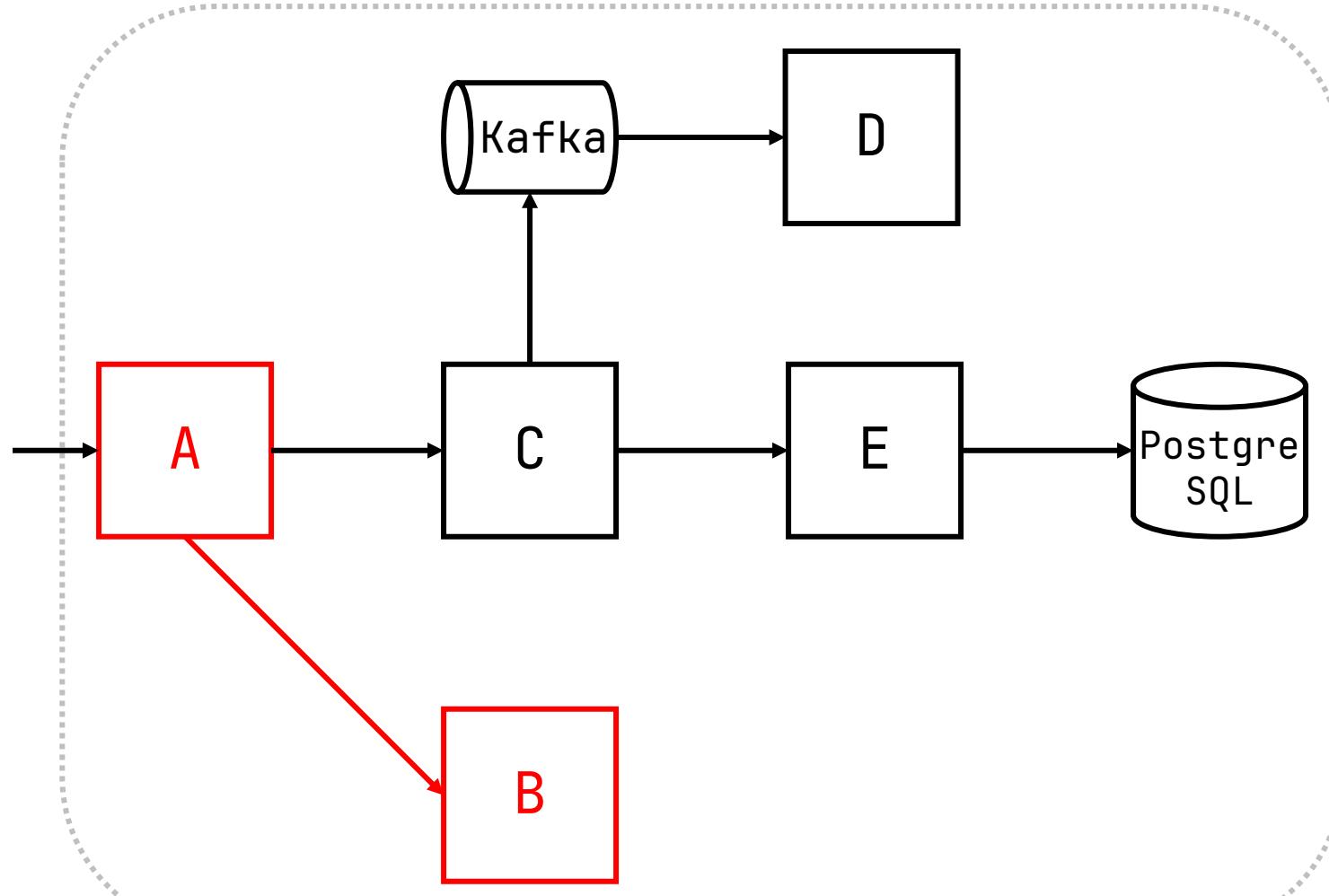


# Распределённые трассировки



TraceId = {UUID}  
span user→A

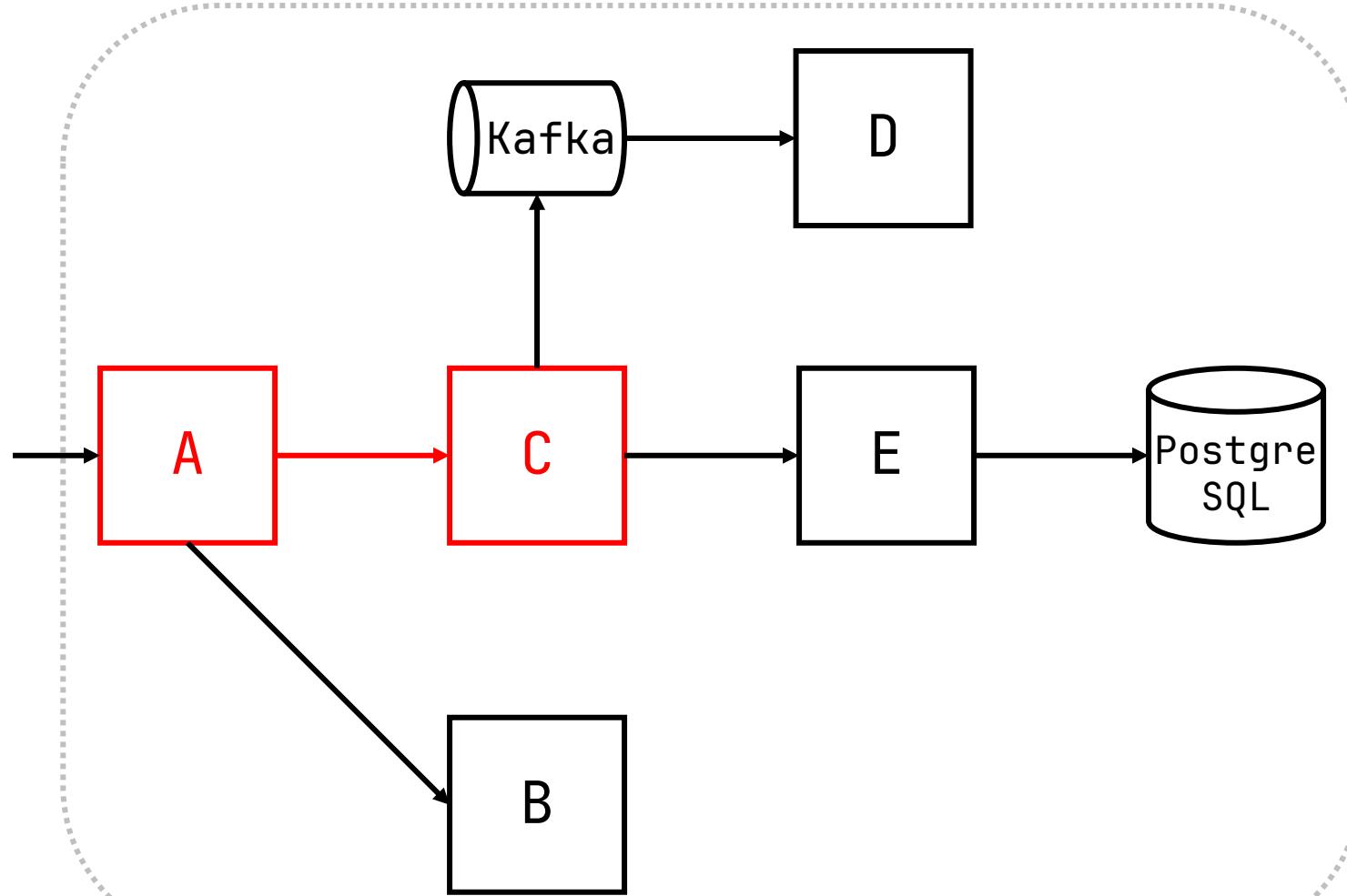
# Распределённые трассировки



TraceId = {UUID}

span user->A  
span A->B

# Распределённые трассировки



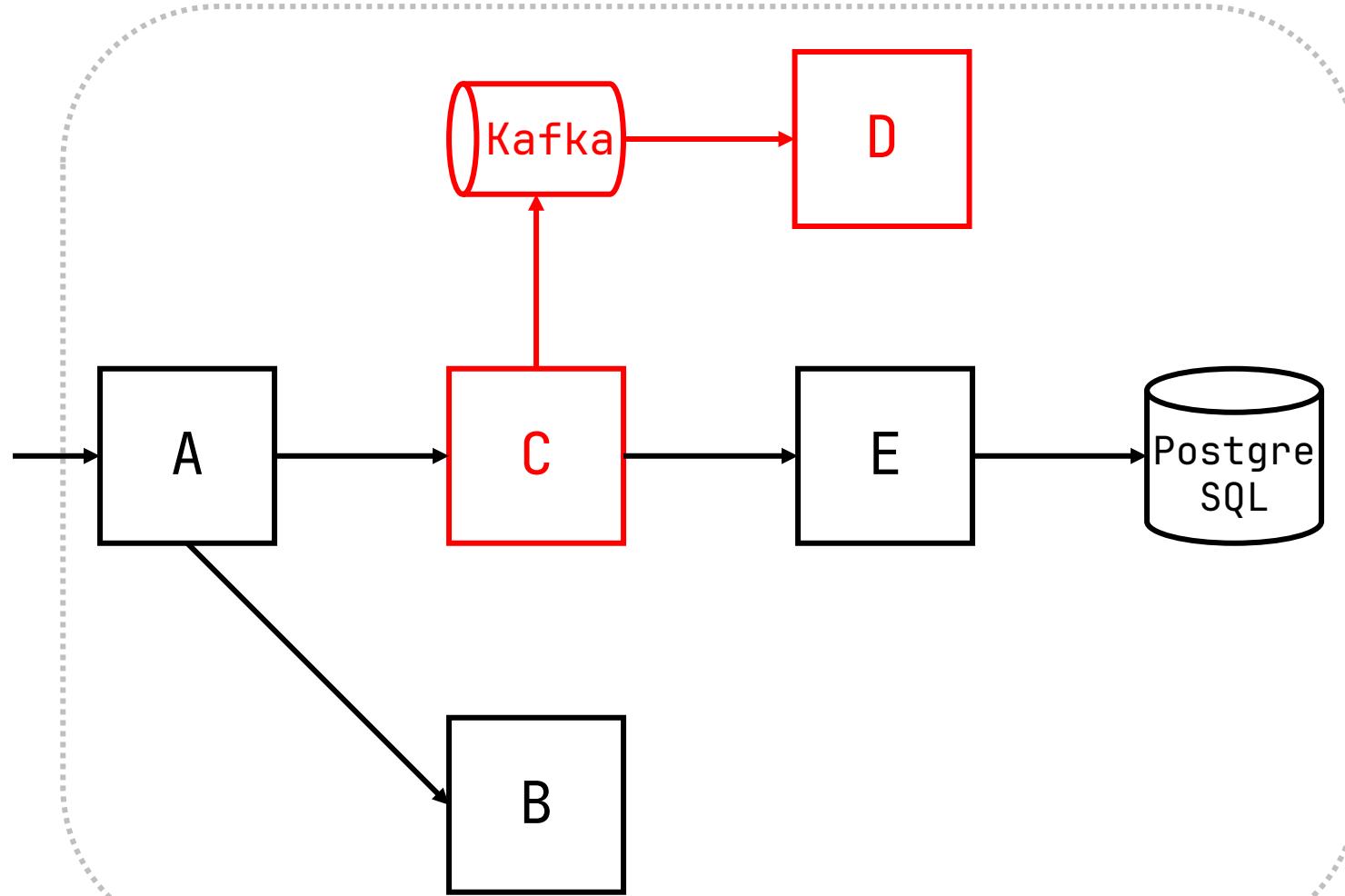
TraceId = {UUID}

span user->A

span A->B

span A->C

# Распределённые трассировки



TraceId = {UUID}

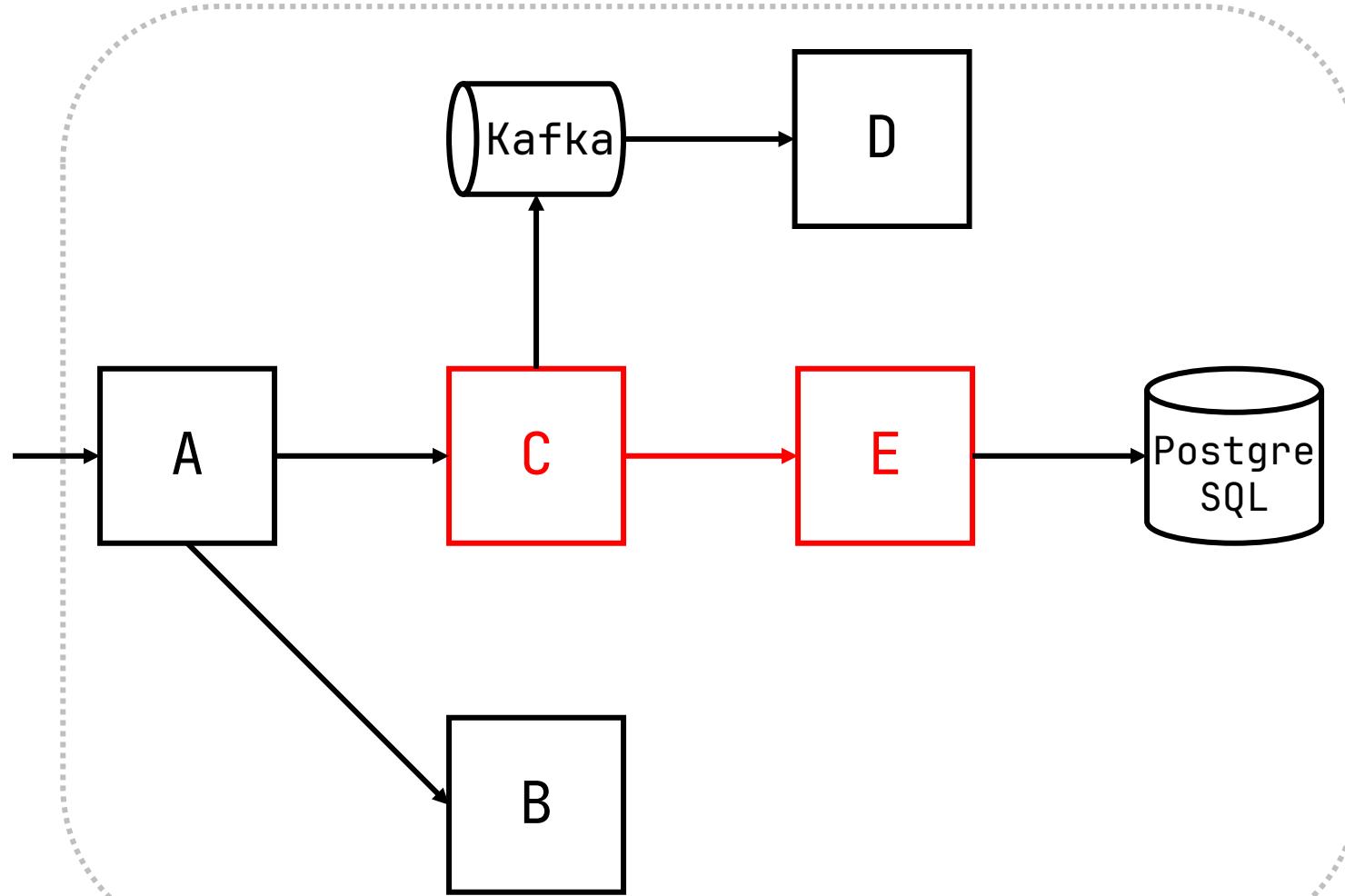
span user->A

span A->B

span A->C

span C->D

# Распределённые трассировки



TraceId = {UUID}

span user→A

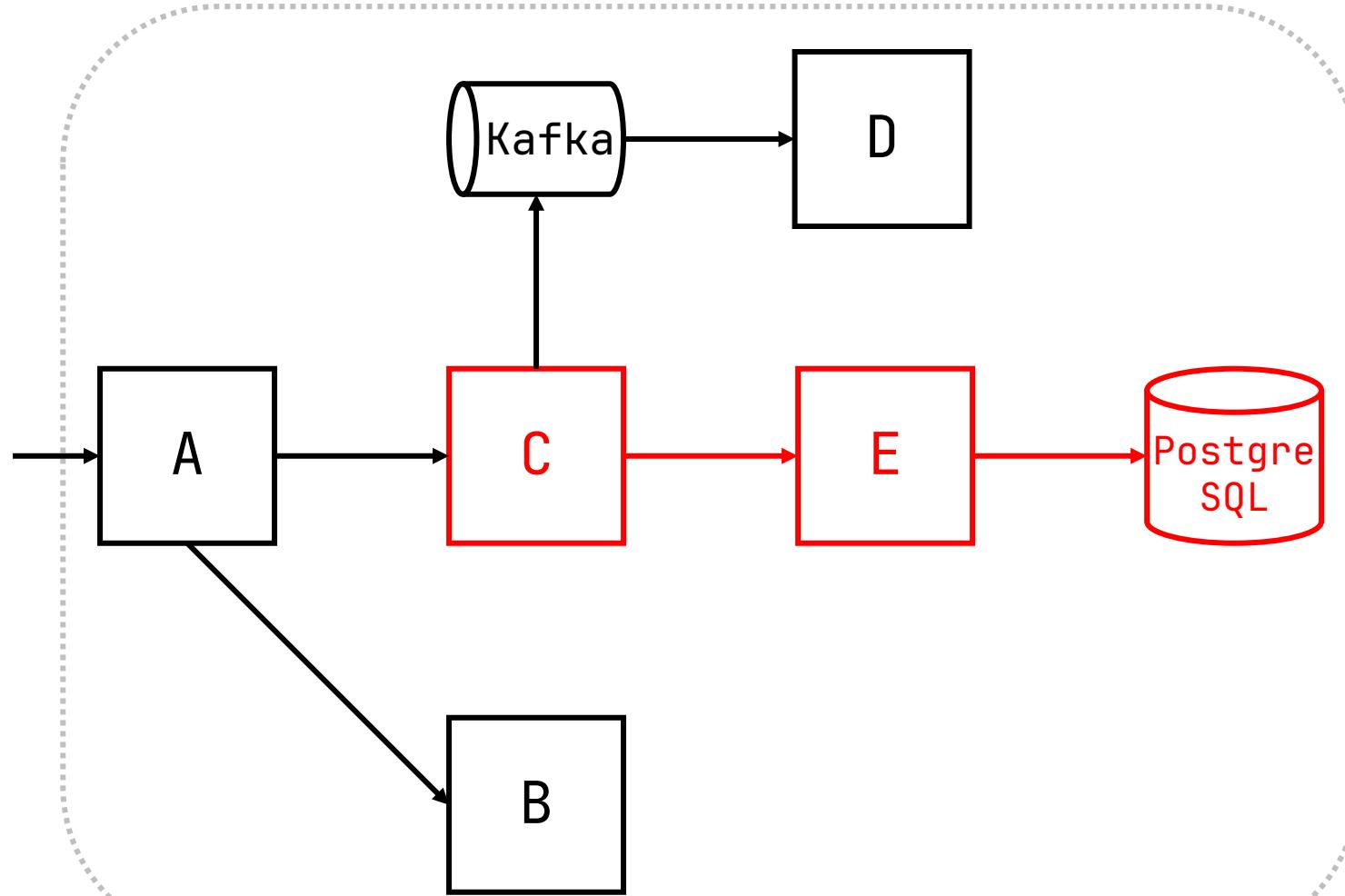
span A→B

span A→C

span C→D

span C→E

# Распределённые трассировки



TraceId = {UUID}

span user→A

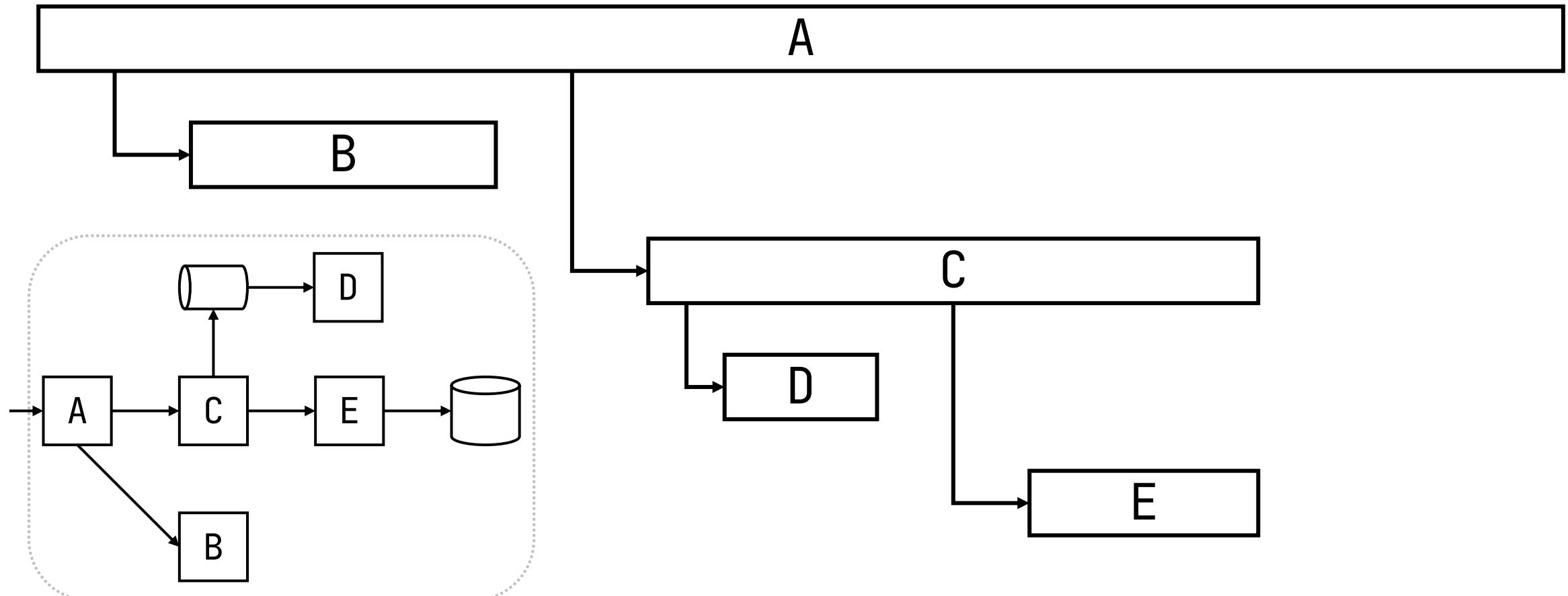
span A→B

span A→C

span C→D

span C→E

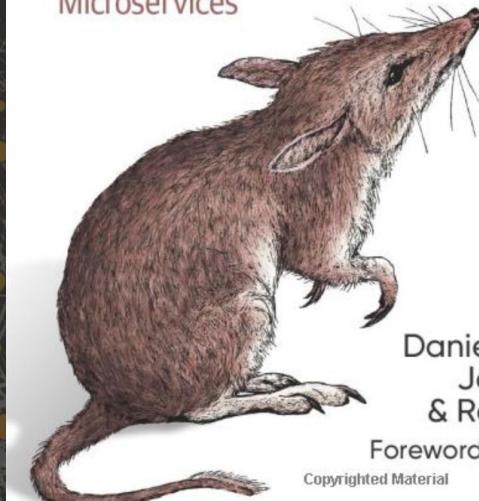
# Распределённые трассировки



O'REILLY®

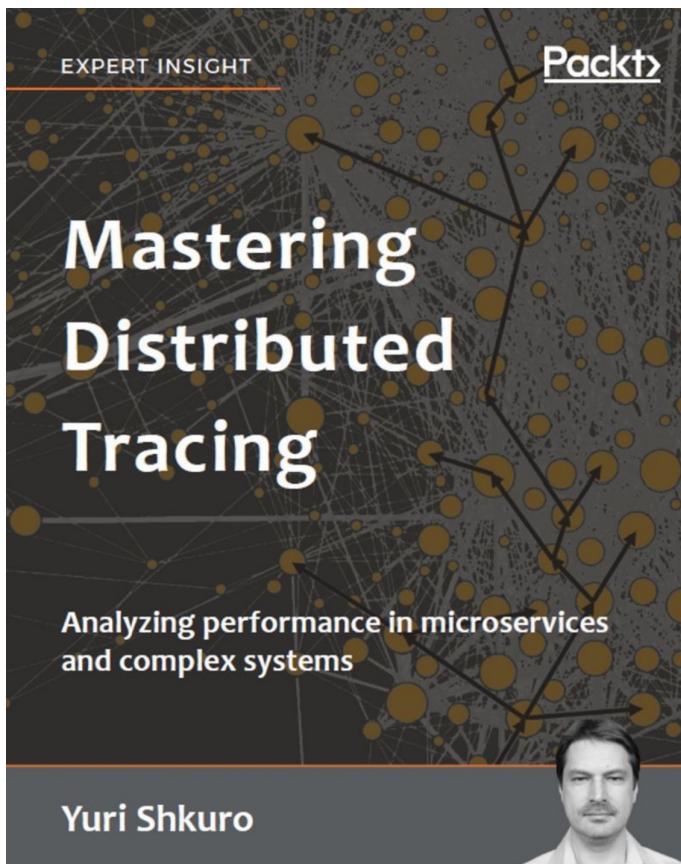
# Distributed Tracing in Practice

Instrumenting, Analyzing, and Debugging Microservices



Austin Parker,  
Daniel Spoonhower,  
Jonathan Mace  
& Rebecca Isaacs  
Foreword by Ben Sigelman

Copyrighted Material



# Две главные книги

- Mastering Distributed Tracing, 2019
- Distributed Tracing in Practice, 2020

# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки



# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Application Performance Management\*

- New Relic
- Dynatrace
- Datadog
- Elastic APM
- Grafana Enterprise Stack

*\*) Часто называют Monitoring*

# **Application Performance Management\***

**\*)** Часто называют Monitoring





**Vostok Hercules:**  
**3 года доставляем**  
**телеинформацию**  
**– полёт нормальный**



**Григорий  
Кошелев**

Контур

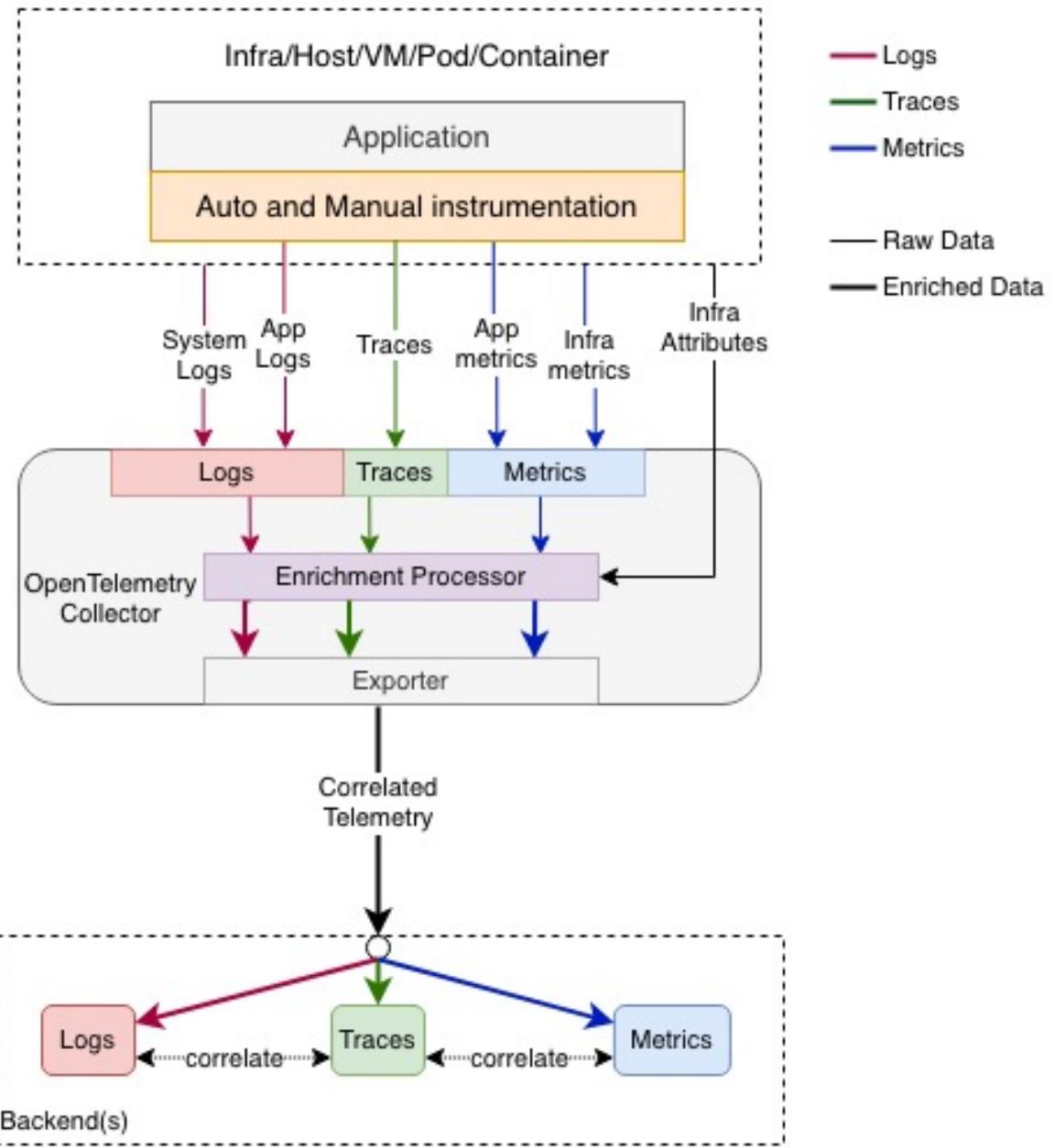
# OpenTelemetry

# OpenTelemetry

- Логи
- Метрики
- Распределённые  
трассировки

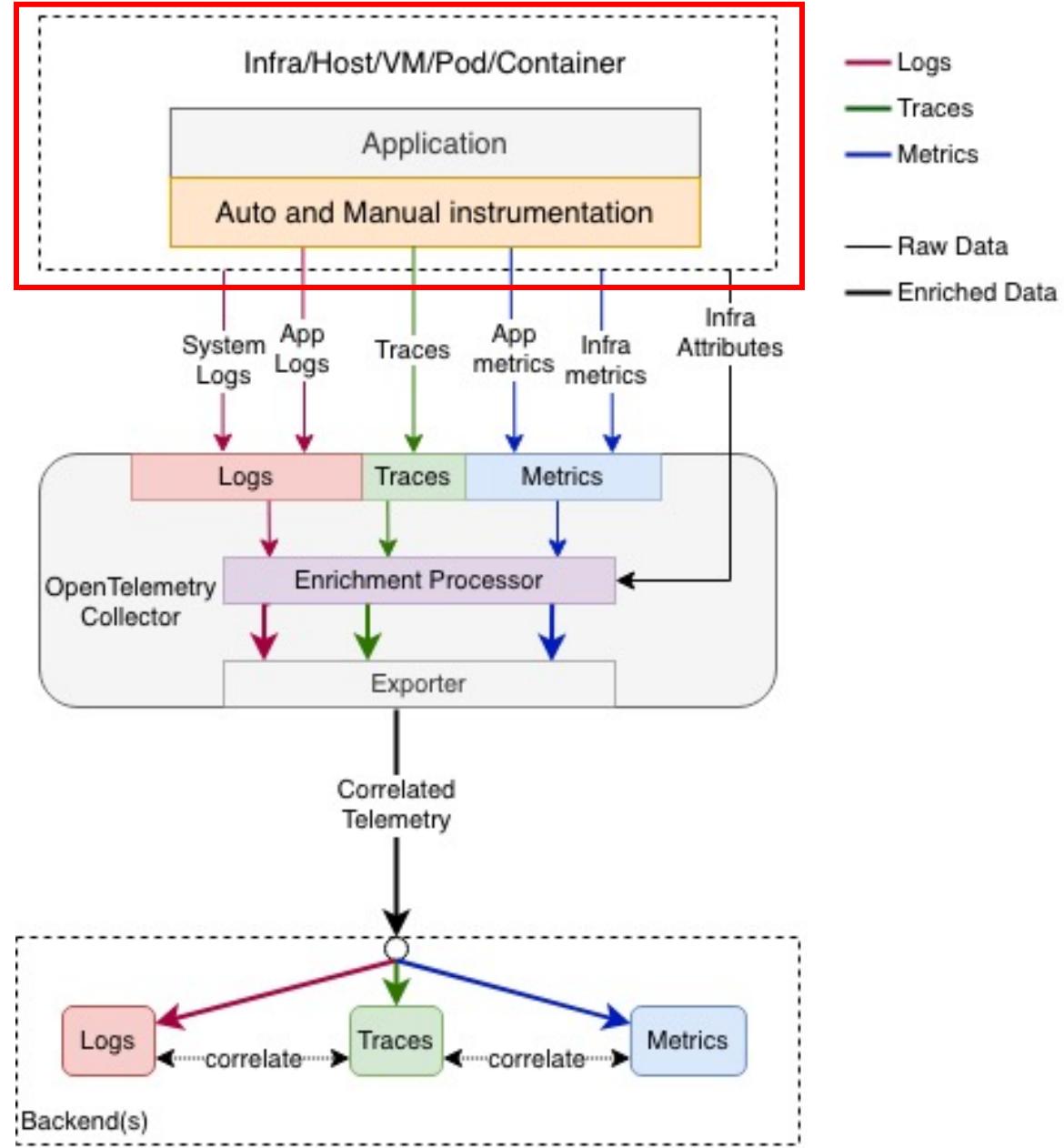
# OpenTelemetry

## OpenTelemetry Collection



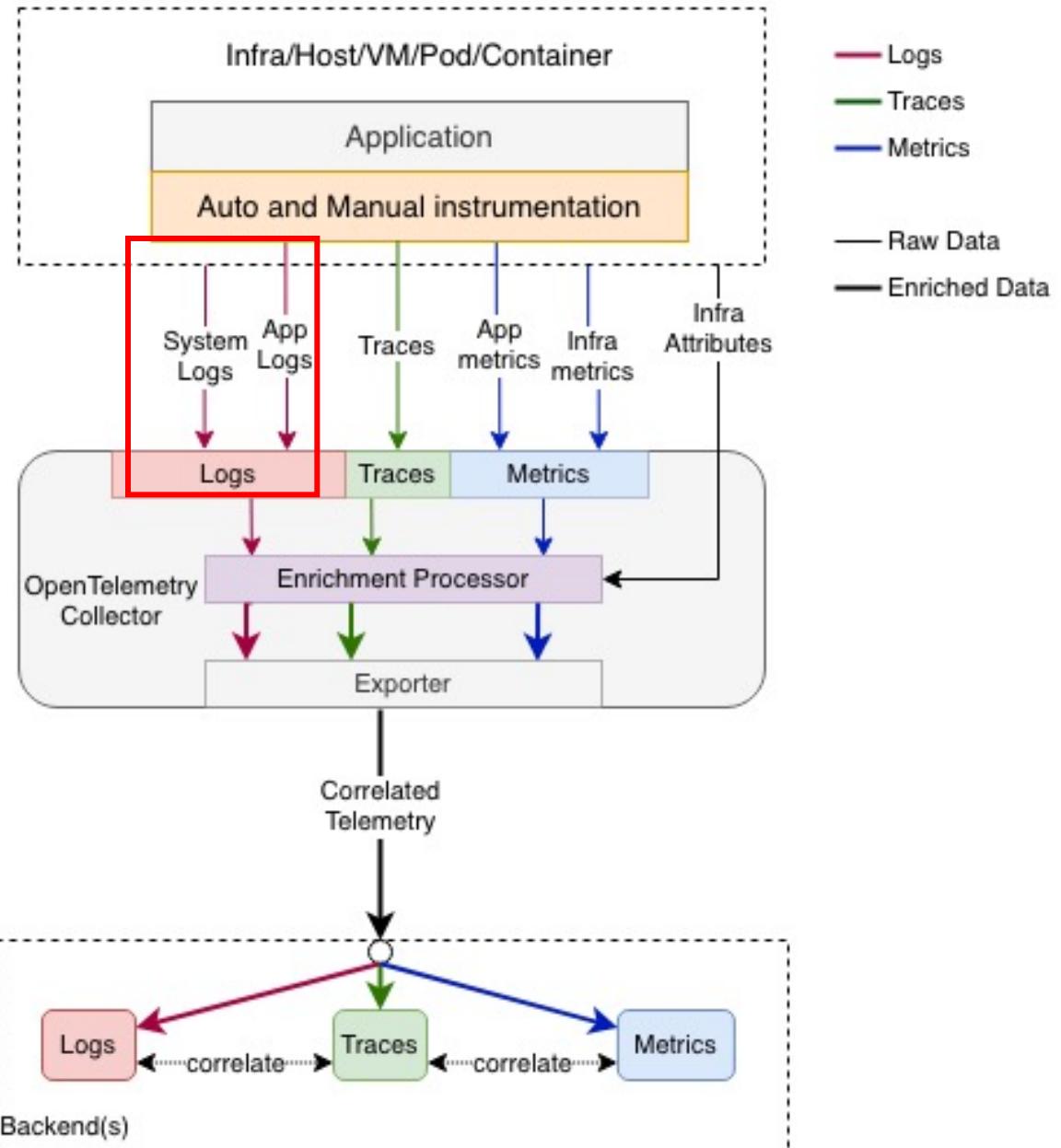
# OpenTelemetry

## OpenTelemetry Collection



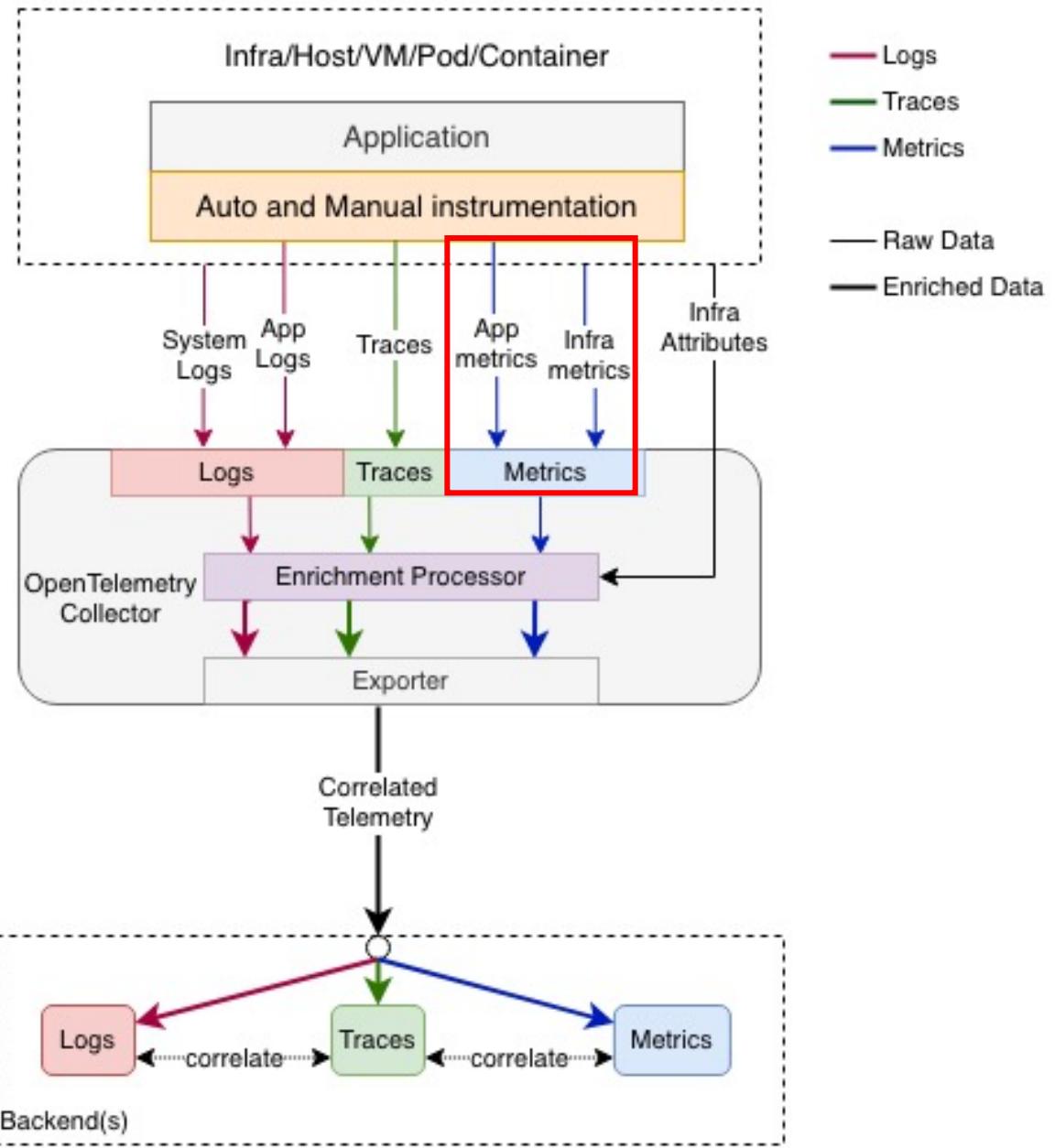
# OpenTelemetry

## OpenTelemetry Collection



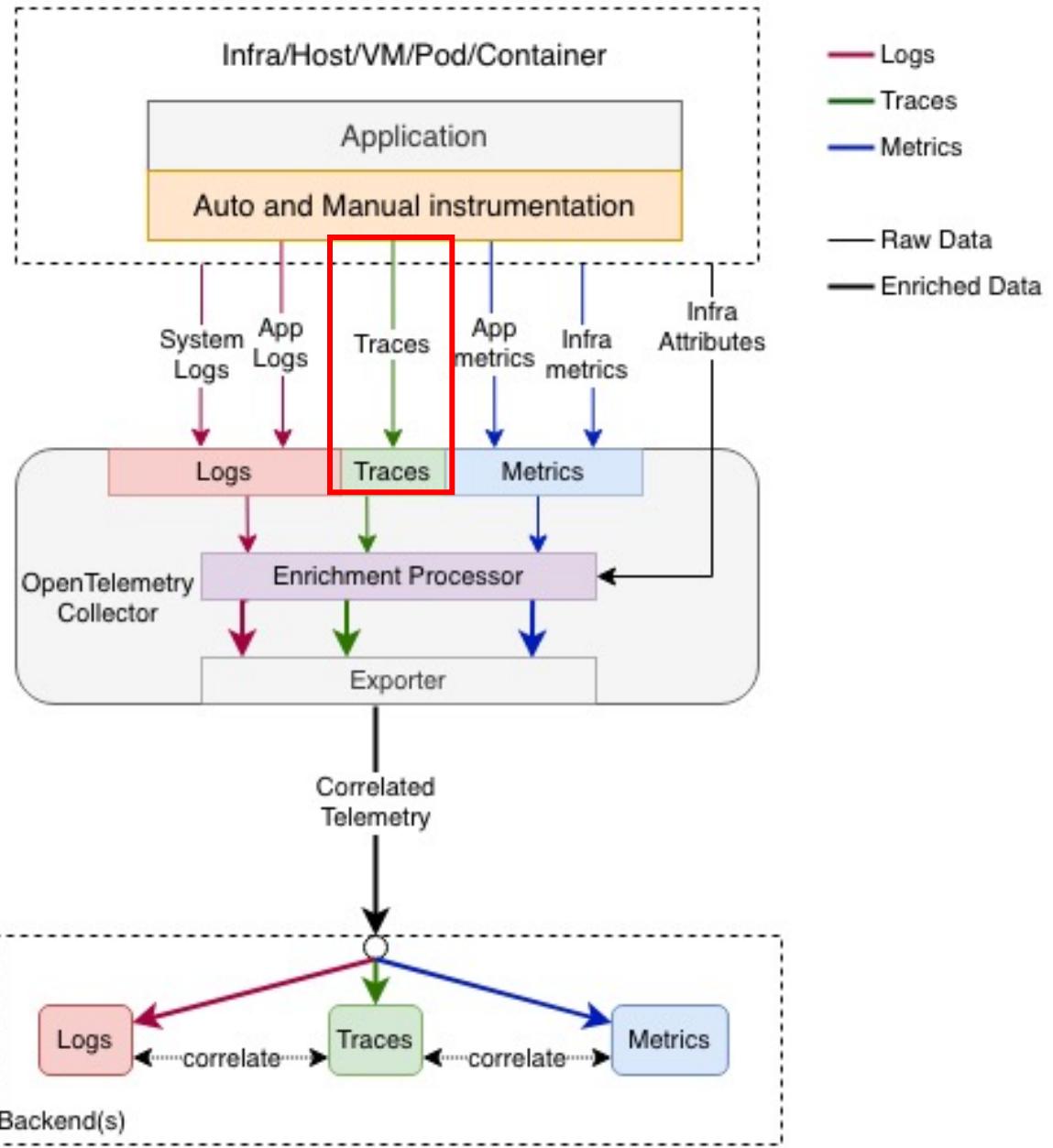
# OpenTelemetry

## OpenTelemetry Collection



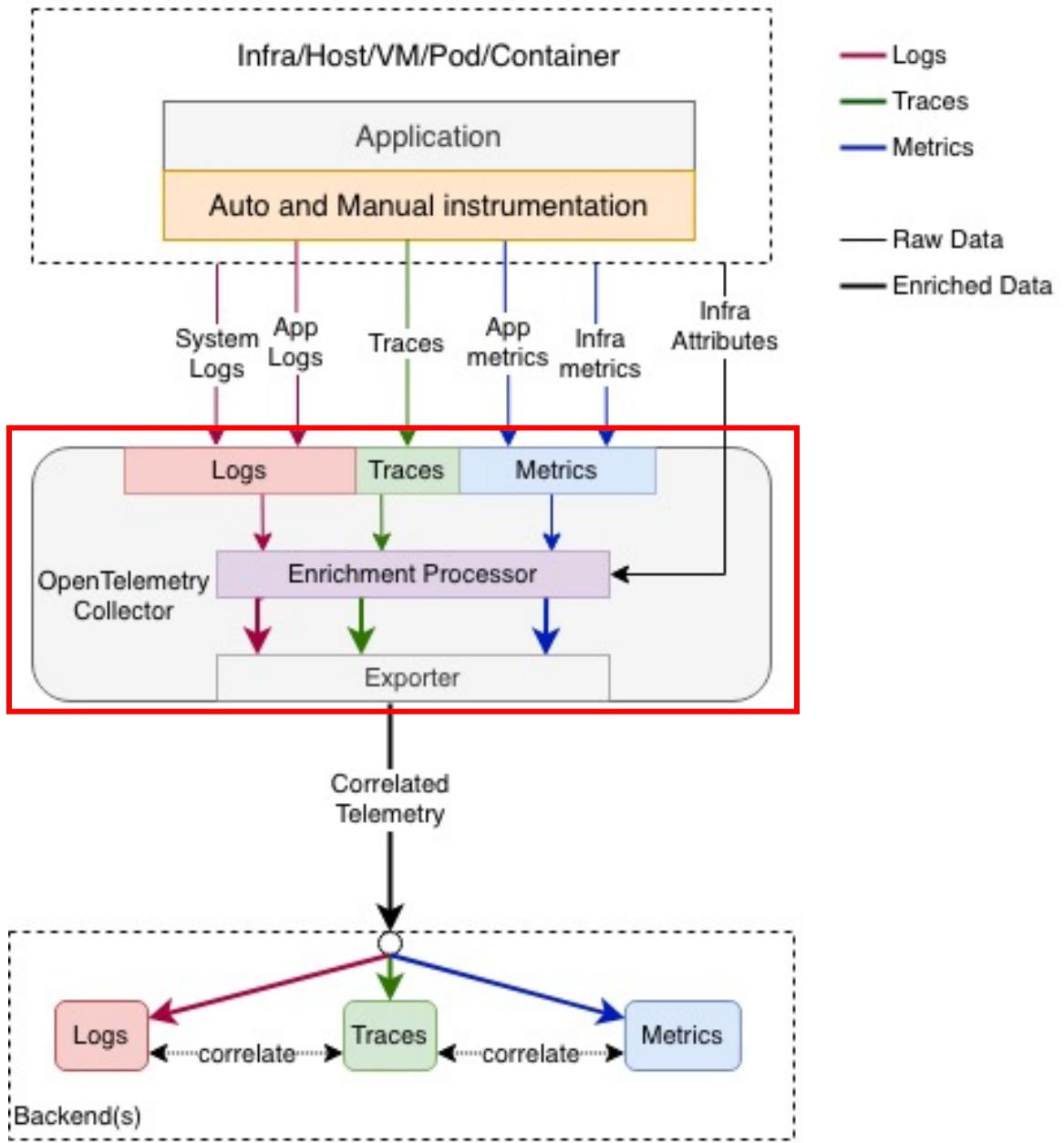
# OpenTelemetry

## OpenTelemetry Collection



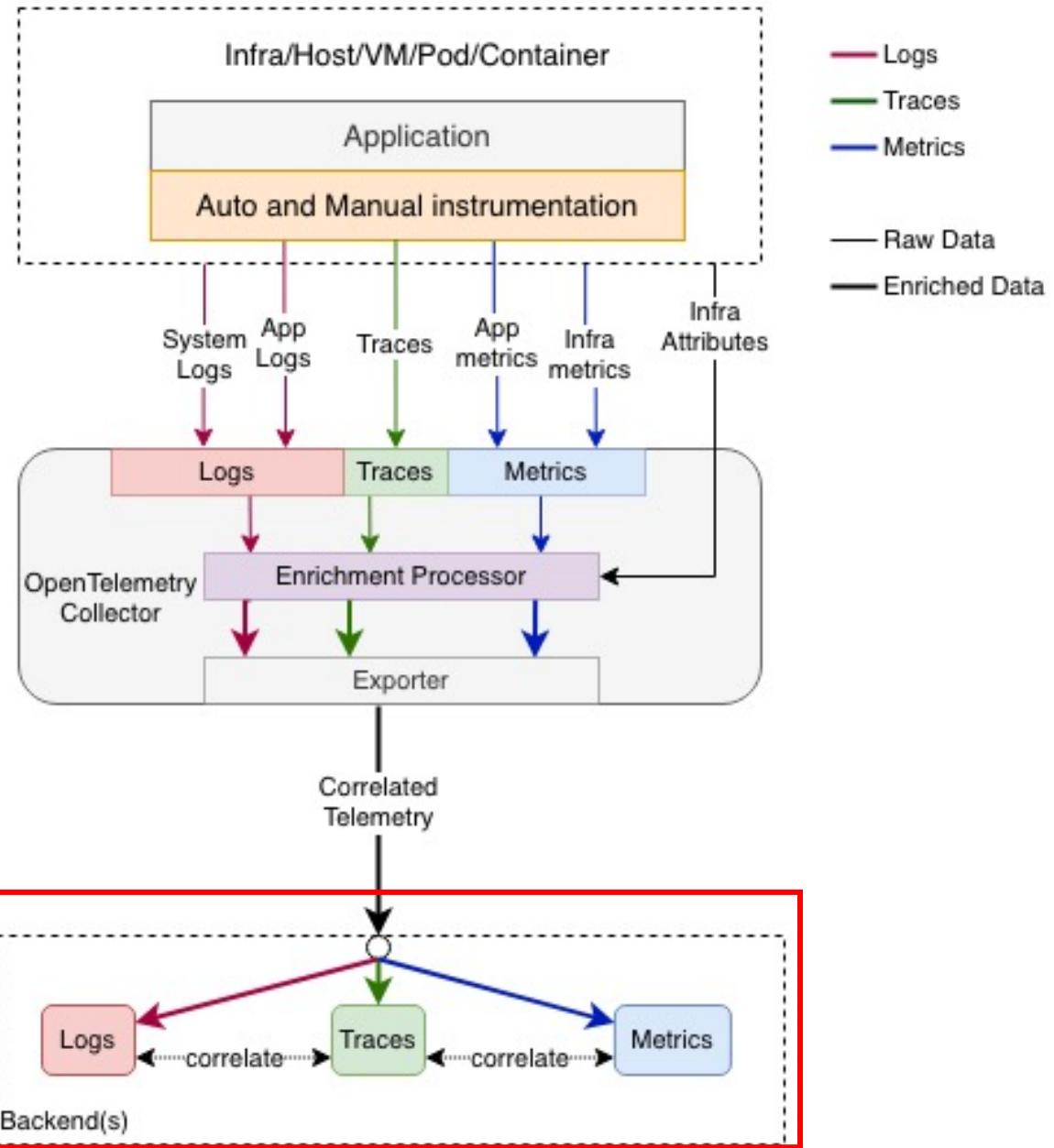
# OpenTelemetry

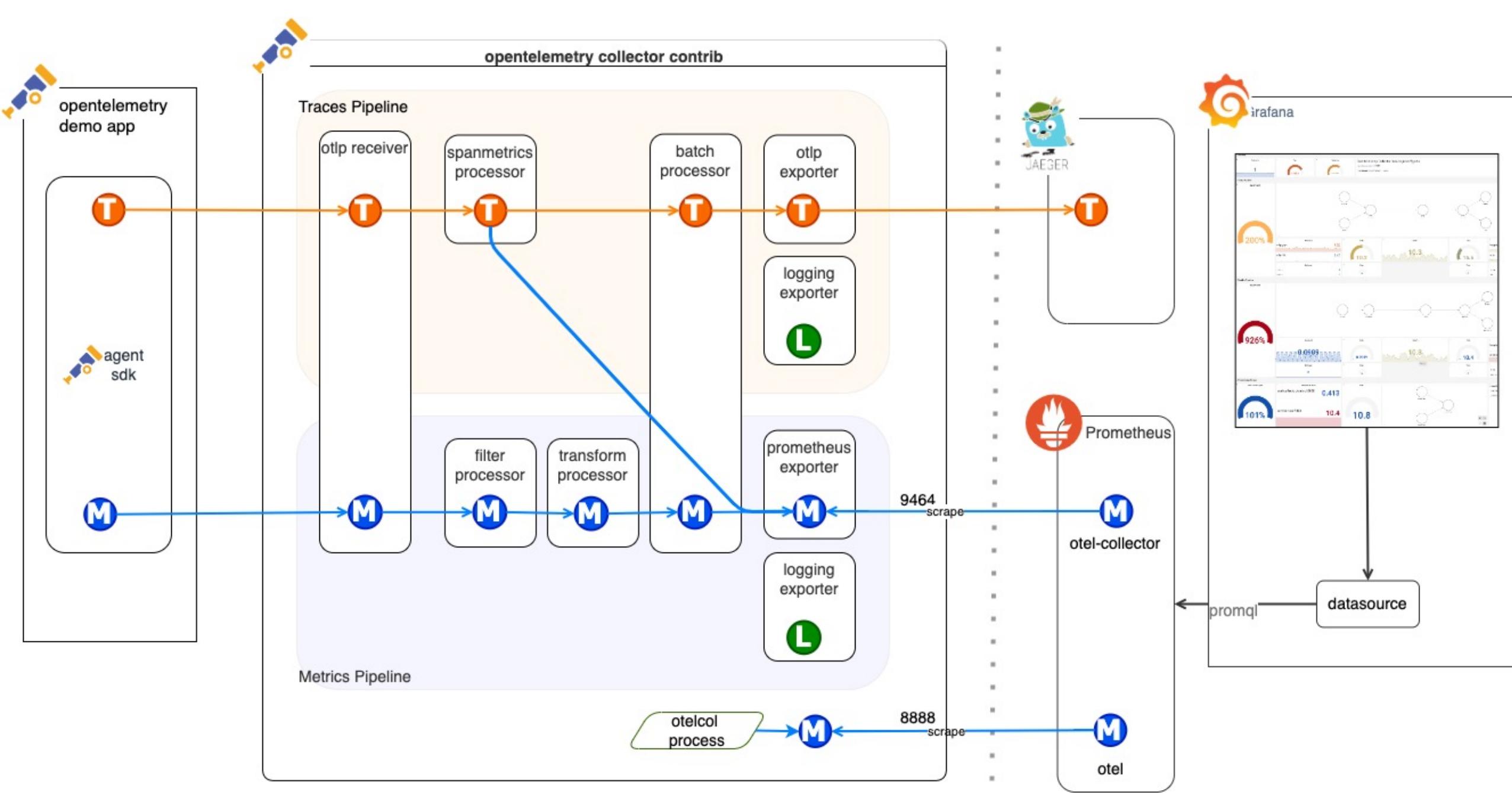
## OpenTelemetry Collection



# OpenTelemetry

## OpenTelemetry Collection





# **Зачем Observability?**

# **Зачем Observability?**

# Зачем Observability?

Система  
работает?

# Зачем Observability?

Система  
работает?

да / нет

# Зачем Observability?

Система  
работает?

да / нет

Насколько хорошо?

# Зачем Observability?

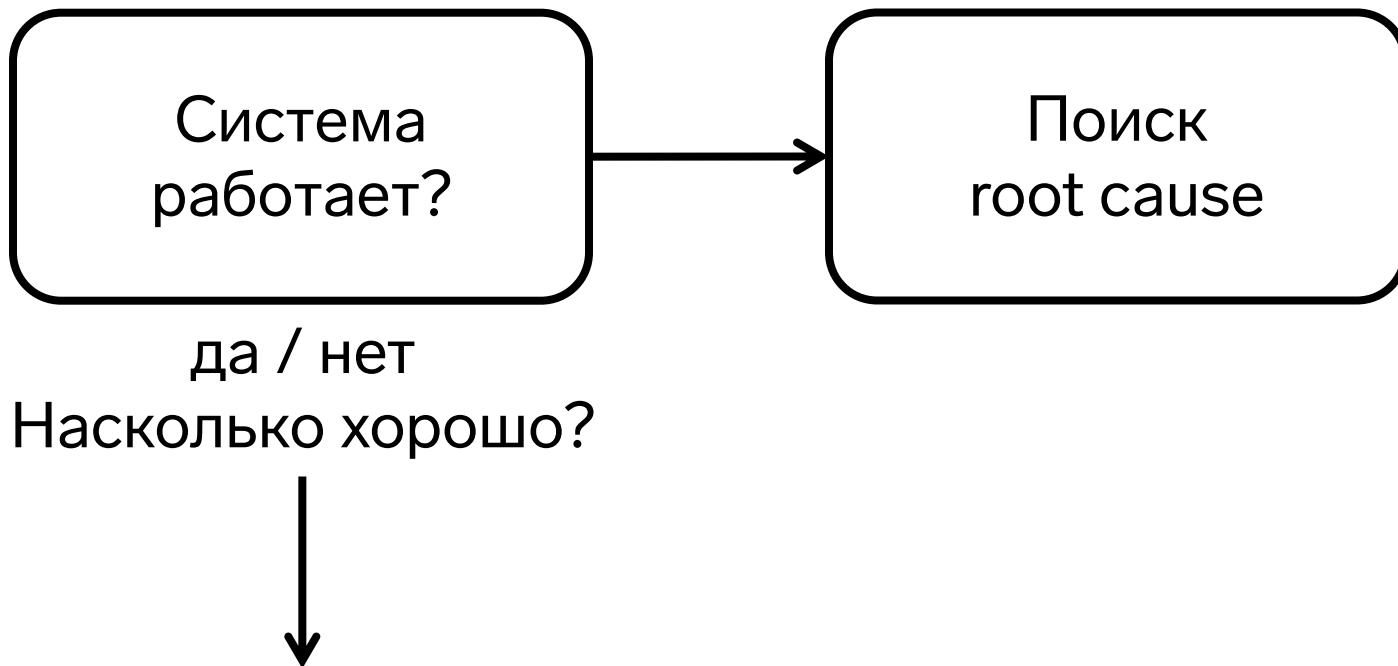
Система  
работает?

да / нет

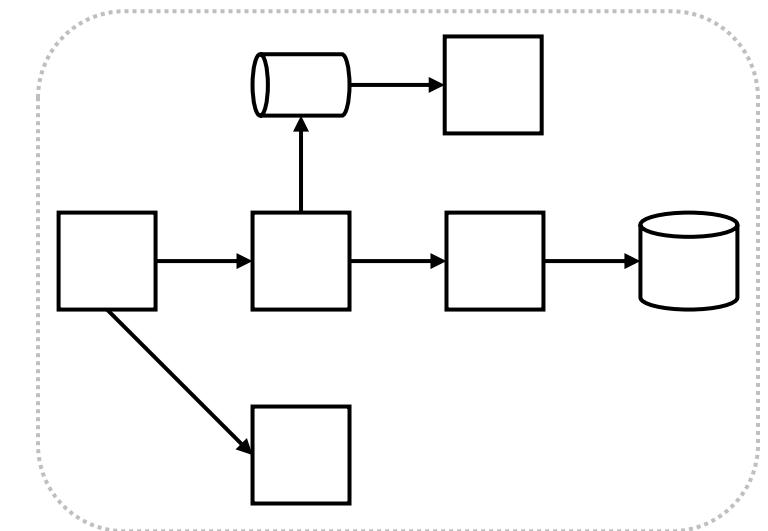
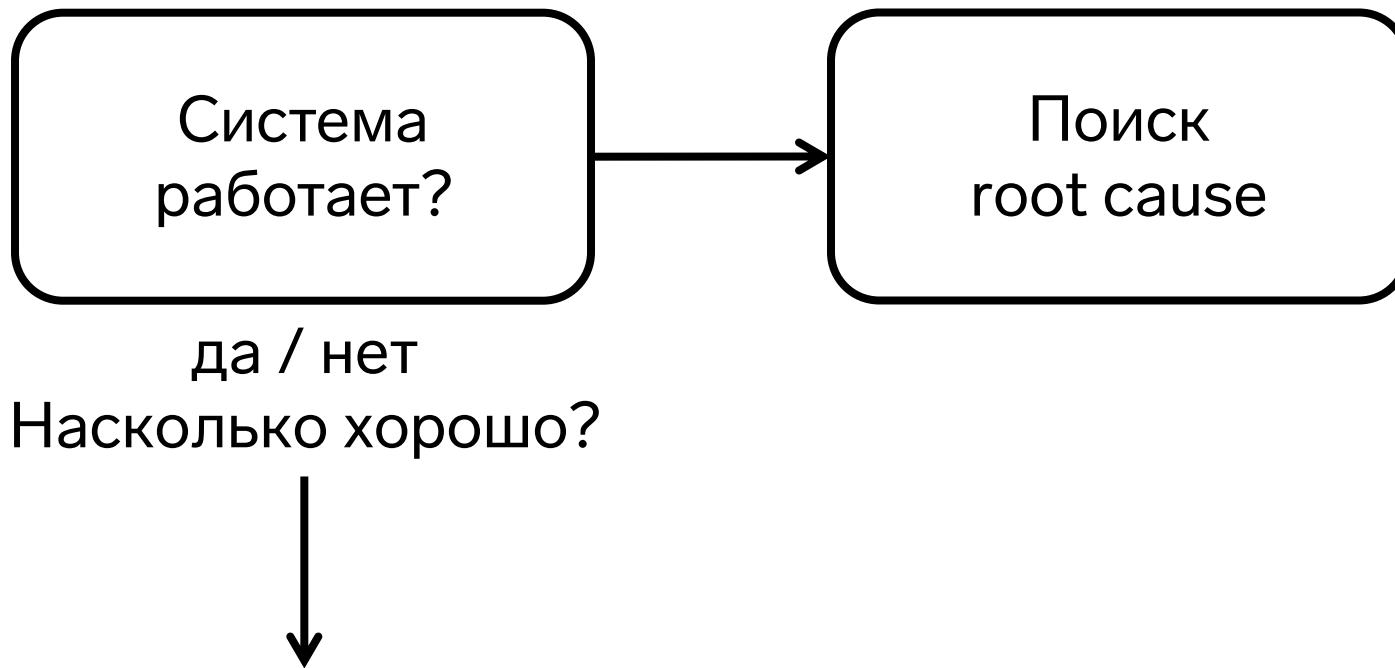
Насколько хорошо?



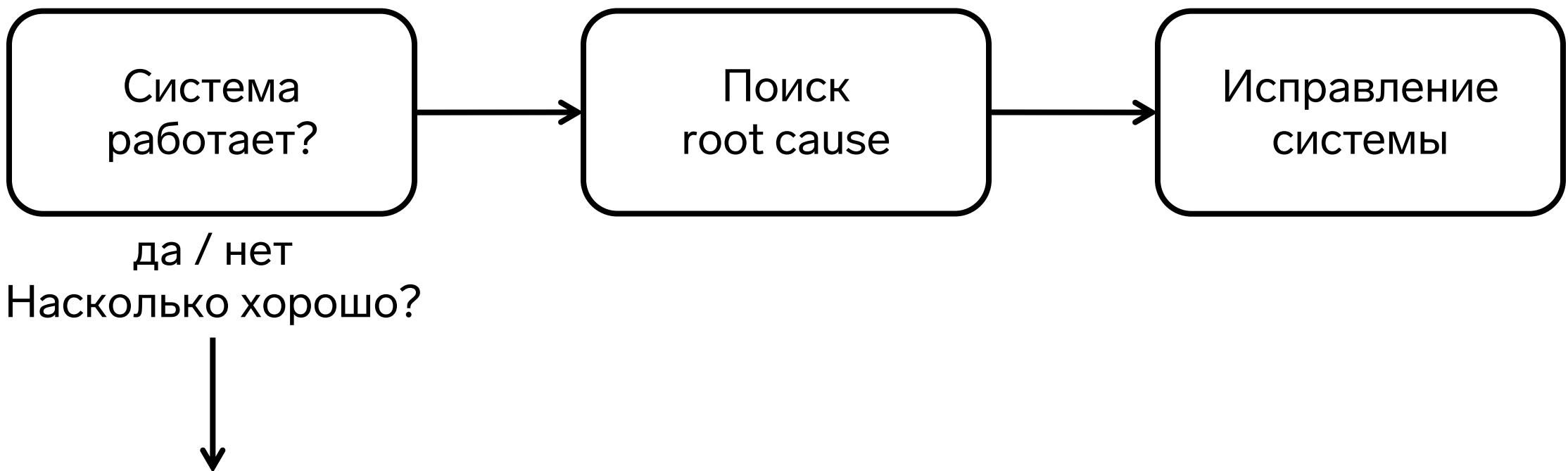
# Зачем Observability?



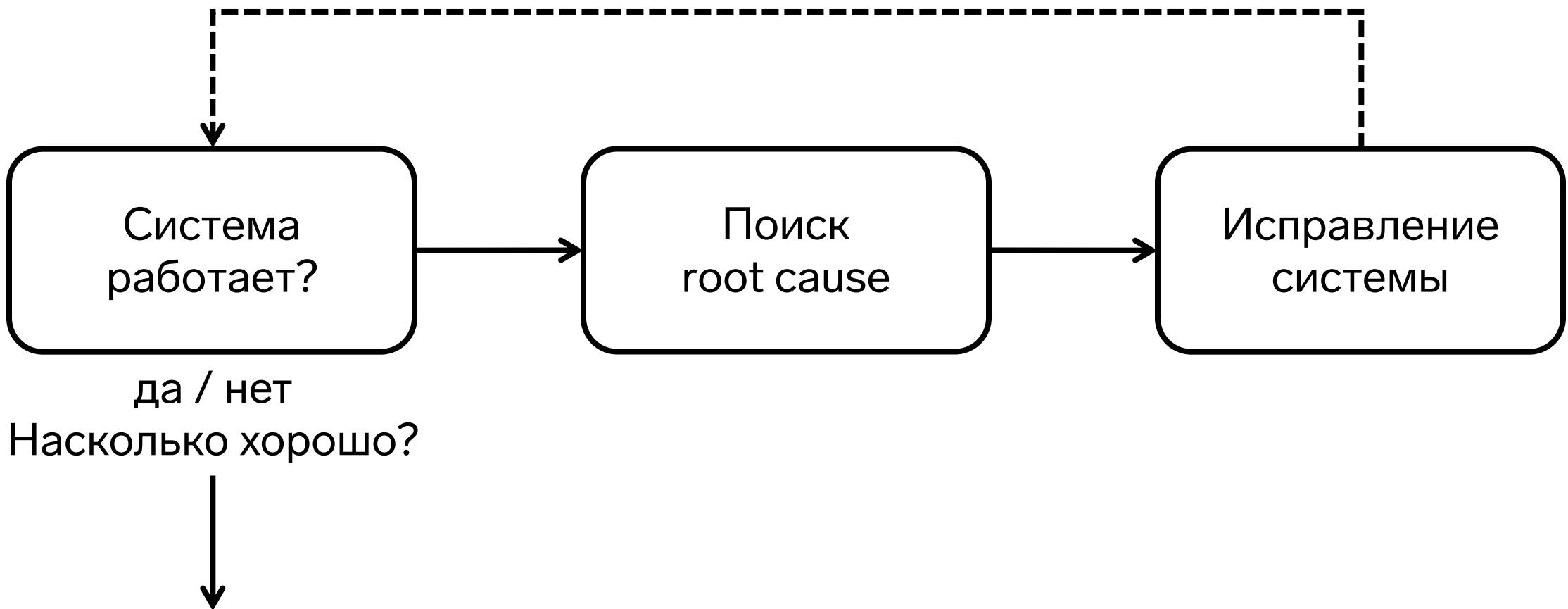
# Зачем Observability?



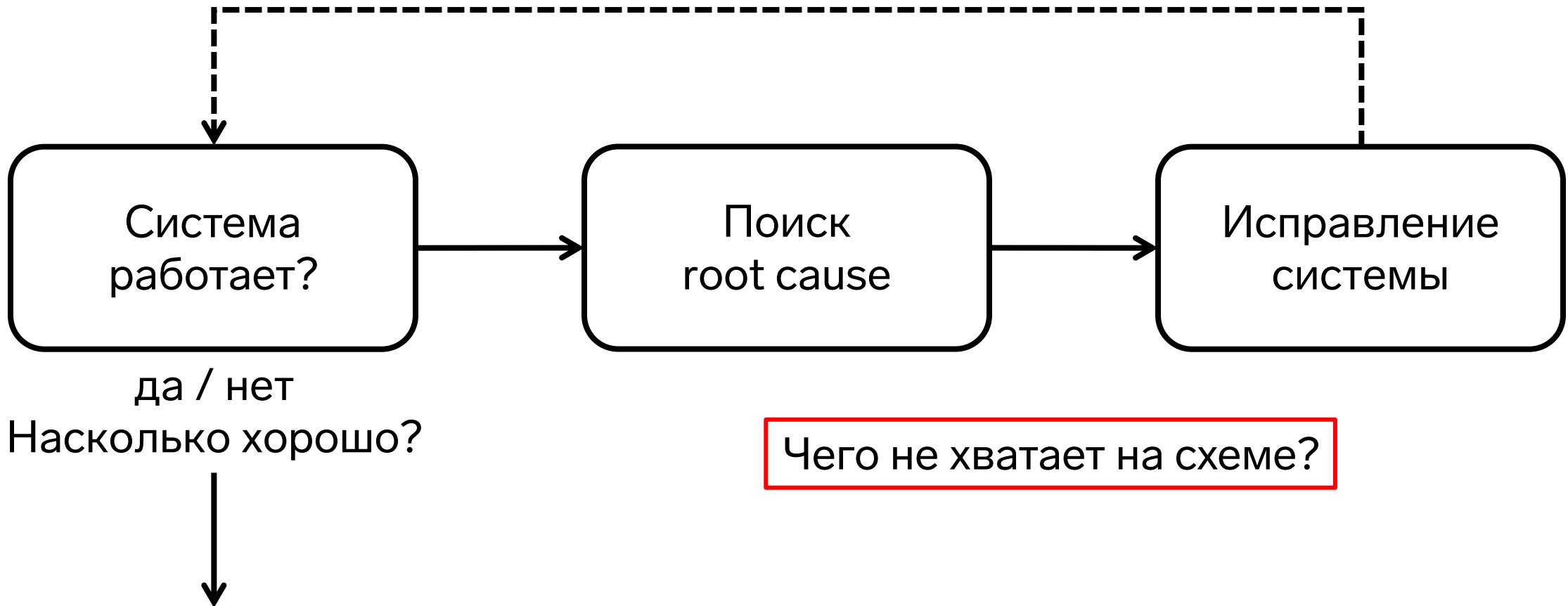
# Зачем Observability?



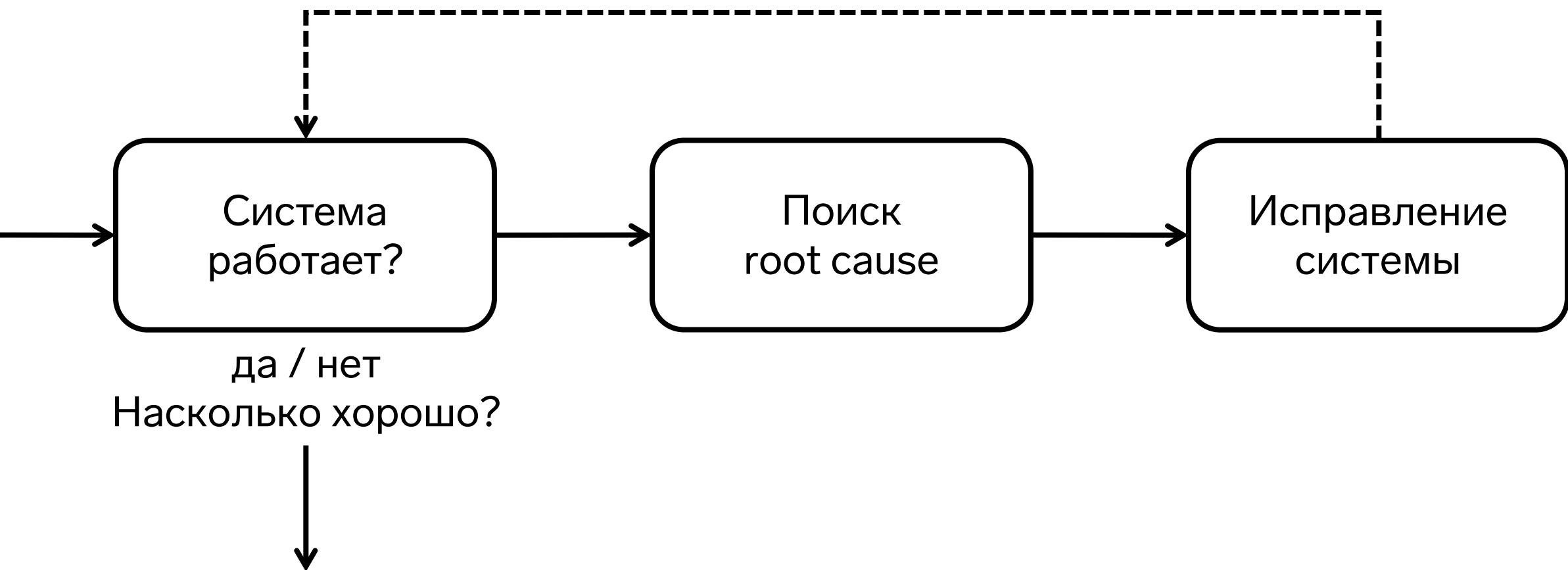
# Зачем Observability?



# Зачем Observability?

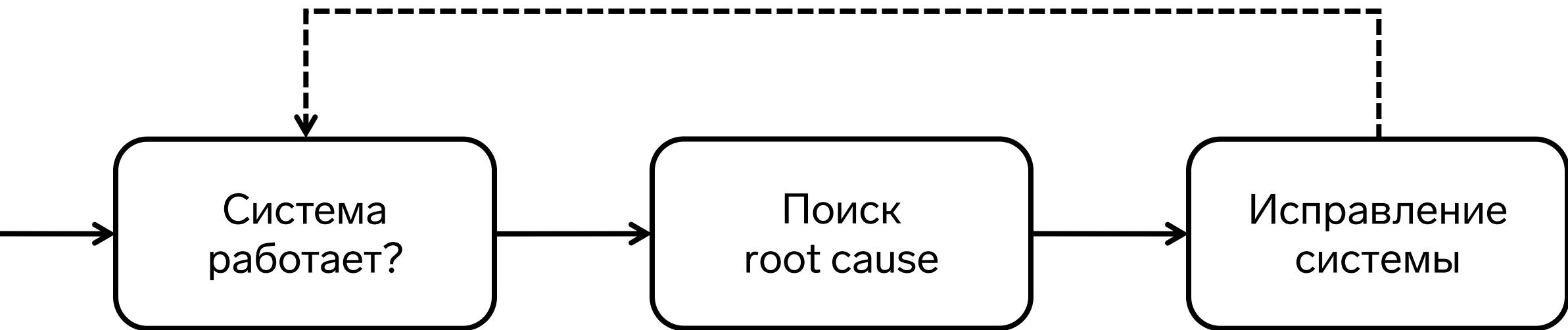


# Зачем Observability?



# Когда и как узнавать, что система сломалась?

# Когда и как узнавать, что система сломалась?



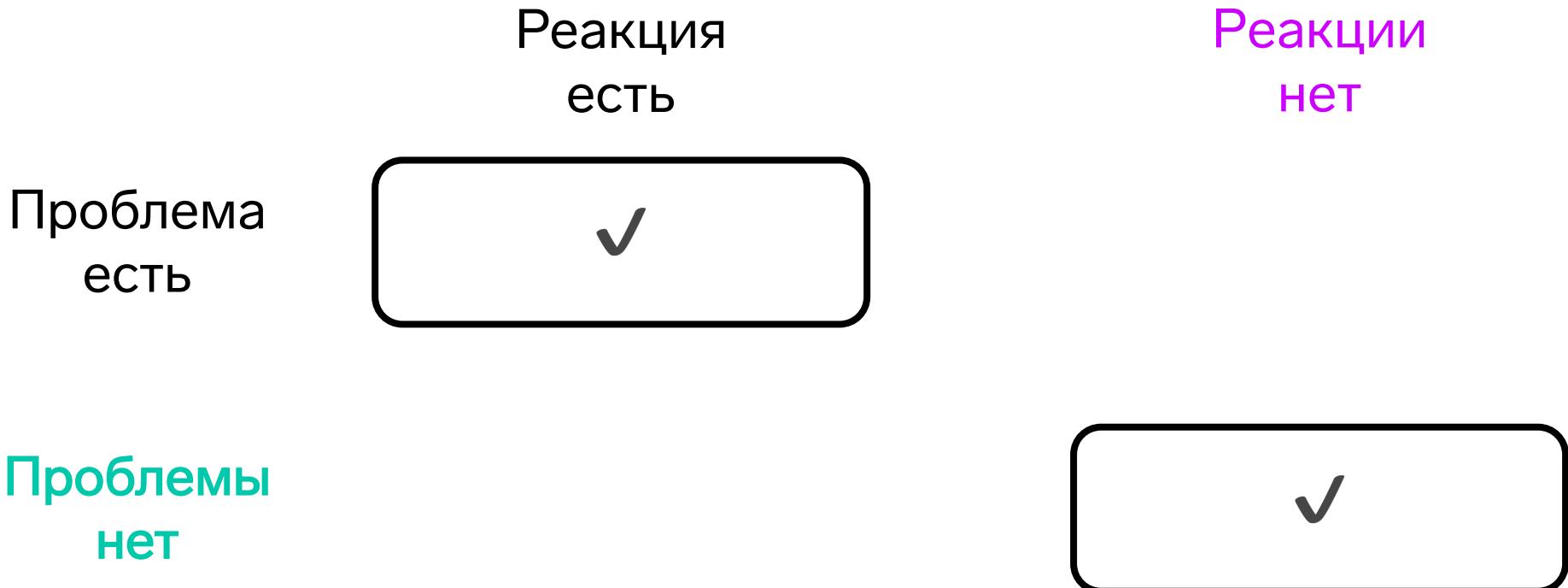
# Когда и как узнавать, что система сломалась?

Реакция  
есть

Проблема  
есть



# Когда и как узнавать, что система сломалась?



# Когда и как узнавать, что система сломалась?

Реакция  
есть

Проблема  
есть



Реакции  
нет

Пользователи  
недовольны

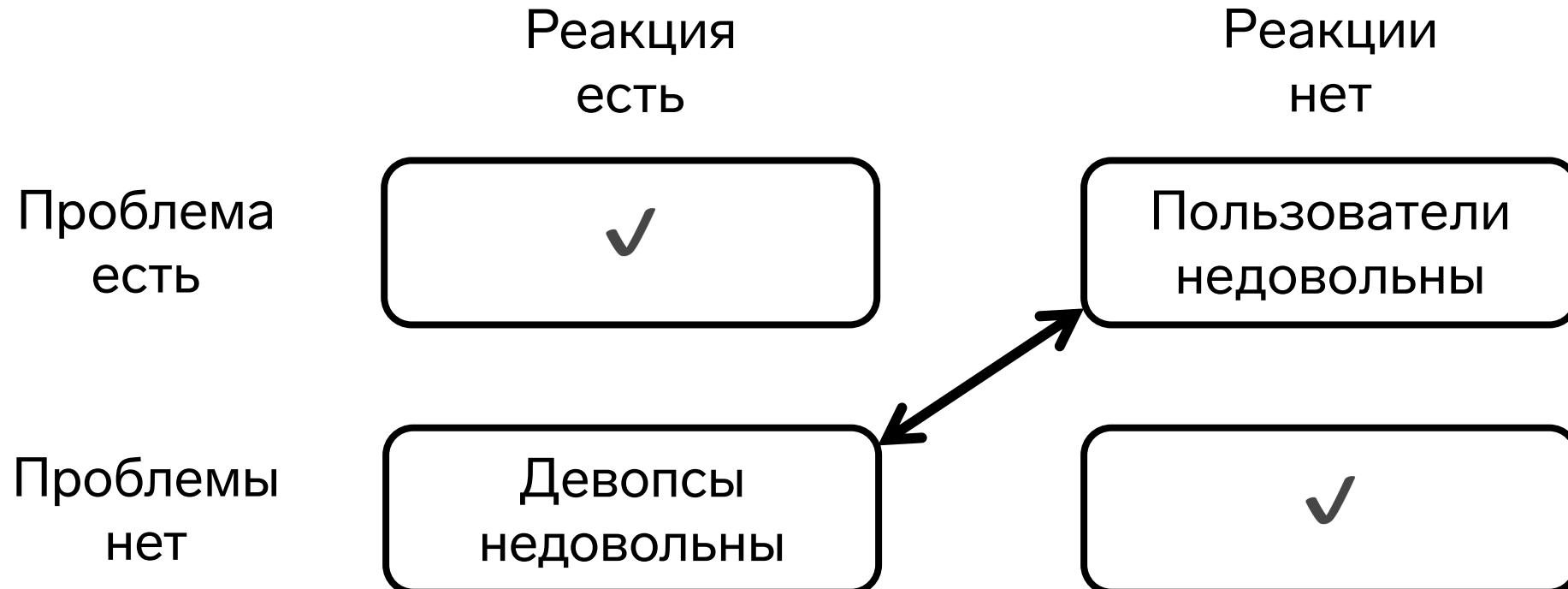
Проблемы  
нет



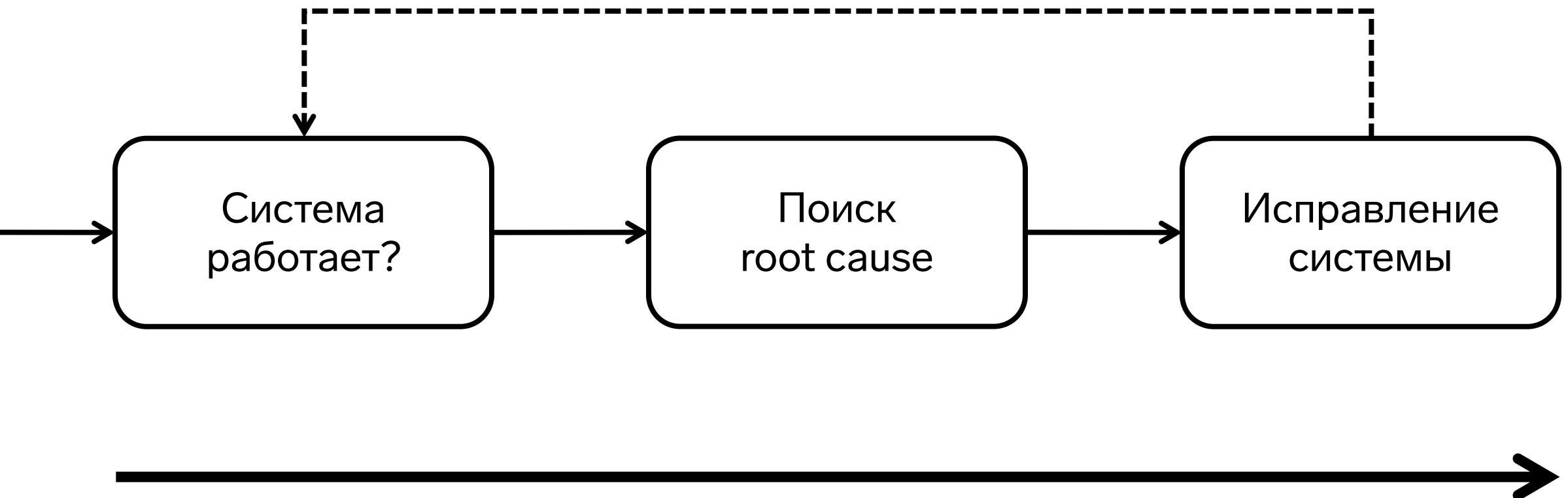
# Когда и как узнавать, что система сломалась?



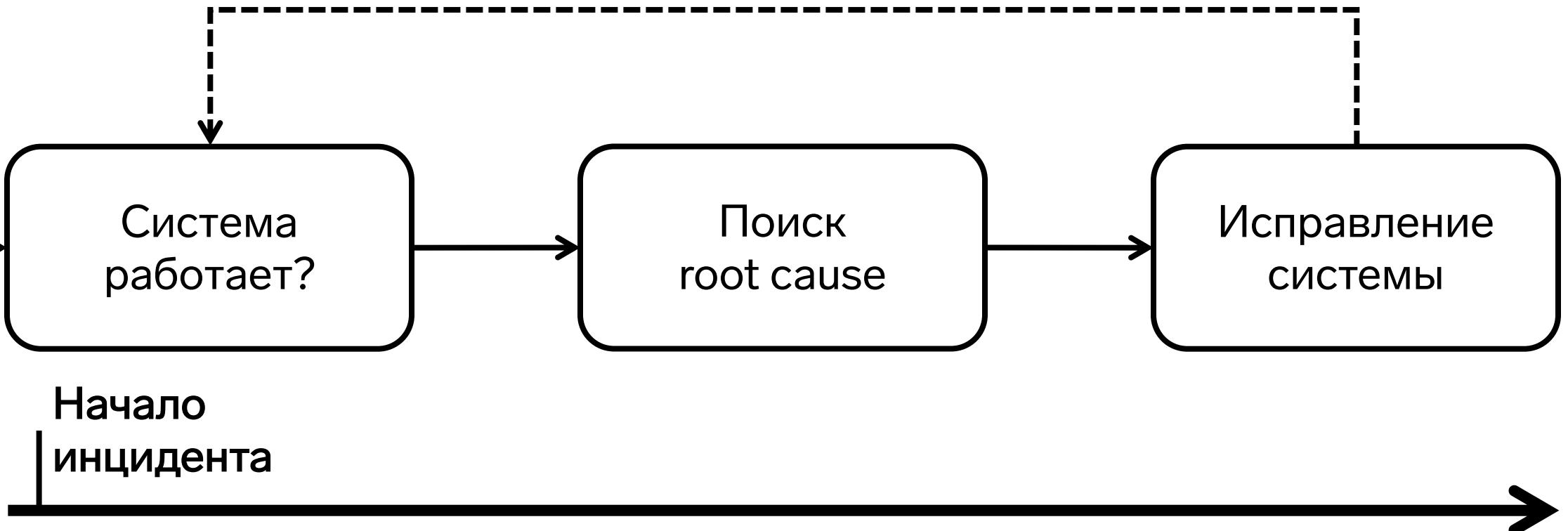
# Когда и как узнавать, что система сломалась?



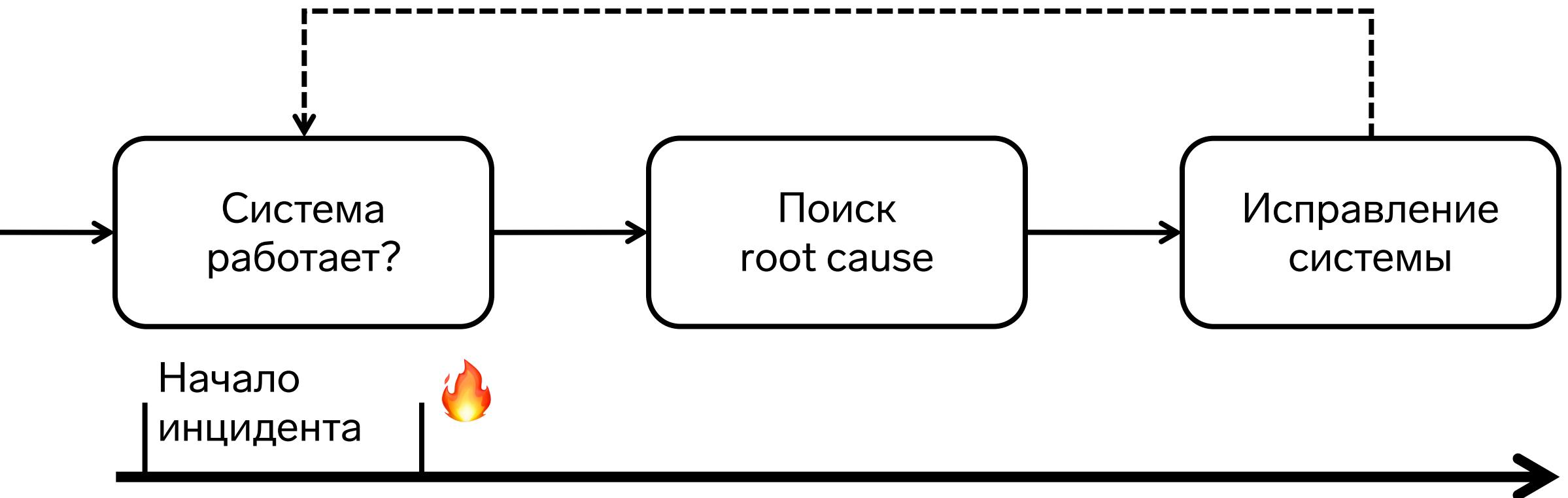
# Когда и как узнавать, что система сломалась?



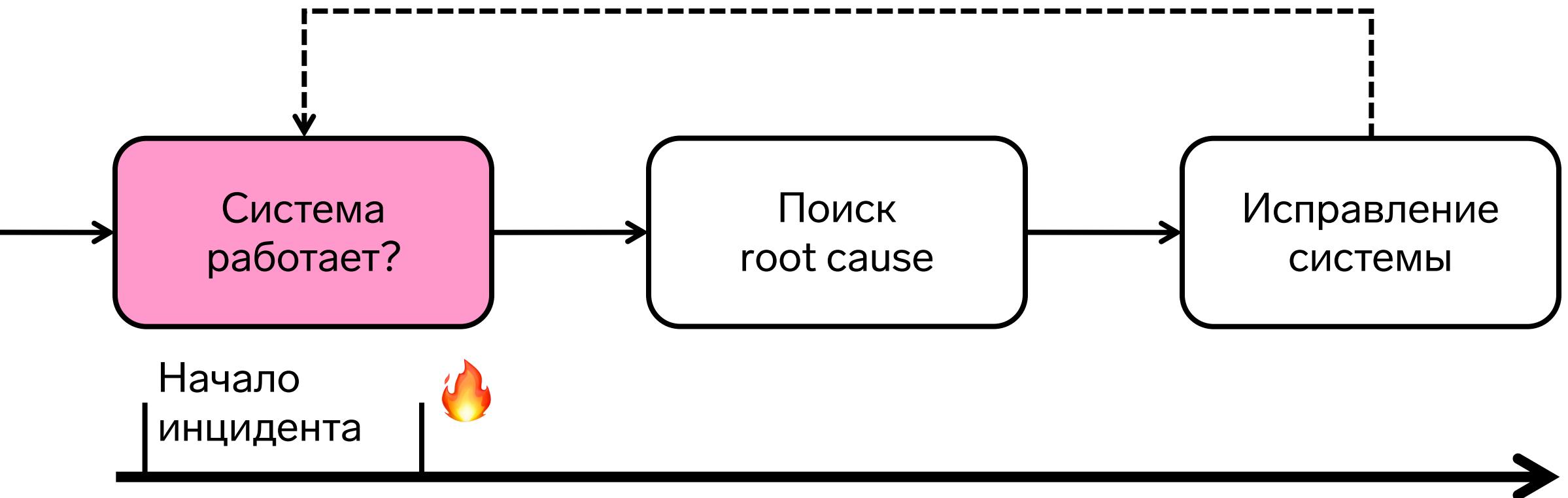
# Когда и как узнавать, что система сломалась?



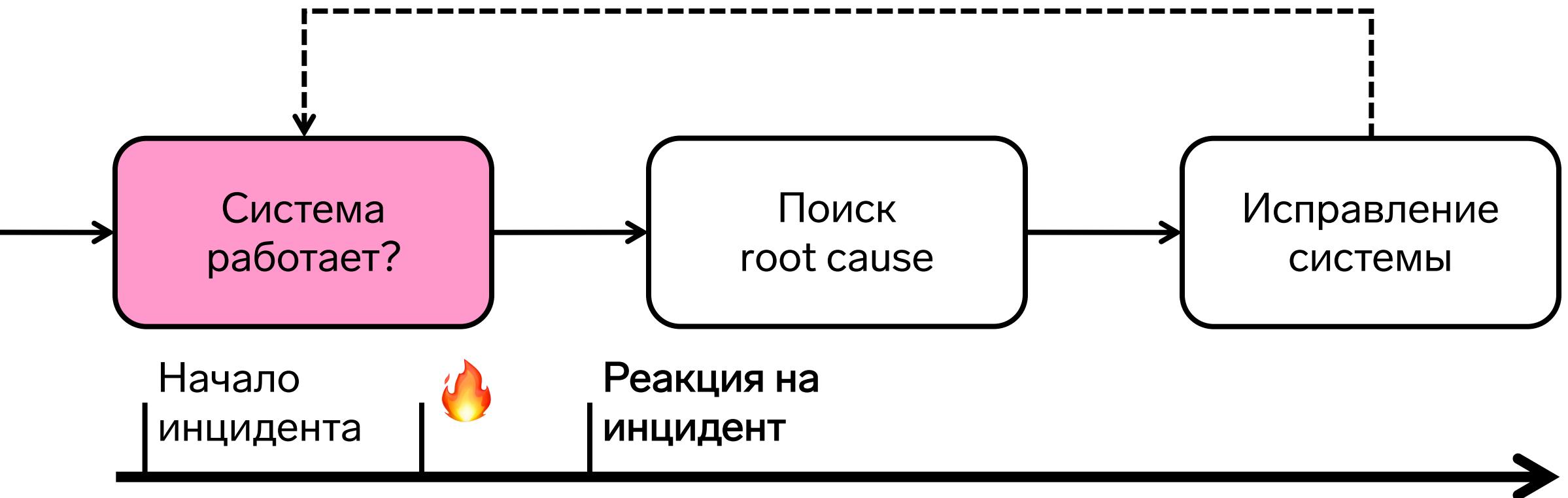
# Когда и как узнавать, что система сломалась?



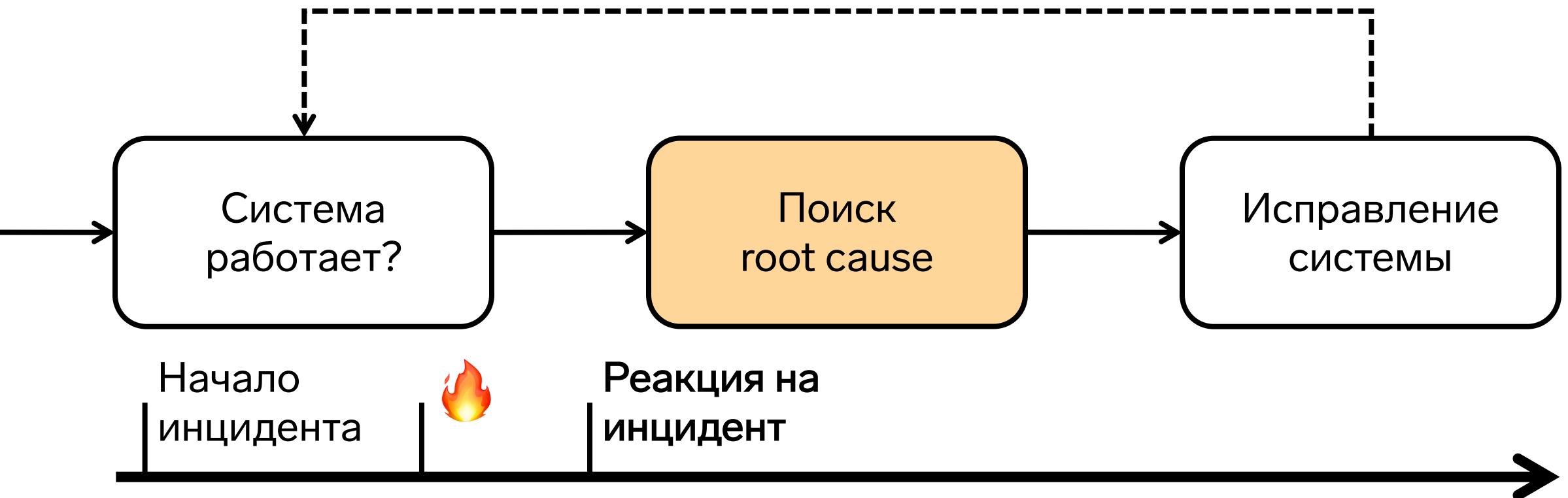
# Когда и как узнавать, что система сломалась?



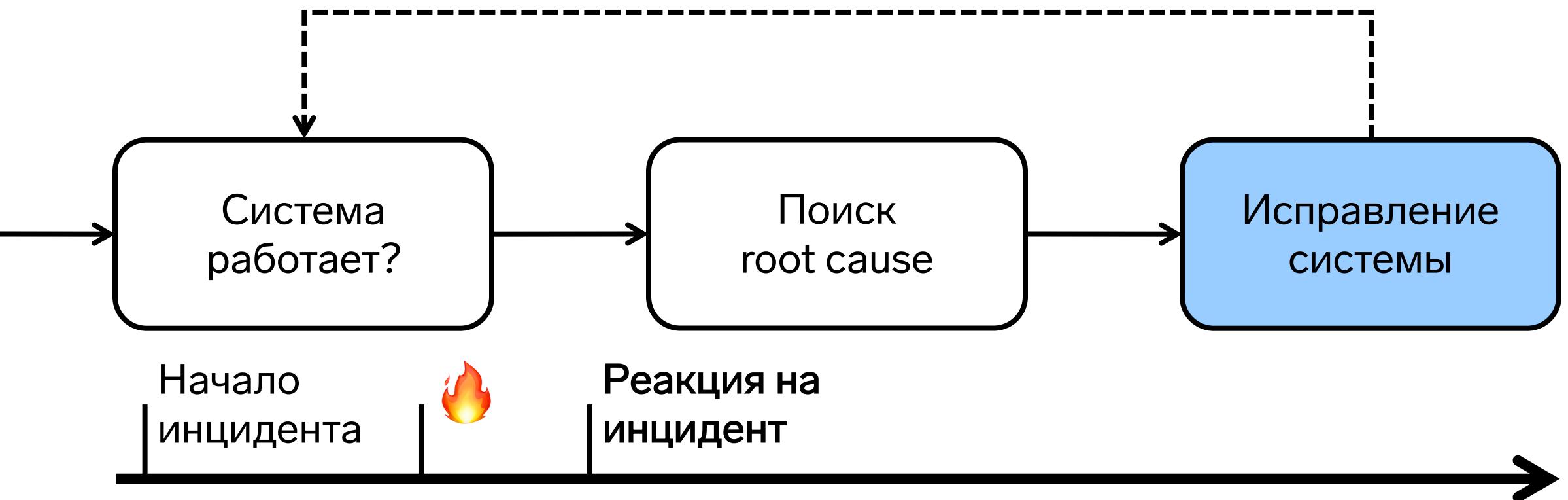
# Когда и как узнавать, что система сломалась?



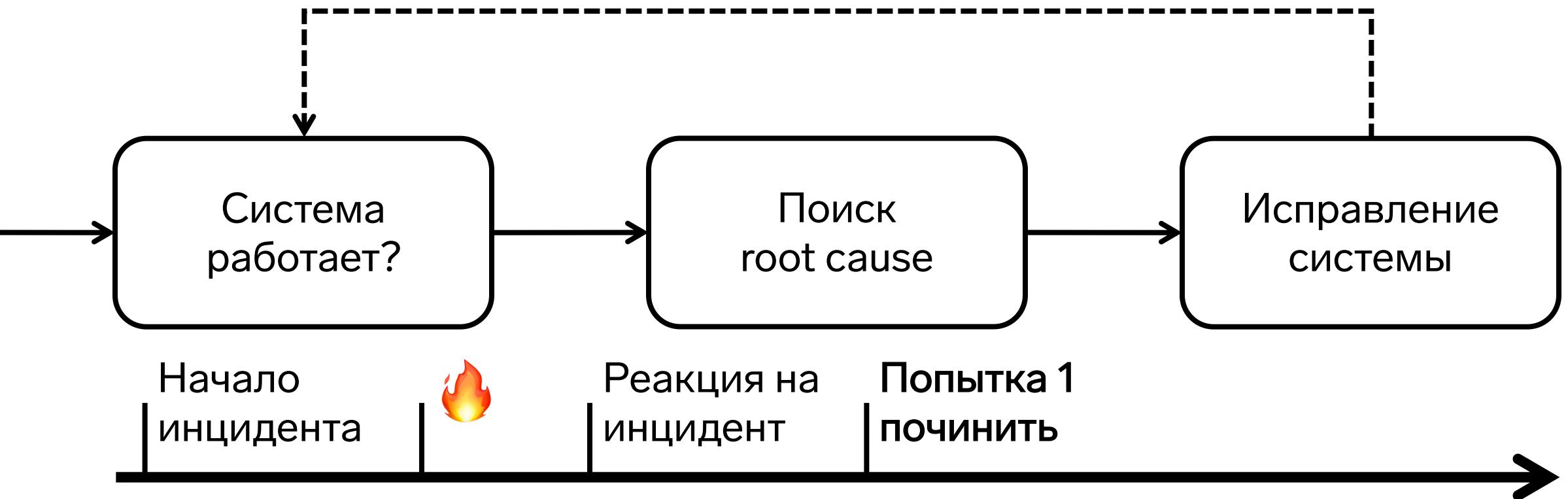
# Когда и как узнавать, что система сломалась?



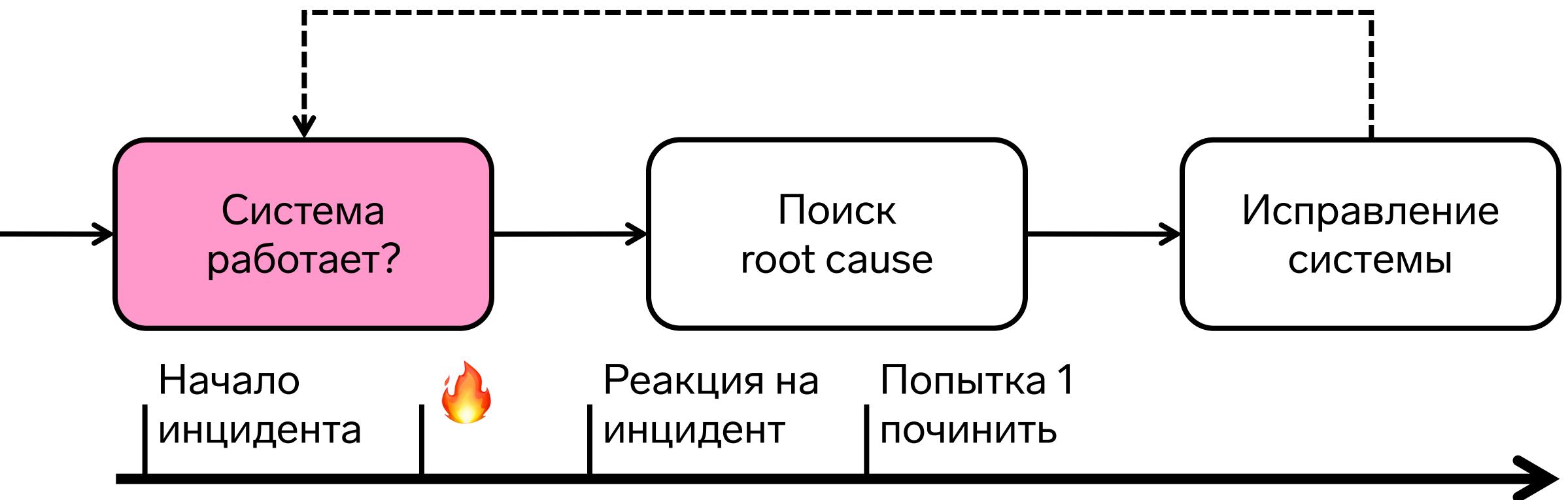
# Когда и как узнавать, что система сломалась?



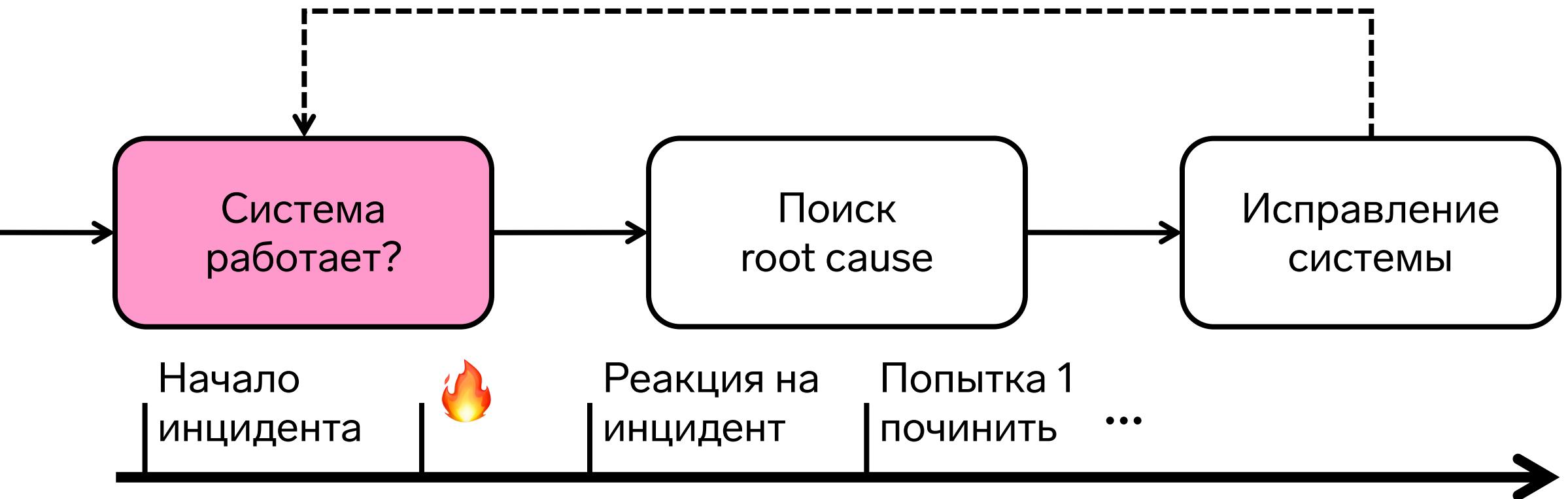
# Когда и как узнавать, что система сломалась?



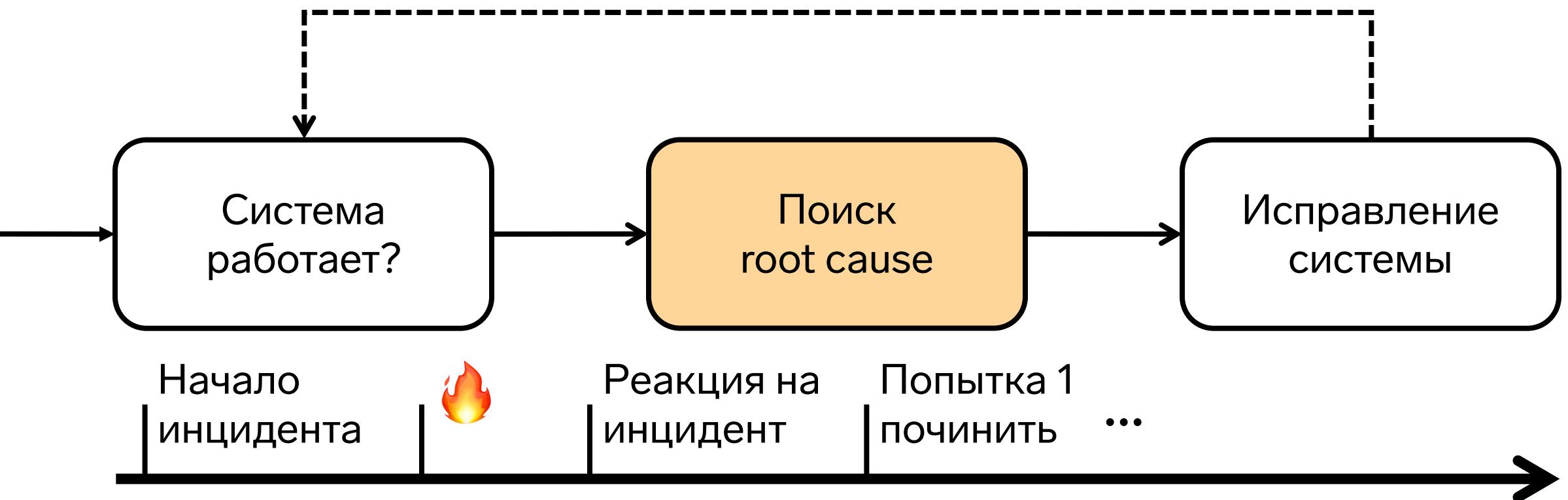
# Когда и как узнавать, что система сломалась?



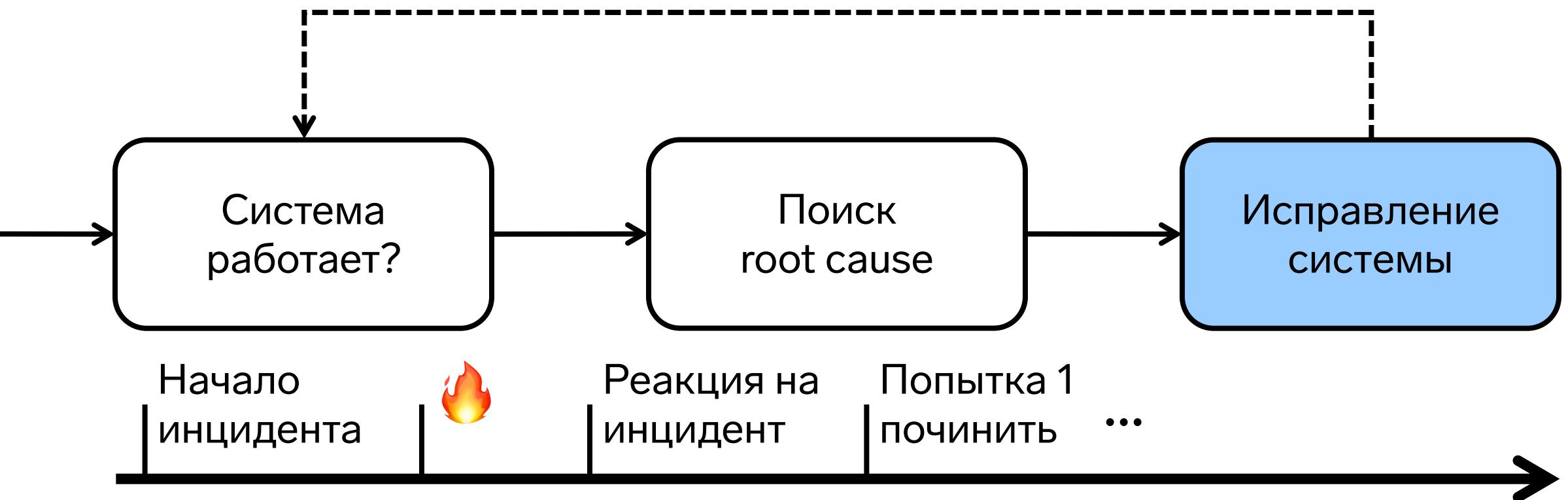
# Когда и как узнавать, что система сломалась?



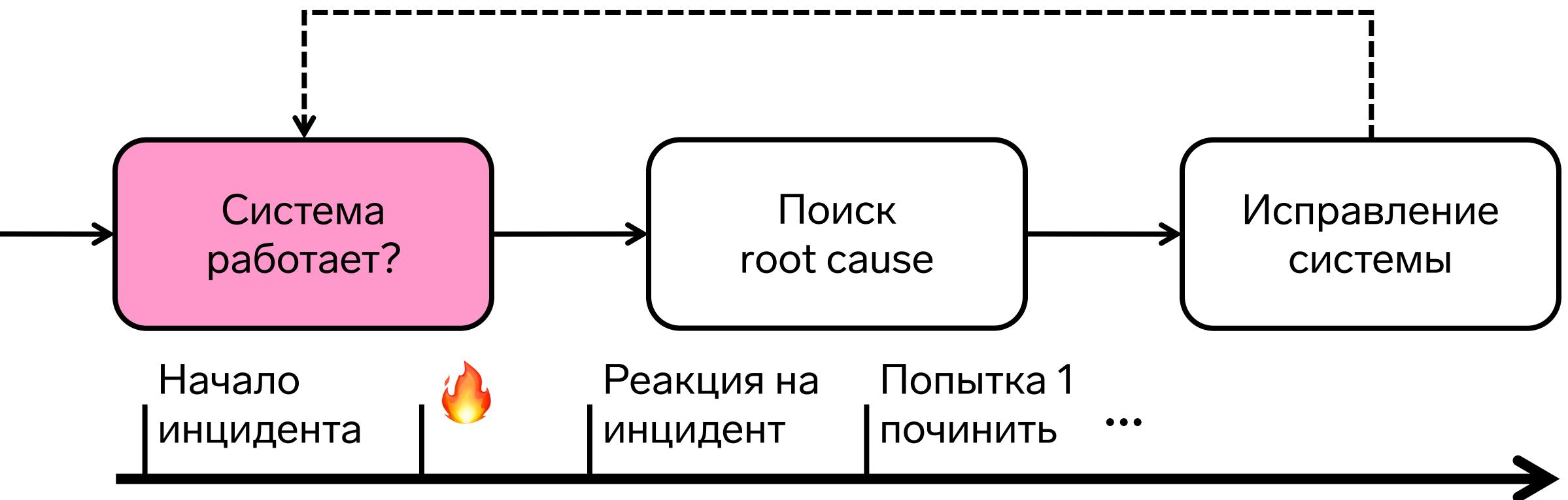
# Когда и как узнавать, что система сломалась?



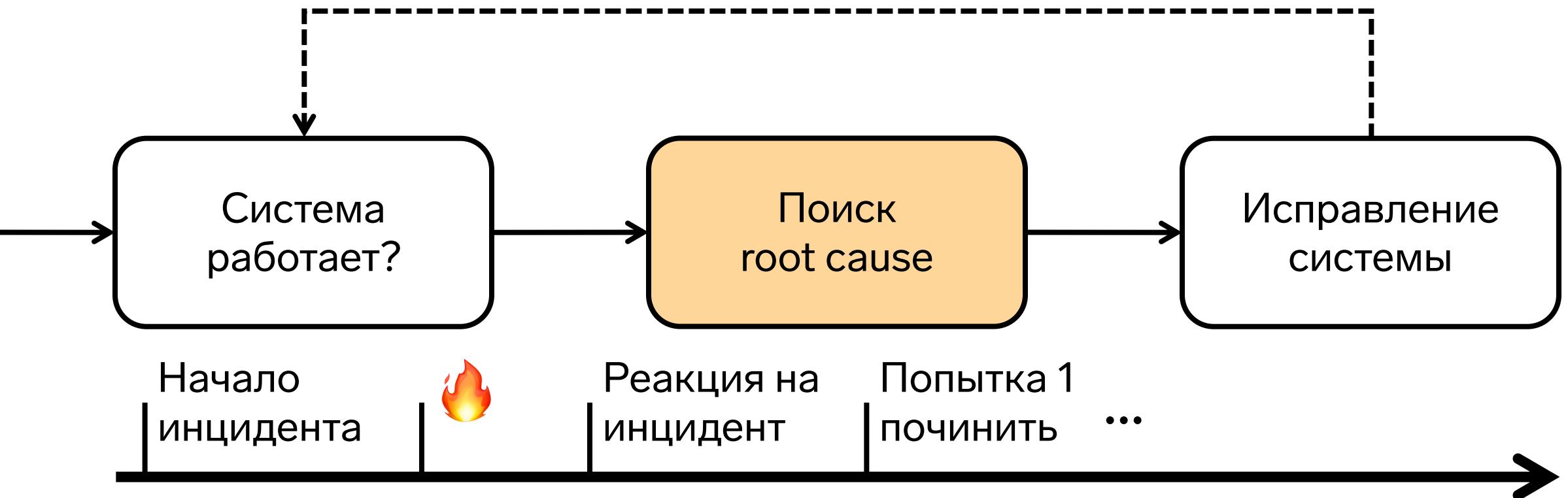
# Когда и как узнавать, что система сломалась?



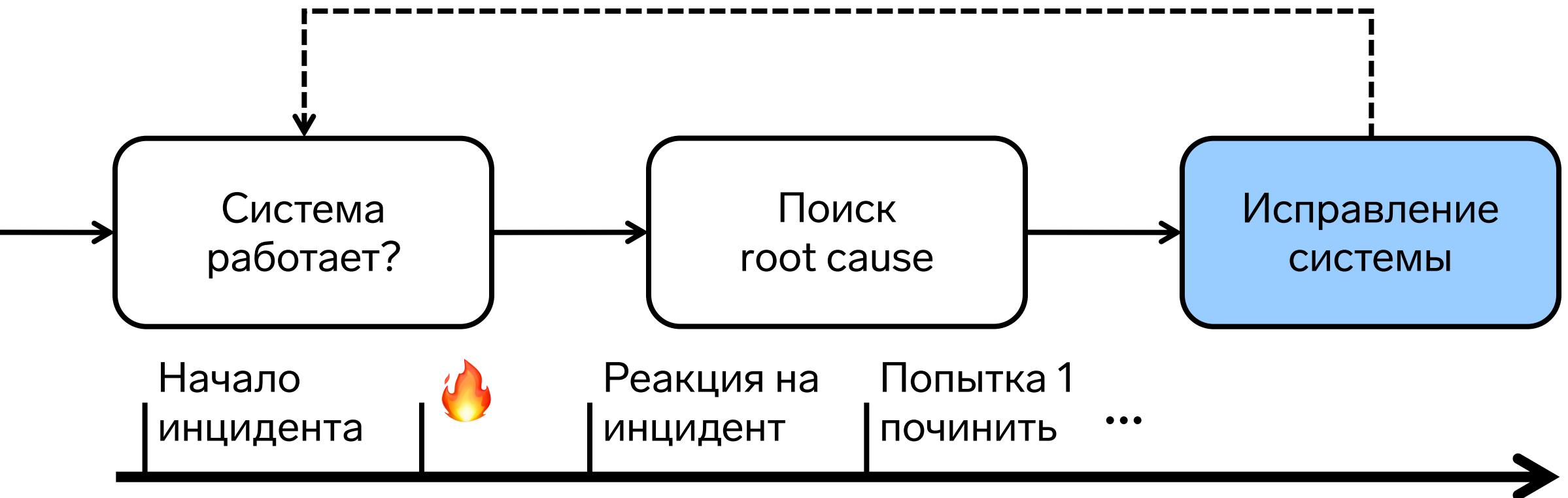
# Когда и как узнавать, что система сломалась?



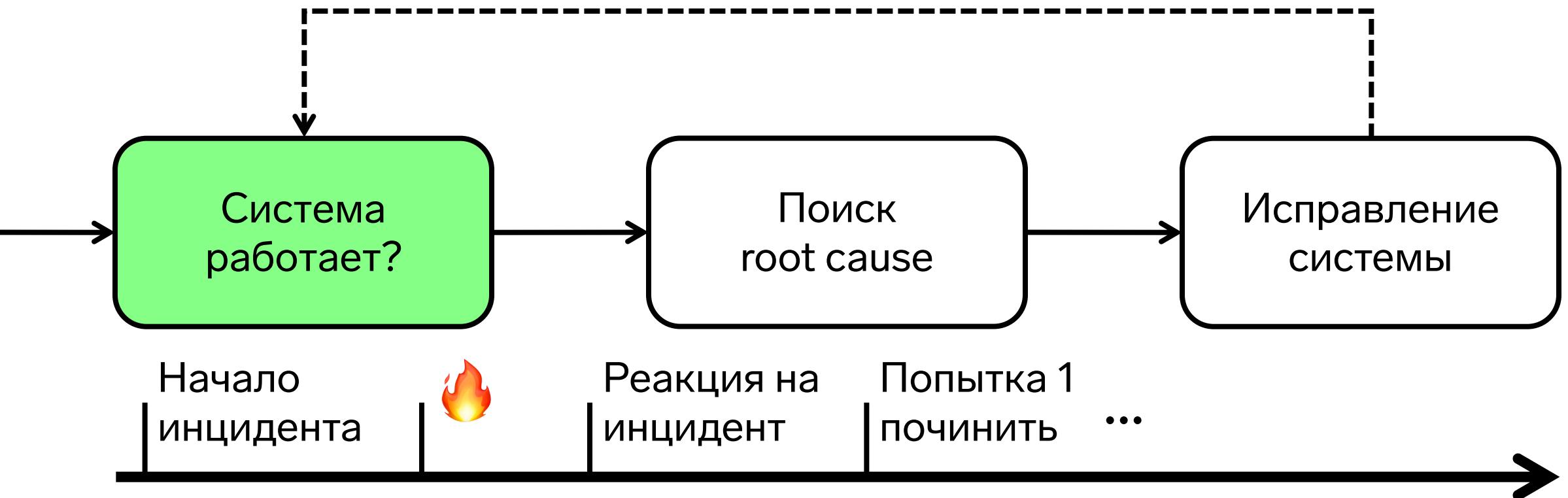
# Когда и как узнавать, что система сломалась?



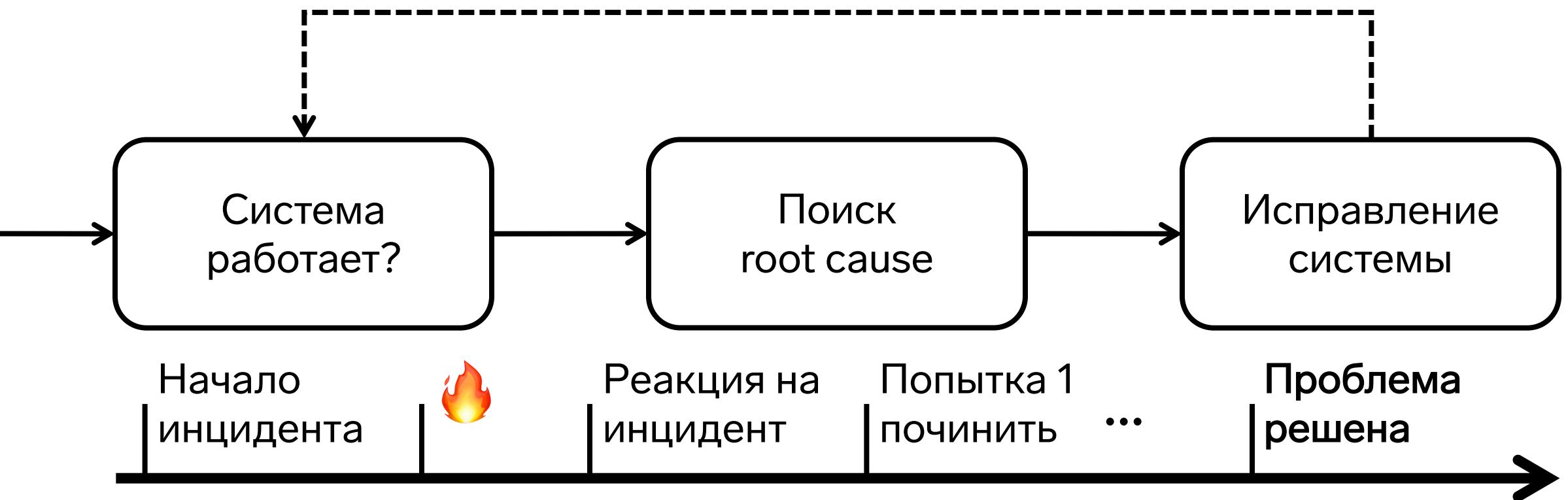
# Когда и как узнавать, что система сломалась?



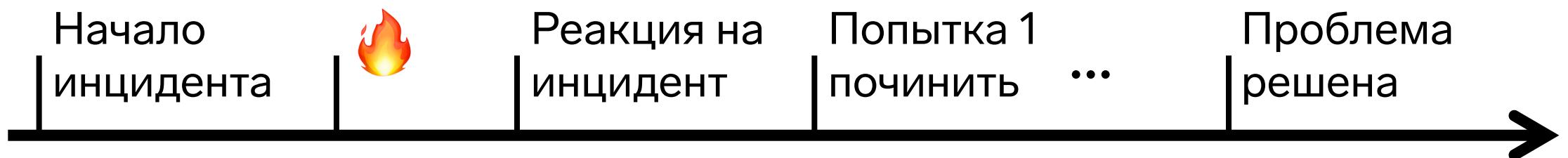
# Когда и как узнавать, что система сломалась?



# Когда и как узнавать, что система сломалась?

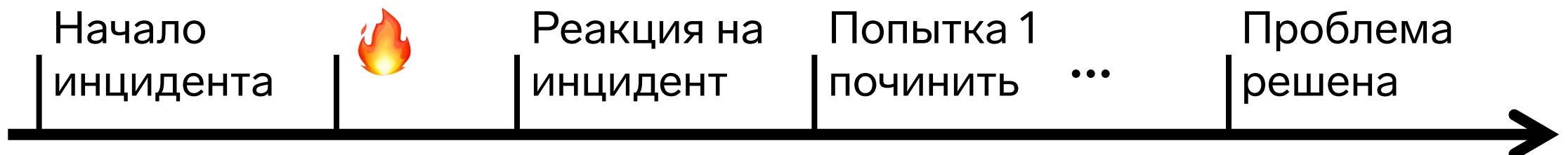


# Когда и как узнавать, что система сломалась?



# Когда и как узнавать, что система сломалась?

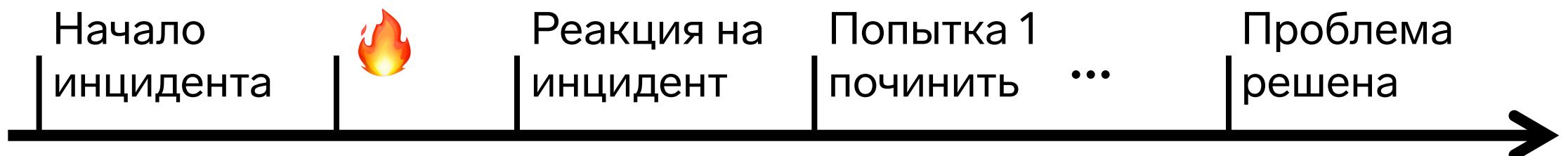
SLO – Service Level Objective



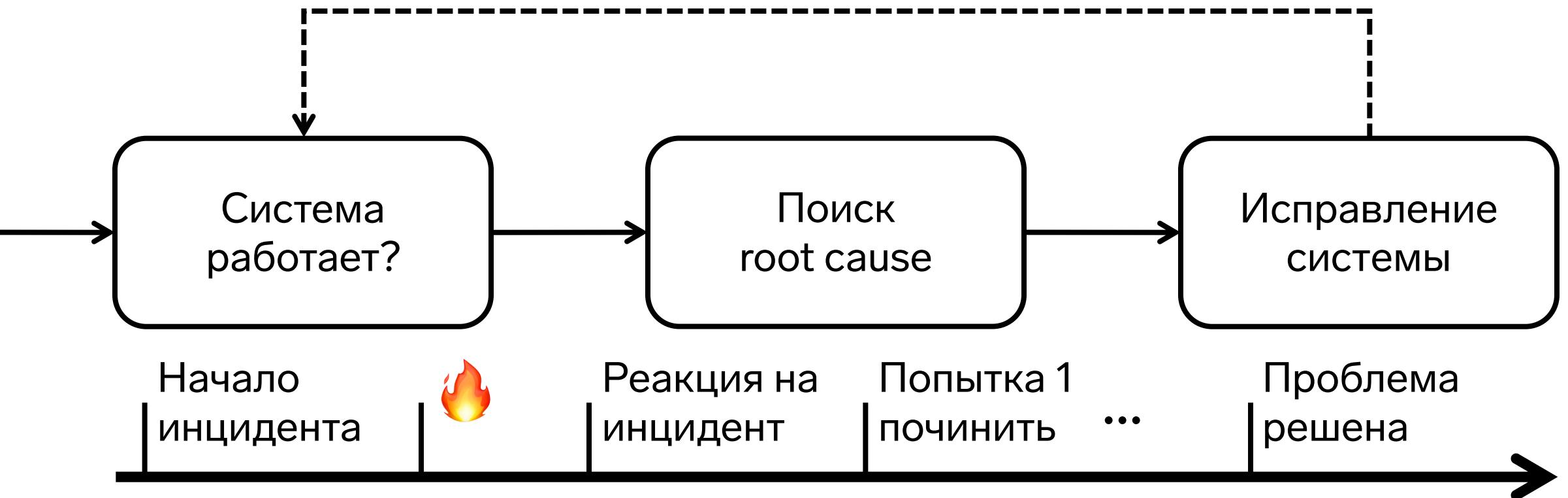
# Когда и как узнавать, что система сломалась?

SLO – Service Level Objective

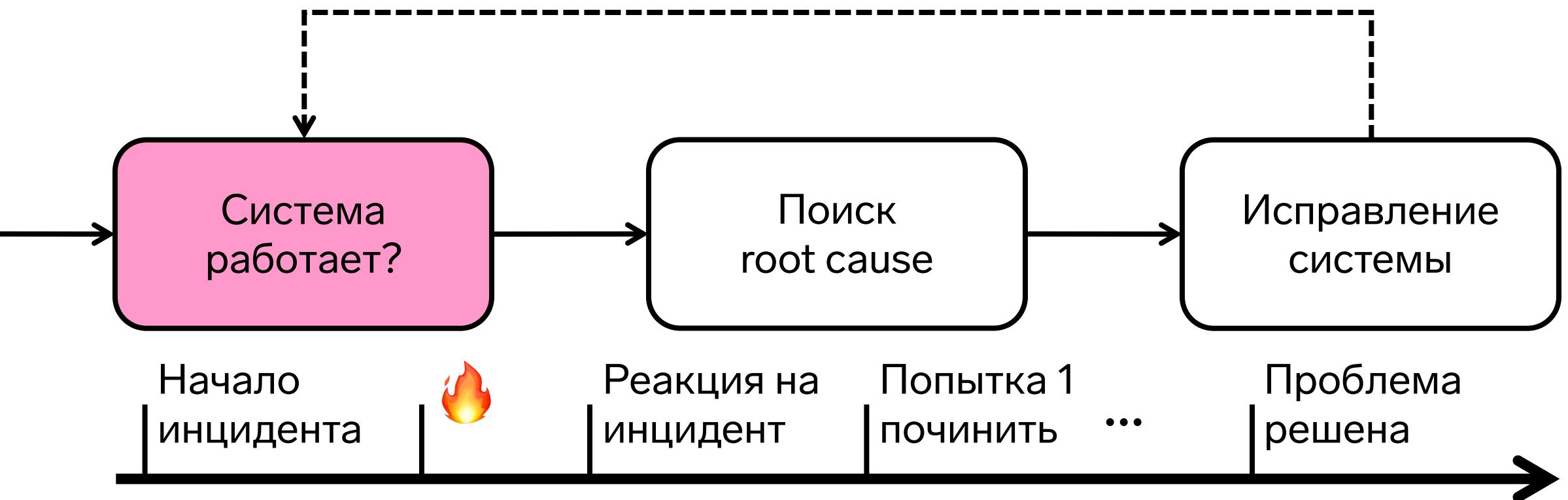
- Критичность
- Оперативность



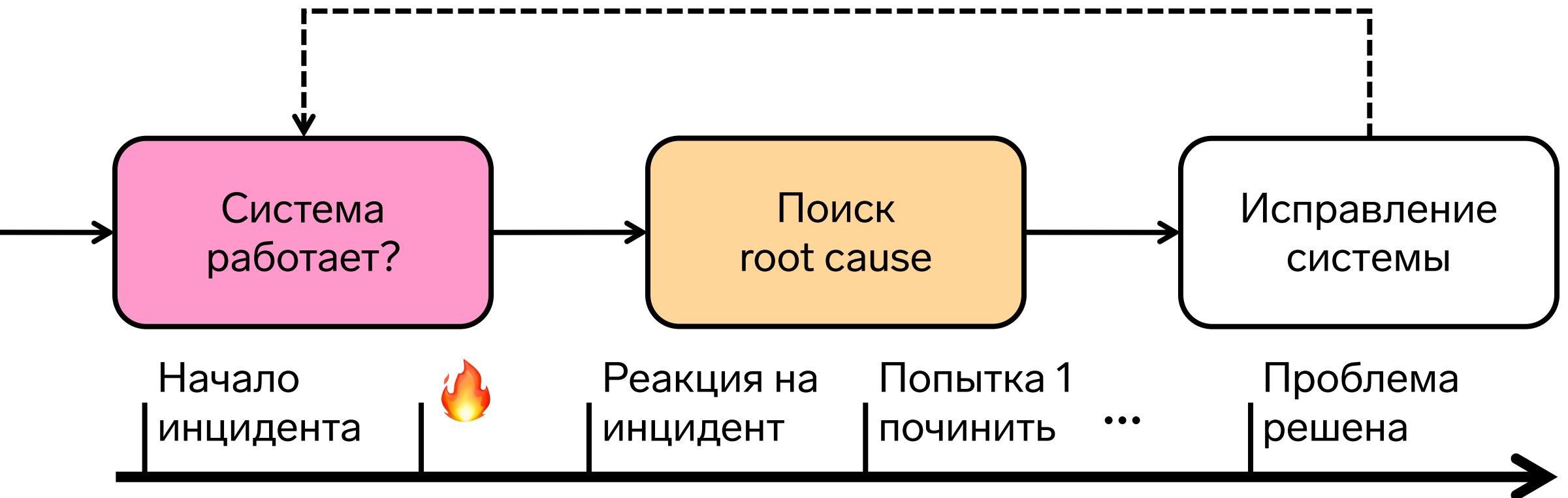
# Когда и как узнавать, что система сломалась?



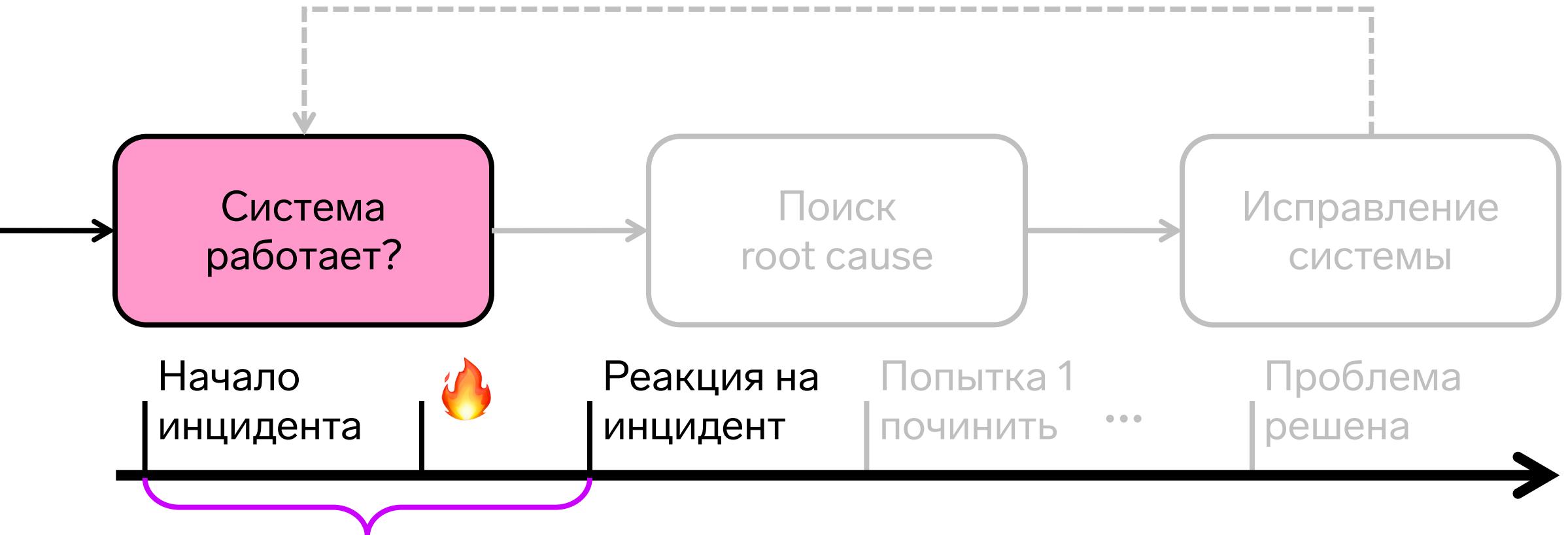
# Когда и как узнавать, что система сломалась?



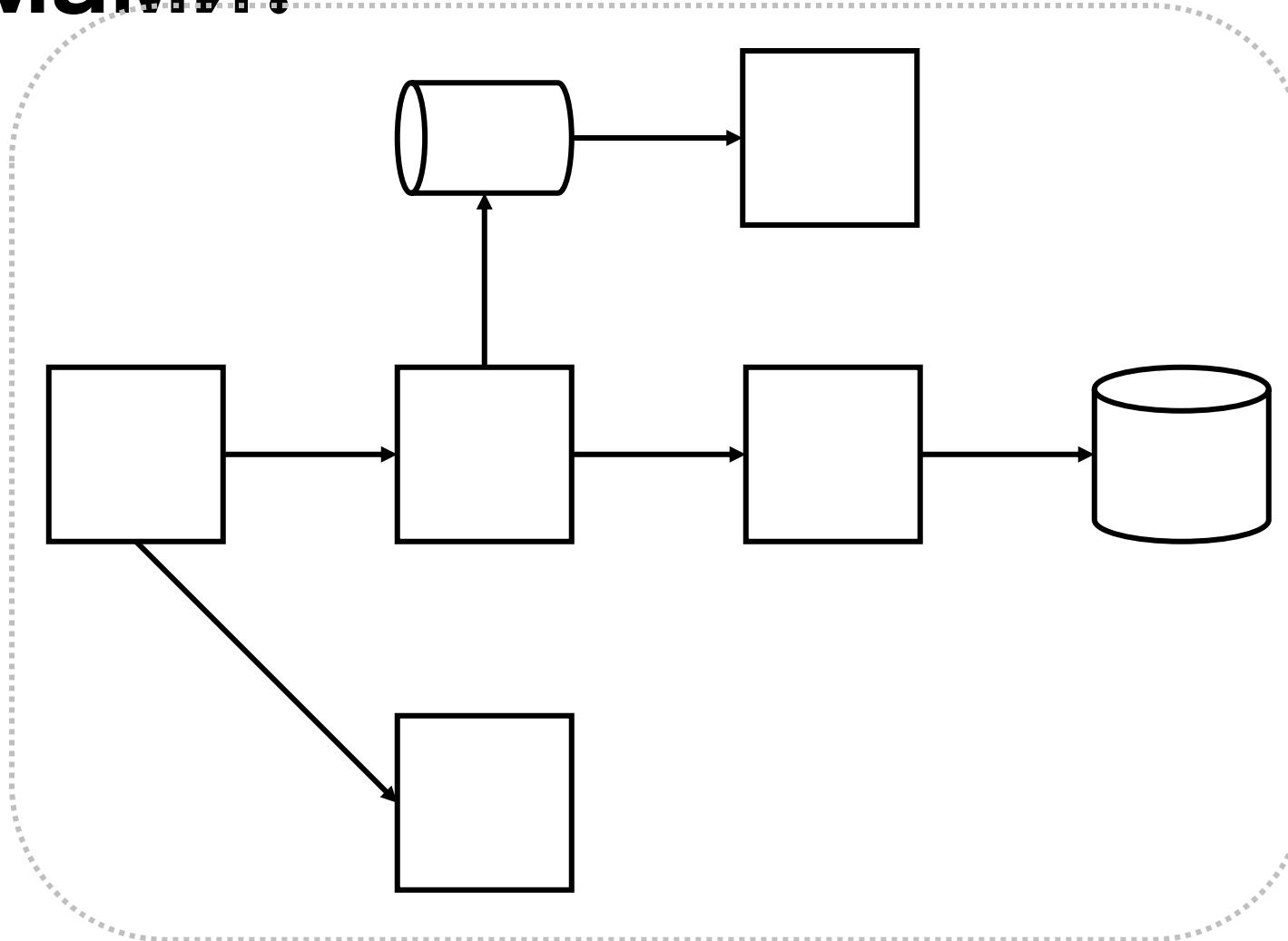
# Когда и как узнавать, что система сломалась?



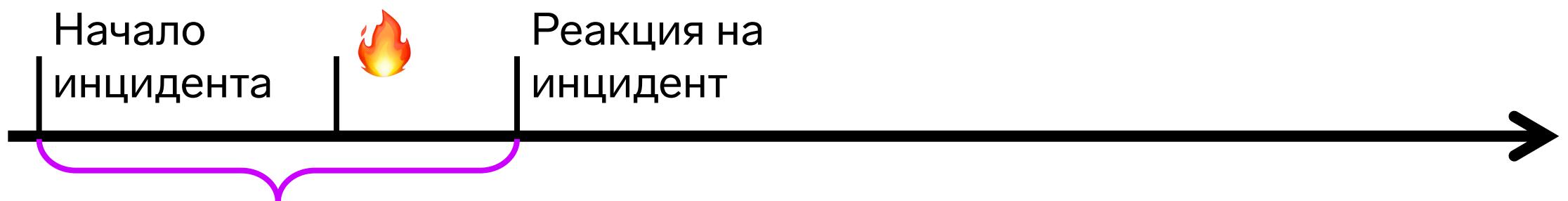
# Когда и как узнавать, что система сломалась?



# Как это связано с распределёнными системами?

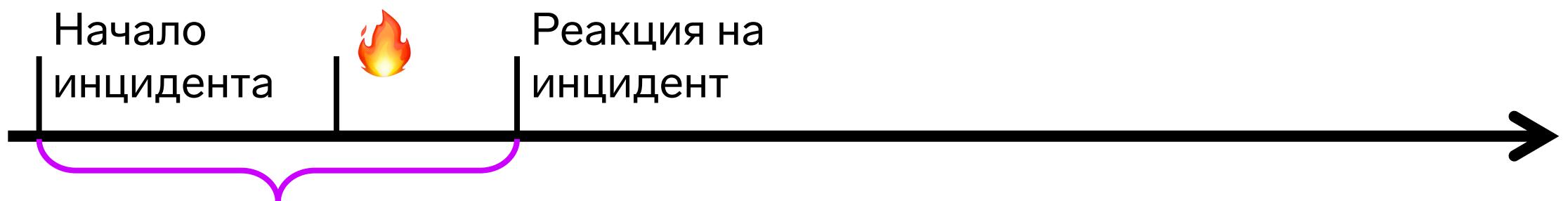


# Активный мониторинг



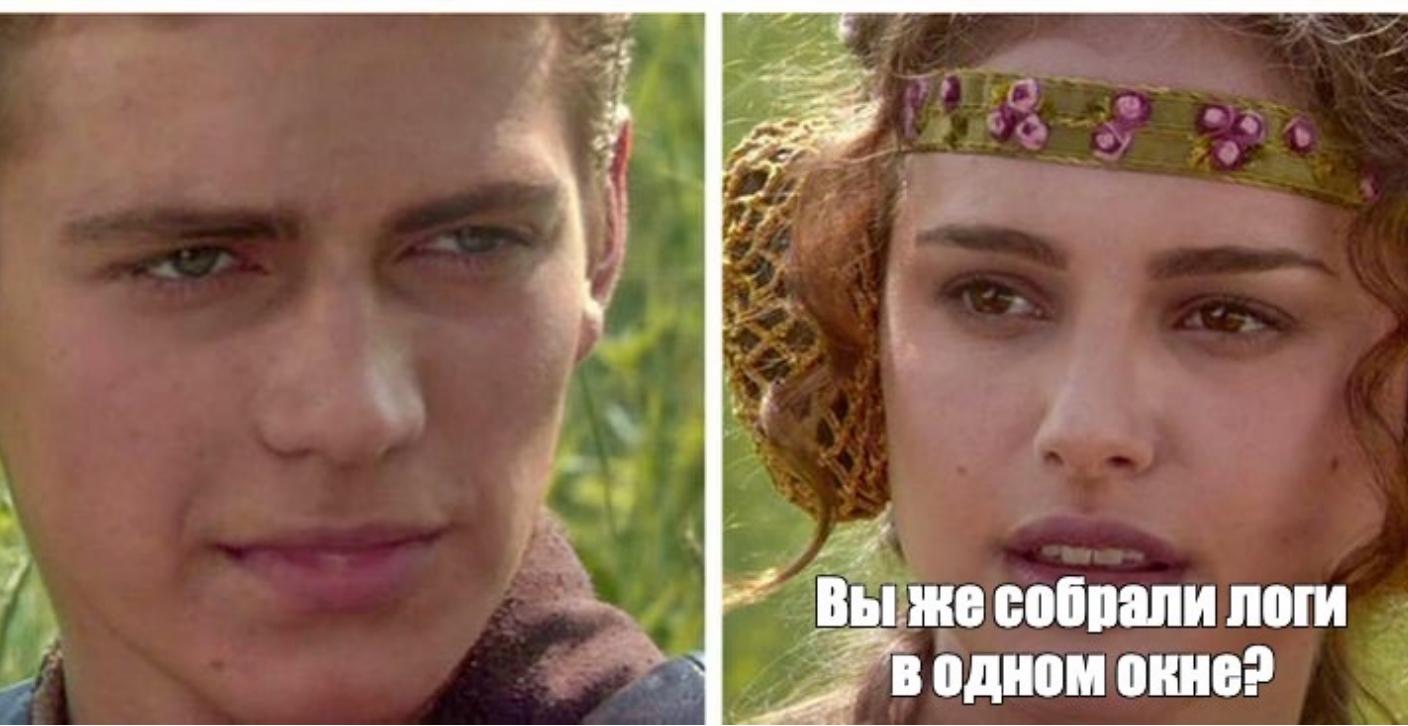
# Активный мониторинг

Подход №1: **смотрим глазками в логи**



# Активи

Подход №1





# Активный мониторинг

Подход №1: **смотрим глазками в логи**

В чём подвох?

# Активный мониторинг

Подход №1: **смотрим глазками в логи**

Реакция  
есть

Проблема  
есть



Реакции  
нет

Пользователи  
недовольны

Проблемы  
нет

Девопсы  
недовольны



# Активный мониторинг

Подход №1: **смотрим глазками в логи**

Реакция  
есть

Проблема  
есть

Реакции  
нет

Пользователи  
недовольны

Проблемы  
нет

Девопсы  
недовольны

# Активный мониторинг

Подход №2: **смотрим глазками на графики**



# Активный мониторинг

Подход №2: **смотрим глазками на графики**

В чём подвох?

# Активный мониторинг

Подход №2: **смотрим глазками на графики**

В чём подвох?

— Степень покрытия плохо масштабируется

# Активный мониторинг

Подход №2: **смотрим глазками на графики**

В чём подвох?

- Степень покрытия плохо масштабируется
- Страдает чувствительность

# Активный мониторинг

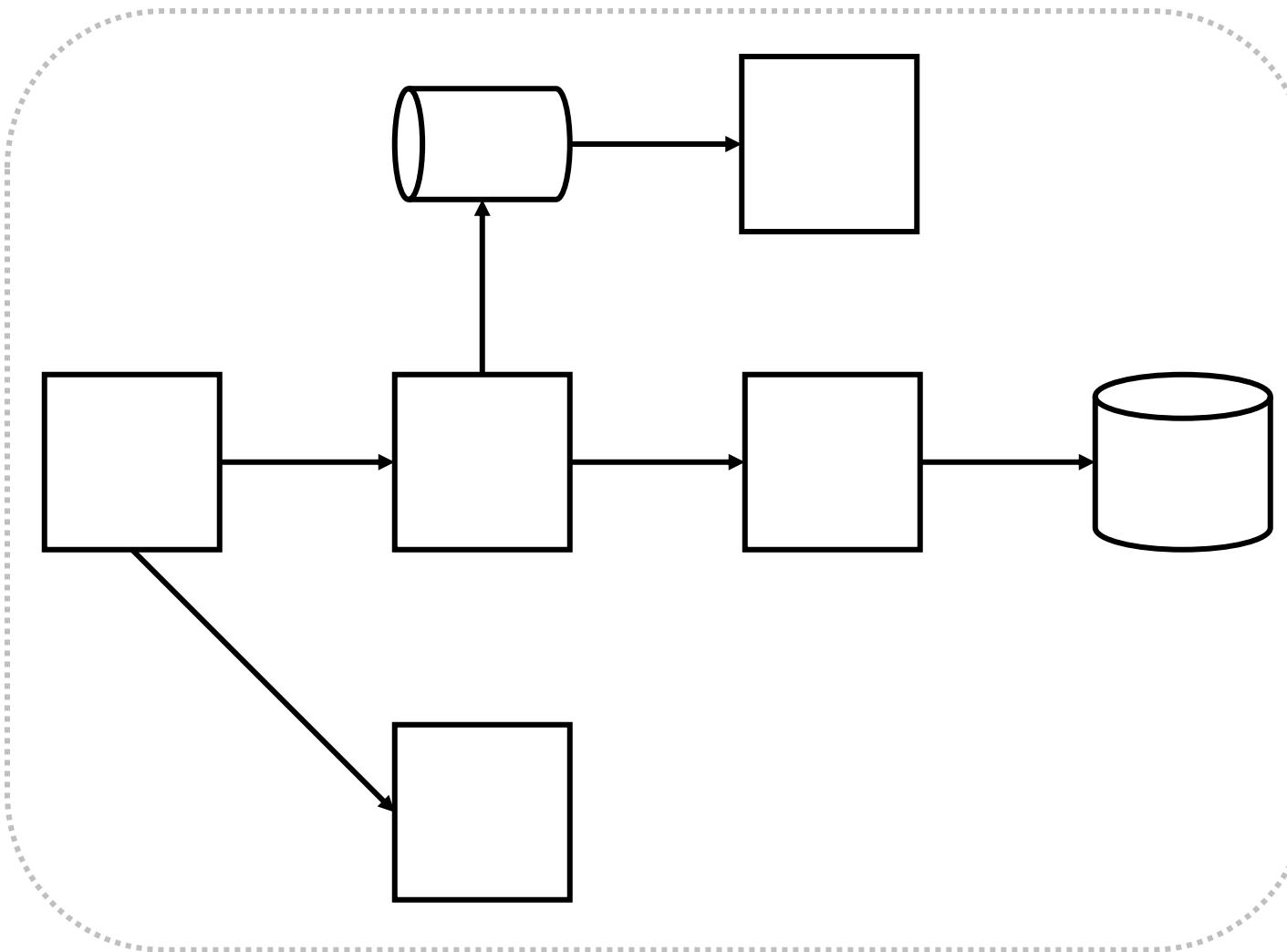
Подход №2: **смотрим глазками на графики**

В чём подвох?

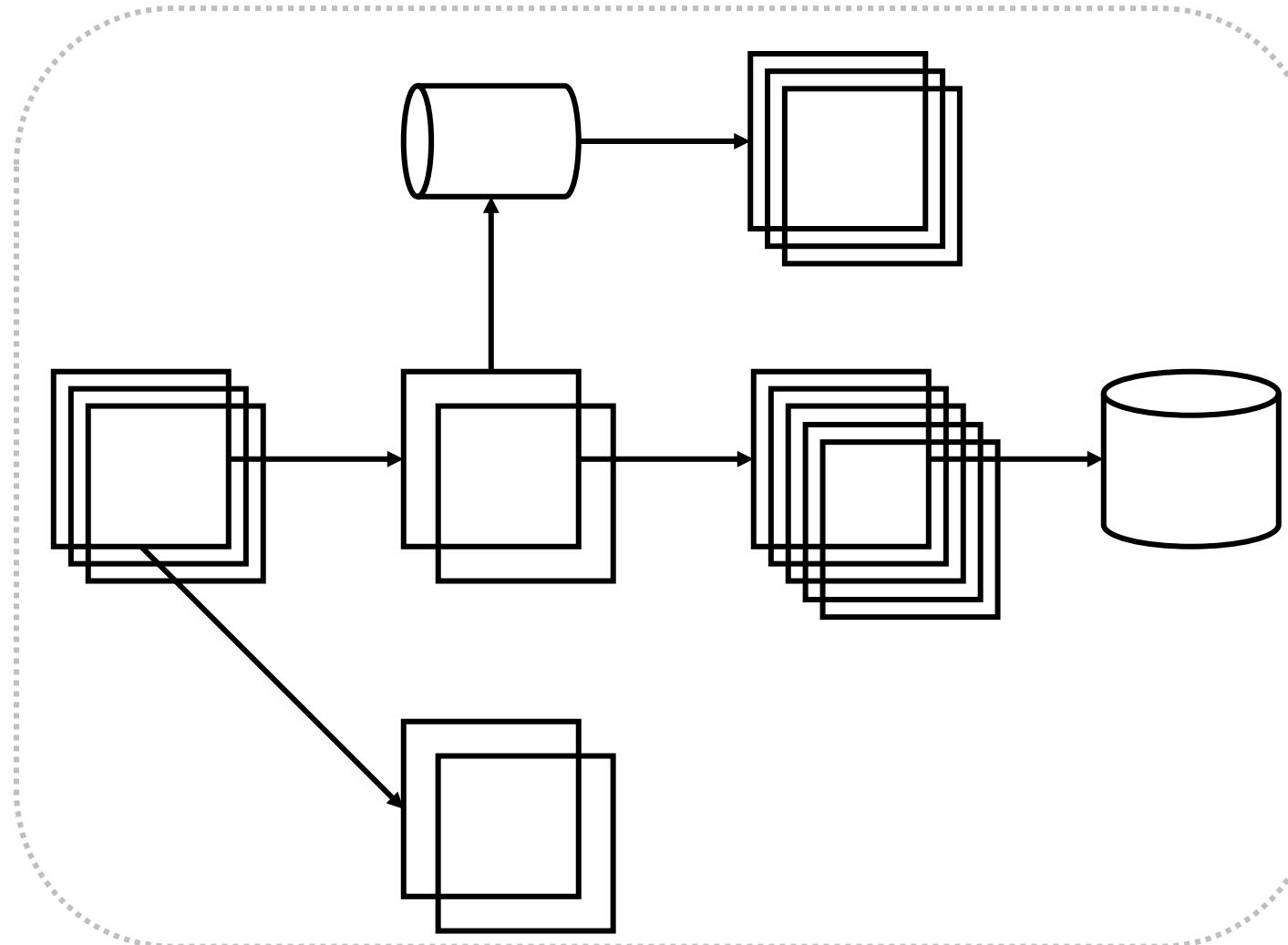
- Степень покрытия плохо масштабируется
- Страдает чувствительность

А что с корректностью **кластерных** метрик?

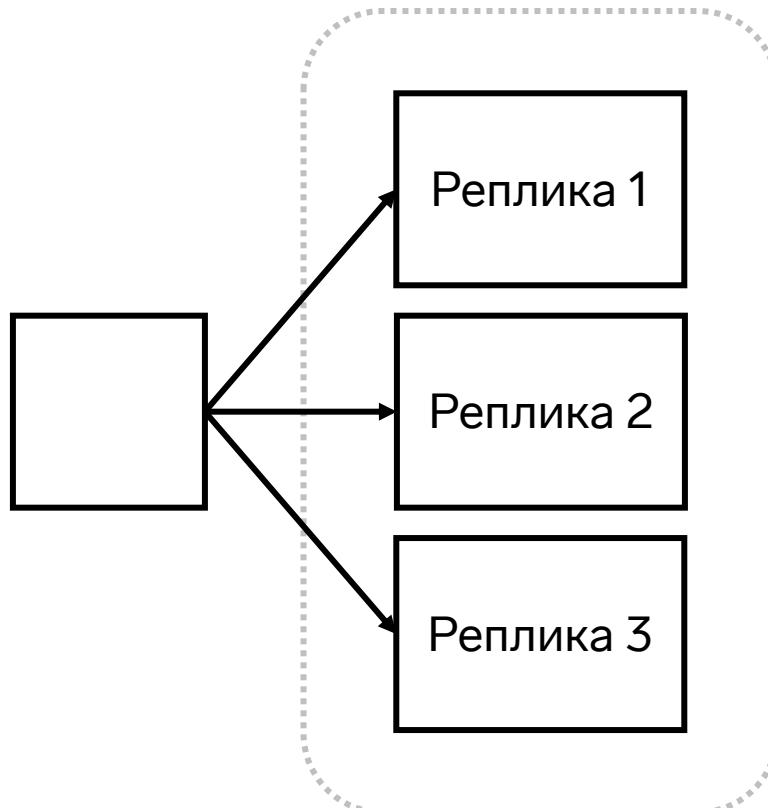
# Кластерные метрики



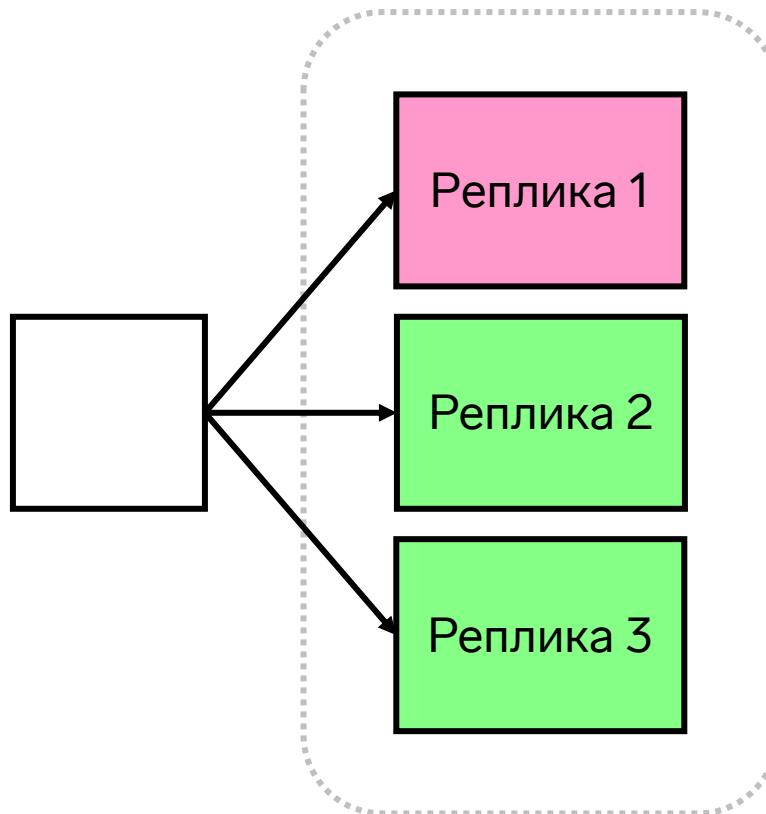
# Кластерные метрики



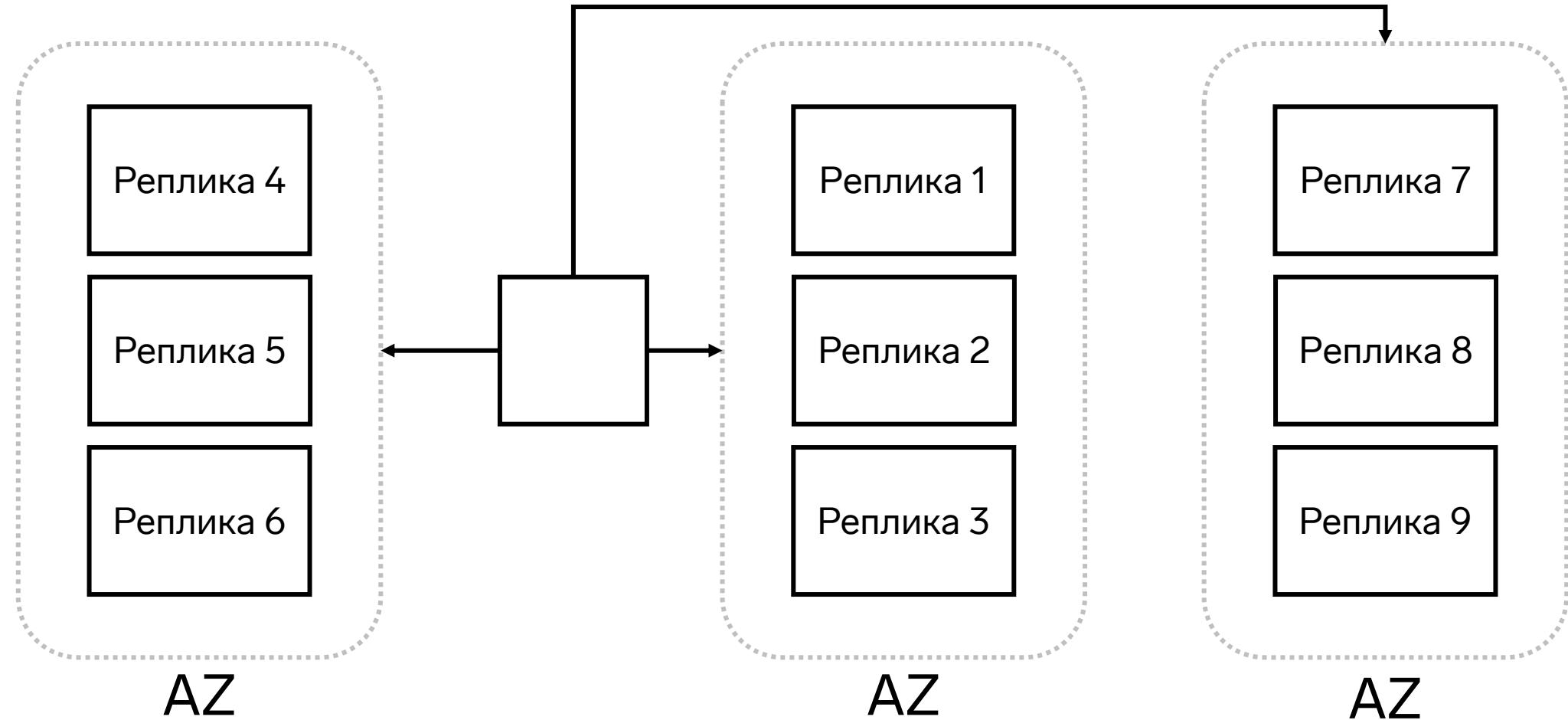
# Кластерные метрики



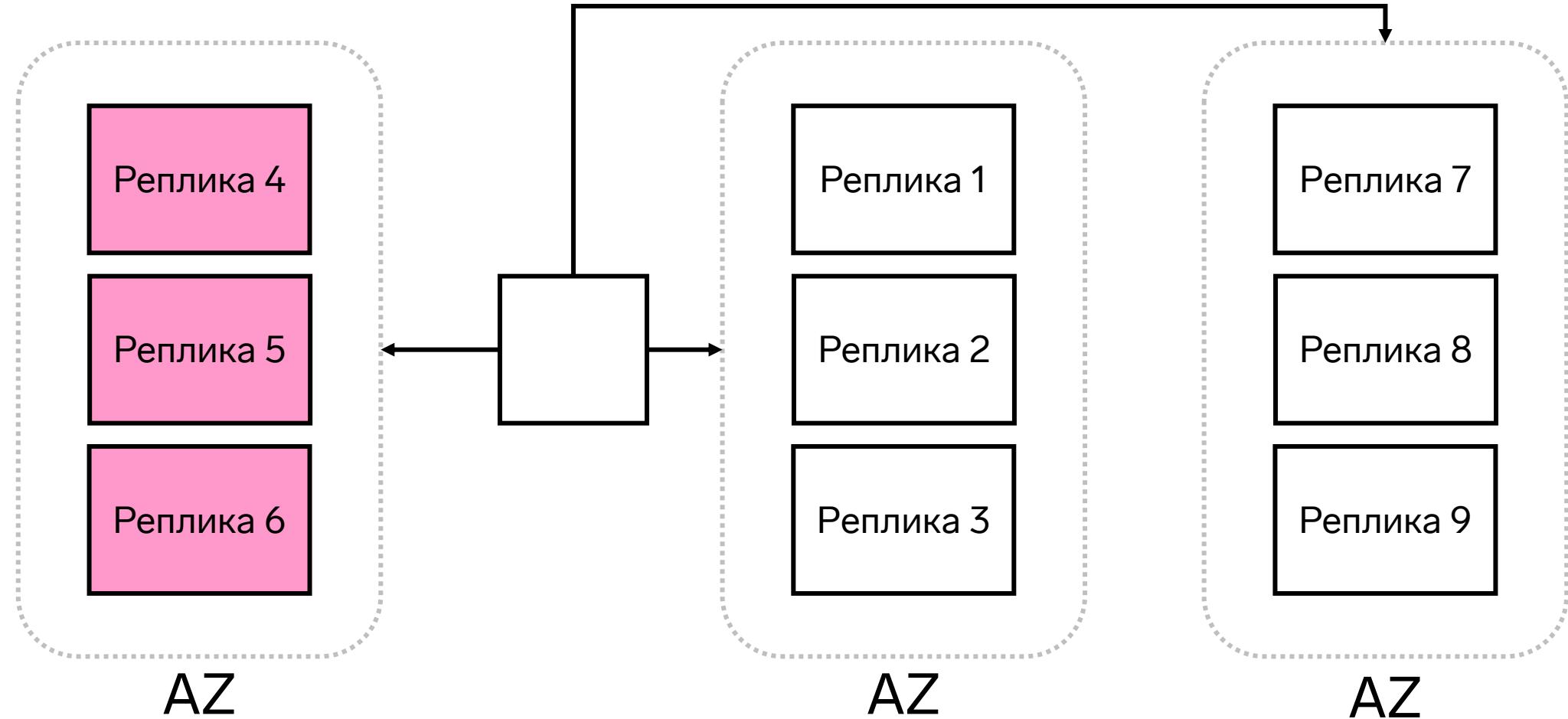
# Кластерные метрики



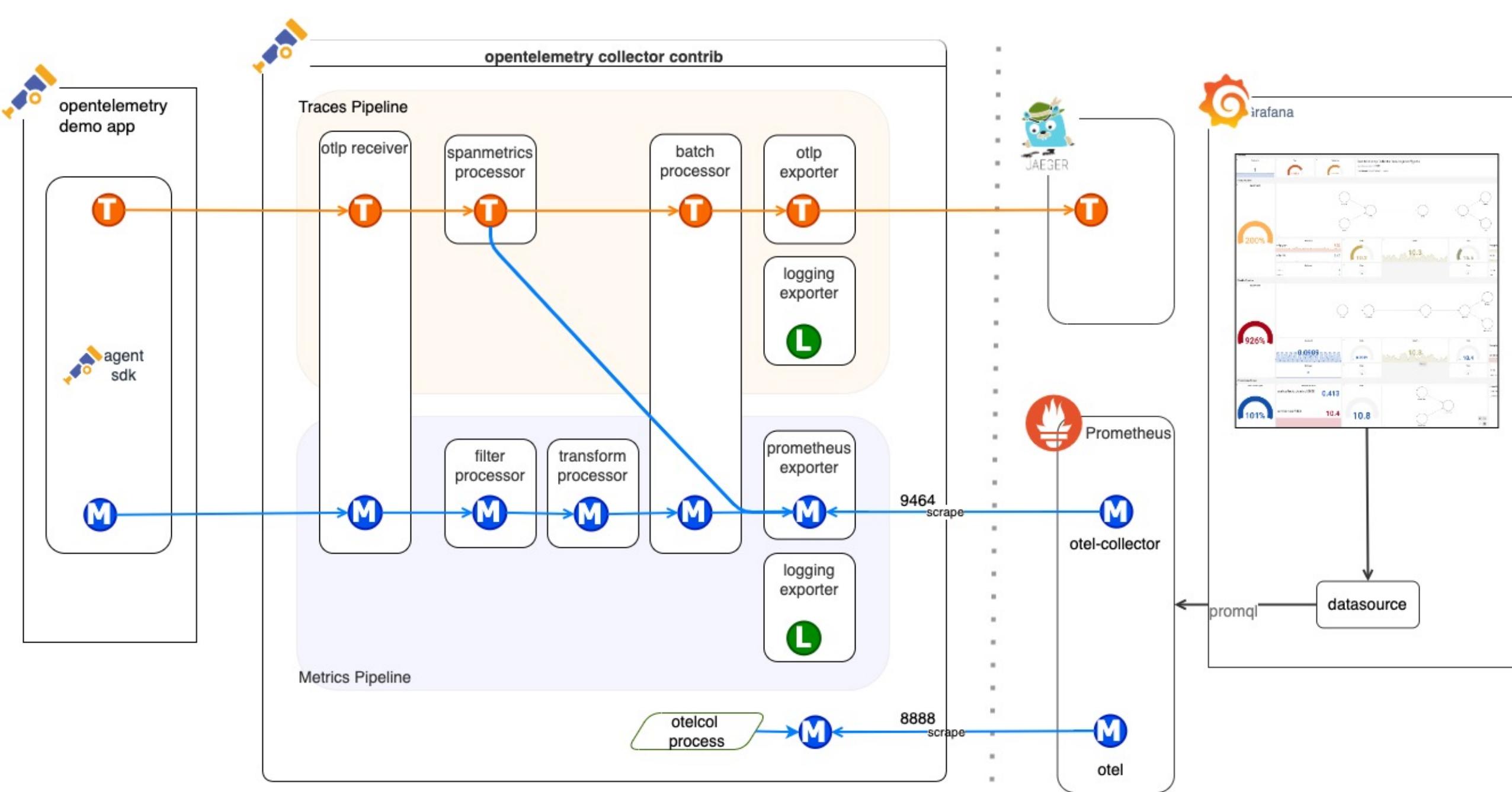
# Кластерные метрики

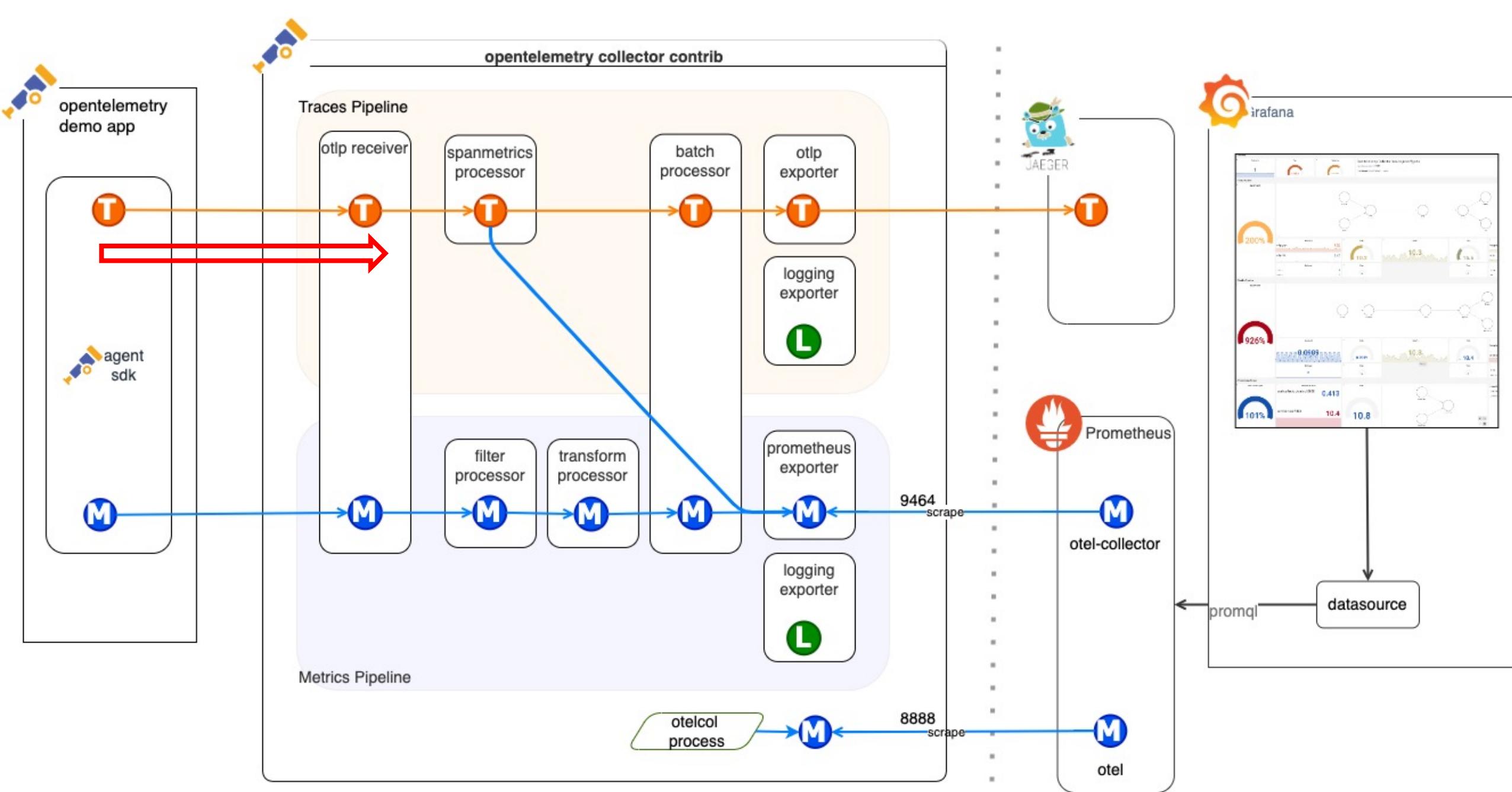


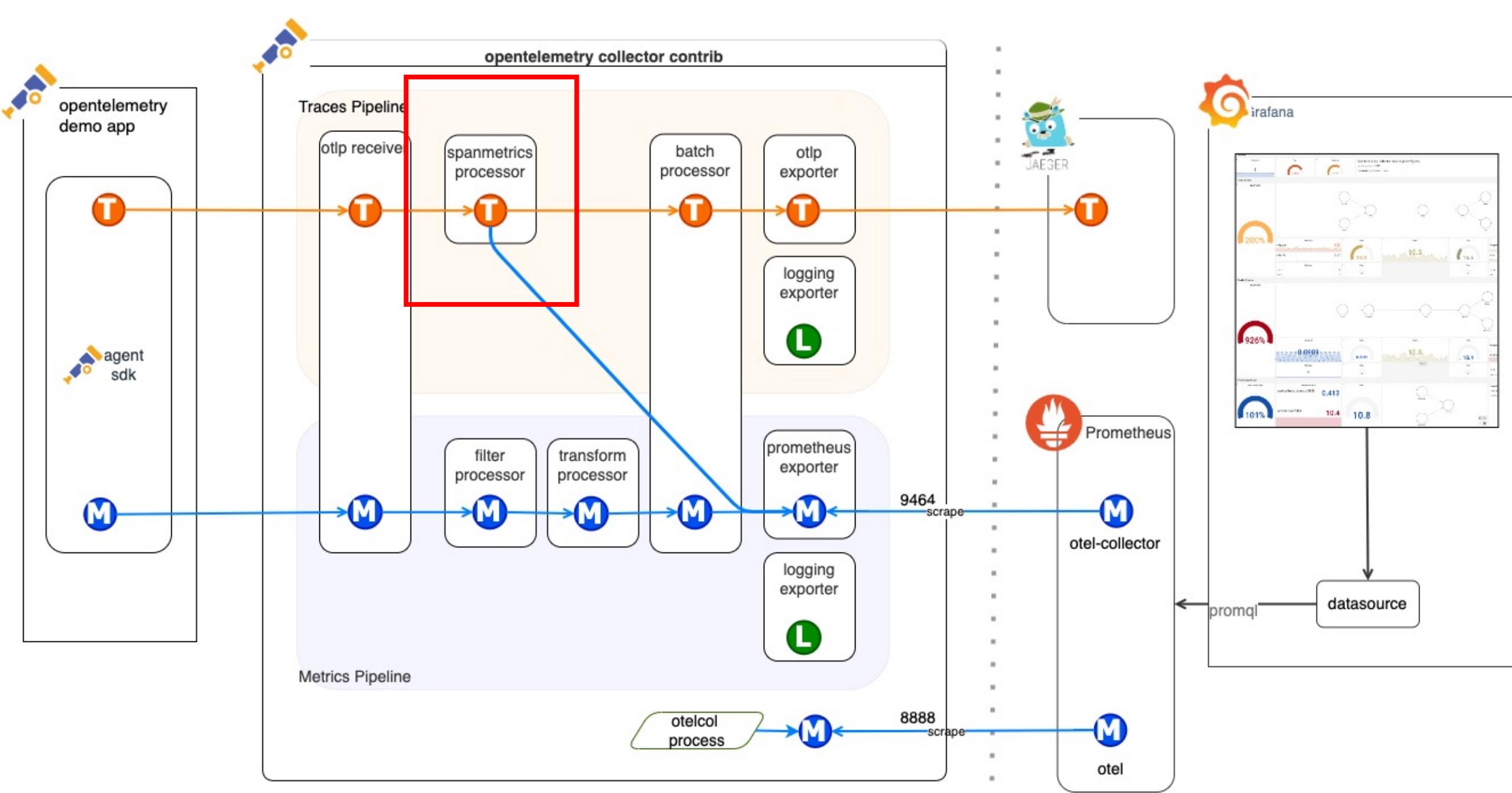
# Кластерные метрики

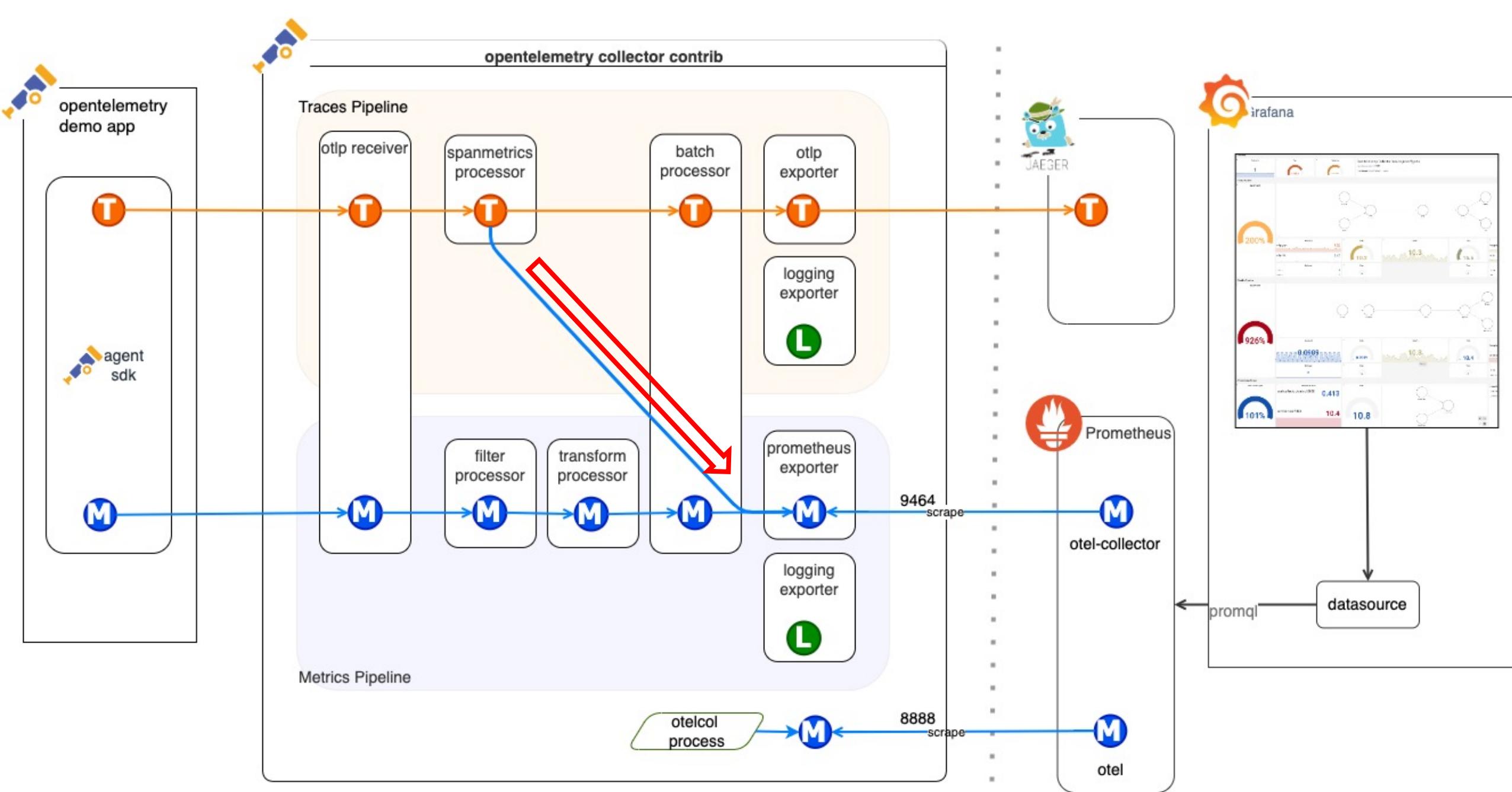


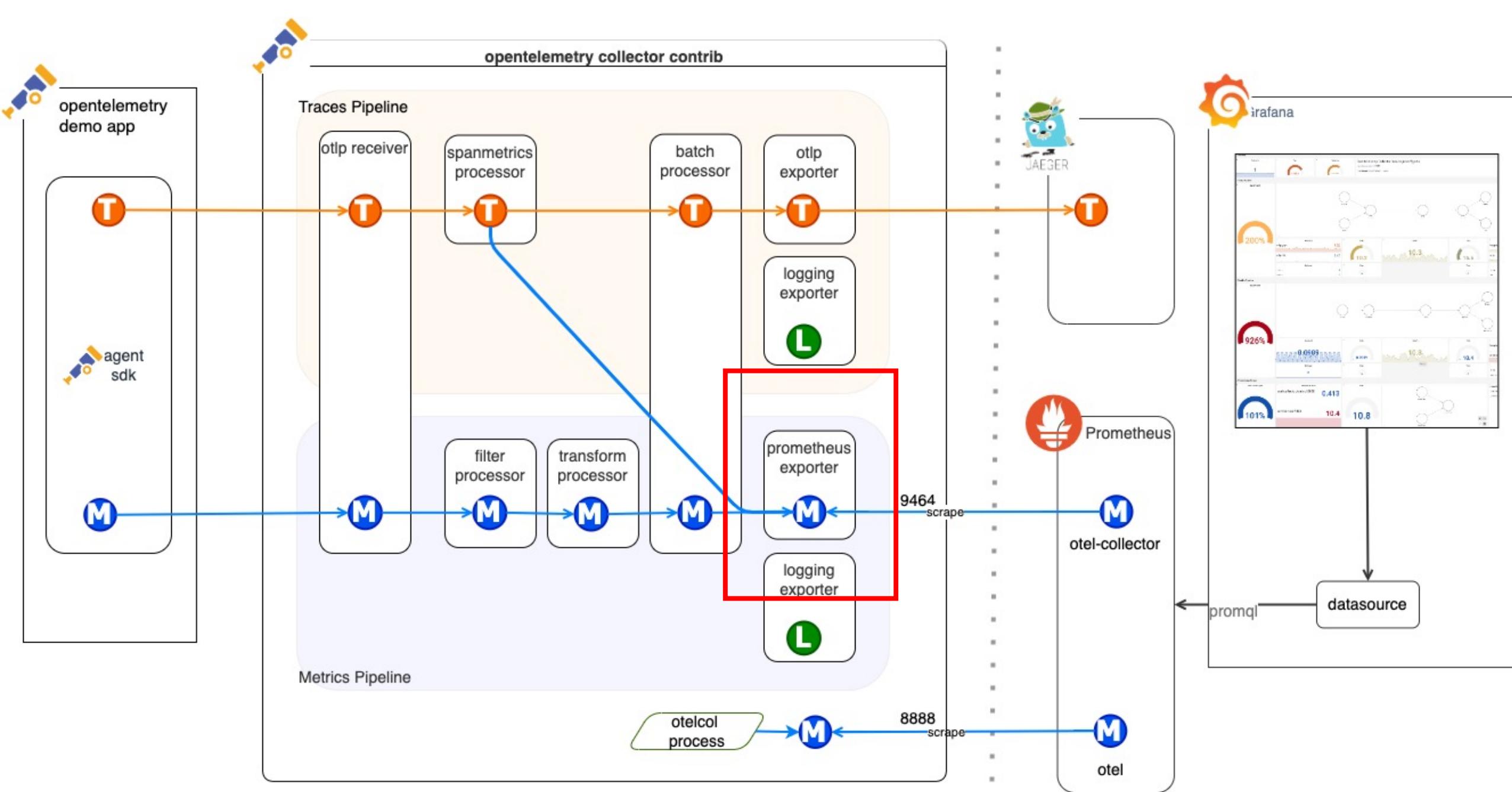
# **Кластерные метрики из трассировок**











Jul 20th, '24 / 4 min read

# Convert OpenTelemetry Traces to Metrics with SpanMetrics

Already implemented tracing but lack metrics? With SpanConnector, you can convert trace data into actionable metrics. Here's how to configure it.



Prathamesh Sonpatki

Share:    

# Кластерные метрики из трассировок

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- `start_time`, `end_time`

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*
- http.response.status\_code

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*
- http.response.status\_code
- http.request.size
- http.response.size

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*
- http.response.status\_code
- http.request.size
- http.response.size
- кастомные атрибуты

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*
- http.response.status\_code
- http.request.size
- http.response.size
- кастомные атрибуты

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
- client.address + port
- server.address + port
- url.\*
- http.response.status\_code
- http.request.size
- http.response.size
- **кастомные атрибуты**

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - http.response.status\_code
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - Errors
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - http.response.status\_code
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - **Errors**
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - **http.response.status\_code**
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - **Errors**
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - **http.response.status\_code** -> 4xx 5xx
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - **Errors**
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - http.response.status\_code
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - Errors
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- `start_time`, `end_time`
  - `client.address + port`
  - `server.address + port`
  - `url.*`
  - `http.response.status_code`
  - `http.request.size`
  - `http.response.size`
  - кастомные атрибуты
- RED method
- `Rate`
  - `Errors`
  - `Duration`

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- `start_time`, `end_time`
  - `client.address + port`
  - `server.address + port`
  - `url.*`
  - `http.response.status_code`
  - `http.request.size`
  - `http.response.size`
  - кастомные атрибуты
- RED method
- `Rate`
  - Errors
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- `start_time`, `end_time`
  - `client.address + port`
  - `server.address + port`
  - `url.*`
  - `http.response.status_code`
  - `http.request.size`
  - `http.response.size`
  - кастомные атрибуты
- 
- RED method
  - `Rate`
  - Errors
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

- start\_time, end\_time
  - client.address + port
  - server.address + port
  - url.\*
  - http.response.status\_code
  - http.request.size
  - http.response.size
  - кастомные атрибуты
- RED method
- Rate
  - Errors
  - Duration

# Кластерные метрики из трассировок

Что внутри HTTP Tracing Span?

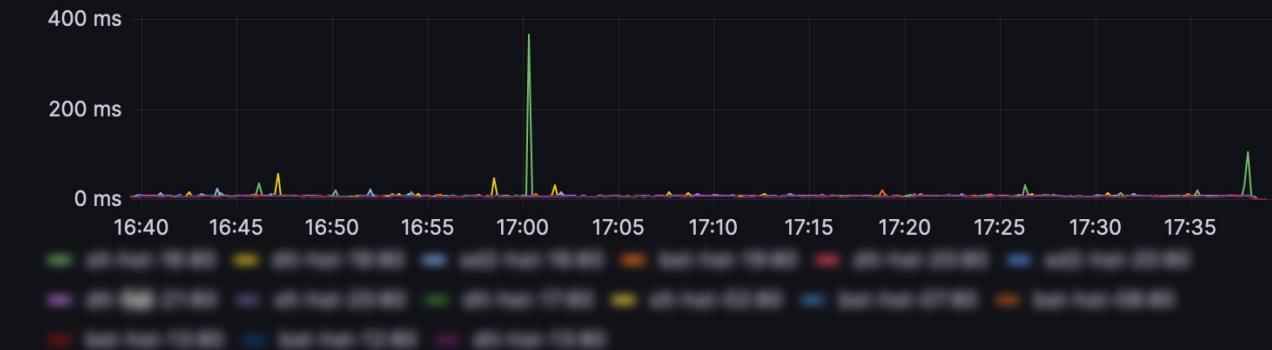
- `start_time`, `end_time`
  - `client.address + port`
  - `server.address + port`
  - `url.*`
  - `http.response.status_code`
  - `http.request.size`
  - `http.response.size`
  - кастомные атрибуты
- RED method
- Rate
  - Errors
  - Duration

## ✓ Snitch (http server)

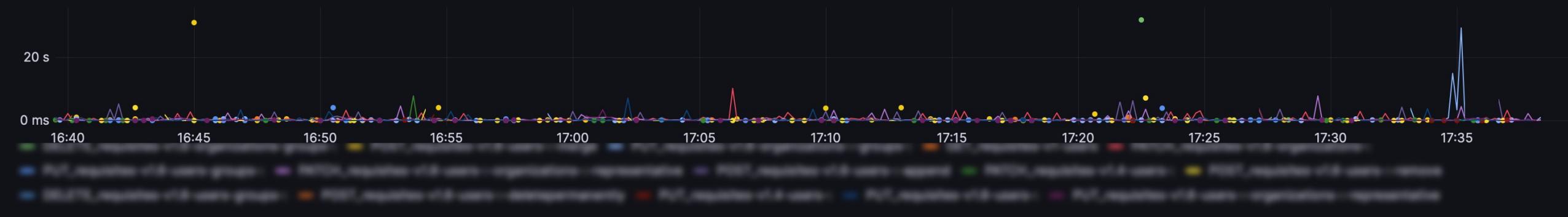
RPS per replica



Latency per replica (p95)



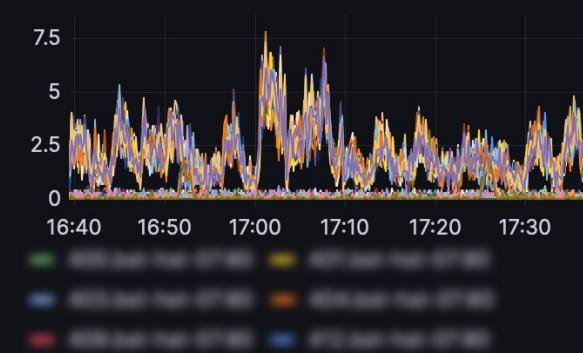
Latency per url (top-15) (p95)



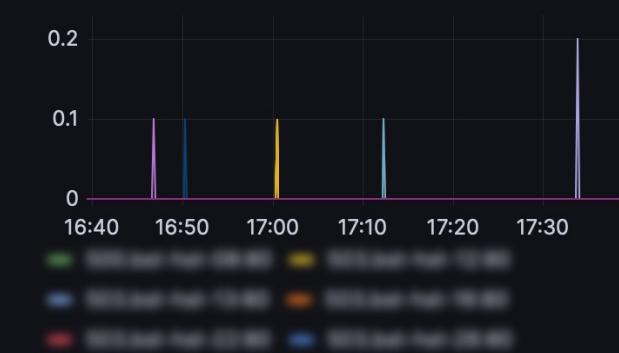
2xx and 3xx codes per replica (RPS)



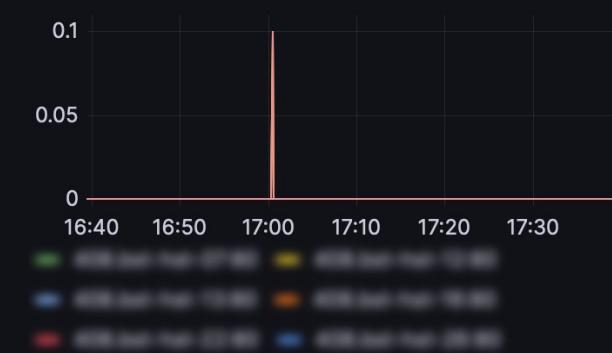
4xx client errors (RPS)



5xx server errors per replica (RPS)



4xx net and other errors per replica (RPS)

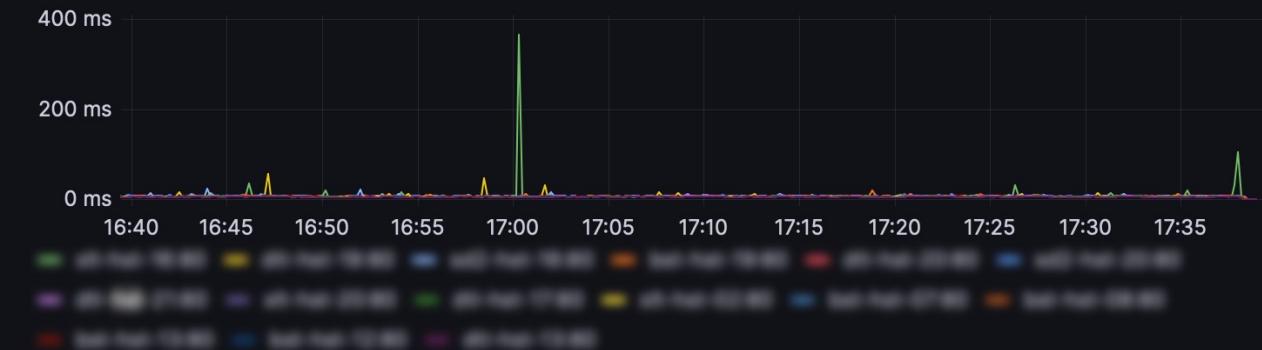


## Snitch (http server)

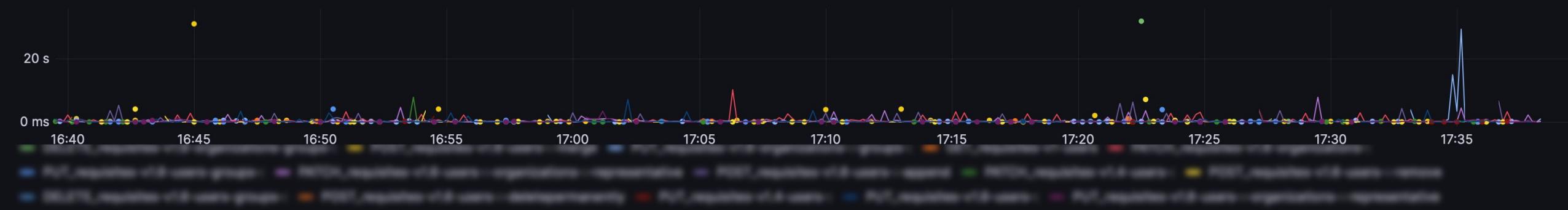
RPS per replica



Latency per replica (p95)



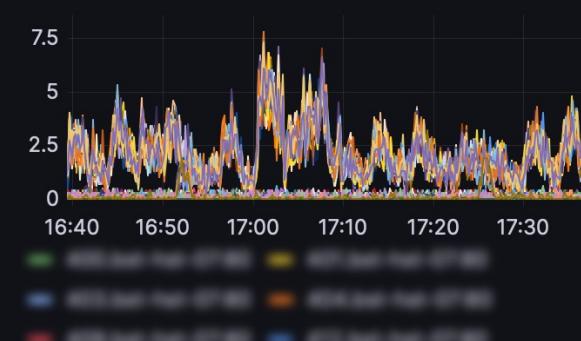
Latency per url (top-15) (p95)



2xx and 3xx codes per replica (RPS)



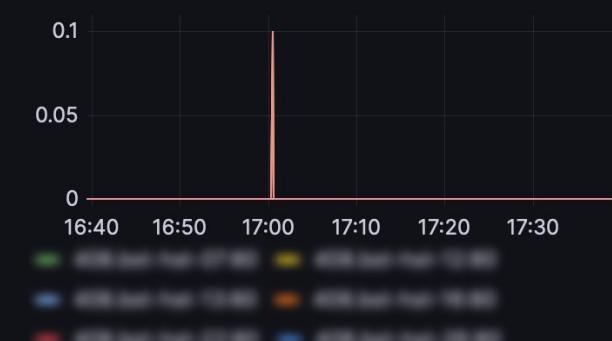
4xx client errors (RPS)



5xx server errors per replica (RPS)



4xx net and other errors per replica (RPS)

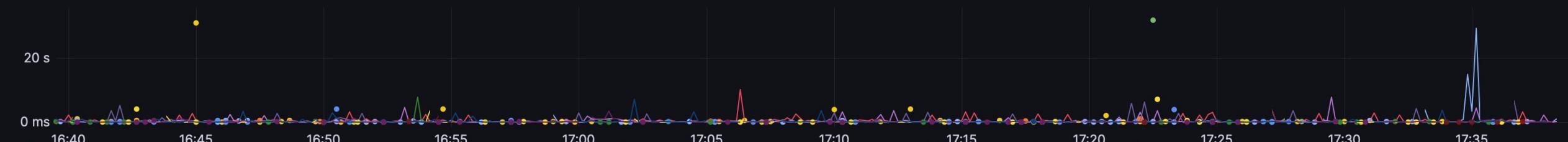


## ✓ Snitch (http server)

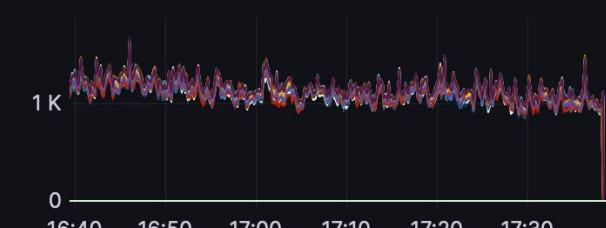
RPS per replica



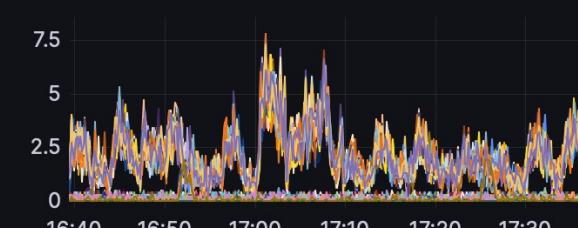
Latency per url (top-15) (p95)



2xx and 3xx codes per replica (RPS)



4xx client errors (RPS)



Latency per replica (p95)



5xx server errors per replica (RPS)



4xx net and other errors per replica (RPS)



## ✓ Snitch (http server)

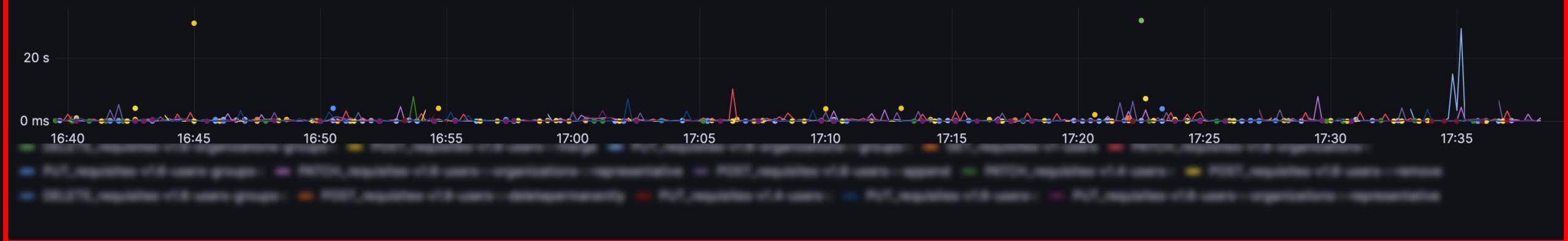
RPS per replica



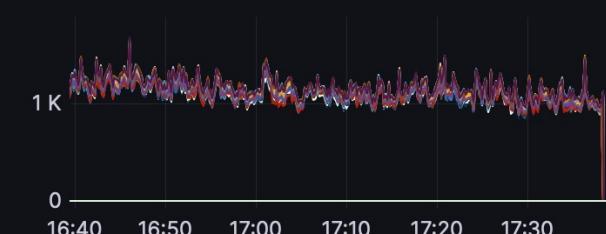
Latency per replica (p95)



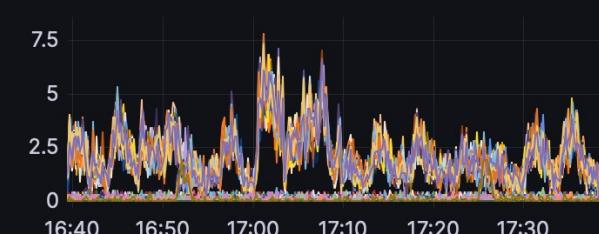
Latency per url (top-15) (p95)



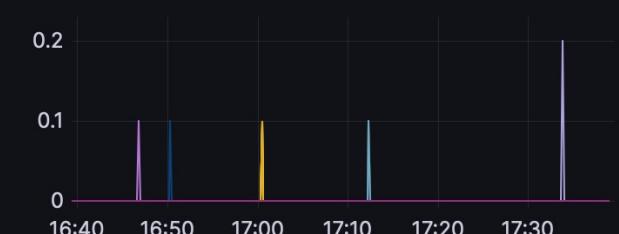
2xx and 3xx codes per replica (RPS)



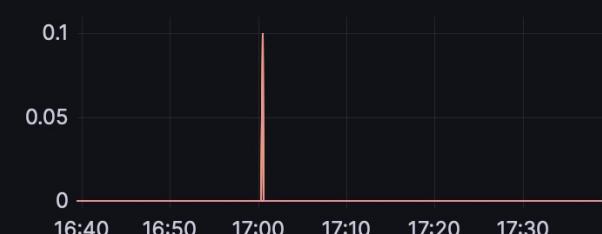
4xx client errors (RPS)



5xx server errors per replica (RPS)



4xx net and other errors per replica (RPS)



## ✓ Snitch (http server)

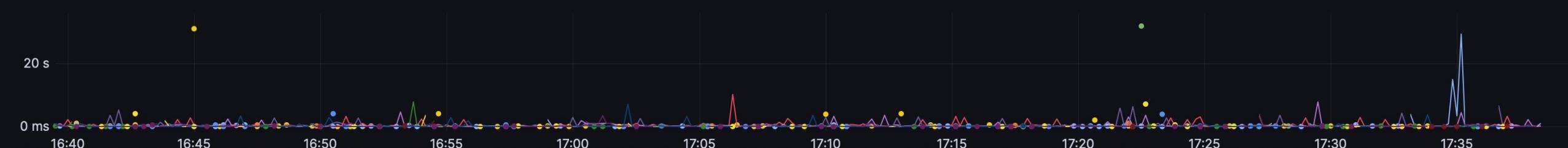
RPS per replica



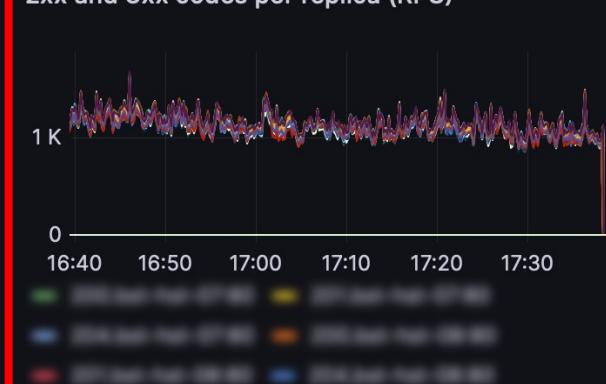
Latency per replica (p95)



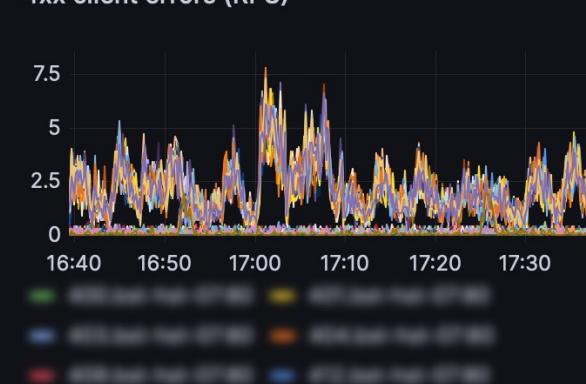
Latency per url (top-15) (p95)



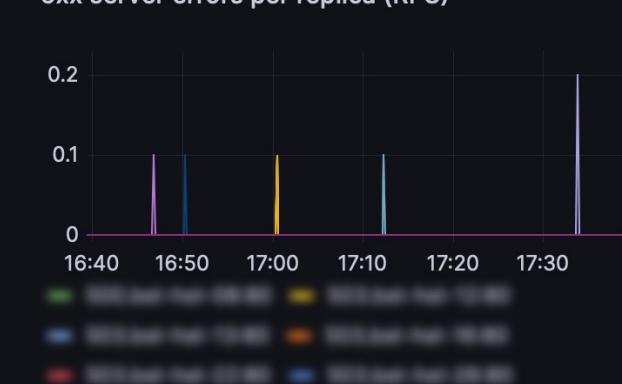
2xx and 3xx codes per replica (RPS)



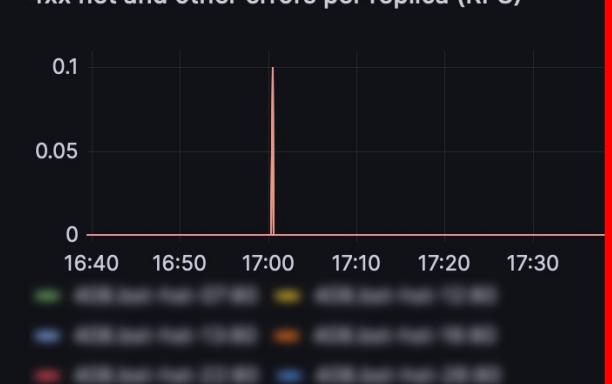
4xx client errors (RPS)



5xx server errors per replica (RPS)



4xx net and other errors per replica (RPS)



# **Активный мониторинг**

Мы немного увлеклись...

# **Активный мониторинг**

Мы немного увлеклись...

Что дальше?

# Eliminating Toil

## Избавляемся от рутины

«If a human operator needs to touch your system during normal operations, you have a **bug**. The definition of normal changes as your systems grow.»

Carla Geisser, Google SRE  
(эпиграф к 5 главе, SRE book)

# Телеметрия

Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

## Расширенная телеметрия

# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

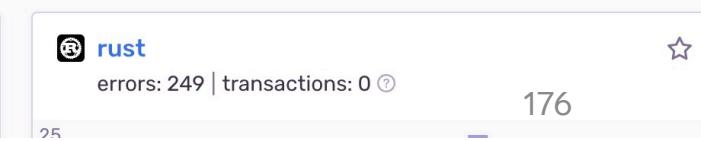
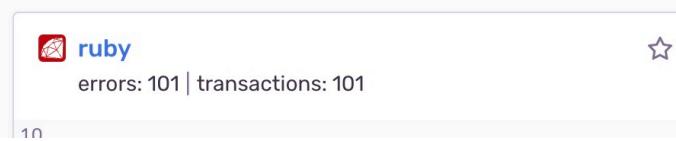
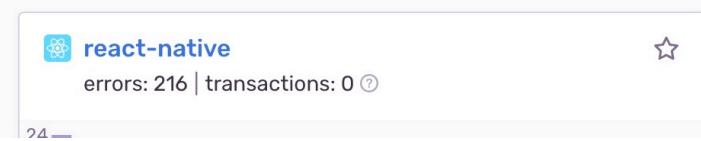
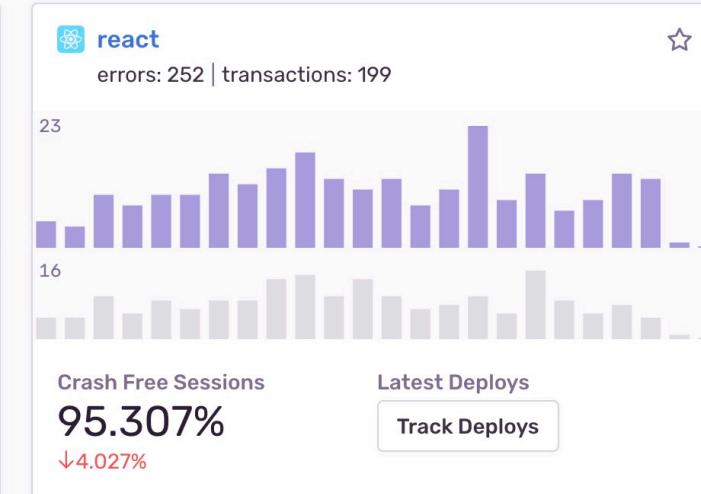
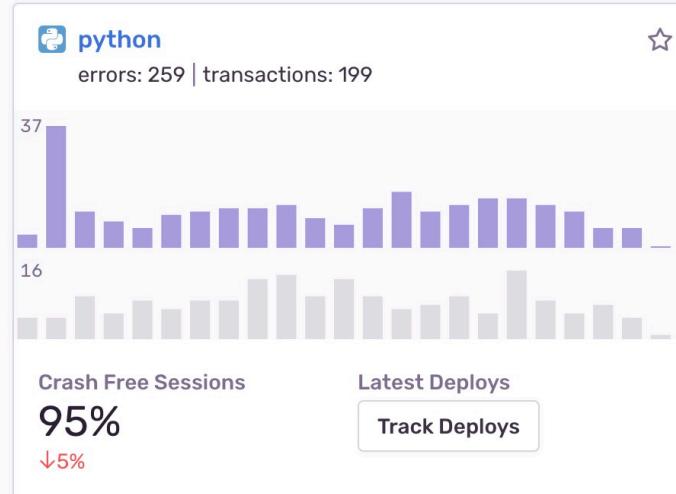
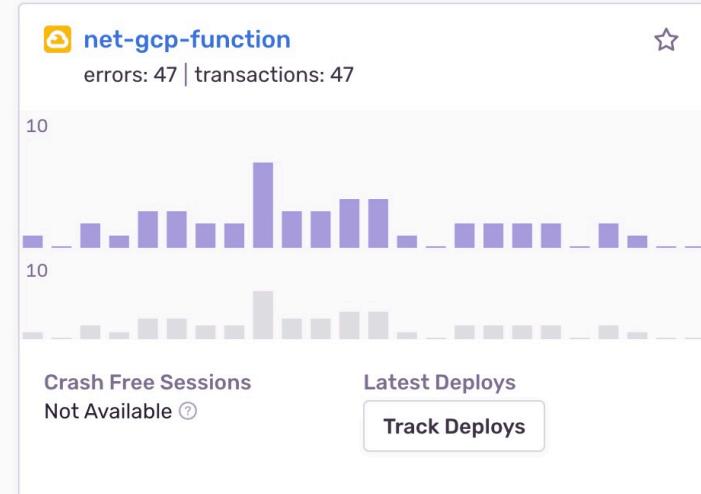
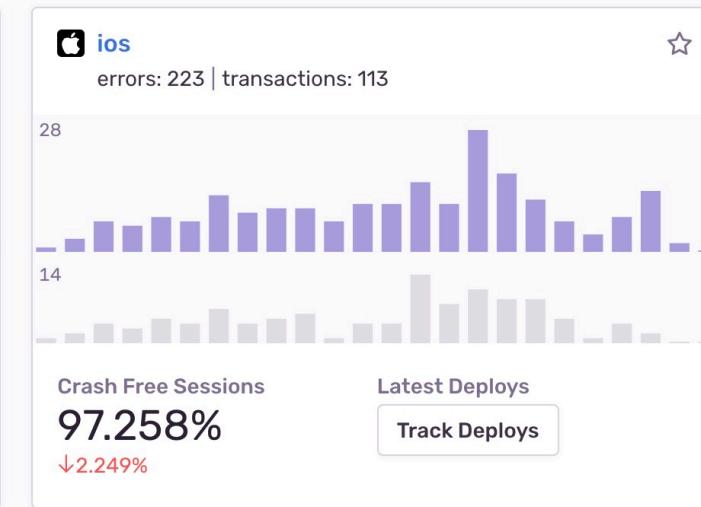
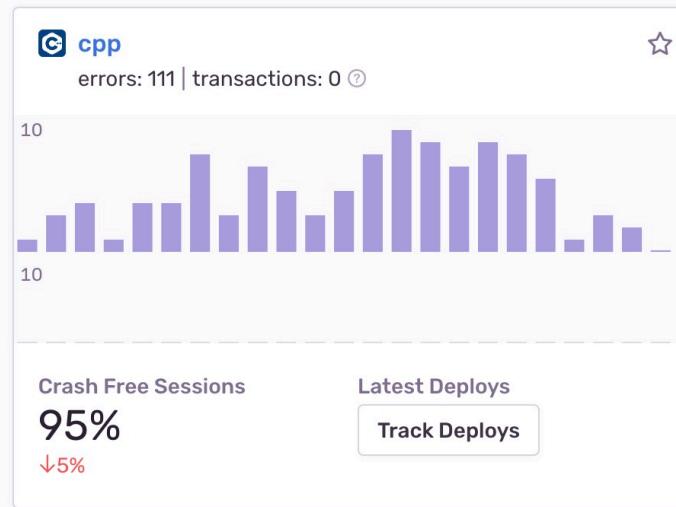
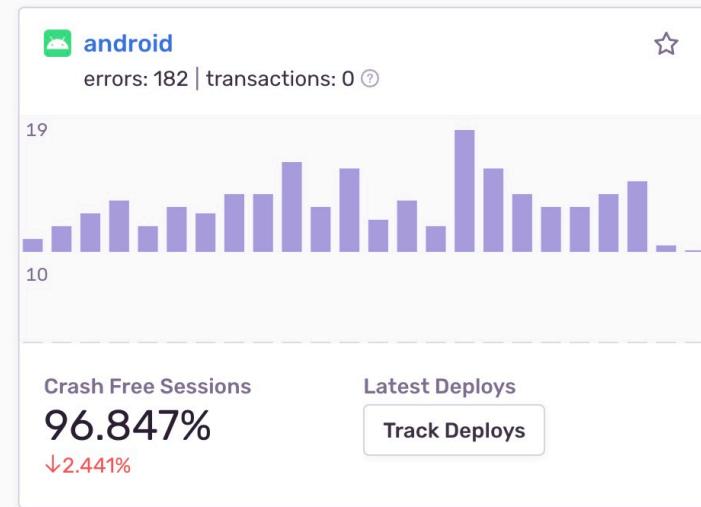
## Расширенная телеметрия

- **Ошибки**



# Projects

## DO #definite-opossum



**DO** Definite Opossu...  
definite-opossum@...

- Projects
- Issues
- Performance
- Releases
- User Feedback
- Alerts
- Discover
- Dashboards
- Activity
- Stats
- Settings

Help

Collapse

← react-native ⚡



All Environments



## Error Promise\$argument\_0(HomeScreen) Regression

● Unhandled Promise Rejection

ISSUE # REACT-NATIVE-2

EVENTS 131

USERS 112

ASSIGNEE

Details Activity 0 User Feedback 0 Attachments Tags Events Merged Issues

Event 9b2761cc5d624ecb99d069b972dead50 | JSON (15.8 KiB)

May 7, 2022 1:09:16 AM UTC

◀ Older Newer ▶

### TAGS

E edyth@example.com ID: 1527

iOS Version: 13.3

iPad Air (4th generation)

device iPad Air (4th generation) device.family iOS dist 2.5.1.0 environment production handled yes level error  
mechanism generic os iOS 13.3 os.name iOS release 3.2 user id:1527

### EXCEPTION (most recent call first)

App Only Full Raw

#### Error

Unhandled Promise Rejection

mechanism generic handled true

</> app:///HomeScreen.tsx in Promise\$argument\_0 at line 177:24

```
172.           </TouchableOpacity>
173.           <View style={styles.spacer} />
174.           <TouchableOpacity
175.             onPress={() => {
176.               new Promise(() => {
177.                 throw new Error('Unhandled Promise Rejection');
```

### Ownership Rules

Create Ownership Rule

### All Environments

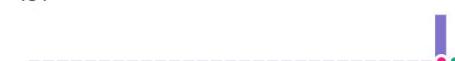
LAST 24 HOURS

130



LAST 30 DAYS

131



LAST SEEN

an hour ago in release 3.2

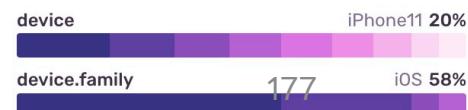
FIRST SEEN

a day ago in release 3.2

### ISSUE TRACKING

Track this issue in Jira, GitHub, etc. >

### Tags



# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

## Расширенная телеметрия

- **Ошибки**

# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые  
трассировки

## Расширенная телеметрия

- Ошибки
- Алерты

# Алерты

- PagerDuty
- Prometheus Alertmanager
- Grafana Alerting
- Moira

# Алерты

- PagerDuty
- Prometheus Alertmanager
- Grafana Alerting
- Moira

# K8S [ ] - deployment unavailable in namespace

[Edit](#) [Maintenance](#) [Other](#)

Target  
(prometheus)

```
☐ min_over_time(sum by (deployment) (kube_deployment_status_condition{namespace='  
', condition="Available", status="false"})[10m])
```

Description

Срабатывает, если Pod'ы некоторого деплоимента не могут запуститься более  
Причину такого поведения лучше смотреть в describe deployment/pod.

Где смотреть:



Не меняйте и не удаляйте триггер руками. Он управляетя автоматикой (Moira operator).

Cluster

Prometheus Remote

Value

Error: 1. Set DEL if has no value for 300 seconds

Schedule

Everyday 00:00—23:59 (GMT 00:00)

Tags

failed

deploy

# K8S [ ] - deployment unavailable in namespace

[Edit](#) [Maintenance](#) [Other](#)

Target  
(prometheus)

`min_over_time(sum by (deployment) (kube_deployment_status_condition{namespace='[REDACTED]', condition="Available", status="false"})[10m])`

Description

Срабатывает, если Pod'ы некоторого деплоимента не могут запуститься более 10 минут. Причину такого поведения лучше смотреть в `describe deployment/pod`.

Где смотреть:



Не меняйте и не удаляйте триггер руками. Он управляется автоматикой (Moira operator).

Cluster

Prometheus Remote

Value

Error: 1. Set DEL if has no value for 300 seconds

Schedule

Everyday 00:00—23:59 (GMT 00:00)

Tags

failed

deploy

# K8S [ ] - deployment unavailable in namespace

[Edit](#) [Maintenance](#) [Other](#)

Target  
(prometheus)

```
 min_over_time(sum by (deployment) (kube_deployment_status_condition{namespace='  
', condition="Available", status="false"})[10m])
```

Description

Срабатывает, если Pod'ы некоторого деплоимента не могут запуститься более  
Причину такого поведения лучше смотреть в describe deployment/pod.

Где смотреть:



Не меняйте и не удаляйте триггер руками. Он управляется автоматикой (Moira operator).

Cluster

Prometheus Remote

Value

Error: 1. Set DEL if has no value for 300 seconds

Schedule

Everyday 00:00—23:59 (GMT 00:00)

Tags

failed

deploy

## K8S [ ] - deployment unavailable in namespace

[Edit](#) [Maintenance](#) [Other](#)

Target  
(prometheus)

```
☐ min_over_time(sum by (deployment) (kube_deployment_status_condition{namespace='  
', condition="Available", status="false"})[10m])
```

Description

Срабатывает, если Pod'ы некоторого деплоимента не могут запуститься более  
Причину такого поведения лучше смотреть в describe deployment/pod.

Где смотреть:



Не меняйте и не удаляйте триггер руками. Он управляется автоматикой (Moira operator).

Cluster

Prometheus Remote

Value

Error: 1. Set DEL if has no value for 300 seconds

Schedule

Everyday 00:00—23:59 (GMT 00:00)

Tags

failed

deploy

 OK [Production]

#### План действий

- Проанализируй:

- 
- 
- 
- 
- 
- Дашборд

- [Дашборд](#)

- В зависимости от результата:

This trigger is manipulated by [ansible](#).

01:33 (GMT+03:00):

 [1 ответ](#) | Отслеживать



## План действий

- Проанализируй:
  - 
  - 
  - 
  - 
  - 
  - Дашборд
- Дашборд
- В зависимости от результата:

This trigger is manipulated by [ansible](#).

01:33 (GMT+03:00):

1 ответ | Отслеживать

OK [Production]



 OK

This trigger is manipulated by [ansible](#).

01:33 (GMT+03:00):

 Прокомментировать

# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые трассировки

## Расширенная телеметрия

- Ошибки
- Алерты

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

# Логи и ошибки

# Логи и ошибки

- Структурированное логирование

# Логи и ошибки

- Структурированное логирование (JSON)

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования
  - \* **Ошибка** предполагает участие инженера

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования
  - \* Ошибка предполагает участие инженера
  - \* Снижение фонового шума

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования
  - \* Ошибка предполагает участие инженера
  - \* Снижение фонового шума
- Группировка ошибок по повторяющимся признакам

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования
  - \* Ошибка предполагает участие инженера
  - \* Снижение фонового шума
- Группировка ошибок по повторяющимся признакам
- Бюджет на ошибки

# Логи и ошибки

- Структурированное логирование (JSON)
- Проброс Tracing Context (Traceld, SpanId)
- Конвенции по именованию атрибутов
- Разграничение уровней логирования
  - \* Ошибка предполагает участие инженера
  - \* Снижение фонового шума
- Группировка ошибок по повторяющимся признакам
- Бюджет на ошибки
- Заменить метриками, где это имеет смысл



# Проектирование метрик и алертов

# Проектирование метрик и алERTов

RED method

- Rate
- Errors
- Duration

# Проектирование метрик и алERTов

RED method

- Rate
- Errors
- Duration

USE method

- Utilization
- Saturation
- Errors

# Проектирование метрик и алERTов

4 Golden Signals

RED method

- Rate
- Errors
- Duration

USE method

- Utilization
- Saturation
- Errors

# Проектирование метрик и алERTов

4 Golden Signals

- Traffic
- Errors
- Saturation
- Latency

RED method

- Rate
- Errors
- Duration

USE method

- Utilization
- Saturation
- Errors

# Проектирование метрик и алERTов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

RED method

- Rate
- Errors
- Duration

USE method

- Utilization
- Saturation
- Errors



**DevOops**  
2023

# SRE и перформанс, или Когда производительно сть имеет значение



**Григорий  
Кошелев**

Контур

# Проектирование метрик и алертов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

# Проектирование метрик и алERTов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

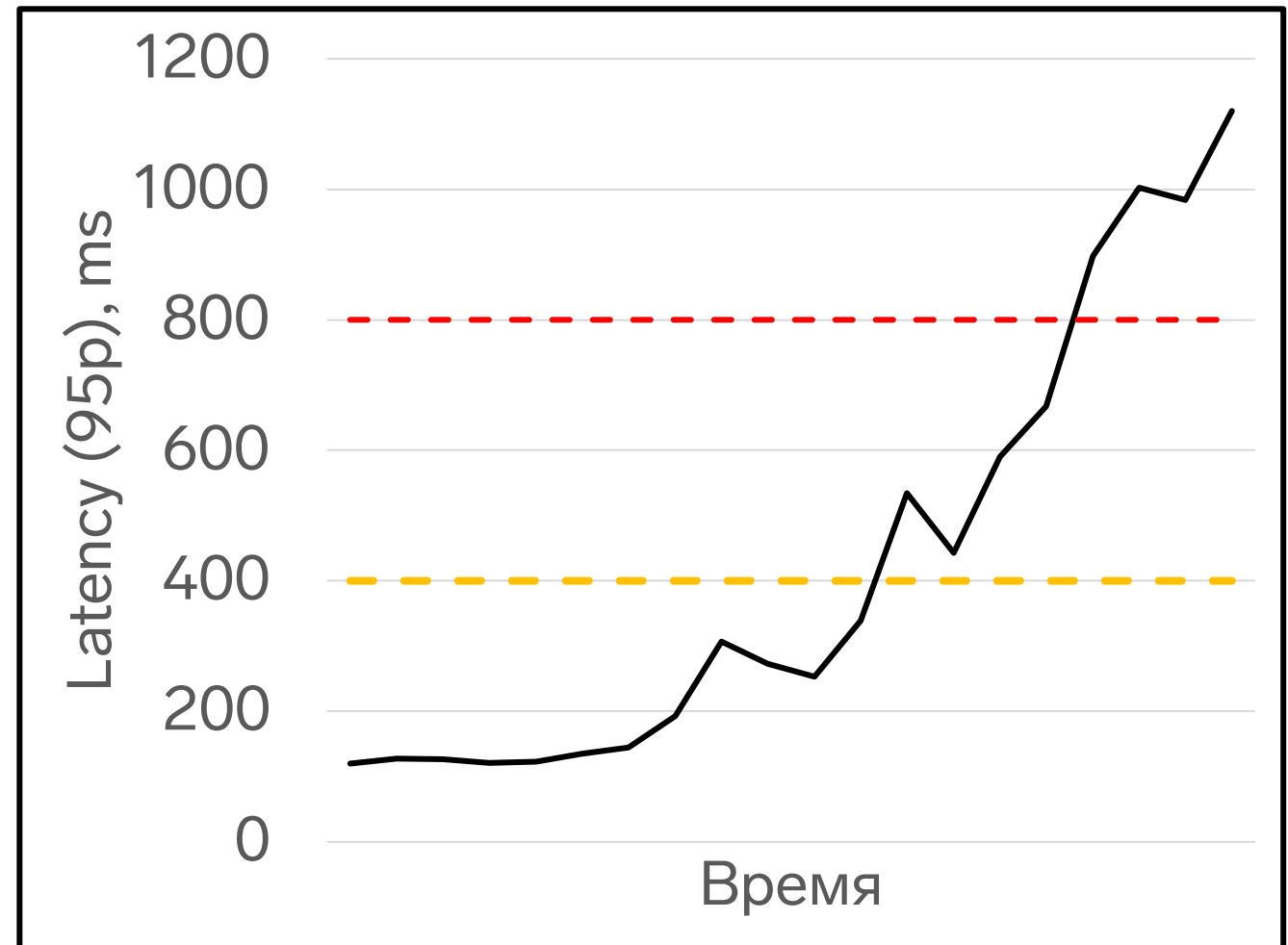
SLO как инструмент  
ранней диагностики

# Проектирование метрик и алERTов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

SLO как инструмент ранней диагностики

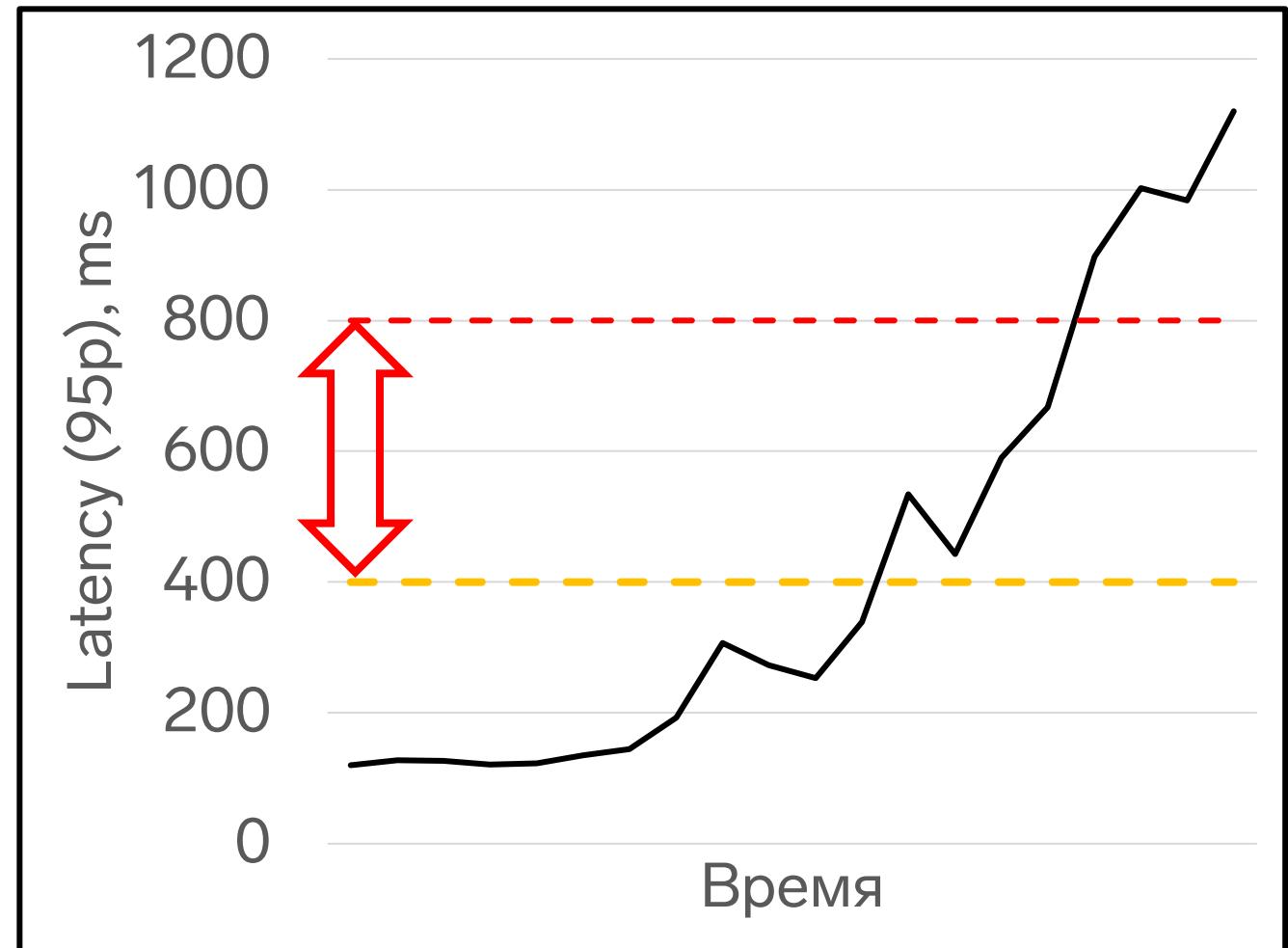


# Проектирование метрик и алERTов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

SLO как инструмент ранней диагностики

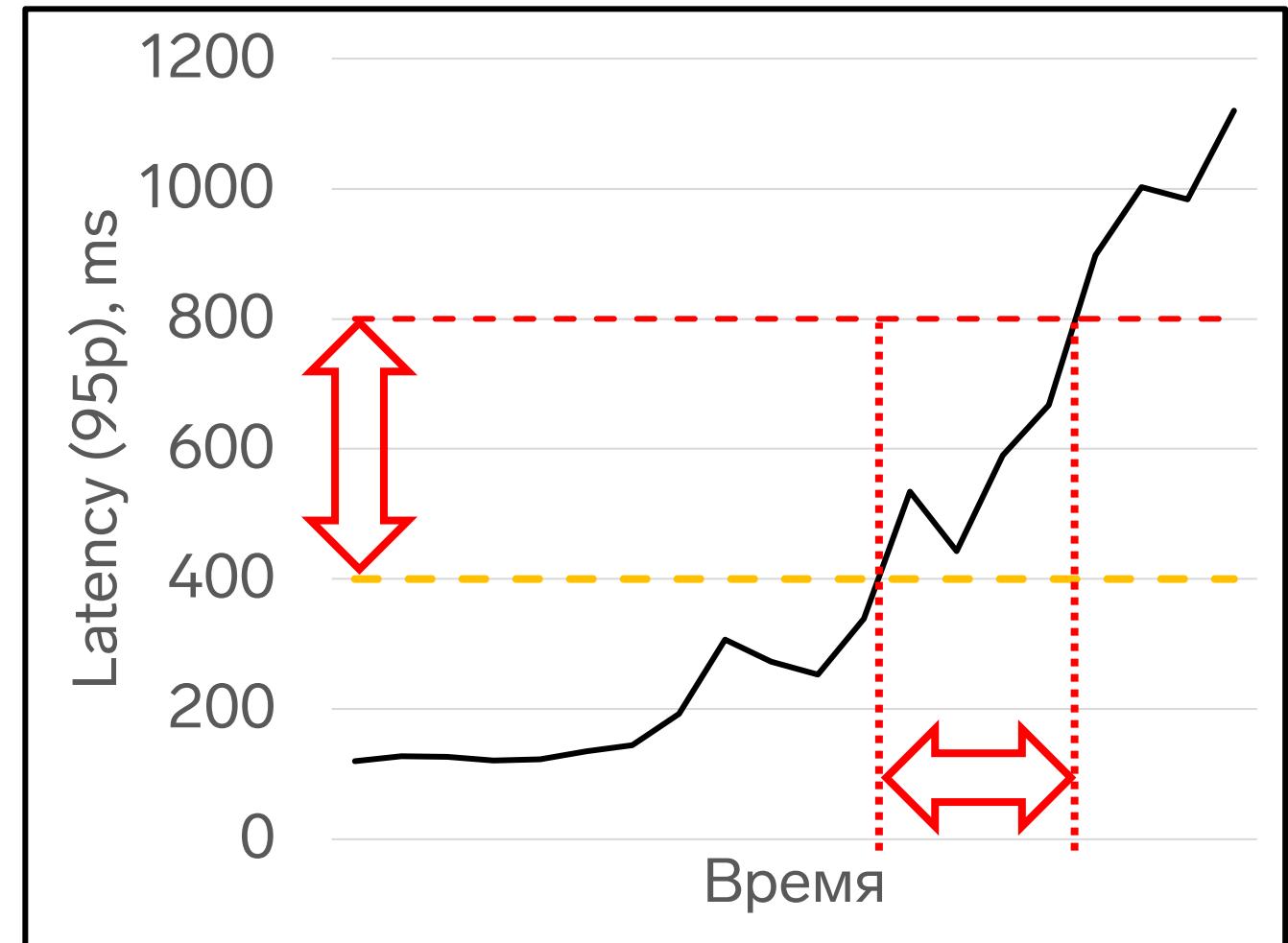


# Проектирование метрик и алERTов

4 Golden Signals = TESLa

- Traffic
- Errors
- Saturation
- Latency

SLO как инструмент ранней диагностики



# **Проектирование метрик и алертов**

# Проектирование метрик и алертов

- Конвенции по именованию атрибутов

# Проектирование метрик и алертов

- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному

# Проектирование метрик и алERTов

- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному
- Лучшее покрытие у сценариев с SLO

# Проектирование метрик и алертов

- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному
- Лучшее покрытие у сценариев с SLO
- Разграничение алертов по критичности

# Проектирование метрик и алертов

- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному
- Лучшее покрытие у сценариев с SLO
- Разграничение алертов по критичности
  - \* Нарушение SLO предполагает участие инженера

# Проектирование метрик и алертов

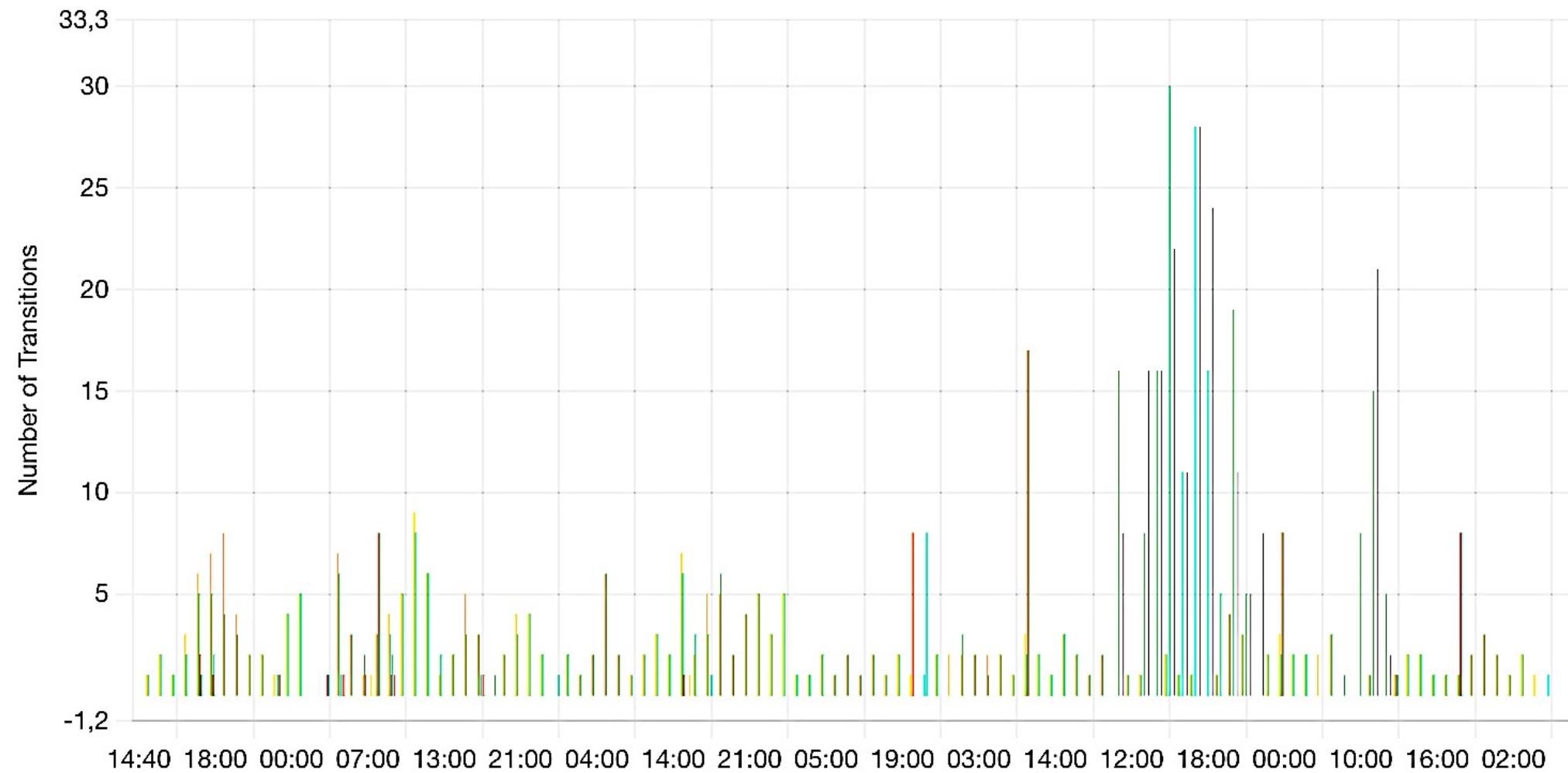
- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному
- Лучшее покрытие у сценариев с SLO
- Разграничение алертов по критичности
  - \* Нарушение SLO предполагает участие инженера
  - \* Снижение фонового шума

## Trigger transitions

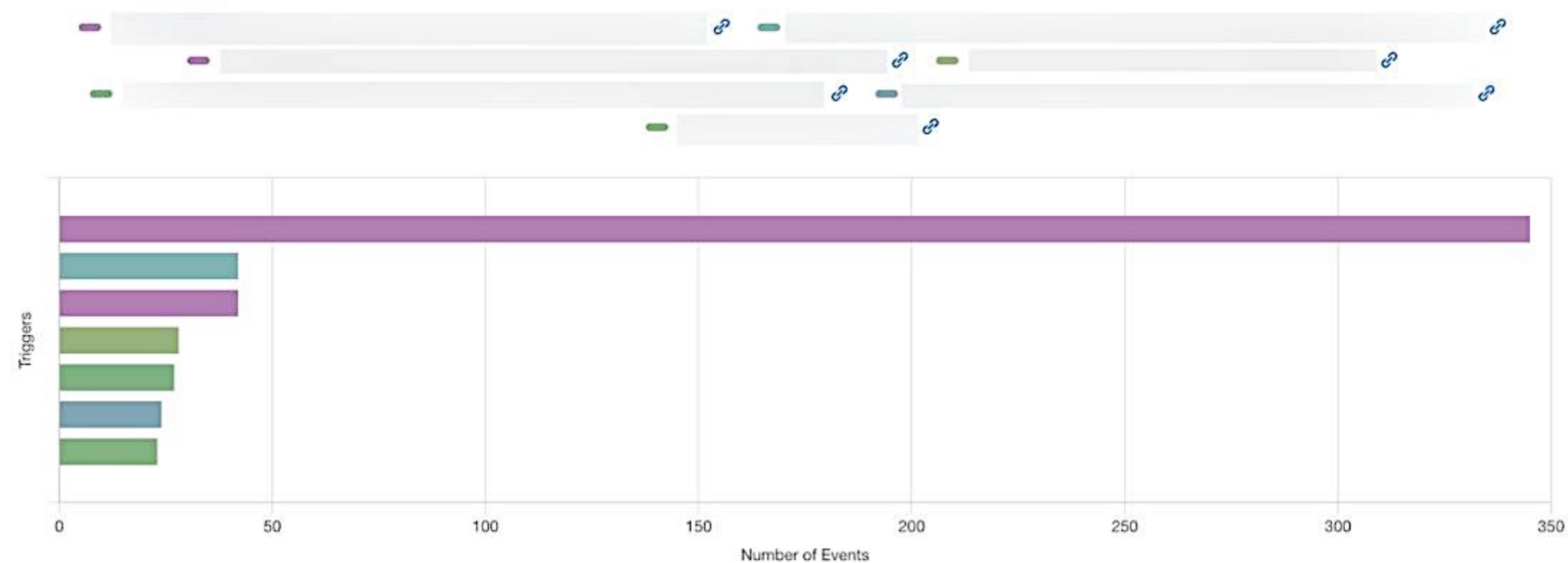
Select Interval hour ▾

OK to WARN    WARN to OK    OK to ERROR    ERROR to OK    NODATA to OK

WARN to ERROR    TEST to TEST ▾



## Noisiness

[Contacts](#)    [Triggers](#)

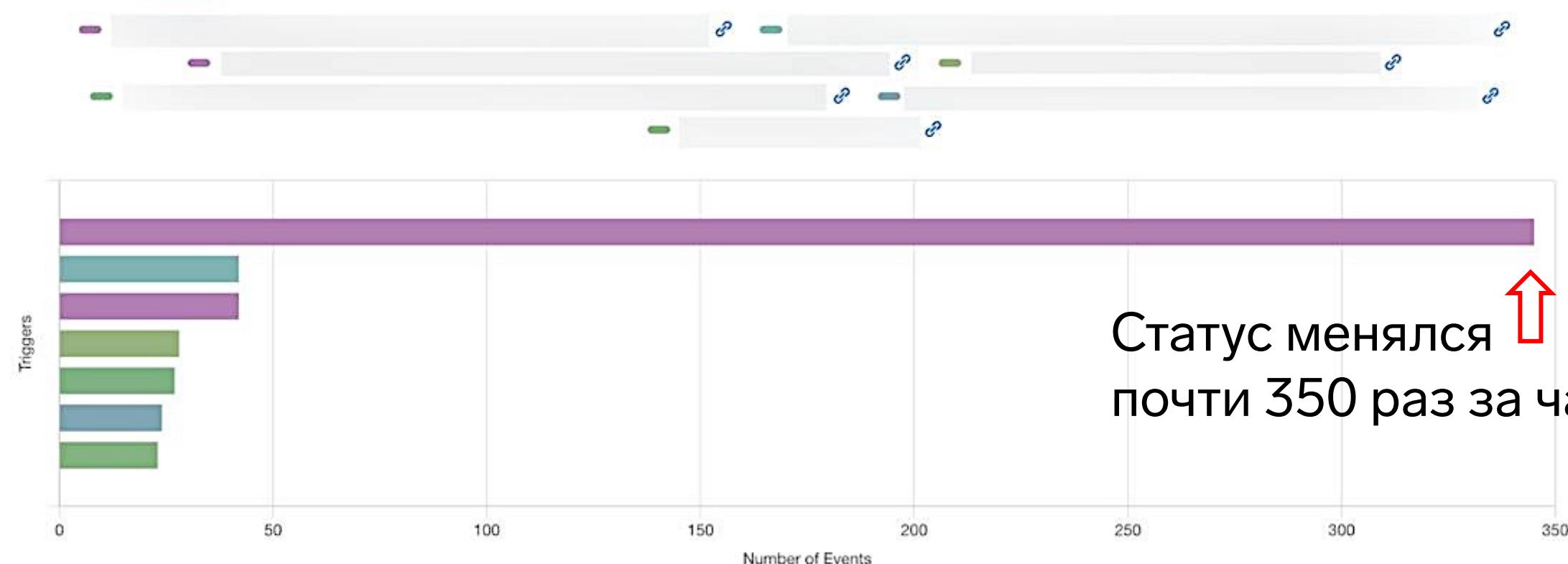
2025/04/09 09:47:46

– 2025/04/09 10:47:46

[Apply time range](#)

1 2 3 4 5

## Noisiness

[Contacts](#)    [Triggers](#)

# Проектирование метрик и алертов

- Конвенции по именованию атрибутов
- Дашборды проектируются от общего к частному
- Лучшее покрытие у сценариев с SLO
- Разграничение алертов по критичности
  - \* Нарушение SLO предполагает участие инженера
  - \* Снижение фонового шума
- Бюджет на ошибки

# Телеметрия

## Базовая телеметрия

- Логи
- Метрики
- Распределённые трассировки

## Расширенная телеметрия

- Ошибки
- Алерты

Трейсинг ошибок и алертинг становятся основным инструментом раннего обнаружения деградации

# «Интересные» трассировки

# **«Интересные» трассировки**

# «Интересные» трассировки

- Ошибки (HTTP: 4xx, 5xx)

# «Интересные» трассировки

```
processors:  
  tail_sampling:  
    decision_wait: 30s  
    num_traces: 50000  
    expected_new_traces_per_sec: 1000  
  policies: [  
    {  
      name: errors-policy,  
      type: status_code,  
      status_code: { status_codes: [ERROR] }  
    },  
    {  
      name: randomized-policy,  
      type: probabilistic,  
      probabilistic: { sampling_percentage: 25 }  
    }  
  ]
```

# «Интересные» трассировки

```
processors:  
  tail_sampling:  
    decision_wait: 30s  
    num_traces: 50000  
    expected_new_traces_per_sec: 1000  
    policies: [  
      {  
        name: errors-policy,  
        type: status_code,  
        status_code: { status_codes: [ERROR] }  
      },  
      {  
        name: randomized-policy,  
        type: probabilistic,  
        probabilistic: { sampling_percentage: 25 }  
      }  
    ]
```

# «Интересные» трассировки

```
processors:  
  tail_sampling:  
    decision_wait: 30s  
    num_traces: 50000  
    expected_new_traces_per_sec: 1000  
  policies: [  
    {  
      name: errors-policy,  
      type: status_code,  
      status_code: { status_codes: [ERROR] }  
    },  
    {  
      name: randomized-policy,  
      type: probabilistic,  
      probabilistic: { sampling_percentage: 25 }  
    }  
  ]
```

# «Интересные» трассировки

- Ошибки (HTTP: 4xx, 5xx)
- **Долгие запросы** (время выполнения > 1 сек)

# «Интересные» трассировки

- Ошибки (HTTP: 4xx, 5xx)
- **Долгие запросы** (время выполнения > 1 сек)

```
processors:  
  tail_sampling:  
    decision_wait: 30s  
    num_traces: 50000  
    expected_new_traces_per_sec: 1000  
  policies: [  
    {  
      name: slow-policy,  
      type: latency,  
      latency: { threshold_ms: 1000 }  
    }  
  ]
```



Search

Lucene



Last 15 minutes



http.request.targetService.keyword:

status.keyword: warning

hosting.keyword:

[+ Add filter](#)

Filter...



Client

Env

Operation

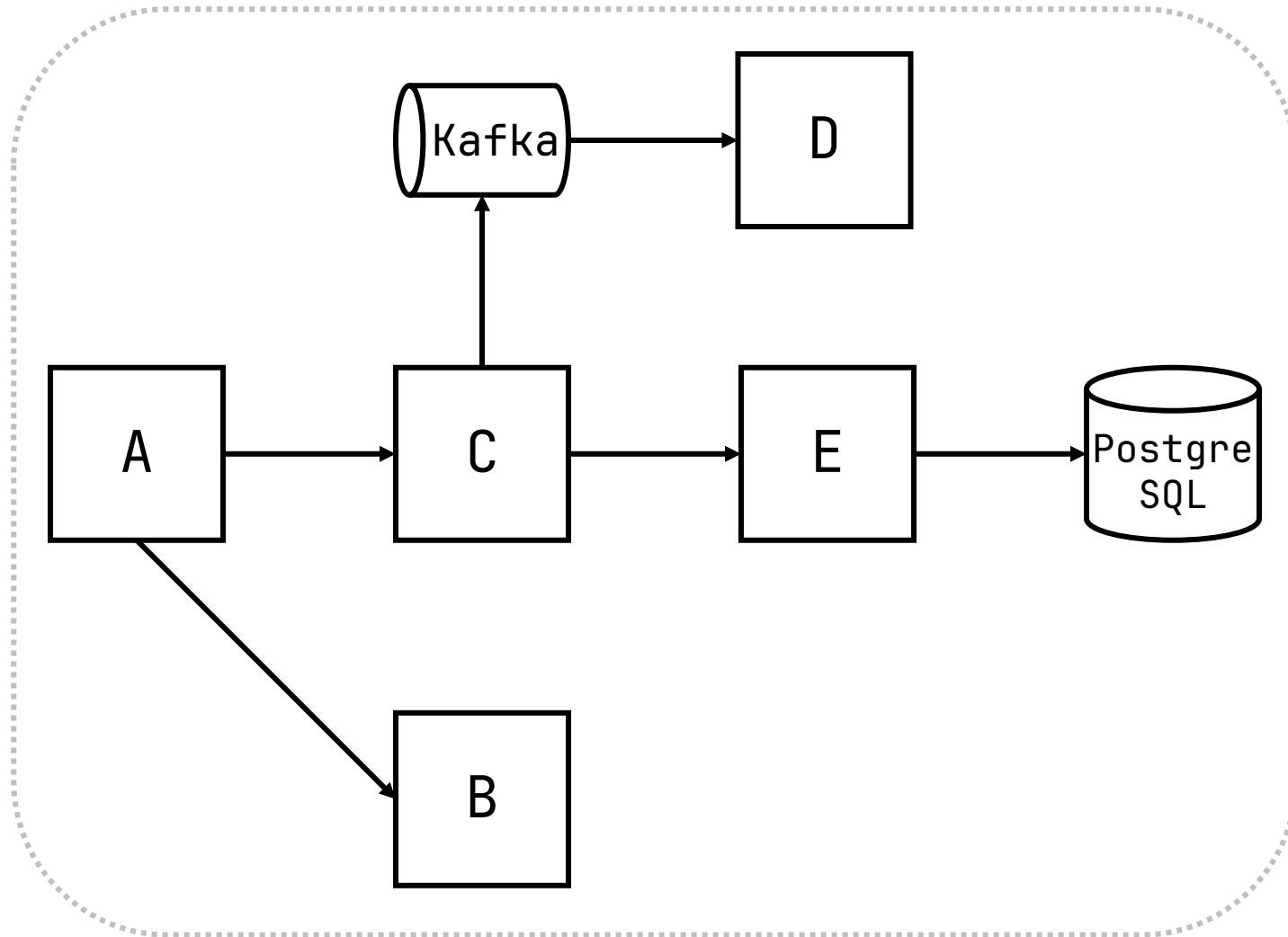
	default	GET: ~
	default	POST: v1/contents/copy
	default	POST: v1/contents/js/initpartialupload-v2
	default	POST: ~/~/v1/contents/copy
	default	POST: ~/~/v1/contents/copy

Export: [Raw](#)  [Formatted](#)

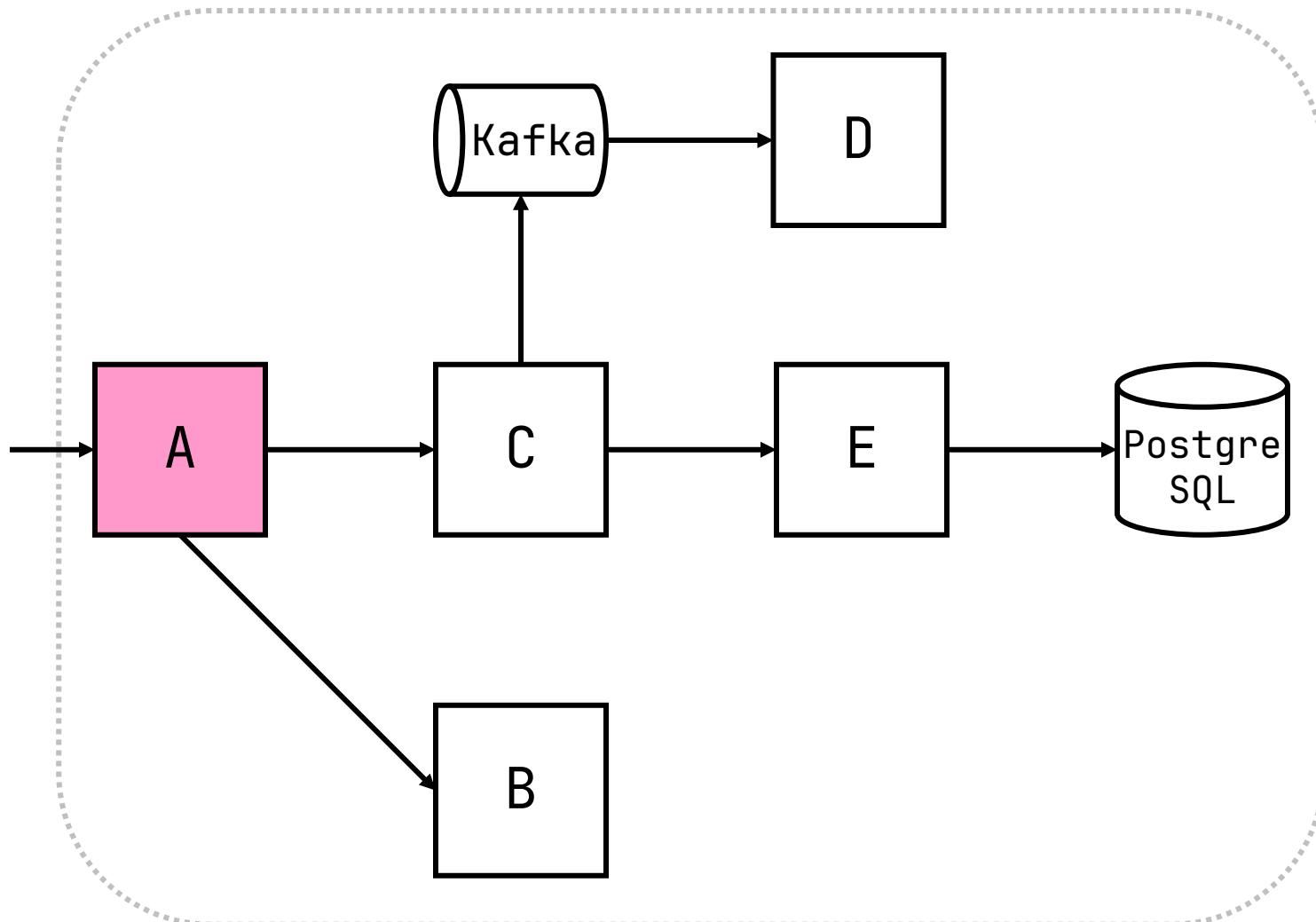


# Поиск root cause

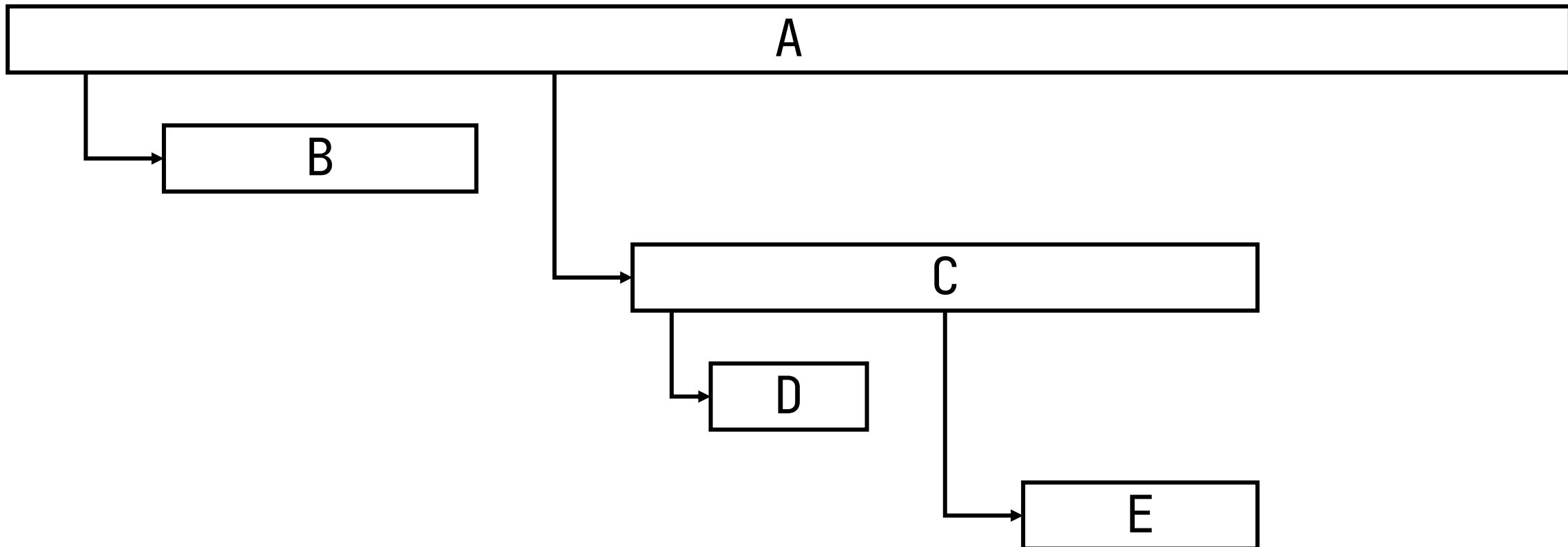
# Поиск root cause



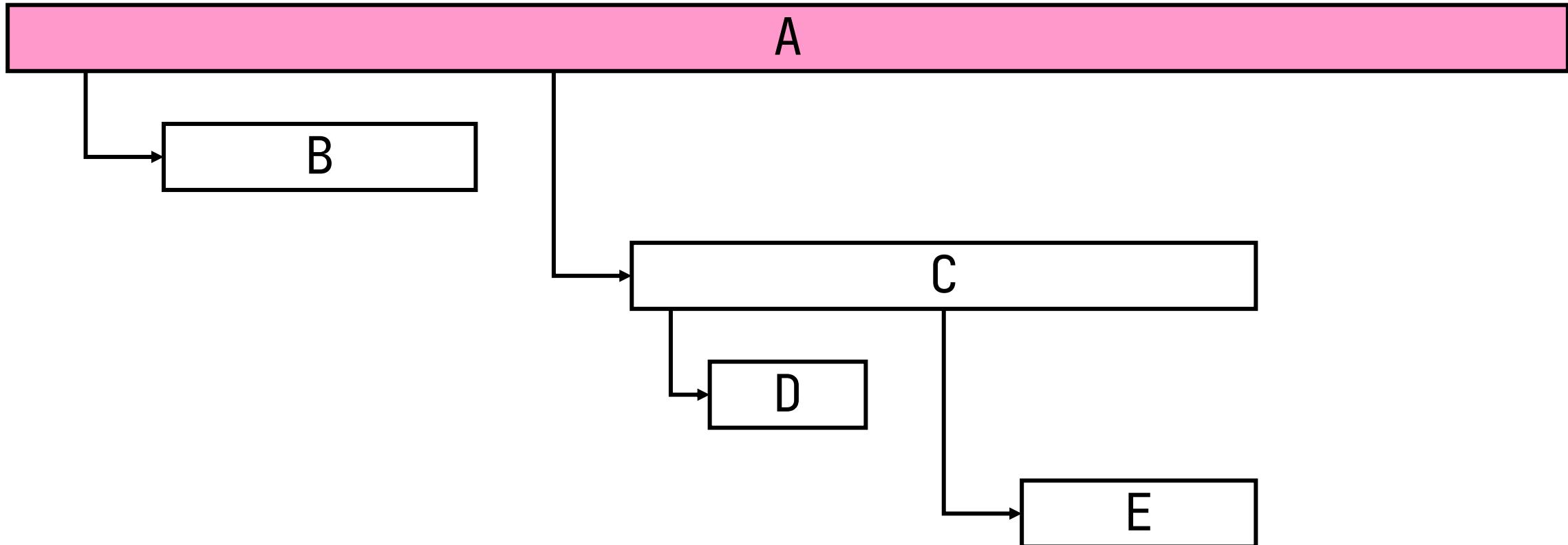
# Поиск root cause



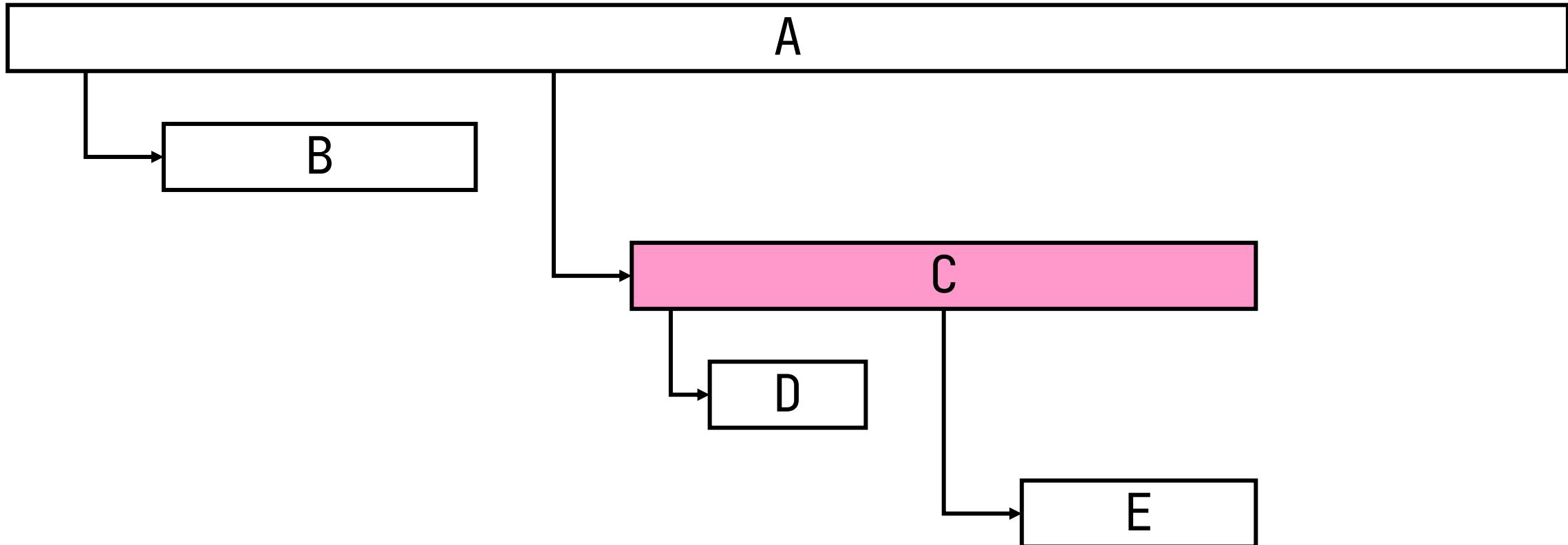
# Поиск root cause



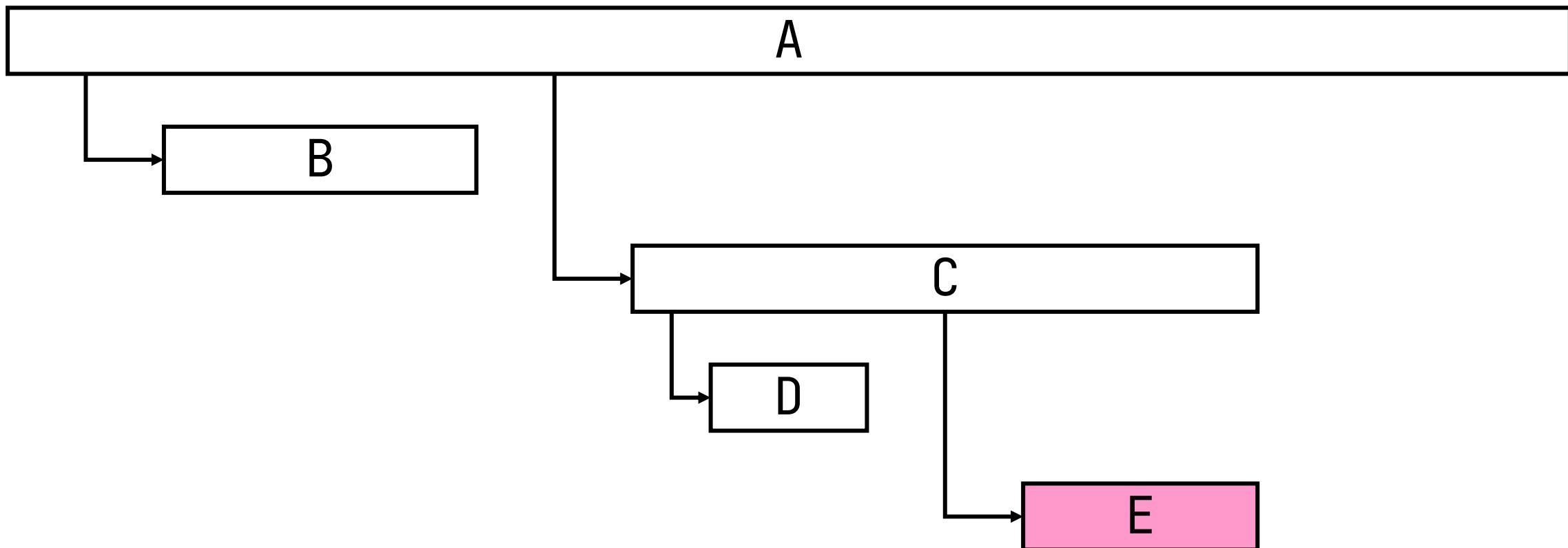
# Поиск root cause



# Поиск root cause



# Поиск root cause



# Поиск root cause

# Поиск root cause

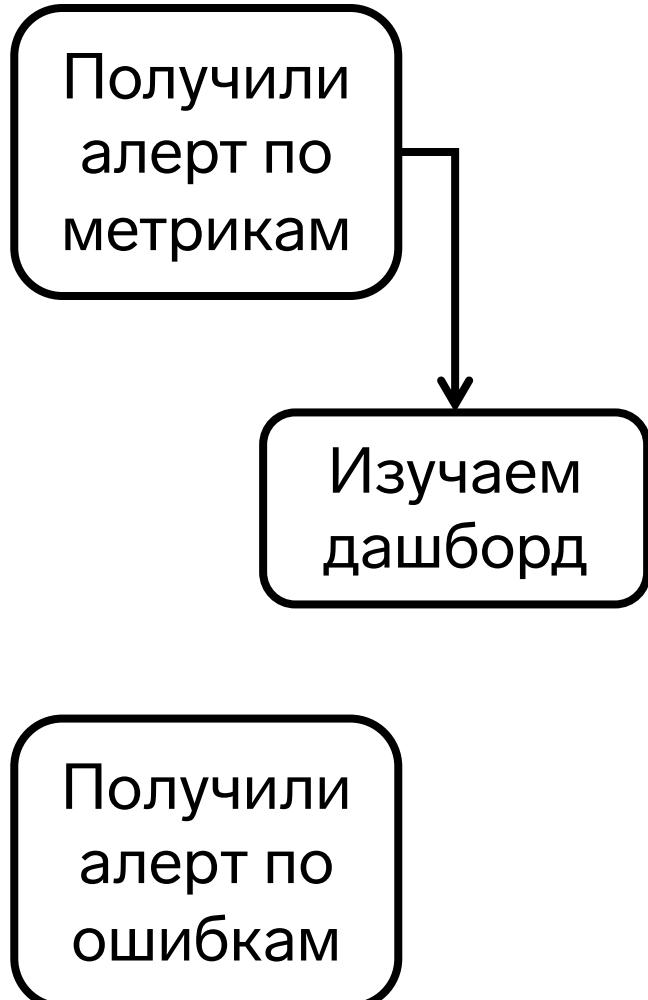
Получили  
алерт по  
метрикам

# Поиск root cause

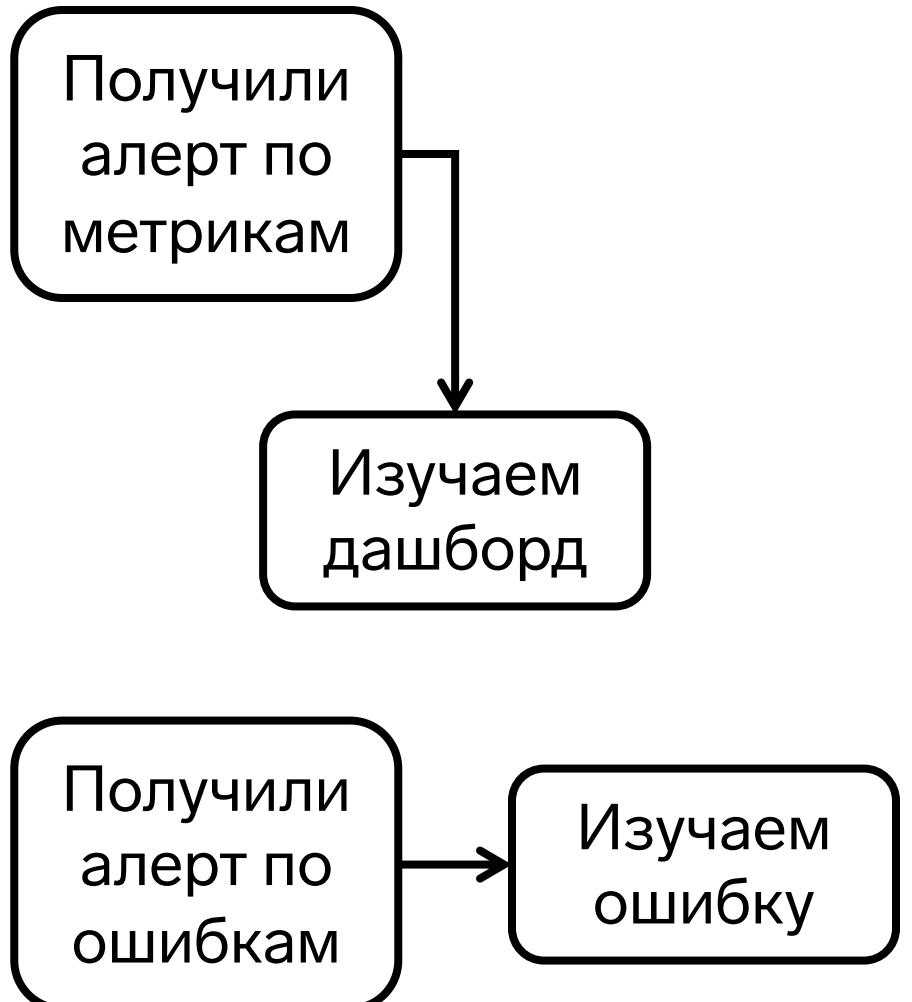
Получили  
алерт по  
метрикам

Получили  
алерт по  
ошибкам

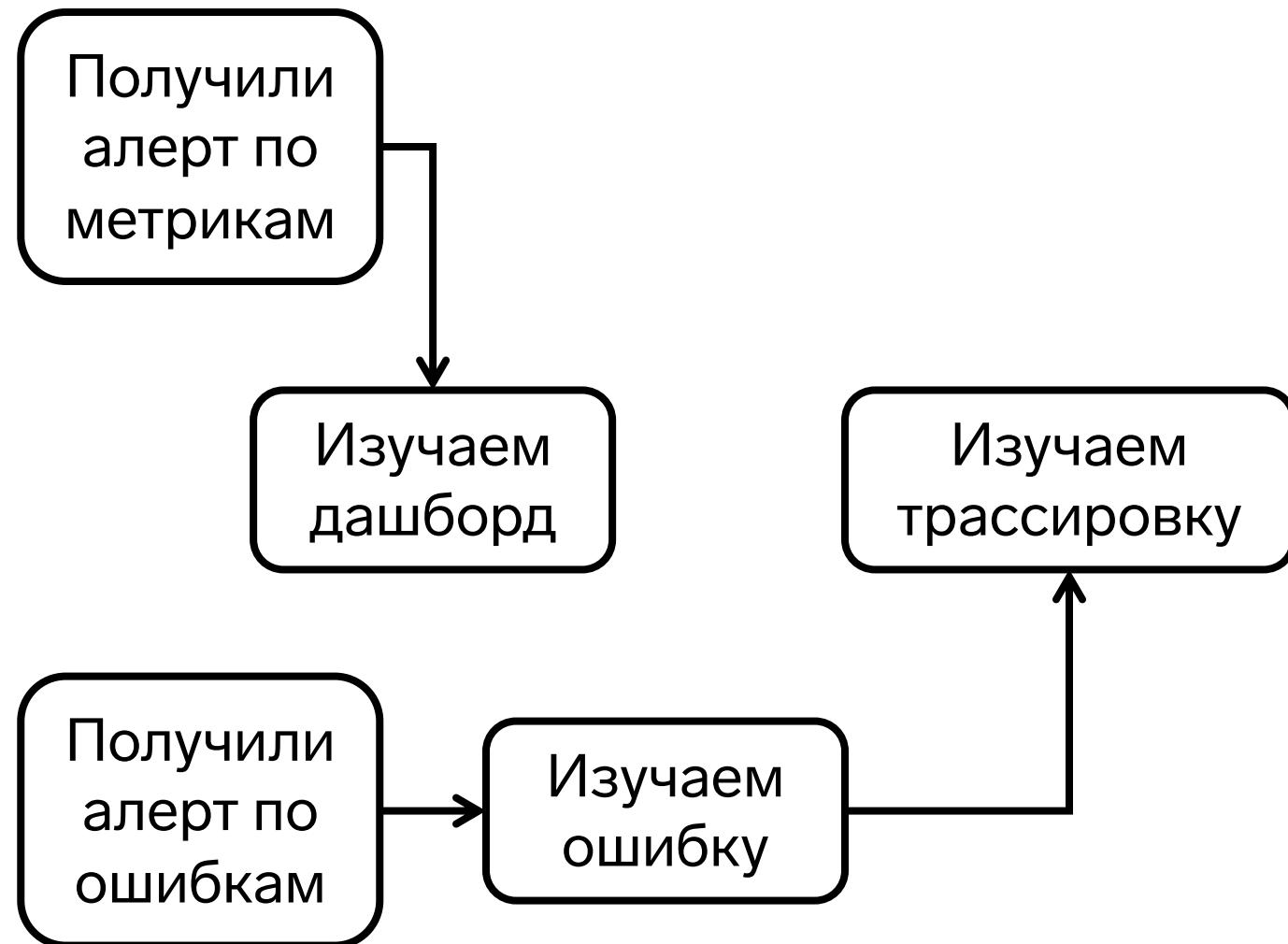
# Поиск root cause



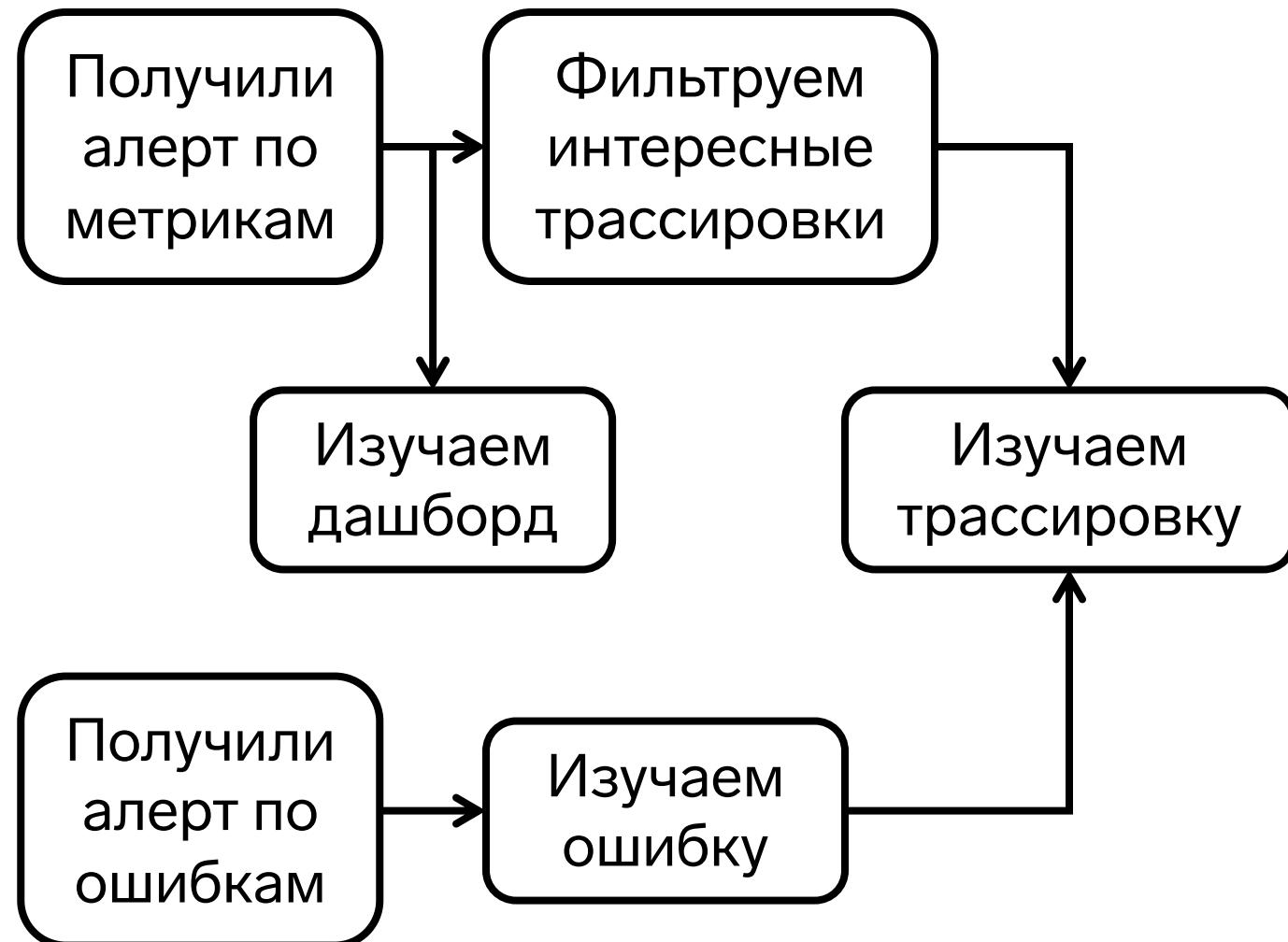
# Поиск root cause



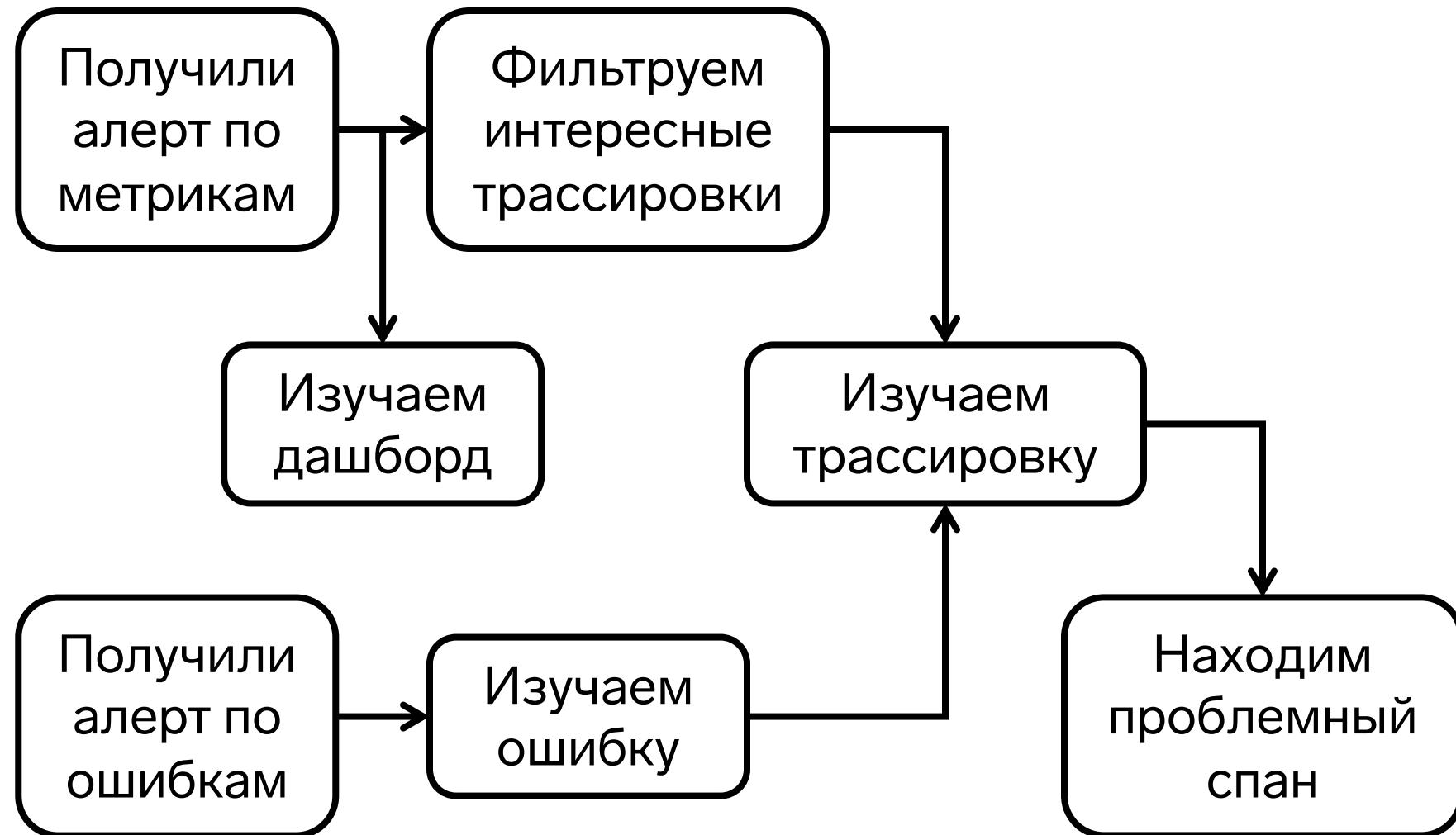
# Поиск root cause



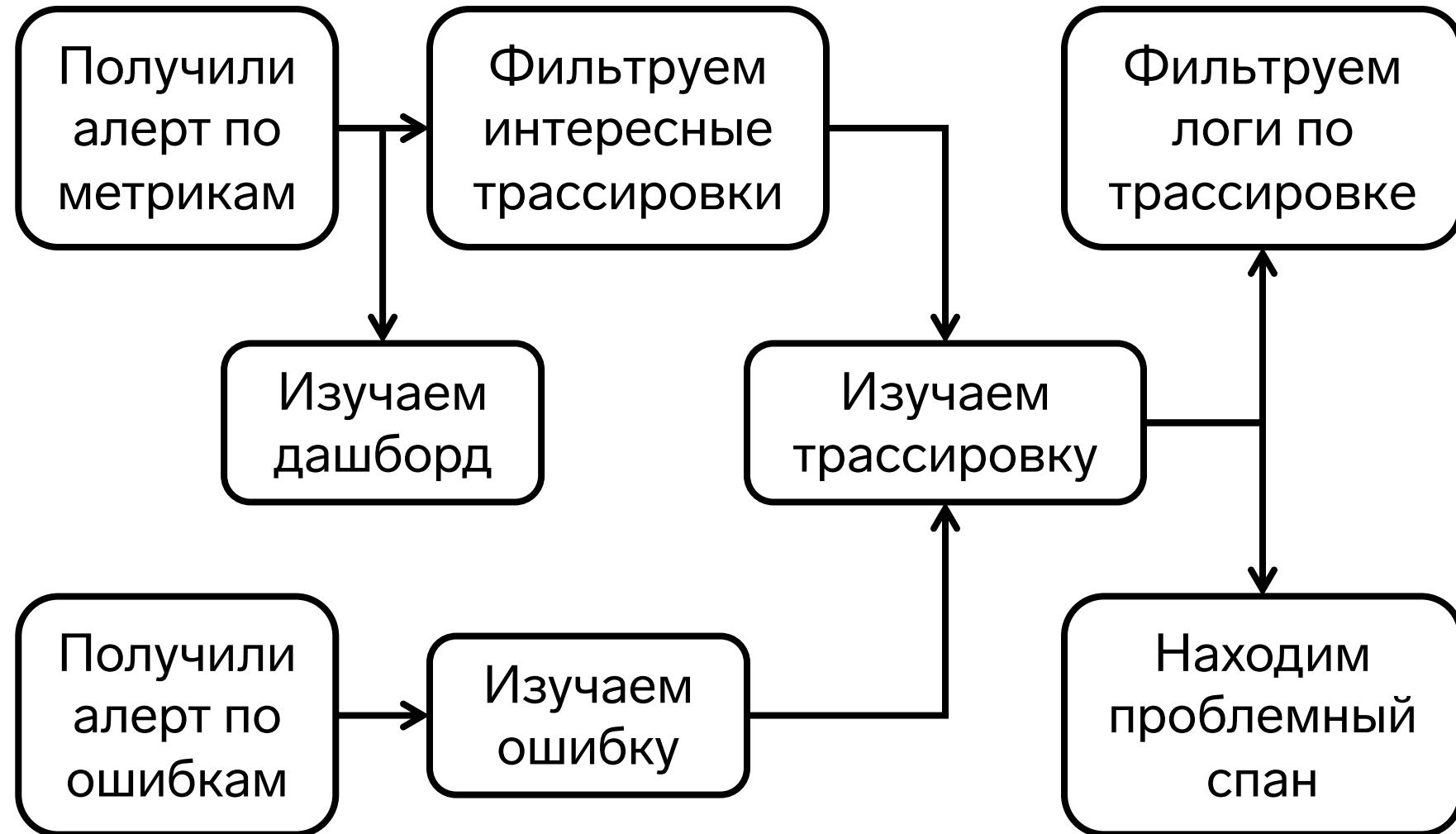
# Поиск root cause



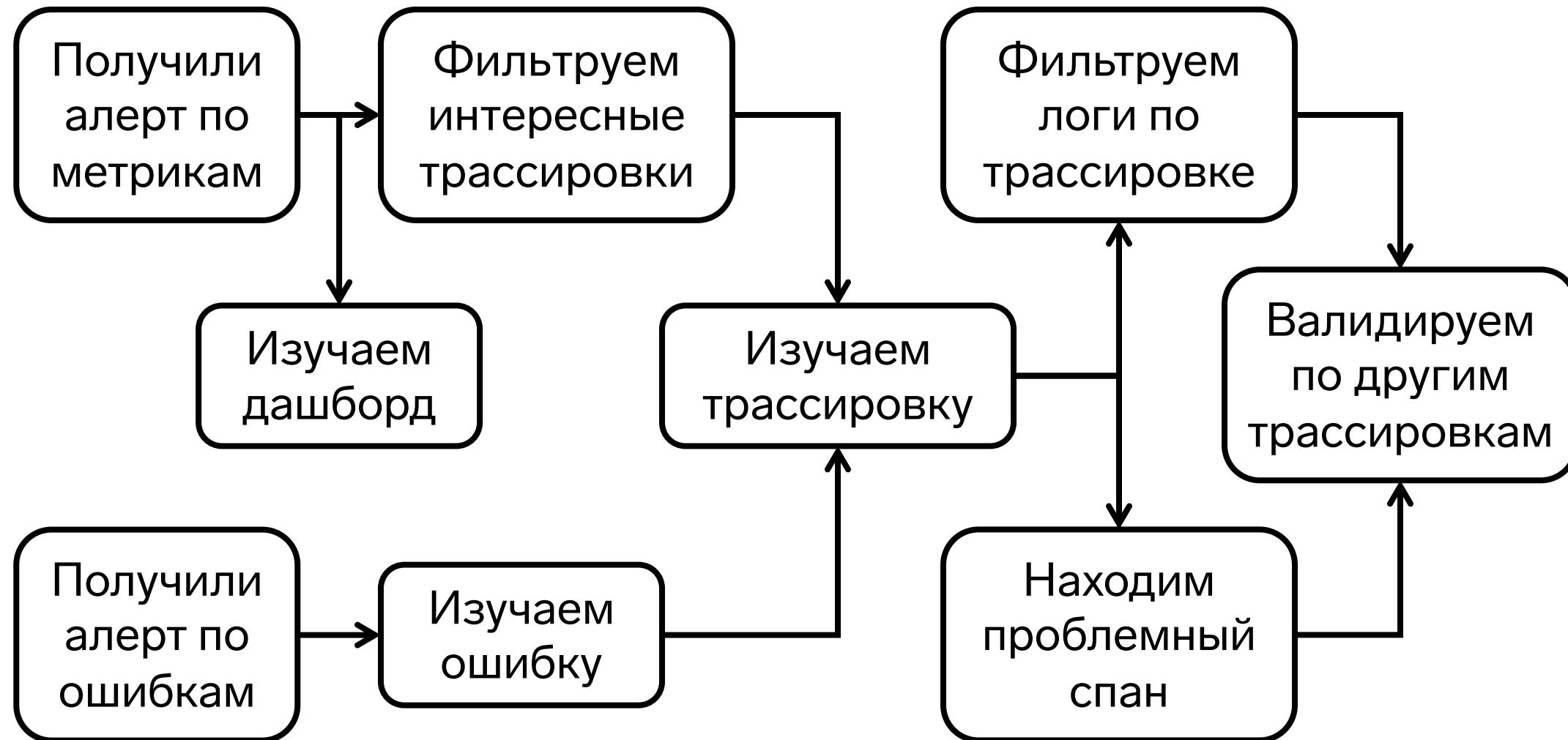
# Поиск root cause



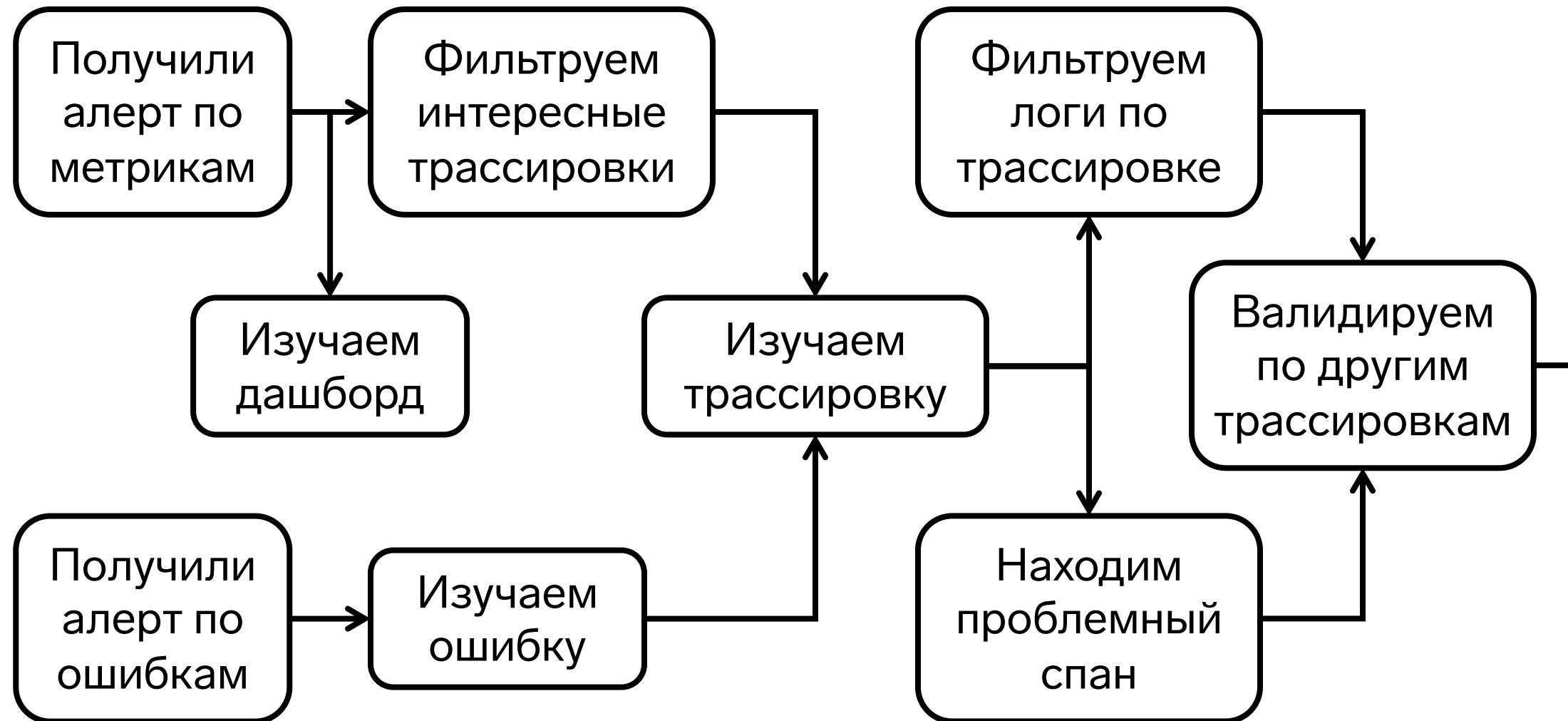
# Поиск root cause



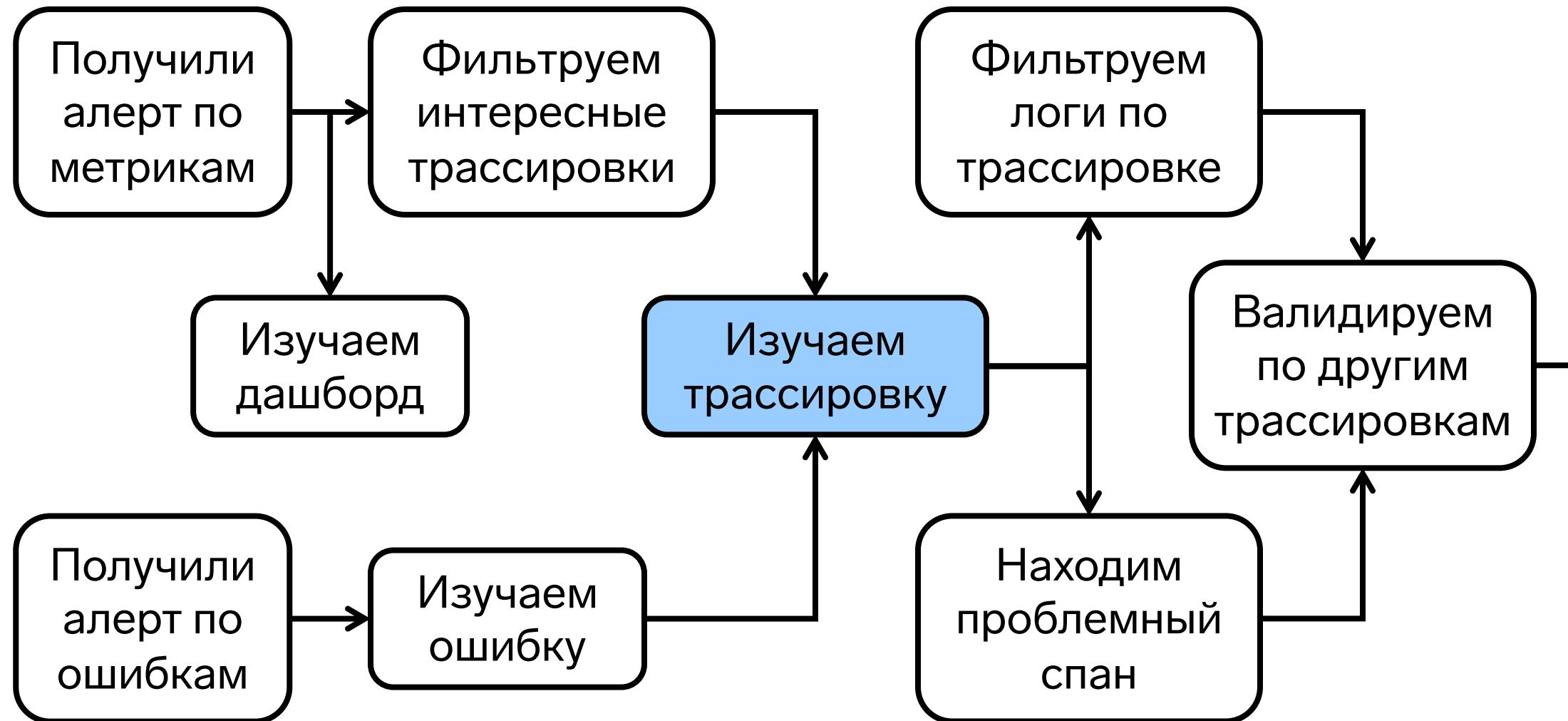
# Поиск root cause



# Поиск root cause



# Поиск root cause

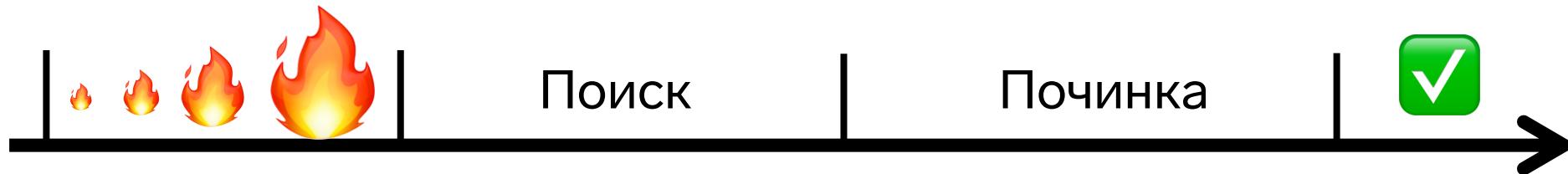


- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

# Выводы

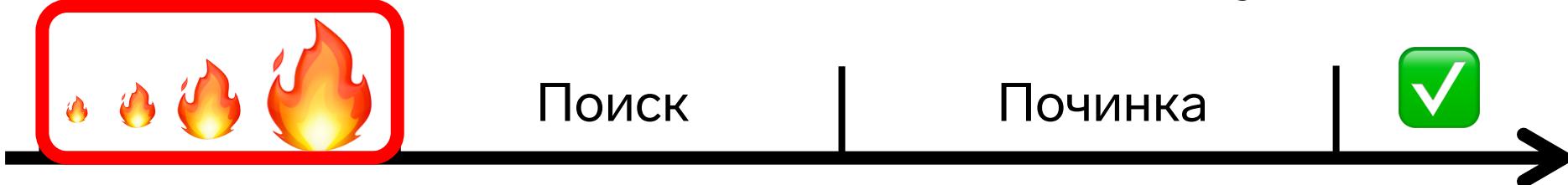
# Выводы

1. SLO определяют потребность в Observability



# Выводы

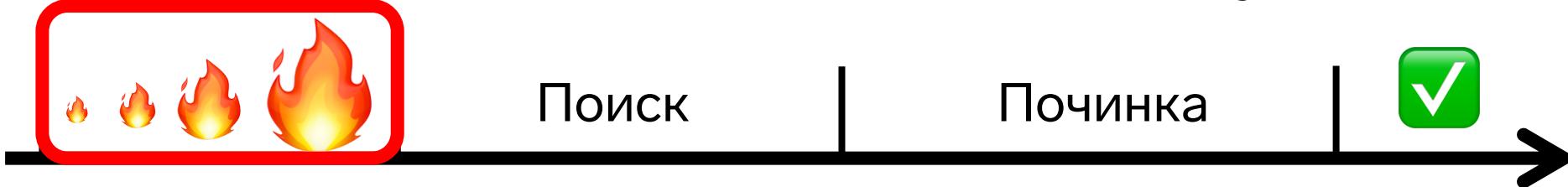
1. SLO определяют потребность в Observability



2. Аналитический подход к дашбордам и алертам

# Выводы

1. SLO определяют потребность в Observability



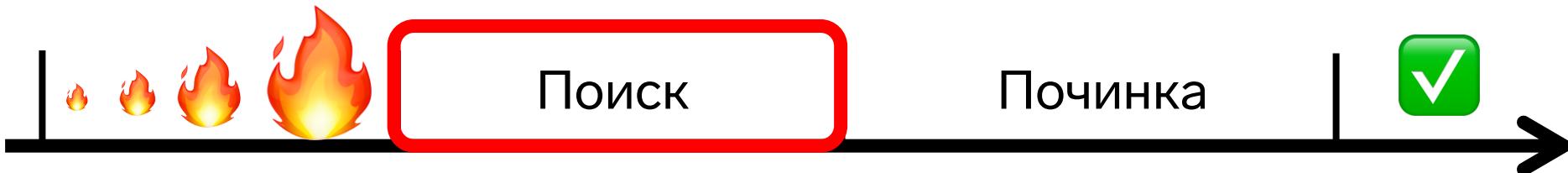
2. Аналитический подход к дашбордам и алертам



Будить или не будить?

# Выводы

1. SLO определяют потребность в Observability



2. Аналитический подход к дашбордам и алертам



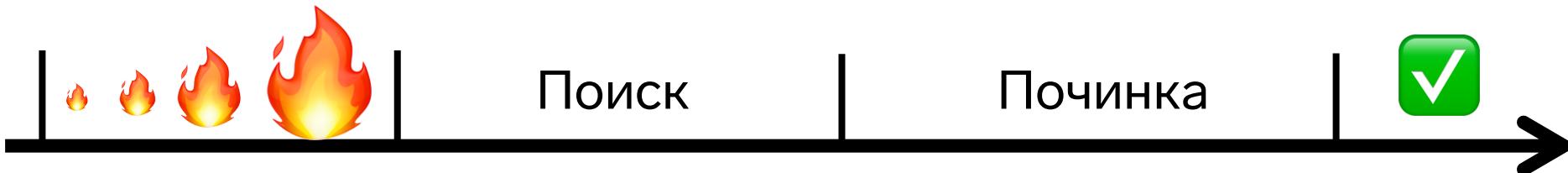
Будить или не будить?

3. Распределённые трассировки — новая нефть



# Выводы

1. SLO определяют потребность в Observability



2. Аналитический подход к дашбордам и алертам



Будить или не будить?

3. Распределённые трассировки — новая нефть



# **Спасибо за внимание!**

Другие доклады и материалы:

[https://tg.me/chnl\\_GregoryKoshelev](https://tg.me/chnl_GregoryKoshelev)



**Контур**



Григорий Кошелев  
Контур

Работа над ошибками



ие!

Другие доклады и материалы:

[https://tg.me/chnl\\_GregoryKoshelev](https://tg.me/chnl_GregoryKoshelev)



Контур



Григорий Кошелев  
Контур

Работа над ошибками



ие!

Другие доклады и материалы:

[https://tg.me/chnl\\_GregoryKoshelev](https://tg.me/chnl_GregoryKoshelev)



В чем (еще)  
польза  
распределенных  
трассировок?



Григорий  
Кошелев

Контур





Григорий Кошелев  
Контур

Работа над ошибками



и

DevOps  
2023

SRE и производительность, или Когда производительность имеет значение



Григорий Кошелев  
Контур

Другие доклады и материалы:

[https://tg.me/chnl\\_GregoryKoshelev](https://tg.me/chnl_GregoryKoshelev)



В чем (еще) польза распределенных трассировок?



Григорий Кошелев

Контур



# Спасибо за внимание!

<https://tech.kontur.ru>



Контур