Антропоориентированный мониторинг

Обо мне

* Меня зовут Виталий Слободин
* Senior Engineer в GitLab
* Фанат чистого кода
* "Стабильный продакшн - спокойный инженер"

Антропоориентированный мониторинг

Антропоморфный

по форме, устройству схожий с человеком, его телом; человекообразный; также основанный на таком сходстве.

Это история об очеловечивании системы мониторинга

Что было до (2017 год)

* Переезд на prometheus

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры
* 24 правила для уведомлений

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры
* 24 правила для уведомлений
* 10 правил записи

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры
* 24 правила для уведомлений
* 10 правил записи
* 21 дашбоард grafana

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры
* 24 правила для уведомлений
* 10 правил записи
* 21 дашбоард grafana
* около 2000 различных метрик

Что было до (2017 год)

* Переезд на prometheus
* 5 инженеров инфраструктуры
* 24 правила для уведомлений
* 10 правил записи
* 21 дашбоард grafana
* около 2000 различных метрик
* ~100к семплов в секунду

Что есть сейчас (март 2020)

Что есть сейчас (март 2020)

* Prometheus до сих пор с нами

Что есть сейчас (март 2020)

* Prometheus до сих пор с нами
* Каждое окружение содержит свои собственные серверы Prometheus

Что есть сейчас (март 2020)

* Prometheus до сих пор с нами
* Каждое окружение содержит свои собственные серверы Prometheus
* Отдел инфраструктуры с 40 инженерами

Что есть сейчас (март 2020)

* Prometheus до сих пор с нами
* Каждое окружение содержит свои собственные серверы Prometheus
* Отдел инфраструктуры с 40 инженерами
* 280 правил для уведомлений

Что есть сейчас (март 2020)

* 800 правил сбора данных

Что есть сейчас (март 2020)

* 800 правил сбора данных
* около 80к различных метрик

Что есть сейчас (март 2020)

* 800 правил сбора данных
* около 80к различных метрик
* 431 дашбоард Grafana

Что есть сейчас (март 2020)

* 800 правил сбора данных
* около 80к различных метрик
* 431 дашбоард Grafana
* 10 миллионов данных в секунду

Как мы пришли к этому?

Зачем?

kubernetes
11004
kubernetes
4+
11004
@datamattsson
IDEAS INCLUDED
Pod
Service
StatefulSet
DaemonSet
Deployment

Ничто не вечно

(кроме PHP)

Оптимизация это постоянный процесс.

Зачем?

Масштабирование приложения (- |)

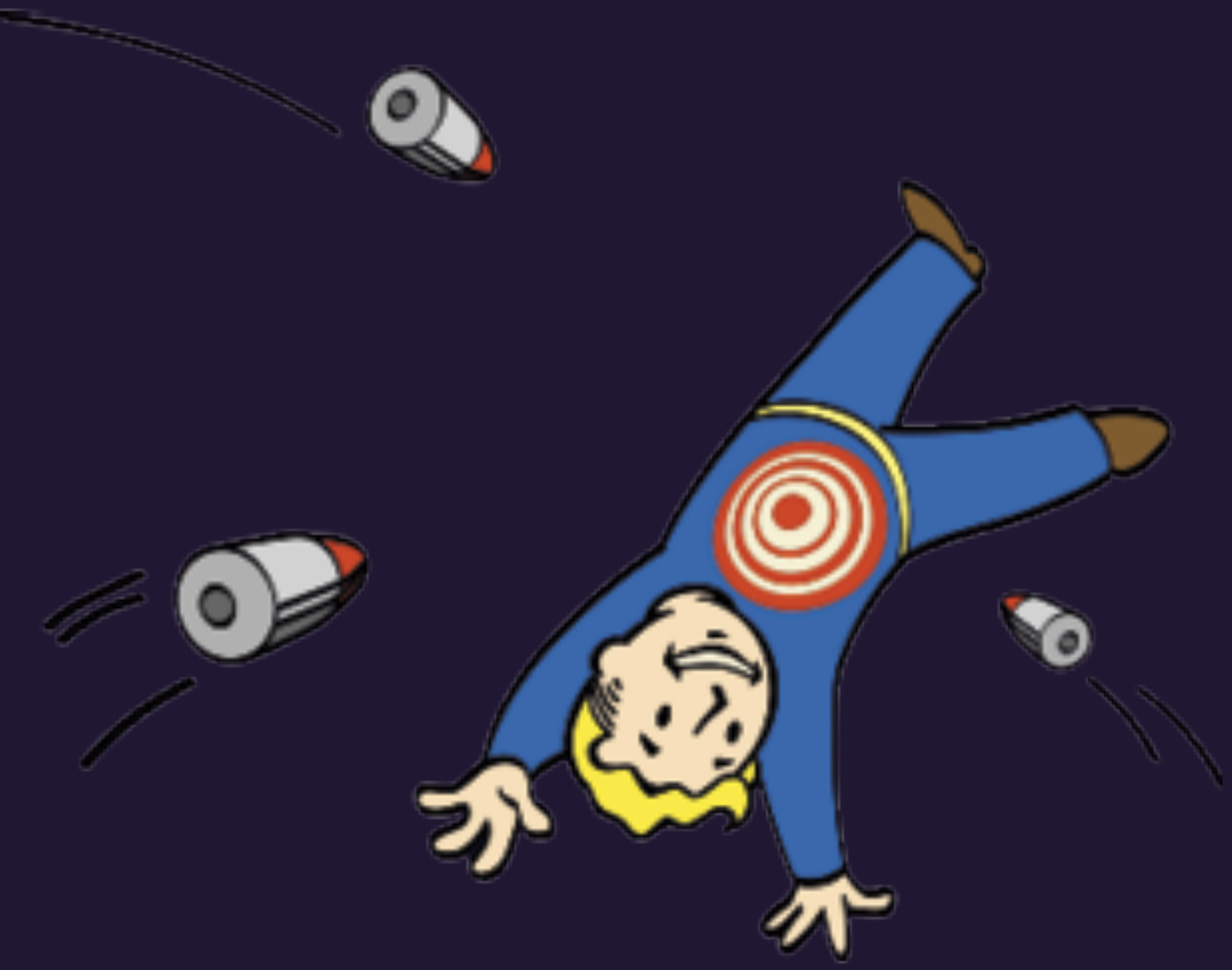
Зачем?

Система мониторинга не должна быть мешать работе

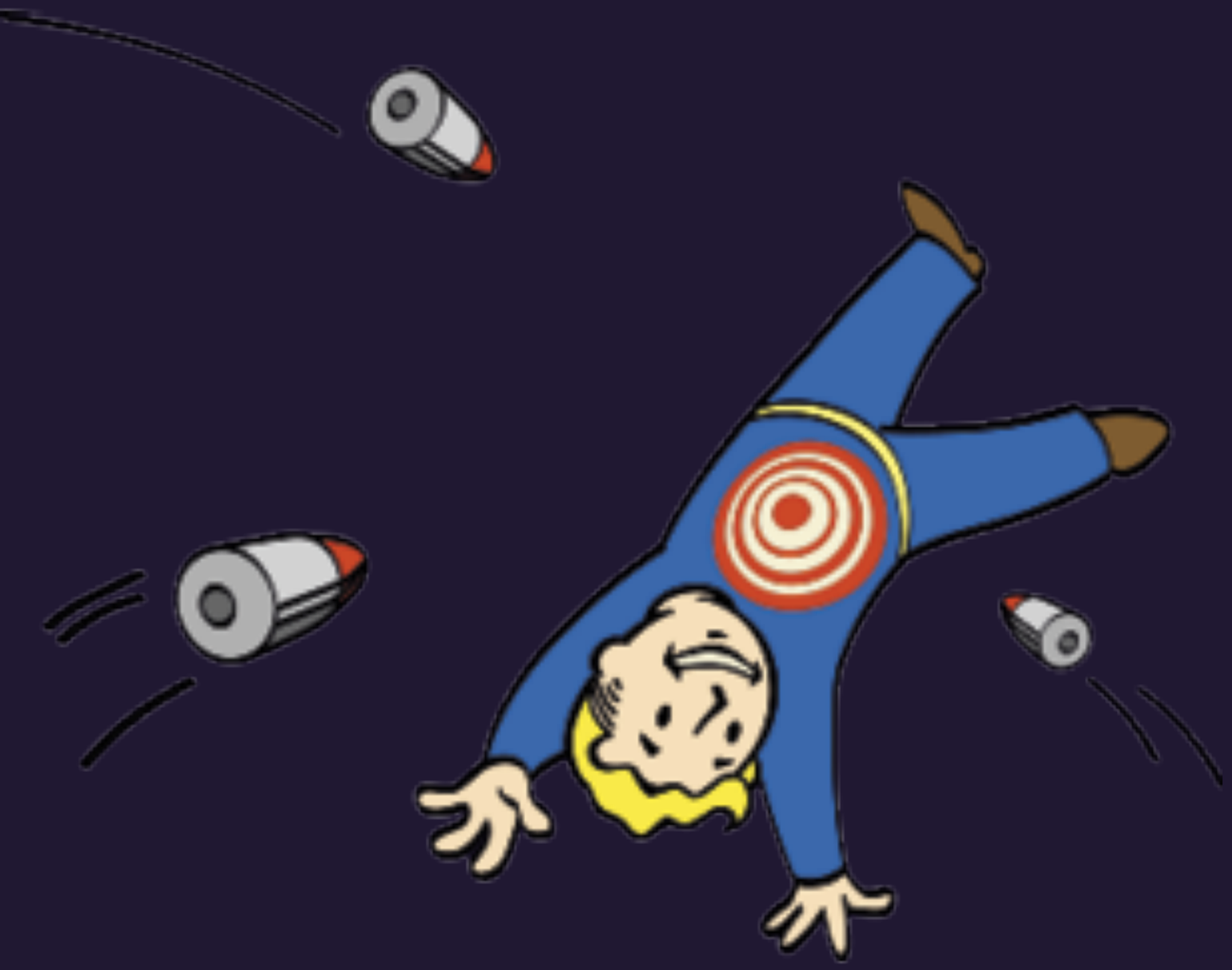
Зачем?

Система мониторинга должна быть помощником

Как мы пришли к этому?

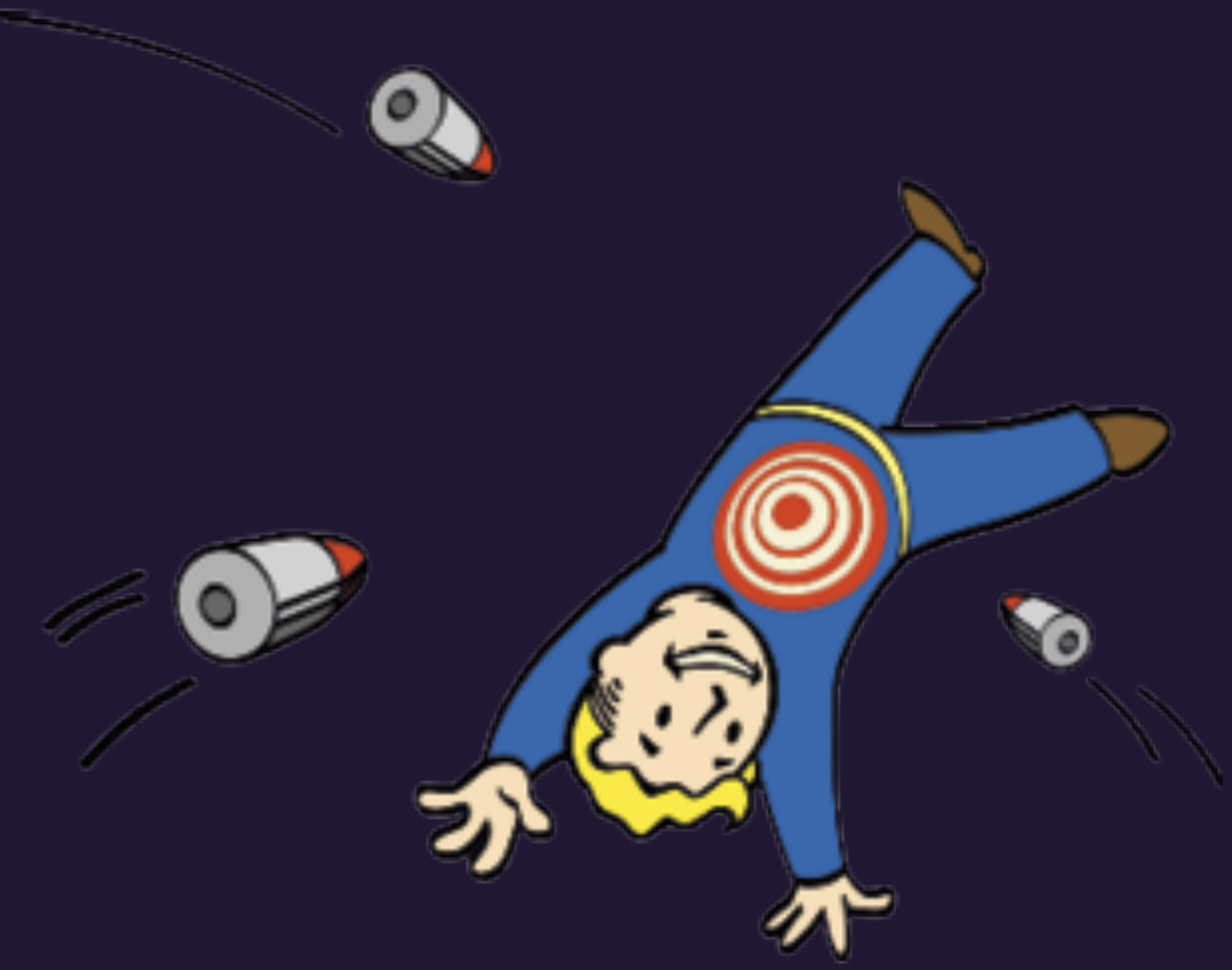


Проблема: цель постоянно двигалась



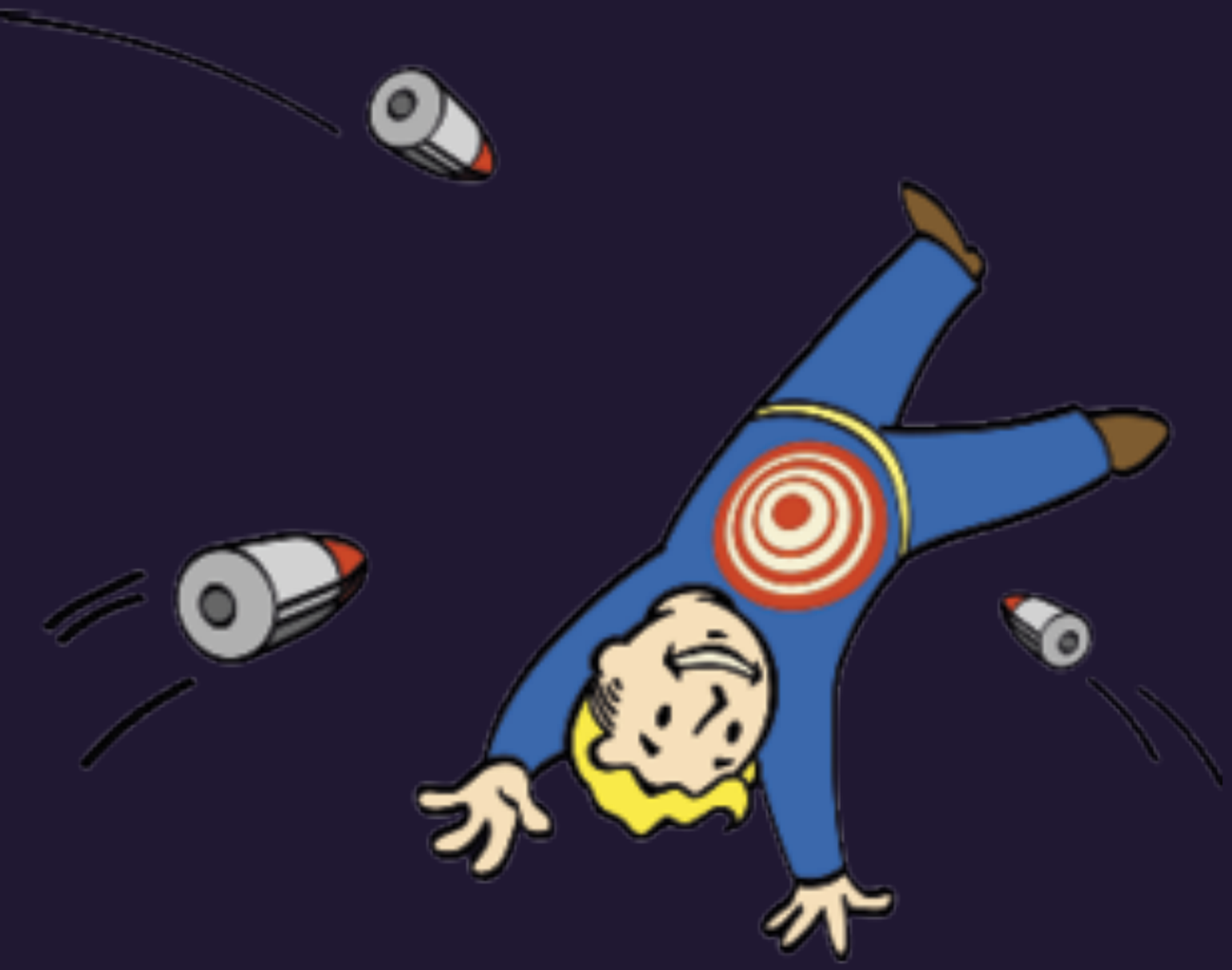
Проблема: цель постоянно двигалась

* Каждый инцидент имел новую причину



Проблема: цель постоянно двигалась

* Каждый инцидент имел новую причину
* Каждый деплой - новые места отказа



Проблема: цель постоянно двигалась

* Каждый инцидент имел новую причину
* Каждый деплой – новые места отказа
* Инфраструктура меняется тоже

Низкое качество уведомлений

Низкое качество уведомлений

* постоянно срабатывали

Низкое качество уведомлений

* постоянно срабатывали
* сложно определить критичность

Низкое качество уведомлений

* постоянно срабатывали
* сложно определить критичность
* "усталость" уведомлений

Низкое качество уведомлений

* постоянно срабатывали
* сложно определить критичность
* "усталость" уведомлений
* "АСК!"

Система уведомлений на пользователях

Guilherme Domingues @dominguesgg · Aug 17, 2018

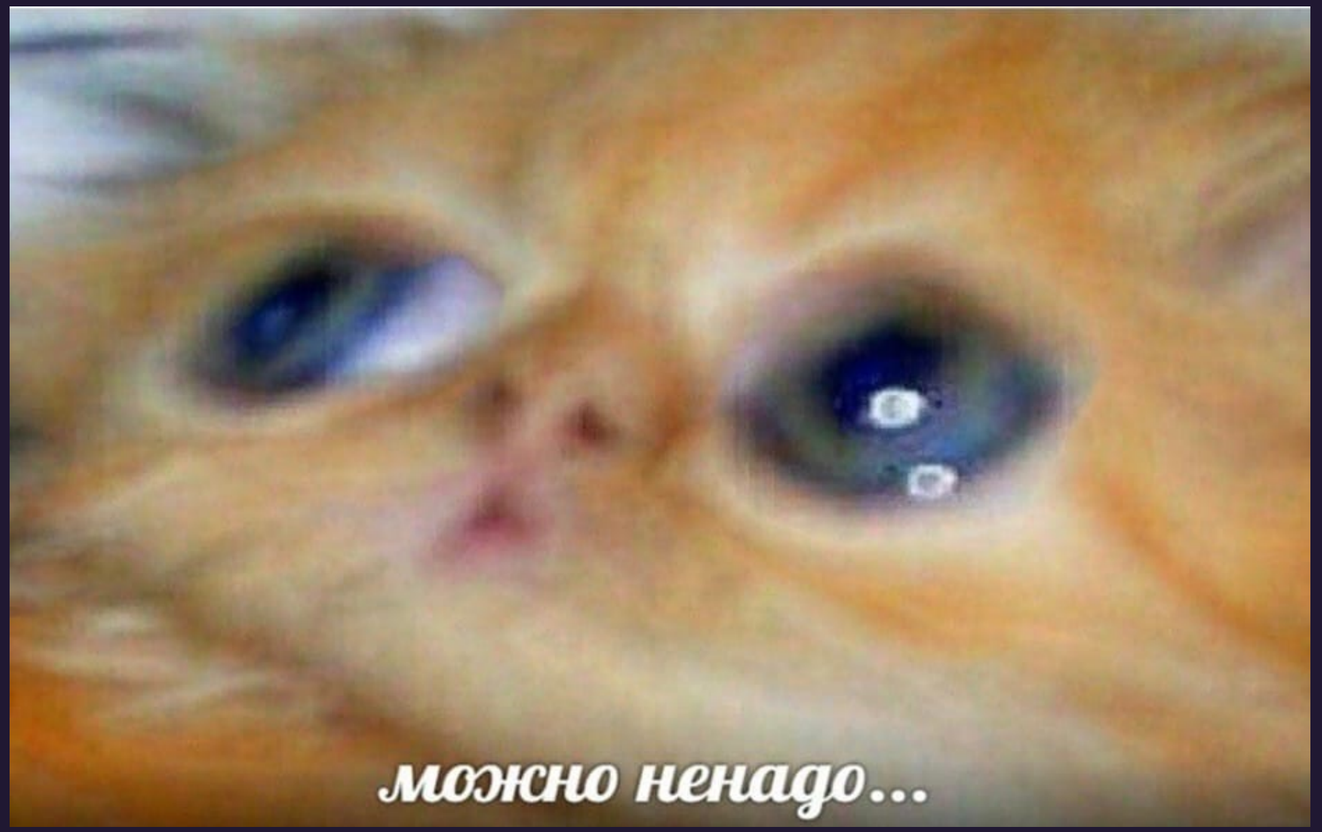
Is @gitlab down? I'm gettin' 503 Service Unavailable Error. But @gitlabstatus says that everything is fine!

1

Система уведомлений на пользователях

* используйте пользователей вместо системы мониторинга
* сообщения коллег в Slack
* твиттер
* инстаграмм

Система уведомлений на пользователях



Наша система уведомлений не масштабируется вместе с бизнесом

But Wait...
THERE'S MORE!!!

Устаревающие сигналы

* никогда не срабатывают

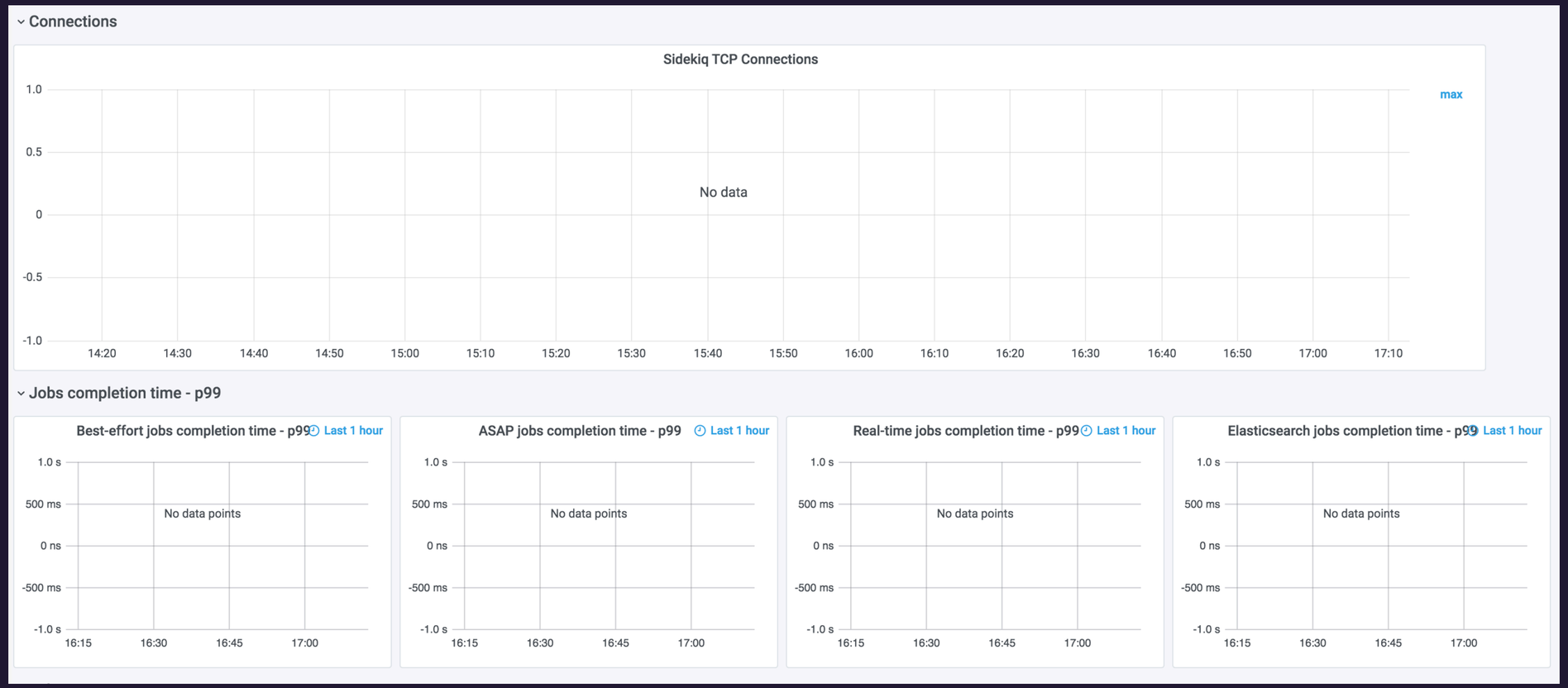
Устаревающие сигналы

* никогда не срабатывают
* работают на метриках, которых уже давно нет

Устаревающие сигналы

* никогда не срабатывают
* работают на метриках, которых уже давно нет
* техдолг

Сломанные дашбоарды



Сломанные дашбоарды

* метрики поменялись

Сломанные дашбоарды

* метрики поменялись
* забыли обновить

Сломанные дашбоарды

* метрики поменялись
* забыли обновить
* обнаружили во время инцидента

Все плохо?

No! Мы знаем наши проблемы

No! Мы знаем наши проблемы

* метрики приложения и платформы

No! Мы знаем наши проблемы

* метрики приложения и платформы
* сигналы и уведомления

No! Мы знаем наши проблемы

* метрики приложения и платформы
* сигналы и уведомления
* дашбоарды

Вспомним цели

* Масштабирование приложения (- |)
* Система мониторинга не должна мешать работе
* Система мониторинга должна быть помощником

Метрики

4 золотых сигнала Google[3]

[3] https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems

4 золотых сигнала Google[3]

* latency

[3] https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems

4 золотых сигнала Google[3]

* latency
* трафик

[3] https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems

4 золотых сигнала Google[3]

* latency
* трафик
* ошибки

[3] https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems

4 золотых сигнала Google[3]

* latency
* трафик
* ошибки
* насыщенность

[3] https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems

GitLab.com's "ARES"

* Apdex (%)

GitLab.com's "ARES"

* Apdex (%)
* RPS

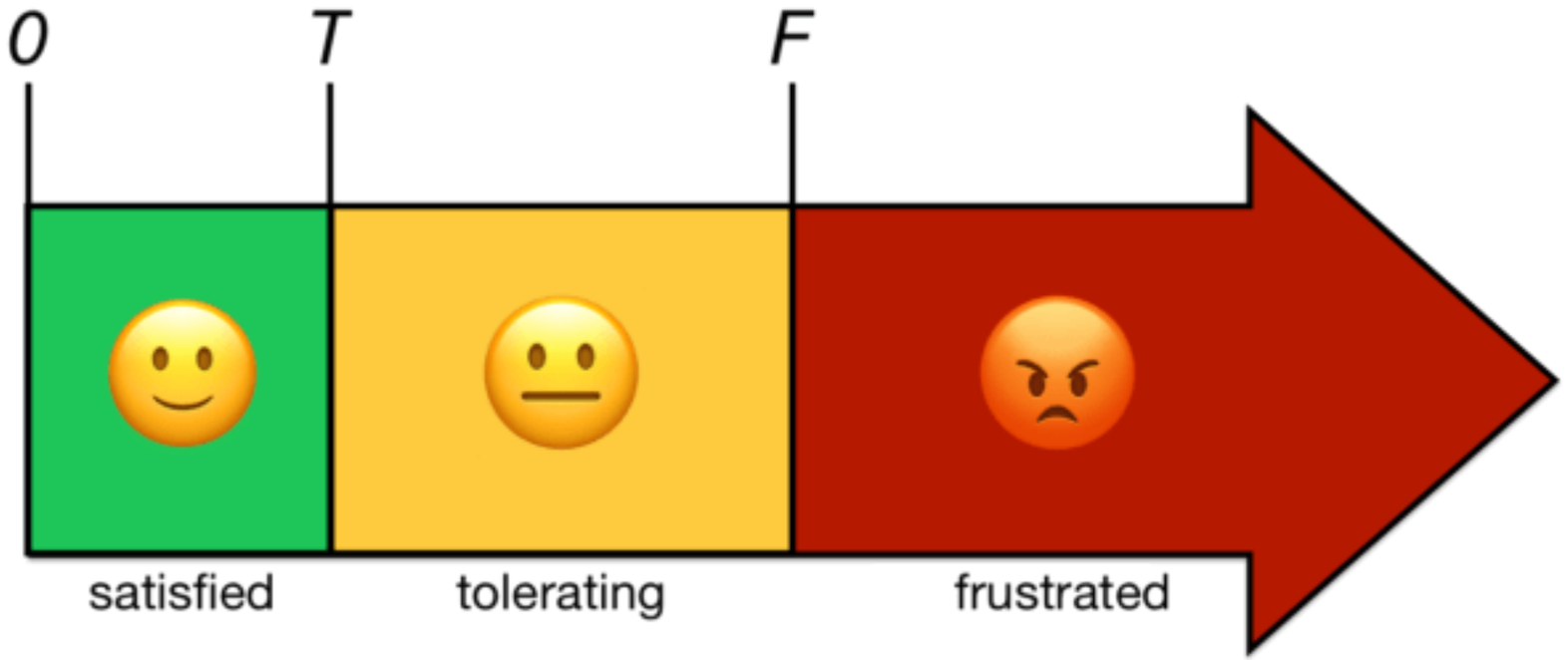
GitLab.com's "ARES"

* Apdex (%)
* RPS
* Errors (%)

GitLab.com's "ARES"

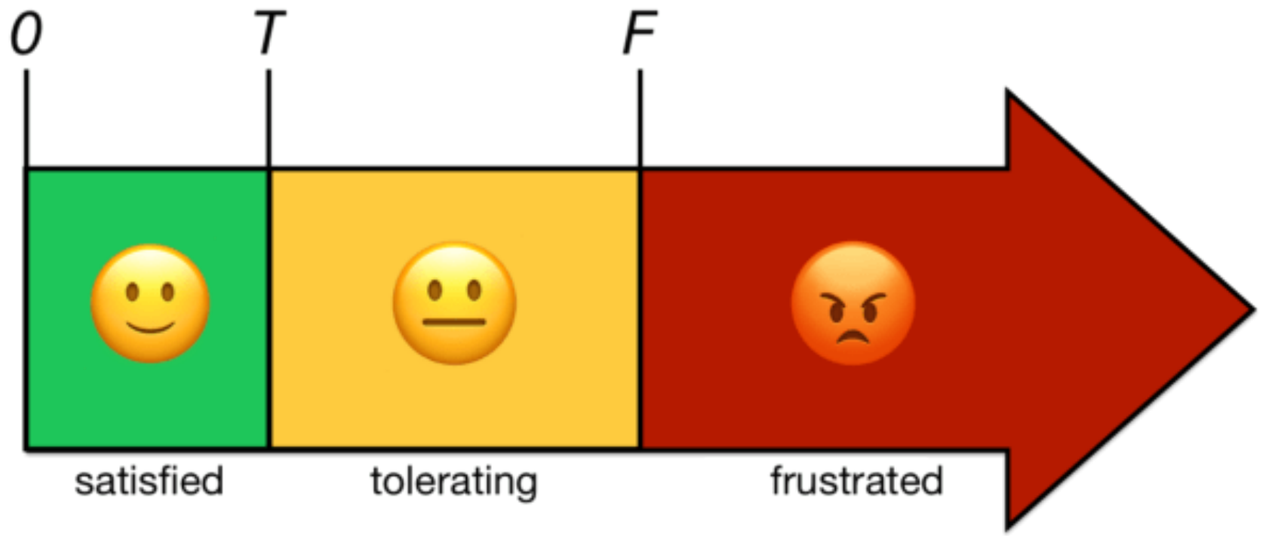
* Apdex (%)
* RPS
* Errors (%)
* Saturation (%)

Apdex

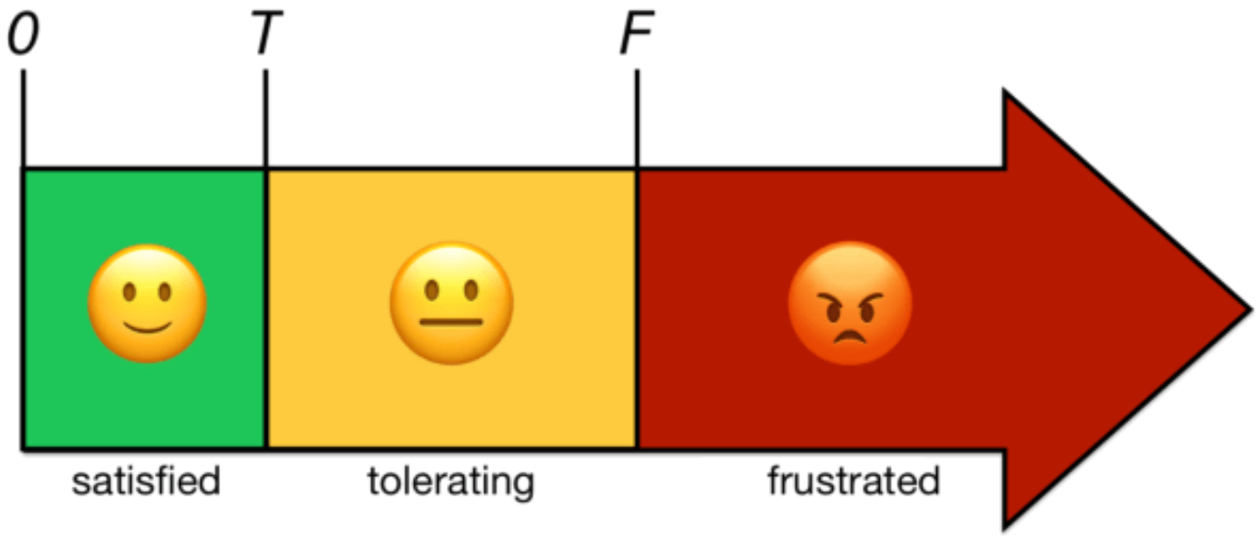


Apdex

* 2 порога, 3 значения

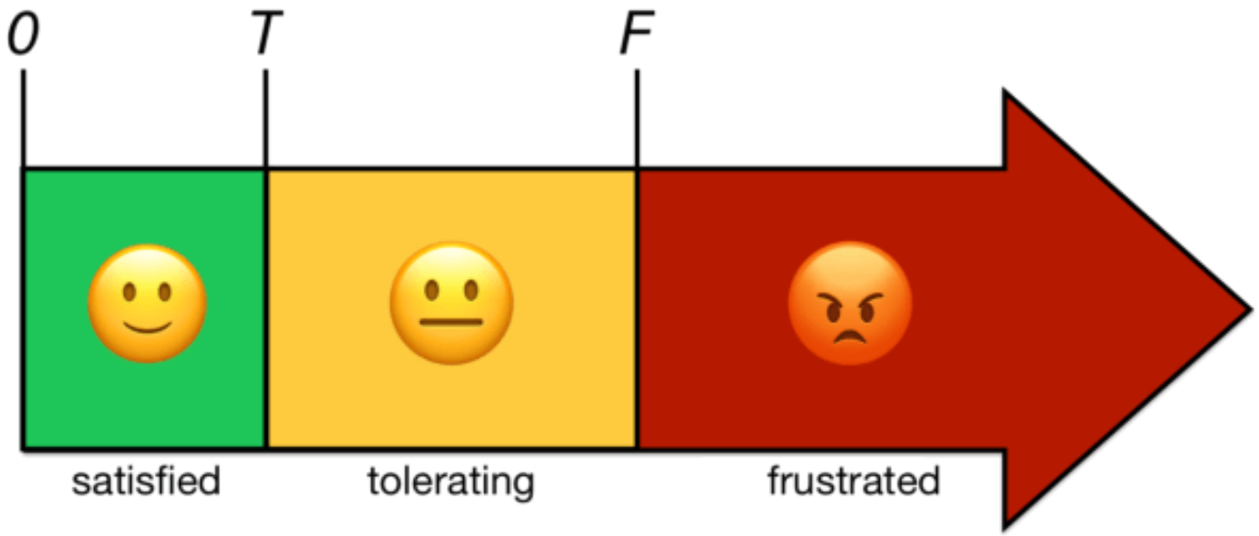


Apdex



* 2 порога, 3 значения
* Среднее значение за отрезок времени

Apdex



* 2 порога, 3 значения
* Среднее значение за отрезок времени
* Пороговые значения могут быть для сервиса или end point

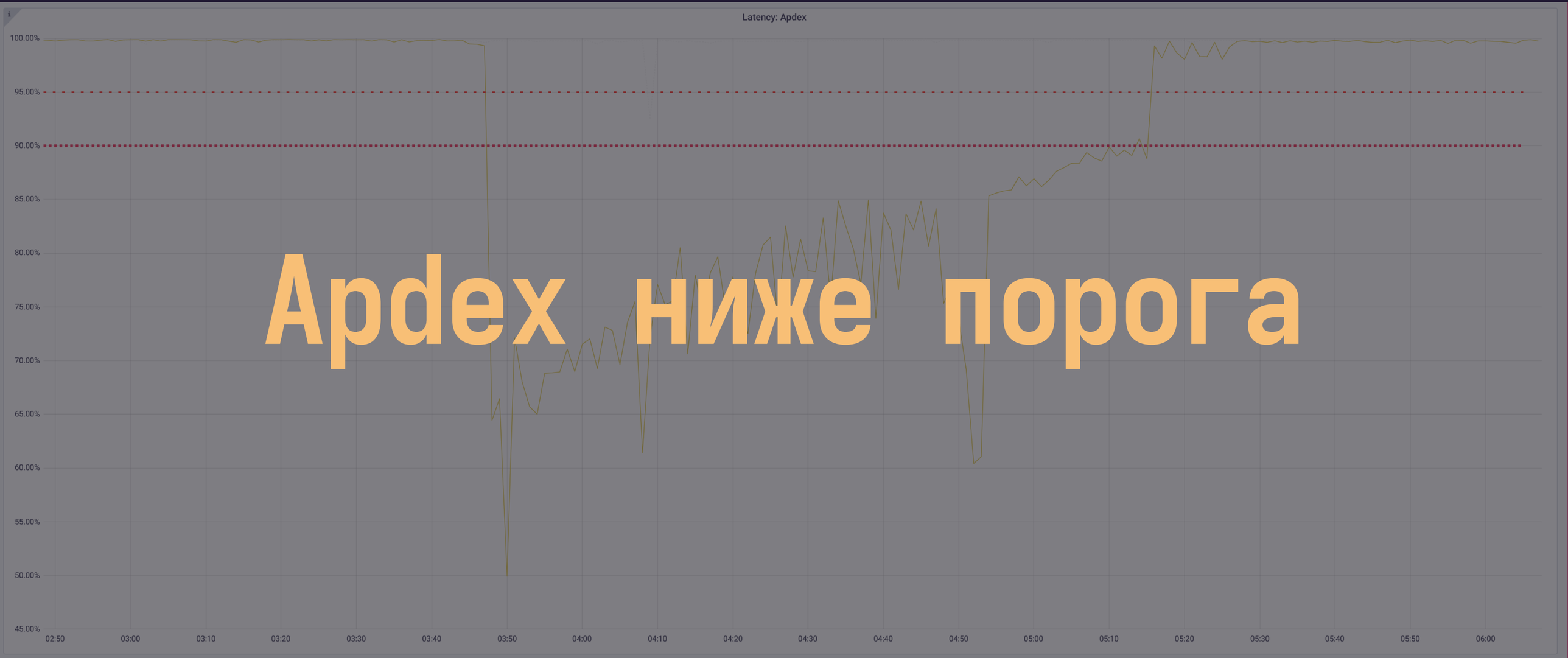
Как улучшить качество уведомлений?

ARES дает нам 4 главных сигнала

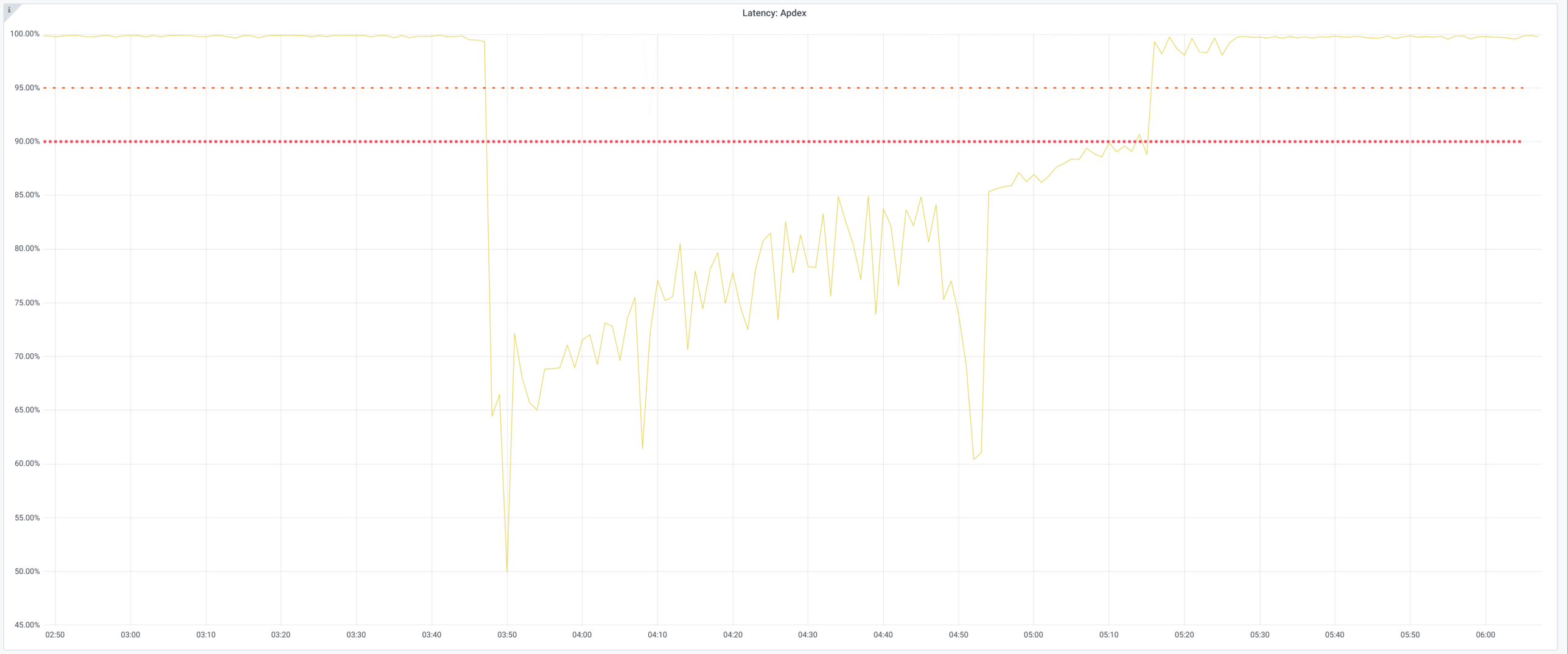
ARES дает нам 4 главных сигнала

два - симптомы

два - причина



Apdex ниже порога


Latency: Apdex
100.00%
95.00%
90.00%
85.00%
80.00%
75.00%
70.00%
65.00%
60.00%
55.00%
50.00%
45.00%
02:50
03:00
03:10
03:20
03:30
03:40
03:50
04:00
04:10
04:20
04:30
04:40
04:50
05:00
05:10
05:20
05:30
05:40
05:50
06:00

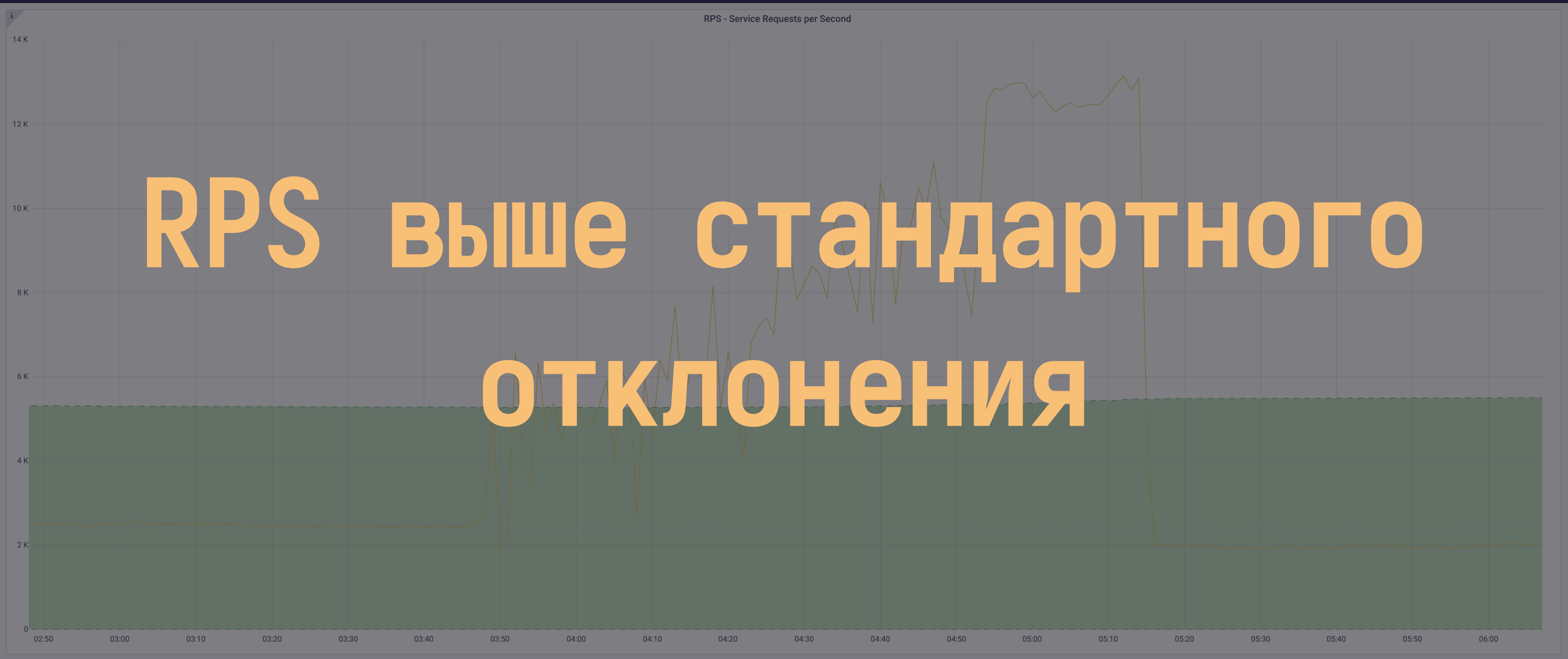
Apdex ниже порога

Apdex одного веб-сервиса ниже 99.9% в течение 5 минут

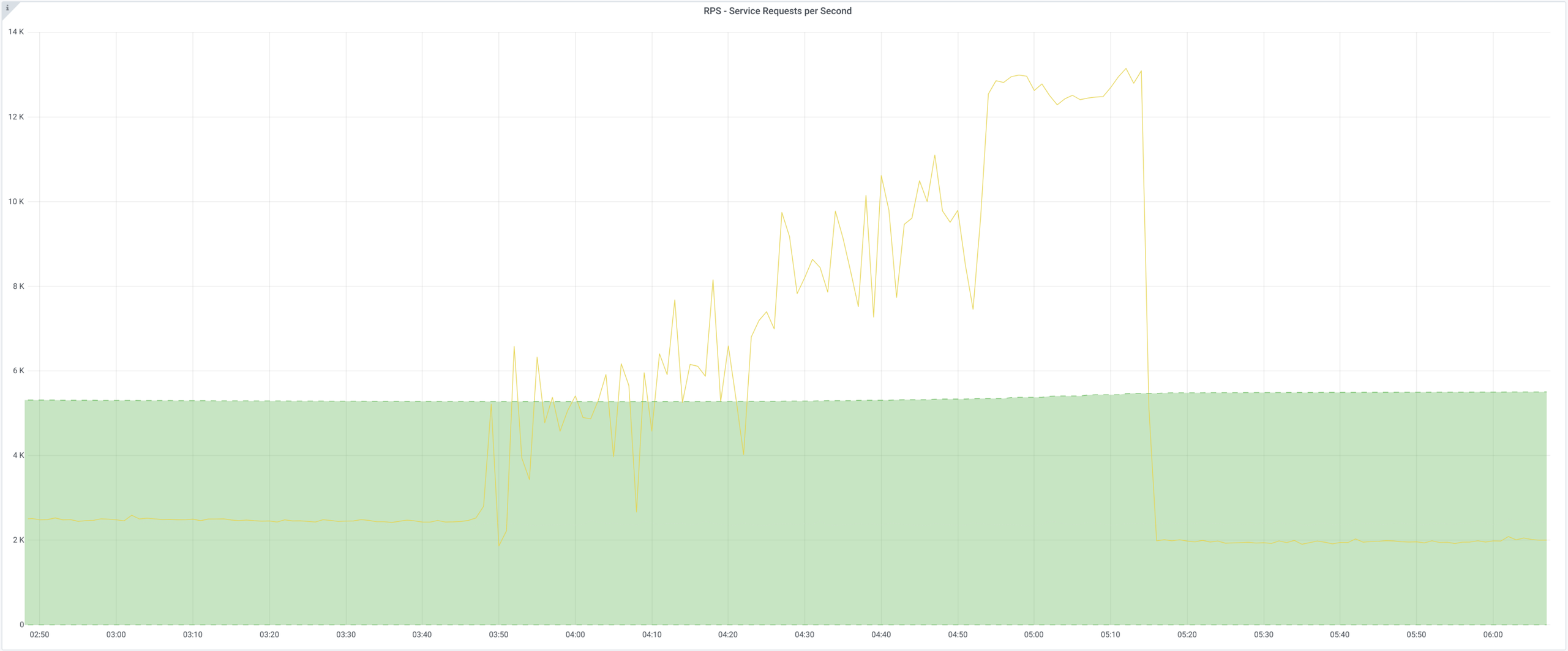
Apdex ниже порога

Apdex одного веб-сервиса ниже 99.9% в течение 5 минут

Это симптом



RPS выше стандартного отклонения


RPS - Service Requests per Second
14 K
12 K
10 K
8 K
6 K
4 K
2 K
0
02:50
03:00
03:10
03:20
03:30
03:40
03:50
04:00
04:10
04:20
04:30
04:40
04:50
05:00
05:10
05:20
05:30
05:40
05:50
06:00

RPS выше стандартного отклонения

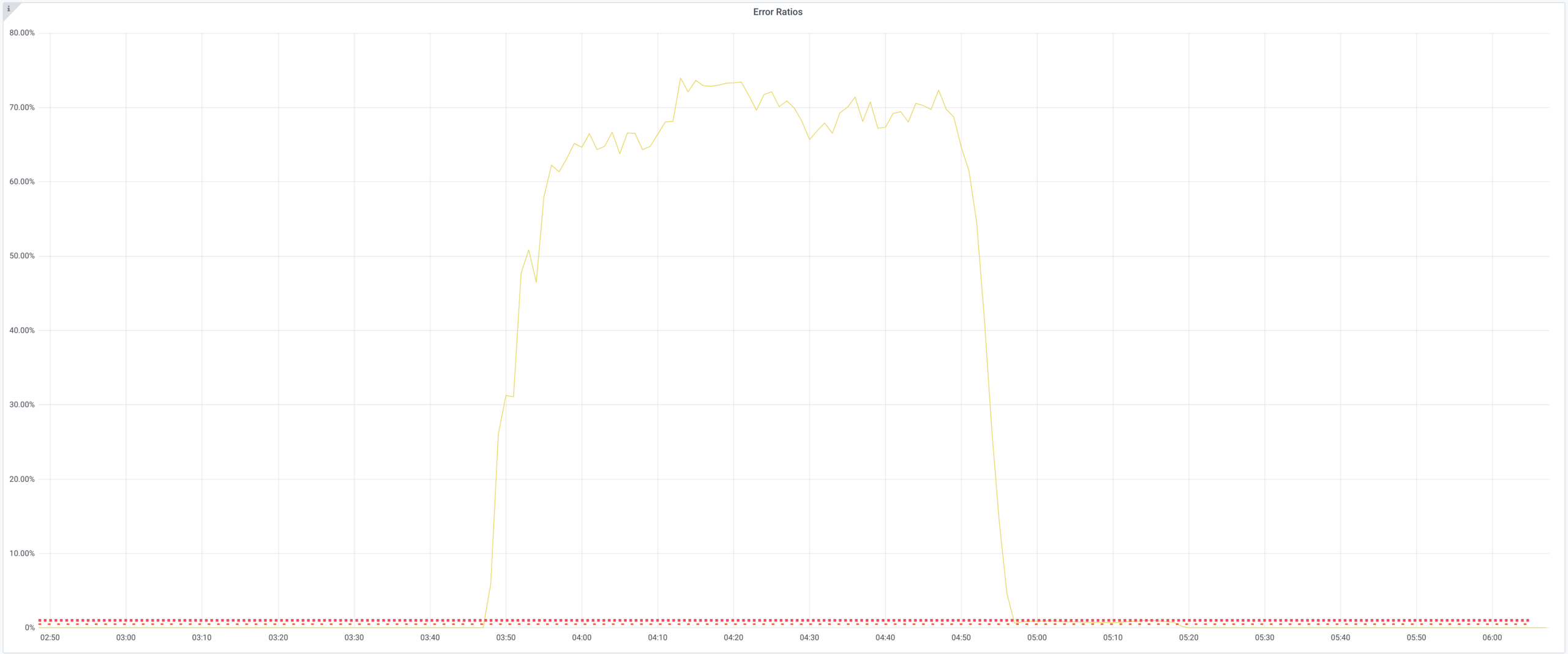
Кол-во запросов вне зоны тройного стандартного отклонения (3σ)

RPS выше стандартного отклонения

Кол-во запросов вне зоны тройного стандартного отклонения (3σ)

Это причина

Ошибок выше крыши


Error Ratios
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0%
02:50
03:00
03:10
03:20
03:30
03:40
03:50
04:00
04:10
04:20
04:30
04:40
04:50
05:00
05:10
05:20
05:30
05:40
05:50
06:00

Ошибок выше крыши

Кол-во ошибок выше порога 0.1% в течение 5 минут

Ошибок выше крыши

Кол-во ошибок выше порога 0.1% в течение 5 минут

Это причина

Насыщенность выше порога

Saturation
100.0%
90.0%
80.0%
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
10.0%
0%
02:50
03:00
03:10
03:20
03:30
03:40
03:50
04:00
04:10
04:20
04:30
04:40
04:50
05:00
05:10
05:20
05:30
05:40
05:50
06:00

Насыщенность выше порога

Это причина

ARES дает 2 сигнала-симптома

* Apdex
* Error rate

ARES дает 2 сигнала-симптома

Что у пользователя проблемы

ARES дает 2 сигнала-причины

* RPS
* Saturation

ARES дает 2 сигнала-причины

У пользователя могут быть проблемы

Первый подход для улучшения сигналов

```
- alert: ApdexOutOfBounds
  expr: |
    gitlab_component_apdex:ratio_1m < 0.999
  for: 5m
  labels:
    alerttype: symptom
  annotations:
    description: |
      The `{{ $labels.service }}` service is not meeting its lat
ency SLOs
```

```
- alert: ApdexOutOfBounds
  expr: |
    gitlab_component_apdex:ratio_1m < 0.999
  for: 5m
  labels:
    alerttype: symptom
  annotations:
    description: |
      The `{{ $labels.service }}` service is not meeting its lat
ency SLOs
```

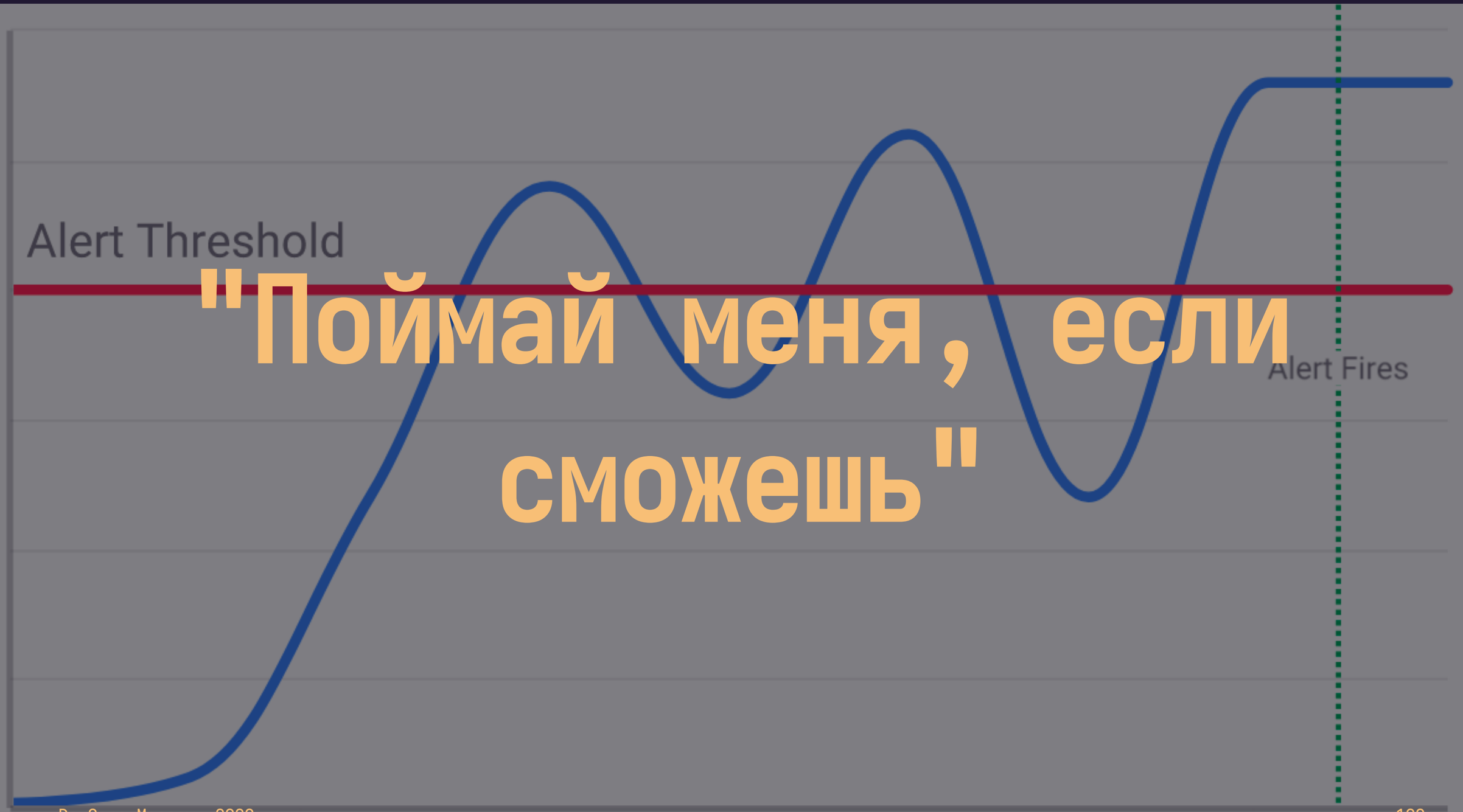
Но есть проблема

kubernetes
11004
kubernetes
4+
11004
@datamattsson
IDEAS INCLUDED
Pod
Service
StatefulSet
DaemonSet
Deployment

Метрики могут быть "хитрые"

Метрики могут быть "хитрые"

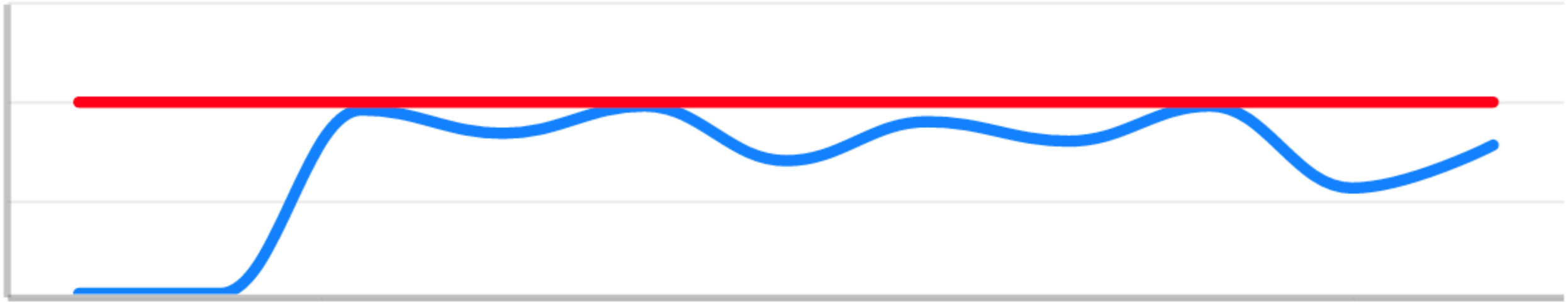
* double dipper
* lurker
* flappy bird



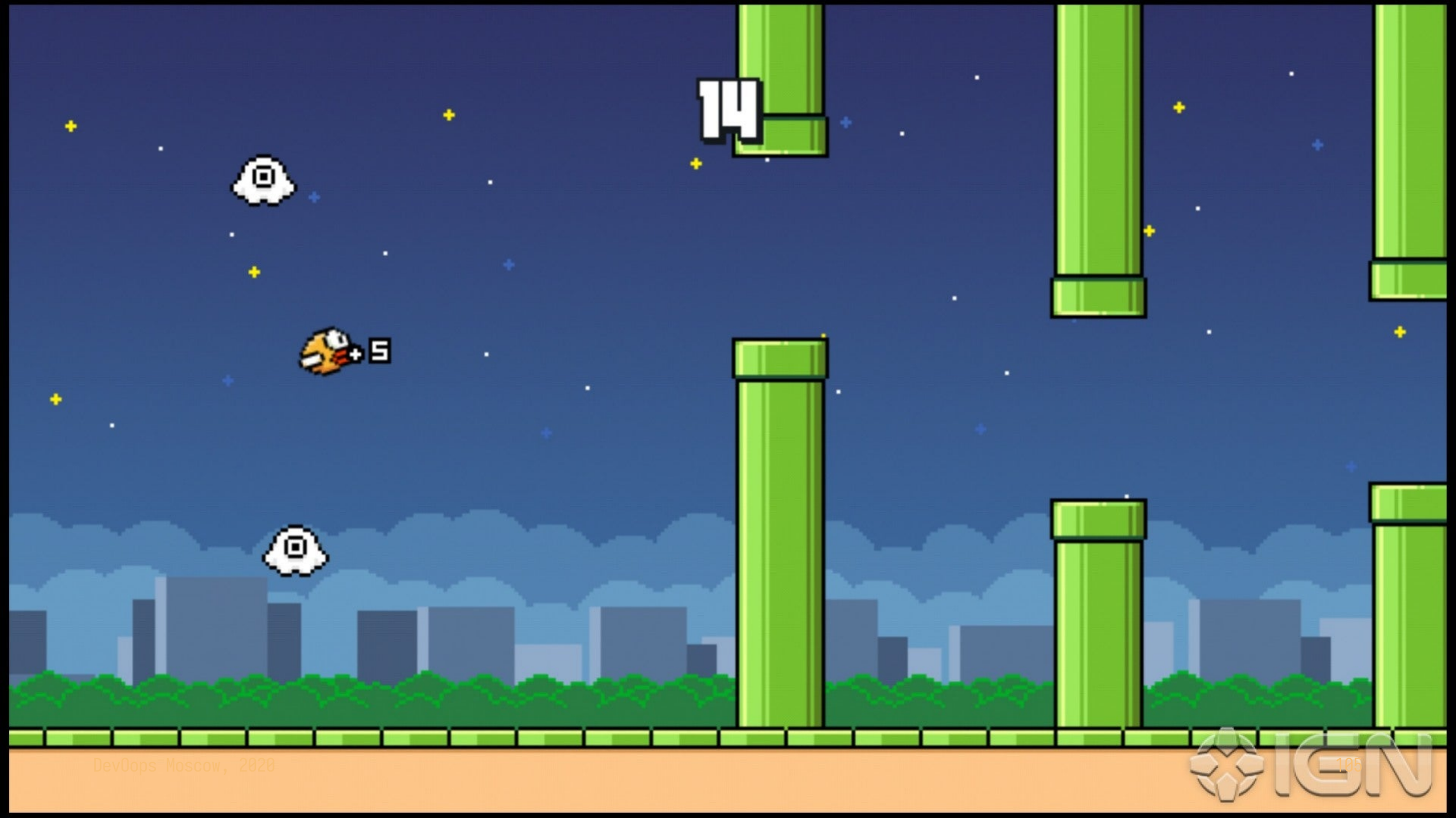
"Поймай меня, если сможешь"

Alert Threshold
Alert Fires

Недоперепил



Flappy bird


14
+5

Alert Fires
Alert Fires
Alert Fires

Multiwindow + Multi-burn-rate

https://landing.google.com/sre/workbook/chapters/alerting-on-slos/

Система мониторинга не должна мешать работе

SAME
BUT DIFFERENT

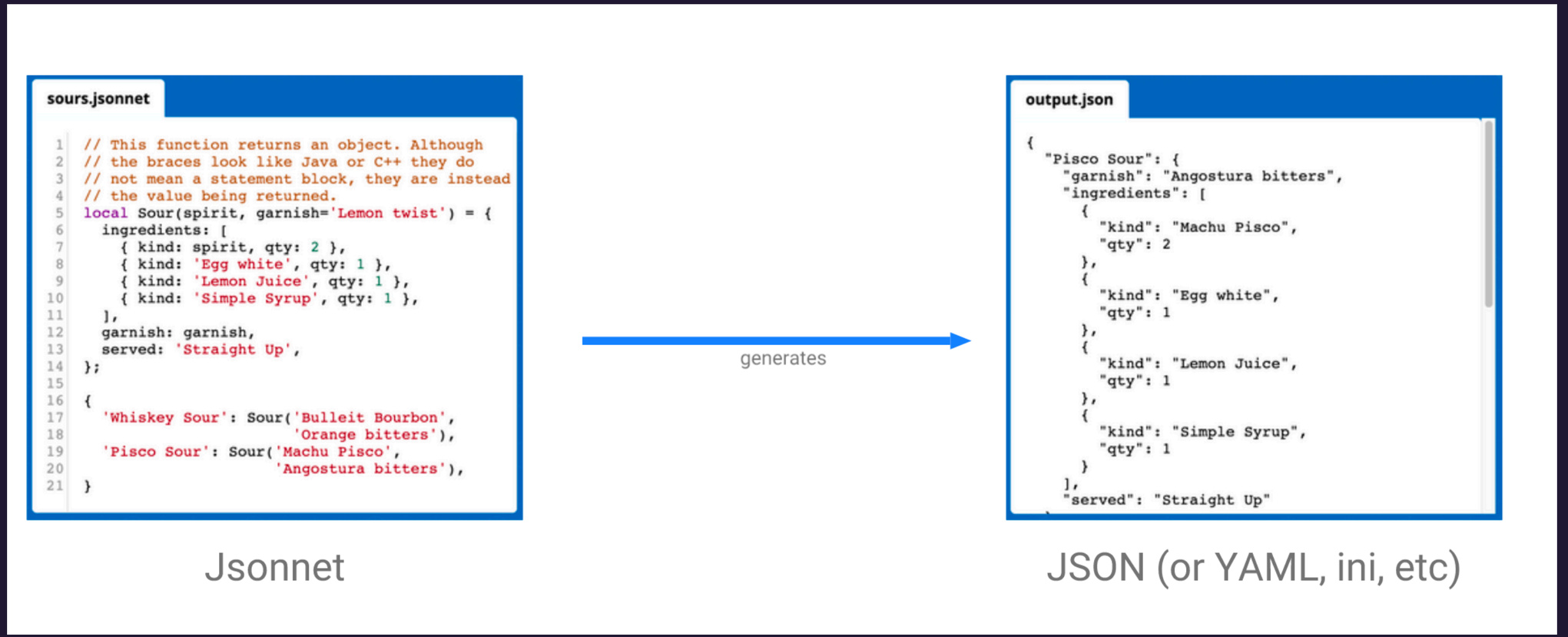
SSOT для сигналов и дашборд

Front-end!

Static Site Generator

Jsonnet

Jsonnet

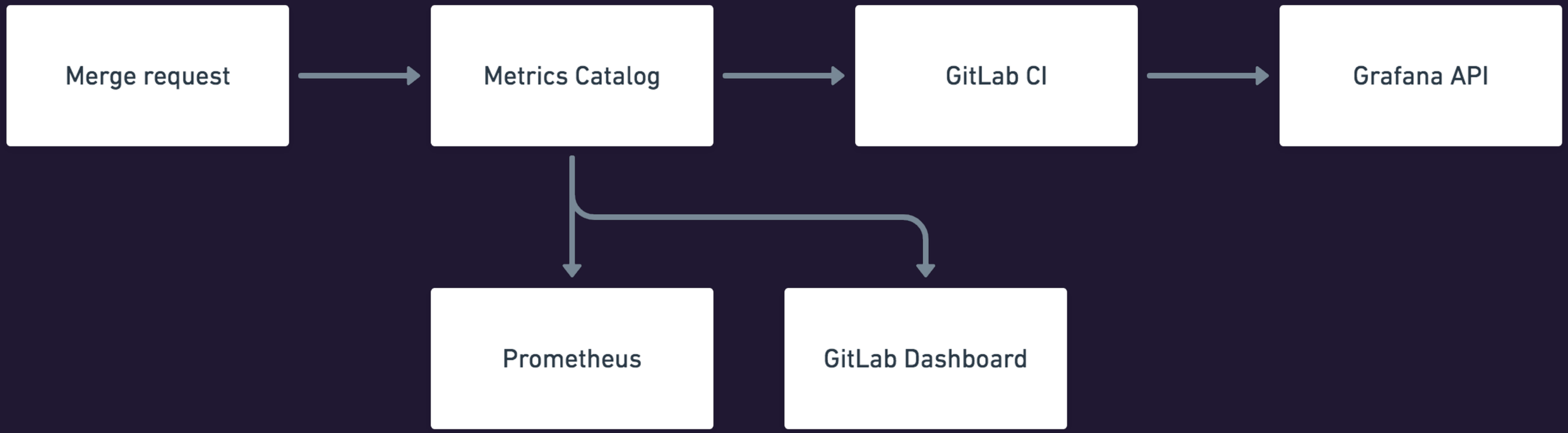

sours.jsonnet
// This function returns an object. Although
// the braces look like Java or C++ they do
// not mean a statement block, they are instead
// the value being returned.
local Sour(spirit, garnish='Lemon twist') = {
ingredients: [
{ kind: spirit, qty: 2 },
{ kind: 'Egg white', qty: 1 },
{ kind: 'Lemon Juice', qty: 1 },
{ kind: 'Simple Syrup', qty: 1 },
],
garnish: garnish,
served: 'Straight Up',
};
{
'Whiskey Sour': Sour('Bulleit Bourbon',
'Orange bitters'),
'Pisco Sour': Sour('Machu Pisco',
'Angostura bitters'),
}
generates
output.json
{
"Pisco Sour": {
"garnish": "Angostura bitters",
"ingredients": [
{
"kind": "Machu Pisco",
"qty": 2
},
{
"kind": "Egg white",
"qty": 1
},
{
"kind": "Lemon Juice",
"qty": 1
},
{
"kind": "Simple Syrup",
"qty": 1
}
],
"served": "Straight Up"
Jsonnet
JSON (or YAML, ini, etc)

Jsonnet

Функциональный язык шаблонов

JSON с комментариями, циклами, функциями.

Workflow


Merge request
Metrics Catalog
GitLab CI
Grafana API
Prometheus
GitLab Dashboard

Metric catalog

* ключевые метрики

Metric catalog

* ключевые метрики
* SLO для каждого сервиса

#define

* ключевые метрики
* SLO для каждого сервиса
* межсервисные зависимости

#generate

* конфигурация Prometheus

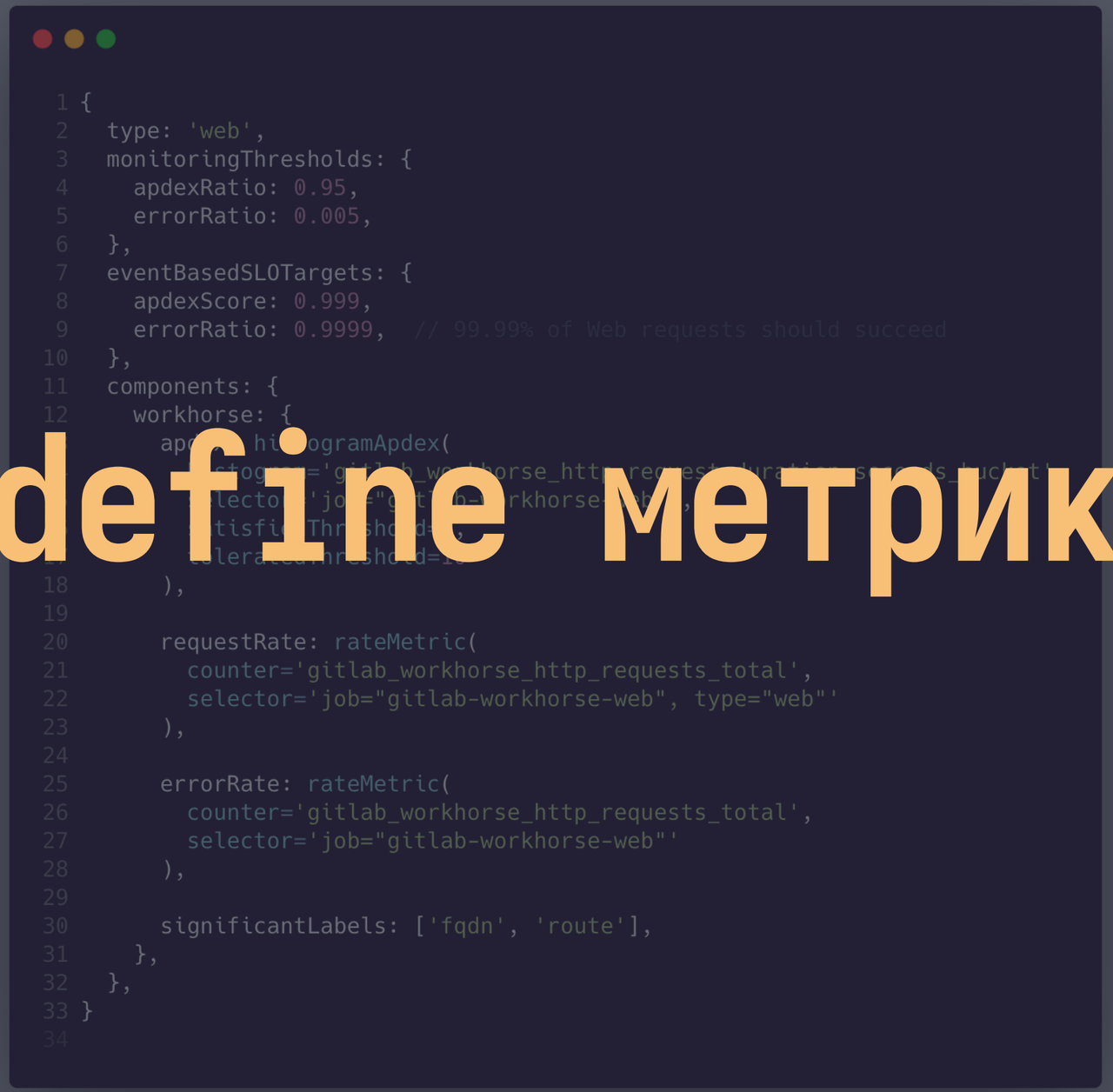
#generate

* конфигурация Prometheus
* Grafana

#generate

* конфигурация Prometheus
* Grafana
* GitLab Dashboard

#define метрики



```
{
  type: 'web',
  monitoringThresholds: {
    apdexRatio: 0.95,
    errorRatio: 0.005,
  },
  eventBasedSLOTargets: {
    apdexScore: 0.999,
    errorRatio: 0.9999,  // 99.99% of Web requests should succeed
  },
  components: {
    workhorse: {
      apdex: histogramApdex(
        histogram='gitlab_workhorse_http_request_duration_seconds_bucket',
        selector='job="gitlab-workhorse-web"',
        satisfiedThreshold=1,
        toleratedThreshold=10
      ),

      requestRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web", type="web"'
      ),

      errorRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web"'
      ),

      significantLabels: ['fqdn', 'route'],
    },
  },
}

```

```
{
  type: 'web',
  monitoringThresholds: {
    apdexRatio: 0.95,
    errorRatio: 0.005,
  },
  eventBasedSLOTargets: {
    apdexScore: 0.999,
    errorRatio: 0.9999,  // 99.99% of Web requests should succeed
  },
  components: {
    workhorse: {
      apdex: histogramApdex(
        histogram='gitlab_workhorse_http_request_duration_seconds_bucket',
        selector='job="gitlab-workhorse-web"',
        satisfiedThreshold=1,
        toleratedThreshold=10
      ),

      requestRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web", type="web"'
      ),

      errorRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web"'
      ),

      significantLabels: ['fqdn', 'route'],
    },
  },
}

```

```
{
  type: 'web',
  monitoringThresholds: {
    apdexRatio: 0.95,
    errorRatio: 0.005,
  },
  eventBasedSLOTargets: {
    apdexScore: 0.999,
    errorRatio: 0.9999,  // 99.99% of Web requests should succeed
  },
  components: {
    workhorse: {
      apdex: histogramApdex(
        histogram='gitlab_workhorse_http_request_duration_seconds_bucket',
        selector='job="gitlab-workhorse-web"',
        satisfiedThreshold=1,
        toleratedThreshold=10
      ),

      requestRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web", type="web"'
      ),

      errorRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web"'
      ),

      significantLabels: ['fqdn', 'route'],
    },
  },
}

```

```
{
  type: 'web',
  monitoringThresholds: {
    apdexRatio: 0.95,
    errorRatio: 0.005,
  },
  eventBasedSLOTargets: {
    apdexScore: 0.999,
    errorRatio: 0.9999,  // 99.99% of Web requests should succeed
  },
  components: {
    workhorse: {
      apdex: histogramApdex(
        histogram='gitlab_workhorse_http_request_duration_seconds_bucket',
        selector='job="gitlab-workhorse-web"',
        satisfiedThreshold=1,
        toleratedThreshold=10
      ),

      requestRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web", type="web"'
      ),

      errorRate: rateMetric(
        counter='gitlab_workhorse_http_requests_total',
        selector='job="gitlab-workhorse-web"'
      ),

      significantLabels: ['fqdn', 'route'],
    },
  },
}
```

```
- record: gitlab_component_ops:rate_5m
  labels:
    component: puma
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(http_request_duration_seconds_count{job="gitlab-rails",type="web"}[5m])
    )
- record: gitlab_component_ops:rate_30m
  labels:
    component: puma
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(http_request_duration_seconds_count{job="gitlab-rails",type="web"}[30m])
    )
- record: gitlab_component_ops:rate_1h
  labels:
    component: workhorse
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(gitlab_workhorse_http_requests_total{job="gitlab-workhorse-web", type="web"}[1h])
    )
```

#generate

```
- record: gitlab_component_ops:rate_5m
  labels:
    component: puma
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(http_request_duration_seconds_count{job="gitlab-rails",type="web"}[5m])
    )
- record: gitlab_component_ops:rate_30m
  labels:
    component: puma
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(http_request_duration_seconds_count{job="gitlab-rails",type="web"}[30m])
    )
- record: gitlab_component_ops:rate_1h
  labels:
    component: workhorse
    tier: sv
    type: web
  expr: |
    sum by (environment,stage) (
      rate(gitlab_workhorse_http_requests_total{job="gitlab-workhorse-web", type="web"}[1h])
    )
```

Мониторинг еще один инженер?

Мониторинг еще один инженер?

No

Но он стал ближе к людям

Но он стал ближе к людям

* Понятные ключевые метрики

Но он стал ближе к людям

* Понятные ключевые метрики
* Фокус на симптомах (меньше проблем у инженеров)

Но он стал ближе к людям

* Понятные ключевые метрики
* Фокус на симптомах (меньше проблем у инженеров)-
* Metrics Catalog as SSOT

Но он стал ближе к людям

* Понятные ключевые метрики
* Фокус на симптомах (меньше проблем у инженеров)-
* Metrics Catalog as SSOT
* Генерация dashboards

Спасибо!