



Observability и балансировка в Kafka: брокеры, топики, клиенты

Виктор Корейша,
руководитель направления *Managed Services* в Ozon



 viktor@koreysha.ru

 @koreysha

Виктор Корейша

Руководитель направления Managed Services



Kafka



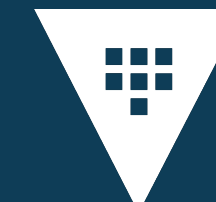
S3



Ceph



Redis



Vault



Подкасты



ПК конференций

- ➔ Ural Digital Weekend
- ➔ TeamLeadConf
- ➔ YaTalks



Kafka в Ozon

200

Брокеров в двух
больших кластерах



100 000

Партиций в примерно
10 000 топиков

200K

Активных
токенов



2024

17M RPS

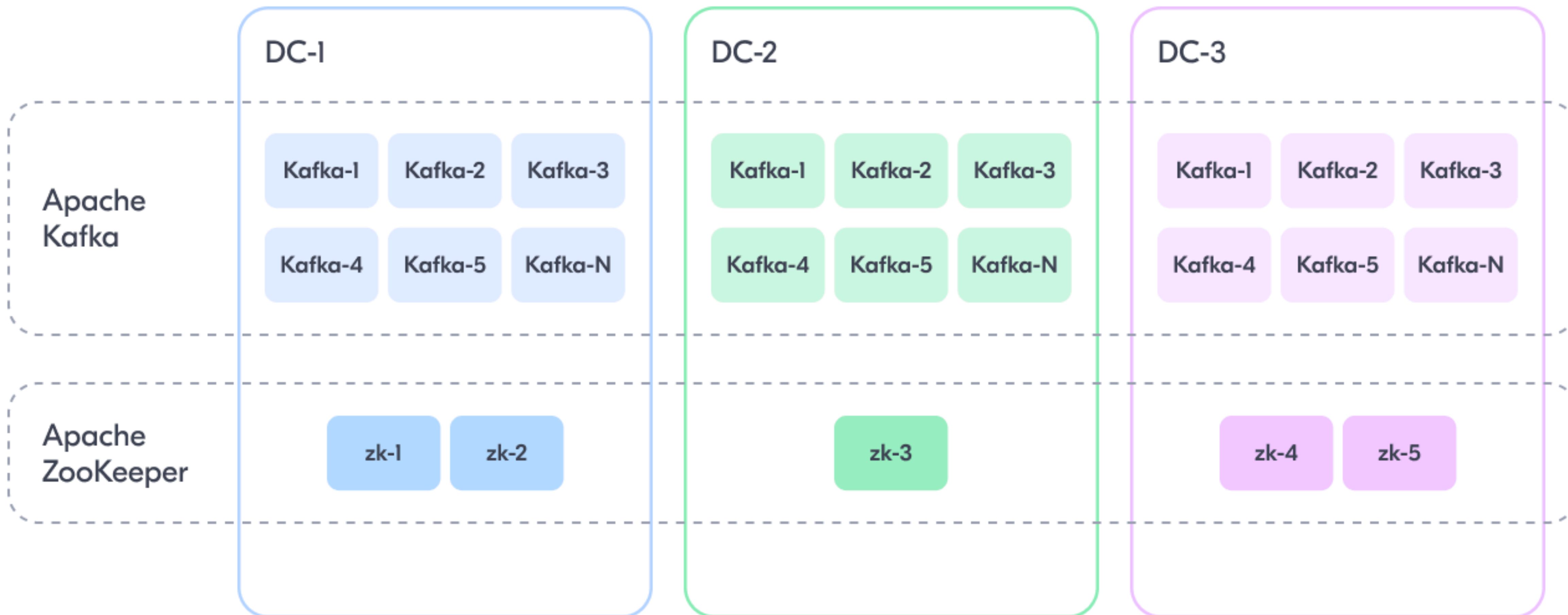
Пиковое значение в самом
большом кластере



1PiB

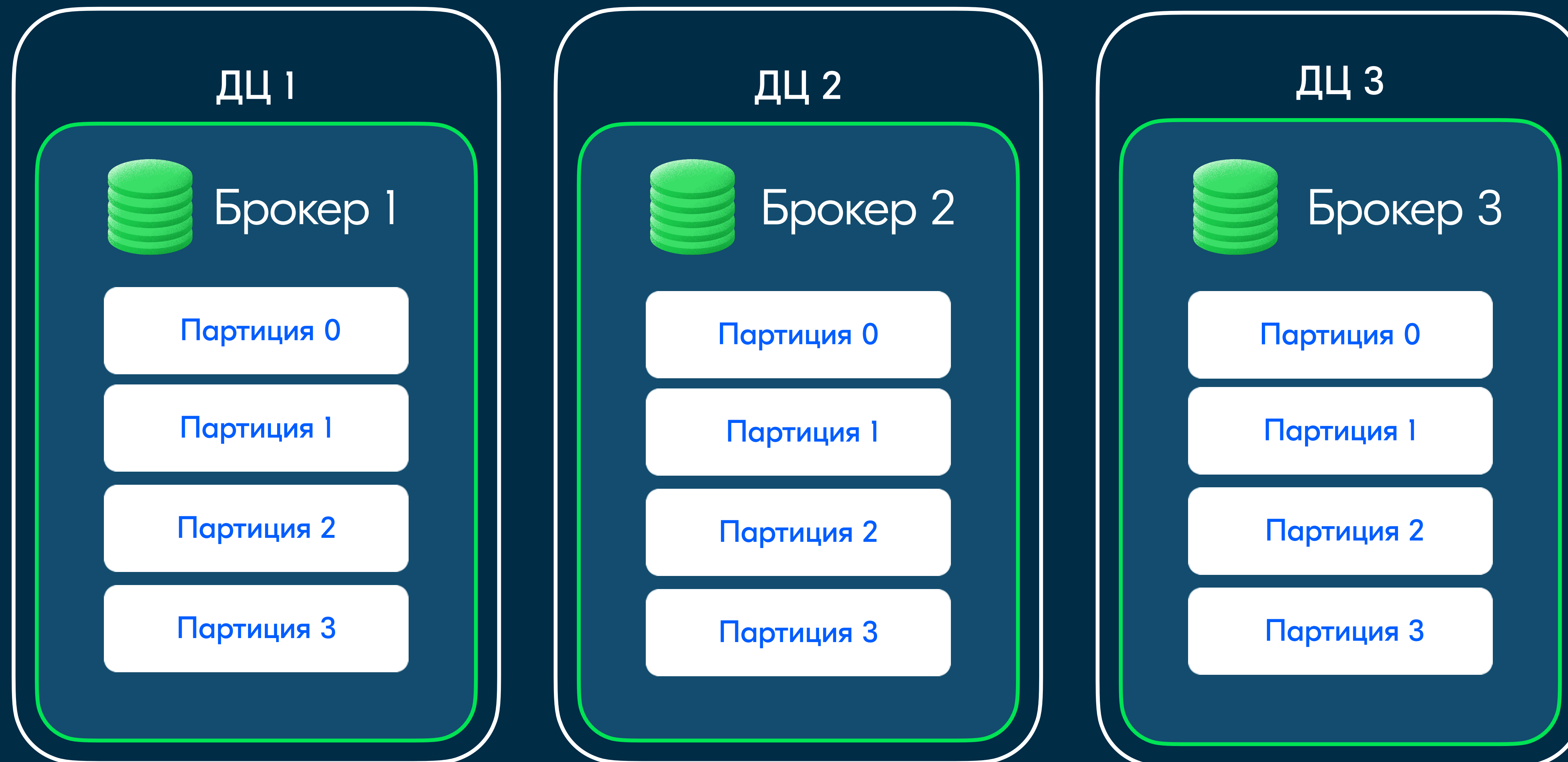
И сотни Gib трафика

Kafka в Ozon



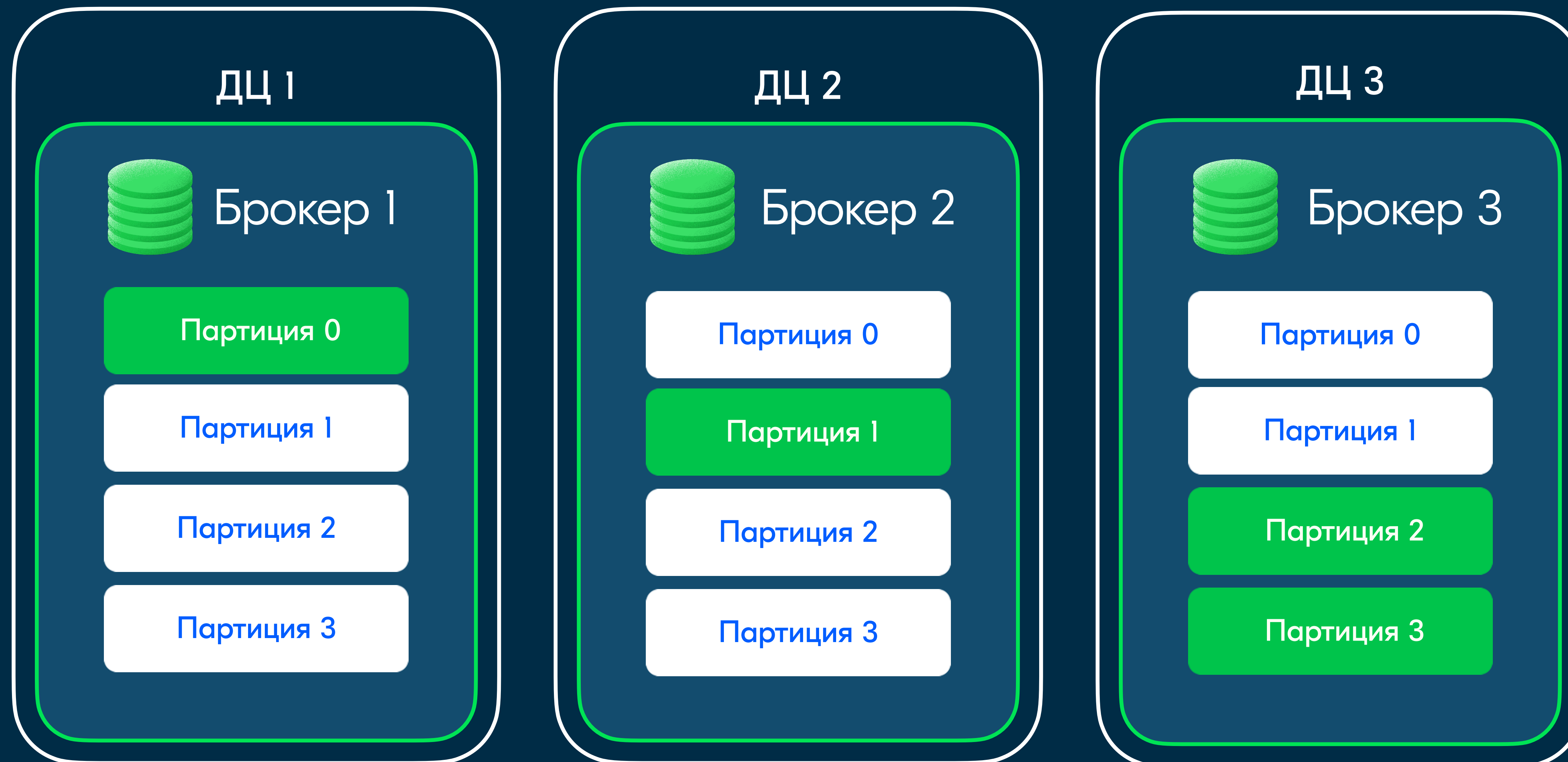
Кafka в Ozon

Типовая конфигурация топиков



Кafka в Ozon

Типовая конфигурация топиков



Два кластера на всех



**Два коммунальных
кластера:**

Два кластера на всех



Два коммунальных кластера:

- + Простая интеграция
- + Управление утилизацией
- + Делим по профилю нагрузки

Два кластера на всех



Два коммунальных кластера:

- + Простая интеграция
- + Управление утилизацией
- + Делим по профилю нагрузки

3K

сервисов на трех разных языках



Два кластера на всех



Два коммунальных кластера:

- + Простая интеграция
- + Управление утилизацией
- + Делим по профилю нагрузки

3K

сервисов на трех разных языках



20K

топиков, с разными форматами данных, объемами информации и пропускной способностью

Два кластера на всех



Два коммунальных кластера:

- + Простая интеграция
- + Управление утилизацией
- + Делим по профилю нагрузки

3К

сервисов на трех разных языках



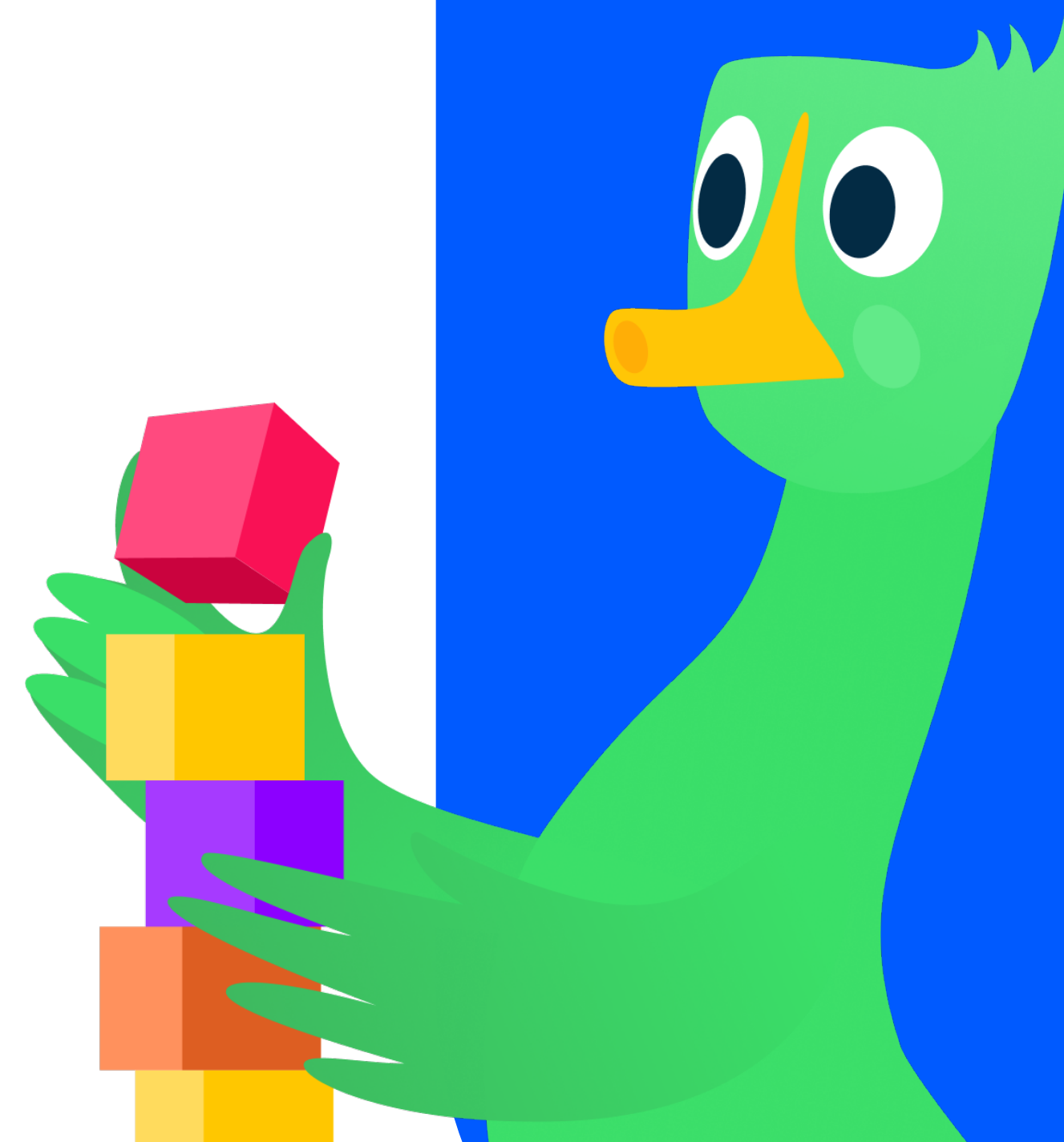
20К

топиков, с разными форматами данных, объемами информации и пропускной способностью

А ещё

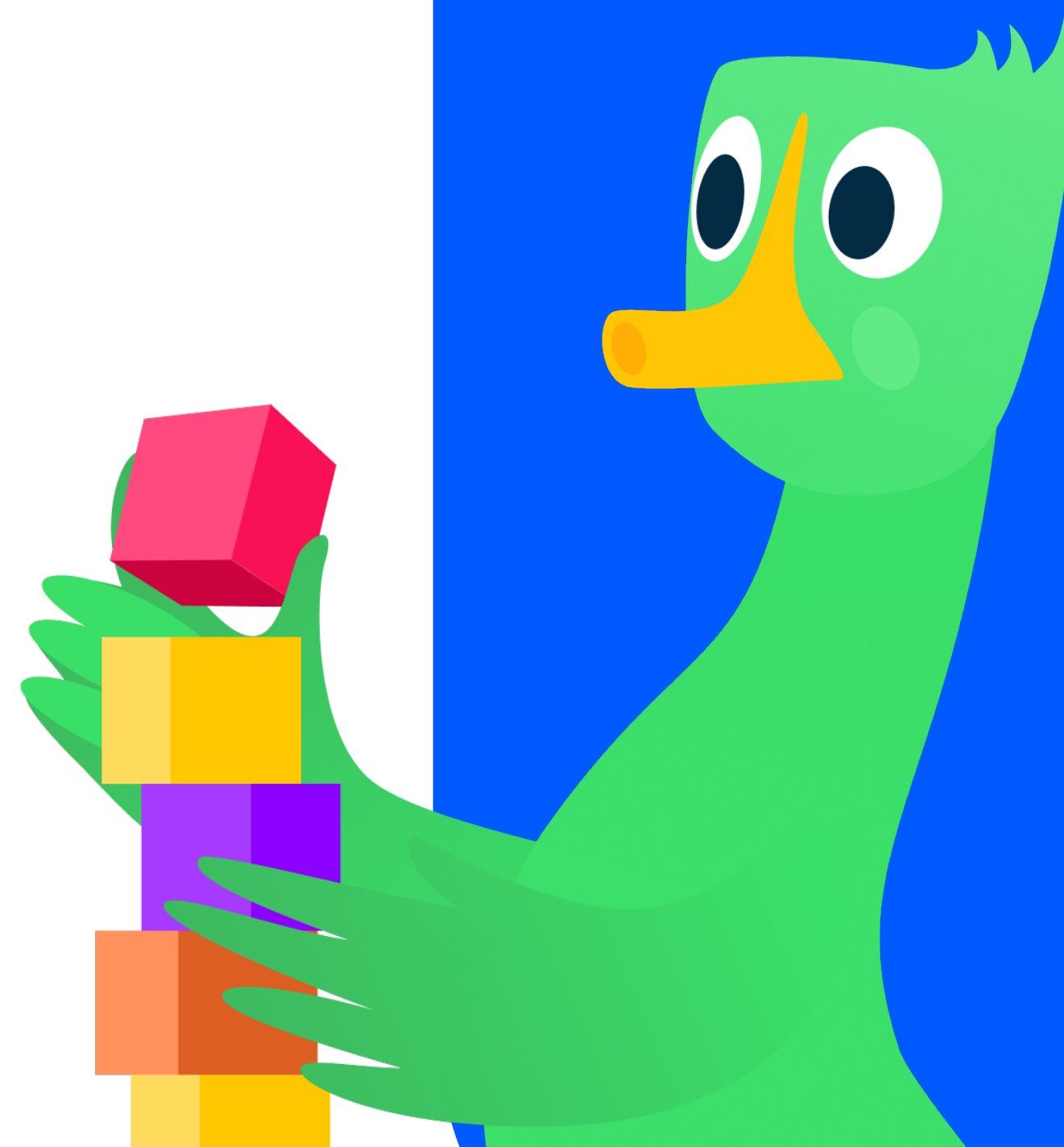
- DAG-и AF
- «Коробочные» решения
- И собственная инфраструктура

Роли Kafka в Ozon



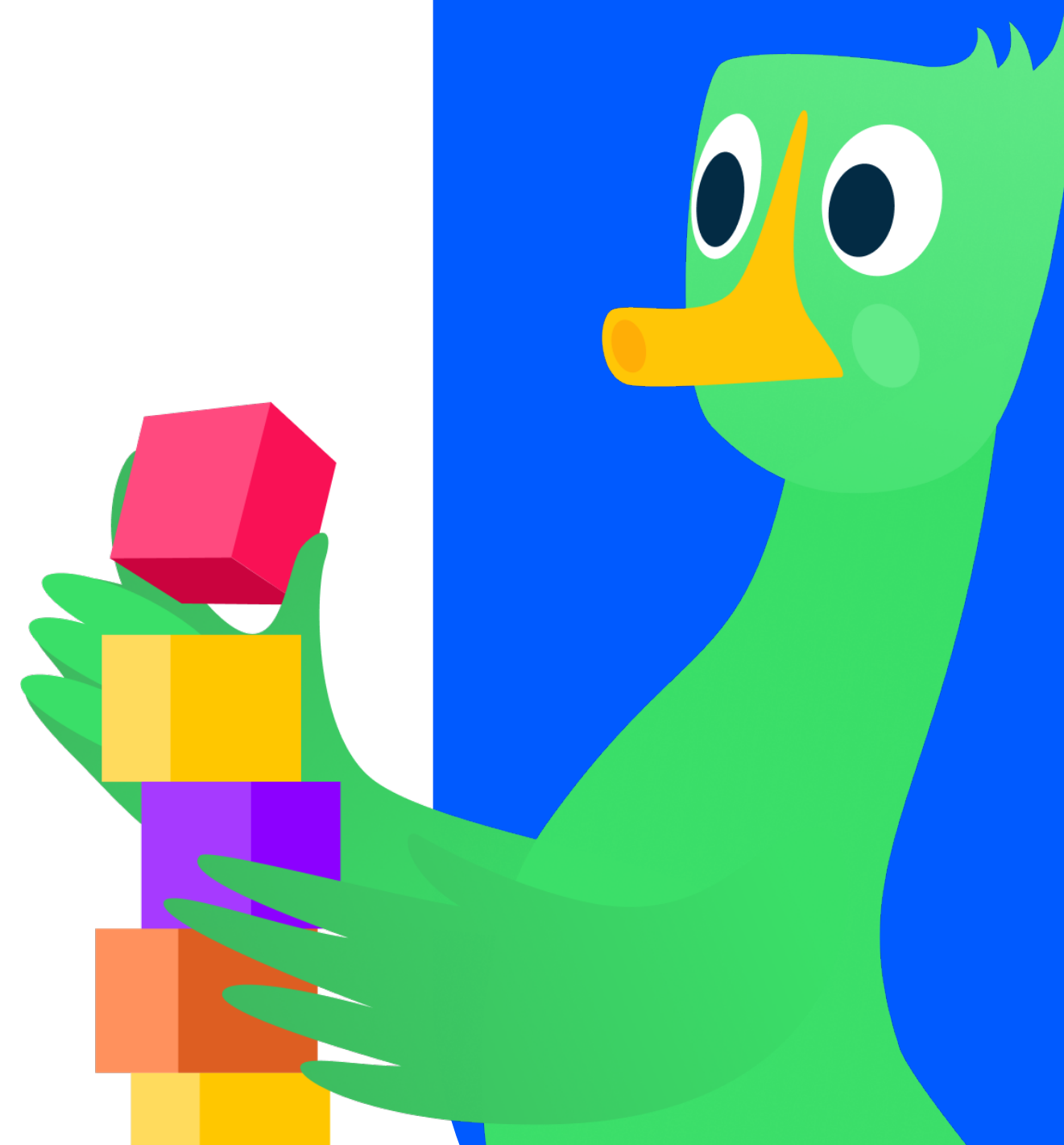
Роли Kafka в Ozon

- Шина данных: Kafka — основной инструмент асинхронного взаимодействия



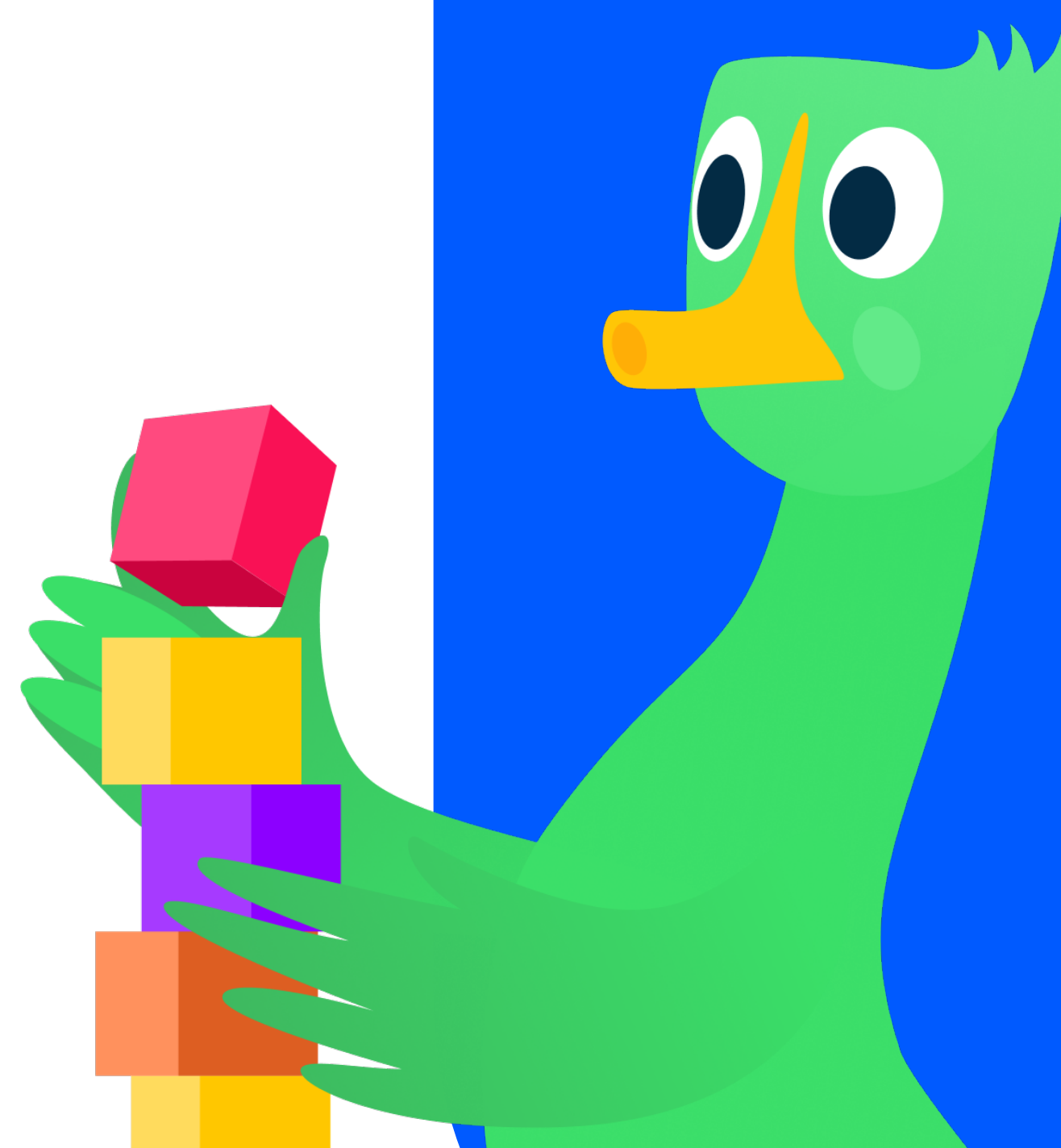
Роли Kafka в Ozon

- Шина данных: Kafka — основной инструмент асинхронного взаимодействия
- Буфер в системах обработки данных



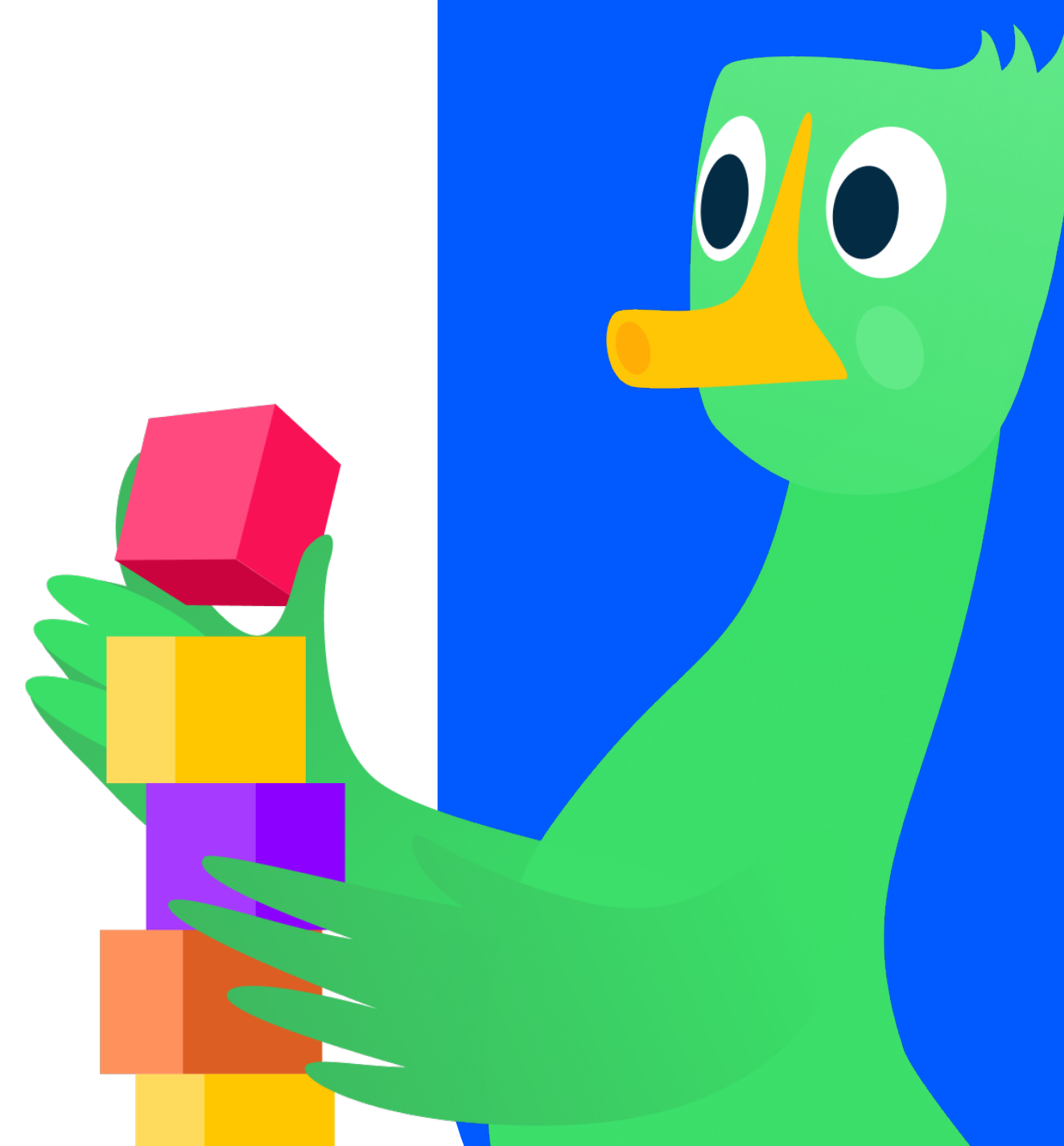
Роли Kafka в Ozon

- Шина данных: Kafka — основной инструмент асинхронного взаимодействия
- Буфер в системах обработки данных
- Средство мультикаста: доставка данных большому числу потребителей



Роли Kafka в Ozon

- Шина данных: Kafka — основной инструмент асинхронного взаимодействия
- Буфер в системах обработки данных
- Средство мультикаста: доставка данных большому числу потребителей



Мы не используем Kafka для долговременного хранения данных и как базу данных. Почти не используем для потоков обработки.

Сегодня мы поговорим о....

Сегодня мы поговорим о....



Kafka

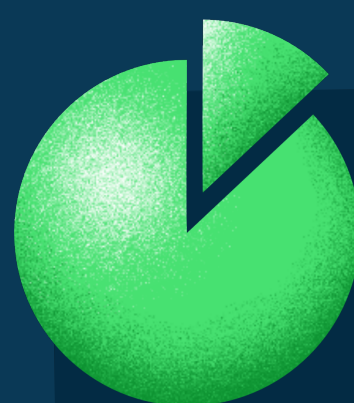
Ресурсах Kafka-кластера

Сегодня мы поговорим о....



Kafka

Ресурсах Kafka-кластера



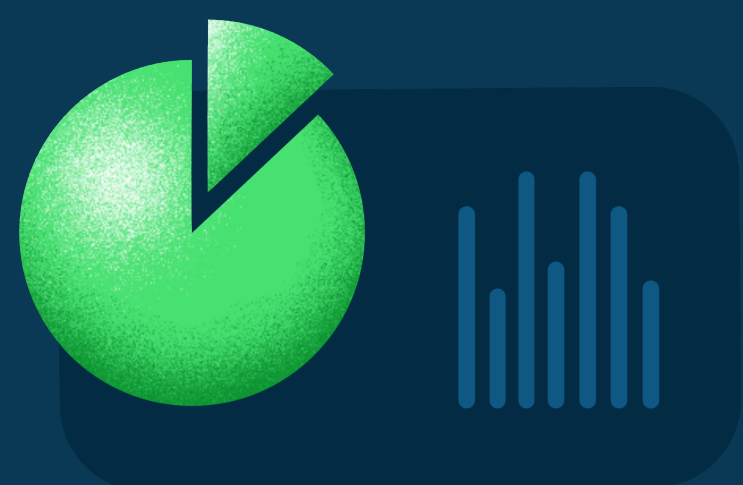
Мониторинге кластера и брокеров

Сегодня мы поговорим о....

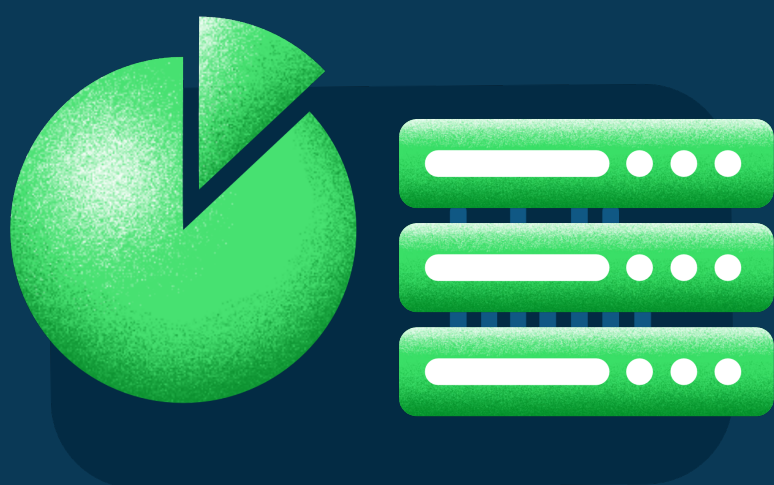


Kafka

Ресурсах
Kafka-кластера



Мониторинге
кластера и брокеров



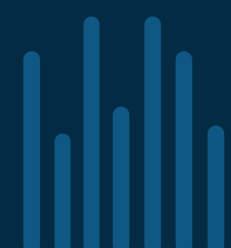
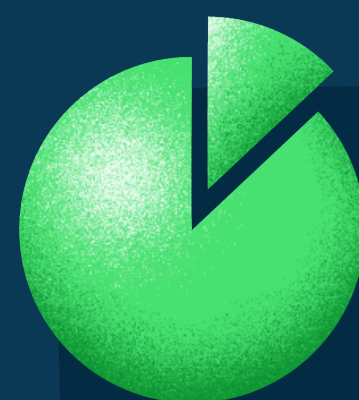
Топиках
и потребителях

Сегодня мы поговорим о....

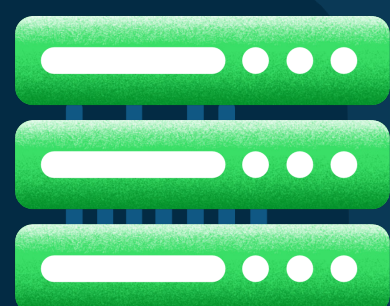
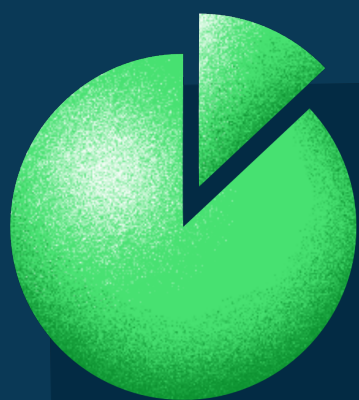


Kafka

Ресурсах
Kafka-кластера



Мониторинге
кластера и брокеров



Топиках
и потребителях

О том, как работает Kafka,
термины и типовые ошибки



Ресурсы

Управление кластером

Управление ресурсами

Что является ресурсом для кластера?

CPU

Сеть

Диск

RAM

Управление ресурсами

Что является ресурсом для кластера?

- CPU
- Диск (место)
- Сеть
- Количество партиций
- Диск (запись/чтение)

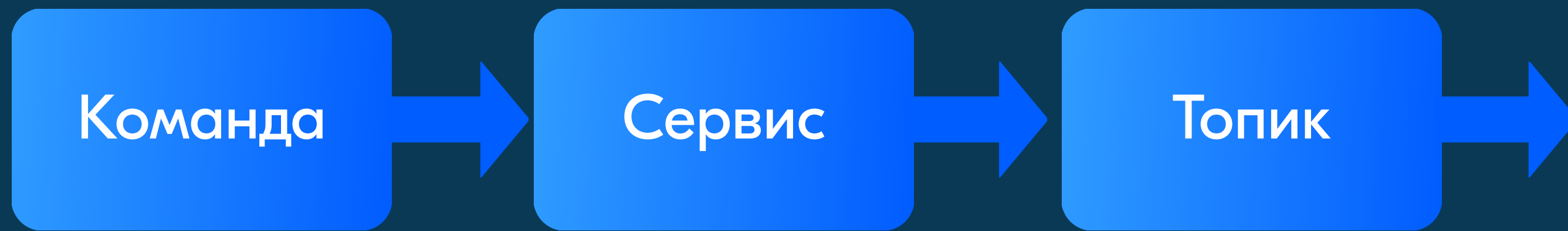
Управление ресурсами

Что является ресурсом для кластера?

- **CPU**
 - Может увеличиваться внезапно
 - Не масштабируется вертикально (в рамках брокера)
- **Сеть**
 - Может увеличиваться внезапно
 - Не масштабируется вертикально (в рамках брокера)
- **Диск (запись/чтение)**
 - Потребление зависит от нагрузки
 - Не масштабируется вертикально
 - Асинхронное использование
- **Диск (место)**
 - Предсказуемая утилизация
 - Масштабируется вертикально в рамках брокера
- **Количество партиций**
 - Предсказуемая утилизация
 - Масштабируется в рамках брокера

Управление ресурсами

Партиции и место на диске



Детерминированные ресурсы —
распределяем

✕

Редактировать ресурс Kafka topic

План для ресурса

☐ default.small

Партиции: 1

Ретеншн: 100 MiB

Реплики: 3

300 MiB

☐ default.medium

Партиции: 2

Ретеншн: 1 GiB

Реплики: 3

6 GiB

☐ default.large

Партиции: 5

Ретеншн: 2 GiB

Реплики: 3

30 GiB

☒ custom Текущий 921.6 MiB

Партиции 3 ⓘ + -

Ретеншн 102 ⓘ MiB ▾ + -

Реплики 3 ⓘ

▼ Дополнительные параметры

Compression type producer ⓘ ▾

Max message bytes 10485760 ⓘ + -

Min insync replicas 2 ⓘ

Segment ms 604800000 ⓘ + -

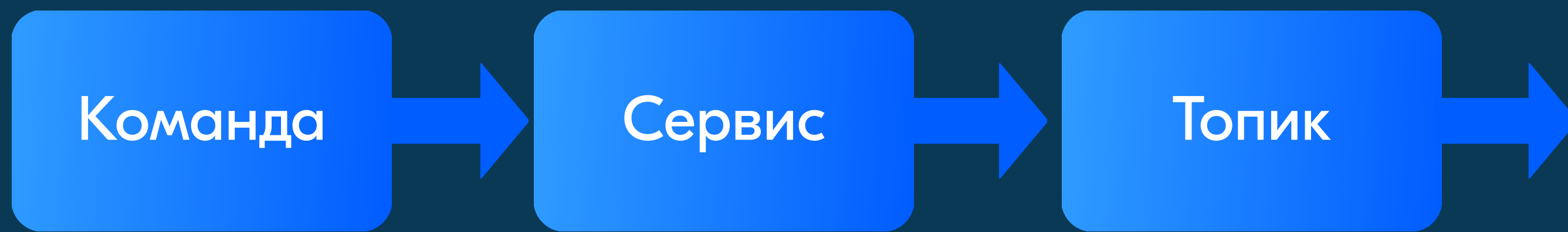
Cleanup Policy delete ⓘ ▾

☐ Unclean leader election enable ⓘ

Retention ms 86400000 ⓘ + -

Управление ресурсами

Партиции и место на диске



Детерминированные ресурсы —
распределяем

Недетерминированные ресурсы —
контролируем

Редактировать ресурс Kafka topic

План для ресурса

☐ default.small

Партиции: 1
Ретеншн: 100 MiB
Реплики: 3

300 MiB

☐ default.medium

Партиции: 2
Ретеншн: 1 GiB
Реплики: 3

6 GiB

☐ default.large

Партиции: 5
Ретеншн: 2 GiB
Реплики: 3

30 GiB

☒ custom
Текущий

921.6 MiB

Партиции 3 *i* + -

Ретеншн 102 *i* MiB *i* + -

Реплики 3 *i*

Дополнительные параметры

Compression type producer *i* v

Min insync replicas 2 *i*

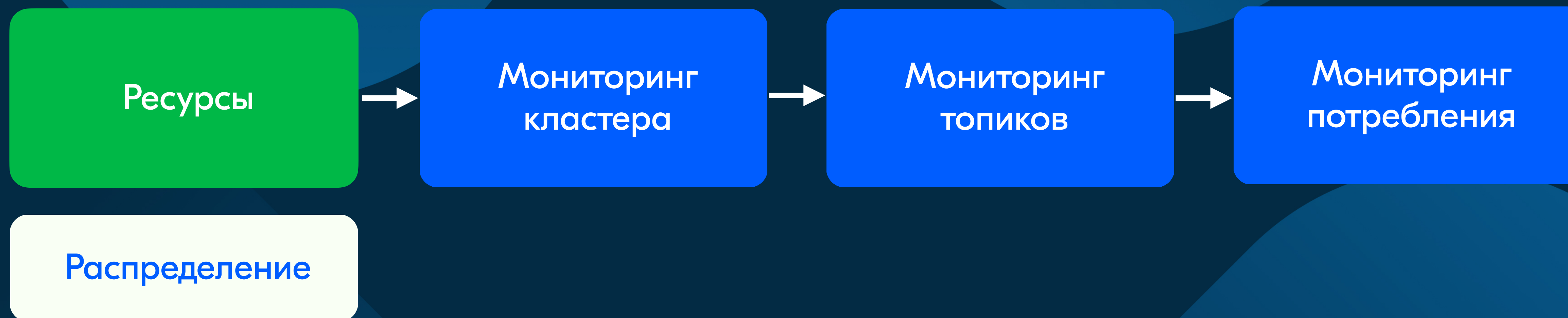
Cleanup Policy delete *i* v

Retention ms 86400000 *i* + -

Max message bytes 10485760 *i* + -

Segment ms 604800000 *i* + -

Unclean leader election enable *i*



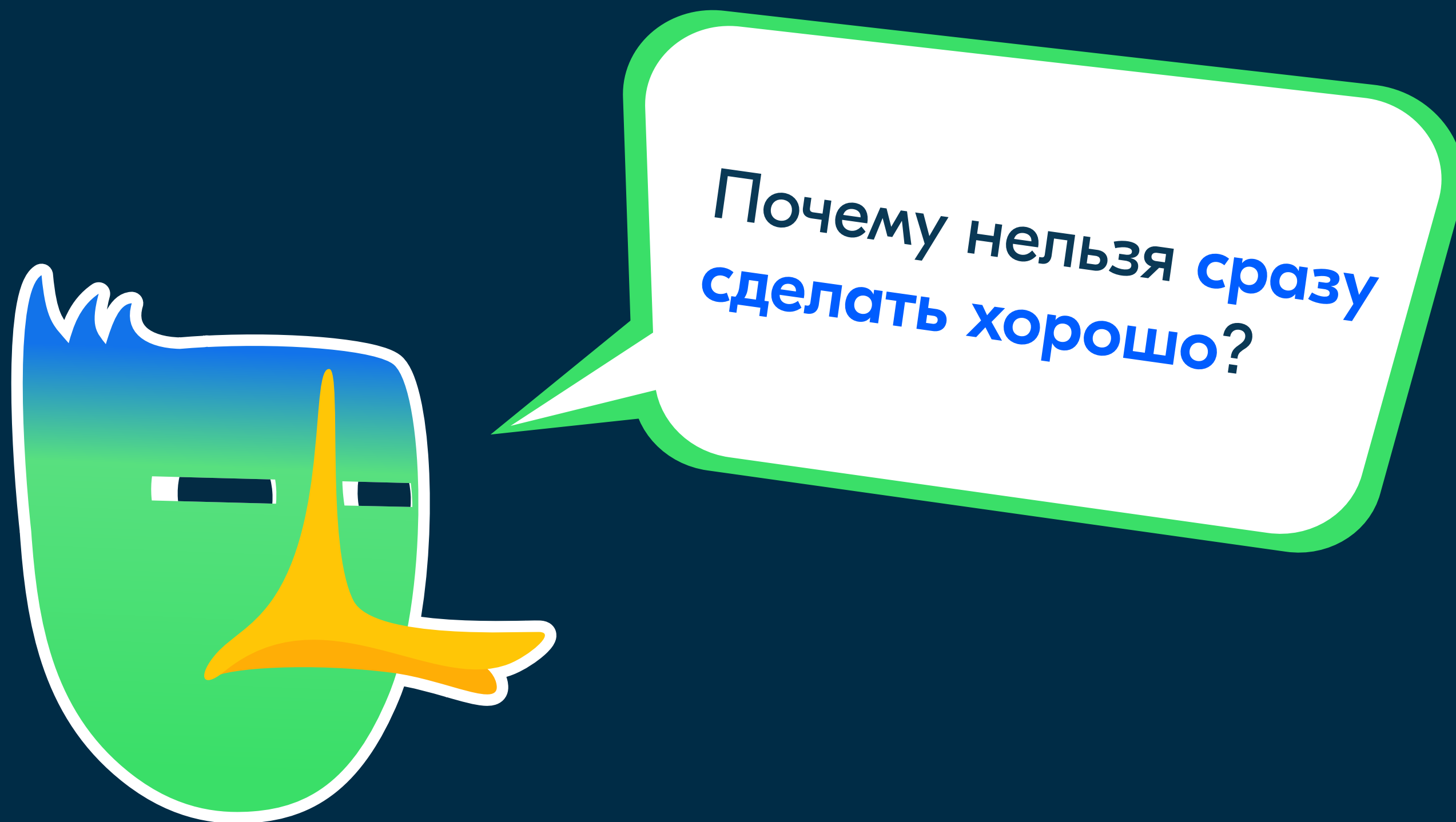
Kafka Broker Load

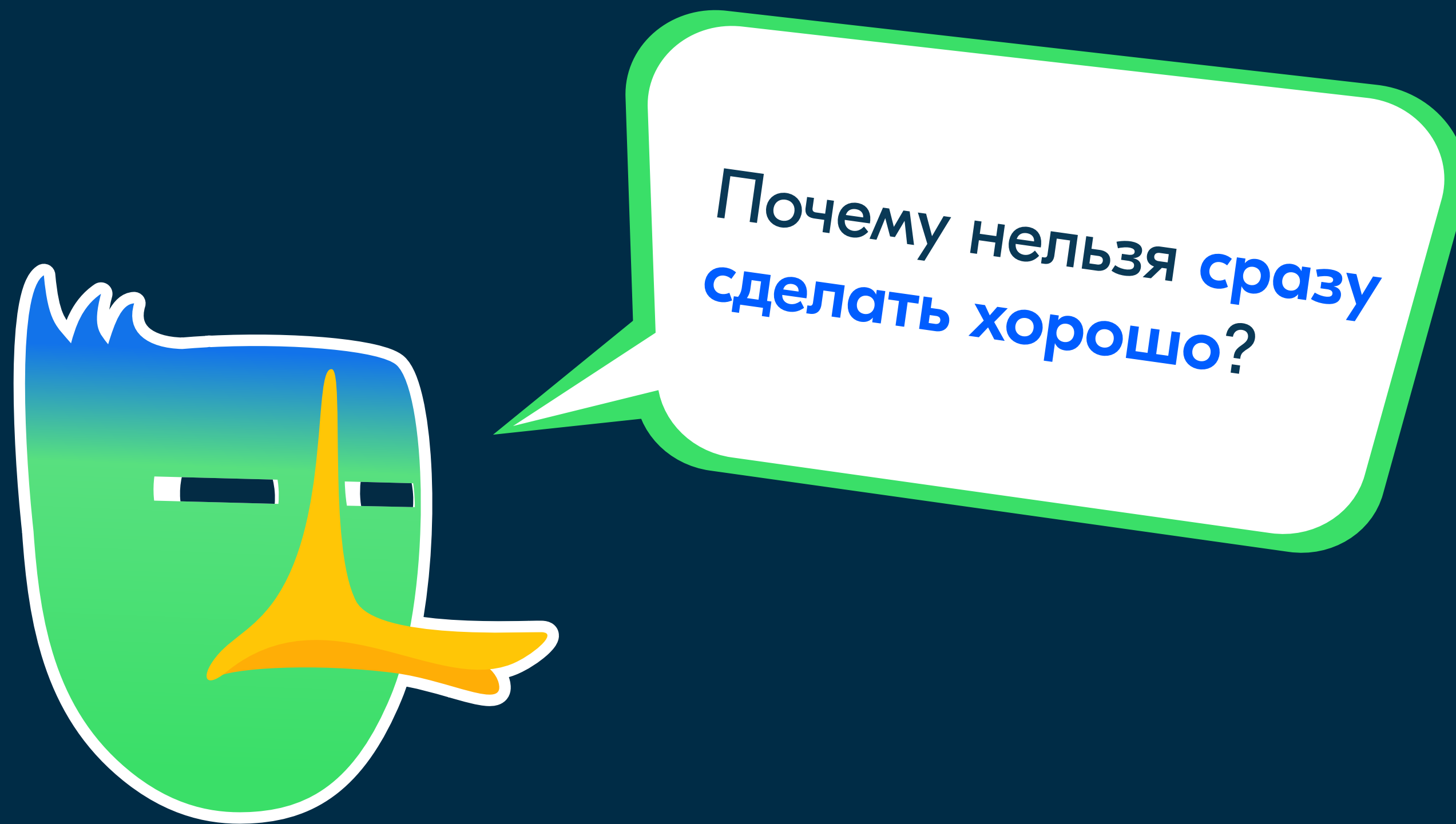
Broker			Topic/Partition		Disk/Cpu	
ID	State	Host	#Replicas	#Leaders	Disk Used	CPU Used
1028	ALIVE		3516	2113	123.25 GB	58.56 %
1019	ALIVE		7696	7579	60.74 GB	44.35 %
1023	ALIVE		7770	309	199.68 GB	35.82 %
1024	ALIVE		9205	4052	278.78 GB	67.09 %
1018	ALIVE		9294	1730	413.85 GB	22.30 %
1027	ALIVE		13346	1229	351.35 GB	48.11 %

Состояние кластера

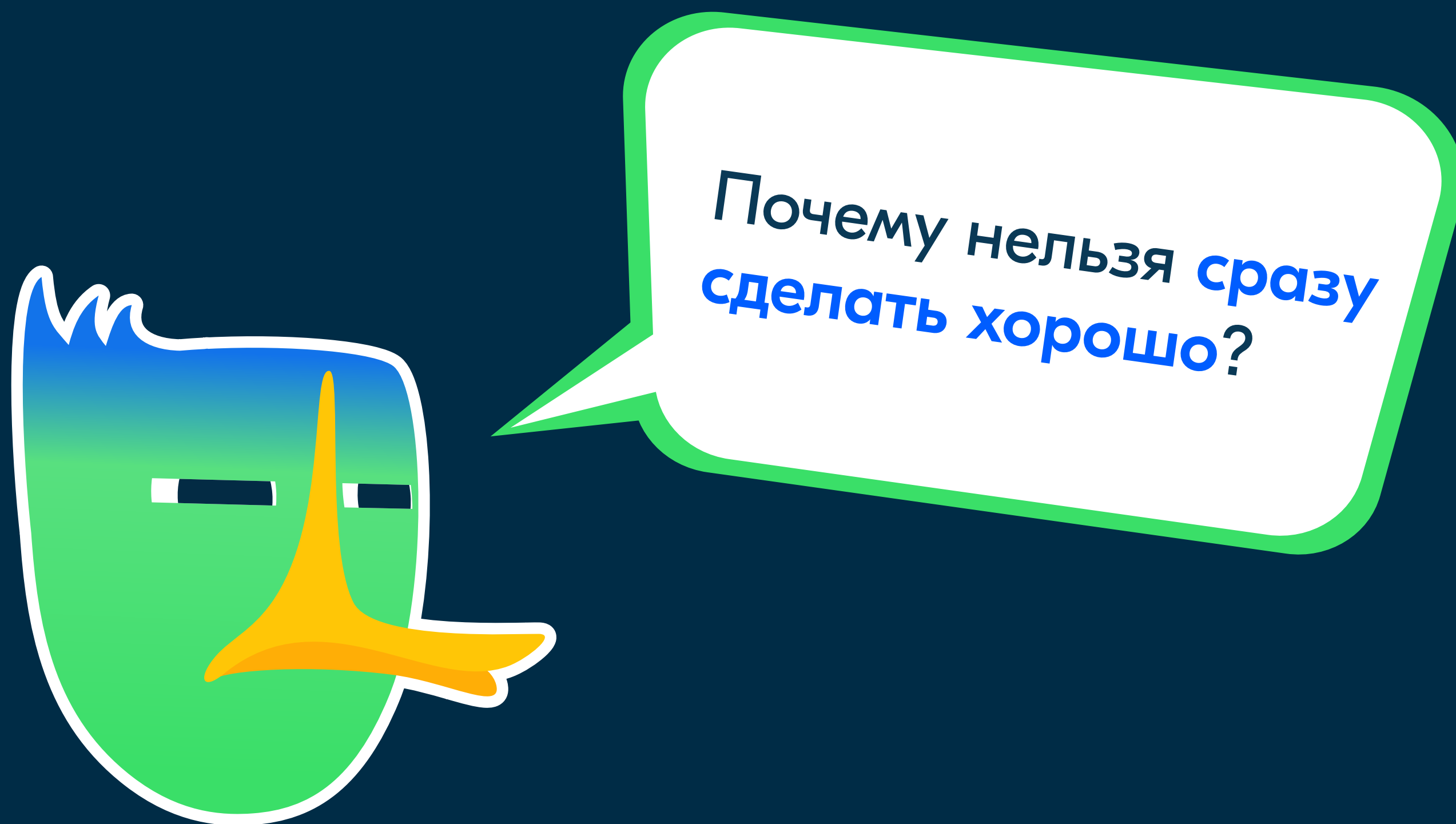
Kafka Broker Load

Broker			Topic/Partition		Disk/Cpu	
ID	State	Host	#Replicas	#Leaders	Disk Used	CPU Used
1028	ALIVE		3516	2113	123.25 GB	58.56 %
1019	ALIVE		7696	7579	60.74 GB	44.35 %
1023	ALIVE		7770	309	199.68 GB	35.82 %
1024	ALIVE		9205	4052	278.78 GB	67.09 %
1018	ALIVE		9294	1730	413.85 GB	22.30 %
1027	ALIVE		13346	1229	351.35 GB	48.11 %

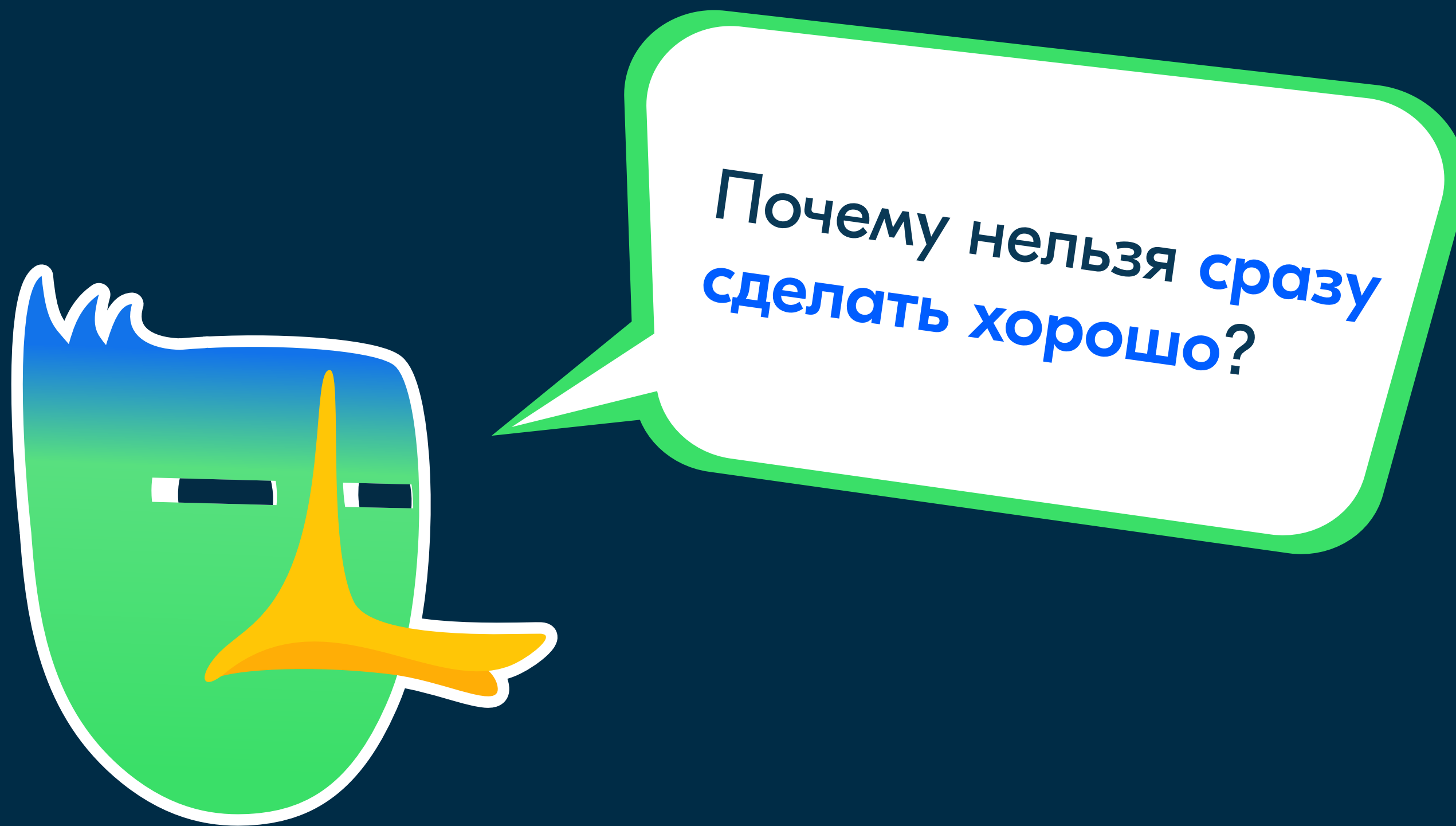




- Брокеры выходят из строя



- Брокеры выходят из строя
- Новые вводятся в эксплуатацию



- Брокеры выходят из строя
- Новые вводятся в эксплуатацию
- Нагрузка на топики меняется со временем

Балансировка с помощью Kafka-tools

Почему нет?

Балансировка с помощью Kafka-tools

Почему нет?

1. Нет возможности учитывать распределения по ДЦ



Балансировка с помощью Kafka-tools

Почему нет?

1. Нет возможности учитывать распределения по ДЦ
2. Балансировка осуществляется «на глаз» — результат получается далеким от идеального



Балансировка с помощью Kafka-tools

Почему нет?

1. Нет возможности учитывать распределения по ДЦ
2. Балансировка осуществляется «на глаз» — результат получается далеким от идеального
3. Если запустить полную перебалансировку — брокеры и сеть не справятся. Приходится дробить и выполнять по несколько партиций за раз

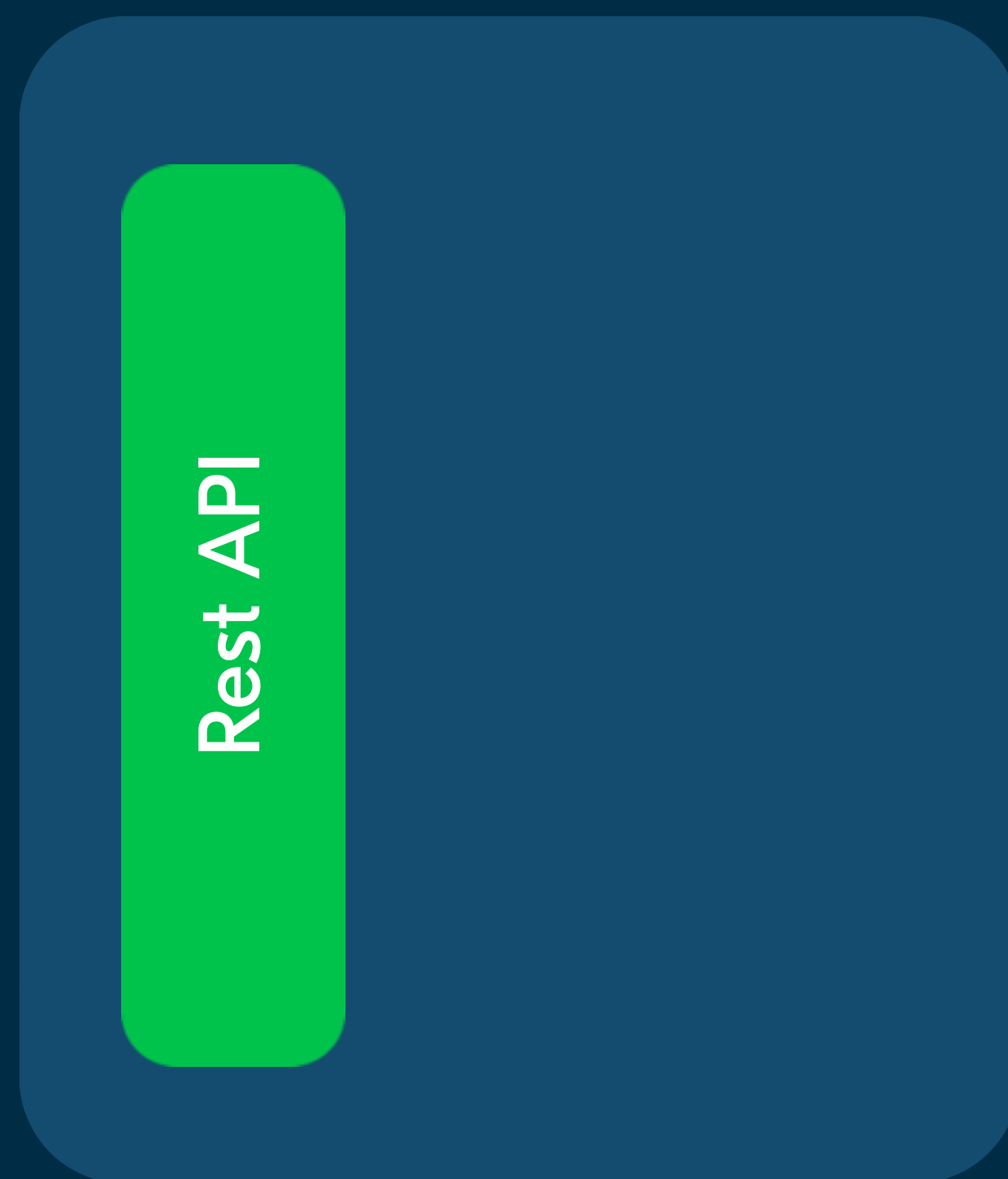


Что такое Cruise Control?

Как работает «на пальцах»?

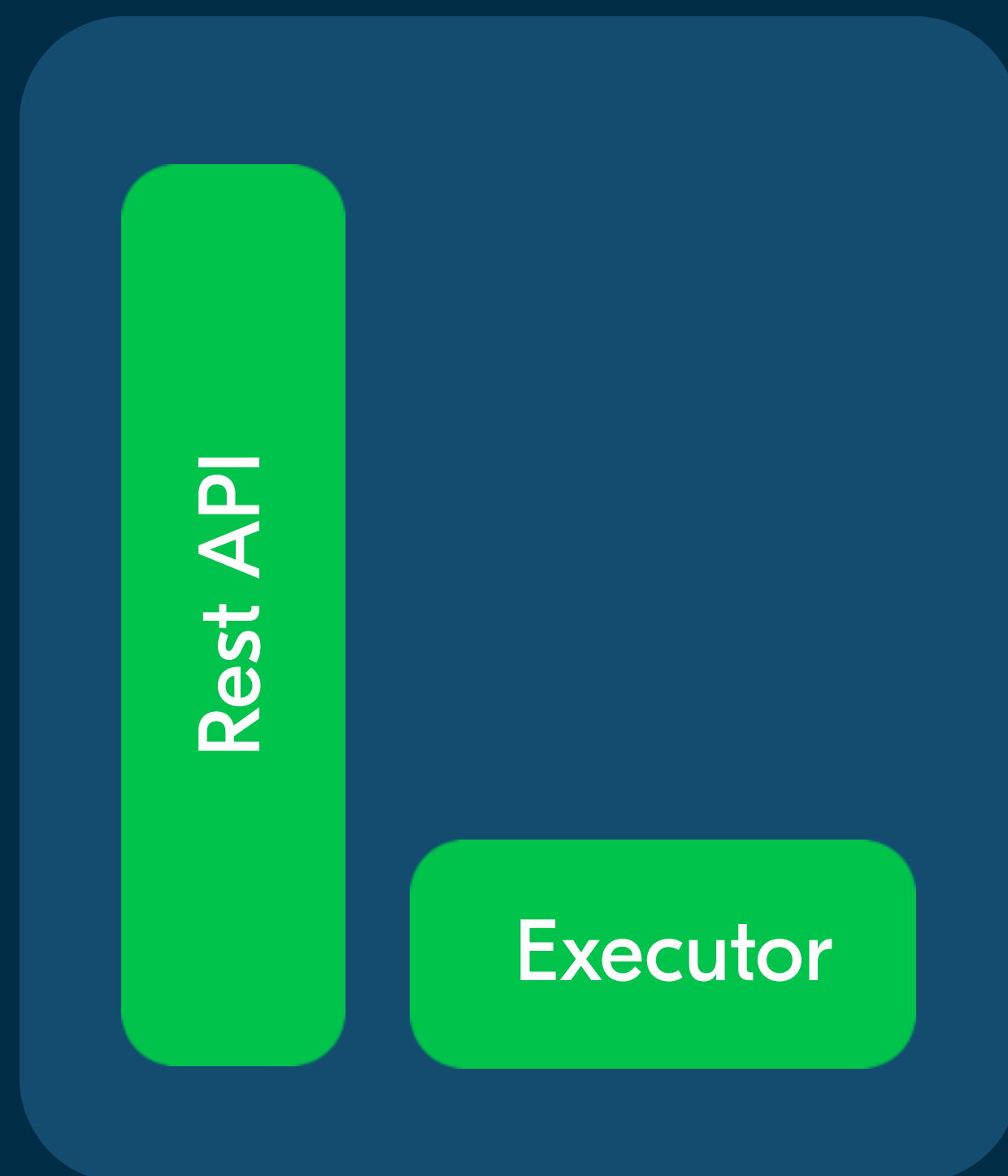
Как устроен Cruise Control

Cruise Control



Как устроен Cruise Control

Cruise Control

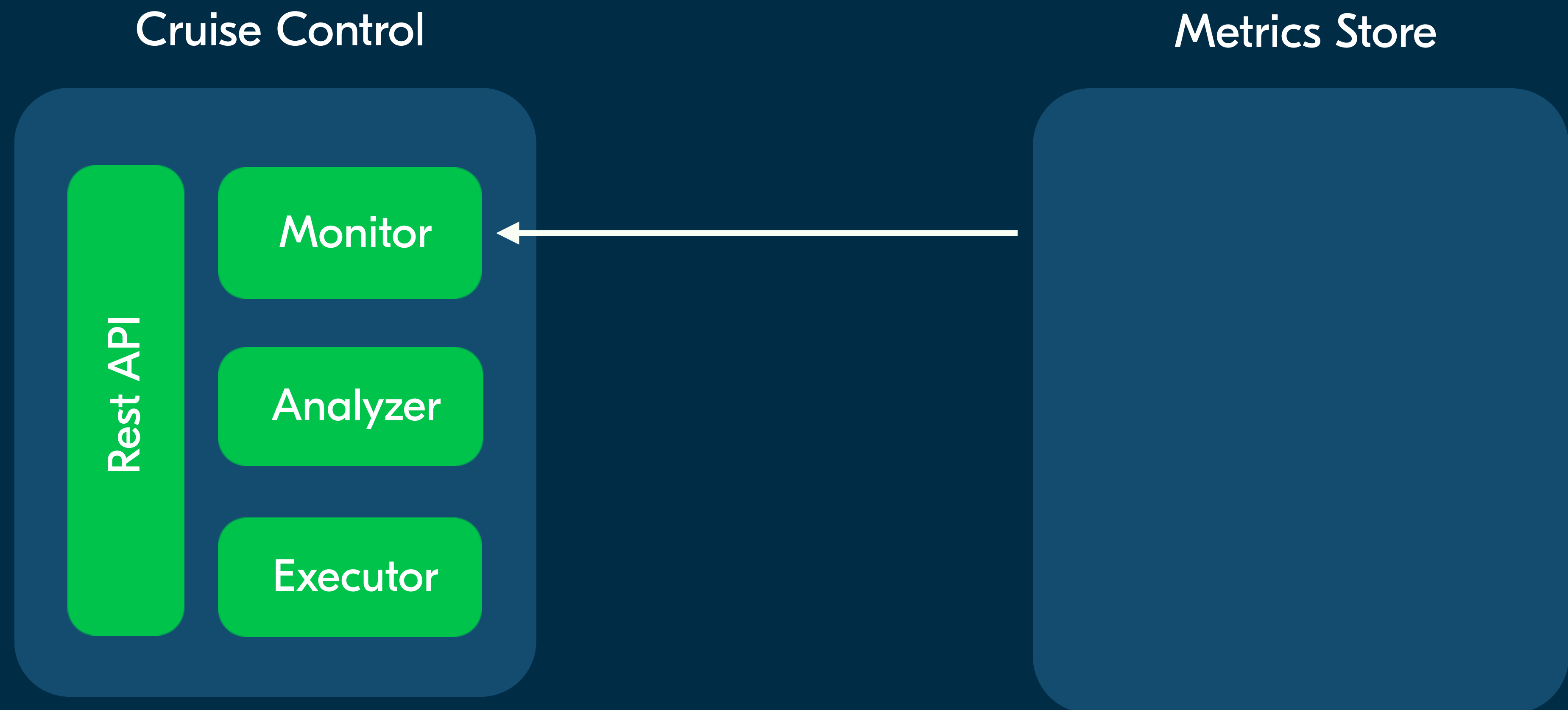


Как устроен Cruise Control

Cruise Control

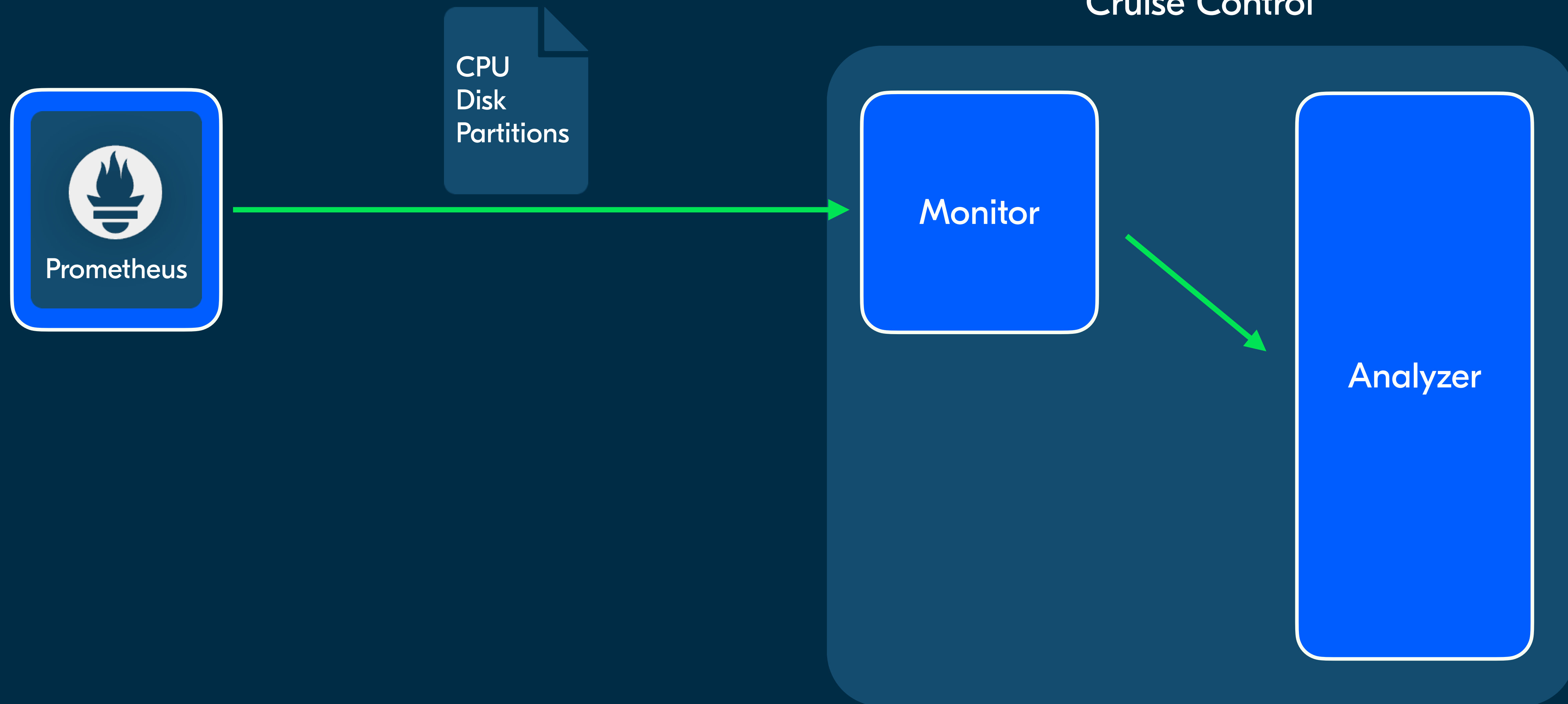


Как устроен Cruise Control



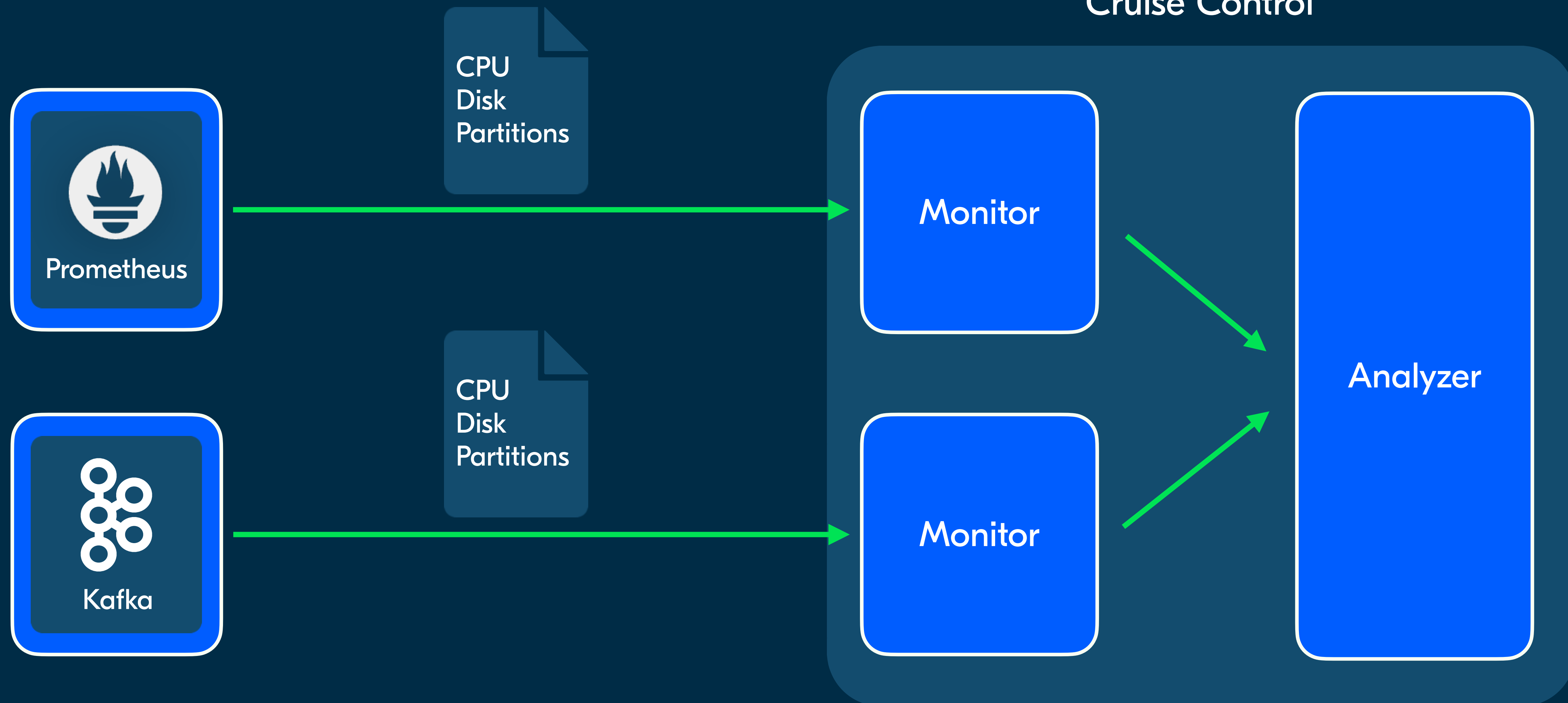
Модуль Monitor

Как устроен Cruise Control



Модуль Monitor

Как устроен Cruise Control



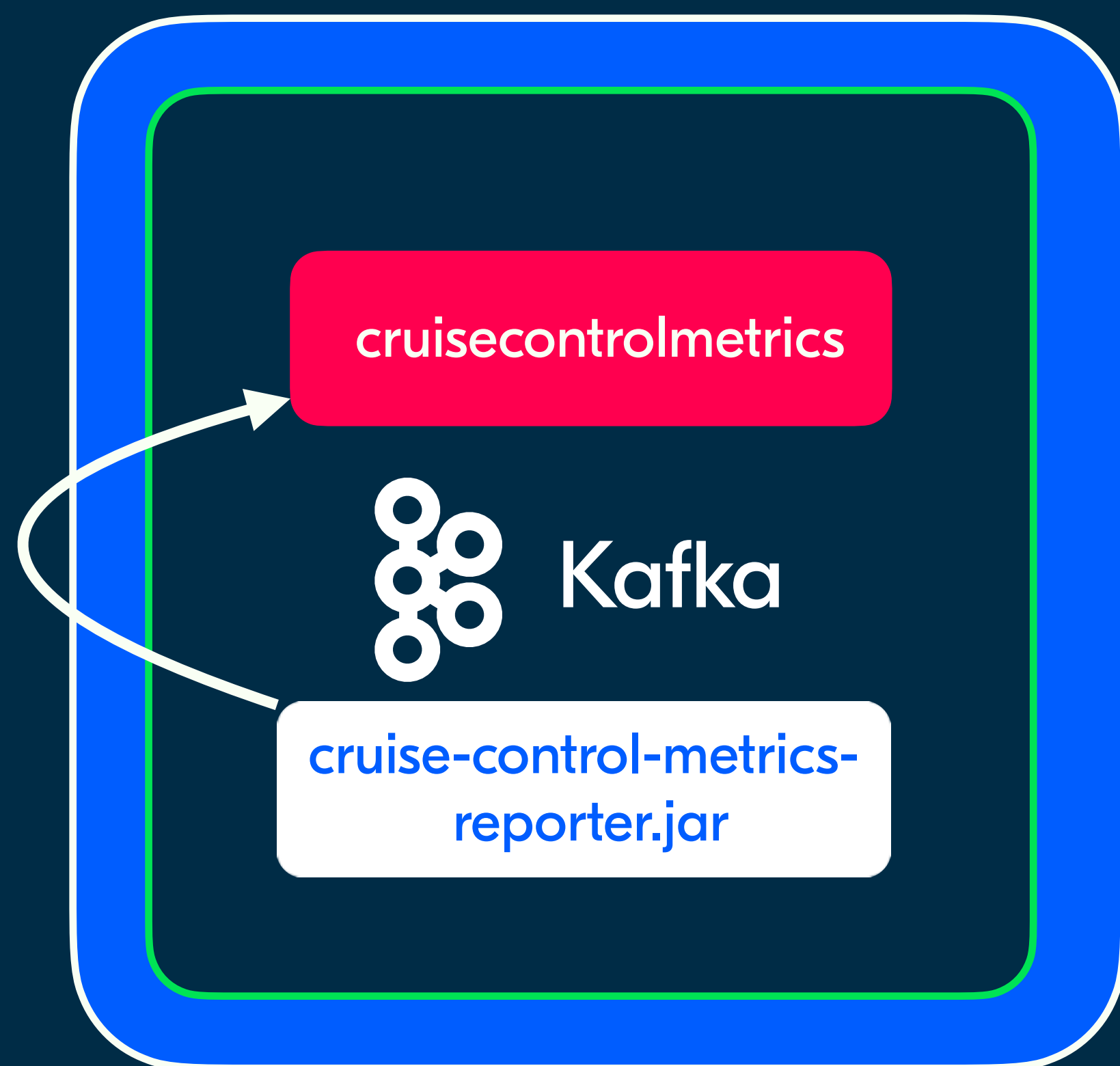
Модуль Monitor

Как устроен Cruise Control



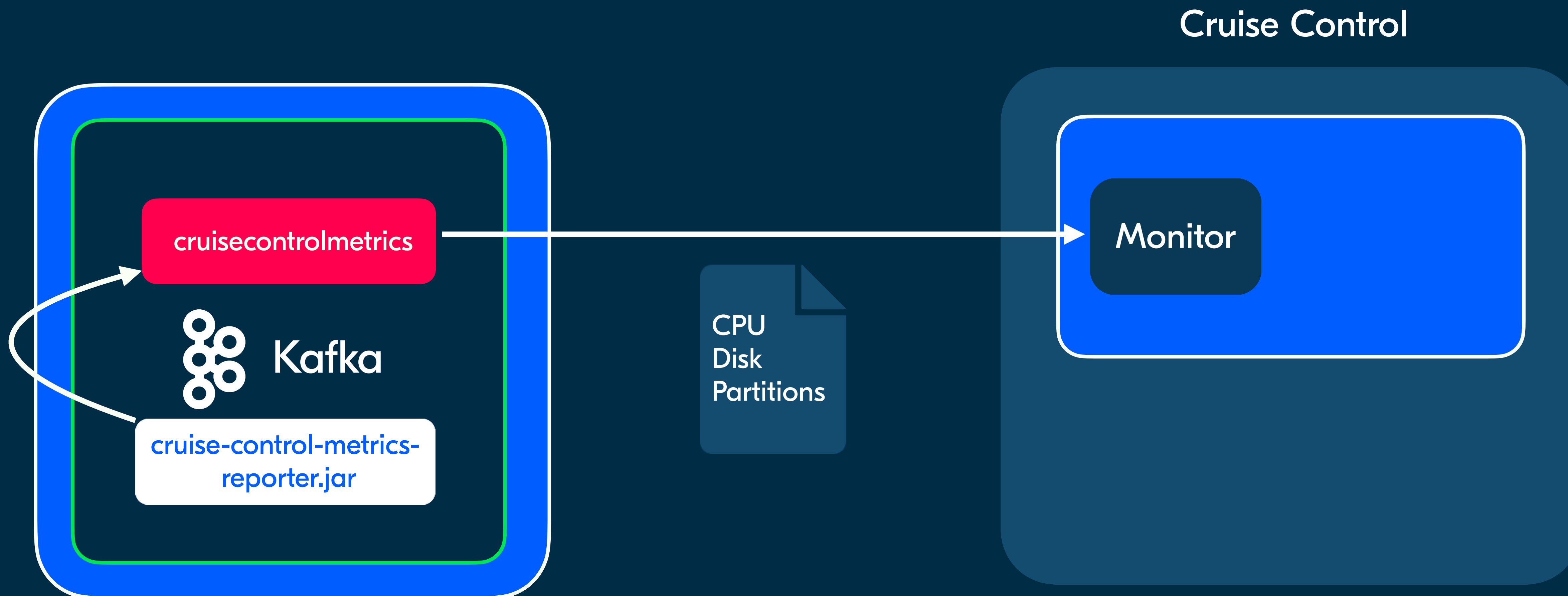
Модуль Monitor

Как устроен Cruise Control



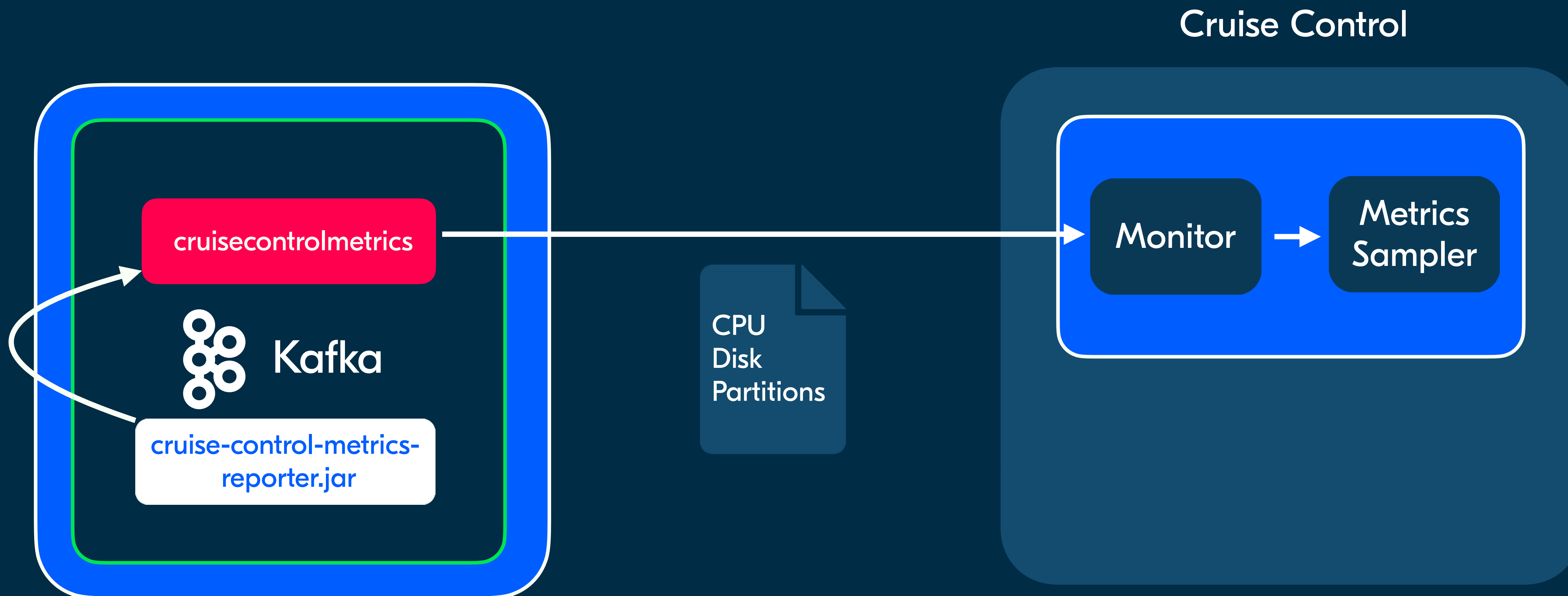
Модуль Monitor

Как устроен Cruise Control



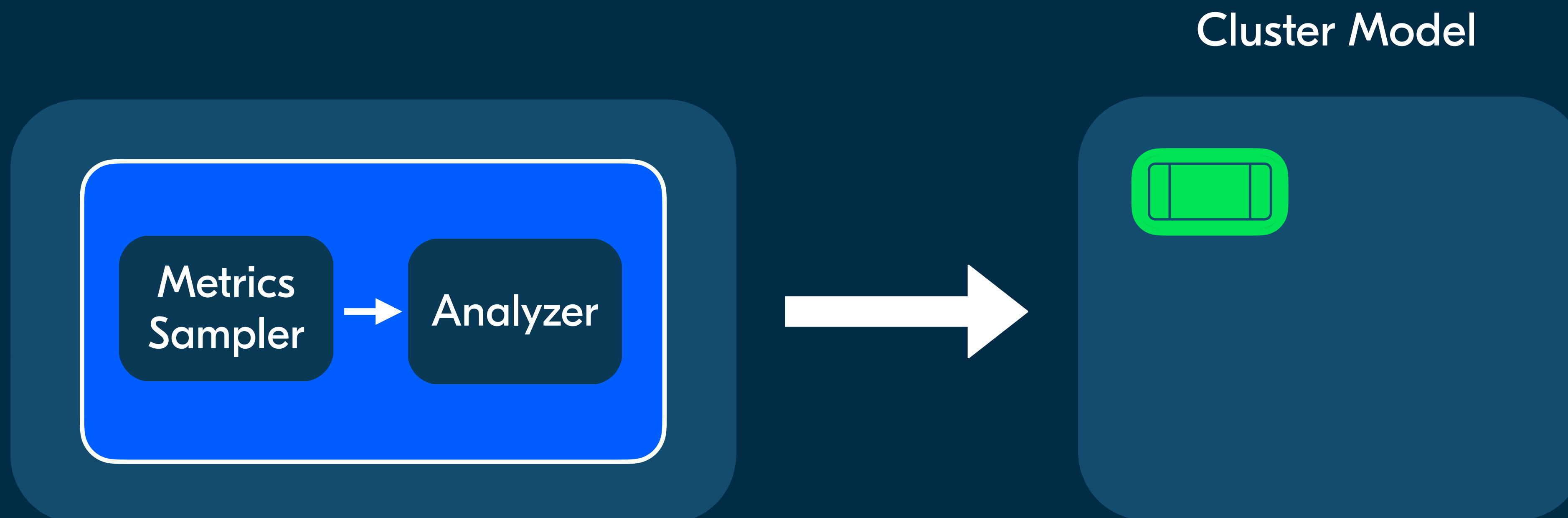
Модуль Monitor

Как устроен Cruise Control



Модуль Analyzer

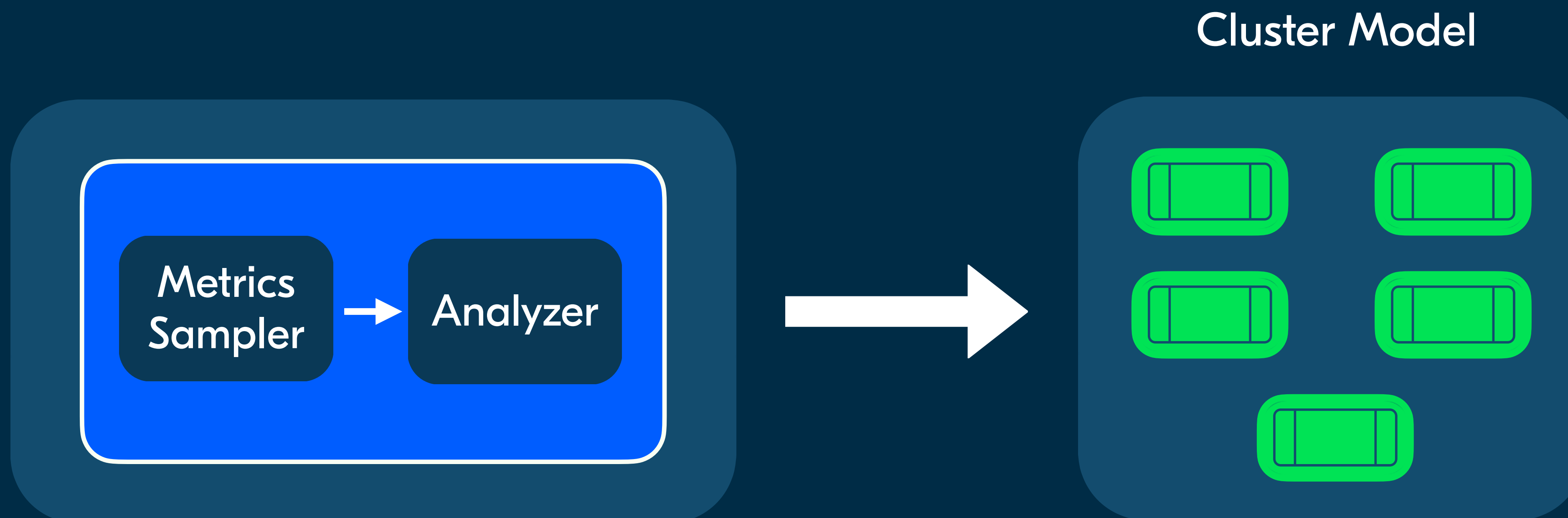
Как устроен Cruise Control



Analyzer вычитывает из sampler-а агрегаты метрик с частотой `metric.sampling.interval.ms`

Модуль Analyzer

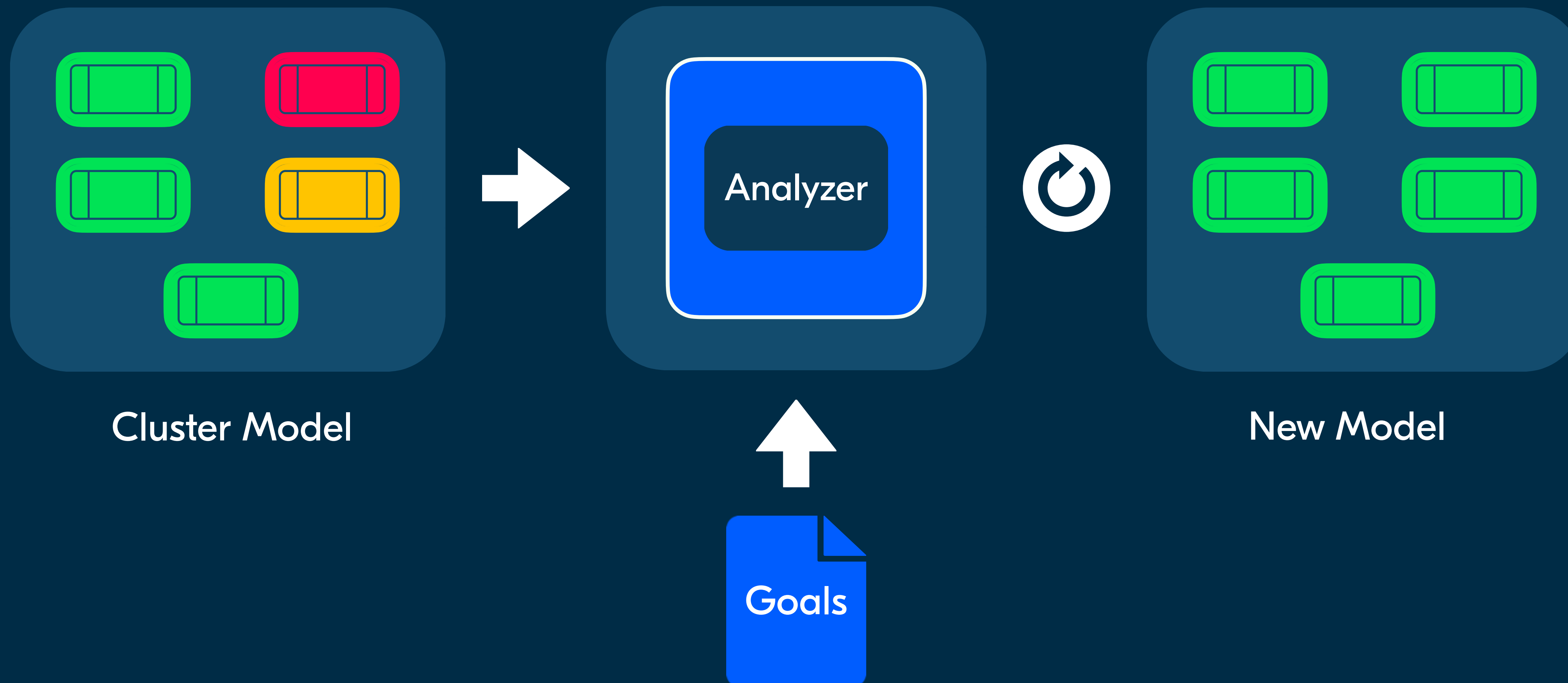
Как устроен Cruise Control



Analyzer вычитывает из sampler-а агрегаты метрик с частотой `metric.sampling.interval.ms`

Модуль Analyzer

Как устроен Cruise Control



Балансировка ресурсов: Cruise Control

Распределяем ресурсы по брокерам и ДЦ

- Снижаем неравномерность утилизации
- Перебалансировка кластера по одной кнопке
- Легкий ввод/вывод брокера в/из эксплуатации
- Автоматическое восстановление RF

Балансировка ресурсов: Cruise Control

Распределяем ресурсы по брокерам и ДЦ

- Снижаем неравномерность утилизации
- Перебалансировка кластера по одной кнопке
- Легкий ввод/вывод брокера в/из эксплуатации
- Автоматическое восстановление RF

Подробнее о сложностях внедрения в статье на Хабре

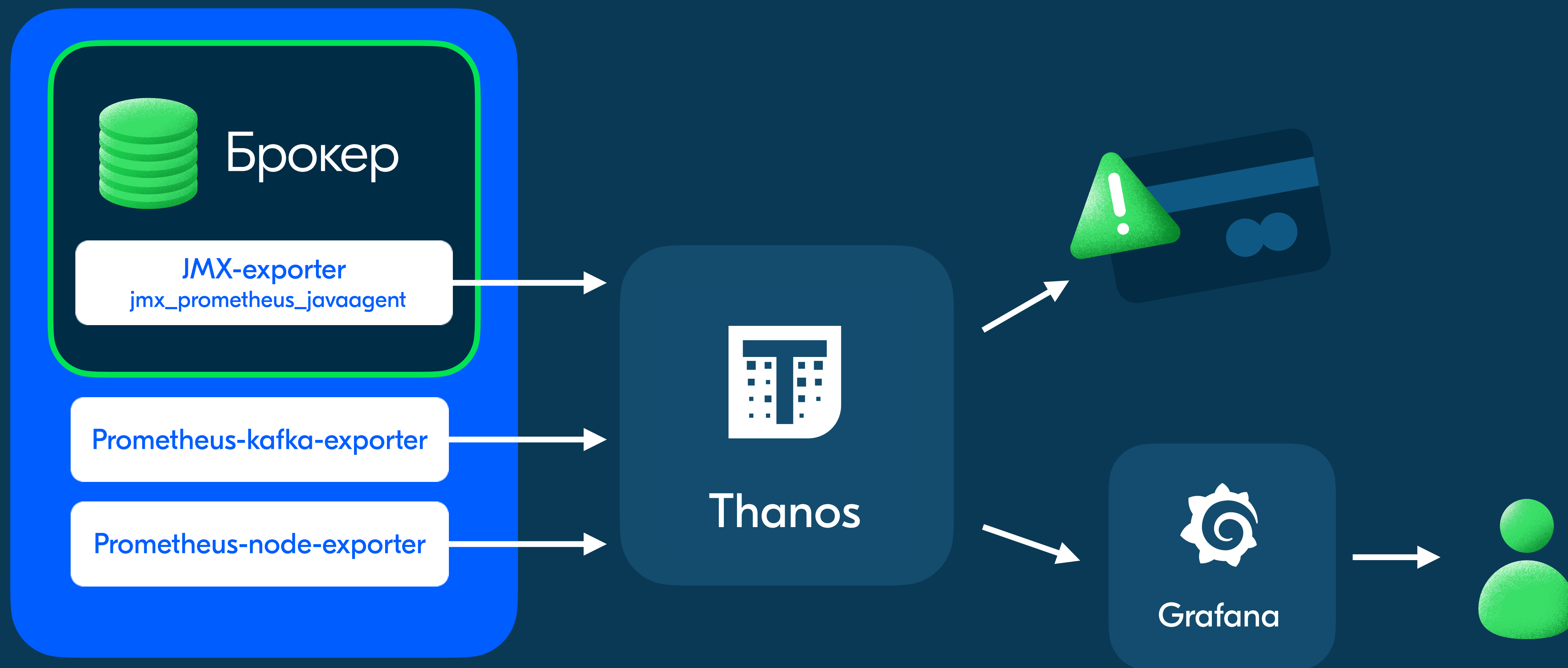




Мониторинг кластера

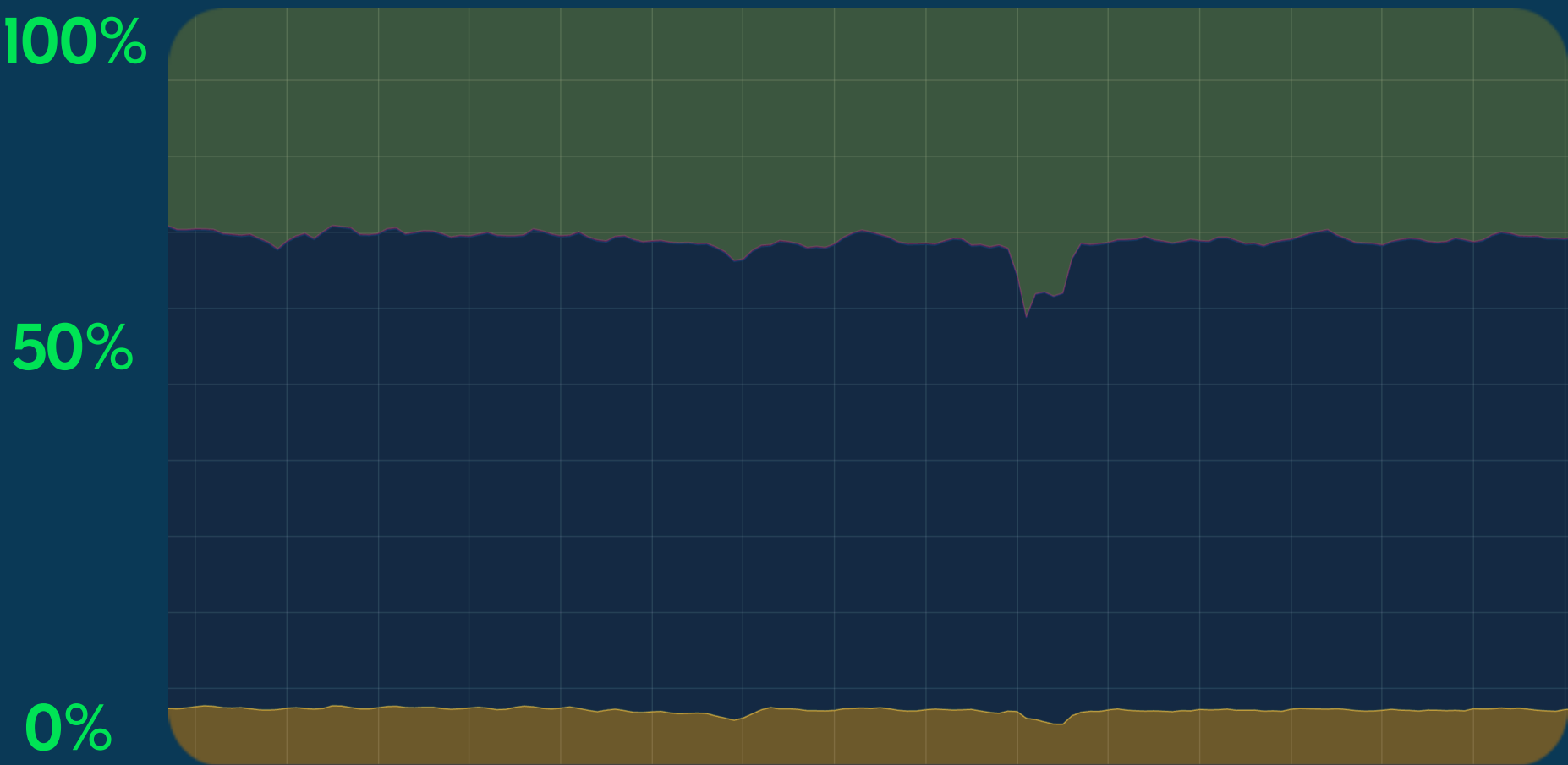
Какие данные мы собираем о «здоровье» брокеров?

Как устроен мониторинг

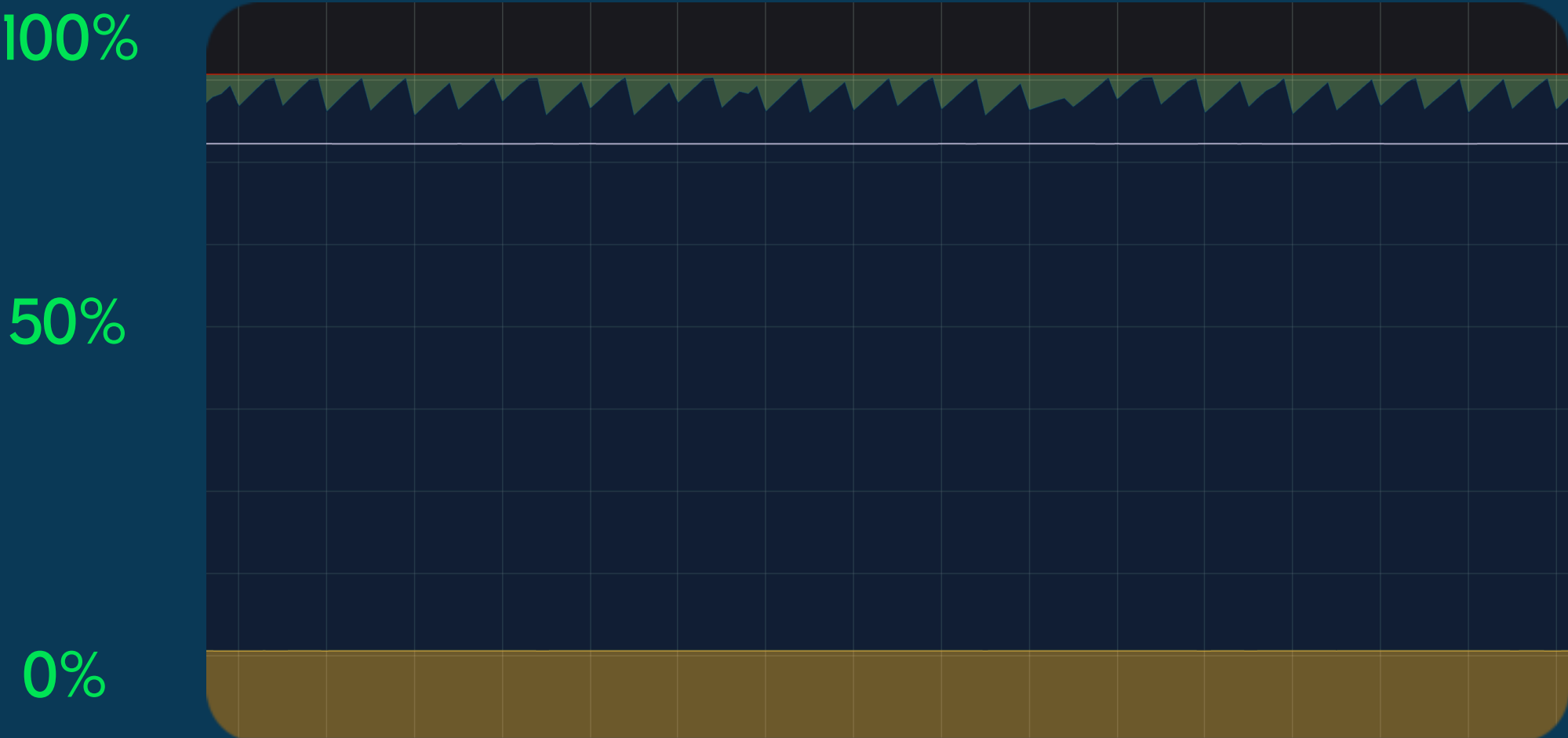


System details

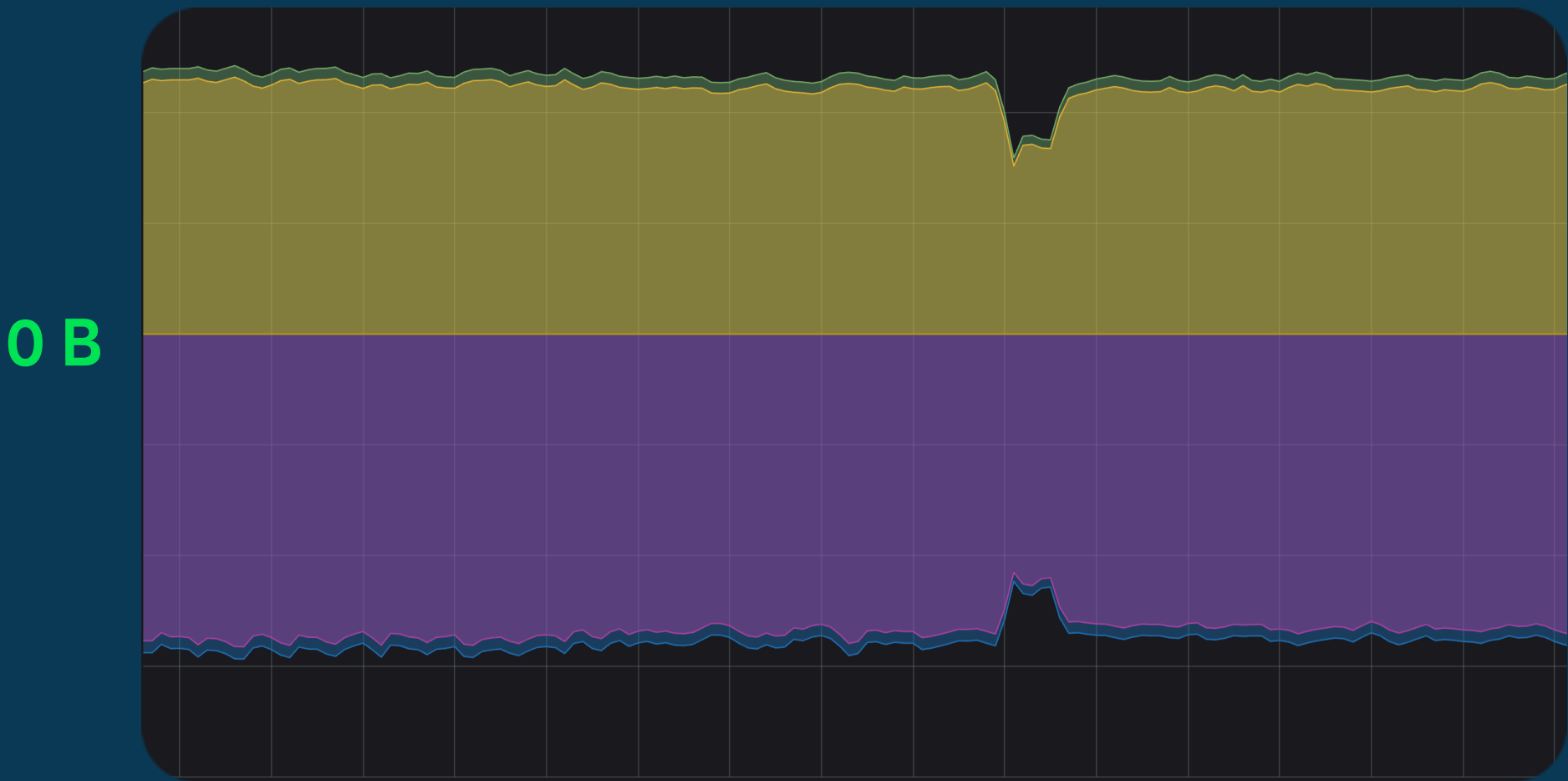
CPU



Memory



Network Traffic



Disk Space Used



Latency (q0.99)

Интегральная метрика всех живых брокеров

64ms

32ms

45



OffsetCommit

Produce

Latency (q0.99)

Интегральная метрика всех живых брокеров

Latency (q0.99) < 90ms



Latency (q0.99) > 90ms



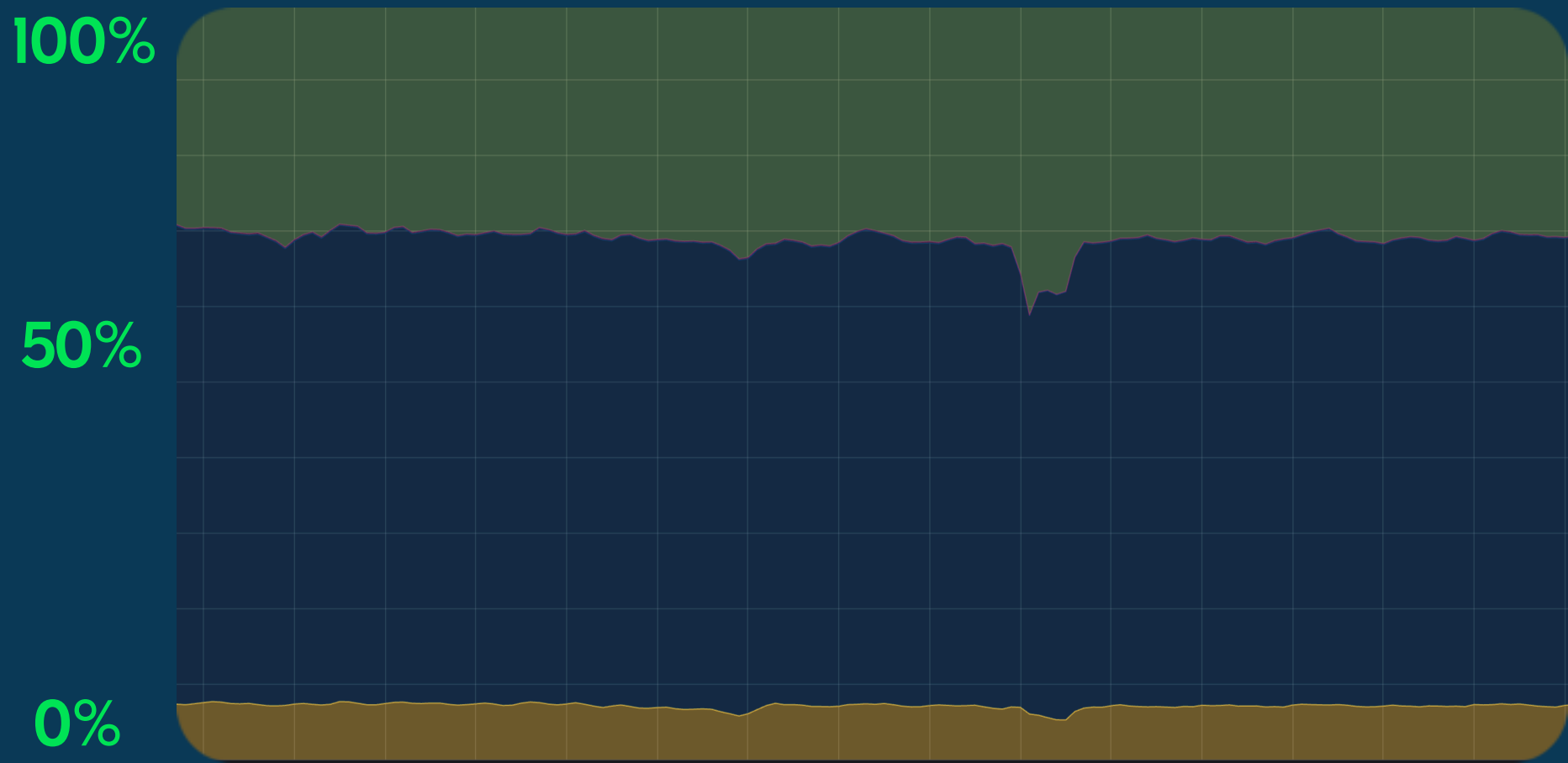
Latency (q0.99)

Интегральная метрика всех живых брокеров

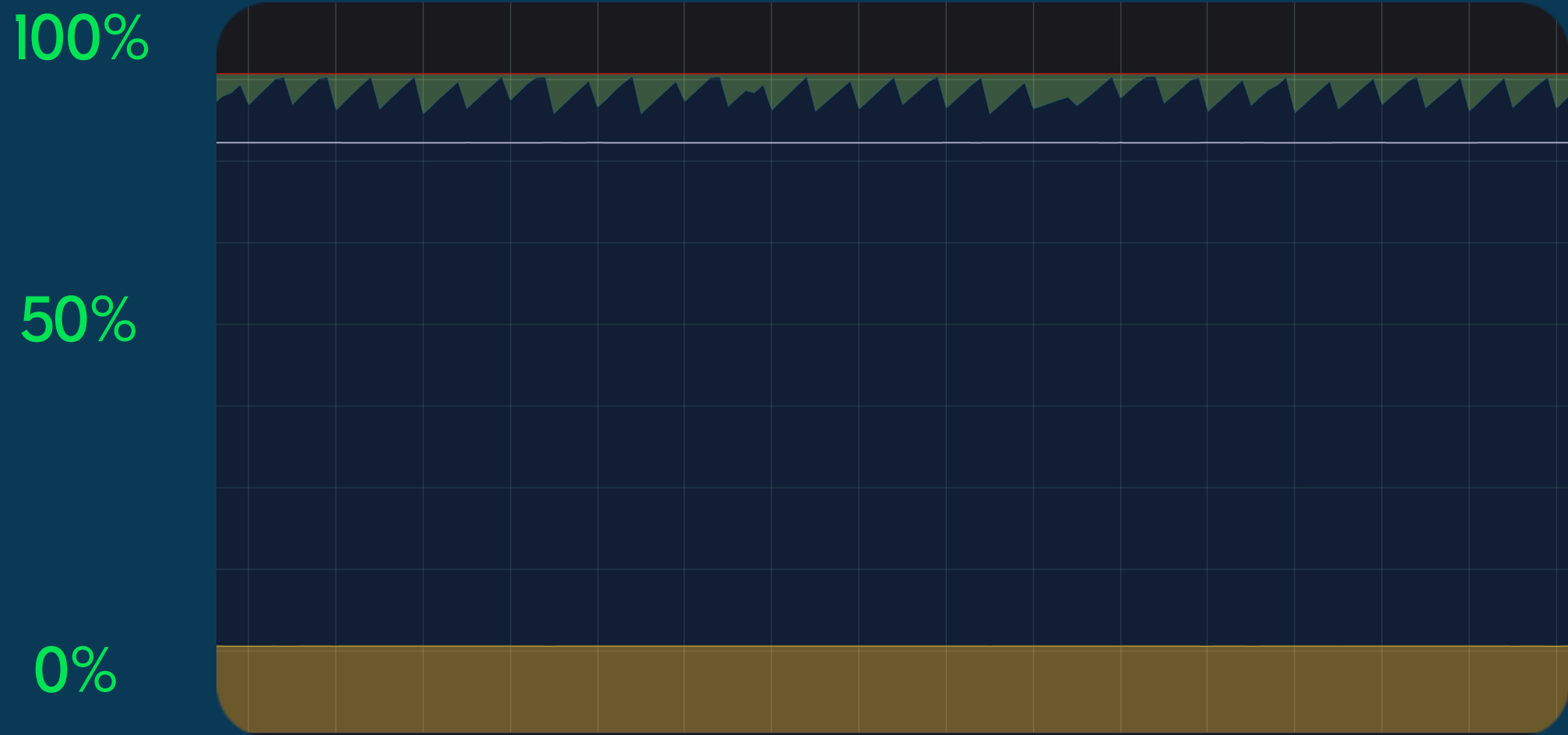


В Багдаде все спокойно

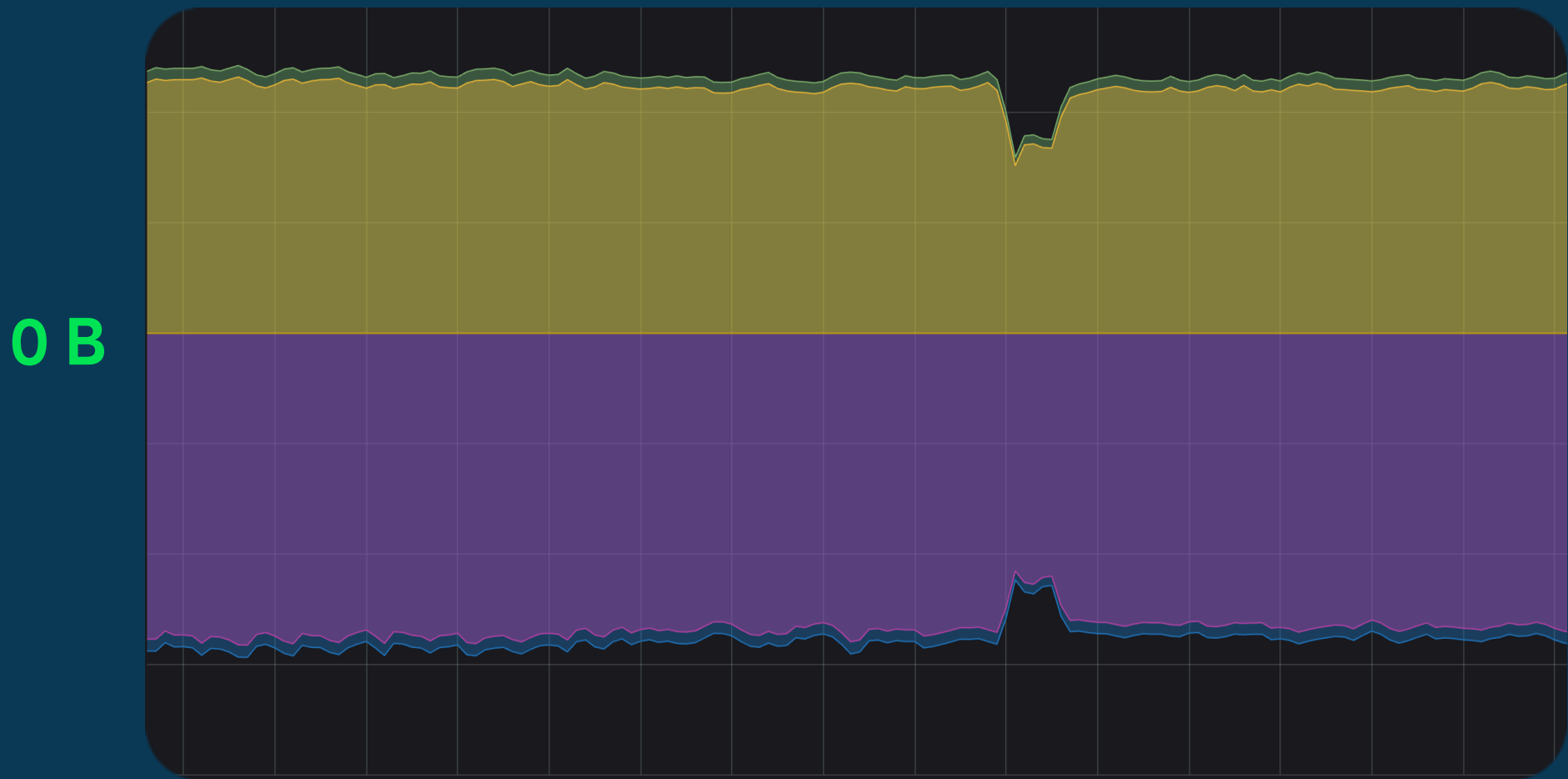
CPU



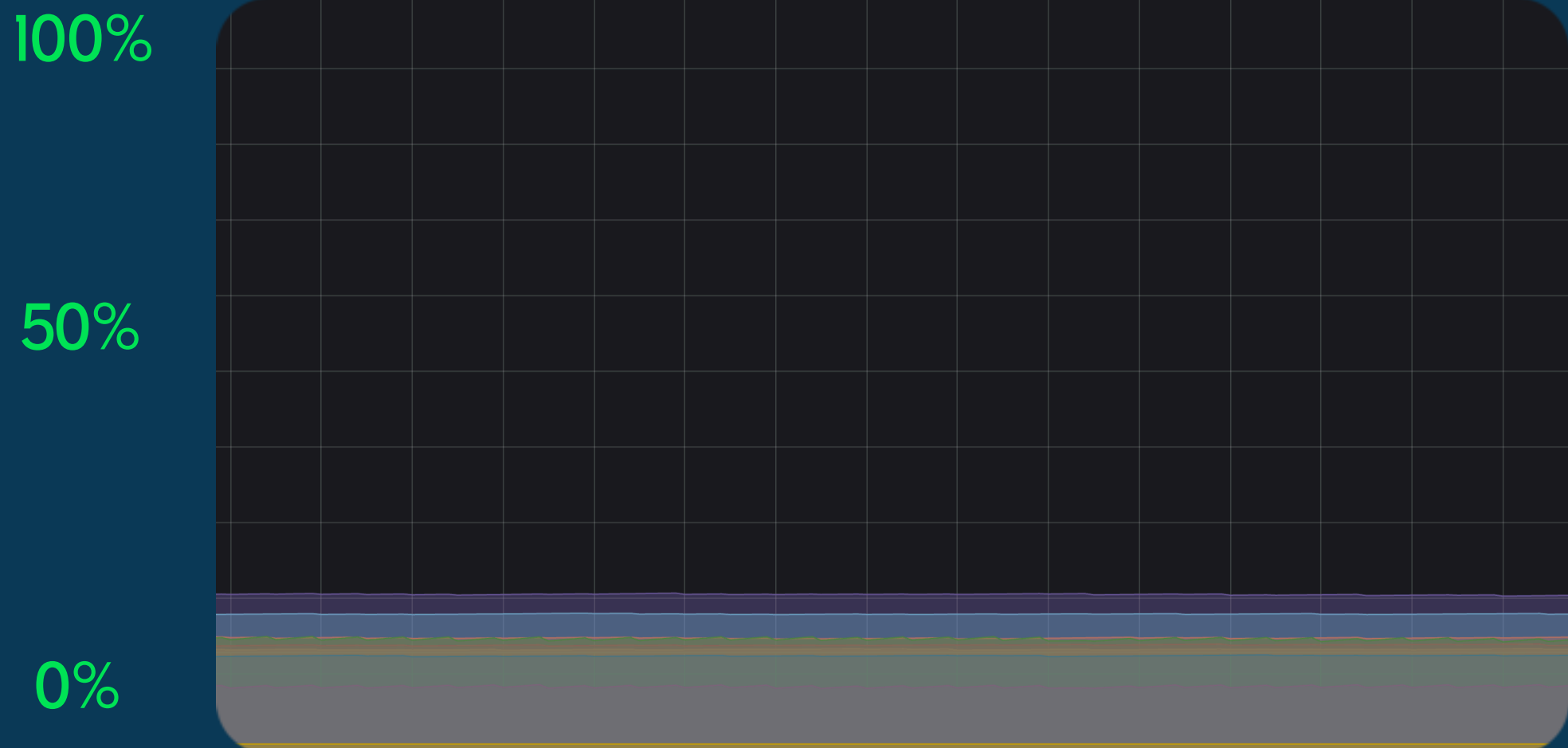
Memory



Network Traffic



Disk Space Used



Проблема!

Latency растёт:

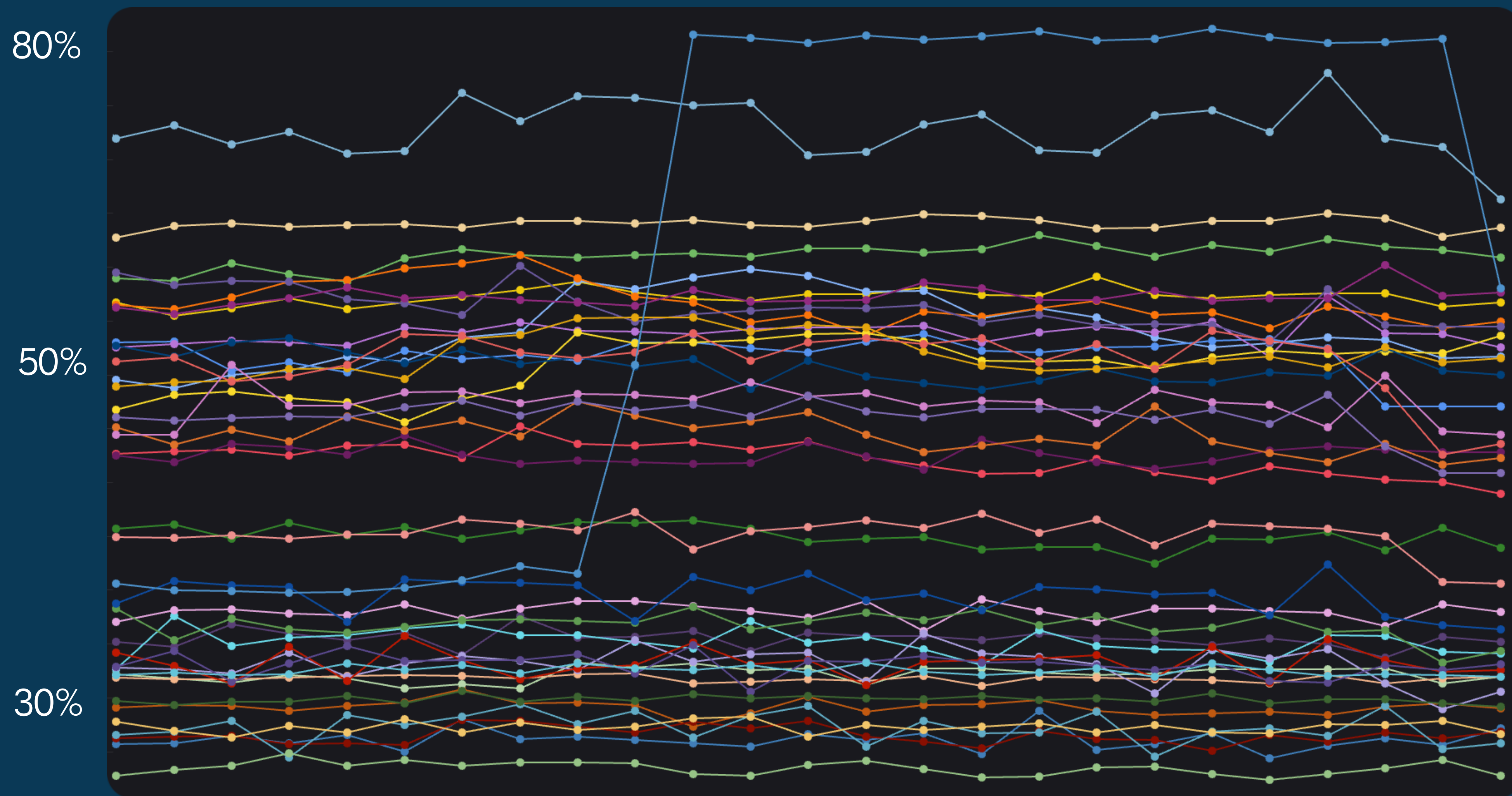
- offset commit — у всех
- produce — у нескольких брокеров

При этом CPU, Memory,
Network, Disk не «в полке»



Загруженность IO thread

Нормальная работа



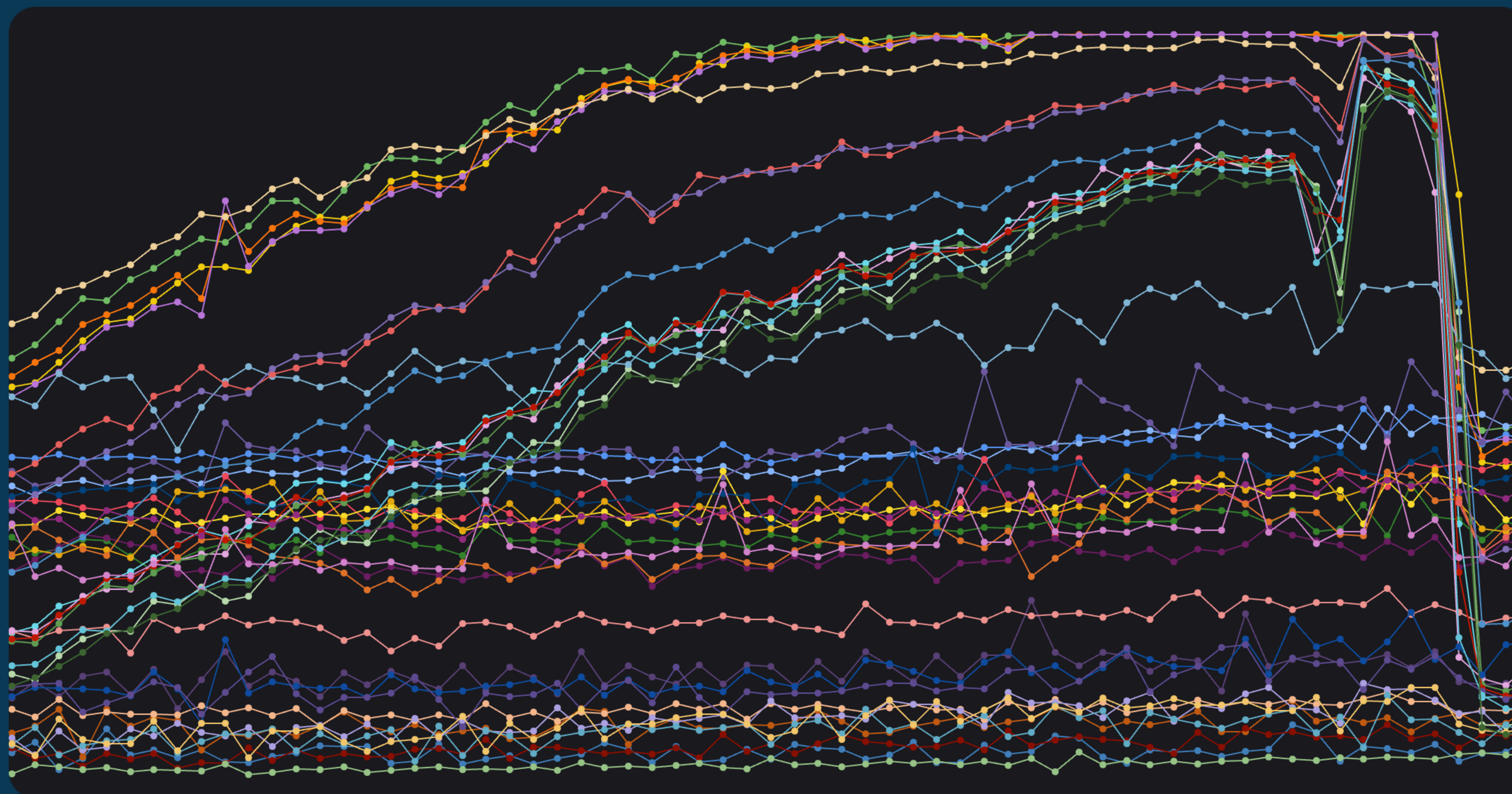
Загруженность IO thread

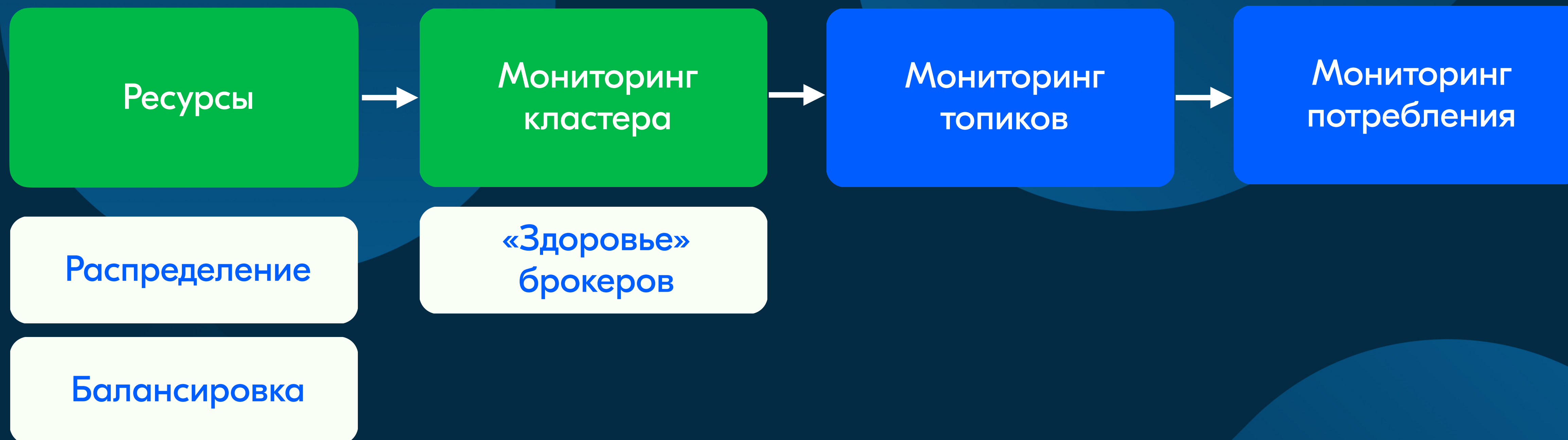
Нагрузочное тестирование

100%

60%

20%





Мониторинг кластера

История одного инцидента, или Зачем нужен профайлинг

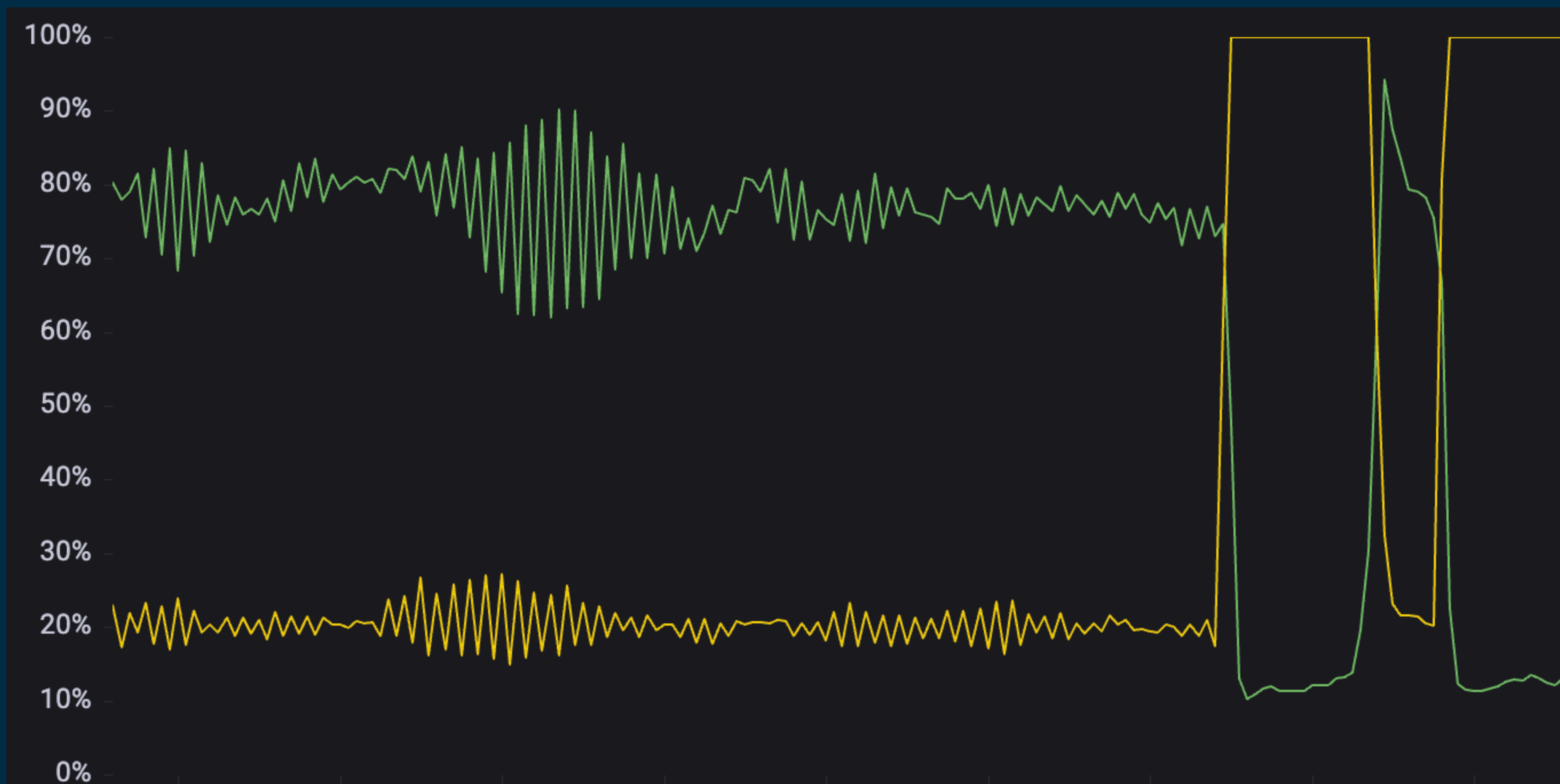
Загруженность IO-тредов по брокерам



Загруженность IO-тредов по брокерам



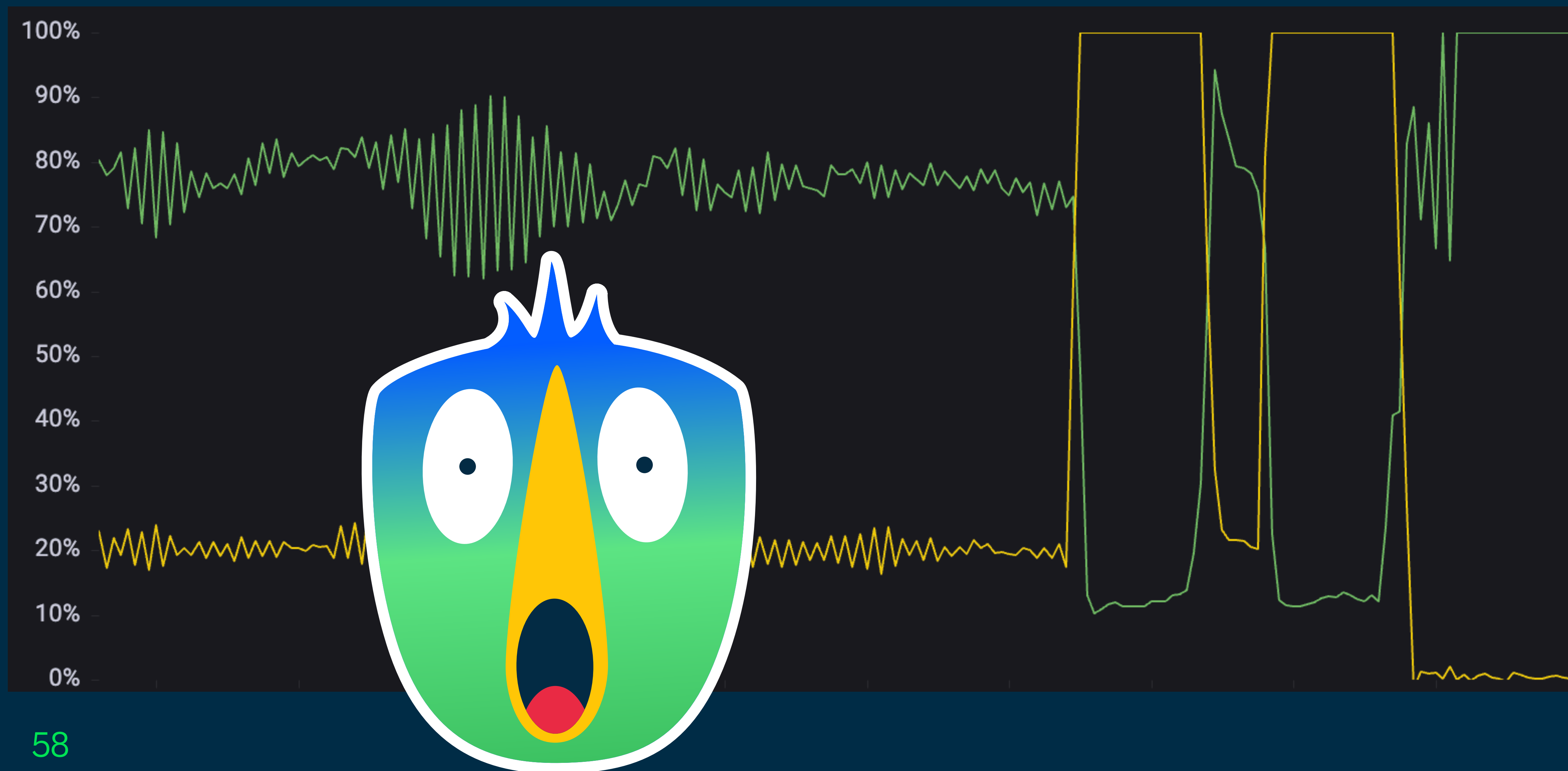
Загруженность IO-тредов по брокерам



Загруженность IO-тредов по брокерам



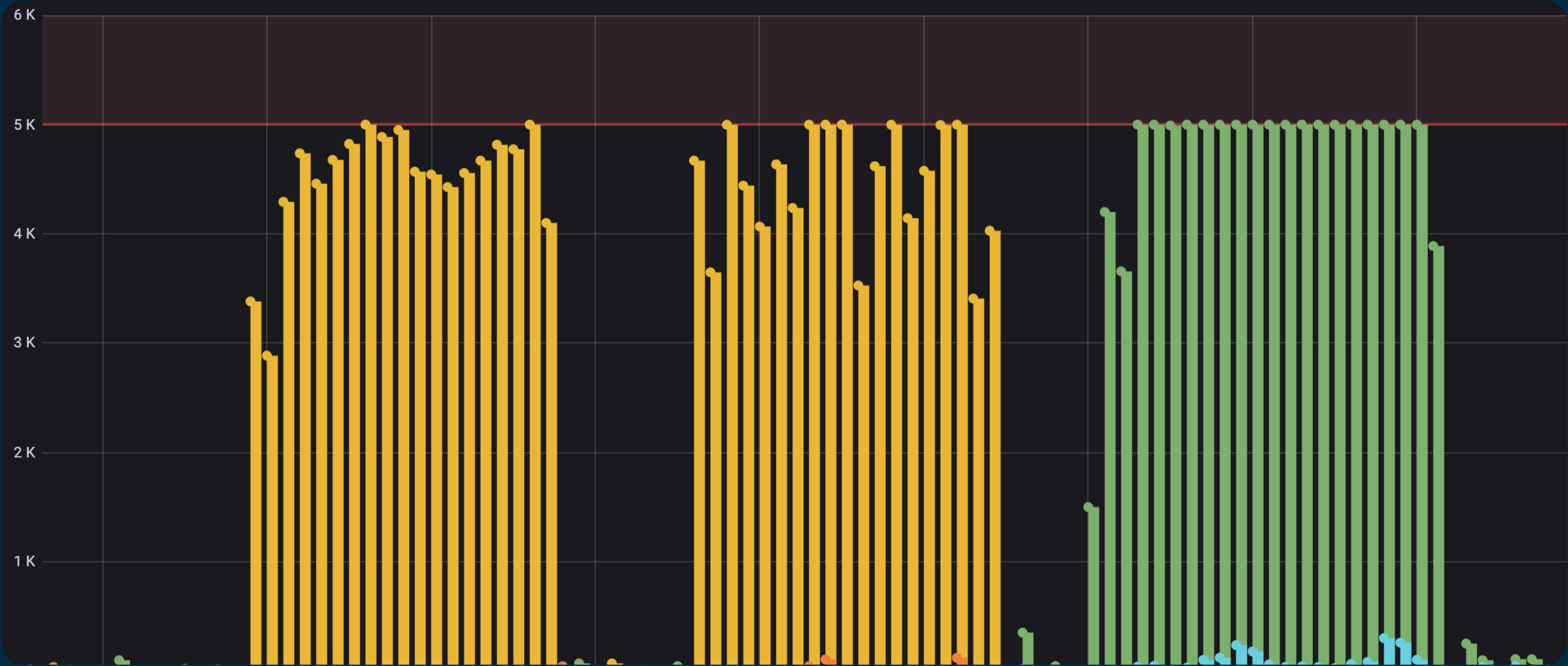
Загруженность IO-тредов по брокерам



Загруженность IO-тредов по брокерам



Request/Response Queue size



Поиски

Что мы сделали?

ozon{tech

Поиски

Что мы сделали?

1. Нашли лидер-партицию, которая переехала с брокера 1 на брокер 2

Поиски

Что мы сделали?

1. Нашли лидер-партицию, которая переехала с брокера 1 на брокер 2
2. Увидели, что у этого топика резко возросло число чтений

Поиски

Что мы сделали?

1. Нашли лидер-партицию, которая переехала с брокера 1 на брокер 2
2. Увидели, что у этого топика резко возросло число чтений
3. Было несколько консьюмер-групп — и все они оказались не при делах

Поиски

Что мы сделали?

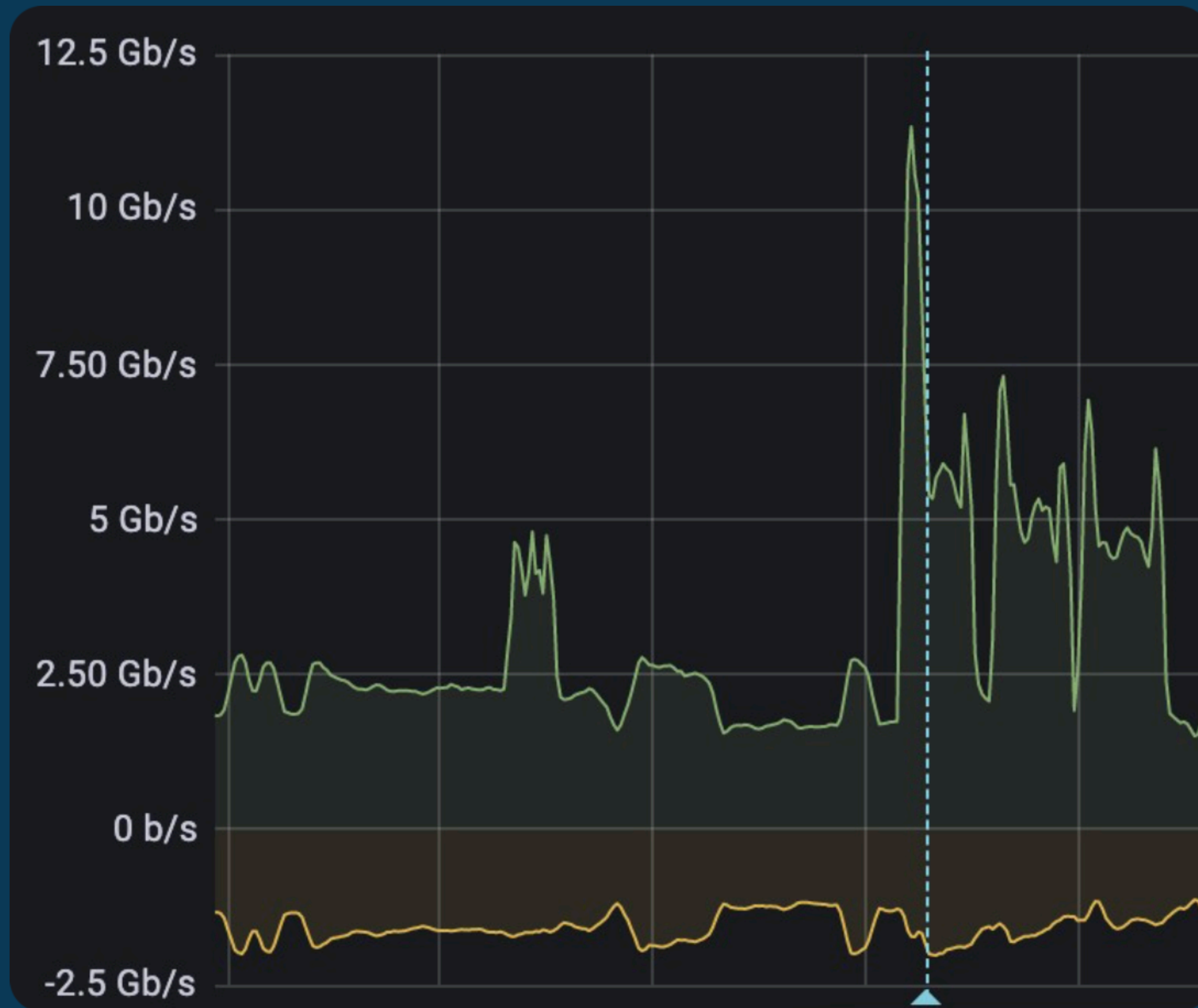
1. Нашли лидер-партицию, которая переехала с брокера 1 на брокер 2
2. Увидели, что у этого топика резко возросло число чтений
3. Было несколько консьюмер-групп — и все они оказались не при делах
4. Вспомнили, что один из коллег, накануне задавал странные вопросы:

Поиски

Что мы сделали?

1. Нашли лидер-партицию, которая переехала с брокера 1 на брокер 2
2. Увидели, что у этого топика резко возросло число чтений
3. Было несколько консьюмер-групп — и все они оказались не при делах
4. Вспомнили, что один из коллег, накануне задавал странные вопросы:

А что, если будет сервис с **500** подами, которые будут стартовать, и каждый будет **одновременно** вычитывать все сообщения из определенного топика?



А что, если будет сервис с **500** подами, которые будут стартовать, и каждый будет **одновременно** вычитывать все сообщения из определенного топика?

И тогда мы решили повторить

На dev-окружении, чтобы больше ничего не задеть

И тогда мы решили повторить

На dev-окружении, чтобы больше ничего не задеть

1. К сожалению, там у нас не было столько подов

И тогда мы решили повторить

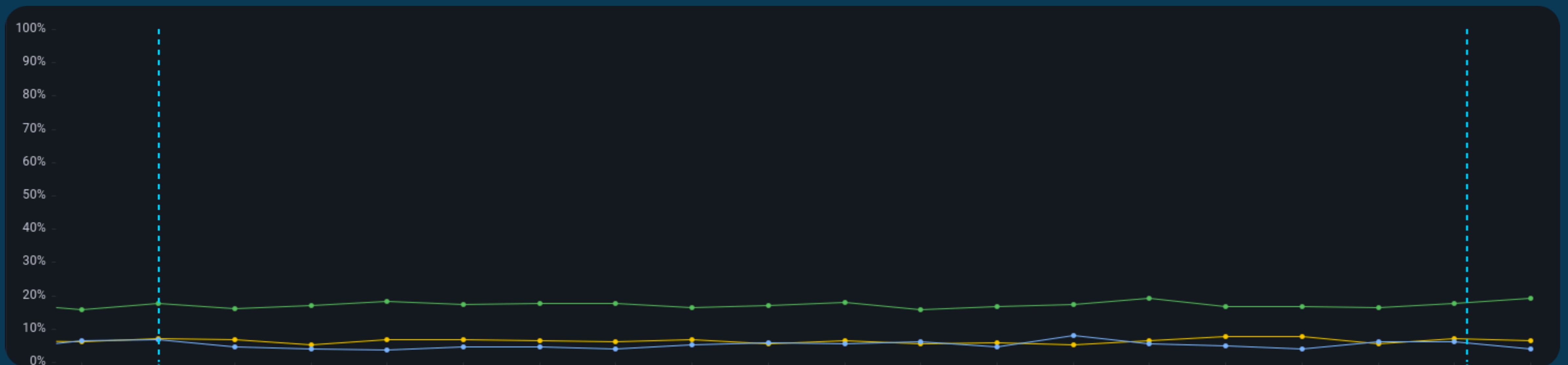
На dev-окружении, чтобы больше ничего не задеть

1. К сожалению, там у нас не было столько подов
2. Да и фоновая нагрузка почти нулевая

И тогда мы решили повторить

На dev-окружении, чтобы больше ничего не задеть

1. К сожалению, там у нас не было столько подов
2. Да и фоновая нагрузка почти нулевая



И тогда мы решили повторить

На prod с тем же топиком, с тем же сервисом очень аккуратно

И тогда мы решили повторить

На prod с тем же топиком, с тем же сервисом очень аккуратно

1. Но у топика добавилось еще 2 партиции

И тогда мы решили повторить

На prod с тем же топиком, с тем же сервисом очень аккуратно

1. Но у топика добавилось еще 2 партии
2. А суммарный размер уменьшился в несколько раз

И тогда мы решили повторить

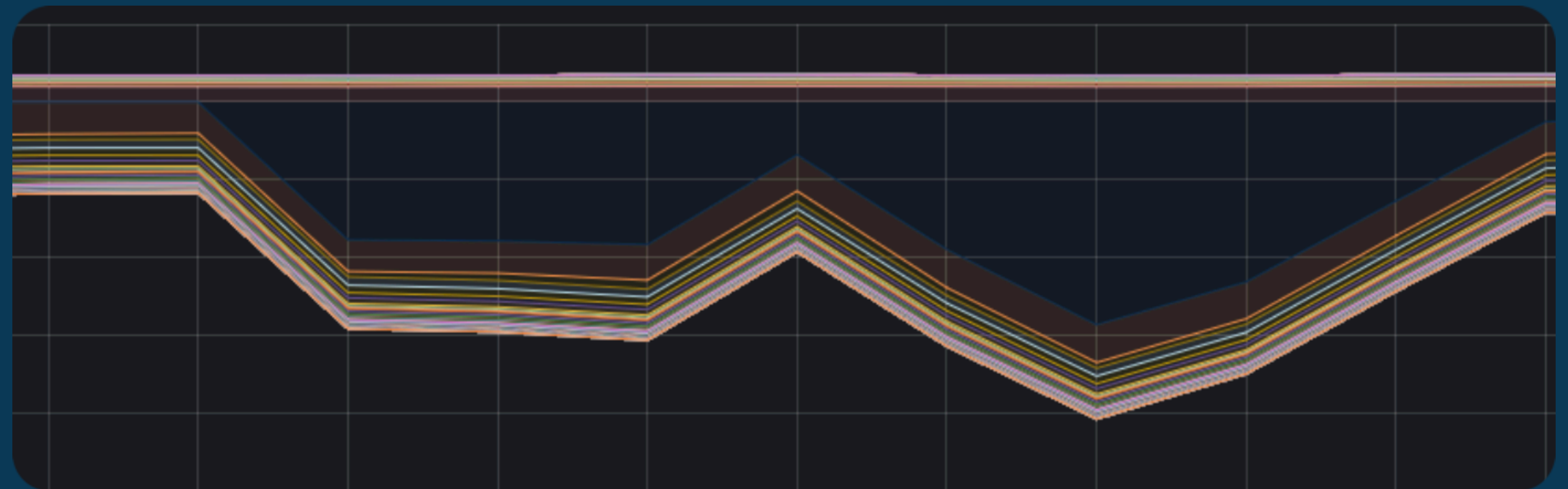
На prod с тем же топиком, с тем же сервисом очень аккуратно

1. Но у топика добавилось еще 2 партиции
2. А суммарный размер уменьшился в несколько раз
3. Короче...повторить не вышло

И тогда мы решили повторить

На prod с тем же топиком, с тем же сервисом очень аккуратно

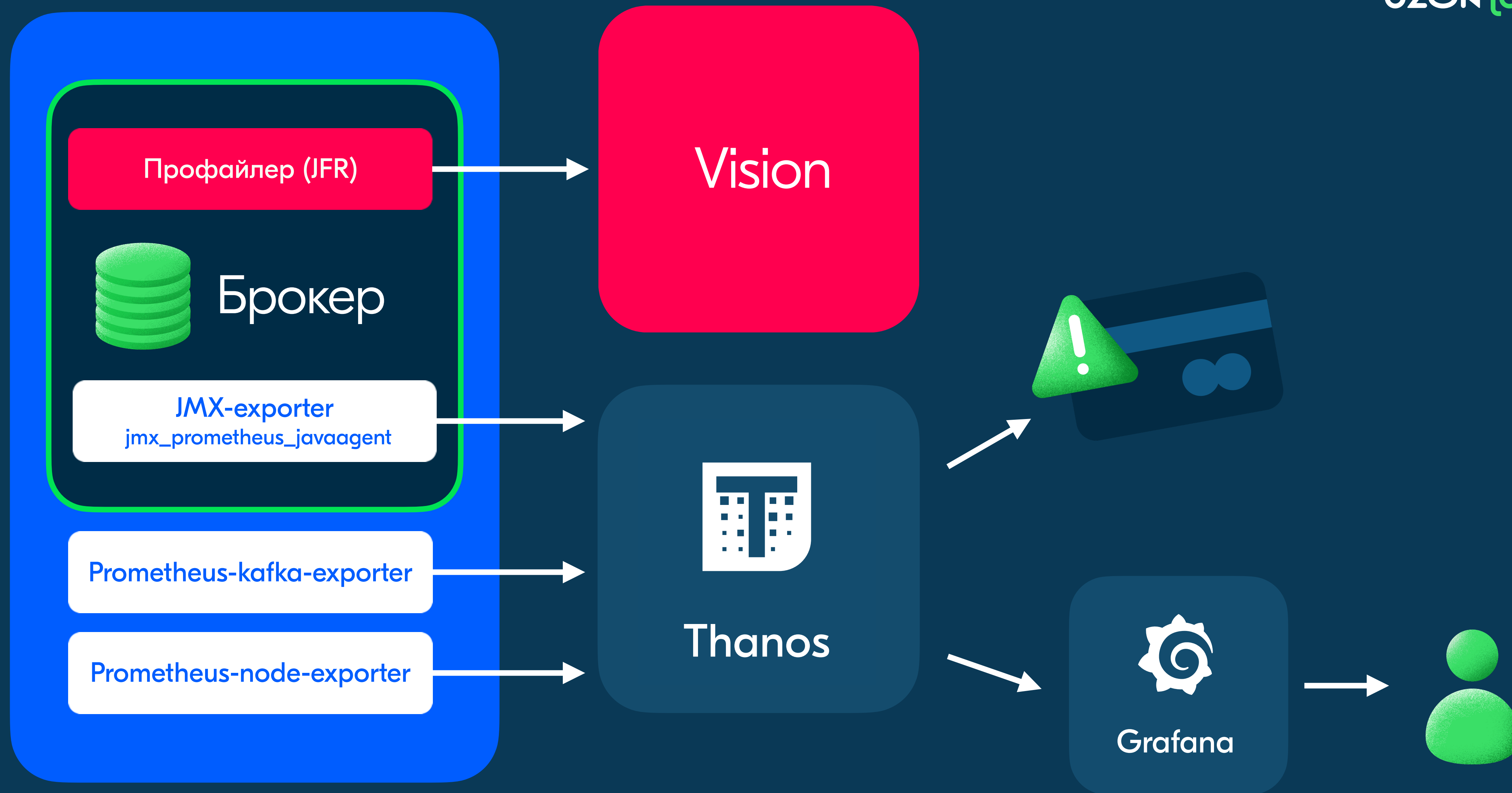
1. Но у топика добавилось еще 2 партии
2. А суммарный размер уменьшился в несколько раз
3. Короче...повторить не вышло



Зачем повторять?

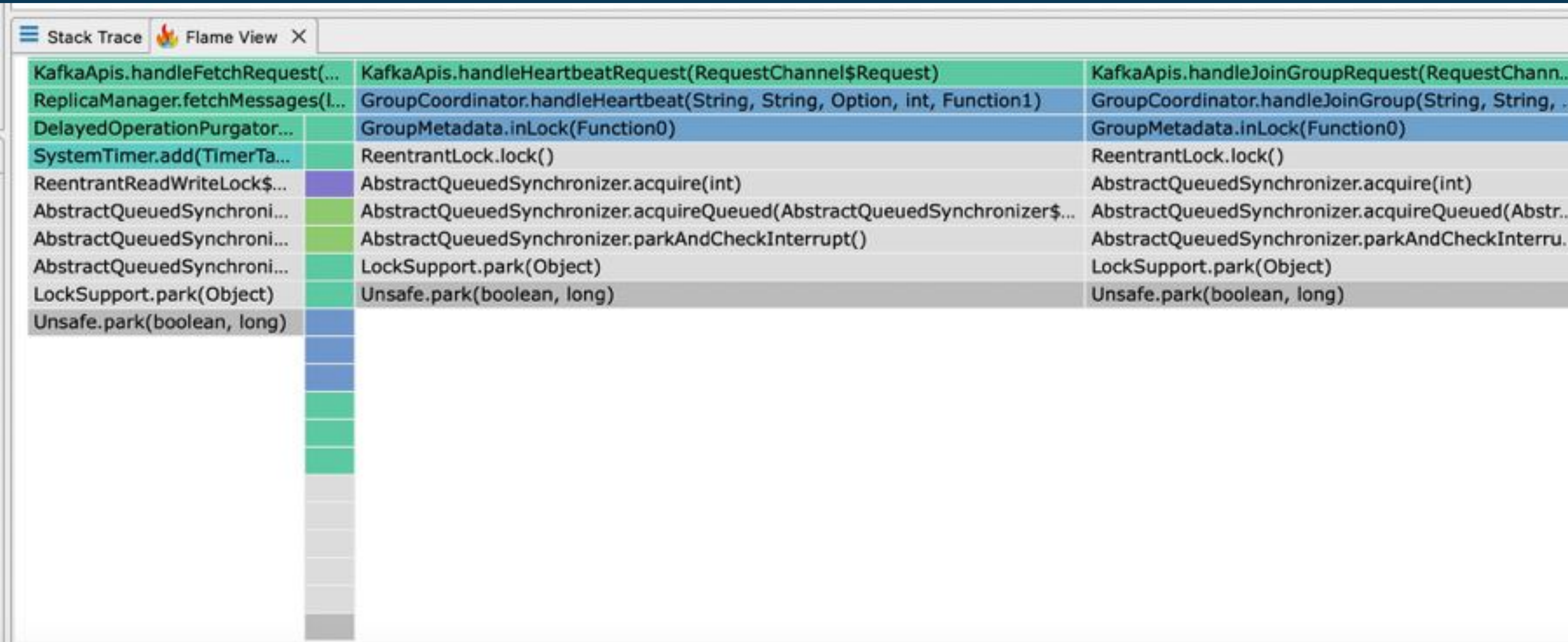
Нам важно понять, **чем конкретно был занят брокер**, чтобы не допустить этого в будущем.

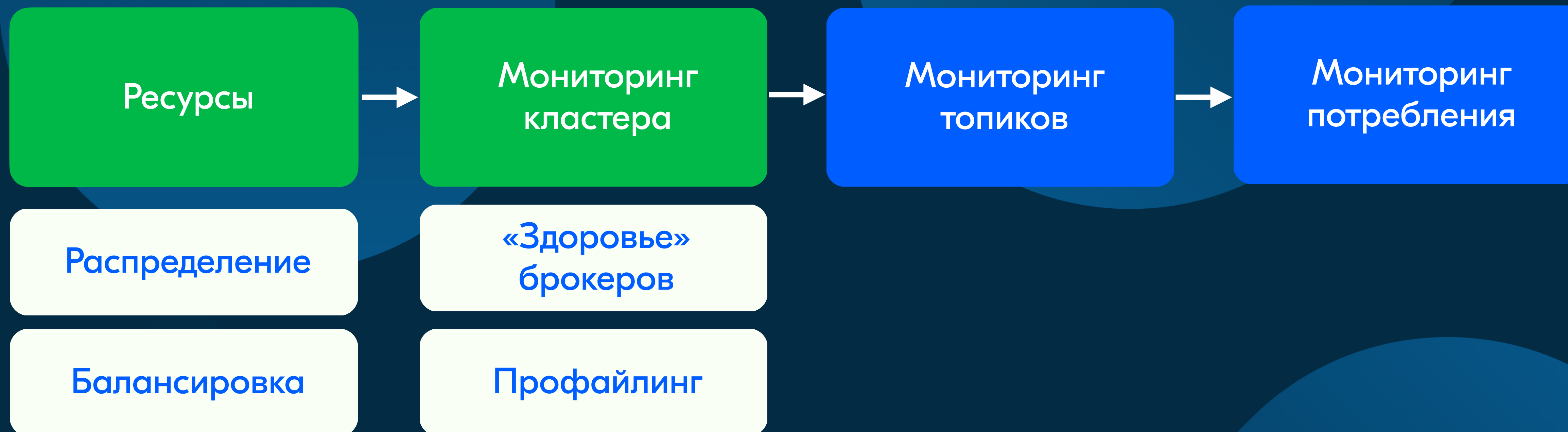
Для этого нужно снять дампы —
профайлинг трейдов



Профайлинг

Пример дампа — видно конкретные методы, которые занимают треды



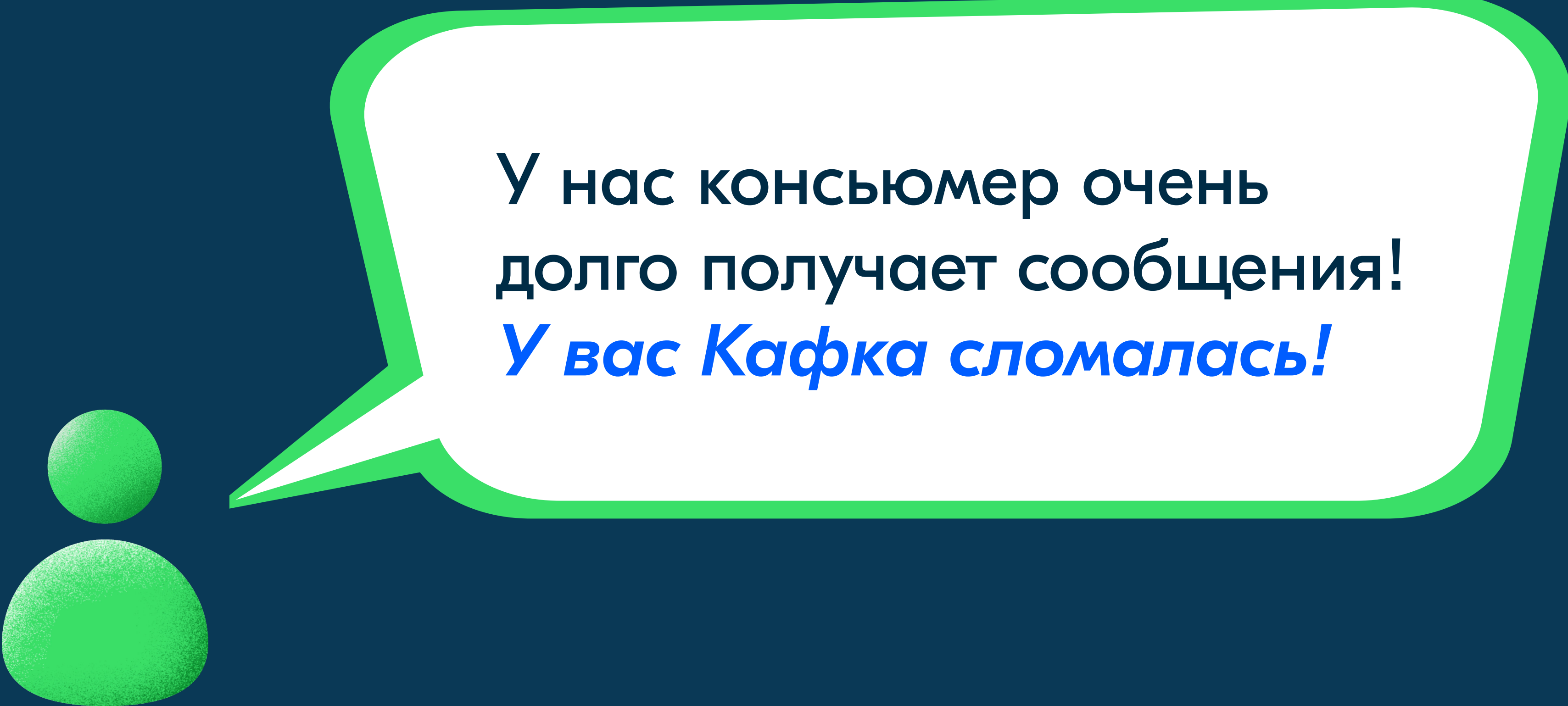


Мониторинг кластера

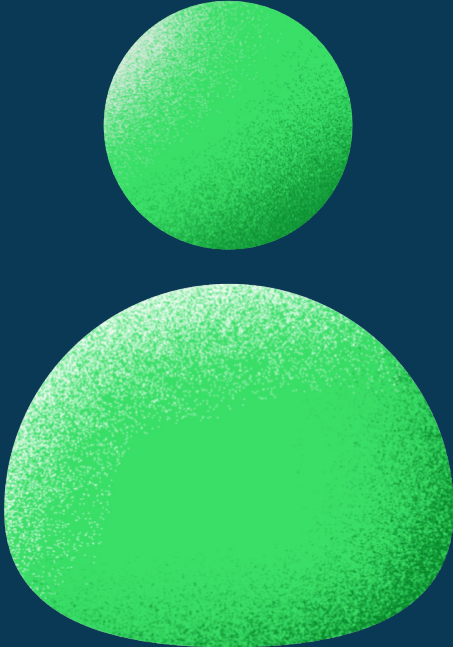
Почему мониторить брокеры изнутри недостаточно?

Могут ли наши показатели Latency не давать объективной картины?

Что, если продуктовый сервис испытывает проблемы, а мы их не видим?




У нас консьюмер очень
долго получает сообщения!
*У вас **Кафка** сломалась!*



У нас консьюмер очень
долго получает сообщения!
У вас Кафка сломалась!

Все показатели в норме.
*Проблема не
на нашей стороне*



А что если...

ozon{tech

А что если...

1. «показатели» врут?



А что если...

1. «показатели» врут?
2. проблема в инфраструктуре «снаружи» брокера?



А что если...

1. «показатели» врут?
2. проблема в инфраструктуре «снаружи» брокера?
3. проблема специфична для какого-то размера батча или типа операции?



А что если...

1. «показатели» врут?
2. проблема в инфраструктуре «снаружи» брокера?
3. проблема специфична для какого-то размера батча или типа операции?
4. мы могли найти проблему раньше, чем ее заметят продуктовые разработчики?



1. «показатели» врут?
2. проблема в инфраструктуре «снаружи» брокера?
3. проблема специфична для какого-то размера батча или типа операции?
4. мы могли найти проблему раньше, чем ее заметят продуктовые разработчики?

Решение:

Реализовать «модельный» сервис, который:

- Производит стандартные операции на целевом kafka-кластере(produce, fetch, commit, metadata)
- Сдает метрики об этих операциях в Prometheus, по которым можно оценивать состояние кластера/брокеров

- Лидеры партиций равномерно на всех брокерах
- Сообщения отправляются в партиции через round robin
- Produce, fetch, metadata
- Пишем разными батчами и разными размерами сообщений
- Записываем время выполнения и ошибки (в том числе тайм-ауты)
- **ack=1** и **ack=all**

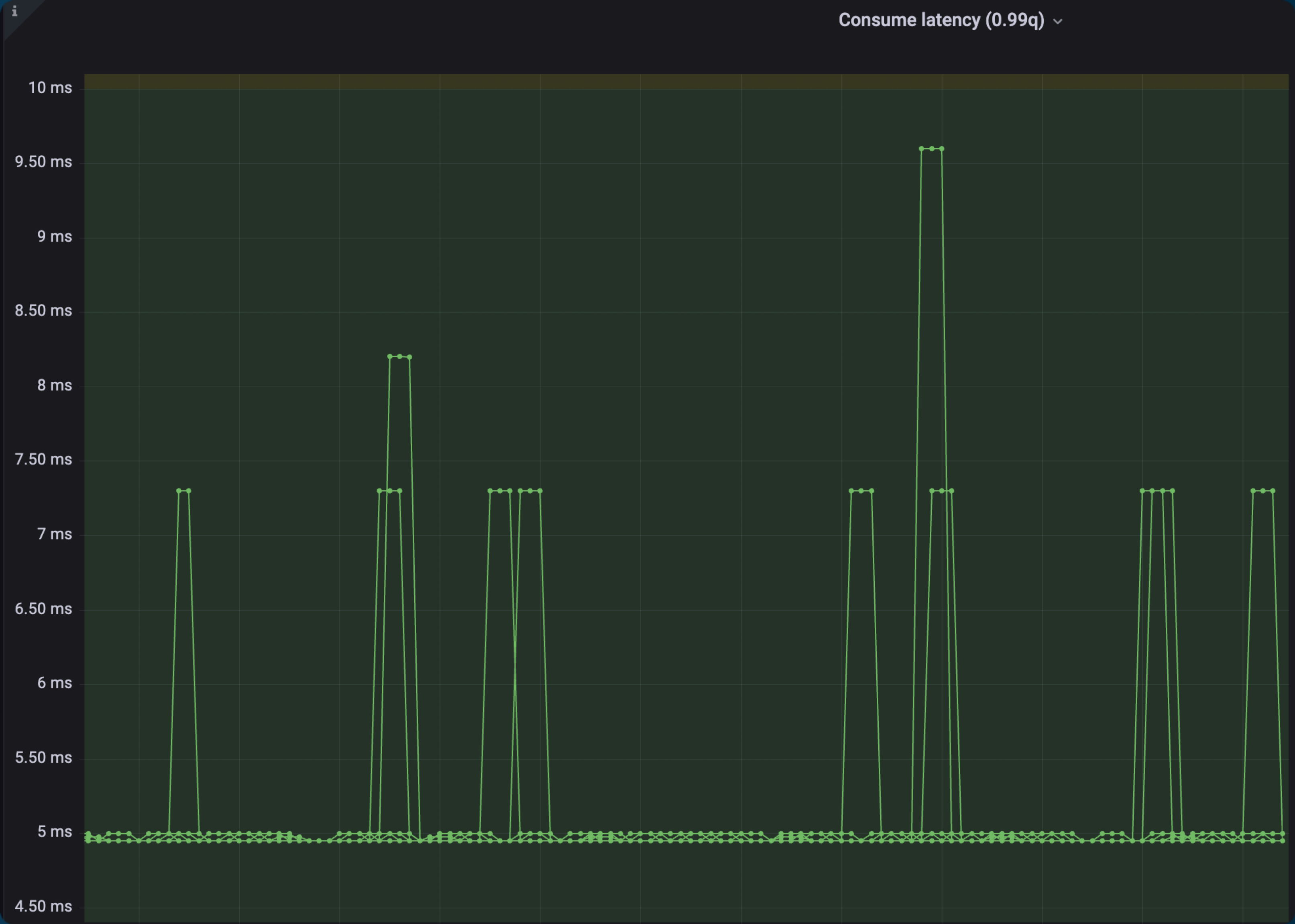
Produce

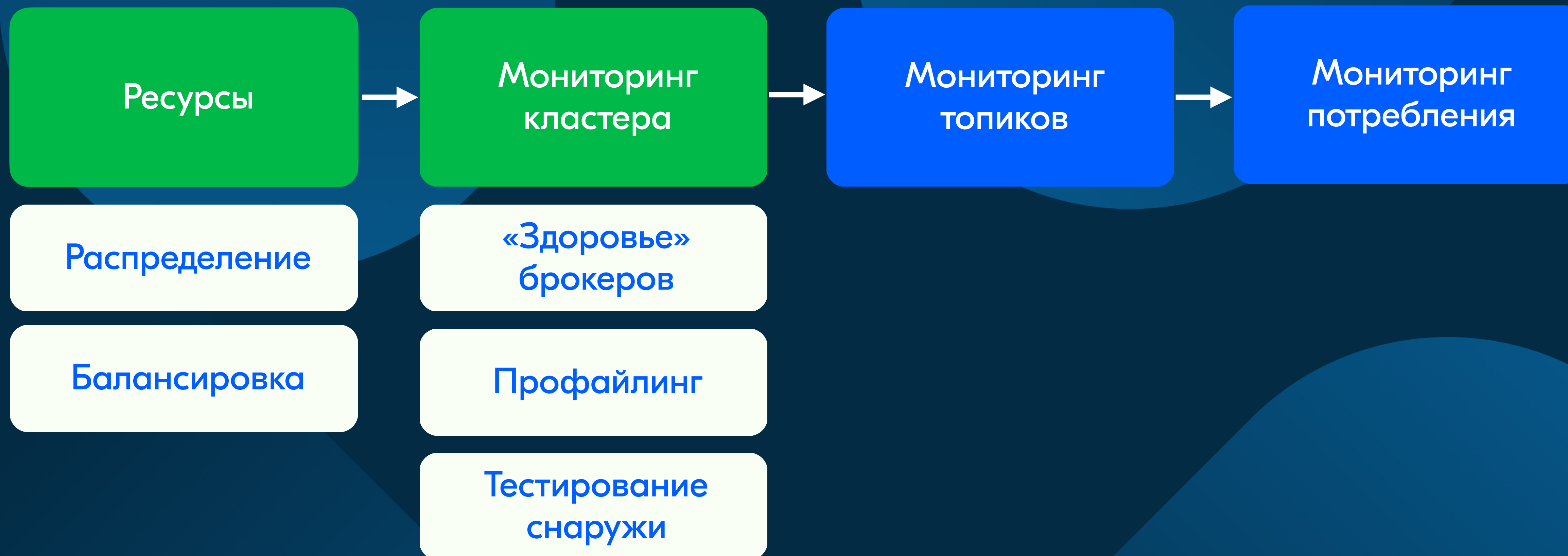


Metadata



Consume





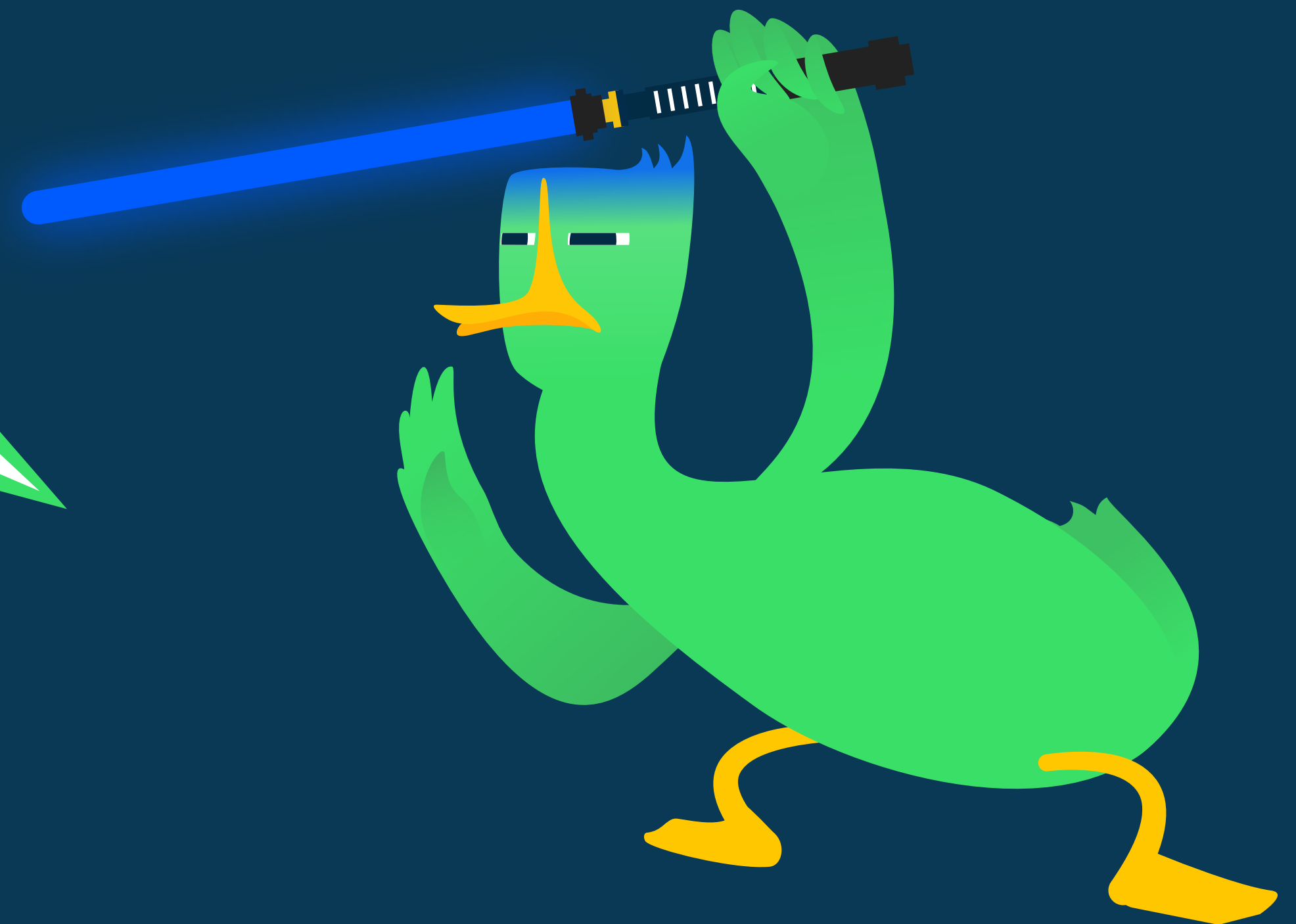
Мониторинг топики

Базовые алерты

Базовые алерты

Продуктовые команды должны знать, если у них что-то идет не так

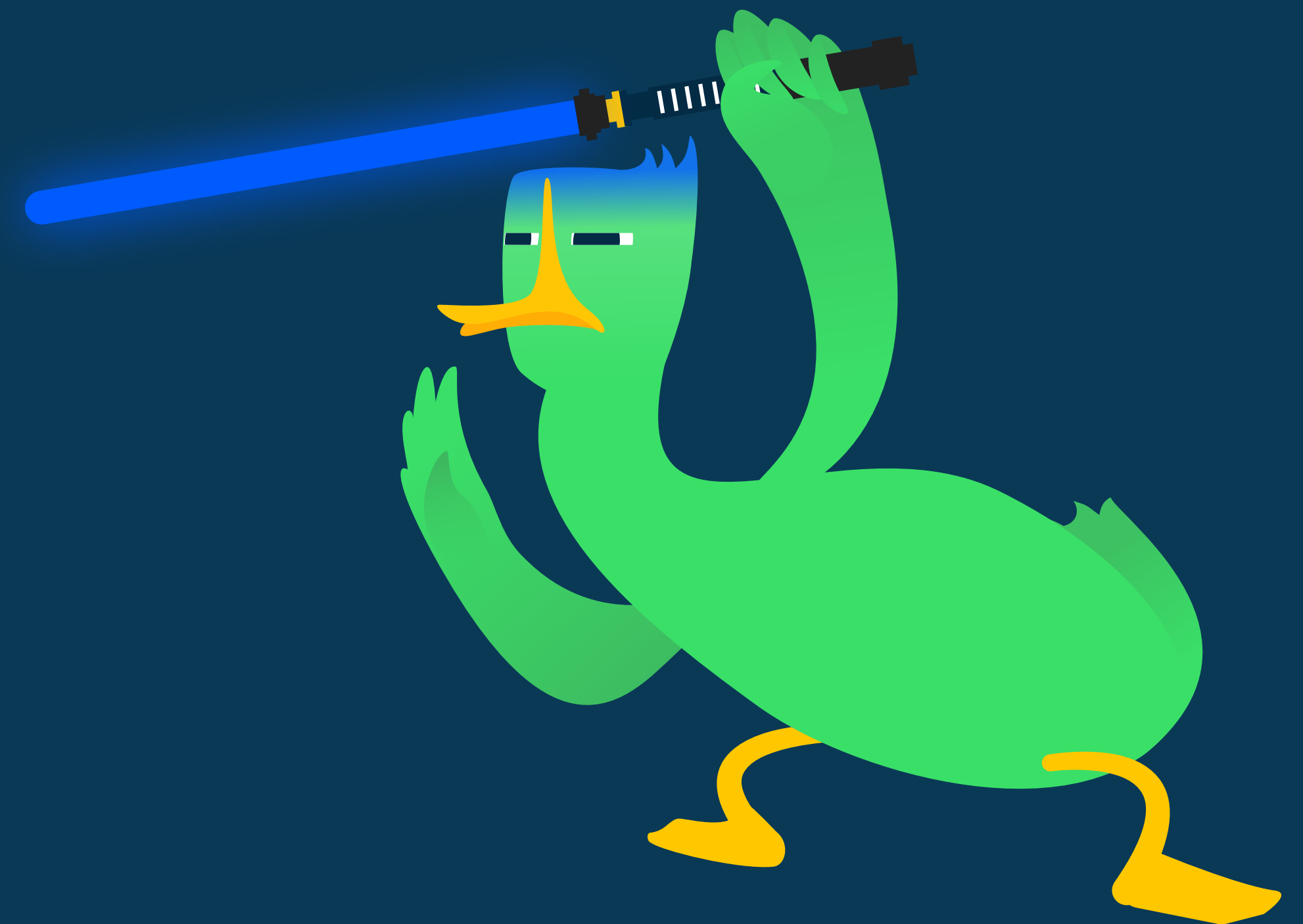
Алерты получает
любой владелец
топика «из коробки»

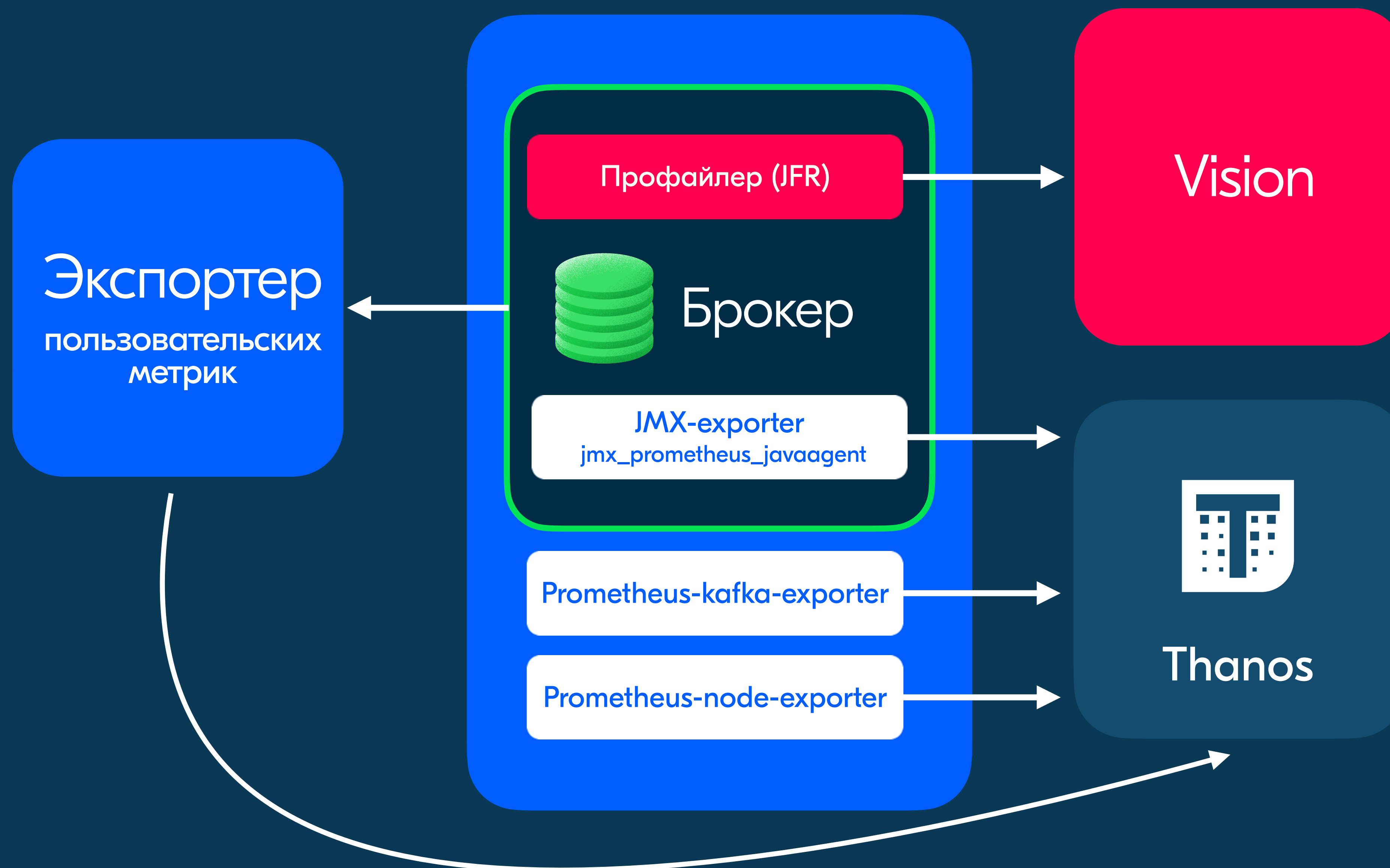


Базовые алерты

Их получает любой владелец топика «из коробки»

- Лаг консьюмера в записях составляет более X записей
- Топик пуст более X дней
- Консьюмер-группа потребляет не все партиции топика
- У топика нет консьюмеров





Экспортер собирает и агрегирует

Экспортер собирает и агрегирует

ozon{tech

Лаги консьюмеров

Экспортер собирает и агрегирует

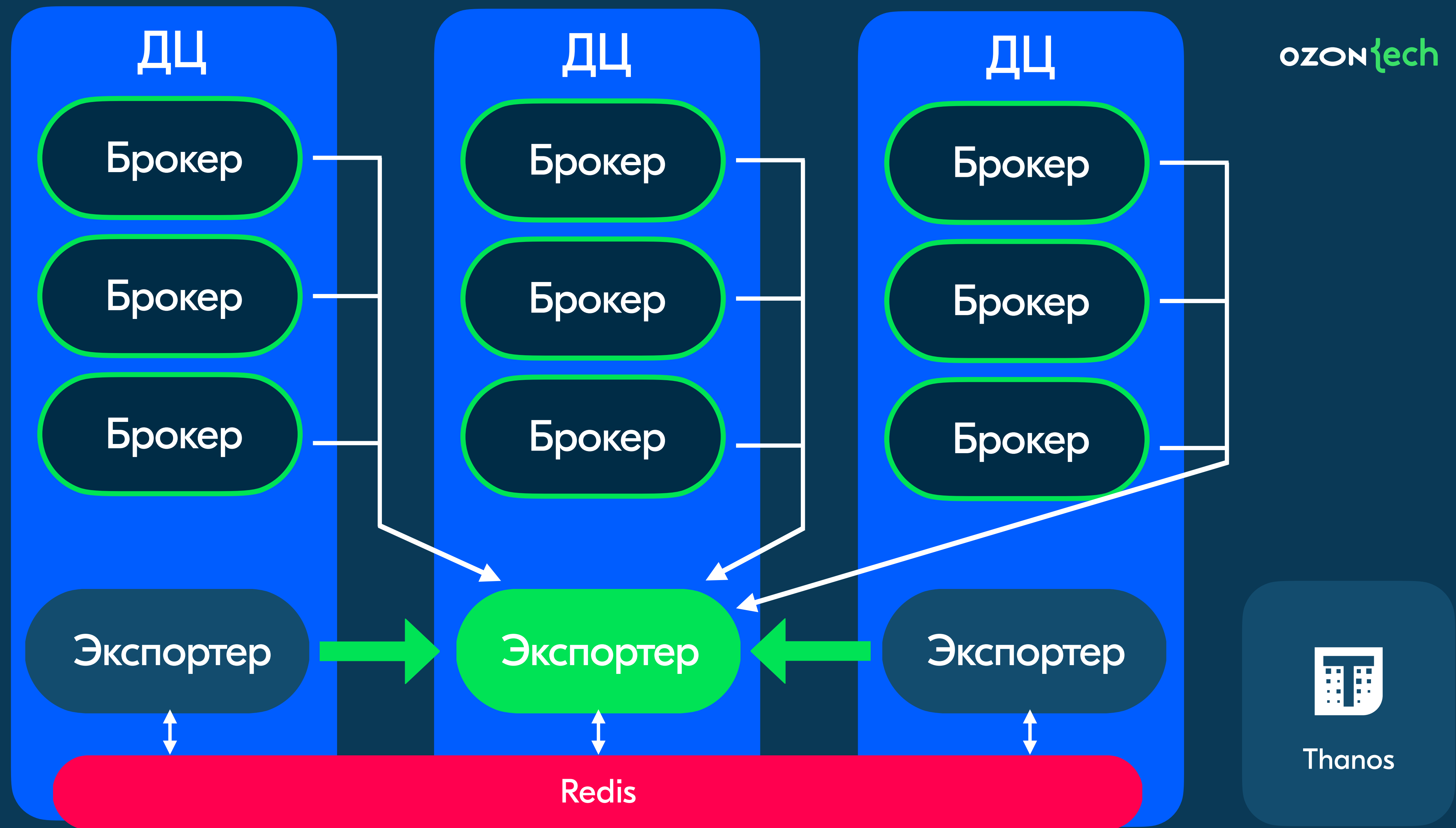
Лаги консьюмеров

Log size

Лаги консьюмеров

Log size

Связи консьюмер-группа — топик





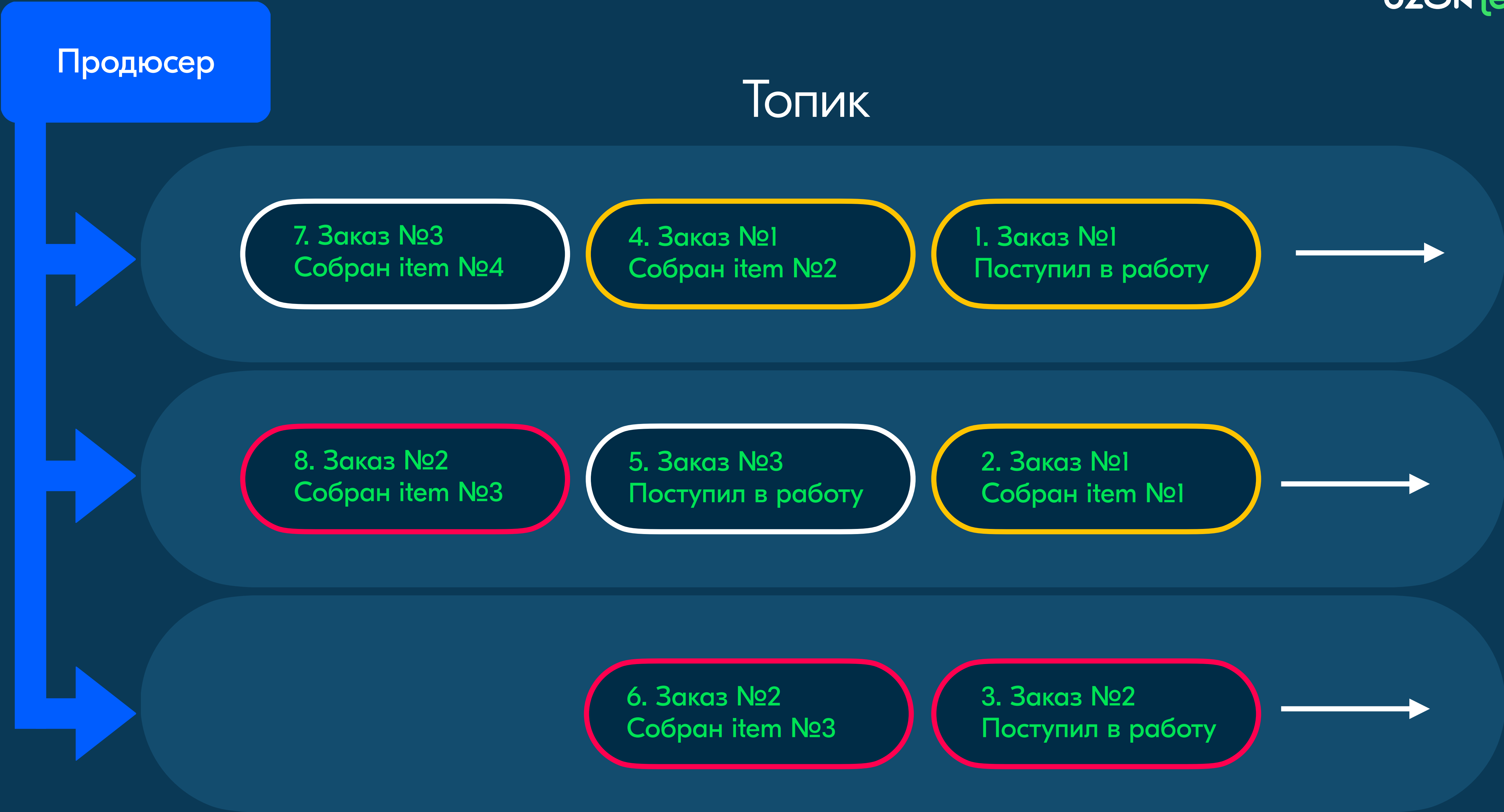
Мониторинг топики

Неравномерная нагрузка на партиции топика

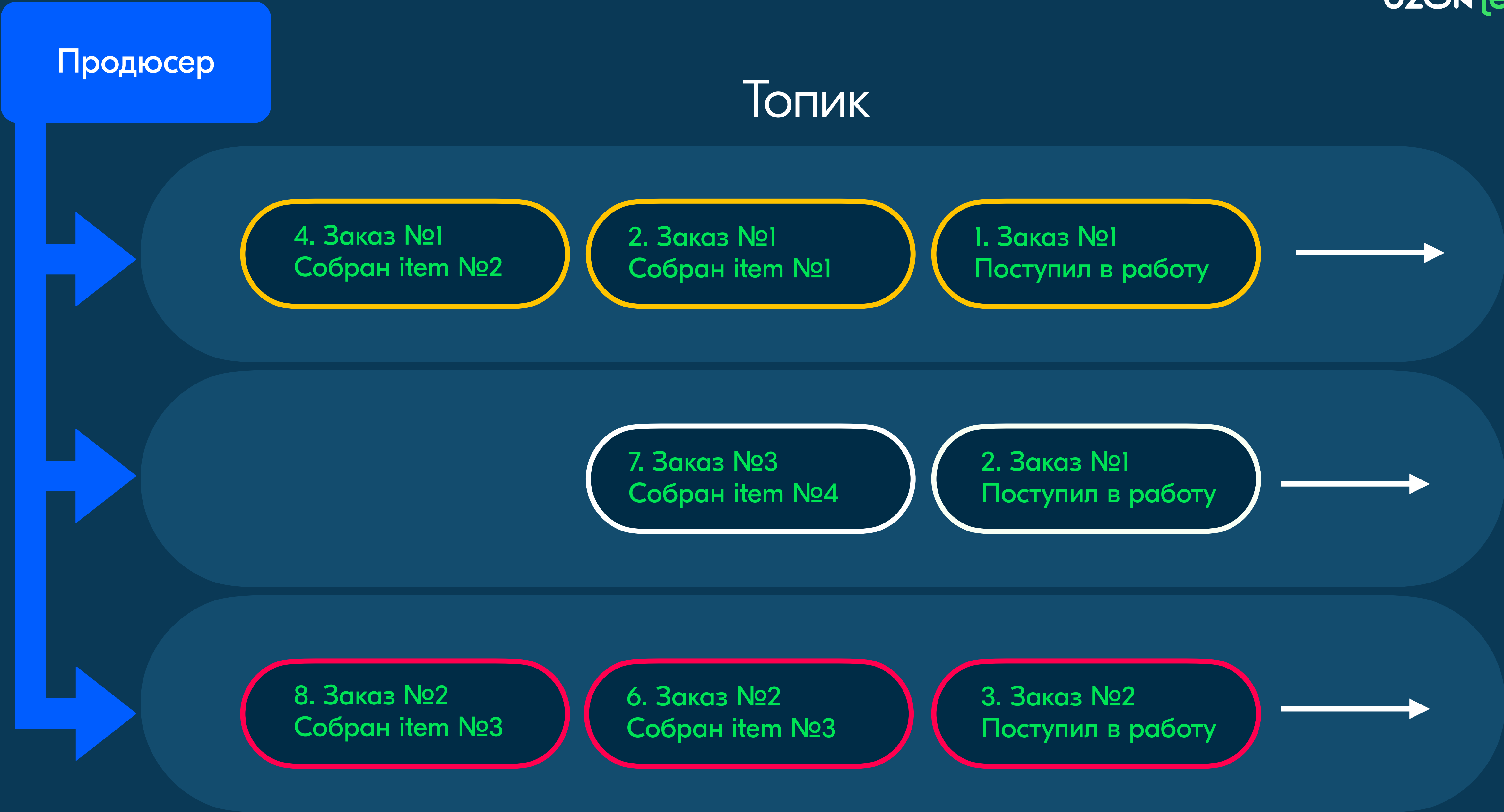
Неравномерная нагрузка на партии топики



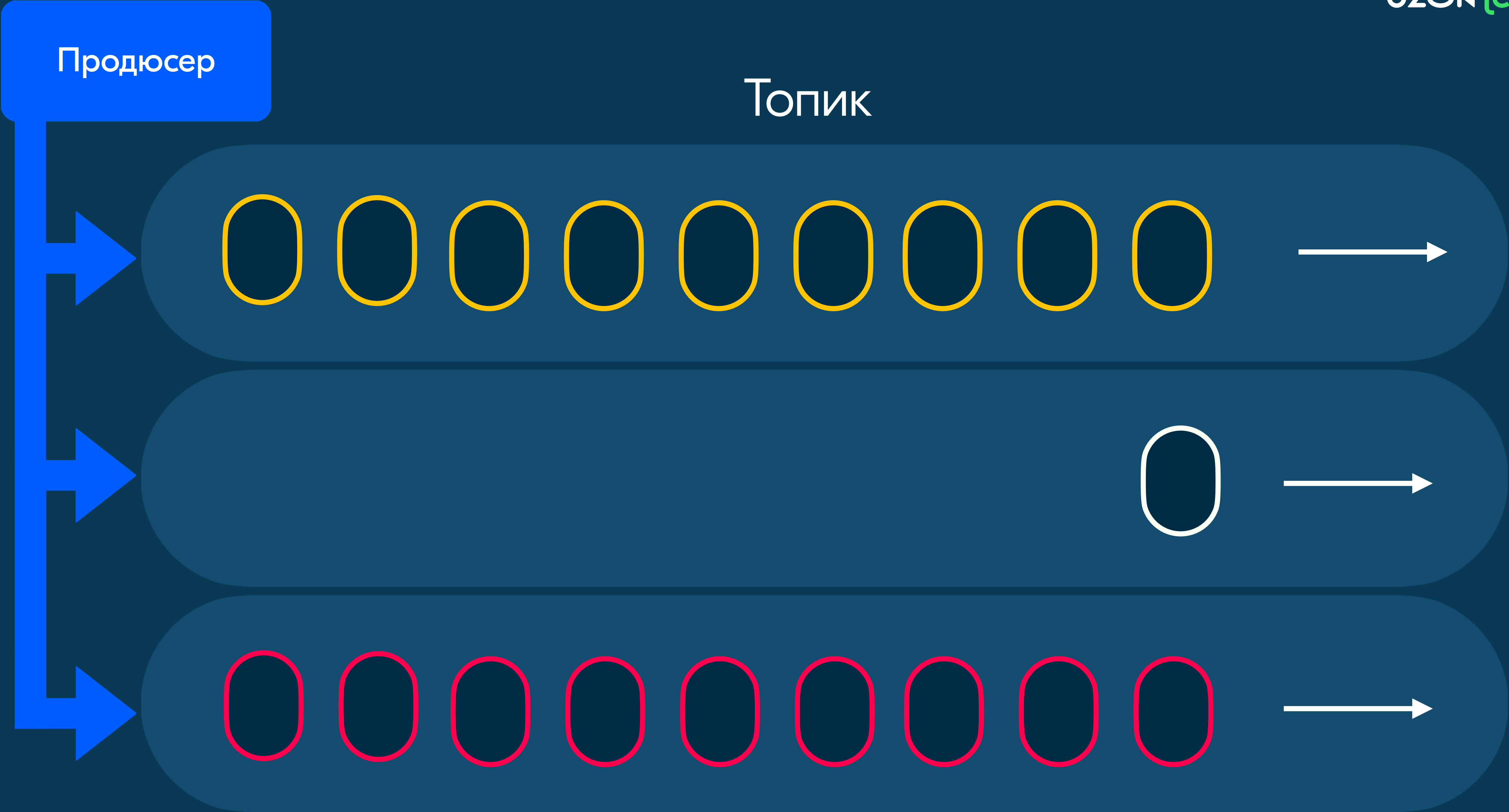
Неравномерная нагрузка на партии топики



Неравномерная нагрузка на партии топики



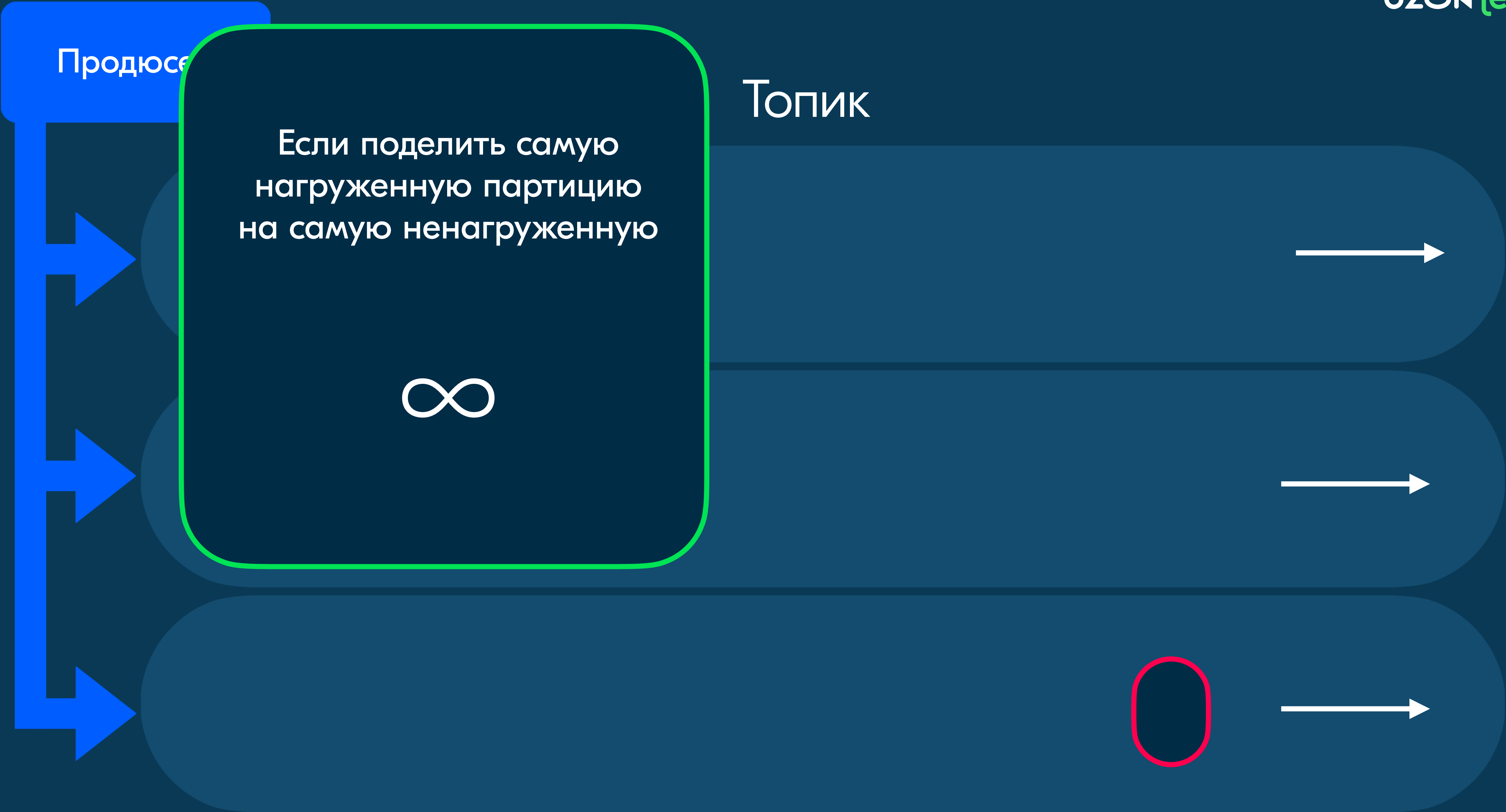
Неравномерная нагрузка на партии топики



Неравномерная нагрузка на партии топика

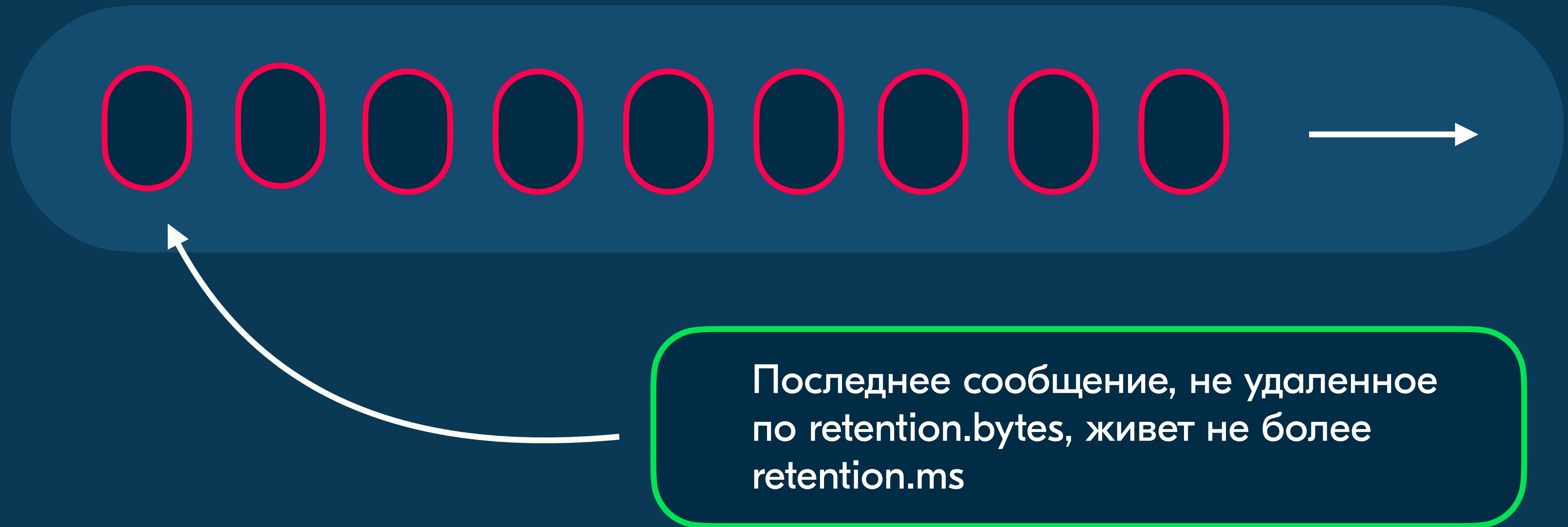


Неравномерная нагрузка на партии топика



Нагруженные топики

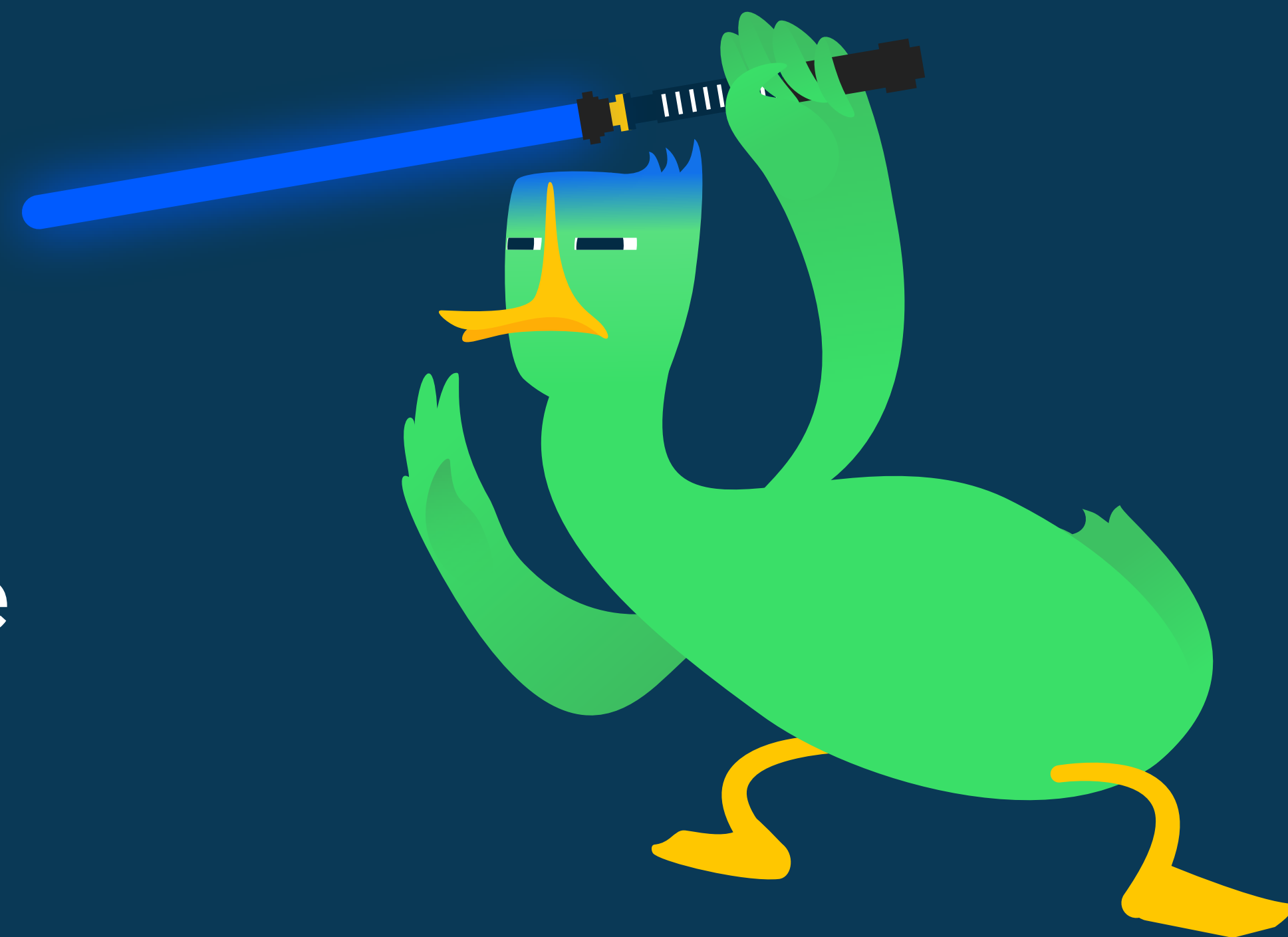
Вводим понятие



Базовые алерты

Их получает любой владелец НАГРУЖЕННОГО топика «из коробки»

- Отношение нагрузки максимально нагруженной партиции к минимально нагруженной партиции за последний час
- Нагрузка на топик более X Гб/партиция/день
- Сообщение в топике хранится менее X часов





Мониторинг потребителей

Не все умеют «готовить» Кафку, и это нормально

Потребители могут вести себя странно

Requests Metadata



**Мы можем смотреть запросы
по брокеру/топику,
но не можем по потребителю**

Теми средствами, которые обсуждали выше

Как получить метрики per IP/service?

Собрать с клиентов

Как получить метрики per IP/service?

Собрать с клиентов

- Большой зоопарк клиентов: Go, C#, Java, Python

Как получить метрики per IP/service?

Собрать с клиентов

- Большой зоопарк клиентов: Go, C#, Java, Python
- Большое количество клиентов — даже простое обновление займет месяцы

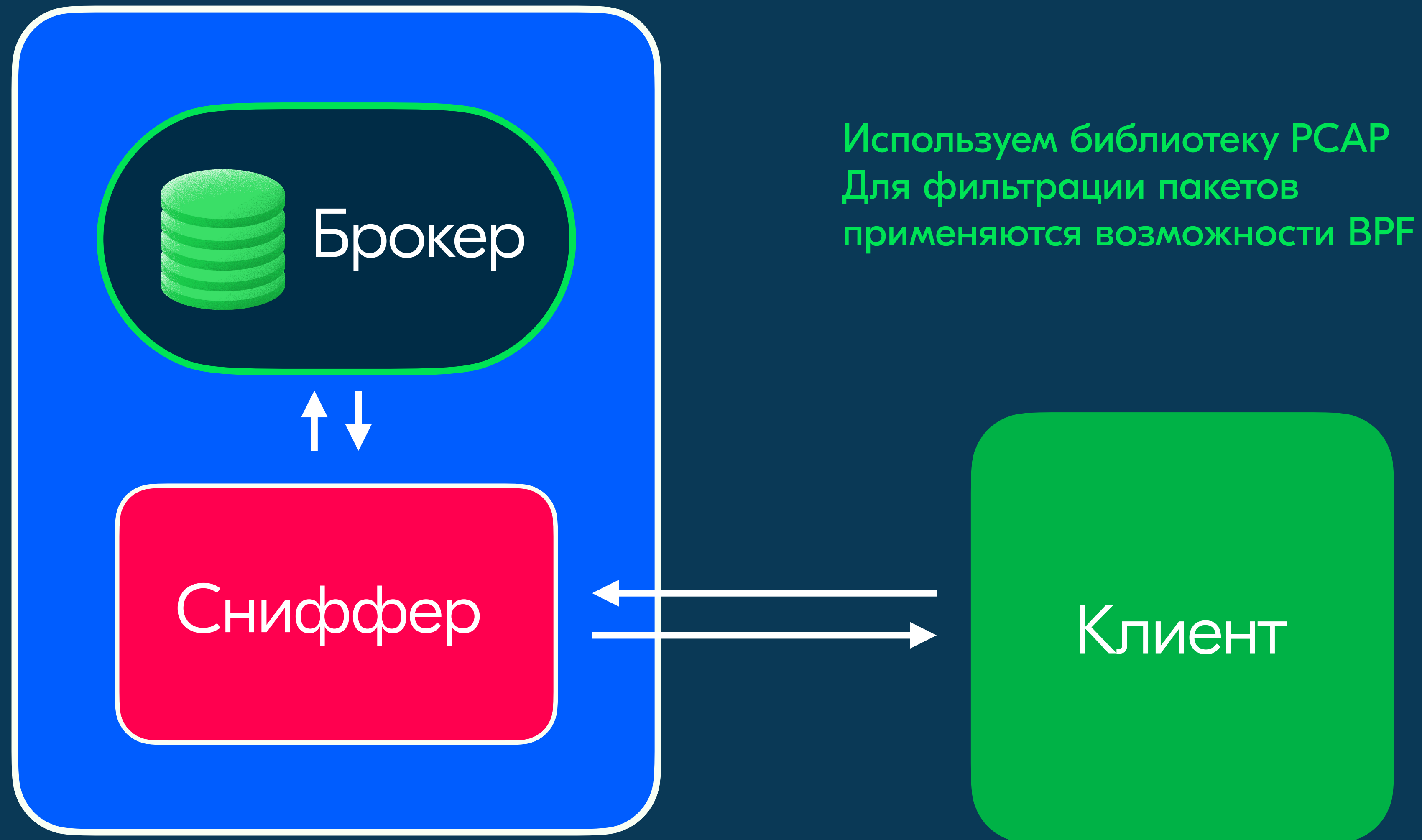
Как получить метрики per IP/service?

Собрать с клиентов

- Большой зоопарк клиентов: Go, C#, Java, Python
- Большое количество клиентов — даже простое обновление займет месяцы
- Нет способа убедиться, что метрики собираются корректно

Как получить метрики per IP/service?

Сниффер



Как получить метрики per IP/service?

Сниффер

- Не справляется с нашими объемами

Как получить метрики per IP/service?

Сниффер

- Не справляется с нашими объемами



Есть зафиксированные
проблемы
с производительностью

Есть теоретическая
ВОЗМОЖНОСТЬ СОСТОЯНИЯ
ГОНКИ ПАКЕТОВ

Как получить метрики per IP/service?

Сниффер

- Не справляется с нашими объемами
- Не работает с SSL

Как получить метрики per IP/service?

Сниффер

- Не справляется с нашими объемами
- Не работает с SSL



Собрать-то пакеты можно,
а вот прочитать нельзя :(

Как получить метрики per IP/service?

Сниффер

- Не справляется с нашими объемами
- Не работает с SSL
- **eBPF-проба?**

Как получить метрики per IP/service?

Сниффер

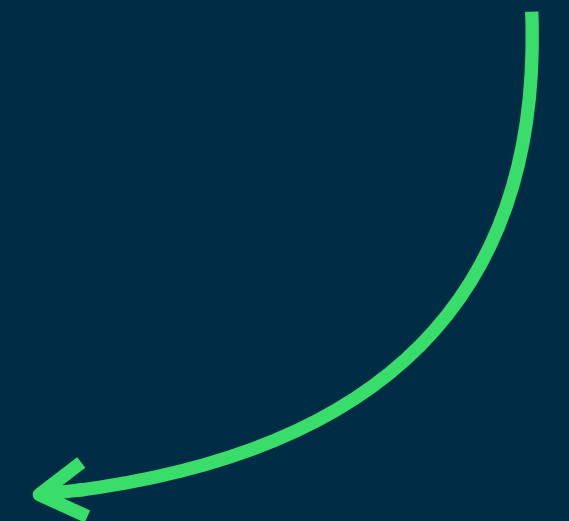
- Не справляется с нашими объемами
- Не работает с SSL
- **eBPF-проба?**



Не так-то просто залезть в JVM :(
Но вы можете попробовать:

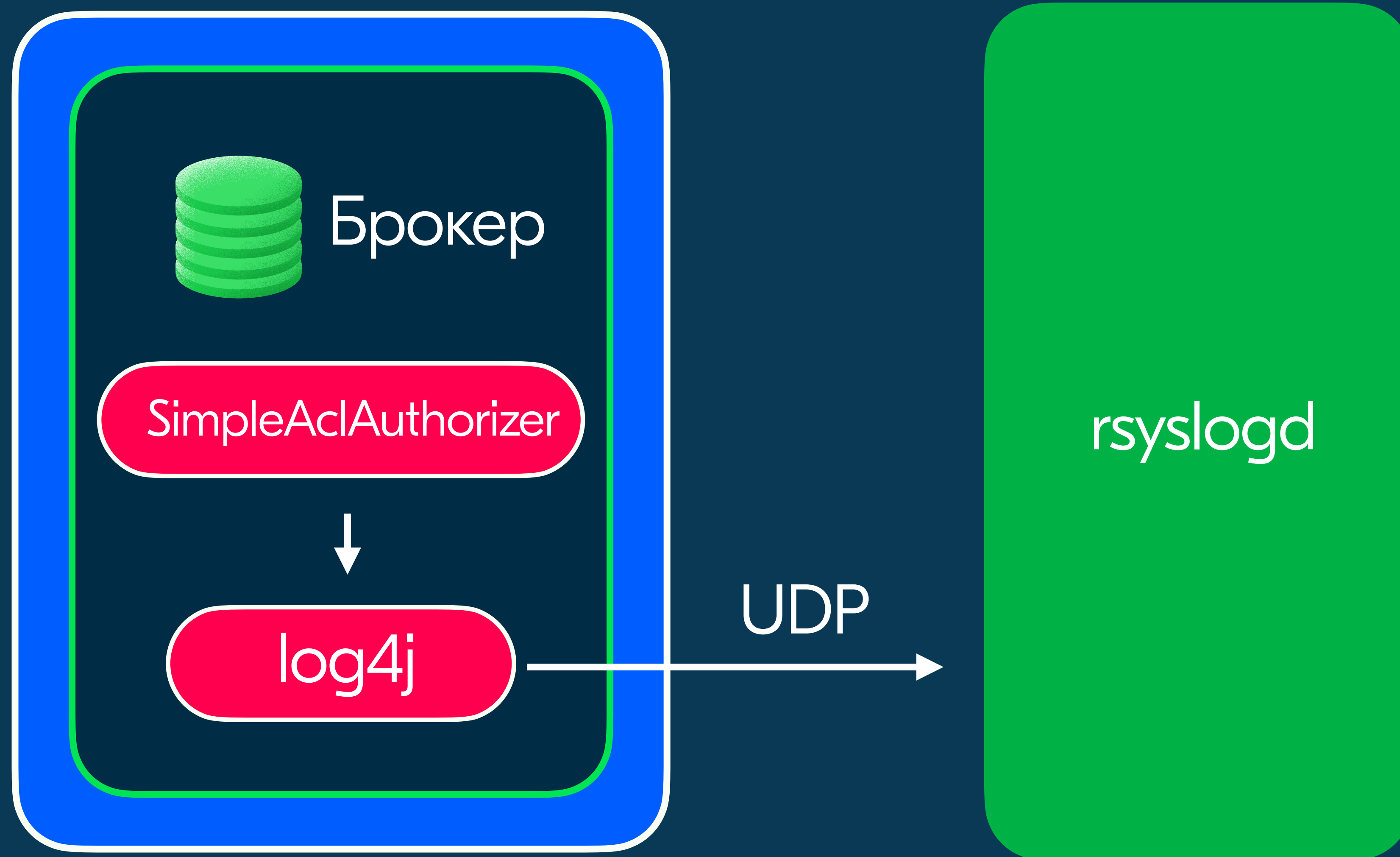


Полезные
материалы
про eBPF




Как получить метрики per IP/service?

Через логи



Как получить метрики per IP/service?

Через логи



Отправка по udp
занимала много
времени и была,
предположительно,
блокирующей, это
привело к замедлению
в обработке запросов

ogd



- Собрать с клиентов
- Сниффать трафик
- Включать логи



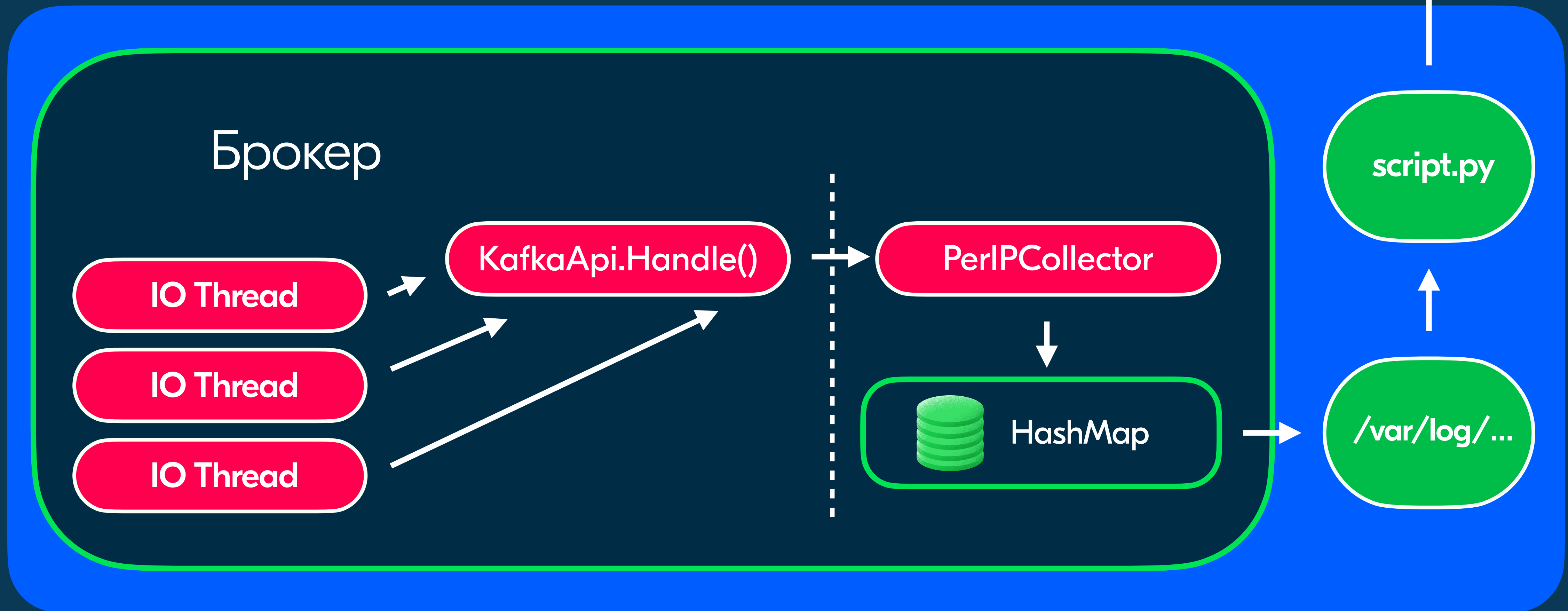
- Собрать с клиентов
- Сниффать трафик
- Включать логи



- Форкнуть Кафку

Как получить метрики per IP/service?

Форкнуть Кафку :)



Заключение



«Прожорливость» средств observability

На нашем профиле и наших объемах

	На брокерах	Снаружи брокера
Экспортеры на брокере	3-4 ядра/брокер	—
Экспортер пользовательских метрик	<1 ядро/брокер	Под в кубере, ~0,5 ядра, 4Гб
Профайлинг	<1 ядро/брокер	Место в s3
Дополнительный механизм в форке	<1 ядро/брокер	—
Модельный сервис	Незначительно	Под в кубере, ~0,5 ядра

Обзор инфраструктуры вокруг больших Kafka-кластеров в Ozon

Мы поговорили

1. Ресурсы кластера
2. Мониторинг серверов/брокеров
3. Мониторинг топиков
4. Мониторинг потребителей

Обзор инфраструктуры вокруг больших Kafka-кластеров в Ozon

Мы поговорили

1. Ресурсы кластера
2. Мониторинг серверов/брокеров
3. Мониторинг топиков
4. Мониторинг потребителей

Осталось за рамками

1. Управление кластером
2. Ограничения потребителей
3. Авторизация
4. Типизация сообщений в топиках
5. ...



Спасибо за внимание

Виктор Корейша,
руководитель направления Managed Services, Ozon

viktor@koreysha.ru

