Deckhouse

От алерта к причине за 10 минут

Как ускорить диагностику инцидентов с Deckhouse Observability Platform

Deckhouse

Владимир Гурьянов

Технический директор продукта
Deckhouse Observability Platform

Чем занимаюсь:

Руковожу разработкой направления
Observability в Deckhouse

Опыт:

- С 2011 занимаюсь эксплуатацией приложений
- С 2021 работаю в компании «Флант»
- С 2023 занимаюсь развитием направления Observability

ФЛАНТ

Синергия опыта вендора, интегратора,
сервисной и консалтинговой компании

Deckhouse

Deckhouse — продуктовое подразделение, разработчик продуктов для построения надёжной enterprise-инфраструктуры

DaaS

DaaS — комплексное DevOps-сопровождение инфраструктуры в режиме 24/7 силами выделенной DevOps-команды

Express 42

«Экспресс 42» — DevOps-консалтинг. От анализа узких мест в ИТ-процессах до создания роадмапа изменения ИТ для реализации цифровой трансформации

Для кого

- **Разработчики** — поведение своих сервисов: производительность, ошибки, путь запроса
- **Platform-команды, SRE, DevOps, NOC** — работоспособность сервисов и расследование инцидентов
- **CIO/CTO** — доступность цифровых сервисов, SLA, ИТ-бюджет
- **Менеджеры по мощностям и ИТ-активам** — загрузка ресурсов, учёт оборудования, закупки

Скачать краткий обзор DOP в PDF

План вебинара

01 Расследование инцидентов

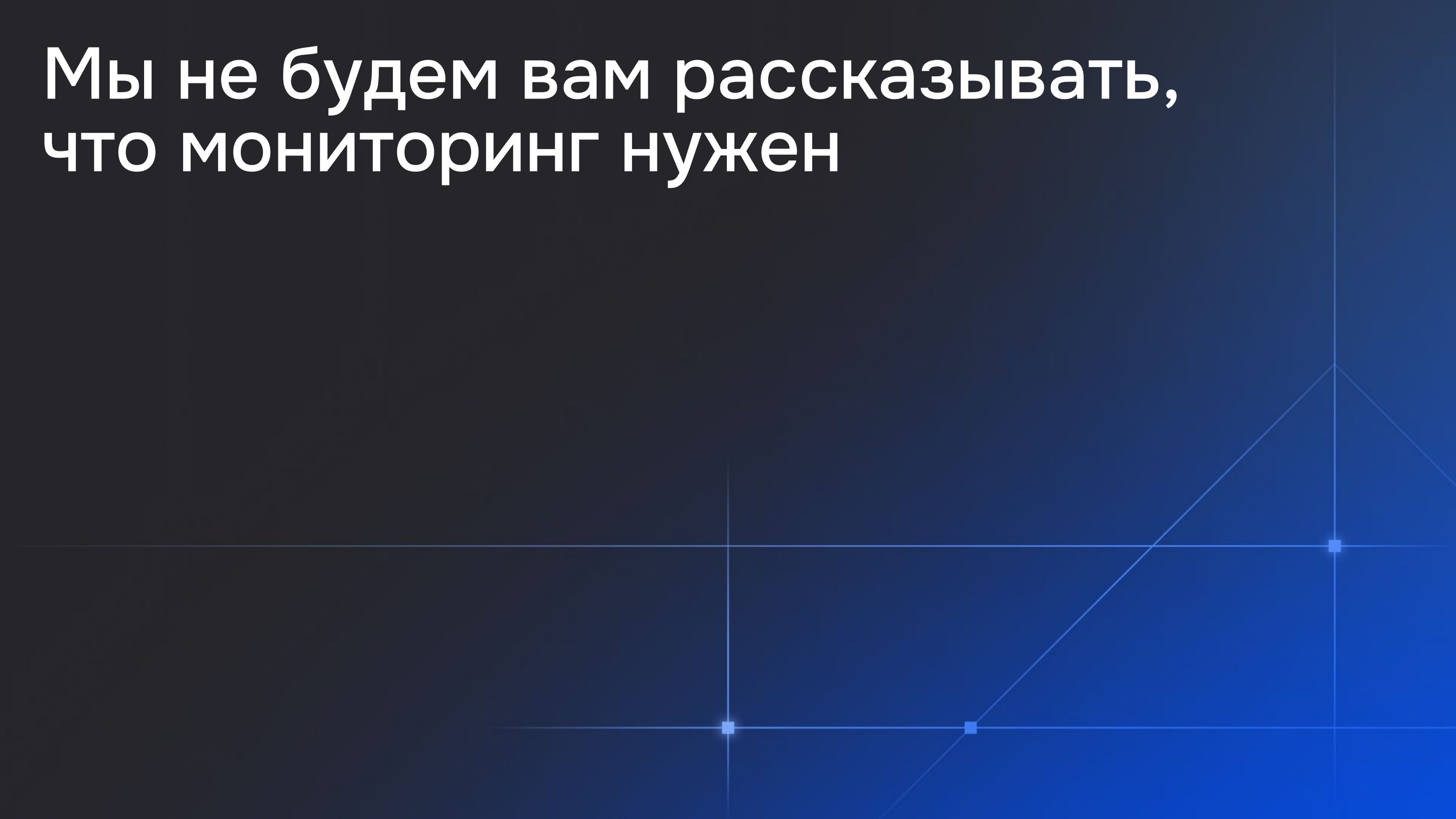
02 Deckhouse Observability Platform

03 Планы по развитию

04 Live-демо

05 Q&A-секция

Мы не будем вам рассказывать, что мониторинг нужен

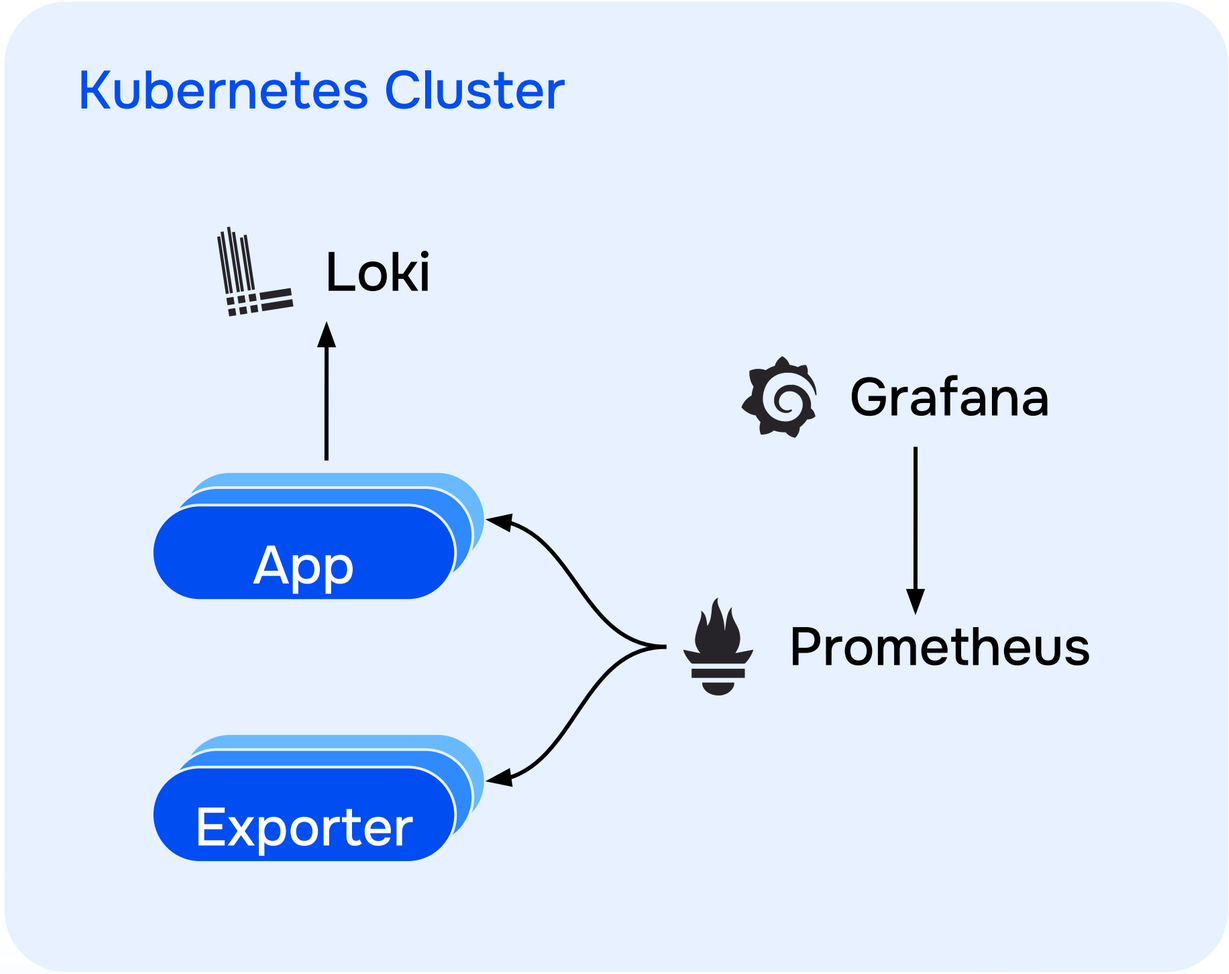


Вебинар observability в DKP

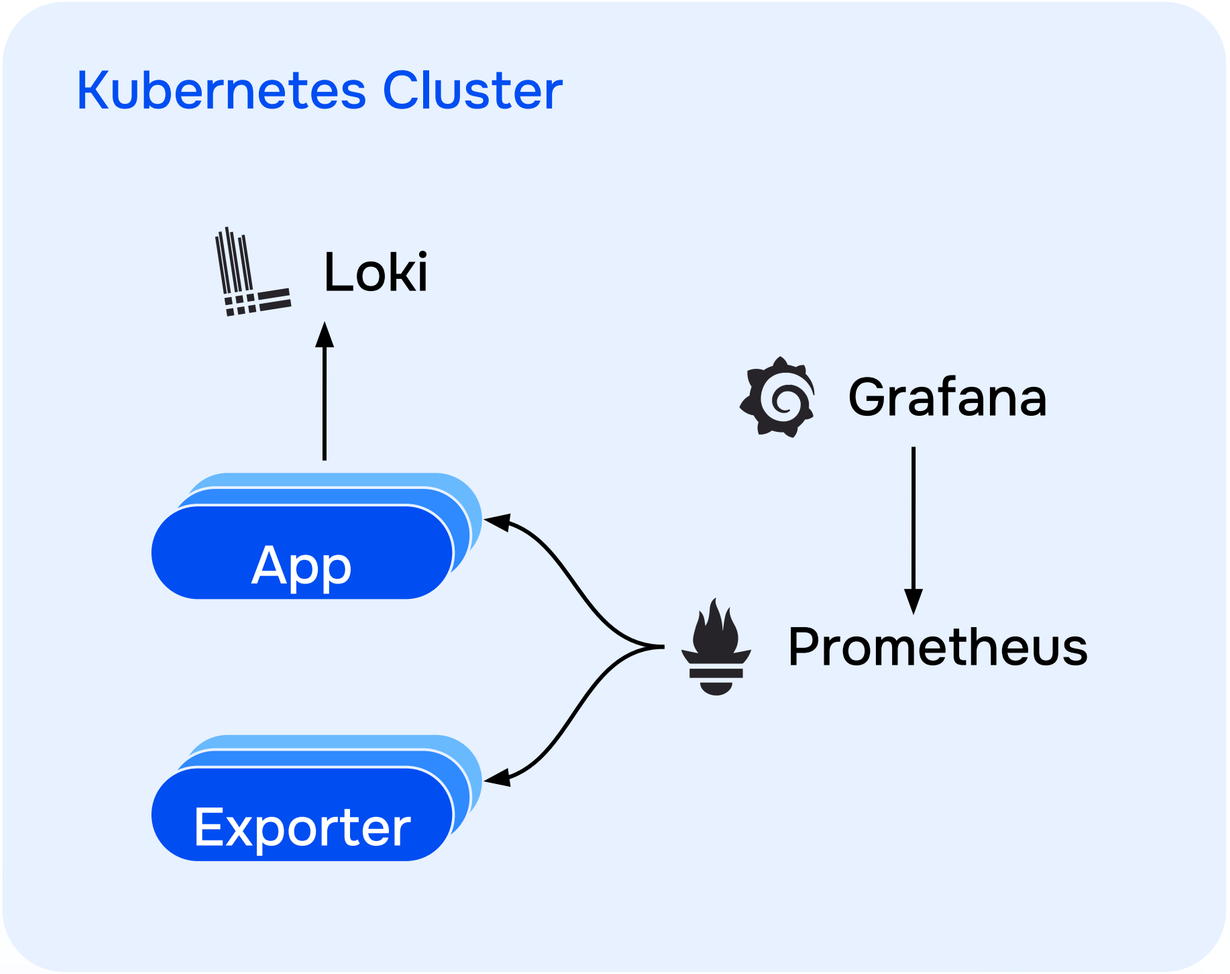
Смотреть

- Зачем нужен мониторинг
- Как устроен мониторинг Kubernetes
- Как собирать метрики в Kubernetes
- Как визуализировать собранные метрики
- Как настроить отправку уведомлений

В итоге получилась такая схема



Хорошо работает



Ho...

Есть разные компоненты вашей инфраструктуры

Хранилища

Гипервизоры

Системное ПО

Kubernetes-кластеры

Приложения

Сетевое оборудование

SNMP

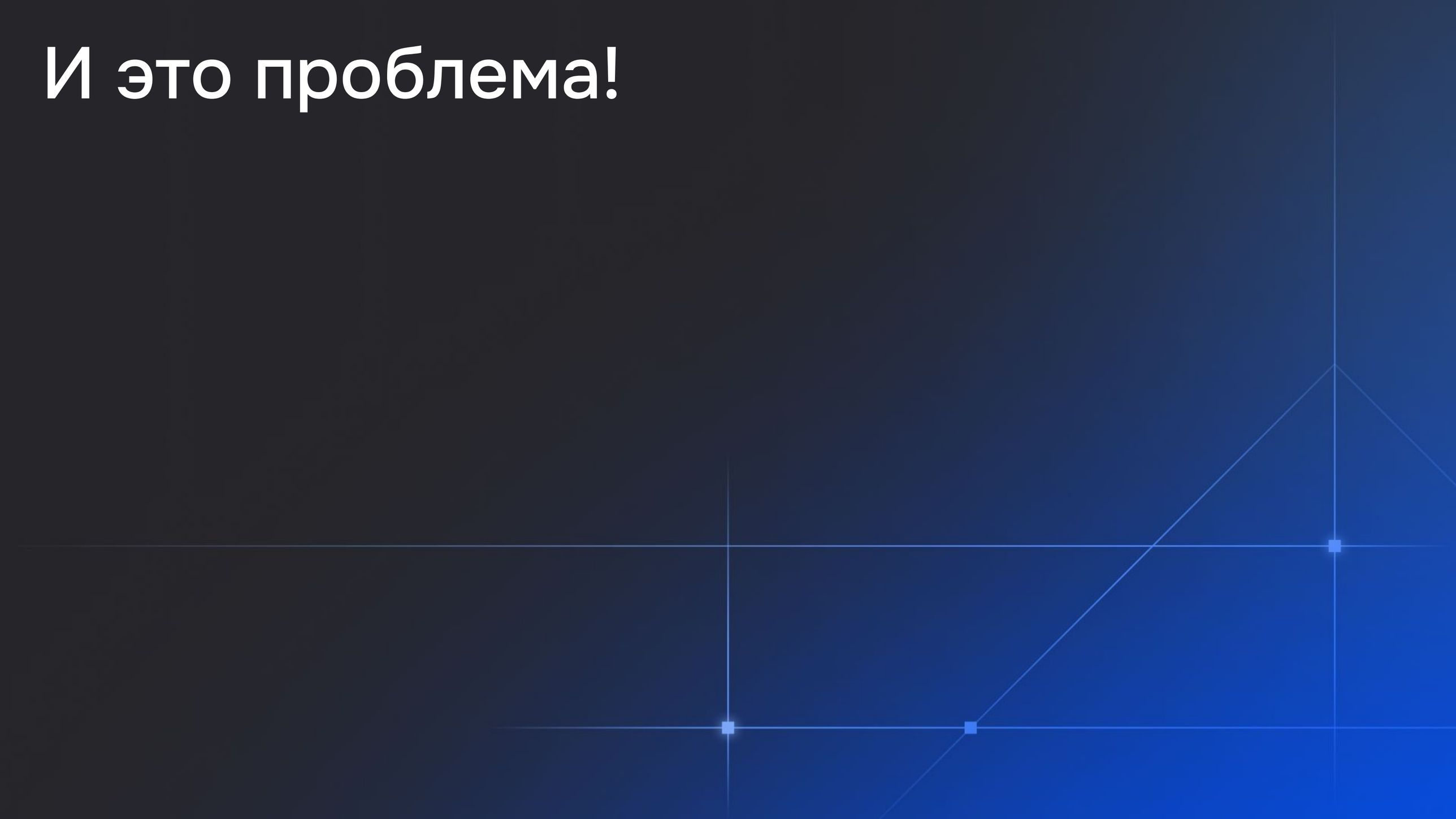
BM

Железо

Windows

Linux

И это проблема!



Пример: как выглядит расследование инцидента?

Пример

01 Получаем алерт, что время обработки части запросов увеличилось до 2 сек

02 Идём в систему мониторинга Kubernetes Cluster

03 Видим, что приложения в одном из кластеров медленно отвечают

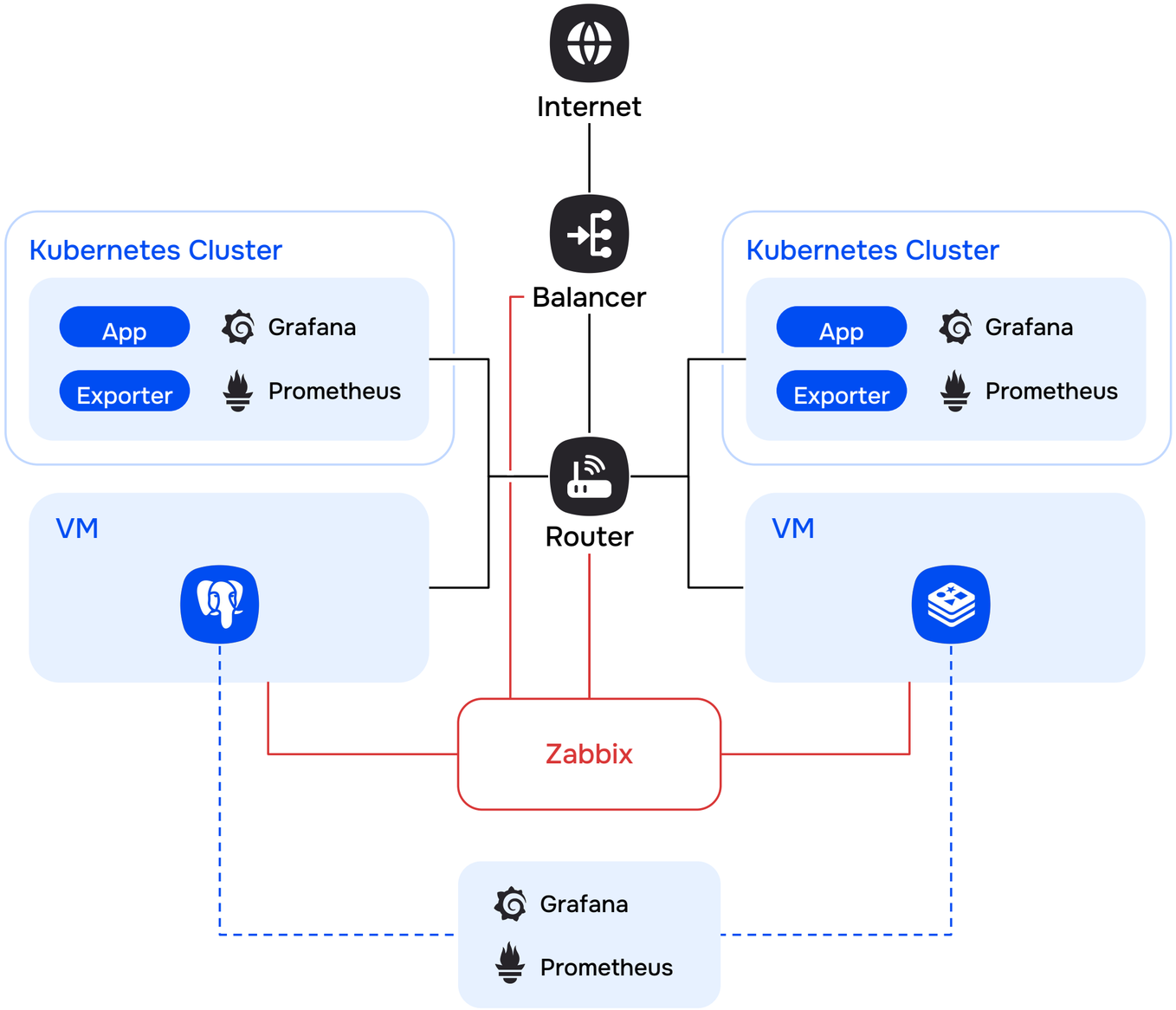
04 Смотрим логи приложения — ошибок нет

05 Смотрим метрики связанных компонентов

06 Видим возросшую нагрузку на БД

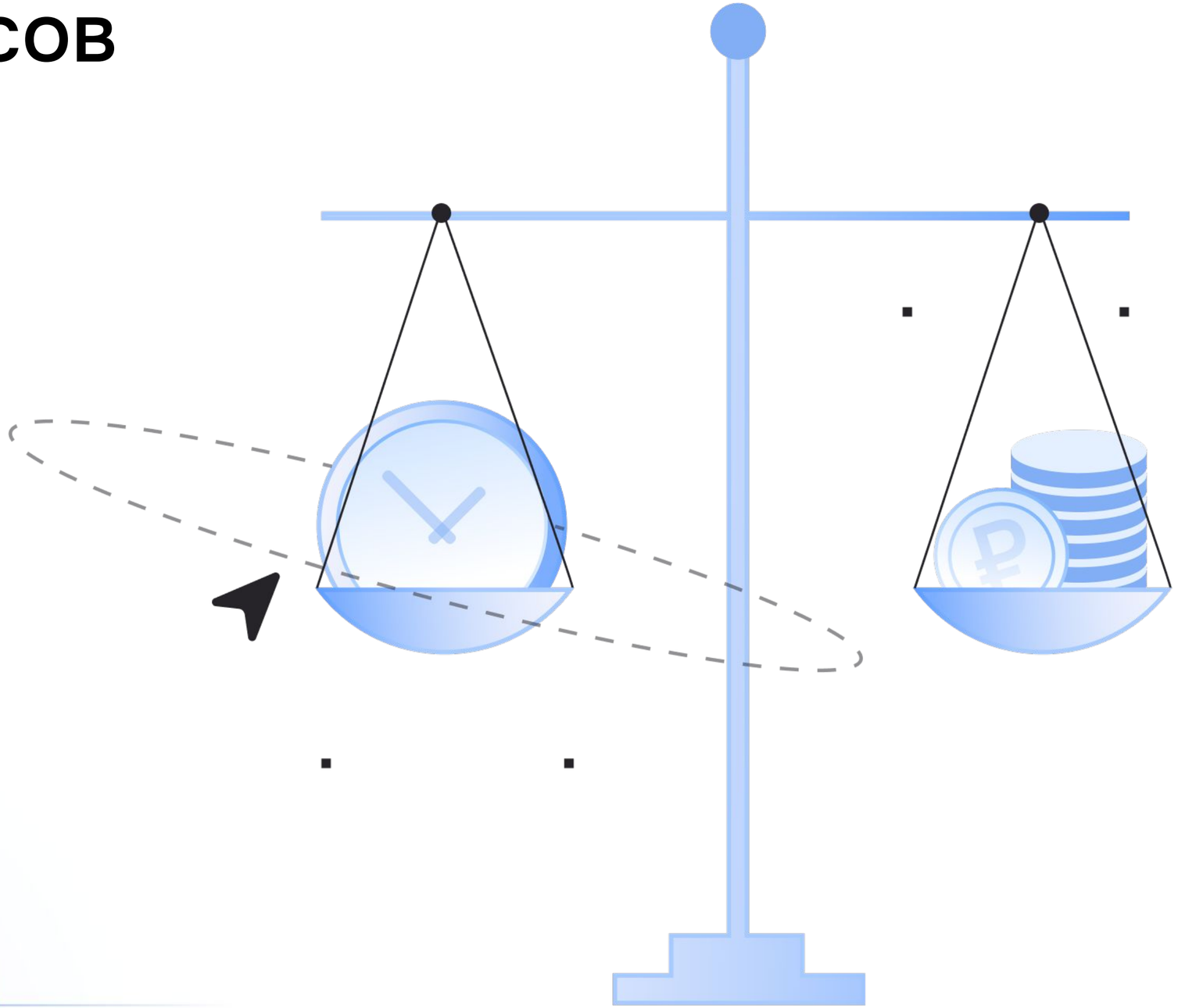
07 По логам сервера видим, что есть ошибки диска

08 В итоге: один из SSD вышел из строя

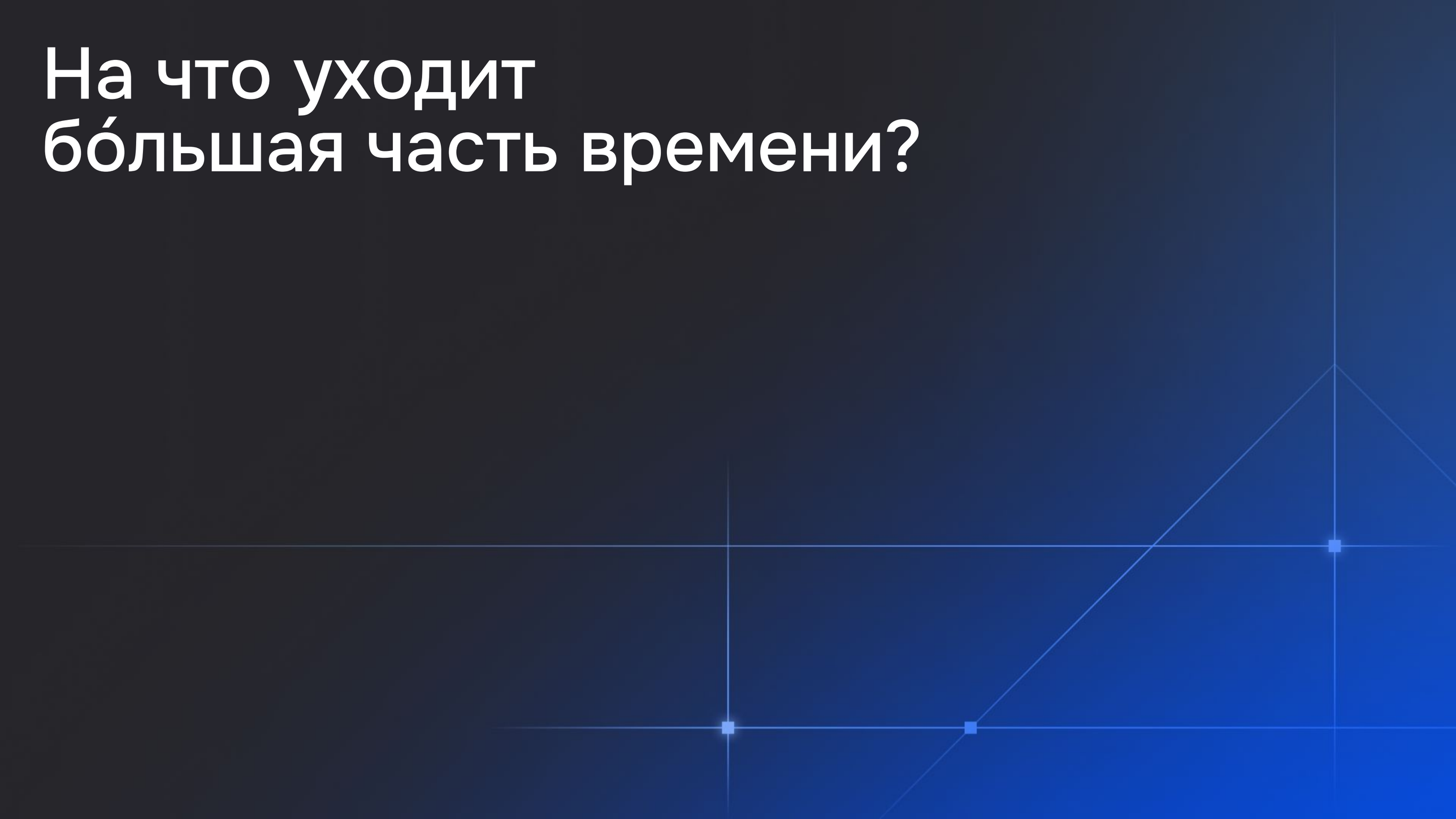


От 1 часа до 3 часов

в среднем уходит на расследование
такого инцидента



На что уходит бóльшая часть времени?



01

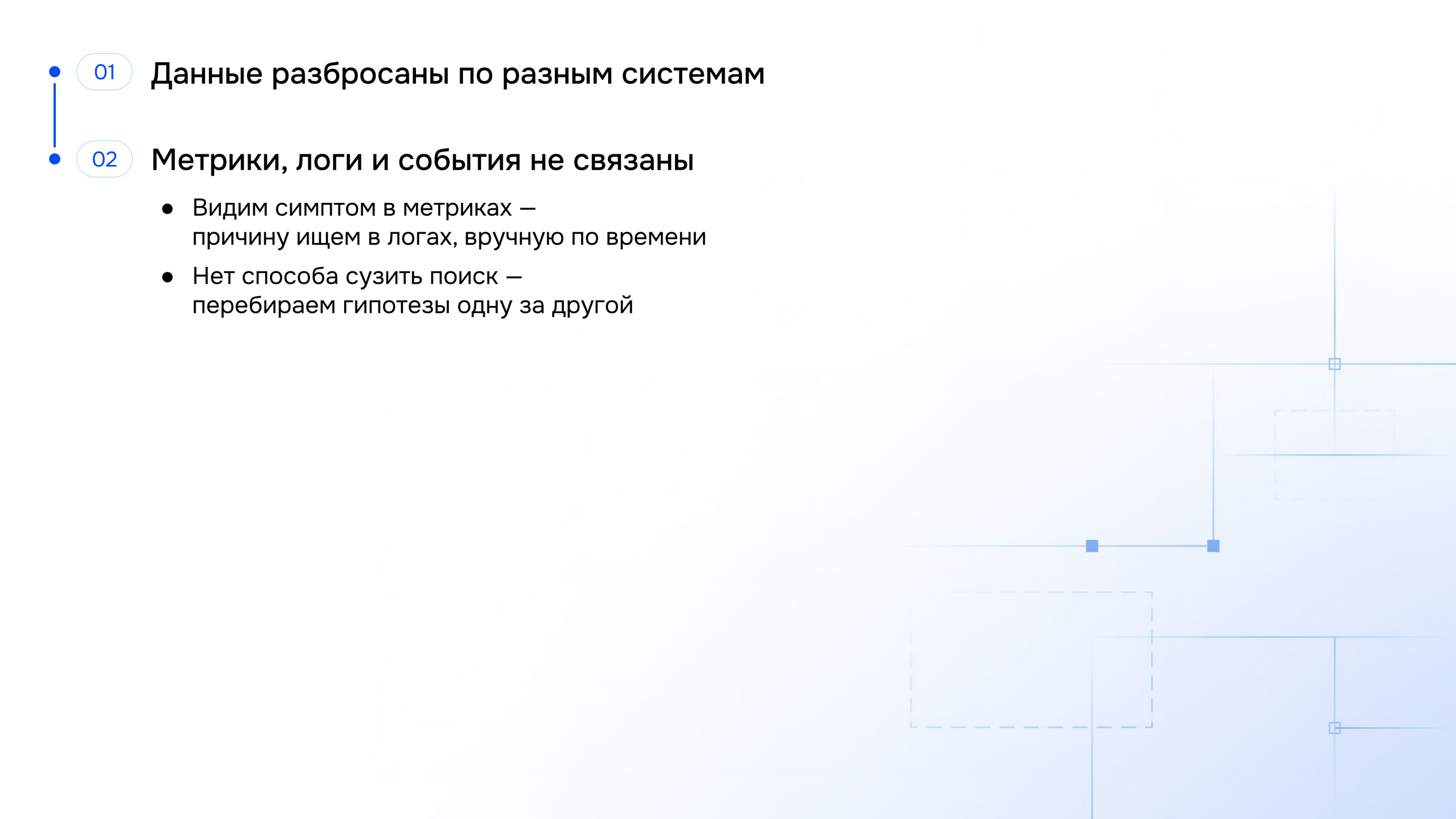
Данные разбросаны по разным системам

- Zabbix, Prometheus, логи —
  везде свой интерфейс и свой поиск
- Время, имена метрик и лейблы
  не совпадают — сопоставляем вручную
- К части систем у инженера может
  не быть доступа — ждём коллегу

01 Данные разбросаны по разным системам

02 Метрики, логи и события не связаны

- Видим симптом в метриках —
  причину ищем в логах, вручную по времени
- Нет способа сузить поиск —
  перебираем гипотезы одну за другой



01 Данные разбросаны по разным системам

02 Метрики, логи и события не связаны

03 Части данных нет — идём на сервер

- top, dmesg, логи — смотрим руками
- Видим состояние «сейчас»,
  а не в момент инцидента

01 Данные разбросаны по разным системам

02 Метрики, логи и события не связаны

03 Части данных нет — идём на сервер

Чтобы не тратить время
на диагностику и сбор
контекста по разным системам

Нужна система, которая
позволяет собрать метрики,
логи и трейсы в одном месте

И это...

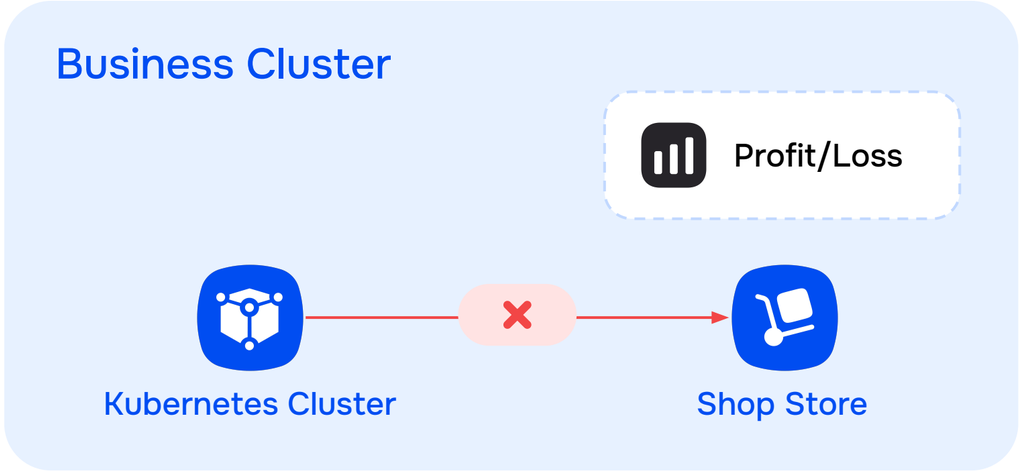
Deckhouse Observability Platform

Централизованное решение наблюдаемости
для всей инфраструктуры и приложений

Кейс: DOP в крупном холдинге

Исходные данные (до внедрения)

Показатель	Значение
Критичных инцидентов в месяц	50
Среднее время восстановления (MTTR)	4 часа
Стоимость часа простоя бизнес-сервиса	1,5 млн ₽
Инженеров в группе поддержки	6
Стоимость часа работы инженера	6000 ₽
Время ручного расследования инцидента	2,5 часа



Ежемесячные потери до

Простои	50 × 4 × 1,5 млн = 300 млн ₽
Трудозатраты	50 × 6 × 6000 × 2,5 = 4,5 млн ₽
Итого потерь в месяц	**304,5 млн ₽**

Кейс: DOP в крупном холдинге

Что дало внедрение DOP

снизилось

На 20 %

количество инцидентов

снизилось

На 45 %

трудозатраты на расследование

снизилось

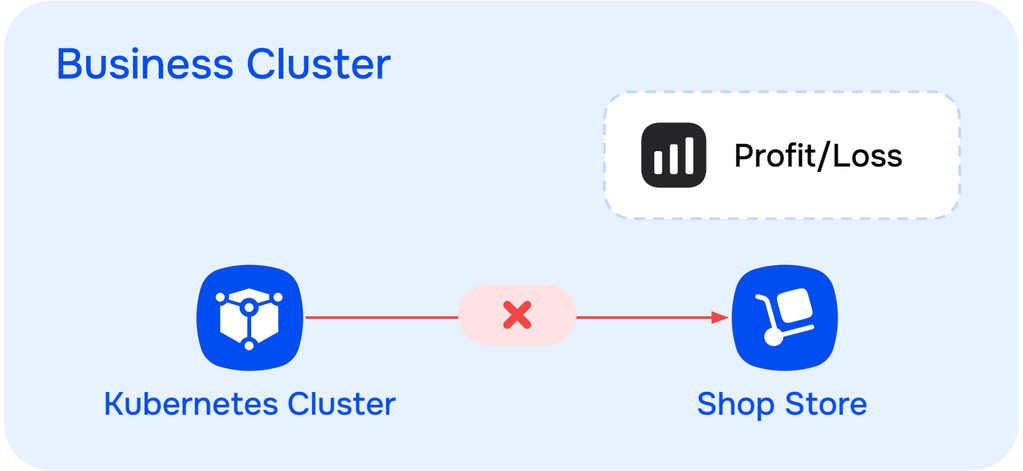
MTTR на 95 %

с 4 ч до ~12 мин

повысилось

До 95 %

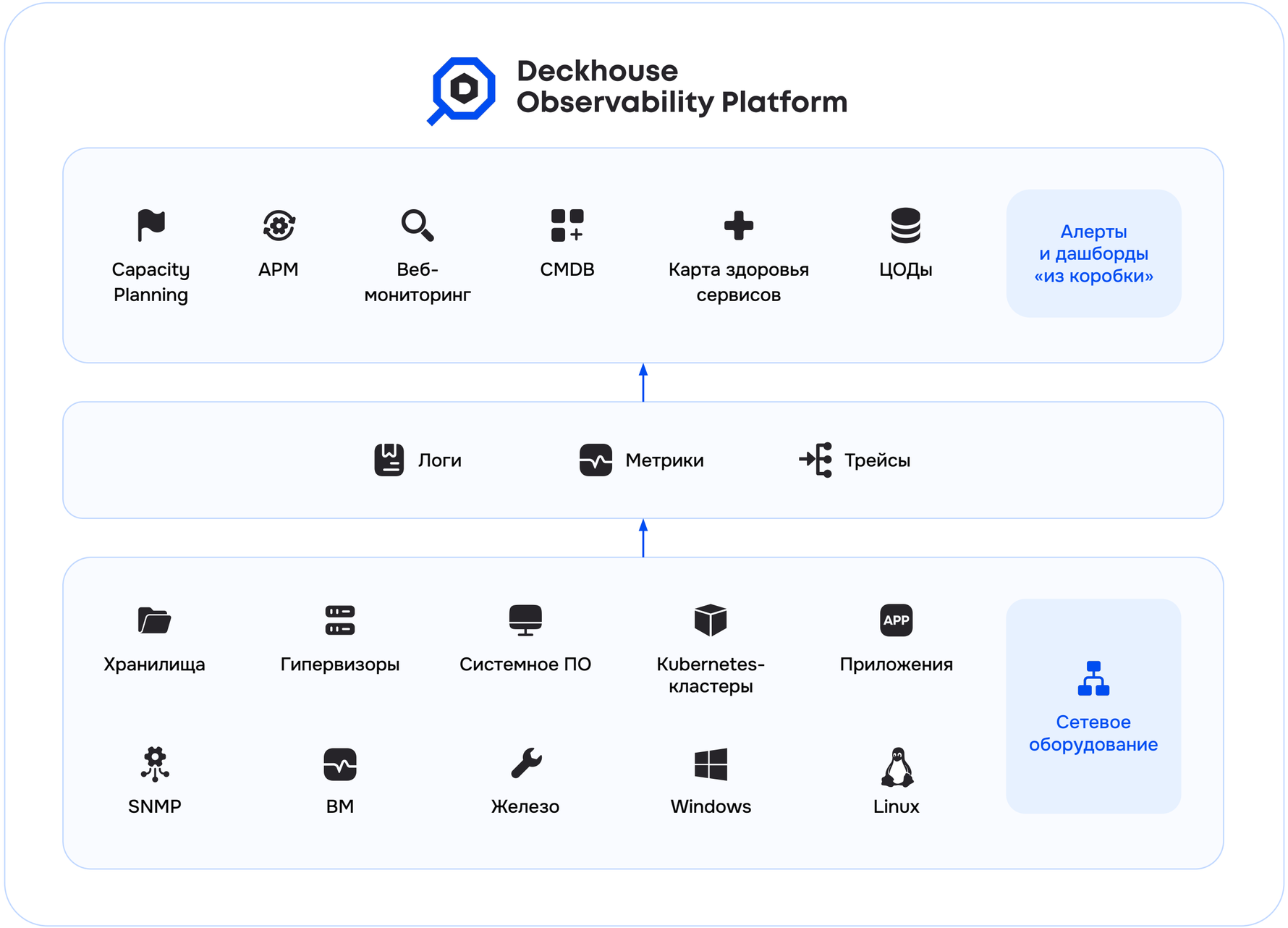
проактивное обнаружение инцидентов

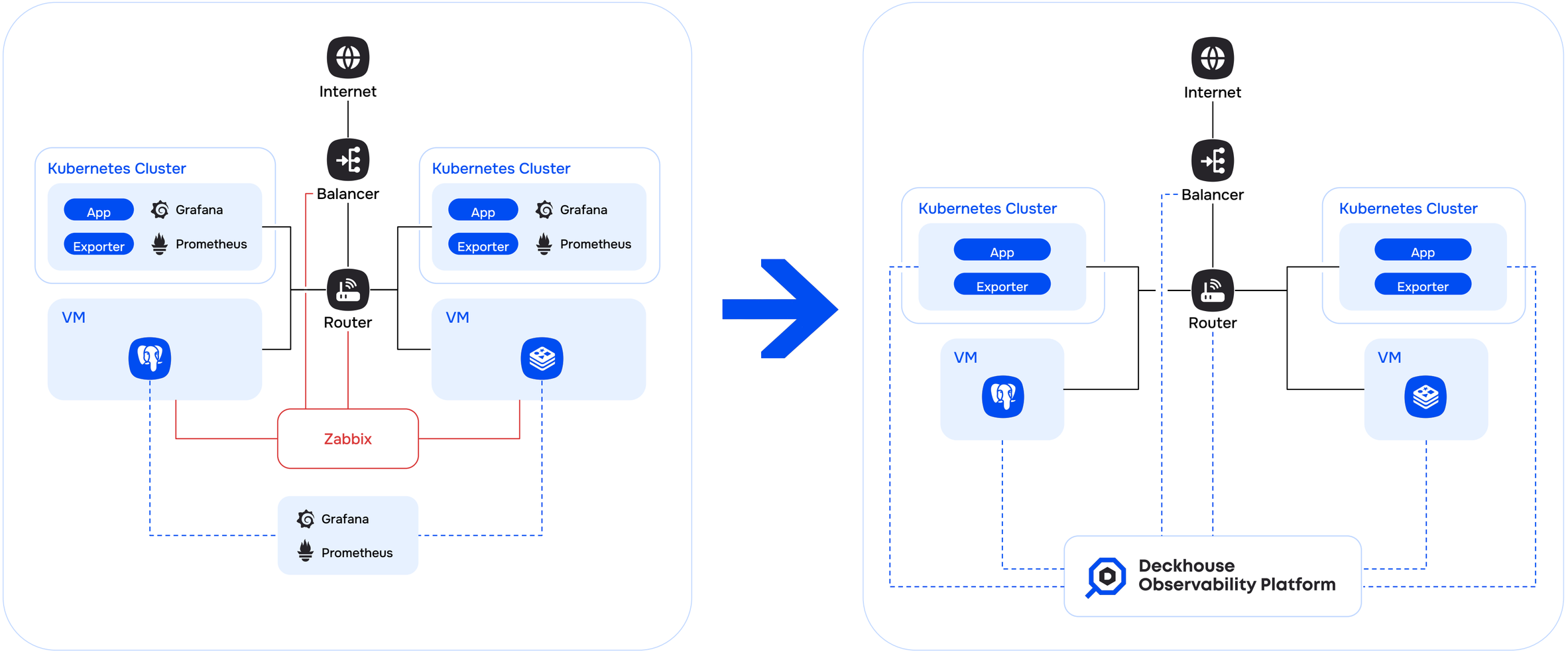


Ежемесячная экономия после

На простоях	300 млн × 20 % = 60 млн ₽
На трудозатратах	4,5 млн × 45 % = 2,025 млн ₽
Общая месячная экономия	**62,025 млн ₽**
Общая годовая экономия	**744,3 млн ₽**

Архитектура при использовании DOP


Deckhouse
Observability Platform
Capacity Planning
APM
Веб-мониторинг
CMDB
Карта здоровья сервисов
ЦОДы
Алерты и дашборды «из коробки»
Логи
Метрики
Трейсы
Хранилища
Гипервизоры
Системное ПО
Kubernetes-кластеры
Приложения
APP
Сетевое оборудование
SNMP
ВМ
Железо
Windows
Linux


Internet
Balancer
Kubernetes Cluster
App
Exporter
Grafana
Prometheus
Kubernetes Cluster
App
Exporter
Grafana
Prometheus
Router
VM
VM
Zabbix
Grafana
Prometheus
Internet
Balancer
Kubernetes Cluster
App
Exporter
Kubernetes Cluster
App
Exporter
Router
VM
VM
Deckhouse
Observability Platform

Но сколько это всё потребляет ресурсов?

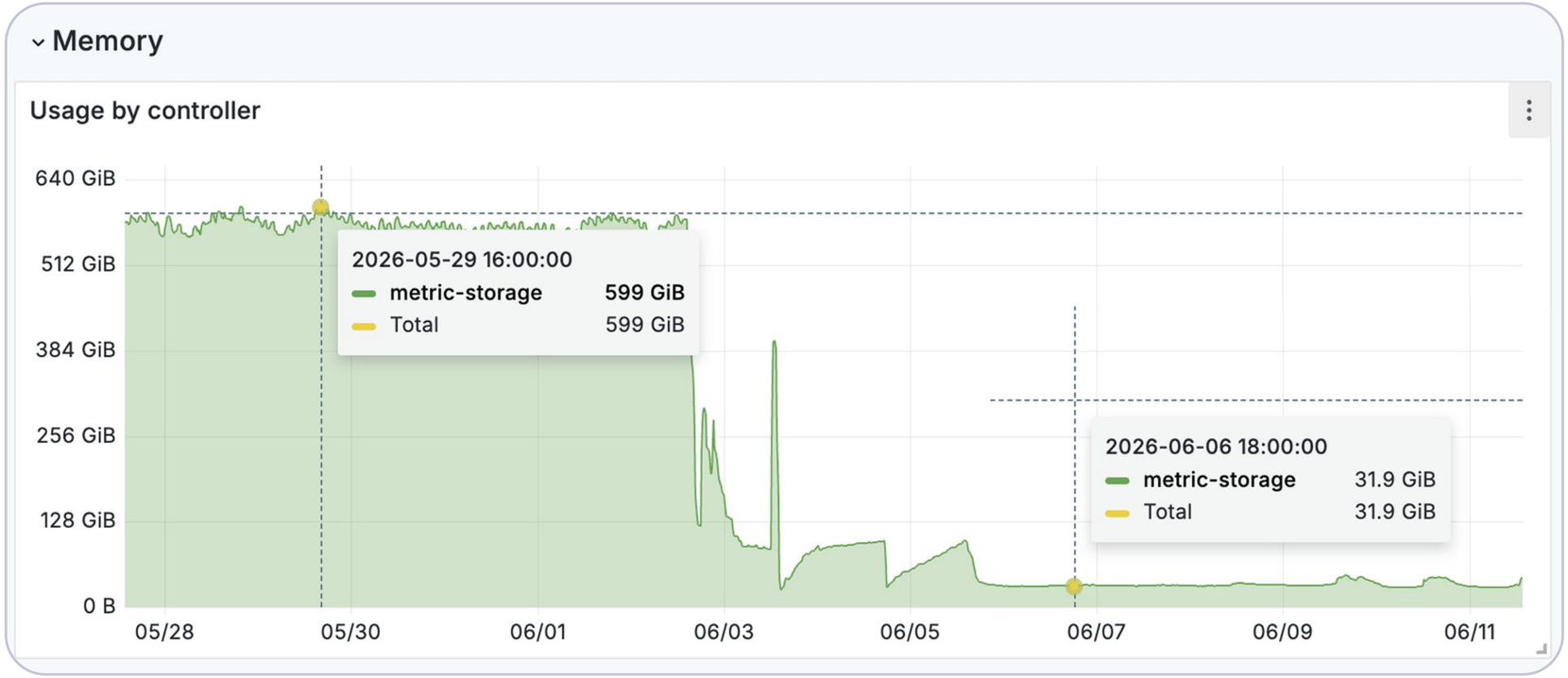
Оптимизировали хранилище метрик

снизилось

В 20 раз меньше

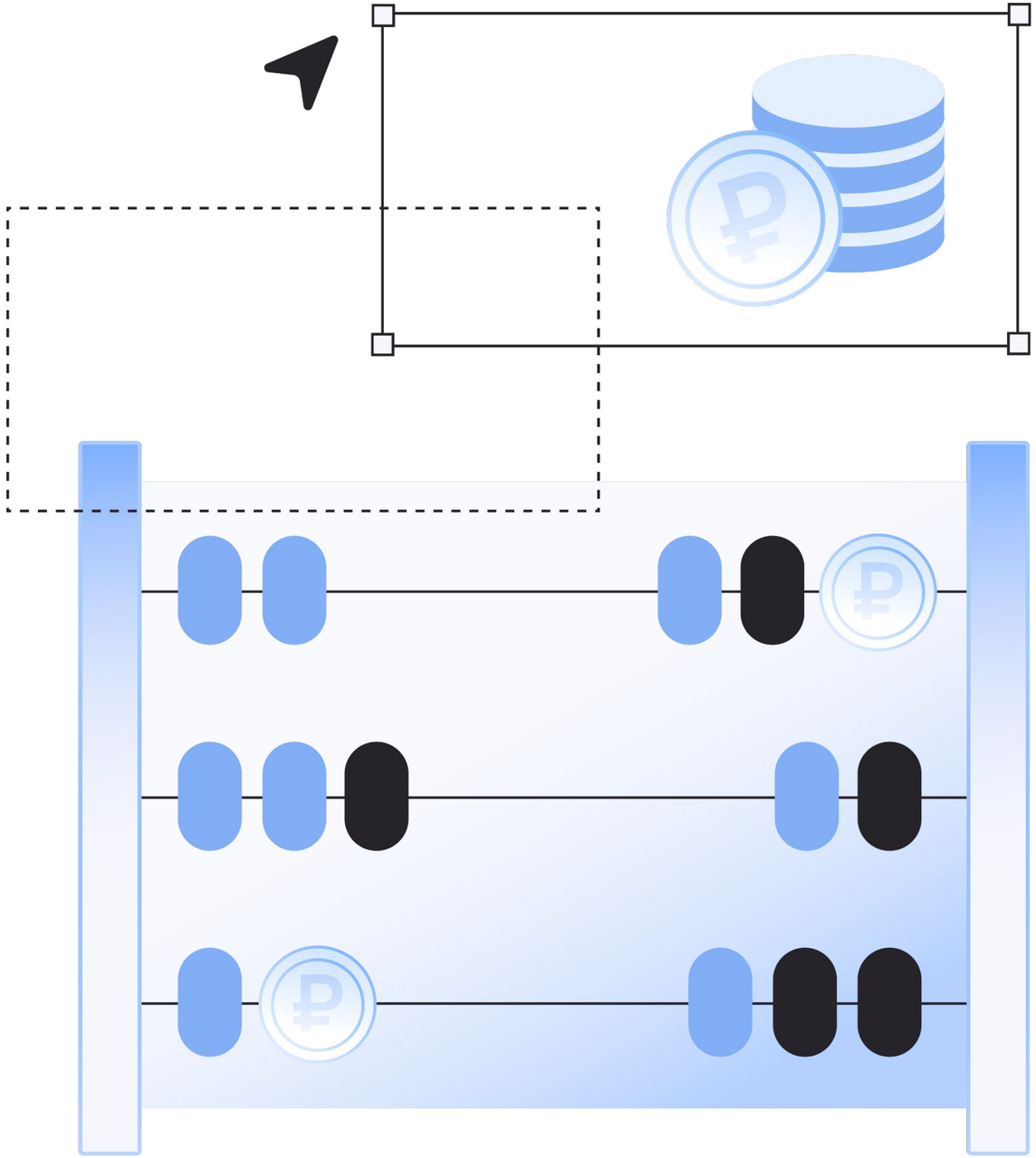
потребление памяти

Меньше памяти → меньше расходы на мониторинг → ниже стоимость владения



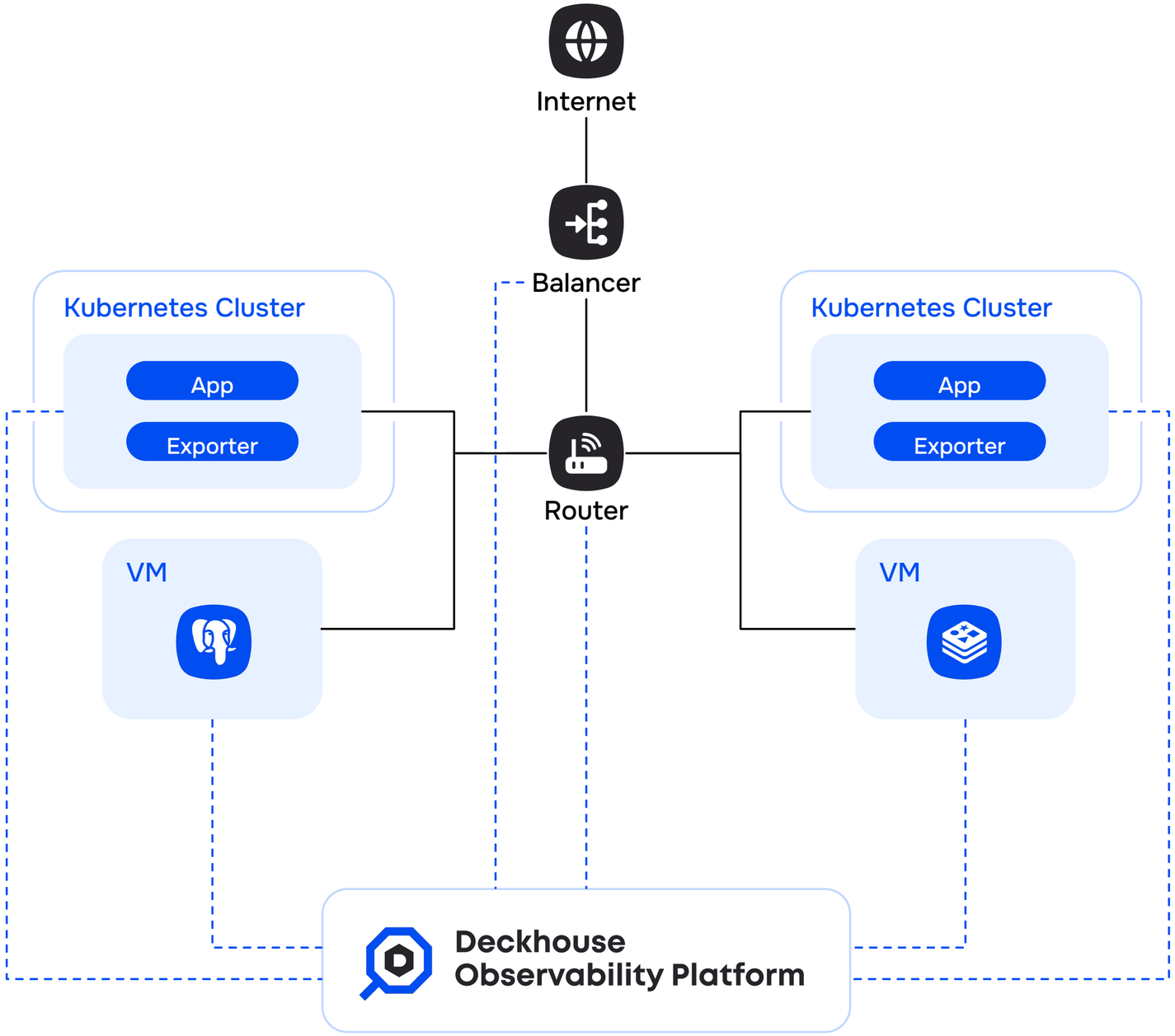
~10 млн ₽

составила экономия



Пример: как выглядит расследование инцидента с DOP?

Пример


Internet
Balancer
Kubernetes Cluster
App
Exporter
Router
Kubernetes Cluster
App
Exporter
VM
VM
Deckhouse
Observability Platform

DOP: мониторинг доступности веб-сайтов

Централизованное наблюдение веб-сервисов и API из разных регионов

- **Что проверяет:** HTTP(S)-запросы с валидацией кодов ответа, содержимое ответов, сетевую связность, доступность TCP-портов, DNS-записи и срок действия SSL-сертификатов
- Графики и алерты «из коробки»
- Отчет по SLO доступности сайтов
- Возможность проверки доступности из разных зон

01

Контроль доступности

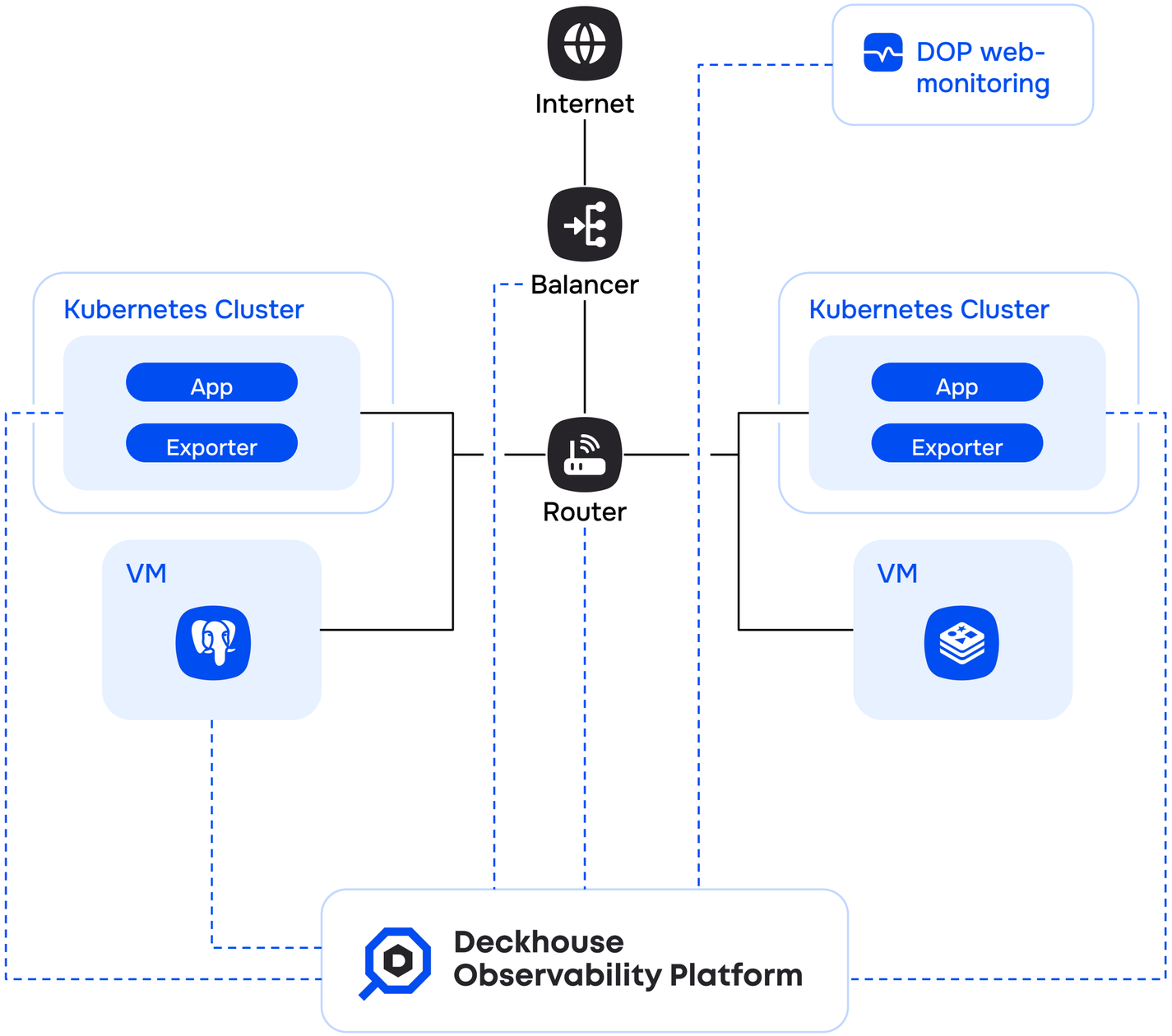
- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

Пример

01 Получаем алерт от веб-мониторинга, что время выполнения запросов увеличилось

02 Смотрим алерты и метрики нужного сервиса — видим реальное увеличение времени отклика

03 Ищем трейсы для таких запросов



Но напрямую с трейсами работать неудобно!

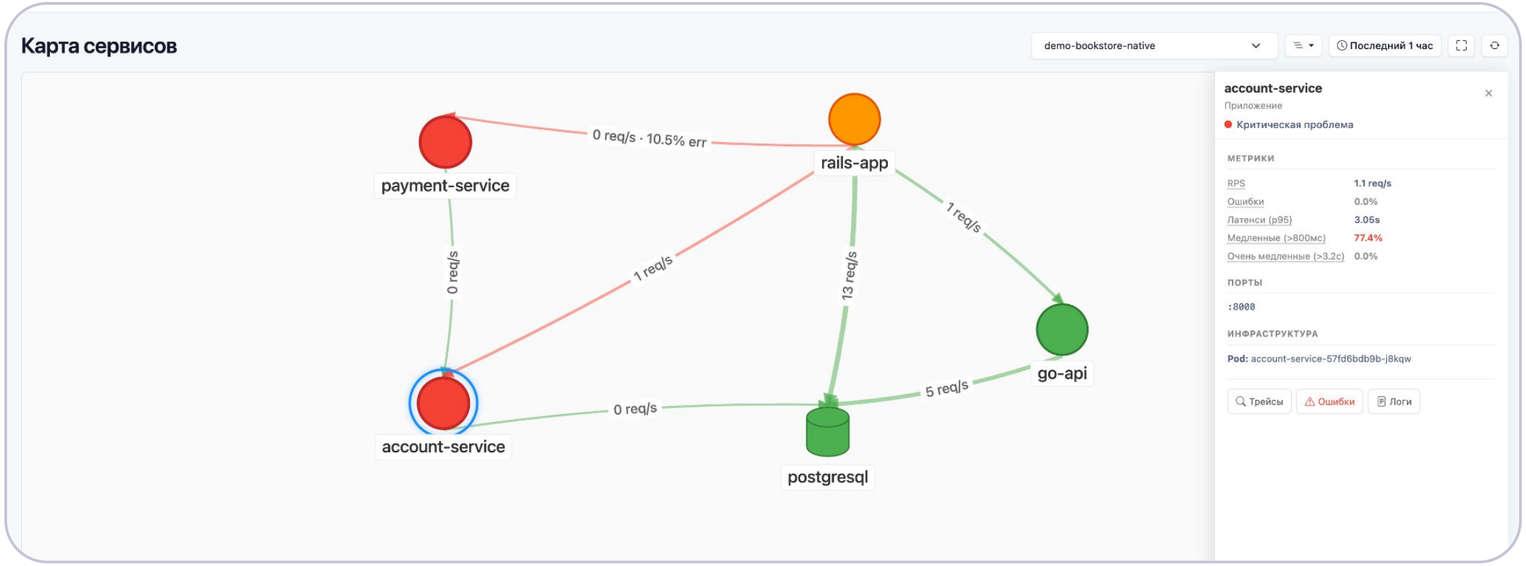
DOP: Application Performance Monitoring

Автоматический анализ производительности
и ошибок во всех сервисах приложения

- **Что показывает:** время ответа,
ошибки и Apdex по каждому сервису
- Автоматический анализ сервисов и отображение
проблемных сервисов и операций
- Просмотр логов, связанных с трейсом
- Автоматическое построение карты
сервисов — включая БД, кэши и очереди

DOP: APM-карта сервисов

Карта сервисов позволяет оценить, какие ещё сервисы были затронуты и насколько критичен сбой



01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

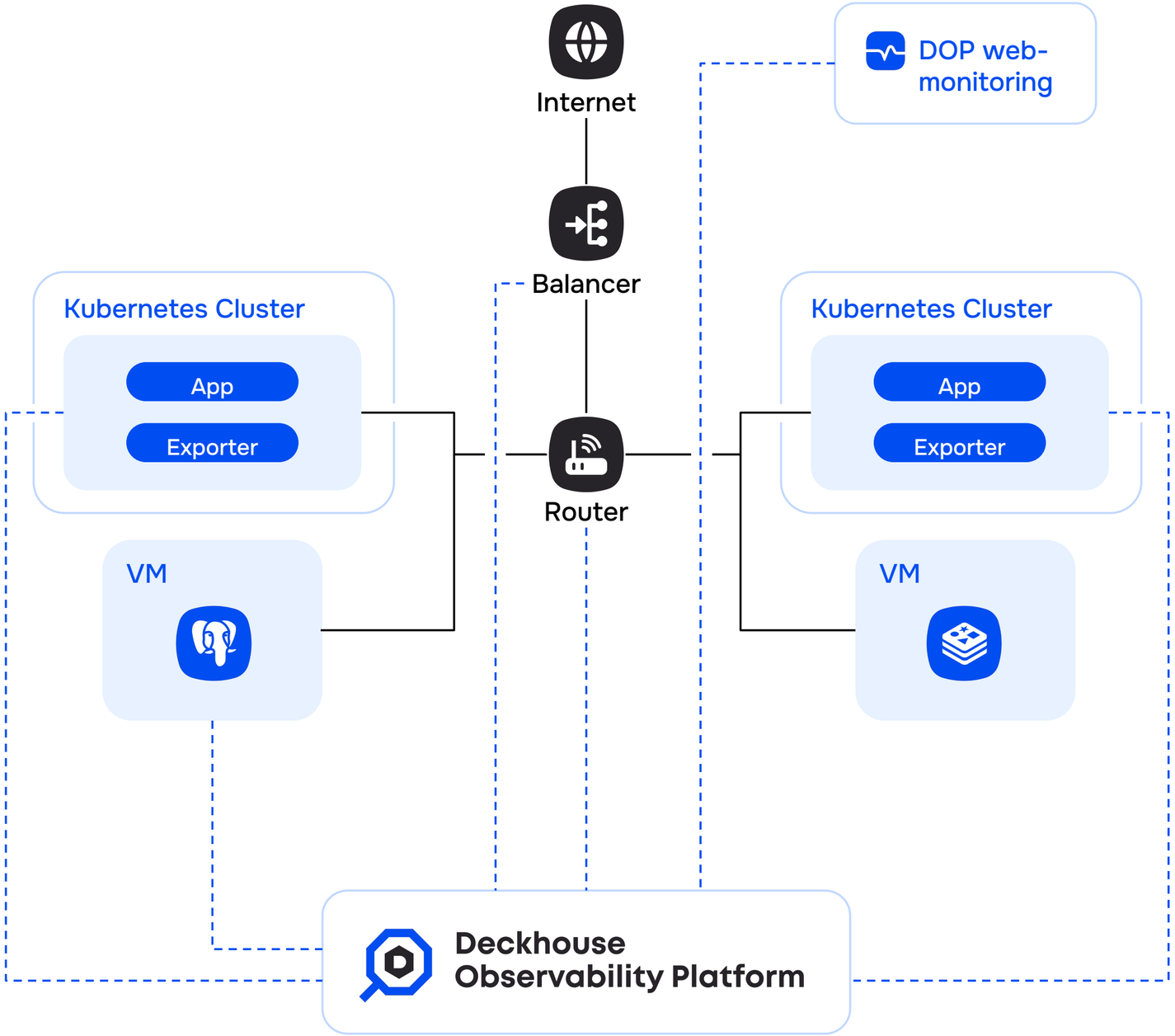
- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

Пример

01 Получаем алерт от веб-мониторинга, что время выполнения запросов увеличилось

02 Смотрим алерты и метрики нужного сервиса — видим реальное увеличение времени отклика

03 ~~Ищем трейсы для таких запросов,~~ Находим проблемный сервис в APM и видим, что виной этому медленные запросы в БД



Пример

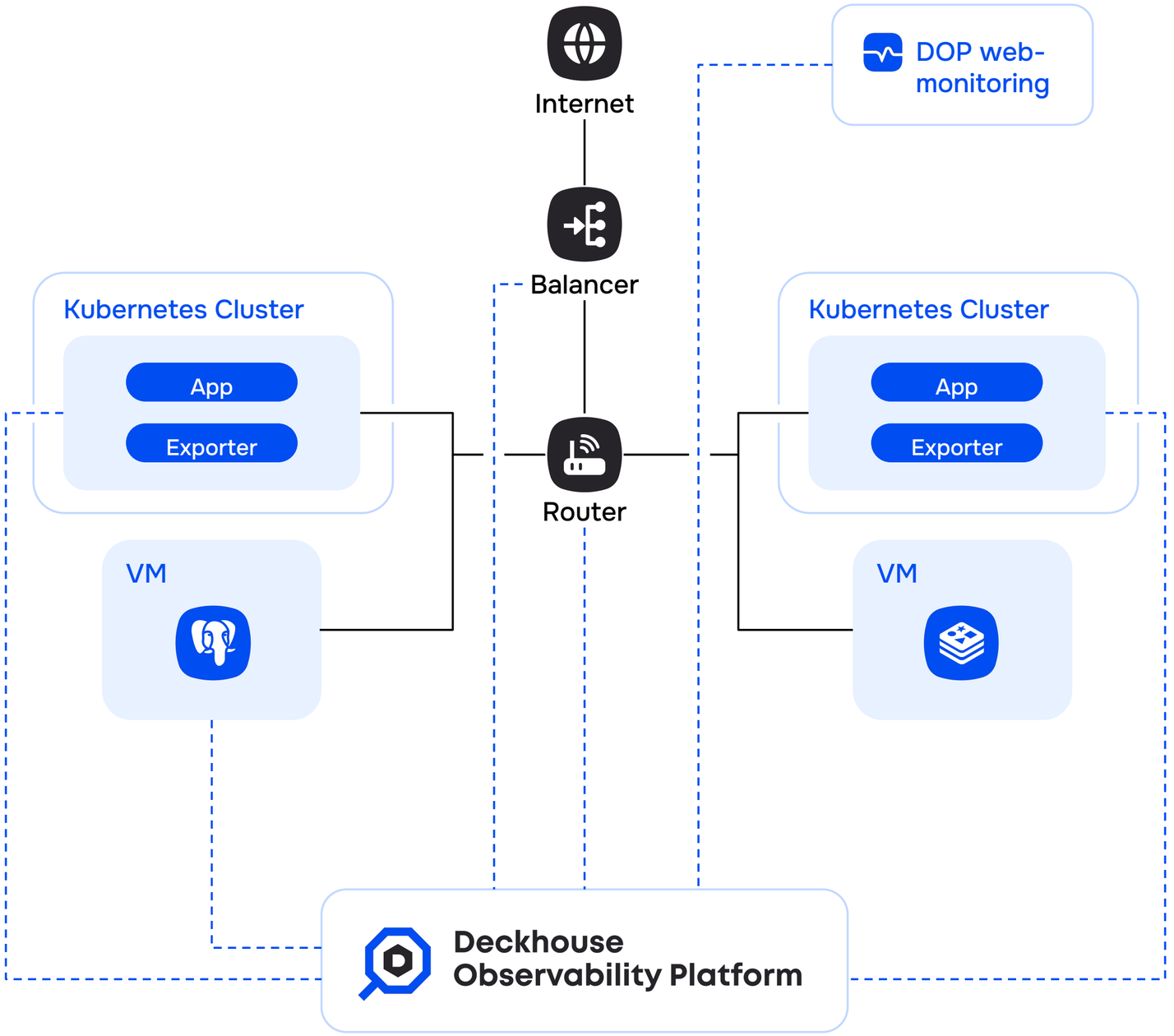
01 Получаем алерт от веб-мониторинга, что время выполнения запросов увеличилось

02 Смотрим алерты и метрики нужного сервиса — видим реальное увеличение времени отклика

03 Находим проблемный сервис в APM и видим, что виной медленные запросы в БД

04 По алертам/дашбордам от ВМ с БД видим, что проблема с диском

05 Переключаем трафик на другой инстанс БД



Как узнать,
какой диск менять?

Мониторинг CMDB

Автоматическая инвентаризация серверов до уровня компонентов

- **Что хранит:** конфигурацию каждого сервера — CPU, память, диски, контроллеры, сетевые карты; для каждого компонента модель, слот и серийный номер
- Вся информация собирается агентом автоматически — без ручного ввода
- История изменений: установка, удаление, перемещение компонентов

ТОПОЛОГИЯ ХРАНИЛИЩ

Контроллер Broadcom / LSI SAS3008 PCI-Express Fusion-MPT SAS-3 8 PHY, 2 PD 0000:01:00.0

- [port-0:0] SAS SSD 447 GB SAMSUNG MZ7LH480 /dev/sda
- [port-0:1] SAS SSD 447 GB SAMSUNG MZ7LH480 /dev/sdb

NVMe Прямое подключение PCIe

- NVME SSD 6.99 TB INTEL SSDPF2KX076T1 /dev/nvme0n1
- NVME SSD 3.49 TB INTEL SSDPF2KX038TZ /dev/nvme1n1

01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

03 Инфраструктура и оборудование

- Актуальная инвентаризация серверов до серийного номера компонента — без ручного учёта
- История изменений оборудования: установки, замены, перемещения

Пример

01 Получаем алерт от веб-мониторинга, что время выполнения запросов увеличилось

02 Смотрим алерты и метрики нужного сервиса — видим реальное увеличение времени отклика

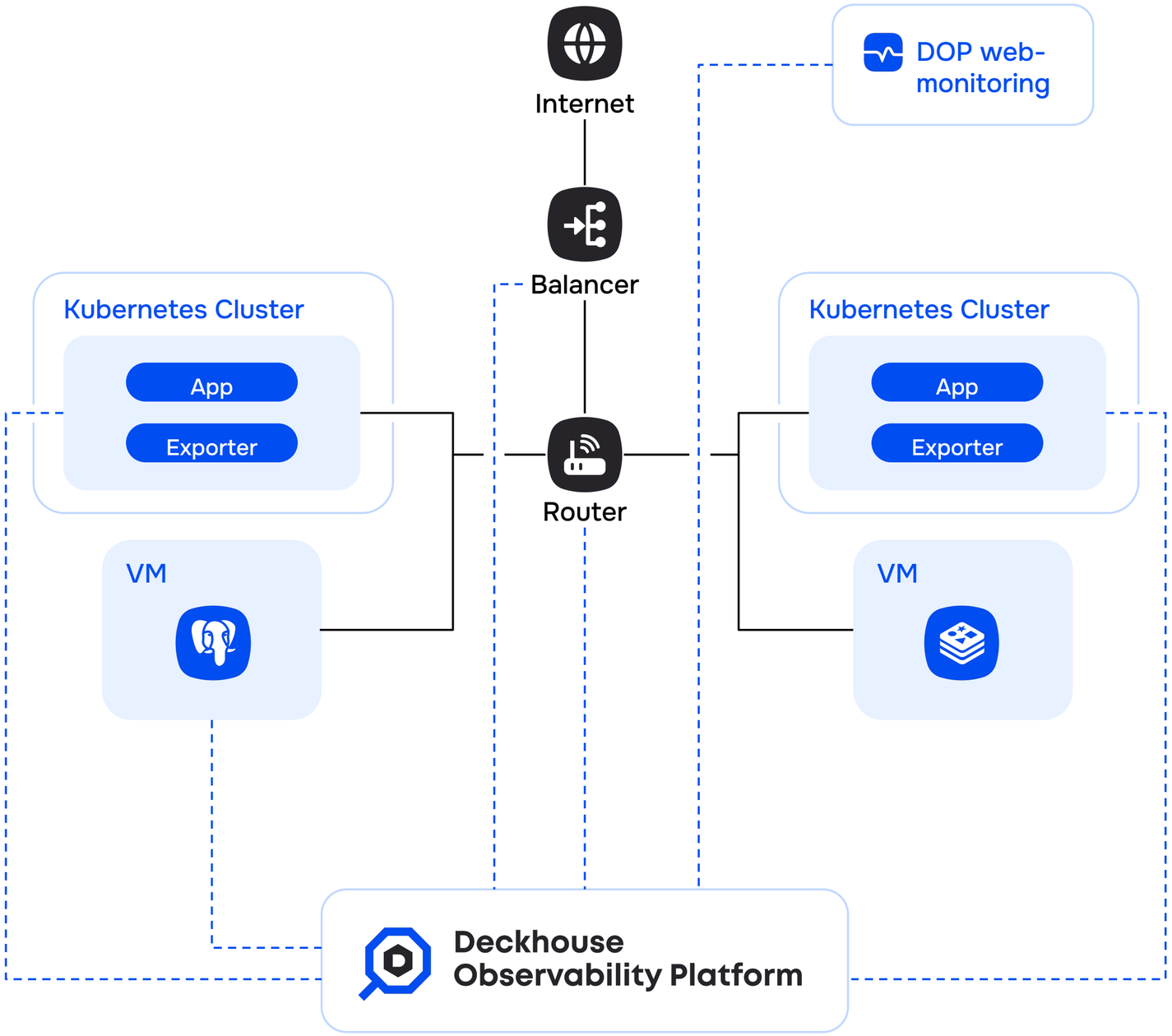
03 Находим проблемный сервис в APM и видим, что виной медленные запросы в БД

04 По алертам/дашбордам от ВМ с БД видим, что проблема с диском

05 Переключаем трафик на другой инстанс БД

06 По CMDB получаем серийник и слот, куда установлен нужный диск

07 В разделе веб-мониторинга смотрим, не выходим ли мы за границы SLO



okmeter.ru

Редактировать | Подробный дашборд | Логи | История изменений

4	0	0	100.0%	97.64%	—
ПРОВЕРКИ			ДОСТУПНОСТЬ ЗА 1 ЧАС	ДОСТУПНОСТЬ ЗА 1 ДЕНЬ	ДОСТУПНОСТЬ ЗА 30 ДНЕЙ

Хост	okmeter.ru
Период	30s
Зоны	Russia, Europe
Пробы	1
Изменён	04 июня 17:01

СТАТУС ПО ЗОНАМ

ЗОНА	ДОСТУПНОСТЬ 1 ЧАС	ДОСТУПНОСТЬ 1 ДЕНЬ	ДОСТУПНОСТЬ 30 ДНЕЙ	HTTP TTLB
europe	100.0%	97.78%	—	487 ms
russia	100.0%	97.64%	—	478 ms

Пример: история одного коммутатора

Без DOP

01 Выход из строя коммутатора

02 Получаем более 100 алертов

03 Ручной анализ взаимосвязи этих алертов

04 Обнаружение сбойного коммутатора

05 Замена

Срок ~от 30 минут до 1 часа

С DOP

01 Выход из строя коммутатора

02 Получаем более 100 алертов

03 Карта инфраструктуры

04 Обнаружение сбойного коммутатора

05 Замена

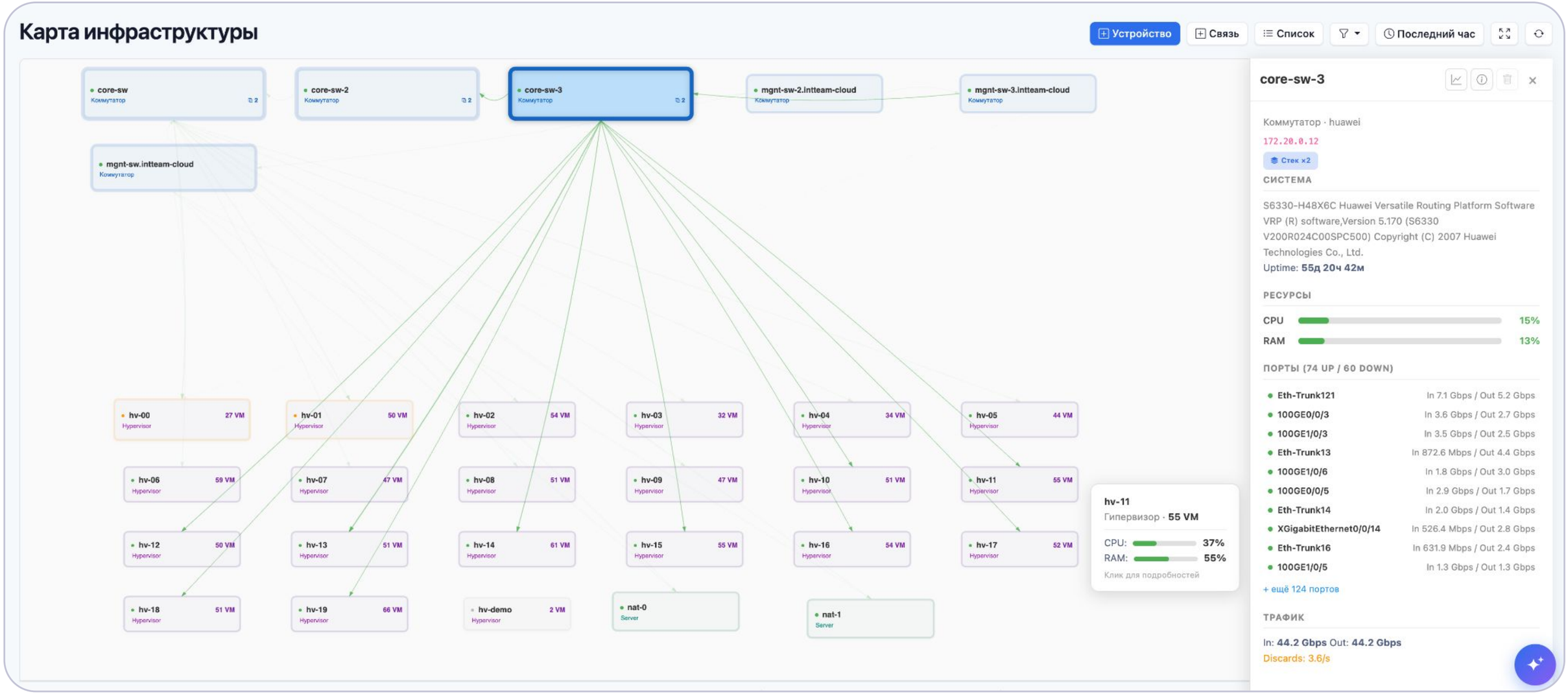
Срок ~от 5 минут

Мониторинг инфраструктуры

Автоматическое наблюдение за физическими устройствами: серверы, гипервизоры, сетевое оборудование

- **Что показывает:** состояние и загрузку каждого устройства, связи между устройствами, ошибки на портах и интерфейсах
- Карта инфраструктуры со связями между устройствами
- Оценка влияния сбоя устройства на зависимые компоненты

Карта инфраструктуры



01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

03 Инфраструктура и оборудование

- Актуальная инвентаризация серверов до серийного номера компонента — без ручного учёта
- История изменений оборудования: установки, замены, перемещения

Что нужно, чтобы собрать все эти данные?

- Postgres — postgres-exporter
- Linux — node-exporter
- Redis — redis-exporter
- Smart — smartctl-exporter
- Router — snmp-exporter
- Balancer (nginx) — nginx-exporter
- И ещё с десяток приложений для мониторинга

Проблемы:

- Много различных приложений, которые надо не забыть поставить
- Надо отслеживать уязвимости и своевременно обновлять
- У каждого свой набор настроек
- Потребление ресурсов
- Зачастую нет дашбордов и алертов, либо они неоптимальны

Что можно сделать вместо этого?

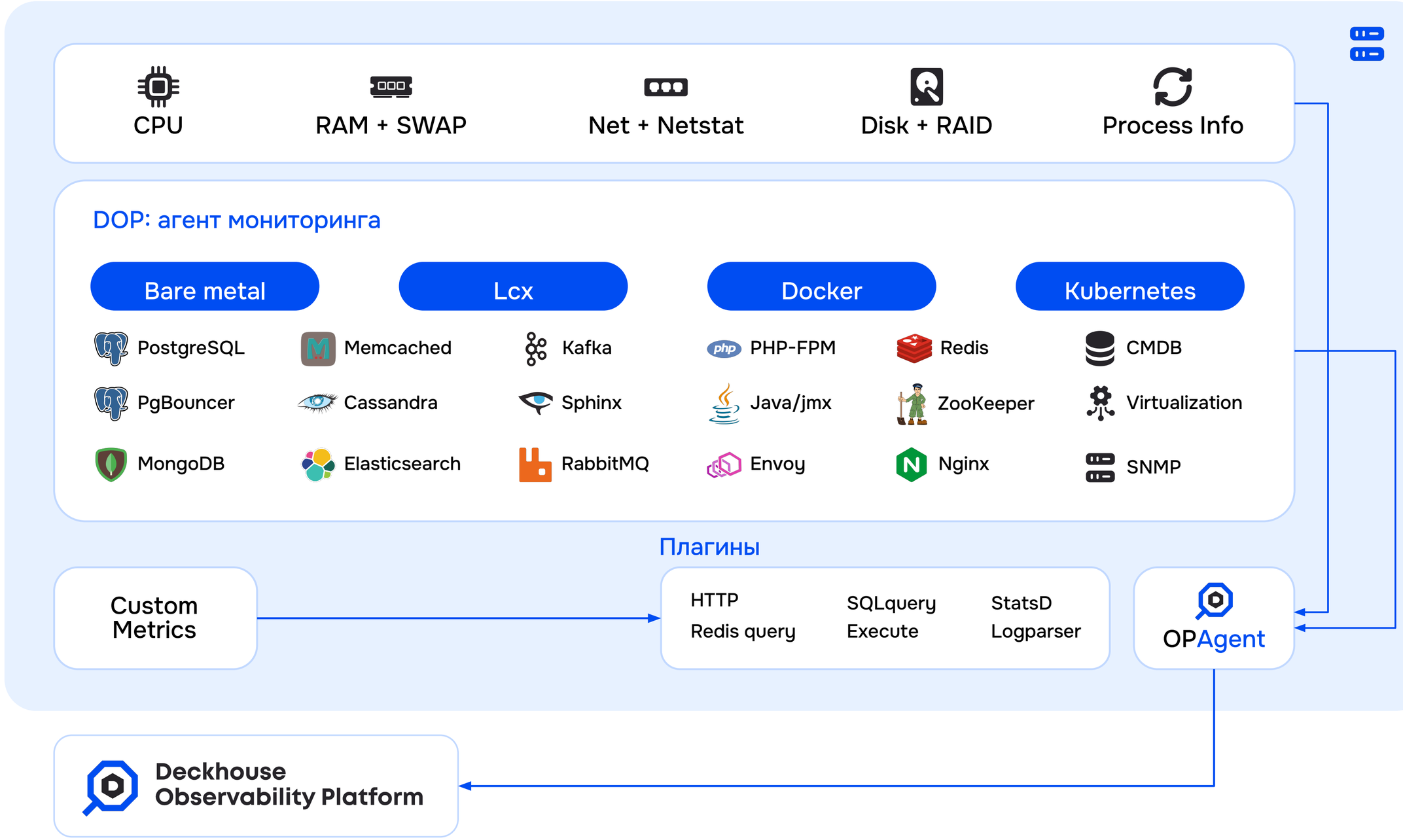
Агент мониторинга DOP — всё в одном!

Как это работает?

01 Обнаруживает компоненты инфраструктуры и установленное ПО в формате автодискавери

02 Сам собирает метрики

03 Преднастроенные дашборды и алерты «из коробки»


CPU
RAM + SWAP
Net + Netstat
Disk + RAID
Process Info
DOP: агент мониторинга
Bare metal
Lcx
Docker
Kubernetes
PostgreSQL
Memcached
Kafka
PHP-FPM
Redis
CMDB
PgBouncer
Cassandra
Sphinx
Java/jmx
ZooKeeper
Virtualization
MongoDB
Elasticsearch
RabbitMQ
Envoy
Nginx
SNMP
Плагины
Custom Metrics
HTTP
Redis query
SQLquery
Execute
StatsD
Logparser
OPAgent
Deckhouse Observability Platform

И в отличие от Open Source-экспортеров, с DOP не нужно:

- подбирать отдельные решения под компоненты ИТ-системы
- вручную собирать дашборды
- настраивать алерты под каждое ПО

Аналога агента DOP не существует

01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

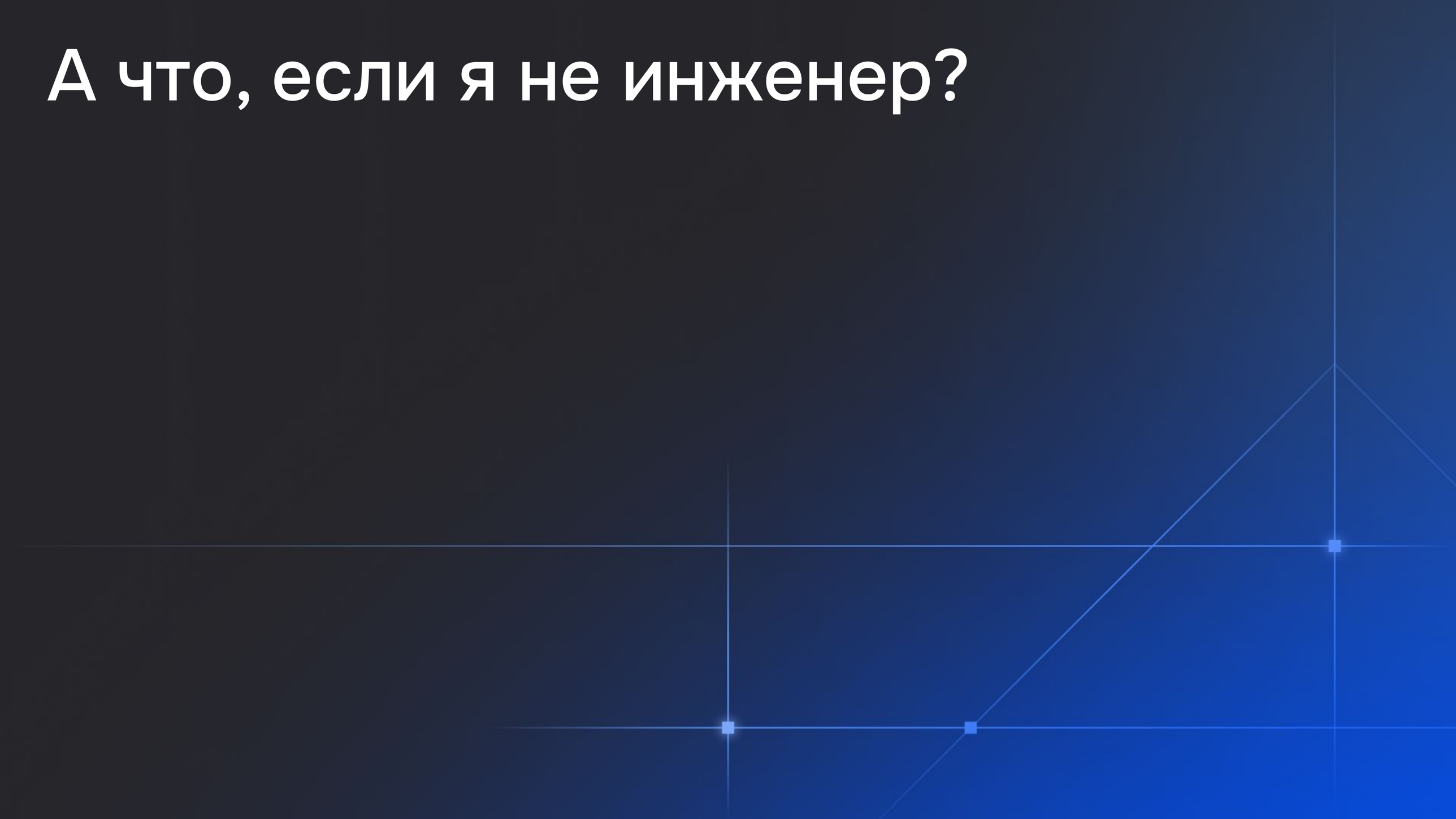
03 Инфраструктура и оборудование

- Актуальная инвентаризация серверов до серийного номера компонента — без ручного учёта
- История изменений оборудования: установки, замены, перемещения

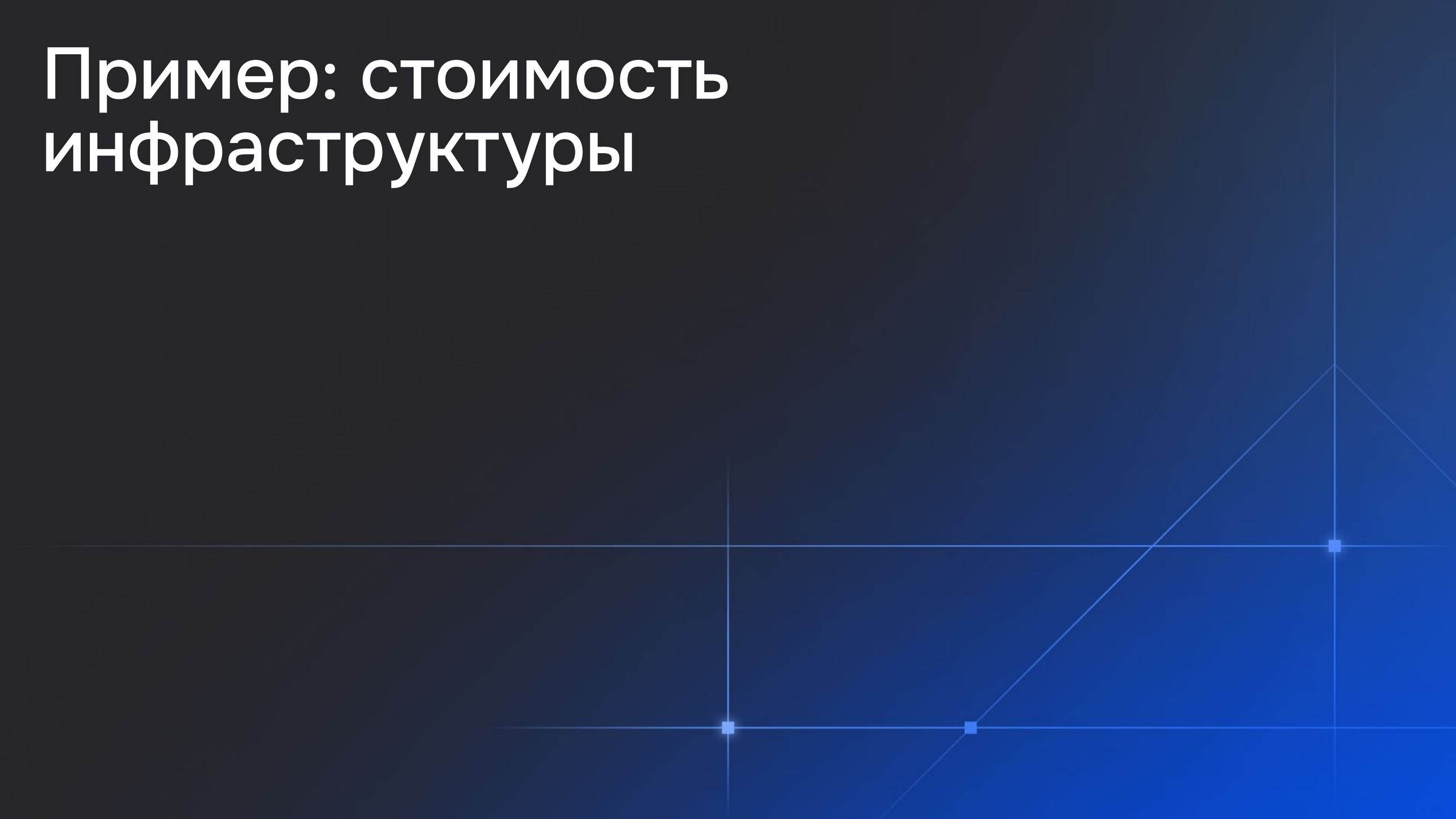
04 Простота эксплуатации

- Мониторинг всего стека одним агентом — от железа до БД и очередей
- Готовые дашборды и алерты для всего, что обнаружил агент

А что, если я не инженер?



Пример: стоимость инфраструктуры



Без DOP

01 Финдиректор

02 Хочу знать, сколько стоит инфраструктура и сколько ресурсов простаивает

03 Задача для инженера

04 Приоритизация

05 Бэклог

06 В работе

07 Результат

08 Вторая итерация с уточнениями

Срок ~от 1 до 3 дней

С DOP

01 Финдиректор

02 AI-ассистент (который понимает контекст)

DOP: AI-ассистент

Упрощает работу с платформой и ускоряет расследование инцидентов

- **Что умеет:** создаёт алерты, дашборды и сайленсы;
  расследует инциденты, анализируя все доступные данные;
  объясняет, что на графиках и как это интерпретировать

- Понимает контекст: знает,
  какие проект и страница открыты

- Отвечает на вопросы по платформе
  и помогает с настройкой — по документации

01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

03 Инфраструктура и оборудование

- Актуальная инвентаризация серверов до серийного номера компонента — без ручного учёта
- История изменений оборудования: установки, замены, перемещения

04 Простота эксплуатации

- Мониторинг всего стека одним агентом — от железа до БД и очередей
- Готовые дашборды и алерты для всего, что обнаружил агент
- Данные мониторинга доступны любому сотруднику — вопросом к AI-ассистенту
- Рутинные операции и сбор данных по инциденту — через ассистента

Без DOP

01 Финдиректор

02 Хочу знать, сколько стоит инфраструктура и сколько ресурсов простаивает

03 Задача для инженера

04 Приоритизация

05 Бэклог

06 В работе

07 Результат

08 Вторая итерация с уточнениями

Срок ~от 1 до 3 дней

С DOP

01 Финдиректор

02 AI-ассистент (который понимает контекст)

03 Результат

Срок **от 20 до 40 минут**

Несколько фактов о самой платформе

01 Ролевая модель доступа
и журнал аудита действий

02 Интеграция с корпоративной
аутентификацией: AD, Keycloak и другие

03 Отказоустойчивость:
HA-режим по умолчанию

04 Горизонтальное масштабирование
за счёт шардирования данных

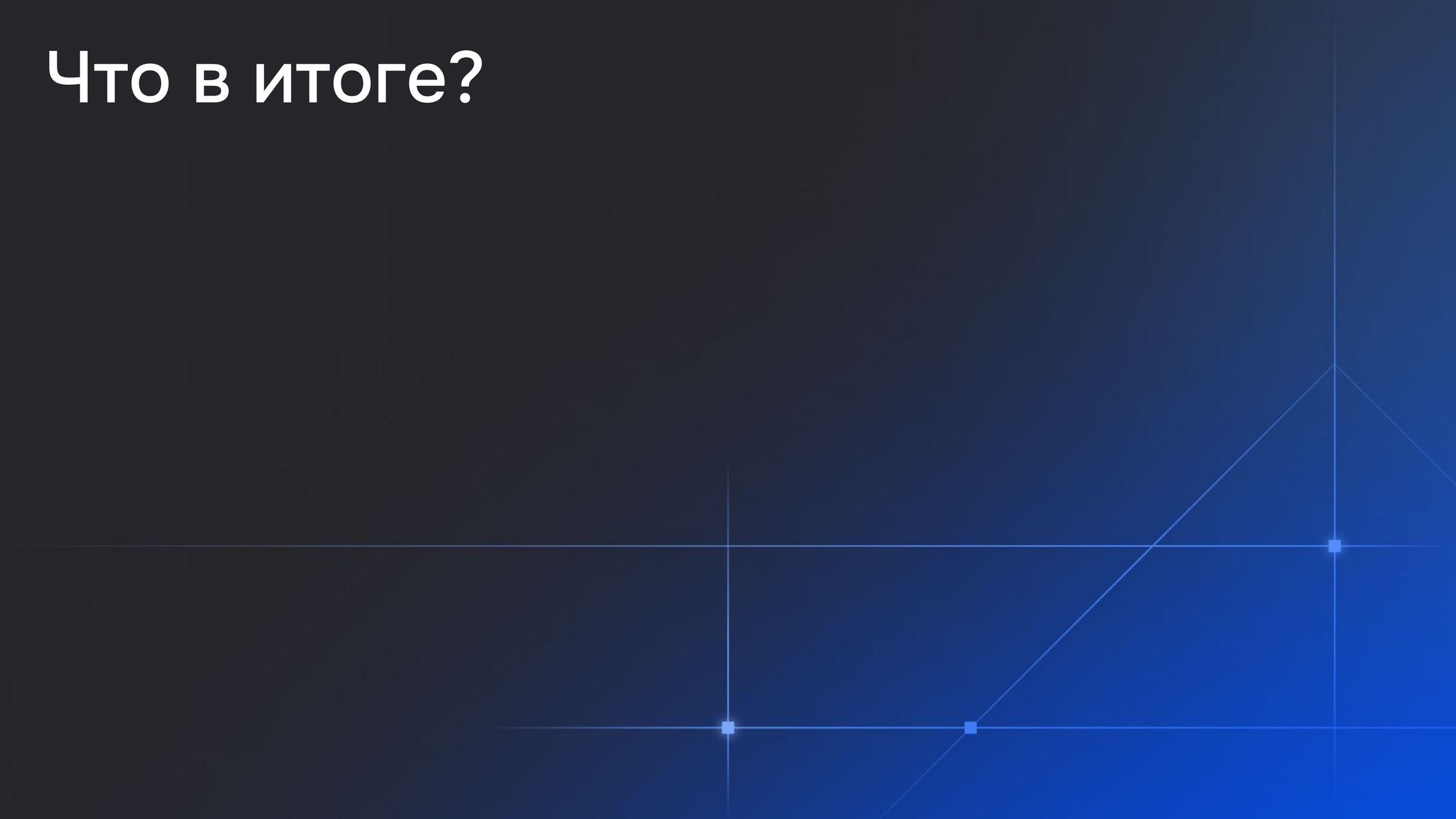
05 Многолетнее хранение
метрик, логов и трейсов

06 Оптимизированное хранилище метрик —
кратно меньше потребление памяти

07 Включена в реестр отечественного ПО

Deckhouse
Observability Platform

Что в итоге?



01 Контроль доступности

- Постоянный контроль доступности сервисов из внешних точек и разных регионов
- Объективные данные для расчёта и подтверждения SLA/SLO

02 Расследование инцидентов

- Поиск причины деградации без ручного разбора: от алерта до проблемной операции и её логов
- Видимость всех сервисов приложения: состояние, ошибки, время ответа и зависимости
- Метрики, логи и трейсы в одном интерфейсе — с переходами между ними

03 Инфраструктура и оборудование

- Актуальная инвентаризация серверов до серийного номера компонента — без ручного учёта
- История изменений оборудования: установки, замены, перемещения

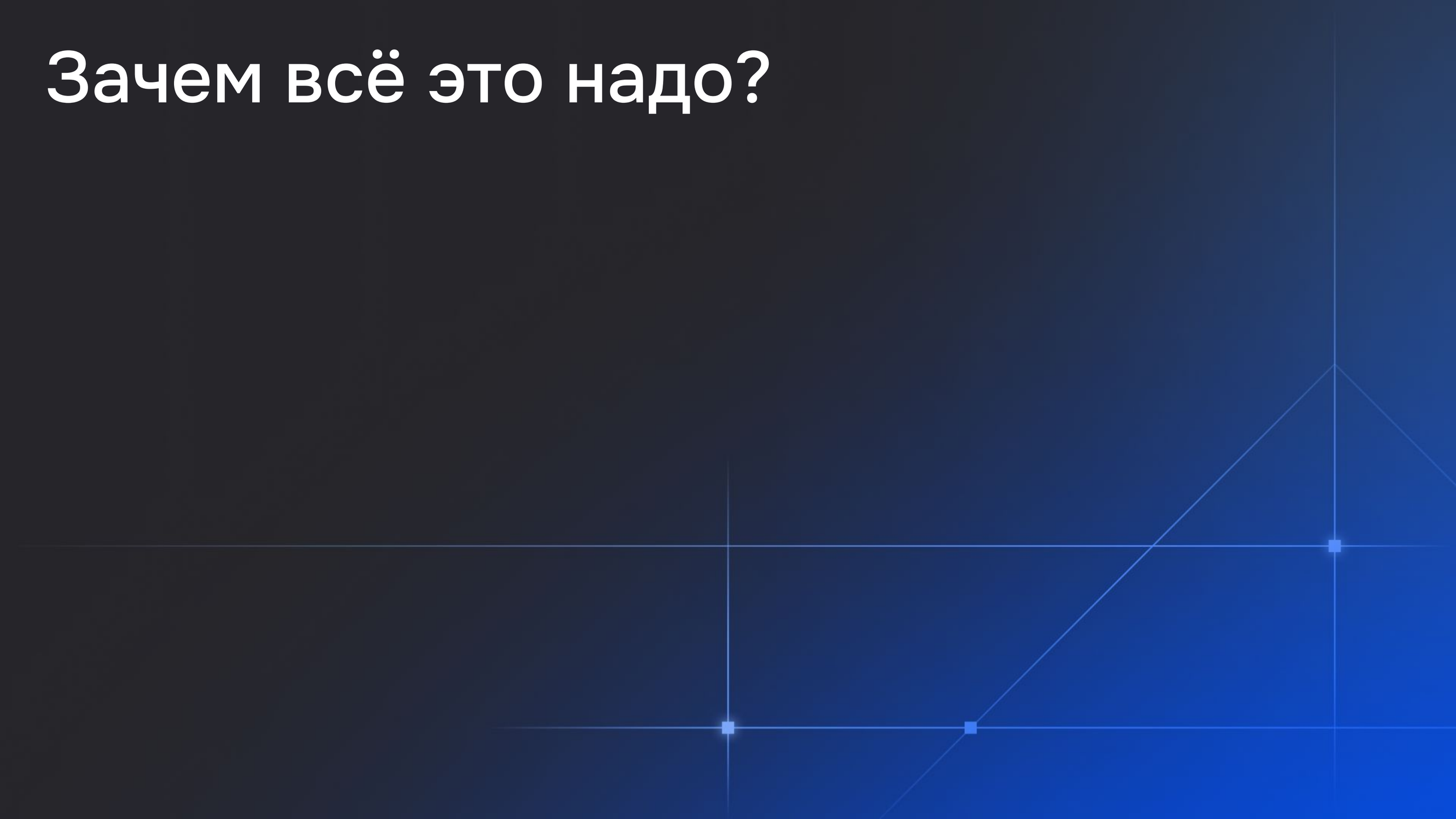
04 Простота эксплуатации

- Мониторинг всего стека одним агентом — от железа до БД и очередей
- Готовые дашборды и алерты для всего, что обнаружил агент
- Данные мониторинга доступны любому сотруднику — вопросом к AI-ассистенту
- Рутинные операции и сбор данных по инциденту — через ассистента

05 Enterprise-платформа

- Ролевая модель доступа и журнал аудита действий
- Интеграция с корпоративной аутентификацией: AD, Keycloak и другие
- Отказоустойчивость: HA-режим по умолчанию
- Горизонтальное масштабирование за счёт шардирования данных
- Многолетнее хранение метрик, логов и трейсов
- Оптимизированное хранилище метрик — кратно меньше потребление памяти
- Включена в реестр отечественного ПО

Зачем всё это надо?



Зачем всё это?

Эффективные компоненты платформы

Кратно меньше затрат
на инфраструктуру мониторинга

Все данные и инструменты анализа в одном месте

Инцидентов меньше, а их решение
быстрее. Меньше потерь для бизнеса

Что ещё планируется в 2026 году?

План развития продукта на 2026 год

01 **Ресурсно-сервисная модель (карта здоровья)**

Отображает текущее состояние ресурсов и сервисов, помогает увидеть, где возникла проблема и на какие системы она влияет

02 **Статистика использования ресурсов инфраструктуры**

Показывает реальную и пиковую нагрузку инфраструктурных компонентов, сервисов и выделенных мощностей. Позволяет спланировать ресурсы на будущие периоды

03 AIOps-ассистент и RCA

Анализ алертов, телеметрии и связей между компонентами для поиска корневой причины инцидента и сбора контекста. Ассистент предложит шаги по решению проблемы и сможет выполнить часть из них

04 Прогнозирование и аномалии — ML/AI

Система использует ML-модели, графы зависимостей, исторические данные, карты сервисов и инфраструктуры, Это позволяет обнаруживать риски сбоев и предотвращать их до того, как проблема отразится на доступности сервиса

05 Сертификация ФСТЭК России

Соответствие требованиям регулятора для использования в защищённых средах*

* Начало процедуры сертификации

Хотите проверить, как DOP сможет снизить время на устранение инцидентов?

Расскажите нам о вашей инфраструктуре и типовых инцидентах. Мы покажем, как с помощью DOP можно объединить данные о состоянии систем, быстрее локализовать проблему и перейти от алерта к причине.

Запросить демонстрацию

Deckhouse Professional Services

Клиентам

Услуги по внедрению и адаптации решений экосистемы Deckhouse, миграции приложений и оптимизации управления их жизненным циклом

Целостный подход

Не только внедрение технологий, но и оптимизация процессов поставки цифровых продуктов, опирающихся на экосистему Deckhouse

Быстрый Time to Value

Использование отработанных методологий и лучших практик для быстрого и надёжного внедрения экосистемы Deckhouse, чтобы обеспечить максимально быстрый возврат инвестиций

Передача знаний

Зрелый подход к оформлению документации, вебинары для администраторов и пользователей Deckhouse, обучение и сертификация в Deckhouse Академии

Адаптация работ под клиента

От выполнения работ под ключ до архитектурного надзора или помощи в оформлении документации — возможны различные сценарии реализации услуг

Практико-ориентированное обучение

8 дней

Администрирование Deckhouse Kubernetes Platform (DKP)

10 дней

Использование DKP: запуск и администрирование приложений

6 дней

Инструменты безопасности в DKP

5 дней

Использование системы мониторинга в DKP

5 дней

Сетевые возможности DKP

Deckhouse Academy

Бесплатный курс для клиентов и партнёров

Возможности и инструменты DKP

Сертификация специалистов

Официальное подтверждение знаний и навыков работы с Deckhouse Kubernetes Platform:

01 Администратор DKP

02 Инженер безопасности DKP

Deckhouse Academy

Промокод со скидкой 15 % на все курсы Deckhouse Академии

Напишите промокод

DOP15

На почту education@flant.ru

До 17 августа включительно

Все курсы Deckhouse Академии

Deckhouse

Узнайте подробнее

о Deckhouse Observability Platform
и оставьте заявку на демонстрацию

Скачать краткий обзор DOP в PDF

Оставить заявку

+7 (495) 721-10-27

deckhouse.ru