Dooka

Высокопроизводительный мониторинг веб-сервисов

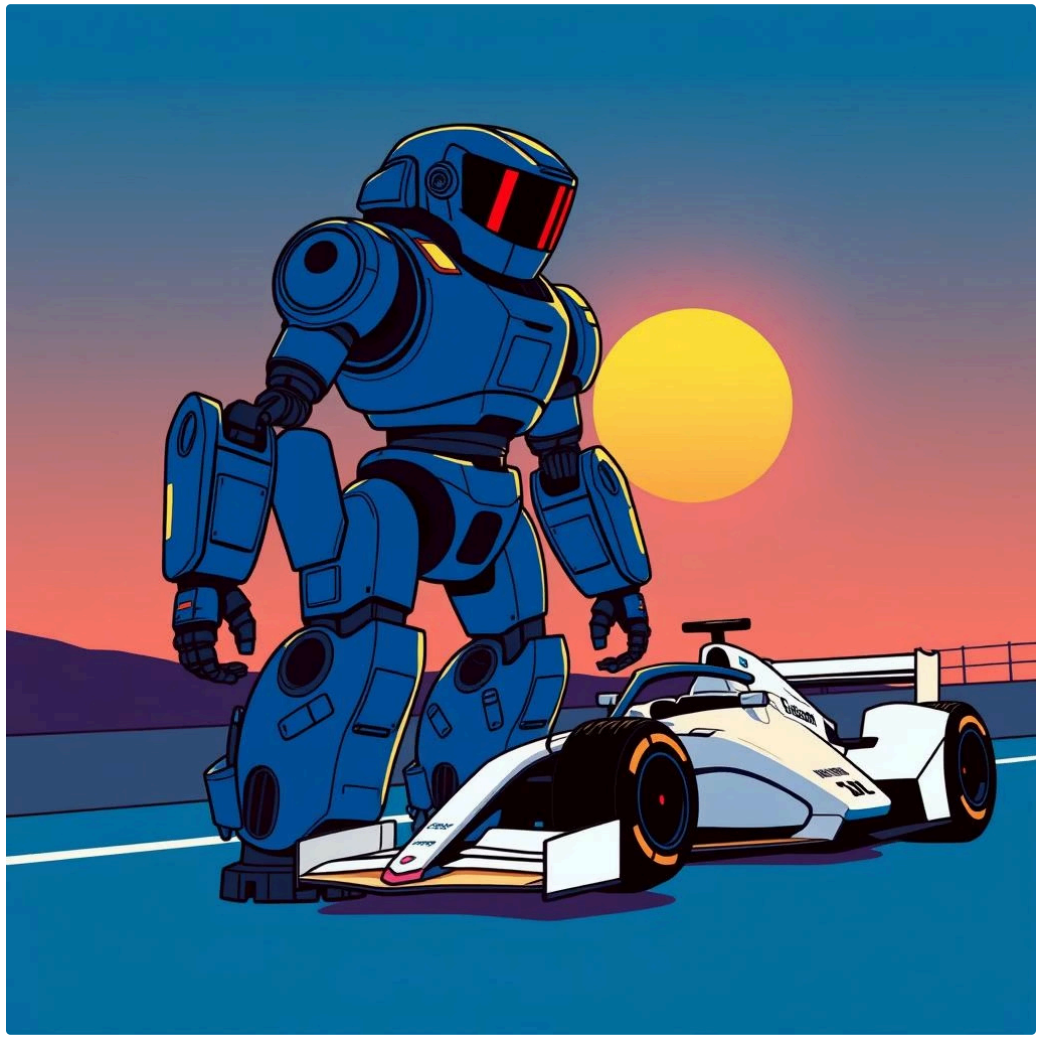
Команда: **Nice Development**

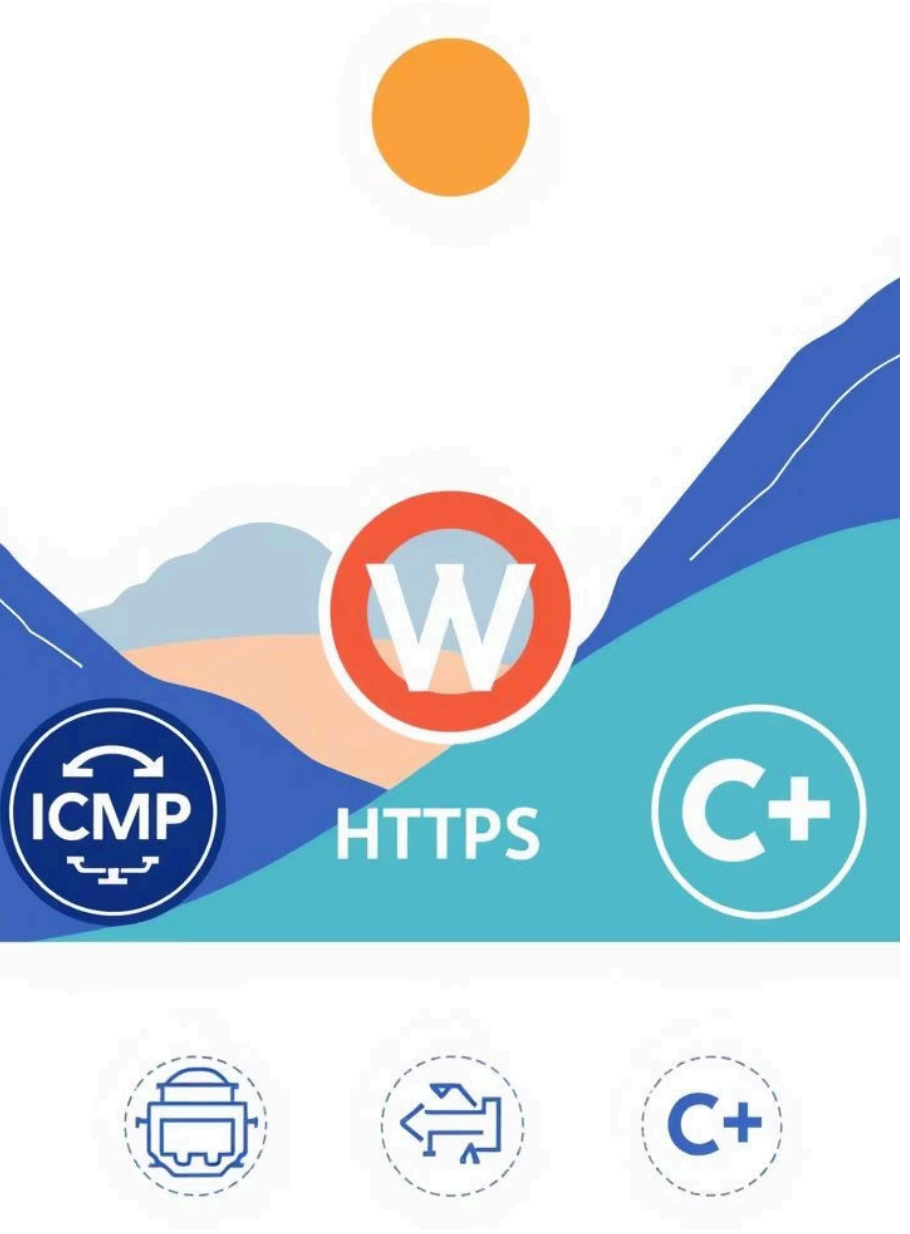
Хакатон: T1, PingTower

Существующие решения — это **overkill**

Prometheus, Zabbix, Grafana мощны, но сложны. Они требуют долгой настройки и мощных ресурсов. Часто их функционал избыточен для базового мониторинга доступности.

Нам нужен был **легкий, быстрый и специализированный инструмент**.





Производительность и простота через правильный выбор технологий

Гибкость проверок

Выбор протокола под задачу: **ICMP** для скорости сети и **HTTP/S** для веб-сервисов.

Производительность

Нагруженные модули на **C++** для сбора метрик.

Масштабируемость

Четкое разделение ответственности между микросервисами.

Архитектура: Микросервисы, gRPC и очереди

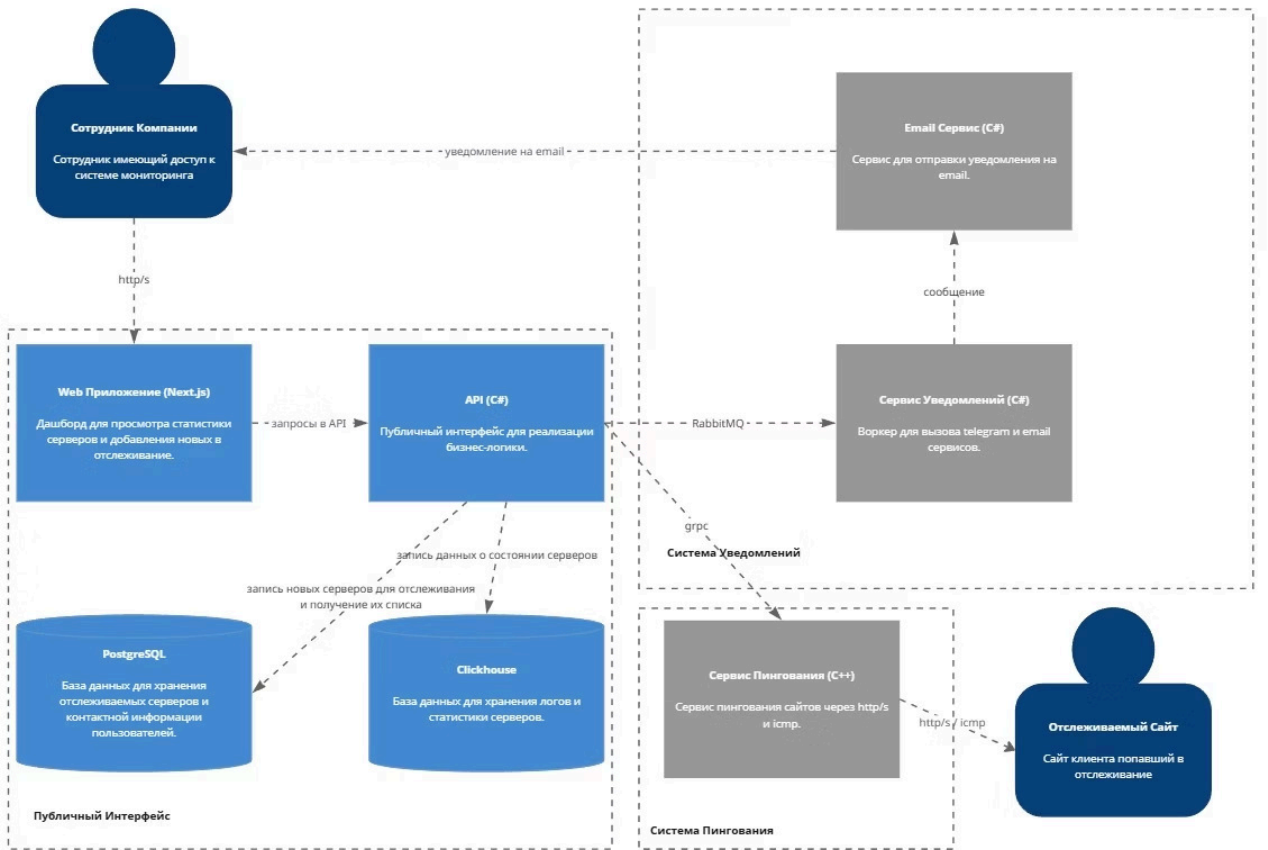


Диаграмма Контейнеров для Системы Мониторинга
Описание взаимодействия контейнеров Системы Мониторинга
Последнее обновление: 20.09.2025

- **Ping Sytem:** Сбор метрик (ICMP/HTTP), gRPC-связь с ядром.
- **WebAPI (Ядро):** Прием метрик, бизнес-логика, БД, RabbitMQ.
- **Notification System:** Асинхронная обработка очереди, отправка уведомлений.
- **Dashboard (Next.js):** Потребляет данные через REST API.
- **БД: PostgreSQL** (конфигурация), **ClickHouse** (метрики, >1M записей).

Почему мы выбрали именно этот стек?

C++ → gRPC → C#

Максимальная производительность сбора данных и эффективная сериализация.

PostgreSQL

OpenSource база данных.

ClickHouse

Идеален для временных рядов, молниеносные запросы к истории.

Next.js

SSR, готовый к продакшену фронтенд, быстрое получение данных через API.

Жизнь одного пинг-запроса

01

Расписание

C# API говорит C++ агенту: "Пингуй **example.com** каждые 30s".

02

Сбор

C++ агент выполняет HTTP-запрос, замеряет время, получает статус.

03

Транспорт

Данные летят в C# API по **gRPC**.

04

Сохранение

C# API пишет метрику в **ClickHouse**.

05

Оповещение

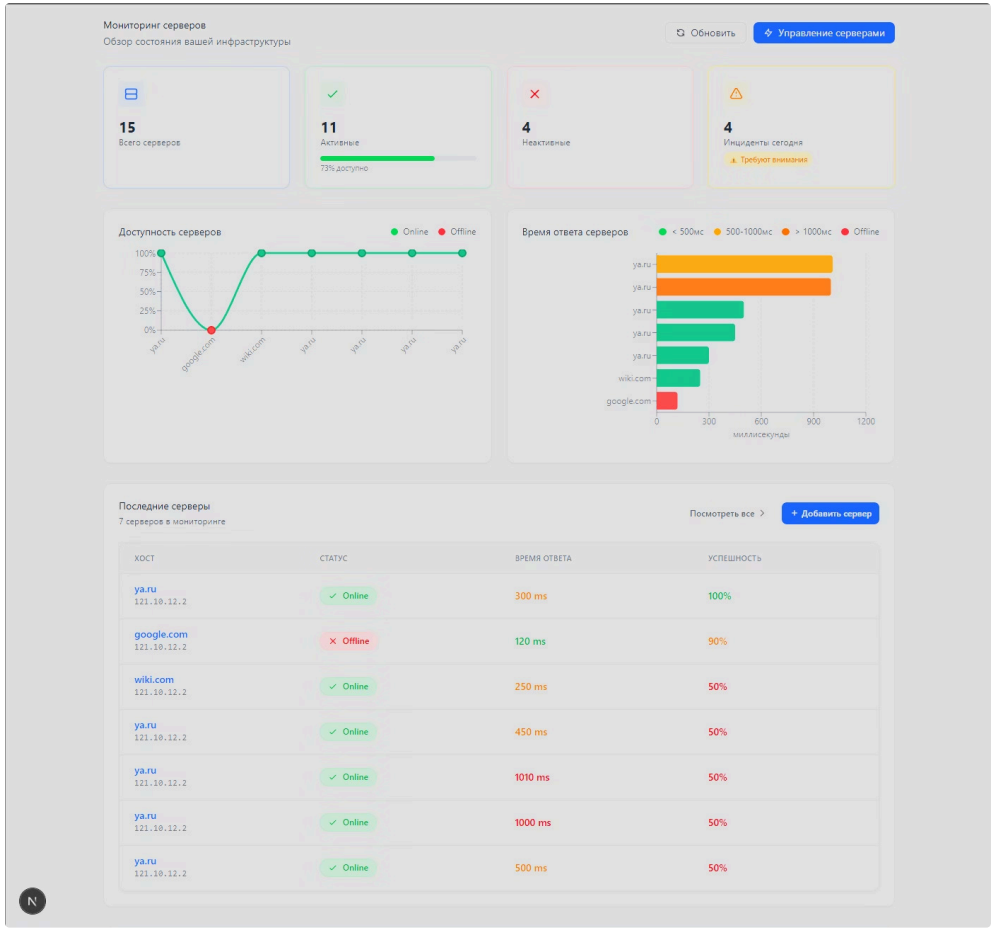
C# API кладет сообщение в **RabbitMQ**.

06

Уведомление

Сервис уведомлений забирает сообщение и шлет **email**.

Вся информация — в одном месте



- **Цветовая индикация:** Зеленый/Желтый/Красный для быстрой оценки статуса.
- **Графики времени отклика:** На основе данных из ClickHouse.
- **История инцидентов:** Когда, какой сервис и сколько времени был недоступен.
- **Uptime:** Расчет процента доступности за выбранный период.

Планы по развитию Dooka

→ Новые каналы уведомлений

Telegram, Slack Webhooks.

→ Расширенные проверки

SSL/TLS certificate expiration, проверка тела ответа (JSON Path).

→ Публичное REST API

Для интеграции в внешние системы и CI/CD.

→ Группировка уведомлений

Чтобы избежать "спама" при массовых сбоях.

Нас вдохновляют сложные задачи

Кирилл

Ping System C++

Семён

C# API, БД

Илья

Next.js Dashboard, UI/UX

Александр

DevOps

Владимир

Капитан, Архитектура

Спасибо за внимание!