

МИКРОСЕРВИСЫ

Когда и зачем

ОБО МНЕ



Алексей Кузнецов

- 6 лет опыта продакшен разработки
- последние 1.5 года - портал GeekBrains
- специализация: бекенд

ВНАЧАЛЕ БЫЛ МОНОЛИТ

ПЛЮСЫ

- скорость разработки
 - просто добавлять новый функционал
 - просто рефакторить и изменять код
 - просто развернуть приложение

МИНУСЫ

- растущая со временем сложность
 - только логическое разграничение функционала
 - протекающие абстракции
 - рефакторинг кода затруднен
- временные издержки
 - больше кода

МИКРОСЕРВИСЫ

ПЛЮСЫ

- жесткое разграничение функционала
 - каждый микросервис выполняет одну задачу
 - проще тестировать
 - проще поддерживать
- больше возможностей для масштабирования
- проще разрабатывать на больших проектах

МИНУСЫ

- организация взаимодействия между сервисами
 - описание протоколов до разработки
 - поддержка нескольких версий
- сложнее конфигурировать деплой
- сложнее настроить мониторинг и сбор логов
- больше точек отказа

КОГДА ИСПОЛЬЗОВАТЬ МИКРОСЕРВИСЫ

- ожидается большая нагрузка
- необходимо разрабатывать и обновлять части приложения независимо друг от друга
- разработчики хотят спать по ночам

А ЕСЛИ НУЖНО ЗДЕСЬ И СЕЙЧАС

Пишем монолит. По возможности организуем код так, чтобы компоненты системы можно было достаточно легко отделить в будущем.

РАСПИЛИМ МОНОЛИТ

Кандидаты на вынос в микросервисы

- сервис авторизации
- сервисы коммуникации (чат)
- сервисы аналитики/агрегации данных

Перед тем, как отделять сервис, подумайте, что вам это даст.

Уменьшая сложность компонентов вы увеличиваете сложность взаимодействия между ними.

КАК ЗАПУСКАТЬ

Docker, Docker или Docker?

КАК ОБЩАТЬСЯ МЕЖДУ СОБОЙ

- API (json+rest)
- RPC (gRPC + protobuf)
- Message brokers
 - RabbitMQ
 - Apache Kafka
 - Redis?
- Auth (JWT)

А ЧТО НАСЧЕТ VCS

- каждый сервис в своем репозитории
- монорепозиторий с набором сервисов
- микросервисы в рамках одного приложения (Umbrella application (elixir), Hanami (ruby))

**МИКРОСЕРВИСЫ - НЕ
СЕРЕБРЯНАЯ ПУЛЯ**

КОНТАКТЫ

<https://github.com/semenovDL>

<http://telegram.me/semenovDL>

ССЫЛКИ

DOCKER

Docker

ОЧЕРЕДИ

RabbitMQ

Apache Kafka

Redis

Use Redis as message broker

АВТОРИЗАЦИЯ

JWT

UMBRELLA

Umbrella in Elixir

Umbrella in Hanami

SYSTEM DESIGN

Гайд по дизайну масштабируемых систем

en ru