IT's TINKOFF

Монолит и Микросервисы

Переходим с микросервисов на монолит

Данил Ахтаров

18.10.2022

Скачать
презентацию

Данил Ахтаров

Архитектор

github.com/daxartio

t.me/daxtar

Наш путь

До 2019

Монолит

- 1 процесс (нельзя масштабировать)

Наш путь

До 2019

Монолит

- 1 процесс (нельзя масштабировать)

2022

Микросервисы

- Масштабируемая система

Наш путь

До 2019

Монолит

- 1 процесс (нельзя масштабировать)

2022

Микросервисы

- Масштабируемая система

2023

Монолит

- Опять?

Agenda

- Микросервисы
- Проблемы одной команды
- Почему мы выбрали монолит
- Выводы

01

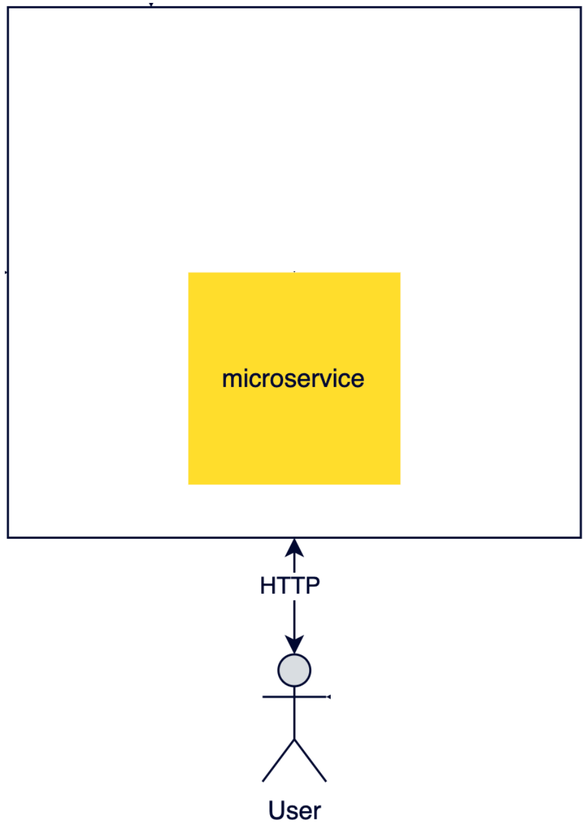
Микросервисы

1. Что это такое
2. Архитектура
3. Плюсы и Минусы

Как вы думаете, что такое микросервис?

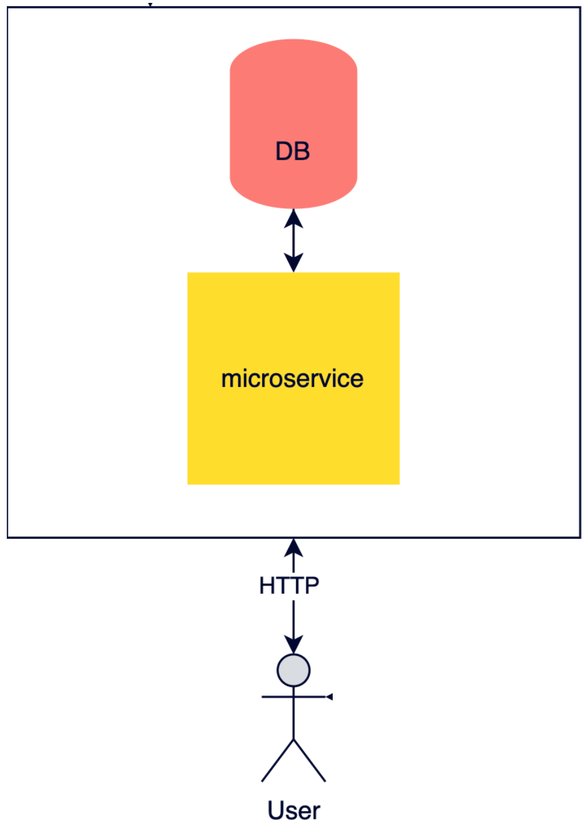
Микросервис

- Единица проекта, которая отвечает за свою бизнес логику



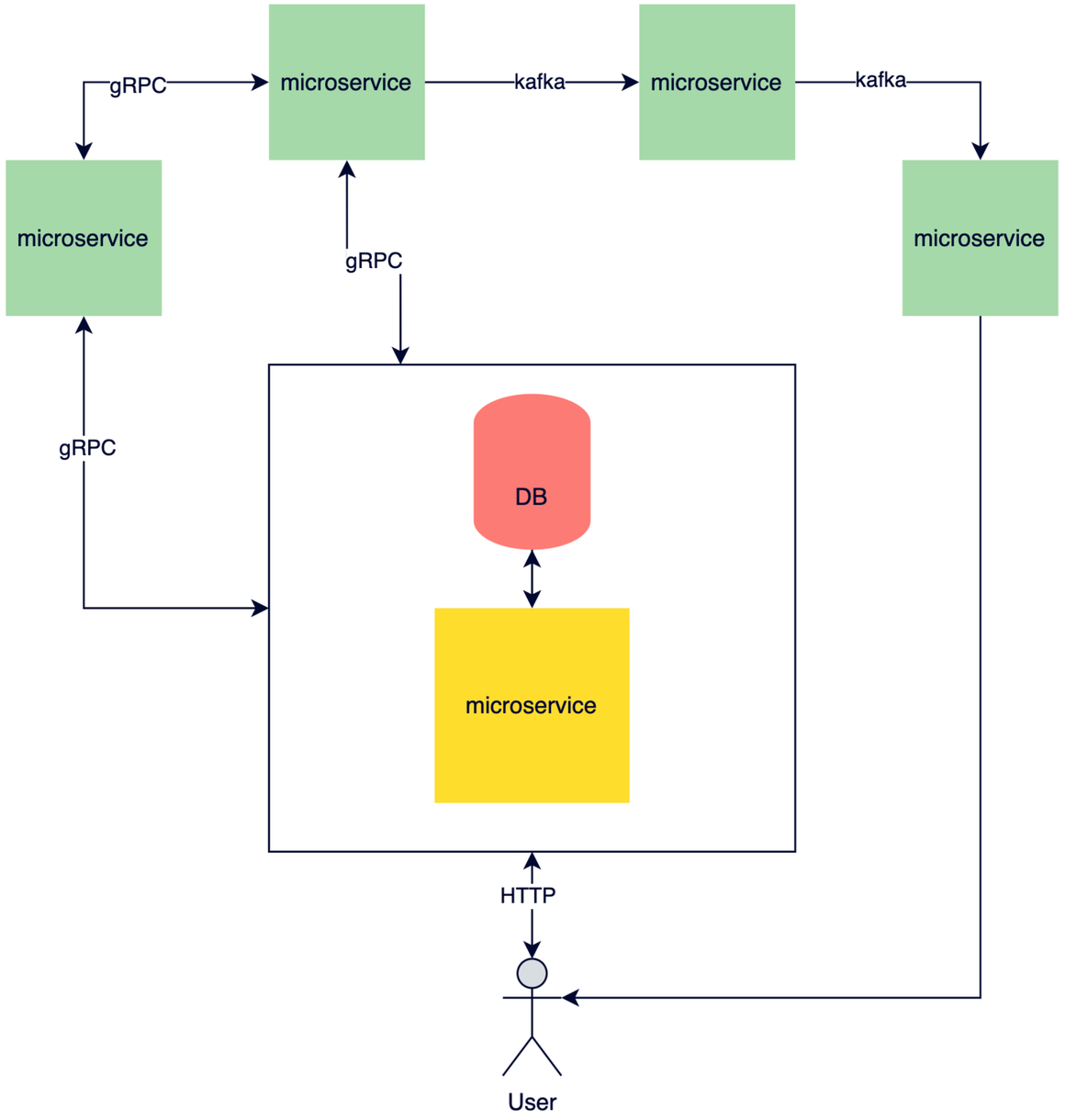
Микросервис

- Единица проекта, которая отвечает за свою бизнес логику
- Своя база данных

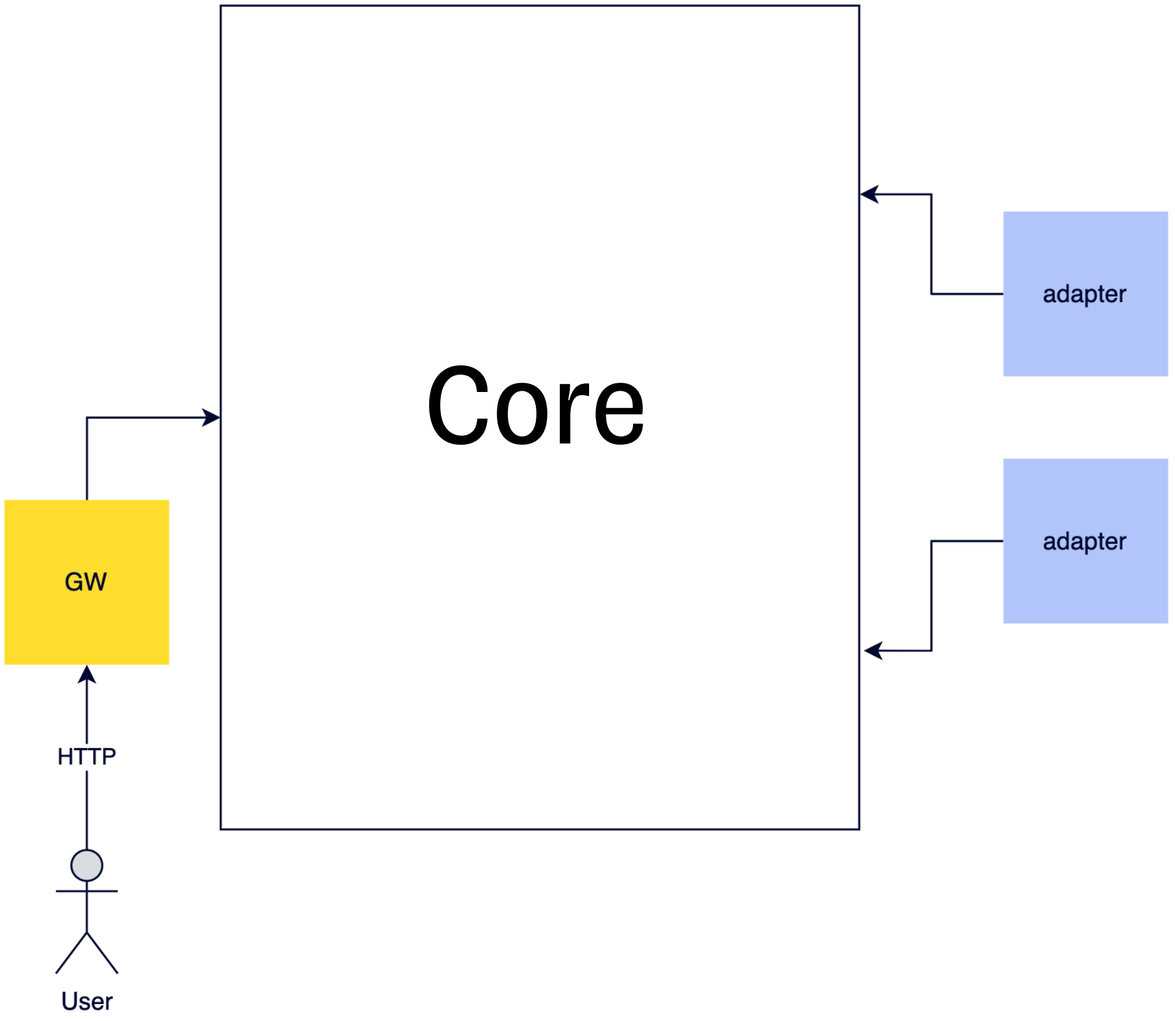


Микросервис

- Единица проекта, которая отвечает за свою бизнес логику
- Своя база данных
- API (Application Programming Interface)

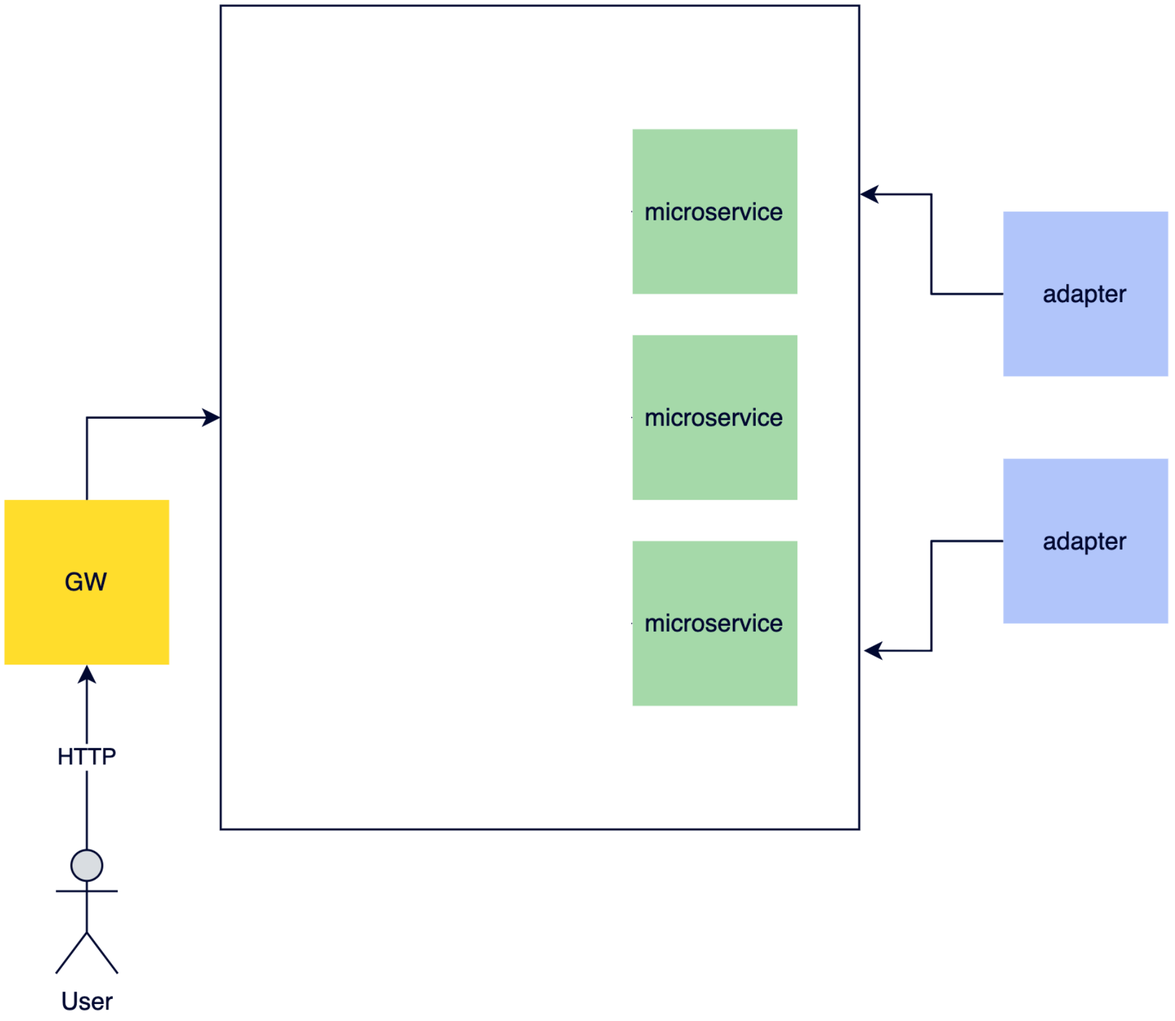


Архитектура



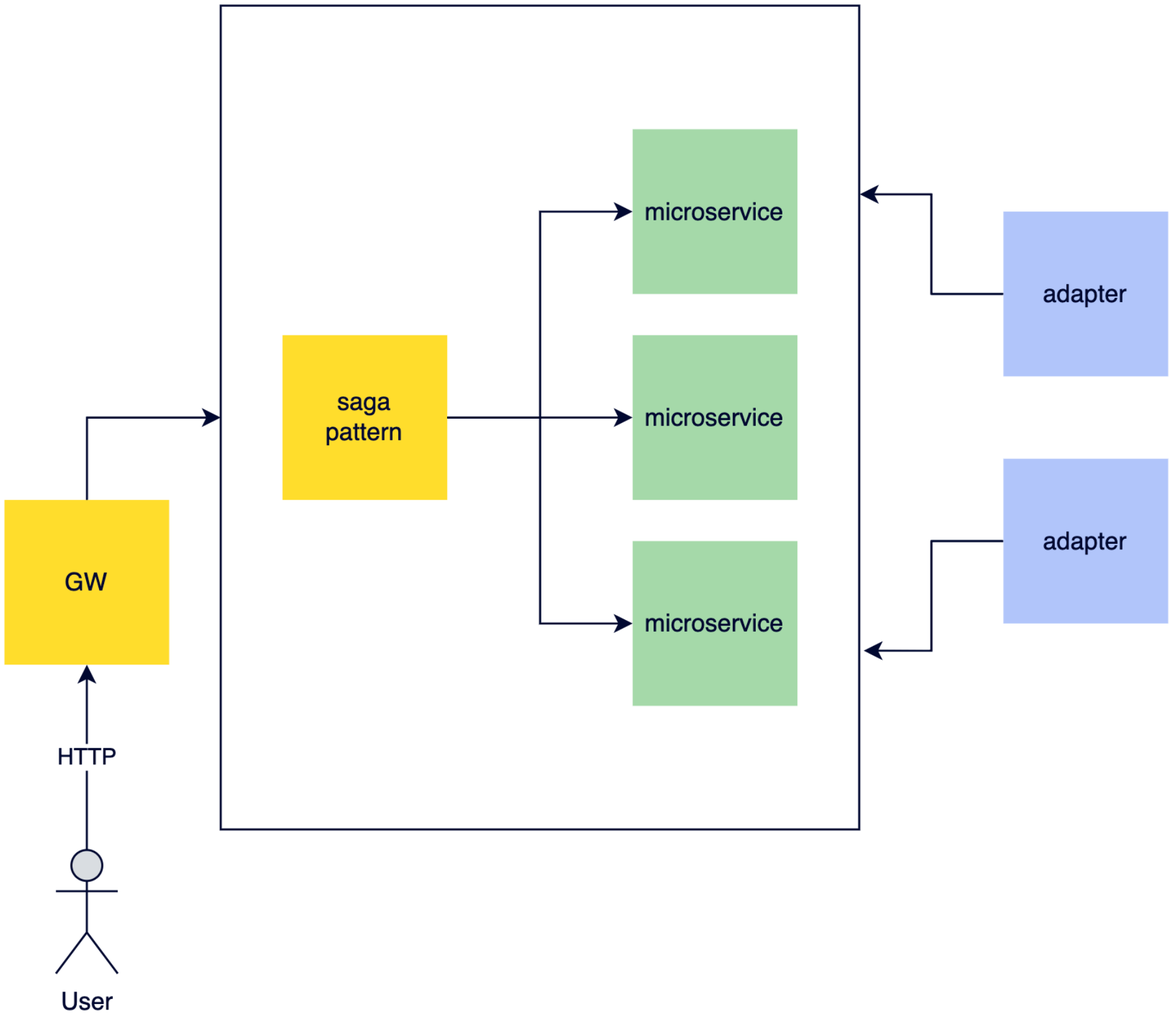
- Гексагональная архитектура
  - Порты — это интерфейсы нашего приложения
  - Адаптеры — реализация наших портов

Архитектура



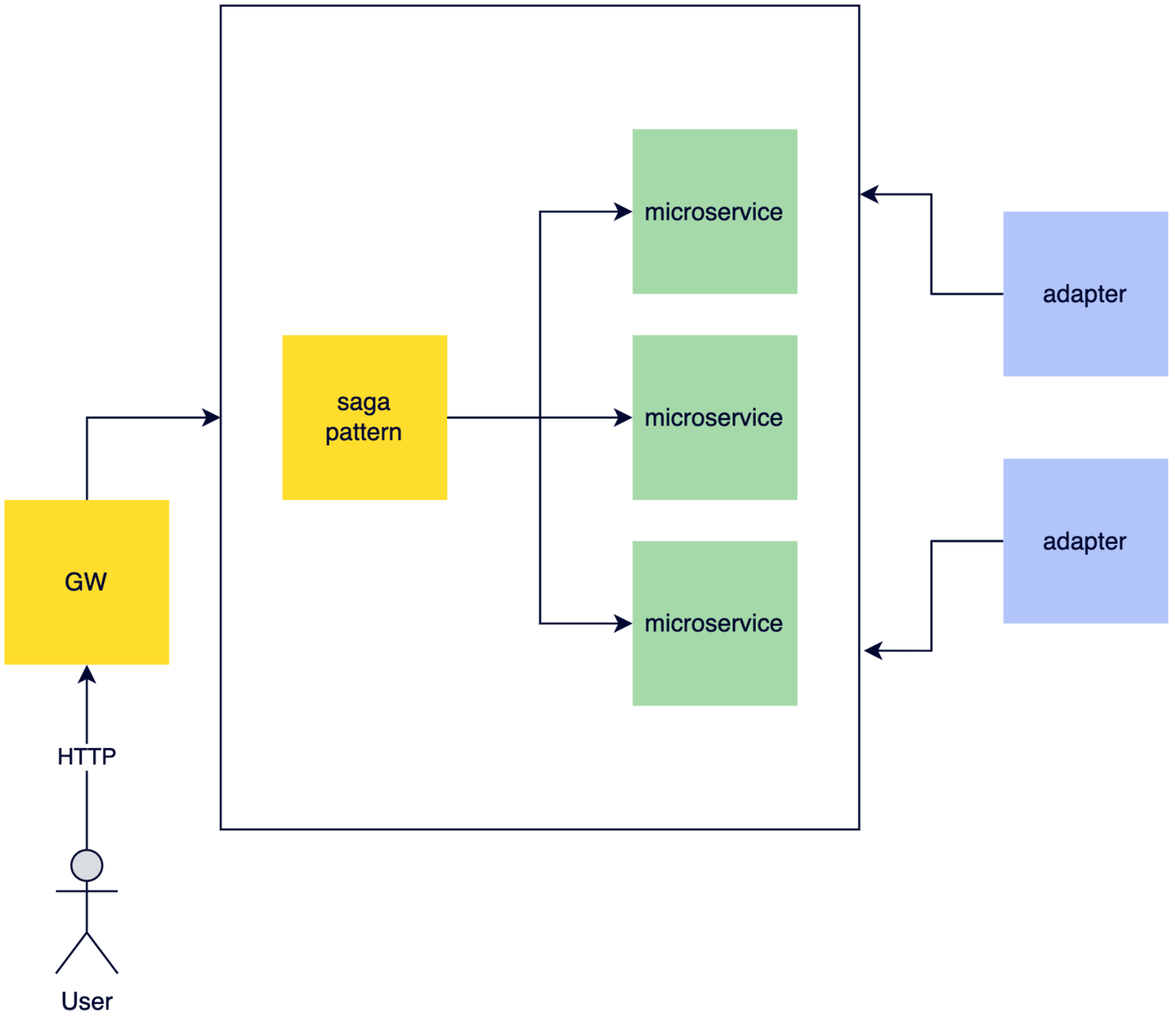
- Гексагональная архитектура
  - Порты — это интерфейсы нашего приложения
  - Адаптеры — реализация наших портов
- DDD (Domain Driven Design)
  - Программные абстракции
  - Предметная область

Архитектура



- Гексагональная архитектура
  - Порты — это интерфейсы нашего приложения
  - Адаптеры — реализация наших портов
- DDD (Domain Driven Design)
  - Программные абстракции
  - Предметная область
- Saga Pattern
  - Оркестрация

Архитектура



- Гексагональная архитектура
  - Порты — это интерфейсы нашего приложения
  - Адаптеры — реализация наших портов
- DDD (Domain Driven Design)
  - Программные абстракции
  - Предметная область
- Saga Pattern
  - Оркестрация
  - https://microservices.io/patterns/data/saga.html

Сравнение старой и новой архитектуры

Монолит

- 1 процесс (нельзя масштабировать)
- 50 Гб память
- PyPy

Микросервисы

- 30 микросервисов
- 660 контейнеров
- 300 Гб
- 50 хостов
- CPython
- Накладные расходы на сеть
- Latency

Что у нас получилось с микросервисами

Плюсы

- Масштабирование

Минусы

- 300 Гб и 50 хостов

Что у нас получилось с микросервисами

Плюсы

- Масштабирование
- Каждый сервис просто поддерживать

Минусы

- 300 Гб и 50 хостов
- Задача требует изменения в нескольких сервисах

Что у нас получилось с микросервисами

Плюсы

- Масштабирование
- Каждый сервис просто поддерживать
- Можно использовать разные технологии

Минусы

- 300 Гб и 50 хостов
- Задача требует изменения в нескольких сервисах

Что у нас получилось с микросервисами

Плюсы

- Масштабирование
- Каждый сервис просто поддерживать
- Можно использовать разные технологии

Минусы

- 300 Гб и 50 хостов
- Задача требует изменения в нескольких сервисах
- Распределенные транзакции

Проблемы с микросервисами

Проблемы с микросервисами

1

Уход в крайность

Проблемы с микросервисами

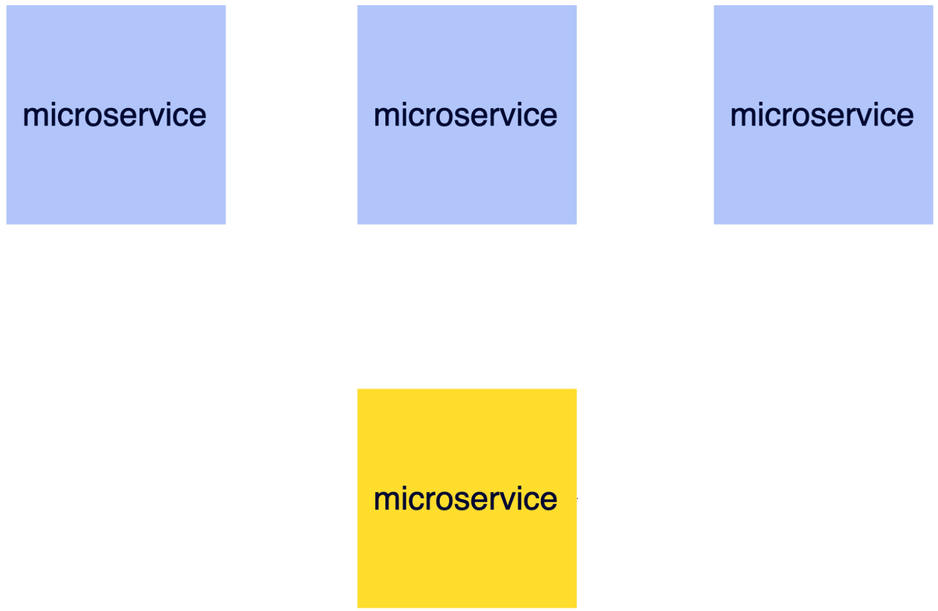
1

Уход в крайность

Проблемы с микросервисами

1

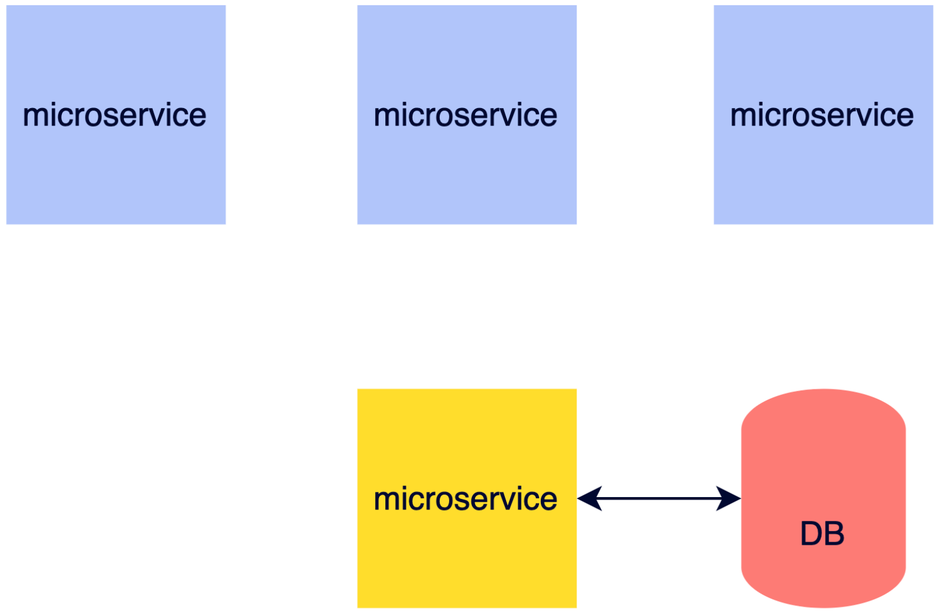
Уход в крайность



Проблемы с микросервисами

1

Уход в крайность



Отдельный микросервис для кнопки со своей базой и на отдельном хосте

Проблемы с микросервисами

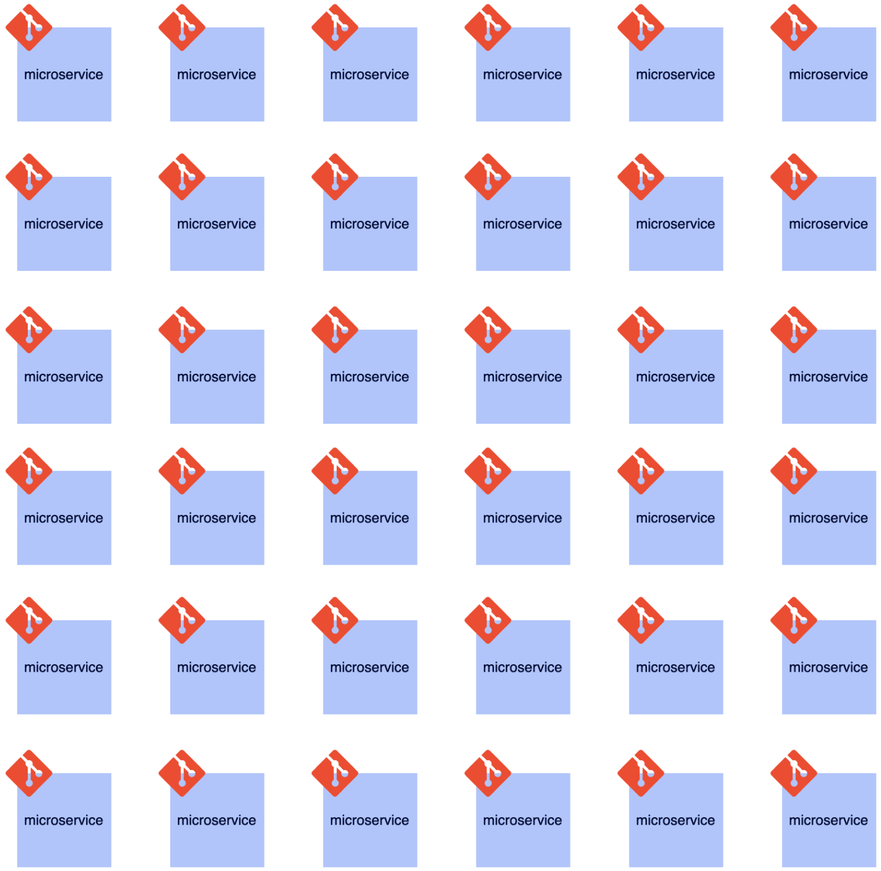
2

Проект сложно поддерживать одной команде

Проблемы с микросервисами

2

Проект сложно поддерживать одной команде



Проблемы с микросервисами

3

Увеличивается time to market

Проблемы с микросервисами

4

DevOps / SRE

Проблемы с микросервисами

4

DevOps / SRE

- Сделать так, чтобы это работало
- Логи
- Мониторинг
- Deploy

Проблемы с микросервисами

- Уход в крайность
- Проект сложно поддерживать одной команде
- Увеличивается time to market
- DevOps / SRE

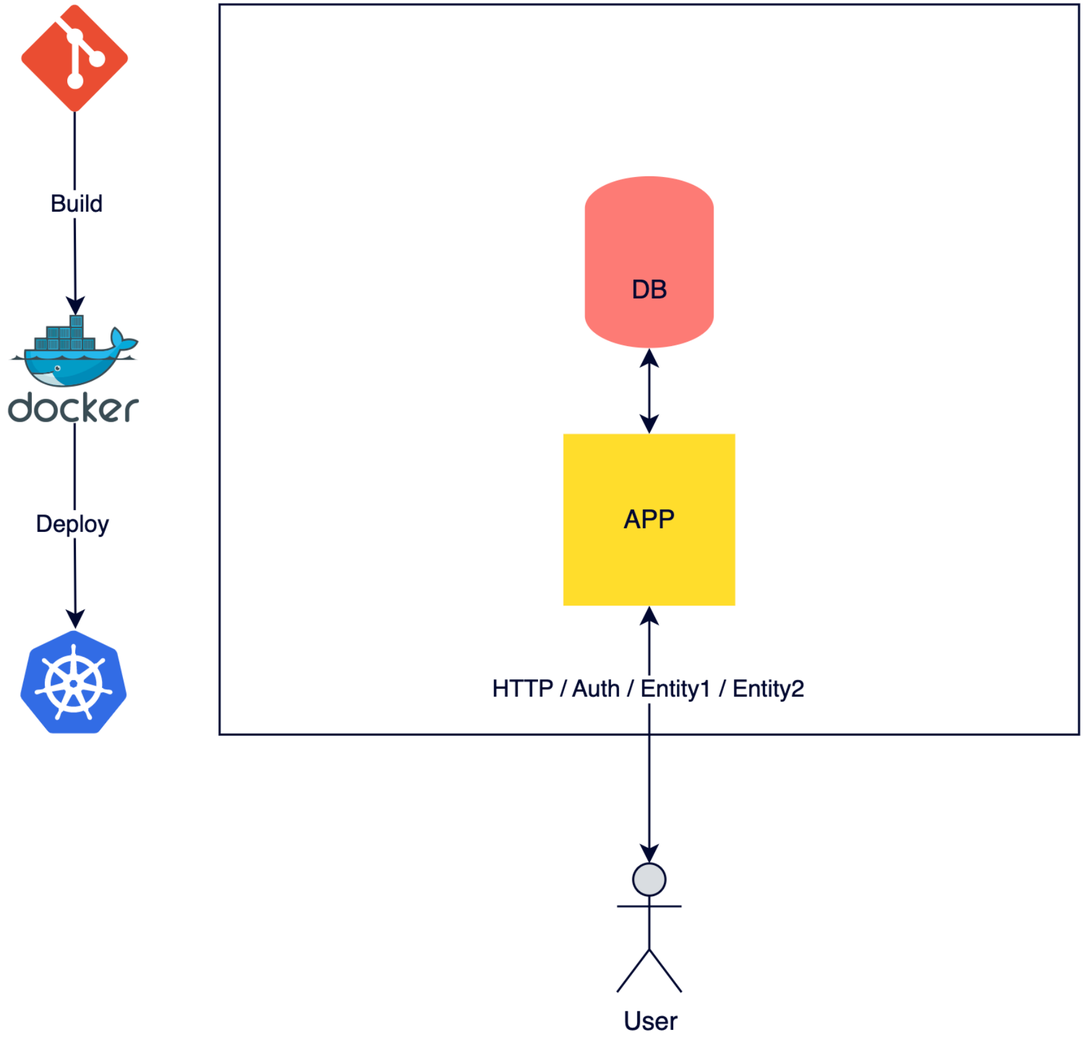
02

Монолит

1. Что это такое
2. Как мы видим наш монолит
3. Выводы

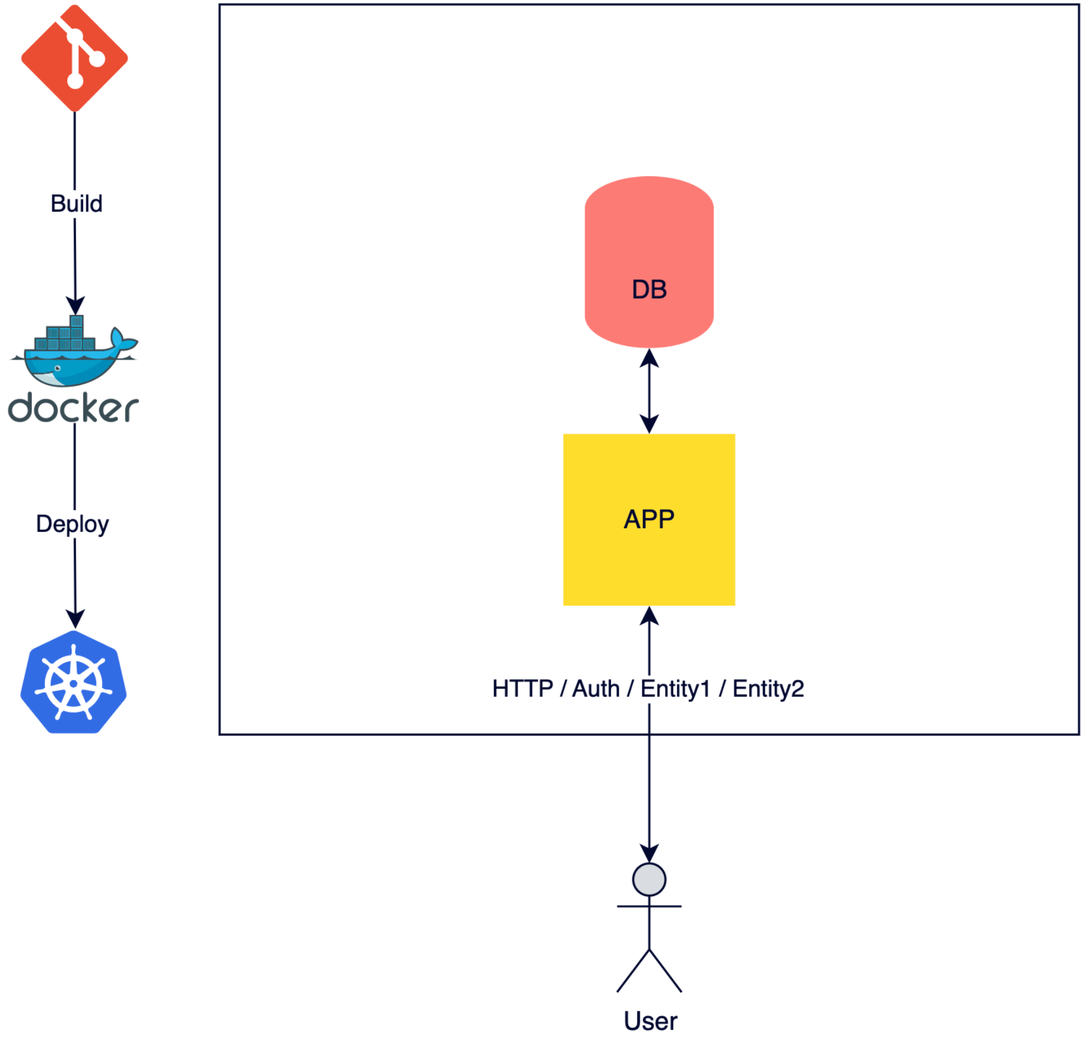
Монолит

- Монолитное приложение с одной кодовой базой
  - Консистентный код
  - Один репозиторий
  - Деплой приложения целиком



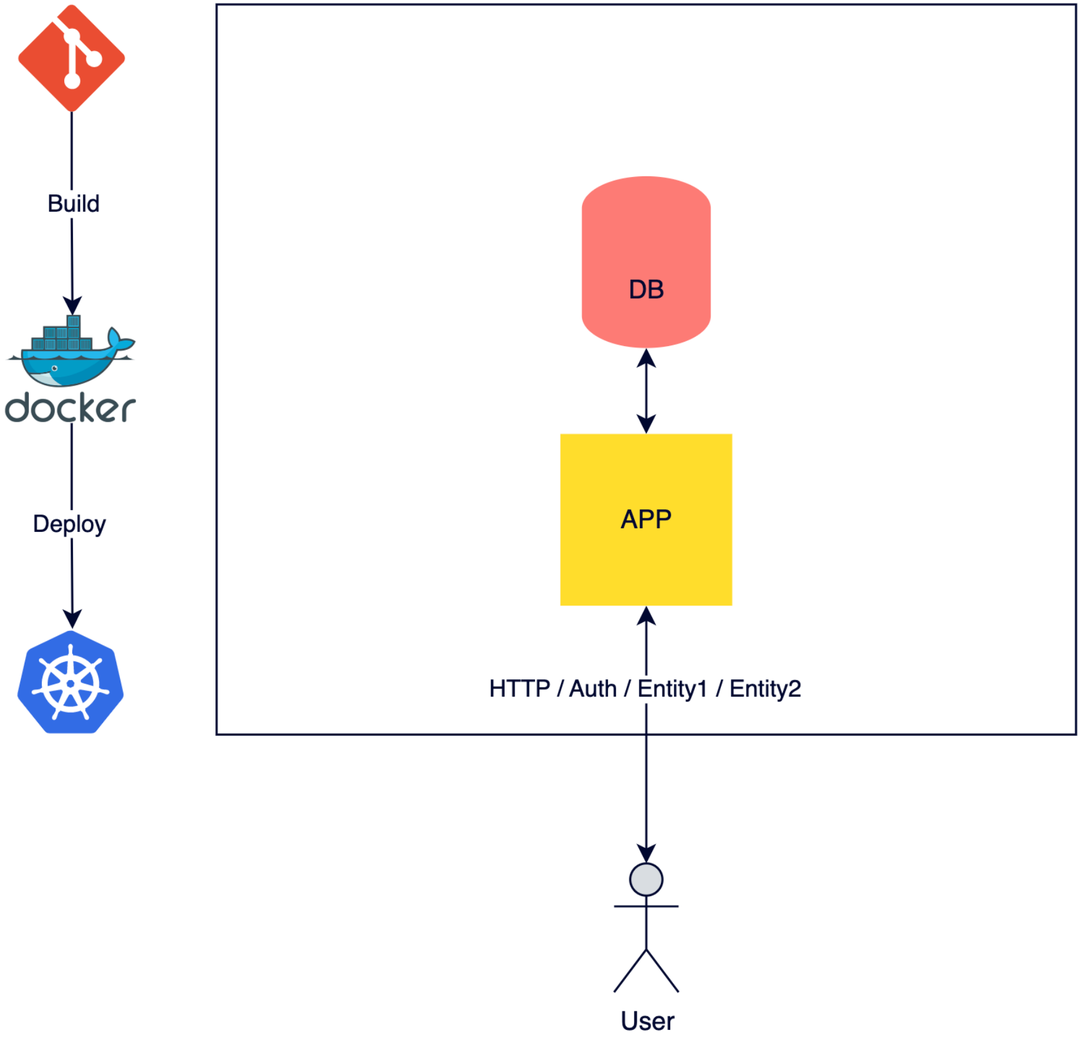
Монолит

- Монолитное приложение с одной кодовой базой
  - Консистентный код
  - Один репозиторий
  - Деплой приложения целиком
- Самодостаточное приложение

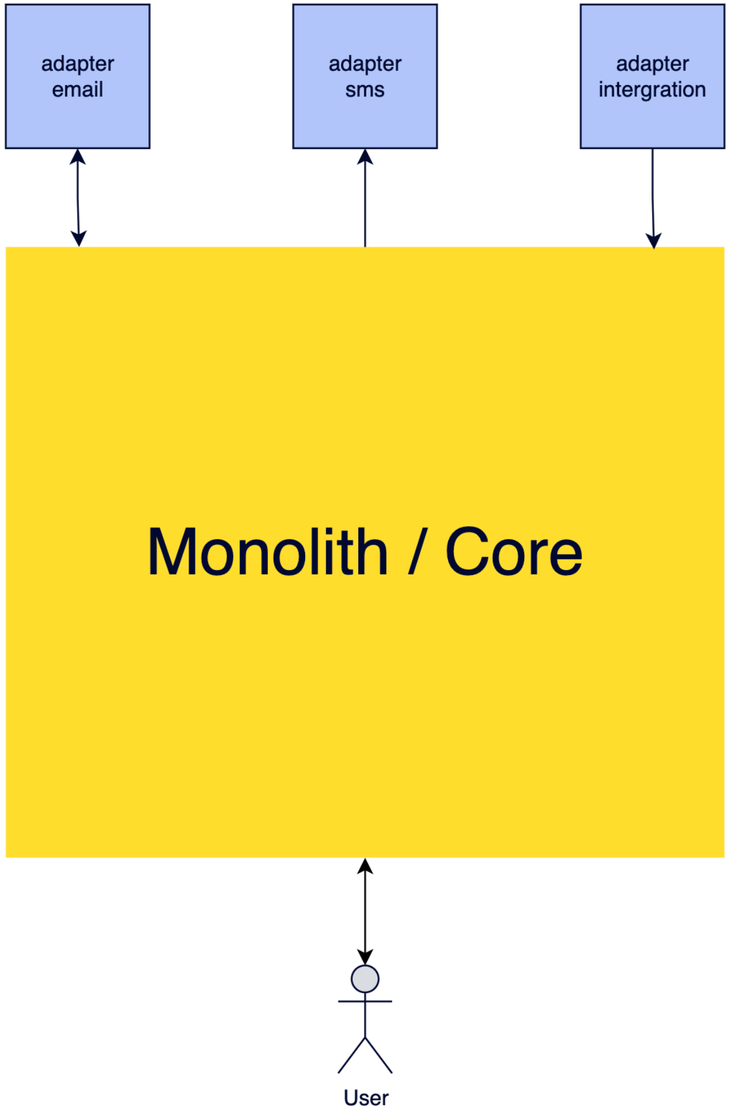


Монолит

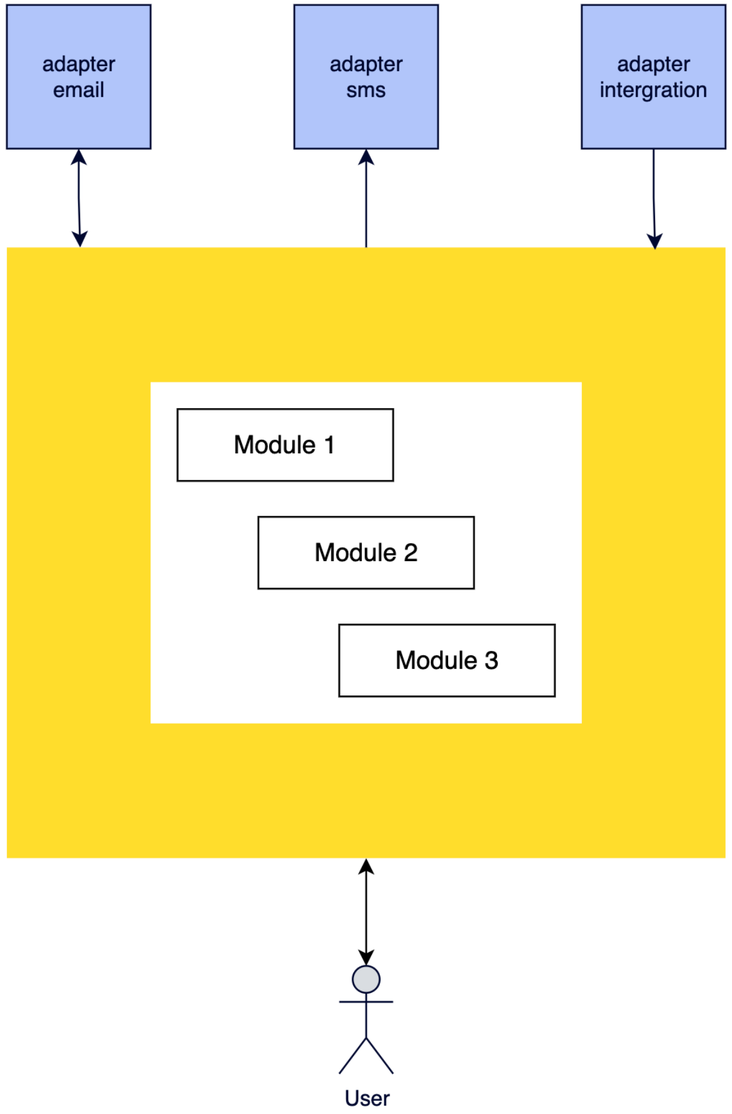
- Монолитное приложение с одной кодовой базой
  - Консистентный код
  - Один репозиторий
  - Деплой приложения целиком
- Самодостаточное приложение
- Доменом является ваш продукт целиком



Правильный монолит



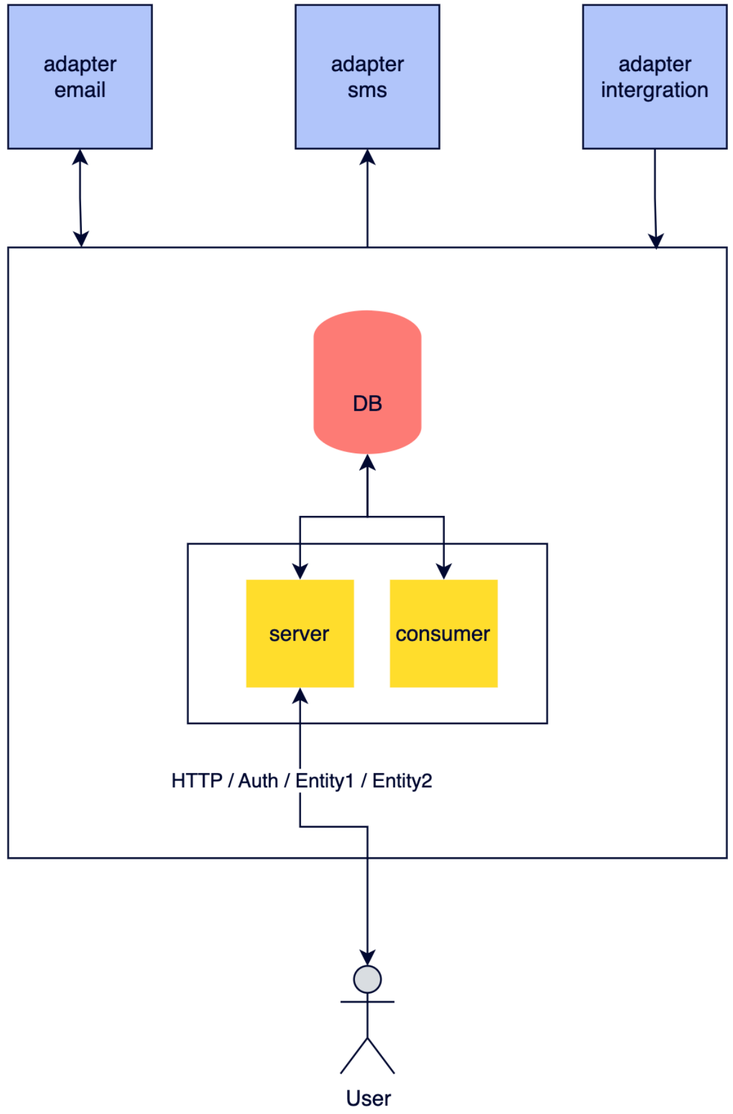
- Гексагональная архитектура
  - Монолит отвечает за БЛ продукта
  - Адаптеры выделены в отдельные сервисы



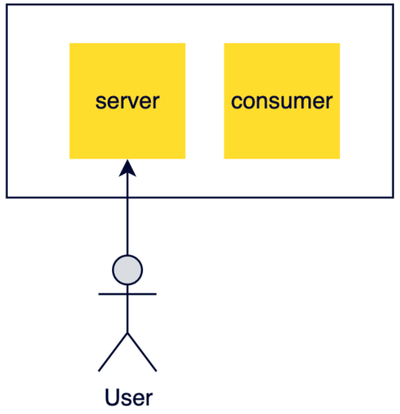
Правильный монолит

- Гексагональная архитектура
  - Монолит отвечает за БЛ продукта
  - Адаптеры выделены в отдельные сервисы
- DDD (Domain Driven Design)
  - Выделять модули на уровне кода

Правильный монолит

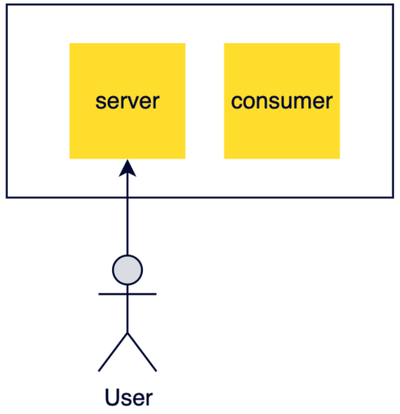


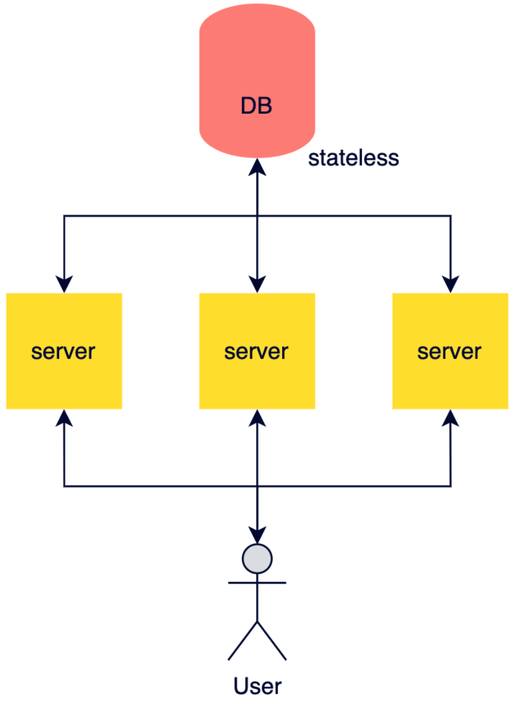
- Гексагональная архитектура
  - Монолит отвечает за БЛ продукта
  - Адаптеры выделены в отдельные сервисы
- DDD (Domain Driven Design)
  - Выделять модули на уровне кода
- Разные типы нагрузки должны быть разделены по ресурсам



Масштабирование

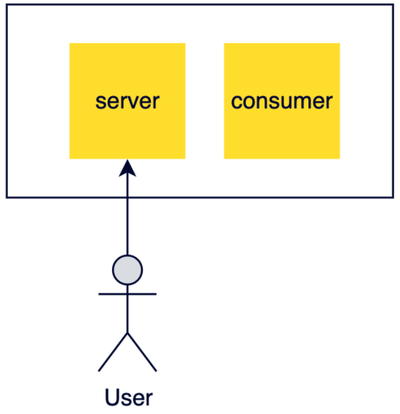
- Разные типы нагрузки должны быть разделены по ресурсам

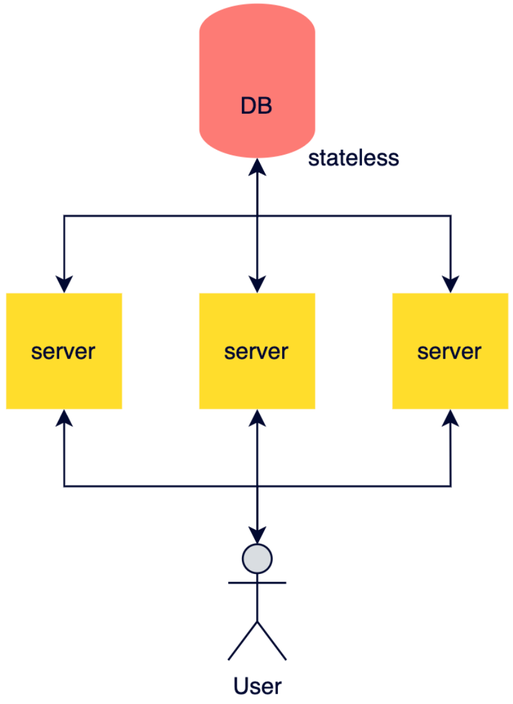


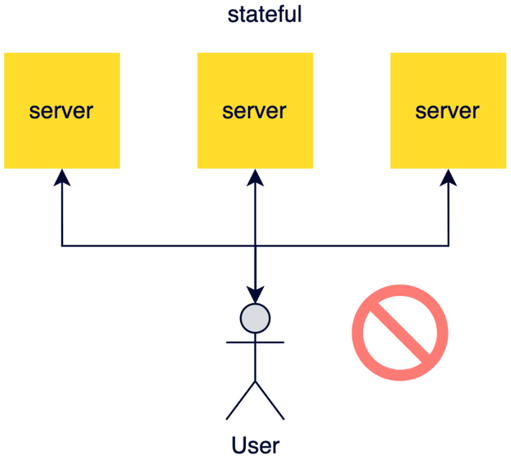


Масштабирование

- Разные типы нагрузки должны быть разделены по ресурсам
- Делать stateless приложение

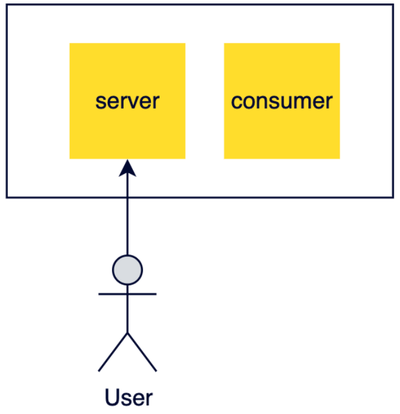


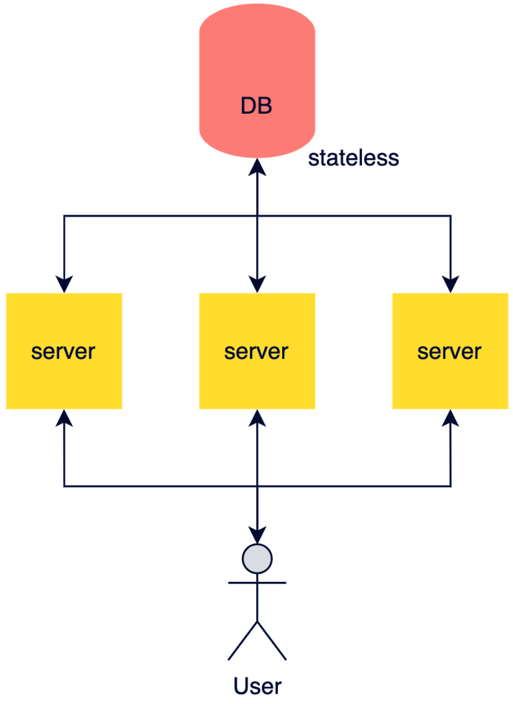


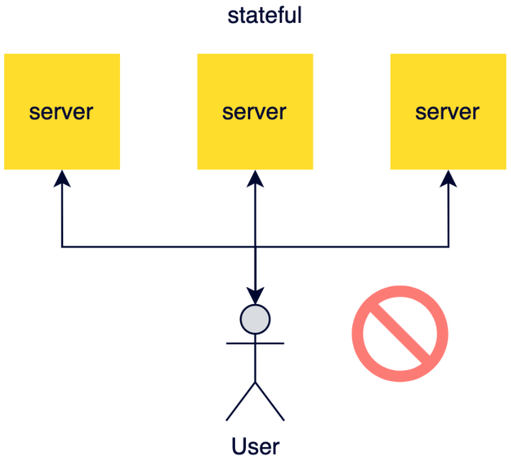


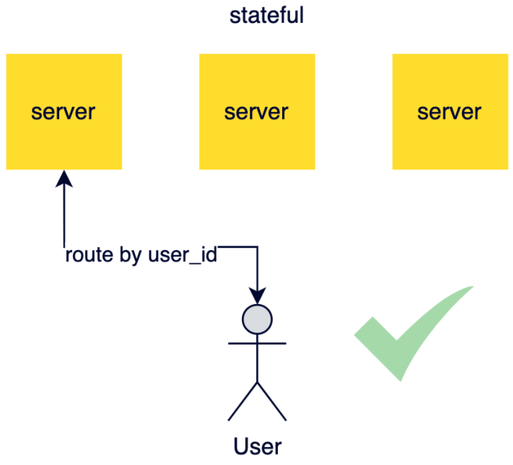
Масштабирование

- Разные типы нагрузки должны быть разделены по ресурсам
- Делать stateless приложение









Масштабирование

- Разные типы нагрузки должны быть разделены по ресурсам
- Делать stateless приложение
- Применять шардирование

Выводы

Переход с микросервисов на монолит

Простота

Когнитивная нагрузка с разработчика уменьшена

Контроль за качеством

Требуется “жесткое” внедрение практик проектирования кода

Масштабирование

Грамотная архитектура поможет избежать минусов монолита

Спасибо за внимание

IT's TINKOFF

Ахтаров Данил | Архитектор

github.com/daxartio

t.me/daxtar