SolarLab>_

Банально про архитектуру и микросервисы

Алексей Андреев https://t.me/AlekseyAndreev1984

Содержание

SolarLab>_

Почему банально(история из жизни)

Введение в Apache Kafka

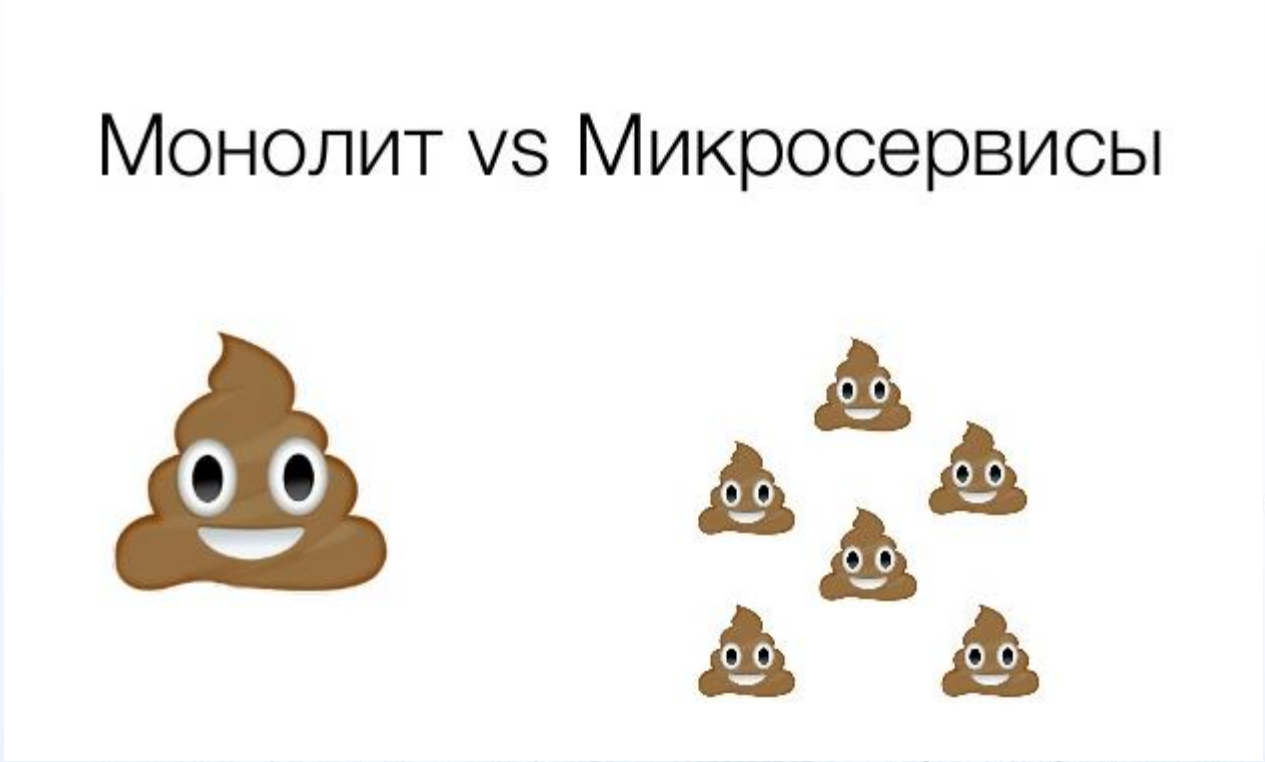
А давайте сделаем MVP

MVP – это minimum VIABLE product: минимально жизнеспособный продукт

SolarLab>_

Микросервисы - как я их вижу

Микросервисы - как я их вижу



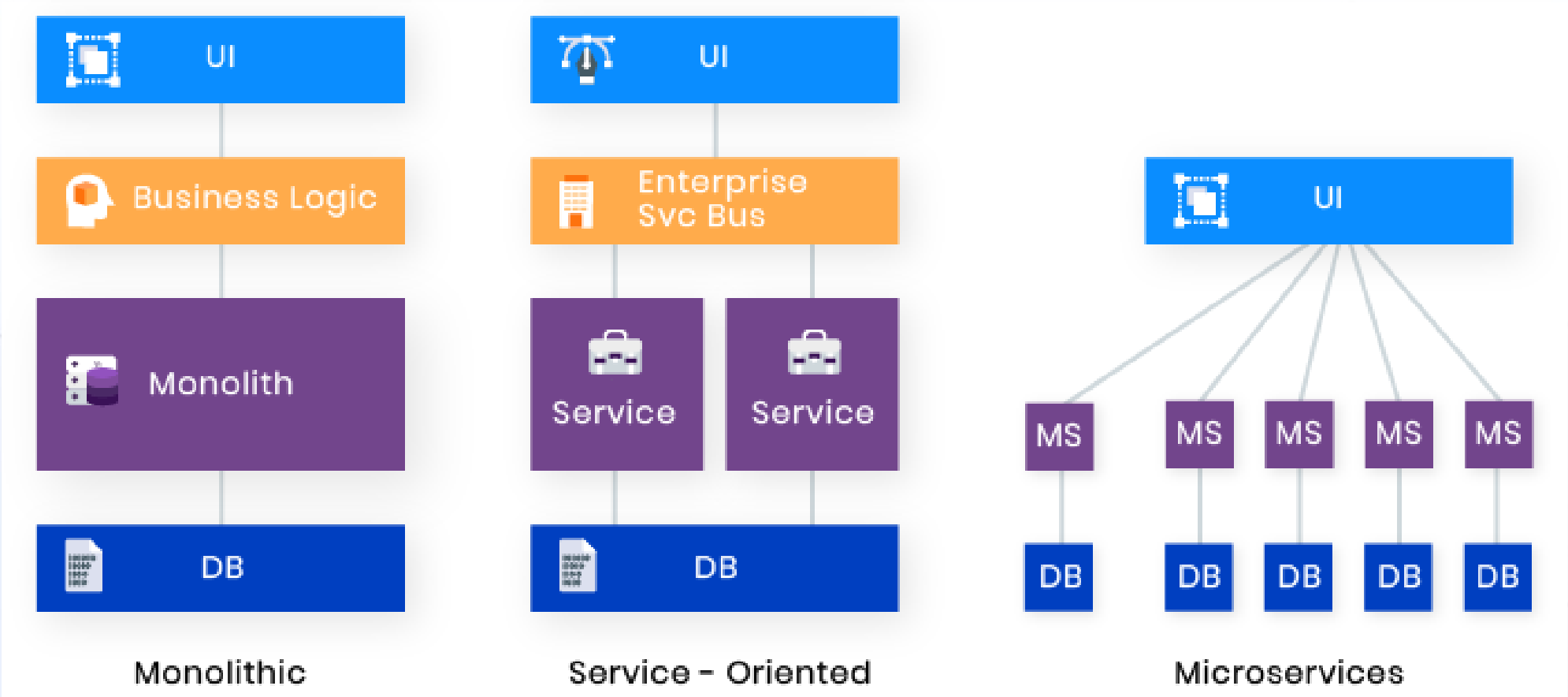
Характеристики микросервисов

- Архитектура микросервисов состоит из небольших, независимых и слабо связанных между собой служб.
- Каждая служба является отдельной базой кода, которой может управлять небольшая команда разработчиков.
- Службы можно развертывать независимо друг от друга. Разработчики могут обновлять существующую службу без повторной сборки и повторного развертывания всего приложения.
- Службы отвечают за сохранение собственных данных или внешнего состояния. В этом состоит отличие от традиционной модели, в которой сохранение данных обрабатывается на отдельном уровне.
- Службы взаимодействуют между собой с помощью четко определенных API-интерфейсов. Сведения о внутренней реализации каждой службы скрыты от других служб.
- Службы не должны(но могут) совместно использовать один и тот же стек технологий, библиотеки или платформы.

SolarLab>_

Отличие микросевисов от SOA

Отличие микросевисов от SOA



Отличие микросевисов от SOA

- **Микросервисы** — это эволюция SOA, но с упором на **минимализм, автономию сервисов** и **децентрализацию**.
- **SOA** — более «тяжелый» подход, ориентированный на **корпоративную интеграцию** и стандартизацию.

Выбор зависит от контекста: микросервисы подходят для динамичных проектов, SOA — для интеграции сложных legacy-систем.

SolarLab>_

Советы по ведению микросервисов

Советы по ведению микросервисов

- Если у вас есть несколько сервисов на одной и той же технологии(например API .net), тогда выгодно иметь готовый шаблон сервиса, который уже содержит в себе всю необходимую информацию для старта,чтобы не писать совсем с нуля
- Если у вас код повторяется – выносите в пакеты
- На каждый сервис своя БД. Не ходите к соседним сервисам за данными. Это уже SOA
- Сделайте начальное создание сервиса(его билд + деплой + тесты) в едином стиле и как можно более безболезненно
- Если микросервисы ходят к друг другу за данными – то обычно это вырождается в лапшисервисы
- Избегать лапши сервисов(Service Spaghetti)

SolarLab>_

Вопросы?

SolarLab>_

Выводы

Выводы

- Кафкой можно пользоваться
- Есть и другие продукты
- Кафка не является серебряной пулей и решением всех проблем, но знать её полезно
- Кафка умеет работать с огромными потоками данных

SolarLab>_

Ссылки

Ссылки

- https://github.com/confluentinc/confluent-kafka-dotnet
- https://kafka.apache.org/
- https://youtu.be/ghKnX5fuW5s?si=UWTF6ptQgirZn_6d
- https://www.youtube.com/watch?v=xrYefL1YlAo&list=PLmW6i9wxgkBzPZkYVQAniALZE0roLuERO&index=23&t=588s
- https://www.youtube.com/watch?v=-AZOi3kP9Js&list=PLmW6i9wxgkBzPZkYVQAniALZE0roLuERO&index=24&t=1374s
- Effective Kafka: A Hands-on Guide to Building Robust and Scalable Event-Driven Applications https://apachekafkabook.com/
- Apache Kafka. Потоковая обработка и анализ данных Бестселлеры O'Reilly https://www.piter.com/collection/all/product/apache-kafka-potokovaya-obrabotka-i-analiz-dannyh
- https://habr.com/ru/articles/270339/
- https://habr.com/ru/companies/slurm/articles/550934/
- **Битва брокеров сообщений**

  https://habr.com/ru/companies/yandex_praktikum/articles/700608/

Вопросы?

SolarLab>_

Спасибо за внимание