СБЕР УНИВЕРСИТЕТ

Сетевое взаимодействие. Распределенные системы.

Trenkunov Anton

План занятия

1. Микросервисная и монолитная архитектура
2. Синхронное взаимодействие через REST, SOAP
3. HATEOAS подход
4. Асинхронное взаимодействие через Message Broker. Использование Apache Kafka

Стандартный процесс разработки

- Начинается с маленького приложения, которое постепенно обрастает функционалом.
- Чтобы ускорить процесс разработки, команда разрастается.
- Со временем система разрастается до таких размеров, что не все в команде понимают как работает приложение в целом.
- Процесс разработки замедляется из-за сильной связанности приложения. Изменения в одной части приложения влекут изменения в других частях приложения. Часто это становится источником ошибок.
- Время между релизами увеличивается вплоть до нескольких релизов в месяц.

Монолит или микросервисы?

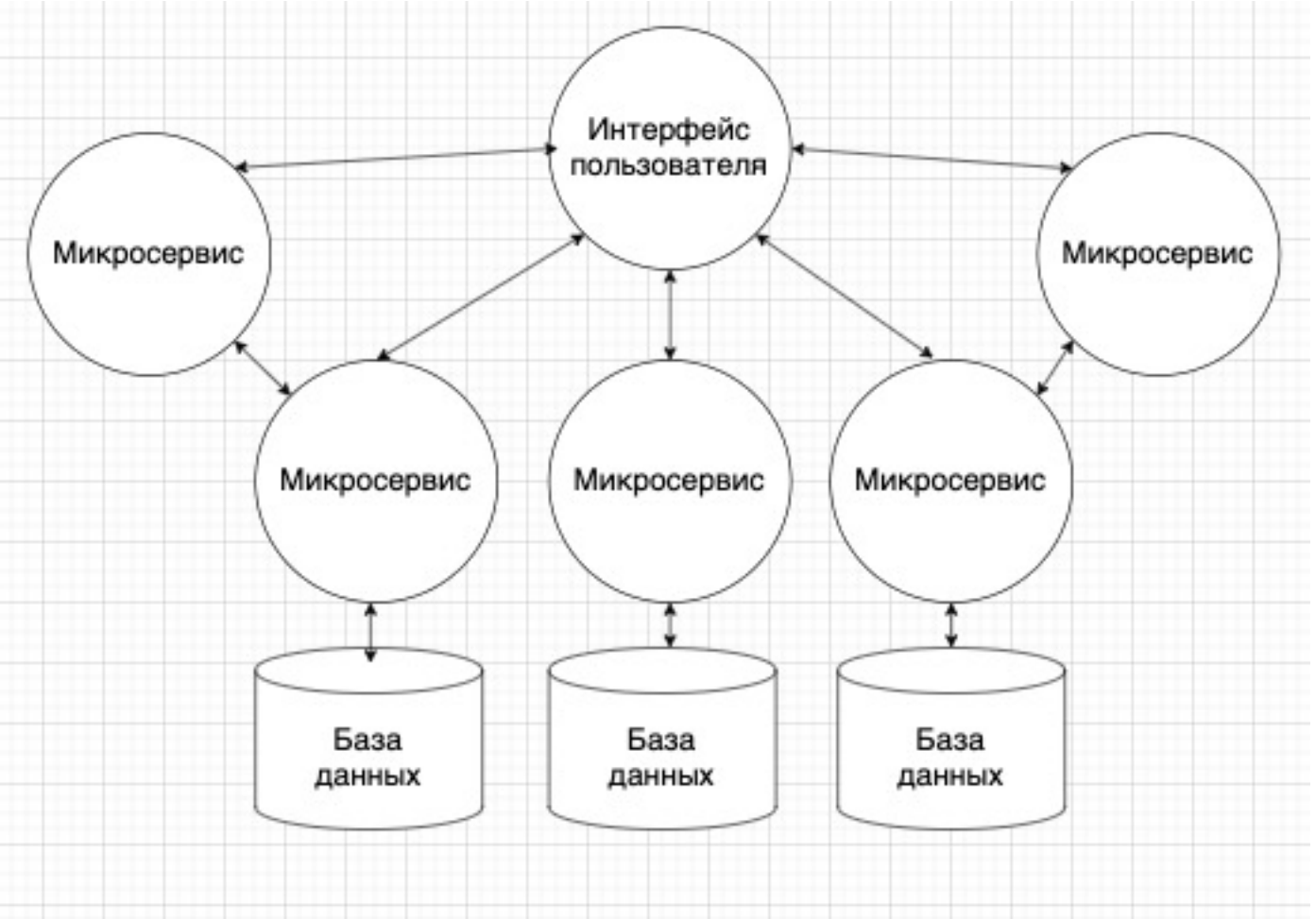
- Нет ничего плохого в использовании того или иного подхода.
- Выбор подхода должен быть оправдан целями бизнеса, как и вся разработка коммерческого приложения.
- Не нужно следовать за модой на микросервисы. Часто использование монолита тоже бывает целесообразным.
- Чаще всего разработка ведется от монолита к микросервисам: в микросервис выделяется определенная логика, которая может быть использована разными частями приложения или даже другим приложением.

Монолитная архитектура

Особенности монолита

Представляет собой приложение поставляемое при помощи одного развертывания (например, WAR или EAR архив).
Взаимодействие между интерфейсом пользователя, бизнес-логикой и слоем доступа к данным происходит внутри одного модуля.

Микросервисная архитектура



Преимущества микросервисов

Упрощение операций: разработка, тестирование, CI, CD.
Гибкость и масштабируемость разработки, исполнения, сопровождения.
Независимость команд разработки (меньше зависимость от обновлений системы).
Легкий, быстрый и независимый выпуск обновлений. Возможность быстрого отката обновлений к предыдущему состоянию.

Недостатки (они тоже есть)

Необходимо правильное разделение на микросервисы. Оптимальной технологии разделения нет.
Запросы/ответы могут быть потеряны или поступать в неправильном порядке, микросервис может перестать отвечать или стать недоступным.
Отладка микросервисов намного сложнее – каждый вызов проходит через ряд

SOAP-взаимодействие

SOAP (Simple Object Access Protocol) — протокол обмена структурированными сообщениями в распределённой вычислительной среде.

Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC).

Практика

REST-взаимодействие

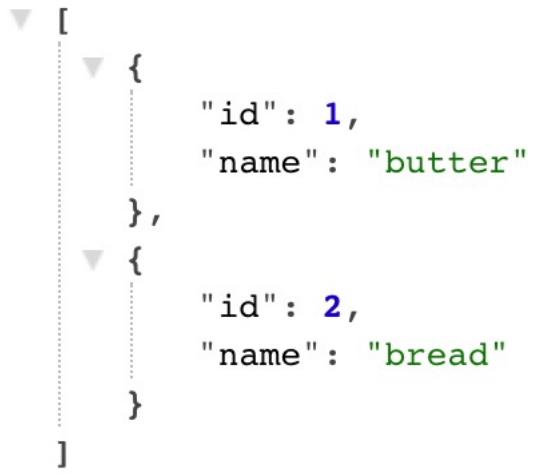
REST (Representational State Transfer) — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

Практика

HATEOAS

Hypermedia As The Engine Of Application State

Как обычно? Например, http://localhost:8080/products возвращает:



Как правило, когда мы выполняем запрос REST, мы получаем только данные, а не какие-либо действия с ними.

HATEOAS

С HATEOAS запрос на REST ресурс дает как данные, так и действия, связанные с данными.


```
{
  "_embedded": {
    "productList": [
      {
        "id": 1,
        "name": "butter",
        "_links": {
          "self": {
            "href": "http://localhost:8080/products/1"
          }
        }
      },
      {
        "id": 2,
        "name": "bread",
        "_links": {
          "self": {
            "href": "http://localhost:8080/products/2"
          }
        }
      }
    ]
  }
}
```

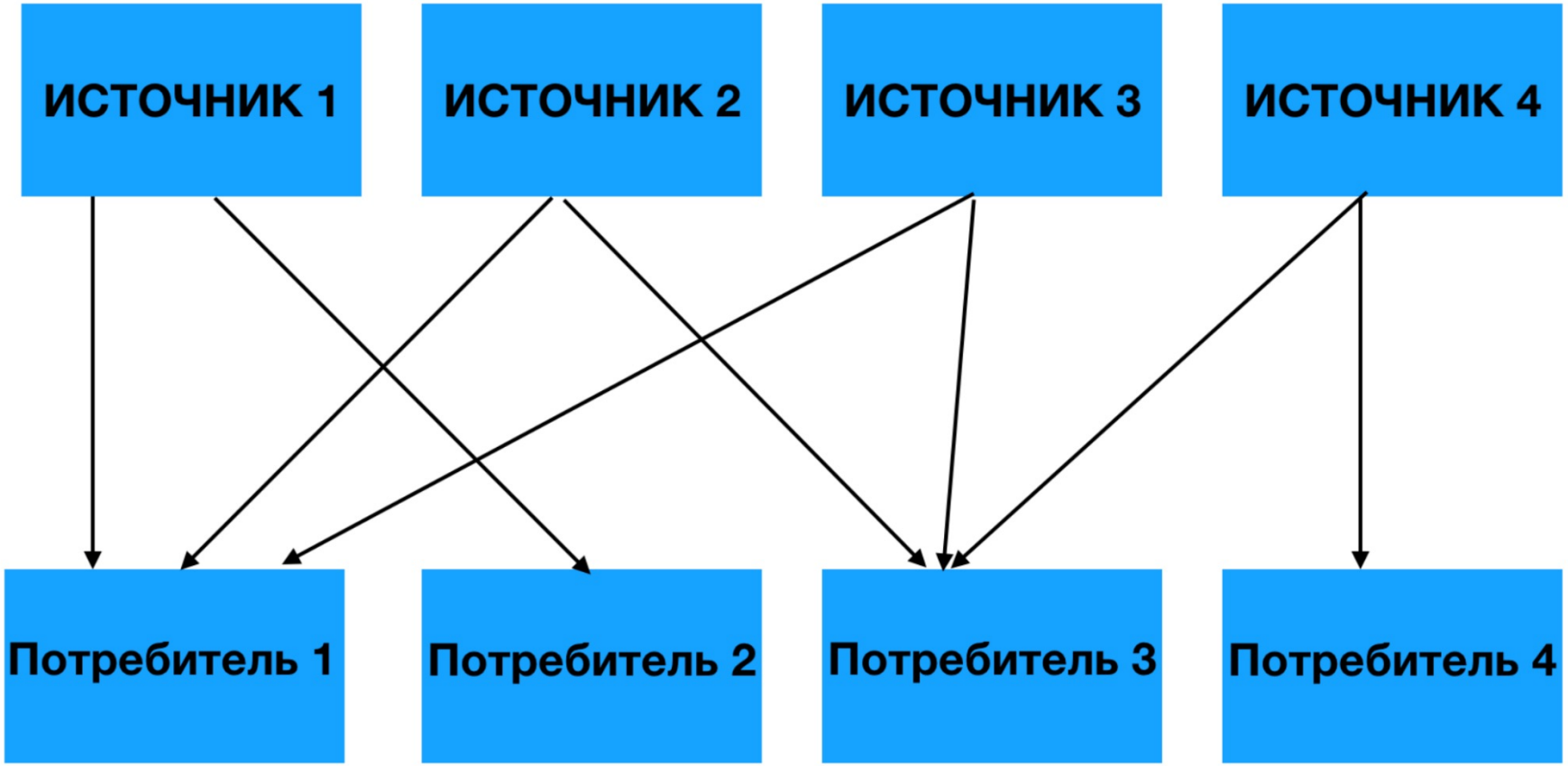
Практика

Передача сообщений

Простейший пример

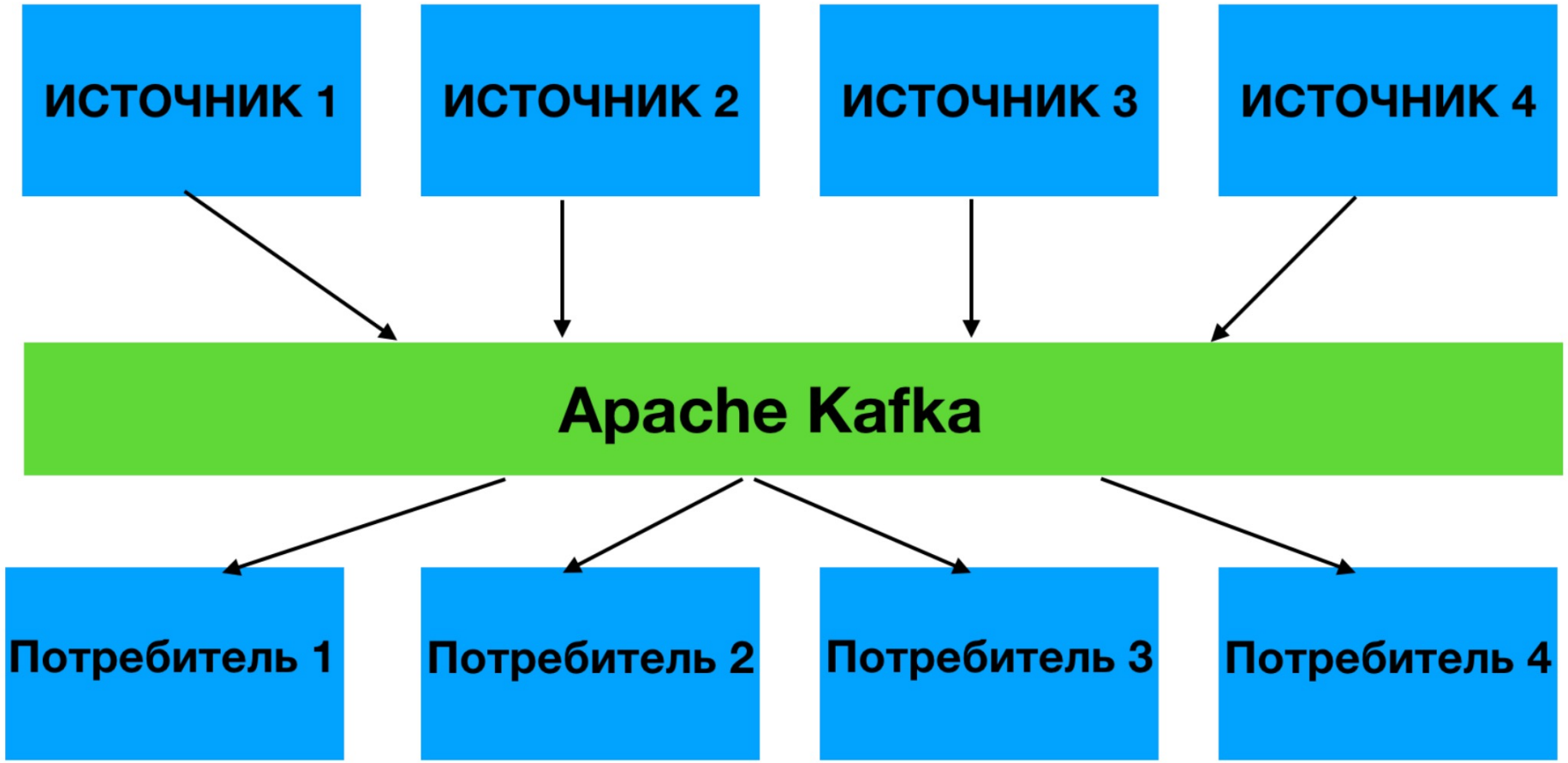
Передача сообщений

В реальности



Передача сообщений

Использование брокера



Apache Kafka

Apache Kafka — распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache. Написан на языках программирования Java и Scala.

Спроектирован как распределённая, горизонтально масштабируемая система, обеспечивающая наращивание пропускной способности как при росте числа и нагрузки со стороны источников, так и количества систем-подписчиков.
Подписчики могут быть объединены в группы.
Поддерживается возможность временного хранения данных для последующей пакетной обработки

Практика

Спасибо!