3
4

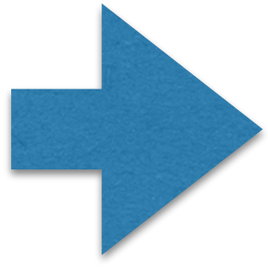
3
4

2 3 4 5 4 3 2

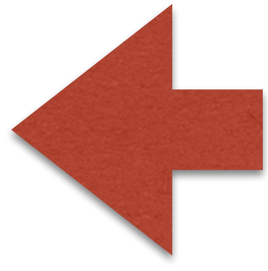
4
3

4
3

Проверка работоспособности кликера



Проверка работоспособности кликера



Микросервисы

первая кровь

Сячин Максим

maxsyachin@gmail.com

https://twitter.com/FinneTrolle

https://github.com/finnetrolle

http://www.finnetrolle.ru

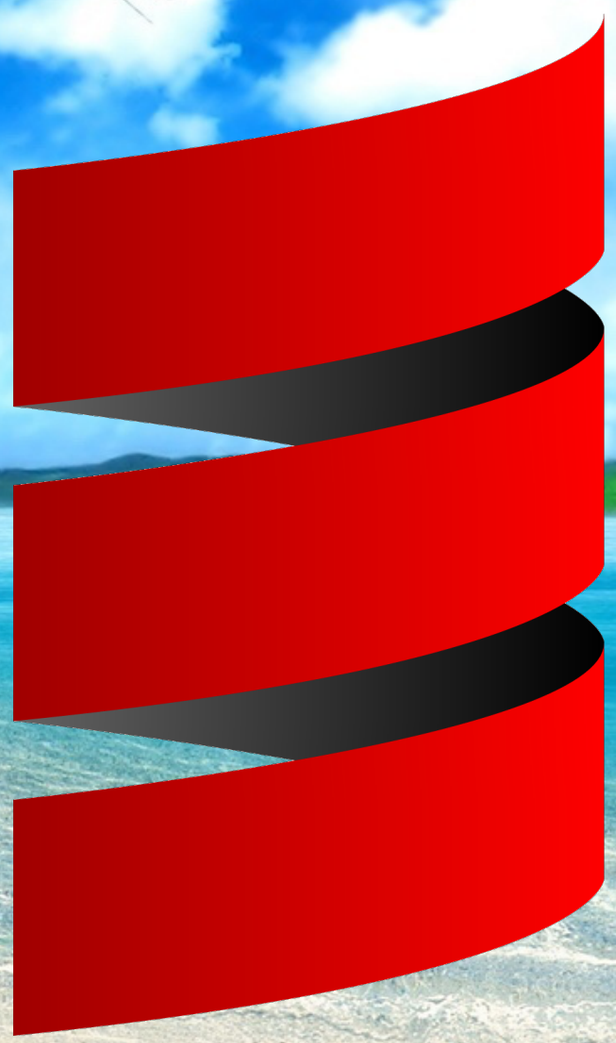
Ведущий разработчик

Опыт разработки: 11 лет


Borland©
Interface
Function
SQL
begin
Delphi™
Enterprise
7
Complete e-bussiness development
©Borland Software Corporation


C++


Success
Business
Customer
Management
Standard
Development
Consistency
Business
Optimal




ПОЧТА
РОССИИ

ПОЧТА РОССИИ

Подписка онлайн beta

Санкт-Петербург

Помощь

Максим Сячин

Газеты и журналы по подписке

Поиск по названию, теме или индексу издания

Газета Налоги Право Звезды

Подборки По алфавиту По теме

Популярное и новое

Ещё 10 >

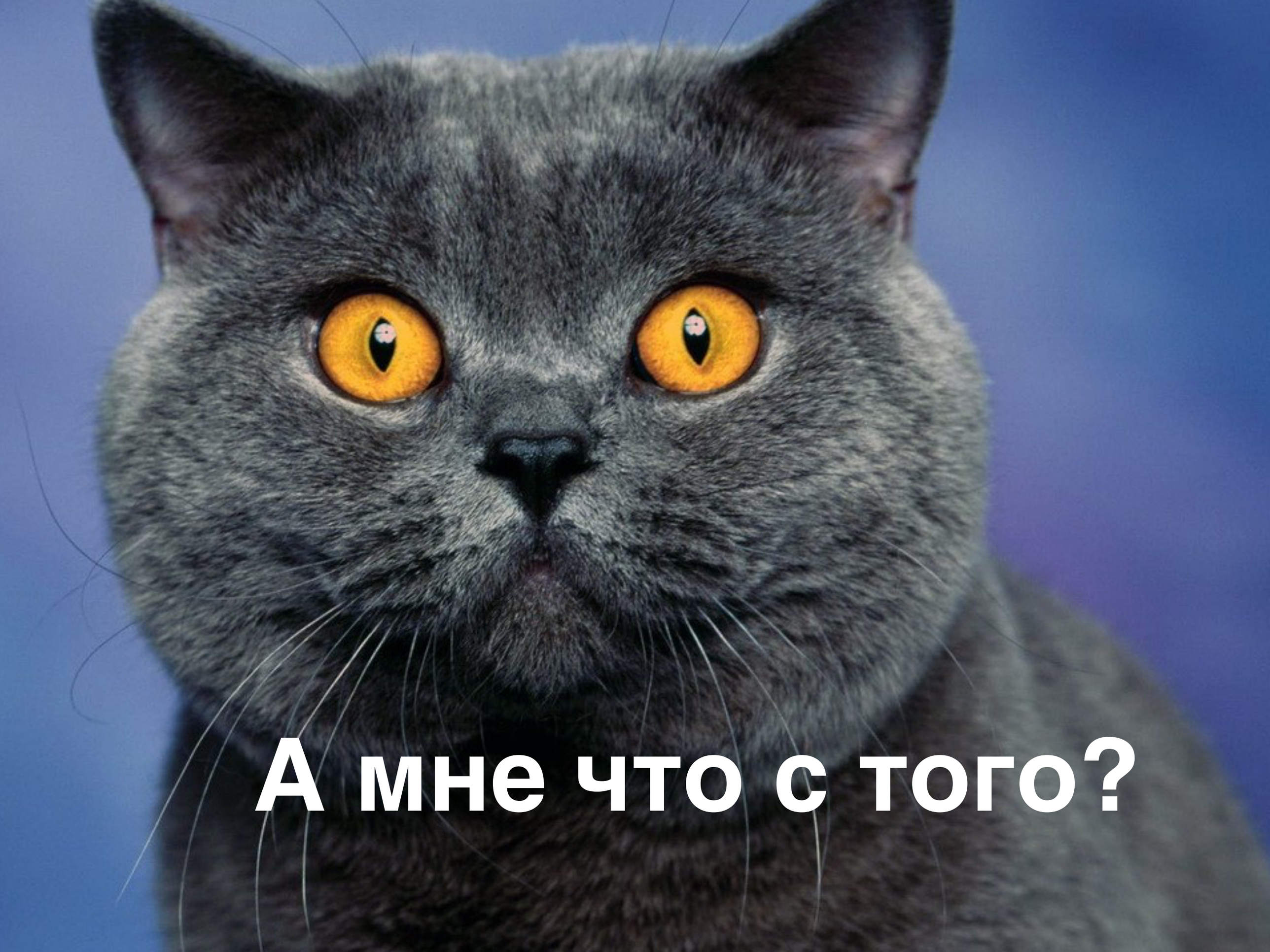
Популярная Механи...
Журнал 1 раз в месяц

Наука и жизнь
Журнал 1 раз в месяц

Весёлые картинки
Журнал 1 раз в месяц

Договоренности

- Все что я рассказываю - мой личный опыт
- Для вопросов будет секция QA в конце выступления
- Для больших вопросов - дискуссионная зона после выступления


А мне что с того?

Где набираетесь опыта?

- Собственные проекты
- Работа или фриланс
- Open source проекты
- Дипломные и научные работы

```
try {
    Map<Booking, List<Subscription>> bookings = subscriptions.stream()
            .collect(Collectors.groupingBy(Subscription::getBooking));
    CatalogueDefinerDataProvider defProvider = dataProviderFactory.getCatalogueDefinerDataProvider(
            subscriptions.stream().map(Subscription::getStartDate).collect(toSet()),
            subscriptions.stream().map(Subscription::getIndex).collect(toSet()));
    EndOrderDates endOrderDates = nsi.getEndOrderDates();
    ExternalRepository<String, Tcfps> tcfpsRepo = new ExternalRepository<>(Tcfps::getCode, nsi.getTcfpsList());
    for (Map.Entry<Booking, List<Subscription>> entry : bookings.entrySet()) {
        ValidationResult validation = validator.validate(entry.getValue(), entry.getKey().getDate(),
                defProvider, endOrderDates, tcfpsRepo);
        if (validation.containsError()) {
            validationResult.add(validation);
            continue;
        } else {
            checkAnnulmentAndReaddressing(entry.getVa
            if (validation.containsWarnings()) {
                bookingWorkflow.processEvent(entry.g
            } else {
                bookingWorkflow.processEvent(entry.getKey(), Book
            }
        }
        if (validation.getViolations().size() > 0) {
            validationResult.add(validation);
            validationWarnings.addWarning(entry.getKey(), validation.getWarnings());
        }
    }
} catch (CatalogueNotFoundException e) {
    validationResult.add(ValidationResult.getInstance().addViolations(
            ERROR, UploadBookingResultType.CATALOGUE_VALIDATE,
            loc.getMessage("validation.catalogue.not.found", new SimpleDateFormat("yyyy-MM-dd").format(e.getDate()))));
} catch (ReleaseScheduleIsInvalidException e) {
    validationResult.addAll(e.getInvalidKeys().stream().map(k -> ValidationResult.getInstance().addViolations(ERROR,
            UploadBookingResultType.EXTERNAL_SYSTEM_ERROR,
            loc.getMessage("validation.catalogue.havenot.releaseschedule", k.getPublicationCode(), k.getTcfpsCode(), k.getId())))
            .collect(toList()));
    faultHandler.onExternalSystemException(e);
} catch (ExternalSystemException e) {
    faultHandler.onExternalSystemException(e);
}
```

Целью сценария является обработка и размещение в сервисе подписки информации о заказах, полученной из УФПС. Как инициируется этот процесс, и кто является его инициатором, пока в данном сценарии не определяется (пока это веб-форма интерфейс).

Данные получаются из ИС «Подписка-УФПС» в виде набора файлов с заказами в формате «Новый формат АПР».

Примечание: ИС «Подписка-УФПС» должна быть настроена для формирования и передачи заказов сервису подписки по каталожным ценам.

Основной сценарий

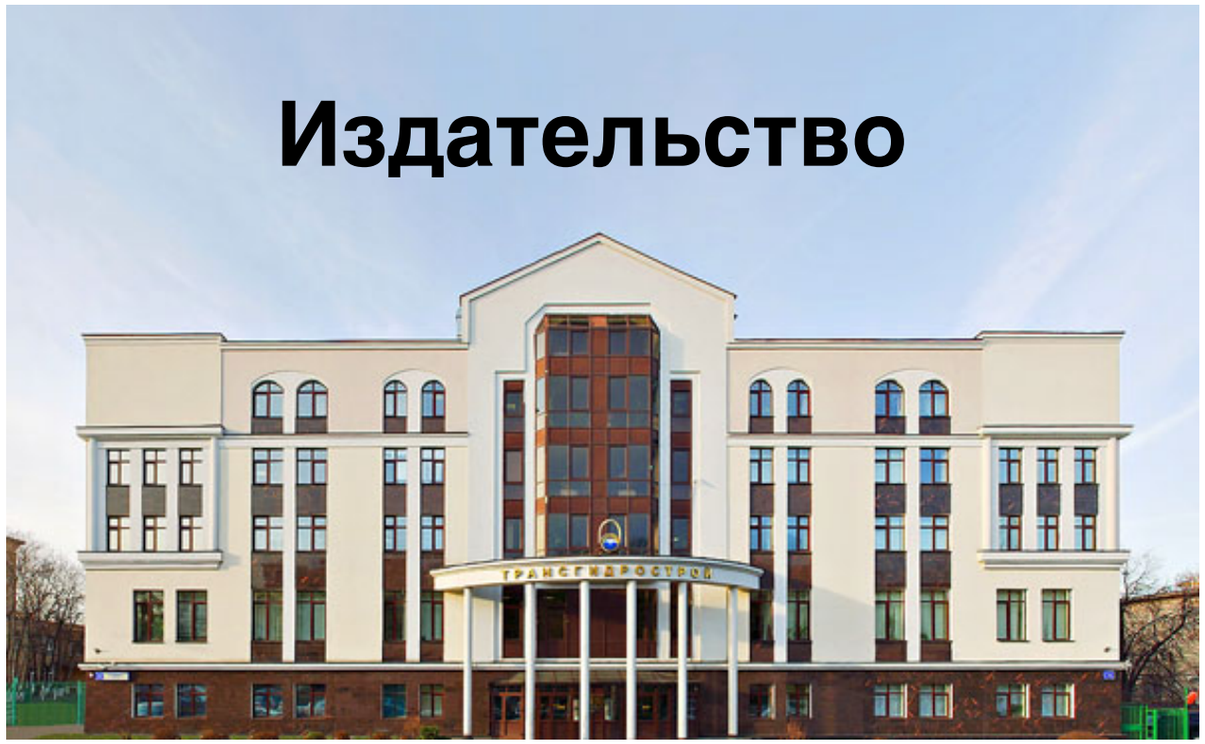
1. Сервис подписки получает набор файлов с заказами из УФПС. Набор файлов состоит из
    a. файла с описанием заказа СП-6,
    b. файла с карточками СП-5,
    c. файла с доставочными карточками СП-1 (для адрес
    d. файла с текстовой справкой.
2. Сервис подписки проводит следующие проверки:
    a. Валидацию данных на формат в соответствии с описанием формата
    b. Проверку данных на целостность в соответствии с логической мод
    c. Валидацию на допустимый вид заказа в СП6: могут быть толь я 1, 6, 8. При другом виде заказа ошибка: "Заказ <номер заказа>. <вид заказа> - недопустимый вид заказа."
    d. Дата начала заказа должна быть одинаковой для заказа СП- арточек СП-5, и его доставочных карточек СП-1.
    e. Подписной индекс и минимальный срок подписки у карточек СП-5 и связанных карточек СП-1 должны быть одинаковыми.
    f. Сумма по заказу СП-6 должна совпадать с суммой сумм за п й период всех его карточек СП-5. Сумма за подписной период карточки СП-5 должна совпадать с суммой стоимостей подписки за ко ы всех ее доставочных карточек СП-1, если для полученного подписного индекса указана адресная система распространения.
    g. Для СП5 и СП1: сумма за подписной период должна быть равна стоимости за МСП умноженной на число комплектов за подписной период.
    h. Каждый отмеченный месяц должен быть кратен МСП. При нарушении заказ не принимается, текст ошибки "Заказ <номер заказа>. СП5 <номер СП5 в заказе>. Индекс <подписной индекс>. Заказанные комплекты отмечены в месяцах не кратных МСП".
    i. Не должны быть отмечены как заказанные МСП невыхода: издание должно выходить хотя бы раз в период месяцев: [отмеченный месяц - МСП +1, отмеченный месяц]. Для комплекта достаточно, чтобы в этот период выходило хотя бы одно издание из комплекта. При нарушении заказ не принимается, текст ошибки "Заказ <номер заказа>. СП5 <номер СП5 в заказе>. Индекс <подписной индекс>. Заказанные комплекты отмечены в месяцах невыхода".
    j. Валидацию на соответствие каталогам и срокам приема.
    k. Для заказов на аннуляцию и переадресовку: количества заказанных комплектов на трактовках, на которых производится уменьшение тиража, должны оставаться положительными для всех месяцев календарного года, содержащего подписное полугодие. См. Правило

Проект с микросервисами - лучший вариант работы для студента

A long time ago in a galaxy far,

far away. . . .

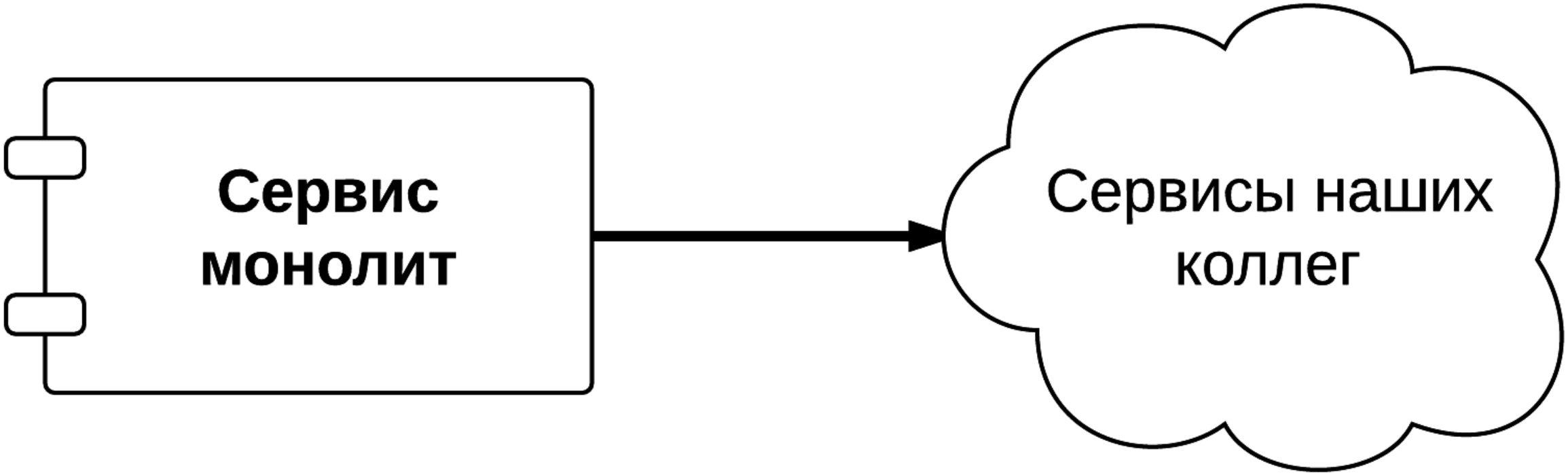

ПОЧТА РОССИИ

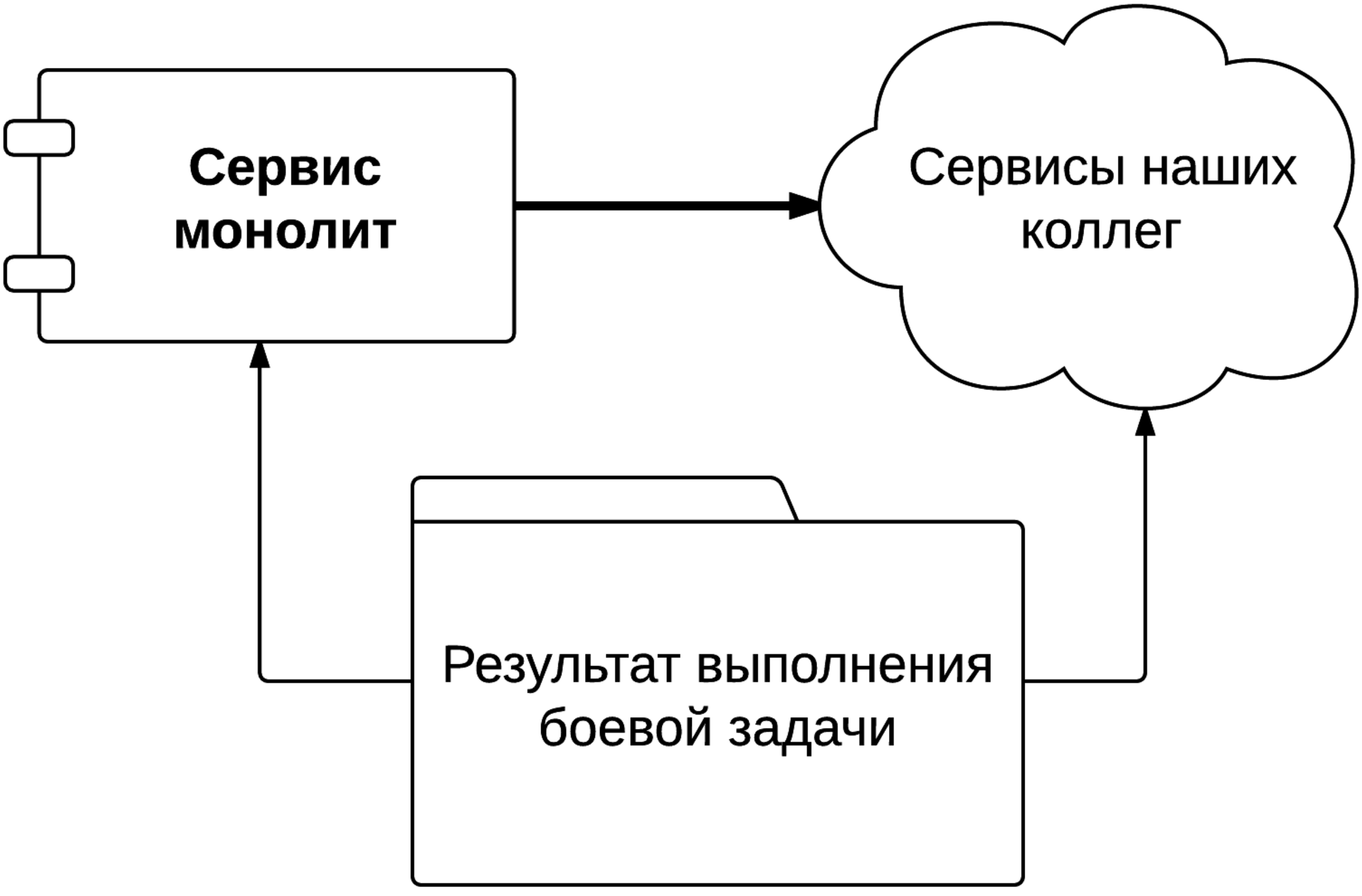

Издательство
ТРАНСГИДРОСТРОЙ


МАП
МЕЖРЕГИОНАЛЬНОЕ АГЕНТСТВО ПОДПИСКИ
1000 советов дачнику
1000 советов
СПАСИБО, ЧТО ВЫБИРАЕТЕ НАС!
КАТАЛОГ РОССИЙСКОЙ ПРЕССЫ
ПОЧТА РОССИИ
2015 первое полугодие
Сваты
подпишись!
СВАТ ПЛОХОГО НЕ ПОСОВЕТУЕТ!
ВСЕ ЛУЧШЕЕ – ДЕТЯМ!
моя прекрасная Дача
Годовым подписчикам подарок!
Добровольно сертифицированная Система Менеджмента Качества при предоставлении услуги «Распространение периодической печатной продукции»
ГОСТ Р

Что мы делаем?

- Сопровождаем процесс подписки
- Печатаем документы для сторон
- Строим разные графики и отчеты


Сервис
монолит
Сервисы наших
коллег


Сервис монолит
Сервисы наших коллег
Результат выполнения боевой задачи

Что такое микросервисы?

In short, the microservice architectural style is an approach to developing a single application as a **suite of small services**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are **built around business capabilities** and **independently deployable** by fully automated deployment machinery. There is a **bare minimum of centralized management** of these services, which may be written in different programming languages and use different data storage technologies.

-- James Lewis and Martin Fowler

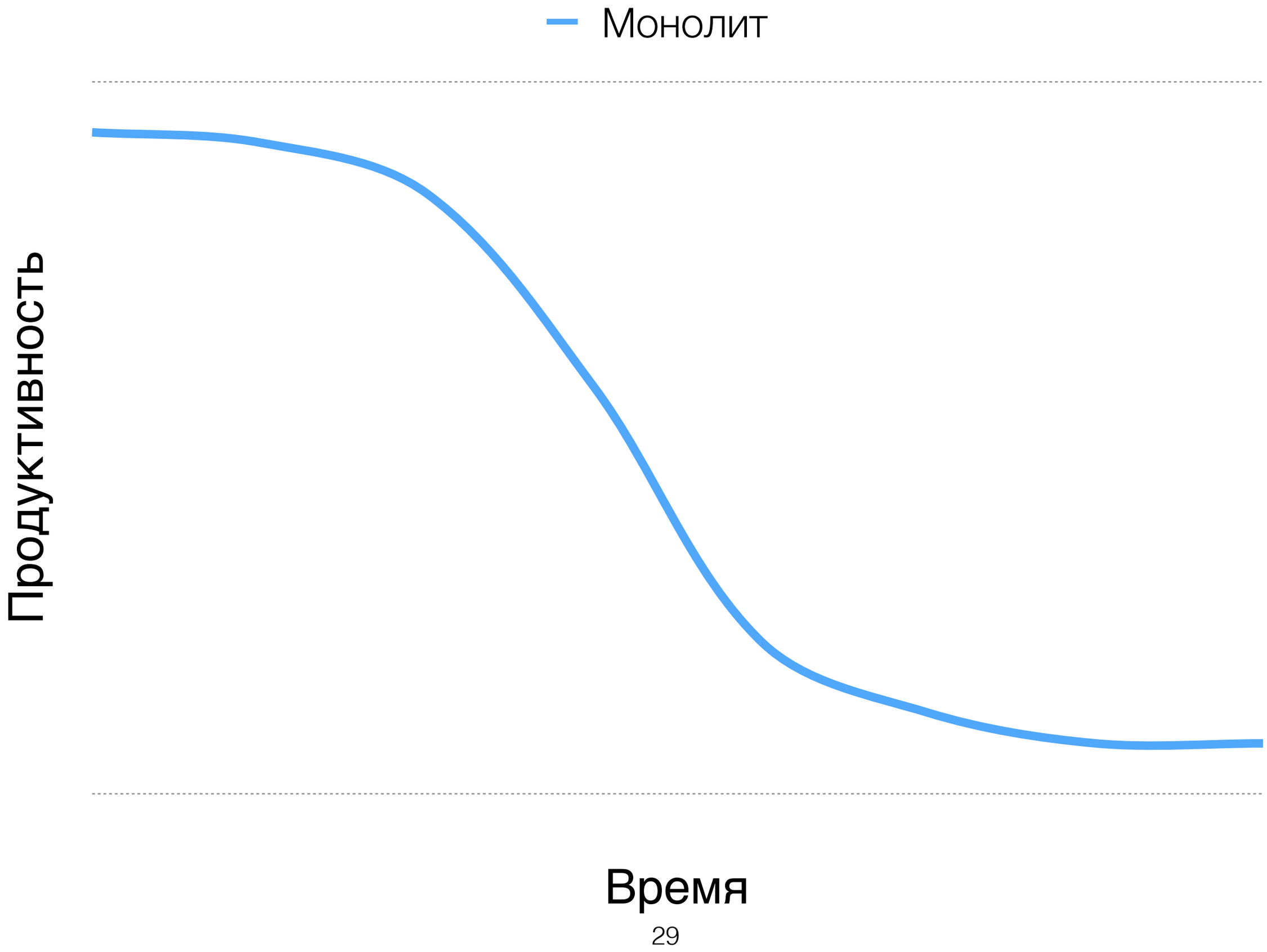
http://martinfowler.com/microservices/

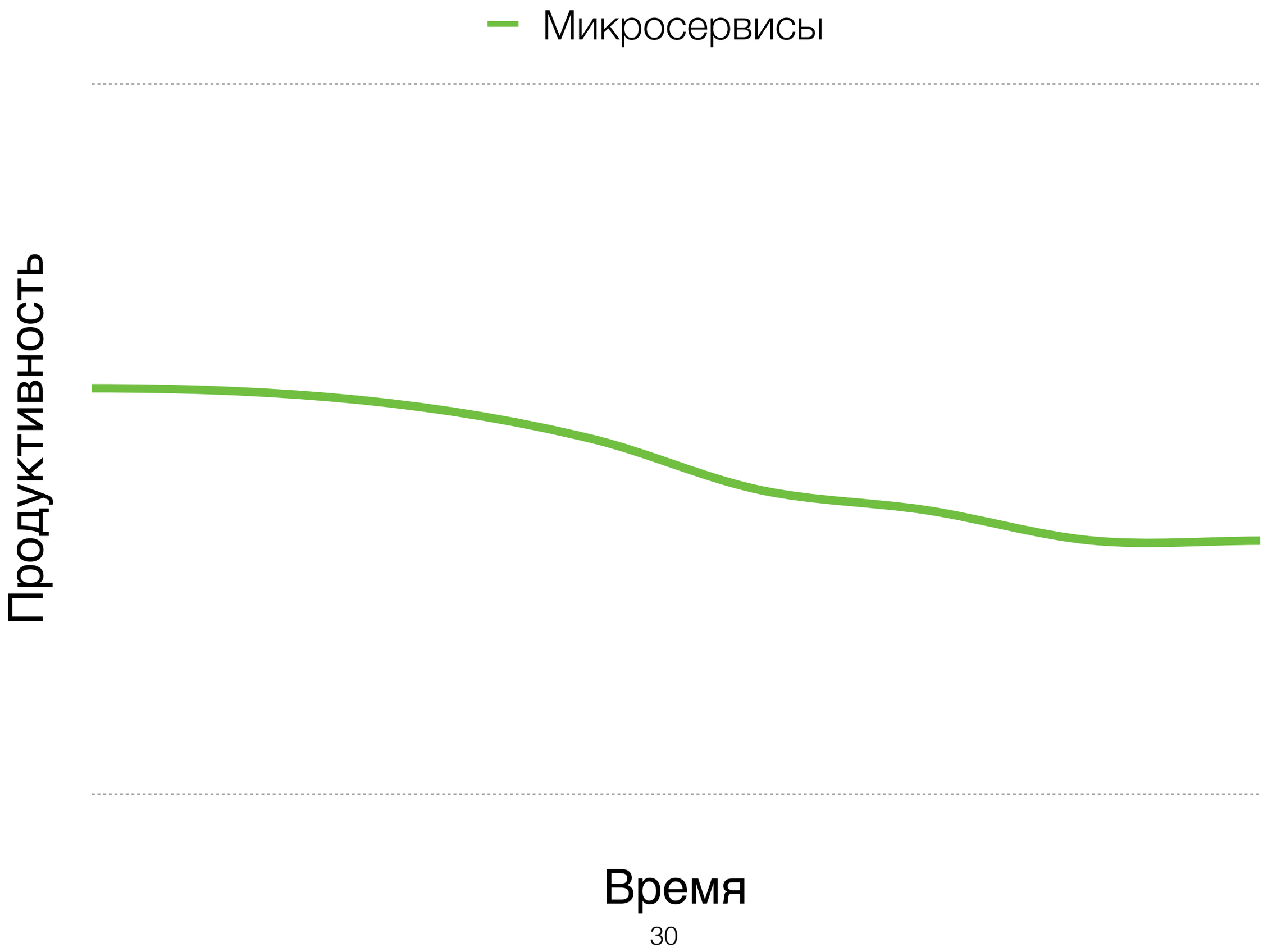
Продуктивность

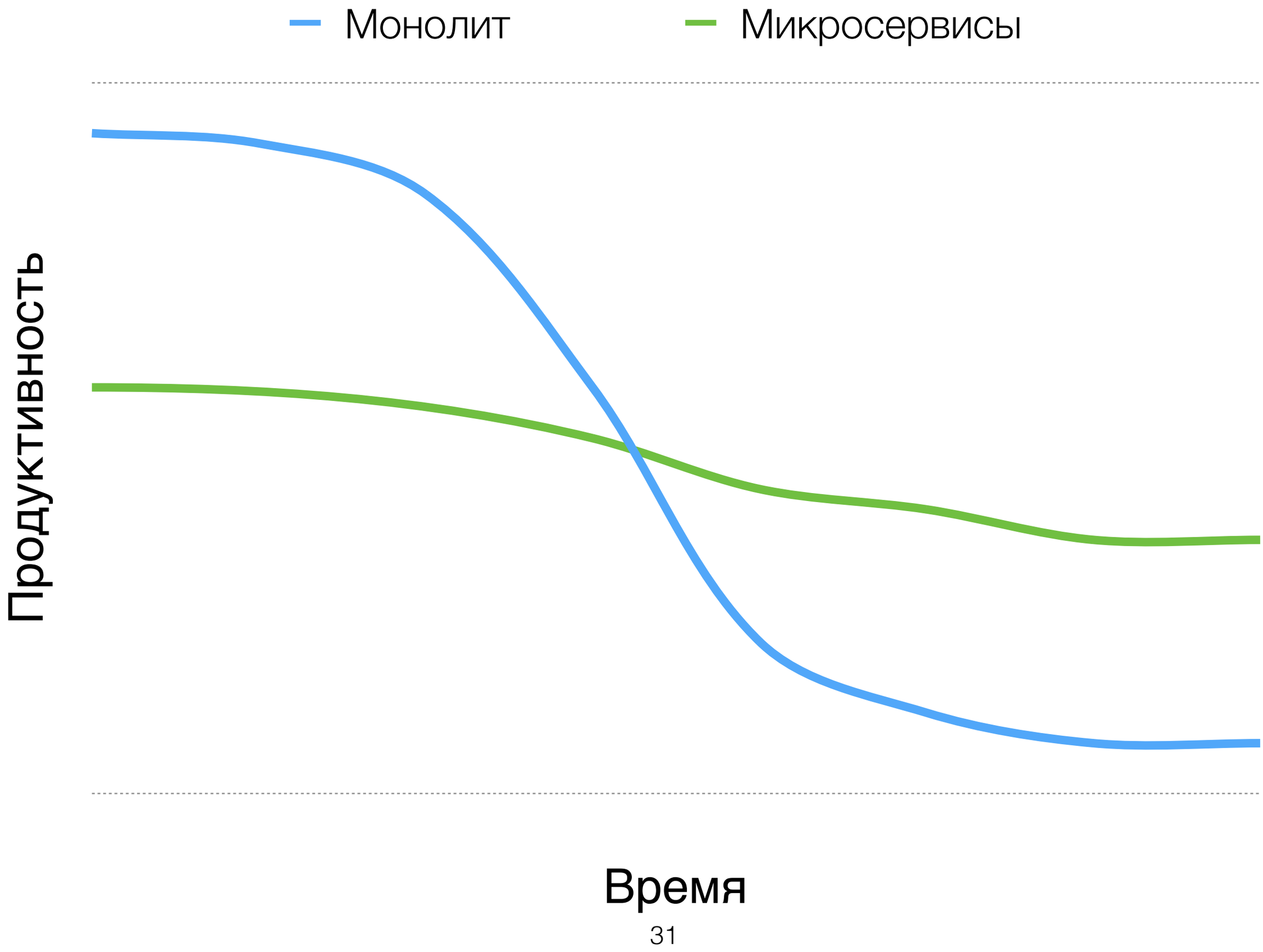
Чем выше продуктивность - тем быстрее можно внедрить новую функциональность

Чем дольше разрабатывается проект - тем больше в нем функциональности и тем выше его сложность

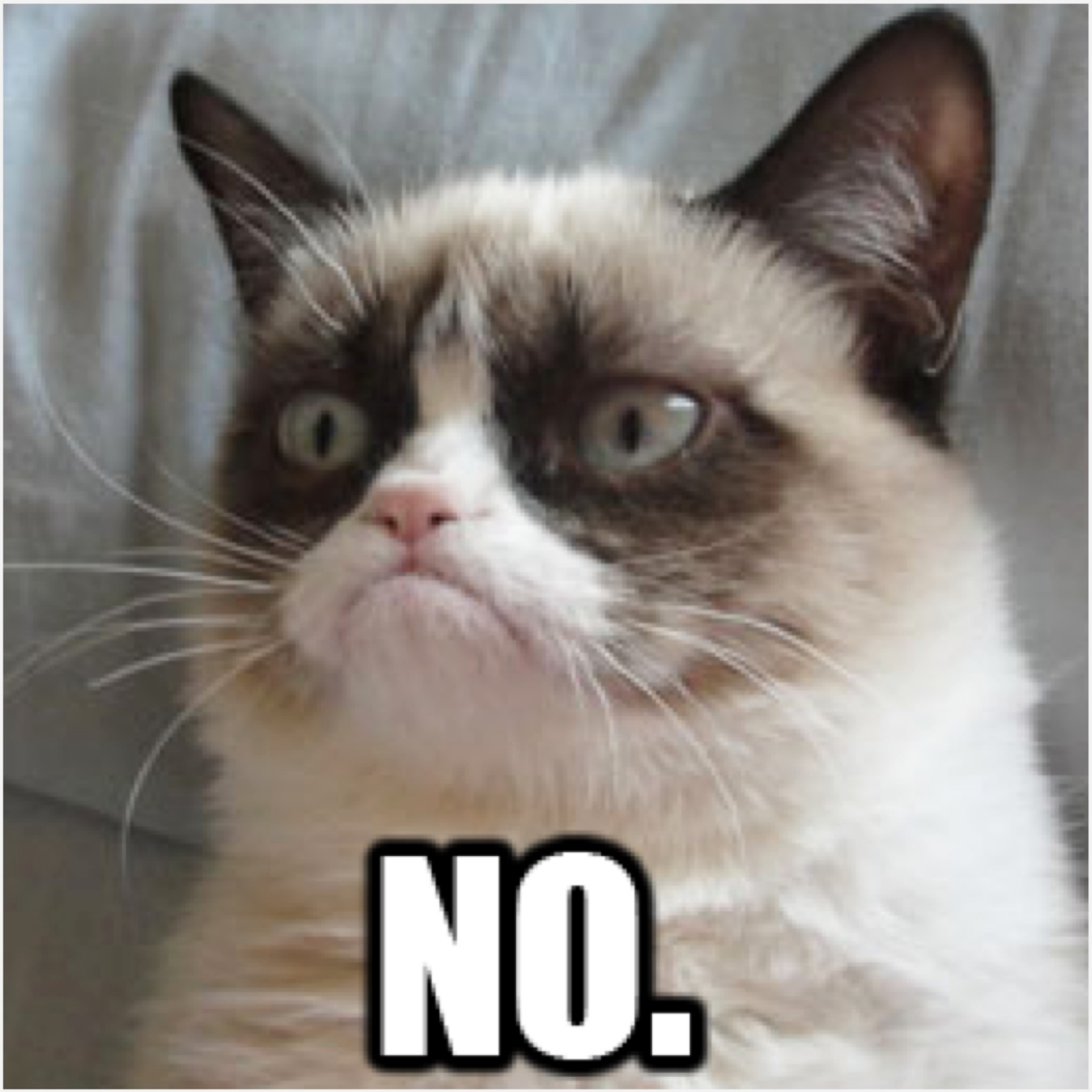
Время

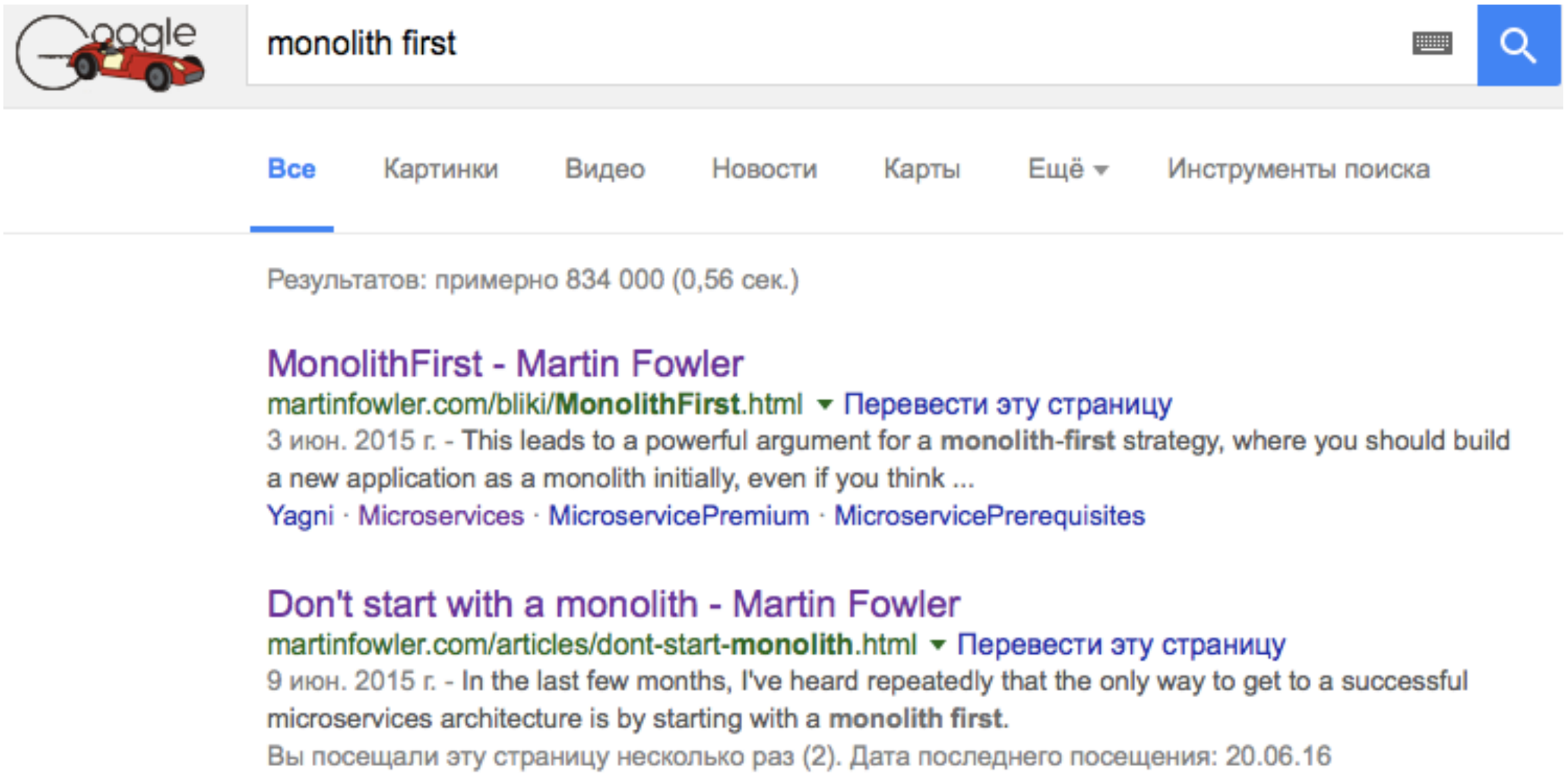

Монолит
Продуктивность
Время


Микросервисы
Продуктивность
Время


Монолит
Микросервисы
Продуктивность
Время

Грааль найден!


NO.



Monolith First

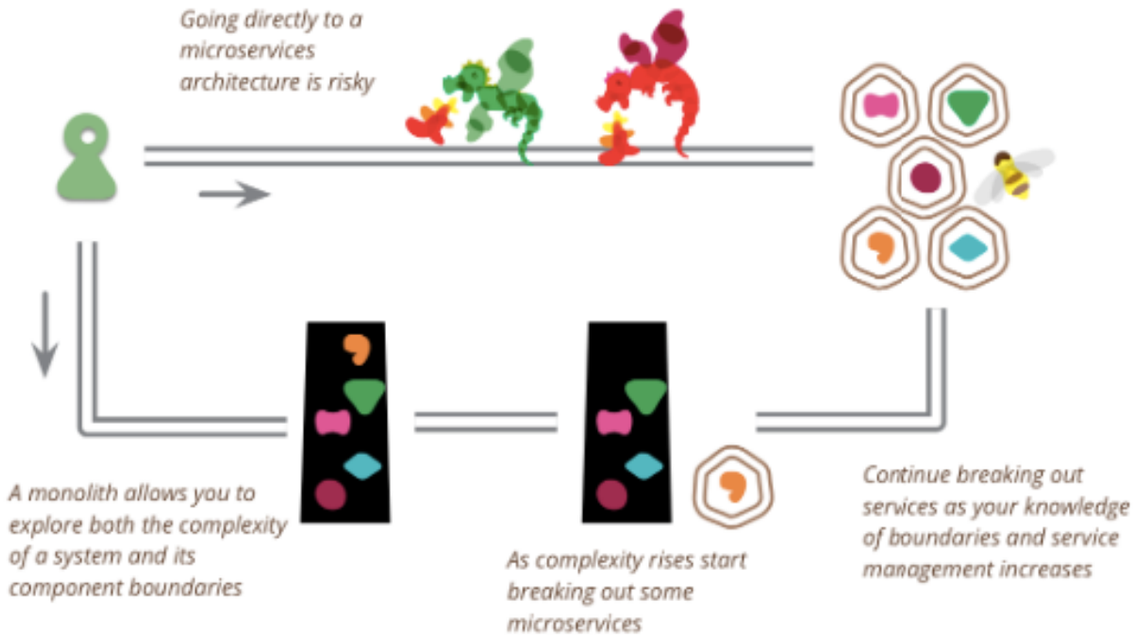
MonolithFirst

Martin Fowler
3 June 2015

As I hear stories about teams using a microservices architecture, I've noticed a common pattern.

1. Almost all the successful microservice stories have started with a monolith that got too big and was broken up
2. Almost all the cases where I've heard of a system that was built as a microservice system from scratch, it has ended up in serious trouble.

This pattern has led many of my colleagues to argue that **you shouldn't start a new project with microservices, even if you're sure your application will be big enough to make it worthwhile.** .



Don’t start with a monolith

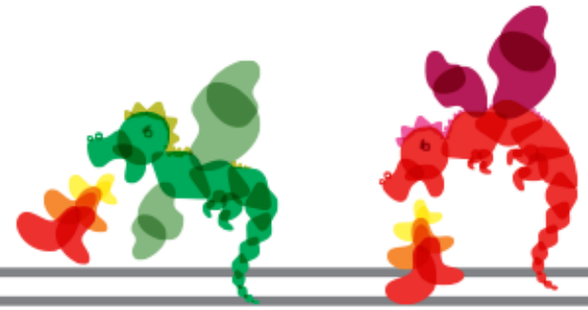
… when your goal is a microservices architecture

Stefan Tilkov

Stefan Tilkov is a co-founder and principal consultant at innoQ, a technology consulting company with offices in Germany and Switzerland. He has been involved in the design of large-scale, distributed systems for more than two decades, using a variety of technologies and tools ranging from C++ and CORBA over J2EE/Java EE and Web Services to REST and Ruby on Rails. He has authored numerous articles and a book (“REST und HTTP”, German), and is a frequent speaker at conferences around the world.

09 June 2015

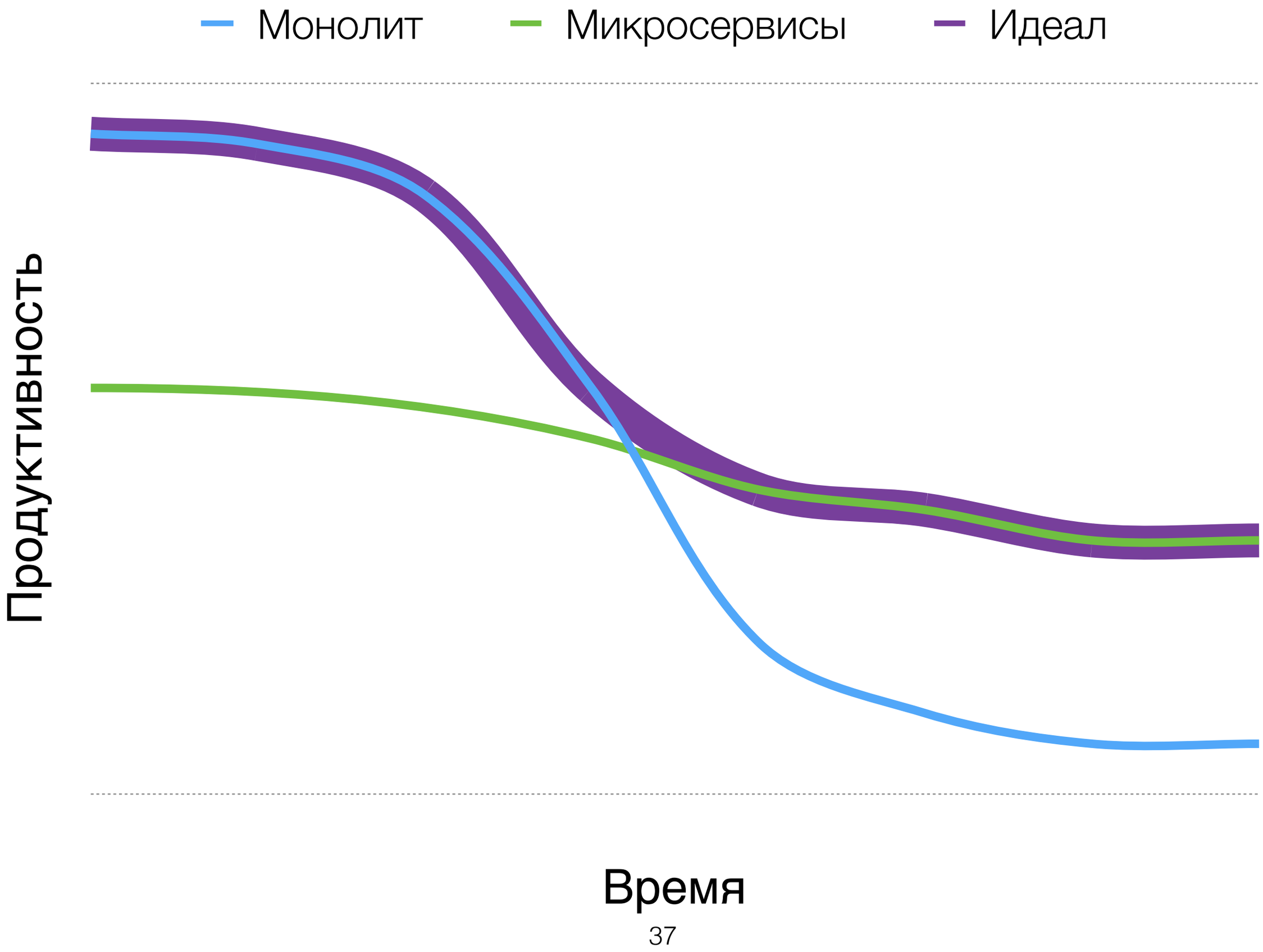
Going directly to a microservices architecture is risky



A monolith allows you to explore both the complexity of a system and its component boundaries

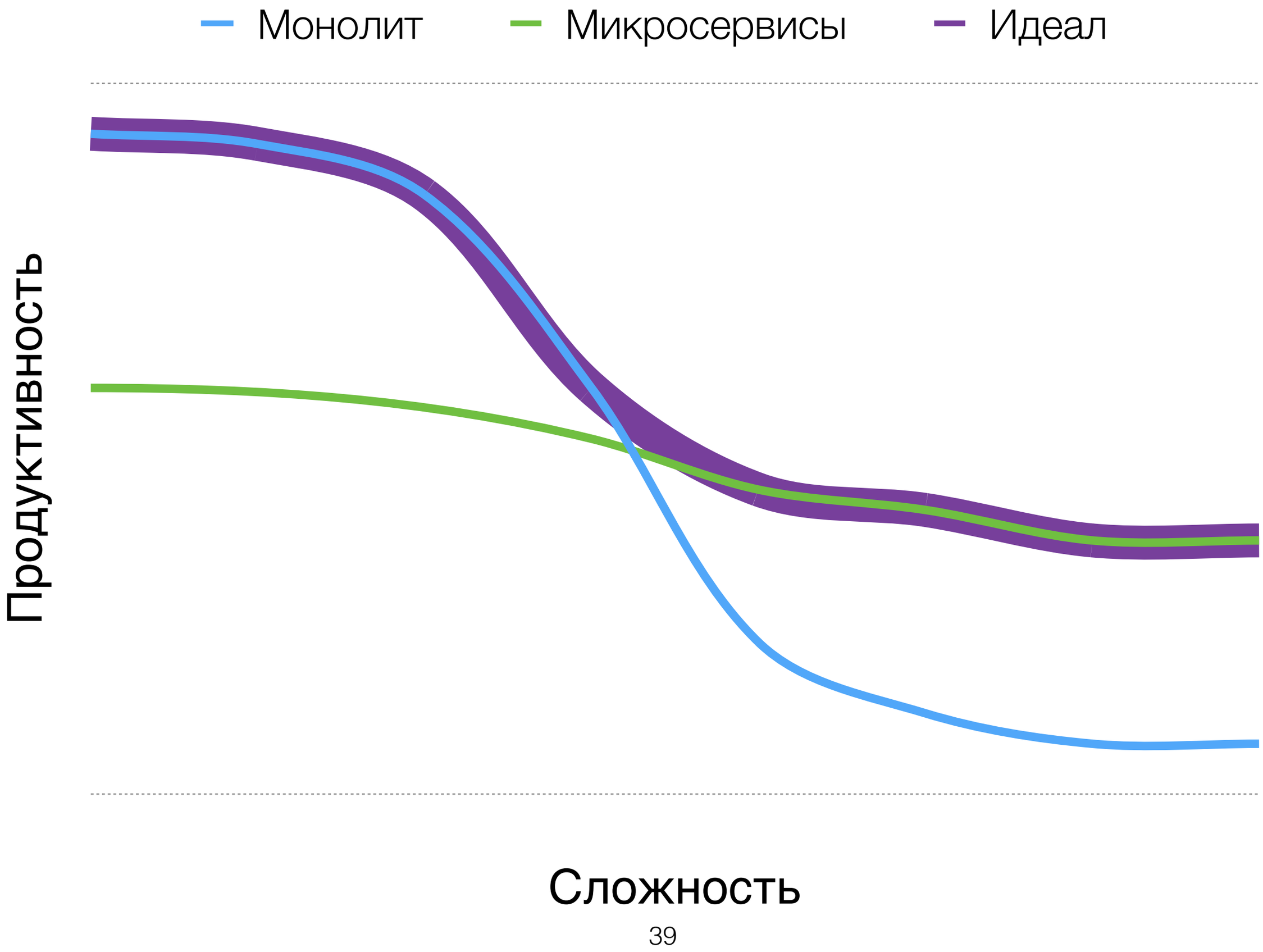
As complexity rises start breaking out some microservices

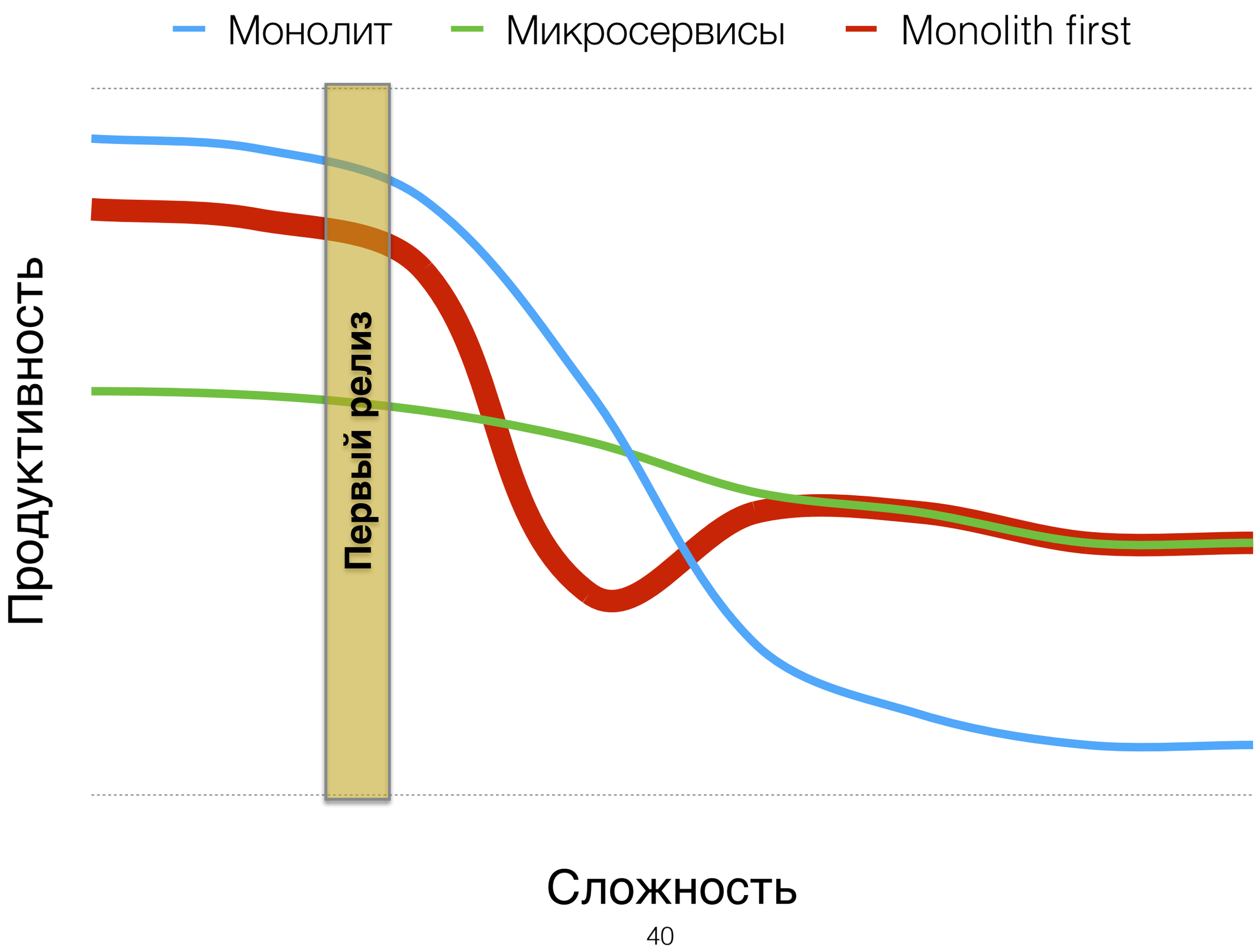
Continue breaking out services as your knowledge of boundaries and service management increases


Монолит
Микросервисы
Идеал
Продуктивность
Время

Обязательные требования

- Интерфейсы для разбиения
- Не допускать высокую связность
- Высокая дисциплина
- Время на разбиение

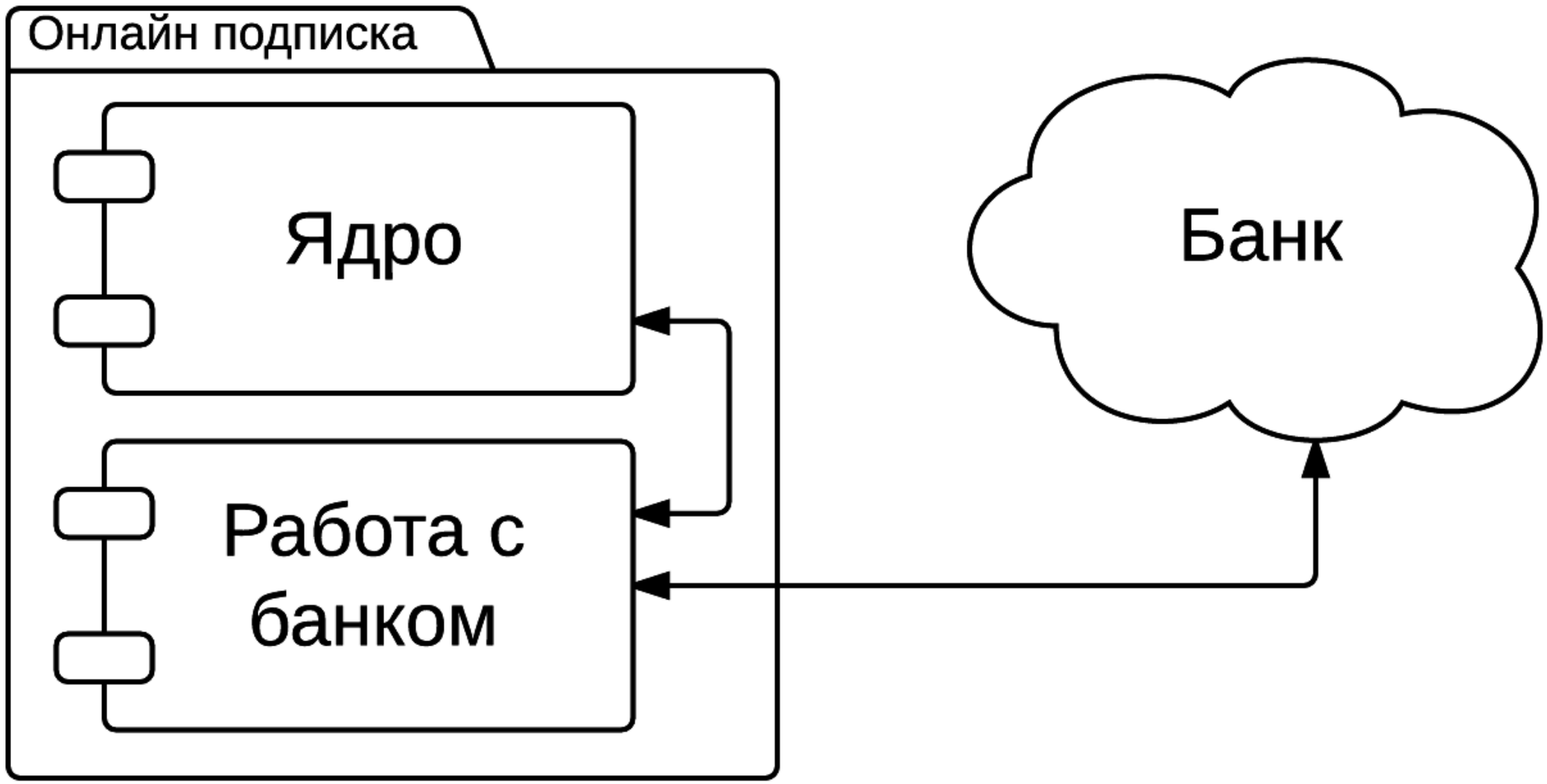

Монолит
Микросервисы
Идеал
Продуктивность
Сложность

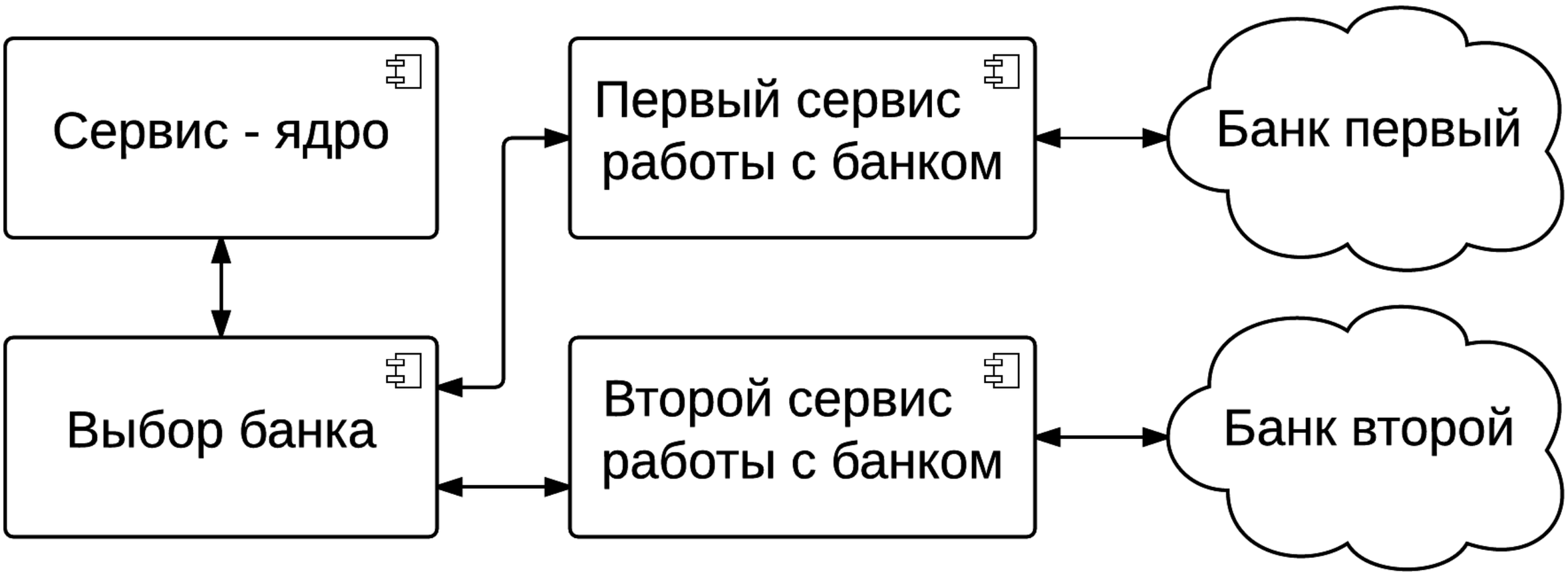

Монолит
Микросервисы
Monolith first
Продуктивность
Первый релиз
Сложность

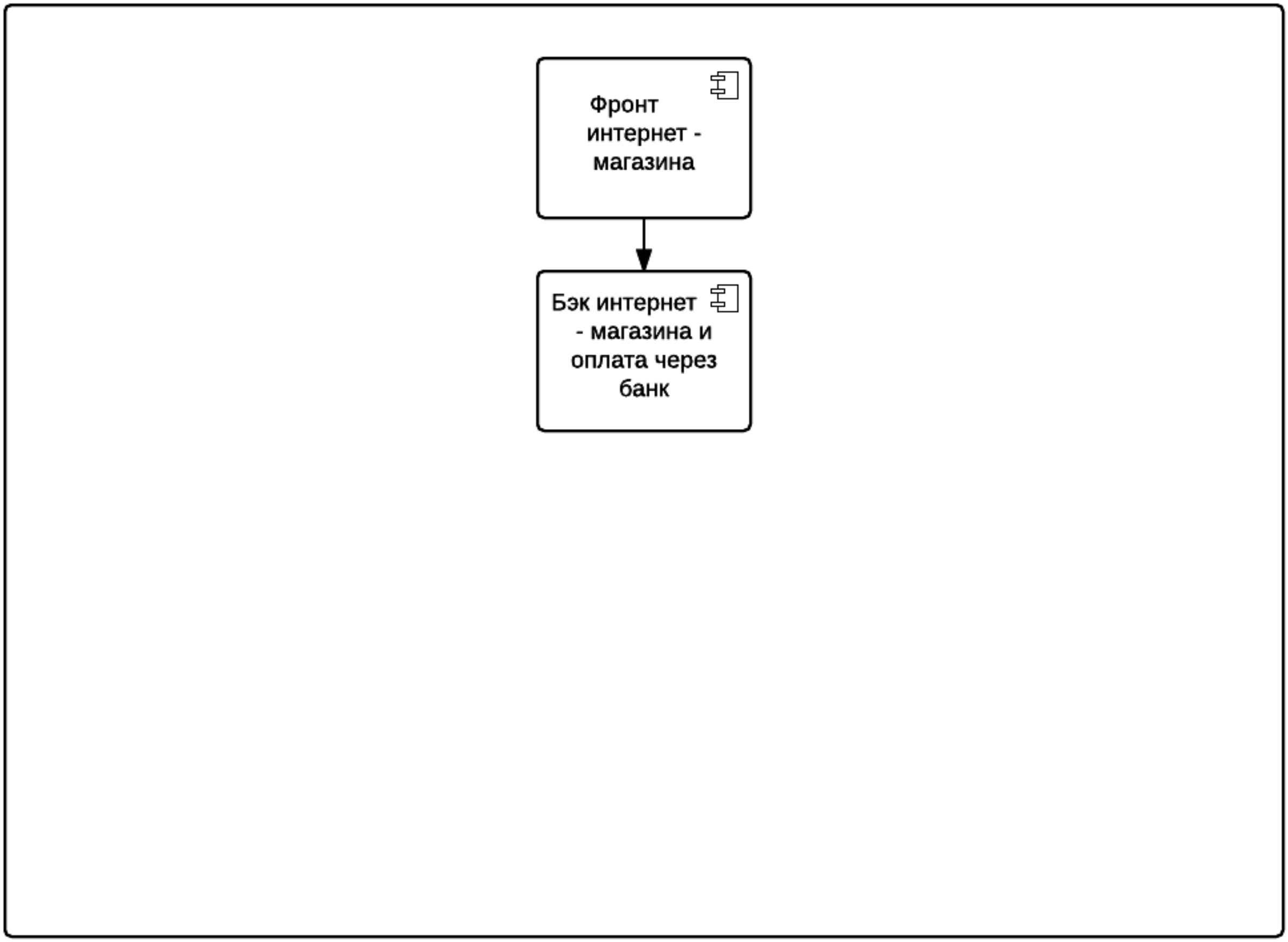
Как Monolith First помог нашему проекту?

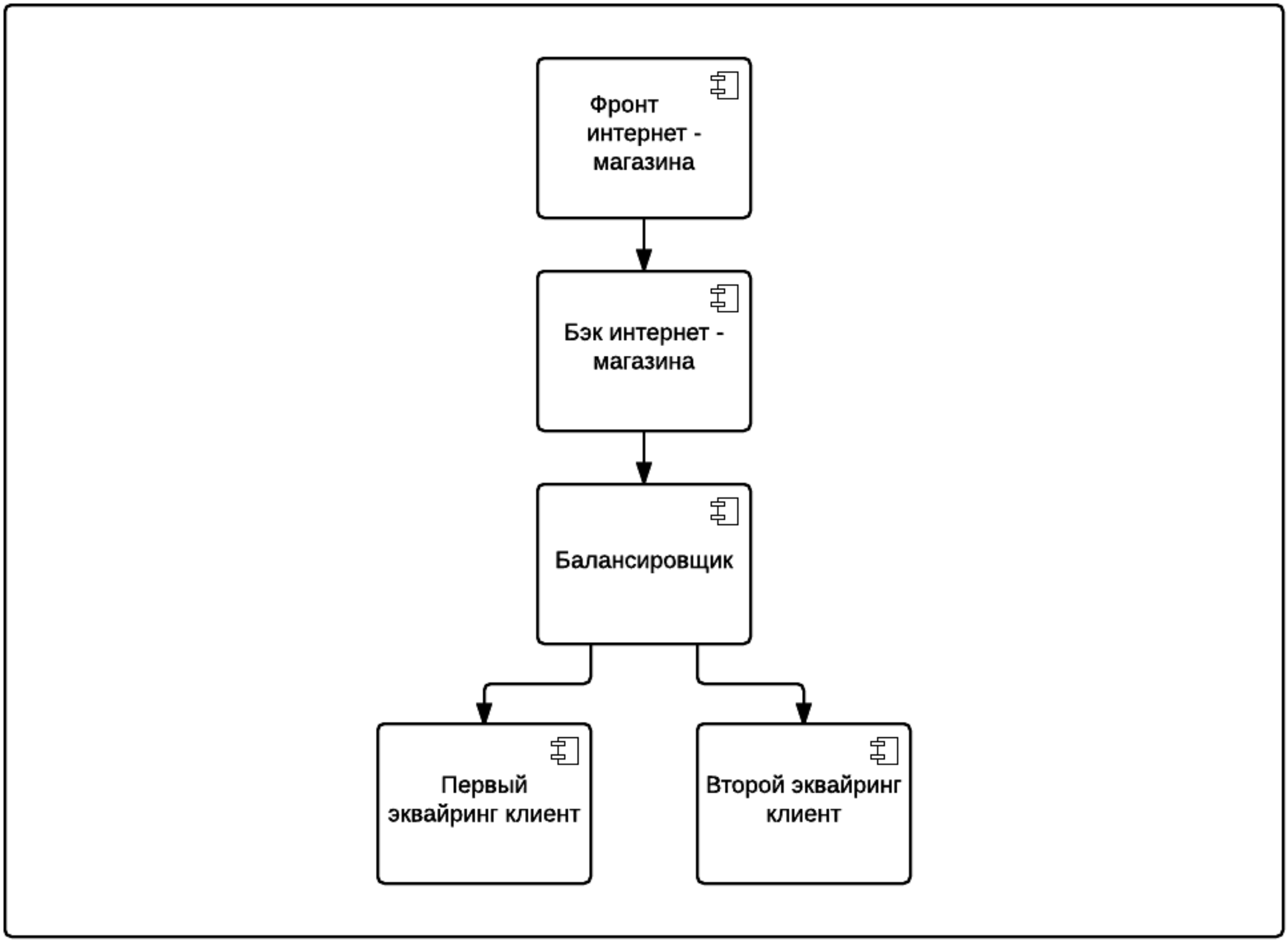
- Ускорен выпуск релиза (мы успели!)

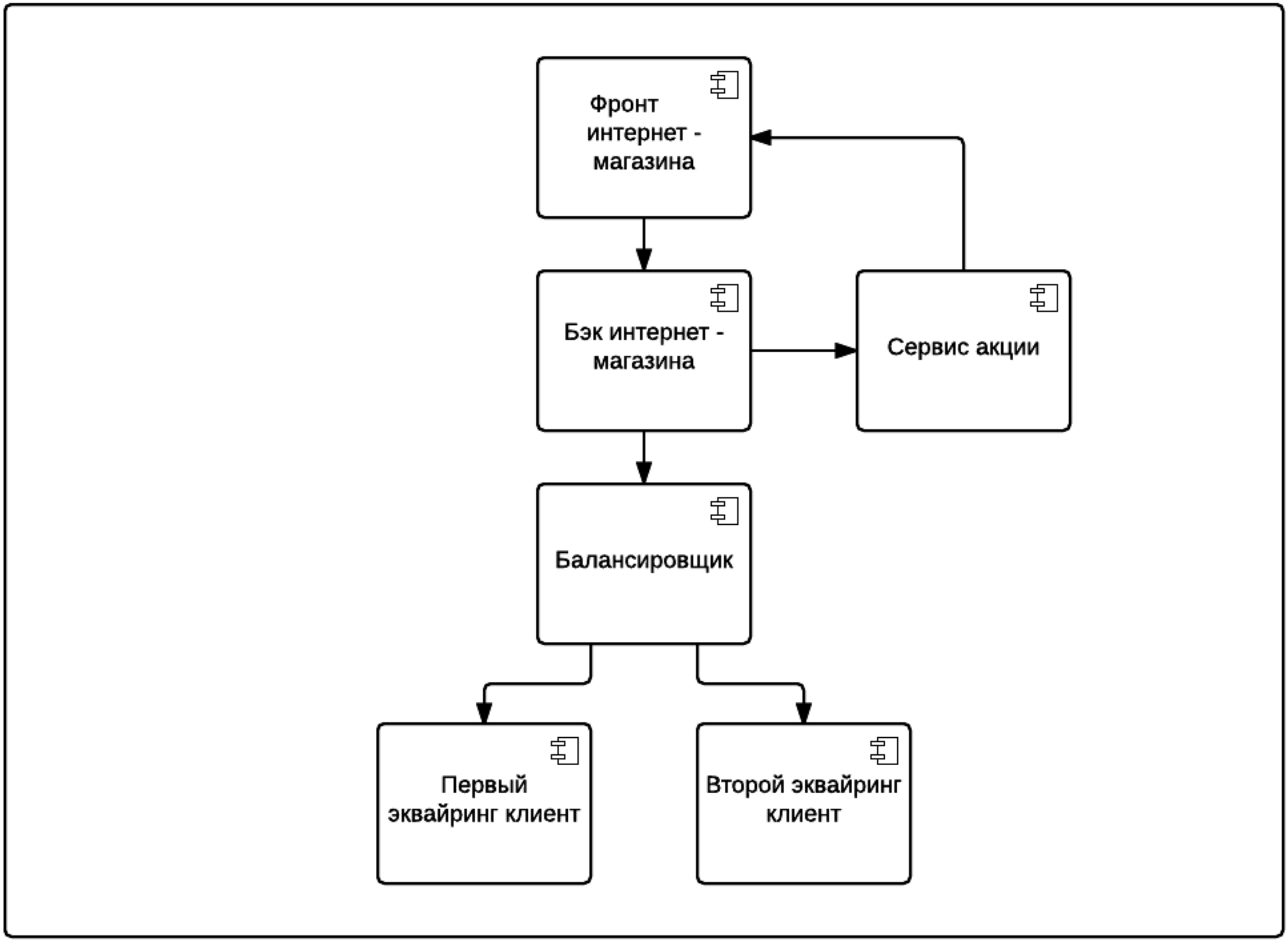
- Определились границы между модулями

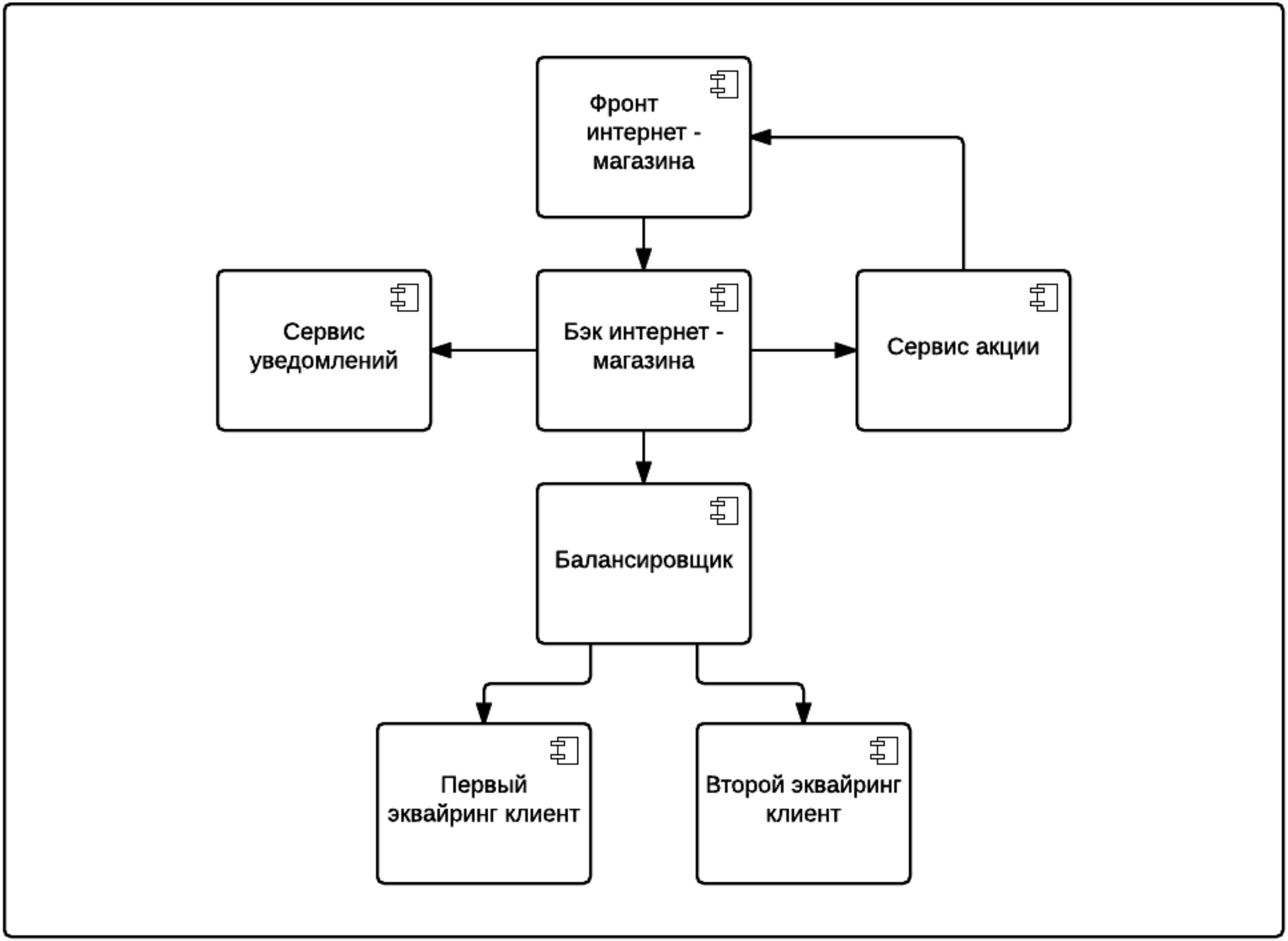

Онлайн подписка
Ядро
Работа с банком
Банк

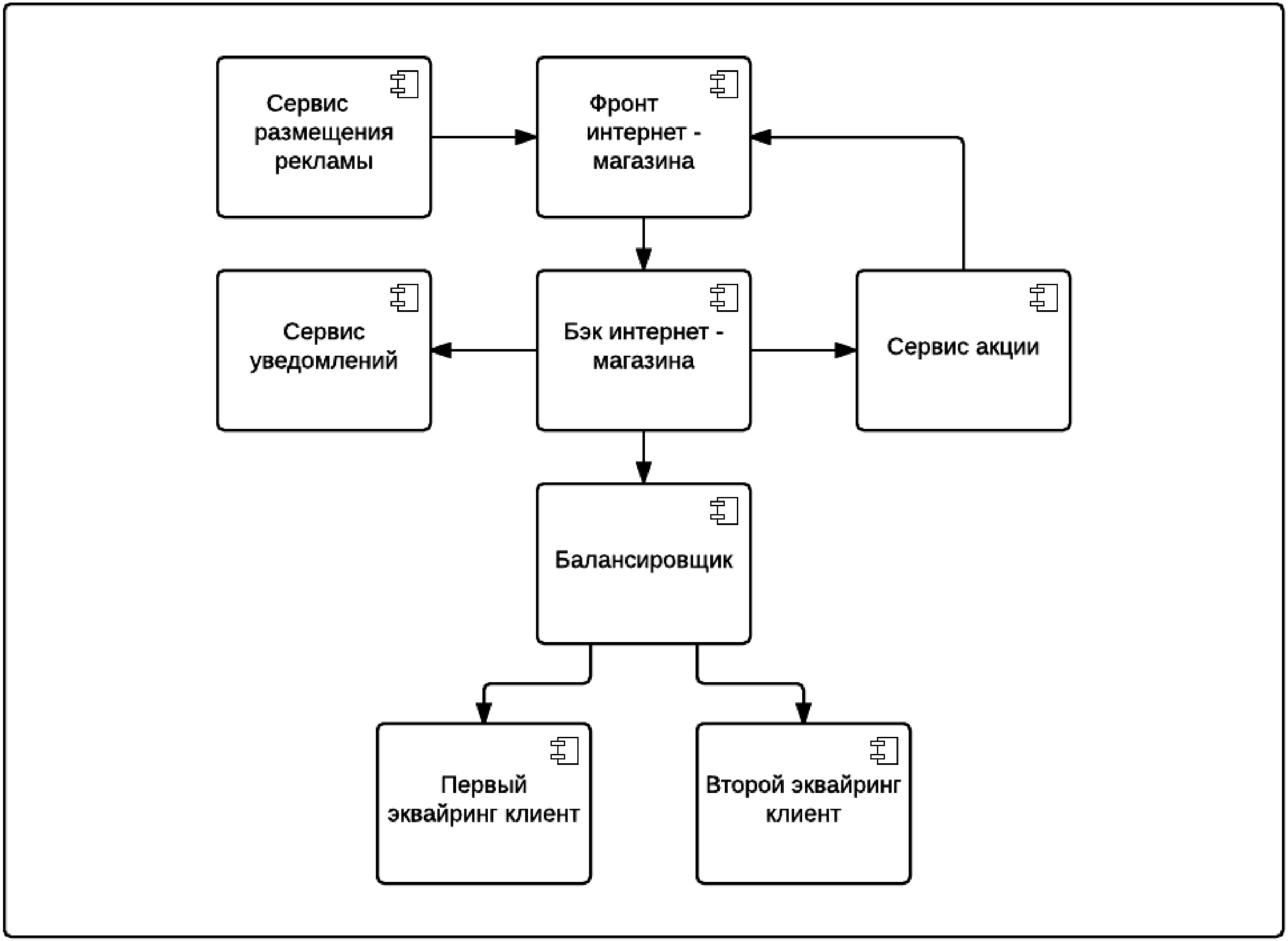

Сервис - ядро
Первый сервис работы с банком
Банк первый
Выбор банка
Второй сервис работы с банком
Банк второй

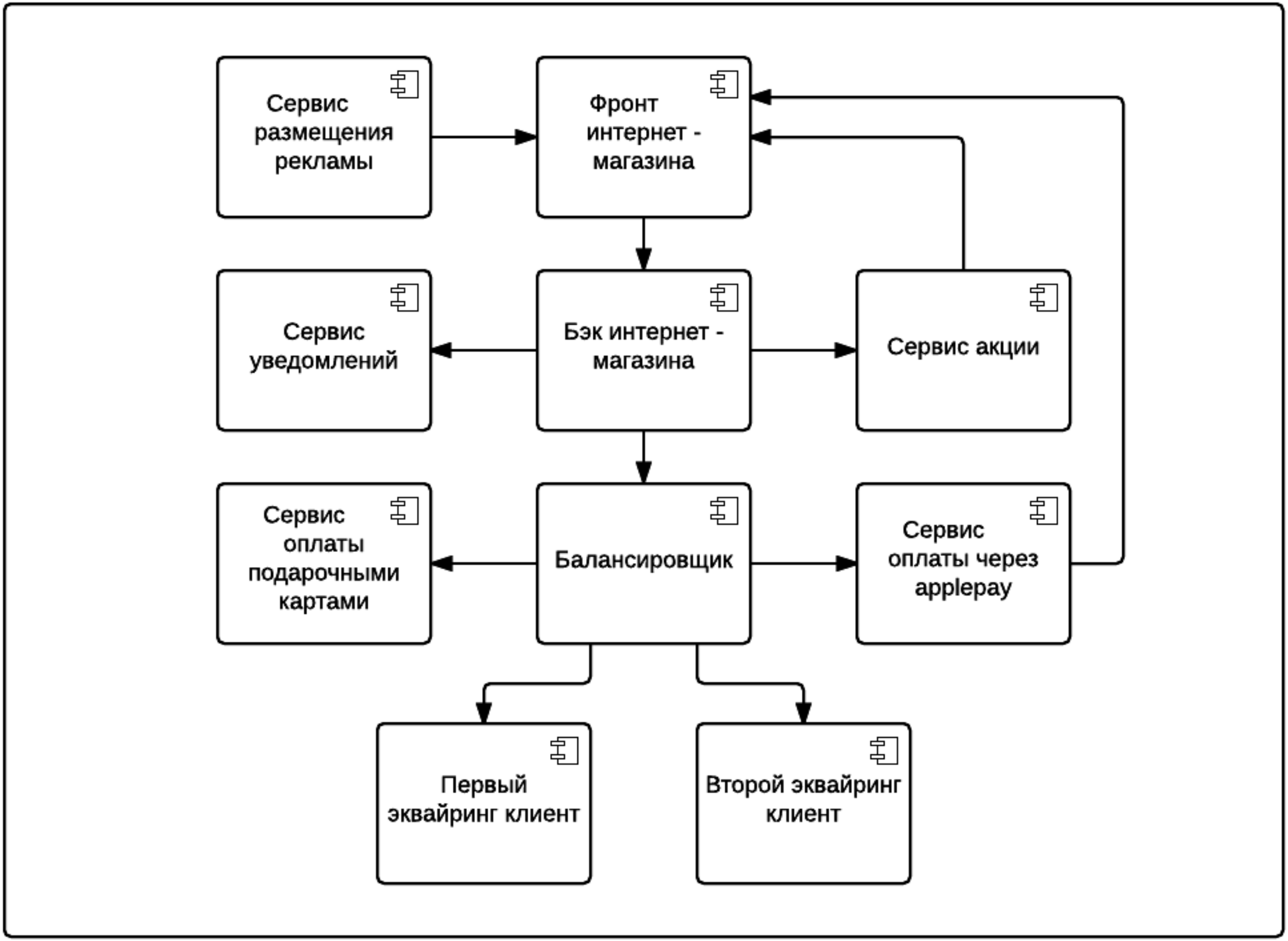

Фронт интернет - магазина
Бэк интернет - магазина и оплата через банк


Фронт интернет - магазина
Бэк интернет - магазина
Балансировщик
Первый эквайринг клиент
Второй эквайринг клиент


Фронт интернет - магазина
Бэк интернет - магазина
Сервис акции
Балансировщик
Первый эквайринг клиент
Второй эквайринг клиент


Фронт интернет - магазина
Сервис уведомлений
Бэк интернет - магазина
Сервис акции
Балансировщик
Первый эквайринг клиент
Второй эквайринг клиент


Сервис размещения рекламы
Фронт интернет - магазина
Сервис уведомлений
Бэк интернет - магазина
Сервис акции
Балансировщик
Первый эквайринг клиент
Второй эквайринг клиент


Сервис размещения рекламы
Фронт интернет - магазина
Сервис уведомлений
Бэк интернет - магазина
Сервис акции
Сервис оплаты подарочными картами
Балансировщик
Сервис оплаты через applepay
Первый эквайринг клиент
Второй эквайринг клиент

Недостатки большого количества сервисов

- Большая операционная сложность
- Труднее воспроизвести окружение для тестирования
- Отказы - это данность

Преимущества большого количества сервисов

- Горизонтальное масштабирование из коробки
- Низкая связность между сервисами...
- ... и между командами!
- Каждая часть системы достаточно мала чтобы...

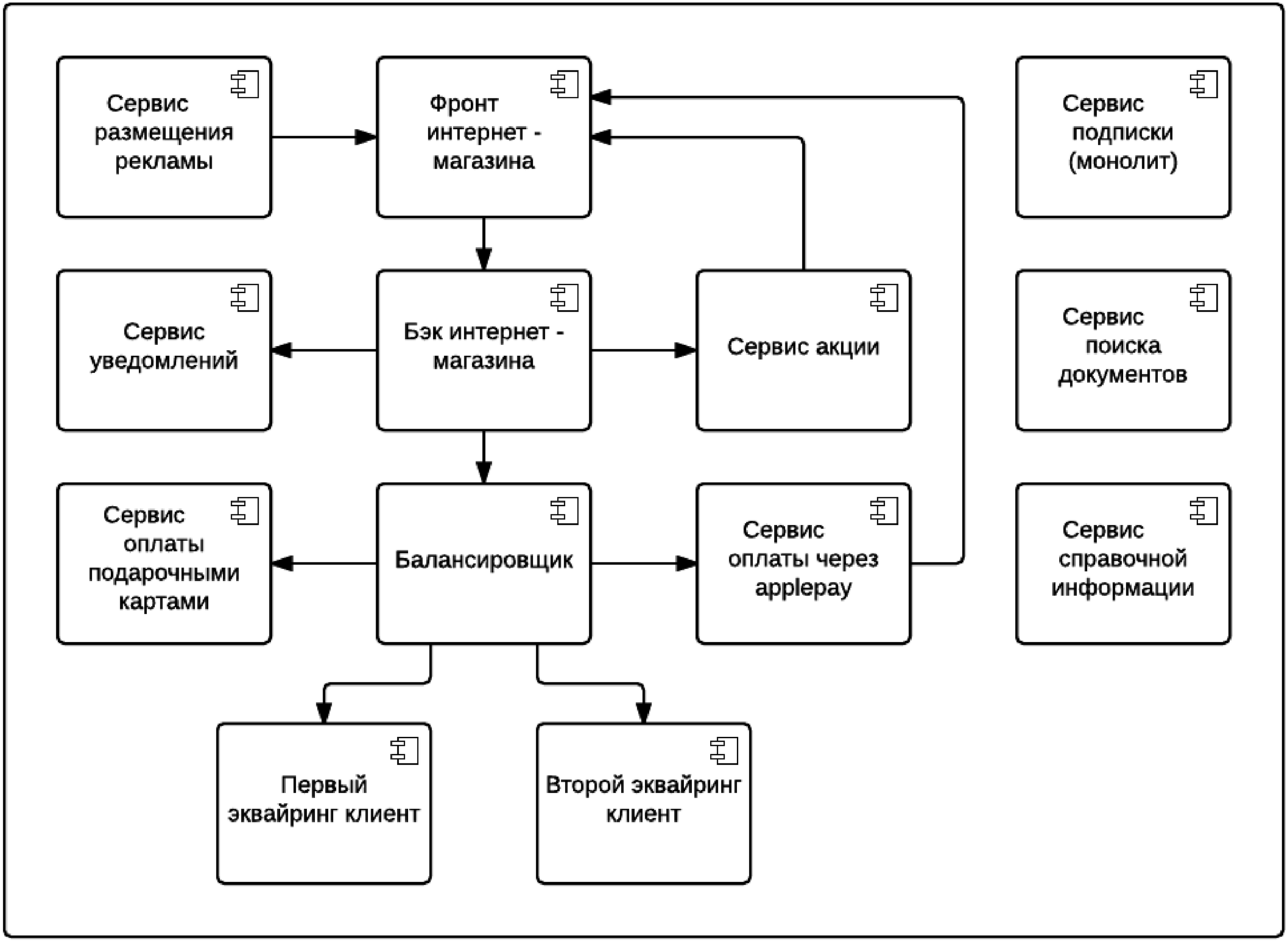
Найти небольшую задачу =>

Быстро разобраться =>

Быстро выполнить =>

Раньше начать приносить пользу =>

Хороший первый день на новой работе!


Сервис размещения рекламы
Фронт интернет - магазина
Сервис подписки (монолит)
Сервис уведомлений
Бэк интернет - магазина
Сервис акции
Сервис поиска документов
Сервис оплаты подарочными картами
Балансировщик
Сервис оплаты через applepay
Сервис справочной информации
Первый эквайринг клиент
Второй эквайринг клиент

А как там у Netflix?

- Более 600 сервисов
- Десятки тысяч нод EC2 в Amazon
- Миллиарды запросов ежедневно

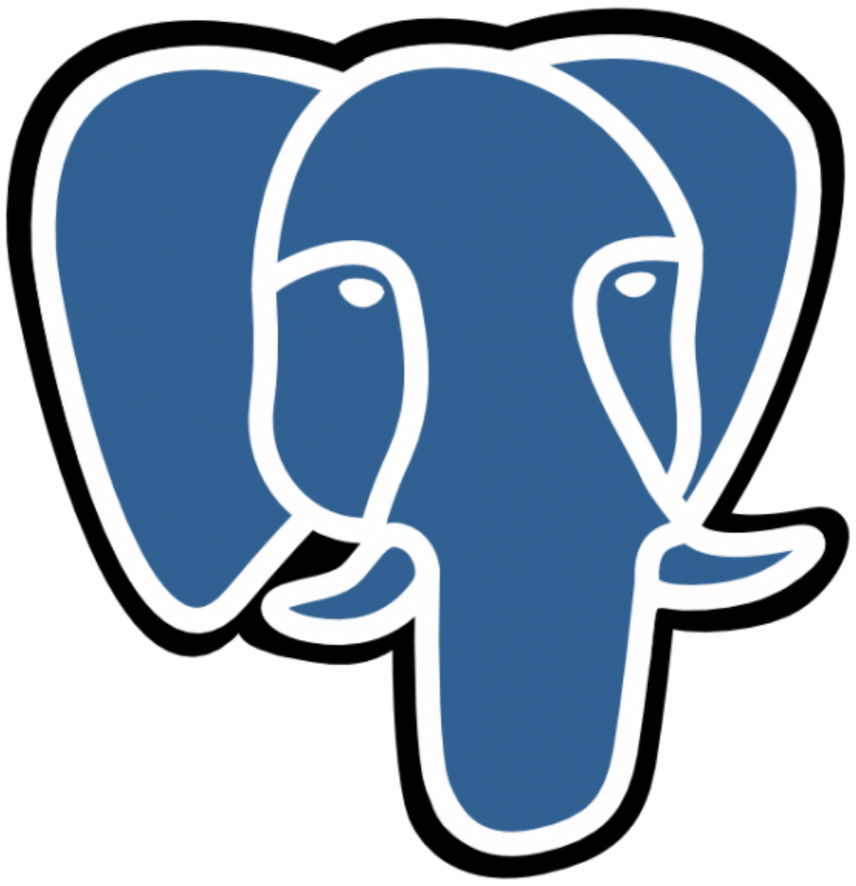
Команды ответственны за жизненный цикл разработанных приложений

- Разработка
- Релиз / деплой
- Обслуживание сервисов в проде

Ад перфекциониста

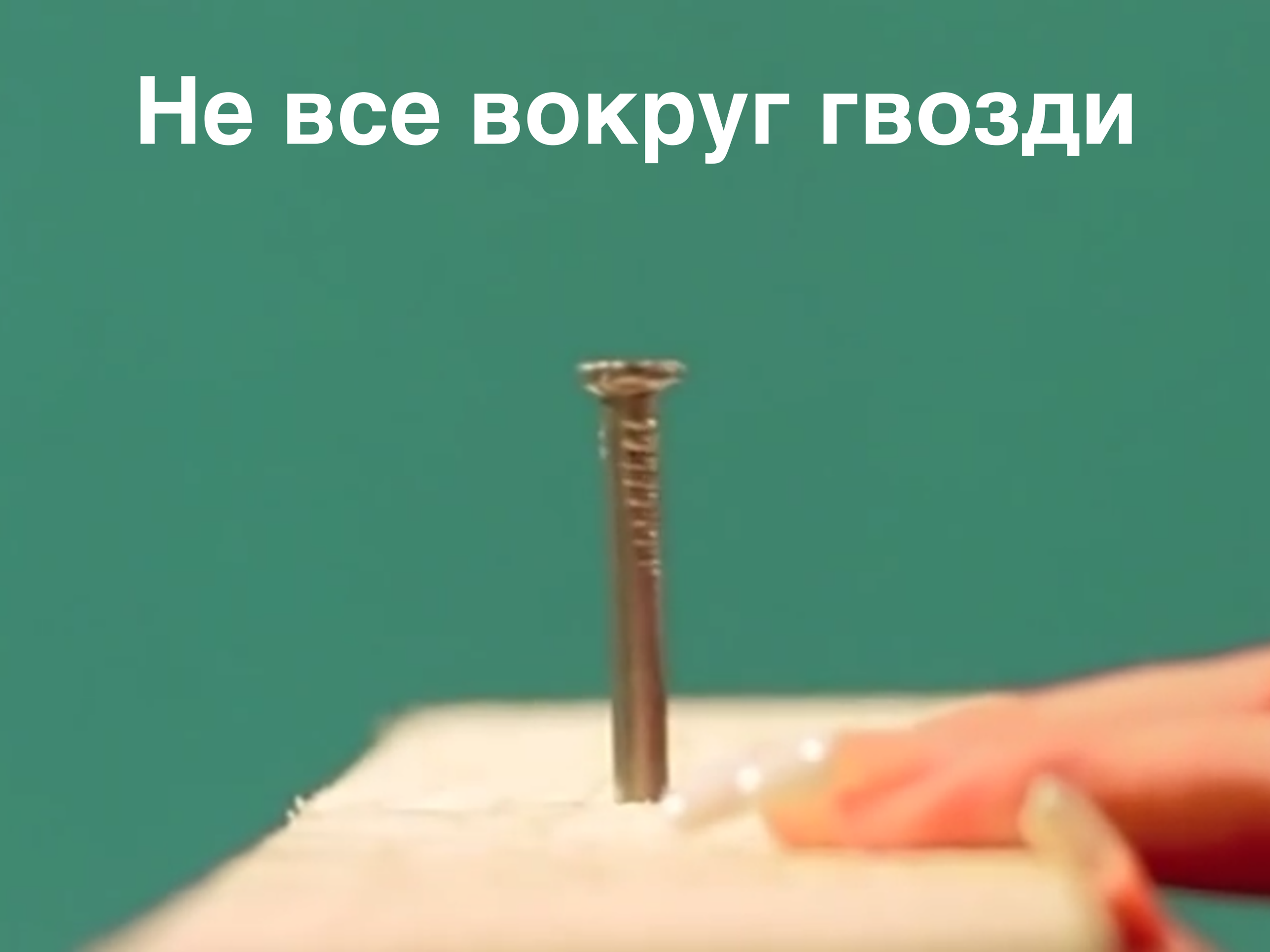
Я ЧЕЛОВЕК ПРОСТОЙ

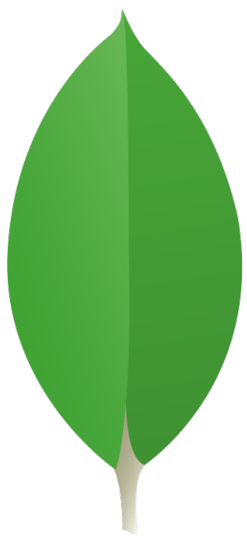
РАБОТАЕТ - НЕ ТРОГАЮ



PostgreSQL

Не все вокруг гвозди

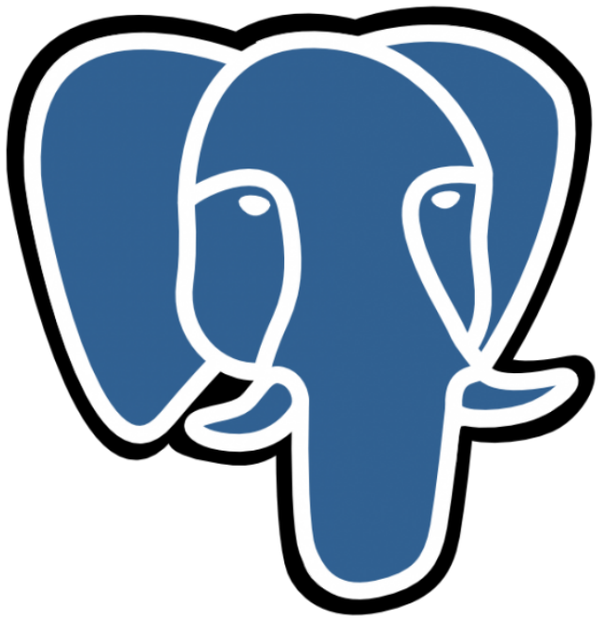




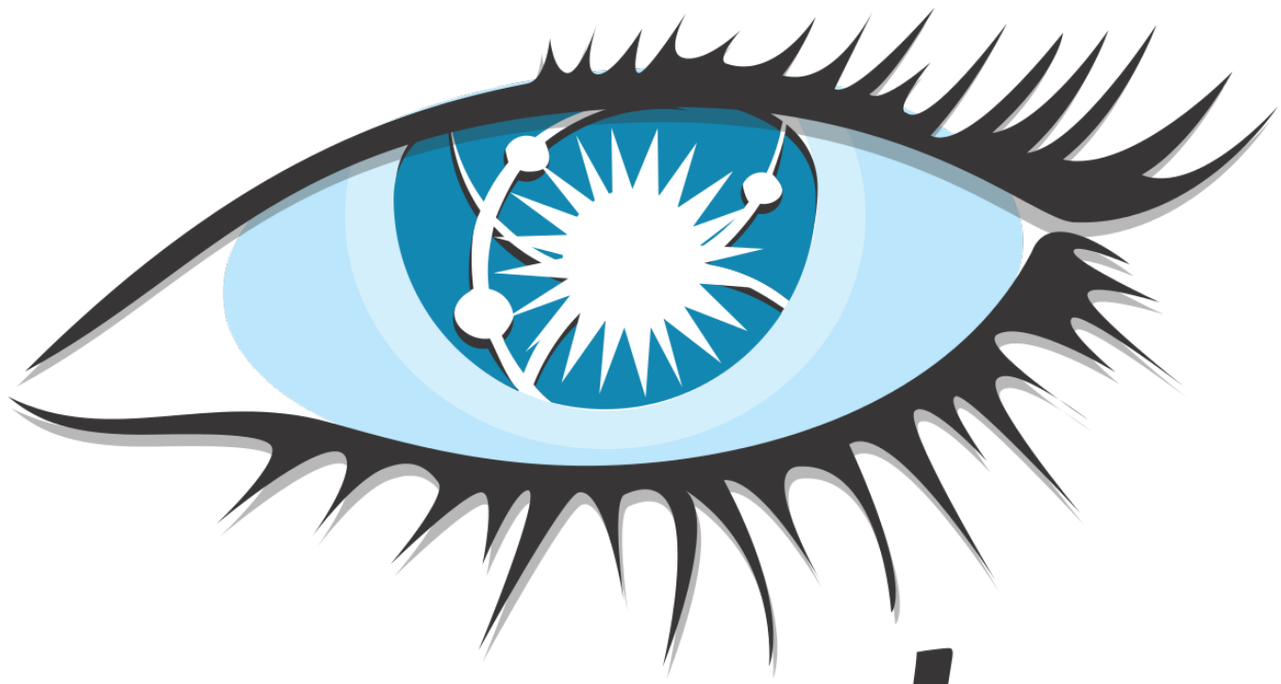
mongoDB


neo4j

Что лучше?



PostgreSQL



cassandra

Зависит от задачи

Мы попробовали

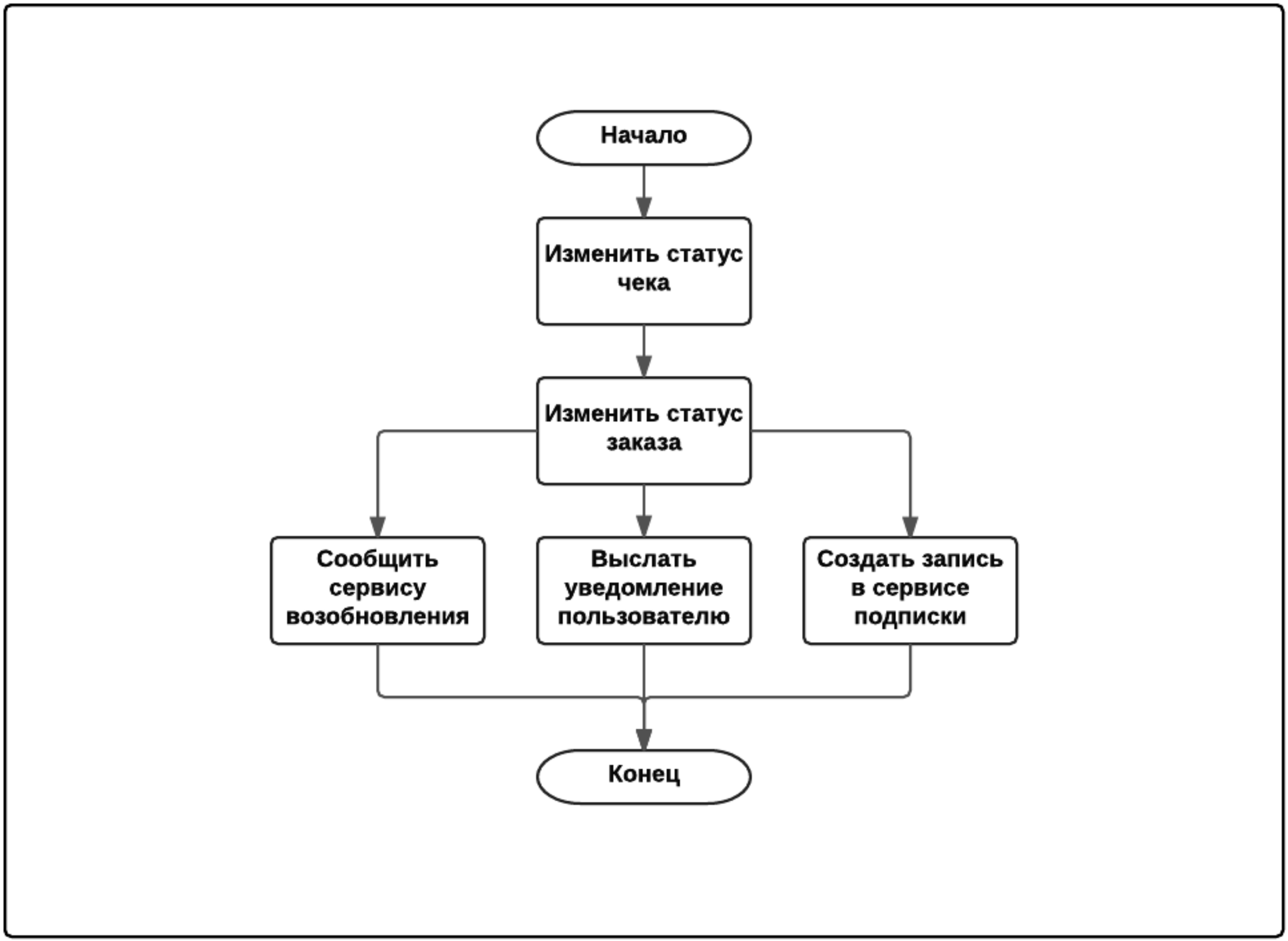
Java, Kotlin

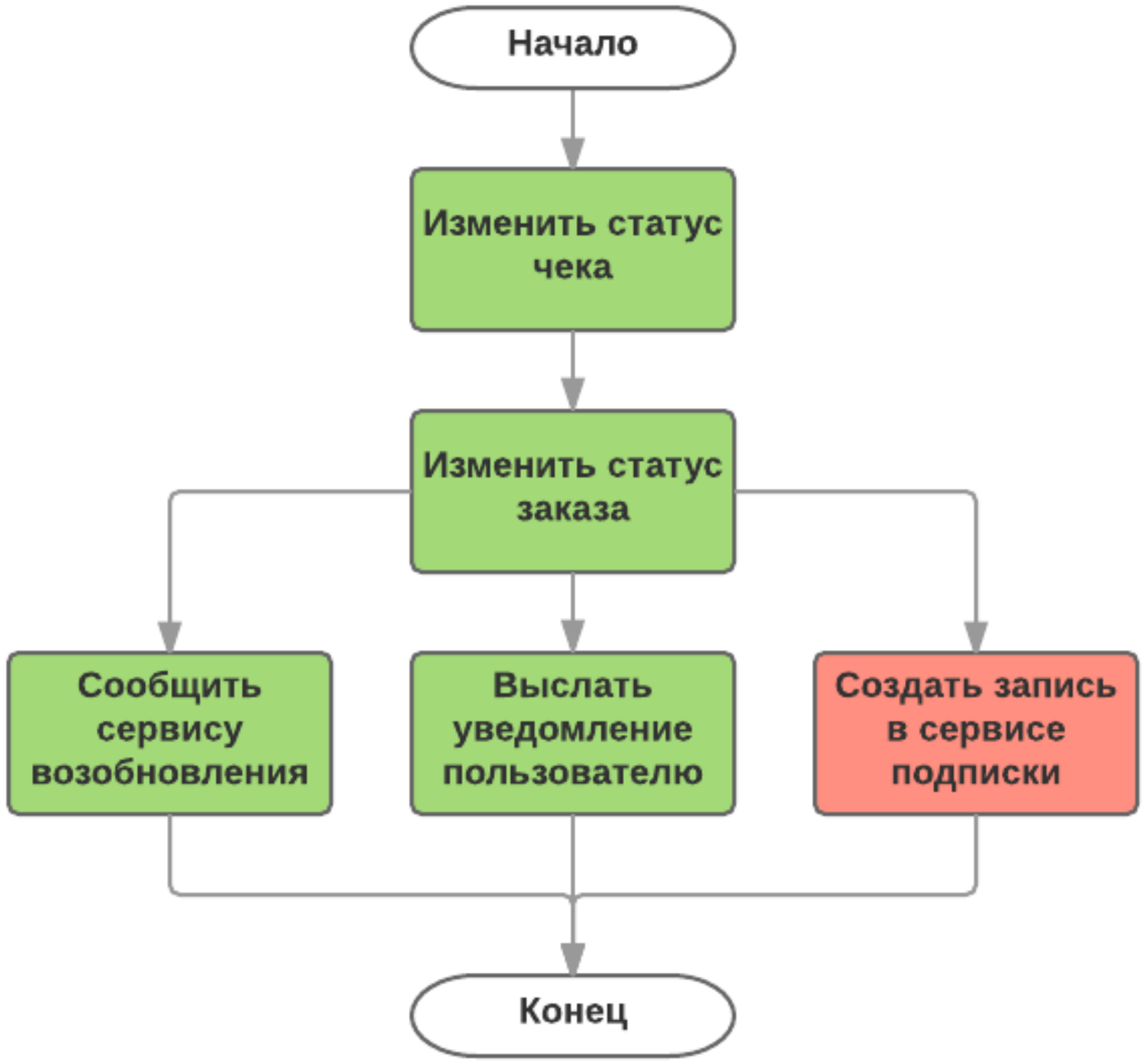
Hibernate, JOOQ

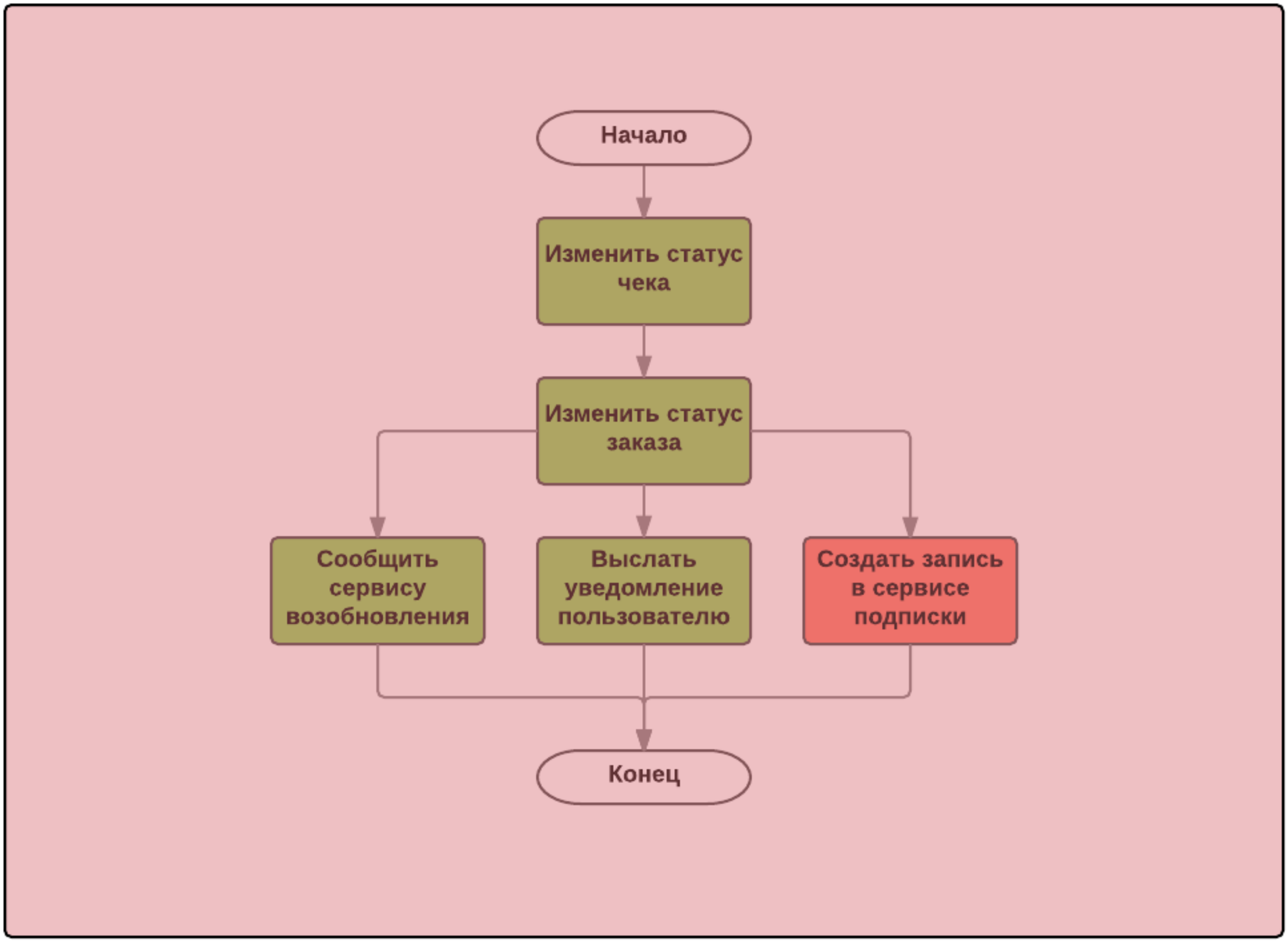
RabbitMQ, Kafka

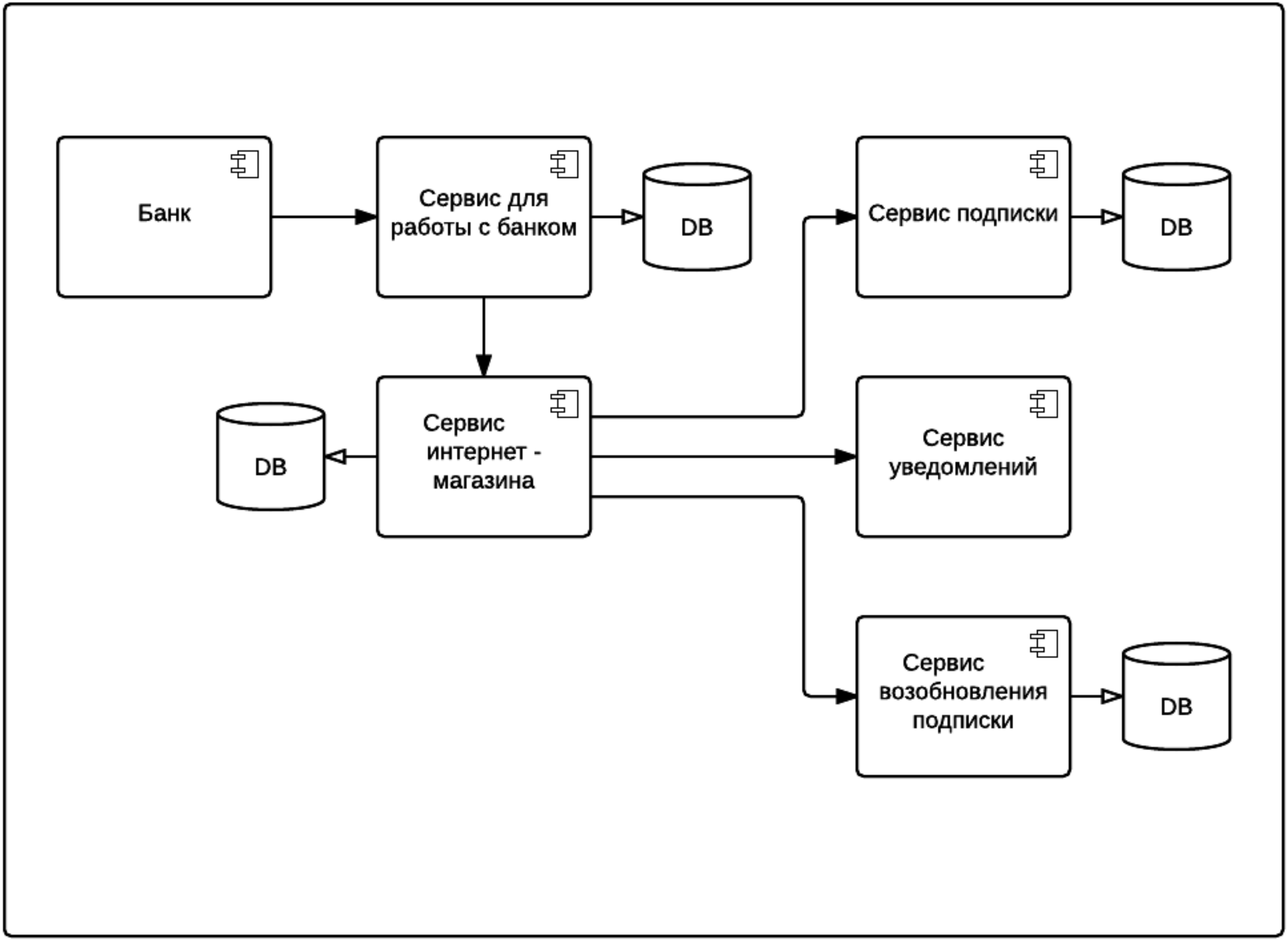
PostgreSQL, Redis, Hazelcast

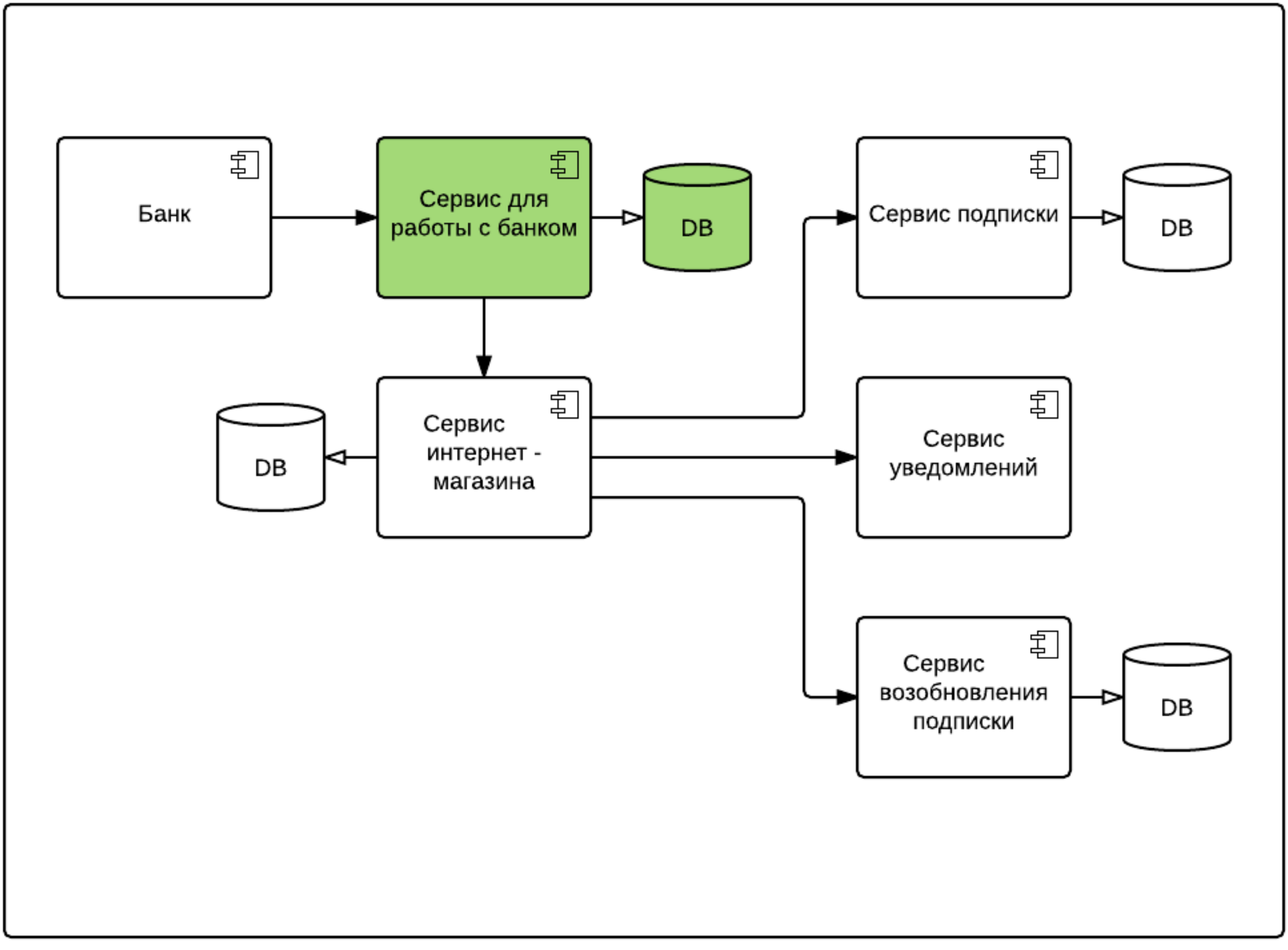
Jersey, Feign, Retrofit

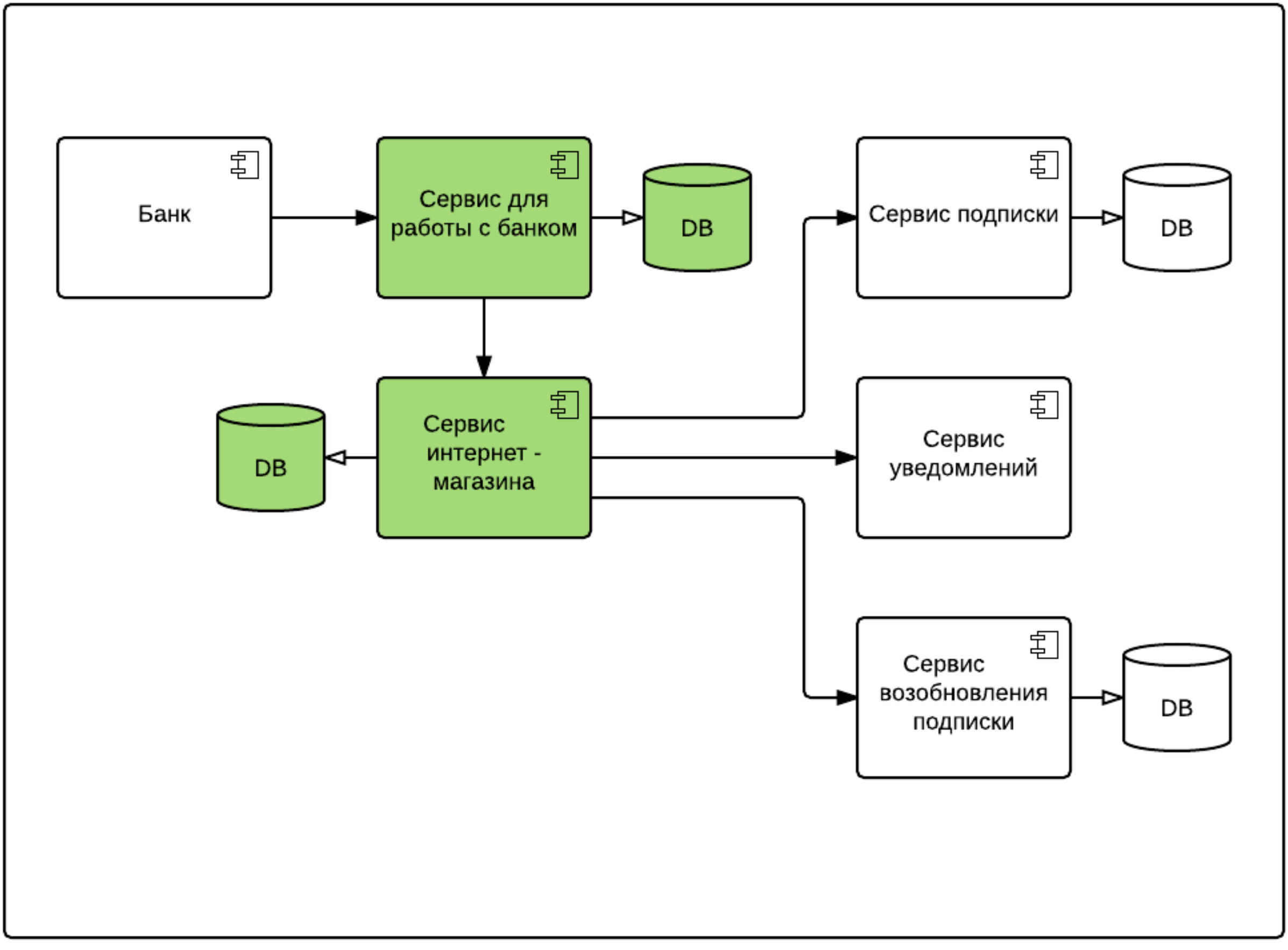

Начало
Изменить статус чека
Изменить статус заказа
Сообщить сервису возобновления
Выслать уведомление пользователю
Создать запись в сервисе подписки
Конец

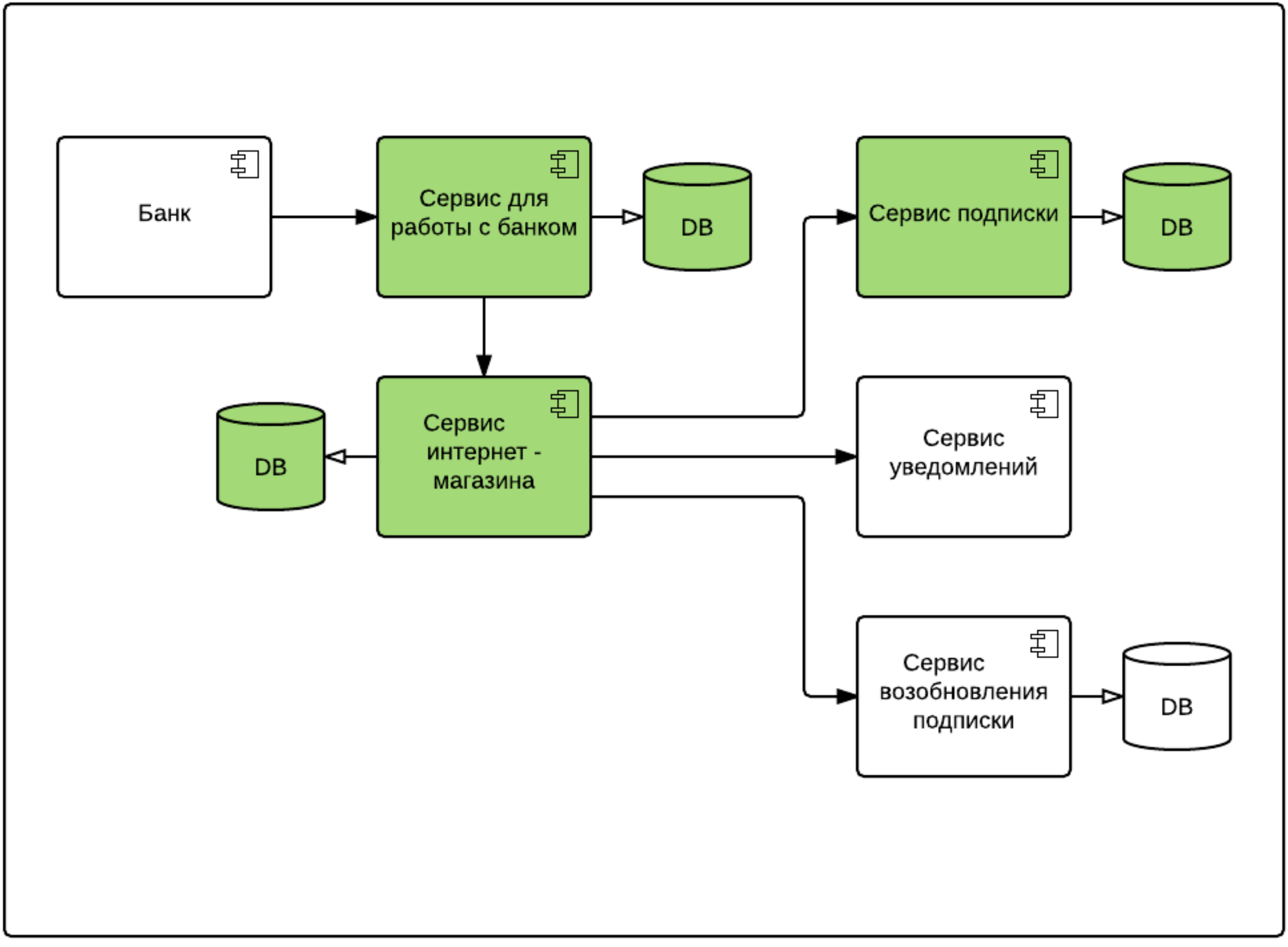

Начало
Изменить статус чека
Изменить статус заказа
Сообщить сервису возобновления
Выслать уведомление пользователю
Создать запись в сервисе подписки
Конец

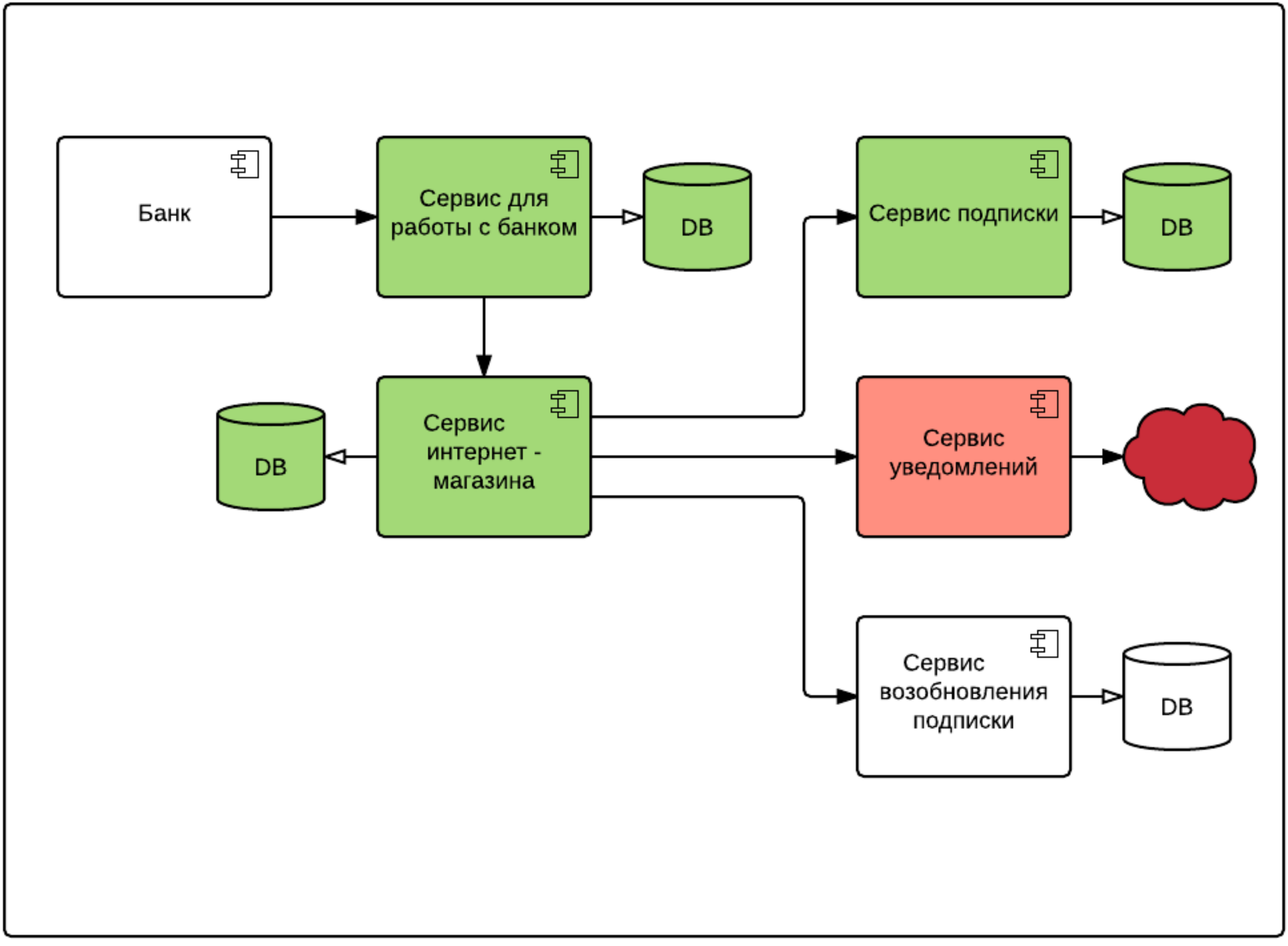

Начало
Изменить статус чека
Изменить статус заказа
Сообщить сервису возобновления
Выслать уведомление пользователю
Создать запись в сервисе подписки
Конец

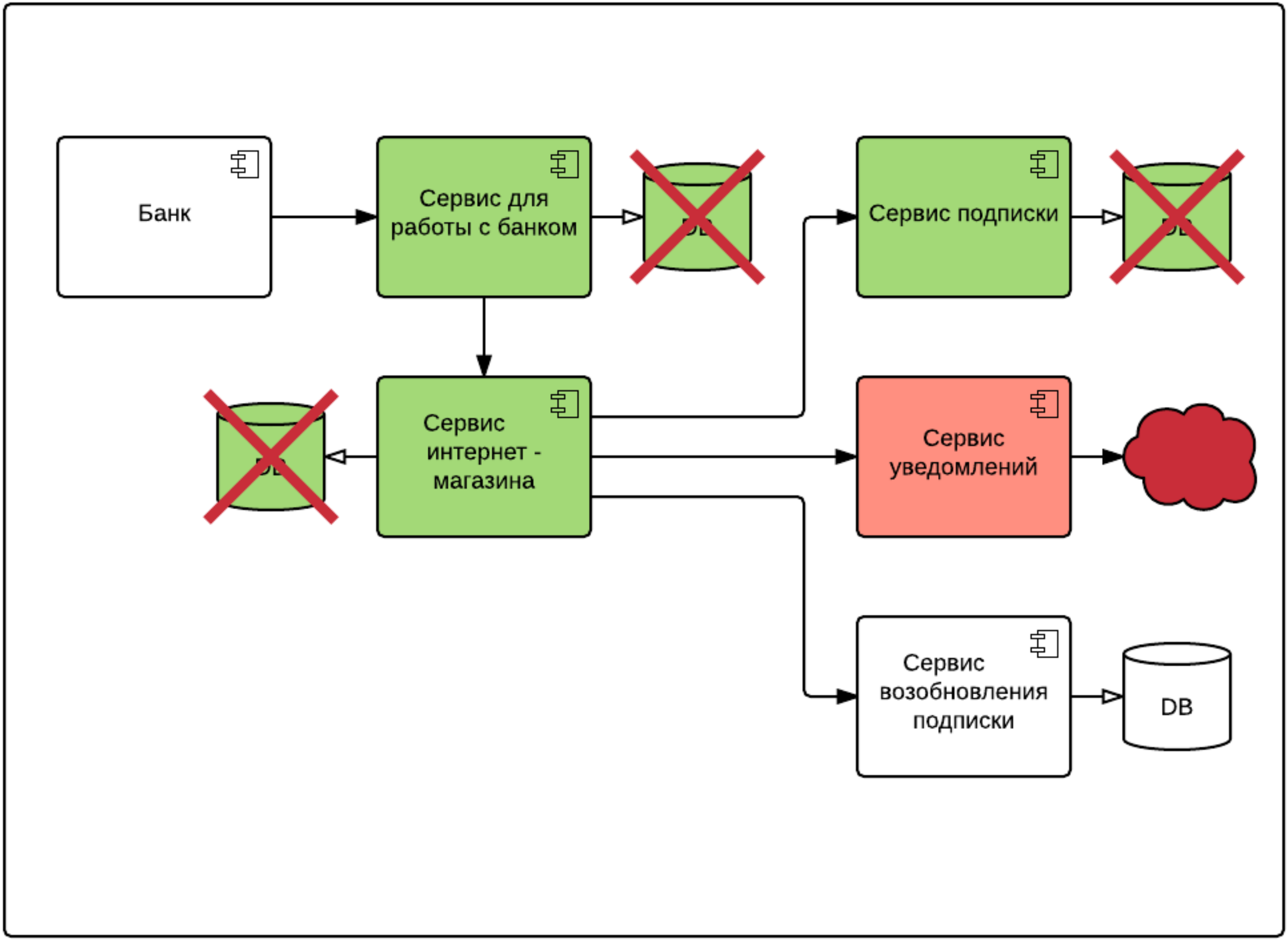

Банк
Сервис для работы с банком
DB
Сервис подписки
DB
DB
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

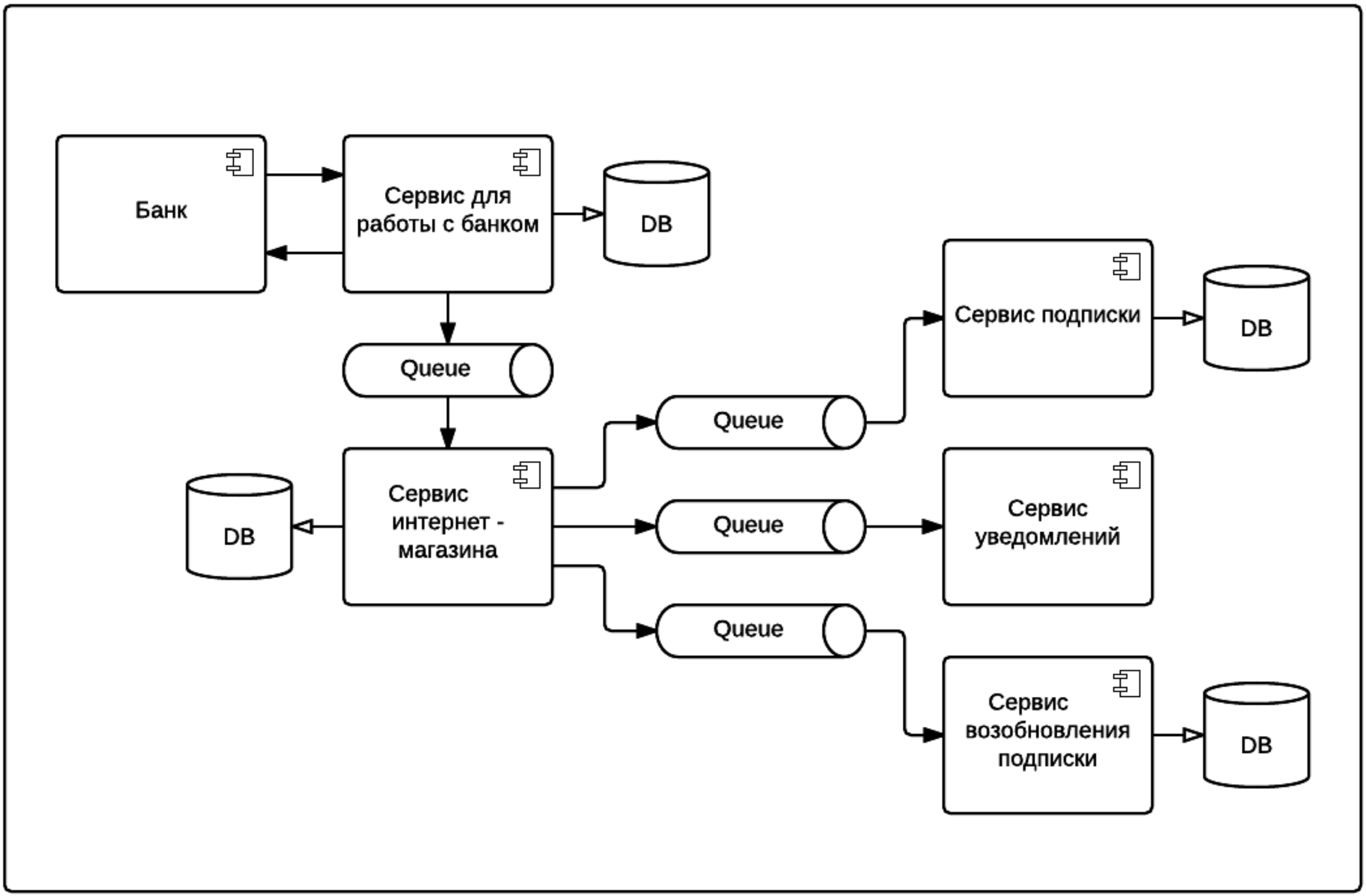

Банк
Сервис для работы с банком
DB
Сервис подписки
DB
DB
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

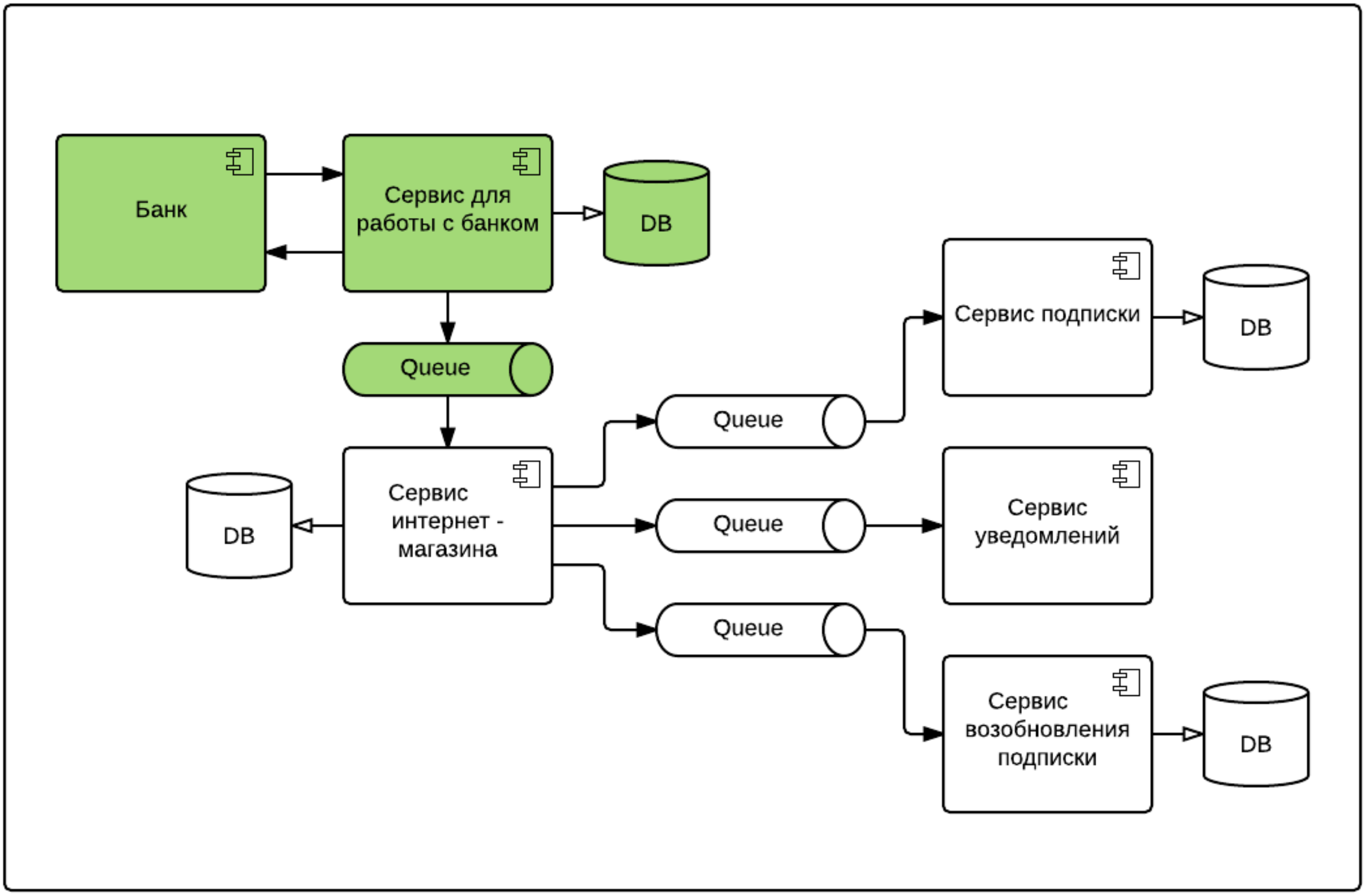

Банк
Сервис для работы с банком
DB
Сервис подписки
DB
DB
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

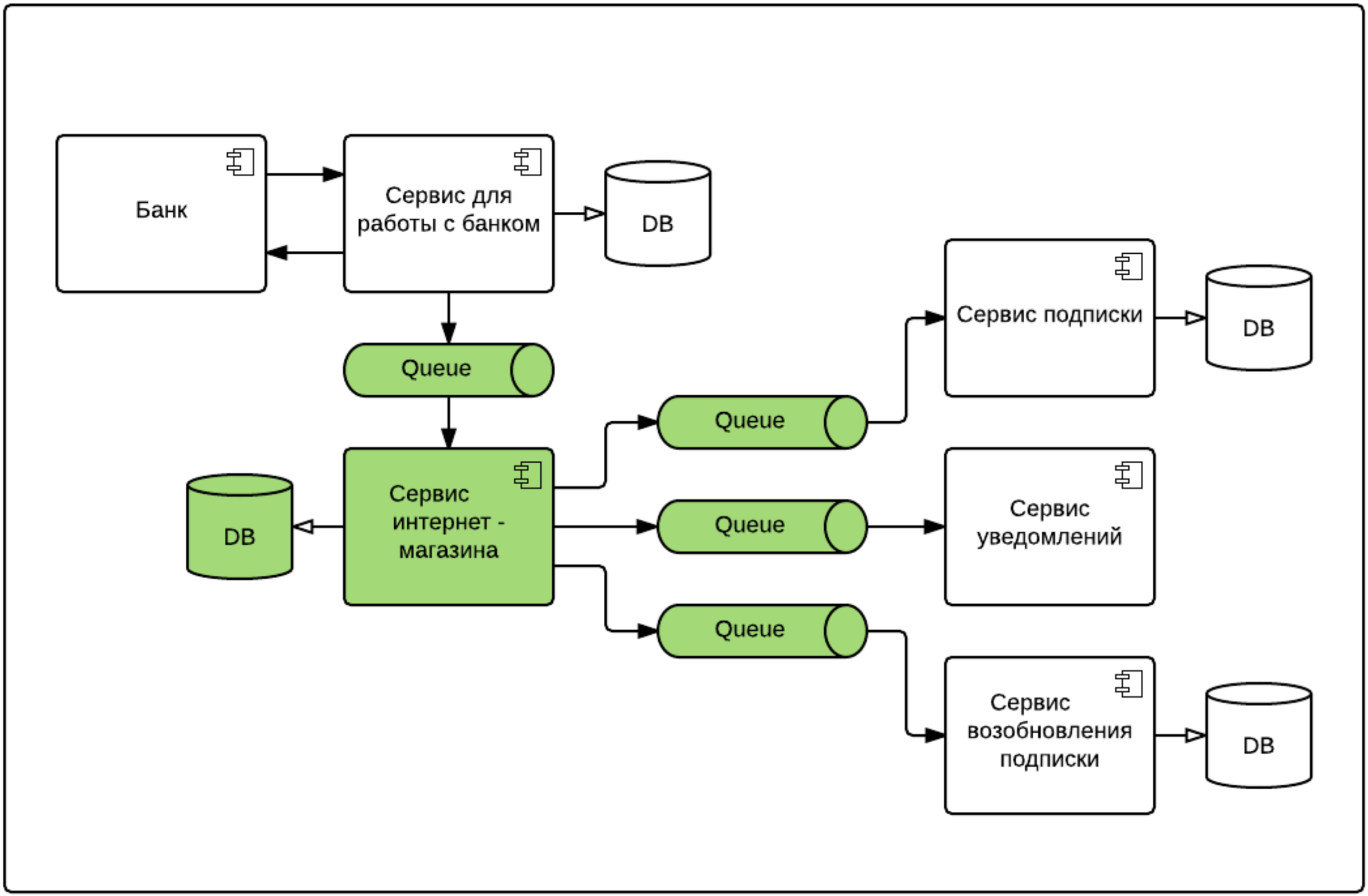

Банк
Сервис для работы с банком
DB
Сервис подписки
DB
DB
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

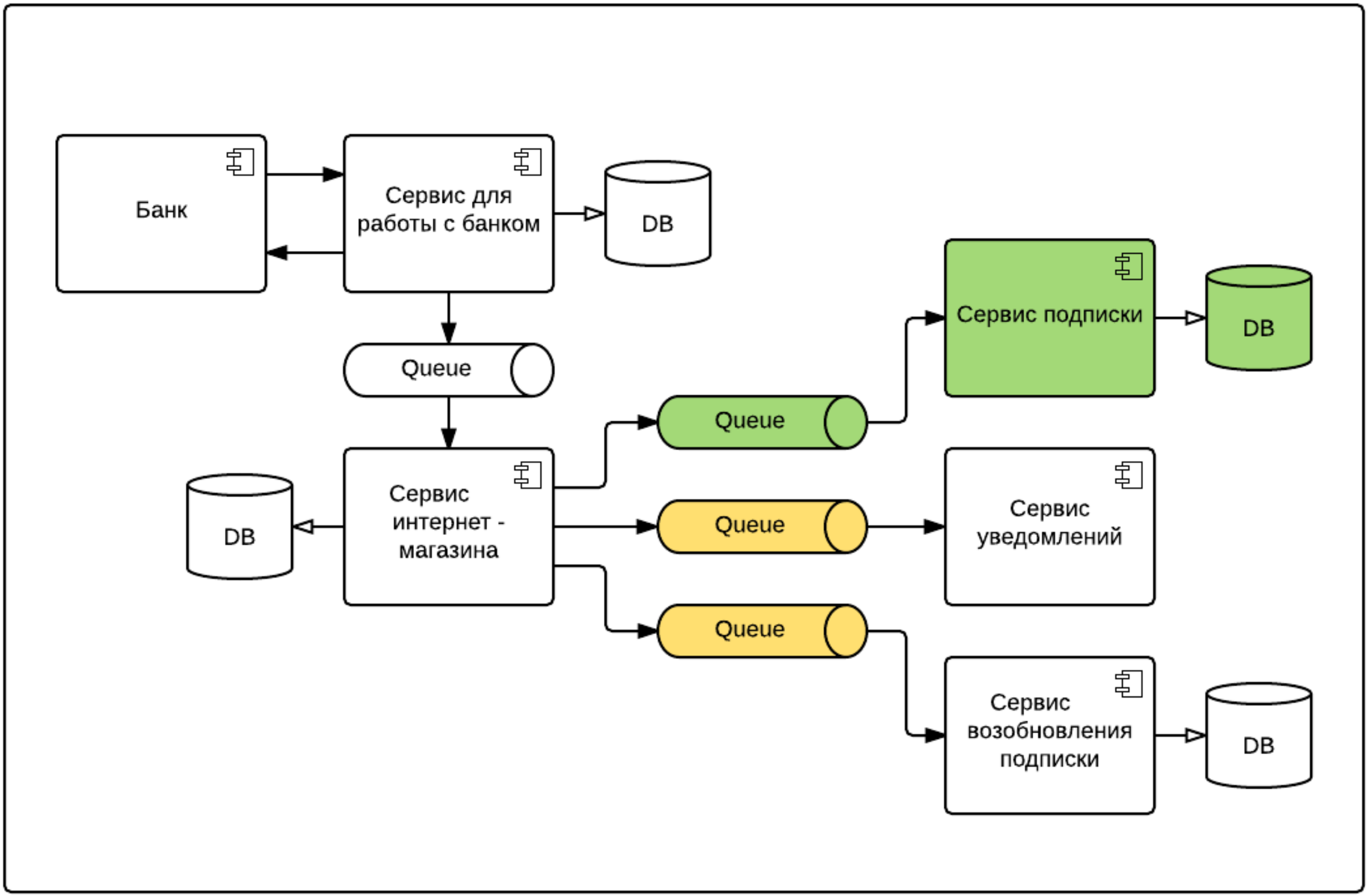

Банк
Сервис для работы с банком
DB
Сервис подписки
DB
DB
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

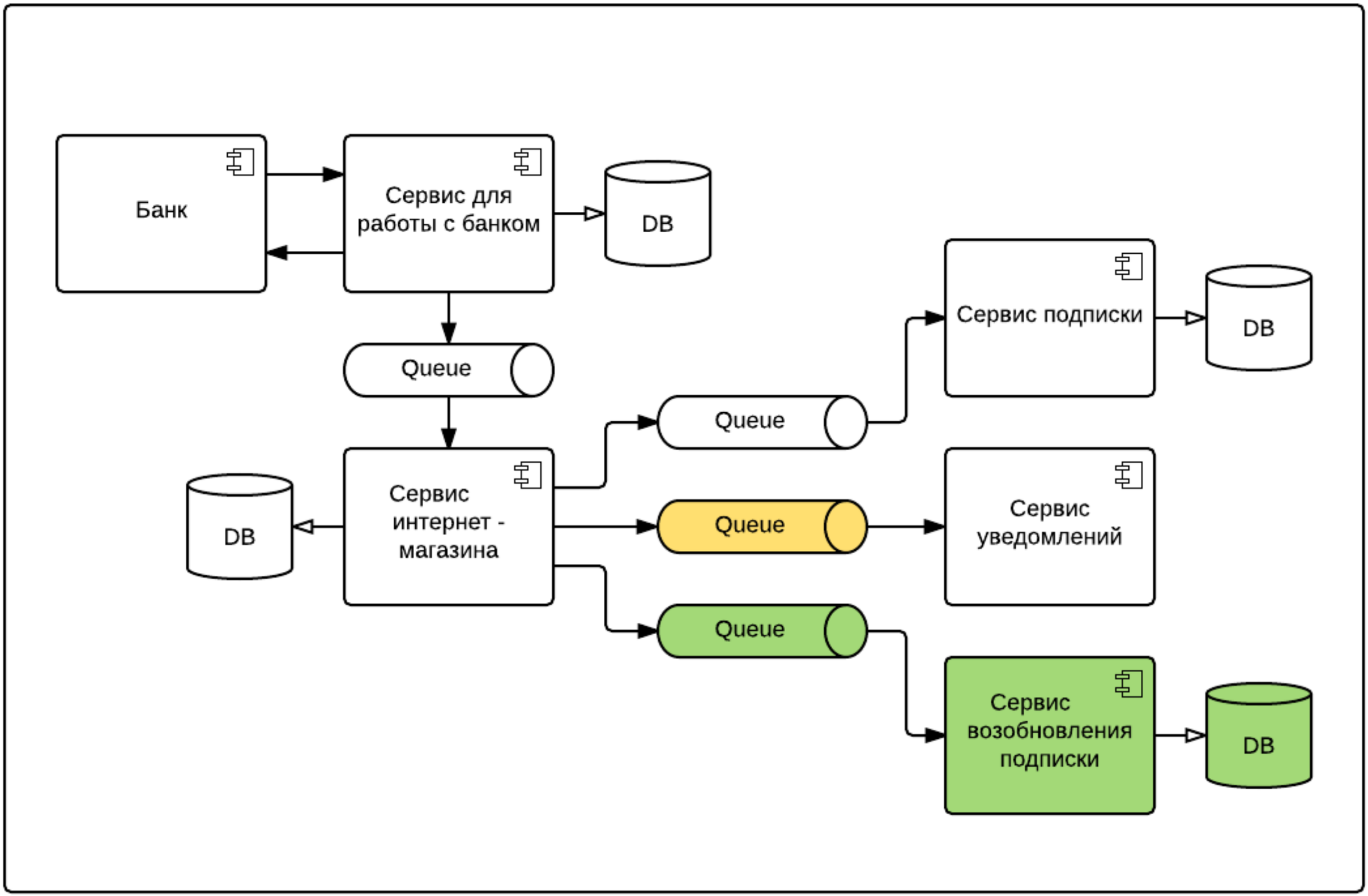

Банк
Сервис для работы с банком
Сервис подписки
Сервис интернет - магазина
Сервис уведомлений
Сервис возобновления подписки
DB

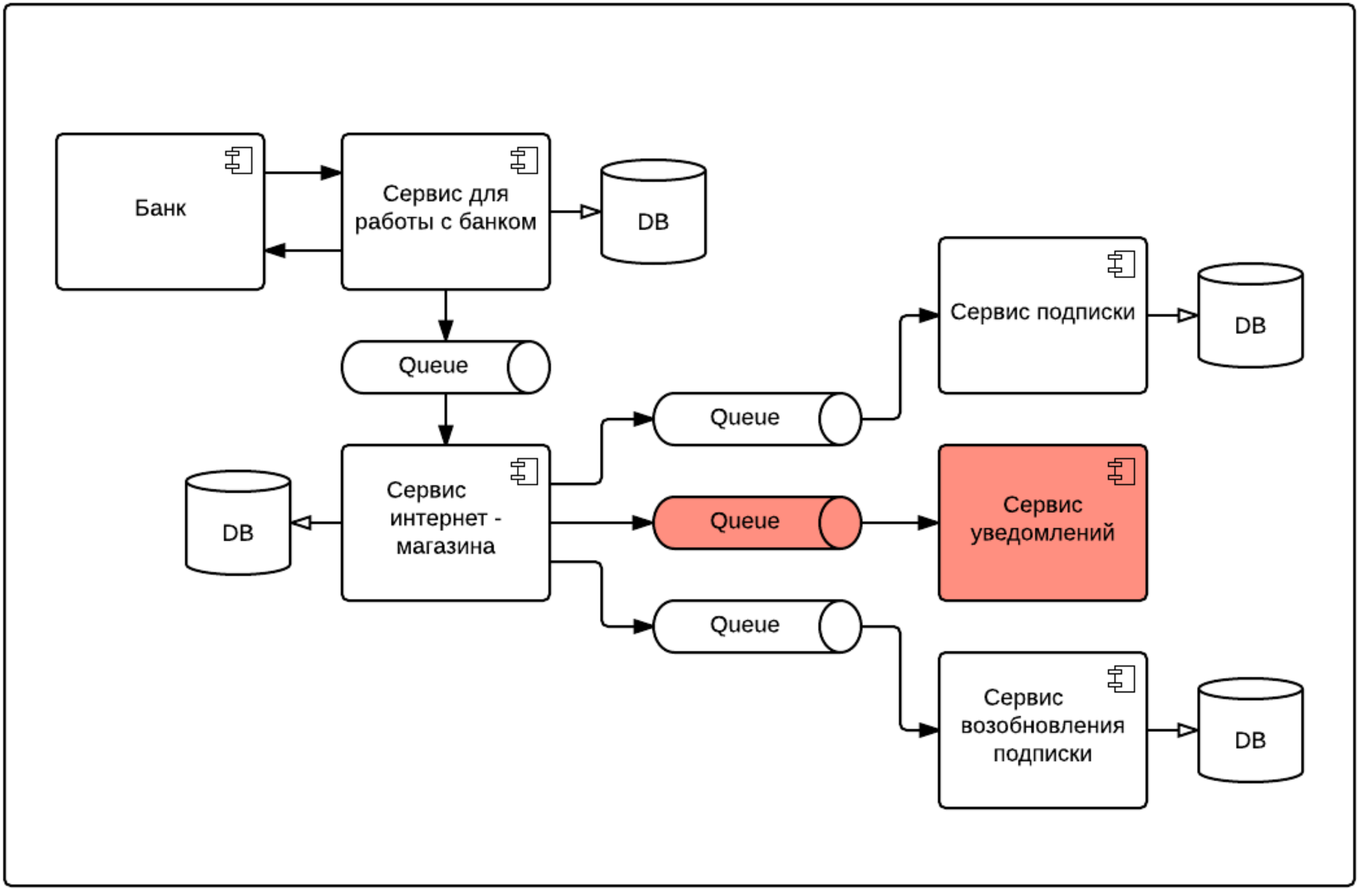

Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB

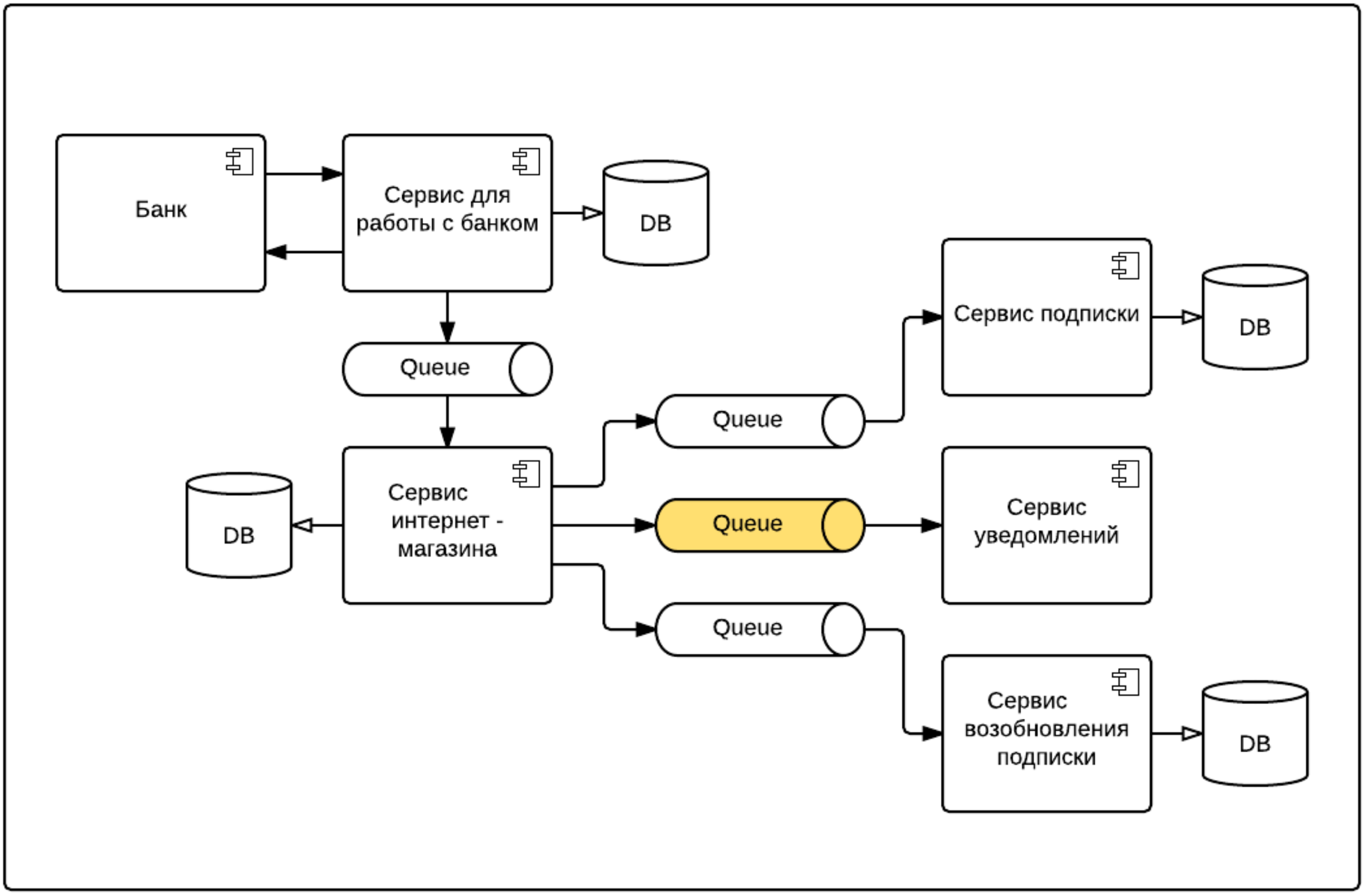

Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB

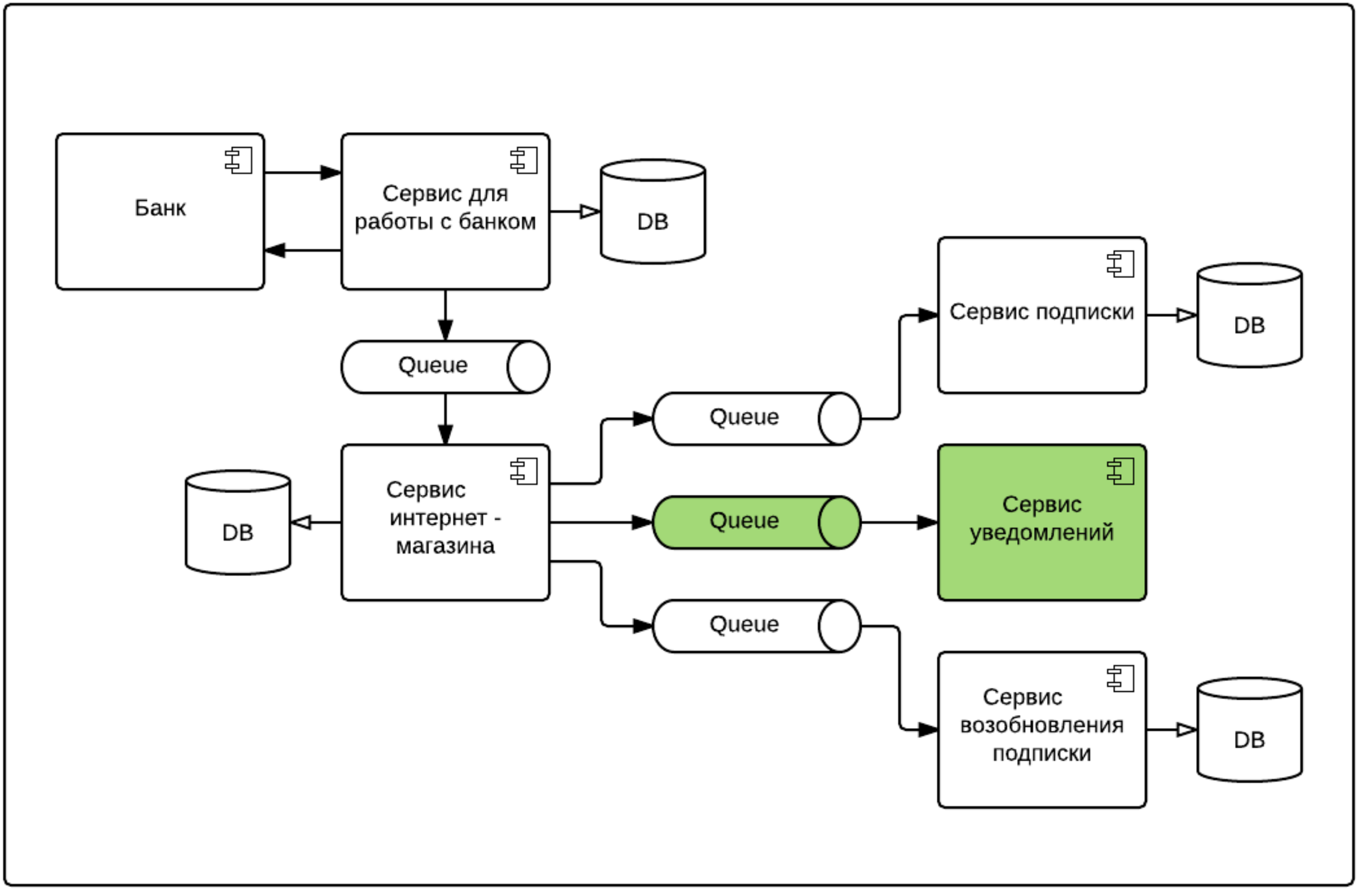

Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB

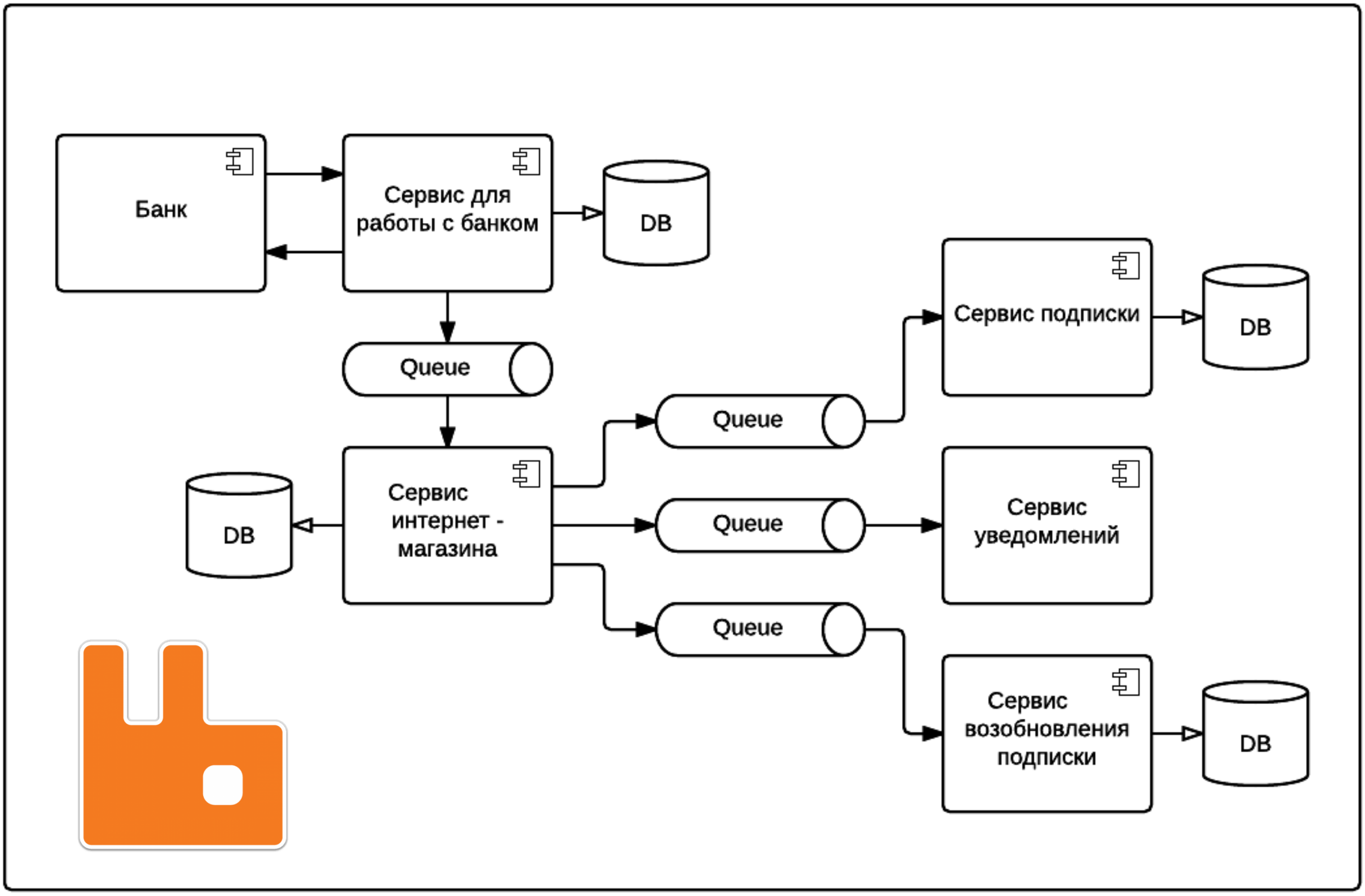

Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB


Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB


Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
Сервис интернет - магазина
DB
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB


Банк
Сервис для работы с банком
DB
Queue
Queue
Сервис подписки
DB
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB


Банк
Сервис для работы с банком
DB
Queue
Queue
Сервис подписки
DB
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB


Банк
Сервис для работы с банком
DB
Queue
Сервис подписки
DB
Queue
DB
Сервис интернет - магазина
Queue
Сервис уведомлений
Queue
Сервис возобновления подписки
DB

Eventual consistency

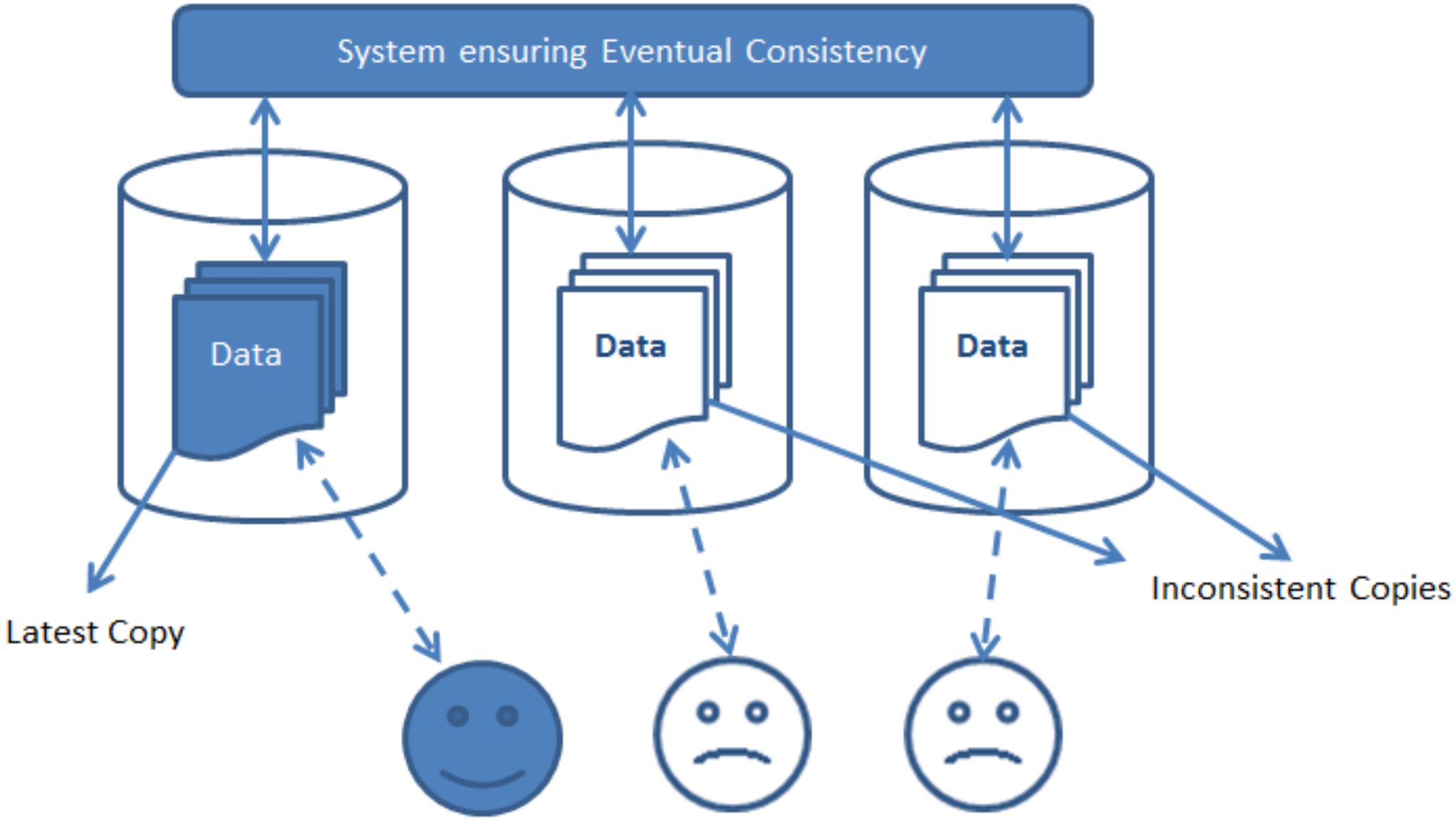


Figure showing a stale state where 2 copies of data are inconsistent with the latest one.

Минусы зоопарка технологий

- Не всегда существуют лучшие практики
- Высокая общая сложность системы
- Больше сервисов для обслуживания
- Сложнее управлять разработкой
- Высокий риск появления SHOK

Single Head Of Knowledge

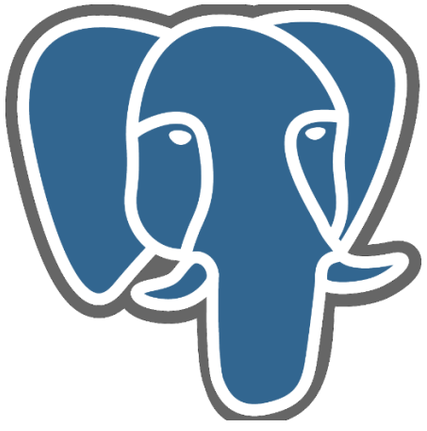
Дорога в ад для любого проекта

Плюсы свободы выбора инструментов

- Каждой задаче - лучший инструмент
- Новое взамен устаревшего
- Расширяет кругозор

+ 10 к количеству скиллов в резюме

Наш скромный стек



Platforms

Apache HTTP Server

Apache Tomcat

Bottle (Python)

...

Persistance

Cassandra

RDBMS (MySQL)

in-memory caches

Amazon S3

CDN

...

Programming Languages

Java

JavaScript

Groovy

Clojure

Scala

Dart

Python

Ruby

- НЕ ХОДИ ТУДА, ТАМ ТЕБЯ ЖДУТ НЕПРИЯТНОСТИ.

- НУ КАК ЖЕ ТУДА НЕ ХОДИТЬ? ОНИ ЖЕ ЖДУТ.

Композитный ответ

Особо жирный JSON

Композитный JSON

http://catalogue/api/subscription/1234

```
1    {
2      "id": 1234,
3      "title": "Новости кровавого энтерпрайза",
4      "country": "Российская Федерация",
5      "language": "Русский",
6      "priceGroups": [...],
21     "regionalOptions": [...],
761    "indexes": [...],
2825   "themes": [...],
2830   "cover": {"id": 58917...},
2838   "documents": {"id": 100974...},
2883   "versions": [...],
2934   "regionGroups": [...]
3024   }
```

Какая цена для издания в определенном регионе?

```
{
  "id": 1234,
  "title": "Новости кровавого энтерпрайза",
  "country": "Российская Федерация",
  "language": "Русский",
  "priceGroups": [
    {
      "id": 7432,
      "title": "Каталожная цена",
      "prices": {
        "П1712": {
          "priceMin": 350.00
        }
      }
    }
  ],
  "regionalOptions": [
    {
      "tcfpsCode": "44",
      "priceGroupId": 7432
    },
```

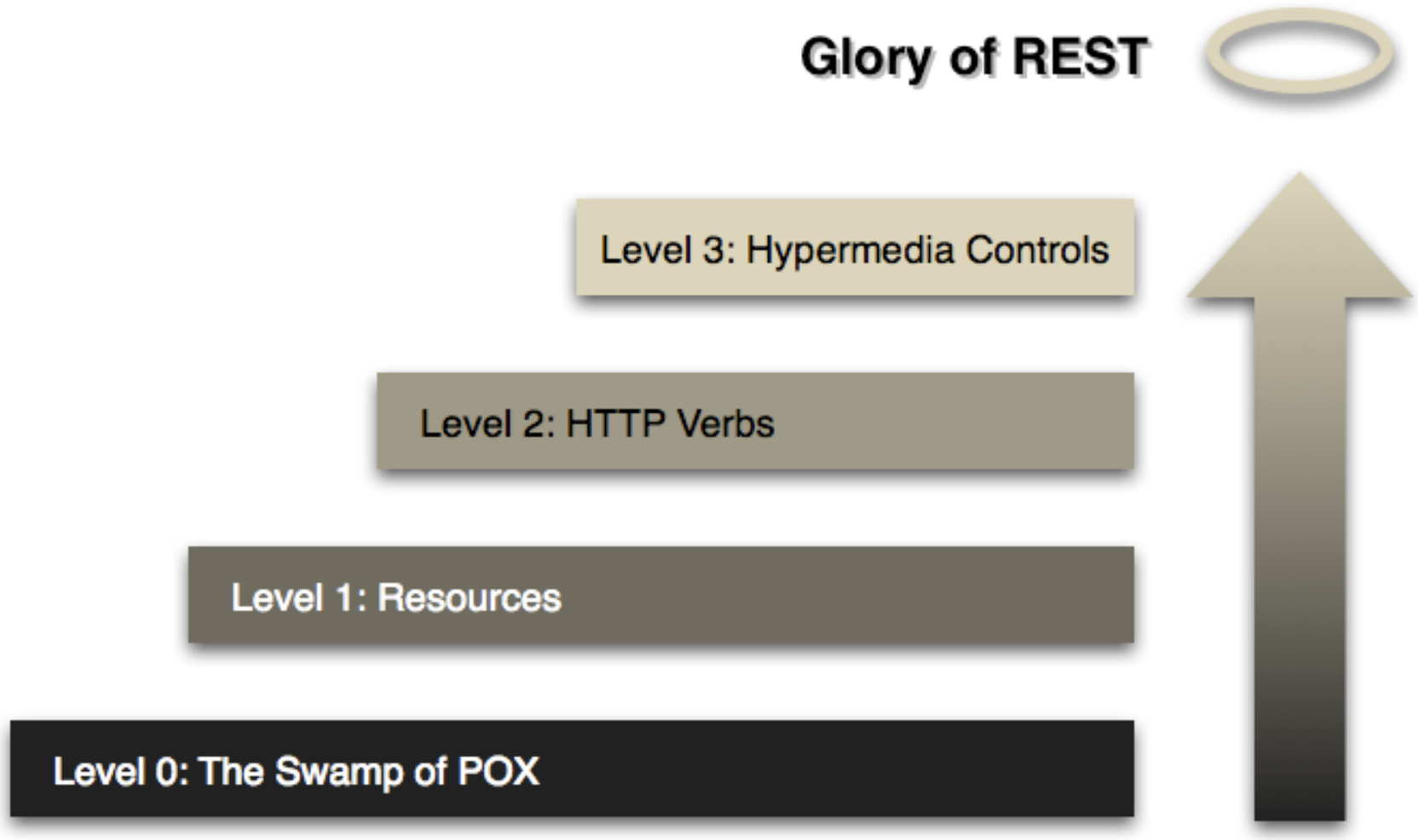
http://catalogue/api/subscription/1234/prices


```
[
  {"tcfpsCode":"908", "priceMin":350.00},
  {"tcfpsCode":"909", "priceMin":340.00},
  {"tcfpsCode":"910", "priceMin":325.00}
]
```

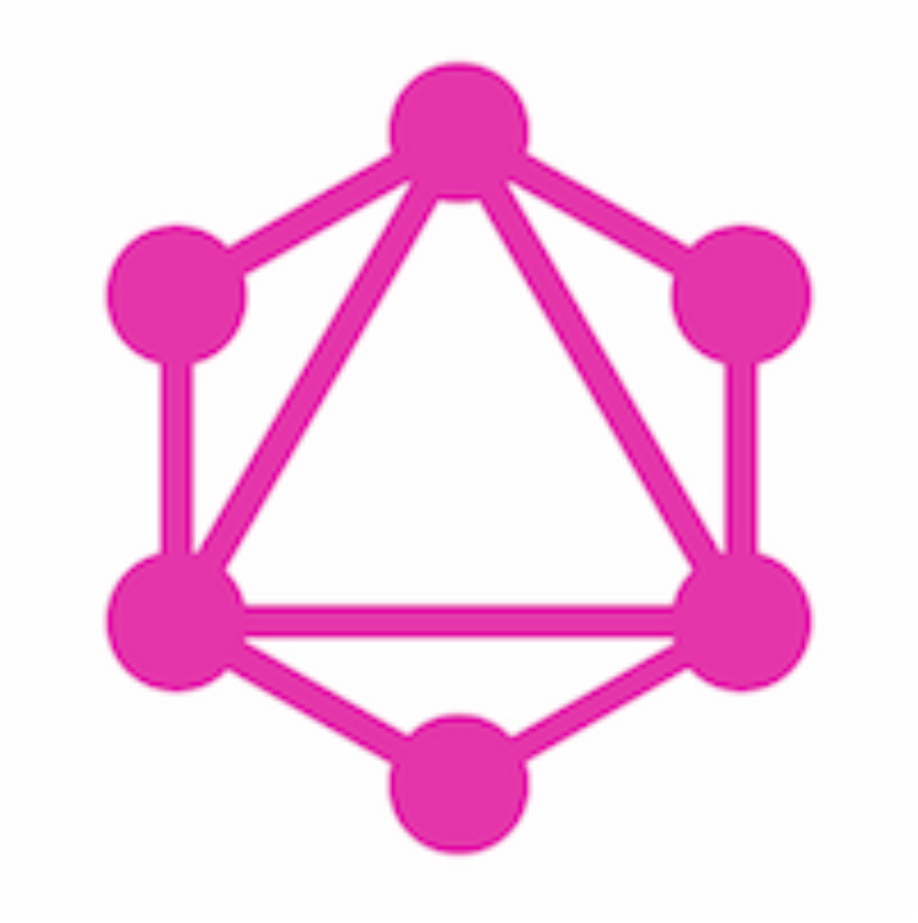
http://catalogue/api/subscription/1234/region/908/priceMin


```
{
  "priceMin":350.00
}
```

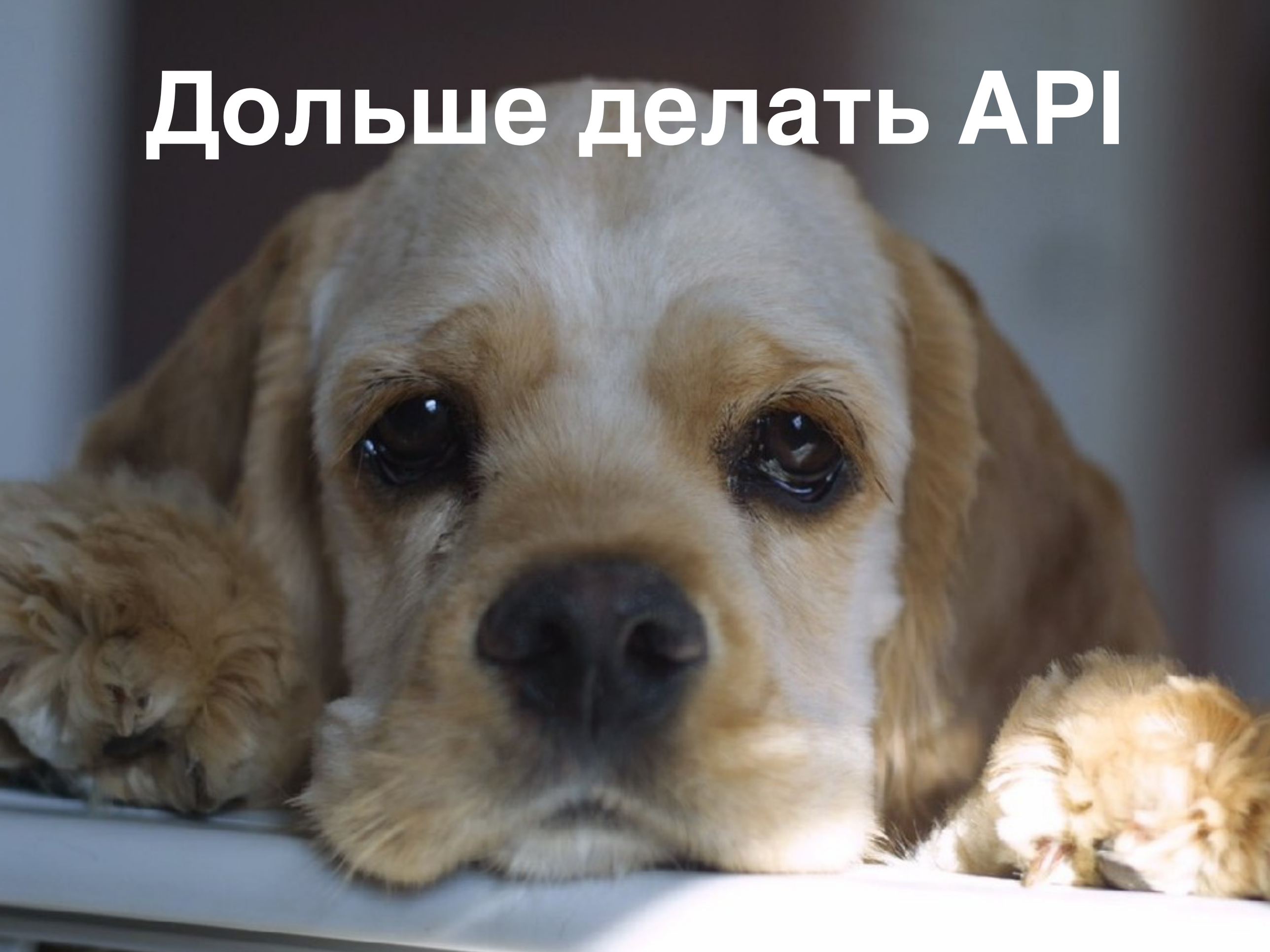
Richardson maturity model



GraphQL



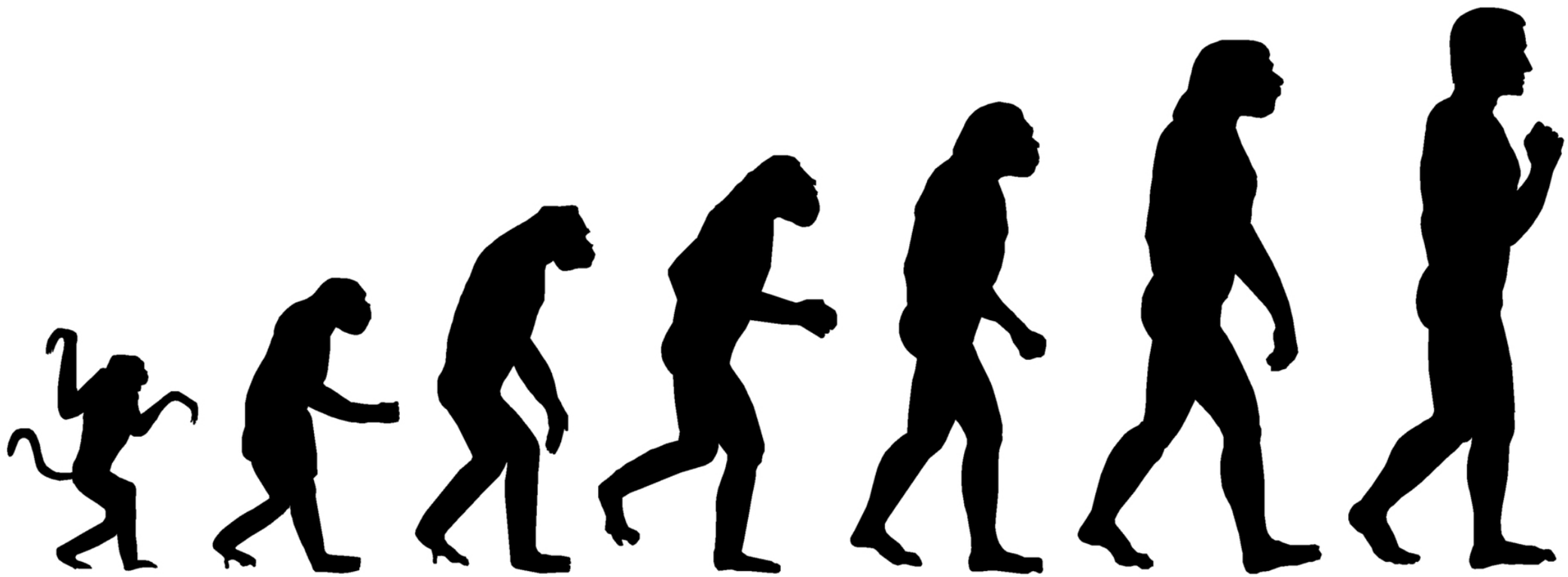
Дольше делать API



- **Результат качественнее**
- **Экономия трафика**
- **Экономия ресурсов**

Вы приблизитесь к искусству создания красивого и функционального API

Эволюция API



```
{
  "date":"10.10.2016",
  "id":1994809,
  "name":"Vasiliy",
  "role":"Manager",
  "age":25,
  "quality":"success"
}
```

```
{
  "date":"12.10.2016",
  "id":"49002343-7295-478d-bc1b-072fd46c22ba"
  "name":"Alonya",
  "role":"Manager",
  "age":22,
  "quality":"very success"
}
```

Серьезно?!

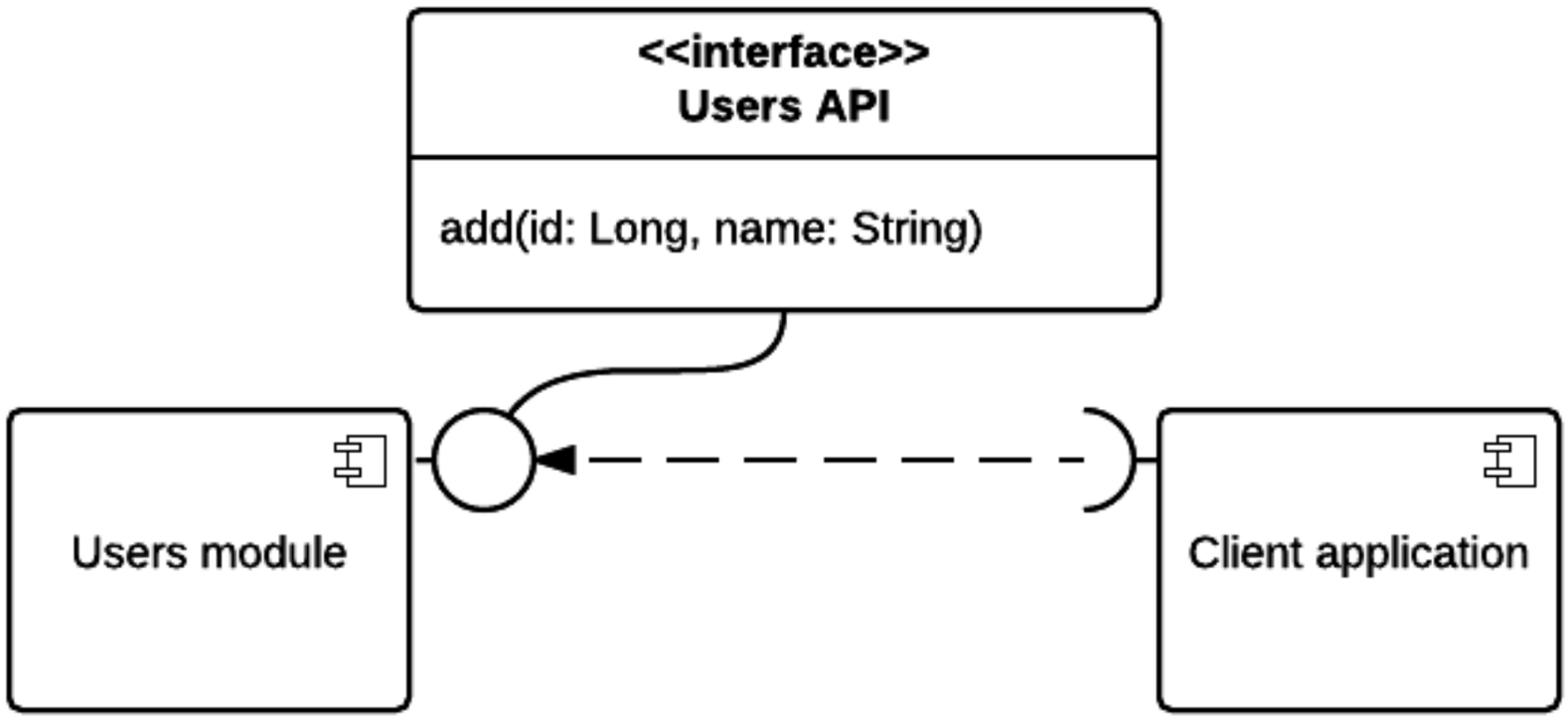
Генератор проблем - изменение API

- Изменяется метод доступа
- Изменяются имена полей
- Изменяются типы полей

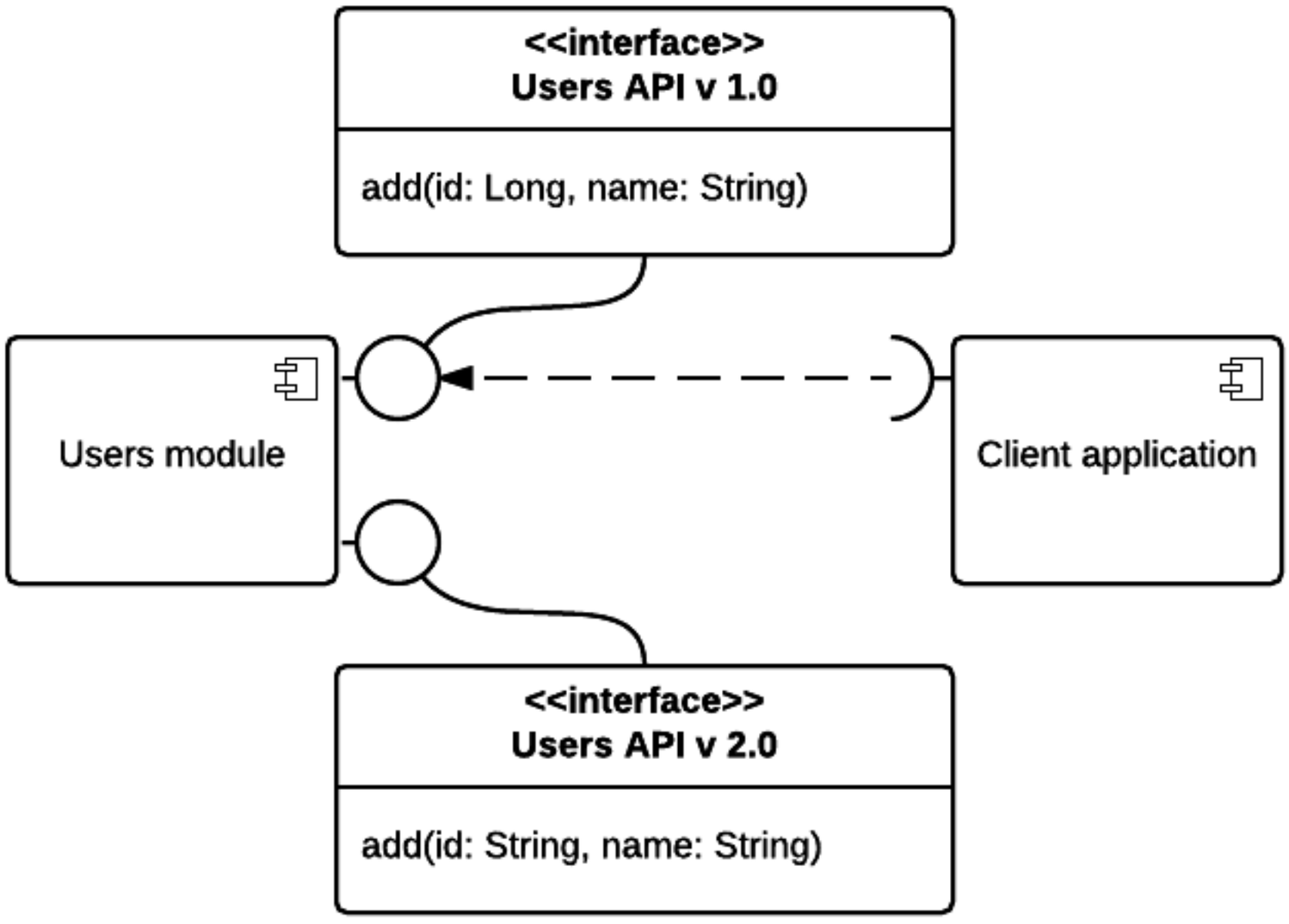
Что хотелось бы?

- Вовремя узнавать об изменениях API внешних сервисов
- Иметь возможность поддерживать изменения в течение времени, не сразу

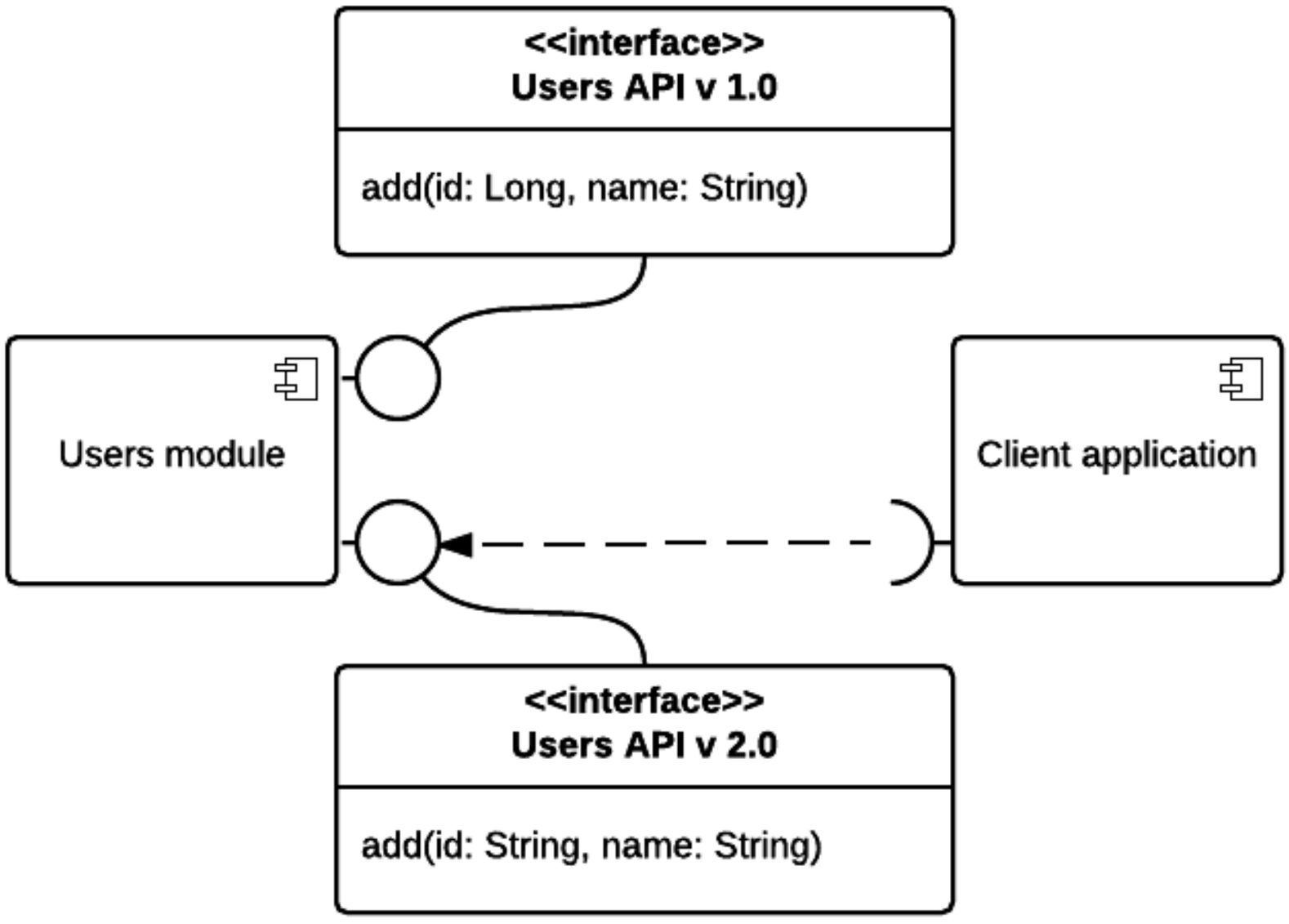
Expand and Contract



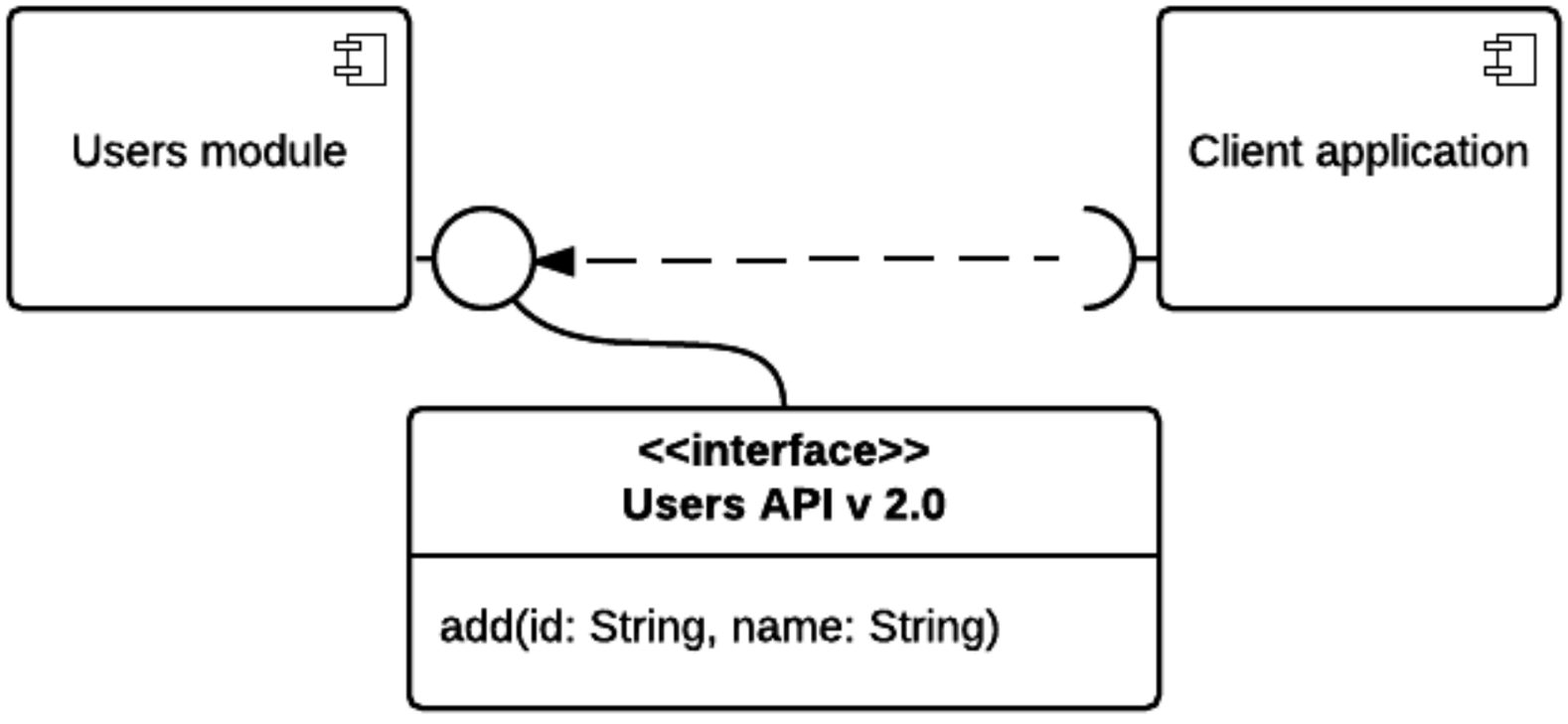
Expand and Contract



Expand and Contract



Expand and Contract




Dropbox

API v1 -> v2 migration timeline

- **28 июня 2016** - API v1 становится deprecated
- **28 сентября 2016** - ваши новые пользователи получат уведомление
- **28 марта 2017** - все ваши пользователи получат уведомление по email
- **28 июня 2017** - API v1 будет выключен

API v1 -> v2 migration guide

Note: API v2 only uses OAuth 2. The new endpoints do not have the `/2` prefix, but rather, just begin with `/oauth2`.

v1	v2
`/1/oauth2/authorize`	`/oauth2/authorize`
`/1/oauth2/token`	`/oauth2/token`
`/1/oauth2/token_from_oauth1`	*Not available;* Use v1 endpoint.
`/1/disable_access_token`	`/oauth2/token/revoke`

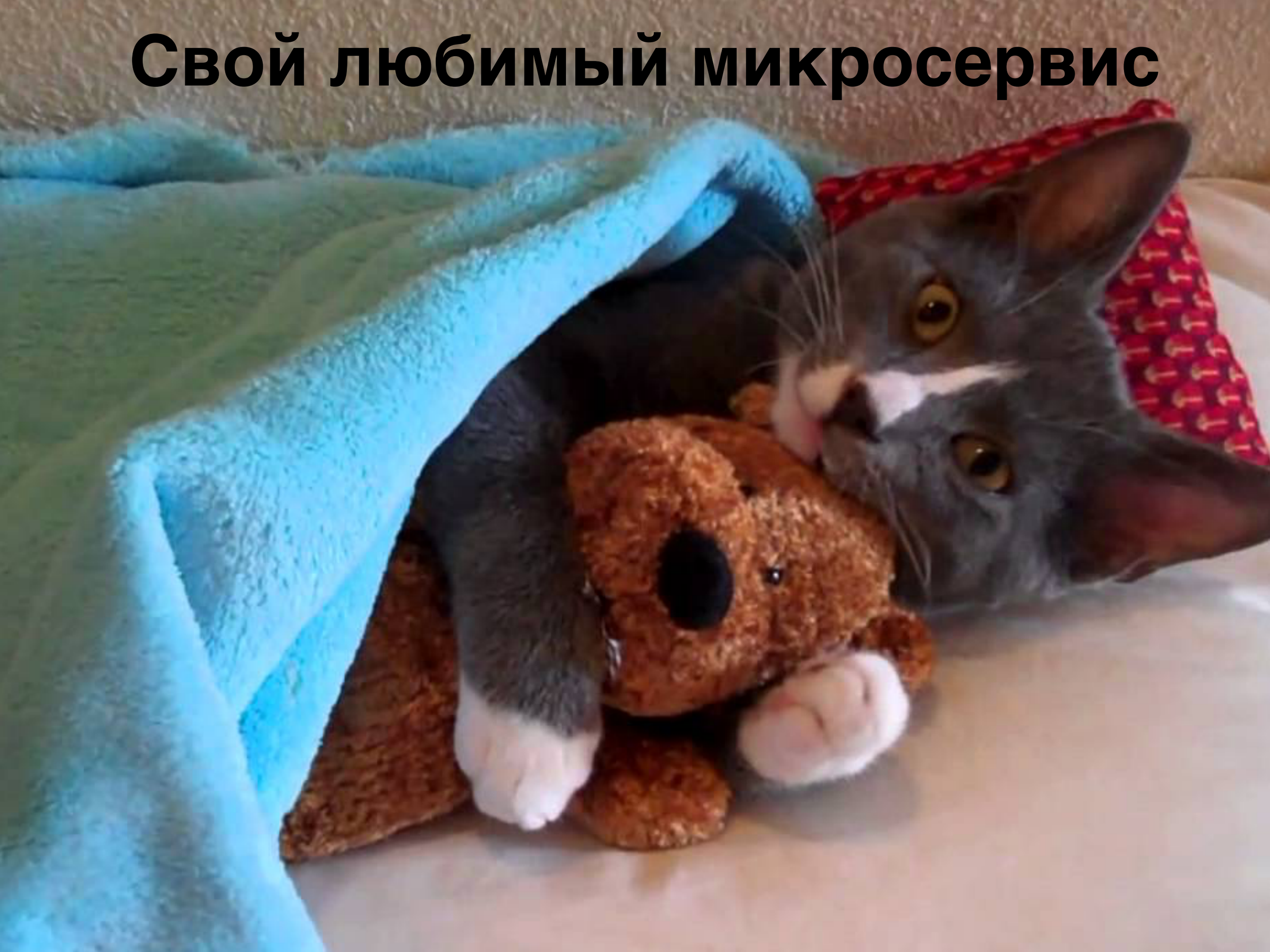
PER ASPERA

AD ASTRA

Отмечено, что

- Быстрее "въезжаешь" в проект
- Сложности оркестрации приходят постепенно
- Знакомишься с новыми технологиями
- Определяешься с любимой специализацией

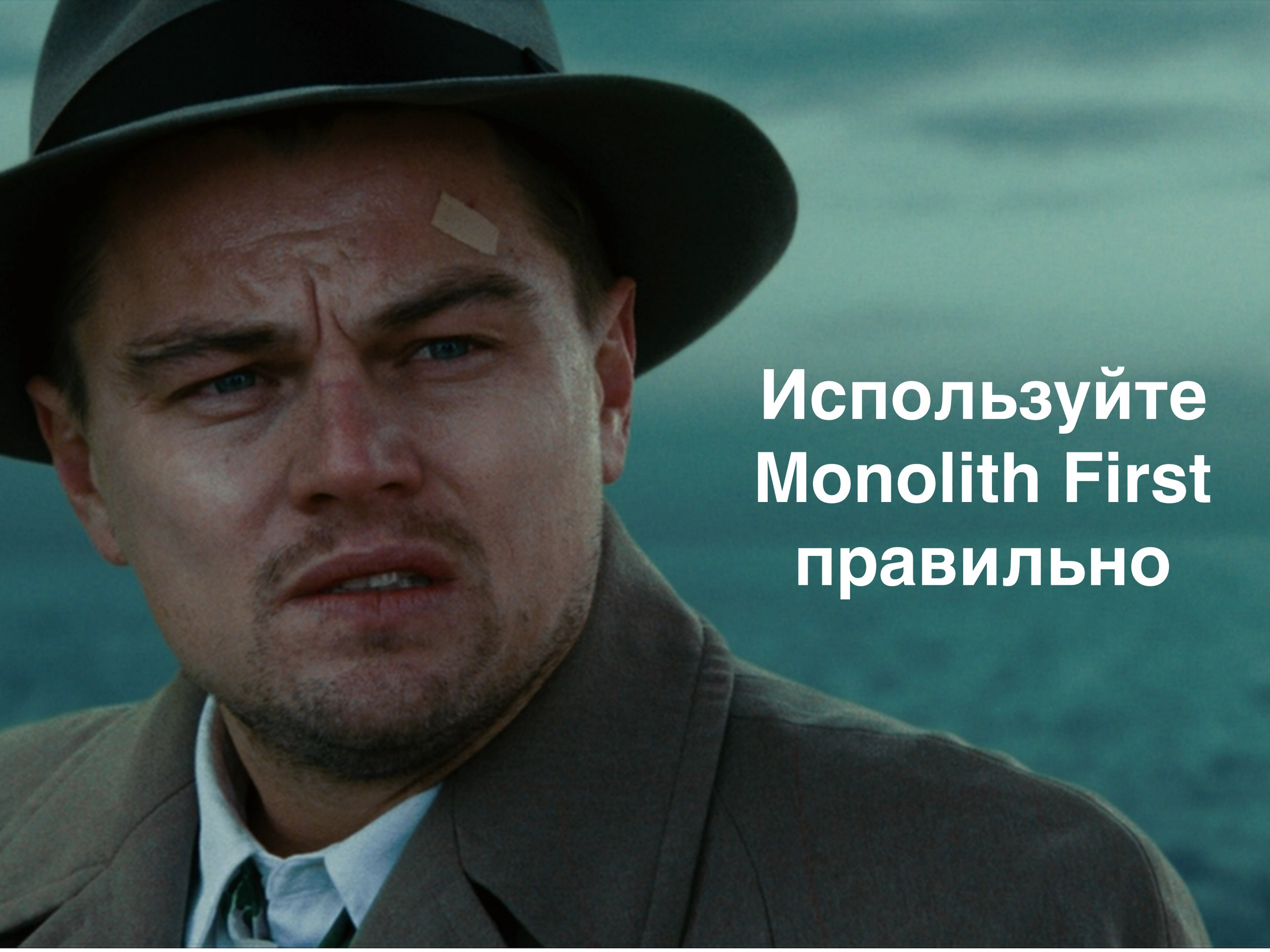
Свой любимый микросервис



Проект с микросервисами - лучший вариант работы для студента

Говорят, вы перешли на микросервисы

И как они вам?



Используйте Monolith First правильно


Следите
за вашим
зоопарком
сервисов

Каждой задаче свой инструмент

Тщательно продумывайте API ваших приложений

Изменяйте API по паттерну Expand and Contract

That's all Folks!