



Урок 4 > Высокоуровневый дизайн

> Оглавление

- > [Оглавление](#)
- > [Высокоуровневый дизайн](#)
 - > [Общий дизайн](#)
- > [Сокращатель ссылок](#)
 - [Первоочередные особенности](#)
- > [Хостинг текстов](#)
 - [Первоочередные особенности](#)
- > [Автодополнение](#)
 - [Первоочередные особенности](#)
- > [Облачный диск](#)
 - [Первоочередные особенности](#)
- > [Фотоаппы](#)
 - [Первоочередные особенности](#)
- > [Телеграм](#)
 - [Первоочередные особенности](#)
- > [Твиттер](#)
 - [Первоочередные особенности](#)
- > [Нетфликс](#)
 - [Первоочередные особенности](#)
- > [Сервис такси](#)
 - [Первоочередные особенности](#)

> Высокоуровневый дизайн

Под высокоуровневым дизайном мы будем понимать такой дизайн, который минимально учитывает основные сценарии взаимодействия с системой, но при

этом не отражает всю сложность взаимодействия между большим количеством реальных компонентов и не принимает во внимание ту нагрузку, под которой система будет в продуктивной среде.

Можно исходить из того, что реализации высокоуровневого дизайна было бы условно достаточно для демонстрации MVP вашего продукта на ноутбуке одного из фаундеров.

В дальнейших лекциях мы будем нанизывать детали для дополнительного функционала и повышать масштабируемость, отзывчивость и устойчивость нашей системы.

> Общий дизайн

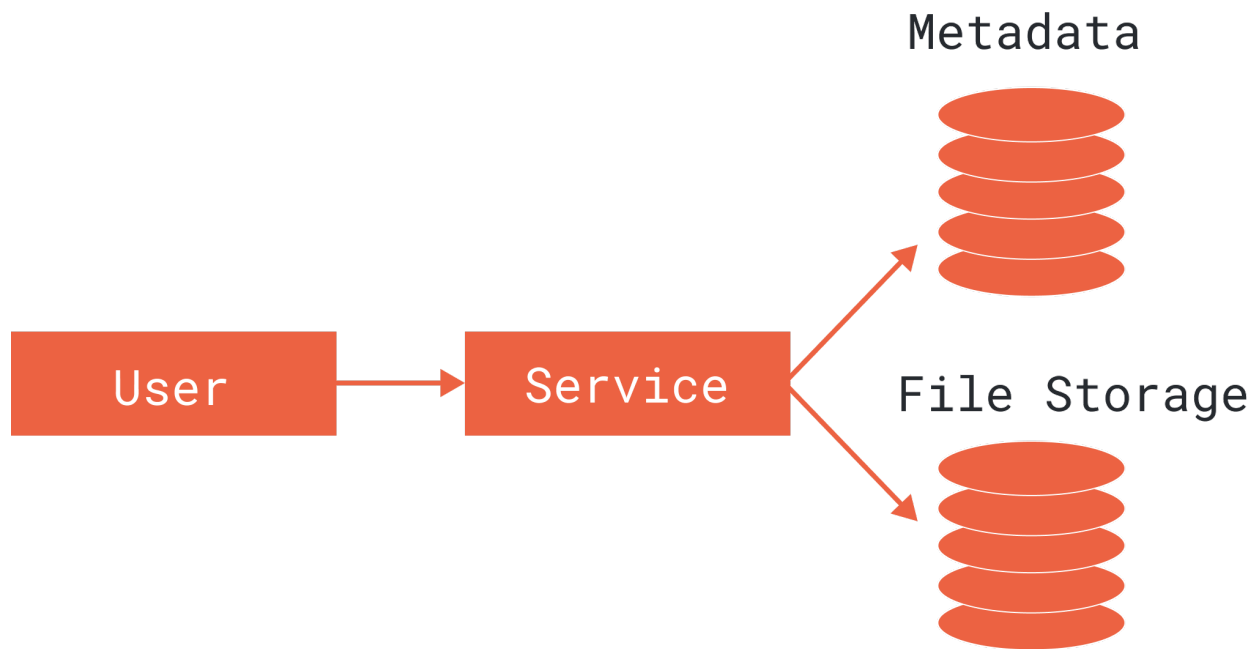
В голове можно всегда держать общий дизайн, подходящий почти для всего:

- Часть системы взаимодействует непосредственно с юзером (какой-то front-end)
- Часть системы работает скрыто от пользователя под капотом (какой-то back-end)
- Часть системы заточена на долгосрочное хранение данных (базы данных, файловые системы и т.п.)



Общий дизайн

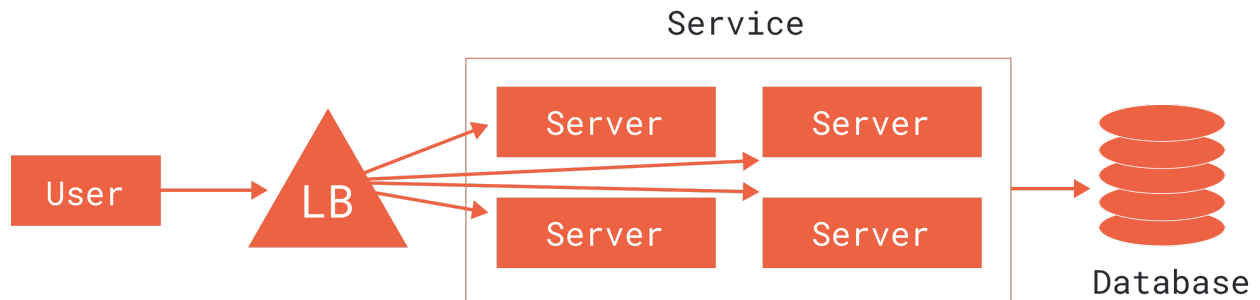
Адаптация под специфику задачи



Для системы онлайн-просмотра фильмов мы можем отметить важность работы с тяжелыми файлами - добавим хранилище



Взаимодействие разных пользователей. Для системы заказа такси можно отметить, что с системой взаимодействуют как пассажиры, так и водители

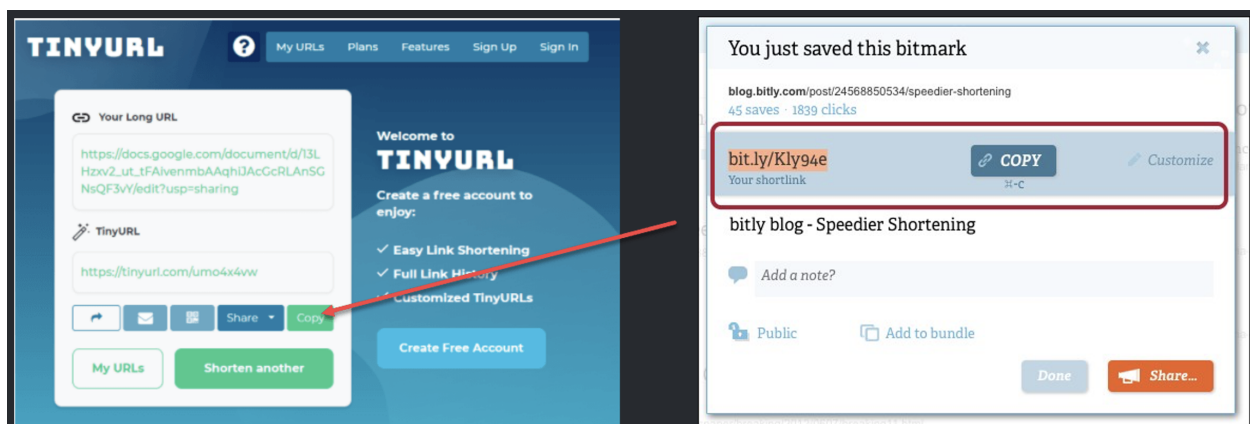


Для системы с миллионами пользователей можно сразу отразить необходимость балансировки нагрузки

> Сокращатель ссылок

Зачем нужен данный сервис?

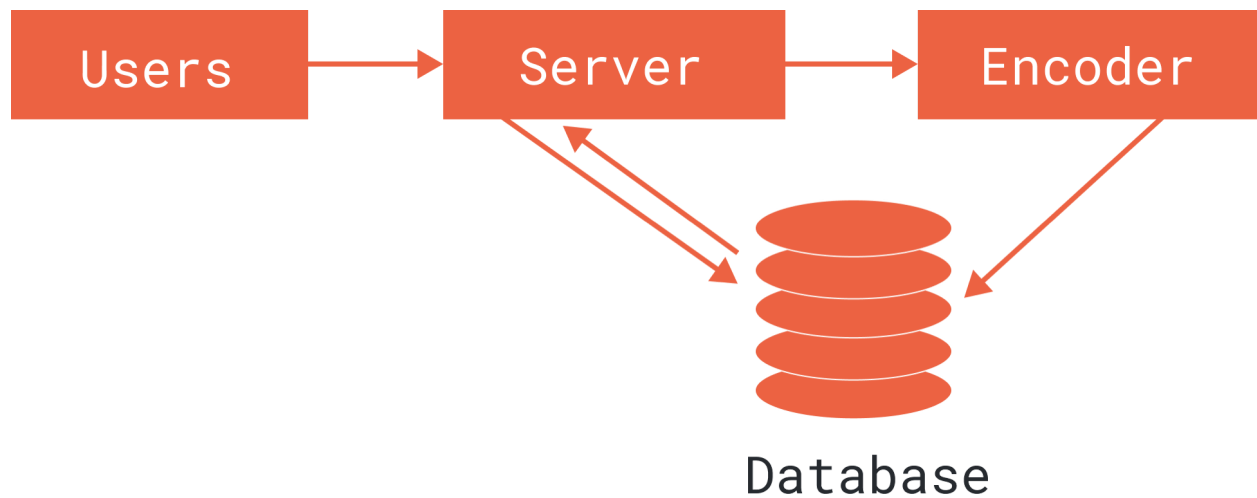
Чтобы превращать длинные ссылки в короткие и делиться ими.



Первоочередные особенности

- Система сама по себе имеет довольно простой функционал – прочитать или создать для ссылки короткий алиас. Можно иметь два модуля: один отвечает на запросы пользователей, а второй генерит ссылки.
- При минимальном масштабировании необходимо, чтобы алиасы не повторялись между серверами. Сервера должны синхронизироваться между собой.

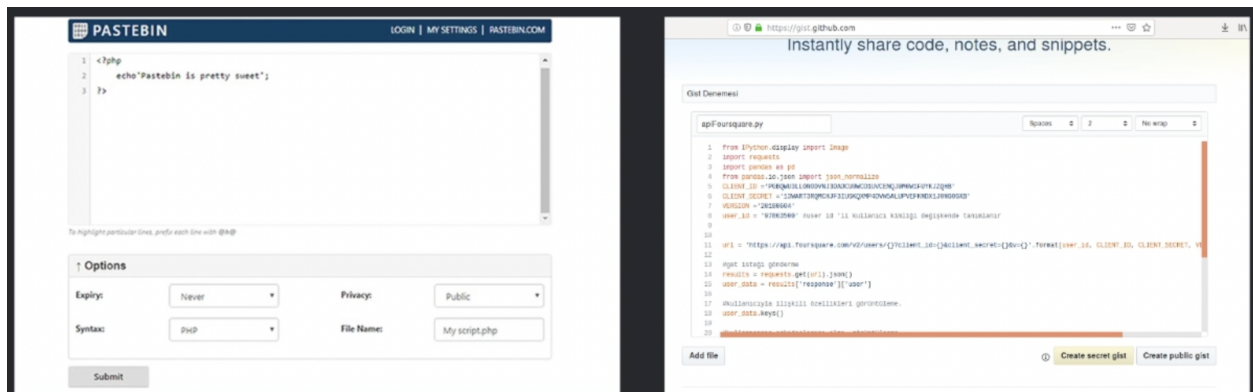
- Можно отразить, что нам стоит озаботиться отдельным кодирующим сервисом для их создания.



> Хостинг текстов

Зачем нужен такой сервис?

Чтобы делиться кусками кода и искать в нем баги вместе.

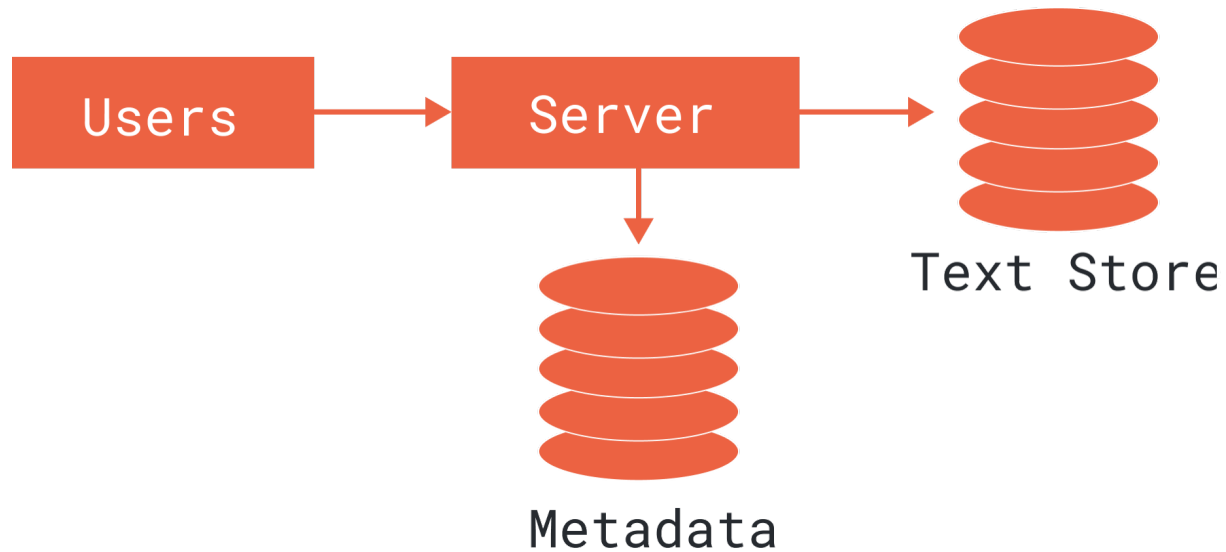


Первоочередные особенности

- По сравнению с простым сокращателем ссылок можем хранить текст любого размера

- Стоит сразу подумать о том, что хранить большие записи и малые метаданные лучше отдельно
- Можно отразить, что будем использовать базу данных для метаданных и отдельное хранилище для текстов

Сервер обрабатывает все запросы и общается с двумя базами: одна с метаданными, а другая - хранилище больших текстов



> Автодополнение

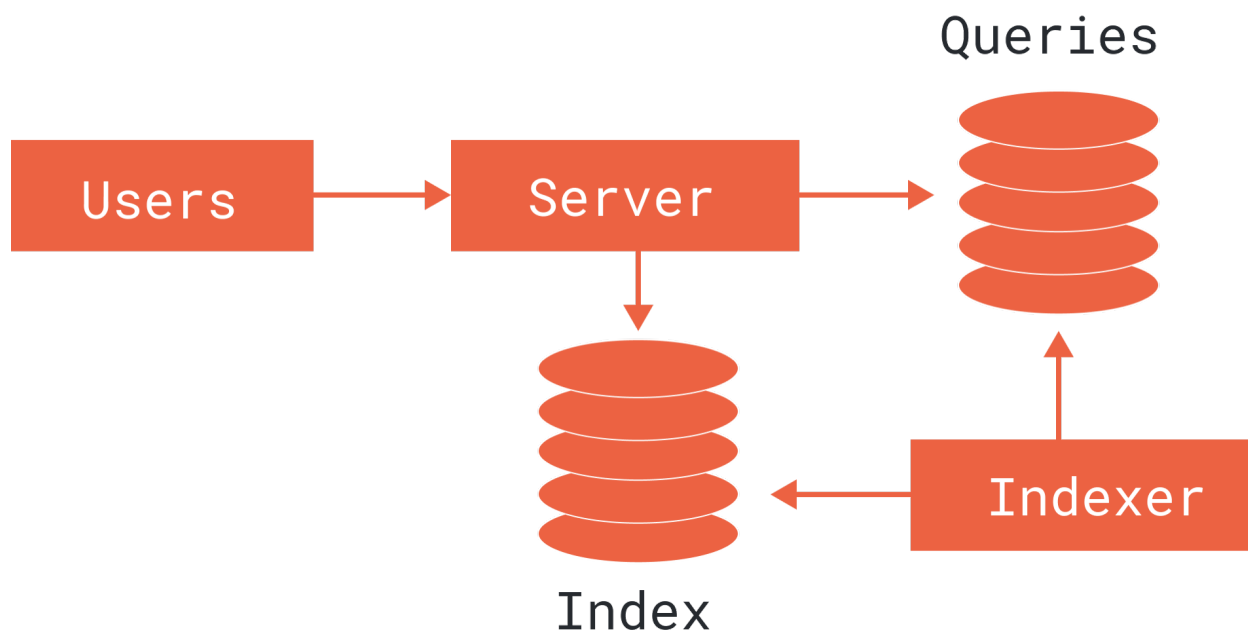
Зачем нужен такой сервис?

Чтобы при вводе первых букв запроса предлагались варианты дополнения. Например, при поиске «Н» сразу дополнялось до «НФТ купить».



Первоочередные особенности

- Сервис под капотом хранит все существующие запросы из другой системы (поиска)
- Мы должны на его основе заранее построить специальный индекс для поиска слов с префиксом
- Можно отразить, что при обращении к системе мы идем в поисковый индекс, построенный на основе запросов

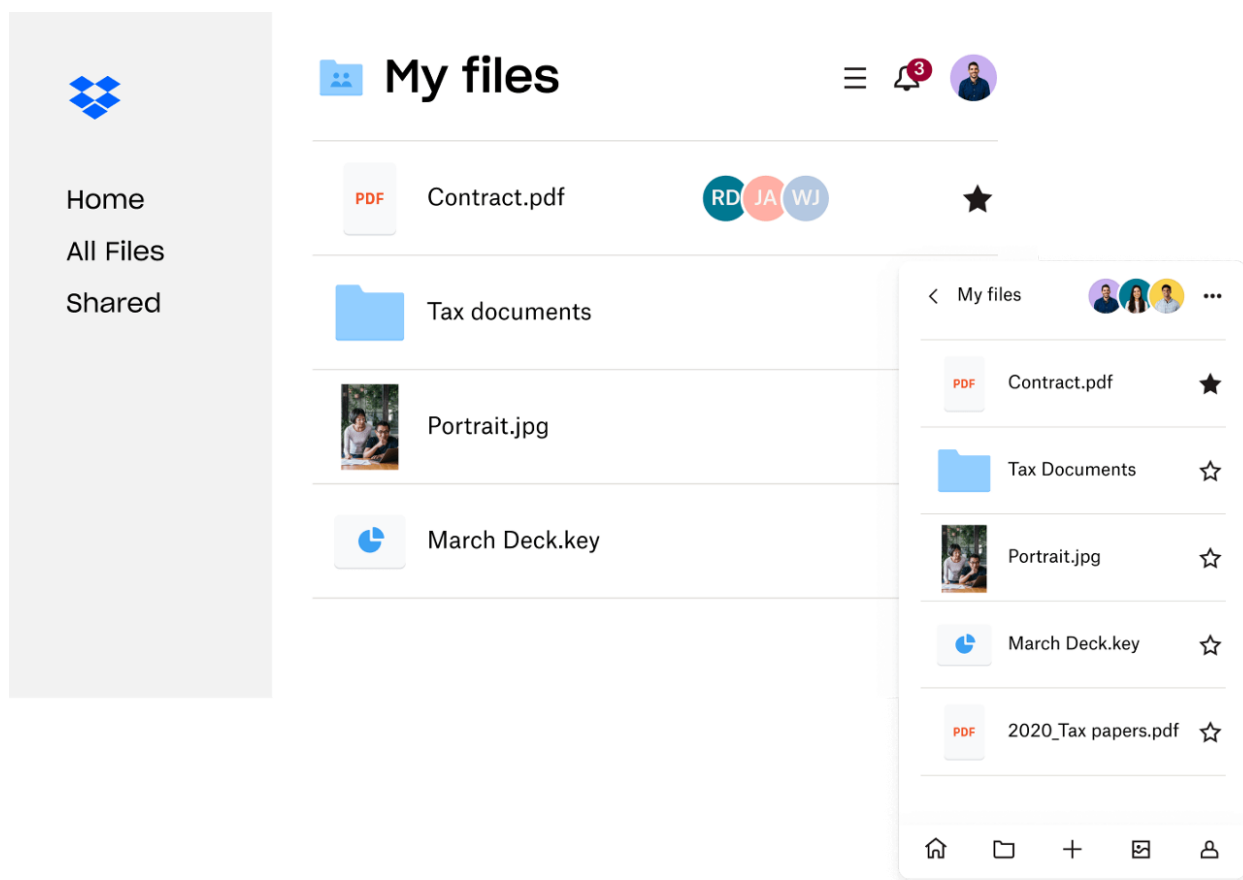


Сервер логирует запросы и обращается к построенному на основе предыдущих запросов индексу

> Облачный диск

Зачем нужен такой сервис?

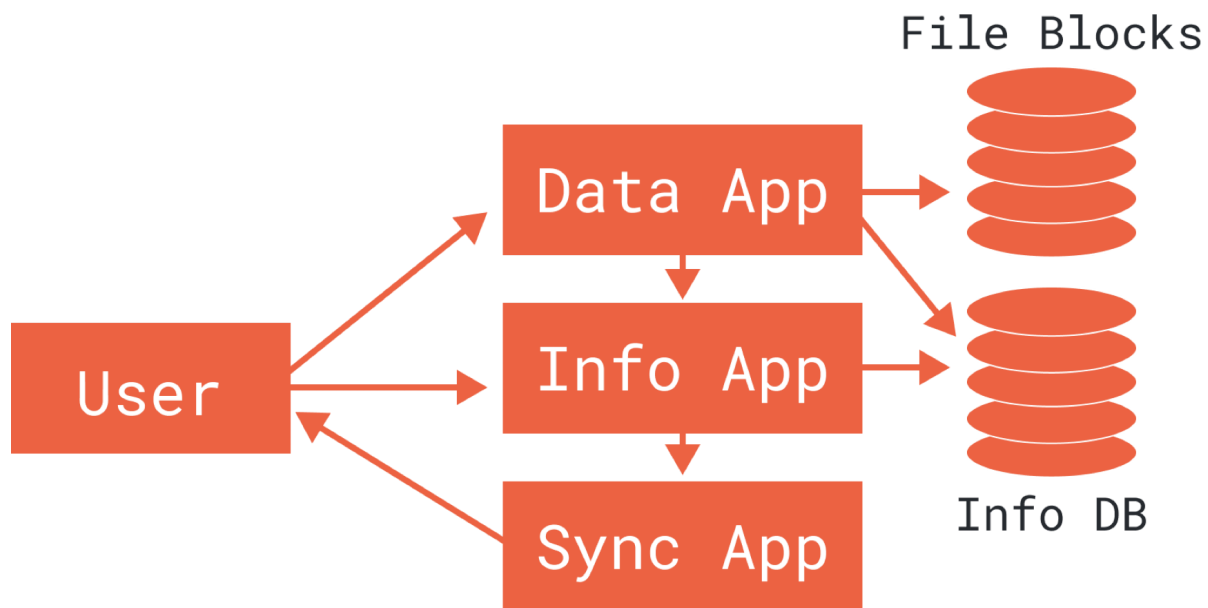
Чтобы файлы были везде доступны.



Первоочередные особенности

- Файлы могут быть любого размера, стоит использовать отдельно БД для метаданных и файловое хранилище

- Клиентское приложение должно уметь показывать структуру файлов, и при необходимости загружать/скачивать
- При обновлении файлов другим клиентам приходят уведомления об этом
- Можно отразить использование 2-х типов хранилища и компоненты загрузки, обновления и синхронизации

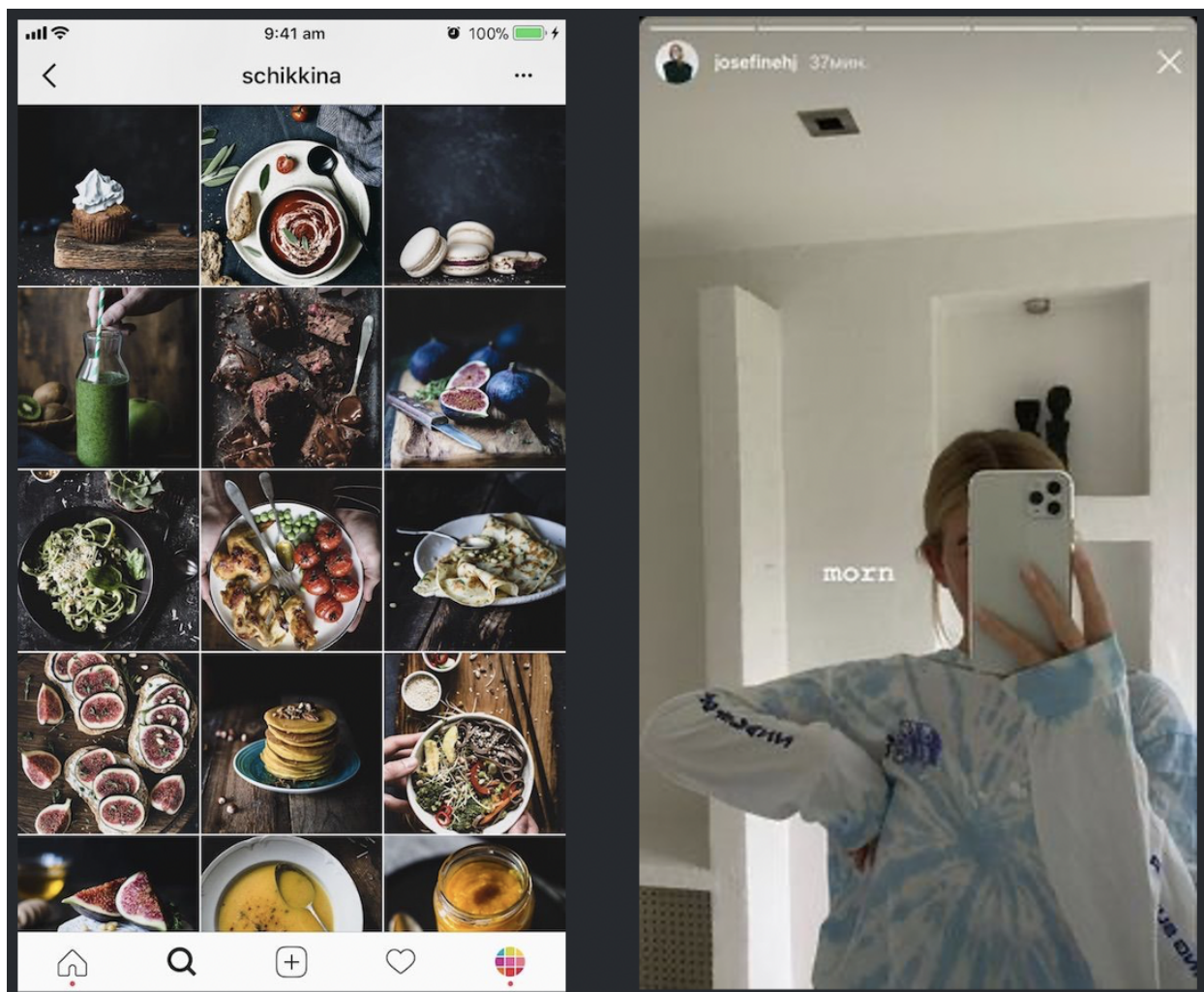


Пользователю могут показываться данные как факт (info), данные могут загружаться (data) и синхронизироваться (sync)

> Фотоаппы

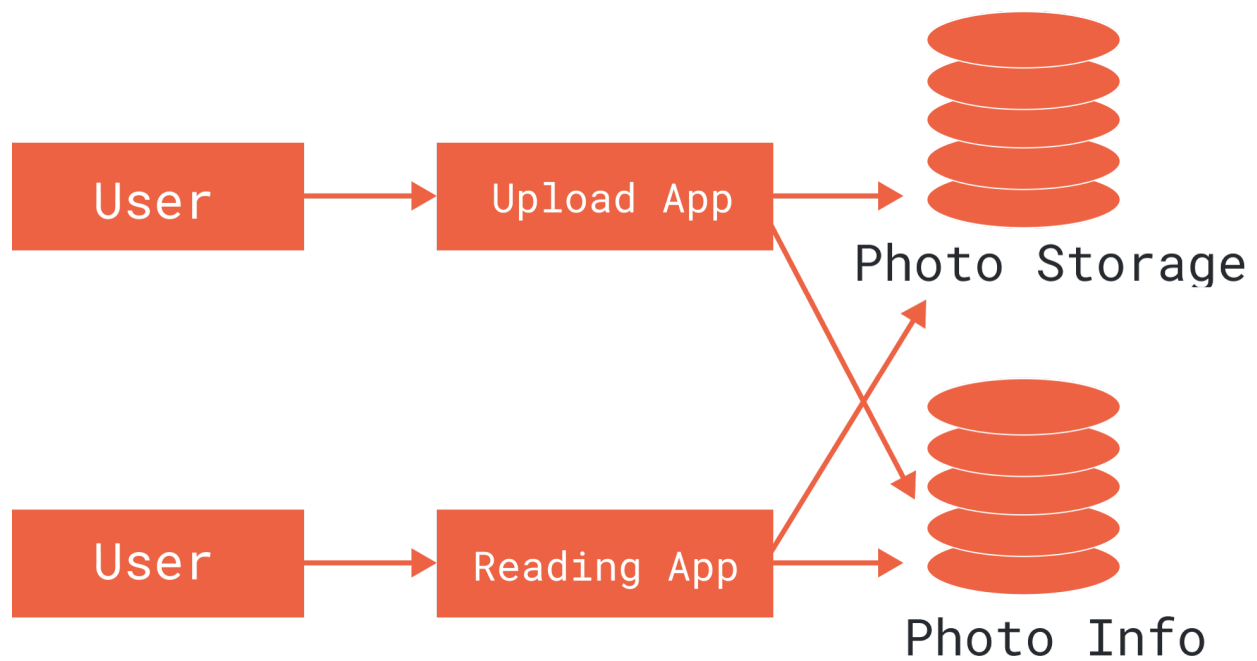
Зачем нужен такой сервис?

Чтобы делиться фотографиями в зеркале и с едой с друзьями.



Первоочередные особенности

- Основных сценариев использования приложения два – загрузка или просмотра фотографий
- Фотографии занимают несравнимо больше, чем метаданные о снимках, постах и самих пользователей
- Можно отразить оба сценария в виде микро-приложений и необходимость разделения хранилища



Два разных сценария взаимодействия

> Телеграм

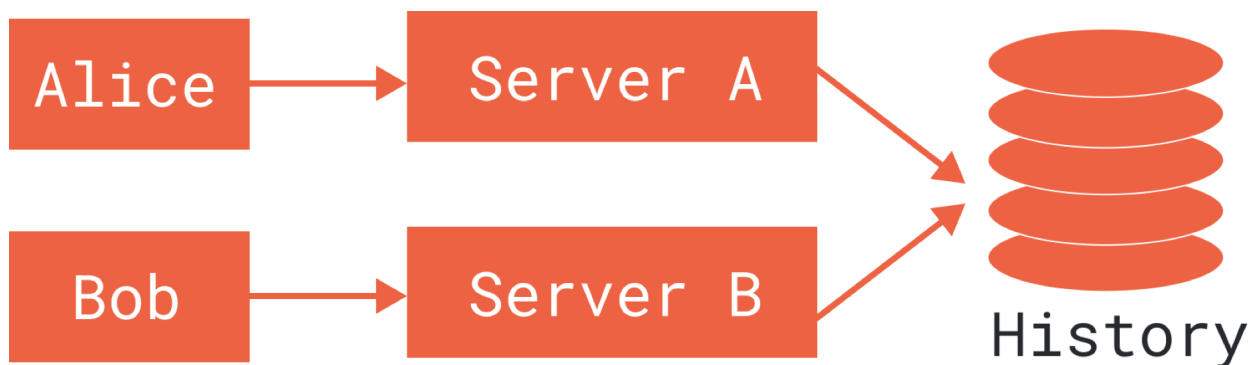
Зачем нужен такой сервис?

Чтобы от общения вас отвлекали с работы, где не купили Slack.



Первоочередные особенности

- Пользователь может обращаться либо к истории сообщений, либо вести беседы
- При беседе двух пользователей необходимо быстро доставлять их сообщения друг другу
- Можно отразить необходимость взаимодействия серверов, к которым обращаются пользователи



> Твиттер

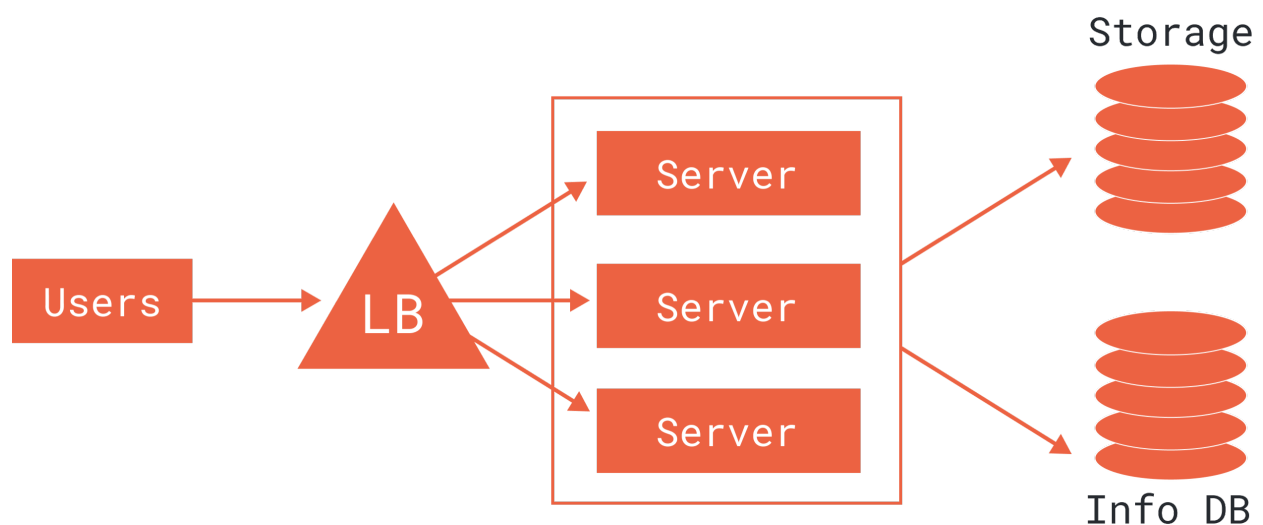
Зачем нужен такой сервис?

Чтобы делиться короткими сообщениями со всеми.



Первоочередные особенности

- Огромное количество пользователей, а значит, необходимость масштабирования
- Посты могут содержать тяжелые фото или видео, которые лучше хранить отдельно от данных про пользователей
- Можно сразу отметить наличие балансировщика нагрузки, а также наличие разных типов хранилищ

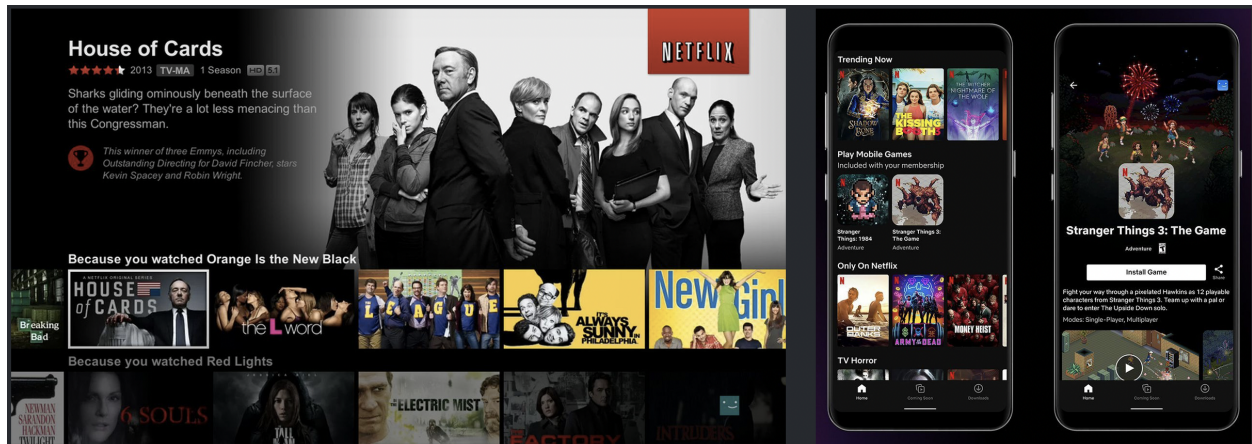


Balancer получает запрос и перераспределяет его на нужный сервер

> Нетфликс

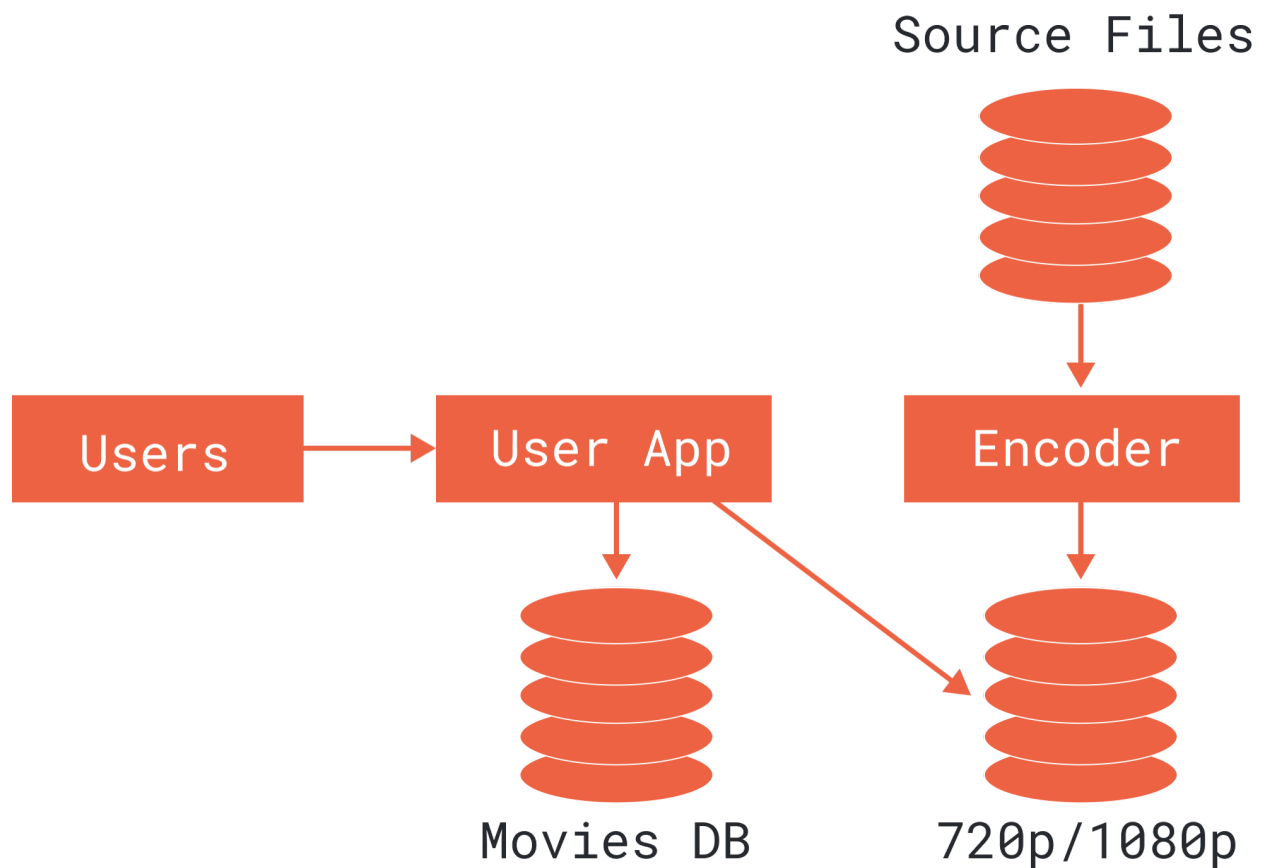
Зачем нужен такой сервис?

Чтобы смотреть любимые фильмы, сериалы и передачи.



Первоочередные особенности

- Пользователи могут просматривать как информация о фильмах, так и несравнимо более тяжелые фильмы как таковые
- Пользователи могут смотреть фильмы с разных устройств, значит нужно подготовить версии фильмов разного размера
- Можно отметить использование нескольких типов хранилищ и необходимость подготовки фильмов под разные форматы

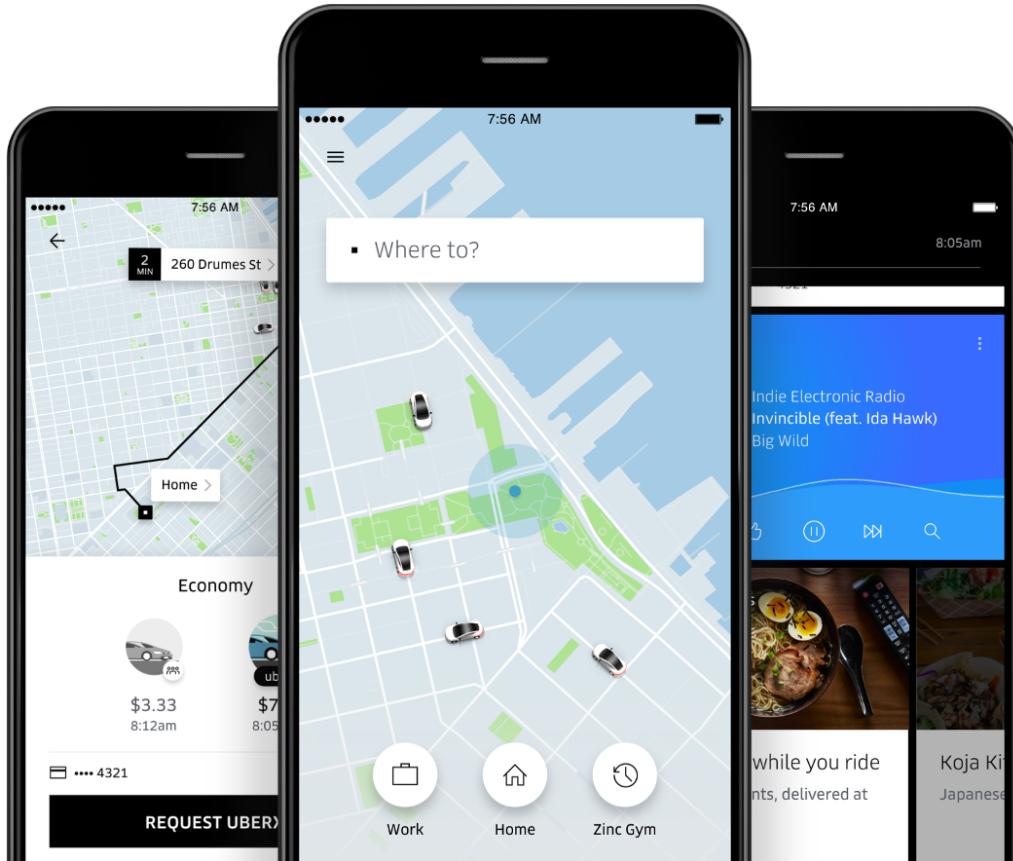


Пользовательское приложение общается с БД информации о фильмах и БД файлов, которые заранее приведены к нужному разрешению

> Сервис такси

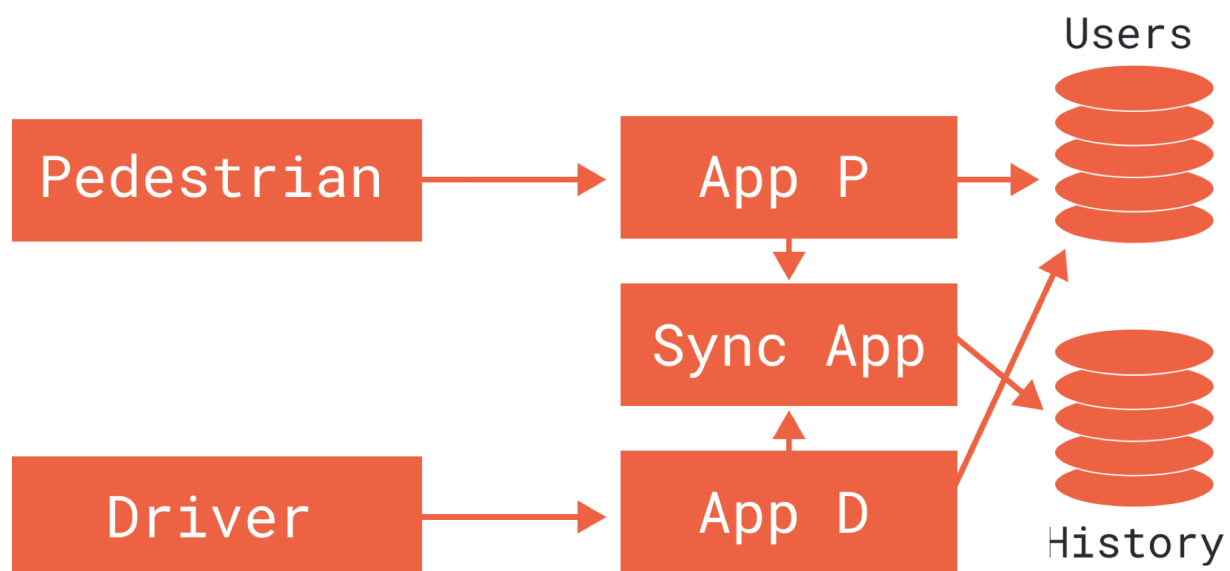
Зачем нужен такой сервис?

Чтобы легко доехать из точки А в точку Б с водителем.



Первоочередные особенности

- Есть фактически отдельные сервисы для водителя и пассажира как пользователей сервиса
- Нужно как обрабатывать текущую поездку, так и хранить исторические вместе с информацией о пользователях
- Можно отразить раздельность сервисов, необходимость ведения поездок и хранения истории



Два разных приложения взаимодействуют с базой данных пользователей, после завершения взаимодействия данные попадают в историю