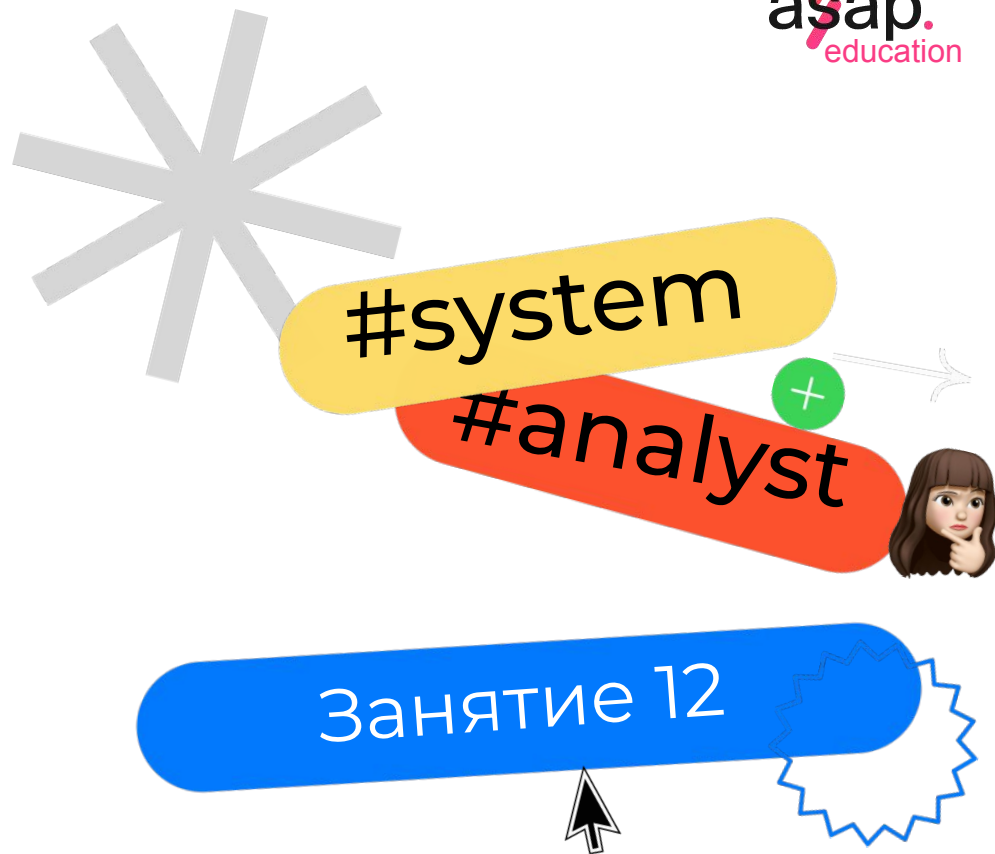


Основы web-приложений

Люба Вайгель

Ментор ASAP Education



**Какие UML диаграммы
вы помните?**

Какие виды UML диаграмм
чаще всего составляет
системный аналитик?

План занятия

Веб-приложения

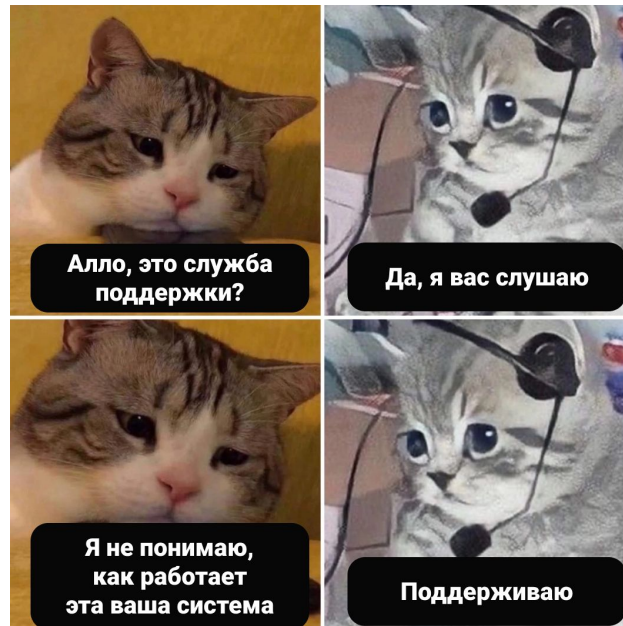
Из чего состоят и как работают

Клиент-серверная архитектура

Что такое архитектура, из чего состоит и как проектируется

Микросервисы

Что это такое, когда и для чего используются



Веб-приложения



Веб-приложение

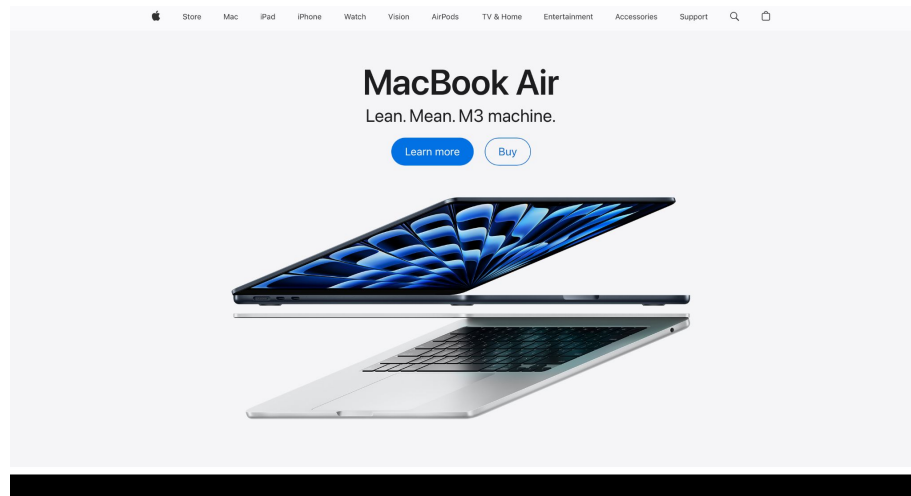
- Веб-приложение — это программа, которую можно открыть через интернет с помощью браузера
- Оно дает пользователям возможность с ним взаимодействовать и может использоваться для разнообразных целей, таких как ведение бизнеса, общение, обучение и др.

Примеры

- YouTube
- Airbnb
- Яндекс
Карты
- Озон

Веб-приложение vs. Сайт

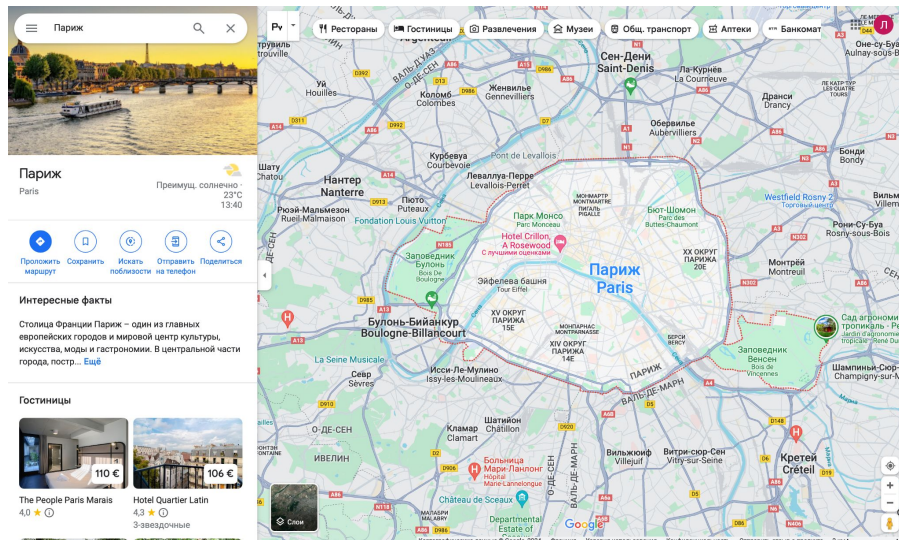
- Обычные сайты - это набор статических страниц с информацией
- Веб-приложение отличается от обычного сайта тем, что позволяет пользователям взаимодействовать с ним, вводя данные и получая результаты



apple.com - сайт

Веб-приложение vs. Сайт

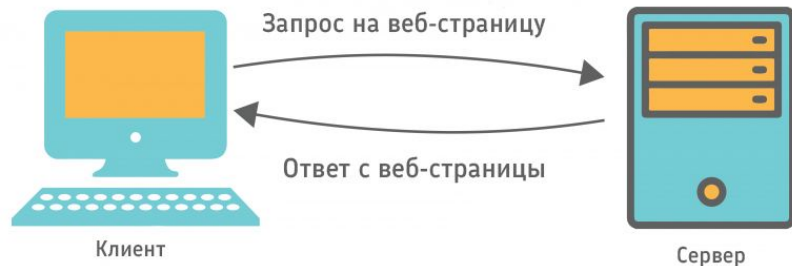
- Обычные сайты - это набор статических страниц с информацией
- Веб-приложение отличается от обычного сайта тем, что позволяет пользователям взаимодействовать с ним, вводя данные и получая результаты



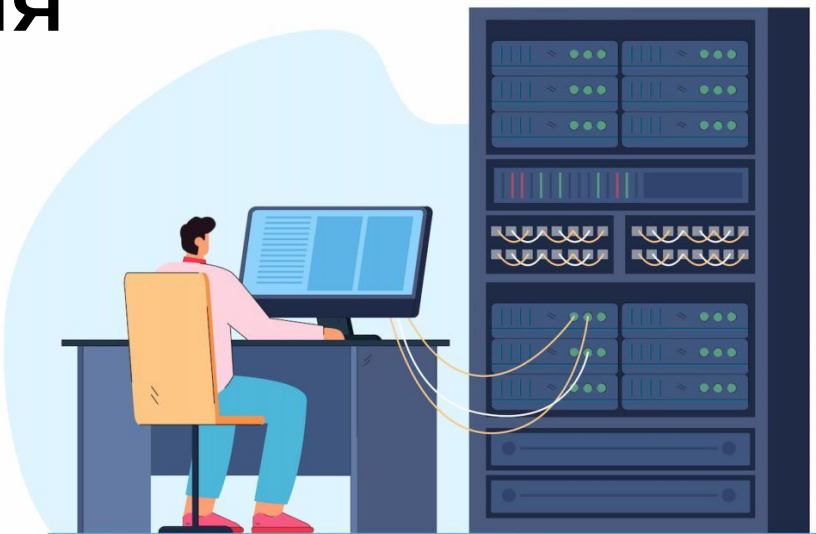
Google Maps - веб-приложение

Веб-приложение

- Любое сложное приложение — это не только картинка в браузере, это ещё и данные, которые пользователи используют или создают. Эти данные нужно уметь хранить, обрабатывать и выводить.
- Хранением и обработкой данных обычно занимается **сервер** (бэкенд)
- А архитектура, в которой участвуют сервер (бэкенд) и клиент (браузер, фронтенд) называется **клиент-серверной**



Клиент-серверная архитектура



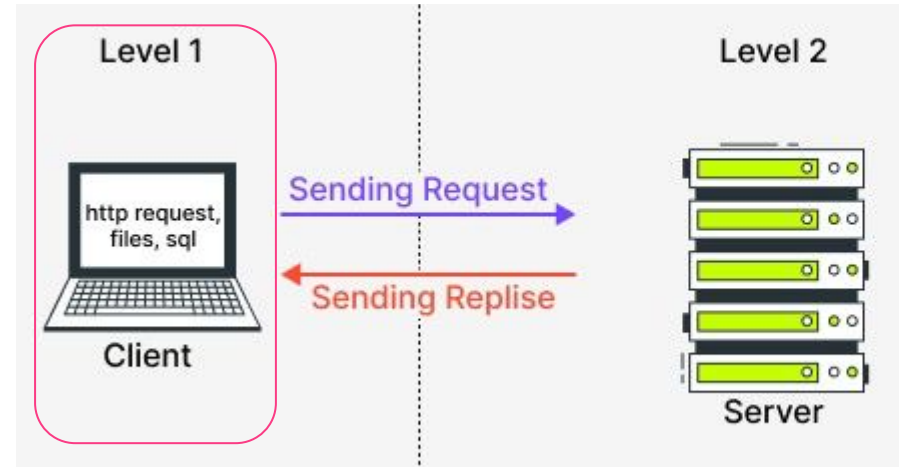
Клиент-серверная архитектура

- Архитектура — это описание системы на самом высоком уровне
- **Клиент-серверная архитектура** описывает, как взаимодействуют между собой клиент (в нашем случае фронтенд) и сервер (бэкенд)



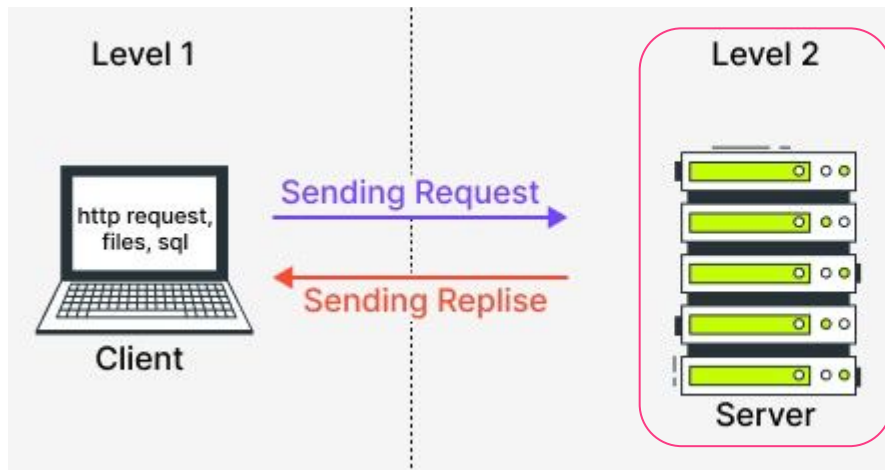
Клиент

- Клиент обращается к серверу с **запросами**
- Роль клиента для сервера в том, чтобы сообщить серверу, что нужно сделать с данными, которые хранятся в базе, или с данными, которые он передаёт
- Роль клиента для пользователя в том, чтобы представить данные в удобном виде и предоставить механизмы для их обновления
- Для веба клиент почти всегда браузер



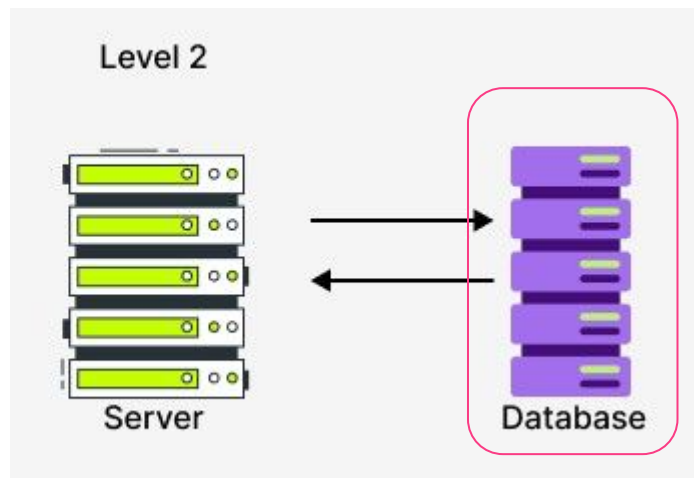
Сервер

- Сервер принимает запросы от клиента
- Его роль в том, чтобы сохранять информацию от клиента в базе данных, обрабатывать её и предоставлять к ней доступ по некоторым правилам - **бизнес-логике**
- На сервере помимо общения с клиентом могут запускаться какие-то фоновые задачи

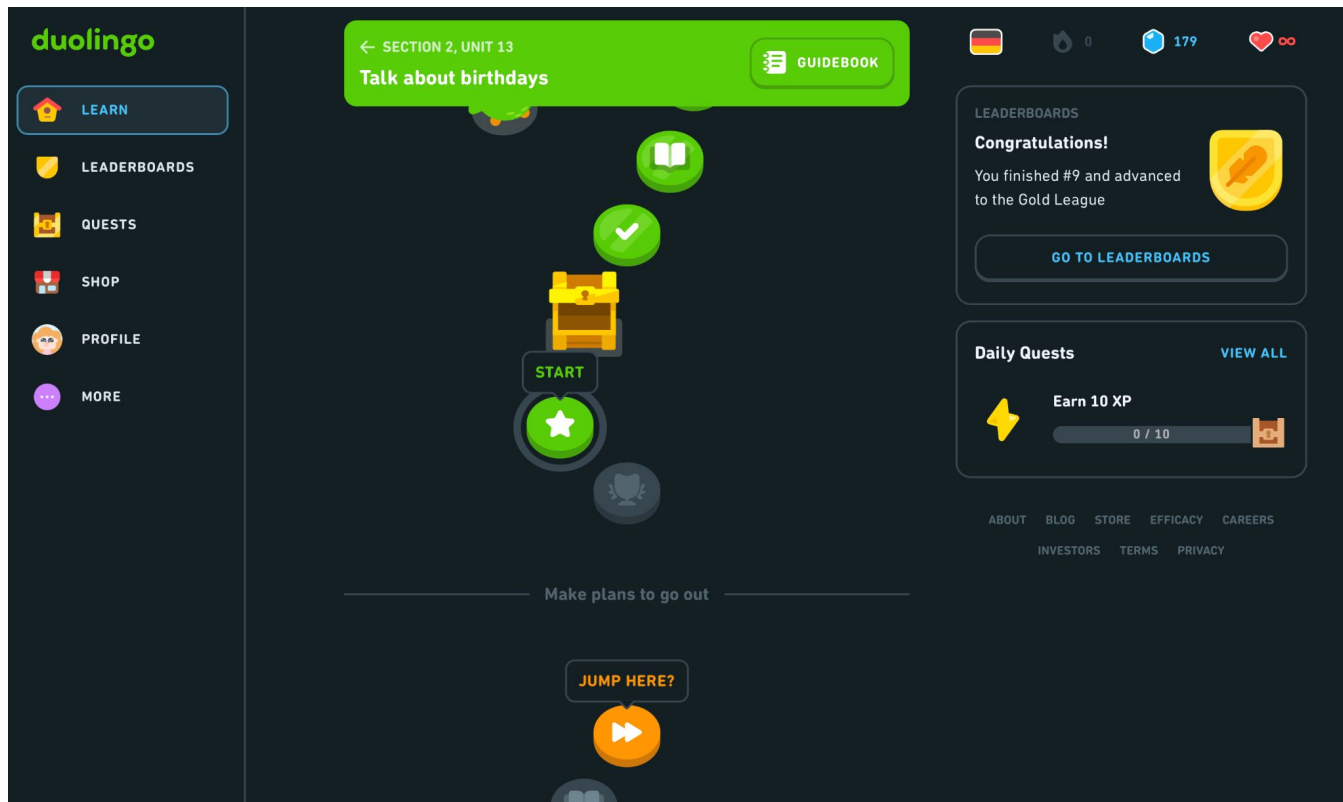


База данных

- База данных (БД) — это хранилище всей пользовательской и служебной информации
- Её роль в том, чтобы обеспечивать быстрый и бесперебойный доступ к этой информации и собственно хранение



Сайт или веб-приложение?



Сайт или веб-приложение?



Википедия
Свободная энциклопедия

Заглавная страница
Содержание
Избранные статьи
Случайная статья
Текущие события
Пожертвовать

Участие
Сообщить об ошибке
Как править статьи
Сообщество
Форум
Свежие правки
Новые страницы
Справка

Инструменты
Ссылки сюда
Связанные правки
Служебные страницы
Постоянная ссылка
Сведения о странице
Получить короткий URL
Скачать QR-код
Развернуть всё

Печать/экспорт
Скачать как PDF
Версия для печати

Заглавная Обсуждение

Вы не представились системе Обсуждение Вклад Создать учётную запись Войти

Читать Просмотр кода История

Добро пожаловать в **Википедию**,
свободную энциклопедию, которую может редактировать каждый.

Сейчас в Википедии **1 996 868 статей** на русском языке.

[Порталы](#) [Справка](#) [Создать статью](#)

Избранная статья

Альфред Модсли

Альфред Персиваль Модсли (англ. *Alfred Percival Maudslay*, 18 марта 1850, Лондон, Англия — 22 января 1931, Херефорд) — британский путешественник-исследователь, фотограф и археолог-любитель. Один из пионеров научной майялистики.

Родился в семье инженера; его двоюродный брат Генри Модсли-младший проводил археологические раскопки в Иерусалиме. После окончания университета Альфред Модсли в 1872—1880 годах состоял на колониальной службе в Вест-Индии, Австралии и Океании. В период 1881—1894 годов провёл семь экспедиций в регионы Мексики и Гватемалы, населённые народами майя. Вёл раскопки на шести археологических объектах: [Копан](#), [Киригуа](#), [Яшчилан](#), [Чичен-Ица](#), [Тикале](#) и [Паленке](#). Занимался обмерами и фотографированием памятников, а также делал целлюлозные и гипсовые слепки наиболее примечательных скульптур, ныне хранящиеся в [Британском музее](#) (около 400 единиц). В составе многотомной энциклопедии «*Biologia Centrali-Americana*» Модсли выпустил пять томов, посвящённых археологии (1889—1902). В конце жизни занимался переводом на английский язык важного исторического



Из новых материалов

Знаете ли вы?

- **Эпоха** подчинения Англии французским герцогам перетекла во **времена войны** (*на илл.*) английских королей за власть над Францией.
- Накануне **Первой мировой войны** потомки **60 миллионов европейских мигрантов** составляли 38 % населения мира.
- В **атомных шахматах** **короли** не могут **есть** и ищут спасения друг у друга, а пешки — самые взрывоустойчивые фигуры.
- **Убийца Лермонтова завещал** не ставить на своей могиле надгробия, чтобы память о нём исчезла.
- **Принцип**, введённый генералом **иезуитов** (*на илл.*), применяли и **нацисты**, и **коммунисты**.
- **Американский психолог** с помощью чернильных пятен установила, что большинство подсудимых по **Нюрнбергскому процессу** не имели психических отклонений.
- **«Азиатский тиран»** был более чем в два раза меньше **своих** родственников.
- В самом начале XX века **модерн** был известен как **англосаксонский стиль** и имел **поддержку** со стороны **русской императрицы**.



Сайт или веб-приложение?

Хабр | КАК СТАТЬ АВТОРОМ


Моя лента | Все потоки | Разработка | Администрирование | Дизайн | Менеджмент | Маркетинг | Научпоп

Войти


Все потоки

СТАТЬИ +78 | ПОСТЫ +6 | НОВОСТИ +39 | ХАБЫ | АВТОРЫ | КОМПАНИИ

Все подряд


 youngmup 18 минут назад

TDD: да или нет?








 Простой | 4 мин | 102

Программирование*, TDD*, Промышленное программирование*

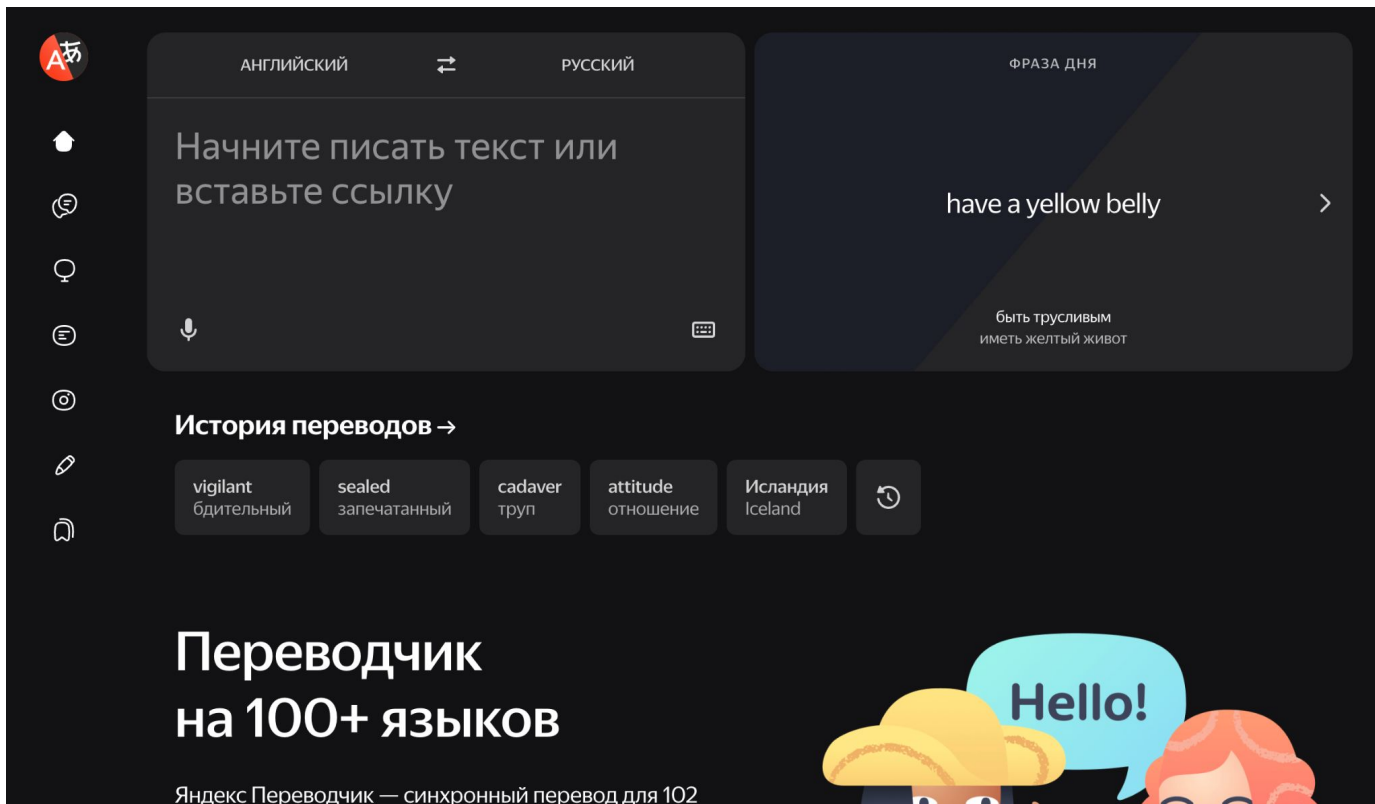
Мнение



ЛУЧШИЕ БЛОГИ

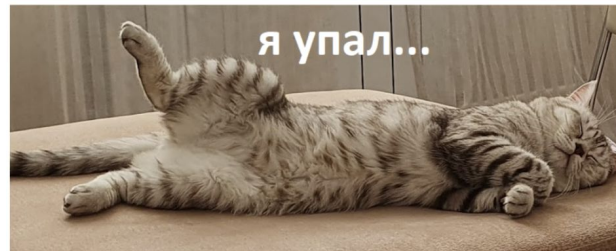
	RUVDS.com	2934.06
	Selectel	1637.51
	Timeweb Cloud	1444.33
	MTC	1261.67
	Сбер	920.06
	Яндекс	726.5
	OTUS	703.69

Сайт или веб-приложение?



Микросервисы

МОНОЛИТ



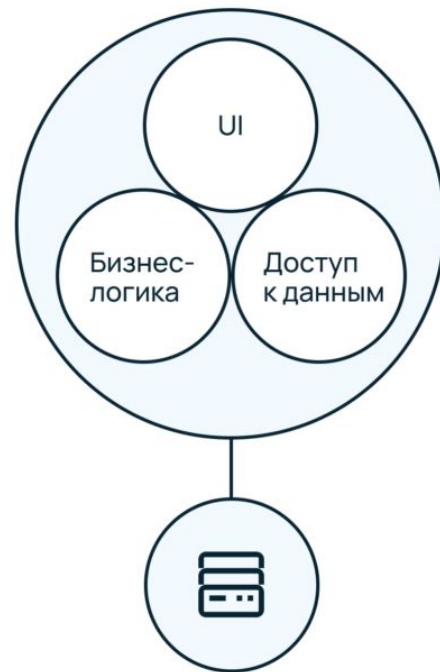
микросервисы



Монолитная архитектура

В классическом понимании клиент-серверная архитектура - монолитная

Монолитная архитектура означает, что приложение — большой связанный модуль, все компоненты которого спроектированы для совместной работы, используя общую память и ресурсы



Монолитная архитектура

ПРЕИМУЩЕСТВА

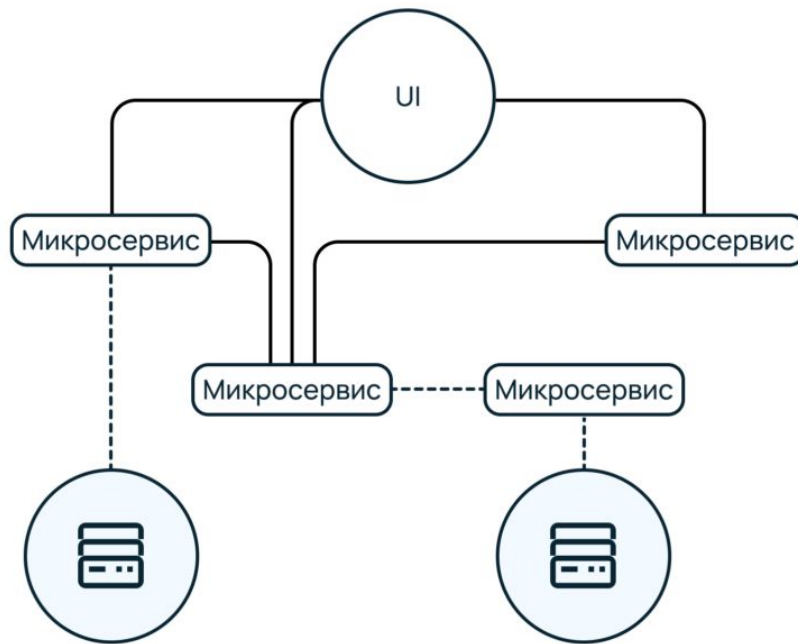
- Простота разработки и тестирования
- Небольшие накладные расходы
- Производительность

НЕДОСТАТКИ

- Масштабируемость
- Сложность поддержки и обновлений
- Сложность отладки и тестирования
- Высокий уровень зависимостей
- Ограниченный технологический стек

Микросервисная архитектура

Микросервис — это отдельное приложение, как правило, очень небольшое, которое поддерживает ограниченную функциональность



Перерыв

Никто:

Я на работе каждые полчаса:

перерывчик на кофе



Микросервисная архитектура

ПРЕИМУЩЕСТВА

- Высокая скорость внедрения нового функционала
- Отсутствие ограничений в стеке технологий
- Высокая масштабируемость
- Высокая производительность приложения

НЕДОСТАТКИ

- Высокая стоимость разработки
- Сложность разработки и обслуживания
- Увеличение нагрузки на инфраструктуру
- Угроза потери данных
- Необходимость иметь опытных разработчиков

Когда использовать МСА?

- **Масштабируемость.** Когда проект требует горизонтального масштабирования, чтобы обеспечить обработку больших объемов данных или запросов
- **Гибкость.** Когда проект нуждается в гибкой архитектуре, чтобы разрабатывать и внедрять новые функции или изменять существующие без значительного влияния на другие компоненты
- **Резилентность.** Когда проект требует высокой доступности и отказоустойчивости, чтобы устойчиво обрабатывать сбои и сбои в одном или нескольких компонентах

Когда использовать МСА?

- **Постоянное развитие.** Когда проект развивается со временем и требует способности к непрерывному развертыванию и масштабированию новых сервисов
- **Разделение ответственности.** Когда проект имеет сложную бизнес-логику, которой легче управлять и понимать, разделяя ее на небольшие автономные сервисы
- **Ускорение разработки.** Когда проект требует повышенной скорости разработки и поставки функций, чтобы эффективно конкурировать на рынке

Когда использовать МСА?

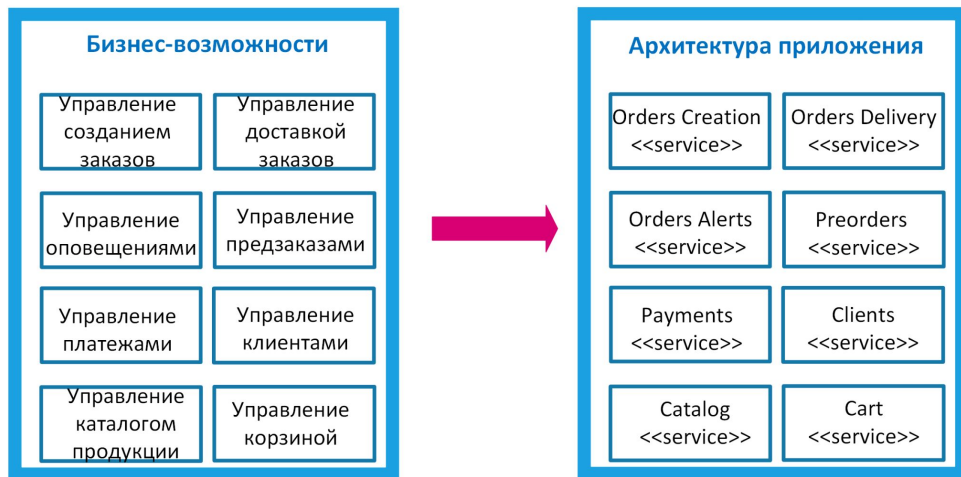
Микросервисная архитектура - хороший выбор для проектов со сложной функциональностью, требованиями к масштабируемости и гибкости, которым важна высокая скорость разработки и высокая доступность

Примеры использования МСА

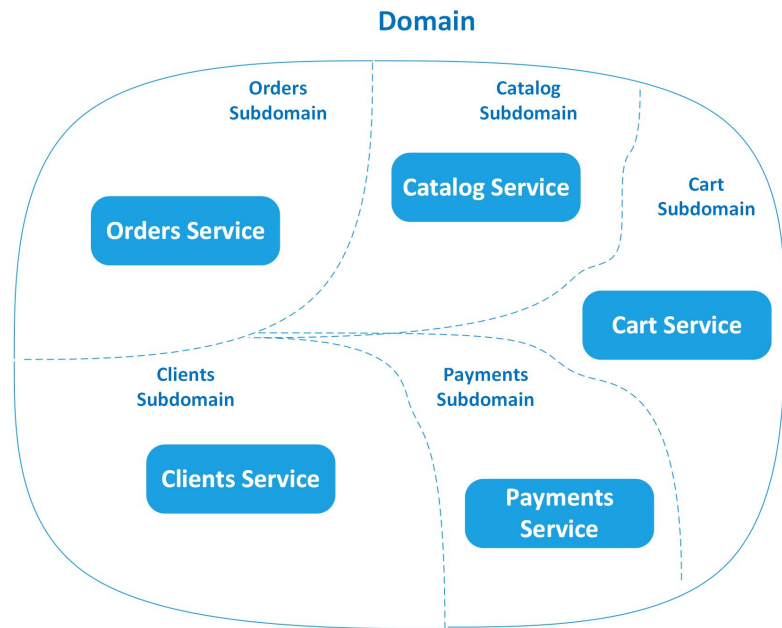
- Веб-приложения
- Мобильные приложения
- Облачные вычисления
- IoT (интернет вещей)
- Большие корпоративные системы

Разделение на микросервисы

1 Разделение по бизнес возможностям



2 Разделение на под-домены



Общение между сервисами

СИНХРОННОЕ

Отправляем запрос и ждем ответ, после чего можем отправить другой запрос

API (REST, SOAP, gRPC)

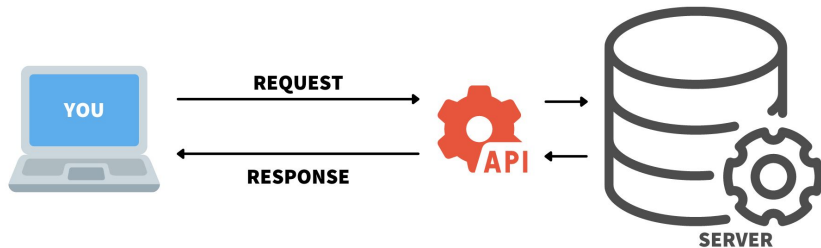
АСИНХРОННОЕ

Отправляем сообщение, а ответ придет когда-нибудь потом или он в принципе не предусмотрен

Очереди (RabbitMQ, Kafka)

API

- Это набор правил, по которым сервисы взаимодействуют друг с другом
- С помощью API программа запрашивает у другого приложения данные, либо просит выполнить некую операцию



Типы API

- **REST:** основан на HTTP-протоколе и использует запросы HTTP для обмена данными
- **SOAP:** основан на XML и использует SOAP-протокол для обмена данными
- **gRPC:** вызывает функции с удаленного сервера, основан на HTTP/2

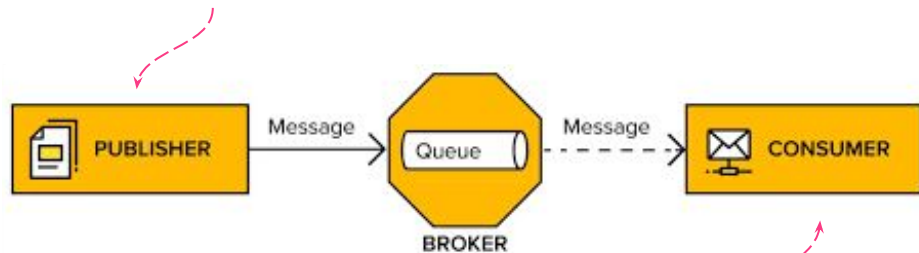
Примеры API

- [Каталог API Яндекс Карт](#)
- [API | VK для разработчиков](#)
- [API сервиса DaData](#)

Очереди

Очереди предоставляют буфер для временного хранения сообщений и конечные точки, которые позволяют подключаться к очереди для отправки и получения сообщений в асинхронном режиме

публикует сообщение



читает сообщение

Очереди поддерживают получение сообщений двумя методами:

- метод **Pull** подразумевает периодический опрос очереди получателем по поводу наличия новых сообщений
- метод **Push** — отправку уведомления получателю в момент прихода сообщения

**В каких случаях правильнее
использовать монолитную
архитектуру?**

**Какие микросервисы можно
выделить при реализации
приложения для заказа такси?**