

Отчет по первой неделе

В рамках первой недели был разработан бумажный проект сервиса для видеозвонков пользователей Авито, включающий:

- impact map,
- user map,
- архитектуру на уровне сервисов,
- архитектуру на уровне технологий.

На первом этапе были проанализированы требования к сервису, заранее представленные заказчиками. В соответствии с этими требованиями была разработана impact-карта (рисунок 1). Определена основная бизнес-задача, решаемая сервисом, определены основные действующие лица, то, как они могут помочь или помешать достичь цели, и то, с помощью чего это технически возможно.

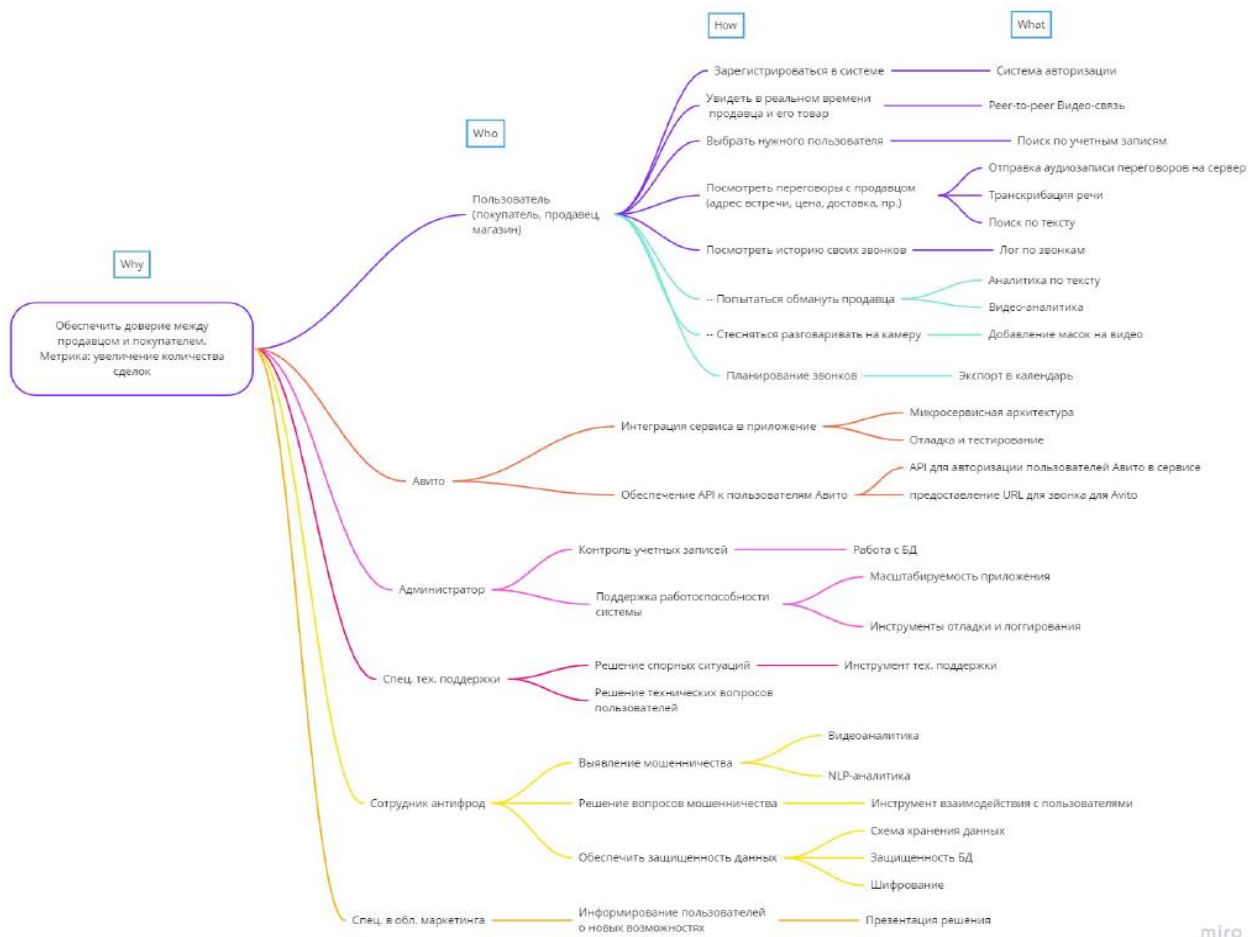


Рисунок 1 – Impact map

Далее составлена user-карта (рисунок 2), на которой отображены основные пользователи разрабатываемого сервиса, доступный им функционал. Функционал декомпозирован на задачи, которые необходимо решить в рамках релизов. Определены задачи на первый релиз.

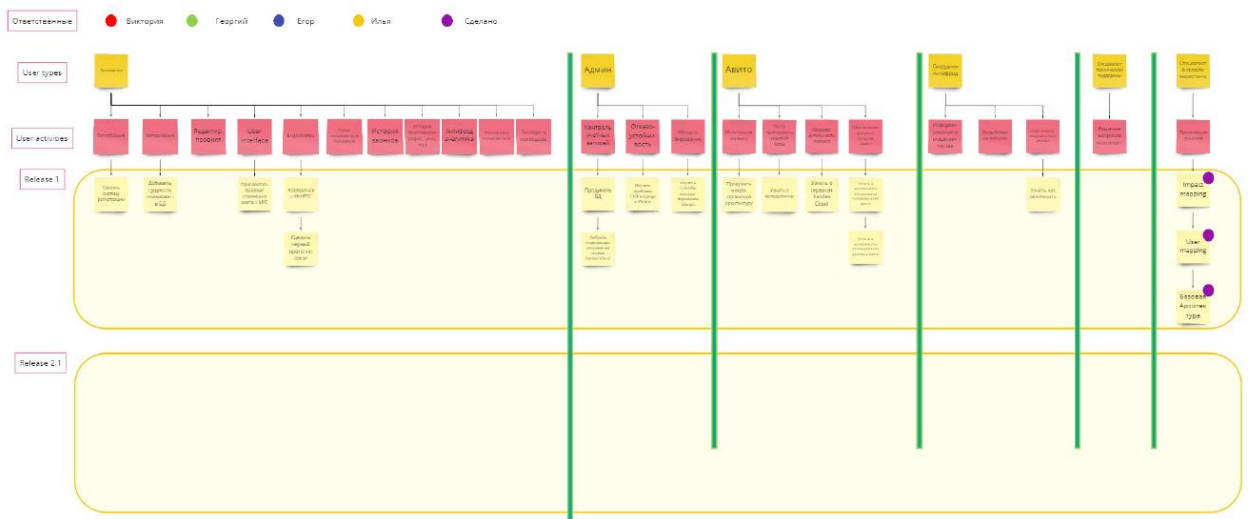


Рисунок 2 – User map

После промежуточной встречи с заказчиками была проработана архитектура сервиса на уровне требуемых сервисов (рисунок 3). Для отказоустойчивости предложено использовать сервис защиты от DDoS атак и сервис балансировки нагрузки. Поскольку требуется, чтобы архитектура была микросервисной, предлагается отдельно использовать Frontend-сервер, реализующий функции графического интерфейса пользователя, взаимодействие с протоколом WebRTC, дополнительную фичу маскирования лица пользователя на изображении. Для работы WebRTC требуется отдельный сигнальный сервер. Бизнес логика должна выполняться в рамках отдельного сервиса. Предполагается использование сервисов хранения медиа-данных и хранения табличных данных.

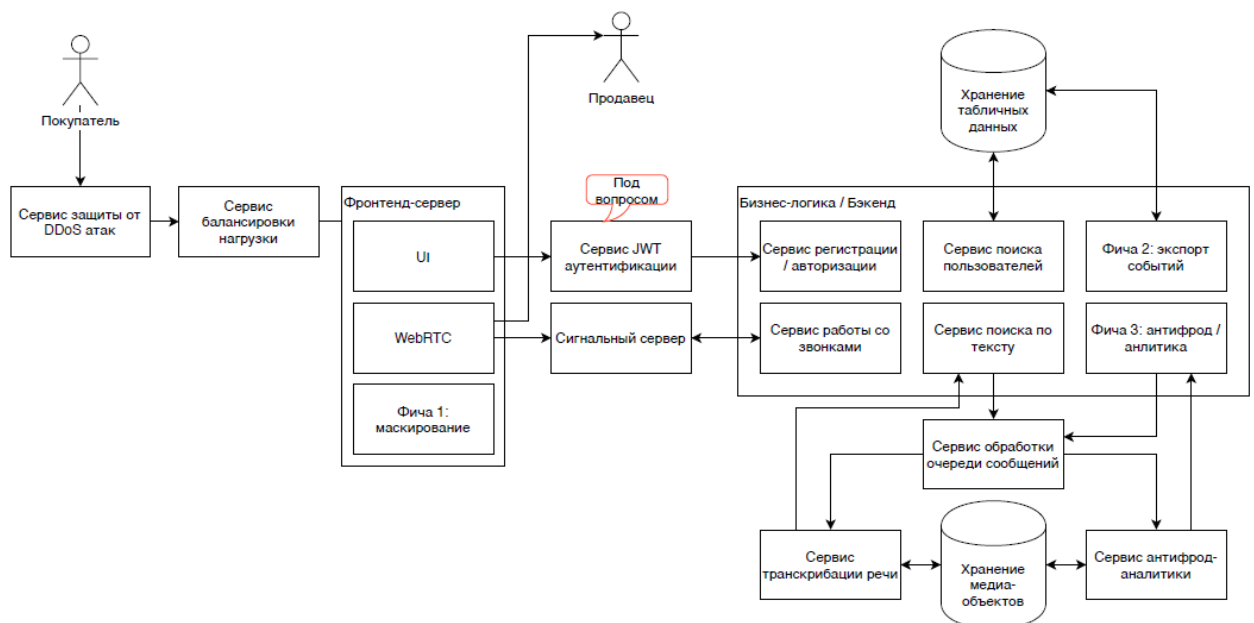


Рисунок 3 – Архитектура на уровне сервисов

На основании предыдущей схемы была разработана архитектура приложения с указанием используемых технологий и решений от Яндекс Облака (рисунок 4). Также на схеме показана очередность разработки компонентов. Для обеспечения горизонтальной масштабируемости и нагрузочной способности приложения предлагается использовать балансировщик нагрузки и Yandex Instance Groups, для создания одинаковых инстансов виртуальной машины с запущенным приложением при увеличении количества запросов. В качестве основного языка программирования для реализации бизнес-логики используется Python. Приложение реализуется на основе Django REST framework с фронтендом, отправляющим Ajax-запросы к бэкенду.

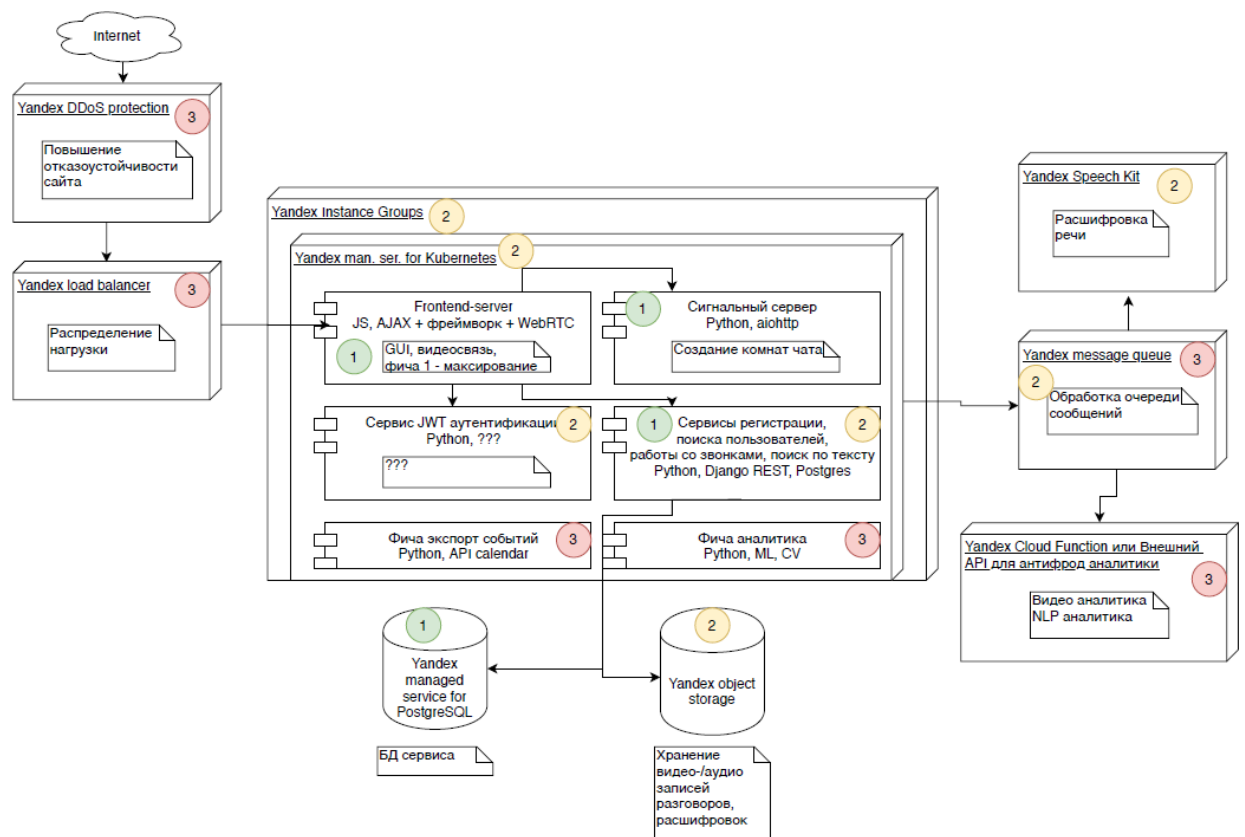


Рисунок 4 – Архитектура приложения с указанием используемых технологий и сервисов Яндекс Облака