**1.  # ADR - выбор архитектурного подхода**

Участники: Геворгян Арсен, Афанасьев Никита, Сахабутдинов Тимур

Дата: 19.09.24

Статус:  принят

**## Контекст**

Основные задачи включают в себя обеспечение высокой доступности, защиту данных пользователей и возможность обработки большого количества одновременных подключений. Ограничения включают бюджет на инфраструктуру и необходимость минимизации задержек при воспроизведении музыки.

**## Рассматриваемые варианты**

Выбор архитектурного подхода:

1. Бессерверный - В бессерверном подходе логика обработки и данные распределены между клиентами, без необходимости в центральном сервере. Бюджет - минимальный.

1. Серверный - В серверном подходе все данные и логика обработки находятся на центральном сервере. Пользователи подключаются к этому серверу для получения музыкального контента. Бюджет - максимальный.

1. P2P - В P2P подходе пользователи напрямую обмениваются данными друг с другом, без необходимости в центральном сервере. Каждый пользователь может быть как потребителем, так и поставщиком контента. Бюджет - минимальный.

**## Решение**

Мы выбрали серверный подход.

**## Обоснование**

Плюсы:

* Полный контроль над инфраструктурой, легкое добавление обновлений.
* Возможность реализации сложных бизнес-логик.
* Высокая отказоустойчивость и безопасность.

Минусы:

* Высокие эксплуатационные расходы.
* Сложность в масштабировании.
* Необходимость в управлении инфраструктурой.

**## Последствия**

**Плюсы**

* Легко обновлять контент и работать с ним.
* Ограничивать доступность контента в связи какими-либо санкциями.
* Возможность вводить резервные сервера и расширять проект.

**Минусы**

* Огромные инвестиции в проект в покупку серверов.
* Возможность к полному отключению сервера из-за Ddos-атак.
* Качество обслуживания зависит от интернета пользователя.

**## Риски**

* Сложность в управлении серверной инфраструктурой. - Сложность заключается в масштабируемости и выделении безопасности. Для устранения этой проблемы необходимо предоставить курсы постоянным разработчикам, чтобы они могли исправлять проблемы.
* Необходимость покупать новый сервер при большом количестве новых клиентов. - Возможно будет не хватать денег на новый сервер, для этого нужно сделать резервный счет, благодаря которому в крайнем случае можно будет купить новый сервис.
* Возможность сбоев в работе серверов. - для этого нужно делать постоянное резервное копирование -> сис. администраторам предоставить курс.

**## Затронутые области**

* Безопасность: Реализация мер безопасности для защиты данных пользователей.
* Масштабируемость: Обеспечение возможности горизонтального масштабирования.
* Отказоустойчивость: Реализация механизмов для обеспечения высокой доступности.

**## История**

19.09.24: Первоначальное обсуждение вариантов архитектуры.