**# ADR - [Выбор архитектурного стиля]**

Участники: Трепалин, Нефонтов, Багаутдинов

Дата: 19.09.2024

Статус: Принято

**## Контекст**

После выбора языка программирования для сервера нам нужно определить как наша система будет структурирована и организована

Бюджет: 300 рублей

Команда: Нефонтов Багаутдинов Трепалин(3 человека)

**## Рассматриваемые варианты**

Монолитная архитектура

Описание:

* Все компоненты системы объединены в одно приложение

Плюсы:

* Простота и легкость разработки - Компоненты монолитной системы тесно связаны, поэтому писать и тестировать такой код сравнительно легко
* Высокая производительность и эффективность - Поскольку все компоненты выполняются в рамках одного процесса, отпадает нужда в межпроцессном взаимодействии.

Минусы:

* Масштабирование - Масштабировать монолиты не так уж просто, ведь все приложение масштабируется как единое целое.
* Сложно вводить обновления: Любые изменения или обновления требуют повторного развертывания всего приложения, и это ведет к увеличению времени простоя и возможным сбоям.

Микросервисная архитектура

Описание:

* Система разбита на независимые сервисы, каждый из которых выполняет свою функцию

Плюсы:

* Масштабируемость - Каждый микросервис можно масштабировать независимо от других, исходя от его конкретных потребностей
* Каждый сервис работает обособленно, так что можно подобрать для него разные технологии и фреймворки. Это открывает целый мир возможностей и позволяет командам выбирать лучшие инструменты для реализации конкретных задач.
* Непрерывная доставка и развертывание - Сервисы не имеют привязки друг к другу, благодаря чему их можно разрабатывать, тестировать и развертывать обособленно.

Минусы:

* Сложность с управлением распределенной системой
* Сложно поддерживать взаимодействие и согласованность данных между сервисами

Модульная архитектура

Описание:

* Система разделена на модули, которые могут быть развернуты и обновлены независимо.
* Баланс между монолитной и микросервисной архитектурой

Плюсы:

* Возможность масштабирования отдельных модулей
* Упрощение управления и обновления

Минусы:

* Возможные сложности с интеграцией модулей

**## Решение**

Микросервисная архитектура

**## Обоснование**

Микросервисная архитектура была выбрана из-за её высокой масштабируемости и возможности независимого развертывания компонентов, что необходимо для нашего приложения. Это позволит быстро добавлять новые функции и реагировать на запросы пользователей.

Монолитная архитектура не подошла нам из-за ограничений в масштабировании и сложности внедрения изменений.

Модульная архитектура дополнительных затрат на проектирование и управление, что в контекте нашего маленького бюджета очень проблематично.

**## Последствия**

Плюсы:

* Легкое масштабирование отдельных компонентов.
* Повышенная гибкость в разработке и развертывании.
* Возможность для отдельных команд работать независимо.

Минусы:

* Сложное управление взаимодействиями между сервисами, особенно с ростом их числа.

**### Риски**

- Потребность в увеличении команды

Меры:

* увеличить команду или же игнорировать и делать втроем

- Сложности с синхронизацией данных между микросервисами

Меры:

* Подумать как синхронизировать данные или игнорировать

**## Затронутые области**

Microservice, API, инфраструктура для развертывания, data-management

**## История**

Вследствие того что это наш третий опыт мы принимаем такие решения.