## ADR - Выбор серверного подхода для фотохостинга

Участники: Зыкин Алексей  
Дата: 5 декабря 2024 года  
Статус: -

**Контекст**

Разрабатывается приложение для фотохостинга, аналогичное Instagram. Основные задачи включают хранение и обработку изображений, управление пользовательскими данными, обеспечение масштабируемости и высокой доступности. Основные ограничения включают необходимость быстрой обработки изображений, поддержку большого числа одновременно подключенных пользователей и защиту данных пользователей.

**Рассматриваемые варианты**

**Монолитная архитектура**  
Плюсы: Простота разработки и развертывания, легкость в управлении.  
Минусы: Сложность масштабирования, высокая связанность компонентов, трудности с внедрением новых технологий.

**Микросервисная архитектура**  
Плюсы: Высокая масштабируемость, возможность независимого развертывания и разработки, гибкость в использовании технологий.  
Минусы: Сложность управления, необходимость в оркестрации (управлении сервисами), потенциальные проблемы с производительностью из-за сетевых вызовов.

**Функциональная архитектура (serverless)**  
Плюсы: Автоматическое масштабирование, отсутствие необходимости в управлении серверами.  
Минусы: Ограничения по времени выполнения функций, сложности с состоянием и долгосрочным хранением данных.

**Решение**

Выбран **микросервисный подход** для разработки серверной части приложения.

**Обоснование**

**Микросервисная архитектура** была выбрана из-за её способности обеспечивать высокую масштабируемость и гибкость. Это особенно важно для фотохостинга, где ожидается значительное количество пользователей и загружаемых изображений. Микросервисы позволят нам разделить функциональность на независимые компоненты (например, обработка изображений, управление пользователями, хранение данных), что упростит разработку и тестирование.

**Монолитная архитектура** была отвергнута из-за трудностей с масштабированием и внедрением новых функций.

**Функциональная архитектура** оказалась недостаточно гибкой для работы с большими объемами данных и сложными процессами обработки изображений.

**Последствия**

Выбор микросервисной архитектуры приведет к более сложной инфраструктуре, требующей управления множеством сервисов. Это потребует дополнительных усилий на настройку и поддержку системы. Однако, преимущества в масштабируемости и гибкости оправдывают эти затраты.

**Риски**

Сложность управления: Множество сервисов могут усложнить развертывание и управление. Для решения этой проблемы можно внедрить систему оркестрации (например, Kubernetes).  
Проблемы с производительностью: Сетевые вызовы между сервисами могут снизить производительность. Для решения необходимо оптимизировать взаимодействие между сервисами и использовать кэширование.  
Сложности с тестированием: Необходимость тестирования множества сервисов может увеличить время на тестирование. Проблема решается внедрением автоматизированных тестов и CI/CD.

**Затронутые области**

* Архитектура серверной части приложения
* База данных
* Системы обработки изображений
* API для взаимодействия с клиентской частью

**История**

Обсуждение архитектурных подходов началось на первом этапе разработки. Были рассмотрены плюсы и минусы каждого подхода. В результате анализа требований проекта было принято решение о выборе **микросервисной архитектуры**.