ГУАП

КАФЕДРА № 43

|  |  |  |
| --- | --- | --- |
| ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: |  |  |

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| доцент, к.т.н., доцент | / |  | / |  | / | Г.С. Армашова-Тельник |
| (должность, учёная степень, звание) |  | (подпись) |  | (дата защиты) |  | (инициалы, фамилия) |

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3

«Система Онлайн торгов для Коллекционных монет»

ПО КУРСУ: «Экономическое обоснование программных проектов»

|  |  |  |  |
| --- | --- | --- | --- |
| РАБОТУ ВЫПОЛНИЛИ СТУДЕНТЫ: | 4134к | / | Р.Р. Усов  Н.А. Костяков  Д.В. Самарин |
|  | (номер группы) |  | (инициалы, фамилия) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | / |  | / |  |
|  |  | (подпись студента) |  | (дата отчета) |

Санкт-Петербург 2025

# Введение

В данном докладе представлено исследование по проектированию микросервисной архитектуры для онлайн-аукциона коллекционных монет. Исходной точкой является опыт разработки онлайн-сервиса для торгов (https://torgi.monety.shop/). Первоначально система реализована на основе монолитной архитектуры, что позволило быстро запустить проект при ограниченных ресурсах небольшой команды. Однако для повышения масштабируемости, отказоустойчивости и гибкости в будущем планируется переход к микросервисной архитектуре.

Кроме технического аспекта, доклад включает анализ бизнес-модели с использованием Business Model Canvas, а также дополнительные визуальные элементы, такие как диаграмма причинно-следственных связей (диаграмма Каору Исикавы), иллюстрирующая влияние различных факторов на проект.

2. Цель проекта

2.1. Формулировка цели по SMART

Цель проекта сформулирована с учётом методологии SMART:

Specific (Конкретность):

Разработать и внедрить архитектуру для онлайн-аукциона коллекционных монет, разделив систему на независимые микросервисы.

Measurable (Измеримость):

Достичь повышения отказоустойчивости системы минимум на 30% и обеспечить масштабирование отдельных сервисов при увеличении нагрузки.

Achievable (Достижимость):

Реализовать переход от монолитной системы к микросервисной архитектуре с использованием Docker, Kubernetes и CI/CD.

Relevant (Актуальность):

Обеспечить гибкость разработки, безопасность транзакций и возможность быстрого внедрения новых функций, что критически важно для онлайн-аукционов редких монет.

Time-bound (Ограниченность во времени):

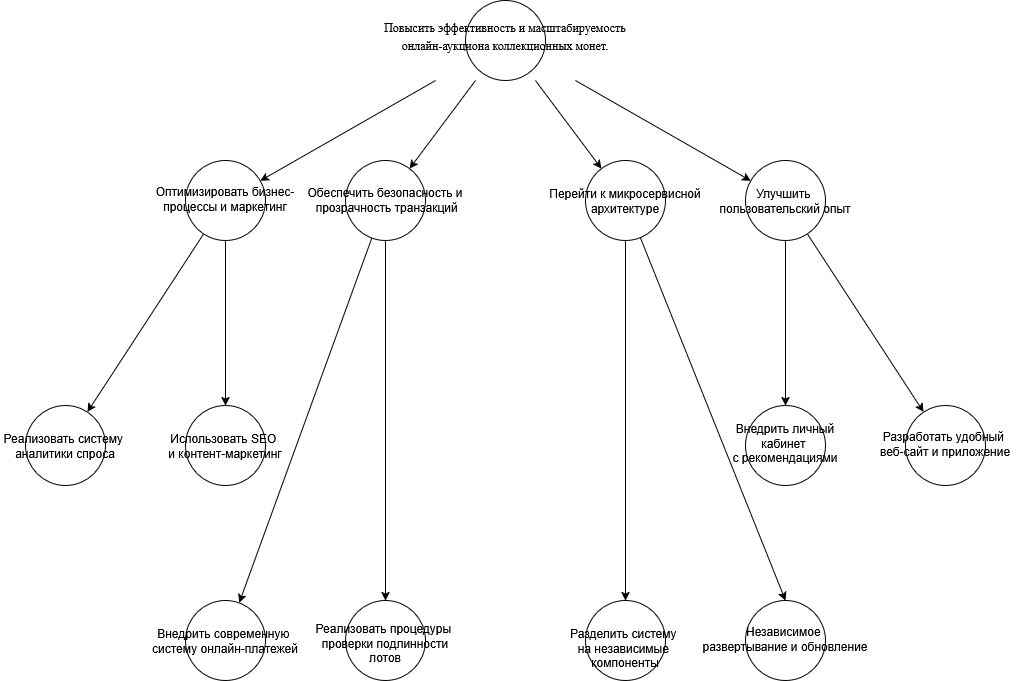
Завершить этап перехода в течение 6 месяцев.

2.2. Дерево целей

Дерево целей визуализирует взаимосвязь главной цели с подцелями:

Главная цель:

Повысить эффективность и масштабируемость онлайн-аукциона коллекционных монет.



Подцель 1:

* Перейти от монолитной архитектуры к микросервисной
* Разделить систему на независимые компоненты
* Обеспечить независимое развертывание и обновление

Подцель 2:

* Обеспечить безопасность и прозрачность транзакций
* Внедрить современную систему онлайн-платежей
* Реализовать процедуры проверки подлинности лотов

Подцель 3:

* Улучшить пользовательский опыт
* Разработать интуитивно понятный веб-сайт и мобильное приложение
* Внедрить личный кабинет с персональными рекомендациями

Подцель 4:

* Оптимизировать бизнес-процессы и маркетинг
* Реализовать систему аналитики для оценки спроса и предложения
* Использовать инструменты контент-маркетинга и SEO для привлечения клиентов
* Каждая из подцелей поддерживает общую стратегию по повышению эффективности работы платформы, обеспечивая её устойчивый рост и конкурентоспособность.

3. PEST-анализ

Анализ внешней среды по модели PEST позволяет оценить факторы, влияющие на реализацию проекта.

3.1. Политические факторы

* Регулирование в сфере IT и обработки персональных данных.
* Государственные инициативы по цифровизации и импортозамещению.

3.2. Экономические факторы

* Доступность инвестиций для IT-проектов.
* Рост российского IT-рынка и увеличение бюджетов на цифровизацию.

3.3. Социальные факторы

* Рост цифровой грамотности и изменение потребительских предпочтений.
* Повышенный спрос на удобные и безопасные онлайн-сервисы.

3.4. Технологические факторы

* Развитие технологий контейнеризации (Docker, Kubernetes) и автоматизации CI/CD.
* Постоянное обновление IT-инфраструктуры и развитие облачных сервисов.

3.5. Выводы по PEST-анализу

Внешняя конъюнктура благоприятна для внедрения микросервисной архитектуры, а поддержка цифровых инициатив и рост IT-рынка создают отличные условия для реализации проекта. Однако жёсткое регулирование и экономическая нестабильность требуют дополнительного внимания на этапе планирования.

4. SWOT-анализ

SWOT-анализ оценивает внутренние возможности и ограничения проекта в контексте внешних факторов.

4.1. Сильные стороны

* Опыт команды в разработке онлайн-сервисов и глубокое понимание специфики монолитной системы.
* Быстрое развёртывание и удобство тестирования MVP.
* Чёткое понимание преимуществ микросервисной архитектуры (масштабируемость, отказоустойчивость).

4.2. Слабые стороны

* Ограниченные ресурсы команды (3 человека) для внедрения сложной микросервисной инфраструктуры.
* Необходимость инвестиций в обучение DevOps и изменение подходов разработки.

4.3. Возможности

* Рост IT-рынка и активное внедрение современных технологий.
* Привлечение инвестиций и стратегических партнёров при масштабировании проекта.
* Возможность быстрого внедрения новых функциональных возможностей за счёт независимого развертывания сервисов.

4.4. Угрозы

* Усиление конкуренции на рынке онлайн-аукционов.
* Возможное увеличение затрат на инфраструктуру и техническое сопровождение микросервисов.
* Риски, связанные с интеграцией множества независимых сервисов.

4.5. Выводы по SWOT-анализу

Сильные стороны проекта, такие как опыт команды и преимущества микросервисного подхода, уравновешиваются ограниченными ресурсами и возможными затратами, что требует поэтапного внедрения и тщательного планирования.

5. Бизнес-модель Canvas (BMC)

Модель Business Model Canvas структурирует ключевые элементы бизнес-модели онлайн-аукциона коллекционных монет.

5.1. Ценностное предложение

Платформа предоставляет коллекционерам удобный и безопасный способ покупки, продажи и обмена коллекционных монет, а для банков – инструмент анализа спроса и установления оптимальных цен.

5.2. Клиентские сегменты

* Коллекционеры монет (начинающие и опытные):

Этот сегмент включает людей, увлечённых нумизматикой, от новичков до опытных коллекционеров. Для начинающих важно получить образовательные материалы, рекомендации по выбору и истории монет, а для опытных коллекционеров – доступ к редким и эксклюзивным лотам, аналитика по рынку и возможность обмена опытом с единомышленниками.

* Дилеры и магазины нумизматики:

Эти клиенты заинтересованы в постоянном поступлении лотов, возможности быстро реализовывать товар и повышении узнаваемости своего бренда. Платформа позволяет дилерам и специализированным магазинам расширить географию продаж и наладить прямой контакт с целевой аудиторией.

* Инвесторы в редкие монеты:

Для этого сегмента ключевым моментом является прозрачность ценообразования и возможность анализа рыночных тенденций. Инвесторы ищут объекты, которые могут стать хорошим вложением средств, и ценят возможность получать детальную информацию о характеристиках и истории продаж каждой монеты.

* Аукционные дома:

Аукционные дома могут использовать платформу как дополнительный канал для проведения торгов, расширяя аудиторию и повышая ликвидность лотов. Сотрудничество позволяет аукционным домам объединять традиционные и онлайн-форматы продаж для более эффективного проведения торгов.

* Банки и финансовые организации:

Для банков проект представляет интерес как источник аналитических данных. Благодаря платформе банки могут анализировать спрос на определённые коллекционные монеты, оценивать рыночную динамику и корректировать ценовую политику. Кроме того, банковские структуры могут предлагать эксклюзивные инвестиционные продукты, связанные с редкими монетами.

5.3. Каналы продвижения и продаж

* Веб-сайт и мобильное приложение:

Основные цифровые инструменты, через которые осуществляется продажа и продвижение услуг. Веб-сайт и мобильное приложение должны быть интуитивно понятными, адаптированными под разные устройства, обеспечивать быстрый доступ к каталогу лотов, истории продаж и аналитике.

* Перенаправление трафика из розничного магазина:

Если у компании уже существует физический магазин, его клиентская база может стать важным каналом привлечения пользователей. Специальные акции, скидки или информационные рассылки могут мотивировать посетителей магазина перейти на онлайн-платформу для участия в аукционах.

* Социальные сети (Facebook, Instagram, TikTok, LinkedIn):

Использование социальных платформ позволяет охватить широкую аудиторию за счёт таргетированной рекламы, публикации историй успеха коллекционеров, а также организации онлайн-мероприятий и конкурсов. Каждая платформа позволяет задействовать разные форматы контента (видео, фото, статьи), что повышает вовлечённость аудитории.

* SEO и контент-маркетинг (блог о нумизматике, истории монет):

Оптимизация сайта под поисковые системы и создание качественного контента помогают привлекать органический трафик. Ведение блога с аналитическими материалами, историческими справками и рекомендациями для коллекционеров повышает авторитет платформы и способствует росту числа постоянных пользователей.

* Партнёрства с аукционными домами, музеями и нумизматическими сообществами:

Сотрудничество с профильными организациями открывает доступ к целевой аудитории, позволяет проводить совместные мероприятия, обмениваться экспертной информацией и привлекать новых клиентов через проверенные каналы.

* Email-рассылки и push-уведомления о редких лотах:

Регулярные рассылки с персонализированными предложениями, новостями о новых лотах и специальными акциями помогают поддерживать интерес пользователей и стимулировать повторные посещения платформы.

5.4. Взаимоотношения с клиентами

* Поддержка через чат-боты и персональных консультантов:

Интегрированные системы поддержки помогают оперативно отвечать на вопросы пользователей, проводить консультации по выбору лотов и решать технические проблемы. Чат-боты обеспечивают круглосуточную поддержку, а персональные консультанты — индивидуальный подход для клиентов с особыми запросами.

* Личный кабинет с рекомендациями по покупке и продаже:

Пользовательский интерфейс должен включать функционал личного кабинета, где клиенты могут видеть историю своих сделок, получать персональные рекомендации на основе анализа их активности и предпочтений, а также управлять подписками и уведомлениями.

* Сообщество коллекционеров, рейтинги продавцов:

Создание площадки для общения коллекционеров и возможность оценки продавцов повышает доверие к платформе. Социальные функции, такие как форумы, рейтинги и отзывы, способствуют формированию активного и вовлечённого сообщества.

* Программы лояльности (бонусы за активность и отзывы):

Для стимулирования постоянного использования платформы можно внедрить бонусные программы, скидки, кэшбэк и другие поощрения за активное участие, положительные отзывы и рекомендации друзьям.

* Образовательные материалы по инвестициям в монеты:

Предоставление обучающих курсов, вебинаров, статей и видеоматериалов помогает пользователям лучше разбираться в рынке коллекционных монет, что в свою очередь повышает их уверенность в инвестиционных решениях и способствует увеличению продаж.

5.5. Источники дохода

* Продажи монет с маржинальностью от 30 до 200%:

Основной источник дохода — разница между закупочной и продажной ценой. При правильном подборе лотов и оптимизации закупочных цен платформа может достигать высокой маржинальности, как в случае с «Нумизматическим набором 2024», закупаемым по выгодной цене и продаваемым с существенной наценкой.

* Реклама на сайте:

Размещение таргетированной рекламы внутри платформы или на её страницах позволяет генерировать дополнительный доход. Рекламодатели могут быть как компании, связанные с нумизматикой, так и смежные отрасли (например, финансовые учреждения или антикварные магазины).

* Платные подписки за доступ к эксклюзивным лотам:

Платформа может предлагать премиум-доступ к эксклюзивным предложениям, аналитическим материалам или раннему просмотру лотов через модель платной подписки, что дополнительно диверсифицирует источники дохода.

5.6. Ключевые ресурсы

* IT-инфраструктура (веб-сайт, мобильное приложение, серверы):

Фундамент платформы — надежная IT-инфраструктура, обеспечивающая высокую производительность, безопасность и масштабируемость системы. Веб-сайт и мобильное приложение являются основными точками взаимодействия с клиентами.

* База данных монет с характеристиками, ценами и историей продаж:

Обширная и регулярно обновляемая база данных позволяет хранить подробную информацию о каждом лоте, что является важным ресурсом для аналитики, формирования рекомендаций и обеспечения прозрачности сделок.

* Система онлайн-платежей с высокой степенью защиты:

Интеграция современных систем платежей обеспечивает безопасность транзакций, защиту персональных данных пользователей и быстрое проведение операций, что повышает доверие клиентов к платформе.

* Команда маркетологов, разработчиков и специалистов по нумизматике:

Квалифицированные специалисты играют ключевую роль в развитии платформы, от разработки технических решений до продвижения продукта на рынке и обеспечения экспертной оценки коллекционных монет.

* Поставщики монет:

Надежные поставщики и партнёры, предоставляющие лоты, являются важным ресурсом для формирования ассортимента и обеспечения эксклюзивности предложения.

5.7. Ключевые виды деятельности

* Разработка и поддержка платформы:

Регулярное обновление функционала, поддержка IT-инфраструктуры и внедрение новых технологических решений обеспечивают стабильную работу платформы и её конкурентоспособность.

* Маркетинг и привлечение коллекционеров и продавцов:

Реализация маркетинговых стратегий, включая онлайн-рекламу, SMM, SEO и контент-маркетинг, позволяет привлекать новых пользователей, удерживать текущих клиентов и формировать лояльное сообщество.

* Организация онлайн-аукционов:

Проведение торгов в режиме реального времени, настройка аукционных лотов и обеспечение прозрачности торгового процесса являются ключевыми элементами деятельности платформы.

* Проверка подлинности и сертификация монет:

Внедрение процессов проверки подлинности с участием экспертов и сертификационных агентств помогает снизить риск подделок и повысить доверие пользователей.

* Обеспечение безопасности сделок и платежей:

Постоянный контроль за безопасностью IT-систем, мониторинг транзакций и соблюдение стандартов защиты данных являются критически важными для поддержания репутации платформы.

5.8. Ключевые партнеры

* Аукционные дома:

Сотрудничество с аукционными домами позволяет расширить аудиторию, интегрировать традиционные форматы торгов с онлайн-платформой и обеспечить дополнительные каналы сбыта для коллекционных лотов.

* Банкнотно-монетные дворы и частные минцеи:

Эти партнеры предоставляют качественные и проверенные лоты, что способствует повышению доверия к платформе и улучшению ассортимента коллекционных монет.

* Онлайн-платежные системы:

Партнерство с ведущими платежными системами обеспечивает надежность и скорость проведения транзакций, а также интеграцию с различными финансовыми инструментами.

* Эксперты-нумизматы и сертификационные агентства:

Экспертиза независимых специалистов помогает в проверке подлинности лотов, сертификации и формировании экспертных оценок, что существенно снижает риски мошенничества и подделок.

5.9. Структура затрат

* Разработка и поддержка платформы:

В эту статью входят затраты на создание программного обеспечения, оплату труда разработчиков, техническую поддержку и обновление функционала платформы.

* Хостинг и серверные мощности:

Расходы на аренду серверов, облачные сервисы, обеспечение безопасности и масштабируемости IT-инфраструктуры.

* Маркетинг и реклама:

Инвестиции в онлайн-рекламу, продвижение в социальных сетях, SEO, создание контента и проведение PR-мероприятий для привлечения пользователей.

* Комиссии платежных систем:

Расходы, связанные с обработкой транзакций, комиссионные сборы, обеспечивающие проведение безопасных онлайн-платежей.

* Оплата экспертов для сертификации монет:

Затраты на услуги профессиональных нумизматов и сертификационных агентств, необходимых для проверки подлинности и оценки лотов.

5.10. Проблемы, решаемые проектом и стейкхолдеры

Проблемы:

* Отсутствие удобного и безопасного рынка для коллекционеров.
* Риск подделок и мошенничества.
* Недостаток прозрачного ценообразования.
* Ограниченные возможности для проведения аукционов.
* Разрозненность нумизматического сообщества.

Стейкхолдеры:

Программисты – получают оплату за реализацию системы.

Заказчик – получает конкурентное и перспективное решение.

Банки – анализируют спрос для оптимизации цен.

Нумизматы – получают доступ к редким лотам.

Службы доставки – обеспечивают логистику лотов.

CRM-системы – интегрируются для управления большим числом лотов.

6. Диаграмма причинно-следственных связей

В презентации представлена диаграмма Каору Исикавы, которая иллюстрирует причинно-следственные связи в проекте. Эта диаграмма помогает определить, какие факторы положительно или отрицательно влияют на успешное выполнение проекта, а также выявить основные «вредные» и «помогающие» факторы.

В контексте нашего проекта ключевые элементы диаграммы могут быть распределены по следующим категориям:

Технологии:

* Используемые инструменты разработки (Docker, Kubernetes, CI/CD)
* Проблемы интеграции микросервисов
* Сложности поддержки IT-инфраструктуры

Организационные факторы:

* Компетенции и опыт команды
* Стратегия управления проектом
* Наличие процессов мониторинга и тестирования

Ресурсы:

* Финансовые затраты на разработку и поддержку платформы
* Инвестиции в маркетинг и рекламу
* Доступность квалифицированных специалистов (DevOps, эксперты по нумизматике)

Методы и процессы:

* Процедуры проверки подлинности монет
* Механизмы взаимодействия с клиентами (чат-боты, личные кабинеты)
* Стандарты обеспечения безопасности транзакций

Каждый из этих элементов оказывает влияние на достижение главной цели проекта. Диаграмма позволяет визуально отобразить, какие факторы способствуют успеху (помогающие факторы) и какие могут стать источником риска (вредные факторы), что помогает своевременно скорректировать стратегию и оптимизировать процессы.

7. План реализации проекта

На основе анализа и диаграммы причинно-следственных связей предлагается следующий план реализации:

1. Анализ текущей монолитной системы и выделение компонентов, пригодных для преобразования в микросервисы.
2. Постепенное разделение системы с внедрением контейнеризации (Docker) и оркестрации (Kubernetes).
3. Настройка CI/CD для автоматизации тестирования и развертывания новых версий.
4. Внедрение системы мониторинга и логирования для отслеживания работы микросервисов.
5. Обучение команды новым подходам и технологиям DevOps.
6. Постоянный анализ факторов, выявленных на диаграмме Исикавы, и корректировка стратегии в зависимости от внешних и внутренних условий.

8. Заключение

В ходе лабораторной работы проведён всесторонний анализ проекта, охватывающий как техническую сторону (архитектура системы), так и бизнес-процессы (бизнес-модель Canvas). Исследование показало, что для старта разработки онлайн-аукциона коллекционных монет монолитная архитектура является оправданным выбором для небольшой команды. Однако при росте нагрузки и расширении функционала переход к микросервисной архитектуре позволит обеспечить необходимую масштабируемость, отказоустойчивость и гибкость системы.

Дополнительные визуальные инструменты, такие как диаграмма причинно-следственных связей, помогают глубже понять влияние различных факторов на реализацию проекта, а чётко структурированная бизнес-модель Canvas обеспечивает комплексное рассмотрение всех ключевых аспектов работы платформы.

9. Список источников

1. O’Reilly Microservices Adoption Report (2020, 2023)
2. GitLab State of DevOps Report (2021)
3. Отчёт Минцифры РФ о развитии российского ИТ-сектора (2023)
4. Аналитика РБК и CNews по внедрению микросервисов в России (2023)
5. Исследование Gartner: прогноз по росту микросервисных систем (2022-2025)
6. Nixys – «Монолит или микросервисы: какую IT-инфраструктуру предпочитают крупные компании в России?» (https://nixys.ru/monolit-ili-mikroservisy-kakuju-it-infrastrukturu-predpochitajut-krupnye-kompanii-v-rossii/)
7. ComNews – «Почему микросервисная архитектура – фундамент современного банковского IT-ландшафта?» (https://www.comnews.ru/content/234270/2024-07-15/2024-w29/1013/pochemu-mikroservisnaya-arkhitektura-fundament-sovremennogo-bankovskogo-it-landshafta)
8. Atlassian – «Микросервисы против монолита: сравнение архитектур» (https://www.atlassian.com/ru/microservices/microservices-architecture/microservices-vs-monolith)
9. WorkSolutions – «Микросервисы vs монолит: разница архитектур и руководство по переходу» (<https://worksolutions.ru/blog/mikroservisy-vs-monolit-raznicza-arhitektur-i-rukovodstvo-po-perehodu/>)

Приложение 1:

Практика №1.

# Введение

Сегодня мы представляем вам исследование по **проектированию микросервисной архитектуры**, основанное на нашем опыте разработки **онлайн-сервиса для торгов**.

<https://torgi.monety.shop/>

Наша работа была посвящена созданию онлайн-платформы для проведения аукционов, в рамках которой пользователи могут регистрироваться, делать ставки и получать уведомления о торгах в реальном времени. Первоначально этот сервис был спроектирован в монолитной архитектуре, однако при детальном анализе стало очевидно, что переход к микросервисному подходу мог бы значительно улучшить масштабируемость, отказоустойчивость и гибкость системы.

Сегодня мы расскажем вам о:

1. Принципах микросервисной архитектуры и её преимуществах.
2. Сравнении с монолитным подходом на примере моей системы.
3. Реальных примерах влияния микросервисов на эффективность ведущих IT-компаний.

# Что такое микросервисная архитектура?

Микросервисная архитектура – это один из наиболее популярных и гибких подходов к проектированию программных систем. Она основана на разделении приложения на независимые сервисы, которые взаимодействуют друг с другом через API или системы обмена сообщениями. Каждый такой сервис выполняет отдельную бизнес-функцию, что делает систему более отказоустойчивой, масштабируемой и удобной в разработке.

# Краткая история возникновения микросервисов

Хотя концепция разбиения системы на модули существовала давно, микросервисный подход в его современном виде начал активно развиваться в 2010-х годах. До этого подавляющее большинство программных решений строились на монолитной архитектуре, где все модули и компоненты работали как единое целое.

С ростом IT-индустрии возникли проблемы:

1. Обновление больших приложений стало слишком сложным.
2. Изменение одной части системы могло повлиять на всю систему.
3. Масштабирование было дорогостоящим и неэффективным.

Такие гиганты, как Netflix, Amazon, Uber, Google, eBay, столкнулись с этими ограничениями и начали внедрять концепцию микросервисов, позволяющую разрабатывать, развертывать и масштабировать отдельные части системы независимо друг от друга.

# Ключевые принципы микросервисной архитектуры

1. Разделение бизнес-логики на независимые сервисы

В микросервисной архитектуре каждая часть системы выполняет одну конкретную задачу. Например, в интернет-магазине могут быть отдельные микросервисы для:

1. Обработки заказов
2. Регистрации и авторизации пользователей
3. Системы платежей
4. Службы доставки
5. Уведомлений и email-рассылок

Эти микросервисы могут работать независимо друг от друга, что упрощает модификацию, тестирование и развертывание.

2. Независимое развертывание и масштабирование

1. Если один микросервис перегружен, можно масштабировать только его, не затрагивая остальные.
2. Если требуется внести изменения в один модуль – можно обновить только этот микросервис без остановки всей системы.

3. Гибкость в выборе технологий

Каждый микросервис может использовать разные языки программирования, базы данных и инструменты, подходящие для своей задачи. Например:

1. Go – для высоконагруженных API
2. Python – для машинного обучения
3. Node.js – для работы с реальным временем
4. PostgreSQL – для транзакционных данных
5. MongoDB – для хранения логов и событий

Такой подход делает систему более оптимизированной и производительной.

4. Взаимодействие сервисов через API или системы очередей сообщений

Микросервисы могут взаимодействовать друг с другом с помощью:

1. REST API (HTTP-запросы, JSON)
2. GraphQL (гибкие запросы к данным)
3. gRPC (более быстрый двоичный протокол)
4. Системы очередей сообщений (RabbitMQ, Kafka, NATS)

Это позволяет сервисам не зависеть друг от друга напрямую, а взаимодействовать асинхронно.

5. Высокая отказоустойчивость

Если один микросервис выходит из строя, это не ломает всю систему. Другие сервисы могут продолжать работать в штатном режиме.

Например, если сервис платежей недоступен, пользователь все равно может:

1. Добавить товар в корзину
2. Получить уведомления о скидках
3. Просматривать каталог

Таким образом, отказоустойчивость системы повышается, а критические сбои становятся менее разрушительными.

# Технические аспекты микросервисной архитектуры

1. Контейнеризация (Docker, Kubernetes)

Для эффективного управления микросервисами каждый сервис разворачивается в отдельном контейнере. Это позволяет:

1. Запускать микросервисы в изолированной среде.
2. Быстро разворачивать новые версии сервисов.
3. Управлять нагрузкой с помощью Kubernetes, который автоматически масштабирует контейнеры при росте трафика.

2. Автоматизация CI/CD (Continuous Integration / Continuous Deployment)

Микросервисные системы активно используют CI/CD-подход, который включает:

1. Автоматическую проверку кода после каждого изменения.
2. Автоматическое развертывание новых версий сервисов.
3. Мониторинг и откат изменений, если что-то пошло не так.

3. Мониторинг и логирование

Из-за большого количества сервисов важно следить за их работой.

Используются:

1. Prometheus и Grafana – для мониторинга метрик.
2. ELK Stack (Elasticsearch, Logstash, Kibana) – для сбора и анализа логов.
3. Jaeger, Zipkin – для трассировки запросов между сервисами.

# Ключевые преимущества микросервисов

1. Гибкость разработки – легко добавлять новые функции и обновлять существующие.
2. Лучшее масштабирование – можно увеличивать только нагруженные компоненты, а не всю систему.
3. Отказоустойчивость – сбой в одном сервисе не ломает всю систему.
4. Ускорение разработки – команды могут работать независимо друг от друга.
5. Использование разных технологий – можно выбрать лучший стек для каждой задачи.

# Когда микросервисная архитектура НЕ подходит?

1. Слишком сложная для небольших проектов

1. Если приложение небольшое, микросервисы создадут ненужную сложность.
2. Например, для обычного веб-сайта монолит будет лучше.

2. Требует сильной команды DevOps

1. Нужно уметь настраивать контейнеризацию, автоматический деплой, мониторинг.
2. Без грамотного DevOps-специалиста внедрение микросервисов будет сложным.

3. Увеличение сетевой нагрузки

1. В монолите компоненты вызывают друг друга напрямую.
2. В микросервисах данные передаются через сеть, что может замедлить работу системы.

Сравнение микросервисной и монолитной архитектуры:

Почему на начальном этапе мой проект реализован как монолит, учитывая команду из двух разработчиков?

Мы уже рассмотрели ключевые аспекты микросервисной архитектуры, её преимущества и недостатки, а также провели сравнительный анализ с монолитным подходом. Теперь важно не просто теоретически сопоставить эти архитектуры, но и учесть реальные условия разработки – в данном случае маленькую команду из трех человек, работающих над проектом.

1. Почему монолит был наилучшим выбором для старта разработки?

При проектировании любого программного решения важно учитывать ресурсы, время и масштабы разработки. Наш проект разрабатывался тремя людьми, что накладывает определенные ограничения:

1. Ограниченные ресурсы – отсутствие отдельной команды DevOps для настройки сложной инфраструктуры.
2. Необходимость быстрого создания работающего прототипа – MVP (Minimal Viable Product) должен был быть готов в разумные сроки.
3. Фокус на разработку бизнес-логики, а не на поддержание распределенной системы. Проще тестировать и отлаживать – вся логика находится в одном коде.

# Главное преимущество монолита на старте:

# Минимальные накладные расходы на инфраструктуру, быстрое развертывание и удобная отладка кода.

Если бы изначально мы выбрали микросервисную архитектуру, это потребовало бы:

1. Разработки отдельных сервисов и API для связи между ними.
2. Настройки системы контейнеризации (Docker + Kubernetes).
3. Настройки CI/CD для каждого микросервиса.
4. Управления мониторингом и логированием множества сервисов.

Для команды из трех человек это бы значительно увеличило сложность и сроки разработки.

# 2. Почему на начальном этапе монолитная архитектура лучше?

Монолит – это быстрый старт и удобство разработки.

На старте разработки важно быстро проверить работоспособность идеи, а не сразу строить масштабируемую архитектуру.

Как реализовано в нашем проекте?

1. Все функции аукциона, пользователей и уведомлений находятся в одном приложении.
2. Разработано как единый FastAPI-сервис, который работает с MongoDB.
3. Разворачивается в одном контейнере Docker, не требуя сложной оркестрации.

Какие ограничения у монолита, если система будет расти?

1. Невозможно масштабировать отдельные компоненты (например, если растёт нагрузка на аукционы).
2. Любое обновление требует перезапуска всей системы.
3. Сложнее работать над разными частями кода параллельно, если разработчиков станет больше.

Вывод: Монолит – это хороший выбор для старта, но он может стать узким местом при росте нагрузки.

# 3. Когда микросервисная архитектура становится оправданной?

Хотя монолит является оптимальным выбором на старте, при дальнейшем росте системы переход к микросервисной архитектуре будет логичным шагом.

1. Если в проекте станет больше разработчиков, они смогут работать над отдельными микросервисами.
2. Если появится большая нагрузка на аукционы, можно будет масштабировать сервис аукционов отдельно.
3. Если потребуется гибкость, микросервисная архитектура позволит обновлять отдельные модули без остановки всей системы.

# Статистика внедрения микросервисной архитектуры и её влияние на отрасль

Микросервисная архитектура (МСА) продолжает набирать популярность в мировой практике разработки программного обеспечения, предлагая гибкость, масштабируемость и ускоренное внедрение изменений.

Мировые тенденции:

1. Рост использования микросервисов: Согласно отчёту O'Reilly за 2020 год, около 68% компаний уже внедрили микросервисную архитектуру в своих проектах.

https://libeldoc.bsuir.by/bitstream/123456789/52022/1/Shirshov\_Transformaciya.pdf?utm\_source=

1. Преимущества для бизнеса: Исследования показывают, что внедрение МСА способствует повышению эффективности и производительности программных систем, что положительно сказывается на бизнес-показателях компаний.

https://moitvivt.ru/ru/journal/article?id=1561&utm\_source=

Российский сегмент:

В России наблюдается активное развитие ИТ-отрасли, что создаёт благоприятные условия для внедрения современных архитектурных подходов, включая микросервисы.

1. Рост ИТ-рынка: В 2023 году объём российского ИТ-рынка достиг 3,06 трлн рублей, что на 12% больше по сравнению с предыдущим годом. Такой рост обусловлен, в том числе, переходом на отечественные решения и активным развитием новых технологий.
2. Импортозамещение и развитие отечественных решений: После ухода ряда западных вендоров российские компании начали активно разрабатывать собственное программное обеспечение. Около 76% крупных и средних российских компаний инициировали создание собственного ПО, что способствует внедрению современных архитектурных подходов, таких как микросервисы.
3. Рынок CRM-систем: В 2023 году объём российского рынка CRM-систем превысил 28 млрд рублей. Компании всё чаще выбирают решения, основанные на микросервисной архитектуре, что позволяет им быстро адаптироваться к изменениям и обеспечивать гибкость бизнес-процессов.

Влияние на компании:

1. Переход на микросервисную архитектуру позволяет компаниям:
2. Ускорить разработку и внедрение новых функций.
3. Обеспечить масштабируемость и отказоустойчивость систем.
4. Сократить время простоя и повысить удовлетворённость пользователей.

# Таблица: Ключевые статистические данные о микросервисной архитектуре

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория | Показатель/Аспект | Описание | Пример значения/статистика | + | - |
| Уровень внедрения в мире | К 2020 году около 68% компаний внедрили микросервисы. |  |  |  |  |
| Производительность | Время отклика (Response Time) | Среднее время обработки запроса микросервисом. | 50-200 мс (в зависимости от нагрузки и оптимизации) | Высокая производительность за счет изоляции сервисов. | Задержки из-за межсервисного взаимодействия (сети, API-вызовы). |
| Производительность | Пропускная способность (Throughput) | Количество запросов, обрабатываемых в единицу времени. | 1000-5000 запросов/сек (в зависимости от инфраструктуры) | Легко масштабировать отдельные сервисы под нагрузку. | Сложность балансировки нагрузки между сервисами. |
| Масштабируемость | Количество микросервисов | Общее число микросервисов в системе. | 50-100 микросервисов (в среднем для крупных компаний) | Гибкость в масштабировании отдельных компонентов. | Увеличение сложности управления множеством сервисов. |
| Масштабируемость | Горизонтальное масштабирование | Возможность добавлять новые экземпляры микросервисов для обработки нагрузки. | Увеличение на 10 экземпляров при росте нагрузки на 20% | Быстрое реагирование на изменения нагрузки. | Высокие затраты на инфраструктуру. |
| Надежность | Время безотказной работы (Uptime) | Процент времени, когда система работает без сбоев. | 99,9% (для хорошо спроектированных систем) | Изоляция сбоев: отказ одного сервиса не влияет на всю систему. | Сложность обеспечения отказоустойчивости всех сервисов. |
| Надежность | Частота отказов (Failure Rate) | Количество сбоев на 1000 запросов. | 0,1% (в хорошо настроенных системах) | Легче локализовать и исправить ошибки. | Необходимость мониторинга и обработки сбоев в каждом сервисе. |
| Сложность управления | Количество зависимостей между микросервисами | Число связей между микросервисами. | 200+ зависимостей (в крупных системах) | Модульность и независимость сервисов. | Высокая сложность управления зависимостями и версионированием. |
| Сложность управления | Время развертывания (Deployment Time) | Среднее время, необходимое для развертывания изменений в микросервисе. | 5-10 минут (при использовании CI/CD) | Быстрое внедрение изменений в отдельные сервисы. | Сложность координации развертывания множества сервисов. |
| Затраты | Стоимость инфраструктуры | Затраты на поддержку инфраструктуры микросервисов. | (в зависимости от масштаба) | Оптимизация затрат за счет масштабирования только нужных сервисов. | Высокие начальные затраты на инфраструктуру и инструменты. |
| Затраты | Затраты на разработку и поддержку | Затраты на команду разработчиков и DevOps. | (в зависимости от масштаба) | Ускорение разработки за счет независимых команд. | Необходимость высококвалифицированных специалистов. |
| Гибкость | Поддержка различных технологий | Возможность использовать разные языки программирования и фреймворки. | Node.js, Python, Java, Go и др. | Выбор лучшего инструмента для каждой задачи. | Сложность интеграции разнородных технологий. |
| Безопасность | Уровень безопасности | Защита данных и API в микросервисной архитектуре. | Зависит от реализации (OAuth, JWT, шифрование) | Изоляция сервисов повышает безопасность. | Сложность обеспечения безопасности всех сервисов. |
| Тестирование | Сложность тестирования | Уровень сложности тестирования микросервисов. | Высокая (необходимость тестирования каждого сервиса и их взаимодействия) | Легче тестировать небольшие изолированные компоненты | Требуется больше времени и ресурсов на интеграционное тестирование |

Однако стоит учитывать, что внедрение МСА требует значительных инвестиций в обучение персонала, изменение инфраструктуры и внедрение систем мониторинга и управления микросервисами.

В целом, мировые и российские тенденции показывают, что микросервисная архитектура становится ключевым элементом в стратегии развития ИТ-систем, способствуя повышению эффективности и конкурентоспособности компаний.

# Заключение: Монолит сейчас, микросервисы – в будущем

1. Для команды из 3 человек монолит – логичный выбор, так как он позволяет быстро разрабатывать и развертывать систему.
2. Микросервисная архитектура становится актуальной при росте нагрузки, когда нужно масштабирование и высокая отказоустойчивость.
3. В будущем, при росте системы, можно разделить функционал на независимые сервисы и сделать систему более гибкой.

Главный вывод:

На начальном этапе разработки нашего проекта монолитная архитектура была наилучшим выбором, учитывая маленькую команду и необходимость быстрой разработки. Однако при дальнейшем развитии переход к микросервисному подходу позволит решить проблемы масштабируемости и отказоустойчивости.

Источники и аналитика:

Основные:

1. O’Reilly Microservices Adoption Report (2020, 2023)
2. GitLab State of DevOps Report (2021)
3. Отчёт Минцифры РФ о развитии российского ИТ-сектора (2023)
4. Аналитика РБК и CNews по внедрению микросервисов в России (2023)
5. Исследование Gartner: прогноз по росту микросервисных систем (2022-2025)

Дополнительные:

1. **Nixys** – "Монолит или микросервисы: какую IT-инфраструктуру предпочитают крупные компании в России?" https://nixys.ru/monolit-ili-mikroservisy-kakuju-it-infrastrukturu-predpochitajut-krupnye-kompanii-v-rossii/
2. **ComNews** – "Почему микросервисная архитектура – фундамент современного банковского IT-ландшафта?" https://www.comnews.ru/content/234270/2024-07-15/2024-w29/1013/pochemu-mikroservisnaya-arkhitektura-fundament-sovremennogo-bankovskogo-it-landshafta
3. **Atlassian** – "Микросервисы против монолита: сравнение архитектур" https://www.atlassian.com/ru/microservices/microservices-architecture/microservices-vs-monolith
4. **WorkSolutions** – "Микросервисы vs монолит: разница архитектур и руководство по переходу" https://worksolutions.ru/blog/mikroservisy-vs-monolit-raznicza-arhitektur-i-rukovodstvo-po-perehodu/

Приложение 2:

Практика №2.

Модель Business Model Canvas (BMC), разработанная Александром Остервальдером и Ивом Пинье, — это инструмент для описания, анализа и разработки бизнес-моделей. Она состоит из 9 ключевых блоков, которые помогают структурировать идею бизнеса и понять его ключевые элементы.

Приведу описание системы онлайн торгов коллекционными монетами в формате модели А. Остервальдера и И. Пинье

1. Ценностное предложение

Платформа предоставляет коллекционерам удобный и безопасный способ покупать, продавать и обменивать коллекционные монеты, которые зачастую могут быть недоступны в рознице, а банкам, которые не знают, по какой цене выставлять монеты, прикинуть спрос на товар

2. Клиентские сегменты

• Коллекционеры монет (начинающие и опытные)

• Дилеры и магазины нумизматики

• Инвесторы в редкие монеты

• Аукционные дома

• Банки и финансовые организации (покупка инвестиционных монет)

3. Каналы продвижения и продаж

• Веб-сайт и мобильное приложение

• Перекачка трафика из розничного магазина на сайт торгов

• Социальные сети (Facebook, Instagram, TikTok, LinkedIn)

• SEO и контент-маркетинг (блог о нумизматике, истории монет)

• Партнёрства с аукционными домами, музеями и нумизматическими обществами

• Email-рассылки и push-уведомления о редких монетах

4. Взаимоотношения с клиентами

• Поддержка клиентов через чат-ботов и персональных консультантов

• Личный кабинет с рекомендациями по покупке и продаже

• Сообщество коллекционеров, рейтинги продавцов

• Программы лояльности (бонусы за активность и отзывы)

• Образовательные материалы по инвестициям в монеты

5. Источники дохода

• С продаж: Если получать монеты напрямую от центробанка, предположительная маржинальность может колебаться от 30 до 200% в зависимости от тиража монеты. Например «Нумизматический набор 2024» стоил в закупке 400 рублей напрямую у центробанка, а на системе торгов, получилось реализовать каждый набор по 1000 рублей

• С рекламы на сайте

• Платные подписки за доступ к эксклюзивным лотам

6. Ключевые ресурсы

• IT-инфраструктура (веб-сайт, мобильное приложение, серверы)

• База данных монет с характеристиками, ценами, историей продаж

• Система онлайн-платежей с защитой сделок

• Команда маркетологов, разработчиков и специалистов по монетам

• Поставщик монет

7. Ключевые виды деятельности

• Разработка и поддержка платформы

• Маркетинг и привлечение коллекционеров и продавцов

• Организация онлайн-аукционов

• Проверка подлинности и сертификация монет

• Обеспечение безопасности сделок и платежей

8. Ключевые партнеры (Key Partners)

• Аукционные дома

• Банкнотно-монетные дворы и частные минцеи

• Онлайн-платежные системы

• Эксперты-нумизматы и сертификационные агентства

9. Структура затрат

• Разработка и поддержка платформы

• Хостинг и серверные мощности

• Маркетинг и реклама

• Комиссии платежных систем

• Оплата экспертов для сертификации монет

Какие проблемы решит проект?

1. Отсутствие удобного и безопасного рынка

2. Риск подделок и мошенничества

3. Отсутствие прозрачного ценообразования

4. Ограниченные возможности для аукционов

5. Разрозненность нумизматического сообщества

Стейкхолдеры:

• Программисты, которые получат оплату за простую работу по реализации нетрудной системы

• Заказчик, который получит перспективную работу

• Банки, которые тоже продают монеты, смогут более четко видеть спрос на определенные позиции

• Нумизматы, которые смогут купить редкие лоты

• Службы доставки, которые будут возить лоты

• CRM системы (лотов то много будет)