

ООО «АВИСКЛАУД»
ИНН/КПП 5959003518/595901001
ОГРН 1195958026747
e-mail: aviscloud@yandex.ru
телефон: +79097260177



Описание технической архитектуры программного обеспечения

1. Общие сведения

Наименование ПО: Программный модуль «Копия сделки | Повторные сделки»

Версия: 1.0.0

Разработчик: ООО «АВИСКЛАУД»

2. Обозначения, сокращения и определения

- API — Application Programming Interface, программный интерфейс приложения.
- UI — User Interface, пользовательский интерфейс.
- DTO — Data Transfer Object, объект передачи данных.
- HTTPS — защищённый протокол передачи данных.
- amoCRM — облачная CRM-система.
- ПО — программное обеспечение.

3. Введение

Программный модуль «Копирование сделки» относится к классу интеграционных клиент-серверных решений, предназначенных для расширения функциональных возможностей облачной CRM-системы amoCRM. Он представляет собой инструмент автоматизации, позволяющий пользователям создавать новые сделки на основе уже существующих, переносить их между воронками и изменять ответственных лиц без необходимости ручного ввода данных.

Главная цель модуля — обеспечить конечным пользователям удобный и быстрый механизм работы с карточками сделок, снизить вероятность ошибок при повторном вводе информации и ускорить выполнение бизнес-процессов. Модуль поддерживает различные сценарии применения, включая копирование сделки внутри одной воронки по принципу «шаблона», перенос сделки в другую воронку с сохранением ключевых данных, а также передачу сделки другому сотруднику для распределения нагрузки.

Программный модуль ориентирован на организации, использующие amoCRM для управления продажами и взаимодействия с клиентами: от индивидуальных предпринимателей и небольших компаний до средних и крупных предприятий. Он может применяться в отделах продаж, клиентского сервиса и постпродажного сопровождения для повышения эффективности обработки заявок.

Функциональность модуля может быть расширена за счёт интеграции с дополнительными механизмами amoCRM, адаптации логики копирования под отраслевые особенности или добавления новых параметров при передаче данных между воронками.

4. Архитектура ПО

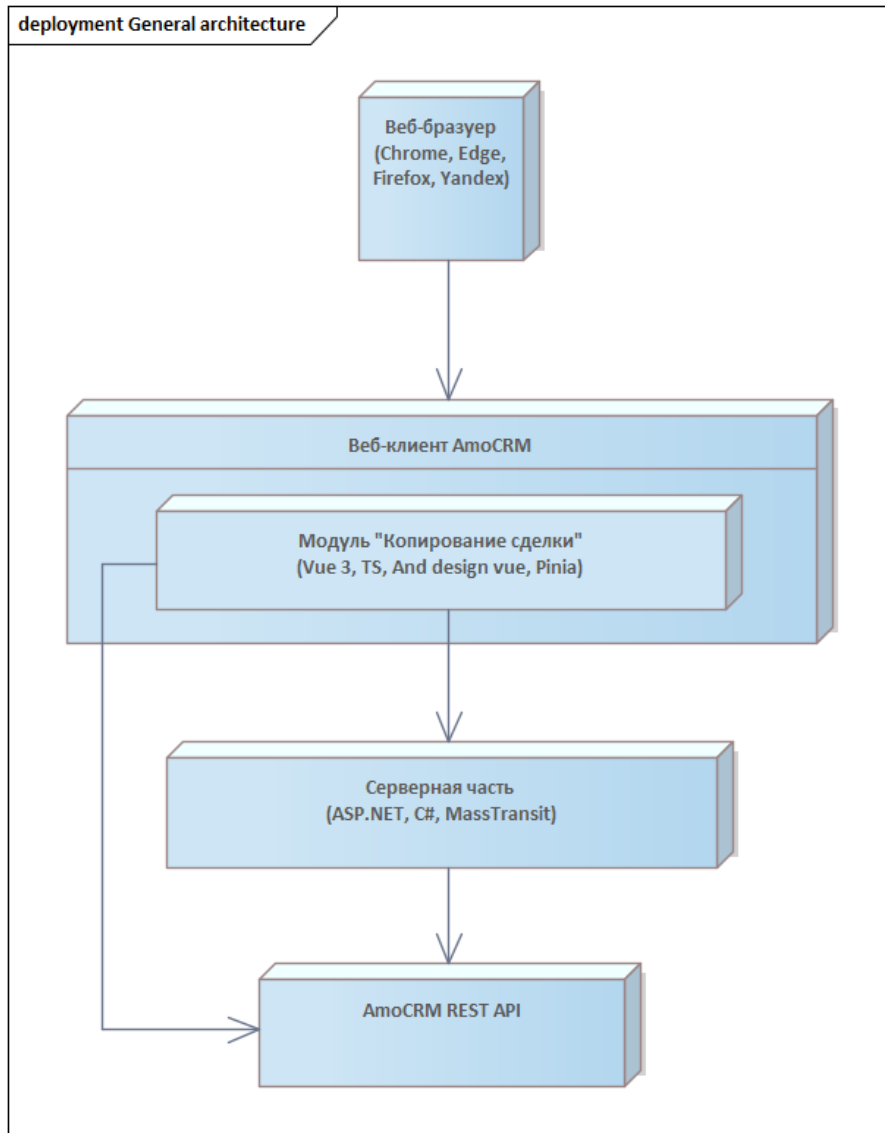
4.1. Общая архитектура

Программный модуль «Копирование сделки» реализован по клиент-серверной архитектуре и состоит из двух основных частей: клиентской и серверной. Клиентская часть представляет собой веб-интерфейс, интегрируемый в систему amoCRM, и обеспечивает взаимодействие пользователя с функциональностью модуля. Серверная часть функционирует как отдельный backend-сервис, реализующий бизнес-логику копирования сделок, обработку команд и взаимодействие с API amoCRM.

Взаимодействие между клиентской и серверной частью осуществляется по протоколу HTTPS с использованием REST API. Клиентская часть инициирует запросы, а серверная выполняет обработку данных, вызывает методы API amoCRM и возвращает результаты для отображения пользователю.

Взаимодействие с amoCRM осуществляется по стандартной схеме интеграции, при этом модуль использует следующие подходы:

- Обмен данными осуществляется по протоколу HTTPS с использованием механизмов авторизации, встроенных в amoCRM для интеграций.
- Для получения информации о сделках, компаниях, контактах и пользовательских полях используются REST API-endpoint-ы amoCRM.



4.2. Клиентская часть

Клиентская часть — это программный модуль, полностью исполняющийся в браузере пользователя в составе веб-интерфейса amoCRM. Она отвечает за визуализацию данных, обработку пользовательских действий и координацию работы с бизнес-логикой и данными, получаемыми через API amoCRM.

Архитектура клиентской части реализована в виде многоуровневой модульной структуры, разделённой на логические слои, что обеспечивает удобство сопровождения, расширяемость и тестируемость решения.

Основные уровни архитектуры:

1. Web (Сборка приложения) — отвечает за инициализацию модуля, подключение глобальной конфигурации. Это точка входа, откуда запускается весь жизненный цикл приложения.
2. Presentation (UI) — реализует пользовательский интерфейс и визуализацию данных, взаимодействуя с бизнес-логикой.
3. App (Бизнес-логика) — слой, содержащий сценарии использования, управление состоянием, обработку пользовательских действий и координацию между Presentation и Data Access.
4. Data Access (Работа с данными) — слой, отвечающий за взаимодействие с внешними источниками данных (в данном случае API amoCRM). Здесь формируются запросы, обрабатываются ответы, выполняется маппинг DTO в модели.
5. Infrastructure (Общие ресурсы) — библиотека общих типов, утилит, констант, валидаторов, которые используются во всех остальных слоях.

Основные инструменты и библиотеки, применяемые в разработке, включают:

- Фреймворк: Vue 3 (Composition API) — обеспечивает реактивный UI, компонентную архитектуру и оптимальную производительность.
- Язык: TypeScript — повышает надёжность кода и облегчает поддержку.
- UI-библиотека: Ant Design Vue — набор готовых визуальных компонентов с адаптацией под дизайн amoCRM.
- Управление состоянием: Pinia — централизованное хранилище состояния приложения.
- Валидация данных: zod — для проверки входных параметров и данных, получаемых с сервера.
- HTTP-клиент: axios — для выполнения запросов к API amoCRM.

4.3. Серверная часть

Серверная часть программного модуля «Копирование сделки» реализована в виде монолитного приложения и предназначена для выполнения бизнес-логики, связанной с обработкой операций копирования сделок и интеграцией с системой AmoCRM. Архитектура приложения

построена по многоуровневому принципу, что обеспечивает структурированность кода, удобство сопровождения и расширения функциональности.

В составе серверной части выделяются следующие уровни:

- Операционный слой, включающий контроллеры и конфигурацию приложения, а также реализующий REST-endpoint для копирования сделки и отдельный endpoint для приёма входящих вебхуков от цифровых воронок amoCRM;
- Слой Application, в котором сосредоточена бизнес-логика модуля, реализованная с помощью обработчиков команд и запросов (CQRS Handlers);
- Слой инфраструктуры, обеспечивающий взаимодействие с внешними системами, включая абстрактный сервис для доступа к API amoCRM.

Взаимодействие между клиентской частью и сервером осуществляется по протоколу HTTPS через REST API. Для асинхронной обработки команд внутри приложения используется внутренняя очередь заданий, функционирующая в памяти и построенная на основе пакета MassTransit. Собственная система управления базами данных и механизмы кэширования в серверной части отсутствуют, так как все необходимые данные извлекаются и обрабатываются напрямую из AmoCRM. Такой подход обеспечивает упрощённую архитектуру, снижение эксплуатационных затрат и надёжную интеграцию с внешней CRM-системой.

4.4. Хранение данных

В архитектуре постоянное серверное хранение данных отсутствует, так как модуль получает всю необходимую информацию напрямую из API amoCRM.

Долговременное хранение данных, включая настройки цифровых воронок (Digital pipelines), полностью обеспечивается на стороне amoCRM. При сохранении настроек все параметры отчёта передаются в amoCRM и записываются в рамках механизма интеграции, что исключает необходимость самостоятельного управления их жизненным циклом со стороны модуля. Таким образом, модуль выступает только как потребитель, а полная

ответственность за долговременное хранение и защиту информации возлагается на amoCRM.

5. Развёртывание системы

Развёртывание программного модуля «Копирование сделки» осуществляется в соответствии с его архитектурными особенностями и предполагает установку как клиентской, так и серверной части. Клиентская часть интегрируется непосредственно в интерфейс amoCRM и распространяется в виде оптимизированного архивного пакета, который загружается администратором через стандартные средства управления модулями. Серверная часть развёртывается в контейнерной среде с использованием оркестратора Docker Swarm и обеспечивает выполнение бизнес-логики и интеграцию с API amoCRM.

Подготовка клиентской части к развёртыванию начинается с процесса сборки. Исходный код, разработанный на языке TypeScript с использованием фреймворка Vue 3, компилируется с помощью Node.js версии 20.11.0 и менеджера пакетов npm версии 10.2.4. Для оптимизации и минификации применяется Webpack версии 5.73.0 и выше, а также вспомогательные инструменты (Babel, плагины для работы со стилями и ресурсами). На завершающем этапе сборки используется плагин zip-webpack-plugin, формирующий итоговый ZIP-архив, содержащий JavaScript-код, CSS-стили, шрифты и графические ресурсы.

Развёртывание выполняется в пользовательском интерфейсе amoCRM через встроенный раздел «Мое приложение» в amoMarket. Администратор CRM загружает подготовленный ZIP-архив посредством стандартной формы, после чего модуль автоматически регистрируется системой и становится доступным для процедуры публикации и использования в рамках аккаунта amoCRM в соответствии с установленными правами доступа.

Обновление клиентской части осуществляется аналогично первичной установке: администратор загружает новый архив, собранный из актуальной версии исходного кода. amoCRM заменяет существующие файлы модуля новыми без необходимости ручного удаления или выполнения миграций, что делает процесс обновления предсказуемым, быстрым и безопасным.

Серверная часть программного модуля «Копирование сделки» развёртывается в среде контейнерной оркестрации Docker Swarm, что обеспечивает отказоустойчивость, возможность горизонтального масштабирования и централизованное управление сервисами. Сборка и публикация образов выполняются в рамках пайплайнов GitLab CI/CD, после чего готовые контейнерные образы размещаются в GitLab Container Registry.

Развёртывание осуществляется с использованием Docker stack-файлов, в которых задаются параметры сервисов, сетей и переменных окружения. Оркестратор Docker Swarm автоматически распределяет контейнеры по рабочим узлам кластера, контролирует их состояние и при необходимости выполняет перезапуск. Доступ к серверной части обеспечивается через прокси-сервер, настроенный на маршрутизацию запросов к соответствующему сервису внутри кластера.

Такой подход позволяет упростить процессы доставки обновлений, обеспечивает предсказуемую среду исполнения и гарантирует, что каждая версия модуля, опубликованная в Container Registry, может быть воспроизведена и развернута в производственной инфраструктуре.