Экзо Платформа

Описание функциональных характеристик программного обеспечения и информация, необходимая для его установки и эксплуатации

Версия программного обеспечения: 1.0.0

Правообладатель: Индивидуальный предприниматель Архипов Дмитрий Викторович
ИНН 781431389513, ОГРНИП 306784706200235

2026

1. Общие сведения

1.1. Наименование и назначение

Наименование: **«Экзо Платформа»**.

Программное обеспечение предназначено для создания и эксплуатации прикладных информационных систем предприятия без разработки серверного кода. Прикладное решение не программируется, а **моделируется**: структура данных описывается схемами, бизнес-логика — исполняемыми процессами, обмен с внешними системами — описаниями их программных интерфейсов, права доступа — ролями и правилами. По этим моделям платформа самостоятельно создаёт хранилище, программные интерфейсы и пользовательский интерфейс и обеспечивает их работу.

По решаемым задачам ближайшим функциональным аналогом является платформа «1С:Предприятие 8»: и там, и здесь предметом поставки является среда, в которой заказчик или партнёр создаёт собственное прикладное решение, а не готовое отраслевое приложение.

Отличие в устройстве: платформа построена на открытых отраслевых стандартах — JSON Schema для описания данных, BPMN 2.0 для процессов, OpenAPI, SOAP, OData, GraphQL и СМЭВ для интеграций, OAuth 2.0 и OpenID Connect для доступа. Прикладное решение описывается в терминах этих стандартов, а не собственного языка платформы.

1.2. Правообладатель и распространение

Исключительное право на программное обеспечение принадлежит правообладателю — индивидуальному предпринимателю Архипову Дмитрию Викторовичу. Реквизиты правообладателя приведены на титульном листе настоящего документа.

Программное обеспечение свободно реализуется на территории Российской Федерации: права использования предоставляет правообладатель по лицензионному договору. Формы лицензии, единицы лицензирования и порядок определения стоимости изложены в документе «Тарифная политика», опубликованном на сайте правообладателя.

1.3. Состав программного обеспечения

Платформа поставляется как набор модулей. Заказчик приобретает нужный ему набор; модули работают как совместно, так и по отдельности, и по мере развития платформы в её состав включаются новые модули.

Модуль	Назначение
Registry	Управление данными: подключение к системам хранения, описание моделей данных, ведение записей, связи между каталогами, программные интерфейсы к данным
Relay	Интеграция и медиация программных интерфейсов: приведение разнородных внешних систем к единому формату взаимодействия, маршрутизация, преобразование, гарантированная доставка
Workflow	Исполнение и мониторинг процессов: выполнение процессов в нотации BPMN 2.0, служебные и пользовательские задачи, события, инциденты
Identity	Единый вход и управление доступом: аутентификация пользователей и приложений, выдача и проверка токенов доступа, пользователи, роли, сеансы
Графический интерфейс	Единый веб-интерфейс платформы, через который ведётся моделирование и повседневная работа со всеми модулями

Модули взаимодействуют через программные интерфейсы: Workflow обращается к Registry за данными и к Relay за вызовами внешних систем, все модули проверяют токены доступа, выданные Identity. Каждый модуль при этом самодостаточен и может использоваться

отдельно от остальных.

1.4. Область применения

Платформа применяется для построения прикладных систем предприятия, в том числе:

- автоматизации бизнес-процессов и документооборота;
- внутренних корпоративных инструментов и порталов самообслуживания;
- систем массового обслуживания и оказания услуг, включая государственные;
- отраслевых решений в промышленности, строительстве, логистике, розничной торговле, финансовом и банковском секторе;
- управления персоналом, продажами, финансами;
- модернизации унаследованных информационных систем — за счёт подключения их данных и интерфейсов без переписывания;
- интеграционных решений, объединяющих разнородные системы предприятия.

2. Функциональные характеристики

2.1. Платформа в целом

- **Единая модель прикладного решения.** Данные, процессы, интеграции и права доступа описываются декларативно и хранятся как версионируемые описания, а не как исходный код.
- **Автоматическое порождение интерфейсов.** По описанию данных создаются программные интерфейсы REST и GraphQL и экранные формы; отдельная разработка серверной части и клиентской части для типовых операций не требуется.
- **Единый вход.** Пользователь аутентифицируется один раз, его роли и права передаются во все модули в составе токена доступа.
- **Открытые стандарты** на всех границах: JSON Schema, OpenAPI, BPMN 2.0, DMN, FEEL, OAuth 2.0, OpenID Connect, RSQL, GeoJSON, OGC API.
- **Контейнерная поставка.** Все модули поставляются образами контейнеров и разворачиваются одинаково, независимо от операционной системы узла.

2.2. Модуль Registry — управление данными

Модуль предоставляет единый интерфейс к разнородным системам хранения данных.

Поддерживаемые системы хранения: PostgreSQL (включая российские сборки), MySQL и MariaDB, Oracle Database, Microsoft SQL Server, SQLite, MongoDB, Couchbase. Работа со всеми ведётся через один и тот же интерфейс; различия систем хранения скрыты драйверами модуля.

Основные возможности:

- **Описание моделей данных** на языке JSON Schema. По схеме модуль сам создаёт таблицу или коллекцию в выбранной системе хранения, включая типы столбцов, обязательность и ограничения.
- **Каталоги** — единица управления данными: каталог связывает схему с подключением к конкретной базе данных.
- **Подключение существующих баз данных.** Модуль умеет читать структуру уже существующих таблиц и коллекций и заводить по ним каталоги без изменения структуры — это позволяет включать в контур унаследованные системы.
- **Ведение записей**: создание, чтение, изменение, удаление, пакетные операции, постраничная выдача с навигацией по стандарту RFC 8288.
- **Связи между каталогами**, в том числе между каталогами, расположенными в разных системах хранения; связанные данные раскрываются в одном запросе.
- **Виртуальные каталоги** — каталоги-представления, собирающие или проецирующие данные других каталогов, в том числе каталогов удалённых экземпляров платформы.
- **Фильтрация на языке RSQL** — единый синтаксис запросов для всех систем хранения,

включая сравнение, вхождение, сопоставление с образцом, полнотекстовый и геопространственный поиск.
- **Полнотекстовый поиск** средствами самой системы хранения, где она это поддерживает.
- **Геоданные**: хранение геометрий в собственных пространственных типах систем хранения, публикация каталогов в формате GeoJSON и по стандарту OGC API — Features.
- **Файловые вложения**: запись может ссылаться на файл в хранилище платформы; проверяется тип и размер файла, файл привязывается к записи.
- **Форматно-логический контроль** входных данных по схеме до записи в хранилище.
- **Разграничение доступа** к данным на уровне полей и строк.
- **Программные интерфейсы** REST и GraphQL, порождаемые для каждого каталога автоматически, с описанием в формате OpenAPI.

2.3. Модуль Relay — интеграция и медиация программных интерфейсов

Модуль приводит разнородные внешние системы к единому формату взаимодействия и выступает как посредник между потребителями и поставщиками программных интерфейсов.

Поддерживаемые стандарты: SOAP 1.2, OpenAPI 3.x, OData 3.x и 4.x, GraphQL, gRPC (proto3), СМЭВ 3.x — в роли как потребителя, так и поставщика.

Основные возможности:

- **Нормализация интерфейсов.** Потребитель работает с единым форматом независимо от протокола внешней системы.
- **Построение описаний по документам поставщика**: WSDL, XSD, $metadata, описания OpenAPI преобразуются в схемы данных и в описание операций подключения.
- **Маршрутизация и проксирование** запросов, включая правила преобразования данных при передаче — сопоставление полей, преобразование значений, изменение структуры.
- **Синхронный и асинхронный обмен** с гарантированной доставкой сообщений, очередями и повторной отправкой.
- **Форматно-логический контроль** запросов и ответов по схемам.
- **Разграничение доступа** к отдельным запросам и к отдельным полям в них.
- **Журналирование** запросов и протокольного обмена для аудита и разбора инцидентов.
- **Эмуляция внешних систем** и сценарии автоматического тестирования интеграций — подключение можно проверить без доступа к боевой системе поставщика.
- **Электронная подпись по ГОСТ Р 34.10-2012** для взаимодействия со СМЭВ, включая вычисление хэш-значений по ГОСТ через криптопровайдер, установленный у заказчика.

2.4. Модуль Workflow — исполнение и мониторинг процессов

Модуль исполняет процессы в нотации **BPMN 2.0** и управляет их жизненным циклом: описание, экземпляр, переменные, история выполнения.

Основные возможности:

- **Развёртывание описаний процессов** с версионированием: каждое развёртывание создаёт новую версию, ранее запущенные экземпляры продолжают работать на своей.
- **Исполнение процесса**: события начала, промежуточные и граничные события, шлюзы (исключающий, параллельный, включающий, по событию), подпроцессы, вызов процесса как переиспользуемого элемента, множественные экземпляры.
- **Служебные задачи через коннекторы** к внешним системам: REST, GraphQL, веб-перехватчики; коннекторы вызываются фоновыми заданиями с повторами при отказе.
- **Пользовательские задачи** с экранными формами: форма описывается схемой JSON Schema, вводимые данные проверяются по ней, поддерживаются черновики.
- **События процесса**: корреляция сообщений «один к одному», широковещательная рассылка сигналов «один ко многим», таймеры, в том числе повторяющиеся.
- **Выражения и правила**: язык FEEL для условий и преобразований, таблицы решений DMN.
- **Инциденты** — фиксация ситуаций, которые процесс не может обработать сам (например, исчерпание повторов при вызове внешней системы), их разбор и повторный

запуск шага после устранения причины.
- **Наблюдение за исполнением**: состояние экземпляров, значения переменных, история выполнения, перечень ожидающих задач.
- **Графическое моделирование и диагностика**: модели готовятся в стандартных графических редакторах BPMN и публикуются в модуль прямо из редактора; ход исполнения разбирается через инциденты, историю выполнения и просмотр переменных экземпляра.
- **Совместимость с Camunda 8**: принимаются модели в формате Camunda 8 / Zeebe с их расширениями, языком выражений FEEL и таблицами решений DMN, и воспроизводится семантика их исполнения — модели, подготовленные для Camunda, работают без переделки.

Извлечение, преобразование и загрузка данных (ETL). Платформа выполняет перенос данных между системами средствами трёх модулей, без отдельного инструмента: извлечение выполняет Relay — он обращается к внешней системе по её протоколу и приводит ответ к единому формату; преобразование описывается процессом Workflow, где правила заданы выражениями FEEL и таблицами решений, а длительные переносы разбиты на шаги с повторами при отказе; загрузку выполняет Registry, записывая данные в целевое хранилище по описывающей его схеме. Расписание переноса задаётся таймерами процесса, а ход и результат каждого запуска видны в истории экземпляров и в журнале обмена.

2.5. Модуль Identity — единый вход и управление доступом

Модуль является центром аутентификации и авторизации платформы.

Основные возможности:

- **Аутентификация пользователей** по паролю, с точками расширения для корпоративного каталога LDAP и внешних провайдеров OAuth.
- **Сервер авторизации OAuth 2.0**: поток авторизационного кода с PKCE для приложений с участием пользователя, поток учётных данных клиента для межсервисного взаимодействия.
- **Провайдер OpenID Connect 1.0**: идентификационные токены, конечная точка сведений о пользователе, метаданные обнаружения, публикация ключей проверки подписи (JWKS).
- **Токены доступа в формате JWT** с подписью RS256: остальные модули проверяют их локально по открытому ключу либо обращаются к конечной точке интроспекции.
- **Отзыв и интроспекция** токенов.
- **Управление пользователями и ролями**, назначение ролей, программный интерфейс для средств администрирования.
- **Управление приложениями-клиентами**: регистрация, адреса перенаправления, области доступа, секреты.
- **Управление сеансами**: перечень активных входов пользователя и принудительное завершение сеанса.
- **Настраиваемый состав утверждений токена**: для каждого приложения правила на языке FEEL определяют, какие сведения о пользователе и его правах попадут в токен.
- **Ограничение области действия токена** адресатами по RFC 8707.

2.6. Графический интерфейс платформы

Единый веб-интерфейс, через который ведётся и моделирование прикладного решения, и повседневная работа с ним:

- визуальный редактор моделей данных и связей между ними;
- ведение записей: списки с фильтрацией и постраничной навигацией, экранные формы, работа с файлами, отображение геоданных на карте;
- конструктор экранных форм;
- работа с процессами: публикация, запуск, наблюдение за экземплярами, выполнение пользовательских задач, разбор инцидентов;
- настройка интеграций и проверка подключений;
- администрирование доступа: пользователи, роли, приложения, сеансы;
- редакторы схем, выражений и исходных текстов настроек со встроенной проверкой.

3. Системные требования

3.1. Аппаратные требования

Минимальная конфигурация — все компоненты, включая систему хранения, на одном узле; ознакомительная установка и небольшие прикладные решения: 2 ядра, 4 ГБ оперативной памяти, 40 ГБ дискового пространства.

Рекомендуемая конфигурация — промышленная эксплуатация, система хранения на отдельном узле: узел приложений — 4 ядра, 8 ГБ оперативной памяти, 60 ГБ дискового пространства; узел системы хранения — от 4 ядер и 8 ГБ оперативной памяти, дисковое пространство по объёму прикладных данных.

Потребление растёт от числа одновременно работающих пользователей и интенсивности фоновых заданий — вызовов внешних систем и срабатываний таймеров процессов; дисковое пространство — от объёма прикладных данных, файловых вложений и глубины хранения журналов.

Требования к каналу связи определяются объёмом интеграционного обмена и не являются критичными для работы самой платформы.

3.2. Программные требования

Компонент	Требование
Операционная система узла	Любой дистрибутив GNU/Linux с поддержкой контейнеров, в том числе российские: Astra Linux, РЕД ОС, ALT Linux
Среда исполнения контейнеров	Docker 24 и выше либо совместимая
Система хранения платформы	Отдельная установка не требуется: модули работают на встроенной файловой системе хранения, входящей в состав образов. PostgreSQL 14 и выше, включая российские сборки (Postgres Pro), подключается по решению заказчика
Дополнительно для модуля Registry	По потребности заказчика — любая из поддерживаемых систем хранения прикладных данных (см. п. 2.2)
Криптопровайдер	Требуется только при использовании электронной подписи по ГОСТ в модуле Relay
Клиентское рабочее место	Веб-браузер, поддерживающий современные стандарты, включая российские сборки на движке Chromium

Среда исполнения самих модулей (интерпретатор и библиотеки) входит в состав образов контейнеров и отдельной установки не требует.

Внешняя система управления базами данных **не является обязательной зависимостью**. Установка запускается и работает без неё: каждый модуль хранит свои данные в файле в примонтированном каталоге и создаёт структуру хранения при первом запуске.

PostgreSQL рекомендуется для промышленной эксплуатации в двух случаях: при высокой нагрузке — когда с платформой одновременно работает значительное число пользователей и фоновых заданий, — и при требованиях к резервированию и кластеризации: репликации, аварийном переключении, работе нескольких экземпляров модуля с одними и теми же данными. Встроенная файловая система хранения таких возможностей не предоставляет. Выбор задаётся одной переменной окружения; перенос ранее накопленных данных при переходе выполняется штатными средствами систем хранения.

3.3. Совместимость с отечественным программным обеспечением

Платформа поставляется образами контейнеров и не зависит от конкретного дистрибутива операционной системы: требуется лишь наличие среды исполнения контейнеров, входящей в состав всех перечисленных российских операционных систем.

Внешняя система управления базами данных не требуется, а там, где заказчик её применяет, поддерживается PostgreSQL — в том числе российские сборки, включённые в реестр отечественного программного обеспечения (Postgres Pro).

Операционные системы. Платформа работает под управлением любой операционной системы, в которой доступна среда исполнения контейнеров Linux: российских дистрибутивов GNU/Linux — Astra Linux, РЕД ОС, ALT Linux, — прочих дистрибутивов GNU/Linux, а также macOS и Windows со средой, запускающей контейнеры Linux. Требование к операционной системе исчерпывается наличием такой среды: сам программный код и его зависимости входят в состав образов и от дистрибутива не зависят.

Системы хранения. Модули работают на встроенной файловой системе хранения, не требующей установки, либо на PostgreSQL 14 и выше — включая российские сборки, входящие в реестр отечественного программного обеспечения (Postgres Pro). Выбор задаётся строкой подключения и не влияет на состав поставки.

Прикладные данные, которыми управляет модуль Registry, дополнительно могут храниться в любой из систем, перечисленных в п. 2.2.

4. Установка

4.1. Порядок установки

Установка выполняется администратором заказчика и состоит из следующих шагов.

1. **Подготовка узла.** Установить среду исполнения контейнеров и средство запуска групп контейнеров.
2. **Получение дистрибутива.** Загрузить образы контейнеров модулей — из реестра образов правообладателя по выданным учётным данным либо из переданного архива, если узел работает в закрытом контуре.
3. **Подготовка систем хранения** — шаг выполняется, если используется внешняя СУБД. Создать базы данных для модулей и учётные записи к ним: каждому модулю требуется собственная база данных. На встроенной системе хранения шаг не нужен — она создаётся при первом запуске модуля.
4. **Настройка параметров.** Задать переменные окружения каждого модуля: строки подключения к базам данных, адреса взаимодействующих модулей, параметры аутентификации, ключи подписи. Полный перечень переменных с назначением и значениями по умолчанию приведён в руководстве администратора.
5. **Инициализация баз данных.** Выполнить миграции схемы — при первом запуске модуль создаёт необходимые таблицы самостоятельно.
6. **Первичная настройка доступа.** В модуле Identity создать администратора, зарегистрировать приложения остальных модулей и графического интерфейса, задать роли.
7. **Запуск.** Запустить контейнеры модулей и графического интерфейса.

Порядок запуска: сначала внешние системы хранения, если они используются, затем Identity, затем остальные модули, затем графический интерфейс. Модули, не получившие ответа от Identity, отвечают на
запросы отказом в доступе, поэтому его недоступность не приводит к потере данных.

4.2. Проверка работоспособности после установки

Признаком успешной установки является следующее.

- Каждый модуль отвечает на запрос к конечной точке своего описания программного

интерфейса и возвращает документ OpenAPI.
- Модуль Identity выдаёт токен доступа по потоку учётных данных клиента и публикует метаданные обнаружения и ключи проверки подписи.
- Модуль Registry принимает описание схемы, создаёт по ней каталог и выполняет запись и чтение записи.
- Модуль Workflow принимает описание процесса, публикует его и запускает экземпляр.
- Модуль Relay принимает описание внешней системы и выполняет проверочный вызов встроенного эмулятора.
- Графический интерфейс открывается в браузере, вход выполняется учётной записью, созданной при первичной настройке.

Пошаговые проверочные запросы с ожидаемыми ответами приведены в руководстве администратора.

5. Эксплуатация

5.1. Типовые сценарии работы

Создание прикладного решения. Описывается модель данных и создаются каталоги в модуле Registry — после этого данные доступны через порождённые программные интерфейсы и через экранные формы. Описывается процесс в нотации BPMN и публикуется в модуле Workflow; шаги процесса обращаются к данным через Registry, к внешним системам — через Relay, а действия, требующие участия человека, попадают пользователям как задачи с формами. Права доступа задаются ролями в модуле Identity и действуют одинаково во всех модулях и интерфейсах.

Подключение внешней системы. В модуль Relay загружается описание интерфейса поставщика (WSDL, XSD, OpenAPI, $metadata) или указывается адрес, по которому оно доступно; модуль строит схемы и перечень операций. Подключение проверяется на эмуляторе, после чего используется процессами и прикладными решениями.

Включение унаследованной системы. Существующая база данных подключается к модулю Registry как источник; структура читается автоматически, данные становятся доступны через единые интерфейсы платформы без изменения самой базы и без остановки системы, которая её использует.

Повседневная работа пользователя. Пользователь входит через единый вход, работает со списками и формами, выполняет назначенные задачи процессов, наблюдает ход процессов, к которым имеет доступ.

5.2. Администрирование

Перечень регулярных операций администратора; порядок выполнения — в руководстве администратора.

- Управление пользователями, ролями и приложениями, завершение сеансов.
- Подключение и отключение источников данных, изменение схем и каталогов.
- Публикация новых версий процессов и форм, разбор инцидентов.
- Наблюдение за состоянием модулей и просмотр журналов.
- Резервное копирование баз данных и файлового хранилища, восстановление из копии.
- Обновление версий модулей.

6. Сведения о правах

Исключительное право на все модули платформы принадлежит правообладателю. Каждый модуль зарегистрирован как программа для ЭВМ:

Модуль	Свидетельство	Дата
Relay	№ 2026684891	18.08.2026
Registry	№ 2026689726	29.09.2026
Workflow	№ 2026689732	29.09.2026
Identity	№ 2026688896	22.09.2026

Сведений, составляющих государственную тайну, программное обеспечение не содержит. Принудительное обновление и управление программным обеспечением из-за пределов Российской Федерации не осуществляется: установка обновлений выполняется администратором заказчика по его решению.