

# Платформа автоматизации с ИИ-агентами «Nodul»

## Описание функциональных характеристик

---

### АННОТАЦИЯ

Документ содержит сведения о функциональных характеристиках программного обеспечения «Платформа автоматизации с ИИ-агентами Nodul» (далее — «ПО», «Nodul»).

Документ предназначен для пользователей программного обеспечения и сотрудников организации-разработчика.

---

### СОДЕРЖАНИЕ

1. Общие сведения
    - 1.1. Наименование программы
    - 1.2. Основные сведения
    - 1.3. Назначение программы
    - 1.4. Особенности применения
  2. Перечень реализуемых функций
  3. Описание характеристик
    - 3.1. Общие характеристики
    - 3.2. Функциональные характеристики
    - 3.3. Прочие характеристики качества программного обеспечения
- 

## 1. ОБЩИЕ СВЕДЕНИЯ

### 1.1. Наименование программы

Полное наименование программы: «Платформа автоматизации с ИИ-агентами Nodul».

В рамках настоящего документа употребляются также обозначения «Nodul» и «ПО».

### 1.2. Основные сведения

«Платформа автоматизации с ИИ-агентами Nodul» — российское программное обеспечение класса low-code / iPaaS (Integration Platform as a Service), предназначенное для автоматизации бизнес-процессов на визуальном канвасе с опорой на искусственный интеллект.

ИИ в продукте — не вспомогательная опция, а опорная линия: в платформу встроены ИИ-агенты на основе больших языковых моделей (LLM), технология RAG (поиск с дополнением контекста по корпоративным данным), поддержка протокола MCP (Model Context Protocol) и встроенный ИИ-ассистент, помогающий пользователю собирать и отлаживать сценарии в режиме диалога.

Серверная инфраструктура размещается на территории Российской Федерации.

## 1.3. Назначение программы

### 1.3.1. Функциональное назначение

ПО предназначено для:

- построения и эксплуатации ИИ-агентов и чат-ботов, способных автономно выполнять задачи в рамках заданных инструкций и инструментов;
- автоматизации бизнес-процессов и интеграции корпоративных информационных систем без написания кода или с минимальным кодом;
- обработки и трансформации данных между источниками: извлечение, обогащение, маршрутизация, синхронизация записей между сервисами;
- предоставления специалистам средств визуальной разработки сценариев автоматизации.

### 1.3.2. Эксплуатационное назначение

ПО реализовано в виде распределённого веб-приложения. В варианте поставки on-premise серверная часть разворачивается в инфраструктуре заказчика средствами Kubernetes (поставляются Helm-чарты). Пользователи работают с ПО через современный веб-браузер; собранные сценарии и ИИ-агенты исполняются на стороне платформы по поступающим триггерам или по расписанию.

## 1.4. Особенности применения

ПО применяется в организациях для автоматизации операционных и интеграционных процессов. Пользователями ПО являются бизнес-аналитики, инженеры автоматизации, разработчики и администраторы корпоративных информационных систем. Сценарии могут работать круглосуточно без участия человека по событиям из подключённых сервисов, по расписанию или по входящим HTTP-запросам.

---

## 2. ПЕРЕЧЕНЬ РЕАЛИЗУЕМЫХ ФУНКЦИЙ

ПО обеспечивает выполнение следующих основных функций.

### Функции искусственного интеллекта (опорная линия продукта):

- построение ИИ-агентов на базе больших языковых моделей с использованием инструкций (системного сообщения), вызова инструментов и контекстной памяти; поддержка базового, модульного и мультиагентного режимов работы;
- предоставление специальных инструментов рассуждения (Think и Plan) для повышения точности и предсказуемости агента;
- подключение более 300 ИИ-моделей на едином балансе кредитов, в том числе российских (YandexGPT, GigaChat) и зарубежных (GPT-4o, Claude, Gemini, Llama, Mistral, DeepSeek, Stable Diffusion и других); возможность использования собственных API-ключей для подключения произвольного LLM-провайдера (Custom LLM Connection);
- технология RAG: загрузка и индексация корпоративных документов (узел RAG Search и хранилище AI Data Storage) и подключение результата поиска к ИИ-агенту;
- поддержка протокола MCP (Model Context Protocol): подключение внешних MCP-серверов в качестве источников инструментов для ИИ-агентов и публикация сценариев Nodul как MCP-инструментов для внешних ИИ-клиентов;
- встроенный ИИ-ассистент сборки и отладки сценариев (AI scenario helper), а также узел AI node для генерации текста и узел AI GPT Router для маршрутизации запросов между моделями;

- генерация и редактирование пользовательского кода в узлах сценария с использованием встроенных ИИ-инструментов.

### **Визуальный конструктор и исполнение сценариев:**

- сборка сценариев на канвасе в виде узлов и связей между ними; поддержка ветвлений, итераций, ожиданий, маршрутизации, резервных связей;
- запуск сценария по триггерам: входящий HTTP-запрос (вебхук), расписание, входящее сообщение электронной почты (Mailhook), однократный запуск (Run once), а также через узел «Вход» в составных сценариях;
- базовые узлы потока: HTTP-запрос, JSON-разбор, итератор, ожидание, ответ вебхуку, установка и чтение переменных и глобальных переменных;
- исполнение пользовательского кода в узлах JavaScript, Node.js и Bun; узел Headless-браузера для автоматизации работы с веб-страницами; узлы для работы с файлами и FTP;
- разделение веток Production и Development в сценарии; публикация и история версий сценариев; импорт и экспорт; копирование сценариев и узлов;
- журнал истории выполнения с входами, выходами, длительностью и ошибками каждого узла; повторный запуск сценария с произвольного шага;
- штатные средства обработки ошибок: игнорирование ошибок, автоматический перезапуск узла с настраиваемой задержкой, перезапуск при некорректном ответе (опрос), резервные связи для альтернативного маршрута выполнения;
- управление производительностью: настройка лимита параллельных выполнений, расширенные вычислительные ресурсы (Engine Tier 1).

### **Интеграции и работа с данными:**

- готовые узлы для распространённых сервисов (CRM, мессенджеры, электронная почта, средства продуктивности, базы данных, файловые хранилища) с автоматической OAuth-авторизацией и единым реестром учётных данных (авторизаций);
- собственные пользовательские базы данных (коллекции) с операциями выборки, изменения и поиска; AI Data Storage для подготовки данных под RAG;
- работа с произвольными внешними API через узел HTTP-запроса; собственные пользовательские узлы (custom nodes) с пользовательскими параметрами;
- выделенный исходящий IP-адрес платформы для добавления в белый список внешних сервисов.

### **Командная работа и администрирование:**

- управление рабочим пространством: пользователи с разграничением прав доступа, папки сценариев;
  - управление подпиской и тарифами; статистика использования и контроль кредитного баланса;
  - настройки безопасности и пользовательский профиль.
-

## 3. ОПИСАНИЕ ХАРАКТЕРИСТИК

### 3.1. Общие характеристики

#### 3.1.1. Состав и структура

ПО реализовано как распределённое веб-приложение и состоит из следующих основных компонентов:

- **веб-интерфейс** (одностраничное приложение), исполняемый в браузере пользователя; обеспечивает работу с визуальным конструктором сценариев, отладчиком, ИИ-ассистентом, настройками рабочего пространства;
- **API и серверная часть** — приём запросов от веб-интерфейса, управление учётными записями, рабочими пространствами, сценариями и тарификацией;
- **движок исполнения сценариев** — оркестрация запусков, выполнение узлов сценария в изолированных средах, передача данных между узлами, ведение журнала выполнения;
- **сервисы интеграций** — подключение к внешним приложениям и провайдерам ИИ-моделей;
- **подсистема хранения** — реляционная и документная СУБД, объектное хранилище (см. п. 3.2.5).

Поставка для развёртывания на стороне заказчика осуществляется в виде Helm-чартов для Kubernetes; каждый компонент представлен отдельным контейнерным образом и масштабируется независимо.

#### 3.1.2. Программное обеспечение, необходимое для функционирования

Для функционирования серверной части ПО заказчику необходимо обеспечить:

- кластер Kubernetes актуальной поддерживаемой версии;
- пакетный менеджер Helm 3.x для установки и обновления чартов поставки;
- ingress-контроллер и средства управления TLS-сертификатами, обеспечивающие доступ к веб-интерфейсу по HTTPS;
- СУБД MongoDB и MySQL — поставляются в составе инсталляции либо могут использоваться внешние managed-инстансы;
- S3-совместимое объектное хранилище для хранения файлов и крупных артефактов выполнений (например, Yandex Object Storage, MinIO или иное совместимое хранилище);
- исходящий доступ кластера в сеть Интернет: текущая версия ПО требует возможности обращаться к внешним сервисам — провайдерам ИИ-моделей и внешним API, используемым в сценариях пользователя. Эксплуатация в полностью изолированном контуре без доступа в Интернет в текущей версии не поддерживается.

Для рабочего места пользователя необходимо общее программное обеспечение:

- актуальная версия одного из распространённых веб-браузеров (Chromium-совместимые: Google Chrome, Microsoft Edge, Yandex Browser, Opera; либо Mozilla Firefox).

#### 3.1.3. Технические средства, необходимые для функционирования

Минимальные требования к ресурсам Kubernetes-кластера для базовой инсталляции ПО:

- ЦПУ — не менее 2 ядер;
- ОЗУ — не менее 4 ГБ;
- дисковое пространство — не менее 50 ГБ;
- архитектура x86-64.

Приведённые требования являются минимально допустимыми и предназначены для пилотной инсталляции и инсталляции с невысокой нагрузкой. Для промышленной эксплуатации с большим количеством сценариев и параллельных запусков ресурсы должны быть увеличены пропорционально нагрузке; благодаря модульной архитектуре масштабирование выполняется штатными средствами Kubernetes.

Минимальные требования к рабочему месту пользователя:

- персональный компьютер с любой операционной системой, на которой работает поддерживаемый браузер;
- разрешение экрана не менее 1366×768 пикселей; рекомендуется не ниже Full HD (1920×1080);
- стабильное сетевое подключение к серверной части ПО.

#### **3.1.4. Средства разработки**

ПО разработано с использованием распространённых языков программирования общего назначения и открытых фреймворков. Серверная и клиентская части реализованы на TypeScript / JavaScript; пользовательский код, написанный в узлах сценария, выполняется средствами Node.js и Bun. Сборка, тестирование и развёртывание ПО выполняются средствами с открытым исходным кодом.

### **3.2. Функциональные характеристики**

#### **3.2.1. Режим функционирования**

ПО ориентировано на круглосуточный непрерывный режим работы. Активные сценарии запускаются автоматически по поступающим триггерам (HTTP-вебхук, расписание, входящее сообщение электронной почты) либо вручную («Run once»). Допускаются периодические остановки для выполнения регламентных и обновляющих работ; средствами Kubernetes возможно поэтапное обновление компонентов без полной остановки сервиса (rolling update).

#### **3.2.2. Пользователи**

Пользователями ПО являются:

- бизнес-пользователи и аналитики, собирающие сценарии автоматизации;
- разработчики и инженеры автоматизации, расширяющие сценарии собственным кодом и API-интеграциями;
- администраторы рабочих пространств, отвечающие за состав команды и распределение прав доступа.

#### **3.2.3. Количественные характеристики**

Количество пользователей в рабочем пространстве, количество сценариев и количество одновременных запусков определяются параметрами выбранного тарифа (для облачной эксплуатации) либо параметрами развёрнутого Kubernetes-кластера и настройками лимита параллельных выполнений (для self-hosted-инсталляции). Конкретные тарифные лимиты приведены в эксплуатационной документации; для self-hosted-инсталляции лимиты настраиваются администратором заказчика.

#### **3.2.4. Сетевое взаимодействие**

Между рабочим местом пользователя и серверной частью используется протокол HTTPS. Внутри Kubernetes-кластера компоненты ПО взаимодействуют по протоколам HTTP/gRPC и сетевым каналам соответствующих СУБД. ПО выполняет исходящие HTTP/HTTPS-запросы к внешним сервисам — провайдерам ИИ-моделей и интеграциям, заданным пользователем в

сценарии. Для регистрации платформы во внешних сервисах (например, в качестве клиентского источника в белых списках) используется выделенный исходящий IP-адрес.

### **3.2.5. Сбор и хранение данных**

Метаданные сценариев, журналы выполнения, пользовательские коллекции данных и индексы для RAG-поиска хранятся в документной СУБД MongoDB. Системные сущности (учётные записи, рабочие пространства, тарификация) — в реляционной СУБД MySQL. Файлы пользователей, артефакты выполнений и крупные двоичные данные — в S3-совместимом объектном хранилище.

## **3.3. Прочие характеристики качества программного обеспечения**

### **3.3.1. Надёжность**

ПО построено на принципах модульной микросервисной архитектуры. Распределение по контейнерам Kubernetes позволяет использовать штатные средства оркестратора — поэтапное обновление, проверки готовности, автоматический перезапуск упавших экземпляров, горизонтальное масштабирование — для обеспечения отказоустойчивости. На уровне сценария реализованы штатные процедуры восстановления:

- автоматический перезапуск узла сценария при ошибке с настраиваемым числом попыток и интервалом;
- перезапуск узла при некорректном ответе (режим опроса);
- ветвление обработки ошибок («резервные связи») для перехвата отказов и продолжения сценария по альтернативной ветке;
- сохранение полной истории выполнения для последующего разбора и повторного запуска проблемного шага.

### **3.3.2. Расширяемость**

Платформа спроектирована модульно. Расширение функциональности возможно следующими способами:

- собственный пользовательский код в узлах JavaScript, Node.js и Bun без ограничений общего назначения;
- собственные HTTP-интеграции через узел HTTP-запроса с произвольными внешними API;
- собственные пользовательские узлы (custom nodes) с пользовательскими параметрами и логикой;
- подключение внешних MCP-серверов как дополнительных источников инструментов для ИИ-агентов;
- использование собственных API-ключей для подключения произвольных провайдеров LLM.

### **3.3.3. Защищённость**

ПО разрабатывается с применением принципов безопасной разработки. Поддерживаются:

- разграничение прав пользователей в рабочем пространстве;
- хранение учётных данных внешних сервисов в виде отдельных авторизаций (credentials), отделённых от тела сценариев и доступных только в момент исполнения;
- интеграция с механизмами защиты, предоставляемыми Kubernetes и инфраструктурой заказчика (сетевые политики, секреты, RBAC, дисковое шифрование);
- средства контроля поведения ИИ-агентов (guardrails) на уровне инструкций, перечня доступных инструментов и проверки результатов вызовов;

- изоляция исполнения пользовательского кода в отдельных средах исполнения.

#### **3.3.4. Эргономичность**

ПО имеет графический пользовательский интерфейс типа «канвас», в котором сценарий собирается из визуальных блоков (узлов), соединённых линиями связи. Пользователю доступны: встроенные подсказки, поиск по узлам и интеграциям, инспектор данных предыдущих узлов, а также встроенный ИИ-ассистент, способный объяснять, проектировать и отлаживать сценарии в режиме диалога. Язык пользовательского интерфейса — русский (предусмотрена также английская локализация).

#### **3.3.5. Сопровождаемость**

Эксплуатация ПО не требует от конечных пользователей специальных знаний, кроме общих навыков работы с веб-приложениями и понимания собственных бизнес-процессов. Развёртывание и обслуживание серверной части ПО (Helm, Kubernetes, СУБД, объектное хранилище) выполняется силами эксплуатационной службы заказчика в соответствии с эксплуатационной документацией, поставляемой вместе с продуктом.

Сопровождение со стороны разработчика выполняется в рамках регистрации и обработки обращений пользователей. Каналы поддержки:

- форум сообщества Nodul: <https://community.nodul.ru> — основной канал;
- Telegram-сообщество: <https://t.me/nodul> — для оперативных вопросов;
- электронная почта: [help@nodul.ru](mailto:help@nodul.ru) — для случаев недоступности аккаунта или критических инцидентов.

Режим работы службы технической поддержки разработчика: с понедельника по пятницу с 10:00 до 22:00 (центральноевропейское время; по московскому времени — с 11:00 до 23:00). В выходные дни обращения обрабатываются по возможности.

#### **3.3.6. Переносимость (мобильность)**

Серверная часть ПО реализована в виде контейнерных образов и запускается на любой совместимой платформе Kubernetes — в публичных облаках, в инфраструктуре заказчика (on-premise) и в гибридных конфигурациях. Клиентская часть выполняется в любом современном веб-браузере без необходимости устанавливать на рабочее место пользователя дополнительное программное обеспечение.