

**ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ В ОБЛАСТИ ПОДГОТОВКИ И
ОЦЕНКИ СООТВЕТСТВИЯ ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ
«СОКОЛ» (ППО «СОКОЛ») ВЕРСИИ 1.0**

ОПИСАНИЕ ПРОГРАММЫ

Страниц 12

МОСКВА

2026

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Обозначение и наименование изделия

Полное наименование изделия: Прикладное программное обеспечение автоматизации деятельности в области подготовки и оценки соответствия объектов информатизации «СОКОЛ».

Сокращённое наименование изделия: ППО «СОКОЛ».

ППО «СОКОЛ» является централизованной платформой авторизации и аутентификации с прикладным модулем «Планировщик проектов». Платформа позволяет автоматизировать деятельность в области подготовки и оценки соответствия объектов информатизации.

Модуль «Планировщик проектов» не является самостоятельным приложением, а является частью «ППО СОКОЛ».

Состав изделия:

- микросервис авторизации AuthService (единый источник истины по пользователям, организациям, ролям, продуктам и подпискам);
- панель администратора платформы (Admin UI);
- клиентский интерфейс входа и личного кабинета (Auth UI);
- микросервис модуля «Планировщик проектов» (Planning Backend);
- клиентский интерфейс планировщика (Planning UI).

ППО «СОКОЛ» распространяется в виде интернет-сервиса (SaaS-решение), специальные действия по установке на стороне пользователя не требуются. Сторонние лицензионные ключи для базовой функциональности не требуются: комплекс является собственной разработкой ООО «Иновационные технологии» с использованием открытого программного обеспечения.

Модульная модель поставки: доступ организации к прикладному модулю «Планировщик проектов» определяется записью о приобретении продукта (purchase) в AuthService. Без действующей подписки модуль организации недоступен.

1.2. Языки программирования, на которых написана программа

При разработке ППО «СОКОЛ» использовались следующие языки и среды исполнения:

- Python 3.14.6 — серверная логика микросервисов AuthService и Planning Backend (FastAPI, SQLAlchemy 2.x async, Alembic);
- TypeScript / JavaScript (ECMAScript) — клиентские приложения Admin UI и Planning UI (React 19, Vite 8);
- SQL — схема и миграции баз данных MariaDB;
- языки описания конфигурации развертывания (systemd Quadlet / контейнерные манифесты, Nginx).

Для сборки программных компонентов ППО «СОКОЛ» используется оборудование со следующими программно-аппаратными характеристиками:

№ АРМ	Характеристики АРМ	Операционная система, ПО	Функции
1	Системный блок на базе платформы Intel: CPU Intel Core i5 2.0 GHz и выше; 16 GB RAM; 512 GB SSD; сетевой адаптер 1 Gbit/s; монитор, клавиатура, манипулятор «мышь».	ОС: ALT SP SERVER 10 (допускаются Astra Linux, РЕД ОС, Ubuntu LTS). Инструменты: Python 3.14.6, Node.js 26.8.2, Vite 8, TypeScript 7, Git 2.x, Podman / Docker. Дополнительно: ФИКС-UNIX 1.0, auditd.	Среда сборки исходного текста в загрузочные модули (wheels Python, статические бандлы UI, контейнерные образы)
2	Сервер Lenovo ThinkSystem SR630 Type 7X02 или эквивалент: Intel Xeon Silver 10C 2.2 GHz; 128 GB RAM; дисковый массив SAS/SSD не менее 4 ТБ; сетевая карта 1 Gbit/s.	ОС: ALT SP SERVER 10. Сервисы: GitLab 19.x; контейнерный runtime; MariaDB, Valkey, MinIO — для стендов сборки и приемочных испытаний.	Среда хранения исходного текста и объектного кода (GitLab), сборочный контур CI

Оборудование, указанное в таблицах выше, размещается по адресу: г. Москва, ул. Обручева, д. 36, к. 1, кабинет № 6.

1.3. Менеджмент документации и конфигурации ПО

Среда разработки ППО «СОКОЛ» включает:

- инструментарий конфигурации и сборки: Python 3.14.6, Node.js 26.8.2, Vite 8, Alembic, Git 2.x;
- систему управления версиями GitLab;

- контейнерную сборку (Dockerfile/ Quadlet) и обратный прокси Nginx.

В область действия системы управления конфигурацией включаются: исходный код программы; требования и эксплуатационные документы; загрузочные модули, в том числе модули сторонних разработчиков; инструментальные средства; сведения об обновлениях и устранении уязвимостей; перечень выявленных уязвимостей. Ответственность за полноту и актуальность информации по проекту лежит на руководителе отдела разработки.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1. Назначение изделия и классы решаемых задач

ППО «СОКОЛ» предназначен для централизованного управления идентификацией сотрудников, контрагентов и представителей организаций-заказчиков, а также для планирования, декомпозиции и контроля исполнения проектов в рамках приобретенного организацией модуля.

Классы решаемых задач:

- аутентификация по адресу электронной почты и паролю; выдача и проверка JWT (в том числе через JWKS); хранение сессионных признаков в защищённых cookie;
- ведение справочника организаций (исполнитель / заказчик), отделов, должностей, организационных ролей и связей «контрагент — организация»;
- учет продуктов (модулей) платформы и фактов их приобретения организацией с опциональным сроком действия;
- управление проектами, этапами и задачами, командой проекта, комментариями и версионизируемыми файлами (артефактами);
- разграничение доступа по организационным ролям, ролям в проекте и атрибутам ресурса (RBAC + ABAC);
- ведение истории смены статусов с обязательным указанием основания;
- разграничение внутренней и внешней (заказчик) области видимости задач и файлов, включая единственный предпроектный этап.

2.2. Роли, доступные в системе

Ролевая модель ППО «СОКОЛ» является двухуровневой и принципиально важна для корректной эксплуатации модуля «Планировщик проектов». Первый уровень —

организационные роли, хранящиеся в AuthService и назначаемые пользователю как свойство учётной записи. Второй уровень — роли в конкретном проекте, назначаемые члену команды проекта в Planning Backend. Дополнительно выделяется платформенная роль суперадминистратора.

Организационная роль определяет, что пользователь вправе делать в контуре своей организации (кадры, структура, создание проектов). Роль в проекте определяет, что он вправе делать внутри конкретного проекта, в команду которого включён. Один и тот же сотрудник может иметь разные проектные роли в разных проектах.

2.2.1. Платформенная роль

Роль (код)	Наименование	Назначение
superadmin	Суперадмин	Доступ к панели администратора платформы. Управление всеми пользователями, организациями, отделами, должностями, продуктами (модулями), покупками и связями контрагентов. Справочник организационных ролей доступен только для просмотра: создание, изменение и удаление ролей из панели запрещены.

2.2.2. Организационные роли (AuthService, используются в Planning)

Справочник organization_roles фиксирован. Каждому пользователю обязательно назначается ровно одна организационная роль (внешний ключ с ограничением RESTRICT).

Код	Наименование	Назначение и ключевые полномочия
admin	Администратор	Максимальные привилегии в рамках своей организации: управление сотрудниками, отделами, должностями, заказчиками и контрагентами. Может создавать проекты и управлять командой проекта. В планировщике относится к верхнему контуру полномочий (TOP_ROLES).
manager	Руководитель проектов	Управление проектами: создание проектов и этапов, назначение координатора, формирование команды проекта, назначение ответственных за этапы из числа начальников отделов. Входит в TOP_ROLES.

supervisor	Начальник отдела	Ответственный за этапы проекта. Назначает исполнителей задач из состава своего отдела (при условии, что сотрудники включены в команду проекта) и контролирует сроки и качество.
employee	Исполнитель	Базовый доступ. Является исполнителем задач в рамках этапов. Видит организацию, отделы и должности в режиме чтения. Не создает проекты и не назначает ответственных за этапы.

Типы субъектов, к которым применяются организационные роли:

- штатный сотрудник организации-исполнителя (organization_id задан, организация с признаком is_customer = нет);
- представитель организации-заказчика (организация с признаком is_customer = да, связанная с исполнителем через parent_org);
- контрагент (ИП / самозанятый или представитель внешней организации): не обязан иметь «домашнюю» организацию, связывается с организациями-исполнителями через таблицу contractors_orgs; признаки is_companу и companу_name носят информативный характер.

2.2.3. Роли в проекте (Planning Backend)

Роль в проекте назначается при включении пользователя в команду проекта и хранится в сущностях ProjectRole и TeamMember. Поле «роль в проекте» (project_job_title) является информационным наименованием участия и не заменяет код роли. Повторное включение одного и того же пользователя в одну команду проекта не допускается.

Код	Наименование	Типичный носитель	Кратко о правах
project_admin	Администратор проекта	Администратор организации, координатор	Полный набор прав на проект, этапы, задачи, команду, комментарии и артефакты, включая жесткое удаление.
project_lead	Управление проекта	Руководитель проектов, начальник отдела	Чтение проекта; управление этапами и задачами; назначение исполнителей; работа с комментариями и файлами. Нет создания проекта и управления

			составом команды.
project_member	Исполнитель	Штатный исполнитель задач	Чтение проекта / этапа / задачи; смена статуса своей задачи; заполнение отчета; комментарии; загрузка файлов.
project_viewer	Просмотр проекта	Наблюдатель	Только чтение проекта, этапов, задач, отчётов, состава исполнителей, комментариев и файлов.
external_collaborator	Внешний сотрудник	Контрагент в команде проекта	Работа в задачах (статус, отчёт, файлы, комментарии) без управления этапами и командой проекта.
customer_collaborator	Сотрудник заказчика	Представитель заказчика	Просмотр проекта и этапов; взаимодействие только с задачами внешней видимости и предпроектными задачами.

Правила связки организационной и проектной ролей, реализованные в коде и интерфейсе:

- координатором проекта может быть назначен только штатный сотрудник с организационной ролью «Руководитель проектов» (manager);
- ответственным за этап может быть назначен только член команды проекта с организационной ролью «Начальник отдела» (supervisor);
- при добавлении начальника отдела в команду поле проектной роли предзаполняется значением «Управление проекта» (project_lead) и недоступно для произвольного изменения;
- исполнителей задачи можно выбирать только из сотрудников отдела ответственного за родительский этап, а также из контрагентов — и только если они уже включены в команду данного проекта;
- нельзя исключить члена команды проекта, пока за ним сохраняются назначения внутри проекта (ответственный за этап, исполнитель задачи);

- заказчик не видит задачи с областью видимости «внутренняя»; для предпроектной задачи область видимости фиксирована как «внешняя» (заказчик) и не может быть изменена.

2.2.4. Матрица полномочий в проекте (RBAC)

Ниже — сжатая матрица по реестру Permission планировщика. Полный перечень кодов прав приведён в модуле `src/core/security/permissions.py`, назначение ролям — в `PROJECT_ROLE_PERMISSIONS` и `ORG_ROLE_PERMISSIONS` сервиса полномочий.

Группа прав	Адм. проекта	Управлен ие	Исполнит ель	Просмотр	Внешний	Заказчик
Проект: чтение	да	да	да	да	да	да
Проект: изменение / команда / удаление	да	нет	нет	нет	нет	нет
Этап: создание / назначение / удаление	да	частично	нет	нет	нет	нет
Задача: создание / назначение	да	да	нет	нет	нет	нет
Задача: смена статуса / отчёт	да	да	да	нет	да	нет
Комментарии: создание / свои	да	да	да	нет	да	да
Файлы: загрузка / новая версия	да	да	да	нет	да	да
Файлы: изменение доступа	да	да	нет	нет	нет	нет

У роли «Управление проектом» есть чтение, смена статуса и редактирование этапа, но нет создания этапа, назначения ответственного и удаления этапа. Сотрудник заказчика читает задачу и отчёт, создаёт комментарии и загружает файлы в пределах внешней видимости; смену статуса штатной задачи ему не поручают.

2.2.5. Полномочия организационного контура

Право	Администрато р	Рук. проектов	Нач. отдела	Исполнитель
Пользователи: просмотр	да	да	да	да
Пользователи: создание / изменение / удаление	да	да	создание и изменение	нет
Организация: просмотр	да	да	да	да
Организация: изменение / создание / удаление	да	да	нет	нет
Отделы: полный контур	да	да	просмотр и изменение	только просмотр
Должности: назначение сотрудников	да	да	да	нет

Начальник отдела не обладает правом жёсткого удаления пользователя (USER_DELETE).

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Структура программы

ППО «СОКОЛ» версии 1.0 построен по микросервисной архитектуре. AuthService является единым источником истины по идентичности. Planning Backend не хранит пароли и не создает учетные записи самостоятельно: он запрашивает актуальные сведения о пользователе у AuthService и кэширует необходимые атрибуты члена команды проекта (ФИО, email, организационную роль, тип субъекта).

Основные подсистемы:

1. Подсистема аутентификации и сессий (AuthService: пароль Argon2, JWT / JWKS, cookie).
2. Подсистема управления идентичностью и организационной структурой (пользователи, организации, отделы, должности, организационные роли, контрагенты).
3. Подсистема каталога продуктов и подписок (products, purchases, срок действия).
4. Подсистема панели администратора платформы (Admin UI и API /admin).

5. Подсистема управления проектами (проекты, этапы, задачи, статусы и история статусов).
6. Подсистема команды проекта и исполнителей задач.
7. Подсистема разграничения доступа (RBAC по ролям и ABAC по членству в команде, отделу, типу субъекта, области видимости).
8. Подсистема комментариев (древовидные обсуждения у проекта, этапа, задачи).
9. Подсистема артефактов (MinIO: загрузка, версионирование, список авторов цепочки, матрица доступа).
10. Подсистема журналирования и аудита (прикладные логи, история статусов, фиксация инициатора).

3.2. Взаимодействие компонентов

Клиентские приложения обращаются к соответствующим HTTP API. Planning Backend проверяет JWT, выпущенный AuthService (через JWKS), и при необходимости запрашивает карточку пользователя по внутреннему ключу AUTH_INTERNAL_API_KEY. Файлы задач и аватары размещаются в объектном хранилище MinIO (отдельные bucket для артефактов и аватаров). Короткоживущие служебные данные могут храниться в Redis.

Персистентность: каждая служба использует собственную схему MariaDB. Миграции выполняются Alembic. Удаление ключевых справочников ограничено политикой внешних ключей RESTRICT (нельзя удалить продукт, на который существуют покупки; нельзя удалить организационную роль, пока на неё ссылаются пользователи).

3.3. Основные сущности

Контур	Сущности	Назначение
AuthService	User, UserAddress, Organization, OrgAddress, Department, Position, OrganizationRole, ContractorOrg, Product, Purchase	Идентичность, оргструктура, подписки на модули
Planning	Project, Stage, Task, ProjectRole, TeamMember, TaskPerformer, Status, StatusChange, Comment, Artifact, ArtifactAccess, CompletionData	Декомпозиция работ, команда, статусы, файлы, отчеты о завершении

Статусы проекта, этапа и задачи унифицированы: Новый (new), В работе (ongoing), Реализован (completed), Заморожен (frozen), Отменен (cancelled), Согласование (pending). Смена статуса сопровождается записью StatusChange с указанием основания, инициатора и пары «исходный — целевой» статус.

Способ завершения задачи (CompletionMethod): смена статуса (standard), текстовое поле (text), загрузка файла (artifact). Для исполнителя переход в статус «Реализован» без выполнения выбранного способа блокируется; руководящий состав может завершить задачу в обход условия с дополнительным подтверждением.

Область видимости задачи (GlobalScope): internal — скрыта от заказчика; customer — видна заказчику. Файлы внешних задач показываются заказчику после перевода задачи в статус «Реализован», за исключением предпроектных задач, где взаимодействие заказчика допускается на всём жизненном цикле.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входные данные поступают:

- из веб-форм клиентских приложений (учетные данные, карточки организаций и сотрудников, реквизиты проектов / этапов / задач, комментарии, основания смены статуса);
- из загружаемых пользователем файлов (артефакты задач, версии документов, итоговые файлы завершения);
- из конфигурационных файлов окружения служб (параметры MariaDB, Valkey, MinIO, домены cookie, ключи внутренней интеграции);
- из JWT и JWKS при межсервисных и пользовательских запросах.

Выходные данные формируются:

- на экране дисплея пользователя (таблицы, карточки, диаграммы статусов этапов, ленты комментариев, цепочки версий файлов);
- в ответах HTTP API (JSON);
- в объектном хранилище MinIO (содержимое файлов);
- в журналах служб и в таблице истории статусов.

№	Расположение	Назначение
1	переменные окружения контейнера / secrets	Конфигурация БД, ключи, домены cookie, URL AuthService
2	том MariaDB	Персистентные данные идентичности и планировщика
3	том Valkey	Служебный кэш / временные признаки

4	тома MinIO	Артефакты и аватары
5	журналы служб (log/)	Прикладные логи
6	репозиторий GitLab	Исходный текст и артефакты сборки

5. ВЫЗОВ И ЗАГРУЗКА

ППО «СОКОЛ» поставляется в виде набора контейнерных образов и unit-файлов Quadlet / скриптов start_service.sh. Рекомендуемый порядок запуска контура Заказчика:

11. Подготовить изолированный контур (ОС семейства Linux, в том числе ALT SP, Astra Linux, РЕД ОС).
12. Развернуть MariaDB, Valkey и MinIO с паролями из секретов.
13. Применить миграции Alembic AuthService, загрузить справочники организационных ролей и продуктов, создать учетную запись суперадминистратора.
14. Запустить AuthService (backend, Nginx, Auth UI, Admin UI).
15. Применить миграции Planning Backend, загрузить справочники статусов и проектных ролей.
16. Запустить Planning Backend и Planning UI, указав AUTH_BACKEND_BASE_URL и внутренний ключ.
17. В панели администратора завести организацию-исполнителя, оформить покупку модуля «Планировщик проектов», создать администратора организации.

Доступ пользователей осуществляется через веб-браузеры Yandex Browser, Vivaldi Browser, Brave Browser и Google Chrome актуальных версий. Авторизация — по email и паролю. Самостоятельная регистрация извне платформы не предусмотрена: учетные записи создаются администратором организации либо суперадминистратором платформы.

Тестовый контур Разработчика доступен в поддерживаемых средах (local / staging / VM / выделенный контур аудита). Доменные имена cookie и список разрешённых Origin задаются переменными окружения и не должны совпадать между контурами.