# Предисловие

Цель проекта:

Создание приложения для защищенного электронного документооборота (ЭДО), позволяющего пользователям безопасно обмениваться документами, обрабатывать их и хранить без необходимости подключения к сети Интернет.

Технический результат проекта:

Разработка приложения для защищенного ЭДО в рамках закрытого информационного контура, обеспечивающего сбор, хранение, обработку, обмен и предоставление конфиденциальной информации.

Из требований к закрытому контуру можно выделить несколько основных особенностей разрабатываемого приложения, которые нужно согласовывать и на которые нужно уделять больше внимания:

1. Количество одновременных пользователей системы
2. Насколько распределена система по серверам
3. Необходимая скорость развертки приложения
4. Вариант на Django

Технологический стек проекта

|  |  |
| --- | --- |
| Компонент | Технология / Решение |
| Язык | Python 3.10+ (с типами, async-поддержка в Django) |
| Web Framework | Django 4.x (class-based views, DRF для API) |
| ORM | Django ORM + Alembic или встроенные миграции |
| БД | PostgreSQL |
| Хранилище файлов | MinIO |
| Криптография | PyCA Cryptography (AES 256 GCM, подписи) |
| Auth / SSO | Django Auth + LDAP (django-auth-ldap) + JWT |

Очереди и фоновые задачи Celery + Redis

|  |  |
| --- | --- |
| Компонент | Технология / Решение |
| Контейнеризация | Docker |
| Логирование | Django + Loguru + Filebeat |
| Метрики / Мониторинг | Prometheus + Grafana (через Django middleware) |
| UI | Django Templates или SPA на React (через DRF) |
| Desktop-клиент (опц.) | PyQt / Electron с API-интеграцией через REST |

Ключевые модули и алгоритмы

* 1. Модуль аутентификации пользователей:
     + - Форма логина и пароля с корпоративной почтой. o
       - Отправка одноразового TOTP-кода (PyOTP) на email через SMTP.
       - Интеграция с LDAP (через django-auth-ldap) для централизованной авторизации.
  2. Модуль управления доступом:
     + - Ролевое разграничение доступа (user/manager/admin) на основе django.contrib.auth.Group.
       - Возможность настройки прав через административную панель Django Admin или кастомный интерфейс.
       - Ограничение доступа к представлениям через декораторы и mixin'ы (например, UserPassesTestMixin).
  3. Модуль безопасности передачи данных:
     + - Все соединения обрабатываются через HTTPS (настройка TLS в nginx).
       - Файлы шифруются с использованием AES-256 перед сохранением (PyCA Cryptography).
       - Хранение ключей через ENV или систему Vault.
  4. Модуль проверки целостности документов:
     + - Каждая версия документа хешируется (SHA-256).
       - Реализуется цепочка версий с prev\_hash и current\_hash.
       - Проверка цифровых подписей на основе сертификатов в формате PKCS#12.
  5. Модуль управления журналом и статусом документов:
     + - Ведение журнала действий: загрузка, подпись, изменение статуса.
       - Статусы: создан, подписан, отправлен, архивирован.
       - Поиск, фильтрация и отчёты доступны через Django Admin и API.
  6. Модуль аналитики и мониторинга:
     + - * Django middleware для сбора метрик (время отклика, количество ошибок, загрузка Celery).
         * Метрики экспортируются в Prometheus, отображаются в Grafana.
         * Поведенческий анализ через SQL-запросы к журналам (например, выявление аномалий по активности пользователей)