Описание системы электронного документооборота

**Архитектурный обзор**

Система построена на микросервисной архитектуре с четким разделением компонентов:

1. Бэкенд-сервер: Django + Django REST Framework

2. Хранилище документов: MinIO (S3-совместимое объектное хранилище)

3. Криптографический модуль: PyHanko + Cryptography + pycades

4. Десктоп-клиент: Python Webview

5. База данных: PostgreSQL с шифрованием на уровне таблиц

6. Кеширование: Redis для сессий и OTP-кодов

**Детализированный функционал**

**1. Система безопасности**

Аутентификация:

- Двухфакторная аутентификация (2FA) с OTP по email

- OAuth 2.0 интеграция с корпоративными провайдерами

- Сессионные токены с JWT

- Автоматический выход после 15 минут неактивности

Шифрование:

- Сквозное шифрование документов (AES-256-CBC)

- Ключи шифрования индивидуальные для каждого документа

- Хранение IV (вектор инициализации) вместе с данными

- PKCS7 паддинг для обеспечения кратности блоков

Контроль доступа:

- RBAC (Role-Based Access Control) с 7 уровнями прав:

can\_manage\_documents = models.BooleanField(default=True)

can\_forward\_documents = models.BooleanField(default=False)

can\_sign\_documents = models.BooleanField(default=False)

can\_view\_statistics = models.BooleanField(default=False)

can\_modify\_groups = models.BooleanField(default=False)

can\_modify\_users = models.BooleanField(default=False)

can\_assign\_permissions = models.BooleanField(default=False)

- Мандатный контроль доступа (MAC) для конфиденциальных документов

**2. Управление документами**

Жизненный цикл документа:

1. Загрузка → Шифрование → Сохранение в MinIO → Регистрация в БД

2. Версионирование с неизменяемым журналом изменений

3. Распространение через:

- Индивидуальные доступы

- Групповые политики

- Иерархические цепочки подписания

4. Архивация по политикам хранения

Особенности хранения:

- Автоматическое версионирование в MinIO

- Хеширование SHA-256 для контроля целостности

- Метод "тепло-холодного" хранения:

- Активные документы - SSD-кластер

- Архивные документы - Glacier-подобное хранение

Операции с документами:

- Просмотр с водяными знаками

- Аннотирование PDF

- Сравнение версий

- Пакетная обработка

- Автоматическая классификация по контенту

3. Электронная подпись

Криптоинфраструктура:

- Работа с КриптоПро через pycades

- Поддержка TSP (Time-Stamp Protocol)

- Интеграция с аппаратными токенами

Процесс подписания:

1. Хеширование документа (SHA-384)

2. Формирование подписи PAdES

3. Встраивание в PDF:

```python

def sign\_pdf\_document(document, certificate, password):

writer = IncrementalPdfFileWriter(io.BytesIO(document))

signer = signers.SimpleSigner.load\_pkcs12(certificate, password)

pdf\_signer = PdfSigner(metadata, signer)

return pdf\_signer.sign\_pdf(writer)

```

4. Валидация через OCSP/CRL

Типы подписей:

- Простая (стандартная)

- Усиленная (квалифицированная)

- Групповая

- Согласующая

---

**4. Система аудита**

Мониторинг событий:

- Полнотекстовый поиск по логам

- Визуализация цепочек действий

- Интеграция с SIEM-системами

Фиксируемые события:

- Доступ к документам

- Изменения прав

- Попытки нарушения политик

- Системные операции

Особенности:

- Неизменяемое журналирование

- Цифровая подпись записей аудита

- Экспорт в машиночитаемых форматах

**5. Администрирование**

Управление пользователями:

- Гибкие ролевые профили

- Массовые операции

- Импорт/экспорт из LDAP/Active Directory

- Временные доступы

Управление группами:

- Иерархическая структура

- Наследование прав

- Делегирование управления

- Динамические группы по атрибутам

Политики:

- Сроки хранения

- Автоматическое удаление

- Квоты дискового пространства

- Гео-ограничения

**Десктоп-приложение: Детали реализации**

Технические характеристики

Архитектура:

- Облегченный клиент (4.2 MB)

- Нативный рендеринг через WebKit/EdgeHTML

- Офлайн-режим с синхронизацией

- Фоновые обновления

Интеграция с ОС:

- Системный трей с уведомлениями

- Перетаскивание файлов в интерфейс

- Контекстное меню проводника

- Горячие клавиши:

- Ctrl+S - Сохранить копию

- Ctrl+P - Подписать

- Ctrl+F - Отправить

Безопасность:

- Песочница для исполнения кода

- Защита от инъекций

- Верификация TLS-сертификатов

- Шифрованный локальный кеш

Производительность:

- Многопоточная обработка

- Предзагрузка ресурсов

- Адаптивная буферизация

- Компрессия LZ4