

Описание программного продукта

Slaidoteka

Exported on 09/12/2025

Table of Contents

1	Назначение	3
2	Функциональные возможности	4
2.1	Архитектура	5
2.2	Используемый стек технологий	5
2.3	Функциональные требования.....	6
2.4	Нефункциональные требования	7
2.5	Эксплуатационные характеристики.....	8
2.6	Обеспечение отказоустойчивости и высокой доступности	9

1 Назначение

«**Слайдотека**» – программный продукт для хранения, обработки и распространения презентационных материалов (слайдов) в корпоративной и образовательной среде. Он предназначен для создания централизованной библиотеки слайдов, которая облегчает повторное использование учебных и информационных презентаций в компаниях и учебных заведениях. Система поддерживает совместное использование контента, поиск по содержимому слайдов и обеспечивает контроль качества и безопасности материалов. В корпоративном сценарии «Слайдотека» служит базой знаний для сотрудников (например, для обмена маркетинговыми презентациями, обучающими материалами), а в образовательном – платформой для хранения лекционных материалов и презентаций преподавателей, доступных студентам. Таким образом, продукт повышает эффективность обмена знаниями и **предотвращает размножение дубликатов слайдов**, позволяя находить и повторно использовать уже готовый контент.

2 Функциональные возможности

«Слайдотека» реализована по микросервисной архитектуре и состоит из нескольких специализированных сервисов, каждый из которых отвечает за свою группу функций. Ключевые сервисы системы:

- **Core API** – центральный сервер приложения, предоставляющий REST API для взаимодействия с клиентскими приложениями (веб-интерфейсами). Core API обрабатывает запросы пользователей, оркеструет выполнение бизнес-логики и координирует работу других микросервисов. В частности, он принимает загрузку презентаций, инициирует их обработку, осуществляет поиск по слайдам, применяет правила доступа и агрегирует данные для выдачи результатов на фронтенд.
- **Parser** – сервис парсинга презентаций. Отвечает за обработку загруженных файлов презентаций (форматы *PPTX*, *PDF* и др.). Parser извлекает из файла отдельные слайды: конвертирует слайды в изображения, извлекает текстовое содержимое и метаданные. Полученные данные сохраняются в хранилище для дальнейшего использования. Этот сервис позволяет поддерживать **все распространённые форматы презентаций** и унифицировать их представление внутри системы.
- **AI Tagging** – сервис автоматического тегирования, использующий технологии искусственного интеллекта. Получает на вход результаты парсинга (изображения и текст слайдов) и выполняет анализ содержимого: распознаёт текст, объекты на изображениях, ключевые темы. На основе этого сервис присваивает слайдам семантические теги (метки) и категории, что обеспечивает интеллектуальный поиск и навигацию по библиотеке слайдов. Кроме того, **AI Tagging** выполняет *модерацию контента* – выявляет запрещённые или нежелательные материалы (например, нецензурную лексику, конфиденциальные сведения, неподходящие изображения) и помечает такие слайды для проверки администратором.
- **Viewer** – компонент просмотра и выдачи контента пользователям. Предоставляет функциональность поиска презентаций и просмотра слайдов через веб-интерфейс. Viewer обеспечивает быстрый рендеринг слайдов (просмотр изображений слайдов, отображение текста) и удобную навигацию. Пользователи могут находить нужные материалы по ключевым словам или тегам, а Viewer запрашивает данные через Core API и отображает найденные слайды. Фактически Viewer реализован как фронтенд-приложение (Web UI) для конечных пользователей – сотрудников или студентов.
- **Admin Panel** – административная панель (веб-интерфейс и соответствующий сервис) для администраторов и модераторов системы. Позволяет управлять пользователями и их ролями, настраивать права доступа (RBAC) на различные разделы и функции, а также просматривать и утверждать (или отклонять) контент, помеченный системой как требующий модерации. Через Admin Panel администраторы могут просматривать журнал действий, статистику использования, управлять справочниками тегов и экспортировать презентации при необходимости. Данный сервис обеспечивает разделение контуров (корпоративный vs образовательный) – администратор видит только данные своего контура.
- **Notification Service** – сервис уведомлений, ответственный за оповещение пользователей и администраторов о важных событиях. Он формирует и рассылает уведомления (например, электронные письма или внутрисистемные оповещения) о завершении обработки загруженной презентации, о выявлении запрещённого контента на слайде, о предоставлении доступа к материалам и т. п. Notification Service работает асинхронно, получая события из других компонентов (например, сигнал от AI Tagging о готовности результатов тегирования) и гарантируя своевременное информирование всех заинтересованных лиц.

Каждый из указанных сервисов масштабируется независимо и взаимодействует с другими через чётко определённые интерфейсы. Это обеспечивает гибкость расширения системы: при необходимости можно добавлять новые сервисы (например, дополнительные модули анализа контента) без нарушения работы существующих компонентов.

2.1 Архитектура



2.2 Используемый стек технологий

Название	Описание	Лицензия	Источник
C# .NET	Модульная платформа для разработки программного обеспечения с открытым исходным кодом	MIT License	https://learn.microsoft.com/ruru/dotnet/csharp/
PostgreSQL	Свободная реляционная объектносистема управления базами данных (СУБД)	PostgreSQL License	www.postgresql.org ¹
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети		https://ru.wikipedia.org/wiki/REST
Docker	Открытая платформа для разработки, доставки и запуска приложений	Apache License 2.0	https://www.docker.com/

¹ <http://www.postgresql.org>

2.3 Функциональные требования

Пользователи и доступ

- Тенантность: изоляция данных и настроек по организациям/брендам.
- Роли: Администратор, Модератор, Редактор, Наблюдатель, Сервис-аккаунт (API).
- SSO: OIDC/SAML2, SCIM-провижининг, MFA (по политике тенанта).
- RBAC/ABAC: права по коллекциям, проектам, тегам, экспортам, API-ключам.

Загрузка и приём контента

- Каналы: веб-интерфейс (drag&drop), REST API, пакетный импорт из S3/MinIO.
- Поддерживаемые форматы: PPTX, PDF (+ изображения внутри), (опц.) ODP.
- Ограничения: лимиты размера файла/количества слайдов по тарифу/тенанту.
- Контроль версий: автоматическое создание версии при каждом аплоаде/изменении.

Обработка и парсинг

- Разбор презентаций на слайды, миниатюры, извлечение текста и структуры.
- OCR для встроенных изображений; детекция языка; нормализация текста.
- Извлечение служебных метаданных (автор, заголовки, шрифты, размеры).

ИИ-тегирование и поиск

- Автотеги: тема, аудитория, предметная область, язык, сложность, чувствительность.
- Модерация/комплаенс: детекция запрещённого/чувствительного контента в тексте и на изображениях; политика по «пакетам правил» (организационные/юридические).
- Ручная разметка, массовое редактирование тегов, справочники/синонимы.
- Поиск: полнотекстовый и фасетный; подсветка фрагментов; «похожие слайды/деки» (similarity search).

Каталог и публикация

- Коллекции, разделы, курируемые подборки; закреплённые/рекомендуемые материалы.
- Просмотрщик: предпросмотр слайдов, сравнение версий, заметки и комментарии.

- Публикация: внутренняя ссылка, внешний доступ (по политике), срок действия, водяные знаки, запрет скачивания, «soft-DRM».
- Экспорт: PDF, PPTX (с оговорёнными ограничениями), CSV/JSON метаданных.

Команда и процессы

- Комментарии, упоминания, задачи на доработку, статусы («новая», «в обработке», «требуется модерации», «опубликована», «заблокирована»).
- Уведомления: email
- Журнал модерации и аудита (кто/что/когда изменил и почему).

Интеграции и API

- REST API (CRUD контента/метаданных/тегов/коллекций), вебхуки событий.
- Импорт.
- BI/аналитика: дашборды в Superset.
- Управление политиками хранения (ретеншн версий, архивация).

Администрирование

- Квоты/лимиты (объём хранилища, параллельные обработки, экспорт).
- Управление очередями/ретраями, перевод задач из DLQ, перезапуски пайплайнов.

2.4 Нефункциональные требования

Производительность (целевые значения)

- API (p95): ≤ 200 мс для CRUD и навигации; поиск (p95): ≤ 500 мс.
- Индексация/включение в поиск: ≤ 60 сек после завершения парсинга.
- Полный цикл обработки 100-слайдовой презентации: ≤ 3 мин (MVP ≤ 5 мин).
- Параллельные загрузки: ≥ 200 одновременно на тенант (горизонтальное масштабирование).

Масштабируемость

- Горизонтальный масштаб воркеров парсинга/ИИ/индексации.
- Независимые очереди/топики по типам задач и (опц.) по тенантам.

- Feature-flags и деградация неключевых функций при пиках.

Доступность и надёжность

- Доступность платформы: MVP 99.9%, целевая 99.95%.
- Безостановочные деплои (blue/green или canary), миграции с backward-совместимостью.
- Объектное хранилище с версионированием и erasure coding.

Безопасность и комплаенс

- Шифрование: at-rest (KMS), in-transit (TLS 1.2+), mTLS внутри кластера.
- Полный аудит действий (SEC, RBAC, IP allow-lists), WAF/Rate-Limit.
- Соответствие GDPR/ФЗ-152 (персональные данные), импортозамещение по требованию (Astra/РЕД ОС, Postgres/MinIO/RabbitMQ).
- OWASP ASVS L2; секреты — Vault; политика паролей и MFA.

Наблюдаемость и поддерживаемость

- Метрики/трейсинг/логи: Prometheus + Grafana и др.
- SLOs, алерты по латентности, глубине очередей, ошибкам обработки.
- IaC: Terraform + Helm; 12-Factor; стандартизированные health-checks.

Совместимость и UX

- Браузеры: последние 2 версии Chrome/Edge/Safari/Firefox.
- Локализации: RU/EN (расширяемо); доступность WCAG 2.1 AA.
- Мобильная адаптация (просмотр, поиск, комментарии).

2.5 Эксплуатационные характеристики

SLO/эксплуатационные метрики

- SLO API availability 99.95%, индикатор 99.9%.
- RPO ≤ 15 мин (журналы WAL/репликации), RTO ≤ 2 часа (DR).
- Лимиты по умолчанию: размер файла ≤ 1 ГБ, ≤ 1 000 слайдов/деку (настраиваемо).
- Пропускная способность пайплайна: не менее 1 000 слайдов/минуту на кластер из 10 воркеров (целевое).

Мониторинг и алертинг

- Дашборды: глубина очередей, лаг обработки, пиковая латентность, 5xx, ошибки парсинга/ИИ, процент блокировок модерации.
- Алерты: L1 (SLA/API), L2 (очереди/лаг), L3 (качество ИИ/рост ошибок OCR).

Логи, хранение и бэкапы

- Retention логов: 30 дней «hot», 365 дней архив.
- Бэкапы: Postgres PITR (WAL + ежедневные снапшоты), объектное хранилище с версионированием и lifecycle-политиками (архив/ Glacier-аналог).

Среды и релизы

- Environments: dev/stage/prod с изоляцией; выпуск релизов по GitOps.
- Миграции БД: forward-only, feature-flags; откаты через disable-фич и roll-forward.

Операции и поддержка

- Регламенты (runbooks) по инцидентам, ручной пере-триаж DLQ, ретраи.
- Плановые окна обслуживания без даунтайма (canary/blue-green).

Квоты и биллинг (для мульти-тенанта)

- Хранилище, экспорты/месяц, параллельные обработки, API-rate-limits.
- Отчёты по потреблению (для внутреннего биллинга/кросс-чарджа).

2.6 Обеспечение отказоустойчивости и высокой доступности

Инфраструктура и сеть

- Kubernetes в двух и более зонах доступности; ingress-контроллер с health-checks.
- Авто-скейлинг по метрикам (HPA/VPA), резерв по ёмкости $\geq 30\%$.

Данные и очереди

- Postgres: Patroni/pg_auto_failover, синхронная репликация между AZ, чтение с реплик; регулярные проверки восстановления.

- Объектное хранилище: MinIO cluster (erasure coding, ≥ 4 ноды) или S3 с версионированием, cross-region replication по политике.
- Redis: Redis Cluster или Sentinel-файловер (в зависимости от паттерна использования).
- RabbitMQ: кластер с quorum-очередями, mirrored-политики, DLX для ошибок; (опц.) Kafka для событий высокого объёма.

Сервисный слой

- Все сервисы — stateless; состояние в БД/хранилище/очередях.
- Идемпотентность операций (идемпотент-ключи), ретраи с экспоненциальной задержкой, circuit-breakers, тайм-ауты.
- Backpressure: ограничение параллельных обработок, очереди при пиках, graceful-degradation (отложенные миниатюры/«тяжёлые» ИИ-фили).

DR и тестирование устойчивости

- DR-сайт «тёплый» (warm standby): лог-шиппинг, периодическая валидация восстановления.
- $RPO \leq 15$ мин, $RTO \leq 2$ ч (целевые).
- Chaos-инженерия: регулярные фейловер-учения (БД, брокер, объектное хранилище).

Безопасность и защита

- WAF, rate-limit, защита от переполнения очередей/штормов событий.
- Сегментация сети, mTLS между сервисами, Vault для секретов.