ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

СИСТЕМА РЕДАКТИРОВАНИЯ И СОВМЕСТНОЙ РАБОТЫ

СИСТЕМА ХРАНЕНИЯ ДАННЫХ

3.3

АРХИТЕКТУРА

Версия 2

На 31 листах

Дата публикации: 15.04.2025

Москва
2025

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис», «MyOffice» и «Squadus» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе используются следующие сокращения с соответствующими расшифровками (табл. 1).

Таблица 1 — Сокращения и обозначения

Сокращение, термин	Расшифровка и определение
API	Application Programming Interface, интерфейс программирования приложений
Auth SSO	Single Sign-On, процесс аутентификации, позволяющий пользователю один раз войти в систему с одним набором учетных данных для доступа к нескольким приложениям или службам
AMQP	Advanced Message Queuing Protocol, открытый протокол прикладного уровня для передачи сообщений между компонентами систем
CO	Система редактирования и совместной работы
PGS	Система хранения данных
PSN	Приложение почты, календаря и контактов («МойОфис Почта»)
REST API	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети
S3 хранилище	Сервис хранения объектов, предлагаемый поставщиками облачных услуг
SPA	Single Page Application, веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX
SIEM	Security information and event management, система (или технология) управления информацией и событиями безопасности
Inventory	Файл, содержащий набор управляемых хостов для автоматизации установки и управления конфигурацией для сервиса Ansible
IOPS	Input/Output Operations Per Second, количество операций ввода/вывода
UI	User interface, интерфейс пользователя
ОС	Операционная система
СУБД	Система управления базами данных, комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными
Тенант	Логический объект, включающий в себя совокупность вычислительных ресурсов, репозиторий и пользователей
ПО	Программное обеспечение

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

В настоящем документе описана архитектура продукта и взаимодействие сервисов Системы редактирования и совместной работы и Системы хранения данных..

1.2 О компонентах

Система хранения данных — компонент, предназначенный для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей.

Система редактирования и совместной работы — компонент, предназначенный для индивидуального и совместного редактирования текстовых и табличных документов, а также просмотра и демонстрации презентаций.

Представленные компоненты входят в состав следующих продуктов:

– «МойОфис Документы Онлайн»;

– «МойОфис Профессиональный 2»;

– «МойОфис Профессиональный 3»;

– «МойОфис Схема»;

– Squadus PRO.

Подробное описание возможностей продукта приведено в документе «Функциональные возможности».

1.3 Перечень технической документации

С помощью технической документации, представленной в таблице 2, осуществляется развертывание серверной части, настройка и администрирование продукта.

Комплект документации распространяется на компоненты продукта:

– Система редактирования и совместной работы (CO);

– Система хранения данных (PGS).

Таблица 2 — Перечень технической документации

Наименование документа	Используемые компоненты	Содержание документа
«Системные требования»	CO, PGS	Системные и программные требования к продукту
«Архитектура»	CO, PGS	Описание архитектуры продукта для выбора типа установки и выделения ресурсов для серверов
«Система редактирования и совместной работы. Руководство по установке»	CO	Порядок установки системы редактирования и совместной работы
«Система хранения данных. Руководство по установке»	PGS	Порядок установки системы хранения данных
«Руководство по настройке»	CO, PGS	Настройка серверов продукта после установки и в ходе эксплуатации системы, а также процессов мониторинга и логирования
«Руководство по администрированию»	CO, PGS	Функции управления тенантом в ходе эксплуатации системы
«Руководство по резервному копированию»	PGS	Порядок резервного копирования баз данных, расположенных в системе хранения данных
«Руководство по мониторингу»	CO, PGS	Функции отображения текущего состояния системы и отдельных сервисов
«Руководство по работе с API»	CO, PGS	Набор методов для автоматизированного управления пользователями, группами, общими папками, доменами и тенантами

2 ОБЩАЯ КОМПОНЕНТНАЯ СХЕМА

На рисунке 1 представлена общая схема взаимодействия компонентов продукта.

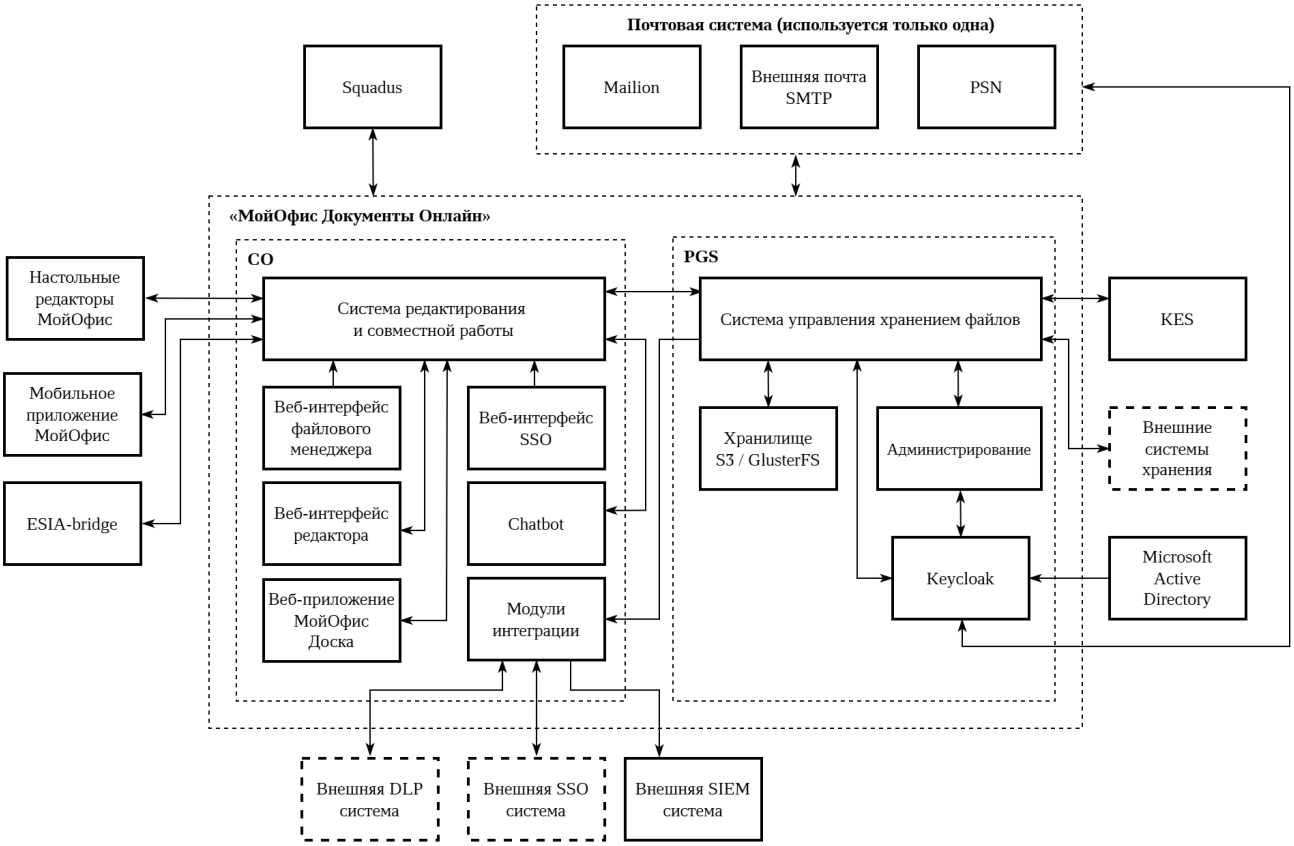

Рисунок 1 — Общая компонентная схема

2.1 Состав компонентов Системы редактирования и совместной работы

Состав и назначение компонентов Системы редактирования и совместной работы приведены в таблице 3.

Таблица 3 — Состав и назначение компонентов СО

Наименование компонента	Описание
Система редактирования и совместной работы	Серверный компонент, обеспечивающий: – совместное редактирование документов в Облаке с использованием любой платформы (веб, мобильные или настольные приложения); – взаимодействие клиентских приложений с Системой хранения данных; – интеграцию с почтовыми системами
Веб-интерфейс файлового менеджера	Веб-приложение «МойОфис Документы» предназначено для организации структурированного хранения файлов, выполнения операций с файлами и папками, настройки совместного доступа
Веб-интерфейс SSO	Веб-интерфейс предназначен для авторизации, аутентификации в системе и управления профилем пользователя. Главная страница доступа к приложениям (лендинг)
Веб-интерфейс редакторов	Включает в себя следующие редакторы: – веб-приложение «МойОфис Текст» — для создания и форматирования текстовых документов; – веб-приложение «МойОфис Таблица» — для создания электронных таблиц, ведения расчетов, анализа данных и просмотра сводных отчетов; – веб-приложение «МойОфис Презентация» — для создания, оформления и демонстрации презентаций
Мобильные приложения МойОфис	Мобильное приложение «МойОфис Документы» предназначено для просмотра и редактирования текстовых документов, электронных таблиц и презентаций, просмотра PDF-файлов, а также доступа к облачным хранилищам на смартфонах и планшетах с ОС Android, iOS и iPadOS. На устройствах под управлением ОС Аврора доступна возможность авторизоваться в продукте и осуществлять просмотр и редактирование файлов, размещенных в хранилище
chatBot	Сервис, обеспечивающий работу виджета Squadus в окне веб-редактора
Модули интеграции	Сервисы, обеспечивающие взаимодействие с внешними системами: DLP, SIEM, SSO

Система редактирования и совместной работы предусматривает подключение пользователей настольных редакторов из комплекта приложений «МойОфис Стандартный» (подробнее см. в документе «Руководство по настройке»).

2.2 Состав компонентов Системы хранения данных

Система управления хранением файлов — серверный компонент, обеспечивающий:

– хранение объектов пользователя;

– хранение общих корпоративных объектов;

– поиск по имени и содержимому;

– выполнение файловых операций для пользователя;

– разграничение прав доступа;

– аутентификацию и валидацию статусов учетных записей пользователей и прав доступа к системе.

В качестве веб-интерфейса используется файловый менеджер.

Состав и назначение компонентов Системы хранения данных приведены в таблице 4.

Таблица 4 — Состав и назначение компонентов Системы хранения данных

Наименование компонента	Описание
Keycloak	Сервис содержит список пользователей системы и их атрибуты. Предусмотрена возможность синхронизации пользователей из внешних каталогов (Active Directory). Keycloak обеспечивает хранение списка тенантов и их атрибутов продукта. Для управления параметрами установки от имени суперадминистратора используется графический интерфейс
Хранилище	Тип файлового хранилища определяется на этапе первичной установки компонента. Используется два типа хранилища: – файловая система (GlusterFS); – объектное хранилище S3 (minIO)*
Администрирование	Серверный компонент, обеспечивающий: – управление политиками безопасности и другими настройками тенанта (организации); – ролевую модель доступа к системе; – управление доменами, пользователями, группами пользователей, общими папками, публичными ссылками от имени администратора системы; – настройку доступности функциональных блоков продукта для пользователей. Для управления компонентом используется веб-интерфейс
** — Для хранилища типа S3 предусмотрена интеграция с другим S3-хранилищем в уже существующей инфраструктуре.*	

Для обеспечения уровня безопасности в Системе хранения данных предусмотрена интеграция с Kaspersky Endpoint Security (KES) (подробнее см. в документе «Руководство по настройке»).

3 ОПИСАНИЕ АРХИТЕКТУРЫ

3.1 Архитектурная схема Системы редактирования и совместной работы

На рисунке 2 представлена схема взаимодействия подсистем Системы редактирования и совместной работы. Прямоугольниками выделены кластерные решения для сервисов `etcd`, `redis`, `rabbitmq`.

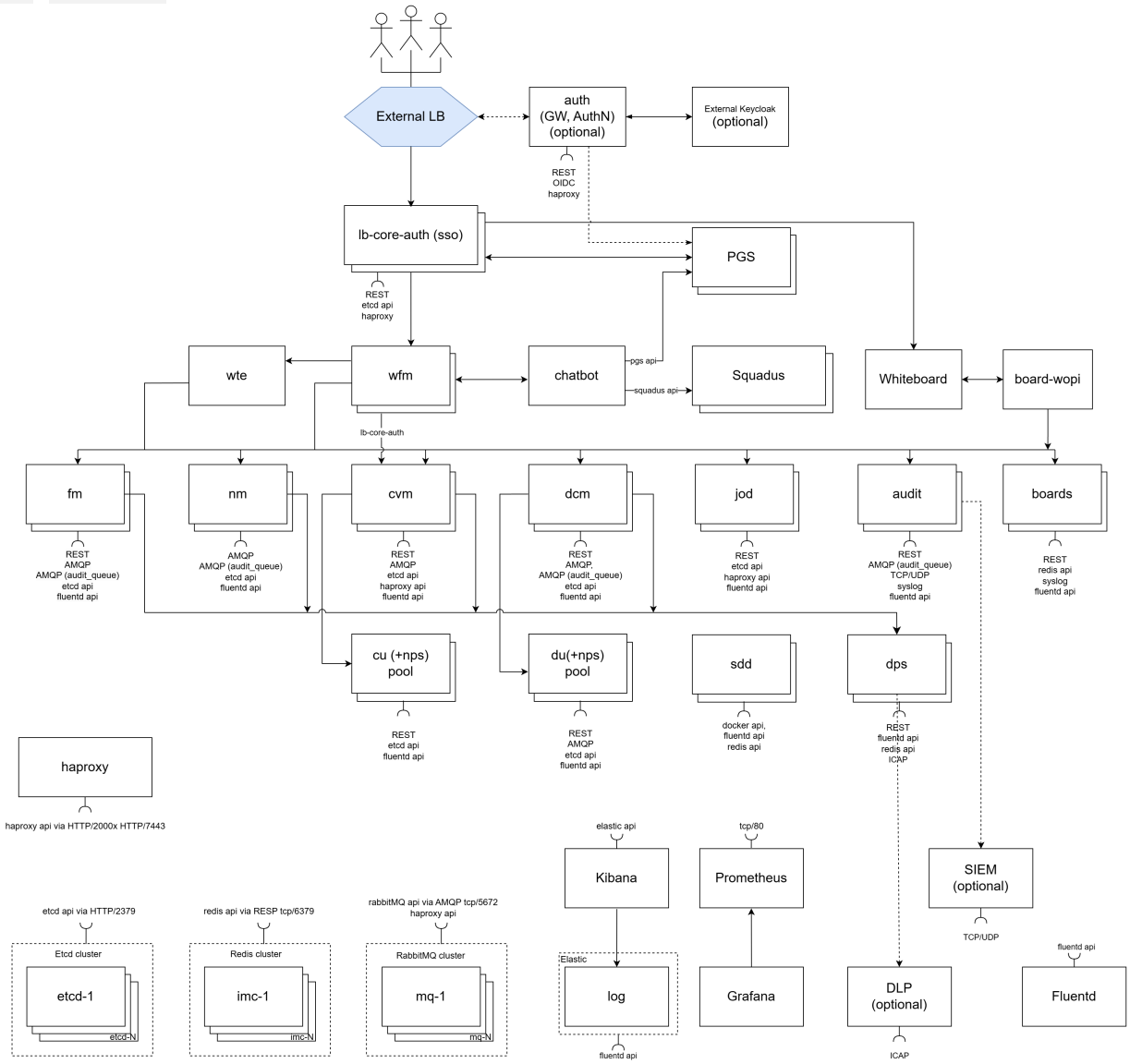

Рисунок 2 — Архитектурная схема Системы редактирования и совместной работы

3.1.1 Описание архитектуры Системы редактирования и совместной работы

Описание подсистем и сервисов Системы редактирования и совместной работы приведено в таблице 5.

Таблица 5 — Описание подсистем и сервисов Системы редактирования и совместной работы

Наименование подсистемы или сервиса	Описание
Auth	Компонент, состоящий из 2 сервисов: – GW — API-шлюз; – Authn — сервис аутентификации. Сервисы работают в тандеме и обеспечивают возможность интеграции с External Keycloak (SSO-системой)
Lb-core-auth (nginx-wfe)	Расширенный веб-сервер NGINX с поддержкой Lua. Отвечает за авторизацию, доступность API, балансировку
HAProxy	Балансировщик нагрузки между внутренними сервисами
FM	File Manager, сервис файлового менеджера. Отвечает за создание, удаление, добавление файлов, предоставление доступа другим пользователям
NM	Notification Manager, сервис управления уведомлениями
CVM	Conversion Manager, сервис управления конвертированием файлов. При конвертации документов использует пул CU юнитов или Jod
CU	Conversion Unit, экземпляр процесса конвертора различных форматов файлов (офисных или изображений)
DCM	Document Collaboration Manager, сервис управления редактированием документов. При редактировании документов использует пул DU юнитов
DPS	Data Protection Service, сервис сбора и передачи событий по REST API во внешнюю DLP-систему по протоколу ICAP
DLP	Data Loss Prevention, внешняя система защиты от потери данных
DU	Document Unit, экземпляр процесса редактирования и коллаборации документов
JOD	Java OpenDocument Converter, сервис конвертации офисных файлов устаревших форматов (например Microsoft Office 1997) в современные форматы или PDF
Audit	Сервис аудита, получает события аудита из сервисов и Системы хранения данных через RabbitMQ. Выполняет отправку событий аудита (TCP/UDP) в syslog или SIEM (не входит в стандартную поставку)
Elastic	Elastic (Opendistro Elastic Search), поисковый и аналитический сервис. Отвечает за сбор, обработку и хранение логов всех сервисов
Fluentd	Сервис, который собирает логи от всех сервисов и передает их в Elastic Search
Kibana	Kibana, инструмент визуализации и изучения данных, анализ логов приложения, использует данные из Elastic
Etcd	Хранилище типа «ключ-значение».

Наименование подсистемы или сервиса	Описание
	Отвечает за хранение свойств (настроек) всех сервисов
Redis	noSQL in-memory хранилище, типа «ключ-значение». Отвечает за хранение кешированной информации при работе с облаком (файлы, профили пользователей, токены)
RabbitMQ	Брокер сообщений на основе протокола AMQP. Отвечает за отправку сообщений между сервисами, например за отправку уведомлений, событий аудита
NPS	Native Process Service, сервис управления CU, DU
SDD	Service Detector Docker, обнаружение сервисов. Опрашивает сервисы NPS через Docker API и сокет
Prometheus	Инструмент сбора метрик со всех сервисов
Grafana	Инструмент визуализации метрик приложения. Получает метрики из Prometheus
WFM	Web file manager (SPA веб-приложение). Веб-клиент файлового менеджера (Виджет вложений). Отвечает за операции с файлами (кроме редактирования) и просмотр файлов по публичным ссылкам
Chatbot	NodeJs сервис. Отвечает за создание чата по документу, добавления пользователей и обновления списка пользователей в чате
WTE	Web text editors (Веб-клиент офисного приложения). Отвечает за редактирование и чтение файлов
External Keycloak	Внешняя SSO-система, сохраняющая в себе настройки системы, данные по тенантам и пользователям

3.2 Архитектурная схема Системы хранения данных

На рисунке 3 представлена схема взаимодействия компонентов Системы хранения данных.

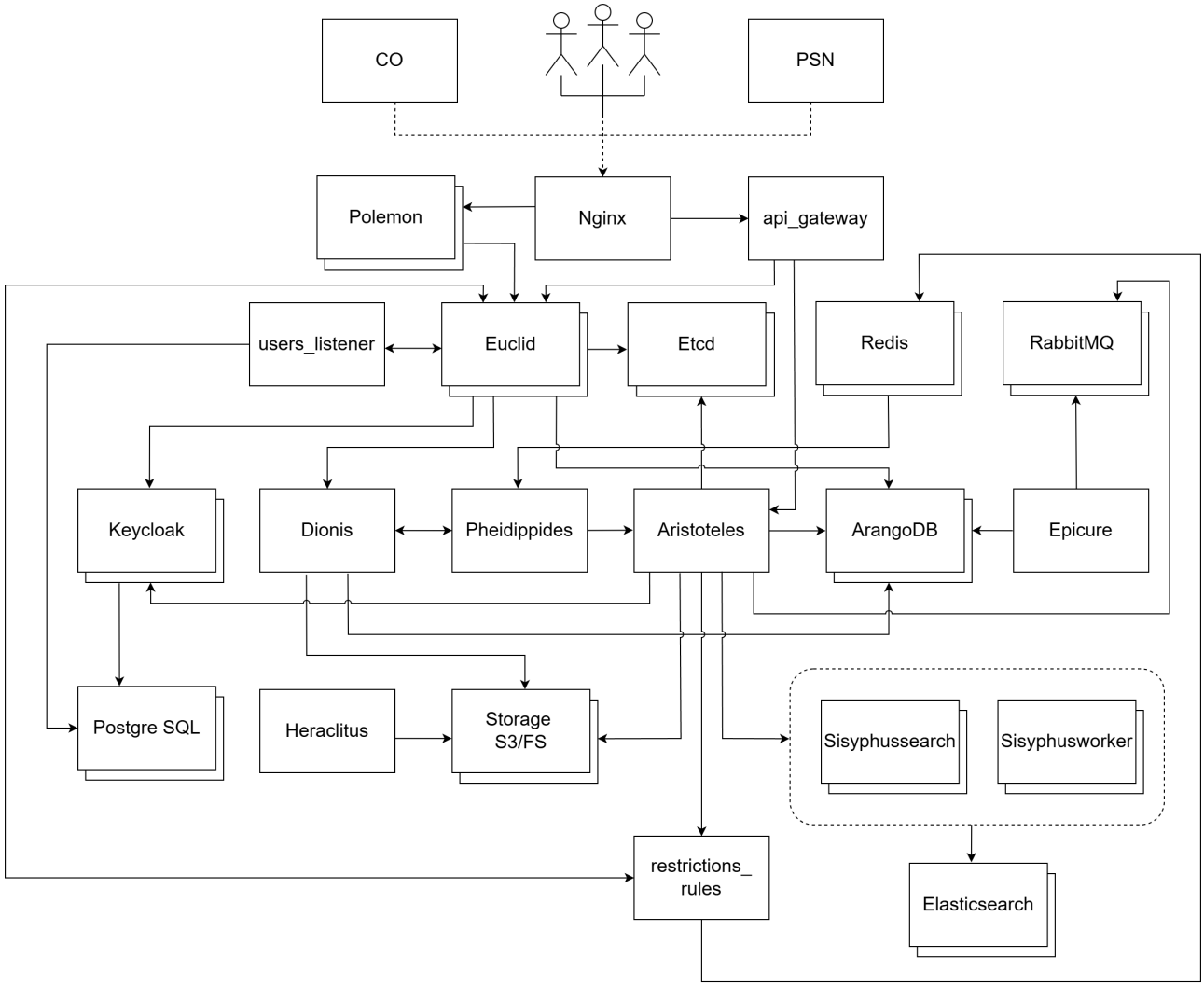

Рисунок 3 — Архитектурная схема Системы хранения данных

3.2.1 Описание архитектуры Системы хранения данных

Описание подсистем и сервисов Системы хранения данных приведено в таблице 6.

Таблица 6 — Описание подсистем и сервисов Системы хранения данных

Наименование сервиса	Описание
Arangodb	База данных, содержащая метаданные файлов (в т.ч. информацию о владельце документа, правах доступа и пр.)
Aristoteles	Сервер приложений, выступающий backend-частью для компонентов Системы редактирования и совместной работы в части выполнения файловых операций, разграничения прав доступа, версионирования, фиксации истории событий по объектам
Dionis	Сервис, отвечающий за удаление и переназначение прав доступа для объектов пользователей
Elasticsearch	Сервис, отвечающий за поиск по содержимому по хранящимся файлам
Epicure	Сервис формирования и отправки сообщений безопасности с последующей отправкой в аудит-системы (SIEM)
Etcd	Сервис, содержащий конфигурацию приложений, при кластерном развертывании также используется сервисом Postgres для создания кластера
Euclid	Rest API сервис, выступающий backend-частью для компонента polemon (веб-администрирование). Отвечает за администрирование пользователей в системе
Heraclitus	Сервис очистки архивных данных, удаленных пользователями из корзины. Может настраивать сроки хранения архивных данных и автоудаления их с диска по заданному расписанию
Keycloak	Сервис SSO, сохраняет в себе настройки системы, данные по тенантам и пользователям
Nginx	Прокси-сервис, обеспечивающий доступ до: rabbitmq, aristotels, euclid
Pheidippides	Сервис, осуществляющий обработку событий в Redis каналах (автоматическая блокировка IP-адресов/публичных ссылок)
Polemon	Сервис веб-администрирования Euclid (веб-интерфейс административной панели)
Postrgres	PostgreSQL, база данных для сервиса авторизации Keycloak
RabbitMQ	Очередь сообщений. Используется для передачи документов в elasticsearch для поиска по содержимому документов и для передачи межкомпонентных уведомлений от Системы хранения данных в Систему редактирования и совместной работы об изменении настроек хранилища
Redis	База данных «ключ-значение» для неперсистентных данных (в основном используется для хранения токенов и других авторизационных данных)
Sisyphus_ sisyphussearch	Сервис, осуществляющий поиск по содержимому документов в elasticsearch
Sisyphus_ sisyphusworker	Сервис, осуществляющий передачу файлов из rabbitMQ в elasticsearch

Наименование сервиса	Описание
Storage S3/FS	Блок storage осуществляет хранение файлов системы. В качестве хранилища FS в проекте используется GlusterFS. В качестве хранилища S3 в проекте используется minIO. Для хранения данных в объеме свыше 100 ТБайт рекомендуется использовать S3
api_gateway	Внутренний прокси-сервер PGS. Расположен после Nginx и перенаправляет запросы с постфиксом /pgsapi, /adminapi в соответствующие сервисы
user_listener	Сервис для прослушивания событий в PostgreSQL. Используется при интеграции с внешними каталогами (AD, FreeIPA, OpenLDAP). Отслеживает события изменений статуса пользователя и отправляет запросы в Euclid по http для обновления статуса пользователя
restrictions_rules	Сервис с правилами запретов и исключений. Интерфейс управления доступен из административной панели

4 ТИПОВЫЕ СХЕМЫ УСТАНОВКИ

4.1 Конфигурация без отказоустойчивости

Конфигурация без отказоустойчивости может использоваться при условии предоставления отказоустойчивости на уровне виртуализации.

Данная конфигурация характеризуется тем, что все серверные роли развертываются на единственном сервере. В такой конфигурации роли устанавливаются:

– на несколько виртуальных серверов с объединением ролей;

– на несколько виртуальных серверов, где одному серверу соответствует одна роль.

Установка такого типа не требует использования подсистемы балансировки.

Пример расчета аппаратных требований для установки на двух виртуальных серверах приведен в таблице 7.

В данном разделе приведены требования для развертывания системы без отказоустойчивости со следующим максимальным профилем эксплуатации:

– всего пользователей — 999;

– количество одновременно активных пользователей — 400;

– количество документов, редактируемых одновременно — 200.

⚠ Существует ограничение на количество пользователей ≤1000 из-за невозможности масштабирования конфигурации без отказоустойчивости (standalone). Требования идентичны для систем с общим количеством пользователей от 1 до 999.

Таблица 7 — Аппаратные требования для конфигурации без отказоустойчивости

Роли серверов	CPU, vCPU	RAM, Гбайт	SSD, Гбайт
operator*	1	4	50
Все роли Системы хранения данных	8	20	100
Все роли Системы редактирования и совместной работы	8	20	100
* — *сервер с ролью operator рекомендуется размещать на отдельном виртуальном сервере. После установки сервер с ролью operator не используется и может потребоваться для переустановки системы или ее сервисов.*			

Примеры файла inventory для Системы редактирования и совместной работы представлены в приложении Б, для Системы хранения данных — в приложении Г.

4.2 Кластерная отказоустойчивая конфигурация

В кластерной отказоустойчивой конфигурации каждая критическая роль реплицируется на разных виртуальных серверах. Разные роли могут быть объединены на одном виртуальном сервере. Архитектурных ограничений по объединению ролей нет. Виртуальные серверы разносятся по разным физическим серверам или гипервизорам.

Если один из серверов роли прекратил свою работу, общая распределенная роль продолжит работу сервиса на других серверах, и система сохранит свою работоспособность в полном объеме. Если все сервера роли прекратят работу, то система потеряет часть функциональности или станет полностью недоступна.

В данном разделе приведены рекомендуемые требования для развертывания системы в режиме кластера со следующим профилем эксплуатации:

– всего пользователей — 5000;

– количество одновременно активных пользователей — 2500;

– количество документов, редактируемых одновременно — 315.

Пример расчета аппаратных требований для отказоустойчивой установки приведен в таблице 8.

Таблица 8 — Аппаратные требования для кластерной отказоустойчивой конфигурации

Наименование роли*	Количество серверов	CPU, vCPU	RAM, Гбайт	SSD, Гбайт	HDD, Гбайт
operator	1	1	4	0	50
LB	2	2	4	50	0
core	2	10	24	100	0
infra	1	4	16	100	0
mq+imc+etcd	3	6	12	50	0
PGS-APP	2	10	12	0	70
STORAGE	2	6	12	0	11155
STORAGE-A	1	6	12	1 550	0
PGS-BE	3	10	14	0	3 350
PGS-DB	2	12	10	160	0
PGS-LOG	1	4	8	0	100
* — описания ролей представлены в приложении А					

При распределении ролей для кластерной отказоустойчивой конфигурации необходимо учитывать следующие требования:

– сервер `operator` не рекомендуется устанавливать на одном виртуальном сервере с приложением, но он может быть выключен после установки;

– для кластерных ролей `etcd / mq / imc` необходимо использовать минимум 3 узла, для `etcd` рекомендуется использовать 5 узлов;

– роли `cvm/cu-pool` и `dcm/du-pool` в такой конфигурации объединены;

– роль `PGS-LOG` не является критической;

– роль `LB` — внешний балансировщик.

Примеры файла inventory для Системы редактирования и совместной работы представлены в приложении В, для Системы хранения данных — в приложении Д.

4.3 Требования для кластера с профилем более 5000 пользователей

Для кластерной установки, на которой планируется работа более 5000 пользователей, необходимо обратиться к вендору для расчета размеров серверных ресурсов и получения рекомендаций по объединению ролей.

4.4 Расчет требований IOPS

Требования IOPS к различным ролям представлены в таблице 9.

Таблица 9 — Требования IOPS для ролей

Наименование роли [1]	Среднее значение IOPS	Максимальное значение IOPS [2]
PGS-APP	15	300
PGS-BE	30	100
PGS-STORAGE [3]	30	100
PGS-DB	50	100
PGS-LOG	20	50

[1] — описание ролей см. Приложении А

[2] — значения полученные при тестах эквивалентных работе системы при развертывании PGS на 80к пользователей

[3] — данный расчет приведен для при использовании MinIO с erasure set = 16.

Для работы сервиса пропускная способность дисковой подсистемы должна быть не ниже среднего значения IOPS, для комфортной работы пропускная способность должна быть выше максимального требования IOPS.

4.5 Типовая схема масштабирования

Полноценное масштабирование возможно использовать только для кластерной отказоустойчивой конфигурации.

Для standalone конфигурации возможно использование вертикального масштабирования с учетом ограничения Docker и других системных сервисов.

Переход от standalone конфигурации к кластерной выполняется при полной переустановке продукта. В Системе хранения данных переход обеспечивается с помощью предварительного резервирования баз данных.

Для Системы редактирования и совместной работы в первую очередь следует масштабировать узлы кластера с ролями:

- `dcm`, `du-pool` и `cu-pool` (влияет на количество одновременно открытых документов);
- `cvm` (влияет на количество конвертаций, скорость загрузки, скачивания, печати документов).

Для Системы хранения данных в первую очередь следует масштабировать серверы с ролью `ArangoDB`. Горизонтальное масштабирование невозможно, при увеличении нагрузки следует увеличить аппаратные мощности серверов.

4.6 Системные учетные записи

В таблице 10 и 11 представлены системные учетные записи, необходимые для связи сервисов системы.

Таблица 10 — *Системные учетные записи CO*

Имя учетной записи	Имя сервиса	Описание
app-co	PGS	Получение настроек тенантов
couser	Etcd_browser	UI конфигурации Etcd
couser	Openresty	Пользователь CO Manage API
admin	Grafana	Пользователь Grafana UI
admin	Elasticsearch	Пользователь ElasticSearch/Kibana UI
kibana	Elasticsearch	Пользователь для связи Kibana и ElasticSearch
couser	RabbitMQ	Пользователь RabbitMQ Management UI и подключения сервисов
root	RabbitMQ	Пользователь RabbitMQ для управления очередями и конфигурации во время деплоя
rabbitmq	RabbitMQ	Пользователь в PGS RabbitMQ для федерации с CO RabbitMQ

Таблица 11 — Системные учетные записи PGS

Имя учетной записи	Имя сервиса	Описание
app-co	Keycloak	Получение настроек тенантов
pgs	Keycloak	Управление тенантами/ Учетная запись в keycloak
root	Arango	Пользователь ARANGO DB
keycloak	Postgres	Пользователь СУБД Postgres
admin	Grafana	Пользователь Grafana UI
rabbitmq	RabbitMQ	Пользователь в RabbitMQ

ПРИЛОЖЕНИЕ А

Описание ролей при расчете аппаратных требований

Таблица А.1 — Роли для установки standalone

Наименование	Описание
PGS	Содержит службы хранилища, БД пользователей, администрирования, API хранилища
CO	Содержит службы SSO, веб-редакторов, collaboration API

Таблица А.2 — Роли для кластерной установки Системы хранения данных

Наименование	Описание
PGS-APP	Сервер вычислений, обработки запросов, API, keycloak, внутренний балансировщик
STORAGE	Сервер хранения данных (glusterfs)
STORAGE-A	Арбитр серверов хранения данных (glusterfs arbitrator)
PGS-BE	Сервер индексного поиска, хранения кэш, конфигурации, очередей (elasticsearch, etcd, arangodb_agent, redis, rabbitmq)
PGS-DB	Сервер БД пользователей, метаданных, авторизации (postgres, arangodb)
PGS-LOG	Сервер сбора логов компонента *Системы хранения данных* (registry, syslog. роли мониторинга), infrastructure

Таблица А.3 — Роли для кластерной установки Системы редактирования и совместной работы

Наименование	Описание
lb-core-auth	Сервер балансировки нагрузки *Системы редактирования и совместной работы*
infra	Сервер, объединяющий инфраструктурные роли сбора логов и мониторинга *Системы редактирования и совместной работы*. Может содержать роль chatbot
etcd	Подсистема конфигурации с использованием Etcd
core-cvm	Сервис управления импортом, экспортом и индексированием документов
cu-pool	Пул контейнеров с конвертерами документов
core-dcm	Сервер управления редактированием, коллаборации и документного API
du pool	Пул контейнеров с модулями редактирования документов в режиме коллаборации
core-fm	Подсистема сервиса файлового API
core-nm	Подсистема сервиса push-уведомлений
imc	Сервер кеширования сессий и хранения промежуточных результатов в памяти
mq	Сервер очереди сообщений и подписок
core	При сокращенном составе ролей — совмещенные роли *-core-* для *Системы редактирования и совместной работы*

Таблица А.4 — Технические роли

Наименование	Описание
operator	Технологическая роль. Рабочее место, с которого производится установка всех компонентов
LB	Сервер балансировки нагрузки для всех компонентов (используется только при кластерной установке)

ПРИЛОЖЕНИЕ Б

Пример файла inventory Системы редактирования и совместной работы (установка standalone)

```
all:
  children:

    co:
      children:
        co_chatbot:
          hosts:
            co-infra-1.installation.example.net:

        co_etcd:
          hosts:
            co-infra-1.installation.example.net:

        co_mq:
          hosts:
            co-infra-1.installation.example.net:

        co_cvm:
          hosts:
            co-infra-1.installation.example.net:

        co_cu:
          hosts:
            co-infra-1.installation.example.net:

        co_dcm:
          hosts:
            co-infra-1.installation.example.net:

        co_du:
          hosts:
            co-infra-1.installation.example.net:

        co_fm:
          hosts:
            co-infra-1.installation.example.net:

        co_jod:
          hosts:
            co-infra-1.installation.example.net:

        co_nm:
          hosts:
            co-infra-1.installation.example.net:

        co_imc:
          hosts:
            co-infra-1.installation.example.net:

        co_lb_core_auth:
          hosts:
            co-infra-1.installation.example.net:

        co_infra:
          hosts:
            co-infra-1.installation.example.net:
```

```
    co_setup:
      hosts:
        co-infra-1.installation.example.net:
```

ПРИЛОЖЕНИЕ В
Пример файла inventory Системы редактирования и совместной работы (кластерная установка)

```
all:
  children:

    co:
      children:
        co_etcd:
          hosts:
            etcd-1.installation.example.net:
            etcd-2.installation.example.net:
            etcd-3.installation.example.net:
            etcd-4.installation.example.net:
            etcd-5.installation.example.net:

        co_mq:
          hosts:
            mq-1.installation.example.net:
            mq-2.installation.example.net:
            mq-3.installation.example.net:

        co_cvm:
          hosts:
            cvm-1.installation.example.net:
            cvm-2.installation.example.net:

        co_cu:
          hosts:
            cvm-1.mcs.installation.example.net:
            cvm-2.mcs.installation.example.net:

        co_dcm:
          hosts:
            dcm-1.installation.example.net:
            dcm-2.installation.example.net:
            dcm-3.installation.example.net:

        co_du:
          hosts:
            dcm-1.installation.example.net:
            dcm-2.installation.example.net:
            dcm-3.installation.example.net:

        co_fm:
          hosts:
            fm-1.installation.example.net:
            fm-2.installation.example.net:

        co_jod:
          hosts:
            cvm-1.installation.example.net:
            cvm-2.installation.example.net:

        co_nm:
          hosts:
            nm-1.installation.example.net:
            nm-2.installation.example.net:
```

```
        co_imc:
          hosts:
            imc-1.installation.example.net:
            imc-2.installation.example.net:
            imc-3.installation.example.net:

        co_lb_core_auth:
          hosts:
            auth-1.installation.example.net:
            auth-2.installation.example.net:

        co_infra:
          hosts:
            log.installation.example.net:

    co_setup:
      hosts:
        auth-1.installation.example.net:
        auth-2.installation.example.net:
        cvm-1.installation.example.net:
        cvm-2.installation.example.net:
        dcm-1.installation.example.net:
        dcm-2.installation.example.net:
        dcm-3.installation.example.net:
        etcd-1.installation.example.net:
        etcd-2.installation.example.net:
        etcd-3.installation.example.net:
        etcd-4.installation.example.net:
        etcd-5.installation.example.net:
        fm-1.installation.example.net:
        fm-2.installation.example.net:
        imc-1.installation.example.net:
        imc-2.installation.example.net:
        imc-3.installation.example.net:
        log.installation.example.net:
        mq-1.installation.example.net:
        mq-2.installation.example.net:
        mq-3.installation.example.net:
        nm-1.installation.example.net:
        nm-2.installation.example.net:
```

ПРИЛОЖЕНИЕ Г

Пример файла inventory PGS (установка standalone)

```
all:
  children:
    pgs:
      children:
        pythagoras:
          hosts:
            host.installation.example.net:
        keycloak:
          hosts:
            host.installation.example.net:
        arangodb:
          hosts:
            host.installation.example.net:
              volume_device_arangodb: "False"
              volume_device_arangodb_path:            \
"/dev/disk/by-uuid/<UUID>"
        arangodb_agent:
          hosts:
            host.installation.example.net:
        search:
          hosts:
            host.installation.example.net:
              volume_device_elasticsearch: "False"
              volume_device_elasticsearch_path:       \
 "/dev/disk/by-uuid/<UUID>"
        redis:
          hosts:
            host.installation.example.net:
        rabbitmq:
          hosts:
            host.installation.example.net:
        postgres:
          hosts:
            host.installation.example.net:
              volume_device_postgres: "False"
              volume_device_postgres_path:            \
"/dev/disk/by-uuid/<UUID>"
        etcd:
          hosts:
            host.installation.example.net:
        nginx:
          hosts:
            host.installation.example.net:
        infrastructure:
          hosts:
            host.installation.example.net:
    storage:
      hosts:
        host.installation.example.net:
```

ПРИЛОЖЕНИЕ Д

Пример файла inventory PGS (кластерная установка)

```
all:
  children:
    pgs:
      children:
        pythagoras:
          hosts:
            host1.example.com:
            host2.example.com:
        keycloak:
          hosts:
            host1.example.com:
            host2.example.com:
        arangodb:
          hosts:
            host1.example.com:
              volume_device_arangodb: False
              volume_device_arangodb_path: "/dev/disk/by-uuid/<UUID>"
            host2.example.com:
              volume_device_arangodb: False
              volume_device_arangodb_path: "/dev/disk/by-uuid/<UUID>"
        arangodb_agent:
          hosts:
            host1.example.com:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
            host2.example.com:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
            host3.example.com:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
        search:
          hosts:
            host1.example.com:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: "/dev/disk/by-uuid/<UUID>"
            host2.example.com:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: "/dev/disk/by-uuid/<UUID>"
            host3.example.com:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: "/dev/disk/by-uuid/<UUID>"
        redis:
          hosts:
            host1.example.com:
            host2.example.com:
            host3.example.com:
        rabbitmq:
          hosts:
            host1.example.com:
            host2.example.com:
            host3.example.com:
        etcd:
          hosts:
            host1.example.com:
            host2.example.com:
```

```
            host3.example.com:
        nginx:
          hosts:
            host1.example.com:
            host2.example.com:
        postgres:
          hosts:
            host1.example.com:
              volume_device_postgres: False
              volume_device_postgres_path: "/dev/disk/by-uuid/<UUID>"
            host2.example.com:
              volume_device_postgres: False
              volume_device_postgres_path: "/dev/disk/by-uuid/<UUID>"
        infrastructure:
          hosts:
            host3.example.com:
        co_lb:
          hosts:
            co-lb1.example.com:
            co-lb2.example.com:
        co_auth:
          hosts:
            co-auth1.example.com:
            co-auth2.example.com:
    storage: #3 minimum
      hosts:
        host1.example.com:
        host2.example.com:
        host3.example.com:
  vars:
    PGS_CLUSTER: true
    ARANGO_CLUSTER: true
    DEFAULT_DOMAIN: 'example.com'
    ENV: ''
    NGINX_GOST_ENABLED: false
    ADMIN_INTERFACE_EXT_PORT: "443"
    API_INTERFACE_EXT_PORT: "443"
    CUSTOM_CA: false # if True put ca.crt to certificates folder
    HERACLITUS_CRON: "0 2 * * *"  # running time for service heraclitus
    # Service passwords
    KEYCLOAK_PASSWORD: "your_strong_password"
    KEYCLOAK_REALM_PASSWORD: "your_strong_password"
    KEYCLOAK_POSTGRES_PASSWORD: "your_strong_password"
    pgs_observability_mode: "internal"
    ARANGODB_PASSWORD: "your_strong_password"
    pgs_rabbitmq_password: "your_strong_password"
    REDIS_PASSWORD: "your_strong_password"
    PATRONI_REPLICATION_PASSWORD: "your_strong_password" # ignore when installing
postgres in standalone mode
    GRAFANA_ADMIN_PASSWORD: "your_strong_password"
    # Configurations
    SELINUX_ENABLED: false # Only for CentOS

    # default tenant configuration
    default_tenant:
      # required parameters
      ADMIN_PASSWORD: "your_strong_password"
      ADMIN_RECOVERY_EMAIL: "recovery@email.ru"
```

```
      # optional parameters (remove variables or leave values unchanged to use
defaults)
      MAX_USERS: "1000" # default: "1000"
      QUOTA_PER_USER: "1000000000" # value in bytes; default: "1000000000" (~1
GB)
      QUOTA_PER_GROUP: "10737418240"

    # Storage type 'fs' or 's3'
    storage:
      type: "fs"

      # If selected storage type is 'fs', select path for filesystem storage
      fs:
        path: "/media/storage/" # Take care about last slash
        retention_file_time: "30" # File storage time after delete from Trash,
days.

    # If selected storage type is 's3', enter configuration of s3 storage
      s3:
        minio_used: False
        minio_access_key: ""
        minio_secret_key: ""
        use_old_minio: True #please set True if you installed version of Minio
older than RELEASE.2022-06-25T15-50-16Z

        url: "http://minio:9000"
        secret_key: "" # Min 8 symbols
        access_key: ""
        bucket: ""
        service_name: "s3"
        region_name: "myoffice"
        acl: ""
        s3_max_capacity: ""

    # System settings
    system:
      TIMEZONE: "Europe/Moscow"

    # Integration variables
    # CO API addrress:port. format "https://co-api-host:8443/"
    co:
      coapiurl: "https://co-api-host:8443/"
      # CO auth-nodes load balancer
      co_lb: false
      vip_auth: "co-vip-ip"
      lb_keepalived_pass: "your_strong_password"
    # FS TOKEN SALT EXT (Synchro it with CO variable)
    installation_commons:
      fs_token_salt_ext: "oek02lk40_ldsjj3_sffgrfg"
      # FS App encryption settings.
      fs_app_encryption_key:
"1403386F683139F1975730171DC40967DE67201170AA5B1CBCCF894DD7815942"
      fs_app_encryption_iv: "AEB2DB5F3F3F4B4E3289C94C507F89D6"
      fs_app_encryption_salt: ""
      auth_encryption_key:
"1403386F683139F1975730171DC40967DE67201170AA5B1CBCCF894DD7815942"
      auth_encryption_iv: "AEB2DB5F3F3F4B4E3289C94C507F89D6"
      auth_encryption_salt: "BBA9B6CDBD87F78A"
```

```
      fs_app_password: "" # pwgen 10 1
      openresty_api_username: "couser" #Login to CO Api. Default 'couser' Sync it
with CO_MANAGE_API_USERNAME in CO
      openresty_api_password: "copass" #Pass to CO Api. Default 'copass'. Sync it
with CO_MANAGE_API_PASSWORD in CO
      two_fa_encryption_key: ""  # you should generate and put 64 key for using
for 2FA, you can use: pwgen 64 1
      audit_log_enabled: false
      chatbot_enabled: false


    POSEIDON_INTEGRATION: false
    POSEIDON:
      PBM_URL: https://pbm.myoffice-app.ru
      PBM_USER_PASSWORD: ""
      SSL_VERIFY: false
```

ПРИЛОЖЕНИЕ Е

Перечень изменений в документе

В данном приложении представлен перечень изменений относительно даты публикации и версии документа.

18.03.2025	Подготовлен документ версии 1.
15.04.2025	Подготовлен документ версии 2 со следующим изменением: обновлена общая компонентная схема