ПРОГРАММА ДЛЯ ЭВМ

ДатаБазис

Функциональные характеристики

Санкт-Петербург

2026 г.

Содержание

1. ОПИСАНИЕ

Таблица 1 - Описание

Наименование ПО	Программное обеспечение «ДатаБазис» (далее Платформа, ПО, «ДатаБазис»)

2. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Таблица 2 - Перечень русскоязычных терминов и сокращений

Сокращение / термин	**Полное наименование / определение**
ПО	Программное обеспечение

Таблица 3 - Перечень англоязычных терминов и сокращений

Сокращение / термин	**Полное наименование / определение**
API	Application Programming Interface (интерфейс прикладного программирования) — описание способов взаимодействия одной компьютерной программы с другими.
BI	ПО, предназначенное для построения различных визуальных форматов для аналитики данных и контроля за бизнес- и техническими метриками
JSON	JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript. Формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON
Kubernetes	Портативная расширяемая платформа с открытым исходным кодом для управления контейнеризованными рабочими нагрузками и сервисами, которая облегчает как декларативную настройку, так и автоматизацию.
ETL	Extract, transform, load. Общий термин для всех процессов миграции данных из одного источника в другой (другие связанные с этим термины – экспорт, импорт, конвертация).

3. ЦЕЛИ И ЗАДАЧИ РАЗРАБАТЫВАЕМОГО ПО

Таблица 4 - Бизнес-цели

№	**Бизнес-цели**
1.	Развитие систем аналитической отчетности за счет использования данных, обладающих следующими характеристиками:

	- объединенные (полученных из разных источников данных); - верифицированные (за счет валидаций при загрузке и преобразованиях и проверок качества данных); - описанные и имеющие понятное происхождение (в каталоге данных описаны данные и построены схемы их происхождения) - нормализованные (при возможности и необходимости) - агрегированные (при возможности и необходимости)
2.	Передача обработанных данных в другие сервисы.

Таблица 5 - Цели и задачи проекта

№	Цель ПО	Задача, решаемые ПО
1.	Сбор и консолидация данных из различных в едином хранилище данных.	Реализовать загрузку в хранилище данных передаваемого со стороны Заказчика данных из различных источников. Реализовать хранение и обработку данных, в т.ч. нормализованных и агрегированных.
2.	Повышение прозрачности обработки и анализа данных путем реализации бизнес-глоссария, каталога данных, модели данных.	Настроить обновление описаний и схем данных каталога данных, бизнес-глоссария на регулярной основе.
3.	Развитие аналитики данных	Запуск на регулярной основе процессов преобразования данных и сборки витрин аналитической отчетности, в т.ч. агрегированных. Вывод данных в визуальные форматы в BI-инструментах
4.	Поставка данных в другие сервисы	Запуск на регулярной основе процедур передачи данных в другие сервисы.

4. ИСХОДНЫЕ ДАННЫЕ

4.1. Источники разработки документа и нормативные ссылки

1. ГОСТ 34. Разработка автоматизированной системы управления (АСУ).
2. ГОСТ Р 59792-2021. Национальный стандарт Российской Федерации. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем.
3. ГОСТ Р 59793— 2021. Комплекс стандартов на автоматизированных системы.

Автоматизированные системы. Стадии создания.

5. ПОЛЬЗОВАТЕЛИ (ФУНКЦИОНАЛЬНЫЕ РОЛИ)

Таблица 6 Группы пользователей

№	Группа пользователей	Основная функциональность группы
1.	Администратор Системы	На всем протяжении функционирования Системы обеспечивает управлением правами доступа, аудит, распределение дискового пространства и других ресурсов платформы, модификацию структур БД.
2.	Владелец домена данных	Организация и сопровождение оперативного процесса управления данными, который включает поиск, мониторинг и изменение, интеграцию и организацию эффективного, безопасного использования информационных активов. Принимает решения относительно определений бизнес-терминов, политик, качества данных, требований к доступности и хранению данных
3.	Бизнес-эксперт домена	Управляет данными, является экспертом в предметной области с глубоким пониманием конкретного набора данных и его использования как бизнес-актива. Формирует политики и требования к качеству данных. Отвечает за бизнес-формулирование зон ответственности и доступа к данным.
4.	Технический эксперт домена данных	Специалист, проводящий экспертизу по домену, отвечающий за техническую сторону функционирования домена. Отвечает за структуру полей данных, модели базы данных, а также за управление технической средой, хранения, обслуживания и использования данных. Участвует в формулировании требований к качеству данных
5.	Аналитик данных	Занимается анализом данных, определяет

		требования к данным, поиском аномалий и требованиями к качеству и использованию. Участвует в проектировании модели данных.
6.	Аналитик качества данных	Отвечает за обеспечение соответствия данных требованиям к качеству данных. Формирует правила проверки, управляет отчётности по проверкам и настройке мониторинга качества данных.
7.	Администратор информационной безопасности	Контролирует применение и соблюдение общего регламента защиты данных в организации, без права вносить изменения в настройки доступов.

6. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

6.1.Функциональные модули

Перечень функциональных модулей Системы:

- Модуль хранения данных
- Модуль ETL
- Модуль управления данными
- Модуль отчетности
- Модуль администрирование и управление доступом
- Модуль журналирования и мониторинга

6.1.1. Модуль хранения данных

Таблица 7 Требования к функциям модуля Хранение данных

№	Наименование функции	Описание функции
6.1.1.1.	Архитектурное решение	Архитектура и модель данных должны обеспечивать расширения функциональных возможностей Системы, как путем подключения новых источников данных, так и за счёт расширения аналитических показателей и признаков в рамках уже внедренных блоков. Модуль хранения данных должен обеспечивать возможность извлечения данных по состоянию на заданную дату (извлечение среза данных). Модуль хранения данных должен обеспечивать ведение и проверку контрольных процедур и отчетов, подтверждающих корректность загруженных данных, соответствие бизнес-логике и модели данных.

		Модуль хранения данных должен обеспечивать наличие изолированных уровней хранения данных, обновление ядра хранилища не должно влиять на работу пользовательских витрин.
6.1.1.2.	Основная функциональность	Модуль хранения данных должен обладать гибкостью в применении различных подходов и моделей Системы: от полностью денормализованного хранения при подходе Data Lake до моделей Data Vault (1.0/2.0), Снежинка, Звездочка. Должны поддерживаться наиболее распространенные форматы данных, такие как Map, Array, Tuple, JSON и др.
6.1.1.3.	Масштабируемость	Модуль хранения данных должен расширяться путем добавления узлов (логических и физических): производительность решения должна повышаться с помощью добавления мощностей для обработки данных (напр. CPU, память, другие ресурсы).
6.1.1.4.	Информационная безопасность	Разграничение прав доступа (авторизация) пользователей, как по доступу к отчётам, так и на уровне данных. Обеспечение доступности данных с помощью автоматического резервного копирования хранилища данных и критичных приложений.
6.1.1.5.	Сохранение данных при авариях	Должна быть обеспечена сохранность информации при наступлении указанных событий: • Отключение сетевого электропитания • Отказ отдельных технических средств • Завершение работы системы • Отказ одного узла системы Должно быть обеспечено резервное копирование: • Файлов табличных пространств СУБД • Файлов инициализации и управляющих параметров СУБД

6.1.2. Модуль ETL

Таблица 9 Требования к функциям модуля ETL

№	Наименование функции	Описание функции

6.1.2.1.	Интеграция с внешними источниками	Система должна поддерживать возможность обмена (экспорта/импорта) данных в смежные системы через программные интерфейсы, поддерживающие форматы данных такие как JSON, XML, CSV, Parquet и т.д. Обновление программного и аппаратного обеспечения Системы не должно быть привязано к обновлению программных/аппаратных релизов систем-источников, интеграционный слой Системы должен обеспечивать независимость загрузки данных от обновления систем источников.
6.1.2.2.	Обработка и загрузка данных в Систему	Предполагается реализация тестовых процессов обработки синтетических данных, предоставленных Заказчиком, выполняющих демонстрационную функциональность, без поставки реальных данных. Загрузка связанных данных в Систему должна осуществляться с соблюдением принципов консистентности, независимо от типов и состава источников зависимых данных. Модуль должен обеспечить возможности повторной загрузка загружавшихся ранее данных обеспечивая идемпотентность. Модуль должен обеспечить реализацию процедур обработки недостоверных данных. Недостоверные данные (с ошибками, отсутствующими значениями) не должны помещаться в хранилище. Исполнитель предоставляет инструмент для указания правил качества данных и примеры валидации. Система должна предоставлять возможность выполнения операций по обработке данных, как в автоматическом, так и в ручном режиме, с возможностью запуска задач по расписанию и составления сценариев обработки данных с возможностью включения пользовательских действий для операций, требующих принятия решения или дополнительного контроля.
6.1.2.3.	Интеграция с внешними сервисами	Система должна передавать данные в другие сервисы в регулярном и/или ручном режиме

6.1.2.4.	Контроль за состоянием и работоспособностью модуля	Система должна обеспечивать возможность отслеживания статуса ETL-процесса с указанием количества элементов данных, извлеченных из систем источников, участвующих в ETL-процессе. Модуль должен обеспечить контроль процесса обработки информационного потока с возможностью проверки состояний процесса. Подсистема должна обеспечивать ведение журнала системных сообщений и ошибок.

6.1.3. Модуль Управление данными

Таблица 10 Требования к функциям модуля Управление данными

№	Наименование функции	Описание функции
6.1.3.1.	Формирование и ведение метаданных, включающих бизнес - описание (наименование, методология формирования описываемых данных, временные характеристики, единицы измерения, система источник и т.д.)	Обеспечение уникального кодирования транзакционных данных, группировку данных (например, по видам бизнеса, а также любым другим определённым в ходе эксплуатации группам), описание данных на естественном языке. Определение свойств данных, которые могут в явном виде отсутствовать среди полей в исходной системе, например, временная детализация (годовая, квартальная …), способ формирования (данные нарастающим итогом, данные за отдельный период), тип данных (план, факт, прогноз …); перечень свойств возможно расширять в ходе эксплуатации системы. Определение свойств аналитических разрезов описываемых данных, например: опциональность заполнения (возможно условная - для отдельных компаний или видов деятельности). Возможность определения места хранения данных в машиночитаемом виде с возможность предоставления этой информации (адреса систем, параметры подключения, наименования полей и колонок, ограничения выборки данных) для реализации автоматического доступа к данным. Возможность определения лиц, ответственных за ведение метаданных и лиц, ответственных за сами данные в источниках.

		Возможность определения уровня доступности данных (общедоступные, ограниченного доступа, персональные данных, коммерческая тайна и др.). Экспорт/импорт описаний метаданных в настраиваемых форматах данных. версионность метаданных. группировки и раздельное управление метаданными.
6.1.3.2.	Поиск в метаданных	Поиск записей по вхождению или полному соответствию (по одному или нескольким атрибутам). Определение происхождения производных данных с возможностью визуализации связей, как математических, так и логических.
6.1.3.3.	Обеспечение качества данных	Платформа данных должна обеспечивать ведение и проверку контрольных процедур и отчётов, подтверждающих корректность загруженных данных, соответствие бизнес-логике и модели данных. Создание и настройку стратегий поиска дубликатов. Возможность автоматического запуска предварительно настроенных проверок, которые в процессе эксплуатации могут быть расширены. Система показателей оценки качества информации. Журнал аудита по обеспечению качества информации. Для определенных правилами, регламентом и процессом данных предусматриваются подпрограммы их верификации – сравнение с источником. Подпрограммы аудита данных - осуществляют поиск в данных неполных записей, дублирующихся записей, неверных значений и отсутствия синхронизации данных в разных средах.

6.1.4. Модуль Отчетность

Таблица 11 Требования к функциям модуля Отчетность

№	Наименование функции	Описание функции
6.1.4.1.	Проектирование отчетных форм и витрин данных	В качестве базового слоя модуль опирается на денормализованные витрины данных, собираемые и

		хранимые в модуле хранения данных (датамарты) Реализуется в виде графического web-интерфейса, в котором пользователь без навыков программирования может создавать, редактировать и управлять настройкой регламентных отчетов и информационных аналитических панелей (дэшбордов).

6.1.5. Модуль Администрирование и управление доступом

Требования к модулю администрирование и управление доступом изложены в разделе 7.

6.1.6. Модуль журналирования и мониторинга

Таблица 12 Требования к функциям модуля Журналирования и мониторинга

№	Наименование функции	Описание функции
6.1.6.1.	Мониторинг системы и ее модулей	Система должна иметь механизмы централизованного сбора и хранения лог- сообщений компонентов используемых систем, а также предоставлять интерфейс для их выборки и анализа. Должны быть предустановленные измерения производительности, собираемые на уровне элементов, услуг, систем и приложений (напр., использование CPU, памяти, доступность). Должен поддерживаться мониторинг сервисов хранения данных, с возможностью корректировки порогов (например, порогов по нагрузке, на свободное место) и аварийных сигналов по достижении данных порогов.
6.1.6.2.	Оповещения от модуля мониторинга	Должны быть обеспечены функции оповещения администраторов о возникновении внештатных ситуаций, требующих вмешательства (например, потеря данных, сбой процессов). Должно поддерживаться оповещение администраторов об отказах по электронной почте.

6.2. Общие требования к Системе

6.2.1. Взаимодействие со смежными системами

6.2.1.1. Общие требования к взаимодействию со смежными системами

Таблица 13 Требования к взаимодействию со смежными системами

№	Функциональное требование	Целевое значение
6.2.1.1.1.	Взаимодействие со смежными системами	- Возможность получения и публикации данных в объеме предоставленных полномочий. - Обеспечение целостности данных в Системе после импорта информации из внешней системы. - Последовательный обмен информацией с несколькими внешними системами. Установка порядка обмена с несколькими внешними системами. - Настройка расписаний обмена с внешними системами. - Возможность ручного запуска обмена с внешней системой из административного интерфейса. - Настраиваемая возможность импортировать в Системе только часть данных внешних систем. - Настраиваемая возможность экспортировать из Системе только часть данных во внешние системы. - Возможность дополнительной обработки информации по каждому объекту обмена (поддержка скриптовых языков, фильтрация данных с использованием языка запросов SQL). - Передача только новых и изменённых объектов при обмене информацией с внешними системами. - Ведение журнала обмена информацией с внешними системами. Для каждой сессии обмена фиксируется: имя пользователя, инициирующего обмен; результат выполнения обмена с детализацией по типам объектов; дата-время обмена; длительность, количество и объём объектов обмена;
6.2.1.1.2.	Выгрузка данных по настраиваемым сценариям во внешние системы	Должна быть реализована возможность выгрузки данных по настраиваемым сценариям во внешние системы с помощью адаптеров.

6.2.1.2.Перечень персональных данных, обрабатываемых в Системе

Получение, хранение и передача персональных данных и коммерческой тайны не предполагается.

6.2.2. Хранение информации

Таблица 14 Требования к хранению информации

№	Наименование функции	Описание функции
6.2.2.1.	Хранение данных	Централизованное хранение в едином хранилище различных видов данных: реляционных, объектных, текущих, архивных, данных о событиях

7. ТРЕБОВАНИЯ ПО ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

7.1. Состав и содержание технических мер по обеспечению безопасности информации, обрабатываемой в Системе:

7.2. В Системе на прикладном уровне должны быть реализованы решения по выполнению следующих требований:

7.2.1. Идентификация и аутентификация субъектов доступа и объектов доступа:

В части идентификации и аутентификации субъектов доступа и объектов доступа в Системе должны быть реализованы следующие функции:

– идентификация и аутентификация пользователей и процессов, запускаемых от имени этих пользователей;

– однозначная идентификация и аутентификация пользователей для всех видов доступа;

– аутентификация пользователя с использованием паролей и/или аппаратных средств;

– возможность однозначного сопоставления идентификатора пользователя с запускаемыми от его имени процессами;

– формирование идентификатора, однозначно идентифицирующего пользователя;

– присвоение идентификатора пользователю;

– запрет повторного использования идентификатора пользователя в течении всего периода эксплуатации Системы;

– блокирование идентификатора пользователя после установленного Администратором Системы времени неиспользования (не более 90 дней);

– генерация и выдача начальной аутентификационной информации;

– установление характеристик пароля:

а) задание минимальной сложности пароля с определяемыми Администратором Системы требованиями к регистру, количеству символов, сочетанию букв верхнего и нижнего регистра, цифр и специальных символов;

б) задание минимального количества измененных символов при создании новых паролей;

в) задание максимального и минимального времени действия пароля;

г) запрет на использование пользователями определенного оператором числа последних использованных паролей при создании новых паролей;

– установка требования обязательной смены пароля при первом входе пользователя в Системе или после сброса пароля Администратором Системы;

– обновление аутентификационной информации с установленной периодичностью;

– защита аутентификационной информации от неправомерного доступа к ней и модифицирования;

– исключение отображения для пользователя действительного значения аутентификационной информации и (или) количества вводимых пользователем символов аутентификационной информации. Вводимые символы пароля могут отображаться условными знаками«*», «•» или иными знаками.

7.2.2. Управление доступом субъектов доступа к объектам доступа.

В части управления доступом в Системе должно быть обеспечено выполнения следующих функций:

– определение типа учетной записи: внутренний или внешний пользователь, учетная запись приложения, гостевая учетная запись, временная учетная запись;

– пересмотр и, при необходимости, корректировка учетных записей с периодичностью, определяемой Администратором Системы;

– объединение учетных записей в группы (при необходимости);

– заведение, активация, блокирование и уничтожение учетных записей пользователей;

– предоставление пользователям прав доступа к объектам доступа Системы на основе задач, решаемых пользователями в Системе и взаимодействующих с ней ИСиС;

– уничтожение (удаление) временных учетных записей пользователей, предоставленных для однократного (ограниченного по времени) выполнения задач в Системе;

– доступ к объектам доступа в соответствии с разделением полномочий (ролей);

– ограничение количества неуспешных попыток входа в Системе (доступа к Системе) за период времени, установленный Администратором Системы;

– блокирование учетной записи пользователя при превышении пользователем количества неуспешных попыток входа в Систему (доступа к Системе);

– в Системе должна быть реализована возможность ролевого метода управления доступом, определены типы доступа (операции по чтению, записи, удалению, выполнению и иные операции, разрешенные к выполнению пользователем (группе пользователей) или запускаемому от его имени процессу при доступе к объектам доступа) и реализованы правила ограничения доступа в соответствии с матрицей доступа;

– ограничение на число параллельных (одновременных) сеансов (сессий), основанное на идентификаторах пользователей и (или) принадлежности к определенной роли. В Системе для всех категорий пользователей запуск параллельных (одновременных) сеансов (сессий) от их имени с разных устройств (средств вычислительной техники) должен быть запрещен, а с одного и того же устройства ограничен количеством не более 2;

– в Системе в случае попытки входа под учетной записью пользователя или администратора, для которых достигнуто максимальное значение допустимых параллельных сеансов, при успешной аутентификации пользователя или администратора;

– блокирование сеанса доступа пользователя, после установленного Администратором Системы времени его бездействия (неактивности) в Системе или по запросу пользователя;

– блокирование сеанса доступа пользователя в Системе до прохождения им повторной идентификации и аутентификации;

– запрет действий пользователя в обход установленных процедур идентификации и аутентификации.

7.2.3. Регистрация событий безопасности.

В Системе должна быть реализована регистрация следующих событий:

– попытки входа субъектов в Системе (с регистрацией следующих параметров: время и дата события, идентификатор субъекта доступа, результат попытки входа (успешный или неуспешный));

– выход субъектов из Системы (с регистрацией следующих параметров: время и дата события, идентификатор субъекта доступа);

– формирование выходных форм (при выдаче на экран или печать), запрашиваемых пользователем Системы (с регистрацией следующих параметров: дата и время события, идентификатор субъекта доступа, сведения об объекте доступа (краткое содержание, наименование, вид, шифр, код);

– запуск ПО, предназначенных для обработки защищаемой информации Системы (с регистрацией следующих параметров: дата и время события, идентификатор субъекта доступа, идентификатор объекта доступа (запускаемого ПО));

– создание/изменение/удаление учетной записи (дата и время события, идентификатор субъекта доступа, тип события ИБ, краткое описание события ИБ);

– создание/изменение пароля учетной записи;

– назначение/изменение/удаление прав пользователей (дата и время события, идентификатор субъекта доступа, тип события ИБ, краткое описание события ИБ);

– блокировка учетной записи пользователя (дата и время события, идентификатор субъекта доступа, тип события ИБ, краткое описание события ИБ);

– удаление /изменение информации (дата и время события, идентификатор субъекта доступа, тип события ИБ, краткое описание события ИБ).

Доступ к записям аудита и функциям управления механизмами регистрации (аудита) должен быть только у пользователей с ролью «Администратор» и «Администратор ИБ».

8. ОПИСАНИЕ КОМПОНЕНТОВ ПО

8.1. Модуль хранения данных. Состоит из СУБД ClickHouse (далее СУБД).

СУБД представляет собой распределенную систему из, как минимум, 2 узлов (нод), развернутых на отдельных виртуальных машинах, управляемых операционной системой РЕД ОС.

Между которыми производится синхронизация (репликация данных), что позволяет избежать потерь данных при авариях на одном из узлов.

Процесс репликации нод управляется с помощью ClickHouse Keeper, развернутого на 3 отдельных нодах.

Для защиты от потери данных предусматривается регулярное резервное копирование данных (backup), полное копирование еженедельно, инкрементальное ежедневно. Хранятся 2 полные копии за последние 2 недели. Подробнее его настройка приведена в руководстве администратора.

СУБД используется для хранения различных видов данных, а также для выполнения расчетов на основе хранимых данных.

Хранение данных осуществляется следующим образом:

1) Сырой слой данных – база данных ODS (далее БД), содержащая совокупность таблиц, по структуре в точности повторяющих наборы данных, получаемые из источника с добавлением даты загрузки в СУБД. Также БД включает ряд вспомогательных таблиц: буферные (содержат последний загруженный набор данных) и проверочную (для записи контрольных значений из проверочного файла: количество строк, расчетный период, код поставщика и др.).

2) Слой хранения детальных данных – БД DDS, содержащая совокупность нормализованных таблиц с данными, разложенными по сущностям. БД может содержать ряд статичных справочников (не требующих реализации ETL процесса).

3) Слой витрин – БД datamart Слой предназначен для предоставления тематически ориентированных, согласованных и готовых к анализу наборов данных для конкретных бизнес-направлений, команд или аналитических задач.

4) Слой витрин агрегированных данных для дальнейшей передачи в модуль отчетности – БД datamart_aggrs. Таблицы данного слоя содержат агрегированные данные, очищенные от персональных данных, и коммерческой тайны.

5) Сервисная БД support содержит таблицу, хранящую информацию, решающую задачи поддержки и сопровождения: ошибок расчетов (validation_errors).

Для управления объектами хранения данных: создания, удаление и изменения баз данных, таблиц и колонок используется репозиторий product/Datamart/migrations в Gitflic. Репозиторий содержит текущую структуру СУБД, а также историю изменений СУБД.

Доступ к СУБД открыт по портам 10004 (native) и 10003 (http) через CHproxy.

Реализация ролевой модели и управление доступом.

Управление ролевой моделью и доступом производится в модуле администрирования и управления доступом.

Мониторинг и оповещения об инцидентах описаны в модуле мониторинга.

Метаданные для передачи в модуль управления данных.

При создании БД, таблицы, представления и колонки добавляются комментарии.

8.2. Модуль ETL. Состоит из Airflow – сервиса планирования и оркестрации запуска регулярных процессов обработки данных. Развернут в Kubernetes, на основе компонентов, взятых из репозиториев РЕД ОС.

В модуле посредством действий пользователей в веб-интерфейсе производится управление DAGs:

1) Запуск регулярных расчетов согласно заданному расписанию.

2) Ручные действия над DAGs: запуск, перезапуск, остановка процессов, пометка успешными/неуспешными предыдущих запусков.

3) Настройка и применение параметров для ручного запуска DAGs.

В модуле осуществляется управление регулярными расчетными процедурами.

При непрохождении ряда проверок, означающих критические проблемы с качеством данных, расчеты принудительно завершаются, пользователям направляются уведомления о неуспешном завершении расчета (проверки подробнее описаны в Приложении 2 к ПЗ).

Код для исполнения в DAG-ах доставляется автоматизированным процессом CI/CD (подробнее в описании системы обеспечения DevOps) из репозитория вида company-airflow-dags сервиса хранения и доставки кода Gitflic.

Для хранения метаданных компонента разворачивается СУБД PostgreSQL, развернутая на виртуальной машине, управляемой операционной системой РЕД ОС.

Управление ролевой моделью и доступом производится в модуле администрирования и управления доступом.

Мониторинг и оповещения об инцидентах описаны в модуле мониторинга.

Метаданные по проводимым расчётам автоматически собираются модулем управления метаданными.

При создании каждого DAG-а для каждого шага расчета задается источник данных (inlet) и целевой объект (outlet). Указанные параметры передаются в модуль управления данными для формирования визуальных схем процессов расчета данных (lineage).

8.3. Модуль управления метаданными. Состоит из сервиса OpenMetaData. Развернут в Kubernetes.

Сервис OpenMetaData собирает метаданные со следующих модулей Системы:

1) Модуля хранения данных (ClickHouse).

2) Модуля ETL (Airflow).

3) Модуля отчетности (Superset).

На основе собранных метаданных решаются следующие задачи:

1) Аккумулируются описания СУБД, баз данных, таблиц (а также представлений), колонок из ClickHouse, метрик и дашбордов из Superset. Объекты выстроены в иерархические связи, каждый имеет текстовое описание, отражающее суть хранимых в нем данных. Полнотекстовый поиск, позволяет находить нужный объект, в т.ч при нечетких критериях поиска.

2) Выстраиваются визуальные схемы процессов обработки данных, позволяющие отследить происхождение данных. На схемах отражены как объекты (таблицы, файлы, источники, получатели), так и процессы, перемещающие данные от одного объекта к другому: запросы ClickHouse и Superset, DAG’s Airflow, материализованные представления ClickHouse.

3) Производится разметка метаданных дополнительными признаками, которых нет в исходных системах, позволяющими определить владельцев данных. Разметка тэгами позволяет определить относимость данных к персональным и другим категориям чувствительных данных. Иерархическая доменная классификация позволяет отнести данные к различным бизнес-областям и подобластям.

4) Ведется бизнес-глоссарий, описывающий основные метрики, позволяющий установить их математический и бизнес-смысл, сопоставить между собой, исключить дублирование и нечеткую трактовку терминов.

5) Производится управление качеством данных. Настраиваются проверки качества данных и направляются оповещения в различные каналы коммуникаций (электронная почта, мессенджеры). Проверки и отправка оповещений запускается автоматически по заданному расписанию.

Все объекты модуля и их признаки поддерживают версионность, что позволяет проследить историю изменений объекта и его свойств.

Для хранения метаданных компоненты разворачивается сервисная СУБД PostgreSQL, развернутая на виртуальной машине, управляемой операционной системой РЕД ОС.

Реализована единая ролевая модель и управление доступом из модуля администрирования и управления доступом.

Реализован мониторинг и оповещения об инцидентах, который подробнее описан в модуле мониторинга.

8.4. Модуль отчетности. Состоит из BI-системы Superset, развернутой в Kubernetes (развернута в демонстрационных целях, не предполагалась в Системе по ТЗ).

Модуль решает задачу сбора удобных для восприятия пользователей отчетных форм и визуализации бизнес-метрик. В демонстрационных целях предоставляются:

1) Соединение (коннект) с СУБД ClickHouse. Соединение позволяет просмотреть и выполнить запросы к объектам БД datamart_aggrs, для построения отчётных форм.

2) Представления данных (Datasets). Сервис позволяет получать представления данных, повторяющие таблицы исходной СУБД, либо с наложением дополнительных условий.

3) Чарты – формат, позволяющий без написания SQL-кода (на основе представлений данных из п.2) создавать отчетную форму с условиями отбора и рассчитанными метриками, а также создавать визуальные представления.

4) Дашборды – форматы, позволяющие объединять несколько визуальных форматов, созданных в чартах, устанавливать общую фильтрацию и удобный вывод на экран.

Реализована ролевая модели и управление доступом.

Управление ролевой моделью и доступом производится в модуле администрирования и управления доступом.

Осуществляется мониторинг работы модуля. Мониторинг и оповещения об инцидентах описаны подробно в модуле мониторинга.

В модуль управления данных передаются метаданные по следующим объектам: представления данных, чарты, дашборды. Описания метрики формируются непосредственно в модуле управления метаданными.

8.5. Модуль администрирования и управления доступом. Состоит из сервиса Keycloak, развернутого в Kubernetes из компонентов, размещённых в репозиториях РЕД ОС.

Модуль решает задачи:

1) Управления авторизацией и аутентификацией пользователей в компонентах, где предусмотрена работа пользователей: ClickHouse, Airflow, OpenMetaData, Grafana, Superset, Gitflic, OpenDashboards, ArgoCD.

2) Управление ролевой моделью доступов.

3) Сбора пользовательских событий и событий информационной безопасности пользовательских компонентов ПО.

Для решения задачи в сервисе реализуется функциональность, описанная в пп. 3.13.1-3.13.16.

Интеграция с пользовательскими компонентами реализуется стандартным механизмом Keycloak по протоколу OpenID Connect.

Для хранения метаданных компоненты используется сервисная СУБД PostgreSQL, развернутая на виртуальной машине, управляемой операционной системой РЕД ОС.

Мониторинг и оповещения об инцидентах описаны в модуле журналирования и мониторинга.

8.6. Модуль журналирования и мониторинга. Состоит из Grafana, VictoriaMetrics, Opensearch, OpenDashboards, Vector, AlertManager, развернутых в Kubernetes. Перечисленные сервисы развернуты из компонентов, размещенных в репозиториях РЕД ОС.

Модуль решает задачу сбора метрик и логов, их хранения и регулярного наблюдения для всех компонентов из всех модулей Системы, а также отправки уведомлений о критических событиях по метрикам и логам компонент.

Для решения задачи сбора метрик мониторинга и реализации оповещений VictoriaMetrics реализуется следующая функциональность:

1) Компонент vmagent собирает метрики с узлов системы и Kubernetes по сервисным селекторам и маршрутизирует в пул компонент vminsert.

2) Компонент vminsert шардирует и реплицирует данные по кластерам vmstorage.

3) Компонент vmselect читает данные с vmstorage и отдает их в alertmanager и Grafana.

4) Компонент vmauth маршрутизирует чтение и запись данных по метрикам согласно лейблам, префиксов и других правил разметки.

AlertManager агрегирует работу с оповещениями в рамках Платформы данных (кроме оповещений по проверкам качества данных в OpenMetaData): направляет оповещения в различные каналы коммуникаций (электронная почта, мессенджеры).

В компоненте Grafana реализуется ряд визуальных форм (дашбордов) мониторинга состояния компонентов Платформы данных.

Компонент Vector реализует следующая функциональность:

1) Агенты Vector устанавливаются на каждом поде Kubernetes и каждой виртуальной машине вне Kubernetes (для СУБД модуля хранения ClickHouse, кластера сервисного PostgreSQL и Gitflic). Преобразования в логах выполняются непосредственно в агентах.

2) Собираются логи всех уровней.

3) Буферизация логов производится на диске, в буфере хранится до 10 тыс. записей. При невозможности записи логов в буфер (переполнении буфера) новые записи отбрасываются.

4) Логи под лейблами контроллеров Kubernetes не собираются.

5) Полностью исключается лейбл файла для избежания проблемы с попыткой сбора логов с отключенных/несуществующих подов.

6) Собранные и обработанные логи записывются в Opensearch.

Компоненты Opensearch и OpenDashboards поставляются с минимальными стандартными настройками, что позволяет в дальнейшем настроить вывод необходимых логов компонентов Платформы данных в визуальные форматы (дашборды), исходя из реальных потребностей мониторинга ПО.

Реализовывается ролевая модель и управление доступом. Управление ролевой моделью и доступом производится в модуле администрирования и управления доступом.

8.7. Сервисная подсистема обеспечения Devops. Предназначена для поддержки пользовательского функционала Платформы. Состоит из сервисов Gitflic, Gitflic-runner, ArgoCD, Ansible, Terraform, Vault. ArgoCD и Vault развертываются в Kubernetes. Gitflic и Gitflic-runner являются свободно распространяемыми продуктами отечественного вендора и входят в реестр отечественного ПО. Gitflic и Gitflic-runner поставляются в виде образа виртуальной машины для быстрого развёртывания Системы. Ansible и Eerraform используются в процедурах CI/CD, запускаемых из Gitflic и выполняются на Gitflic runner.

Перечисленные сервисы решают следующие задачи:

1) Хранение и доставка исходного кода, манифестов и образов контейнеров, бандров и helm чартов

2) Хранение секретов, ключей и технических учетных записей.

3) Управление конфигурациями.

4) Управление обновлением компонентов.

5) Реализация процедур миграций и непрерывной доставки изменений

В компоненте Gitflic реализуется хранение программного кода, необходимого для развертывания и управления изменениями, репозитории содержат:

1) DDL-запросы для создания баз данных, таблиц, ролей и пользователей в СУБД ClickHouse и сервисной СУБД PostgreSQL.

2) Настройки и выполняемый код DAGs Airflow (Python).

3) Файлы дашбордов Grafana в формате JSON.

4) Различные конфигурации компонентов, например для проверок и оповещений в AlertManager или манифесты для развёртывания приложений с помощью ArgoCD.

5) Дистрибутив компонентов Платформы данных для первоначального развертывания.

В Gitflic используется агент (runner) с типом Docker.

В сервисе Vault производится хранение секретов и настроек:

1) Параметров DAG, переменных окружения и настроек соединений для Airflow.

2) Параметры настройки Keycloak.

3) Параметры настройки ArgoCD.

4) Параметры компонентов как клиентов Keycloak.

5) Данные технических учетных записей.

6) И другие данные, являющиеся секретными или зависящими от среды выполнения

Для хранения данных репозиториев Gitflic используется сервисная СУБД PostgreSQL, развернутая на виртуальной машине, управляемой операционной системой РЕД ОС.

Реализовывается ролевая модель и управление доступом. Управление ролевой моделью и доступом производится для компонента Gitflic в модуле администрирования и управления доступом.

Мониторинг состояния сервиса осуществляется в модуле журналирования и мониторинга.