

Описание технической архитектуры

Настоящий документ содержит описание архитектуры платформы PIMS, предназначенной для сбора, обработки, хранения данных о технологическом оборудовании и процессах на промышленных предприятиях.

На Схеме 1 представлена архитектура платформы, которая состоит из следующих блоков: коннекторов сбора данных, серверного ПО (ядро), хранилища данных и графических web-приложений. Ядро системы представлено набором сервисов, которые взаимодействуют друг с другом. Для упрощения использования схемы связи между сервисами не отображены, они обозначены в описании серверного ПО.

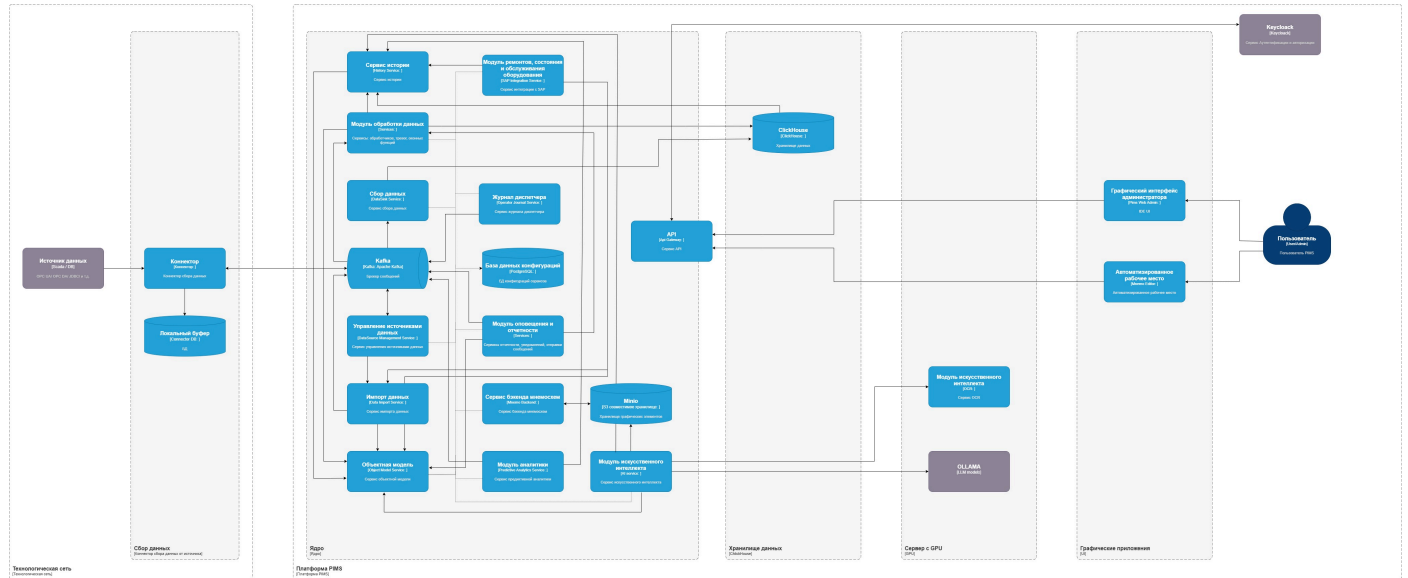


Схема 1. Архитектура платформы

Коннектор

Коннектор представляет собой программное обеспечение, которое устанавливается в технологической сети предприятия для сбора данных (значения тегов) из источников и передачи их в платформу через брокер сообщений.

Программа коннектора реализована на языке `C#` и представляет собой универсальный модуль получения данных от источников с помощью разных протоколов. Параметры запуска коннектора содержатся в файле `appsettings.json`.

Конфигурацию, в которой содержатся основные настройки сбора данных, коннектор получает из платформы (соответствующей очереди брокера сообщений) и сохраняет в файл `scrappers.conf`.

Коннектор имеет возможность принять команду на браунинг тэгов, где под браунингом понимается обнаружение всех доступных тэгов на источнике.

Коннектор содержит локальный буфер - БД SQLite, позволяющий избежать потери данных между источником и платформой в случае, если между коннектором и платформой какое-то время отсутствовала сетевая связь.

Коннектор получает данные от источника в соответствии с одним из поддерживаемых им протоколов, записывает их в локальный буфер, преобразует значения в сообщения формата `Protobuf` и отправляет в брокер сообщений в соответствующую очередь. После отправки в платформы данные помечаются на удаление и удаляются в течение заданного времени. Если данные не помечены, то произойдет повторная попытка отправки их в платформу. Преобразование сообщения из формата `Protobuf` происходит в сервисе платформы.

Серверное ПО

Серверное ПО или ядро платформы построено в формате сервис-ориентированной архитектуры и состоит из брокера сообщений, набора сервисов и баз данных конфигураций.

Брокер сообщений служит входным буфером пакетов данных, а также каналом передачи данных между сервисами платформы. Располагается в корпоративной сети. В качестве брокера сообщений используется `Kafka`.

Базы данных конфигурации сервисов являются хранилищем настроек сервисов и платформы. В качестве системы управления базами данных используется `PostgreSQL`.

S3 совместимое хранилище предназначено для хранения графической библиотеки, документов и конфигураций мнемосхем. По умолчанию в системе используется `Minio`.

Сервисы платформы представляют собой `web`-сервисы и предназначены для обработки данных, поступающих в платформу и сохранения значений в хранилище данных.

Сервисы платформы реализованы на языке `C#`. Каждый сервис отвечает за какую-то из функций платформы и взаимодействует с другими сервисами напрямую и посредством `API` платформы. Ниже в Таблице 1 представлено описание каждого из сервисов.

Таблица 1. Информация о сервисах

	☰ Имя сервиса ...	☰ Назначение ...	☰ Модуль (сх... ...	☰ Обращаетс... ...	☰ Инфраструкту... ...	+
1	Сервис сбора данных (DataSink Service)	Сервис отвечает за вычитывание данных из брокера сообщений и сохранение этих данных в хранилище данных			Хранилище данных (ClickHouse), Брокер сообщений (Kafka)	
2	Сервис управления источниками данных (DataSource Management Service)	Сервис отвечает за мониторинг состояния существующих коннекторов и конфигурирование параметров сбора данных и списка тегов		Сервис импорта	Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)	
3	Сервис тревог (Alarms service)	Сервис отвечает за формирование тревог на основе проверки соответствия значения тега условиям, настроенным пользователем		Сервис объектной модели Сервис истории	Хранилище данных (ClickHouse), Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)	
4	Сервис обработчиков (Handlers Service)	Сервис отвечает за преобразование значения тега, получаемого от источника, в соответствии с заданной логикой		Сервис объектной модели	Брокер сообщений (Kafka) Хранилище конфигураций (PostgreSQL)	
5	Сервис истории (History Service)	Сервис отвечает за обращение в хранилище данных для получения истории и актуальных значений тегов		Сервис объектной модели	Хранилище данных (ClickHouse)	
6	Сервис объектной модели (Object Model Service)	Сервис, который представляет собой иерархическое хранилище объектов, их атрибутов и сопоставление имен переменных объекта идентификаторам тэгов. Позволяет связать теги с источника с цифровой моделью предприятия			Хранилище конфигураций (PostgreSQL)	
7	Сервис импорта данных (Data Import Service)	Сервис отвечает за внесение в платформу данных, не получаемых с источников		Сервис объектной модели	Брокер сообщений (Kafka)	
8	Сервис API (Api Gateway)	Сервис отвечает за взаимодействие		Все сервисы платформы кроме Сервиса отправки	Keycloak	

9	Сервис оконных функций (Window Functions Service)	сервисов платформы и сторонних сервисов. Точка входа для сторонних сервисов Сервис отвечает за расчет показателей на основе истории тега за интервал - окно и пользовательского скрипта		сообщений и OCR сервиса Сервис объектной модели	Хранилище данных (ClickHouse), Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)	
10	Сервис отчетности (Reporting Service)	Сервис отвечает за формирование файлов, содержащих данные из хранилища данных, преобразованные и отформатированные в соответствии с пользовательской логикой	Модуль оповещения и отчетности	Сервис API - запросы через библиотеку на python	Хранилище конфигураций (PostgreSQL)	
11	Сервис уведомлений (Triggers Service)	Сервис отвечает за создание уведомления для рассылки получателям, в случае срабатывания условия - триггера	Модуль оповещения и отчетности	Сервис тревог, Сервис истории, Сервис отправки сообщений, Сервис отчетности	Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)	
12	Сервис отправки сообщений (Messaging Service)	Сервис отвечает за формирование и отправку сообщения на email	Модуль оповещения и отчетности		Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka), Почтовый сервер	
13	Сервис интеграции с SAP (SAP Integration Service)	Сервис отвечает за отправку в SAP данных по значениям тегов с заданной частотой и получение из SAP данных о состоянии оборудования и ТОиР	Модуль ремонтов, состояния и обслуживания оборудования	Сервис импорта данных, Сервис истории, Сервис объектной модели	Хранилище конфигураций (PostgreSQL) SAP S4	
14	Сервис предиктивной аналитики (Predictive Analytics Service)	Сервис отвечает за прогнозирование вероятности возникновения отказов оборудования и времени отказа на основе исторических данных с оборудования	Модуль аналитики	Сервис истории	Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)	
15	Сервис бэкенда мнемосхем (Mnemo Backend)	Сервис отвечает за серверную часть редактора мнемосхем и хранение библиотеки			Хранилище конфигураций (PostgreSQL), S3 совместимое хранилище (Minio)	

16	Сервис журнала диспетчера (Operator Journal Service)	графических элементов, мнемосхем и их иерархии. Сервис отвечает за возможность внесения в платформу текстовых заметок пользователями системы			Хранилище конфигураций (PostgreSQL), Брокер сообщений (Kafka)
17	Сервис искусственного интеллекта (AI Service)	Сервис отвечает за построение базы знаний, поиск аномалий и корреляций в данных, формирование пояснений к корреляциям, функционал ИИ-помощника	Модуль индустриального искусственного интеллекта	Сервис объектной модели, Сервис OCR, Сервис истории	Хранилище конфигураций (PostgreSQL), Ollama, S3 совместимое хранилище (Minio)
18	Сервис OCR (OCR)	Сервис отвечает за возможность распознавания текста из загруженных PDF-файлов	Модуль индустриального искусственного интеллекта		База данных обработки документов (SQLite), GPU (NVIDIA CUDA)
+					

Хранилище данных

Хранилищем данных выступает база данных временных рядов, которая хранит значения тэгов, принимаемых платформой, а также все данные, генерируемые платформой (новые производные тэги, расчётные значения оконных функций, список тревог). В качестве базы данных используется ClickHouse.

В отдельных таблицах хранятся: значения временных рядов для тегов, история тревог и результаты вычислений оконных функций.

Оптимизация поиска и получения актуальных значений достигается за счет использования материализованных представлений (materialized view) в ClickHouse.

Ограничение операций записи и чтения из БД осуществляется на уровне сервисов.

Графические web-приложения

Для взаимодействия пользователя с сервисами платформы, а также для получения доступа к ее функциональности, реализованы 2 графических приложения: Графический интерфейс администратора и Автоматизированное рабочее место.

Графический интерфейс администратора (IDE) представляет собой web-приложение, позволяющее с помощью API конфигурировать сервисы и настройки работы платформы.

Также графический интерфейс позволяет просматривать текущие значения выбранных тэгов, обработчиков, переменных объектов и тренды изменения этих значений.

Автоматизированное рабочее место (АРМ) представляет собой web-приложение для создания и редактирования мнемосхем и диспетчеризации тревог. Приложение предоставляет возможность отображения мнемосхем в режиме просмотра с возможностью ограниченного взаимодействия платформой (квитирования тревог, запрос отчётов).

Оба графических приложения реализованы на языке `Javascript`.

Оба приложения взаимодействуют с серверным ПО платформы посредством API. Доступ к API осуществляется на основании токена, получаемого пользователем при аутентификации и авторизации в KeyCloack.

Токен содержит информацию о перечне ролей, назначенных пользователю, на основании которых в приложениях Графический интерфейс администратора и Автоматизированное рабочее место становится доступной часть функционала.

Технологический стек

- Backend – .Net и EntityFramework.
- Frontend – Typescript + Vue 3 (composition api), pinia, vite, primevue.

- БД – ClickHouse, Postgres, S3 хранилище (MinIO), SQLite.
- Аутентификация и управление внутренними пользователями Системы на базе Keycloak.
- Сборка в Docker контейнерах, оркестрация контейнеров на базе Kubernetes.
- Реализация EDA на базе Kafka.
- Интеграция с внешними системами на базе: REST JSON.
- Тестирование в ручном режиме на базе разработанной библиотеки тест-кейсов.
- Хранение кода, CI\CD – на базе GitLab.
- Работа с логами и метриками – Opensearch + OpenSearch Dashboards, Prometheus/VictoriaMetrics + Grafana.
- Работа с LLM моделями Ollama.