САЙБОКС

ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА

СОДЕРЖАНИЕ

1 Термины и сокращения

Термины и сокращения, используемые в контексте настоящего документа:

Термин/сокращение	Определение
Организация	Юридическое лицо, использующее продукт в своей деятельности
Компонент	Отдельно выделенный микросервис, подсистема, подмодуль или компонент с открытым исходным кодом, входящий в состав модуля или продукта и применяемый для автоматизации определенного технологического или бизнес-процесса организации
Модуль	Отдельно выделенная система, сервис или приложение, входящее в состав продукта, предназначенное для автоматизации определенного технологического или бизнес-процесса организации
Продукт	Группа модулей / модулей и компонентов, целью которых является широкое покрытие автоматизацией ряда технологических или бизнес-процессов организации
ML	Machine learning, машинное обучение
ETL	Общий термин для всех процессов миграции данных из одного источника в другой (другие связанные с этим термины – экспорт, импорт, конвертация данных, парсинг файлов, web-scrapping и пр.)
python	мультипарадигмальный высокоуровневый язык программирования общего назначения
React	JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.
Nginx	веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах.
Prometheus	это бесплатное программное приложение, используемое для мониторинга и оповещения о событиях
Helm	инструмент для управления Kubernetes-приложениями
PostgreSQL	свободная объектно-реляционная система управления базами данных (СУБД).
Apache Kafka	распределённый программный брокер сообщений с открытым исходным кодом
Kubernetes (K8s)	открытое программное обеспечение для оркестровки контейнеризированных приложений
Grafana	открытая платформа визуализации, мониторинга и анализа данных, адаптированная под IT-системы
GraphQL	язык запросов данных и язык манипулирования данными с открытым исходным кодом для построения веб ориентированных программных интерфейсов
REST	архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

Термин/сокращение	Определение
VM	Виртуальная машина
S3	(Simple Storage Service) — облачный сервис хранения данных,
URL	(Uniform Resource Locator) — адрес ресурса в сети Интернет
API	(Application Programming Interface) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными
WEB	(Всемирная паутина) — распределённая система, предоставляющая доступ к связанным между собой документам
СУБД	Система управления базами данных
GPU	Graphics Processing Unit - один из видов микропроцессоров, который управляет памятью видеокарт
HTTPS	расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
KeyDB	нереляционная система управления базами данных (СУБД) класса NoSQL.
SSO	Технология единого входа (Single Sign-On) — это возможность использовать один идентификатор для доступа ко всем разрешённым ИТ-ресурсам и системам
GUI	Графический интерфейс пользователя - способ взаимодействия пользователя с компьютером с использованием графических элементов, таких как окна, кнопки и меню.
Инференс	В контексте моделей машинного обучения (ML) относится к процессу применения обученной модели к новым данным для получения предсказаний или выводов.
Репозиторий	Место, где хранятся и поддерживаются какие-либо данные.
Kafka	Распределённый программный брокер сообщений с открытым исходным кодом
gRPC	Система удалённого вызова процедур (RPC) с открытым исходным кодом

2 Общие сведения

Программный продукт Сайбокс (далее — продукт) предназначен для разработки ML моделей и их подготовки к промышленному внедрению.

Основные функции продукта:

- Разработка и отладка программных алгоритмов, в т.ч. пайплайнов обучения и применения ML моделей, с использованием языков программирования python, R и т.п.
- Подготовка и упорядоченное хранение наборов данных, используемых для работы с ML моделями
- Обогащение пользовательских данных с помощью инструмента разметки данных
- Создание и упорядоченное хранение артефактов ML моделей
- Запуск готовых ML моделей в виде самостоятельных сервисов (инференс)
- Автоматизация процессов с помощью ETL механизмов

Основные преимущества продукта:

- Командная работа в составе проектов
- Гибкое распределение вычислительных ресурсов
- Совместный доступ к артефактам данных и ML моделей с возможностью их переиспользования
- Сокращение времени, затрачиваемого на подготовку и тестирование прототипов модельных сервисов

3 Архитектура

Продукт построен по принципу многоуровневой архитектуры на едином технологическом стеке и с использованием общих архитектурных подходов.

3.1 Состав продукта

Продукт состоит из плотно интегрированных компонентов общего и специального назначения в составе следующих модулей:

- модуль управления организационной структурой;
- модуль управления ресурсами;
- модуль управления процессами.

На рисунке ниже приведена верхнеуровневая схема архитектуры продукта, на которой отображены основные компоненты модулей в составе продукта:

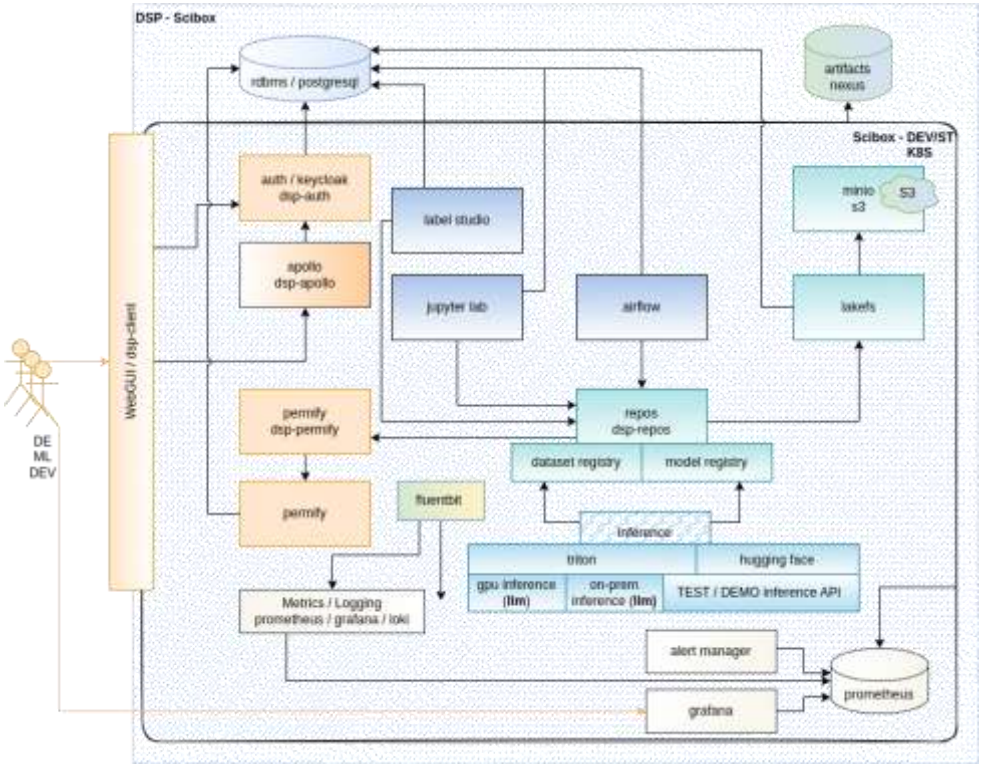

Рис. 1 Верхнеуровневая схема архитектуры продукта

3.2 Используемый стек технологий

Перечень компонентов, в том числе с открытым исходным кодом, которые используются при разработке и эксплуатации продукта:

Название	Описание	Лицензия	Источник
Python	мультипарадигмальный высокоуровневый язык программирования общего назначения	Python Software Foundation License	https://www.python.org/

Название	Описание	Лицензия	Источник
React	JavaScript-библиотека для создания пользовательских интерфейсов (используется для Frontend-части модуля)	MIT License	https://react.dev/
nginx	HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения	2-clause BSD License	https://nginx.org/
KeyDB	Система управления базами данных NoSQL-типа для временного хранения и быстрого доступа к сессионным данным	BSD-3-Clause license	https://docs.keydb.dev/
Apollo GraphQL	Сервер GraphQL (Язык запросов данных и язык манипулирования данными с открытым исходным кодом для построения веб ориентированных программных интерфейсов) от Apollo.	MIT	https://github.com/apollographql/apollo-server
Keycloak (quarcus)	Управление идентификацией и доступом с открытым исходным кодом	Apache License 2.0	https://www.keycloak.org/
Helm	Менеджер пакетов для Kubernetes	Cloud Native Computing Foundation	https://helm.sh/
Kubernetes	Платформа с открытым кодом для оркестрации контейнеризированных приложений	Apache License 2.0	https://kubernetes.io/ru/
PostgreSQL	Объектно-реляционная система управления базами данных	PostgreSQL License	https://www.postgresql.org/
lakeFS	Система контроля версий для озер данных.	Apache 2.0	https://lakefs.io/
Prometheus	ПО, используемое для мониторинга событий и оповещения	Apache License 2.0	https://prometheus.io/
Grafana	Мультиплатформенное веб-приложение для аналитики и интерактивной визуализации	GNU Affero General Public License, version 3.0	https://grafana.com/
Triton inference server	Стандартизированное ПО для развертывания и выполнения моделей искусственного интеллекта	BSD 3-Clause "New" or "Revised" License	https://www.nvidia.com/en-us/ai-data-science/products/triton-inference-server/
Apache Kafka	Распределённый программный брокер сообщений с открытым исходным кодом	Apache License 2.0	https://kafka.apache.org/
Nexus OSS edition	По для централизации и управления репозиториями	Eclipse Public License 1.0	https://www.sonatype.com/products/sonatype-nexus-oss
Label Studio	Инструмент маркировки данных с открытым исходным кодом	Apache-2.0 license	https://labelstud.io/

Название	Описание	Лицензия	Источник
Jupyter	Проект по разработке программного обеспечения с открытым исходным кодом,открытых стандартов и сервисов для интерактивных вычислений на нескольких языках программирования	3-Clause BSD License	https://jupyter.org/
Apache Airflow	Открытое программное обеспечение для создания, выполнения, мониторинга и оркестровки потоков операций по обработке данных.	Apache-2.0 license	https://airflow.apache.org/
Permify	Сервис авторизации с открытым исходным кодом	Apache-2.0 license	https://docs.permify.co/

4 Функциональные возможности

Продукт предоставляет следующие основные функциональные возможности:

- Разработку и отладку программных алгоритмов, в т.ч. пайплайнов обучения и применения ML моделей, в Jupyter Lab с использованием языков программирования python, R и т.п.
- Подготовку и упорядоченное хранение наборов данных, используемых для работы с ML моделями, их версионирование.
- Обогащение пользовательских данных с помощью инструмента разметки данных.
- Создание и упорядоченное хранение артефактов ML моделей, их версионирование.
- Запуск готовых ML моделей в виде самостоятельных сервисов (инференс).
- Автоматизация процессов с помощью ETL механизмов
- Гибкое управление вычислительными ресурсами.
- Командную работу пользователей.

Продукт состоит из отдельных модулей, которые предоставляют функциональные возможности для обеспечения конкретных бизнес-задач продукта.

Ниже приводятся функциональные возможности входящих в состав продукта модулей.

4.1 Модуль управления организационной структурой

Модуль управления организационной структурой предназначен для организации контекста рабочего пространства пользователей, разграничения доступа к ресурсам.

В рамках продукта модуль управления организационной структурой обеспечивает следующие основные функциональные возможности:

- Аутентификацию пользователей;
- авторизацию доступа к объектам продукта;
- управление пользовательскими пространствами;
- управление проектами;
- изоляцию ресурсов на уровне пространств;
- совместный доступ к ресурсам и процессам в контексте проектов и пространств.

Модуль не предоставляет независимый пользовательский интерфейс.

4.2 Модуль управления ресурсами

Модуль управления ресурсами предназначен для организации централизованных репозиториев для хранения артефактов наборов данных, ML моделей и прочих данных, а также доступа к ним пользователей.

В рамках продукта модуль управления ресурсами обеспечивает следующие основные функциональные возможности:

- Управление подключаемым хранилищами;
- Управление версионируемыми репозиториями наборов данных;
- Управление версионируемыми репозиториями ML моделей;
- Доступ к данным, размещенным в хранилищах и репозиториях.

Модуль не предоставляет независимый пользовательский интерфейс.

4.3 Модуль управления процессами

Модуль управления процессами предназначен для создания пользователем различных активных процессов, сервисов и приложений, а также для взаимодействия с ними.

В рамках продукта модуль управления процессами обеспечивает следующие основные функциональные возможности:

- Создание, запуск, остановку, взаимодействие с сервером Jupyter для разработки и отладки программных алгоритмов, пайплайнов обучения и применения ML моделей, анализа данных, тестирования гипотез и т.п.;
- Создание, запуск, остановку, взаимодействие с модельным сервисом (инференс ML модели) для проверки его работоспособности, демонстрации функционала бизнес-заказчику, а также для встраивания его в более сложные пайпланы и приложения;
- Создание, запуск, остановку, взаимодействие с сервером Airflow для построения и тестирования ETL процессов;
- Создание, запуск, остановку, взаимодействие с сервисом разметки данных Label studio.

Модуль не предоставляет независимый пользовательский интерфейс, однако некоторые пользовательские серверы могут предоставлять собственные API и GUI для взаимодействия с ними.

5 Интеграция

Интеграционное взаимодействие с внешними системами и между компонентами и модулями продукта определяется средствами используемого программного обеспечения.

Для интеграционного взаимодействия в продукте используются:

- обмен данными посредством API-интерфейсов через поддерживаемые протоколы:
  - GraphQL – GraphQL Schema Language;
  - REST – Swagger (OpenAPI);
  - gRPC – IDL (Interface Definition Language);
- обмен данными с помощью брокера сообщений Kafka;

Для пользовательского взаимодействия в продукте используется графический интерфейс пользователя.

6 Эксплуатационные характеристики

В главе приводится описание требований, которым соответствует продукт.

6.1 Требования к развертыванию

Продукт состоит из тесно интегрированных компонентов, которые могут разворачиваться в как функционально-замкнутой программной среде во внутреннем контуре организации, так и в приватном облаке инфраструктурного провайдера (T1, VK, Yandex и т.п.).

В качестве инструмента для оркестрации контейнеризированных приложений (автоматизации их развертывания, масштабирования и координации в условиях кластера) рекомендуется использовать Kubernetes.

При необходимости разработки и эксплуатации ML моделей, функционирующих с GPU, в целевом Kubernetes кластере должны присутствовать соответствующий узлы с GPU. Рекомендуется включать в состав кластера узлы с идентичной конфигурацией (количество CPU, оперативной памяти, параметрами GPU, жесткими дисками).

СУБД рекомендуется развертывать на отдельных узлах, не входящих в состав k8s кластера.

Производительность и стабильность работы продукта напрямую зависит от производительности компонентов инфраструктуры: количества и быстродействия CPU, объема оперативной памяти, объема и быстродействия жестких дисков, пропускной способности локальной сети.

Минимальные требования к платформе контейнеризации:

Платформа	Ядра CPU (шт.)	ОЗУ (ГБ)	HDD (Гб)
kubernetes	36	120	2000

6.2 Требования к программному обеспечению

В качестве инструмента для оркестрации контейнеризированных приложений (автоматизации их развертывания, масштабирования и координации в условиях кластера) рекомендуется использовать Kubernetes.

В продукте предъявляются следующие требования к системному и прикладному программному обеспечению:

Название	Версия	Лицензия
Kubernetes	Не ниже 1.27.5	Apache License 2.0
PostgreSQL	Не ниже 15.5	PostgreSQL License
Sonatype Nexus Repository OSS edition	3.63.0-01	Apache License 2.0

6.3 Требования к информационному обеспечению

Взаимодействие с GUI осуществляется с помощью WEB браузера с поддержкой HTML5.

Взаимодействие с GraphQL API компонентов продукта осуществляется с использованием протокола HTTPS.

6.4 Требования к ролевой модели и безопасности

Ролевая модель предусматривает три уровня пользовательских ролей.

	Тип роли	Назначение
1	Общесистемные роли	Роли, действие которых распространяется на весь продукт.
2	Роли пространства	Пользовательское пространство - изолированная организационная единица первого уровня. Роли уровня пространства имеют соответствующие полномочия только в контексте целевого пространства.
3	Роли проекта	Проект - изолированная организационная единица второго уровня, подчиненная пространству. Роли уровня проекта имеют соответствующие полномочия только в контексте целевого проекта.

Пользователю должны предоставляться минимально необходимые для выполнения функциональных обязанностей полномочия в соответствии с ролевой моделью, предусмотренной продуктом.

Для контроля доступа к операциям в модулях продукта должны использоваться следующие базовые роли:

	Наименование Роли	Тип роли	Описание доступных действий и полномочий
1	Администратор пространства	Роль уровня пространства	- Управление проектами в контексте пространства (создание, изменение, удаление) - Управление ресурсами дочерних проектов (назначение, отзыв ресурса, в т.ч. и доступных в пространстве GPU) - Полные права в контексте дочернего проекта (по аналогии с Администратором проекта)
2	Пользователь проектов	Роль уровня пространства	Пользователи с данной ролью представляют команду пространства, имеющую возможность получить конкретную роль уровня проекта в целевом проекте. Доступные действия в контексте пространства: - получение "только для чтения" параметров пространства - получение списков доступных ресурсов (хранилища, репозитории наборов данных и моделей).

3	Администратор проекта	Роль уровня проекта	Полные права в контексте целевого проекта: - полные права на управление ресурсами (хранилища, репозитории наборов данных и моделей). - полные права на управление пользовательскими серверами (JupyterLab, inference, Airflow, LabelStudio). - управление ролями пользователей проекта (добавление в команду проекта, изменение полномочий, удаление из команды проекта)
4	DS специалист	Роль уровня проекта	Создание репозиториев наборов данных и моделей, работа с их содержимым. Создание (удаление) серверов JupyterLab, Inference, Airflow, LabelStudio, запуск и работа с ними.