**Документация, содержащая описание функциональных характеристик модуля программного обеспечения**

**Комплекс модулей «Платформа Предикто»**

**г. Москва 2025**

[1. Комплекс модулей «Платформа Предикто» 5](#_abseosmffane)

[2. Модуль "Персонализация контента" 5](#_bmzakiowu4pk)

[3. Архитектура модуля «Персонализация контента» 7](#_fuxsqccpbuer)

[4. Схема работы 8](#_spk4gfqhsfnh)

[5. Используемые языки программирования и фреймворки 20](#_pf377y6aljju)

[6. Описание технической инфраструктуры 20](#_9nne77bsuqky)

[7. Эксплуатация системы 21](#_s1q4gko3eb1g)

[7. Предназначение ПО Модуль «Предсказание оттока» 22](#_rxiusmqobul1)

[8. Описание функциональных характеристик модуля «Предсказание оттока» 22](#_egs7c1ojio82)

[9. Архитектура модуля «Предсказание оттока» 22](#_mnsi0rqi02ae)

[10. Схема работы модуля «Предсказание оттока» 23](#_3idki31in34i)

[11. Используемые языки программирования и фреймворки модуля «Предсказание оттока» 25](#_x6192k3qe3ls)

[13. Предназначение Модуля «Сквозная аналитика» 26](#_jlzt7upfcou8)

[14. Описание функциональных характеристик Модуля «Сквозная аналитика» 27](#_nda5vu3rdfhw)

[15. Архитектура модуля «Сквозная аналитика» 28](#_4zfjkg8bc5vl)

[16. Схема работы модуля «Сквозная аналитика» 29](#_uho51fcfwn7f)

[17. Используемые языки программирования и фреймворки модуля «Сквозная аналитика» 31](#_782y9iezpz2a)

[18. Эксплуатация системы модуля «Сквозная аналитика» 32](#_pgk9f3vzxbem)

[19. Предназначение модуля «Инфраструктура для сервиса» 34](#_lusrlvgsjzi1)

[20. Описание функциональных характеристик «Инфраструктура для сервиса» 34](#_spnujurgr7sn)

[21. Архитектура модуля «Инфраструктура для сервиса» 36](#_4lci96se9rvx)

[22. Технические и эксплуатационные характеристики модуля «Инфраструктура для сервиса» 37](#_dqk5hqg9vulb)

[23. Используемые языки программирования и фреймворки модуля «Инфраструктура для сервиса» 38](#_zduagnqgakc5)

[24. Эксплуатация системы модуля «Инфраструктура для сервиса» 39](#_vtx7lfr5y39y)

[25. Предназначение модуля "BI и adhoc" 40](#_bdlmb3pycqj1)

[26. Описание функциональных характеристик модуля "BI и adhoc" 40](#_hfbpckpfo4ns)

[26.1 BI (Business Intelligence) 40](#_5tm7q2ib436y)

[26.2 AdHoc-отчеты 40](#_t4t1s1kbevaf)

[27. Архитектура модуля "BI и adhoc" 40](#_yp3nblpv45kh)

[28. Используемые языки программирования и фреймворки модуля "BI и adhoc" 41](#_c0ic8g81ze3o)

[Логическая модель данных (табличное представление и ER) и описание справочников 41](#_ertf5zre98ia)

[29. Схема работы модуля "BI и adhoc" 42](#_o42vgrgbcv4q)

[30. Эксплуатация системы модуля "BI и adhoc" 43](#_hndc1nd2vksj)

[31. Назначение модуля «Telegram бот» 44](#_u8baoeuj932l)

[32. Функциональные требования модуля «Telegram бот» 44](#_4mr2pyaam4qn)

[33. Архитектура модуля «Telegram бот» 44](#_616p9t2xihav)

[34. Схема работы модуля «Telegram бот» 45](#_tu49x9fx2dcs)

[35. Используемые языки программирования и фреймворки 45](#_f1km0ndxyaxb)

[Реализация 45](#_hvcmxrhql0p)

[36. Последовательность потоков данных модуля «Telegram бот» 45](#_l03dhxvdlbq6)

[37. Описание алгоритмов и математического обеспечения модуля «Telegram бот» 45](#_xqpcx5yilvap)

[38. Логическая модель данных и справочники модуля «Telegram бот» 45](#_tavnv3sj88j9)

[39. Роли модуля «Telegram бот» 46](#_7tissdisr173)

[40. Модуль «Модерация контента» 47](#_z7nixxlw7dmi)

[41. Назначение модуля «Модерация контента» 47](#_gpk62jcsynam)

[42. Функциональные возможности модуля «Модерация контента» 47](#_wqq80x4vfv8d)

[43. Архитектура и принцип работы модуля «Модерация контента» 47](#_qx5myiguucxn)

[Принцип работы 47](#_bukc3xp8kv2a)

[Структура входного и выходного файлов модуля «Модерация контента» 48](#_2penylrqmrip)

[44. Используемые языки программирования и фреймворки модуля «Модерация контента» 50](#_qhc22np3oh53)

[45. Возможные ошибки и решения модуля «Модерация контента» 50](#_xirqi35rhje0)

[46. Модуль «Сегментатор данных» 50](#_65bfswssbp75)

[47. Назначение модуля «Сегментатор данных» 50](#_3sut7fe6effz)

[48. Функциональные возможности модуля «Сегментатор данных» 51](#_16mt3crutyvo)

[49. Архитектура и принцип работы модуля «Сегментатор данных» 51](#_4ls06p6ko6i3)

[53. Роли и полномочия модуля «Сегментатор данных» 55](#_sfy1cgcbm2en)

[54. Используемые языки программирования и фреймворки модуля «Сегментатор данных» 55](#_c68imox5zlzs)

[55. Модуль «Тегизация» 56](#_uxwsgqfuxmj)

[56. Назначение модуля «Тегизация» 56](#_5iufcs7m0ltw)

[57. Функциональные возможности модуля «Тегизация» 56](#_l3gkj4f5h3f1)

[58. Архитектура и принцип работы модуля «Тегизация» 57](#_5tu2rn18qtwc)

[59. Используемые языки программирования и фреймворки модуля «Тегизация» 58](#_xc60nrr3rypl)

[- Модуль «Рекомендательная система для стриминговых сервисов» 58](#_cmm46i126d55)

[- Назначение модуля «Рекомендательная система для стриминговых сервисов» 58](#_ts3wpoedj4l9)

[- Функциональные возможности модуля «Рекомендательная система для стриминговых сервисов» 59](#_ev1x8s185mjq)

[Основные возможности модуля: 59](#_cpcu0etudbyl)

[60. Архитектура и принцип работы модуля «Рекомендательная система для стриминговых сервисов» 59](#_mzas2gsn8rlr)

[61. Используемые языки программирования и фреймворки модуля «Рекомендательная система для стриминговых сервисов» 61](#_c36rm2r1e9mu)

[62. Описание модуля «Поисковый модуль для стриминговых сервисов» 62](#_tphbz0quj79s)

[63. Описание функциональных характеристик модуля «Поисковый модуль для стриминговых сервисов» 62](#_ecs7jrprvcs9)

[64. Архитектура модуля «Поисковый модуль для стриминговых сервисов» 63](#_2fq93cn10bcb)

[65. Схема работы модуля «Поисковый модуль для стриминговых сервисов» 66](#_wi4xspqftz87)

[66. Модуль «BI аналитика для стриминговых сервисов» 68](#_r8e07vu1cxor)

[67. Назначение модуля «BI аналитика для стриминговых сервисов» 68](#_tnbbzl639ime)

[68. Функциональные возможности модуля «BI аналитика для стриминговых сервисов» 68](#_yimv833x4tph)

[69. Архитектура и принцип работы модуля «BI аналитика для стриминговых сервисов» 69](#_aeej687w2juc)

[70. Используемые языки программирования и фреймворки модуля «BI аналитика для стриминговых сервисов» 70](#_jp5kc83w4j3)

[71. Используемые языки программирования и фреймворки 71](#_qua85uxxs3ql)

[72. Описание технической инфраструктуры 72](#_s1jf7coecc1j)

# 

# 

# Комплекс модулей «Платформа Предикто»

Комплекс модулей «Платформа Предикто» предназначен для построения, эксплуатации и развития аналитических, рекомендательных и пользовательских цифровых сервисов на базе данных медиаплатформ. Он объединяет в себе средства сбора, хранения, обработки, анализа и визуализации пользовательских и контентных данных, а также решения для повышения удержания, персонализации взаимодействия и принятия бизнес-решений.

**Технические характеристики**

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Облачная среда | Yandex Cloud (или совместимая среда) |
| Оркестрация | Kubernetes (Yandex Managed Kubernetes) |
| Контейнеризация | Docker |
| СУБД | PostgreSQL, ClickHouse, Redis, Qdrant |
| Хранилища | Yandex Object Storage, S3-совместимые системы |
| Технологии ML | PyTorch, CatBoost, scikit-learn, XGBoost |
| Языки разработки | Python 3.8+, SQL, Bash |
| Протоколы взаимодействия | REST, gRPC, HTTPS, WebSocket |
| Средства мониторинга | Prometheus, Grafana, Loki, DataSphere |
| СУБД для логов и событий | ClickHouse, OpenSearch |
| BI/Adhoc аналитика | Apache Superset |

# 

# Модуль "Персонализация контента"

Модуль предназначено для обработки поступающих запросов и формирования персонализированных рекомендаций видеоконтента для пользователей онлайн-кинотеатра Premier. Основные задачи модуля:

* Сбор и анализ данных каталога контента и пользовательских действий;
* Расчет персонализированных подборок контента с применением алгоритмов машинного обучения;
* Передача сформированных рекомендаций через API в клиентские системы;
* Обеспечение высокой точности рекомендаций на основе истории просмотров, предпочтений и статистики популярности контента.

**Описание функциональных характеристик**

* Персонализированные предложения по контенту. Генерация индивидуальных рекомендаций на основе предпочтений пользователей и их истории просмотров. Размещение персонализированных рекомендаций на полке "Вам понравится".
* Управление профилем. Пользователи могут управлять своими профилями, корректировать предпочтения и получать соответствующие рекомендации.
* Алгоритмы рекомендаций с использованием методов машинного обучения - коллаборативная фильтрация (выявление схожих пользователей и их предпочтений) и контентная фильтрация (рекомендации на основе характеристик контента). Гибридные подходы (комбинация различных методов, включая ранжирование с использованием градиентного бустинга).
* Распределение видов запросов к системе:

а) Персонализированные (с учетом истории пользователя);

б) Групповые (на основе статистики просмотра);

с) "Холодные" (для пользователей без истории взаимодействий).

* Сбор и анализ данных о взаимодействиях пользователей (просмотры, клики, лайки), анализ состояния системы и логирование ошибок, анализ эффективности рекомендаций.

# Архитектура модуля «Персонализация контента»

Модуль представляет из себя сервис, который собирает данные и предоставляет рекомендации для формирования рекомендательных полок.

Модуль "Персонализация контента" реализует микросервисную архитектуру, обеспечивающую гибкость, масштабируемость и отказоустойчивость системы. Основные компоненты модуля:

Обработка данных – агрегирует данные о пользователях, истории просмотров и контенте из внешних источников;

ML-рекомендации – включает алгоритмы коллаборативной и контентной фильтрации, а также гибридные модели на основе глубокого обучения;

API – предоставляет интерфейсы взаимодействия с внешними сервисами для получения и передачи рекомендаций;

Хранение данных – отвечает за хранение пользовательских профилей, метаданных контента и предварительно рассчитанных рекомендаций;

Логирование – контроль за работой системы, анализирует ошибки и собирает метрики производительности.

Пользователи получают доступ к персонализированному контенту через интернет в мобильном приложении и веб-приложении, а также на смарт-телевизорах и других устройствах. Сами интерфейсные приложения (например, мобильное приложение) не входят в рамки разработки Модуля.

Модель видеоконтента (тайтла) в рекомендательной системе представляет собой сущность, обладающую следующими атрибутами:

* Идентификатор – числовой код;
* Название – текстовое наименование. Для сериалов или шоу используется наименование всей франшизы, например «Реальные пацаны»;
* Тип видеоконтента – фильм, сериал, шоу;
* Возрастные ограничения (возрастной рейтинг) - 0+, 6+, 12+, 16+, 18+;
* Принадлежность к жанрам;
* Главный жанр;
* Географическая доступность;
* Категорийные признаки контента;
* Возможность просмотра без наличия платной подписки.

Запрос к рекомендательной системе определяет характер и ограничения для рекомендаций. Запрос может содержать следующую информацию:

* Тип рекомендации;
* Идентификатор пользователя;
* Возрастные ограничения;
* Географические ограничения;
* Категории контента;
* Жанровые ограничения.

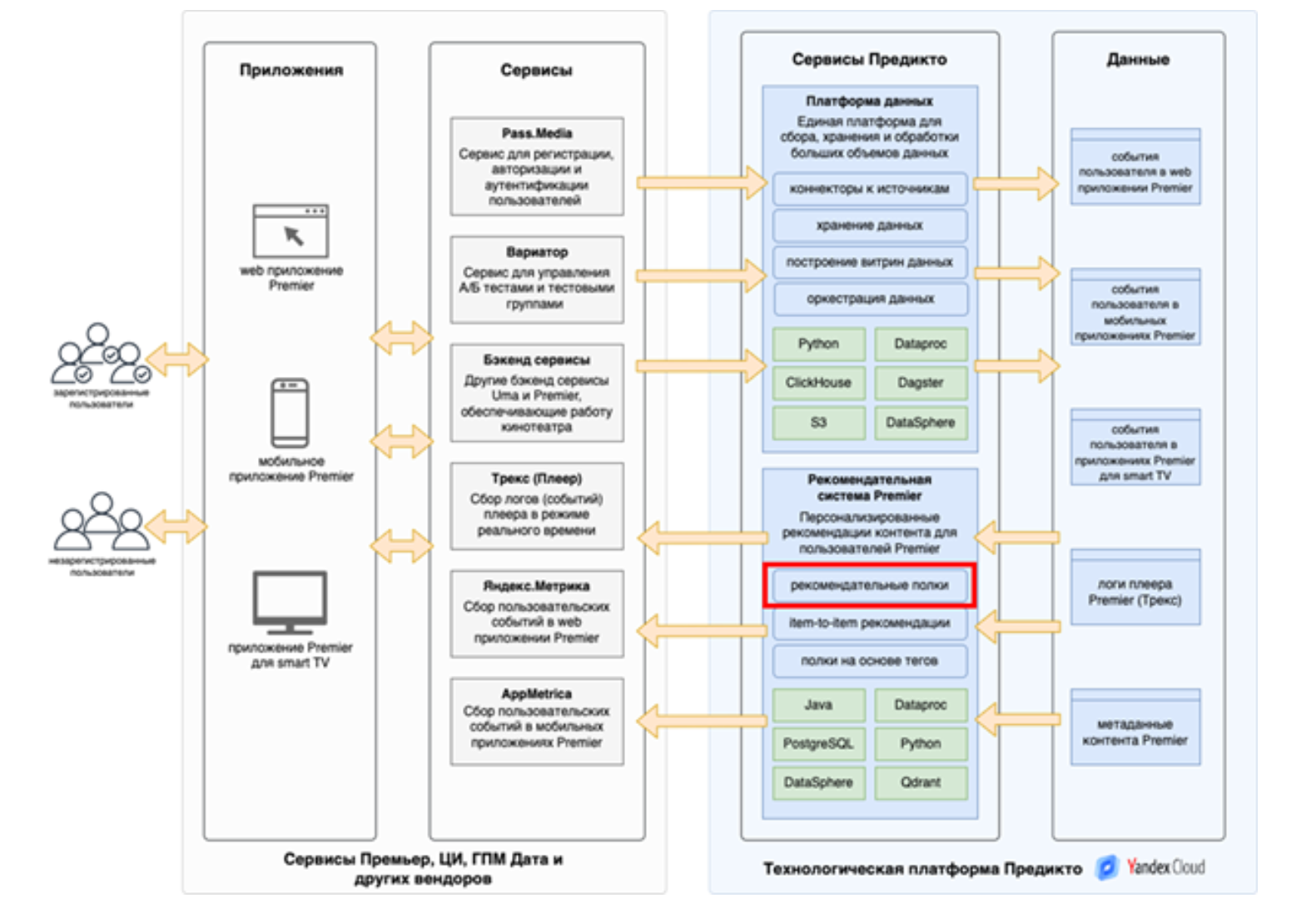


Рис. 1 Архитектурная схема модуля «Персонализация контента»

# Схема работы

Ролевая модель подразумевает две роли:

***Зарегистрированный пользователь*** - пользователь, прошедший регистрацию и авторизацию. Имеет доступ к контенту в соответствии со своей подпиской. Получает персонализированные рекомендации при наличии достаточной истории просмотров в сервисе. Зарегистрированный пользователь идентифицируется по ID;

***Незарегистрированный пользователь*** - пользователь, не прошедший регистрацию и/или авторизацию. Таким пользователям доступна только часть контента (например, раздел «Бесплатное»). Незарегистрированные пользователи не получают персональные рекомендации.

Вне зависимости от типа запроса и выполняемого вида рекомендации все рекомендательные функции могут быть сведены к следующим основным этапам:

1. Анализ и обработка рекомендательного запроса;
2. Отбор кандидатов;
3. Фильтрация;
4. Ранжирование;
5. Формирование ответа

1. Анализ и обработка рекомендательного запроса

Сервис рекомендаций получает запросы от внешних систем, в частности от сервисов Uma.Tech. Каждый запрос проверен на корректность. В случае валидности запроса должна быть вызвана функция формирования рекомендаций соответствующего типа с применением ограничений, указанных в запросе. В случае некорректного запроса сервис должен возвращать сообщение о ошибке вызывающей стороне.



*Рис. 1 - Алгоритм анализа и обработка рекомендательного запроса*

1. Отбор кандидатов

При отборе кандидатов для формирования рекомендаций пользователя идентифицируется на основании ID пользователя из запроса рекомендаций.

Возможны три ситуации:

«холодный» пользователь - пользователь не указан в запросе. В этом случае некоторые типы рекомендательных пайплайнов могут формировать так называемые «холодные рекомендации», если они заложены в рекомендательный сервис. Так как информация о пользователе полностью отсутствует, то фильтрация задействуется только на основе ограничений запроса.

«теплый» пользователь - пользователь указан, но для него в системе присутствует только часть необходимой информации и персональные рекомендации еще не были предрасчитаны. В этой ситуации, при наличии необходимой информации, часть пайплайнов могут выполнить подбор кандидатов либо использовать холодные рекомендации, но накладывать на этапе фильтрации и ранжирования все возможные типы ограничений или предпочтений, известные на текущий момент.

«горячий» пользователь – пользователь, о котором сервис уже имеет достаточно информации для формирования полноценных персональных рекомендаций.

1. Фильтрация

В каждом типе рекомендаций может формироваться свой набор фильтров. В общем случае для работы сервиса требуются следующие виды фильтров:

* фильтр «включенных» тайтлов фильтр бесплатных тайтлов фильтр возрастного ограничения фильтр жанра
* фильтр географической доступности
* фильтр категорийных признаков
* фильтр пересечения с другим типом рекомендаций фильтр просмотренных тайтлов

Фильтр 1

Запрос рекомендаций

Построение цепочки фильтров

Список кандидатов

Список кандидатов

для ранжирования

Фильтр 2

Фильтр N

Блок фильтрации

*Рис. 2 - Алгоритм фильтрации кандидатов*

1. Ранжирование

В зависимости от типа рекомендательного пайплайна отфильтрованные кандидаты могут быть переупорядочены по различным критериям. Итоговый порядок кандидатов должен отражать максимальный интерес для пользователя с точки зрения Сервиса.

Некоторые наборы кандидатов могут быть уже заранее отсортированы по предпочтениям пользователей, в этом случае при ранжирование используются только те признаки, которые не учтены в начальном наборе.

Ранжирование может выполняться по следующим критериям:

* Сортировка на основе векторной близости векторов признаков (embeddings) для пользователя и кандидата. Чем ближе оказываются вектора, тем более подходящим для пользователя должен оказаться контент;
* Сортировка на основе новизны контента. Новые тайтлы более вероятно окажутся интересны пользователю, чем те которые он уже просматривал или видел в рекомендациях;
* Сортировки на основе специальных бизнес-правил. Например, целевое поднятие тайтлов в рейтинге.

Финальным этапом в блоке ранжирования может выступать блок ограничения – лимиттер. В этом блоке отсекаются кандидаты, если существует ограничение на максимальное количество кандидатов, либо формируется пустой список кандидатов, если не удалось подобрать минимально необходимое их число.

Сортировка 1

Запрос рекомендаций

Построение цепочки ранжирования

Список кандидатов

Список кандидатов

для ранжирования

Сортировка 2

Сортировка N

Блок ранжирования

Лимиттер

*Рис. 3 - Алгоритм ранжирования кандидатов*

1. Формирование ответа

Генерация рекомендаций для разных контентных полок осуществляется различными пайплайнами согласно следующим принципам:

1. Полка главной страницы «Популярное» формируется двухуровневой моделью рекомендаций, в которой кандидатными моделями первого уровня выступают:

ALS, LightFM, BPR, строящиеся на матрице взаимодействия, ячейки которой заполняются на основе данных об успешности просмотров фильмов и сериалов (для фильмов такая оценка выводится из доли просмотра, для сериала – на основе работы отдельной модели, предсказывающей вероятность начала просмотра следующего сезона). При заполнении ячеек матрицы применяется дисконтирование по времени;

аналогичная ALS, ограниченная бесплатными тайтлами;

контентная модель, определяющая близость текстовых embeddings описаний тайтлов к описаниям успешно просмотренных пользователем;

DSSM-модель с пользовательскими и контентными признаками, обученная на успешности просмотров.

Помимо этих моделей, отдельно обучаются:

ALS на жанрах и возрастных ограничениях для получения векторных представлений этих сущностей и пользователей;

ALS на матрице, ячейки которой заполняются данными о соотношении кликов по постерам и их показам (impressions), для учета персонального интереса к постеру и обеспечения динамичности рекомендаций.

Выборка кандидатов от перечисленных моделей ранжируется моделью градиентного бустинга, обучаемой на гибридный просмотрово-кликовый регрессионный таргет с фичами, полученными из всех полученных векторных представлений (как меры близости), фичами популярности, новизны, контентными фичами тайтлов.

При обучении и валидации кандидатных моделей и итогового ранжирования используется хронологическое разбиение сформированного датасета во избежание data leaks.

2. Полка главной страницы «Вам понравится» формируется секвенциональной трансформерной моделью типа XLNet, обученной на предсказании следующего тайтла в последовательности по событиям просмотров и кликов на постеры. В итоговом ранжировании используются рейтинги (IMDB, Кинопоиск) и предсказание индивидуальной вероятности предпочтения от ранжируюшей модели полки «Популярное».

3. Холодные бейзлайны, показываемые на упомянутых выше полках пользователям без достаточной истории или без авторизации, формируются на основе популярных в последнем периоде и трендовых (резко набирающих популярность тайтлов) таким образом.

4. Из полок отфильтровываются пересечения между собой и находящимися выше главными редакторскими. Фильтруются также ранее просмотренные тайтлы.

5. Жанровые полки формируются засчет отбора тайтлов нужного жанра из следующих после отображаемых тайтлов «Вам понравится».

6. Полка «Похожее на X» формируется на основе выбора якорного тайтла из последних наиболее успешно просмотренных пользователем, и последующего поиска ближайших соседей в векторной БД Qdrant. Аналогичным образом формируются и рекомендации после просмотра в плеере (с учетом естественной логики порядка серий сериалов).

7. Переранжирование редакторских полок ведется сервисом, работающим в онлайн-режиме на основе модели градиентного бустинга из полки «Популярное», фичи для которой предрасчитываются, в частности, на основе векторных представлений кандидатных моделей и записываются в БД Postgre.

Потоки данных включая протоколы обмена и сценарии, включая альтернативные ветвления, с указанием акторов взаимодействий в виде диаграмм DFD, sequence.



*Рис. 4 Диаграмма DFD организации потоков данных от источников до сервиса API*

Формат хранения данных в витрине с информацией о просмотрах, непосредственно используемой для генерации рекомендаций, приведен в следующей таблице 1:

Таблица 1

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Пример** | **Описание** |
| passport\_id | --5DvmO6RhSsUnD-uyIXa | Passmedia ID пользователя |
| tv\_id | 20834 | tv\_id тайтла |
| season\_num | 1 | Номер сезона сериала, для фильмов 1 |
| episode\_num | 5 | Номер эпизода сериала, для фильмов 1 |
| user\_ep\_total\_view\_duaration\_s | 2629 | Время просмотра эпизода в секундах |
| user\_ep\_first\_view\_dt | 2023-12-11 19:22:33 | Старт просмотра эпизода |
| pdp\_created\_at | 2023-12-12 09:04:36.381263 | Время создания записи |
| pdp\_updated\_at | 2023-12-13 09:04:28.11316 | Время актуализации записи |
| actual\_title\_ru | Последователи | Название тайтла |
| season\_duration\_s | 16769 | Общая продолжительность эпизода в секундах |
| user\_seas\_total\_view\_duarat | 14557 | Суммарное время просмотра сезона в секундах |
| ion\_s | 0.8680899278430437 | Доля просмотра сезона |
| user\_ep\_seconds\_from\_origin\_median | 10705105 | Медианное время в секундах с выхода эпизода до просмотра |
| user\_ep\_seconds\_from\_origi | 2642 | Медианное время в секундах между просмотрами эпизодов в сезоне |
| has\_next\_season | false | Начал ли пользователь просмотр следующего сезона |
| user\_nextseas\_percent\_view | 0.8680899278430437 | Процент просмотра пользователем следующего сезона |
| ed actual\_tvtype\_name | series | Тип тайтла (фильм, сериал, шоу, трансляция) |

Таблица 2 - Поля витрины с информацией о метаданных каталога

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| absolute\_url |  |
| accessibility |  |
| age\_restriction |  |
| can\_subscribe |  |
| content |  |
| contenttvstype |  |
| countries |  |
| description |  |
| external\_ids |  |
| genres |  |
| global\_release |  |
| has\_allow\_download |  |
| id |  |
| is\_active |  |
| keywords |  |
| labels |  |
| last\_updated\_ts |  |
| last\_video\_add\_ts |  |
| local\_release |  |
| name |  |
| original\_title |  |
| picture |  |
| poster\_url |  |
| provider |  |
| related\_showcase |  |
| restriction\_notices |  |
| sensitive\_content |  |
| slogan |  |
| slug |  |
| smoking\_restriction |  |
| studios |  |
| type |  |
| uniform\_url |  |
| year |  |
| year\_end |  |
| year\_start |  |
| persons |  |
| studio |  |
| rightholder |  |
| is\_original |  |
| geo\_tag |  |
| available\_to\_geo\_RU |  |
| available\_to\_geo\_WW |  |
| available\_to\_age\_0 |  |
| available\_to\_age\_6 |  |
| available\_to\_age\_12 |  |
| available\_to\_age\_16 |  |
| available\_to\_age\_18 |  |
| is\_erotics |  |
| is\_test |  |
| item\_id |  |
| genre\_ids |  |
| country\_ids |  |

Таблица 3 - Поля таблицы с информацией о кликах из Яндекс.Метрики

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| watchID |  |
| deviceCategory |  |
| counterID |  |
| date |  |
| dateTime |  |
| URL |  |
| clientID |  |
| lastTrafficSource |  |
| regionCountry |  |
| cid |  |
| session\_id |  |
| user\_id |  |
| user\_auth |  |
| user\_profile\_id |  |
| screen\_name |  |
| event\_group |  |
| event\_name |  |
| event\_category |  |
| event\_action |  |
| event\_label |  |
| event\_context |  |
| event\_json |  |
| user\_age\_category |  |
| channel\_id |  |
| content\_id |  |
| content\_category\_id |  |
| content\_subcategory\_id |  |
| content\_list\_position |  |
| content\_list\_id |  |
| content\_type |  |
| ab\_name |  |
| ab\_variant |  |
| user\_subscription |  |
| user\_subscription\_id |  |
| touch\_point |  |

Таблица 4 - Поля таблицы с информацией о показах постеров (impressions) из Яндекс.Метрики

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| date |  |
| clientID |  |
| impression\_id |  |
| url |  |
| sess\_num |  |
| json\_pass\_id |  |
| authorized |  |
| country |  |
| json\_object\_id |  |
| json\_response\_model\_tag |  |
| json\_items\_id |  |
| ua\_device\_type |  |

# Используемые языки программирования и фреймворки

|  |  |
| --- | --- |
| **Часть Сервиса** | **Языки программирования и фреймворки** |
| Data Engineering | Python PySpark Clickhouse Postgresql Dagster |
| Machine Learning | Python (implicit, RecTools, PyTorch, CatBoost) MlFlow |
| Backend | Java, Postgresql |
| DevOps | Terragrunt, Helm, GitOps, ArgoCD, Yandex Cloud |

# Описание технической инфраструктуры

Основные компоненты инфраструктуры:

Облачная платформа: Yandex Cloud

Система виртуализации: Yandex Compute Cloud

Контейнеризация: Docker и Kubernetes для управления развертыванием сервисов

Балансировка нагрузки: Yandex Application Load Balancer и Yandex Network Load Balancer

Облачное хранилище: Yandex Object Storage для хранения данных и моделей

База данных: Yandex Managed Service for PostgreSQL, Redis (в контейнерах) и Qdrant для векторного поиска

Платформа машинного обучения: Yandex Data Sphere для обработки ML-моделей

Логирование:

Prometheus – сбор метрик производительности

Grafana – визуализация и анализ данных

Loki – сбор и анализ логов

Tempo – распределенный трейсинг

CI/CD: ArgoCD, Terragrunt и Helm для автоматического развертывания

Средства безопасности: Аутентификация и авторизация пользователей, защита API от SQL-инъекций и XSS-атак, шифрование данных при передаче и хранении

Механизмы отказоустойчивости: Автоматический перезапуск сервисов при сбоях, горизонтальное масштабирование, ежедневное резервное копирование баз данных

# 7. Эксплуатация системы

7.1 Запуск системы

Развертывание инфраструктуры через Terraform и Helm. Запуск контейнеров с использованием Kubernetes. Проверка доступности API и БД.

7.2 Управление

Управление параметрами и конфигурацией осуществляется через DevOps-инструменты (ArgoCD, GitOps). Настройки системы хранятся в конфигурационных файлах в Yandex Cloud.

7.3 Резервное копирование и восстановление

Автоматическое резервное копирование БД раз в сутки.

Возможность восстановления данных в течение 1 часа.

5.4 Обновление

Обновления ПО разворачиваются через CI/CD. Обновления моделей машинного обучения выполняются по заранее определенному графику.

# Предназначение ПО Модуль «Предсказание оттока»

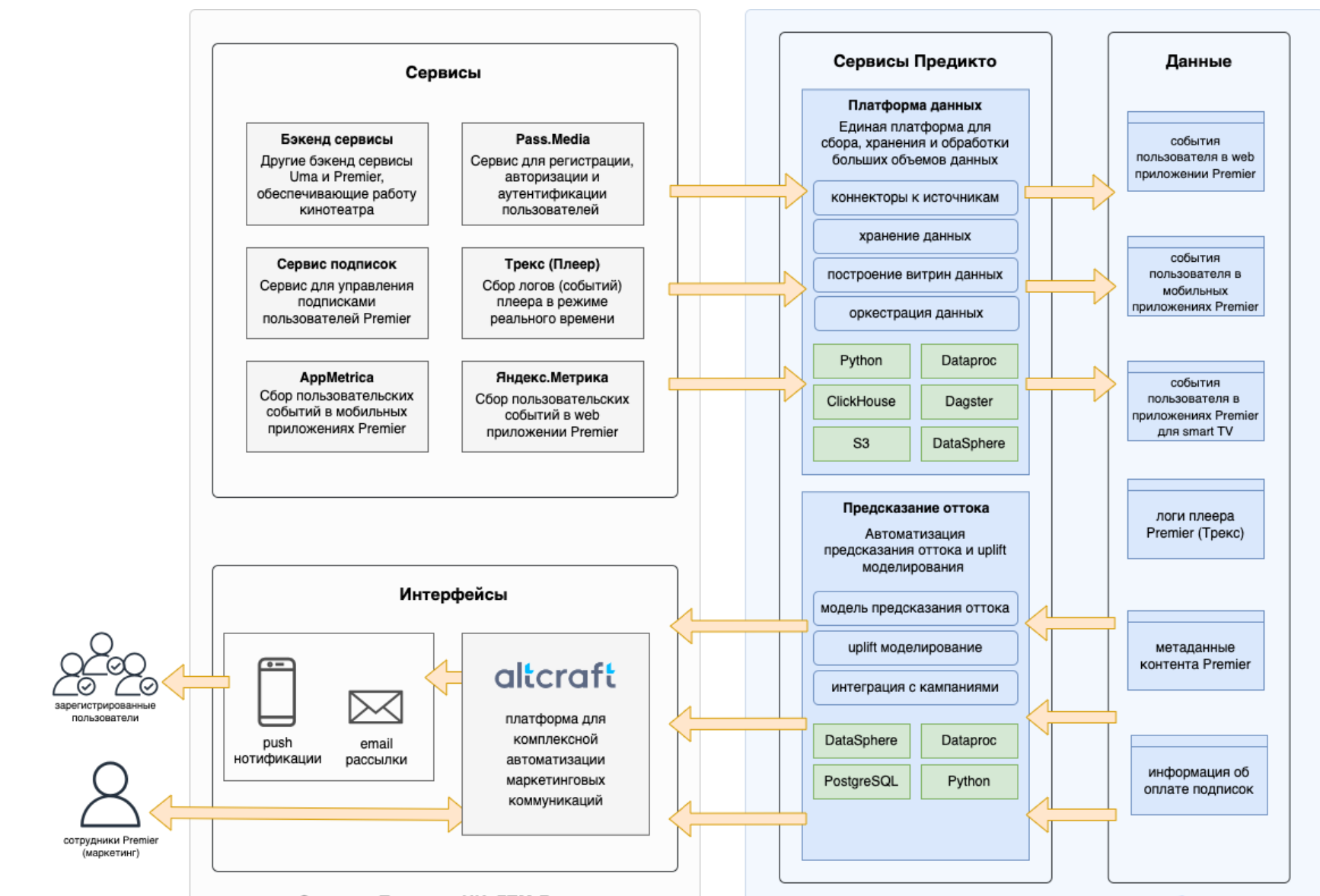
Модуль "Предсказание оттока" предназначен для заблаговременного определения вероятности оттока подписчиков. Это позволяет эффективно выстраивать коммуникацию с пользователями с целью их удержания и повышения бизнес-метрик Заказчика.

# Описание функциональных характеристик модуля «Предсказание оттока»

Модуль предоставляет вероятность оттока по каждому пользователю (скоринг пользователей).

* В основе продукта - ML-модели предсказания оттока.
* Прогнозирование вероятности оттока в течение 30 дней + 2 дня на "возврат".
* Отдельные модели для триалистов и пользователей с платной подпиской.
* ROC-AUC моделей не ниже 0.7.
* Финальный результат - витрина с вероятностями оттока, обновляемая ежедневно.
* Инструмент мониторинга качества модели.
* Интерпретация результатов с возможностью определения ключевых атрибутов.
* Используемые данные:
  + Биллинг
  + Статистика пользования сервисом
  + Поведение пользователей
  + Ключевые характеристики пользователей
* Информация для рекомендаций оптимальных каналов взаимодействия с пользователем и ценностных предложений.

# Архитектура модуля «Предсказание оттока»



*Рис. 1 Диаграмма концептуальной архитектуры модуля*

Функциональность: предсказание оттока, интеграция с маркетинговыми инструментами.

Технологический стек: Yandex.Облако (Data Sphere, Dataproc), Python, PostgreSQL

Перечень сервисов, явно или неявно являющихся источником данных для модуля «Предсказание оттока»:

Pass.Media - cервис для регистрации, авторизации и аутентификации пользователей Премьер;

Бэкенд сервисы - различные бэкенд сервисы, обеспечивающие работу кинотеатра;

Сервис подписок - cервис для управления подписками пользователей, содержащий информацию об оплатах подписки;

Трекс (Плеер) - cбор логов (событий) плеера в режиме реального времени

AppMetrica - cервис для сбора пользовательских событий в мобильных приложениях;

Яндекс.Метрика - cервис для сбора пользовательских событий в web приложении.

# Схема работы модуля «Предсказание оттока»

Источники данных:

|  |  |
| --- | --- |
| Яндекс Метрика | События пользователя в web приложении |
| АппМетрика | События пользователя в мобильных приложениях |
| Trex | События пользователя в приложениях для smart TV |
| Player/Trex | Логи плеера (Трекс) |
| Бэкенд сервисы/Каталог контента | Метаданные контента |
| Бэкенд сервисы/Сервис подписок | Информация об оплате подписок (биллинг) |

Ролевая модель

|  |  |
| --- | --- |
| Администраторы | разработка и эксплуатация модуля. |
| Сотрудники партнеров | работа с маркетинговыми кампаниями через Altcraft |
| Пользователи | получение персонализированной маркетинговой коммуникации |

Все данные попадают в **Платформу данных** которая собирает информацию из всех источников. Сохраняет и упорядочивает её. Создаёт отчёты и прогнозы.

Технологии: Python, ClickHouse, S3, DataSphere, Dataproc, Dagster.

На основе собранных данных модель машинного обучения предсказывает, кто может отказаться от подписки

Использует разные факторы такие как просмотры, платежи, активность. Строит прогноз для каждого пользователя. Помогает выбрать лучший способ удержания (например, скидки или персональные рекомендации).

Технологии: DataSphere, Dataproc, PostgreSQL, Python.

Предсказания передаются в Altcraft – систему для маркетинга.

Если пользователь рискует уйти, ему можно отправить push-уведомление или письмо с персональным предложением.

**Идентификация, авторизация и аутентификация пользователей**

Администраторы - Сотрудники, отвечающие за разработку и эксплуатацию продукта. Используют инструменты разработки ПО и консоль Яндекс. Облака Идентификация, авторизация и аутентификация таких сотрудников происходит на основании аккаунтов Яндекс

Сотрудники (и/или подрядчики, парнтеры и т.д.) не взаимодействуют напрямую с модулем «Предсказание оттока». Они имеют возможность работать с результатами работы модуля через платформу маркетинговой коммуникации Altcraft, данные в которую передаются из модуля «Предсказание оттока». Сама платформа Altcraft не является частью модуля «Предсказание оттока»

Пользователи. Персонализированный скоринг вероятности оттока получают зарегистрированные пользователи при наличии достаточного количества информации о их действиях. Идентификация зарегистрированных пользователей для предсказания оттока происходит на основании ID. Дальнейшее маркетинговое взаимодействие с пользователями (на основе результатов работы модуля) происходит через инструменты платформы Altcraft

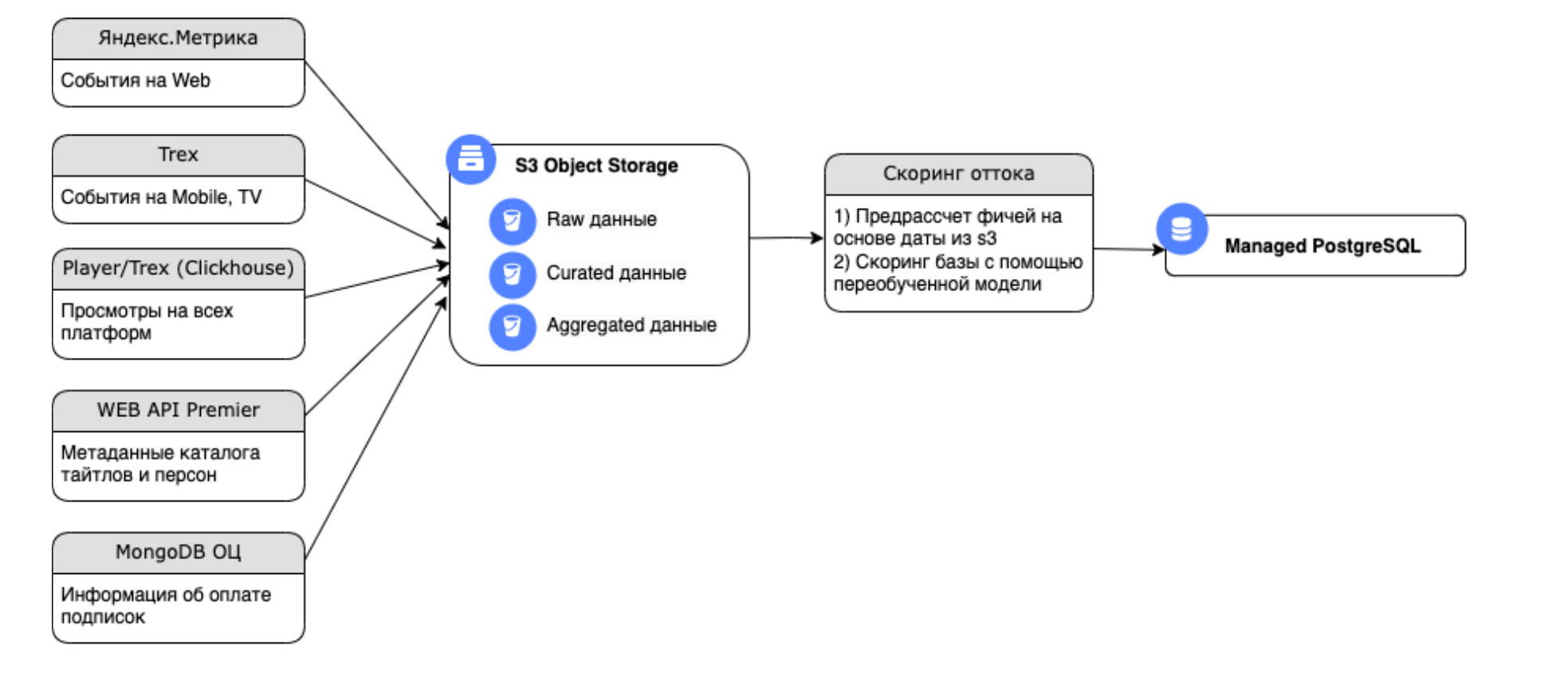
Пайплайн скоринга клиентов осуществляется в 3 фазы:

Фаза предрасчета фичей (Apache Spark):

* Биллинг
* История просмотров
* Взаимодействие с web-сервисом (Яндекс.Метрика, Trex)

Фаза переобучения модели (градиентный бустинг, временная кросс-валидация, учет сезонных факторов).

Фаза обновления данных (выгрузка вероятностей оттока в PostgreSQL, мониторинг качества предсказаний).

**

*Рис. 2 Потоки данных от источников до сервиса API рекомендаций организованы в соответствии со следующей диаграммой DFD*

# Используемые языки программирования и фреймворки модуля «Предсказание оттока»

|  |  |
| --- | --- |
| Языки программирования | Python |
| Фреймворки и библиотеки | ML-библиотеки: CatBoost, Scikit-learn, XGBoost |
| Обработка данных | Pandas, NumPy |
| Хранилища данных | PostgreSQL, S3 (Yandex.Cloud) |
| Оркестрация | Apache Airflow, Dagster |
| Облачные сервисы | Yandex DataSphere, Dataproc |

1. **Эксплуатация системы модуля «Предсказание оттока»**

Администрирование компонентов

PostgreSQL - Регулярные обновления и мониторинг производительности.Настройка резервного копирования и восстановления. Оптимизация запросов и индексов для улучшения скорости обработки данных.

Data Sphere - Управление вычислительными ресурсами. Мониторинг выполнения задач машинного обучения. Настройка окружения для ML-разработки.

S3 бакет - Управление правами доступа. Контроль версий и управление архивами данных. Мониторинг использования и оптимизация хранилища.

Внешние интеграции

Altcraft - Передача данных о предсказанном оттоке для настройки маркетинговых кампаний. Обратная связь о результатах взаимодействия с пользователями.

PostgreSQL - Интеграция с аналитическими сервисами для построения BI-отчетов. Использование SQL-запросов для получения агрегированной информации о пользователях.

Yandex.Cloud - Автоматизированное масштабирование инфраструктуры. Настройка логирования и мониторинга через встроенные инструменты Yandex.Cloud.

# Предназначение Модуля «Сквозная аналитика»

Модуль «Сквозная аналитика» предназначен для комплексного сбора, обработки и визуализации данных о пользовательском поведении на медиаплатформе. Он позволяет анализировать эффективность маркетинговых каналов, отслеживать путь пользователя от первого касания до целевого действия, а также оптимизировать стратегию контентного и продуктового развития.

Модуль интегрируется с «Платформой данных» и модулем «Сегментатор данных», система поддерживает сегментацию аудитории, персонализированную аналитику и автоматизированную отчетность. Основные возможности модуля включают построение карт клиентского пути (CJM), анализ предпочтений пользователей, мониторинг подписок, оценку рентабельности контента и выявление ключевых факторов влияния на конверсию.

Модуль «Сквозная аналитика» помогает медиаплатформам повышать эффективность взаимодействия с аудиторией, улучшать качество контента и оптимизировать маркетинговые расходы, предоставляя инструменты для принятия решений на основе данных.

# Описание функциональных характеристик Модуля «Сквозная аналитика»

# 

14.1. Аналитика и визуализация данных

Анализ поведения пользователей: отслеживание действий, построение карт клиентских путей (CJM) и выявление поведенческих паттернов.

Аналитика пользовательских действий с разбивкой по сегментам и когортах.

Создание интерактивных дашбордов для мониторинга ключевых метрик (продажи, конверсии, маркетинговые расходы и др.).

Формирование аналитического профиля пользователя на основе истории его взаимодействий с медиаплатформой.

14.2. Оптимизация и поддержка бизнес-процессов

Оптимизация рекламных каналов и маркетинговых кампаний на основе данных.

Анализ популярности контента для корректировки стратегии закупок и производства собственного контента.

Мониторинг активности подписок для повышения эффективности стратегий удержания и привлечения клиентов.

14.3. Работа с сегментами данных

Разделение пользовательской базы на сегменты с предоставлением аналитики по ним.

Экспорт сформированных сегментов в CSV-файл с идентификаторами пользователей и их атрибутами.

Передача данных во внешние системы через API.

Возможность гибкого формирования новых сегментов с их периодическим обновлением.

14.4. Интерфейсы доступа

Доступ к аналитическим отчетам и инструментам сегментации через удобный интерфейс.

Интуитивно понятный пользовательский интерфейс с четкой навигацией, описаниями и возможностью настройки.

Управление правами доступа для безопасной работы нескольких пользователей одновременно.

14.5. Сбор и интеграция данных

Интеграция информации из различных источников, включая веб-аналитику (Yandex Metrica, AppMetrica), биллинговые системы, медиаплееры и другие сервисы.

Хранение агрегированных данных в масштабируемой базе с учетом высоких требований к производительности.

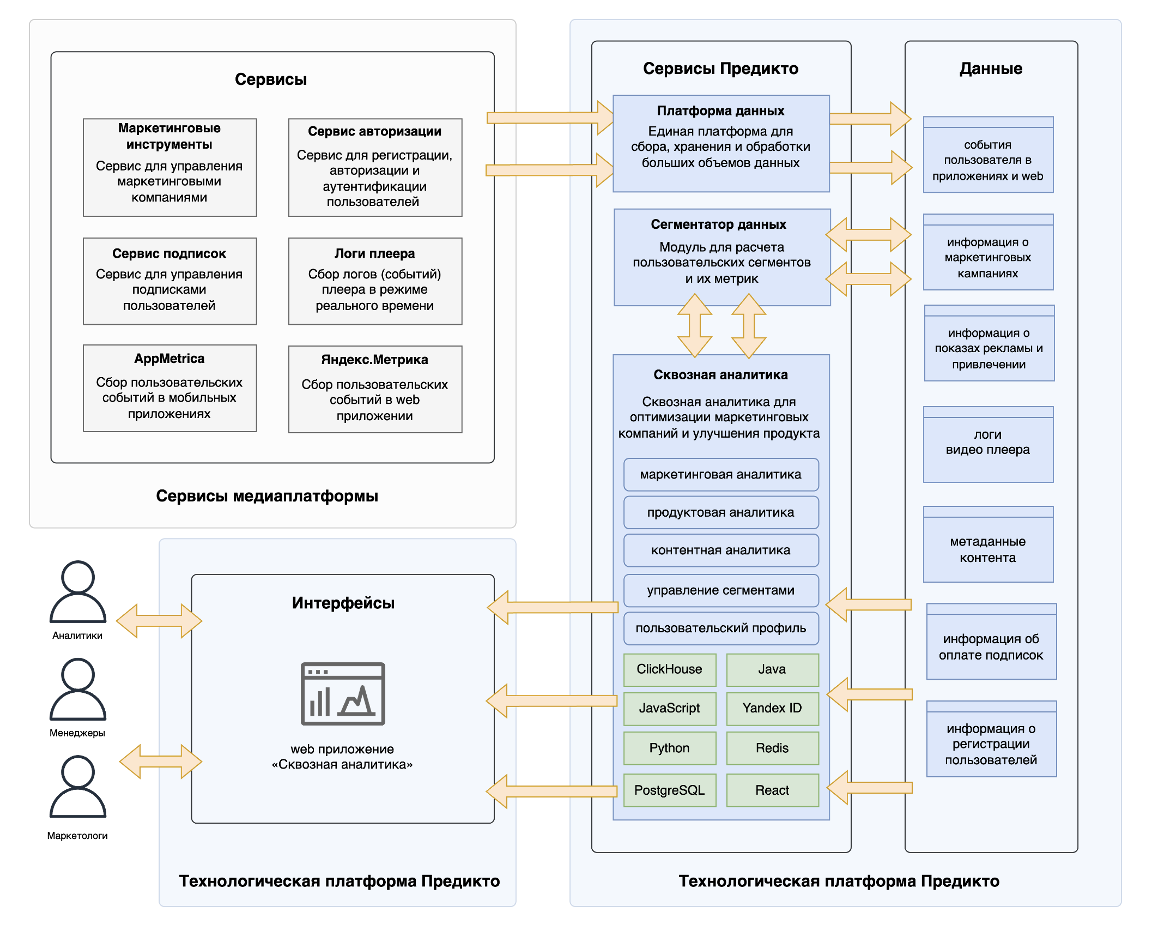
Реализация ETL-процессов (извлечение, трансформация, загрузка) для обеспечения чистоты и консистентности данных.

# Архитектура модуля «Сквозная аналитика»

Модуль «Сквозная аналитика» отвечает за обработку и анализ данных, связанных с маркетинговыми кампаниями, продуктами и контентом. Этот модуль использует несколько источников данных, включая информацию о пользовательских событиях из приложений и веб-сайтов, а также данные о метках и показателях рекламы. Внешние сервисы медиаплатформы, такие как сервисы для авторизации, подписки и сбора пользовательских данных, например, AppMetrica и Яндекс.Метрика предоставляют информацию, которая используется для анализа пользовательских сегментов и создания отчетности.

Платформа данных — это единая среда для хранения и обработки больших объемов информации, которая соединяет модули платформы. Модуль «Сегментатор данных» позволяет выделять пользовательские сегменты для анализа, а модуль «сквозная аналитика» обрабатывает и оптимизирует маркетинговые и продуктовые данные. Вся информация взаимодействует через интерфейсы, предназначенные для аналитиков, маркетологов и менеджеров, которые используют веб-приложение для работы с данными и получения аналитических отчетов.

Технологическая часть платформы включает базы данных, такие как PostgreSQL и ClickHouse, а также различные языки программирования и фреймворки, такие как Python, Java, React, Redis и JavaScript.



*Рис.1 Концептуальная архитектура модуля «Сквозная аналитика»*

# Схема работы модуля «Сквозная аналитика»

Сбор данных

Модуль получает данные из различных внешних источников, подключённых к медиаплатформе.

Плеерные логи: информация о просмотрах, перемотках, паузах и досмотрах.

AppMetrica и Яндекс.Метрика: действия пользователей в мобильных и веб-приложениях.

Сервис авторизации: данные о регистрации и входе пользователей.

Сервис подписок: сведения об оплатах, продлениях, отменах подписок.

Маркетинговые инструменты: информация о каналах привлечения, кампаниях, бюджетах и показах рекламы.

Интеграция данных

Поступающие данные агрегируются и обрабатываются в модуле «Платформа данных», где:

* осуществляется ETL-процесс (извлечение, трансформация, загрузка);
* данные хранятся в масштабируемой базе (ClickHouse, PostgreSQL);
* очищаются, нормализуются и подготавливаются для аналитики.

Расчёт сегментов пользователей

С помощью модуля «Сегментатор данных», интегрированного с «Платформой данных», производится:

* выделение пользовательских сегментов по заданным критериям (возраст, активность, география и т. д.);
* обновление сегментов в реальном времени;
* экспорт сегментов в CSV и через API.

Обработка данных в модуле «Сквозная аналитика». На этом этапе модуль

* анализирует поведение пользователей;
* строит карты путей клиентов (Customer Journey Map);
* рассчитывает метрики (MAU, DAU, ROMI, удержание, досмотры и пр.);
* объединяет данные по различным источникам и каналам.

Визуализация данных

Пользователи получают доступ к данным через настраиваемый веб-интерфейс.

Интерактивные дашборды по продуктовой, маркетинговой и контентной аналитике. Метрики активности, удержания, трафика, подписок. Дашборды на уровне сегментов и отдельных пользователей. Инструменты фильтрации и сравнения сегментов.

# Используемые языки программирования и фреймворки модуля «Сквозная аналитика»

Java (Backend)

Язык: Java (OpenJDK)

Фреймворки и библиотеки:

Spring Framework / Spring Boot

FlywayDB (миграции БД)

clickhouse-jdbc (интерфейс к ClickHouse)

spring-boot-starter-graphql (GraphQL API)

postgresql (драйвер для PostgreSQL)

hibernate-validator (валидация)

java-telegram-bot-api (интеграция с Telegram)

JavaScript / TypeScript (Frontend)

Языки: JavaScript, TypeScript

Фреймворки и библиотеки:

React (основа UI)

Redux / @reduxjs/toolkit (управление состоянием)

styled-components (стилизация компонентов)

axios (работа с HTTP-запросами)

@apollo/client (GraphQL клиент)

primereact, rc-tooltip, recharts (визуальные компоненты и графики)

formik (формы), history, gqlmin, graphql

webpack, regenerator-runtime, web-vitals

Python (обработка данных, ETL, интеграция)

Язык: Python

Библиотеки и фреймворки:

Dagster (оркестрация данных)

PySpark (обработка Big Data)

Polars, Pandas (анализ данных)

deltalake (работа с lake-хранилищами)

PyArrow (работа с колоночными форматами)

СУБД и хранилища

ClickHouse — аналитическая СУБД (Yandex Managed Service for ClickHouse®)

PostgreSQL — хранение настроек и служебных данных

Redis — кэширование (open source, Docker или Yandex Managed)

Yandex Object Storage — облачное хранилище для данных

Инфраструктурные и вспомогательные компоненты

Контейнеризация: Docker, Kubernetes (Yandex Managed Service)

ОС в контейнерах: Ubuntu, Debian

Балансировка нагрузки: Yandex Application / Network Load Balancer

Веб-сервер / прокси: Nginx

Виртуализация и хостинг: Yandex Cloud (SaaS и Compute Cloud)

Авторизация: Yandex ID

# Эксплуатация системы модуля «Сквозная аналитика»

Запуск системы

Запуск модуля «Сквозная аналитика» осуществляется в рамках инфраструктуры Yandex Cloud с использованием контейнерной среды Kubernetes и Docker.

Развёртывание backend-сервисов (Java + Spring Boot), frontend-интерфейса (React), а также ETL-компонентов (Python, Dagster).

Инициализация подключений к базам данных (ClickHouse, PostgreSQL, Redis).

Подключение к источникам данных (AppMetrica, Яндекс.Метрика, логам плеера и т. д.) через API и коннекторы.

Активация пользовательского интерфейса с поддержкой авторизации через Yandex ID.

Проверка доступности сегментатора данных и «Платформы данных».

Для первой настройки предусмотрены шаблоны конфигураций, обеспечивающие автоматическое подключение к основным системам платформы «Предикто».

Управление

Модуль поддерживает централизованное управление через интерфейс администратора и встроенные инструменты доступа:

* Настройка ролей и прав пользователей (администраторы, аналитики, маркетологи и т. д.).
* Мониторинг состояния систем (веб-интерфейс, подписки, дашборды).
* Управление параметрами сегментации, фильтрами, отображением метрик.
* Отслеживание производительности ETL-процессов и аналитических вычислений.
* Управление подключением к внешним источникам данных через API.

Резервное копирование и восстановление

Модуль «Сквозная аналитика» поддерживает механизмы резервного копирования всех критически важных компонентов.

ClickHouse — автоматические snapshot'ы и экспорт таблиц на Yandex Object Storage.

PostgreSQL — регулярное бэкапирование настроек и метаданных (Yandex Managed Backup или pg\_dump).

Redis — дампы состояния in-memory кэша по расписанию. Файлы конфигураций и пользовательских дашбордов — хранение в объектном хранилище (Yandex Object Storage).

Восстановление может быть выполнено из любого из предыдущих snapshot'ов или экспортов по команде администратора. Все действия по восстановлению документированы и могут быть автоматизированы через скрипты CI/CD.

Обновление

Обновление системы осуществляется через систему управления контейнерами (CI/CD pipeline) с минимальным простоем.

Использование версионирования Docker-образов.

Тестирование обновлений в staging-среде.

Обновление производится поэтапно: backend, frontend, ETL-слой, интерфейс сегментации.

При необходимости применяется rolling update для минимизации влияния на пользователей.

Обновление дашбордов и метрик производится с сохранением пользовательских конфигураций.

Все обновления сопровождаются документацией, описанием новых функций и возможностью отката к предыдущей версии при необходимости.

# Предназначение модуля «Инфраструктура для сервиса»

Модуль «Инфраструктура для сервиса» инфраструктуру для загрузки, хранения и обработки больших объемов данных, а также интеграцию с другими модулями Предикто. Включает S3-совместимое объектное хранилище, ПО для оркестрации данных и коннекторы для выгрузки данных из популярных источников.

Основные задачи модуля:

* Организация хранения данных;
* Обработка больших данных;
* Оркестрация данных;
* Интеграция с источниками данных;
* Обеспечение качества данных;
* Обеспечение взаимодействия между модулями.

# Описание функциональных характеристик «Инфраструктура для сервиса»

20.1 Организация хранения данных

Модуль предоставляет объектное хранилище для больших объемов данных, которое разделяется на несколько логических слоев:

raw — необработанные данные, поступившие из внешних источников.

curated — очищенные и подготовленные данные для аналитики.

aggregated — агрегированные данные, готовые для использования в аналитических отчетах и моделях.

Каждому слою данных присваиваются соответствующие права доступа, что позволяет точно контролировать, кто и какие данные может использовать. Также модуль поддерживает интеграцию с ПО для хранения данных в формате Parquet и Delta Lake или аналогичными форматами, что обеспечивает гибкость в работе с различными типами данных.

20.2 Обработка и оркестрация данных

Модуль использует масштабируемую и распределенную систему для обработки данных на основе открытой технологии Apache Spark. Это позволяет эффективно обрабатывать большие объемы данных с использованием параллельных вычислений.

Также включен компонент оркестрации данных, который позволяет:

* Запускать код на языке Python для выполнения различных задач обработки.
* Планировать выполнение задач по расписанию.
* Отслеживать статус выполнения задач, а также ошибки, возникшие при обработке данных.
* Управлять зависимостями между задачами, чтобы гарантировать корректность выполнения всех шагов обработки данных.

20.3 Интеграция с источниками данных

Модуль предоставляет готовые компоненты (коннекторы) для загрузки данных из различных источников, включая реляционные и нереляционные базы данных, такие как:

PostgreSQL

ClickHouse

MongoDB

Кроме того, доступны коннекторы к популярным сервисам аналитики пользовательских событий, таким как Яндекс.Метрика и AppMetrica, для загрузки данных через их API. Для обработки данных в реальном времени также предусмотрена возможность интеграции с потоковыми источниками данных через Kafka.

20.4 Обеспечение качества данных

Модуль включает инструменты для мониторинга качества данных, которые интегрируются в процессы загрузки и обработки. Это обеспечивает:

* Контроль за качеством данных на каждом этапе их обработки.
* Технологические возможности для загрузки данных как в потоковом, так и в пакетном режиме.
* Решения для отслеживания зависимости данных, их происхождения и изменений, что позволяет обеспечить прозрачность и надежность в работе с данными.

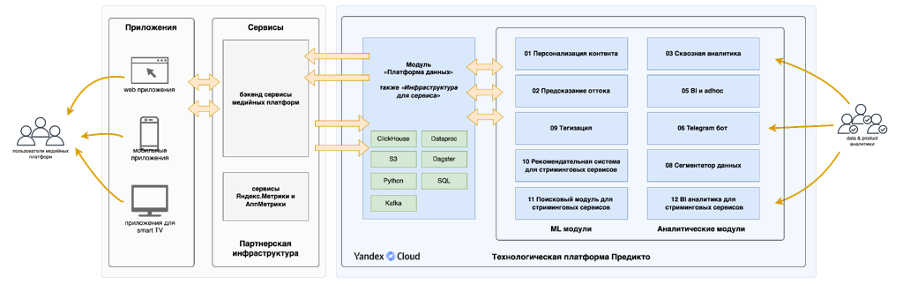
20.5 Обеспечение взаимодействия модулей

Возможность получать данные из модуля «Инфраструктура для сервиса» в необходимом объеме и скорости, как в потоковом, так и в пакетном режиме, используя стандартизированные протоколы и форматы данных.

Загружать данные в модуль «Инфраструктура для сервиса» из других модулей платформы Предикто с использованием тех же стандартов, что обеспечивает безошибочный обмен данными между модулями.

# Архитектура модуля «Инфраструктура для сервиса»

Архитектура строится вокруг интеграции приложений, данных и аналитики для создания персонализированного опыта и поддержки аналитических задач, используя мощные инфраструктурные и ML инструменты.



*Рис.1 Концептуальная архитектура модуля «Инфраструктура для сервиса»*

Пользователи медийных платформ взаимодействуют с различными приложениями (web, мобильными, для smart TV) через платформу.

Web-приложения, мобильные приложения и приложения для smart TV направляют данные к сервисам.

Сервисы, предоставляемые для медийных платформ, включают backend-сервисы для обработки данных и сервисы аналитики, такие как Яндекс.Метрика и AppMetрика.

Инфраструктура для сервиса включает модуль «Платформа данных», который используется для обработки и хранения данных. ClickHouse, Dataproc, S3, Dagster, Python, SQL, Kafka — различные технологии и инструменты для обработки данных.

ML модули. Модели машинного обучения (ML), такие как персонализация контента, предсказание оттока, теги, рекомендательные системы, поисковые модули и другие, используют данные из платформы для создания аналитики.

Аналитические модули. Подключение к аналитическим системам для выполнения задач сквозной аналитики, BI, работы с Telegram-ботами, сегментированием данных и создания BI-анализа для стриминговых сервисов.

Data & product аналитики. Данные и аналитики продуктов осуществляют мониторинг и анализ всей работы платформы, обеспечивая связь между сервисами и модулями.

# Технические и эксплуатационные характеристики модуля «Инфраструктура для сервиса»

Доступность инфраструктуры

Критически важные приложения и данные реплицируются в три дата-центра, расположенных в разных зонах доступности, что обеспечивает высокую степень отказоустойчивости и минимизирует риск потери данных в случае локальных сбоев. Инфраструктура предоставляет возможности для надежного хранения и регулярного бэкапирования данных, что гарантирует сохранность информации в любых условиях. Для обеспечения бизнес-непрерывности предусмотрены механизмы аварийного восстановления, которые активируются при масштабных сбоях или катастрофах, затрагивающих одну из зон доступности. Инфраструктура гарантирует возможность эксплуатации высокодоступных и высоконагруженных приложений, которые способны выдерживать значительные нагрузки.

Архитектура модулей.

При проектировании системы использована модульная архитектура, что позволяет разбить платформу «Предикто» на отдельные функциональные части. Это обеспечивает независимую разработку, тестирование и внедрение каждого модуля, а также упрощает их интеграцию через модуль «Платформа данных». Разработана система, оптимизирующая потребление аппаратных ресурсов. В частности, используются такие подходы, как кэширование данных и распределенные вычисления, что значительно повышает производительность системы.

Модуль «Платформа данных» был разработан с использованием публичной облачной платформы Яндекс.Облако. Все интеграции с другими модулями платформы проектируются с учетом того, что они также будут развернуты в Яндекс.Облаке. Для снижения операционных издержек, связанных с созданием и эксплуатацией инфраструктуры, активно используются управляемые сервисы, предоставляемые Яндекс.Облаком. Это позволяет существенно уменьшить затраты на обслуживание и повышение доступности.

Инфраструктура должна обеспечивать горизонтальное масштабирование программного обеспечения до 2000 одновременно используемых CPU и 20 одновременно используемых GPU, что позволяет справляться с большими объемами данных и интенсивными вычислениями. Для долгосрочного хранения данных система поддерживает хранение нескольких петабайт информации, обеспечивая надежную работу с большими объемами данных. Масштабирование инфраструктуры и выделение новых ресурсов может происходить в течение нескольких минут, в отдельных случаях — до нескольких часов, что позволяет оперативно адаптировать систему к изменяющимся требованиям. Для обеспечения высокой доступности и масштабируемости программного обеспечения используются такие методы, как кластеризация серверов и балансировка нагрузки.

Требования к окружениям.

В рамках модуля предусмотрено разделение контуров для разработки и контуров промышленной эксплуатации, что позволяет изолировать процессы разработки от реальной эксплуатации на уровне доступа к инфраструктуре и данным. Каждый функциональный компонент развернут как минимум в двух окружениях: development (для разработки и тестирования) и production (для промышленной эксплуатации). При необходимости могут быть созданы дополнительные окружения, такие как staging (предварительное тестирование) или QA (контроль качества).

# Используемые языки программирования и фреймворки модуля «Инфраструктура для сервиса»

Язык - Python:

Библиотеки и фреймворки

Polars — это высокопроизводительная библиотека для обработки данных, разработанная как альтернатива Pandas. Она использует многопоточность для эффективной работы с большими объемами данных и является одним из ключевых инструментов для анализа данных в модуле.

Pandas — это популярная библиотека Python для работы с табличными данными. Она предоставляет мощные структуры данных (например, DataFrame), которые позволяют эффективно обрабатывать, очищать и анализировать данные в табличном формате.

Delta Lake — это слой хранения данных для Apache Spark, который поддерживает транзакции ACID, улучшая надежность данных. Он используется для управления большими наборами данных с возможностью версионирования и поддержкой гибкой обработки данных.

PyArrow — это библиотека Python для работы с форматом Apache Arrow, который используется для быстрого обмена данными между различными системами и обеспечения эффективной сериализации. PyArrow

используется для оптимизации производительности в распределенных вычислениях.

Dagster — это фреймворк для оркестрации данных, который предоставляет возможности для разработки и управления конвейерами данных. Он используется для создания, тестирования и управления сложными задачами и процессами обработки данных.

PySpark — это интерфейс Python для Apache Spark, который используется для распределенной обработки данных. Он позволяет обрабатывать большие объемы данных с использованием кластеров вычислительных ресурсов и поддерживает работу с данными в реальном времени и пакетном режиме.

# Эксплуатация системы модуля «Инфраструктура для сервиса»

Развертывание и подготовка окружений.

Прежде чем запустить систему, необходимо развернуть инфраструктуру в соответствующих окружениях. Это включает подготовку окружений development (для разработки и тестирования) и production (для промышленной эксплуатации).

Установка и настройка зависимостей:

Все необходимые зависимости и библиотеки должны быть установлены через PIP.

Настройка системы.

После установки зависимостей требуется настроить конфигурационные файлы системы. Это включает параметры подключения к хранилищу данных, настройку сетевых соединений и обеспечение безопасности системы.

Запуск компонентов системы.

После подготовки и настройки всех компонентов можно запускать основные сервисы, такие как Apache Spark, Kafka, Dagster, а также базы данных, хранилища данных и другие компоненты, необходимые для функционирования системы.

Проверка работоспособности.

После запуска системы необходимо провести проверку работоспособности всех компонентов, чтобы убедиться в корректности их взаимодействия и функциональности. Это включает в себя тестирование обработки и загрузки данных, мониторинг производительности и проверку целостности данных.

# Предназначение модуля "BI и adhoc"

Программное обеспечение модуль "BI и adhoc" предназначено для предоставления корректных, обновляемых отчетов о состоянии бизнеса онлайн-кинотеатра для маркетинговых, продуктовых и контентных аналитиков компании, а также для менеджмента всех уровней.

# Описание функциональных характеристик модуля "BI и adhoc"

## 26.1 BI (Business Intelligence)

Автоматизированная обработка и анализ данных из различных источников.

Интерактивные дашборды с визуализацией ключевых показателей.

Инструменты для прогнозного анализа и выявления трендов.

Интеграция с внешними базами данных и API.

Настраиваемые отчеты и диаграммы для детального анализа.

## 26.2 AdHoc-отчеты

Настройка параметров отчетов пользователем.

Возможность построения сложных запросов без знаний SQL.

Динамическое изменение фильтров и группировки данных.

Экспорт отчетов в различные форматы (Excel, PDF, CSV).

Совместное использование отчетов и дашбордов в команде.

# Архитектура модуля "BI и adhoc"

Модуль использует следующие объекты:

BI-сервер, размещенный на виртуальной машине в Облаке; БД ClickHouse, размещенная в Облаке;

Модуль «Платформа данных».

Платформа данных

БД Clickhouse

BI-сервер

billing, content

billing, content

*Рис.1 Схема работы Модуля*

# Используемые языки программирования и фреймворки модуля "BI и adhoc"

|  |  |
| --- | --- |
| Языки программирования | Python, SQL. |
| Системы управления базами данных | PostgreSQL, ClickHouse. |
| Инструменты обработки данных | Apache Airflow, dbt. |
| Средства управления доступом | Active Directory |

## Логическая модель данных (табличное представление и ER) и описание справочников

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| dt | Date | Дата смотрения |
| passport\_id | String | Идентификатор пользователя |
| subscription\_kind | String | Тип подписки |
| subscription\_id | String | Идентификатор подписки (id транзакции) |
| tv\_id | Int32 | Идентификатор тайтла |
| season\_num | Int32 | Сезон |
| episode\_num | Int32 | Эпизод |
| view\_type | Int32 | Платформа смотрения |
| tvtype\_name | LowCardinality(String) | Тип контента |
| title\_ru | LowCardinality(String) | Название тайтла (без знаков препинания в lowercase) |
| cnt\_views | Int32 | Количество просмотров |
| cnt\_views\_3min | Int32 | Количество просмотров 3+ мин |
| view\_duration\_s | Int32 | Длительность смотрения |
| cnt\_views\_wt\_tt | Int32 | Количество просмотров без трейлеров и тизеров |
| cnt\_views\_3min\_wt\_tt | Int32 | Количество просмотров 3+ мин без трейлеров и тизеров |
| view\_duration\_s\_wt\_tt | Int32 | Длительность смотрения без трейлеров и тизеров |
| created\_at | DateTime | Дата создания |
| advert\_mode | String | Флаг рекламного просмотра |
| age\_restriction | String | Возрастное ограничение |
| country\_name | String | Первая страна производитель |
| first\_date\_watch | Date | Дата первого просмотра |
| genre\_id | UInt64 | Id основного жанра |
| genre\_name | String | Название основного жанра |
| geo\_region\_title\_ru | String | Регион просмотра |
| kids\_flag | UInt64 | Флаг контента 18+ (если age\_restriction=18+, то adult, иначе kids) |
| LT\_continouos | UInt64 | Непрерывный LT |
| LTV | UInt64 | LTV |

Эти данные хранятся в ClickHouse и используются для построения аналитических отчетов.

# Схема работы модуля "BI и adhoc"

Ежедневно со стороны ПО «Платформа данных» данные сохраняются в БД ClickHouse в определенные витрины и агрегаты.

Эти источники используются для построения дашбордов в BI-сервере.

Все алгоритмы заложены на стороне ПО «Платформа данных».

В основном используются агрегации, оконные сдвиги, объединения и соединения таблиц.

***Основные сущности модуля***

Пользователь – конечный пользователь системы, взаимодействующий с дашбордами.

Создатель дашбордов (редактор) – специалист, формирующий витрины и агрегаты для дашбордов.

Сервер – инфраструктурный компонент, обеспечивающий хранение и обработку данных.

Дашборды – визуализированные отчеты на основе данных.

Данные для дашбордов – агрегированные показатели для анализа.

***Процесс обработки данных***

Редактор использует модуль «Платформа данных» для генерации витрин и агрегатов.

Сервер получает данные для дашбордов в режиме онлайн или путем выгрузок по расписанию.

Дашборды размещаются на сервере и используют данные из источников.

Пользователи взаимодействуют с дашбордами с помощью интерактивных фильтров и настроек.

# Эксплуатация системы модуля "BI и adhoc"

Для запуска модуля необходимо наличие корректно настроенного серверного окружения с установленными зависимостями.

Подключение к базе данных ClickHouse осуществляется автоматически при запуске сервиса.

Запуск системы выполняется с помощью командного интерфейса или через интерфейс администрирования.

При запуске система проводит самотестирование, проверяя целостность конфигурации и доступность внешних сервисов.

Управление модулем осуществляется через веб-интерфейс.

Доступ к дашбордам и их настройке регулируется ролями и правами пользователей.

Администратор системы может управлять правами доступа, настройками обновлений и резервного копирования.

Логи работы системы хранятся и доступны для анализа в случае возникновения ошибок.

## 

Обновления системы включают исправление ошибок, улучшение производительности и добавление новых функций.

Перед обновлением создается резервная копия текущего состояния системы.

Установка обновлений осуществляется в автоматическом режиме с возможностью отката в случае критических ошибок.

Обновления публикуются в специальном разделе администрирования, где администратор может выбрать подходящее время для их применения.

## **Назначение модуля «Telegram бот»**

Модуль «Telegram бот» предназначен для предоставления актуальных отчетов о состоянии бизнеса онлайн-кинотеатра «Премьер» в интерфейсе мессенджера Telegram. Он обеспечивает удобный доступ к аналитическим данным менеджменту всех уровней.

## **Функциональные требования модуля «Telegram бот»**

Модуль «Telegram бот» обеспечивает удобную интеграцию бизнес-аналитики с Telegram, предоставляя оперативные отчеты менеджменту. Для обеспечения надежной работы рекомендуется проводить регулярное тестирование и мониторинг процессов передачи данных.

Модуль выполняет следующие функции:

* Получение актуальных и исторических данных по оплатам подписок (биллингу), смотрению, посещению и действиям пользователей.
* Получение актуальных и исторических метаданных по контенту.
* Хранение данных в базе.
* Агрегация данных для формирования отчетов.
* Отправка агрегированных отчетов через Telegram бот.
* Управление пользователями, имеющими доступ к информации в боте.

## **Архитектура модуля «Telegram бот»**

Модуль Telegram бота включает следующие компоненты:

Сервер, размещенный на виртуальной машине в Yandex Cloud.

Мессенджер Telegram, через который пользователи получают отчеты.

База данных ClickHouse, используемая для хранения и обработки данных.

Платформа данных, отвечающая за сбор и подготовку данных.

Платформа данных

БД Clickhouse

Tableau

billing, content

billing, content

*Рис.1 Схема работы Модуля*

## 

## **Схема работы модуля «Telegram бот»**

Данные из «Платформы данных» загружаются в ClickHouse.

По расписанию или по запросу агрегированные данные подготавливаются для отправки.

Telegram бот получает промпты и данные для формирования отчетов.

Бот отправляет сообщения пользователям согласно их ролям и доступам.

## **Используемые языки программирования и фреймворки**

### Реализация

Язык программирования: Python

Используемые библиотеки:

* + Pandas
  + Telebot
  + Telegram Bot calendar

## **Последовательность потоков данных модуля «Telegram бот»**

Ежедневное обновление витрин в ClickHouse.

Формирование агрегатов и структурированных отчетов.

Передача данных Telegram боту для отправки пользователям.

## **Описание алгоритмов и математического обеспечения модуля «Telegram бот»**

Используются методы агрегации, оконных сдвигов, объединений и соединений таблиц.

Все вычисления происходят на стороне «Платформы данных».

## **Логическая модель данных и справочники модуля «Telegram бот»**

episode\_views\_by\_viewer

Основные поля:

dt – дата смотрения

passport\_id – идентификатор пользователя

subscription\_kind – тип подписки

subscription\_id – идентификатор подписки

tv\_id – идентификатор тайтла

season\_num, episode\_num – сезон и эпизод

cnt\_views, cnt\_views\_3min, view\_duration\_s – просмотры и длительность смотрения

age\_restriction, genre\_name, geo\_region\_title\_ru – метаданные контента

Extended\_billing\_aggregate

Основные поля:

passport\_id – id пользователя

start\_date, end\_date – даты начала и окончания подписки

money – стоимость подписки

LTV, LT\_total – жизненная ценность пользователя

content\_dash\_table\_till\_user\_driver\_allocation\_from\_2020\_history

Основные поля:

tv\_id, title\_ru, season\_num, dt – идентификаторы контента

watching\_time, alloc\_revenue – данные о выручке и времени просмотра

## **Роли модуля «Telegram бот»**

|  |  |
| --- | --- |
| Пользователь | Получение отчетов в Telegram |
| Администратор | Управление пользователями |
| Сервер | Обработка и отправка отчетов |
| Платформа данных | Сбор и подготовка информации |

# Модуль «Модерация контента»

Модуль «Модерация контента» предназначен для автоматизированного анализа и разметки видеоконтента (фильмы, сериалы) в онлайн-кинотеатре. Он выполняет модерацию контента в соответствии с установленными критериями, обеспечивая безопасную и здоровую среду для пользователей платформы.

# Назначение модуля «Модерация контента»

Модуль разработан для автоматизированной проверки видеоконтента на наличие нежелательных элементов, таких как:

* Пропаганда наркотиков, алкоголя и курения;
* Сцены с ЛГБТ-контентом;
* Контент, нарушающий политику платформы.

# Функциональные возможности модуля «Модерация контента»

Модуль выполняет следующие задачи:

* Анализ видеофайлов и их метаданных, включая информацию из сервисов Кинопоиск и IMDB;
* Определение наличия ключевых признаков (наркотики, курение, алкоголь, ЛГБТ и др.);
* Формирование разметки контента и предоставление отчетов по результатам модерации;
* Хранение данных о модерации в стандартизированном формате.

# Архитектура и принцип работы модуля «Модерация контента»

### Принцип работы

Модуль загружает входной файл с данными о контенте.

Запрашивает информацию о контенте в сервисах Кинопоиск и IMDB.

Анализирует полученные данные, используя набор предустановленных критериев.

Формирует выходной файл с разметкой контента и классификацией по заданным категориям.

### 

### 

### 

### 

### 

### Структура входного и выходного файлов модуля «Модерация контента»

Входной файл (XLSX):

|  |  |
| --- | --- |
| **Название столбца** | **Описание** |
| Название фильма | Название фильма на русском языке |
| Оригинальное название | Оригинальное название фильма |
| Год | Год выпуска фильма |
| Страна производства | Страна производства фильма |
| Кинопоиск | Ссылка на фильм в сервисе Кинопоиск |
| IMDB | Ссылка на фильм в сервисе IMDB |

Выходной файл (XLSX):

|  |  |
| --- | --- |
| **Название столбца** | **Описание** |
| Название фильма | Название фильма на русском языке |
| Оригинальное название | Оригинальное название фильма |
| Год | Год выпуска фильма |
| Страна производства | Страна производства фильма |
| Кинопоиск | Ссылка на фильм в сервисе Кинопоиск |
| IMDB | Ссылка на фильм в сервисе IMDB |
| Метки Кинопоиск | Полный список тегов фильма в сервисе Кинопоиск |
| Разметка курения | Цветовая разметка наличия или отсутствия признаков курения |
| Теги курения | Список тегов, связанных с курением (если есть) |
| Разметка алкоголя | Цветовая разметка наличия или отсутствия признаков алкоголя |
| Теги алкоголя | Список тегов, связанных с алкоголем (если есть) |
| Разметка наркотиков | Цветовая разметка наличия или отсутствия признаков наркотиков |
| Теги наркотиков | Список тегов, связанных с наркотиками (если есть) |
| Разметка ЛГБТ | Цветовая разметка наличия или отсутствия признаков ЛГБТ |
| Теги ЛГБТ | Список тегов, связанных с ЛГБТ (если есть) |

# 

# 

# Используемые языки программирования и фреймворки модуля «Модерация контента»

|  |  |
| --- | --- |
| Языки программирования | Python |
| Используемые библиотеки | Pandas, Selenium, Requests, OpenPyXL, Loguru, charset-normalizer, chardet, Levenshtein, random\_user\_agent, transliterate, undetected\_chromedriver, user\_agent, CatBoost |
| Контейнеризация | Docker, Yandex Managed Service for Kubernetes |
| Репозитории контейнеров | Docker Hub, mirror.gcr.io, Yandex Container Registry |
| СУБД | ClickHouse (Yandex Managed Service for ClickHouse) |

# Возможные ошибки и решения модуля «Модерация контента»

Ошибка подключения к интернету – проверить соединение;

Нет данных в IMDB/Кинопоиске – контент не найден в базе, требуется ручная проверка;

Ошибка анализа файла – убедиться в корректности структуры входного файла.

В этом случае:

Проверьте наличие соединения с сетью Интернет;

Проверьте доступность сервиса Кинопоиск, открыв сайт https://www.kinopoisk.ru/ в браузере; Проверьте доступность сервиса IMDB, открыв сайт https://www.imdb.com в браузере.

Если все сервисы доступны, но ошибка все равно возникает, обратитесь к разработчику.

# Модуль «Сегментатор данных»

Модуль «Сегментатор данных» предназначен для автоматизированного сбора, обработки и анализа данных пользователей, с возможностью их сегментации по различным критериям. Он предоставляет интерфейс для гибкой настройки сегментов и их выгрузки в различных форматах.

# Назначение модуля «Сегментатор данных»

Модуль разработан для использования в компаниях, работающих с большими объемами данных, включая:

* Электронную коммерцию (анализ эффективности маркетинговых кампаний);
* Рекламные агентства (оптимизация рекламных кампаний);
* Разработчиков мобильных приложений (анализ поведения пользователей);
* Компании-разработчики ПО (оптимизация работы сервисов);
* Онлайн-сервисы (повышение удовлетворенности клиентов).

Аутентификация проходит через сервисы Яндекс ID; Авторизация на основе ролевой модели ( две роли: Администратор, Пользователь); Управление сегментами (создание, редактирование, удаление). Выгрузка данных сегментов в форматах CSV и XLSX. Интеграция с различными источниками данных (PostgreSQL, ClickHouse). Автоматизированное формирование аналитических отчетов.

# Функциональные возможности модуля «Сегментатор данных»

Модуль «Сегментатор данных» выполняет следующие функции:

* Авторизация на основе ролевой модели (Администратор, Пользователь).

Управление сегментами, включая:

* Создание новых сегментов.
* Редактирование существующих сегментов.
* Удаление сегментов.
* Переименование и копирование сегментов.

Настройка сегментов:

* Настройка фильтров по атрибутам.
* Группировка фильтров.
* Выгрузка сегментов в различных форматах (CSV, XLSX).
* Просмотр и анализ ранее сформированных сегментов.
* Интеграция с источниками данных, такими как PostgreSQL и ClickHouse.
* Автоматизированное формирование аналитических отчетов.
* Поиск и фильтрация сегментов по заданным критериям.

Работа с пользователями (для Администратора):

* Поиск пользователей.
* Создание и удаление пользователей.
* Назначение ролей и управление доступами.

# 

# Архитектура и принцип работы модуля «Сегментатор данных»

# 

Принцип работы

1. Пользователь загружает данные из различных источников.
2. Происходит обработка данных и разбиение их на сегменты по заданным критериям.
3. Пользователь получает сформированные сегменты и может выгрузить их в различных форматах.

Сегменты доступны для анализа и интеграции с другими сервисами.

Структура данных

Модель данных PostgreSQL:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| id | integer | Уникальный идентификатор |
| name | varchar(50) | Имя сегмента |
| rule | jsonb | Правило выгрузки |
| updated\_at | timestamp(6) | Дата обновления |

Модель данных ClickHouse:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| user\_id | String | Идентификатор пользователя |
| date | Date | Дата события |
| platform\_main | String | Платформа пользователя |
| cohort | Date | Когорта |

Модуль состоит из следующих компонентов:

Клиентская часть (Web, Mobile, другие устройства);

Сервер визуализации и обработки метрик, включающий:

* + Frontend app (отображение данных);
  + Backend app (обработка данных);
  + Базы данных PostgreSQL и Redis;

Сервер сбора метрик, включающий:

* + S3 БД (хранение данных);
  + Главная БД ClickHouse (агрегация данных);
  + Модуль сбора данных;
  + Модуль обработки данных;
  + Dagster оркестратор (управление потоками данных);

Сервисы сбора аналитических данных (интеграция с внешними источниками).

Сервисы сбора

аналитических данных

Клиент

Web

mobile

Другие устройства

Сервисы визуализации сбора

метрик

Frontend app **Component**

Backend app **Component**

Postgresql DB

Redis DB

Сервисы сбора

метрик

Главная БД

ClickHouse

S3

БД

Модуль обработки данных

Модуль сбора

данных

**Dagster**

**оркестратор**

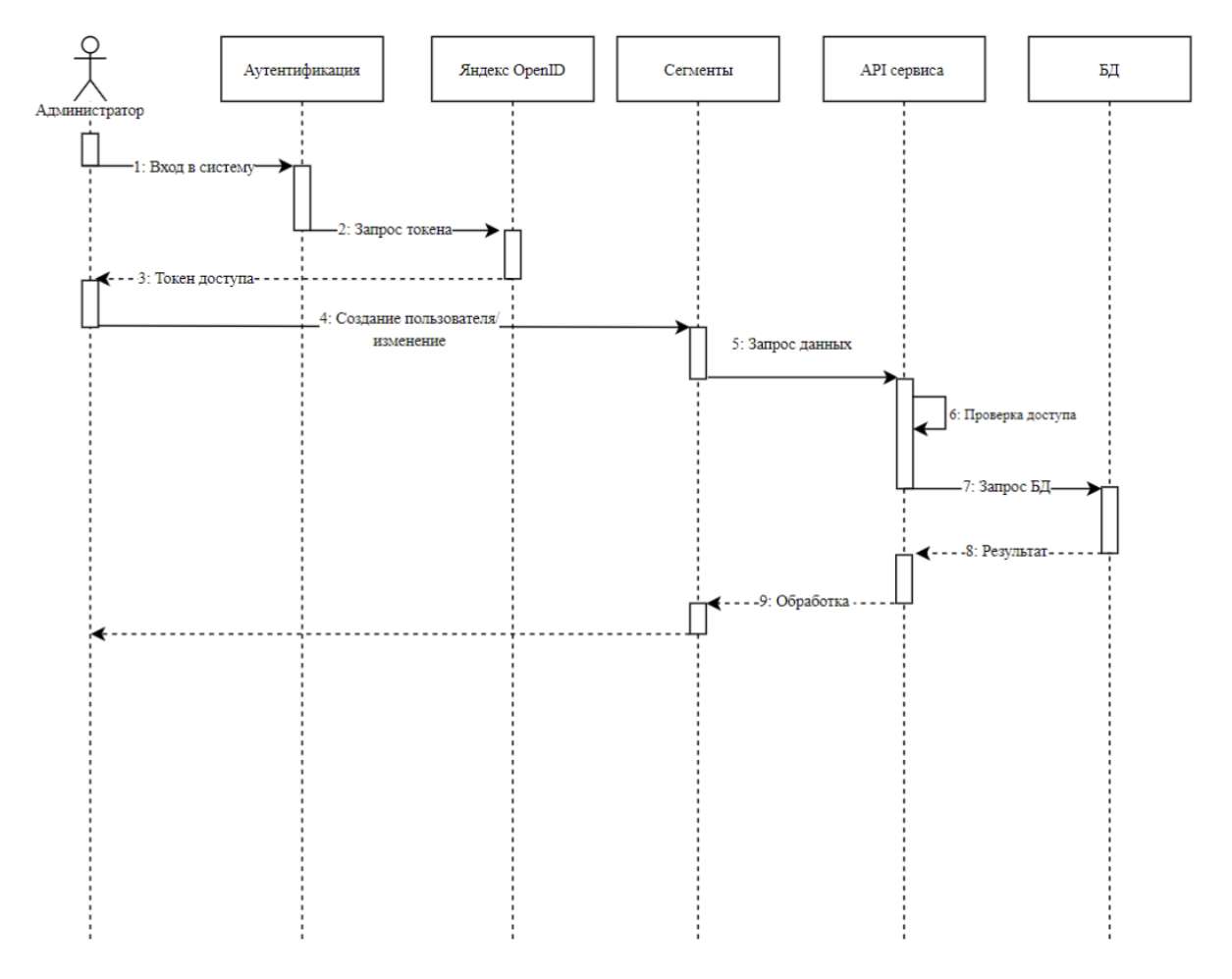
Сервис метрик

**API**

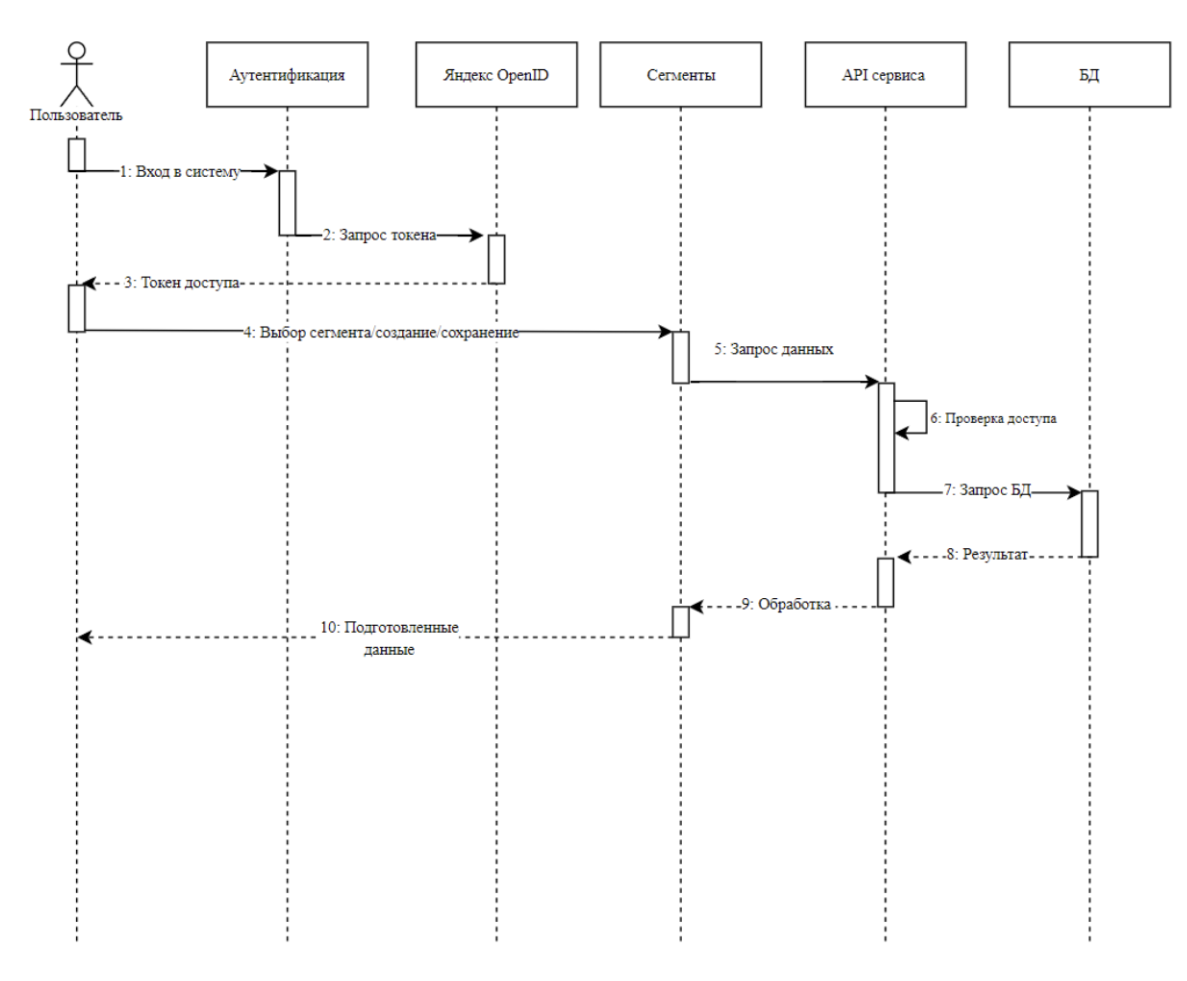
*Рис. 1 Общая схема сервиса*

Потоки данных, включая протоколы обмена и сценарии

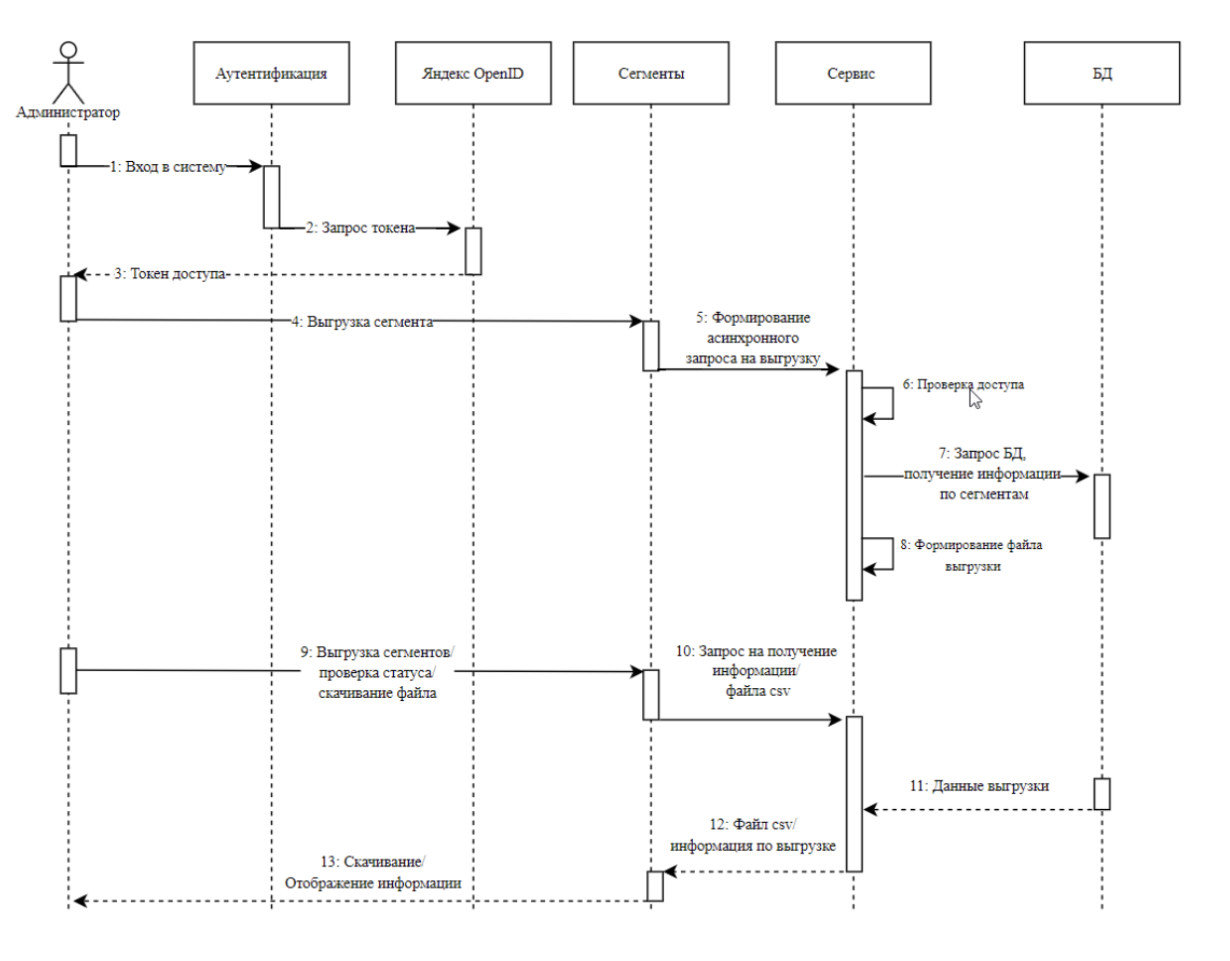
Ниже представлены основные сценарии в виде диаграмм последовательности.



*Рис. 2 Диаграмма последовательности: Создание пользователя/изменение.*

**

*Рис. 3 Диаграмма последовательности: Выбор сегмента/создание/изменение.*

**

*Рис. 4 Диаграмма последовательности: Выгрузка сегментов.*

# Роли и полномочия модуля «Сегментатор данных»

# 

Администратор

* Управление пользователями (создание, удаление, назначение ролей);
* Контроль доступа к данным и функциям системы;
* Настройка параметров выгрузки и отчетности.

Пользователь

* Просмотр и управление сегментами данных;
* Выгрузка данных;
* Анализ сегментов через веб-интерфейс.

# Используемые языки программирования и фреймворки модуля «Сегментатор данных»

База данных: PostgreSQL, ClickHouse

Язык программирования: Python

Фреймворки: Pandas, FastAPI, GraphQL

Инфраструктура: Yandex Cloud, Docker, Kubernetes

## **Модуль «Тегизация»**

Модуль "Тегизация" предназначен для автоматического формирования тематик контента на основе интересов пользователей. Он использует алгоритмы машинного обучения для анализа действий пользователей, обработки контента и формирования персонализированных тегов.

Данный модуль является частью системы персонализации контента и предназначен для автоматизации процесса категоризации видеоматериалов, что позволяет минимизировать ручной труд редакторов и повысить удовлетворенность пользователей.

## **Назначение модуля «Тегизация»**

Модуль "Тегизация" выполняет следующие ключевые задачи:

* Автоматическая классификация контента по тематическим тегам.
* Формирование персонализированных рекомендаций на основе интересов пользователей.
* Интеграция с рекомендательной системой для отображения релевантного контента.
* Обеспечение механизма обновления тегов при изменении пользовательских предпочтений.

## **Функциональные возможности модуля «Тегизация»**

Основные функции

* Получение и обработка данных о пользователях и контенте.
* Анализ пользовательских предпочтений на основе истории взаимодействий.
* Генерация тегов и тематических кластеров.
* Формирование персонализированных подборок контента.
* API для интеграции с внешними сервисами и рекомендательными системами.
* Поддержка механизма "холодного старта" для новых пользователей.

Работа с тегами

* Генерация тегов на основе семантического анализа контента.
* Кластеризация тегов для упрощения поиска и навигации.
* Определение релевантных тегов для каждого пользователя.
* Динамическое обновление тегов с учетом изменений предпочтений пользователей.

Рекомендательный алгоритм

* Определение близости пользователя к тематическим кластерам.
* Анализ интересов пользователя по различным категориям контента.
* Генерация подборок с учетом истории взаимодействий и предпочтений

## **Архитектура и принцип работы модуля «Тегизация»**

Общая архитектура

Модуль "Тегизация" построен на микросервисной архитектуре и состоит из нескольких ключевых компонентов:

Сервис обработки данных – получает и обрабатывает данные о пользователях и контенте.

Модуль машинного обучения – выполняет анализ данных и формирует тематические кластеры.

API-сервис – предоставляет доступ к тегам и персонализированным рекомендациям.

Система хранения данных – включает базы данных PostgreSQL и ClickHouse для хранения тегов и истории взаимодействий.

Принцип работы

* Получение данных из внешних источников (каталоги контента, данные пользователей).
* Анализ контента и создание тегов на основе модели машинного обучения.
* Определение пользовательских предпочтений и формирование персонализированных кластеров.
* Передача данных в рекомендательную систему и генерация персонализированных подборок.
* Обновление тегов с учетом изменений в поведении пользователей.

Потоки данных

Система получает данные от пользователей и из внешних источников.

Анализируются текстовые описания, метаданные и взаимодействия пользователей.

Генерируются тематические теги и кластеры.

Теги передаются в рекомендательную систему и используются для персонализации контента.

**Роли и полномочия модуля «Тегизация»**

Основные роли в системе

* **Администратор** – управляет настройками модуля, редактирует данные, следит за корректностью работы системы.
* **Пользователь** – получает персонализированные подборки контента на основе тегов.
* **Редактор контента** – может вносить корректировки в тематические теги.

Полномочия ролей

**Администратор:** доступ ко всем функциям, настройке алгоритмов и мониторингу работы модуля.

**Редактор контента:** просмотр и правка тегов для улучшения качества категоризации.

**Пользователь:** просмотр тематических подборок без возможности редактирования тегов.

## **Используемые языки программирования и фреймворки модуля «Тегизация»**

Backend: Python (Flask, FastAPI), Java (Spring Boot)

Machine Learning: PyTorch, CatBoost, Scikit-learn

Data Processing: Apache Spark, Polars, PyArrow, DeltaLake

Database: PostgreSQL, ClickHouse

Infrastructure: Yandex Cloud, Kubernetes, Docker, Terraform

## **Модуль «Рекомендательная система для стриминговых сервисов»**

Рекомендательная система для стриминговых сервисов представляет собой инструмент, использующий алгоритмы анализа данных и машинного обучения для предоставления пользователям персонализированных рекомендаций контента.

Основная цель системы – повышение вовлеченности пользователей, увеличение времени просмотра и улучшение пользовательского опыта.

Система учитывает данные, включая историю просмотров, лайки, комментарии, подписки, тренды и предпочтения схожих пользователей, чтобы формировать наиболее релевантные рекомендации.

Данный модуль интегрируется в платформу и обеспечивает рекомендации для различных типов пользователей.

## **Назначение модуля «Рекомендательная система для стриминговых сервисов»**

Модуль "Рекомендательная система для стриминговых сервисов" предназначен для персонализированного подбора видеоконтента на основе анализа данных о поведении пользователей. Основные задачи модуля:

* Удержание пользователей за счет релевантных рекомендаций.
* Персонализация контента на основе индивидуальных предпочтений.
* Открытие нового контента для пользователей.
* Увеличение вовлеченности за счет подбора контента, стимулирующего активное взаимодействие (лайки, комментарии, подписки).

## 

## **Функциональные возможности модуля «Рекомендательная система для стриминговых сервисов»**

### Основные возможности модуля:

1. Персонализированные рекомендации – подбор контента на основе истории просмотров, лайков, комментариев, поисковых запросов.
2. Рекомендации по категориям – учет жанра и тематической направленности контента.
3. Контекстные рекомендации – подбор похожих видео, формирование плейлистов.
4. Коллаборативные рекомендации – подбор контента, популярного среди пользователей с аналогичными интересами.
5. Алгоритмические рекомендации – использование машинного обучения для более точных рекомендаций.
6. Рекомендации для новых пользователей – формирование рекомендаций при отсутствии данных (решение проблемы "холодного старта").
7. Трендовые рекомендации – анализ актуальных тем и трендов.
8. Мультиплатформенные рекомендации – синхронизация данных между устройствами (смартфоны, ПК, Smart TV).
9. Экспериментальные рекомендации – тестирование новых алгоритмов, A/B тестирование.
10. Учет пользовательского контента (UGC) – включение пользовательского контента в рекомендации.

## **Архитектура и принцип работы модуля «Рекомендательная система для стриминговых сервисов»**

## 

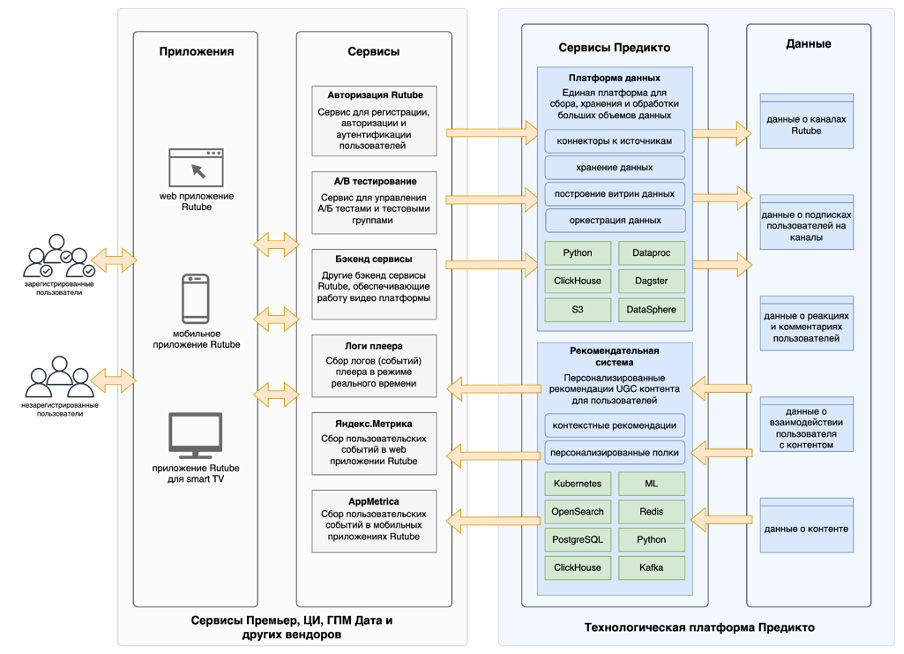


Рис. 1 диаграмма концептуальной архитектуры модуля рекомендательной системы для стриминговых сервисов

Модуль рекомендательной системы построен на многослойной архитектуре, включающей следующие уровни:

**Сбор данных**

Система анализирует пользовательские данные, такие как:

Просмотренные видео.

Реакции (лайки, дизлайки, комментарии).

Поисковые запросы.

Подписки на каналы.

**Анализ и обработка данных**

Используются следующие методы и алгоритмы:

Коллаборативная фильтрация – анализ поведения схожих пользователей.

Контентная фильтрация – анализ схожих характеристик контента (жанр, теги, описание).

Гибридные методы – комбинация коллаборативной и контентной фильтрации.

Машинное обучение – использование нейросетей и моделей для улучшения рекомендаций.

**Генерация рекомендаций**

На основе анализа данных система формирует персонализированные подборки контента, включая:

Видео, схожие с уже просмотренными.

Популярные видео среди пользователей с аналогичными интересами.

Новые видео из подписок.

Трендовые видео.

**Интерактивная адаптация**

Система анализирует реакцию пользователей на предложенные рекомендации (например, пропуск видео, лайки/дизлайки) и адаптирует последующие подборки в соответствии с их предпочтениями.

## **Используемые языки программирования и фреймворки модуля «Рекомендательная система для стриминговых сервисов»**

**Python** – основной язык программирования.

**Модуль реализован с использованием современных технологий и инструментов:**

**Серверная инфраструктура**

Kubernetes – оркестрация контейнеров.

Docker – контейнеризация сервисов.

Nginx – веб-сервер и балансировщик нагрузки.

FastAPI + Uvicorn – фреймворк для создания API.

**Базы данных**

PostgreSQL – основная база данных.

Redis – кэширование запросов.

OpenSearch – полнотекстовый поиск.

ClickHouse – хранение больших объемов логов.

**Машинное обучение и анализ данных**

Библиотеки:

* + scikit-learn, PyTorch, XGBoost, CatBoost – обучение моделей.
  + Sentence Transformers – работа с текстовыми эмбеддингами.
  + NumPy, SciPy, Pandas, Polars – анализ данных.
  + Dagster, Apache Spark – обработка данных.
  + Optuna – гиперпараметрическая оптимизация моделей.

**Логирование и мониторинг**

Prometheus + Grafana – сбор и анализ метрик.

Grafana Loki – логирование событий.

**Интеграционные сервисы**

API Rutube – взаимодействие с видеоплатформой.

Яндекс.Метрика, AppMetrica – анализ пользовательского поведения.

Kafka – потоковая обработка данных.

# Описание модуля «Поисковый модуль для стриминговых сервисов»

Поисковый модуль для стриминговых сервисов — это технологический инструмент, который использует алгоритмы и методы анализа данных, машинного обучения и обработки естественного языка (NLP) для быстрого и точного поиска контента, соответствующего запросам пользователей. Основная цель системы — обеспечить удобный и эффективный доступ к разнообразному контенту, включая пользовательский контент (UGC, user-generated content), фильмы, сериалы, музыкальные видео и другие материалы.

Поисковый модуль ориентирован на обработку запросов в условиях большого объема данных и разнообразия контента. Он учитывает как текстовые запросы, так и контекстные данные, такие как популярность контента, релевантность, предпочтения пользователя и текущие тренды. Это позволяет системе предлагать наиболее подходящие результаты, даже если запросы формулируются неточно или содержат опечатки.

# Описание функциональных характеристик модуля «Поисковый модуль для стриминговых сервисов»

63.1 Поиск по разным типам сущностей: Поддержка поиска по различным сущностям, таким как каналы, видео, персонажи и другие элементы контента. Это позволяет пользователям находить не только видео, но и другие типы контента, например, авторов или каналы.

63.2 Поиск по разным типам контента: Модуль поддерживает поиск по различным типам контента, таким как пользовательский контент (UGC), фильмы, сериалы, музыкальные клипы, короткие видео и другие виды материалов. Это обеспечивает гибкость и точность в поиске различных видов контента.

63.3 Поиск по новому контенту: Включает функциональность быстрого поиска нового контента, например, трансляций или видео, которые только что начали или только что были загружены. Это позволяет пользователям всегда быть в курсе актуальных и свежих материалов.

63.4 Поддержка сложных запросов: Модуль способен обрабатывать длинные и сложные запросы, позволяя фильтровать и сортировать результаты по категориям, жанрам, дате публикации и другим параметрам. Это предоставляет пользователям больше возможностей для уточнения своих запросов.

63.5 Ранжирование результатов: Поисковая система учитывает популярность, релевантность и качество контента при ранжировании результатов поиска. Это гарантирует, что наиболее подходящие материалы будут отображаться в верхней части списка результатов, а менее релевантные — внизу.

63.6 Автодополнение и исправление ошибок: Система поддерживает автодополнение запросов по мере ввода текста и автоматически исправляет опечатки, чтобы обеспечить точность поиска, даже если запрос введен с ошибками.

63.7 Персонализация: Модуль учитывает историю поиска и предпочтений пользователя для повышения точности и релевантности результатов. Это позволяет системе предлагать контент, который соответствует интересам и предпочтениям конкретного пользователя.

63.8 Транслитерация: Модуль учитывает транслитерацию и возможные ошибки, связанные со сменой раскладки клавиатуры, чтобы улучшить качество поиска и сделать его более точным для пользователей, вводящих запросы на разных языках или в разных раскладках.

# Архитектура модуля «Поисковый модуль для стриминговых сервисов»

В рамках интеграции поискового модуля для стриминговых сервисов с крупной медиа платформой активно используются различные внешние сервисы, которые обеспечивают функциональность, необходимую для эффективной работы системы поиска и взаимодействия с пользователями. Включение этих сервисов в архитектуру поискового модуля позволяет не только повысить качество поиска, но и улучшить пользовательский опыт за счет персонализации, аналитики и оптимизации.

***Внешние сервисы, участвующие в интеграции***

Авторизация: Сервис для регистрации, авторизации и аутентификации пользователей. Этот сервис обеспечивает безопасность и доступность персонализированного контента для каждого пользователя. Он позволяет уникально идентифицировать пользователя, а также управлять его доступом к различным материалам на платформе.

A/B тестирование: Сервис, который разделяет пользователей на группы для проведения A/B тестов. Это позволяет экспериментировать с различными вариантами интерфейса или функционала поискового модуля, чтобы выбрать наиболее эффективный для улучшения пользовательского опыта и повышения вовлеченности.

Бэкенд сервисы: Ряд различных бэкенд сервисов, которые обеспечивают работу всей видео платформы. Эти сервисы взаимодействуют с поисковым модулем для обеспечения правильного отображения результатов поиска, предоставляя пользователю доступ к видео, каналам и другому контенту. Также эти сервисы играют ключевую роль в обработке запросов и обеспечении масштабируемости системы.

Логи плеера: Сервис для сбора логов (событий) видео плеера в режиме реального времени. Этот сервис необходим для отслеживания пользовательских действий, таких как паузы, воспроизведение, переходы по видео и другие важные события. Эти данные позволяют улучшить точность персонализированных рекомендаций и анализировать поведение пользователей.

AppMetrica: Сервис для сбора пользовательских событий в мобильных приложениях. AppMetrica помогает собирать подробные данные о взаимодействии пользователей с мобильным приложением, включая клики, просмотры контента, а также другие действия, которые могут быть использованы для улучшения поиска и персонализации результатов.

Яндекс.Метрика: Сервис для сбора пользовательских событий в web приложении (Desktop и мобильный Web). Яндекс.Метрика предоставляет подробную информацию о поведении пользователей в web-версии платформы, включая анализ запросов, популярности контента и других ключевых показателей. Эти данные используются для улучшения поисковых алгоритмов и предложения более релевантного контента.

***Интеграция внешних сервисов с поисковым модулем***

Интеграция этих внешних сервисов с поисковым модулем помогает создать динамичную и гибкую систему, способную эффективно работать с большим объемом данных и запросов пользователей.

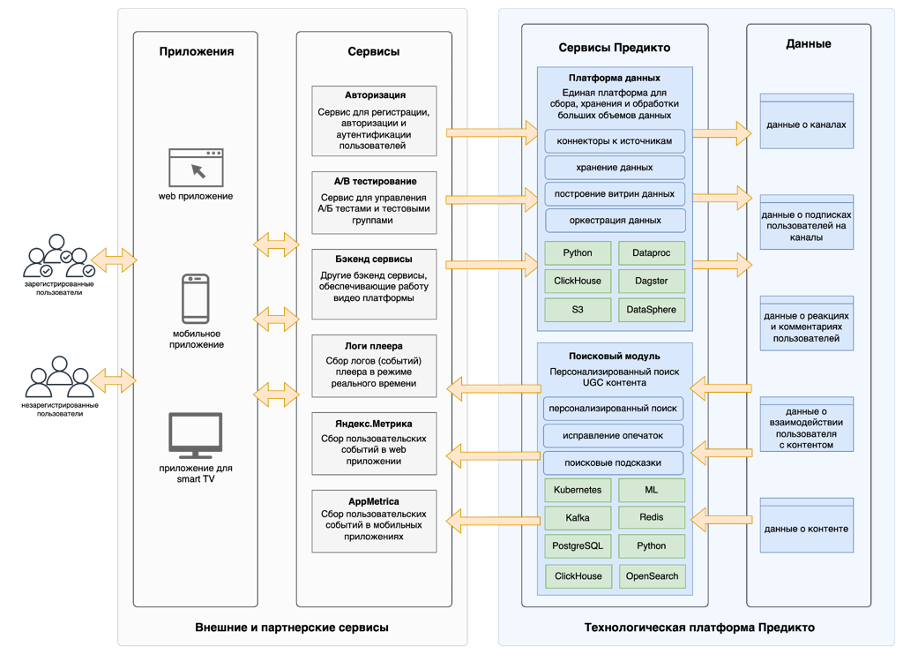
Авторизация служит основой для персонализированного подхода, где каждый пользователь получает результаты поиска, адаптированные под его предпочтения и историю взаимодействия с контентом.

A/B тестирование позволяет тестировать различные версии поисковых алгоритмов и интерфейса, что способствует оптимизации пользовательского опыта и повышению эффективности работы поисковой системы.

Бэкенд сервисы предоставляют необходимые ресурсы для обработки запросов пользователей, поиска контента и взаимодействия с базами данных, а также обеспечивают поддержку высоких нагрузок на систему.

Логи плеера и AppMetrica обеспечивают сбор данных о поведении пользователей в реальном времени, что позволяет анализировать эффективность поиска и адаптировать результаты под изменения в пользовательских предпочтениях.

Яндекс.Метрика и AppMetrica используются для сбора статистики, анализа пользовательского опыта и выявления возможных узких мест в функциональности поисковой системы



*рис.1 - концептуальная архитектура Поискового модуля для стриминговых сервисов*

# Схема работы модуля «Поисковый модуль для стриминговых сервисов»

Уровень приложений

Пользователи получают доступ к поисковому модулю через различные интерфейсные приложения: мобильное приложение, веб-приложение, а также через смарт-телевизоры и другие устройства. Эти интерфейсные приложения не являются частью поискового модуля, однако играют ключевую роль в обеспечении взаимодействия пользователя с системой поиска.

Взаимодействие между интерфейсными приложениями и поисковым модулем осуществляется через API. Это позволяет приложениям отправлять запросы к поисковому модулю, получать результаты поиска и отображать их пользователю. API служит как стандартный интерфейс для обмена данными между интерфейсными приложениями и ядром поисковой системы, обеспечивая гибкость и масштабируемость решения.

Уровень внешних сервисов

Для корректной работы поискового модуля, он интегрирован с рядом внешних сервисов, которые предоставляют дополнительные данные и функциональность. Эти сервисы взаимодействуют с платформой Предикто через стандартизированные протоколы и API.

Платформа данных

Для интеграции и работы с внешними сервисами используется Платформа данных, которая является ключевым компонентом инфраструктуры. Платформа данных отвечает за сбор, хранение и обработку больших объемов данных, поступающих из различных источников. Интеграция с внешними сервисами может осуществляться как по пакетной, так и по потоковой обработке данных, в зависимости от специфики источника.

Модуль Инфраструктура для сервиса (или Платформа данных) реализует все необходимые функции по получению данных, их хранению и дальнейшей обработке. Этот модуль также отвечает за обеспечение высокой доступности данных и их обработку в реальном времени.

Поисковый модуль для стриминговых сервисов

Основной функционал поискового модуля заключается в персонализированном поиске контента на базе собранных данных о пользователях и контенте, с использованием технологий машинного обучения для улучшения качества поиска.

Основные компоненты поискового модуля включают:

Kubernetes: Платформа для оркестрации контейнеров, которая управляет развертыванием и масштабированием компонентов поискового модуля.

ML: Модели машинного обучения для ранжирования результатов поиска и персонализации контента, что позволяет точно подбирать материалы в соответствии с предпочтениями пользователя.

OpenSearch: Мощный поисковый движок, используемый для поиска и сортировки релевантного контента для каждого запроса.

Redis: Используется для кэширования данных и запросов, ускоряя процессы поиска и улучшая отклик системы.

PostgreSQL: Основная база данных, которая используется для хранения данных о пользователях, контенте и их взаимосвязях.

ClickHouse: Система для хранения и анализа больших объемов логов поискового модуля, что позволяет анализировать поведение пользователей и улучшать результаты поиска.

Kafka: Шина данных для обработки потоковых данных, которая помогает передавать данные между различными сервисами платформы в реальном времени.

Уровень данных

В поисковом модуле используются несколько категорий данных, которые являются ключевыми для работы с контентом и пользовательскими предпочтениями:

Данные о контенте: Информация о видео, включая название, описание, продолжительность, дату публикации и другие метаданные.

Данные об авторах/каналах: Сведения о каналах, такие как название, описание и время создания, что помогает пользователям находить авторов или каналы с интересующим контентом.

Данные о реакциях пользователей: Лайки, комментарии и другие реакции пользователей на контент, что позволяет учитывать пользовательскую активность в процессе поиска.

Данные о подписках пользователей: Информация о том, на какие каналы или авторов подписаны пользователи, что позволяет учитывать предпочтения при формировании поисковых результатов.

Данные о взаимодействии пользователей с контентом: Сюда включаются просмотры, досмотры, перемотки и другие действия, которые помогают системе точнее предсказывать, что может заинтересовать пользователя.

### Модуль «BI аналитика для стриминговых сервисов»

Модуль «BI аналитика для стриминговых сервисов» предназначен для продуктовой аналитики видео платформ, таких как Rutube. Он предоставляет инструменты для анализа данных о пользователях, контенте и их взаимодействиях, что позволяет улучшать рекомендации, поиск и пользовательский опыт.

Модуль интегрируется с другими системами платформы данных, поисковым и рекомендательным модулями, а также поддерживает гибкость работы с данными через подготовленные дашборды и Adhoc-запросы с использованием SQL и Big Data технологий.

### 

### Назначение модуля «BI аналитика для стриминговых сервисов»

Модуль предназначен для анализа ключевых метрик видео платформ и взаимодействий пользователей. Его основные задачи:

* Мониторинг ключевых метрик (время просмотра, CTR, retention rate).
* Анализ пользовательского поведения.
* Оптимизация поиска и рекомендаций.
* Оценка эффективности контента.
* Поддержка A/B тестирования.
* Интерактивные дашборды и Adhoc-запросы.

### 

### Функциональные возможности модуля «BI аналитика для стриминговых сервисов»

**Визуализация**

Интерактивные дашборды (DAU, MAU, время просмотра).

A/B тестирование.

Автоматические отчеты.

Adhoc аналитика (детализация данных с SQL-запросами).

**Аналитика поиска**

Анализ популярных запросов.

Оценка релевантности результатов.

Анализ нулевых результатов поиска.

**Аналитика рекомендаций**

Оценка CTR, времени просмотра и retention rate.

Персонализация рекомендаций.

**Пользовательская аналитика**

Поведенческая аналитика (пути пользователей, частота возвращений).

Сегментация пользователей (холодные/горячие, демографические данные).

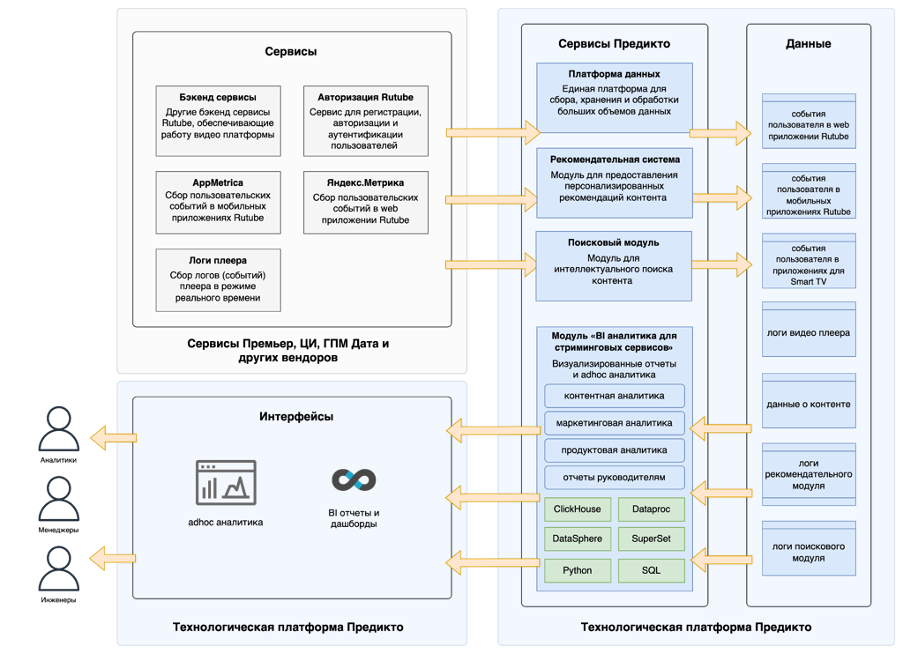
**Аналитика контента**

Определение популярных видео, категорий и трендов.

Оценка влияния метаданных на вовлеченность.

### 

### Архитектура и принцип работы модуля «BI аналитика для стриминговых сервисов»



*рис. 1 - диаграмма концептуальной архитектуры модуля «BI аналитика для стриминговых сервисов»*

Модуль интегрирован с платформой данных и получает информацию из внешних сервисов, включая:

* Авторизацию пользователей.
* A/B тестирование.
* Логи плеера.
* Данные о просмотрах и взаимодействиях.
* Поисковые запросы и результаты.
* Рекомендательные алгоритмы.

**Взаимодействие с другими модулями и сервисами.**

Платформа данных - универсальный модуль, отвечающий за сбор, хранение и обработку данных.

Рекомендательная система - источник данных о рекомендациях для анализа их эффективности.

Поисковый модуль - источник данных о пользовательских запросах и результатах поиска.

Внешние сервисы:

Авторизация Rutube

A/B тестирование

Бэкенд сервисы Rutube

Логи плеера

AppMetrica

Яндекс.Метрика

**Уровень данных**

Данные о контенте - название видео, описание, длительность, время публикации.

Данные об авторах/каналах - название, описание, дата создания.

Данные о подписках - информация о подписках пользователей на авторов и каналы.

Данные о взаимодействиях - просмотры, досмотры, перемотки.

Логи работы - данные о поисковых запросах, ответах и рекомендациях.

### Используемые языки программирования и фреймворки модуля «BI аналитика для стриминговых сервисов»

Базовые технологии

**Язык программирования**: Python

Инфраструктурные программные средства

Базовая ОС: Ubuntu

Система виртуализации: Виртуализация в Yandex Cloud

Контейнеризация:

* Docker
* Kubernetes (Yandex Managed Service for Kubernetes®)

Облачное хранилище: Yandex Cloud (Yandex Object Storage)

BI и визуализация: Apache Superset

Обработка данных:

* Dagster (open source версия, Docker)
* Apache Spark (часть Yandex Data Processing)

Используемые библиотеки: Polars, Pandas, deltalake, PyArrow, Dagster, PySpark

# 

Основными техническими средствами, обеспечивающими функционирование модуля BI аналитики, являются:

ClickHouse - Высокопроизводительная колоночная СУБД для adhoc аналитики по агрегированным витринам данных.

SuperSet - Инструмент для визуализации и анализа данных.

Dataproc (Apache Spark) - Масштабируемая система для распределенной обработки больших объемов данных.

DataSphere: Интерактивный ноутбук для adhoc анализа данных.

# Используемые языки программирования и фреймворки

Язык программирования

Python: Основной язык разработки, который используется для реализации всех компонентов системы, включая обработку данных, машинное обучение и работу с API.

Библиотеки и фреймворки

Обработка данных

Polars: Быстрая и эффективная библиотека для обработки данных в формате таблиц, работающая с большими объемами данных.

Pandas: Один из самых популярных инструментов для работы с данными в табличной форме, обеспечивающий удобный интерфейс для анализа и манипуляции данными.

deltalake: Библиотека для работы с форматом Delta Lake, который используется для управления данными в виде больших файлов и эффективного выполнения запросов.

PyArrow: Библиотека для работы с Apache Arrow, которая используется для обеспечения быстрого и эффективного обмена данными между системами.

Psycopg: Пакет для взаимодействия с PostgreSQL, используемый для работы с реляционными базами данных.

SciPy: Библиотека для научных и инженерных вычислений, обеспечивающая широкий набор математических и статистических функций.

Numba: Библиотека для ускорения вычислений с помощью JIT-компиляции Python кода на языке машинных команд.

Модели машинного обучения

scikit-learn: Популярная библиотека для машинного обучения, которая предоставляет простой и эффективный инструментарий для построения моделей и анализа данных.

PyTorch: Гибкий фреймворк для создания и обучения нейронных сетей, который активно используется для глубокого обучения и искусственного интеллекта.

CatBoost: Библиотека для градиентного бустинга, которая оптимизирована для работы с категориальными данными.

XGBoost: Мощная библиотека для реализации алгоритмов градиентного бустинга, широко используемая для решения задач классификации и регрессии.

LightFBM: Фреймворк для быстрого градиентного бустинга, специально предназначенный для работы с большим объемом данных.

Implicit: Библиотека для создания рекомендательных систем с использованием матричной факторизации.

Optuna: Оптимизационная библиотека для автоматического подбора гиперпараметров моделей машинного обучения.

Обработка текста и NLP

PyMorphy2: Библиотека для морфологического анализа русского языка.

Levenshtein: Библиотека для вычисления расстояния Левенштейна, используется в задачах обработки текста и анализа строк.

Sentence Transformers: Библиотека для преобразования предложений в эмбеддинги, используется для задач обработки естественного языка (NLP).

Операции с графами

Dagster: Фреймворк для разработки и оркестрации рабочих процессов и данных, используемый для построения надежных и масштабируемых систем обработки данных.

Дополнительные инструменты

rbloom: Библиотека для реализации фильтров Блума, применяемая для эффективной проверки принадлежности элемента множеству, что используется в рекомендательных системах и для работы с большими данными.

# Описание технической инфраструктуры

Облачные сервисы и виртуализация

Yandex Cloud: Используется для аренды виртуальных серверов и хранения данных в Yandex Object Storage.

Виртуализация: Используются Yandex Compute Cloud и Docker для контейнеризации и оркестрации с помощью Kubernetes.

Базы данных и кэширование

PostgreSQL: Основная база данных для хранения информации о пользователях и контенте.

Redis: Кэширование данных и запросов.

OpenSearch: Поисковый движок для обработки запросов.

ClickHouse: Хранение и анализ логов.

Машинное обучение

Yandex Data Sphere и Apache Spark для обработки данных и обучения моделей.

Dagster для управления рабочими процессами и данными.

PyTorch, scikit-learn, XGBoost, CatBoost для машинного обучения и построения моделей.

Системы мониторинга.

Prometheus и Grafana для сбора и анализа метрик.

Grafana Loki для сбора и анализа логов.

Инструменты для обработки текста.

PyMorphy2: Морфологический анализ русского языка.

Levenshtein: Вычисление расстояния Левенштейна для улучшения обработки текстовых запросов.

Sentence Transformers: Преобразование предложений в эмбеддинги для улучшения поиска.

Безопасность. API интеграция с внешними сервисами через стандартизированные протоколы. Используется x-authorization-token для авторизации.