

Заказчик:	МТС.Тета, Демиденко Савва Александрович
Название проекта:	Аналитическое хранилище для KION
Исполнители	Сафонов Николай Юрьевич, Чертанов Денис Алексеевич

## ТЕХ. ЗАДАНИЕ

### 1. Вводы системы:

Endpoint: /api/user-event/save

Протокол: http

Метод: POST

Заголовки:

1. Authorization: Bearer <json web token>

Формат тела запроса: JSON

Поля в теле запроса:

1. user\_id - идентификатор пользователя;
2. video\_id - идентификатор видео;
3. event\_time - время события;
4. event\_type - тип события.

### 2. Выводы системы:

Сервис отвечает на запрос /api/user-event/save одним из следующих кодов:

1. 200 – Запрос успешно обработан;
2. 400 – Неправильный формат запроса;
3. 401 – Отсутствует jwt-токен, либо он не валиден;
4. 503 – Сервис не может сейчас обработать запрос из-за перегрузки или внутреннего сбоя.  
Клиент может попытаться переотправить запрос через какое-то время.

### 3. Атрибуты системы:

1. Нагрузка на запись новых событий — 8 000 RPS;
2. Нагрузка на чтение — 250 RPS;
3. Доступность сервиса — 99,9 %.

### 4. Атрибуты системной среды:

Хранилище должно гибко масштабироваться (пиковые нагрузки могут превышать обычные в 3 раза).

## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1. Возможность отправить событие в систему с помощью POST запроса;
2. Возможность аутентификации с помощью JWT;
3. Возможность осуществлять SQL запросы по всем событиям.

## **НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ**

Требования к продукту:

1. Нагрузка на запись новых событий — 8 000 RPS;
2. Нагрузка на чтение — 250 RPS;
3. Доступность сервиса — 99,9 %;
4. Должны храниться все события за все время.

Организационные требования:

1. Использование языка программирования Java или Go;
2. Использование принципов SOLID при разработке;
3. Написание отчета по результатам работы.

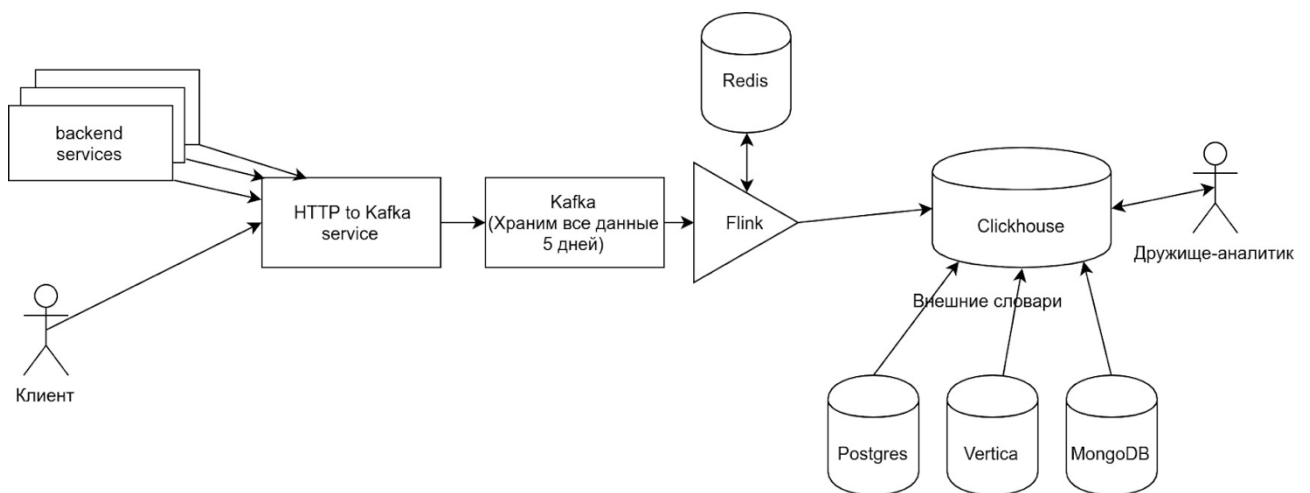
## **ОГРАНИЧЕНИЯ ПРОЕКТИРОВАНИЯ**

Система должна разрабатываться с использованием языка Java или Go.

## Приложение 1. АНАЛИЗ КОНКУРЕНТОВ В ЧАСТИ ФУНКЦИОНАЛА

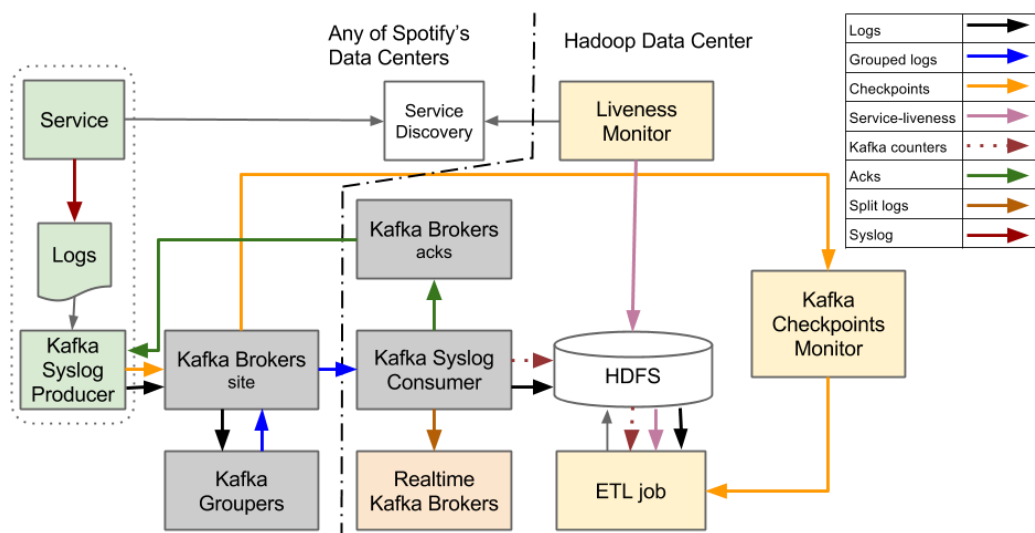
### Ivi

Сервис Ivi использует собственную систему сбора и хранения действий пользователей. Цель их системы - предоставление инструмента анализа данных для аналитиков. Для этих целей используется Clickhouse. Такая система не подходит для решения нашей задачи, поскольку она спроектирована без учета сценария чтения истории пользователя по ключу под высокой нагрузкой. Clickhouse хорошо подходит для выполнения аналитических запросов по большому объему данных, но по результатам наших измерений не очень хорошо справляется с чтением большого количества широких строк с низкой задержкой из-за того, что хранит данные по колонкам.



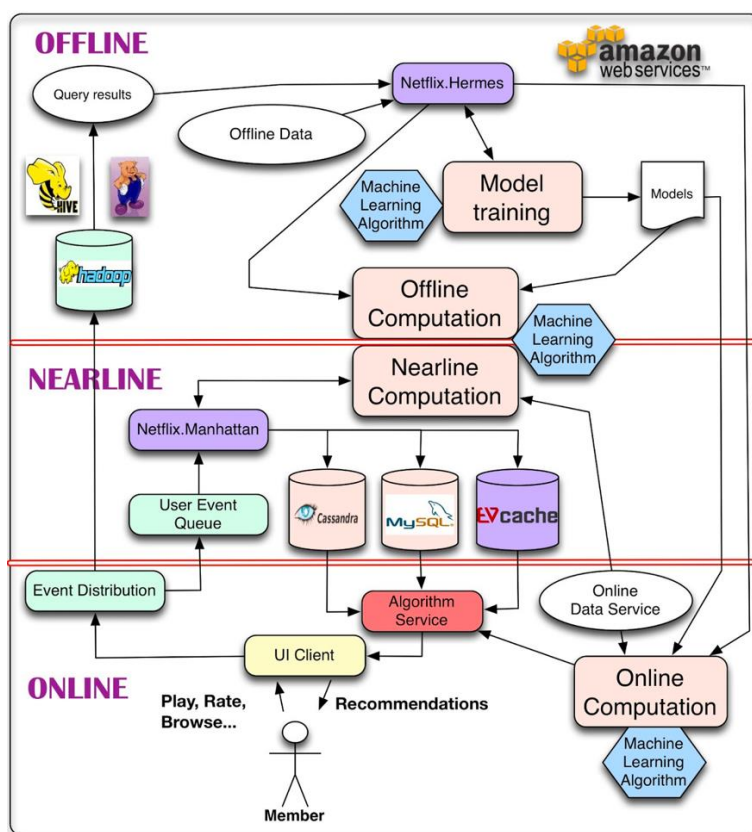
### Spotify

Сервис Spotify собирает поток событий пользователей с помощью своей системы доставки событий. Их система использует кластер Hadoop над HDFS для обработки последних поступивших событий. Такая система не подходит для решения нашей задачи. Hadoop может использоваться для оффлайн аналитики с использованием таких инструментов как например HQL, но не подходит для сценария получения истории пользователя по ключу, который является критическим для решения поставленной задачи.



## Netflix

Сервис Netflix составляет персональные рекомендации. Для этого сервис собирает события пользователей и обрабатывает их в своей системе. В архитектуре системы выделяются два сценария работы с данными пользователей - онлайн и оффлайн вычисления. При проектировании системы учитывалось использование данных в рекомендательных алгоритмах на основе машинного обучения, в связи с чем архитектура состоит из большого количества компонентов, которые избыточны для решения нашей задачи.



Заказчик

Личная подпись

*[Signature]* Данилов С

Ответственный по проекту

Личная подпись

*[Signature]* Данилов Р

Дата:

26 марта 2022