Snowplow
и утерянное руководство

Олег Кочергин

whoami

- внедрял piwik(matomo), дважды snowplow
  и видел много разных велосипедов :)

- software development background since ‘00
- in data since ’16
- dev → data engineer → head of data → cdo
- Positive Technologies
  ex. Alfa-Bank, ex. SberHealth
- SmartData resident

Цель и план доклада

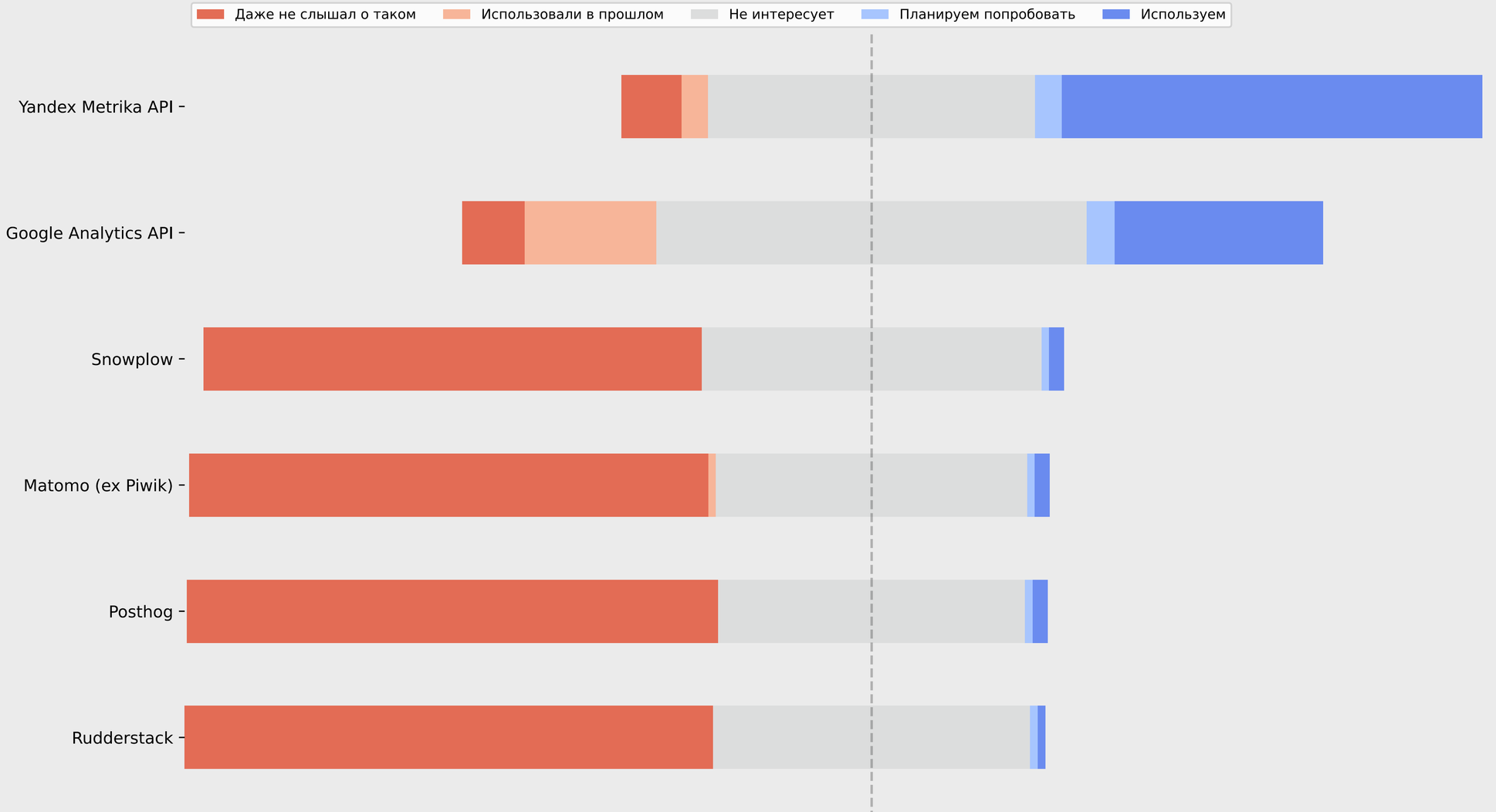
- Рассказать про задачу clickstream
- Про то какими интрументами задача может решаться
- О сложностях создания своих велосипедов
- Deep dive tutorial в Snowplow

Что за зверь такой, clickstream?

- Логи действий пользователя, веб/маркетинговая-аналитика
- Вебхуки сервисов атрибуции типа appsflyer / adjust
- На основе кликстрима считают метрики / воронки и прочим образом анализируют поведение пользователей
- Чтобы что? Чтобы сделать продукт/сайт лучше :)

Про доступные решения

и немного спойлеров



Польза от собственного решения

- Вы владеете данными

Польза от собственного решения

- Вы владеете данными
- Вас не блочит adblock / safari и прочие

Польза от собственного решения

- Вы владеете данными
- Вас не блочит adblock / safari и прочие
- Можно реализовать стриминговые кейсы обработки когда и если надо будет

Польза от собственного решения

- Вы владеете данными
- Вас не блочит adblock / safari и прочие
- Можно реализовать стриминговые кейсы обработки когда и если надо будет
- Это гораздо приятнее чем условный LogsAPI

Про велосипеды

Решение для clickstream можно сделать самостоятельно, но кроличья нора глубже чем кажется:

- Необходимо разработать и поддерживать JS / iOS / Android SDK
- Придумать расширяемый протокол
- Учесть все приколы браузеров с куками и прочим
- Стриминг, приколы с geoip ... итп
- ...

Голосуйте за Snowplow

Snowplow

очень коротко

- Open Source*, года примерно с 2014
- SDK для всего — от JS/Android/iOS до Arduino
- Набор сервисов для получения данных от SDK, обогащения и записи в целевую БД
- Поддержка разных брокеров от Kafka и NSQ до Kinesis и Google PubSub
- Подобие дата-контрактов

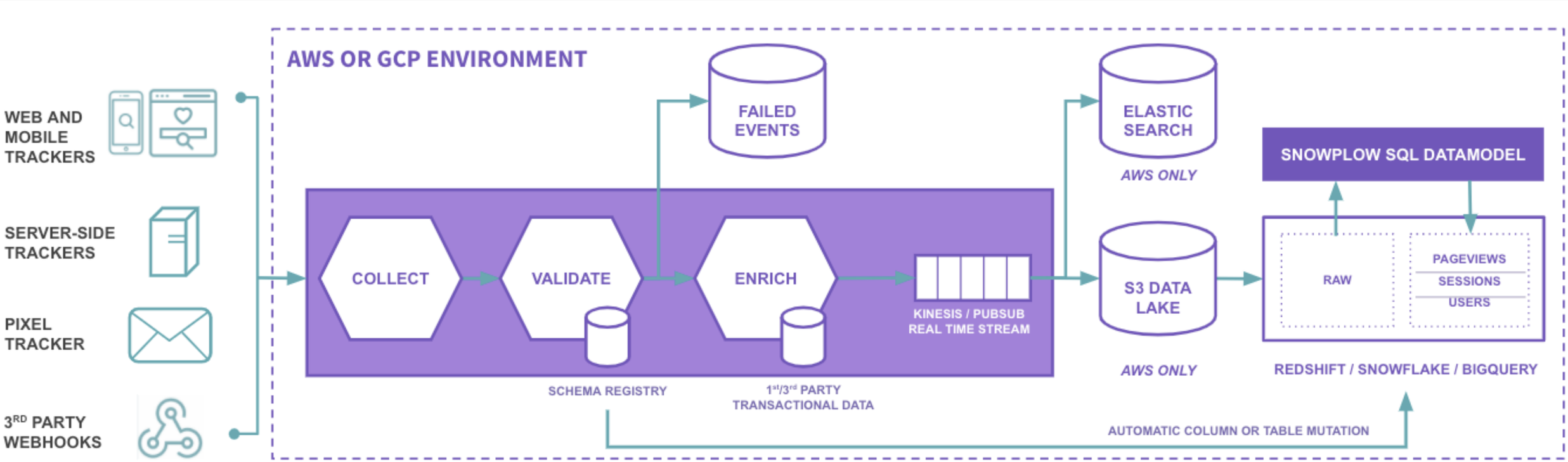
Но есть ньюанс

Пара слов про лицензию

- В конце 2023 - начале 2024 проект поменял лицензию и, фактически, запретил коммерческое использование

- Варианты:
  1) Использовать версию 2.10.0
  2) Принять риски
  3) OpenSnowcat

Архитектура



Компоненты Snowplow

- **SDK**

Компоненты Snowplow

- **SDK**
- **Collector**
  Принимает данные от трекеров, валидирует формат и укладывает в буффер (kafka / nsq / kinesis / pubsub) в чистый (всё что прошло валидацию) и грязный топики с сырьем. Данные в топиках сериализованы с помощь Thrift.

Компоненты Snowplow

- **SDK**
- **Collector**
  Принимает данные от трекеров, валидирует формат и укладывает в буффер (kafka / nsq / kinesis / pubsub) в чистый (всё что прошло валидацию) и грязный топики с сырьем. Данные в топиках сериализованы с помощь Thrift.
- **Enrich**
  Достаёт данные из чистого сырого топика, обогащает (от geoip и парсинга user-agent до кастомных трансформаций на JS), раскладывает по канонической модели и записывает в чистый обогащенный топик (если всё ок) или в грязный. Формат данных — TSV (ссылка)

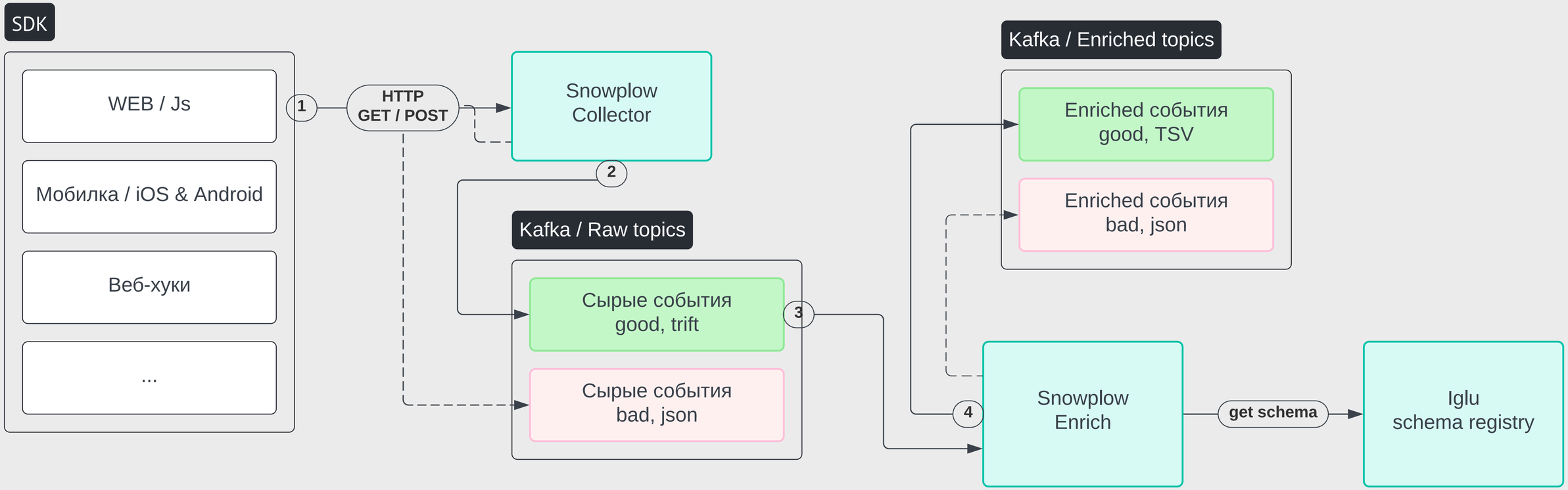
Компоненты Snowplow

- **SDK**
- **Collector**
  Принимает данные от трекеров, валидирует формат и укладывает в буффер (kafka / nsq / kinesis / pubsub) в чистый (всё что прошло валидацию) и грязный топики с сырьем. Данные в топиках сериализованы с помощь Thrift.
- **Enrich**
  Достаёт данные из чистого сырого топика, обогащает (от geoip и парсинга user-agent до кастомных трансформаций на JS), раскладывает по канонической модели и записывает в чистый обогащенный топик (если всё ок) или в грязный. Формат данных — TSV (ссылка)
- **Loader**
  Сервисы-загрузчики обогащенного после Enrich этапа потока событий в целевую БД

Компоненты Snowplow

- **SDK**
- **Collector**
  Принимает данные от трекеров, валидирует формат и укладывает в буффер (kafka / nsq / kinesis / pubsub) в чистый (всё что прошло валидацию) и грязный топики с сырьем. Данные в топиках сериализованы с помощь Thrift.
- **Enrich**
  Достаёт данные из чистого сырого топика, обогащает (от geoip и парсинга user-agent до кастомных трансформаций на JS), раскладывает по канонической модели и записывает в чистый обогащенный топик (если всё ок) или в грязный. Формат данных — TSV (ссылка)
- **Loader**
  Сервисы-загрузчики обогащенного после Enrich этапа потока событий в целевую БД
- **Iglu Schema registry**
  Хранит и отдает по запросу enrich/loader JSON-схемы событий

Еще одна архитектура



Пара слов про структуру событий

aka Canonical Event

- Примерно 130 полей
- **Общие для всех / переиспользуемые**
  - Application fields
  - Date / time fields
  - Event / transaction fields
  - Snowplow version fields
  - User-related fields
  - Device and operating system fields
  - Location fields
  - IP address-based fields
  - Metadata fields
  - Marketing / traffic source fields
- **Специфичные для веба**
  - Page Fields
  - Document fields
  - Browser fields
- **Специфичные для конкретных типов событий**
  - Page views
  - Page pings
  - E-commerce transactions
  - Structured events
  - Self-describing events

https://docs.snowplow.io/docs/understanding-your-pipeline/canonical-event/

Пара слов про структуру событий

Custom context

- Расширяемость модели событий реализована через так называемый custom context
- Это отдельное поле в канонической схеме события
- Которое содержит массив json объектов с указанием на схему конкретного объекта

```
tracker.addGlobalContexts({
    tag: 'my-new-tag',
    globalContexts: [
        {
            schema: 'iglu:com.snowplowanalytics.snowplow/ad_impression/jsonschema/1-0-0',
            data: {impressionId: 'my-ad-impression-id'},
        },
    ]
});
```

Подробнее про компоненты

Collector

Что это: Scala сервис который принимает события, минимально валидирует и обогащает

- обогащает: дописывает IP адрес / User Agent / HTTP Headers и Cookies
- принимает события как с помощью GET, так и POST запросов
  - GET полезен для пикселей: например, встроить в письмо и трекать открытия
  - POST для отправки пачки событий разом или для веб-хуков
- умеет в редиректы: например, трекать переходы по ссылкам
- сбрасывает куку на свой домен для возможности кросс-доменного отслеживания
- если всё хорошо с валидацией, то запаковывает сырое событие в thrift формат и записывает в топик
- если всё плохо: например битый json или проблемы с валидацией то сообщение в формате json улетит в “плохой” топик.

https://github.com/snowplow/stream-collector/blob/master/examples/config.kafka.extended.hocon

Подробнее про компоненты

Iglu Schema Registry

Что это: сервис или, в самом простом варианте, просто веб-сервер со статикой, который хранит описание всех схем контекстов и кастомных событий в формате JSON Schema.

Используется сервисом сервисом Enrich для валидации и обогащения событий и стандартными сервисами Loader для генерации схем БД.

⚠️ *Крайне важно, чтобы сервис был развернут отказоустойчиво, поскольку невозможность сервисом Enrich обновить кеш схем создаст проблемы валидации событий: они, конечно, не будут потеряны безвозвратно, но попадут в “плохие” топики и потребуется их оттуда спасать*

- Можно использовать общедоступный iglucentral.com
- Можно задеплоить свой, как статический веб сайт, через github/gitlab pages, s3 static вебсайт или прост контейнер nginx
- В конфигах сервисов можно использовать несколько schema registry серваков

com.sendgrid/bounce/jsonschema/3-0-0


```
{
  "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/1-0-0#",
  "description": "Schema for a SendGrid bounce event. Property descriptions derived from the SendGrid
documentation: https://sendgrid.com/docs/for-developers/tracking-events/event/",
  "self": {
    "vendor": "com.sendgrid",
    "name": "bounce",
    "version": "3-0-0",
    "format": "jsonschema"
  },
  "type": "object",
  "properties": {
    "ip": {
      "description": "The IP address used to send the email. For open and click events, it is the IP
address of the recipient who engaged with the email.",
      "type": "string",
      "maxLength": 45
    },
    "timestamp": {
      "description": "The timestamp of when the message was sent",
      "type": "string",
      "format": "date-time"
    },
    "email": {
      "description": "The email address of the recipient",
      "type": "string",
      "maxLength": 320
    },
```

Подробнее про компоненты

Iglu Schema Registry nginx docker

```
FROM alpine/git AS source
WORKDIR /snowplow
RUN git clone --depth 1 https://github.com/snowplow/iglu-central.git

FROM nginx:alpine
RUN sed -i '/default_type/c\    default_type text/plain;' /etc/nginx/nginx.conf
COPY --from=source /snowplow/iglu-central/schemas/ /usr/share/nginx/html/schemas
```

Подробнее про компоненты

Enrich

Что это: очередной Scala сервис, его задача — взять событие из сырого топика, провалидировать с помощью JSON схем из Iglu и “обогатить” его.

Реализованные “обогащалки”:

- MaxMind IP
- Campaign attribution / referer parser / iab spiders&robots
  Эти трансформеры позволяют извлечь в отдельные поля маркетинговые метки, проставить тип источника трафика и “пробить робота по айпи”
- User-Agent, два типа парсеров: YAUAA и более простой UA parser
- Cookie / HTTP Header extractor
- Custom JS transform

https://docs.snowplow.io/docs/enriching-your-data/available-enrichments/

Подробнее про компоненты

Enrich / пример конфигурации

```
{
  "schema": "iglu:com.snowplowanalytics.snowplow/referer_parser/jsonschema/2-0-0",
  "data": {
    "name": "referer_parser",
    "vendor": "com.snowplowanalytics.snowplow",
    "enabled": true,
    "parameters": {
      "internalDomains": [],
      "database": "referers-latest.json",
      "uri": "https://s3-eu-west-1.amazonaws.com/snowplow-hosted-assets/third-party/referer-parser/"
    }
  }
}
```

Подробнее про компоненты

Loaders

Компоненты, которые берут обогащенные события из брокера и записывают в целевую БД

- Реализованы
  - облачные для Databricks/Snowflake/BigQuery
  - S3 / Lakehouse
  - Postgres

https://docs.snowplow.io/docs/pipeline-components-and-applications/loaders-storage-targets/

Подробнее компоненты

Loaders

Но я их использовать не рекомендую :)

Подробнее компоненты

Loaders

Но я их использовать не рекомендую :)

- Они мутируют целевую схему
  (да, стриминговая джоба которая делает alter на схему)

Подробнее компоненты

Loaders

Но я их использовать не рекомендую :)

- Они мутируют целевую схему
  (да, стриминговая джоба которая делает alter на схему)
- Лоадер для кликхауса в инкубаторе последние 3 года и не обновлялся

Ооокэй, как же тогда загружать данные?!

Грузим ~~снег~~ события

Варианты

- Spark / Flink / Kafka Connect джоба
  1) читаем из топика, работаем с каноническим TSV через analytics sdk*
  2) реализуем запись куда нам нужно напрямую
  (ребята из ДетМира завтра расскажут про Spark → Hadoop)

https://docs.snowplow.io/docs/destinations/analytics-sdk/

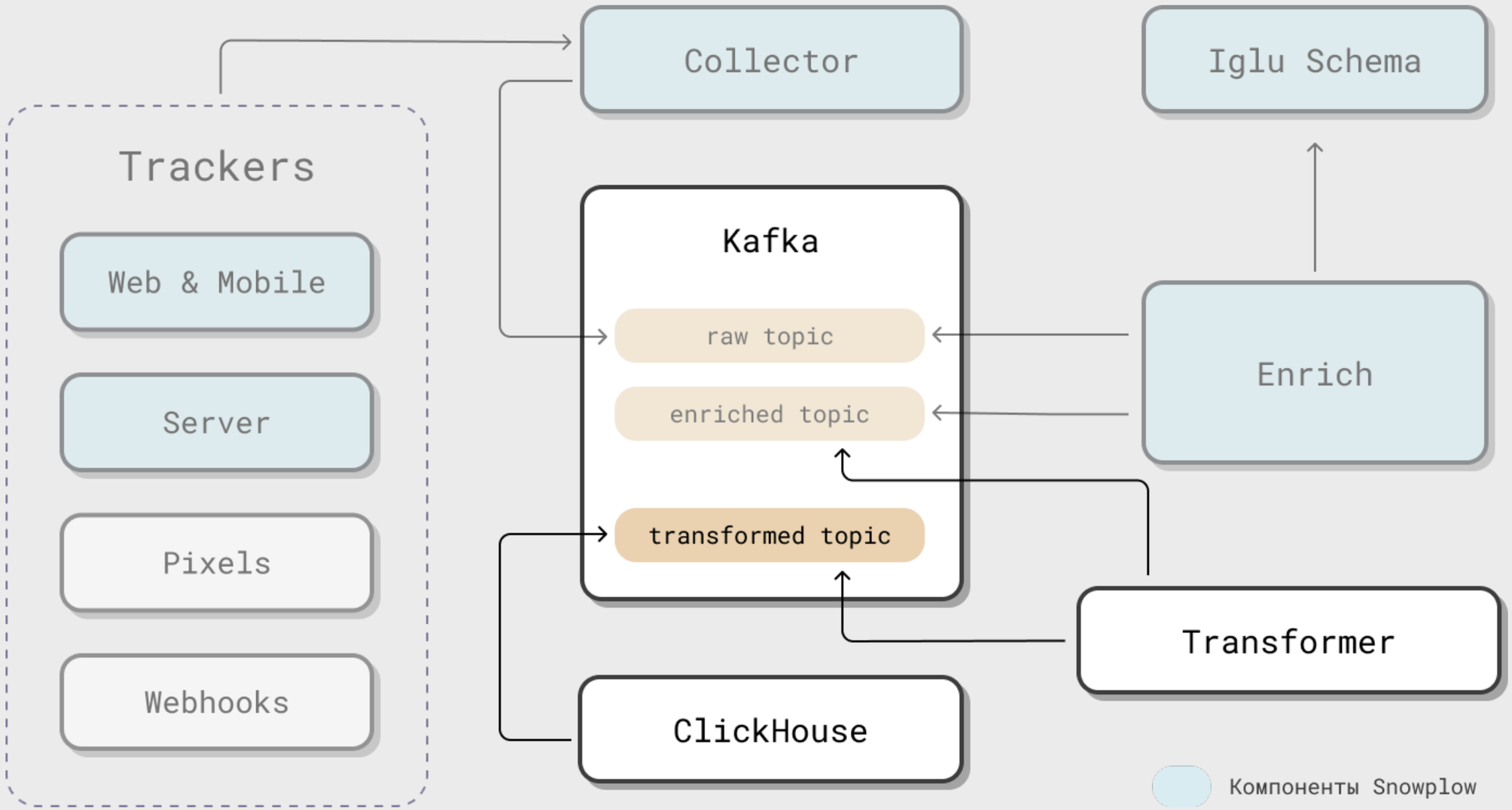
Грузим ~~снег~~ события

Варианты

- Spark / Flink / Kafka Connect джоба
  1) читаем из топика, работаем с каноническим TSV через analytics sdk*
  2) реализуем запись куда нам нужно напрямую
  (ребята из ДетМира завтра расскажут про Spark → Hadoop)

- Faust джоба, отдельный топик и clickhouse kafka engine
  1) читаем из топика, работаем с каноническим TSV через analytics sdk*
  2) пишем JSON с небольшой коррекцией полей в отдельный топик
  3) clickhouse kafka engine
  4) profit

https://docs.snowplow.io/docs/destinations/analytics-sdk/


Collector
Iglu Schema
Trackers
Web & Mobile
Server
Pixels
Webhooks
Kafka
raw topic
enriched topic
transformed topic
Enrich
Transformer
ClickHouse
Компоненты Snowplow

Бест-практисы и прочие советы

в collector сервисе
включайте партишнинг по IP

в collector сервисе включайте партишнинг по IP

и не забывайте о том, чтобы ваш ingress прокидывал оригинальный IP

не забывайте про DLQ топики
monitoring, replay, итп

необходимо придумать свою архитектуру загрузки в хранилку

необходимо придумать свою архитектуру загрузки в хранилку

потом опенсорснуть и рассказать на smartdata

в мобильных sdk
не забывайте про flush
перед “закрытием приложения”

в мобильных sdk не забывайте про flush перед “закрытием приложения”

или не используйте отправку через POST

вам могут и будут отправлять
события из далекого прошлого
или будущего

Как итог

- Свой трекинг нужен и полезен, даже на внутренних продуктах
- Хорошо взвесьте за/против прежде чем начинать разработку своего велосипеда

Вопросы?
Наброски?!