CDC от источника до хранилища: как в банке Синара построили CDC с применением продуктов Arenadata

Иван Клименко

Фархад Замалетдинов

ARENADATA

О себе

Основное направление – интеграционные решения в процессах наполнения корпоративных хранилищ данных. Специализируюсь на сервисах Kafka и Nifi.

Строил интеграции для наполнения КХД в Аскона, Синара.

Сейчас архитектор в Arenadata.

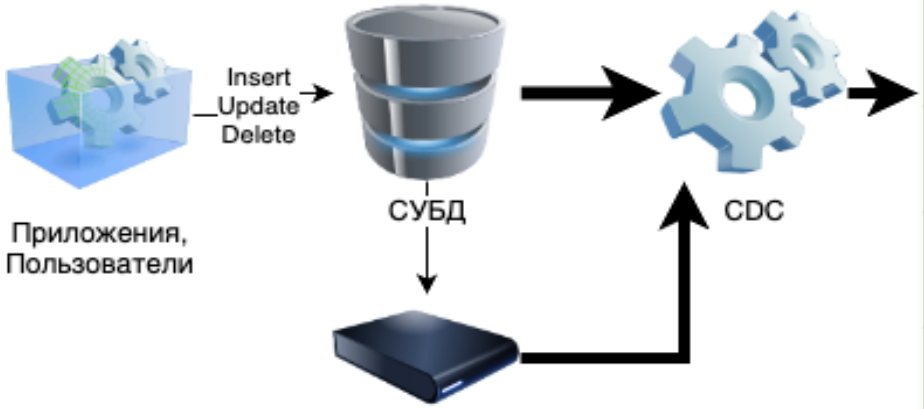
Последние 5 лет — разработчик хранилищ данных. В Банке Синара развиваю CDC и Nifi в команде, разрабатывающей DWH на базе продуктов от Arenadata и PostgreSQL

О чем говорим сегодня

- Что такое CDC и зачем нужна эта технология
- Организация хранилища данных в банке «Синара»
- Как строили загрузку и с какими трудностями столкнулись
- Внедрение CDC. Почему выбрали Debezium
- Организация пайплайна доставки данных в Greenplum от Oracle и его модификации
- Трудности и как мы их преодолели
- Перспективы, планы

ARENADATA

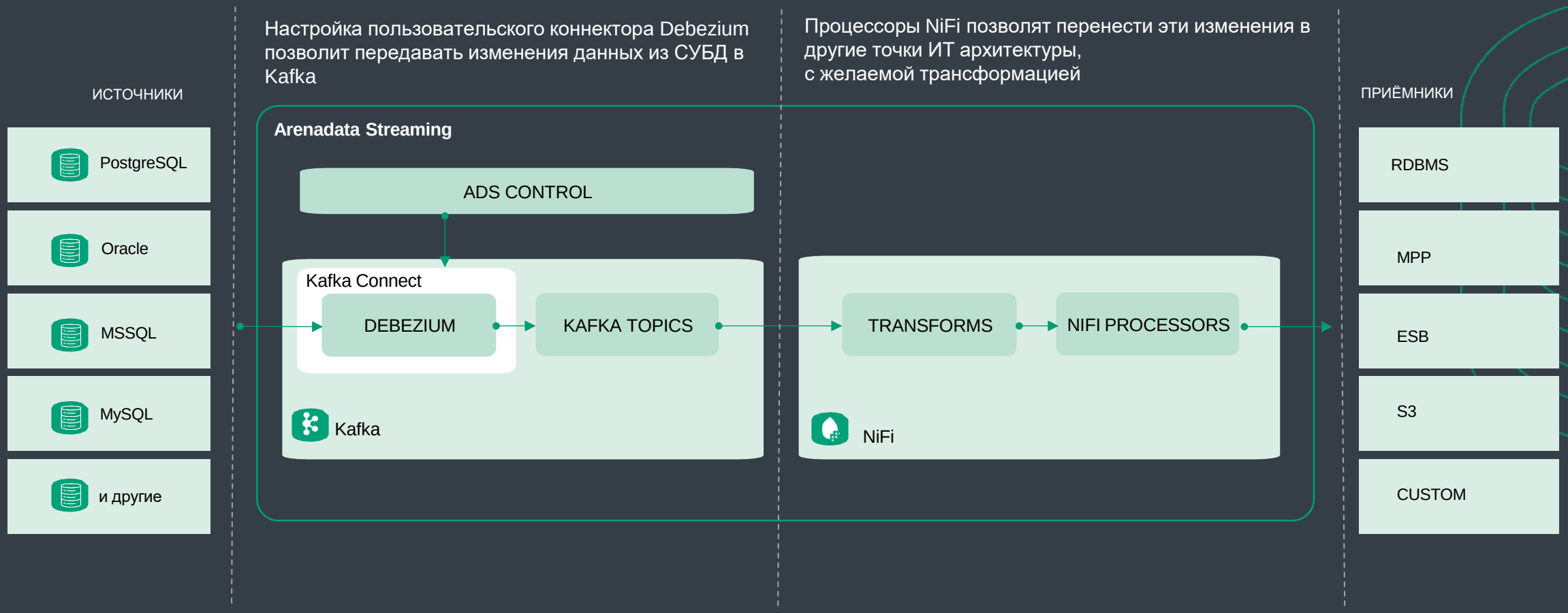
Что такое CDC



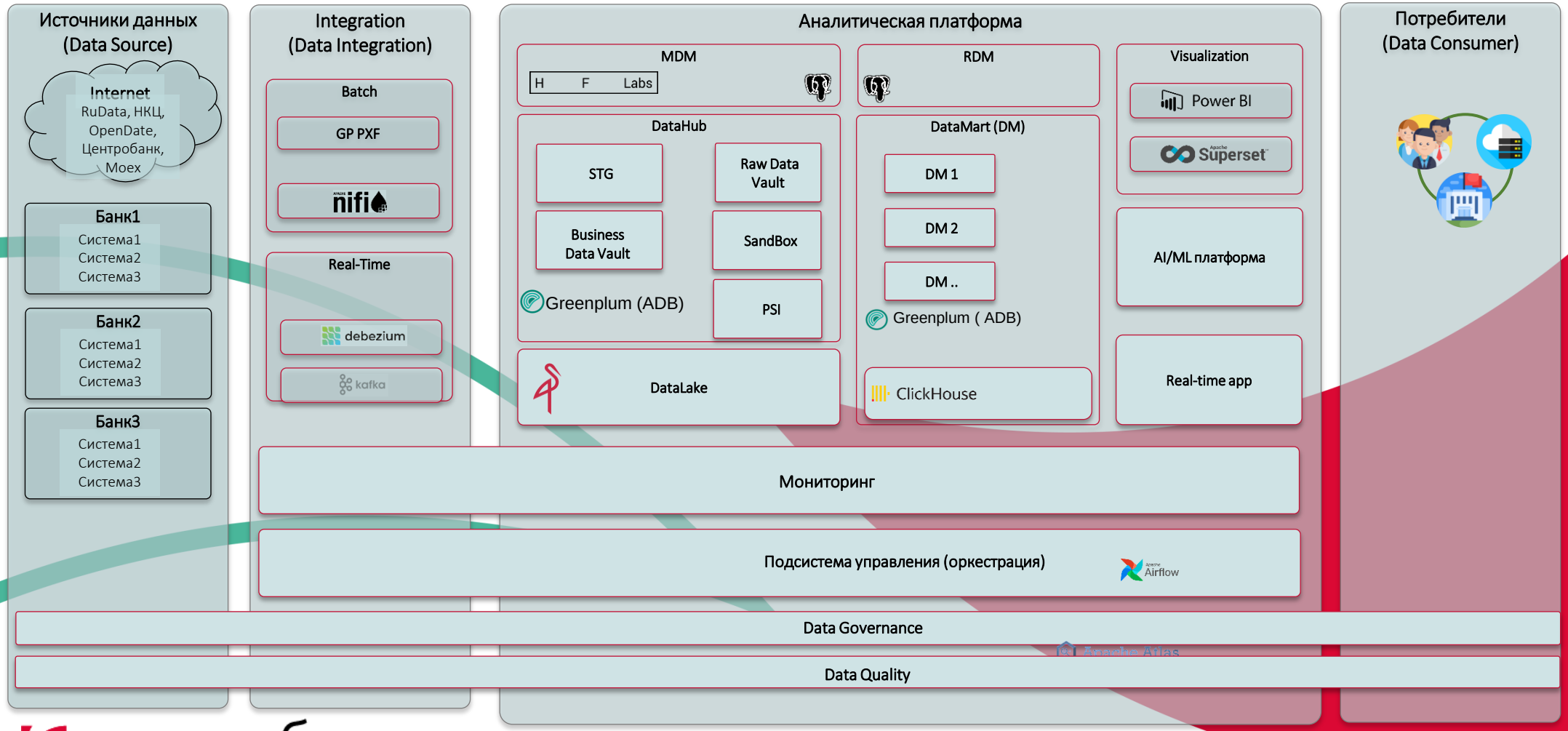
Operation	Data	Metadata
Insert	{"before":null,"after": {"id":10}}	{"table":"some_table", "schema":"some_schema", "offset":100}
Update	{"before": {"id":10},"after": {"id":20}}	{"table":"some_table", "schema":"some_schema", "offset":120}
Delete	{"before": {"id":20},"after":null}	{"table":"some_table", "schema":"some_schema", "offset":130}
DDL	ALTER TABLE...	{"table":"some_table", "schema":"some_schema", "offset":200}

```
{
  "payload":
   {
     "before":null,
     "after" : {
       "id": 10
     }
   },
  "op":"c",
  "source":{
    "table":"some_table",
    "schema":"some_schema",
    "offset": 100
  }
}
```

Типовая схема CDC с компонентами Arenadata Strreaming



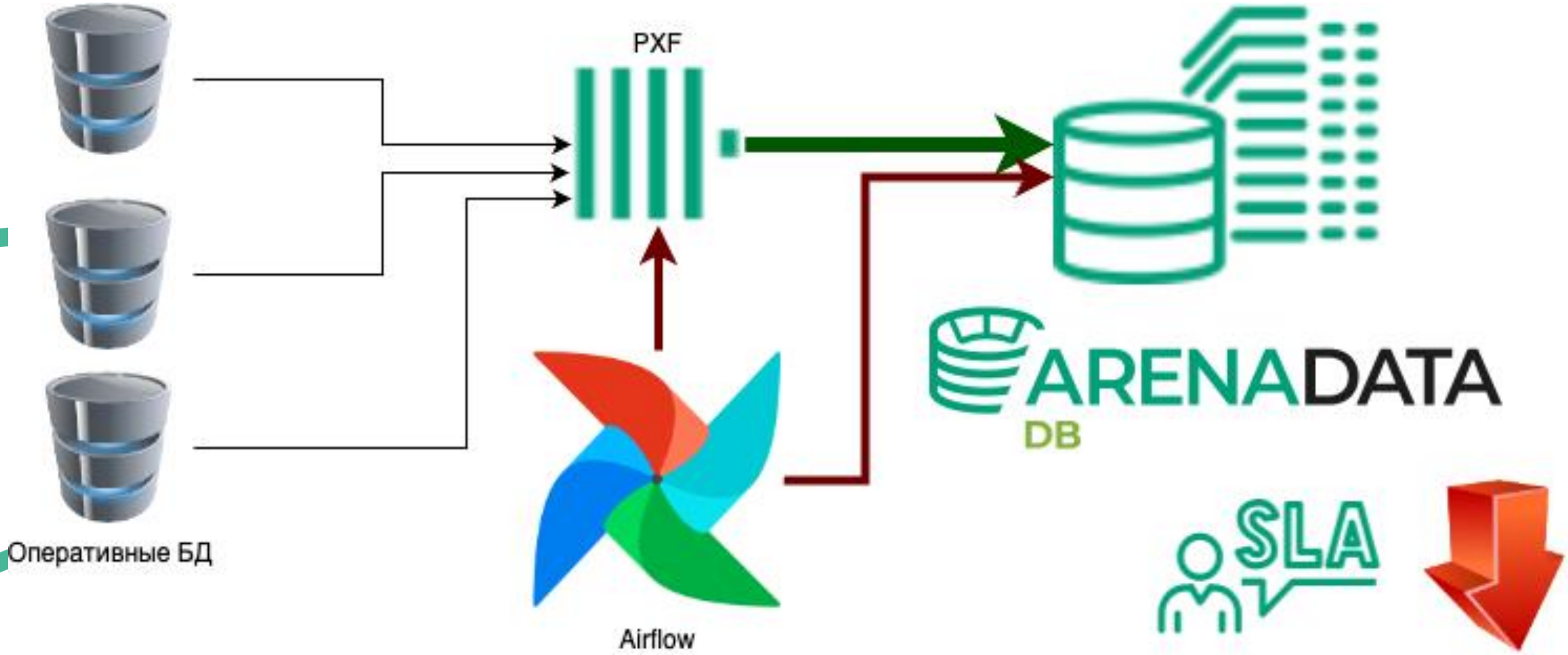
Аналитическая платформа в банке Синара



синара банк

AD

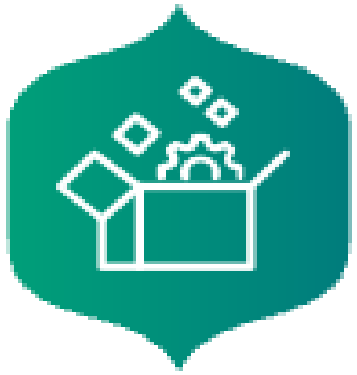
Как наполняли хранилище GreenPlum



синара банк

AD

Почему Debezuim, а не готовое решение от вендора?



OpenSource

Стабильное развитие

Защита от санкций

Комьюнити

Экспертиза

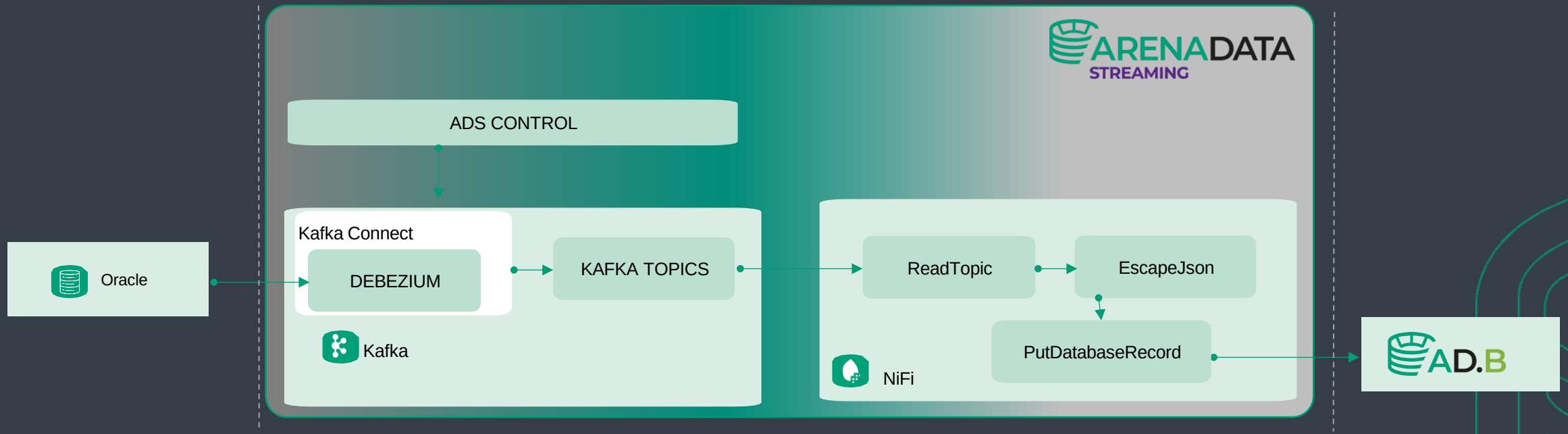
Документация

Известный API

Доступ к коду

Поддержка нескольких СУБД

Обкатка архитектуры



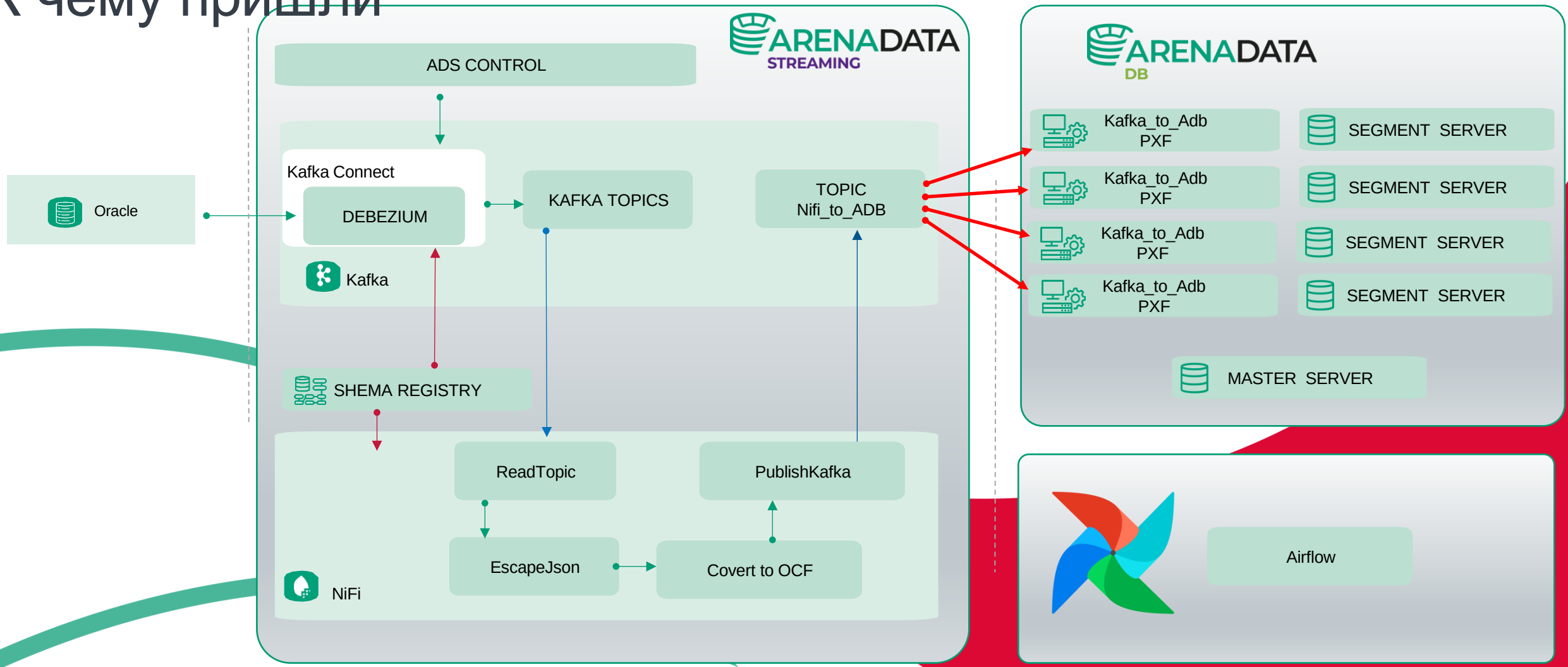
Плюсы:
Универсальный поток обработки данных
Одна универсальная таблица для всех поступающих данных

Минусы:
JSON занимает много места

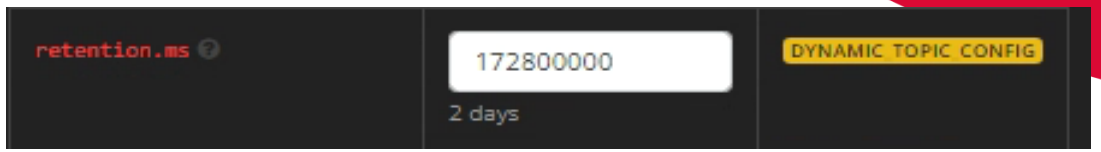
Грузить в GreenPlum через мастер очень плохо

SLA был еще хуже, чем вначале

К чему пришли



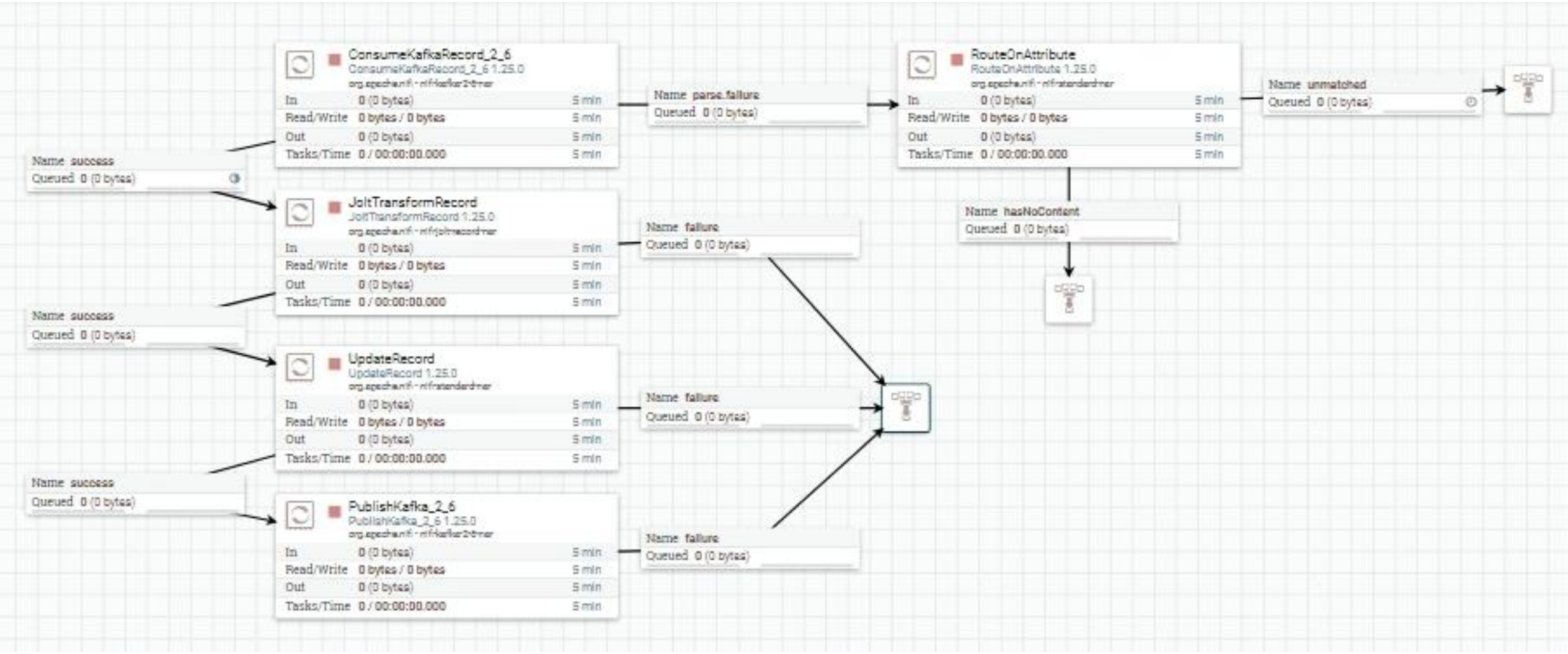

```
"key.converter": "io.confluent.connect.avro.AvroConverter",
```



синара банк

AD

Как организован пайплан в Nifi



Как реализована таблица и вьюхи

```sql
CREATE TABLE src_geb_3card.dbz_kafka_topic_listener (
    headers text NULL,
    topic text NULL,
    "partition" int4 NULL,
    "offset" int4 NULL,
    "timestamp" int8 NULL,
    "before" text NULL,
    "after" text NULL,
    "version" text NULL,
    connector text NULL,
    "name" text NULL,
    ts_ms int8 NULL,
    "snapshot" text NULL,
    db text NULL,
    "sequence" text NULL,
    "schema" text NULL,
    "table" text NULL,
    txid text NULL,
    scn text NULL,
    commit_scn text NULL,
    lcr_position text NULL,
    rs_id text NULL,
    ssn int4 NULL,
    redo_thread int4 NULL,
    user_name text NULL,
    op text NULL,
    value int8 NULL,
    "transaction" text NULL,
    hashdiff text NULL,
    created_at text NULL,
    dml_dttm timestamp NULL,
    upd_dttm timestamp NOT NULL DEFAULT timezone('utc'::text, now())
)
WITH (  appendonly=true, orientation=row, compresstype=zstd, compresslevel=5)
DISTRIBUTED BY (hashdiff)
PARTITION BY RANGE(dml_dttm)
```

```sql
CREATE MATERIALIZED VIEW src_geb_3card.vm_dbz_n01
TABLESPACE pg_default
AS SELECT
        CASE
            WHEN d.op = 'd'::text THEN d.before
            ELSE d.after
        END AS json_value,
    d.scn,
    d.commit_scn,
    d."offset",
        CASE
            WHEN d.op = 'd'::text THEN 'D'::text
            WHEN d.op = 'c'::text THEN 'I'::text
            WHEN d.op = 'u'::text THEN 'U'::text
            ELSE d.op
        END AS dml_tp,
    d.dml_dttm,
    d.hashdiff
   FROM src_geb_3card.dbz_kafka_topic_listener d
  WHERE d."table" = 'N01'::text
  AND d.dml_dttm > '2024-07-26 19:06:35'::timestamp without time zone
WITH DATA
DISTRIBUTED BY (hashdiff);
```

Код создания типичный для источника и легко может быть автоматизирован.

синара банк

AD

Скрипты, позволяющие подключить топик как внешнюю таблицу и наладить обмен

```
DROP SERVER IF EXISTS kafka_server CASCADE;
CREATE SERVER kafka_server
FOREIGN DATA WRAPPER kadb_fdw
OPTIONS (
k_brokers 'bds-ads1:9092,bds-ads2:9092,bds-ads3:9092');
```

```
DROP FOREIGN TABLE IF EXISTS kafka_table;
CREATE FOREIGN TABLE kafka_table(a INT, b TEXT)
SERVER kafka_server
OPTIONS ( format 'avro', k_topic
'topic_adb_to_kafka',k_consumer_group
'group',k_seg_batch '100',k_timeout_ms '1000');
```

```
INSERT INTO src_geb_3card.dbz_kafka_topic_listener
SELECT * FROM src_geb_3card.ext_dbz_kafka_topic_listener;
```

синара банк

AD

Перспективы

Подключить все источники в банковской группе, в том числе на других СУБД (PostgreSQL, MsSQL)

Автоматизировать процесс генерации представлений на основе изменений схем данных

Наладить CI/CD процесс для доставки новых конфигураций коннекторов Debezium

синара банк

AD