



Оmnиканальность

## Meet-up команды OMNI-5. Служебный Сервис «Стриминг в ПД»

Стрим «Развитие технологических сервисов  
омниканальной платформы»

январь 2022





# Контакты

---

## Команда

ОМНИ-5

## ИТ лидер команды

Исаев Павел Александрович [pisaev@vtb.ru](mailto:pisaev@vtb.ru)

## Архитектор

Мамврийский Сергей Юрьевич [smamvriiskii@vtb.ru](mailto:smamvriiskii@vtb.ru)

## Аналитик

Астраханский Алексей Юрьевич [astrahanskii@vtb.ru](mailto:astrahanskii@vtb.ru)

# Сервисы

---

1425. Прикладной журнал

1609. Стриминг в Платформу данных



# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Термин	Определение
ФЛК	Форматно-логический контроль
ЛМД	Логическая модель данных
ФМД	Физическая модель данных
СУБО	Система устойчивых бизнес-операций
ПД	Платформа данных
CDC	(англ. Change Data Capture) Захват изменений данных
ПМИ	Программа и методика испытаний
Агрегат	Кластером связанных сущностей, рассматриваемых как единица для изменения данных. Например, агрегат «Универсальная заявка»



# ЦЕЛИ СЕРВИСА

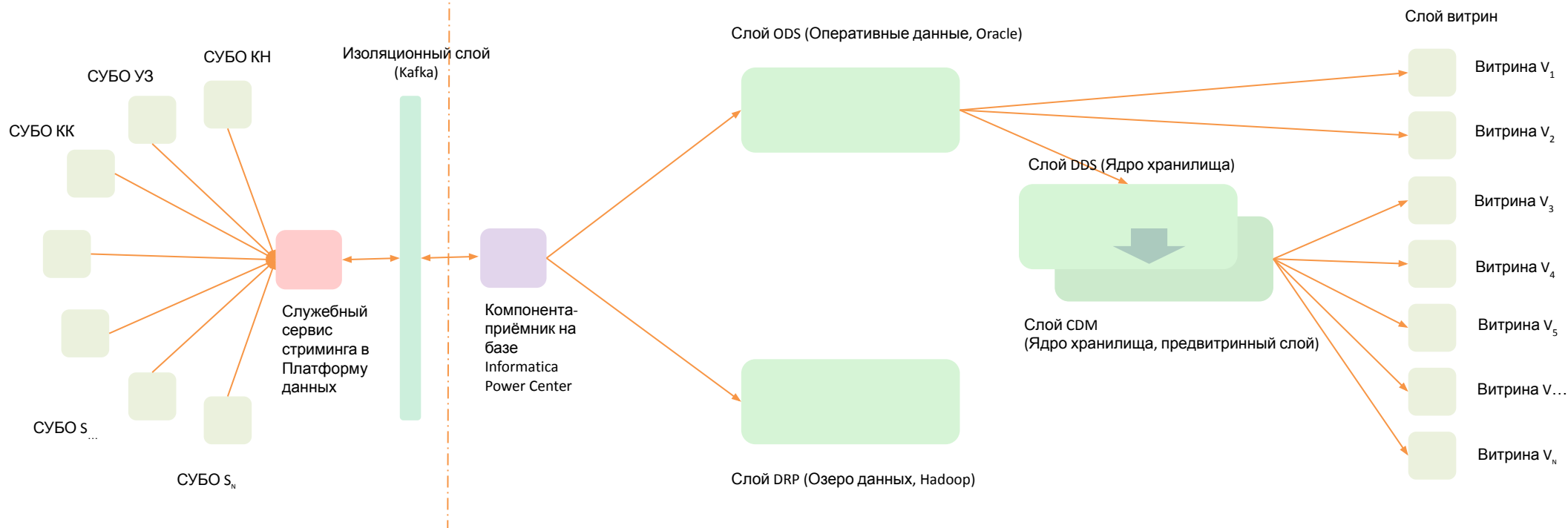


Служебный сервис Стриминг в платформу данных служит для непрерывной передачи всех подтвержденных изменений от СУБО к потребителю данных изменений (например Платформа данных)

Изменения передаются в виде сообщений формализованного вида.

**Платформа  
Оmnikanальность**

**Платформа  
Данных**





# ТРЕБОВАНИЯ К СЕРВИСУ

## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ:

- ✓ Сервис должен предоставлять возможность передачи данных из СУБО в Платформу Данных
- ✓ Сообщения об изменениях должны отправляться в ПД в режиме онлайн
- ✓ Сервис должен предоставлять информацию о статусе доставки сообщений в ПД
- ✓ Сервис должен предоставить механизм ФЛК для передаваемых сообщений в ПД
- ✓ Сервис должен предоставить компонент по управлению схемами сообщений СУБО

## Дополнительные требования со стороны ПД:

- ✓ Иметь изоляционный слой между СУБО и Платформой Данных (не работаем напрямую с каждым сервисом отдельно)
- ✓ Получать консистентную информацию в согласованном виде (формат сообщения согласован, сообщение СУБО соответствует согласованной схеме)
- ✓ Не заниматься определением какая СУБО является мастером для атрибута или сущности
- ✓ Получение служебных сообщений от СУБО при работе со слоем ODS

## НЕ ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ:

- ✓ Отказ или неработоспособность сервиса стриминга не влияет на работоспособность СУБО
  - ✓ Доступность сервиса 99,99%
  - ✓ RTO = 2 мин
  - ✓ RPO = 0\*
- \* при условии организации «Гарантированной Доставки»





## 1. ВЛАДЕЛЕЦ ЛМД - АРХИТЕКТУРА ДАННЫХ

- Архитектура Данных осуществляет ведение ЕМД и ЛМД
- Архитектура Данных должна обеспечить ведение специальных значений для атрибутов, которые обязательные в ЛМД, но могут не содержаться в конкретной СУБО
- ЛМД обновляется по правилам «Прямой совместимости»: возможно добавление атрибутов, удаление необязательных атрибутов.

## 2. ВЛАДЕЛЕЦ КОНТРАКТА - СУБО

- СУБО отвечать за соответствие отправляемых данных согласованному контракту
- Версионирование Контракта осуществляется по правилам версионирования Публичного API
- СУБО отвечает за создание и публикацию Схем Сообщений в Реестр Схем Сообщений

## 3. СООБЩЕНИЕ ДЛЯ ПД – СУЩНОСТЬ ЛМД (ЧАСТЬ АГРЕГАТА)

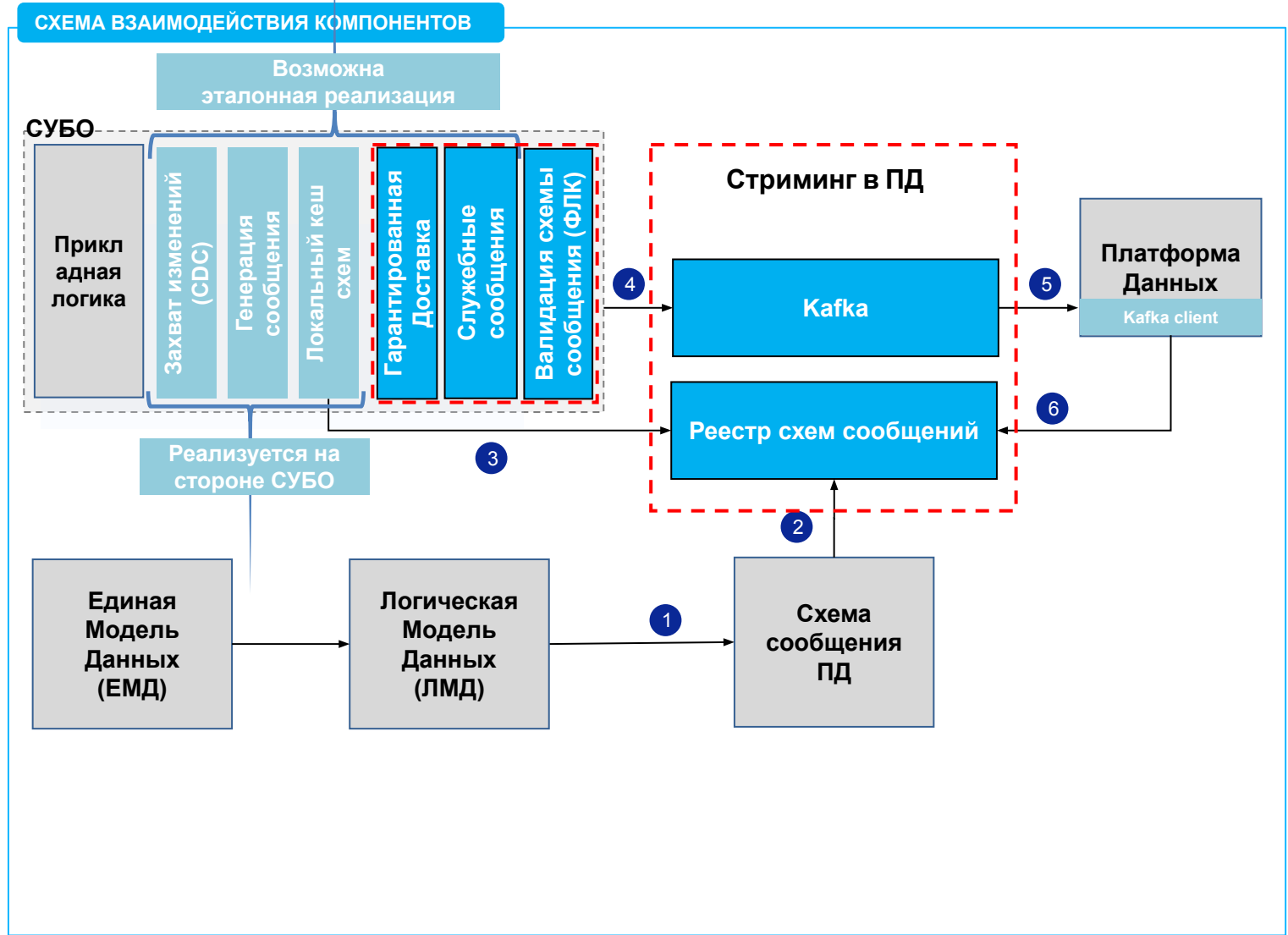
- Сообщение является логически целостной сущностью одного из агрегатов ЛМД.
- Для Сообщения задается Схема Сообщения, на основе которой происходит ФЛК

## 4. СЛУЖЕБНЫЙ СЕРВИС СТРИММИНГА

- Предоставляет компонент, который осуществляет ФЛК исходящего сообщения на стороне СУБО
- Публикует на Портале Разработки эталонную реализацию паттерна «Transactional Outbox», как часть механизма «Гарантированной Доставки»



# КОНЦЕПТУАЛЬНАЯ СХЕМА (БАЗОВЫЙ ВАРИАНТ)



## ОПИСАНИЕ ВЗАИМОДЕЙСТВИЯ

**Сервис, предоставляющий механизм репликации данных в Платформу данных**

- 1 Формирование из ЛМД «Схемы сообщения ПД»
- 2 Публикация «Схемы сообщения ПД» в «Реестр схем сообщений»
- 3 Загрузка «Схемы сообщения ПД» для проверки исходящего сообщения (форматный контроль)
- 4 Отправка исходящего сообщения для ПД\*
- 5 Получение входящего сообщения от СУБО
- 6 Загрузка «Схемы сообщения ПД» для проверки входящего сообщения (форматный контроль)

\* - Открытый вопрос №



# НЕОБХОДИМЫЕ ФУНКЦИИ ДЛЯ СУБО

## 1. ЗАХВАТ ИЗМЕНЕНИЙ (CDC)

- Необходимо реализовать функционал, который захватывает данные бизнес-транзакции для последующей генерации сообщения в Стриминг
- ✓ Эталонная реализация – получение Вектор изменений

## 2. ГЕНЕРАЦИЯ СООБЩЕНИЯ

- Реализация функционал по трансформации бизнес-данных СУБО (ФМЛ) в сообщения для ПД (ЛМД)
- СУБО-специфичная функция. Эталонная реализация не запланирована

## 3. ЛОКАЛЬНЫЙ КЕШ СХЕМ

- Необходимо реализовать механизм кеширования схем сообщений на стороне Поставщика, чтобы минимизировать зависимость от доступности «Реестра схем сообщений» при обработке каждого сообщения

## 4. ГАРАНТИРОВАННАЯ ДОСТАВКА

- Гарантированная доставка до Kafka Стриминга. Необходимо в том или ином виде реализовать паттерн Transactional Outbox
- Реализация механизмом для всех или части сценарием отказов. Например, переправка сообщений, механизм retry при недоступности части компонентов

## 5. СЛУЖЕБНЫЕ СООБЩЕНИЯ

- Необходим функционал по отправке служебных сообщений для корректной работы интеграционного потока. Например, heartbeat-сообщения для закрытия временных интервалов на стороне Потребителя.

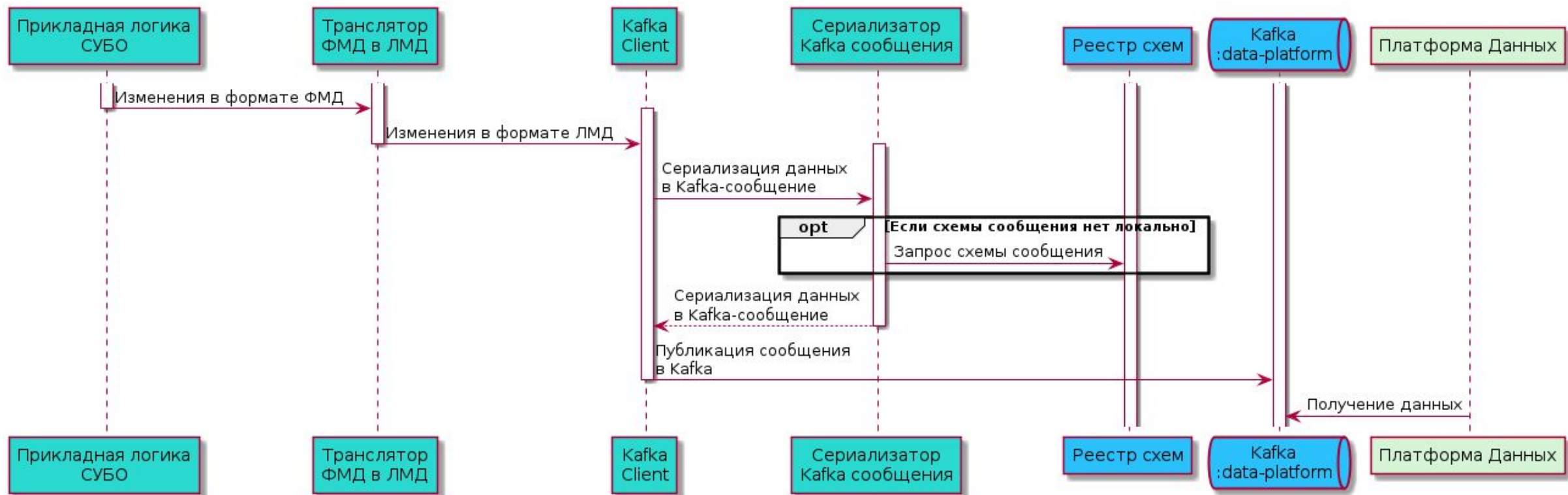
## 6. ВАЛИДАЦИЯ СХЕМЫ СООБЩЕНИЯ (ФЛК)

- Реализация функционал по проверке соответствия исходящего сообщения схеме сообщения (ФЛК)






# ПОСЛЕДОВАТЕЛЬНОСТЬ ВЗАИМОДЕЙСТВИЯ



 - Компонент СУБО

 - Компонент СС Стриминг

 - Компонент Платформы Данных



## Сообщение – набор данных характеризующий состояние сущности ЛМД

```
{
  "changeId": "1010000",
  "changeTimestamp": "2022-01-09T22:04:33.770Z",
  "changeOwner": "owner",
  "changeType": "CreatePartnerCreaditApplication",
  "changeKey": "id1235",
  "businessOperationId": "d16fddac-cb3a-4fca-84d9-74dee6e8d3d8",
  "payload": {
    "partnerCreditApplication": {
      "client": {
        "firstName": "John",
        "secondName": "II",
        "lastName": "Doe",
        "document": {
          "type": "Паспорт РФ",
          "num": "123456",
          "date": "01.01.2001"
        }
      },
      "contactInfo": {
        "type": "phone",
        "num": "+74957772424"
      }
    }
  }
}
```

### Состав сообщения:

**changeId** - уникальный идентификатор изменения.

**changeTimestamp** - время, когда было зафиксировано изменение.

**changeOwner** - владелец изменения. Может состоять из латинских букв, цифр и символа подчеркивания.

**changeType** - тип изменения. Например, CreatePartnerCreaditApplication (Создание Кредитной заявки от Партнера)

**changeKey** - ключ, влияющий на выбор партиции в топике. От этого зависит порядок обработки сообщений.

**businessOperationId** - идентификатор бизнес операции в виде строки.

**payload** - сущность ЛМД.



# ТЕРМИНЫ

## Схема сообщения – описание структуры сообщения.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://vtb.ru/schemas/PartnerCreditApplication",
  "title": "Кредитное заявление от Партнера",
  "type": "object",
  "properties": {
    "changeTimestamp": {
      "type": "string"
    },
    "changeOwner": {
      "type": "string"
    },
    "changeKey": {
      "type": "string"
    },
    "payload": {
      "type": "object",
      "properties": {
        "partnerCreditApplication": {
          "type": "object",
          "properties": {
            "client": {
              "type": "object",
              "properties": {
                "firstName": {
                  "type": "string"
                },
                "lastName": {
                  "type": "string"
                }
              }
            },
            "contactInfo": {
```

- Используется стандарт JSON Schema версия draft-07
- Ключевые слова для описания схемы:
  - "\$schema" - используется для задания версии драфта схемы.

`"$schema": http://json-schema.org/draft-07/schema#`

- "\$id" - Используется для указания уникального идентификатора документа, под которым схема регистрируется в Реестре Схем.
- Ограничения:
  - Схема должна содержать все ссылочные типы<sup>2</sup>

`"$id": "http://vtb.ru/schemas/PartnerCreditApplication"1`

<sup>1</sup> – Правила формирования идентификатора схем еще не утверждены

<sup>2</sup> – Ограничение действует на время Пилота, в последствие возможно будет снято



## Сбор/Контроль отправки/Хранение<sup>1</sup>:

- Отправка изменений от прикладного сервиса в Kafka CC Стриминг в Платформу данных
- Мониторинг процесса репликации
- Сохранение ряда ВИ в БД СС, для последующей отправки (задержка в репликации)<sup>2</sup>

## Визуализация:

- Интерфейс для просмотра очередей репликации
- Интерфейс для просмотра очередей ошибок репликации

<sup>1</sup> – На данном этапе допускается потеря данных

<sup>2</sup> – Сохранение изменений в БД на этапе Пилота не прорабатывается



# ДЕМОНСТРАЦИЯ 1. МОНИТОРИНГ ОЧЕРЕДЕЙ

## К Л С К Л

VTB



Главная страница

Стримминг в ПД

Мониторинг Кафка

Ошибки Кафка

### Мониторинг Кафка

Выберите Endpoint  
main

Поиск

Партиция	Топики	Start	End	Offset	Lag
0	streaming.smart_replication_change_request_selfcheck_default.RETRY.main	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default.ERROR.main	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default	1	1	1	0
	streaming.smart_replication_change_request_selfcheck_default.DLT.main	0	0	0	0
1	streaming.smart_replication_change_request_selfcheck_default.RETRY.main	1	101	19	82
	streaming.smart_replication_change_request_selfcheck_default.ERROR.main	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default.DLT.main	0	0	0	0
2	streaming.smart_replication_change_request_selfcheck_default.RETRY.main	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default.ERROR.main	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default	0	0	0	0
	streaming.smart_replication_change_request_selfcheck_default.DLT.main	0	0	0	0

Выйти



# ДЕМОНСТРАЦИЯ 2. ОШИБКИ РЕПЛИКАЦИИ



[Главная страница](#)

[Стримминг в ПД](#)

[Мониторинг Кафка](#)

**Ошибки Кафка**

[Выйти](#)

## Ошибки Кафка

selfcheck

Выберите Endpoint  
main

Идентификатор	Ключ	Время создания	Тип	Ид. операции	Версия	Содержимое	Ид. Транзакции	Текст ошибки	Сжатие
4146a01a-0bf8-4...	changeKey 18	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId18		changeVersionId18		-
66f30853-40be-...	changeKey 19	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId19		changeVersionId19		-
aeb55acf-06a7-4...	changeKey 20	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId20		changeVersionId20		-
cfbff40c-f4e4-45...	changeKey 21	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId21		changeVersionId21		-
39ce7c9f-0ae7-4...	changeKey 22	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId22		changeVersionId22		-
07809286-0d5c-...	changeKey 23	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId23		changeVersionId23		-
04452932-6494-...	changeKey 24	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId24		changeVersionId24		-
336cd87a-3a02-...	changeKey 25	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId25		changeVersionId25		-
d23f80c7-ed15-...	changeKey 26	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId26		changeVersionId26		-
b8b248ba-2871-...	changeKey 27	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId27		changeVersionId27		-
d601bc00-891c-...	changeKey 28	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId28		changeVersionId28		-
ddc898f8-a556-...	changeKey 29	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId29		changeVersionId29		-
b087a0ae-3be6-...	changeKey 30	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId30		changeVersionId30		-
3911ed7f-f716-4...	changeKey 31	25.05.2021, 17:49	default	BusinessOperatio...	changeVersionId31		changeVersionId31		-



## ИНТЕГРАЦИЯ



### Обязательные шаги:

1. Заведение RDS на интеграцию
2. Подключение библиотеки smart-replication-client
3. Зарегистрировать СУБО в СС
4. Согласовать ЛМД с архитектурой данных и ПД
5. Разработать схему сообщения на основе ЛМД
6. Опубликовать схему сообщения в Реестре схем
7. Выполнять первоначальную инициализацию СУБО в ПД<sup>1</sup>



OPENSIFT



### Функции библиотеки:

- Валидация сообщений перед отправкой в Kafka
- Компрессия/декомпрессия сообщений<sup>2</sup>
- Использование бинарного формата сообщения.<sup>2</sup>



PostgreSQL



<sup>1</sup> Процесс первоначальной инициализации СУБО в ПД на уточнении

<sup>2</sup> Функционал требует дополнительного уточнения



## ПЛАНЫ НА БУДУЩЕЕ

- Реализация паттерна transactional outbox для обеспечения гарантированной доставки
- Разработка эталонного механизма служебных сообщений
- Переход на использование бинарного формата сообщений





## Полезные ССЫЛКИ

Код РИС **1609**

Код АПД **63.04.07**

Wiki <https://wiki.corp.dev.vtb/x/W3QAB>

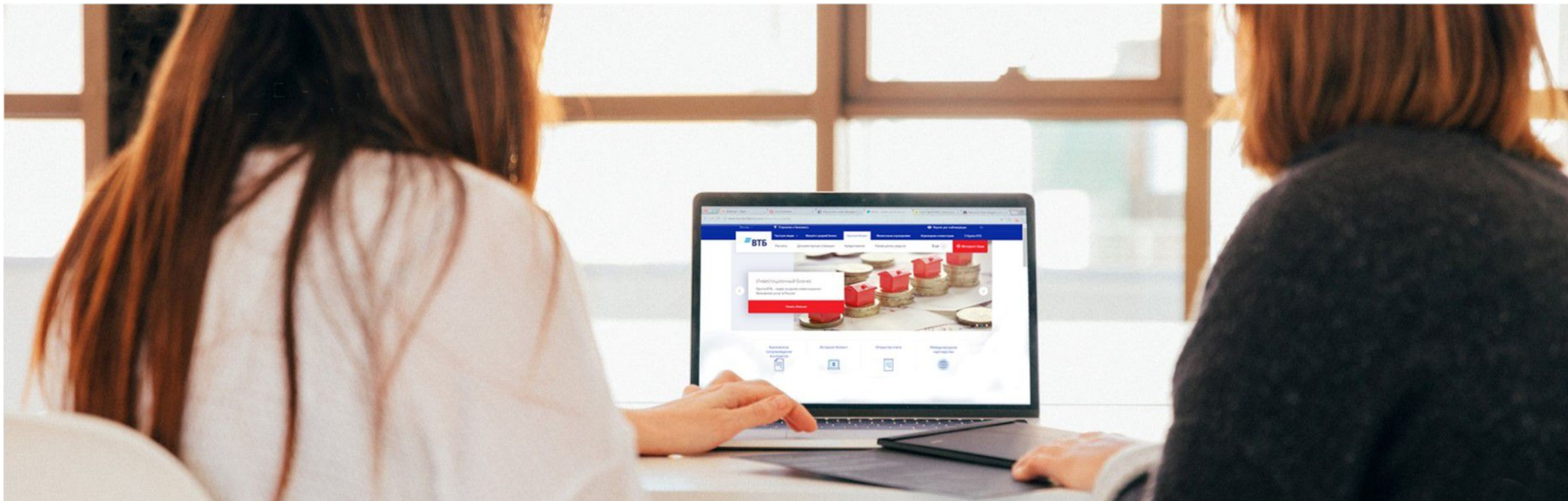
Jira <https://task.corp.dev.vtb/projects/OMNI5/issues>

Артефакты <https://wiki.corp.dev.vtb/x/jdNNEQ>

Руководство по подключению  
<https://wiki.corp.dev.vtb/x/EU2IFw>



Чат <https://t.me/joinchat/l4koZ8emnJVjNDc6>



**Ваши вопросы?**

Спасибо за внимание!

