




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы



Защита проекта

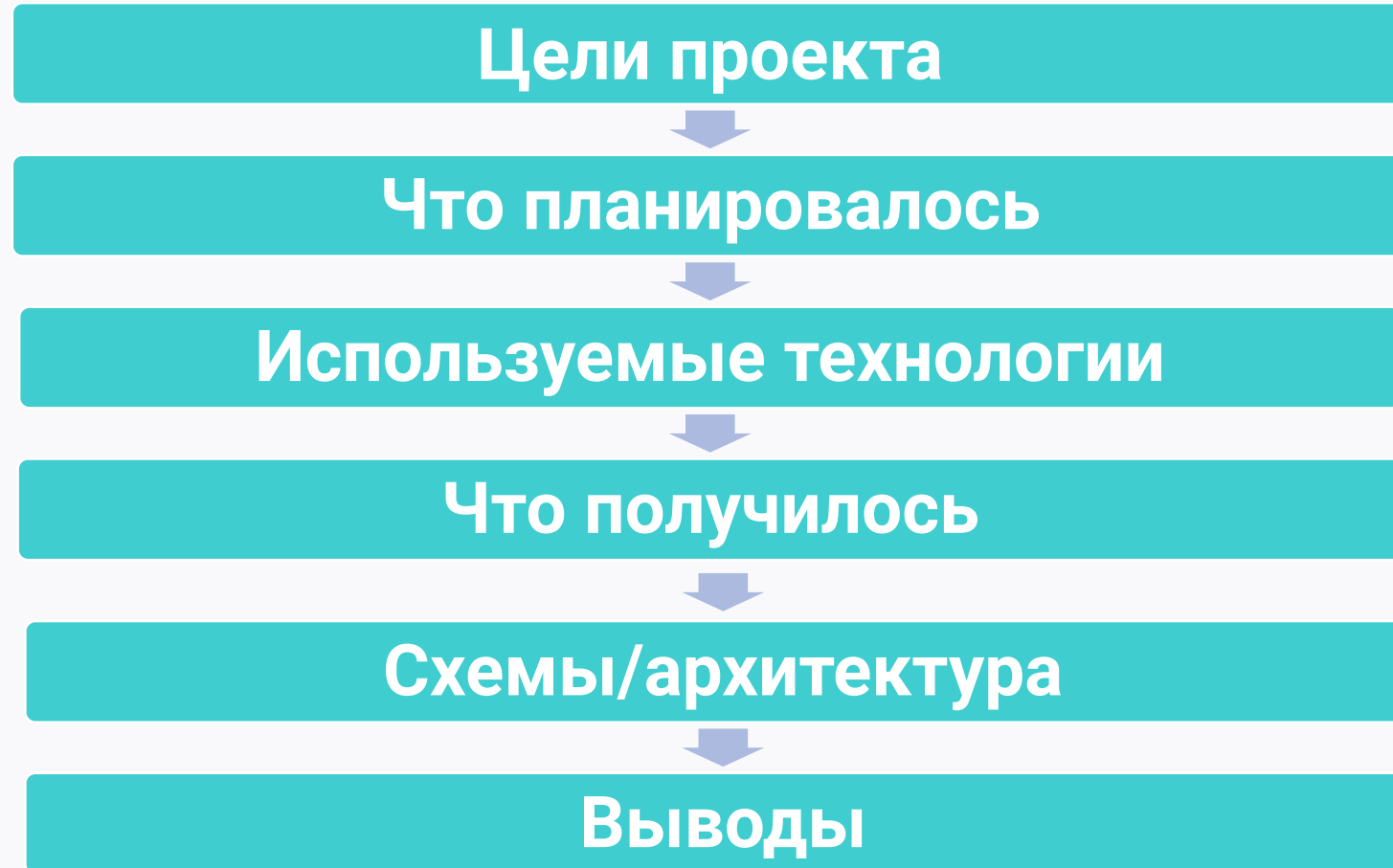
Тема: потоковый сервис видео-связи с постобработкой



Поляков Андрей

Backend Software Engineer at Speech Technology Center
(@speechpro)

План защиты



Цели проекта

Какие цели вы поставили и какие задачи решили своим проектом

1

Создать отказоустойчивую систему потоковой передачи данных ("отложенного" реального времени) с постобработкой видео

2

Проверить возможности обработки потоковой видео и аудио информации с помощью средств (библиотек) языка Java

3

Применить полученные на курсе знания для создания highload-приложения

Что планировалось

- 1 Создать клиентское приложение для совершения групповых видео/аудио звонков
- 2 Изучить работу с websocket, обработку видео-данных с opencv, стримами kafka
- 3 Настроить вычислительный кластер с service discovery и централизованным хранением конфигураций
- 4 Объединить Apache Ignite Service Grid+Compute Grid с обычными spring сервисами и клиентским приложением
- 5

Что было в начале, что знали до курса, сколько времени заняло выполнение проекта



Используемые технологии

- 1 **Основа сервисов:** Java + Spring Boot.
Фронтенд: JavaFX + CSS
- 2 Service Discovery: Consul
и Ignite TCP Discovery
- 3 **Буферизация данных:** in-memory key-value DB
(RocksDB) и реплицируемый распределенный кэш
Ignite (+ H2DB) с бекапами
- 4 **База данных:** PostgreSQL (хранение пользователей
и сессий звонков)
- 5 **Сервер конфигурации:** Spring Cloud Config Server
Передача данных: REST и MQ Kafka

Какие технологии
использовались и какое у
вас мнение о новых
технологиях



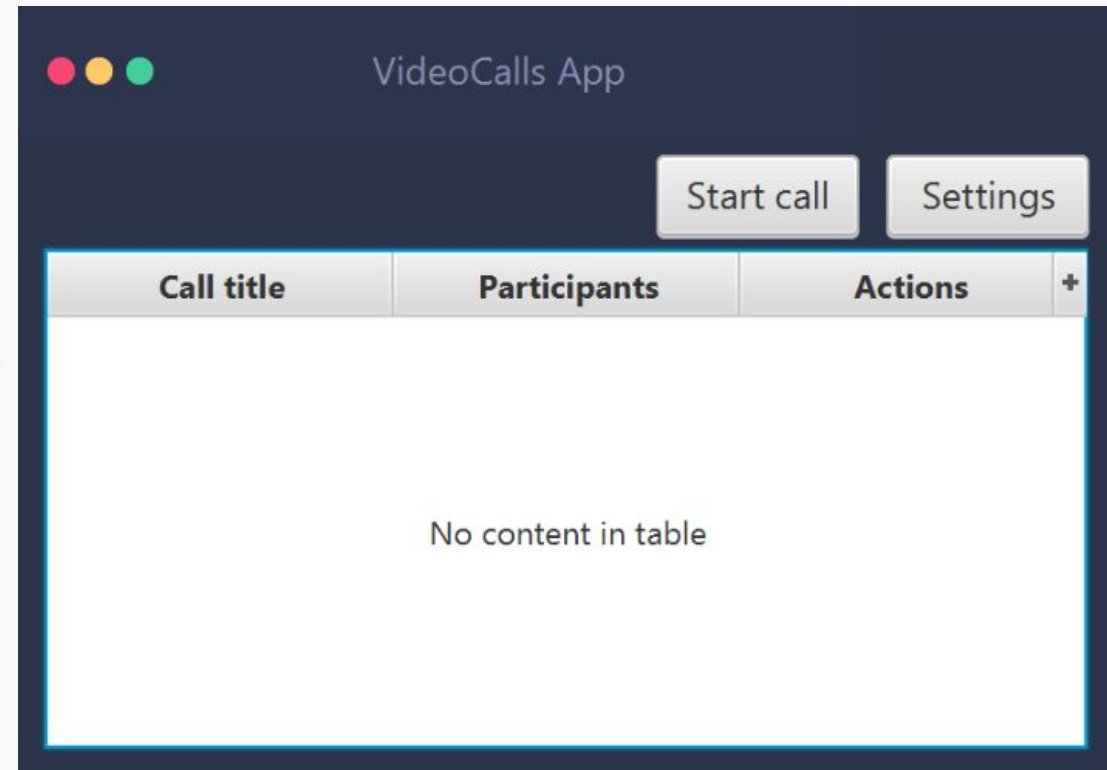
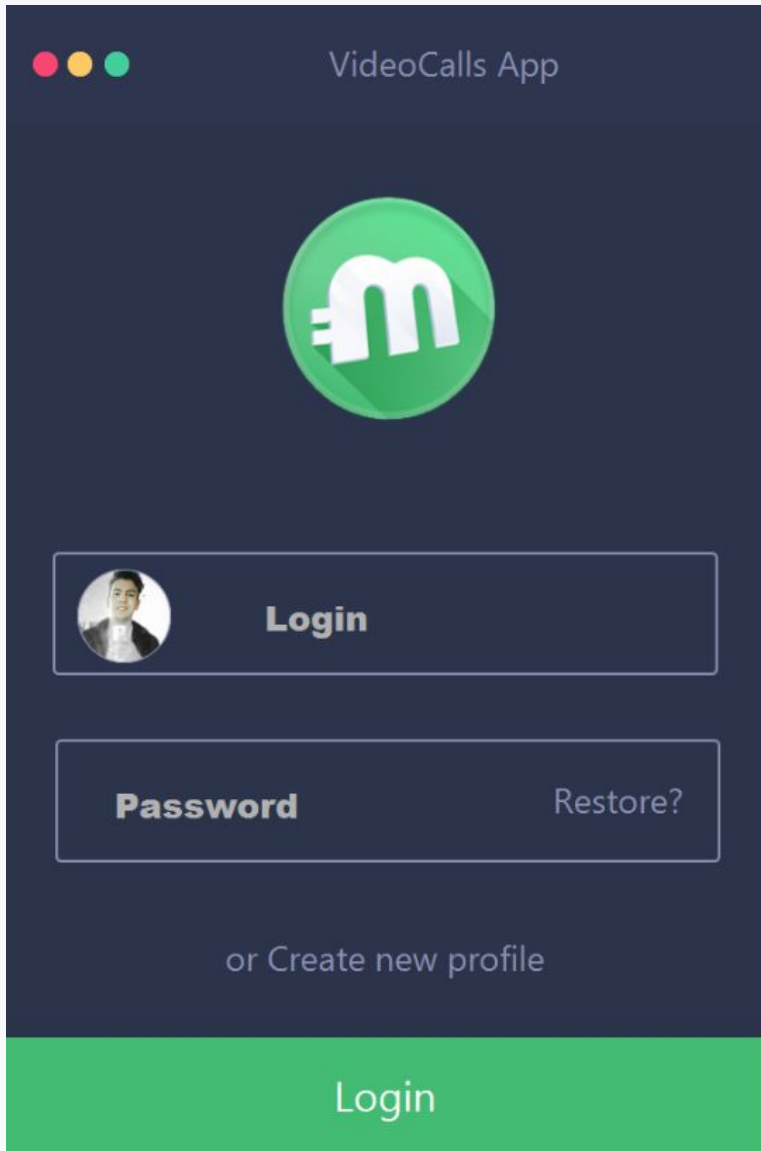
Что получилось

Репозиторий проекта:

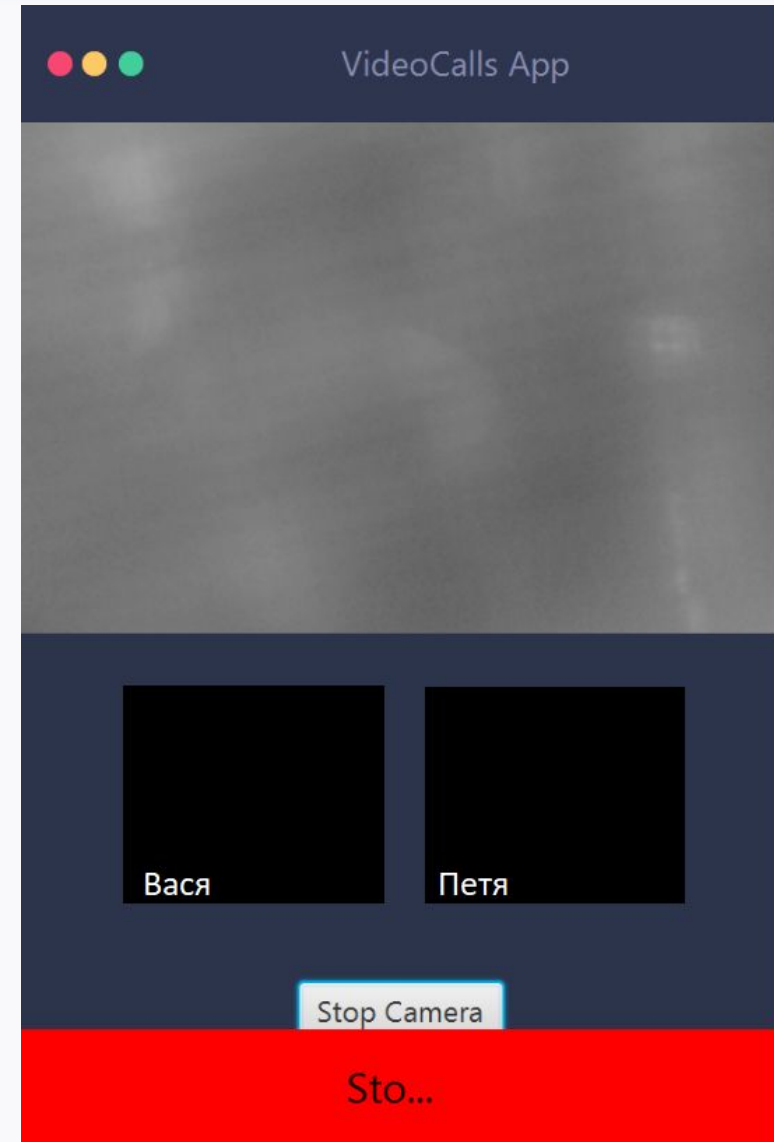
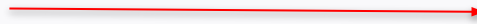
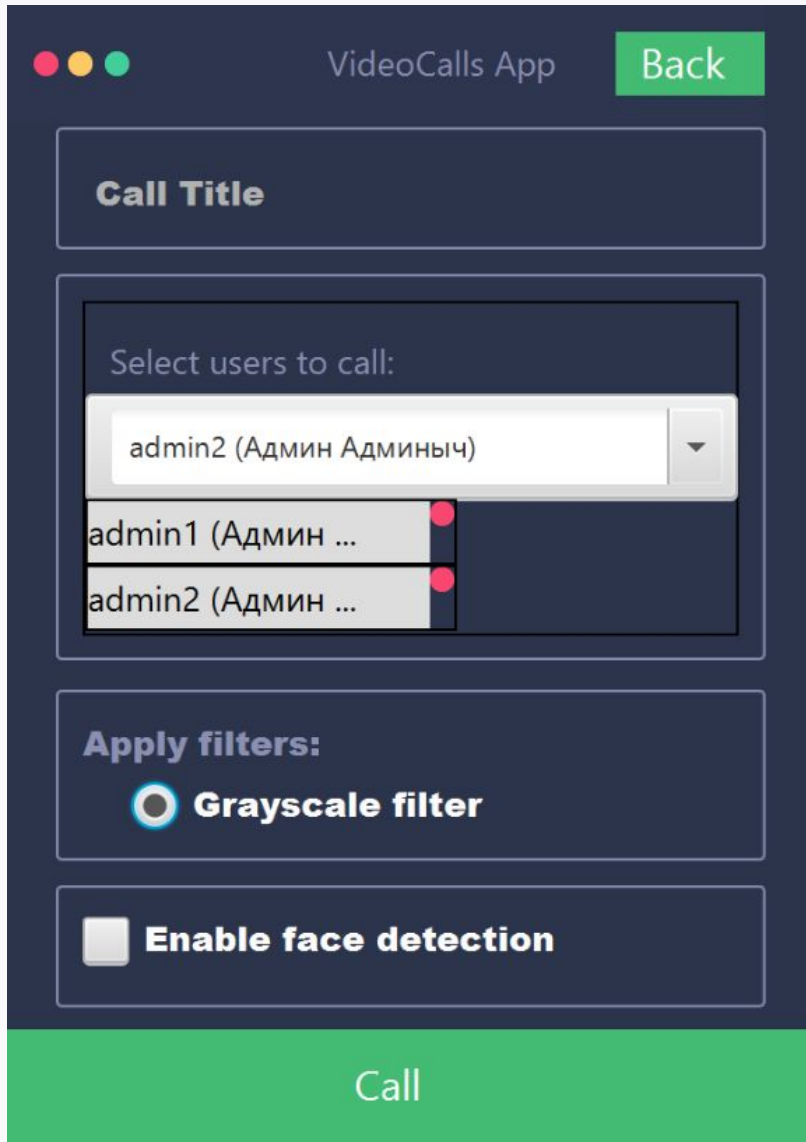
<https://github.com/AVPGenium/video-service>

Что получилось

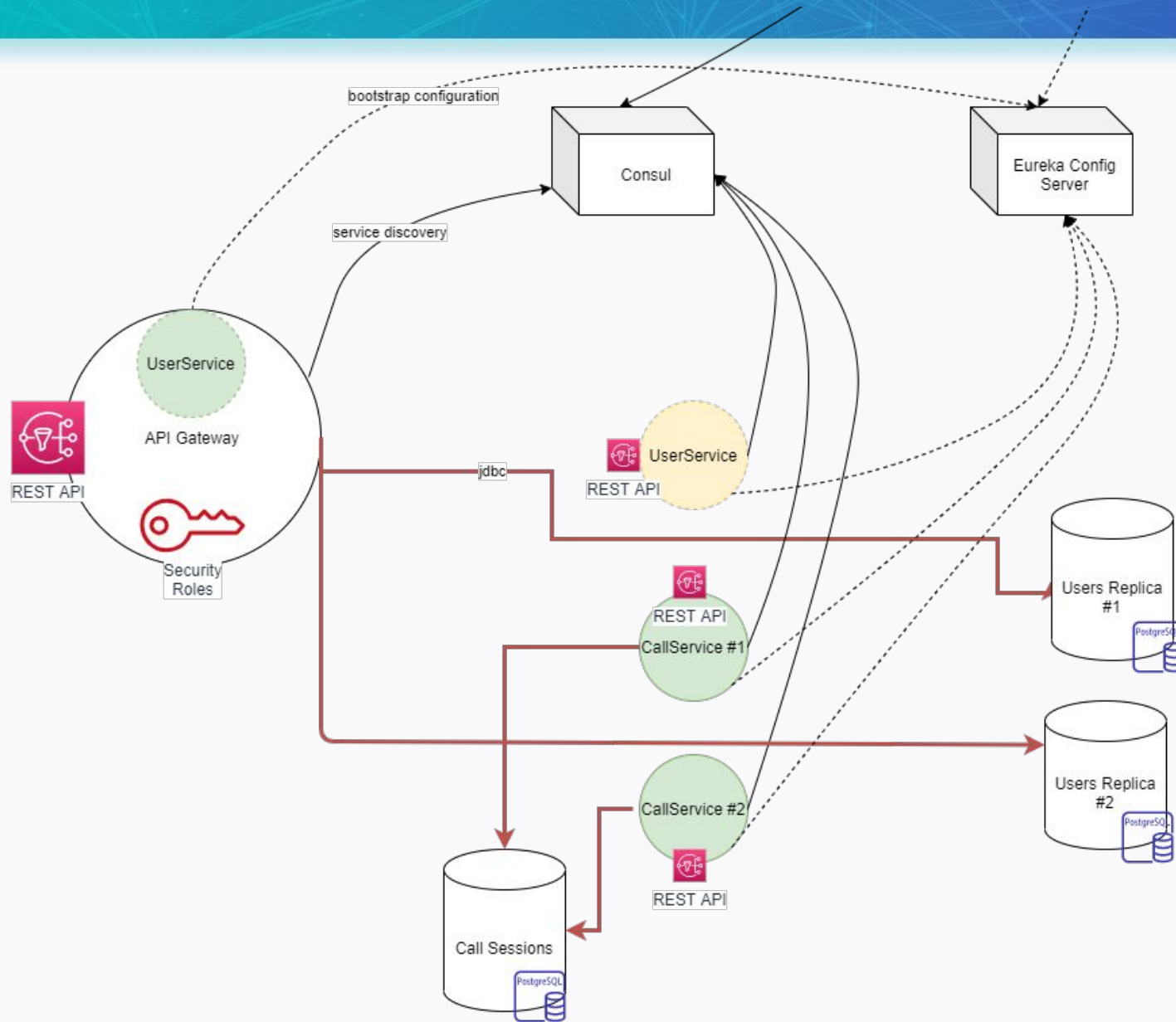
Клиентское приложение:



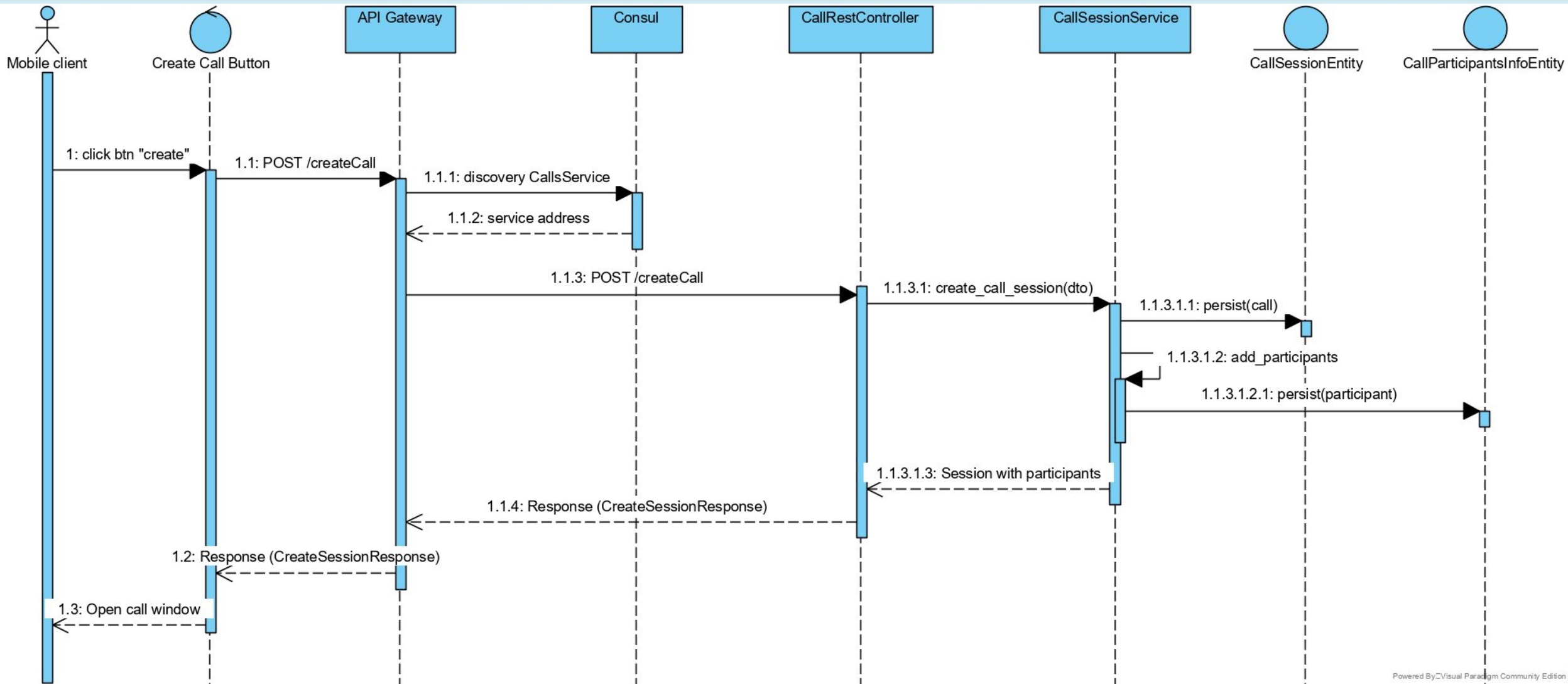
Что получилось



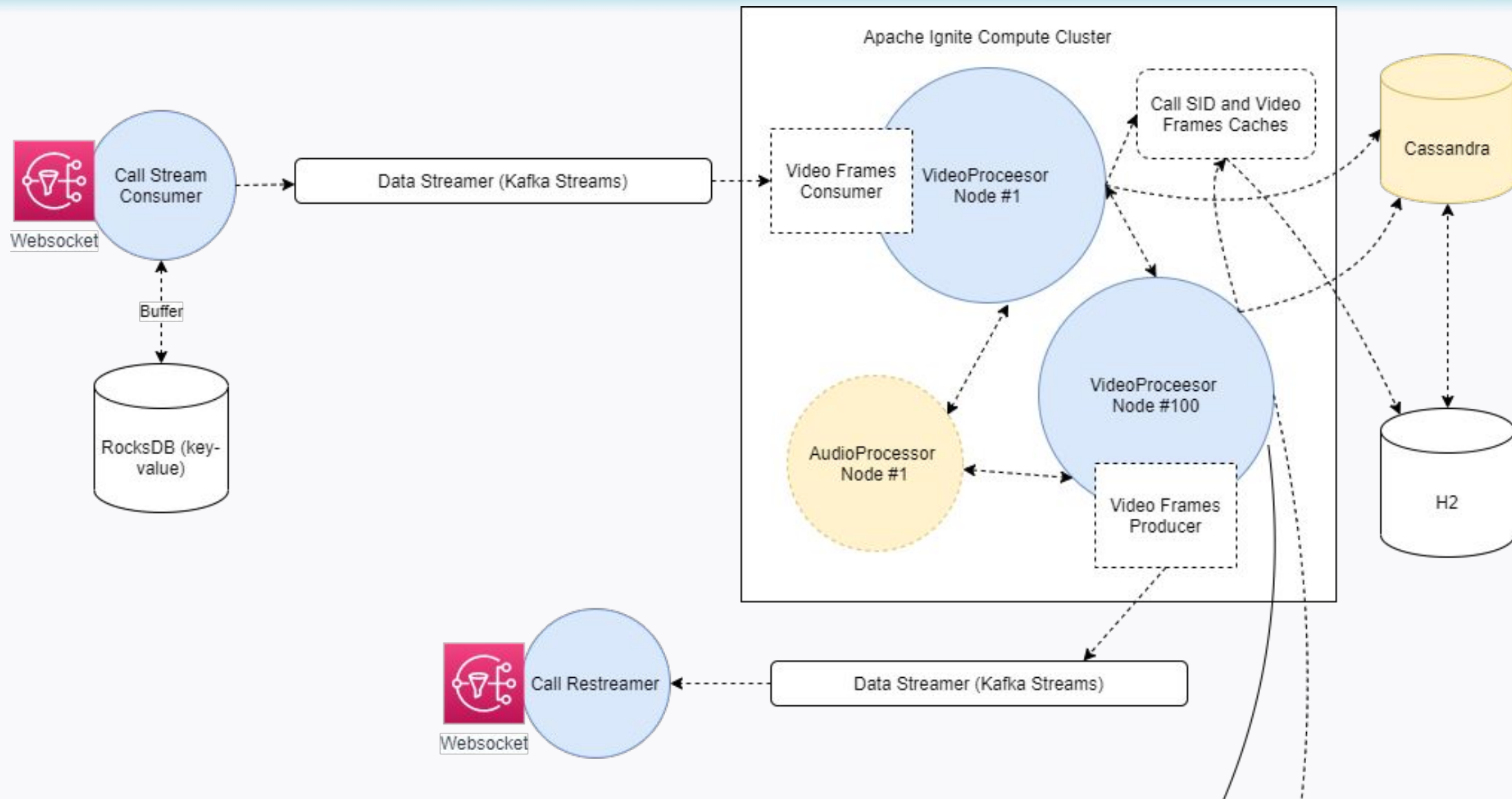
Схемы (архитектура, БД)



Схемы (архитектура, БД)



Пайплайн обработки стрима звонка



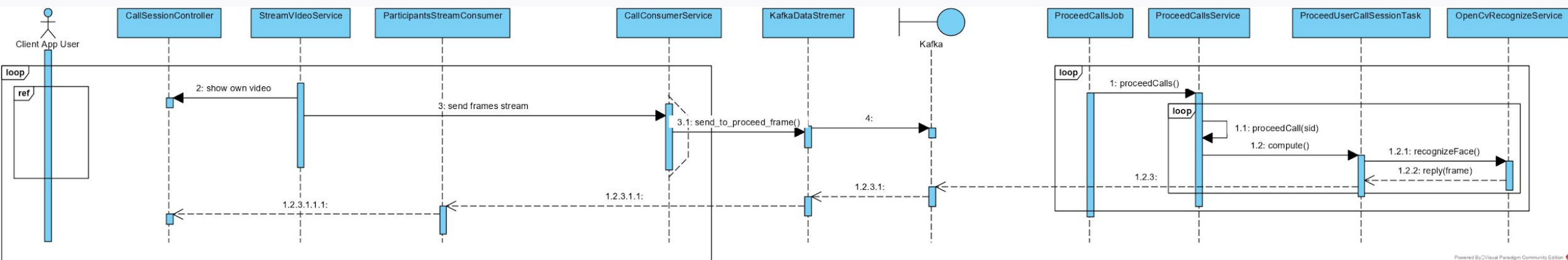
WebRTC vs WebSocket + DataStreamer

WebRTC - стандарт и поддерживается многими браузера. Нет необходимости передавать каждый фрейм по отдельности.

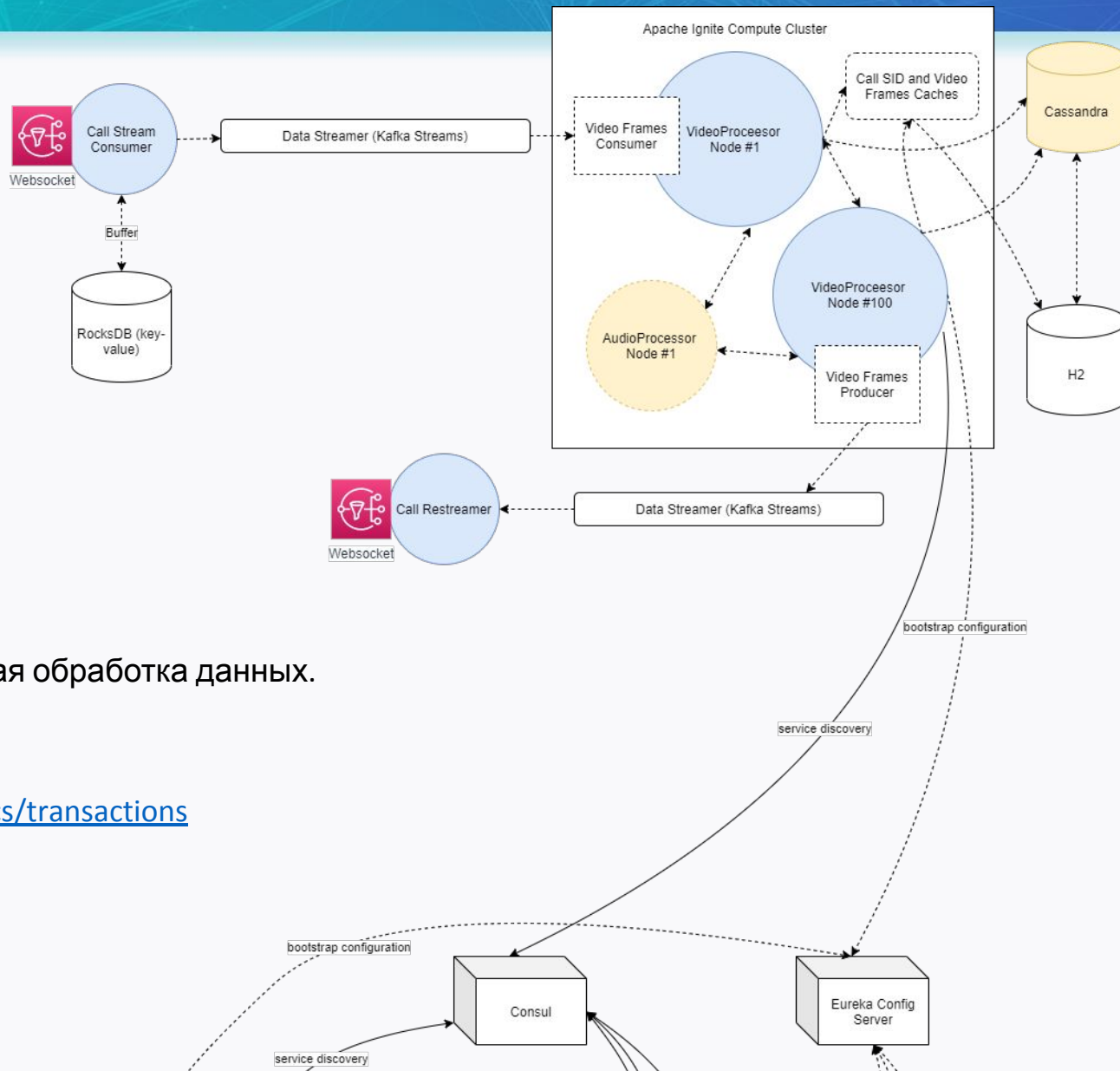
НО

нельзя преобразовать в формат понятный для openCV. Поэтому передаются сериализованные объекты openCV и обрабатываются в пайплайне.

Пайплайн обработки стрима звонка



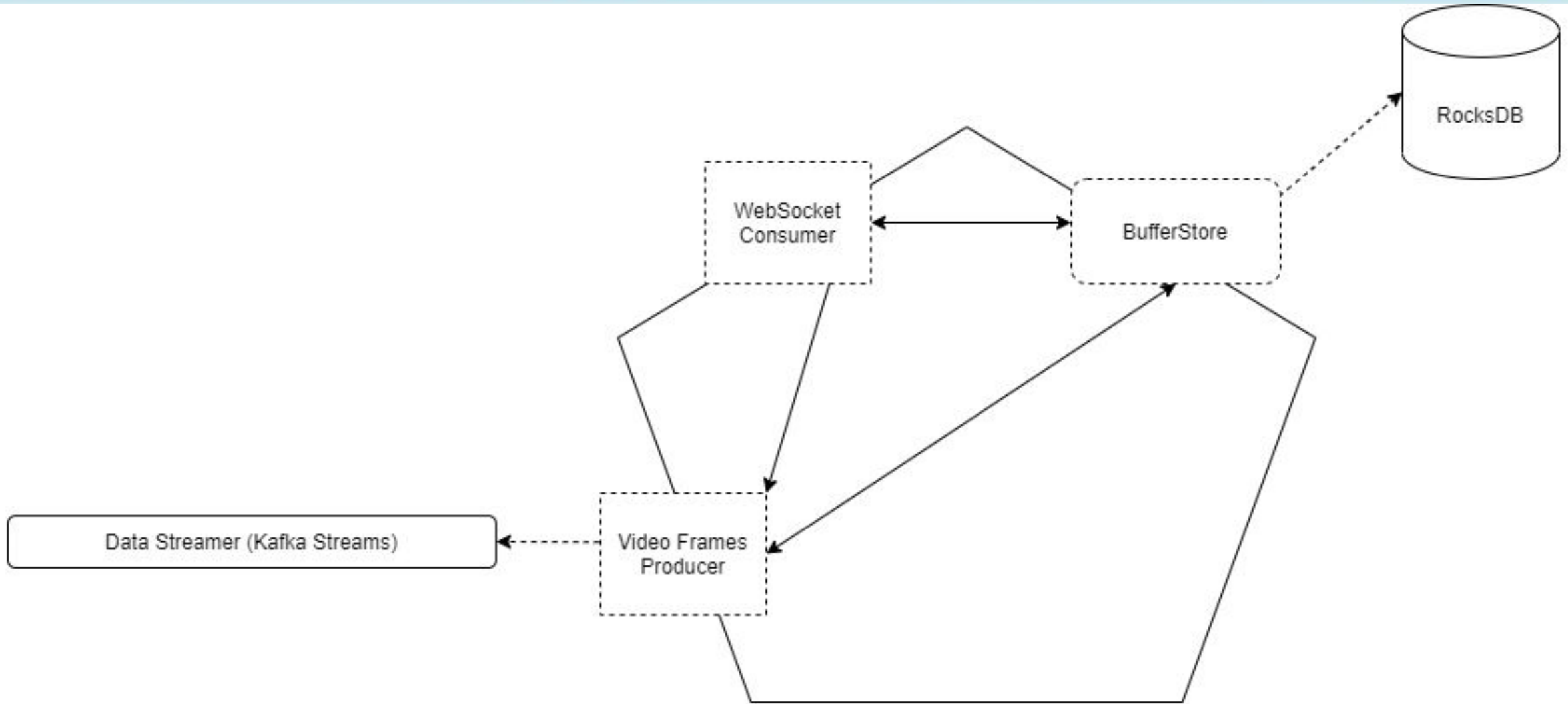
Схемы (архитектура, БД)



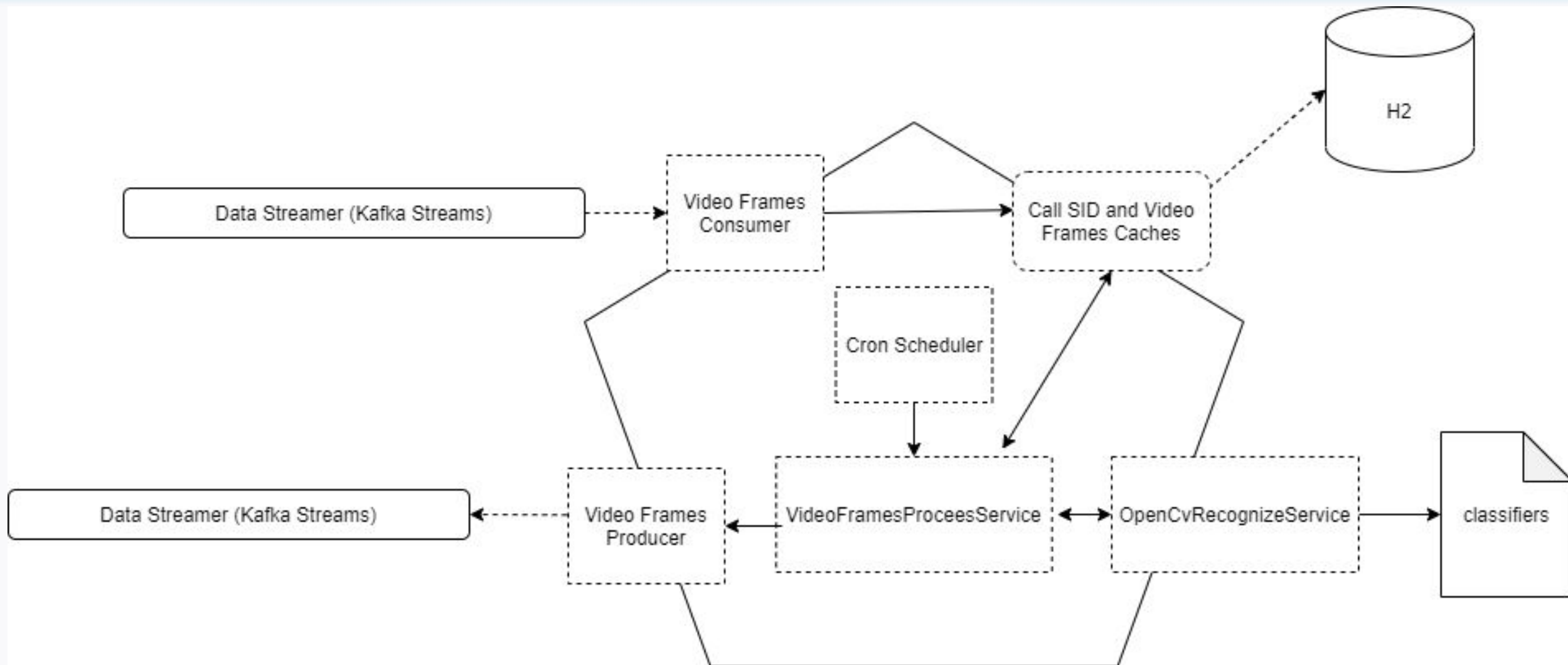
* Эндрю Дж. Пселтис: Потокковая обработка данных.
Конвейер реального времени

* Apache Ignite: ACID for caches -
<https://apacheignite.readme.io/docs/transactions>

CallConsumerService

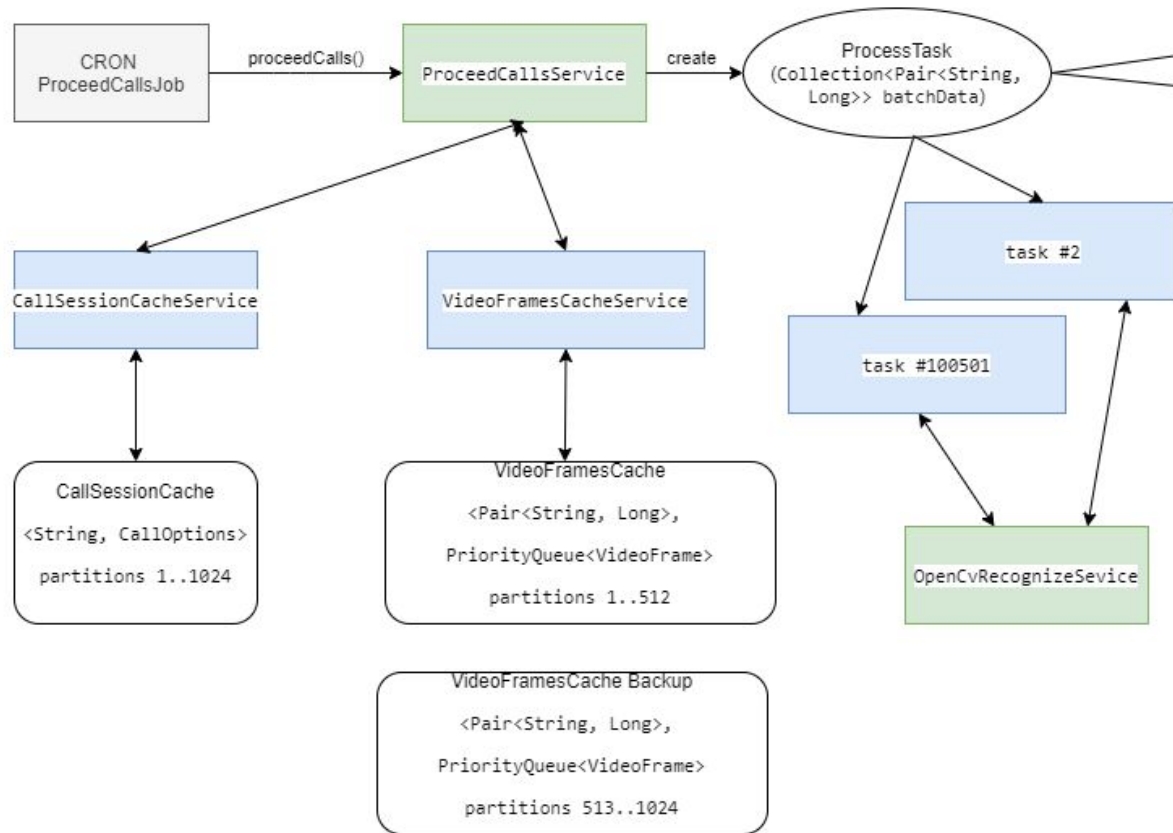


VideoProcessorService

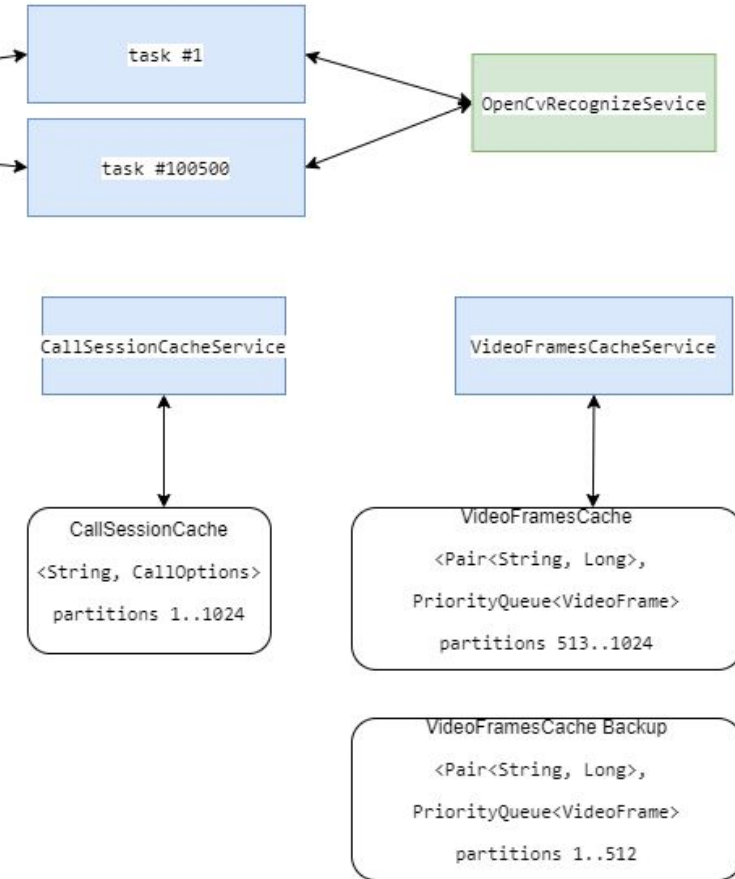


VideoProcessorService

Node 1



Node 2



Вывод и планы по развитию

- 1 Планируется добавить сервисы по передаче/обработке аудио-данных
- 2 Использовать результаты обработки аудио в обработке видео
- 3 Перевести клиентское приложение на JS
- 4 Добавить docker + kubernetes
- 5 доделать MVP: взаимодействие по веб-сокеты, взаимодействие между сервисами, ...

Запланируйте пару минут на рефлекссию в конце защиты проекта и расскажите о планах по развитию





Спасибо за внимание!

фото

Поляков Андрей Владимирович

Backend Software Engineer at Speech Technology Center
(@speechpro)

<https://github.com/AVPGenium>
polyakov-a@speechpro.com