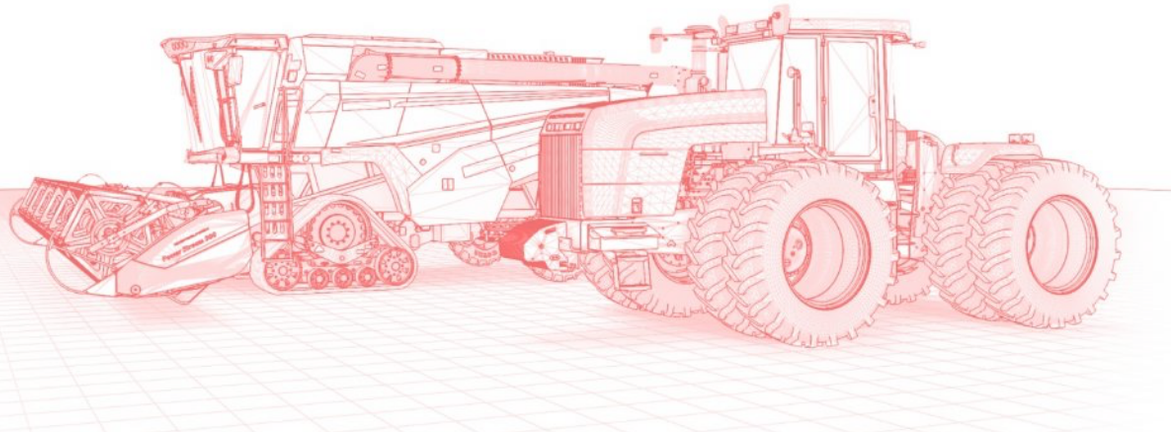
РОСТСЕЛЬМАШ

Internet Of Harvesting с помощью Apache Ignite

Онлайн мониторинг потока разнообразных данных для «умного земледелия»



РОСТСЕЛЬМАШ

Куценко Виталий

- Lead Java-backend в ЦПТ Агроцифра

  > 20 лет в разработке

  > 13 лет в мире Java

- За это время строил системы различных размеров и профиля: ERP, CRM, highload, скоринг и финансы.

- Авторский спецкурс в РАНХиГС

РОСТСЕЛЬМАШ

О чем расскажу

РОСТСЕЛЬМАШ

О чем расскажу

- Зачем нужны информационные технологии сельскому хозяйству

РОСТСЕЛЬМАШ

О чем расскажу

- Зачем нужны информационные технологии сельскому хозяйству
- Идеальная и реальная лямбда-архитектура

РОСТСЕЛЬМАШ

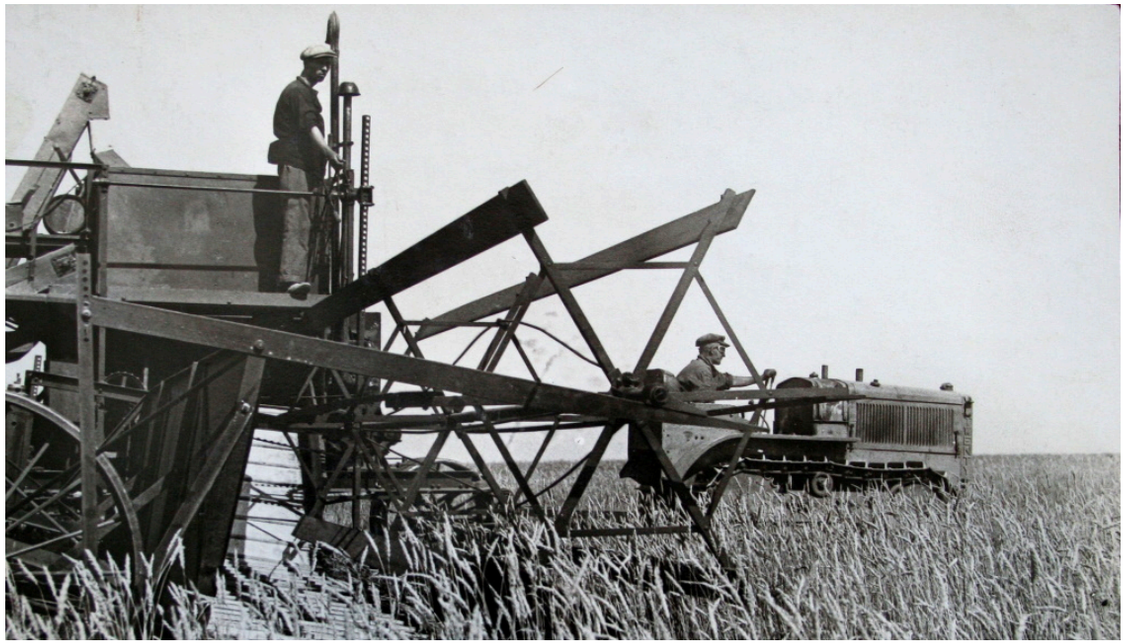
О чем расскажу

- Зачем нужны информационные технологии сельскому хозяйству
- Идеальная и реальная лямбда-архитектура
- Применение Apache Ignite для горячего хранилища

Как выглядят комбайны?

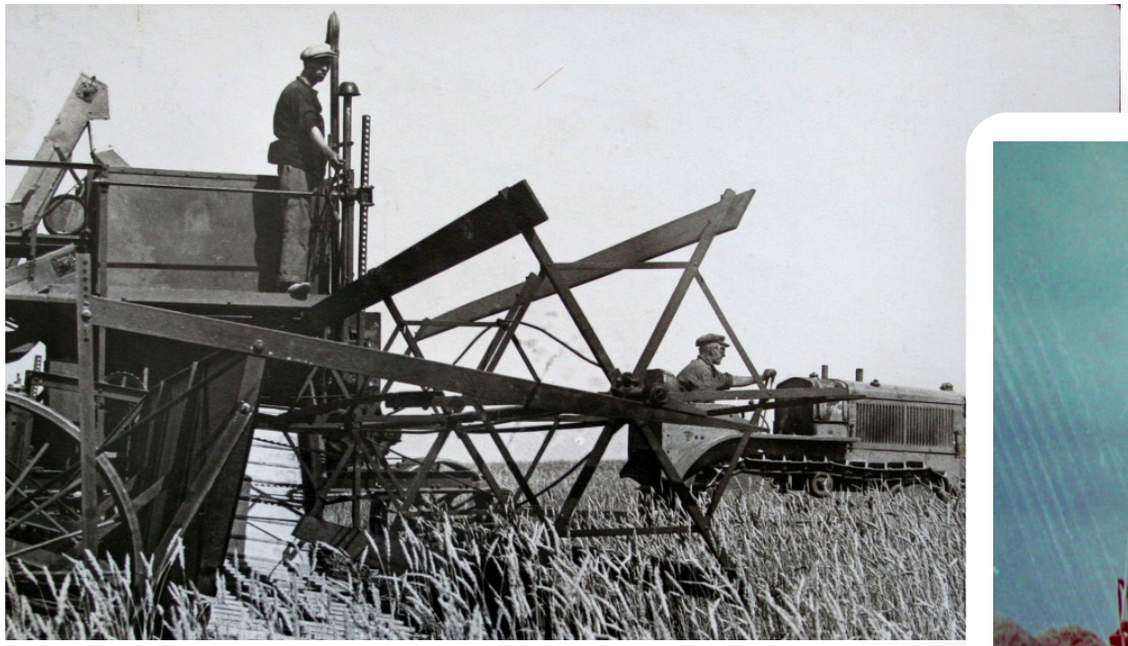
РОСТСЕЛЬМАШ

Как выглядят комбайны?



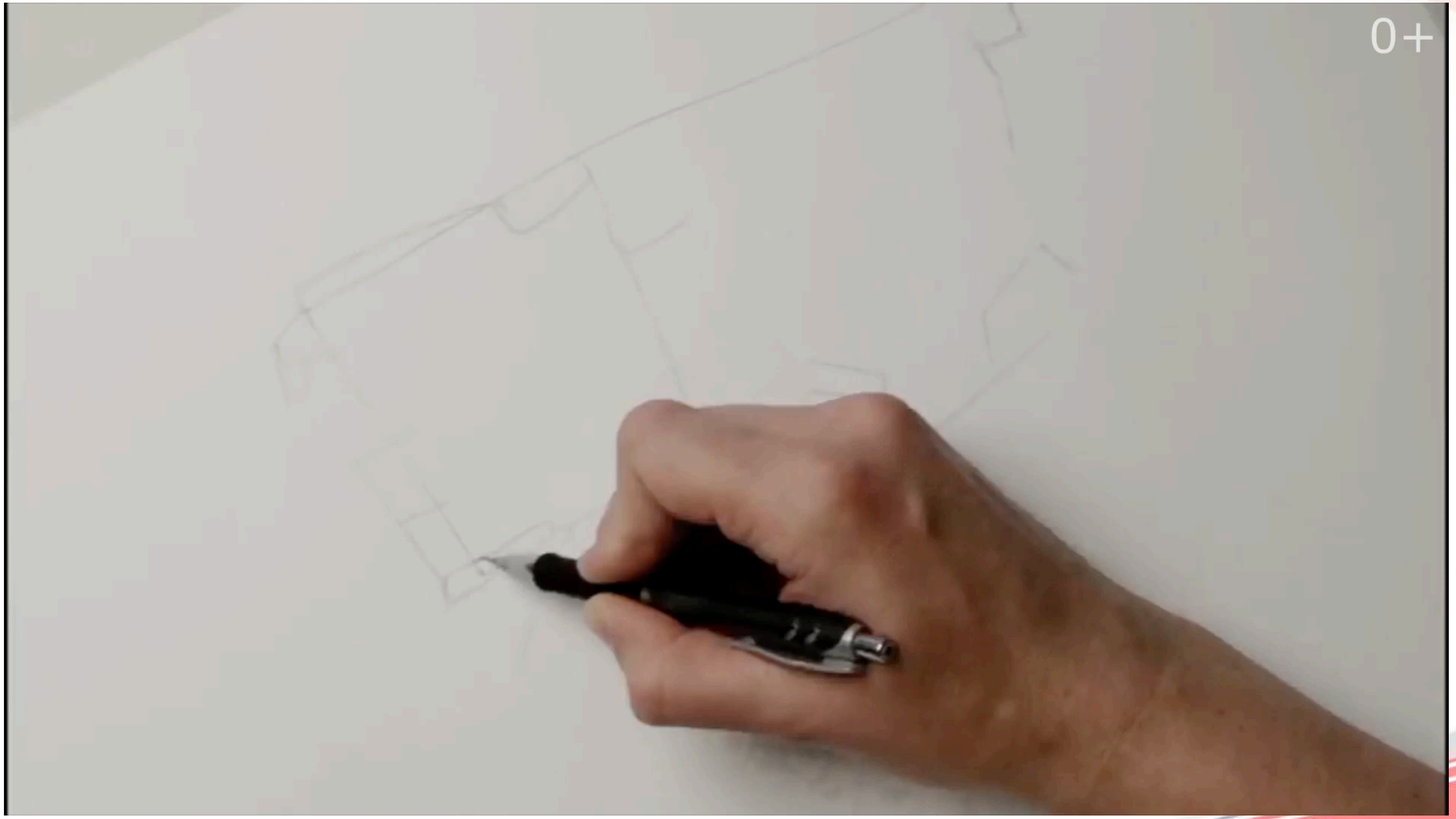
РОСТСЕЛЬМАШ

Как выглядят комбайны?



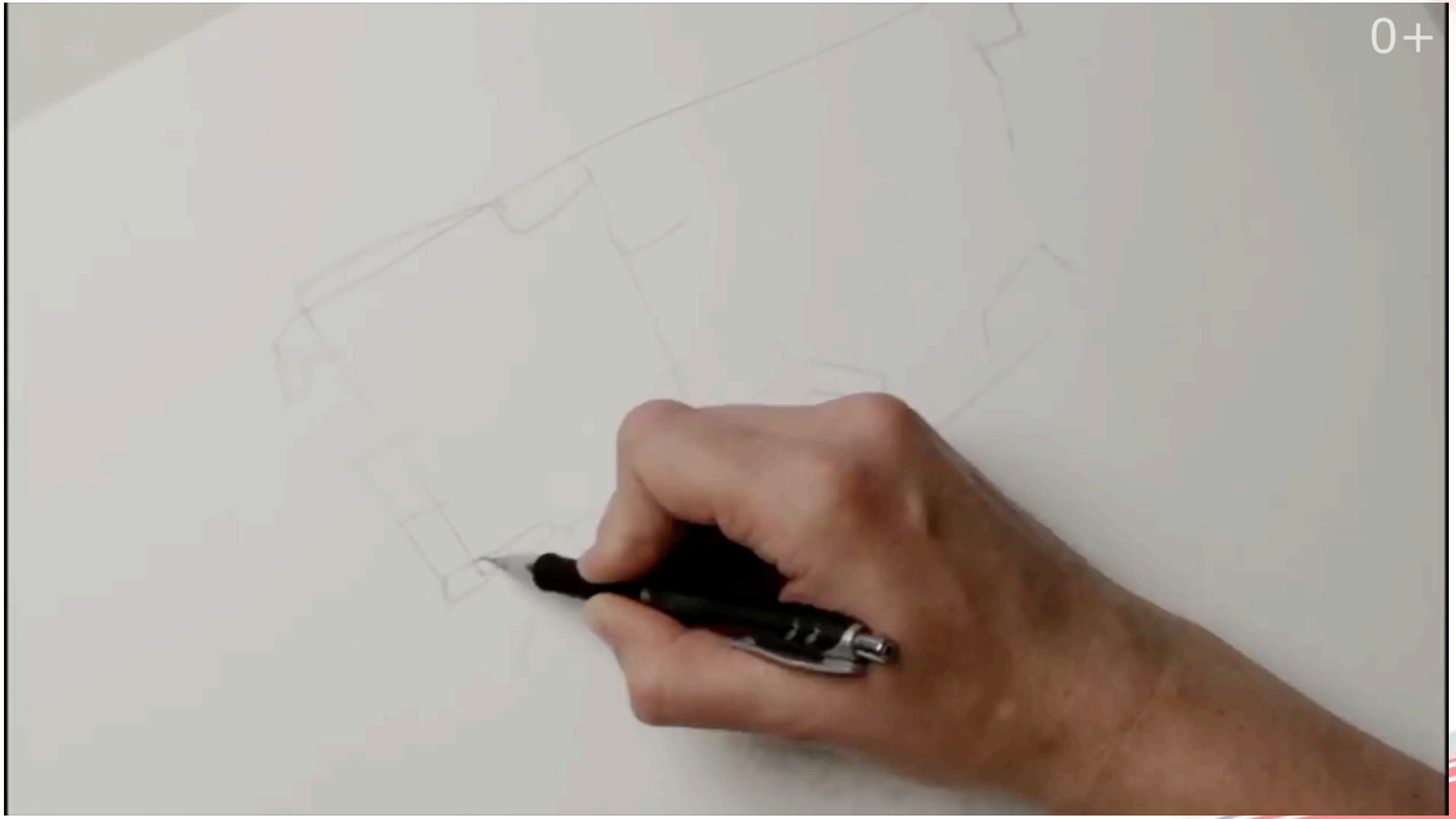
Современные комбайны

РОСТСЕЛЬМАШ



Современные комбайны

РОСТСЕЛЬМАШ



Современные комбайны

РОСТСЕЛЬМАШ

Современные комбайны

РОСТСЕЛЬМАШ

Современные комбайны

РОСТСЕЛЬМАШ

Современные комбайны

РОСТСЕЛЬМАШ

РОСТСЕЛЬМАШ

Agritechnika Awards

- RSM Nigth Vision (2019 - серебро, 2020 - золото)
- RSM Пилот 2.0 (2020)
- RSM Контроль глубины (2020)
- RSM Умная метка 1.0 (2020)
- RSM Ок АйДи (2021)

AGRO CODE HUB

РОСТСЕЛЬМАШ

AGRO CODE HUB

РОСТСЕЛЬМАШ

- Перспективные земли для виноградников

AGRO CODE HUB

РОСТСЕЛЬМАШ

- Перспективные земли для виноградников
- Решения для сегментации, трекинга и измерения активности поросят

AGROCODE HUB

РОСТСЕЛЬМАШ

- Перспективные земли для виноградников
- Решения для сегментации, трекинга и измерения активности поросят
- Разработка приложения для повышения урожайности космической клубники

Что хотят пользователи?

РОСТСЕЛЬМАШ

Что хотят пользователи?

РОСТСЕЛЬМАШ

- Видеть подробно информацию о поле и машине за прошлые периоды в различных проекциях

Что хотят пользователи?

РОСТСЕЛЬМАШ

- Видеть подробно информацию о поле и машине за прошлые периоды в различных проекциях
- Видеть происходящее с машинами сейчас и получать актуальную информацию о важных параметрах

Что хотят пользователи?

РОСТСЕЛЬМАШ

- Видеть подробно информацию о поле и машине за прошлые периоды в различных проекциях
- Видеть происходящее с машинами сейчас и получать актуальную информацию о важных параметрах
- Оперативно получать информацию о действиях требующих реакции: заполнение бункера, сливы топлива, выгрузка в чужую машину, критические состояния оборудования

Что хотят пользователи?

РОСТСЕЛЬМАШ

- Видеть подробно информацию о поле и машине за прошлые периоды в различных проекциях
- Видеть происходящее с машинами сейчас и получать актуальную информацию о важных параметрах
- Оперативно получать информацию о действиях требующих реакции: заполнение бункера, сливы топлива, выгрузка в чужую машину, критические состояния оборудования
- Всё это надежно сохранять и красиво отображать

Формализация задачи

Формализация задачи

- **Сбор и сохранение информации**
  - ✓ Много данных (RPS>1000)
  - ✓ Большая вариативность данных
  - ✓ Неравномерность поступления

Формализация задачи

- **Сбор и сохранение информации**
  - ✓ Много данных (RPS>1000)
  - ✓ Большая вариативность данных
  - ✓ Неравномерность поступления
- **Разнообразие аналитики**
  - ✓ Различные срезы по историческим данным
  - ✓ Необходимость оперативного наблюдения за ограниченным кругом показателей

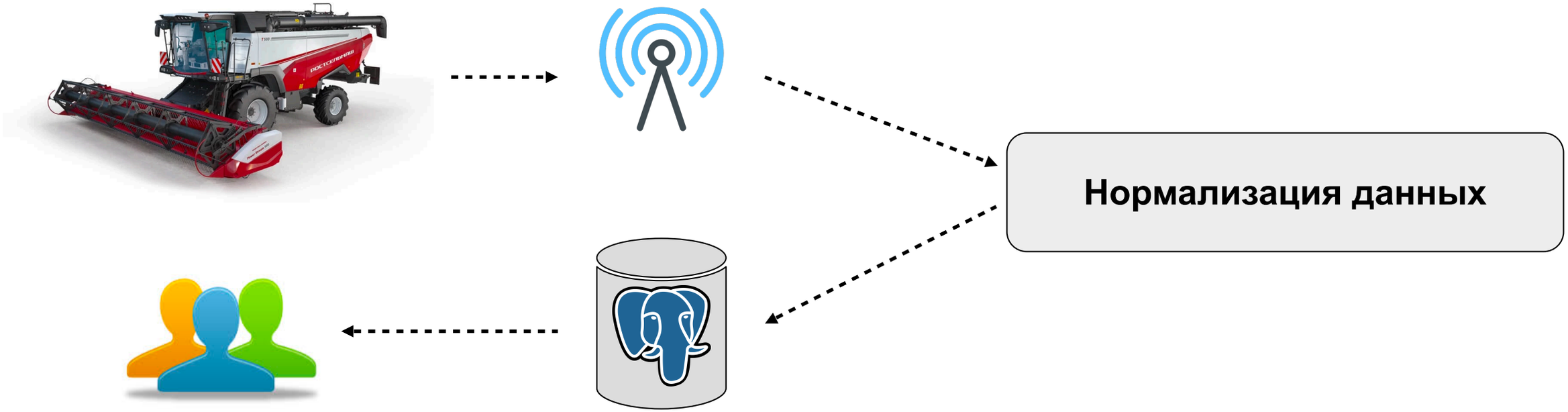
Формализация задачи

- **Желания от бизнеса**
  - ✓ Быстро внедрять новые кейсы
  - ✓ Не тратить много денег на сопровождение
  - ✓ И железа поменьше, поменьше.

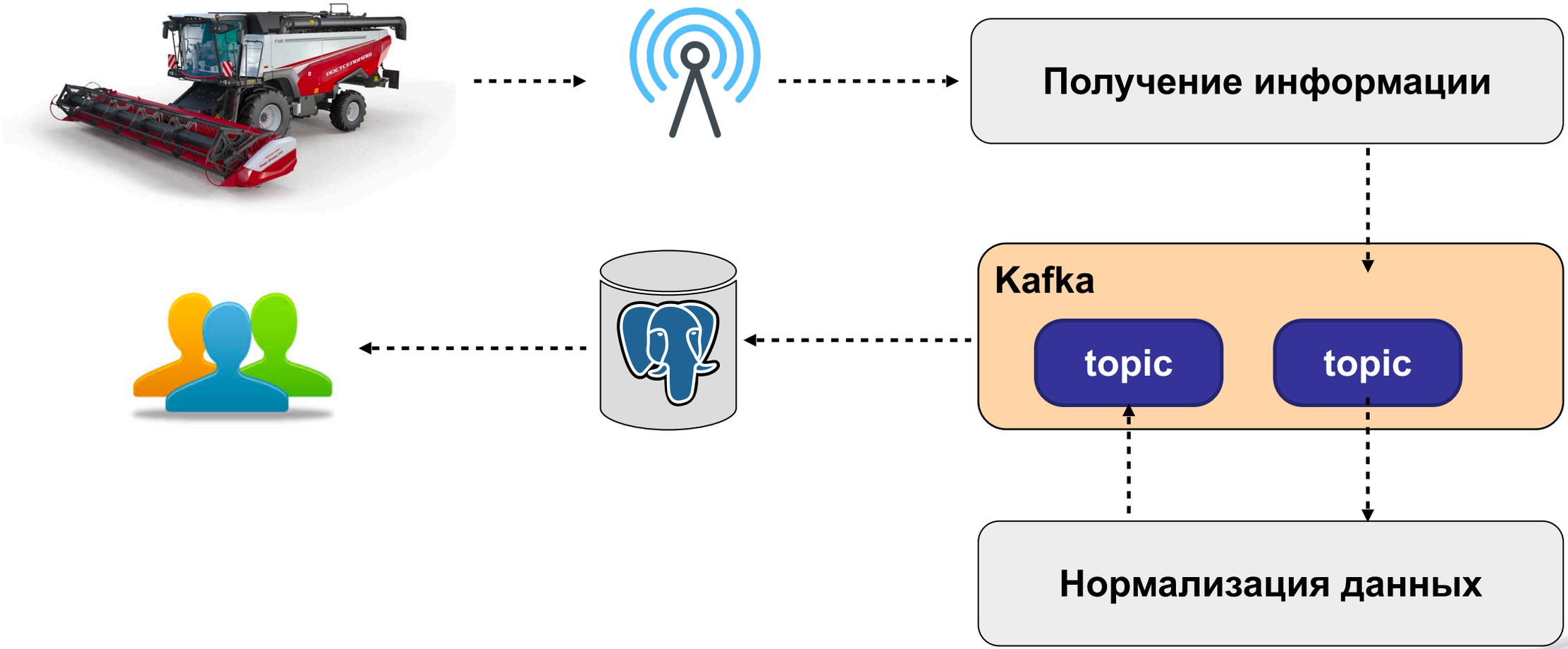
PostgreSQL

- Известное стабильное решение, вся мощь SQL
- Развитый инструментарий для репликации/шардирования
- Развитая поддержка JSON

Архитектура решения



Event Driven Development



PostgreSQL. Проблемы

PostgreSQL. Проблемы

- Реляционная модель

PostgreSQL. Проблемы

- Реляционная модель
- Consistency превалирует над Partition Tolerance

PostgreSQL. Проблемы

- Реляционная модель
- Consistency превалирует над Partition Tolerance
- Шардирование требует усилий и компетенций

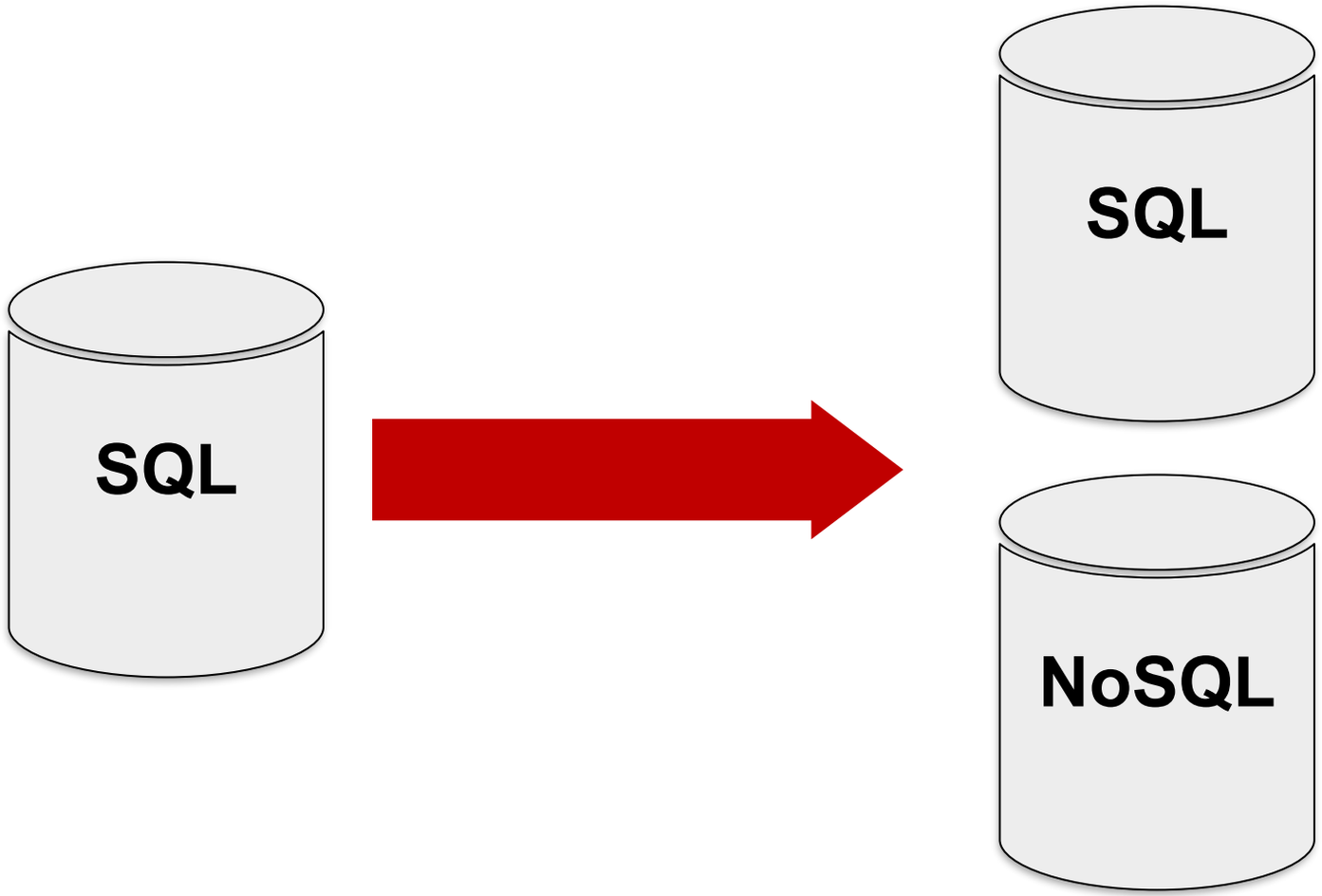
PostgreSQL. Проблемы

- Реляционная модель
- Consistency превалирует над Partition Tolerance
- Шардирование требует усилий и компетенций
- Репликация не насколько быстрая как обещается

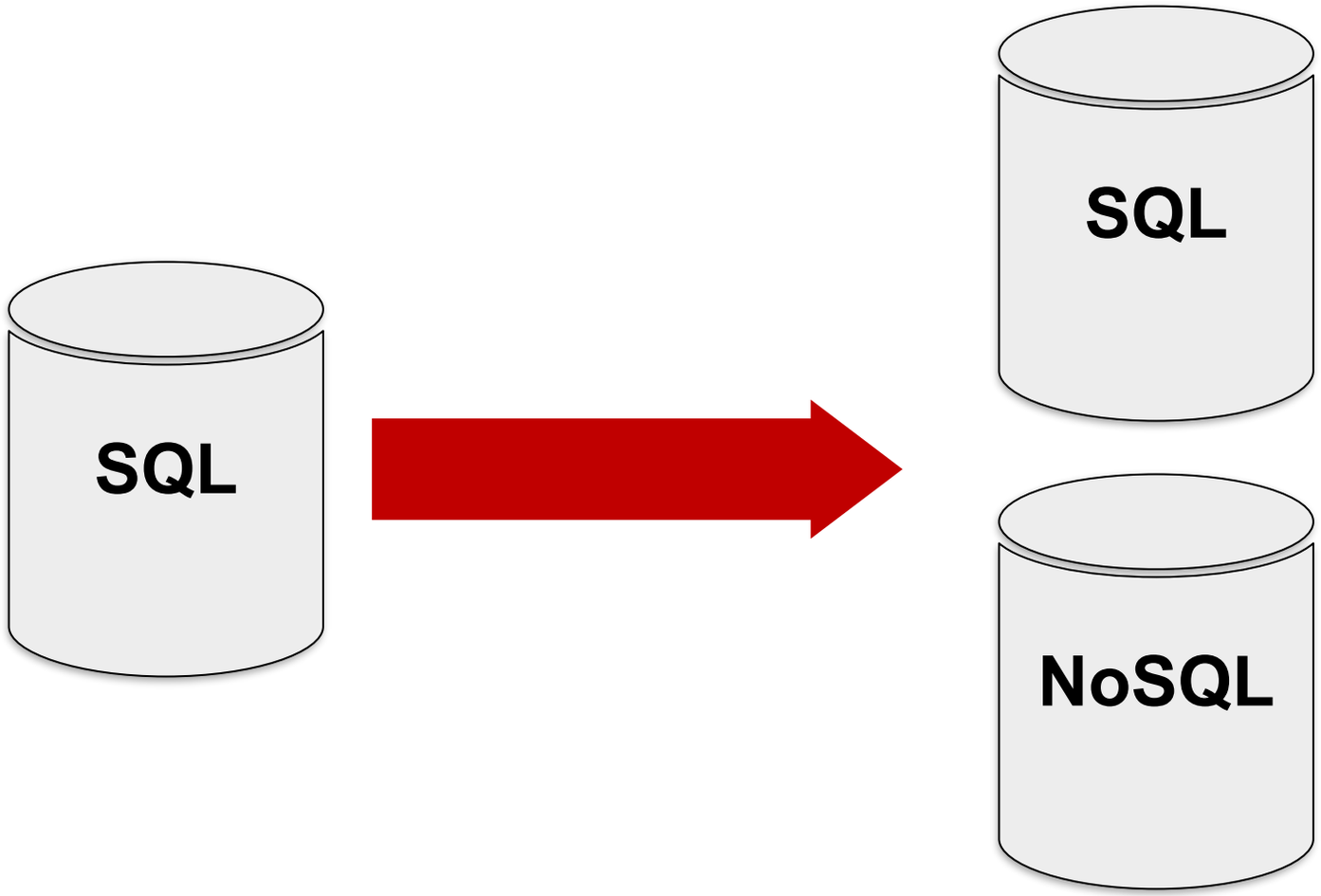
PostgreSQL. Проблемы

- Реляционная модель
- Consistency превалирует над Partition Tolerance
- Шардирование требует усилий и компетенций
- Репликация не насколько быстрая как обещается
- А тут ещё и закон Джина Амдала начинает работать

Что же делать?



Что же делать?



ClickHouse

ClickHouse

- Столбцовый принцип хранения
- Распределённость из коробки
- Как и Kafka, оптимизирована под работу пакетами, есть встроенный KafkaStreamer
- Развитые аналитические функции
- Акцент на Partition Tolerance и Availability

Наступило счастье!

Наступило счастье!

Нет ничего бесплатного

Clickhouse. Слабые стороны

Clickhouse. Слабые стороны

- Нет полноценного оптимизатора запросов

Clickhouse. Слабые стороны

- Нет полноценного оптимизатора запросов
- Низкая скорость точечного чтения

Clickhouse. Слабые стороны

- Нет полноценного оптимизатора запросов
- Низкая скорость точечного чтения
- Проблемы при нагрузке на чтение

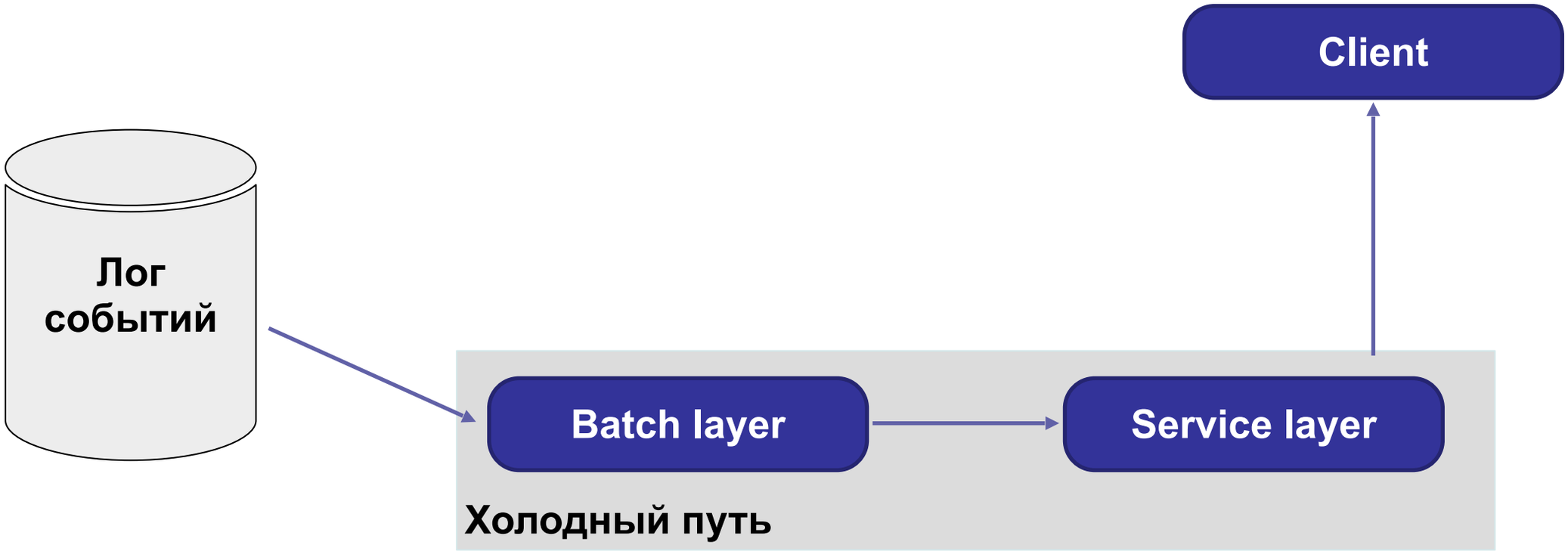
Clickhouse. Слабые стороны

- Нет полноценного оптимизатора запросов
- Низкая скорость точечного чтения
- Проблемы при нагрузке на чтение
- Нет полной консистентности

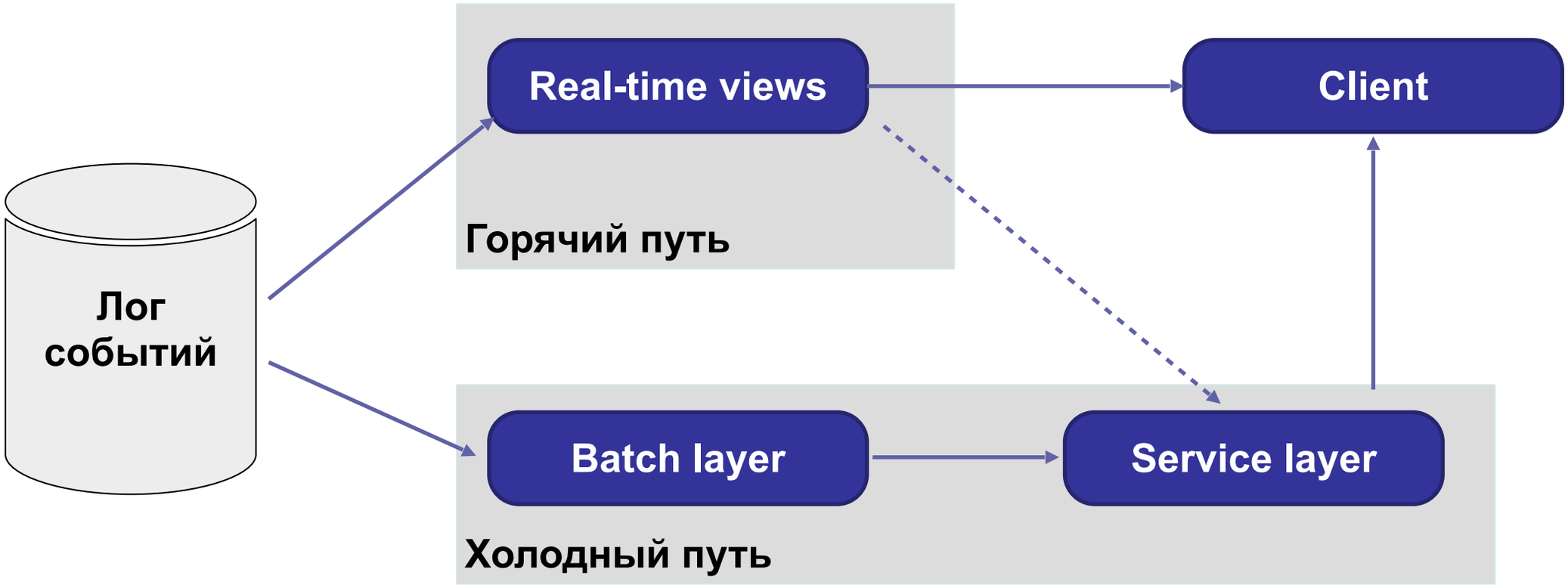
Clickhouse. Слабые стороны

- Нет полноценного оптимизатора запросов
- Низкая скорость точечного чтения
- Проблемы при нагрузке на чтение
- Нет полной консистентности
- Поддержка SQL не полная

Лямбда-архитектура



Лямбда-архитектура



Лямбда-архитектура

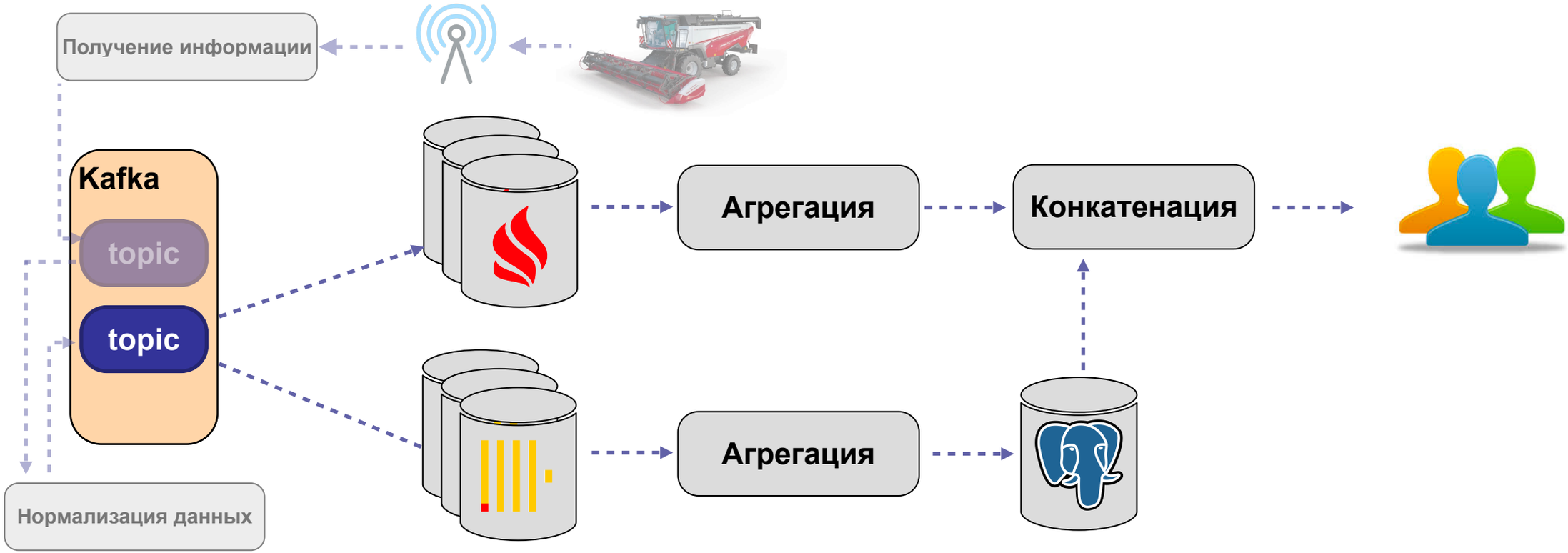
$$\text{Query} = \lambda(\text{complete data}) =$$

$$\lambda(\text{live streaming}) \circ \lambda(\text{stored})$$

Ignite

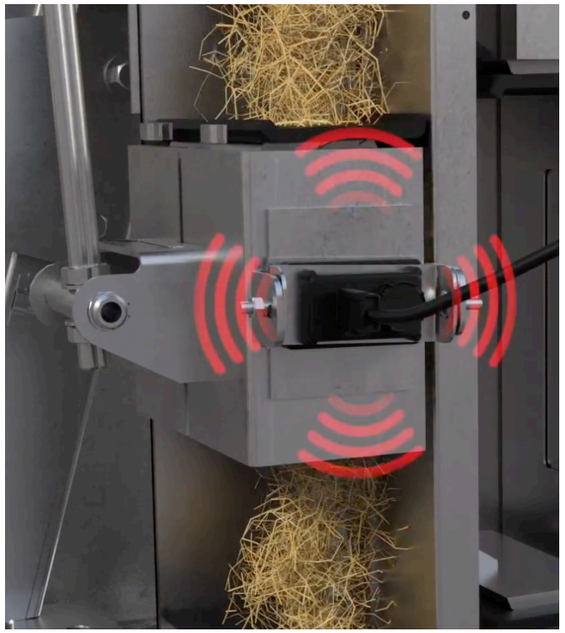
- In-memory DB
- Встраивается в приложение
- Надежные механизмы резервирования данных
- Очень быстрый встроенный KafkaStreamer
- Распределенные запросы и вычисления

λ - архитектура

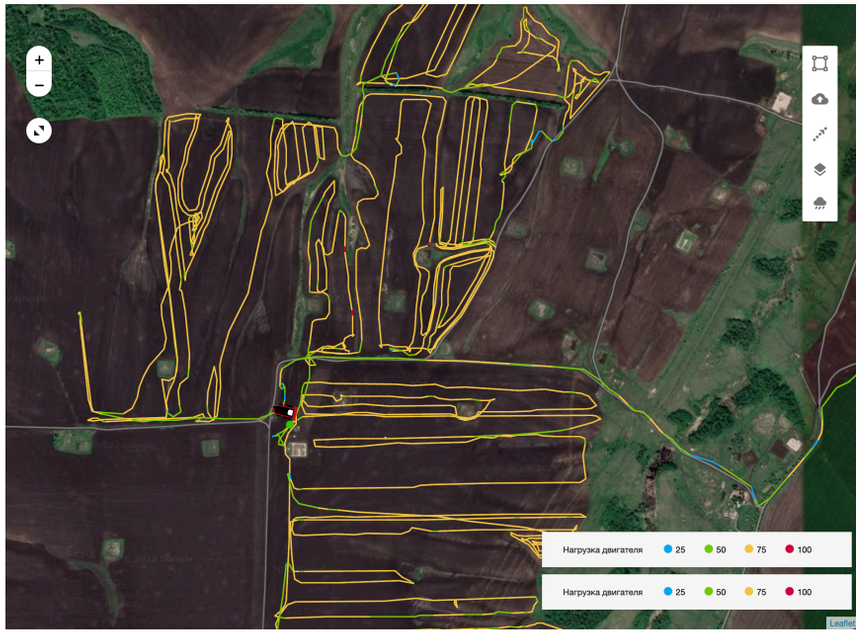


РОСТСЕЛЬМАШ

λ::Актуальные данные

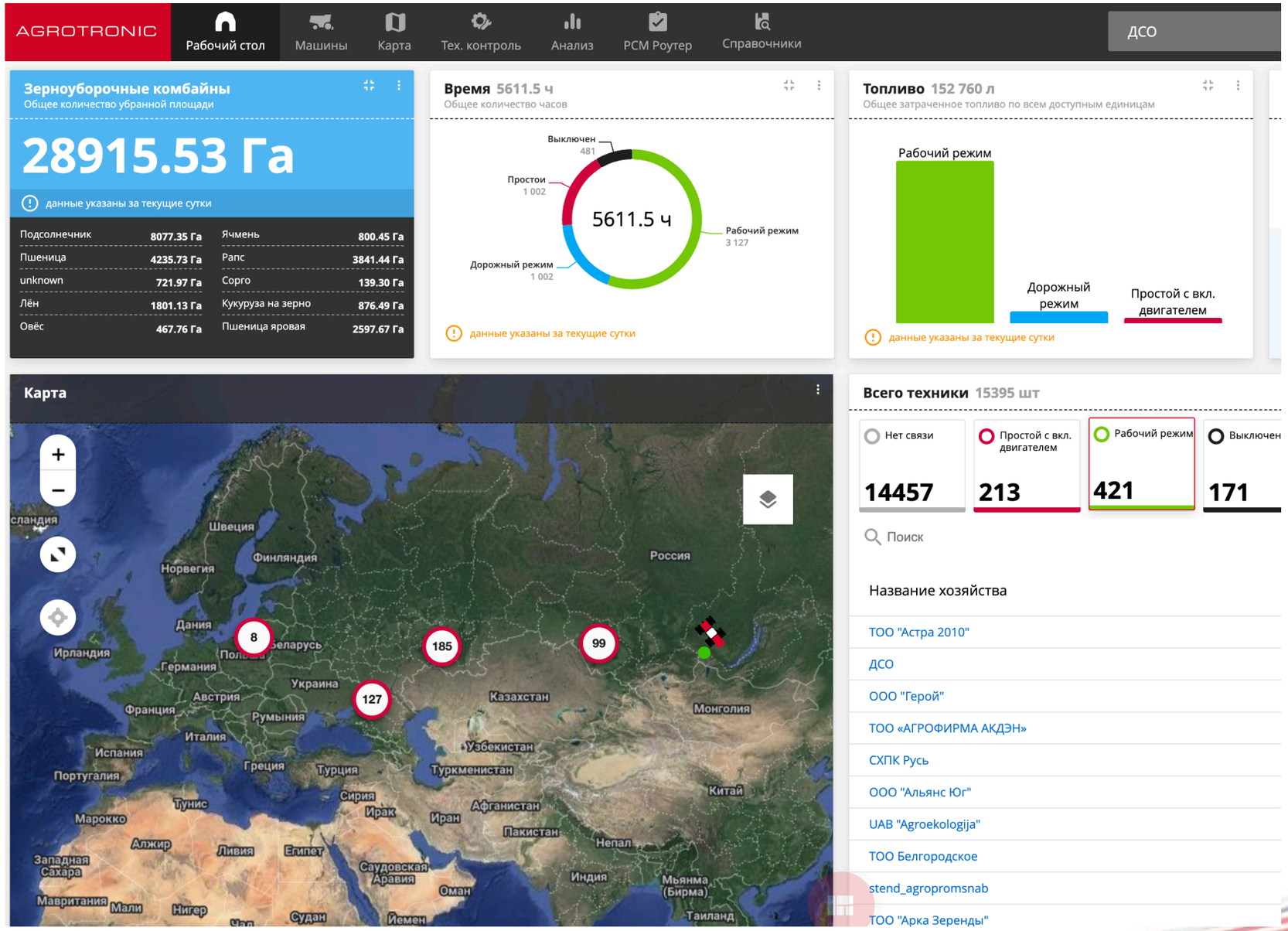


AGROTRONIC

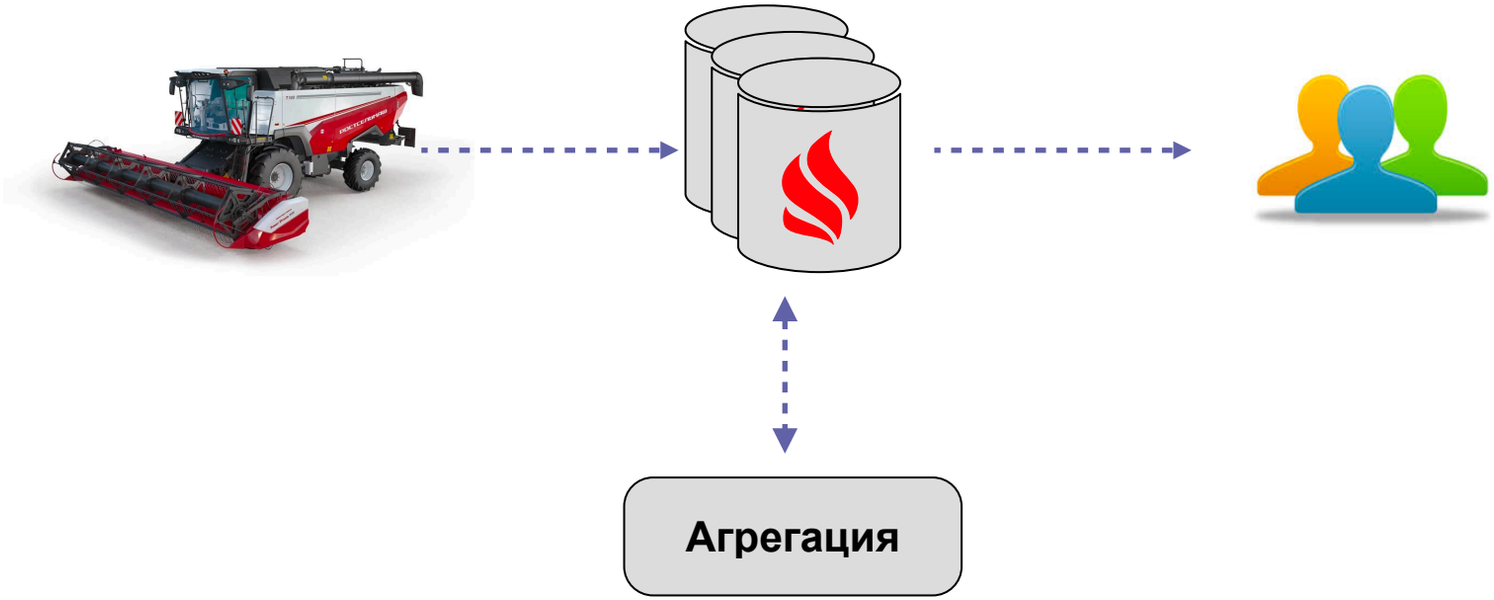


- Данные inMemory + агрегированные
- Возможно наблюдение за машиной в online
- Ограничены отслеживаемые показатели

Применение::Сводные показатели



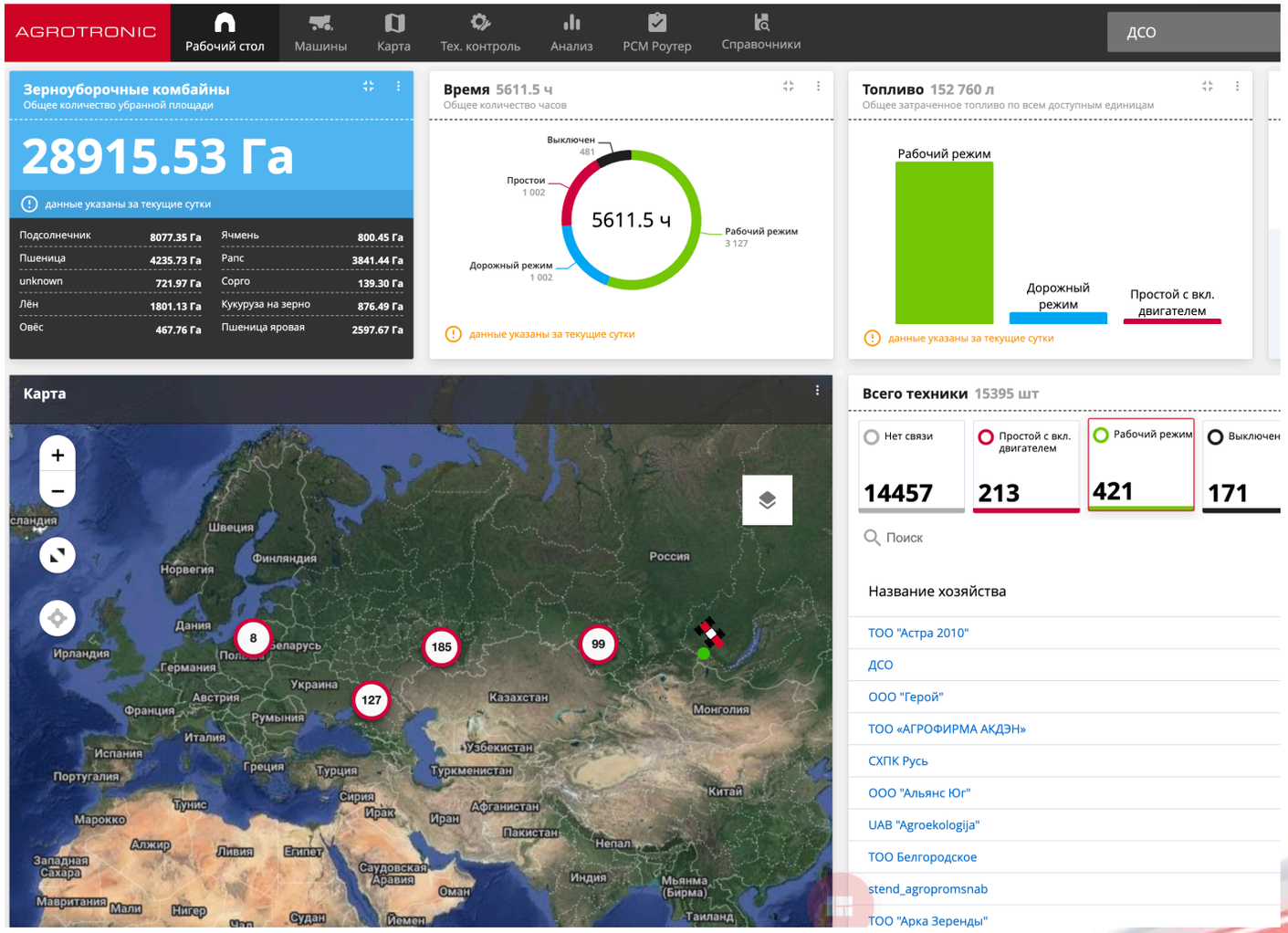
Применение::Сводные показатели



РОСТСЕЛЬМАШ

Применение::Сводные показатели

- Данные inMemory + промежуточный агрегатор
- Сводная информация о машинах в online
- Ограничены отслеживаемые показатели



РОСТСЕЛЬМАШ

Realtime реакция на событие

Realtime реакция на событие

- Заполнение бункера до определенного уровня

Realtime реакция на событие

- Заполнение бункера до определенного уровня
- Критические состояния/ режимы оборудования, сливы топлива

Realtime реакция на событие

- Заполнение бункера до определенного уровня
- Критические состояния/ режимы оборудования, сливы топлива
- Неавторизованный водитель

РОСТСЕЛЬМАШ

Заполнение бункера

```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

РОСТСЕЛЬМАШ

Заполнение бункера

```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

Ссылка на вычислитель

РОСТСЕЛЬМАШ

Заполнение бункера

```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

Ссылка на вычислитель

Запрос

РОСТСЕЛЬМАШ

Заполнение бункера

```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

Ссылка на вычислитель

Запрос

Работа по локальным данным ноды

РОСТСЕЛЬМАШ

Заполнение бункера

```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

Ссылка на вычислитель

Запрос

РОСТСЕЛЬМАШ

Заполнение бункера

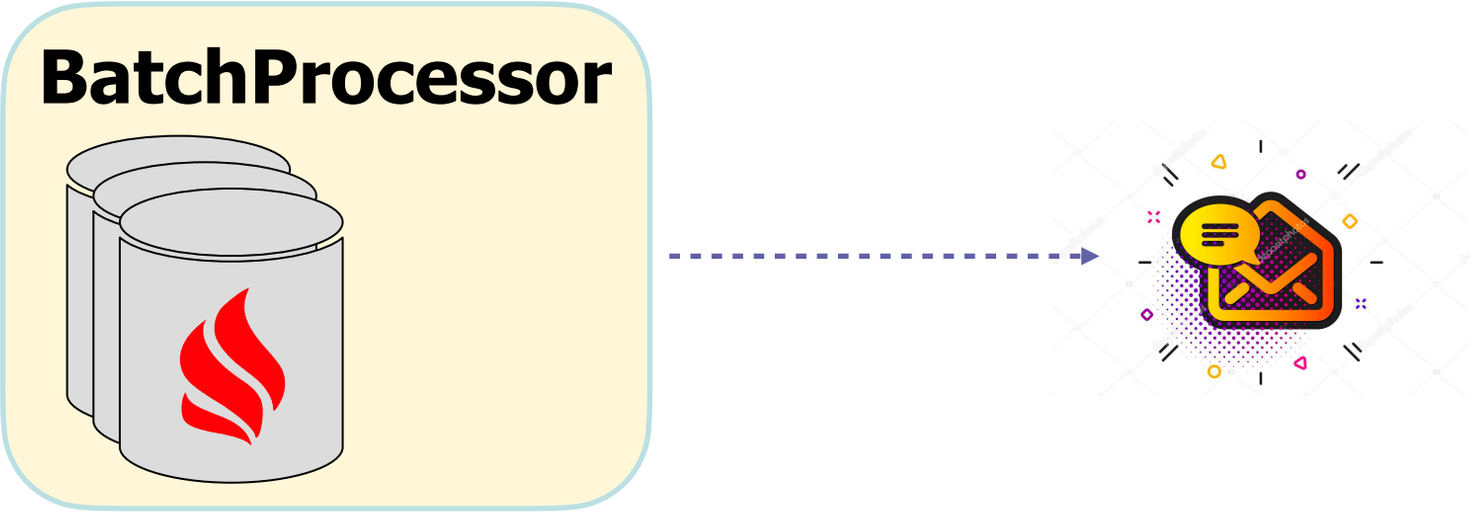
```
public void checkHopperFilling( LocalDateTime rubiconDate ) {
    IgniteCompute compute = ignite.compute();
    SqlFieldsQuery query = new SqlFieldsQuery(
            " select imei, vehicle_time " +
            "   from messages " +
            " where vehicle_time > ? " +
            "   and hooper_load > 0.7 " )
            .setArgs( rubiconDate )
            .setLocal( true );
    compute.broadcast( () -> {
        QueryCursor<List<?>> queryCursor = messagesCache.query( query );
        for ( List<?> item : queryCursor ) {
            notificationService.alertHooperFill(
             item.get( 0 ), item.get( 1 ), 0.7
             );
        }
    } );
}
```

Ссылка на вычислитель

Запрос

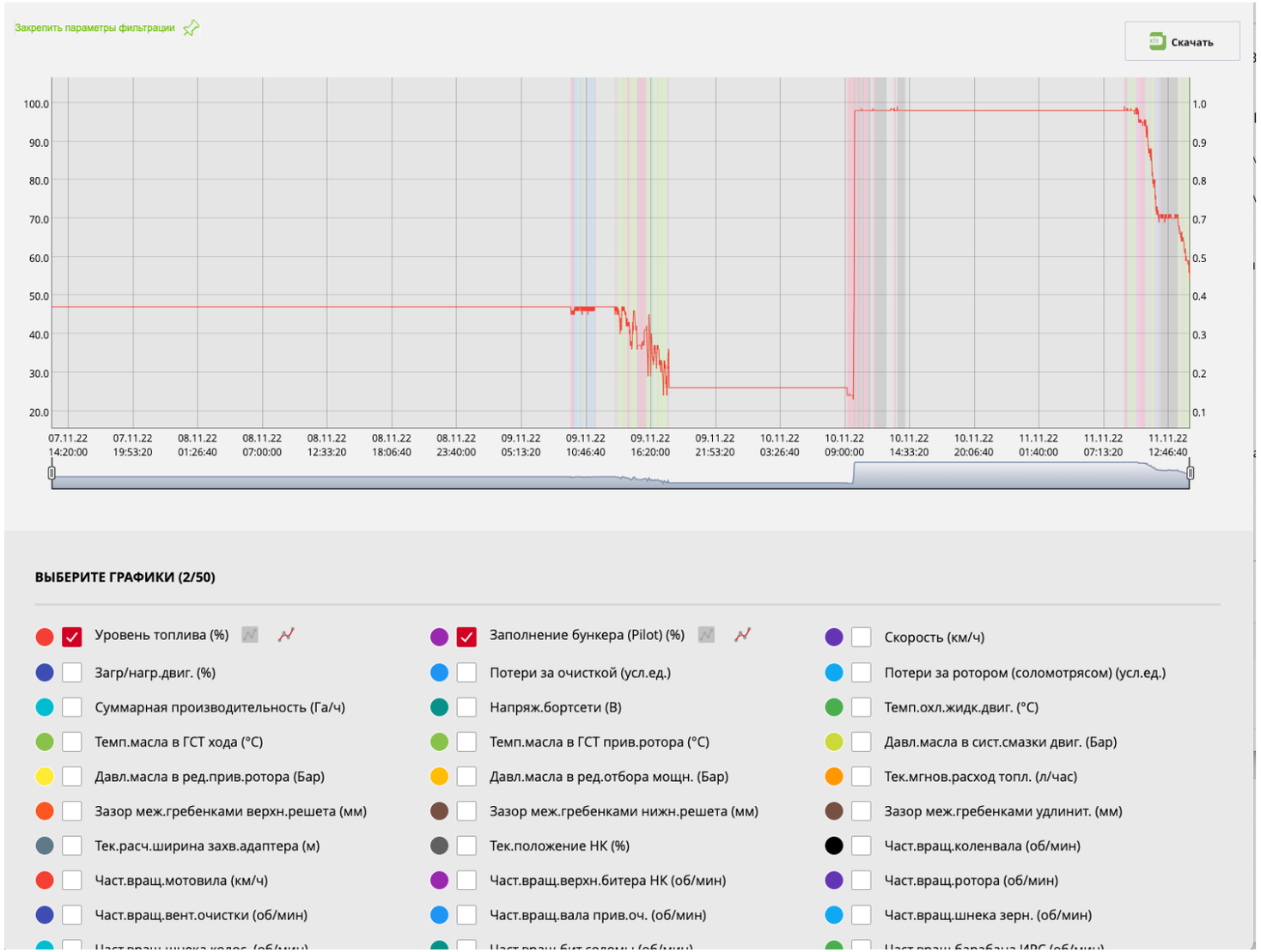
Запуск на исполнение по кластеру

Применение::Реакция на события



- Запрос распределяется на все ноды и может работать только с локальными данными
- Очень быстро работает на свежих данных
- Реализуется на обычном языке программирования (Java, C#, C++)

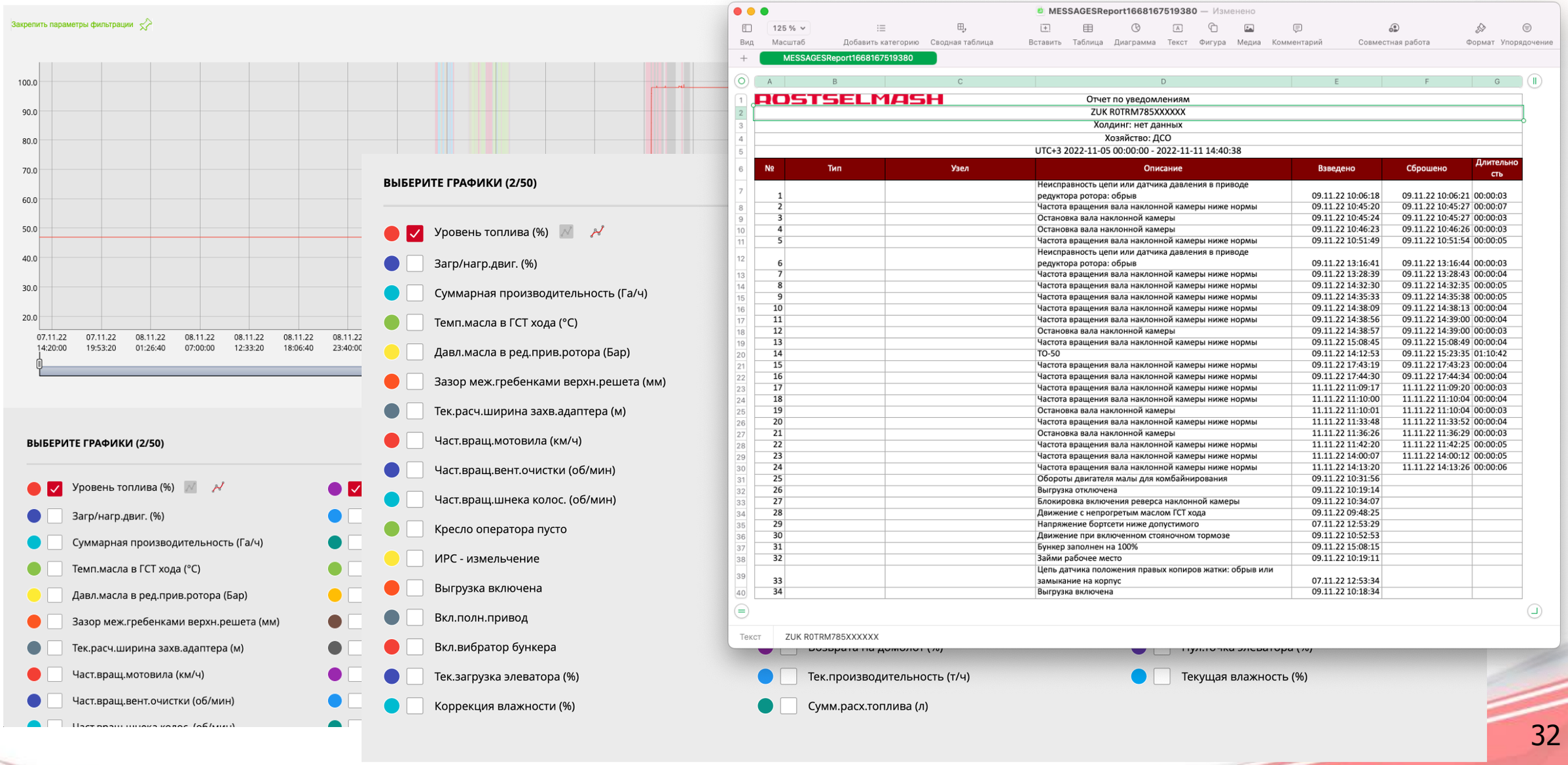
Применение::Отчеты



РОСТСЕЛЬМАШ

Применение::Отчеты

Применение::Отчеты



РОСТСЕЛЬМАШ

Применение::Отчеты

```
select imei, carrier, agriculture, sum(delta) as uploads
from (
          select imei, carrier, agriculture,
                 runningDifference(on_upload) as delta
          from (
                   select imei, carrier, agriculture, on_upload
                     from messages
                            inner join units u on messages.imei = u.imei
                    where u.id in %s
                      and vehicle_time between '%s' and '%s'
                    order by vehicle_time)
          where delta = 1)
group by agriculture, carrier, imei
```

РОСТСЕЛЬМАШ

Применение::Отчеты

```
select imei, carrier, agriculture, sum(delta) as uploads
from (
        select imei, carrier, agriculture,
               runningDifference(on_upload) as delta
        from (
                select imei, carrier, agriculture, on_upload
                  from messages
                         inner join units u on messages.imei = u.imei
                 where u.id in %s
                   and vehicle_time between '%s' and '%s'
                 order by vehicle_time)
        where delta = 1)
group by agriculture, carrier, imei
```

Изменение значения

Применение::Отчеты

- Аналитические возможности Clickhouse
- Хранятся «вечно» и надежно
- Возможны интеграции с BI или BigData

РОСТСЕЛЬМАШ

Архитектура решения

λ-архитектура::Итоги

- Работает только в событийной модели обработки
- Разделены слои хранения данных и слои подготовки представлений
- Требуется разработка сервисов подготовки данных
- Хорошие возможности масштабирования
- Ignite позволяет реализовать многие задачи горячего слоя

λ-архитектура::Итоги

- **Выгода**
  - ✓ Сохранность исторических данных
  - ✓ Баланс скорости и надежности
  - ✓ Масштабируемость
- **Недостатки**
  - ✓ Сложность изменения стратегии хранения и анализа на лету
  - ✓ Общая сложность решения, требующая хороших практик микросервисной архитектуры

Finally

РОСТСЕЛЬМАШ

Finally

- В сельхозе много интересных задач для разработчиков

РОСТСЕЛЬМАШ

Finally

- В сельхозе много интересных задач для разработчиков
- Лямбда архитектура отлично работает для Интернета-вещей

РОСТСЕЛЬМАШ

Finally

- В сельхозе много интересных задач для разработчиков
- Лямбда архитектура отлично работает для Интернета-вещей
- Возможности Apache Ignite позволяют сделать больше, чем просто горячий слой хранения

РОСТСЕЛЬМАШ

Спасибо за внимание