

Сказ про то, как мы в

СИТИМОБИЛ

DWH строим

от реплик `mySQL` до `Exasol` + `ClickHouse`

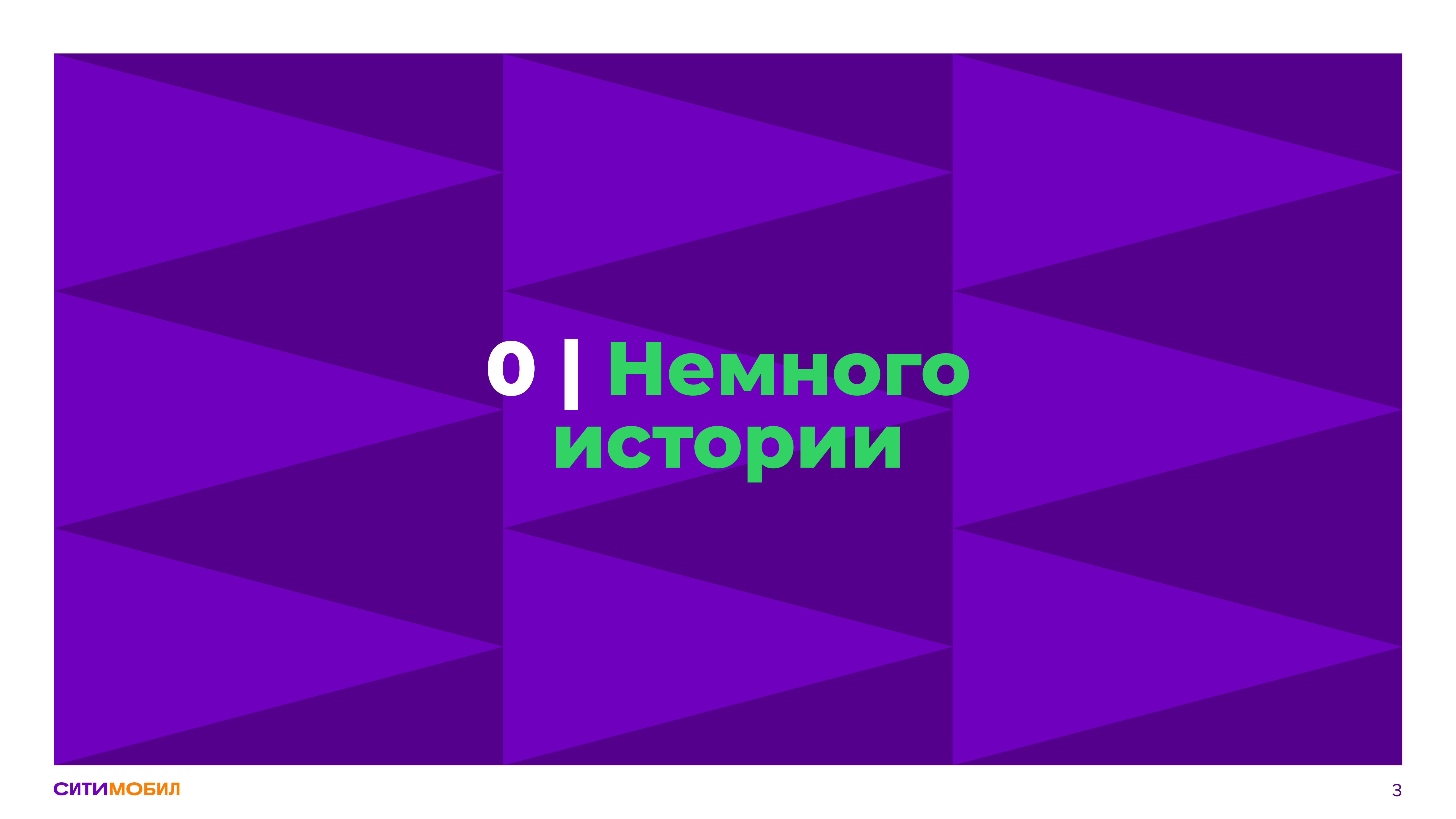
СИТИ**МОБИЛ**

Привет! 

**Я Катя
Колпакова**

Руководитель DWH в Ситимобил





0 | Немного истории

Немного истории

Не только лишь такси единым

Основные даты:

2018 - партнерство с VK Company (устар. Mail.Ru Group)

2019 - совместное предприятие VK и Сбербанк

2019 → 2021

50 человек → 1500 человек

30 таблиц → 2000 таблиц

С 2007 года

Приложение

Появилось у нас еще в те времена, когда это не стало мейнстримом

Сейчас



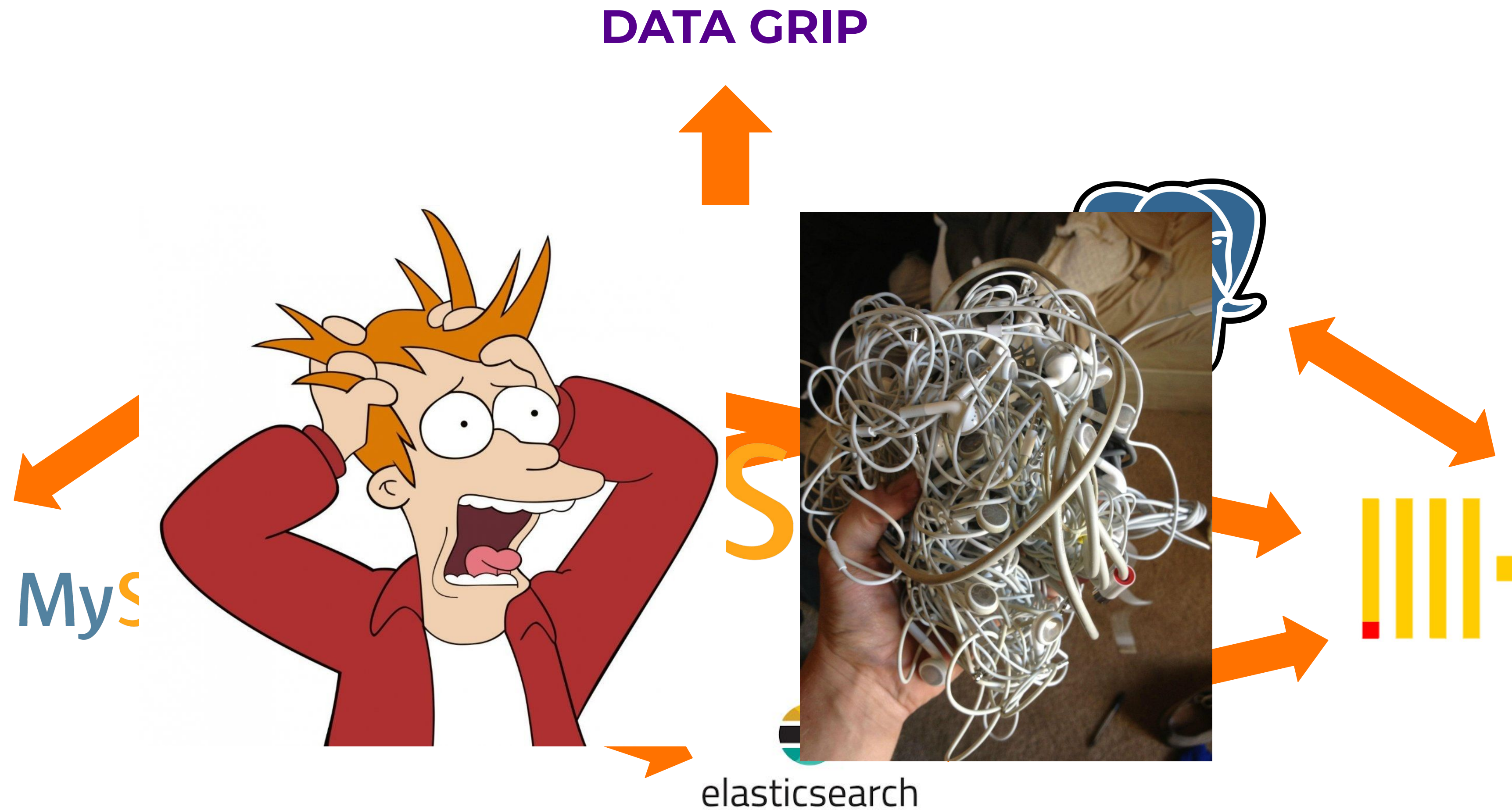
В будущем

Сервис городской мобильности

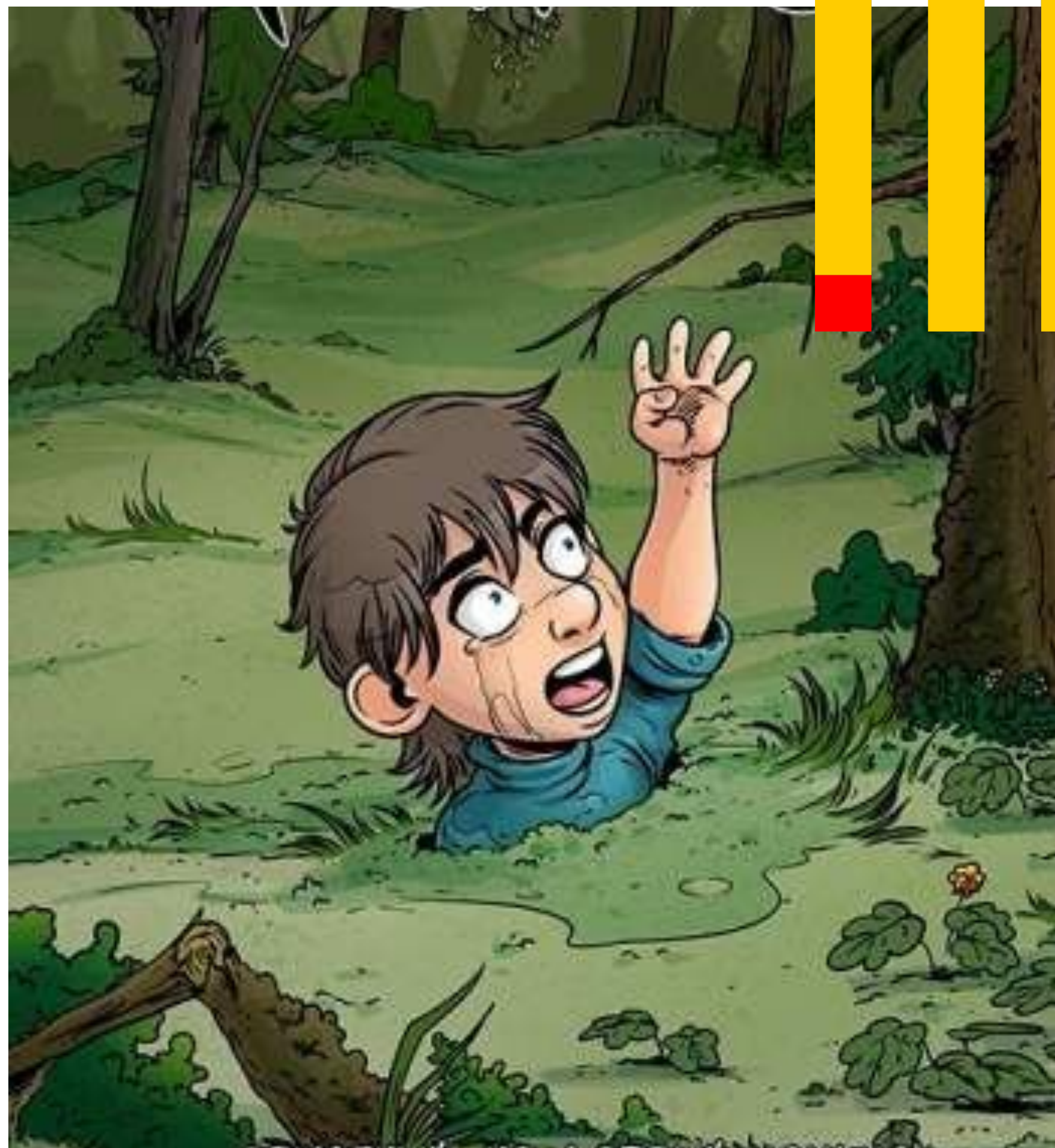


Совсем не аналитическая БД

Компания была маленькой, но планы были большими.



Эпоха Data Lake на Clickhouse



Эпоха ClickHouse Data Lake

Концепция Data Lake не подразумевает никаких сложных манипуляций с данными.

Забрал данные на источнике - добавил в таблицу.



Data Lake на CH

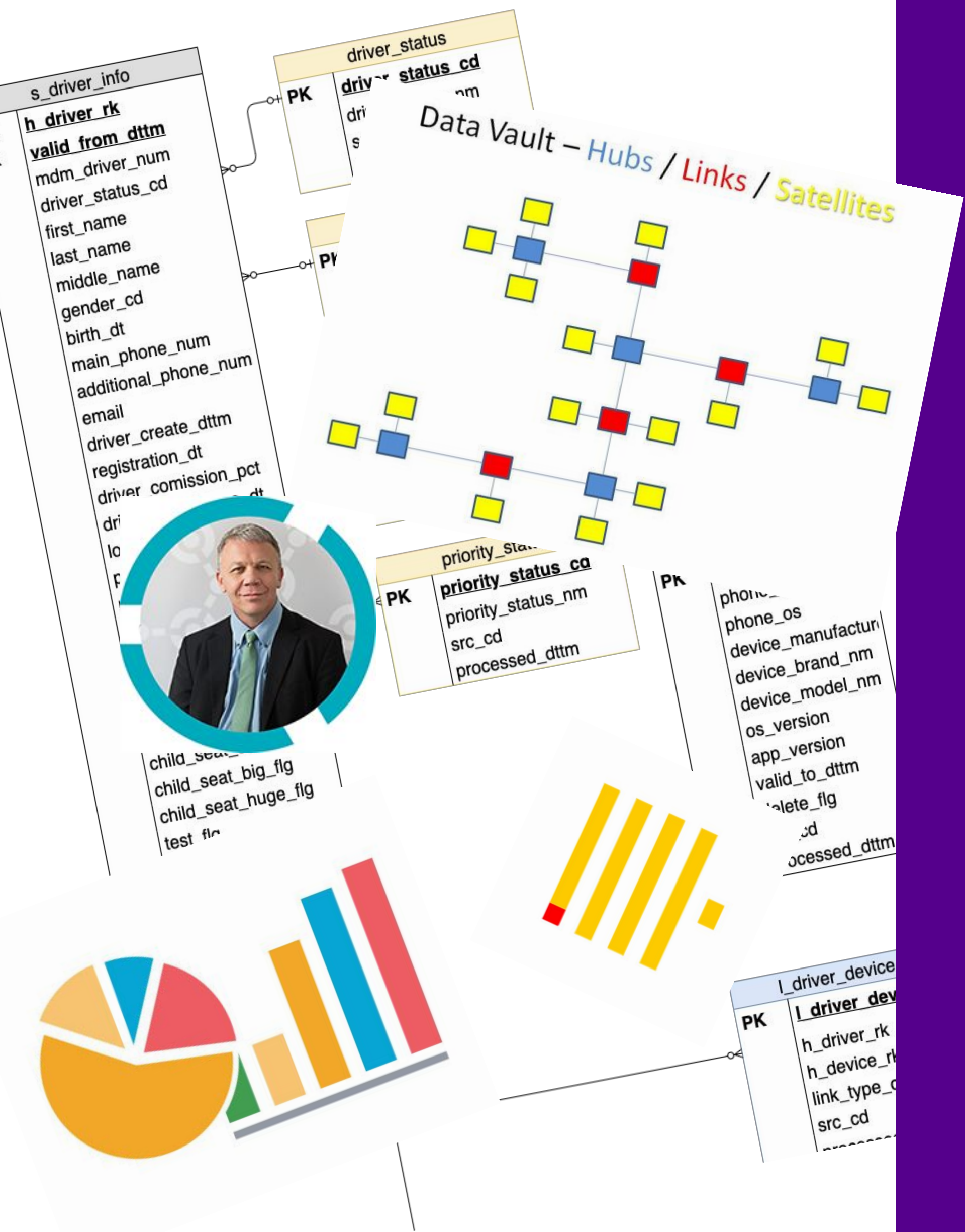
Хорошо:

- ▶ Концепция DL - это просто
- ▶ CH: быстрый select
- ▶ CH + Kafka = ❤️
- ▶ Опенсорность и развитие

Плохо:

- ▶ Нет гарантий, что данные доедут
- ▶ Нет классическому DWH (K, I) с DQ и версионностью
- ▶ SQL не для всех

1.1 | Чего хотелось от DWH и будущей СУБД для DWH



- 1) Возможность без костылей реализовать **классическое** хранилище для аналитики с версионностью **SCD2**
- В качестве логической модели выбран Data Vault
- 2) Скорость выполнения **часто используемых запросов** не должна быть значительно хуже, чем в ClickHouse

Выбор СУБД. Какие таблицы мы джойнили? И зачем?

Это

- ▶ таблица заказов
- ▶ таблица калькуляций*

Они очень часто используются аналитиками в связке**.

* калькуляция - это когда вы открываете приложение и прокладываете маршрут, а приложение рассчитывает цену

**даже сейчас, несмотря на наличие витрин, аналитики часто ищут что-то свое

Таблица заказов:

- ▶ около 300 млн записей за период

Join

- ▶ по id заказа (bigint)

Фильтр

- ▶ по статусу заказа и/или полигону и/или любому другому измерению

Таблица калькуляций:

- ▶ около 1 млрд записей за период

Агрегация 🙌

- ▶ sum, count, etc.

НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ

**И ОКАЗАТЬСЯ БЫСТРЕЕ
CLICKHOUSE**

risovach.ru

Результаты тестирования различных MPP RDBMS По увеличению времени выполнения первого запроса

Мы всегда выступаем за соблюдение
скоростного режима **на дорогах
общего пользования.**

Но в случае **доставки данных** нам
нужна действительно гоночная
скорость.



Exasol



ClickHouse



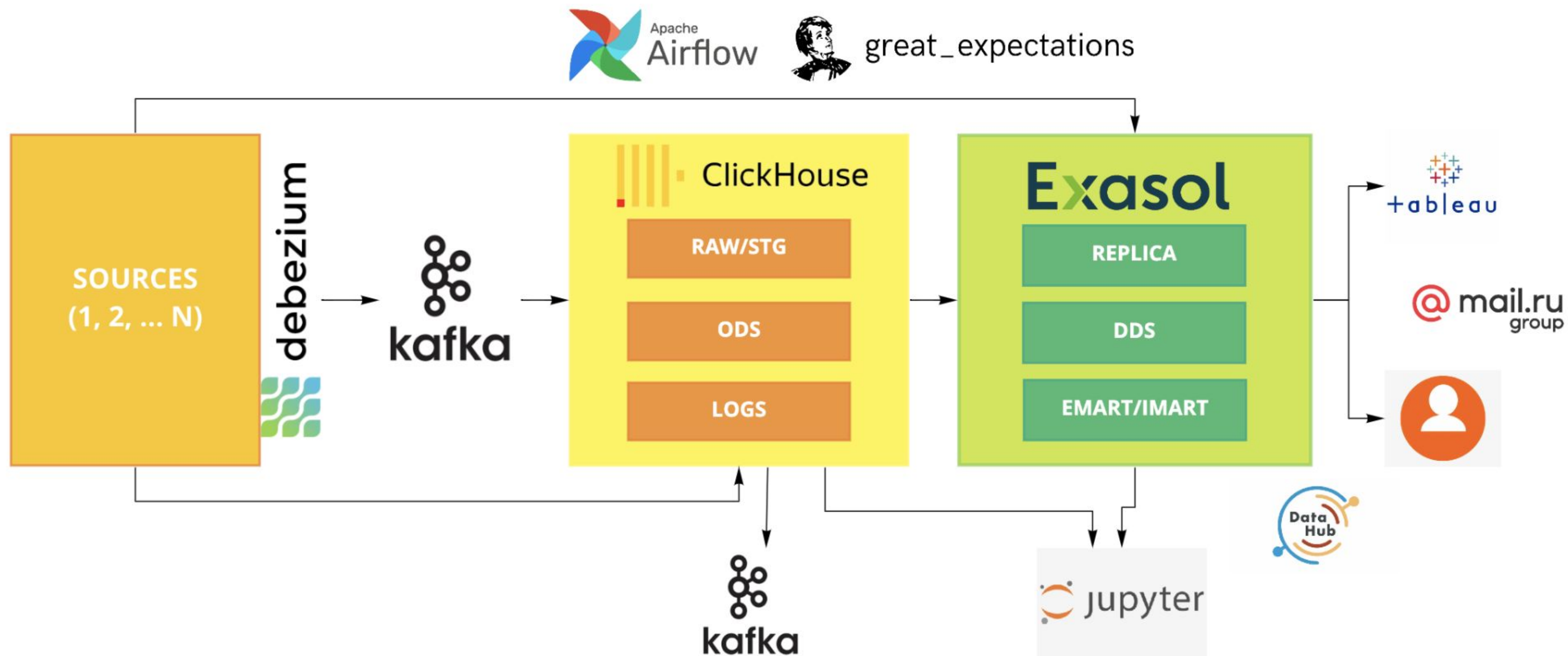
VERTICA



Greenplum

1.2 | Что получилось

Архитектура DWH Технологии



Технологии хранилища



СУБД

ClickHouse - для стейджинга, ods и логов

Exasol - для ядра хранилища и витрин

Модель данных

Data Vault



ELT

Airflow + самописные операторы

DQ

Great Expectations

Самописные проверки



BI

Tableau

Redash

Обновление

1 раз в 15 минут/час/сутки

300 пользователей

Exasol

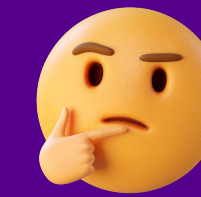
Плюсы и минусы

- In-Memory MPP
- SQL, знакомый каждому аналитику
-> низкий порог вхождения
- Приятным плюсом стало огромное количество аналитических функций из коробки



Плюсы Exasol:

- ▶ Разворачивание нод по щелчку мышью
- ▶ Стабильность базы
- ▶ Можно писать свои UDF
- ▶ Все очень быстро и не требует усилий
- ▶ Хорошо жмет данные



Минусы Exasol:

- ▶ Черный ящик 
- ▶ Нет некоторых привычных типов данных (array)
- ▶ Требовательный к железу

ClickHouse для STG



ClickHouse для STG

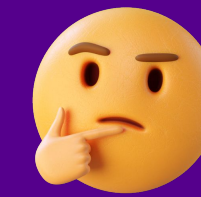
Плюсы и минусы

- Так исторически сложилось, что косим траву молотком



Плюсы CH:

- ▶ Бесплатный
- ▶ Быстрый
- ▶ Экосистема вокруг
- ▶ Дружит с Kafka



Минусы CH:

- ▶ Данные не доезжают (нет транзакций)
- ▶ Сложность отладки
- ▶ Нет нормальных update/delete (не мутаций)



Совет начинающему дэвэхашнику

- Всегда идите от задач, а не от инструмента (даже если это стильная, модная, молодежная технология и ее ну очень хочется использовать)
- “Опенсорсно” не значит “дешево”

02 | Модель данных

Data Vault

Простым языком

Третья нормальная форма и выше.

Пример:

Хаб: Водитель

Линк: Связь Водителя и Машины

Сателлит: Атрибуты (описания)
водителя

3 типа объектов Data Vault

Хаб

Сущность: ее
постоянный ключ, ее
бизнес ключ

Линк

Связи между
сущностями

Сателлит

Или линк-сателлит. Это
атрибуты сущностей (с
версионностью).

Когда это удобно?
Не все модели
данных одинаково
полезны

И Data Vault -- тоже далеко не всегда.

Почему мы выбрали Data Vault?

Автоматизация всего

Возможность
унифицировать все ETL
процессы DWH

Много изменений

Хотим легко
добавлять новое и
менять старое

03 | ELT

ELT Code-Driven

Девиз: автоматизировать все, что
можно автоматизировать.

Мы не создаем потоки
данных вручную для
каждого объекта

Мы используем
ETL/DDDL генератор
и самописные
операторы

ELT без боли и страданий



Большая часть работы по добавлению объекта в DDS выполняется системными аналитиками.

Пример результата анализа

Для добавления информации по сущности мы используем суперсовременный интерфейс (нет)

D	E	F	G	H	I	J	K	L
source_table_name	root_table_name	source_order	source_column_name	source_data_type	where	where_cols	target_order	processing
city_drivers_driver	drivers	1	id	Int64	id<>0	id	1	"id"
driver_points	drivers	2	driver_id	Int64			1	"driver_id"
driver_devices	drivers	2	id_driver	UInt64	id_driver<>0	id_driver	1	"id_driver"
driver_points_city	drivers	2	driver_id	Int64			1	"driver_id"

24	h_ads_campaign	Хаб: Рекламная кампания
25	h_car	Хаб: Машина
26	h_car_ads	Хаб: Участие автомобиля в рекламной кампании
27	h_city	Хаб: Город
28	h_client	Хаб: Клиент
29	h_client_corporation	Хаб: Сотрудник корпоративного клиента
30	h_company	Хаб: Компания
31	h_corporation	Хаб: Корпоративный клиент
32	h_country	Хаб: Страна

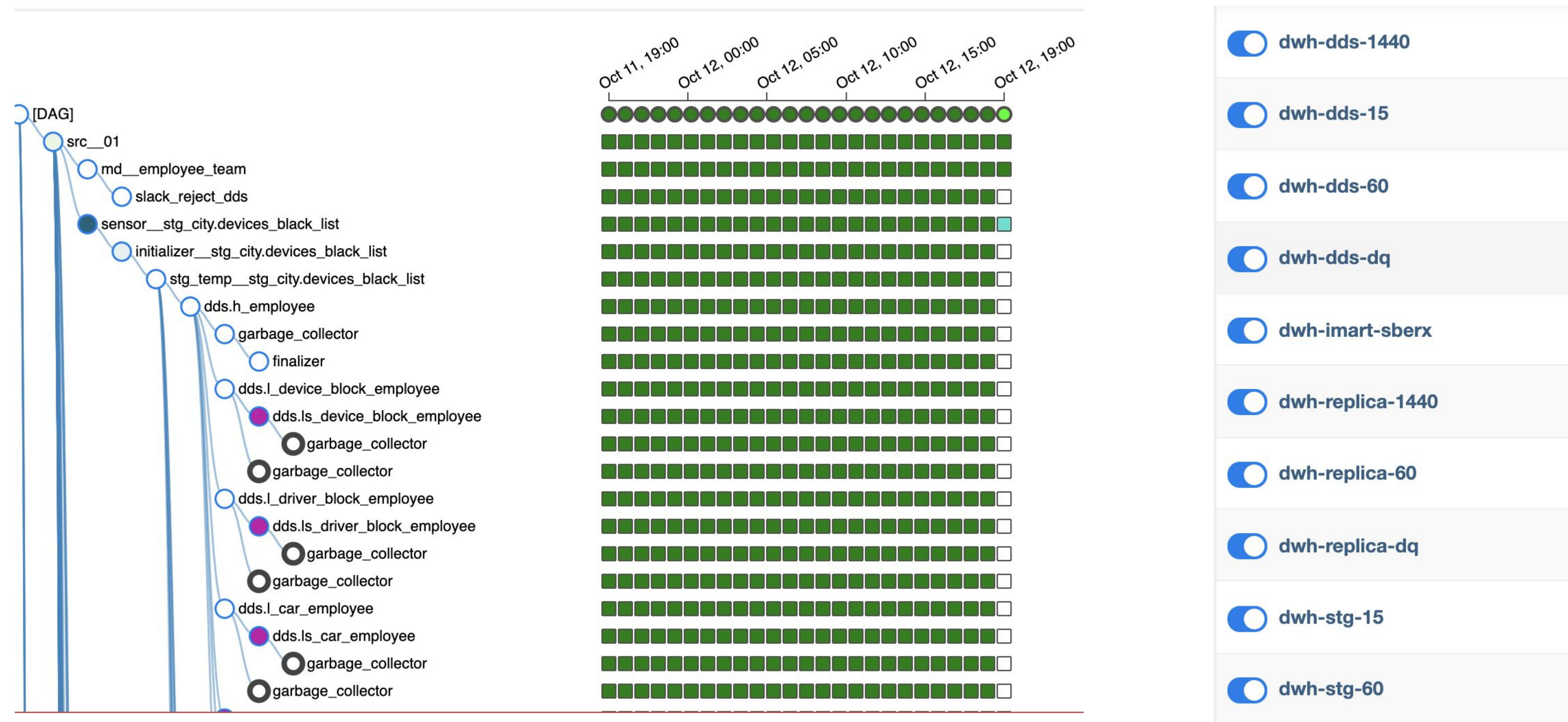
Пример YAML- файла Сгенерированного на основе анализа

- ▶ целевая схема
- ▶ имя целевой таблицы
- ▶ описание целевой таблицы (для автодоков)
- ▶ тип загрузки
- ▶ первичный ключ
- ▶ ключ дистрибуции
 - ▶ перечень полей целевой таблицы
 - ▶ перечень полей таблиц-источника, которые мэпятся в целевой поле, если нужно
 - ▶ типы целевых полей
 - ▶ описание целевых полей для автодоков
 - ▶ генерация/обработка исходных полей
- ▶ описание таблиц-источников
 - ▶ фильтры

h_driver.yml 1.65 KB

```
1  !Hub
2  target_schema: dds
3  target_table_name: h_driver
4  target_table_description: Водитель
5  load_type: APPEND
6  primary_key:
7  - h_driver_rk
8  distribute_by:
9  - h_driver_rk
10 fields: !Fields
11   h_driver_rk: !ExaField
12     data_type: !ExaType 'DECIMAL(8)'
13     comment: ID водителя
14     processing: DEFAULT
15     identity: true
16   h_driver_bk: !ExaField
17     data_type: !ExaType 'VARCHAR(2000000)'
18     source_columns:
19       1:
20       - id
21       2:
22       - driver_id
```


Как это выглядит в Airflow 2.0



Пример генерируемого кода из логов Airflow

```
INSERT INTO REJECT.DDS__H_DRIVER (row_nr, error_text, tag, csv_data, truncated)
SELECT row_nr
      , CASE WHEN LENGTH(h_driver_bk) > 2000000 THEN '"h_driver_bk" length exceeded,' ELSE '' END
      || CASE WHEN LENGTH(REGEXP_REPLACE(TO_CHAR(src_cd), '(\d+)(\.\d+)?', '\1')) > 2 THEN '"src_cd" precision exceeded,' ELSE '' END
      AS ERROR_TEXT
      , '2021-10-12 19:00:00' AS TAG
      , TO_CHAR('"' || REPLACE(h_driver_bk, '"', '\\"') || '"') ||
      ',' || TO_CHAR(src_cd) ||
      ',' || TO_CHAR(processed_dttm) AS CSV_DATA
      , LENGTH(local.CSV_DATA) > 2000000 AS TRUNCATED
FROM ETL.DDS__H_DRIVER__INC_211012_160000_PROCESSING
WHERE LENGTH(h_driver_bk) > 2000000
      OR LENGTH(REGEXP_REPLACE(TO_CHAR(src_cd), '(\d+)(\.\d+)?', '\1')) > 2
[2021-10-12 20:03:43,809] {exasol_hook.py:32} 88307:INFO - Number of affected rows - 0.
[2021-10-12 20:03:43,809] {exasol_hook.py:27} 88307:INFO - COMMIT
[2021-10-12 20:03:44,792] {exasol_hook.py:32} 88307:INFO - Number of affected rows - 0.
[2021-10-12 20:03:44,824] {exasol_hook.py:27} 88307:INFO -
INSERT INTO dds.h_driver
(h_driver_bk, src_cd, processed_dttm)
WITH deduplicated AS (
SELECT DISTINCT
      CAST(T.h_driver_bk AS VARCHAR(2000000)) AS h_driver_bk
      , CAST(T.src_cd AS DECIMAL(2)) AS src_cd
      , CAST(T.processed_dttm AS TIMESTAMP) AS processed_dttm
FROM ETL.DDS__H_DRIVER__INC_211012_160000_PROCESSING T
LEFT JOIN REJECT.DDS__H_DRIVER R
      ON T.row_nr = R.row_nr
      AND R.tag = '2021-10-12 19:00:00'
WHERE R.row_nr IS NULL
)
, ended_table AS (
```

Чего еще интересненького ? Другие операторы

RAW/STG-оператор

- на основе MV и Airflow. схлопывает сырые данные с требуемой гранулярностью
- переписывается для Exasol

MART-оператор

- оператор для загрузки витрин на основе jinja шаблона

Репликатор

- “возьми таблицу откуда-нибудь и загрузи ее в Exasol”
- работает на основе конфига, заполняемого аналитиком

3 других классных оператора

MART-оператор

Оператор для загрузки витрин на основе jinja шаблонов

RAW/STG Оператор

Загрузка данных в СН из ИСТОЧНИКОВ

Репликатор

Первый претендент на перевод в self service

03 | Data Quality

Что такое качество данных?
Высококачественные данные

Высококачественные данные 🙅

=

соответствующие
потребностям и ожиданиям
потребителей

Что такое качество данных?

Метрики качества

- ▶ **Актуальность** - насколько данные отстают от источника
- ▶ **Допустимость** - насколько поля соответствуют области допустимых значений
- ▶ **Полнота** - заполненность в обязательных/необязательных атрибутах
- ▶ **Разумность** - проверки на здравый смысл
- ▶ **Непротиворечивость** данных в различных источниках
- ▶ **Отсутствие дубликатов**
- ▶ **Целостность** - ссылочная связность и отсутствие потерь

FIGHT!! Самописный фреймворк vs Great Expectations



Дуэлянт 1 - great_expectations

Плюсы:

- ▶ Opensource и 21млн\$ инвестиций
- ▶ Интеграция с Slack
- ▶ Потенциальная поддержка DataHub
- ▶ Комьюнити (4,8к в Slack)
- ▶ Datadocs
- ▶ Можно подружить с Airflow

Минусы:

- ▶ Сложный код
- ▶ Нет нативного CH и Exasol
- ▶ Нет запуска по кнопке
- ▶ Много нужно допиливать

Дуэлянт 2 - Самописный инструмент

Плюсы:

- ▶ Получим именно то что хотим
- ▶ Команда знает код

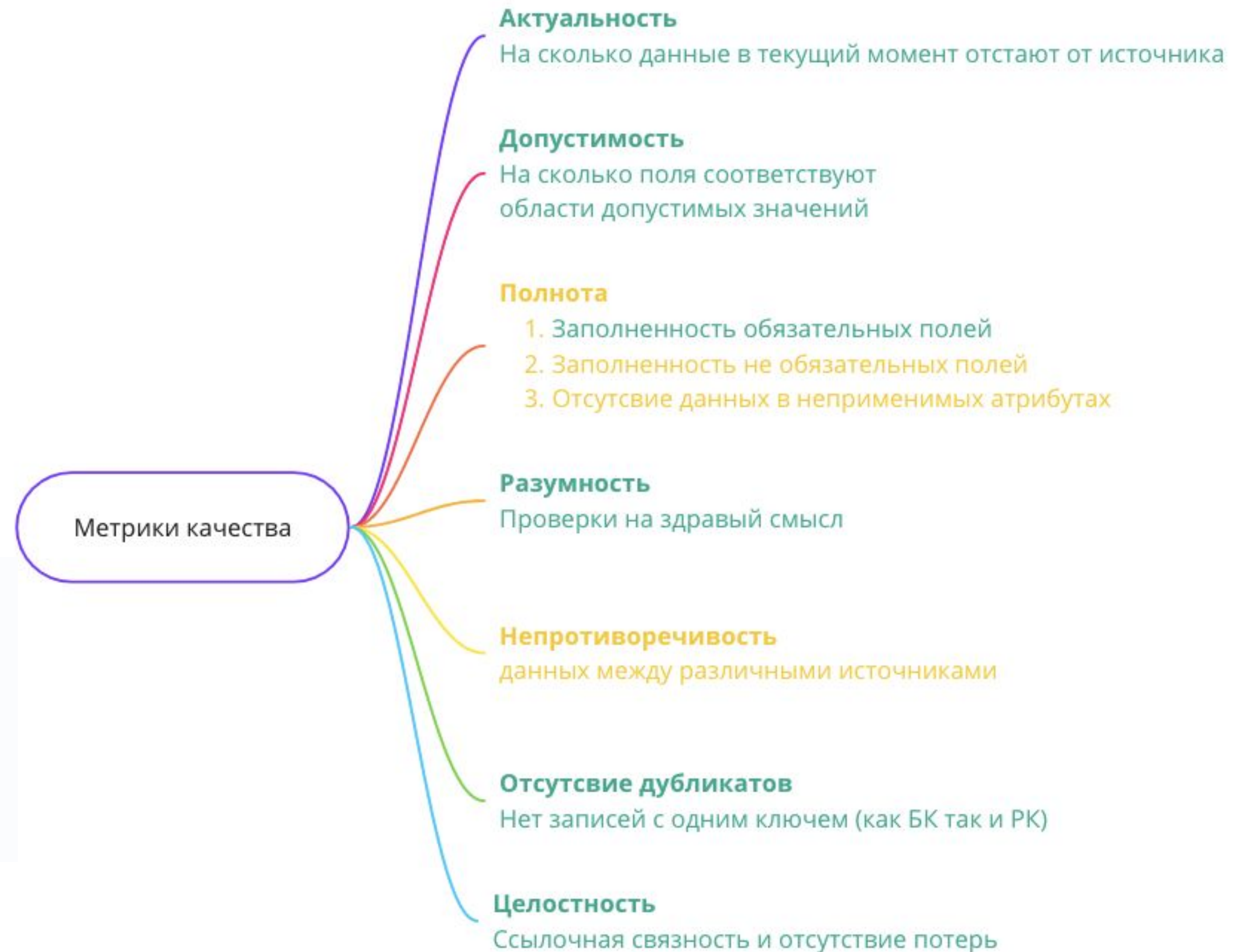
Минусы:

- ▶ Долго разрабатывать
- ▶ Возможны архитектурные просчеты
- ▶ Придется все делать самим
- ▶ Bus фактор

Большие ожидания к фреймворку Great Expectations



Что можно и что можно_с_доработками ...сделать с помощью фреймворка Great Expectations



great_expectations

Expectations

Expectation - некоторое утверждение (ожидание) которое нужно оценить и являющееся семантически значимым для человека, например:

`expect_column_values_to_be_unique` или `expect_column_mean_to_be_between`

Реализованы в виде классов и имеют метод `_validate` который определяет удовлетворяют ли данные утверждению или нет.

Metric - любое наблюдаемое свойство данных, **Expectations** оцениваются с помощью метрик.

Expectation Configuration - описывает определенные ожидания от набора данных. Включает в себя Expectation и набор параметров по которым нужно оценить данные

```
batch.expect_column_values_to_be_in_set(  
    column="idConnectionType",  
    value_set={"$PARAMETER": "urn:great_expectations:stores:my_postgres_db:client_debt_current_count"}  
)
```


02 | great_expectations Expectation Suite

Expectation Suite - набор **Expectation Configuration** являющихся общим описанием набора данных.

Именуются в соответствии с типом данных которые они описывают, например:

city.users или city.tariff

Сохраняется в виде простого json файла на диск.

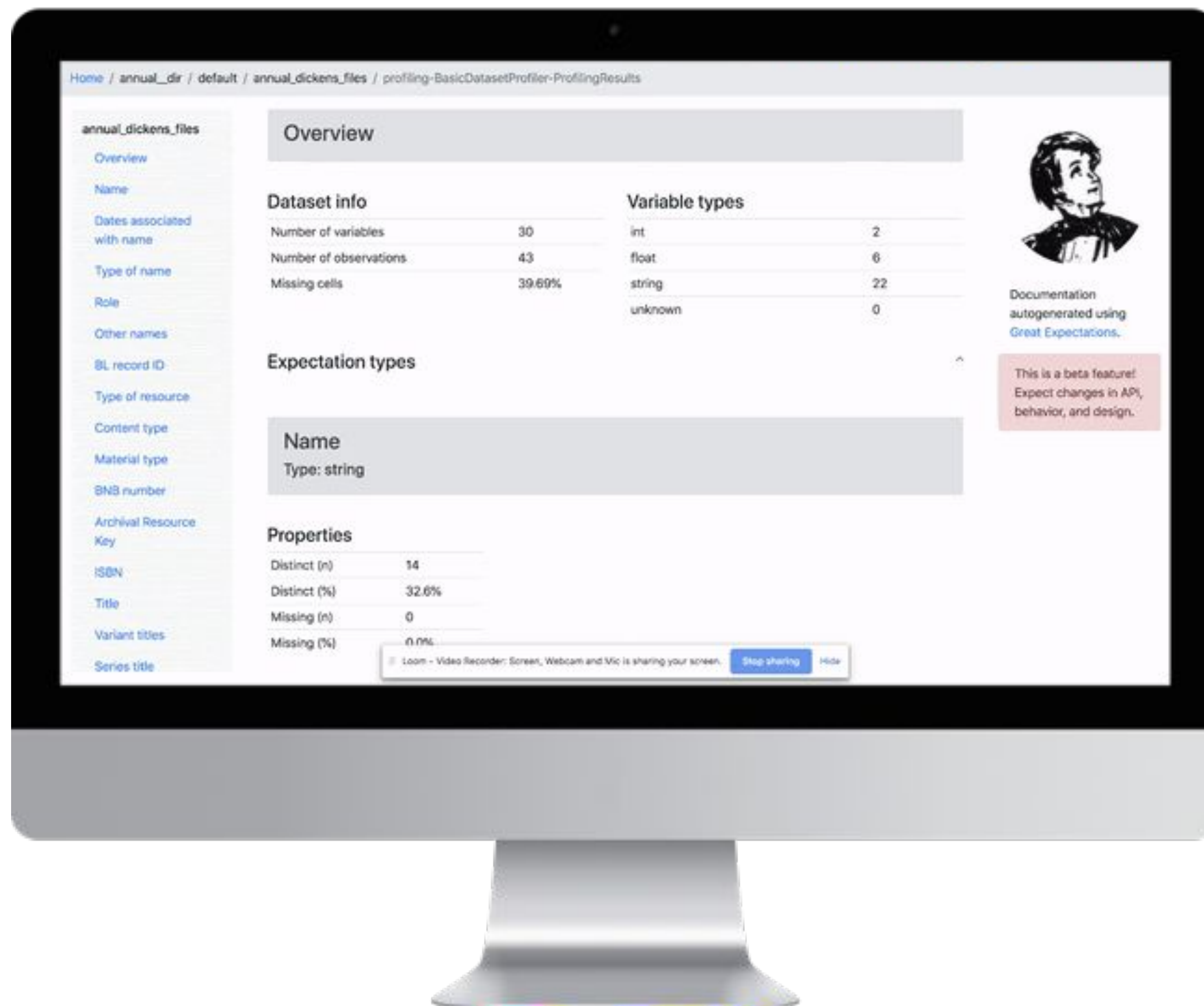
При желании можно сохранять в PostgreSQL

```
{
  "data_asset_type": "Dataset",
  "expectation_suite_name": "taxi.demo",
  "expectations": [
    {
      "expectation_type": "expect_column_mean_to_be_between",
      "kwargs": {
        "column": "passenger_count",
        "max_value": 1.1,
        "min_value": 1.4
      },
      "meta": {}
    }
  ],
  "meta": {
    "great_expectations_version": "0.13.19"
  }
}
```

02 | great_expectations DataDocs

Отображает результаты запуска проверок

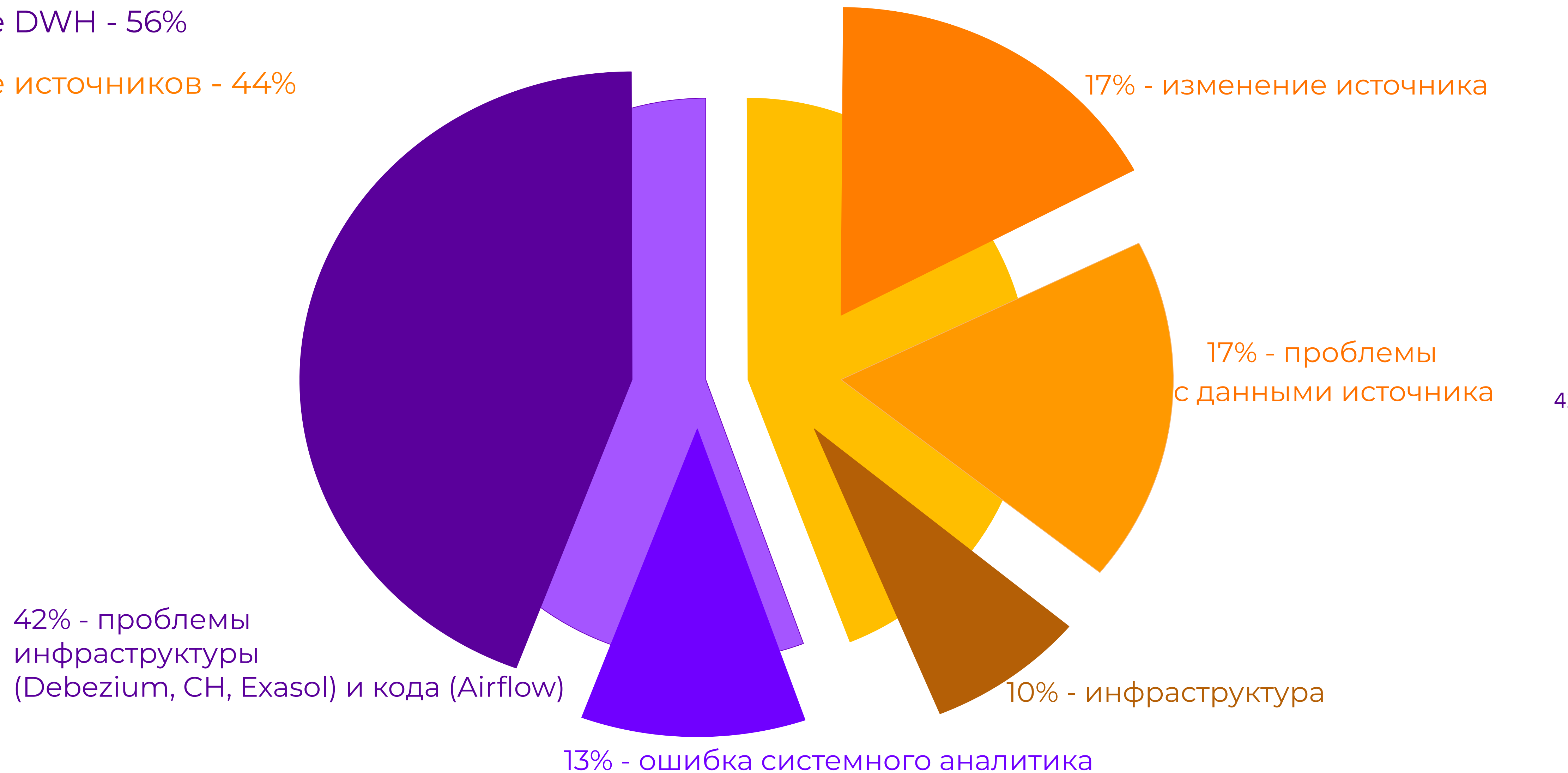
- ▶ Это просто автогенерируемый HTML
- ▶ Превращается в сайт скамливанием папки Nginx
- ▶ Можно кастомизировать, редактируя шаблон



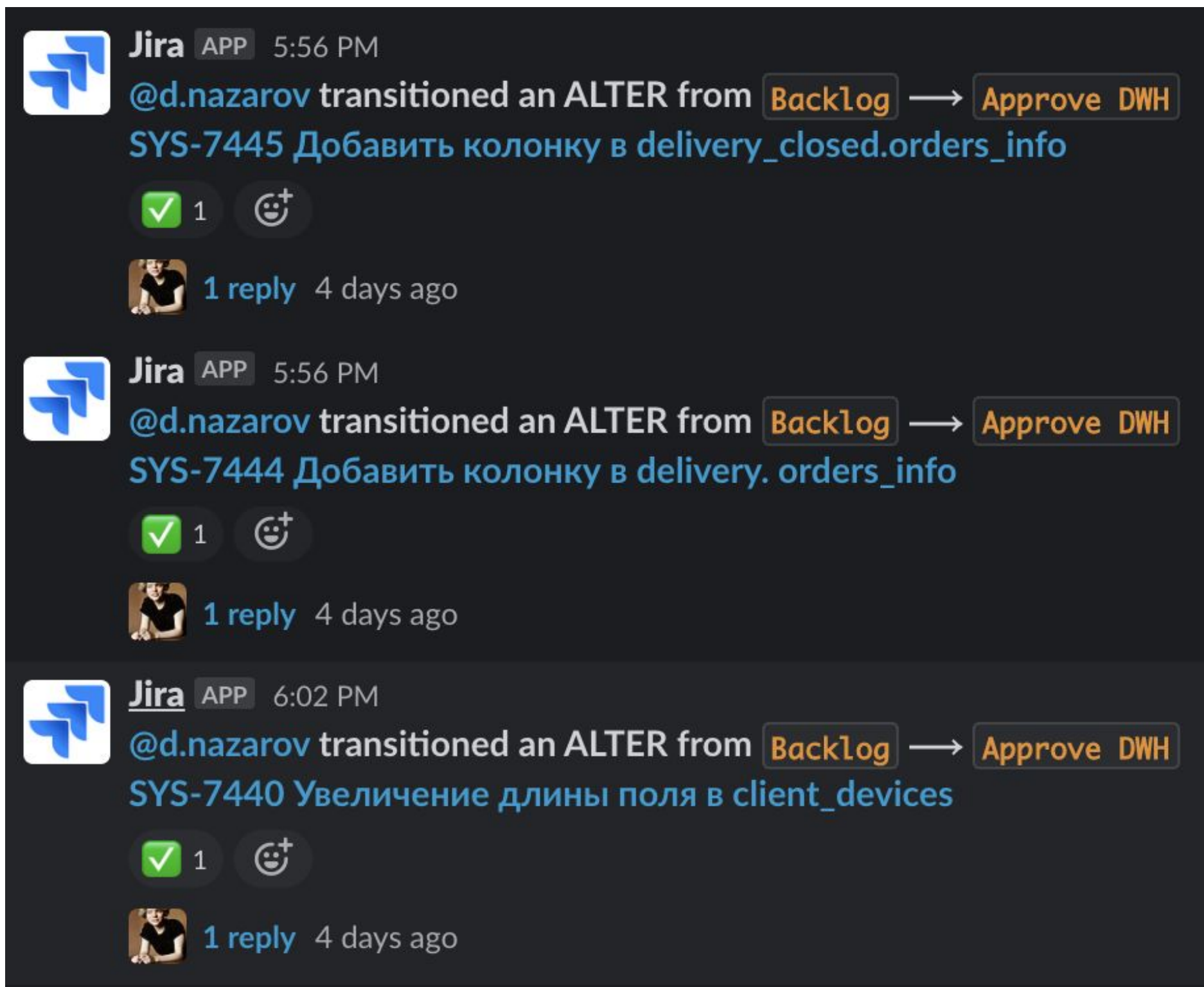
02 | Анализ инцидентов

На стороне DWH - 56%

На стороне источников - 44%

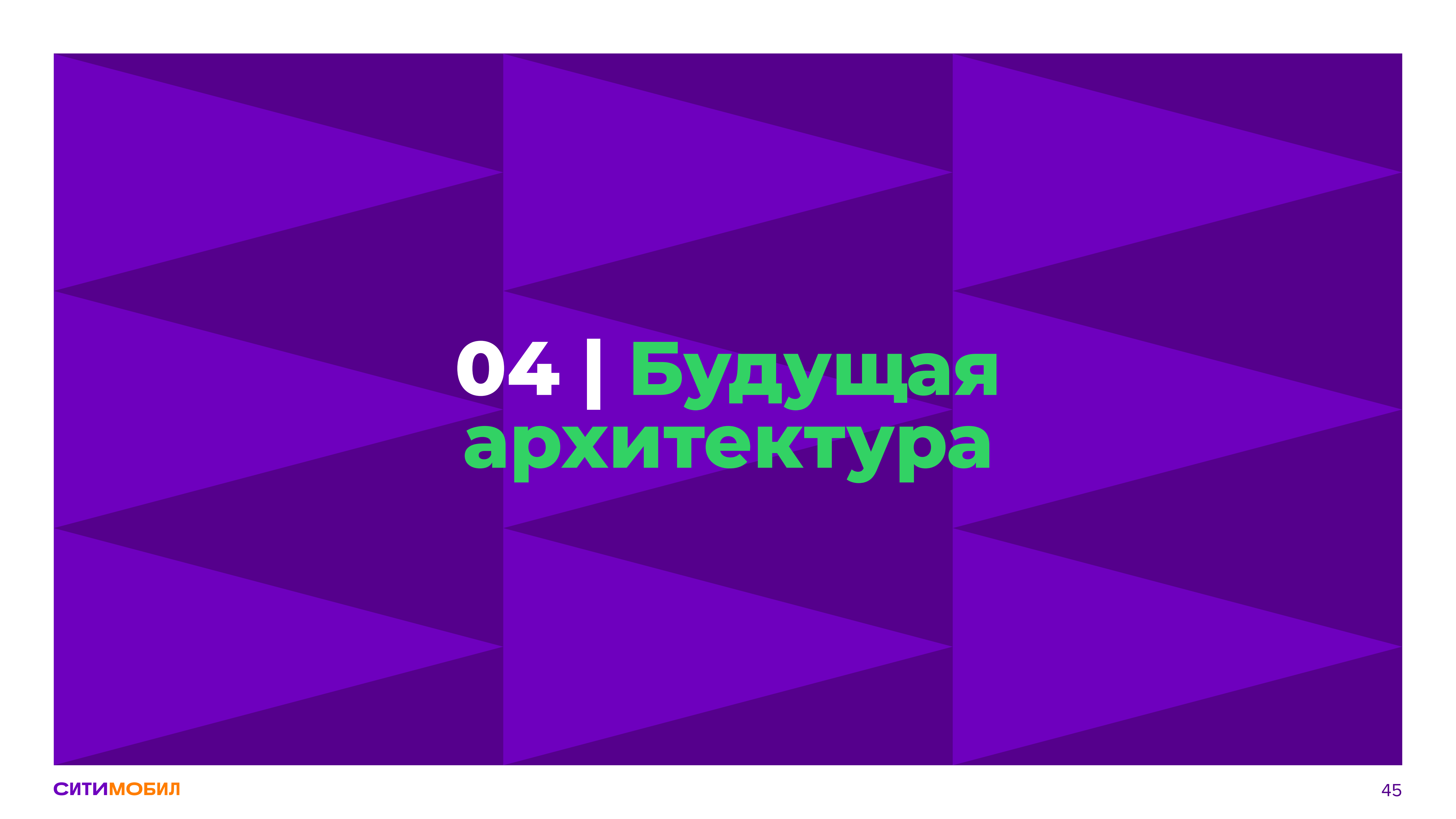


Проблемы источника. Изменение на источнике TOTAL CONTROL



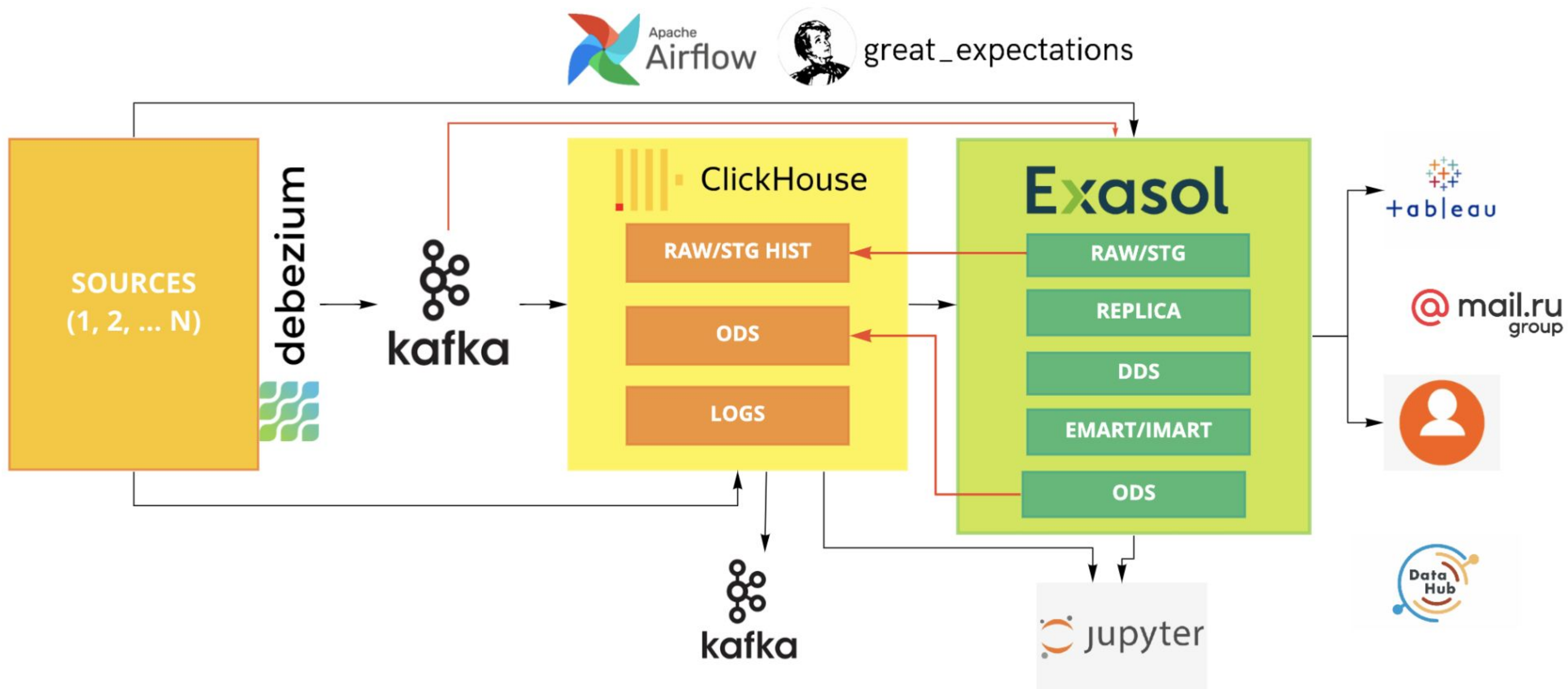
Ошибки на стороне DWH
Жизнеутверждающий мем напоследок





04 | Будущая архитектура

Планируемая архитектура DWH Технологии



Дальнейшие планы на 2022

Резюмируя:

- Автоматизация разработки и тестирования. Автоматизировать все, что можно автоматизировать.
- Обложить мониторингами все, что можно обложить мониторингами
- Развитие DQ
- Полноценное внедрение Data Catalog (с Data Lineage)
- Онлайновый слой
- Переход к парадигме Data Governance внутри компании

**Move STG
to EXASOL**

Data Catalog

как второй шаг к Data
Governance



**Автоматизируй
это**

Self Service DQ

Дать пользователям
возможность управлять
качеством данных

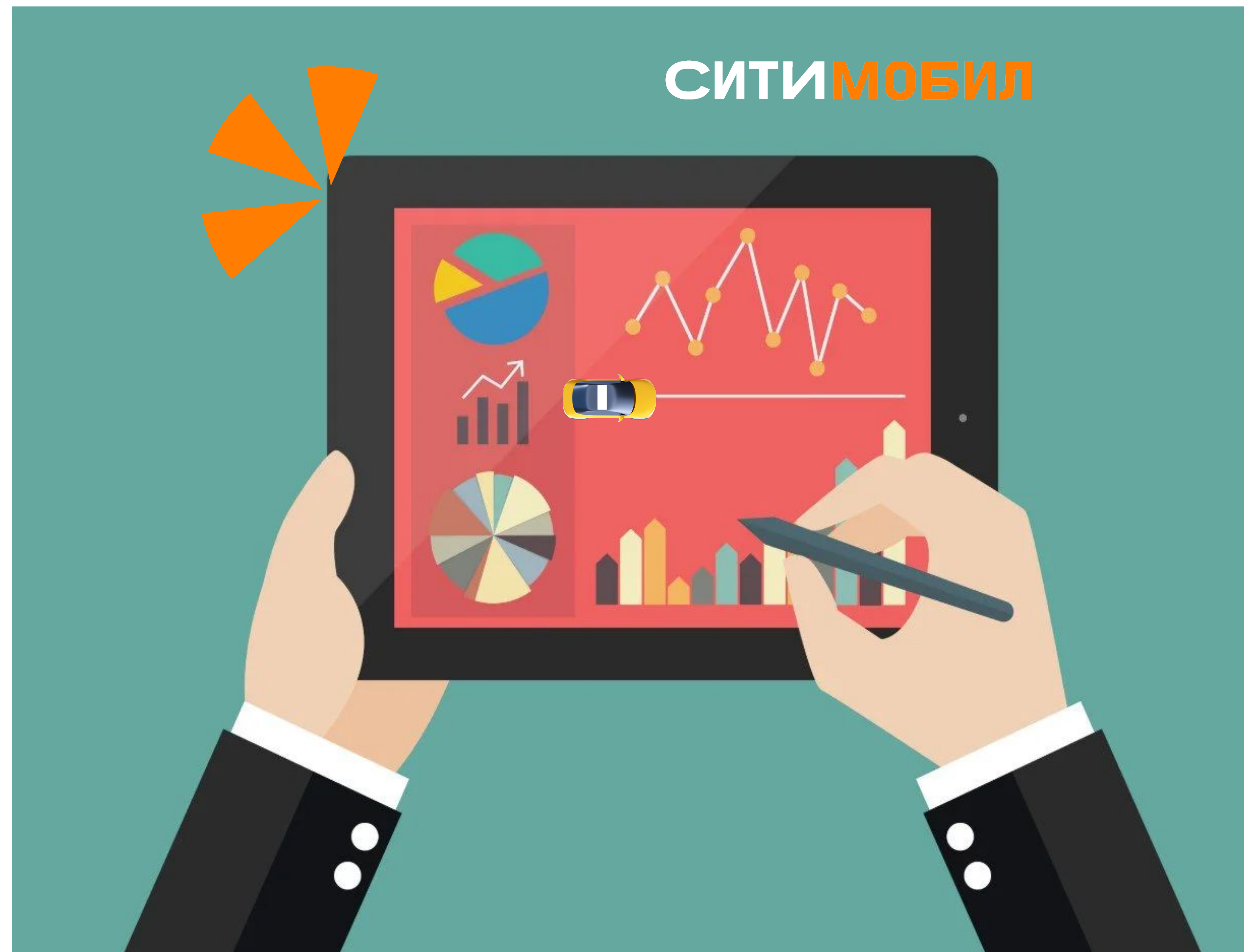
05 | Конечная польза

Смысл DWH это бизнес- метрики

**Хранилище данных - это система
поддержки принятия решений.**

Соответственно, от качества хранилища
зависит качество принимаемых
решений.

**С этой точки зрения качество и
оперативность поставляемых данных
играют очень большую роль.**



Метрики

Конечно же, мы считаем поездки, нашу аудиторию, GMV, и прочее.

Однако, наибольший интерес представляют другие метрики.

Например, у нас очень сложная методология определения **просмотра**.

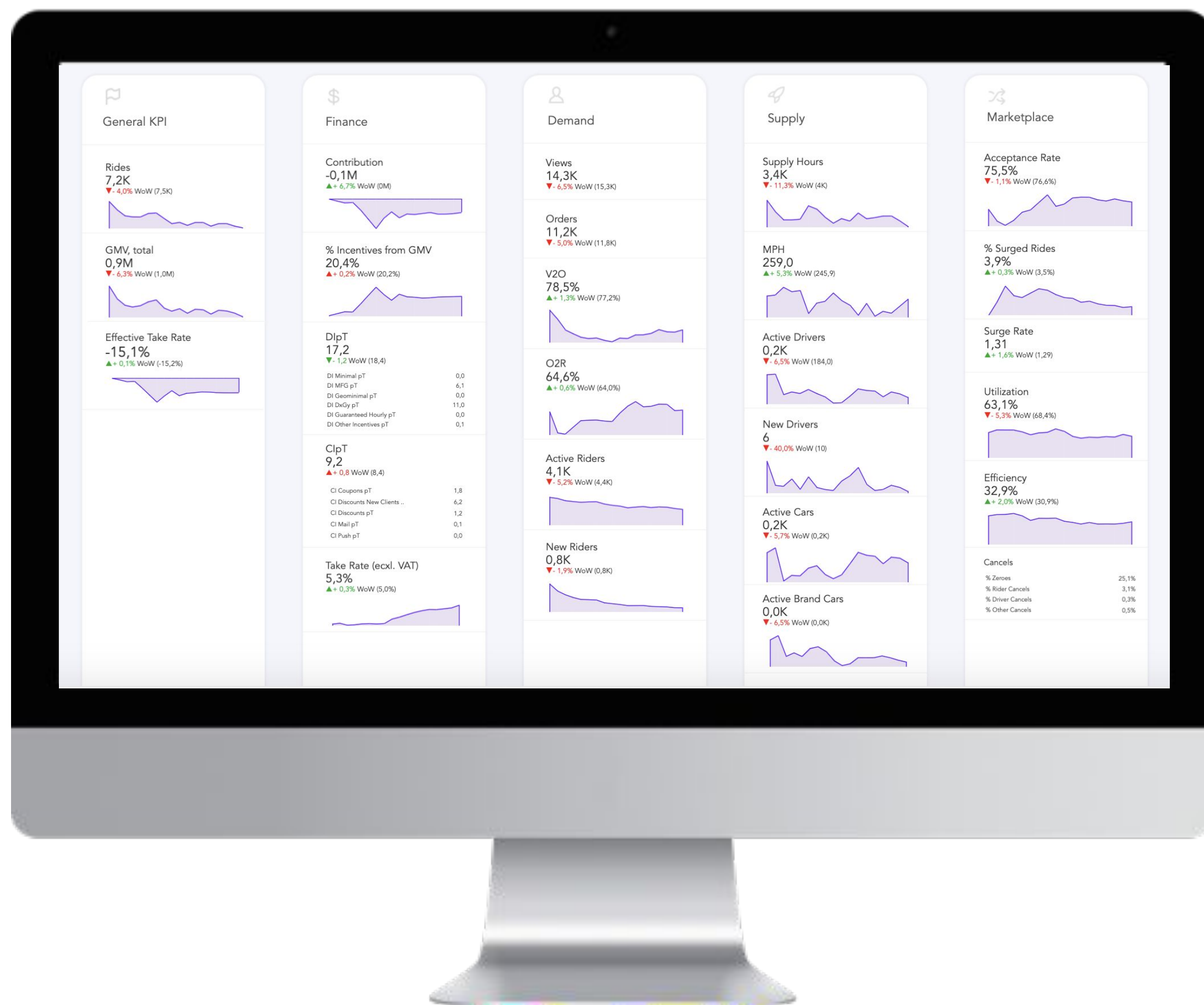
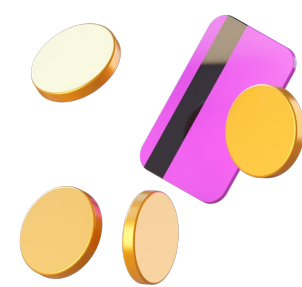
Во времена “до Экзасола” витрина считалась час и имела сомнительное качество.

Сейчас витрина обновляется **каждый час и рассчитывается за 1-2 минуты**.

Аналогичная история с расчетом CI, DI и комиссий.



Supply&Demand Отчет в Tableau



Отчет отражает динамику ключевых метрик маркетплейса

Визуализации позволяют отследить резкие изменения спроса и предложения, а также основных финансовых показателей. На основании этой информации делаются выводы о развитии бизнеса

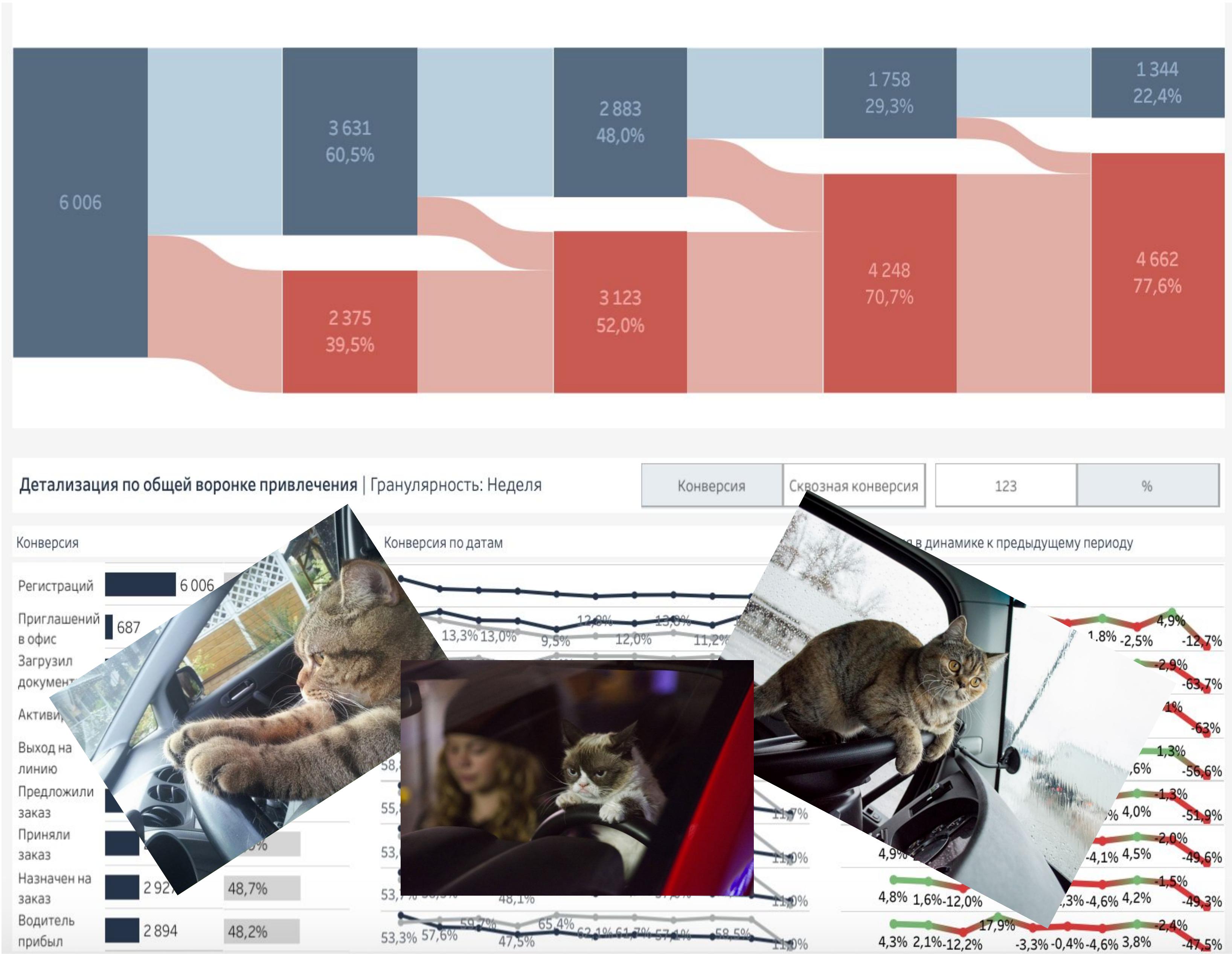
- ▶ Кол-во просмотров
- ▶ Кол-во заказов
- ▶ Исполняемость (O2R)
- ▶ Кол-во активных водителей и пассажиров
- ▶ GMV
- ▶ Surge Rate
- ▶ и другие метрики, которые позволяют управлять маркетплейсом



Воронка водителей

Основной инструмент по работе с привлечением водителей.

Отчет отражает сколько водителей мы теряем на каждом этапе воронки, позволяя выделить проблемные этапы привлечения. Изменение показателей в динамике позволяет делать выводы об эффективности промо-кампаний.



СИТИМОБИЛ

**Спасибо за
внимание!**

Вопросы?

