OZONtech

Как сделать каталог данных быстро

В условиях нескольких зрелых хранилищ

Наталья Журавлёва
Руководитель группы поддержки DWH

Обо мне

- Работаю в Ozon **больше 5 лет**
- Являюсь **лидом команды аналитиков в отделе DG**
- Застала переход **от MsSQL к колоночной БД, от DataLake — к Anchor DWH**
- Работала **по «обе стороны» данных**

Тезисы

На Data Governance
нет времени

01

Логическая модель —
фундамент каталога данных

02

Нужно думать о **внедрении**
нового подхода разработки
хранилища

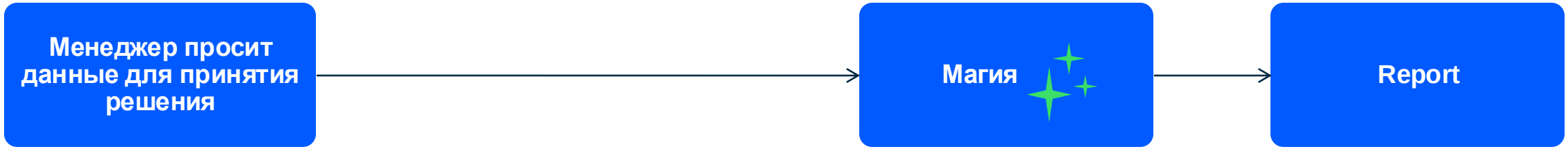
03

Что можно показать
в каталоге данных, помимо
описания данных

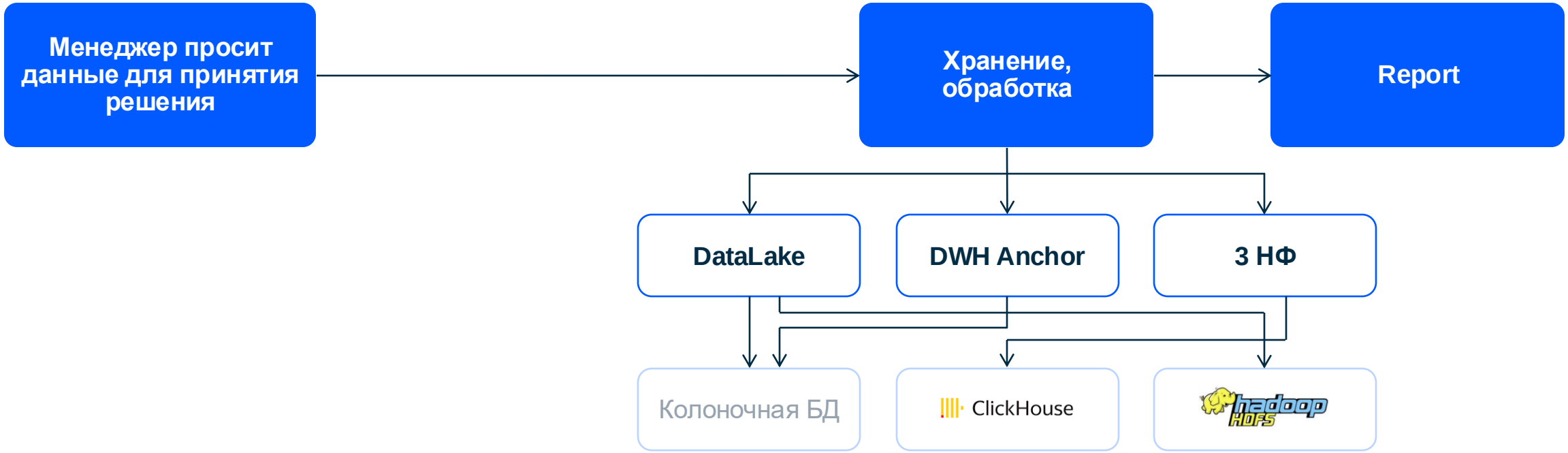
04

Ресурсов на Data Catalog нет

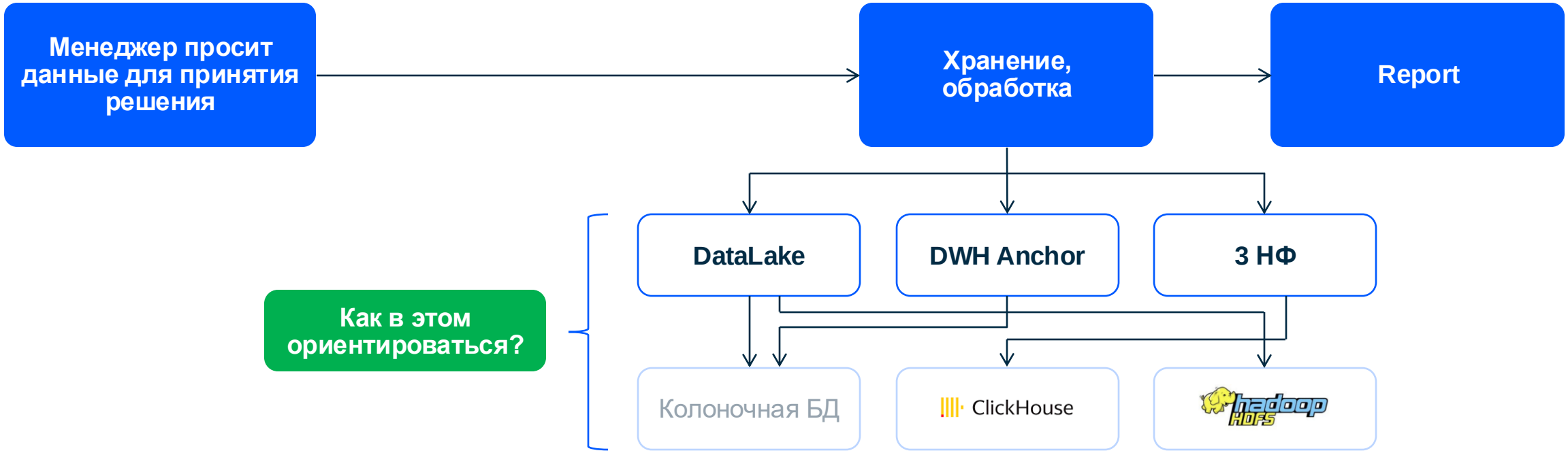
Как можно быстрее



Ресурсов на Data Catalog нет

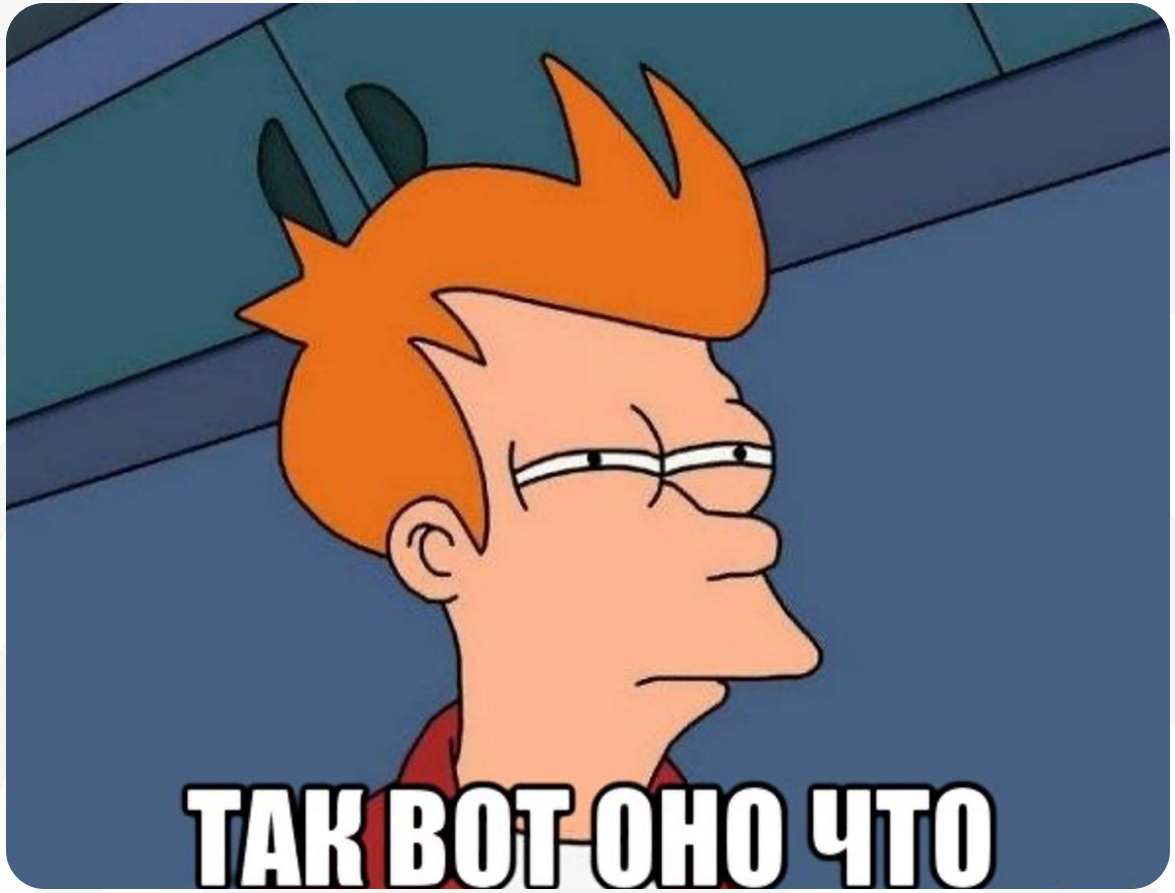


Ресурсов на Data Catalog нет

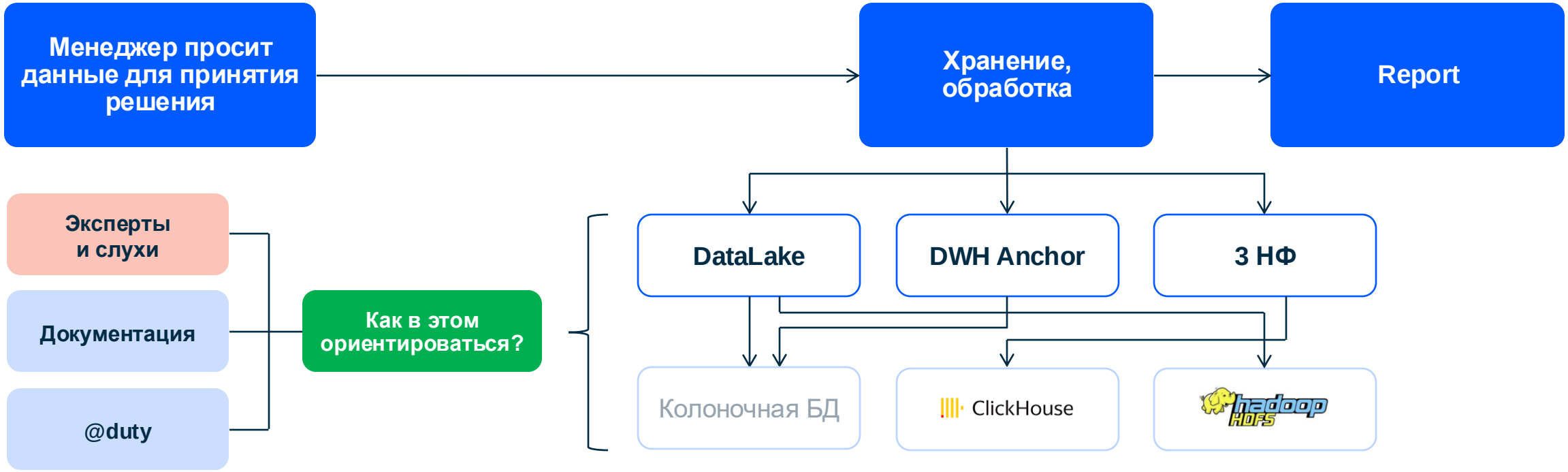


Прошло N времени

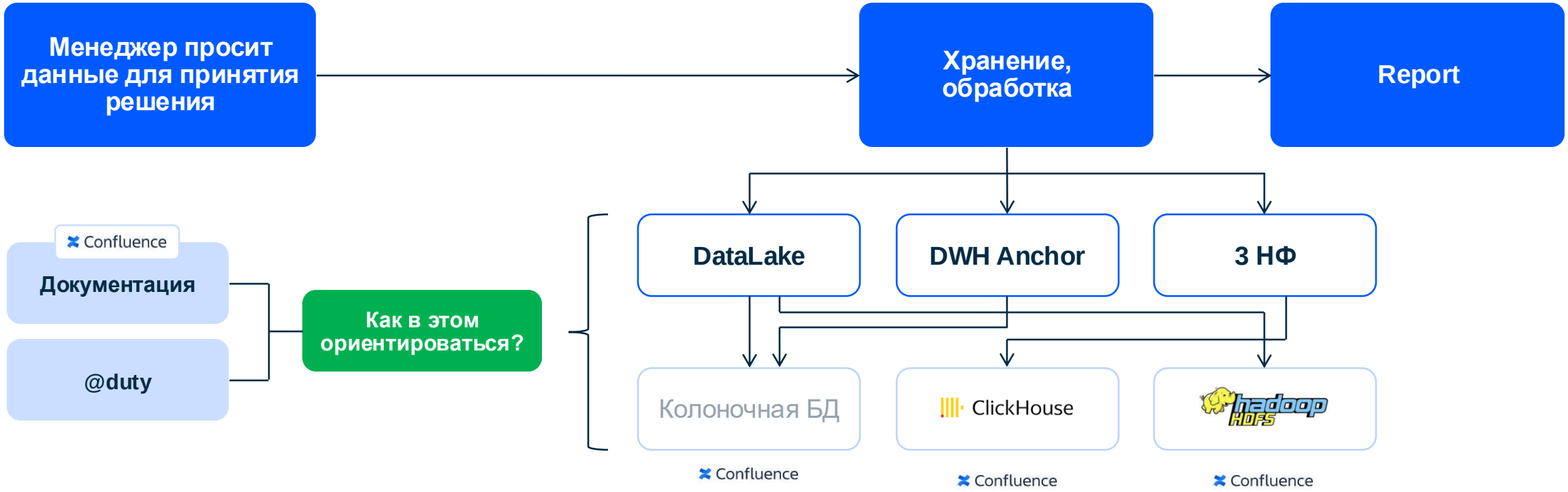
- **Данные в отчётах разные**
- **Завалили вопросами**
- **Сложность в разработке нового контента**



Очень упрощённый flow данных



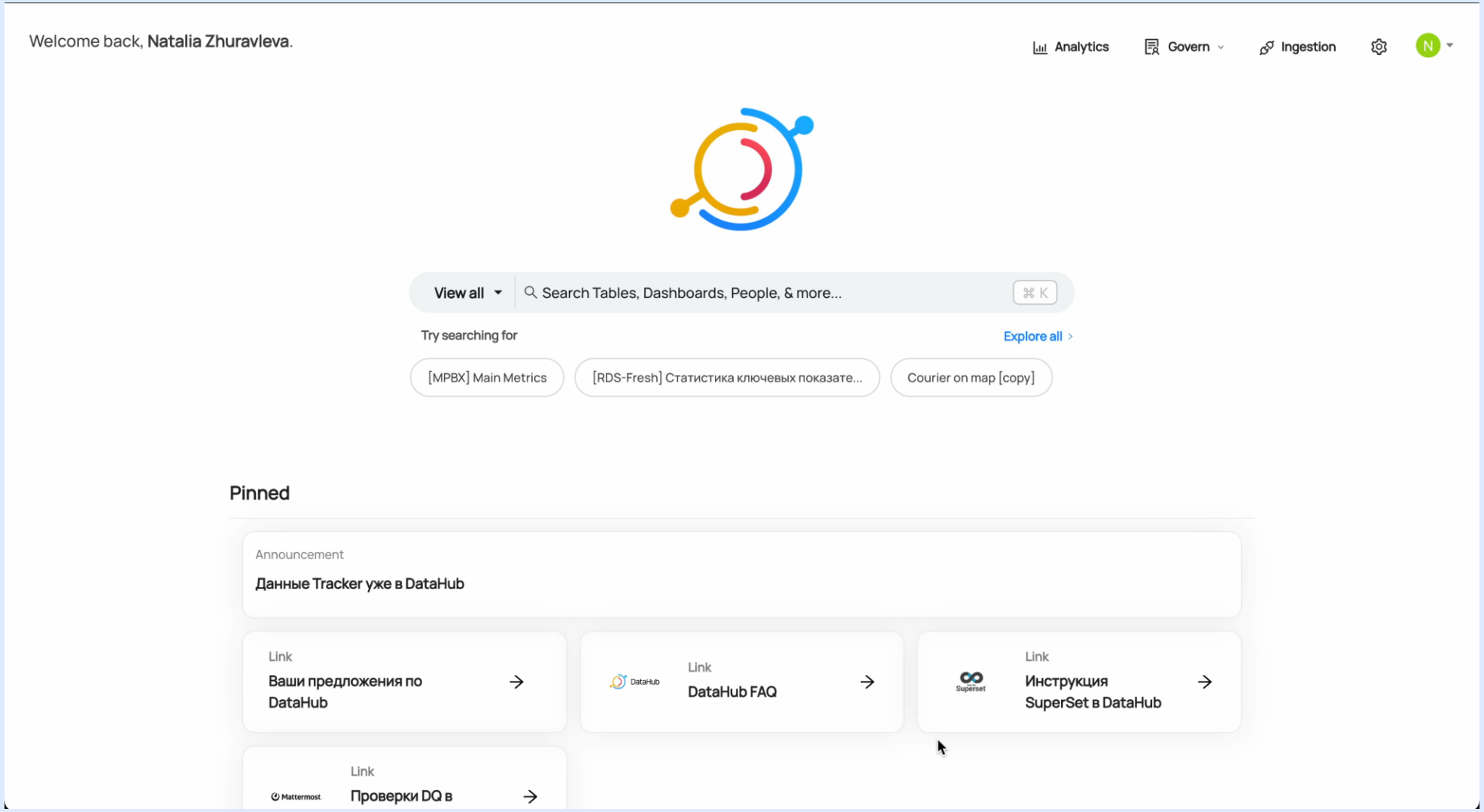
Очень упрощённый flow данных



Почему нужно идти дальше

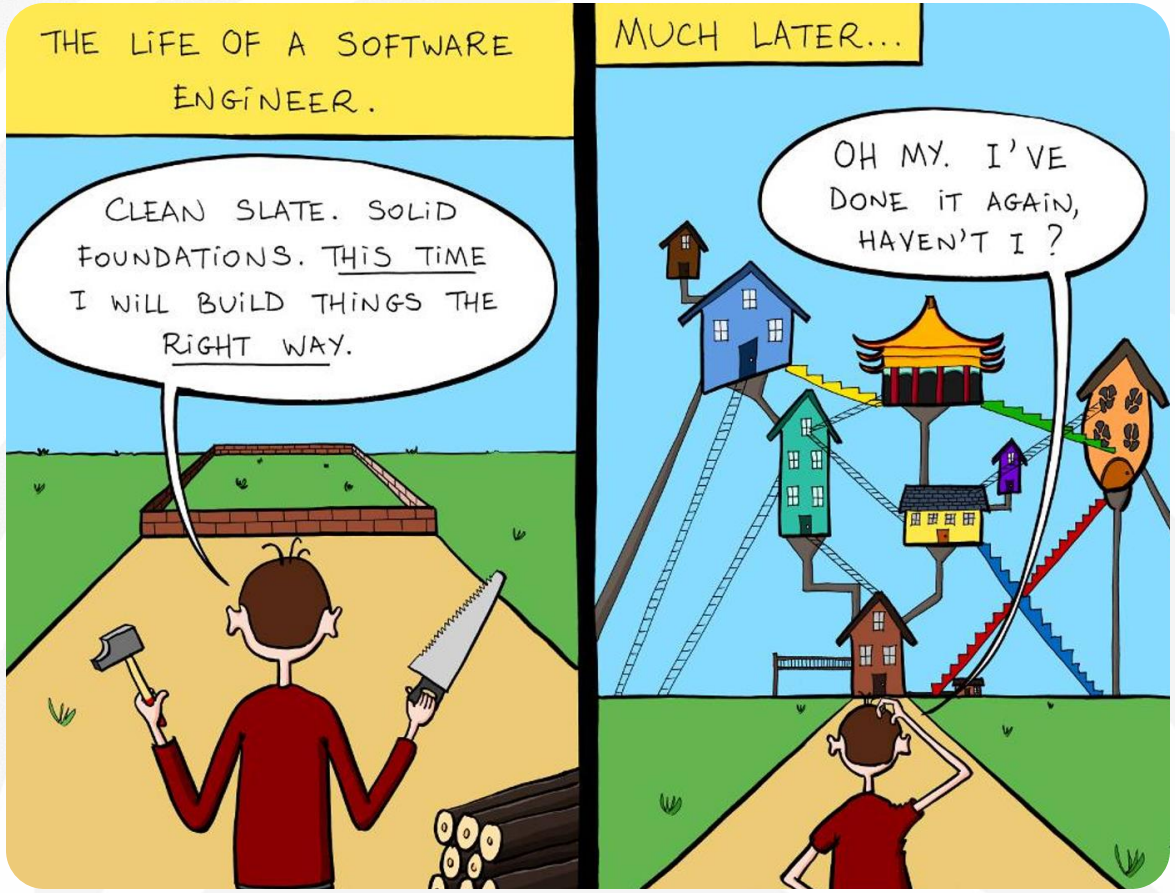
- Сложно поддерживать
- Недостаточно информации
- Есть база знаний на будущее

К чему пришли за полгода



Скажи **«нет»**

- **Превращение каталога данных в болото**
- **Ручное ведение каталога**
- **Не интегрирован в процесс разработки хранилища**



Каталог без модели = деньги на ветер

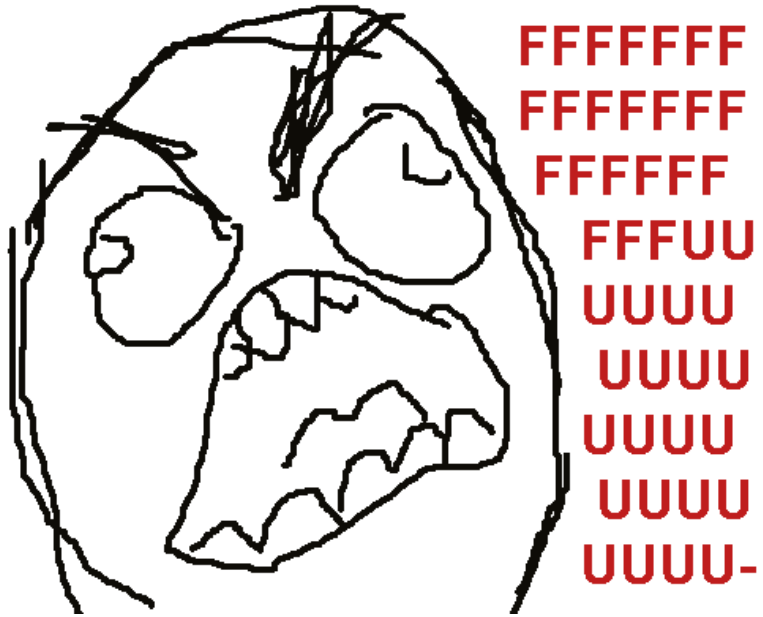
Каталог данных
без модели данных

Каталог данных
с моделью данных

С точки зрения пользователя

Ввожу в поиск слово «заказ»

< 1 2 3 ••• 15 > 10 / page

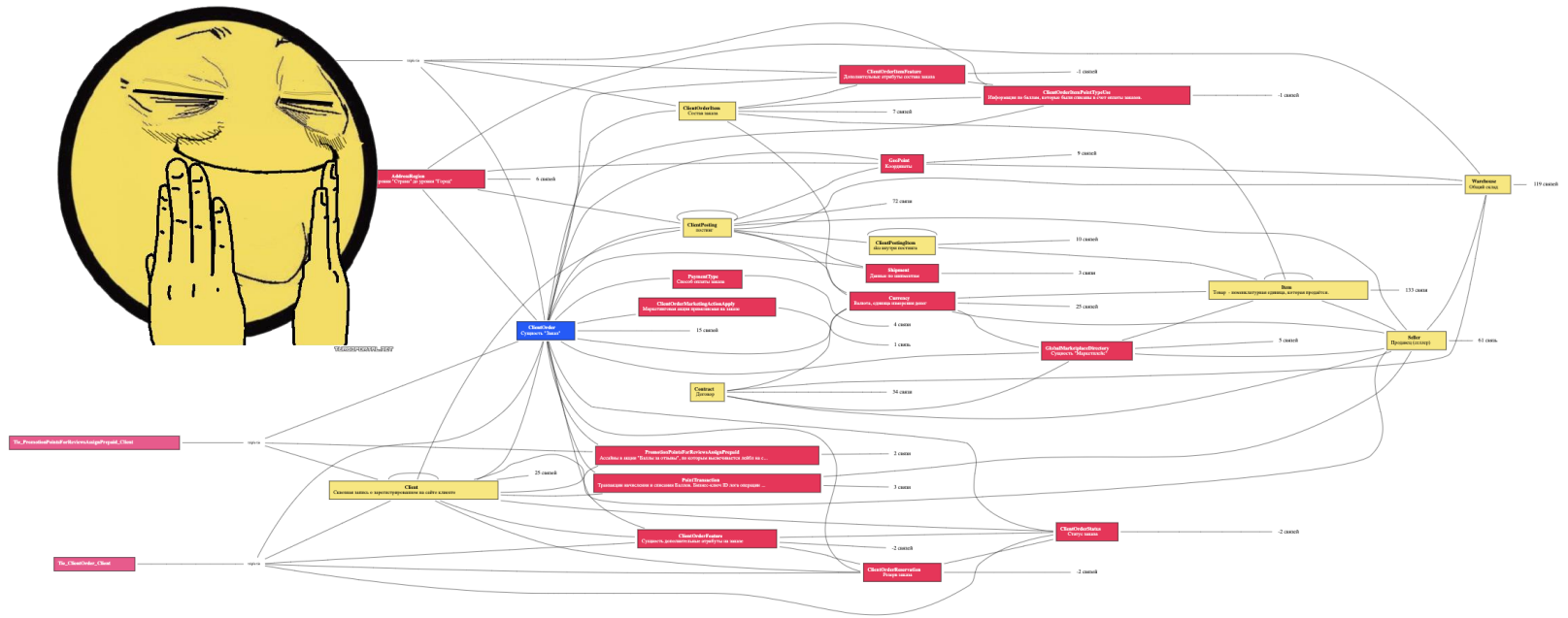


С точки зрения пользователя

Поиск «заказ» среди логического представления данных

Содержание

- Граф модели
- Готовые запросы
- Логические сущности
- Атрибуты
- Связи



Как по книжке

	Концептуальная модель	Логическая модель	Физическая модель
Названия	+	+	
Связи	+	+	
Атрибутивный состав		+	
Первичный ключ		+	+
Вторичный ключ		+	+
Имя таблицы			+
Имя столбца			+
Тип данных столбца			+

Наш подход

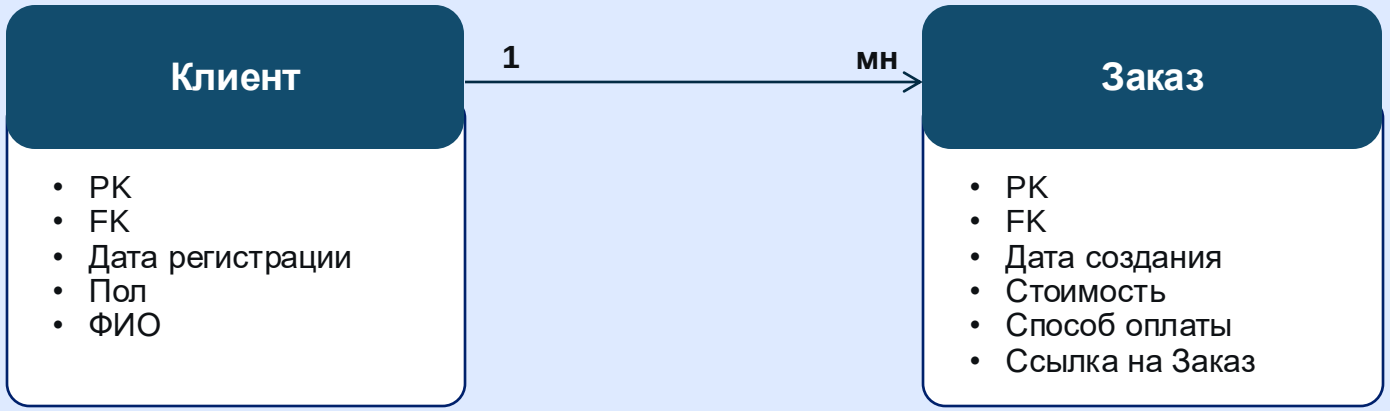
	Концептуальная модель	Логическая модель	Физическая модель
Названия	+	+	
Связи	+	+	
Атрибутивный состав		+	
Первичный ключ		+	+
Вторичный ключ		+	+
Имя таблицы			+
Имя столбца			+
Тип данных столбца			+

Наш подход

	Концептуальная модель	Логическая модель	Физическая модель
Названия	+	+	
Связи	+	+	
Атрибутивный состав		+	
Первичный ключ		+	+
Вторичный ключ		+	+
Имя таблицы			+
Имя столбца			+
Тип данных столбца			+

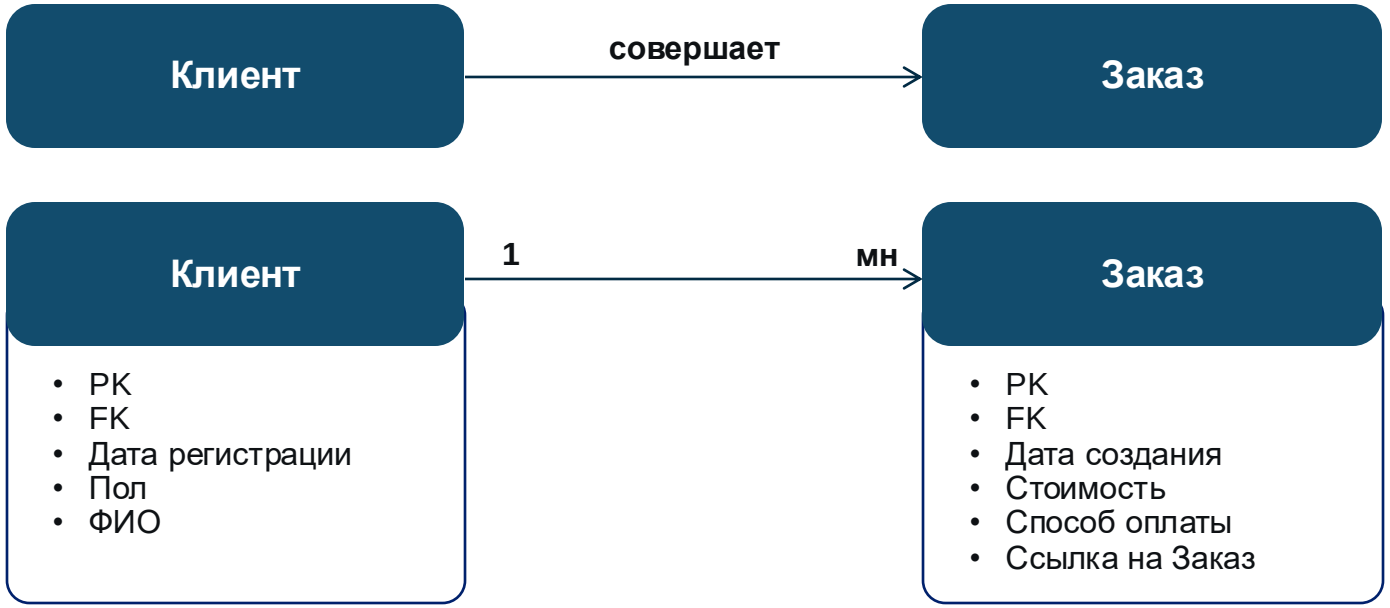
Разница модулей на примере

Разница модулей на примере



Логический уровень

Разница модулей на примере



dwh_data.Hub_Client

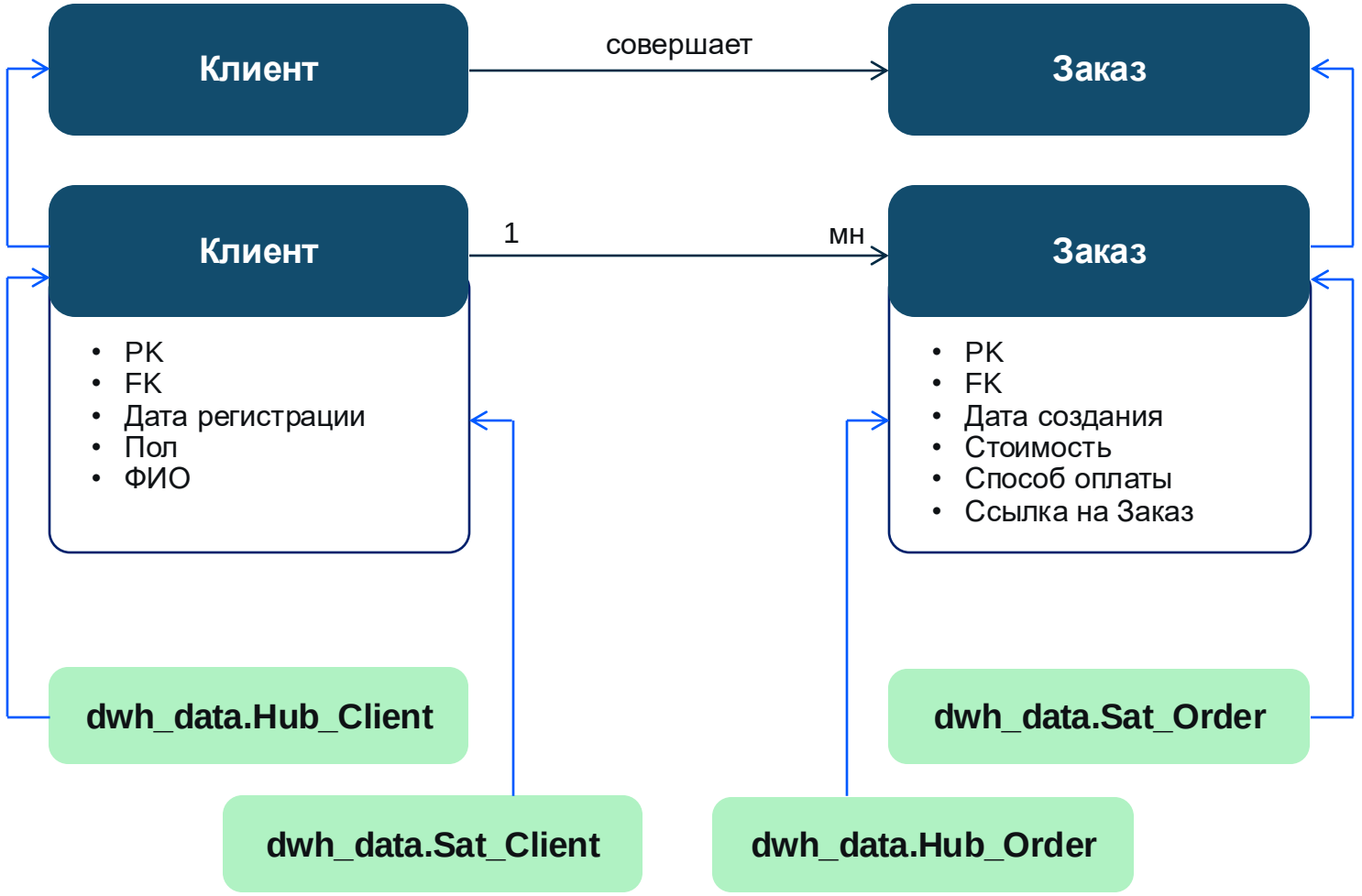
dwh_data.Sat_Order

dwh_data.Sat_Client

dwh_data.Hub_Order

Физический уровень

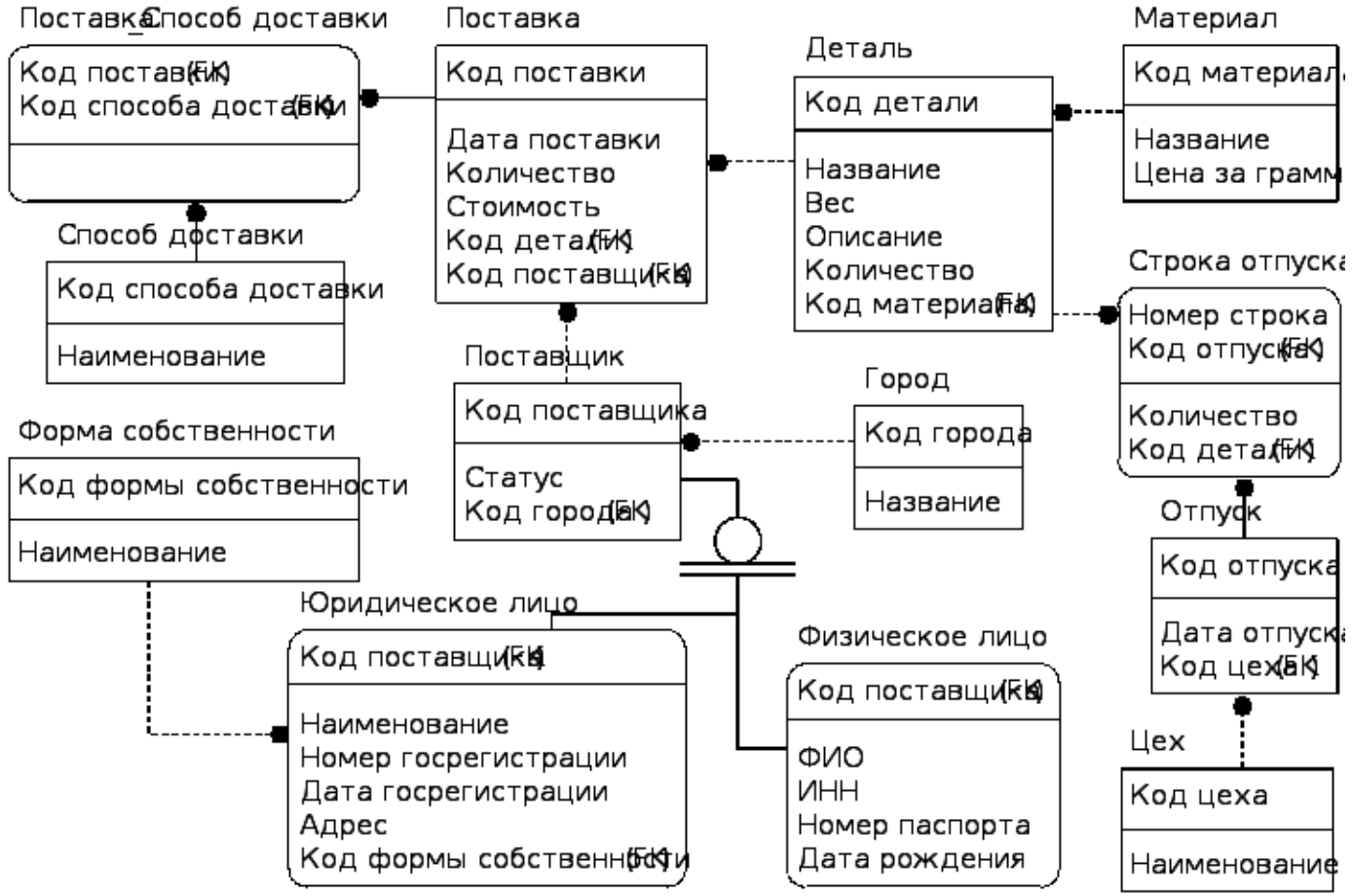
Связь между уровнями



Неправильный, но жизненный подход

В чём сложность? ⚡

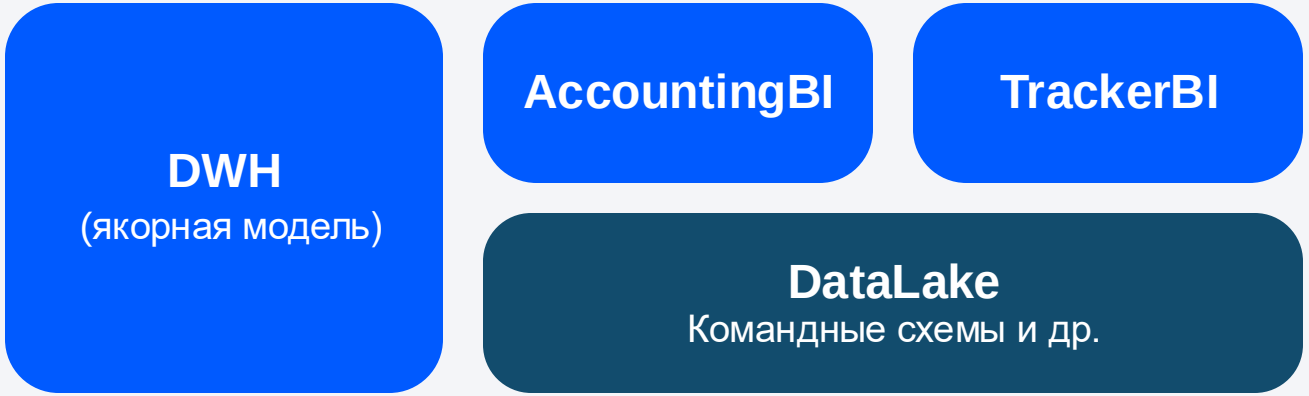
- **Много времени**
- **Экспертные кадры**
- **Творческий процесс**
- **Как её хранить?**
- **Как её изменять?**



Как мы сделали модель

На входе:

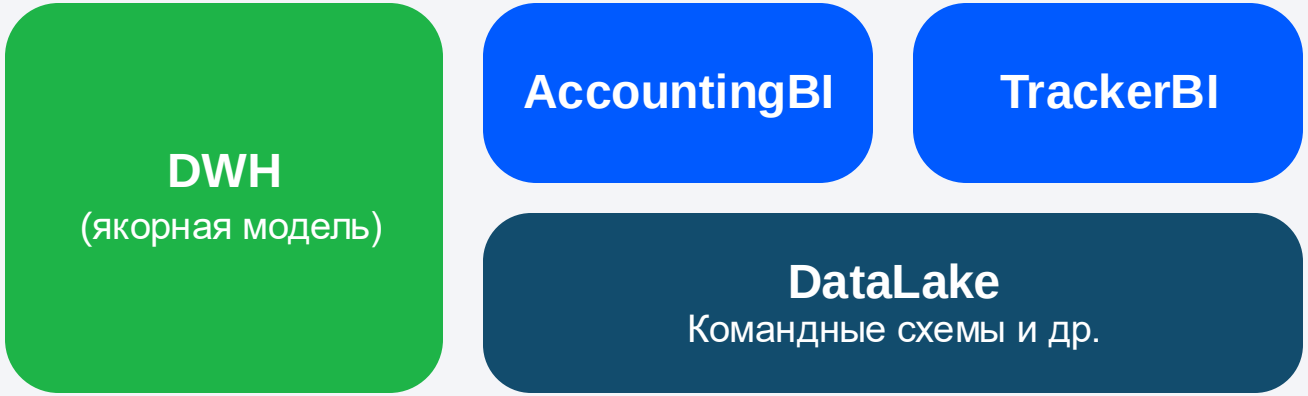
- **Три** зрелых хранилища
- **Петабайты** данных
- Несколько баз данных
- **750** якорей, **2500** атрибутов, **1000+** связей и др.



Начали с DWH

Распарсили названия таблиц в детальном слое

- **Якоря = сущности**
- **Атрибуты = атрибуты сущностей**
- **Связи = ссылки на другие сущности**



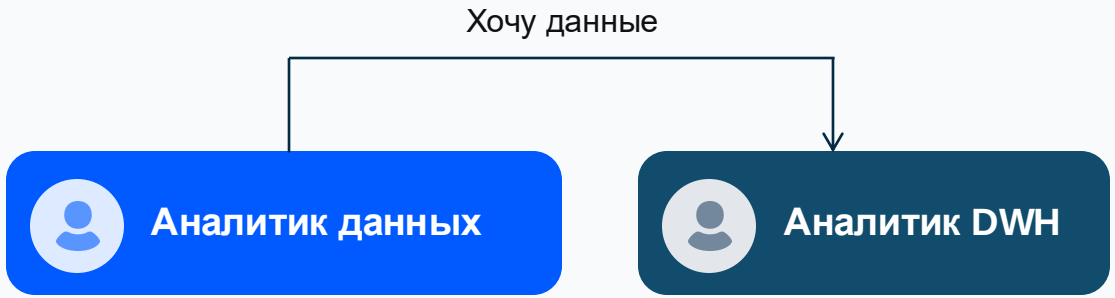
Получили модель

```
AbVariant:
  Description: вариант A/B теста
  Concept: AB
  Attributes:
    Name:
      Description: человеко-читаемое название варианта A/B теста
    IsControl:
      Description: признак того контрольной группы
    AbVariantId:
      Description:  ключ AbVariant
      PK: true
```

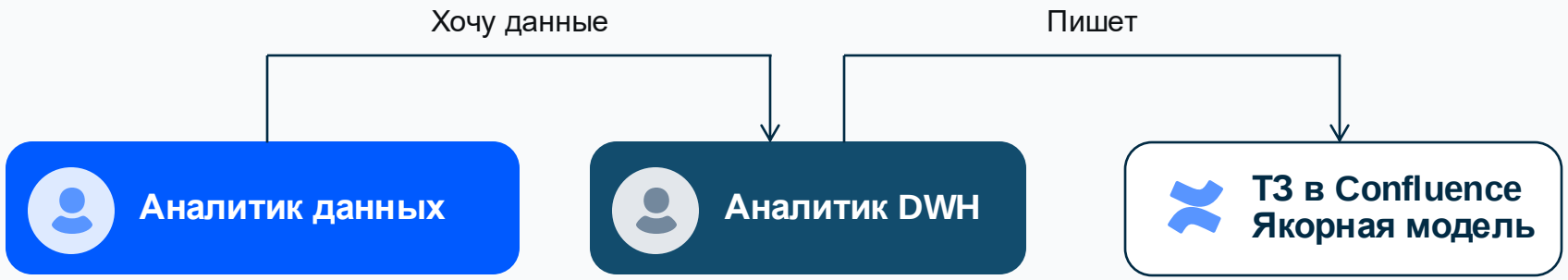
Место хранения — **yaml-файл**

Изменение yaml **через MR**

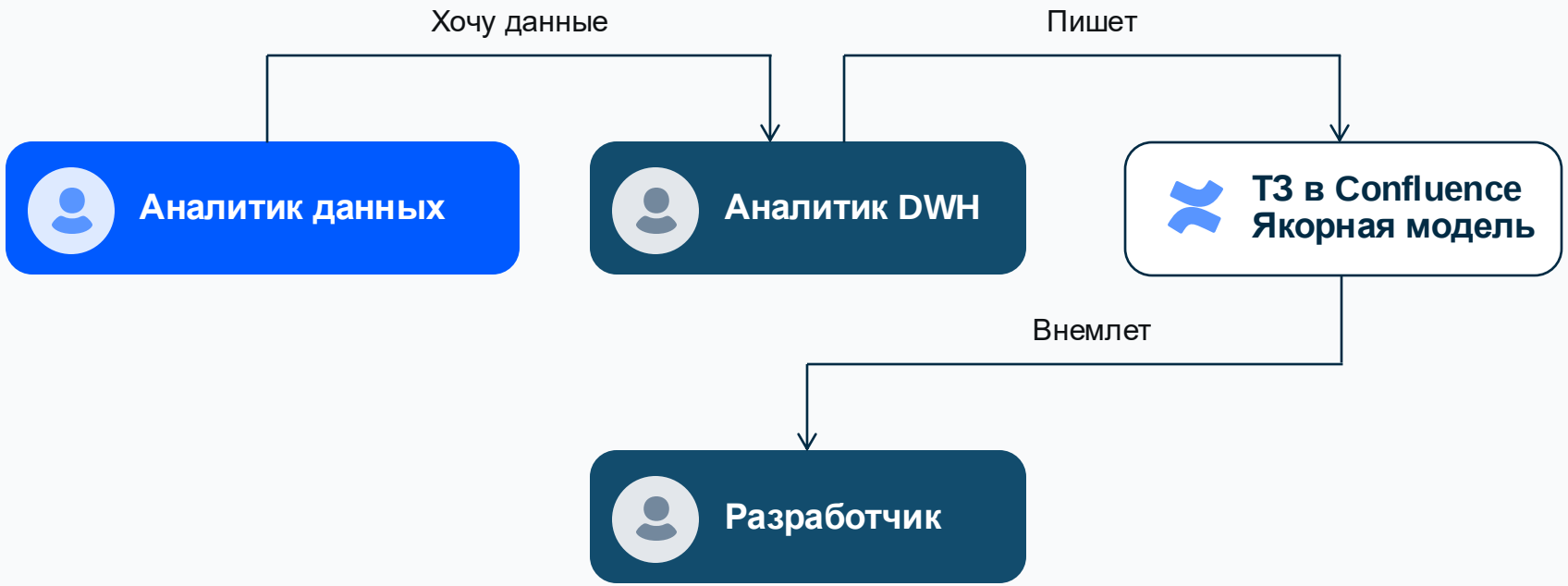
Как было **«до»**



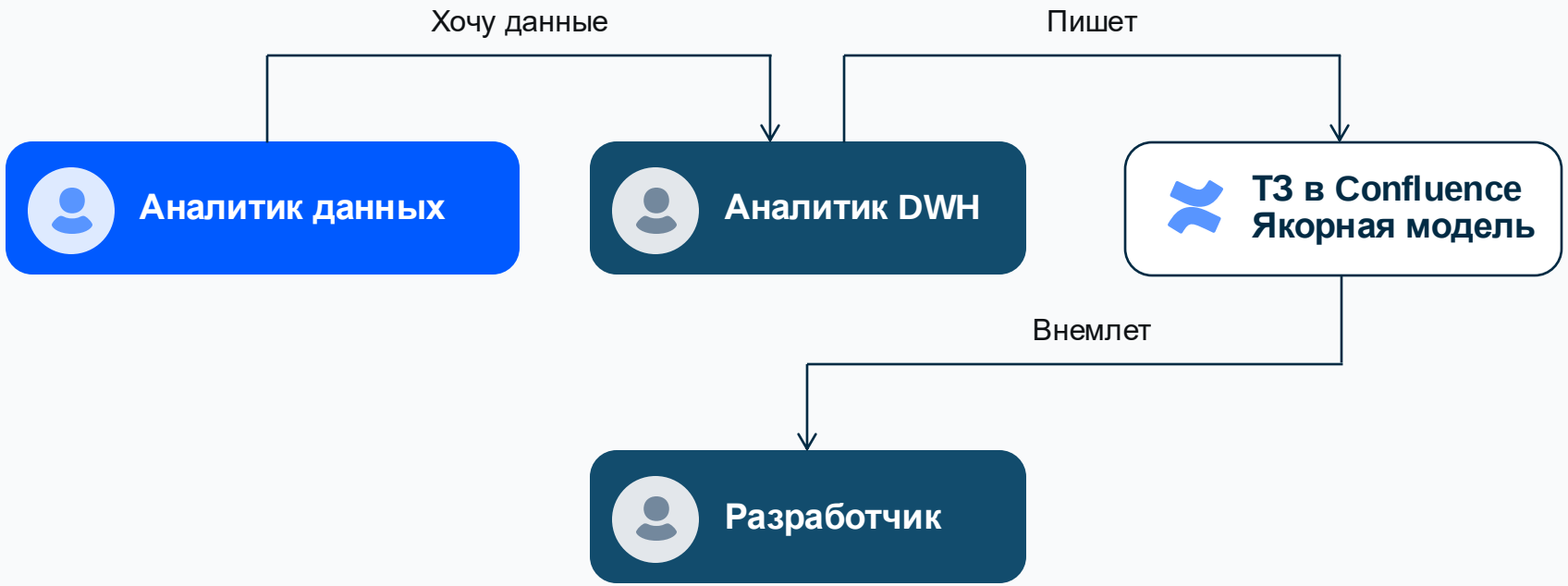
Как было **«до»**


Хочу данные
Пишет
Аналитик данных
Аналитик DWH
ТЗ в Confluence
Якорная модель

Как было **«до»**


Хочу данные
Пишет
Аналитик данных
Аналитик DWH
Т3 в Confluence
Якорная модель
Внемлет
Разработчик

Как было **«до»**

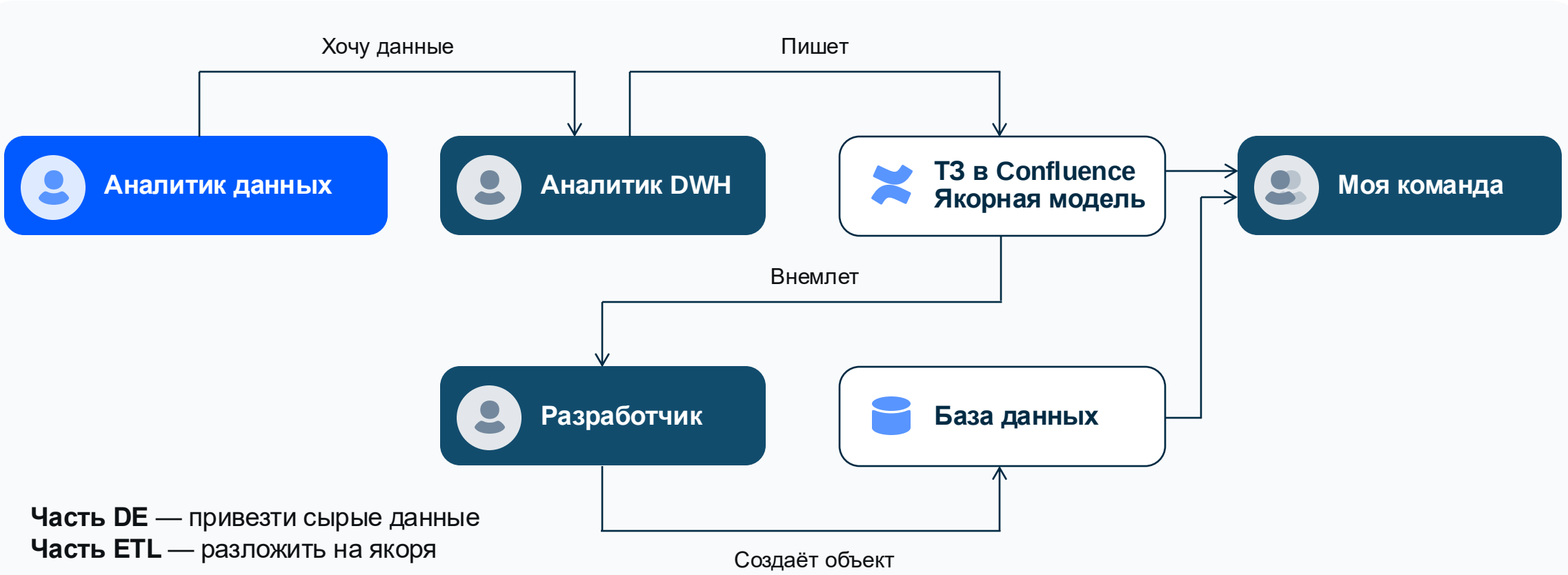


Часть DE — привезти сырые данные
Часть ETL — разложить на якоря

Как было **«до»**

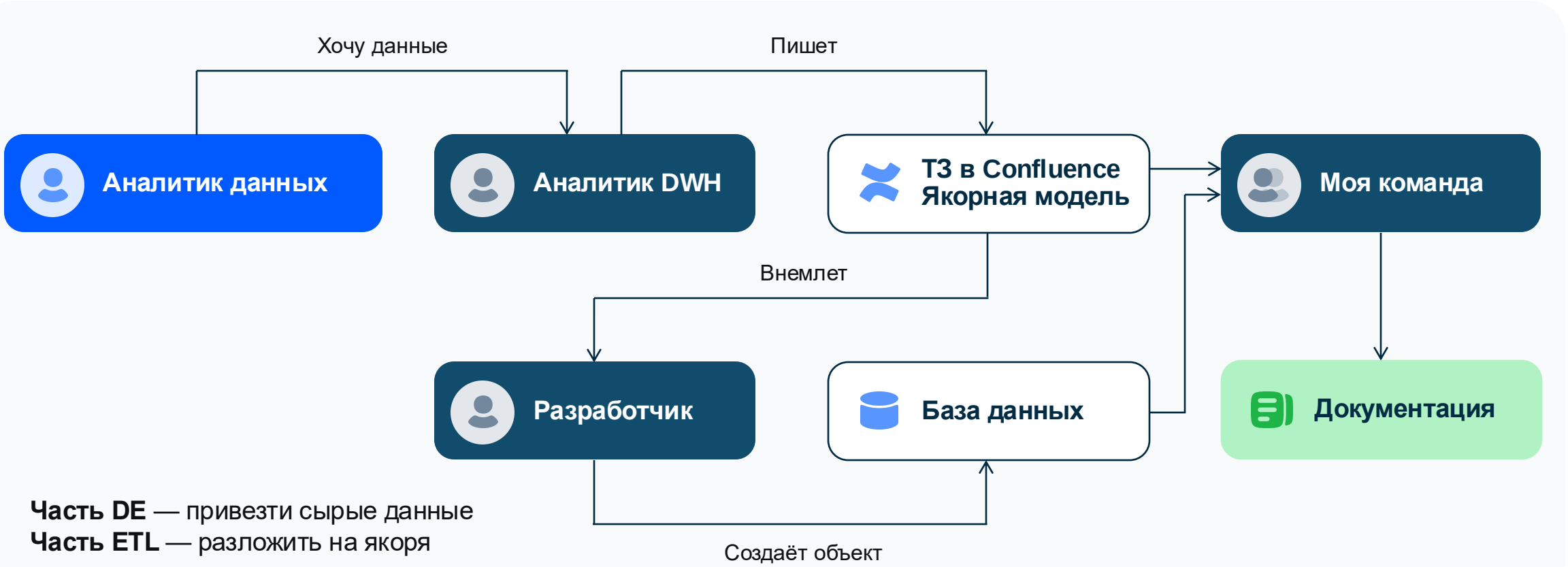
Часть DE — привезти сырые данные
Часть ETL — разложить на якоря

Как было **«до»**



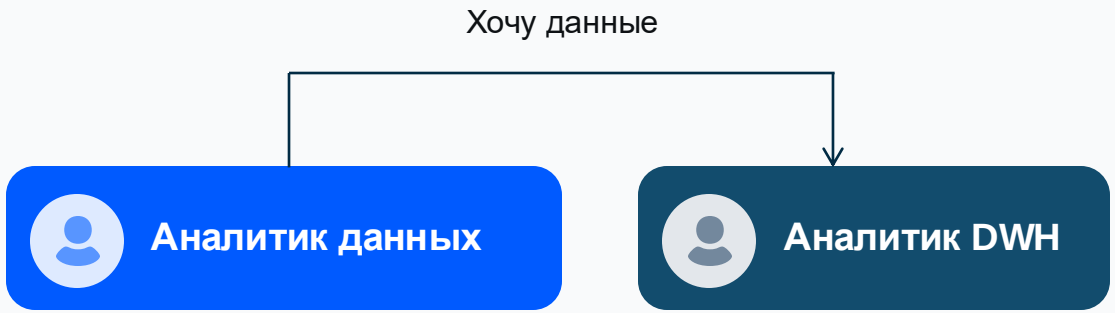
Часть DE — привезти сырые данные
Часть ETL — разложить на якоря

Как было **«до»**

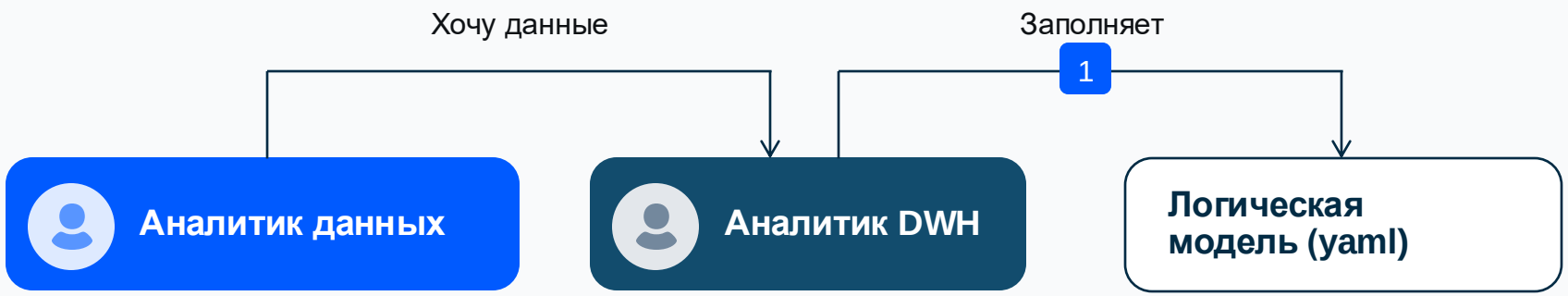


Часть DE — привезти сырые данные
Часть ETL — разложить на якоря

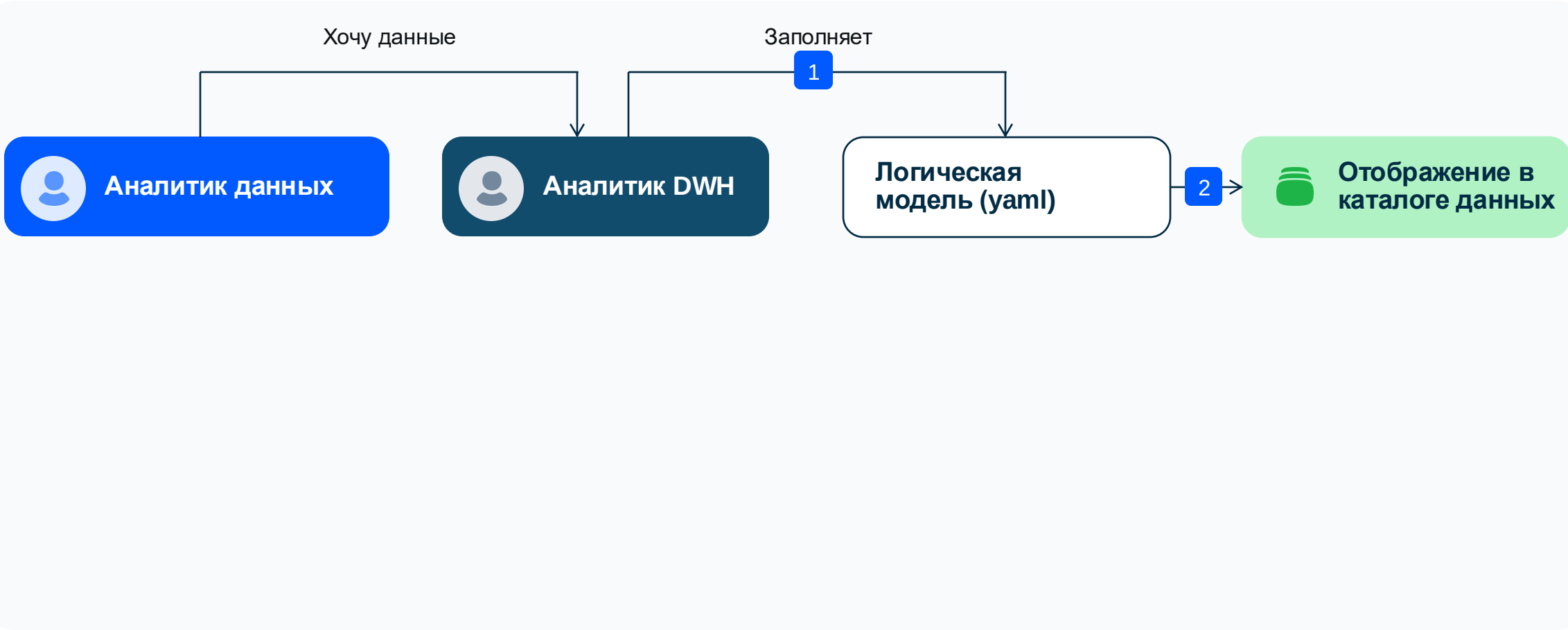
Flow работы с лог. моделью



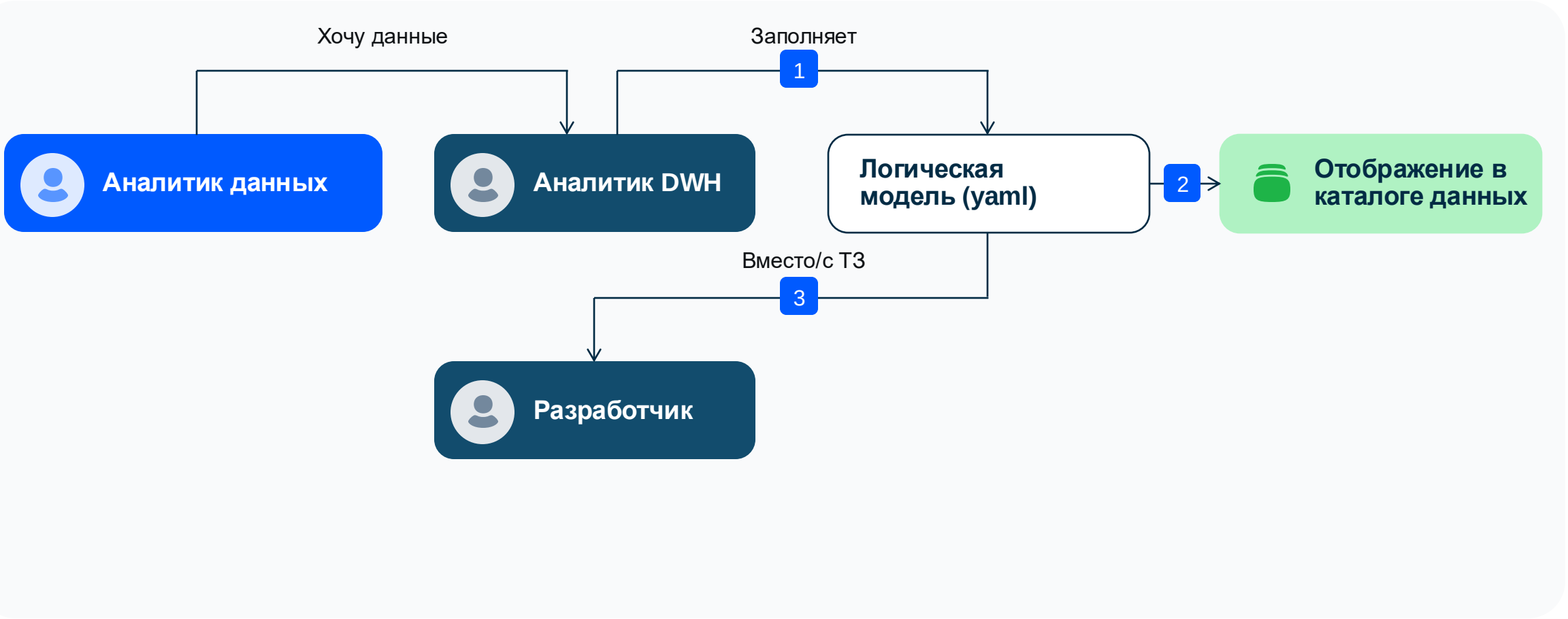
Flow работы с лог. моделью



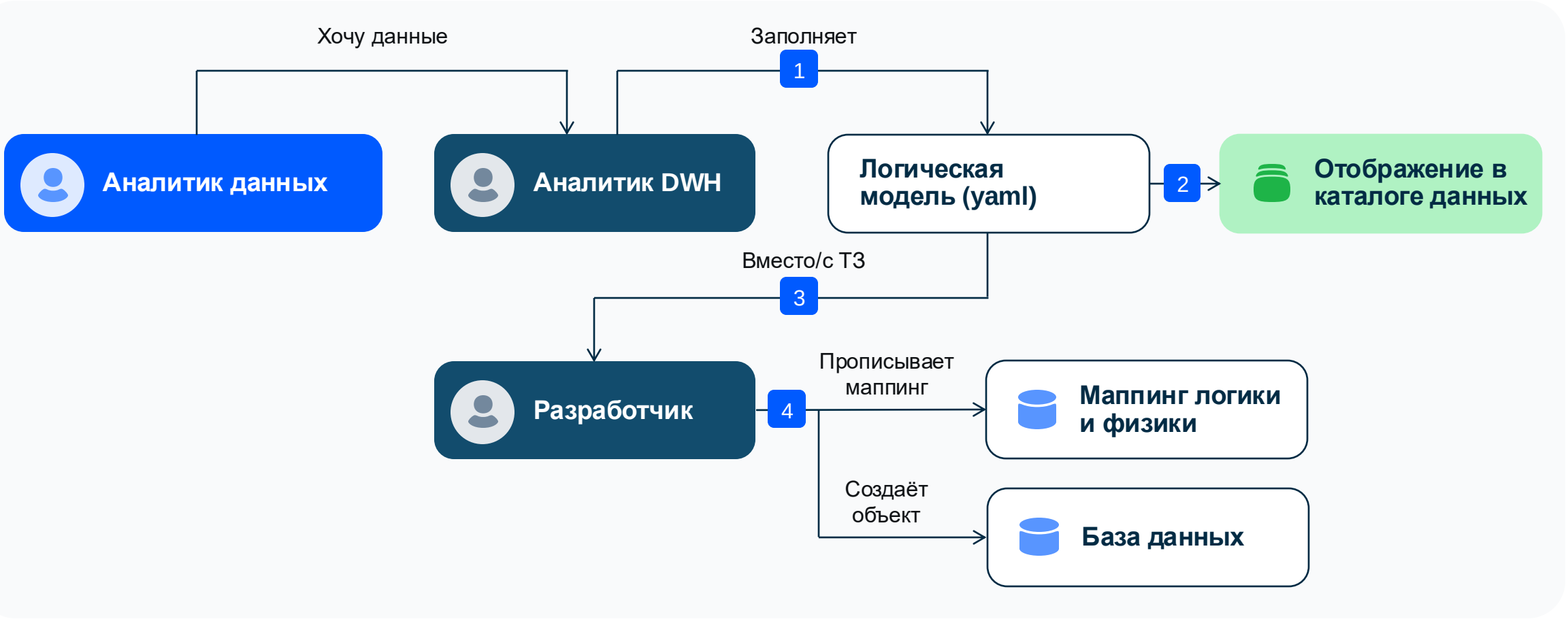
Flow работы с лог. моделью



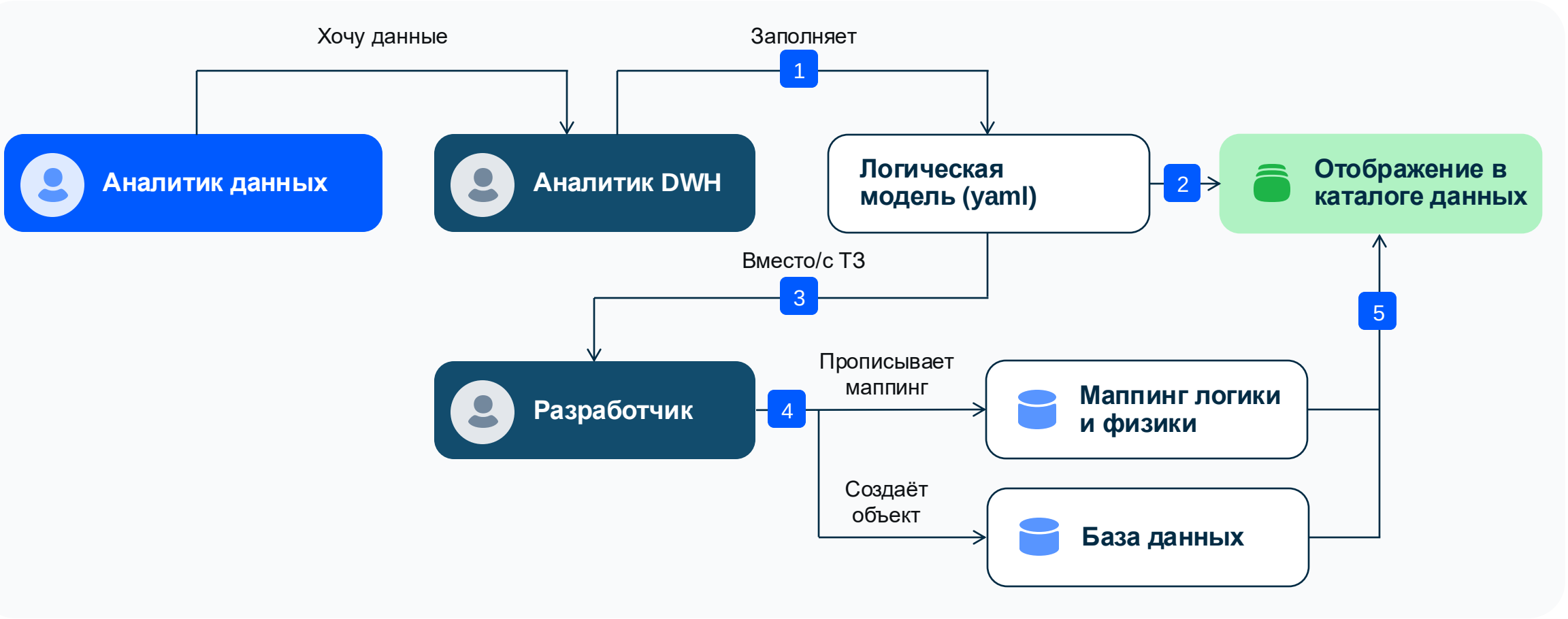
Flow работы с лог. моделью



Flow работы с лог. моделью

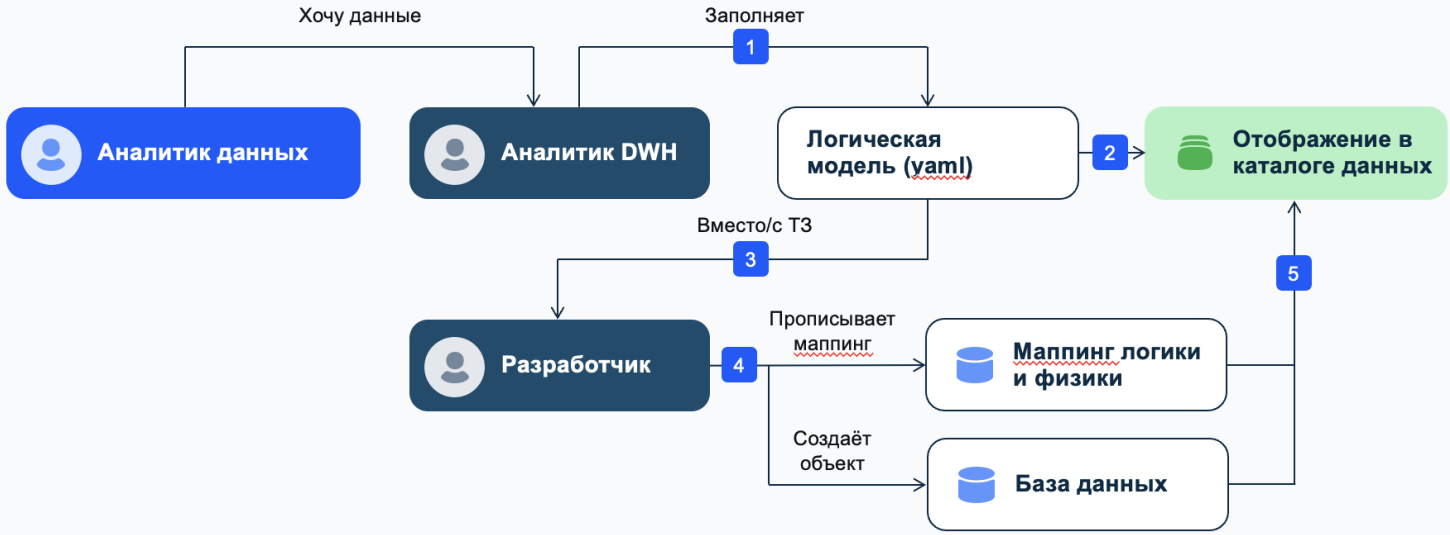


Flow работы с лог моделью



Flow работы с лог. моделью

Преимущества процесса



- Каждый оставляет «след» в каталоге данных
- Отсутствие ручного заведения объектов в каталоге данных
- Каждый отвечает за свою часть

Уроборос построения модели

- Быстрее
- Каталог обновляется самостоятельно
  без ручных корректировок
- Прошли весь путь полностью,
  с остальными хранилищами будет проще

- Процесс перестроить сложно
  без доп. инструментов моделирования
- Сработало только для якорной модели
- Другие хранилища придется
  «дружить вручную»

План работы с другими хранилищами

План работы с другими хранилищами

Лог. модель
для всех
хранилищ

План работы с другими хранилищами

Лог. модель для всех хранилищ

Линк между хранилищами

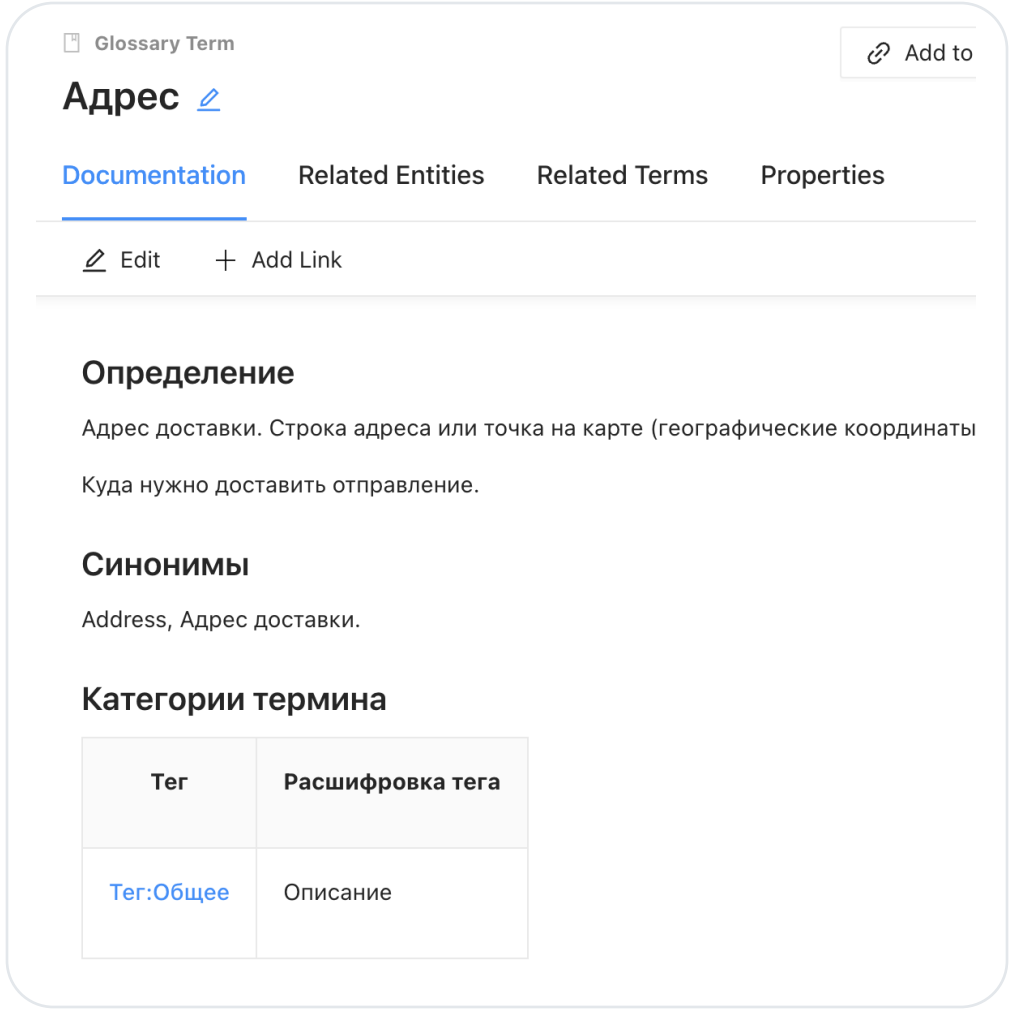
План работы с другими хранилищами

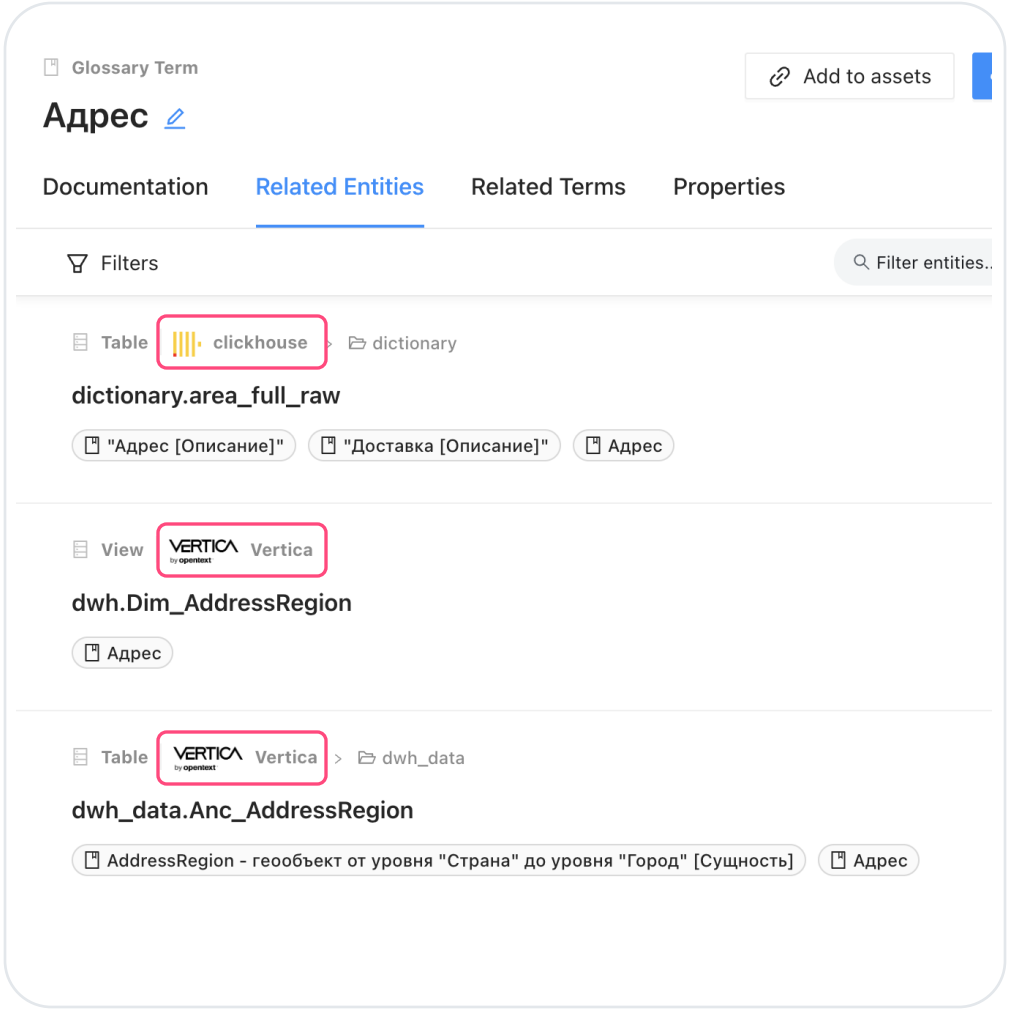
Лог. модель для всех хранилищ

Линк между хранилищами

Объединить хранилища

К чему стремиться

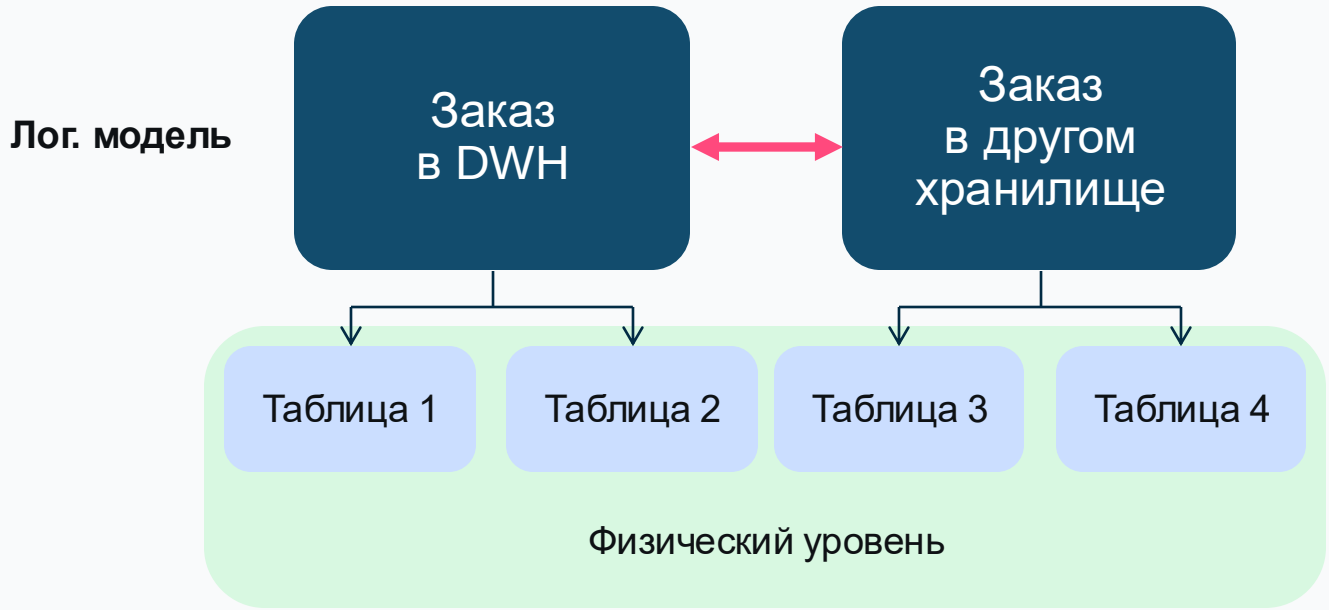




Как это сделать?

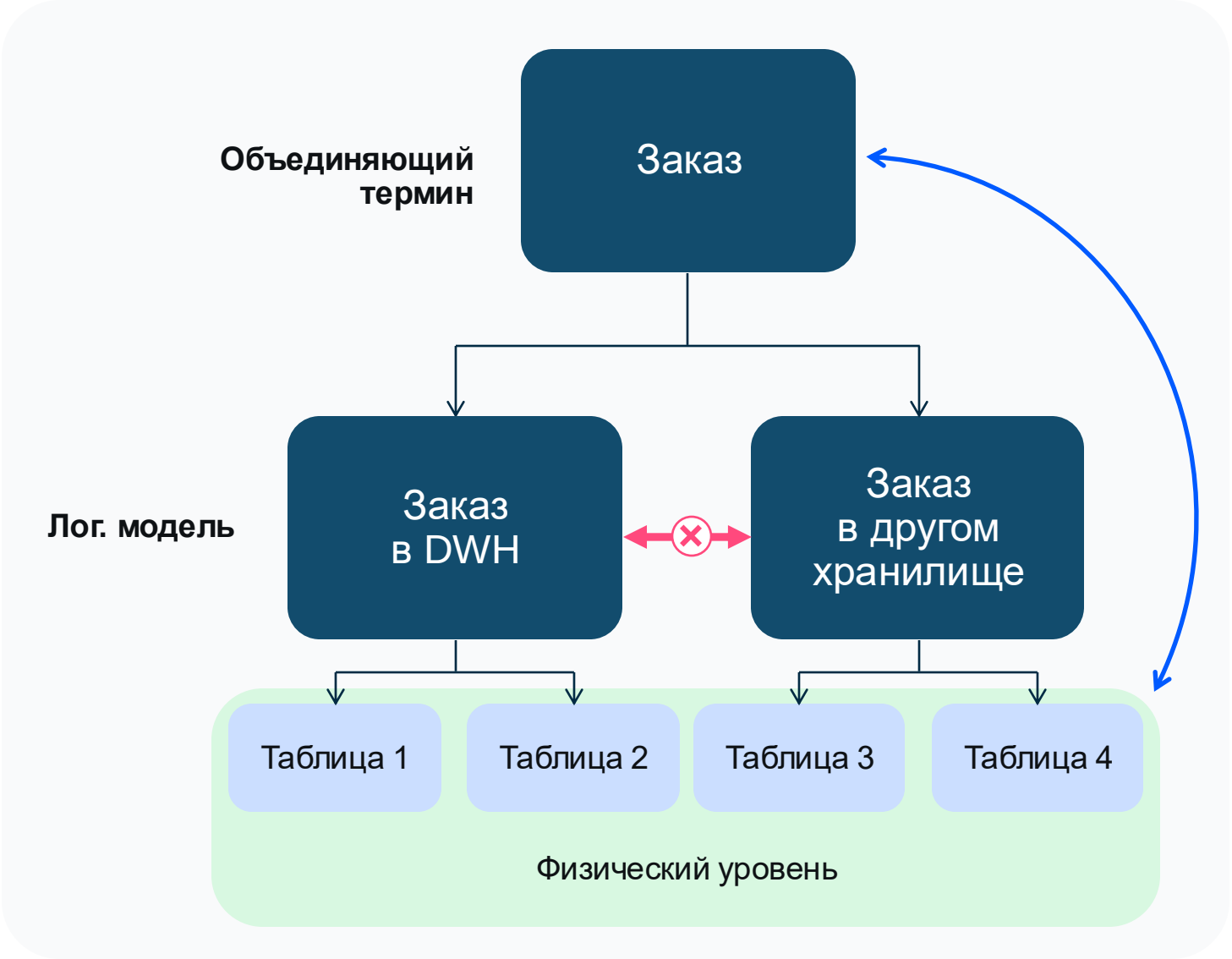
Идеально

- **Сделать лог. модель др. хранилищ**
- **Найти связи между объектами лог. моделей разных хранилищ**

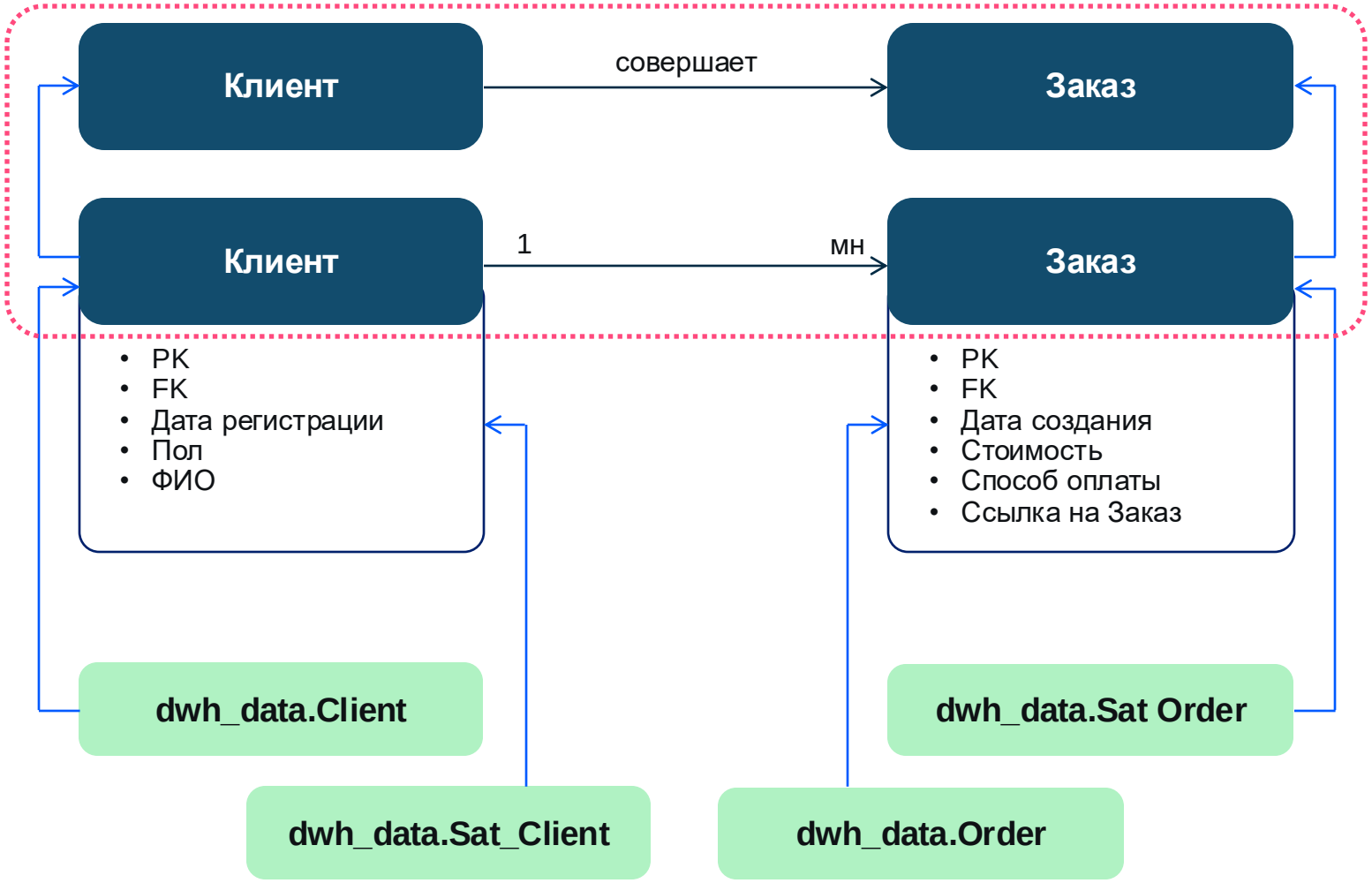


Идеально

- **Пользователю больше кайфа**
- **Много ручной работы, увы и ах**

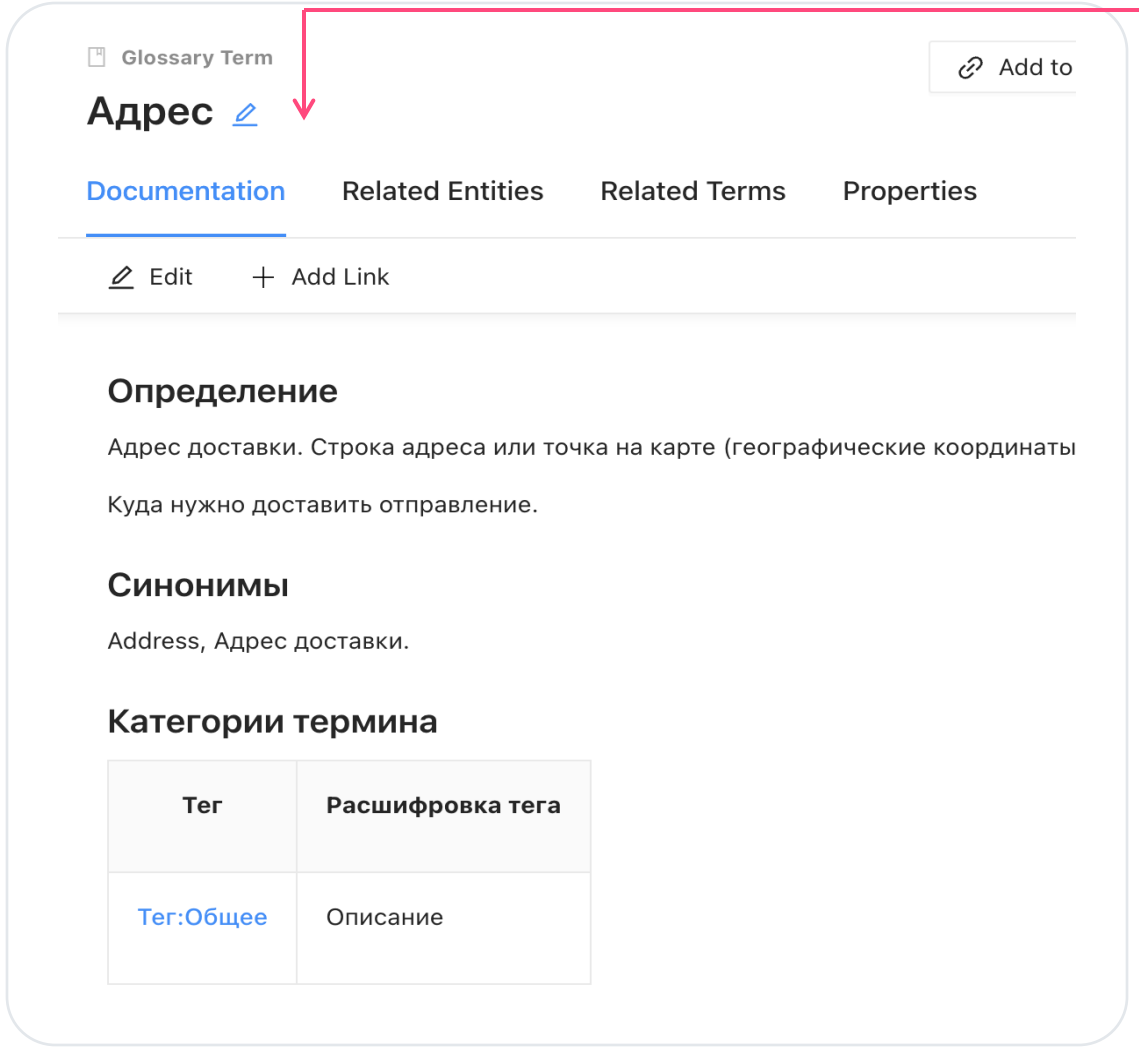


От логики к концепции



Неправильный, но жизненный подход

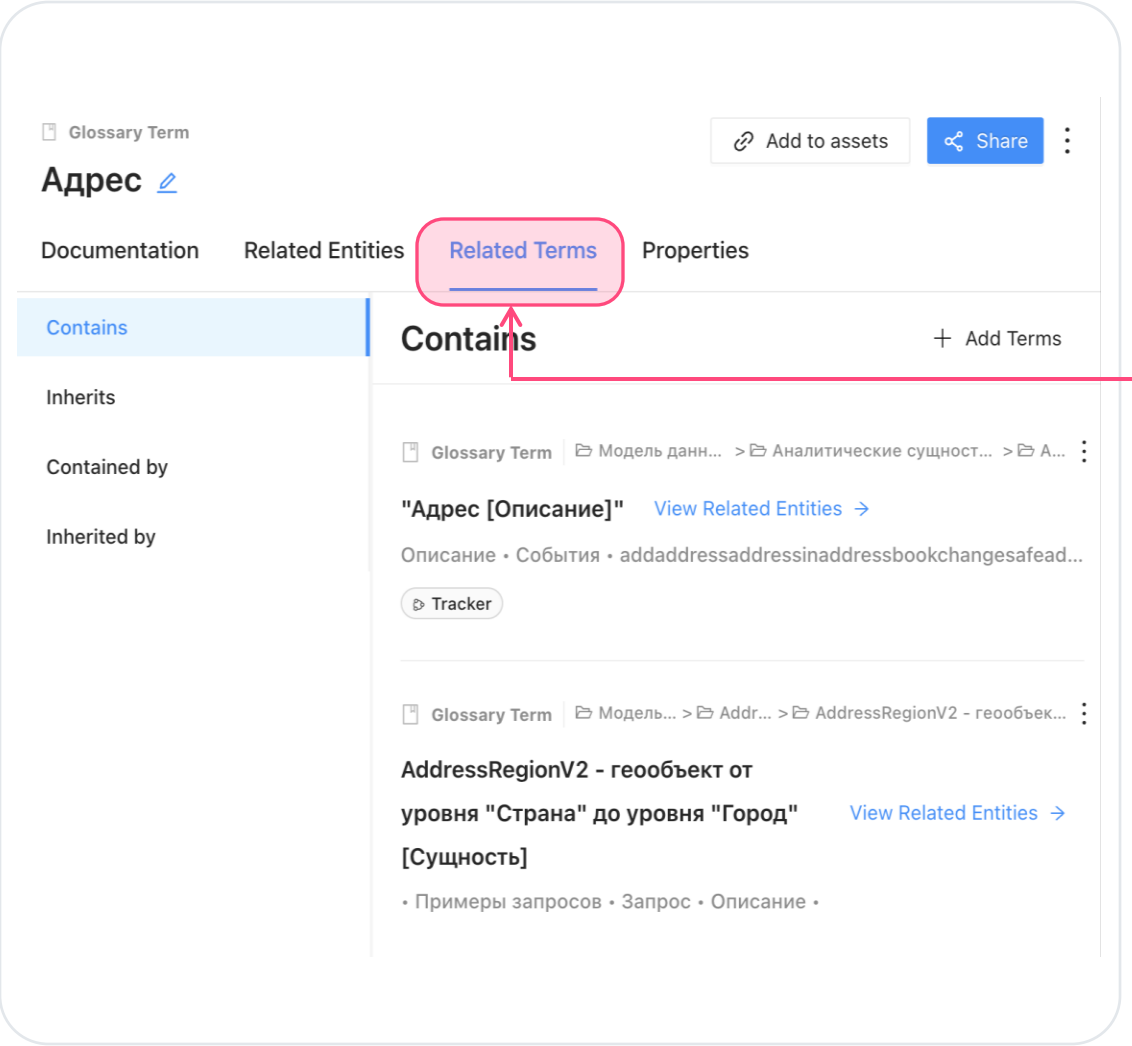
Пользователь находит термин в глоссарии



Объединяющий термин

Адрес

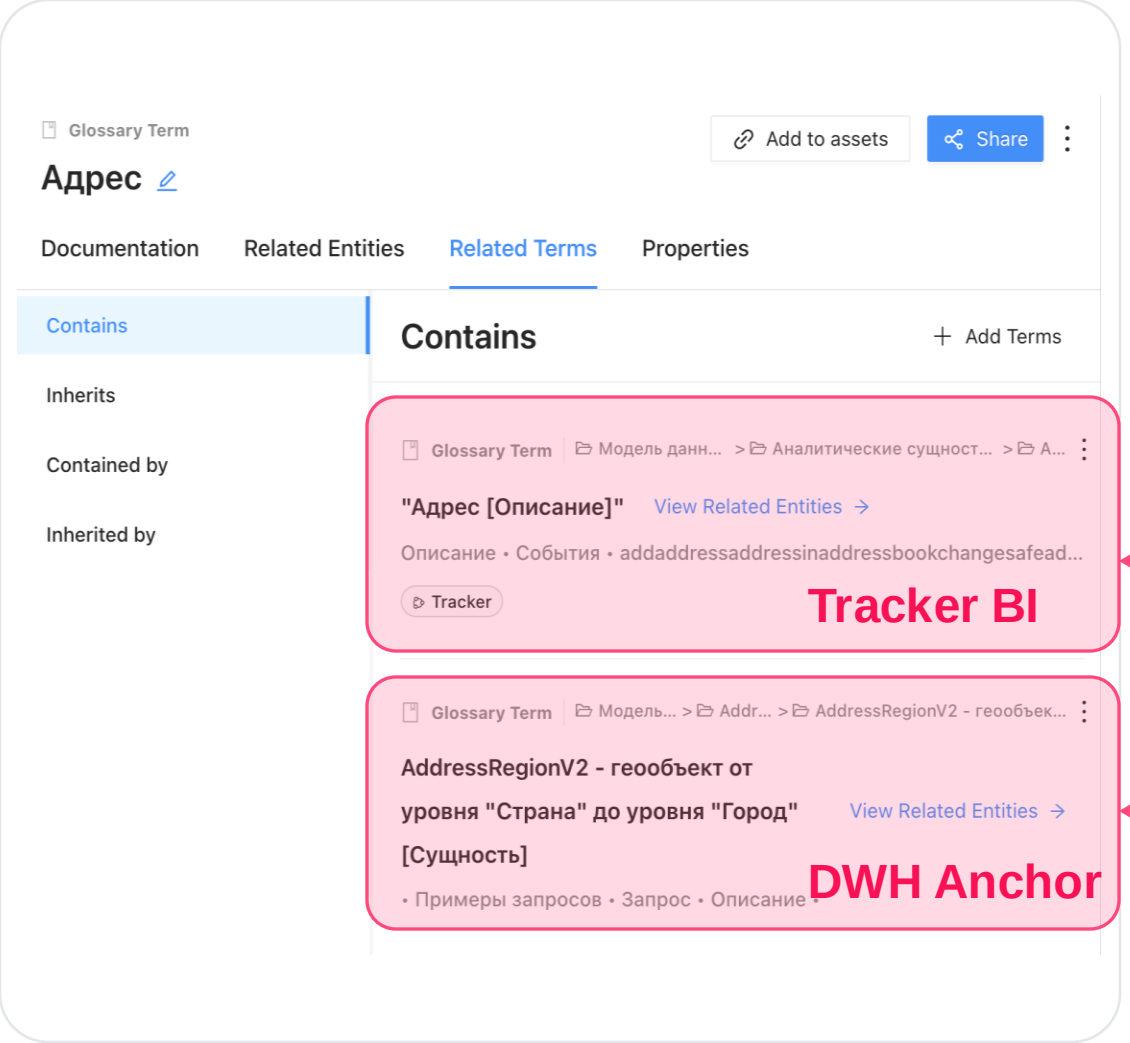
В каких лог. моделях присутствует

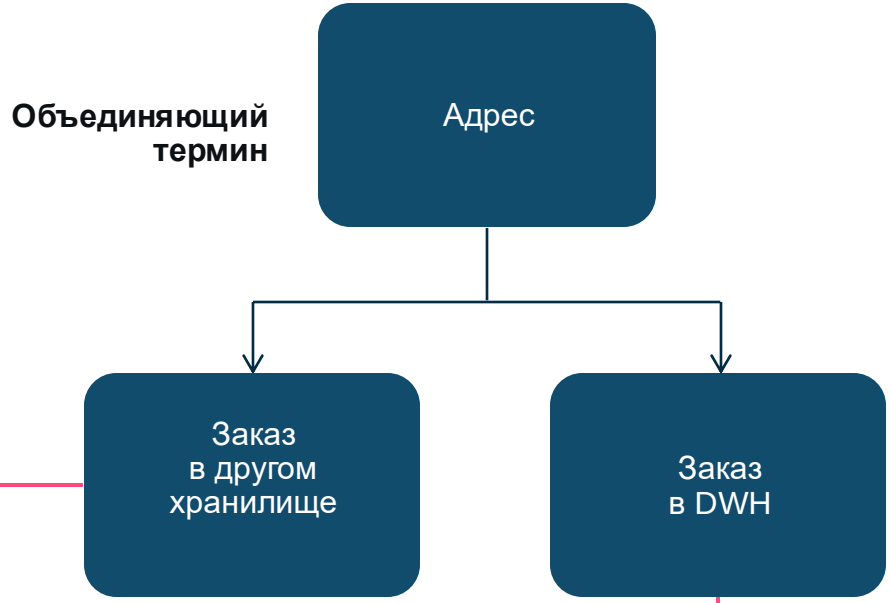


Объединяющий термин

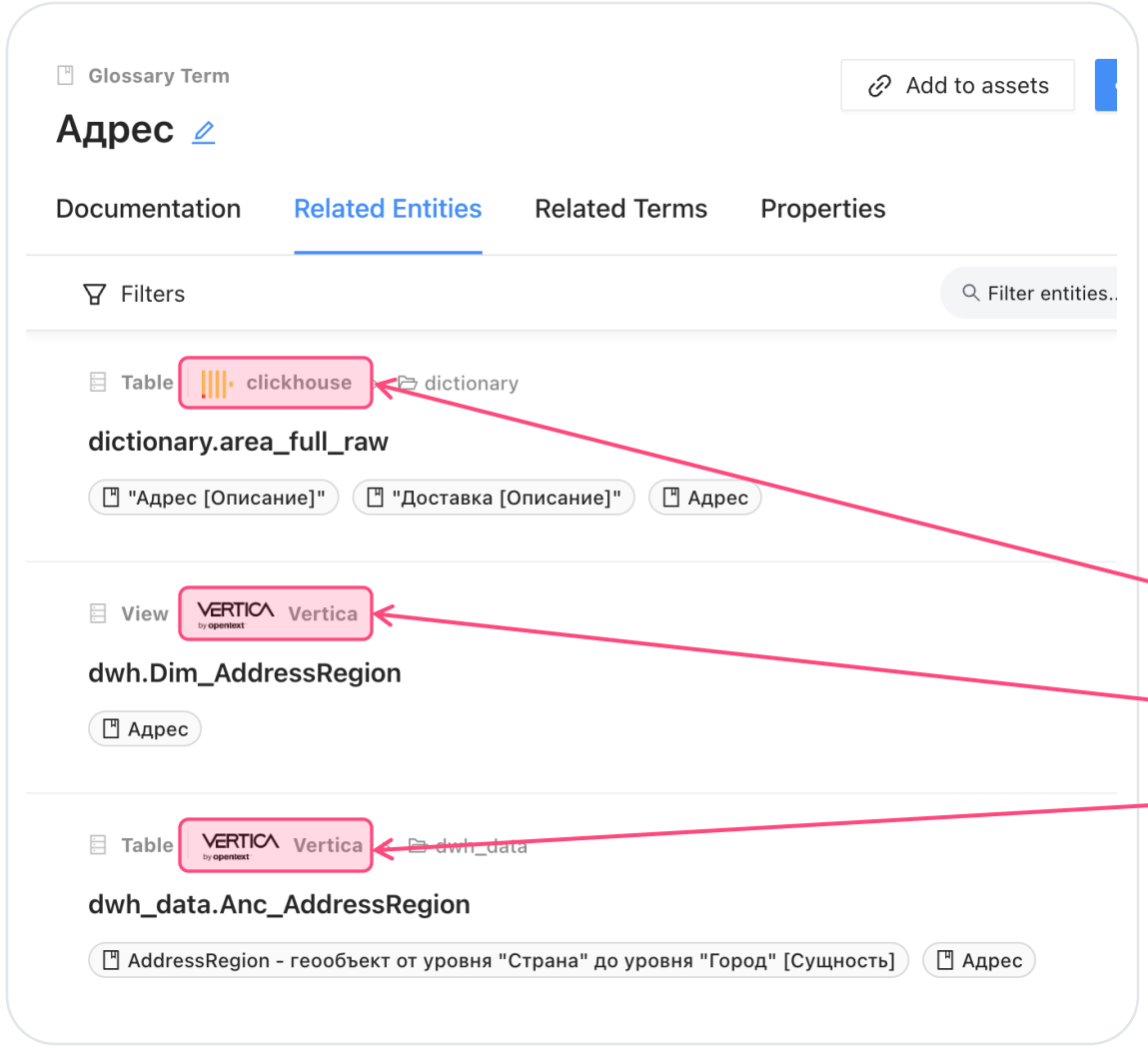
Адрес

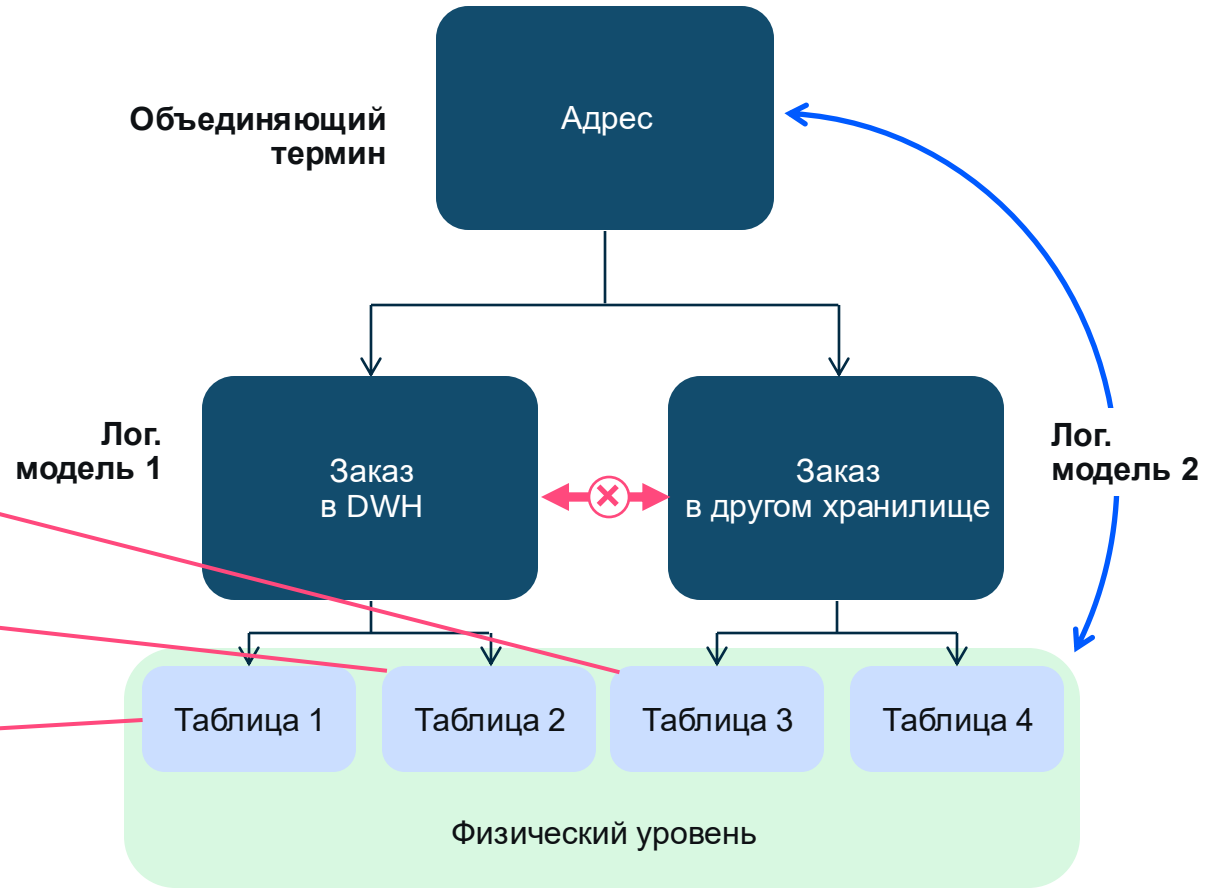
Пользователь находит термин в глоссарии





Сразу видит таблицы с нужными данными





Какие **фичи** были добавлены в каталог

Фичи, которые были сделаны для удобства **пользователя**

01

Фичи, которые были сделаны для удобства **стюарда**

02

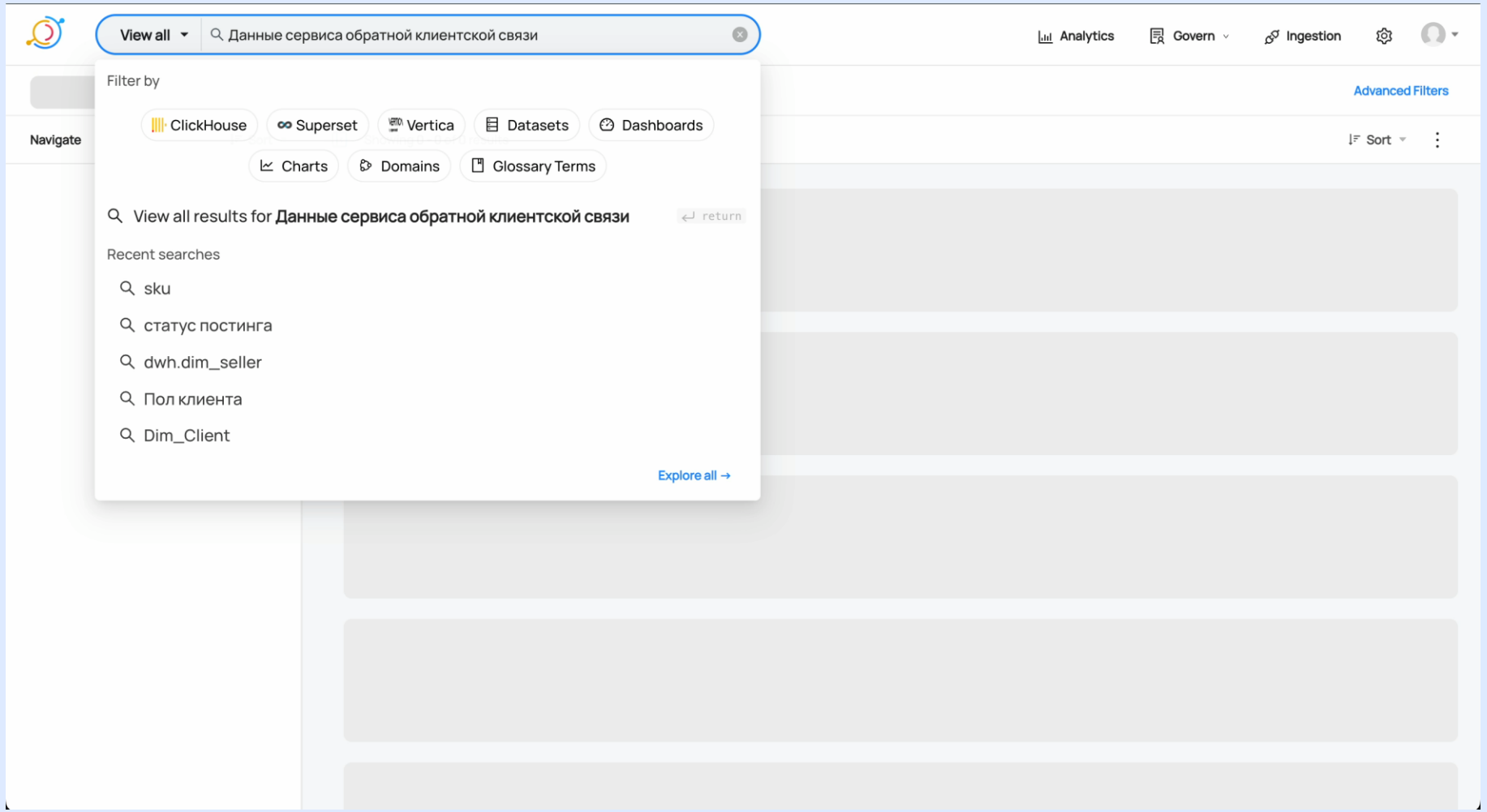
Сборная документация

- **Contract**
  - **Введение**
  - **Bank - Справочник Банков РФ**
    - Bank - Справочник Банков РФ [Сущность]
    - Bank_Address - Адрес отделения банка РФ
    - Bank_Name - Наименование отделения банка РФ
    - Contract_Bank
    - PayProps_Bank
  - **Contract - Договор**

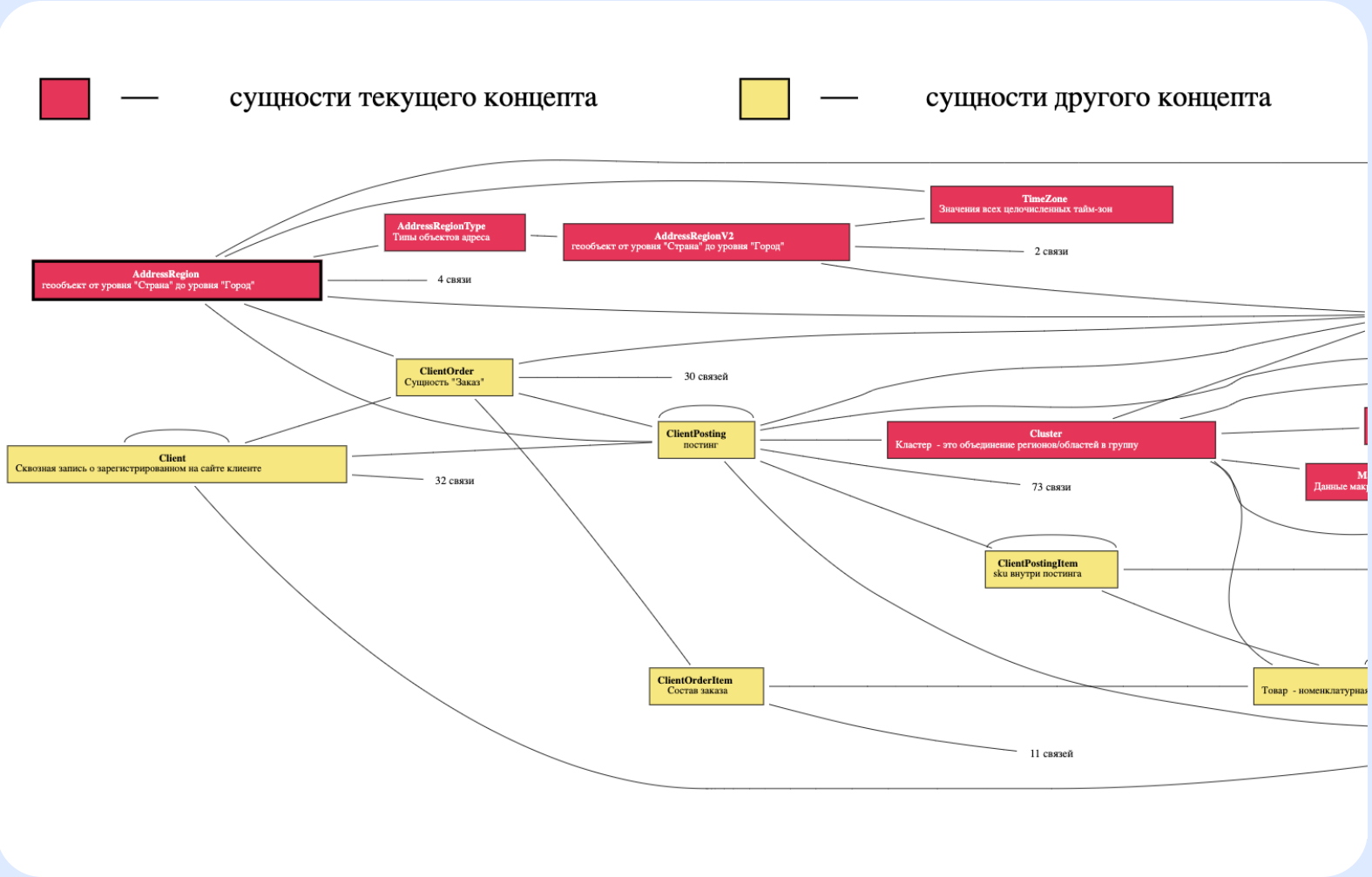
Attribute

Сущность	Атрибут	Описание
AbExperiment	AbExperiment_CreatedTime	Время создания эксперимента.
AbExperiment	AbExperiment_ExperimentEndTime	Время окончания влияния на пользователей.
AbExperiment	AbExperiment_ExperimentStartTime	Время старта эксперимента (когда он начинает влиять на пользователей).
AbExperiment	AbExperiment_Name	Человеко-читаемое название варианта A/B эксперимента.
AbVariant	AbVariant_IsControl	Признак того, что данный вариант является контрольной группой.
AbVariant	AbVariant_Name	Человеко-читаемое название варианта A/B теста.

Сборная документация



Графы модели



- **Для тех, кто любит смотреть, а не читать**
- **Дополнительная навигация**

Примеры запросов

```
SELECT ClientOrderId,
       aa.SourceKey,
       a.ClientPostingId,
       a.ParentClientPostingId
FROM dwh_data.Anc_ClientPosting aa
INNER JOIN dwh_data.Tie_ClientPosting_ClientPosting_Parent a
ONa.ClientPostingId = aa.ClientPostingId
INNER JOIN dwh_data.Tie_ClientPosting_ClientOrder b
ONa.ClientPostingId = b.ClientPostingId
WHERE SourceKey ilike ''66068576-0038-%''
```

- **Неактуальные запросы не добавляются**
- **Один запрос может быть примером на N объектов**

Примеры запросов

```
SELECT ClientOrderId,
       aa.SourceKey,
       a.ClientPostingId,
       a.ParentClientPostingId
FROM dwh_data.Anc_ClientPosting aa
INNER JOIN dwh_data.Tie_ClientPosting_ClientPosting_Parent a
ONa.ClientPostingId = aa.ClientPostingId
INNER JOIN dwh_data.Tie_ClientPosting_ClientOrder b
ONa.ClientPostingId = b.ClientPostingId
WHERE SourceKey ilike ''66068576-0038-%''
```

3 таблицы в запросе = ±20 страниц в каталоге, где он находится

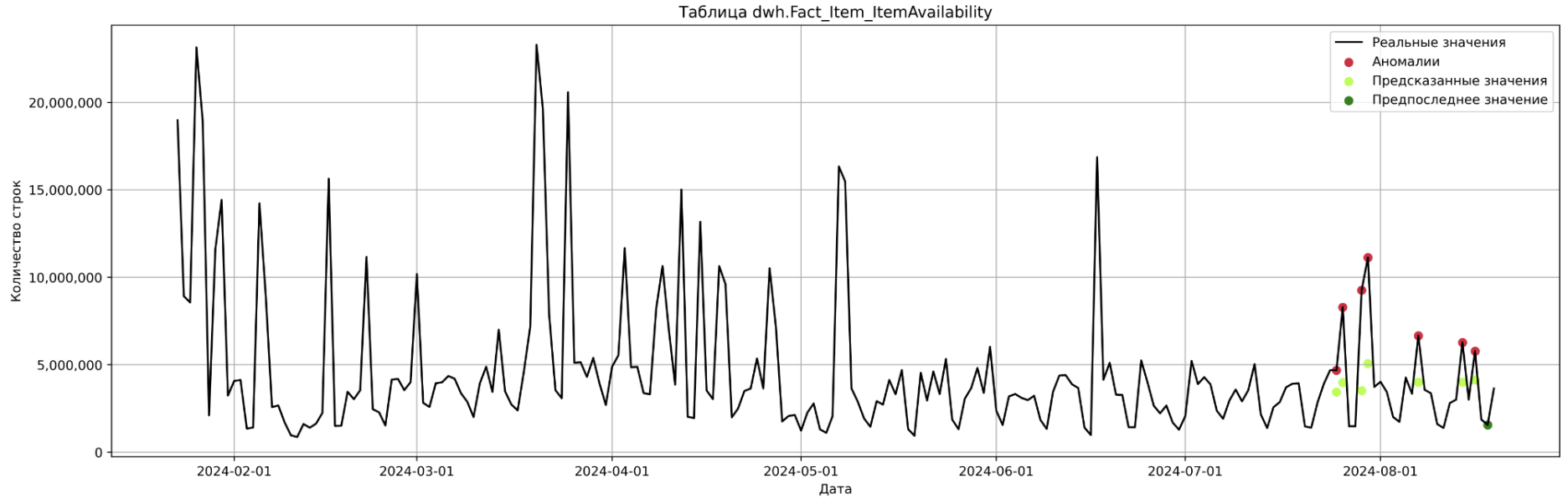
Информация о Data Quality

Schema | View Definition | Documentation | Lineage | Properties | Queries | Stats | Validation | Incidents

Основная проверка на аномальность

- **Метод проверки:** тройное экспоненциальное сглаживание.
- **Время проверки:** 20.08 в 09:11 МСК.
- **Расписание:** 9:00 МСК.

В таблице по полю StartTime **за предыдущий день (19.08) всё в порядке**. Ожидалось 3 807 274 строк, получено 3 644 664 строк.



Затрачиваемые ресурсы и результаты

Полгода на описание
1,5 хранилищ

6 аналитиков,
1 разработчик, 1 девопс

Вопросы в чатах
ушли на 70%

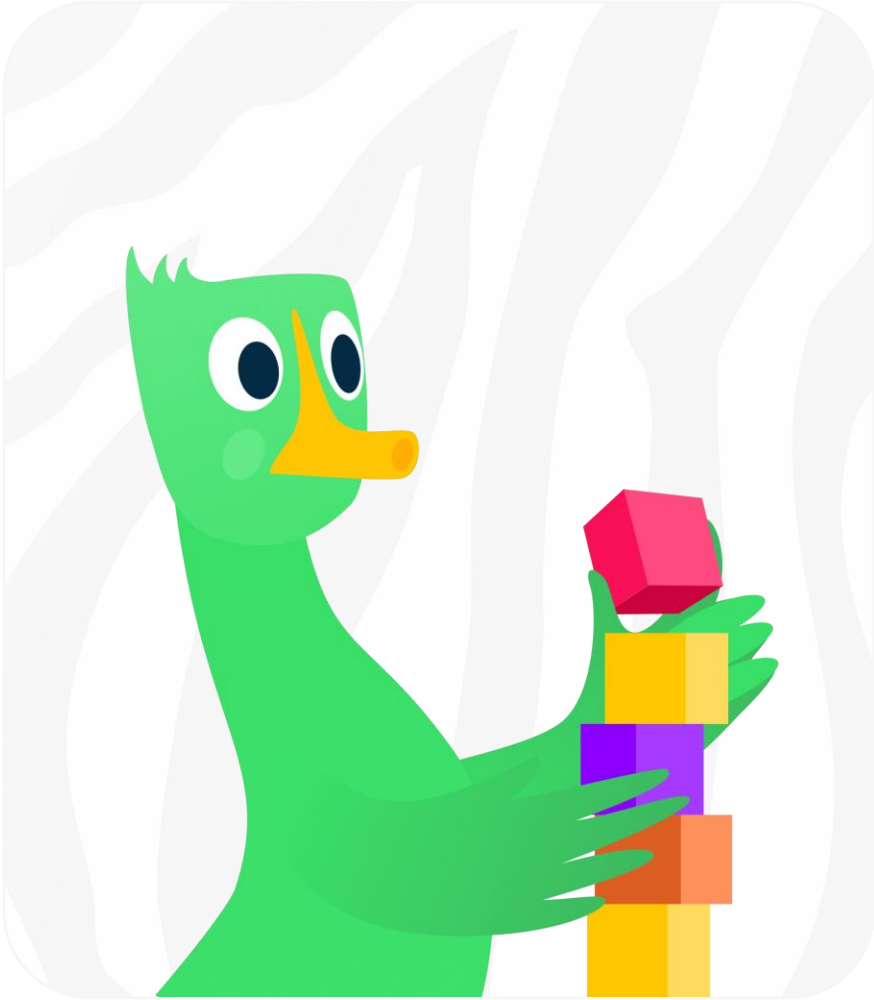
- **Нашли ошибки в логике хранилища**
- **Время на документацию 4 часа в неделю — меньше часа в неделю**
- **Ответ дежурного — в течение нескольких минут**

Дельные мысли

- Начать с логической модели
- Собирайте документацию уже сейчас
- Не пытаться объединять разные хранилища на старте, если это не было заложено изначально архитекторами хранилищ
- Снизу вверх — подход, которого не нужно стыдиться ☺
- MVP обошелся малой кровью, а вот вспомогательные инструменты — дорогое удовольствие

Темы для следующего раза

- Сколько времени ушло на сбор и консолидацию единой модели для всех хранилищ
- Процесс управления единой моделью и мониторинг ее чистоты
- Какие трудности были с добавлением «метрик»



OZON tech

Спасибо за внимание!

Наталья Журавлёва
Должность
tg: @omon_ra

Ссылка QR-кода