как мы в хранилище маркета начали писать документацию к нашим объектам

Павел Колодкин
Руководитель разработки в хранилище данных маркета

ГОРОДСКИЕ СЕРВИСЫ

познакомимся!

01 Работаю в хранилище Маркета с октября 2019 года

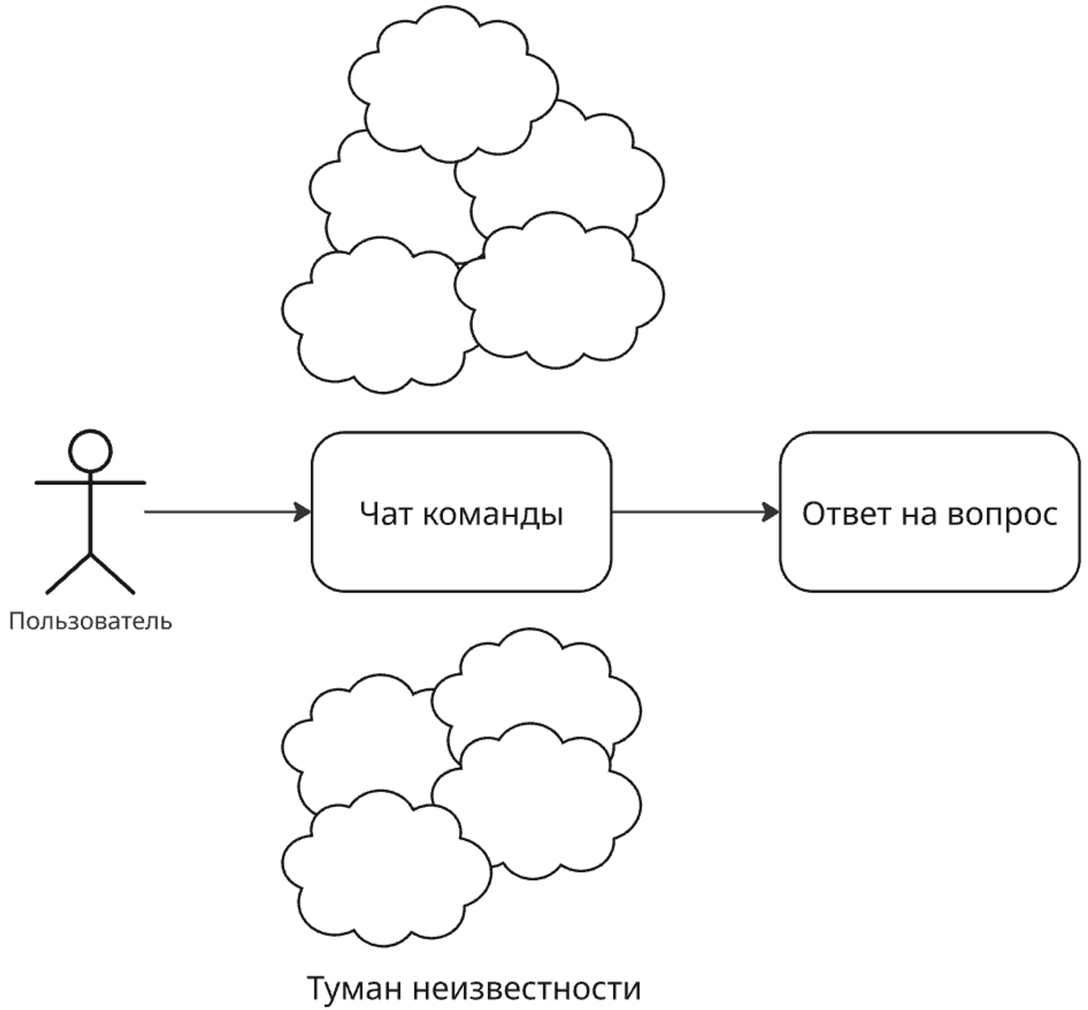
02 Наблюдал и участвовал в процессах развития документации, и сегодня я расскажу вам про них

а можно ли вообще работать без документации?

да

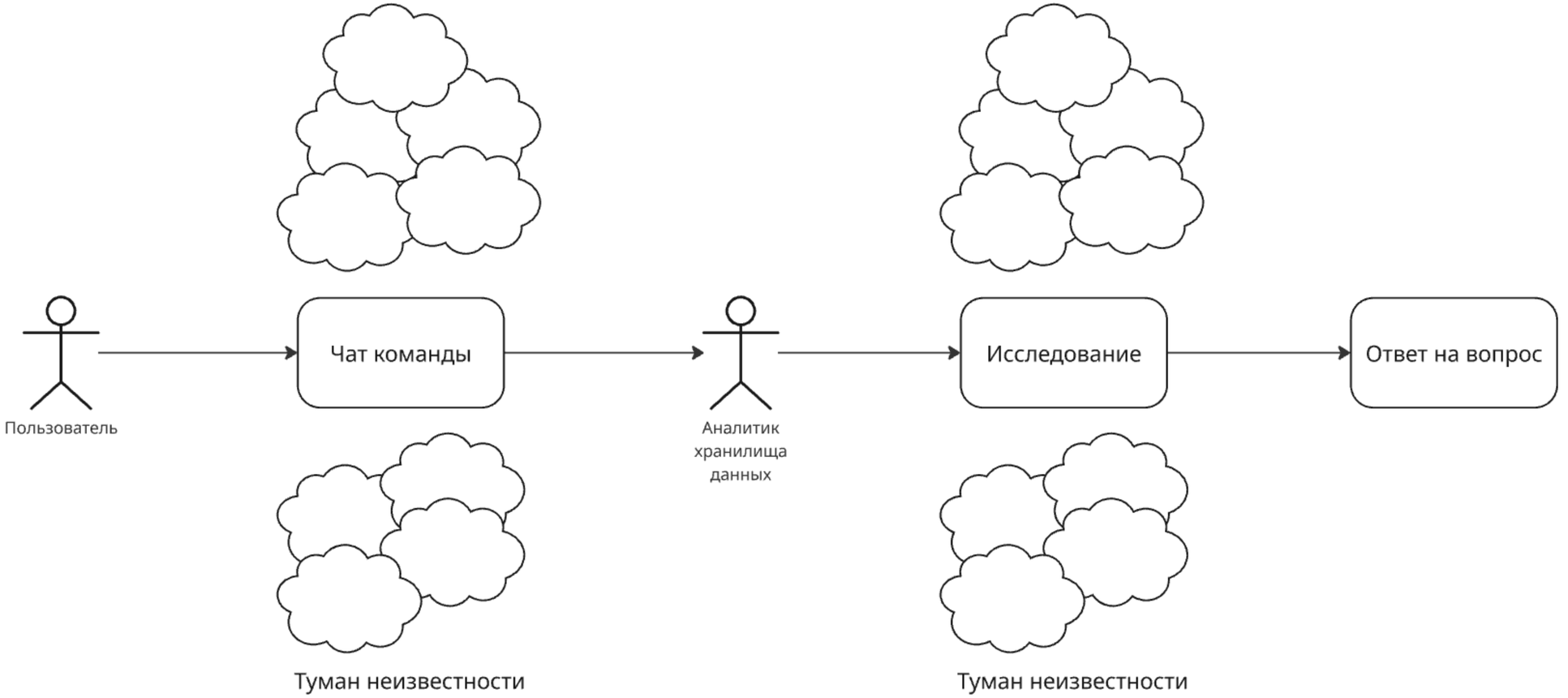
как найти нужную информацию

как найти нужную информацию



успех?

как найти нужную информацию



проблематика

01

Мы не знаем, какие данные
есть у нас в хранилище

проблематика

01

Мы не знаем, какие данные
есть у нас в хранилище

02

Приходится переделывать
одну и ту же работу
по несколько раз

проблематика

01

Мы не знаем, какие данные
есть у нас в хранилище

02

Приходится переделывать
одну и ту же работу
по несколько раз

03

Долгая адаптация
новых членов команды

проблематика

01

Мы не знаем, какие данные
есть у нас в хранилище

02

Приходится переделывать
одну и ту же работу
по несколько раз

03

Долгая адаптация
новых членов команды

04

Сложно производить
миграции пользователей
на новые источники данных

О ЧЁМ ПОГОВОРИМ

раздел

01

теория

вопросы, на которые нам необходимо ответить

Что описывать?

вопросы, на которые нам необходимо ответить

Что описывать?

Как описывать?

вопросы, на которые нам необходимо ответить

Что описывать?

Как описывать?

Когда это делать?

что и как описывать

01 Diátaxis

02 CTRT

03 Information Mapping

04 EPPO

diátaxis

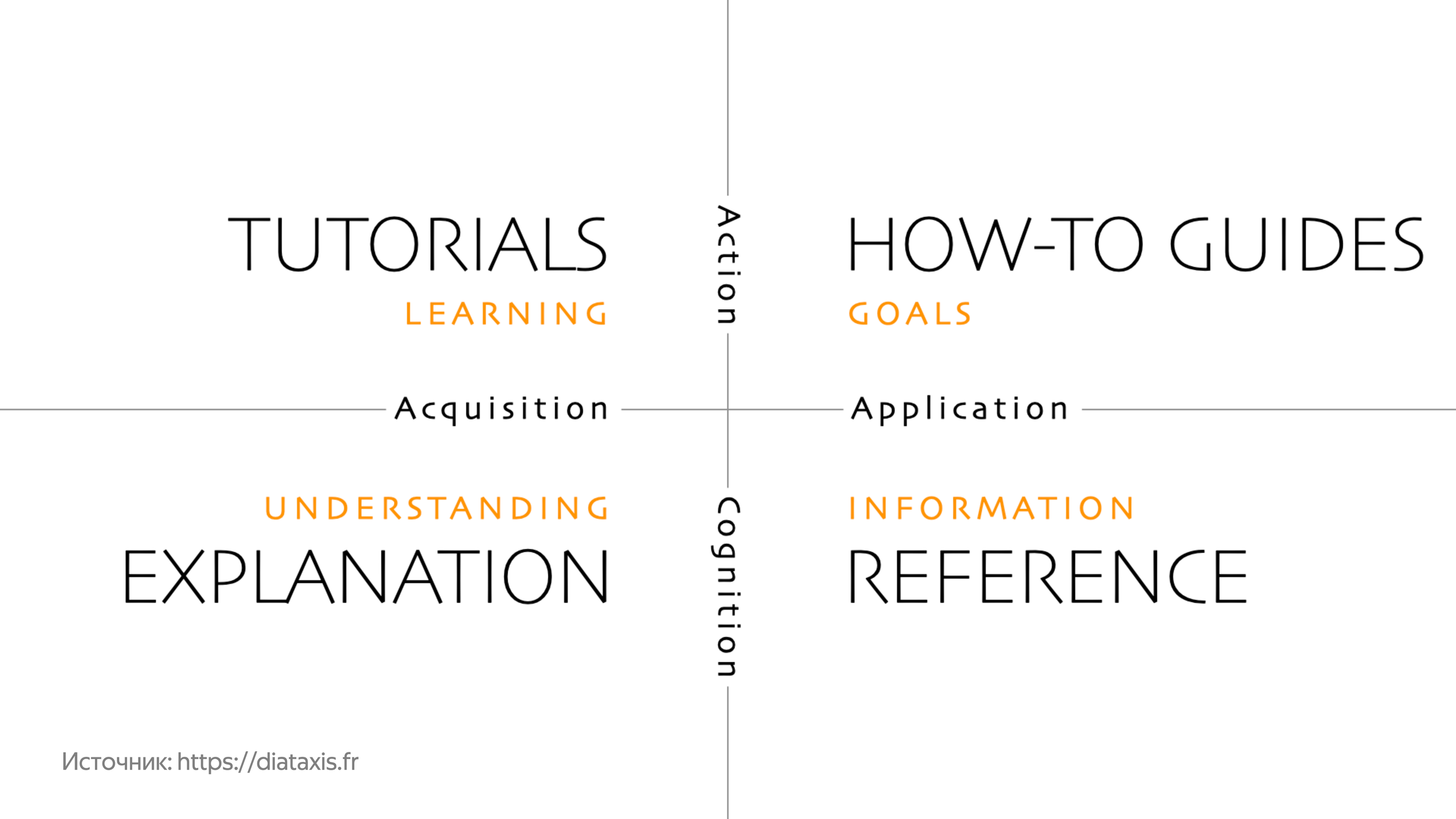
01 4 домена: Tutorial, How-to Guide, Explanation, Reference

02 Отделяет обучение от выполнения задач

03 Структура под цель пользователя

04 Короткие, завершённые единицы контента

05 Универсально для любых продуктов



Источник: https://diataxis.fr

CTRT

01 Concept, Task, Reference (+Troubleshooting)

02 Чёткое разделение по типу информации

03 Лёгкий поиск и навигация

04 Часто сочетают с шаблонами и ярлыками

information mapping / structured writing

01 Метод Роберта Хорна

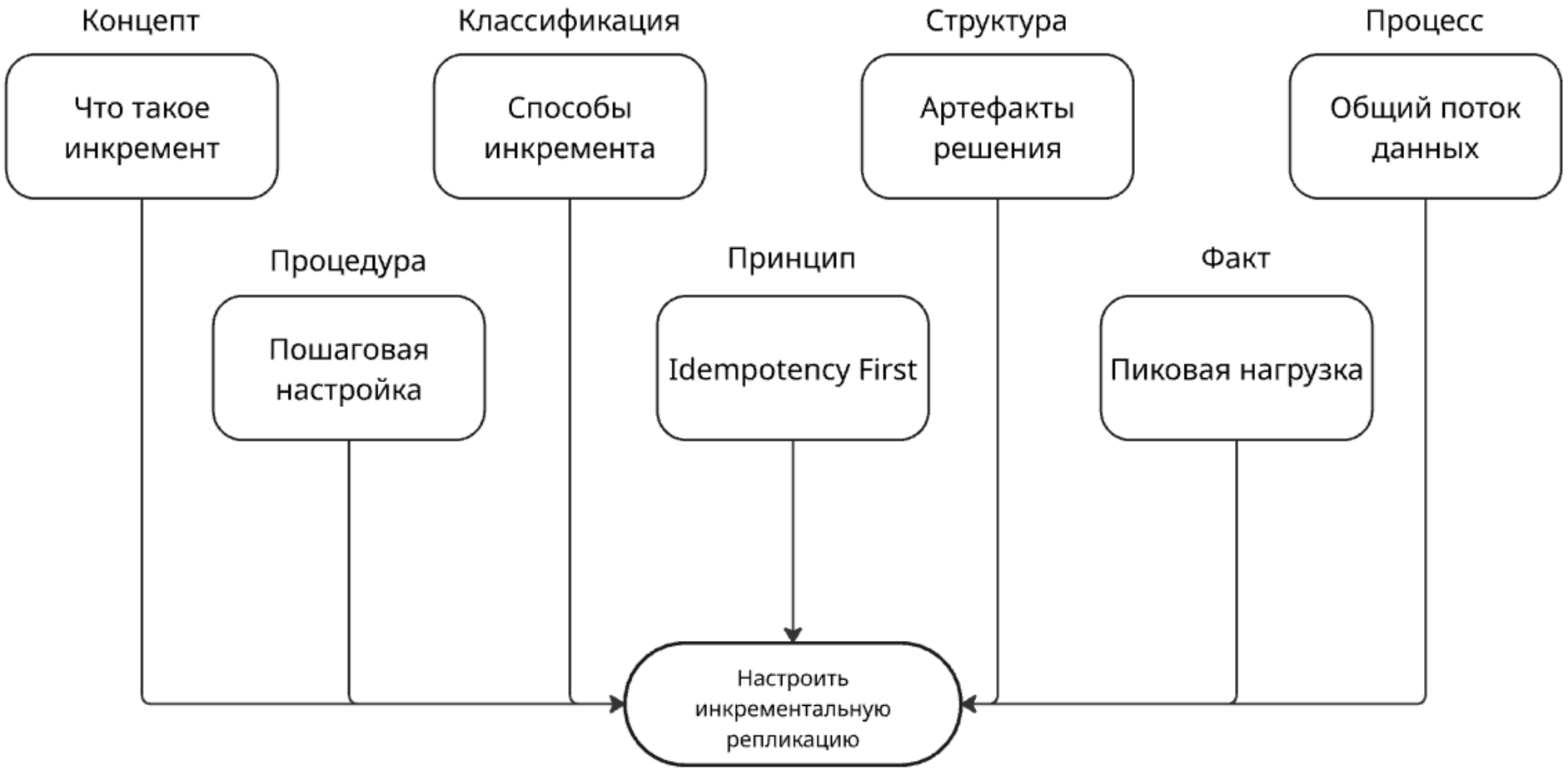
02 Делит текст на информационные «блоки» и «карты»

03 7 категорий: процедура, концепт, принцип, факт, структура, классификация, процесс

04 Когнитивная эргономика, белые поля, визуальные маркеры

05 Улучшает читаемость и усвоение

пример


Концепт
Что такое инкремент
Классификация
Способы инкремента
Структура
Артефакты решения
Процесс
Общий поток данных
Процедура
Пошаговая настройка
Принцип
Idempotency First
Факт
Пиковая нагрузка
Настроить инкрементальную репликацию

every page is page one (EPPO)

01 Принцип Марка Бейкера (2013)

02 Каждая страница — самостоятельная точка входа

03 Узкий фокус темы + обилие ссылок

04 Минимум иерархии, максимум сети

05 Ориентировано на веб-поиск

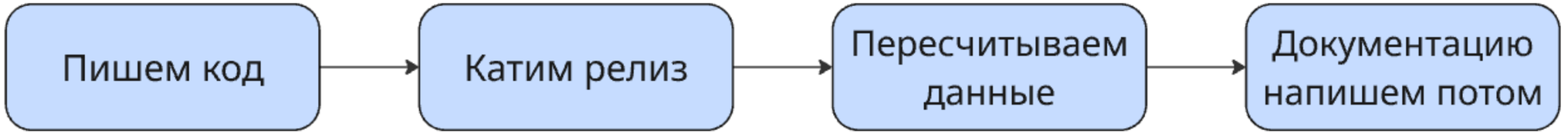
ИТОГО

Фреймворк	Типы контента	Фокус	Единица	Сильные стороны
Diátaxis	Tutorial, How-to, Explanation, Reference	Цель пользователя	Статья	Чёткое разделение доменов
CTR/CTRT	Concept, Task, Reference, Troubleshooting	Поиск типа инфы	Тема	Простота, полнота цикла
Information Mapping	процедура, концепт, принцип, факт, структура, классификация, процесс	Когнитивная эргономика	Блок / карта	Возможность переиспользования одной информации в нескольких статьях

когда описывать?

когда описывать?

ПОТОМ



проактивные подходы документирования

Documentation-Driven Development (DDD)	Schema-First / Design-First	Metadata-First
Contract-First	Continuous Documentation	Living Documentation

documentation-driven development

01 Документация пишется первой

02 Качественная дока встроена

03 Требует дисциплины

schema-first (design-first)

01 Сначала контракт/схема, потом реализация

02 Популярен для API: OpenAPI или ER-модели до кодирования

03 Хорошо, когда вы договариваетесь между двумя системами, для разработки внутри хранилища скорее не подходит

contract-first

01 Контракт-ориентированный (для интеграций/сервисов)

02 Определяем интерфейс взаимодействия (API, формат файлов) до разработки

03 Команды договариваются о формате

04 Нужно время на согласование, жесткость изменений

05 Отлично подходит на «сырых» слоях хранилища

06 Каждый контракт содержит схему, но не каждая схема является полноценным контрактом — кроме схемы содержит также набор нефункциональных требований

metadata-first

01 Сначала моделируем метаданные (описания), код делаем потом

02 Код можно генерировать впоследствии по документации

03 Требует специального инструмента для ведения и хранения метаинформации

CI/CD + documentation

01 Документация = часть workflow, пишется по чуть-чуть всегда

02 Привязана к коду (например, хранится в репозитории рядом)

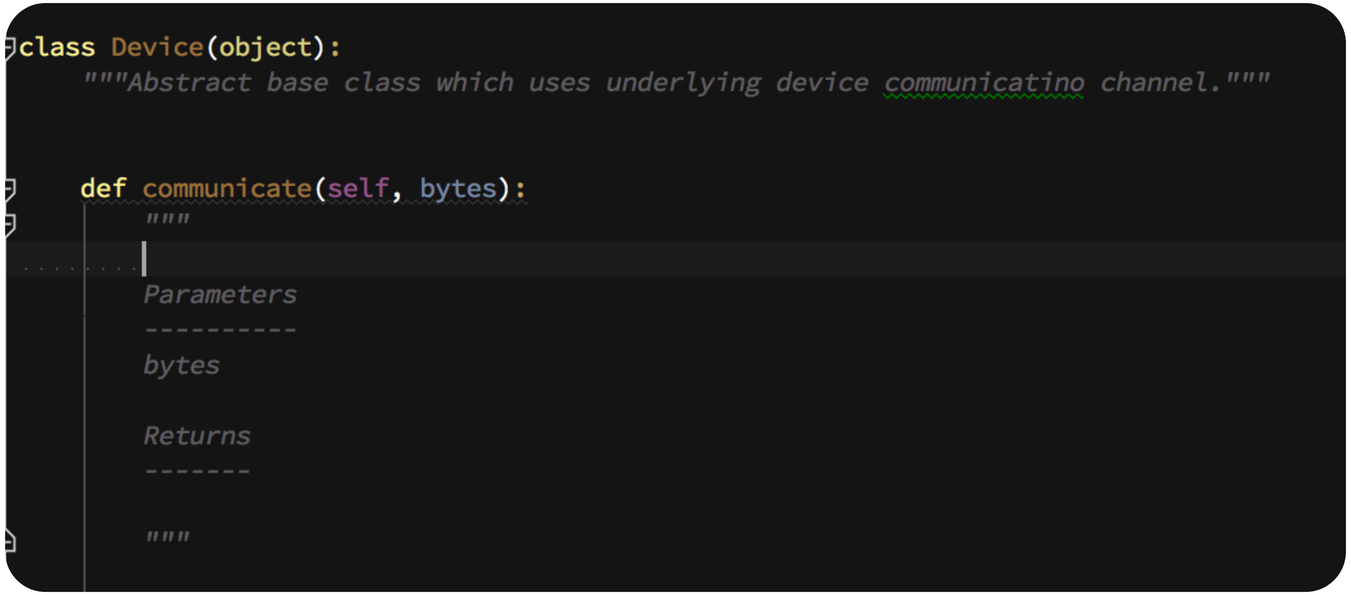
03 Всегда актуальна (CI проверяет, обновляется вместе с релизами)

auto documentation

01 Автогенерация из кода/комментов

02 Документация всегда синхронизирована с софтом

03 Интерактивность: пользователи могут получать инфо в реальном времени, внутри инструментов разработки



сравнение подходов

Подход	Момент появления документации	Главный артефакт / инструмент	Уровень автоматизации	Сильные стороны	Слабые стороны	Лучшие случаи применения
Post-Implementation (Code-First)	**После** написания кода и запуска ETL	Wiki-страницы, ручные описания, комментарии в SQL	Низкий	• Быстрый старт разработки • Удобен для прототипов	• Документация устаревает или не пишется • Трудно поддерживать lineage	Небольшие пилоты, PoC, одноразовые выгрузки
Documentation-Driven Development (DDD)	**До** начала кодирования	README / спецификация функции, примеры использования	Средний (генерация тестов из примеров)	• Требования проясняются заранее • Документация всегда актуальна	• Требует дисциплины и времени на старте	API, библиотеки, открытые модули, когда важно DX
Schema-First / Design-First	**До** реализации (проектируем схему)	OpenAPI, GraphQL SDL, JSON/Avro Schema, DDL	Высокий (генерация stubs, доки, валидация)	• Ясный контракт данных • Параллельная работа команд	• Нужно уметь моделировать • Менять схему дорого	API, новые таблицы DWH, интеграции с фронтом
Metadata-First	**До** кода: описываем метаданные, потом генерим	Data Catalog, WhereScape, BIML, dbt-models	Очень высокий (код и дока из метаданных)	• Единый источник правды • Масштабируемость, lineage «из коробки»	• Сложное внедрение • Ограничения платформы	Крупные DWH, Data Vault, DataOps-платформы
Contract-First	**До** реализации, после согласования с потребителями	Data Contract YAML, AsyncAPI, Pact, SLA-док	Высокий (валидация, контрактное тестирование)	• Чёткие договорённости, SLA • Независимая работа команд	• Долгое согласование • Жёсткое versioning	Микросервисы, B2B обмен, Data Mesh домены
Continuous Documentation	**Во время** каждой задачи (in-process)	Docs-as-Code в Git, CI-шаги проверки	Средний (линтеры, автодеплой портала)	• Документация никогда не дрейфует • Встроена в workflow	• Нужна культура и проверки	Agile-команды, продукты с частыми релизами
Living Documentation	**Автоматически** при сборке/деплое	Swagger UI, dbt docs, Data Catalog сканеры	Очень высокий (полная автогенерация)	• Всегда актуальна, интерактивна • Минимум ручного труда	• «Сухая» без бизнес-контекста • Требует CI/CD и инструментов	API-платформы, большие каталоги данных

раздел

02

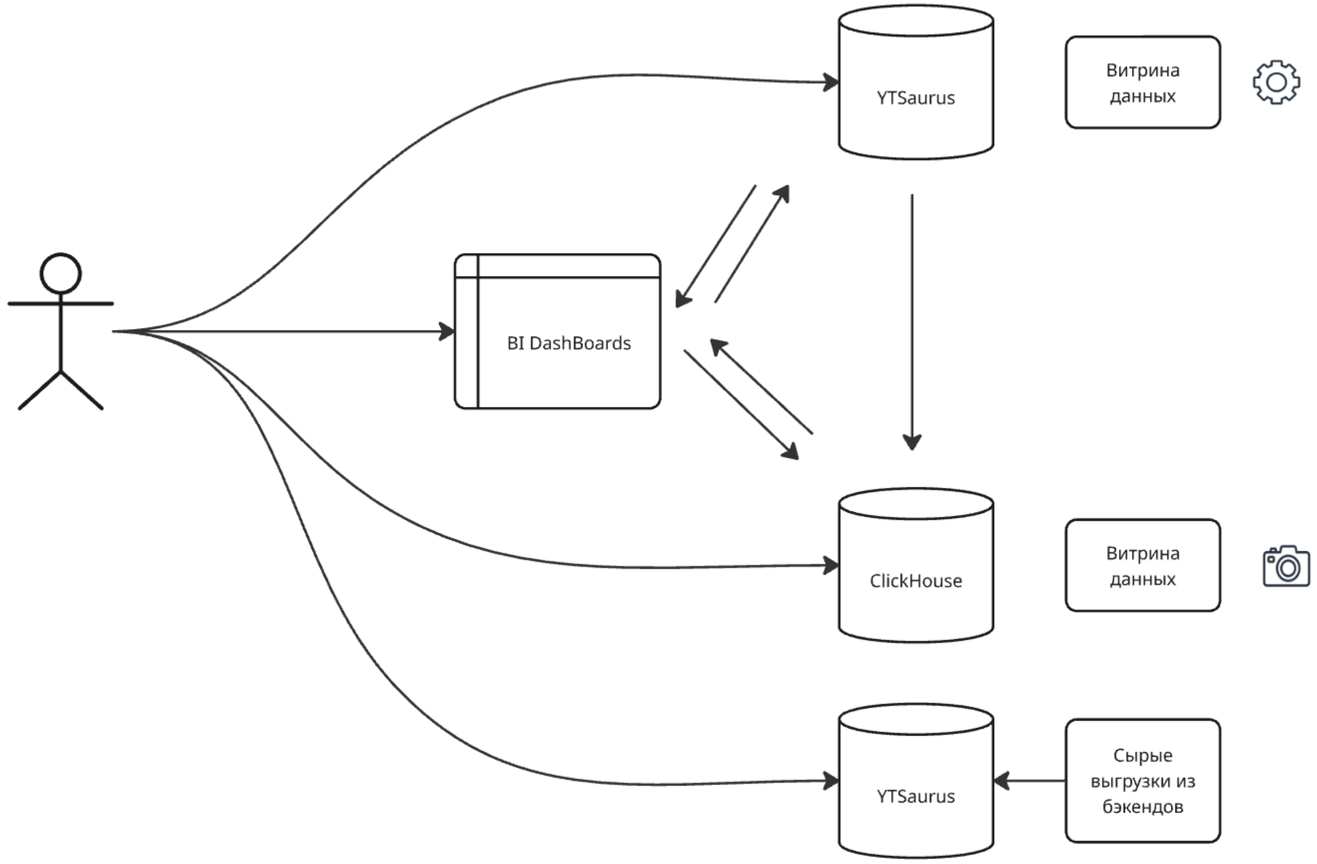
наш опыт

исходная точка, в которой мы находимся

2023 год

- Документацией к объектам хранилища по большей части служат технические артефакты
- Справочной информации нет или она плохо структурирована
- Знания о метриках и использовании витрин хранятся в головах отдельных аналитиков

как устроено потребление данных в Маркете



какие задачи решает пользователь с документацией хранилища

01 Поиск необходимой информации и ответов на вопросы

02 Использование новых атрибутов

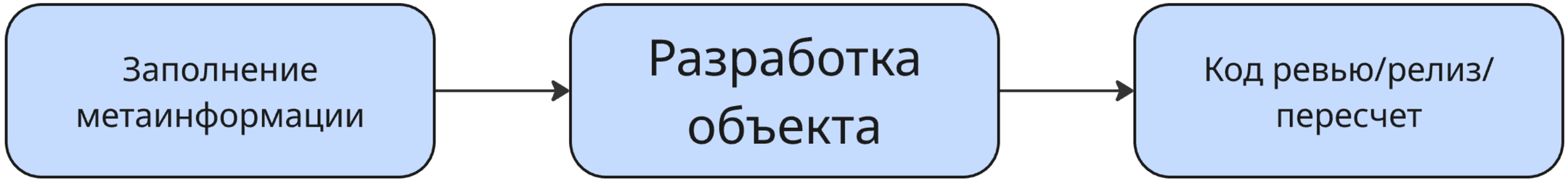
03 Правильный расчет метрик

04 Сделать исследование, ответить на вопрос бизнеса, опираясь на данные

05 Разработка аналитических прототипов

metadata-first

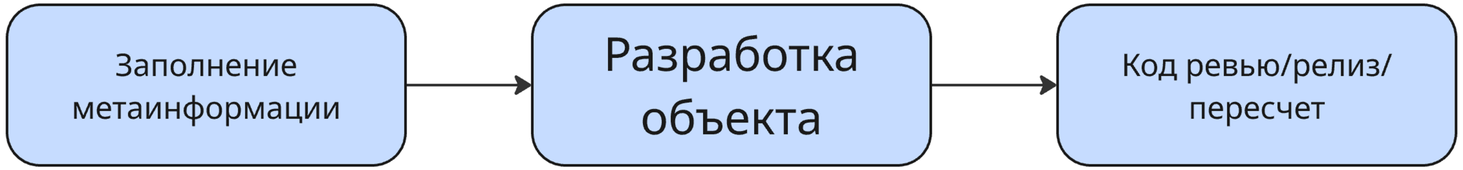
возьмём metadata-first, так как она больше всего описывает процессы хранилища



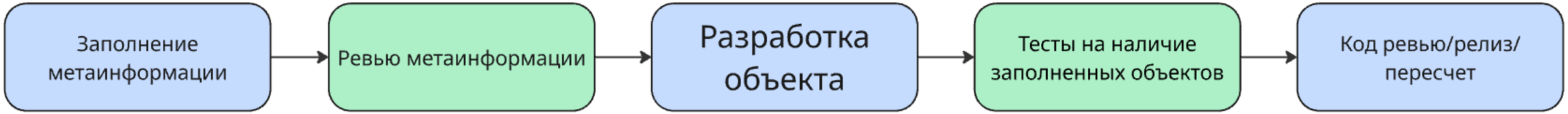
metadata-first

Name	Type	Title	Description
susr1_name	string	-	Признак1
susr2_name	string	-	Признак2
susr3_name	string	-	Признак3
susr4_name	string	-	Признак4
susr5_name	string	-	Признак5

контроль исполнения



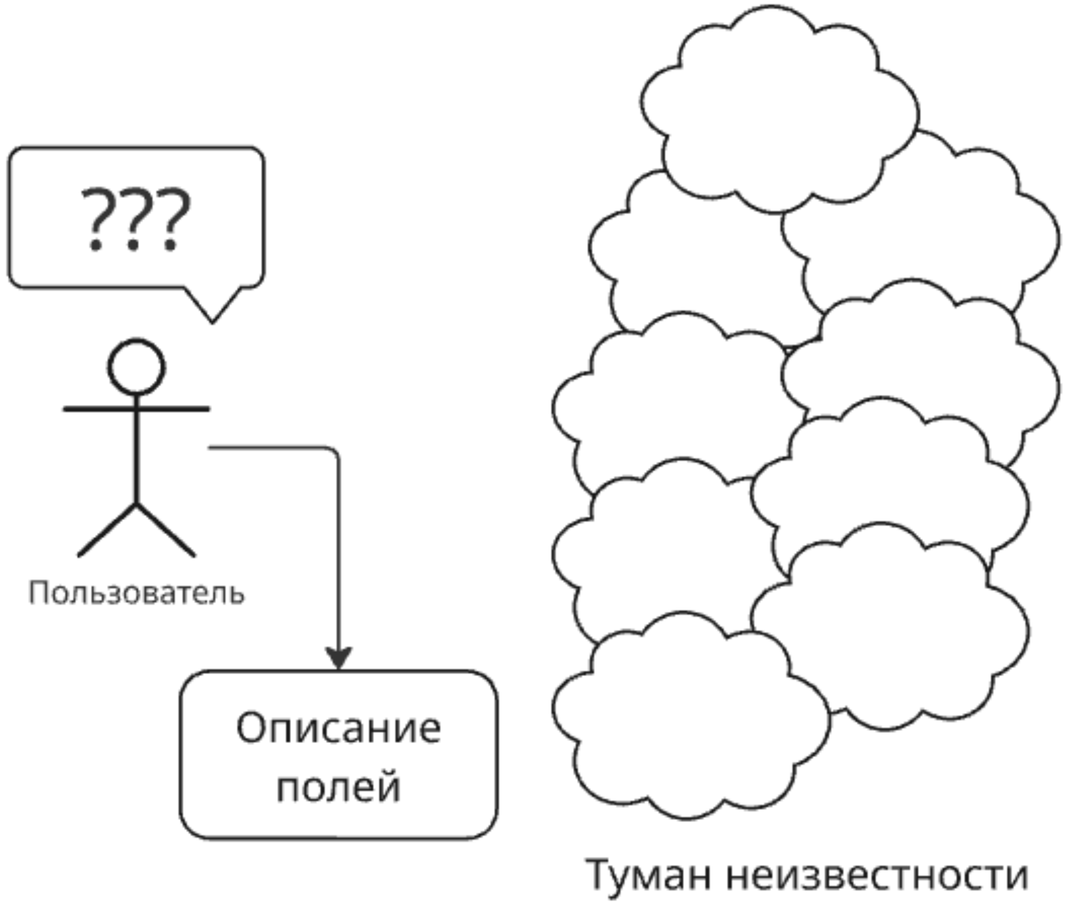
контроль исполнения



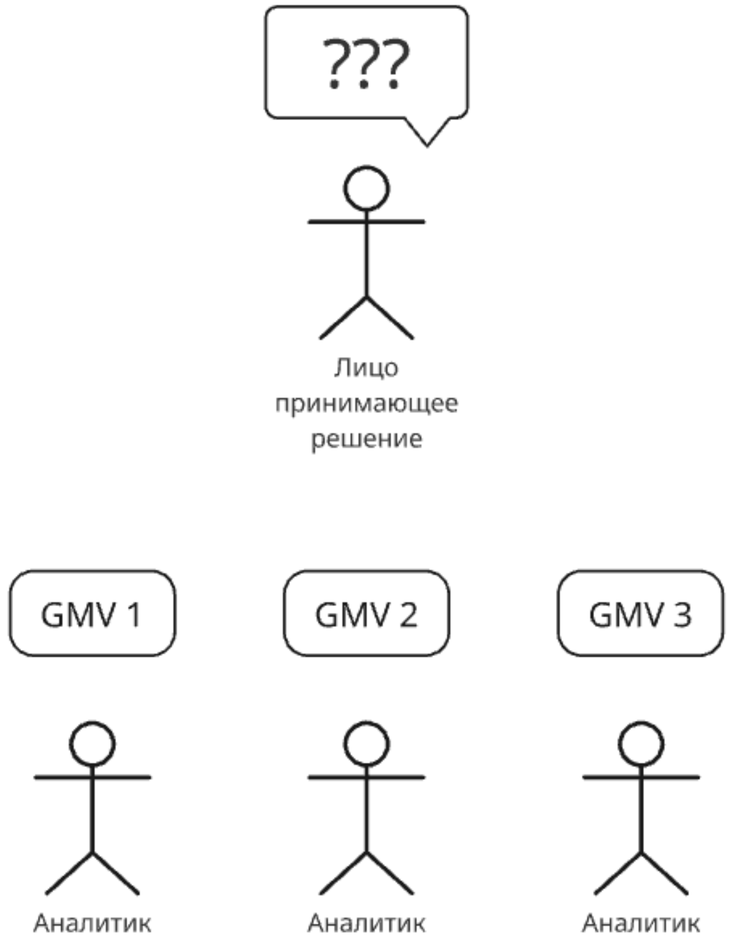
контроль исполнения

	Name	Type	Title	Description
	item_id	int64	Идентификатор позиции заказа	-
	actual_delivery_date_deviation_day_cnt	int64	Отклонение от текущей ПДД (в днях)	Разница между датами фактической и актуальной плановой датой доставки в днях Подробнее про даты доставки в статьях Плановые даты доставки и Фактические дата и время доставки. Подробнее про алгоритм расчёта метрики
	actual_model_id	int64	Идентификатор модели товара, актуальный на момент расчета витрины	Наименование категории (Уровень 7)
	actual_model_name	string	Наименование модели, актуальное на момент расчета витрины	Подробнее про модели по ссылке.
	actual_model_type	string	Тип модели	Меняется редко. Примеры значений: • GURU, • CLUSTER, • PARTNER, • EXPERIMENTAL
	actual_msku	int64	Идентификатор товарной позиции, единица учёта запасов Маркета (MSKU), актуальный на момент расчета витрины	Идентификатор карточки товара на Маркете. Если товар имеет различные модификации, msku присваивается конкретной модификации товара. Подробнее о связях с другими сущностями ассортимента по ссылке. Рассчитан через ssku, но если на позиции заказа не заполнен ssku(или у ssku нет связи с msku), мы заполняем actual_msku так же, как и поле msku. Описание логики ssku см в соответствующем атрибуте.

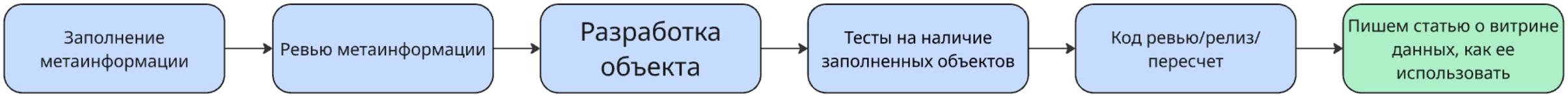
Что там с нашим пользователем



Что там с нашим пользователем



описание объекта



описание объекта

SLA

Готовность в CH: **9:00**

Ссылка

Ссылка на dm_order_item в YT

Ссылка на мету dm_order_item в DC (таблица и схема)

Ссылка на датасет DL

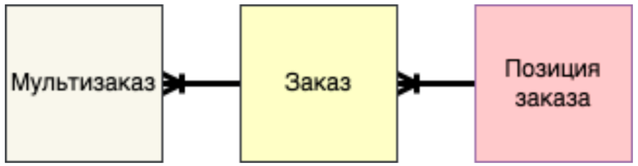
Краткое описание

Витрина данных «Позиции заказов» является основным источником данных о продажах на маркетплейсе Яндекс Маркет.
Продажа – один из этапов пользовательского опыта в Маркете, наступает после инициации покупателем процесса оплаты заказа.

Как использовать

Ключ	**item_id -** идентификатор позиции заказа
Партиционирование	месячное по **msk_item_created_dttm** - дата и время создания позиции заказа (МСК)
Обновление	**ежедневно** – пересчёт за текущий месяц и 2 предыдущих **на выходных** – за последние 12 месяцев

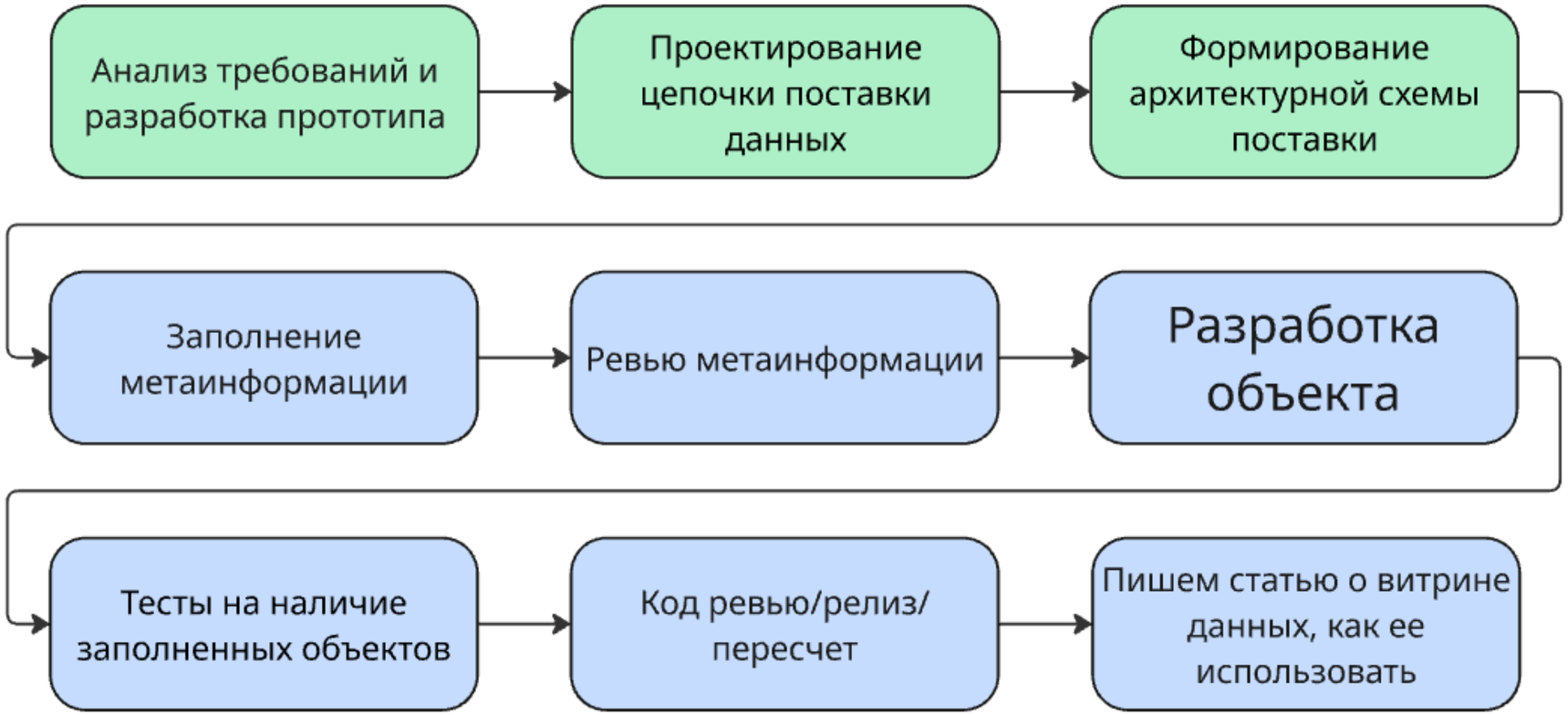
Ключевые сущности витрины данных:



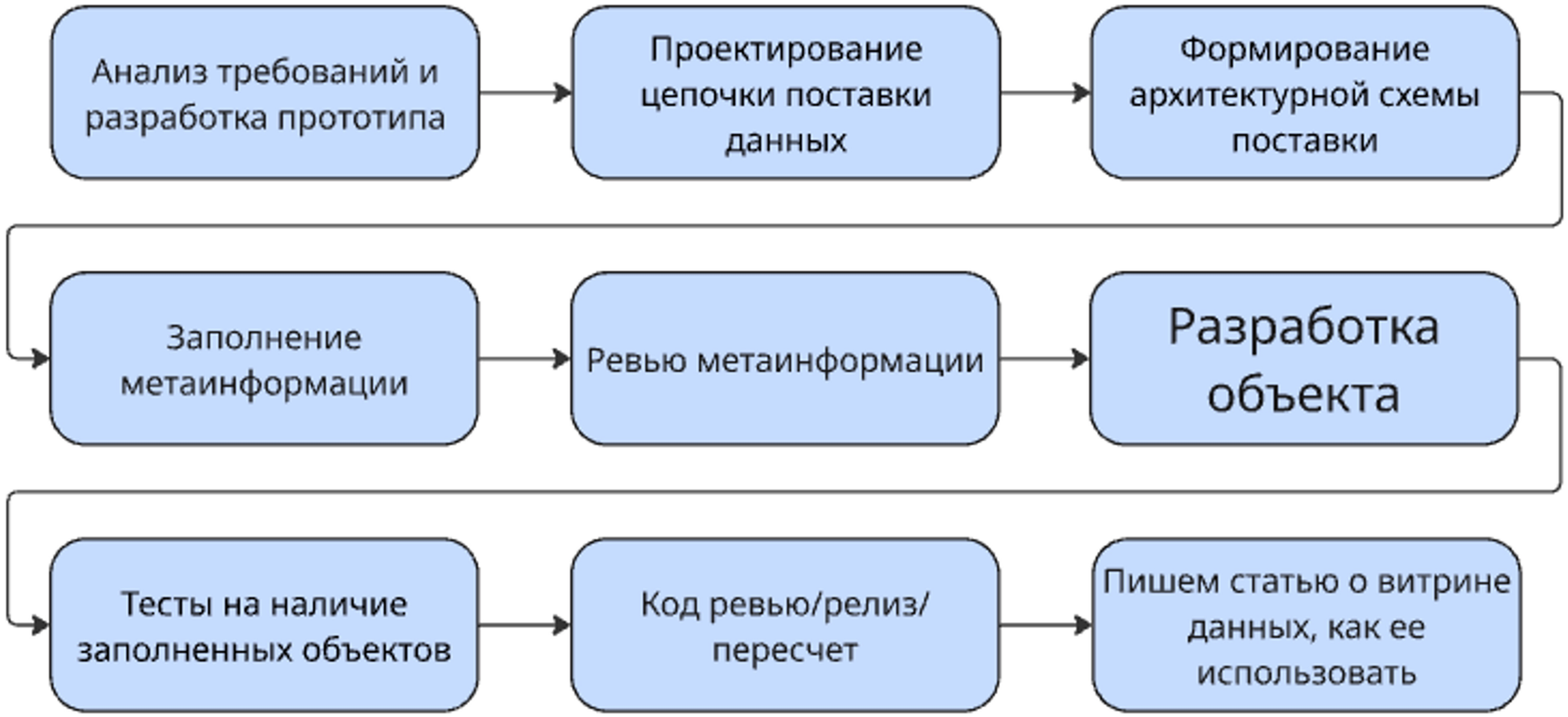
Примеры

когда описывать?

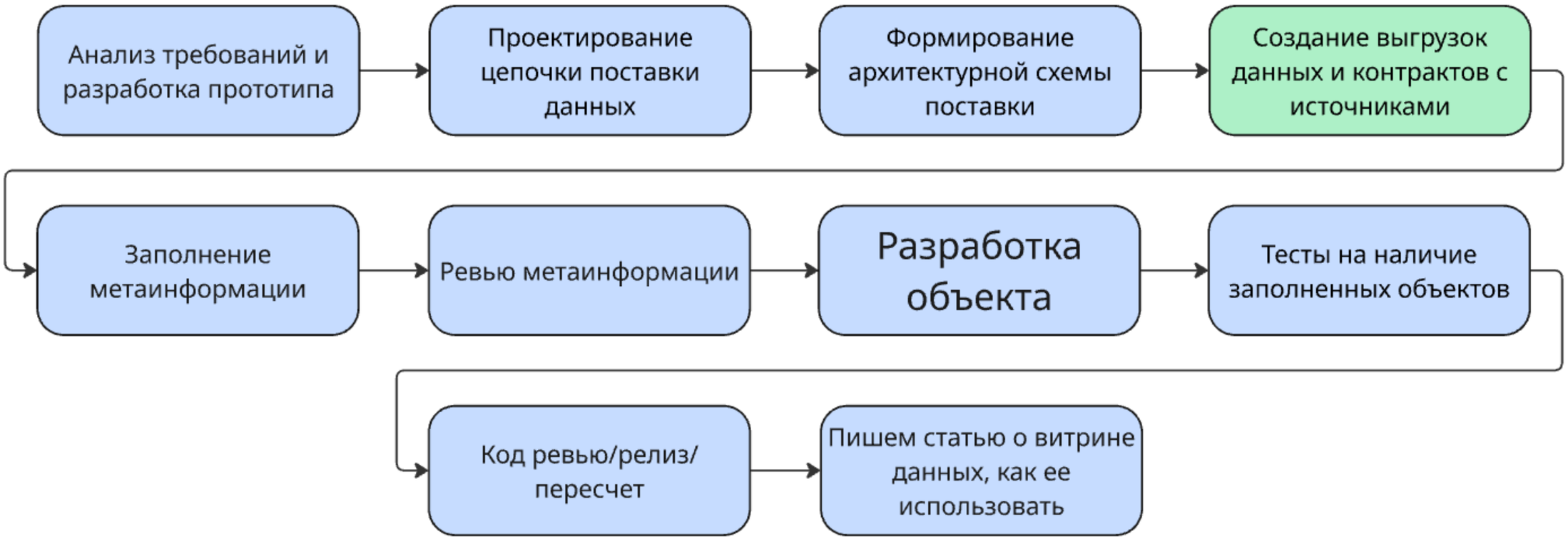
описание требований к объекту



взаимодействие с источниками данных



взаимодействие с источниками данных



взаимодействие с источниками данных

Контракт к команде Ценообразования

Active ∞ - ∞ OK

Overview Targets Monitoring Escalation Subscriptions Revisions Timeline

New revision

Description

Контракт к команде ДЦО команды Ассортимента на поставку логов Цен и параметров товаров конкурентов в DWH.

Поставка необходима для расчета реф. цен по доменам и параметра оплаты только наличными.

Общие договоренности для поставки

1. Система источник - **ДЦО**
2. Представление данных на YT - **Лог сервиса**
3. Объем данных на дату контракта - **60 Gb/день**
4. Прогноз роста объема за 12 месяцев - **18 Tib/день**
5. Чьи вычислительные и ресурсы хранения - **хранение на стороне источника**
6. Партиционирование - **дневное по дате выгрузки (время записи лога)**
7. Отставание данных - T-1
8. Список кластеров YT - **Arnold, Hahn -** имеется в виду, что будет поставлено хотя бы на 1 из двух кластеров
9. Глубина хранения данных - **7 дней**
10. Время готовности - **9:00 UTC+3**
11. Частота загрузок - **ежедневно**
12. Ключ поставки - market_sku, url

Data Entities

Enter text to filter entities... 1 entities

System	Cluster	Path	Owners
yt	hahn	//home/market/production/monetize/dynamic_pricing/parsing/hatter/reference_prices/*	robot-mrkt-money-dco

Collapse details

Contract params

Epics

Version params

Providers: Ценообразование, Разработка

Consumers: Команда Ассортимента DWH Марк..., Разработка

Validity period: ∞ - ∞

результаты

01

Контракты
с источниками данных

результаты

01

Контракты
с источниками данных

02

Принципиальные схемы
построения цепочки расчетов

результаты

01

Контракты
с источниками данных

02

Принципиальные схемы
построения цепочки расчетов

03

Описания объектов
и мета-информация о них

результаты

01

Контракты
с источниками данных

02

Принципиальные схемы
построения цепочки расчетов

03

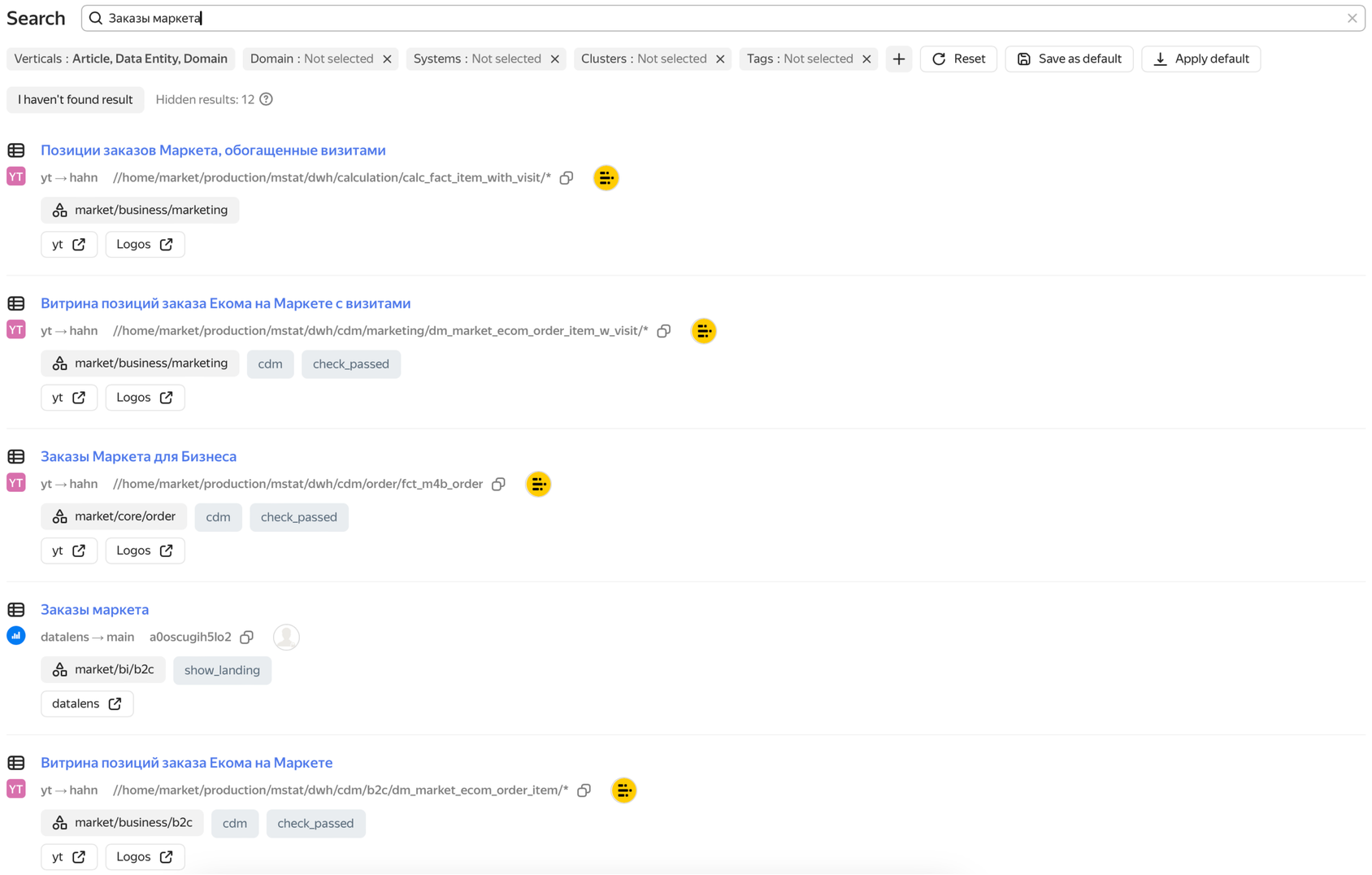
Описания объектов
и мета-информация о них

04

Статьи с примерами
и использованием
о витринах данных

как это закрывает потребности пользователя?

поиск необходимой информации



использование новых атрибутов в хорошо знакомой витрине данных

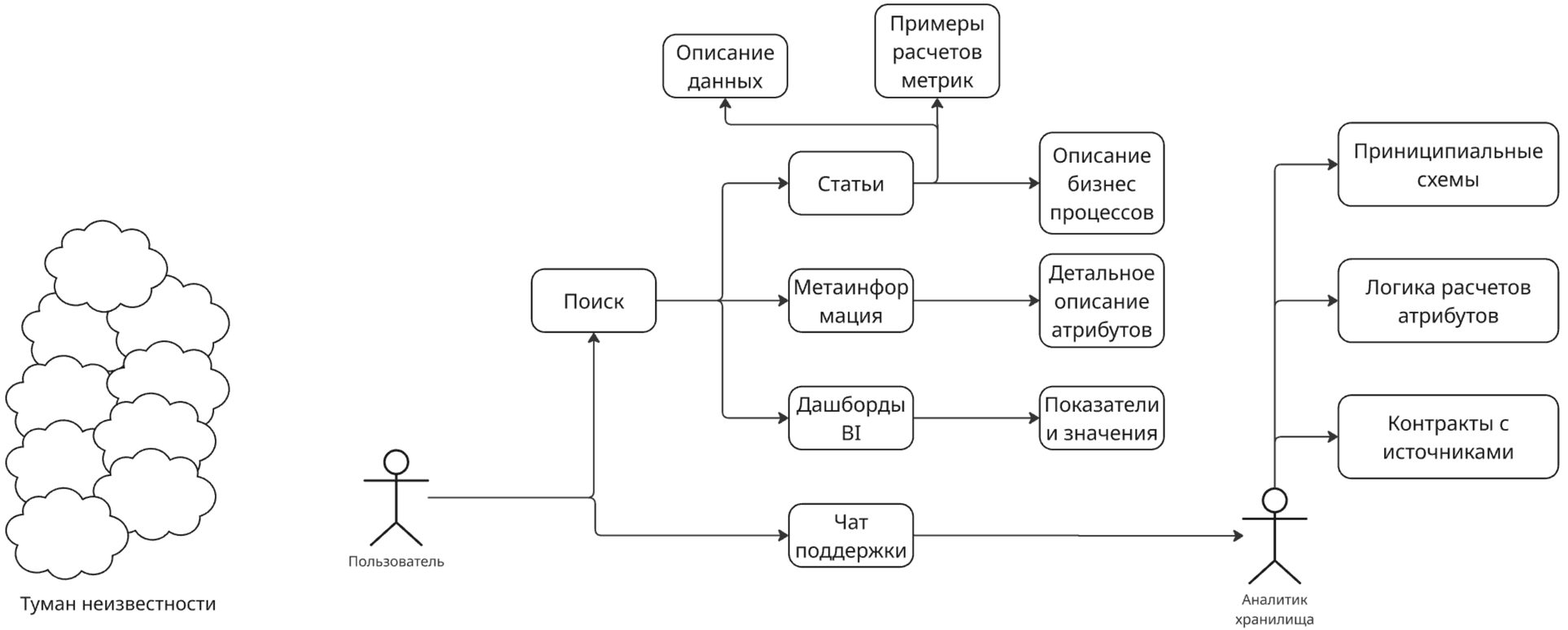
	Name	Type	Title	Description
📌	item_id	int64	Идентификатор позиции заказа	-
	actual_delivery_date_deviation_day_cnt	int64	Отклонение от текущей ПДД (в днях)	Разница между датами фактической и актуальной плановой датой доставки в днях Подробнее про даты доставки в статьях Плановые даты доставки и Фактические дата и время доставки. Подробнее про алгоритм расчёта метрики
	actual_model_id	int64	Идентификатор модели товара, актуальный на момент расчета витрины	Наименование категории (Уровень 7)
	actual_model_name	string	Наименование модели, актуальное на момент расчета витрины	Подробнее про модели по ссылке.
	actual_model_type	string	Тип модели	Меняется редко. Примеры значений: • GURU, • CLUSTER, • PARTNER, • EXPERIMENTAL
	actual_msku	int64	Идентификатор товарной позиции, единица учёта запасов Маркета (MSKU), актуальный на момент расчета витрины	Идентификатор карточки товара на Маркете. Если товар имеет различные модификации, msku присваивается конкретной модификации товара. Подробнее о связях с другими сущностями ассортимента по ссылке. Рассчитан через ssku, но если на позиции заказа не заполнен ssku(или у ssku нет связи с msku), мы заполняем actual_msku так же, как и поле msku. Описание логики ssku см в соответствующем атрибуте.

правильный расчёт метрик над знакомыми данными

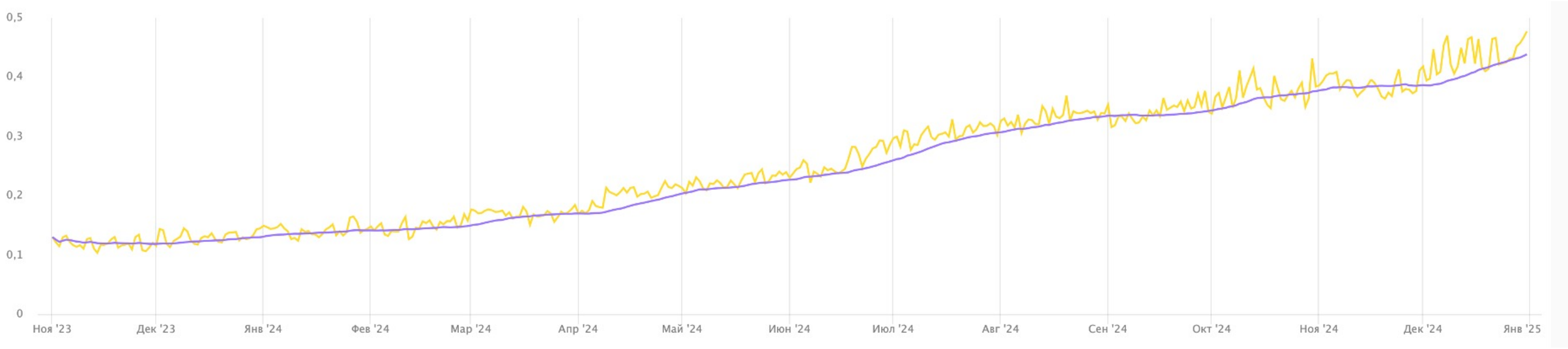
Примеры

- созданные/оформленные/доставленные заказы (кол-во, GMV) за 1 месяц - запрос
  Помним, что партиционирование месячное по дате создания позиции заказа, поэтому метрики по доставленным заказам в начале месяца первой выбранной партиции будут занижены.
  Т.е. если вы смотрите заказы по дате доставки, лучше смотреть одновременно на несколько партиций от даты создания заказа (range). Например, цифры доставленных товаров будут ниже , если по заказу, созданному в январе смотреть только партицию 2024-01. Лучше смотреть на range ‘2024-01’ , ‘2024-03’.;
- найти заказы, проданные по схеме DBS(aka DSBS) - запрос ;
- выделить случаи **возврата**, когда **весь заказ отменён**, но **товары** заказа **не отменены** и в статусе Доставлены - запрос ;

Как там наш пользователь?



выросло использование хранилища данных



раздел

03

ВЫВОДЫ

рекомендации / best practices

01

Отслеживайте качество документации

02

Помните о потребностях пользователей

ВЫВОД

01

не обязательно брать какой-то один подход и следовать только ему, потому что ни один подход не является полным с точки зрения всех ваших процессов

ВЫВОД

02

процессы ведения документации можно внедрять итеративно

ВЫВОД

03

несмотря на то, что пользователи могут не понимать необходимости траты ресурсов хранилища на написание документации, в итоге, вся документация есть благо

ВЫВОД

04

вообще-то, документация — это шире, чем просто заполнить вики

пишите документацию

ГОРОДСКИЕ СЕРВИСЫ

SmartData 2025

готов ответить на вопросы

Паша Колодкин

Руководитель разработки в хранилище данных маркета

раздел

04

Секретные слайды

Почему бы не писать доку с помощью AI?

По каким артефактам писать доку?

А что первичнее, дока или код?