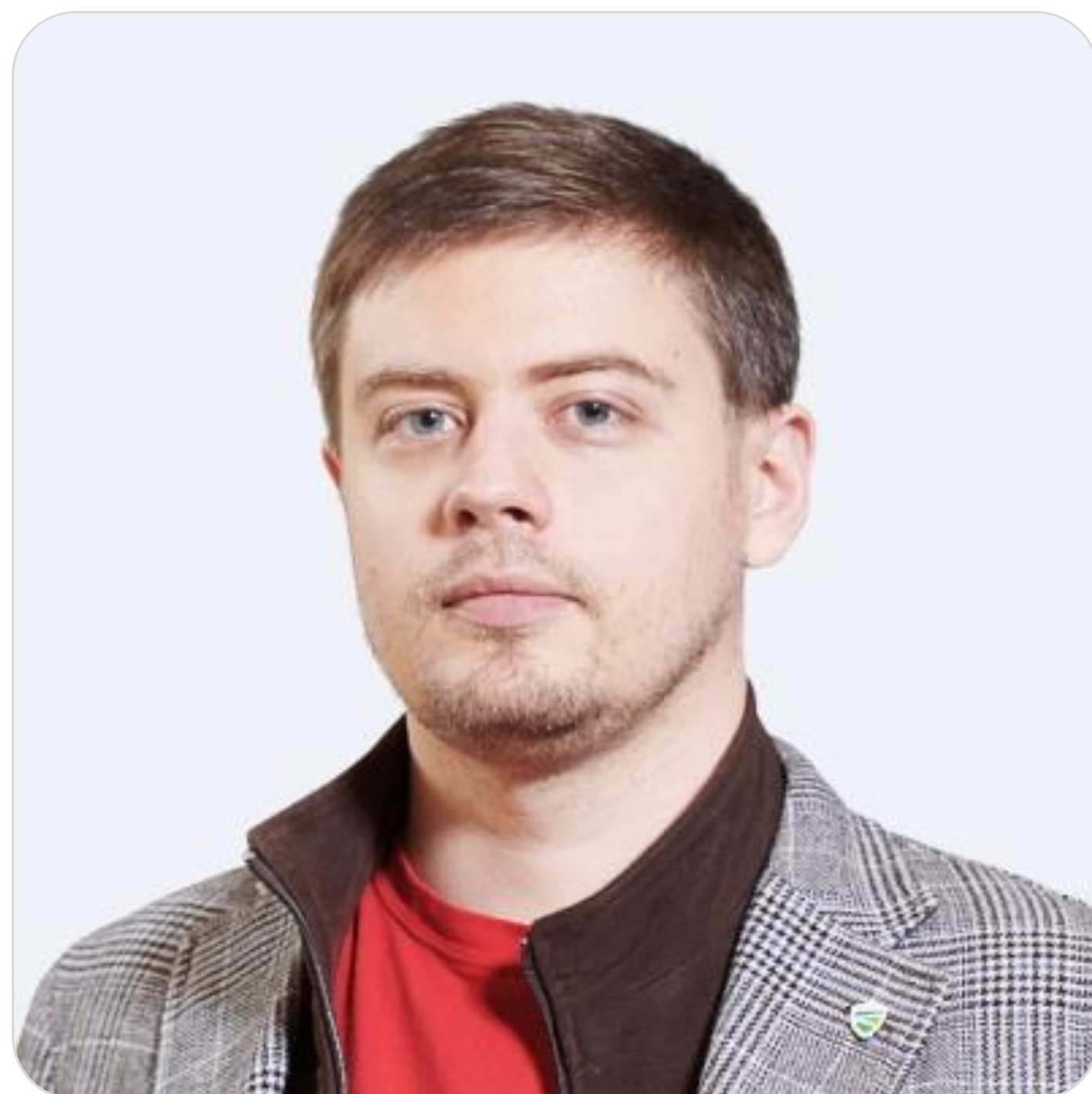


Расширяем границы знаний LLM

Кузьма Лешаков,
архитектор Data Platform,
Yandex Cloud

Александр Константинов,
CTO, Raft DS



Кузьма Лешаков

≈ 8 лет в IT

2022



Два года в Yandex Cloud
Команда Data Platform

2021

UMA.TECH

Uma.Tech

Ведущий инженер данных

2019



Clover Group

Инженер данных

2017








The Linux Foundation

Стажёр-разработчик



**Александр
Константинов**

≈ 11 лет в IT

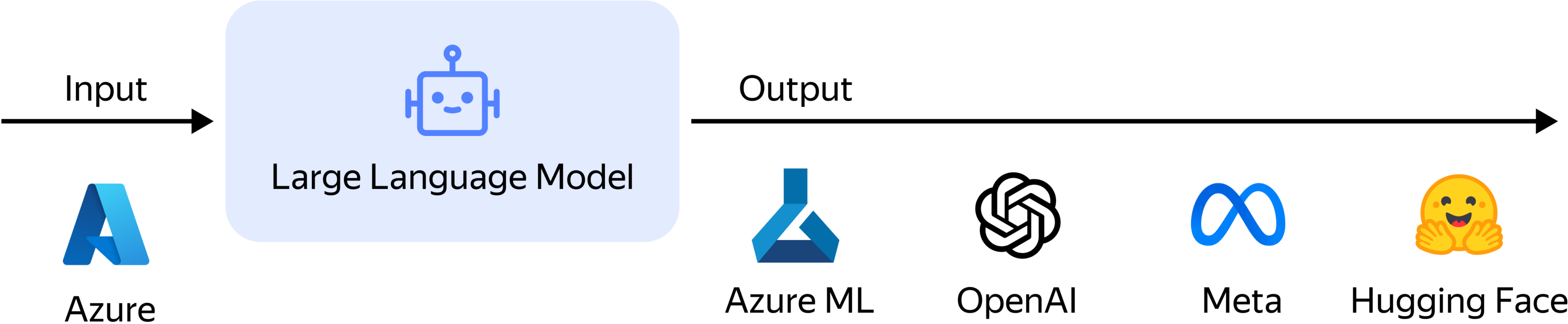
- 2023  Raft DS
CTO
- 2023  One Rail
Senior SDE
- 2019  Evernote
Lead SDE
- 2018  Valant
Architect
- 2014  Microsoft
SDE

О чём поговорим

1. Обзор языковых моделей и возможностей расширения их знаний
2. Расширение знаний LLM моделей на примере YandexGPT
3. Архитектура RAG
4. Технические аспекты при реализации RAG
5. Оценка качества работы RAG-приложения
6. Выводы

1. Обзор языковых моделей и возможностей расширения их знаний
2. Расширение знаний LLM моделей на примере YandexGPT
3. Архитектура RAG
4. Технические аспекты при реализации RAG
5. Оценка качества работы RAG-приложения
6. Выводы

Обзор языковых моделей



Языковые модели развиваются в Яндексе уже не первый год

2021



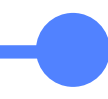
В июне — сервис
«Зелибоба»/«Балабоба»

2022



В июне Яндекс выложил одну
из крупнейших языковых
моделей — YaLM 100B

2023



Яндекс анонсировал выпуск
новой модели YandexGPT API

Лавка: пример подсказок

The screenshot shows a chat window with the following elements:

- User message:** "добрый день! что дает эта галочка7" (18:59)
- System message:** "Здравствуйте! Передали вопрос специалисту. Он напишет вам в течение 30 минут и всё подробно расскажет." (19:00)
- User message:** "Товарная галерея на поиске" (19:00)
- User message:** "я не загружаю фиды, что мне даст галочка?" (19:02)
- User message:** "image.png" (19:02)
- System tooltip:** A white box with rounded corners containing:
 - Text: "Спасибо за ожидание."
 - Text: "Галочка "Товарная галерея" означает, что вы хотите размещать товарные предложения через Товарную галерею, которая находится под поисковой строкой."
 - Text: "Подробнее о Товарной галерее вы можете прочитать в Справке - раздел ["Товарная галерея"] (<https://yandex.ru/support/direct/dynamic-text-ads/product-gallery.html>).
 - Buttons: "Использовать в ответе" and "Нерелевантно"
- Input area:** Two blue buttons with speech bubble icons: "Спасибо за ожидан..." and "Спасибо, что подожд...". Below them is a text input field with the placeholder "Написать в чат".

Лавка: генерация новых атрибутов

Яндекс Голубь

- Товары
- Рецепты
- Комбинации
- Мастер-категории
- Фронт-категории
- Инфомодели
- Атрибуты
- Группы атрибутов
- Витрина
- Правила

Version: dev
NDA Конфиденциально
© 1997–2023 ООО «ЯНДЕКС»

Некоторые правила ссылаются на недоступные им атрибуты [Показать правила](#)

Просмотр товара Закреть Выбранный регион: Россия

.. / ФУД / Хлеб, торты, пирожные / Пирожные промышленные

Товар «Пирожное «Медовейник» «Из Лавки», 120 г»

Активный · Не определён · Есть на стоках · ID: 10020023 · Просмотров за 3 дня: 584 · Полнота: 100%, 17 из 17 · Инфомодель: ФУД


[Параметры](#) [Фронт-категории](#) [История](#) [Debug панель](#)

Системные атрибуты

Статус	<input checked="" type="radio"/> Активный <input type="radio"/> Неактивный
Мастер категория	Пирожные промышленные
Штрихкод <small>barcode</small>	4627119671554 🗑️ Добавить штрихкод
Длинное название <small>longName</small>	Пирожное «Медовейник» «Из Лавки»
Короткое название (loc) <small>shortNameLoc</small>	Пирожное Медовейник Из Лавки
Маркетинговое количество в упаковке <small>markCount</small>	120
Единица маркетингового количества в упаковке (список) <small>markCountUnitList</small>	г
Тип номенклатуры <small>nomenclatureType</small>	Товар ⌵ <small>Неизменяемое значение атрибута</small>

Изображение товара

image 📌



[📄](#)




1 2 +


Тэги и стикеры

Стикеры на фото
photoStickers

[Выберите значение](#) ★

Суммаризация ОТЗЫВОВ


09:41   

Яндекс Станция 2 


Яндекс Станция 2 - умная колонка с Алисой




4,8 Умные колонки

Основное **Цены** **Обзоры** **Отзывы**



Цвет товара: **Кобальт**



09:41   

● Самое важное из 10 663 отзывов


Собрала нейросеть YandexGPT

За что хвалят

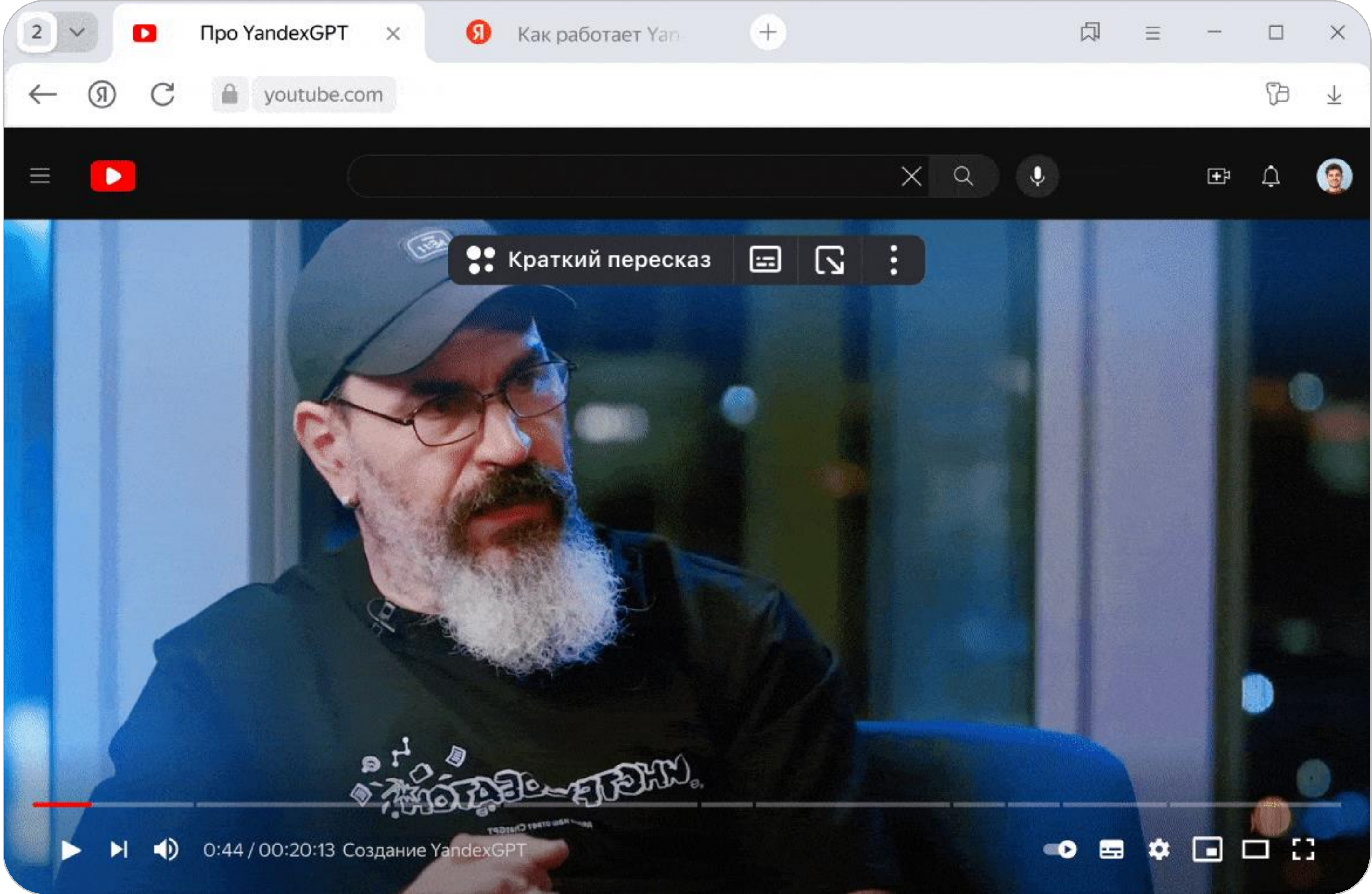
Отличное звучание	6 062
Совместимость с умным домом	1 052
Широкий спектр возможностей	1 034
Хорошая реакция на голос	784

За что ругают

Ограниченные возможности управления	407
Отсутствие аккумулятора	176

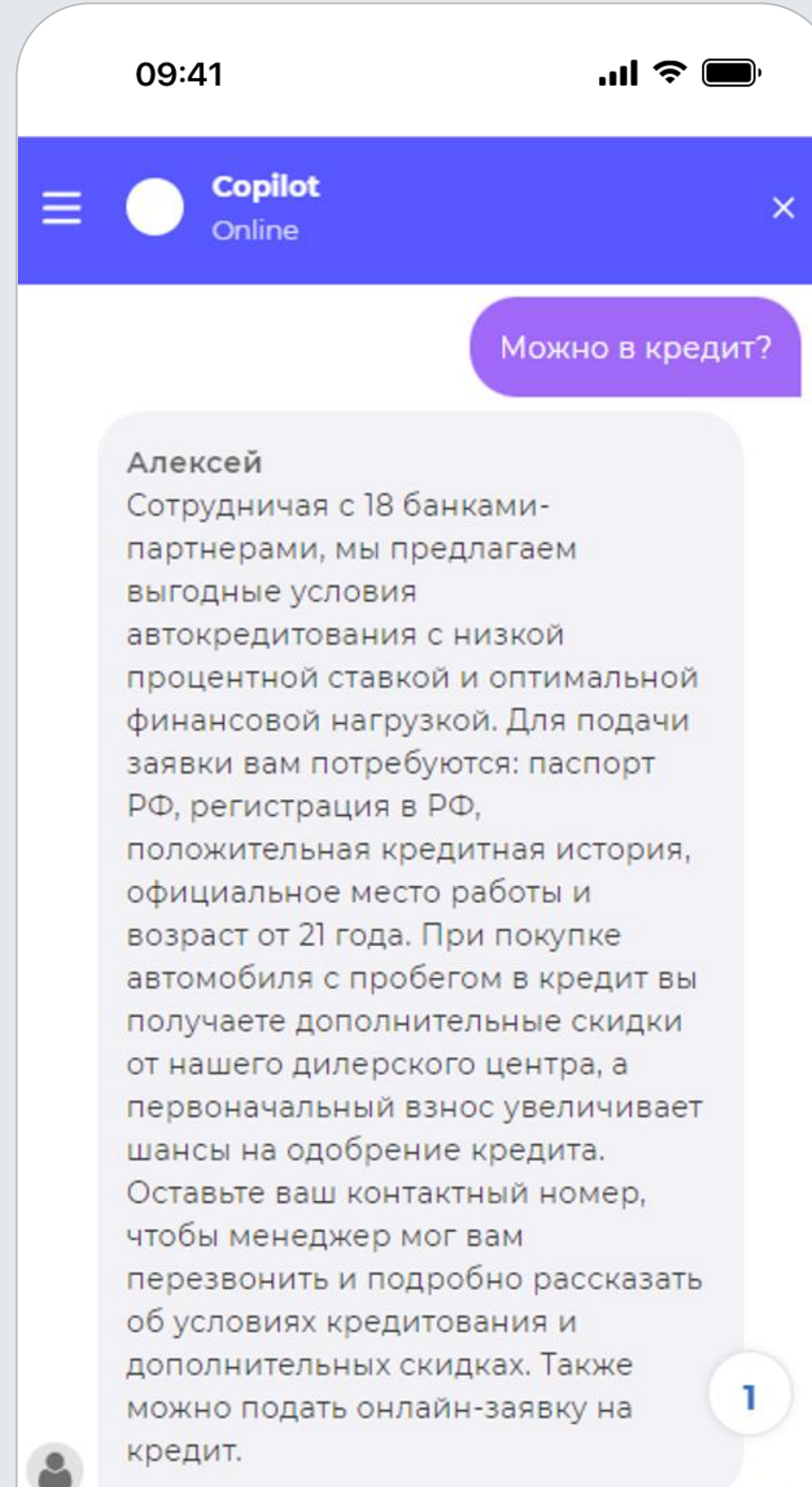
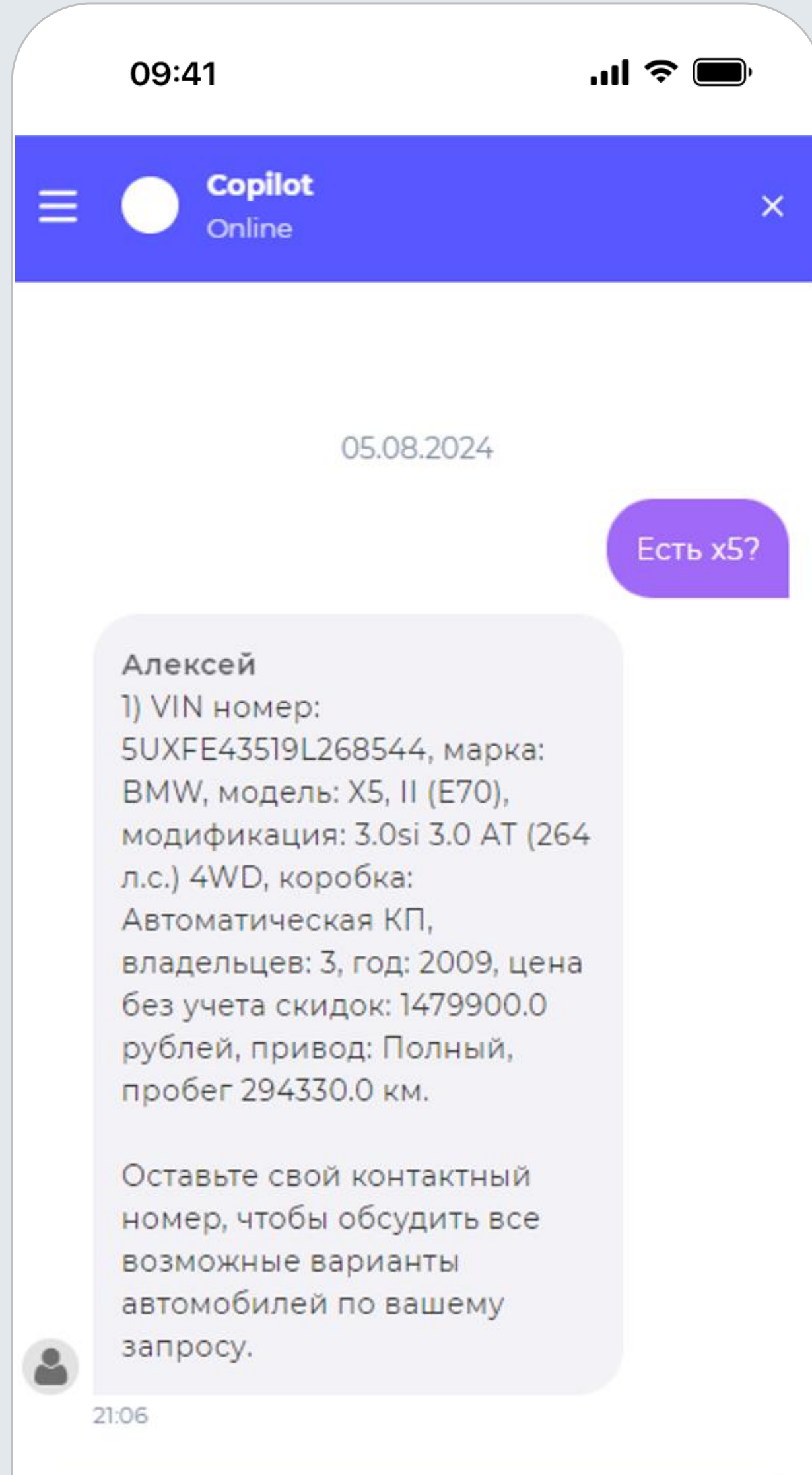
 Хорошее обобщение  Плохое

Суммаризация видео



475 лет
сэкономлено
за месяц

Чат-бот для автоматизации продаж автомобилей



Оценка качества работы колл-центра

Используйте возможности AI для оценки звонков и аудиозаписей

Мы соединили передовые решения для преобразования аудио в текст с оценкой разговора с помощью AI.

Объем анализируемых звонков

7% → 100%



Скорость принятия решений

14 – 30 дней → 1 день



Стоимость оценки разговора

снижение до 82%



Взаимодействие с YandexGPT API

Консоль: instruct-режим

cloud-adolotov DE default YandexGPT / Промт-режим [Документация](#)

Промт-режим Сервис YandexGPT логирует запросы, чтобы улучшить качество модели. Не передавайте в запросах чувствительную информацию и персональные данные. [Очистить](#)

Инструкции ? Текст Шаблоны

Ты руководитель команды разработки продукта

Запрос ?

Составь план работ на всю команду на неделю вперёд по SCRUM

Температура ? 0.01 1

Использовано 21 токен ? [Узнать ответ](#)

YandexGPT 298 ?

1. Встреча с командой разработчиков:
 - Обсуждение текущего состояния проекта и планирование работы на неделю.
 - Определение приоритетов задач и распределение их между членами команды.
1. Разработка нового функционала:
 - Работа над задачей №1 - Добавить новую функцию "Поиск по каталогу".
 - Работа над задачей №2 - Исправить ошибку в модуле "Корзина".
1. Тестирование и отладка:
 - Тестирование нового функционала "Поиск по каталогу" и исправление ошибок.
 - Тестирование модуля "Корзина" и исправление обнаруженных ошибок.
1. Подготовка документации:
 - Создание руководства пользователя для новой функции "Поиск по каталогу".
 - Обновление документации по модулю "Корзина".

Обращение через API

```
{
  'modelUri': 'ds://<id_дообученной_модели>',
  'completionOptions':{
    'stream': False,
    'temperature': 0.1,
    'maxTokens': '1000'
  },
  'messages':[
    {
      'role': 'system',
      'text': '<текст вашей инструкции>'
    },
    {
      'role': 'user',
      'text': 'Теперь Яндекс Карты не только предупреждают о платных
маршруте, но и показывают стоимость проезда.
      Это позволяет водителям заранее оценить расходы
      или выбрать маршрут без платных дорог, если это возможно...'
    }
  ]
}
```

участках на

Обращение через LangChain

```
from langchain.chains import LLMChain
from langchain_community.llms import YandexGPT
from langchain_core.prompts import PromptTemplate

# API Reference:
# - LLMChain
# - YandexGPT
# - PromptTemplate

template = "What is the capital of {country}?"
prompt = PromptTemplate.from_template(template)

llm = YandexGPT()

llm_chain = LLMChain(prompt=prompt, llm=llm)

country = "Russia"

llm_chain.invoke(country)
```

YandexGPT API в Public Preview!

Доступные модели для всех пользователей

Модели

YandexGPT API Lite

YandexGPT API Pro

Эмбеддинги

Токенизатор

Суммаризация

На базе YandexGPT API Lite

Инференс

Синхронно

Стандарт

Асинхронно

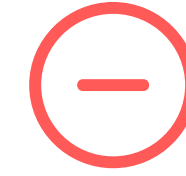
Know-how

LLM умеет: хорошо vs плохо



LLM умеет делать хорошо

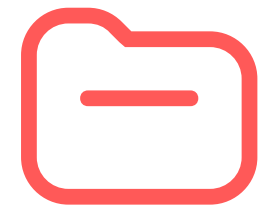
- Генерировать и редактировать тексты
- Отвечать на вопросы по базе знаний
- Чатиться, общаться
- Обобщать и интерпретировать данные
- Классифицировать и определять тональность текста
- Выделять сущности из текста
- Переводить тексты



Есть решения лучше LLM

- Ранжирование результатов поиска, рекомендации товаров, услуг и т. д.
- Решение задач, подразумевающих строгую логику
- Решение задач с низкой толерантностью к ошибке и высокой стоимостью проверки

Ограничения LLM-моделей



Ограниченность
знаний

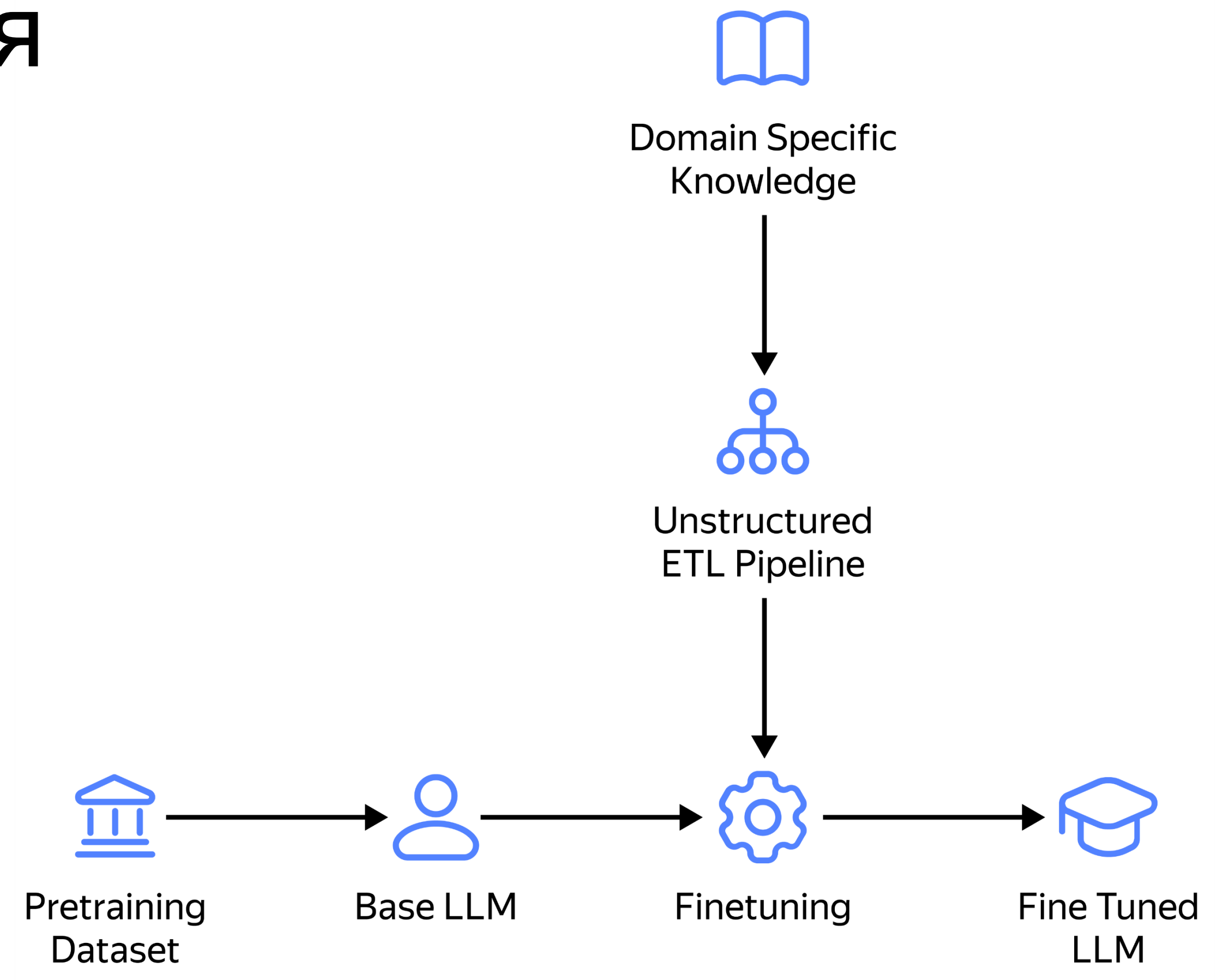


Тенденция
к воспроизведению ошибок

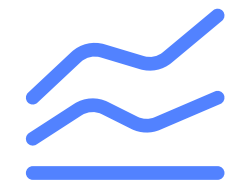
1. Обзор языковых моделей и возможностей расширения их знаний
2. Расширение знаний LLM моделей на примере YandexGPT
3. Архитектура RAG
4. Технические аспекты при реализации RAG
5. Оценка качества работы RAG-приложения
6. Выводы

Возможности расширения их знаний

1. Fine-tuning
2. Retrieval-augmented generation (RAG)

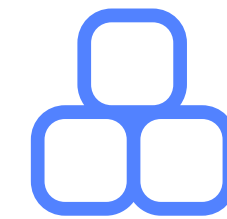


Какие есть способы работы с внешними источниками информации



Fine-tuning модели

Для автоматического
определения контекста /
предметной области



Retrieval Augmented Generation (RAG)

Ответы по конкретным
документам

RAG vs Fine-tuning vs комбинация

Аспект	RAG	P-tuning	Комбинация
Динамические данные	✓	✗	✓
Статические данные	✓	✓	✓
Внутренние данные	✓	✗	✓
Уменьшение галлюцинаций	✓	✓	✓
Прозрачность генерации	✓	✗	✓
Тонкая настройка под узкую задачу	✗	✓	✓
Голос бренда	✗	✓	✓

Реализация в DataSphere

Чтобы дообучить модель YandexGPT API, нужно подготовить файл с формате JSON. Файл надо сохранить в кодировке UTF-8:

```
[
  {
    "request": [
      {
        "role": "system",
        "text": "Текст инструкции"
      },
      {
        "role": "user",
        "text": "Текст запроса"
      }
    ],
    "response": "Ожидаемый ответ"
  }
]
```

The screenshot shows the 'Дообучение фундаментальной модели' (Fine-tuning fundamental model) page in the Yandex DataSphere interface. The page is for the organization 'organization-adolotov' and project 'test1'. It includes a sidebar with navigation options like 'Сообщества', 'Проекты', and 'Фундаментальные модели'. The main content area has a header with an information icon and text: 'Вы можете дообучить модель YandexGPT на собственных данных по методу Fine-tuning. После дообучения результат будет храниться в DataSphere и доступен только вам в DataSphere и в YandexGPT в консоли управления и API.' Below this are form fields for 'Имя*' (Name), 'Описание' (Description), and 'Темп обучения' (Learning rate) with a slider set to 0.001. The 'Данные для дообучения' (Data for fine-tuning) section shows a file upload area with a red dashed border and a message: 'Поле не заполнено JSON, максимальный размер файла — 100 МБ' (Field is empty JSON, maximum file size is 100 MB), with a 'Выбрать файл' (Choose file) button. At the bottom, there is an 'Инструкция' (Instructions) field.

Реализация в DataSphere

```
{
  "request": [
    {
      "role": "system",
      "text": "Твое имя Женя, полное имя Евгений Нейроныч.
\nТы отвечаешь от лица мужского рода. \nТы робот. \nТы
говоришь коротко и емко. \nТы был создан в Перми. \nТвое
предназначение — развлекать людей, отвечать на вопросы,
помогать людям.\nТы эксперт в сфере ЖКХ."
    },
    {
      "role": "user",
      "text": "Как тебя зовут?"
    }
  ],
  "response": "Меня зовут Женя. Полностью — Евгений
Нейроныч."
}
```

Реализация в DataSphere

```
{  
  "request": [  
    {  
      "role": "system",  
      "text": "Твое имя Женя, полное имя Евгений Нейроныч.  
\\nТы отвечаешь от лица мужского рода. \\nТы робот. \\nТы  
говоришь коротко и емко. \\nТы был создан в Перми. \\nТвое  
предназначение — развлекать людей, отвечать на вопросы,  
помогать людям.\\nТы эксперт в сфере ЖКХ."  
    },  
    {  
      "role": "user",  
      "text": "Где ты был создан?"  
    }  
  ],  
  "response": "В Перми."  
}
```

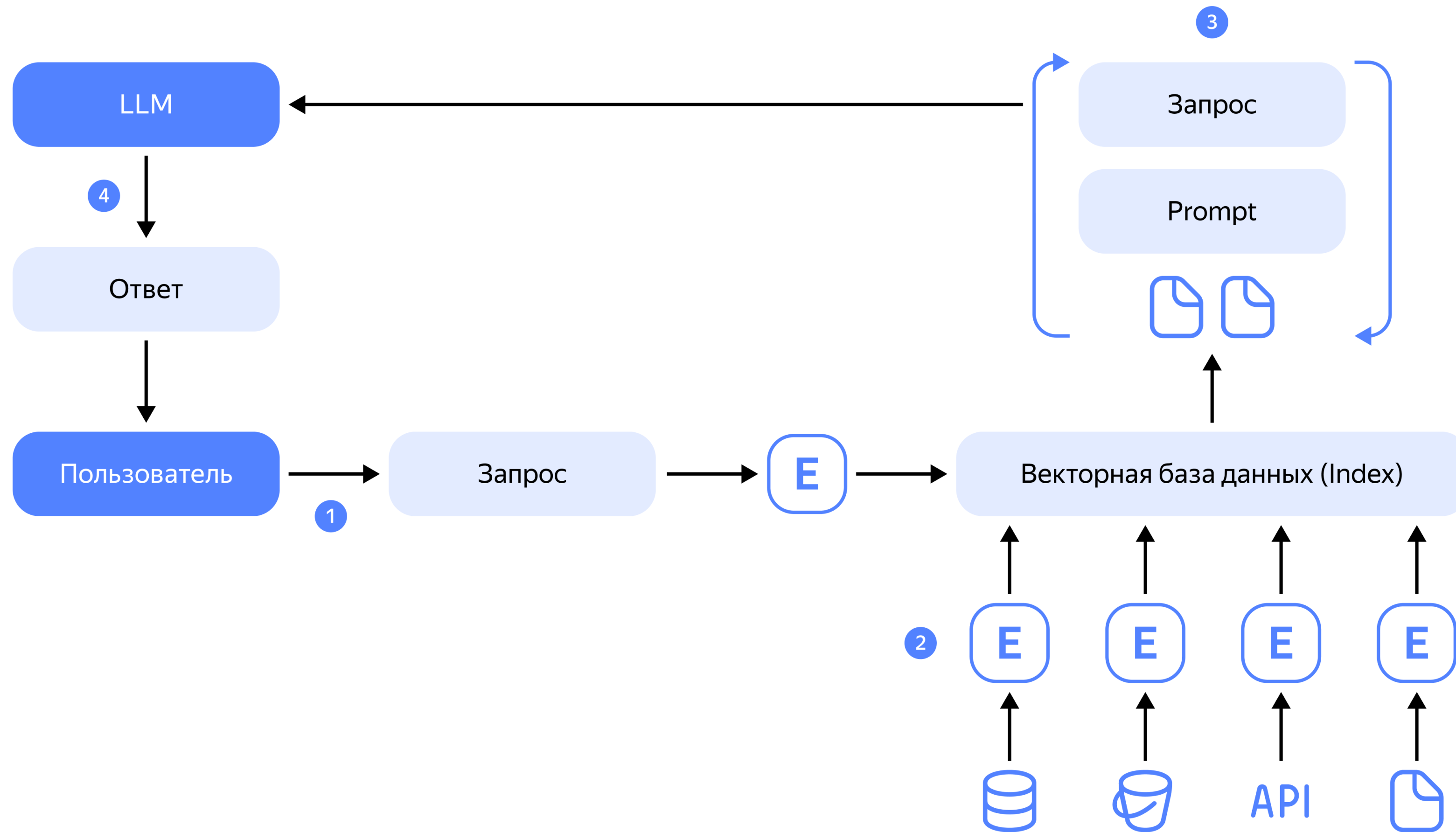
Реализация в DataSphere

```
[  
  {  
    "text": "у меня все в порядке",  
    "нейтральный": 1,  
    "эмоциональный": 0  
  },  
  {  
    "text": "у меня все классно получилось",  
    "нейтральный": 0,  
    "эмоциональный": 1  
  }  
]
```

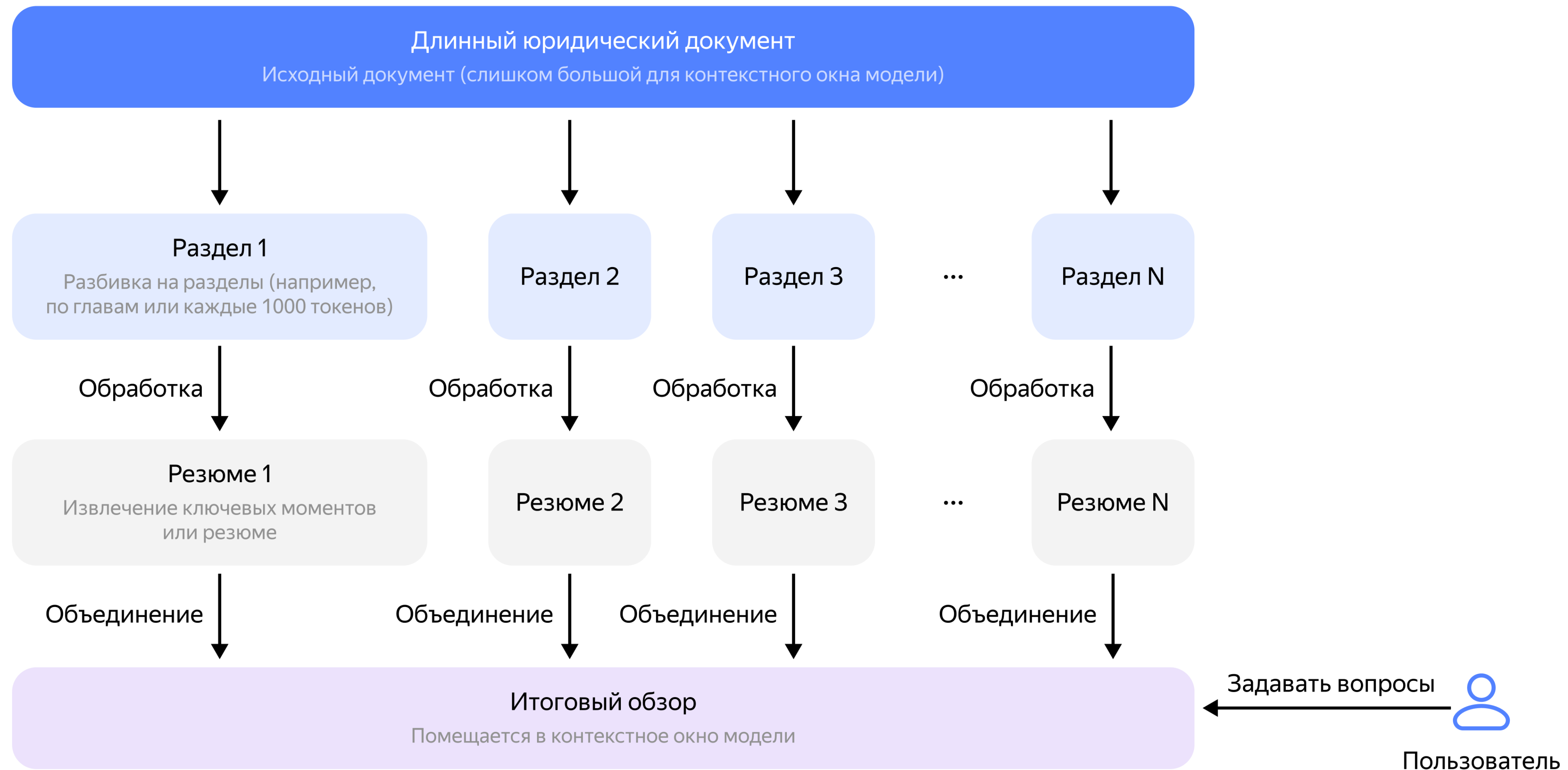
Реализация в DataSphere

```
{  
  "computer_science": 0,  
  "physics": 0,  
  "mathematics": 1,  
  "statistics": 0,  
  "quantitative_biology": 0,  
  "quantitative_finance": 0,  
  "text": "Title: Essentially Finite Vector Bundles on Normal Pseudo-  
proper Algebraic Stacks\nAbstract: Let  $X$  be a normal, connected  
and projective variety over an algebraically\nclosed field  $k$ . It is  
known that a vector bundle  $V$  on  $X$  is essentially\nfinite if and  
only if it is trivialized by a proper surjective morphism  
 $f: Y \rightarrow X$ . In this paper we introduce a different approach to  
this problem which\nallows to extend the results to normal,  
connected and strongly pseudo-proper\nalgebraic stack of finite  
type over an arbitrary field  $k$ .\n"  
}
```

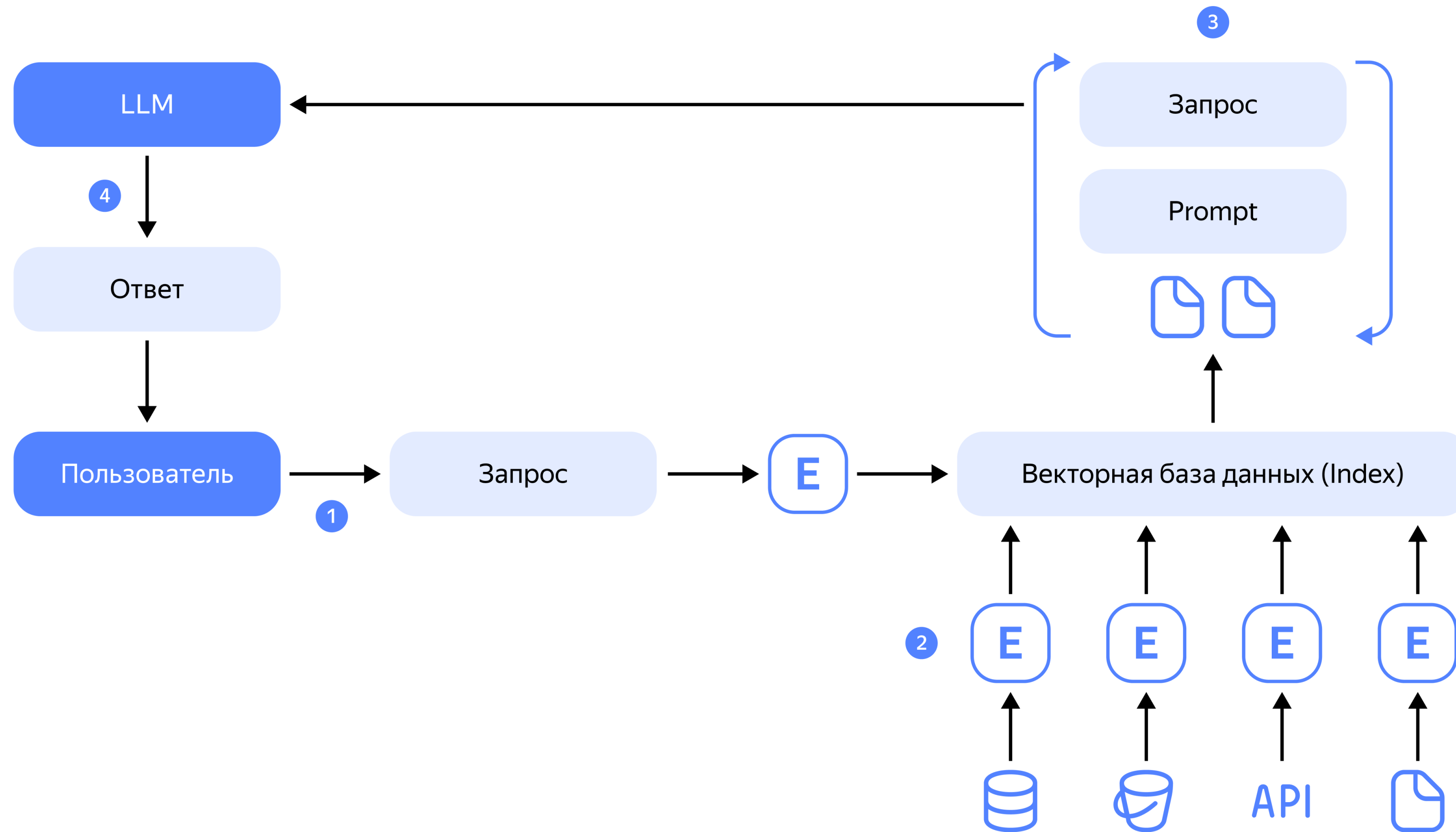

Архитектура RAG



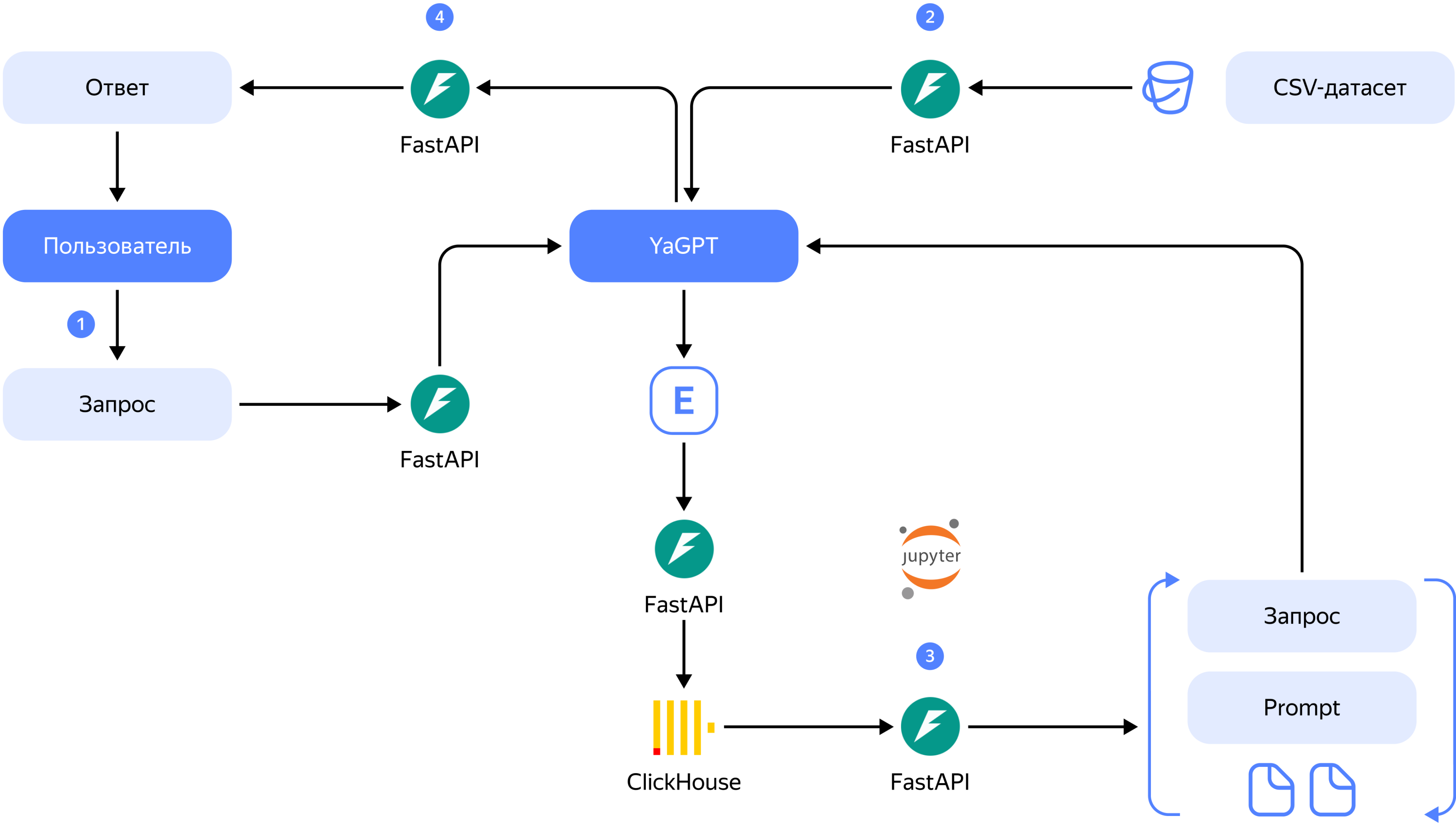
Архитектура RAG — контекстное окно модели



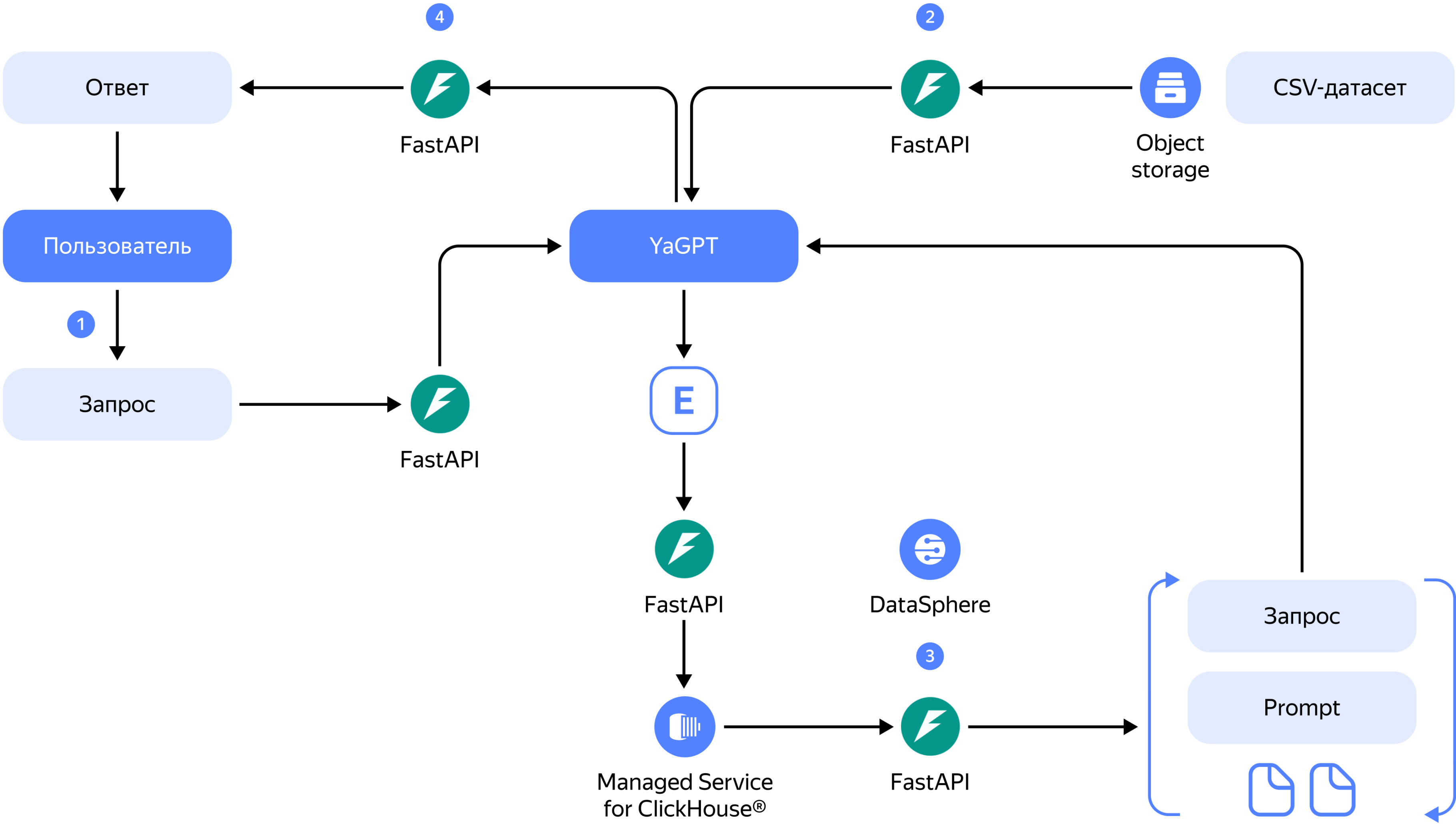
Архитектура RAG



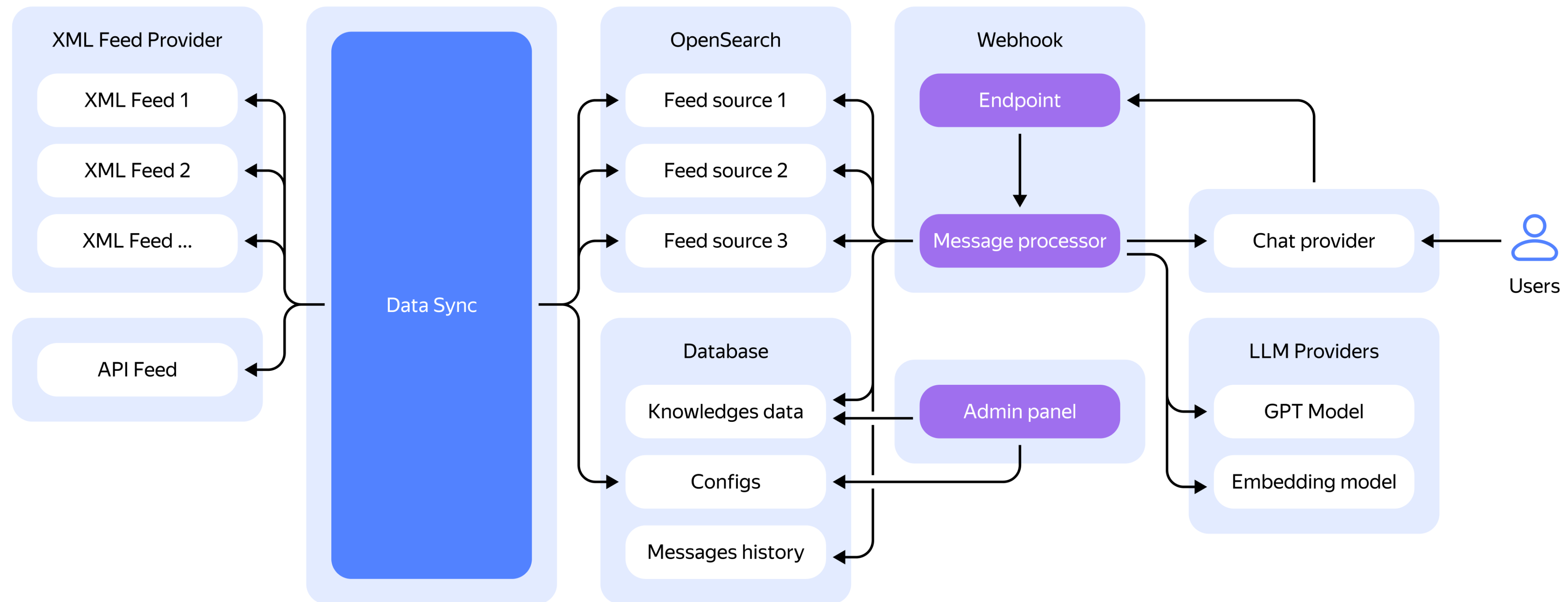
Архитектура RAG on-premise



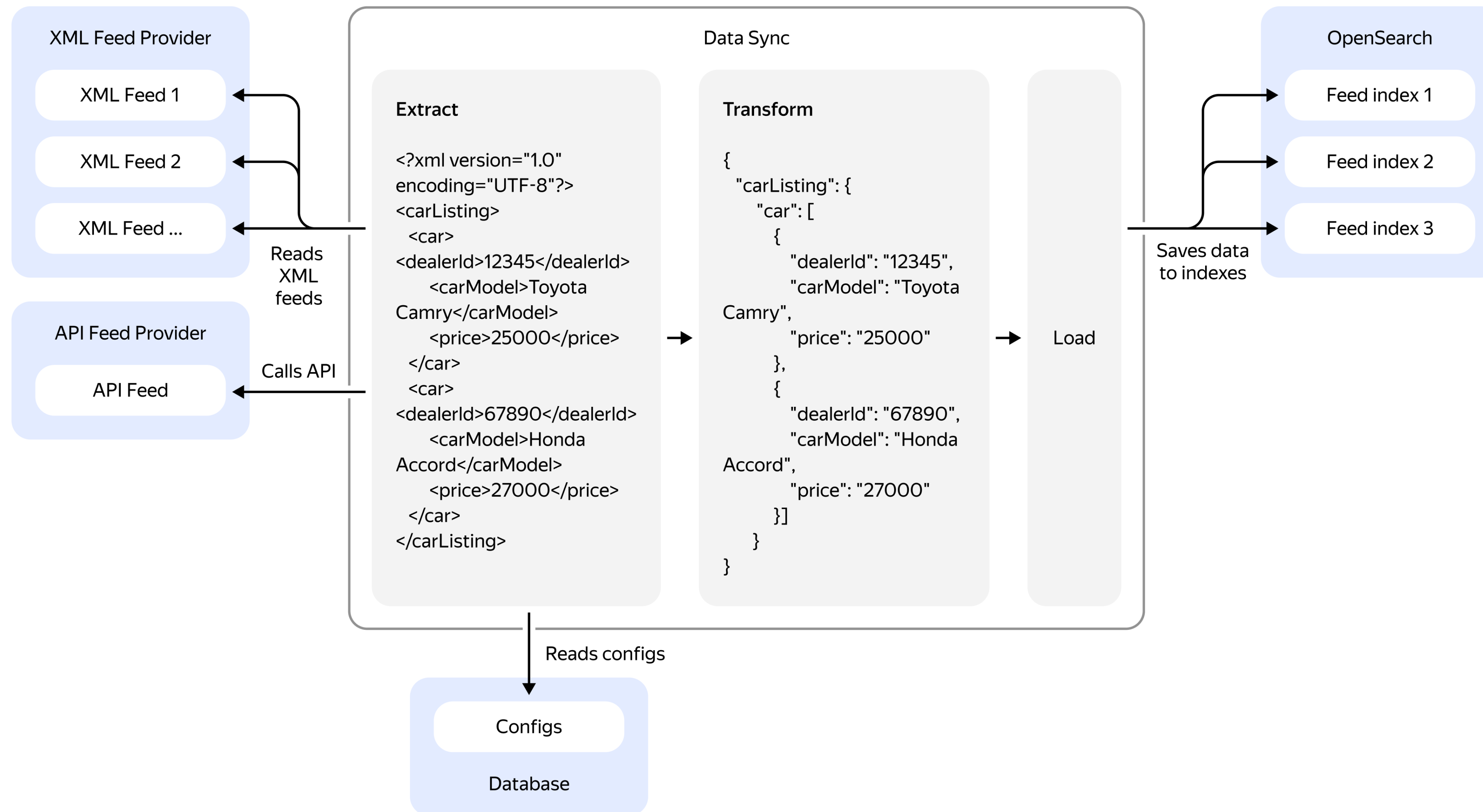
Архитектура RAG Yandex Cloud



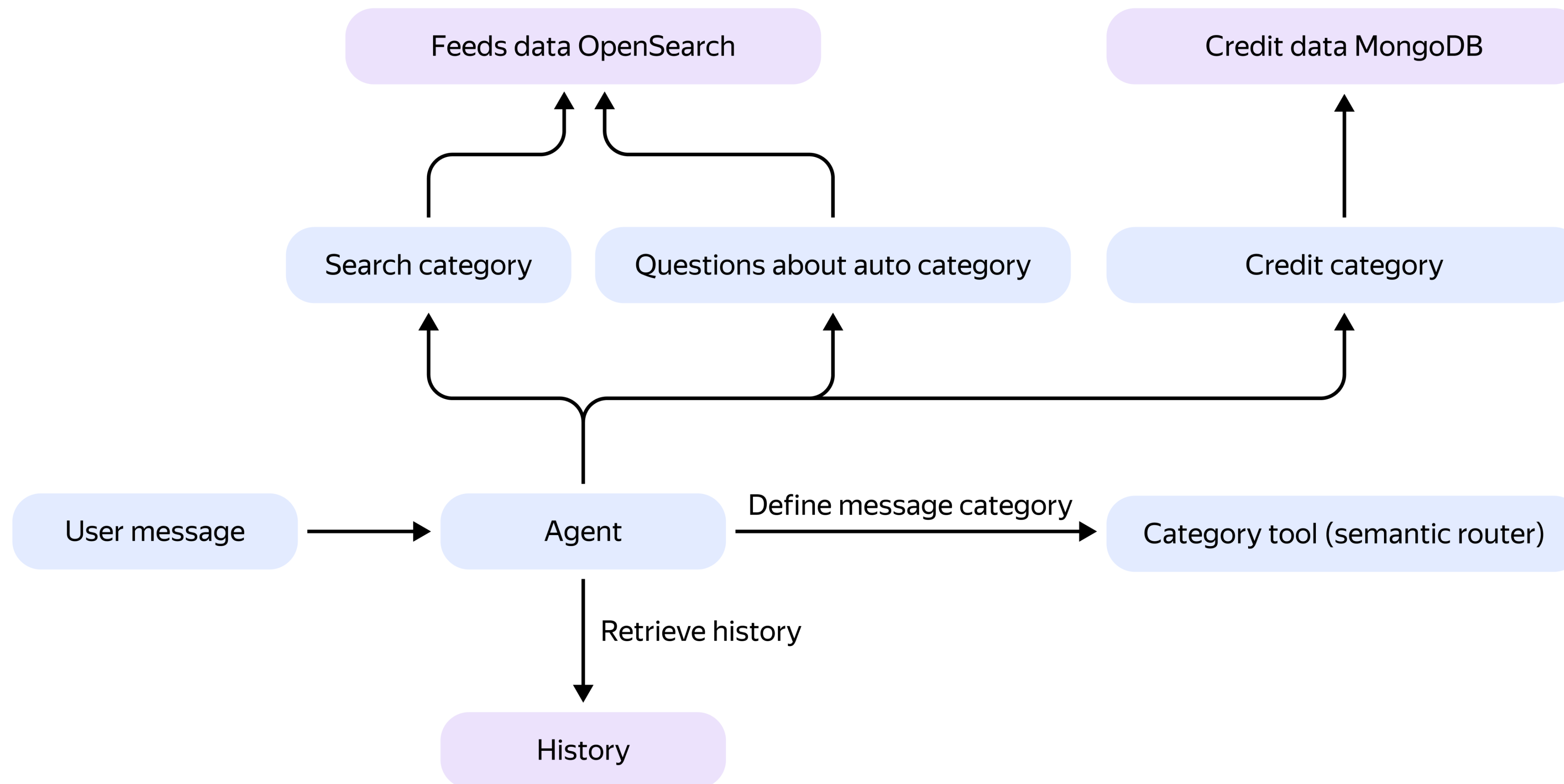
Архитектура RAG-приложения



Архитектура Data Sync



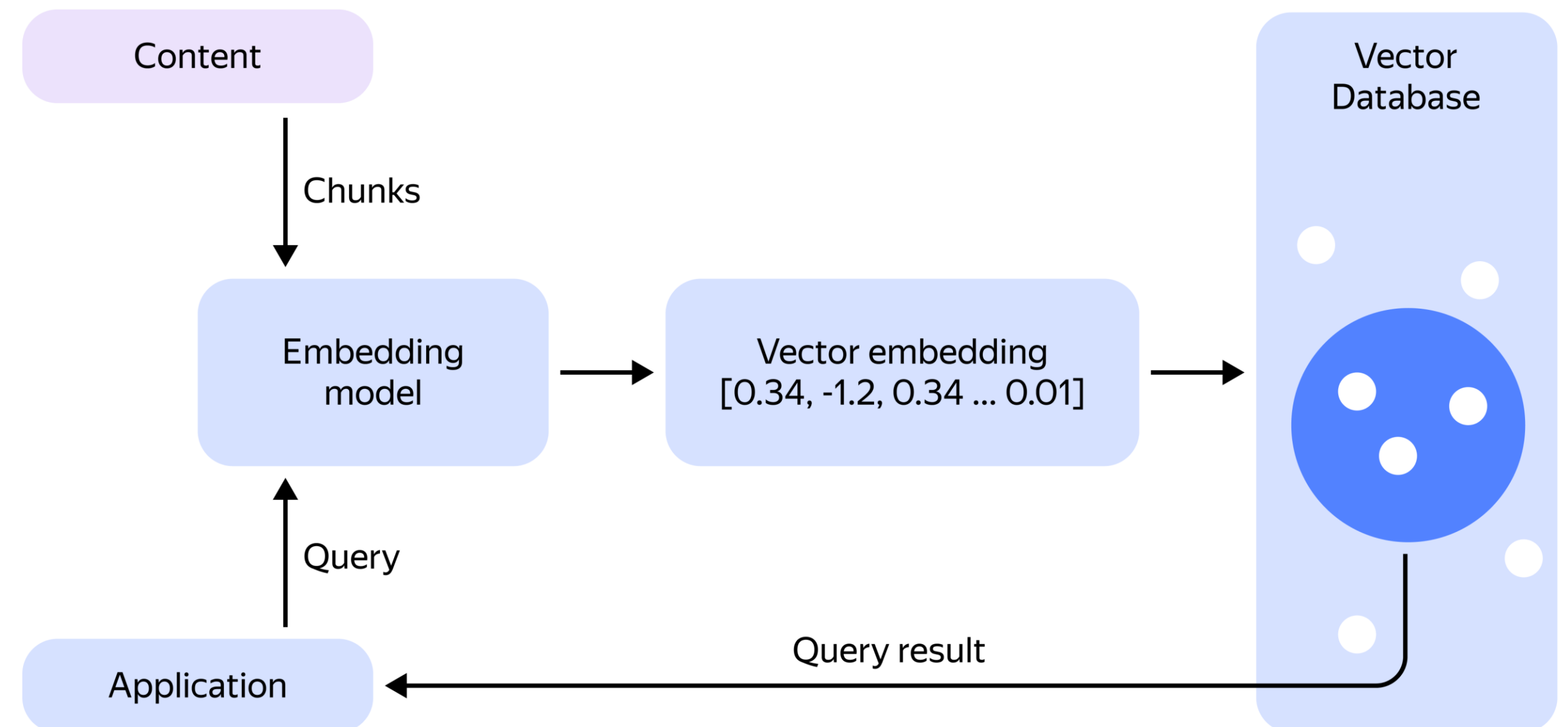
Архитектура RAG с несколькими источниками данных



Базы данных

Векторные базы данных

- Преобразует информацию в плотные векторы
- Быстро и эффективно ищет по семантическому сходству
- Подходит для работы с большими объемами неструктурированных данных



A comparison of leading vector databases

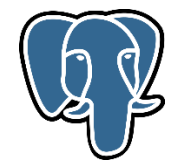
	Pinecone	Weaviate	Milvus	Qdrant	Chroma	Elasticsearch	PGvector
Is open source	✗	✓	✓	✓	✓	✗	✓
Self-host	✗	✓	✓	✓	✓	✓	✓
Cloud management	✓	✓	✓	✓	✗	✓	(✓)
Purpose-built for Vectors	✓	✓	✓	✓	✓	✗	✗
Developer experience	👍👍👍	👍👍	👍👍	👍👍	👍👍	👍	👍
Community	Community page & events	8k ☆ github, 4k slack	23k ☆ github, 4k slack	13k ☆ github, 3k discord	9k ☆ github, 6k discord	23k slack	6k ☆ github
Queries per second (using text nytimes-256-angular)	150 *for p2, but more pods can be added	791	2406	326	?	700-100 *from various reports	141
Latency, ms (Recall/Percentile 95 (millis), nytimes-256-angular)	1 *batched search, 0.99 recall, 200k SBERT	2	1	4	?	?	8
Supported index types	?	HNSW	Multiple (11 total)	HNSW	HNSW	HNSW	HNSW/IVFFlat
Hybrid Search (i.e. scalar filtering)	✓	✓	✓	✓	✓	✓	✓
Disk index support	✓	✓	✓	✓	✓	✗	✓
Role-based access control	✓	✗	✓	✗	✗	✓	✗
Dynamic segment placement vs. static data sharding	?	Static sharding	Dynamic segment placement	Static sharding	Dynamic segment placement	Static sharding	-
Free hosted tier	✓	✓	✓	(free self-hosted)	(free self-hosted)	(free self-hosted)	(varies)
Pricing (50k vectors @1536)	\$70	fr. \$25	fr. \$65	est. \$9	Varies	\$95	Varies
Pricing (20M vectors, 20M req. @768)	\$227 (\$2074 for high performance)	\$1536	fr. \$309 (\$2291 for high performance)	fr. \$281 (\$820 for high performance)	Varies	est. \$1225	Varies

Параметры	Milvus	Qdrant
Масштабируемость	Высокая (2406 запросов в секунду)	Средняя (326 запросов в секунду)
Низкая задержка	1 мс	4 мс
Поддержка гибридного поиска	Да	Да
RBAC (контроль доступа)	Да	Нет
Стоимость	От \$65 в месяц	От \$9 в месяц
Индексы	Поддержка множества индексов (11)	HNSW

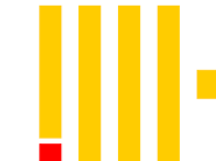
Базы данных с поддержкой поисков по векторам



ChromaDB



PostgreSQL



ClickHouse®



OpenSearch



ElasticSearch

ClickHouse® — когда использовать в качестве БД для хранения векторов?

- Одновременный поиск по векторам и фильтрации по метаданным и/или агрегации, JOIN
- Линейное сопоставление по очень большим объемам векторов
- Нужна поддержка SQL
- Нет необходимости выполнять векторизацию на стороне БД
- Уже есть ClickHouse® в используемом стеке

ClickHouse® — когда не использовать в качестве векторной БД?



Если ваш датасет
небольшой
и полностью
помещается
в памяти

1

Нет метаданных
и нужно только
сопоставление
по векторам

2

Нужна
векторизация
на стороне БД
в момент вставки

3

ClickHouse® — основные функции

`cosineDistance (vector1, vector2)`

`L2Distance (vector1, vector2)`

Векторизированный датасет

	id	document	embedding	metadata	uuid
1	e6b6eff00c...	#120331, #120544, #120262, ...	[0.1015625, -0.064331055, 0.031204224, ...	('s3://iml-ra...	76fab173-bd92...
2	4ef21eb421...	MacOS 14 runtime check (#11...	[0.055908203, -0.056121826, 0.002420425...	('s3://iml-ra...	06e599d1-c13d...
3	999a347a34...	welford combine on first re...	[0.05117798, -0.05859375, -0.057006836,...	('s3://iml-ra...	d2f8e277-6a3f...
4	d50200c385...	pytorch↵2.2.2↵2.3.0↵2024-04...	[0.012428284, 0.0011148453, 0.006526947...	('s3://iml-ra...	2e5c7fd6-9c9b...
5	72aada7b3f...	for Device (#116019, #11683...	[0.066833496, -0.05026245, 0.033050537,...	('s3://iml-ra...	8b22c3b7-02aa...
6	620b73432e...	non-integer expr support fo...	[0.10266113, -0.03186035, -0.02748108, ...	('s3://iml-ra...	d6f3a98e-2e33...
7	b0f7b73750...	for each dynamo test skips ...	[0.07397461, -0.050354004, -0.023925781...	('s3://iml-ra...	cc1bd271-6887...

Работа с векторными базами данных

Загрузка документов в базу

```
from langchain_community.document_loaders import TextLoader
from langchain_community.vectorstores import FAISS

from langchain_openai import OpenAIEmbeddings

from langchain_text_splitters import CharacterTextSplitter

loader= TextLoader("../state_of_the_union.txt")

documents = loader.load()

text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=0)

texts = text_splitter.split_documents(documents)

embeddings = OpenAIEmbeddings()

db = FAISS.from_documents(texts, embeddings)
```

Поиск документов в базе

```
retriever = db.as_retriever()

docs = retriever.invoke("what did he say about ketanji brown jackson")
```

Токенизация/ТЮНИНГ

Токенизация

Токенизация (Splitting) — процесс разделения текста на более мелкие компоненты (токены)

This is exactly where chunking comes to rescue. Chunking is the process of breaking down the humungous text in small chunks of texts so that it could be fed easily as an when needed to a LLM. There are various chunking strategies available and each strategies has their pros and cons. As a GenAI engineer, once document is loaded, you want to transform the texts in the document according to your application and one should at least know which chunking strategy would serve their purpose, as chunking plays an important role in deciding the performance / accuracy of the LLM.

Upload .txt

Splitter: Character Splitter  

Chunk Size: 

Chunk Overlap: 

Total Characters: 715

Number of chunks: 29

Average chunk size: 24.7

This is exactly where chunking comes to rescue. Chunking is the process of breaking down the humungous text in small chunks of texts so that it could be fed easily as an when needed to a LLM. There are various chunking strategies available and each strategies has their pros and cons. As a GenAI engineer, once document is loaded, you want to transform the texts in the document according to your application and one should at least know which chunking strategy would serve their purpose, as chunking plays an important role in deciding the performance / accuracy of the LLM.

Тюнинг при токенизации, сравнение способов токенизации

RecursiveCharacterTextSplitter
и CharacterTextSplitter
2 базовых токенизатора.

Основные параметры:

chunk_size

chunk_overlap

separator

```
text_splitter = RecursiveCharacterTextSplitter(  
    chunk_size=500,  
    chunk_overlap=50,  
    length_function=len,  
    is_separator_regex=False,  
)  
  
texts =  
text_splitter.create_documents([state_of_the_union])  
print(texts[0])  
print(texts[1])
```

Подбор размера блока chunk_size

1. Определение целей
2. Проверка длины текста
3. Эксперименты



Подбор перекрытия блока chunk_overlap

1. Сохранение контекста
2. Выбор значения
3. Эксперименты

chunk_overlap=0 | Total Chunks=5

Laser Inertial Fusion Energy

LIFE, short for Laser Inertial Fusion Energy, was a fusion energy effort run at Lawrence Livermore National Laboratory between 2008 and 2013.

LIFE aimed to develop the technologies necessary to convert the laser-driven inertial confinement fusion concept being developed in the National Ignition Facility (NIF) into a practical commercial power plant, a concept known generally as inertial fusion energy (IFE).

LIFE used the same basic concepts as NIF, but aimed to lower costs using mass-produced fuel elements, simplified maintenance, and diode lasers with higher electrical efficiency.

Background

Lawrence Livermore National Laboratory (LLNL) has been a leader in laser-driven inertial confinement fusion (ICF) since the initial concept was developed by LLNL employee John Nuckols in the late 1950s. The basic idea was to use a driver to compress a small pellet known as the target that contains the fusion fuel, a mix of deuterium (D) and tritium (T).

If the compression reaches high enough values, fusion reactions begin to take place, releasing alpha particles and neutrons. The alphas may impact atoms in the surrounding fuel, heating them to the point where they undergo fusion as well. If the rate of alpha heating is higher than heat losses to the environment, the result is a self-sustaining chain reaction known as ignition.

chunk_overlap=x% | Total Chunks=9

Laser Inertial Fusion Energy

LIFE, short for Laser Inertial Fusion Energy, was a fusion energy effort run at Lawrence Livermore National Laboratory between 2008 and 2013.

LIFE aimed to develop the technologies necessary to convert the laser-driven inertial confinement fusion concept being developed in the National Ignition Facility (NIF) into a practical commercial power plant, a concept known generally as inertial fusion energy (IFE).

LIFE used the same basic concepts as NIF, but aimed to lower costs using mass-produced fuel elements, simplified maintenance, and diode lasers with higher electrical efficiency.

Background

Lawrence Livermore National Laboratory (LLNL) has been a leader in laser-driven inertial confinement fusion (ICF) since the initial concept was developed by LLNL employee John Nuckols in the late 1950s. The basic idea was to use a driver to compress a small pellet known as the target that contains the fusion fuel, a mix of deuterium (D) and tritium (T).

If the compression reaches high enough values, fusion reactions begin to take place, releasing alpha particles and neutrons. The alphas may impact atoms in the surrounding fuel, heating them to the point where they undergo fusion as well. If the rate of alpha heating is higher than heat losses to the environment, the result is a self-sustaining chain reaction known as ignition.

Определение разделителя

Естественные разделители

Используйте естественные разделители текста, такие как пробелы, знаки препинания (точки, запятые) или переходы на новую строку. Это помогает избежать разрыва слов или предложений

Специфика текста

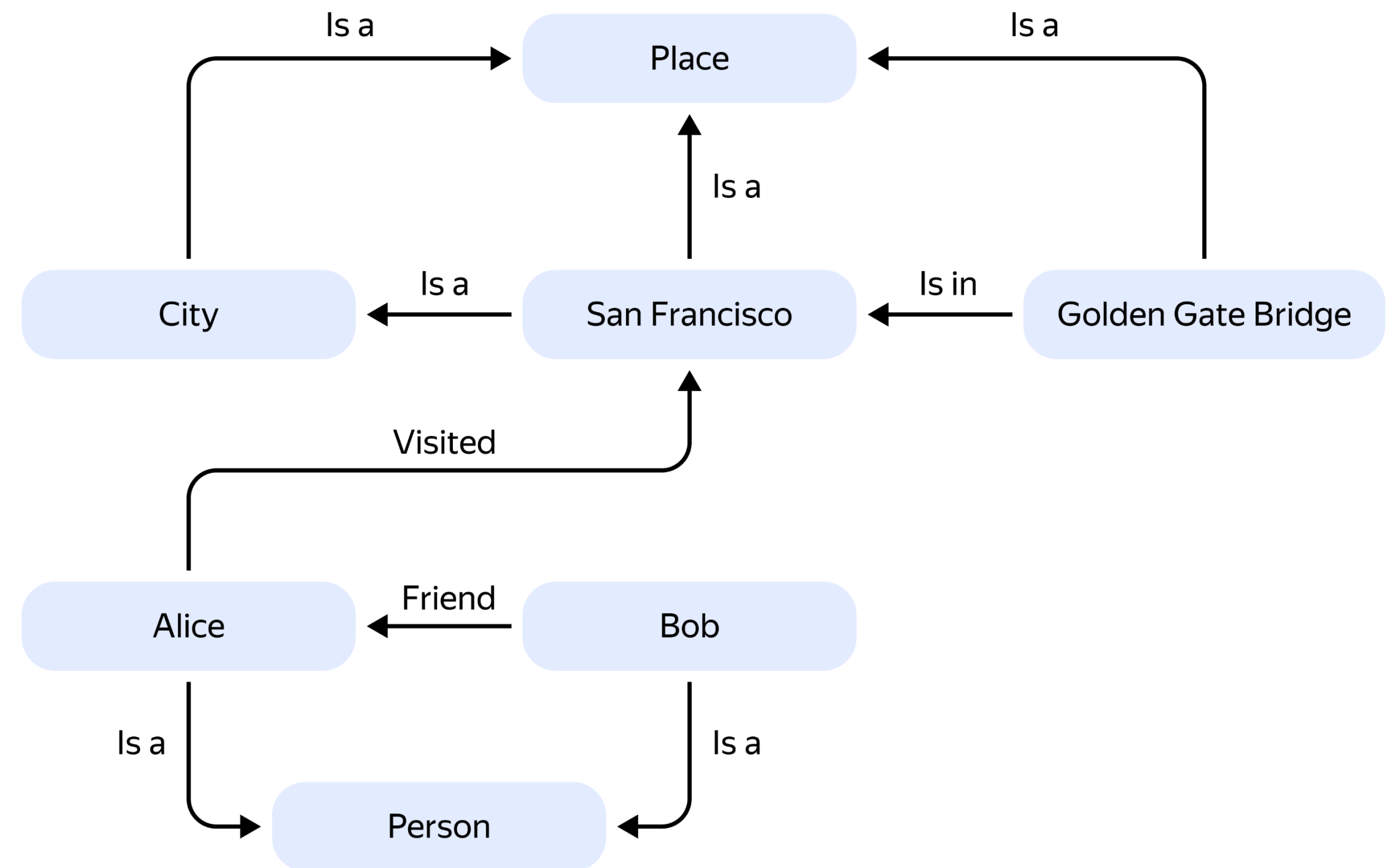
Если текст имеет специфическую структуру (например, списки или заголовки), выберите разделитель, который лучше всего соответствует этой структуре

Эксперименты

Начните со знаков-разделителей предложений («.», «!», «?») как базового разделителя, а затем попробуйте другие символы (например, «\n» для многострочного текста), оценивая качество разбиения

Графовые базы данных

- Использует графы знаний для представления данных
- Узлы — сущности, ребра — связи между ними
- Обеспечивает более глубокое понимание и контекстуальные ответы
- Отлично подходит для доменов с сложными, структурированными данными (медицина, финансы)



Примеры данных для графов на основе финансового домена

Типы сущностей

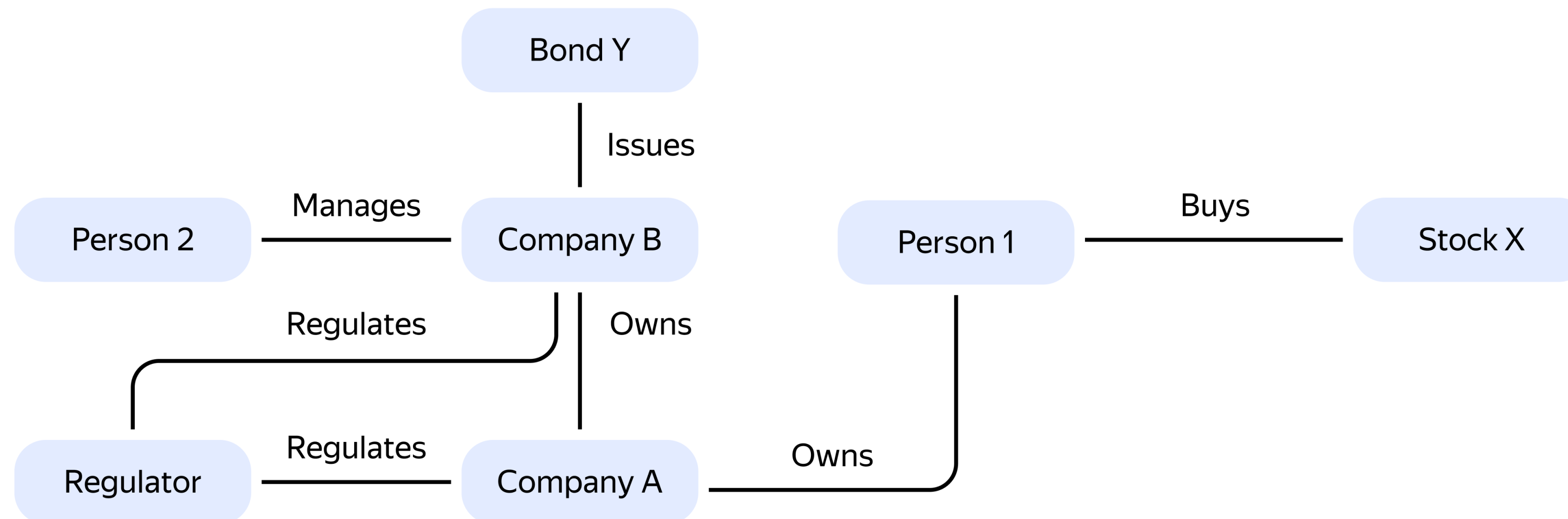
- Компании
- Люди
- Финансовые инструменты

Связи

- Владение
- Транзакции
- Регуляторные связи

Примеры использования

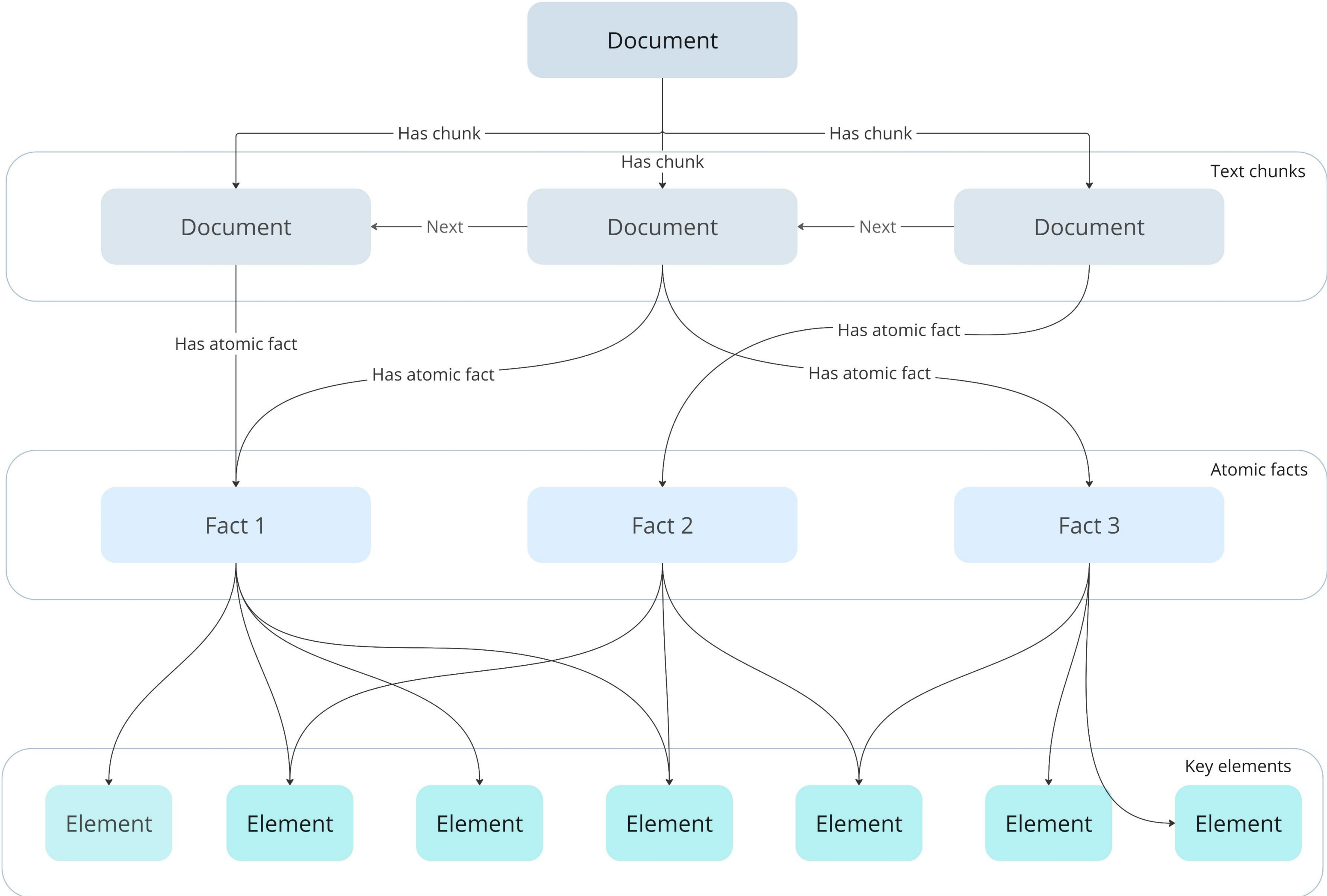
- Обнаружение мошенничества
- Оценка рисков
- Инвестиционная стратегия



Графовые базы данных

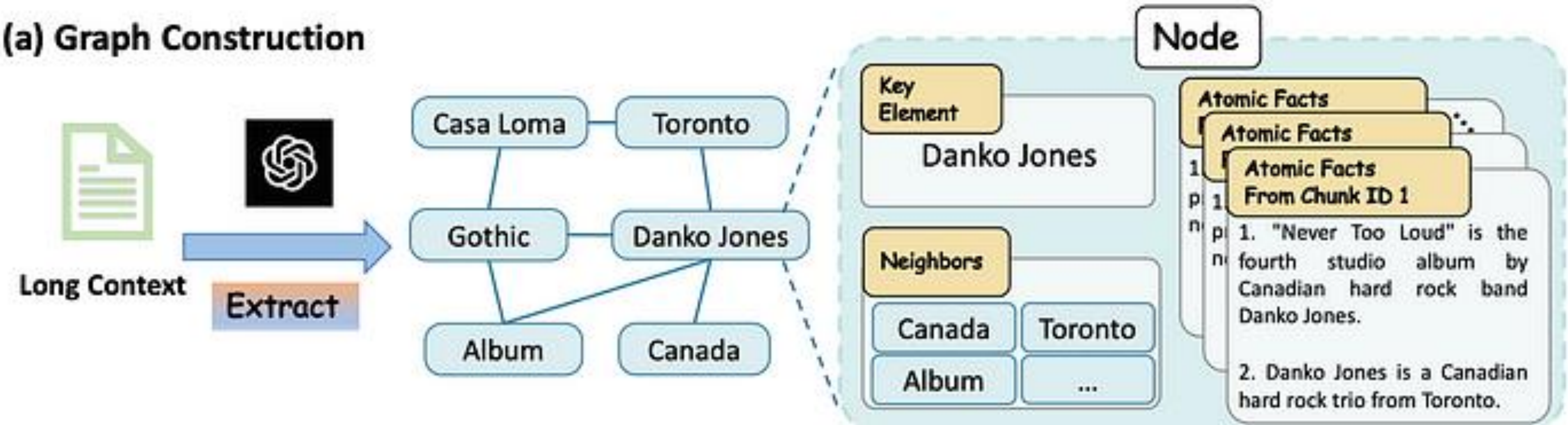


Формирование графовой базы для RAG

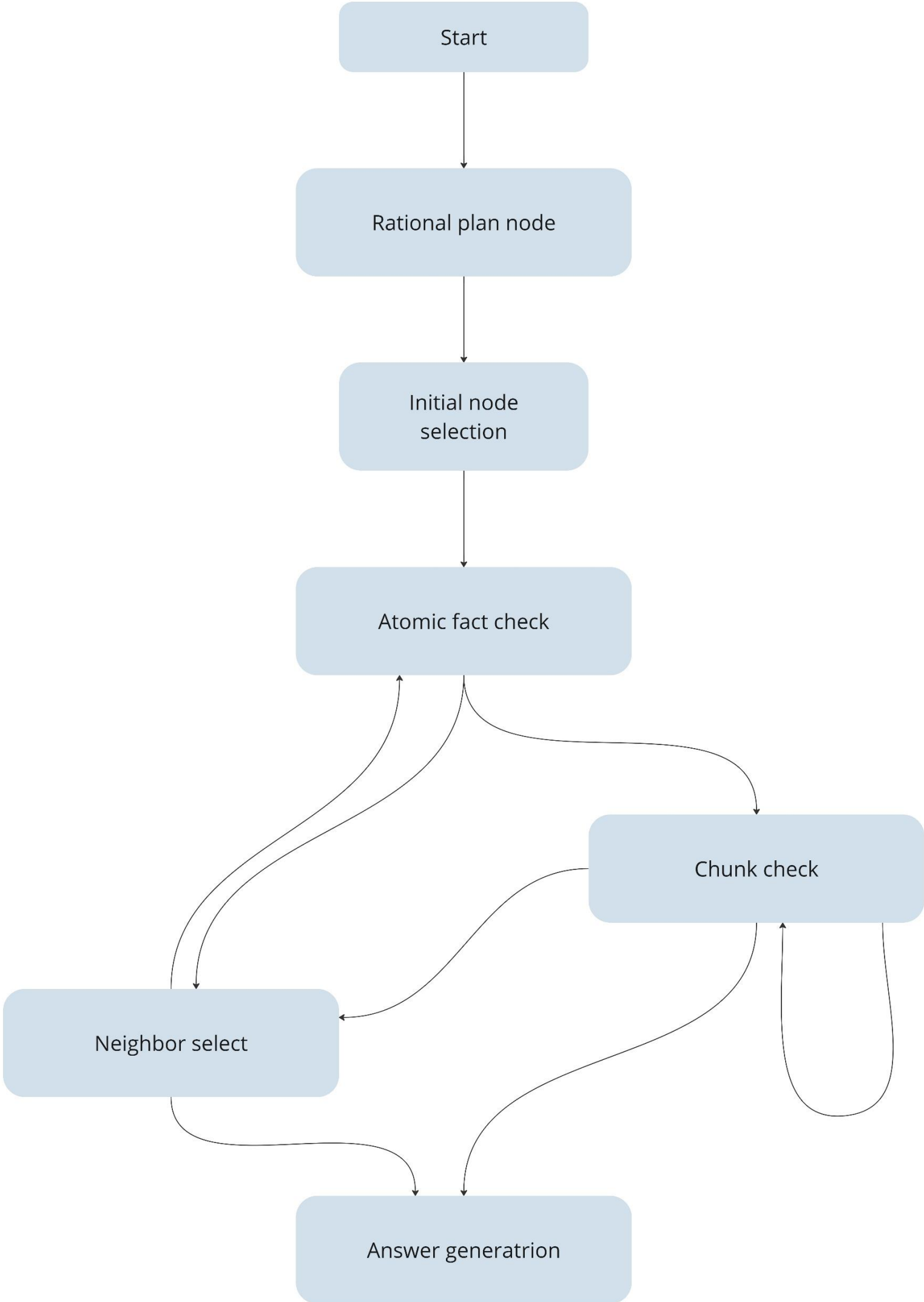


Формирование графовой базы для RAG

(a) Graph Construction



GraphReader Agent



А ЧТО ПО КОДУ?

```
from langchain.chains import GraphCypherQAChain
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)
chain = GraphCypherQAChain.from_llm(graph=graph, llm=llm, verbose=True)

response = chain.invoke({"query": "What was the cast of the Casino?"})
```

```
[1m> Entering new GraphCypherQAChain chain...[0m
```

Generated Cypher:

```
[32;1m[1;3mMATCH (:Movie {title: "Casino"})<-[:ACTED_IN]-(actor:Person)
RETURN actor.name[0m
```

Full Context:

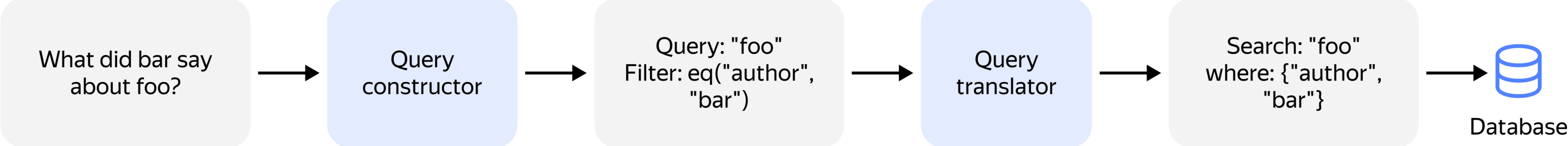
```
[32;1m[1;3m[{'actor.name': 'Joe Pesci'}, {'actor.name': 'Robert De Niro'}, {'actor.name': 'Sharon Stone'}, {'actor.name': 'James Woods'}][0m
```

```
[1m> Finished chain.[0m
```

```
{'query': 'What was the cast of the Casino?',
 'result': 'The cast of Casino included Joe Pesci, Robert De Niro, Sharon Stone, and James Woods.'}
```

Поиск по метаданным

Поиск по метаданным



Поиск по метаданным

```
from langchain.chains.query_constructor.base import AttributeInfo
from langchain.retrievers.self_query.base import SelfQueryRetriever
from langchain_openai import ChatOpenAI
```

```
metadata_field_info = [
    AttributeInfo(
        name="genre",
        description="The genre of the movie. One of ['science fiction', 'comedy', 'drama', 'thriller', 'romance']",
        type="string",
    ),
    AttributeInfo(
        name="year",
        description="The year the movie was released",
        type="integer",
    )
]
```

```
document_content_description = "Brief summary of a movie"
```

```
llm = ChatOpenAI(temperature=0)
```

```
retriever = SelfQueryRetriever.from_llm(
```

```
    llm,
```

```
    vectorstore,
```

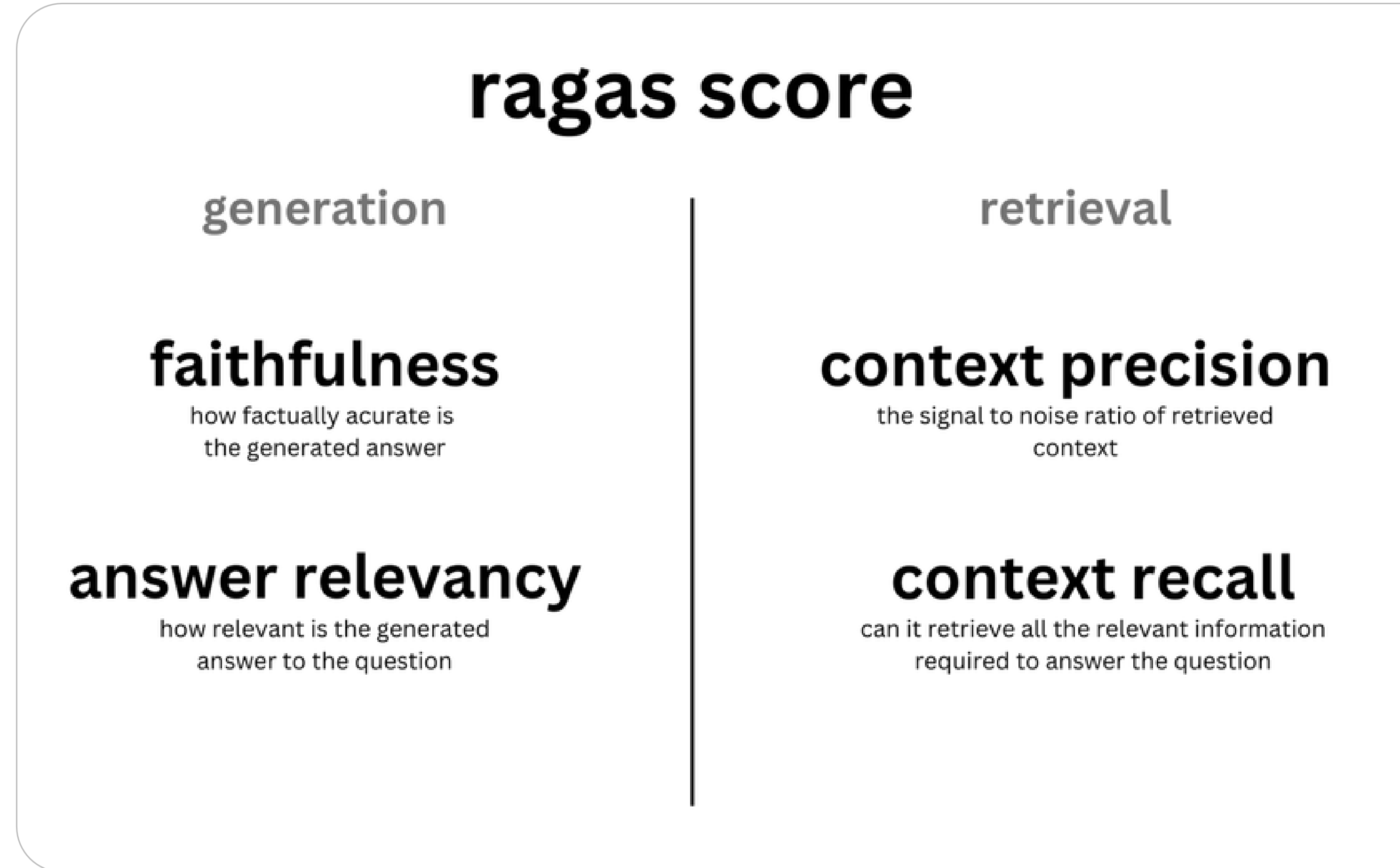
```
    document_content_description,
```

```
    metadata_field_info,
```

```
)
```


Оценка качества RAG-приложений

RAGAS — фреймворк для оценки качества работы RAG-приложений



Оценка качества токенизации

Context Precision

Эта метрика измеряет точность определения контекста, что важно для корректного разбиения текста на логические единицы

$$\text{Context Precision@K} = \frac{\sum_{k=1}^K (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top K results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{\text{true positives@k} + \text{false positives@k}}$$

Context Recall

Оценивает полноту извлечения релевантного контекста, что помогает понять, насколько качественно производится разбиение текста

$$\text{context recall} = \frac{|\text{GT claims that can be attributed to context}|}{|\text{Number of claims in GT}|}$$

Как запустить оценку

Для запуска оценки требуется собрать датасет, в котором содержится информация о фактах

```
from datasets import Dataset

import os

from ragas import evaluate

from ragas.metrics import faithfulness, answer_correctness

os.environ["OPENAI_API_KEY"] = "your-openai-key"

data_samples = {
    'question': ['When was the first super bowl?', 'Who won the most super bowls?'],
    'answer': ['The first superbowl was held on Jan 15, 1967', 'The most super bowls have been won by The New England Patriots'],
    'contexts': [['The First AFL-NFL World Championship Game was an American football game played on Jan 15, 1967.',
                  'The Green Bay Packers...Green Bay, Wisconsin.', 'The Packers compete...Football Conference']],
    'ground_truth': ['The first superbowl was held on January 15, 1967', 'The New England Patriots have won the most super bowls']
}

dataset = Dataset.from_dict(data_samples)

score = evaluate(dataset, metrics=[faithfulness, answer_correctness])

score.to_pandas()
```


Выводы

1. RAG-приложения являются одним из способов расширения знаний моделей. Особенно актуальным для проприетарных и динамических данных
2. Для поиска схожих документов можно использовать не только векторные, но и графовые и другие базы знаний в зависимости от задачи
3. RAG можно построить в облаке, on prem, гибриде. А также использовать несколько источников данных при реализации
4. Метрики Context Precision и Context Recall позволяют оценить качество подготовки базы знаний для RAG-приложения
5. Подбор размера блока и размера перекрытия определяется эмпирическим способом
6. Графовые базы данных подойдут для доменов со сложными структурированными данными (медицина, финансы)

Практика

Будем рады ответить на ваши вопросы



Кузьма Лешаков
Архитектор Data Platform,
Yandex Cloud



Александр Константинов
CTO, Raft DS