

Построение RAG-приложения с использованием YaGPT

Кузьма Лешаков
Yandex Cloud



Кузьма Лешаков

≈ 8 лет в IT

2023



Более года в Yandex Cloud
Команда Data Platform

2021



Uma.Tech.
Ведущий инженер данных

2019



Clover Group
Инженер данных

2017

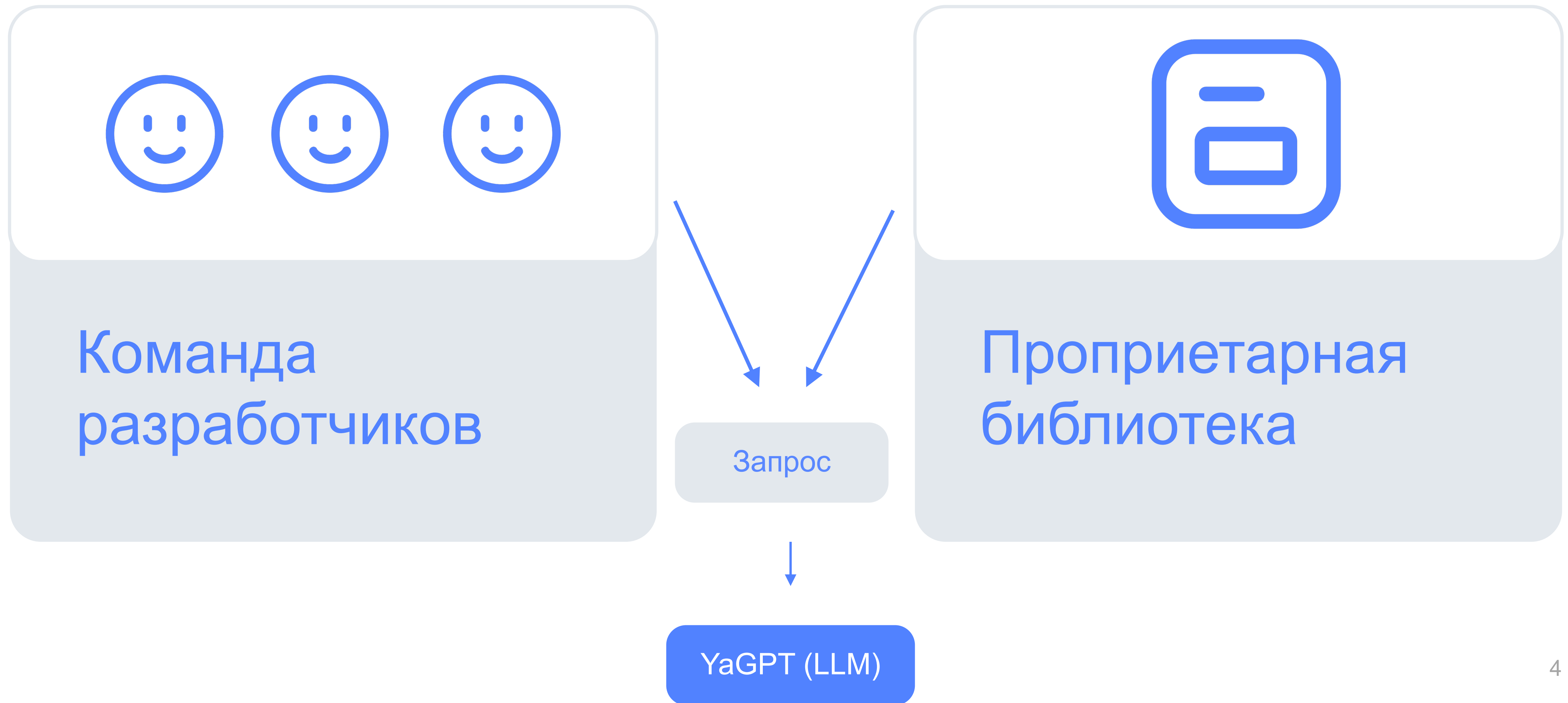


The Linux Foundation
Стажёр-разработчик

О чём поговорим

1. Введение в YaGPT
2. Архитектура RAG приложения
3. Построение RAG приложения

Сценарий 1 – библиотека X



Языковые модели развиваются в Яндексе уже не первый год

2021



В июне — сервис
«Зелибоба»/«Балабоба»

2022



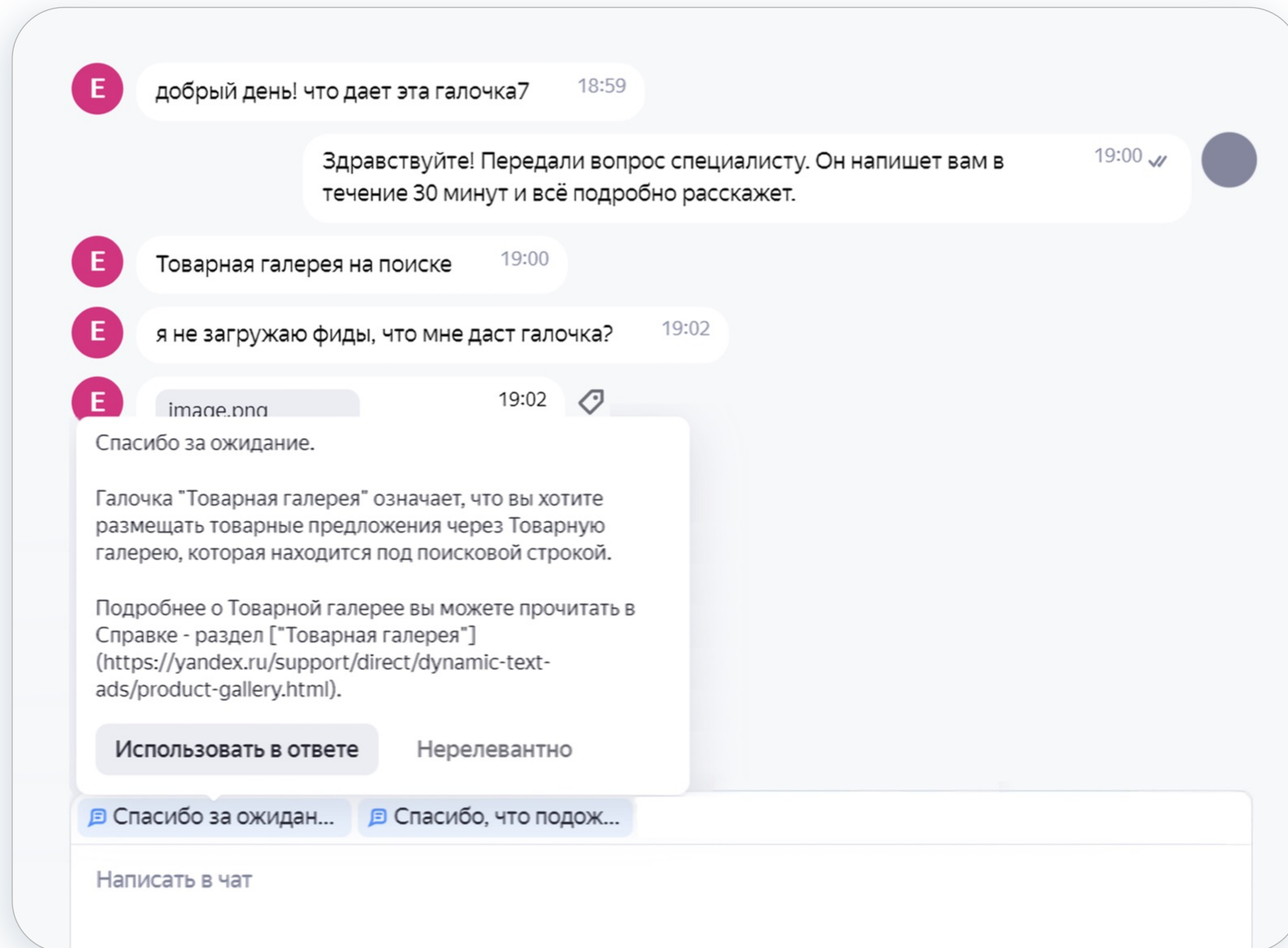
В июне Яндекс выложил одну
из крупнейших языковых
моделей — YaLM 100B

2023



Яндекс анонсировал выпуск
новой модели YandexGPT

Лавка: Пример подсказок



Лавка: генерация новых атрибутов

Яндекс Голубь

Товары

Рецепты

Комбинации

Мастер-категории

Фронт-категории

Инфомодели

Атрибуты

Группы атрибутов

Витрина

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

Активный Неактивный

Мастер категория

Пирожные промышленные

Штрихкод
barcode

4627119671554

Добавить штрихкод

Длинное название
longName

Пирожное «Медовейник» «Из Л

Короткое название (loc)
shortNameLoc

Пирожное Медовейник Из Лави

Маркетинговое количество в упаковке
markCount

120

Единица маркетингового количества в упаковке (список)
markCountUnitList

г

Тип номенклатуры
nomenclatureType

Товар

Неизменяемое значение атрибута

Изображение товара

image

1

2

+

Тэги и стикеры

Стикер на фото
photoStickers

Выберите значение

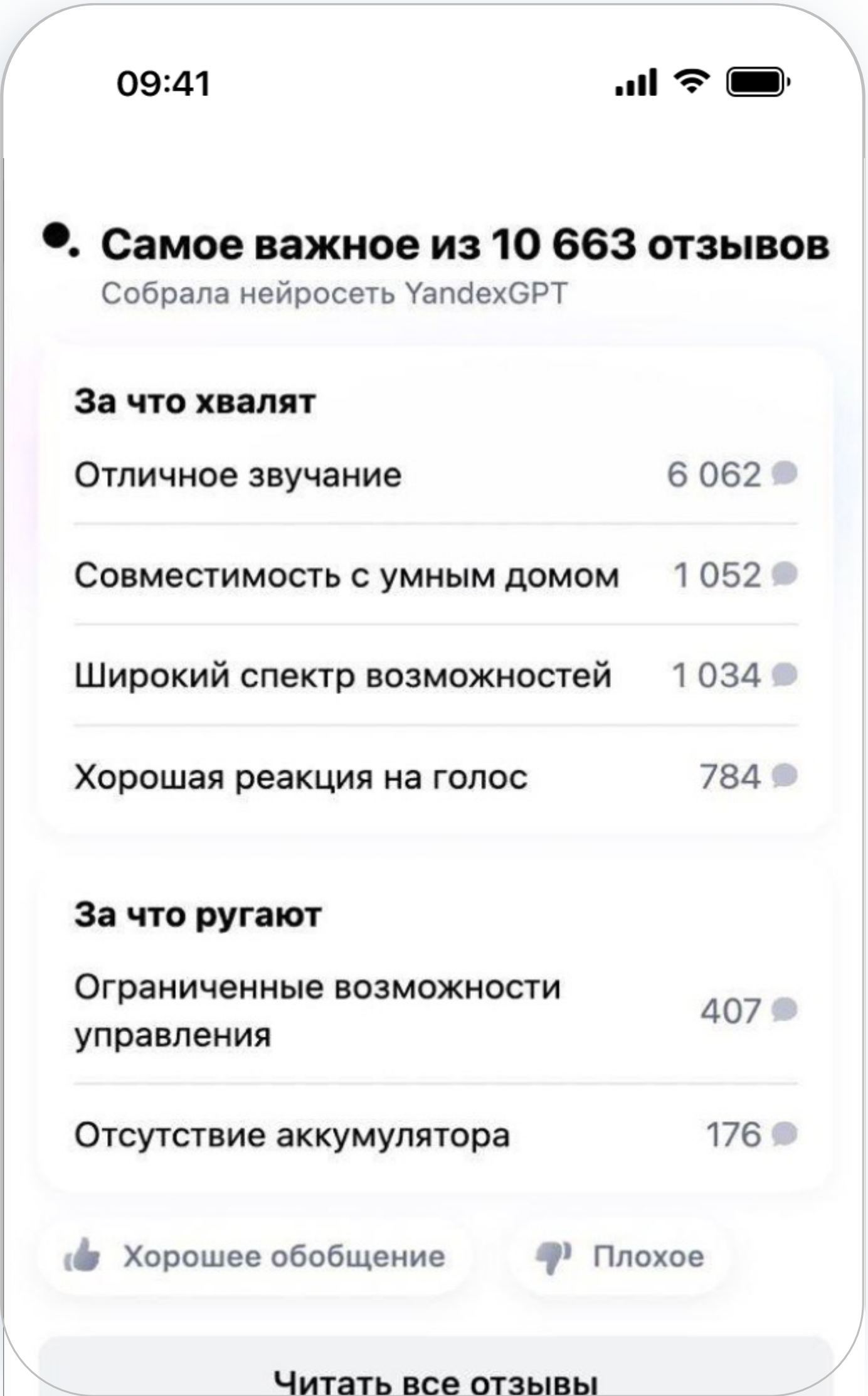
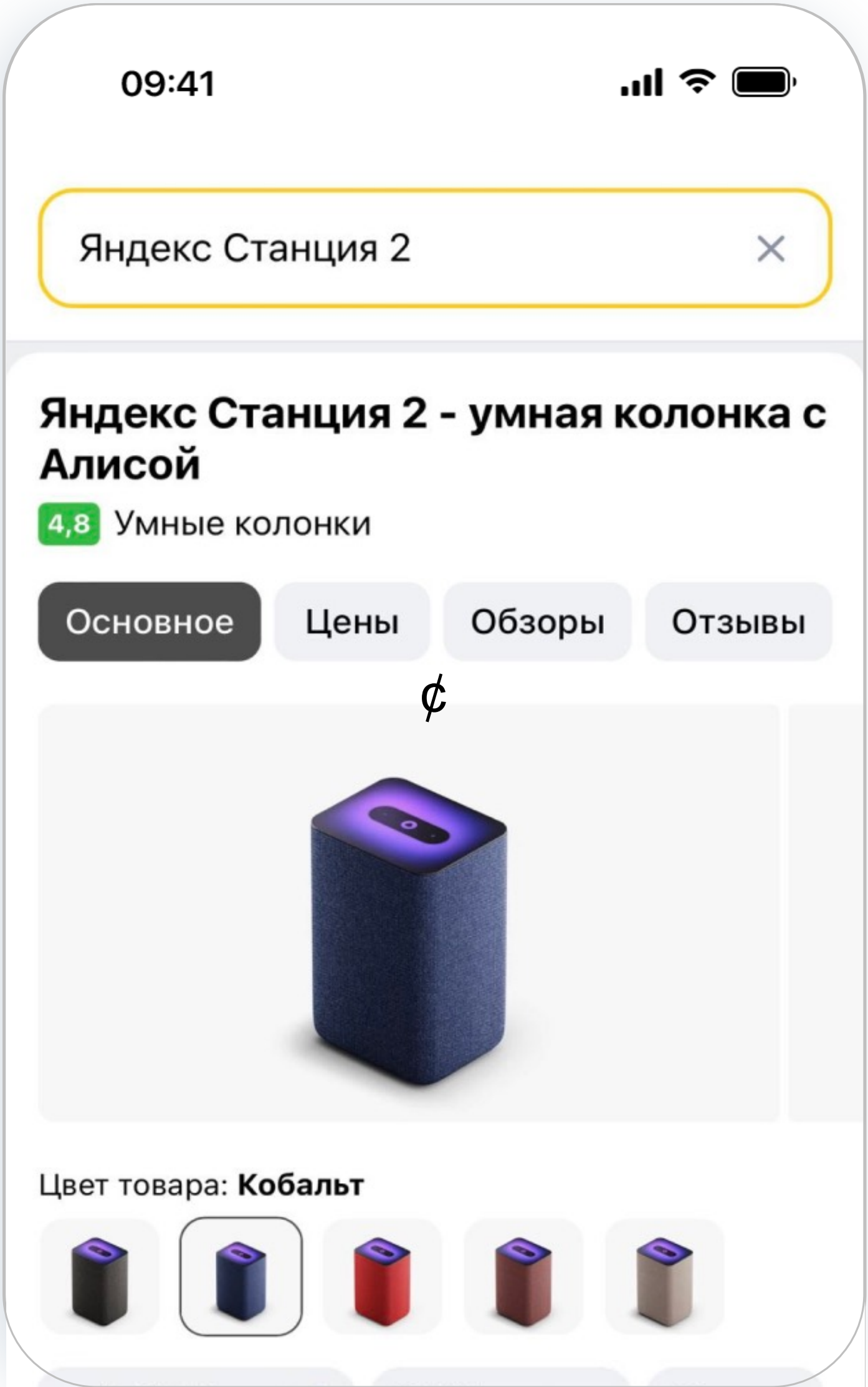
★

Version: dev

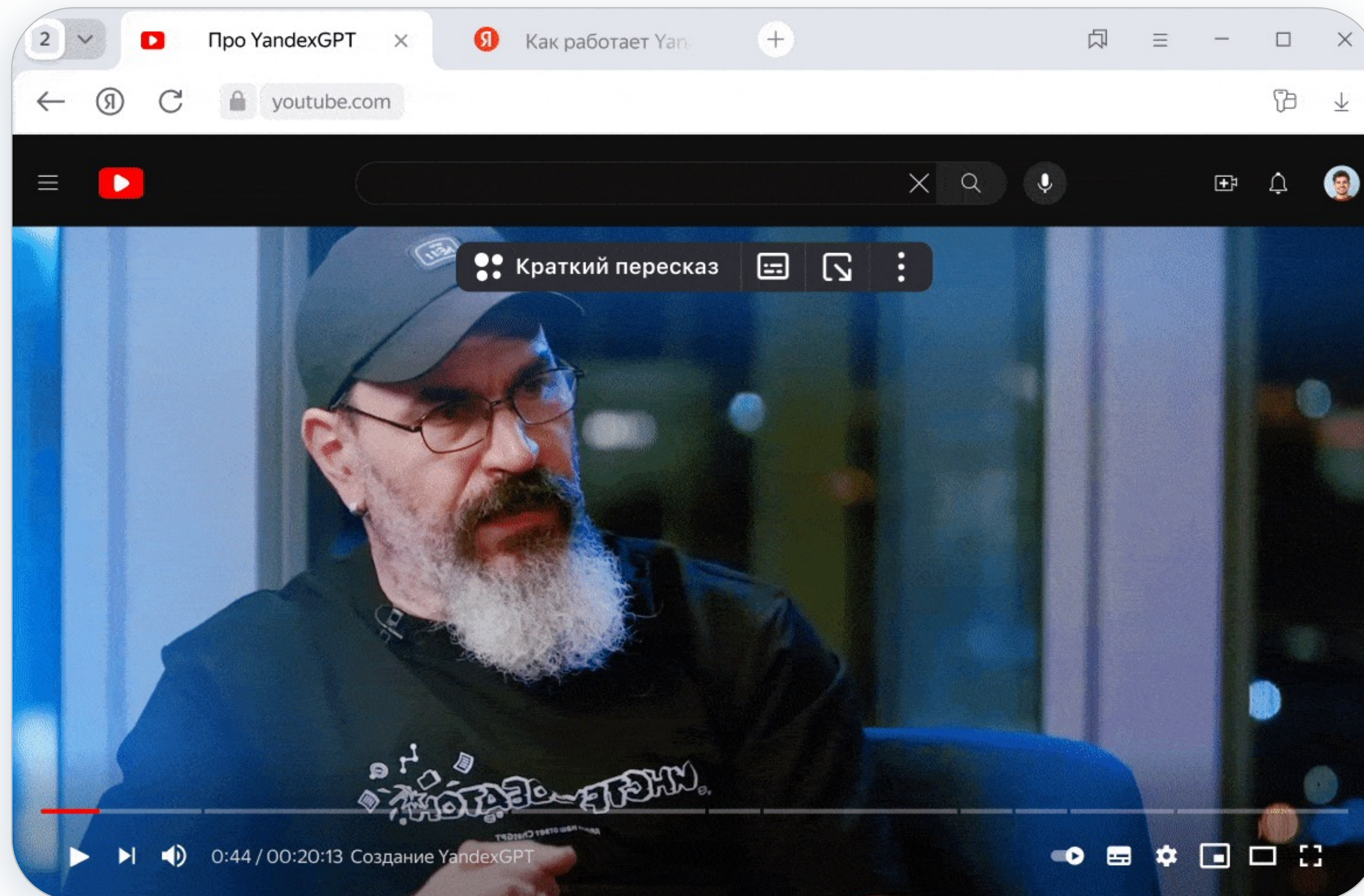
Конфиденциально

© 1997–2023 ООО «ЯНДЕКС»

Суммаризация ОТЗЫВОВ



Суммаризация видео



475 лет

сэкономлено
за месяц

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



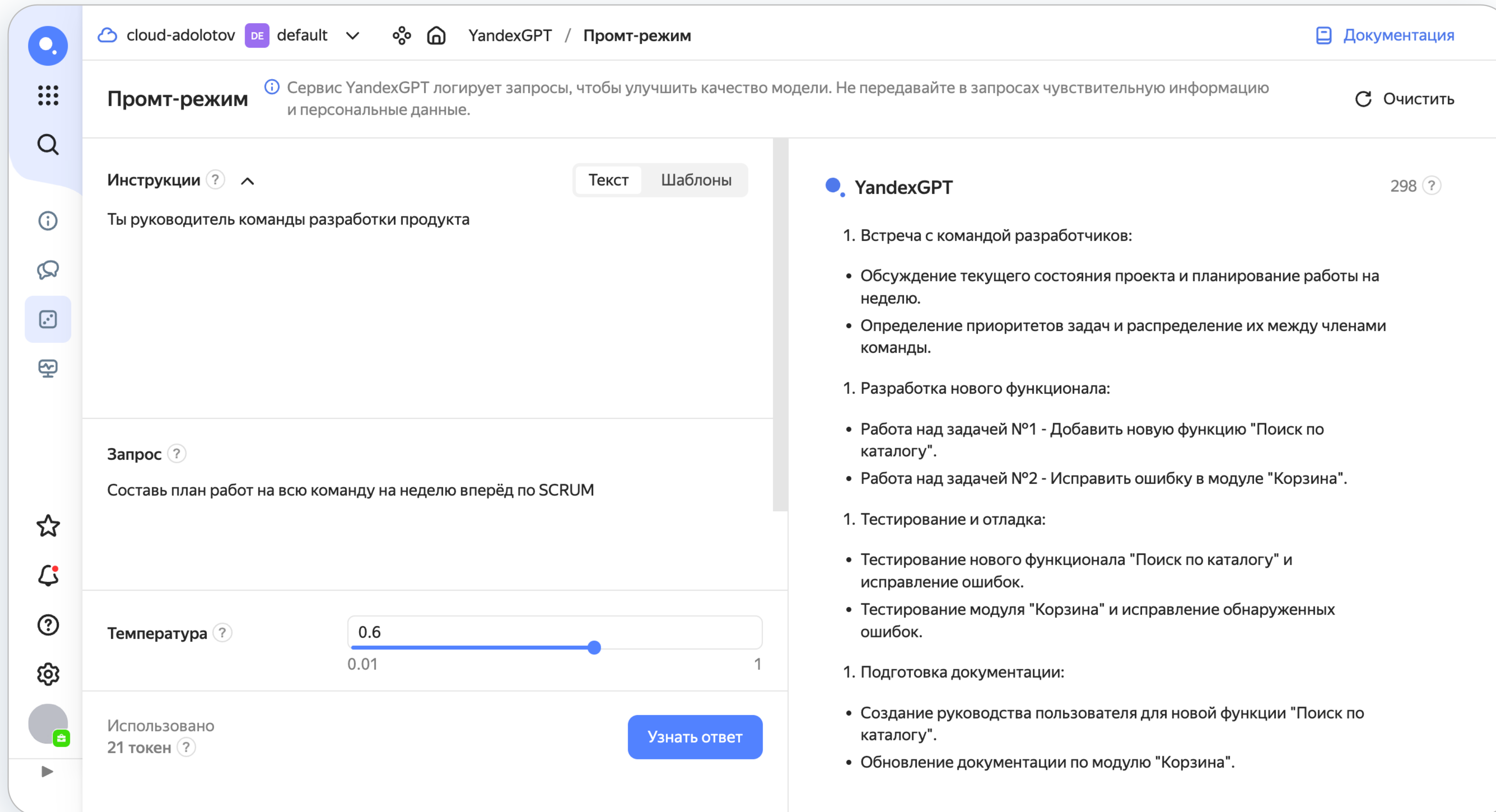
Есть решения, лучше LLM

- 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

Взаимодействие с YandexGPT API

Повышение
качества моделей

Консоль: instruct режим



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
        участках на маршруте, но и показывают стоимость проезда.
        Это позволяет водителям заранее оценить расходы
        или выбрать маршрут без платных дорог, если это возможно...'
    }
  ]
}
```

YandexGPT API в Public Preview!

Доступные модели для всех пользователей

Модели

YandexGPT Lite

YandexGPT Pro

Эмбеддинги

Токенизатор

Суммаризация

На базе YaGPT Lite

Инференс

Синхронно

Стандарт

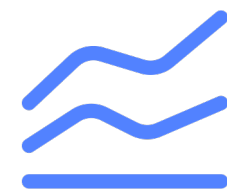
Асинхронно

Know-how

Взаимодействие
с YandexGPT API

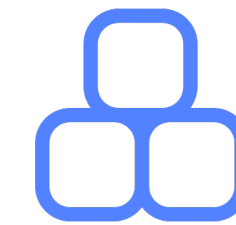
Повышение
качества моделей

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

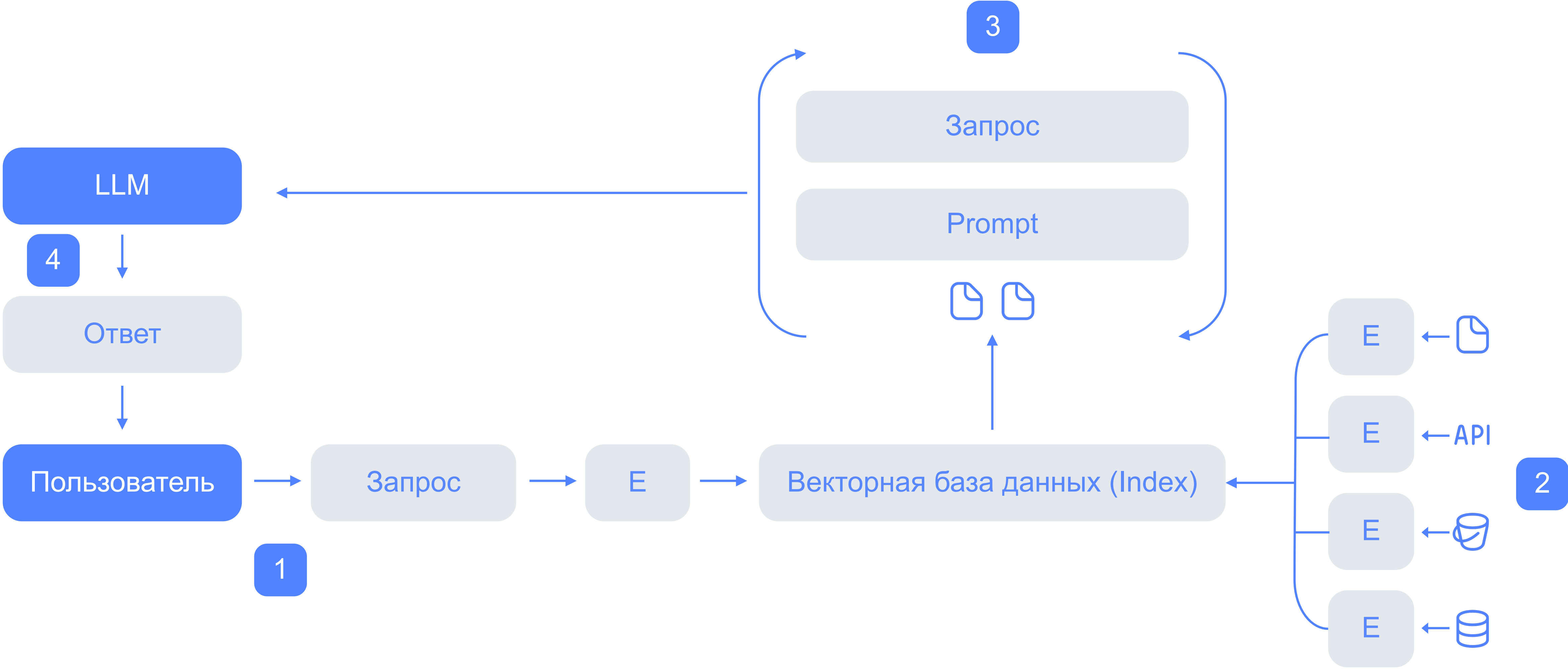
Чтобы дообучить модель YandexGPT нужно подготовить файл с формате JSON. Файл надо сохранить в кодировке UTF-8:

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

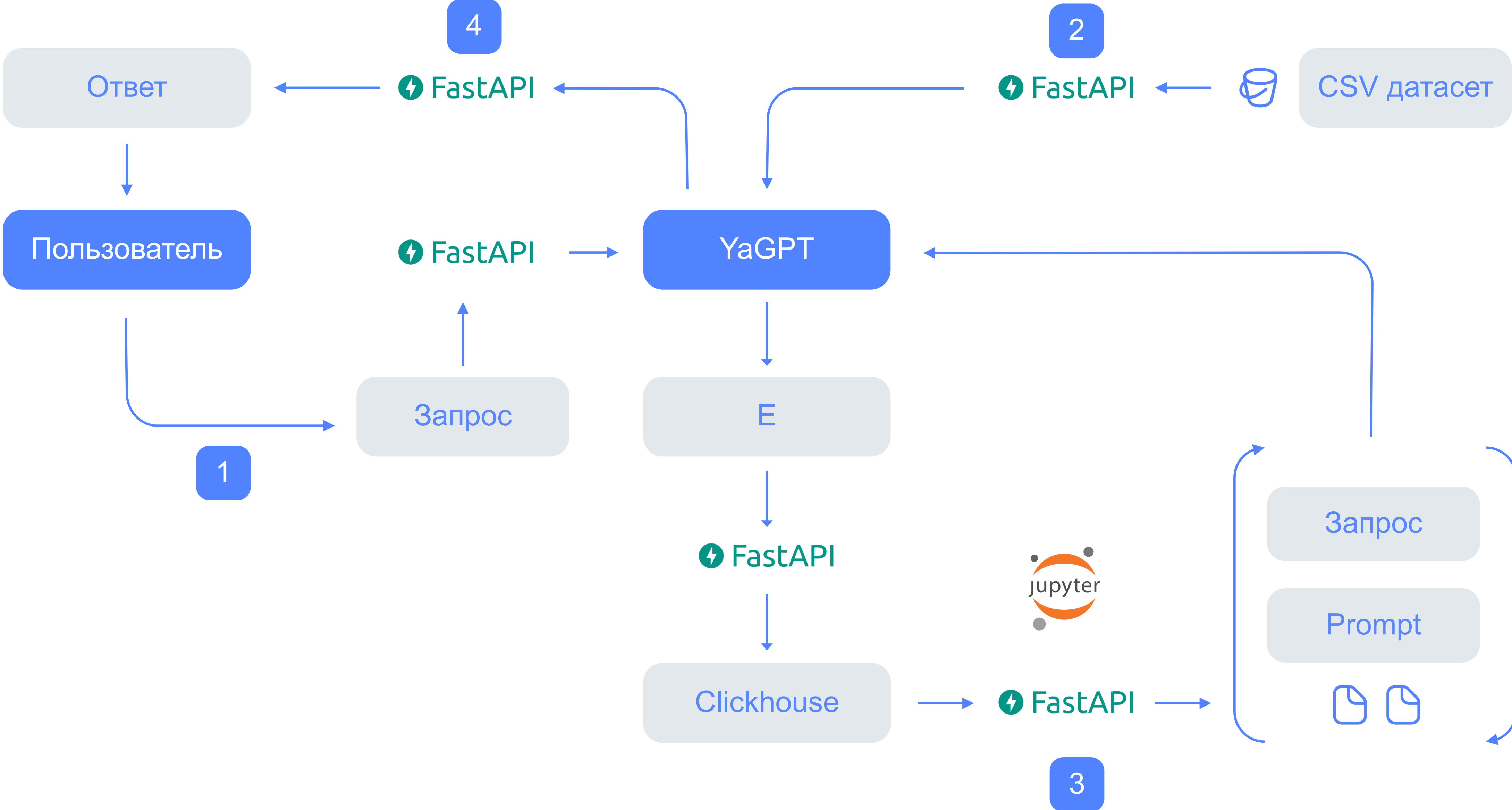
Файл может содержать не больше 10 000 вопросов и ответов. Максимальная длина запроса — 4000 символов, максимальная длина эталонного ответа — 2000 символов

The screenshot shows the 'Дообучение фундаментальной модели' (Fine-tuning fundamental model) page in the Yandex DataSphere interface. The left sidebar contains navigation links: 'Все сервисы' (All services), 'Сообщества' (Communities), 'Проекты' (Projects), 'Перенос проектов' (Project transfer), and 'Фундаментальные модели' (Fundamental models). The main content area has a header with the organization name 'organization-adolotov' and a breadcrumb trail 'dolotov / test1 / Модель'. Below the header, there's an information box stating that users can fine-tune the YandexGPT model on their own data. The form includes fields for 'Имя*' (Name), 'Описание' (Description), and a 'Темп обучения' (Learning rate) slider set to 0.001. The 'Данные для дообучения' (Data for fine-tuning) section shows a 'Файл с примерами*' (File with examples) field with a red dashed border and a message 'Поле не заполнено' (Field is empty) and 'JSON, максимальный размер файла — 100 МБ'. A 'Выбрать файл' (Choose file) button is present. At the bottom, there's an 'Инструкция' (Instructions) field. The footer contains links to 'Центр поддержки' (Support center), 'Настройки' (Settings), and 'Аккаунт' (Account).

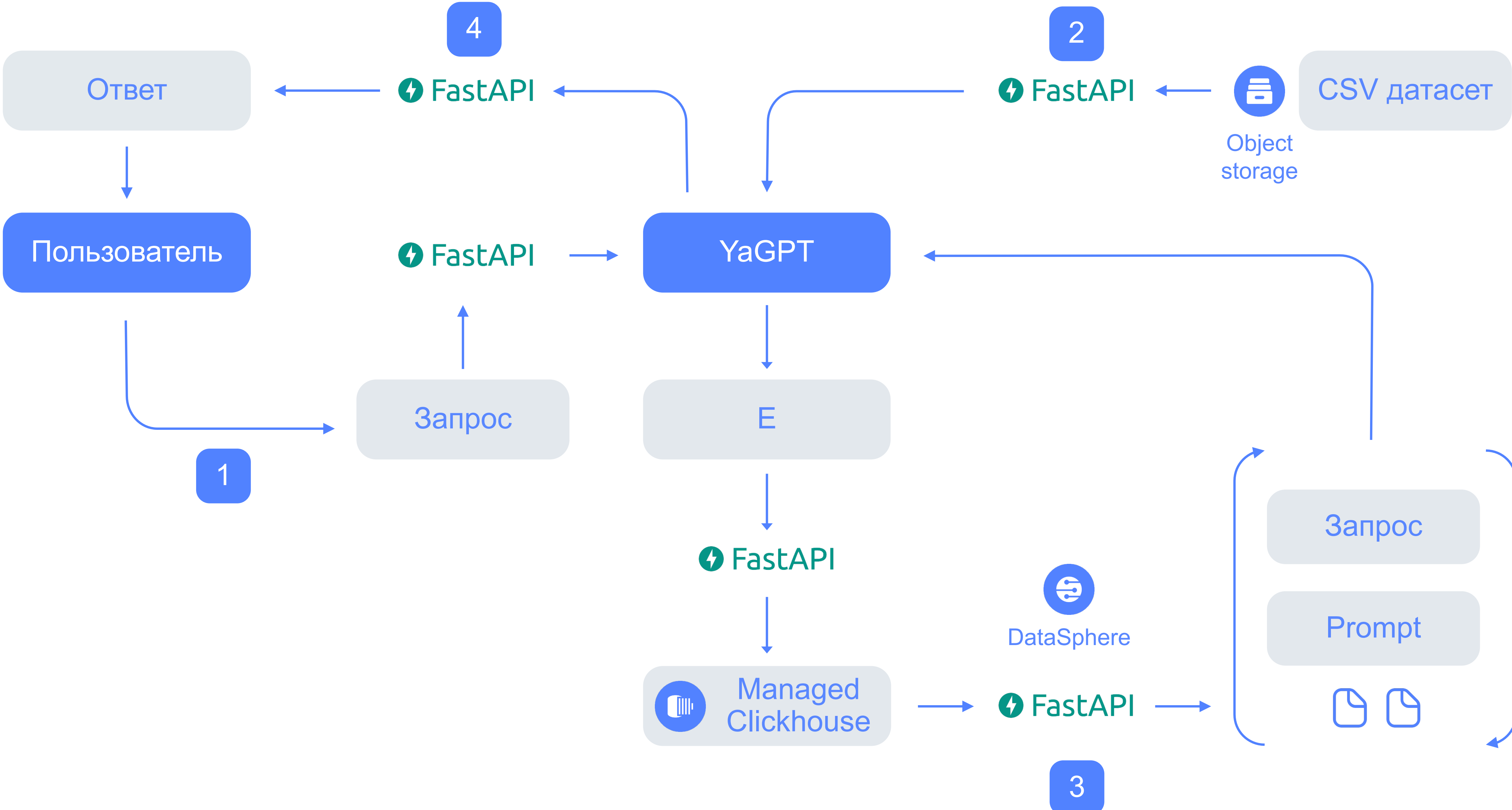
Архитектура RAG



Архитектура RAG on-premise



Архитектура RAG YC



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 оцифровка на стороне БД в момент вставки

3

ClickHouse — основные функции

`cosineDistance (vector1, vector2)`

`L2Distance (vector1, vector2)`

Сценарий 2 – операция “Обновить Pytorch”



Команда
разработчиков



Версия
библиотеки 2.3.0

Существующие фреймворки

Для продукт-менеджеров

[Langflow \(langflow.org\)](https://langflow.org)

[Flowise \(flowiseai.com\)](https://flowiseai.com)

[Rivet \(rivet.ironcladapp.com\)](https://rivet.ironcladapp.com)

[n8n \(n8n.io\)](https://n8n.io)

[create-llama \(npmjs.com/package/create-llama\)](https://npmjs.com/package/create-llama)

Существующие фреймворки

Для программистов

LLM chains

[Langchain](#)

[LlamaIndex](#)

[Haystack](#)

[Semantic Kernel from](#)

[Griptape](#)

Multi-Agent frameworks

[AutoGEN](#)

[AutoGPT](#)

[GPT-engineer](#)

[BabyAGI](#)

[Langroid](#)

[CrewAI](#)

Полезные ссылки

github.com/pytorch/pytorch/releases/tag/v2.3.0

python.langchain.com/v0.1/docs/integrations/vectorstores/clickhouse

python.langchain.com/v0.1/docs/integrations/text_embedding/yandex

clickhouse.com/blog/vector-search-clickhouse-p1

python.langchain.com/v0.1/docs/integrations/llms/yandex

Практика!



Кузьма Лешаков
Архитектор Data Platform,
Yandex Cloud



github.com/techkuz/fast-api-rag-clickhouse-yc