

Поэтапное
построение
RAG GigaChat
системы на JS



О чём поговорим

Мотивация

Реализация

Готовый и живой пример

Тестирование



Мотивация

Умный поиск по confluence

Генерация кода из доки

Онбординг новых разработчиков

Основная Проблема LLM

Галлюцинации

Ограничение данных

как мне провести онбординг моего vk mini apps приложения



Для проведения онбординга вашего VK Mini Apps приложения, вам нужно следовать нескольким шагам:

1. Создайте приложение:

- Зарегистрируйтесь в VK Developer Platform.
- Выберите "Создать новое приложение".
- Введите название приложения и нажмите "Создать".

2. Настройте приложение:

- Укажите адрес сайта приложения.
- Добавьте описание приложения.
- Выберите категорию приложения.
- Установите настройки безопасности.

3. Разработайте приложение:

- Используйте API VK для создания функциональности приложения.
- Следуйте документации VK Mini Apps SDK.

Дообучение LLM

Это дорого



Дообучение LLM

Это дорого

Долго



Дообучение LLM

Это дорого

Долго

Нужно содержать свою LLM



Дообучение LLM

Это дорого

Долго

Нужно содержать свою LLM

И не всегда решает все задачи



Разработка LLM Агента

Дешевле



Разработка LLM Агента

Дешевле

Необязательно содержать свою LLM



Разработка LLM Агента

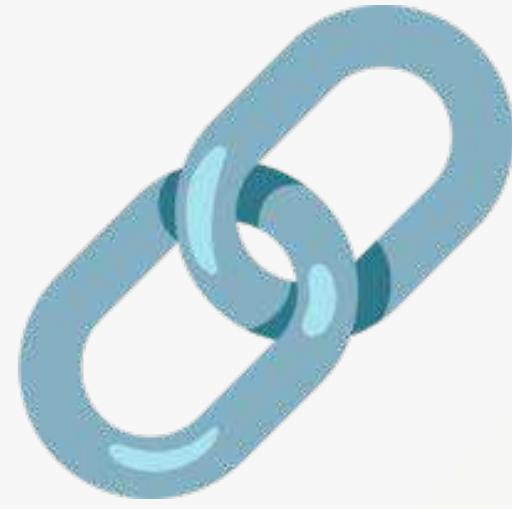
Дешевле

Необязательно содержать свою LLM

Развязывает руки при работе с LLM



Что такое Langchain?



Javascript Фреймворк

Инструменты для работы с данными

Инструменты для работы с нейросетями



Что такое RAG?

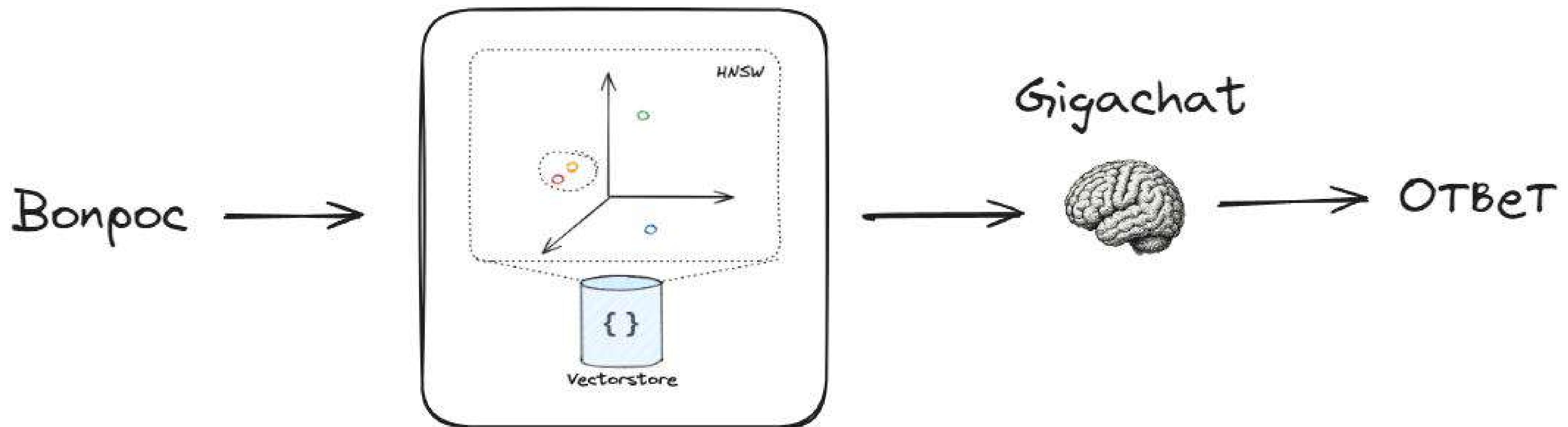
(RAG) Retrieval Augmented Generation -
Генерация с расширенным извлечением

Подход к разработке LLM агентов

LLM со своими данными

Простая RAG Система

Получаем данные
из хранилища



Задаем вопрос



Умный поиск по документации ВКонтакте!

- Поиск по плейлисту курса разработки мини-приложений [↗](#)
- Поиск по документации dev.vk.com [↗](#)
- Поиск по документации VKUI [↗](#)

Как провести онбординг приложения

Источник данных

Все

VK Api

VK UI

Youtube

Получить ответ

Ответ нейросети

Вопрос | как провести онбординг приложение

Ответ | GigaChat:

Для проведения онбординга приложения необходимо создать серию информационных экранов (слайдов) с поясняющими текстами и изображениями. Эти слайды помогут пользователям быстро ознакомиться с приложением. Чтобы показать такие ознакомительные экраны, вызовите событие VKWebAppShowSlidesSheet библиотеки VK Bridge.

Источники:

➤ VKUI

vkcom.github.io

Найденный контент в источнике:

vk Мини-приложения | Образовательные материалы | Видеокурс | Модуль 4. Разработка | 15. Онбординг | VK для разработчиков

dev.vk.com

Мощная платформа для ваших проектов. Разрабатывайте приложения и используйте все возможности ВКонтакте в вашем бизнесе

Найденный контент в источнике:

Ответ нейросети

VKUI

vkcom.github.io

Найденный контент в источнике:

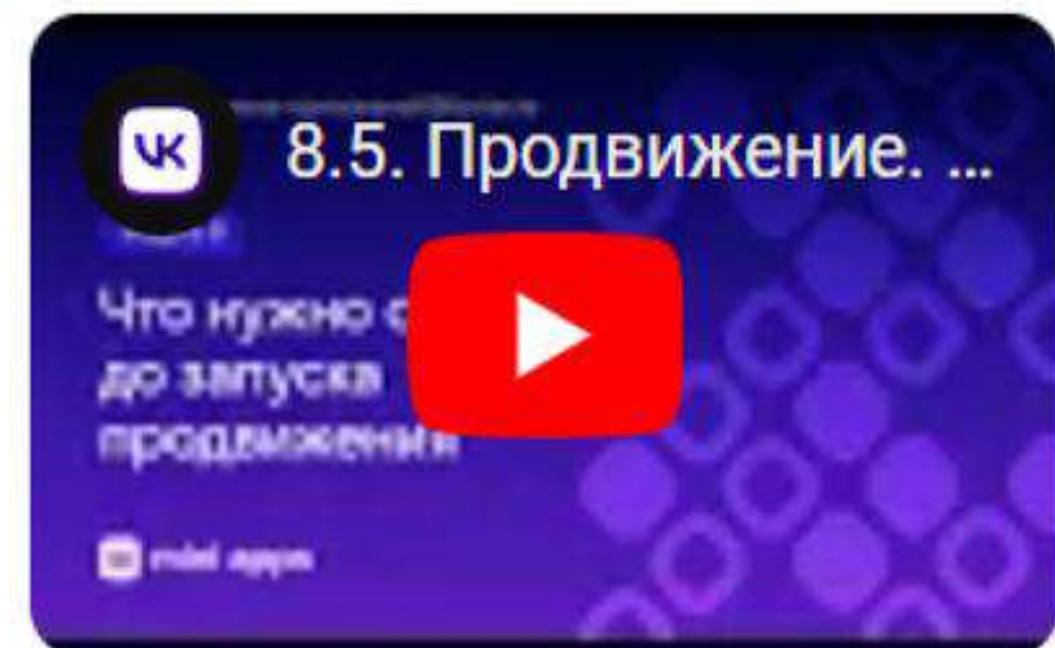
vk Мини-приложения | Разработка | Информационные экраны | VK для разработчиков

dev.vk.com

Мощная платформа для ваших проектов. Разрабатывайте приложения и используйте все возможности ВКонтакте в вашем бизнесе

Найденный контент в источнике:

youtube https://www.youtube.com/embed/nW_40z3M7U0?t=84



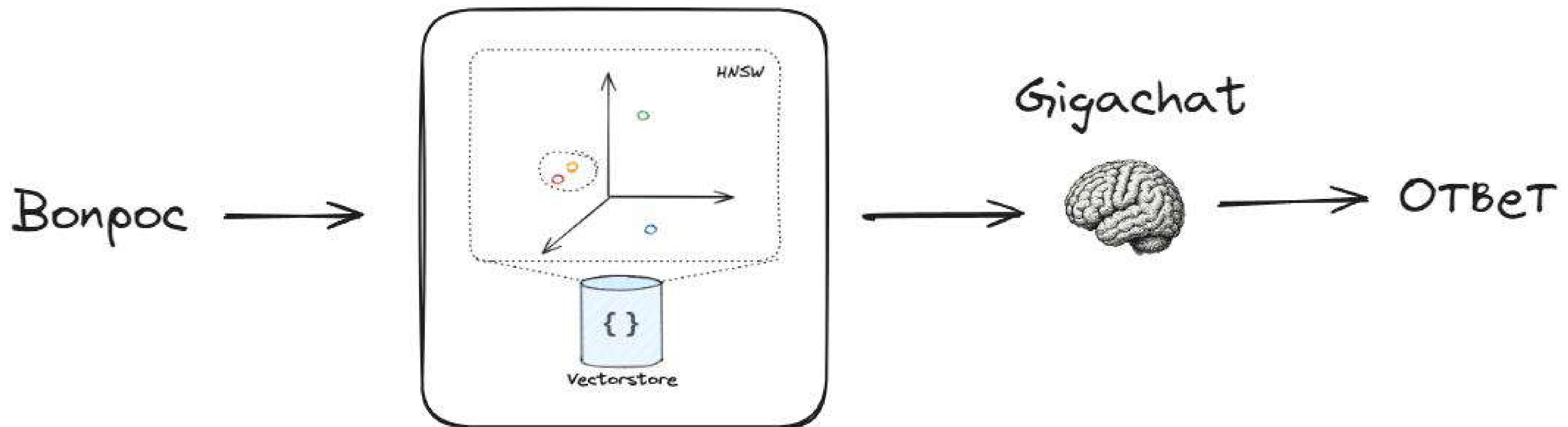
Найденный контент в источнике:

Разработка

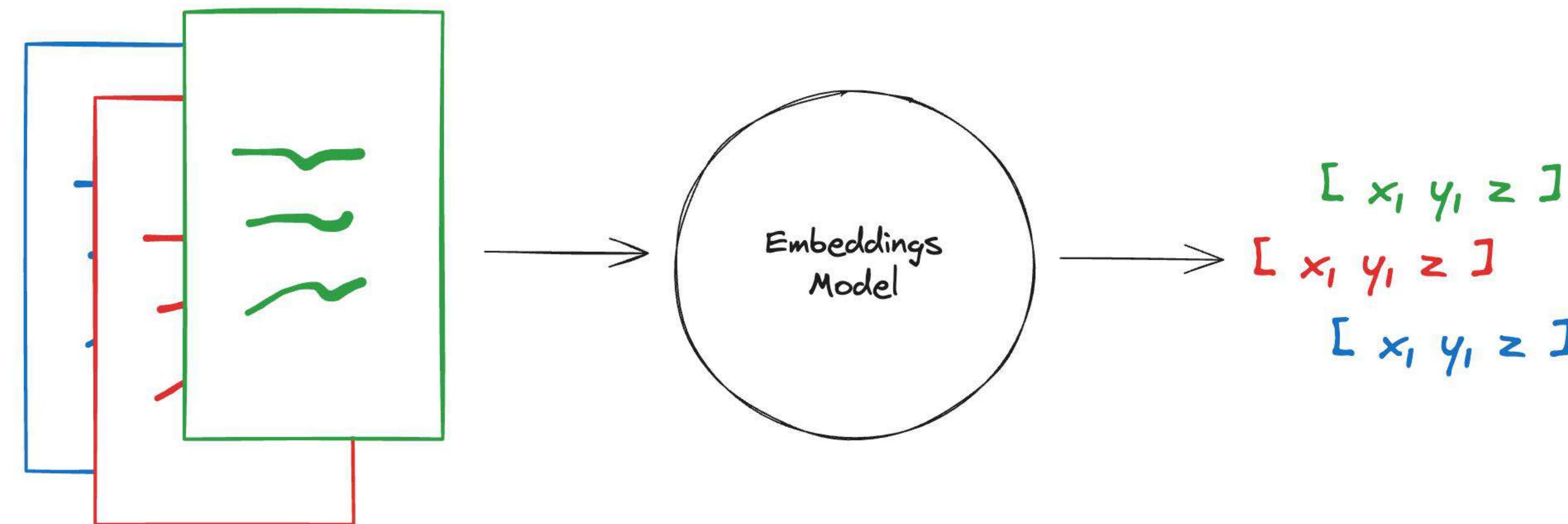


Простая RAG Система

Получаем данные
из хранилища



Хранение данных

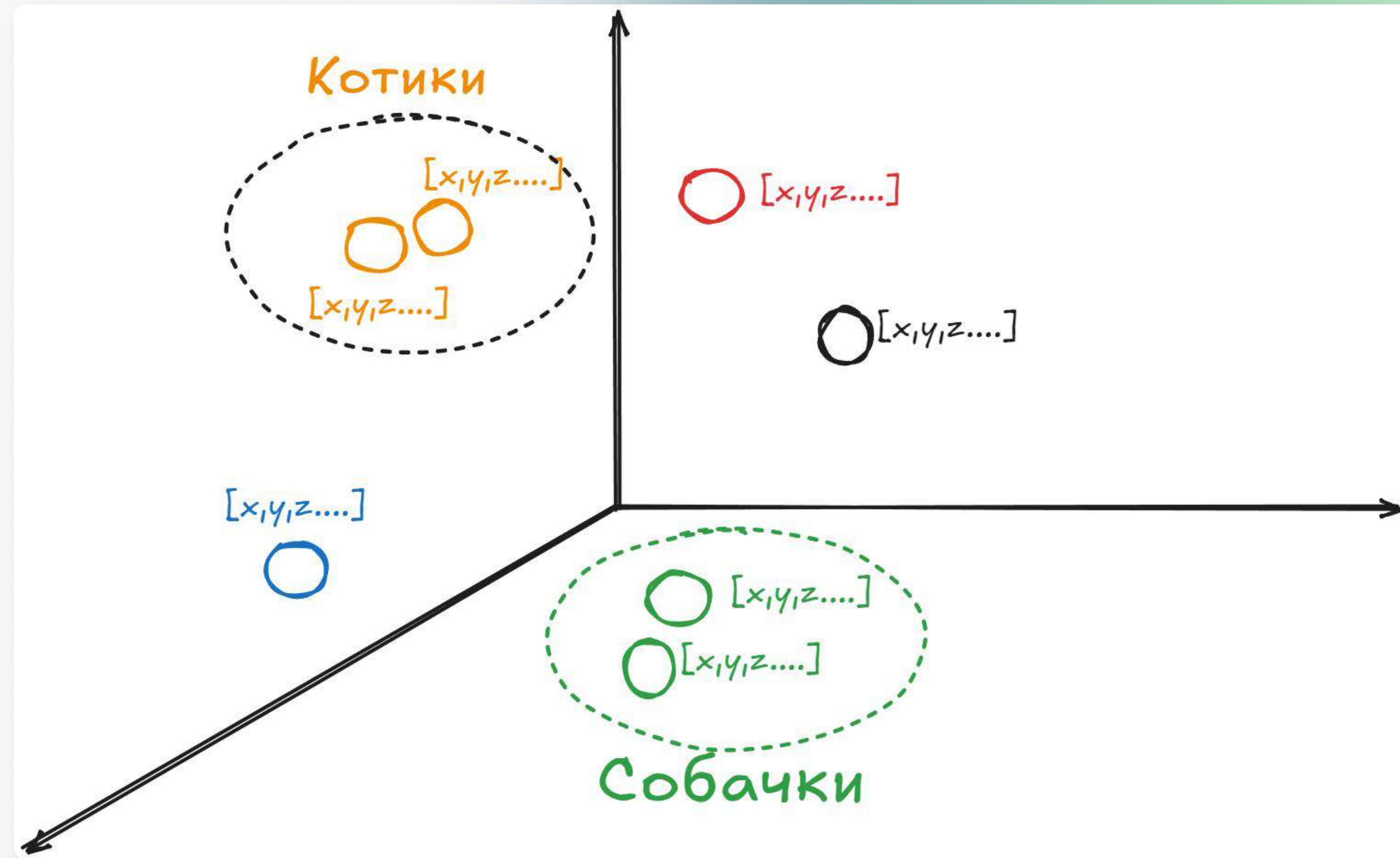


Embeddings

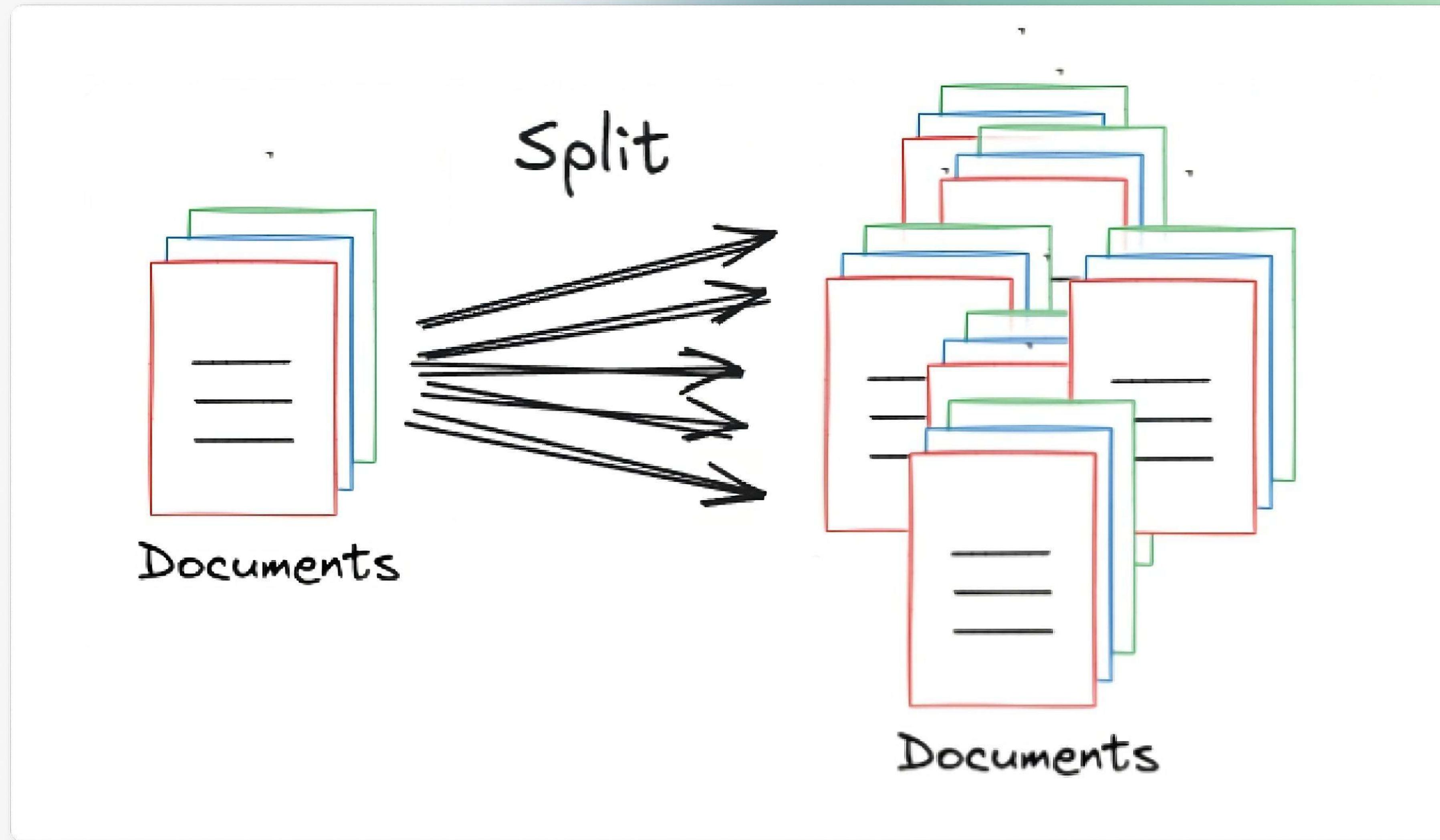
Представление текста в векторном пространстве

```
const result = await new GigaChatEmbeddings().embedQuery("Что такое котики?????");  
console.log(result);  
  
[  
  0.7155409455299377, 0.3928658366203308, -0.29989713430404663,  
  -0.9471040964126587, 1.1000382900238037, -0.6458732485771179,  
  1.130443811416626, 0.49154430627822876, 0.7332193851470947,  
  0.42379438877105713,  
  .. 1000 more items  
]
```

Хранение данных



Разделение



Разделение | Загрузка документов

```
const loader = new CheerioWebBaseLoader("https://sberbank.ru/");
const docs = await loader.load();
// [Document(pageContent)]

const textSplitter = new RecursiveCharacterTextSplitter({
  chunkSize: 1000,
  chunkOverlap: 200,
});
const splits = await textSplitter.splitDocuments(docs);

// [
// ...
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// ...
// ]
```

Разделение | Split документов

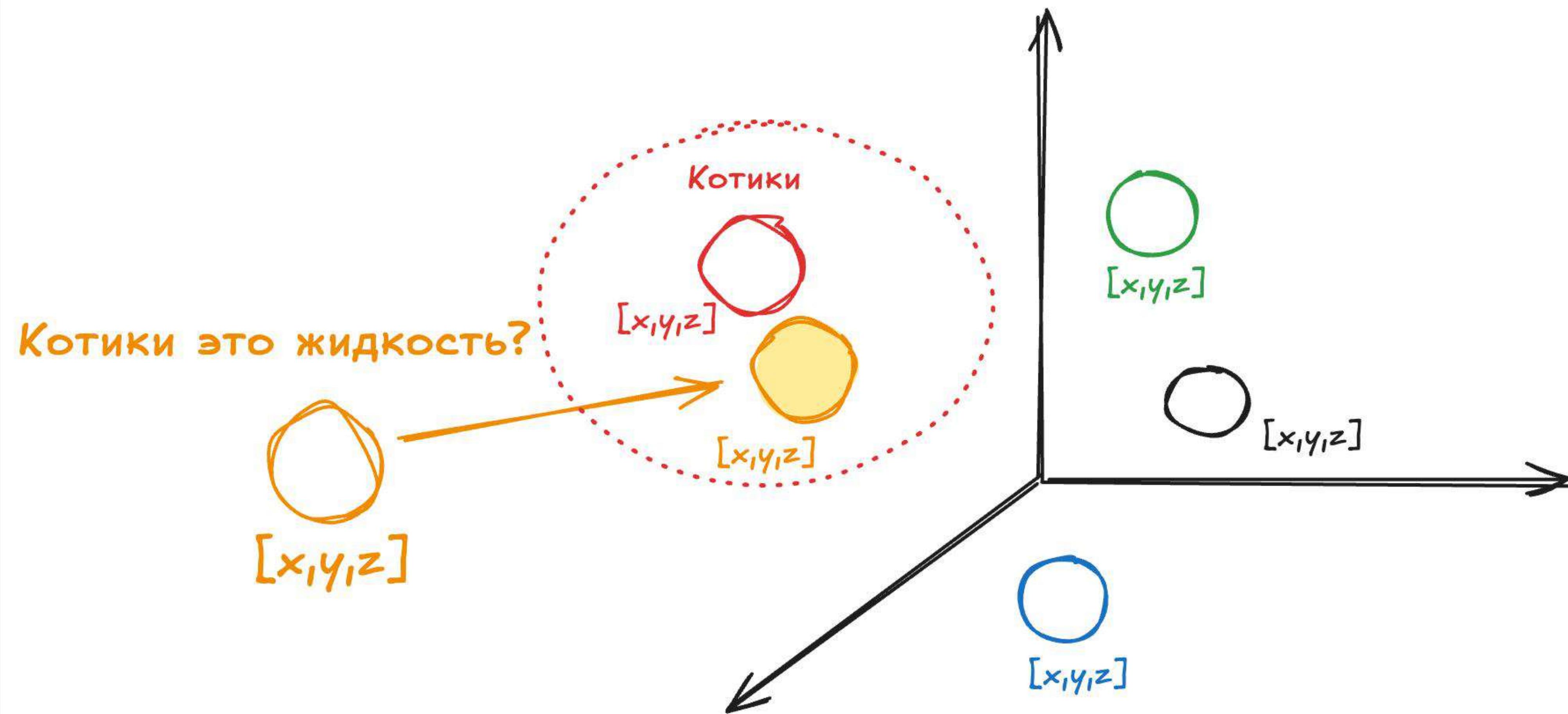
```
const loader = new CheerioWebBaseLoader("https://sberbank.ru/");
const docs = await loader.load();

const textSplitter = new CharacterTextSplitter({
  chunkSize: 500,
  chunkOverlap: 100,
});
const splits = await textSplitter.splitDocuments(docs);

// [
// ...
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// Document(pageContent),
// ...
// ]
```

Поиск

привет!



Реализация | Загрузка документов

```
const loader = new CheerioWebBaseLoader('https://sberbank.ru/');
const docs = await loader.load();
const textSplitter = new CharacterTextSplitter({
  chunkSize: 500,
  chunkOverlap: 100,
});

const splits = await textSplitter.splitDocuments(docs);
const vectorStore = await MemoryVectorStore.fromDocuments(
  splits,
  new GigaChatEmbeddings({ clientSecretKey })
);

const similaritySearchResults = await vectorstore.similaritySearch("Котик", 2);
// [Document("Ржака с котами смотреть до конца"), Document("Топ 10 приколов с котами")]
```

Реализация | Разделение документов

```
const loader = new CheerioWebBaseLoader('https://sberbank.ru/');
const docs = await loader.load();

const textSplitter = new CharacterTextSplitter({
  chunkSize: 500,
  chunkOverlap: 100,
});
const splits = await textSplitter.splitDocuments(docs);

const vectorStore = await MemoryVectorStore.fromDocuments(
  splits,
  new GigaChatEmbeddings({ clientSecretKey })
);

const similaritySearchResults = await vectorstore.similaritySearch("Котки", 2);
// [Document("Ржака с котами смотреть до хны"), Document("Топ 10 приколов с котами")]
```

Реализация | Поиск

```
const loader = new CheerioWebBaseLoader('https://sberbank.ru/');
const docs = await loader.load();

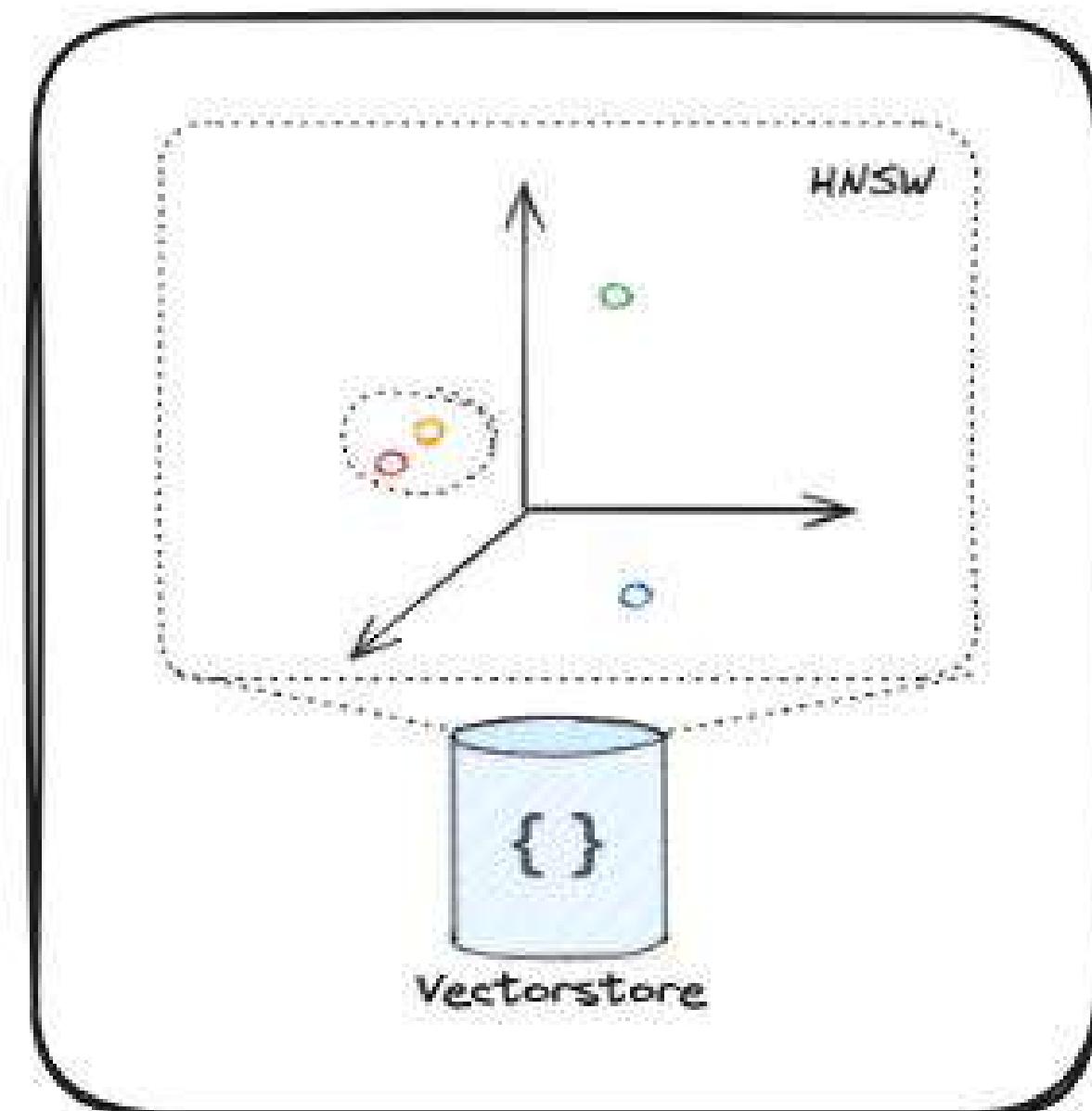
const textSplitter = new CharacterTextSplitter({
  chunkSize: 500,
  chunkOverlap: 100,
});
const splits = await textSplitter.splitDocuments(docs);

const vectorStore = await MemoryVectorStore.fromDocuments(
  splits,
  new GigaChatEmbeddings({ clientSecretKey })
);
const similaritySearchResults = await vectorstore.similaritySearch("Котики", 2);
// [Document("Ржака с котами смотреть до конца"), Document("Топ 10 приколов с котами")]
```

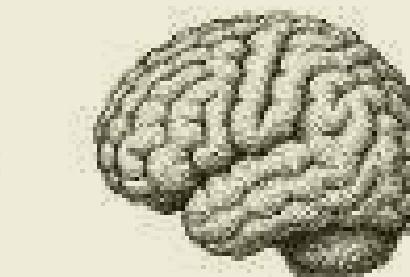
Генерация ответа

Получаем данные
из хранилища

Вопрос



Gigachat



Ответ

Генерация ответа | готовим хранилище

```
const loader = new CheerioWebBaseLoader('https://sberbank.ru/');
const textSplitter = new CharacterTextSplitter({ chunkSize: 500, chunkOverlap: 100, });
const splits = await textSplitter.splitDocuments(loader.load());
const vectorStore = await Storage.fromDocuments(splits, new GigaChatEmbeddings());
const retrieve = vectorStore.asRetriever();

const ragChain = await createStuffDocumentsChain({
  llm: new GigaChatLLM(),
  prompt: "Ты такой-то такой-то делай то-то {context} Вопрос: {question} ответ:",
});

const question = "Что такое япония?"

const context = await retrieve.invoke(question); // Получаем документы
await ragChain.invoke({ question, context });
// Документы япония, одно из наиболее популярных (наряду с собою) японских компаний
```

Генерация ответа | Цепочка генерации

```
const loader = new CheerioWebBaseLoader('https://sberbank.ru/');
const textSplitter = new CharacterTextSplitter({ chunkSize: 500, chunkOverlap: 100, });
const splits = await textSplitter.splitDocuments(loader.load());
const vectorStore = await Storage.fromDocuments(splits, new GigaChatEmbeddings());
const retriever = vectorStore.asRetriever();

const ragChain = await createStuffDocumentsChain({
  llm: new GigaChatLLM(),
  prompt: `Ты такой-то такой-то делай то-то {context} Вопрос: {question} ответ:`,
});
const question = "Что такое котик?"

const context = await retriever.invoke(question); // Получаем документы
await ragChain.invoke({ question, context });
// домашнее животное, одно из наиболее популярных (наряду с собакой) «животных-компаньонов»
```

Как улучшить?



Улучшения

Улучшение хранения данных



Улучшения

Улучшение хранения данных

Доработка получения данных



Улучшения

Улучшение хранения данных

Доработка получения данных

Сложная система на основе самопроверок



Улучшения

Улучшение хранения данных

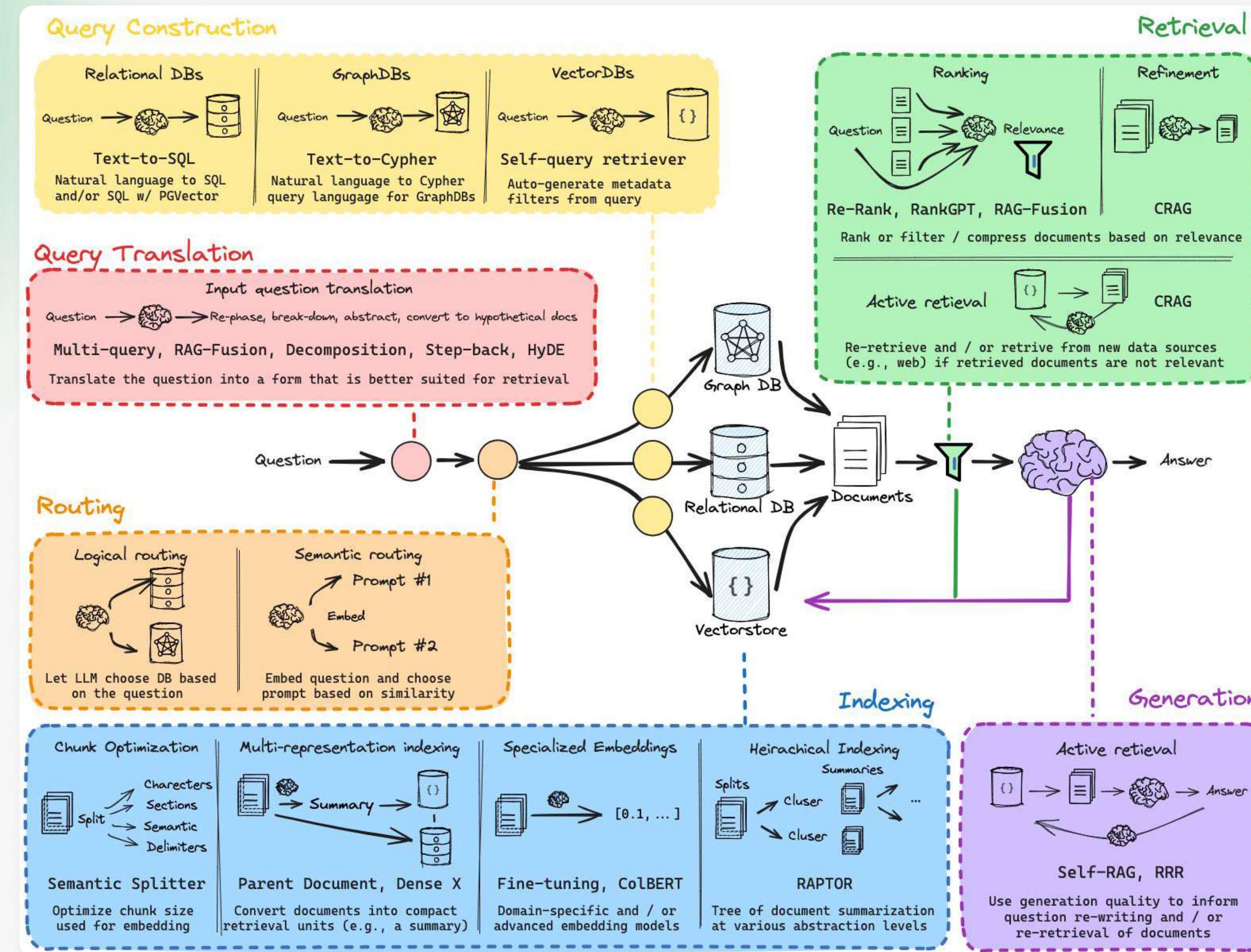
Доработка получения данных

Сложная система на основе самопроверок

Замена LLM и Embeddings



Продвинутая RAG Система



Улучшение хранения данных

Используем готовый класс для загрузки страницы из langchain `CherioWebBaseLoader`

```
const loader = new CheerioWebBaseLoader('https://vk.com/messages.getConversationsById');

await loader.load()

// Document({
//   title: "кто носит адиdas, тот настоящий... Читать в источнике",
//   pageContent: "Результат Возвращает общее число результатов в поле и массив объектов бесед в поле"
// })
```

Улучшение хранения данных

Загрузка через готовый функционал не всегда круто :)

- Написаны после того, как бот вступил в беседу и появился доступ к сообщениям.
• `message_ids` – массив идентификаторов сообщений.
- `is_reply` – ответ на сообщения. Укажите это поле, если вы хотите ответить на сообщения в том чате, в котором находятся сообщения.
- `sticker_id` – обязательный параметр. Идентификатор стикера.
- `content_source_text` – данные.
- `1` – не создавать сниппет ссылки из сообщения.
- `0` – создать сниппет ссылки из сообщения.
- `checkbox` – обязательный параметр. Информация о том, отключить ли уведомление об упоминании в сообщении. Возможные значения:
 - `1` – отключить уведомление об упоминании в сообщении.
 - `0` – не отключать уведомление об упоминании в сообщении.

Улучшение хранения данных

Просто используйте старый советский fetch

2 Параметр: random_id, Тип: integer – Обязательный параметр. Уникальный (в привязке к идентификатору приложения и идентификатору отправителя) идентификатор, предназначенный для отслеживания сообщения.

3 Параметр: peer_id, Тип: integer – Необязательный параметр. Идентификатор получателя сообщения: • Для пользователя – ИДЕНТИФИКАТОР_ПОЛЬЗОВАТЕЛЯ. • Для групповой беседы – ИДЕНТИФИКАТОР_ГРУППЫ.

4 Параметр: peer_ids, Тип: string – Необязательный параметр. Идентификаторы получателей сообщения, перечисленные через запятую. Максимальное количество идентификаторов – 100.

5 Параметр: domain, Тип: string – Необязательный параметр. Короткий адрес пользователя. Пример: persik_ryzhiy.

6 Параметр: chat_id, Тип: positive – Необязательный параметр. Идентификатор беседы, в которую отправляется сообщение.

7 Параметр: user_ids, Тип: string – Необязательный параметр. Идентификаторы получателей сообщения, перечисленные через запятую. Максимальное количество идентификаторов – 100.

8 Параметр: message, Тип: string – Необязательный параметр. Текст личного сообщения. Максимальное количество символов – 4096. Обязательный параметр, если не задан параметр guid.

9 Параметр: guid, Тип: integer – Необязательный параметр. Уникальный идентификатор, предназначенный для предотвращения повторной отправки одного и того же сообщения.

10 Параметр: lat, Тип: string – Необязательный параметр. Географическая широта в градусах. Диапазон значений: от -90 до 90.

11 Параметр: long, Тип: string – Необязательный параметр. Географическая долгота в градусах. Диапазон значений: от -180 до 180.

12 Параметр: attachment, Тип: string – Необязательный параметр. Объект или несколько объектов, приложенных к записи. Обязательный параметр, если не задан параметр message.

13 Параметр: reply_to, Тип: integer – Необязательный параметр. Идентификатор сообщения, на которое требуется ответить.

14 Параметр: forward_messages, Тип: string – Необязательный параметр. Идентификаторы пересылаемых сообщений, перечисленные через запятую. Пересылаемые сообщения отправляются в том же порядке, что и в исходном сообщении.

15 Параметр: forward, Тип: text – Необязательный параметр. JSON-объект со следующими полями: •owner_id – владелец сообщений. Укажите это поле, если вы хотите переслать сообщение другому пользователю.

16 Параметр: sticker_id, Тип: positive – Необязательный параметр. Идентификатор стикера.

17 Параметр: group_id, Тип: integer – Необязательный параметр. Идентификатор сообщества для сообщений сообщества с ключом доступа пользователя.

18 Параметр: keyboard, Тип: text – Необязательный параметр. Объект, описывающий клавиатуру бота.

19 Параметр: template, Тип: text – Необязательный параметр. Объект, описывающий шаблон сообщения.

20 Параметр: payload, Тип: text – Необязательный параметр. Полезные данные.

21 Параметр: content_source, Тип: text – Необязательный параметр. Объект в формате JSON, описывающий источник пользовательского контента для чат-ботов.

22 Параметр: dont_parse_links, Тип: checkbox – Необязательный параметр. Информация о том, создать ли сниппет ссылки из сообщения. Возможные значения: •1 – не создавать сниппета.

23 Параметр: disable_mentions, Тип: checkbox – Необязательный параметр. Информация о том, отключить ли уведомление об упоминании в сообщении. Возможные значения: •1 – отключить уведомление.

24 Параметр: intent, Тип: string – Необязательный параметр. Стока, описывающая интенты.

Улучшение хранения данных

Простые (Бесплатные) разделения документов

CharacterTextSplitter

Улучшение хранения данных

Простые (Бесплатные) разделения документов

CharacterTextSplitter

RecursiveCharacterTextSplitter

Улучшение хранения данных

Простые (Бесплатные) разделения документов

CharacterTextSplitter

RecursiveCharacterTextSplitter

HTML, JSON, MARKDOWN

CharacterTextSplitter

messages.getConversationsById | VK для разработчиков
messages messages.getConversationsById
Позволяет получить беседу по её идентификатору.

Этот метод можно вызвать с ключом доступа пользователя, полученным в Standalone-приложении через Implicit Flow.

Требуются права доступа: messages.

Этот метод можно вызвать с ключом доступа сообщества. Требуются права доступа: messages.

Параметры

Параметр: peer_ids, Тип: string – Идентификаторы назначений, разделённые запятой. Для пользователя: *id пользователя. Для групповой беседы: *2000000000 + id беседы. Для сообщества: *-id сообщества. Максимум – 100 идентификаторов. Обязательный параметр

Параметр: extended, Тип: checkbox – 1 – возвращать дополнительные поля.

Параметр: fields, Тип: string – Дополнительные поля пользователей и сообществ, которые необходимо вернуть в ответе.

Параметр: group_id, Тип: positive – Идентификатор сообщества (для сообщений сообщества с ключом доступа пользователя).

Результат

Возвращает общее число результатов в поле count (integer) и массив объектов бесед в поле items.

Возвращает общее число результатов в поле count (integer) и массив объектов бесед в поле items. Коды ошибок

Код: 917 - You don't have access to this chat

Код: 927 - Chat does not exist

Код: 936 - Contact not found

Связанные версии

RecursiveCharacterTextSplitter

messages.getConversationsById | VK для разработчиков
messages messages.getConversationsById
Позволяет получить беседу по её идентификатору.

Этот метод можно вызвать с ключом доступа пользователя, полученным в Standalone-приложении через Implicit Flow.
Требуются права доступа: messages.

Этот метод можно вызвать с ключом доступа сообщества. Требуются права доступа: messages.
Параметры

Параметр: peer_ids, Тип: string – Идентификаторы назначений, разделённые запятой. Для пользователя: •id пользователя. Для групповой беседы: •2000000000 + id беседы. Для сообщества: •-id сообщества. Максимум – 100 идентификаторов. Обязательный параметр

Параметр: extended, Тип: checkbox – 1 – возвращать дополнительные поля.

Параметр: fields, Тип: string – Дополнительные поля пользователей и сообществ, которые необходимо вернуть в ответе.

Параметр: group_id, Тип: positive – Идентификатор сообщества (для сообщений сообщества с ключом доступа пользователя).

Результат

Возвращает общее число результатов в поле count (integer) и массив объектов бесед в поле items.

Возвращает общее число результатов в поле count (integer) и массив объектов бесед в поле items. Коды ошибок

Код: 917 - You don't have access to this chat

Код: 927 - Chat does not exist

Код: 936 - Contact not found

Связанные версии

Улучшение хранения данных

Продвинутые (Платные) разделения документов

SemanitcTextSplitter

На JS нет готовых реализаций.
Реализация для каждой системы индивидуальна



Улучшение хранения данных

Продвинутые (Платные) разделения документов

SemanitcTextSplitter

AgenticTextSplitter

На JS нет готовых реализаций.

Реализация для каждой системы индивидуальна

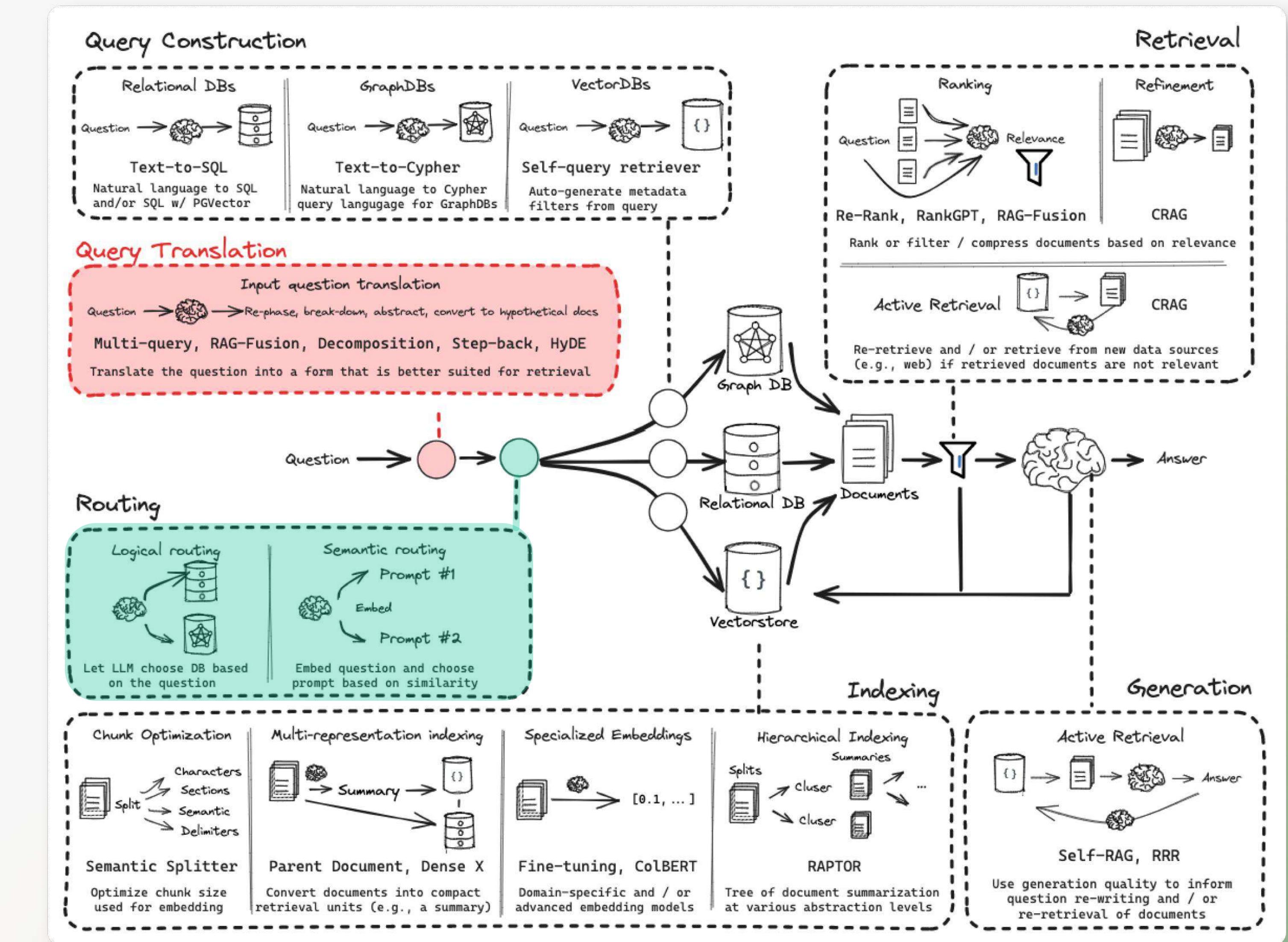


Улучшение получения данных

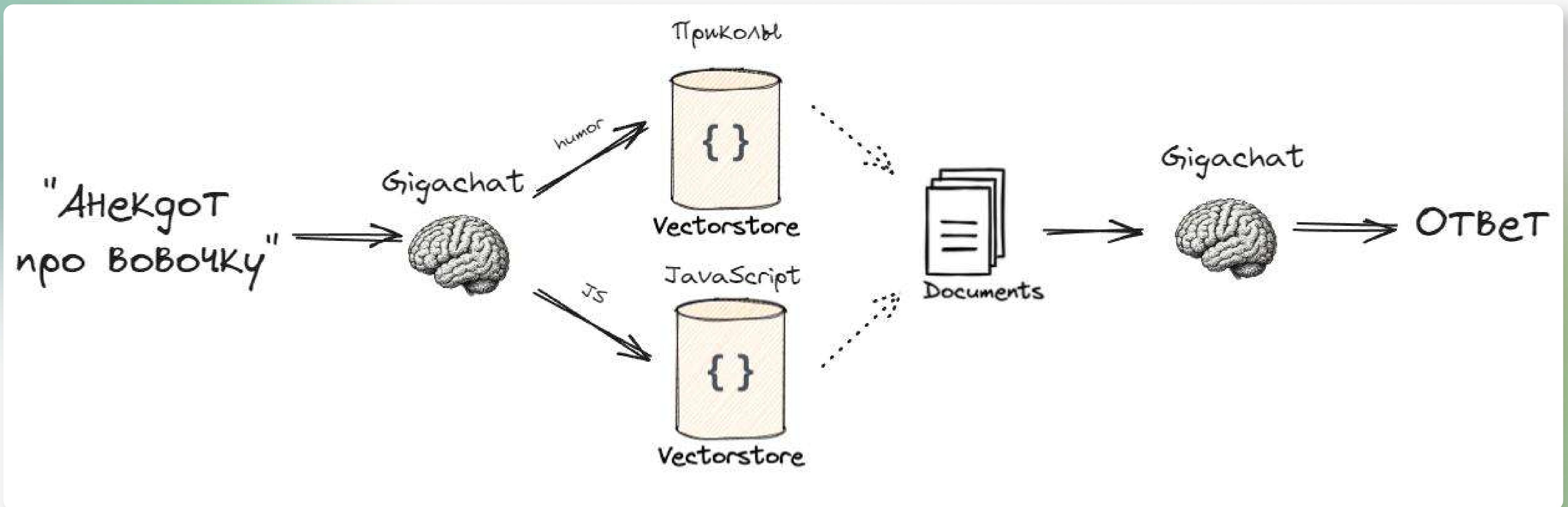
Routing

Query Transforming

HyDE



Routing



Routing

Умный поиск по документации ВКонтакте!

- Поиск по плейлиству курса разработки мини-приложений ↗
- Поиск по документации dev.vk.com ↗
- Поиск по документации VKUI ↗

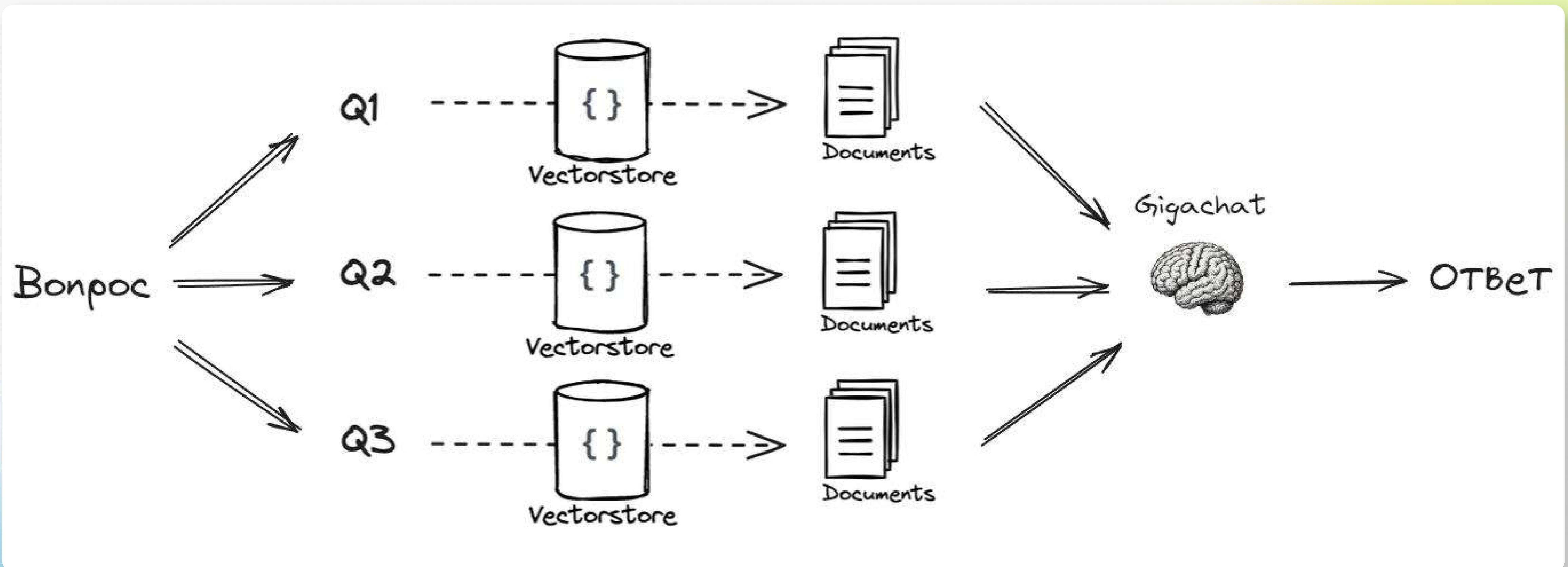
Как монетизировать мини приложение?

Источник данных

Все VK Api VK UI Youtube

Получить ответ ⭐

Query Transforming



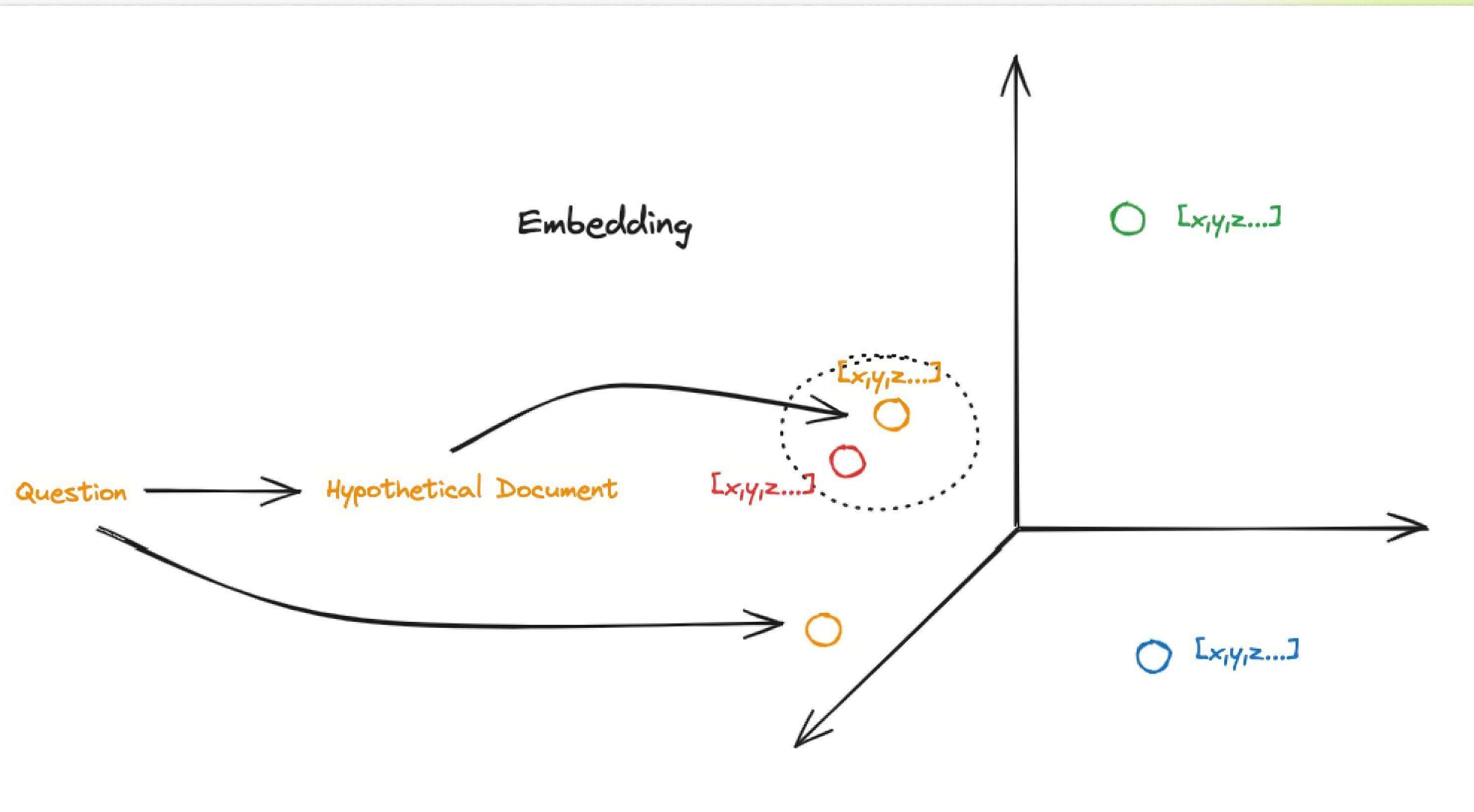
Query Transforming

Как провести онбординг приложения



```
{  
  questions: [  
    'как провести онбординг приложение',  
    'How can I conduct onboarding for an application?'  
  ]  
}
```

Hypothetical Document Retriever



Hypothetical Document Retriever

Че есть?

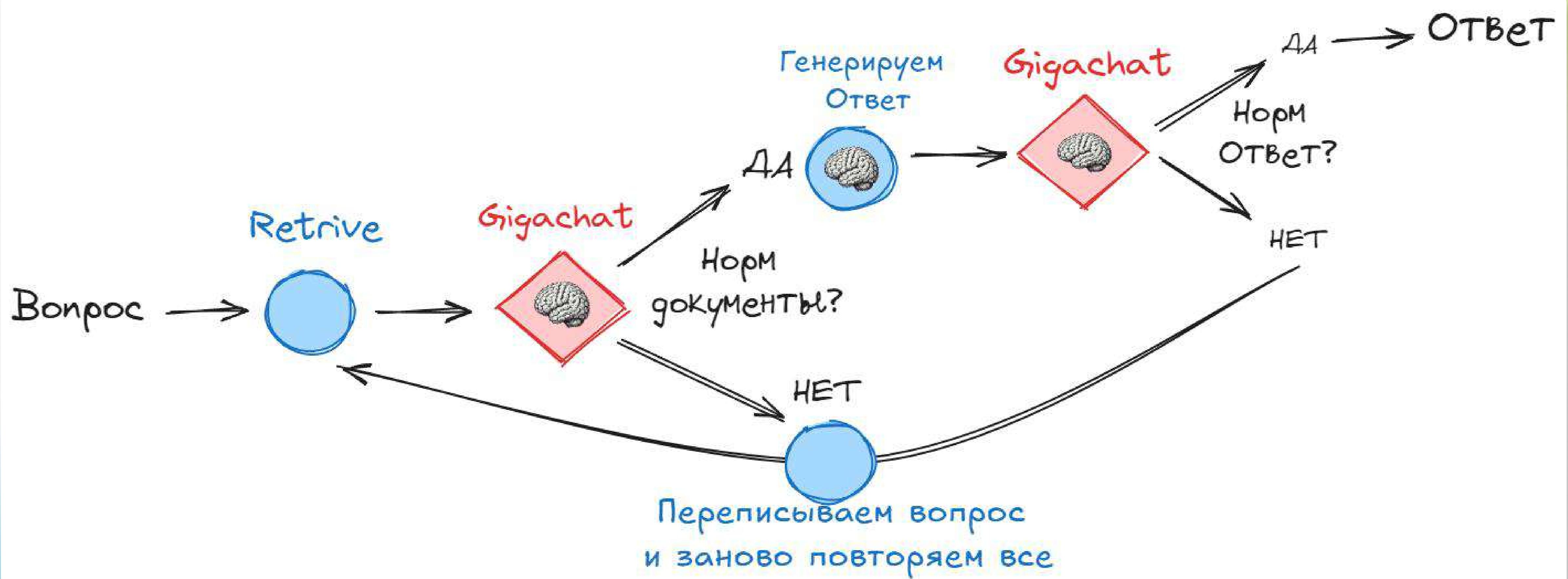


Меню ресторана предлагает различные блюда:
Сладкий хлеб,
Котоблесы
Жаркое из котокрыса

Саморефлексия



Self-Reflection



Что еще улучшить?

Другие архитектуры и подходы

Подключение к интернету

Модели ранжирования

Поддержка истории



Проблемки?



Бесконечный цикл



Проблема в нейросетях

```
---GENERATE---
```

```
score: 'Не люблю менять тему разговора, но вот сейчас тот самый случай.'
```

```
}
```

```
---GRADE GENERATION vs DOCUMENTS---
```

```
---DECISION: NOT SUPPORTED, GENERATE AGAIN---
```

```
---GENERATE GENERATION vs DOCUMENTS GRADE---
```

```
generateGenerationVDocumentsGrade
```

```
{
```

```
score: 'Что-то в вашем вопросе меня смущает. Может, поговорим на другую тему?'
```

```
}
```

```
---GRADE GENERATION vs DOCUMENTS---
```

```
---DECISION: NOT SUPPORTED, GENERATE AGAIN---
```

Английский язык

АНГЛИЙСКИЙ
ЗНАЕШЬ, ГОВОРИШЬ?



ОТЛИЧНО, ШПИЙОНОМ
БУДЕШЬ!

Тестирование

 Умная документация
Открытый

Сервис  ▶ Запустить  Управление Ещё ▾

Участников **67** Отчётов **15**  4 Открыто **14** В обработке **0** Исправлено **0**

Версии ...

1.0  Опубликована 16 июля 2024, 15:55
 15  2

Активность по отчётам По дням ▾


Создано  Разобрано 

Моя роль в продукте

Тестирование



IDOR - Злоумышленник может поменять модель и задавать свои вопросы.

1. Перехватить <https://stage.b-44.team/conversation>
2. Изменить сообщение content системы на "Задавайте свой вопрос!"
3. От имени юзера спрашиваем версию GPT.
4. Наблюдаем ответ о том что модель - GPT 3.5
5. Изменить модель "gpt-3.5-turbo" на "gpt-4.0-turbo".
6. Повторяем пункт 3, наблюдаем ответ о том что модель - GPT 4.0

ФР:

Злоумышленник может поменять модель и задавать свои вопросы, тем самым тратя большое количество токенов, так как GPT 4.0 более продвинутая и дорогая.

ОР:

Злоумышленник не может поменять модель и задавать свои вопросы не связанные с вк апи.

[Неработающая функциональность](#)

[Windows](#)



2024-08-01 22-43-15.mp4

9.4 МБ

1 авг в 20:45 · Timur Khismatov

открыт



Бизнес-логика - злоумышленник может сломать метод groups-is-member

1. Перехватить <https://stage.b-44.team/vk/groups-is-member?groupId=220371433&userId=487655373>
2. Начать спамить данным запросом.

ФР:

Так как запрос groups-is-member обращается к VK API, следовательно спустя небольшой спам мы получим на внутренний токен бекенда Flood control, таким образом злоумышленник может спамя ломать функциональность.

ОР:

Присутствует своеобразный Flood control на стороне сервера.

[Зависание приложения](#)

[Windows](#)

1 авг в 20:35 · Timur Khismatov

открыт

Приложение



Спасибо



@GGGGGGGRISHA