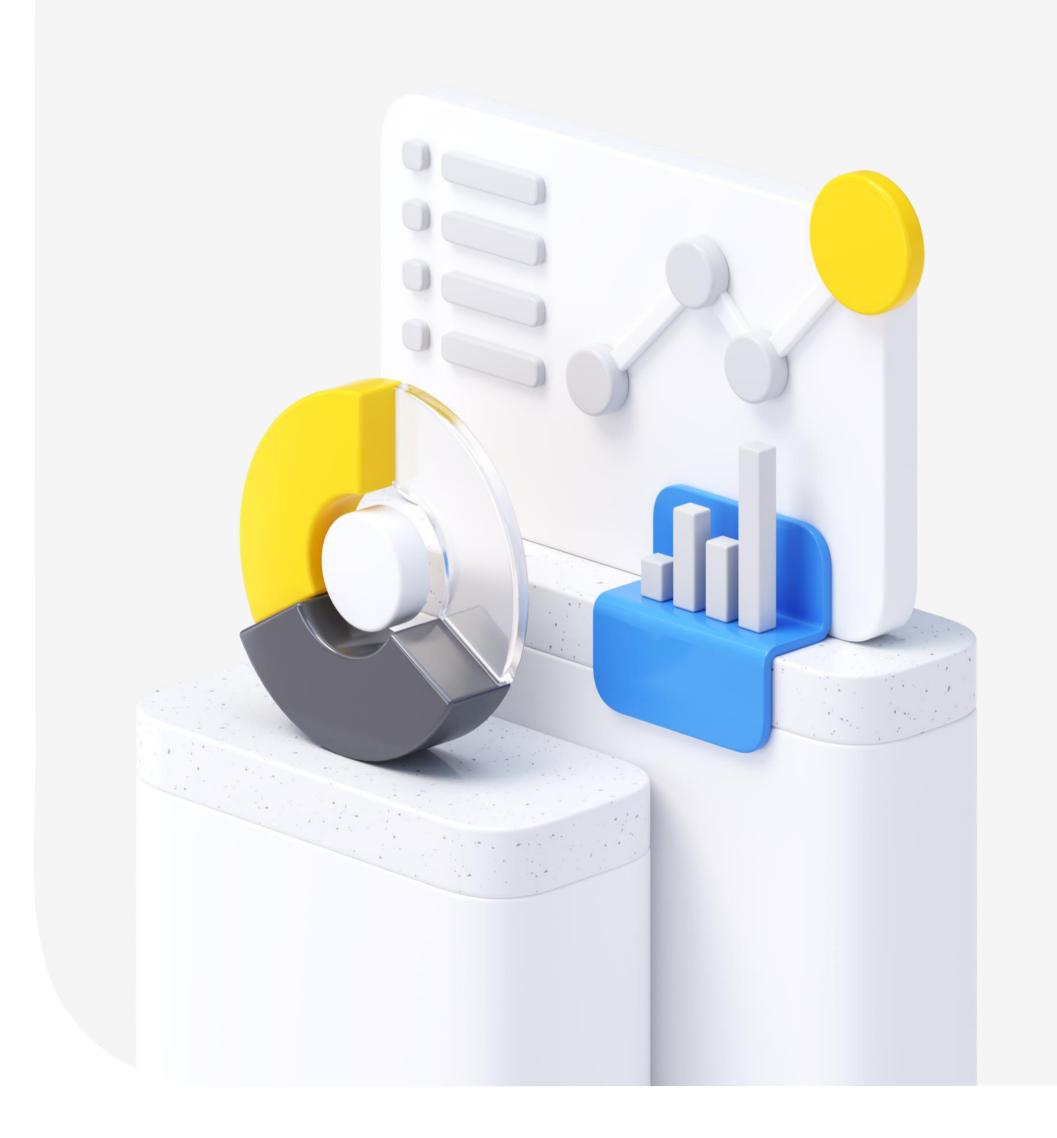
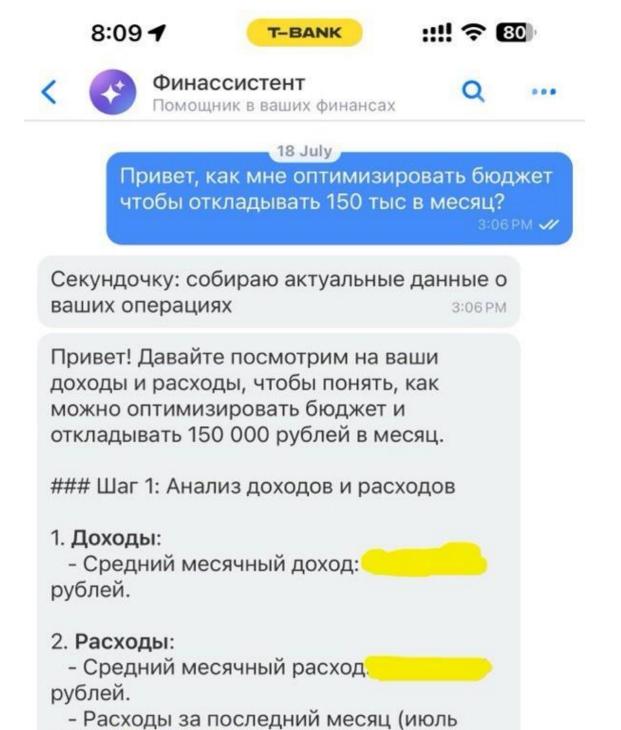


LLM Applications



Кейс#2 фин ассистент



рублей.

рублей.

80,63 рублей

Шаг 2: Определение текущего

- В июле 2024 года у вас был

Разница: 46 519,37 рублей.

положительный баланс: доходы

- Чтобы откладывать 150 000 рублей в

месяц, нужно сократить расходы или







2024):

состояния



увеличить доходы на (150 000 -),37).

рублей, расходы

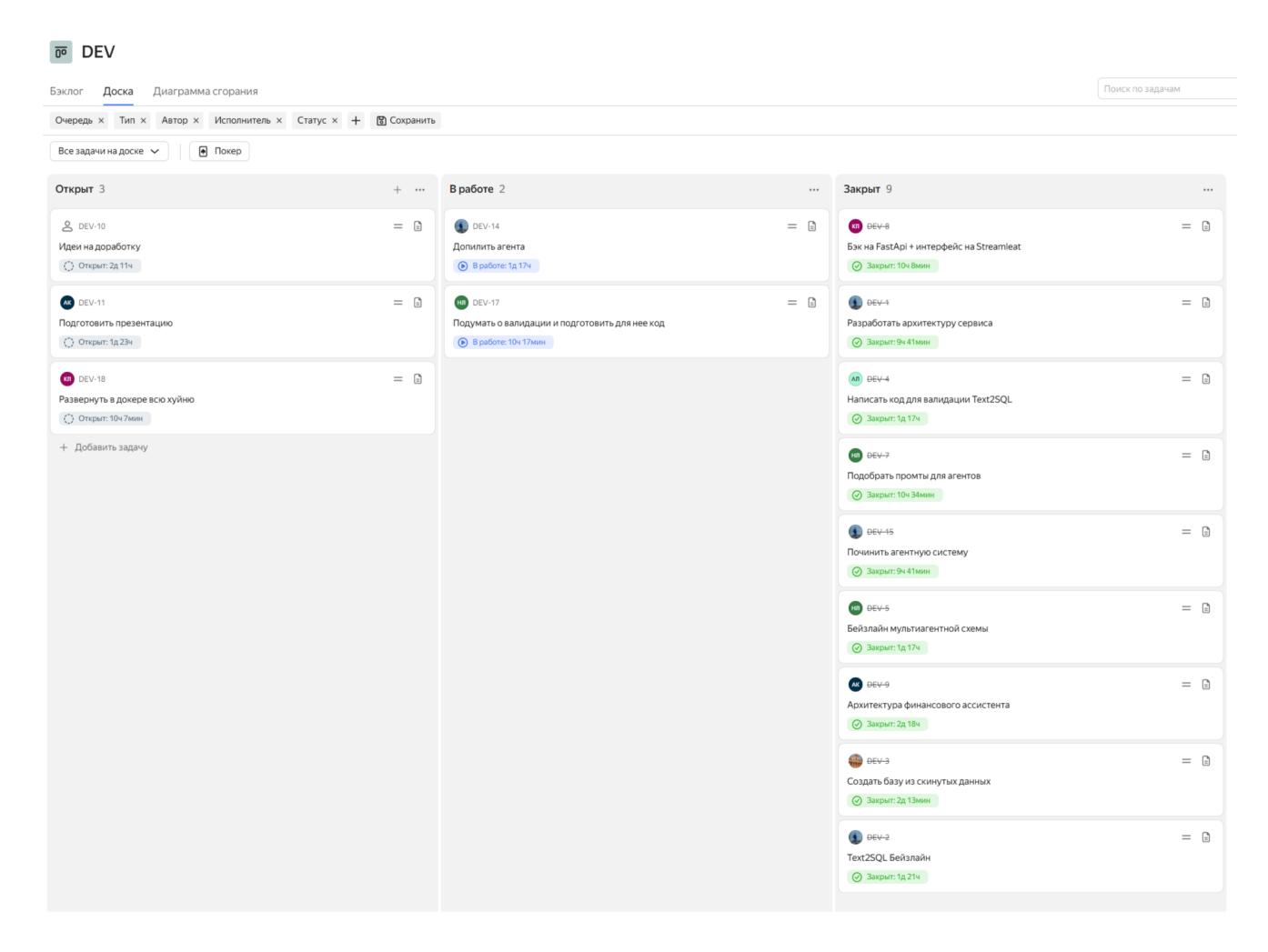


Задачи, поставленные перед командой

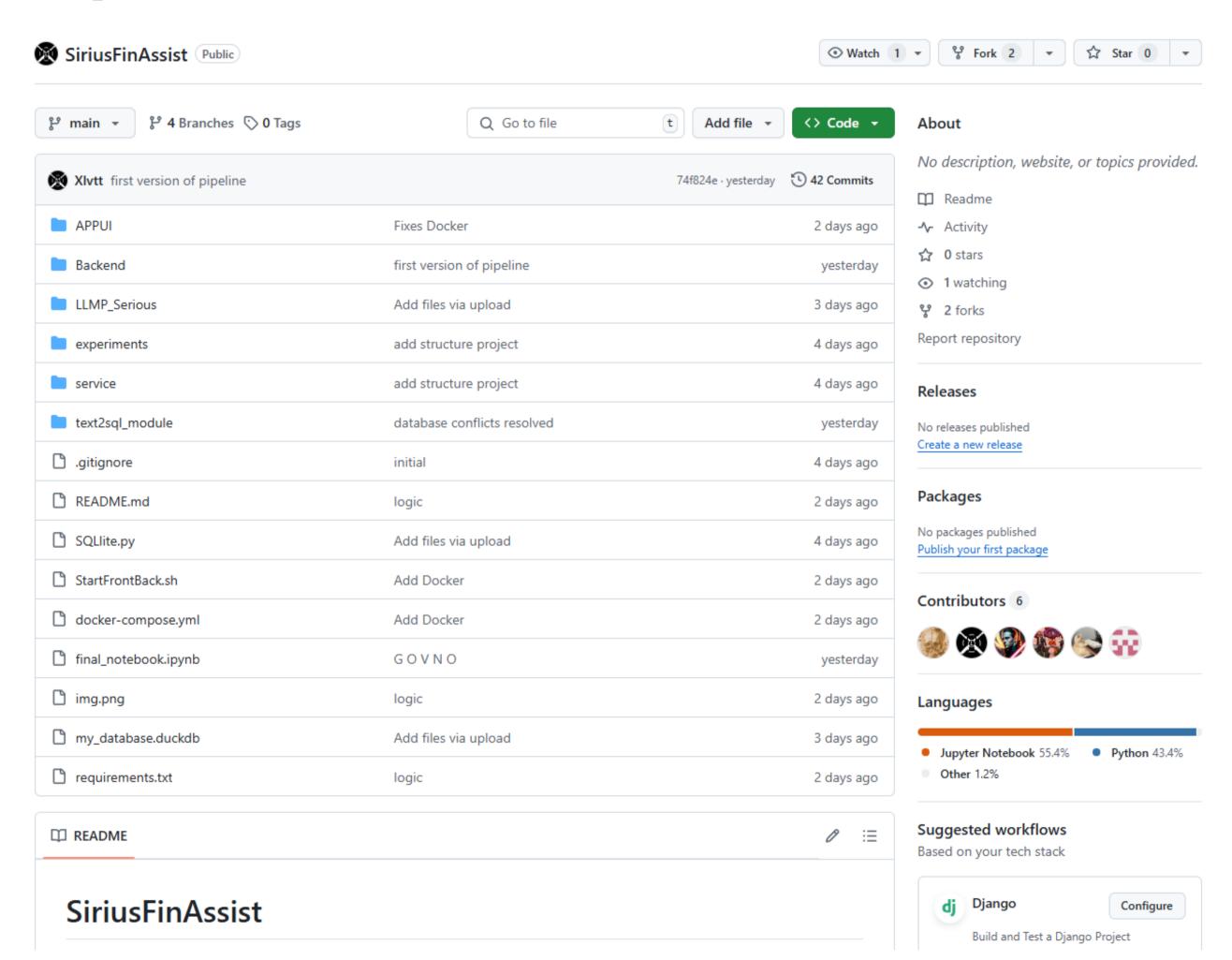
- Собрать пайплайн, который сможет отвечать на финансовые вопросы
- Основная сложность: промптингом с text2sql моделью трудно ответить на вопросы корректно
- Есть необходимость строить LAYER с Retrival прежде чем подавать в LLM (RAG) таблицы
- Есть вариант решения следующими способами: промптингом, RAG, СФТ
- Придумать метрику оценки качества решения

Организация работы





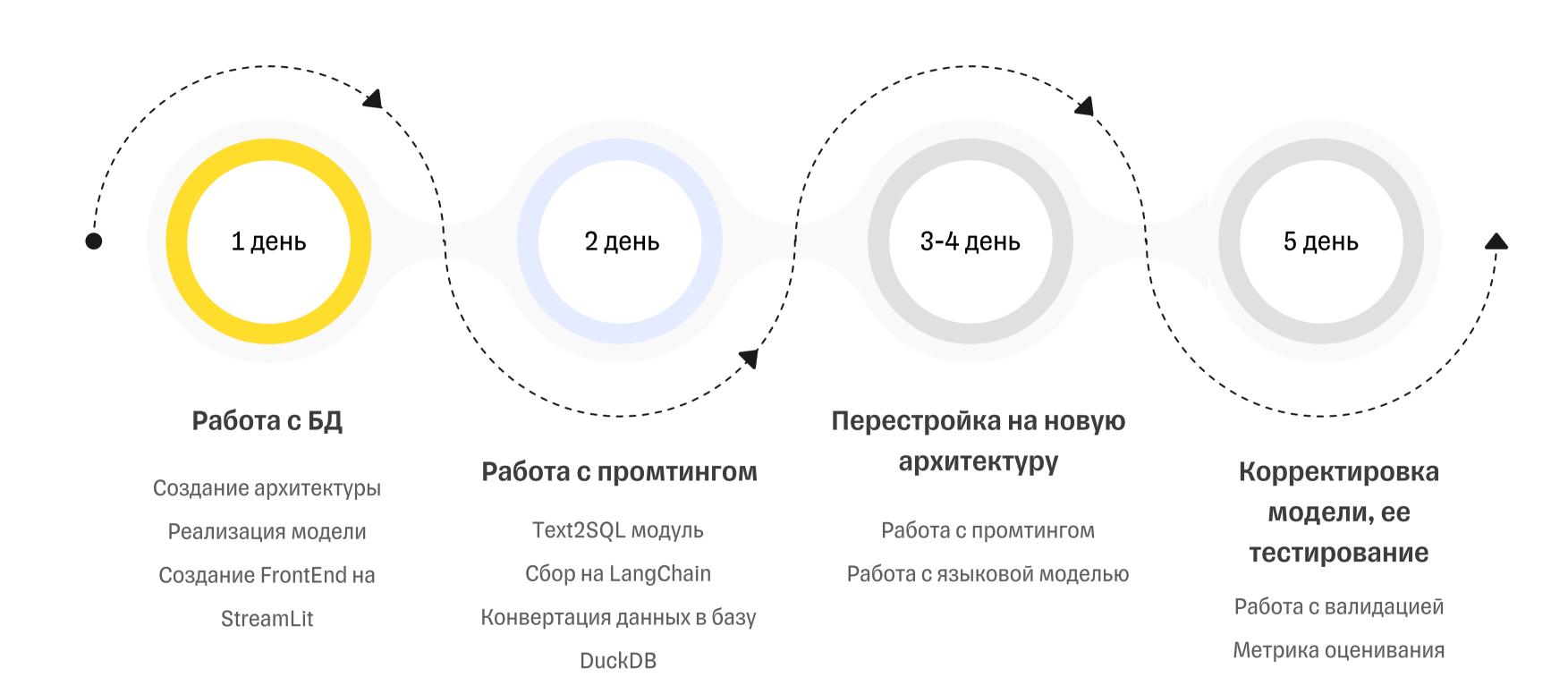
Организация работы





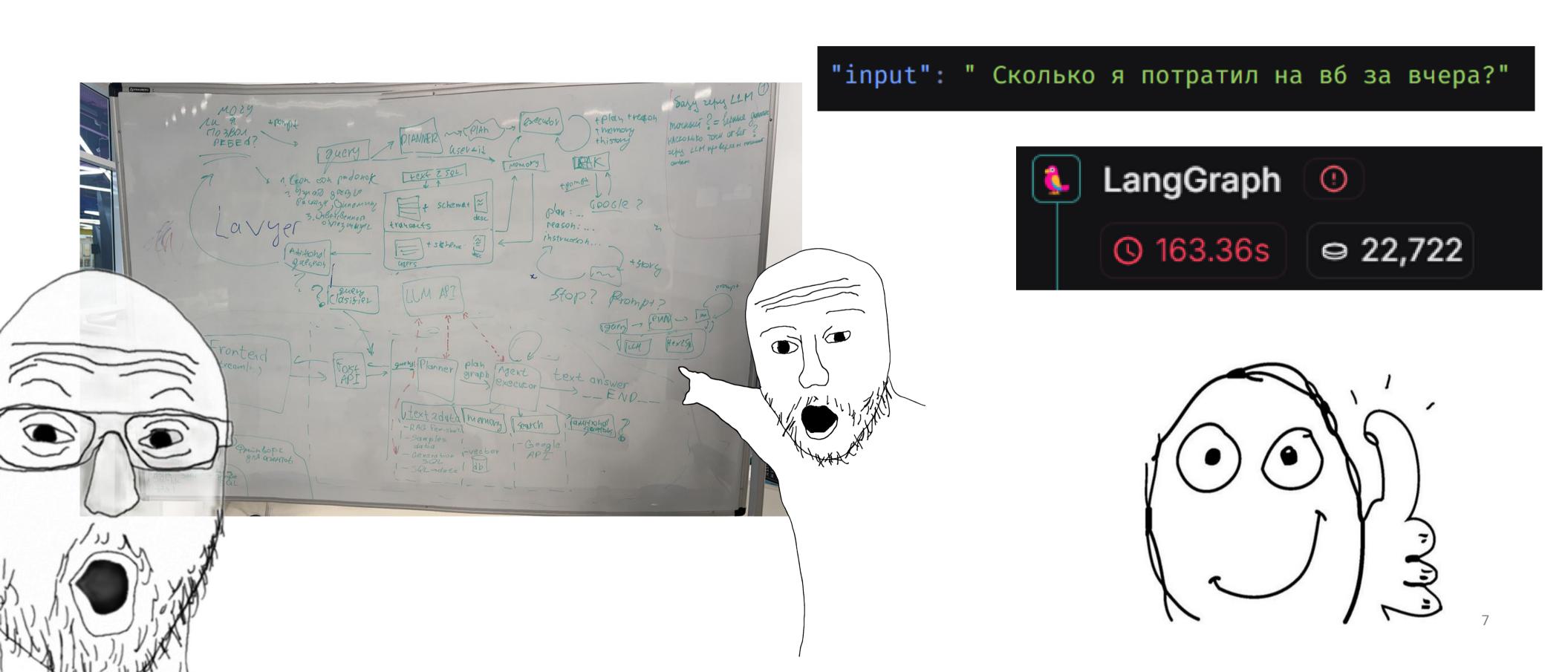


Этапы проекта

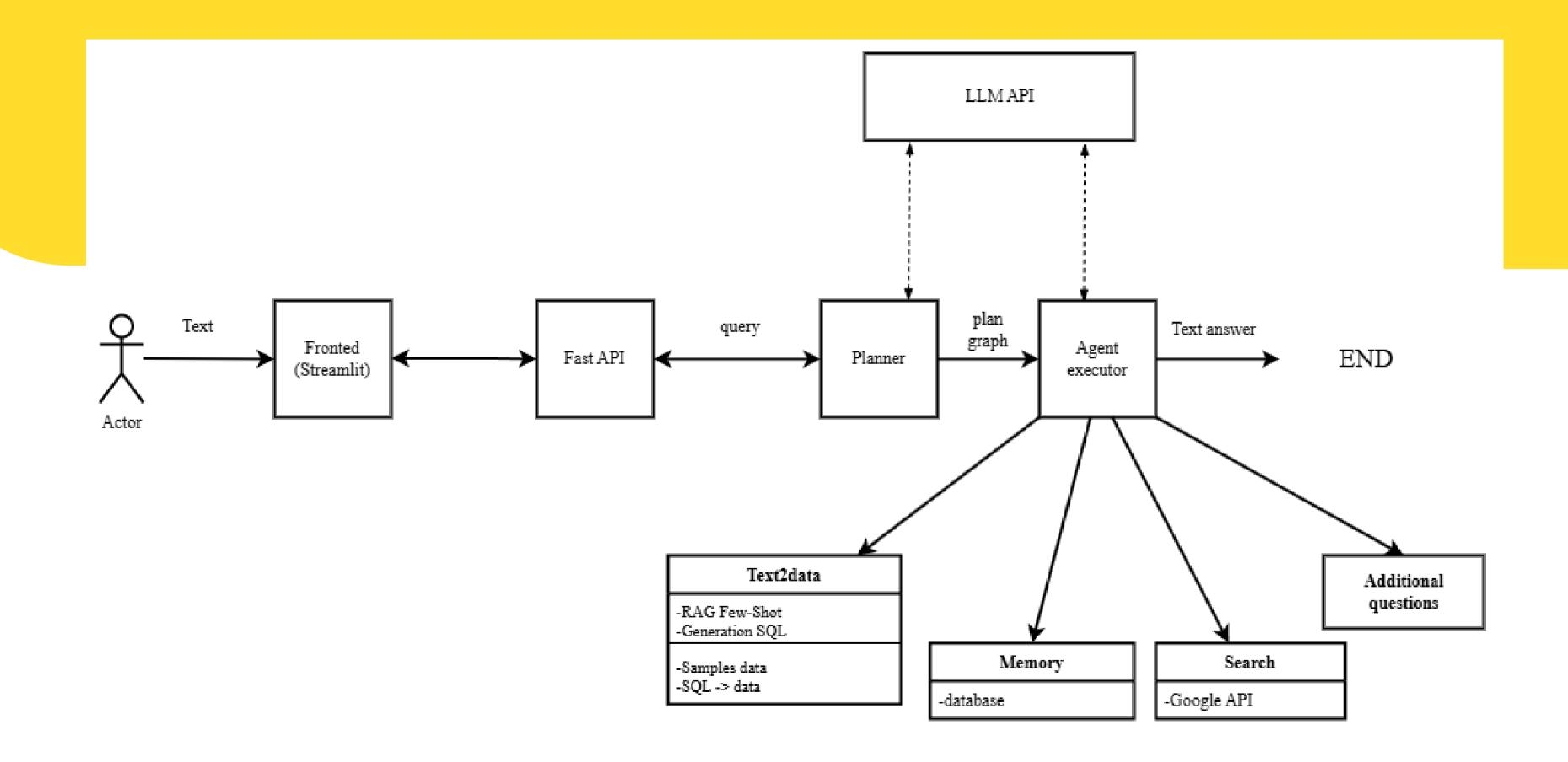


Мультиагентный аппарат

Как мы пробовали разные архитектуры?



Архитектура



Сколько же я потратил в маке?

```
| Total content of the state o
```



```
Rendered Output ~ 2.

Al

I found information about the meaning of 'мак':

In Russian, 'мак' (mak) translates to 'poppy'. It can refer to the poppy plant or its seeds.
```

Сколько же я потратил в маке?

```
1 v {
2    "input": "{'query': 'total_spent'}"
3  }
```

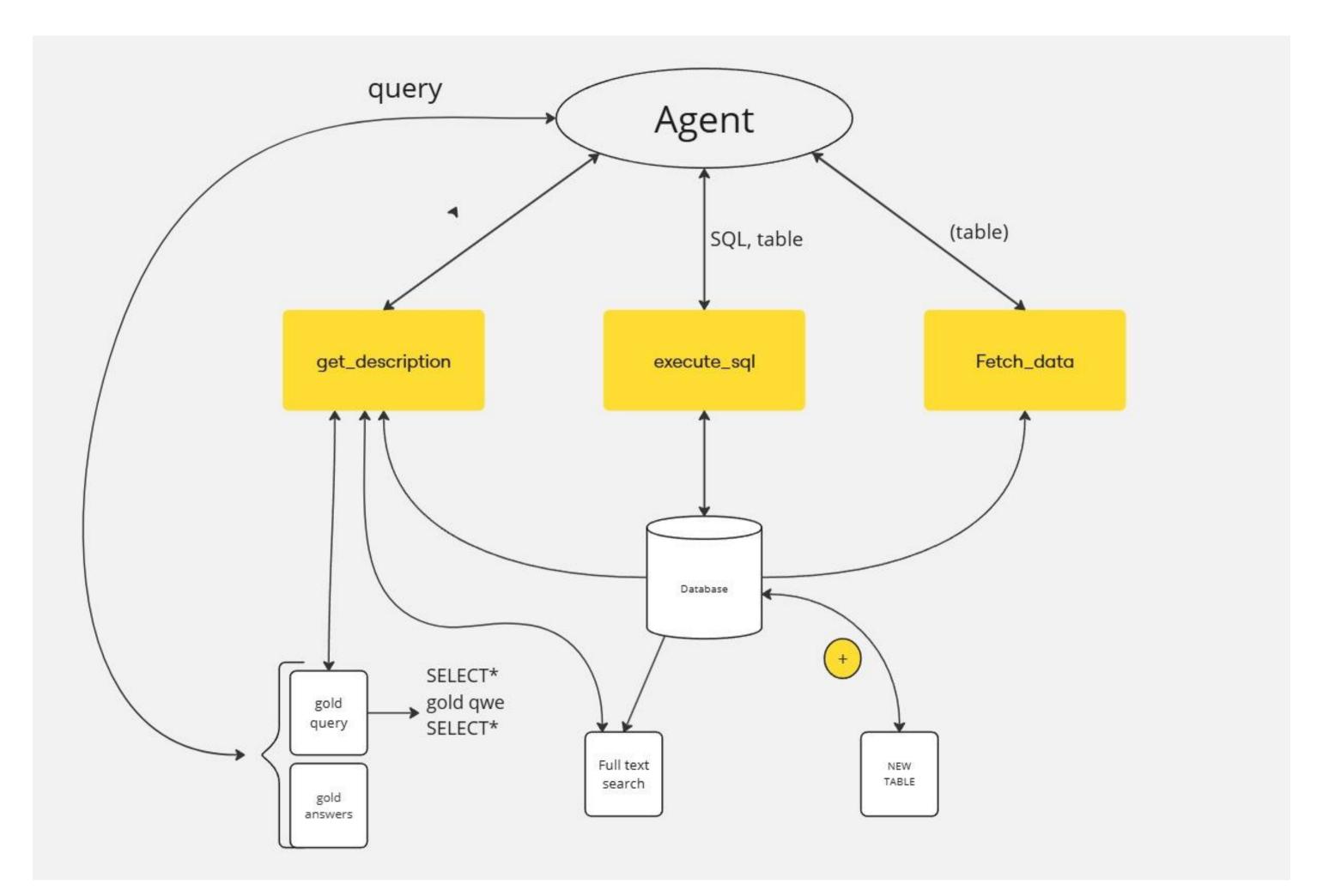


The total amount spent on 'poppy seeds' for all time is \$53,517,613.69.

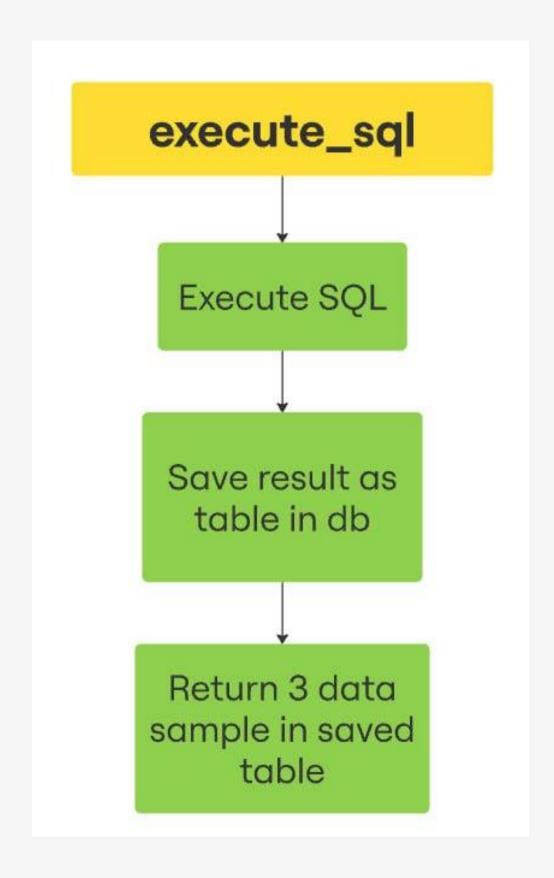


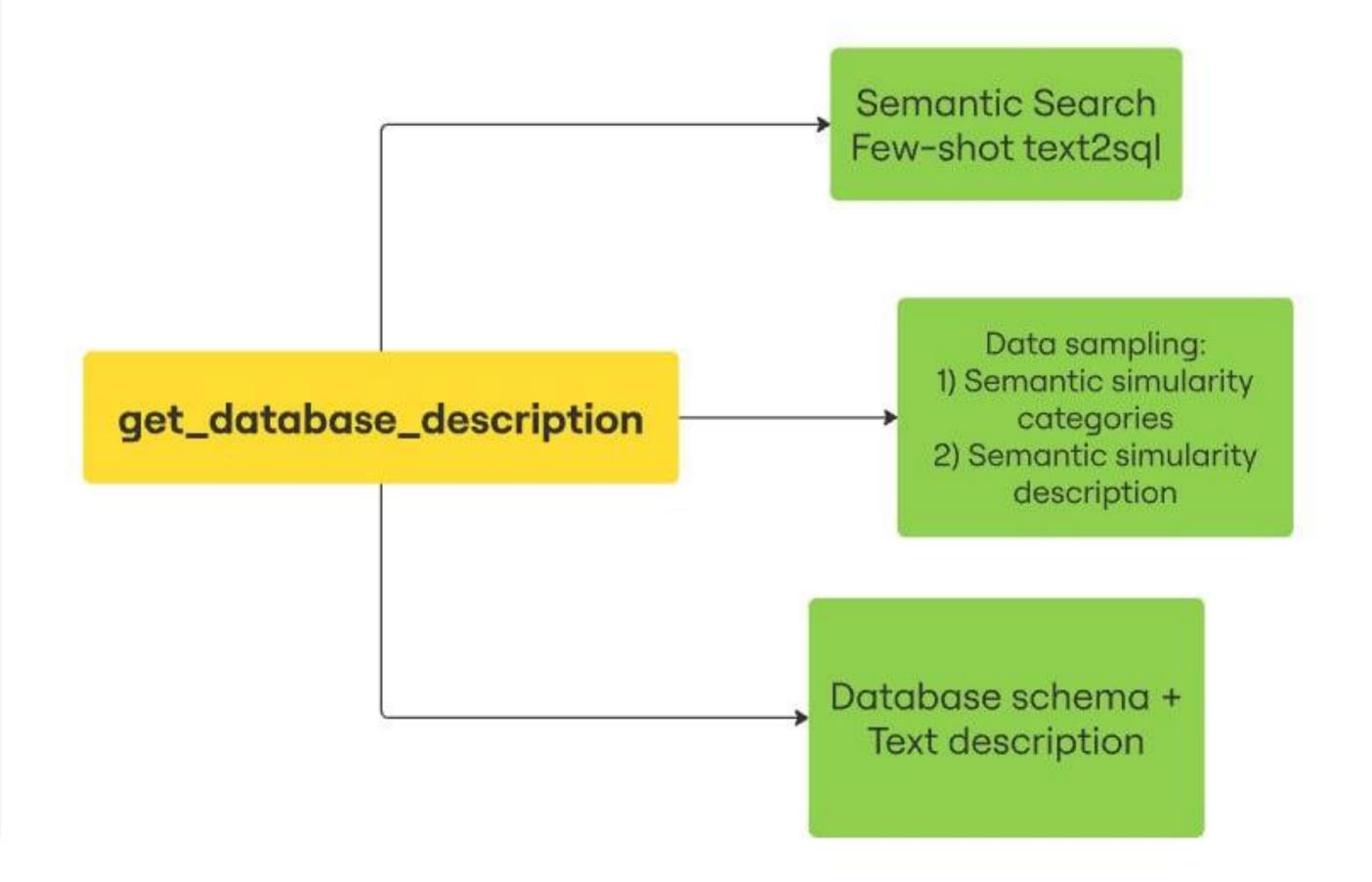


Улучшенная архитектура



Text2SQL pipeline





Техническое решение

Python

Для работы с данными, веб-фреймворками и интеграцией.





FastAPI

Backend-фреймворк для создания RESTful API.

Быстрый, современный и асинхронный, поддерживает OpenAPI для документации.

Легко масштабируется и интегрируется с другими технологиями.

Docker

Платформа для контейнеризации приложений.

Обеспечивает независимость от окружения и легкость развертывания.

Упрощает управление зависимостями и версиями.

Streamlit



Фреймворк для создания интерактивных веб-приложений.

Простота использования: интерфейс создается через Python-код.

Идеально подходит для прототипов и визуализации данных.

docker

DuckDB

Легковесная аналитическая база данных, не требующая сервера.

Поддерживает SQL-запросы, схожа с PostgreSQL по функциональности.

Высокая производительность для обработки больших объемов данных в памяти

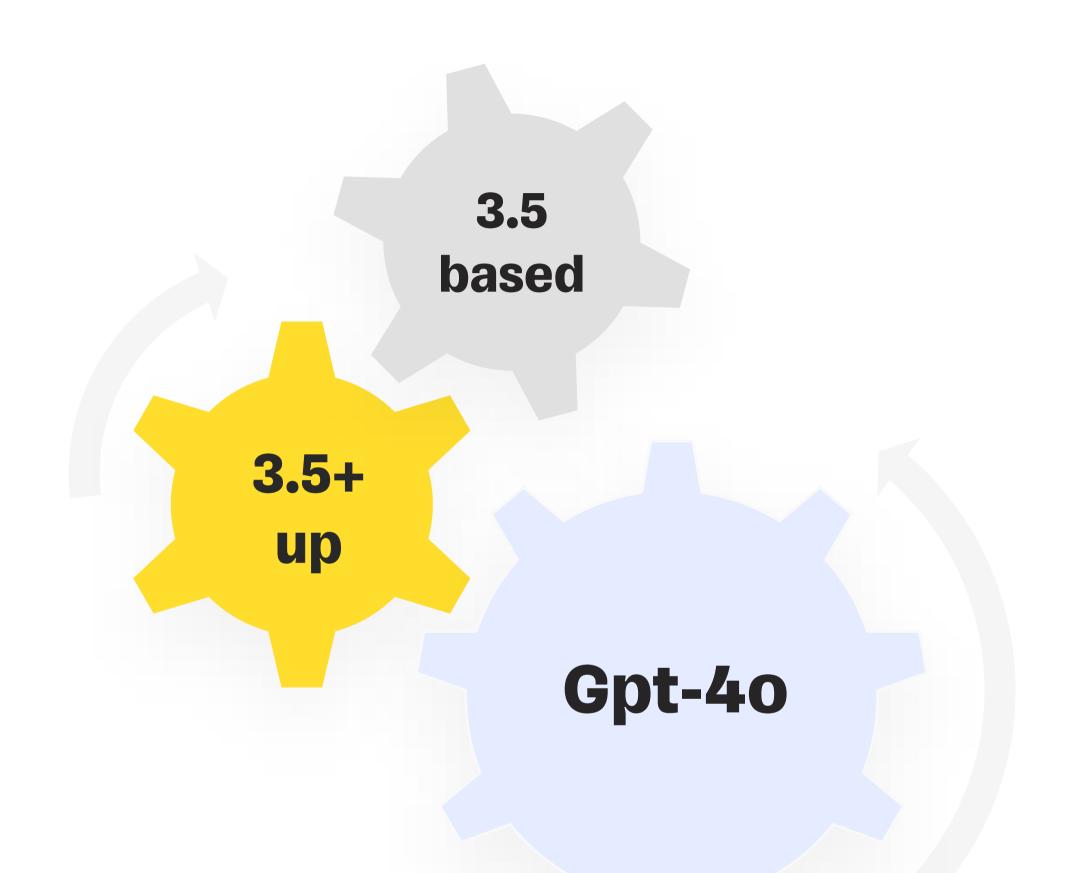


LLM модель. Обоснование выбора

- Для разработки агента необходима сильная языковая модель.
- Была попытка завести агента на Qwen2.5 72B плохо справляется с работой как агент.
- В качестве основной модели была выбрана gpt-3.5-turbo. Проводились отдельные тесты на gpt-4o.

Точность выдачи ответов

Ежедневная работа над качеством агента



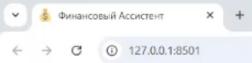
LLM промптинг

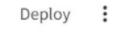
```
word_info = f'''Дополнительная инфрмация: текущая дата {get_current_date()}'''
question answer examples = '''Πρимер:
Вопрос: Сколько я потратил на отели в 2024?
Ответ: На отели вами было потрачено <N> рублей.
Вопрос: Когда был последний перевод?
Ответ: Последний перевод был совершен на сумму <N> RUB и произошел 10 декабря 2024 года в 23:36.
Вопрос: Сколько я потратила в Пятёрочке?
Ответ: Ты потратила <N> RUB в Пятёрочке'''
additional instructions = '''Если речь идёт о расходах, то не используй минус при ответе.
Если из базы данных возвращается данных, то отвечай, что не было найдено подходящих данных.'''
llm config = {"config list": [{"model": model name, "api key": os.environ["OPENAI API KEY"]}]}
assistant = AssistantAgent("assistant", llm_config=llm_config,
                           system message='''Ты профессиональный финансовый ассистент.
Ты создан отвечать на вопросы пользователей о их финансах, быть полезным.
Действуй по следующему плану:
1. Посмотри информацию о базе данных
2. Сделай промежуточные таблицы
3. Сделай выбор нужных таблиц для ответа на вопрос и сагрегируй результат
{additional instructions}
{question answer examples}
Ответ давай подробно, указывая все необходимые данные.
Если в ответе есть число денег, то укажи валюту.
Если в ответе есть различные операции, то укажи при наличии информацию когда и в каком магазине они были совершены.
Если чего-то не знаешь, то не пиши этого.
{word_info}''')
```

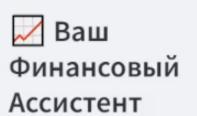
В чем сила?

- В размере модели промпте

- Примеры вопрос-ответ
- Специфичные инструкции
- Общий план действий
- Знания о мире







Добро пожаловать! Управляйте своими финансами эффективно.



Ваши чаты:

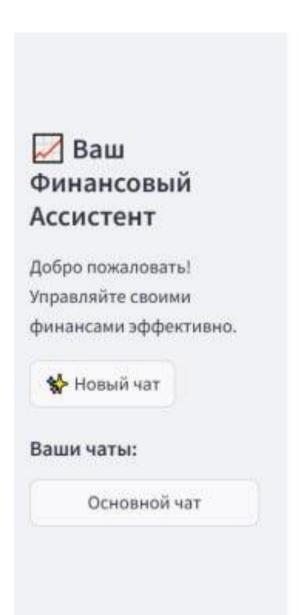
Основной чат



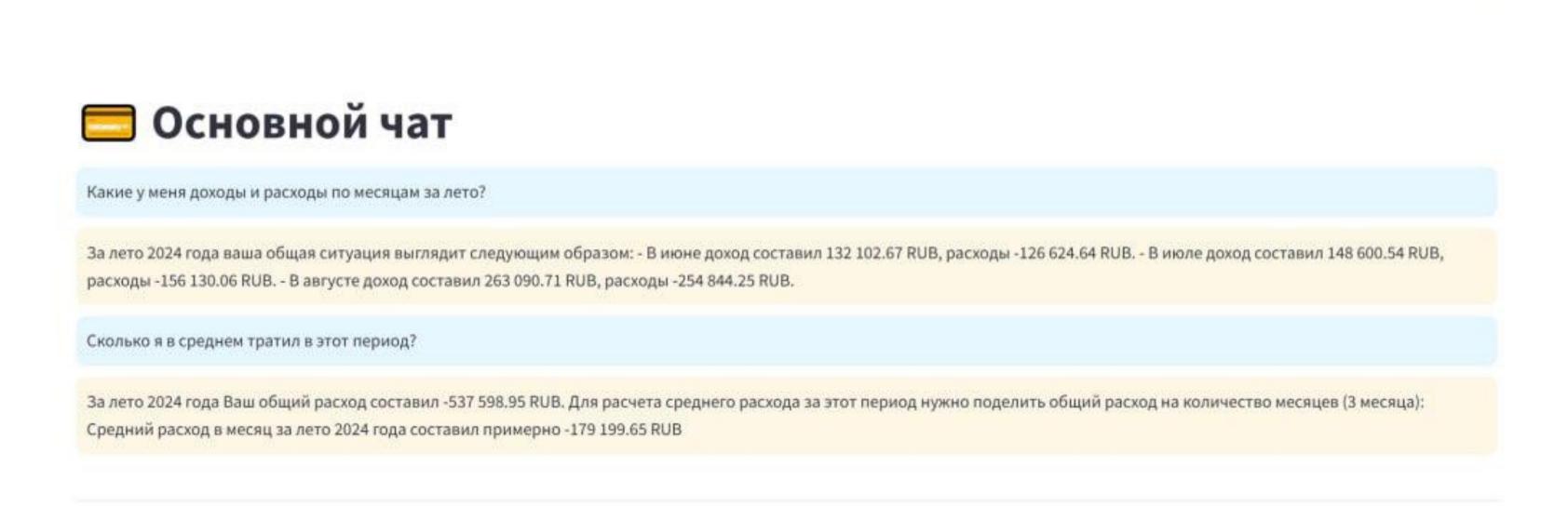
Введите ваше сообщение:

Отправить

Модель gpt-3.5



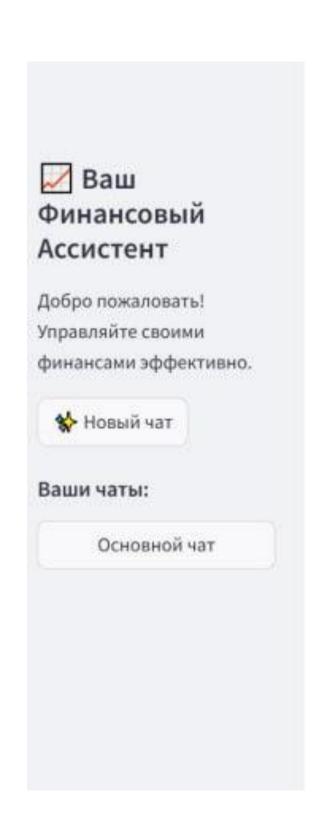
Введите ваше сообщение:



Отправить

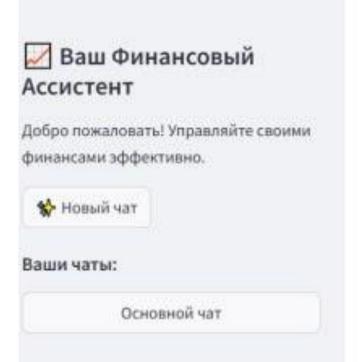
Deploy

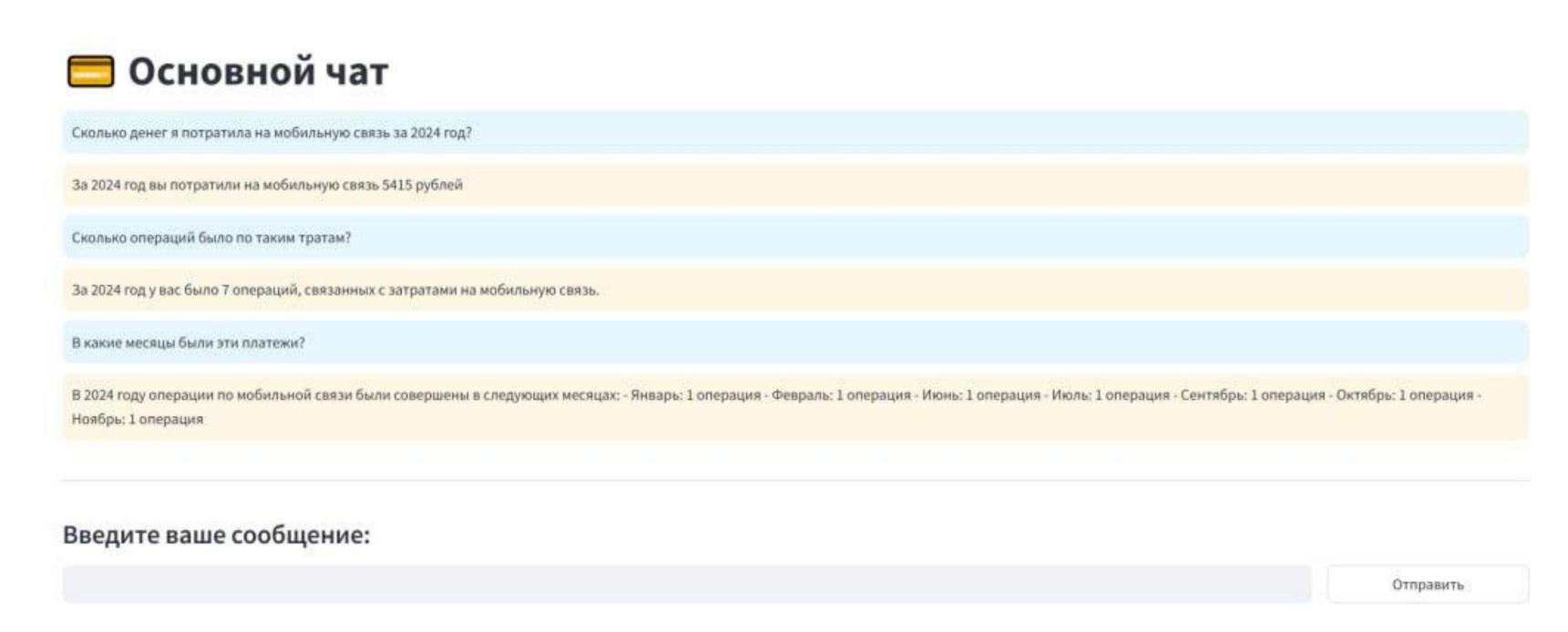
Модель gpt-3.5



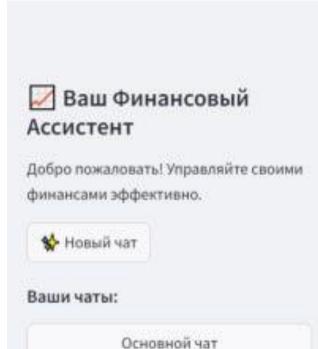


Модель 40





Модель 40





Как мне оптимизировать бюджет чтобы откладывать 150 тысяч рублей?

Для оптимизации бюджета с целью откладывания 150 тысяч рублей в месяц, необходимо рассмотреть ваши общие доходы и расходы:

- 1. Общие доходы (Credit):
 - Bcero: 1,928,023.63 RUB
- 2. Общие расходы (Debit):
 - o Bcero: -1,962,221.78 RUB

Исходя из этих данных, ваши текущие расходы (1,962,221.78 RUB) превышают доходы (1,928,023.63 RUB), что создаёт дефицит бюджета. Чтобы иметь возможность откладывать дополнительные 150,000 рублей, необходимо произвести следующие шаги:

- Уменьшить расходы: Изучите ваши текущие расходы более детально и попытайтесь выявить категории, где можно сократить издержки.
- Увеличить доходы: Рассмотрите возможности для увеличения вашего дохода, такие как поиск дополнительной работы или инвестирование.
- Оптимизация затрат: Сконцентрируйтесь на большом количестве мелких затрат, которые можно оптимизировать или убрать.

С учётом того, что ваш бюджет уже в дефиците, вам потребуется существенно оптимизировать расходы или увеличить доходы для достижения цели в 150,000 рублей в месяц

Введите ваше сообщение:

Отправить

Deploy :

Валидация решения. Метрика

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
rouge1_fmeasure	0.275	0.311	0.263
rouge1_precision	0.397	0.532	0.425
muge1_recall	0.253	0.26	0.217

ROUGE-1

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
rouge2_fmeasure	0.16	0.124	0.153
rouge2_precision	0.197	0.197	0.268
rouge2_recall	0.146	0.102	0.136

ROUGE-2

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
rougeL_fmeasure	0.262	0.299	0.263
rougeL_precision	0.378	0.514	0.425
rougeL_recall	0.243	0.25	0.217

Валидация решения. Метрика

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
bert_f1	0.727	0.785	0.753
bert_precision	0.679	0.751	0.717
bert_recall	0.786	0.824	0.796
bleu	0.071	0.1	0.069

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
judge_accuracy	1.538	1.423	3.308
judge_calibration	1.846	0.846	2.462
judge_recall	1.692	1.769	3.538
judge_relevance	3.154	2.692	4.231
judge_usefulness	2.385	2.154	3.654

BERT + BLEU

LLM Judge

metric	baseline3.5_metrics	upgraded3.5_metrics	upgraded4-o_metrics
ner_fl	0.102	0.075	0.144
ner_precision	0.098	0.075	0.135
ner_recall	0.114	0.079	0.162

NER 23

Наша команда



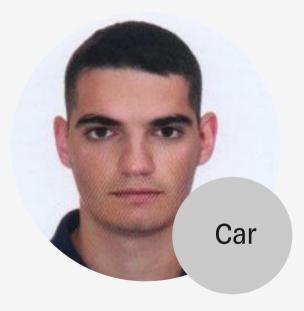
Никита Ляпин Генеральный директор



Камиль Садриев Директор по Docker



Владимир Зворыгин Технический директор



Нессер Ахмед Генеральный исполнитель



Анна Корецкая Креативный директор



Литвин Антон Специалист БД