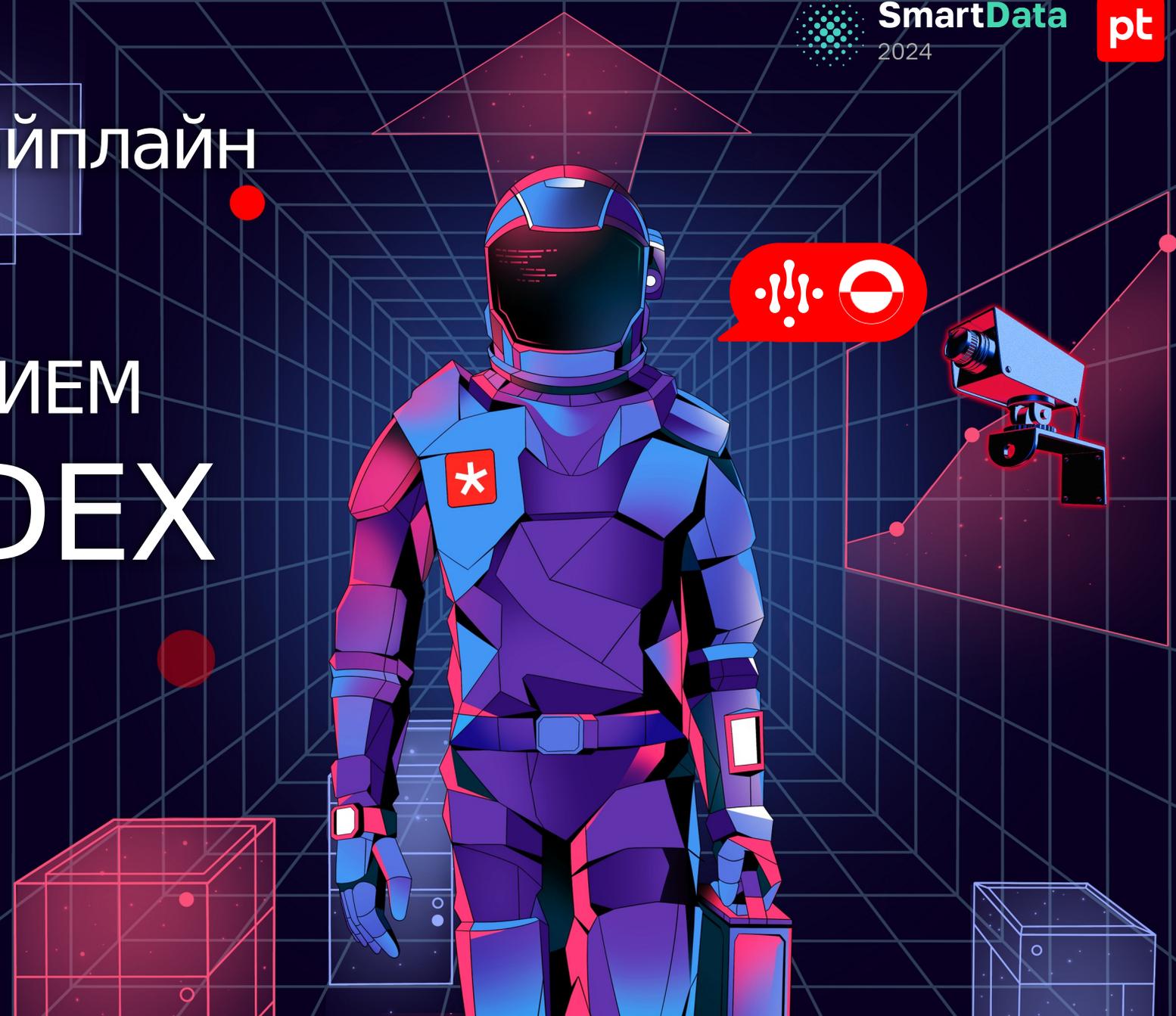


# Как построить пайплайн RAG С ИСПОЛЬЗОВАНИЕМ LLAMAINDEX

Алина Кочева  
Алсу Нурутдинова



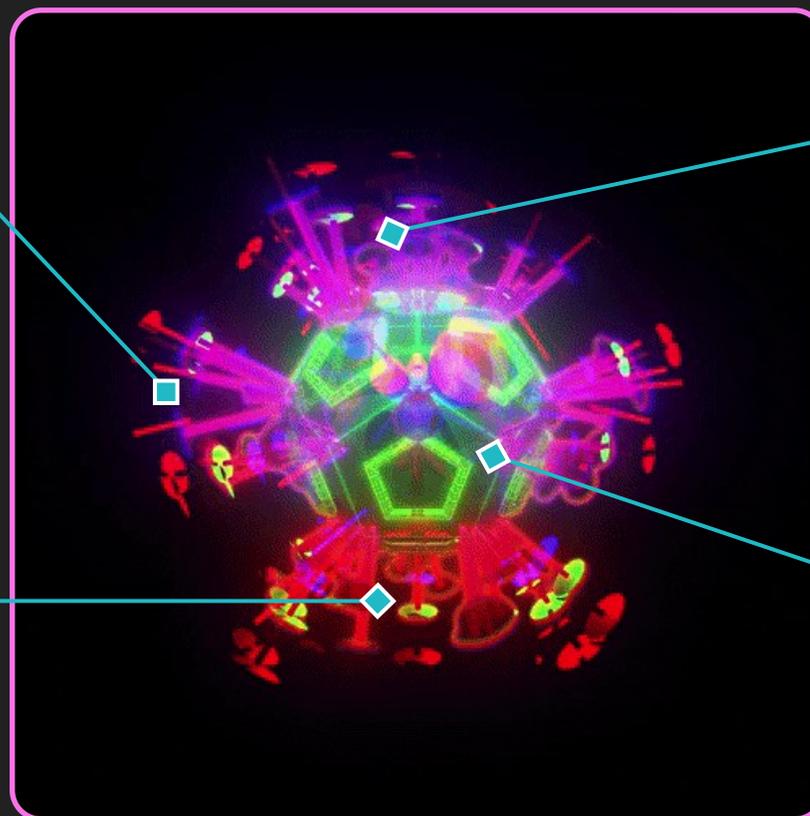
# Agenda

## 1 О задаче

Чат-бот для  
техподдержки

## 3 Pipeline

Схема пайплайна для  
задачи



## 2 Про RAG

Как работает. Как  
создать и настроить.

## 4 Про результаты

Оценка,  
оптимизация,  
рекомендации,  
выводы



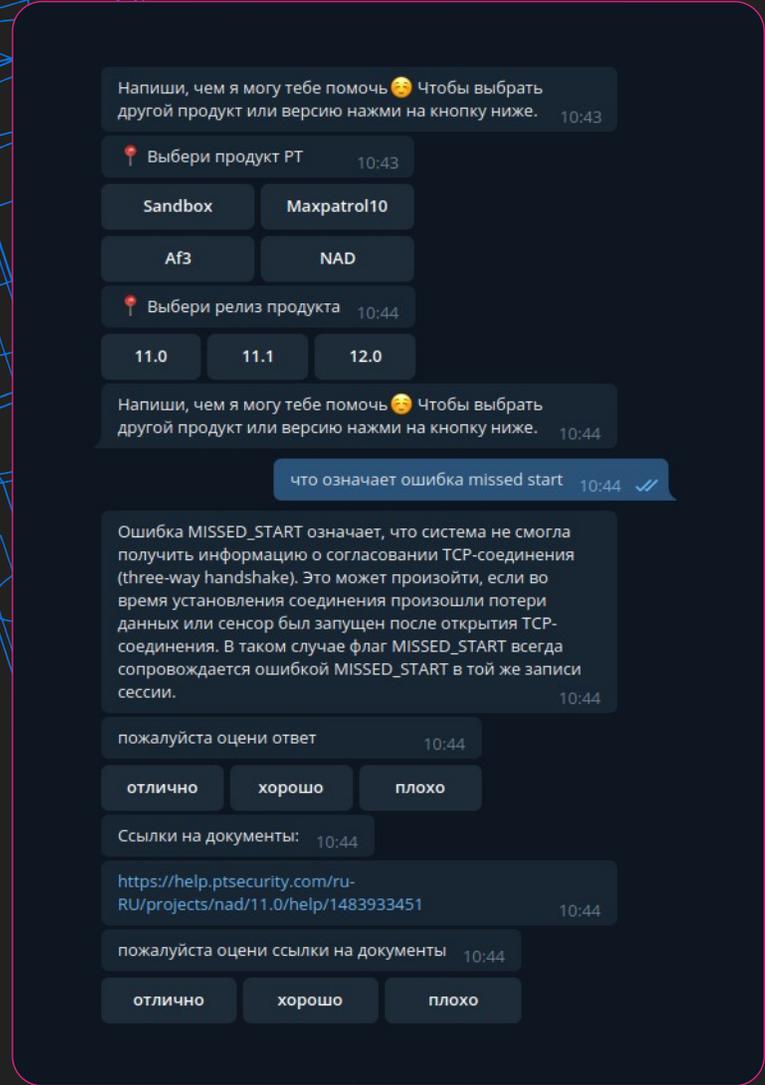
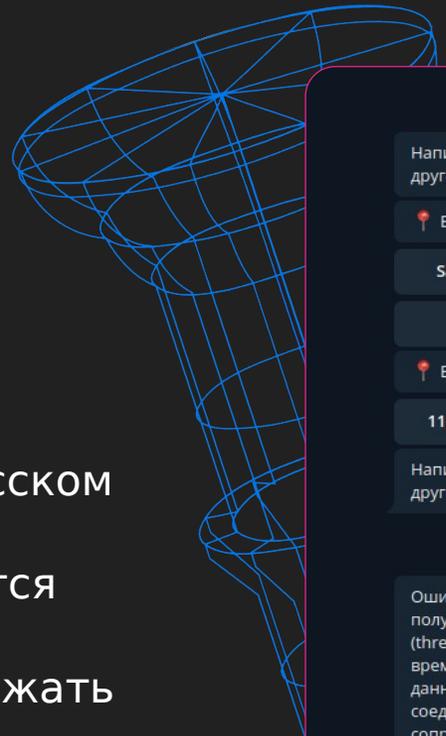
# Task

## Требования

- / Есть публичный портал-справка по продуктам компании.
- / Бот должен отвечать четко по документации компании.
- / Замена первой линии техподдержки.

## Ограничения

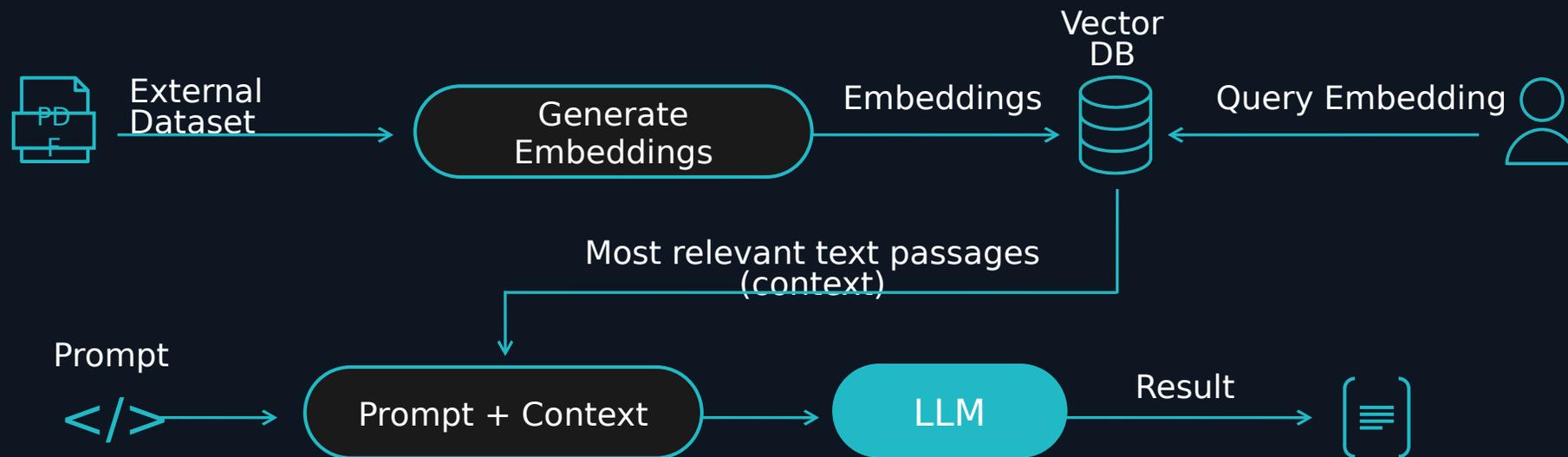
- / Документация на русском
- / Постоянно обновляется
- / Запросы могут содержать конфиденциальную информацию
- / Не можем использовать открытые языковые модели



# What is RAG ?

Retrieval Augmented Generation

## RAG Structure



# Why RAG

Обновление информации в режиме реального времени, не требует затрат для переобучения

Принимает многие форматы данных, есть LangChan, llamaindex, нужна настройка

Большие базы знаний, чат-боты

Сводит к минимуму риск утечки конфиденциальных данных и "галлюцинаций"



Требует дообучения,

Дорого, нужны данные и ресурсы

требует обработки, Дорого, нужны ресурсы, подбора модели данных и ресурсы

Для специализированных задач

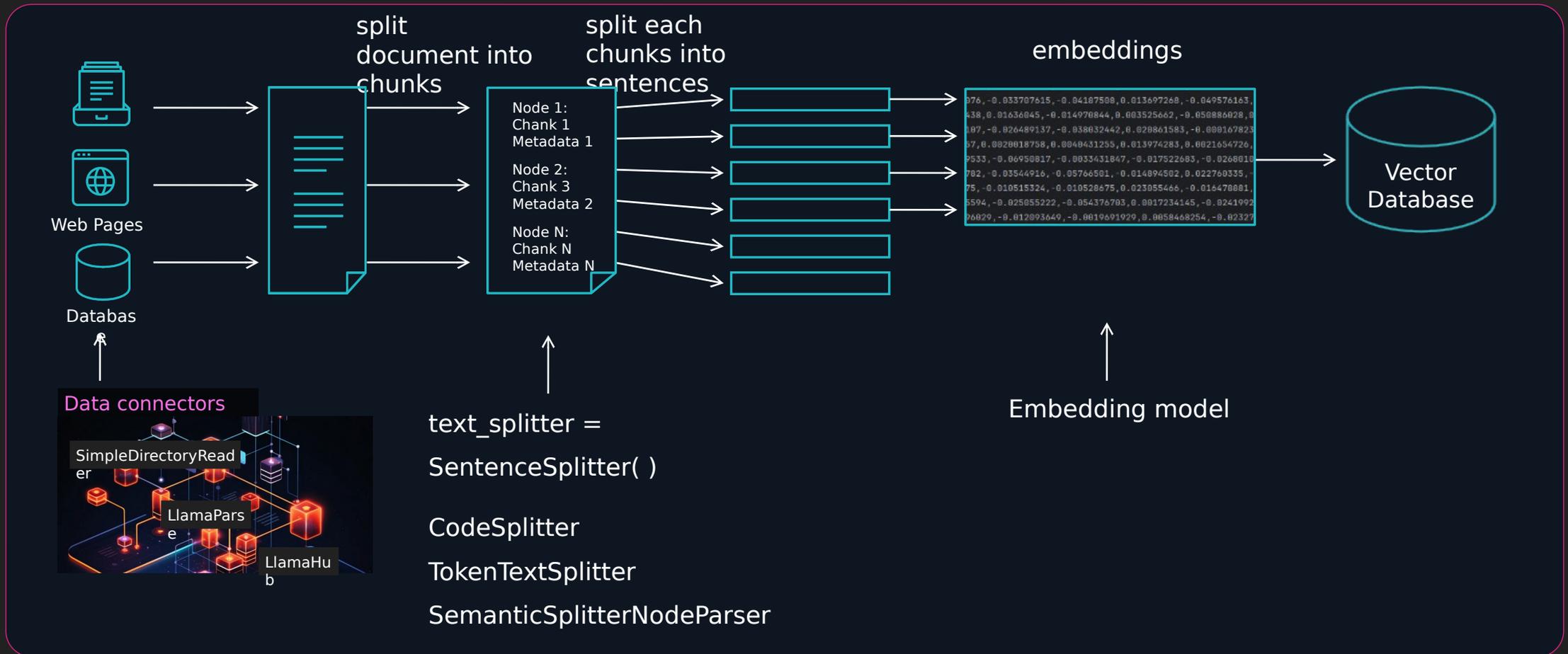
# LLAMAINDEX vs LANGCHAIN

- Простота
- Интересные инструменты
- Для RAG



- Многоцелевой фреймворк
- Для широкого комплекса задач

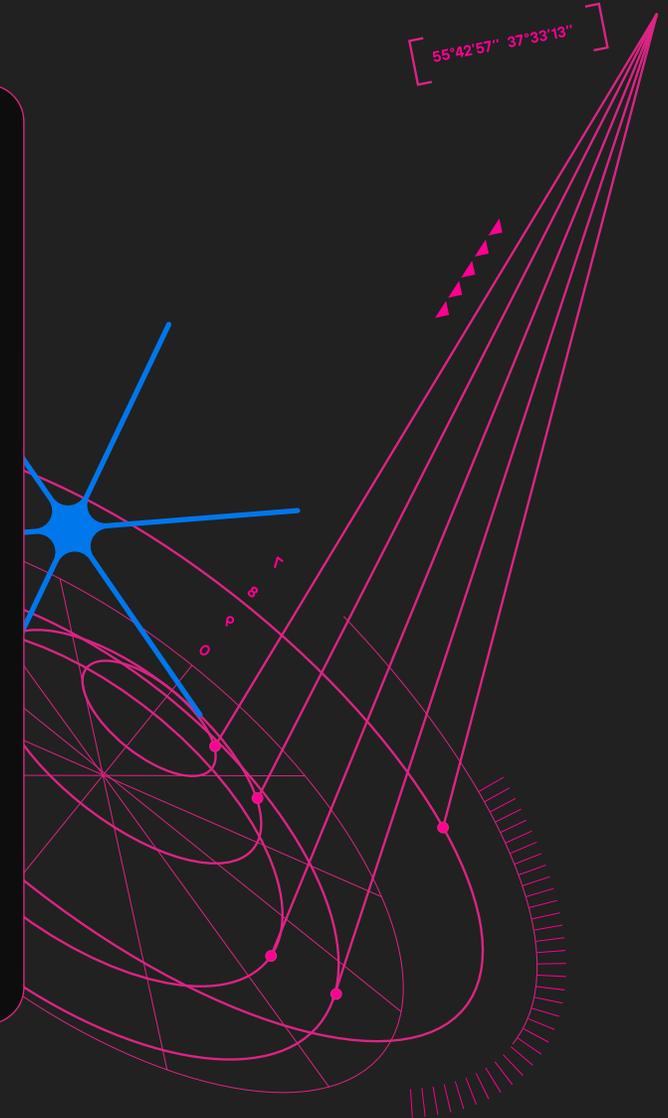
# Indexing Stage



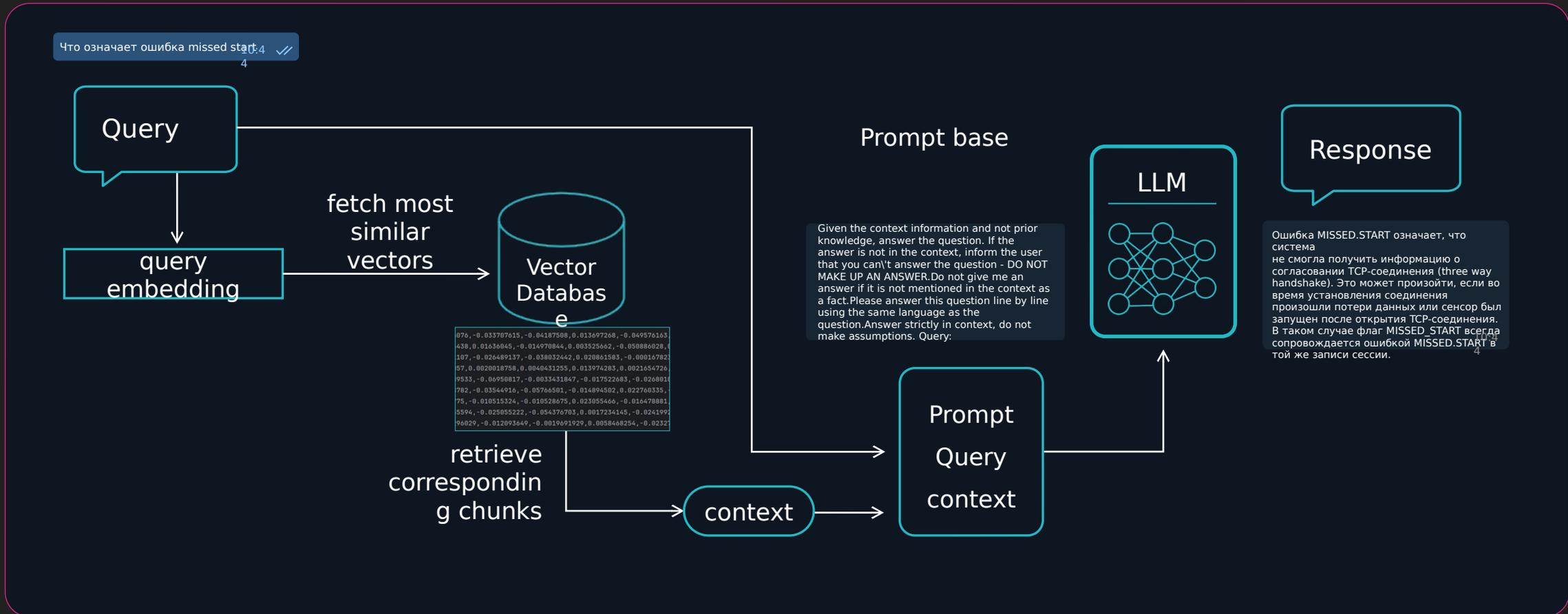
# Создание индекса

```
def get_index(index_name: str, embedding: object) -> object:
    """
    Функция для создания индекса.
    :param index_name: str - имя индекса.
    :param embedding: object - модель внедрения.
    :return: object - контекст хранилища.
    """
    service_context = ServiceContext.from_defaults(
        llm=llm_model,
        embed_model=embedding,
        chunk_size=cchunk_size,
    )

    vector_store = WeaviateVectorStore(weaviate_client=client, index_name=str(index_name))
    storage_context = VectorStoreIndex.from_vector_store(
        vector_store, service_context=service_context
    )
    return storage_context
```



# Retrievers. Query Engines.





# Пример использования индекса

```
def process_prompt(prompt: str, index_name: str, embedding: object) -> object:
    """
    Функция для обработки запросов.
    :param prompt: str - текст запроса.
    :param index_name: str - имя индекса.
    :param embedding: object - модель внедрения.
    :return: object - ответ на запрос.
    """

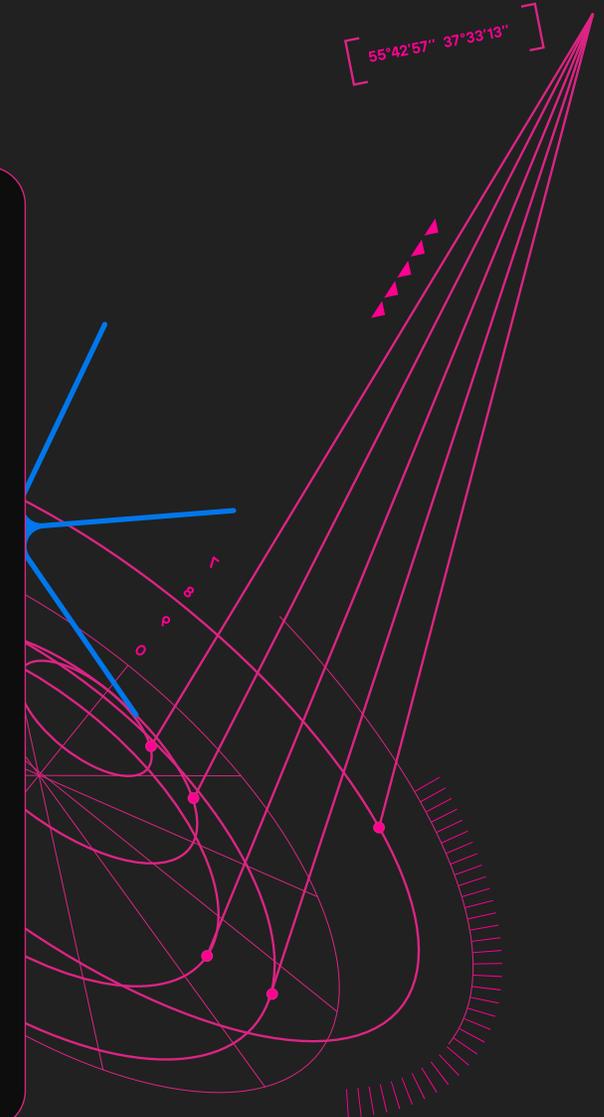
    text_qa_template = PromptTemplate(prompt_base)

    index = get_index(index_name, embedding)

    query_engine = index.as_query_engine(text_qa_template=text_qa_template,
                                        similarity_top_k=5,
                                        )

    response = query_engine.query(prompt)
    return response

# Пример использования
response = process_prompt(prompt, index_name, embedding)
```



# RAG pipeline

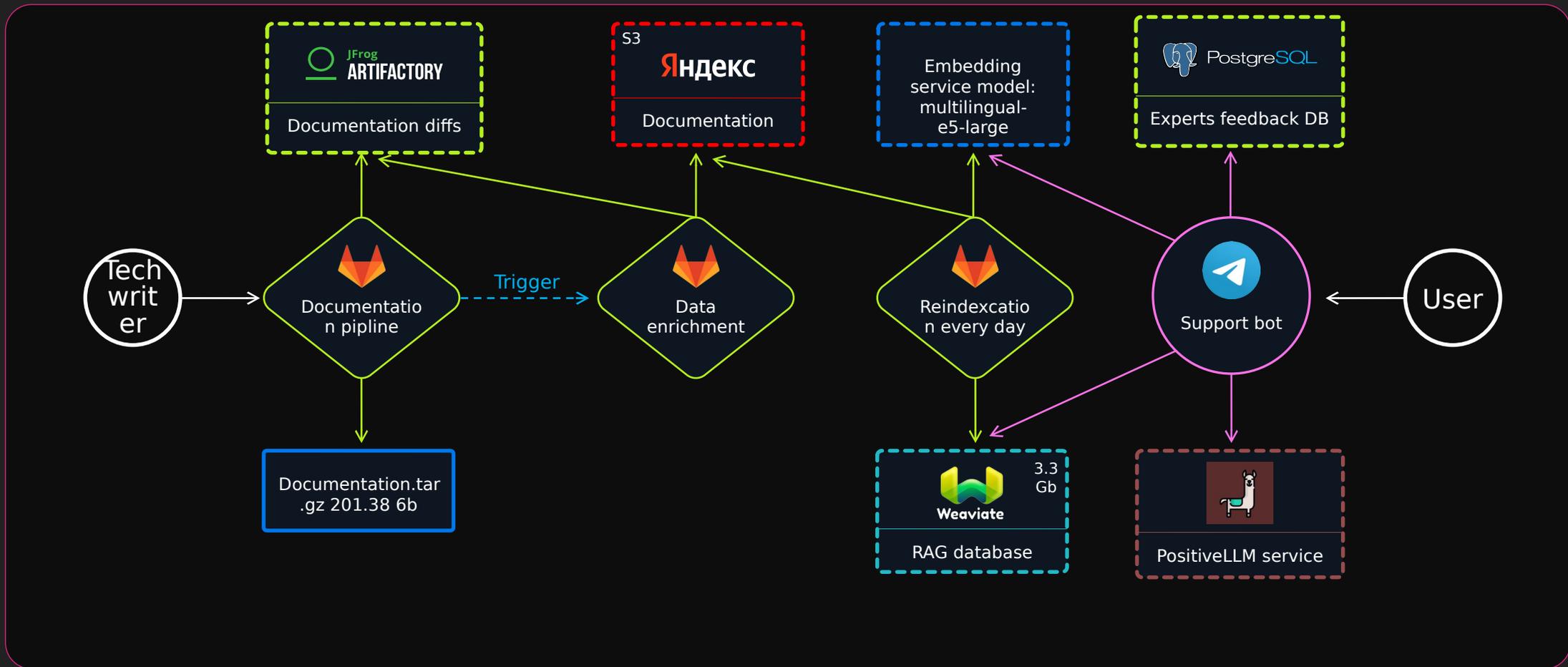
Решили сделать  
небольшую  
автоматизацию  
на базе GitLab Pipelines.



## Почему GitLab:

- / поддержка GitLab на корпоративном уровне в РТ
- / наличие Scheduled Pipelines, позволяющая запускать сборки по расписанию из кода в репозитории, сохранять результаты каждого запуска
- / возможность уведомлять нас в случае падения пайплайна

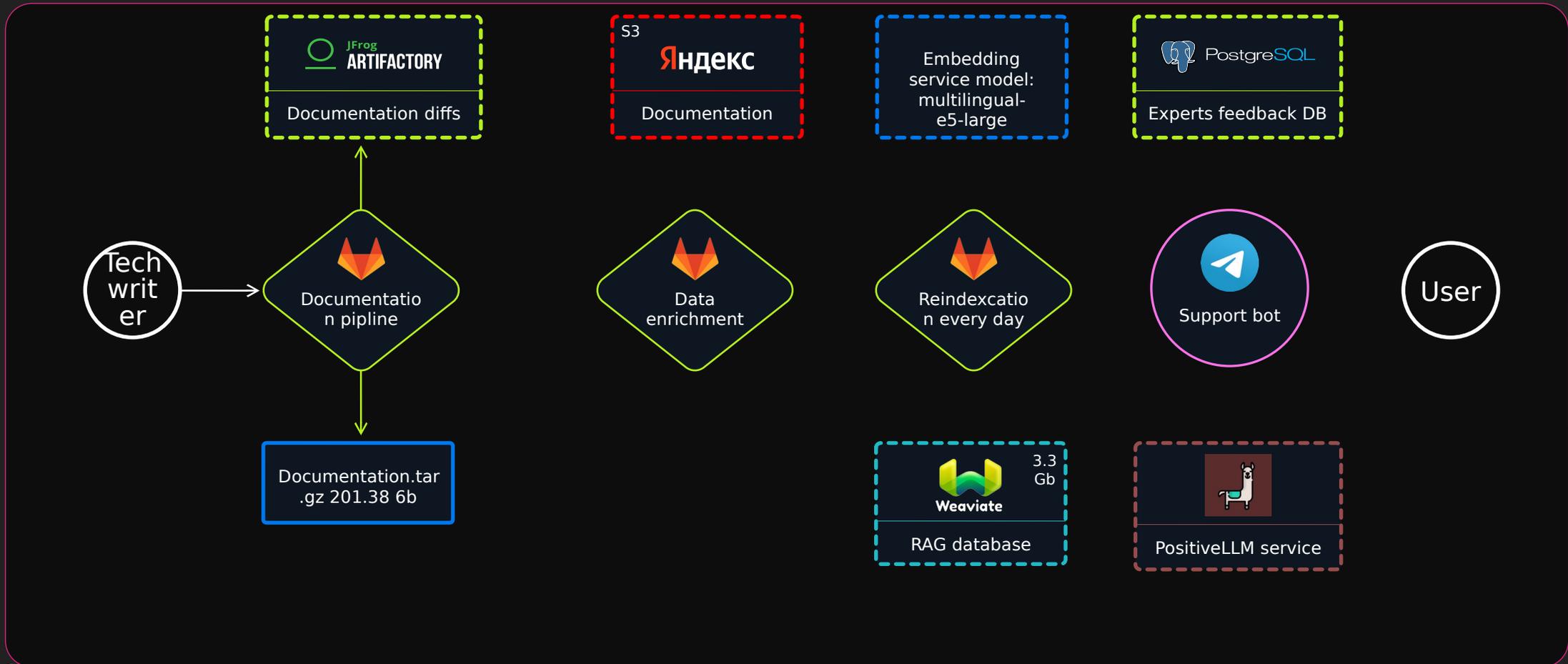
# RAG pipeline



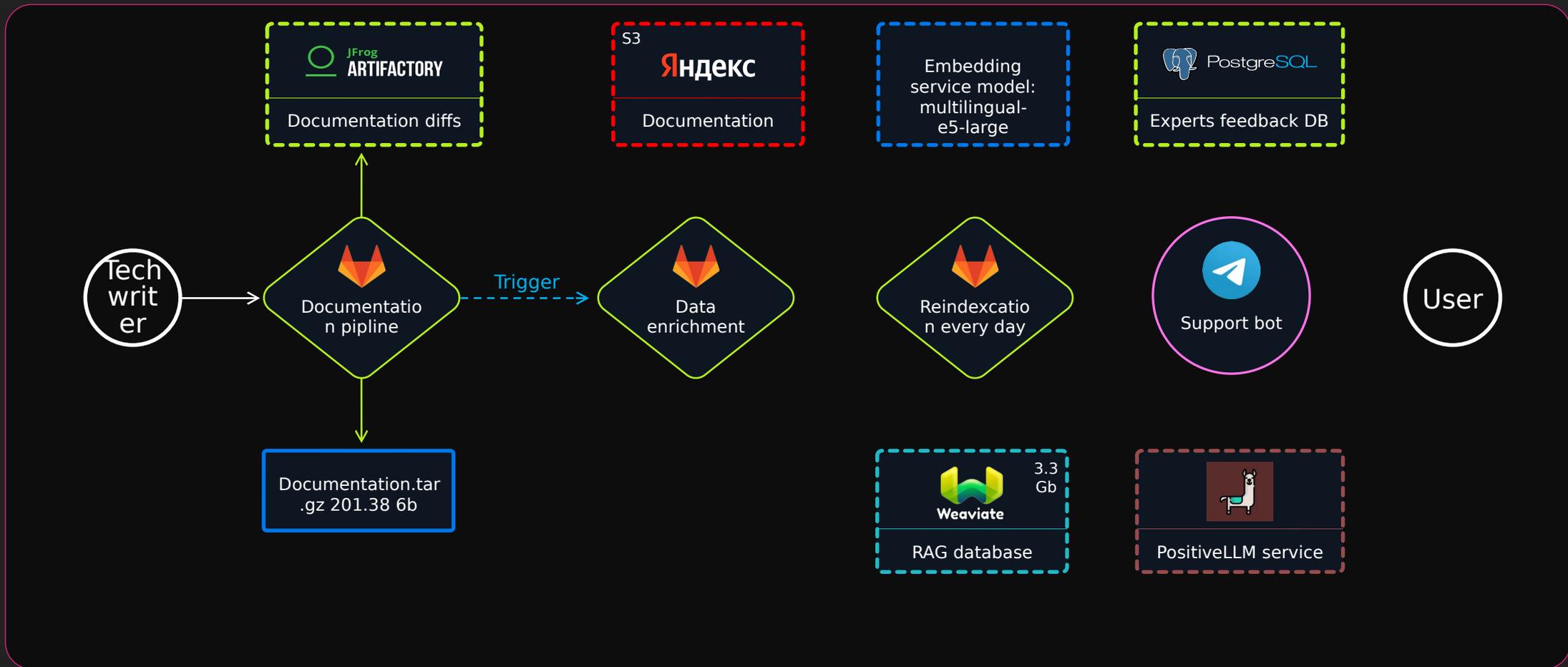
# RAG pipeline



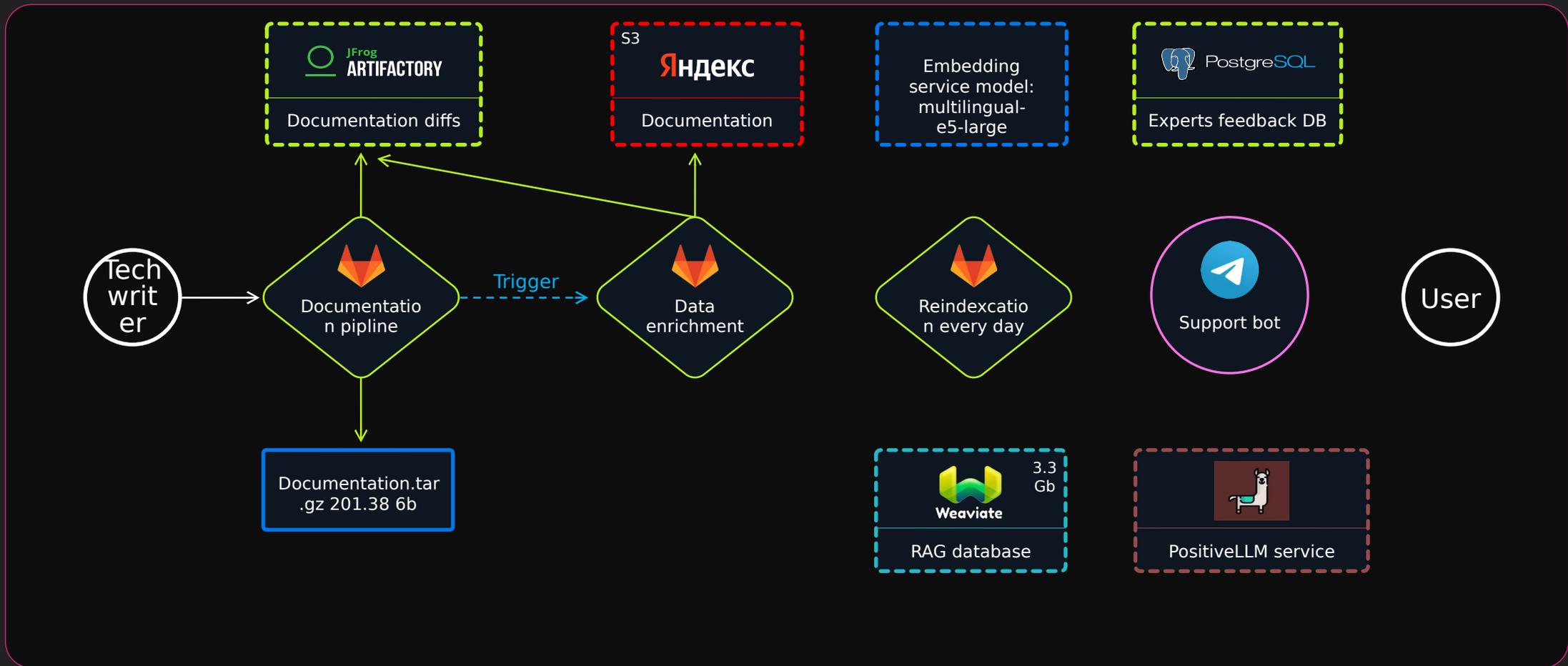
# RAG pipeline



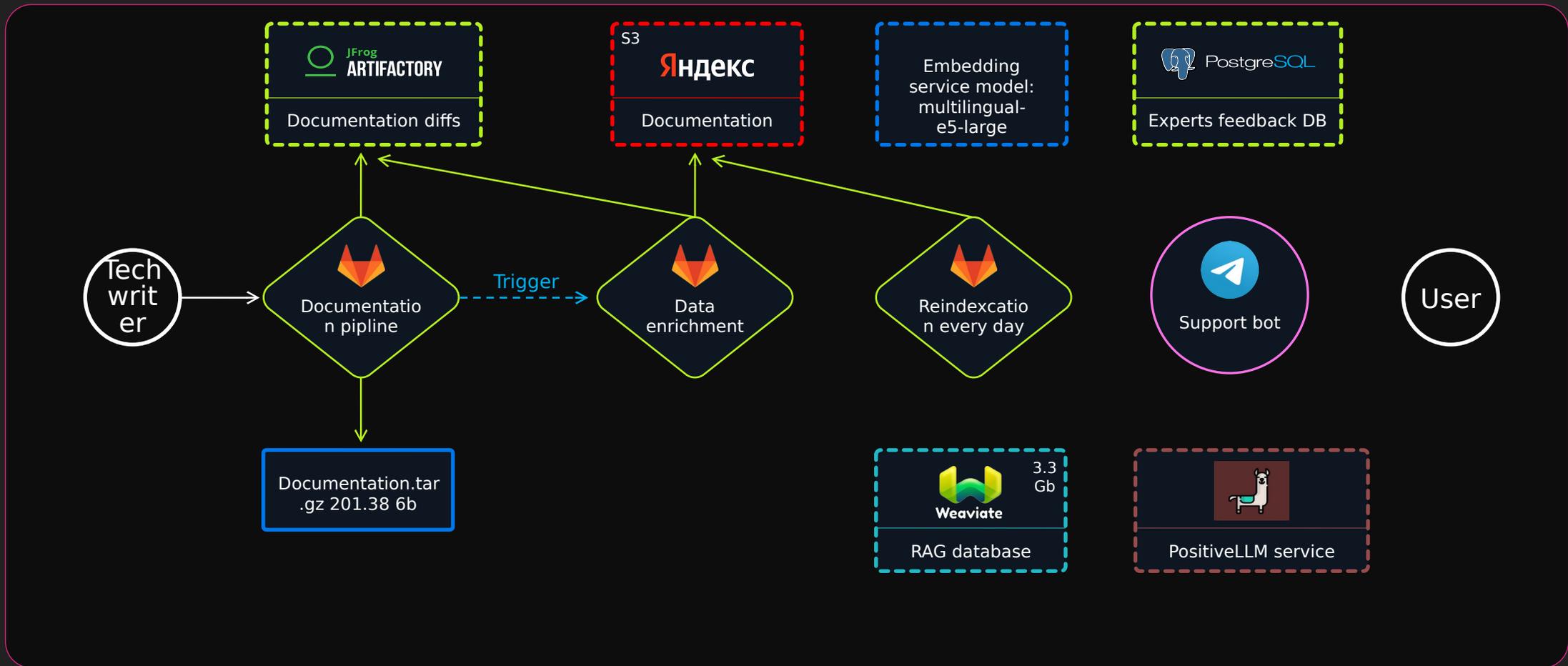
# RAG pipeline



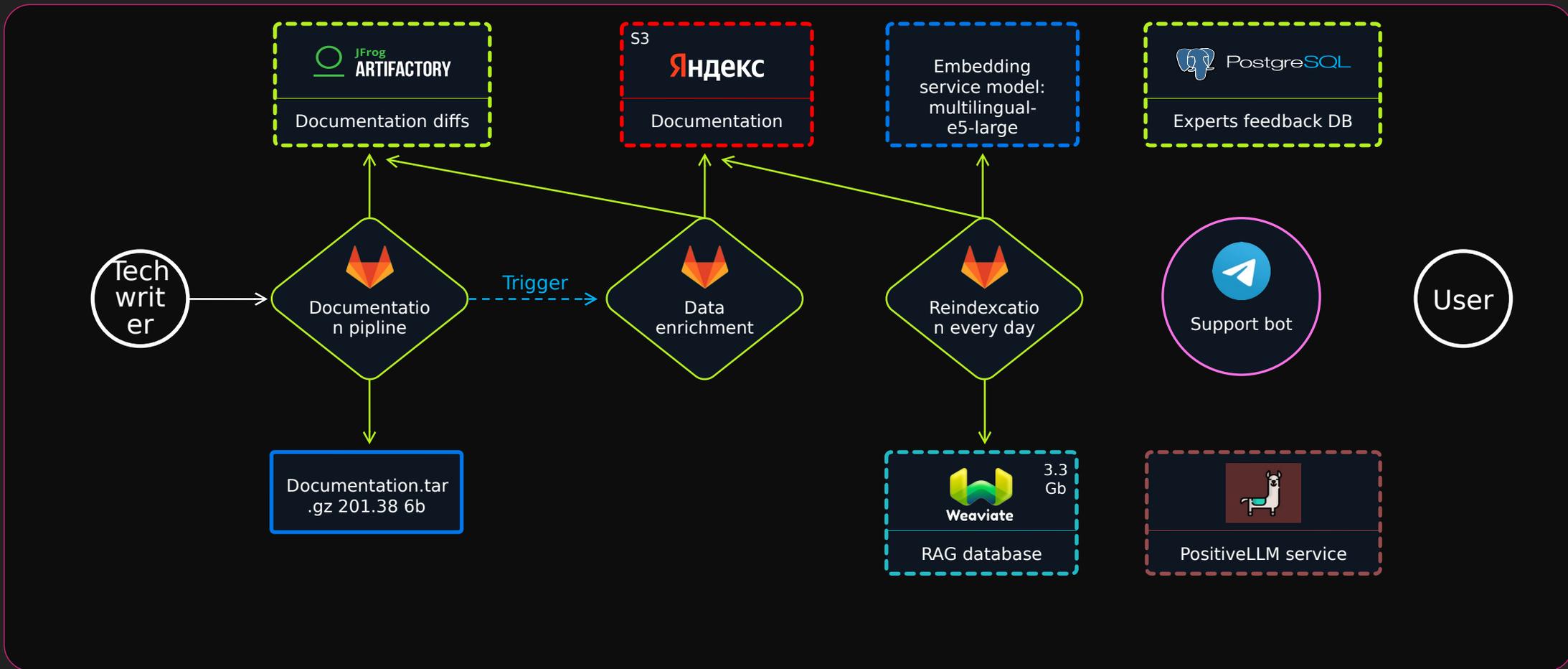
# RAG pipeline



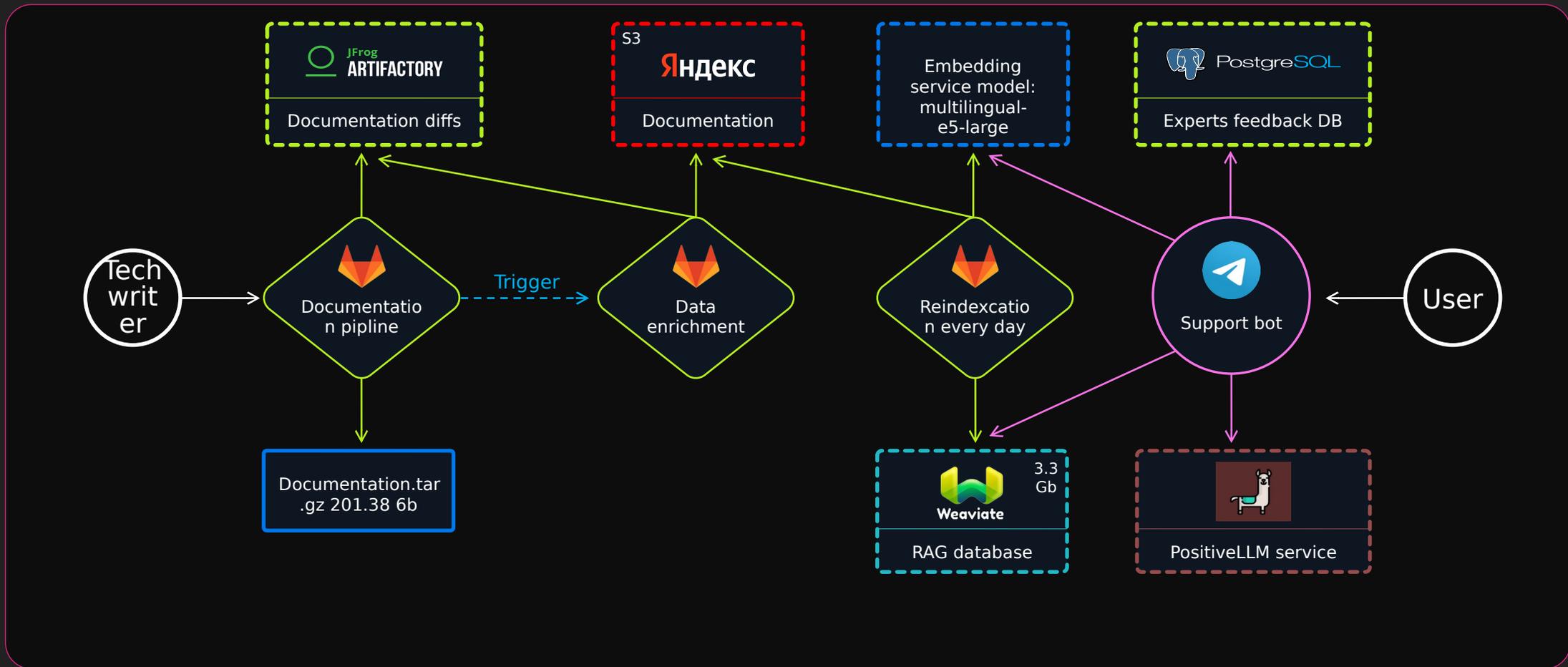
# RAG pipeline



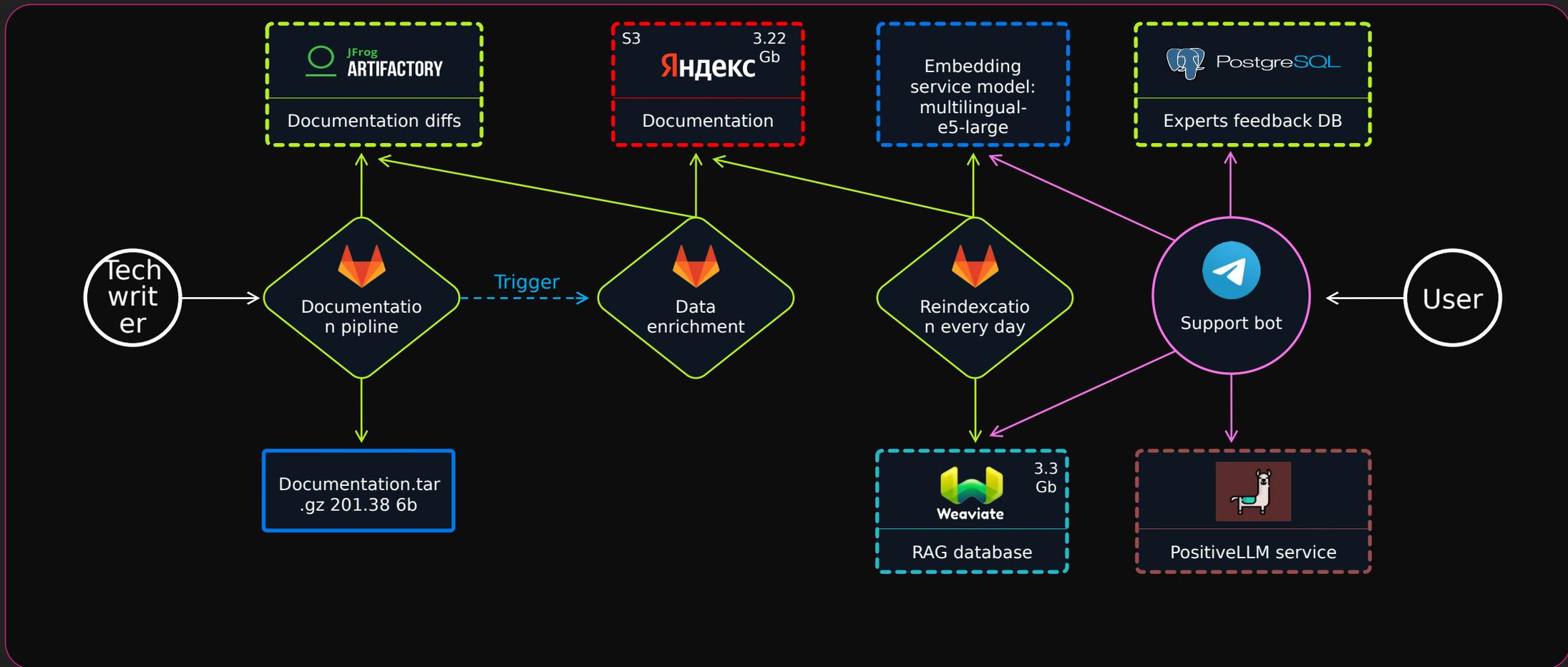
# RAG pipeline



# RAG pipeline



# RAG pipeline



**Что мы ждем от RAG?**  
**Как мы оцениваем RAG?**  
**Как повысить эффективность RAG?**

# Что мы ждем от RAG?



Релевантность  
контекста  
вопросу



Баланс между  
релевантностью  
и разнообразием



Наличие механизмов  
смягчения  
галлюцинаций и  
предвзятости

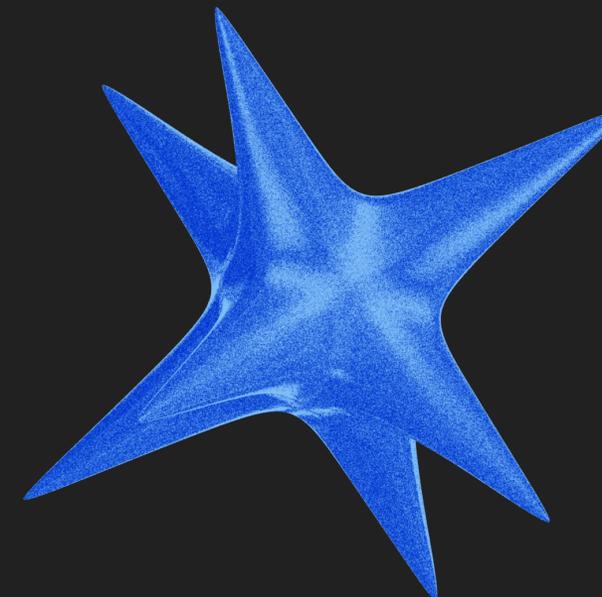
Как составить  
исключение для DDoS  
attacker detected

Галлюцинация:

Чтобы составить исключение для D DoS-атакующего IP-адреса, следуйте указанным ниже шагам:

1. Выберите тип пороговых значений обнаружения DDoS-атак в раскрывающемся списке "Пороговые значения для DDoS-атак" (По умолчанию или Пользовательская настройка).
2. Нажмите кнопку "Изменить". Откроется страница "Пороговые значения для DDoS-атак".
3. Нажмите кнопку "Создать" над списком пороговых значений.
4. Введите IP-адрес DDoS-атакующего в поле "IP-адрес".
5. Установите флажок "Исключить" для этого IP-адреса.
6. Введите причину исключения в поле "Причина исключения".
7. Нажмите кнопку "Сохранить", чтобы сохранить исключение.

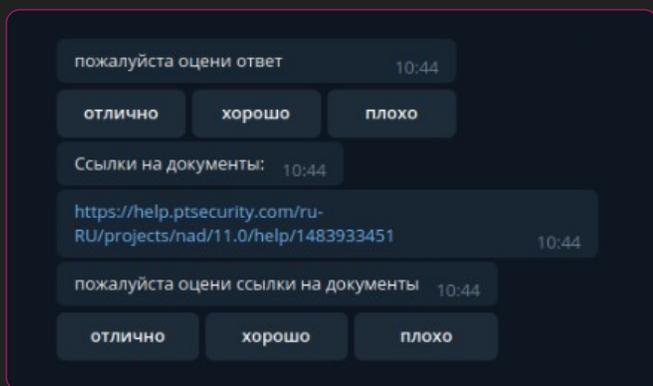
Теперь DDoS-атакующий IP-адрес будет исключен из обнаружения атак.



# Что мы ждем от RAG?

- Найден ли правильный файл? Найден ли правильный контекст? Качество ответа LLM.

## 1 Оценки экспертов



пожалуйста оцени ответ 10:44

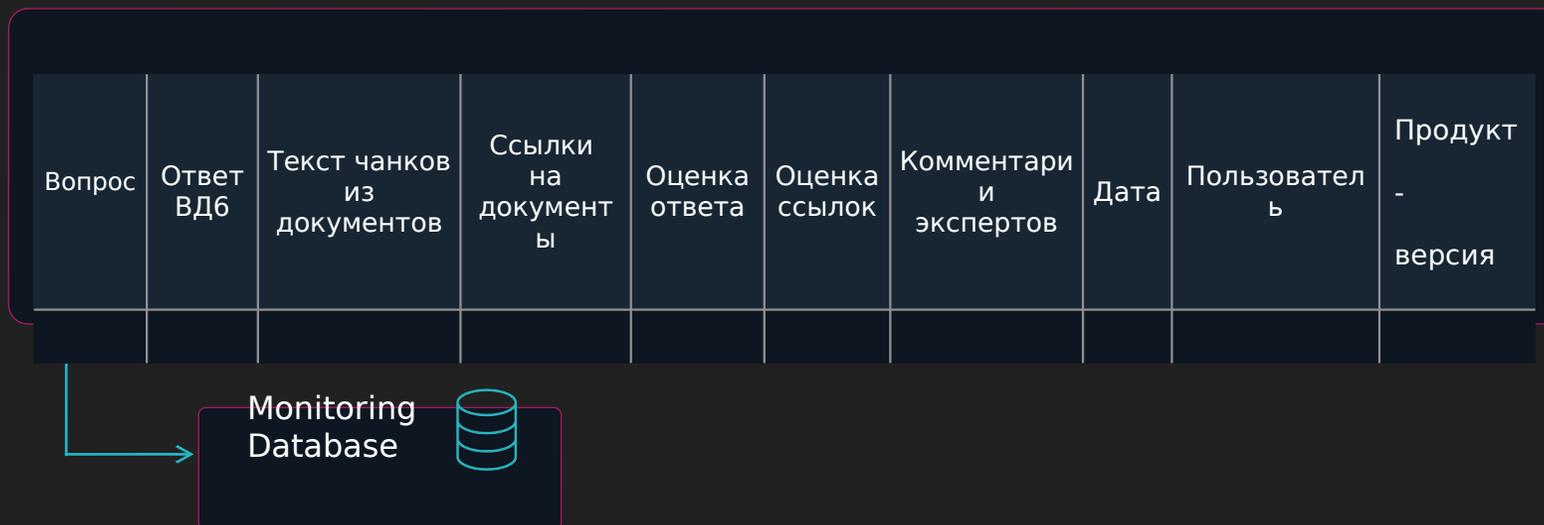
отлично хорошо плохо

Ссылки на документы: 10:44

<https://help.ptsecurity.com/ru-RU/projects/nad/11.0/help/1483933451> 10:44

пожалуйста оцени ссылки на документы 10:44

отлично хорошо плохо

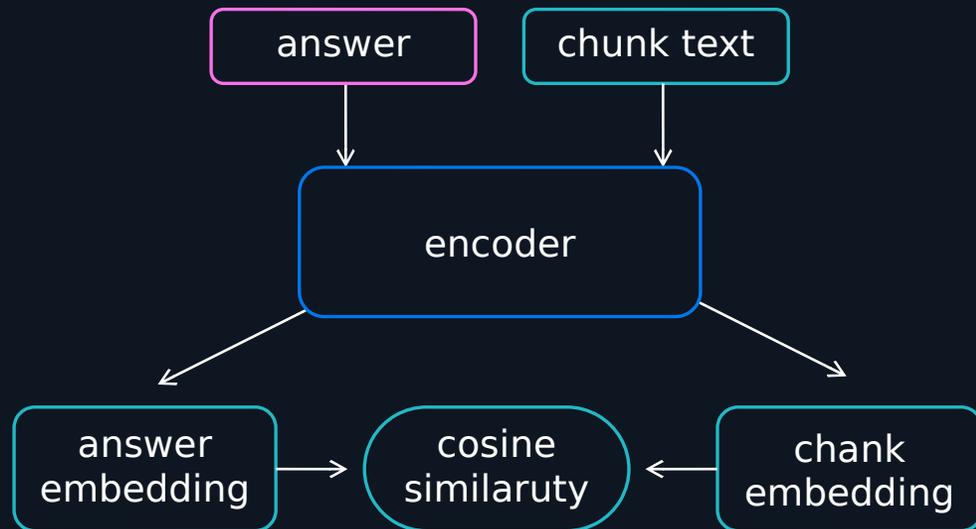


## 2 Валидационная выборка

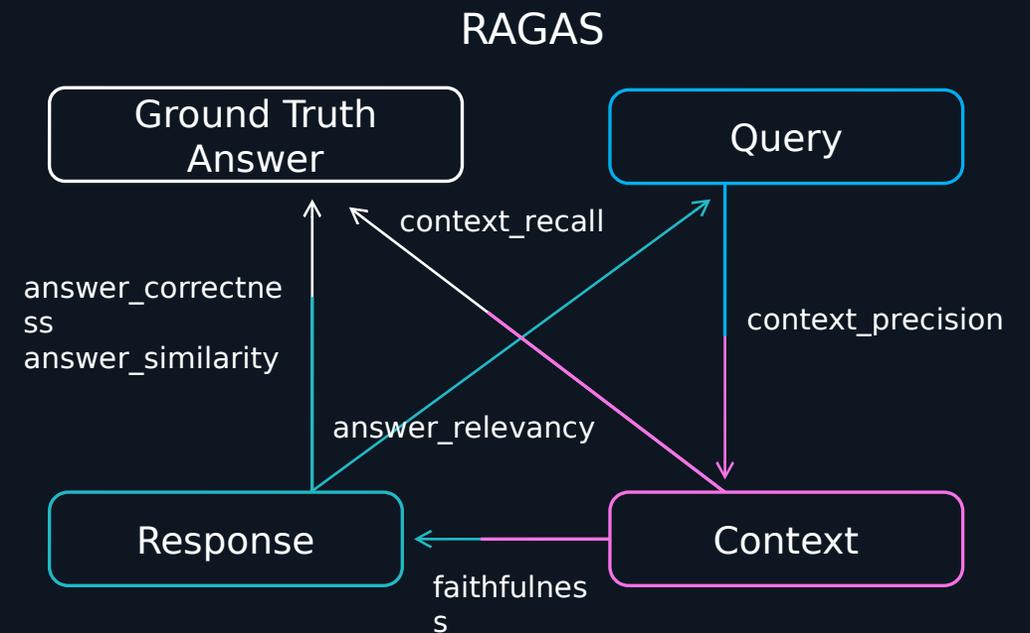
#	Вопрос	Правильный ответ	Контекст	Файл	Комментарии эксперта

# Оценки RAG

## Simple comparison



## With LLM



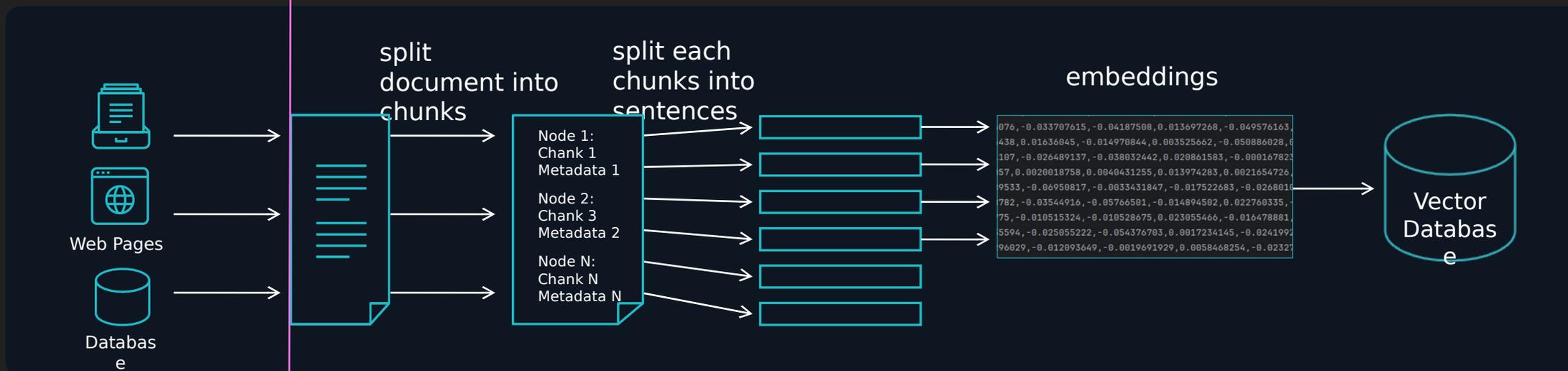
# Как улучшить RAG?

## Preparation

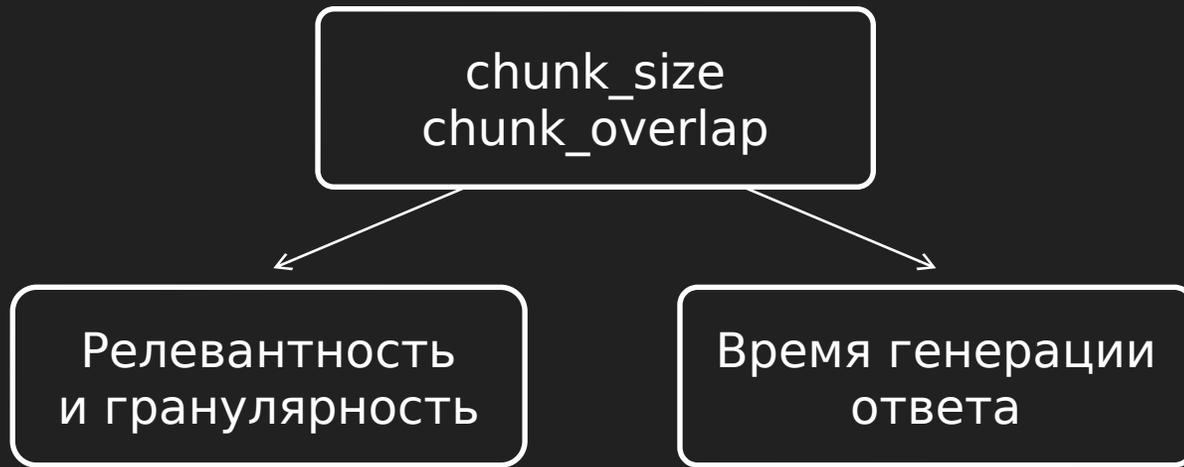
- Выбор формата и парсера
- Препроцессинг данных

## Pre-retrieval optimization

- Выбор эмбединг модели
- Оптимальный размер чанка
- Добавление метаданных

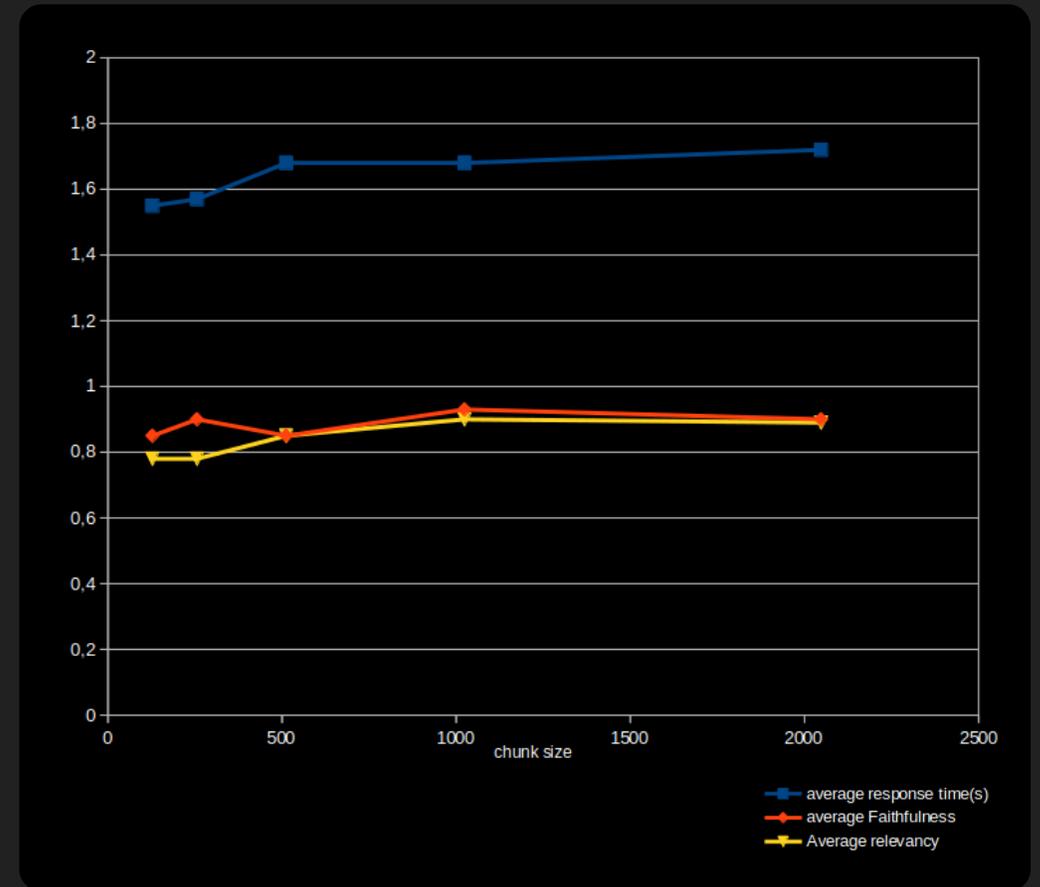


# Подбор параметров индексации



Average  
Faithfulness  
Average relevancy

Average response  
time(s)



# Добавляем метаданные в ноды

## Метаданные для

Заголовок  
документа,  
ключевые слова



формируем  
правильный набор  
нод

- TitleExtractor
- KeywordExtractor
- QuestionsAnsweredExtractor
- SummaryExtractor

```
from llama_index.node_parser.extractors import (
    MetadataExtractor,
    TitleExtractor
)

# Создаем тип сборщика метаинформации
metadata_extractor = MetadataExtractor(
    extractors=[
        TitleExtractor(nodes=5) # указываем количество нод с одним title
    ]
)

# Создаем парсер для нод с нужным свойством
node_parser = SimpleNodeParser(
    metadata_extractor=metadata_extractor
)

nodes1 = response.source_nodes
```

# Добавляем метаданные в ноды

Метаданные для

Получения номера документа



формируем ссылки

Ссылки на документы:

10:4

1

<https://help.ptsecurity.com/ru-RU/projects/nad/11.0/help/1483933451>

10:4

1

пожалуйста оцени ссылки на документы

10:4

4

# Vector DB for RAG

 <b>Pinecone</b>	Proprietary composite index
 <b>Milvus</b>  <b>zilliz</b>	Flat, Annoy, IVF, HNSW/RHNSW (Flat/PQ), DiskANN
 <b>Weaviate</b>	Customized HNSW, HNSW (PQ), DiskANN (in progress...)
 <b>drant</b>	Customized HNSW
 <b>chroma</b>	HNSW
 <b>LanceDB</b>	IVF (PQ), DiskANN (in progress...)
 <b>vespa</b>	HNSW + BM 25 hybrid
 <b>Vald</b>	NGT
 <b>elasticsearch</b>	Flat (brute force), HNSW
 <b>redis</b>	Flat (brute force), HNSW
 <b>pgvector</b>	IVF (Flat), IVF (PQ) in progress...

Source:



# Vector DB for RAG

**Vector DB Comparison**  
by Superlinked | Last Updated: Today

[Get insights](#)
[Give us a star](#)
Views

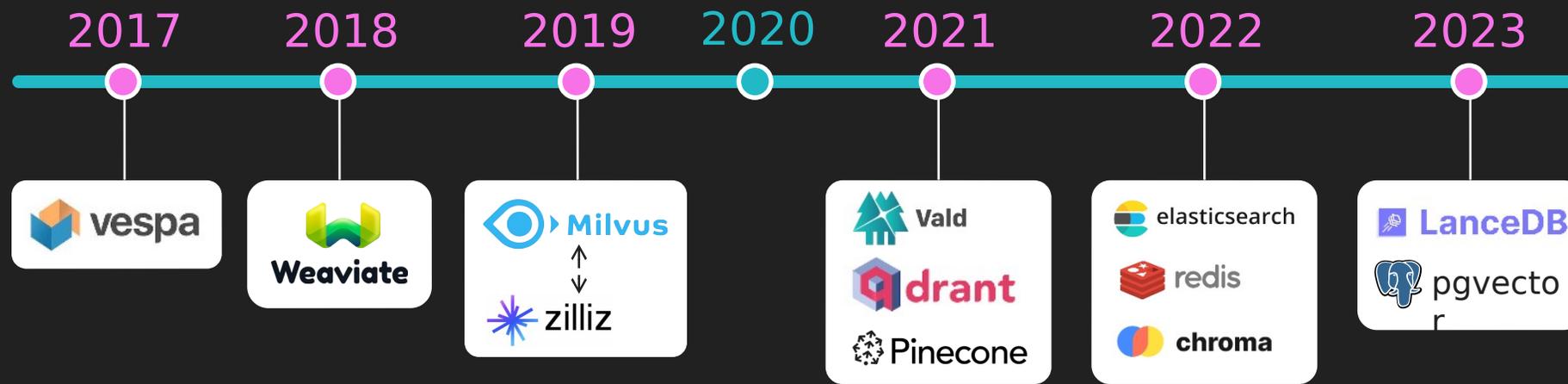
Vendor	About				Search							
	OSS	License	Dev Lang	VSS Launch	Filters	Hybrid Search	Facets	Geo Search	Multi-Vector	Sparse	BM25	
Activeloo...	✓	MPL 2.0	python c++	2023	✓	-	-	✗	✓	✗	✗	
Anari AI	✗	Proprietary	-	2023	-	✗	-	✗	-	-	✗	
Apache C...	✓	Apache-2.0	java	2023	✓	✓	-	-	-	✗	-	
Apache S...	✓	Apache-2.0	java	2022	✓	✓	✓	✓	✓	-	✓	
ApertureDB	-	-	-	-	-	-	-	-	-	-	-	
Azure AI S...	✗	Proprietary	c++	2023	✓	✓	✓	✓	✓	✗	✓	
Chroma	✓	Apache-2.0	python	2022	✓	✗	-	-	✗	✗	✗	
ClickHouse	✓	Apache 2.0	c++	2022	✓	✗	✓	✓	✓	✗	✗	
CrateDB	✓	Apache 2.0	java	2023	✓	-	✓	✓	✓	-	✓	
DataStax ...	✓	Proprietary	java go	2023	✓	✓	✓	✓	✗	✗	✗	
Elasticsea...	✗	Elastic Lice...	java	2021	✓	✓	✓	✓	✓	✓	✓	
Epsilla	✓	GPL-3.0	c++	2023	✓	✓	-	-	✓	✓	-	
GCP Verte...	✗	-	-	2021	✓	✗	-	-	✗	✗	✗	
KDB.AI	✗	Proprietary	python	2023	✓	-	-	-	-	✗	-	
LanceDB	✓	Apache-2.0	rust	2023	✓	✓	-	-	✗	-	✓	
Marqo	✓	Apache-2.0	python	2022	✓	✓	-	-	✓	✗	✓	
Meilisearch	✓	MIT	rust	2023	✓	✓	✓	✓	✓	✗	✗	
Milvus	✓	Apache-2.0	go c++	2019	✓	✓	-	✗	✓	✓	✓	
MongoDB...	✗	GNU GPL ...	c++ java	2023	✓	✓	✓	✓	✓	✗	✓	
MyScale	✗	Proprietary	c++	2023	✓	✓	✓	✓	-	✗	✗	

Source:



# Vector DB for RAG

- Open source
- Self-hosted
- Gooooood documentation
- HNSW
- Full-text search
- Dedicated vector databases
- Infrastructure cost



Source:



# Спасибо за внимание!



Кочева Алина  
[akocheva@ptsecurity.com](mailto:akocheva@ptsecurity.com)

Нурутдинова Алсу  
[anurutdinova@ptsecurity.com](mailto:anurutdinova@ptsecurity.com)

