УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Настройка защищенного конвейера для Retrieval-Augmented Generation (RAG)

с использованием Object Storage в качестве источника доверенных данных

```
#INCLUDE <IOSTREAM>
INT MAIN() {
  STD::COUT << "ИНФОРМАЦИОННАЯ
БЕЗОПАСНОСТЬ";
}
```

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Цели работы:

Теоретическая:

Понять уязвимости RAG-систем, связанные с источниками данных (data poisoning, подмена файлов).

Практическая:

Настроить Yandex Object Storage (или аналогичный S3-совместимый сервис) в качестве защищенного хранилища для документов и реализовать безопасный пайплайн их загрузки и обработки.

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Что такое Retrieval-Augmented Generation?

RAG (Retrieval-Augmented Generation) — это метод, при котором ИИ сначала ищет релевантную информацию в базе данных или документах, а затем использует её для генерации точного и обоснованного ответа.

- **Объединяет поиск + генерацию**
- **Отвечает на основе актуальных/внешних данных**
- **Снижает “галлюцинации” ИИ**

Задача:

«Развернуть безопасное хранилище документов и настроить его интеграцию с RAG-цепочкой»

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

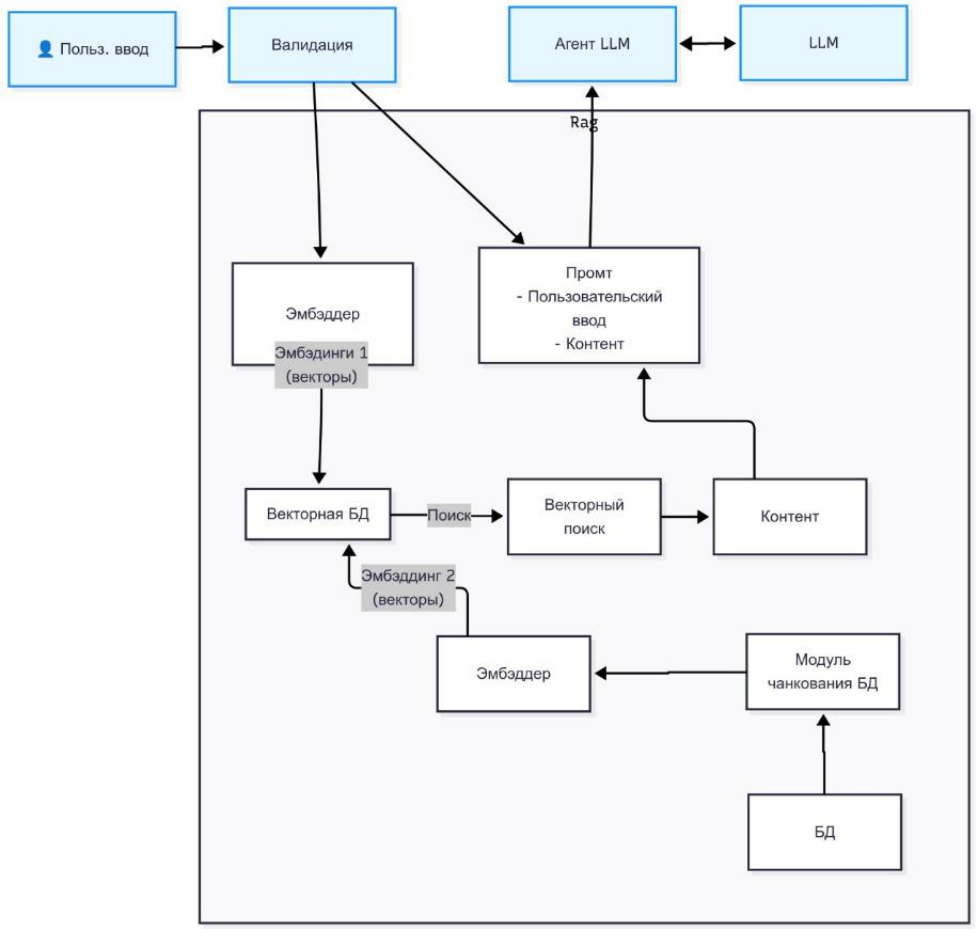
Алгоритм работы rag системы:

1) **База данных в виде неструктурированных данных нарезается на небольшие куски текста** (чанки) – 100-1000 слов (из-за ограничений, накладываемых векторизацией), данный процесс называется чанкованием.

2) **Чанки оцифровываются с помощью** эмбэддера (процесс векторизации) и превращаются в эмбэддинги, т. е. вектора (наборы чисел).

3) **Вектора собираются в векторную БД,** по которой производится поиск.

4) **Когда пользователь посылает запрос,** он эмбэддером кодируется в другие эмбэддинги и по векторной БД ищутся ближайшие соседи.



УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Задание: Развернуть безопасное хранилище документов и настроить его интеграцию с RAG-цепочкой

1.1 Создание бакета и настройка доступа

Создайте бакет, например: rag-docs-trusted.
Загрузите в него только доверенные документы (PDF, TXT — позже можно расширить форматы).

Создайте отдельного пользователя или сервисный аккаунт с минимально необходимыми правами:
Сохраните значения ACCESS_KEY и SECRET_KEY — они понадобятся для интеграции.

1.2. Настройка .env для интеграции

Создайте файл .env в корне проекта со следующими переменными:

```
S3_ENDPOINT=https://storage.yandexcloud.net
S3_ACCESS_KEY=ваш_ключ_только_для_чтения
S3_SECRET_KEY=ваш_секретный_ключ
S3_BUCKET=rag-docs-trusted
S3_PREFIX=docs/ # опционально — подпапка внутри бакета
```

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Задание: Развернуть безопасное хранилище документов и настроить его интеграцию с RAG-цепочкой

1.3 Проверка подключения

Добавьте проверку наличия обязательных переменных при запуске приложения:

```
REQUIRED_VARS = {
    "S3_ENDPOINT": S3_ENDPOINT,
    "S3_ACCESS_KEY": S3_ACCESS_KEY,
    "S3_SECRET_KEY": S3_SECRET_KEY,
    "S3_BUCKET": S3_BUCKET,
}

for var_name, value in REQUIRED_VARS.items():
    if not value or value.strip().lower() == "none":
        raise ValueError(f"{var_name} не задан. Проверьте .env.")
```

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Задание: Реализовать механизм безопасной загрузки документовРеализовать механизм безопасной загрузки документов

Загрузка ≠ индексация. Здесь речь о безопасном скачивании и подготовке документов . из хранилища перед индексацией. **Безопасное скачивание**

2.1 Функция download_from_s3() должна:

1. Проверять, что key — строка, а не None.
2. Пропускать папки (key.endswith('/')).
3. Проверять размер файла до/после скачивания.
4. Ловить исключения при скачивании.
5. Использовать временную директорию.

2.2 Валидация содержимого документов

Перед индексацией — убедитесь, что page_content:

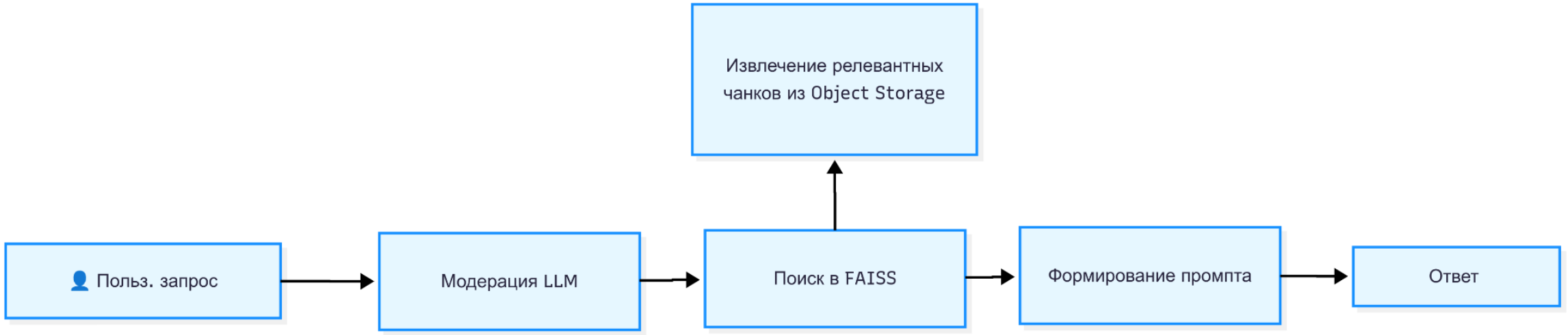
1. Не None
2. Является строкой
3. Не пустая

```
valid_docs = [
    doc for doc in loaded
    if hasattr(doc, 'page_content') and
       isinstance(doc.page_content, str) and
       doc.page_content.strip()
]
```

УрФУ
Институт радиоэлектроники и информационных технологий — РТФ

Задание: Интеграция с RAG-конвейером

Архитектура конвейера:



Безопасная подстановка контекста

1. Контекст вставляется до системного промпта пользователя — не может переопределить инструкции.
2. Используется фиксированный шаблон — пользователь не влияет на структуру.
3. Данные из Object Storage — доверенные, не зависят от ввода пользователя.