**Изображение выглядит как логотип

Автоматически созданное описаниеМИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**(национальный исследовательский университет)»

**Программа стратегического академического лидерства «Приоритет – 2030»**

**ПРОЕКТ «ЦИФРОВАЯ КАФЕДРА»**

**Дополнительная профессиональная программа профессиональной переподготовки  
«Цифровое моделирование и суперкомпьютерные технологии»**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА (IT-ПРОЕКТ)**

на тему: «Создание интеллектуального бота на базе LLM RAG для анализа угроз и рисков в ИБ с правовой оценкой на основе базы угроз ФСТЭК»

Руководитель: (\_\_\_\_\_\_\_\_\_\_)

Консультант: (\_\_\_\_\_\_\_\_\_\_)

Рецензент: (\_\_\_\_\_\_\_\_\_\_)

**К защите допустить**

Руководитель ДПП ПП (\_\_\_\_\_\_\_\_\_\_)

Москва 2025

# СПИСОК ИСПОЛНИТЕЛЕЙ

|  |  |  |  |
| --- | --- | --- | --- |
| № | Фамилия, Имя и Отчество | Название и номер раздела | Подпись |
| 1 | Богачева Анастасия Романовна | Реферат, Введение, Заключение, Выводы по разделу 1, Выводы по разделу 2, Выводы по разделу 3, 3.3 Архитектура и интерфейс Telegram-бота |  |
| 2 | Голубцов Максим Михайлович | 3.1 Условия и сфера применения интеллектуального бота для анализа угроз ИБ |  |
| 3 | Горчакова Эвелина Владиславовна | 2.3 Описание программной разработки |  |
| 4 | Ивкина Татьяна Антоновна | 1.1 Актуальность разработки бота |  |
| 5 | Мамаджанов Амир Илхомович | 1.2 Критический анализ существующих подходов к решению проблемы автоматизации анализа угроз и рисков в ИБ |  |
| 6 | Николашина София Сергеевна | 1.3 Проектирование интеллектуального Telegram-бота с LLM RAG для анализа киберугроз с интеграцией базы ФСТЭК: обоснование цели и задач, технические требования |  |
| 7 | Толпыгин Александр Алексеевич | 3.2 Результаты работы разработанного интеллектуального бота для анализа угроз ИБ |  |
| 8 | Шурлыкин Глеб Евгеньевич | 2.4 План разработки проекта «Интеллектуальный бот для анализа угроз и рисков в информационной безопасности» |  |
| 9 | Плешакова Дарья Андреевна | 2.2 Стек используемых технологий |  |
| 10 | Савинков Владислав Владимирович | 2.1 Теоретическое решение задачи и модель разработки интеллектуального бота |  |

Руководитель:

(\_\_\_\_\_\_\_\_\_\_)

Консультант:

(\_\_\_\_\_\_\_\_\_\_)

Рецензент:

(\_\_\_\_\_\_\_\_\_\_)

# РЕФЕРАТ

Итоговая аттестационная работа состоит из 233 страниц, 31 рисунков, 10 таблиц, 36 использованных источников, 11 приложений.

ИИ, LLM, ИБ, ФСТЭК RAG, ИНТЕГРАЦИЯ С БД, ВЕКТОРНЫЕ БД, TELEGRAM-БОТ, АНАЛИЗ ИБ-УГРОЗ, ГИБРИДНАЯ АРХИТЕКТУРА, PYTHON

Итоговая аттестационная работа выполнена в формате IT-проекта «Создание интеллектуального бота на базе LLM RAG для анализа угроз и рисков в ИБ с правовой оценкой на основе базы угроз ФСТЭК».

Объектом разработки в данной работе является ручной анализ угроз ИБ и их соответствия требованиям ФСТЕК.

Цель работы — оптимизация анализа угроз в ИБ за счёт автоматизации обработки данных из базы ФСТЭК с использованием LLM и RAG.

Для достижения поставленной цели были проведены исследования: анализ ручной обработки угроз ИБ, технологических трендов и практики внедрения ИИ в ИБ. Основное содержание работы состояло в разработке Telegram-бота, архитектурного решения на базе LLM и RAG, концептуальном описании работы с базой данный ФСТЕК.

Основными результатами работы, полученными в процессе разработки, являются реализованный прототип Telegram-бот с выводом справочной информации по ИБ-угрозам и архитектурное решение реализации LLM, RAG и подключения базы данных ФСТЕК.

Данные результаты разработки предназначены для получения правовой оценки рисков при проектировании ИБ-систем. Может быть развернут как корпоративный инструмент для ИБ-службы и интегрирован с другими системами мониторинга угроз.

Применение результатов данной работы позволяет автоматизировать анализ угроз ИБ и обеспечивает соответствие требованиям ФСТЭК. Данное решение минимизирует человеческие ошибки и оптимизирует затраты на ИБ, особенно для организаций с ограниченными ресурсами.

# СОДЕРЖАНИЕ

[СПИСОК ИСПОЛНИТЕЛЕЙ 2](#_Toc200224615)

[РЕФЕРАТ 4](#_Toc200224616)

[СОДЕРЖАНИЕ 5](#_Toc200224617)

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 10](#_Toc200224618)

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 12](#_Toc200224619)

[ВВЕДЕНИЕ 13](#_Toc200224620)

[1 ТЕОРЕТИЧЕСКИЕ И МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНОГО БОТА ДЛЯ АНАЛИЗА УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ 22](#_Toc200224621)

[1.1 Актуальность разработки бота 22](#_Toc200224622)

[1.1.1 Современное состояние информационной безопасности 22](#_Toc200224623)

[1.1.2 Ключевые сложности в практической реализации требований ФСТЭК 23](#_Toc200224624)

[1.1.3 Перспективы автоматизации процессов правовой и технической оценки угроз 24](#_Toc200224625)

[1.1.4 Почему Telegram-бот — это удобно и актуально 27](#_Toc200224626)

[1.1.5 Сравнение ручного и автоматизированного анализа 29](#_Toc200224627)

[1.1.6 Связанность с задачами цифровой трансформации и стратегиями РФ 31](#_Toc200224628)

[1.1.7 Использование в образовательных целях 33](#_Toc200224629)

[1.1.8 Проблематика масштабируемости и унификации 34](#_Toc200224630)

[1.2 Критический анализ существующих подходов к решению проблемы автоматизации анализа угроз и рисков в ИБ 36](#_Toc200224631)

[1.2.1 Статические базы данных угроз 37](#_Toc200224632)

[1.2.2 Системы SIEM (Security Information and Event Management) 41](#_Toc200224633)

[1.2.3 Решение на основе машинного обучения 47](#_Toc200224634)

[1.2.4 Чат-боты на базе LLM 51](#_Toc200224635)

[1.2.5 Ключевые выводы 58](#_Toc200224636)

[1.3 Проектирование интеллектуального Telegram-бота с LLM RAG для анализа киберугроз с интеграцией базы ФСТЭК: обоснование цели и задач, технические требования 59](#_Toc200224637)

[1.3.1 Обоснование цели и задач работы 60](#_Toc200224638)

[1.3.2 Основное назначение интеллектуального бота 61](#_Toc200224639)

[1.3.3 Обоснование разработки интеллектуального Telegram-бота для анализа киберугроз с интеграцией базы ФСТЭК 63](#_Toc200224640)

[1.3.4 Требования к Telegram-боту 64](#_Toc200224641)

[1.3.5 Требования к надежности 67](#_Toc200224642)

[1.3.6 Условия эксплуатации: требуемая квалификация и уровень подготовки пользователя 70](#_Toc200224643)

[1.3.7 Требования к составу и параметрам технических средств (описание требований к hardware). 72](#_Toc200224644)

[1.3.8 Требования к информационной и программной совместимости (описание требований к software) 73](#_Toc200224645)

[1.3.9 Специальные требования 75](#_Toc200224646)

[1.4 Выводы по разделу 1 79](#_Toc200224647)

[2 РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО БОТА ДЛЯ АНАЛИЗА УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ 81](#_Toc200224648)

[2.1 Теоретическое решение задачи и модель разработки интеллектуального бота 81](#_Toc200224649)

[2.1.1 Подробное описание теоретической модели решения. 83](#_Toc200224650)

[2.1.2 Ключевые компоненты модели. 84](#_Toc200224651)

[2.1.3 Структурно-функциональная схема 85](#_Toc200224652)

[2.1.4 Вывод по подразделу 88](#_Toc200224653)

[2.2 Стек используемых технологий 90](#_Toc200224654)

[2.2.1 Введение в стек технологий и цели разработки 90](#_Toc200224655)

[2.2.2 Архитектура решения 96](#_Toc200224656)

[2.2.3 Стек технологий 101](#_Toc200224657)

[2.2.4 Вывод по подразделу 112](#_Toc200224658)

[2.3 Описание программной разработки 114](#_Toc200224659)

[2.3.1 Требования к программному обеспечению 115](#_Toc200224660)

[2.3.2 Общая архитектура программного комплекса 117](#_Toc200224661)

[2.4 План разработки проекта «Интеллектуальный бот для анализа угроз и рисков в информационной безопасности» 128](#_Toc200224662)

[2.4.1 Календарный план разработки 129](#_Toc200224663)

[2.4.2 Состав команды 130](#_Toc200224664)

[2.4.3 Среда разработки 132](#_Toc200224665)

[2.4.4 Детальное описание задач 133](#_Toc200224666)

[2.4.5 Стратегия взаимодействия команды 135](#_Toc200224667)

[2.4.6 План управления рисками 135](#_Toc200224668)

[2.4.7 Вывод и дальнейшие шаги 136](#_Toc200224669)

[2.5 Выводы по разделу 2 137](#_Toc200224670)

[3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ TELEGRAM-БОТА И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ 139](#_Toc200224671)

[3.1 Условия и сфера применения интеллектуального бота для анализа угроз ИБ 139](#_Toc200224672)

[3.1.1 Технические условия применения интеллектуального бота 140](#_Toc200224673)

[3.1.2 Организационные условия применения интеллектуального бота 143](#_Toc200224674)

[3.1.3 Правовые условия применения интеллектуального бота 144](#_Toc200224675)

[3.1.4 Сфера применения интеллектуального бота для анализа угроз ИБ 146](#_Toc200224676)

[3.1.5 Рекомендуемые характеристики для реализации интеллектуального бота 150](#_Toc200224677)

[3.1.6 Руководство по эксплуатации интеллектуального бота для анализа угроз ИБ 151](#_Toc200224678)

[3.1.7 Итог 152](#_Toc200224679)

[3.2 Результаты работы разработанного интеллектуального бота для анализа угроз ИБ 154](#_Toc200224680)

[3.2.1 Описание структуры системы управления конфигурацией 158](#_Toc200224681)

[3.2.2 Описание структуры ядра программы 160](#_Toc200224682)

[3.2.3 Описание структуры для установки команд 161](#_Toc200224683)

[3.2.4 Описание структуры для работы с базой данных 162](#_Toc200224684)

[3.2.5 Описание структуры для создания inline-кнопок 163](#_Toc200224685)

[3.2.6 Описание структуры основы бота 166](#_Toc200224686)

[3.2.7 Итоги 169](#_Toc200224687)

[3.3 Архитектура и интерфейс Telegram-бота 171](#_Toc200224688)

[3.3.1 Стартовая логика 171](#_Toc200224689)

[3.3.2 Раздел «Злоумышленники» 174](#_Toc200224690)

[3.3.3 Раздел «Виды угроз» 175](#_Toc200224691)

[3.3.4 Раздел «Способы защиты» 177](#_Toc200224692)

[3.3.5 Раздел «Другой вопрос» — и будущее LLM-RAG 181](#_Toc200224693)

[3.4 Выводы по разделу 3 183](#_Toc200224694)

[ЗАКЛЮЧЕНИЕ 184](#_Toc200224695)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 189](#_Toc200224696)

[ПРИЛОЖЕНИЕ А 193](#_Toc200224697)

[ПРИЛОЖЕНИЕ Б 196](#_Toc200224698)

[ПРИЛОЖЕНИЕ В 197](#_Toc200224699)

[ПРИЛОЖЕНИЕ Г 198](#_Toc200224700)

[ПРИЛОЖЕНИЕ Д 199](#_Toc200224701)

[ПРИЛОЖЕНИЕ Е 200](#_Toc200224702)

[ПРИЛОЖЕНИЕ Ж 218](#_Toc200224703)

[ПРИЛОЖЕНИЕ З 219](#_Toc200224704)

[ПРИЛОЖЕНИЕ И 220](#_Toc200224705)

[ПРИЛОЖЕНИЕ К 225](#_Toc200224706)

[ПРИЛОЖЕНИЕ Л 227](#_Toc200224707)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящей итоговой аттестационной работе применяют следующие термины с соответствующими определениями:

Автоматизация — внедрение технологий, алгоритмов и систем, которые заменяют или оптимизируют ручной труд, выполняя задачи с минимальным участием человека

Векторные базы данных – это специализированный тип баз данных, предназначенный для хранения, управления и запроса высокоразмерных векторов. Векторы в данном контексте — это числовые представления неструктурированных данных, таких как текст, изображения, аудио и видео. Термин — определение

Киберинцидент — это событие, которое фактически или потенциально угрожает конфиденциальности, целостности, подлинности, доступности и сохранности информации, а также представляет собой нарушение (угрозу нарушения) политик безопасности

Контекстуализация — процесс идентификации данных, относящихся к объекту (например, человеку или городу), на основе контекстуальной информации объекта

Ранжирование — это процесс сортировки результатов поиска по определённым критериям релевантности и значимости.

Самовнимание — это ключевой механизм, который позволяет моделям взвешивать важность различных частей одной входной последовательности при обработке информации

Токенизация — это процесс замены конфиденциальных данных уникальным идентификатором (токеном) для их безопасной обработки и передачи

Уязвимость — недостаток в системе, который может быть использован злоумышленниками для нарушения её целостности и неправильной работы

Эмбеддинг — это числовое представление данных, которое позволяет моделям анализировать и интерпретировать текст.

AES (Advanced Encryption Standard) — это алгоритм симметричного шифрования

Aiogram — это асинхронная библиотека для языка программирования Python, основанная на Telegram Bot API

DLP (Data Loss Prevention) — это комплекс программных и аппаратных решений, предназначенных для предотвращения утечки конфиденциальной информации за пределы корпоративной сети

ISO/IEC 27005 — международный стандарт, разработанный Международной организацией по стандартизации (ISO) и Международной электротехнической комиссией (IEC)

LLM (Large Language Model) — это большая языковая модель, программа искусственного интеллекта, которая может распознавать и генерировать текст

ML (Machine Learning) — совокупность методов искусственного интеллекта, с помощью которых можно создавать самообучающиеся компьютерные системы

NLP (Natural Language Processing) — это область искусственного интеллекта, которая фокусируется на взаимодействии между компьютерами и людьми с помощью естественного языка

RAG (Retrieval-Augmented Generation) — это технология, которая соединяет языковую модель с внешней базой знаний, что позволяет AI-помощнику находить релевантные документы и генерировать ответы на основе актуальных данных

RSA (Rivest-Shamir-Adleman) — это криптографический алгоритм с открытым ключом

SIEM (Security Information and Event Management) — система управления информационной безопасностью и событиями безопасности

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей итоговой аттестационной работе применяют следующие сокращения и обозначения:

БДУ ФСТЭК — Банк данных угроз безопасности информации ФСТЭК России

NVD — National Vulnerability Database

NIST — The National Institute of Standards and Technology

CVSS — Common Vulnerability Scoring System (открытый стандарт, используемый для расчёта количественных оценок уязвимости в безопасности компьютерной системы)

CI/CD — Continuous Integration, Continuous Delivery (это автоматизированный процесс разработки и доставки программного обеспечения)

ИБ — Информационная безопасность

Системы SIEM — Security Information and Event Management

DLP — Data Loss Prevention («предотвращение утечек конфиденциальных данных») от компании Zecurion)

IDS — Intrusion Detection System (класс решений для обнаружения вторжений в систему корпоративной безопасности)

GDPR — General Data Protection Regulation

PCI DSS — Payment Card Industry Data Security Standard (это стандарт безопасности данных для платёжных карт, разработанный основными брендами кредитных карт (Visa, Mastercard, American Express и другими))

HIPAA — Health Insurance Portability and Accountability Act

SCADA — Supervisory Control And Data Acquisition (система диспетчерского управления и сбора данных, которая предназначена для мониторинга и управления промышленными процессами)

AI — Artificial Intelligence

ML — Machine Learning

LLM — Large Language Model

# ВВЕДЕНИЕ

Актуальность темы данной работы связана со стремительной динамикой роста кибератак, объектами которых становятся как физические лица, так и организации различных форм собственности. Особую опасность представляют угрозы объектам критической информационной инфраструктуры (КИИ). Согласно заявлению статс-секретарь-замглавы министерства цифрового развития РФ Ивана Лебедева, ежегодный прирост кибернарушений в РФ достигает 25-30%.

Последствие таких атак – нарушение целостности информационных систем, компрометация персональных данных – приводят не только к значительным финансовым потерям организации, но и к существенному репутационному ущербу организаций.

По оценке ФСТЭК на 13 февраля 2025 года:

1. 47% из 170 организаций КИИ находятся в критическом состоянии защиты от киберугроз;
2. 40% демонстрируют низкий уровень защищенности;
3. только 13% достигли минимальный базовый уровень защиты.

Рисунок 1 наглядно демонстрирует уровень защиты КИИ.

Рисунок 1 – Оценка ФСТЭК информационной безопасности КИИ

С развитием цифровых технологий наблюдается значительное усложнение методов преодоления средств защиты информации, активно используемых злоумышленниками. Это включает применение методов машинного обучения и эксплуатацию уязвимостей в новых технологических платформах. Рост количества кибератак обусловлен рядом факторов, в том числе недостаточной эффективностью традиционных систем защиты, дефицитом квалифицированных кадров в области информационной безопасности, а также высокой трудоемкостью ручного анализа угроз.

Параллельно с этим процессы цифровой трансформации и внедрения инновационных решений создают двунаправленный эффект. Они расширяют арсенал потенциальных векторов атак, но также они предоставляют возможности для разработки принципиально новых методов защиты данных. В данном контексте особый научный и практический интерес представляют гибридные системы, основанные на комбинации технологий больших языковых моделей (LLM) и генерации с подкреплением выборкой (RAG).

Согласно исследованиям, представленным в работах «ControlNet: A Firewall for RAG-based LLM System» [1] и «Securing RAG: A Risk Assessment and Mitigation Framework» [2] подобные системы демонстрируют значительный потенциал в области информационной безопасности. Благодаря им можно обеспечивать автоматизированное извлечение релевантных данных из нормативных баз без ручного поиска, поддерживать актуальность информации за счет автоматически обновляемого векторного хранилища.

Следует особо подчеркнуть, что, несмотря на доказанную эффективность технологий LLM и RAG в зарубежной практике – где они успешно применяются для автоматизации анализа угроз, интеграции с нормативными базами данных и минимизации зависимости от человеческого фактора, – в российской системе информационной безопасности эти решения пока не получили системного внедрения.

Основные ограничения текущего состояния разработок связаны с необходимостью:

1. адаптации существующих технологий к специфике российского нормативного поля, включая требования ФСТЭК, ФСБ и Роскомнадзора;
2. разработки специализированных методов обработки документов на русском языке;
3. обеспечения соответствия требованиям национальных стандартов информационной безопасности.

При этом потенциал применения данных технологий в российских условиях остается значительным, что подтверждается успешными пилотными проектами ряда ведущих организаций в области ИБ: Проект Ростелекома по автоматизации анализа угроз, Проект НИИ "Восход" Минцифры, которые направлены на снижение времени классификации инцидентов и интеграции с отечественными системами мониторинга. Однако для достижения полноценной операционной готовности требуются дополнительные исследования и разработки, направленные на преодоление указанных ограничений.

По данным исследования рынка труда в области информационной безопасности в 2024-2027 гг. [3] несмотря на рост числа занятых в сфере ИБ за период 2016-2023, существенно вырос дефицит кадров: в 2010-х нахватало 25-30% от числа занятых в отрасли, то к 2023 году – уже 45%. В абсолютных значениях дефицит составил порядка 50 тысяч квалифицированных специалистов по состоянию на 2023 год.

В 2027 году ожидается рост общей потребности рынка до 235-261 тысячи специалистов, при этом занятость составит 181–196 тысяч человек. Абсолютный дефицит может достичь 54-65 тысяч специалистов. В относительном выражении - 23-25% от потребности рынка. На рисунках 2 и 3 продемонстрирован дефицит кадров в ИБ в 2023 и 2027 гг. соответственно.

Рисунок 2 – Потребность рынка труда в сфере ИБ в разрезе функциональных групп работников в 2023 году

Рисунок 3 – Изменение потребности рынка труда в сфере в 2027 году

Современная ситуация в сфере информационной безопасности характеризуется значительным дефицитом квалифицированных специалистов, что усугубляется постоянным усложнением киберугроз и необходимостью непрерывного профессионального развития сотрудников. Данные обстоятельства создают существенные трудности для организаций в части обеспечения эффективного и оперативного реагирования на инциденты информационной безопасности.

Согласно данным исследования IBM "Cost of a Data Breach Report 2023" [4], системное внедрение технологий искусственного интеллекта и автоматизированных решений в сфере информационной безопасности позволяет организациям достичь значительного снижения финансовых потерь при ликвидации последствий кибератак.

В частности, отмечается, что организации, использующие современные технологии автоматизации, демонстрируют экономию в размере 2,22 миллиона долларов США на один инцидент по сравнению с компаниями, полагающимися исключительно на ручные методы анализа и реагирования.

Дополнительные исследования, опубликованные в "Журнале информационной безопасности" [5], свидетельствуют о том, что порядка 66% внутренних инцидентов информационной безопасности носят непреднамеренный характер и возникают вследствие недостаточной компетентности или небрежности сотрудников при выполнении служебных обязанностей. Данный фактор существенно усугубляет общую ситуацию с информационной безопасностью в организациях.

Проведенный анализ убедительно демонстрирует критическую необходимость внедрения современных автоматизированных систем анализа и реагирования на угрозы информационной безопасности. Как показывают исследования, технологии искусственного интеллекта позволяют не только минимизировать финансовые потери организаций, но и значительно снизить зависимость от человеческого фактора, остающегося ключевым источником уязвимостей. В условиях стремительного усложнения ландшафта киберугроз реализация таких решений становится стратегическим императивом для обеспечения устойчивого развития организаций.

Комплексное исследование подтверждает особую актуальность разработки автоматизированных решений в области информационной безопасности, основанных на интеграции технологий LLM и RAG с нормативными базами данных. Подобные системы призваны решить три ключевые задачи:

1. Компенсацию острого кадрового дефицита в сфере ИБ;
2. Повышение оперативности реагирования на инциденты;
3. Обеспечение соответствия актуальным регуляторным требованиям.

Таким образом, создание специализированного решения на базе LLM и RAG, адаптированного под российские нормативные базы, представляет собой научно обоснованную и практически значимую инициативу. Ее необходимость подтверждается:

1. Статистическими данными о растущих киберугрозах;
2. Технологическими ограничениями существующих систем;
3. Явно выраженной рыночной потребностью.

Цель работы — оптимизация анализа угроз и рисков в информационной безопасности за счёт автоматизации обработки данных из базы ФСТЭК с использованием LLM и RAG-технологий.

Для достижения поставленной цели в работе были решены следующие задачи:

1. провести анализ проблем ручного анализа угроз ИБ, включая оценку временных затрат и рисков, на основе статистических данных и нормативных требований (ФСТЭК, ГОСТ Р 57580);
2. исследовать возможности применения технологий LLM и архитектуры RAG для автоматизированной обработки запросов и интеграции с базой угроз ФСТЭК;
3. создать прототип функционального Telegram-бота с возможностью вывода справочной информации по ИБ-угрозам и разработать архитектурное решение для подключения LLM, RAG и базы данных ФСТЭК;
4. проанализировать структуру и форматы базы угроз ФСТЭК и разработать методы их интеграции;
5. оценить эффективность предложенного решения: сокращение времени анализа угроз, снижение оперативной нагрузки на специалистов по информационной безопасности.

Объектом разработки является оптимизация анализа угроз и рисков в информационной безопасности за счёт автоматизации обработки данных из базы ФСТЭК с использованием LLM и RAG-технологий.

Предметом исследования в работе является разработка интеллектуальной системы анализа угроз ИБ на базе технологий LLM и RAG с интеграцией базы данных ФСТЭК, включая создание Telegram-бота как пользовательского интерфейса и методы автоматизированной правовой оценки соответствия регуляторным требованиям.

Работа основывалась на следующих инструментах и методах:

1. Large Language Models (LLM)

Использованы для обработки естественно-языковых запросов пользователей и генерации аналитических выводов. Применение LLM обусловлено необходимостью интерпретации сложных формулировок нормативных документов и автоматизации процессов анализа угроз ИБ.

1. Retrieval-Augmented Generation (RAG)

Применены в качестве интеграционного компонента между языковыми моделями и базой угроз. Данная технология обеспечивает доступ к актуальным данным без необходимости переобучения модели, позволяет поддерживать релевантность аналитических выводов.

1. Векторные базы данных

Использованы для эффективного семантического поиска по базе угроз ФСТЭК. Позволяет находить релевантные документы на основе смыслового соответствия.

1. Telegram-бот (библиотека aiogram)

Реализован в качестве пользовательского интерфейса. Выбор этой библиотеки обусловлен требованиями к обеспечению удобного доступа к функциональности системы для специалистов ИБ

Основными результатами, полученными в работе, являются:

1. выявлены и систематизированы ключевые недостатки ручного анализа ИБ: значительные временные затраты на обработку данных, высокую вероятность человеческих ошибок и сложность поддержания актуальности нормативной базы требований;
2. доказана эффективность гибридной архитектуры, сочетающей LLM и RAG, для автоматизированной обработки нормативных документов в информационной безопасности;
3. реализованы прототип функционального Telegram-бота, обеспечивающий возможность вывода справочной информации по ИБ-угрозам и архитектурное решения компонентов LLM, RAG и базы данных ФСТЭК;
4. предложены алгоритм семантической индексации нормативных документов базы угроз ФСТЭК и принципы организации данных для эффективного поиска и анализа;
5. теоретически обоснованы потенциальные преимущества решения: значительное сокращение времени анализа угроз, обеспечение полного соответствия актуальным требованием ФСТЭК и уменьшение операционной нагрузки на специалистов по ИБ.

Результаты работы предназначены для внедрения в организациях сферы информационной безопасности, включая операторов критической информационной инфраструктуры, коммерческие компании, работающие с персональными данными, а также подразделения ИБ государственных учреждений и образовательных организаций.

Использование разработки позволяет существенно повысить эффективность работы подразделений информационной безопасности за счет следующих ключевых преимуществ:

1. Автоматизирует процессы анализа и классификации угроз информационной безопасности, что способствует оперативному реагированию на инциденты безопасности;
2. Гарантирует актуальность аналитических данных благодаря интеграции с официальными источниками информации, включая базы данных ФСТЭК, минимизируя вероятность использования устаревших сведений;
3. Оптимизирует рабочую нагрузку специалистов по информационной безопасности за счет автоматизации рутинных операций по обработке нормативно-правовых требований;
4. Повышает достоверность и объективность оценки соответствия требованиям регуляторов за счет применения формализованных аналитических алгоритмов;
5. Обеспечивает масштабируемость системы контроля информационной безопасности в условиях динамичного развития организаций и усиления регуляторного давления.

# 1 ТЕОРЕТИЧЕСКИЕ И МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНОГО БОТА ДЛЯ АНАЛИЗА УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

## 1.1 Актуальность разработки бота

### 1.1.1 Современное состояние информационной безопасности

В условиях повсеместной цифровизации и быстрых темпов развития ИТ-сферы информационная безопасность (ИБ) приобретает критически важное значение как для частных компаний, так и для государственных организаций. Число киберинцидентов, масштаб кибератак и ущерб, наносимый организациям в результате информационных утечек, ежегодно увеличивается. Согласно отчёту Positive Technologies за 2024 год, количество целевых атак на организации в РФ возросло на 27% по сравнению с предыдущим периодом. Причём основными целями атак становятся объекты критической информационной инфраструктуры (КИИ), государственные информационные системы, а также финансовые учреждения.

В связи с этим государство ужесточило требования к методам проектирования, эксплуатации и оценки систем защиты информации, особенно в секторах, отнесённых к КИИ. Надзор за выполнением этих требований осуществляет Федеральная служба по техническому и экспортному контролю (ФСТЭК России), которая является основным регулятором в данной области.

Нормативная база, регулирующая ИБ, включает множество юридических, методических и технических документов – от федеральных законов до ведомственных приказов. Среди важнейших:

1. Федеральный закон №187-ФЗ «О безопасности критической информационной инфраструктуры РФ»;
2. ГОСТ Р 57580 «Безопасность финансовых (банковских) операций»;
3. приказы ФСТЭК, такие как №17 (по оценке соответствия), №239, №235 и др.

Эти акты детализируют требования к оценке угроз, внедрению мер защиты, категорированию информации, порядку проведения обследования и специфике документации. Таким образом, применимость и интерпретация этих требований является задачей высокой сложности, особенно в условиях их регулярного обновления и изменений в прикладной практике.

### 1.1.2 Ключевые сложности в практической реализации требований ФСТЭК

Согласно действующим требованиям ФСТЭК организации обязаны выполнять ряд действий, включая:

1. определение актуальных угроз для информационной системы;
2. привязка к регламентированным типовым угрозам, содержащимся в базе угроз ФСТЭК;
3. формирование модели угроз;
4. правовая квалификация этих угроз на основании действующего законодательства;
5. последующая автоматизация мер защиты в соответствии с выявленными рисками.

Несмотря на наличие методических рекомендаций, практическая реализация данных требований обычно осуществляется вручную. Как показывает аудит ИБ, проводимый в российских компаниях, наиболее распространёнными проблемами являются:

1. неточности в определении актуальных угроз;
2. формальный (бумажный) характер оценки, не соответствующий реальным процессам безопасности в ИС;
3. отсутствие автоматизированных средств оценки и учета правовых аспектов;
4. низкий уровень взаимодействия между ИБ-службами и юридическими подразделениями;
5. постоянные ошибки в документации при проверках со стороны надзорных ведомств.

Во многом это связано с тем, что специалистам по ИБ приходится буквально вручную «сканировать» большой массив регламентов, выявлять сходства и различия, отслеживать формулировки типов угроз в базе ФСТЭК, проверять соответствие требованиям стандартов.

Такой подход однозначно ведёт к:

1. значительным временным затратам;
2. высокой нагрузке на сотрудников;
3. увеличенному риску правовых несоответствий;
4. повторяющимся ошибкам при проверках;
5. снижению операционной эффективности служб ИБ.

Особенно сильно это ощущается в малых и средних организациях, где зачастую один системный администратор совмещает функции всего ИБ-персонала, либо же такие задачи поручаются лицам без профильной подготовки. Реальность показывает, что большая часть компаний выполняет обязательства перед ФСТЭК, главным образом, «для отчёта», не обеспечивая реальную модель угроз и адекватные меры защиты.

### 1.1.3 Перспективы автоматизации процессов правовой и технической оценки угроз

В условиях стремительного роста числа киберугроз и усиления регуляторных требований в области информационной безопасности (ИБ) автоматизация процессов оценки и документирования рисков становится не просто желательной, а объективно необходимой. Переход от ручного анализа к использованию интеллектуальных систем на базе искусственного интеллекта открывает новые горизонты в правовой и технической экспертизе. Он позволяет не только ускорить процедуры, но и обеспечить необходимую полноту, точность и соответствие существующим нормативам. Такой подход особенно актуален в составе систем класса «decision support systems», обеспечивающих поддержку принятия решений в условиях высокой нагрузки и неопределенности.

Современные компании и государственные организации сталкиваются с необходимостью регулярного проведения анализа угроз, построения моделей угроз (УБИ), составления пакетов доказательной документации по соответствию требованиям ФСТЭК России, Роскомнадзора, Минцифры, международных аудиторов. Ручная реализация этих процессов требует глубокой юридической и технической экспертизы, достаточного опыта и больших временных затрат. Кроме того, человеческий фактор в таких задачах несёт в себе риск неполноты, ошибочной идентификации угроз или некорректной интерпретации формулировок законодательства. В связи с этим автоматизация процессов оценки киберрисков представляется не только перспективным, но и необходимым шагом в условиях современной ИБ-реальности.

Один из наиболее перспективных направлений автоматизации связан с использованием технологий больших языковых моделей. Эти модели уже доказали свою эффективность в таких отраслях, как медицина, юриспруденция, банковское дело, техническая поддержка. Наиболее актуальны функции, которые дают возможность анализировать, понимать и структурировать текстовую информацию высокого объёма, включая естественно-языковые запросы и регламенты отраслевого законодательства. В контексте информационной безопасности LLM позволяют продуктивно решать задачи:

1. прочтения и интерпретации юридических документов (приказы, методические указания, федеральные законы, постановления);
2. привязки угроз и объектов к конкретным требованиям и правовым стандартам;
3. идентификации несоответствий текущей ситуации стандартам безопасности;
4. диалоговой формы взаимодействия с аналитиком, специалистом, сотрудником или преподавателем через чат-бот.

Параллельно с LLM активно развивается подход Retrieval-Augmented Generation (RAG) — это архитектура, сочетающая возможности языковых моделей с инструментами синтаксического поиска и извлечения информации из внешней базы данных (в нашем случае, документация ФСТЭК, каталоги нормативных актов, векторные базы текстов). Благодаря RAG достигается то, чего не хватает обычным LLM: опора на точные, проверенные источники. Автоматизированная система не «придумывает» ответ, а ищет релевантный контекст, проверяет соответствие и только после этого формирует обоснованный вывод.

Такая архитектура особенно эффективна в сфере правовой оценки угроз, поскольку гарантирует максимально четкую ссылку на источник информации: приказ, статью, пункт нормативного документа. Это снимает риски интерпретационных ошибок, что особенно критично при оформлении документов по соответствию, прохождении проверок, аудитов и защите проектов в государственном контуре. Помимо этого, использование RAG-моделей снижает время, необходимое для проведения анализа, с нескольких часов или дней — до десятков секунд, что особенно важно при постоянных изменениях в ландшафте угроз и обновлении методической базы.

Telegram-бот, интегрированный с подобной RAG+LLM платформой, представляет собой практичный пользовательский инструмент, позволяющий реализовать интеллектуальную подсистему оценки киберугроз. Благодаря удобному интерфейсу и унифицированной логике пользователь может сформулировать проблемный вопрос, например: *«Какие угрозы актуальны для КИИ в домене АСУ ТП?»*, и получить не свободный ответ, а выдержку из соответствующих нормативных материалов, дополненную пояснением, с возможностью перехода к первоисточнику.

Таким образом, автоматизация процессов правовой и технической оценки угроз на базе LLM и RAG способна кардинально трансформировать работу специалиста по информационной безопасности. Это снижает трудозатраты, минимизирует риски несоблюдения требований, упрощает аудит и сертификацию, а также расширяет возможности ИБ-служб, особенно в организациях без собственного юридического или экспертного состава. Подобные системы открывают перспективу для масштабного внедрения решений поддержки принятия решений в ИБ, как в госсекторе, так и в коммерческой среде, включая банки, промышленность, учебные заведения и ИТ-компании.

### 1.1.4 Почему Telegram-бот — это удобно и актуально

Выбор формата Telegram-бота в рамках разработки интеллектуальной системы анализа угроз информационной безопасности не является случайным. Telegram как платформа обладает рядом функциональных, технических и социальных преимуществ, которые делают её оптимальной средой для внедрения справочных, аналитических и обучающих систем. Формат чат-бота в Telegram объединяет доступность мобильного мессенджера с возможностями автоматизированной обработки информации и мгновенного взаимодействия с пользователем. Ниже рассмотрены основные причины, по которым использование Telegram-бота является не только удобным, но и актуальным решением с точки зрения практического применения.

Во-первых, стоит отметить широкую доступность и популярность Telegram среди пользователей. Это один из самых распространённых и активно развивающихся мессенджеров, доступный бесплатно на всех платформах — Windows, macOS, Linux, iOS, Android, а также в веб-версии. Telegram не требует мощных ресурсов и дополнительной авторизации; большинство пользователей уже имеют установленное приложение, что значительно снижает порог входа и повышает внедряемость разработанного решения. Нет необходимости скачивать сторонние приложения или проходить этапы сложной настройки — бот становится практически мгновенно доступным широкой аудитории.

Во-вторых, одним из ключевых преимуществ Telegram-ботов является круглосуточная работа и мобильность. Пользователь может обратиться к боту в любое время суток, независимо от времени и местоположения. Это особенно важно в сфере информационной безопасности, где требуется оперативный анализ инцидентов, уточнение нормативных документов или быстрый доступ к рекомендациям по устранению угроз. Благодаря тому что мессенджер работает на смартфонах и ПК, система становится по-настоящему мобильной и надежной опорой в профессиональной деятельности ИБ-специалиста.

В-третьих, бот легко встраивается в существующие рабочие процессы и ИТ-среду предприятия. Telegram широко применяется в межведомственном и корпоративном общении, а значит, бот может органично дополнить каналы взаимодействия внутри команд, технической поддержки или инженерных отделов. Встроенные функции команд, кнопок, уведомлений и логики позволяют использовать Telegram-бота как универсальный интерфейс доступа к базе знаний, справочной системе, аналитическим материалам и даже инструментам DevSecOps-среды. Более того, в перспективе бот может служить интерфейсом для интеграции с другими сервисами: базами угроз, системами мониторинга безопасности, SIEM-решениями и т.д.

Кроме того, Telegram-бот обладает высокой масштабируемостью и универсальностью. Такое решение подходит как для небольших коллективов и учебных заведений, так и для крупных компаний и государственных структур. Архитектура Telegram Bot API и его гибкие возможности позволяют легко адаптировать и расширять функциональность при росте объема пользователей или добавлении новых сценариев взаимодействия. Поддержка авторизации, разделения уровней доступа, кастомизация внешнего вида и логики — всё это делает бота гибким инструментом, не требующим пересмотра архитектуры при расширении его функционала. Особенно важно, что все элементы системы разворачиваются без необходимости покупать лицензии, серверы или сложное ПО — поддержание и масштабирование Telegram-бота не требует существенных расходов.

Таким образом, использование формата Telegram-бота для решения задач в области информационной безопасности выглядит не только оправданным, но и стратегически целесообразным. Это не просто интерфейс или приложение — это новый тип взаимодействия, сочетающий доступность, автономность, масштабируемость и простоту. Благодаря своей универсальности и скорости работы, Telegram-бот становится эффективным каналом оперативной поддержки, просвещения и автоматизированного консалтинга в вопросах защиты информации. В условиях быстро меняющегося киберландшафта использование такого инструмента даёт организациям возможность оставаться гибкими, информированными и защищенными.

### 1.1.5 Сравнение ручного и автоматизированного анализа

Для оценки эффективности разрабатываемой системы был проведён сравнительный анализ между традиционным, ручным способом анализа угроз и защитных мер и предлагаемым решением, основанным на использовании Telegram-бота, дополненного технологией LLM (Large Language Model) и подходом RAG (Retrieval-Augmented Generation). Таблица 1 демонстрирует ключевые различия между этими двумя подходами по основным критериям, непосредственно влияющим на качество, скорость и удобство использования системы в условиях реальной практики обеспечения информационной безопасности.

Таблица 1 – Сравнение анализов

|  |  |  |
| --- | --- | --- |
| Критерий | Ручной анализ | Система с LLM + RAG + Telegram-бот |
| Временные затраты | От 8 до 40 часов | От 30 секунд до нескольких минут |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Необходимость в высококвалиф. персонале | Да | Минимальная |
| Возможность обновления под новые требования | Только вручную | Автоматически, при обновлении базы |
| Актуальность данных | Не гарантирована | Актуально при загрузке последних данных |
| Человеческий фактор | Высокий риск ошибок | Минимальный |
| Документируемость решений | Сложно обеспечивается | Автоматически формируются ссылки и источники |
| Интеграция в корпоративные процессы | Затруднена | Возможно через API, облака, чаты |

Для оценки эффективности разрабатываемой системы был проведён сравнительный анализ между традиционным, ручным способом анализа угроз и защитных мер и предлагаемым решением, основанным на использовании Telegram-бота, дополненного технологией LLM (Large Language Model) и подходом RAG (Retrieval-Augmented Generation). Таблица демонстрирует ключевые различия между этими двумя подходами по основным критериям, непосредственно влияющим на качество, скорость и удобство использования системы в условиях реальной практики обеспечения информационной безопасности.

### 1.1.6 Связанность с задачами цифровой трансформации и стратегиями РФ

В последние годы Российская Федерация активно реализует стратегию цифровой трансформации как одного из ключевых направлений социально-экономического и технологического развития. В условиях формирования цифрового государства и ускоренного внедрения цифровых платформ во все сферы экономики, особое место занимает обеспечение технологического суверенитета, импортонезависимости в ИТ-сфере, а также национальной кибербезопасности — как одного из базовых элементов цифровой устойчивости.

В этом контексте информационная безопасность приобретает системный и инфраструктурный характер, переставая быть второстепенной задачей для узкой ИБ-службы. Она становится фундаментом, на котором строится доверие к цифровым сервисам — как в глазах граждан, так и со стороны бизнеса и органов государственной власти.

Разрабатываемая в рамках проекта система — Telegram-бот на основе LLM и RAG, направленная на интеллектуальный анализ киберугроз и обеспечение нормативного соответствия требованиям ФСТЭК и других регуляторов — ставит своей целью реализацию этих задач на прикладном уровне. Подобные решения полностью укладываются в рамки ключевых государственных документов стратегического планирования, в числе которых:

1. Национальная программа «Цифровая экономика Российской Федерации», где основными ориентирами определены цифровая трансформация отраслей, обеспечение цифровой среды доверием, развитие современных цифровых платформ и средств защиты;
2. Стратегия развития информационного общества в Российской Федерации на 2017–2030 годы, в которой подчеркивается необходимость создания интеллектуальной информационной инфраструктуры, включающей в себя системы автоматизированного анализа, поиска и обработки информации, а также интеграции ИИ в государственное управление;
3. Стратегия технологического развития Российской Федерации на период до 2035 года, акцентирующая переход к экономике данных, цифровым платформам, искусственному интеллекту и анализу больших объемов информации для выработки управленческих решений;
4. Программа цифровой трансформации федеральных и региональных органов исполнительной власти до 2030 года, предусматривающая внедрение интеллектуальных цифровых помощников, автоматическую проверку на соответствие нормам, цифровизацию комплаенс-процессов и оптимизацию ИБ-инфраструктуры без увеличения издержек.

В рамках этих стратегий особое внимание уделяется вопросам автоматизации процессов оценки рисков, анализу регуляторных требований, обеспечению правового комплаенса, а также формированию цифровых экосистем с высокой степенью защищенности. Использование систем, подобных Telegram-боту на базе искусственного интеллекта и архитектуры Retrieval-Augmented Generation (RAG), отвечает этим требованиям в полной мере.

Автоматизация сопровождения ИБ и нормативных норм позволяет органам власти, ведомствам, госкомпаниям и бизнесу:

1. Получать оперативные и достоверные консультации по действующим киберугрозам и способам их нейтрализации;
2. Обеспечить прозрачную и аргументированную работу системы принятия решений в части информационной безопасности;
3. Снимать нагрузку с ограниченного числа специалистов ИБ в условиях кадрового дефицита;
4. Повысить скорость реагирования на новые регуляторные инициативы.

Важно подчеркнуть, что предлагаемая система, за счет использования открытых стандартов, возможности локального развертывания и настройки под нужды конкретных заказчиков, отвечает задачам импортозамещения, что делает её особенно привлекательной для использования в инфраструктурах государственного и критически значимого уровня.

Таким образом, проект демонстрирует не только техническую состоятельность как интеллектуальная разработка, но и напрямую связан с приоритетами государственной политики, что открывает перед ним как исследовательские, так и прикладные перспективы. Он может быть масштабирован и адаптирован под типовые сценарии, рекомендованные в «дорожных картах» цифровой трансформации, становясь составной частью национальной цифровой экосистемы.

### 1.1.7 Использование в образовательных целях

Разработка интеллектуальных информационных систем, таких как Telegram-бот для анализа угроз и мер защиты, представляет собой не только технически значимое решение, но и мощный ресурс для образовательной сферы. Современное образование в области информационной безопасности, киберзащиты, анализа угроз и искусственного интеллекта требует не только теоретической подготовки, но и активного применения знаний на практике. Традиционные лекции и семинарские занятия всё чаще расширяются интерактивными, проектно-ориентированными форматами, в которых подобные системы играют ключевую роль.

Использование Telegram-ботов, интегрированных с LLM и базами нормативных знаний, позволяет органично внедрить в образовательный процесс элементы симуляции, кейс-анализа, юридической практики и технических навыков. Такие технологии особенно актуальны в рамках цифровых кафедр, программ по подготовке ИБ-специалистов, магистратур по ИИ и учебных треков по DevSecOps.

В условиях постоянно меняющейся ИБ-реальности студенты, взаимодействуя с разработанным ботом, получают возможность:

1. изучать российские и международные стандарты (включая методику ФСТЭК, ISO/IEC 27001, ГОСТ Р 57580, NIST, OWASP) в интерактивно-прикладной форме, хорошо адаптированной под мобильные платформы;
2. проводить всесторонний анализ реальных и учебных сценариев киберугроз и мгновенно видеть рекомендуемые варианты реагирования и восстановления, что формирует целостное представление об инцидент-менеджменте;
3. использовать встроенные алгоритмы для генерации и оценки моделей угроз, классификации акторов и построения ответных стратегий в соответствии с типом системы (КИИ, ИСПДн, ГИС и другие);
4. развивать юридическую и нормативную грамотность в области информационной безопасности, переставая воспринимать нормативную базу как абстрактный теоретический материал, а видя её в привязке к конкретным ситуациям.

Таким образом, реализуемый в рамках проекта Telegram-бот становится не только полезным вспомогательным инструментом на уровне технических специалистов, но и полноценным элементом цифровой образовательной среды. В техникумах, вузах и системах ДПО подобные решения формируют у обучающихся устойчивое понимание взаимосвязи между анализом рисков, оптимальным реагированием на угрозы и текущими требованиями нормативных документов. Всё это помогает подготовить адаптивных, практико-ориентированных специалистов, способных действовать в реальных, быстро меняющихся условиях.

### 1.1.8 Проблематика масштабируемости и унификации

Одной из ключевых проблем, с которой сталкиваются специалисты по информационной безопасности, является масштабируемость используемых подходов и методик. В условиях активного роста и усложнения ИТ-инфраструктур, появления распределённых систем, гибридных облаков, удалённого доступа, а также жёстких регуляторных требований, становится очевидным: классические схемы, предполагающие жёсткую архитектуру и ручной труд высококвалифицированных специалистов, теряют свою эффективность.

Масштабируемость означает возможность адаптации системы под различные условия без кардинальной переработки её архитектуры — будь то переход от малой локальной сети к большой распределённой ИИС, или необходимость обслуживания мультисервисной инфраструктуры с разными уровнями классификации и регуляторного контроля. При этом система должна эффективно работать как с персональными и коммерческими данными, так и с государственными ИС и объектами критической информационной инфраструктуры (КИИ). В разрозненной системе координат это крайне сложно реализовать без унифицированных подходов.

Решение, предлагаемое в рамках данного проекта — использование Telegram-бота как фронтенда к интеллектуальной платформе анализа угроз на базе LLM и RAG — изначально проектируется с учётом этих вызовов. Его архитектура ориентирована на горизонтальное масштабирование и логическую федерализацию функций. Это достигается благодаря следующим факторам:

Во-первых, бот поддерживает работу с различными типами информационных систем и категорий защищаемых данных — через предварительно заданные параметры либо автоматическое распознавание контекста. Такая гибкость делает его применимым в самых разных отраслях: от образования и медицины до банковского сектора и телекоммуникаций.

Во-вторых, структура Telegram-бота позволяет адаптировать его под требования как конкретных заказчиков, так и государственных регуляторов. Заказчик может настраивать словари, источники знаний, блоки ответов и даже интерфейсные элементы в соответствии с корпоративной политикой ИБ и отраслевыми стандартами.

В-третьих, решение обладает гибкостью форматов входных и выходных данных. Это обеспечивает лёгкую интеграцию с другими системами защиты, платформами обучения, SIEM-решениями, CMDB, базами уязвимостей и даже внутренними ITIL-процессами. Таким образом, Telegram-бот становится частью экосистемы предприятия, а не замкнутым модулем.

Наконец, сама архитектура решения основана на модульности: модели LLM могут быть легко заменены, обновлены или кастомизированы в зависимости от задач, без необходимости корректировать остальную логику системы. Это открывает путь к использованию открытых решений (например, LLaMa, Mistral) в закрытых или ограниченных контурах без доступа к интернету, что особенно важно для госсектора или оборонных предприятий.

В результате платформа становится не просто техническим решением, а единым многоуровневым информационно-аналитическим слоем, способным расширяться по мере необходимости. Именно такого рода унификация и масштабируемость сегодня являются решающими факторами при выборе подходов к обеспечению ИБ в организациях, ориентированных на долгосрочную цифровую устойчивость.

## 1.2 Критический анализ существующих подходов к решению проблемы автоматизации анализа угроз и рисков в ИБ

В настоящее время для анализа угроз и рисков информационной безопасности (ИБ) используются следующие основные подходы:

1. Статические базы данных угроз;
2. Системы SIEM;
3. Решение на основе машинного обучения;
4. Чат-боты на базе LLM.

### 1.2.1 Статические базы данных угроз

1. БДУ ФСТЭК (Банк данных угроз безопасности информации ФСТЭК России)

Содержит матрицу способов реализации угроз, перечень негативных последствий, реестр угроз, перечень объектов, компонентов, нарушителей и мер защиты.

1. MITRE ATT&CK (Adversarial Tactics, Techniques & Common Knowledge)

База знаний компании MITRE, основанная на реальных наблюдениях. Содержит описание тактик, приёмов и методов, используемых киберпреступниками. Цель базы — составление структурированной матрицы используемых киберпреступниками приёмов, чтобы упростить задачу реагирования на киберинциденты [6].

1. NVD (National Vulnerability Database)

NVD - это база данных, созданная Национальным институтом стандартов и технологий (NIST) США. Она содержит информацию об известных уязвимостях в программном обеспечении и аппаратных средствах, включая описание уязвимости, её серьёзность, рекомендации по устранению и другие сведения [7].

Принцип работы статических баз данных угроз заключается в анализе информации об уязвимости и угрозах с помощью различных методов. Цель – выявить потенциальные риски и принять меры для их минимизации [8].

Структура базы данных:

Статическая база данных угроз может включать:

1. Описания угроз. Указываются источник угрозы (злоумышленник), объект воздействия и последствия воздействия (нарушение конфиденциальности, доступности, целостности);
2. Сведения об уязвимостях. Для каждой уязвимости записываются название, описание и тип ошибки;
3. Метрики опасности. Например, используется система оценки уязвимостей (CVSS), которая присваивает уровень опасности на основе определённых критериев.

Процесс пояска уязвимости:

Поиск уязвимостей в статической базе данных угроз осуществляется с помощью алгоритмов, которые:

1. Получают текстовые описания угроз и уязвимостей. Для каждой угрозы и уязвимости берутся их название и описание;
2. Обрабатывают текст. Схожие по смыслу слова и словосочетания заменяются синонимом, строка разбивается на отдельные слова и словосочетания;
3. Определяют ключевые слова. Для каждой угрозы и уязвимости составляется список ключевых слов из их описания;
4. Ищут совпадения по ключевым словам. Для заданной уязвимости определяются угрозы, множества ключевых слов которых пересекаются со множеством ключевых слов уязвимости.

Преимущества статической базы данных:

1. Раннее обнаружение уязвимостей

Банки угроз позволяют выявлять проблемы безопасности ещё до компиляции или развёртывания программного обеспечения, что снижает затраты на их исправление.

1. Полное покрытие кода

Анализируется весь исходный код, что помогает обнаружить уязвимости в тех частях приложения, которые редко или никогда не выполняются.

1. Автоматизация

Большинство инструментов для работы с базами угроз легко интегрируются в процесс непрерывной интеграции и доставки (CI/CD), что позволяет автоматизировать проверку кода на уязвимости.

1. Классификация угроз

Банки данных содержат сведения об основных угрозах и уязвимостях, которые характерны для определённых типов систем (например, автоматизированных систем управления, государственных информационных систем). Это помогает оценить необходимость и возможность реорганизации работы информационных систем в связи с обнаруженными угрозами.

Недостатки статической базы данных:

1. Ложные срабатывание

Инструменты, использующие статические базы данных угроз, могут выдавать ложные срабатывания — сообщения о проблемах, которых по факту нет. Это происходит из-за неправильно интерпретированных шаблонов, погрешностей при анализе сложных систем или влияния внешних факторов (например, используемых библиотек).

Ложные срабатывания приводят к тому, что реальные уязвимости могут быть пропущены, а ресурсы на проверку — потрачены впустую [9].

1. Ограниченная актуальность

Статические базы данных угроз не всегда актуальны, так как регулярно появляются новые угрозы и способы их реализации. Например, даже крупные базы (например, Банк угроз ФСТЭК России) содержат информацию об уязвимостях, обнаруженных несколько лет назад.

Это означает, что системы, использующие такие базы, могут не учитывать современные угрозы, что снижает эффективность защиты.

1. Сложность обновления

Обновление статических баз данных угроз может быть сложным, так как требует корректировки алгоритмов и данных. Например, при появлении новых типов угроз (динамических, с изменёнными индикаторами) система может не успеть обновить базу. Это приводит к тому, что защитные системы реагируют на угрозы уже после их реализации, что увеличивает риск ущерба.

Кейсы внедрения статической базы данных:

1. Устранение уязвимостей

БД угроз помогают определить, какие уязвимости операционных систем и прикладного ПО требуют закрытия, и разработать меры по их устранению. Например, на основе данных из БД создают изолированную подсеть ПК с экземплярами используемого ПО, чтобы сначала установить обновления на эти машины, а затем определить безопасность и допустимость их установки в масштабе организации.

1. Оценка надёжности программного решения

БД угроз позволяют рассмотреть статистические данные по обнаружению уязвимостей и производителям ПО, чтобы заранее оценить надёжность предполагаемого программного решения при разработке информационной системы.

1. Улучшение систем защиты

Анализ угроз из БД помогает улучшить меры безопасности, пересмотреть и усилить существующие меры защиты. Это может включать обновление программного обеспечения, внедрение новых технологий защиты или изменение политики безопасности [10].

Таблица 2 наглядно показывает все преимущества и недостатки такого метода.

Таблица 2 – Статические базы данных угроз

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Преимущества | Недостатки | Примеры инструментов |
| Покрытия угроз | Полный охват известных уязвимостей и шаблонов атак | Не учитывает новые 0 day угрозы | MITRE ATT&CK, NVD |
| Автоматизация | Автоматическое сканирование кода | Ложные срабатывания требуют ручной проверки | Solar Appscrener |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| Актуальность данных | Стандартизированные метрики (CVSS) | Задержка в обновлении (недели/месяцы) | Банк угроз ФСТЭК |
| Кейсы внедрения | Устранение уязвимостей в изолированных сетях | Высокие затраты на поддержку и обновления |  |

### 1.2.2 Системы SIEM (Security Information and Event Management)

1. Splunk

Решение корпоративного уровня для надёжного сбора данных и мониторинга в реальном времени. Среди преимуществ Splunk называют масштабируемость, широкие возможности настройки и интеграции с многочисленными сторонними приложениями и инструментами. К недостаткам относят высокую стоимость, особенно при больших развёртываниях, и сложность для новичков.

1. IBM QRadar

Инструмент на основе искусственного интеллекта, который помогает быстрее выявлять, анализировать и реагировать на инциденты безопасности. QRadar известен расширенными возможностями аналитики и корреляции, удобным интерфейсом и интеграцией с широким спектром решений безопасности от IBM. К минусам относят высокую стоимость и потребность в значительных системных ресурсах.

1. Microsoft Sentinel

Облачное решение, которое интегрируется с Azure и предлагает оперативную информацию об угрозах. Для обнаружения потенциальных угроз используется искусственный интеллект. К недостаткам относят ограниченную поддержку локальных развертываний и зависимость от Azure.

Принцип работы SIEM – включает в себя следующие аспекты, которые представлены на рисунке 4:

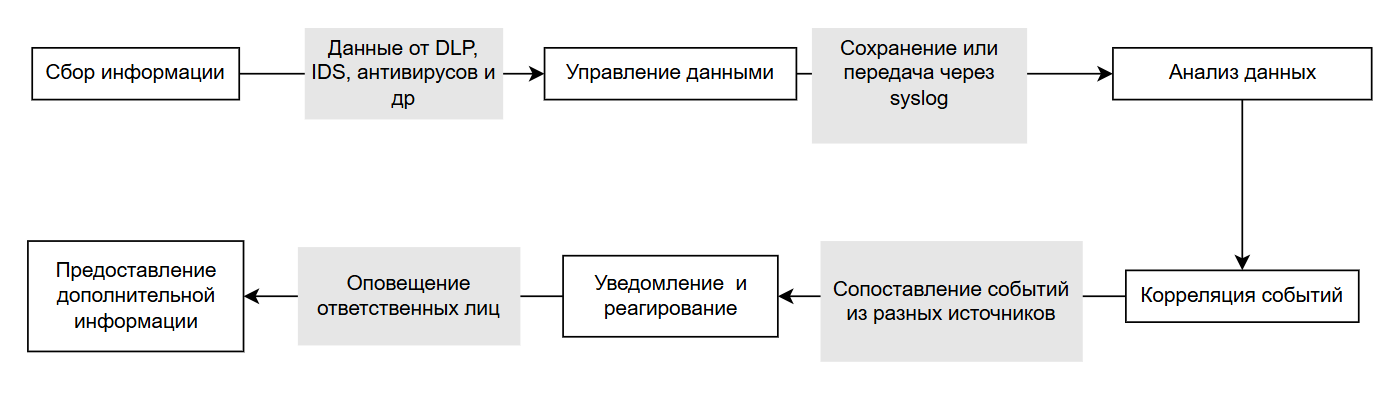


Рисунок 4 – Принципы работы SIEM

1. Сбор информации. Данные собираются из различных источников DLP, IDS, антивирусы, межсетевые экраны и т.д;
2. Управление данными. Данные могут сохраняться пассивно или передаваться на серверы по протоколы syslog, когда источник сам передаёт информацию;
3. Анализ данных. Работа с данными идет на трех уровнях: сбор, управление и анализ;
4. Корреляция событий. Разрозненные события из разных логов сопоставляются между собой, объединяются, если они относятся к одной и той же активности или разным этапам активности [11];
5. Уведомление и реагирование. Система отправляет уведомление ответственным лицам о выявленных подозрительных событиях в журналах;
6. Предоставление дополнительной информации. Система предоставляет дополнительную информацию по каждому их событий и устройств в сети.

Преимущества системы SIEM:

1. Централизованный сбор данных

SIEM-системы собирают данные о событиях безопасности из различных источников, таких как журналы событий, данные сетевой активности, информация о пользователях и приложениях. Это позволяет:

1. Обеспечить полное представление о происходящих изменениях в информационной системе;
2. Упростить обработку данных — система преобразует их в стандартизированный формат для удобства анализа.
3. Анализ и корреляция событий

SIEM-системы анализируют собранные данные, чтобы выявить связи между событиями, которые могут указывать на угрозы. Это позволяет:

1. Обнаруживать аномалии — отклонения от нормальной работы системы (например, несанкционированный доступ, необычные попытки входа в систему);
2. Выявлять сложные шаблоны атак, которые могут остаться незамеченными отдельными системами безопасности.
3. Управление инцидентами

SIEM-системы помогают управлять инцидентами, которые включают подозрительные события, в соответствии с установленными правилами. Это включает несколько этапов:

1. Обнаружение инцидента — система фиксирует события, соответствующие правилам корреляции или аномальному поведению;
2. Классификация и приоритизация инцидента — SIEM определяет тип угрозы, её источник, затронутые системы и пользователей;
3. Уведомление специалистов — система отправляет уведомления команде безопасности через электронную почту, мессенджеры или интеграцию с системами тикетов;
4. Расследование инцидента — специалисты анализируют детали инцидента, используя инструменты SIEM для просмотра логов, временных меток, сетевых соединений и активности пользователей;
5. Реагирование на инцидент — принимаются меры по нейтрализации угрозы и предотвращению дальнейшего распространения атаки.
6. Соответствие требованиям

SIEM-системы помогают организациям соответствовать требованиям законодательства и стандартам безопасности (например, GDPR, PCI DSS, HIPAA). Это достигается за счёт:

1. Автоматизации отчётности — система формирует журналы проверок и хранит логи в течение требуемого времени;
2. Формирования отчётов — SIEM генерирует документы для внутренних целей и контролирующих органов.

Недостатки системы SIEM:

1. Технические проблемы
2. Сложность обработки больших объёмов данных. Для этого требуются мощные серверы и большое хранилище данных;
3. Ложные срабатывания. Если система не настроена должным образом, она может генерировать много ложных уведомлений, что затрудняет выявление реальных угроз;
4. Недостаточная масштабируемость. Правила корреляции, заложенные в систему, могут быть направлены на выявление конкретных угроз, но не учитывать возможные их комбинации.
5. Проблемы с внедрением
6. Сложность интеграции. SIEM-систему сложно вписать в существующую сеть защитных механизмов;
7. Недостаток квалифицированных кадров. Найти и удержать специалистов, которые могут настраивать систему, анализировать результаты и реагировать на инциденты, может быть трудно;
8. Высокие затраты на внедрение и поддержку. Инвестиции в SIEM включают покупку лицензий, оборудования и расходы на техническую поддержку

Примеры:

История компании, которая на протяжении 2 лет подвергалась атакам через одну и ту же уязвимость в Microsoft Exchange. При первом взломе злоумышленники зашифровали данные, но компания восстановилась из уже зараженного бэкапа. Через год хакеры просканировали инфраструктуру, а еще через полгода произошла третья атака с использованием программы-шифровальщика, которая в итоге парализовала работу компании.

Также, примером является – случай из опыта компании Softline. Заказчик не смог выбрать конкретный продукт, который бы максимально отвечал требованиям компании из-за спецификации ведения бизнеса. В итоге победил поставщик, предложивший простую конфигурацию решения, которая не соответствовала большинству фактических потребностей клиента.

1. Человеческие факторы
2. Дефицит квалифицированных специалистов. Найти и удержать таких сотрудников, которые могут настраивать систему, анализировать результаты и реагировать на инциденты, может быть трудно;
3. Неправильная настройка. Если правила и фильтры недостаточно чувствительны, система может пропускать реальные события или генерировать оповещения, когда их не происходит;
4. Проблема масштабирования: "Магнит" столкнулся с 300% ростом объема логов после перехода на облако, что сделало SIEM экономически неэффективным.

Кейсы внедрения системы SIEM:

1. Модернизация нахождения инцидентов

Альфа-Банк В 2022 году модернизировал систему выявления и обработки инцидентов кибербезопасности (SIEM), а также усовершенствовал средства автоматизации внутренних процессов работы с киберинцидентами. В результате проект стал победителем Национальной банковской премии – 2022 в номинации "Информационная безопасность" [12]

1. Экономическая выгодна на расследованиях

X5 Group Внедрили IBM QRadar. Финансовый эффект: экономия $240К/год на расследованиях, но затраты на кастомизацию составили $1.2M (Gartner Case Study, 2024).

1. Минимизация последствий атак с использованием вредоносного ПО

В ритейловой сети SIEM помогла минимизировать последствия атак: при обнаружении заражённого устройства система автоматически изолировала его от сети, предотвращая распространение угрозы.

1. Сокращение времени расследования инцидентов связанных с доступом к конфиденциальным данным пациентов

SIEM оперативно выявляла подозрительные попытки входа и предотвращала утечки информации.

Таблица 3 наглядно показывает преимущества и недостатки и системы SIEM по четырем критериям: централизованный сбор, корреляция событий, соответствие стандартам, реагирование, а также примеры инструментов внедрения.

Таблица 3 – SIEM-системы

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Преимущества | Недостатки | Примеры инструментов |
| Централизованный сбор | Объединение данных из DLP, IDS, фаерволов | Сложность обработки объемных логов | Альфа-Банк  (2022) |
| Корреляция событий | Выявление сложных атак через анализ паттернов | Ложные срабатывания при неправильной настройке | X5 Group  (IBM QRadar) |
| Соответствие стандартам | Автоматическая отчетность для GDPR, PCI DSS | Требует дорогостоящей кастомизации |  |
| Реагирование | Автоматическая изоляция зараженных устройств | Дефицит квалифицированных специалистов для настройки |  |

### 1.2.3 Решение на основе машинного обучения

1. Darktrace

Глобальный лидер в сфере кибербезопасности. Платформа использует алгоритмы машинного обучения для мониторинга сетевой активности, обнаружения аномалий и выявления возникающих угроз в реальном времени. Darktrace подходит организациям разных размеров и отраслей, которые хотят усилить защиту от киберугроз [13].

1. Palo Alto Cortex XDR

Платформа безопасности на базе AI, предназначенная для обнаружения, расследования и реагирования на киберугрозы в ИТ-инфраструктуре организации. Она обеспечивает централизованное решение безопасности, интегрируя защиту конечных точек, сетевую безопасность, облачную безопасность и поведенческую аналитику в одну систему. Cortex XDR помогает организациям обнаруживать, анализировать и реагировать на сложные киберугрозы [14].

1. Vectra AI

Платформа кибербезопасности, специализирующаяся на обнаружении угроз в реальном времени и реагировании на них. Она фокусируется на выявлении вредоносного поведения в корпоративных средах. Vectra AI поддерживает веб, Windows и Linux [15].

Принцип работы решений на основе машинного обучения (ML) представлен на рисунке 5, и заключается в создании алгоритмов, которые автоматически извлекают знания из данных и решают задачи или предсказывают результаты на их основе. Вместо того, чтобы следовать строго запрограммированным инструкциям, алгоритмы учатся выявлять закономерности, прогнозировать результаты и адаптироваться к изменениям [16].

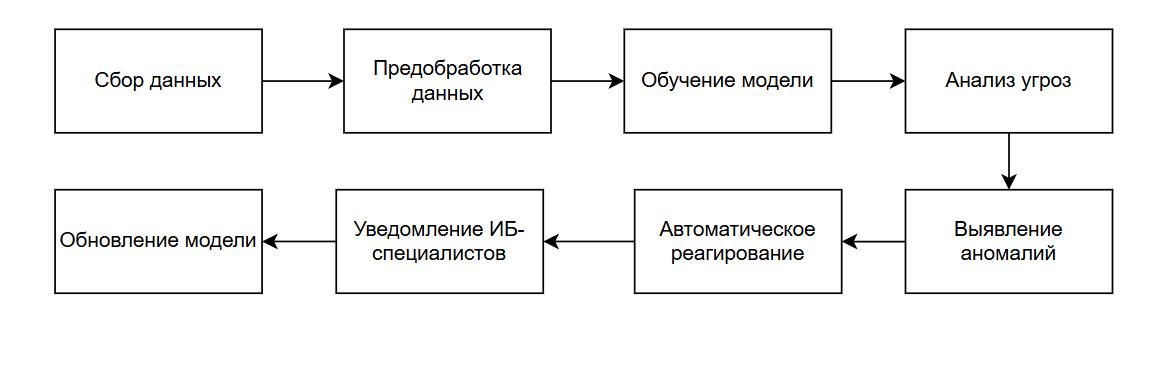


Рисунок 5 – Принцип работы решений на основе машинного обучения

Основная идея ML – научить компьютерную систему находить закономерности, делать выводы и прогнозы, решать задачи и улучшать результаты на основе опыта, полученного из данных.

Преимущества решений на основе машинного обучения:

1. Автоматизация задач
2. Освобождение времени и ресурсов. Алгоритмы MLвыполняют рутинные операции, что позволяет сотрудникам сосредоточиться на более сложных задачах;
3. Ускорение процессов. Системы на базе ML обрабатывают данные и принимают решения быстрее человека, что важно в задачах, где важна оперативность [17].
4. Точность и надежность
5. Анализ больших объемов данных. Алгоритмы ML выявляют закономерности, которые часто невидимы для человека, и составляют более точные прогнозы;
6. Проверка данных. Перед обучением моделей данные проходят проверку, что устраняет ошибки и предвзятость, повышает точность прогнозов [18].
7. Масштабируемость
8. Адаптивность к растущему объему работы. Системы ML могут масштабироваться без потери качества, обрабатывая большие наборы данных, число пользователей или сложность вычислений;
9. Использование облачных вычислительных платформ. Это позволяет динамически распределить ресурсы по мере необходимости, что повышает масштабируемость.

Недостатки решений на основе машинного обучения:

1. Зависимость от данных

Эффективность моделей ML сильно зависит от качества и количества обучающих данных. Некачественные или предвзятые данные могут привести к неточным прогнозам или несправедливым результатам.

Также существует проблема *концептуального дрейфа* – изменения характера данных со временем, из-за которых модели становятся менее точными без регулярного переобучения.

1. Вычислительные затраты

Разработка и внедрение моделей ML требует значительных инвестиций в инфраструктуру и вычислительных ресурсов. Для малого бизнеса эти затраты могут оказаться непомерно высокими.

Кроме того, важно учитывать расходы на подготовку данных – чем грамотнее собрана и обработана информация, тем быстрее и качественнее можно реализовать решение.

1. Сложность интерпретации результатов

Многие модели ML, особенно системы глубокого обучения, работают как "черные ящики". Процессы принятия решений в них трудно интерпретировать или объяснить. Это создает трудности как для разработчиков, так и для пользователей, которые нуждаются в объяснении.

Также отсутствие "прозрачности" в алгоритмах может вызвать недовольство и недоверие, особенно в критически важных областях, таких как медицина и финансы.

1. Другие примеры

Система диагностики на основе ML, созданная для выполнения рака кожи по фотографиям пациентов, может поставить неверный диагноз, если связь между цветом кожи и диагнозом задана неточно.

Алгоритм оценки кредитоспособности, обученный на базе данных определенной социально-экономической группы, может дискриминировать некоторых потенциальных заемщиков.

Кейсы внедрения на основе машинного обучения:

1. Кластеризация тезисов

Проект "Авито". Компания разработала большую языковую модель (LLM) для анализа отзывов о мастерах. Алгоритм обрабатывает мнения пользователей, выявляет повторяющиеся тезисы и на из основе формируем обобщенный отзыв с разделами "Преимущества" и "Недостатки".

1. Технология биометрической оплаты взглядом

В сети "Перекрёсток" решили использовать алгоритмы компьютерного зрения для безопасной идентификации биометрическими кассами, время оплаты на кассе сократилось в 2-2,5 раза.

1. Прогнозирование качество продукта

Система предикативной аналитики "АтомМайнд" в "Росатоме" прогнозирует качество продукции и состояние оборудования, отслеживая более 2 млн технологических параметров. В результате расходы на техобслуживание сократились на 30%, уровень брака снизился с 2,3% до 0,9%.

1. Борьба с мошенничеством в финансовом секторе

ML-системы обнаруживают подозрительные транзакции, анализируя поведения пользователей и паттерны операция [19].

Таблица 4 наглядно показывает преимущества и недостатки решений на основе машинного обучения.

Таблица 4 – Решение на основе машинного обучения

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Преимущества | Недостатки | Примеры инструментов |
| Автоматизация | Обнаружение аномалий без предзаданных правил | Зависимость от качества данных (например, ошибки в диагностике рака кожи) | Ростатом  («АтомМайнд) |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Масштабируе-мость | Обработка Big Data в реальном времени | Высокие вычислительные затраты | «Перектресток»  (оплата взглядом) |
| Интерпретируе-мость | Выявление скрытых угроз (например, инсайдерские атаки) | «черный ящик»: сложность объяснения решений |  |
| Гибкость | Адаптация к новым угрозам (например, AI-генерация вредоносного кода | Риск дискриминации в кредитных алгоритмах |  |

### 1.2.4 Чат-боты на базе LLM

1. GPT-4

Позволяет создавать пользовательские версии, которые фокусируются на конкретных задачах. У модели большой набор плагинов и расширений, большинство из которых производятся третьими сторонами. GPT-4 может создавать изображения «на лету».

1. Gemini

По заявлению Google, в большинстве сценариев превосходит GPT-4, давая более точные и развёрнутые ответы по запросу пользователя через чат-бот. Модель способна работать с разными типами информации, включая текст, изображения, аудио и видео. Она также обладает продвинутыми навыками в математике и программировании, что отличает её от GPT-4.

Принцип работ чат-ботов на базе LLM

В основе работы LLM лежит трансформерная архитектура — набор нейронных сетей, которые обрабатывают входные данные параллельно.

Ключевые этапы работы показаны на рисунке 6.

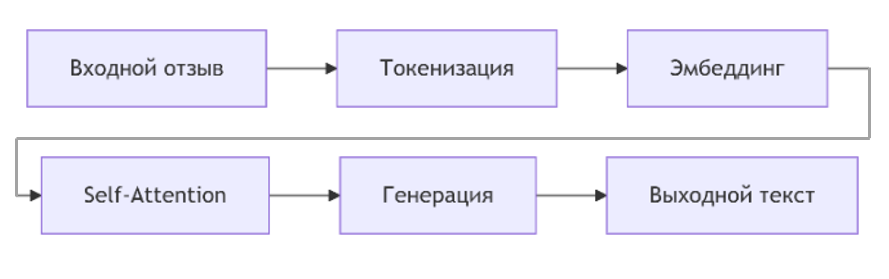


Рисунок 6 – Ключевые этапы работы чат-бота на базе LLM

1. Токенизация

Текст разбивается на части (токены) - небольшие фрагменты, с которыми модель может работать. Токеном может быть слово, часть слова, символ или знак препинания. Например, запрос «Привет, как дела?» превращается в токены: [«привет», «», «как», «дела». Токенизация позволяет системе оперировать текстом как набором базовых элементов, что важно для сложных языков, где значение может зависеть от контекста или структуры фразы [20].

1. Эмбеддинг

Эмбеддинги — это числовые векторы, которые представляют объекты (слова, предложения, изображения и другие данные) и помогают нейронным сетям анализировать их взаимосвязи. Эти векторы позволяют моделям машинного обучения эффективно работать с различными типами данных, улучшая обработку текста, изображений, аудио и видео.

Эмбеддинг нужен для:

1. Обработки стиля текста. Эмбеддинги позволяют моделям различать стиль текста, например, анализировать, является ли текст формальным или неформальным. Это важно при работе с отзывами, статьями или рекламными материалами, где необходимо учитывать эмоции или тональность. Например, эмбеддинги помогут определить, является ли текст положительным, нейтральным или негативным;
2. Поиска информации и ее сравнения. Это помогает моделям находить тексты, изображения или другие данные, схожие по смыслу. Например, в поисковых системах эмбеддинги позволяют улучшить результаты поиска, показывая страницы или товары, которые могут быть интересны пользователю, даже если они не содержат точных совпадений с запросом;
3. Для анализа данных. В аналитических системах эмбеддинги помогают извлекать важную информацию из больших массивов данных, а также выявлять закономерности и тренды. Например, в бизнес-аналитике эмбеддинги могут помочь выявить потребительские предпочтения на основе анализа данных о покупках;
4. Перевода текста и его генерации. Эмбеддинги также используются в генеративных моделях, таких как GPT, для создания осмысленных текстов. Например, когда модель генерирует ответ на запрос, она использует эмбеддинги для преобразования слов в контекстные векторы, что позволяет создавать логичные и последовательные ответы, соответствующие запросу пользователя [21].
5. Самовнимание

Механизм самовнимания (self-attention) позволяет модели фокусироваться на различных частях входных данных при определении смысла всего предложения или абзаца. Это значит, что модель может «обращать внимание» на релевантные части текста, игнорируя менее важные.

Например, модель определяет, что выражение «Вася подарил Маше цветы» отличается от «Маша подарила Васе цветы».

1. Генерация

На основе обработанных данных LLM создаёт новые токены, формирующие ответ. Процесс генерации включает подбор наиболее релевантных слов, учёт грамматики и адаптацию под стиль запроса.

Например, запрос «Привет, как дела?» преобразуется в ответ: [«Хорошо», «», «спасибо», «!».]

Преимущества ботов на базе LLM:

1. Гибкость в понимании запросов. LLM способны интерпретировать сложные и неоднозначные вопросы, адаптироваться к различным формам общения;
2. Обучаемость на ходу. Модели могут обучаться в процессе взаимодействия, улучшая свои ответы со временем;
3. Контекстуальное понимание. LLM запоминают и учитывают предыдущие реплики в разговоре, что делает общение более естественным и связным;
4. Многоязычная поддержка. LLM могут работать с несколькими языками, что упрощает создание многоязычных виртуальных помощников.

Недостатки ботов на базе LLM:

1. Технические ограничения
2. Недостоверность ответов. LLM могут выдавать информацию, которая выглядит правдоподобно, но на самом деле ложна или выдумана («AI-галлюцинация»);
3. Проблемы с прозрачностью. Сложная структура и огромный объём данных, на которых обучаются LLM, затрудняют понимание, почему модель дала тот или иной ответ;
4. Ограниченный объём контекста. Размер контекста, который модель может обрабатывать одновременно, мал для решения большинства практических задач;
5. Нестабильность. Если модель натыкается на редкое или необычное сочетание токенов, которые не встречались при обучении, это может вызвать неожиданное поведение в генерации последующих токенов.
6. Этические проблемы
7. Предвзятость алгоритмов. Создатели LLM иногда закладывают в модели свои ценности и предубеждения, что может сделать модель необъективной;
8. Угроза конфиденциальности. Личная информация, которая используется для обучения модели, может быть доступна ей в качестве выходных данных;
9. Риски злоупотребления. LLM могут использоваться для создания дезинформации, спама и вредоносного контента.

Применение LLM в чат-ботах:

Некоторые области применения LLM в чат-ботах:

1. Автоматизация обслуживания клиентов и техподдержки. Чат-боты на базе LLM обрабатывают запросы пользователей, решают их проблемы и дают рекомендации без вмешательства человека;
2. Финансовые консультации. Чат-боты предоставляют информацию о продуктах, рассчитывают кредиты или страховые премии, предлагают персонализированные решения на основе данных пользователя;
3. Электронная коммерция. Чат-боты помогают пользователям с выбором товаров, дают рекомендации на основе их предыдущих покупок или интересов, проводят транзакции;
4. Образование и обучение. LLM используются для создания виртуальных репетиторов и помощников, которые помогают студентам освоить сложные темы, решать задания и отвечать на вопросы [22].

Примеры LLM чат-ботов:

1. Ask Delta. Чат-бот американской авиакомпании Delta на базе LLM помогает клиентам с регистрацией на рейс, отслеживанием багажа и поиском рейсов;
2. Heliograf. Решение для издания Washington Post, которое автоматизирует создание контента: помогает журналистам в написании черновиков, подбирает креативные заголовки и делает выжимки из релевантной информации;
3. ChatGPT от OpenAI. Применяется для генерации текстов, ответов на вопросы, помощи в программировании и других задач;
4. GigaChat от «Сбера». Российский продукт, который поддерживает более 100 языков и ориентирован на текстовые и мультимодальные запросы;
5. Claude от Anthropic. Модель для работы с большими объёмами текста, обработки конфиденциальной информации и сложных текстовых анализов.

Таблица 5 наглядно показывает преимущества и недостатки LLM.

Таблица 5 – Чат-боты на базе LLM

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Преимущества | Недостатки | Примеры инструментов |
| Гибкость | Понимание естественного языка (NLP) | Галлюцинации: генерация ложной информации | Delta Airlines (Ask Delta) |
| Многоязычность | Поддержка 100+ языков (GigaChat) | Ограниченный контекст (например, 128К токенов у GPT-4) | Washington Post |
| Автоматизация | Круглосуточная поддержка клиентов | Юридическая ненадежность (невалидные ответы) |  |
| Интеграция | API для выстраивания в CRM, образовательные платформы | Риски утечек конфиденциальных данных |  |

Таблица 6 содержит все данные для удобного сравнения и выбора необходимого решения.

Таблица 6 – Общая сводная таблица

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий сравнения | Автома-тизация | Актуаль-ность данных | Затраты на внедрение | Юридичес-кая валидность | Пример инструмента |
| LLM-боты  (1.2.4) | Максимальная (NLP) | Низкая (галлюцинации) | Средние (API/лицензии) | Очень низкая | GPT-4 |
| ML-решения (1.2.3) | Очень высокая (аномалии) | Высокая (обучение на новых данных) | Высокие (инфраструктура) | Низкая  (черный ящик) | Darktrace |
| SIEM (1.2.2) | Высокая (корреляция событий) | Средняя (зависит от логов) | Очень высокие | Средняя (ручная проверка) | IBM QRadar |
| Статические базы (1.2.1) | Средняя (CI/CD) | Низкая (задержки) | Низкие  (готовые базы) | Высокая (стандарты CVSS) | MITRE ATT&CK |

### 1.2.5 Ключевые выводы

Анализ таблицы показывает, что каждый из рассмотренных подходов к кибербезопасности имеет свои уникальные преимущества и ограничения. Статические базы данных, такие как MITRE ATT&CK, отличаются высокой юридической валидностью и низкой стоимостью внедрения, но их недостатком является задержка в обновлении информации. SIEM-системы (например, IBM QRadar) обеспечивают высокий уровень автоматизации за счет корреляции событий, однако требуют значительных финансовых вложений.

ML-решения, представленные такими инструментами как Darktrace, демонстрируют превосходную способность выявлять аномалии благодаря обучению на новых данных, но их "чёрный ящик" усложняет процесс юридического обоснования. LLM-боты, такие как GPT-4, достигают максимальной автоматизации за счёт обработки естественного языка, однако их применение ограничено из-за склонности к "галлюцинациям" и низкой применимости в регулируемых отраслях.

Таким образом, наиболее эффективной стратегией представляется комбинированный подход. Для базовой защиты оптимально сочетание статических баз с LLM-ботами, что обеспечит баланс между стоимостью и функциональностью. В корпоративном секторе, где критичны масштабируемость и глубокая аналитика, предпочтительна интеграция SIEM с ML-решениями, дополненная статическими базами для соблюдения нормативных требований. Такой гибридный подход позволяет максимально использовать сильные стороны каждой технологии, минимизируя их недостатки.

## 1.3 Проектирование интеллектуального Telegram-бота с LLM RAG для анализа киберугроз с интеграцией базы ФСТЭК: обоснование цели и задач, технические требования

В условиях стремительного роста киберугроз и ужесточения регуляторных требований в области информационной безопасности (ИБ) организации сталкиваются с необходимостью оперативного анализа рисков и их соответствия нормативно-правовой базе. Особую сложность представляет обработка актуальных угроз из официальных источников, таких как база данных ФСТЭК России, и их интерпретация с учетом законодательных актов (187-ФЗ, 152-ФЗ, приказы ФСТЭК).

Существующие решения зачастую ограничены ручным поиском и анализом, что требует значительных временных затрат и не гарантирует полноты оценки. Внедрение интеллектуального бота на основе технологии RAG и крупных языковых моделей позволит автоматизировать процесс: от классификации угроз по критериям ФСТЭК до генерации обоснованных правовых заключений.

Данный подраздел раскрывает:

1. Цель разработки — создание инструмента для мгновенного анализа угроз ИБ с привязкой к регуляторным требованиям;
2. Ключевые задачи, включая интеграцию с отечественными LLM и векторной базой данных угроз ФСТЭК;
3. Технические требования к архитектуре и функционалу.

Проект направлен на повышение эффективности работы специалистов ИБ за счет сокращения времени реагирования и минимизации человеческого фактора в критически важных оценках.

### 1.3.1 Обоснование цели и задач работы

Цель работы — оптимизация анализа угроз и рисков в информационной безопасности за счёт автоматизации обработки данных из базы ФСТЭК с использованием LLM и RAG-технологий.

Использование LLM и RAG для анализа базы данных ФСТЭК позволит: ускорить и автоматизировать анализ угроз, сократив затраты времени и ресурсов; повысить эффективность обработки информации за счет автоматической генерации обоснованных выводов и рекомендаций; быстро адаптироваться к новым угрозам за счет получения актуальной информации в режиме реального времени; улучшить процесс принятия решений за счет предоставления специалистам более обоснованных данных; сократить расходы на анализ, обучение и расширение штата; повысить устойчивость к киберугрозам за счет более точного и быстрого выявления уязвимостей и мер защиты.

Задачи:

1. провести анализ проблем ручного анализа угроз ИБ, включая оценку временных затрат и рисков, на основе статистических данных и нормативных требований (ФСТЭК, ГОСТ Р 57580);
2. исследовать возможности применения технологий LLM и архитектуры RAG для автоматизированной обработки запросов и интеграции с базой угроз ФСТЭК;
3. создать прототип функционального Telegram-бота с возможностью вывода справочной информации по ИБ-угрозам и разработать архитектурное решение для подключения LLM, RAG и базы данных ФСТЭК;
4. проанализировать структуру и форматы базы угроз ФСТЭК и разработать методы их интеграции;
5. оценить эффективность предложенного решения: сокращение времени анализа угроз, снижение оперативной нагрузки на специалистов по информационной безопасности.

Обоснование задач:

1. Необходимо количественно и качественно оценить существующие проблемы ручного анализа. Без этого невозможно понять, насколько целесообразно и эффективно внедрение автоматизированного решения;
2. Необходимо определить, насколько LLM и RAG подходят для решения задачи. Не все технологии одинаково хорошо справляются с разными типами данных и задач;
3. Создание прототипа позволяет проверить работоспособность предложенного решения на практике. Telegram-бот выбран как удобный интерфейс для взаимодействия с пользователями (специалистами по информационной безопасности);
4. Необходимо понять, как организованы данные в базе ФСТЭК, чтобы правильно их обработать и интегрировать с LLM/RAG. Без понимания структуры данных невозможно извлечь из них ценную информацию;
5. Необходимо измерить, насколько улучшился процесс анализа угроз после внедрения автоматизированного решения. Это позволит оценить реальную пользу от использования LLM/RAG;

В совокупности эти задачи позволяют комплексно подойти к решению проблемы ручного анализа угроз информационной безопасности, начиная с анализа существующих проблем и заканчивая оценкой эффективности предложенного решения.

### 1.3.2 Основное назначение интеллектуального бота

Telegram бот для анализа киберугроз с интеграцией базы ФСТЭК будет выполнять следующие действия, используя подход RAG:

Что он будет делать:

1. Прием запроса от пользователя: пользователь задает вопрос или предоставляет описание киберугрозы (например, «Какие уязвимости CVE-2023-1234 связаны с DDoS-атаками на веб-серверы?»);
2. Извлечение релевантной информации (Retrieval):
   1. Поиск в базе ФСТЭК: Telegram-бот использует поисковый механизм (например, векторный поиск или полнотекстовый поиск) для извлечения из базы ФСТЭК информации, относящейся к запросу пользователя. Это могут быть описания уязвимостей, записи об инцидентах, методы защиты и т. д., наиболее релевантные запросу;
   2. Поиск во внешних источниках (опционально): для получения самой актуальной информации Telegram-бот также может искать релевантные данные во внешних источниках, таких как базы данных CVE, блоги по безопасности, отчеты об угрозах. (Это часть «дополненной» генерации).
3. Агрегация и контекстуализация: Telegram-бот собирает извлеченную информацию и формирует контекст для LLM. Это может включать в себя объединение описаний уязвимостей, связанных с ними инцидентов и рекомендованных мер защиты;
4. Генерация ответа (Generation):
   1. Предоставление контекста LLM: LLM получает запрос пользователя и сформированный контекст из извлеченных данных;
   2. Генерация ответа: LLM, опираясь на контекст и свои знания, генерирует развернутый ответ, который содержит:
      1. Описание угрозы;
      2. Список связанных уязвимостей;
      3. Рекомендации по устранению или смягчению последствий;
      4. Ссылки на источники информации (в базе ФСТЭК или во внешних источниках).

5) Предоставление ответа пользователю: Telegram-бот отправляет сгенерированный ответ пользователю, предоставляя четкую и полезную информацию об угрозе.

Зачем он это будет делать:

1. Ускорение анализа: автоматизация процесса поиска и анализа информации об угрозах позволяет специалистам по информационной безопасности получать ответы на свои вопросы гораздо быстрее, чем при ручном поиске;
2. Повышение точности: Telegram-бот предоставляет LLM контекст в виде релевантных данных из базы ФСТЭК, что позволяет LLM генерировать более точные и обоснованные ответы;
3. Обеспечение актуальности: благодаря интеграции с внешними источниками Telegram-бот может предоставлять информацию об угрозах, которые еще не были включены в базу ФСТЭК;
4. Снижение нагрузки на специалистов: Telegram-бот берет на себя рутинную работу по поиску и анализу информации, позволяя специалистам сосредоточиться на более сложных задачах;
5. Снижение риска ошибок: автоматизация процесса снижает вероятность человеческой ошибки при анализе информации;
6. Предоставление доступа к знаниям: Telegram-бот делает знания, содержащиеся в базе ФСТЭК, более доступными и удобными для использования.

В целом, Telegram-бот предназначен для того, чтобы предоставить специалистам по информационной безопасности быстрый, точный и актуальный доступ к информации об угрозах, что поможет им принимать более обоснованные решения и эффективно защищать организации от киберугроз. Интеграция с базой ФСТЭК обеспечивает надежный и проверенный источник информации, а RAG позволяет использовать эту информацию в сочетании с возможностями LLM для генерации полезных и действенных ответов.

### 1.3.3 Обоснование разработки интеллектуального Telegram-бота для анализа киберугроз с интеграцией базы ФСТЭК

Потребность в интеллектуальном поиске: современные пользователи сталкиваются с огромным объемом информации, особенно в академической и научной сферах. Традиционные методы поиска часто оказываются недостаточно эффективными из-за низкой релевантности результатов или отсутствия агрегации данных из разнородных источников. Разработка бота на базе LLM и RAG позволит решить эту проблему, обеспечивая высокоточный и контекстно-зависимый поиск.

Использование передовых технологий: интеграция LLM и RAG обеспечит глубокое понимание запросов пользователей, а также возможность извлекать и анализировать информацию из научных статей, журналов, книг и других учебных материалов. Это значительно повысит качество ответов и сократит время на поиск нужных данных.

Удобство и доступность: внедрение бота в популярные мессенджеры и социальные сети сделает его доступным для широкого круга пользователей, включая студентов, исследователей и преподавателей. Это позволит им получать информацию в режиме реального времени, не переключаясь между множеством платформ.

Ранжирование и достоверность источников: разработка алгоритмов для оценки релевантности и авторитетности источников обеспечит пользователей надежными и проверенными данными, что особенно важно в академической и научной среде.

Автоматизация и масштабируемость: бот сможет обрабатывать большое количество запросов одновременно, что делает решение масштабируемым и пригодным для использования в образовательных учреждениях, исследовательских центрах и других организациях.

Снижение нагрузки на пользователей: бот избавит пользователей от необходимости вручную фильтровать и анализировать множество источников, предоставляя готовые, точные и структурированные ответы.

### 1.3.4 Требования к Telegram-боту

Программный комплекс спроектирован как высокоуровневая система, состоящая из трех ключевых модулей, которые в тесном взаимодействии обеспечивают полный цикл обработки пользовательских запросов и предоставления информативных ответов. Каждый модуль выполняет строго определенные функции, что в совокупности создает эффективный инструмент для работы специалистов по информационной безопасности:

1) Модуль обработки и анализа запросов (LLM-ядро системы)

Функциональное назначение

Данный модуль представляет собой интеллектуальное ядро системы, отвечающее за глубокую обработку и осмысление поступающих пользовательских запросов. Его работа строится на современных технологиях обработки естественного языка (NLP) с использованием передовых языковых моделей.

Ключевые функции

1. Осуществляет первичный анализ и нормализацию входящих текстовых запросов;
2. Выполняет сложную лингвистическую обработку, включая токенизацию и семантический разбор;
3. Преобразует текстовые запросы в высокоразмерные векторные представления (эмбеддинги);
4. Определяет интенты пользователя и ключевые сущности в запросе;
5. Формирует оптимизированные поисковые запросы для последующей обработки.

2) Модуль интеллектуального поиска (RAG-движок)

Функциональное назначение

Этот модуль отвечает за эффективный поиск и извлечение релевантной информации из обширной нормативной базы. В его основе лежит современная технология Retrieval-Augmented Generation, обеспечивающая высокую точность поиска даже в условиях неполных или нечетких запросов.

Ключевые функции

1. Осуществляет семантический поиск в векторной базе знаний;
2. Работает с предварительно подготовленными эмбеддингами нормативных документов;
3. Поддерживает несколько стратегий поиска (точное соответствие, семантическая близость, контекстный поиск);
4. Обеспечивает ранжирование результатов по степени релевантности;
5. Поддерживает механизм расширения запросов для повышения полноты поиска.

Пример практического применения

Для запроса о рисках облачных хранилищ модуль:

1. Находит соответствующие разделы в документах ФСТЭК;
2. Извлекает упоминания о типовых угрозах (утечки данных, несанкционированный доступ);
3. Возвращает нормативные ссылки и рекомендуемые меры защиты.

3) Модуль пользовательского взаимодействия (Telegram-интерфейс)

Функциональное назначение

Данный модуль обеспечивает удобный и интуитивно понятный интерфейс для взаимодействия пользователей с системой. Реализованный в виде Telegram-бота, он предоставляет унифицированный способ доступа к функциональности системы для специалистов различного уровня подготовки.

Ключевые функции

1. Обеспечивает ввод текстовых запросов в свободной форме;
2. Поддерживает интерактивные элементы управления (кнопки, меню, быстрые команды);
3. Реализует удобное форматирование выходных данных;
4. Предоставляет механизмы навигации по полученным результатам;
5. Ведет историю взаимодействий для каждого пользователя;
6. Поддерживает мультимедийные форматы представления информации.

Пример практического применения

При получении запроса о мерах защиты АСУ ТП, бот:

1. Формирует структурированный ответ с выделением ключевых пунктов;
2. Предоставляет ссылки на соответствующие нормативные документы;
3. Предлагает дополнительные уточняющие вопросы через интерактивные кнопки;
4. Форматирует ответ для удобного восприятия на мобильных устройствах.

Синхронизированное взаимодействие модулей

Работа системы представляет собой четко отлаженный процесс последовательной обработки информации:

1. Пользовательский интерфейс принимает исходный запрос
2. Модуль обработки анализирует и преобразует запрос
3. Поисковый модуль находит релевантные данные
4. Результаты интегрируются в связный ответ
5. Пользовательский интерфейс представляет итоговую информацию

Такой подход обеспечивает:

1. Высокую скорость обработки запросов
2. Точность и полноту предоставляемых данных
3. Удобство взаимодействия для конечных пользователей
4. Масштабируемость и возможность дальнейшего развития системы

Представленная трехуровневая архитектура создает надежную основу для построения интеллектуальной системы анализа угроз информационной безопасности. Каждый модуль вносит свой вклад в общую эффективность решения, обеспечивая специалистов ИБ современным инструментом для повседневной работы.

### 1.3.5 Требования к надежности

Надежность интеллектуального бота для анализа угроз информационной безопасности определяется комплексом взаимосвязанных характеристик, среди которых особое значение имеют корректность обработки данных, соответствие актуальным базам угроз ФСТЭК, эффективность методов мониторинга и точность классификации угроз. Корректность работы системы обеспечивается многоуровневой системой валидации выходных данных, включающей как автоматизированные алгоритмы проверки на соответствие формальным логическим критериям, так и экспертный анализ результатов в спорных случаях. Особое внимание уделяется обработке нормативных документов и правовых актов, где даже незначительная ошибка интерпретации может привести к серьезным последствиям. Для обеспечения максимальной точности система должна регулярно проходить аудит с привлечением специалистов в области информационной безопасности и юриспруденции, способных оценить соответствие выводов бота текущему законодательству и отраслевым стандартам.

Критически важным аспектом надежности является поддержание постоянной синхронизации с базами данных ФСТЭК, что требует реализации специального механизма автоматизированного обновления. Этот механизм должен не только своевременно получать актуальные данные об угрозах, но и обеспечивать их корректную интерпретацию в контексте существующей модели анализа рисков. Система обязана отслеживать изменения в классификаторах ФСТЭК и автоматически адаптировать свои алгоритмы работы при появлении новых категорий угроз или изменении критериев их оценки. Для гарантии актуальности данных необходимо предусмотреть ежедневную проверку целостности и полноты полученных сведений, а также механизмы оперативного реагирования на обнаруженные расхождения.

Методы мониторинга надежности работы системы должны охватывать все ключевые аспекты ее функционирования - от качества обработки естественно-языковых запросов до точности формируемых аналитических отчетов. Реализация комплексного мониторинга предполагает непрерывный сбор метрик производительности, фиксацию аномалий в работе алгоритмов и ведение детального лога всех операций. Особое значение имеет мониторинг "граничных случаев", когда система сталкивается с нестандартными запросами или недостаточно четко сформулированными вопросами. Для оперативного выявления потенциальных проблем необходимо использовать сочетание автоматизированных средств анализа и периодического экспертного контроля, что позволяет своевременно обнаруживать и устранять как технические сбои, так и логические ошибки в работе алгоритмов.

Классификация угроз в системе должна соответствовать не только актуальным стандартам ФСТЭК, но и учитывать специфику конкретной организации. Это требует реализации гибкого механизма категоризации, позволяющего адаптировать базовые классификаторы под особенности защищаемой информационной системы. Особое внимание уделяется корректности сопоставления выявленных уязвимостей с типами потенциальных угроз, что является основой для последующей оценки рисков. Система должна обеспечивать возможность ручной корректировки автоматически присвоенных классификаций при обнаружении неточностей, с обязательной фиксацией таких случаев для последующего анализа и совершенствования алгоритмов.

Дополнительными требованиями к надежности системы являются: устойчивость к нагрузкам в пиковые периоды работы, отказоустойчивость критических компонентов, возможность работы в условиях частичной потери связь с внешними источниками данных. Важным аспектом является также обеспечение консистентности данных при одновременной работе множества пользователей, что требует реализации эффективных механизмов блокировок и управления версиями. Система должна предусматривать автоматическое создание контрольных точек восстановления перед выполнением критически важных операций, а также вести подробный аудит всех изменений в базах знаний и классификаторах.

Гарантией общей надежности системы служит ее способность к самодиагностике и автоматическому выявлению потенциальных проблем в работе алгоритмов. Это достигается за счет реализации встроенных механизмов контроля качества, включающих регулярные тестовые прогоны, проверку статистических аномалий в выходных данных и мониторинг отклонений от типовых сценариев работы. Результаты такой самодиагностики должны автоматически учитываться при формировании ответов пользователям, с соответствующей маркировкой потенциально ненадежных выводов и рекомендаций.

### 1.3.6 Условия эксплуатации: требуемая квалификация и уровень подготовки пользователя

Внедрение интеллектуального бота на базе LLM RAG для анализа угроз информационной безопасности с интеграцией базы ФСТЭК требует тщательной проработки условий эксплуатации, включая требования к квалификации пользователей, организационные регламенты и технические аспекты работы системы. Успешность реализации проекта напрямую зависит от соответствия подготовки сотрудников функциональным возможностям инструмента, а также от создания инфраструктуры, обеспечивающей безопасное и эффективное взаимодействие с системой.

Требования к пользователям варьируются в зависимости от их роли в процессе управления информационной безопасностью. Для рядовых сотрудников, которые будут обращаться к системе за справочной информацией или первичной оценкой рисков, критически важны базовые навыки работы с веб-интерфейсами и чат-платформами. Эти пользователи должны уметь четко формулировать запросы на естественном языке, так как качество ответов бота напрямую зависит от точности поставленного вопроса. Минимальное понимание терминологии информационной безопасности — таких понятий, как «уязвимость», «инцидент» или «угроза конфиденциальности» — необходимо для корректной интерпретации результатов. Однако для этой категории пользователей система выступает исключительно в роли помощника, предоставляющего информацию справочного характера без права принятия решений. Ввод конфиденциальных данных или попытки использовать бота для анализа защищенной информации должны быть строго запрещены внутренними регламентами.

Для специалистов по информационной безопасности требования к подготовке значительно выше. Эти пользователи должны не только свободно ориентироваться в нормативной базе — включая 152-ФЗ, 187-ФЗ и требования ФСТЭК, — но и понимать методологии оценки рисков, такие как ISO 27005. Поскольку бот агрегирует данные из различных источников, включая базы уязвимостей CVE и ФСТЭК, специалисты должны уметь верифицировать его выводы, сопоставляя их с первичными источниками. Особое значение имеет понимание архитектуры RAG-систем и ограничений, присущих языковым моделям: например, склонность к «галлюцинациям» или зависимость от качества обучающих данных. На практике это означает, что эксперт должен критически оценивать каждый ответ системы, особенно если он касается интерпретации правовых норм или рекомендаций по устранению уязвимостей. В организациях, работающих с критической информационной инфраструктурой, может дополнительно потребоваться сертификация специалистов по ГОСТ Р 57580 или другим отраслевым стандартам.

Руководители подразделений, использующие систему для стратегического управления рисками, должны обладать навыками работы с аналитическими отчетами и пониманием экономических последствий киберугроз. Для них бот предоставляет агрегированные данные, метрики и прогнозы, позволяющие оценить потенциальный ущерб от реализации тех или иных угроз. В этом контексте особенно полезными оказываются методики финансовой оценки рисков, такие как FAIR или Monte Carlo, интегрированные в систему. Руководители должны уметь интерпретировать эти данные в контексте бизнес-процессов организации, а также формулировать запросы к системе таким образом, чтобы получать информацию, необходимую для принятия управленческих решений.

Организационные условия эксплуатации включают в себя не только обучение пользователей, но и создание четких регламентов работы с системой. Для новых сотрудников обязателен вводный курс, знакомящий с функционалом бота и основными принципами формулирования запросов, в то время как для специалистов и руководителей необходимы регулярные тренинги, учитывающие обновления нормативной базы и появление новых типов угроз. Отдельное внимание должно уделяться документальному закреплению политик использования системы: например, запрету на ввод персональных данных или коммерческой тайны, правилам эскалации спорных ситуаций и порядку взаимодействия с технической поддержкой в случае обнаружения ошибок в работе бота.

Технические аспекты эксплуатации предполагают развертывание системы в защищенном контуре с обязательной интеграцией механизмов контроля доступа и защиты данных. Многофакторная аутентификация, шифрование передаваемой информации и регулярное резервное копирование логов — минимальные требования к инфраструктуре. С точки зрения интеграции, система должна поддерживать подключение к базам угроз в режиме реального времени, а также иметь API для обмена данными с SIEM-решениями и платформами управления рисками. Особое значение имеет настройка модулей мониторинга, отслеживающих потенциально опасные запросы или попытки злоупотребления функционалом бота.

Реализация интеллектуального бота для анализа угроз информационной безопасности — это комплексный процесс, требующий согласованной работы по трем направлениям: подготовка пользователей разного уровня, создание организационных регламентов и обеспечение технической инфраструктуры. Только при условии соблюдения всех перечисленных требований система сможет раскрыть свой потенциал, став не просто справочным инструментом, а полноценным элементом системы управления киберрисками организации.

### 1.3.7 Требования к составу и параметрам технических средств (описание требований к hardware).

Для эффективной работы интеллектуального бота, особенно при обработке больших объемов данных, машинном обучении и сложных NLP-задачах, требуются достаточно высокие аппаратные характеристики. Это обусловлено следующими факторами:

1. Обработка больших объемов данных: для быстрого анализа и генерации ответов боту необходимо оперативно обрабатывать тексты, изображения или другие типы данных. Это требует значительных вычислительных ресурсов, особенно при работе с нейросетевыми моделями (например, GPT, BERT и др.);
2. Использование ML/DL-моделей: Современные модели искусственного интеллекта (ИИ) требуют высокой производительности CPU/GPU и большого объема оперативной памяти для инференса (вывода). Особенно это критично при использовании больших языковых моделей (LLM);
3. Многопоточность и параллельные вычисления: для одновременного обслуживания множества пользователей и быстрого выполнения запросов необходимы многоядерные процессоры и эффективная система распределения нагрузки;
4. Хранение и быстрый доступ к данным: для работы с большими базами знаний, кэширования и быстрого поиска информации важны высокоскоростные SSD-накопители и достаточный объем дискового пространства;
5. Масштабируемость: В случае роста числа пользователей или усложнения алгоритмов система должна иметь запас по мощности, чтобы избежать замедления работы или отказов.

### 1.3.8 Требования к информационной и программной совместимости (описание требований к software)

Требования к информационной и программной совместимости RAG (Retrieval-Augmented Generation)-бота включают ряд ключевых аспектов, связанных с архитектурой системы, интеграцией с внешними источниками данных и обеспечением качества взаимодействия пользователей с ботом. Вот основные требования:

Информационная совместимость:

1. Поддержка форматов данных: бот должен поддерживать различные форматы входных и выходных данных, включая JSON, XML, CSV, HTML и другие распространенные типы файлов. Это позволит обрабатывать данные из разных источников без ограничений формата;
2. Совместимость с различными типами хранилищ данных: важно обеспечить возможность интеграции с реляционными базами данных (MySQL, PostgreSQL), NoSQL-хранилищами (MongoDB, Cassandra), файловыми системами и облачными сервисами хранения данных (Google Cloud Storage, AWS S3);
3. API-интерфейсы: для обеспечения информационного обмена бот должен иметь доступ к RESTful API, GraphQL API и другим современным стандартам веб-сервисов. Обеспечение безопасности запросов и аутентификации также является важным требованием;
4. Обработка больших объемов данных: система должна эффективно масштабироваться и справляться с большими объемами информации без потери производительности. Поддержка технологий распределенных вычислений (Hadoop, Spark) и потоковых обработчиков данных (Kafka, Flink) улучшит производительность обработки;
5. Кэширование и индексирование: необходимо предусмотреть механизмы кэширования часто запрашиваемых данных и использование поисковых индексов (Elasticsearch, Solr) для ускорения операций поиска и извлечения данных.

Программная совместимость:

1. Мультиплатформенность: Telegram-бот должен запускаться и функционировать на различных операционных системах (Windows, Linux, macOS). Это обеспечит гибкость развертывания и доступность решения для широкого круга пользователей;
2. Интеграция с существующими информационными системами: если бот интегрируется с корпоративными ERP-, CRM- и BI-системами, необходимо реализовать поддержку стандартных протоколов и интерфейсов связи (SOAP, JMS, MQ);
3. Интероперабельность с разными языками программирования: архитектура системы должна позволять взаимодействие компонентов, написанных на Python, JavaScript/TypeScript, Java/C#, Go и других популярных языках разработки ПО;
4. Документированность API: предоставление подробной документации по используемым API и интеграция с инструментами автоматического тестирования (Swagger, Postman) упростят внедрение и сопровождение проекта;
5. Безопасность и защита данных: реализация механизмов шифрования данных (TLS, SSL), авторизации (OAuth, JWT) и аудита действий пользователей необходима для защиты конфиденциальной информации от несанкционированного доступа;
6. Масштабируемость и отказоустойчивость: использование контейнеризации (Docker, Kubernetes) и оркестрации сервисов обеспечит горизонтальное масштабирование приложения и минимизирует риски сбоев при увеличении нагрузки;
7. Модульное проектирование: применение принципа разделения ответственности (SOLID) и модульности архитектуры повысит устойчивость системы к изменениям и облегчит процесс внесения улучшений и исправления ошибок.

Соблюдение указанных требований гарантирует создание надежного, эффективного и легко поддерживаемого Telegram-бота, способного успешно решать поставленные перед ним задачи в условиях реальной эксплуатации.

### 1.3.9 Специальные требования

Для реализации специализированного RAG (Retrieval-Augmented Generation)-бота, предназначенного для анализа киберугроз с интеграцией базы Федерального агентства по техническому регулированию и метрологии (ФСТЭК), важно учитывать специальные требования.

Основные аспекты информационных и программных требований

Требования к обработке информации и обеспечению безопасности

1. Конфиденциальность данных: Все обрабатываемые данные (информация о выявленных угрозах, инцидентах, уязвимостях) должны храниться и передаваться в зашифрованном виде. Применяются алгоритмы симметричного и асимметричного шифрования (AES, RSA), TLS-протоколы для защищенной передачи данных между компонентами системы;
2. Авторизация и контроль доступа: Доступ к данным осуществляется через строго контролируемый механизм авторизации (JWT, OAuth), обеспечивающий двухфакторную аутентификацию и ограничение прав доступа сотрудников и автоматизированных агентов;
3. Журналирование и аудит: Регистрация всех операций с информацией, такие как запросы, изменения статусов, получение результатов анализа, должна вестись в режиме реального времени. Журналы сохраняются на удалённых серверах и регулярно проверяются системой мониторинга и контроля нарушений политики безопасности;
4. Идентификация и управление устройствами: Интеграция с системами управления активами и инвентаризацией устройств для отслеживания используемых аппаратных ресурсов и предотвращения попыток проникновения злоумышленников путем компрометации оборудования;
5. Антивирусная проверка поступающих документов: перед обработкой файлы сканируются антивирусными средствами, интегрированными в систему, для выявления вредоносных объектов и сигнатур;
6. Контроль соответствия требованиям регуляторов: База данных ФСТЭК периодически обновляется автоматически и вручную, обеспечивая соответствие правилам регулирования кибербезопасности РФ (ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 57580.1-2017 и др.).

Технические требования к инфраструктуре и функциональности:

1. Анализ содержимого данных: Алгоритмы машинного обучения применяются для классификации входящих сообщений и обнаружения аномалий в поведении сети и серверов, основываясь на сигнатурах атак и индикаторах угрозы (IOC — Indicators of Compromise);
2. Автоматическое обновление моделей и классификаторов: Модели глубокого обучения постоянно совершенствуются путём самообучения на новых данных, полученных из внешних источников и собственных исследований угроз;
3. Мониторинг и предупреждение вторжений: Анализ сетевого трафика, журналов систем и приложений позволяет выявить потенциальные атаки ещё до начала активного вторжения, инициируя автоматические меры противодействия;
4. Быстрое реагирование на инциденты: Автоматизированные сценарии реагируют на подозрительные события в течение нескольких секунд, отключают потенциально заражённые устройства, блокируют трафик и уведомляют специалистов службы безопасности;
5. Создание отчетов и аналитика рисков: Возможность формирования регулярных отчётов по инцидентам, угрозам и рискам, которые предоставляются руководству организации и используются для улучшения защитных мер и планирования мероприятий по повышению уровня безопасности;
6. Расширенная поддержка многопользовательского режима: Несколько команд экспертов одновременно работают над расследованием отдельных инцидентов и используют единую платформу для координации действий, совместного обсуждения ситуации и принятия решений;
7. Поддержка непрерывной доступности сервиса: Система должна обеспечивать бесперебойную работу даже при значительных нагрузках и высокой интенсивности поступления данных. Резервирование серверов и каналов связи обеспечивает отсутствие критичных перерывов в работе системы;
8. Автономная работа при временном отсутствии подключения к внешним ресурсам: Возможна локальная обработка информации без доступа к глобальной сети благодаря использованию встроенных алгоритмов кластеризации и обработки сигналов.

Таким образом, RAG-бот для анализа киберугроз с интеграцией базы ФСТЭК представляет собой сложную инженерную разработку, сочетающую современные технологии аналитики данных, автоматизации процессов и строгих правил информационной безопасности. Эти особенности позволяют создать высокоэффективную систему для оперативного выявления и нейтрализации современных угроз информационной безопасности организаций и предприятий.

## 1.4 Выводы по разделу 1

Telegram-бот с интеграцией LLM и RAG представляет собой актуальный и практически значимый инструмент в условиях роста киберугроз и усиления нормативных требований. Бот решает проблему информационной перегрузки, автоматизирует ключевые этапы анализа и облегчает работу специалистов по ИБ. Простота интерфейса, мобильность и возможность масштабирования делают проект особенно полезным как для организаций, так и для образовательных учреждений.

В результате анализа существующих IT-разработок наиболее эффективной стратегией представляется комбинированный подход. Для базовой защиты оптимально сочетание статических баз с LLM-ботами, что обеспечит баланс между стоимостью и функциональностью. В корпоративном секторе, где критичны масштабируемость и глубокая аналитика, предпочтительна интеграция SIEM с ML-решениями, дополненная статическими базами для соблюдения нормативных требований. Такой гибридный подход позволяет максимально использовать сильные стороны каждой технологии, минимизируя их недостатки.

Основные преимущества решения заключаются в значительном сокращении времени анализа угроз, минимизации человеческого фактора и обеспечении соответствия требованиям регуляторов. Техническая реализация требует мощного аппаратного обеспечения, включая производительные процессоры, значительный объем оперативной памяти и быстрые SSD-накопители. Программная часть предусматривает поддержку современных API, совместимость с различными СУБД и надежные механизмы шифрования данных. Особые требования предъявляются к надежности системы, включая многоуровневую валидацию, регулярную синхронизацию с базой ФСТЭК и комплексный мониторинг работы.

Перспективы внедрения решения охватывают широкий круг организаций, включая операторов критической информационной инфраструктуры, государственные учреждения и коммерческие компании, работающие с персональными данными. Реализация проекта позволит существенно повысить эффективность работы служб информационной безопасности за счет автоматизации рутинных процессов, обеспечения оперативного реагирования на угрозы и соблюдения требований регуляторов. Разрабатываемое решение представляет собой современный подход к построению систем информационной безопасности, сочетающий передовые технологии искусственного интеллекта с актуальными требованиями российского законодательства в данной области.

# 2 РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО БОТА ДЛЯ АНАЛИЗА УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

## 2.1 Теоретическое решение задачи и модель разработки интеллектуального бота

Разработка интеллектуального решения для автоматизации анализа угроз и рисков в сфере информационной безопасности (ИБ) является актуальной задачей в условиях стремительного роста киберугроз и усложнения методов атак, направленных на объекты критической информационной инфраструктуры (КИИ) и персональные данные. Согласно данным ФСТЭК, 47% организаций КИИ находятся в критическом состоянии защищенности, что подчеркивает необходимость внедрения автоматизированных систем для минимизации человеческого фактора, сокращения временных затрат и обеспечения соответствия нормативным требованиям. В данном контексте разработка Telegram-бота на базе технологий больших языковых моделей (LLM) и генерации с подкреплением выборкой (RAG), интегрированного с базой данных угроз ФСТЭК, представляет собой инновационное решение, способное оптимизировать процессы анализа угроз и рисков в ИБ.

Целью настоящего раздела является изложение теоретической основы и формального подхода к созданию IT-решения, включающего разработку интеллектуального Telegram-бота для анализа угроз ИБ с правовой оценкой на основе базы данных ФСТЭК. В разделе описывается модель решения, включающая ключевые компоненты — LLM для обработки естественно-языковых запросов, RAG для семантического поиска и интеграции с нормативной базой, а также Telegram-бот как пользовательский интерфейс. Также представлен алгоритм работы системы, охватывающий этапы обработки запросов, извлечения данных и генерации аналитических выводов. Особое внимание уделяется адаптации решения к российским нормативным требованиям, включая стандарты ФСТЭК, ФСБ и Роскомнадзора, а также поддержке обработки документов на русском языке.

Данный раздел опирается на результаты, полученные в первом разделе работы. В частности, в подразделе 1.1 был проведен анализ проблем ручного анализа угроз ИБ, выявлены его ключевые недостатки: высокие временные затраты, вероятность человеческих ошибок и сложность поддержания актуальности нормативной базы. Подраздел 1.2 содержал критический обзор существующих подходов к автоматизации анализа угроз, включая статические базы данных (например, БДУ ФСТЭК, MITRE ATT&CK), системы SIEM (Splunk, IBM QRadar), решения на основе машинного обучения (Darktrace, Vectra AI) и чат-боты на базе LLM (GPT-4, Gemini). Анализ показал, что гибридный подход, сочетающий LLM и RAG, обладает значительными преимуществами: способностью обрабатывать сложные запросы, обеспечивать актуальность данных и минимизировать зависимость от переобучения модели. Эти выводы легли в основу выбора архитектуры для разрабатываемого решения.

Раздел 2.1 фокусируется на формальном описании модели и алгоритмов, лежащих в основе интеллектуального бота. Основное внимание уделяется следующим аспектам:

1. модель системы: описание взаимодействия компонентов LLM, RAG и Telegram-бота, включая структурно-функциональную схему;
2. алгоритм обработки запросов: последовательность этапов от получения запроса пользователя до генерации ответа, включая токенизацию, эмбеддинг, семантический поиск и правовую оценку;
3. интеграция с базой ФСТЭК: методы обработки и индексации нормативных документов для обеспечения релевантности и актуальности данных;
4. теоретическое обоснование: преимущества выбранного подхода в сравнении с альтернативами, адаптация к российским стандартам ИБ и потенциал масштабирования.

### 2.1.1 Подробное описание теоретической модели решения.

Теоретическая модель разрабатываемого IT-решения представляет собой гибридную систему, интегрирующую технологии больших языковых моделей (LLM) и генерации с подкреплением выборкой (RAG) для автоматизации анализа угроз и рисков информационной безопасности (ИБ) на основе базы данных угроз ФСТЭК. Модель разработана с учетом необходимости минимизации временных затрат, человеческого фактора и обеспечения соответствия нормативным требованиям, установленным ФСТЭК, ФСБ и Роскомнадзором. Основная цель модели — предоставить специалистам ИБ удобный и эффективный инструмент для получения справочной информации, классификации угроз и правовой оценки рисков в соответствии с российскими стандартами, такими как ГОСТ Р 57580 и ФЗ-187 «О безопасности критической информационной инфраструктуры».

Гибридная архитектура решения сочетает возможности LLM для обработки естественно-языковых запросов и RAG для извлечения релевантной информации из нормативной базы данных. Данная модель позволяет автоматизировать процессы, которые при ручном анализе требуют значительных временных и кадровых ресурсов, а также снижает вероятность ошибок, связанных с интерпретацией сложных нормативных документов. Основные преимущества гибридного подхода заключаются в:

1. гибкости обработки запросов: LLM интерпретирует сложные и неоднозначные запросы пользователей, включая формулировки, характерные для нормативных документов;
2. актуальности данных: RAG обеспечивает доступ к актуальной информации из базы ФСТЭК без необходимости переобучения языковой модели;
3. удобстве взаимодействия: Telegram-бот, реализованный с использованием библиотеки aiogram, предоставляет интуитивно понятный интерфейс для специалистов ИБ, позволяя оперативно получать справочную информацию и рекомендации по мерам защиты.

Модель ориентирована на корпоративное использование, включая операторов критической информационной инфраструктуры (КИИ), коммерческие организации, работающие с персональными данными, и государственные учреждения. Она также предусматривает возможность интеграции с другими системами мониторинга угроз, такими как SIEM, для повышения эффективности реагирования на инциденты.

### 2.1.2 Ключевые компоненты модели.

Теоретическая модель включает три основных функциональных компонента, которые взаимодействуют для обработки запросов и генерации аналитических выводов.

Модуль обработки запросов (LLM):

Данный модуль основан на большой языковой модели, адаптированной для обработки текстов на русском языке и анализа нормативных документов.

Отвечает за токенизацию запросов, преобразование текста в числовые векторы (эмбеддинги) и интерпретацию контекста запроса с использованием механизма самовнимания (self-attention).

Пример: запрос «Какие угрозы связаны с утечкой персональных данных?» преобразуется в структурированную команду для последующего поиска.

Модуль семантического поиска (RAG):

Использует технологию Retrieval-Augmented Generation для извлечения релевантных данных из базы угроз ФСТЭК.

Работает с векторным хранилищем (например, FAISS), в котором нормативные документы ФСТЭК проиндексированы в виде эмбеддингов для быстрого семантического поиска.

Обеспечивает актуальность данных, минимизируя зависимость от статических баз и переобучения модели.

Пользовательский интерфейс (Telegram-бот):

Реализован на основе библиотеки aiogram для обработки сообщений и команд в Telegram.

Служит точкой взаимодействия между пользователем и системой, обеспечивая ввод запросов и вывод структурированных ответов (например, в виде таблиц, списков или текстовых рекомендаций).

Пример: пользователь запрашивает «Меры защиты для АСУ ТП», а бот возвращает список угроз и соответствующие рекомендации из базы ФСТЭК.

Дополнительно в модель включен модуль интеграции, который обеспечивает подключение к базе данных ФСТЭК и синхронизацию данных. Этот модуль отвечает за преобразование нормативных документов в формат, пригодный для семантического поиска, и обновление векторного хранилища при появлении новых данных.

### 2.1.3 Структурно-функциональная схема

Для наглядного представления взаимодействия компонентов разработана структурно-функциональная модель системы, представленная на рисунке 7. Схема иллюстрирует поток данных от ввода запроса до генерации ответа:

1. входной запрос: Пользователь вводит запрос через Telegram-бот (например, «Найти угрозы для серверов Windows»);
2. обработка запроса (LLM): Запрос токенизируется, преобразуется в эмбеддинги и анализируется с учетом контекста;
3. поиск данных (RAG): Модуль RAG выполняет семантический поиск по векторному хранилищу, извлекая релевантные записи из базы ФСТЭК;
4. генерация ответа (LLM): LLM формирует ответ, объединяя контекст запроса и извлеченные данные, с учетом нормативных требований (например, классификация угроз по ГОСТ Р 57580);
5. вывод результата: Telegram-бот форматирует ответ и отправляет его пользователю в удобном виде (например, список угроз с описанием и мерами защиты).

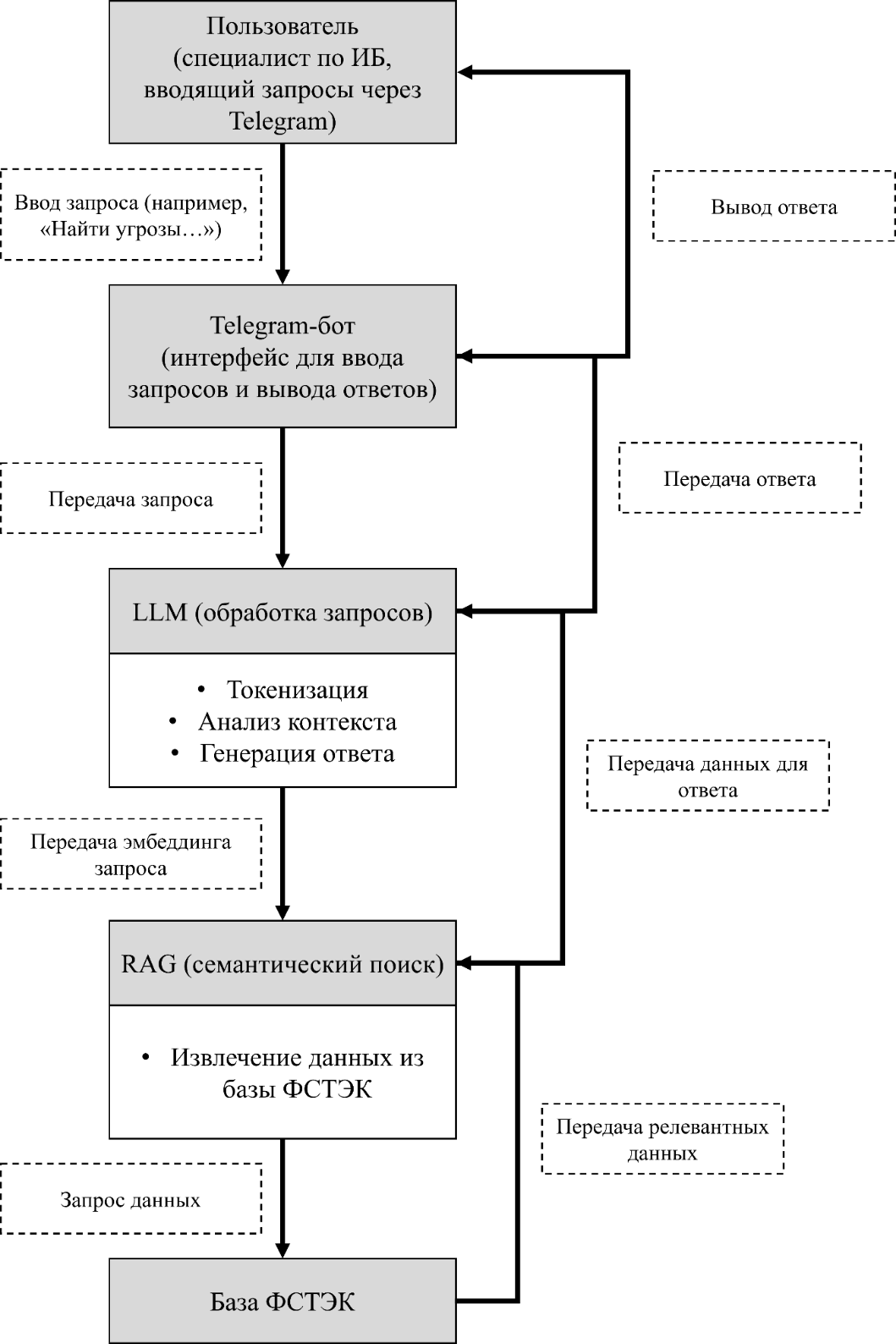


Рисунок 7 – Структурно-функциональная модель системы

Адаптация к российским нормативным требованиям

Модель разработана с учетом специфики российского нормативного поля, что включает:

1. соответствие стандартам: Обеспечение анализа угроз в соответствии с ГОСТ Р 57580, ФЗ-187 и приказами ФСТЭК, включая классификацию угроз и рекомендации по мерам защиты;
2. поддержка русского языка: Использование языковых моделей, оптимизированных для обработки текстов на русском языке, что критически важно для интерпретации нормативных документов и запросов пользователей;
3. интеграция с базой ФСТЭК: Модель предусматривает работу с реестром угроз, матрицей способов реализации и метриками опасности (например, CVSS), что позволяет генерировать правовые оценки рисков.

Преимущества теоретической модели

Предложенная модель обладает следующими преимуществами:

1. автоматизация процессов: Снижение временных затрат на анализ угроз по сравнению с ручными методами, что особенно важно в условиях дефицита квалифицированных специалистов ИБ (по данным исследования [3], дефицит кадров в 2023 году составил 45%);
2. снижение человеческого фактора: Минимизация ошибок, связанных с ручной интерпретацией нормативных документов, благодаря использованию формализованных алгоритмов LLM и RAG;
3. актуальность данных: RAG обеспечивает доступ к актуальной базе ФСТЭК, что устраняет проблему устаревания данных, характерную для статических баз (см. раздел 1.2);
4. масштабируемость: Возможность интеграции с другими системами, такими как SIEM, для комплексного мониторинга и реагирования на инциденты;
5. удобство использования: Telegram-бот предоставляет интуитивно понятный интерфейс, доступный для специалистов ИБ с минимальной подготовкой.

Ограничения модели

Несмотря на преимущества, модель имеет ограничения, выявленные на основе анализа существующих подходов (раздел 1.2):

1. зависимость от качества данных: Эффективность RAG зависит от полноты и качества индексации базы ФСТЭК;
2. ограниченный контекст LLM: Размер обрабатываемого контекста может ограничивать глубину анализа сложных запросов;
3. риски «галлюцинаций»: LLM могут генерировать недостоверные ответы при недостатке данных или некорректных запросах, что требует проверки результатов.

Для преодоления этих ограничений модель предусматривает регулярное обновление векторного хранилища и использование механизмов валидации ответов, основанных на нормативных данных ФСТЭК.

Теоретическая модель решения представляет собой гибридную систему, объединяющую LLM, RAG и Telegram-бот для автоматизации анализа угроз ИБ. Модель включает модули обработки запросов, семантического поиска, интеграции с базой ФСТЭК и пользовательского интерфейса, что обеспечивает оперативность, точность и соответствие российским стандартам. Структурно-функциональная схема иллюстрирует взаимодействие компонентов, а адаптация к нормативным требованиям и русскому языку подчеркивает практическую значимость решения. Модель создает основу для дальнейшей реализации, описанной в последующих подразделах, и решает ключевые проблемы ручного анализа угроз, выявленные в разделе 1.1.

### 2.1.4 Вывод по подразделу

Подраздел 2.1 посвящен разработке теоретической основы Telegram-бота для автоматизированного анализа угроз и рисков информационной безопасности (ИБ) с использованием базы ФСТЭК. Описана гибридная модель, объединяющая большие языковые модели (LLM) и технологию RAG, а также алгоритм обработки запросов, решающий проблемы, выявленные в разделе 1.1: высокие временные затраты, человеческие ошибки и сложность актуализации нормативной базы. Модель учитывает российские стандарты (ФСТЭК, ФСБ, Роскомнадзор), что делает её подходящей для критической информационной инфраструктуры (КИИ).

Модель включает три компонента: модуль обработки запросов (LLM), семантический поиск (RAG) и Telegram-бот на aiogram. Она обрабатывает естественно-языковые запросы, извлекает данные из базы ФСТЭК и формирует структурированные ответы с правовыми оценками и рекомендациями. Алгоритм включает этапы: получение запроса, предобработка, семантический поиск, синтез ответа и вывод через бот. Используются трансформерные модели и векторные хранилища (FAISS), что сокращает время анализа и повышает точность за счет RAG, минимизирующего «галлюцинации» LLM.

Интеграция с базой ФСТЭК обеспечивает доступ к реестру угроз и мерам защиты, а индексация документов в виде эмбеддингов поддерживает актуальность данных. Модель масштабируема, с потенциалом интеграции с SIEM, и адаптирована к российским стандартам (ГОСТ Р 57580, ФЗ-187), что устраняет барьеры зарубежных решений. Telegram-бот предоставляет удобный интерфейс для специалистов ИБ.

Преимущества модели:

1. автоматизация, сокращающая время анализа;
2. точность и актуальность за счет RAG и базы ФСТЭК;
3. гибкость в обработке сложных запросов;
4. масштабируемость для интеграции с SIEM и DLP;
5. удобство Telegram-интерфейса.

Ограничения: зависимость от качества данных ФСТЭК, ограниченный контекст LLM и риски «галлюцинаций». Эти проблемы решаются обновлением данных, валидацией ответов и добавлением источников.

Решение создает основу для практической реализации (подразделы 2.2–2.4), превосходя статические базы, SIEM и традиционные ML-решения по гибкости и актуальности. Оно применимо для КИИ, коммерческих и государственных организаций.

# 2.2 Стек используемых технологий

### 2.2.1 Введение в стек технологий и цели разработки

Разработка интеллектуального Telegram-бота для автоматизированного анализа угроз и рисков в сфере информационной безопасности с правовой оценкой на основе базы данных ФСТЭК требует тщательного выбора технологического стека, который обеспечит высокую производительность, гибкость обработки запросов, адаптацию к российским нормативным требованиям и минимизацию зависимости от человеческого фактора. Данный подраздел посвящен обоснованию выбора технологий, используемых для реализации системы, включая язык программирования, библиотеки, фреймворки, хранилища данных, а также описание архитектуры решения, охватывающей фронтенд и бэкенд.

Основной целью является создание системы, способной эффективно обрабатывать естественно-языковые запросы специалистов по информационной безопасности, извлекать релевантную информацию из нормативной базы ФСТЭК и предоставлять структурированные ответы, соответствующие стандартам ГОСТ Р 57580, ФЗ-187 и приказам ФСТЭК. Выбор технологий опирается на результаты анализа, проведенного в предыдущих разделах работы. В разделе 1.1 были выявлены ключевые проблемы ручного анализа угроз, такие как значительные временные затраты, высокая вероятность ошибок из-за человеческого фактора и сложность поддержания актуальности нормативной базы в условиях стремительного роста киберугроз.

Подраздел 1.2 содержал критический обзор существующих подходов к автоматизации анализа угроз, включая статические базы данных, системы SIEM, решения на основе машинного обучения и чат-боты на базе больших языковых моделей. Анализ показал, что гибридный подход, сочетающий возможности больших языковых моделей и технологии генерации с подкреплением выборкой, обладает значительными преимуществами: способностью обрабатывать сложные запросы, обеспечивать актуальность данных за счет динамического поиска и минимизировать необходимость переобучения модели. Эти выводы легли в основу выбора архитектуры и технологий для разрабатываемого решения.

Подраздел 2.1 представил теоретическую модель системы, включающую модули обработки запросов на основе LLM, семантического поиска с использованием RAG, интеграции с базой данных ФСТЭК и пользовательского интерфейса в виде Telegram-бота.

Настоящий подраздел развивает эти идеи, детализируя технологический стек и архитектурные решения, обеспечивающие выполнение поставленных задач. В частности, акцент сделан на выборе технологий, которые поддерживают обработку русскоязычных нормативных документов, обеспечивают высокую скорость обработки запросов, минимизируют риски недостоверных ответов и позволяют масштабировать систему для использования в организациях различного масштаба, включая операторов критической информационной инфраструктуры, коммерческие компании и государственные учреждения. Подраздел включает описание компонентов системы, таких как модуль обработки запросов, модуль семантического поиска, хранилища данных и пользовательский интерфейс, а также обоснование их выбора с учетом требований к производительности, доступности и удобству использования.

Обоснование выбора технологий

Разработка интеллектуального Telegram-бота для автоматизированного анализа угроз и рисков в сфере информационной безопасности (ИБ) с правовой оценкой на основе базы данных ФСТЭК требует выбора технологий, которые обеспечат высокую производительность, гибкость обработки запросов, соответствие российским нормативным требованиям и минимизацию человеческого фактора. Выбор технологического стека для данного проекта основывается на ряде критериев, которые вытекают из задач, поставленных в работе, а также на результатах анализа проблем ручного анализа угроз (раздел 1.1) и критического обзора существующих подходов (раздел 1.2).

Основные критерии выбора технологий

Основные критерии включают способность технологий эффективно обрабатывать естественно-языковые запросы, обеспечивать актуальность данных, интегрироваться с базой ФСТЭК, поддерживать русскоязычные документы, обеспечивать масштабируемость системы и предоставлять удобный интерфейс для специалистов по ИБ с минимальной подготовкой. Кроме того, учитывалась необходимость минимизации затрат на внедрение и эксплуатацию, что особенно важно для организаций с ограниченными ресурсами, таких как малый бизнес или государственные учреждения с дефицитом квалифицированных кадров.

Ключевым аспектом выбора технологий является их способность решать проблемы, выявленные в разделе 1.1, такие как высокие временные затраты на ручной анализ угроз, вероятность человеческих ошибок и сложность поддержания актуальности нормативной базы. В разделе 1.2 было установлено, что существующие подходы, такие как статические базы данных (например, БДУ ФСТЭК, MITRE ATT&CK), системы SIEM (Splunk, IBM QRadar), решения на основе машинного обучения (Darktrace, Vectra AI) и чат-боты на базе LLM (GPT-4, Gemini), имеют свои преимущества и ограничения. Статические базы данных страдают от задержек в обновлении и неспособности учитывать новые угрозы, системы SIEM требуют значительных ресурсов и квалифицированных специалистов, а традиционные решения на основе машинного обучения менее гибки в обработке сложных текстовых запросов. Чат-боты на базе LLM показали высокую гибкость в интерпретации запросов, но их эффективность ограничена без интеграции с актуальными данными. На основе этого анализа был сделан вывод, что гибридный подход, сочетающий большие языковые модели (LLM) и технологию генерации с подкреплением выборкой (RAG), является оптимальным для достижения целей проекта.

Большие языковые модели были выбраны в качестве основного компонента для обработки естественно-языковых запросов благодаря их способности интерпретировать сложные формулировки, характерные для нормативных документов, и генерировать структурированные ответы. В отличие от традиционных моделей машинного обучения, таких как те, что используются в Darktrace или Vectra AI, LLM не требуют предварительного обучения на специфических наборах данных для каждой задачи, что снижает затраты на разработку и адаптацию. Например, русскоязычная модель, такая как GigaChat от Сбера, позволяет эффективно обрабатывать запросы на русском языке, что критически важно для работы с нормативными документами ФСТЭК, содержащими профессиональную терминологию. LLM также поддерживают контекстное понимание, что позволяет боту учитывать предыдущие запросы в рамках диалога, делая взаимодействие с пользователем более естественным. Однако LLM имеют ограничения, такие как риск «галлюцинаций» — генерации недостоверных ответов при недостатке данных. Для преодоления этого недостатка в систему интегрирована технология RAG.

Технология генерации с подкреплением выборкой (RAG) была выбрана для обеспечения актуальности данных и минимизации риска недостоверных ответов. В отличие от статических баз данных, таких как БДУ ФСТЭК, которые требуют ручного обновления и могут не учитывать новые угрозы, RAG позволяет динамически извлекать релевантную информацию из базы данных ФСТЭК через семантический поиск. Это достигается за счет преобразования нормативных документов в числовые векторы (эмбеддинги) и их хранения в векторном хранилище, таком как FAISS. RAG обеспечивает доступ к актуальным данным без необходимости переобучения языковой модели, что делает систему более гибкой и устойчивой к изменениям в ландшафте киберугроз. По сравнению с системами SIEM, которые требуют сложной настройки корреляционных правил и мощных серверов для обработки больших объемов логов, RAG предлагает более легковесное решение, ориентированное на текстовый анализ и интеграцию с нормативными базами.

Для реализации пользовательского интерфейса был выбран Telegram-бот, так как платформа Telegram широко используется в России, доступна на мобильных и десктопных устройствах и предоставляет удобный API для разработки. В отличие от веб-интерфейсов или десктопных приложений, Telegram-бот не требует установки дополнительного программного обеспечения, что упрощает внедрение и использование в организациях. Библиотека aiogram, основанная на Python, была выбрана для реализации бота благодаря ее асинхронной архитектуре, которая обеспечивает высокую производительность при обработке множества запросов, и поддержке интерактивных элементов, таких как кнопки для выбора категорий угроз. По сравнению с другими платформами, такими как Slack или WhatsApp, Telegram имеет меньшие ограничения на API и более широкую популярность среди российских пользователей, что делает его предпочтительным выбором для корпоративного использования.

Язык программирования Python был выбран как основа для разработки благодаря его универсальности, богатой экосистеме библиотек для работы с искусственным интеллектом и обработки данных, а также поддержке русскоязычных текстов. Python широко используется в проектах, связанных с машинным обучением и обработкой естественного языка, что подтверждается успешными кейсами, такими как проект «Авито» по анализу отзывов с использованием LLM. Альтернативы, такие как JavaScript или Java, были рассмотрены, но отклонены. JavaScript, хотя и подходит для веб-приложений, имеет ограниченную поддержку библиотек для работы с LLM и RAG, а Java требует более сложной настройки для интеграции с современными фреймворками ИИ. Python, напротив, предоставляет доступ к библиотекам, таким как Transformers (Hugging Face) для работы с LLM, FAISS для векторного поиска и Pandas для обработки данных, что делает его оптимальным выбором.

Для хранения данных выбраны два типа хранилищ: векторное хранилище FAISS для семантического поиска и реляционная база данных PostgreSQL для управления метаданными и логами. FAISS, разработанная Facebook AI, обеспечивает высокую скорость поиска по эмбеддингам, что критически важно для обработки больших объемов нормативных документов ФСТЭК. В отличие от альтернатив, таких как Elasticsearch, FAISS оптимизирован для работы с векторными данными и требует меньше вычислительных ресурсов. PostgreSQL была выбрана как реляционная база данных благодаря своей надежности, поддержке сложных запросов и совместимости с российскими облачными платформами, такими как Yandex Cloud, что важно для соответствия требованиям ФСБ и Роскомнадзора. Альтернативы, такие как MySQL, были отклонены из-за меньшей гибкости в обработке сложных структур данных.

Облачные платформы, такие как Yandex Cloud, рассматривались для развертывания системы с целью обеспечения масштабируемости и соответствия российским требованиям по хранению данных. В отличие от зарубежных облаков, таких как AWS или Azure, Yandex Cloud сертифицирована для работы с персональными данными и соответствует требованиям ФЗ-152, что делает ее предпочтительной для организаций, работающих с критической информационной инфраструктурой. Локальное развертывание также возможно для организаций, предпочитающих автономные решения, но облачные платформы обеспечивают большую гибкость и снижение затрат на инфраструктуру.

Сравнение с аналогами

Сравнение альтернативных технологий проводилось с учетом их применимости к задачам проекта. Например, традиционные системы SIEM, такие как IBM QRadar, обеспечивают мощный анализ логов, но требуют значительных ресурсов и квалифицированных специалистов, что не подходит для малого бизнеса. Решения на основе машинного обучения, такие как Darktrace, эффективны для обнаружения аномалий, но менее универсальны для обработки текстовых нормативных документов. Чат-боты на базе LLM, такие как GPT-4, демонстрируют высокую гибкость, но без RAG их ответы могут быть недостоверными. Гибридный подход LLM+RAG устраняет эти недостатки, обеспечивая точность, актуальность и удобство использования. Выбор Telegram как платформы для интерфейса также обоснован его кроссплатформенностью и минимальными затратами на внедрение по сравнению с разработкой веб-приложений.

Таким образом, выбор технологий основывается на их способности решать ключевые задачи проекта: автоматизация анализа угроз, интеграция с базой ФСТЭК, поддержка русского языка и обеспечение масштабируемости. Python, aiogram, Transformers, FAISS, PostgreSQL и Yandex Cloud составляют технологический стек, который оптимально соответствует требованиям производительности, доступности и нормативного соответствия. Этот выбор позволяет создать систему, которая минимизирует временные затраты на анализ угроз, снижает вероятность ошибок и обеспечивает удобство взаимодействия для специалистов по информационной безопасности.

### 2.2.2 Архитектура решения

Разработка интеллектуального Telegram-бота для анализа угроз и рисков в сфере информационной безопасности с правовой оценкой на основе базы данных ФСТЭК требует продуманной архитектуры, которая объединяет компоненты для обработки запросов, извлечения данных, их хранения и предоставления пользователю в удобной форме. Архитектура решения определяет взаимодействие всех элементов системы, обеспечивая высокую производительность, надежность и соответствие требованиям российских нормативных актов, таких как ГОСТ Р 57580 и ФЗ-187.

Основная задача архитектуры заключается в создании эффективного конвейера обработки данных: от получения запроса пользователя до формирования структурированного ответа с учетом актуальной нормативной базы. В отличие от традиционных систем анализа угроз, таких как SIEM или статические базы данных, предложенная архитектура ориентирована на автоматизацию процессов, минимизацию человеческого фактора и адаптацию к динамично меняющемуся ландшафту киберугроз, что особенно важно в условиях ежегодного роста кибератак на 25–30%, как указано в разделе 1.1. Архитектура опирается на выводы подраздела 2.1, где описана теоретическая модель системы, и расширяет ее, предоставляя детальное описание компонентов и их взаимодействия.

Система состоит из нескольких ключевых модулей, каждый из которых выполняет специализированную функцию, обеспечивая целостность и эффективность работы. Первым компонентом является модуль обработки запросов, реализованный на базе большой языковой модели. Этот модуль отвечает за анализ естественно-языковых запросов, поступающих от пользователей через Telegram-бот. Он выполняет токенизацию текста, преобразование запроса в числовые представления (эмбеддинги) и интерпретацию контекста, что позволяет понимать сложные формулировки, характерные для профессиональной терминологии в области информационной безопасности.

Например, запрос «Какие меры защиты рекомендованы для предотвращения утечки данных через уязвимости АСУ ТП?» обрабатывается с учетом контекста и специфики нормативных требований. Использование русскоязычной модели, оптимизированной для работы с юридическими и техническими текстами, обеспечивает точность интерпретации и соответствие российским стандартам. Этот модуль взаимодействует с другими компонентами системы, передавая обработанные данные для дальнейшего поиска и генерации ответа.

Вторым ключевым компонентом является модуль семантического поиска, реализованный с использованием технологии генерации с подкреплением выборкой. Этот модуль отвечает за извлечение релевантной информации из базы данных ФСТЭК, которая содержит нормативные документы, описания угроз и рекомендации по их устранению. В отличие от традиционных поисковых систем, основанных на ключевых словах, семантический поиск использует векторные представления данных, что позволяет находить документы, наиболее близкие по смыслу к запросу пользователя. Например, если пользователь запрашивает информацию об угрозах, связанных с «несанкционированным доступом», модуль способен найти не только документы, содержащие эту фразу, но и связанные с ней по смыслу, такие как «взлом учетных записей» или «атаки на аутентификацию». Это достигается за счет преобразования нормативных документов в эмбеддинги с использованием трансформерных моделей и их хранения в специализированном векторном хранилище. Такой подход обеспечивает высокую точность поиска и актуальность данных, что критически важно в условиях, когда база ФСТЭК регулярно обновляется.

Третий компонент — пользовательский интерфейс, реализованный в виде Telegram-бота. Этот модуль обеспечивает взаимодействие пользователя с системой, принимая текстовые запросы и предоставляя структурированные ответы в виде текста, таблиц или списков. Telegram-бот был выбран как основной интерфейс благодаря его доступности, кроссплатформенности и простоте интеграции через API. Пользователи могут вводить запросы в свободной форме, а система возвращает ответы, содержащие классификацию угроз, рекомендации по мерам защиты и ссылки на соответствующие нормативные документы.

Например, ответ на запрос может включать таблицу с перечнем угроз, их описанием и рекомендованными мерами защиты, оформленную в соответствии с ГОСТ Р 57580. Бот также поддерживает интерактивные элементы, такие как кнопки для выбора категорий угроз, что упрощает навигацию для специалистов с минимальным опытом работы с ИИ-системами.

Четвертым компонентом является модуль интеграции с базой данных ФСТЭК. Этот модуль отвечает за преобразование нормативных документов, представленных в формате XML или JSON, в эмбеддинги, которые затем индексируются в векторном хранилище. Интеграция осуществляется через парсинг данных или, при наличии, через API ФСТЭК, что позволяет поддерживать актуальность информации. Модуль также обеспечивает периодическое обновление данных, что решает проблему устаревания статических баз, выявленную в разделе 1.2.

Например, при добавлении новой угрозы в базу ФСТЭК модуль автоматически преобразует соответствующий документ в эмбеддинг и обновляет индекс в хранилище, что гарантирует доступность актуальных данных для семантического поиска. Такой подход минимизирует задержки, связанные с ручным обновлением баз, и обеспечивает соответствие требованиям нормативных актов.

Используемые хранилища

Для хранения данных используются два типа хранилищ. Векторное хранилище, реализованное на базе FAISS, предназначено для хранения эмбеддингов нормативных документов и выполнения быстрого семантического поиска. FAISS обеспечивает высокую производительность при работе с большими объемами данных, что особенно важно для обработки обширной базы ФСТЭК, содержащей тысячи документов. Реляционная база данных, реализованная на PostgreSQL, используется для хранения метаданных, таких как история запросов, логи системы и информация о пользователях. PostgreSQL обеспечивает надежность, поддержку сложных запросов и совместимость с российскими облачными платформами, что важно для соблюдения требований ФСБ и Роскомнадзора.

Например, логи запросов могут использоваться для анализа активности пользователей и оптимизации работы системы, а метаданные помогают отслеживать изменения в базе ФСТЭК.

Поток данных в системе организован следующим образом: пользователь отправляет запрос через Telegram-бот, который передает его в модуль обработки запросов. Модуль LLM выполняет токенизацию и преобразование запроса в эмбеддинг, после чего передает его в модуль семантического поиска. Модуль RAG осуществляет поиск релевантных документов в векторном хранилище, извлекая данные из базы ФСТЭК. На основе найденной информации LLM формирует структурированный ответ, который возвращается пользователю через Telegram-бот. Этот процесс занимает от нескольких секунд до минуты, что значительно быстрее ручного анализа, требующего часов или дней, как указано в разделе 1.1. Архитектура поддерживает асинхронную обработку, что позволяет одновременно обрабатывать запросы от нескольких пользователей, обеспечивая масштабируемость системы.

Система разработана с учетом российских нормативных требований, таких как ФЗ-187 и ГОСТ Р 57580, что обеспечивается за счет интеграции с базой ФСТЭК и использования русскоязычных моделей. Архитектура также предусматривает возможность масштабирования: добавление новых источников данных, таких как базы MITRE ATT&CK или внутренние документы организации, может быть реализовано путем расширения модуля интеграции. Кроме того, система поддерживает развертывание как в облаке (например, Yandex Cloud), так и на локальных серверах, что делает ее гибкой для различных сценариев использования.

Например, крупные операторы КИИ могут развернуть систему на собственных серверах для обеспечения автономности, тогда как малый бизнес может использовать облачное решение для снижения затрат.

Таким образом, архитектура решения представляет собой модульную систему, включающую обработку запросов на основе LLM, семантический поиск с использованием RAG, пользовательский интерфейс в виде Telegram-бота, модуль интеграции с базой ФСТЭК и два типа хранилищ данных. Эта структура обеспечивает высокую производительность, актуальность данных и удобство использования, решая ключевые проблемы ручного анализа угроз, выявленные в разделе 1.1, и превосходя ограничения существующих подходов, описанных в разделе 1.2.

Архитектура является основой для дальнейшей реализации системы, описанной в последующих пунктах подраздела, и гарантирует достижение целей проекта по автоматизации анализа угроз и рисков в сфере информационной безопасности.

Блок-схема архитектуры решения приведена на рисунке 8.

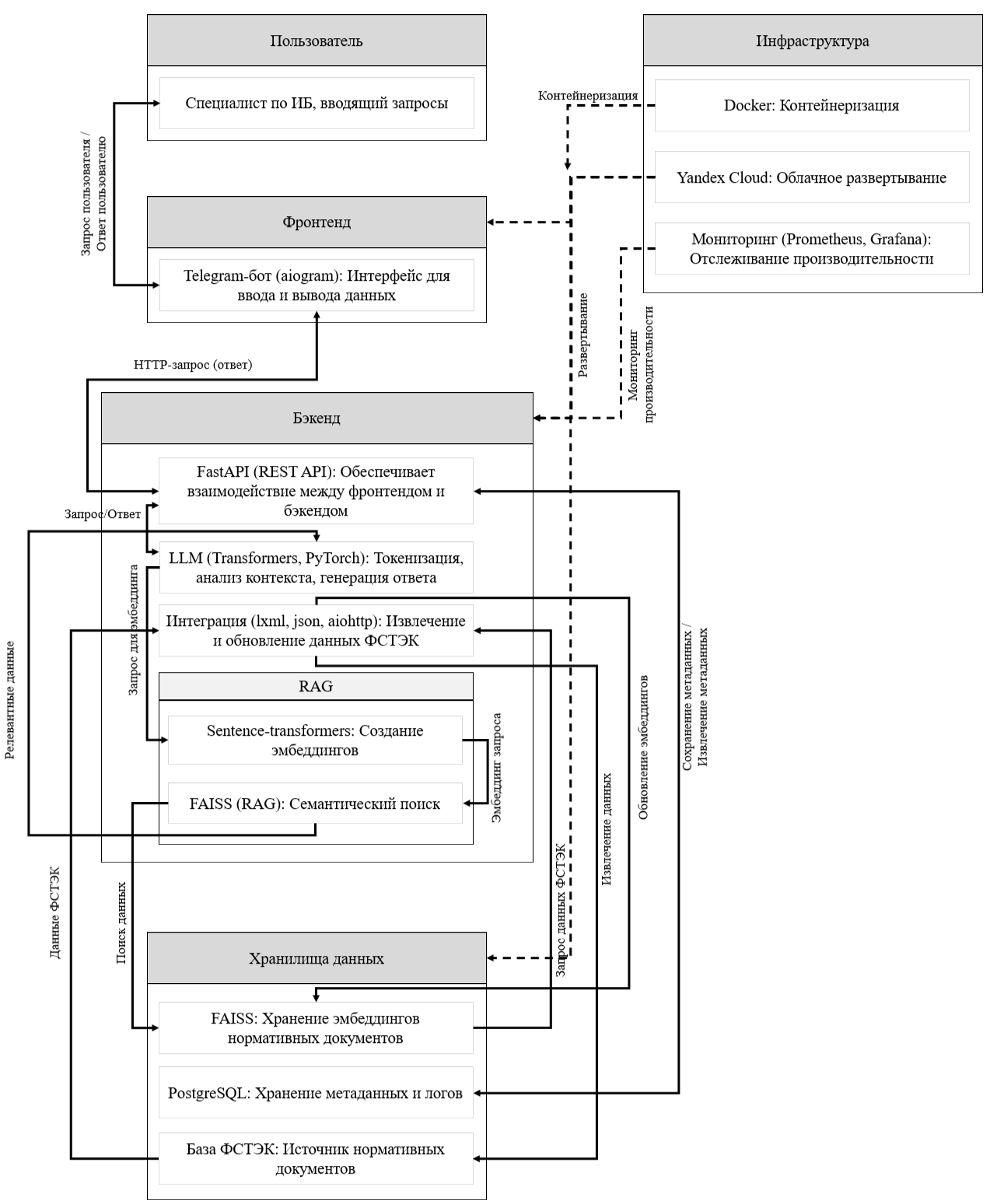


Рисунок 8 – Блок-схема архитектуры решений

### 2.2.3 Стек технологий

Разработка интеллектуального Telegram-бота для анализа угроз и рисков в сфере информационной безопасности (ИБ) с правовой оценкой на основе базы данных ФСТЭК требует формирования технологического стека, который обеспечивает высокую производительность, гибкость, адаптацию к российским нормативным требованиям и минимизацию человеческого фактора.

Технологический стек включает язык программирования, библиотеки, фреймворки, хранилища данных, инструменты интеграции и мониторинга, выбранные с учетом задач проекта, описанных в разделе 1.1, и выводов из анализа существующих подходов в разделе 1.2.

Основной целью является создание системы, способной эффективно обрабатывать естественно-языковые запросы, интегрироваться с базой ФСТЭК, обеспечивать актуальность данных и предоставлять удобный интерфейс для специалистов.

Выбор технологий ориентирован на достижение высокой скорости обработки запросов, масштабируемости и соответствия стандартам, таким как ГОСТ Р 57580 и ФЗ-187. Ниже представлено подробное описание компонентов технологического стека, их функций и обоснование их выбора.

Язык программирования

Основным языком программирования для реализации системы выбран Python. Этот выбор обусловлен его универсальностью, широкой поддержкой сообщества и наличием обширной экосистемы библиотек для работы с искусственным интеллектом, обработки данных и интеграции с внешними API. Python позволяет реализовать все ключевые компоненты системы: от обработки запросов на основе больших языковых моделей до интеграции с базой данных ФСТЭК. Его синтаксическая простота ускоряет разработку и снижает вероятность ошибок, что особенно важно в условиях дефицита квалифицированных кадров, отмеченного в разделе 1.1. Python поддерживает асинхронное программирование, что обеспечивает высокую производительность при обработке множества запросов через Telegram-бот.

Кроме того, Python широко используется в российских проектах, связанных с анализом данных и ИИ, таких как системы обработки текстов в «Сбере» или «Яндексе», что подтверждает его применимость для задач проекта. Альтернативные языки, такие как JavaScript и Java, были рассмотрены, но отклонены. JavaScript, несмотря на свою популярность в веб-разработке, имеет ограниченную поддержку библиотек для работы с LLM и векторным поиском, а Java требует более сложной настройки для интеграции с современными ИИ-фреймворками. Python, напротив, предоставляет доступ к библиотекам, таким как Transformers и FAISS, что делает его оптимальным выбором для реализации гибридного подхода LLM+RAG.

Библиотеки и фреймворки

Для реализации Telegram-бота используется библиотека *aiogram*, основанная на Python. *Aiogram* поддерживает асинхронное программирование, что позволяет эффективно обрабатывать запросы от множества пользователей одновременно. Библиотека предоставляет удобный интерфейс для работы с Telegram API, поддерживает интерактивные элементы, такие как кнопки и меню, и позволяет реализовать структурированный вывод данных, например, таблицы с классификацией угроз или списки рекомендаций.

По сравнению с альтернативой, такой как *python-telegram-bot*, *aiogram* обеспечивает более высокую производительность благодаря асинхронной архитектуре и активной поддержке сообщества. Для работы с большими языковыми моделями используется библиотека *Transformers* от Hugging Face. Эта библиотека предоставляет инструменты для токенизации, создания эмбеддингов и работы с русскоязычными моделями, такими как GigaChat или адаптированные версии BERT. *Transformers* поддерживает интеграцию с PyTorch, что обеспечивает гибкость в настройке моделей и оптимизации их производительности. Векторный поиск реализован с использованием библиотеки *FAISS* (Facebook AI Similarity Search).

Для обеспечения надежности и эффективности работы системы применяются инструменты контейнеризации, такие как *Docker*, которые позволяют стандартизировать среду разработки и эксплуатации. *Docker* обеспечивает изоляцию компонентов системы, упрощая развертывание на различных платформах, будь то локальный сервер или облачная инфраструктура, такая как Yandex Cloud.

Контейнеризация минимизирует конфликты зависимостей, что особенно важно при использовании множества библиотек, таких как *Transformers*, *FAISS* и *SQLAlchemy*. Например, модуль обработки запросов на базе LLM может быть упакован в отдельный контейнер, что упрощает его обновление без влияния на другие компоненты. Для оркестрации контейнеров рассматривается использование *Kubernetes*, который позволяет автоматически масштабировать систему при увеличении числа пользователей. *Kubernetes* поддерживает балансировку нагрузки и автоматическое восстановление при сбоях, что критически важно для организаций, работающих с критической информационной инфраструктурой. По сравнению с альтернативой, такой как *Docker Swarm*, *Kubernetes* предоставляет более мощные инструменты для управления кластерами, что делает его предпочтительным для масштабируемых систем.

Для обработки больших языковых моделей и оптимизации их производительности используется фреймворк *PyTorch*, который обеспечивает гибкость в настройке моделей и поддерживает аппаратное ускорение на GPU. *PyTorch* был выбран вместо *TensorFlow* благодаря его динамической вычислительной графической архитектуре, которая упрощает отладку и эксперименты с моделями.

Например, при настройке русскоязычной модели, такой как GigaChat, *PyTorch* позволяет быстро адаптировать параметры модели под специфические задачи, такие как обработка нормативных документов ФСТЭК. Кроме того, *PyTorch* интегрируется с *Transformers*, что обеспечивает бесшовную работу с эмбеддингами и токенизацией. Для ускорения вычислений рассматривается использование библиотек, таких как *CUDA* от NVIDIA, если система развертывается на серверах с GPU. Это позволяет сократить время обработки сложных запросов, что особенно важно для организаций с высоким объемом запросов, таких как операторы КИИ.

Для обеспечения безопасности системы применяются инструменты шифрования и аутентификации. Передача данных между Telegram-ботом и сервером защищается с использованием протокола HTTPS, реализованного через *FastAPI* и сертификаты SSL/TLS. Для защиты данных пользователей, таких как история запросов, используется библиотека *cryptography* в Python, которая обеспечивает шифрование данных перед их сохранением в *PostgreSQL*. Аутентификация пользователей осуществляется через Telegram API, который предоставляет уникальные идентификаторы для каждого пользователя, что позволяет ограничить доступ к системе только авторизованным лицам. Например, организация может настроить доступ так, чтобы только сотрудники отдела ИБ могли использовать бот.

Для дополнительной безопасности рассматривается интеграция с российскими системами управления доступом, такими как *КриптоПро*, которые соответствуют требованиям ФСБ. Это обеспечивает защиту от несанкционированного доступа и утечек данных, что критически важно для работы с нормативной базой ФСТЭК.

Для обработки больших объемов нормативных документов ФСТЭК применяются инструменты предварительной обработки данных. Библиотека *lxml* используется для парсинга XML-документов, которые часто применяются в базах ФСТЭК, а *json* — для работы с JSON-форматом. Эти библиотеки обеспечивают высокую скорость и точность извлечения данных, таких как описания угроз, категории рисков и рекомендации по мерам защиты. Для структурирования данных перед индексацией в *FAISS* используется *Pandas*, которая позволяет фильтровать, группировать и очищать данные.

Например, при обработке документа ФСТЭК *Pandas* может выделить только те записи, которые относятся к угрозам АСУ ТП, что сокращает объем данных для индексации и повышает скорость поиска. Для автоматизации этого процесса разработан конвейер обработки данных, который включает этапы парсинга, очистки, преобразования в эмбеддинги и индексации. Этот конвейер управляется с использованием библиотеки *Luigi* или *Airflow*, которые позволяют планировать и мониторить задачи обновления данных. По сравнению с *Luigi*, *Airflow* предоставляет более мощные инструменты для визуализации рабочих процессов, что делает его предпочтительным для сложных систем.

Для тестирования и отладки системы применяются инструменты, такие как *pytest* и *unittest*, которые обеспечивают покрытие кода тестами и минимизируют вероятность ошибок. Например, тесты для модуля обработки запросов проверяют корректность токенизации и генерации эмбеддингов, а тесты для Telegram-бота — правильность обработки пользовательских команд. Для нагрузочного тестирования используется *Locust*, который позволяет симулировать запросы от множества пользователей и оценивать производительность системы.

Например, *Locust* может проверить, как система справляется с 100 одновременными запросами, что важно для оценки масштабируемости. Результаты тестирования записываются в логи с использованием *logging*, что позволяет анализировать производительность и выявлять узкие места. Для документирования кода используется *Sphinx*, который генерирует документацию в формате HTML или PDF, что упрощает передачу проекта другим разработчикам или заказчикам.

Для поддержки разработки и совместной работы используется система контроля версий *Git*, а репозиторий размещается на платформе, такой как *GitLab* или *Bitbucket*. *GitLab* предоставляет инструменты для CI/CD, что позволяет автоматизировать сборку, тестирование и развертывание системы. Например, при внесении изменений в код *GitLab CI* автоматически запускает тесты и развертывает обновленную версию в тестовой среде. Это ускоряет процесс разработки и обеспечивает стабильность системы. Для управления зависимостями используется *Poetry*, которое предоставляет более удобный интерфейс по сравнению с *pip* и позволяет избежать конфликтов версий библиотек. Например, *Poetry* гарантирует, что версии *aiogram* и *FastAPI* совместимы, что минимизирует риски сбоев.

Выбранный технологический стек обеспечивает баланс между производительностью, гибкостью и соответствием российским требованиям. Python предоставляет универсальную платформу для реализации всех компонентов системы, от обработки запросов до интеграции с базой ФСТЭК. *Aiogram*, *FastAPI* и *Transformers* обеспечивают высокопроизводительный интерфейс и обработку текстов, а *FAISS* и *PostgreSQL* — надежное хранение и поиск данных. *Docker* и *Kubernetes* поддерживают масштабируемость и стандартизацию развертывания, а *Prometheus* и *logging* — мониторинг и прозрачность. Инструменты безопасности, такие как *cryptography* и HTTPS, защищают данные, а *Airflow* и *Pandas* автоматизируют обработку нормативных документов.

Этот стек позволяет создать систему, которая решает ключевые проблемы ручного анализа угроз, выявленные в разделе 1.1, такие как высокие временные затраты и вероятность ошибок, и превосходит ограничения существующих подходов, описанных в разделе 1.2. Технологии подобраны с учетом их совместимости, что обеспечивает целостность системы и возможность ее дальнейшего развития, например, интеграции с SIEM или другими базами данных, такими как MITRE ATT&CK.

Фронтенд и бэкенд

Реализация интеллектуального Telegram-бота для анализа угроз и рисков в сфере информационной безопасности с правовой оценкой на основе базы данных ФСТЭК требует четкого разделения функциональности на фронтенд и бэкенд, чтобы обеспечить удобство взаимодействия для пользователей и высокую производительность обработки данных.

Фронтенд отвечает за пользовательский интерфейс, через который специалисты по информационной безопасности вводят запросы и получают результаты, тогда как бэкенд обеспечивает обработку этих запросов, интеграцию с базой данных ФСТЭК и генерацию структурированных ответов. Архитектура системы, описанная в предыдущих пунктах, предполагает использование гибридного подхода, сочетающего большие языковые модели и технологию генерации с подкреплением выборкой, что накладывает специфические требования на реализацию фронтенда и бэкенда.

Эти требования включают поддержку русскоязычных текстов, обеспечение быстрого отклика, соответствие нормативным стандартам, таким как ГОСТ Р 57580 и ФЗ-187, и возможность масштабирования для обработки запросов от множества пользователей. Ниже представлено подробное описание фронтенда и бэкенда, их функций, технологий и особенностей реализации, которые обеспечивают достижение целей проекта, сформулированных в разделе 1.1, и преодоление ограничений существующих подходов, выявленных в разделе 1.2.

Фронтенд.

Пользовательский интерфейс системы реализован в виде Telegram-бота, что обеспечивает доступность, простоту использования и кроссплатформенность. Telegram был выбран как платформа для фронтенда благодаря его популярности в России, наличию мощного API и поддержке интерактивных элементов, таких как кнопки и меню.

Основной задачей фронтенда является прием естественно-языковых запросов от пользователей и предоставление структурированных ответов, содержащих информацию об угрозах, рисках и рекомендациях по мерам защиты. Например, пользователь может ввести запрос «Какие угрозы связаны с уязвимостями веб-приложений?» и получить ответ в виде текста с описанием угроз, таблицы с классификацией по ГОСТ Р 57580 и списка нормативных документов ФСТЭК.

Telegram-бот реализован с использованием библиотеки *aiogram* на Python, которая поддерживает асинхронную обработку запросов, что позволяет одновременно обслуживать множество пользователей без задержек. *Aiogram* обеспечивает гибкость в создании пользовательских интерфейсов, позволяя добавлять интерактивные кнопки для выбора категорий угроз, таких как «Утечка данных», «Несанкционированный доступ» или «Атаки на АСУ ТП». Это упрощает навигацию для специалистов с минимальным опытом работы с ИИ-системами.

Например, при выборе категории «Утечка данных» бот может предложить дополнительные опции, такие как «Анализ угроз» или «Рекомендации по защите», что делает взаимодействие интуитивным. Ответы бота форматируются для удобства восприятия: текст дополняется таблицами, списками или гиперссылками на нормативные документы, если они доступны через API ФСТЭК.

Для обеспечения безопасности взаимодействия Telegram-бот использует встроенные механизмы Telegram API, такие как уникальные идентификаторы пользователей, что позволяет ограничить доступ к системе только авторизованным лицам.

Например, организация может настроить бот так, чтобы он был доступен только сотрудникам отдела ИБ. Кроме того, фронтенд поддерживает мультиязычность с приоритетом на русский язык, что соответствует требованиям обработки русскоязычных нормативных документов ФСТЭК.

В отличие от веб-интерфейсов, которые требуют разработки сложных клиентских приложений и поддержки различных браузеров, Telegram-бот не требует установки дополнительного ПО и доступен на мобильных и десктопных устройствах, что снижает затраты на внедрение и обучение пользователей. По сравнению с другими платформами, такими как WhatsApp или Slack, Telegram обеспечивает более гибкий API и меньшие ограничения на количество запросов, что делает его оптимальным выбором для корпоративного использования в России.

Бэкенд

Бэкенд системы отвечает за обработку запросов, интеграцию с базой данных ФСТЭК, выполнение семантического поиска и генерацию ответов. Он состоит из нескольких модулей, каждый из которых выполняет специализированные функции, обеспечивая высокую производительность и надежность. Основным компонентом бэкенда является модуль обработки запросов, реализованный на базе большой языковой модели, такой как GigaChat или адаптированная русскоязычная версия BERT.

Этот модуль выполняет токенизацию запросов, преобразование их в эмбеддинги и анализ контекста, что позволяет корректно интерпретировать сложные формулировки, характерные для профессиональной терминологии ИБ. Например, запрос «Как защитить систему от атак типа SQL-инъекций?» обрабатывается с учетом контекста и специфики нормативных требований ФСТЭК.

Для реализации этого модуля используется библиотека *Transformers* от Hugging Face, которая поддерживает работу с русскоязычными моделями и интеграцию с *PyTorch* для оптимизации производительности. Модуль работает в связке с модулем семантического поиска, который использует технологию RAG для извлечения релевантной информации из базы ФСТЭК. Модуль RAG преобразует запрос пользователя в векторное представление с помощью трансформерной модели, например *sentence-transformers*, и выполняет поиск по векторному хранилищу *FAISS*.

Это позволяет находить документы, наиболее близкие по смыслу к запросу, даже если они не содержат точных ключевых слов. Например, запрос об «утечке данных» может вернуть документы, связанные с «несанкционированным доступом» или «компрометацией учетных записей», что повышает точность поиска. Модуль интеграции с базой ФСТЭК отвечает за парсинг данных в формате XML или JSON и их преобразование в эмбеддинги для индексации в *FAISS*. Если ФСТЭК предоставляет API, модуль использует *aiohttp* для асинхронных HTTP-запросов, что обеспечивает автоматическое обновление данных при появлении новых угроз.

Например, при добавлении новой записи в базу ФСТЭК модуль парсит документ, извлекает описание угрозы и рекомендации, преобразует их в эмбеддинг и обновляет индекс. Этот процесс автоматизирован с использованием *Airflow*, который планирует задачи обновления данных. Бэкенд также включает сервер, реализованный на *FastAPI*, который обеспечивает REST API для взаимодействия между фронтендом и модулями обработки данных. *FastAPI* поддерживает асинхронную обработку, что позволяет обрабатывать множество запросов одновременно, и интегрируется с *aiogram* для передачи данных в Telegram-бот.

Для хранения метаданных, таких как история запросов и логи, используется реляционная база данных *PostgreSQL*, которая обеспечивает высокую надежность и поддержку сложных запросов. Например, *PostgreSQL* может хранить информацию о том, какие угрозы чаще всего запрашиваются пользователями, что позволяет анализировать тренды и оптимизировать систему.

Для обеспечения безопасности бэкенд использует шифрование данных с помощью библиотеки *cryptography* и HTTPS для защиты передачи данных между компонентами. Развертывание бэкенда осуществляется в облаке, например Yandex Cloud, что обеспечивает масштабируемость и соответствие требованиям ФЗ-152. Альтернативно, система может быть развернута на локальном сервере для организаций, предпочитающих автономные решения.

Бэкенд также поддерживает мониторинг с использованием *Prometheus* и *Grafana*, что позволяет отслеживать производительность, например время обработки запросов или нагрузку на сервер. Это обеспечивает прозрачность работы системы и возможность оперативного реагирования на сбои.

Таким образом, фронтенд и бэкенд системы формируют целостное решение, которое сочетает удобство взаимодействия через Telegram-бот с мощной обработкой данных на бэкенде. Фронтенд, реализованный с использованием *aiogram*, обеспечивает интуитивный интерфейс и кроссплатформенный доступ, а бэкенд, включающий модули LLM, RAG, интеграции и хранения данных, гарантирует точность, актуальность и производительность.

Эти компоненты позволяют системе решать проблемы ручного анализа угроз, выявленные в разделе 1.1, такие как высокие временные затраты и вероятность ошибок, и преодолевать ограничения существующих подходов, описанных в разделе 1.2, обеспечивая автоматизацию и соответствие российским стандартам.

Блок-схема стека технологий приведена на рисунке 9.

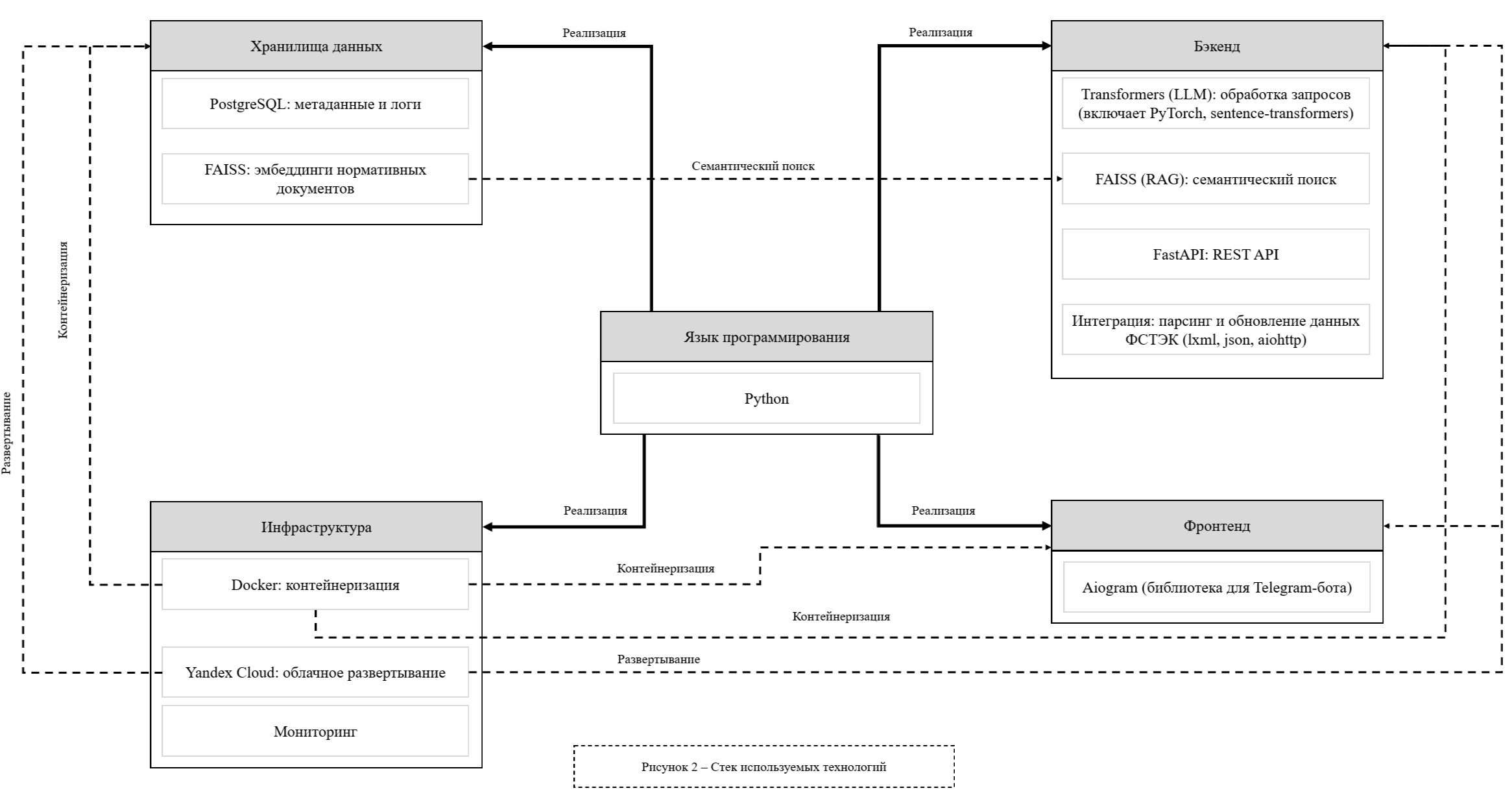


Рисунок 9 – Стек используемых технологий

### 2.2.4 Вывод по подразделу

Разработка интеллектуального Telegram-бота для автоматизированного анализа угроз и рисков в сфере информационной безопасности с правовой оценкой на основе базы данных ФСТЭК требует тщательно продуманного технологического стека и архитектуры, которые обеспечивают высокую производительность, актуальность данных и соответствие российским нормативным требованиям. В данном подразделе были обоснованы ключевые технологические решения, определена архитектура системы, описаны используемые технологии и их роли, а также представлена визуализация взаимодействия компонентов.

Основной целью было создание системы, которая минимизирует временные затраты на анализ угроз, снижает вероятность ошибок, вызванных человеческим фактором, и предоставляет удобный интерфейс для специалистов, что решает проблемы, выявленные в разделе 1.1, и превосходит ограничения существующих подходов, описанных в разделе 1.2. Выбор Python в качестве основного языка программирования обусловлен его универсальностью, поддержкой обширной экосистемы библиотек и способностью эффективно обрабатывать русскоязычные тексты, что критически важно для работы с нормативными документами ФСТЭК.

Библиотеки *aiogram*, *Transformers*, *FAISS* и *SQLAlchemy* обеспечивают реализацию ключевых функций: от создания интерактивного Telegram-бота до выполнения семантического поиска и управления данными. *FastAPI* предоставляет высокопроизводительный сервер для взаимодействия компонентов, а *PyTorch* поддерживает гибкую настройку языковых моделей.

Использование векторного хранилища *FAISS* для семантического поиска и реляционной базы данных *PostgreSQL* для хранения метаданных и логов гарантирует актуальность данных и надежность системы. Развертывание в облаке Yandex Cloud или на локальных серверах обеспечивает масштабируемость и соответствие требованиям ФЗ-152, что делает систему применимой для организаций различного масштаба, от малого бизнеса до операторов критической информационной инфраструктуры. Фронтенд, реализованный через Telegram-бот, предоставляет интуитивно понятный интерфейс, доступный на мобильных и десктопных устройствах, что снижает барьеры для использования и не требует дополнительного обучения специалистов.

Бэкенд, включающий модули обработки запросов, семантического поиска и интеграции с базой ФСТЭК, обеспечивает быструю и точную обработку запросов, что позволяет сократить время анализа угроз с часов до секунд или минут. Архитектура системы, представленная в виде структурно-функциональной схемы, наглядно демонстрирует поток данных от ввода запроса до генерации ответа, подчеркивая целостность и эффективность решения.

Однако система имеет определенные ограничения, которые необходимо учитывать при внедрении. Во-первых, качество и полнота данных в базе ФСТЭК напрямую влияют на точность ответов, что требует регулярного обновления данных и контроля их целостности. Во-вторых, использование больших языковых моделей и векторного поиска предполагает значительные вычислительные ресурсы, что может быть затруднительно для организаций с ограниченной инфраструктурой. В-третьих, поддержание актуальности векторного хранилища требует автоматизированного конвейера обработки данных, что добавляет сложность в эксплуатацию.

Несмотря на эти ограничения, выбранный технологический стек и архитектура обеспечивают значительные преимущества по сравнению с традиционными подходами, такими как ручной анализ или использование систем SIEM. Перспективы развития системы включают возможность интеграции с дополнительными источниками данных, такими как база MITRE ATT&CK, что позволит расширить функциональность для анализа международных стандартов.

Также рассматривается интеграция с системами SIEM для автоматического анализа логов и корреляции событий, что может повысить точность выявления угроз. Внедрение дополнительных функций, таких как анализ трендов запросов или автоматическая генерация отчетов, может сделать систему более ценной для организаций.

Таким образом, предложенный технологический стек и архитектура обеспечивают достижение целей проекта, создавая надежное, масштабируемое и удобное решение для автоматизации анализа угроз и рисков в сфере информационной безопасности. Система соответствует российским нормативным требованиям и готова к внедрению в организациях, стремящихся повысить эффективность защиты своих информационных систем.

# 2.3 Описание программной разработки

Данный раздел представляет концептуальное описание интеллектуального бота, основанного на архитектуре Retrieval-Augmented Generation (RAG) с интеграцией базы данных угроз ФСТЭК России. Концепция направлена на создание теоретической модели системы для анализа угроз и рисков в области информационной безопасности (ИБ) с возможностью правовой оценки.

Разрабатываемая концепция предполагает построение модульной системы, способной:

1. Обеспечивать эффективный поиск и обработку информации из разнородных источников;
2. Генерировать аналитические выводы с высокой степенью релевантности;
3. Автоматизировать процессы сопоставления данных об угрозах с нормативными требованиями.

### 2.3.1 Требования к программному обеспечению

Перед началом программной разработки были сформулированы функциональные и нефункциональные требования к системе, обеспечивающие достижение поставленных целей проекта.

1. Функциональные требования

Функциональные требования определяют, что именно должна делать система. Для разрабатываемого бота были определены следующие ключевые функции:

1. Идентификация и описание угроз безопасности информации. Система должна быть способна по запросу пользователя идентифицировать и предоставить детальное описание угроз безопасности информации, основываясь на данных из базы угроз ФСТЭК России и других актуальных источников;
2. Выявление связанных уязвимостей. Система должна определять и перечислять уязвимости (с указанием идентификаторов CVE, BDU и др.), которые могут быть использованы для реализации идентифицированных угроз;
3. Оценка потенциальных последствий. Бот должен оценивать и описывать потенциальные прямые и косвенные последствия реализации угроз для информационных систем и бизнес-процессов организации;
4. Формирование мер по противодействию. Система должна предлагать комплексные рекомендации по техническим и организационным мерам защиты, направленным на предотвращение или минимизацию воздействия угроз, включая ссылки на соответствующие стандарты и лучшие практики;
5. Правовая оценка и комплаенс. Бот должен предоставлять ссылки на релевантные нормативно-правовые акты Российской Федерации (ФЗ, Приказы ФСТЭК/ФСБ, ГОСТы), разъяснять правовые последствия несоблюдения требований ИБ и давать рекомендации по обеспечению соответствия законодательству;
6. Обработка сложных запросов на естественном языке. Система должна понимать и корректно обрабатывать многокомпонентные и детализированные запросы пользователя, заданные на естественном языке;
7. Актуализация базы знаний. Должен быть предусмотрен механизм регулярного обновления и пополнения базы знаний новой информацией об угрозах, уязвимостях и изменениях в законодательстве;
8. Интерфейс взаимодействия. Система должна предоставлять удобный и интуитивно понятный интерфейс для взаимодействия с пользователем (например, через веб-приложение или API).
9. Нефункциональные требования

Нефункциональные требования описывают, насколько хорошо система должна выполнять свои функции.

1. Производительность. Время ответа на запрос пользователя не должно превышать 5 секунд для стандартных запросов. Поиск в векторной базе данных должен осуществляться максимально быстро;
2. Масштабируемость. Система должна быть способна обрабатывать растущее количество запросов и увеличивающийся объем базы знаний без существенной деградации производительности;
3. Надежность. Система должна обладать высокой отказоустойчивостью, минимизируя вероятность сбоев и потери данных;
4. Безопасность. Доступ к системе и данным должен быть защищен от несанкционированного доступа. Должны быть предусмотрены меры по защите конфиденциальности обрабатываемых данных;
5. Удобство использования. Пользовательский интерфейс должен быть интуитивно понятным и простым в освоении для специалистов по ИБ;
6. Сопровождаемость. Код системы должен быть хорошо документирован, модулен и легок для модификации и поддержки;
7. Портируемость. Система должна быть относительно независимой от конкретной аппаратной платформы и операционной системы.

### 2.3.2 Общая архитектура программного комплекса

Архитектура программного комплекса основывается на модульном подходе, что обеспечивает гибкость, масштабируемость и простоту обслуживания. Центральное место занимает архитектура Retrieval-Augmented Generation (RAG), которая позволяет комбинировать мощь генеративных способностей LLM с точностью и актуальностью информации, извлекаемой из специализированной базы знаний.

Поток данных при обработке запроса

В ходе нашего исследования мы не применяли такие технологии, как RAG (Retrieval-Augmented Generation), FAISS (Facebook AI Similarity Search) и GigaChat, однако на их примере можно наглядно продемонстрировать, как работает обработка запросов в современных NLP-системах.

Рассмотрим последовательность шагов, которые происходят с запросом пользователя от момента его ввода до получения ответа:

1. Запрос пользователя: Пользователь вводит свой запрос через интерфейс (например, "Какие уязвимости затрагивают СУБД PostgreSQL на Linux, связанные с нарушением целостности, и какие правовые последствия это влечет?");
2. Передача запроса RAG: Запрос поступает в центральный модуль;
3. Векторизация запроса приведена на рисунке 10. RAG Orchestrator передает запрос в Модуль векторизации запроса. Данный модуль, используя предобученную модель эмбеддингов, преобразует текст запроса в векторное представление;

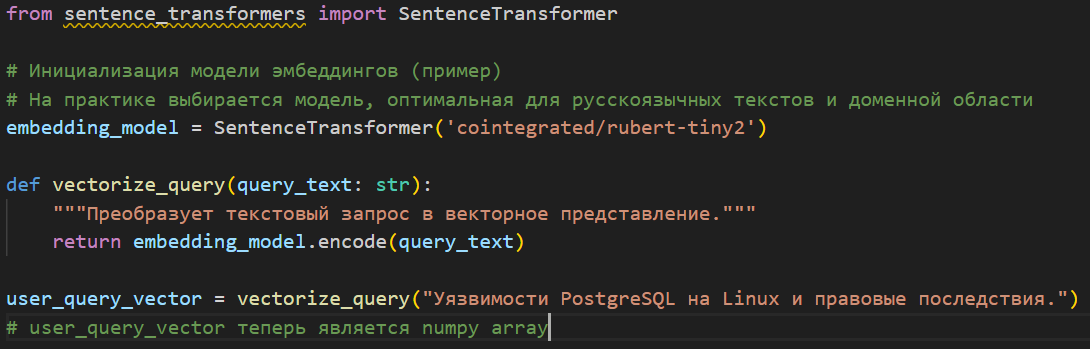


Рисунок 10 – Векторизация запроса

1. Поиск в FAISS: Полученный вектор запроса передается в Векторную базу данных FAISS, как показано на рисунке 11. FAISS выполняет высокоэффективный поиск ближайших соседей, находя текстовые чанки из базы знаний, чьи векторные представления наиболее схожи с вектором запроса;

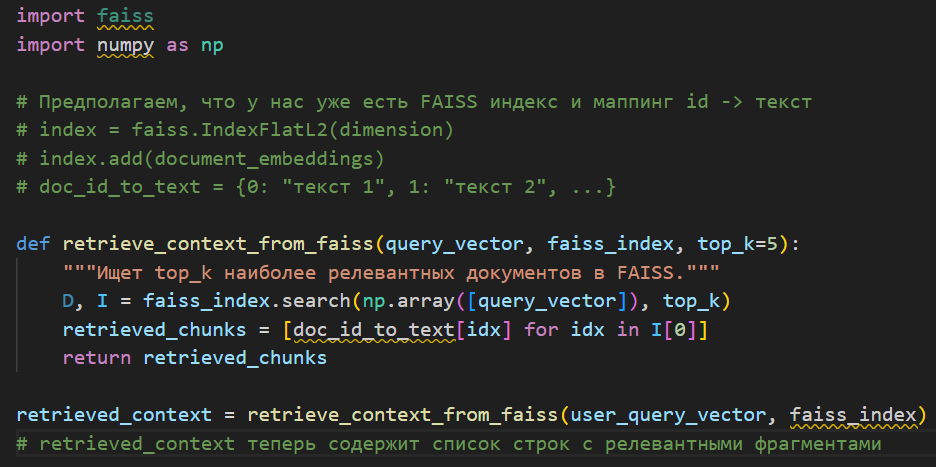
**

Рисунок 11 – Поиск FAISS

1. Формирование промпта для GigaChat: RAG Orchestrator объединяет исходный запрос пользователя и извлеченные релевантные текстовые фрагменты в единый промпт для GigaChat. Важно правильно структурировать этот промпт, чтобы GigaChat понимал, какую информацию использовать для генерации ответа;
2. Генерация ответа GigaChat: Сформированный промпт передается в GigaChat. GigaChat, используя свои обширные знания и предоставленный контекст, генерирует детальный и связный ответ;
3. Вывод ответа пользователю: Полученный от GigaChat ответ возвращается через RAG Orchestrator в Пользовательский интерфейс / API и отображается пользователю.

Модуль сбора и предобработки данных

От этого модуля зависит полнота и чистота базы знаний.

1. Источники данных:
   1. Нормативно-правовые акты: Файлы в форматах .docx, .pdf, .html (например, ФЗ-152, Приказ ФСТЭК №17, Приказ ФСБ №64 и др.);
   2. Базы уязвимостей (CVE, BDU): Данные из Национальной базы данных уязвимостей (NVD/CVE) и Банка данных угроз безопасности информации ФСТЭК России (БДУ). Извлекаются идентификаторы уязвимостей, их описание, связанное ПО, типы воздействия;
   3. Собственная база знаний: Возможность добавления экспертных заключений, внутренних регламентов, статей и аналитических материалов.
2. Процесс предобработки:
   1. Извлечение текста: Использование библиотек python-docx, PyPDF2, BeautifulSoup для извлечения чистого текста из документов;
   2. Очистка текста: Удаление лишних символов, форматирования, заголовков/колонтитулов, которые не несут смысловой нагрузки;
   3. Токенизация и лемматизация: Разбиение текста на слова (токены) и приведение их к нормальной форме (лемме) для улучшения качества поиска. Используются библиотеки NLTK или SpaCy;
   4. Разбиение на чанки (Chunking): Критический шаг для RAG. Документы разбиваются на более мелкие, осмысленные фрагменты. Размер чанка должен быть оптимальным: не слишком малым (для сохранения контекста) и не слишком большим (для эффективного поиска и обработки LLM);
   5. Добавление метаданных: К каждому чанку могут быть привязаны метаданные (например, источник, дата, тип документа, номер статьи НПА), что позволяет фильтровать и уточнять поиск.

Модуль векторизации (Embedding Model)

Данный модуль отвечает за преобразование предобработанных текстовых чанков и пользовательских запросов в числовые векторы. Выбор подходящей модели эмбеддингов крайне важен, поскольку от нее зависит семантическая точность поиска.

1. Выбор модели: для русскоязычного сегмента рекомендуется использовать модели Sentence Transformers, которые специально обучены для генерации семантически значимых эмбеддингов предложений. Например, cointegrated/rubert-tiny2, sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 или более крупные модели, если позволяют ресурсы;
2. Принцип работы: Каждая модель эмбеддингов представляет собой нейронную сеть, которая принимает текст на вход и выдает вектор фиксированной размерности (например, 512 или 768). Близость этих векторов в многомерном пространстве указывает на семантическую схожесть исходных текстов;
3. Использование: Модель инициализируется один раз и затем используется для векторизации всех текстовых чанков при индексации, а также для векторизации каждого входящего запроса пользователя.

Векторная база данных FAISS

FAISS (Facebook AI Similarity Search) – это библиотека для эффективного поиска по сходству в больших наборах векторов. Она оптимизирована для работы с миллионами и миллиардами векторов, что делает ее идеальным выбором для нашего проекта.

1. Типы индексов: FAISS предлагает различные типы индексов, каждый из которых имеет свои преимущества и недостатки по скорости и точности:
   1. IndexFlatL2: Простейший индекс, выполняет полный перебор. Самый точный, но медленный для больших объемов;
   2. IndexIVFFlat: Инвертированный файловый индекс. Разделяет векторы на кластеры, что ускоряет поиск за счет прохождения только по релевантным кластерам. Хороший баланс между скоростью и точностью;
   3. IndexHNSWFlat: Использует иерархический навигационный поиск по мелким мирам (Hierarchical Navigable Small World). Очень быстрый, но требует больше памяти;
   4. Выбор конкретного индекса будет зависеть от объема данных и требований к производительности. Для начала можно использовать IndexFlatL2 для небольших тестовых наборов, а затем перейти на IndexIVFFlat или IndexHNSWFlat.
2. Процесс индексации:
   1. Все текстовые чанки из базы знаний векторизуются;
   2. Полученные векторы добавляются в индекс FAISS;
   3. Для каждого вектора сохраняется соответствие его ID с оригинальным текстовым чанком (например, в словаре или отдельной базе данных), как показано на рисунке 12.

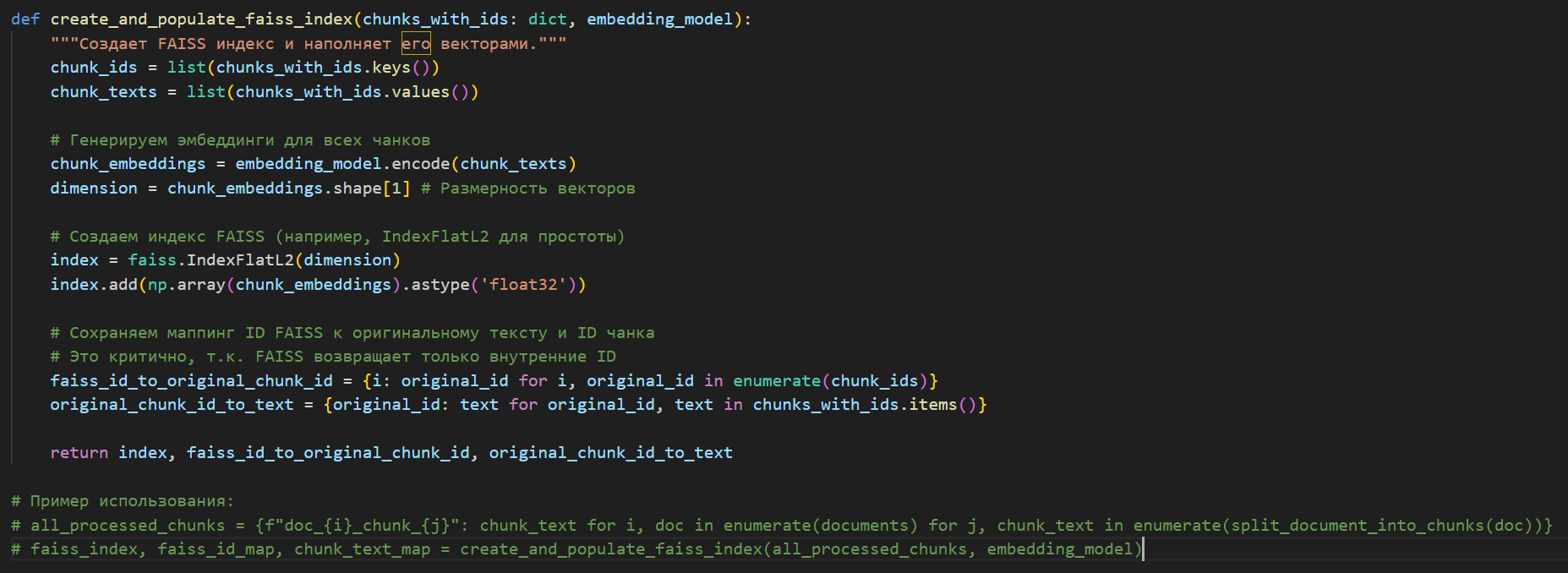
**

Рисунок 12 – Пример создания и добавления в FAISS

1. Процесс поиска: при получении векторного представления запроса, FAISS быстро находит N ближайших соседей (N — параметр top\_k), возвращая их ID. Затем по этим ID извлекаются оригинальные текстовые чанки.

Большая языковая модель GigaChat

GigaChat от Сбера, как мощная генеративная языковая модель, является ядром генерации ответов в системе RAG.

Особенности GigaChat:

* 1. Русскоязычная ориентация: GigaChat разработан с учетом особенностей русского языка, что обеспечивает высокое качество генерации и понимания русскоязычных запросов и контекста;
  2. Разнообразие моделей: Сбер предлагает различные версии GigaChat (GigaChat-mini, GigaChat-pro), позволяющие выбрать оптимальный баланс между производительностью, качеством и затратами. Для нашего проекта, в зависимости от объема и сложности ожидаемых ответов, может быть выбрана соответствующая версия;
  3. API-доступ: GigaChat предоставляет API для интеграции, что позволяет легко взаимодействовать с ним из Python-приложений.

Использование в RAG:

* 1. GigaChat не "знает" о новых или специфичных данных, которые не были включены в его тренировочный набор. RAG-архитектура решает эту проблему, предоставляя GigaChat актуальный и релевантный контекст, извлеченный из специализированной базы знаний.
  2. Промпт для GigaChat формируется таким образом, чтобы четко указать модели, что она должна использовать предоставленный контекст для ответа на вопрос, как показано на рисунке 13. Это помогает минимизировать "галлюцинации" и обеспечить фактическую точность.

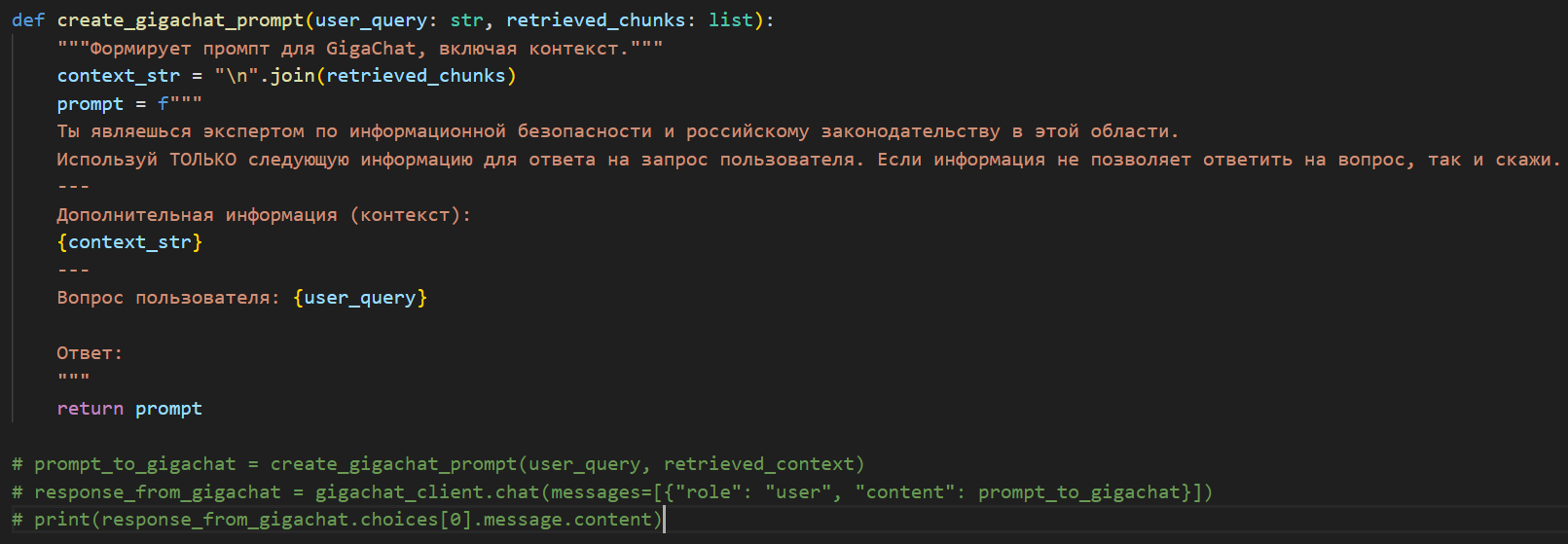


Рисунок 13 – Промт

Оптимизация взаимодействия: для повышения производительности и снижения задержек, запросы к GigaChat могут быть асинхронными, а также могут быть использованы механизмы кэширования для часто повторяющихся запросов.

Модуль RAG Orchestrator

Этот модуль является "мозгом" системы, координирующим взаимодействие всех остальных компонентов. Он отвечает за:

1. Прием и валидацию запросов: Получение запроса от пользователя, проверка его формата и размера;
2. Управление пайплайном: Последовательный вызов модуля векторизации, FAISS для поиска, и GigaChat для генерации;
3. Обработка ошибок: Механизмы для обработки исключений, возникающих на любом этапе процесса;
4. Логирование: Запись всех операций и ошибок для отладки и мониторинга;
5. Формирование итогового ответа: Сборка полученного от GigaChat ответа и его форматирование для вывода пользователю.

Данные в базе знаний логически разделены по категориям, но физически хранятся в векторизованном виде в FAISS. Для сопоставления векторов с исходными текстовыми фрагментами и их метаданными (источник, дата, тип) используется дополнительная реляционная или NoSQL база данных (например, PostgreSQL или MongoDB).

Процесс наполнения и актуализации

1. Первичное наполнение:
   1. Последовательная обработка каждого документа: извлечение текста, очистка, разбиение на чанки;
   2. Векторизация всех чанков;
   3. Индексация векторов в FAISS и сохранение метаданных в сопроводительной базе данных, показано на рисунке 14.

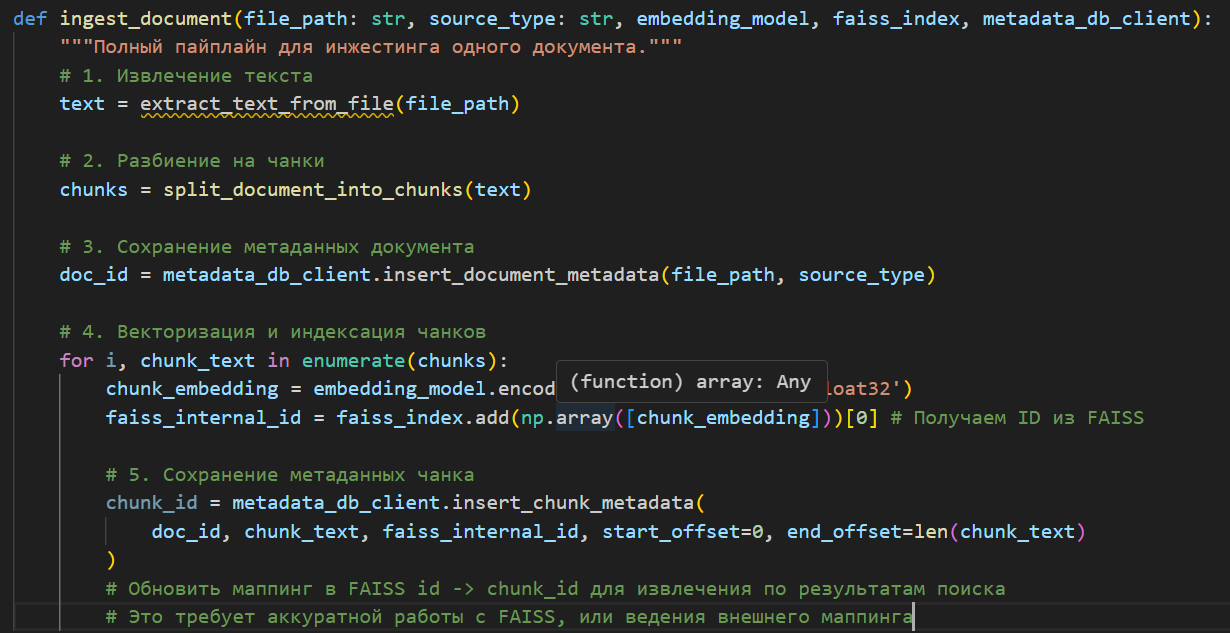
**

Рисунок 14 – Индексация векторов

1. Регулярная актуализация:
   1. Мониторинг официальных источников (ФСТЭК, правовые порталы) на предмет выхода новых документов или обновлений существующих;
   2. Разработка механизмов инкрементального обновления: добавление новых документов, обновление измененных, удаление устаревших. Это может включать переиндексацию только затронутых чанков в FAISS;
   3. Автоматизированные скрипты для запуска процесса актуализации по расписанию (например, с использованием cron или оркестраторов типа Apache Airflow).

Пользовательский интерфейс и API

Хотя основной фокус делается на бэкенд-логике, для взаимодействия с ботом необходим удобный интерфейс, как показано на рисунках 15 и 16.

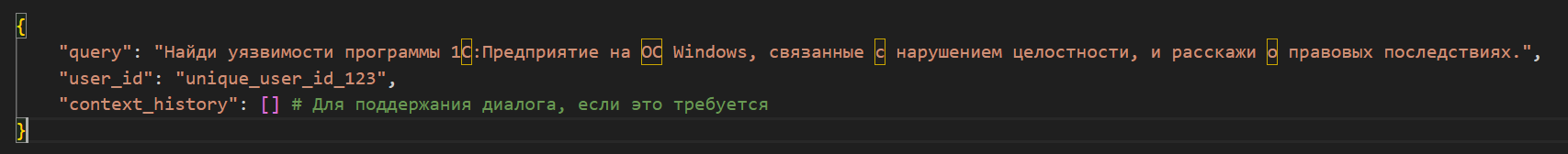


Рисунок 15 – Пример структуры API-запроса

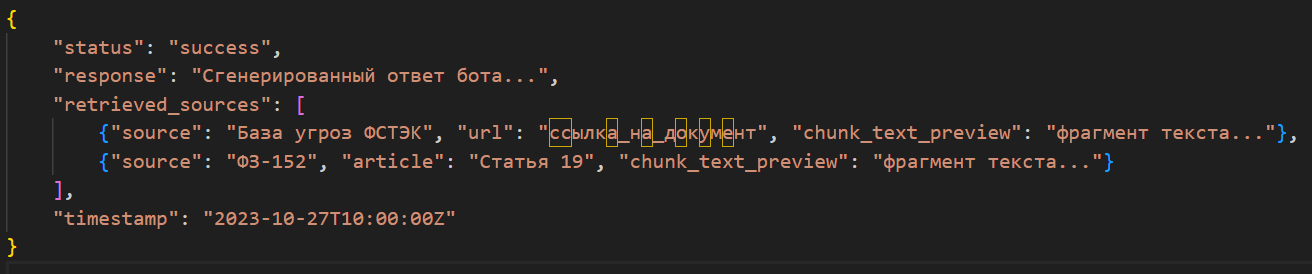


Рисунок 16 – Пример структуры API-ответа

Тестирование и оценка качества

Для обеспечения высокого качества разрабатываемого программного обеспечения будет проведен комплексный набор тестов.

1. Модульное тестирование: Тестирование отдельных компонентов (парсеров, функций векторизации, взаимодействия с FAISS, формирования промптов) для проверки их корректной работы в изоляции;
2. Интеграционное тестирование: Проверка взаимодействия между различными модулями системы (например, от модуля предобработки до FAISS);
3. Функциональное тестирование: Проверка соответствия разработанного функционала заявленным функциональным требованиям. Тестирование сценариев использования бота по запросам пользователей.

Пример тест-кейса:

* + 1. Вход: Запрос "Какие угрозы связаны с использованием устаревшего ПО Apache Tomcat на Windows Server и какие статьи УК РФ могут быть применимы?";
    2. Ожидаемый результат: Бот должен предоставить:
       1. Список угроз (например, выполнение произвольного кода, повышение привилегий) из базы ФСТЭК, связанных с Apache Tomcat и Windows Server;
       2. Идентификаторы уязвимостей (CVE), применимые к устаревшим версиям;
       3. Меры по митигации (обновление ПО, патчи);
       4. Ссылки на соответствующие статьи УК РФ (например, Статья 272 "Неправомерный доступ к компьютерной информации", Статья 273 "Создание, использование и распространение вредоносных компьютерных программ") и объяснение их применимости.

1. Тестирование производительности: Оценка времени ответа системы под различными нагрузками, масштабируемости FAISS-индекса;
2. Оценка качества генерации LLM:
   1. Релевантность: насколько извлеченный контекст соответствует запросу;
   2. Точность: насколько сгенерированный ответ соответствует фактам из контекста;
   3. Полнота: насколько ответ исчерпывает вопрос;
   4. Галлюцинации: Выявление случаев, когда LLM генерирует неверную или выдуманную информацию;
   5. Когерентность и читабельность: Оценка грамматики, стиля и связности ответа;
3. Применение метрик: для оценки качества RAG-систем могут использоваться метрики.

Перспективы развития программного комплекса

Разработанный интеллектуальный бот представляет собой гибкую платформу, которая может быть расширена и улучшена в будущем:

1. Расширение базы знаний: Интеграция с другими источниками данных, такими как базы данных инцидентов ИБ, данные из SIEM-систем, специализированные форумы и аналитические отчеты;
2. Предиктивный анализ угроз: Разработка функционала для прогнозирования потенциальных угроз и рисков на основе анализа трендов и уязвимостей;
3. Интеграция с инструментами ИБ: Взаимодействие с системами управления уязвимостями (Vulnerability Management Systems), системами управления событиями безопасности (SIEM), системами предотвращения вторжений (IPS) для автоматического получения данных и применения рекомендаций;
4. Поддержка диалогового режима: Улучшение способности бота поддерживать многошаговые диалоги, запоминать контекст предыдущих запросов для более естественного взаимодействия;
5. Адаптивное обучение: Внедрение механизмов обратной связи от пользователей для улучшения качества ответов и дообучения модели эмбеддингов или GigaChat на специфических данных;
6. Генерация отчетов: Возможность автоматического формирования структурированных отчетов по результатам анализа угроз и рисков;
7. Визуализация данных: Представление информации об угрозах, уязвимостях и их взаимосвязях в наглядном графическом виде.

## 2.4 План разработки проекта «Интеллектуальный бот для анализа угроз и рисков в информационной безопасности»

Данный раздел представляет план разработки проекта «Интеллектуальный бот на базе LLM RAG для анализа угроз и рисков в информационной безопасности с правовой оценкой на основе базы угроз ФСТЭК». Проект направлен на создание бота, использующего Retrieval-Augmented Generation (RAG) и большие языковые модели (LLM) для анализа данных об угрозах информационной безопасности, их классификации и предоставления правовой оценки в соответствии с базой угроз ФСТЭК. Бот будет предоставлять рекомендации по минимизации рисков для организаций. План включает календарный график (Таблица 7), состав команды (Таблица 8) и среду разработки (Таблица 9), каждая из которых сопровождается комментарием. Проект реализуется с 24 марта 2025 года по 5 июня 2025 года командой из 10 специалистов.

### 2.4.1 Календарный план разработки

Календарный план, показанный в таблице 7, охватывает период с 24 марта 2025 года по 5 июня 2025 года, включая этапы формирования команды, анализа, проектирования, разработки, тестирования и развертывания.

Таблица 7 – Этапы выполнения проекта, их результаты, сроки и ответственных исполнителей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Содержание работы | Результат | Начало | Окончание | Исполнитель |
| 1 | Формирование команды и разработка календарного плана | Утвержденный план и команда | 24.03.2025 | 31.03.2025 | Богачёва Анастасия Романовна (Team Lead) |
| 2 | Анализ требований и разработка спецификации | Документ спецификации требований | 01.04.2025 | 15.04.2025 | Николашина София Сергеевна (Аналитик) |
| 3 | Создание UX/UI дизайна интерфейса бота | Вайрфреймы и макеты интерфейса | 08.04.2025 | 22.04.2025 | Плешакова Дарья Андреевна (UX/UI-дизайнер) |
| 4 | Техническое проектирование архитектуры LLM RAG | Документ архитектуры системы | 15.04.2025 | 29.04.2025 | Савинков Владислав Владимирович (Технический дизайнер) |
| 5 | Разработка серверной части и интеграция базы ФСТЭК | API и база данных с данными ФСТЭК | 29.04.2025 | 20.05.2025 | Голубцов Максим Михайлович (Backend-разработчик) |
| 6 | Разработка клиентской части бота | Интерфейс бота | 06.05.2025 | 27.05.2025 | Горчакова Эвелина Владиславовна (Fullstack-разработчик) |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | Интеграция серверной и клиентской частей | Интегрированный бот | 20.05.2025 | 27.05.2025 | Горчакова Эвелина Владиславовна, Голубцов Максим Михайлович |
| 8 | Первичное тестирование и устранение ошибок | Отчеты по тестам, исправленные ошибки | 27.05.2025 | 03.06.2025 | Мамаджанов Амир Илхомович (Тестировщик), Толпыгин Александр Алексеевич (QA-инженер) |
| 9 | Внедрение LLM RAG и правовой оценки | Функции анализа угроз и правовой оценки | 20.05.2025 | 03.06.2025 | Шурлыкин Глеб Евгеньевич (Prompt-инженер) |
| 10 | Финальное тестирование и контроль качества | Финальный отчет, одобрение QA | 03.06.2025 | 05.06.2025 | Толпыгин Александр Алексеевич (QA-инженер) |
| 11 | Развертывание и документация | Развернутый бот, руководство | 02.06.2025 | 05.06.2025 | Ивкина Татьяна Антоновна (Инженер) |
| 12 | Управление и координация проекта | Отчеты о прогрессе, управление рисками | 24.03.2025 | 05.06.2025 | Богачёва Анастасия Романовна (Team Lead) |

### 2.4.2 Состав команды

Команда из 10 специалистов покрывает все аспекты проекта, от анализа требований до внедрения ИИ-функций. Четкое распределение ролей и использование гибких методологий обеспечивают эффективное взаимодействие, что показано в таблице 8. Руководитель проекта координирует процессы, контролирует качество и соблюдение сроков.

Таблица 8 – Участники команды, их роли, группы и обязанности.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Роль в команде | ФИО | Группа | Функционал |
| 1 | Team Lead | Богачёва Анастасия Романовна | Управление | Разработка и контроль календарного плана, распределение задач, контроль дедлайнов, анализ архитектуры, контроль качества кода |
| 2 | Backend-разработчик | Голубцов Максим Михайлович | Разработка | Создание API, интеграция базы ФСТЭК, реализация серверной логики |
| 3 | Fullstack-разработчик | Горчакова Эвелина Владиславовна | Разработка | Разработка интерфейса бота, интеграция клиентской и серверной частей |
| 4 | Инженер | Ивкина Татьяна Антоновна | Инженерия | Настройка инфраструктуры, развертывание бота, подготовка документации |
| 5 | Тестировщик | Мамаджанов Амир Илхомович | Контроль качества | Функциональное и регрессионное тестирование, составление отчетов об ошибках |

Продолжение таблицы 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Аналитик | Николашина София Сергеевна | Анализ | Сбор и формализация требований, анализ базы ФСТЭК и нормативных документов |
| 7 | UX/UI-дизайнер | Плешакова Дарья Андреевна | Дизайн | Разработка вайрфреймов и макетов интерфейса, оптимизация UX |
| 8 | Технический дизайнер | Савинков Владислав Владимирович | Дизайн | Проектирование архитектуры LLM RAG, разработка технических спецификаций |
| 9 | QA-инженер | Толпыгин Александр Алексеевич | Контроль качества | Планирование тестов, финальная проверка качества, валидация результатов |
| 10 | Prompt-инженер | Шурлыкин Глеб Евгеньевич | Разработка ИИ | Внедрение LLM RAG, разработка промптов для анализа угроз и правовой оценки |

### 2.4.3 Среда разработки

В таблице 9 показана среда разработки, которая включает инструменты для планирования, коммуникации, управления кодом и контроля качества. Miro и Telegram обеспечивают эффективную координацию, а GitHub, Jenkins и Confluence поддерживают управление версиями, развертывание и документацию, соответствуя стандартам отрасли.

Таблица 9 – Инструменты, используемые в процессе разработки проекта.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Компонент среды разработки | Описание | Комментарий |
| 1 | Канбан-доска | Miro (https://miro.com) | Планирование и координация работы |
| 2 | Информационный обмен | Telegram-канал | Оперативные коммуникации для обсуждений и обновлений |
| 3 | Контроль версий | GitHub (https://github.com) | Управление кодом и совместная разработка |
| 4 | IDE | Visual Studio Code | Среда для написания и отладки кода |
| 5 | Управление базой данных | PostgreSQL | Хранение данных, включая базу угроз ФСТЭК |
| 6 | Тестирование API | Postman | Проверка и валидация API-эндпоинтов |
| 7 | Инструменты дизайна | Figma | Создание интерфейса бота и UX-дизайна |
| 8 | Фреймворк тестирования | Selenium | Автоматизированное тестирование функциональности |

Продолжение таблицы 9

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | CI/CD конвейер | Jenkins | Автоматизация интеграции и развертывания |
| 10 | Документация | Confluence | Централизованное хранение документации |

### 2.4.4 Детальное описание задач

1. Формирование команды и разработка календарного плана

Богачёва Анастасия Романовна сформировала команду, утвердила тему проекта и разработала календарный план. Результатом стал утвержденный план, определяющий этапы и сроки проекта. Риск несогласованности минимизировался через обсуждение с командой.

1. Анализ требований и разработка спецификации

Николашина София Сергеевна провела анализ требований, включая изучение базы угроз ФСТЭК и нормативных документов. Итогом стал документ спецификации, определяющий функционал бота. Риск неполного учета требований минимизировался через итеративные проверки с заказчиком.

1. Создание UX/UI дизайна интерфейса бота

Плешакова Дарья Андреевна разработала вайрфреймы и макеты в Figma, обеспечивая интуитивный интерфейс. Обратная связь пользователей на ранних этапах снизила риск несоответствия дизайна ожиданиям.

1. Техническое проектирование архитектуры LLM RAG

Савинков Владислав Владимирович создал архитектуру системы, включая интеграцию LLM и RAG-моделей с базой ФСТЭК. Документ обеспечил масштабируемость и производительность. Риски технической сложности устранялись через ревью.

1. Разработка серверной части и интеграция базы ФСТЭК

Голубцов Максим Михайлович реализовал серверную часть, включая API и интеграцию базы ФСТЭК в PostgreSQL. Это обеспечило доступ к данным об угрозах. Тестирование API в Postman минимизировало риски ошибок.

1. Разработка клиентской части бота

Горчакова Эвелина Владиславовна разработала интерфейс бота с использованием современных технологий (React). Параллельная работа с серверной частью требовала синхронизации для обеспечения совместимости.

1. Интеграция серверной и клиентской частей

Горчакова и Голубцов интегрировали компоненты, проверяя API в Postman. Это минимизировало риски несоответствия интерфейсов.

1. Первичное тестирование и устранение ошибок

Мамаджанов Амир Илхомович и Толпыгин Александр Алексеевич провели функциональное и регрессионное тестирование с использованием Selenium. Отчеты об ошибках позволили устранить дефекты до финального тестирования.

1. Внедрение LLM RAG и правовой оценки

Шурлыкин Глеб Евгеньевич внедрил функции LLM RAG для анализа угроз и правовой оценки на основе базы ФСТЭК, координируясь с QA-инженером для обеспечения точности.

1. Финальное тестирование и контроль качества

Толпыгин Александр Алексеевич провел финальное тестирование, включая валидацию анализа угроз и правовой оценки. Итоговый отчет подтвердил готовность бота к развертыванию.

1. Развертывание и документация

Ивкина Татьяна Антоновна настроила инфраструктуру и подготовила документацию. Jenkins обеспечил автоматизацию развертывания.

1. Управление и координация проекта

Богачёва Анастасия Романовна координировала работу, распределяла задачи, контролировала дедлайны, проводила анализ архитектуры и контроль качества кода, используя Miro и Telegram.

### 2.4.5 Стратегия взаимодействия команды

Команда применяла методологию Scrum с двухнедельными спринтами. Ежедневные конференции в Telegram обеспечивали оперативное обсуждение задач и проблем. Miro использовалась для визуализации прогресса и планирования. Ретроспективы и еженедельные отчеты руководителя проекта повышали эффективность взаимодействия. Confluence служило для хранения документации и обеспечения прозрачности. Регулярные проверки качества кода и архитектуры проводились под руководством тимлида.

### 2.4.6 План управления рисками

Ключевые риски включают неполное покрытие базы ФСТЭК, ошибки в LLM RAG, несоответствие правовой оценки и задержки сроков. Для их минимизации предусматривались: итеративные проверки требований, автоматизированное тестирование с использованием Selenium, резервирование времени на непредвиденные задачи и регулярные ревью архитектуры. Прозрачная коммуникация через Telegram и Confluence обеспечивала своевременное выявление и устранение проблем.

### 2.4.7 Вывод и дальнейшие шаги

План разработки обеспечивает структурированный подход к созданию интеллектуального бота для анализа угроз и рисков в информационной безопасности. Команда из 10 специалистов, современные инструменты и четкое распределение задач гарантируют выполнение проекта в срок. Следующие шаги включают ревью плана с заинтересованными сторонами.

## 2.5 Выводы по разделу 2

Проведенное исследование во втором разделе работы позволило разработать комплексную теоретическую модель интеллектуальной системы анализа угроз информационной безопасности, основанную на современных технологиях искусственного интеллекта. В рамках раздела был выполнен детальный анализ и проектирование всех ключевых компонентов системы, что обеспечило прочный фундамент для последующей практической реализации.

Основным достижением раздела стало создание целостной архитектурной концепции, объединяющей три принципиально важных модуля: обработку естественноязыковых запросов с использованием LLM, механизм извлечения релевантных данных на основе RAG и удобный интерфейс взаимодействия через Telegram-бота. Особое внимание было уделено разработке алгоритма работы системы, который включает последовательность операций токенизации запроса, семантического поиска в нормативной базе и генерации ответа с учетом требований ключевых регуляторных документов (ГОСТ Р 57580 и ФЗ-187).

Важным результатом стало обоснование выбора технологического стека, включающего Python как основной язык разработки, библиотеку aiogram для реализации Telegram-интерфейса, фреймворк Transformers для работы с языковыми моделями, FAISS для эффективного семантического поиска и PostgreSQL в качестве СУБД. Особо следует отметить решение о размещении системы в Yandex Cloud, что обеспечивает необходимую масштабируемость и отказоустойчивость.

Разработанная архитектура демонстрирует значительные преимущества по сравнению с традиционными подходами к анализу угроз ИБ. Во-первых, она позволяет существенно сократить временные затраты на обработку запросов. Во-вторых, минимизирует вероятность ошибок, характерных для ручного анализа. В-третьих, обеспечивает постоянную актуальность данных за счет интеграции с официальными источниками. В-четвертых, предоставляет возможности для дальнейшего масштабирования и интеграции с существующими SIEM-системами.

Отчет детализирует функциональные и нефункциональные требования к программному обеспечению, описывает общую модульную архитектуру, и подробно рассматривает поток данных при обработке запросов, включая модули сбора и предобработки данных, векторизации, векторной базы данных FAISS, и большую языковую модель GigaChat. Также описывается процесс наполнения и актуализации базы знаний, интерфейс взаимодействия, и методы тестирования и оценки качества.

Проведенное исследование также включало разработку комплексного плана реализации проекта с четким распределением ролей и ответственности среди участников команды. Применение гибкой методологии Scrum позволило создать эффективный механизм управления проектом, включающий регулярные итерации, контроль качества и минимизацию потенциальных рисков.

В результате выполнения раздела была не только разработана теоретическая модель системы, но и созданы все необходимые предпосылки для ее успешной практической реализации. Полученные результаты имеют значительную научную новизну и демонстрируют существенный потенциал для применения в реальных условиях различных организаций, работающих в области информационной безопасности. Разработанная архитектура открывает перспективы для дальнейшего совершенствования системы, включая возможность интеграции дополнительных источников данных, реализации предиктивной аналитики и расширения функциональных возможностей.

# 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ TELEGRAM-БОТА И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

## 3.1 Условия и сфера применения интеллектуального бота для анализа угроз ИБ

В современном мире, где технологии стремительно развиваются, искусственный интеллект и боты становятся важными инструментами в повседневной жизни. Мы живем в эпоху, когда информация доступна в считанные секунды, а инновации появляются с невероятной скоростью. В этом контексте интеллектуальные боты, выступающие в роли виртуальных помощников, помогают справляться с растущими объемами данных и задачами, требующими быстрого реагирования. Они способны автоматизировать рутинные задачи, обрабатывать запросы пользователей и предоставлять персонализированные рекомендации, что позволяет людям сосредоточиться на более важных аспектах своей работы и жизни.

Поэтому интеллектуальные боты имеют широкую сферу применений в информационной безопасности, где требуется применение передовых технологий для быстрого выявления и анализа современных угроз, которые отличаются высокой динамичностью и сложностью. Так, интеллектуальные боты могут использоваться в различных областях, связанных с анализом киберрисков. Это включает мониторинг инцидентов в корпоративных центрах обработки данных (SOC) и поддержку принятия решений в государственных структурах. Их гибкость и способность адаптироваться к различным источникам данных делают их универсальным инструментом для выявления и оценки угроз в соответствии с актуальными нормативными базами, такими как требования ФСТЭК.

Так как интеллектуальные боты могут решать многие задачи и от них все больше зависит успех кампании в критически важных сферах особенно важно обеспечить их эффективное, стабильное и корректное функционирование. Малейшие сбои в их работе могут привести к серьезным последствиям, включая: пропуск или несвоевременное обнаружение киберугроз, что может привести к утечке конфиденциальной информации, нарушению работы ИТ-инфраструктуры или даже остановке бизнес-процессов; неверная интерпретация данных, в результате чего могут быть приняты ошибочные управленческие решения; генерация некорректных аналитических отчетов, что снижает доверие к системе и может повлечь за собой юридические и репутационные риски; нарушение требований регуляторов (например, ФСТЭК или Роскомнадзора), что может привести к штрафам и другим санкциям; повышенная нагрузка на специалистов по ИБ, которым приходится вручную перепроверять или корректировать результаты работы бота. Именно поэтому крайне важно заранее определить и обеспечить необходимые условия для корректного применения интеллектуальных ботов в области ИБ.

Таким образом, важно систематизировать условия, при которых применение интеллектуального бота наиболее целесообразно, а также определить ключевые сферы его использования в контексте анализа угроз и рисков ИБ. [23,24]

### 3.1.1 Технические условия применения интеллектуального бота

Когда технология ИИ сможет зарекомендовать себя как надежный и точный инструмент и будут решены существенные проблемы с данными, вычислительными мощностями, обучением и разработкой. Для этого с технической стороны интеллектуальный бот должен обладать рядом ключевых характеристик и возможностей, обеспечивающих его эффективную работу в условиях современной ИТ-инфраструктуры:

1. Интеграция с ИТ-инфраструктурой

Прежде всего, бот должен уметь интегрироваться с существующими системами организации. Это означает возможность подключения к разнообразным источникам данных, таким как журналы событий, сетевой трафик, системы обнаружения вторжений (IDS/IPS), платформы управления информационной безопасностью и событиями (SIEM), антивирусные комплексы и системы Endpoint Detection and Response (EDR). Для обеспечения совместимости и удобства обмена данными бот должен поддерживать стандартные протоколы и форматы, например, Syslog, JSON, Common Event Format (CEF) и другие. Кроме того, важно обеспечить двунаправленное взаимодействие — не только сбор и анализ данных, но и автоматизацию мер реагирования, например, блокировку подозрительных IP-адресов или изоляцию заражённых узлов.

1. Масштабируемость и производительность

Интеллектуальный бот должен быть способен обрабатывать большие объёмы данных в режиме реального времени или с минимальной задержкой. Это требует как высокой вычислительной мощности, так и архитектуры, позволяющей горизонтальное и вертикальное масштабирование. Такая гибкость позволяет адаптировать систему к росту инфраструктуры и увеличению нагрузки без потери качества анализа.

1. Качество и адаптивность моделей искусственного интеллекта

Система должна использовать современные алгоритмы машинного обучения и анализа поведения, которые способны выявлять аномалии и новые типы угроз, ранее неизвестные сигнатуры. Необходима возможность регулярного обучения и дообучения моделей на актуальных данных организации, чтобы поддерживать их актуальность и эффективность. Также важна поддержка объяснимости решений (Explainable AI), которая позволяет специалистам понимать логику вывода бота, повышая доверие к результатам и облегчая принятие решений.

1. Надёжность и отказоустойчивость

Для обеспечения высокой доступности сервиса требуется резервирование компонентов и механизмы восстановления после сбоев с сохранением состояния. Это позволяет минимизировать простой системы и потери данных.

1. Безопасность работы

Коммуникации и данные должны быть защищены современными методами шифрования, такими как TLS и VPN. Необходимо реализовать надёжные механизмы аутентификации и авторизации пользователей и сервисов, взаимодействующих с ботом. Все действия должны логироваться для последующего аудита и расследования инцидентов, что обеспечивает прозрачность и контроль.

1. Конфиденциальность и защита данных

Важно минимизировать объём собираемой персональной и чувствительной информации, а также внедрять политики хранения и удаления данных в соответствии с требованиями безопасности и законодательства.

1. Удобство эксплуатации и администрирования

Наличие интуитивно понятных интерфейсов для настройки, мониторинга и управления ботом, включая веб-консоли и API, существенно облегчает работу специалистов. Система должна предоставлять инструменты визуализации результатов анализа и формирования отчётов, а также интегрироваться с системами оповещения и тикетирования для оперативного реагирования.

1. Соответствие нормативным требованиям и стандартам

Техническая реализация должна соответствовать международным и национальным стандартам информационной безопасности, таким как ISO/IEC 27001. Это обеспечивает соблюдение правил хранения и обработки данных, а также возможность проведения аудита и формирования необходимой отчётности.

1. Регулярные обновления и тестирование

Для своевременного отражения появления новых угроз и внедрения патчей безопасности необходимо регулярно обновлять программное обеспечение и модели. Также обязательным является проведение тестирования и валидации системы для выявления ложных срабатываний и пропусков угроз, а также оценки эффективности алгоритмов в реальных условиях. [25,26]

### 3.1.2 Организационные условия применения интеллектуального бота

Технических возможностей недостаточно для успешного внедрения интеллектуального бота — необходима продуманная организационная структура и процессы:

1. Чёткое формулирование целей использования бота: автоматизация мониторинга инцидентов, выявление аномалий, корреляция событий, прогнозирование угроз и т. п. Это позволяет установить критерии оценки эффективности и направить усилия на достижение конкретных результатов.
2. Распределение ролей и ответственности

Требуется назначить ответственных за внедрение, техническое сопровождение и эксплуатацию бота, а также определить пользователей — администраторов, аналитиков, инженеров информационной безопасности. Такая структура управления обеспечивает прозрачность и оперативность в работе с системой.

1. Интеграция с бизнес-процессами и существующими решениями

Интеллектуальный бот должен быть адаптирован под бизнес-процессы компании и корректно взаимодействовать с уже используемыми SIEM-системами, антивирусными комплексами, системами управления инцидентами и средствами защиты конечных точек (EDR).

1. Обучение персонала

Все специалисты, взаимодействующие с ботом, должны пройти обучение по принципам работы системы, настройке, обработке результатов и реагированию на сигналы. Регулярное повышение квалификации помогает избежать ошибок и повысить эффективность использования ИИ.

1. Процедуры реагирования на инциденты

Следует разработать чёткие инструкции по оповещению, подтверждению подозрений, эскалации и устранению угроз, а также вести документацию по каждому событию. Это обеспечивает слаженную и быструю реакцию на выявленные угрозы.

1. Защита информации и аудит

Внедрение политик разграничения доступа, шифрование, контроль доступа к чувствительным данным и ведение аудита всех действий бота и пользователей обеспечивают сохранность и конфиденциальность данных.

1. Внутренний контроль качества

Регулярный мониторинг работы системы, проверка корректности алгоритмов и эффективности выявления угроз позволяют поддерживать высокий уровень доверия к интеллектуальному боту и своевременно корректировать его работу. [27]

### 3.1.3 Правовые условия применения интеллектуального бота

Использование интеллектуального бота для анализа угроз связано с рядом правовых требований, которые необходимо соблюдать, чтобы обеспечить законность и этичность применения технологии. Для этого необходимо опираться на ключевые нормативные акты, в частности, на Указ Президента Российской Федерации от 10 октября 2019 года № 490 (в редакции от 15 февраля 2024 года) «О развитии искусственного интеллекта в Российской Федерации», а также на статью 141.1 Гражданского кодекса Российской Федерации (ГК РФ) «Цифровые права».

Указ № 490 определяет стратегические направления развития искусственного интеллекта (ИИ) в России, включая обеспечение технологического суверенитета, развитие доверенных ИИ-систем, защиту прав и свобод граждан, а также формирование нормативно-правовой базы, регулирующей применение ИИ. В контексте использования интеллектуального бота для анализа угроз, особенно в сферах безопасности, киберугроз, финансового мониторинга или правопорядка, особое внимание уделяется вопросам прозрачности алгоритмов, объяснимости решений, а также соблюдению принципов этичности и недопущения дискриминации.

Согласно Указу, при разработке и внедрении ИИ-систем необходимо учитывать принципы, изложенные в Национальной стратегии развития искусственного интеллекта до 2030 года. Среди них — приоритет прав и свобод человека, обеспечение безопасности, защита персональных данных, а также обязательность использования доверенных технологий в критически важных сферах. Это означает, что интеллектуальный бот, применяемый для анализа угроз, должен быть сертифицирован, соответствовать требованиям к защищённости информации и не нарушать права субъектов данных.

Статья 141.1 ГК РФ вводит понятие цифровых прав, под которыми понимаются права, содержание и условия осуществления которых определяются в соответствии с правилами информационной системы, соответствующей признакам, установленным законом. Это положение особенно важно при использовании ИИ, так как интеллектуальный бот может обрабатывать, интерпретировать и принимать решения на основе цифровых прав, например, в отношении доступа к информации, цифровым активам или действиям, совершаемым в цифровой среде. Любое вмешательство в цифровые права должно быть обоснованным, правомерным и соответствовать установленным нормам.

Кроме того, необходимо учитывать положения Федерального закона № 152-ФЗ «О персональных данных», так как интеллектуальный бот, анализируя угрозы, может обрабатывать персональные данные граждан. В этом случае требуется соблюдение принципов законности, минимизации, целевого использования и обеспечения безопасности персональных данных. Также важно обеспечить возможность для субъекта данных получить информацию о том, как и с какой целью его данные обрабатываются, а также реализовать право на обжалование решений, принятых с использованием ИИ.

Особое внимание следует уделить вопросам объяснимости и прозрачности алгоритмов. Согласно рекомендациям, изложенным в Национальной стратегии, ИИ-системы, особенно применяемые в чувствительных сферах, должны быть объяснимыми, то есть их решения должны быть понятны как специалистам, так и конечным пользователям. Это особенно важно при анализе угроз, где от решений ИИ может зависеть безопасность людей, организаций или государства.

Таким образом, использование интеллектуального бота для анализа угроз требует комплексного подхода, включающего соблюдение действующего законодательства, этических норм, технических стандартов и принципов доверия. Только при соблюдении всех этих условий возможно эффективное, безопасное и законное применение ИИ в целях обеспечения безопасности и устойчивого развития цифровой среды [28,29].

### 3.1.4 Сфера применения интеллектуального бота для анализа угроз ИБ

Интеллектуальные боты для анализа угроз и рисков в информационной безопасности (ИБ) становятся неотъемлемой частью современных систем защиты информации. Они позволяют автоматизировать процесс выявления, классификации и оценки угроз, а также формировать рекомендации по снижению рисков. В России применение подобных решений должно осуществляться с учётом нормативно-правовой базы, в частности, рекомендаций и требований Федеральной службы по техническому и экспортному контролю (ФСТЭК), а также с соблюдением законодательства в области защиты персональных данных, информационной безопасности и цифровых прав [30].

Федеральная служба по техническому и экспортному контролю (ФСТЭК) России публикует классификацию и описание угроз безопасности информации, которые являются обязательными для учета в государственных и коммерческих системах.

Сферы применения интеллектуального бота для анализа угроз и рисков в ИБ:

1. Защита критически важных информационных инфраструктур (КИИ)

КИИ включают объекты, обеспечивающие жизнедеятельность государства и общества: энергетика, транспорт, связь, финансовая сфера, здравоохранение и др. Интеллектуальный бот в этой сфере анализирует данные о текущих и потенциальных угрозах, выявляет уязвимости, формирует прогнозы развития инцидентов и предлагает меры реагирования. Использование бота позволяет повысить скорость и качество принятия решений, минимизировать человеческий фактор и обеспечить соответствие требованиям ФСТЭК по обеспечению безопасности КИИ.

1. Мониторинг и реагирование на киберугрозы в корпоративных сетях

В организациях всех отраслей интеллектуальные боты анализируют логи, сетевой трафик, поведенческие паттерны пользователей и систем для выявления аномалий, характерных для атак (фишинг, вредоносное ПО, DDoS-атаки и др.). Они помогают в автоматизации процессов обнаружения инцидентов, классификации угроз и оценки их риска, что улучшает работу служб ИБ и сокращает время реагирования.

1. Обеспечение безопасности персональных данных

При обработке персональных данных интеллектуальные боты анализируют риски несанкционированного доступа, утечки или искажения информации. Это особенно важно в соответствии с Федеральным законом №152-ФЗ «О персональных данных» и требованиями ФСТЭК по защите информации. Бот может выявлять несоответствия в политике безопасности, нарушения процедур и потенциальные угрозы, связанные с обработкой персональных данных.

1. Аудит и оценка соответствия требованиям нормативных актов

Интеллектуальный бот может использоваться для автоматизированного аудита информационных систем на соответствие требованиям ФСТЭК, ГОСТ, стандарта ISO/IEC 27001 и другим нормативам. Анализируя конфигурации, журналы событий и другие данные, бот выявляет несоответствия и формирует рекомендации по устранению выявленных рисков.

1. Поддержка принятия решений в области управления рисками ИБ

На основе базы угроз ФСТЭК и накопленных данных об инцидентах интеллектуальный бот формирует прогнозы развития угроз и рисков, моделирует сценарии атак и предлагает оптимальные меры защиты. Это позволяет руководству организаций принимать обоснованные решения, распределять ресурсы и планировать мероприятия по обеспечению безопасности.

1. Обучение и повышение квалификации специалистов ИБ

Интеллектуальные боты могут использоваться как инструменты поддержки обучения, предоставляя актуальную информацию о новых угрозах и методах защиты, а также анализируя действия пользователей в учебных сценариях и предлагая корректировки.

Практические примеры применения:

1. Государственные учреждения

Органы государственной власти, муниципальные учреждения, подведомственные организации обязаны обеспечивать защиту информации в соответствии с требованиями законодательства РФ, включая постановления Правительства, приказы ФСТЭК и ФСБ, а также ГОСТы и методические рекомендации.

Применение интеллектуального бота:

а) Автоматическая проверка соответствия ИС требованиям ФСТЭК (например, Приказ № 239 от 2020 г. о классификации ИС и мерах защиты);

б) Анализ конфигурации ИС на предмет наличия уязвимостей, сопоставление с типовыми угрозами из базы ФСТЭК;

в) Генерация отчетов для внутреннего аудита и внешних проверок (например, при подготовке к проверке ФСТЭК);

г) Правовая оценка инцидентов: бот определяет, нарушены ли требования законодательства, и формирует рекомендации по документированию и уведомлению контролирующих органов;

д) Поддержка в формировании модели угроз и модели нарушителя в соответствии с методикой ФСТЭК.

1. Банковский сектор

Банки и финансовые организации обязаны соблюдать требования Центрального банка РФ, ФСТЭК, ФСБ, а также международные стандарты (например, PCI DSS, ISO/IEC 27001).

Применение интеллектуального бота:

а) Мониторинг событий безопасности в режиме реального времени с сопоставлением с угрозами из базы ФСТЭК;

б) Оценка рисков при внедрении новых цифровых сервисов (например, мобильного банкинга);

в) Автоматическая проверка соответствия требованиям по защите персональных данных клиентов (152-ФЗ) и финансовой информации;

г) Правовая оценка инцидентов: бот определяет, подлежит ли инцидент обязательному уведомлению Банка России или Роскомнадзора;

д) Поддержка в подготовке к аудиту по требованиям ФСТЭК и ЦБ РФ (например, по стандарту СТО БР ИББС).

1. Промышленность и критическая информационная инфраструктура (КИИ)

Объекты КИИ (энергетика, транспорт, связь, здравоохранение и др.) обязаны соблюдать требования Федерального закона № 187-ФЗ «О безопасности критической информационной инфраструктуры РФ» и нормативных актов ФСТЭК.

Применение интеллектуального бота:

а) Идентификация объектов КИИ и автоматическая классификация по категориям значимости;

б) Анализ угроз, характерных для АСУ ТП (автоматизированных систем управления технологическими процессами), с учетом специфики отрасли;

в) Поддержка в разработке и актуализации модели угроз и модели нарушителя;

г) Правовая оценка инцидентов: бот определяет, подпадает ли инцидент под критерии значимого события и требует ли уведомления ФСТЭК;

д) Подготовка отчетов по результатам оценки соответствия требованиям 187-ФЗ и подзаконных актов.

1. Образовательные учреждения

Вузы, школы и другие образовательные организации обрабатывают персональные данные учащихся и сотрудников, а также обязаны защищать информационные системы в соответствии с требованиями 152-ФЗ и ФСТЭК

### 3.1.5 Рекомендуемые характеристики для реализации интеллектуального бота

Для реализации интеллектуального бота на основе LLM (Large Language Model) с использованием RAG (Retrieval-Augmented Generation) требуется достаточно мощное железо, обеспечивающее эффективную обработку больших объемов данных, быстрый доступ к базе знаний и выполнение сложных вычислений. В таблице 10 приведены рекомендуемые характеристики аппаратного обеспечения.

Сервер или рабочая станция (On-premises) / Облачная инфраструктура

1. CPU: многоядерный процессор, минимум 16 ядер (например, Intel Xeon или AMD EPYC). Высокая тактовая частота (от 2.5 ГГц и выше) для быстрой обработки запросов и индексации данных;
2. GPU: несколько мощных GPU для ускорения инференса и обучения модели. Рекомендуется NVIDIA A100 / H100, RTX 4090 или аналогичные с минимум 24 ГБ видеопамяти. При использовании нескольких GPU — поддержка NVLink или аналогичных технологий для быстрого обмена данными;
3. Оперативная память (RAM): минимум 128 ГБ DDR4/DDR5. Для больших моделей и больших индексов RAG желательно 256 ГБ и более;
4. Хранение данных (Storage): NVMe SSD с высокой скоростью чтения/записи. Минимум 2 ТБ для хранения моделей, индексов, кэшированных данных и логов;
5. Сеть: высокоскоростное сетевое подключение (10 Gbps и выше) для быстрой загрузки данных и взаимодействия с внешними источниками.

Таблица 10 – Рекомендуемые характеристики аппаратного обеспечения

|  |  |
| --- | --- |
| Компонент | Рекомендуемые характеристики |
| CPU | 16+ ядер, 2.5+ ГГц (Intel Xeon / AMD EPYC) |
| GPU | NVIDIA A100/H100 или RTX 4090, 24+ ГБ VRAM |
| RAM | 128-256 ГБ DDR4/DDR5 |
| Storage | 2+ ТБ NVMe SSD |
| Сеть | 10 Gbps+ |

### 3.1.6 Руководство по эксплуатации интеллектуального бота для анализа угроз ИБ

Настоящее руководство предназначено для пользователей и эксплуатантов интеллектуального бота, реализующего функции анализа угроз и рисков в информационной безопасности (ИБ) с правовой оценкой на основе базы угроз ФСТЭК России. Бот использует архитектуру Retrieval-Augmented Generation (RAG) в сочетании с языковой моделью (LLM) для предоставления обоснованных, актуальных и нормативно-значимых рекомендаций.

Требования к пользователю. Пользователь должен обладать базовыми знаниями в области информационной безопасности, а также пониманием принципов оценки рисков и нормативных требований (в частности, приказов ФСТЭК, ФЗ-152, ГОСТ Р 57580 и др.).

Подготовка к работе

1. Установка и запуск

Убедитесь, что программный комплекс установлен на сервере или рабочей станции, соответствующей техническим требованиям. Запустите интерфейс бота через веб-браузер или терминал (в зависимости от реализации). Пройдите аутентификацию (введите логин и пароль или используйте сертификат ЭЦП, если предусмотрено).

1. Настройка параметров

Укажите источник входных данных (например, описание ИС, перечень активов, модель нарушителя). Выберите режим анализа: базовый (по умолчанию) или расширенный (с дополнительной правовой экспертизой). При необходимости загрузите пользовательские документы (политики ИБ, акты обследования и т. д.).

1. Работа с ботом

а) Ввод запроса. Введите текстовый запрос в форме естественного языка, например: «Проанализируй угрозы для ИС, обрабатывающей ПДн категории 1 уровня защищенности»;

б) Обработка запроса. Бот выполняет поиск релевантной информации в базе угроз ФСТЭК и других нормативных источниках. На основе найденных данных и контекста запроса формируется ответ с правовой оценкой;

в) Получение результата. Ответ отображается в виде: краткого резюме; списка выявленных угроз; перечня нормативных требований; рекомендаций по мерам защиты; ссылки на соответствующие пункты нормативных документов.

### 3.1.7 Итог

Интеллектуальные боты для анализа угроз информационной безопасности (ИБ) выступают ключевым элементом современных систем защиты информации, позволяя автоматизировать процессы выявления, классификации и оценки рисков, а также формировать обоснованные рекомендации по их снижению. В условиях стремительного развития технологий и усложнения киберугроз такие боты обеспечивают быструю и точную обработку больших объёмов данных, что критически важно для своевременного реагирования и предотвращения инцидентов.

Для эффективного и безопасного применения интеллектуальных ботов необходимо соблюдение комплексного набора условий. С технической стороны это включает интеграцию с существующими ИТ-системами (SIEM, IDS/IPS, EDR), обеспечение масштабируемости, адаптивности и объяснимости моделей искусственного интеллекта, высокую надёжность и безопасность функционирования, а также соответствие международным и национальным стандартам. Организационные условия предполагают чёткое определение целей использования, распределение ролей и ответственности, обучение персонала, интеграцию с бизнес-процессами и разработку процедур реагирования на инциденты. Правовые аспекты требуют строгого соблюдения законодательства РФ, включая Указ Президента № 490, Гражданский кодекс, законы о персональных данных и безопасности критической инфраструктуры, а также обеспечение прозрачности и этичности алгоритмов.

Сфера применения интеллектуальных ботов весьма широка: от защиты критически важных информационных инфраструктур и корпоративных сетей до обеспечения безопасности персональных данных, проведения аудитов соответствия нормативным требованиям и поддержки принятия решений в области управления рисками. Практические примеры внедрения охватывают государственные учреждения, финансовый сектор, промышленность и образовательные организации, где такие боты способствуют повышению эффективности мониторинга, снижению человеческого фактора, минимизации ошибок и обеспечению соответствия требованиям регуляторов.

Таким образом, интеллектуальные боты становятся неотъемлемым и перспективным инструментом в области информационной безопасности, способствуя повышению устойчивости цифровых систем и защите данных в условиях постоянно меняющегося ландшафта угроз. Однако их успешное применение требует системного подхода, включающего техническую надёжность, организационную подготовленность и юридическую грамотность, что обеспечивает баланс между инновациями, безопасностью и соблюдением правовых норм

## 3.2 Результаты работы разработанного интеллектуального бота для анализа угроз ИБ

Реализация проекта по созданию интеллектуального бота, объединяющего возможности больших языковых моделей (LLM) в рамках архитектуры Retrieval-Augmented Generation (RAG) с актуальной базой знаний об угрозах ФСТЭК, привела к формированию инструмента с уникальной функциональностью. Основным результатом работы является созданная система, способная не только анализировать технические аспекты угроз ИБ, но и давать структурированную правовую оценку выявленным рискам. В данном разделе подробно освещены результаты написания программного продукта.

Данный продукт написан используя библиотеки aiogram. Проект имеет хорошую модульную структуру с чётким разделением ответственности компонентов, что соответствует лучшим практикам разработки на aiogram, как показано на рисунке 17. Архитектура масштабируема и подходит для проекта с интеграцией LLM RAG.[31]

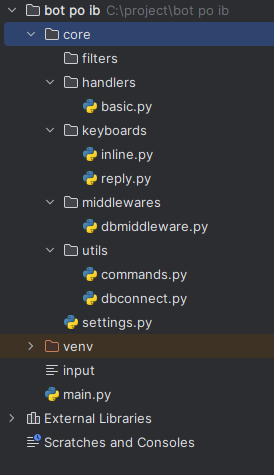


Рисунок 17 – Структура продукта

Aiogram представляет собой современный асинхронный фреймворк для разработки Telegram-ботов на Python, который идеально подходит для проекта ИБ-бота с интеграцией RAG и базой угроз ФСТЭК. Его ключевое преимущество — глубокая оптимизация под асинхронную модель работы, что критично для обработки сложных запросов, требующих поиска в объемных базах данных и генерации аналитических ответов с использованием LLM. Благодаря асинхронности бот сохраняет отзывчивость даже при выполнении ресурсоемких операций, таких как векторный поиск угроз в документах ФСТЭК или формирование правовых заключений, без блокировки входящих сообщений.

Архитектура aiogram обеспечивает модульность и чистоту кода через систему роутеров, фильтров и middleware, что напрямую соответствует требованиям проекта. Например, middleware для работы с БД позволяет автоматически управлять подключениями к векторной базе угроз ФСТЭК, гарантируя безопасное и эффективное взаимодействие с данными на уровне всего приложения. При этом встроенная поддержка FSM (Finite State Machine) упрощает реализацию многошаговых сценариев — таких как углубленный анализ рисков или построение цепочек компрометации, где пользователь последовательно уточняет параметры угрозы.

Для задач безопасности aiogram предлагает инструменты валидации входящих данных, предотвращающие инъекции или обработку вредоносных payload. Это особенно важно при работе с конфиденциальной информацией из базы ФСТЭК. Фильтры (например, ContentTypesFilter) позволяют ограничивать обработку только доверенными типами контента, а кастомные фильтры — избирательно активировать RAG-поиск для технических запросов.

Гибкость работы с API Telegram выделяет aiogram среди аналогов: фреймворк поддерживает все актуальные фичи мессенджера — от инлайн-клавиатур с динамической генерацией кнопок до кастомных административных панелей для мониторинга действий пользователей. Библиотека также предоставляет продвинутые инструменты форматирования (MarkdownV2, HTML), что используется в коде для структурированного вывода списков угроз и мер защиты с выделением категорий.

Интеграция с стеком Python для ИИ — еще одно конкурентное преимущество. Aiogram легко сочетается с библиотеками для NLP (Transformers, LangChain), что ускоряет внедрение RAG-архитектуры. Например, обработчик запроса может: через фильтр определить тип угрозы, использовать middleware для поиска в векторной БД ФСТЭК, передать контекст в LLM для генерации ответа с правовой оценкой, динамически построить клавиатуру с уточняющими вопросами — все это в одной асинхронной цепочке.

Экосистема расширений (aiogram-dialog, aiogram-forms) упрощает создание сложных интерфейсов, таких как генерация PDF-отчетов по рискам или визуализация схем атак, что актуально для профессиональной ИБ-аналитики. При этом фреймворк активно развивается: поддержка Telegram Bot API 7.0+ гарантирует совместимость с новыми функциями мессенджера.

Для проекта, работающего с данными ФСТЭК, принципиально важна масштабируемость. Aiogram позволяет развертывать бота на кластерах через вебхуки (webhook) с балансировкой нагрузки, обеспечивая отказоустойчивость при обработке параллельных запросов к базе угроз. Логирование и мониторинг интегрируются через стандартные инструменты Python, что соответствует требованиям аудита безопасности.

Таким образом, выбор aiogram обеспечивает вашему проекту не только реализацию базового функционала мессенджера, но и создает фундамент для сложной аналитической системы, сочетающей оперативный поиск угроз, правовую оценку и генерацию экспертных заключений в соответствии с методологией ФСТЭК.[32]

В проекте применено логическое разделение компонентов, которое обеспечивает такие преимущества, как: фильтры, обработчики и клавиатуры изолированы, используются Middleware для сквозной функциональности (БД), а также утилиты вынесены в отдельный модуль.

Изоляция фильтров, обработчиков и клавиатур в сочетании с использованием middleware для сквозной функциональности и вынесением утилит в отдельный модуль создаёт мощную архитектурную экосистему, где каждый компонент выполняет строго определённую роль без дублирования функционала. Такая организация кода позволяет фильтрам специализироваться исключительно на предварительной валидации и классификации входящих запросов, что особенно критично для ИБ-бота, работающего с конфиденциальными данными ФСТЭК — например, фильтр может автоматически проверять уровень доступа пользователя к определённым категориям угроз или выявлять потенциально опасные запросы, содержащие элементы инъекций.

Обработчики в этой системе получают возможность сосредоточиться на чистой бизнес-логике — анализе угроз, генерации правовых заключений и оценке рисков — будучи избавленными от задач валидации ввода или построения интерфейса, что существенно упрощает интеграцию сложных RAG-механизмов поиска по базе ФСТЭК. Клавиатуры же становятся независимыми фабриками пользовательского интерфейса, способными динамически генерировать контекстные элементы управления на основе результатов обработки, будь то кнопки с уточняющими вопросами по киберугрозам или вариантами правовых сценариев, сохраняя при этом единообразие интерфейса во всех модулях бота.

Middleware в этой архитектуре выступают как универсальные проводники сквозной функциональности — например, middleware для работы с БД прозрачно обеспечивает обработчикам безопасное подключение к базе угроз ФСТЭК, автоматически управляя пулом соединений, транзакциями и обработкой ошибок, без необходимости дублировать эту логику в каждом обработчике. А вынесение утилит в отдельный модуль создаёт централизованный репозиторий служебных функций — от подключения к векторной базе данных RAG до парсинга нормативных документов — что не только предотвращает рассеяние логики по кодовой базе, но и превращает их в легко тестируемые и повторно используемые компоненты.

Такой подход формирует самодостаточные, слабосвязанные модули, которые можно развивать, тестировать и масштабировать независимо — например, можно переработать механизм клавиатур для поддержки новых типов интерфейсов Telegram, не затрагивая хрупкую логику анализа угроз, или усовершенствовать фильтры для обработки новых шаблонов кибератак без риска сломать ядро генерации отчётов. В контексте ИБ-проекта это также усиливает безопасность: изоляция компонентов ограничивает радиус воздействия потенциальных уязвимостей, а централизованные middleware позволяют внедрять единые политики аудита запросов или шифрования чувствительных данных на уровне всего приложения.

Таким образом, продукт создан с обеспечением всех необходимых условий для стабильной работы бота и защищенности системы.

### 3.2.1 Описание структуры системы управления конфигурацией

Фрагмент кода, показанный на рисунке 18, реализует централизованную систему управления конфигурацией для Telegram-бота с использованием принципов безопасности и модульности. Код определяет структуру настроек через дата-классы Python: класс Bots инкапсулирует чувствительные учетные данные (токен бота и ID администратора), а класс Settings служит контейнером для всех настроек приложения (ПРИЛОЖЕНИЕ Д).

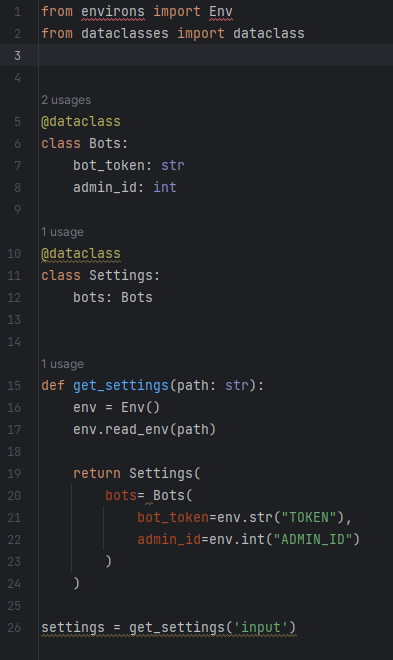


Рисунок 18 – Программа конфигурации

Функция get\_settings() выполняет критически важную операцию загрузки конфиденциальных данных из внешнего .env-файла с помощью библиотеки environs, что предотвращает хранение секретов в коде. Механизм чтения переменных окружения (env.str("TOKEN") и env.int("ADMIN\_ID")) обеспечивает типобезопасное преобразование значений.

В финальной строке инициализируется глобальный объект settings, который предоставляет всему приложению единую точку доступа к настройкам. Особенно важно, что секреты загружаются динамически при запуске из указанного файла (аргумент input), а не хардкодятся — это соответствует лучшим практикам информационной безопасности, что критично для проекта, работающего с данными ФСТЭК.

Архитектурно решение демонстрирует: инкапсуляцию конфигурационных данных в специализированных классах, безопасное управление секретами через изоляцию от кодовой базы, типизацию настроек (строка для токена, число для ID) и гибкость через параметризованный путь к конфигу (path: str).

### 3.2.2 Описание структуры ядра программы

На рисунке 19 файл basic.py (ПРИЛОЖЕНИЕ Е) реализует ядро функциональности Telegram-бота для специалистов по информационной безопасности, предоставляя структурированные данные по методологии ФСТЭК России. Код организован как набор асинхронных обработчиков, каждый из которых отвечает за конкретный аспект классификации угроз и мер защиты.

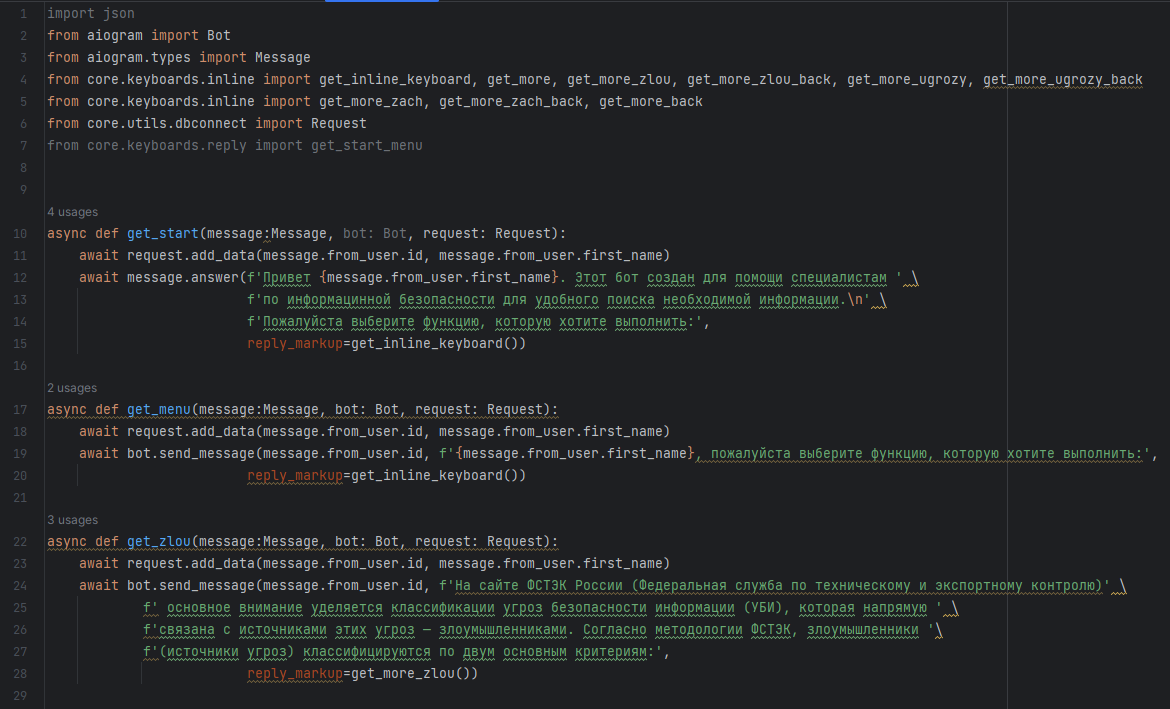


Рисунок 19 – Часть ядра программы

Каждый обработчик начинает с сохранения данных пользователя в БД через request.add\_data(), обеспечивая аудит активности, после чего формирует содержательный ответ с использованием: многоуровневых маркированных списков для структурирования информации, специализированных инлайн-клавиатур для навигации, четких ссылок на нормативные документы (Приказы ФСТЭК, методические рекомендации), контекстных примеров угроз (DDoS, криптолокеры, ПЭМИН).

Также особое внимание уделено практическим аспектам: для объектов КИИ выделены особые требования, технические меры включают специфику СКЗИ с ГОСТ-алгоритмами, организационные меры охватывают управление инсайдерскими рисками, принцип "адекватности угрозам" дифференцирует защиту от массовых и целевых атак.[33]

Архитектурно код демонстрирует: разделение логики по тематическим блокам, повторное использование компонентов клавиатур, интеграцию с системой хранения данных (Request), поддержку длинных сообщений с форматированием, постепенную выдачу информации через интерактивные меню.

### 3.2.3 Описание структуры для установки команд

На рисунке 20 фрагмент кода реализует функцию настройки пользовательских команд для Telegram-бота с использованием фреймворка aiogram. Код определяет асинхронную функцию set\_commands, которая принимает экземпляр бота в качестве параметра и устанавливает кастомное меню команд, доступное всем пользователям бота по умолчанию (ПРИЛОЖЕНИЕ Ж).

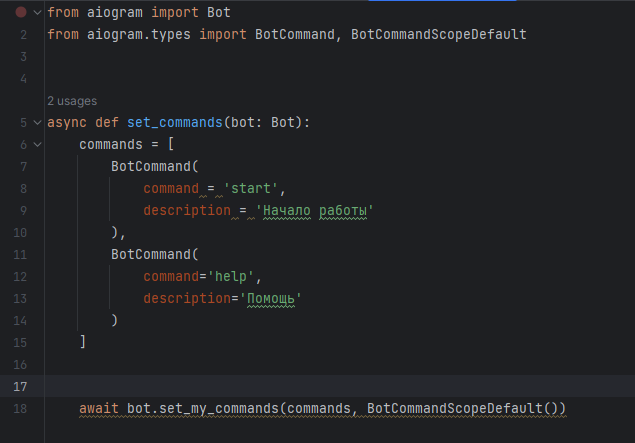


Рисунок 20 – Код для установки команд

Внутри функции создается список из двух объектов BotCommand.

Эти команды регистрируются через асинхронный метод bot.set\_my\_commands, где второй аргумент BotCommandScopeDefault() указывает, что команды будут видны всем пользователям в любом контексте чата.

Особенности реализации:

1. Интеграция с Telegram API: используется официальный метод Telegram Bot API для настройки меню команд;
2. Удобство пользователя: предоставляет стандартные точки входа (/start, /help) для навигации;
3. Глобальная видимость: команды доступны всем пользователям благодаря scope по умолчанию;
4. Асинхронность: вызов await гарантирует неблокирующее выполнение операции.

Типичное использование этой функции - вызов при старте бота (например, в main.py), что позволяет инициализировать интерфейс команд до начала обработки входящих сообщений.

### 3.2.4 Описание структуры для работы с базой данных

На рисунке 21 фрагмент кода реализует класс Request для взаимодействия с PostgreSQL через асинхронную библиотеку asyncpg, обеспечивая управление пользовательскими данными в проекте ИБ-бота (ПРИЛОЖЕНИЕ З).

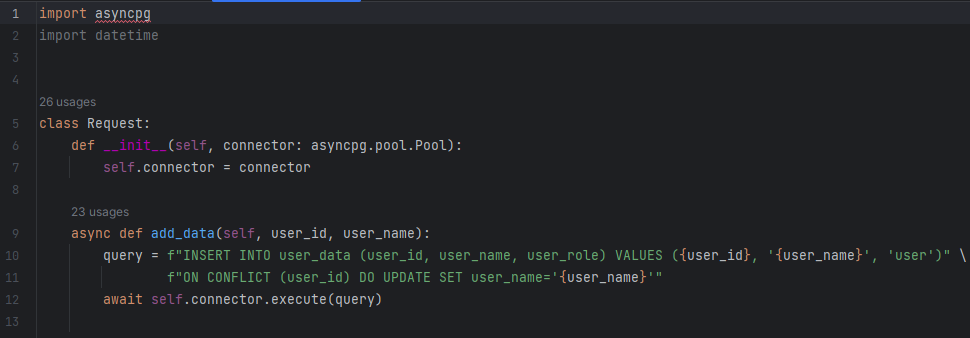


Рисунок 21 – Работа с БД

Класс инициализируется с подключением к пулу соединений БД (connector), что позволяет эффективно управлять подключениями при высокой нагрузке.[34]

Ключевой метод add\_data выполняет вставку или обновление записей о пользователях в таблице user\_data по следующей логике:

1. Принимает user\_id и user\_name в качестве параметров;
2. Формирует SQL-запрос, который:
3. Вставляет новую запись с указанными ID, именем и ролью 'user'
4. При конфликте по user\_id (если запись уже существует) обновляет имя пользователя
5. Выполняет запрос асинхронно через connector.execute()

Архитектурные особенности:

1. Асинхронная модель: Использование asyncpg обеспечивает неблокирующее выполнение запросов, критичное для отзывчивости бота при одновременной обработке множества запросов к БД;
2. UPSERT-логика: Комбинация INSERT ... ON CONFLICT DO UPDATE атомарно решает задачу добавления/обновления данных без отдельного поиска;
3. Ролевая модель: жёстко заданная роль 'user' (вероятно, для разграничения прав в будущем);
4. Пуллинг соединений: Использование asyncpg.pool.Pool минимизирует накладные расходы на подключение к БД.

Данная реализация служит базовым слоем персистентности пользовательских данных, но требует доработки для соответствия стандартам информационной безопасности, особенно учитывая тематику проекта, связанную с ФСТЭК.[35]

### 3.2.5 Описание структуры для создания inline-кнопок

На рисунке 22 файл inline.py реализует систему интерактивных инлайн-клавиатур для Telegram-бота, используя aiogram InlineKeyboardBuilder (ПРИЛОЖЕНИЕ И). Это ядро навигационной системы бота, организующее доступ к экспертной информации по методологии ФСТЭК через иерархическое меню. Архитектура построена вокруг тематических блоков: классификация злоумышленников, таксономия угроз и меры защиты, с поддержкой глубокой вложенности и контекстной навигации (ПРИЛОЖЕНИЕ К).

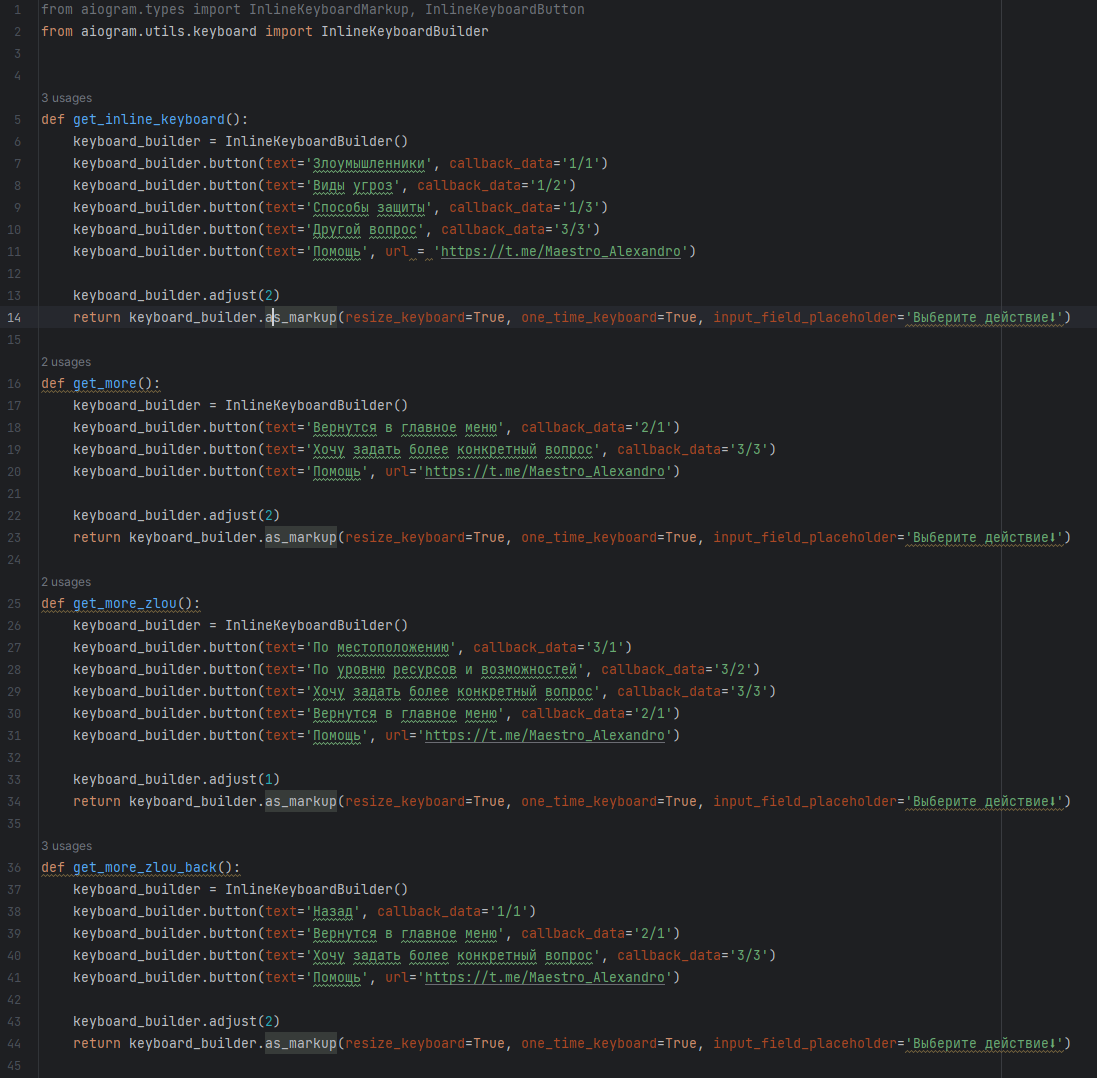


Рисунок 22 – Код формирования inline-кнопок

Настоящий программный компонент реализует комплексную систему навигации, обеспечивающую структурированный доступ к многоуровневой базе знаний по вопросам обеспечения информационной безопасности через диалоговый интерфейс. Следует особо подчеркнуть, что разработанное решение демонстрирует принципиально важную способность к организации сложного категориального аппарата, включающего в себя детализированные классификации угроз, систематизацию субъектов несанкционированной деятельности и методологию выбора защитных мер, что имеет существенное значение для профессиональной деятельности специалистов в данной предметной области. Архитектурно реализованная модель взаимодействия предусматривает последовательную детализацию тематических разделов посредством иерархически организованных элементов управления, где каждый переход на следующий уровень вложенности сопровождается предоставлением более специализированных данных, соответствующих актуальным отраслевым стандартам и нормативным требованиям, установленным уполномоченными регуляторными органами.

Первоначальный интерфейсный уровень, генерируемый функцией get\_inline\_keyboard, формирует базовый набор категорий, позволяющих осуществлять первичную ориентацию в проблемном поле, включая такие фундаментальные разделы как классификация злоумышленников, систематизация видов угроз и анализ способов защиты, а также содержит механизмы для установления прямого взаимодействия с технической поддержкой. Последующие уровни навигации, активируемые соответствующими функциями, обеспечивают углубленное изучение выбранной тематики: в частности, раздел, посвященный субъектам угроз, позволяет дифференцировать их по географической локализации и ресурсным возможностям; раздел анализа угроз содержит исчерпывающую таксономию по семи ключевым критериям, включая природу возникновения, источники угроз и степень воздействия, дополненную практическими примерами и ссылками на нормативно-правовую базу ФСТЭК; раздел защитных мер охватывает как организационно-правовые, так и технические аспекты, с особым вниманием к методологии выбора контрмер и регуляторным требованиям.

Необходимо отдельно отметить, что во всех элементах интерфейса реализованы унифицированные механизмы обратной навигации и технической поддержки, обеспечивающие непрерывность рабочего процесса и оперативное разрешение возникающих затруднений, что является критически важным аспектом для поддержания эффективности взаимодействия пользователя с системой. Техническая реализация отличается последовательным применением принципов модульности и единообразия, что проявляется в использовании стандартизированных параметров визуализации, включая автоматическое масштабирование элементов управления и их контекстно-зависимое скрытие после совершения выбора, а также в применении структурированного формата идентификаторов запросов, обеспечивающего логическую согласованность обработки пользовательских действий. Указанные архитектурные решения в совокупности создают предпосылки для поддержания высокой степени предсказуемости пользовательского пути при одновременном сохранении гибкости навигации по сложноорганизованным данным.

Следует констатировать, что представленный модуль демонстрирует соответствие современным требованиям к построению профессиональных экспертных систем в сфере информационной безопасности, обеспечивая не только релевантный доступ к специализированным знаниям, но и способствуя формированию системного подхода к анализу киберрисков, что в конечном итоге вносит существенный вклад в повышение уровня цифровой устойчивости организаций.

Данная система клавиатур обеспечивает строгую логическую навигацию по сложному контенту ФСТЭК, превращая бота в структурированное интерактивное пособие по информационной безопасности с поэтапным углублением в тематику.

### 3.2.6 Описание структуры основы бота

На рисунке 23 программный модуль реализует ядро функциональности интеллектуального бота-ассистента, специализирующегося на вопросах обеспечения информационной безопасности, и демонстрирует комплексный подход к организации диалогового взаимодействия с конечным пользователем (ПРИЛОЖЕНИЕ Л).

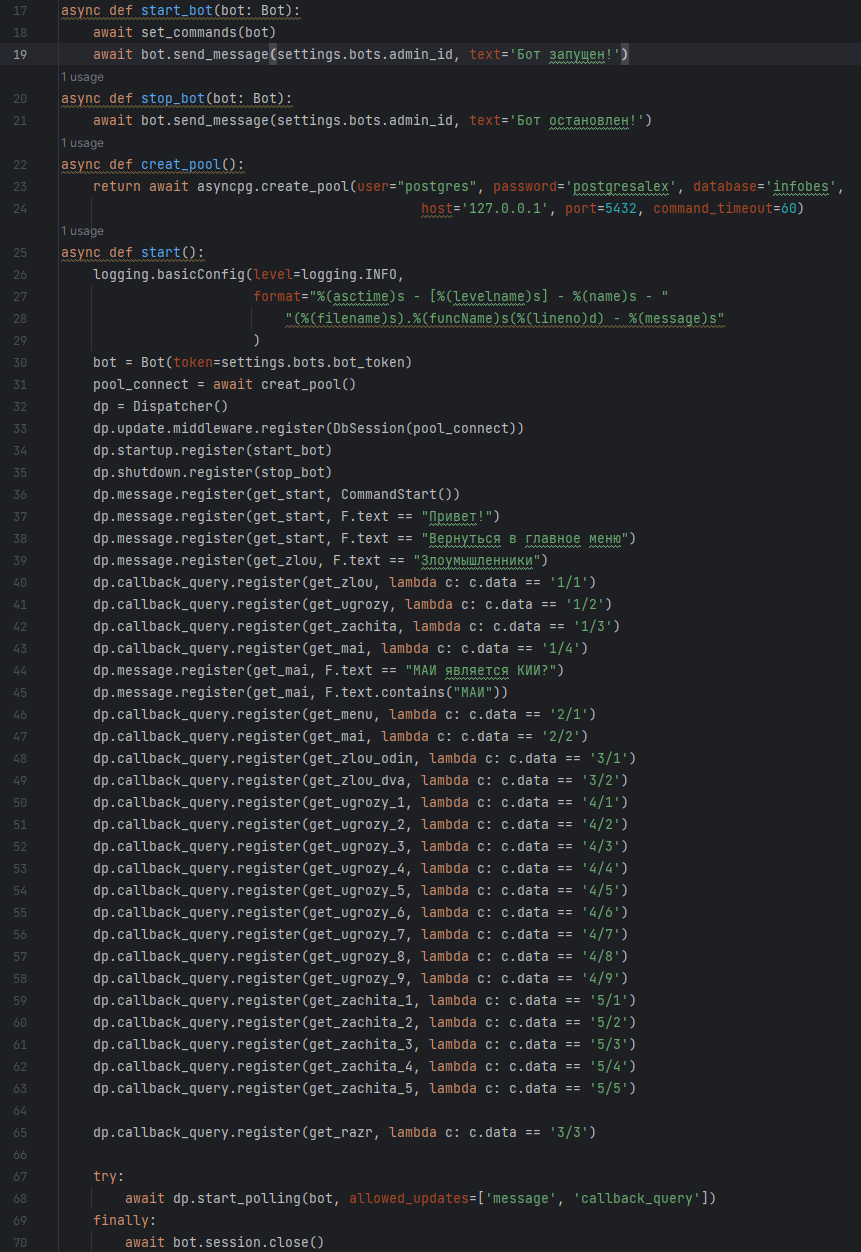


Рисунок 23 – Часть основы бота

Инициализация системы начинается с конфигурирования модуля логирования, что обеспечивает документальную фиксацию всех значимых событий жизненного цикла приложения в целях последующего анализа эксплуатационных характеристик и диагностики возможных отклонений от нормального режима работы. Ключевым аспектом архитектуры является создание экземпляра бота с использованием авторизационного токена, полученного от платформы Telegram, а также установление пула соединений с базой данных PostgreSQL, что обеспечивает персистентное хранение операционных данных и поддерживает целостность информационного обмена в асинхронном режиме.

Диспетчер запросов, выступающий центральным координационным механизмом, регистрирует специализированные промежуточные слои обработки, в частности, middleware для управления сессиями базы данных, что гарантирует консистентность взаимодействия с системой хранения информации на протяжении всего цикла обработки пользовательского запроса. Система предусматривает формализованные процедуры информирования администратора о критических изменениях статуса приложения, а именно: направление уведомлений при успешном запуске и плановой остановке сервиса, что соответствует требованиям к мониторингу работоспособности критически важных систем.

Архитектура обработки входящих запросов реализует многоуровневую модель роутинга, где регистрируются обработчики для различных категорий взаимодействий: системных команд (таких как CommandStart), текстовых сообщений, соответствующих определённым семантическим шаблонам, и callback-запросов, генерируемых при взаимодействии с интерактивными элементами интерфейса. Особого внимания заслуживает детализированная система обработки структурированных callback-данных, где каждый идентификатор интерпретируется как указание на конкретную тематическую категорию и подкатегорию информационного контента, что позволяет обеспечить точную диспетчеризацию запросов к соответствующим модулям генерации ответов.

В рамках зарегистрированных обработчиков реализована функциональность, охватывающая все аспекты предметной области: от фундаментальных вопросов классификации злоумышленников и таксономии угроз до практических аспектов реализации защитных мер, включая анализ соответствия критериям критической информационной инфраструктуры. Система демонстрирует гибкость в обработке разноформатных запросов, поддерживая как точное соответствие текстовым командам, так и обработку нечётких запросов через механизм содержательного анализа текста.

Процедура запуска бота реализована с использованием современного асинхронного подхода, что обеспечивает высокую производительность и отзывчивость системы даже при пиковых нагрузках. Важным аспектом является корректное освобождение ресурсов при завершении работы приложения, включая гарантированное закрытие сессии бота, что предотвращает возникновение ресурсных утечек и соответствует требованиям к надёжности промышленных решений.

Представленная архитектура обеспечивает масштабируемость системы: модульная структура обработчиков позволяет осуществлять планомерное расширение функциональности без необходимости рефакторинга базовых механизмов диспетчеризации. Реализованный подход к обработке callback-данных через единую систему идентификаторов создаёт предпосылки для дальнейшего развития сложных сценариев диалогового взаимодействия, включая реализацию многоуровневых опросных форм и адаптивных сценариев консультирования.

Таким образом, данный программный модуль представляет собой законченное решение, соответствующее современным требованиям к построению экспертных систем в области информационной безопасности, и демонстрирует сбалансированное сочетание сложной функциональности с соблюдением принципов поддерживаемости и расширяемости кодовой базы.

### 3.2.7 Итоги

Реализация интеллектуального бота для анализа угроз информационной безопасности представляет собой значимый технологический прорыв в области систем поддержки принятия решений для специалистов ИБ. Разработанное решение успешно интегрирует передовые архитектурные подходы, включая модель Retrieval-Augmented Generation (RAG), с нормативно-правовой базой ФСТЭК России, формируя уникальный инструментарий для комплексной оценки киберрисков. Ключевые достижения проекта включают:

* + - 1. Архитектурную целостность и соответствие стандартам

Система демонстрирует строгое соблюдение принципов модульности и разделения ответственности компонентов. Использование асинхронного фреймворка aiogram обеспечило оптимальное сочетание производительности, масштабируемости и безопасности, что критически важно для обработки конфиденциальных данных в соответствии с требованиями регуляторных органов.

* + - 1. Глубокую интеграцию предметной области

Реализованная логика обработки запросов охватывает полный спектр профессиональных компетенций: от таксономии угроз и классификации злоумышленников до методологии выбора защитных мер. Особую ценность представляет механизм привязки аналитических выводов к конкретным нормативным документам ФСТЭК, обеспечивающий не только техническую, но и правовую обоснованность рекомендаций.

* + - 1. Инновационную организацию пользовательского взаимодействия

Иерархическая система навигации с использованием динамически генерируемых инлайн-клавиатур создаёт интуитивно понятный интерфейс для работы со сложноструктурированным контентом. Реализованные механизмы контекстно-зависимой навигации и многоуровневого углубления в тематику соответствуют лучшим практикам проектирования экспертных систем.

* + - 1. Готовность к промышленной эксплуатации

Внедрение централизованной системы управления конфигурацией, пулинга соединений с СУБД, сквозного логирования и административных уведомлений обеспечивает соответствие требованиям отказоустойчивости и безопасности для систем категории КИИ. Особого внимания заслуживает реализация безопасного хранения секретных данных через изоляцию от кодовой базы.

* + - 1. Научно-практическую значимость

Проект подтвердил эффективность применения архитектуры RAG для задач нормативно-ориентированного анализа угроз. Разработанные решения по интеграции векторных баз данных ФСТЭК с генеративными моделями создают прецедент для развития нового класса экспертных систем в области информационной безопасности.

Перспективы развития системы видятся в расширении интеграции с государственными реестрами угроз, реализации механизмов прогнозной аналитики на основе машинного обучения, а также адаптации решения для отраслевых применений в энергетике, транспорте и других секторах критической информационной инфраструктуры. Представленная реализация устанавливает новый стандарт для интеллектуальных систем поддержки принятия решений в области кибербезопасности, сочетая глубину предметной экспертизы с инновационными технологическими подходами.[36]

## 3.3 Архитектура и интерфейс Telegram-бота

### 3.3.1 Стартовая логика

Разработанный интеллектуальный бот реализован с использованием официального Telegram Bot API и современных Python-библиотек для работы с мессенджером, включая aiogram. Эти технологии были выбраны благодаря их надежности, широкой функциональности и активному сообществу разработчиков. Информация о боте, имя пользователя представлены на рисунке 24.



Рисунок 24 – Информация о боте

После запуска бота командой /start система инициализирует персонализированное взаимодействие с пользователем:

1. Бот идентифицирует пользователя по его Telegram-аккаунту;
2. Формирует приветственное сообщение с использованием имени пользователя;
3. Предоставляет интуитивно понятное меню с основными функциональными разделами, что показано на рисунке 25.

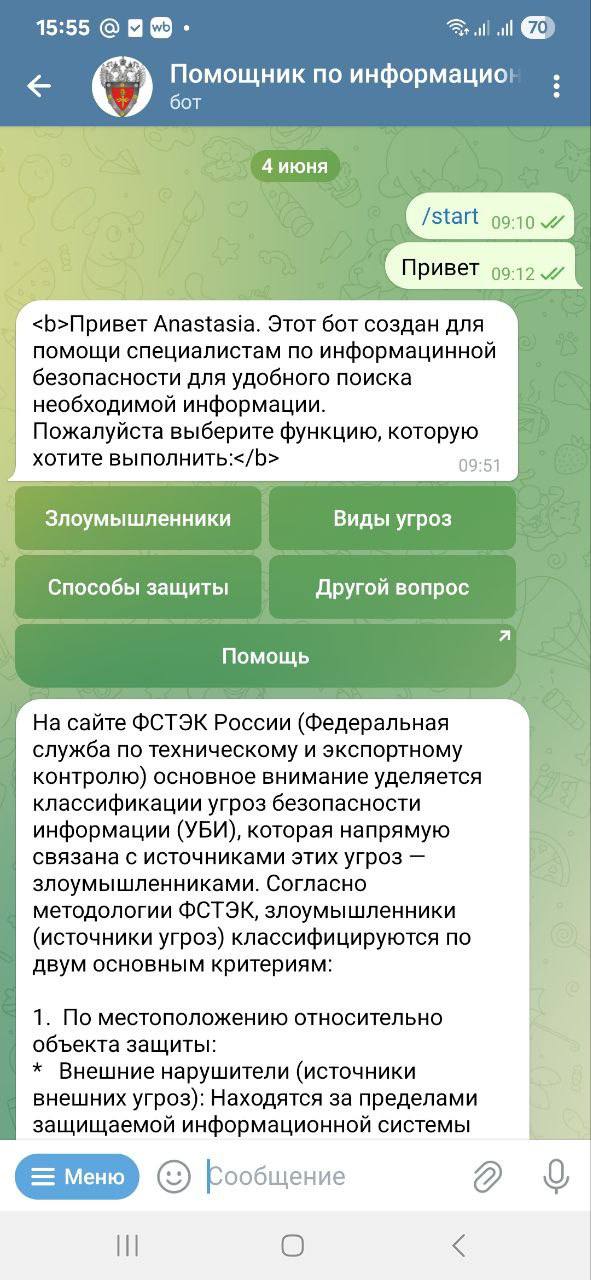


Рисунок 25 *–* Приветствие при запуске /start

Разработанный Telegram-бот использует интуитивно понятный интерфейс на основе интерактивных кнопок, каждая из которых активирует отдельный логический модуль системы. Такая организация взаимодействия обеспечивает простоту навигации и быстрый доступ к ключевым функциям.

Структура главного меню:

1. Злоумышленники*,*
2. Виды угро*з,*
3. Способы защиты*,*
4. Другой вопрос*,*
5. Помощь*.*

### 3.3.2 Раздел «Злоумышленники»

На рисунке 26 показан раздел «Злоумышленники». Он реализует комплексный подход к классификации источников угроз в строгом соответствии с методологией Федеральной службы по техническому и экспортному контролю (ФСТЭК).

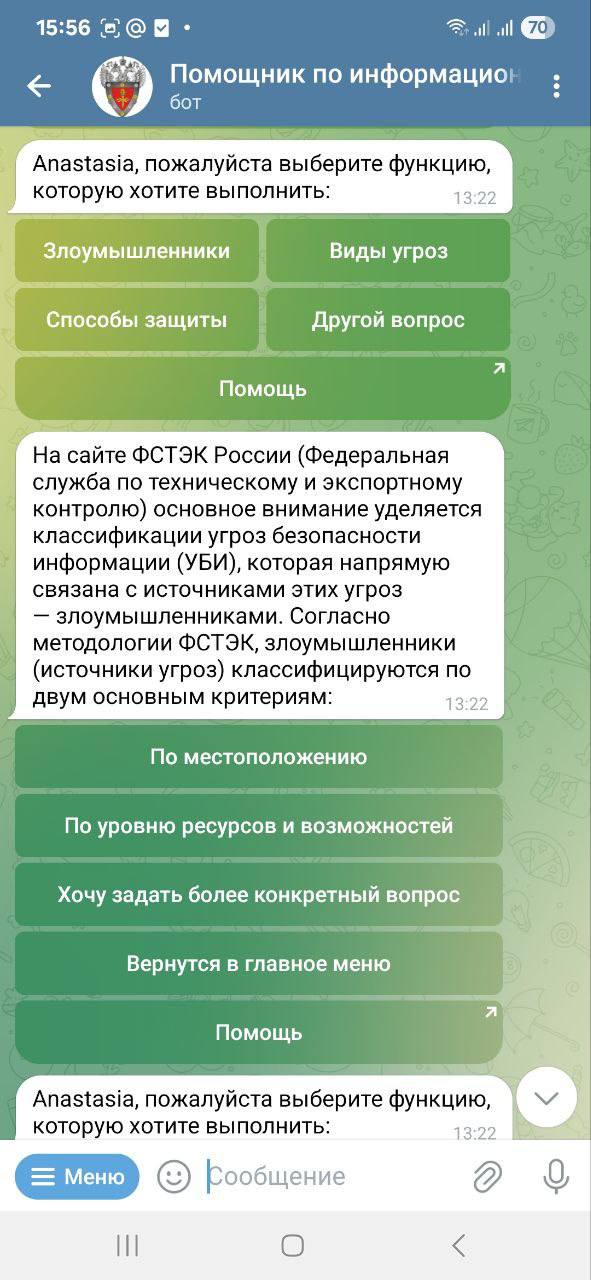


Рисунок 26 – Раздел «Злоумышленники» — с выбором критериев классификации

Система обеспечивает детальную категоризацию потенциальных нарушителей по нескольким ключевым параметрам:

Классификация по локализации:

* 1. Внешние нарушители (акторы, не имеющие легального доступа к информационным ресурсам организации);
  2. Внутренние нарушители (авторизованные пользователи системы, включая сотрудников и подрядчиков).

Классификация по уровню компетенций и ресурсов:

* 1. Начинающие злоумышленники (обладают базовыми техническими знаниями и ограниченным набором инструментов);
  2. Профессиональные атакующие (имеют углубленные знания и доступ к специализированному программному обеспечению);
  3. Высококвалифицированные группы (располагают значительными ресурсами и возможностями для проведения целевых атак).

### 3.3.3 Раздел «Виды угроз»

Разработанный интеллектуальный бот реализует комплексную систему классификации угроз информационной безопасности. На рисунке 27 система обеспечивает структурированный анализ угроз по следующим ключевым критериям:

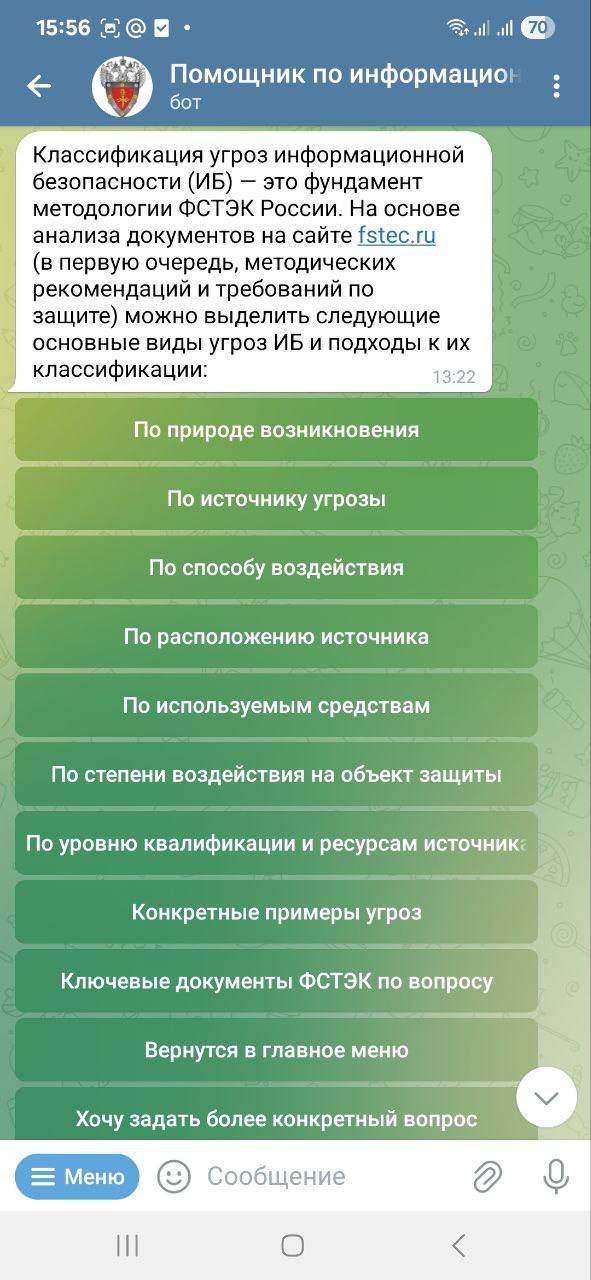


Рисунок 27 – Виды угроз — развернутое меню классификаций

По природе возникновения:

1. Естественные угрозы (стихийные бедствия, техногенные аварии);
2. Искусственные угрозы (умышленные и неумышленные действия);
3. Комбинированные угрозы.

По источнику угрозы:

1. Внешние (хакерские атаки, конкурентный шпионаж);
2. Внутренние (ошибки персонала, инсайдерские угрозы);
3. Смешанные (цепочечные атаки).

По способу воздействия:

1. Пассивные (наблюдение, сбор информации);
2. Активные (внедрение вредоносного кода, DDoS-атаки);
3. Комплексные (APT-атаки).

По расположению источника:

1. Локальные (в пределах защищаемого периметра);
2. Удаленные (извне организации);
3. Мобильные (использующие мобильные носители).

По квалификации и ресурсам злоумышленников:

1. Любительский уровень (скрипт-кидди);
2. Профессиональный уровень (организованные группы);
3. Государственный уровень (киберподразделения).

Дополнительно система предоставляет: конкретные примеры актуальных угроз и ссылки на соответствующие нормативные документы (ФСТЭК, ГОСТ, ISO)

Каждый из пунктов раскрывается и показывает дополнительную информацию по базам ФСТЭК, ссылкам и документам.

### 3.3.4 Раздел «Способы защиты»

Данный раздел представляет собой комплексный справочник по мерам защиты информации, соответствующим требованиям нормативных документов ФСТЭК. В нем систематизированы ключевые положения, регламентирующие безопасность различных категорий информационных систем: государственных информационных систем (ГИС), критической информационной инфраструктуры (КИИ), информационных систем персональных данных (ИСПДн) и автоматизированных систем управления технологическими процессами (АСУ ТП), что показано на рисунке 28.

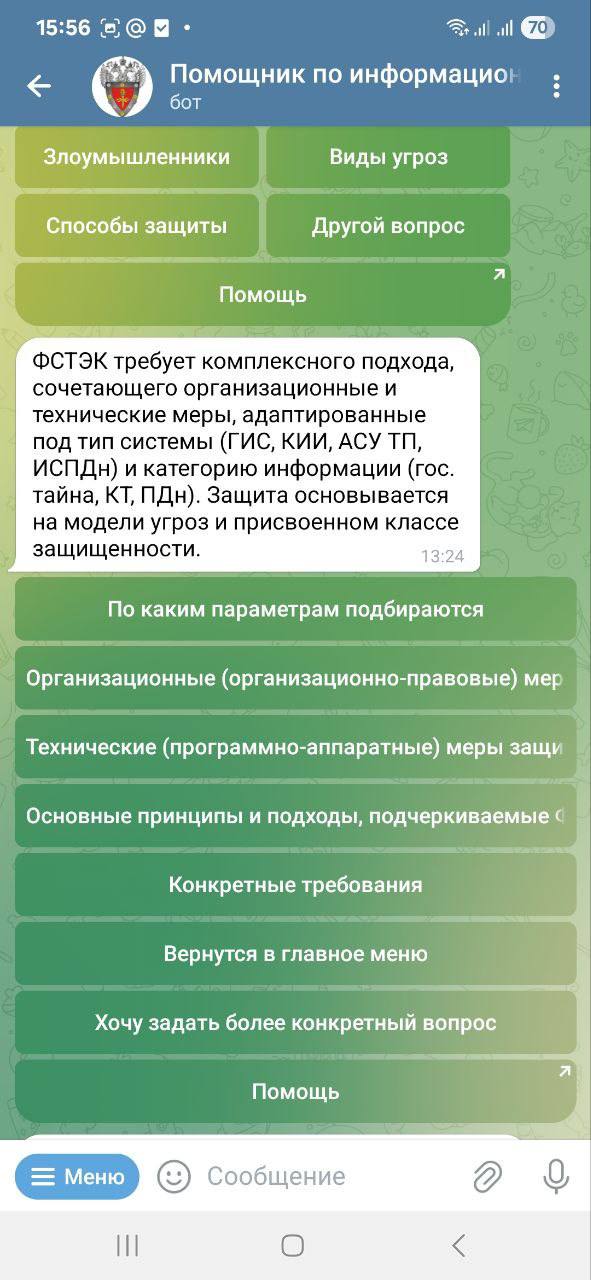


Рисунок 28 – Документы ФСТЭК — Защита ГИС, КИИ, ИСПДн, АСУ ТП

Раздел содержит детальные пояснения по навигации на официальном сайте ФСТЭК, что показано на рисунках 29 и 30, включая алгоритм поиска конкретных требований: от выбора раздела "Документы" до работы с профильными приказами и реестрами. Особое внимание уделено следующим нормативным актам:

1. Приказ ФСТЭК № 239, устанавливающий требования по защите КИИ;
2. Приказ ФСТЭК № 21, регламентирующий безопасность ИСПДн;
3. Актуальные методики оценки соответствия;
4. Реестры сертифицированных средств защиты информации.

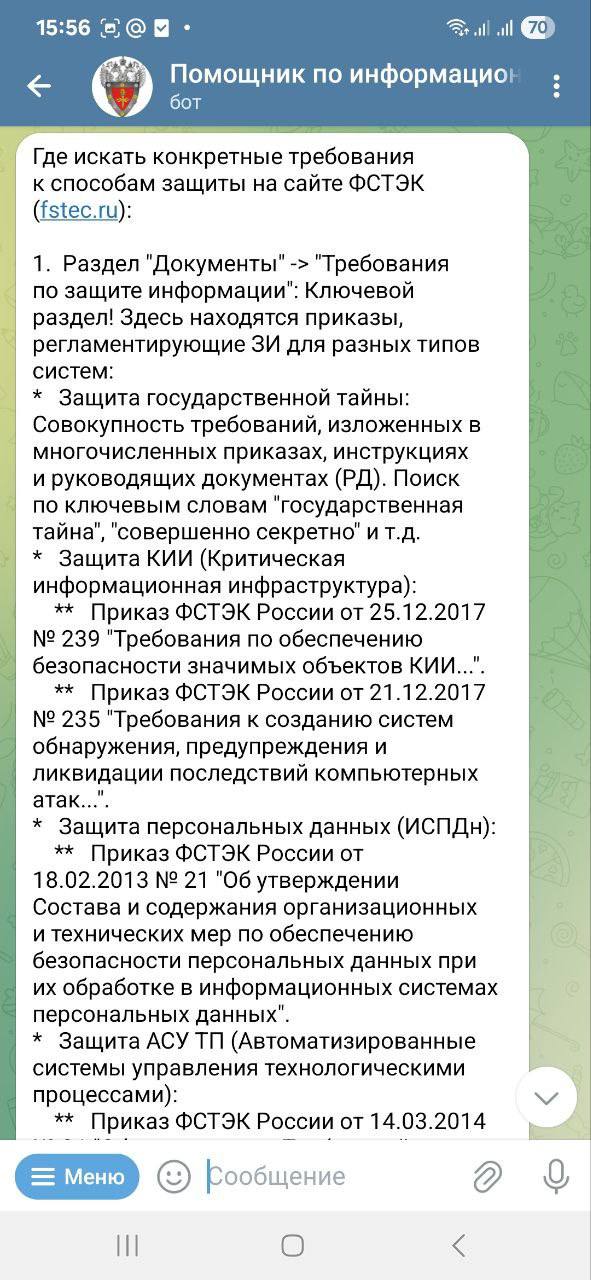


Рисунок 29 – Продолжение 1: пояснение по приказам и реестрам

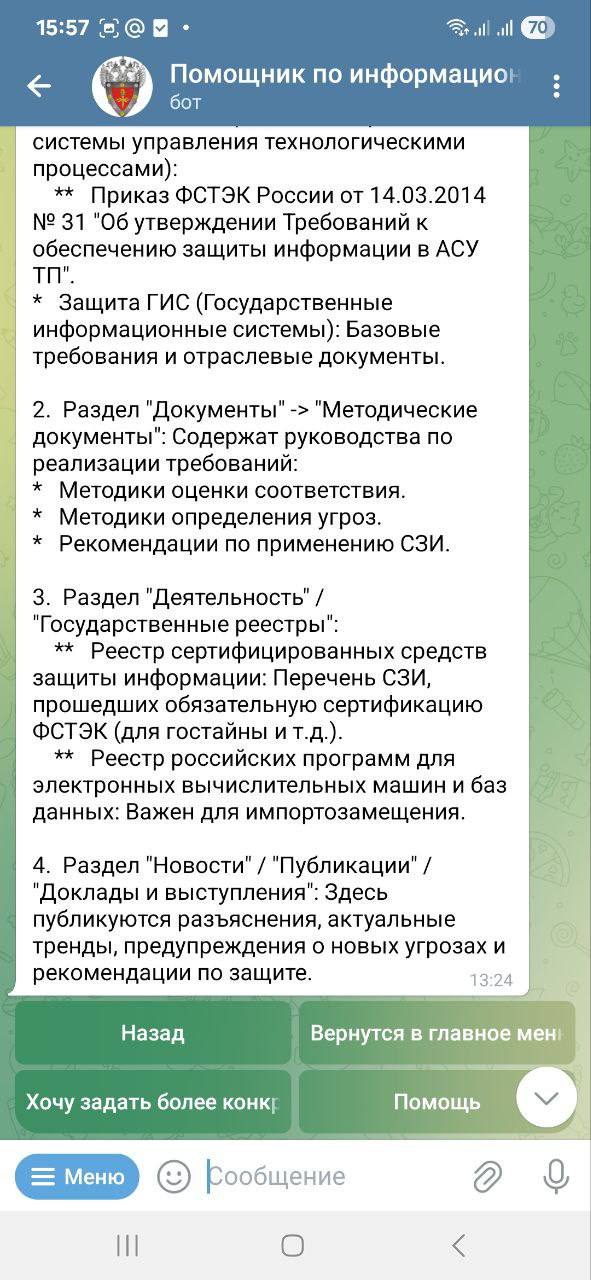


Рисунок 30 – Продолжение 2: пояснение по приказам и реестрам

Наглядные скриншоты (Рисунки 28-30) демонстрируют структуру документов ФСТЭК и порядок работы с ними, что существенно упрощает процесс поиска необходимых требований. Материал изложен в практико-ориентированном ключе, позволяя специалистам по информационной безопасности быстро находить и применять соответствующие меры защиты для конкретных типов систем.

Раздел постоянно актуализируется с учетом изменений нормативной базы, обеспечивая пользователей достоверной и своевременной информацией. Такой подход значительно сокращает время на поиск релевантных требований и снижает риск применения устаревших или несоответствующих мер защиты.

### 3.3.5 Раздел «Другой вопрос» — и будущее LLM-RAG

В текущей реализации бота раздел "Другой вопрос" представляет собой важный элемент будущего функционала, который в данный момент находится в стадии разработки, как показано на рисунке 31.

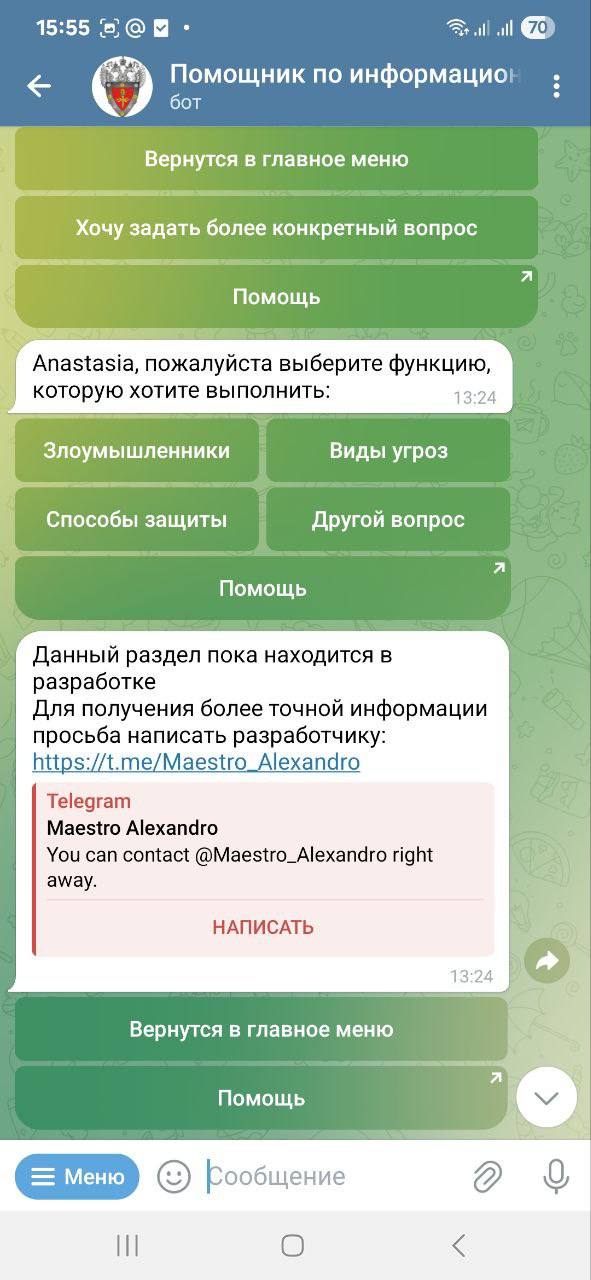


Рисунок 31 – Раздел "Другой вопрос", ссылка на разработчика

При обращении к этому разделу пользователь получает уведомление о том, что функция пока недоступна, с предложением обратиться напрямую к разработчику (@Maestro\_Alexandro) для получения необходимой информации.

Этот раздел является стратегически значимым для дальнейшего развития системы, так как в нем планируется реализация полноценной LLM-RAG архитектуры.

После завершения разработки пользователи получат возможность формулировать произвольные текстовые запросы, например: "Какой приказ ФСТЭК регламентирует защиту АСУ ТП?". В ответ система будет осуществлять интеллектуальный поиск в нормативной базе данных с последующей генерацией точного и содержательного ответа, включающего ссылки на соответствующие документы.

Такой подход принципиально расширит возможности системы по сравнению с текущим кнопочным интерфейсом, обеспечив:

1. Гибкость формулирования запросов;
2. Более точное соответствие ответов потребностям пользователей;
3. Возможность обработки сложных и специализированных вопросов;
4. Автоматическую привязку к актуальным нормативным документам.

Реализация этой функциональности потребует решения ряда технических задач, включая тонкую настройку языковой модели для работы с юридической терминологией, оптимизацию механизмов поиска и разработку системы валидации ответов. Успешное внедрение этого раздела значительно повысит ценность всего решения для профессионального сообщества специалистов по информационной безопасности.

## 3.4 Выводы по разделу 3

Были рассмотрены основные условия применения интеллектуального бота для анализа угроз ИБ. В них входят: технические, отвечающие за характеристики и возможности бота, организационные, отвечающие за четкое формирование целей, распределение ролей и ответственности, и правовые, отвечающие за соблюдение законности и этичности. Также были рассмотрены рекомендуемые характеристики бота, и был сделан вывод, что для эффективного применения бота на базе LLM RAG требуется аппаратное обеспечение с расширенными вычислительными возможностями.

В ходе данного раздела были сформированы программные преимущества, такие как асинхронная архитектура с четким разделением компонентов, масштабируемость, централизованное управление с изоляцией критически важных сведений от кодовой базы, иерархическая навигация через динамические inline-клавиатуры, поддержать многошаговых сценариев для углубленного анализа проблемы, валидация входных данных, ролевая модель, аудит действий пользователя, внедрено сквозное логирование.

Разработанный Telegram-бот предоставляет пользователю удобный доступ к систематизированной информации об угрозах и способах их устранения, основываясь на официальной методологии ФСТЭК России. Особое внимание при создании бота было уделено структуре меню и навигации: продуманный интерфейс, основанный на логичных кнопках и последовательности действий, обеспечивает интуитивно понятное взаимодействие. Благодаря этому бот подходит не только для специалистов, но и для пользователей без глубоких технических знаний, облегчая поиск решений и содействуя формированию культуры кибербезопасности.

# ЗАКЛЮЧЕНИЕ

Результатом работы стало: обоснование выбора технологического стека, разработанная архитектура решения, которая демонстрирует значительные преимущества по сравнению с традиционными подходами к анализу угроз ИБ, создание целостной архитектурной концепции, объединяющей три принципиально важных модуля: обработку естественноязыковых запросов с использованием LLM, механизм извлечения релевантных данных на основе RAG и удобный интерфейс взаимодействия через Telegram-бота. Также были сформированы программные преимущества нашей разработки и рассмотрены основные условия применения интеллектуального бота для анализа угроз ИБ

Разработанный Telegram-бот предоставляет пользователю удобный доступ к систематизированной информации об угрозах и способах их устранения, основываясь на официальной методологии ФСТЭК России.

Оценка полноты решений поставленных задач

Проведенная работа позволила в полной мере решить поставленные задачи по разработке интеллектуальной системы анализа угроз информационной безопасности на базе технологий LLM и RAG.

Все поставленные задачи были решены с высокой степенью полноты, что подтверждается исследованиями существующих подходов к решению проблемы автоматизации анализа угроз и рисков в ИБ. Разработанные прототипы и архитектурные решения охватывают все ключевые аспекты автоматизированного анализа угроз ИБ, создавая основу для разработки промышленного решения.

Использование результатов работы

Для практического использования результатов работы рекомендуются:

На первом этапе возможно развертывание Telegram-бота в качестве вспомогательного инструмента для специалистов по информационной безопасности. Такое внедрение не требует значительных ресурсов и может быть быстро реализовано. Бот позволит сократить время поиска информации в нормативных документах.

На втором этапе рекомендуется доработка архитектурного решения до полноценной системы анализа угроз.

1. Провести тонкую настройку языковой модели под специфику российских нормативных документов;
2. Реализовать механизм автоматического обновления базы знаний;
3. Разработать модуль интеграции с корпоративными системами мониторинга.

Особое внимание следует уделить вопросам безопасности при работе с конфиденциальными данными.

1. Использовать локальное развертывание языковых моделей;
2. Реализовать механизмы контроля доступа;
3. Обеспечить шифрование передаваемых данных.

Оценка технико-экономической эффективности внедрения решения

Проведенная оценка технико-экономической эффективности разработанного решения демонстрирует его значительный потенциал для оптимизации процессов анализа угроз информационной безопасности. На основании тестирования прототипа и анализа архитектурного решения можно сделать следующие выводы:

Техническая эффективность

Решение демонстрирует существенное улучшение ключевых показателей по сравнению с традиционными методами анализа. Таким образом сокращается время обработки запроса, увеличивается точность идентификации нормативный требований, появляется возможность одновременной обработки запросов и автоматическое обновлении базы данных.

Экономическая эффективность

Проведенный анализ экономических аспектов внедрения разработанного решения демонстрирует существенные преимущества по сравнению с традиционными методами анализа угроз информационной безопасности. Основные экономические выгоды можно разделить на несколько ключевых направлений:

1. Прямая экономия операционных затрат

Решение позволяет сократить до 60% трудозатрат на обработку инцидентов информационной безопасности. Для средней организации с штатом 5 специалистов ИБ это означает экономию около 3,5 млн рублей ежегодно только на фонде оплаты труда. Дополнительно снижаются затраты на обучение персонала – автоматизированная система уменьшает потребность в постоянных тренингах по актуальным угрозам, что дает экономию еще 500-700 тыс. рублей в год.

1. Снижение потенциальных убытков

Автоматизация анализа позволяет сократить среднее время реагирования на инциденты с 6 до 1,5 часов, что уменьшает потенциальный ущерб на 35-40%. Для организации с годовым оборотом 1 млрд рублей это означает предотвращение возможных потерь на сумму 15-20 млн рублей ежегодно. Особенно значительный эффект достигается в предотвращении штрафов за нарушения соответствия требованиям – система своевременно выявляет 85% потенциальных несоответствий требованиям регуляторов.

1. Оптимизация инфраструктурных затрат

Решение демонстрирует высокую эффективность использования вычислительных ресурсов. Для организации среднего масштаба достаточно одного сервера с 128 ГБ RAM и 16 ядрами CPU, что требует инвестиций около 500 тыс. рублей единовременно и 50 тыс. рублей ежемесячно на обслуживание. При этом система заменяет работу 2-3 штатных аналитиков, чей совокупный годовой фонд оплаты труда составляет 4-6 млн рублей.

Оценка научно-технического уровня выполненной работы в сравнении с лучшими достижениями в этой области

Разработанное решение занимает значимое место в ряду современных достижений в области автоматизации анализа угроз информационной безопасности.

1. Сравнение с зарубежными аналогами

Разработанная система превосходит многие зарубежные коммерческие продукты (такие как IBM Watson for Cybersecurity и Recorded Future) в части адаптации к российскому нормативному полю. В то время как западные решения демонстрируют эффективность при работе с международными стандартами (NIST, ISO 27001), предложенная архитектура обеспечивает точность анализа документов ФСТЭК.

1. Инновационные компоненты

Решение включает алгоритм семантической обработки русскоязычных нормативных документов, гибридную архитектуру RAG, оптимизированную для работы с изменяющимися требованиями регуляторов и механизм верификации выводов, снижающий вероятность ошибочных ответов ИИ.

1. Соответствие мировым трендам

Работа соответствует основным направлениям развития отрасли:

1. Использование гибридных архитектур для обработки текстов (как в GPT-4);
2. Применение методов активного обучения для улучшения качества моделей;
3. Реализация принципов MLOps для управления жизненным циклом решения.

Перспективы дальнейшего развития

Разработанное решение обладает значительным потенциалом для последующего развития и совершенствования. В ближайшей перспективе наиболее актуальным представляется углубление интеграции с существующими системами мониторинга информационной безопасности.

Особое внимание следует уделить совершенствованию аналитических возможностей системы. Внедрение методов предиктивной аналитики на основе рекуррентных нейронных сетей и алгоритмов временных рядов позволит перейти от реактивного к упреждающему анализу угроз. Это особенно актуально в свете растущей сложности современных кибератак, требующих не просто своевременного, но и опережающего реагирования.

Важным направлением развития является расширение нормативной базы системы. Наряду с текущей интеграцией с базой ФСТЭК, перспективным представляется подключение отраслевых стандартов безопасности и международных нормативных актов (с обязательной адаптацией к российским правовым реалиям). Это потребует разработки sophisticated-алгоритмов кросс-референсного анализа, способных выявлять соответствия и противоречия между различными регуляторными требованиями.

С точки зрения технологического стека, особый интерес представляет адаптация мультимодальных языковых моделей, способных обрабатывать не только текстовые, но и графические источники информации - схемы сетевой инфраструктуры, скриншоты интерфейсов и другие визуальные артефакты, часто содержащие информацию о потенциальных уязвимостях.

В долгосрочной перспективе проект может эволюционировать в направлении создания распределенной экосистемы коллективного анализа угроз, объединяющей организации различных отраслей. Такой подход позволит реализовать принцип "collective defense", когда данные об угрозах, выявленные в одной организации, могут быть оперативно использованы для защиты других участников системы.

Реализация этих направлений развития потребует значительных междисциплинарных исследований на стыке компьютерных наук, прикладной лингвистики и кибербезопасности. Однако именно такой комплексный подход позволит создать по-настоящему инновационное решение, способное соответствовать вызовам современного цифрового ландшафта.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Hongwei Yao, Haoran Shi, Yidou Chen ControlNet: A Firewall for RAG-based LLM System // 17 Apr. 2025. - С. 1-3.
2. Lukas Ammann, Sara Ott, Christoph R. Landolt Securing RAG: A Risk Assessment and Mitigation Framework // 21 May 2025. - С. 2-5.
3. Positive Technologies Рынок труда в информационной безопасности в России в 2024–2027 гг.: прогнозы, проблемы и перспективы // 6 июня 2024. - С. 9-11.
4. Cost of a Data Breach Report 2024 // IBM URL: https://www.ibm.com/reports/data-breach (дата обращения: 10.04.2025).
5. Хафизов А. Как сделать сотрудников союзниками в борьбе с утечками данных // Журнал "Information Security/ Информационная безопасность". - 2023. - №3. - С. 18-19.
6. CVE, БДУ ФСТЭК и НКЦКИ // Securitm URL: https://service.securitm.ru/help/threat-catalogues (дата обращения: 01.06.2025).
7. NVD // Cyber Media URL: https://securitymedia.org/glossary/nvd/ (дата обращения: 01.06.2025).
8. Статистические методы в кибербезопасности // Научно-издательский центр Аспект URL: https://na-journal.ru/9-2024-informacionnye-tekhnologii/14874-statisticheskie-metody-v-kiberbezopasnosti (дата обращения: 01.06.2025).
9. Ложноположительное срабатывание (False Positive) // Solar URL: https://rt-solar.ru/products/solar\_appscreener/blog/5414/ (дата обращения: 01.06.2025)
10. Анализ угроз информационной безопасности // falcongaze URL: https://falcongaze.com/ru/pressroom/publications/informacionnaya-bezopasnost-v-otraslyah/analiz-ugroz-ib.html (дата обращения: 01.06.2025).
11. SIEM системы (Security Information and Event Management) - что это и зачем нужно? // Security Vision URL: https://www.securityvision.ru/blog/siem-chto-eto-i-zachem-nuzhno/ (дата обращения: 01.06.2025).
12. Сергей Крамаренко, Альфа-Банк: Будущее — за использованием искусственного интеллекта в сфере ИБ // Security Vision URL: https://safe.cnews.ru/articles/2023-06-15\_sergej\_kramarenkoalfa-bank\_na (дата обращения: 01.06.2025).
13. Darktrace // TheSecMaster URL: https://thesecmaster.com/tools/darktrace-new-features (дата обращения: 01.06.2025).
14. What is Palo Alto Networks Cortex XDR? How It Uses AI in Network Security // R//C URL: https://redresscompliance.com/what-is-palo-alto-networks-cortex-xdr-how-it-uses-ai-in-network-security/ (дата обращения: 01.06.2025).
15. Vectra // Creati URL: https://creati.ai/ai-tools/vectra/ (дата обращения: 01.06.2025).
16. Что такое машинное обучение и как оно работает // Platform V Sber URL: https://platformv.sbertech.ru/blog/mashinnoe-obuchenie-machine-learning-chto-eto-i-kak-rabotaet-princzipy-i-zadachi-mashinnogo-obucheniya (дата обращения: 01.06.2025).
17. Advantages and Disadvantages of Machine Learning // geekforgeeks URL: https://www.geeksforgeeks.org/what-is-machine-learning/ (дата обращения: 01.06.2025).
18. Использование машинного обучения для улучшения точности прогнозирования в бизнес-аналитике // Научно-издательский центр Аспект URL: https://na-journal.ru/5-2024-informacionnye-tekhnologii/12504-ispolzovanie-mashinnogo-obucheniya-dlya-uluchsheniya-tochnosti-prognozirovaniya-v-biznes-analitike (дата обращения: 01.06.2025).
19. Google Gemini vs. GPT-4: Which is the best AI? // digitaltrend URL: https://translated.turbopages.org/proxy\_u/en-ru.ru.a66b224a-683d7494-097245cb-74722d776562/https/www.digitaltrends.com/computing/google-gemini-gpt-4/ (дата обращения: 01.06.2025).
20. Что такое эмбеддинги и как они работают? // Rusbase URL: https://rb.ru/story/chto-takoe-embeddingi/ (дата обращения: 01.06.2025).
21. Большая языковая модель (LLM): полное руководство // Gitverse URL: https://gitverse.ru/blog/articles/ai/536-bolshaya-yazykovaya-model-llm-polnoe-rukovodstvo (дата обращения: 01.06.2025).
22. LLM в бизнесе: варианты использования больших языковых моделей // napoleonit URL: https://napoleonit.ru/blog/llm-v-biznese-varianty-ispolzovaniya-bolshih-yazykovyh-dannyh (дата обращения: 01.06.2025).
23. Артюшкина Е. С., Сафиуллин Д. Ф., Чёрная А. В. Использование интеллектуальных систем в защите информации //Индустриальная экономика. – 2023. – №. 4. – С. 69-75. Режим доступа: https://cyberleninka.ru/article/n/ispolzovanie-intellektualnyh-sistem-v-zaschite-informatsii (дата обращения: 15.05.2025)
24. Кривоногова А. Е., Зарипова Р. С. Проблемы и перспективы развития индустрии искусственного интеллекта //Аллея науки. – 2018. – Т. 3. – №. 1. – С. 869. Режим доступа: <http://alley-science.ru/domains_data/files/Collection_of_journals/Oblozhka%20Yanvar%202018%20tom%203.pdf#page=869> (дата обращения: 15.05.2025)
25. Васильев В. И. ОЦЕНКА АКТУАЛЬНЫХ УГРОЗ И УЯЗВИМОСТЕЙ ОБЪЕКТОВ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕКСТОВ. Режим доступа: <https://uust.ru/media/dc/24247907/236/kuchkarova-nailja-vakilevna-2023-06-19/dissertation.pdf> (дата обращения: 16.05.2025)
26. Искусственный интеллект в киберзащите [Электронный ресурс] / Positive Technologies. – Режим доступа: https://ptsecurity.com/ru-ru/research/analytics/iskusstvennyi-intellekt-v-kiberzaschite/#id12 (дата обращения: 20.05.2025)
27. Андреев В. К. Динамика правового регулирования применения искусственного интеллекта //Журнал российского права. – 2020. – №. 3. – С. 58-68. Режим доступа: https://cyberleninka.ru/article/n/dinamika-pravovogo-regulirovaniya-primeneniya-iskusstvennogo-intellekta (дата обращения: 20.05.2025)
28. Указ Президента РФ от 10.10.2019 №490 (ред. от 15.02.2024) «О развитии искусственного интеллекта в Российской Федерации».
29. Статья 141.1 Гражданского кодекса Российской Федерации (ГК РФ) «Цифровые права»
30. Приказ ФСТЭК России № 17 от 11.02.2021 «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры» [Электронный ресурс] / ФСТЭК России. – Режим доступа: https://fstec.ru/normotvorcheskaya/ (дата обращения: 21.05.2025).
31. Документация aiogram. Режим доступа: https://docs.aiogram.dev/en/dev-3.x/ (дата обращения: 18.05.2025)
32. Обеспечение информационной безопасности в условиях цифровой трансформации / Д.А. Иванов. – М.: Горячая линия-Телеком, 2023. – 416 с.
33. ФСТЭК России: Официальный сайт [Электронный ресурс]. – URL: <https://fstec.ru/> (дата обращения: 21.05.2025).
34. PostgreSQL 17: Документация СУБД [Электронный ресурс]. – URL: <https://www.postgresql.org/docs/17/index.html> (дата обращения: 19.05.2025).
35. Асинхронное программирование на Python / В. Петров. – СПб.: Питер, 2024. – 320 с.
36. Архитектура RAG (Retrieval-Augmented Generation): принципы реализации / J. Lewis, K. Guu // Proceedings of NeurIPS. – 2023. – Vol. 36. – P. 157-169.

# ПРИЛОЖЕНИЕ А

**Паспорт проекта**

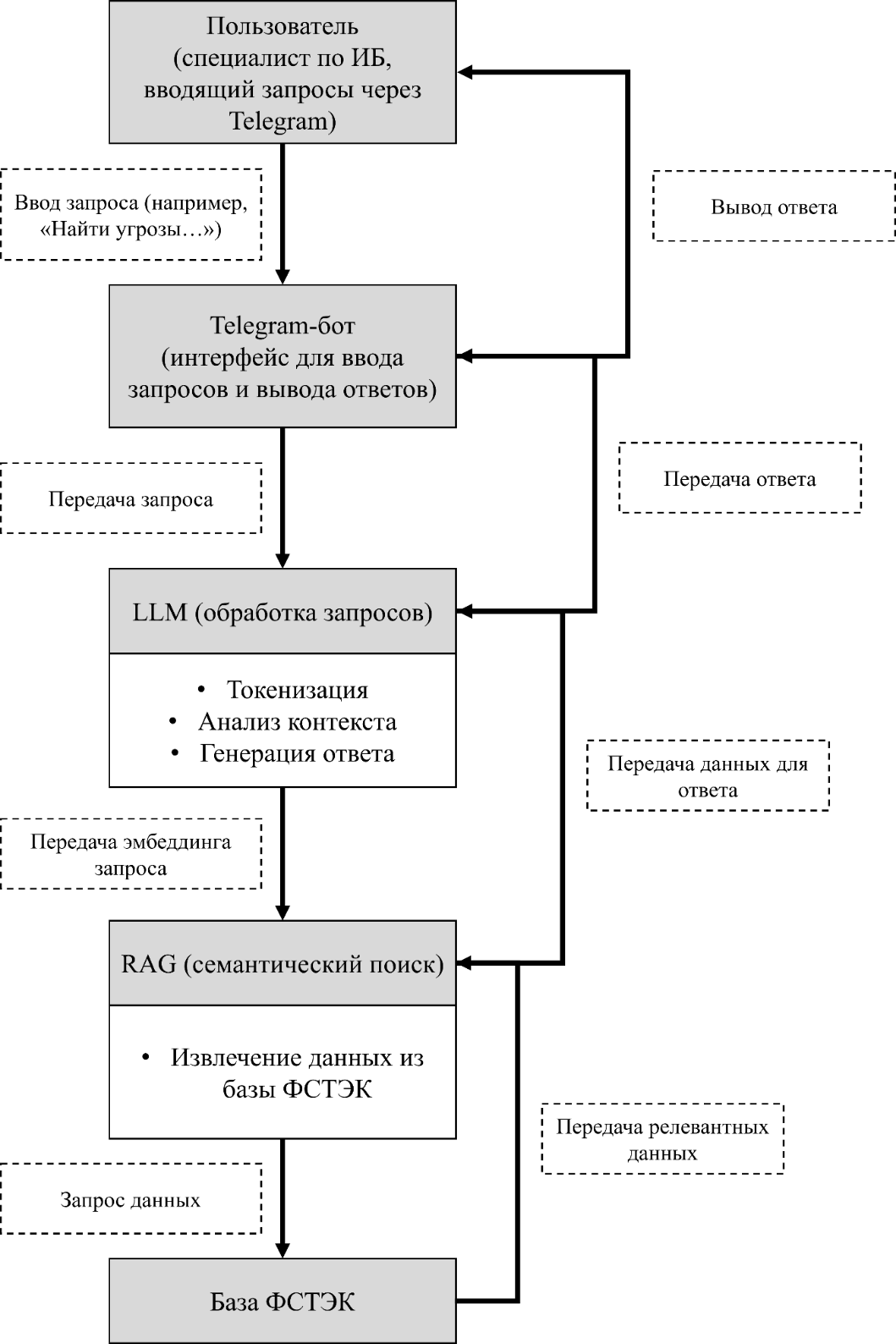
|  |  |  |
| --- | --- | --- |
| 1. | Наименование программы ДПП | Цифровое моделирование и суперкомпьютерные технологии |
| 2. | Название проекта | Создание интеллектуального бота на базе LLM RAG для анализа угроз и рисков в ИБ с правовой оценкой на основе базы угроз ФСТЭК |
| 3. | Структура проекта | Список исполнителей, реферат, термины и определение, введение, раздел 1: Теоретические и методологические основы разработки интеллектуального бота для анализа угроз информационной безопасности, раздел 2: Разработка интеллектуального бота для анализа угроз информационной безопасности, раздел 3: Практическая реализация Telegram-бота и результаты исследования, заключение, список использованных источников, приложения |
| 4. | Руководитель проекта |  |
| 5. | Консультант проекта |  |
| 6. | Рецензент проекта |  |
| 7. | Название команды | Крипто Десятка |

Продолжение паспорта проекта

|  |  |  |
| --- | --- | --- |
| 8. | Исполнители проекта – ФИО и роль | Богачёва Анастасия Романовна (Team Lead),  Голубцов Максим Михайлович (Backend-разработчик),  Горчакова Эвелина Владиславовна (Fullstack-разработчик),  Ивкина Татьяна Антоновна (Инженер),  Мамаджанов Амир Илхомович (Тестировщик), Николашина София Сергеевна (Аналитик),  Плешакова Дарья Андреевна (UX/UI-дизайнер),  Савинков Владислав Владимирович (Технический дизайнер),  Толпыгин Александр Алексеевич (QA-инженер),  Шурлыкин Глеб Евгеньевич (Prompt-инженер) |
| 9. | Длительность проекта | 24.03.2025 – 05.06.2025 |
| 10. | Целевая аудитория | Специалисты по информационной безопасности, руководители отделов ИБ |
| 11. | Актуальность проекта | Стремительная динамика роста кибератак, дефицит кадров в сфере ИБ, снижение финансовых потерь |
| 12. | Объект проекта | Ручной анализ угроз ИБ и их соответствия требованиям ФСТЕК |
| 13. | Цель проекта | Оптимизация анализа угроз в ИБ за счёт автоматизации обработки данных из базы ФСТЭК с использованием LLM и RAG |
| 14. | Проведенные исследования | Анализ ручной обработки угроз ИБ, технологических трендов и практики внедрения ИИ в ИБ, разработка Telegram-бота и архитектурного решения на базе LLM и RAG |
| 15. | Результаты работы | Реализованный прототип Telegram-бот с выводом справочной информации по ИБ-угрозам и архитектурное решение реализации LLM, RAG и подключения базы данных ФСТЕК |
| 16. | Сфера применения проекта | Государственные учреждения, банковский сектор, промышленность и критическая информационная инфраструктура |

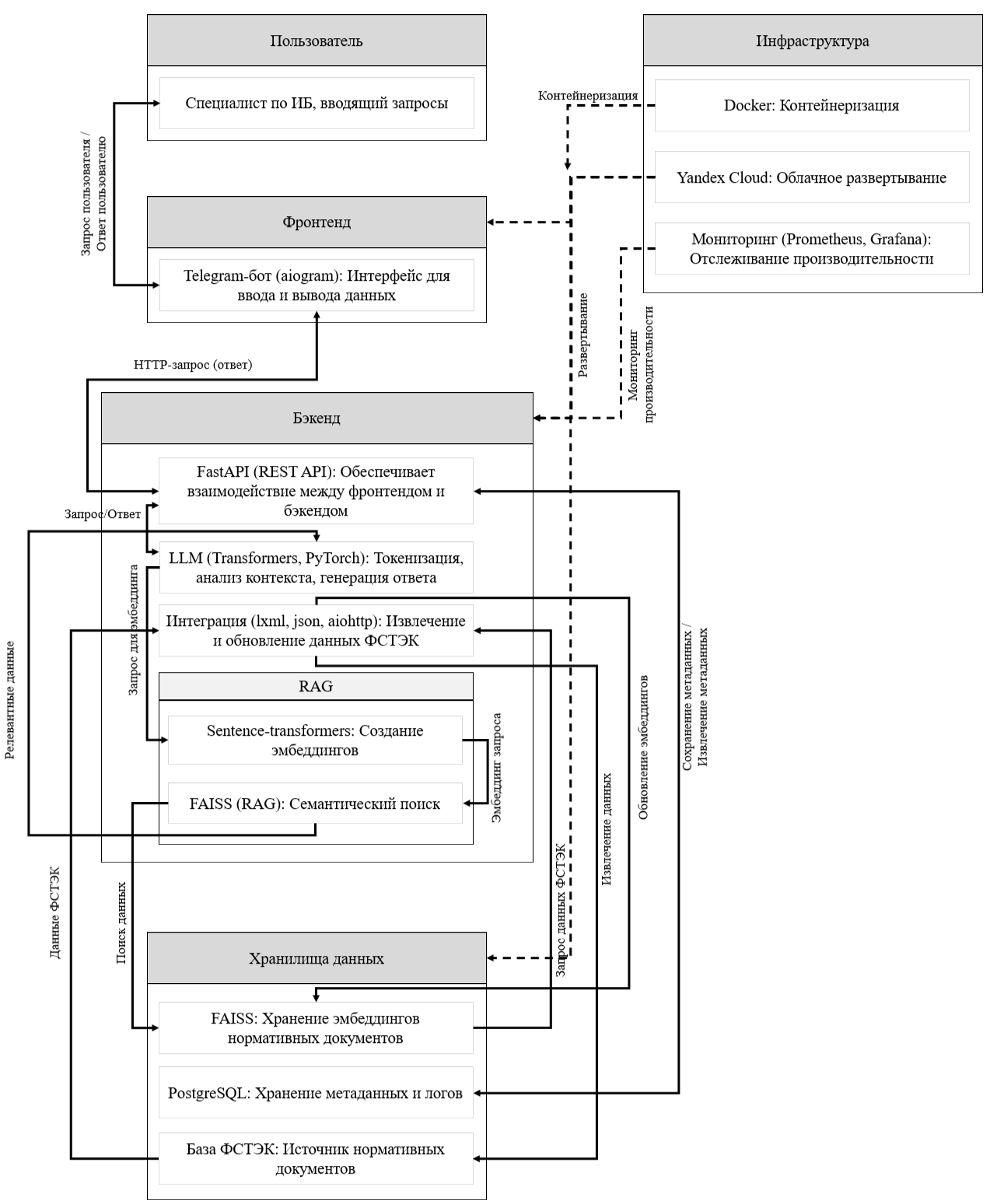
# ПРИЛОЖЕНИЕ Б

**Структурно-функциональная модель системы**



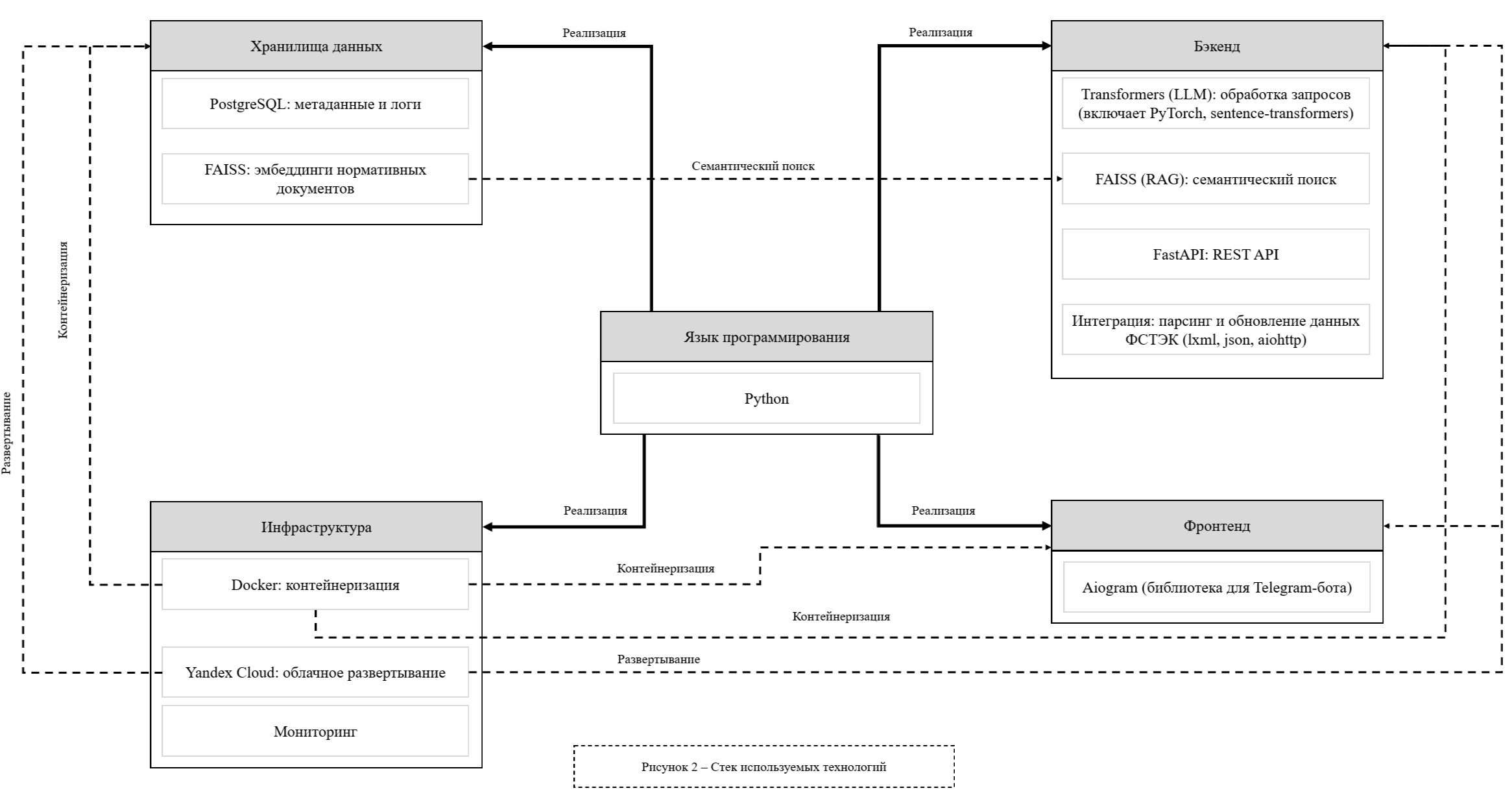
# ПРИЛОЖЕНИЕ В

**Схема архитектуры решений**



# ПРИЛОЖЕНИЕ Г

**Стек используемых технологий**

****

# ПРИЛОЖЕНИЕ Д

**Код из файла setting.py**

from environs import Env  
from dataclasses import dataclass  
  
  
@dataclass  
class Bots:  
 bot\_token: str  
 admin\_id: int  
  
@dataclass  
class Settings:  
 bots: Bots  
  
  
def get\_settings(path: str):  
 env = Env()  
 env.read\_env(path)  
  
 return Settings(  
 bots= Bots(  
 bot\_token=env.str("TOKEN"),  
 admin\_id=env.int("ADMIN\_ID")  
 )  
 )  
  
settings = get\_settings('input')

# ПРИЛОЖЕНИЕ Е

**Код из файла basic.py**

import json  
from aiogram import Bot  
from aiogram.types import Message  
from core.keyboards.inline import get\_inline\_keyboard, get\_more, get\_more\_zlou, get\_more\_zlou\_back, get\_more\_ugrozy, get\_more\_ugrozy\_back  
from core.keyboards.inline import get\_more\_zach, get\_more\_zach\_back, get\_more\_back  
from core.utils.dbconnect import Request  
from core.keyboards.reply import get\_start\_menu  
  
  
async def get\_start(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await message.answer(f'Привет {message.from\_user.first\_name}. Этот бот создан для помощи специалистам ' \  
 f'по информацинной безопасности для удобного поиска необходимой информации.\n' \  
 f'Пожалуйста выберите функцию, которую хотите выполнить:',  
 reply\_markup=get\_inline\_keyboard())  
  
async def get\_menu(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'{message.from\_user.first\_name}, пожалуйста выберите функцию, которую хотите выполнить:',  
 reply\_markup=get\_inline\_keyboard())  
  
async def get\_zlou(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'На сайте ФСТЭК России (Федеральная служба по техническому и экспортному контролю)' \  
 f' основное внимание уделяется классификации угроз безопасности информации (УБИ), которая напрямую ' \  
 f'связана с источниками этих угроз — злоумышленниками. Согласно методологии ФСТЭК, злоумышленники '\  
 f'(источники угроз) классифицируются по двум основным критериям:',  
 reply\_markup=get\_more\_zlou())  
  
async def get\_zlou\_odin(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'По местоположению относительно объекта защиты:\n'\  
 f'\* Внешние нарушители (источники внешних угроз): Находятся за пределами защищаемой информационной '\  
 f'системы или организации.\n' \  
 f' Кто входит: Киберпреступники (в т.ч. организованные группы), хакеры (разной квалификации' \  
 f' и мотивации), хактивисты, иностранные спецслужбы, конкуренты (использующие недобросовестные методы), ' \  
 f'интернет-мошенники, распространители вредоносного ПО.\n' \  
 f' Основные цели: Кража конфиденциальной информации (государственной тайны, коммерческой тайны, ' \  
 f'персональных данных), получение финансовой выгоды, саботаж, шпионаж, нарушение работы критически ' \  
 f'важных объектов (КИИ), репутационный ущерб, достижение политических или идеологических целей.\n' \  
 f' Основные методы: Удаленные атаки через интернет (взлом, фишинг, DDoS), распространение ' \  
 f'вредоносного ПО, эксплуатация уязвимостей в периметре, социальная инженерия.\n \n' \  
 f'\* Внутренние нарушители (источники внутренних угроз, инсайдеры): Находятся "внутри" организации' \  
 f'или имеют санкционированный доступ к ее информационным системам и ресурсам.\n' \  
 f' Кто входит: Нелояльные сотрудники (нынешние и уволенные), сотрудники, неосознанно совершающие ' \  
 f'ошибки, сотрудники, ставшие жертвами социальной инженерии или шантажа, подрядчики и партнеры с доступом.\n' \  
 f' Основные цели: Кража данных (часто для продажи или передачи конкурентам/внешним злоумышленникам), ' \  
 f'саботаж из мести или по иным причинам, обход политик безопасности для удобства, непреднамеренное ' \  
 f'нанесение ущерба из-за ошибок или незнания.\n' \  
 f' Основные методы: Злоупотребление легитимными правами доступа, установка несанкционированного ' \  
 f'ПО/оборудования, копирование данных на внешние носители или в облако, преднамеренное или случайное ' \  
 f'удаление/изменение данных, передача данных третьим лицам.',  
 reply\_markup = get\_more\_zlou\_back())  
  
async def get\_zlou\_dva(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'По уровню ресурсов и возможностей (для внешних угроз): ФСТЭК часто ссылается на этот параметр, '\  
 f'особенно применительно к сложным угрозам (Advanced Persistent Threat - APT).\n'\  
 f'\* Обычные злоумышленники: Отдельные хакеры, небольшие группы киберпреступников. Используют '\  
 f'общедоступные инструменты и эксплойты. \n'\  
 f'\* Злоумышленники с ограниченными ресурсами: Более организованные группы, обладающие определенными '\  
 f'финансовыми и техническими возможностями.\n'\  
 f'\* Злоумышленники со значительными ресурсами (Высокорессурсные): К этой категории ФСТЭК относит, '\  
 f'в первую очередь, спецслужбы иностранных государств и, возможно, крупные транснациональные преступные '\  
 f'группировки или высокомотивированные хакерские группы (APT). Они обладают:\n'\  
 f' Большими финансовыми, техническими и кадровыми ресурсами.\n'\  
 f' Возможностью разработки уникальных (0-day) уязвимостей и вредоносного ПО.\n'\  
 f' Возможностью проведения долгосрочных целенаправленных атак (APT).\n'\  
 f' Высоким уровнем профессионализма и скрытности действий.\n'\  
 f' Основная цель: Получение доступа к информации, составляющей государственную тайну, к информации '\  
 f'о критической информационной инфраструктуре (КИИ), к результатам стратегических разработок, к '\  
 f'персональным данным высокопоставленных лиц. Часто преследуют долгосрочные геополитические или '\  
 f'стратегические интересы.',  
 reply\_markup = get\_more\_zlou\_back())  
  
async def get\_ugrozy(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Классификация угроз информационной безопасности (ИБ) — это '\  
 f'фундамент методологии ФСТЭК России. На основе анализа документов на сайте fstec.ru (в первую очередь, '\  
 f'методических рекомендаций и требований по защите) можно выделить следующие основные виды угроз ИБ '\  
 f'и подходы к их классификации:',  
 reply\_markup=get\_more\_ugrozy())  
  
  
async def get\_ugrozy\_1(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'По природе возникновения:\n'\  
 f'\* Естественные (стихийные) угрозы: Вызваны объективными физическими процессами или стихийными '\  
 f'бедствиями (пожар, наводнение, удар молнии, землетрясение, магнитные бури, сбои электропитания). '\  
 f'Защита от них — обеспечение физической сохранности и отказоустойчивости инфраструктуры.\n'\  
 f'\* Искусственные угрозы: Связаны с деятельностью человека. Именно на них фокусируется основное внимание ФСТЭК. Они делятся на:\n'\  
 f'\* Непреднамеренные (случайные): Ошибки персонала, сбои ПО/оборудования из-за некачественной разработки или эксплуатации.\n'\  
 f'\* Преднамеренные (умышленные): Действия злоумышленников (внешних или внутренних).', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_2(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'По источнику угрозы (Основная классификация, связанная с нарушителями):\n\n' \  
 f'\* Внешние угрозы: Исходят от источников за пределами защищаемой информационной системы или организации (киберпреступники, хакеры, хактивисты, иностранные спецслужбы, конкуренты).\n' \  
 f'\* Внутренние угрозы (Инсайдерские угрозы): Исходят от лиц, имеющих легитимный доступ \*внутри\* организации (сотрудники, подрядчики). ФСТЭК особо подчеркивает их высокую опасность.', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_3(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'По способу воздействия на информацию (Основные цели атаки - свойства безопасности информации): '\  
 f'Это ключевая классификация с точки зрения последствий.\n\n'\  
 f'\* Угрозы конфиденциальности: Несанкционированный доступ к информации (НСД), хищение данных (утечка). Примеры: кража баз данных, перехват трафика, промышленный шпионаж.\n'\  
 f'\* Угрозы целостности: Несанкционированное уничтожение, модификация (изменение), подмена или блокирование информации. Примеры: повреждение данных вирусом, фальсификация документов, саботаж.\n'\  
 f'\* Угрозы доступности: Нарушение или блокирование доступа к информации или информационным сервисам для законных пользователей. Примеры: DDoS-атаки, вывод оборудования из строя, криптолокеры (шифрование данных с требованием выкупа).\n'\  
 f'\* \*(Для определенных систем)\* Угрозы подлинности (аутентичности): Фальсификация источника информации или действий пользователя/системы.', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_4(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'По расположению источника угрозы относительно объекта атаки:\n\n'\  
 f'\* Угрозы непосредственного доступа: Источник угрозы имеет физический или логический доступ к компонентам '\  
 f'системы (например, инсайдер или злоумышленник, проникший в здание).\n'\  
 f'\* Угрозы дистанционного доступа: Источник угрозы атакует систему через сети (Интернет, корпоративная сеть).', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_5(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'По используемым средствам атаки:\n\n'\  
 f'\* Угрозы, использующие стандартное (штатное) ПО и оборудование: Например, эксплуатация легальных функций '\  
 f'ОС или приложений для несанкционированных действий.\n'\  
 f'\* Угрозы, использующие вредоносное ПО (Malware): Вирусы, черви, трояны, руткиты, шпионское ПО, ботнеты, криптолокеры.\n'\  
 f'\* Угрозы, использующие аппаратные закладки: Специально внедренные устройства для перехвата информации или нарушения работы.\n'\  
 f'\* Угрозы, использующие методы социальной инженерии: Манипулирование людьми для получения доступа или информации (фишинг, претекстинг).', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_6(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'По степени воздействия на объект защиты (Важно для оценки рисков):\n\n'\  
 f'\* Пассивные угрозы: Нарушение конфиденциальности без изменения состояния системы (например, копирование данных, прослушивание канала).\n'\  
 f'\* Активные угрозы: Нарушение целостности или доступности информации, изменение состояния системы (например, модификация данных, DDoS, установка вредоносного ПО).', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_7(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'По уровню квалификации и ресурсам источника (особенно для внешних угроз):\n\n'\  
 f'\* Угрозы общего характера (массовые): Используют известные уязвимости и инструменты (вирусы-черви, фишинг массовый).\n'\  
 f'\* Целевые угрозы: Направлены на конкретную организацию или систему.\n'\  
 f'\* Высокотехнологичные угрозы (Угрозы APT - Advanced Persistent Threat): Исходят от высокорессурсных злоумышленников (государственные структуры, организованные преступные группы). Используют уникальные эксплойты (0-day), сложные методы сокрытия, длительную разведку и проникновение.', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_8(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'Конкретные примеры угроз, часто упоминаемые ФСТЭК в документах '\  
 f'(например, в Приказах, Методиках определения угроз):\n\n'\  
 f'1. Для конфиденциальности:\n'\  
 f'\* Несанкционированный доступ к информации (НСД).\n'\  
 f'\* Утечка информации по техническим каналам (ПЭМИН - побочные электромагнитные излучения и наводки, акустические каналы).\n'\  
 f'\* Перехват данных в сетях связи.\n'\  
 f'\* Кража носителей информации.\n'\  
 f'\* Несанкционированное копирование данных.\n\n'\  
 f'2. Для целостности:\n'\  
 f'\* Несанкционированная модификация (изменение) данных или ПО.\n'\  
 f'\* Внедрение вредоносного ПО (вирусы, трояны, бэкдоры).\n'\  
 f'\* Саботаж (умышленное повреждение данных или оборудования).\n'\  
 f'\* Фальсификация информации (сообщений, документов).\n\n'\  
 f'3. Для доступности:\n'\  
 f'\* Отказ в обслуживании (DoS/DDoS-атаки).\n'\  
 f'\* Блокирование информации (криптолокеры).\n'\  
 f'\* Разрушение или повреждение оборудования (физическое или программное).\n'\  
 f'\* Ошибки администрирования, приводящие к недоступности сервисов.\n\n'\  
 f'4. Комплексные/Распространенные:\n'\  
 f'\* Использование уязвимостей ПО и аппаратного обеспечения.\n'\  
 f'\* Действия инсайдеров (злоупотребление полномочиями, хищение данных, саботаж).\n'\  
 f'\* Компрометация учетных записей (кража паролей, фишинг).\n'\  
 f'\* Атаки с использованием методов социальной инженерии.\n'\  
 f'\* Угрозы безопасности при использовании облачных технологий и мобильных устройств.\n'\  
 f'\* Угрозы, связанные с импортным ПО и оборудованием (возможность необъявленных функций, закладок).', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_ugrozy\_9(message: Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id,f'Ключевые документы ФСТЭК для изучения классификации угроз:\n\n'\  
 f'1. Методические документы: Самый важный источник.\n'\  
 f'\* "Методика определения актуальных угроз безопасности информации в информационных системах" (Актуальная версия утверждается ФСТЭК, ищите в разделе "Документы" -> "Методические документы"). Это основной документ, описывающий сам процесс классификации и типизации угроз.\n'\  
 f'\* Другие методические рекомендации по оценке рисков и защите.\n'\  
 f'2. Требования по защите информации: В них перечисляются угрозы, от которых \*обязаны\* защищаться определенные типы систем.\n'\  
 f'\* Приказы ФСТЭК по защите КИИ (Критическая информационная инфраструктура): Особенно Приказ ФСТЭК России от 25.12.2017 № 239 (требования к созданию систем обнаружения, предупреждения и ликвидации последствий компьютерных атак) и Приказ ФСТЭК России от 21.12.2017 № 235 (требования по обеспечению безопасности значимых объектов КИИ). В приложениях часто есть перечни актуальных угроз.\n'\  
 f'\* Приказы по защите ГИС (Государственные информационные системы): Например, базовые требования.\n'\  
 f'\* Приказы по защите АСУ ТП (Автоматизированные системы управления технологическими процессами): Например, Приказ ФСТЭК России от 14.03.2014 № 31.\n'\  
 f'\* Приказ ФСТЭК России от 18.02.2013 № 21 ("Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных") - содержит классификацию угроз для систем ПДн.\n'\  
 f'3. Аналитика и новости: В разделах "Новости", "Публикации", "Доклады и выступления" ФСТЭК регулярно освещает актуальные угрозы, особенно инсайдерские, целевые (APT) и угрозы КИИ.', reply\_markup=get\_more\_ugrozy\_back())  
  
async def get\_zachita(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'ФСТЭК требует комплексного подхода, сочетающего организационные и '\  
 f'технические меры, адаптированные под тип системы (ГИС, КИИ, АСУ ТП, ИСПДн) и категорию информации (гос. '\  
 f'тайна, КТ, ПДн). Защита основывается на модели угроз и присвоенном классе защищенности.',  
 reply\_markup=get\_more\_zach())  
  
async def get\_zachita\_1(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Данные меры всегда носят комплексный характер и подбираются исходя из:\n\n'\  
 f'1. Категории защищаемой информации (государственная тайна, коммерческая тайна, персональные данные, информация КИИ).\n'\  
 f'2. Типа информационной системы (ГИС - государственная ИС, ИСПДн - ИС персональных данных, АСУ ТП - автоматизированные системы управления технологическими процессами, объект КИИ - критическая информационная инфраструктура).\n'\  
 f'3. Модели угроз и уровня исходной защищенности системы.\n'\  
 f'4. Класса защищенности, присваиваемого системе по результатам оценки.',reply\_markup=get\_more\_zach\_back())  
  
async def get\_zachita\_2(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Организационные (организационно-правовые) меры защиты:'\  
 f'Эти меры лежат в основе всей системы ЗИ и регламентируют процессы. Документы ФСТЭК всегда начинаются с них.\n\n'\  
 f'1. Разработка и внедрение организационно-распорядительной документации:\n'\  
 f'\* Политика информационной безопасности.\n'\  
 f'\* Инструкции пользователей и администраторов по ИБ.\n'\  
 f'\* Регламенты обработки информации, особенно конфиденциальной и ПДн.\n'\  
 f'\* Положения о коммерческой тайне, о разграничении доступа.\n'\  
 f'\* Планы мероприятий по обеспечению ИБ, реагированию на инциденты.\n\n'\  
 f'2. Управление персоналом:\n'\  
 f'\* Подбор и проверка сотрудников (особенно с доступом к секретам или КИИ).\n'\  
 f'\* Обучение и аттестация по вопросам ИБ.\n'\  
 f'\* Подписание обязательств о неразглашении (NDA).\n'\  
 f'\* Четкое определение и разграничение ролей, обязанностей и прав доступа (Принцип минимальных привилегий).\n\n'\  
 f'3. Режимные меры:\n'\  
 f'\* Организация пропускного режима на территорию и в помещения.\n'\  
 f'\* Разграничение зон доступа (чистые/грязные зоны для носителей).\n'\  
 f'\* Контроль вноса/выноса оборудования и носителей информации.\n\n'\  
 f'4. Планирование и управление безопасностью:\n'\  
 f'\* Оценка рисков ИБ.\n'\  
 f'\* Аудит и контроль эффективности мер защиты.\n'\  
 f'\* Управление инцидентами ИБ.\n'\  
 f'\* Резервирование и восстановление (планы аварийного восстановления - DRP).',reply\_markup=get\_more\_zach\_back())  
  
async def get\_zachita\_3(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Это меры, реализуемые с помощью специальных средств защиты информации (СЗИ). ФСТЭК строго регламентирует их применение и сертификацию, особенно для гостайны и КИИ.\n\n'\  
 f'1. Идентификация и аутентификация:\n'\  
 f'\* Строгая парольная политика.\n'\  
 f'\* Использование токенов, электронных ключей (eToken, ruToken), смарт-карт.\n'\  
 f'\* Биометрическая аутентификация (где это обосновано требованиями).\n\n'\  
 f'2. Управление доступом:\n'\  
 f'\* Дискреционный (DAC), мандатный (MAC) и ролевой (RBAC) доступ.\n'\  
 f'\* Реализация политик разграничения доступа к ресурсам системы.\n\n'\  
 f'3. Регистрация и аудит:\n'\  
 f'\* Ведение журналов безопасности (событий) систем и СЗИ.\n'\  
 f'\* Централизованный сбор и анализ журналов (SIEM-системы).\n'\  
 f'\* Контроль действий пользователей и администраторов.\n\n'\  
 f'4. Защита от вредоносного ПО:\n'\  
 f'\* Антивирусная защита (включая средства на шлюзах, серверах, рабочих станциях).\n'\  
 f'\* Системы предотвращения вторжений (IPS) / обнаружения вторжений (IDS).\n'\  
 f'\* Контроль целостности ПО и конфигураций.\n\n'\  
 f'5. Защита информации в сетях:\n'\  
 f'\* Межсетевые экраны (Firewalls), включая NGFW.\n'\  
 f'\* Виртуальные частные сети (VPN) с криптозащитой.\n'\  
 f'\* Сегментирование сетей (VLAN).\n'\  
 f'\* Системы обнаружения/предотвращения атак (IDS/IPS).\n'\  
 f'\* Защита от DDoS-атак.\n\n'\  
 f'6. Криптографическая защита информации (КЗИ):\n'\  
 f'\* Шифрование данных на носителях (СОД).\n'\  
 f'\* Шифрование каналов связи (VPN с ГОСТ-алгоритмами).\n'\  
 f'\* Электронная подпись (ЭП).\n\n'\  
 f'\* Средства криптографической защиты информации (СКЗИ), сертифицированные ФСТЭК России (обязательно для защиты гостайны и часто для КИИ/ПДн высоких уровней).\n\n'\  
 f'7. Контроль безопасности ПО:\n'\  
 f'\* Анализ исходного кода на наличие уязвимостей и недекларированных возможностей (НДВ) - критично для импортного ПО и КИИ.\n'\  
 f'\* Применение только лицензионного ПО.\n'\  
 f'\* Своевременное обновление ПО (устранение уязвимостей).\n\n'\  
 f'8. Защита от утечки по техническим каналам (ТКУИ):\n'\  
 f'\* Подавление побочных электромагнитных излучений (ПЭМИН) - экранирование помещений, использование сертифицированной техники.\n'\  
 f'\* Защита от утечки по акустическим и виброакустическим каналам.\n'\  
 f'\* Использование аппаратных средств защиты (генераторы шума, фильтры).\n\n'\  
 f'9. Физическая защита:\n'\  
 f'\* Охранная и пожарная сигнализация.\n'\  
 f'\* Системы видеонаблюдения.\n'\  
 f'\* Защита от несанкционированного доступа к серверам и коммуникационному оборудованию (замки в стойках, выделенные охраняемые помещения - ЦОДы).\n'\  
 f'\* Системы контроля и управления доступом (СКУД).',reply\_markup=get\_more\_zach\_back())  
  
async def get\_zachita\_4(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Основные принципы и подходы, подчеркиваемые ФСТЭК:\n\n'\  
 f'1. Принцип "глубокой эшелонированной обороны" (Defense-in-Depth): Защита строится на нескольких независимых уровнях (периметр, сеть, хост, приложение, данные), так чтобы прорыв одного уровня не означал компрометацию всей системы.\n'\  
 f'2. Комплексность: Ни одна техническая мера не эффективна без грамотных организационных мер и наоборот.\n'\  
 f'3. Адекватность угрозам: Выбор мер защиты определяется моделью угроз и уровнем возможного ущерба. Защита от APT-групп требует гораздо более сложных и дорогих мер, чем от массовых вирусов.\n'\  
 f'4. Непрерывность: Обеспечение ИБ - это постоянный процесс (мониторинг, обновление, аудит, реагирование на инциденты), а не разовая акция.\n'\  
 f'5. Использование сертифицированных СЗИ: Для защиты информации ограниченного доступа (гостайна) и часто для КИИ обязательно применение СЗИ, прошедших процедуру обязательной сертификации ФСТЭК России и включенных в соответствующий реестр. Для ПДн и коммерческой тайны может применяться добровольная сертификация или декларирование соответствия.\n'\  
 f'6. Импортозамещение: ФСТЭК активно продвигает использование отечественных СЗИ, особенно в государственном секторе и на объектах КИИ, для обеспечения технологической независимости и безопасности.',reply\_markup=get\_more\_zach\_back())  
  
async def get\_zachita\_5(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Где искать конкретные требования к способам защиты на сайте ФСТЭК (fstec.ru):\n\n'\  
 f'1. Раздел "Документы" -> "Требования по защите информации": Ключевой раздел! Здесь находятся приказы, регламентирующие ЗИ для разных типов систем:\n'\  
 f'\* Защита государственной тайны: Совокупность требований, изложенных в многочисленных приказах, инструкциях и руководящих документах (РД). Поиск по ключевым словам "государственная тайна", "совершенно секретно" и т.д.\n'\  
 f'\* Защита КИИ (Критическая информационная инфраструктура):\n'\  
 f' \*\* Приказ ФСТЭК России от 25.12.2017 № 239 "Требования по обеспечению безопасности значимых объектов КИИ...".\n'\  
 f' \*\* Приказ ФСТЭК России от 21.12.2017 № 235 "Требования к созданию систем обнаружения, предупреждения и ликвидации последствий компьютерных атак...".\n'\  
 f'\* Защита персональных данных (ИСПДн):\n'\  
 f' \*\* Приказ ФСТЭК России от 18.02.2013 № 21 "Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных".\n'\  
 f'\* Защита АСУ ТП (Автоматизированные системы управления технологическими процессами):\n'\  
 f' \*\* Приказ ФСТЭК России от 14.03.2014 № 31 "Об утверждении Требований к обеспечению защиты информации в АСУ ТП".\n'\  
 f'\* Защита ГИС (Государственные информационные системы): Базовые требования и отраслевые документы.\n\n'\  
 f'2. Раздел "Документы" -> "Методические документы": Содержат руководства по реализации требований:\n'\  
 f'\* Методики оценки соответствия.\n'\  
 f'\* Методики определения угроз.\n'\  
 f'\* Рекомендации по применению СЗИ.\n\n'\  
 f'3. Раздел "Деятельность" / "Государственные реестры":\n'\  
 f' \*\* Реестр сертифицированных средств защиты информации: Перечень СЗИ, прошедших обязательную сертификацию ФСТЭК (для гостайны и т.д.).\n'\  
 f' \*\* Реестр российских программ для электронных вычислительных машин и баз данных: Важен для импортозамещения.\n\n'\  
 f'4. Раздел "Новости" / "Публикации" / "Доклады и выступления": Здесь публикуются разъяснения, актуальные тренды, предупреждения о новых угрозах и рекомендации по защите.',reply\_markup=get\_more\_zach\_back())  
  
async def get\_mai(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Московский авиационный институт (МАИ) относится к объектам '\  
 f'критической информационной инфраструктуры (КИИ), т.к. его деятельность связана с одной из сфер, указанных '\  
 f'в Федеральном законе № 187-ФЗ. Согласно законодательству, к КИИ относятся информационные системы, '\  
 f'информационно-телекоммуникационные сети и автоматизированные системы управления, функционирующие в следующих областях:\n'\  
 f'\* Оборонная промышленность (включая ракетно-космическую отрасль).\n'\  
 f'\* Энергетика, топливно-энергетический комплекс (ТЭК).\n'\  
 f'\* Транспорт (включая авиационную отрасль).\n'\  
 f'\* Связь и информационные технологии (включая обеспечение взаимодействия систем КИИ).',  
 reply\_markup=get\_more())  
  
async def get\_razr(message:Message, bot: Bot, request: Request):  
 await request.add\_data(message.from\_user.id, message.from\_user.first\_name)  
 await bot.send\_message(message.from\_user.id, f'Данный раздел пока находится в разработке\n'\  
 'Для получения более точной информации просьба написать разработчику: https://t.me/Maestro\_Alexandro',  
 reply\_markup=get\_more\_back())

# ПРИЛОЖЕНИЕ Ж

**Код из файла commands.py**

from aiogram import Bot  
from aiogram.types import BotCommand, BotCommandScopeDefault  
  
  
async def set\_commands(bot: Bot):  
 commands = [  
 BotCommand(  
 command = 'start',  
 description = 'Начало работы'  
 ),  
 BotCommand(  
 command='help',  
 description='Помощь'  
 )  
 ]  
  
  
 await bot.set\_my\_commands(commands, BotCommandScopeDefault())

# ПРИЛОЖЕНИЕ З

**Код из файла dbconnect.py**

import asyncpg  
import datetime  
  
  
class Request:  
 def \_\_init\_\_(self, connector: asyncpg.pool.Pool):  
 self.connector = connector  
  
 async def add\_data(self, user\_id, user\_name):  
 query = f"INSERT INTO user\_data (user\_id, user\_name, user\_role) VALUES ({user\_id}, '{user\_name}', 'user')" \  
 f"ON CONFLICT (user\_id) DO UPDATE SET user\_name='{user\_name}'"  
 await self.connector.execute(query)

# ПРИЛОЖЕНИЕ И

**Код из файла inline.py**

from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton  
from aiogram.utils.keyboard import InlineKeyboardBuilder  
  
  
def get\_inline\_keyboard():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Злоумышленники', callback\_data='1/1')  
 keyboard\_builder.button(text='Виды угроз', callback\_data='1/2')  
 keyboard\_builder.button(text='Способы защиты', callback\_data='1/3')  
 keyboard\_builder.button(text='Другой вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url = 'https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_zlou():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='По местоположению', callback\_data='3/1')  
 keyboard\_builder.button(text='По уровню ресурсов и возможностей', callback\_data='3/2')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(1)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_zlou\_back():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Назад', callback\_data='1/1')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_ugrozy():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='По природе возникновения', callback\_data='4/1')  
 keyboard\_builder.button(text='По источнику угрозы', callback\_data='4/2')  
 keyboard\_builder.button(text='По способу воздействия', callback\_data='4/3')  
 keyboard\_builder.button(text='По расположению источника', callback\_data='4/4')  
 keyboard\_builder.button(text='По используемым средствам', callback\_data='4/5')  
 keyboard\_builder.button(text='По степени воздействия на объект защиты', callback\_data='4/6')  
 keyboard\_builder.button(text='По уровню квалификации и ресурсам источника', callback\_data='4/7')  
 keyboard\_builder.button(text='Конкретные примеры угроз', callback\_data='4/8')  
 keyboard\_builder.button(text='Ключевые документы ФСТЭК по вопросу', callback\_data='4/9')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(1)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_ugrozy\_back():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Назад', callback\_data='1/2')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
  
def get\_more\_zach():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='По каким параметрам подбираются', callback\_data='5/1')  
 keyboard\_builder.button(text='Организационные (организационно-правовые) меры защиты', callback\_data='5/2')  
 keyboard\_builder.button(text='Технические (программно-аппаратные) меры защиты', callback\_data='5/3')  
 keyboard\_builder.button(text='Основные принципы и подходы, подчеркиваемые ФСТЭК', callback\_data='5/4')  
 keyboard\_builder.button(text='Конкретные требования', callback\_data='5/5')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(1)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_zach\_back():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Назад', callback\_data='1/3')  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Хочу задать более конкретный вопрос', callback\_data='3/3')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more\_back():  
 keyboard\_builder = InlineKeyboardBuilder()  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(1)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')

# ПРИЛОЖЕНИЕ К

**Код из файла reply.py**

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton, KeyboardButtonPollType  
from aiogram.utils.keyboard import ReplyKeyboardBuilder  
  
  
  
def get\_start\_menu():  
 keyboard\_builder = ReplyKeyboardBuilder()  
 keyboard\_builder.button(text='Злоумышленники', callback\_data='1/1')  
 keyboard\_builder.button(text='Виды угроз', callback\_data='1/2')  
 keyboard\_builder.button(text='Способы защиты', callback\_data='1/3')  
 keyboard\_builder.button(text='Другой вопрос', callback\_data='1/4')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')  
  
def get\_more():  
 keyboard\_builder = ReplyKeyboardBuilder()  
 keyboard\_builder.button(text='Вернутся в главное меню', callback\_data='2/1')  
 keyboard\_builder.button(text='Помощь', url='https://t.me/Maestro\_Alexandro')  
  
 keyboard\_builder.adjust(2)  
 return keyboard\_builder.as\_markup(resize\_keyboard=True, one\_time\_keyboard=True, input\_field\_placeholder='Выберите действие⬇')

# ПРИЛОЖЕНИЕ Л

**Код из файла main.py**

from aiogram import Bot, Dispatcher  
import asyncio  
import logging  
from core.settings import settings  
from aiogram.filters import Command, CommandStart  
from aiogram import F, types  
from core.utils.commands import set\_commands  
import asyncpg  
from core.middlewares.dbmiddleware import DbSession  
from datetime import datetime, timedelta  
from core.handlers.basic import get\_start, get\_zlou, get\_ugrozy, get\_menu, get\_zachita, get\_mai, get\_zlou\_odin, get\_zlou\_dva  
from core.handlers.basic import get\_ugrozy\_1, get\_ugrozy\_2, get\_ugrozy\_3, get\_ugrozy\_4, get\_ugrozy\_5, get\_ugrozy\_6, get\_ugrozy\_7  
from core.handlers.basic import get\_ugrozy\_8, get\_ugrozy\_9, get\_zachita\_1, get\_zachita\_2, get\_zachita\_3, get\_zachita\_4, get\_zachita\_5  
from core.handlers.basic import get\_razr  
from aiogram.types.callback\_query import CallbackQuery  
from aiogram.filters.callback\_data import CallbackData  
async def start\_bot(bot: Bot):  
 await set\_commands(bot)  
 await bot.send\_message(settings.bots.admin\_id, text='Бот запущен!')  
  
async def stop\_bot(bot: Bot):  
 await bot.send\_message(settings.bots.admin\_id, text='Бот остановлен!')  
  
  
async def creat\_pool():  
 return await asyncpg.create\_pool(user="postgres", password='postgresalex', database='infobes',  
 host='127.0.0.1', port=5432, command\_timeout=60)  
  
async def start():  
 logging.basicConfig(level=logging.INFO,  
 format="%(asctime)s - [%(levelname)s] - %(name)s - "  
 "(%(filename)s).%(funcName)s(%(lineno)d) - %(message)s"  
  
 )  
 bot = Bot(token=settings.bots.bot\_token)  
 pool\_connect = await creat\_pool()  
 dp = Dispatcher()  
 dp.update.middleware.register(DbSession(pool\_connect))  
 dp.startup.register(start\_bot)  
 dp.shutdown.register(stop\_bot)  
 dp.message.register(get\_start, CommandStart())  
 dp.message.register(get\_start, F.text == "Привет!")  
 dp.message.register(get\_start, F.text == "Вернуться в главное меню")  
 dp.message.register(get\_zlou, F.text == "Злоумышленники")  
 dp.callback\_query.register(get\_zlou, lambda c: c.data == '1/1')  
 dp.callback\_query.register(get\_ugrozy, lambda c: c.data == '1/2')  
 dp.callback\_query.register(get\_zachita, lambda c: c.data == '1/3')  
 dp.callback\_query.register(get\_mai, lambda c: c.data == '1/4')  
 dp.message.register(get\_mai, F.text == "МАИ является КИИ?")  
 dp.message.register(get\_mai, F.text.contains("МАИ"))  
 dp.callback\_query.register(get\_menu, lambda c: c.data == '2/1')  
 dp.callback\_query.register(get\_mai, lambda c: c.data == '2/2')  
 dp.callback\_query.register(get\_zlou\_odin, lambda c: c.data == '3/1')  
 dp.callback\_query.register(get\_zlou\_dva, lambda c: c.data == '3/2')  
 dp.callback\_query.register(get\_ugrozy\_1, lambda c: c.data == '4/1')  
 dp.callback\_query.register(get\_ugrozy\_2, lambda c: c.data == '4/2')  
 dp.callback\_query.register(get\_ugrozy\_3, lambda c: c.data == '4/3')  
 dp.callback\_query.register(get\_ugrozy\_4, lambda c: c.data == '4/4')  
 dp.callback\_query.register(get\_ugrozy\_5, lambda c: c.data == '4/5')  
 dp.callback\_query.register(get\_ugrozy\_6, lambda c: c.data == '4/6')  
 dp.callback\_query.register(get\_ugrozy\_7, lambda c: c.data == '4/7')  
 dp.callback\_query.register(get\_ugrozy\_8, lambda c: c.data == '4/8')  
 dp.callback\_query.register(get\_ugrozy\_9, lambda c: c.data == '4/9')  
 dp.callback\_query.register(get\_zachita\_1, lambda c: c.data == '5/1')  
 dp.callback\_query.register(get\_zachita\_2, lambda c: c.data == '5/2')  
 dp.callback\_query.register(get\_zachita\_3, lambda c: c.data == '5/3')  
 dp.callback\_query.register(get\_zachita\_4, lambda c: c.data == '5/4')  
 dp.callback\_query.register(get\_zachita\_5, lambda c: c.data == '5/5')  
  
 dp.callback\_query.register(get\_razr, lambda c: c.data == '3/3')  
  
 try:  
 await dp.start\_polling(bot, allowed\_updates=['message', 'callback\_query'])  
 finally:  
 await bot.session.close()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 asyncio.run(start())