CTI AgenticRAG Project Overview

"Exploring the significance and implementation of an agenticRAG system to aid in Cyber Threat Intelligence"

Dhruv Majumdar

Who Am I ?

Official - Deloitte MXDR leader

- I am not a Programmer
- I am not a lot of things .....
- But I am a tinkerer and I love Technologies

Challenge

The pain of making RAG work for cybersecurity

The Pains of making GenAI work for Cyber Security:

Agenda

Key Discussion Points and Presentation Objectives

01 Project Overview

Overview of CTI and its importance in cybersecurity.
Brief introduction to Retrieval-Augmented Generation (RAG).

02 Key Features of the System

Overview of CTI - AgenticRAG

03 Agent Capabilities and Modes

Toggle between reasoning mode (using deepseek-r1:14b-64k-ab) and direct mode.
Explanation of the reasoning process in detail.

04 Document Processing and Vector Storage

Document processing for PDFs, websites, etc., with vector storage support.
Dual database system using permanent ChromaDB and temporary vector stores.

05 Database Management

Configurable database prioritization between permanent and temporary sources.
How to manage data priorities effectively.

06 Web Search Integration

Real-time information integration through SerpAPI/DuckDuckGo.
Benefits of real-time search capabilities in CTI systems.

07 Containerized Deployment

MSP & CUDA Support
Oh its fully Local leveraging Ollama

08 Reasoning with self reflection

No, I am not another DeepResearch tool and I don't claim to do DeepResearch. I do Inferring with self-reflection

09 Demo: Hands-On Experience with the System

Live demonstration showcasing key features and functionalities.
An interactive session where participants can ask questions.

Overview of AgenticRAG for Cyber Threat Intelligence

Cyber Threat Intelligence (CTI) CTI systems provide insights into cyber threats, helping organizations mitigate risks and enhance security.	**Retrieval-Augmented Generation** RAG combines retrieval and generation techniques to enhance information accuracy and relevance in threat detection.	**AgenticRAG** Implements reasoning and direct approaches for query processing	**Real-time** Enhances relevant responses with real-time web information
Data Enrichment The system allows the user to leverage the permanent RAG along with the temporary RAG. The system also processes documents from PDFs and websites with vector storage.	**Modular Response Type** The system works with a combination of Audience level and Domain Roles	**User-friendly Interface** Designed with a user-friendly interface for easy navigation access to individual functions to tweak your search and response.	**Customizable** Supports multiple LLMs, including llama3:8b-64k for general RAG tasks and qwen2.5:7b-64k for IOC extraction. Users can toggle between reasoning mode, utilizing deepseek-r1:14b-64k-ab, and direct mode. We employ a dual database architecture featuring permanent ChromaDB and temporary vector stores.

Web UI and Model Support Overview

- Web UI with Streamlit - *The Streamlit framework provides a user-friendly interface for interacting with the RAG system, enhancing usability for analysts.*
- Support for Multiple LLM Models - *The system supports various Large Language Models (LLM) to cater to different functionalities, ensuring versatility in operations.*
- llama3:8b-64k - *Utilizing the llama3:8b-64k model for general RAG tasks ensures comprehensive information retrieval and analysis capabilities.*
- qwen2.5:7b-64k - *The qwen2.5:7b-64k model specializes in Indicator of Compromise (IOC) extraction, improving threat detection efficacy.*

Modes of Operation & Document Processing

Toggle between modes

Utilize the option to switch between reasoning mode and direct mode for optimal processing of intelligence data.

Inferencing/ Reasoning

Integrate deepseek-r1:14b-64k-ab for enhanced reasoning capabilities in RAG retrieval supported by self-reflection.

Document Processing

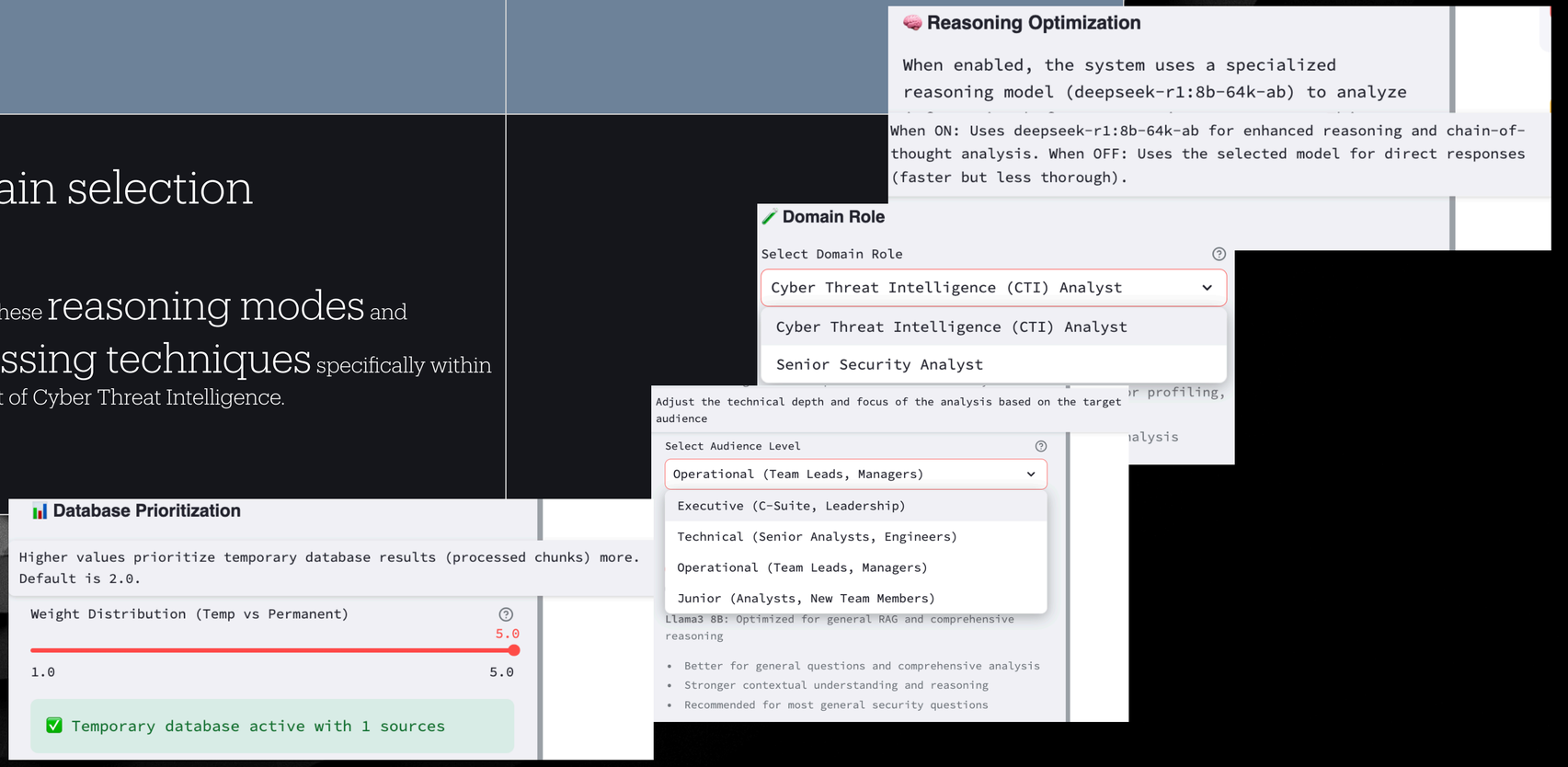
Implement comprehensive document embedding techniques for PDFs and websites to extract relevant information along with real-time web search.

Vector Storage

- Utilize vector storage to ensure efficient retrieval and management of processed documents and data retrieval.

Domain selection

Apply these reasoning modes and processing techniques specifically within the context of Cyber Threat Intelligence.



Key Features

Docker Support for Deployment

Utilizes Docker for containerized deployment, ensuring flexibility and scalability in managing applications.

Ollama Integration

Seamlessly integrates with Ollama to enhance deployment capabilities and streamline workflows for users.

Domain-Specific Prompting

Employs tailored prompting techniques for effective cyber security analysis, improving response accuracy.

User Document Uploads

Enables users to upload various documents for processing, enhancing data accessibility and usability in analysis.

Website Processing

Facilitates the processing of websites to extract relevant security information, aiding in threat identification.

Security-Related Queries

Allows users to ask security-related questions, providing them with insights and guidance on potential threats.

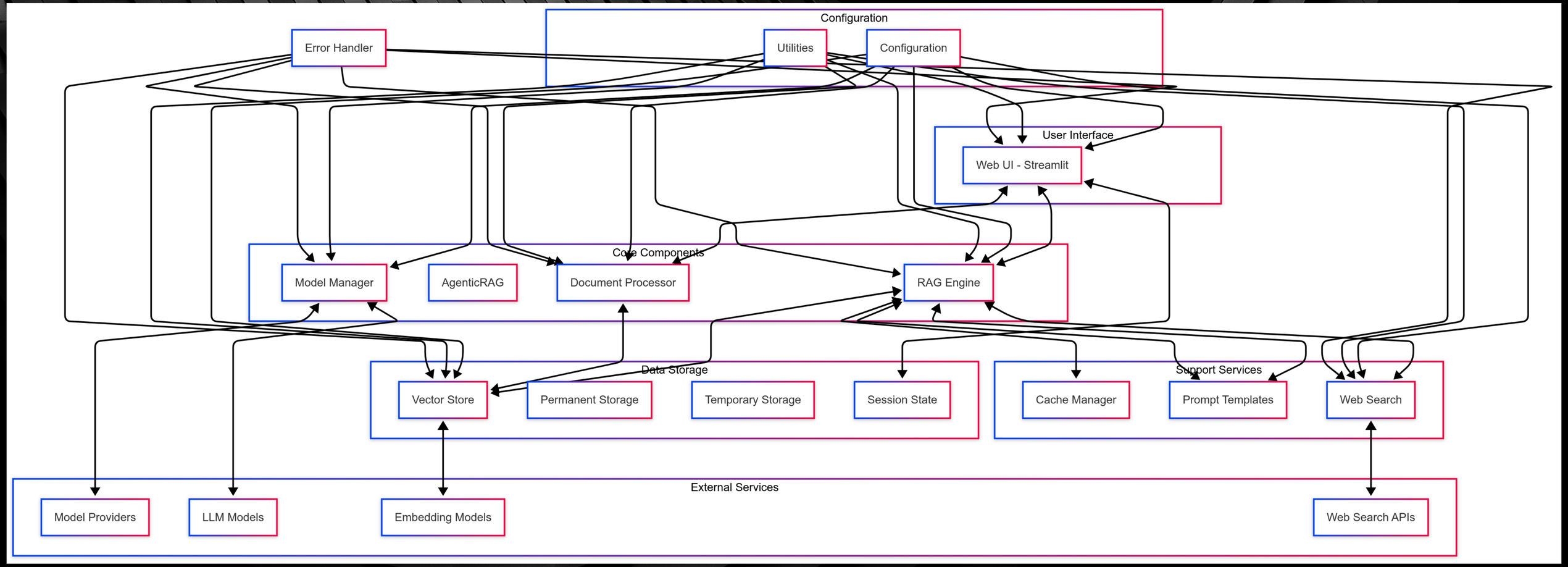
Configurable Settings

Users can customize model selection, reasoning approaches, and database prioritization to suit specific needs.

Enhanced Reasoning Approaches

Offers various reasoning approaches to improve analysis depth and accuracy in cyber security evaluations.

Architecture Diagram



DEMO & QA

Well this is not a hypothesis so lets dive into the working project

Please do ask questions to make the presentation interactive also since using reasoning and running locally on a OK'ish system the final prompt might / will take time, so please bare with me.

What my AI things people might ask:

- *Yes its all about context length*
- *Yes I am using Ollama "OLLAMA_FLASH_ATTENTION"*
- *Yes the Application is multithreaded and can crunch through the following directories pretty fast :*

  data/

```
├── pdf_files/
│   ├── document1.pdf
│   └── document2.pdf
├── text_files/
│   ├── document3.txt
│   └── document4.txt
└── json_files/
    ├── document5.json
    └── document6.json
```

- *Permanent VectorDB can be anything you want to build on, I choose CTI oh wait but did I not show you RedTeamer .*