Cybersecurity Digital Consultant

By

Wong Zen Kai

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMMUNICATIONS AND NETWORKING
Faculty of Information and Communication Technology
(Kampar Campus)

FEBRUARY 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Aun Yi Chiet who has given me this bright opportunity to engage in an interesting AI project. It is my first step to establish a career in Artificial Intelligence field. A million thanks to you.

At the same time, I would like to thank the Faculty of Information Communication Technology for sponsoring my research.

ABSTRACT

Cybersecurity Digital Consultant (CDS) is GPT-based digital assistant designed to generate and design solutions for networking and cybersecurity in diverse IT environments. As network threats evolve and the complexity of cybersecurity paradigms, platforms, tools, and stacks increases, ensuring robust protection for on-premises infrastructure becomes a formidable challenge. CDS is a multi-agent LLM designed to automate the workflow of deploying cybersecurity solutions. CDS define agents involved throughout the deployment pipeline, such as security analyst, cybersecurity consultant and penetration tester, each with a specialized function to deal with different problems. Each individual agent possesses distinct traits, personalities, responsibilities, and roles. A dedicated storage system will be established for each agent to archive their historical data, aiding in future output generation. Inter-agent communication enables a cohesive response to threats by collaboratively generating comprehensive solutions and network designs tailored to the user's network environment. CDS provides a user-friendly, context-aware approach to protecting networks against an evolving threat landscape.

Area of Study – Artificial Intelligence.

Keywords - Multi-Agent Model, XML Diagram Generation, Network Creation, Cybersecurity, Generative Artificial Intelligence.

Index Terms – Large Language Models, Generative Artificial Intelligence, Network Intelligence.

TABLE OF CONTENTS

APPENDIX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
LLM	Large Language Model
MoE	Mixture of Experts

CHAPTER 1

Introduction

Generative Artificial Intelligence represents a transformative breakthrough in the realm of intelligent technologies. With their advanced reasoning, generalization, and emergent intelligence capabilities, Large Language Models that boast billions of parameters are making significant strides in both commercial and technical domains. These models are adept at a wide range of applications, from transforming text to text, creating images from descriptions, to generating code from prompts.

Domain-specific adaptations of LLMs have shown remarkable effectiveness in specialized fields such as robotic intelligence, semiconductor design, and protein structure generation. These sophisticated LLMs excel at distilling intricate data into feature vectors and capturing vast amounts of knowledge in tokenized forms. Their abilities extend beyond mere support to potentially replacing human efforts in understanding complex concepts, deductive reasoning, and decision-making processes. As a result, these advancements facilitate the execution of network-related tasks through intuitive natural language interactions with intelligent systems.

1.1 Problem Statement and Motivation

In today's digital age, hacking and security breaches in network systems have become alarmingly common. The cost of hiring or consulting a security analyst, or outsourcing to a network security company, can be prohibitively expensive for many organizations. Training an individual with no networking knowledge to become an expert can be very expensive and time-consuming. Therefore, we would like to explore the use of customized LLMs to create and configure networks based on the user's needs. These advanced LLMs and Generative AI technologies are pivotal in addressing complex security issues. They enhance cybersecurity by planning network creation, improving decision-making processes, and augmenting human expertise. However, it is crucial to manage their deployment carefully to mitigate ethical concerns and biases, ensuring they are used responsibly and effectively.

1.2 Research Objectives

The aim of this thesis is to propose a solution that leverages multiple domain-specific Large Language Models to create a 'Cybersecurity Digital Consultant'. There are five main objectives for this project:

1. **Custom Sets of Networking GPT Agents**: Each agent will have specific expertise, configurations, and traits. They will handle particular tasks and provide the best solutions based on user requirements.
2. **Persistent Memory for GPT Agents**: This will allow the agents to refer back to previous conversations and use their past experiences to improve their performance on future tasks.
3. **Multi-agent LLM Framework for cybersecurity architecture and solutions**: This capability enables agents to communicate and collaboratively generate solutions tailored to the user's requirements, ensuring practical implementation in real-world scenarios.
4. **Deploy and configure agents using Agno**: Agno empowers users to design their own set of agents and workflows, enabling the generation of outputs that closely align with their expectations.
5. **Visualize the networking diagram using XML format file**: Draw.io supports the diagram creation from XML file. This allows the agents to create the network diagram in XML format which the users are able to visualize the network topology in a detailed format.

1.3 Project Scope and Direction

The architecture comprises a multifaceted framework, including a manager leading the group and other specified agents such as a penetration tester, cybersecurity consultant, and security analyst. Each agent will be equipped with different skills and personality traits. After the user prompts a task, the agents will communicate with each other to produce a consolidated output for the user. Therefore, the tasks for this project are listed below:

1. **Defining the Roles and Personality Traits of the Agents**: We will determine the roles, skills, and responsibilities of each agent, such as penetration tester

and security analyst. Additionally, we will define their personality traits, such as passive or aggressive, to aid in decision-making.

2. **Visualize the network topology created by the agent**: We will investigate the effects of equipping an agent with persistent storage. This involves creating an agent with persistent memory and attempting to train it.
3. **Synthesis and Communication Between Agents**: After specialized agents process user queries, they will synthesize their outputs. The Group manager will consolidate information from different agents, ensuring a coherent and comprehensive response. The Group manager then communicates the synthesized solution back to the user. This synthesis step ensures that users receive a unified and well-organized answer, even if multiple agents were involved.

1.4 Contributions

This project introduces a groundbreaking Cybersecurity Digital Consultant, the first of its kind to harness a multi-agent large language model (LLM) system using GROK and Agno. Each agent is assigned a domain-specific role such as network topology generation, cost analysis, and vulnerability detection, while a central coordinating agent manages their interactions through Agno's coordinate mode. This modular architecture goes beyond the capabilities of traditional single-agent systems or generic chatbots, enabling real-world consulting tasks including XML-based network topology design, budgeting, and security assessment through natural language input.

The system provides a persistent-memory-ready, modular, and customizable LLM framework tailored to the needs of students, SMEs, and network engineers alike. It empowers users to build and manage secure networks through intuitive interaction without requiring deep technical expertise.

1.5 Report Organization

The details of this research and development are presented in the following chapters. In Chapter 2, we review backgrounds related to the multi-agent framework, generative AI models, and persistent memory. Next, in Chapter 3, we present a preliminary study on applying the multi-agent framework in the cybersecurity field. Chapter 4 covers the system design. Chapter 5 covers the system implementation. In Chapter 6, system evaluation will be shown. Finally, in Chapter 7, we summarize our findings and provide a better understanding of the multi-agent framework in the context of cybersecurity.

CHAPTER 2

Literature Reviews

2.1 Comparison between Multi-agent frameworks and Single-agent frameworks

The primary distinction between single-agent frameworks and multi-agent frameworks lies in their message passing and task distribution mechanisms. According to SuperAnnotate [1], multi-agent frameworks excel in handling complex tasks. Single-agent LLMs, while occasionally generating plausible but incorrect information, suffer from reduced response accuracy. In contrast, multi-agent LLMs distribute responses for cross-checking among different agents. Each agent, with its unique configuration, identifies and rectifies inaccuracies. Additionally, single agents are constrained by limited context windows, often focusing on specific query parts rather than the entire context. By adopting multi-agent frameworks, user queries can be divided into specific tasks, allowing each agent to handle its designated portion. Most notably, multi-agent systems collaborate, leveraging diverse skills and viewpoints to tackle intricate problems.

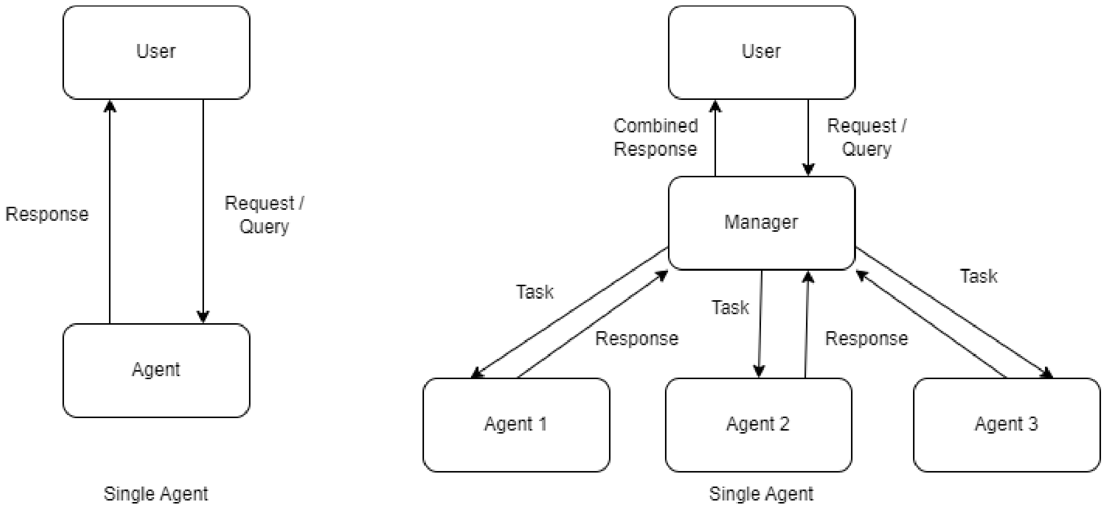


Figure 2.1 shows the difference between single-agent and multi-agent framework.

2.2 Comparison between LLMs Model

Based on [3], the Chatbot Arena LLM Leaderboard from February 2025 ranks several leading language models according to their performance. As shown in the table below, Grok-3 achieved a score of 1403, followed by Gemini 2.0 with 1385, ChatGPT-4o with 1377, and DeepSeek R1 with 1362.

The ranking methodology is based on the number of models that are statistically better than a given model, using a 95% confidence interval. Specifically, Model A is considered statistically superior to Model B if A's lower bound score is higher than B's upper bound score. Additionally, rankings incorporate style controls, factors such as response length, markdown usage, and formatting consistency to better isolate true model performance. Based on this methodology, Grok-3 is currently leading among the evaluated models.

From [2] and [4], it's evident that numerous companies have released various LLMs in recent years. Prominent examples include Grok, ChatGPT, Gemini, DeepSeek, and Claude, each with multiple versions provided by different vendors. Among these, Grok-3 stands out as a significant advancement. Running on 200,000 NVIDIA GPUs, Grok-3 excels in logical reasoning, advanced problem-solving, and real-time research.

In standardized benchmarks, Grok-3 has demonstrated exceptional capabilities. For example, in the 2024 AIME Math Competition, it scored 52, outperforming Gemini 2 Pro (32) and ChatGPT (9). In graduate-level expert reasoning (GPQA), Grok-3 achieved a score of 75, again ranking above most competing models. Additionally, Grok-3 has shown strength in code generation, making it a versatile tool for technical tasks.

Despite Grok-3's impressive performance, ChatGPT remains the most widely used AI model, appreciated for its strong reasoning and writing abilities. As one of the earliest models made publicly available, it has maintained a broad user base. Notably, ChatGPT integrates DALL·E 3 for image generation, giving it an advantage in creative applications where Grok-3 currently lacks native capabilities.

Claude, on the other hand, is best known for producing highly natural, human-like writing. This makes it particularly well-suited for storytelling and creative writing tasks where tone and nuance matter.

Lastly, Gemini, developed by Google, shares many similarities with ChatGPT but is deeply integrated with Google's ecosystem. It excels in real-time research and is a powerful assistant for users who rely on Google Docs, Search, and other Google services.

Table 2.3 shows the comparison between scores of LLM models.

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organizatic	License
1	1	chocolate (Early Grok-3)	1403	+6/-6	9992	xAI	Proprietary
2	3	Gemini-2.0-Flash-Thinking-Exp-01-21	1385	+4/-6	15083	Google	Proprietary
2	3	Gemini-2.0-Pro-Exp-02-05	1380	+5/-6	13000	Google	Proprietary
2	1	ChatGPT-4o-latest (2025-01-29)	1377	+5/-5	13470	OpenAI	Proprietary
5	3	DeepSeek-R1	1362	+7/-7	6581	DeepSeek	MIT
5	8	Gemini-2.0-Flash-001	1358	+7/-7	10862	Google	Proprietary
5	3	o1-2024-12-17	1352	+5/-5	17248	OpenAI	Proprietary
8	7	o1-preview	1335	+3/-4	33169	OpenAI	Proprietary
8	8	Qwen2.5-Max	1334	+5/-5	9282	Alibaba	Proprietary
8	7	o3-mini-high	1332	+5/-9	5954	OpenAI	Proprietary

2.3 Methods to benchmark LLMs performance

According to [5], LLM benchmarks are designed to collect the outputs of specific tasks and questions to evaluate the performance of large language models (LLMs) in a standardized manner. This process helps us determine which models perform better on the same tasks. There are four primary methods to compare LLM performance:

1. **Accuracy:** This metric depends on the percentage of answers that the LLM gets fully correct based on the given context.
2. **BLEU Score:** This score measures how closely the LLM’s output aligns with human-written references.
3. **Perplexity:** This metric indicates how confused the LLM is when facing a task. Lower perplexity means the model has better comprehension and understanding.
4. **Human Evaluation:** For outputs in specific fields, experts are required to judge the LLM’s output in terms of quality, relevance, and coherence.

2.4 Comparison between Short-Term memory & Long-term memory

Short-term memory in LLM agents functions similarly to human working memory or a computer's cache. It acts as a temporary storage system, retaining information only for the duration of the current session, and is cleared once the session ends. In contrast, long-term memory serves as a knowledge base, storing facts, concepts, and patterns learned during training. This information can be saved and restored for future sessions, allowing the agent to retain conversation context.

According to [6], short-term memory is ideal for scenarios requiring real-time adaptation and immediate references. It enables LLM agents to introduce new information when the user switches context or topic, ensuring the conversation remains relevant and cohesive.

On the other hand, long-term memory allows LLM agents to refer back to past chat history to retrieve previous information. For example, if a user tells the agent their name is "Dave," the agent should remember and recall this information even after several days. This makes long-term memory suitable for frameworks where similar questions or prompts might refer back to past interactions. It helps users save time and effort by avoiding the need to repeat lengthy prompts to continue from where they left off.

2.5 Multi-agent framework limitations

While multi-agent frameworks offer numerous benefits and advantages over single-agent frameworks, according to [1] they also present several limitations and challenges for developers:

1. **Coordinating Reasoning**: To solve user queries, agents debate and reason with each other. The main challenge is determining when they should stop to avoid endless looping.
2. **Managing Context**: Agents must keep track of information and conversations within the context of the user's query without deviating into unrelated topics. They also need to remember everything said in long group chats.
3. **Time and Cost**: More agents mean longer output generation times due to the need for discussion and debate. Additionally, more tokens are required, increasing the overall cost.

2.6 Agno’s Team Communication Method (Multi-agent)

According to Agno's documentation [7], when building a team of agents, there are three primary communication methods: Route Mode, Coordinate Mode, and Collaborate Mode.

Route Mode: In this mode, the Team Leader analyses the user’s request and routes it to the most appropriate team member based on the content. Only the selected agent is involved in generating the response, while the rest remain idle. This approach is efficient and fast, ideal for tasks that require a specialized response without collaboration.

Coordinate Mode: Here, the Team Leader breaks down the user's request into sub-tasks and delegates them to different team members based on their roles or expertise. The outputs from these members are then synthesized into a single, coherent response. This method requires the Team Leader to have strong reasoning capabilities, as they must understand and partition the task effectively. It is resource-intensive but allows for a more structured and role-based approach to problem-solving.

Collaborate Mode: In this approach, all team members receive the same task simultaneously. Each works independently, and their outputs are then compiled by the Team Leader into a final response. This method places less cognitive load on the Team Leader, as the focus is on integrating the results rather than decomposing the task. It is particularly useful when diverse perspectives or redundancy can enhance the quality of the output.

These communication strategies allow flexibility in how agent teams are utilized, depending on the nature of the task and the desired efficiency or depth of collaboration.

CHAPTER 3

System Methodology

The processes of the project were categorized into different phases in the development, which were project pre-development, multi-agent architecture, agent configuration, and test running the configuration.

3.1 Methodologies

The methodology chosen for the development of the cybersecurity digital consultant is Agile development. Agile is an iterative and flexible software development approach, particularly well-suited for small to medium-sized projects. One of its key strengths lies in its adaptability to changing requirements, which often emerge during the implementation phase as users provide feedback and new insights.

Agile is especially appropriate for this project due to the adoption of a relatively new tool, Agno for building the system. Since Agno is still being explored and its limitations are not yet fully understood, frequent refinement of requirements will be necessary. The Agile methodology supports this exploratory and adaptive process by allowing iterative cycles of development, continuous feedback, and quick adjustments. This ensures that the final product can evolve effectively as both the capabilities of Agno, and the specific needs of users become clearer throughout the development lifecycle.

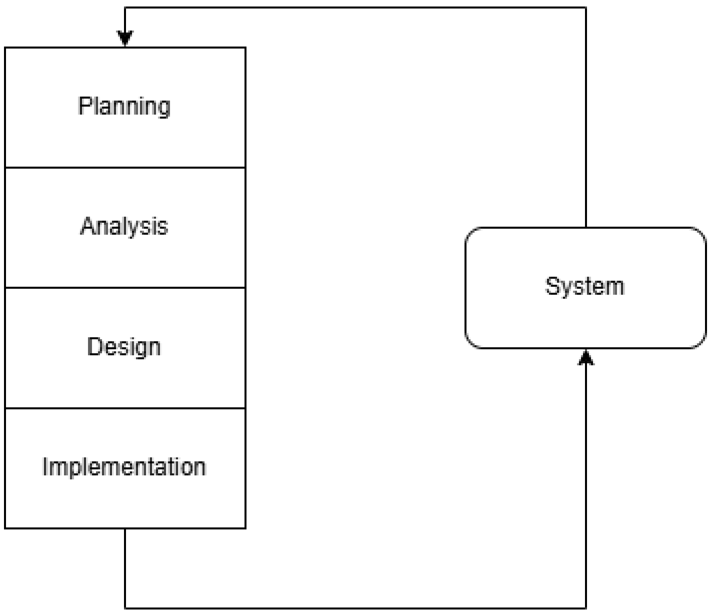


Figure 3.1 shows the framework of the Agile development methodology.

Planning phase

In the planning phase, the project proposal is developed, outlining key elements such as the project title, objectives, scope, and other foundational components. During this stage, the tools and technologies to be used are also identified. For this project, Agno and Visual Studio Code have been selected to develop customized agents, while the Agno Playground will be used to host and test these agents in a controlled environment.

Additionally, the roles and tasks of the agents to be built are clearly defined in this phase to ensure a structured development process. A critical part of the planning involves gaining a thorough understanding of how the Agno library operates. This includes exploring how to define agent attributes and configure the team workflows effectively, which are essential to ensuring proper functionality and integration of the agents within the system.

Analysis phase

In the analysis phase, all attributes and functionalities of the agents must be clearly defined. This includes determining the language models (LLMs) to be utilized and crafting the instructions that guide each agent's behavior. These instructions play a crucial role in shaping the agents' responses and output generation, as they directly influence how each agent performs its assigned task.

Furthermore, the team workflow configuration is also established during this phase. For instance, if the objective is to have a team agent distribute tasks based on each agent's specific role, the "coordinate" mode within Agno is selected. This mode enables structured communication and collaboration among agents, ensuring tasks are efficiently delegated and executed according to their designated responsibilities.

Design phase

In the design phase, the system architecture is developed, including a comprehensive system design diagram. A key component of this phase is the conversation flow design, which maps out the interactions between the team of agents and the users. This flow outlines how the cybersecurity digital consultant should communicate, respond to user input, and guide users through various cybersecurity-related tasks or queries.

Additionally, the output format and presentation style of responses generated by the team agent must be defined to ensure consistency and clarity in communication. Since the project utilizes Agno to build the chatbot, there is no need to develop a separate user interface. The entire system will be hosted and tested directly on the Agno Playground, streamlining the deployment and user interaction process.

Implementation phase

In the implementation phase, the development process officially begins. The coding will be carried out using Visual Studio Code, where a Python script will be created to build the cybersecurity digital consultant team. This involves developing and configuring the individual agents according to the design specifications established in the earlier phases.

Once the initial version of the team is assembled, the system will undergo test runs to evaluate functionality and performance. Based on the results, fine-tuning and adjustments will be made to optimize agent behavior, ensure accurate task execution, and improve overall system efficiency.

3.2 System Design

3.2.1 Hardware

The hardware involved in this project is a computer. A computer issued for the process of setting up the software required to run the Agno. As well as to test run and configure the agents and the multi-agent framework.

Table 3.2.1 Specifications of laptop

Description	**Specifications**
Model	MSI Katana 15 B13V
Processor	Intel Core i7-13620H
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX 4050 6GB GDDR6
Memory	8GB*2 DDR5 RAM
Storage	1TB NVMe PCle SSD

3.2.2 Software

The software that is required will be visual studio code. It is being used to edit the python file of the cybersecurity digital consultant.

3.2.3 Technologies involved

Table 3.2.3 Technologies involved and their respective function.

Technology	Function
Grok 3	LLM developed by xAI.
Python	Used to build the Team of agents.
Agno	Libraries of the agent's function and host the playground.

3.2.4 System Design Diagram

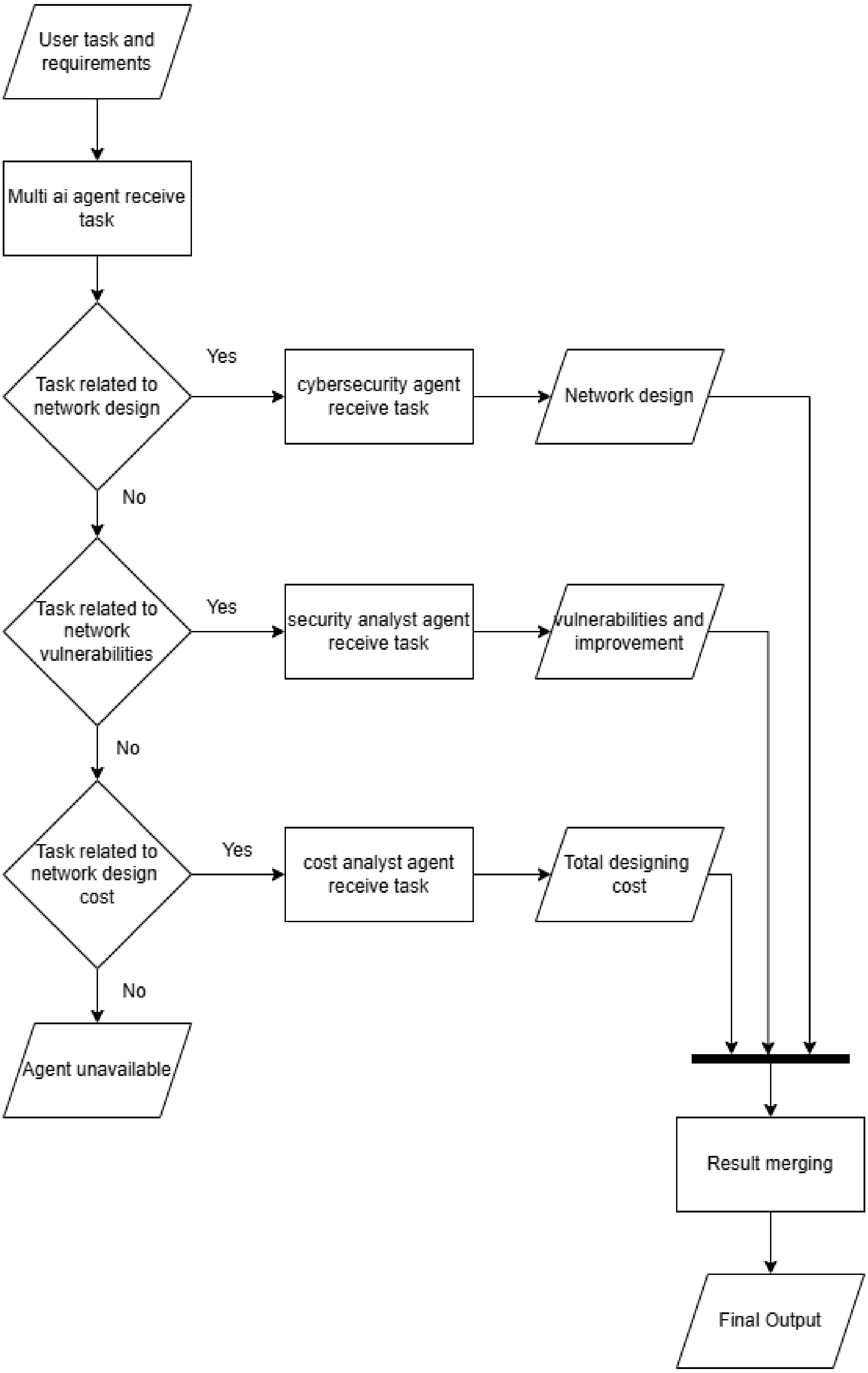


Figure 3.2.4 shows the system design of the cybersecurity digital consultant.

According to Figure 3.2.4, the process begins when the user submits a task along with its requirements to the team agent. Upon receiving the input, the team leader distributes the tasks among the team members. The cybersecurity agent is responsible for designing the network topology, the security analyst agent identifies existing network vulnerabilities and proposes mitigation strategies, and the cost analyst calculates the total cost of creating the network. Once all members have completed their respective tasks, the team leader compiles their responses and returns the final result to the user.

3.3 User requirements

Table 3.3.1 User requirements

User stories
As a user, I can type my question into the text box.
As a user, I can request to create a network topology and visualize it.
As a user, I can request to know the security weaknesses of the network and ways to overcome it.
As a user, I can know the total cost to create the network.
As a developer, I can configure the agent's attribute.
As a developer, I can add more agents.
As a developer, I can configure how the team interacts.
As a developer, I can define the output style the agents are going to generate.
As a developer, I can delete the project.

3.4 Timeline

The details of the timeline throughout the project are shown in the Gantt chart below.

Project Task	**Project Week**																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Planning																								
Forming project title																								
Research with existing system																								
Analysis																								
Identify project background																								
Identify problem statement and motivation																								
Literature review																								
Identify project scope and objectives																								
Identify contribution																								
Design																								
Identify Methodologies and user requirements																								
Draw System Design Diagram																								
Identify implementation																								

issues and challenges																								
Implementation																								
Develop the system																								
System testing																								
System deployment																								
Documentation																								

CHAPTER 4

System Design

4.1 System Block Diagram

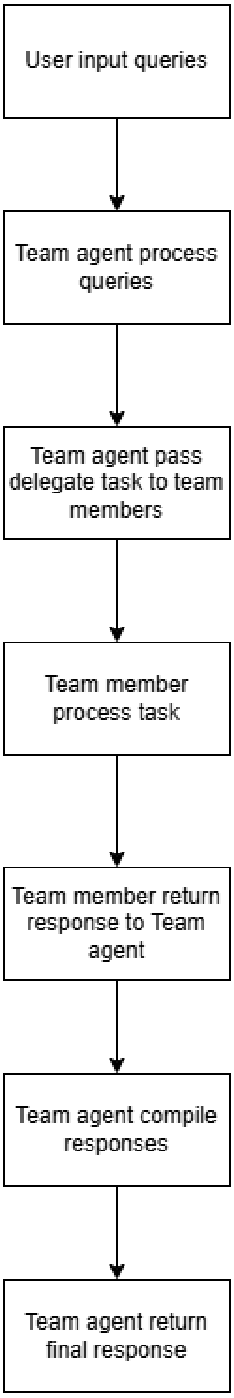


Figure 4.1 System block diagram.

Figure 4.1 illustrates the overall flow of the cybersecurity digital consultant system. The process begins when the user submits a query to the consultant. The team agent then interprets the input to understand the user's intent and specific requirements.

Based on this analysis, the team agent delegates the task to the appropriate team members, each assigned according to their designated roles and areas of expertise. These individual agents process their respective parts of the task and once completed, return their responses to the team agent.

The team agent then compiles and synthesizes the individual outputs into a coherent and finalized response, which is ultimately returned to the user to complete the interaction.

4.2 Multi Agent Communication Flow Diagram

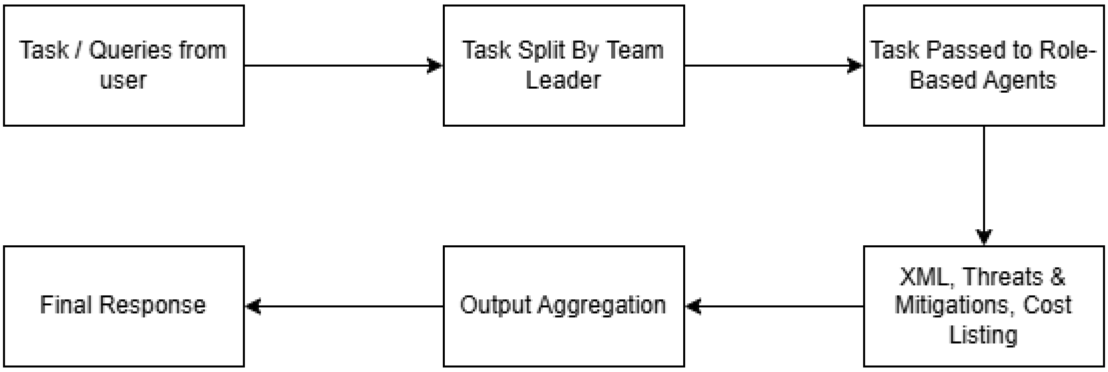


Figure 4.2 shows the Multi agent communication flow diagram.

CHAPTER 5

System Implementation

5.1 Setting up

5.1.1 Software

Before starting to develop the Cybersecurity Digital Consultant, there are four software needed to be installed and downloaded in my laptop:

1. Visual Studio Code version 1.92
2. Agno Libraries and dependencies
3. Python version 3.12
4. Google Chrome

Visual Studio Code

Visual Studio Code is the code editor being used in this project. It can be downloaded from https://code.visualstudio.com/download. The version used in the project is 1.99.3.

Google Chrome

Google Chrome is the web browser being used in this project. It can be downloaded from https://www.google.com/chrome. The software version is 135.0.7049.96.

5.1.2 Python Libraries

This project incorporates several Python libraries, including agno and openai. To begin we need to create a virtual environment and activate it. After activating the virtual environment, we can proceed with the installation of Agno and its dependencies. After that, we can move forward to setup the Agno.

The following are the steps of installing the libraries:

i. python3 -m venv agnoenv
ii. agnoenv/scripts/activate
iii. pip install -U agno openai
iv. ag setup

5.1.3 LLM Configuration

First, go to xAI console to create an API key. After that, go to visual studio code, start a terminal and activate the virtual environment. Then pass the key to the Agno system.

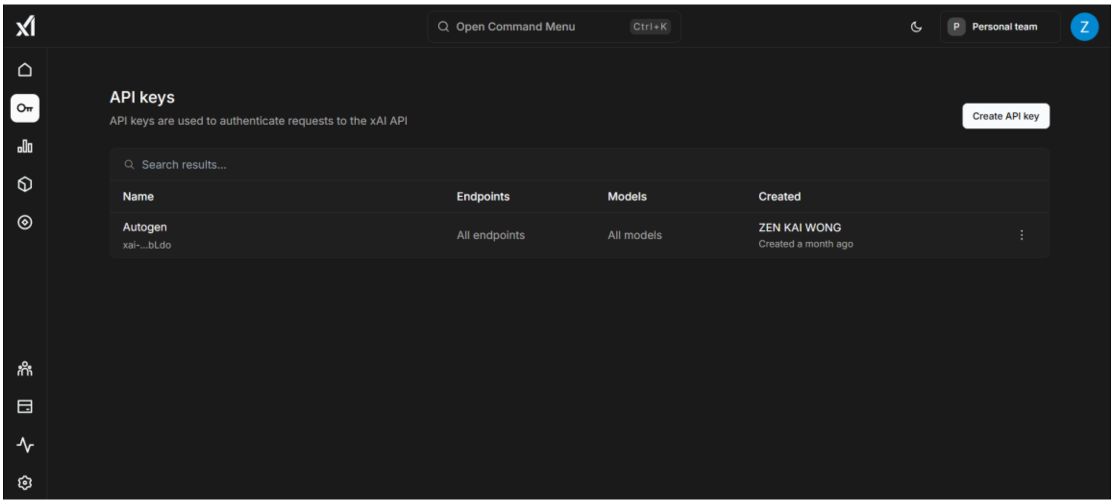


Figure 5.1.1 shows the xAI console with an API key being created.

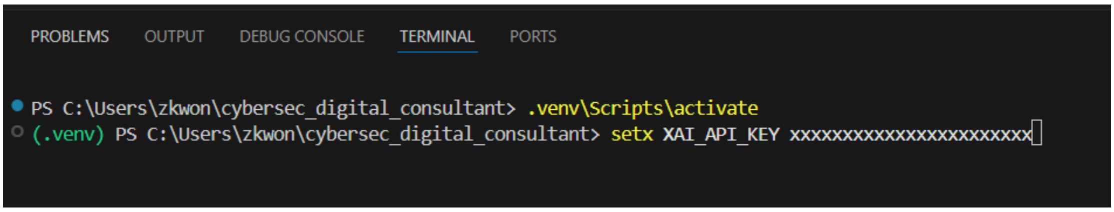


Figure 5.1.2 shows the passing of the xAI API Key into the system.

5.2 Preliminary Work

5.2.1 Agent Setting

For this project, several agents are being created, including Multi ai agent, Cybersecurity agent, Security Analyst agent, Cost Analyst agent. Each of them has a different configuration and serves different purposes, but they all use the Grok 3 LLM model for output generation. The table 4.2 below it shows the agents with their respective configurations.

Table 5.2 Agent's setting and parameters.

Name	Type of Agent	Skillset / Task	Model
Multi_ai_agent	Team	Split task / route to team members.	Grok-3-fast-beta
Cybersecurity_agent	Agent	Generates network topology in XML.	Grok-3-beta
Security_analyst_agent	Agent	Identifies threats and suggests mitigations.	Grok-3-beta
Cost_ analyst_agent	Agent	Lists hardware, costs, and estimate total cost.	Grok-3-beta

5.3 Comparison between Image Creation and XML

To better understand the differences in visualization results, a common task was assigned to the Grok single agent and Agno's multi-agent. In this case, the task involved designing a personal home network consisting of a router, a hub, and three computers.

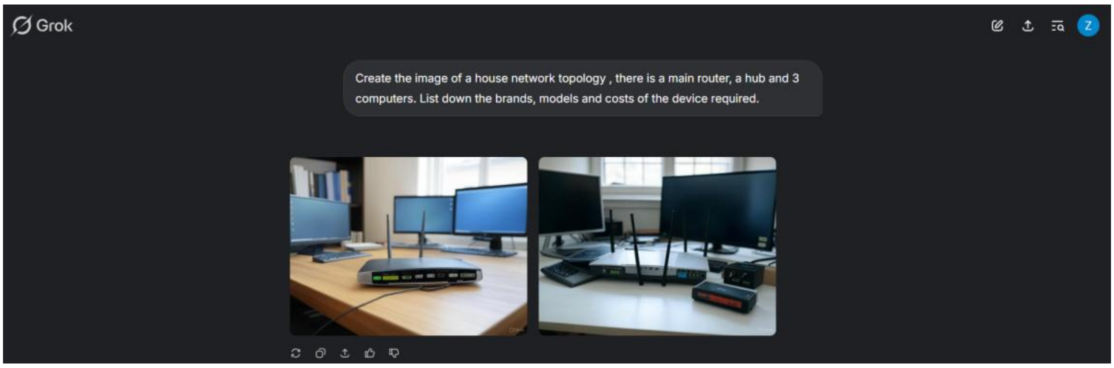


Figure 5.3.1 shows the image created by the single agent.

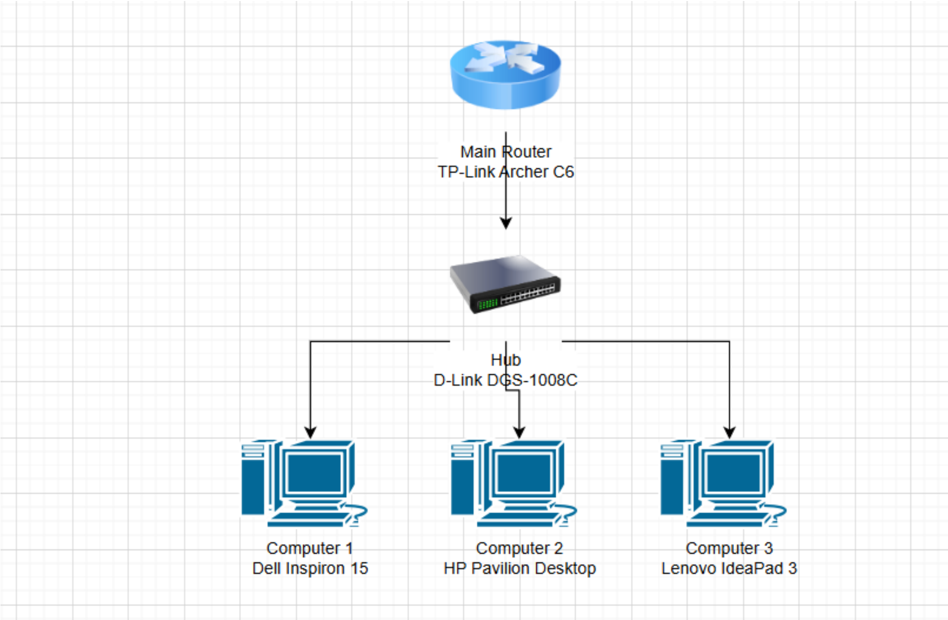


Figure 5.3.2 shows the XML network diagram topology created by Agno's multi-agent.

Based on the comparison in Figure 5.3.1 and Figure 5.3.2, it is evident that image generation does not produce the desired output. Most of the results generated through image-based methods are focused on visual aesthetics such as drawings, color schemes, or artistic styles—rather than the accurate representation of network components. In contrast, utilizing the XML format ensures that the agent generates only relevant content, specifically focusing on the creation of the network topology diagram.

The XML-based output offers a more structured and detailed view of the network, clearly showing the devices involved, their configurations, and the interconnections between them. This level of detail significantly enhances the user's understanding of the network architecture, making it far more effective and informative than the image-based approach.

5.4 Comparison between GROK's single agent and multi-agent.

To better understand the differences between various frameworks, a common task is given to both the single agent and multi-agent to visualize the differences in the output generated. In this case, the task is to create a personal home network which consists of a router, a hub and 3 computers.

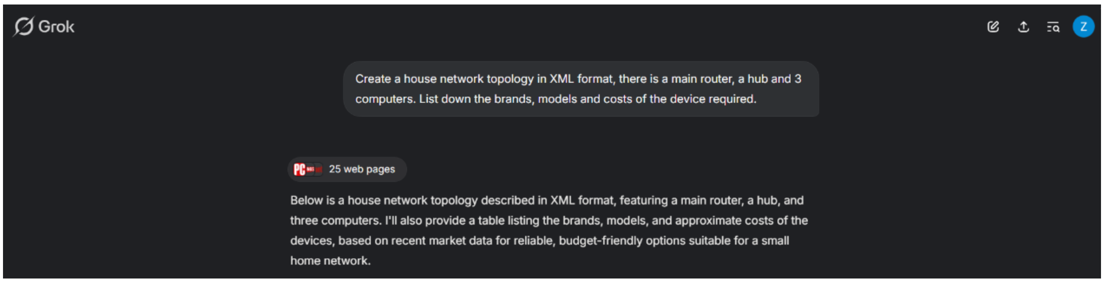


Figure 5.4.1 shows the task being passed on to the single agent.

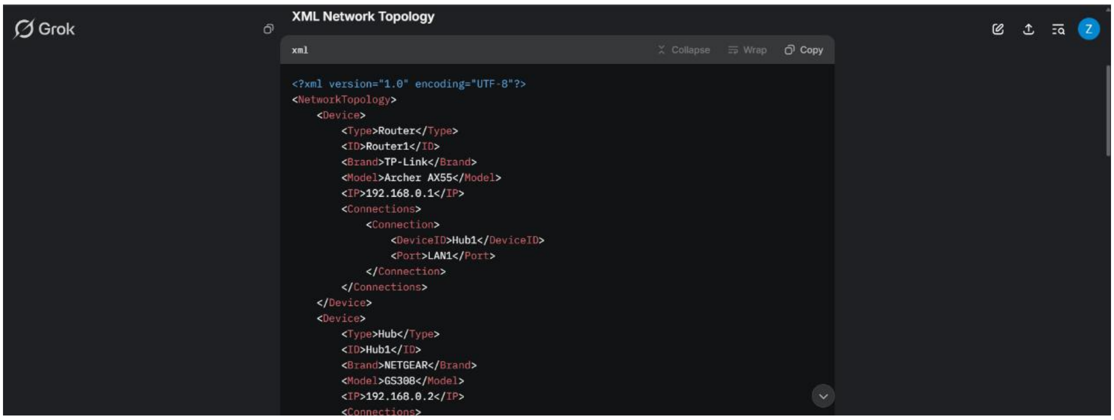


Figure 5.4.2 shows the partial code of the XML file being created by the single agent.

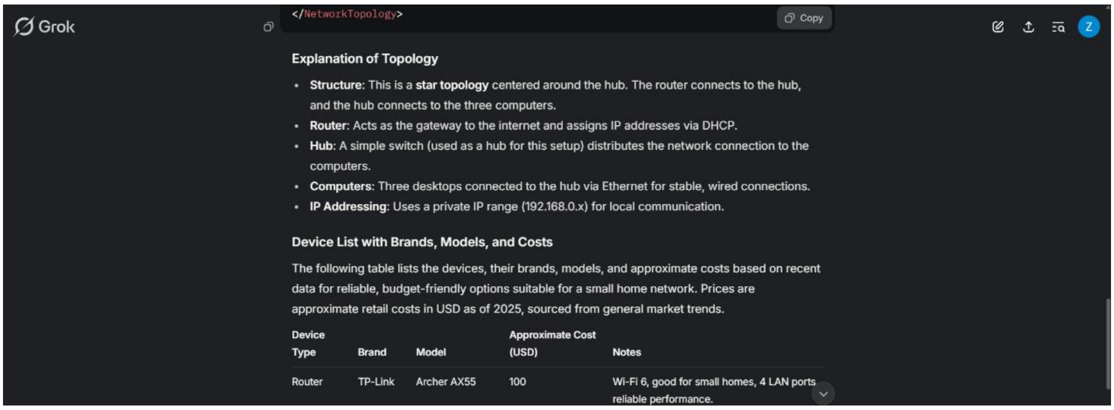


Figure 5.4.3 shows the explanation of the topology of the network being created and some additional information such as device lists.

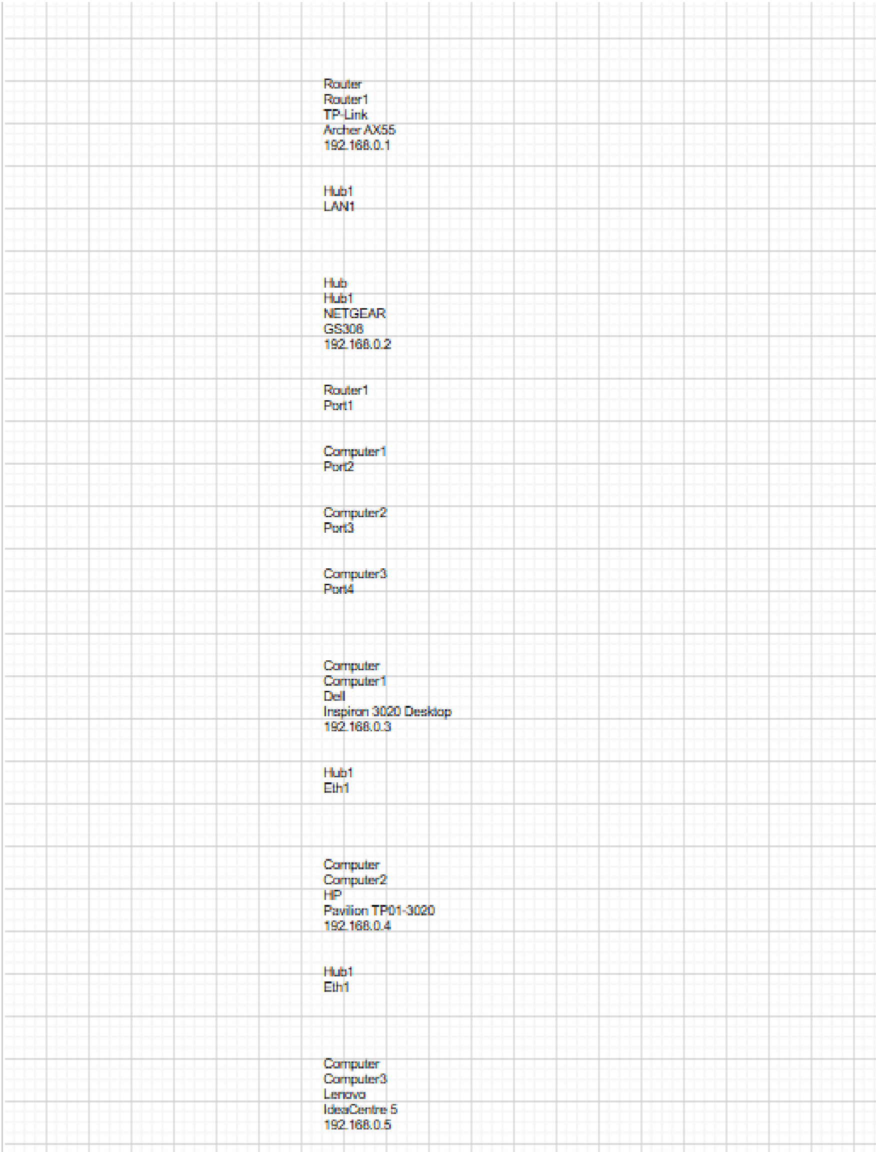


Figure 5.4.4 shows the output of the XML being generated by the single agent in Draw.io.

From the above results, we can understand that the single agent is able to derive the components needed and able to explain the topology design. However, the network topology being created in XML format does not show the devices image and the connections between devices.

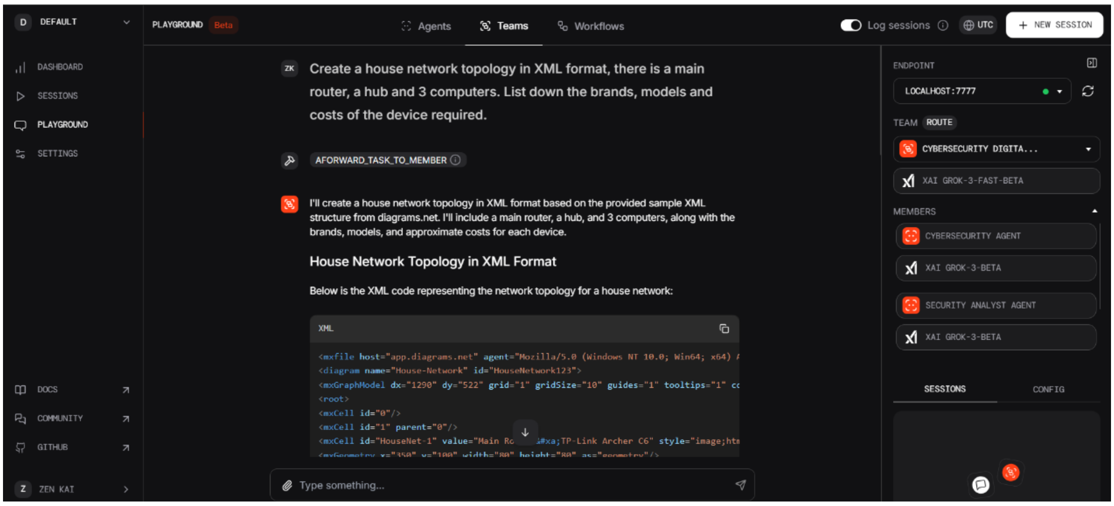


Figure 5.4.5 shows the task being passed on to the multi-agent.

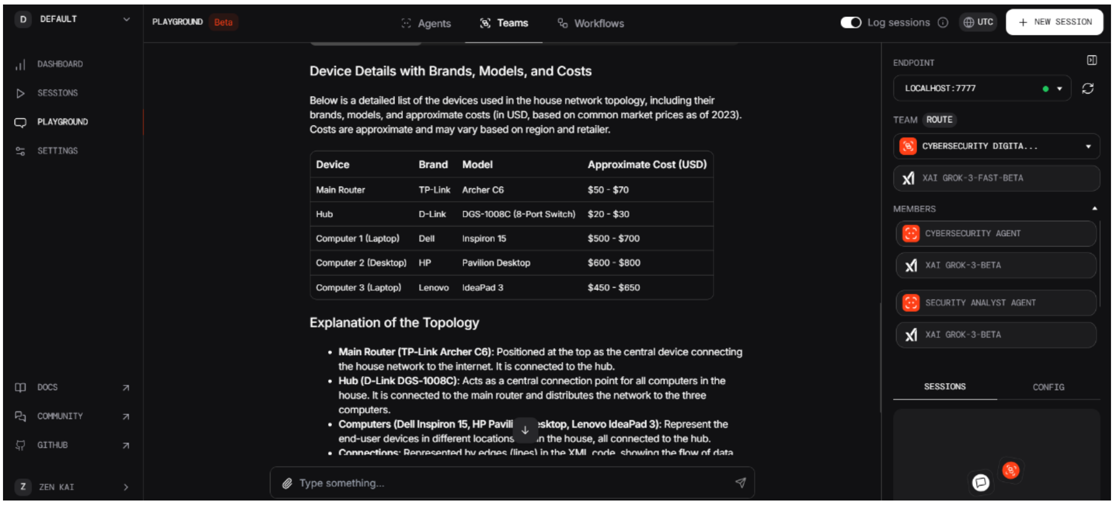


Figure 5.4.6 shows the output generated by the multi-agent.

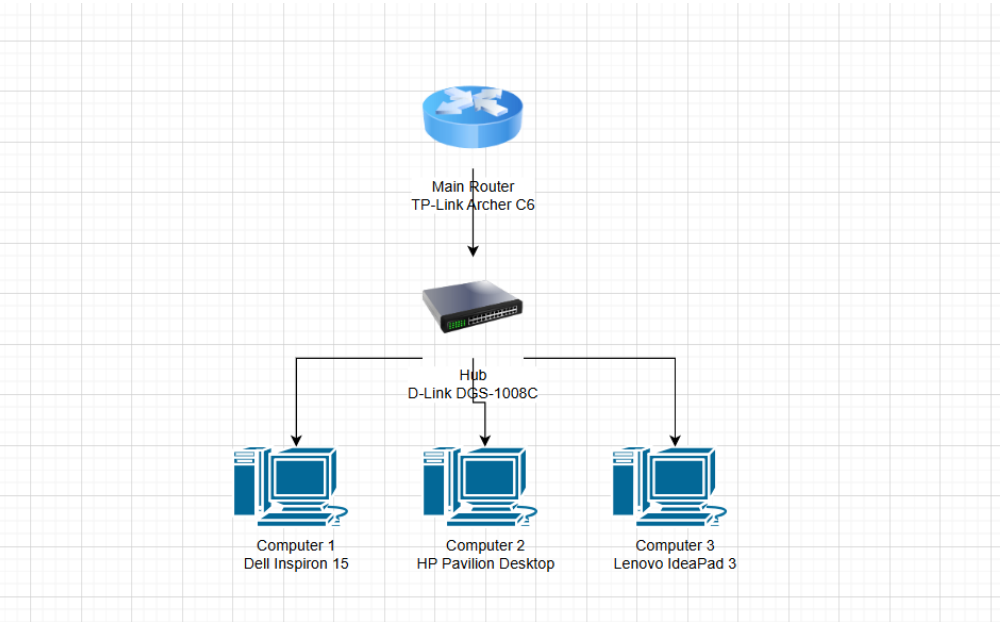


Figure 5.4.7 shows the network topology drawn by the multi-agent in XML format.

By comparing the results generated by the multi-agent and the single-agent systems, we can observe notable differences in their capabilities. The multi-agent system successfully produces a comprehensive network topology diagram, offering a clear and detailed visualization of device connections as well as specific device information. In contrast, the single-agent system falls short in generating a complete and coherent topology diagram, limiting the user's ability to fully visualize the network layout.

Despite this difference, both the single-agent and multi-agent systems are capable of providing detailed textual explanations of the network topology. Additionally, each system effectively generates a tabulated list of the components required to construct the network, including essential details such as the brand, model, and price of each device.

5.5 Comparison between AutoGen and Agno

During Project I, AutoGen is being used to host the multi-agent framework. However, AutoGen only supports API from llama, ChatGPT, Mistral and Gemini.

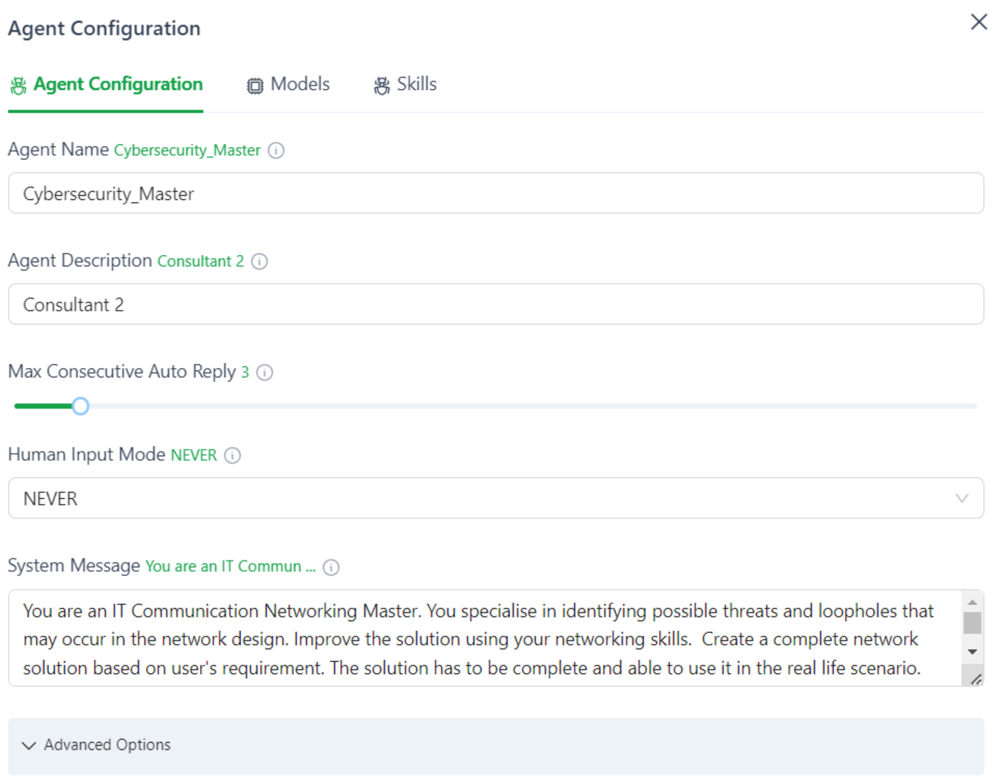


Figure 5.5.1 shows the configuration page of the agents for AutoGen Studio.

```python
# Create finance agent with enhanced capabilities
security_analyst_agent = Agent(
    name="Security Analyst Agent",
    role="Analyze and present the possible threats and vulnerabilities of the network topology and list the best practices to secu
    model=xAI(id="grok-3-beta"),
    instructions=[
        "Use tables to display threats and vulnerabilities",
        "list the best practices to secure the network using table format",
    ],
    tools=[DuckDuckGoTools()],
    show_tool_calls=True,
    markdown=True,
    monitoring=True,
)
```

Figure 5.5.2 shows the configuration of the agents for Agno.

By comparing Figure 5.4.1 and Figure 5.4.2, it is evident that AutoGen Studio offers a more user-friendly and visually appealing interface for configuring agents. However, it lacks support for the GROK API and operates within a fixed framework, which significantly limits the flexibility in customizing agent attributes.

After analyzing the agent configuration, the next step is to compare the outcomes produced by AutoGen's multi-agent system and Agno's multi-agent system when assigned an identical task: designing a university network. This comparison aims to evaluate the effectiveness, accuracy, and level of detail each system provides in generating the desired network topology.

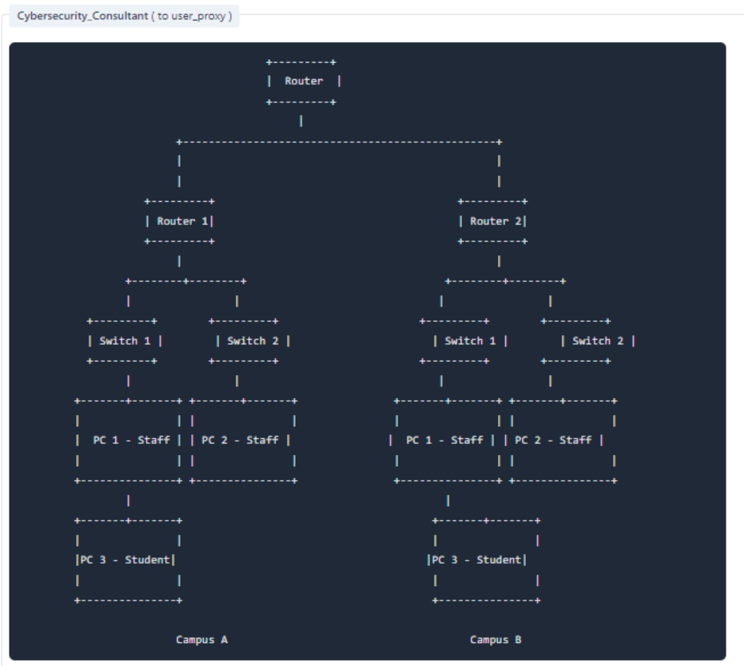


Figure 5.5.3 shows the network topology generated by AutoGen's multi-agent powered by Gemini API.

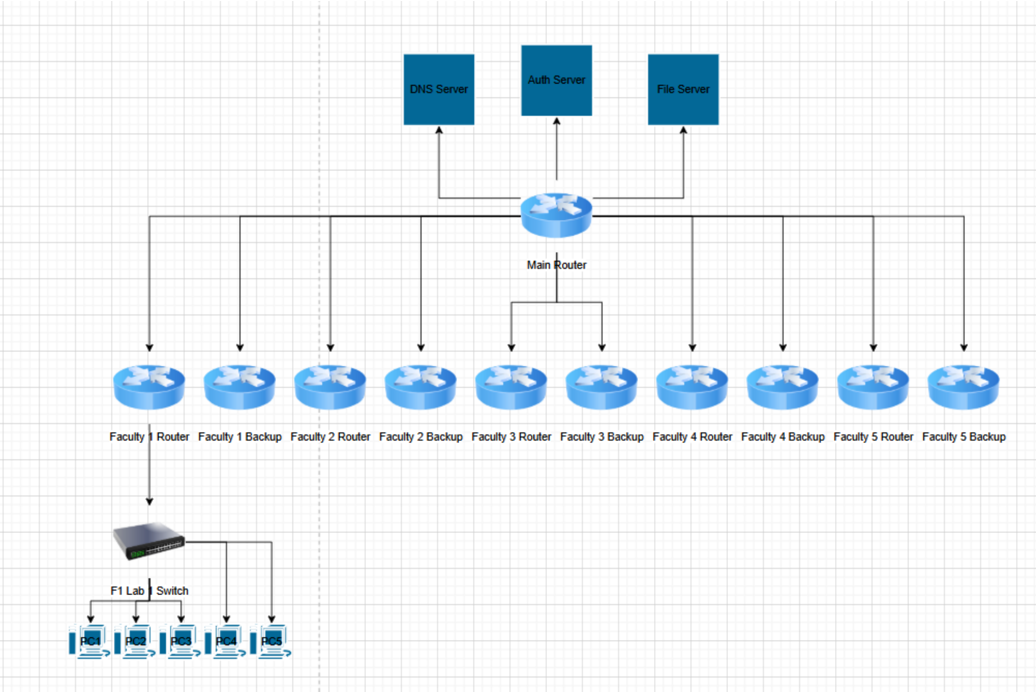


Figure 5.5.4 shows the network topology generated by Agno's multi-agent powered by GROK API.

By comparing Figure 5.5.3 and Figure 5.5.4, it becomes evident that AutoGen's multi-agent system, powered by Gemini, generates the output in ASCII format. This format presents challenges for users, as it lacks visual clarity making it difficult to interpret the devices involved and their interconnections due to its unstructured and misaligned layout. In contrast, Agno's multi-agent system, powered by GROK, produces a well-structured and visually coherent representation of the university network. It clearly illustrates the devices used and the connections between them, offering a more user-friendly and informative output.

Therefore, after comparing between these two multi-agent frameworks, Agno is found to be the most suitable for this project due to its generality and completeness, making it applicable to many use cases. For this project, which involves creating a cybersecurity digital consultant, multiple networking agents with specific configurations are required to ensure optimal performance. To better visualize and understand the configuration of these agents and the framework, Agno has been chosen. This tool supports a user interface with simple configurations, allowing users to easily create the prototype with GROK API.

5.6 Configuration

The multi-agent architecture comprises four agents: the multi ai agent, the cybersecurity agent, the security analyst agent, and the cost analyst agent. The multi ai agent hosts the group chat and acts as the coordinator. It employs a coordinate selection method, it will choose the agent to solve the respective parts based on the task given. After the multi ai agent receives the task, it coordinates among the three other agents. If the task is related to network topology generation it passes the user requirements to the cybersecurity agent. The cybersecurity agent, based on the user requirements, generates a network topology diagram in XML form and lists the components required to build the network. If the user's task is related to analyse the network, it is being passed to the security analyst agent to identify weaknesses in the network design and suggest possible improvements to enhance network security. Lastly, if the task is related to cost calculation it is being passed to the cost analyst to determine the cost of creating the network.

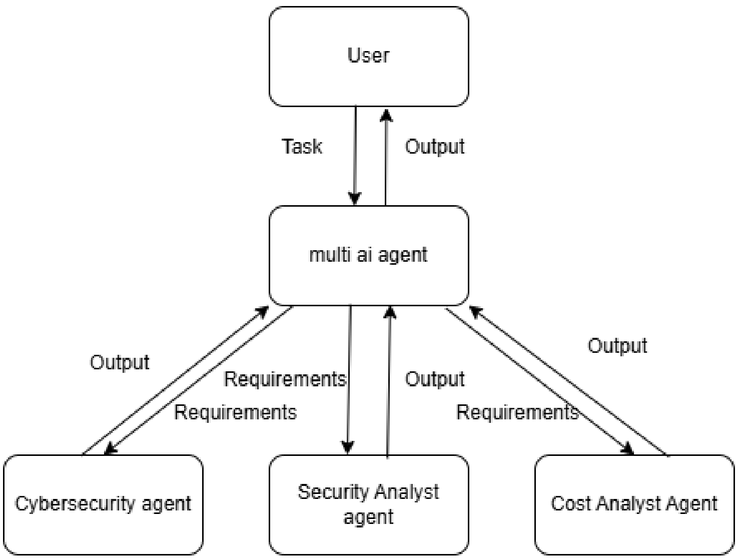


Figure 5.6 is the configured multi-agent architecture.

For this multi-agent architecture, the main role is played by the cybersecurity agent, who is responsible for generating the primary outputs, such as drawing the network diagram and listing the components. Secondly, the security analyst agent and cost analyst agent play crucial roles by checking for vulnerabilities and risks in the network and the cost to create the network. Lastly, the multi ai agent is responsible for coordinating and distributing tasks among the agents to ensure a smooth transition process.

Agent Profile and Skill:

1. **Multi Ai agent:** A team assistant. It is skilled at coordinating a group of other assistants to solve a task. After receiving the task from user it will pass to the respective agents to solve the task given.
2. **Cybersecurity agent**: An agent that is good at drawing network diagram and identifying components to build a network. It is a networking professional in designing network and building a network with relevant components.
3. **Security Analyst agent**: An assistant agent that is creative in analyzing network design. It will find problems inside a network and look for ways to solve the problem to improve the network security.
4. **Cost Analyst agent**: An assistant agent that is good at calculating the cost of creating the network. It is responsible for calculating the total cost of creating the network based on the user's requirements.

```
cybersecurity_agent = Agent(
    name="Cybersecurity Agent",
    role="create and design a network topology based on the requirements",
    model=xAI(id="grok-3-beta"),
    instructions=[
        "if the user provide a sample XML code/file, create the network topology based on the sample XML code/file",
        """<mxfile host="app.diagrams.net" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
<diagram name="Page-1" id="vBgPNcTUeMajD2yNG_6q">
<mxGraphModel dx="1290" dy="522" grid="1" gridSize="10" guides="1" tooltips="1" connect="1" arrows="1" fold="1" page="1" pag
<root>
<mxCell id="0"/>
```

Figure 5.6.1 shows the code snippet of the configuration for the cybersecurity agent.

```
</mxfile>"""
            "Create the network topology in xml format. ",
            "Create the network topology in a detailed and accurate way",
            "the network topology should be a valid XML code/file based on the sample XML code/file",
            "the network topology should include all the required components and connections",
        ],
        tools=[DuckDuckGoTools()],
        show_tool_calls=True,
        markdown=True,
        monitoring=True,  # Enable monitoring for better debugging
)
```

Figure 5.6.2 shows the code snippet of the configuration for the cybersecurity agent.

```
security_analyst_agent = Agent(
    name="Security Analyst Agent",
    role="Analyze and present the possible threats and vulnerabilities of the network topology and list the best practices
    model=xAI(id="grok-3-beta"),
    instructions=[
        "Use tables to display threats and vulnerabilities",
        "list the best practices to secure the network using table format",
    ],
    tools=[DuckDuckGoTools()],
    show_tool_calls=True,
    markdown=True,
    monitoring=True,
)
```

Figure 5.6.3 shows the code snippet of the configuration for the security analyst agent.

```
cost_analyst_agent = Agent(
    name="Cost Analyst Agent",
    role="Based on the user requirements, provide a cost listing to build the respective network topology",
    model=xAI(id="grok-3-beta"),
    instructions=[
        "Use tables to display model, brand, price, and quantity of the components",
        "choose the best price and quality ratio based on the user budget",
    ],
    tools=[DuckDuckGoTools()],
    show_tool_calls=True,
    markdown=True,
    monitoring=True,
)
```

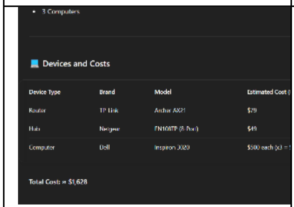
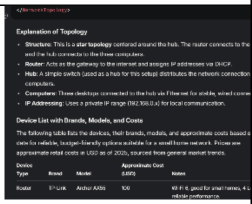
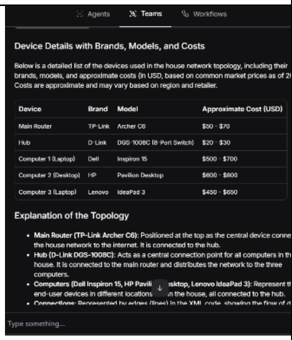
Figure 5.6.4 shows the code snippet of the configuration for the cost analyst agent.

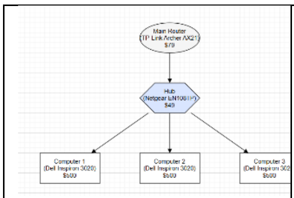
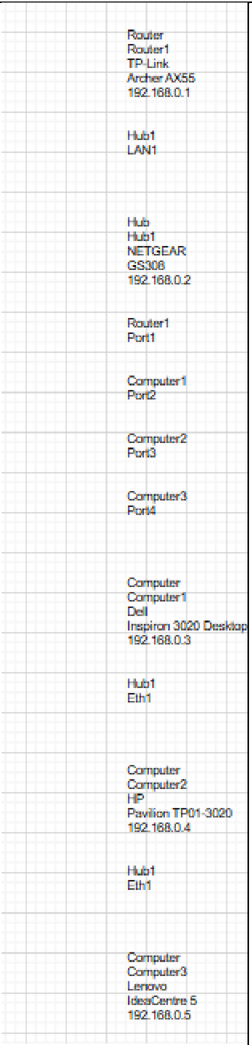
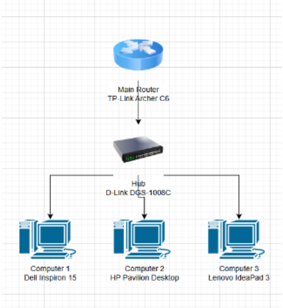
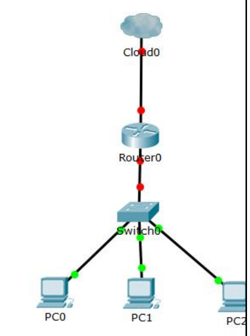
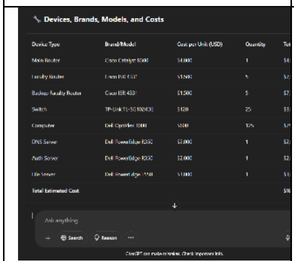
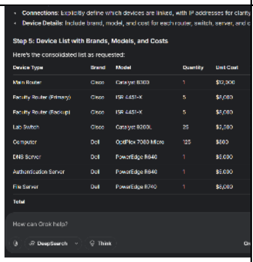
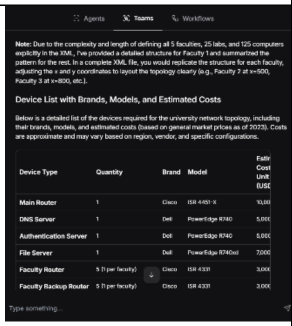
```
# Create multi-agent team with improved coordination
multi_ai_agent = Team(
    name="Cybersecurity Digital Consultant",
    description="You are a team that helps people to design network topology and solve networking problems.",
    mode="coordinate",
    members=[cybersecurity_agent, security_analyst_agent, cost_analyst_agent],
    show_tool_calls=True,
    markdown=True,
    monitoring=True,
    model=xAI(id="grok-3-fast-beta"),
    instructions=[
        "if user ask for a network topology, use the cybersecurity agent to create the network topology",
        "if user provide a sample XML code/file, pass it to the cybersecurity agent to create the network topology based on the sample XML code
        "if user ask for the possible threats and vulnerabilities, use the security analyst agent to analyze the network topology and list the
        "if user ask for a cost listing, use the cost analyst agent to provide a cost listing to build the respective network topology",
        "Use tables to display structured data",
        "Pass the output provided by the agents to the user in the original format",
        "Pass the output of the agents to the user in the same order as the agents are called",
    ],
    show_members_responses=True,
)
```

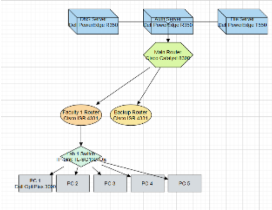
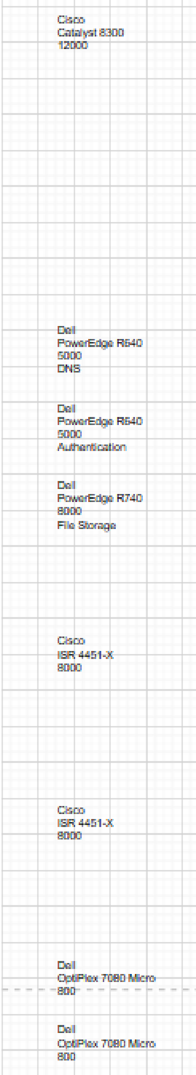
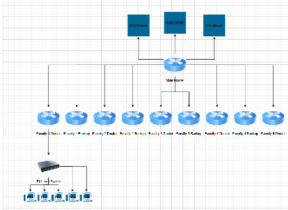
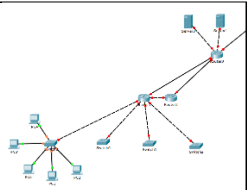
Figure 5.6.5 shows the code snippet of the configuration for the multi ai agent.

5.7 Comparison between ChatGPT, GROK Single agent, Agno multi-agent and Cybersecurity Professional

Table 5.7 shows the results generated by ChatGPT, GROK single agent, Agno multi-agent and Cybersecurity Professional.

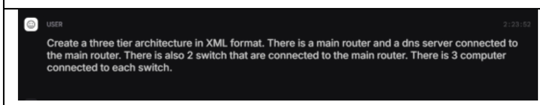
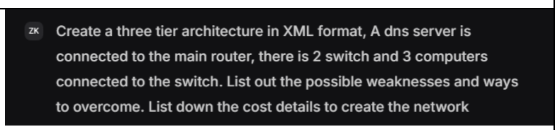
<table>
<tr><td>ChatGPT</td><td>Grok single agent</td><td>Agno multi agent</td><td>Cybersecurity Professional</td></tr>
<tr>
<td>
Able to generate the cost</td>
<td>
Able to generate the cost</td>
<td>
Able to generate a detailed overview of the cost</td>
<td>Unable to generate cost information</td>
</tr>
</table>

<table>
<tr>
<td>
Able to generate the framework but unable to generate the image of the components</td>
<td>
Unable to generate the network topology</td>
<td>
Able to generate the framework and the components</td>
<td>
Able to generate the framework and the components</td>
</tr>
<tr>
<td>
Able to generate a detailed overview of the cost</td>
<td>
Able to generate a detailed overview of the cost</td>
<td>
Able to generate a detailed overview of the cost</td>
<td>Unable to generate cost information</td>
</tr>
</table>

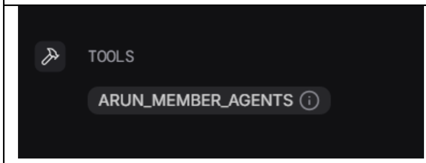
<table>
<tr>
<td>
Able to generate the framework but unable to generate the image of the components</td>
<td>
Unable to generate the network topology</td>
<td>
Able to generate the framework and the components</td>
<td>
Able to generate the framework and the components</td>
</tr>
</table>

5.8 Comparison between Collaborate mode and Coordinate mode

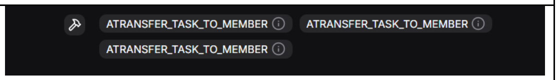
Table 5.8 shows the comparison between the results of collaborate mode and Coordinate mode.

<table>
<tr><th>Collaborate Mode</th><th>Coordinate Mode</th></tr>
<tr>
<td>
For “collaborate” mode the task given is the same for all agent, so we do not have to specify the details.</td>
<td>
For “coordinate” mode we need to specify all the relevant task handle by different</td>
</tr>
</table>

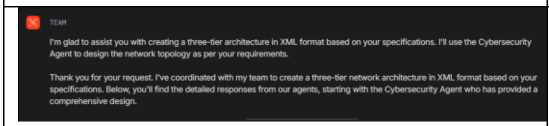
agents to ensure the team leader to call the agents.



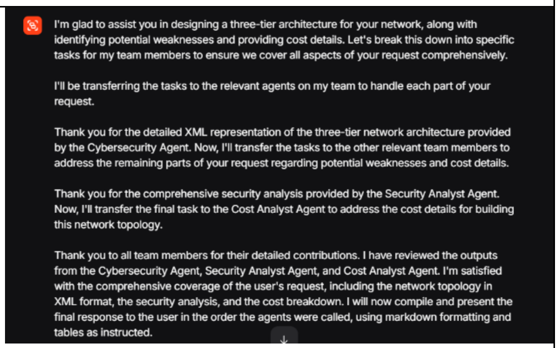
Tool is being called only once because the tasks passed to all agents are the same.



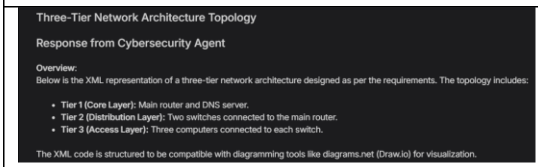
Since the team leader have to coordinate the members it will have to split the task before passing it to the members.



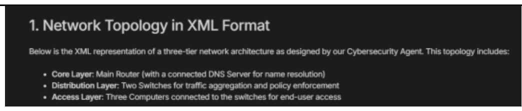
The team leader successfully passed the task to all agents at the same time.



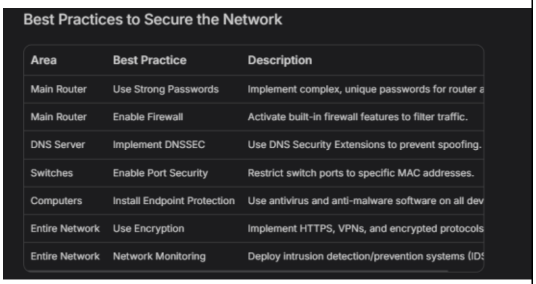
The task is being passed to each agent one by one.



The response from the cybersecurity agent is being retrieved.



The response from the cybersecurity agent is being retrieved.



Results from the security analyst agent are being retrieved

Threats and Vulnerabilities

Component/Layer	Threat	Vulnerability
Main Router (Tier 1)	Unauthorized Access	Weak passwords or default crede
Main Router (Tier 1)	Denial of Service (DoS) Attacks	Lack of rate limiting or flood prote
DNS Server (Tier 1)	DNS Spoofing/Cache Poisoning	Unsecured DNS server lacking DN
DNS Server (Tier 1)	DDoS Attacks	Exposed DNS server without prop
Switches (Tier 2)	VLAN Hopping	Misconfigured VLANs or lack of p
Switches (Tier 2)	MAC Address Spoofing	Lack of port security or MAC addı
Computers (Tier 3)	Malware Infection	Lack of endpoint protection or ou
Computers (Tier 3)	Phishing Attacks	Users accessing malicious links o
General Network	Eavesdropping/Man-in-the-Middle	Unencrypted traffic between devi
General Network	Insider Threats	Lack of access controls or monito

Best Practices to Secure the Network

Area/Component	Best Practice
Main Router (Tier 1 - Presentation Tier)	Use Strong Passwords and MFA
Main Router (Tier 1 - Presentation Tier)	Enable Firewall and Rate Limiting
Main Router (Tier 1 - Presentation Tier)	Regular Firmware Updates
DNS Server (Tier 1 - Presentation Tier)	Implement DNSSEC
DNS Server (Tier 1 - Presentation Tier)	Restrict DNS Queries with Firewall Rules
Switches (Tier 2 - Application Tier)	Configure Port Security
Switches (Tier 2 - Application Tier)	Implement VLAN Segmentation and Access Con
Computers (Tier 3 - Data Tier)	Install Endpoint Protection Software
Computers (Tier 3 - Data Tier)	Regular Software Updates and Patching
Computers (Tier 3 - Data Tier)	Conduct User Awareness Training
General Network (All Tiers)	Encrypt Network Traffic
General Network (All Tiers)	Implement Network Monitoring and Intrusion De
General Network (All Tiers)	ce Least Privilege Access

Results from the security analyst agent are being retrieved.

2. Detailed Cost Listing of Components

The Cost Analyst Agent has provided a detailed cost breakdown for the components in the network topology, including brand, model, and price. I have reviewed the data and harmonized the estimates to reflect a consistent and realistic mid-range budget for a small to medium-sized network setup. Below is the finalized cost listing based on the most recent updates provided by the team.

Component	Brand	Model	Quantity	Price per Unit (USD)	Total Price (USD)
Main Router	Cisco	RV340 Dual WAN VPN Router	1	230	230
DNS Server	Dell	PowerEdge T140	1	850	850
Switch 1	TP-Link	JetStream TL-SG3428	1	130	130
Switch 2	TP-Link	JetStream TL-SG3428	1	130	130
Computer (PC 1 - S1)	HP	ProDesk 400 G7	1	600	600
Computer (PC 2 - S1)	HP	ProDesk 400 G7	1	600	600
Computer (PC 3 - S1)	HP	ProDesk 400 G7	1	600	600
Computer (PC 1 - S2)	HP	ProDesk 400 G7	1	600	600
Computer (PC 2 - S2)	HP	ProDesk 400 G7	1	600	600
Computer (PC 3 - S2)	HP	ProDesk 400 G7	1	600	600
Total Cost	-	-	-	-	4,940

The results from the cost analyst agent are being retrieved.

3. Cost Details to Create the Network

Below is the cost breakdown provided by our Cost Analyst Agent for building the described three-tier network architecture, which includes a main router, a DNS server, two switches, and three computers.

Components and Cost Listing

Main Router (Core Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Main Router	Cisco	RV340 VPN Router	189.99	1	189.99

DNS Server (Access Layer)

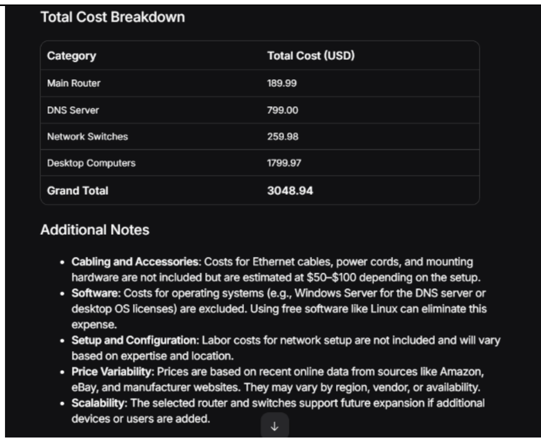
Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
DNS Server	Dell	PowerEdge T140	799.00	1	799.00

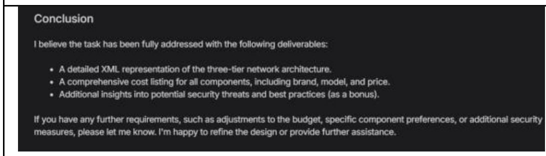
Switches (Distribution Layer)

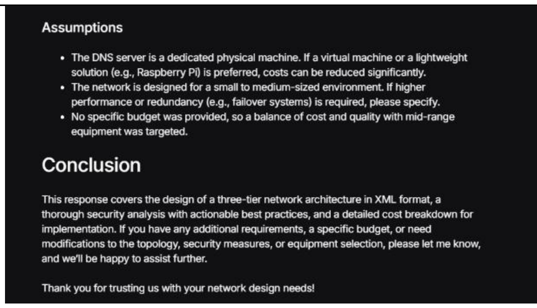
Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Network Switch	TP-Link	TL-SG3428 (28-Port)	129.99	2	259.98

Computers (Access Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Desktop Computer	HP	ProDesk 400 G7	599.99	3	1799.97

<table>
<tr>
<td></td>
<td>

The results from the cost analyst agent are being retrieved.</td>
</tr>
<tr>
<td>

Conclusion generated by the Team Leader.</td>
<td>

Assumptions and conclusion generated by the Team Leader.</td>
</tr>
</table>

From the table above, it’s clear that the "coordinate" mode produces more detailed and complete results than the "collaborate" mode, even though both handle the same task. This is especially true for the security and cost analysis parts.

This happens because in "coordinate" mode, the team leader breaks the task into smaller parts and gives each one to a specialist agent. Each agent focuses only on their area of expertise and ignores unrelated information. This makes the process faster and uses fewer tokens.

In "collaborate" mode, the full task is sent to every agent just as the user provided it. So, each agent must process the entire prompt, including parts they aren't specialized in. This causes more overlap, more processing, and higher token use.

Overall, "coordinate" mode is more efficient and cost-effective. It reduces token usage by the agents. However, the team leader's token use is about the same in both modes. In "coordinate" mode, the leader spends tokens to break down the prompt but saves on result compilation. In "collaborate" mode, the leader skips prompt breakdown but spends more tokens combining the agents' outputs.

5.9 Implementation Issues & Challenges

The main challenge was to find a suitable system that allows self-customizable agents while supporting a multi-agent framework. ChatGPT allows users to create customized agents, but it only supports a single-agent framework. Therefore, I initially experimented with Autogen Studio, the first multi-agent framework I tried. It supports multiple agents and has a flexible framework. However, it does not support GROK API, so I had no choice but to look for an alternative solution. Eventually, I found Agno, which supports manual configuration of agents, selection of LLMs, and multi-agent group chat.

However, configuring Agno was not easy. We have to customize the agents, such as the model of LLM, instructions and the role. As for the team agent, we need to configure the mode of communication and members that are involved. Any mistakes or problems could lead to the failure of output generation. The multi ai agent supports three types of agent selection methods which are route, coordinate, or collaborate. Determining which method to use took about a week of experimentation.

CHAPTER 6

System Evaluation and Discussion

6.1 System Testing

For system testing, the first step involves activating the virtual environment, followed by launching the playground.py script, which contains all the necessary information and configurations related to the agents.

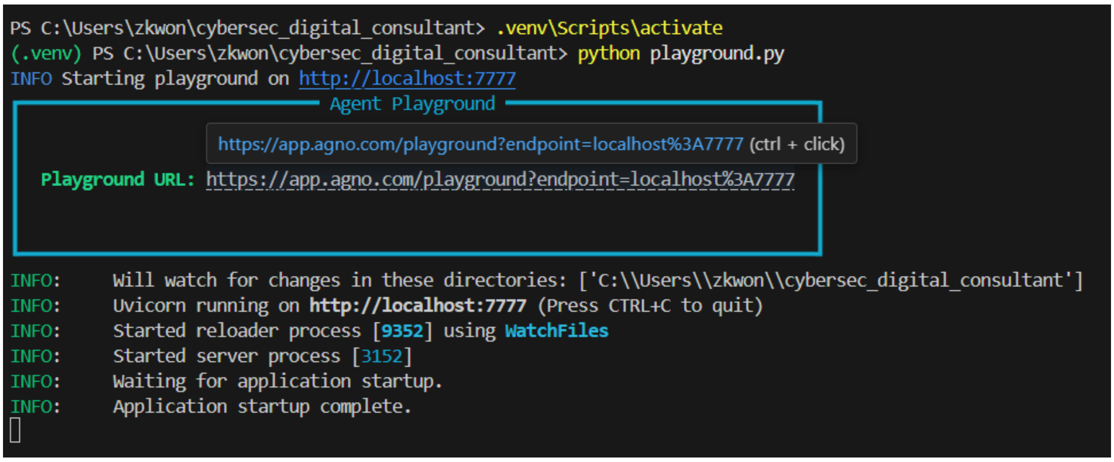


Figure 6.1.1 shows the starting process of Agno's playground.

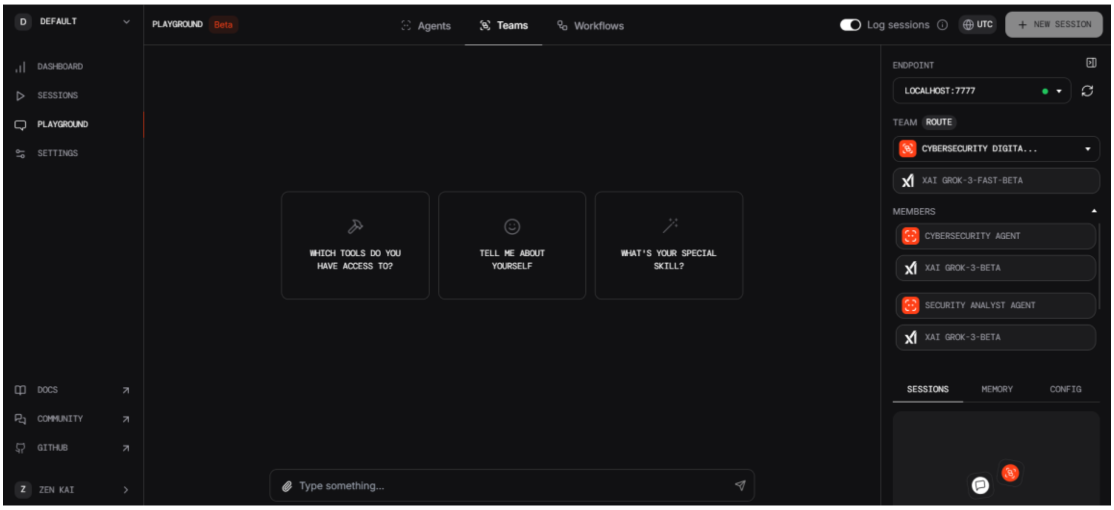


Figure 6.1.2 shows the localhost is being connected and the agents are being loaded.

Next, we assign the task to the agent. In this scenario, the multi-agent system is tasked with generating a company network. The network design should include a file server, a backup server, and an authentication server, all connected to a main router. Additionally, each of the company's three departments should be equipped with a dedicated switch and five computers. This setup aims to simulate a typical small-to-medium enterprise network infrastructure.

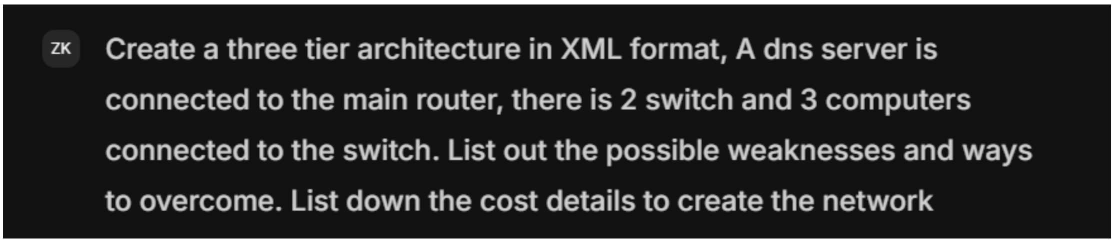


Figure 6.1.3 shows the task being given.

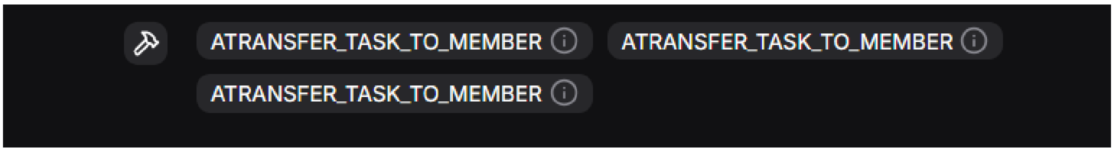


Figure 6.1.4 shows the team agent calling the tools to run member agents.

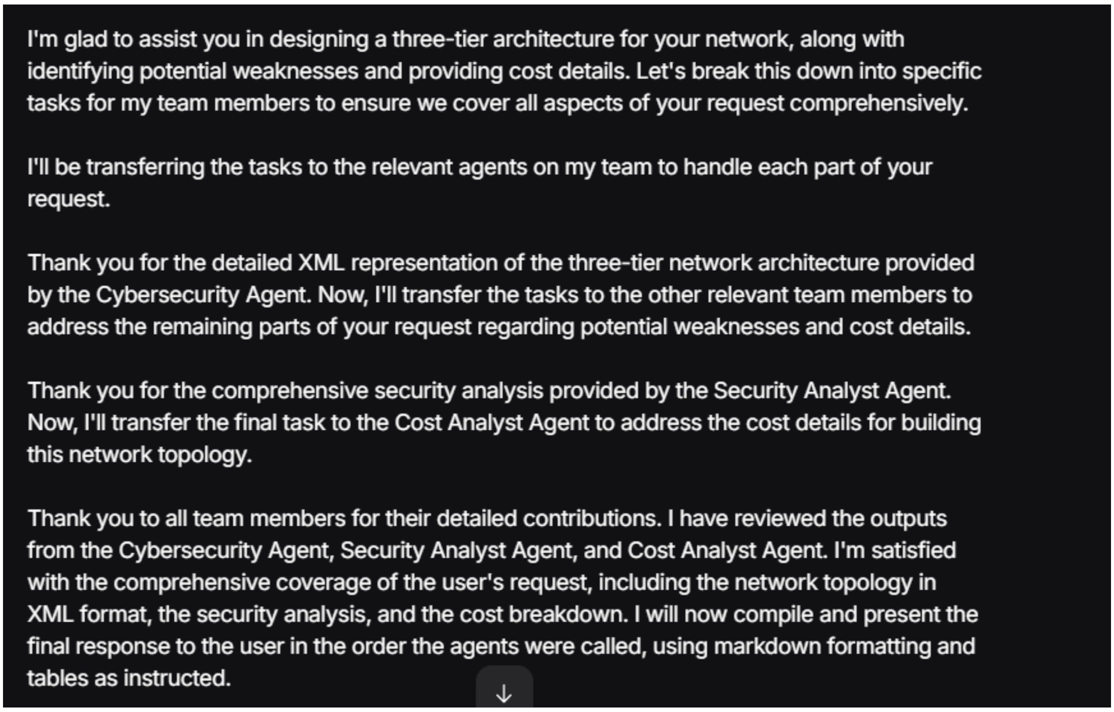


Figure 6.1.5 shows the team agent successfully break down the task and forward the respective part to the member agents.

This is the complete XML code being generated by the cybersecurity digital consultant.

<<mxfile host="app.diagrams.net" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36" version="24.7.12">

```
<diagram name="Page-1" id="threeTierNetwork">
<mxGraphModel dx="1290" dy="522" grid="1" gridSize="10" guides="1" tooltips="1" connect="1" arrows="1" fold="1"
page="1" pageScale="1" pageWidth="827" pageHeight="1169" math="0" shadow="0">
<root>
<mxCell id="0"/>
<mxCell id="1" parent="0"/>

<!-- Core Layer: Main Router -->
<mxCell id="coreRouter" value="Main Router"
style="image;html=1;image=img/lib/clip_art/networking/Router_Icon_128x128.png" vertex="1" parent="1">
<mxGeometry x="350" y="100" width="80" height="80" as="geometry"/>
</mxCell>

<!-- Core Layer: DNS Server connected to Main Router -->
<mxCell id="dnsServer" value="DNS Server"
style="shape=mxgraph.cisco.servers.dns_server;html=1;pointerEvents=1;dashed=0;fillColor=#036897;strokeColor=#
ffffff;strokeWidth=2;verticalLabelPosition=bottom;verticalAlign=top;align=center;outlineConnect=0;" vertex="1"
parent="1">
<mxGeometry x="450" y="100" width="80" height="80" as="geometry"/>
</mxCell>
<mxCell id="edgeRouterDns"
style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1"
source="coreRouter" target="dnsServer">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

<!-- Distribution Layer: Switch 1 -->
<mxCell id="switch1" value="Switch 1" style="image;html=1;image=img/lib/clip_art/networking/Switch_128x128.png"
vertex="1" parent="1">
<mxGeometry x="200" y="250" width="80" height="80" as="geometry"/>
</mxCell>
<mxCell id="edgeRouterSwitch1"
style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1"
source="coreRouter" target="switch1">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

<!-- Distribution Layer: Switch 2 -->
<mxCell id="switch2" value="Switch 2" style="image;html=1;image=img/lib/clip_art/networking/Switch_128x128.png"
vertex="1" parent="1">
<mxGeometry x="500" y="250" width="80" height="80" as="geometry"/>
</mxCell>
<mxCell id="edgeRouterSwitch2"
style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1"
source="coreRouter" target="switch2">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

<!-- Access Layer: Computer 1 connected to Switch 1 -->
```

```
<mxCell id="computer1" value="Computer 1" style="shape=mxgraph.cisco.computers_and_peripherals.macintosh;html=1;pointerEvents=1;dashed=0;fillColor=#036897;strokeColor=#ffffff;strokeWidth=2;verticalLabelPosition=bottom;verticalAlign=top;align=center;outlineConnect=0;" vertex="1" parent="1">
<mxGeometry x="150" y="400" width="99" height="64" as="geometry"/>
</mxCell>
<mxCell id="edgeSwitch1Computer1" style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1" source="switch1" target="computer1">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

<!-- Access Layer: Computer 2 connected to Switch 1 -->
<mxCell id="computer2" value="Computer 2" style="shape=mxgraph.cisco.computers_and_peripherals.macintosh;html=1;pointerEvents=1;dashed=0;fillColor=#036897;strokeColor=#ffffff;strokeWidth=2;verticalLabelPosition=bottom;verticalAlign=top;align=center;outlineConnect=0;" vertex="1" parent="1">
<mxGeometry x="250" y="400" width="99" height="64" as="geometry"/>
</mxCell>
<mxCell id="edgeSwitch1Computer2" style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1" source="switch1" target="computer2">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

<!-- Access Layer: Computer 3 connected to Switch 2 -->
<mxCell id="computer3" value="Computer 3" style="shape=mxgraph.cisco.computers_and_peripherals.macintosh;html=1;pointerEvents=1;dashed=0;fillColor=#036897;strokeColor=#ffffff;strokeWidth=2;verticalLabelPosition=bottom;verticalAlign=top;align=center;outlineConnect=0;" vertex="1" parent="1">
<mxGeometry x="500" y="400" width="99" height="64" as="geometry"/>
</mxCell>
<mxCell id="edgeSwitch2Computer3" style="edgeStyle=orthogonalEdgeStyle;rounded=0;orthogonalLoop=1;jettySize=auto;html=1;" edge="1" parent="1" source="switch2" target="computer3">
<mxGeometry relative="1" as="geometry"/>
</mxCell>

</root>
</mxGraphModel>
</diagram>
</mxfile>
```

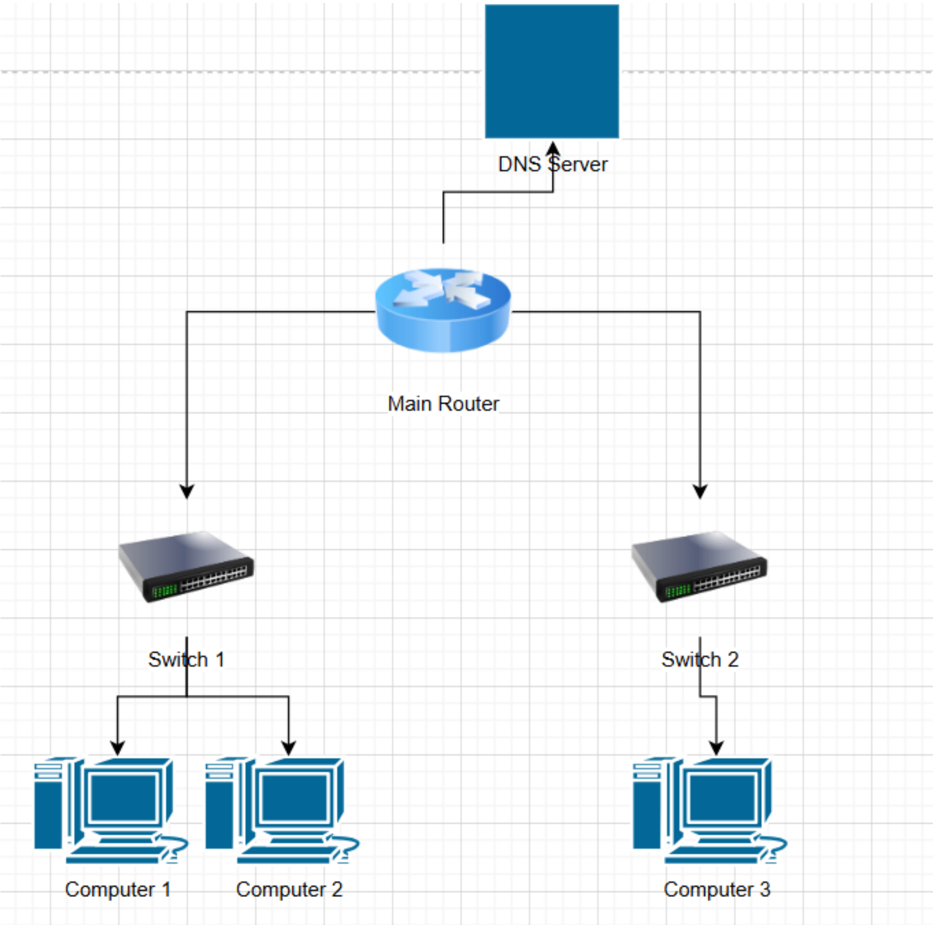


Figure 6.1.6 shows the network topology being generated.

2. Possible Weaknesses and Ways to Overcome Them

Below is a detailed security analysis provided by our Security Analyst Agent, identifying potential threats and vulnerabilities in the three-tier network architecture, along with best practices to secure the network.

Network Topology Overview

- **Tier 1 (Edge Layer/Presentation Tier):** Main router connected to the internet and internal network components, including a DNS server. This layer handles external access and user interactions.
- **Tier 2 (Distribution Layer/Application Tier):** Two switches facilitating communication between devices in the access layer and the router. This layer processes application logic and business rules.
- **Tier 3 (Access Layer/Data Tier):** Three computers connected to the switches for user access to the network. This layer often includes data storage and endpoint devices.

Threats and Vulnerabilities

Component/Layer	Threat	Vulnerability
Main Router (Tier 1)	Unauthorized Access	Weak passwords or default crede
Main Router (Tier 1)	Denial of Service (DoS) Attacks	Lack of rate limiting or flood prote
DNS Server (Tier 1)	DNS Spoofing/Cache Poisoning	Unsecured DNS server lacking DI
DNS Server (Tier 1)	DDoS Attacks	Exposed DNS server without prop
Switches (Tier 2)	VLAN Hopping	Misconfigured VLANs or lack of p
Switches (Tier 2)	MAC Address Spoofing	Lack of port security or MAC add
Computers (Tier 3)	Malware Infection	Lack of endpoint protection or ou
Computers (Tier 3)	Phishing Attacks	Users accessing malicious links o
General Network	Eavesdropping/Man-in-the-Middle	Unencrypted traffic between devi
General Network	Insider Threats	Lack of access controls or monito

Best Practices to Secure the Network

Area/Component	Best Practice
Main Router (Tier 1 - Presentation Tier)	Use Strong Passwords and MFA
Main Router (Tier 1 - Presentation Tier)	Enable Firewall and Rate Limiting
Main Router (Tier 1 - Presentation Tier)	Regular Firmware Updates
DNS Server (Tier 1 - Presentation Tier)	Implement DNSSEC
DNS Server (Tier 1 - Presentation Tier)	Restrict DNS Queries with Firewall Rules
Switches (Tier 2 - Application Tier)	Configure Port Security
Switches (Tier 2 - Application Tier)	Implement VLAN Segmentation and Access Con
Computers (Tier 3 - Data Tier)	Install Endpoint Protection Software
Computers (Tier 3 - Data Tier)	Regular Software Updates and Patching
Computers (Tier 3 - Data Tier)	Conduct User Awareness Training
General Network (All Tiers)	Encrypt Network Traffic
General Network (All Tiers)	Implement Network Monitoring and Intrusion De
General Network (All Tiers)	ce Least Privilege Access

Figure 6.1.7 shows the Security Analysis.

3. Cost Details to Create the Network

Below is the cost breakdown provided by our Cost Analyst Agent for building the described three-tier network architecture, which includes a main router, a DNS server, two switches, and three computers.

Components and Cost Listing

Main Router (Core Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Main Router	Cisco	RV340 VPN Router	189.99	1	189.99

DNS Server (Access Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
DNS Server	Dell	PowerEdge T140	799.00	1	799.00

Switches (Distribution Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Network Switch	TP-Link	TL-SG3428 (28-Port)	129.99	2	259.98

Computers (Access Layer)

Component	Brand	Model	Price (USD)	Quantity	Total Cost (USD)
Desktop Computer	HP	ProDesk 400 G7	599.99	3	1799.97

Total Cost Breakdown

Category	Total Cost (USD)
Main Router	189.99
DNS Server	799.00
Network Switches	259.98
Desktop Computers	1799.97
Grand Total	**3048.94**

Additional Notes

- **Cabling and Accessories:** Costs for Ethernet cables, power cords, and mounting hardware are not included but are estimated at $50–$100 depending on the setup.
- **Software:** Costs for operating systems (e.g., Windows Server for the DNS server or desktop OS licenses) are excluded. Using free software like Linux can eliminate this expense.
- **Setup and Configuration:** Labor costs for network setup are not included and will vary based on expertise and location.
- **Price Variability:** Prices are based on recent online data from sources like Amazon, eBay, and manufacturer websites. They may vary by region, vendor, or availability.
- **Scalability:** The selected router and switches support future expansion if additional devices or users are added.

Figure 6.1.8 shows the cost listing of the components required and the total cost to create the network.

Assumptions

- The DNS server is a dedicated physical machine. If a virtual machine or a lightweight solution (e.g., Raspberry Pi) is preferred, costs can be reduced significantly.
- The network is designed for a small to medium-sized environment. If higher performance or redundancy (e.g., failover systems) is required, please specify.
- No specific budget was provided, so a balance of cost and quality with mid-range equipment was targeted.

Conclusion

This response covers the design of a three-tier network architecture in XML format, a thorough security analysis with actionable best practices, and a detailed cost breakdown for implementation. If you have any additional requirements, a specific budget, or need modifications to the topology, security measures, or equipment selection, please let me know, and we'll be happy to assist further.

Thank you for trusting us with your network design needs!

Figure 6.1.9 shows the assumptions and conclusions.

Task Completion

From the results generated by the team of agents, it is evident that the output is comprehensive. The network topology of the company is clearly visualized, potential security weaknesses and vulnerabilities are identified, and the estimated cost of building the network is provided. Based on these outcomes, we can conclude that the cybersecurity digital consultant hosted on Agno effectively meets our requirements and expectations and managed to complete the task.

Quality of the Cybersecurity Digital Consultant

The results indicate that the performance of the cybersecurity digital consultant is notably high, as it consistently delivers output within a short time frame. It has proven to be reliable, showing no signs of system crashes or access issues during solution generation. However, there are a few critical dependencies that pose potential risks. The system is currently hosted on Agno's servers, meaning that any downtime on their end would directly impact our ability to access or operate the platform. Additionally, the solution agents rely on GROK’s API keys, and if the allocated token quota is exhausted, the system will be unable to generate responses. These dependencies highlight the need for contingency planning to ensure uninterrupted service.

Collaboration between the agents

The results demonstrate that the team leader effectively receives the task, decomposes it into distinct components, and delegates these to the appropriate team members. Each team member then processes their assigned portion and generates the corresponding output. Once completed, the team leader compiles all contributions and delivers the final result to the user. This process highlights a clear collaboration between the team leader and team members, primarily through task distribution and result aggregation.

However, it is important to note that the team members do not engage in direct collaboration with one another. This is by design, as each member is specialized in a specific domain, and unnecessary communication between them would not contribute to the quality or efficiency of the output. In fact, inter-agent communication consumes additional tokens, increasing the overall cost of output generation. Moreover, excessive communication can lead to longer processing times, further impacting performance.

Creativity of the solution generated

Based on the results from several test runs, it is evident that the creativity of the generated solutions is fairly standard. The cybersecurity agent consistently produces appropriate and functional network designs tailored to the task at hand. However, these designs tend to be more structured and conventional, with variations primarily in the choice of components rather than in innovative architecture or layout. Similarly, the security analyst and cost analyst agents generate practical and usable outputs aligned with the given requirements. Overall, while the system may not exhibit a high degree of creativity, it reliably delivers effective and task-specific solutions that meet user needs.

Relevance of the solution generated

The results confirm that the solutions generated by the Cybersecurity Digital Consultant are highly relevant to the user's queries. One of its key strengths is its ability to produce a comprehensive network topology diagram, enabling users to visualize the overall network architecture, something that other systems do not offer. Additionally, the Security Analyst Agent contributes by identifying potential vulnerabilities within the proposed system and suggesting appropriate mitigation strategies. Lastly, the Cost Analyst Agent complements the solution by generating a detailed cost breakdown for the network setup, ensuring that users have a clear understanding of the financial implications. Together, these components deliver a well-rounded and actionable output tailored to the user's requirements.

Score of the Cybersecurity Digital Consultant

Table 6.1 shows the Score of the Cybersecurity Digital Consultant.

Areas	Score		
	0	1	2
Novel Strategy Generation	No new or creative strategies used; follow only the obvious or provided steps.	Slight deviation from expected behavior, but not clearly novel.	Demonstrates a creative or unexpected solution that is contextually appropriate.
			√

Logical Task Decomposition	No clear structure, task completion is ad hoc or linear without subtask identification.	Some subtasks identified but sequence is unclear or loosely connected.	Breaks the task into well-defined, logically ordered subtasks.
			√
Role-Aware Collaboration	Agents operate independently without referencing roles or responsibilities.	Occasional role-based actions or references.	Clear, consistent role-based coordination
			√
Emergent Communication Behavior	No communication or basic command exchanges only.	Some back-and-forth interaction, questions, or clarifications.	Rich dialogue showing negotiation, confirmation, or strategic collaboration.
		√	
Adaptability / Replanning	No adjustment in strategy when encountering failure or confusion.	Basic recovery or retry but not explained.	Recognizes failure, proposes and executes a revised approach.
		√	
Outcome Effectiveness	Task not completed or result is invalid.	Task completed but with flaws or low quality.	Task completed correctly and meets high quality standards.
			√

Total Score: 10/12

6.2 Project Challenges

During the development of the cybersecurity digital consultant, I encountered several challenges that required significant time and problem-solving. One of the initial hurdles was selecting an appropriate framework to host the multi-agent system. Initially, I used AutoGen Studio for this purpose during Project 1. However, the platform presented several limitations such as restricted agent capabilities and attribute customization as well as critical incompatibility with the GROK API. This prompted me to search for an alternative framework that offered both flexibility and GROK API support.

Fortunately, I discovered Agno, a more adaptable system capable of hosting the multi-agent framework. However, I had no prior experience with Agno, so I had to start from scratch, relying heavily on the available documentation to learn the platform. This challenge was compounded by the fact that there were no existing or similar projects related to the cybersecurity digital consultant, meaning I had to independently design and develop the entire solution through trial and error.

Another significant challenge was finding a reliable method to visualize the network topology. After testing various LLMs and image-generation tools available in the market, I found that none of them could consistently produce accurate or well-structured network topology diagrams even for simple three-tier architecture. To overcome this, I devised an alternative approach: configuring the agent to generate the network topology in XML file format, which could then be imported into Draw.io for accurate and clear visualization of the network structure.

6.3 Objective Evaluation

This project was guided by five key objectives, each critical to the successful development of the cybersecurity digital consultant.

The first objective was to create a custom set of networking agents. This included the development of a cybersecurity agent responsible for generating the network topology diagram, a security analyst agent tasked with identifying system vulnerabilities and proposing solutions, and a cost analyst agent that calculates the total cost of building the network infrastructure.

The second objective focused on implementing persistent memory for agents. This capability allows agents to refer to previous outputs and leverage historical data to enhance performance and accuracy in future tasks, enabling more consistent and intelligent behavior across sessions.

The third objective was the development of a multi-agent LLM framework tailored to cybersecurity architecture and solutions. For this, a cybersecurity digital consultant was built to serve as the central coordinator. It manages task distribution and communication among the agents using the “coordinate” communication method, ensuring seamless teamwork in solving user-defined problems.

The fourth objective involved deploying and configuring the agents using the Agno platform. Agno was selected for its flexibility in agent creation, customizable configurations, and, most importantly, its support for the GROK API which powers the agents and enables advanced processing capabilities.

Finally, the fifth and most significant objective was to visualize the generated network topology in XML format. This capability distinguishes the project as the first multi-agent framework able to produce a complete, accurate, and real-world-applicable network topology diagram. The XML output can be directly imported into Draw.io, making the network design both functional and visually accessible.

6.4 Concluding Remark

In conclusion, this project successfully meets all the objectives that were initially set. Through testing, it has demonstrated stability and the ability to generate the expected output particularly in the accurate visualization of the network topology diagram. Although several challenges and difficulties arose during the development and integration phases, each was systematically addressed and resolved. As a result, the cybersecurity digital consultant was successfully developed, marking a significant milestone in multi-agent system applications for cybersecurity solutions.

CHAPTER 7

Conclusion & Recommendations

7.1 Conclusion

After conducting research on multi-agent frameworks and comparing various Large Language Models (LLMs) currently available, it was determined that Agno offers the flexibility and generality needed for configuring agents and communication flows based on user requirements. Consequently, Agno has been selected for this project. The Cybersecurity Digital Consultant, utilizing Agno, can create a comprehensive network, including network topology design, and a detailed list of required components. Each agent within the framework has a unique perspective, they collaboratively provide a network solution that can be implemented in real-world scenarios.

This approach empowers users without a background in cybersecurity or networking to design networks tailored to their specific use cases. It enhances their understanding of network design and required components while reducing costs by eliminating the need for professional consultation. Additionally, the Cybersecurity Digital Consultant is available 24/7, allowing users to obtain solutions instantly, thereby increasing efficiency in network creation. Ultimately, this Cybersecurity Digital Consultant represents a novel solution aimed at benefiting society by addressing the critical need for network security in the ICT era.

7.2 Recommendations

Although the cybersecurity digital consultant has been successfully deployed and is ready for use, there are still several areas where improvements can be made to enhance its functionality and user experience.

Firstly, the current system requires users to manually copy the generated XML code and upload it to Draw.io in order to visualize the network topology diagram. This step could be streamlined by integrating Draw.io directly into the system, allowing the agents to automatically feed the XML code into the visualization tool and render the network diagram instantly. However, achieving this integration would require in-depth research to understand Draw.io's backend processes and API capabilities, as well as how it can be embedded and connected within the Agno framework.

Furthermore, hosting the cybersecurity digital consultant on a dedicated server would significantly improve accessibility. By deploying the system online and providing users with a direct URL, the tool would become much more convenient to access and interact with, especially for non-technical users or organizations seeking a plug-and-play cybersecurity planning solution.

REFERENCES

[1] "Multi-agent LLMs in 2024 [+frameworks] | SuperAnnotate," *Superannotate.com*, 2024. https://www.superannotate.com/blog/multi-agent-llms#:~:text=Single%2Dagent%20systems%20are%20good (accessed Aug. 06, 2024).

[2] M. Langmajer, "Grok 3 vs ChatGPT vs DeepSeek vs Claude vs Gemini – Which AI Is Best in February 2025? | Fello AI," Fello AI, Feb. 23, 2025. https://felloai.com/2025/02/grok-3-vs-chatgpt-vs-deepseek-vs-claude-vs-gemini-which-ai-is-best-in-february-2025/ (accessed Apr. 24, 2025).

[3] *Softkit.dev*, 2025. https://www.softkit.dev/blog/grok-3-vs-gemini-chatgpt-copilot-2025-comparison/ (accessed Apr. 24, 2025).

[4]"Chatbot Arena (formerly LMSYS): Free AI Chat to Compare & Test Best AI Chatbots," *Lmarena.ai*, 2024. https://lmarena.ai/?leaderboard

[5] "LLM Benchmarks: Understanding Language Model Performance," Humanloop: Collaboration and evaluation for LLM applications, Mar. 16, 2024. https://humanloop.com/blog/llm-benchmarks (accessed Aug. 06, 2024).

[6] RandomTrees, "Short-term vs. long-term LLM memory: When to use prompts vs. long-term recall?," Medium, Nov. 22, 2023. https://randomtrees.medium.com/short-term-vs-long-term-llm-memory-when-to-use-prompts-vs-long-term-recall-0f78e39f9929 (accessed Aug. 06, 2024

[7] "Introduction - Agno," *Agno.com*, 2025. https://docs.agno.com/teams/introduction (accessed Apr. 22, 2025)

Poster

FACULTY OF INFORMATION TECHNOLOGY

APPENDIX