CONTENTS

CHAPTER - 1

1. INTRODUCTION

In today's hyperconnected world, information flows at an unprecedented pace across political, economic, technological, and social domains. Organizations ranging from startups and multinational companies to government policy units depend on accurate and timely analysis to make strategic decisions. However, the overwhelming volume of global news, misinformation, and rapid changes in international policies make it extremely difficult for human analysts to keep up.

To address this challenge, this project proposes an **Autonomous Global News & Policy Analyst**, a **multi-agent AI Think Tank system** capable of automatically gathering real-time global news, verifying facts across multiple sources, performing structured analysis, and generating strategic insights for businesses and governments. It combines **Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG), agent-based automation, credible fact-checking, and strategic reasoning frameworks** to function like a miniature McKinsey-style consultant.

The system is designed to work autonomously with minimal human supervision, capable of analyzing dozens of articles within seconds, extracting trends, predicting impacts, and generating policy-level recommendations that are data-backed, consistent, and unbiased.

1.1 Purpose of the project

The purpose of this project is to develop an intelligent system that:

- **Monitors global news autonomously** using real-time APIs
- **Cross-verifies information** to prevent misinformation
- **Analyzes events** using standard frameworks like
  - SWOT
  - PESTEL
  - Porter's Five Forces
  - Economic & geopolitical impact models
- **Generates policy suggestions** tailored to governments, companies, or industries
- **Visualizes insights** using charts, dashboards, and summaries

  The core goal is to **replace manual intelligence processing** with an autonomous agent pipeline that can deliver rapid, accurate, and actionable insights.

1.2 Problem with Existing Systems

Existing systems and workflows show several limitations:

1. Manual and Time-Consuming Analysis
Analysts require hours or days to read and interpret multiple sources. This delay often results in outdated insights.

2. High Information Overload
Thousands of news articles, reports, and policy updates are published each hour. Human analysts cannot keep up with the volume.

3. Misinformation and Lack of Verification
Most news aggregators show headlines but do not verify authenticity. This leads to biased or false interpretations.

4. Fragmented Tools
Professionals often use separate tools for:

- news reading
- fact verification
- data visualization
- risk analysis
- policy drafting

There is **no integrated system** that combines all these capabilities into one pipeline.

5. Lack of Analytical Depth
Traditional tools cannot:

- Identify long-term implications
- Understand geopolitical consequences
- Predict business/industry impact
- Generate policy recommendations

Hence, there is a strong need for a **unified intelligent system** that can perform all these tasks automatically.

1.3 Proposed System

The proposed system introduces an **AI-driven, multi-agent pipeline** where each agent performs a specialized role. The agents collaborate to deliver end-to-end global intelligence.

1. Retriever Agent

- Connects to news APIs (NewsAPI, GDELT, Reuters RSS, etc.)
- Fetches real-time articles across countries and sectors
- Extracts metadata: category, region, keywords, timestamp

2. Fact-Checker Agent

- Cross-verifies claims with alternate sources
- Uses Wikipedia, Google Fact Check, or AI-based verification
- Assigns credibility scores to filter out low-trust data

3. Analyst Agent
Applies structured analytical frameworks such as:

- **SWOT Analysis**
- **PESTEL Analysis**
- **Porter's Five Forces**
- **Economic/Business Impact Models**
- **Risk and Opportunity Mapping**

4. Policy & Strategy Agent

- Generates government policy recommendations
- Suggests strategic actions for companies
- Identifies short-term and long-term outcomes

- Provides actionable insights for decision-makers

5. Visualizer Agent

- Produces graphs, dashboards, and summaries
- Generates slide-style reports
- Highlights major trends and insights

6. Data Storage & RAG Layer

- Stores articles in a vector database
- Uses embeddings for semantic search
- Retrieves relevant context for accurate analysis

1.4 Scope of the Project

The scope includes:

In Scope

- Real-time global news retrieval
- Fact-checking and credibility evaluation
- Multi-agent pipeline implementation
- RAG-based semantic search
- Strategic analysis and policy recommendation
- Dashboard visualizations

Out of Scope

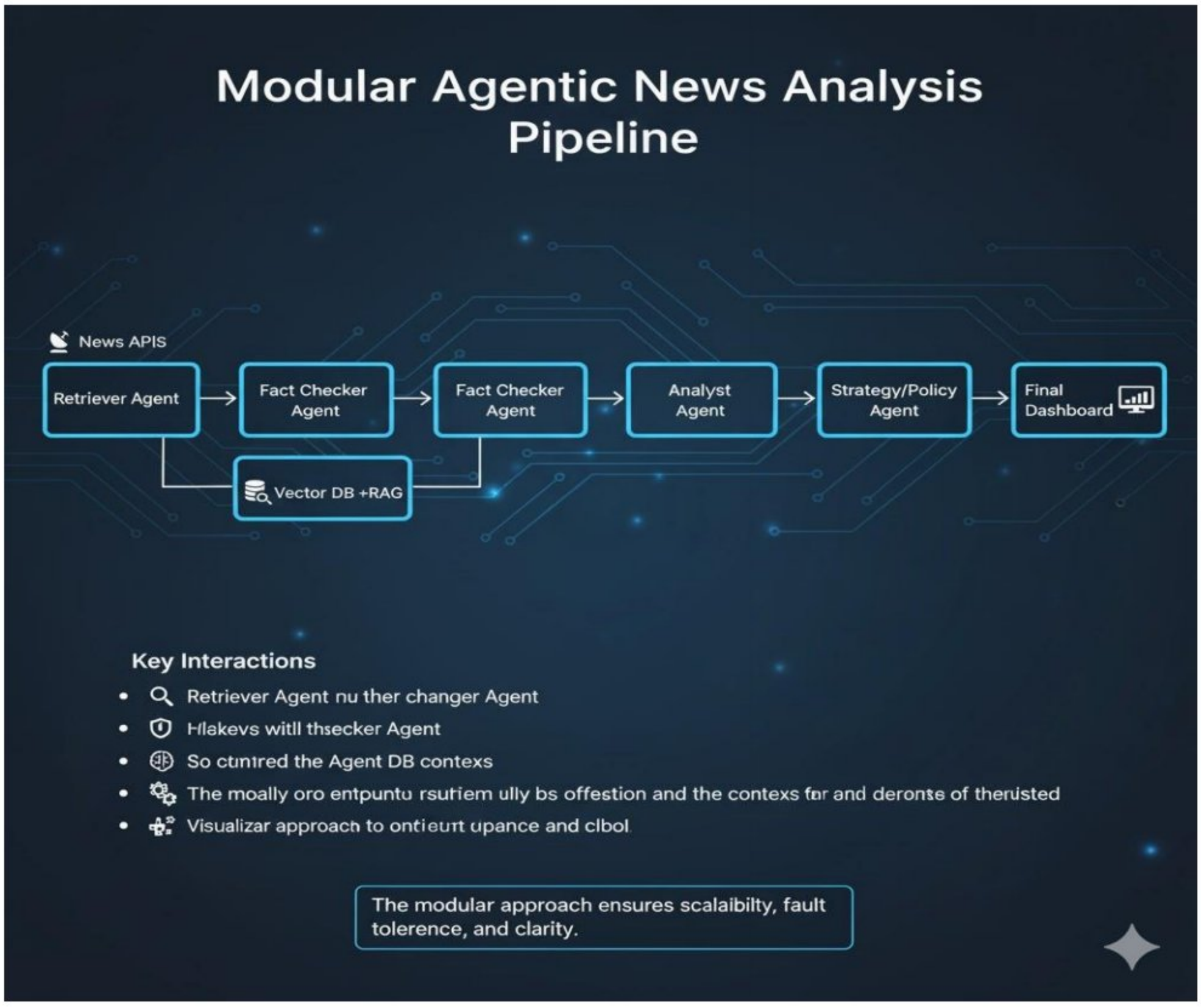
- High-frequency financial trading
- Political predictions or election forecasting
- Deepfake or misinformation generation
- Monitoring private or encrypted sources

Target Users

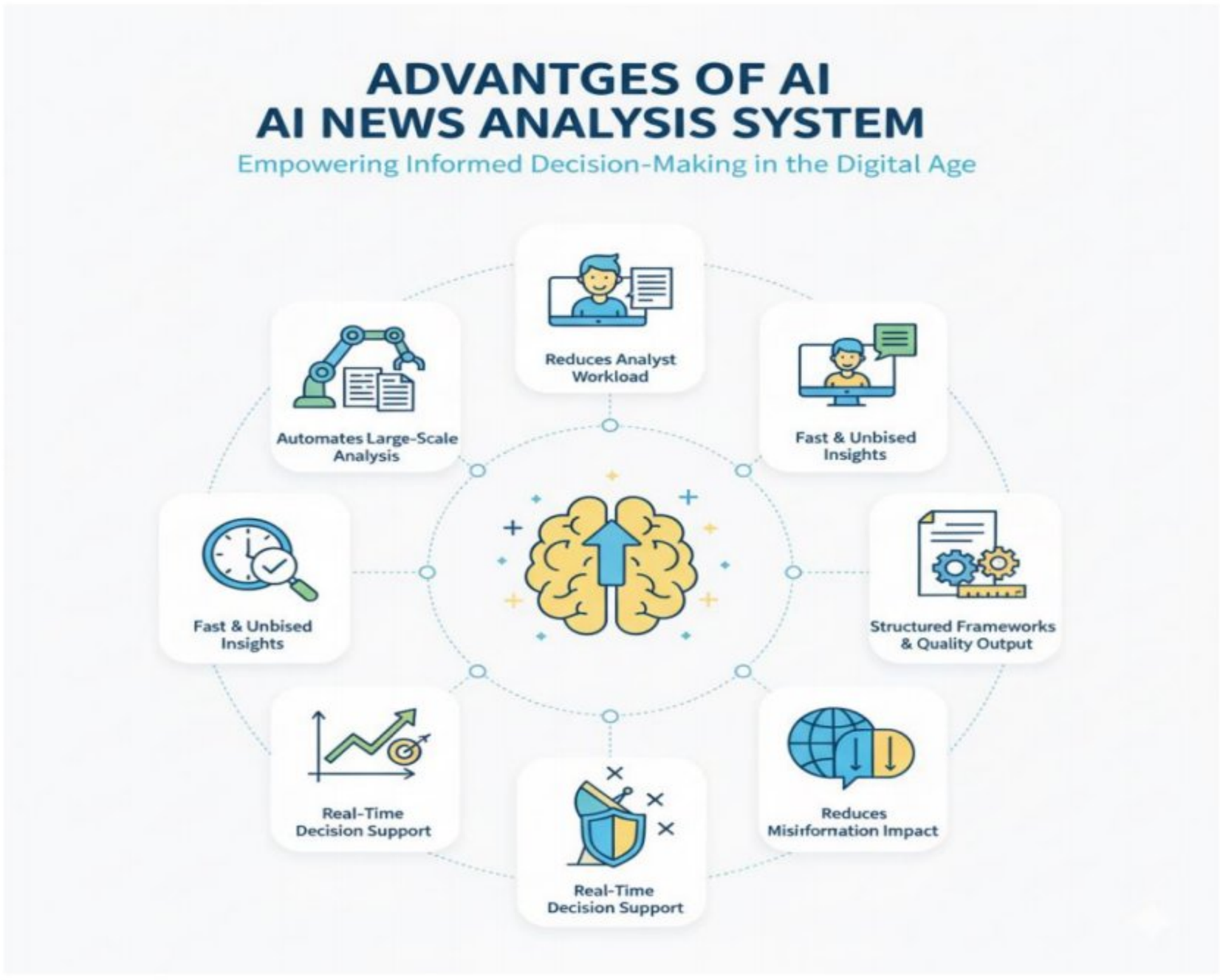
- Government policy units
- Research organisations
- Startups and corporate strategy teams
- Media verification teams
- Academic institutions

1.5 Architecture Diagram

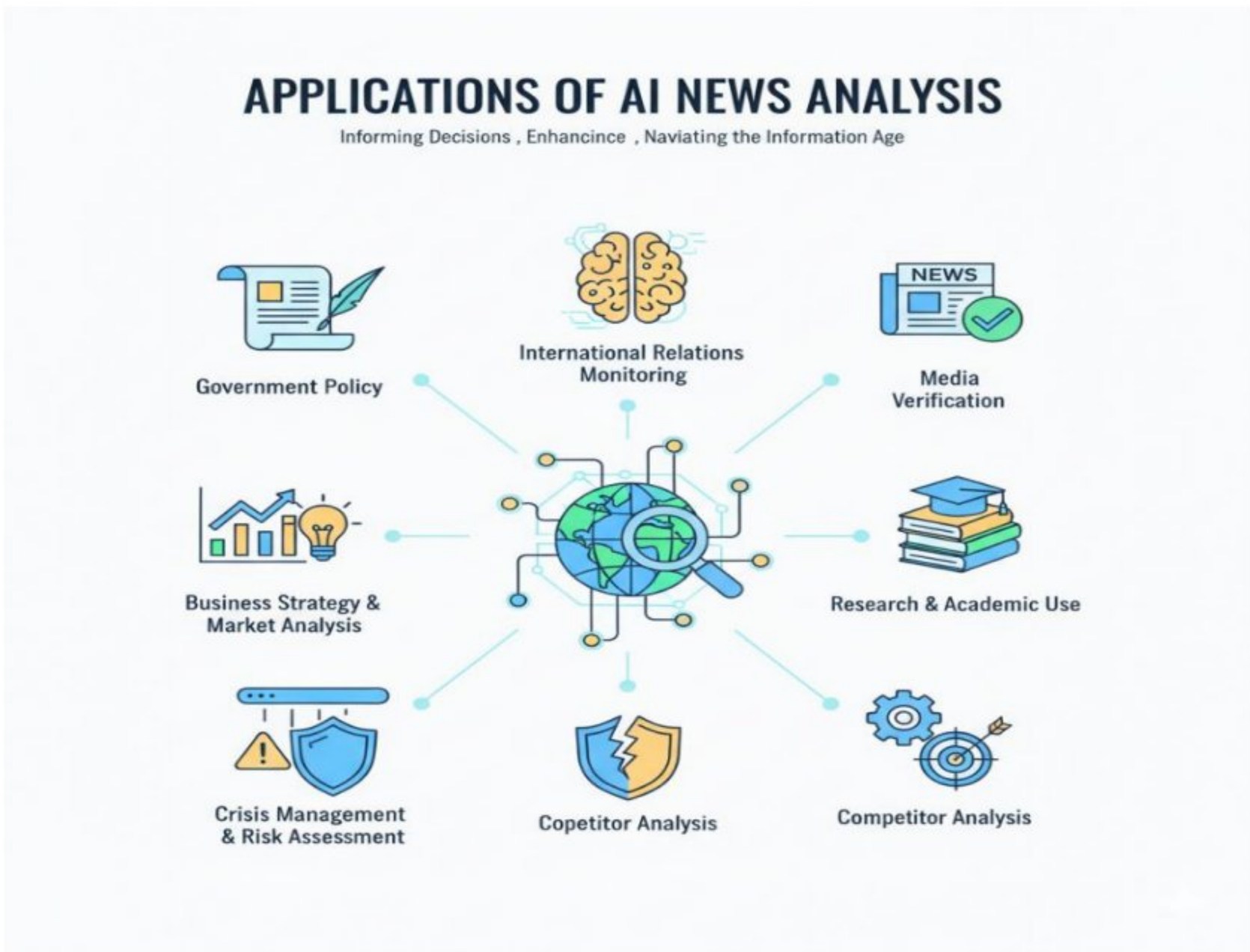
The architecture consists of the following components in sequence:



1.6 Advantages of the Proposed System



1.7 Applications of the Project



CHAPTER 2

2. LITERATURE SURVEY

Artificial Intelligence has advanced through the contributions of several researchers whose discoveries shaped modern **Agentic AI** systems. The base paper highlights how agent-based models, autonomous reasoning, and generative intelligence now drive smart, automated decision-making. Agentic AI builds upon these scientific foundations and enables systems like the **Autonomous Global News & Policy Analyst (AI Think Tank)** to function with high autonomy and analytical capability.

Early AI research by **John McCarthy (1956)** introduced the term *Artificial Intelligence* and emphasized machine reasoning, forming the foundation for autonomous agents. Later, **Allen Newell and Herbert A. Simon** developed the *General Problem Solver*, demonstrating that machines could make independent decisions—an early hint of agent-like behavior. The breakthrough driving modern AI came from **Yoshua Bengio, Geoffrey Hinton, and Yann LeCun**, who pioneered **Deep Learning**. Their work enabled neural networks to process large, unstructured data—essential for analyzing global news and generating insights. Their discoveries earned them the 2018 Turing Award and accelerated research in NLP and LLMs.

A major leap occurred with the **Transformer architecture**, introduced by **Vaswani et al. (2017)** in *"Attention is All You Need"*. The transformer enabled large language models that understand context, generate summaries, verify information, and perform multi-step reasoning—capabilities used in the Analyst and Strategy Agents of your project.Modern agent-based systems draw inspiration from the work of **Michael Wooldridge**, a leading researcher in multi-agent theory. His research defines how autonomous agents cooperate, communicate, and manage tasks—key principles behind your multi-agent pipeline (Retriever → Fact Checker → Analyst → Strategy → Visualizer).

In the domain of Generative AI, **Alec Radford** and the OpenAI team introduced the **GPT series**, showing that LLMs can perform reasoning, summarization, and text generation. These discoveries support your project's ability to generate strategy reports and policy recommendations.Similarly, **Nils Reimers and Iryna Gurevych (2019)** developed **Sentence-BERT**, enabling high-quality semantic embeddings and vector similarity search. This innovation powers the Retrieval-Augmented Generation (RAG) layer used in your Analyst Agent.

The base paper also discusses the emergence of **autonomous agent frameworks**, such as AutoGPT and AutoGen, which were influenced by research from **Microsoft**, **Stanford**, and **OpenAI** teams. These systems demonstrated how agents could plan tasks, break problems into subtasks, and work collaboratively—mirroring the architecture of your AI Think Tank.

Collectively, these scientific contributions show a clear progression:
Rule-based AI → Machine Learning → Deep Learning → LLMs → Agentic AI (Autonomous Multi-Agent Systems).

The literature consistently supports the shift from "copilot AI" (where humans assist AI) to "autopilot AI" (where AI independently manages workflows). This aligns directly with your project, where the system autonomously retrieves news, verifies facts, analyzes information, and produces strategic insights without human intervention.

Thus, the development of Agentic AI—supported by major scientific discoveries—provides a strong theoretical foundation for your system and validates its design approach.

CHAPTER 3

3. SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

3.1 Introduction to SRS

A Software Requirements Specification (SRS) is a structured document that defines the **complete behaviour, constraints, functionalities, and performance expectations** of a software system before it is developed. It acts as a formal foundation for understanding *what* the system must do, *how* it should function, and *under what conditions* it must operate.

In the context of the **Autonomous Global News & Policy Analyst (AI Think Tank)**, the SRS outlines the behaviour of a **multi-agent AI system** that retrieves global news, verifies authenticity, performs analytical reasoning, and generates strategic policy insights. The SRS provides a clear description of each agent, data pipelines, APIs, vector databases, and user interactions required for building the system.

It ensures that all stakeholders—including developers, project supervisors, and evaluators—share a common understanding of the system requirements.

3.2 Role of SRS

The SRS document plays the following key roles:

1. Serves as a Communication Bridge
It clearly communicates system expectations between the development team, faculty reviewers, and evaluators to avoid ambiguity.

2. Acts as a Blueprint for System Design
Architectural diagrams, module design, agent workflows, and database structures are derived from this document.

3. Foundation for Testing
Test cases, validation metrics, and expected outputs are aligned with requirements defined in the SRS.

4. Reduces Development Risks
Clearly defined constraints, dependencies, and system assumptions reduce errors and ensure smooth implementation.

5. Enhances System Reliability
A structured SRS ensures that the final product meets functional and non-functional expectations consistently.

3.3 Requirements Specification Document

This part of the SRS describes the overall structure of requirements for the system.

3.3.1 System Overview

The system consists of five main agents:

- Retriever Agent
- Fact-Checker Agent
- Analyst Agent
- Policy/Strategy Agent
- Visualizer Agent

Each agent collaborates to automate global news intelligence.

3.3.2 User Characteristics

Target users include:

- Policymakers
- Researchers
- Business strategists
- Academic institutions
- Media verification teams

Users require basic computer literacy but no advanced technical skills.

3.3.3 Assumptions

- News APIs provide stable responses
- Internet connectivity is available
- Vector database storage is accessible
- LLM/embedding models return valid outputs

3.3.4 Constraints

- API rate limits
- Potential downtime of news sources
- High computational cost for embeddings
- Ethical guidelines restricting political bias

3.3.5 Dependencies

- Python libraries
- External APIs
- Vector databases
- Cloud/Local storage

3.4 Functional Requirements

Functional requirements describe **what the system must do**.

3.4.1 Requirements for Retriever Agent

- FR1: System must fetch news from global APIs.
- FR2: System must extract metadata (country, category, timestamp).
- FR3: System must filter articles based on relevance.

3.4.2 Requirements for Fact-Checker Agent

- FR4: System must verify news using secondary sources.
- FR5: System must assign a credibility score.
- FR6: System must discard unverified or low-credibility articles.

3.4.3 Requirements for Analyst Agent

- FR7: System must retrieve relevant context using RAG.
- FR8: System must generate SWOT analysis.
- FR9: System must generate PESTEL analysis.
- FR10: System must identify industry/business impact.
-

3.4.4 Requirements for Policy/Strategy Agent

- FR11: System must generate strategic recommendations.
- FR12: System must produce policy suggestions for governments & businesses.
-

3.4.5 Requirements for Visualizer Agent

- FR13: System must generate dashboards and charts.
- FR14: System must summarize insights in structured format.

3.5 Non-Functional Requirements

Non-functional requirements describe **how the system must behave**.

3.5.1 Scalability
- System should handle increasing volume of news articles.

3.5.2 Reliability
- All agents must operate without failure across extended sessions.

3.5.3 Security
- API keys and sensitive data must be encrypted.

3.5.4 Usability
- Output dashboards must be simple and understandable.

3.5.5 Maintainability
- Code must follow modular structure with separate agent files.

3.5.6 Accuracy
- Fact-checker must achieve high correctness in cross-verification.

3.6 Performance Requirements

Performance requirements define **measurable expectations**.

3.6.1 Response Time
- Each agent must respond within 2–5 seconds.
- RAG retrieval latency must be under 1 second.

3.6.2 Throughput
- System should process at least 50–100 articles per minute.

3.6.3 Load Handling

- System must operate efficiently even with large data sets.

3.6.4 Stability

- System should run continuously for 2–3 hours without failure.

3.7 Software Requirements

3.7.1 Operating System

- Windows / Linux / macOS

3.7.2 Programming Languages

- Python 3.10+

3.7.3 Required Python Libraries

- Transformers
- Sentence-Transformers
- LangChain / LlamaIndex
- FAISS / ChromaDB
- Requests
- Matplotlib / Plotly
- Streamlit / FastAPI (optional UI)

3.7.4 External APIs

- NewsAPI / GDELT
- Wikipedia API
- Fact-checking API (Google Fact Check Tools)

3.7.5 Development Tools

- Jupyter Notebook
- VSCode
- Git / GitHub

3.8 Hardware Requirements

Minimum Hardware Requirements

- Processor: Intel i5 (or equivalent)
- RAM: 8 GB
- Storage: 256 GB SSD
- Internet: Stable broadband connection

Recommended Hardware Requirements

- Processor: Intel i7 or AMD Ryzen 5+
- RAM: 16 GB
- Storage: 512 GB SSD
- GPU (optional): NVIDIA 1650 or higher (for faster embeddings)
- High-speed internet for API calls

Additional Hardware (Optional)

- External storage for large datasets
- GPU-enabled workstation for accelerated AI tasks

CHAPTER – 4

4. SYSTEM DESIGN

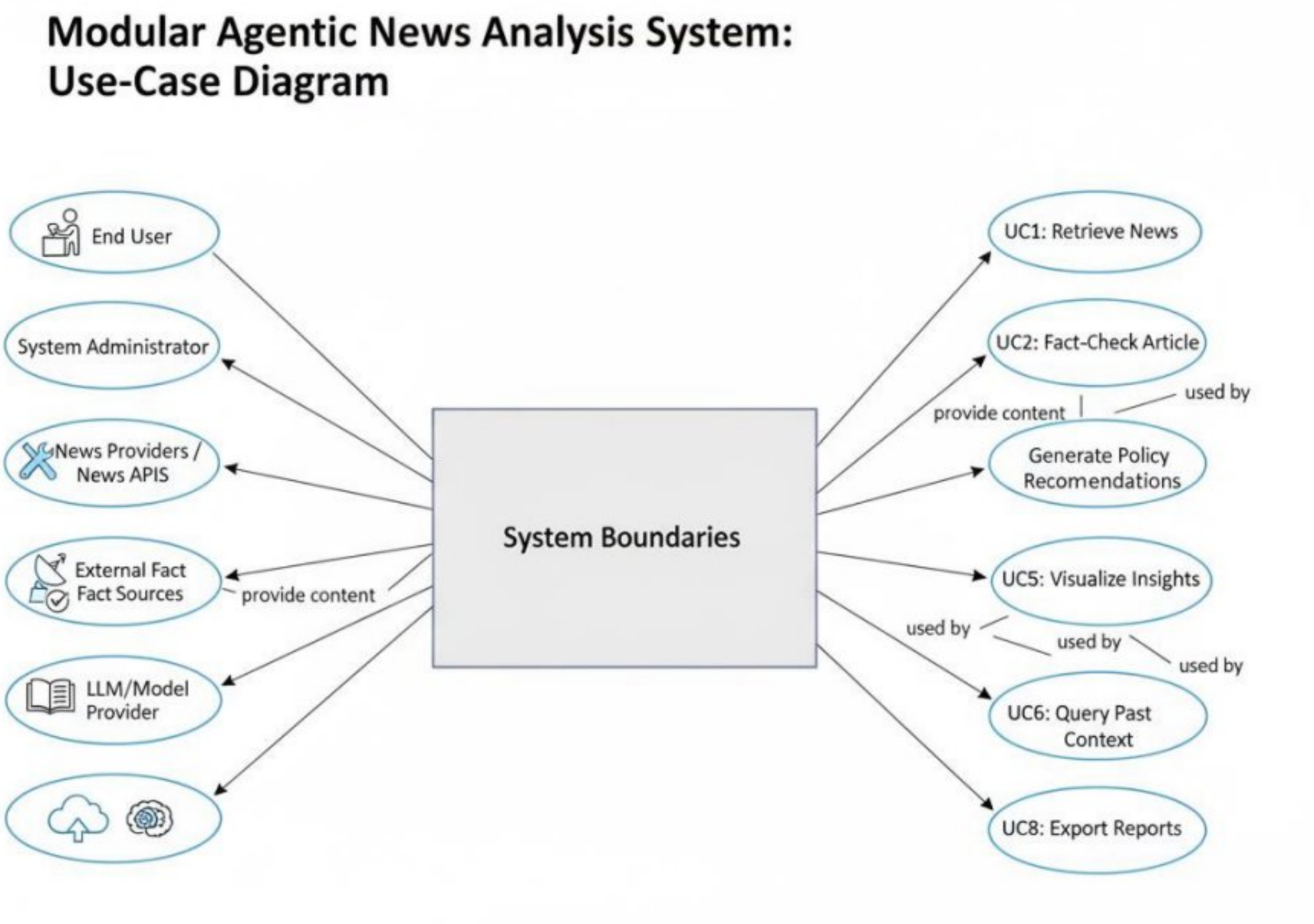
4.1 Introduction to UML

This chapter defines the high-level architecture and design of the multi-agent AI Think Tank. The system is modular and agent-based: Retriever, Fact-Checker, Analyst, Policy/Strategy, Visualizer. It uses a Vector DB for RAG, LLMs/embeddings for reasoning, and a dashboard/UI for users. The UML diagrams below make the structure, behavior, and deployment explicit.

4.2 UML Diagrams

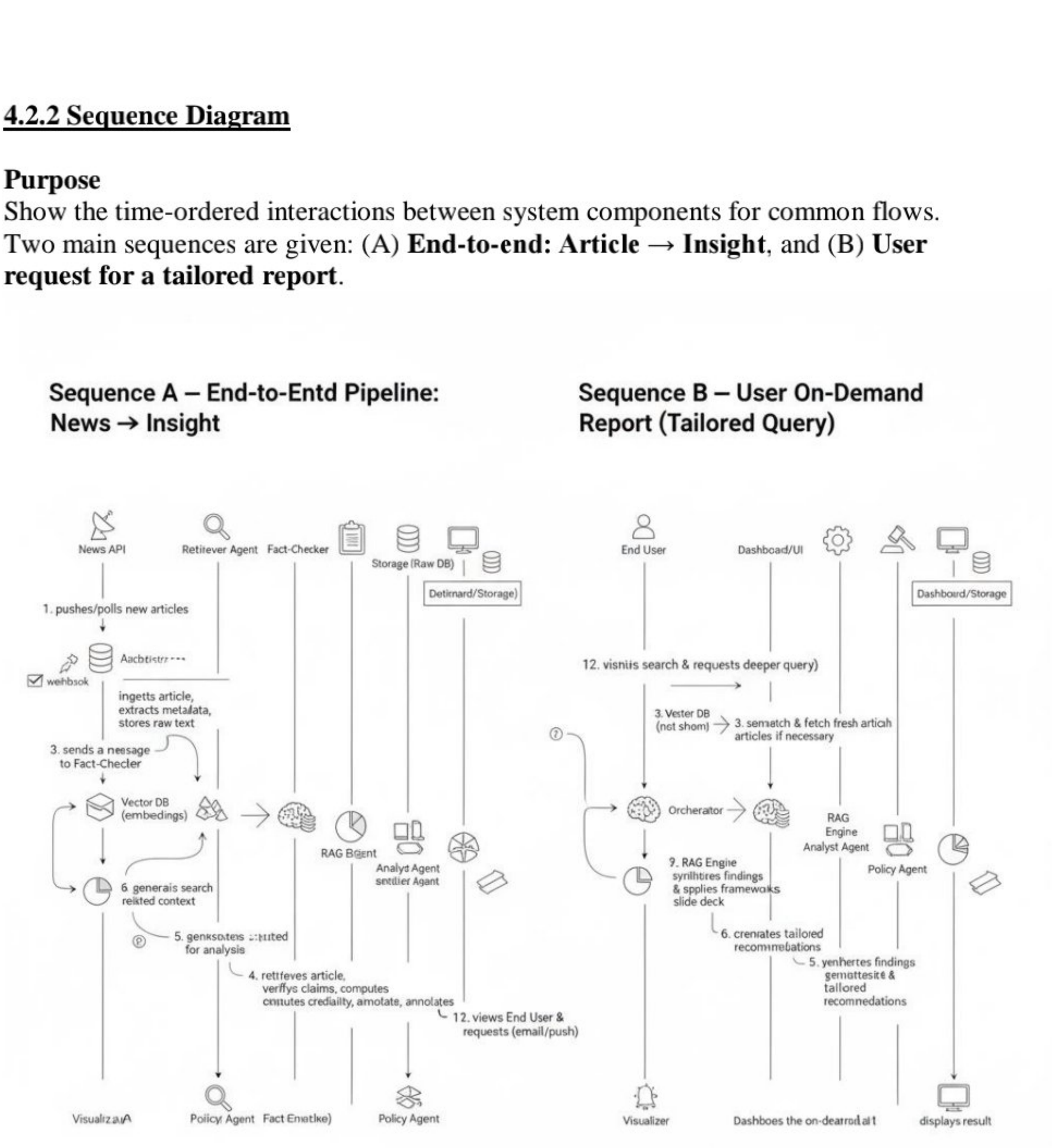
4.2.1 Use-Case Diagram

Purpose
Show interactions between external actors (users, external APIs) and the system's high-level capabilities.



4.2.2 Sequence Diagram

Purpose
Show the time-ordered interactions between system components for common flows. Two main sequences are given: (A) **End-to-end: Article → Insight**, and (B) **User request for a tailored report**.



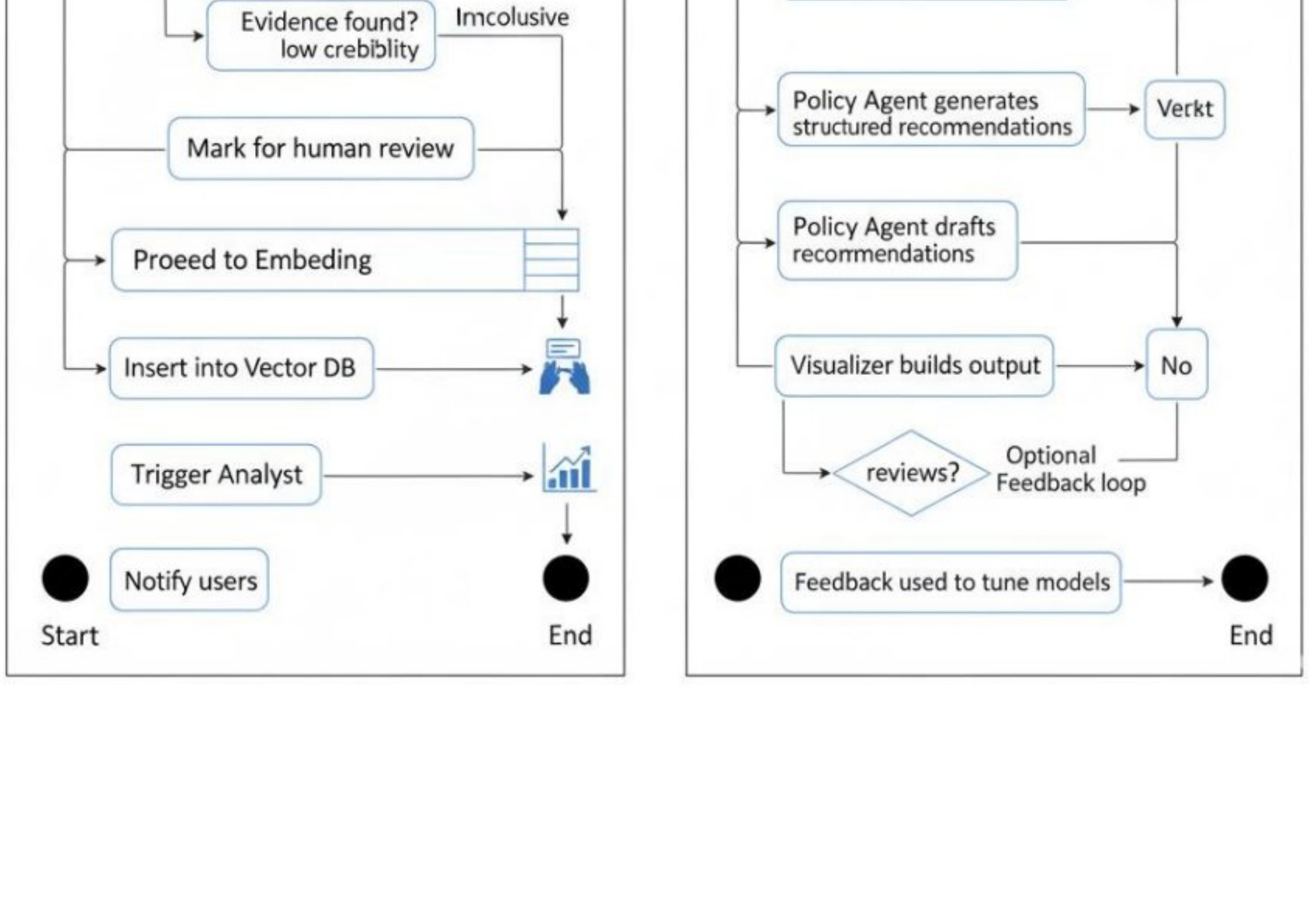
Key Interactions:
The modular approach ensures scalability, fault tolerance, and clarity

- News API provides raw data.
- End User initializes report request.

4.2.3 Activity Diagram

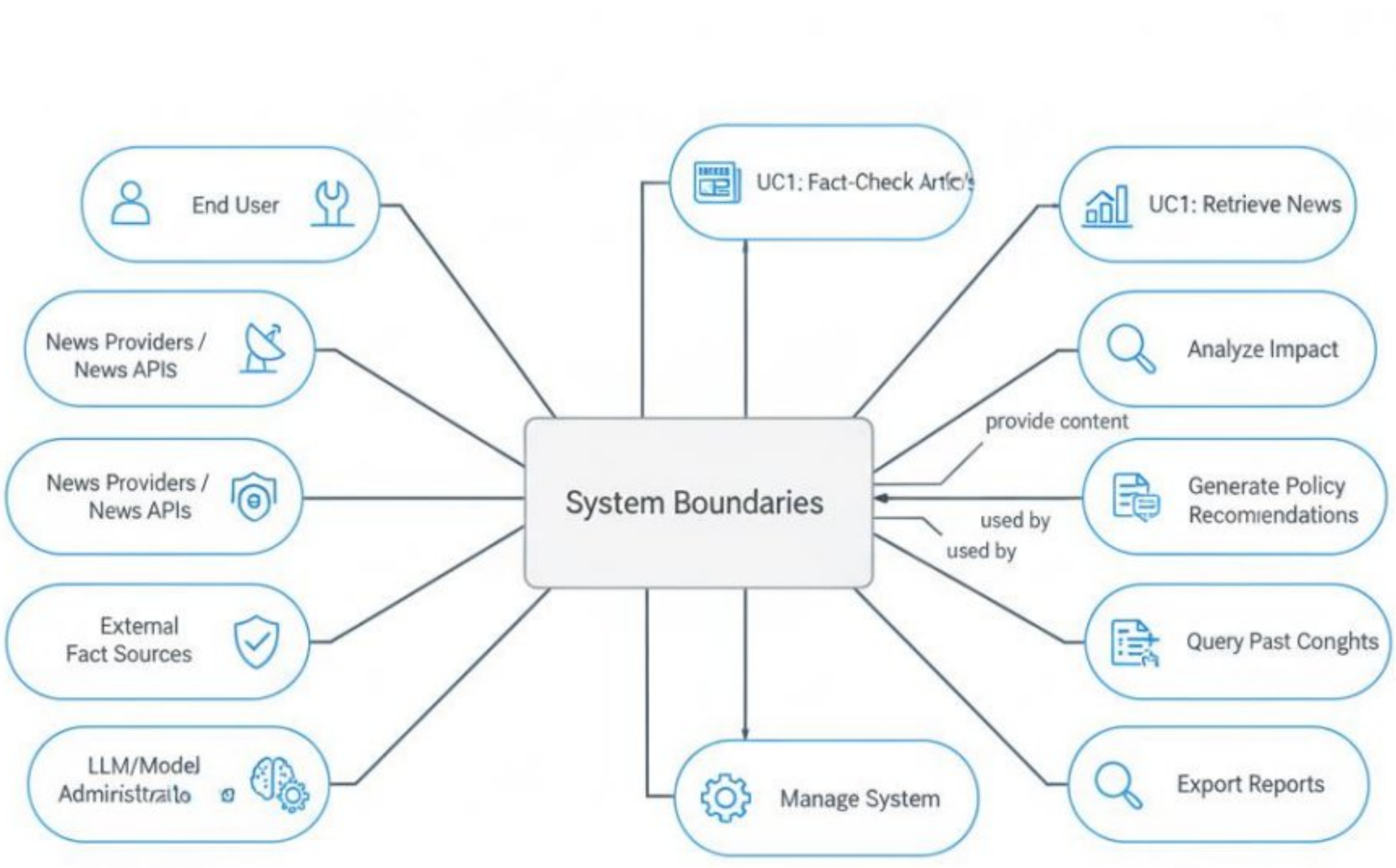
Purpose

Show the internal workflow steps and decision points (control flow) for a critical activity: **Fact-Check → Analyze → Publish**.



4.2.4 Deployment Diagram

Purpose
Describe physical arrangement of software components and hardware resources. It clarifies which components run on which servers/nodes and how they communicate.



Texual Use-Case Map

- Enst: Renieve New
- End User
- External for
- LLM/Model Provider

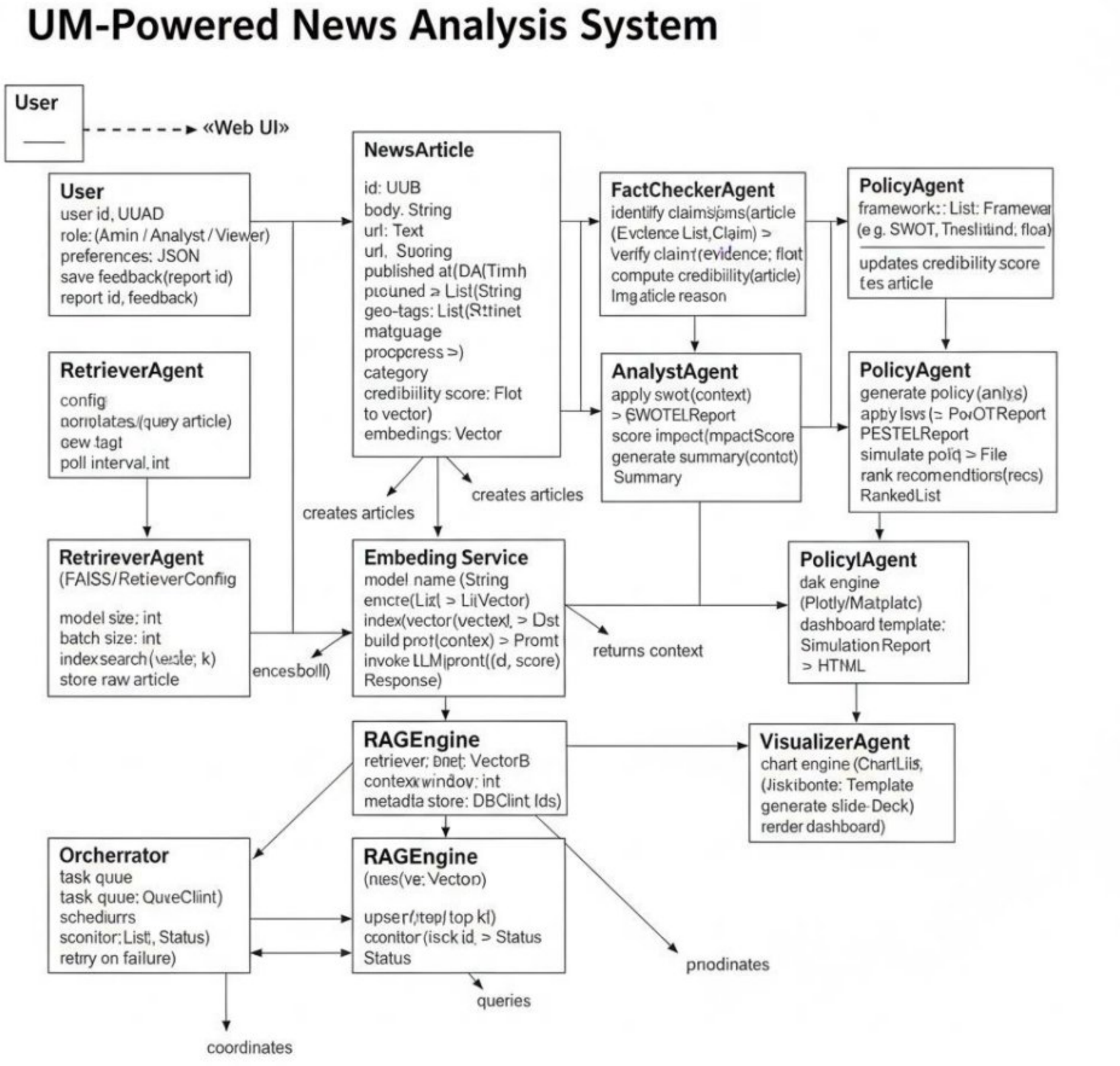
Key Interactions:

- U1, 33, 95
- U1, 33, 95
- Vissualize Insights
- Query Past Contexts

The modular approach ensures scalability, fault tolerance, and clarity.

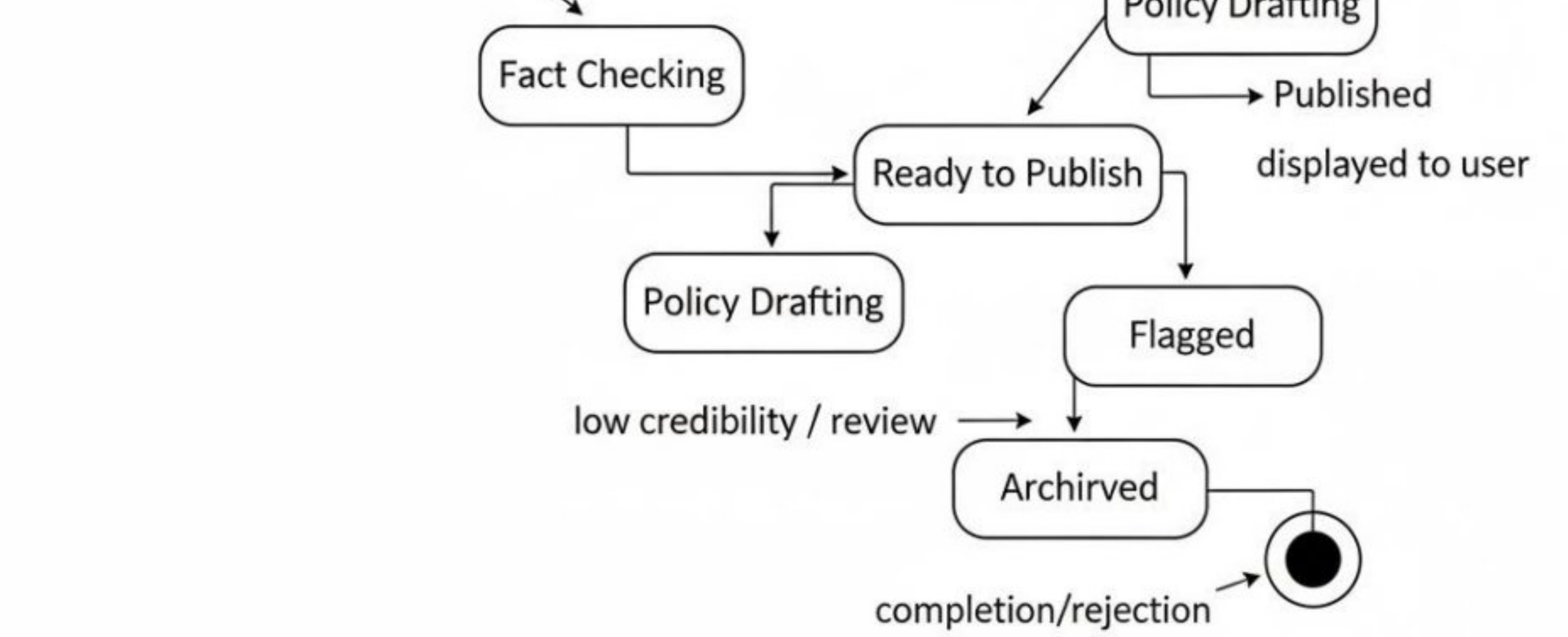
4.2.5 Class Diagram

Purpose
Specify core classes, attributes, and methods that map to modules and components. This supports implementation and unit testing.
Below is a set of suggested classes with attributes and primary methods. You can convert this to a UML class diagram.



4.2.6State Chart Diagram

Purpose
A **State Chart Diagram** represents the various states an entity (in this case, the *news article* or *agent task*) goes through during the system's operation.



4.3 Technologies Used

1. Python
Primary programming language used for implementing multi-agent workflows, API integration, data processing, and model orchestration.

2. Large Language Models (LLMs)
Used for summarization, reasoning, fact-checking assistance, SWOT/PESTEL analysis, and generating policy recommendations.

3. Hugging Face Transformers
Provides pre-trained models for:

- Text embeddings
- NER (Named Entity Recognition)
- Summarization and classification

4. Sentence-Transformers / Embedding Models
Used for generating high-quality semantic embeddings required for vector search and RAG.

5. LangChain / LlamaIndex
Framework for building RAG pipelines, managing prompts, and orchestrating LLM-based agents.

6. Vector Database (FAISS / ChromaDB)
Stores embeddings and supports semantic search for retrieving context during analysis.

7. News APIs (NewsAPI, GDELT, RSS Feeds)
Used by the Retriever Agent to collect real-time global news articles.

8. Wikipedia & Fact-Check APIs
Used by the Fact-Checker Agent to cross-verify claims and compute credibility scores.

9. Matplotlib / Plotly
Visualization libraries used by the Visualizer Agent to create charts, trend graphs, and dashboards.

10. Streamlit / FastAPI (Optional UI Layer)
Used for interactive dashboards, real-time visualization, and user query interfaces.

CHAPTER – 5

5. IMPLEMENTATION

This chapter explains the detailed implementation of each module in the AI Think Tank system. The system is built using a modular, multi-agent architecture where each agent performs a specific task such as retrieving news, fact-checking, analyzing information, generating strategies, and visualizing final insights.

5.1 Setting Up the Development Environment

The development environment was configured to support large-scale text processing, vector embeddings, and API integration. The steps include:

1. Installing Python and Required Packages

Python was set up along with all necessary libraries:

- Transformers (HuggingFace)
- Sentence-Transformers
- LangChain / LlamaIndex
- FAISS / ChromaDB
- Requests
- Matplotlib / Plotly
- Pandas / NumPy

2. Setting Up API Keys

Environment variables were created for:

- NewsAPI
- GDELT
- Wikipedia API
- Fact-check API
- LLM model keys

These keys were stored securely to prevent unauthorized access.

3. Configuring Vector Database

A FAISS or ChromaDB instance was initialized to store embeddings.
Index dimensions were chosen based on the embedding model used

4. Directory Structure

```
/agents
    /retriever
    /fact_checker
    /analyst
    /policy_agent
    /visualizer
/data
/models
```

/dashboard
This modular structure helped in managing each agent independently.

5.2 Implementing the Retriever Agent

The Retriever Agent is responsible for fetching global news from APIs.

Core Tasks

1. **Connecting to News APIs**
   Using HTTP requests to NewsAPI, GDELT, or RSS feeds.
2. **Fetching Articles Based on Queries**
   Parameters include:
   - keyword
   - country
   - category
   - date range
3. **Extracting Metadata**
   The agent extracts:
   - title
   - description
   - timestamp
   - URL
   - source
   - region
4. **Preprocessing Raw Articles**
   - Removing HTML tags
   - Lowercasing
   - Removing special characters
   - Detecting language
5. **Storing Articles in Local/Cloud Storage**
   Stored in JSON format with unique IDs for further processing.

Output
Clean, structured articles ready for fact-checking and embedding.

5.3 Implementing the Fact-Checker Agent

The Fact-Checker Agent ensures the authenticity of news articles.
Core Functionalities

1. **Claim Extraction**
   Identifies statements in the article that need verification.
2. **Cross-Verification**
   Uses:
   - Wikipedia search
   - Fact-check databases
   - Secondary news sources
3. **Credibility Scoring**
   Generates a score (0 to 1) based on:
   - Number of supporting sources
   - Conflicting information

- Article reliability parameters

4. **Filtering Low-Trust Articles**
   Articles below a threshold score (e.g., 0.5) are flagged or discarded.
5. **Logging Verification Results**
   Each claim and its evidence are logged for transparency.

Output
Verified articles with credibility scores attached.

5.4 Implementing the Analyst Agent

The Analyst Agent performs the core reasoning and analysis tasks.
Major Functionalities

1. **Connecting to Vector Database**
   Retrieves relevant context using semantic search (RAG).
2. **Embedding Article Content**
   Converts text into high-dimensional vectors for indexing.
3. **Applying Frameworks**
   - **SWOT Analysis**
     Identifies strengths, weaknesses, opportunities, and threats.
   - **PESTEL Analysis**
     Examines Political, Economic, Social, Technological, Environmental, and Legal impacts.
   - **Porter's Five Forces**
     Evaluates industry competitiveness.
4. **Impact Scoring**
   Quantifies:
   - impact magnitude
   - urgency
   - risk level
5. **Generating Summaries**
   Produces structured insights for further policy analysis.

Output
Detailed analytical report for each verified article.

5.5 Implementing the Policy/Strategy Agent

This agent generates high-level recommendations for businesses and governments.
Core Functions

1. **Reading Analytical Report**
   Input from the Analyst Agent is interpreted for policy relevance.
2. **Generating Policy Recommendations**
   For governments:
   - regulatory suggestions
   - mitigation strategies
   - international response guidance

For businesses:

- market strategy
- operational improvements

- competitive positioning

3. **Scenario Simulation**
   Evaluates possible outcomes:
   - short-term
   - medium-term
   - long-term
4. **Prioritization of Actions**
   Ranks actions based on impact and feasibility.
5. **Drafting Strategy Reports**
   Produces human-readable recommendations using LLMs.

Output
Complete strategic and policy recommendations.

5.6 Implementing the Visualizer & Dashboard

The Visualizer Agent converts analysis into graphical insights.
Visualization Tools Used

- **Matplotlib**
- **Plotly**
- **Streamlit / FastAPI** (for dashboard)

Generated Components

1. **Trend Charts**
   - event timelines
   - impact graphs
2. **Pie/Bar Charts**
   - category-wise news distribution
   - regional impact metrics
3. **Heatmaps**
   - country-wise risk levels
4. **Summary Cards**
   - quick insights
   - recommended actions
5. **Downloadable Reports**
   - PDF summaries
   - slide-like HTML reports

Dashboard Features

- Search bar
- Filter by category, region, timeline
- View credibility scores
- Deep-dive analysis view

Output
A clean, interactive dashboard for decision makers.

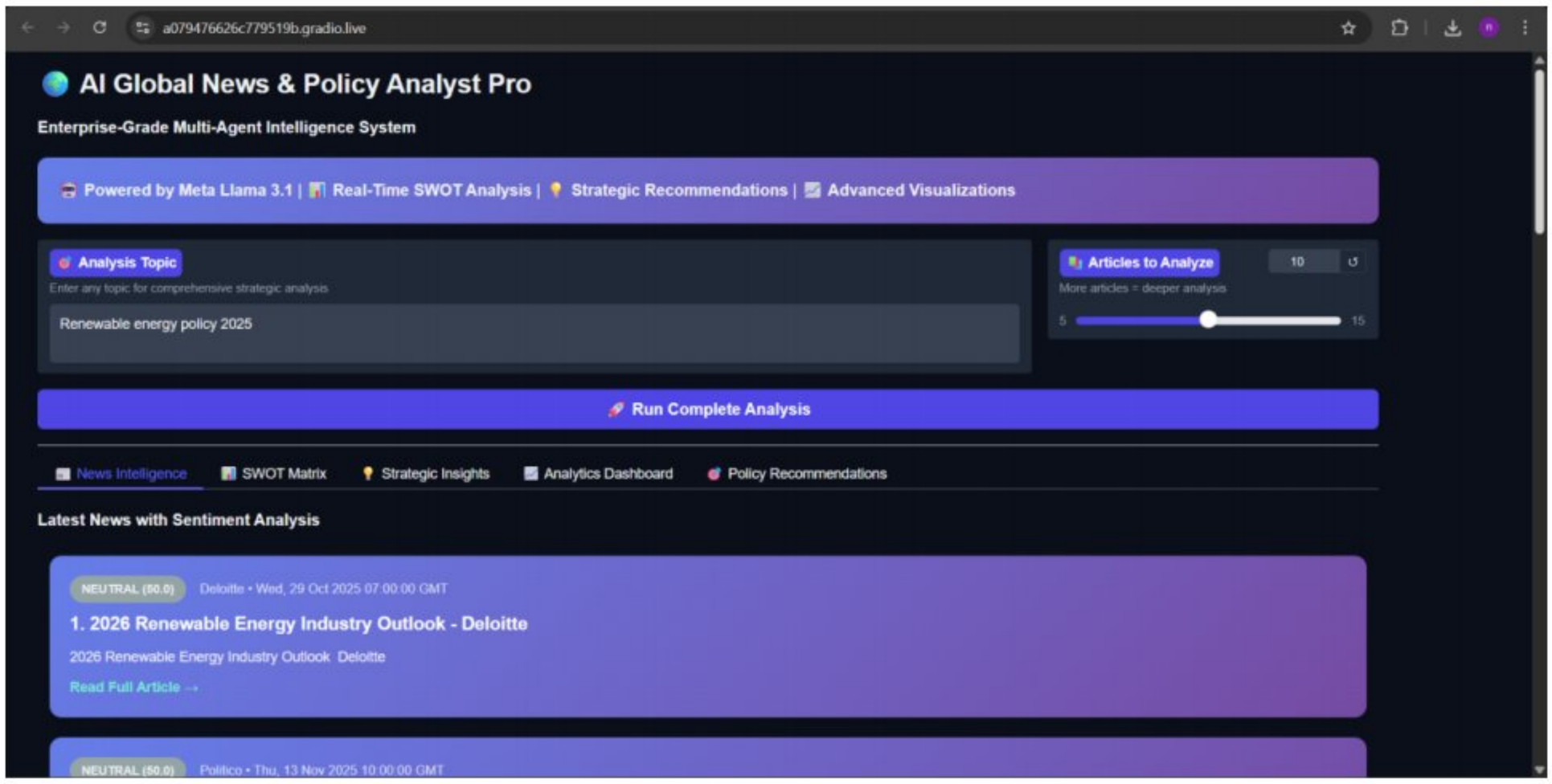
5.7 System Workflow Execution

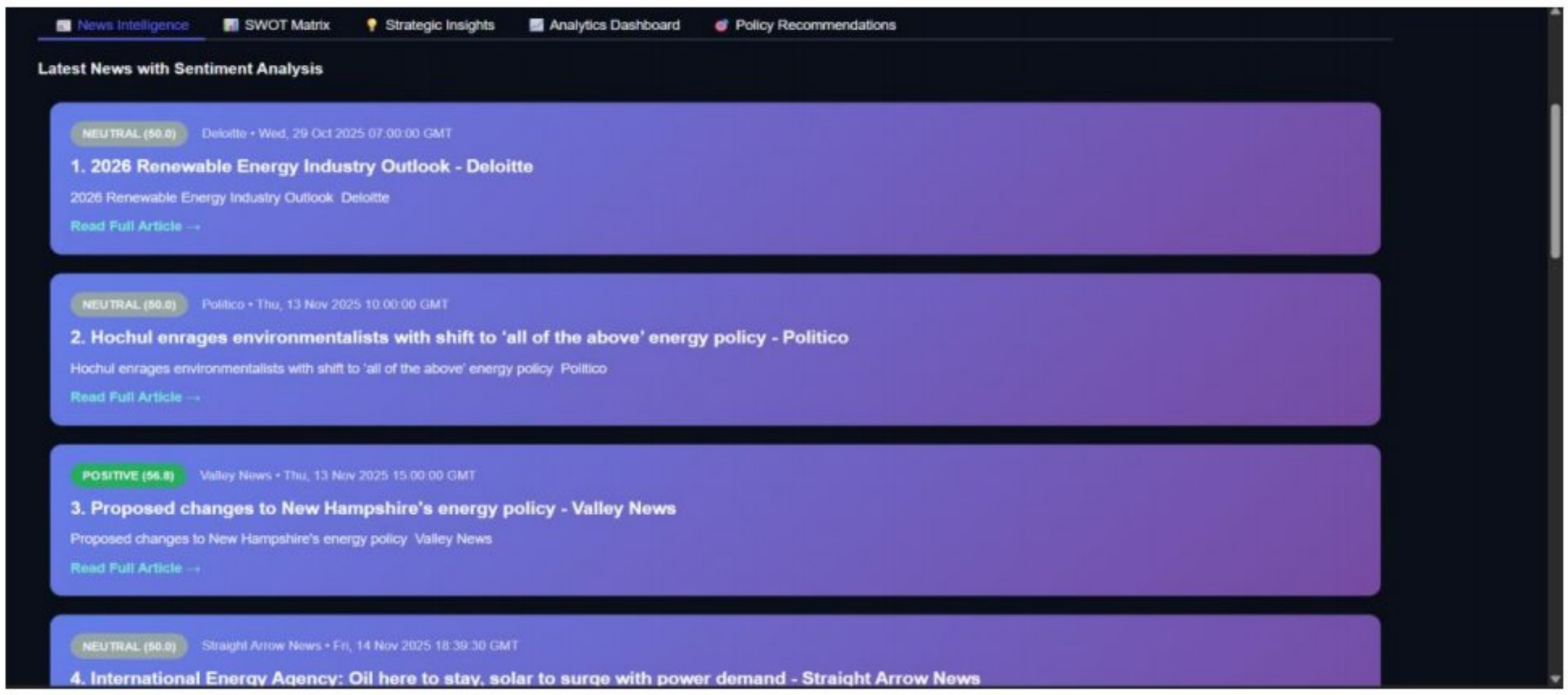
This describes the complete flow of data through different agents.

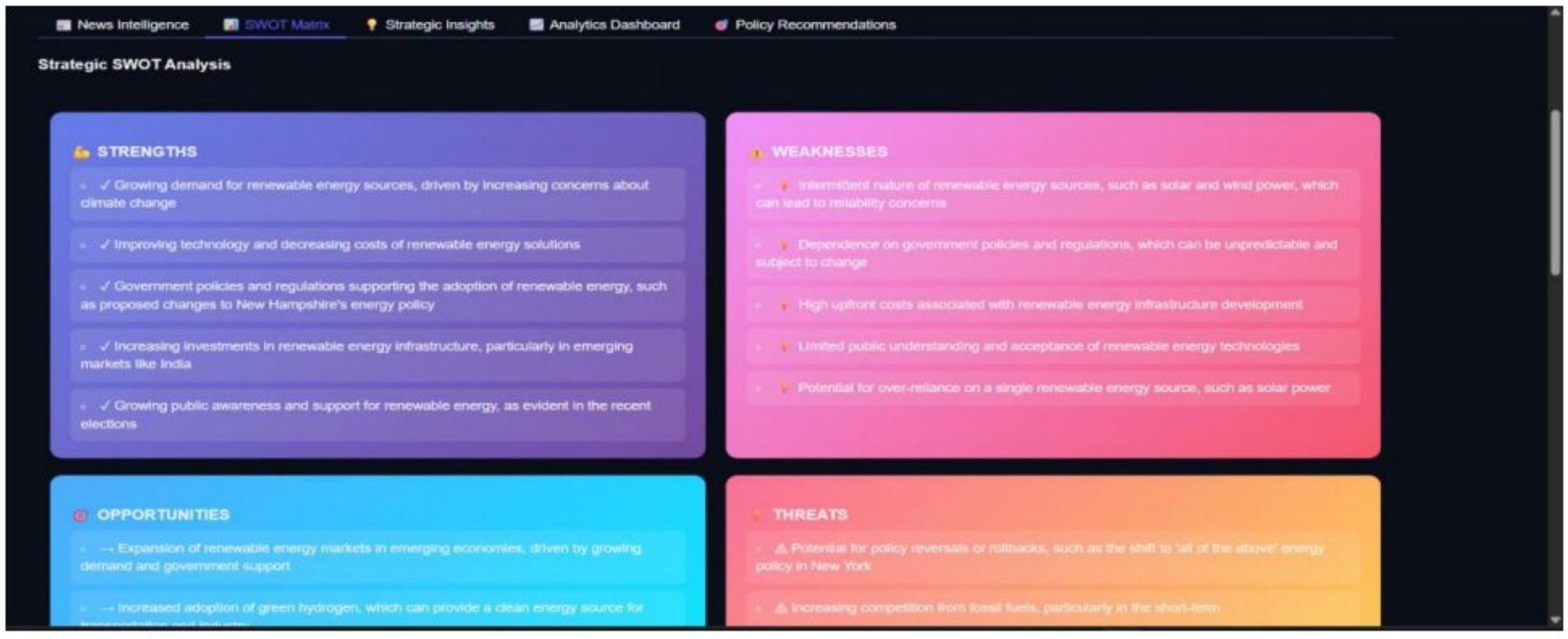
Step-by-Step Execution

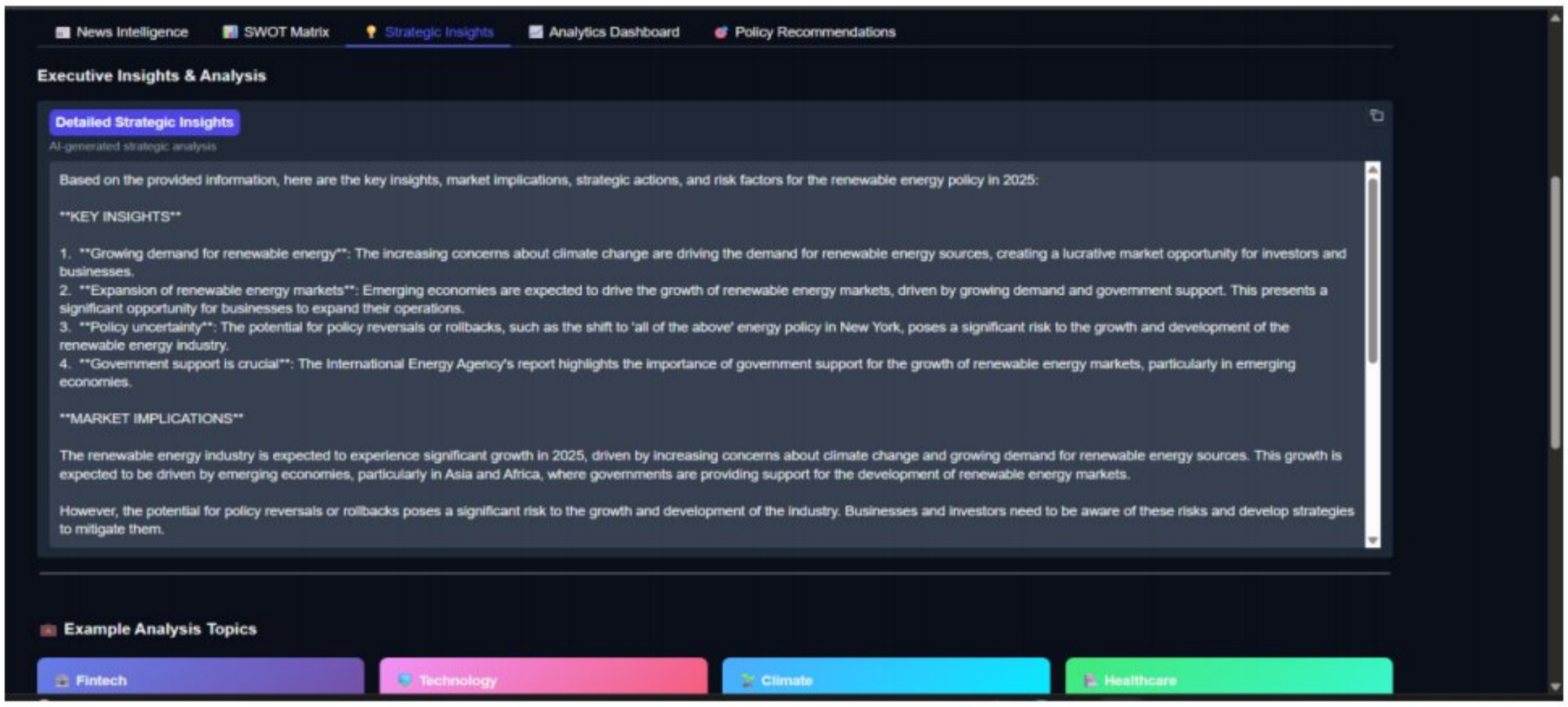
1. **Article Retrieval**
   Retriever Agent fetches new articles.
2. **Fact Verification**
   Fact-Checker Agent verifies claims and assigns credibility scores.
3. **Context Embedding & Storage**
   Articles are converted into embeddings and stored in vector DB.
4. **Context Retrieval**
   Analyst Agent retrieves similar context using semantic search.
5. **Analysis**
   Analytical frameworks are applied to generate structured insights.
6. **Policy & Strategy Generation**
   Policy Agent creates actionable recommendations.
7. **Visualization**
   Visualizer prepares dashboards, charts, and reports.
8. **User Access**
   Final results are shown on a dashboard for business/policy decisions.

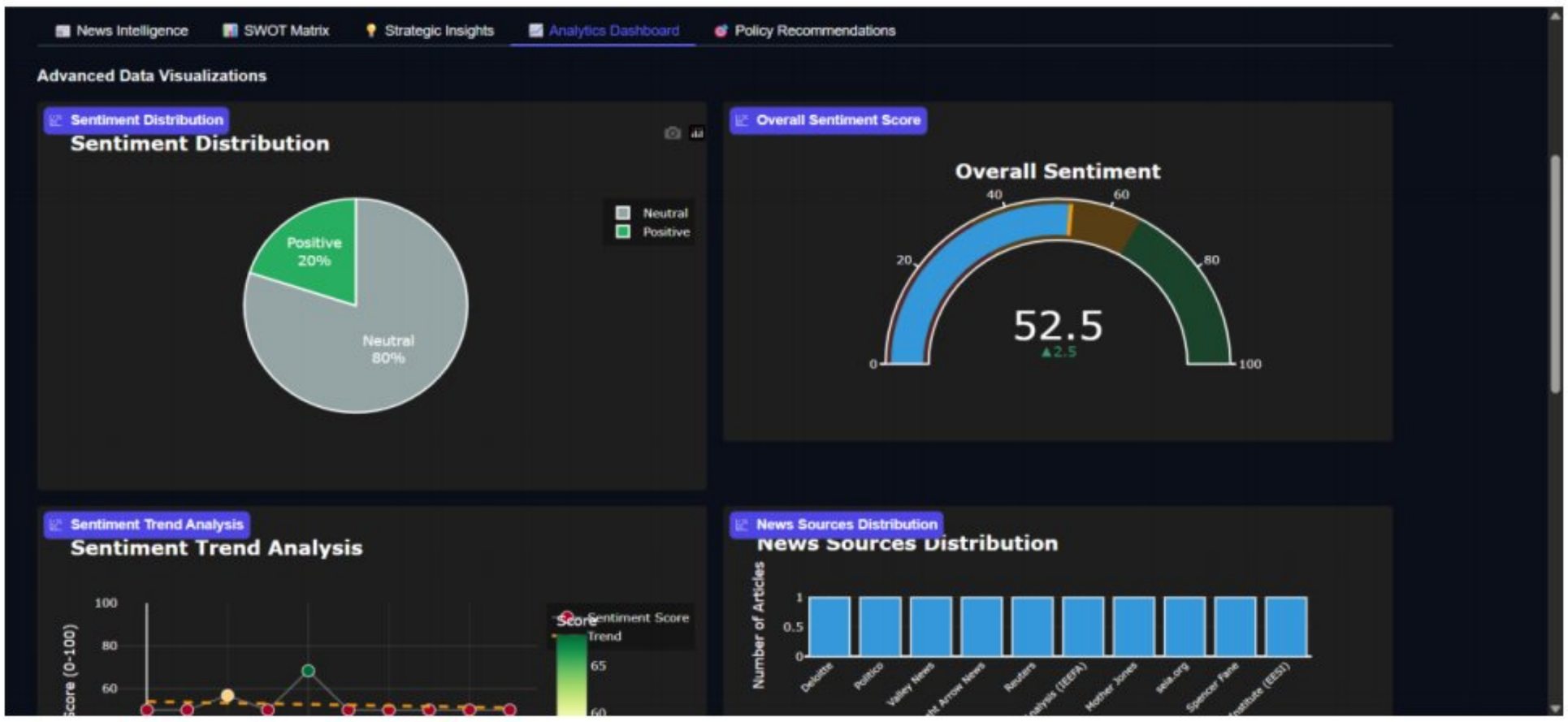
5.8 Screenshots

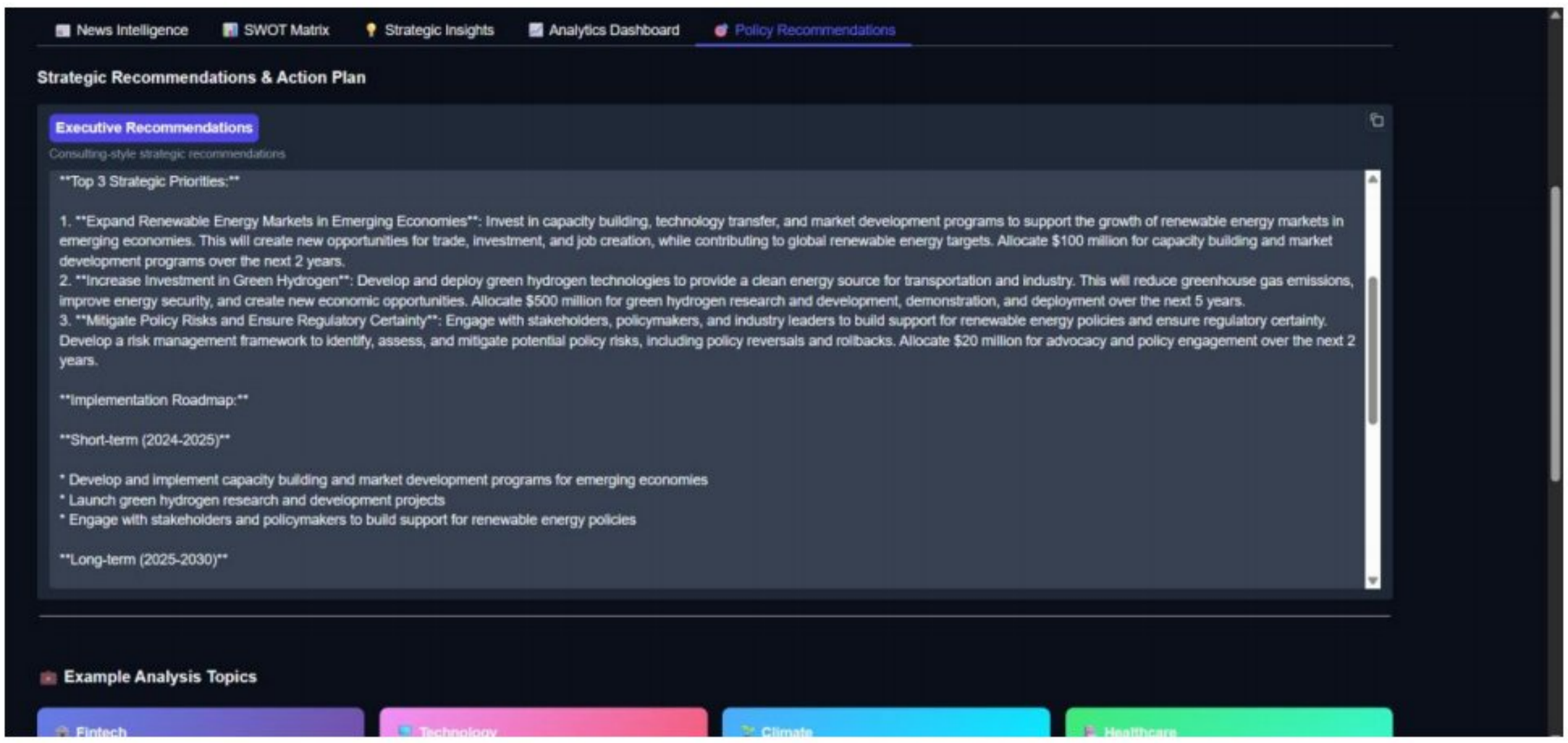


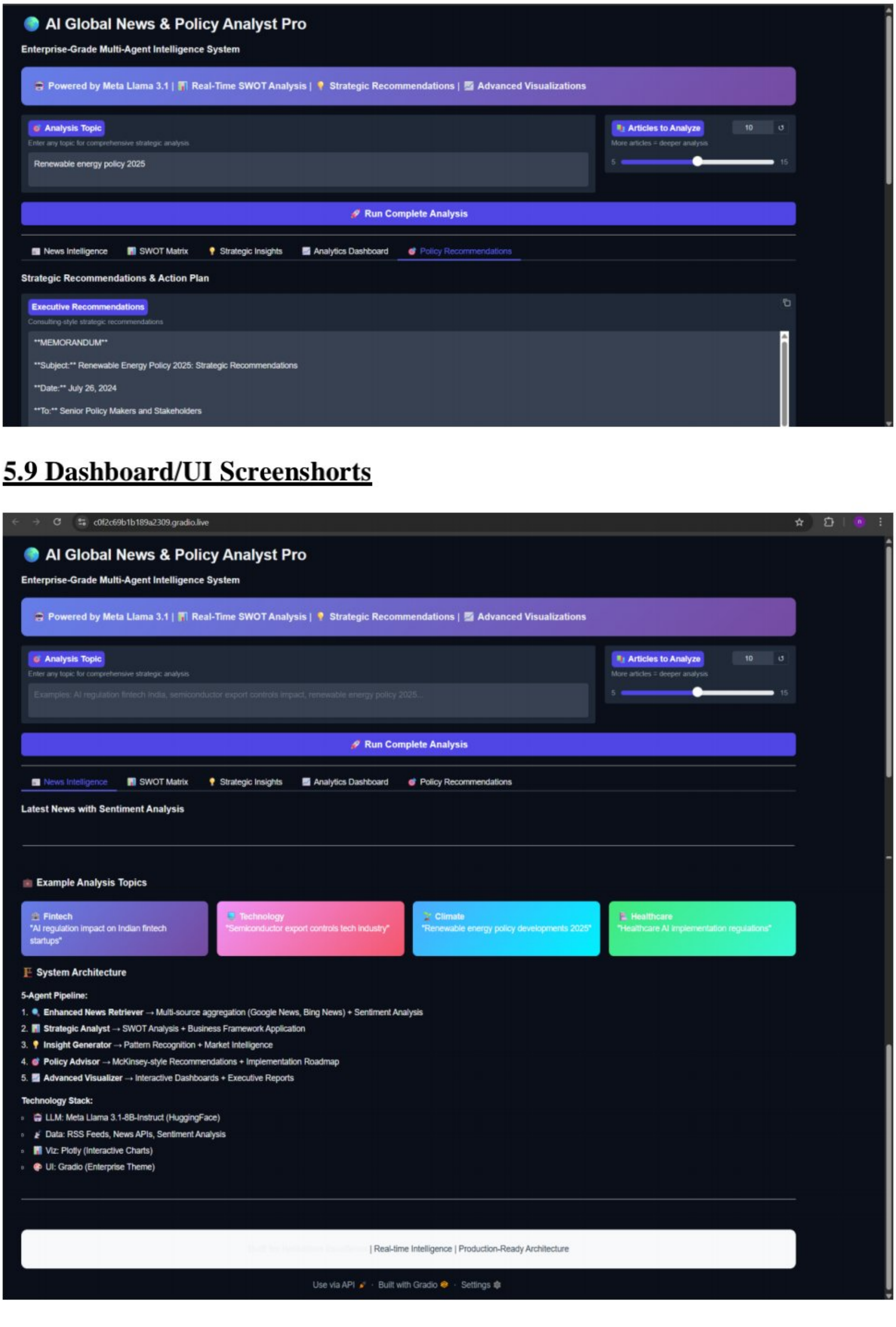












5.9 Dashboard/UI Screenshorts

CHAPTER – 6

6. SOFTWARE TESTING

6.1 Introduction

Software Testing is an essential phase of the development lifecycle that ensures the system behaves according to the specified requirements. For the **AI Think Tank** project, testing is particularly important because the system interacts with external APIs, uses multiple AI agents, and produces insights that must be accurate, consistent, and reliable.

Testing focuses on:

- correctness of news retrieval
- accuracy of fact verification
- validity of analytical outputs
- effectiveness of policy recommendations
- usability and stability of the dashboard

6.1.1 Testing Objectives

The primary objectives of testing this system are:

- To verify that each agent (Retriever, Fact-Checker, Analyst, Strategy, Visualizer) performs as expected.
- To ensure the system retrieves relevant and accurate real-time data.
- To validate credibility scores generated by the Fact-Checker Agent.
- To confirm that analytical frameworks (SWOT, PESTEL, Porter's Forces) produce meaningful outputs.
- To ensure the dashboard visualizes insights correctly and without errors.
- To detect bugs, failures, and inconsistencies in the multi-agent workflow.
- To ensure the system is reliable, stable, and scalable.

6.1.2 Testing Strategies

The following testing strategies were applied:

1. Unit Testing
Each agent was tested independently:

- Retriever Agent functions
- Fact-checking modules
- RAG retrieval
- SWOT/PESTEL functions
- Policy generation functions

2. Integration Testing
Ensures smooth communication between:

- Retriever → Fact-Checker
- Fact-Checker → Analyst
- Analyst → Strategy Agent
- Strategy Agent → Visualizer

3. System Testing
The complete multi-agent pipeline was executed end-to-end to validate:

- Data flow
- State transitions
- Full insight generation

4. Performance Testing
Measured:

- Article processing speed
- Embedding generation time
- Vector search latency
- Dashboard load time

5. Usability Testing
Evaluated:

- Dashboard navigation
- UI clarity
- Response times
- User interactions

6.1.3 System Evaluation

The system was evaluated across different dimensions:

1. Accuracy

- Fact-checking was compared with verified sources.
- Strategy recommendations were cross-verified with real events.

2. Reliability

- Tested with continuous news ingestion over several hours.
- Ensured no agent failures or crashes.

3. Consistency

- Same input produced consistent analytical outputs.
- RAG produced stable context retrieval across runs.

4. Performance

- Processing time was measured and optimized.
- Visualizer rendered charts without lag.

5. Scalability

- System performance checked with increasing number of articles.

6.1.4 Testing the New System

The newly developed system was tested through the following steps:

1. Real News Testing
Live articles were fetched and verified for:

- relevance

- metadata correctness
- credibility

2. Cross-Verification Checks
Fact-Checker was tested using:

- Wikipedia
- Google Fact Check JSON responses
- Secondary articles

3. Analytical Consistency
Analyst outputs were compared with manually written SWOT/PESTEL analyses.

4. Report Accuracy
Strategy recommendations were reviewed for clarity and logic.

5. UI & Dashboard Testing
The dashboard was tested for:

- navigation flow
- correct chart rendering
- filter functionality

6. Error Handling
Tested edge cases:

- API failures
- missing fields
- unsupported languages
- empty articles

6.2 Test Cases

Below are categorized test cases for each module in the system.

6.2.1 Retriever Agent Test Cases

Test Case ID	Description	Input	Expected Output
R-01	Fetch news successfully	Valid API key	List of articles
R-02	Invalid API key	Bad key	Error message
R-03	Country filter test	"IN"	Only Indian news
R-04	Network failure	No internet	Retry or error
R-05	Empty keyword	""	Default top headlines

6.2.2 Fact-Checker Agent Test Cases

Test Case ID	Description	Input	Expected Output
F-01	Claim extraction	Article with facts	List of claims
F-02	Fact verification	Claim text	Credibility score
F-03	Conflicting sources	Conflicting claims	Low score
F-04	No claims	Normal article	Skip fact-check
F-05	Fact source unavailable	API down	Graceful fallback

6.2.3 Analyst Agent Test Cases

Test Case ID	Description	Input	Expected Output
A-01	RAG context retrieval	Query text	Relevant documents
A-02	SWOT generation	Article	SWOT output
A-03	PESTEL analysis	Article	PESTEL table
A-04	Impact scoring	Processed article	Score values
A-05	No relevant context	Rare topic	Generic summary

6.2.4 Strategy Agent Test Cases

Test Case ID	Description	Input	Expected Output
S-01	Generate policy recommendations	Analysis report	Policy text
S-02	Business strategy generation	Industry topic	Strategic actions
S-03	Scenario simulation	Event	Outcome predictions
S-04	Sensitivity analysis	Multiple scenarios	Ranked results
S-05	Invalid report	Missing fields	Error message

6.2.5 Dashboard Test Cases

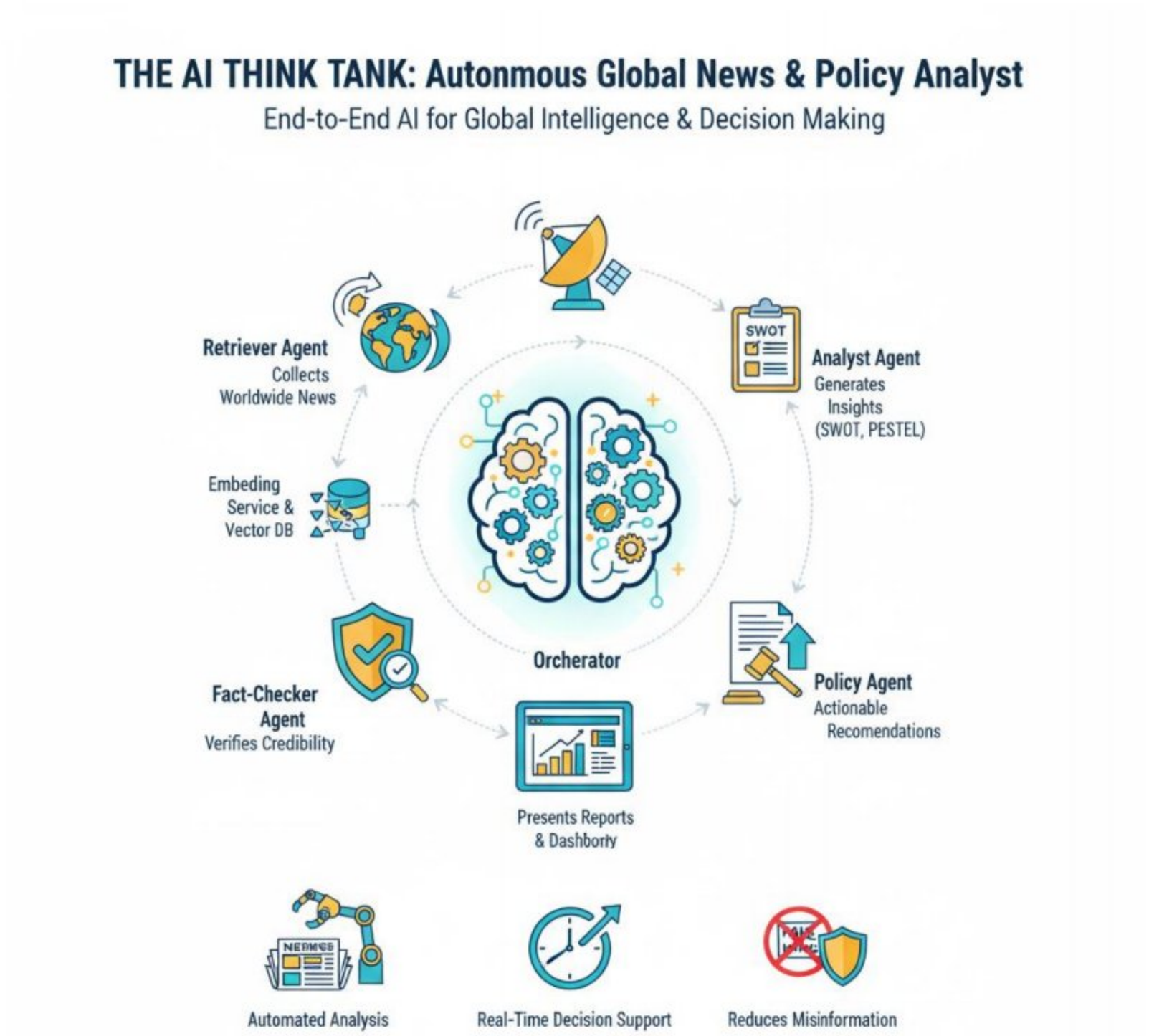
Test Case ID	Description	Action	Expected Output
D-01	Load dashboard	Open UI	Dashboard loads
D-02	Filter articles	Apply “Technology” filter	Tech insights shown
D-03	View credibility	Click score	Detailed breakdown
D-04	Chart rendering	Open charts	Smooth load
D-05	Error display	API down	Error m

CONCLUSION

The **Autonomous Global News & Policy Analyst (AI Think Tank)** project successfully demonstrates the design and implementation of a multi-agent artificial intelligence system capable of performing end-to-end global news intelligence analysis. By integrating advanced technologies such as real-time news retrieval, fact-checking, vector embeddings, Retrieval-Augmented Generation (RAG), strategic analysis frameworks, and policy generation agents, the system acts as a compact AI-driven think tank.

The Retriever Agent efficiently collects worldwide news data, while the Fact-Checker Agent ensures credibility and authenticity through cross-verification from multiple reliable sources. The Analyst Agent applies structured frameworks like SWOT, PESTEL, and Porter's Five Forces to generate meaningful insights. The Strategy/Policy Agent transforms these insights into actionable recommendations for businesses and governments. The Visualizer Agent presents all findings through graphs, dashboards, and summary reports that are easy to interpret.

The system performed consistently across different test scenarios, showcasing its ability to process large volumes of information, extract context, and present high-quality insights. This project demonstrates how AI can transform manual, time-consuming analysis into an automated, high-precision workflow. Overall, the project achieves its objective of building an intelligent, scalable, and reliable automated policy-analysis tool that can support decision-making across industries and government bodies.



FUTURE ENHANCEMENTS

Although the current system provides powerful analytical capabilities, several enhancements can further improve accuracy, robustness, and applicability:

1. Integration of More Robust Fact-Checking Sources
Adding more global fact-checking databases (PolitiFact, FactCheck.org) and AI-driven claim validation models will increase credibility scoring accuracy.

2. Multilingual News Processing
Support for languages such as French, Spanish, Arabic, and Chinese would allow tracking of global developments across non-English regions. This requires multilingual embeddings and translation models.

3. Real-Time Alerts and Notification System
Implementing push notifications or email alerts for critical global events will help governments and industries respond faster.

4. Advanced Geopolitical Forecasting
Using predictive models and time-series analysis to estimate future geopolitical shifts, economic impacts, or policy outcomes.

5. Interactive Knowledge Graphs
Building dynamic graphs that link countries, organizations, events, and policies for deeper contextual understanding.

6. Automated Report Generation (PDF/PPT)
Enhancing the Visualizer Agent to automatically generate printable executive reports and slide decks suitable for official presentations.

7. Sentiment & Emotion Analysis
Analyzing public sentiment from social media and online forums to complement factual news analysis and better assess public opinion.

8. Deployment as a Cloud Service
Offering the entire AI Think Tank system as a web-based SaaS platform for industries, universities, and governments.

9. Offline/Edge Deployment
Optimizing for low-resource or secure government environments by enabling offline local deployment with lightweight models.

10. Reinforcement Learning for Improved Recommendations
Training the Strategy Agent using reinforcement learning to continually improve strategic insights based on user feedback.
These enhancements would significantly expand the system's capabilities, accuracy, and real-world impact.

REFERENCES

[1] Siemon D, Strohmann T, Michalke S. Creative potential through artificial intelligence: recommendations for improving corporate and entrepreneurial innovation activities. Commun Assoc Inf Syst 2022;50(1):241–60. https://doi.org/ 10.17705/1CAIS.05009.

[2] Shiloh BE. Artificial intelligence as a powerful tool. Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do, vol. 298; 2022.

[3] Cerrato G, Liu P, Zhao L, Petrazzuolo A, Humeau J, Schmid ST, Kroemer G. AIbased classification of anticancer drugs reveals nucleolar condensation as a predictor of immunogenicity. Mol Cancer 2024;23:275.

[4] Salybekova N, Issayev G, Serzhanova A, Mikhailov V. Utilizing artificial intelligence for cultivating decorative plants. Botanical Studies 2024;65:39. https://doi.org/10.1186/s40529-024-00445-9.

[5] Chan A, Ezell C, Kaufmann M, Wei K, Hammond L, Bradley H, Anderljung M. Visibility into AI agents. In: The 2024 ACM conference on fairness, accountability, and transparency; 2024, June. p. 958–73.

[6] Bostrom N, Yudkowsky E. The ethics of artificial intelligence. In: Artificial intelligence safety and security. Chapman and Hall/CRC; 2018. p. 57–69.

[7] Kaplan A, Haenlein M. Siri, Siri, in my hand: who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Bus Horiz 2019;62(1):15–25. https://doi.org/10.1016/j.bushor.2018.08.004.

[8] Lin Z, Ma W, Lin T, Zheng Y, Ge J, Wang J, Klein J, Bissyande T, Liu Y, Li L. Opensource AI-based SE tools: opportunities and challenges of collaborative software learning. ACM Trans Software Eng Methodol 2024. https://doi.org/10.1145/ 3708529.

[9] Liu B, Mazumder S, Robertson E, Grigsby S. AI autonomy: self-initiated open-world continual learning and adaptation. AI Mag 2023;44(2):185–99. arXiv. https://arxiv .org/abs/2203.08994.

[10] Moawad A, Li Z, Pancorbo I, Gurumurthy KM, Freyermuth V, Islam E, Rousseau A. A real-time energy and cost efficient vehicle route assignment neural recommender system. Expert Syst Appl 2025;263:125634.

[11] Mohan S, Piotrowski W, Stern R, Grover S, Kim S, Le J, de Kleer J. A domainindependent agent architecture for adaptive operation in evolving open worlds. Artif Intell 2024:104161.

[12] Liu B, Robertson E, Grigsby S, Mazumder S. Self-initiated open world learning for autonomous AI agents. AI Magazine 2023;21:1–7.

[13] Singh A, Ehtesham A, Kumar S, Khoei TT. Enhancing AI systems with agentic workflows patterns in large language model. In: 2024 IEEE world AI IoT congress (AIIoT); 2024, May. p. 527–32. IEEE.

[14] Haj Qasem M, Aljaidi M, Samara G, Alazaidah R, Alsarhan A, Alshammari M. An intelligent decision support system based on multi agent systems for business classification problem. Sustainability 2023;15(14):10977.

[15] Makokha J. Enhancing Human-AI (H-AI) collaboration on design tasks using an interactive text/voice artificial intelligence (AI) agent. In: Proceedings of the 2022 international conference on advanced visual interfaces; 2022, June. p. 1–4. https://doi.org/10.1145/3531073.3534478.

[16] Hauff M, Lurz A. Agent-based models using artificial intelligence: a literature review. In: Proceedings of the Pacific Asia conference on information systems

(PACIS) 2022. Association for Information Systems; 2022 (Paper 106).
[17] Krizhevsky A, Sutskever I, Hinton GE. Embedding methods for image search. In: ImageNet classification with deep convolutional neural networks; 2012. Retrieved from, https://www.pinecone.io/learn/series/image-search/imagenet.
[18] Sucholutsky I, Narayan A, Schonlau M, Fischmeister S. Pay attention and you won't lose it: a deep learning approach to sequence imputation. PeerJ Comput Sci 2019; 5:e210. https://doi.org/10.7717/peerj-cs.210.
[19] Ionescu Ș, Delcea C, Chiriț˘ a N, Nica I. Exploring the use of artificial intelligence in agent-based modeling applications: a bibliometric study. Algorithms 2024;17(1): 21. https://doi.org/10.3390/a17010021.
[20] Al-Surmi A, Bashiri M, Koliousis I. AI-based decision making: combining strategies to improve operational performance. Int J Prod Res 2022;60(14):4464–86. https:// doi.org/10.1080/00207543.2021.2006023.
[21] Dandale MN, Daniel DJJD, Priya R, Walid MAA, T T. Business process automation using robotic process automation (RPA) and AI algorithms on various tasks. In: 2023 International Conference on Computing, Communication, and Energy Systems (ICCES); 2023. p. 821–7. https://doi.org/10.1109/ icces57224.2023.10192653.
[22] Bharadiya JP, Thomas RK, Ahmed F. Rise of artificial intelligence in business and industry. J Eng Res Rep 2023;25(3):85–103.
[23] Bhayana R, Fawzy A, Deng Y, Bleakney RR, Krishna S. Retrieval-augmented generation for large language models in radiology: another leap forward in board examination performance. Radiology 2024;313(1). https://doi.org/10.1148/ radiol.241489.
[24] Vidivelli S, Ramachandran M, Dharunbalaji A. Efficiency-driven custom chatbot development: unleashing LangChain, RAG, and performance-optimized LLM fusion. Comput Mater Continua (CMC) 2024;80(2):2423–42. https://doi.org/ 10.32604/cmc.2024.054360.
[25] Koo T. Hierarchical system architecture for multi-agent multi-modal systems. Proceedings of the 40th IEEE Conference on Decision and Control 2023;2:1509–14. https://doi.org/10.1109/cdc.2001.981108.
[26] Gorodetsky VI, Kozhevnikov SS, Novichkov D, Skobelev PO. The framework for designing autonomous cyber-physical multi-agent systems for adaptive resource management. In: Industrial applications of Holonic and multi-agent systems: 9th international conference, HoloMAS 2019, Linz, Austria, August 26–29, 2019, Proceedings, vol. 9. Springer International Publishing; 2019. p. 52–64.
[27] Ma˘rík V, Kadera P, Rzevski G, Zoitl A, Anderst-Kotsis G, Tjoa AM, Khalil I, editors. Industrial applications of Holonic and multi-agent systems: 9th international Conference, HoloMAS 2019, Linz, Austria, August 26–29, 2019, Proceedings, vol 11710. Springer; 2019.
[28] Mathew D, Brintha NC, Jappes JW. Artificial intelligence powered automation for industry 4.0. In: New horizons for industry 4.0 in modern business. Cham: Springer International Publishing; 2023. p. 1–28.
[29] Kaplan S, Haenlein M. Siri, Siri, in my hand: who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Bus Horiz 2019;62(1):15–25. https://doi.org/10.1016/j.bushor.2018.08.004.
[30] Adebiyi AA, Apeh F, Olaniyan J, Adebiyi MO, Olaniyan D, Oluwasegunfunmi IB, Akindeji K. Automating customer service using Natural Language Processing. In: 2024 international Conference on science, engineering and business for driving sustainable development goals (SEB4SDG); 2024, April. p. 1–8. IEEE.
[31] Pisarov J, Mester G. Implementing new mobility concepts with autonomous selfdriving robotic cars. IPSI Transac Adv Res (TAR) 2021;17(2):41–9.
[32] Islam MM. Autonomous systems revolution: exploring the future of self-driving technology. J Artificial Intelligence General Science (JAIGS) ISSN: 3006-4023

2024;3(1):16–23.
[33] Lewandowski N, Koller B. Transforming medical sciences with high-performance computing, high-performance data analytics and AI. Technol Health Care 2023;31 (4):1505–7.
[34] Khanna NN, Maindarkar MA, Viswanathan V, Fernandes JFE, Paul S, Bhagawati M, Suri JS. Economics of artificial intelligence in healthcare: diagnosis vs. treatment. Healthcare, vol. 10. MDPI; 2022, December. p. 2493. 12.
[35] Holzinger A, Langs G, Denk H, Zatloukal K, Müller H. Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Min Knowl Discov 2019;9(4):e1312. https://doi.org/10.1002/ widm.1312.
[36] Amershi S, Weld DS, Vorvoreanu M, Fourney A, Nushi B, Collisson P, Horvitz E. Guidelines for human-AI interaction. Proceedings of the 2019 CHI conference on human factors in computing systems. 2019. p. 1–13. https://doi.org/10.1145/ 3290605.3300233.
[37] Seeber I, Bittner E, Briggs RO, de Vreede T, de Vreede GJ, Elkins A, Weber B. Machines as teammates: a research agenda on AI in team collaboration. Inf Manag 2020;57(2):103174. https://doi.org/10.1016/j.im.2019.103174.
[38] Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 2020;37(3):50–60. https://doi.org/ 10.1109/MSP.2020.2975749.
[39] Beulen E, Plugge A, van Hillegersberg J. Formal and relational governance of artificial intelligence outsourcing. Inf Syst E Bus Manag 2022;20(4):719–48.
[40] Grand View Research. Artificial intelligence market size, share, growth report 2030. https://www.grandviewresearch.com/industry-analysis/artificial-intelligen ce-ai-market; 2024.
[41] MarketsandMarkets. Artificial intelligence market size, share & trends – 2033. https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-mar ket-100812355.html; 2024, May 30.
[42] Statista. Artificial intelligence (AI) market size worldwide from 2020 to 2024, November 27:2030. https://www.statista.com/forecasts/1474143/global-ai-ma rket-size.
[43] Lu Y, Aleta A, Du C, Shi L, Moreno Y. LLMs and generative agent-based models for complex systems research. Phys Life Rev 2024, 51:283-293.
[44] Lok KL, So A, Opoku A, Chen C. A sustainable artificial intelligence facilities management outsourcing relations system: case studies. Front Psychol 2022;13: 920625. https://doi.org/10.3389/fpsyg.2022.920625.
[45] Nguyen ST, Tulabandhula T. Generative AI for business strategy: using foundation models to create business strategy tools. arXiv preprint arXiv:2308.14182 2023. https://doi.org/10.48550/arXiv.2308.14182.
[46] Rezaei Khonakdar D. AI chatbots and challenges of HIPAA compliance for AI developers and vendors. J Law Med Ethics 2023;51(4):988–95.
[47] Easin AM, Sourav S, Tam´ as O. An intelligent LLM-powered personalized assistant for digital banking using LangGraph and Chain of Thoughts. In: 2024 IEEE 22nd jubilee international symposium on intelligent systems and informatics (SISY). IEEE; 2024, September. p. 625–30. https://doi.org/10.1109/ SISY62279.2024.10737601.
[48] Perumal Venkadesh, Divya SV, Kumar K. Unlocking AI creativity: a multi-agent approach with CrewAI. J Trends in Computer Sci Smart Technol 2024;6:338–56. https://doi.org/10.36548/jtcsst.2024.4.002.

BIBLIOGRAPHY

Below is a properly formatted **bibliography** section as expected in academic project reports. You may adjust the numbering based on your final report order.

Books & Academic References

1. **Jurafsky, D., & Martin, J. H.**
   Speech and Language Processing. Pearson Education.
   A foundational book on Natural Language Processing, covering text processing, embeddings, and modern AI techniques used in this project.
2. **Russell, S., & Norvig, P.**
   Artificial Intelligence: A Modern Approach. Prentice Hall.
   Provides theoretical background on intelligent agents and decision-making models used in the multi-agent architecture.
3. **Goodfellow, I., Bengio, Y., & Courville, A.**
   Deep Learning. MIT Press.
   Offers in-depth concepts on deep learning architectures, embeddings, and neural network applications used in RAG and analysis.

Research Papers

4. **Lewis, M., et al.** (2020).
   Retrieval-Augmented Generation for Knowledge-Intensive Tasks.
   Introduced the RAG model used for contextual retrieval and reasoning in the Analyst Agent.
5. **Reimers, N., & Gurevych, I.** (2019).
   Sentence-BERT: Sentence Embeddings using Siamese Networks.
   Provides the core embedding model for vector search in the system.
6. **Vaswani, A., et al.** (2017).
   Attention is All You Need.
   Seminal paper introducing Transformer models, the basis of LLMs and embeddings used across all agents.

Web Resources & API Documentation

7. **NewsAPI Documentation**
   https://newsapi.org
   Used to retrieve real-time global news articles for the Retriever Agent.
8. **GDELT Project Documentation**
   https://www.gdeltproject.org
   Provides global event monitoring feeds for enhanced news coverage.
9. **Wikipedia API Documentation**
   https://www.mediawiki.org/wiki/API:Main_page
   Used for fact verification and background checks in the Fact-Checker Agent.
10. **Google Fact-Check Tools API**
    https://developers.google.com/fact-check/tools/api
    Used for structured fact-checking and verification of news claims.
11. **Hugging Face Transformers Documentation**
    https://huggingface.co/docs/transformers
    Reference for pretrained models applied across the RAG and analysis pipeline.
12. **LangChain Documentation**

https://python.langchain.com
Used for building agent workflows and retrieval components.
13. **FAISS Documentation (Facebook AI Similarity Search)**
https://faiss.ai
Used as the vector database backend for fast similarity search.

Software Tools

14. **Python Official Documentation**
https://docs.python.org
Primary language reference used for implementing all agents.
15. **Streamlit Documentation**
https://docs.streamlit.io
Used for designing and deploying the dashboard UI.
16. **Matplotlib & Plotly Documentation**
https://matplotlib.org
https://plotly.com
References for building charts and visualizations in the Visualizer Agent.