.pathway

Dynamic Agentic RAG
Team 73

Understanding Problem Statement

Develop an Agentic Retrieval-Augmented Generation (RAG) system using **Pathway** to enhance LLM capabilities for handling complex queries with accuracy and efficiency. RAG combines LLMs with retrieval mechanisms to fetch relevant information from external data sources, enabling contextually rich and precise responses.

Key Features

Dynamic Adaptability

Multi-Agent Collaboration

Auxiliary Web Search

Token Optimization

Key Challenges in traditional RAG

- Hierarchical and Unified Memory Storage
- Consensus Memory Integrity
- Episodic Memory and Communication
- Optimizing Latency and Efficiency
- Addressing Hallucinations
- Absolute Evaluation Metrics

Components of Pipeline

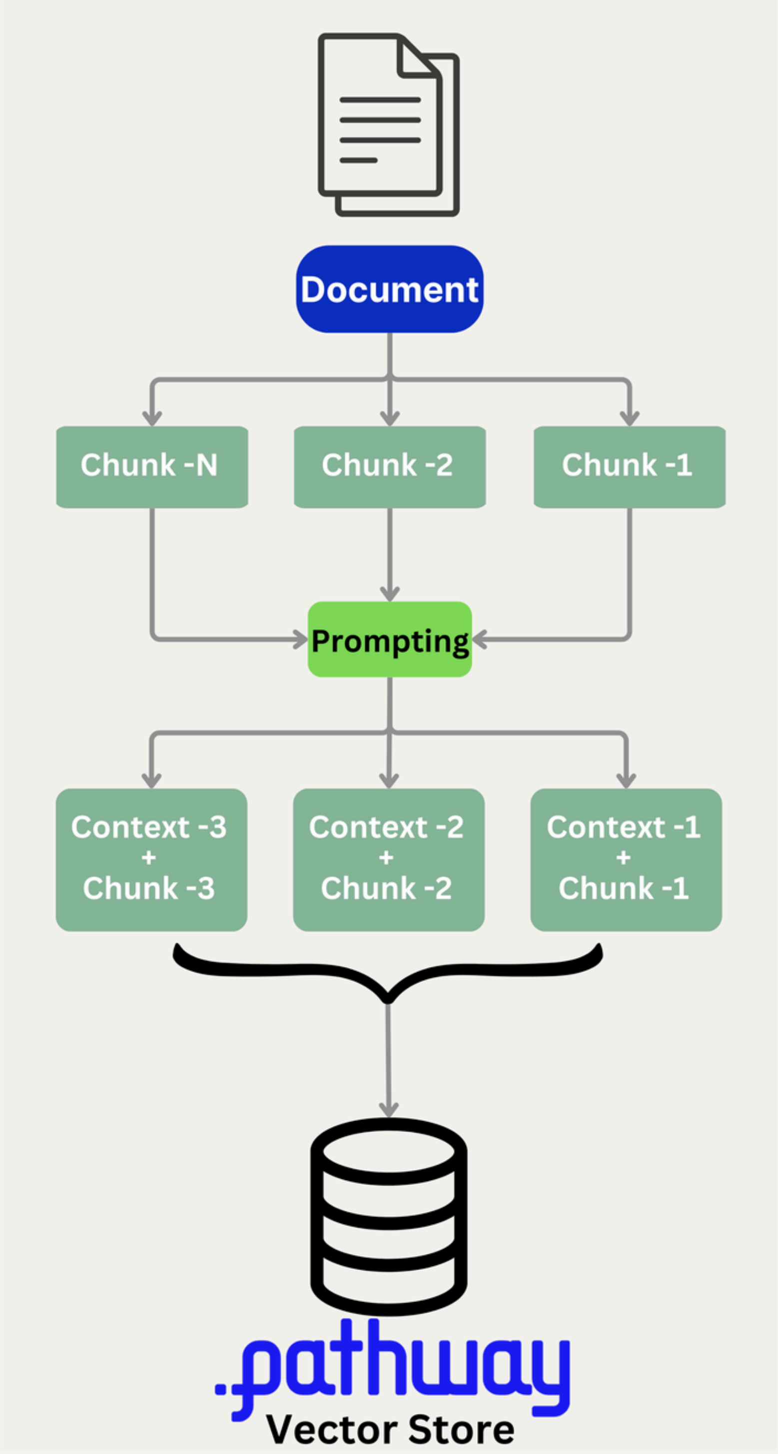
Retrieval

Agents

Generation

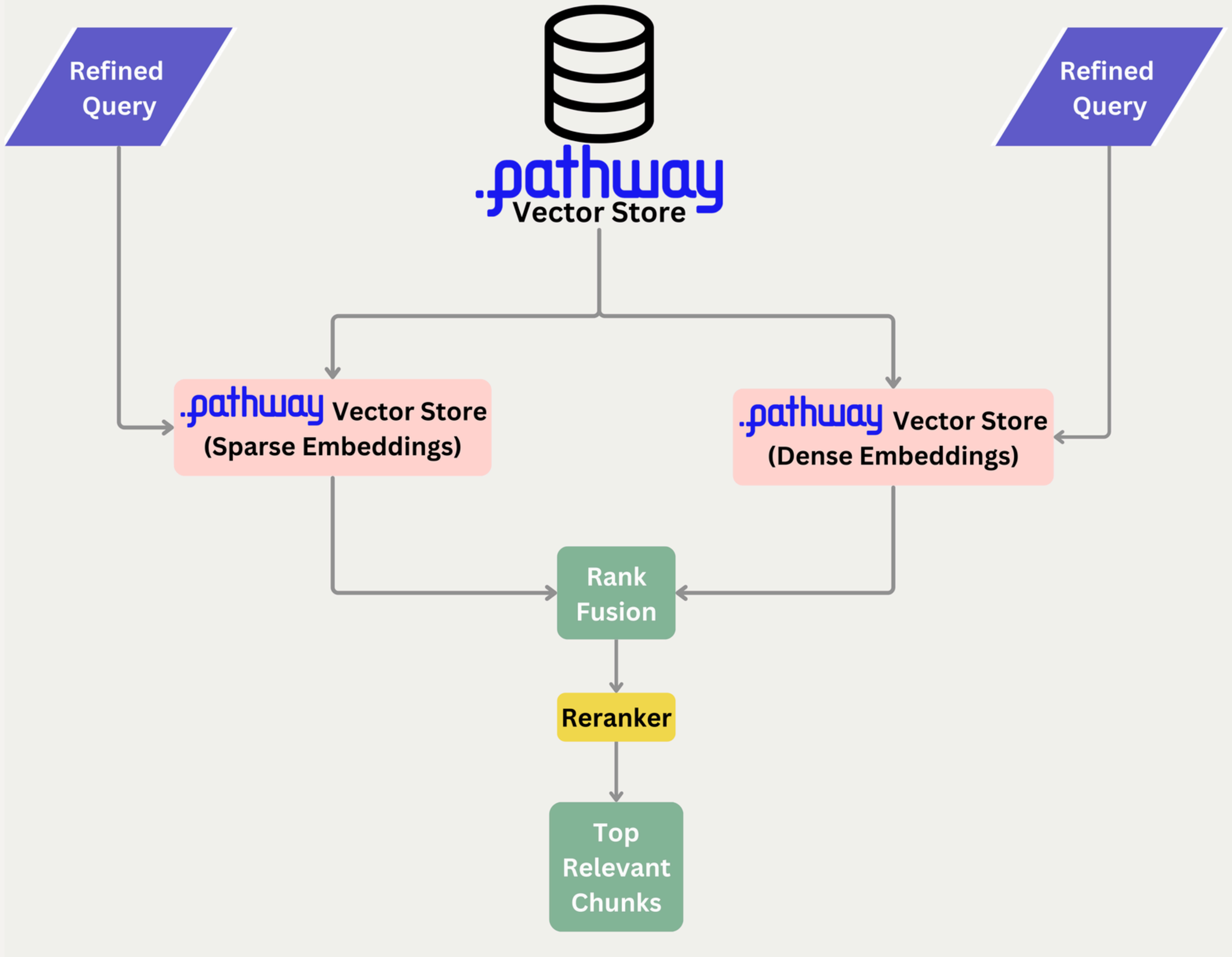
Retrieval

Contextual Retrieval Splitter



- Integrated contextual retrieval splitter in **Pathway's vector store**.
  - Utilizes a recursive text splitter to split chunks.
  - Each chunk is appended with context and document metadata.
- Introduced a document summary storage function:
  - Parses the document and stores its summary in a .txt file, which can later be used for later tasks.

Sparse Encoder



- Integrated sparse encodings in **Pathway's vector store**.
    - Extended the base embedder class to include a Splade encoder.
- Sparse embeddings support contextual retrieval by combining semantic and term-based relevance.

Rank Fusion

- Our pipeline leverages contextual retrieval with rank fusion to extract the most relevant chunks from documents.
- We are using Jina AI reranker in our pipeline .
- The formula for rank fusion is:

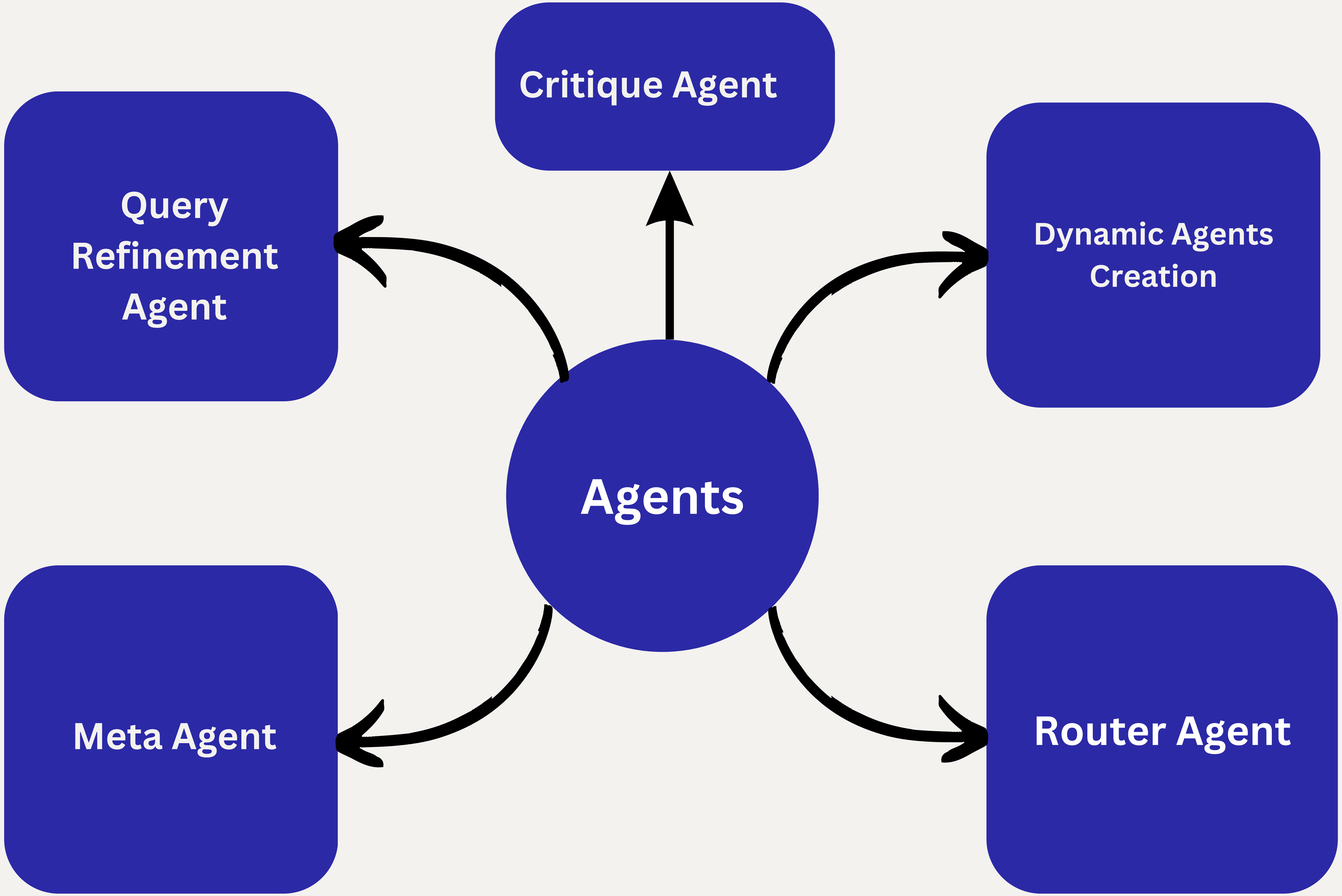
$$\text{score} = \left(\frac{1}{r_d + 10} + \frac{1}{r_s + 10} \right)^{-1}$$

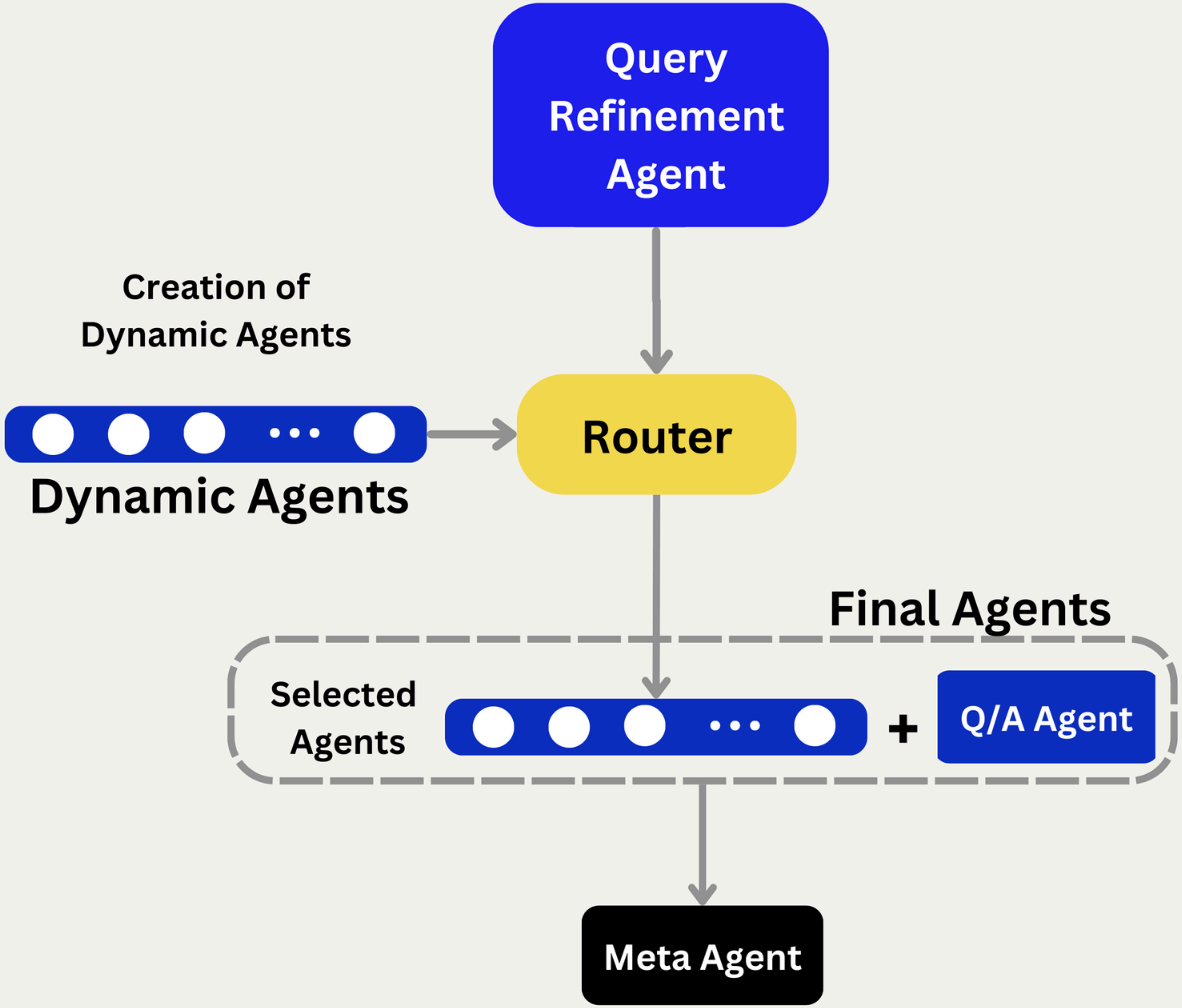
Where:

- r_d : Rank of chunks in dense embedding
- r_s : Rank of chunks in sparse embedding

- **Note**: +10 in the denominator helps ensure that a single dense or sparse embedder does not overly affect a chunk's rank.

Our Agentic Crew

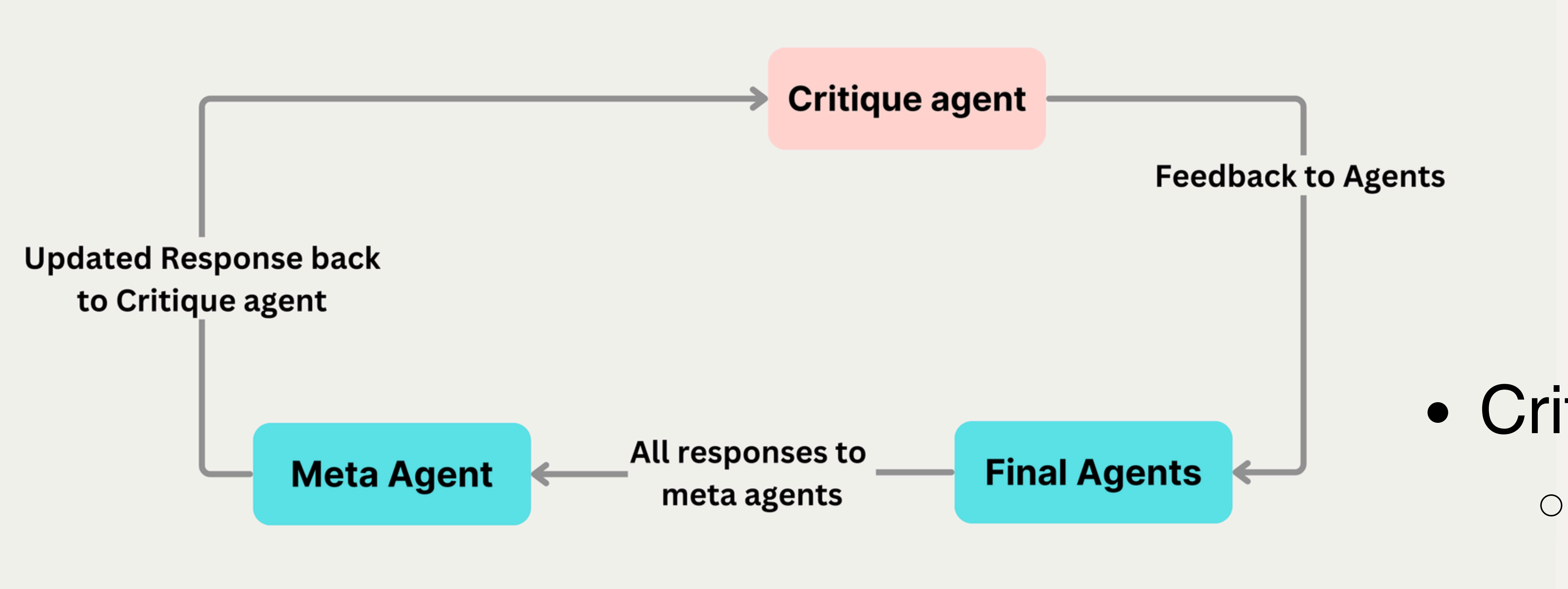

Critique Agent
Query Refinement Agent
Dynamic Agents Creation
Agents
Meta Agent
Router Agent


Query
Refinement
Agent
Creation of
Dynamic Agents
Router
Dynamic Agents
Final Agents
Selected
Agents
+
Q/A Agent
Meta Agent

Generation

Reflection

.pathway



- Reflection :
  - Before getting final response we Integrated reflection which has a critique agent to provide constructive feedback on agent responses.
- Critique agent functionality:
  - Leverages a knowledge base of the original query and contextual information from the document.
- Reflection loop:
  - Conducted over N iterations.

Guardrails

Validates the text scope, clarity, and safety using REGEX filters and LLMs by Guardrails-AI.

1. **Input Query Guardrailing**:
   - Prevents invalid inputs for accurate downstream processing.

2. **Response output Guardrailing**:
   - Aggregates inputs from all agents to refine responses with guardrails. Ensures robust and safe final outputs.

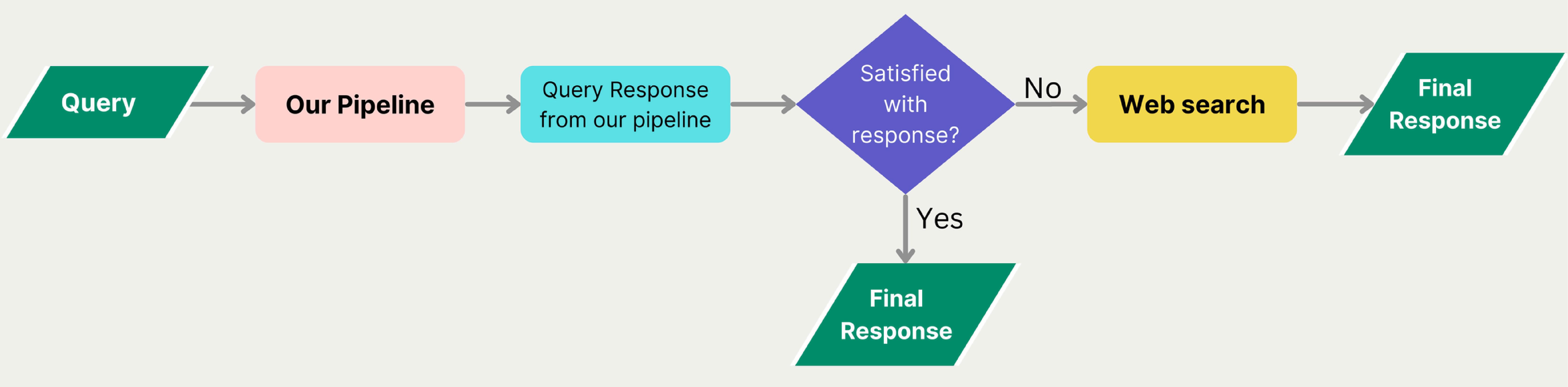
Error Handling

Web Search Option :

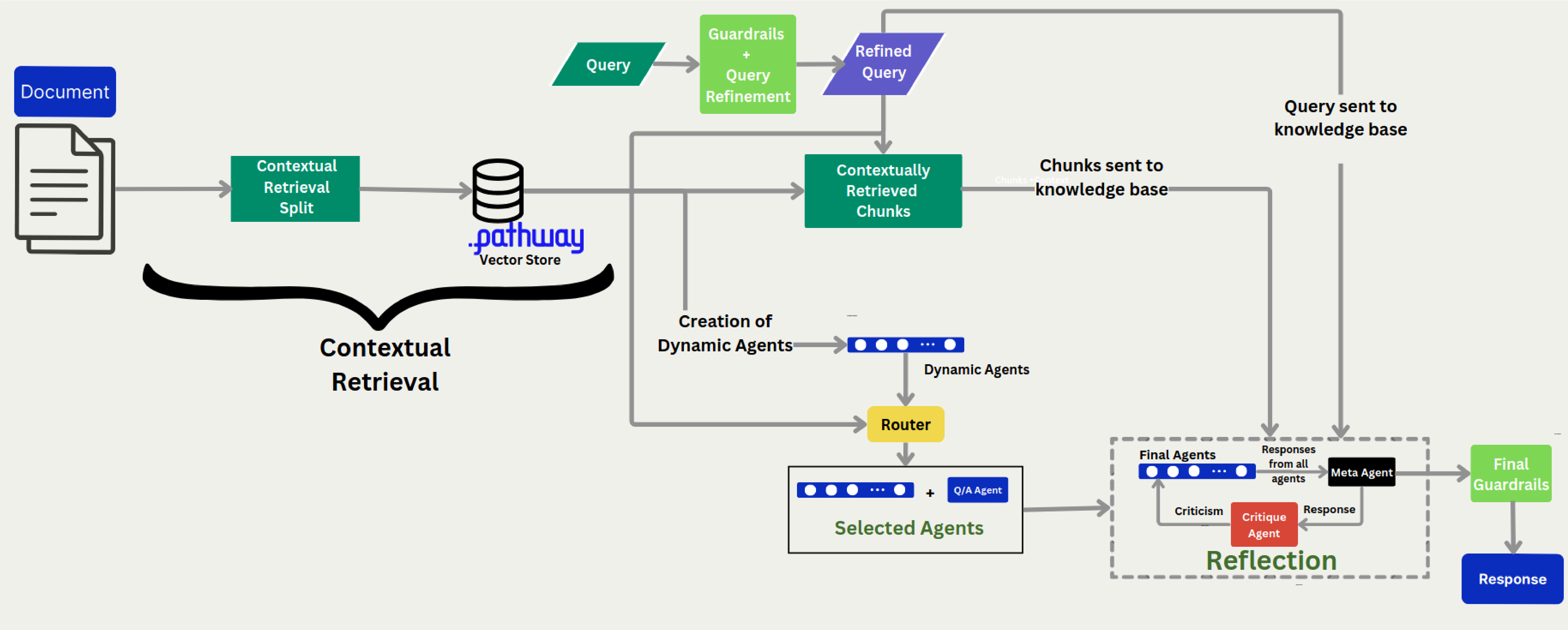
- Uses Jina AI API for web search when user is not satisfied or query received is out of context.
- Fallback mechanism switches to Exa API if Jina AI fails .

Quality Assurance:

- Implements guardrails with safety protocols and context validation.
- Ensures responses align with ethical, contextual, and safety standards.

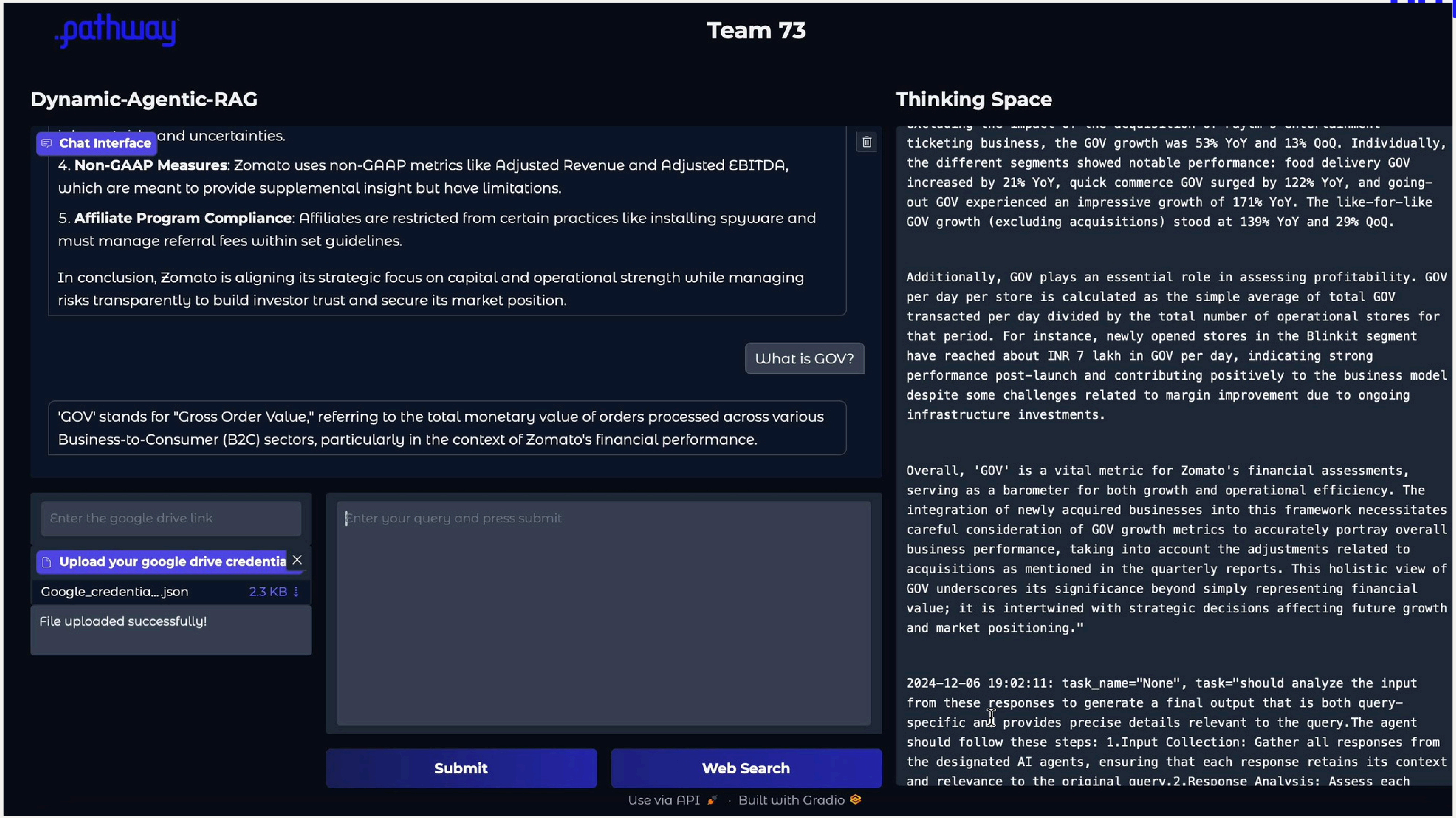


System Architecture


Document
Contextual Retrieval Split
pathway
Vector Store
Contextual Retrieval
Query
Guardrails + Query Refinement
Refined Query
Query sent to knowledge base
Contextually Retrieved Chunks
Chunks sent to knowledge base
Creation of Dynamic Agents
Dynamic Agents
Router
Q/A Agent
Selected Agents
Final Agents
Responses from all agents
Meta Agent
Criticism
Critique Agent
Response
Reflection
Final Guardrails
Response

User Interface

- Intuitive Design with Gradio
- Dynamic Google Drive Processing
- Real-Time Thinking Space
- Web-Search Functionality



INTER IIT TECH MEET 13.0

Deployment

- **Dockerization:**

The entire codebase is orchestrated using docker compose. We have two containers one for our frontend and one for our backend.

- **FastAPI server:**

The frontend and backend communicate seamlessly through a FastAPI server. We use our FastAPI server to stream our agent responses, to our frontend.

Metrics

Answer Relevancy: Evaluates how closely the response matches the prompt in terms of completeness and relevance, avoiding redundant information.

$$\text{Answer Relevancy} = \frac{1}{N}\sum_{i=1}^{N} \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\|\|E_o\|}$$

- E_{g_i} is the embedding of the generated question i.
- E_o is the embedding of the original question.

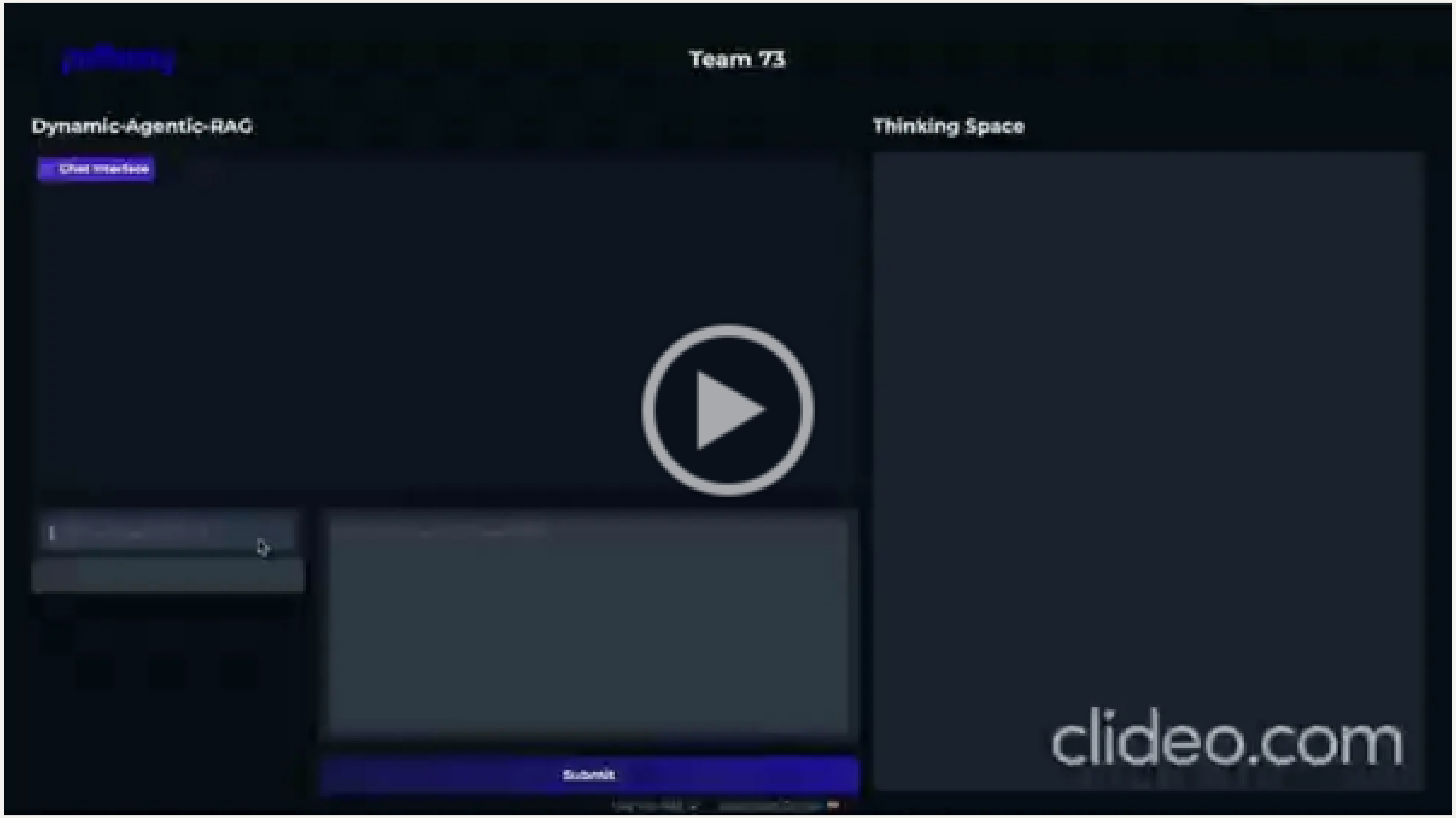
Semantic Similarity: Measures how well the generated response aligns with the ground truth based on semantic meaning.

$$\text{Semantic Similarity} = \frac{E_a \cdot E_g}{\|E_a\|\|E_g\|}$$

- E_a is the embedding of the generated answer.
- E_g is the embedding of the ground truth answer.

Demo Video

.pathway



Result Table

Contextual Retrieval	Reranker	Reflection (n)	Time of Inference	Semantic Similarity	Answer Relevancy
✓	✓	n=0	19.2 sec	88.33%	91.17%
✓	✓	n=1	30.1 sec	88.68%	93.2%
✓	✓	n=2	45.52 sec	89.98%	95.27%
✓	✗	n=0	25.4 sec	85.87%	89.01%
✓	✗	n=1	32.8 sec	87.1%	89.16%
✓	✗	n=2	40.97 sec	88.19%	89.52%
✗	✓	n=0	23 sec	86.23%	89.88%
✗	✓	n=1	37.1 sec	87.26%	90.86%
✗	✓	n=2	38.3 sec	88.14%	91.01%
✗	✗	n=0	20.8 sec	84.23%	86.53%
✗	✗	n=1	30.9 sec	86.14%	88.24%
✗	✗	n=2	45.4 sec	86.99%	89.06%

Average time of Inference across all cases is 32.4575 Sec

Future Enhancements

1. **Knowledge Repository Integration:** Add support for Notion and SharePoint for dynamic content retrieval.
2. **Multimodal Capabilities:** Use VLMs to process images, charts, and visual data and integrate it in Pathway's xpacks.llm .
3. **Tool Integration:** Incorporate mathematical solvers and computational tools for advanced queries.

Business Use-case

- **Legal Document Analysis**: Process and summarize lengthy contracts, identify key clauses, and provide accurate answers to legal queries.
- **Financial Insights and Market Research**: Analyse financial documents, generate summaries, extract insights from market reports, and support decision-making in finance and strategy.
- **Customer Support Automation**: Deliver precise, context-aware responses to customer inquiries, improving efficiency and user satisfaction.

Reference

CrewAi

Self-RAG Reflection

Ragas semantic similarity

Ragas answer relevancy

JinaAI

Guardrails Validators

Exa AI

Thank You