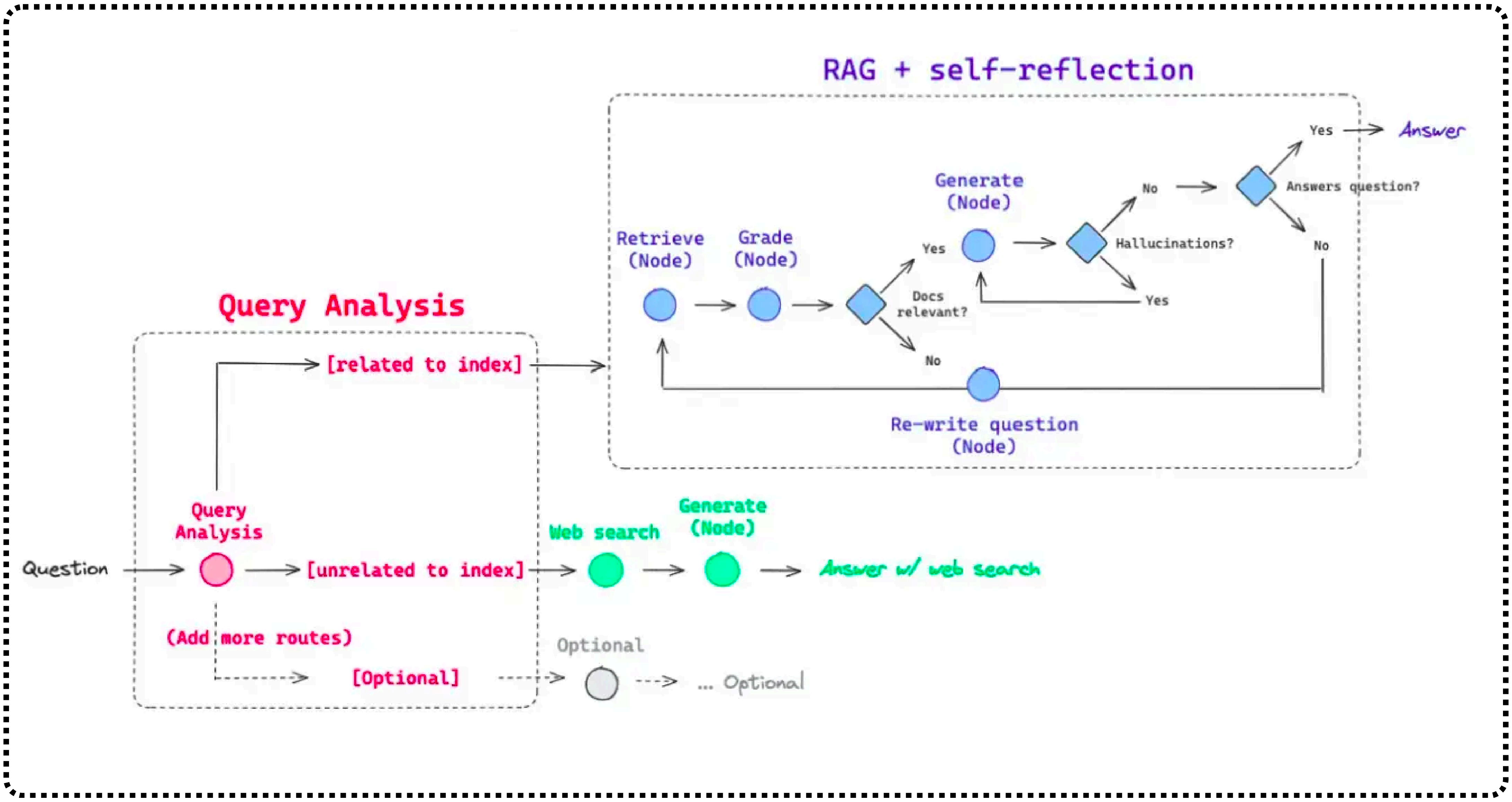
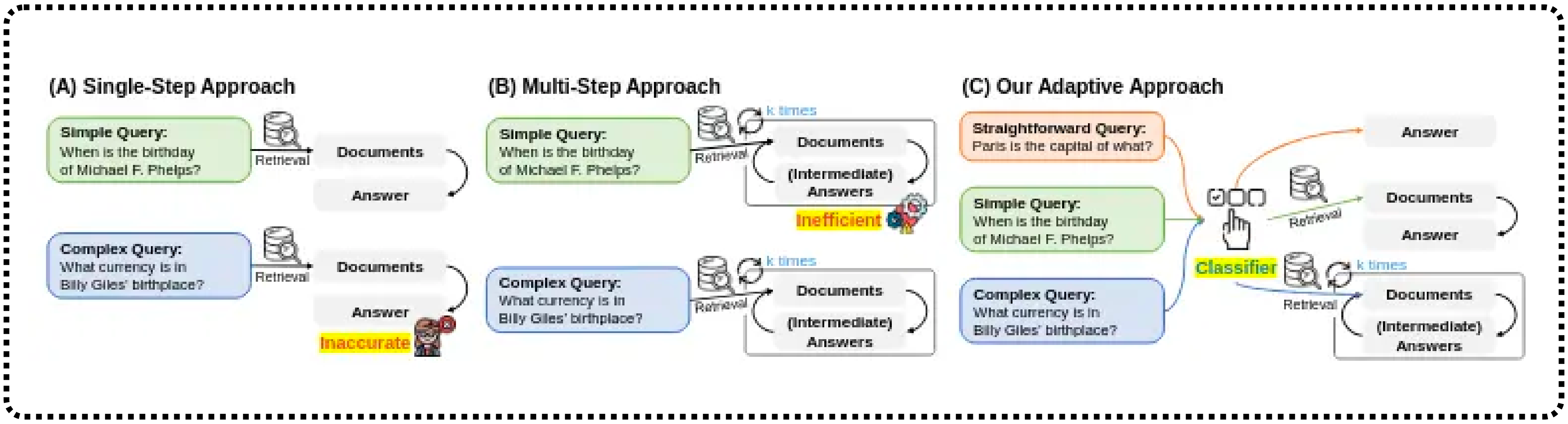
Analytics Vidhya

Guide to Adaptive RAG Systems with LangGraph

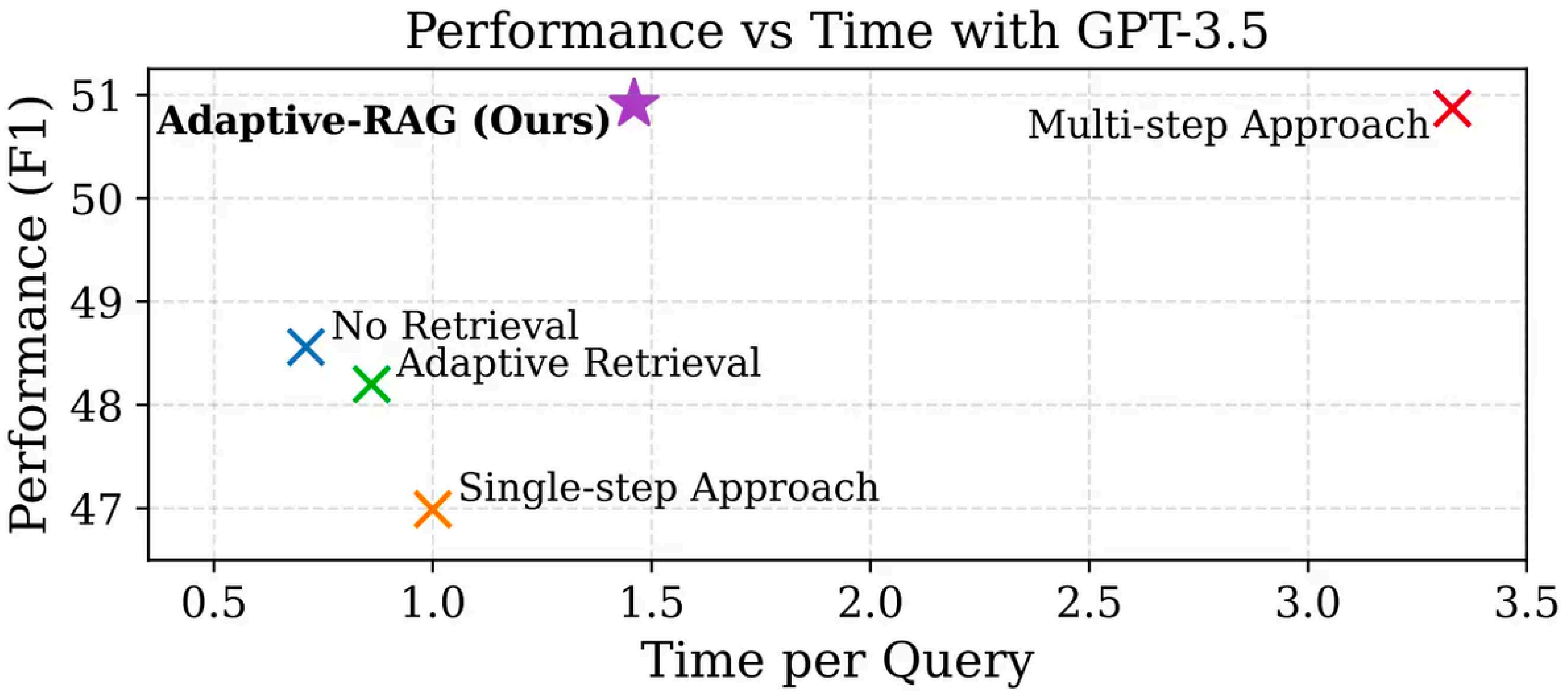



(A) Single-Step Approach
Simple Query: When is the birthday of Michael F. Phelps?
Retrieval
Documents
Answer
Complex Query: What currency is in Billy Giles' birthplace?
Retrieval
Documents
Answer
Inaccurate
(B) Multi-Step Approach
Simple Query: When is the birthday of Michael F. Phelps?
k times
Retrieval
Documents
(Intermediate) Answers
Inefficient
Complex Query: What currency is in Billy Giles' birthplace?
k times
Retrieval
Documents
(Intermediate) Answers
(C) Our Adaptive Approach
Straightforward Query: Paris is the capital of what?
Simple Query: When is the birthday of Michael F. Phelps?
Complex Query: What currency is in Billy Giles' birthplace?
Classifier
Answer
Retrieval
Documents
Answer
k times
Retrieval
Documents
(Intermediate) Answers

Why Adaptive-RAG Works So Well?

- Adaptive-RAG is flexible. It handles each question efficiently, using only the resources needed. This saves computing power and makes users happier because they get precise answers quickly. Adaptive-RAG also updates itself based on question complexity. Because it adapts, it reduces the chance of old or too-general answers. It gives current, specific and reliable answers.

- Adaptive-RAG is very accurate. It understands the complexity of each question and adapts. By matching the retrieval method to the question, Adaptive-RAG makes sure answers are relevant. It also uses the newest available information. This adaptability greatly reduces the chance of getting old or generic answers. Adaptive-RAG sets a new standard for accurate automated question answering.

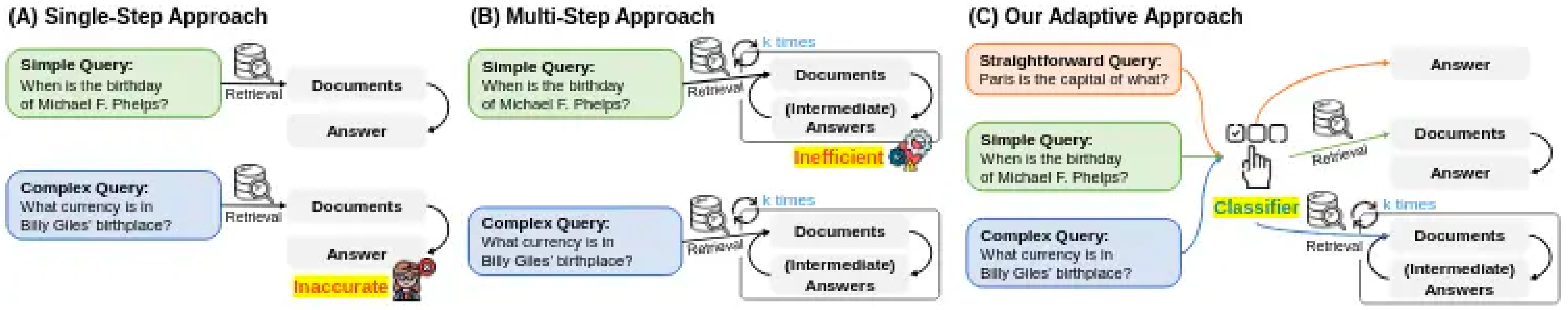


What is Adaptive RAG?

- Adaptive Retrieval-Augmented Generation (Adaptive RAG) is a RAG architecture designed to improve how large language models (LLMs) handle queries. It adjusts the strategy for answering questions based on their complexity.

The Key Idea of Adaptive RAG

- Adaptive RAG selects the best approach for answering a query, whether it's simple or complex. A classifier evaluates the query and decides whether to use a straightforward method, a single-step retrieval, or a multi-step process.


(A) Single-Step Approach
Simple Query: When is the birthday of Michael F. Phelps?
Retrieval
Documents
Answer
Complex Query: What currency is in Billy Giles' birthplace?
Retrieval
Documents
Answer
Inaccurate
(B) Multi-Step Approach
Simple Query: When is the birthday of Michael F. Phelps?
Retrieval
k times
Documents
(Intermediate) Answers
Inefficient
Complex Query: What currency is in Billy Giles' birthplace?
Retrieval
k times
Documents
(Intermediate) Answers
(C) Our Adaptive Approach
Straightforward Query: Paris is the capital of what?
Simple Query: When is the birthday of Michael F. Phelps?
Complex Query: What currency is in Billy Giles' birthplace?
Classifier
Answer
Retrieval
Documents
Answer
Retrieval
k times
Documents
(Intermediate) Answers

Comparison with other approaches

As shown in the above image, here's the comparison of adaptive RAG with a single-step, multi-step approach.

Single-Step Approach

- This method retrieves information in one go and pass it to the LLm to generate an answer directly. Although it works well for simple questions like "What is Agentic RAG?", it struggles with complex queries. For instance "What is the use of Agentic RAG in the healthcare sector?" due to lack of reasoning on the retrieved information.

Multi-Step Approach

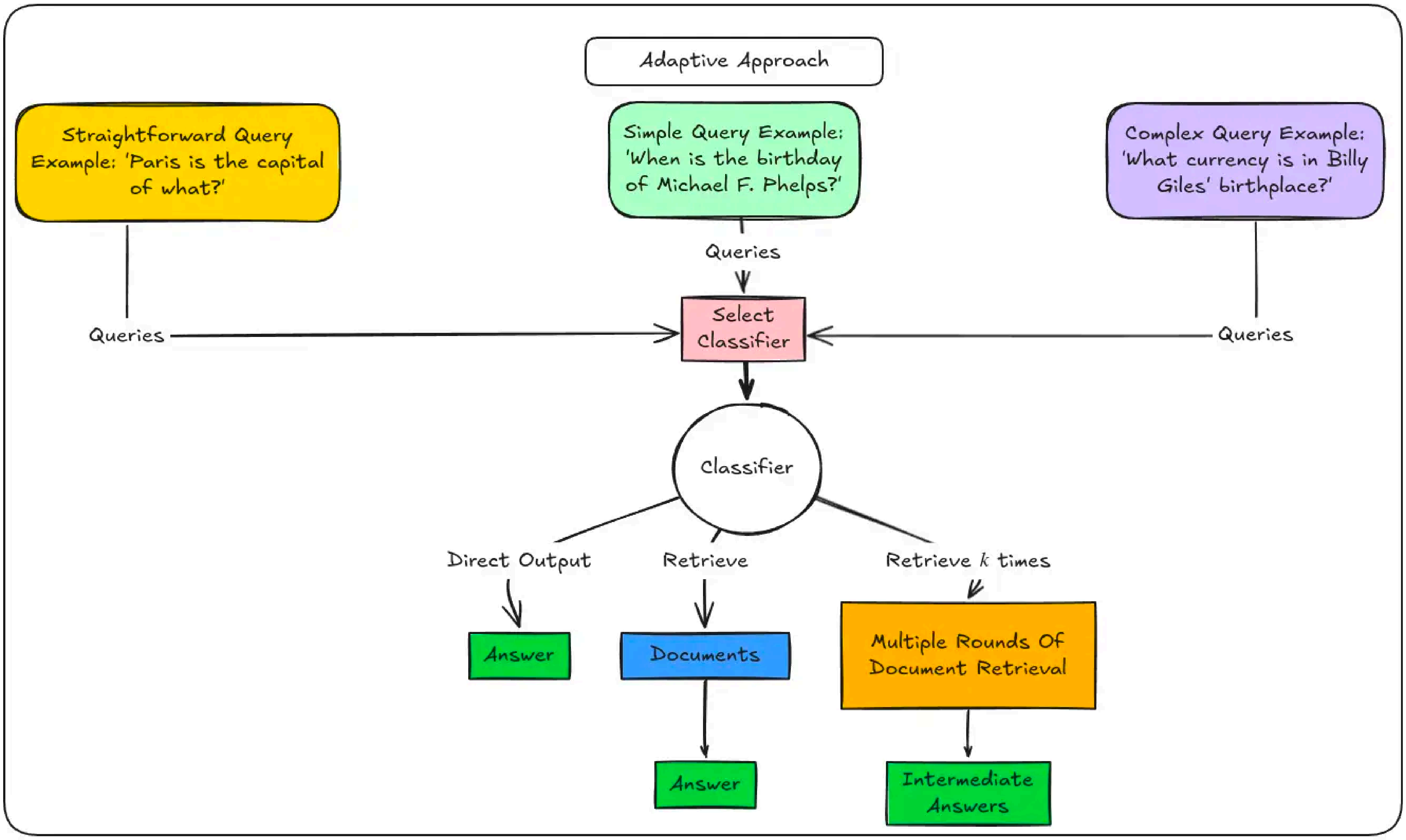
- This method processes all queries through multiple retrieval steps, refining answers iteratively. While effective for complex queries, it can waste resources on simple questions. For instance, repeatedly processing "What is Agentic RAG?" is inefficient.

Adaptive Approach

A classifier determines the complexity of the query and selects the appropriate strategy:

- Straightforward Query: Generates an answer without retrieval (e.g., "What is the capital of India?").

- **Simple Query**: Uses a single-step retrieval.
- **Complex Query**: Employs multi-step retrieval for detailed reasoning.
- **Advantages**: This method reduces unnecessary processing for simple queries while ensuring accuracy for complex ones.

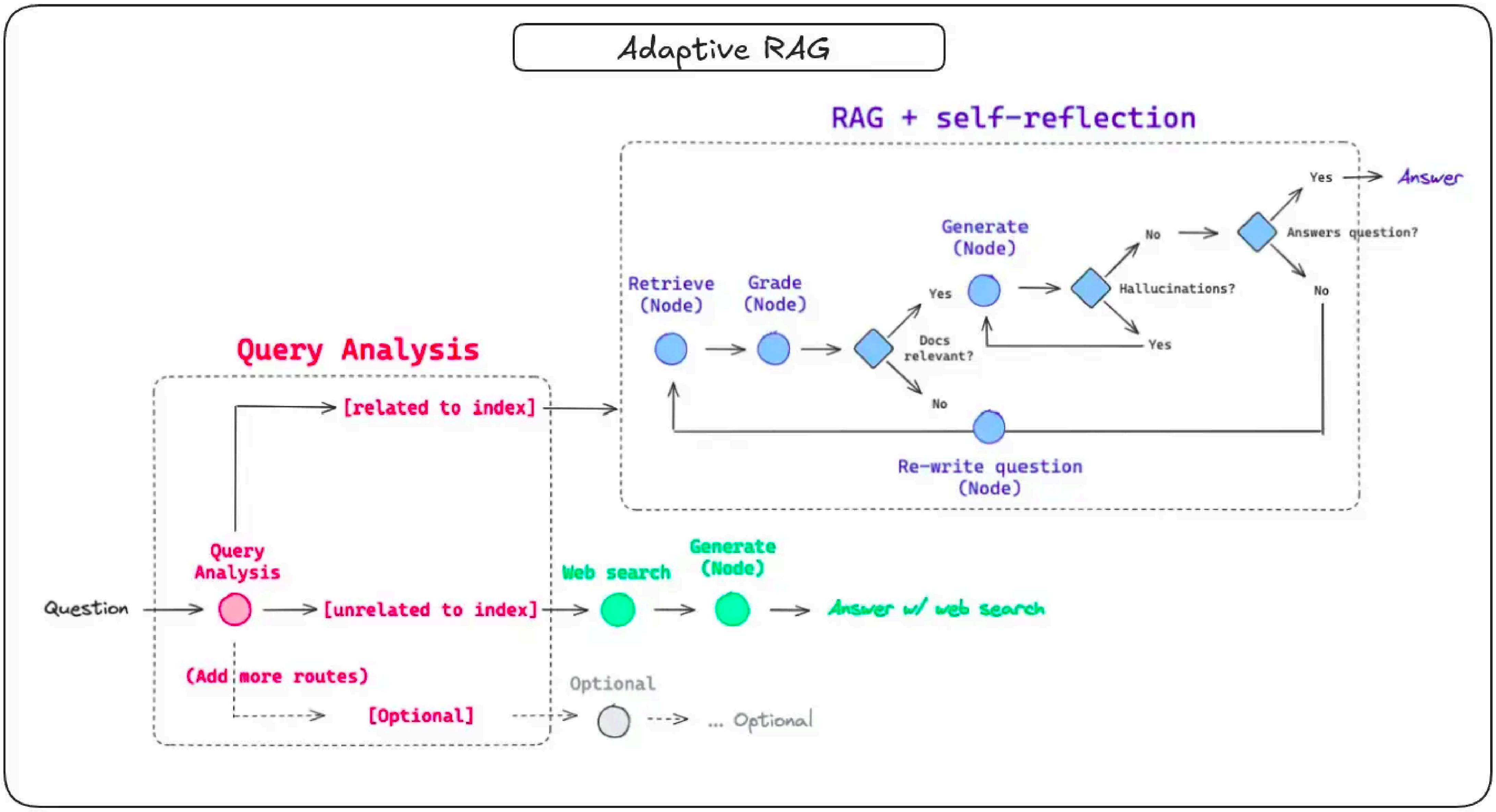

Adaptive Approach
Straightforward Query Example: 'Paris is the capital of what?'
Simple Query Example: 'When is the birthday of Michael F. Phelps?'
Complex Query Example: 'What currency is in Billy Giles' birthplace?'
Queries
Queries
Queries
Select Classifier
Classifier
Direct Output
Retrieve
Retrieve k times
Answer
Documents
Multiple Rounds Of Document Retrieval
Answer
Intermediate Answers

Adaptive RAG Framework

- **Classifier Role**: A smaller language model predicts the complexity of the query. It learns from past outcomes and patterns in the data.
- **Dynamic Strategy Selection**: The framework conserves resources for simple queries and ensures thorough reasoning for complex ones

RAG System Architecture Flow from LangGraph

Here's another example of an adaptive RAG System architecture flow from LangGraph:



Adaptive RAG System Architecture Flow:

1. Query Analysis

The process begins with analyzing the user query to determine the most appropriate pathway for retrieving and generating the answer.

- Route Determination
  - The query is classified into categories based on its relevance to the existing index (database or vector store).
  - Related to Index: If the query is aligned with the indexed content, it is routed to the RAG module for retrieval and generation.
  - Unrelated to Index: If the query is outside the scope of the index, it is routed for a web search or another external knowledge source.

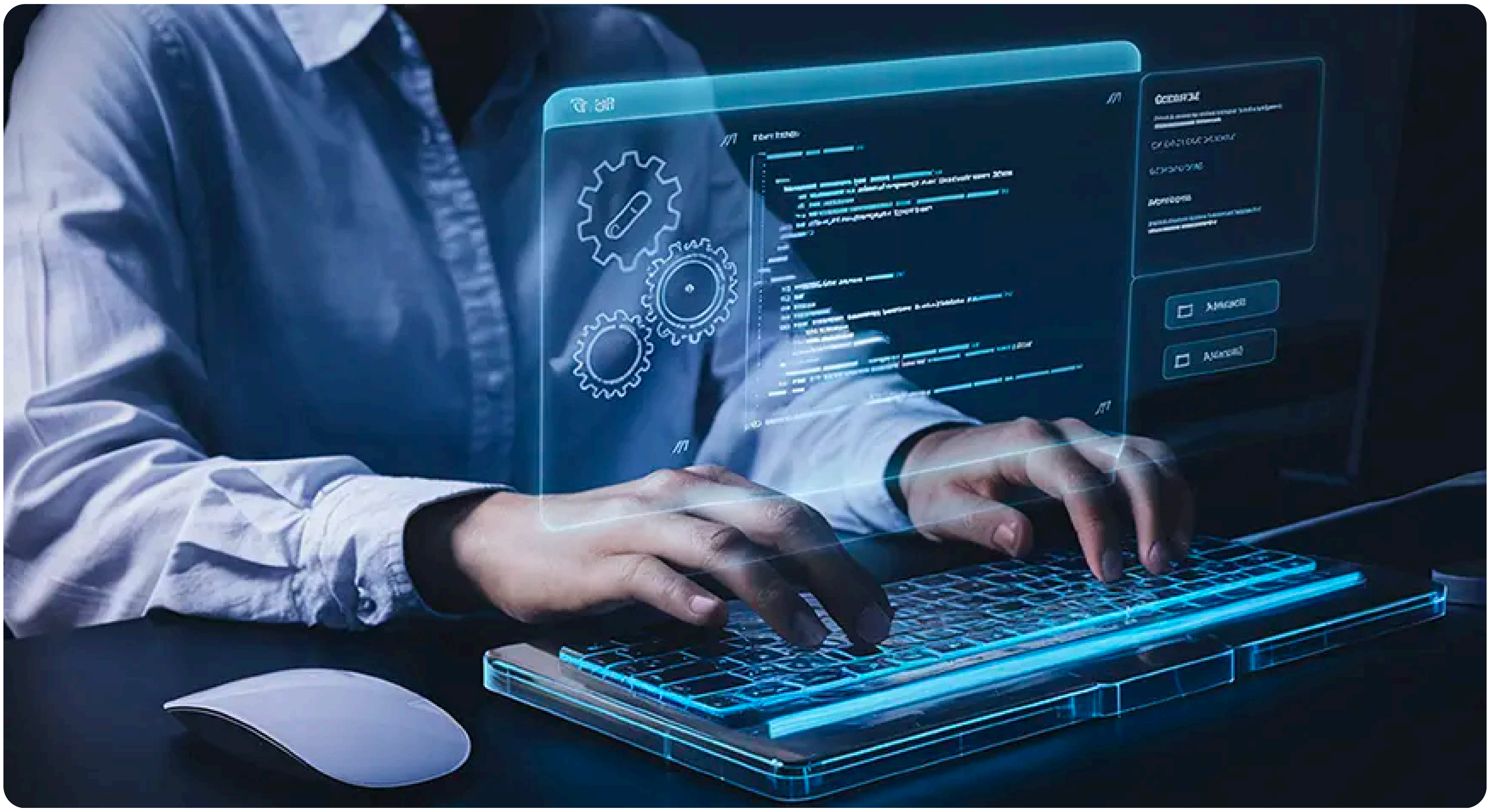
- Optional Routes: Additional pathways can be added for more specialized scenarios, such as domain-specific tools or external APIs.

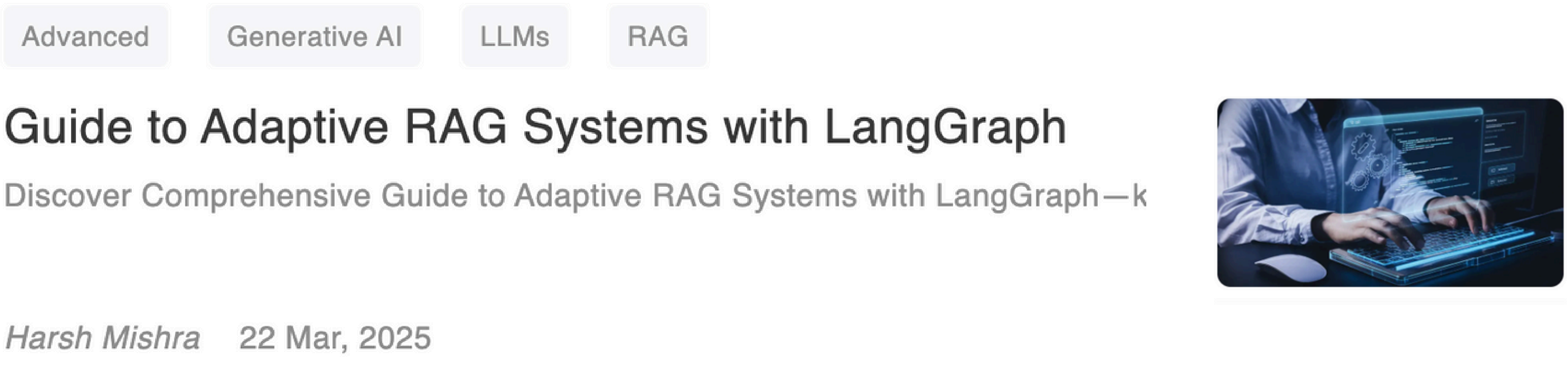
2. RAG + Self-Reflection

If the query is routed through the RAG module, it undergoes an iterative, self-reflective process to ensure high-quality and accurate responses.

- Retrieve Node: Retrieves documents from the indexed database based on the query. These documents are passed to the next stage for evaluation.
- Grade Node: Assesses the relevance of the retrieved documents.If documents are relevant: Proceed to generate an answer. If documents are irrelevant: The query is rewritten for better retrieval and the process loops back to the retrieve node.

For more information, you can visit this article





Advanced Generative AI LLMs RAG

Guide to Adaptive RAG Systems with LangGraph

Discover Comprehensive Guide to Adaptive RAG Systems with LangGraph—k

Harsh Mishra 22 Mar, 2025