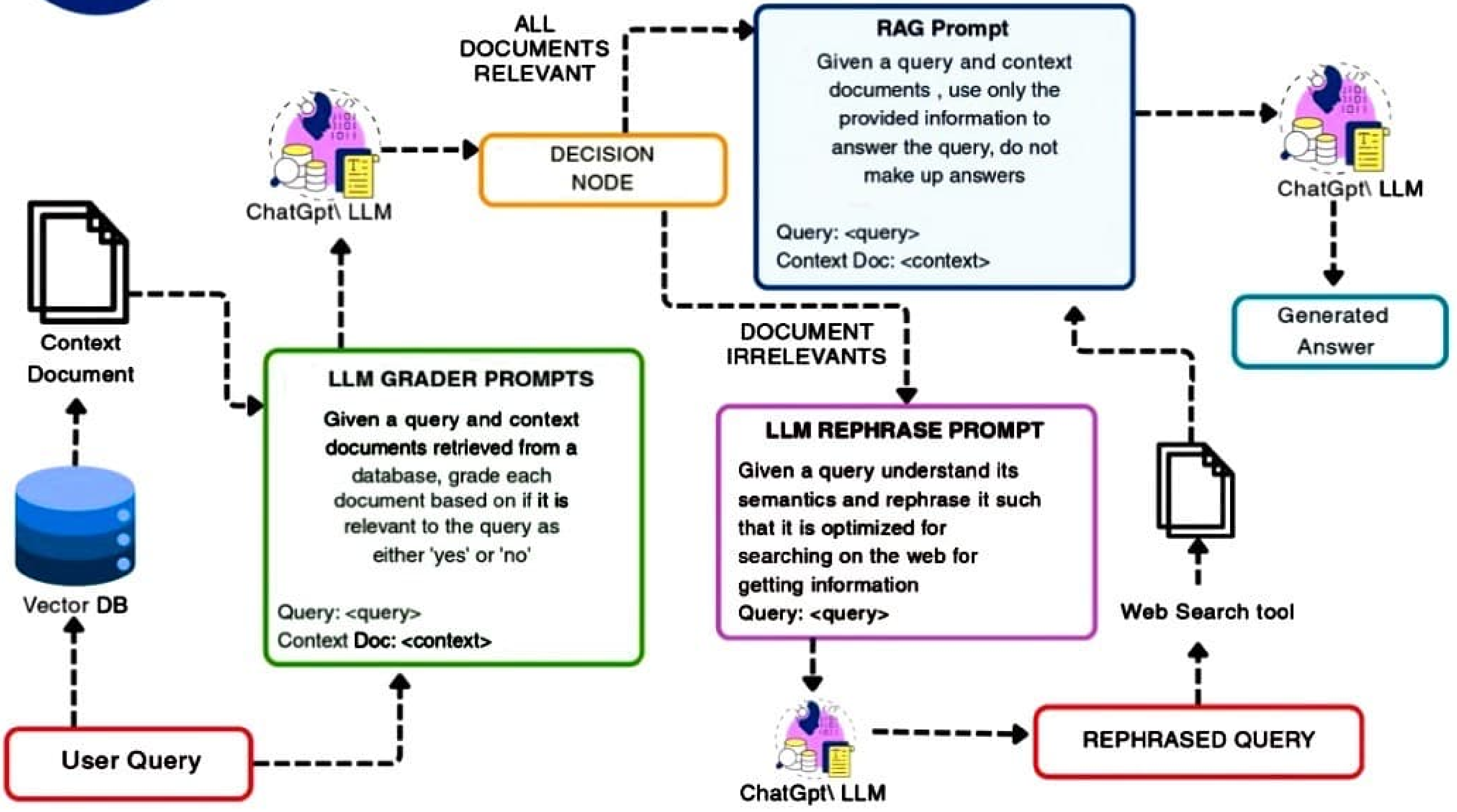
7 Types of Agentic RAG architectures

Follow Chehak K. for practical RAG tutorials

NEXT

1 Agentic Corrective RAG



ALL DOCUMENTS RELEVANT

RAG Prompt

Given a query and context documents , use only the provided information to answer the query, do not make up answers

Query: <query>
Context Doc: <context>

DECISION NODE

ChatGpt\ LLM

ChatGpt\ LLM

Generated Answer

Context Document

DOCUMENT IRRELEVANTS

LLM GRADER PROMPTS

Given a query and context documents retrieved from a database, grade each document based on if it is relevant to the query as either 'yes' or 'no'

Query: <query>
Context Doc: <context>

LLM REPHRASE PROMPT

Given a query understand its semantics and rephrase it such that it is optimized for searching on the web for getting information
Query: <query>

Vector DB

Web Search tool

User Query

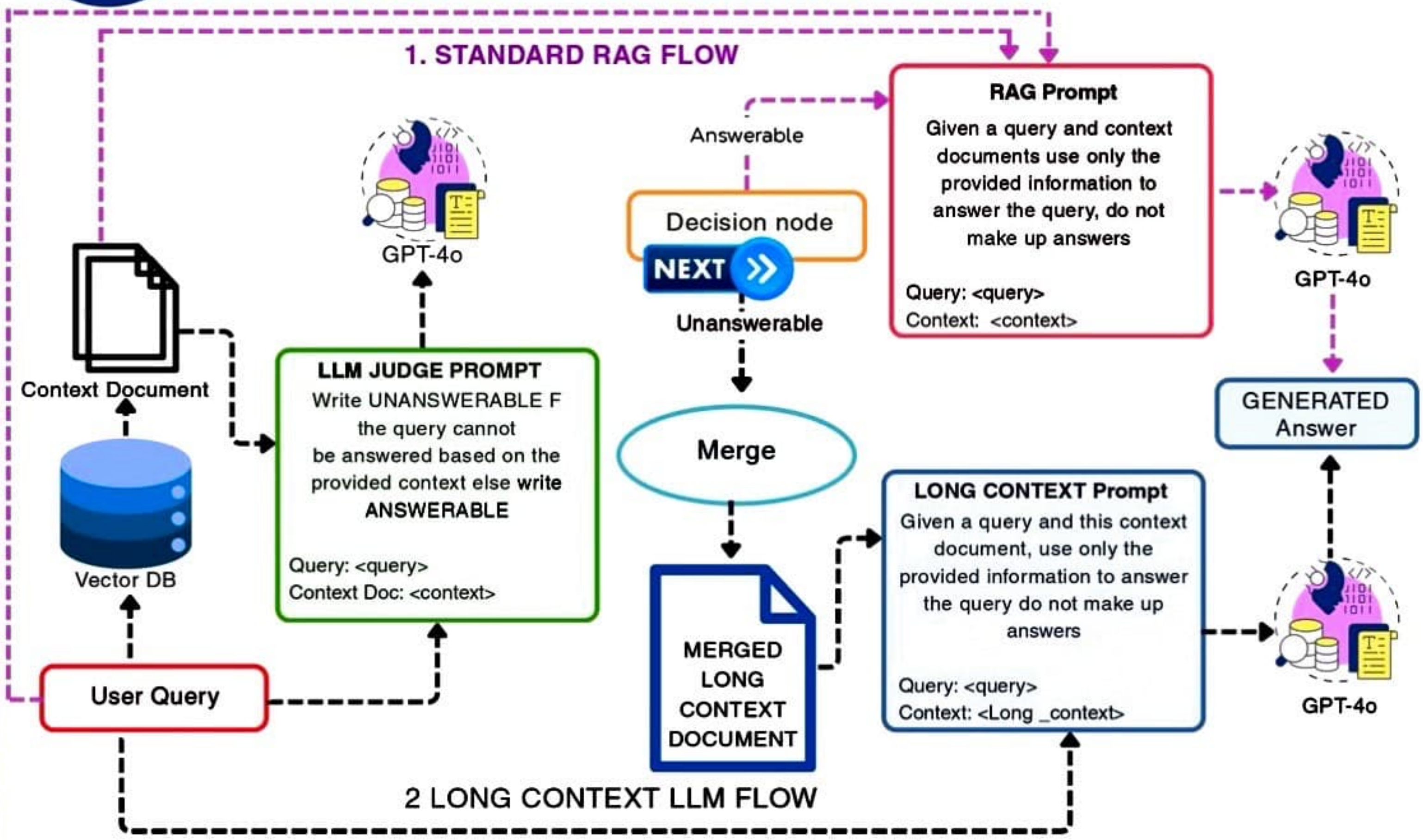
ChatGpt\ LLM

REPHRASED QUERY

- **Iterative Refinement:** Uses feedback loops to revise retrieved documents or answers multiple times.
- **Error Detection:** Identifies inaccuracies, contradictions, or gaps in initial retrievals.
- **Dynamic Corrections:** Automatically fetches supplementary documents to fill knowledge gaps.
- **Multi-Fact Handling**: Optimized for queries requiring synthesis of multiple facts (e.g., "Compare X and Y across A, B, C").
- **Confidence Scoring**: Flags low-confidence responses for re-retrieval or human review.
- **Use Case:** Legal research, medical diagnoses, or technical troubleshooting.

NEXT »

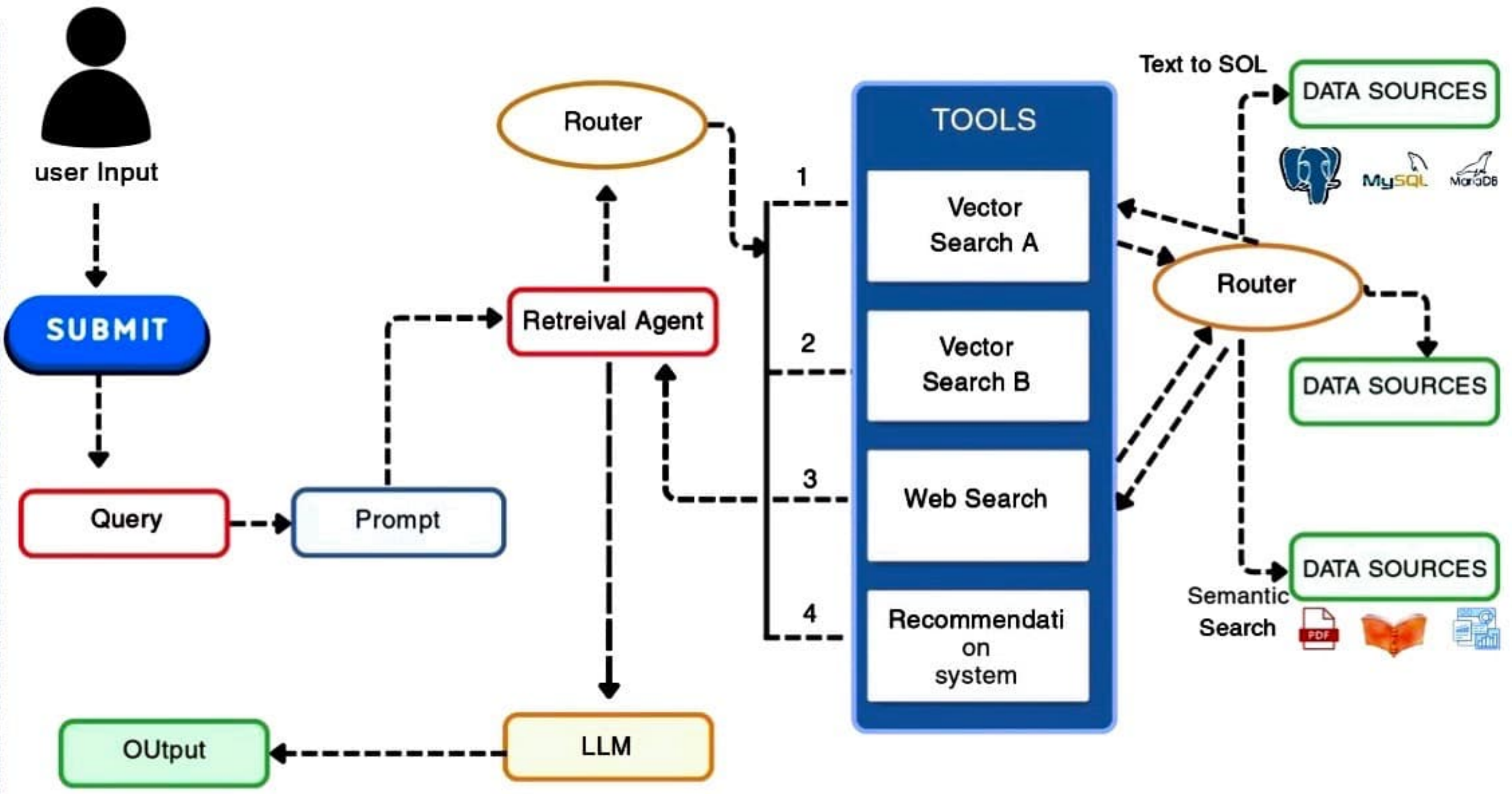
2 Self Route Agentic RAG


1. STANDARD RAG FLOW
GPT-4o
Answerable
Decision node
NEXT
Unanswerable
RAG Prompt
Given a query and context documents use only the provided information to answer the query, do not make up answers
Query: <query>
Context: <context>
GPT-4o
GENERATED Answer
Context Document
Vector DB
LLM JUDGE PROMPT
Write UNANSWERABLE F the query cannot be answered based on the provided context else write ANSWERABLE
Query: <query>
Context Doc: <context>
Merge
MERGED LONG CONTEXT DOCUMENT
LONG CONTEXT Prompt
Given a query and this context document, use only the provided information to answer the query do not make up answers
Query: <query>
Context: <Long _context>
GPT-4o
User Query
2 LONG CONTEXT LLM FLOW

- **Dynamic Path Selection**: Chooses retrieval methods (e.g., vector search, keyword, hybrid) per query.
- **Context-Aware Routing**: Adapts to query complexity (e.g., simple fact vs. multi-step reasoning).
- **Source Optimization**: Prioritizes databases or APIs most likely to contain relevant data.
- **Cost-Efficiency**: Avoids expensive retrievals when simpler methods suffice.
- **Fallback Mechanisms**: Switches strategies if initial retrievals fail.
- **Use Case**: Enterprise search across heterogeneous data lakes.

NEXT

3 Agentic RAG routers

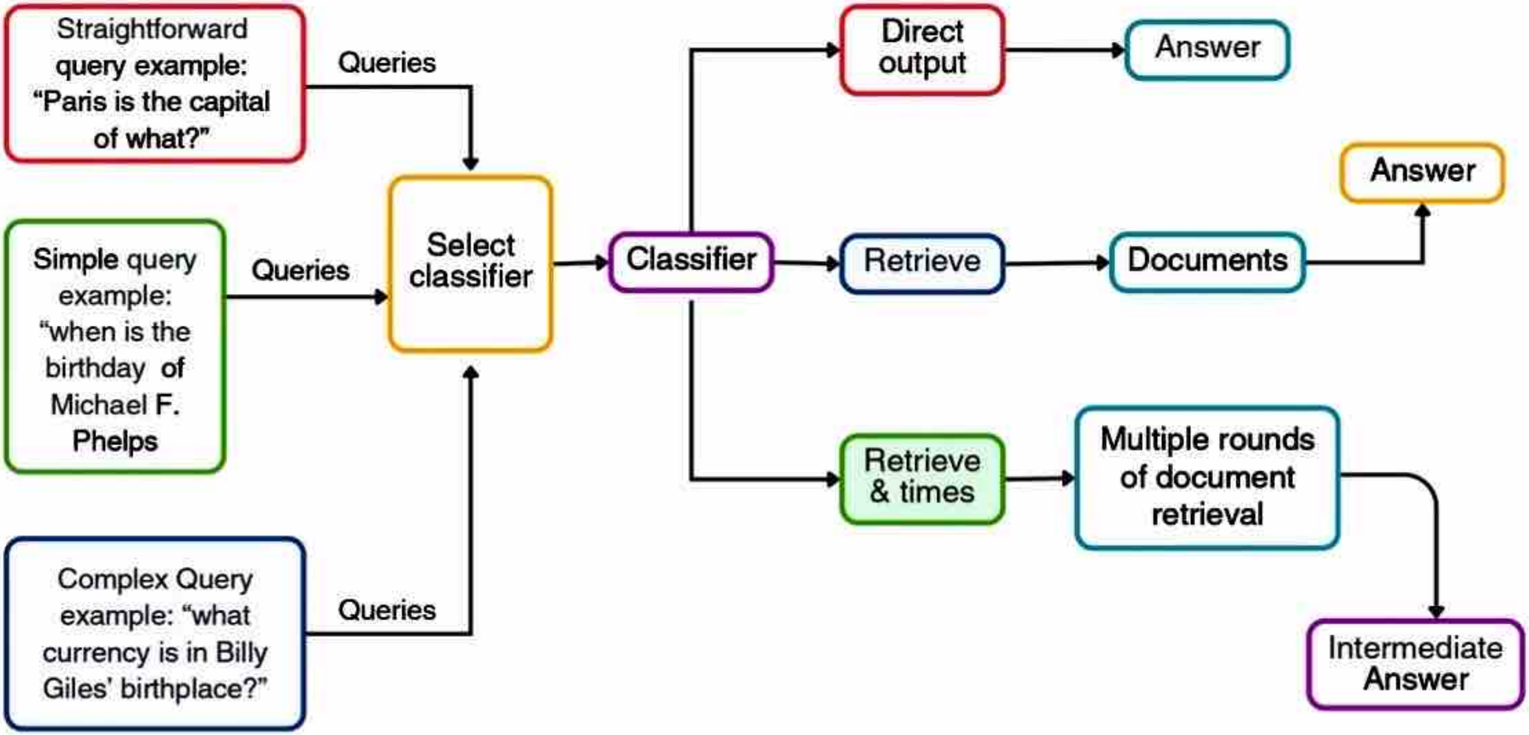


- **Specialized Retrievers**: Routes queries to domain-specific models (e.g., scientific vs. legal retrievers).
- **Method Optimization:** Selects between dense/sparse retrieval based on query semantics.
- **Load Balancing:** Distributes queries across retrievers to avoid bottlenecks.
- **Hybrid Orchestration**: Combines LLM-based routing with heuristic rules.
- **Real-Time Adjustments**: Monitors performance to reroute underperforming paths.
- **Use Case**: Customer support systems with mixed FAQs and docs.

NEXT »

4 Adaptive RAG

Adaptative Approach

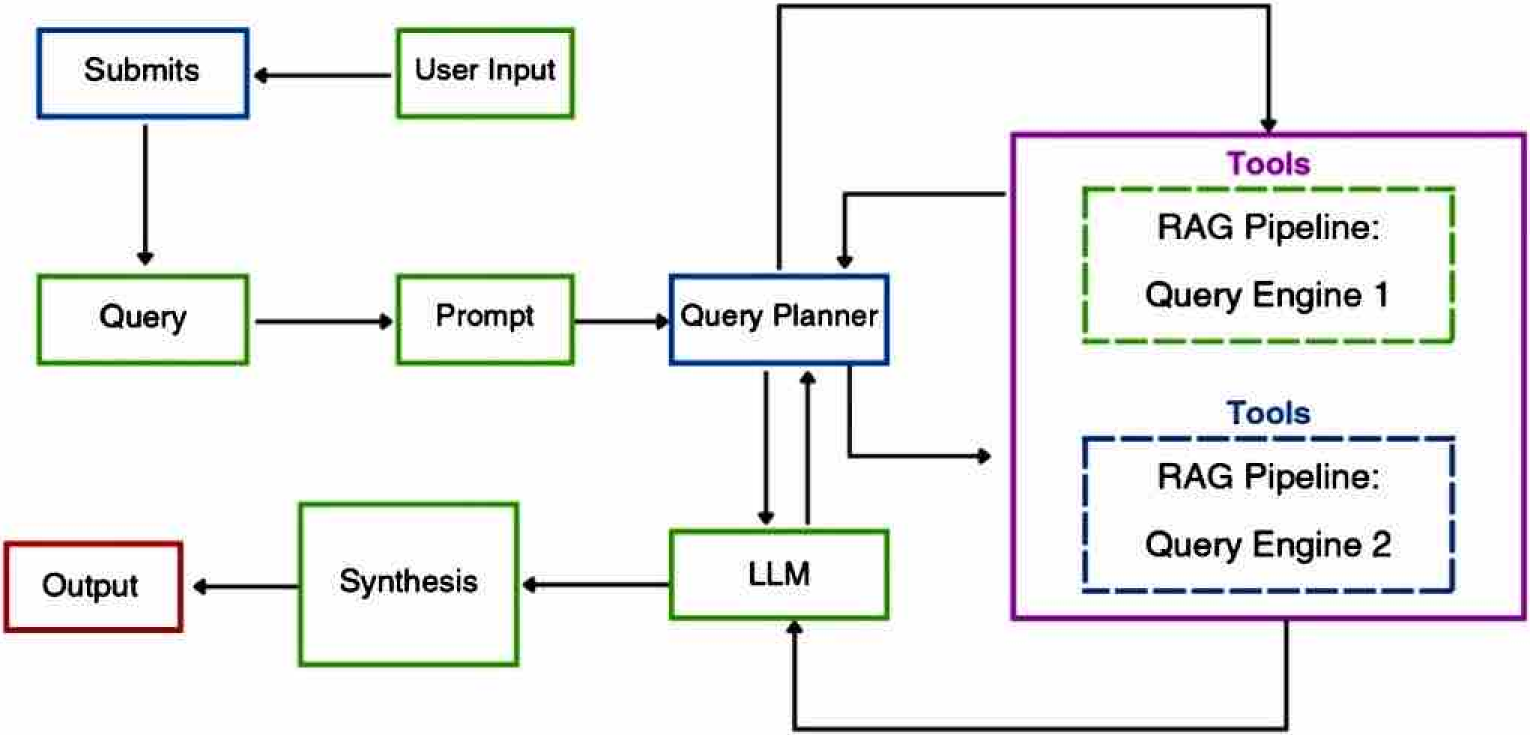


Straightforward query example: “Paris is the capital of what?”
Simple query example: “when is the birthday of Michael F. Phelps
Complex Query example: “what currency is in Billy Giles’ birthplace?”
Queries
Queries
Queries
Select classifier
Classifier
Direct output
Answer
Retrieve
Documents
Answer
Retrieve & times
Multiple rounds of document retrieval
Intermediate Answer

- **Dynamic Depth Control**: Adjusts retrieval scope (e.g., # of documents) based on query difficulty.
- **Generation-Length Tuning**: Expands/shortens answers per user needs (e.g., summary vs. detailed report).
- **Metric-Driven**: Optimizes for latency, accuracy, or cost via real-time feedback.
- **Workload Adaptation**: Scales retrieval intensity during peak vs. low-traffic periods.
- **User Preference Learning**: Customizes outputs based on historical interactions.
- **Use Case:** Streaming analytics or personalized recommendation engines.

NEXT »

5 Query planning agentic RAG

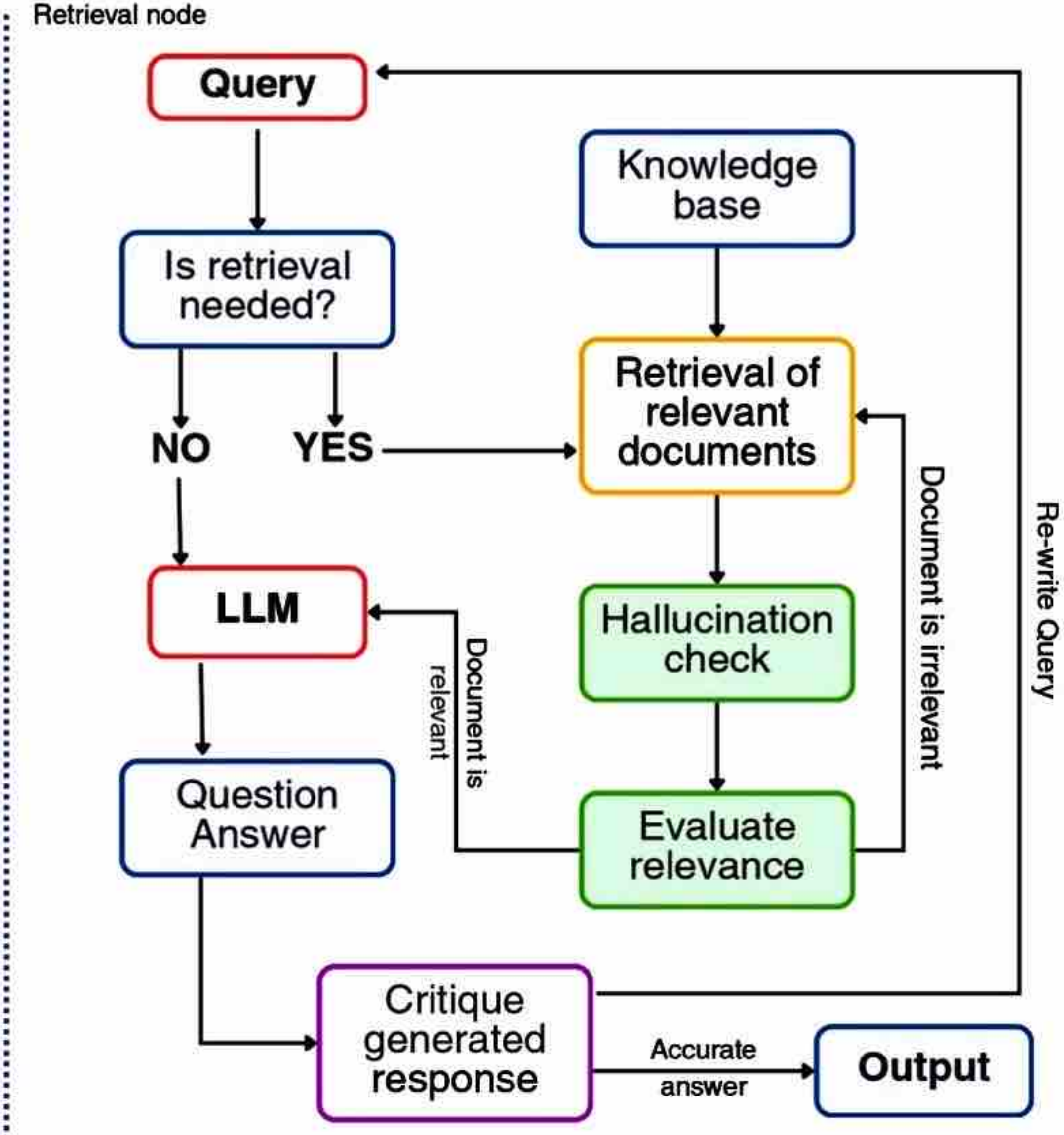


- **Query Decomposition:** Splits complex questions into sub-queries (e.g., "What is X? How does X impact Y?").
- **Execution Planning**: Sequences retrievals and syntheses logically to minimize redundancy.
- **Multi-Hop Support**: Chains retrievals for "A depends on B" scenarios.
- **Intermediate Validation:** Checks sub-query results before proceeding.
- **Parallelization:** Runs independent sub-queries concurrently.
- **Use Case**: Research assistance or business intelligence.

NEXT »

6

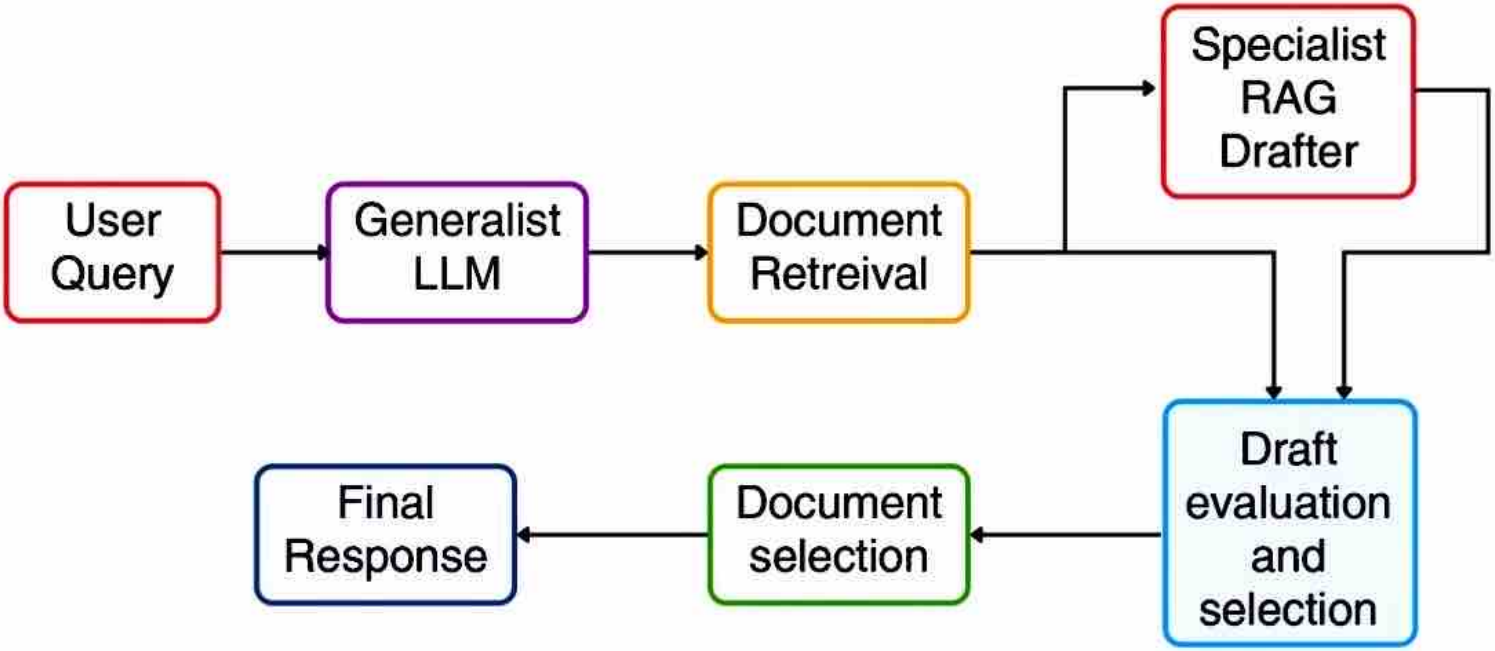
Self reflective RAG



- **Output Auditing:** Validates answers against retrieved docs for consistency.
- **Hallucination Mitigation**: Rejects unsupported claims or low-confidence generations.
- **Retry Mechanisms:** Auto-triggers re-retrieval if self-evaluation fails.
- **Citation Tracking:** Ensures verifiability by linking claims to sources.
- **User Alerts:** Flags uncertain answers for manual review.
- **Use Case**: Journalism, academic writing, or compliance documentation.

NEXT »

7 Speculative RAG



- **Predictive Retrieval**: Pre-fetches documents likely to be needed based on query patterns.
- **Latency Reduction**: Executes retrievals in parallel while the user finishes typing.
- **Context Anticipation:** Uses session history to guess follow-up questions.
- **Cache Optimization**: Stores frequently accessed docs for rapid reuse.
- **Fallback Readiness:** Keeps backup retrievals active for unexpected pivots.
- **Use Case:** Real-time chatbots or voice assistants.