25 RAG Architectures

=

25 Design Patterns

Because one RAG architecture is never enough

Chandra Sekhar
AI Engineer & Educator

25 Types of RAG Overview

1 FOUNDATION

- Standard RAG
- Corrective RAG
- Self RAG

2 ADVANCED

- Speculative RAG
- Fusion RAG
- Agentic RAG
- Adaptive RAG

3 SPECIALIZED

- REFEED
- REALM
- RAPTOR
- REVEAL
- REACT
- REPLUG

4 OPTIMIZED

- MEMO RAG
- ATLAS
- RETRO
- AUTO RAG
- CORAG
- EACO-RAG

5 DOMAIN-SPECIFIC

- RULE RAG
- CORAL
- Iterative RAG
- ConTReGen
- CRAT

Follow Chandra for More

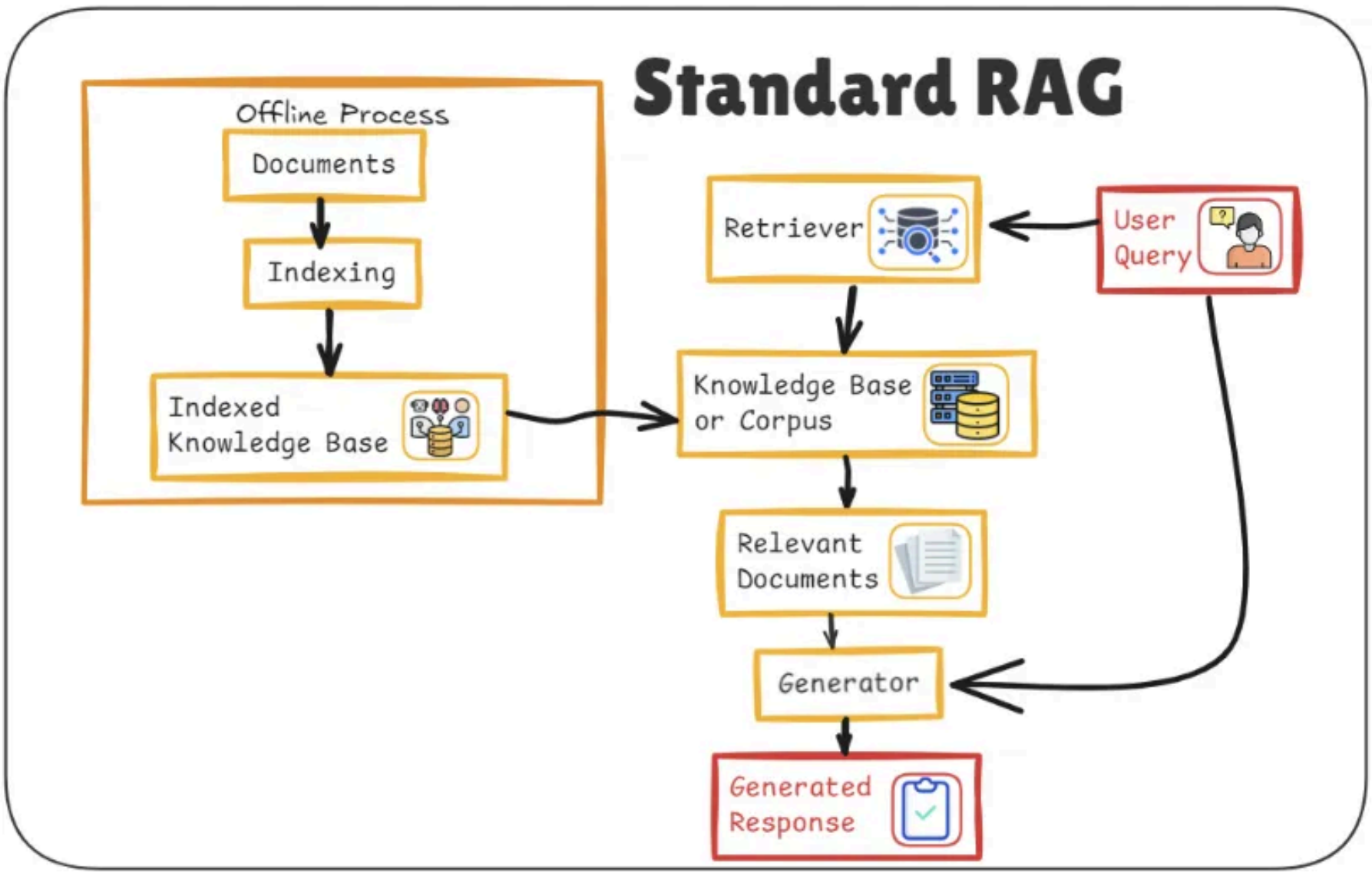
Repost to help others in your network

1 Standard RAG

Combines retrieval with large language models for accurate, context-aware responses.

KEY FEATURES:

- Breaks documents into chunks for efficient information retrieval
- Aims for 1-2 second response times for real-time use
- Enhances answer quality by leveraging external data sources

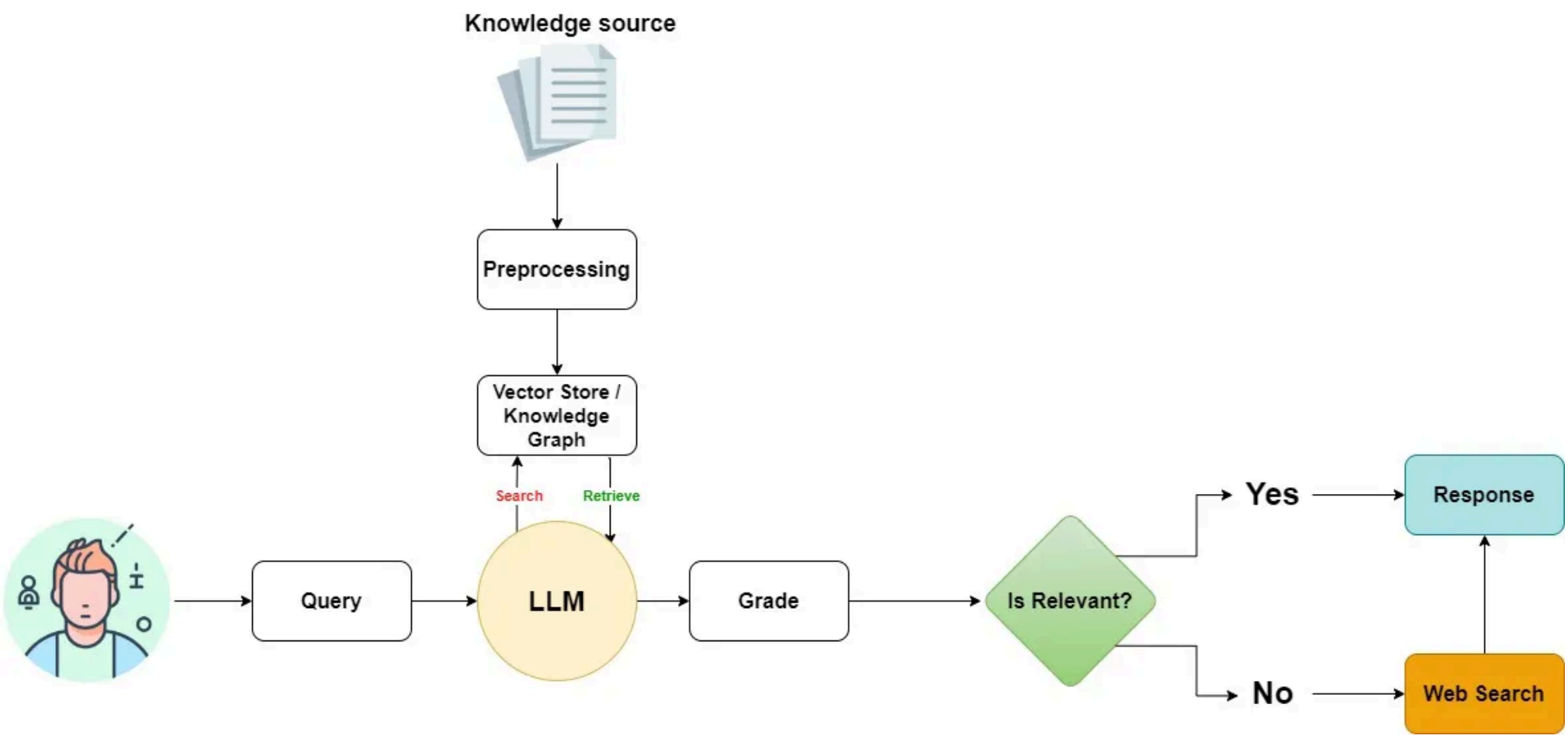


2 Corrective RAG

RAG with validation & correction. Retrieved content is graded, compared against trusted rules or datasets. If inaccurate → system re-searches (web or KB) → Answer is corrected automatically.

BEST FOR:

- ✔ Compliance systems
- ✔ Enterprise QA
- ✔ High-accuracy applications

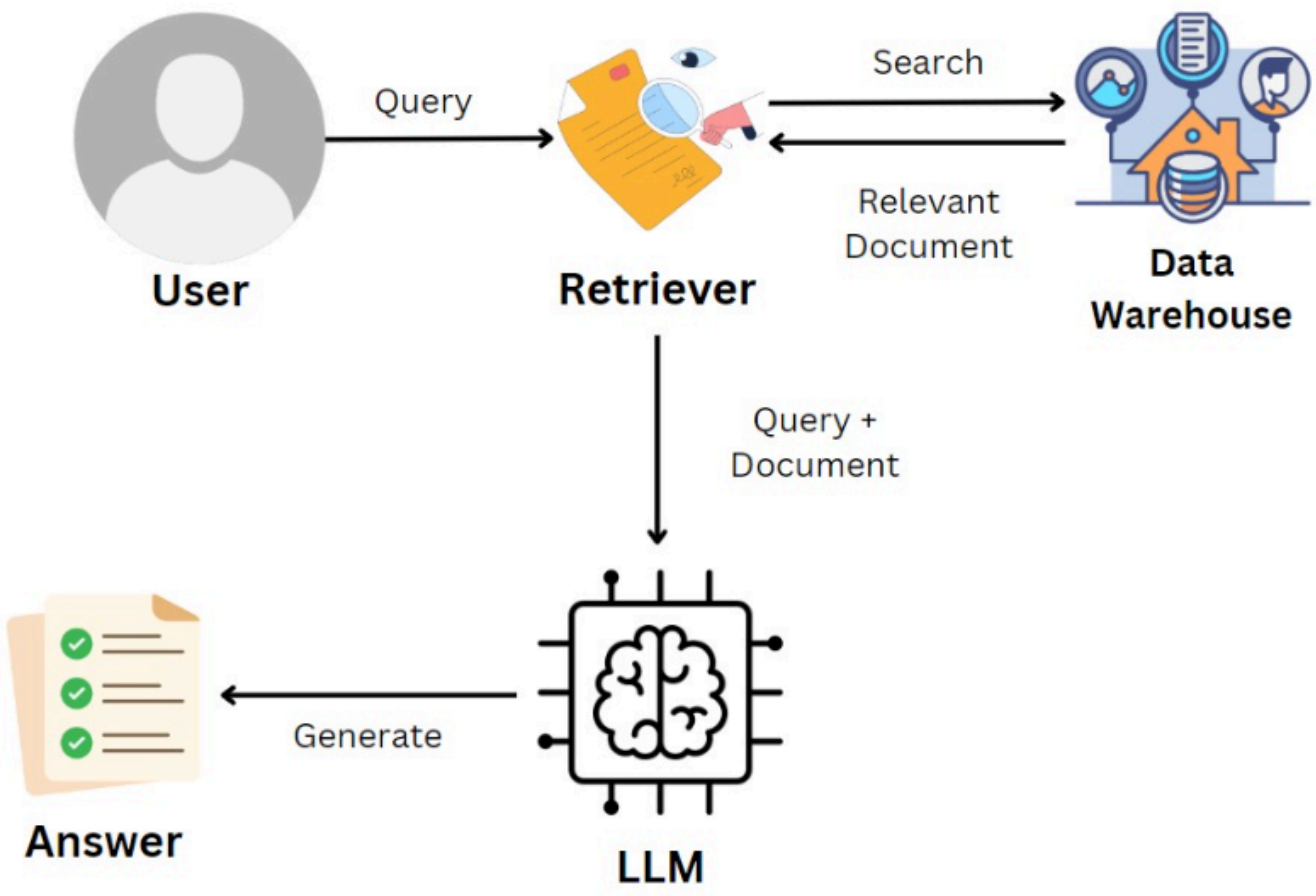


3 Speculative RAG

Uses a small specialist model for drafting and a larger generalist model for verification, ensuring efficiency and accuracy.

KEY FEATURES:

- Parallel Drafting: Speeds up responses by generating multiple drafts simultaneously
- Superior Accuracy: Outperforms standard RAG systems
- Efficient Processing: Offloads complex tasks to specialized models, reducing computational load

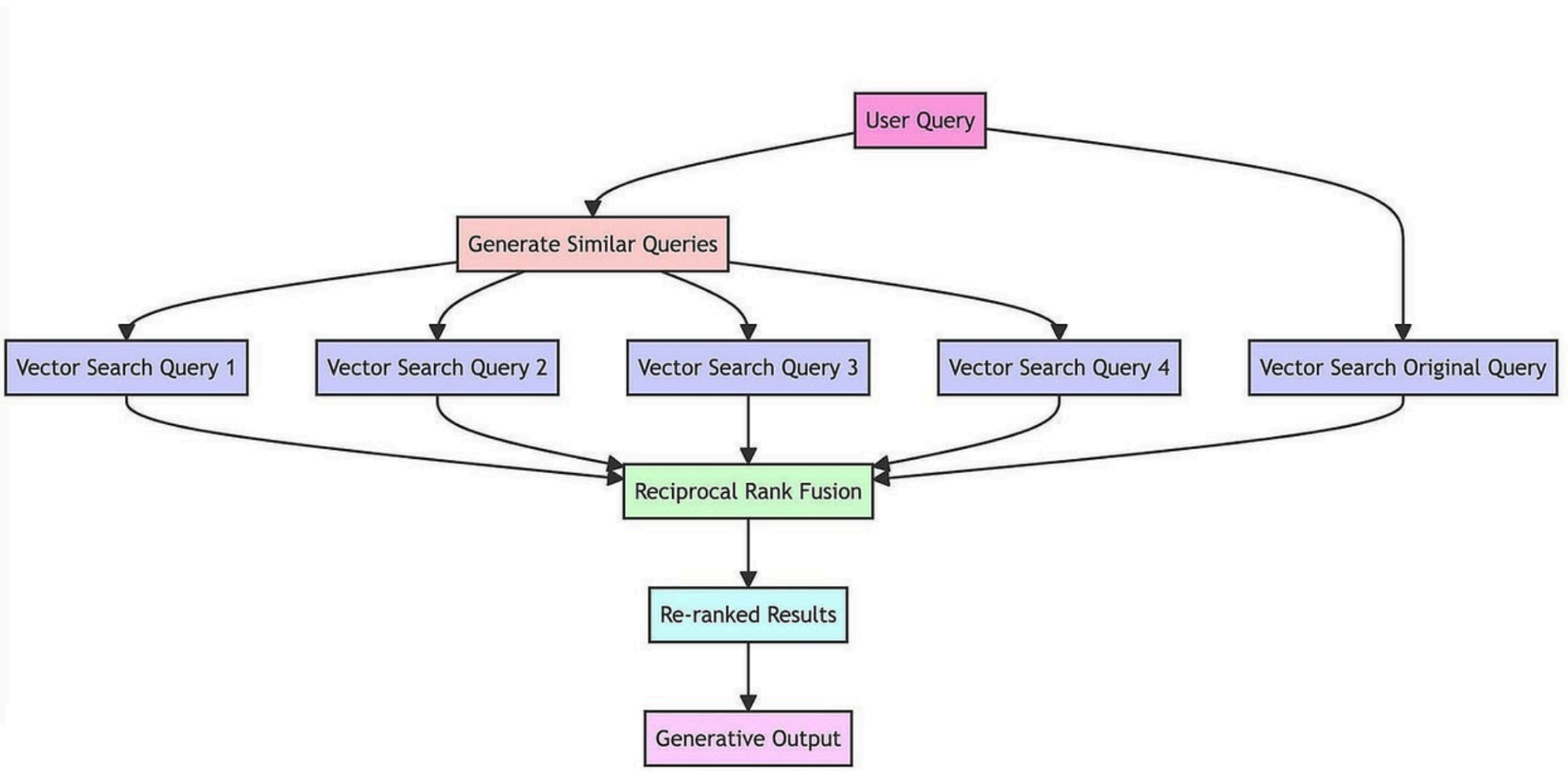


4 Fusion RAG

Integrates multiple retrieval methods and data sources for enhanced response quality.

KEY FEATURES:

- Provides comprehensive answers by leveraging diverse data inputs
- Increases system resilience by reducing dependence on a single source
- Adapts retrieval strategies dynamically based on query context

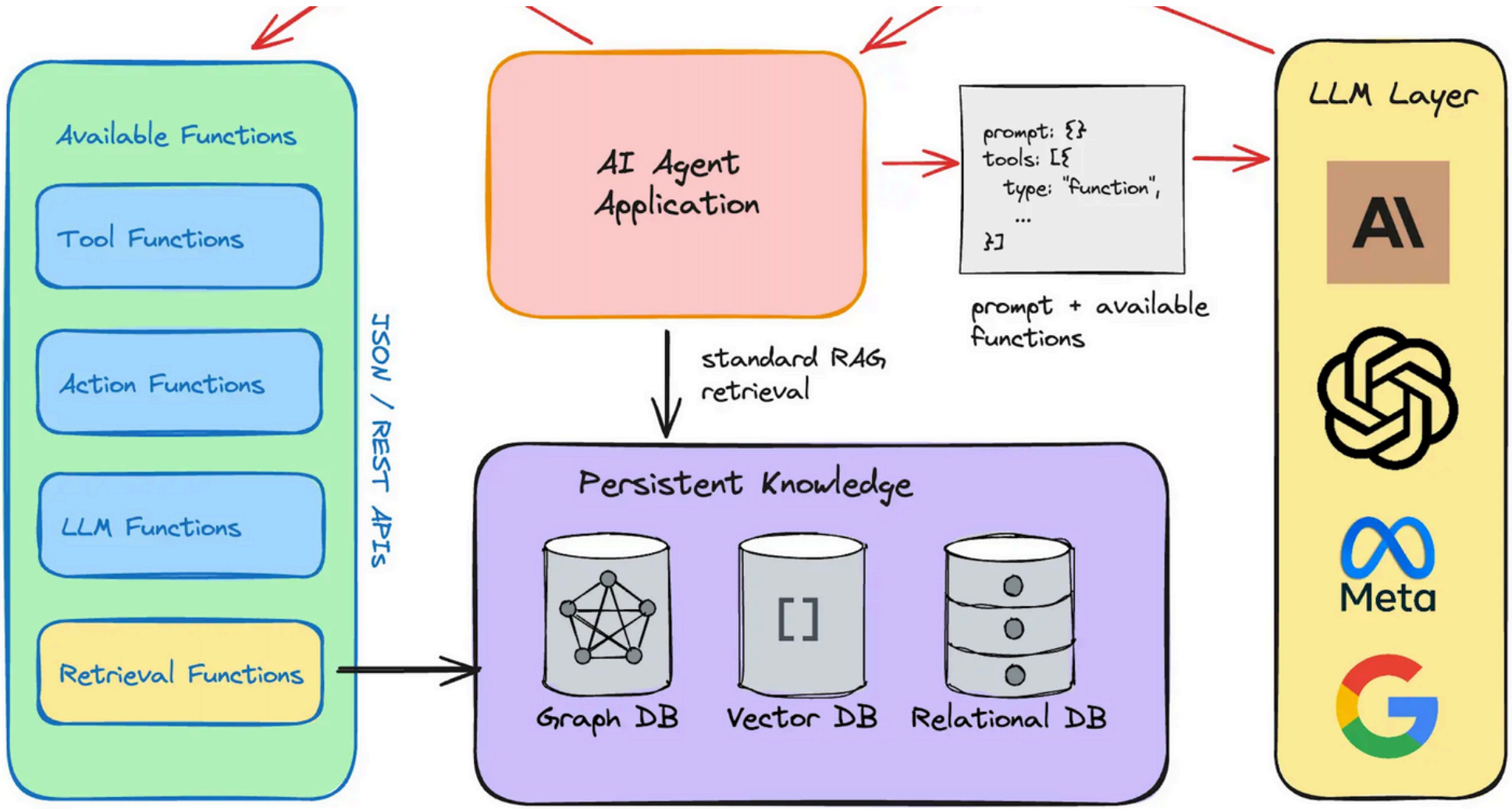


5 Agentic RAG

Uses adaptive agents for real-time strategy adjustments in information retrieval.

KEY FEATURES:

- Accurately interprets user intent for relevant, trustworthy responses
- Modular design enables easy integration of new data sources and features
- Enhances parallel processing and performance on complex tasks by running agents concurrently

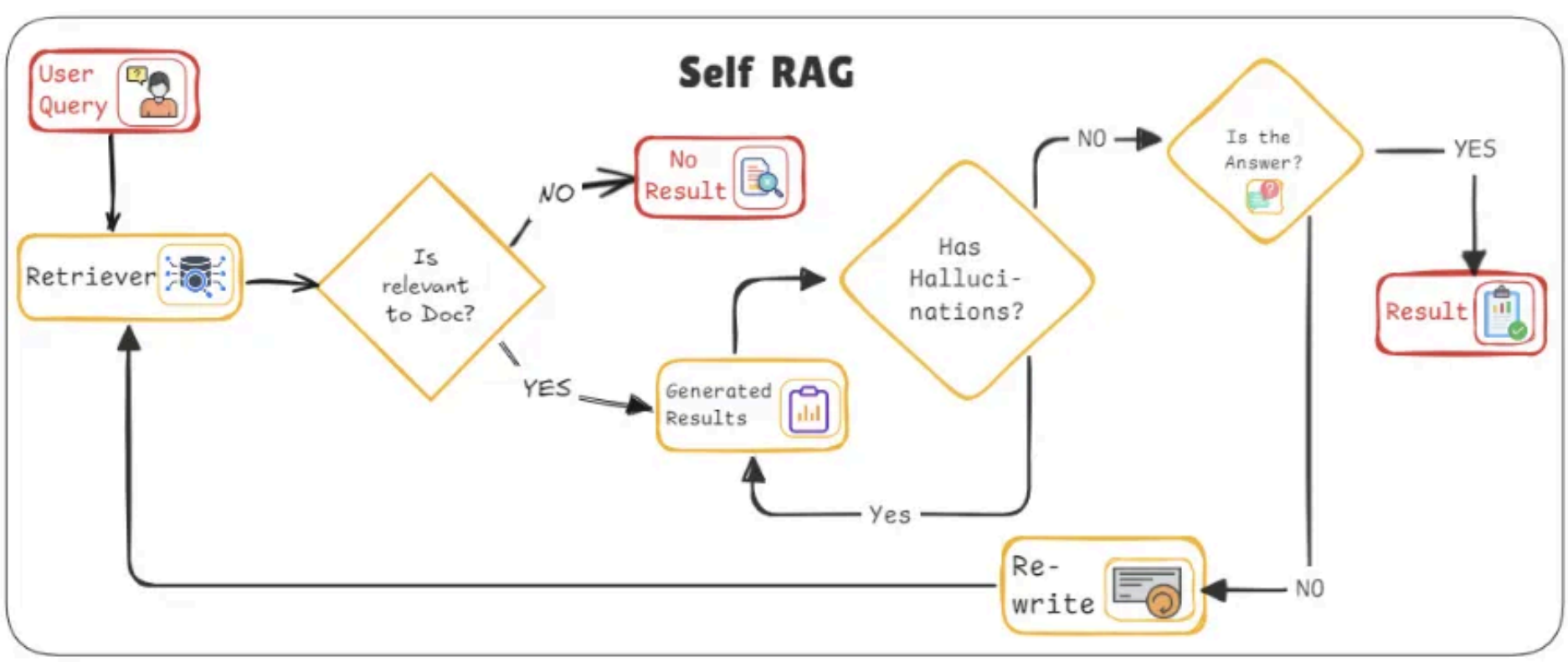


6 Self RAG

Uses the model's own outputs as retrieval candidates for better contextual relevance.

KEY FEATURES:

- Refines responses iteratively, improving consistency and coherence
- Grounds responses in prior outputs for increased accuracy
- Adapts retrieval strategies based on the conversation's evolving context

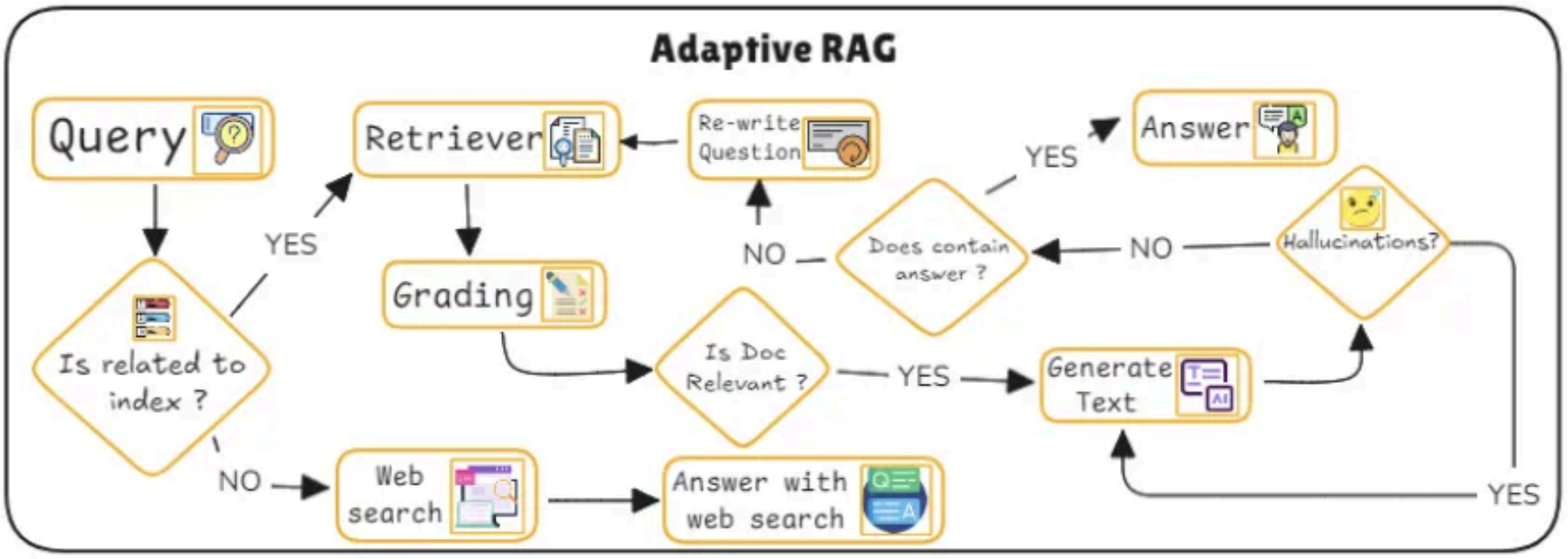


7 Adaptive RAG

Dynamically decides when to retrieve external knowledge, balancing internal and external knowledge.

KEY FEATURES:

- Uses confidence scores from the language model's internal states to assess retrieval necessity
- An honesty probe helps the model avoid hallucinations by aligning its output with its actual knowledge
- Reduces unnecessary retrievals, improving both efficiency and response accuracy

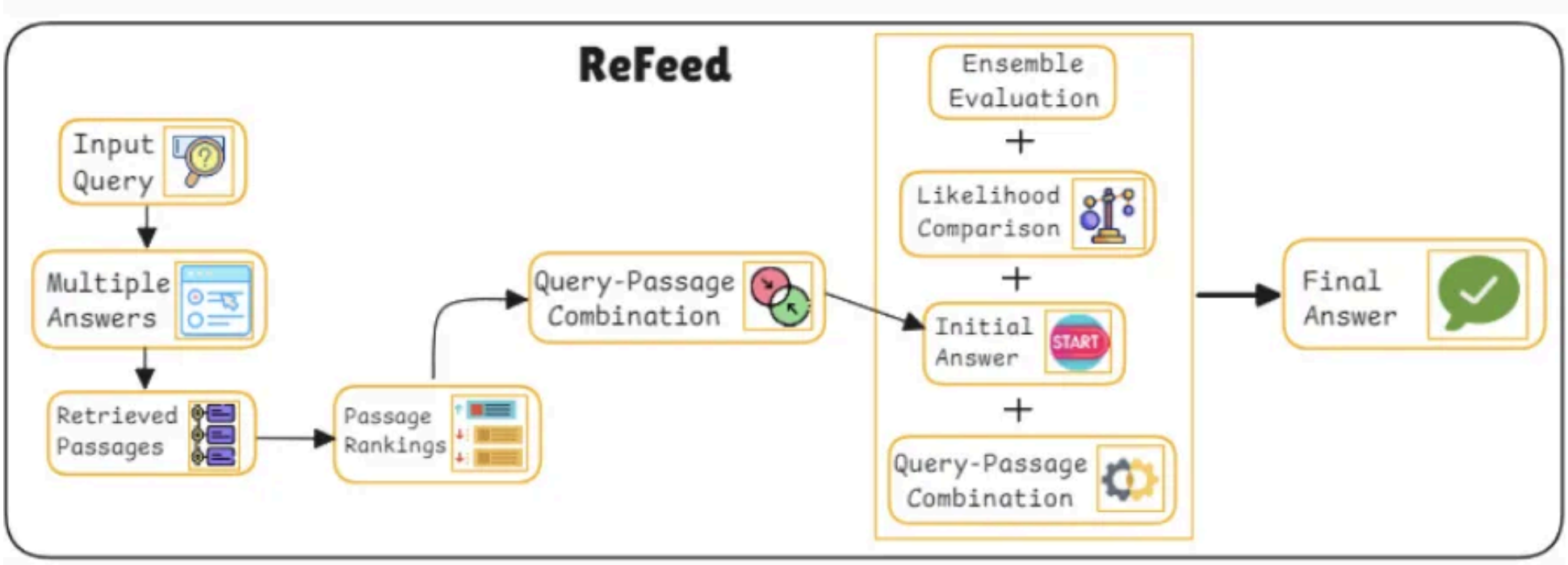


8 REFEED (Retrieval Feedback)

Refines model outputs using retrieval feedback without fine-tuning.

KEY FEATURES:

- Initial answers are improved by retrieving relevant documents and adjusting the response based on the new information
- Generates multiple answers to improve retrieval accuracy
- Combines pre- and post-retrieval outputs using a ranking system to enhance answer reliability

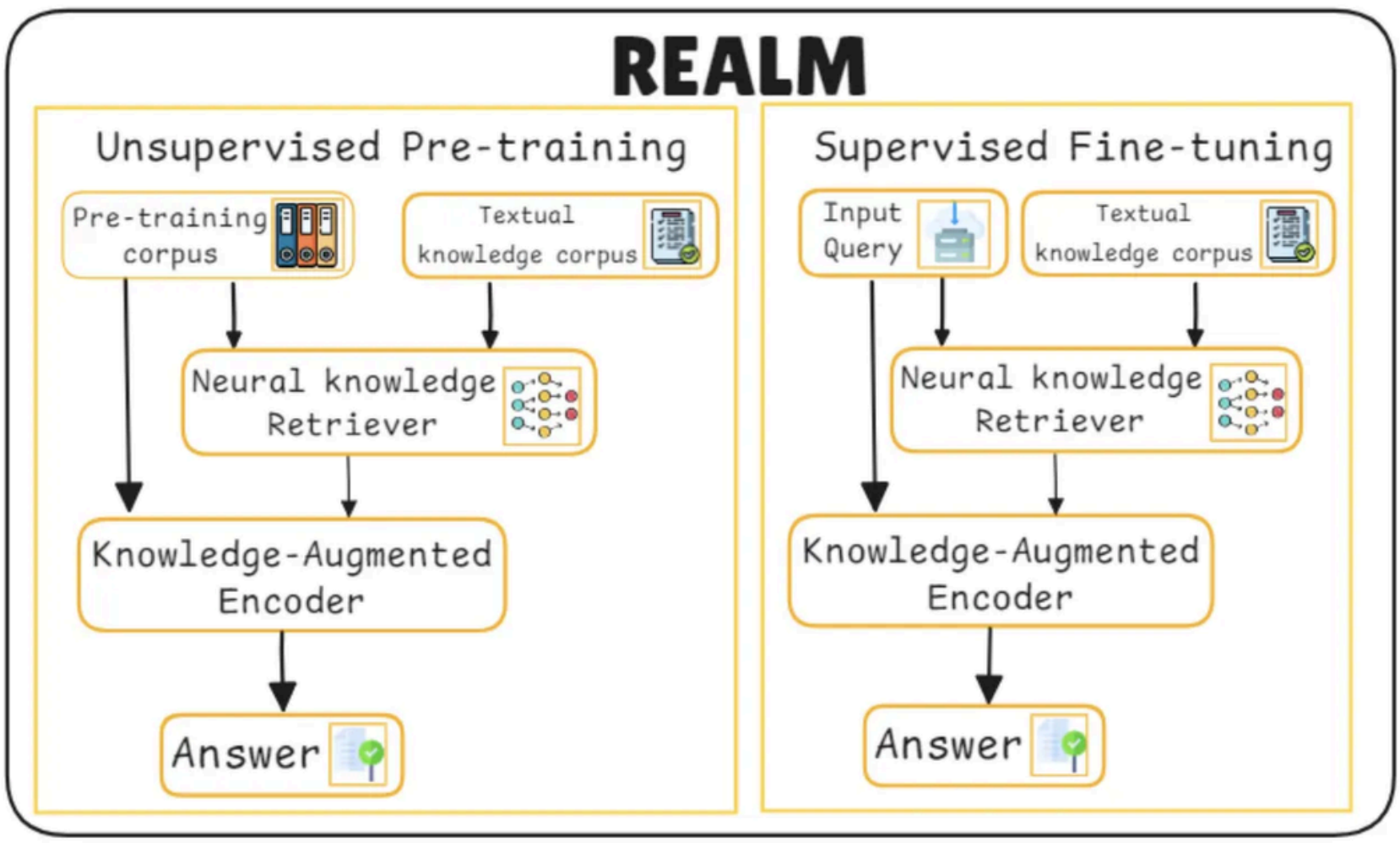


9 REALM

Retrieves relevant documents from large corpora like Wikipedia to enhance model predictions.

KEY FEATURES:

- The retriever is trained with masked language modeling, optimizing retrieval to improve prediction accuracy
- Uses Maximum Inner Product Search to efficiently find relevant documents from millions of candidates during training
- Outperforms previous models in Open-domain Question Answering by integrating external knowledge

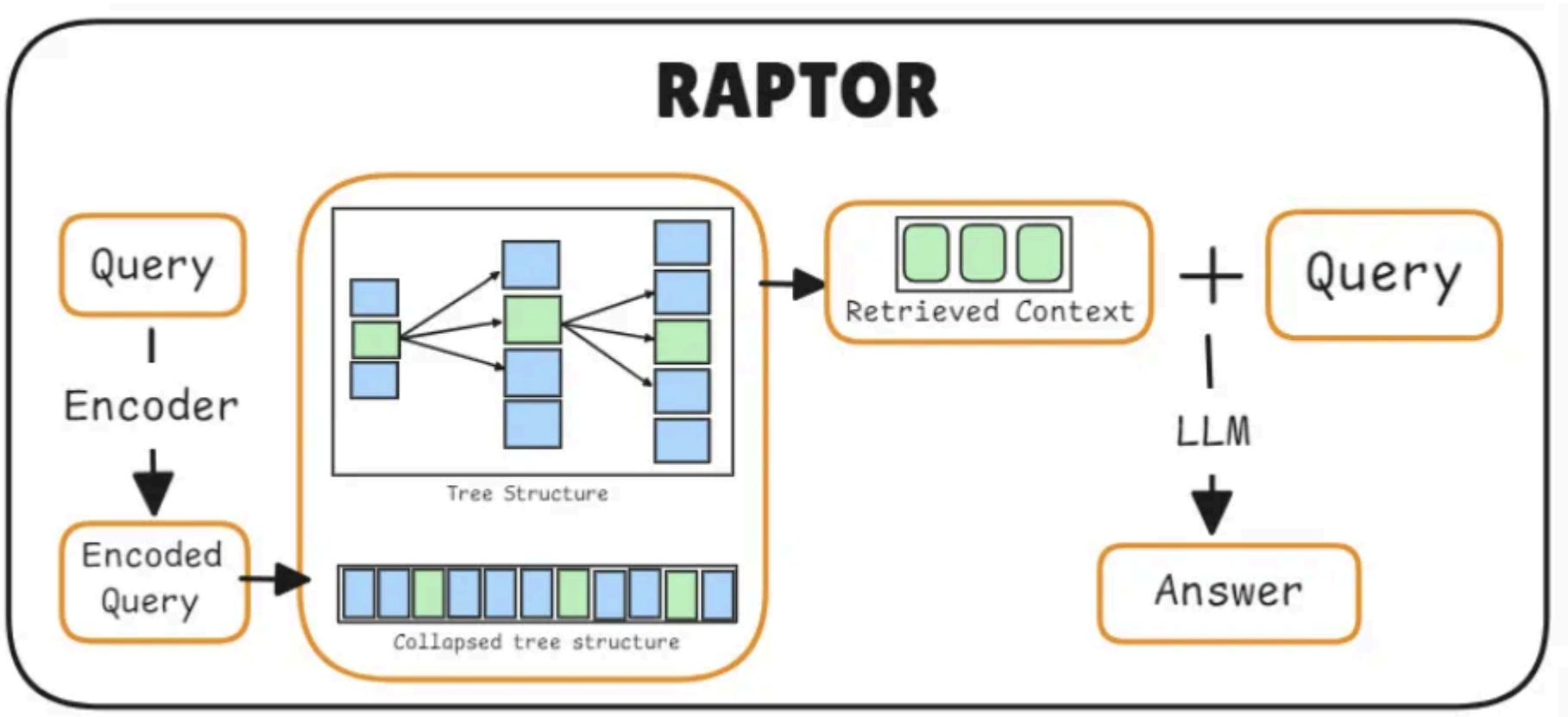


10 RAPTOR (Tree-Organized)

Builds a hierarchical tree by clustering and summarizing text recursively.

KEY FEATURES:

- Enables retrieval at different abstraction levels, combining broad themes with specific details
- Outperforms traditional methods in complex question-answering tasks
- Offers tree traversal and collapsed tree methods for efficient information retrieval

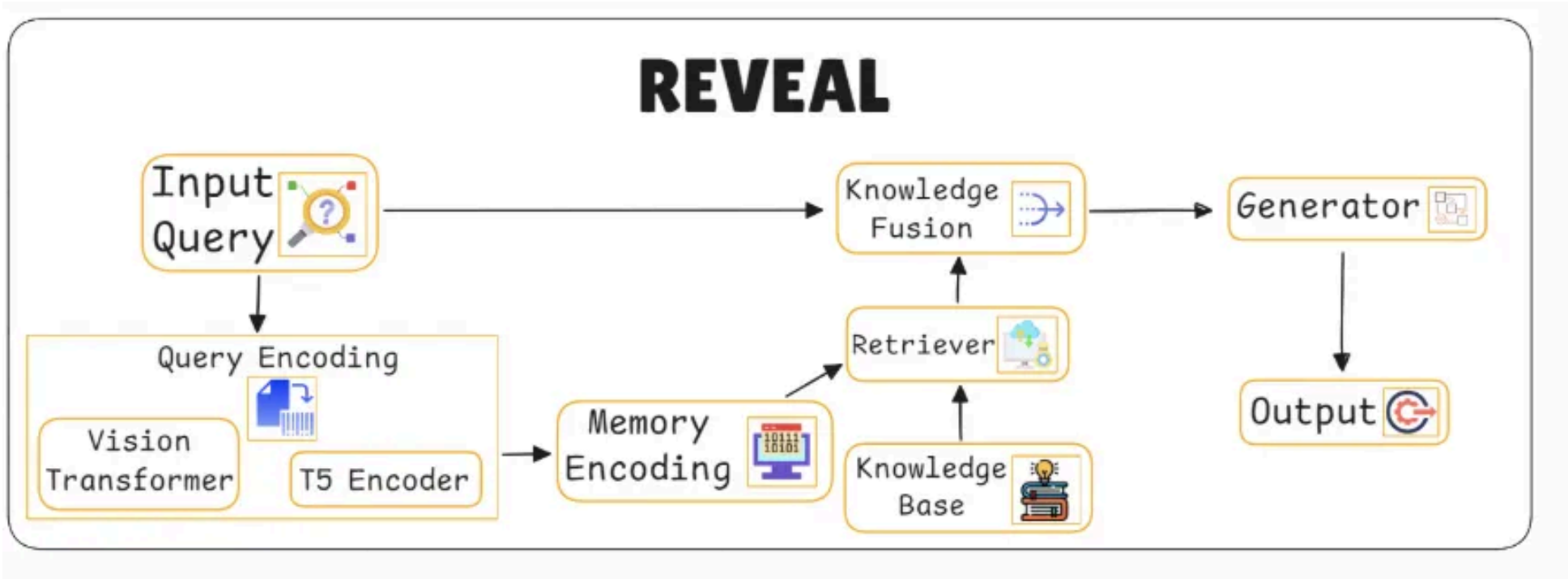


11 REVEAL (for Visual-Language Model)

Achieves strong performance across tasks with fewer training examples, making models efficient, adaptable, and responsive.

KEY FEATURES:

- Components include Vision Transformer, T5 Encoder, and Knowledge Fusion
- Designed for visual-language understanding tasks

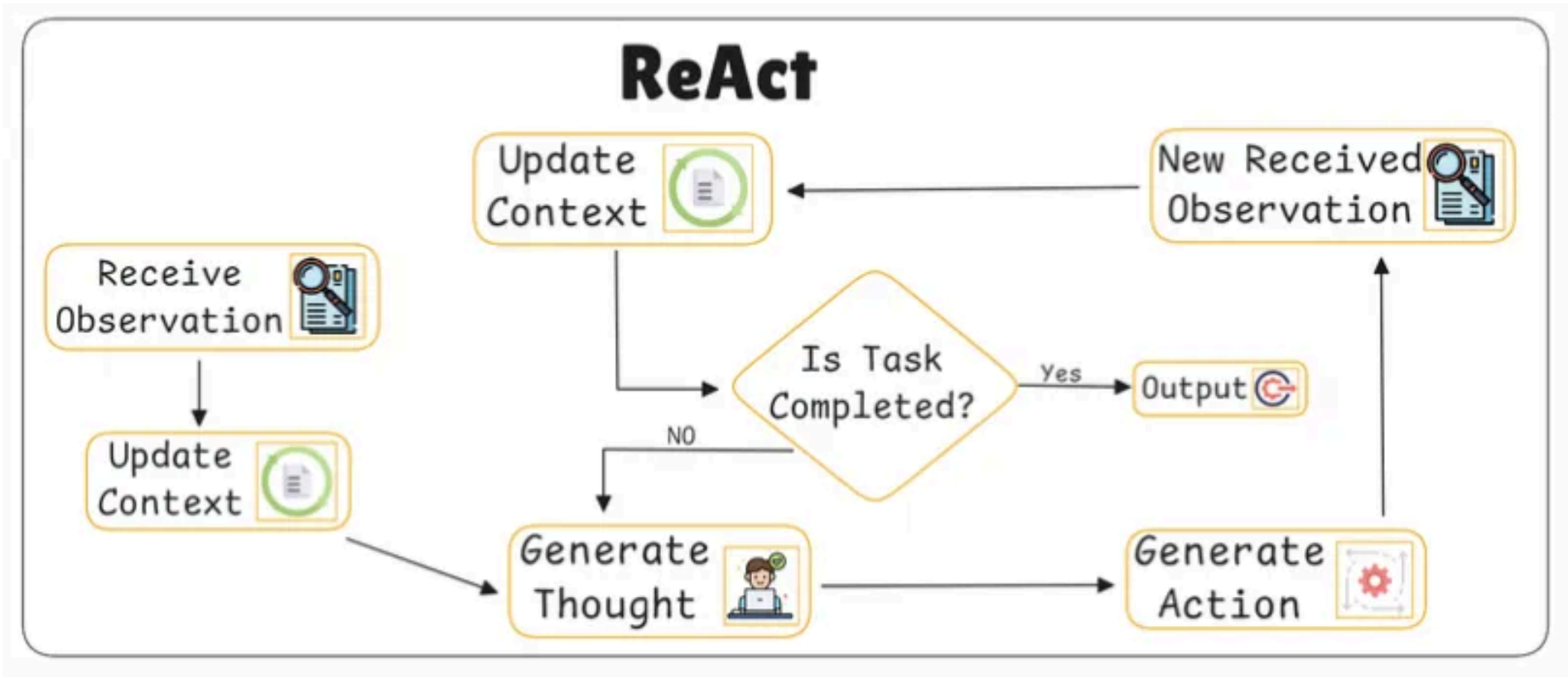


12 REACT

Combines reasoning and action, allowing models to interact with their environment.

KEY FEATURES:

- Maintains situational awareness by updating context with past actions and thoughts
- Generates task-aligned thoughts to guide logical decision-making
- Real-time feedback refines understanding, reducing errors and enhancing transparency
- Minimizes errors by grounding reasoning in real-world facts, reducing hallucinations
- Offers clear, human-like task-solving steps

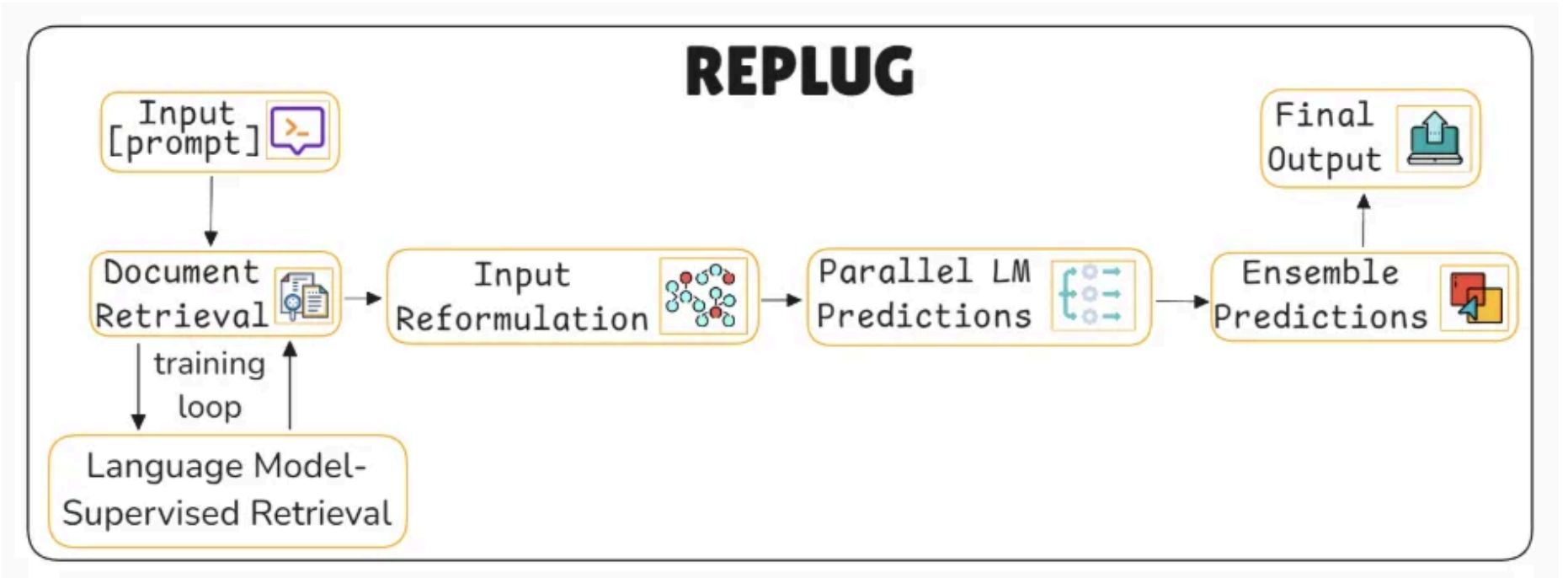


13 REPLUG (Retrieval Plugin)

Enhances LLMs by retrieving relevant external documents to improve prediction accuracy.

KEY FEATURES:

- Treats the language model as a fixed 'black box', prepending retrieved information to the input
- Flexible design works with existing models without modifications
- The retrieval component can be fine-tuned with model feedback, aligning better with the model's needs

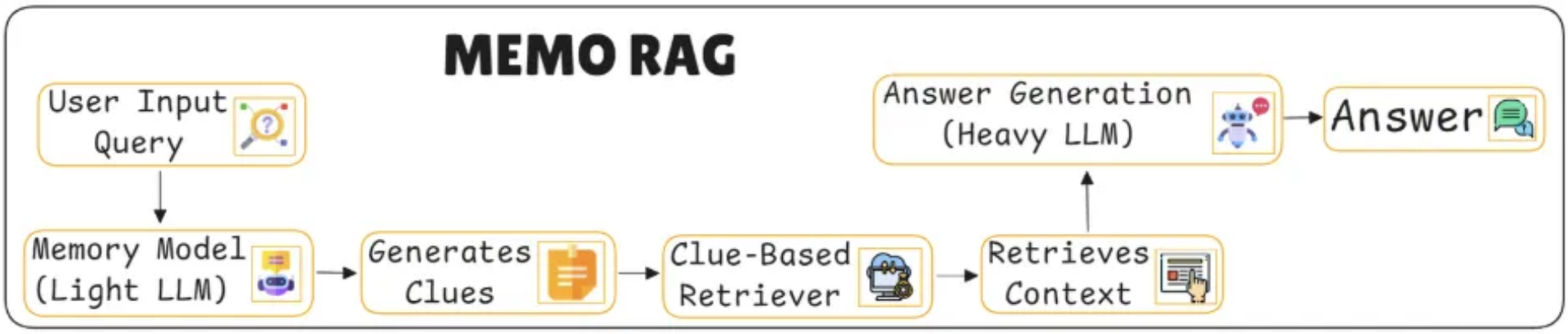


14 MEMO RAG

Combines memory and retrieval to handle complex queries.

KEY FEATURES:

- A memory model generates draft answers that guide the search for external information
- A retriever gathers relevant data which a more powerful language model uses to create a comprehensive final answer
- Helps manage ambiguous queries and efficiently process large amounts of information

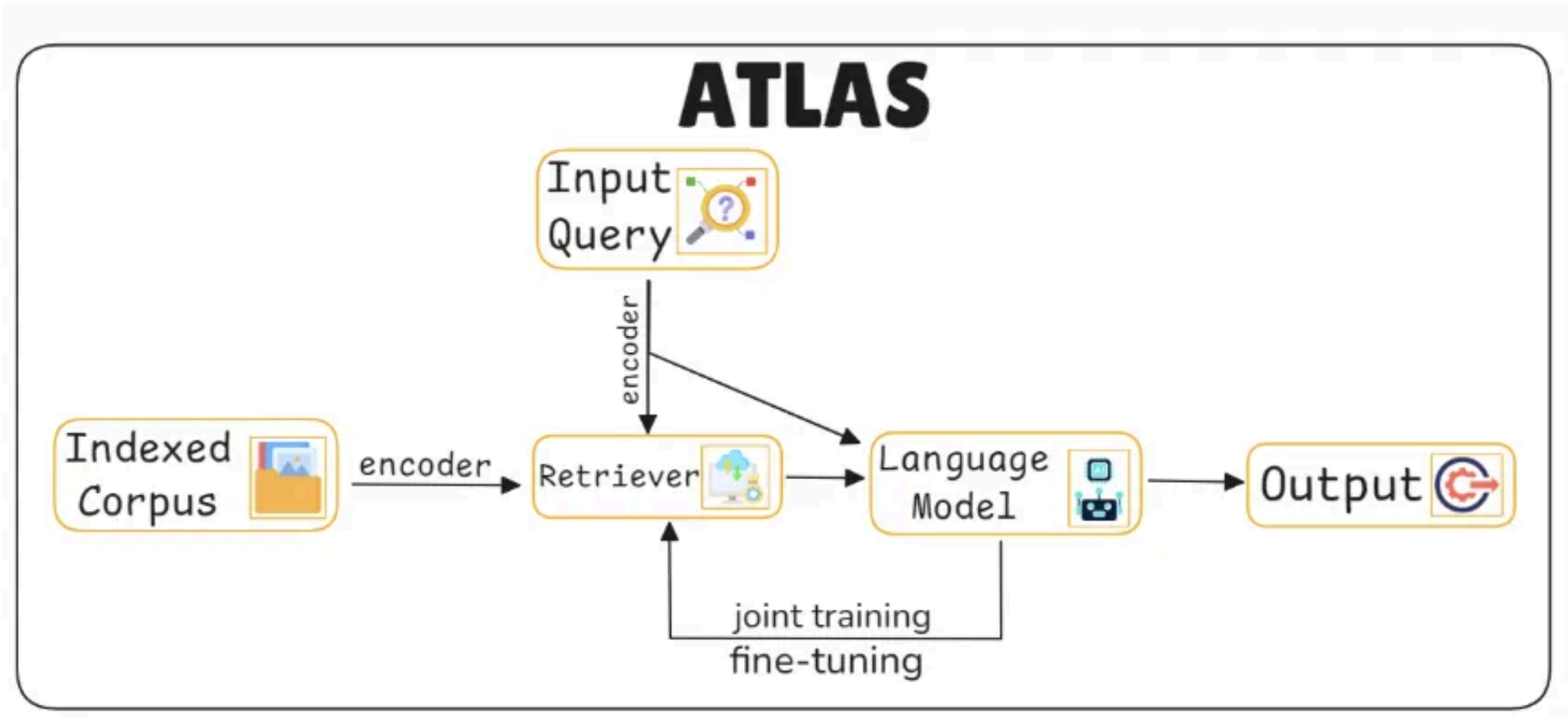


15 Attention-based RAG / ATLAS

ATLAS improves language models by retrieving external documents to enhance accuracy.

KEY FEATURES:

- Uses a dual-encoder retriever to identify the top-K relevant documents
- A Fusion-in-Decoder model integrates query and document information, reducing reliance on memorization
- The document index is updatable without retraining, ensuring it remains current

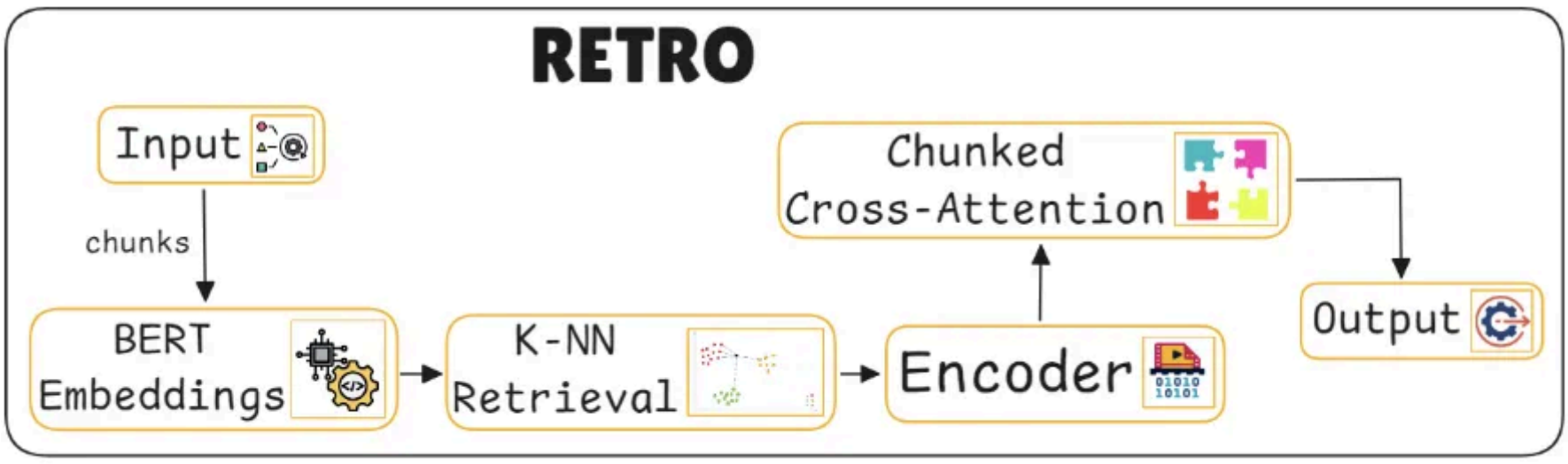


16 RETRO

Splits input text into chunks and retrieves similar information from a large text database.

KEY FEATURES:

- Uses pre-trained BERT embeddings to pull in relevant chunks
- Chunked cross-attention integrates these chunks, improving predictions without a major increase in model size
- Accesses extensive knowledge with lower computational demands than larger models

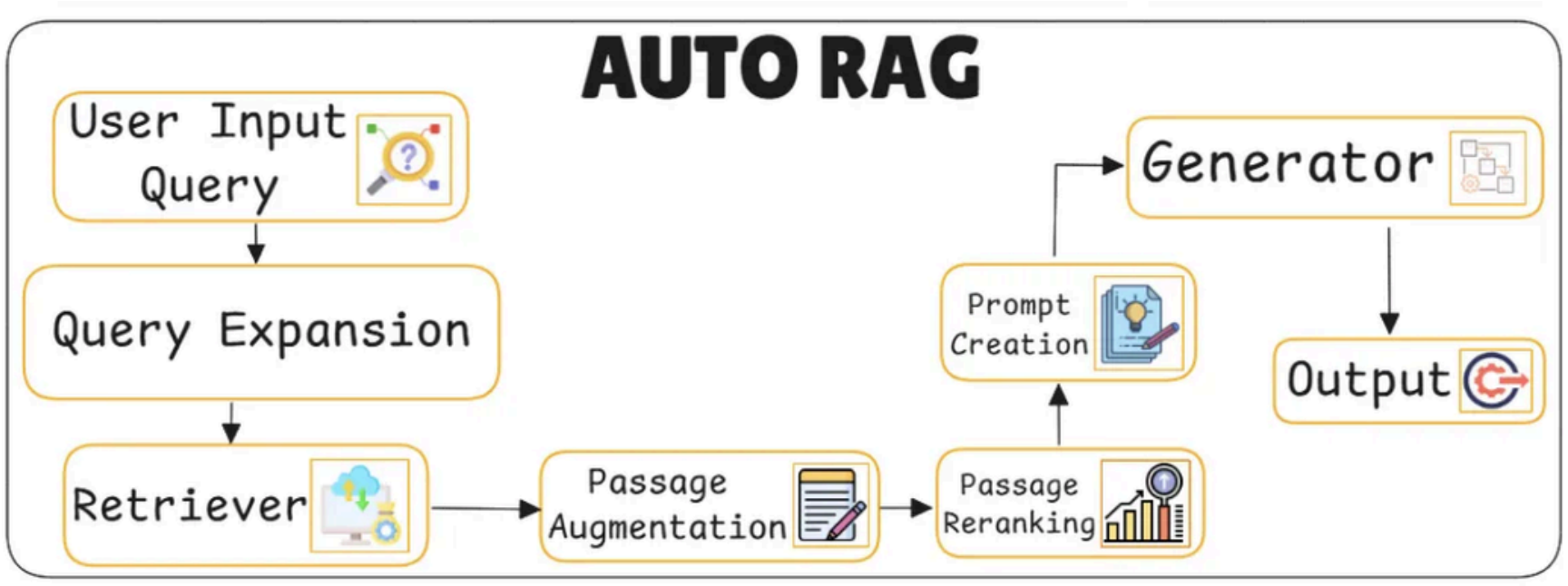


17 AUTO RAG

Automates optimization for RAG systems.

KEY FEATURES:

- Evaluates modules like query expansion, retrieval, and reranking for best performance
- Uses a modular, node-based structure to test various configurations
- A greedy optimization approach enhances efficiency across different datasets

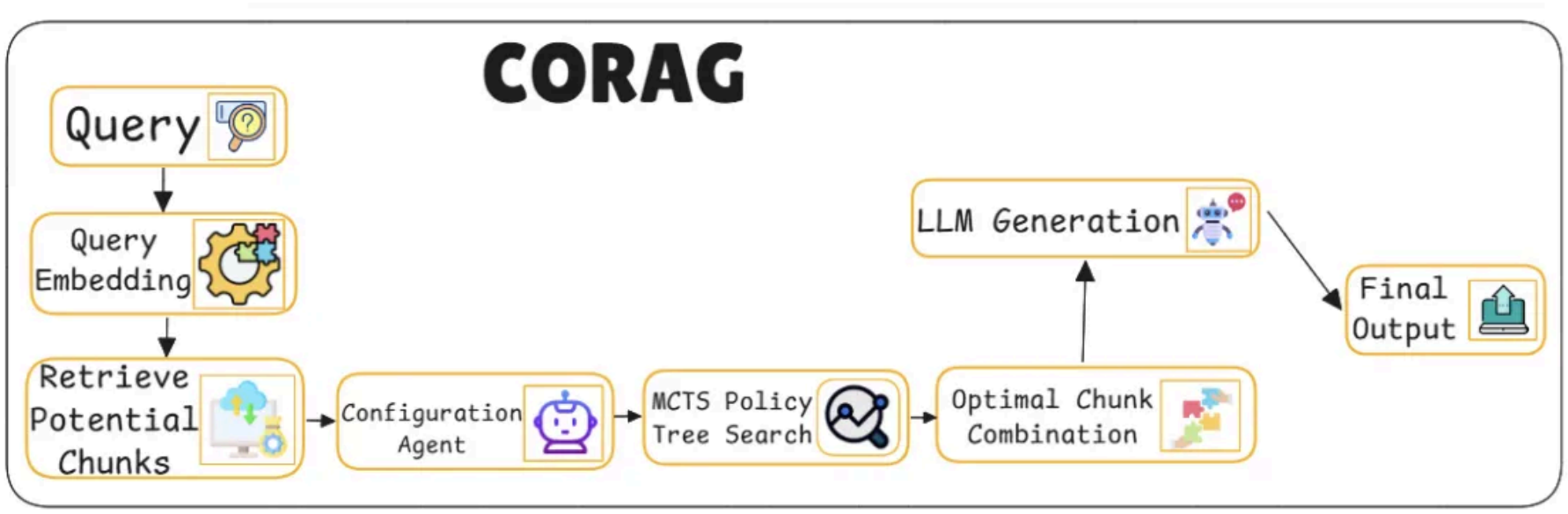


18 CORAG (Cost-Constrained RAG)

Optimizes relevant chunk selection from databases.

KEY FEATURES:

- Tackles three challenges: efficient correlation, non-monotonic utility, and diverse query types
- Uses Monte Carlo Tree Search (MCTS) for optimal chunk combination while factoring in cost constraints
- Achieves up to a 30% improvement over baseline models

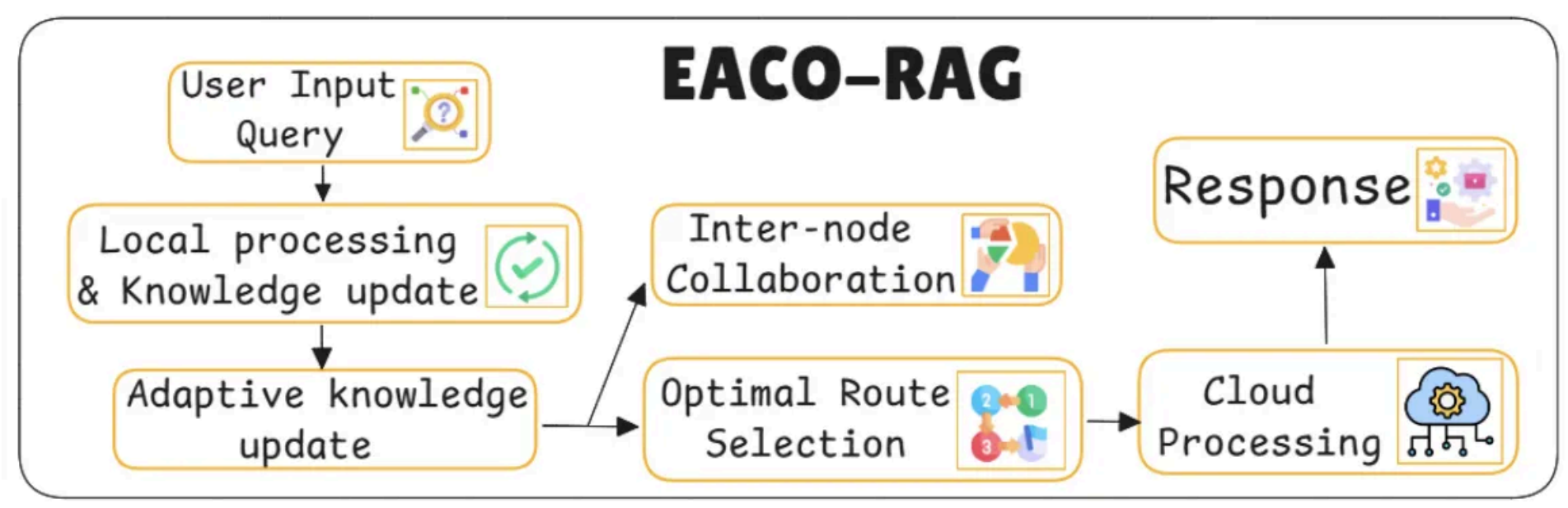


19 EACO-RAG

Enhances RAG with edge computing for faster, efficient responses.

KEY FEATURES:

- Vector datasets are distributed across edge nodes, reducing delays and resource use
- Adaptive knowledge updates and inter-node collaboration improve accuracy
- A multi-armed bandit approach optimizes cost, accuracy, and delay in real-time

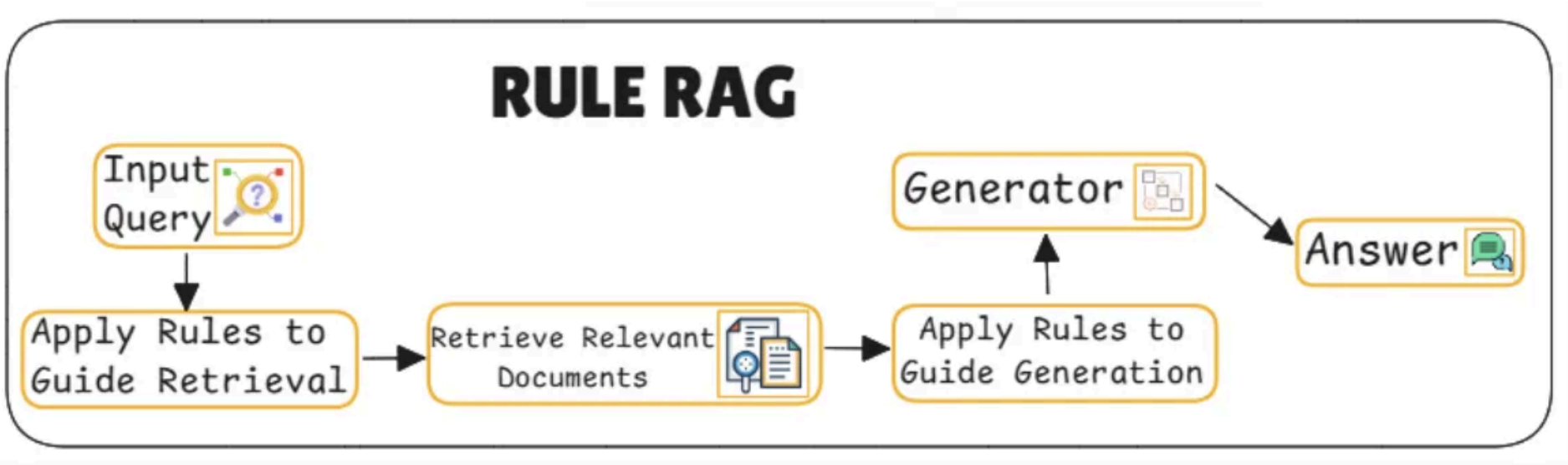


20 RULE RAG

Adds rule-based guidance to RAG for question answering.

KEY FEATURES:

- Retrieves documents logically relevant to queries using predefined rules
- Uses rules to guide answer generation for accuracy and context
- Includes in-context learning (ICL) and a fine-tuned version (FT)

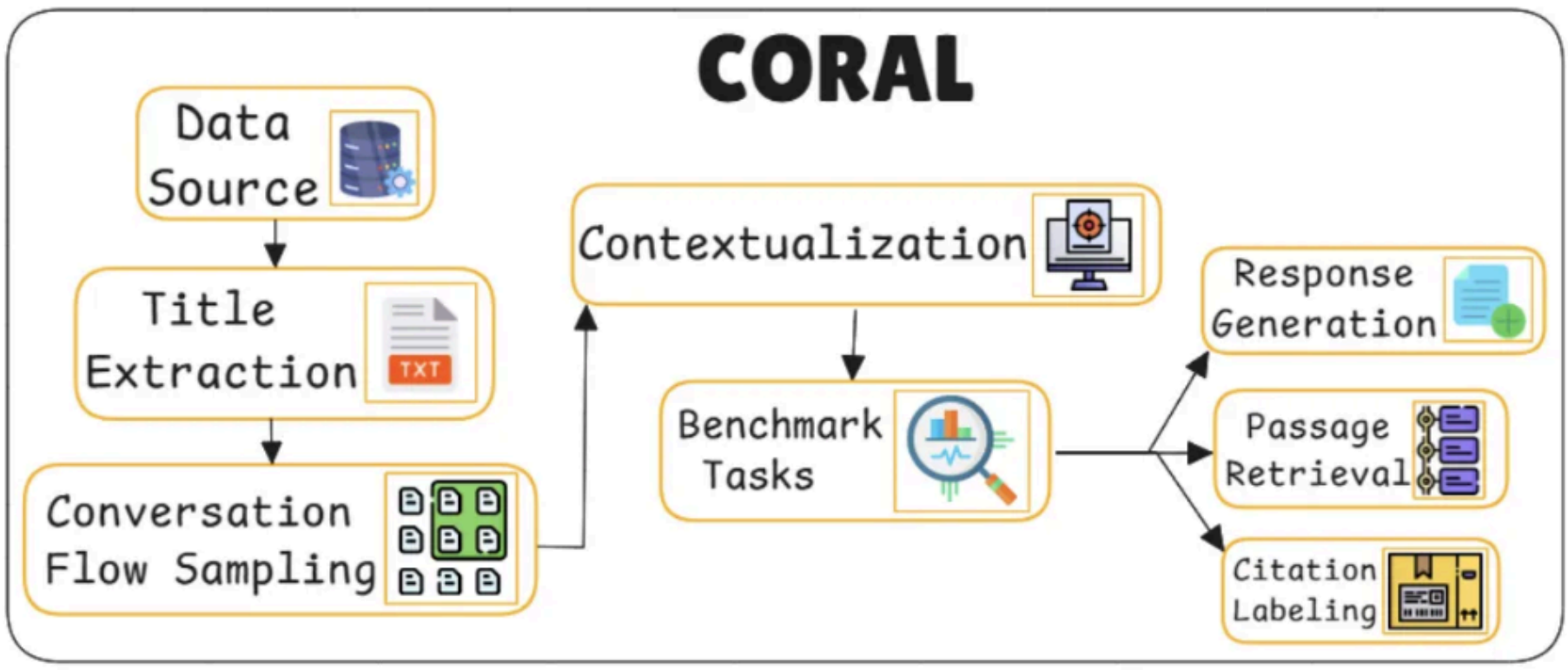


21 Conversational RAG (CORAL)

Benchmarks multi-turn conversational RAG using Wikipedia data.

KEY FEATURES:

- Evaluates passage retrieval, response generation, and citation labeling
- Handles open-domain, realistic, multi-turn conversations
- Bridges single-turn RAG research and real-world multi-turn needs

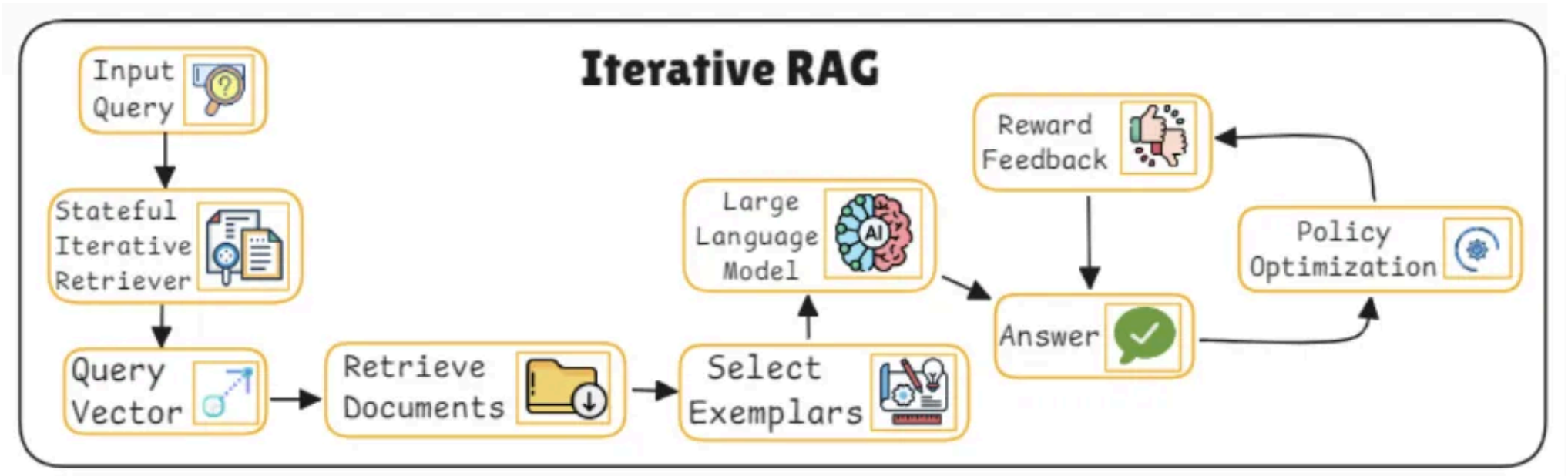


22 Iterative RAG

Performs multiple retrieval steps, refining the search based on feedback from previously selected documents.

KEY FEATURES:

- Retrieval decisions follow a Markov decision process
- Reinforcement learning improves retrieval performance
- Maintains an internal state to adjust future retrieval steps based on accumulated knowledge

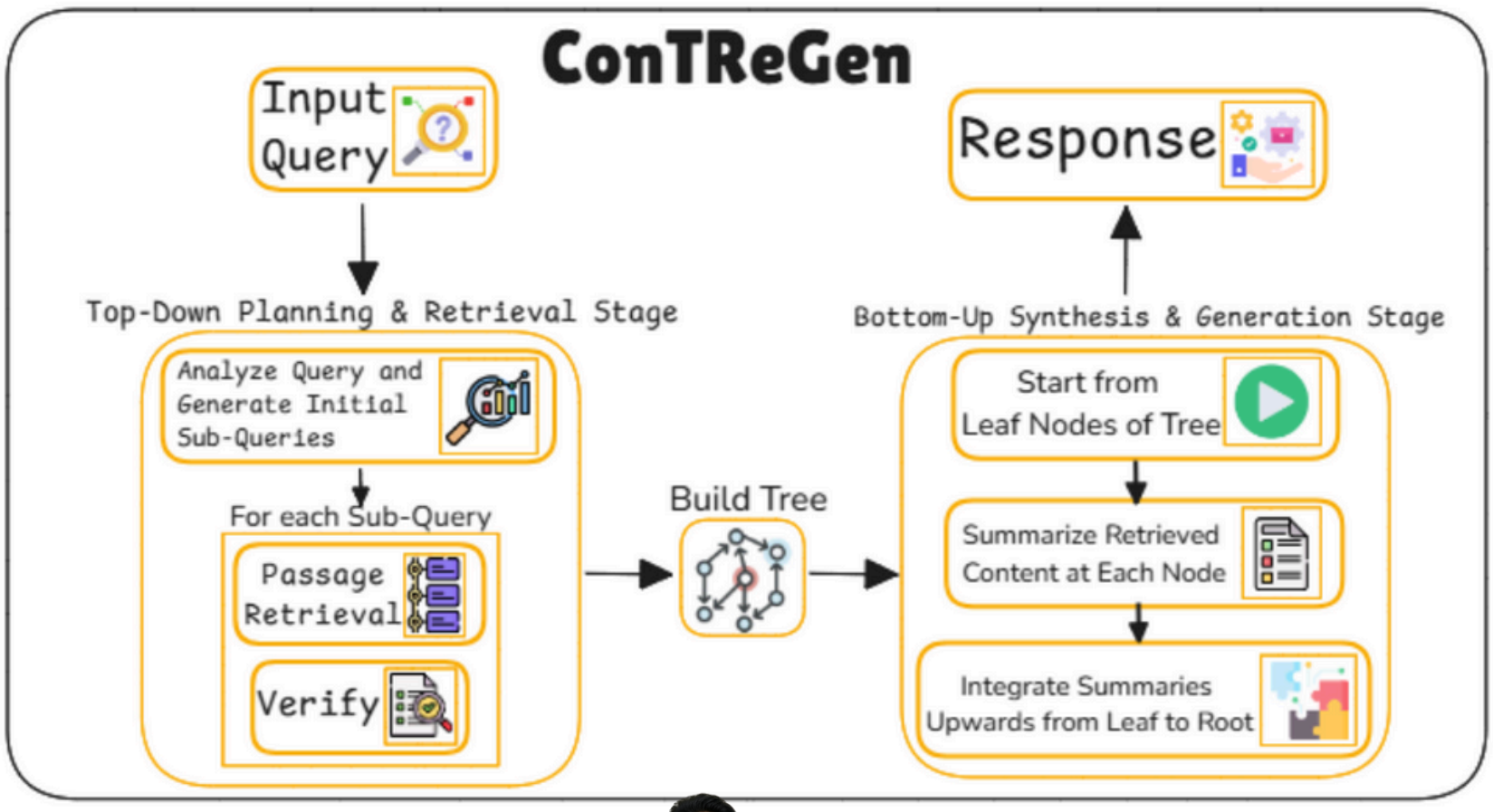


23 ConTReGen (Context-driven Tree-structured Retrieval)

Decomposes complex queries into hierarchical sub-queries, enhancing retrieval depth through a tree-structured approach.

KEY FEATURES:

- Uses a two-stage workflow: top-down exploration creates a tree of passages, followed by bottom-up synthesis to produce long-form responses
- Reduces gaps in information and improves generated content quality
- Enables retrieval at multiple abstraction levels for comprehensive answers

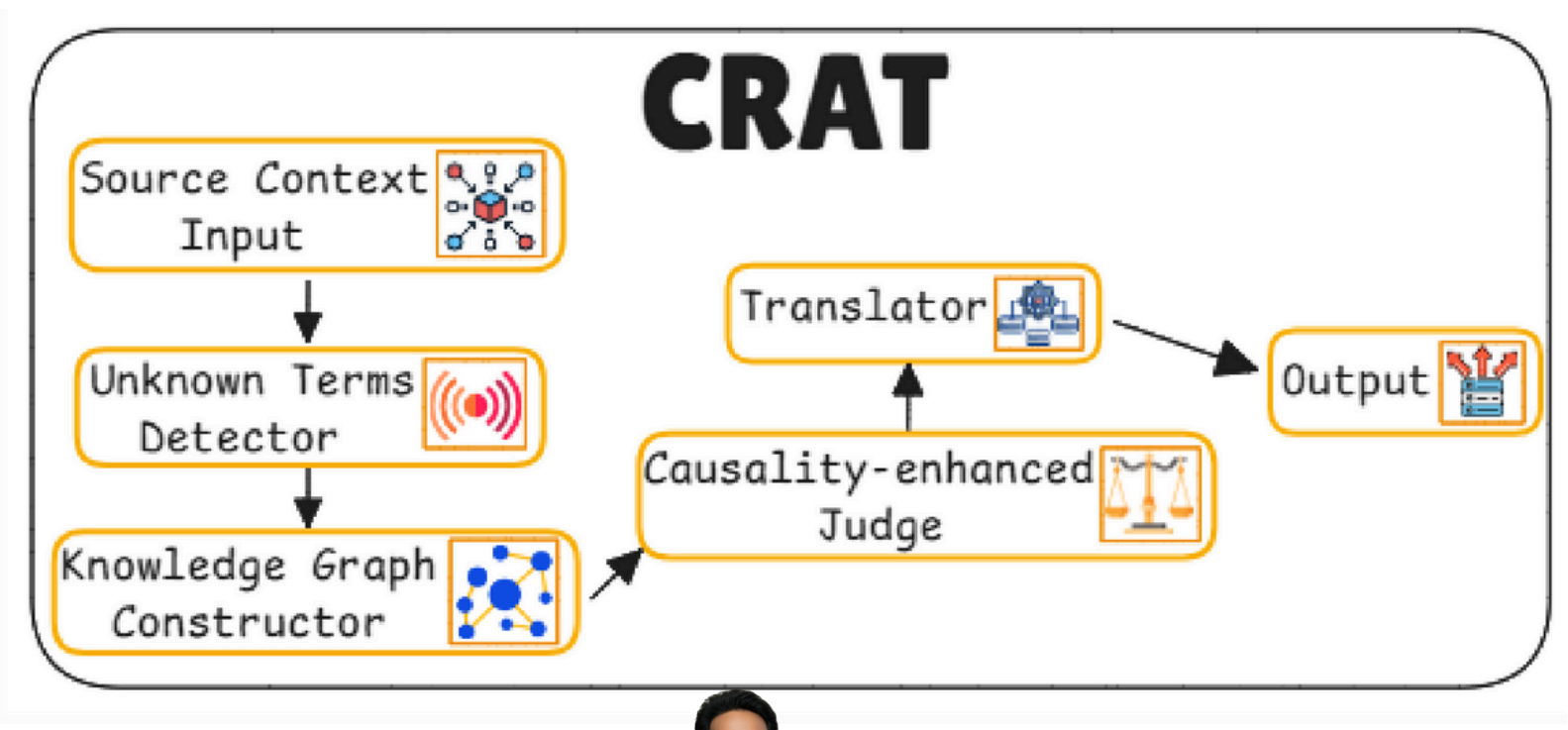


24 CRAT (Causality-Enhanced Reflective and Retrieval-Augmented Translation)

Multi-agent framework that enhances translation by detecting, clarifying, and translating ambiguous terms using causality validation.

KEY FEATURES:

- Combines internal and external knowledge sources to capture context for accurate term use
- Uses a judge agent to validate information and ensure context-aligned translations
- Delivers precise, consistent translations by leveraging validated knowledge and causality relationships

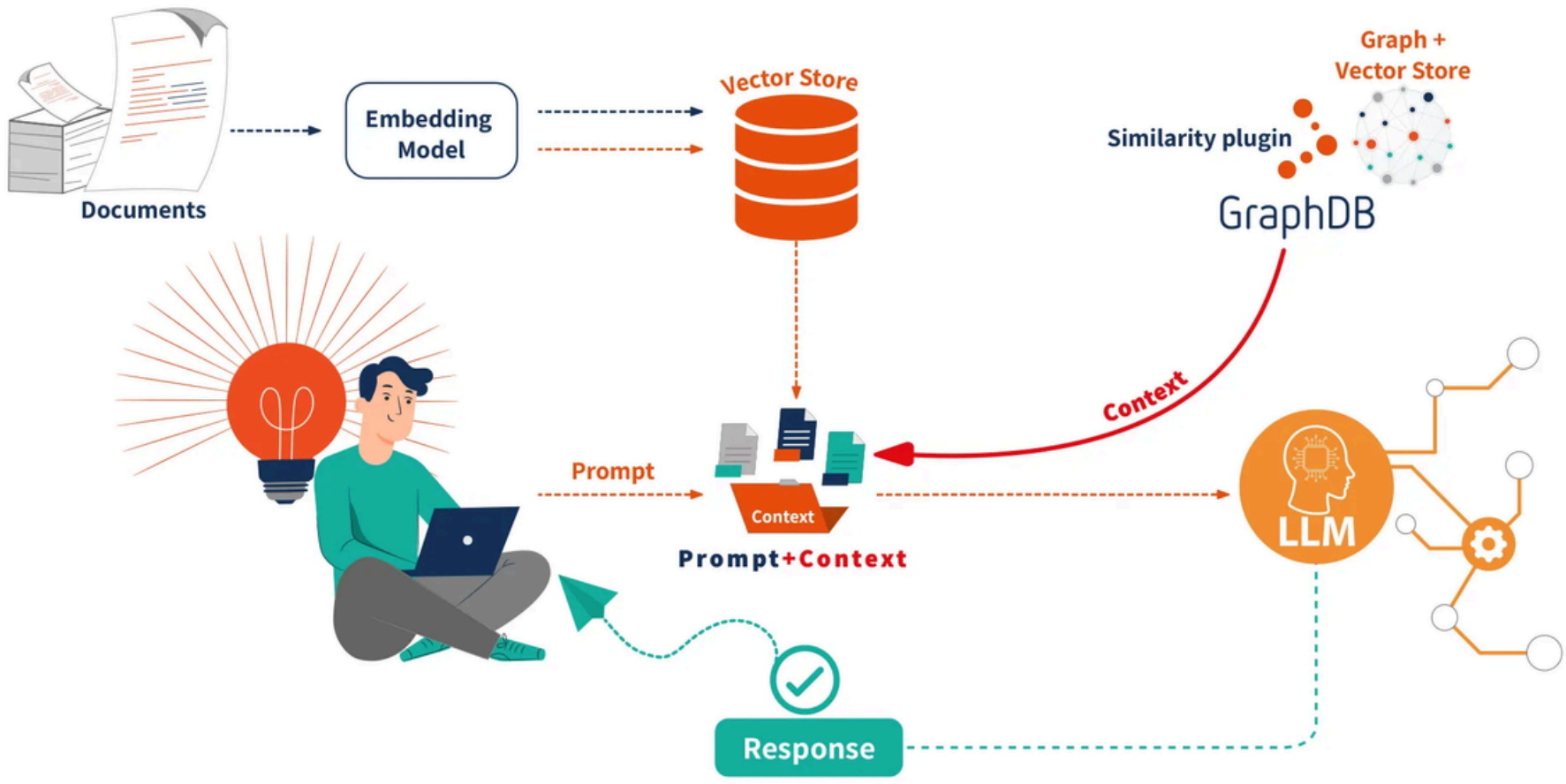


25 Graph RAG

Constructs a knowledge graph on-the-fly, linking relevant entities during retrieval.

KEY FEATURES:

- Leverages node relationships to decide when and how much external knowledge to retrieve
- Confidence scores from the graph guide expansion, avoiding irrelevant additions
- Improves efficiency and accuracy by keeping the knowledge graph compact and relevant



RAG is evolving fast 🚀

From **Standard** → **Advanced** → **Specialized**

If you're building AI products in 2025,
understanding these 25 RAG types is essential.

👍 Like

💬 Comment "RAG"

🔁 Share with your AI team

Follow for more **Agentic AI & GenAI breakdowns** 👨‍💻✨

Follow Chandra for More

Repost to help others in your network