MASTERING LLM
PRESENTS:
COFFEE BREAK
CONCEPTS

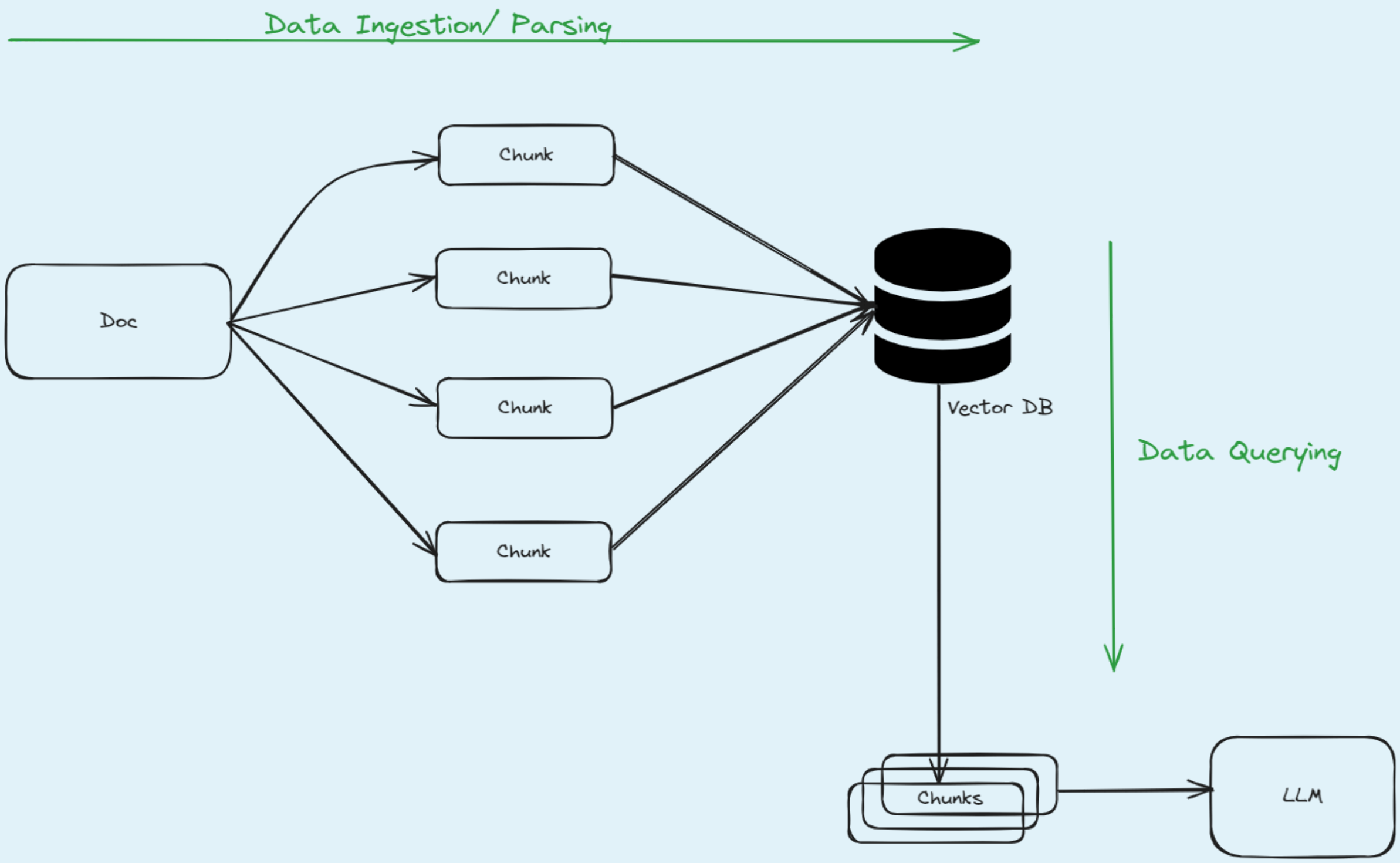
How Agentic RAG solves problem with current RAG limitations

@MASTERING-LLM-
LARGE-LANGUAGE-
MODEL

RAG Framework

The RAG (Retrieval Augmented Generation) framework operates in a specific sequence: Document -> Chunks -> Vector DB -> Chunk Retrieval (Top K) -> LLM

However, this sequence **encounters obstacles when dealing with certain types of queries.**



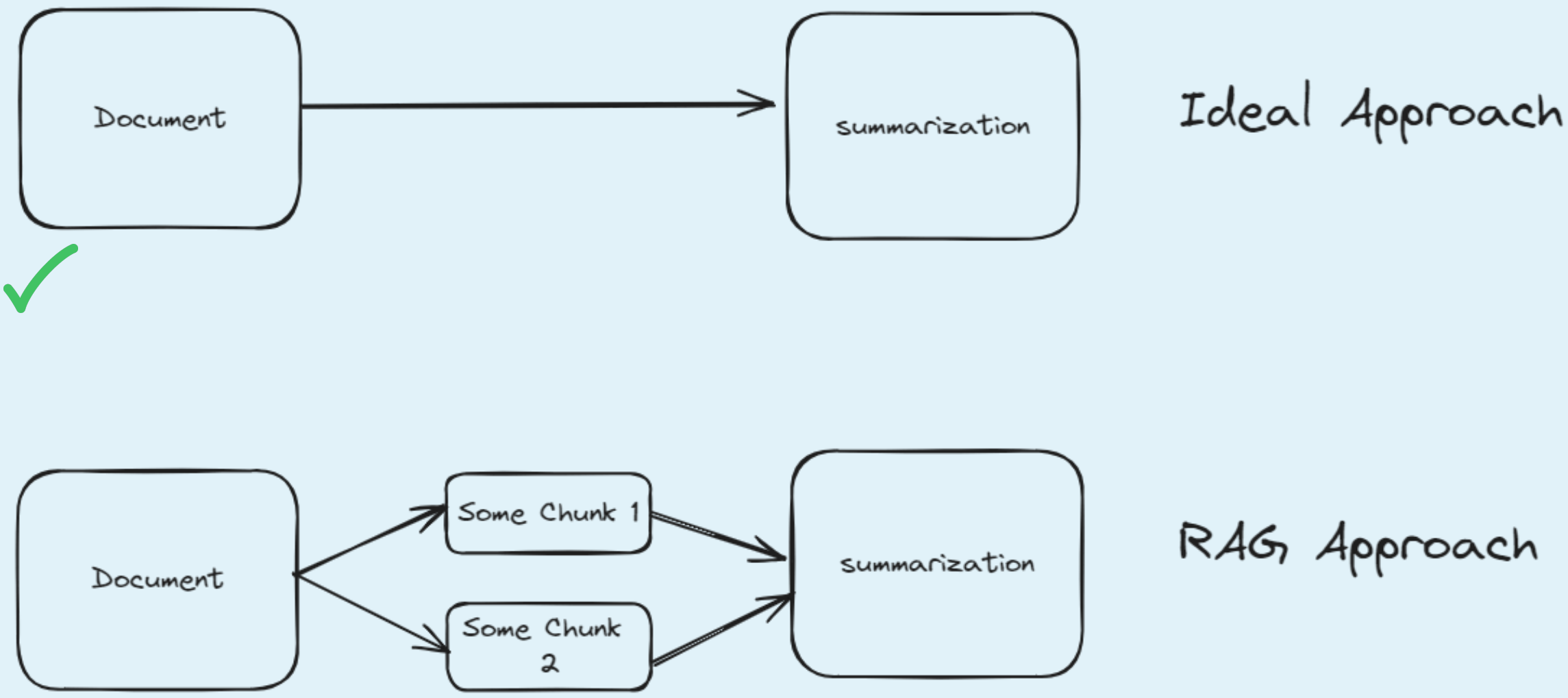
MORE ON THE OTHER SIDE

Next, Lets see what are some of the common limitations?

Problem 1: Summarization

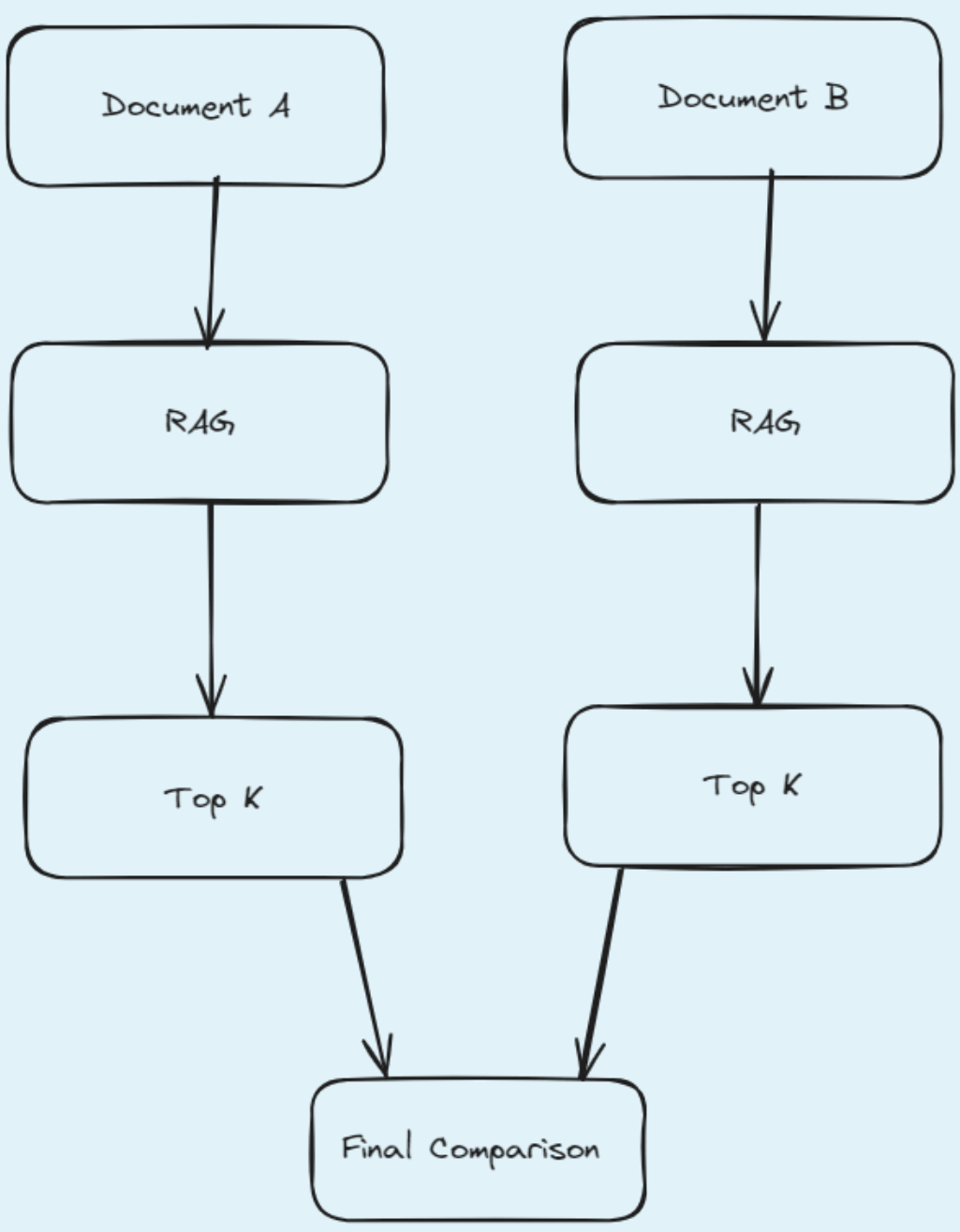
Consider a query like **“Summarize the document”**.

- The conventional RAG approach retrieves the top K chunks and summarizes them.
- But wouldn’t it be more comprehensive if it retrieved all chunks of the document and summarized them?

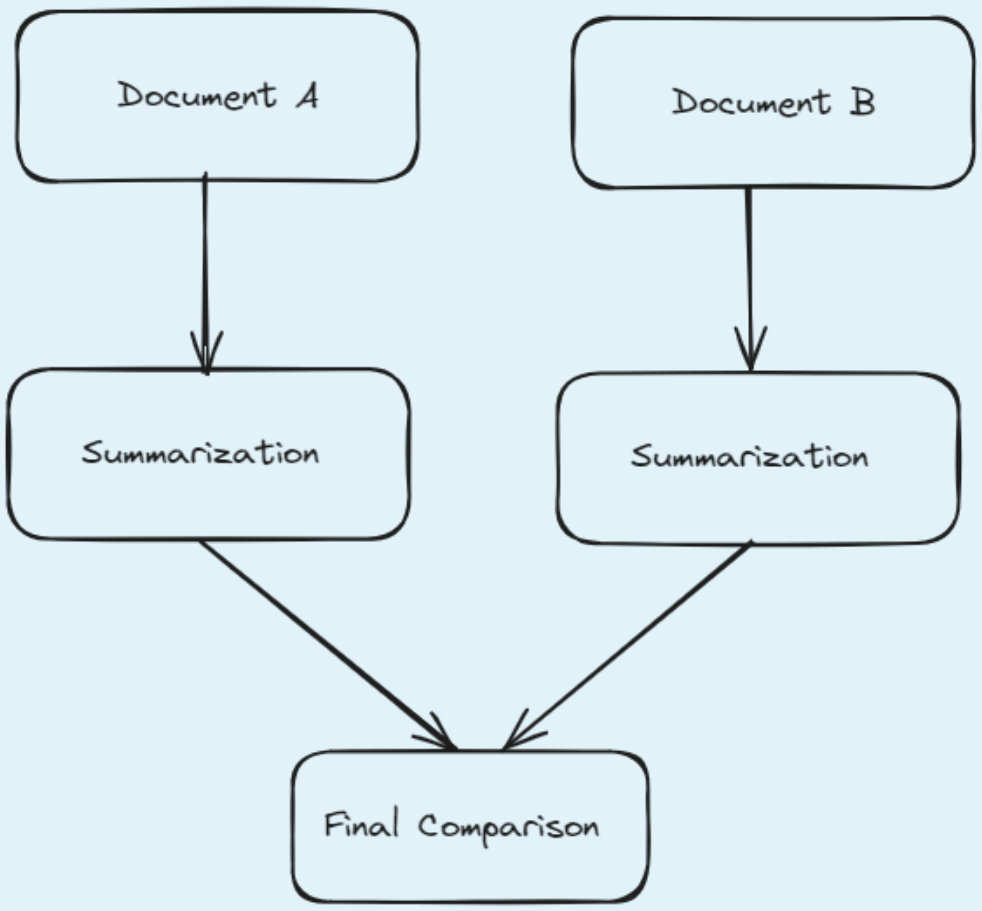


Problem 2: Comparing Documents

- When tasked with comparing Document A and Document B, the **basic RAG retrieves random chunks and attempts to compare these top K chunks**.
- This **doesn't paint an accurate picture** as it doesn't represent the full scope of the documents.



RAG Approach

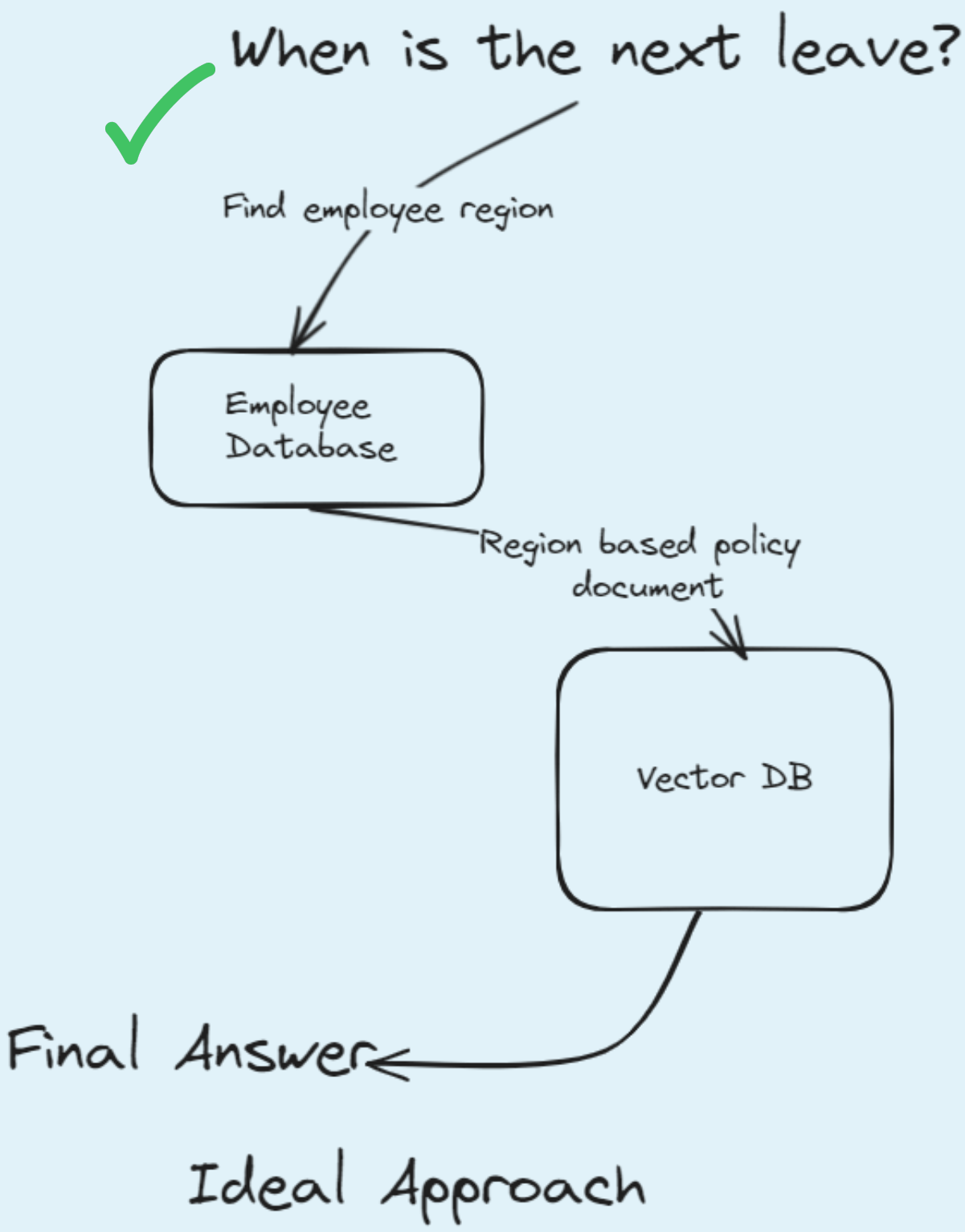


Ideal Approach

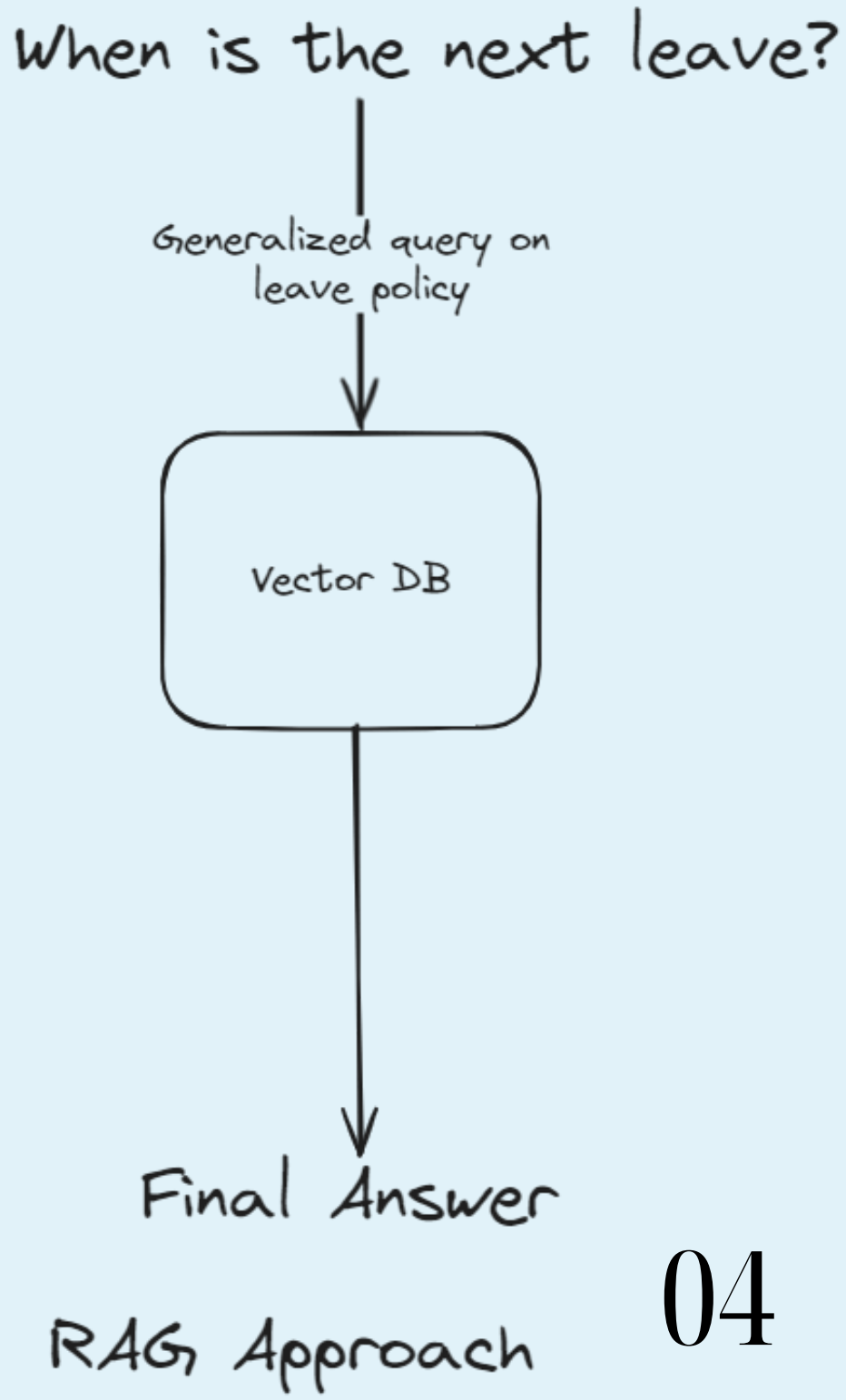
Problem 3: Structured Data Analysis

Consider a question like "**When is the next leave?**".

- The first step is to retrieve the region to which the employee belongs from a structured table.
- Based on the region, the next leave for that region is extracted from the leave policy document.
- This process isn't as straightforward with the current RAG framework.



Ideal Approach

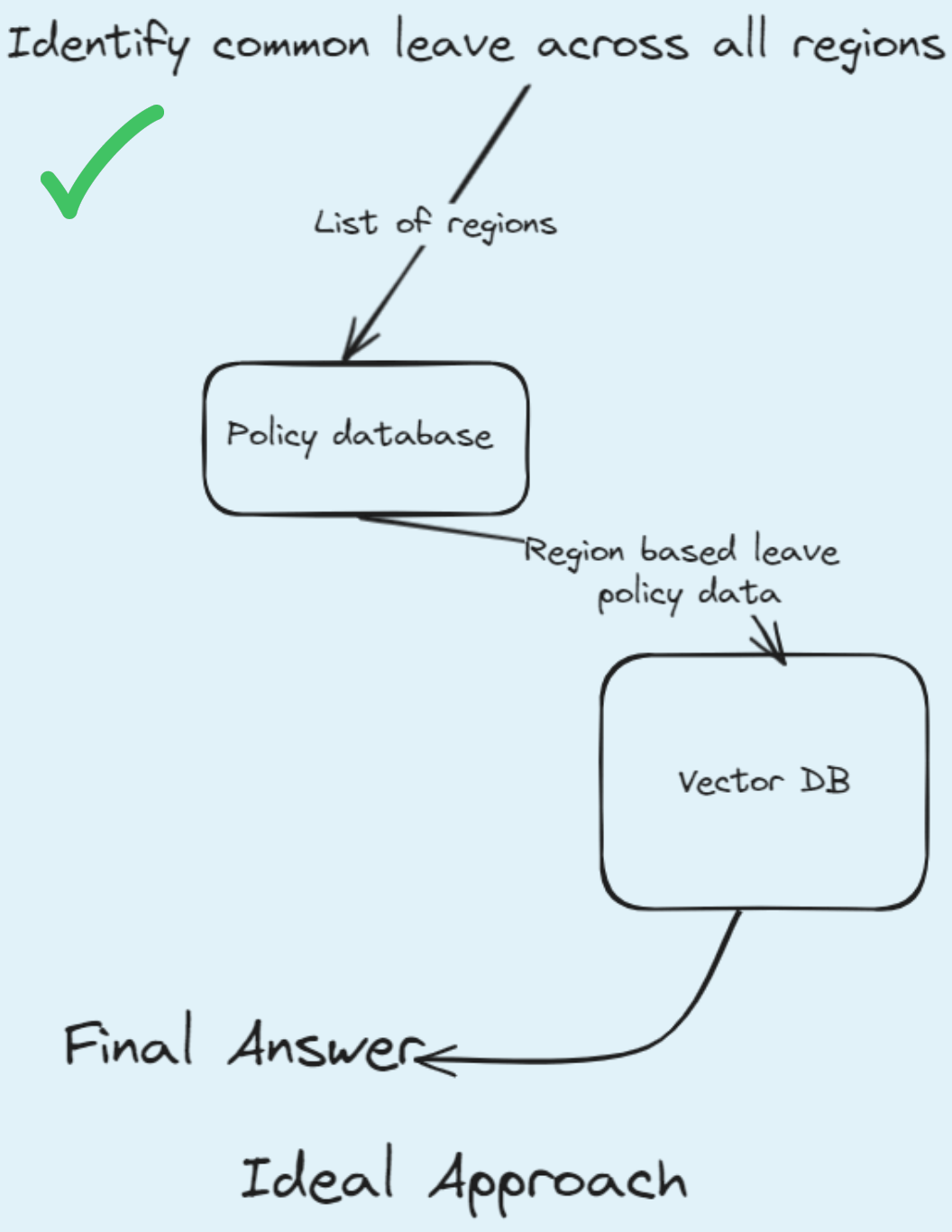


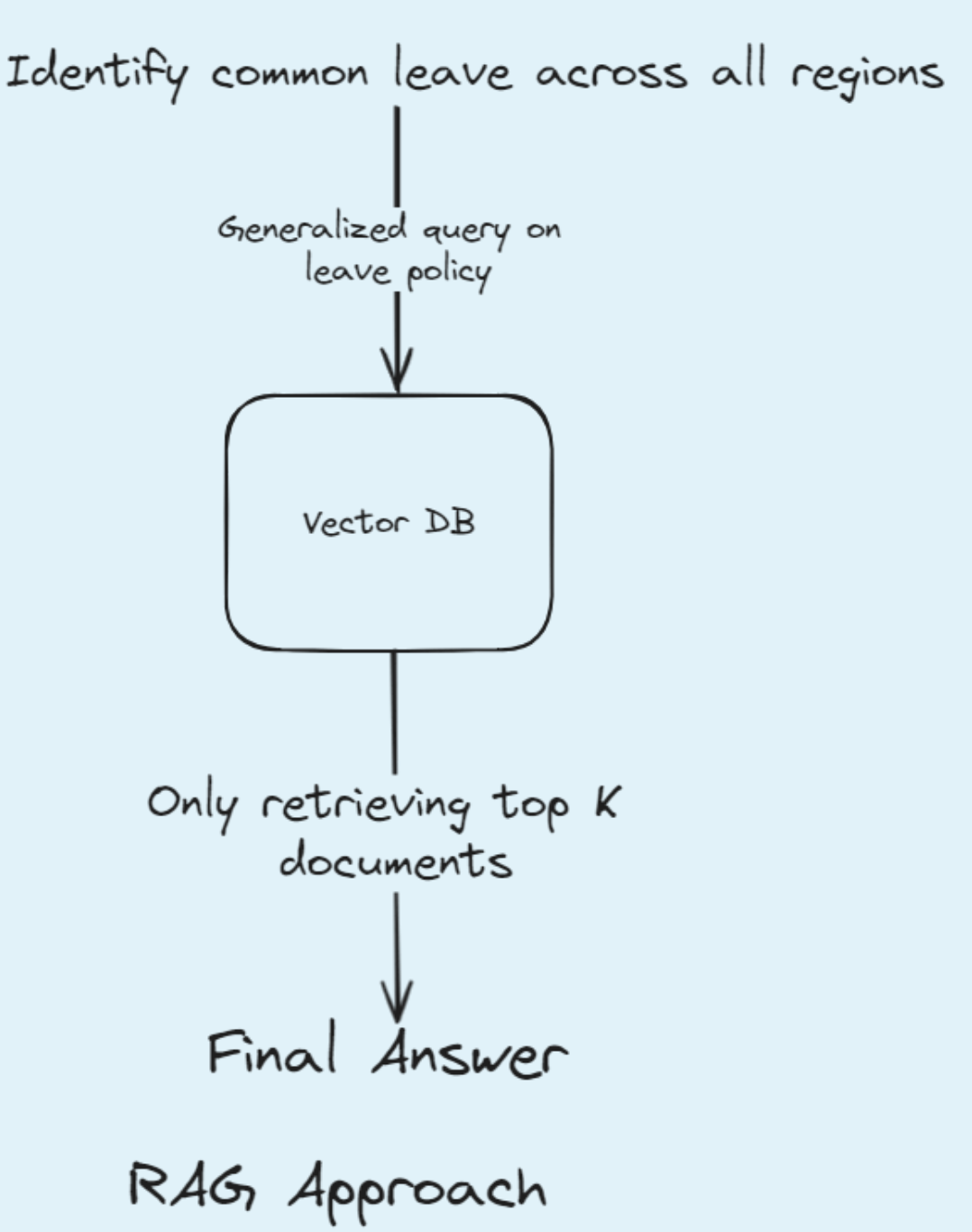
RAG Approach

Problem 4: The Multi-part Question

Consider a question like "**Identify common leave across all regions?**".

- Imagine you have a leave policy document of a company present in 120 countries.
- Since you are passing the top K contexts, the **maximum number of regions that can be compared is limited to K**, where K is the number of chunks passed to LLM.





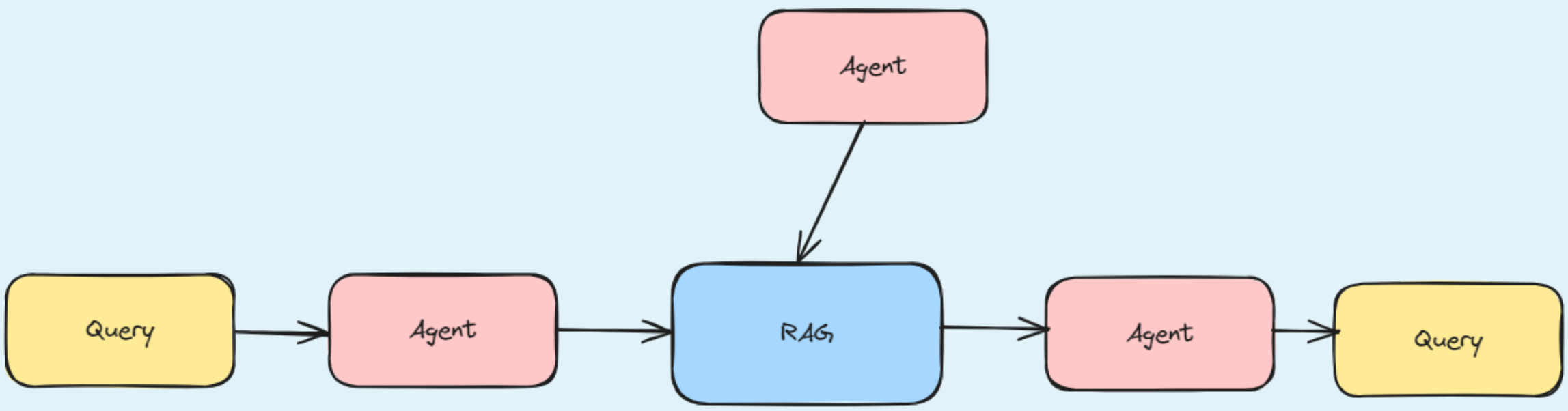
MORE ON THE OTHER SIDE

But wait, how to solve this problems?

Agentic RAG

Agentic RAG can solve this 4 problems by replacing via custom agents.

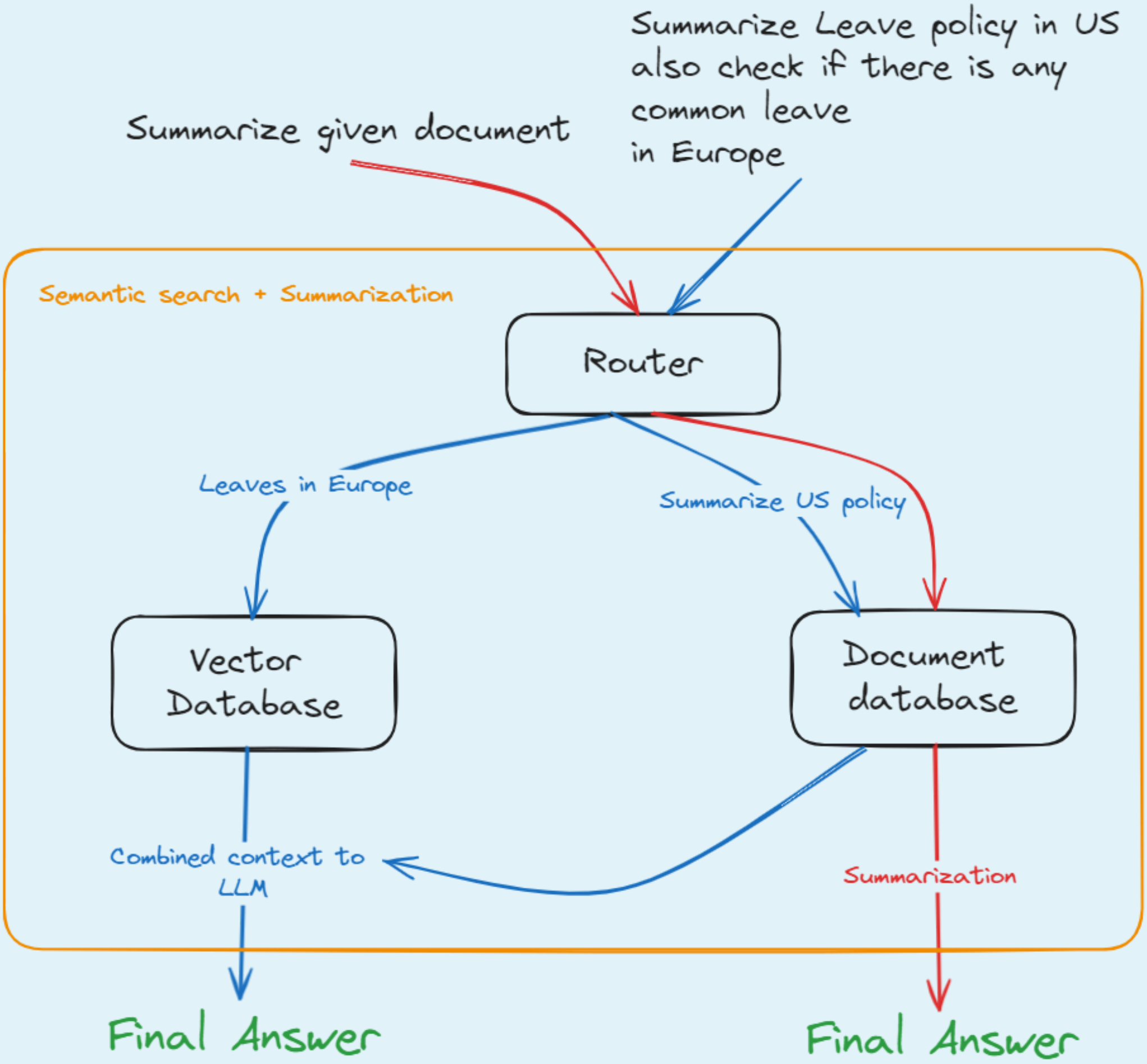
- Agents will interact with multiple systems
- RAG is now one part of this system which agents can use.



- Agents uses LLMs to automate the reasoning and tool selection
- RAG is just another tool which Agent may decides to use.

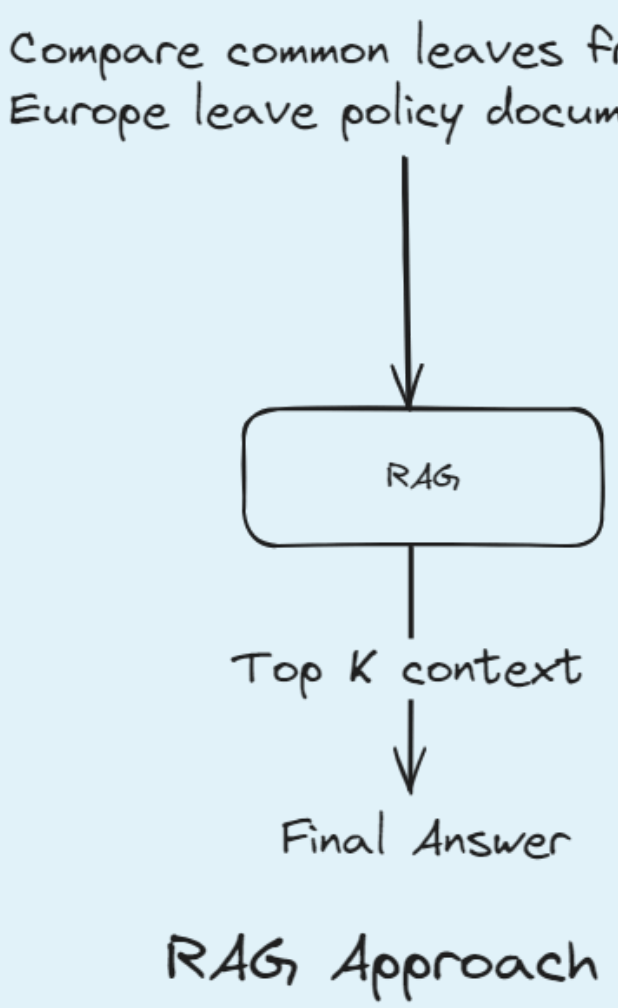
Routing Agent

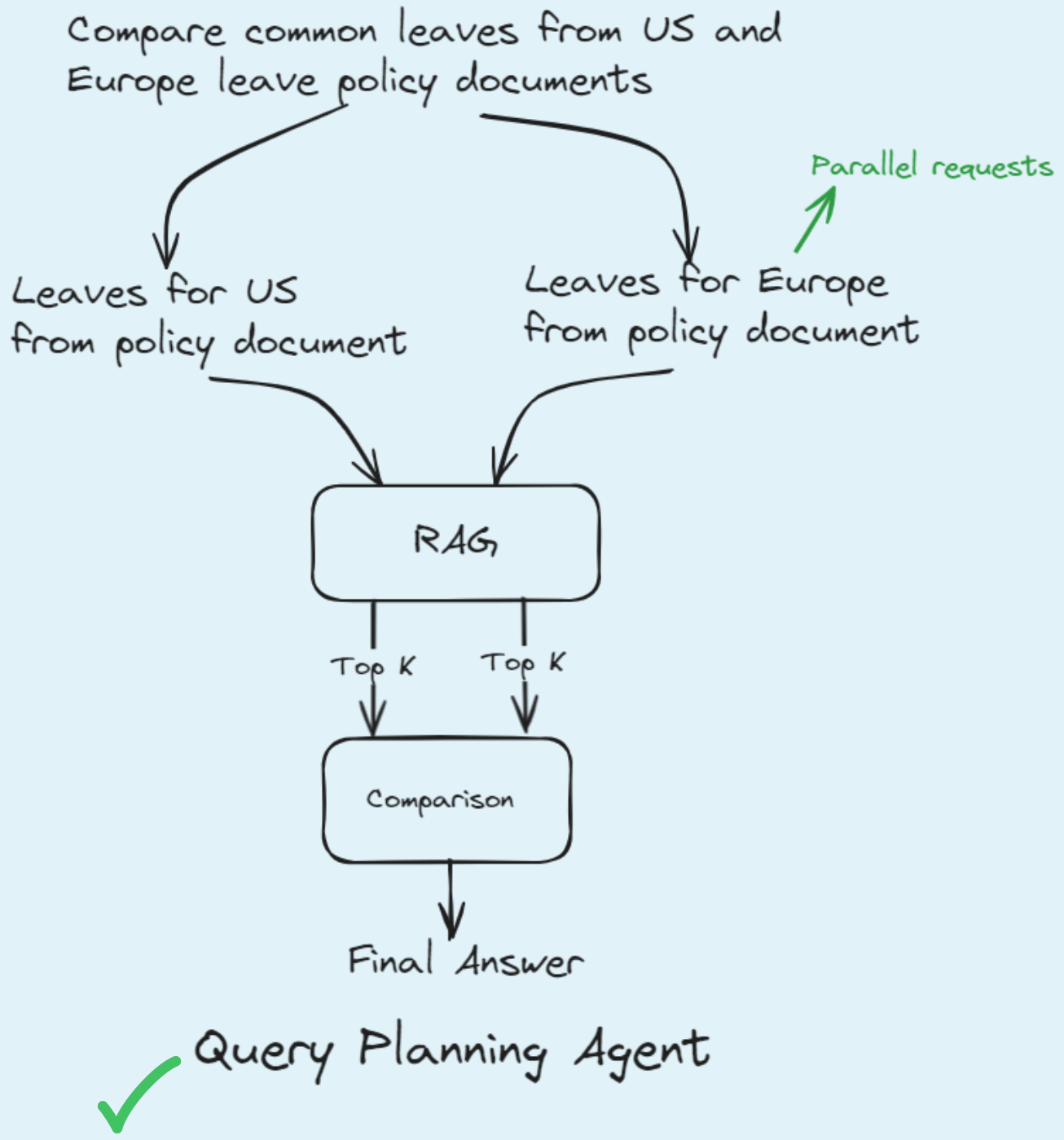
- Routing agents are simple agents which routes the queries.
- An agent can route query in one or multiple tools.
- Remember our question "**Summarize the document**" or a question if we want to combine "**Summarization + Sematic search**" can be solved using below example routing



Query Planning Agent

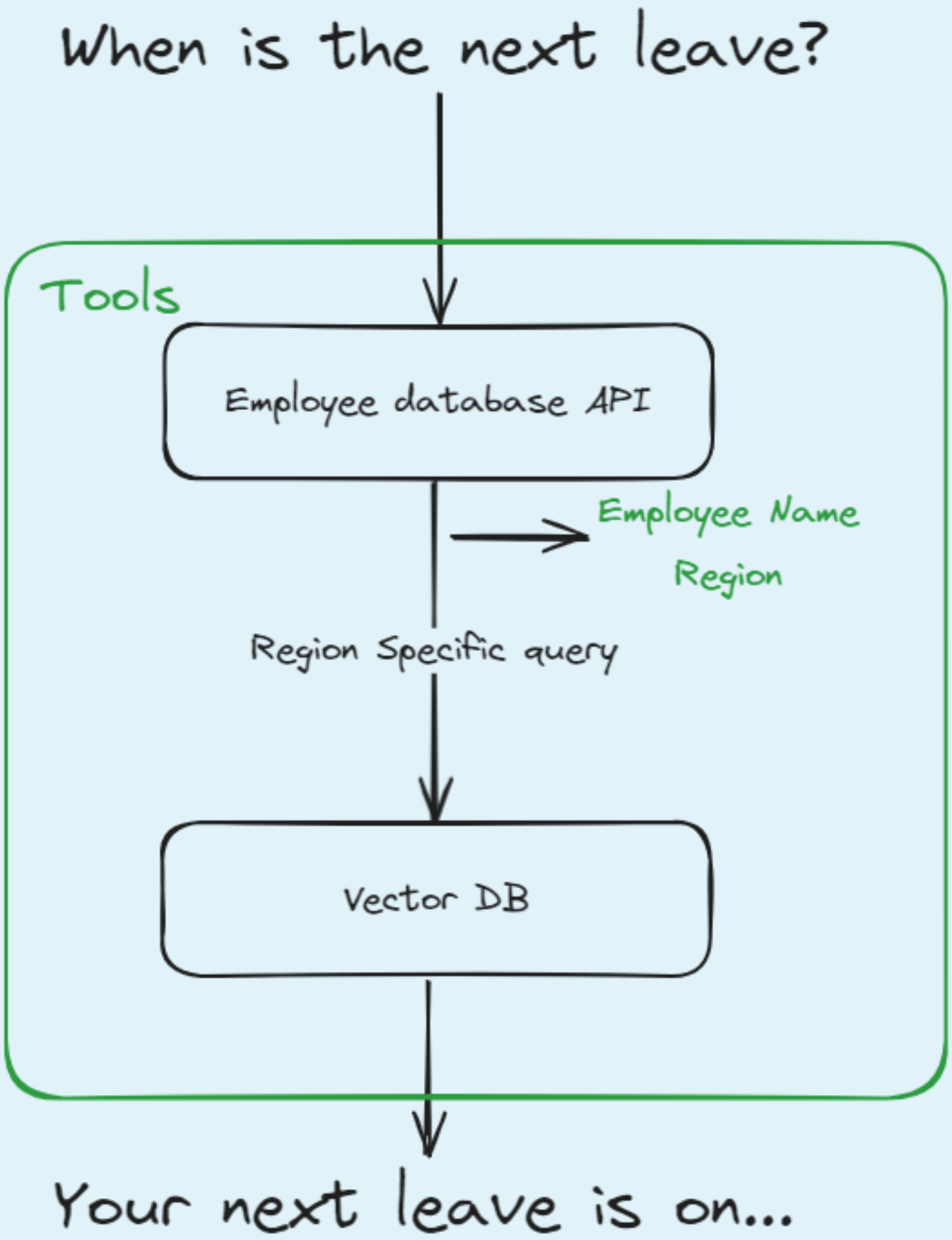
- Query planning agent breaks down the queries into sub-queries.
- Each of the sub-queries can be executed against RAG pipeline.





Tools For Agents

- LLMs can have multiple tools like calling an API, infer parameters for API.
- RAG is now a tool which LLM might use.



Summery

- [x] RAG has limitations when represented with complex questions.

- [x] Few of the use cases like summarization, comparison etc. can't be solve with just RAG

- [x] Agentic RAG can help overcome limitation of RAG.

- [x] Agentic RAG treats RAG as a tool which it can use for semantic search.

- [x] Agents equipped with routing, query planning and tools can out perform traditional RAG applications.

Comment below on which topic you want to understand next in this "**Coffee Break Concepts**" series and we will include those topics in the upcoming weeks

www.masteringllm.com

LLM Interview Course

Want to Prepare yourself for an LLM Interview?

- ✓ 100+ Questions spanning 14 categories
- ✓ Curated 100+ assessments for each category
- ✓ Well-researched real-world interview questions based on FAANG & Fortune 500 companies
- ✓ Focus on Visual learning
- ✓ Real Case Studies & Certification

Coupon Code - LLM50

Coupon is valid till 30th May 2024