

# The Capstone Assembly Manual: Building a Semantic Answer Engine

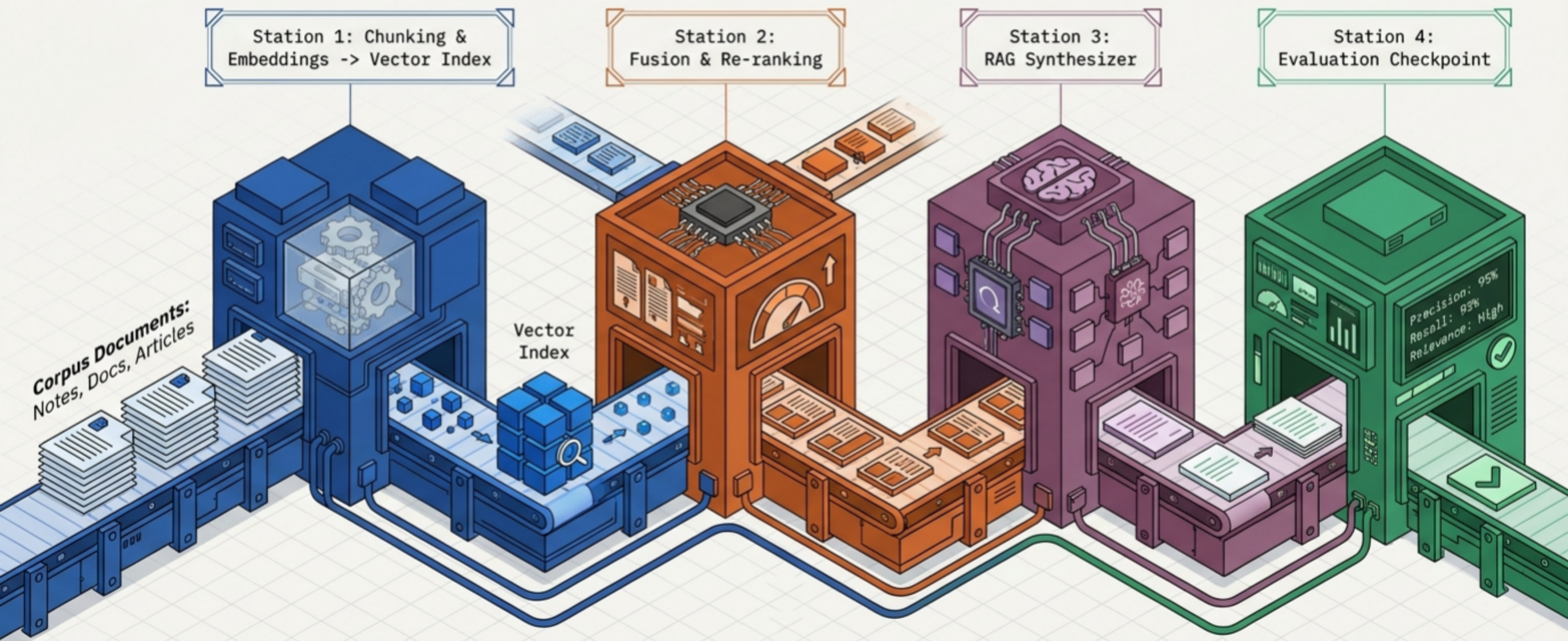


**Goal:** Assemble the course concepts into a single, fully functional answer engine over a custom corpus.



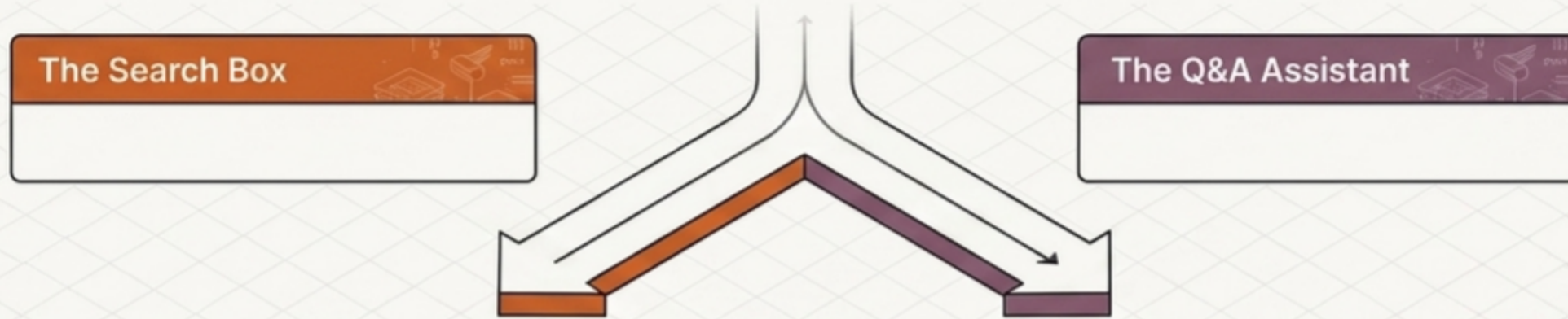
# Everything Converges: The Ingestion-to-Generation Pipeline

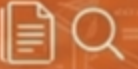
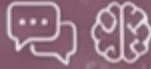
Your **final project** wires ingestion, hybrid retrieval, re-ranking, RAG, and evaluation into one continuous flow.





# The Architectural Fork: Defining the End-User Experience



|                                                                                                                  | The Search Box  | The Q&A Assistant  |
|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  <b>The Output</b>              | Returns exact source passages                                                                      | Returns synthesized, citable answers                                                                  |
|  <b>Architecture Required</b> | Hybrid Retrieval + Re-ranker                                                                       | Full RAG Pipeline (Retrieval + LLM Generator)                                                         |
|  <b>Primary Metric</b>        | nDCG / MRR (Ranking accuracy)                                                                      | Groundedness / Answer Relevance                                                                       |

Select your target experience. Your choice dictates the primary metric used to evaluate your final stack.

# Laying the Foundation: The Minimal SearchStack

---

## Component: Dense Retrieval Only

Documents pass through an encoder and chunker into a Vector Index

---

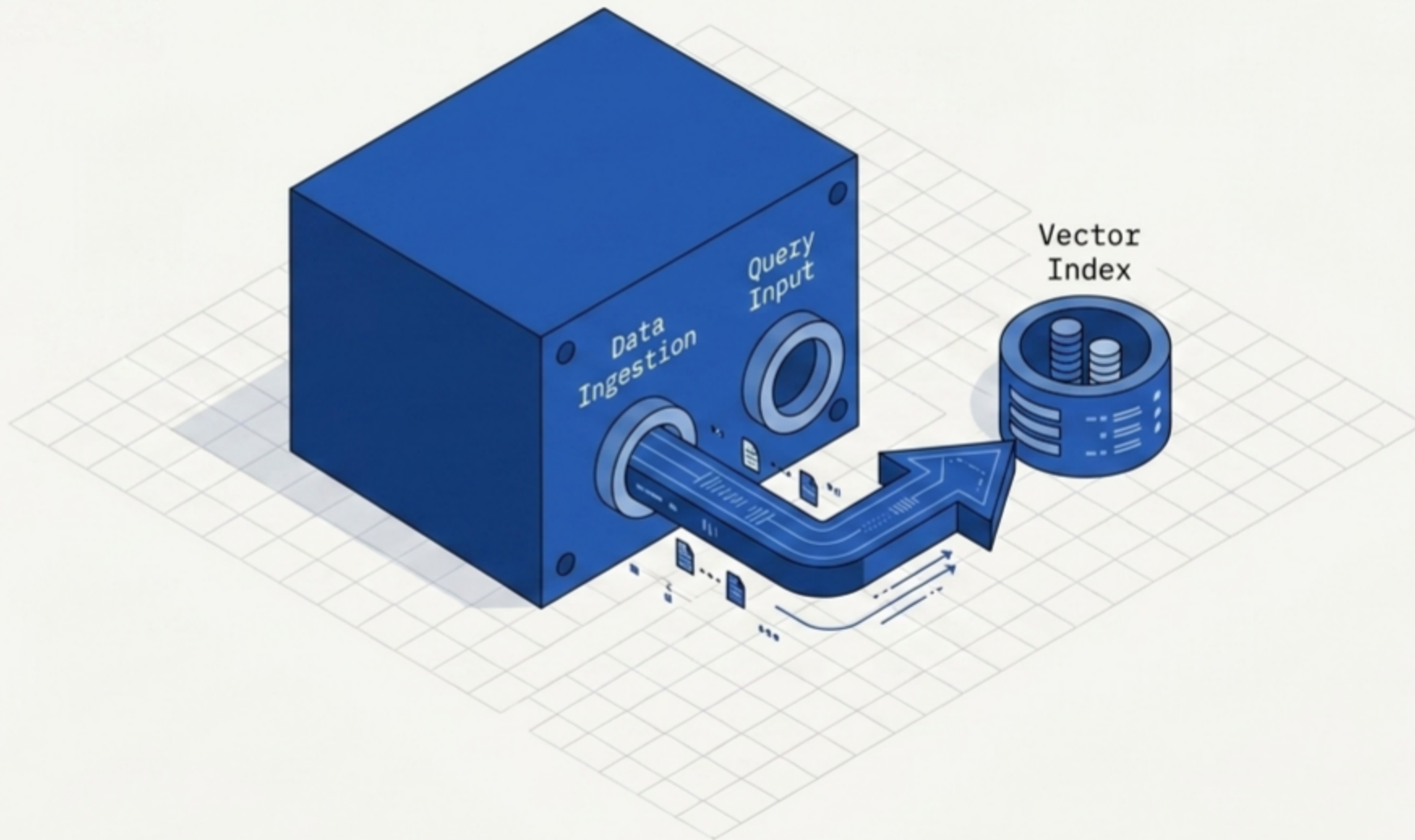
## Mechanics:

Documents pass through an encoder and chunker into a Vector Index

---

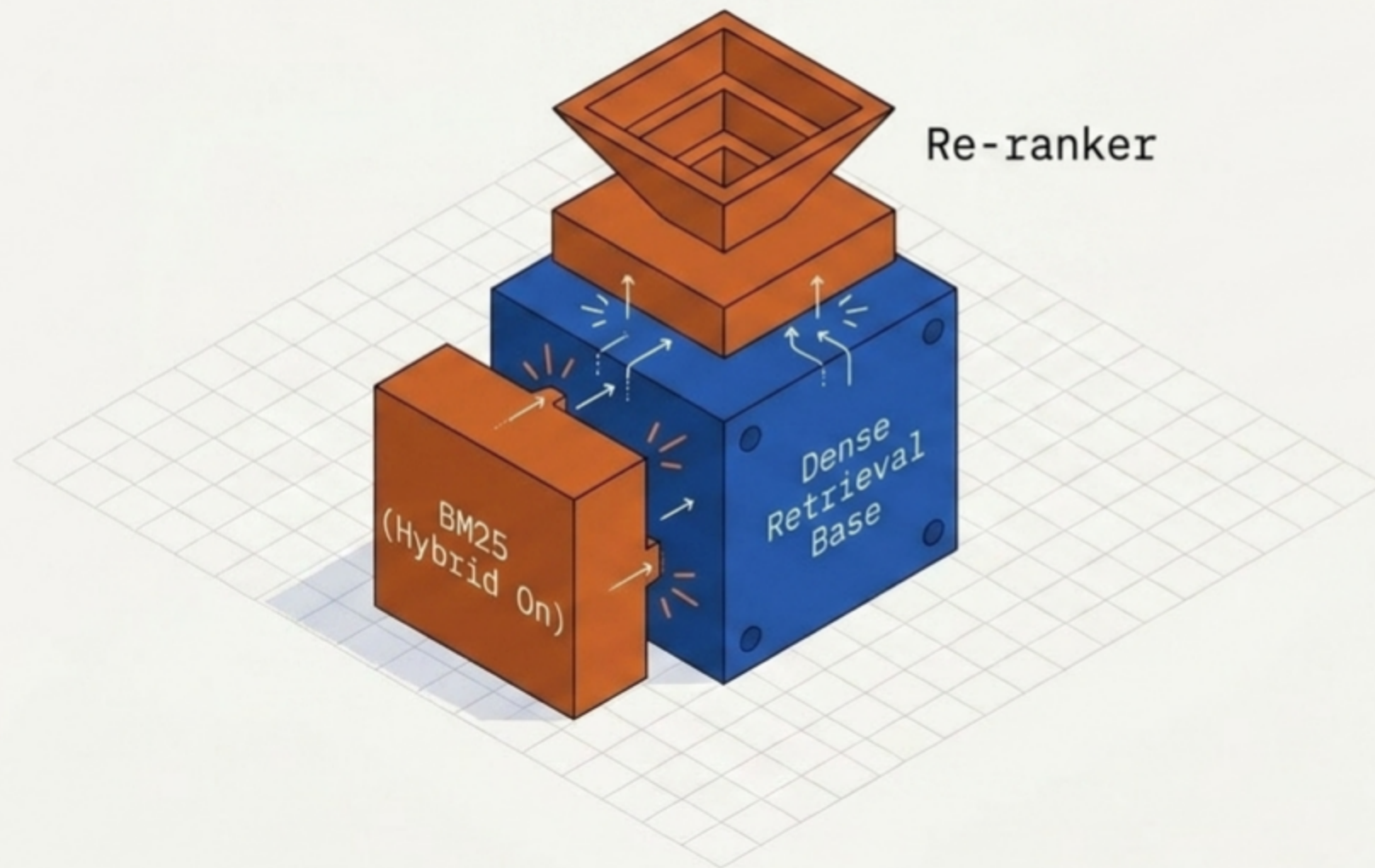
## Action:

Start minimal. Establish your baseline retrieval performance before introducing system complexity.





# Precision Upgrades: Snapping on Hybrid Retrieval and Re-ranking



Add components one at a time. The Hybrid module captures exact-keyword matches, matches, while the Re-ranker ensures the absolute most relevant passages surface to the very top.

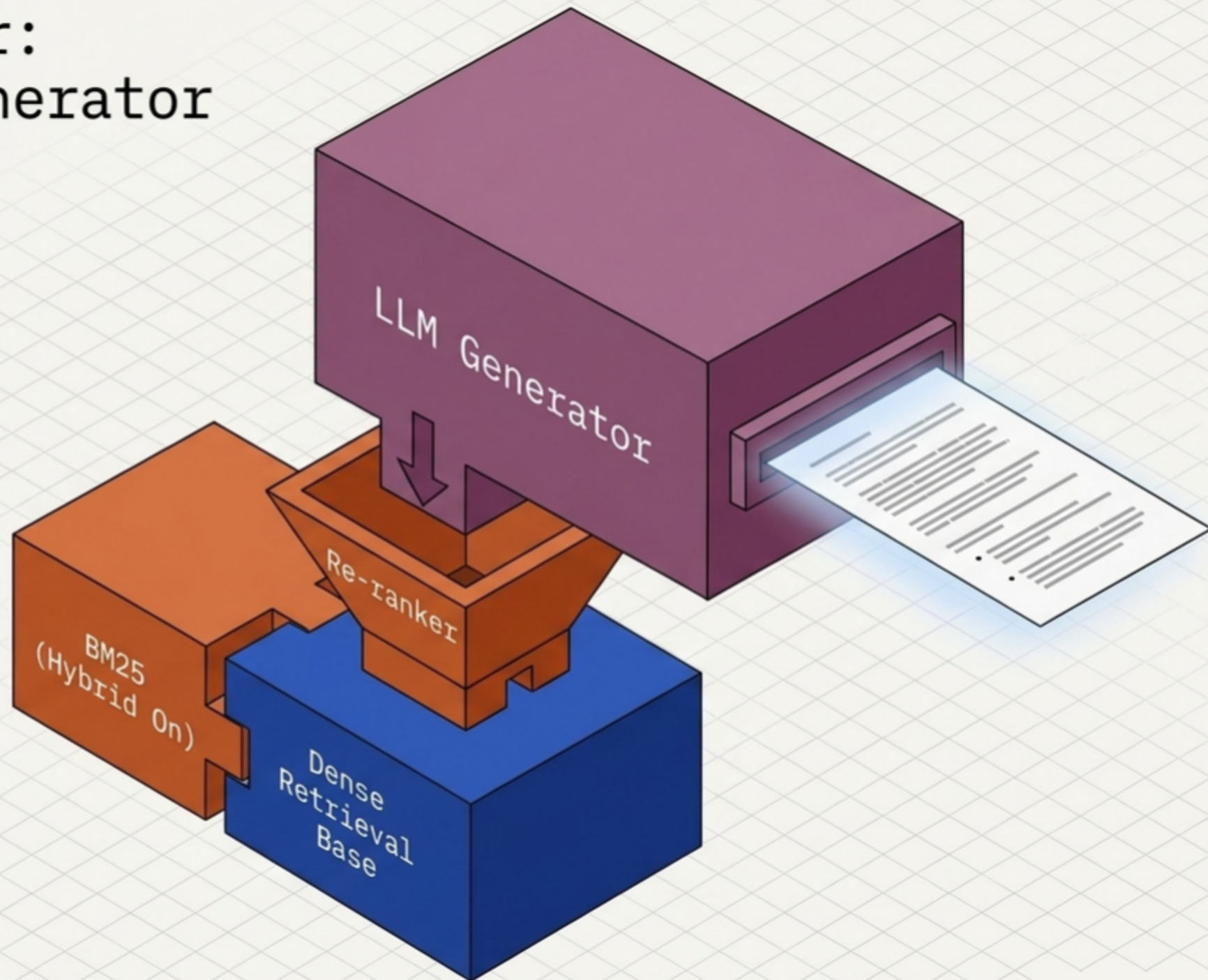


# The Synthesis Layer: Integrating the Generator

Component:  
Retrieval-Augmented  
Generation (RAG)

Function:  
Transforms retrieved  
passages into synthesized,  
coherent answers.

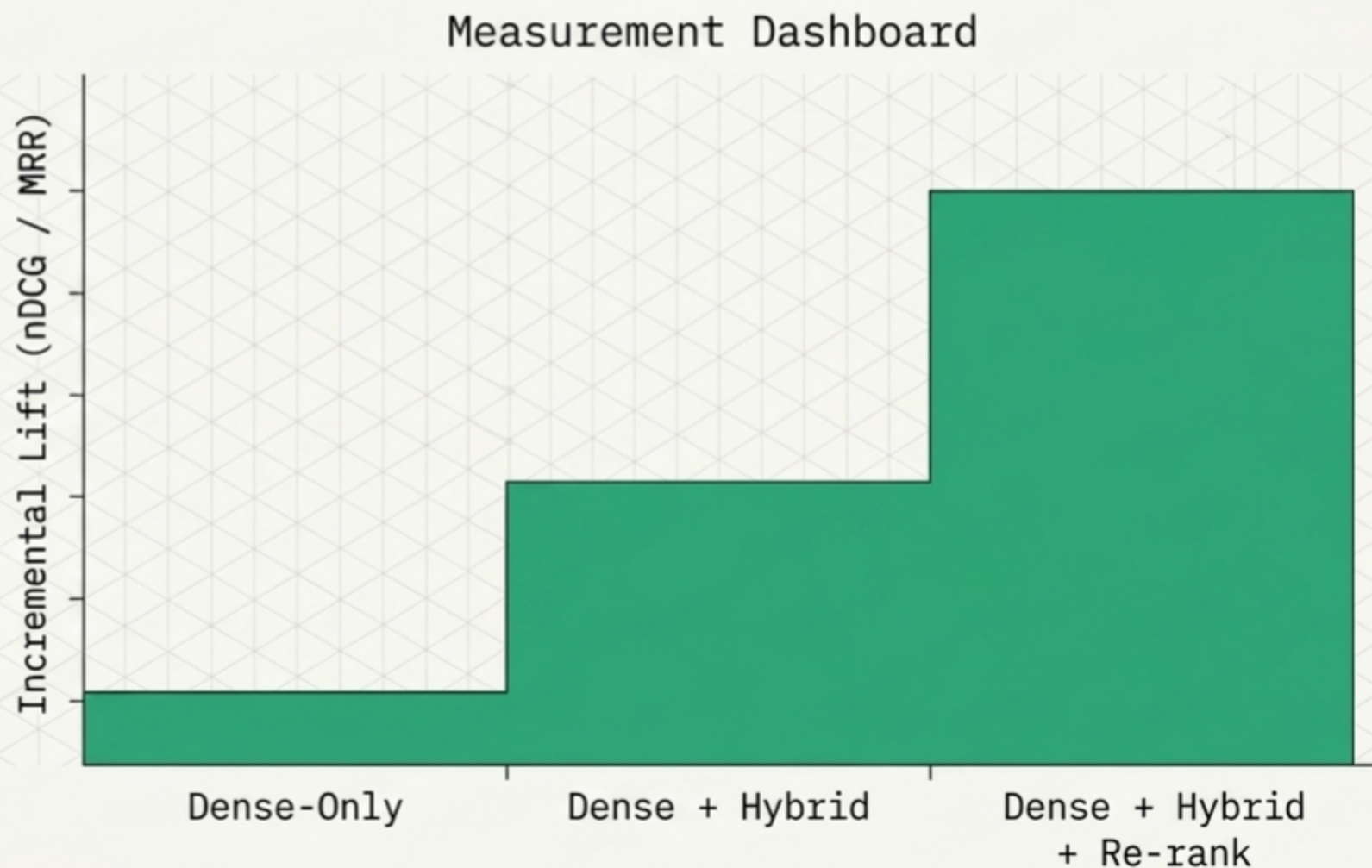
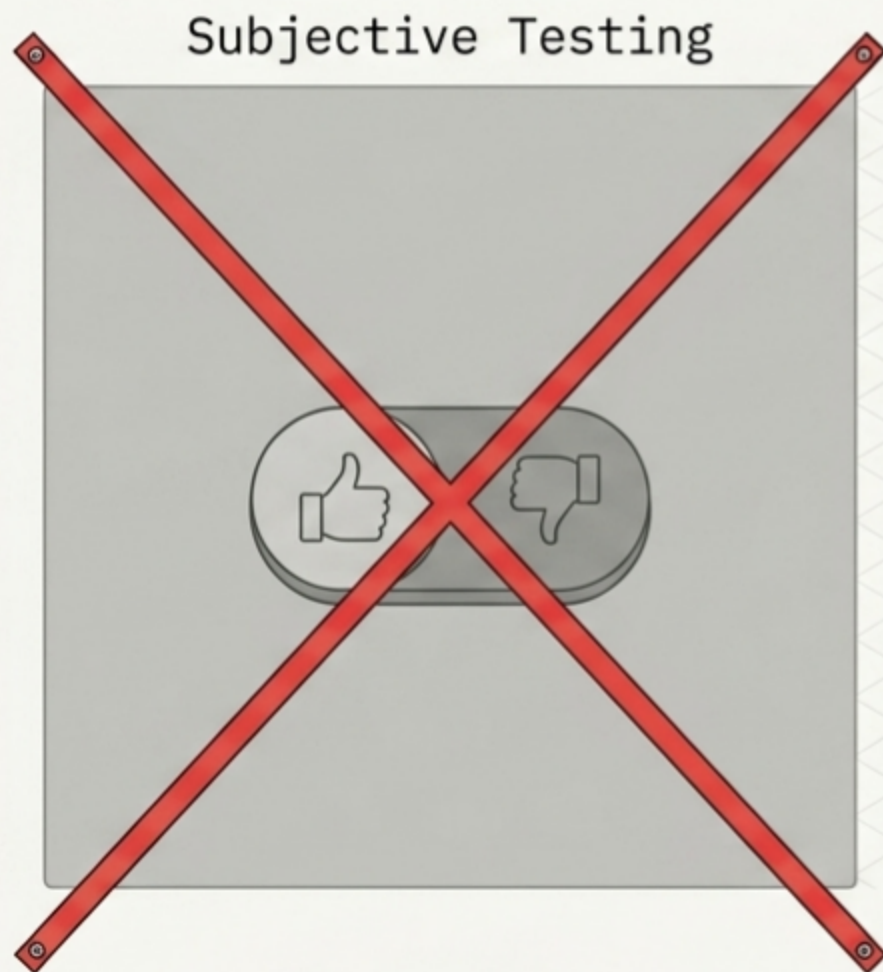
Constraint:  
Use only if your use case  
dictates a Q&A Assistant,  
not a traditional search  
box.





# Let Numbers, Not Vibes, Dictate the Final Configuration

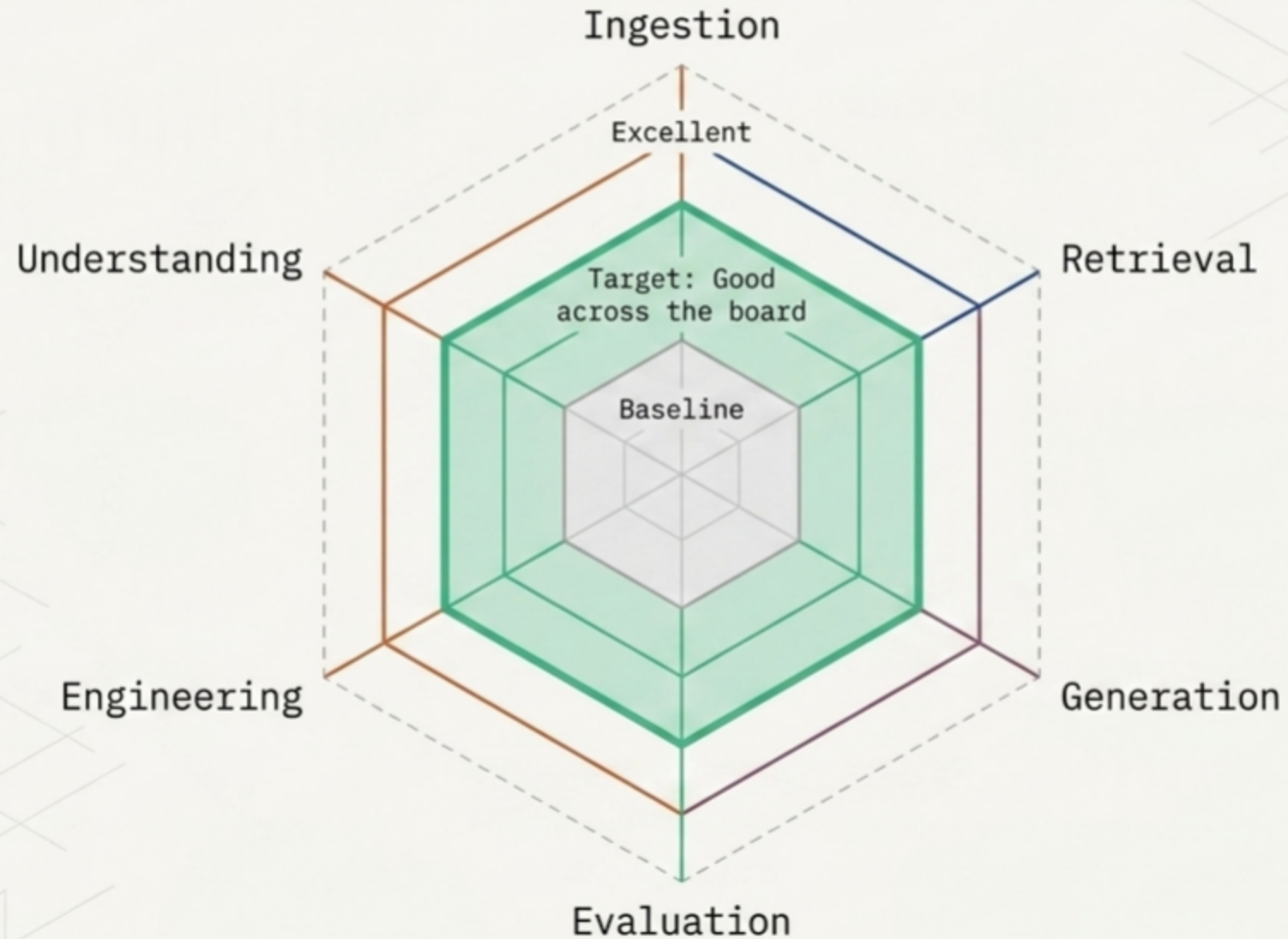
Measure after every single addition. Identify which module provided the biggest lift on your specific data—typically hybrid for mixed queries, or re-rank for precise ordering.



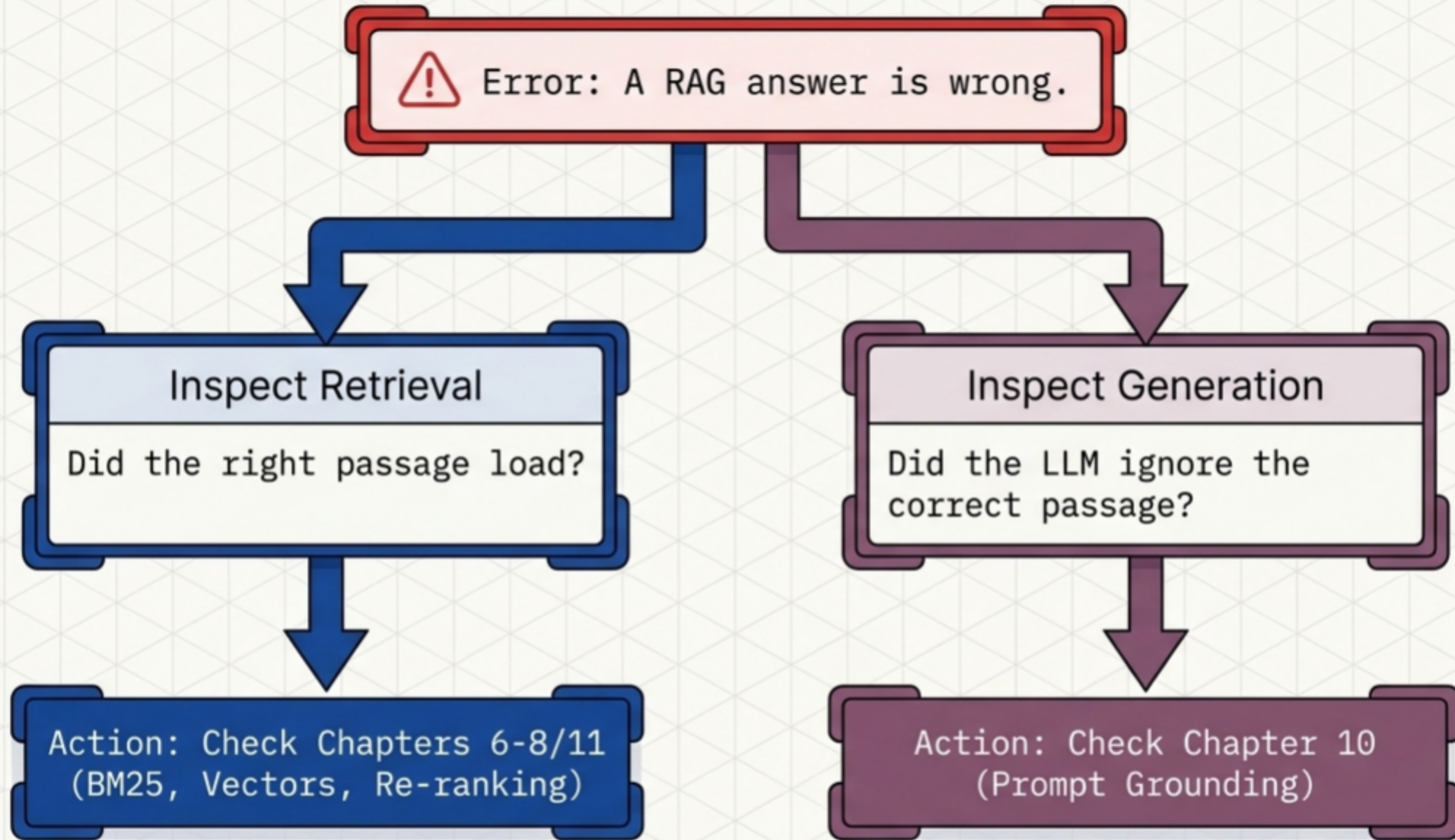


# The Six-Point System Inspection

Before pushing to production, audit the stack. Do not over-optimize a single axis (like Generation) while ignoring critical infrastructure (like Ingestion or Evaluation).







A wrong answer is a symptom, not a root cause. Always verify the retrieval pipeline before rewriting your generator prompts.



# Ship It: Deploying the Integrated Ecosystem

Extend the SearchStack from a notebook script into a portfolio piece.  
Wrap the engine in a minimal UI, connect a persistent database, or expose it as an API. The Answer Engine is complete.

