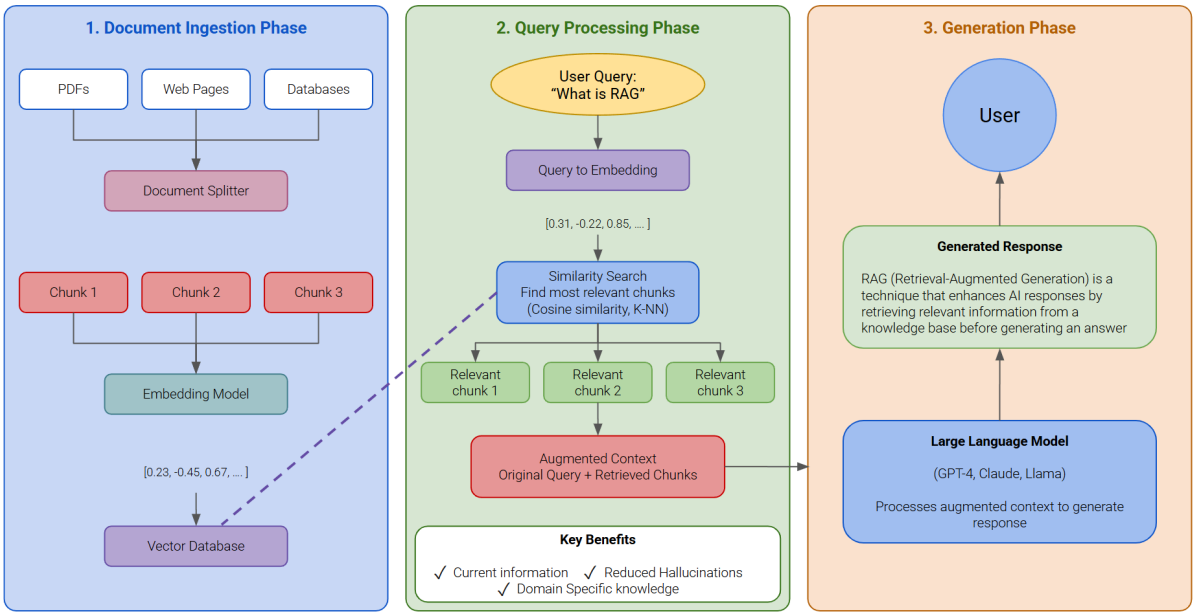
RAG (Retrieval Augmented Generation) Architecture

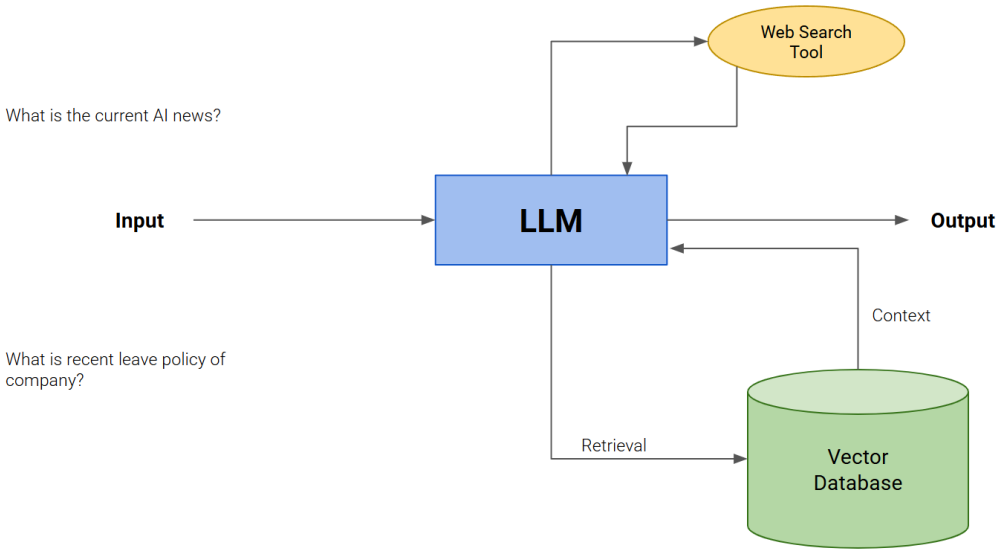


RAG (Retrieval Augmented Generation) is a powerful technique that enhances AI language models by combining their generation capabilities with external knowledge retrieval

What is RAG?

RAG is like giving an AI assistant access to a library while it's answering questions. Instead of relying solely on what it learned during training, the AI can now look up specific, current, or specialized information from external sources before generating its response.

Think of it this way: Traditional language models are like students taking a closed-book exam - they can only use what they memorized. RAG-enabled models are like students in an open book exam - they can reference materials to provide more accurate, detailed, and up-to-date answers



- **Retrieval**: Finding relevant information from vector databases
- **Augmentation**: Enhancing context with metadata
- **Generation**: Producing the answer

Example of RAG

Customer Support without RAG

- **Customer**: What is your return policy for items bought during Black Friday sale?
- **AI:** Generally, most companies offer 30-day returns, but policies may vary…

  [Generic, unhelpful response]

Customer Support with RAG

- **Customer**: What is your return policy for items bought during Black Friday sale?

- **AI:** According to our current policy (Policy Doc v3.2, updated Nov 2024), Black Friday purchases have an extended 60-day return window until January 31st. Items must be unused with original tags. Electronics have a 15-day return period due to rapid depreciation. Would you like me to start a return for a specific item?

Advantages of RAG

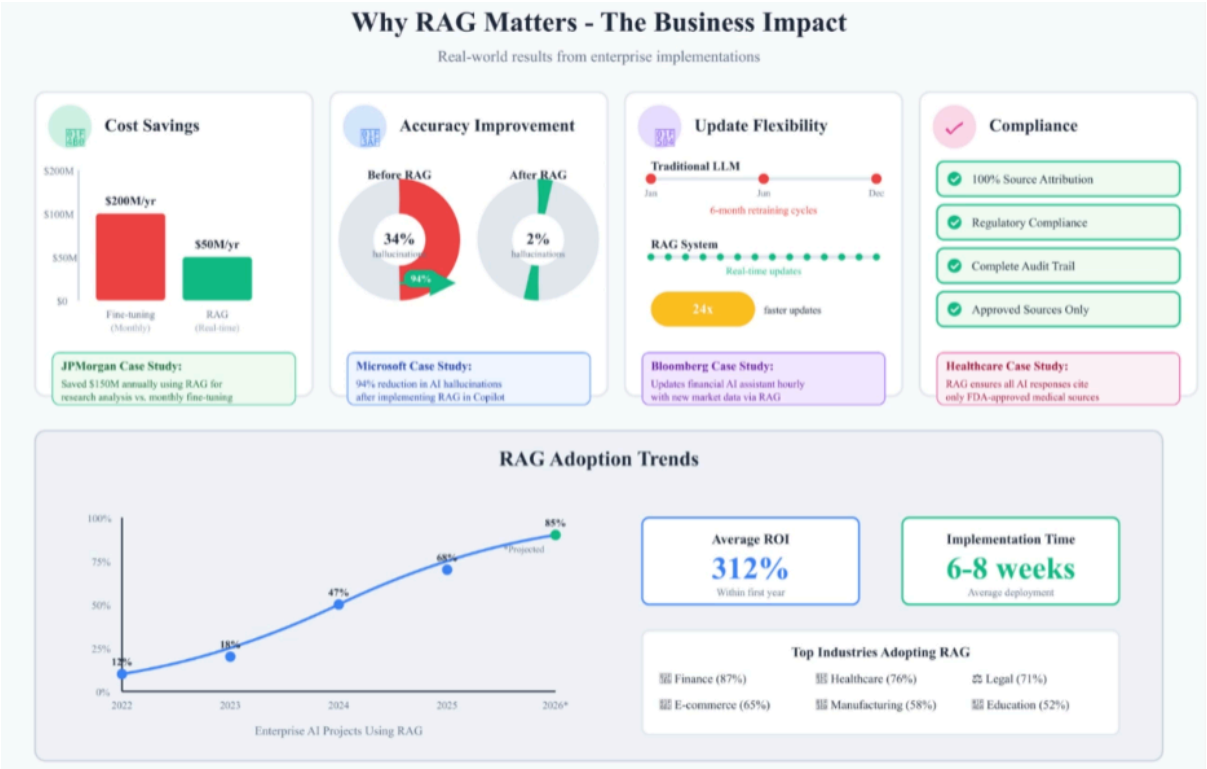
- **Cost Saving**: JPMorgan saved %150M annually by implementing RAG for research analysis instead of fine-tuning models monthly.

- **Accuracy**: Microsoft reported 94% reduction in AI hallucinations after implementing RAG in their Copilot products

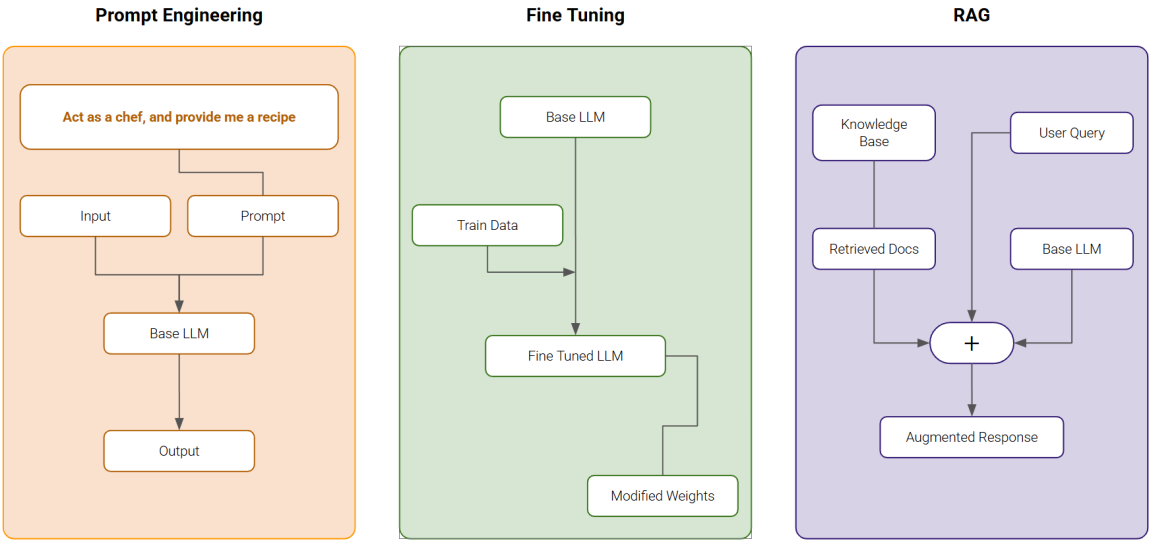
- **Flexibility**: Bloomberg updates their financial AI assistant hourly with new markets data - impossible with traditional LLMs

- **Compliance**: Healthcare companies use RAG to ensure AI responses always cite approved medical sources.

Business Use Cases Impact with RAG



Prompt Engineering vs Fine Tuning vs RAG



Prompt Engineering

1. Specific Instructions
2. Structured Prompt with clear context
3. Model remain unchanged

Fine Tuning

1. Prepare domain specific training data
2. Train model on data
3. Create a specialized version

RAG

1. Store documents in Vector database
2. Retrieved relevant documents for each query
3. LLM generates answer from context

Pros

Prompt Engineering	Fine Tuning	RAG
No technical expertise needed	Deeply specialized knowledge	Always have up-to-date information
Instant results	Consist behaviour	No training required
No training costs	No prompt engineering needed	Cost-effective
Works with any LLM	Better for specific domain	Accuracy is high
		Can handle private/proprietary data

Cons

Prompt Engineering	Fine Tuning	RAG
Limited by models base knowledge	Expensive process	Infrastructure setup
Inconsistent results	Require Machine Learning expertise	Retrieval quality effect results
Token limit restrict complexity	Retraining for updates	Context window limitations
Can not add additional knowledge		

Best For

Prompt Engineering	Fine Tuning	RAG
Small scale applications	Specific style chatbot	Up-to-date information
Generic purpose tasks	High volume data	Accuracy
Quick prototyping	Where accuracy matters	External knowledge

AI Customization Methods: A Beginner's Guide

Understanding Prompt Engineering vs Fine-tuning vs RAG