Just like AI Agents, not every RAG Architecture is the same   
  
But how to identify your perfect RAG Architecture? Let explain...  
  
Depending on your use case, you would only use a few types of RAGs.  
  
To understand that,   
  
📌 Let's break down the popular architectures behind RAGs with their pros and cons:  
  
1. Naive RAG  
  
- Retrieves documents via straightforward embedding similarity.  
- Feeds into LLM to generate an answer.  
  
2. Graph RAG  
  
- Extracts structured knowledge graphs from retrieved text.  
- Uses graph context to enrich the LLM prompt for better reasoning.  
  
3. Hybrid RAG  
  
- Combines standard vector retrieval with graph-based retrieval.  
- Retrieves both dense embeddings and structured graph context.  
  
4. HyDe (Hypothetical Document Embeddings)  
  
- Generates a hypothetical answer document from the user’s query   
- Embeds that hypothetical document and uses it to retrieve real documents.  
  
5. Corrective RAG  
  
- Validates or grades retrieved info via external search (e.g., web).  
- Filters or amends vector DB entries based on validation.  
  
6. Adaptive RAG  
- Analyzes whether the query requires simple or multi-step retrieval.  
- Breaks down complex queries into smaller reasoning steps when needed.  
  
7. Agentic RAG  
- Learn In-depth from here: [**https://lnkd.in/grr5s6\_H**](https://lnkd.in/grr5s6_H)  
  
📌 Best fit Use Cases  
  
- Naive RAG: Best for simple FAQ-style retrieval where direct matching works well.  
  
- Graph RAG: Ideal for exploring complex relationships in structured knowledge bases.  
  
- Hybrid RAG: Great for combining unstructured text and structured graph knowledge in answers.  
  
- HyDe: Useful for retrieving relevant documents when queries are vague or underspecified.  
  
- Corrective RAG: Perfect for ensuring up-to-date or fact-checked information from live sources.  
  
- Adaptive RAG: Suited for handling both straightforward and multi-step, layered questions.  
  
- Agentic RAG: Best for complex tasks needing planning, memory, and multiple data/tool integrations  
  
📌 After months of feedback and iteration, we are finally releasing our first technical cohort, "AI Agent Engineering"  
  
🔗 Enrol here: [**https://lnkd.in/gDEPcXBB**](https://lnkd.in/gDEPcXBB)  
  
If you are a business leader, we've developed frameworks that cut through the hype, including our five-level Agentic AI Progression Framework to evaluate any agent's capabilities in my latest book.  
  
🔗 Book info: [**https://amzn.to/4irx6nI**](https://amzn.to/4irx6nI)  
  
Save 💾 ➞ React 👍 ➞ Share ♻️  
  
& follow for everything related to AI Agents

