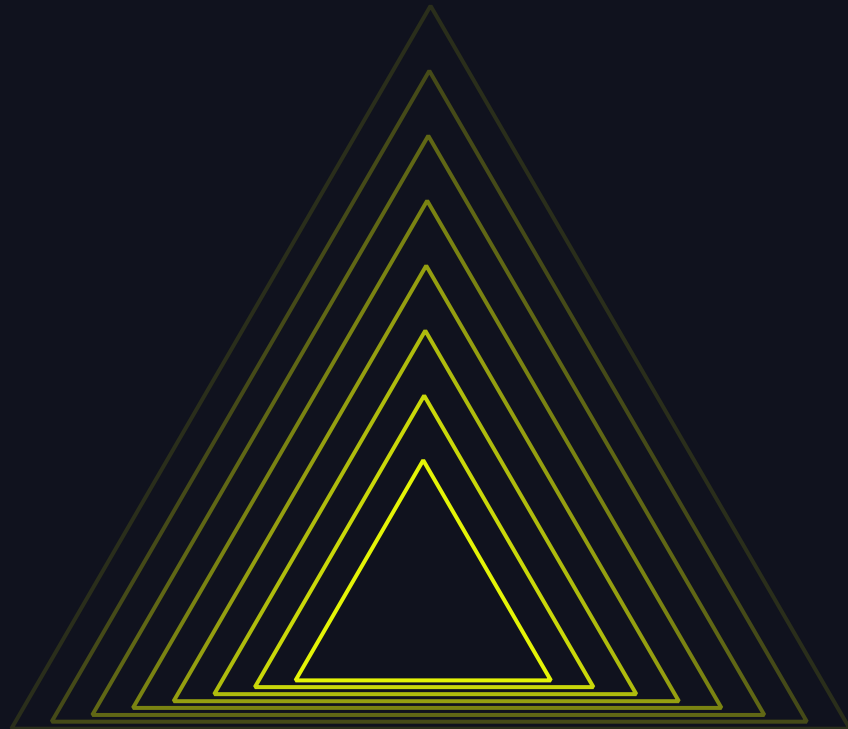




Flow Engineering and Agentic RAG with LangGraph



Erick Friis, LangChain AI
6/12/2024

Agenda

What will we learn today?



Background

- What is LangChain?
- What is flow engineering?
- ...



Build

- How do I get started with LangGraph?
- What are LangGraph graphs, state models, nodes, and edges?
- ...



What's Next

- Where can I follow along?
- Where can I get started with LangGraph?

Context

Raise your hand if you know what it is

- Chat models
- Tool calling models
- LangChain
- LangSmith
- LangChain Expression Language
- Flow Engineering

Context

- Chat models
- Tool calling models
- LangChain
- LangSmith
- LangChain Expression Language
- Flow Engineering

System: You are a helpful assistant

Human: What is the capital of France?

AI: Paris

Human: And what is its population?

AI: Paris' population is approximately 2.28 million

Context

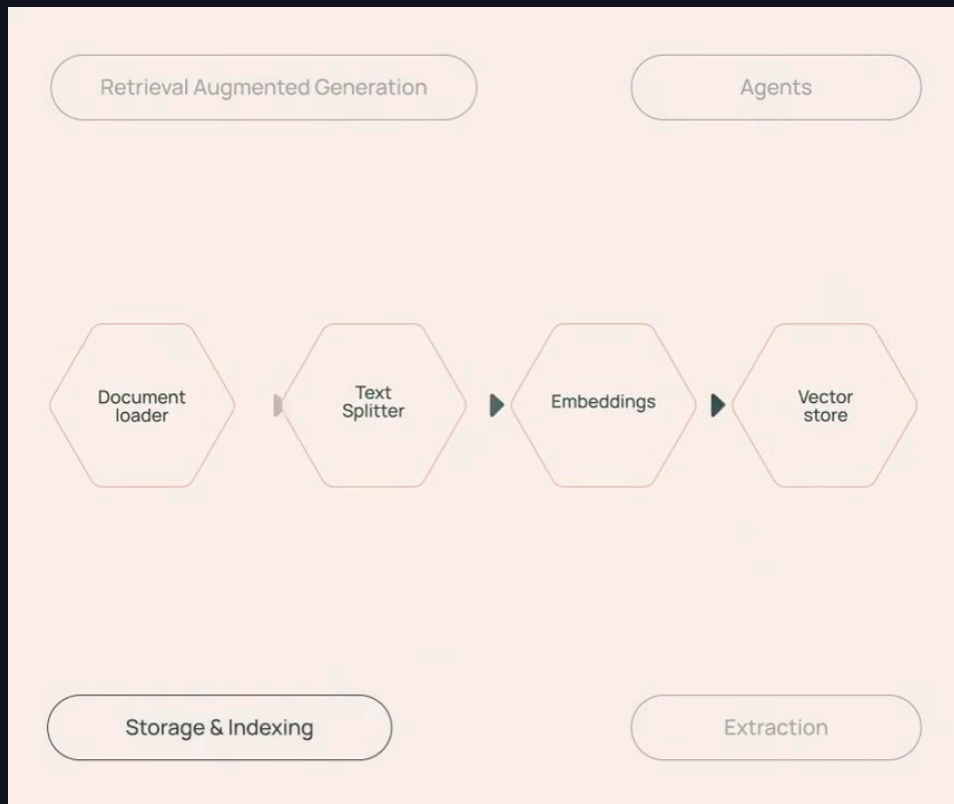
- Chat models
- **Tool calling models**
- LangChain
- LangSmith
- LangChain Expression Language
- Flow Engineering

System: You are a helpful assistant	Tools: - CityInfo: population: int
Human: What is the capital of France?	
AI: Paris	
Human: And what is its population?	

```
CityInfo: {"population": 2280000}
```

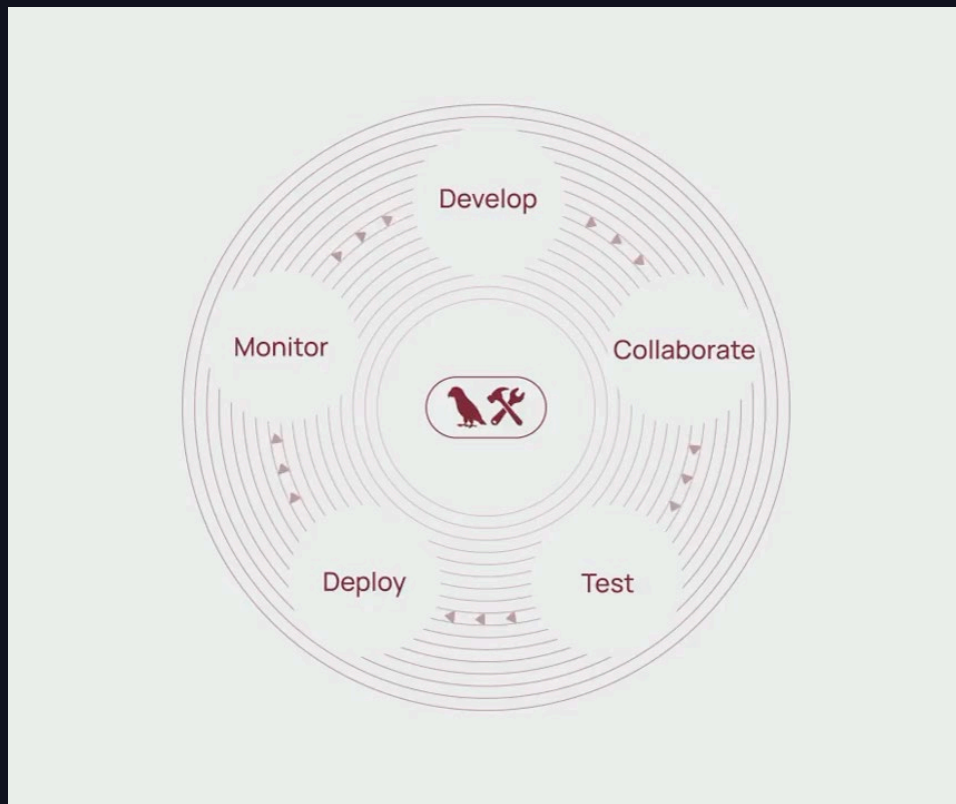
Context

- Chat models
- Tool calling models
- **LangChain**
- LangSmith
- LangChain Expression Language
- Flow Engineering



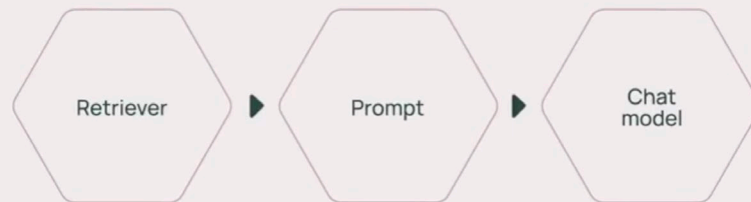
Context

- Chat models
- Tool calling models
- LangChain
- **LangSmith**
- LangChain Expression Language
- Flow Engineering



Context

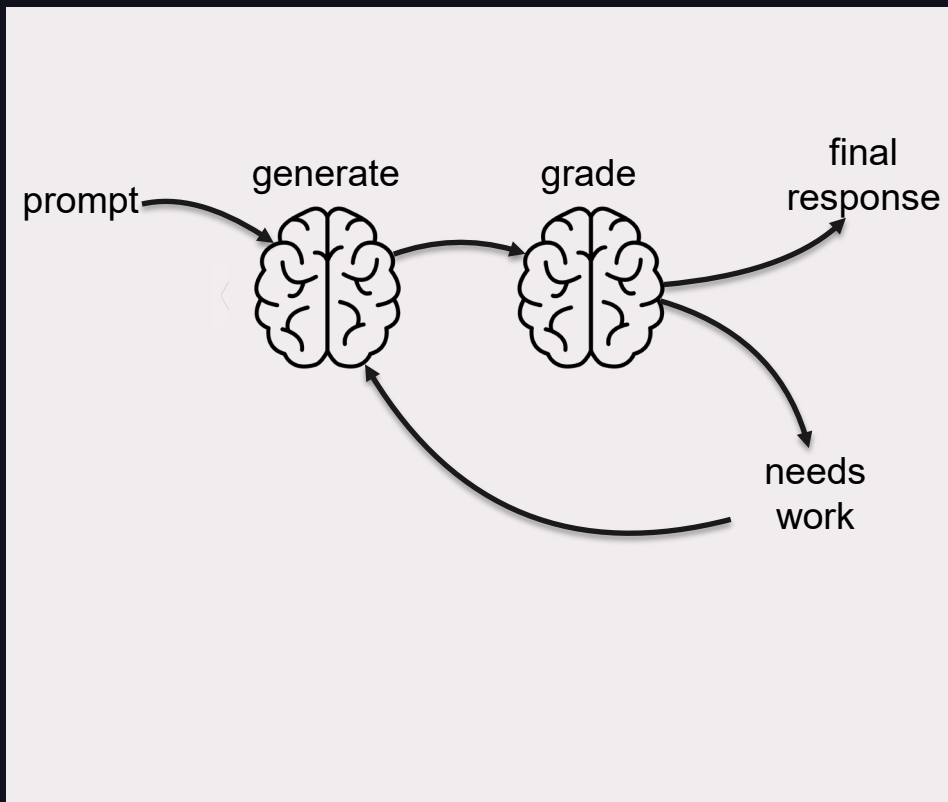
- Chat models
- Tool calling models
- LangChain
- LangSmith
- **LangChain Expression Language**
- Flow Engineering



```
chain = retriever | prompt | chat_model
```


Context

- Chat models
- Tool calling models
- LangChain
- LangSmith
- LangChain Expression Language
- **Flow Engineering**



BUILD



FLOW ENGINEERING

How to define a StateGraph

Defining State

- Which variables do you want to track across your graph's execution?
- Which intermediate artifacts are useful for debugging?

Defining Nodes

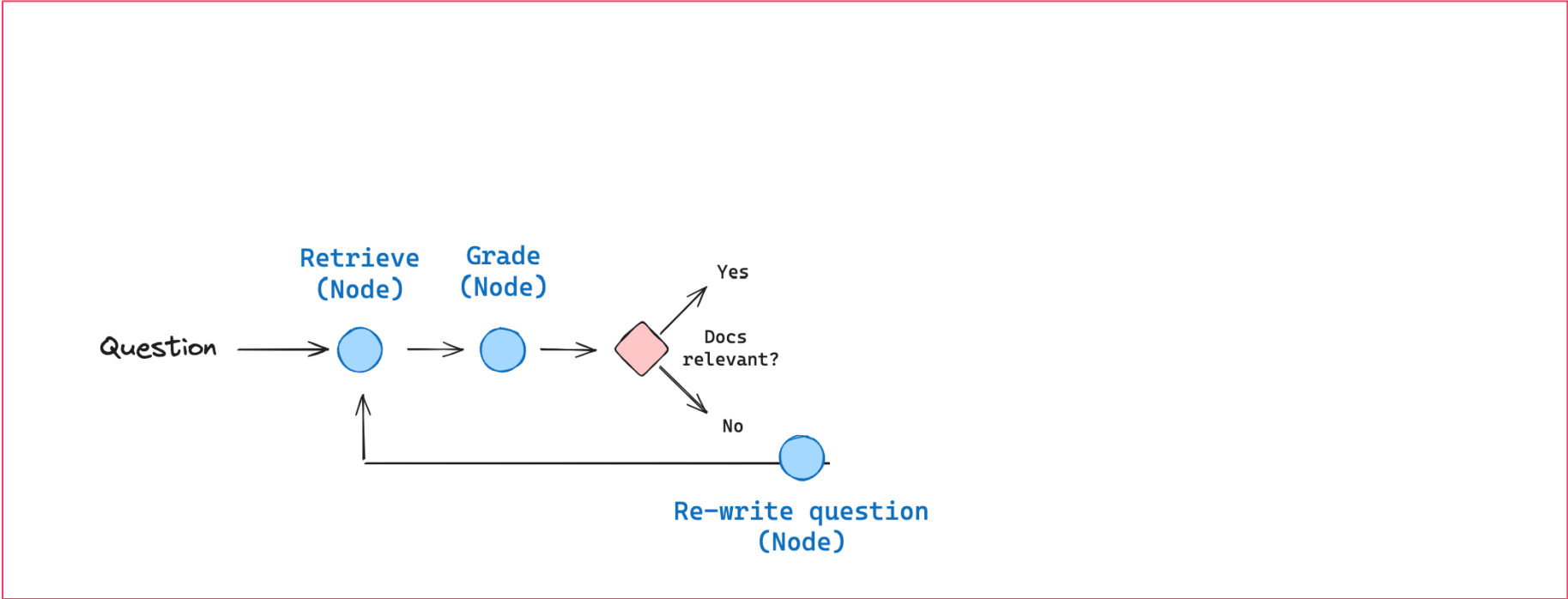
- How can we separate our agent logic?
- Which operations are useful to separate in order to improve debugging?

Connecting with Edges

- Which edges and conditional edges should dictate our graph's execution?

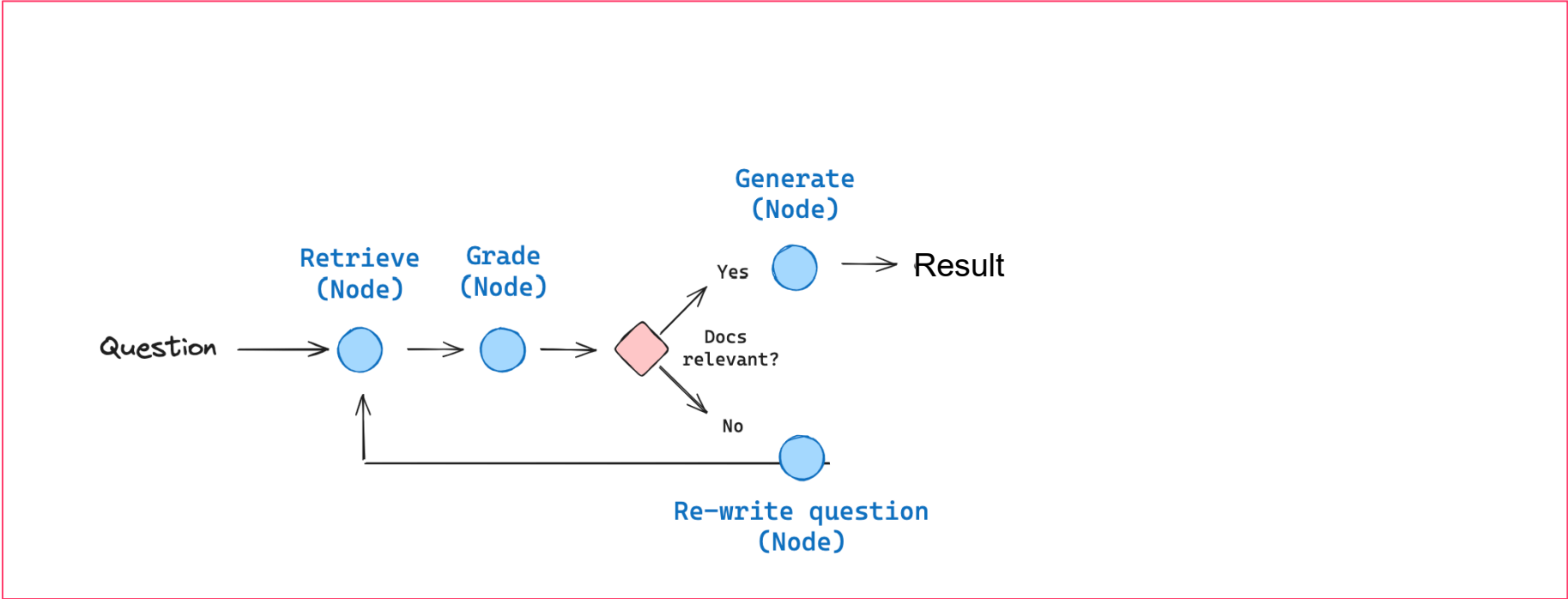
GRADED RETRIEVAL

A Simple Agent



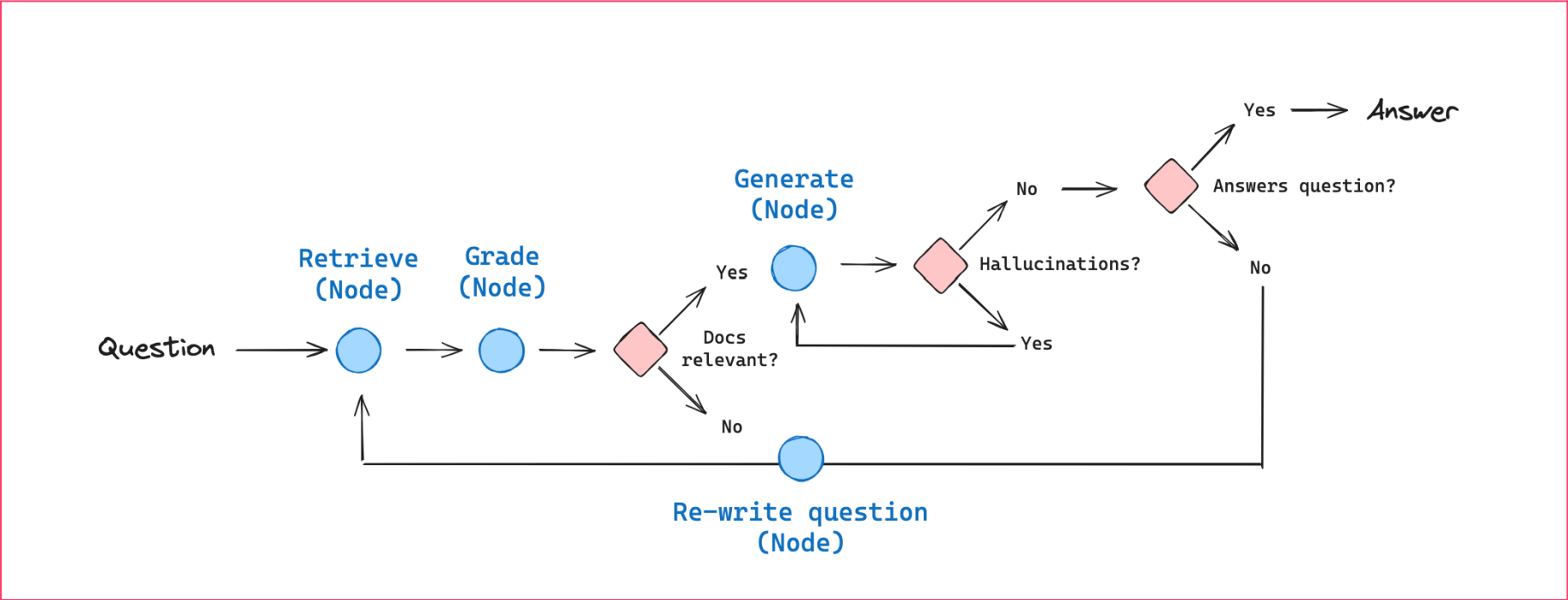
GRADED RETRIEVAL RAG

Adding a second grader



Extension: SELF-RAG

Completing the loop



LangChain LangGraph

Home Tutorials How-to Guides Conceptual Guides Reference

Tutorials

- Intro to LangGraph
- Use cases
- Chatbots >
- Multi-Agent Systems >
- RAG >
 - Langgraph adaptive rag
 - Langgraph adaptive rag local
 - Langgraph agentic rag
 - Langgraph crag
 - Langgraph crag local
 - Langgraph self rag
 - Langgraph self rag local
- Web Research (STORM) >
- Planning Agents >
- Reflection & Critique >
- Evaluation & Analysis >
- Text Mining >
- Web Navigation
- Competitive Programming

Adaptive RAG

Adaptive RAG is a strategy for RAG that unites (1) [query analysis](#) with (2) [active / self-corrective RAG](#).

In the [paper](#), they report query analysis to route across:

- No Retrieval
- Single-shot RAG
- Iterative RAG

Let's build on this using LangGraph.

In our implementation, we will route between:

- Web search: for questions related to recent events
- Self-corrective RAG: for questions related to our index

```

    graph LR
      Q[Question] --> QA[Query Analysis]
      QA -- "[related to index]" --> RAG
      QA -- "[unrelated to index]" --> WS[Web search]
      QA -.-> OR[Optional]
      WS --> AW[Answer w/ web search]
      RAG --> R[Retrieve Node]
      R --> G[Generate Node]
      G --> M[Is this answer?]
      M -- No --> R
      M -- Yes --> RQ[Re-write question Node]
      RQ --> R
      M --> A[Answer]
      OR -.-> O[Optional]
  
```

WHAT 'S NEXT?



WHAT'S NEXT

Future work

- Resources to get started with LangGraph:
 - LangGraph Tutorials: <https://langchain-ai.github.io/langgraph/tutorials/>
 - LangChain YouTube: <https://www.youtube.com/@LangChain>
- Where to follow along: [@LangChainAI on Twitter/X](#), LangChain on LinkedIn
- Coming Soon to LangGraph:
 - Hosting
 - More Examples
- Connect with me [@eyfriis on Twitter/X](#)

