THE ART OF RAG EVALUATION

AI MAKERSPACE

ragas

Presented by:
Dr. Greg & The Wiz 🪄

🎯 ALIGNING OUR AIM

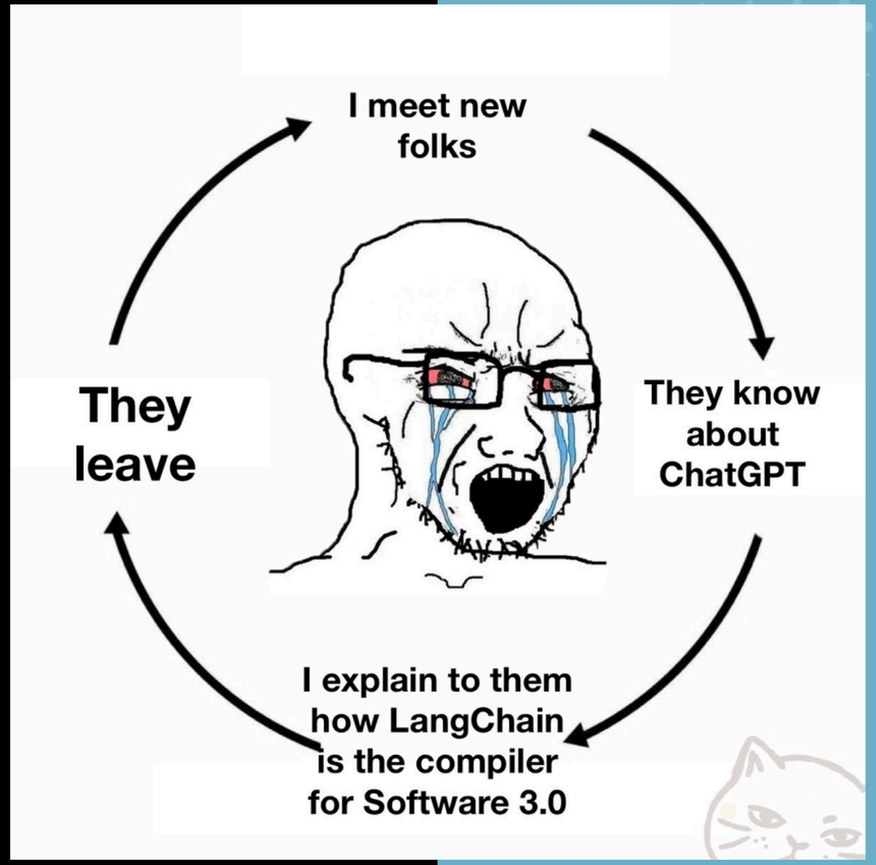
AI
MAKERSPACE

BY THE END OF THE SESSION

- What's new in **LangChain v0.1**
- How to **build** and **evaluate a RAG application**
- How to **improve RAG** with **advanced retrieval**

OVERVIEW

- 🔗 LangChain v0.1.0
- 📚 Retrieval Augmented Generation (RAG)
- 📐 RAG ASessment (RAGAS)
- 👷 Build, Evaluate, Improve
- 🎯 Conclusions & QA



v0.1.0

langchain-core

- Main abstractions, interfaces, functionality

langchain-**community**

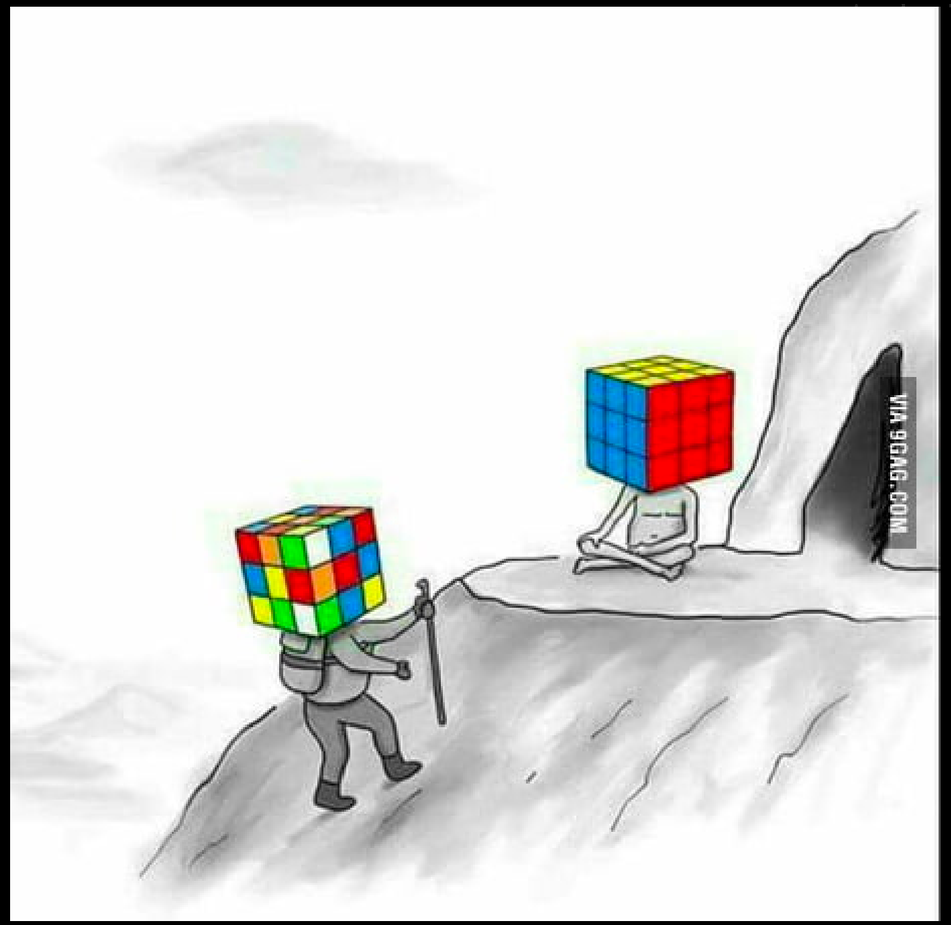
- All third-party integrations

v0.**X**.0

- **Breaking changes ++**

v0.1.**X**

- Bug fixes, new features ++




LangChain v0.1.0

Enabling LLM applications that leverage **context** and **reasoning**.

FOCUS AREAS

👀 Observability

↔ ~700 Integrations

🔗 Composability w/ **LCEL**

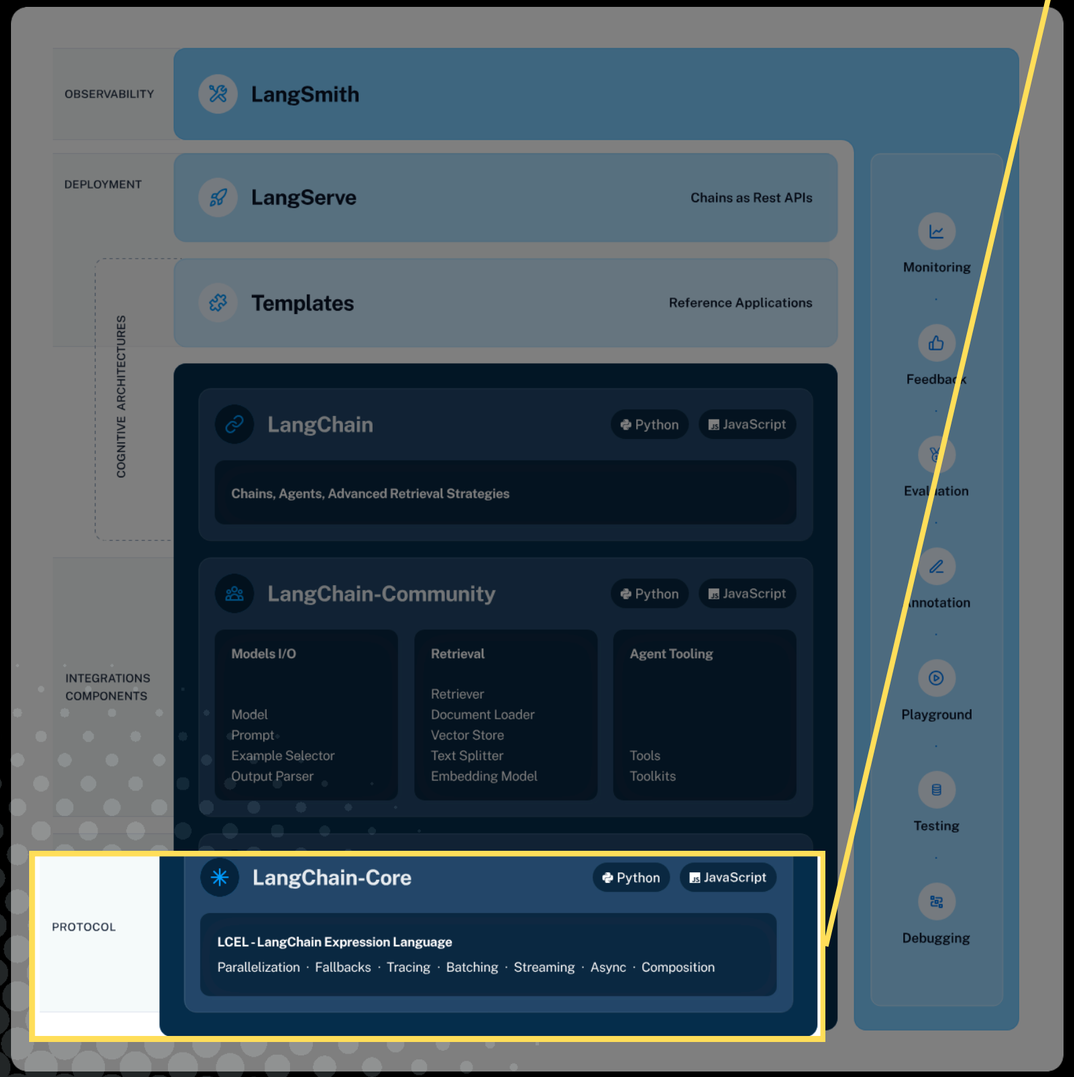
🎏 Streaming Support

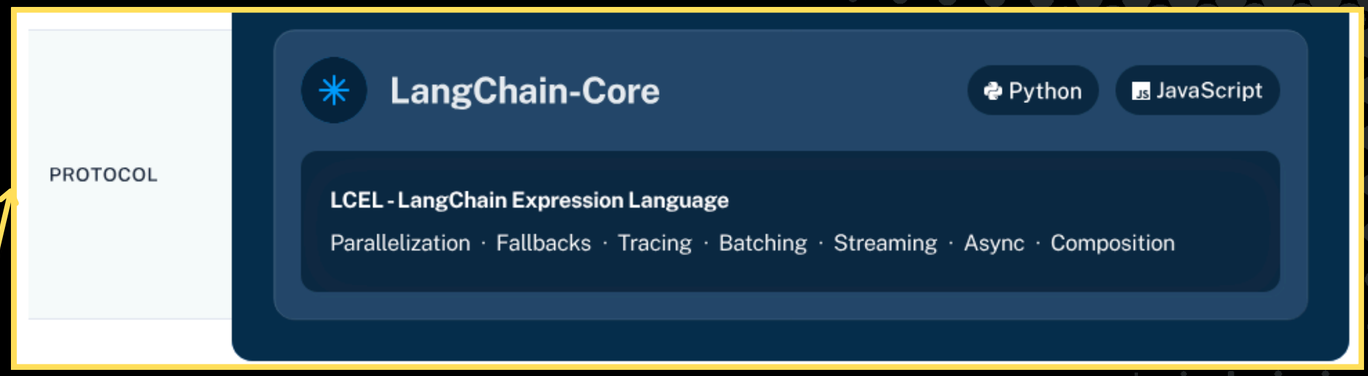
🧱 Output parsing

🔎 **Retrieval**

🤖 Tools + Agents

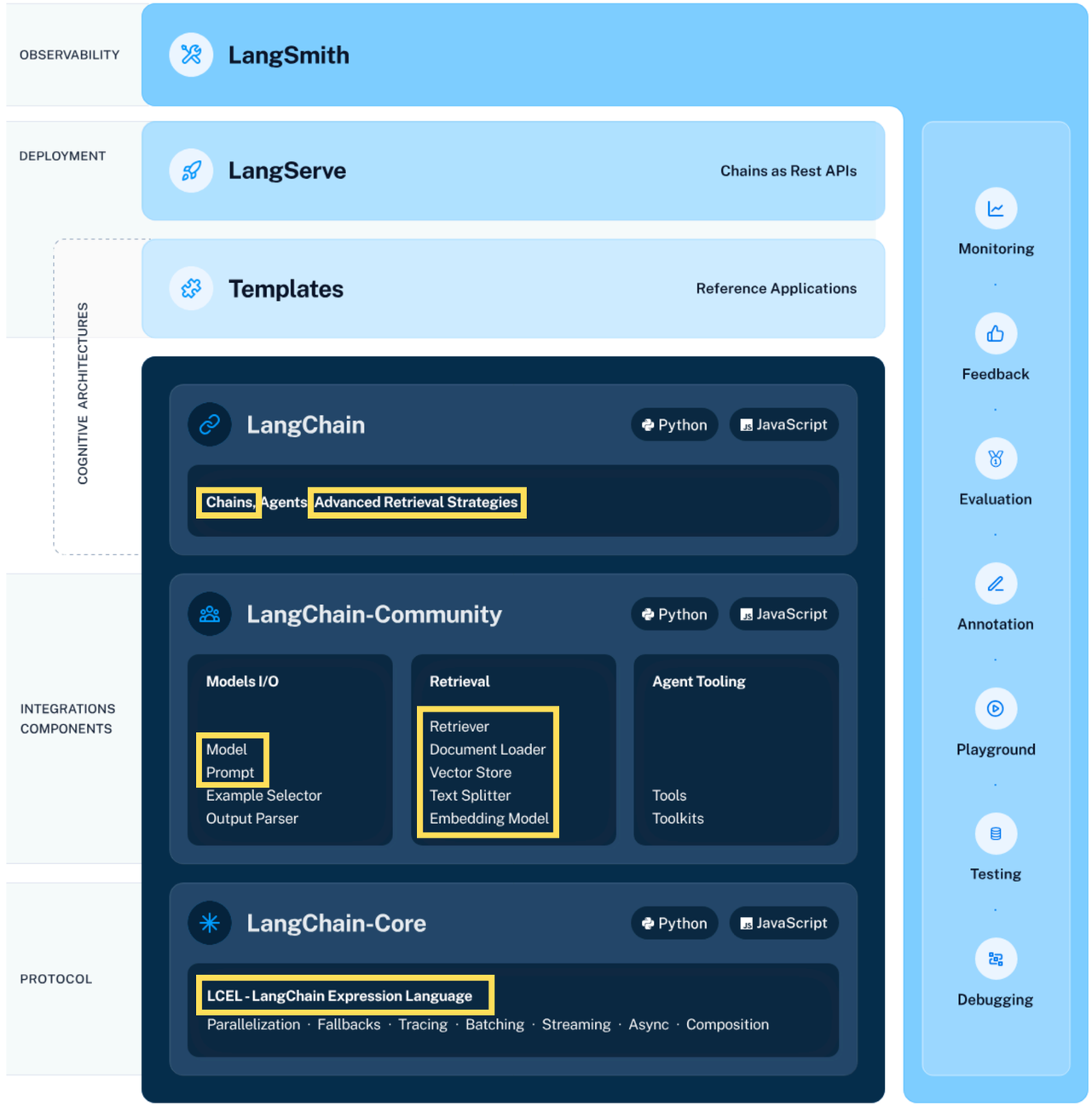
LCEL





- *Declarative way to **easily compose chains** with elegance*
- *Put prototypes into **production with no code changes***

```
llm_chain = prompt | llm
```


OBSERVABILITY
LangSmith
DEPLOYMENT
LangServe
Chains as Rest APIs
COGNITIVE ARCHITECTURES
Templates
Reference Applications
LangChain
Python
JavaScript
Chains, Agents, Advanced Retrieval Strategies
INTEGRATIONS
COMPONENTS
LangChain-Community
Python
JavaScript
Models I/O
Model
Prompt
Example Selector
Output Parser
Retrieval
Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model
Agent Tooling
Tools
Toolkits
PROTOCOL
LangChain-Core
Python
JavaScript
LCEL - LangChain Expression Language
Parallelization · Fallbacks · Tracing · Batching · Streaming · Async · Composition
Monitoring
Feedback
Evaluation
Annotation
Playground
Testing
Debugging

RAG


OBSERVABILITY
LangSmith
DEPLOYMENT
LangServe
Chains as Rest APIs
COGNITIVE ARCHITECTURES
Templates
Reference Applications
LangChain
Python
JavaScript
Chains
Agents
Advanced Retrieval Strategies
INTEGRATIONS
COMPONENTS
LangChain-Community
Python
JavaScript
Models I/O
Model
Prompt
Example Selector
Output Parser
Retrieval
Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model
Agent Tooling
Tools
Toolkits
PROTOCOL
LangChain-Core
Python
JavaScript
LCEL - LangChain Expression Language
Parallelization · Fallbacks · Tracing · Batching · Streaming · Async · Composition
Monitoring
Feedback
Evaluation
Annotation
Playground
Testing
Debugging

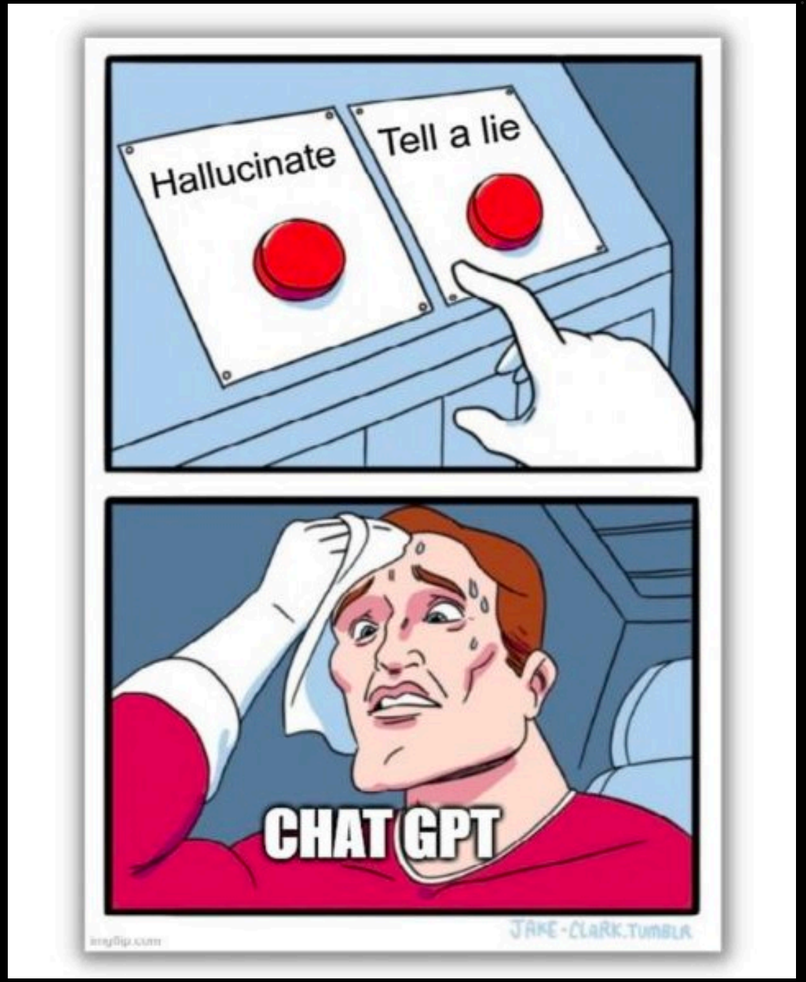
IMPROVE RAG


OBSERVABILITY
LangSmith
DEPLOYMENT
LangServe
Chains as Rest APIs
COGNITIVE ARCHITECTURES
Templates
Reference Applications
LangChain
Python
JavaScript
Chains
Agents
Advanced Retrieval Strategies
INTEGRATIONS
COMPONENTS
LangChain-Community
Python
JavaScript
Models I/O
Model
Prompt
Example Selector
Output Parser
Retrieval
Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model
Agent Tooling
Tools
Toolkits
PROTOCOL
LangChain-Core
Python
JavaScript
LCEL - LangChain Expression Language
Parallelization · Fallbacks · Tracing · Batching · Streaming · Async · Composition
Monitoring
Feedback
Evaluation
Annotation
Playground
Testing
Debugging

Retrieval Augmented Generation (RAG)

HALLUCINATIONS

Confident responses that are **false**.



FACT CHECKING WITH RAG

Retrieval

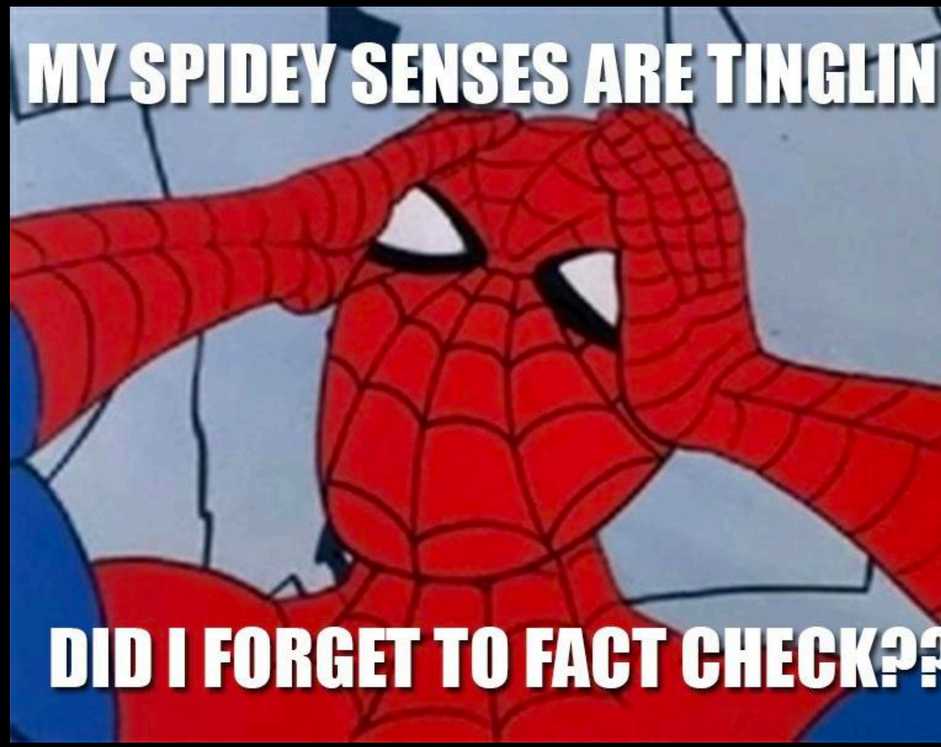
- **Find references** in your docs

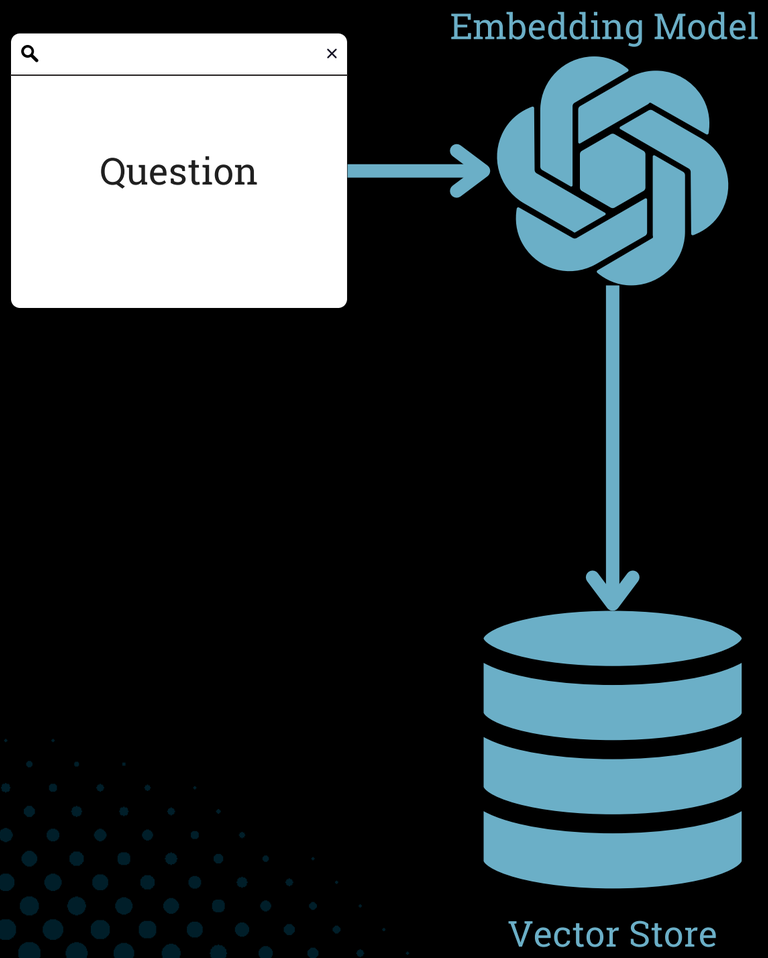
Augmented

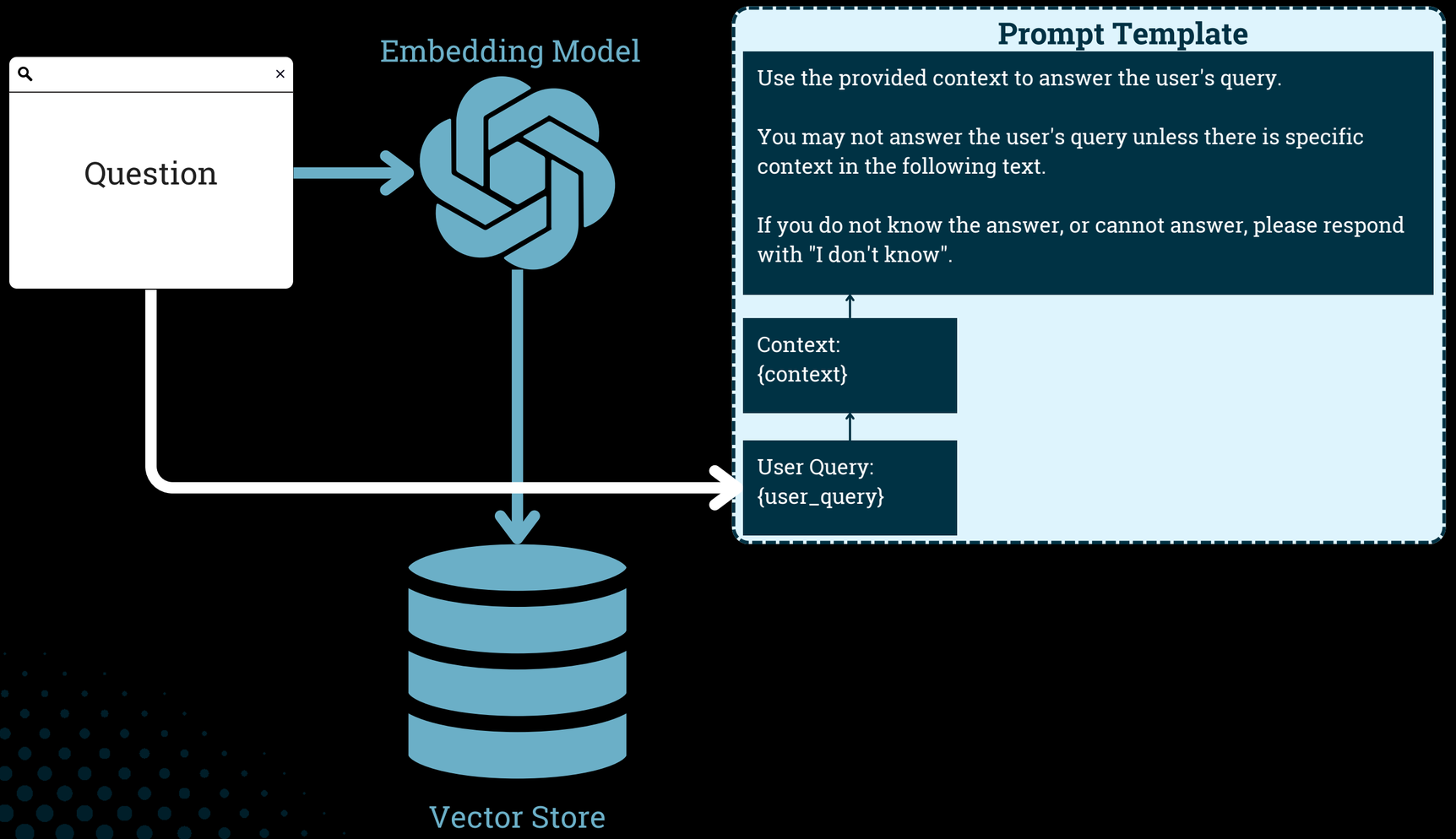
- **Add references** to your prompts

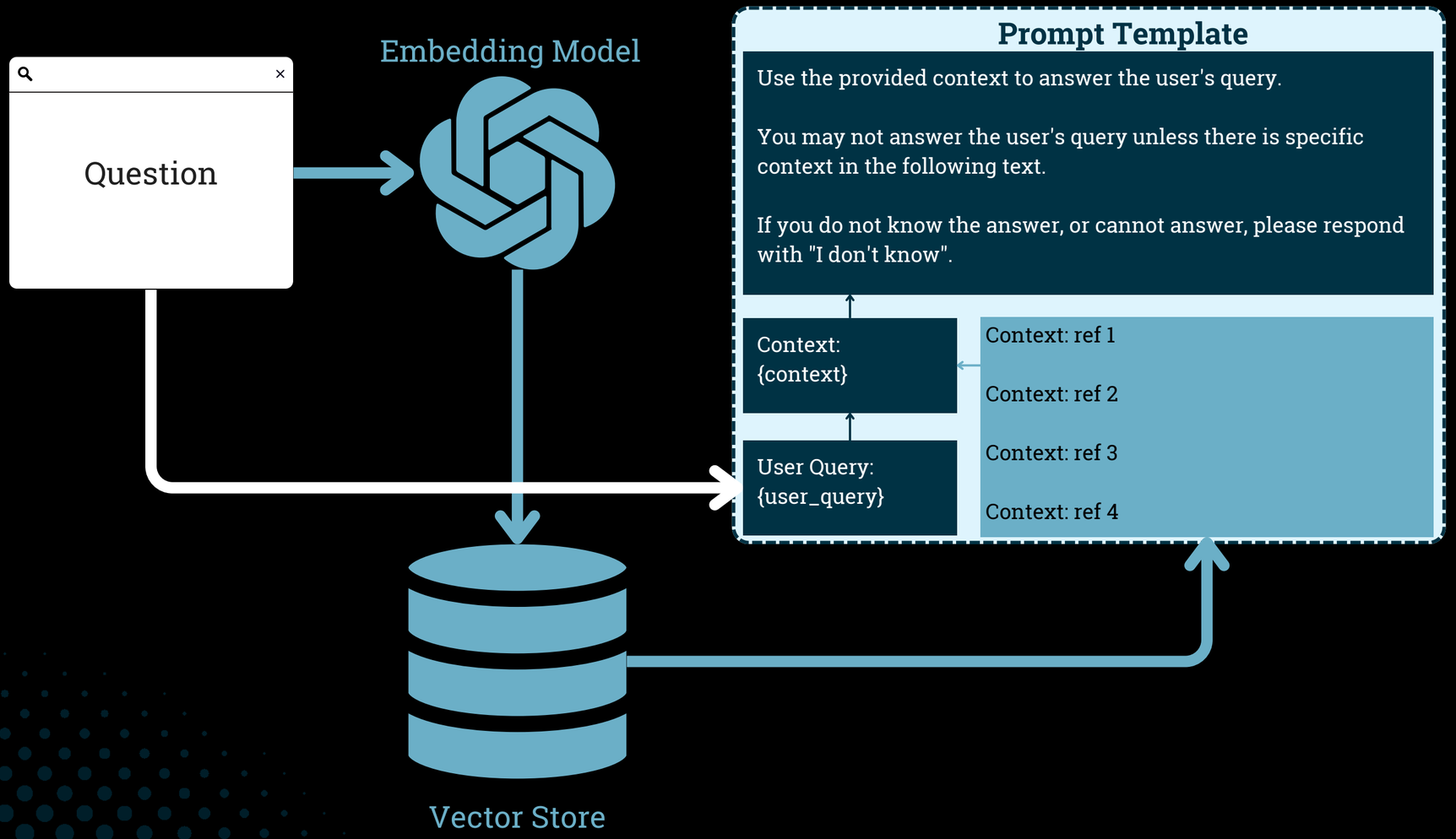
Generation

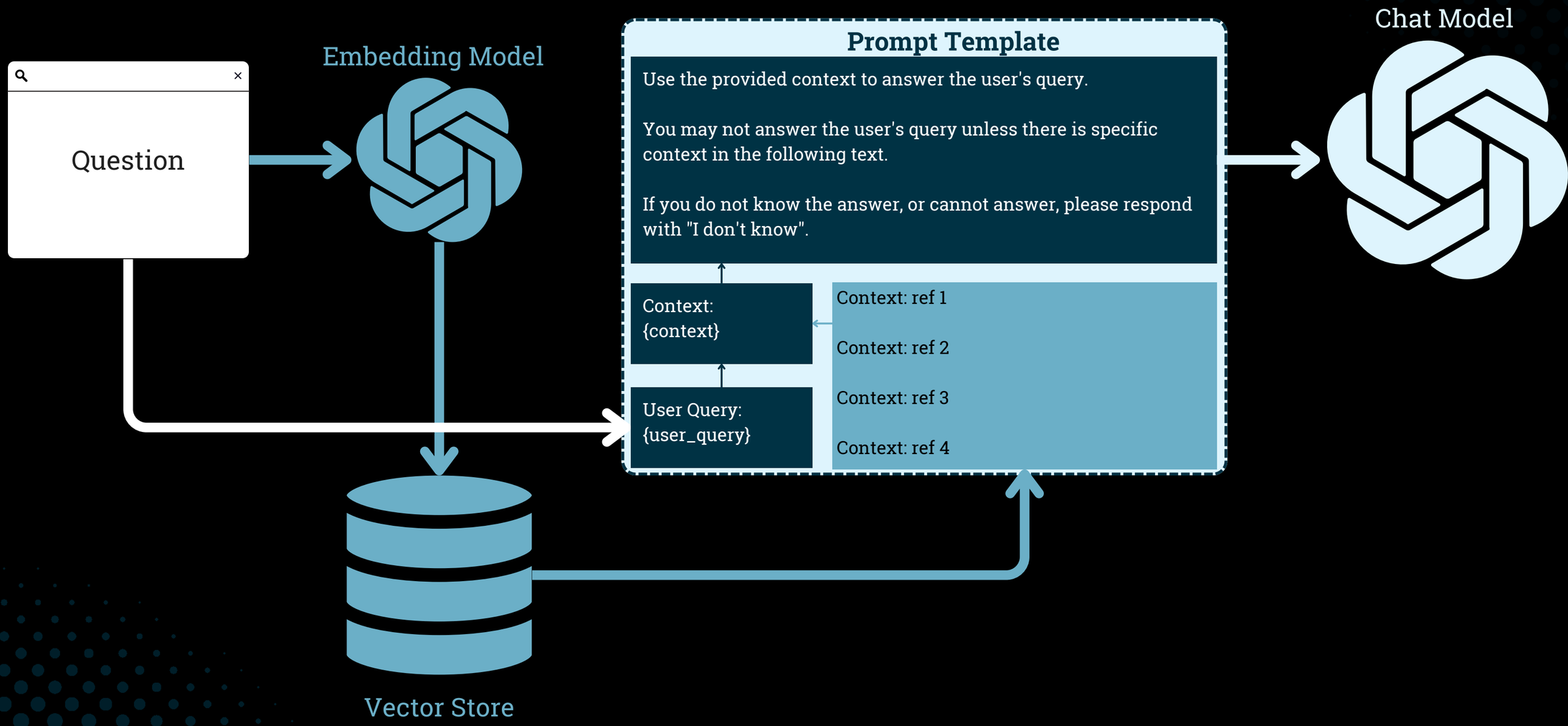
- **Improve answers** to your questions!

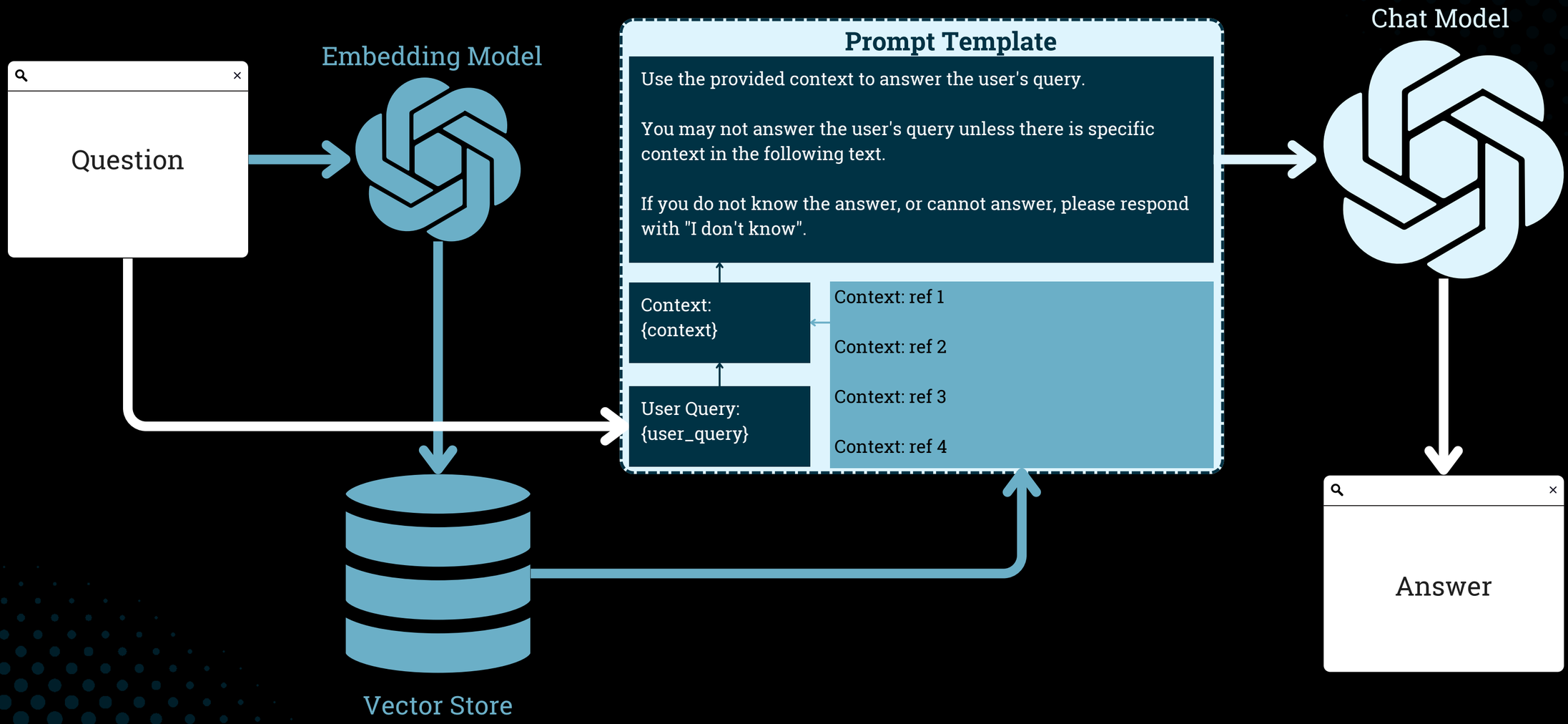



Embedding Model
Question
Vector Store


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context:
{context}
User Query:
{user_query}
Vector Store


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context: {context}
User Query: {user_query}
Context: ref 1
Context: ref 2
Context: ref 3
Context: ref 4
Vector Store


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context: {context}
User Query: {user_query}
Context: ref 1
Context: ref 2
Context: ref 3
Context: ref 4
Chat Model
Vector Store


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context: {context}
User Query: {user_query}
Context: ref 1
Context: ref 2
Context: ref 3
Context: ref 4
Chat Model
Answer
Vector Store

RAG


OBSERVABILITY
LangSmith
DEPLOYMENT
LangServe
Chains as Rest APIs
COGNITIVE ARCHITECTURES
Templates
Reference Applications
LangChain
Python
JavaScript
Chains, Agents, Advanced Retrieval Strategies
INTEGRATIONS
COMPONENTS
LangChain-Community
Python
JavaScript
Models I/O
Model
Prompt
Example Selector
Output Parser
Retrieval
Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model
Agent Tooling
Tools
Toolkits
PROTOCOL
LangChain-Core
Python
JavaScript
LCEL - LangChain Expression Language
Parallelization · Fallbacks · Tracing · Batching · Streaming · Async · Composition
Monitoring
Feedback
Evaluation
Annotation
Playground
Testing
Debugging

MODELS I/O

OpenAI

LangChain

*"**Chat**-Style"*

*Fine-tuned for **chat***

*Often **instruction**-tuned*

System Message	System Message
User Message	**Human** Message
Assistant Message	**AI** Message

PROMPTS

Prompt Template

```
Use the provided context to answer the user's query.

You may not answer the user's query unless there is
specific context in the following text.

If you do not know the answer, or cannot answer,
please respond with "I don't know".
```

```
Context:
{context}
```

```
User Query:
{user_query}
```

```
Context: ref 1

Context: ref 2

Context: ref 3

Context: ref 4
```

CREATE A VECTOR STORE

1. Load Docs
2. Split Text
3. Embed
4. Store Vectors

Retrieval

Retriever
Document Loader
Vector Store
Text Splitter
Embedding Model

Our RAG Build

MODELS

Embedding Model

- e.g,. OpenAI Ada

Chat Model (e.g., LLM)

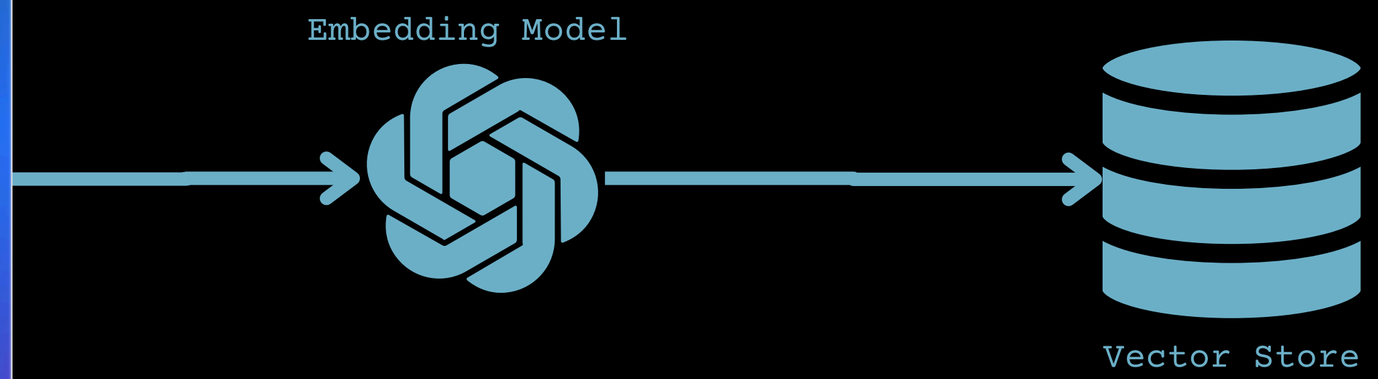
- e.g., OpenAI GPT-3.5-Turbo

Embedding Model

Chat Model

DATA

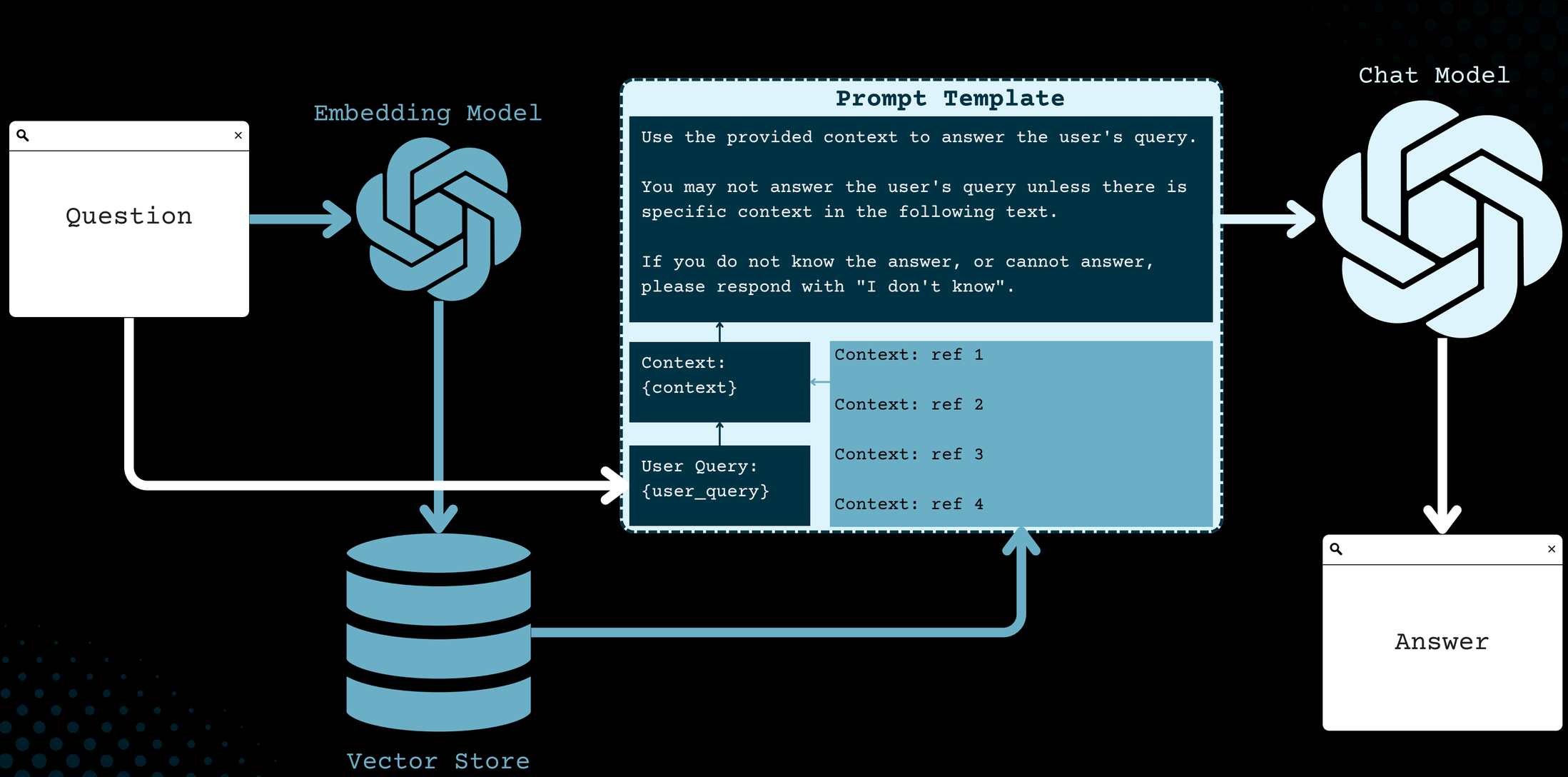

LangChain
v0.1.0
LangChain v0.1.0
Today we're excited to announce the release of langchain 0.1.0, our first stable version. It is fully backwards compatible, comes in both Python and JavaScript, and comes with improved focus through both functional...
LangChain Blog / Feb 9

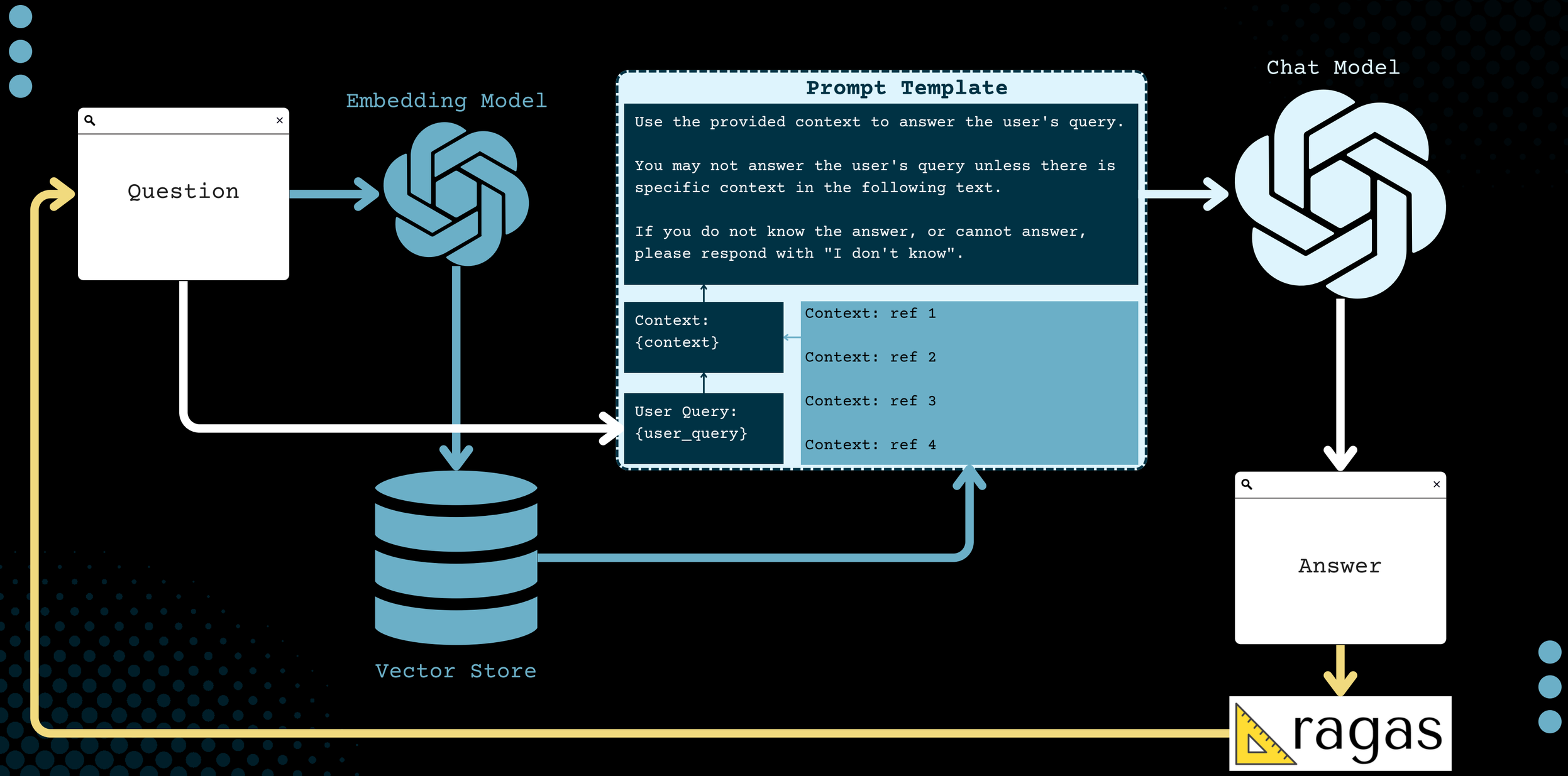

Embedding Model
Vector Store

LangChain v0.1.0 RAG

Presented by
Chris Alexiuk, LLM Wizard 🪄

📐 RAG ASessment (RAGAS)


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context: {context}
User Query: {user_query}
Context: ref 1
Context: ref 2
Context: ref 3
Context: ref 4
Chat Model
Answer
Vector Store


Question
Embedding Model
Prompt Template
Use the provided context to answer the user's query.
You may not answer the user's query unless there is specific context in the following text.
If you do not know the answer, or cannot answer, please respond with "I don't know".
Context: {context}
User Query: {user_query}
Context: ref 1
Context: ref 2
Context: ref 3
Context: ref 4
Chat Model
Answer
Vector Store
ragas

RAG EVALUATION

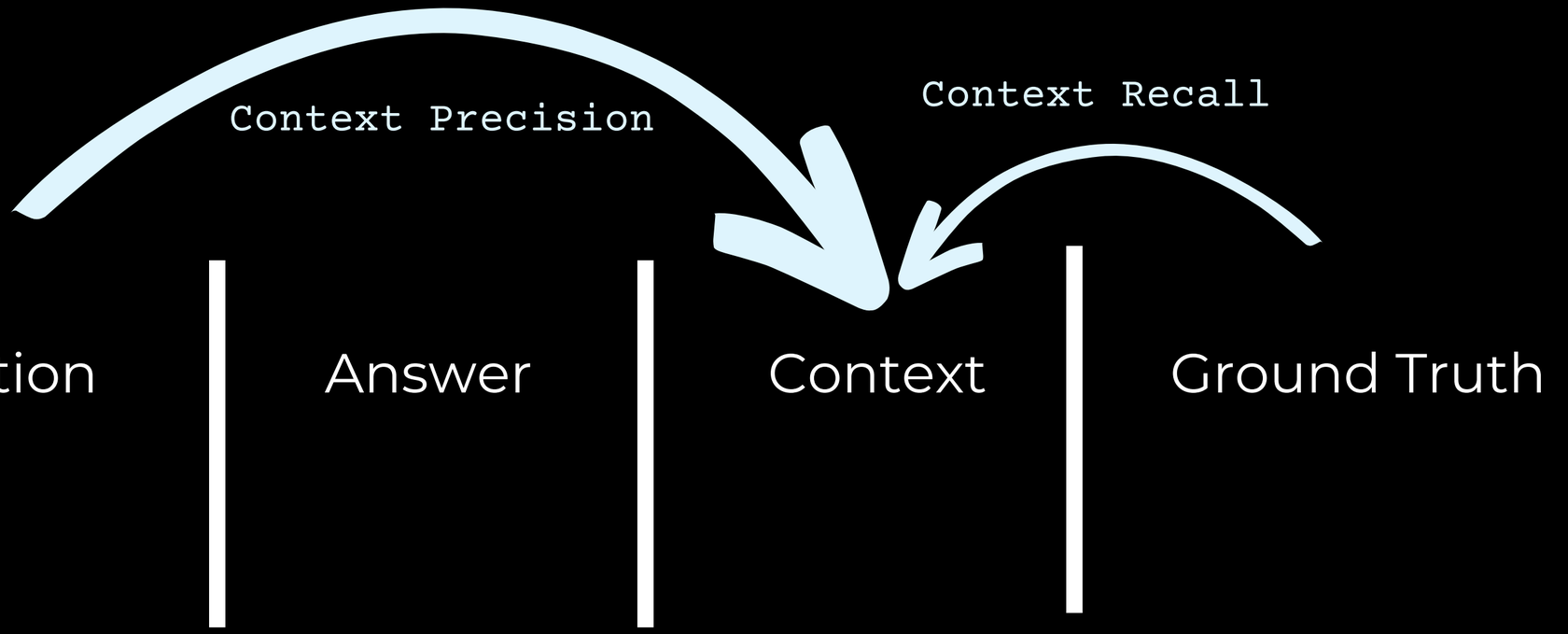
Question | Answer | Context | Ground Truth

RAG EVALUATION

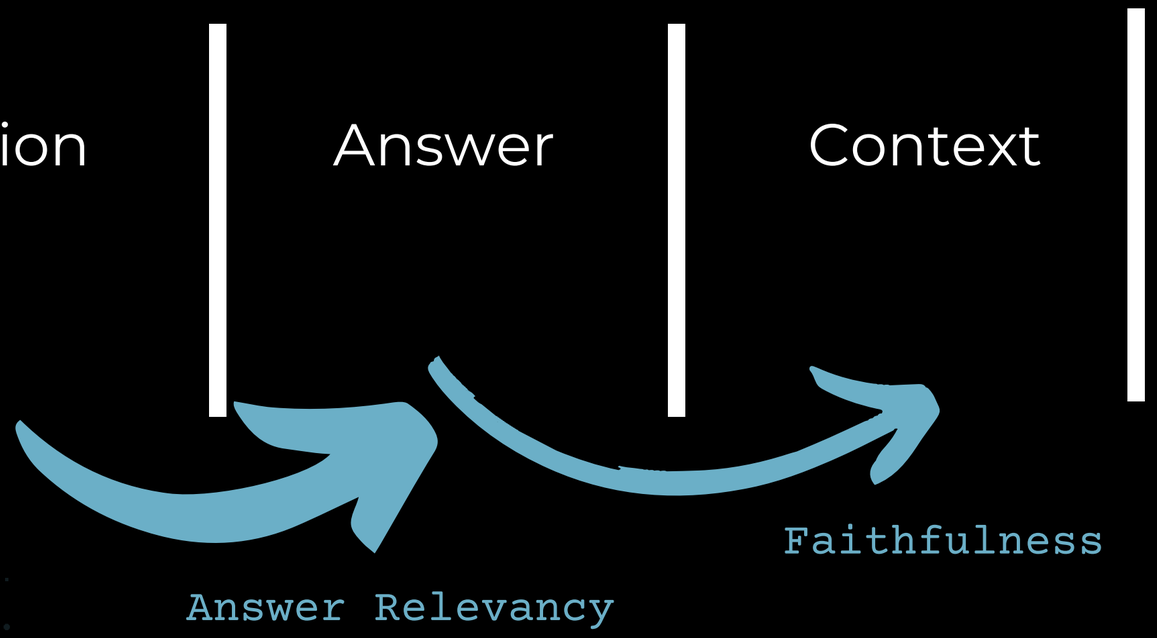
Question | Answer | Context | Ground Truth

RETRIEVAL EVAL

Retrieval



GENERATION EVAL



Generation

RAG EVAL

Retrieval

Context Precision

Context Recall

Question | Answer | Context | Ground Truth

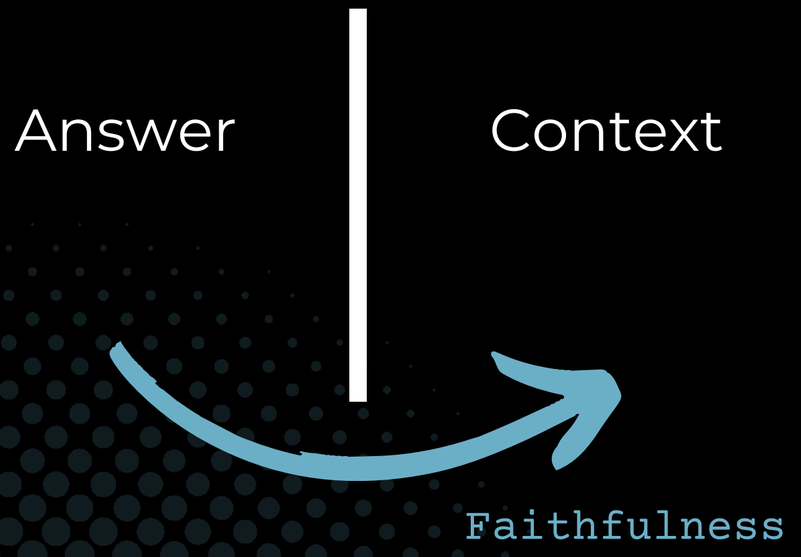
Faithfulness

Answer Relevancy

Generation

FAITHFULNESS

- **Measures** *the factual consistency* of the generated answer against the given context.
- (0,1), Higher -> better



$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}$$

Hint

Question: Where and where was Einstein born?

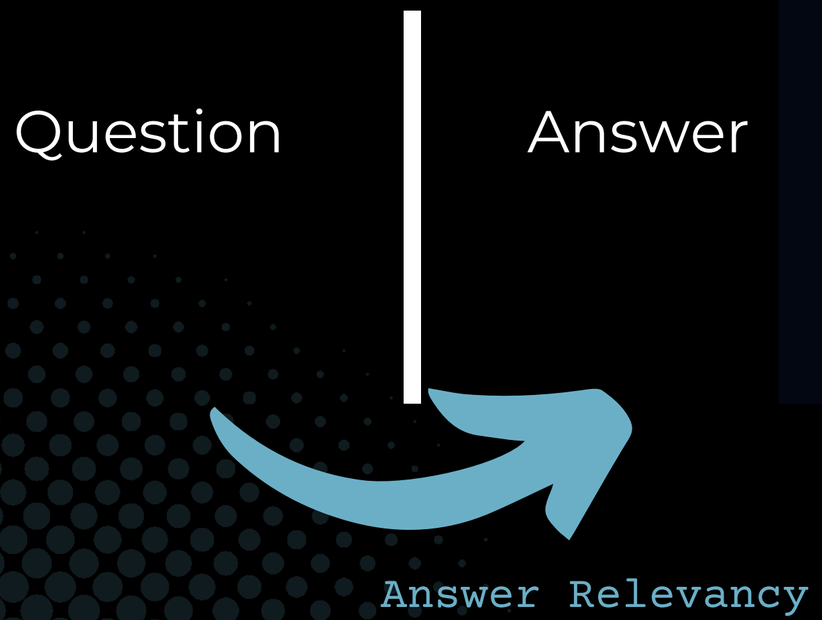
Context: Albert Einstein (born 14 March 1879) was a German-born theoretical physicist, widely held to be one of the greatest and most influential scientists of all time

High faithfulness answer: Einstein was born in Germany on 14th March 1879.

Low faithfulness answer: Einstein was born in Germany on 20th March 1879.

ANSWER RELEVANCY

- **Measures** how relevant is the generated answer to the prompt
- (0,1), Higher -> better



Hint

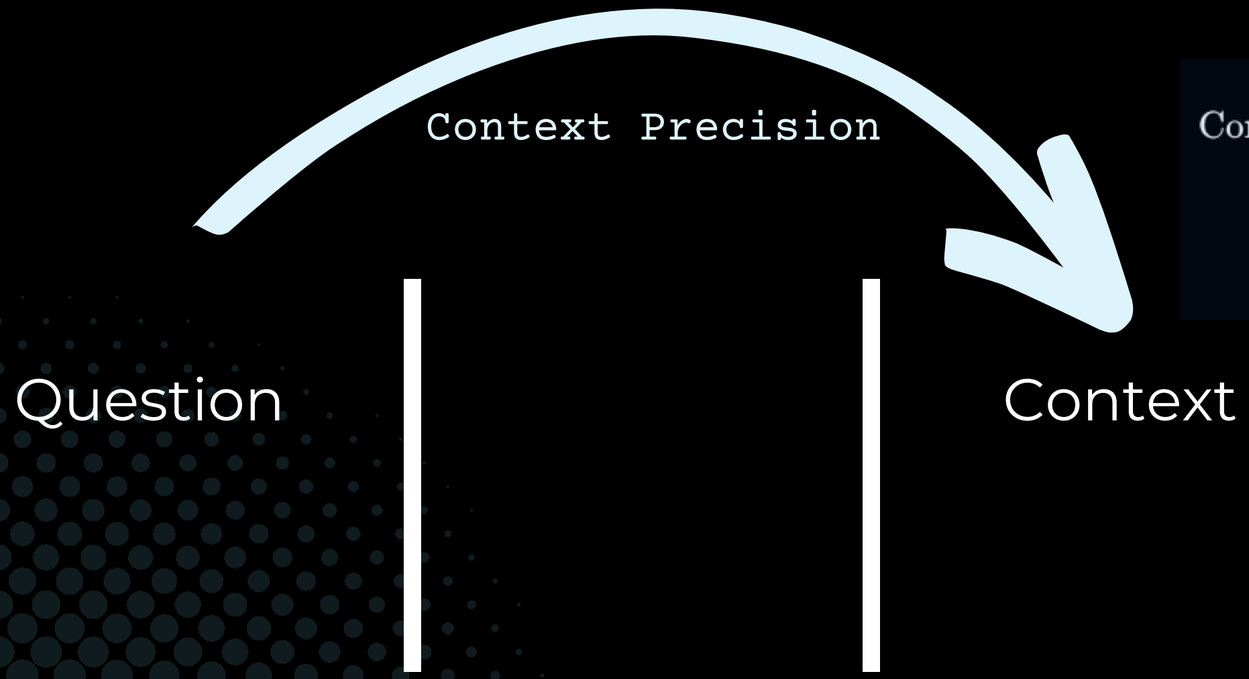
Question: Where is France and what is it's capital?

Low relevance answer: France is in western Europe.

High relevance answer: France is in western Europe and Paris is its capital.

CONTEXT PRECISION

- **Measures** relevancy of retrieved context to prompt
- (0,1), Higher -> better



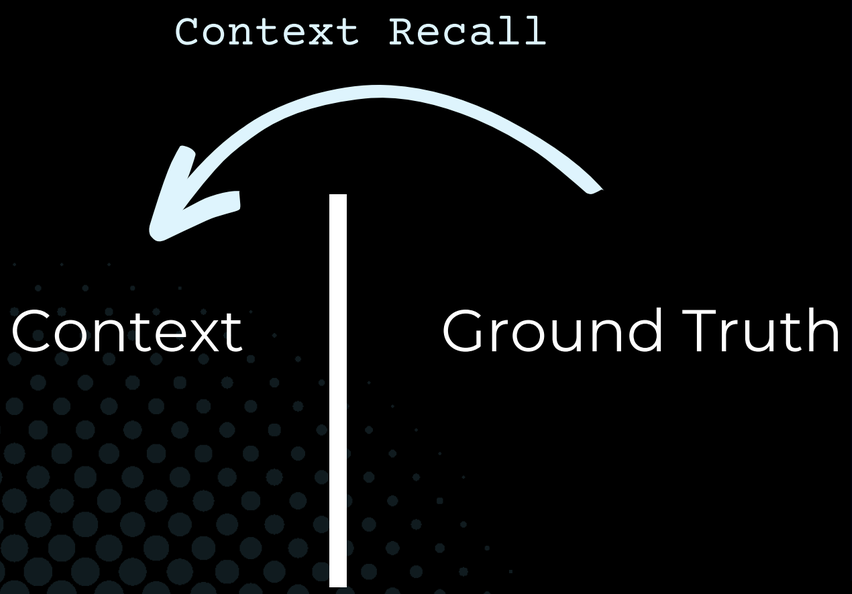
$$\text{Context Precision@k} = \frac{\sum \text{precision@k}}{\text{total number of relevant items in the top K results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k} + \text{false positives@k})}$$

Where k is the total number of chunks in `contexts`

CONTEXT RECALL

- **Measures** recall of the retrieved context
- (0,1), Higher -> better



$$\text{context recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of sentences in GT}|}$$

Hint

Question: Where is France and what is it's capital?

Ground truth: France is in Western Europe and its capital is Paris.

High context recall: France, in Western Europe, encompasses medieval cities, alpine villages and Mediterranean beaches. Paris, its capital, is famed for its fashion houses, classical art museums including the Louvre and monuments like the Eiffel Tower.

Low context recall: France, in Western Europe, encompasses medieval cities, alpine villages and Mediterranean beaches. The country is also renowned for its wines and sophisticated cuisine. Lascaux's ancient cave drawings, Lyon's Roman theater and the vast Palace of Versailles attest to its rich history.

RAG EVAL

Retrieval

Context Precision

Context Recall

Question | Answer | Context | Ground Truth

Faithfulness

Answer Relevancy

Generation

OVERVIEW

RETRIEVER

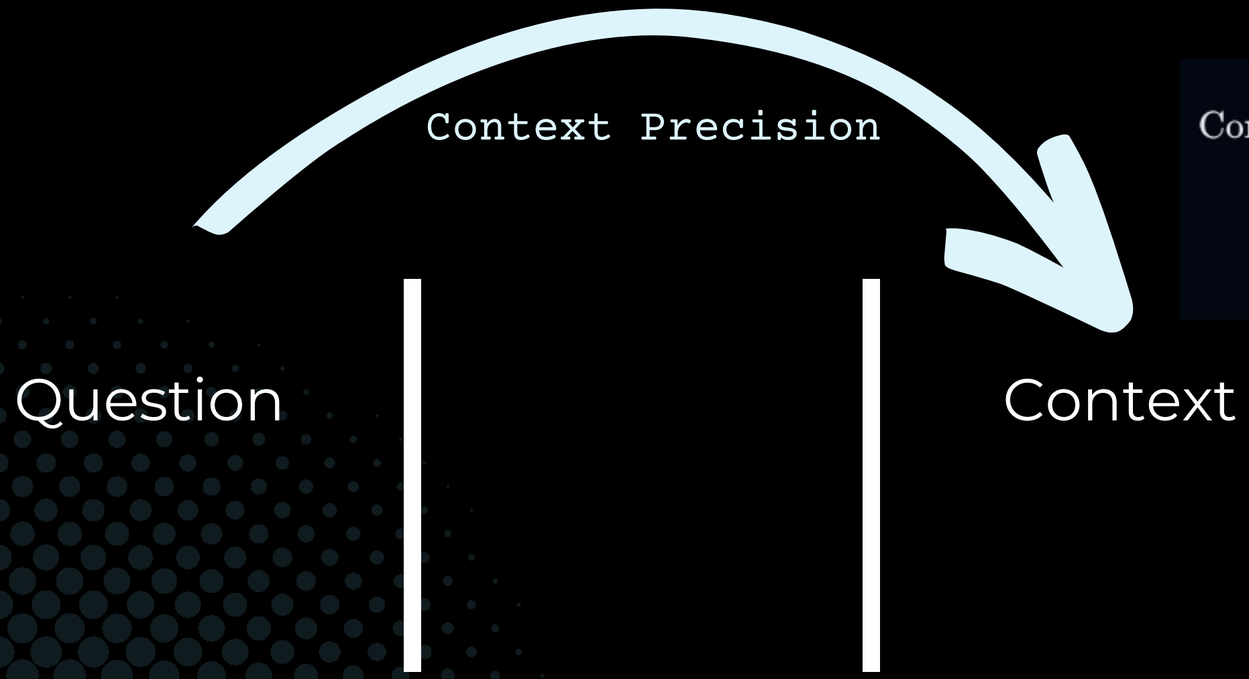
- **Context Precision:** Conveys **quality** of retrieval pipeline
- **Context Recall:** Measures ability to **retrieve** all **necessary** information

GENERATOR

- **Faithfulness**: Measures **hallucinations**
- **Answer Relevancy**: Measures how **“to the point“** answers are to the question

CONTEXT PRECISION

- **Measures** relevancy of retrieved context to prompt
- (0,1), Higher -> better



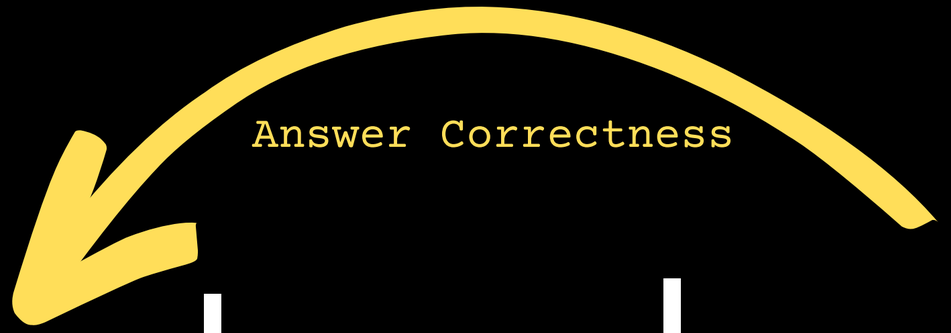
$$\text{Context Precision@k} = \frac{\sum \text{precision@k}}{\text{total number of relevant items in the top K results}}$$

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k} + \text{false positives@k})}$$

Where k is the total number of chunks in `contexts`

“End-to-End” RAG Metrics

ANSWER CORRECTNESS



Question

Answer

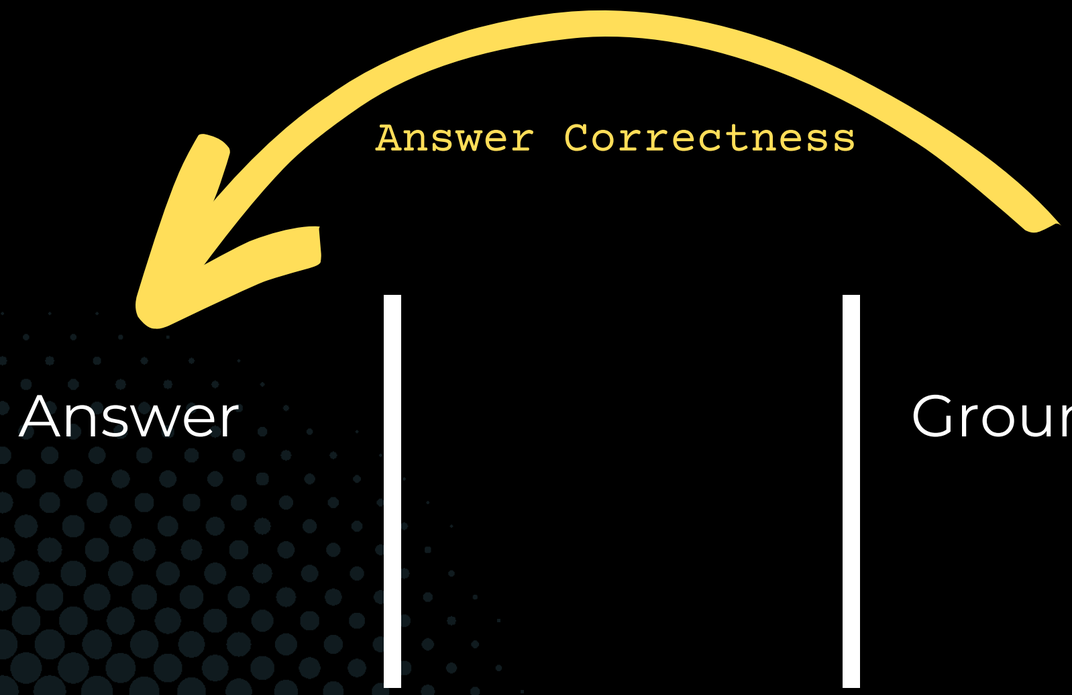
Context

Ground Truth

ANSWER CORRECTNESS

Two aspects: **Semantic** and **factual similarity**

- (0,1), Higher -> better



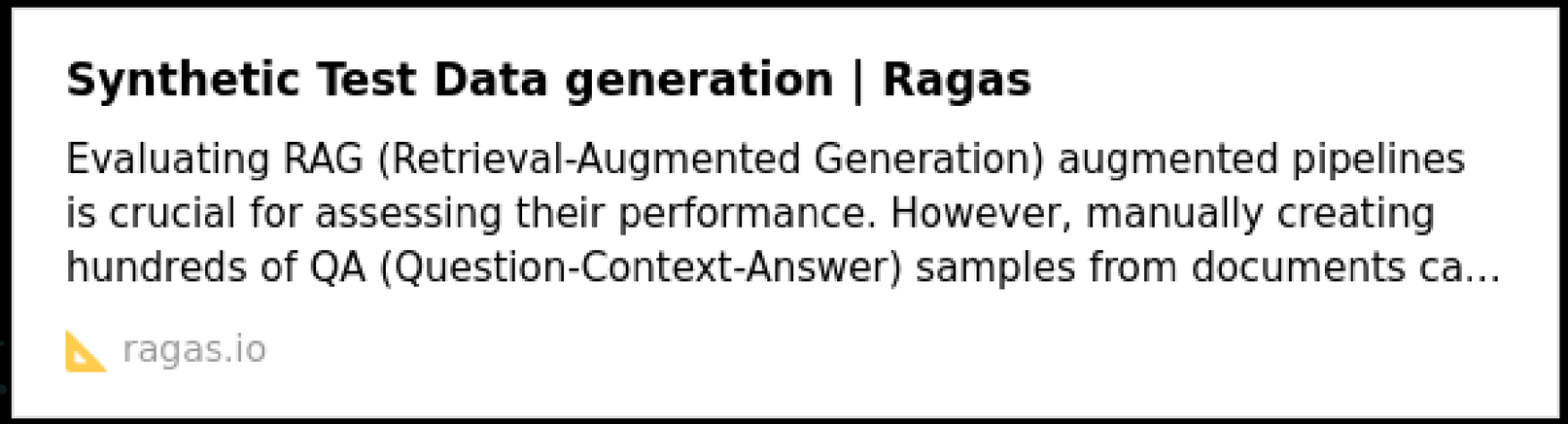
Ground truth: Einstein was born in 1879 at Germany .

High answer correctness: In 1879, in Germany, Einstein was born.

Low answer correctness: In Spain, Einstein was born in 1879.

STEPS TO DOING RAGAS

- Generate [Q, A, Context, GT] data

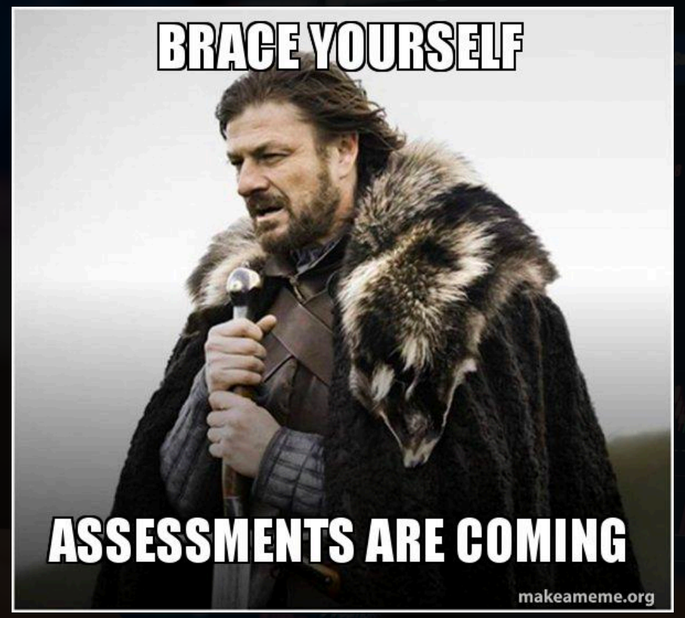


- Running evaluation!

Enhancing Retrieval

MULTI-QUERY RETRIEVER

The MultiQueryRetriever ***automates the process of prompt tuning*** *by using an LLM to* ***generate multiple queries from different perspectives*** *for a given user input query.*



RAGAS + RAG Improvement

Presented by
The Wiz

KEY RESULT

Generation

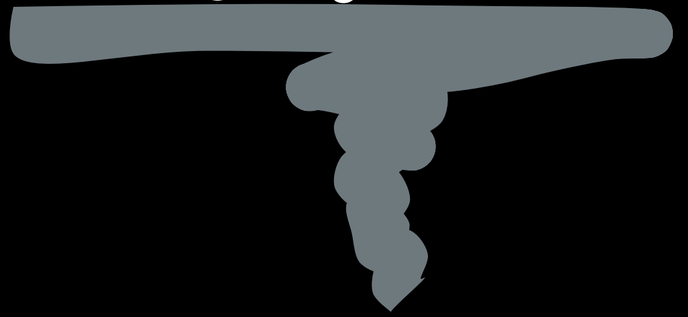
Faithfulness

87.5%

Base Model

88.9%

Multi-Query Retriever



CONCLUSION

- **LangChain v0.1.0** for **RAG**
  - **Stable** releases from here!
  - More **production-ready** than ever

- RAGAS is great for **directional improvements!**
  - **Retriever**: Context Precision, Context Recall
  - **Generation**: Answer Relevancy, Faithfulness

QUESTIONS?

Thank you!

ragas