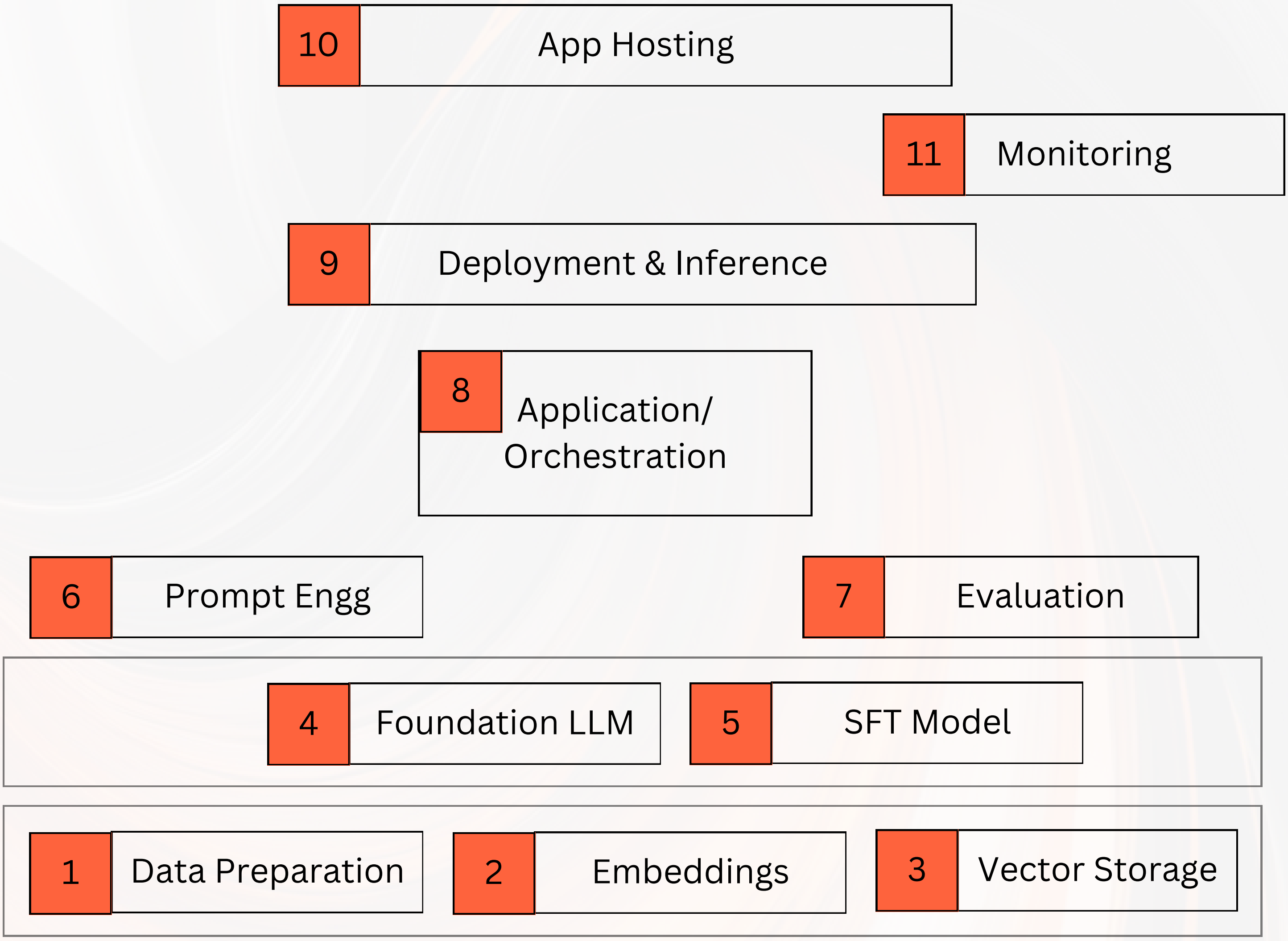
Evolving LLMOps Stack for RAG

The production ecosystem for RAG and LLM applications is still evolving. Early tooling and design patterns have emerged.



10 App Hosting

11 Monitoring

9 Deployment & Inference

8 Application/ Orchestration

6 Prompt Engg

7 Evaluation

4 Foundation LLM

5 SFT Model

1 Data Preparation

2 Embeddings

3 Vector Storage

Data Layer

The foundation of RAG applications is the data layer. This involves -

- Data preparation - Sourcing, Cleaning, Loading & Chunking
- Creation of Embeddings
- Storing the embeddings in a vector store

Data Preparation

Snorkel

LangChain

LlamaIndex

Embeddings

OpenAI

Hugging Face

Vertex.ai

Vector Storage

Faiss

Weaviate

Pinecone

Milvus

Chroma

Qdrant

Popular Data Layer Vendors (Non Exhaustive)

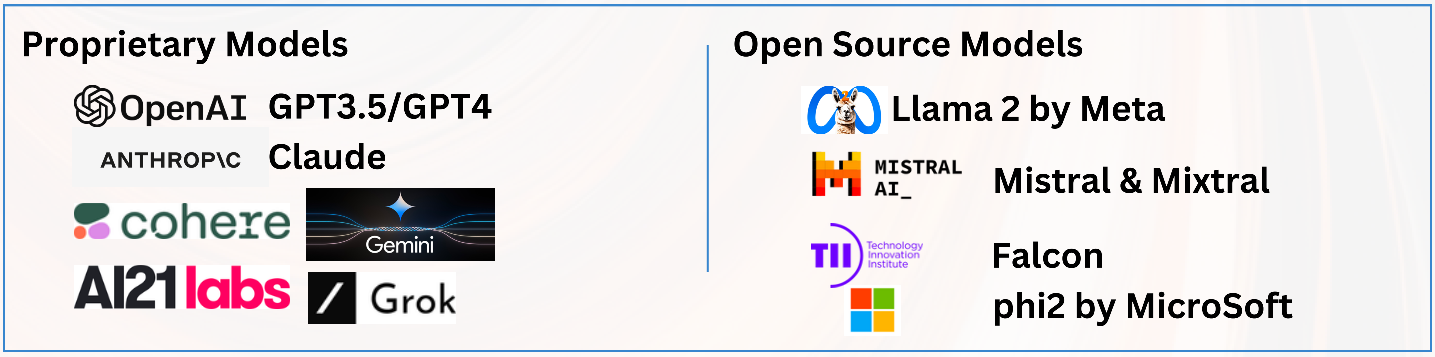
Abhinav Kimothi

Model Layer

2023 can be considered a year of LLM wars. Almost every other week in the second half of the year a new model was released. Like there is no RAG without data, there is no RAG without an LLM. There are four broad categories of LLMs that can be a part of a RAG application

1. **A Proprietary Foundation Model** - Developed and maintained by providers (like OpenAI, Anthropic, Google) and is generally available via an API
2. **Open Source Foundation Model** - Available in public domain (like Falcon, Llama, Mistral) and has to be hosted and maintained by you.
3. **A Supervised Fine-Tuned Proprietary Model** - Providers enable fine-tuning of their proprietary models with your data. The fine-tuned models are still hosted and maintained by the providers and are available via an API
4. **A Supervised Fine-Tuned Open Source Model** - All Open Source models can be fine-tuned by you on your data using full fine-tuning or PEFT methods.

There are a lot of vendors that have enabled access to open source models and also facilitate easy fine tuning of these models



Proprietary Models

GPT3.5/GPT4

Claude

Open Source Models

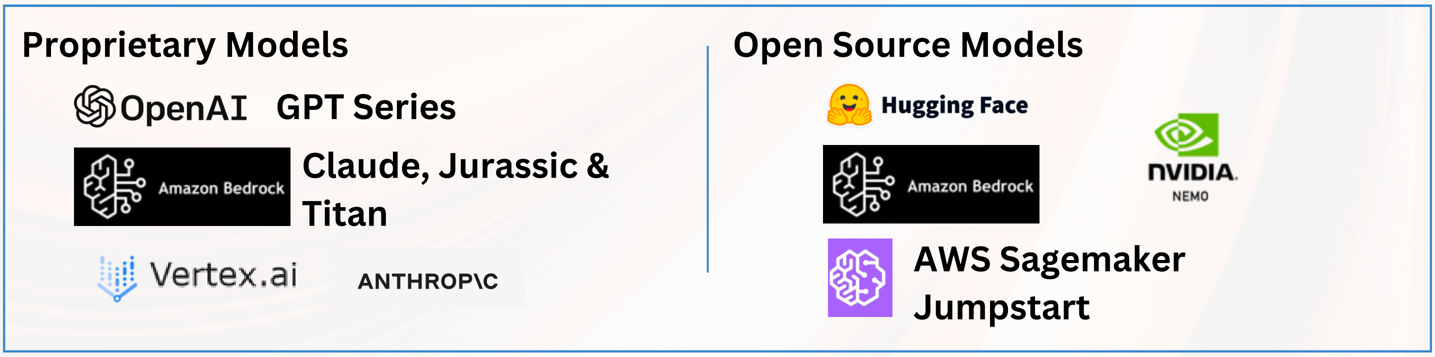
Llama 2 by Meta

Mistral & Mixtral

Falcon

phi2 by MicroSoft

Popular proprietary and open source LLMs (Non Exhaustive)



Proprietary Models

GPT Series

Claude, Jurassic & Titan

Open Source Models

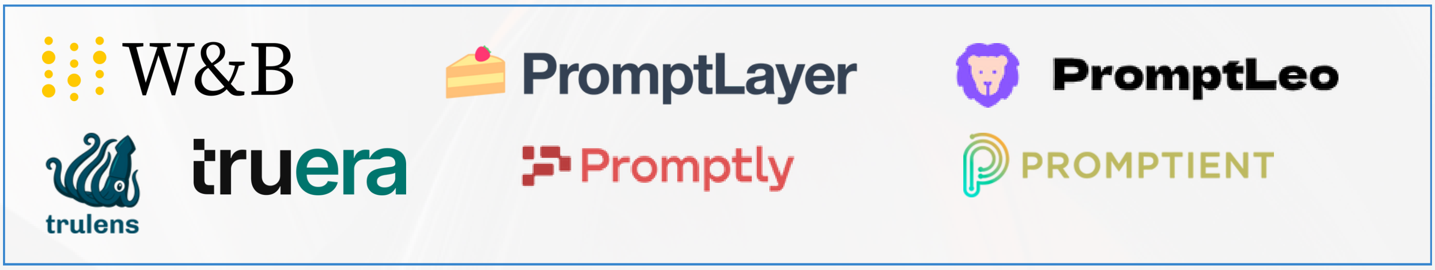
AWS Sagemaker Jumpstart

Popular vendors providing access to LLMs (Non Exhaustive)

Note : For Open Source models it is important to check the license type. Some open source models are not available for commercial use

Prompt Layer

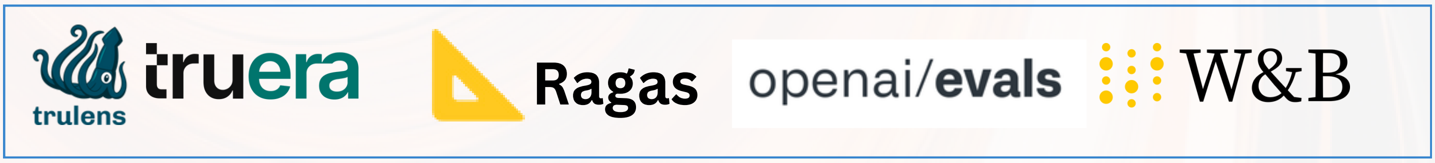
Prompt Engineering is more than writing questions in natural language. There are several prompting techniques and developers need to create prompts tailored to the use cases. This process often involves experimentation: the developer creates a prompt, observes the results and then iterates on the prompts to improve the effectiveness of the app. This requires tracking and collaboration



Popular prompt engineering platforms (Non Exhaustive)

Evaluation

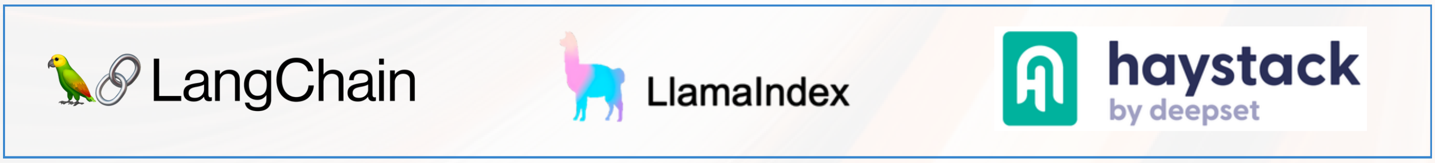
It is easy to build a RAG pipeline but to get it ready for production involves robust evaluation of the performance of the pipeline. For checking hallucinations, relevance and accuracy there are several frameworks and tools that have come up.



Popular RAG evaluation frameworks and tools (Non Exhaustive)

App Orchestration

An RAG application involves interaction of multiple tools and services. To run the RAG pipeline, a solid orchestration framework is required that invokes these different processes.

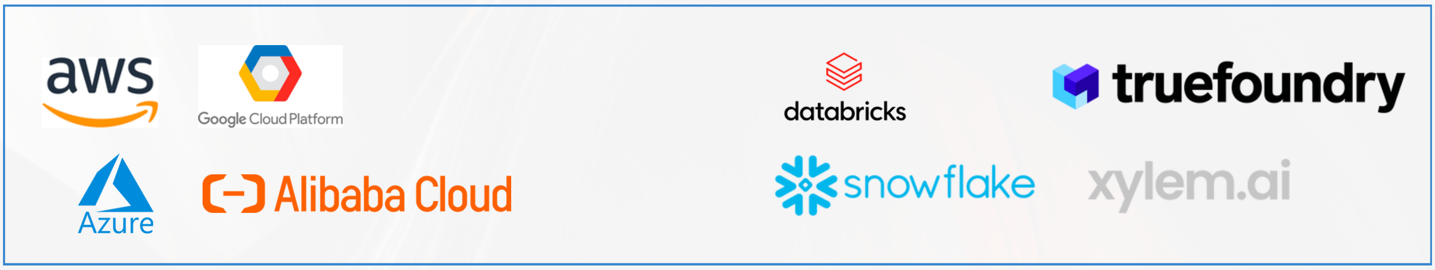


Popular App orchestration frameworks (Non Exhaustive)

Deployment Layer

Deployment of the RAG application can be done on any of the available cloud providers and platforms. Some important factors to consider while deployment are also -

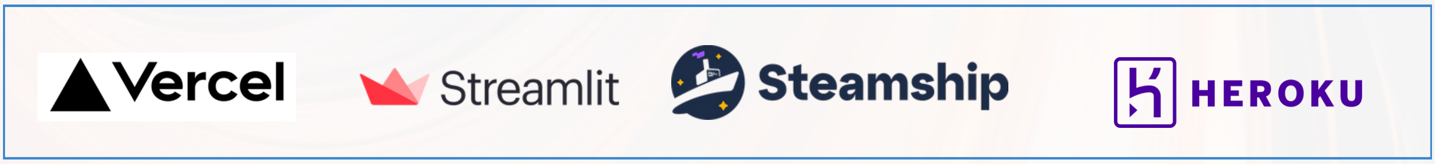
- Security and Governance
- Logging
- Inference costs and latency



Popular cloud providers and LLMOps platforms (Non Exhaustive)

Application Layer

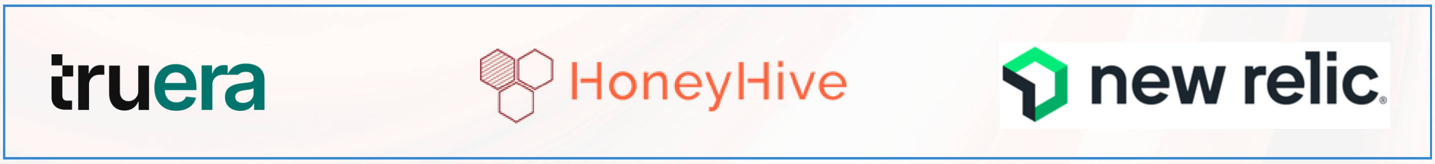
The application finally needs to be hosted for the intended users or systems to interact with it. You can create your own application layer or use the available platforms.



Popular app hosting platforms (Non Exhaustive)

Monitoring

Deployed application needs to be continuously monitored for both accuracy and relevance as well as cost and latency.



Popular monitoring platforms (Non Exhaustive)

Other Considerations

LLM Cache - To reduce costs by saving responses for popular queries
LLM Guardrails - To add additional layer of scrutiny on generations