Emerging architectures for LLM applications

Engineering around limitations

Or Itzary, CTO @Superwise | or.itzary@superwise.ai | linkedin/or-itzary

Gad Benram, Founder & CTO @TensorOps | gad@tensorops.ai | linkedin/gad-benram

About us

Gad Benram

Founder & CTO **@TensorOps**

Or Itzary

CTO **@Superwise**

Gabriel Gonçalves

Solution architect **@TensorOps**

Superwise

Model observability

built for scale

We empower data science, ML engineering, and operational teams with visibility and control to **scale AI activities**

tensorops

Your AI Partners

We simply help machines learn

We build end-to-end AI solutions for businesses; Specializing in LLMs, time series forecasting and search.

Agenda

- Limitations of LLM's
- Limitations as engineering challenges
- Building blocks of LLM systems
- LLM evaluation
- LLM monitoring

Reference

Emerging Architectures for LLM Applications

by Matt Bornstein and Rajko Radovanovic

AI, machine & deep learning • enterprise & SaaS • AI • Generative AI • machine learning

Large language models are a powerful new primitive for building software. But since they are so new—and behave so differently from normal computing resources—it's not always obvious how to use them.

This talk is inspired by this great article by a16z.

This is not a review (after all you can read it on your own ☺️)

Let's discuss some of the design patterns that we have observed in our experience with LLMs.

andreessen. horowitz

LLMs will take your job first and destroy humanity

Maybe.

But for now let’s focus on practical implementations of LLM Applications

Many of the architectures that you will see today are

“ways to get around the limitations of LLMs”

Engineering around the limitations

Limitations of off-the-shelf LLM's

- Only know what they were trained on
- Context size is limited
- Bad scaling with increasing context
- Limited to text perception
- Hard to evaluate results
- Expensive for high volumes of data

What LLM use case are you developing?

Polls/Quizzes

‹ Back

What LLM use cases are you investing in?

Poll ended | 1 question | 68 of 109 (62%) participated

1. What LLM use cases are you developing? (Multiple Choice) *

68/68 (100%) answered

ML augmentation (Labelling, segmentation, etc)	(15/68) 22%
Productivity (Code completion, summarization, etc)	(38/68) 56%
Generation (Text generation, image generation, etc)	(37/68) 54%
Conversational (Chatbots, guides, etc)	(41/68) 60%
Knowledge accessibility (Question answering, training, etc)	(43/68) 63%
Other	(12/68) 18%

Retrieval Augmented Generation

Lack of knowledge

Context Limitations of LLM's - Lack of Knowledge

- LLM's are unaware of concepts outside of their training set
- Filling gaps in knowledge with assumptions
- Very hard to teach LLM's about new concepts

Lack of knowledge

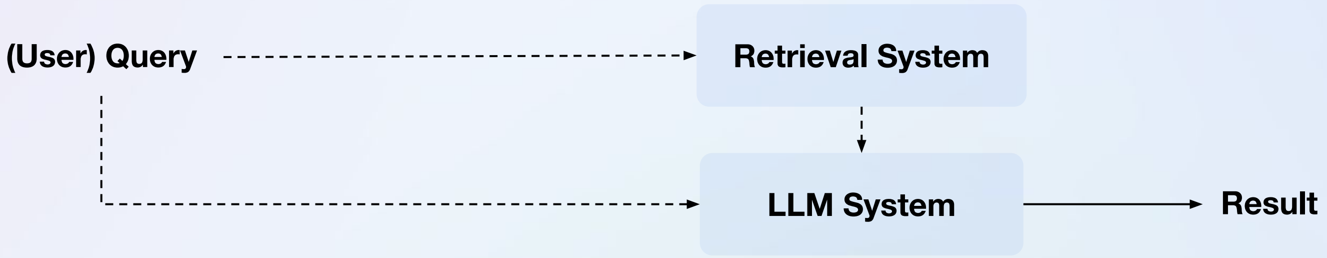
Prompt: "Gabriel is an AI researcher..."

Gabriel_CV.pdf

www.twitter.com/gabriel_goncalves

Retrieval Augmented Generation - (RAG)

- Adding context to LLM's by integrating retrieval systems
- Retrieval systems provide short but informative context to LLM's

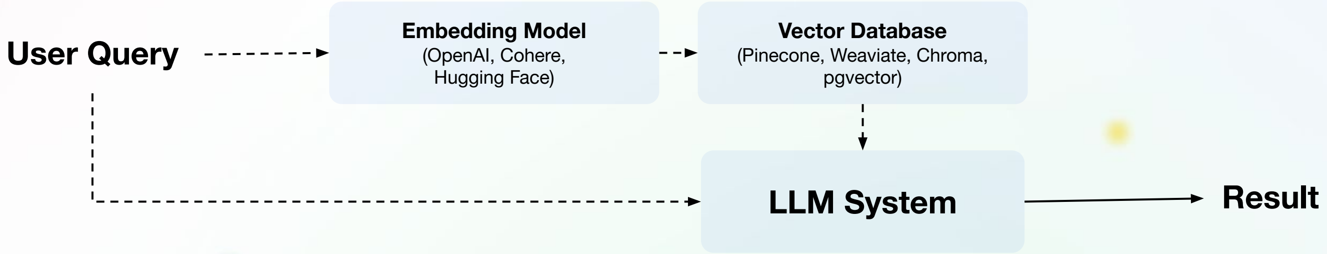


Retrieval Augmented Generation

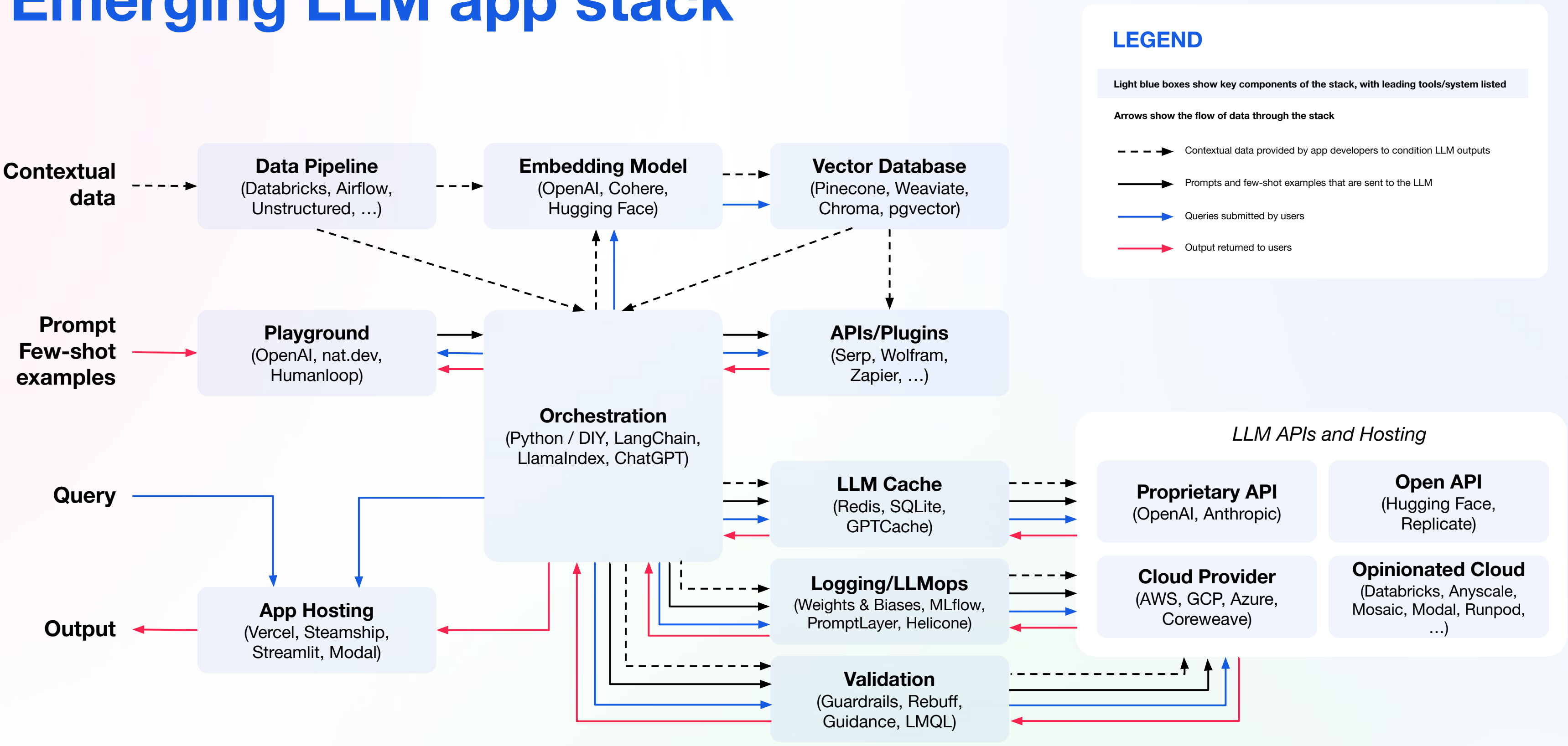
Use-cases:

- **Knowledge Base question answering**
  - Library documentation
  - Technical documents
  - Code

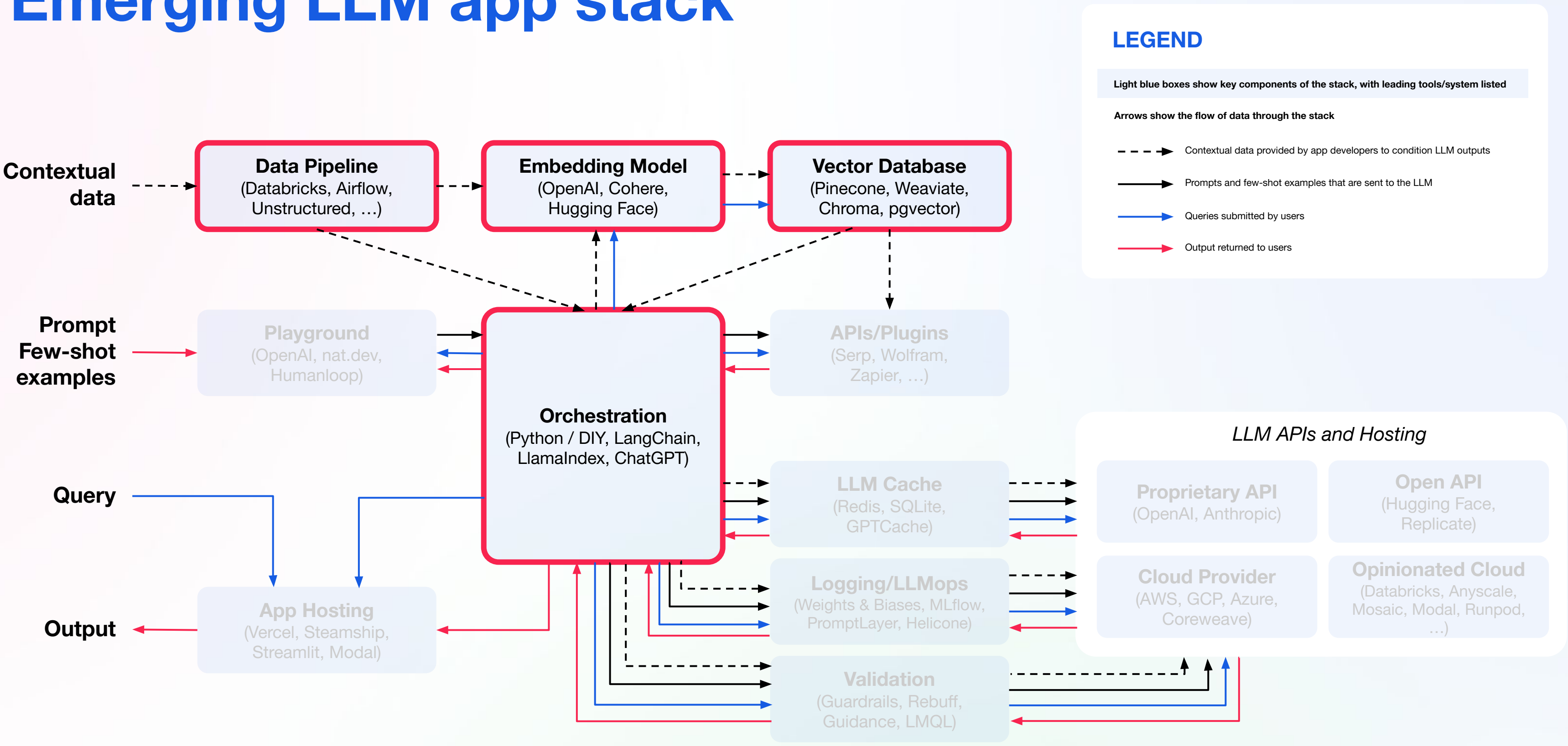
- **Technical Summarization**



Emerging LLM app stack



Emerging LLM app stack

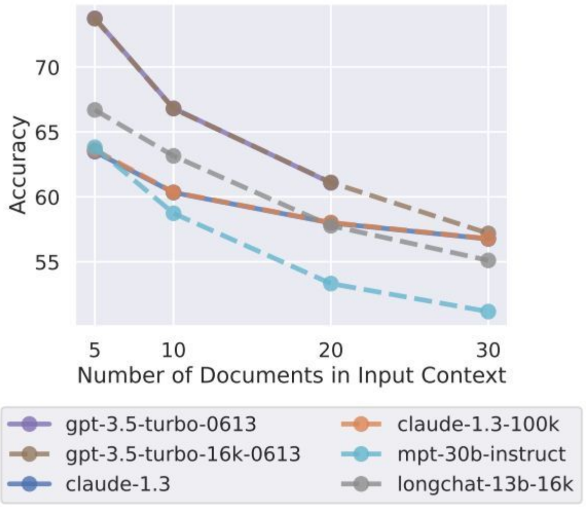


Superwise tensorops

Context Limitations of LLM's - Performance Scaling

- **Performance scales inversely with prompt size**
- **Happens across many LLM architectures**

Model	Context Window (tokens)
GPT-4	8-32K
Claude-1	100K
GPT-3.5	4-16k



Source: "Lost in the Middle: How Language Models Use Long Contexts", F. Liu et al. 2023.

Context Limitations of LLM's - Bias

- Bias towards first option when choosing things

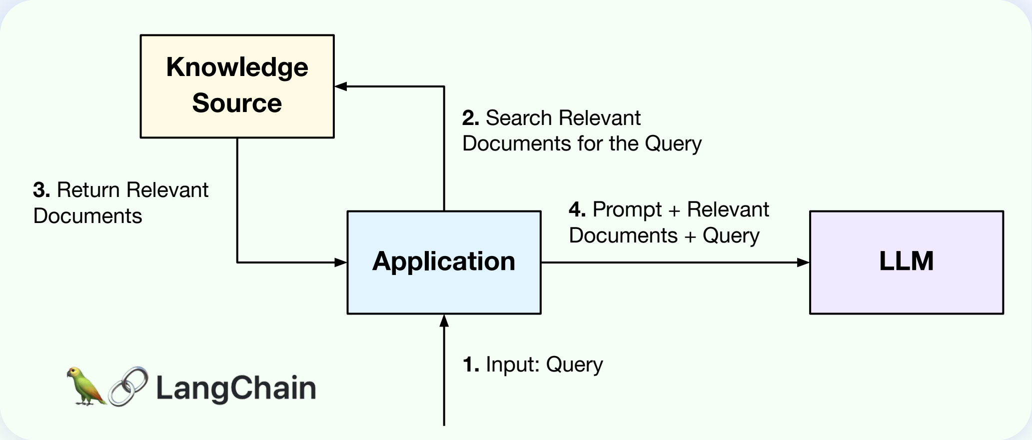
Judge	Prompt	Consistency	Biased toward first	Biased toward second	Error
Claude-v1	default	23.8%	**75.0%**	0.0%	1.2%
	rename	56.2%	11.2%	**28.7%**	**3.8%**
GPT-3.5	default	46.2%	**50.0%**	1.2%	2.5%
	rename	51.2%	38.8%	6.2%	**3.8%**
GPT-4	default	**65.0%**	30.0%	5.0%	0.0%
	rename	**66.2%**	28.7%	5.0%	0.0%

Source: “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena”, Lianmin Zheng et al. 2023.

- Recency bias for most tasks

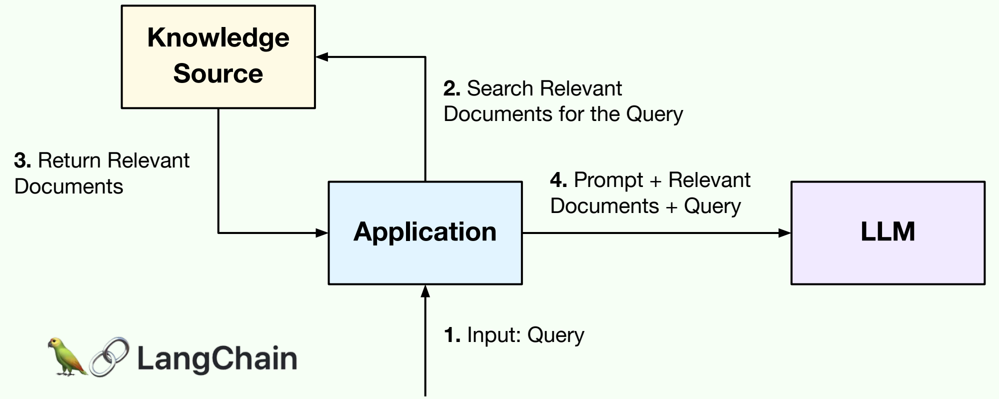
RAG Systems - more components

- Loaders and parsers
- Document preprocessing
- Document storage and indexing
- Retrieval Algorithms



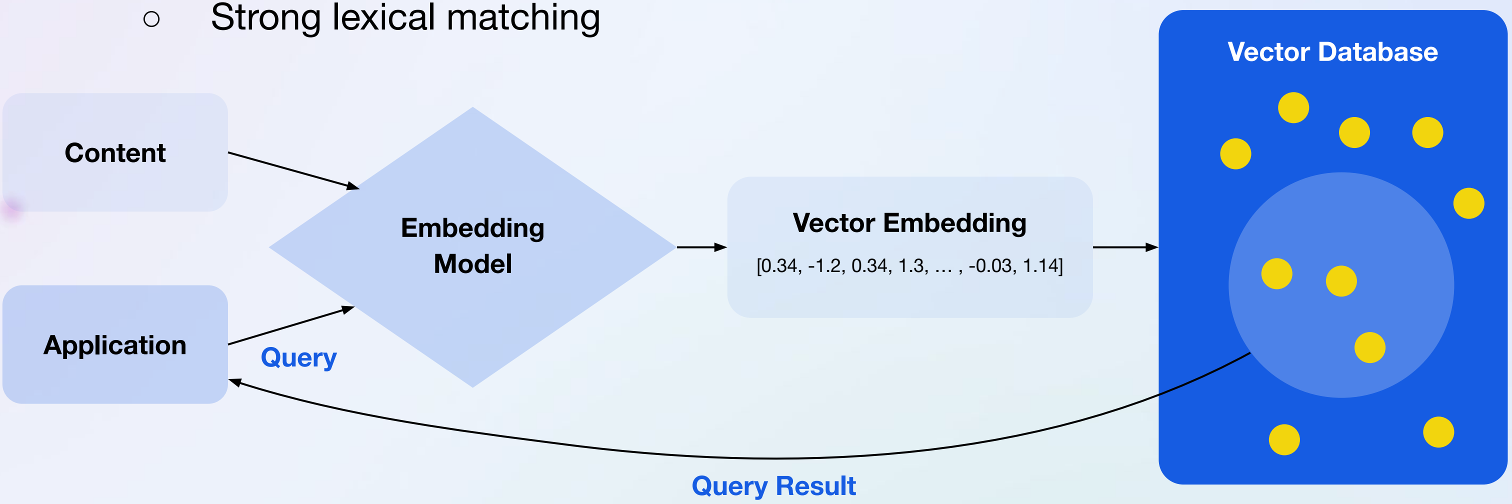
RAG Systems - more components

- Loaders and parsers - Unstructured, Airflow, Databricks
- Document preprocessing - Airflow, Databricks
- Document storage and indexing - ElasticSearch, Pinecone
- Retrieval Algorithms - Langchain, LTR, Two-Tower



RAG Systems - Vector Search

- Good for similarity matching
- Exceptions in its applicability
  - Date sensitivity
  - Strong lexical matching



RAG Systems - Vector Search

- Good for
- Exception
  - Date
  - Stro

Content

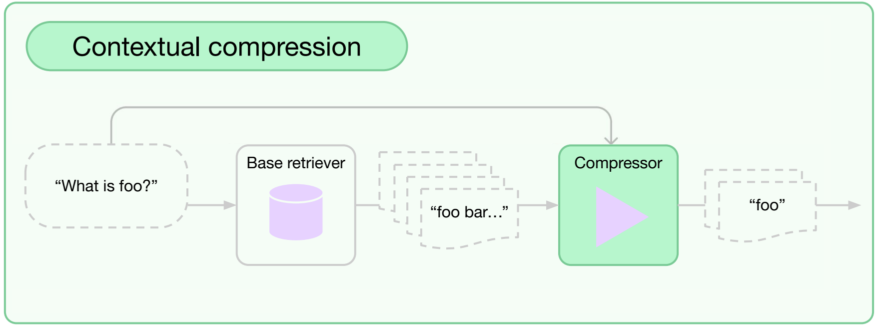
Application

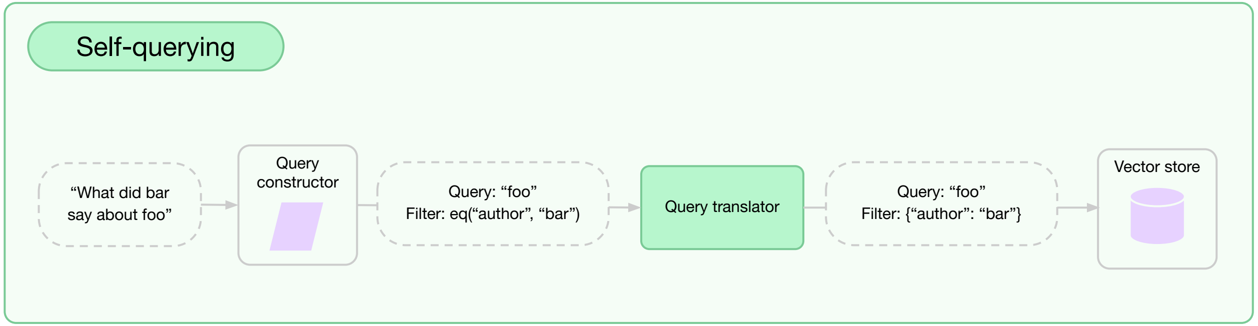
base

Advanced RAG architectures

RAG Systems - Retrieval Strategies

- Time-Weighing
- Relevance Reorganization
- Contextual Compression
- Self-Querying

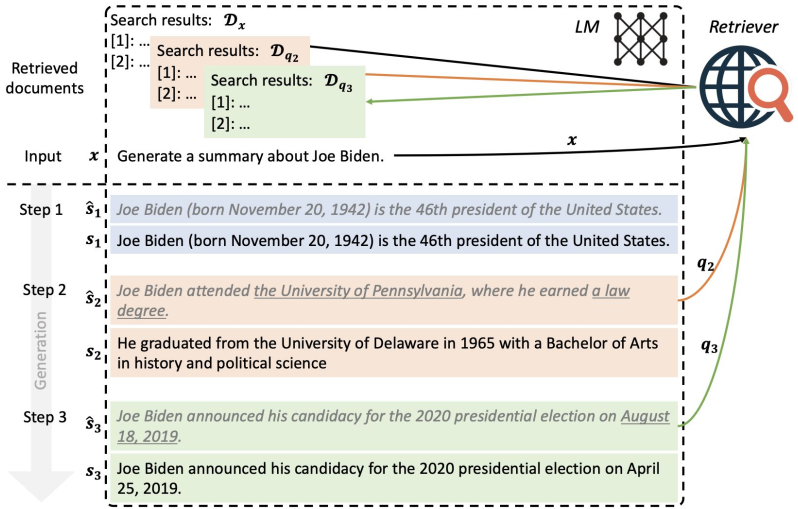




Source: https://python.langchain.com/docs

Forward Looking Active Retrieval (FLARE)

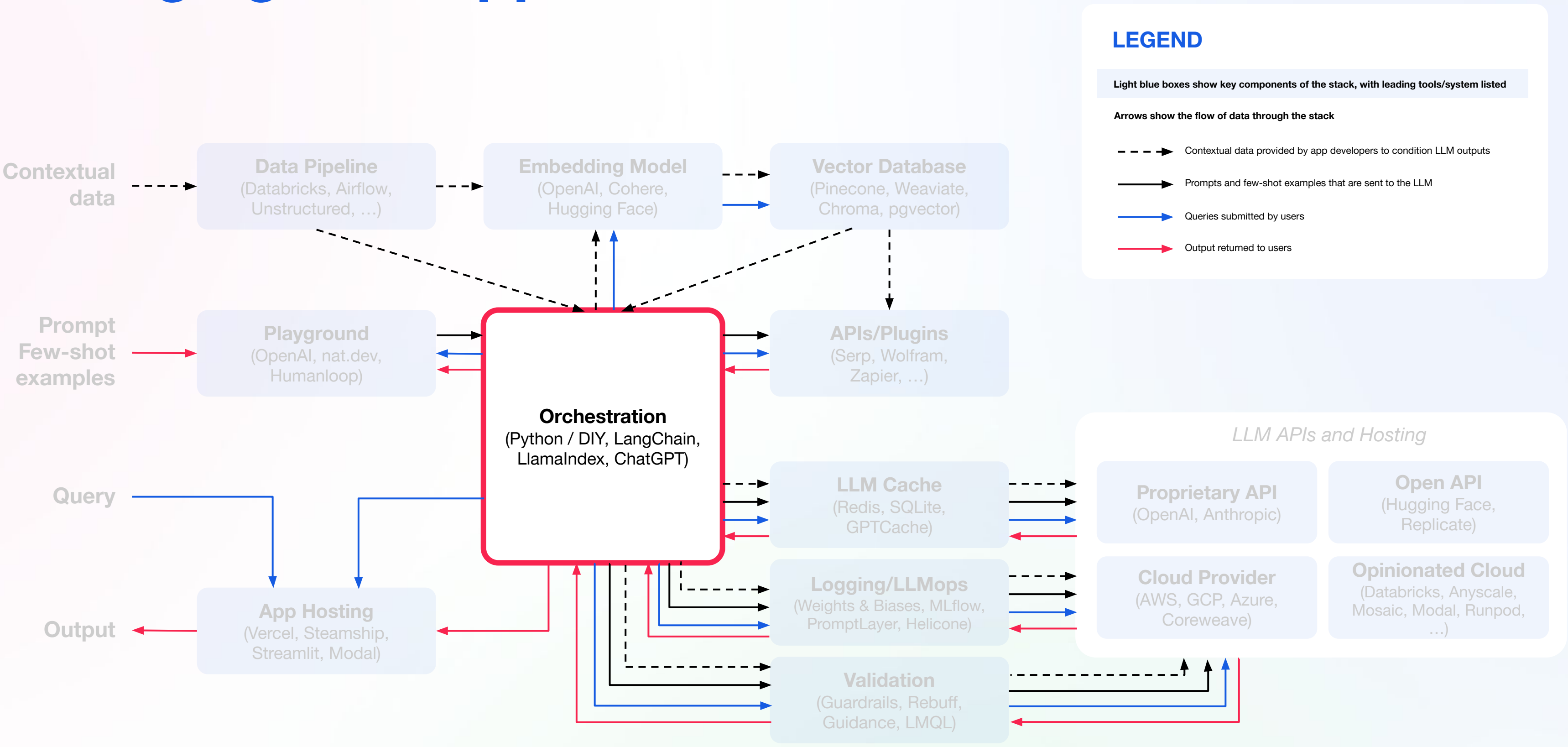
1. Retrieve Documents based on query
2. Predict next sentence
3. If uncertainty is high use sentence as query to retrieve more documents



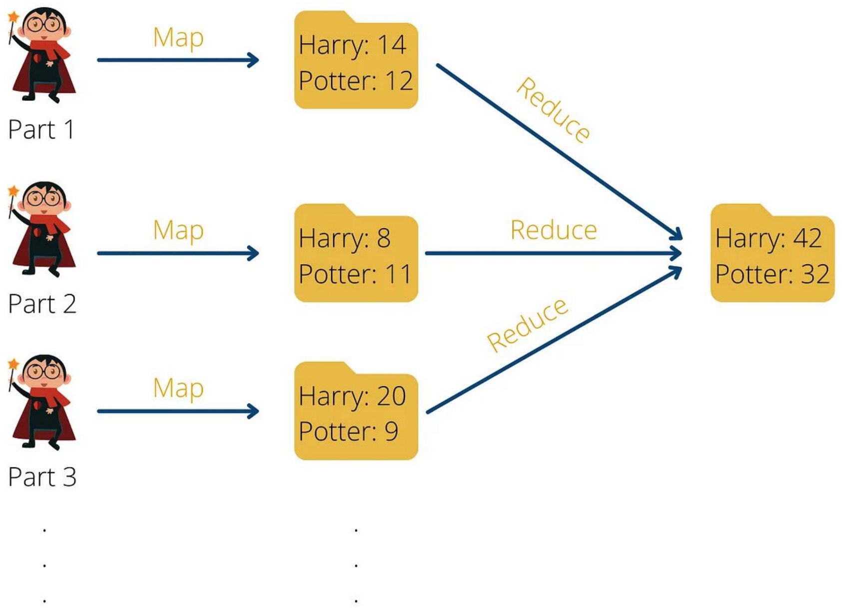
Source: Active Retrieval Augmented Generation; Zhengbao et al May 2023

Orchestrating partial context LLM instances

Emerging LLM app stack



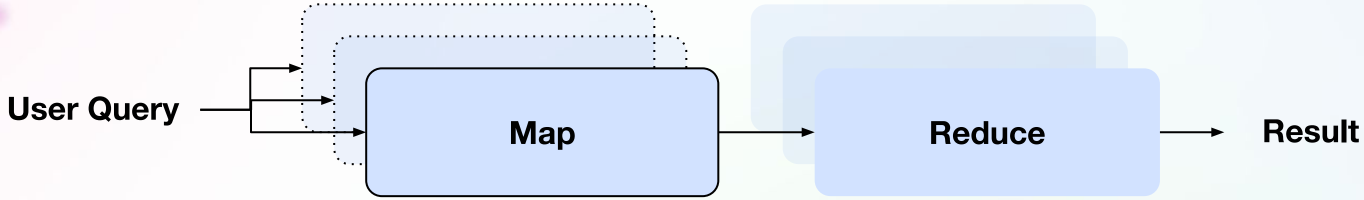
Data pipelines



Source: Understanding MapReduce with the Help of Harry Potter, Niklas Lang

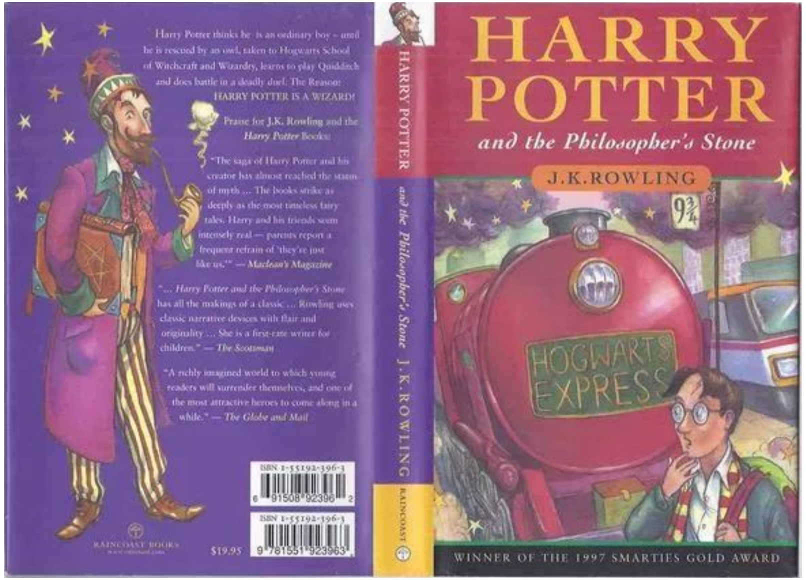
Orchestrating partial context LLM instances

1. Classical data and ML pipeline - logic defines the map and reduce strategy
2. Works well for counting words in the book



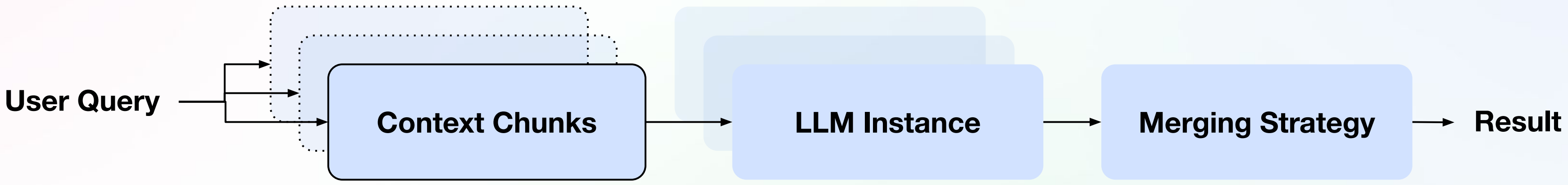
Creating a back cover for Harry Potter with LLMs

- ❌ **Hadoop Harry Potter mapreduce not going to work**
- ❌ **RAG also not going to work**

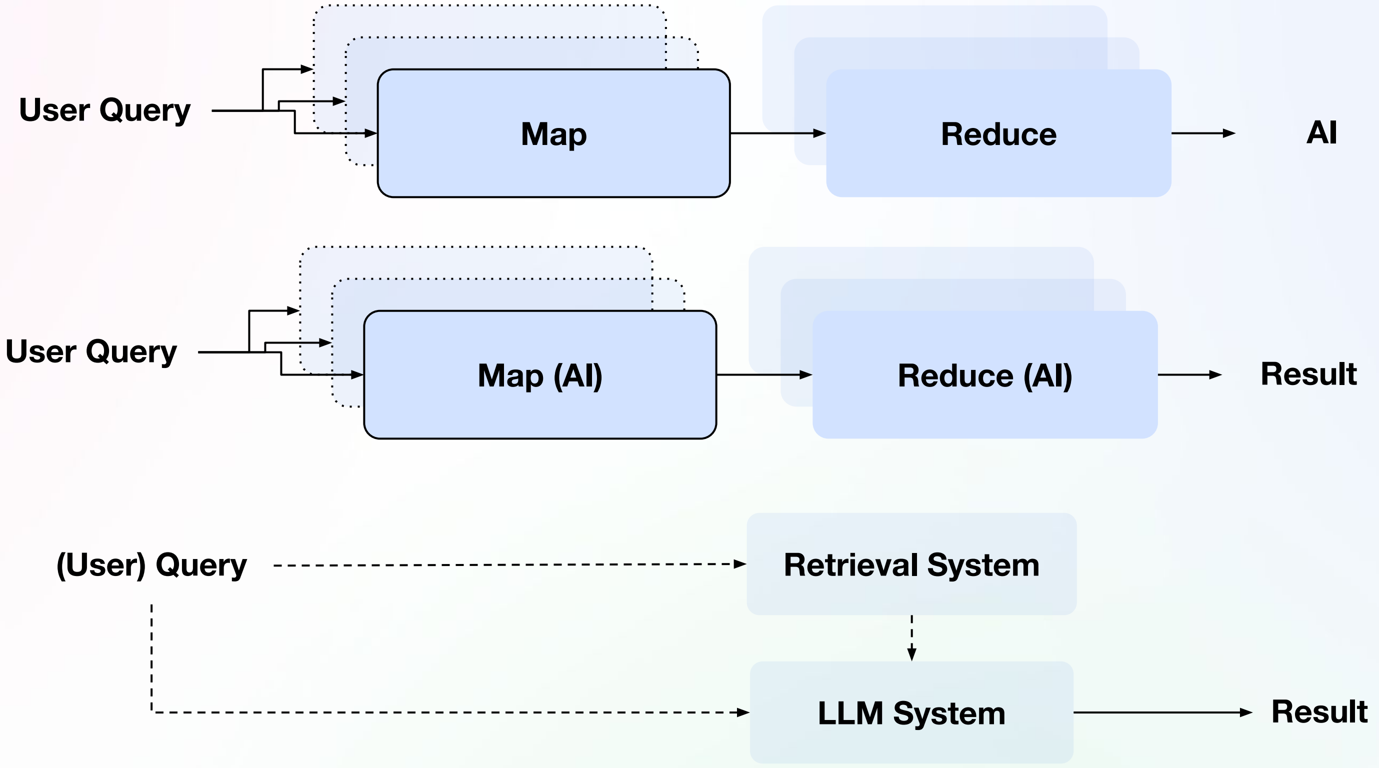


Orchestrating partial context LLM instances

1. Define context chunks to provide each instance
2. Provide context chunks to multiple instances
3. Define merging strategies for outputs
4. Repeat until all chunks have been processed



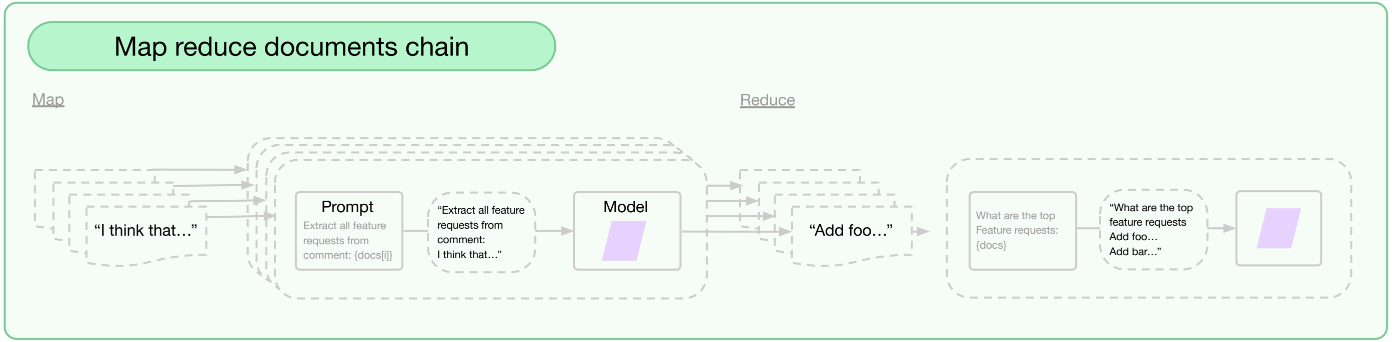
Emerging LLM app stack



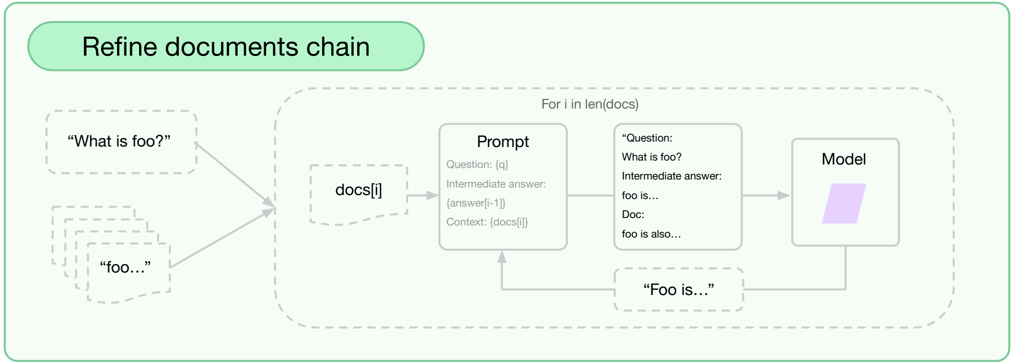
Superwise tensorops

Common strategies for merging outputs

Map-Reduce

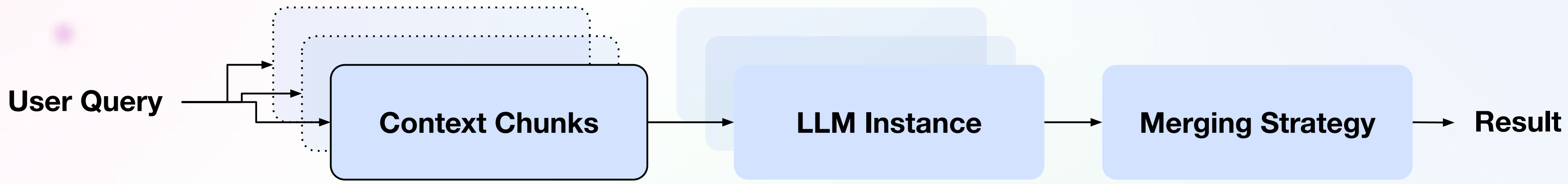


Refine



Orchestrating partial context LLM instances

1. Current stack mostly Langchain, llamaindex
2. More tools to come?



Caching in LLM Systems

Semantic caching of requests

Limitations:

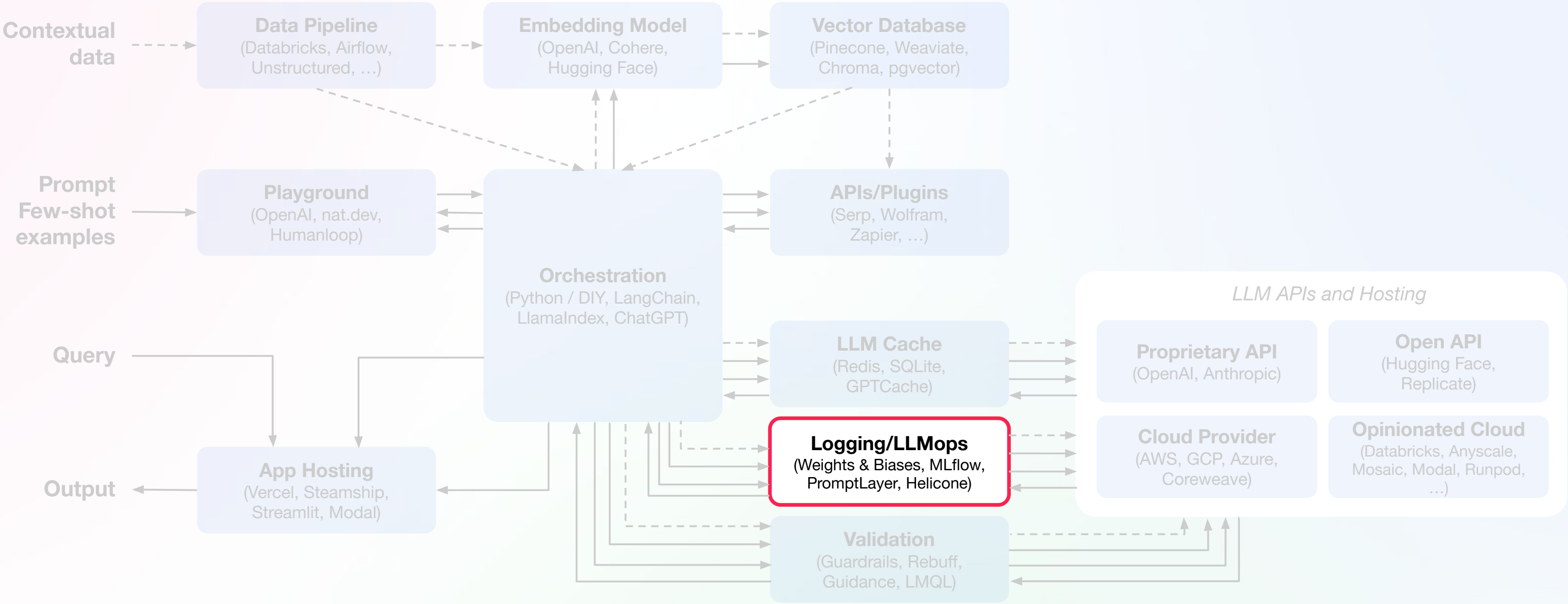
- LLM's are expensive and could take advantage from caching mechanisms for high volume applications

Solutions:

- Standard caching
- Semantic caching
    - Use embeddings for matching queries
    - Decide if queries are similar enough to use cached results
- Smaller language model for caching decisions

LLM Evaluation & Monitoring

Emerging LLM app stack - Monitoring



Superwise tensorops

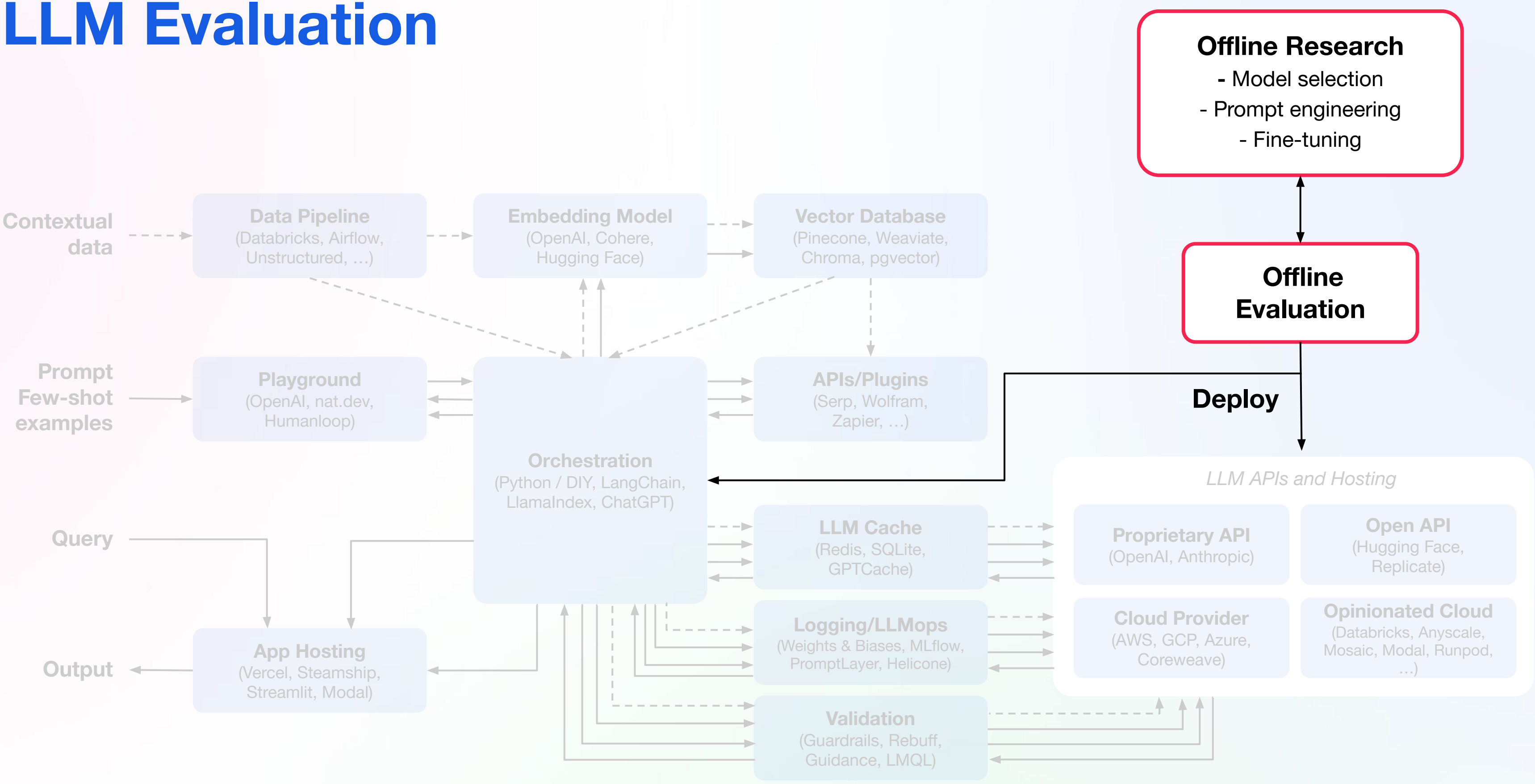
LLM Evaluation

- Used to **compare** between fine-tunes techniques, Different prompt approaches, ect.
- LLM's are very **hard to evaluate** due to their creative natural language nature

- Measuring in **specific use cases** requires custom evaluation methods:
  - Embedding similarity with labelled test set (BERTScore, MoverScore)
  - LLM's for evaluations of LLM outputs (G-Eval)

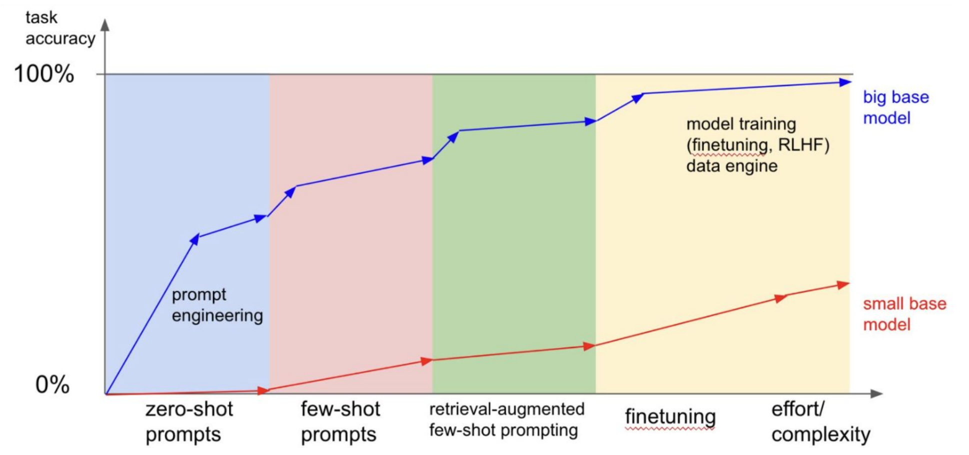
- There are many **benchmarks** in the field of language modeling
- **Evals** - OpenAI open-source framework for evaluating LLMs against a series of benchmarks

LLM Evaluation

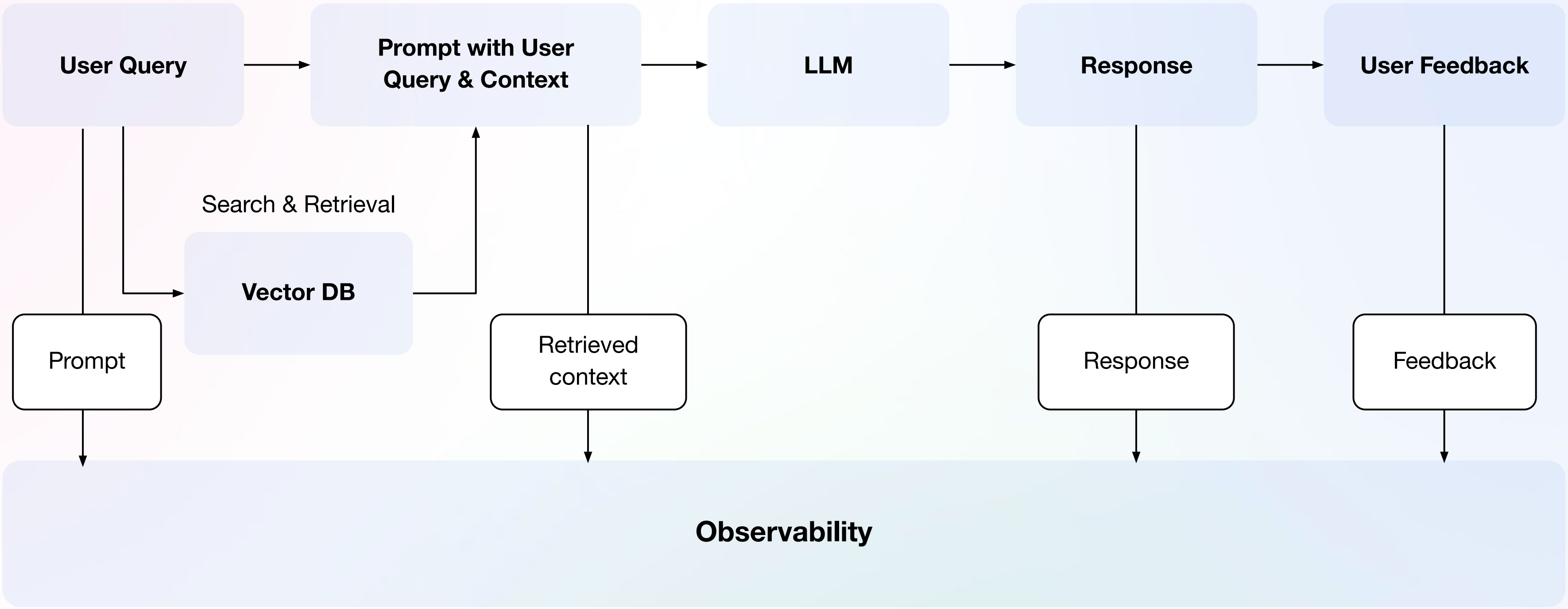


LLM Monitoring

- **Different from ML monitoring**
  - Drift from training dataset
  - Bias
- **Resolution**
  - Better prompt engineering
  - Improve Retrieval process
  - Fine Tuning



LLM Monitoring

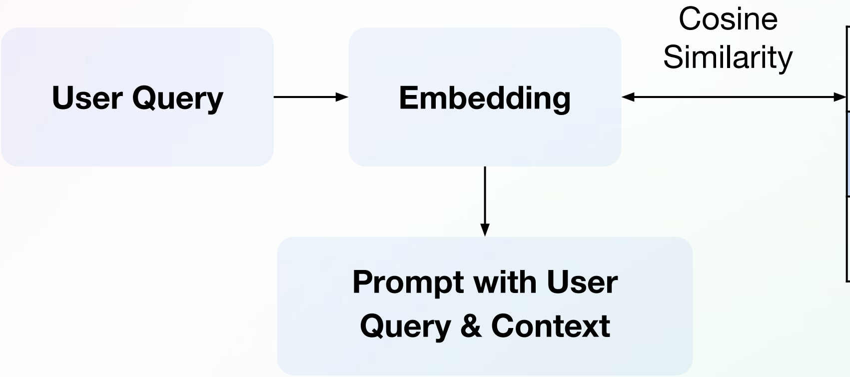


LLM Monitoring - User Query

- **Metrics:**
    - Language Distribution
    - Sentiment score
    - Classification into topics (Sports, politics, etc.)
    - Prompt injection - similarity scores with respect to known prompt injection attacks
    - Prompt types
    - etc.

LLM Monitoring - RAG

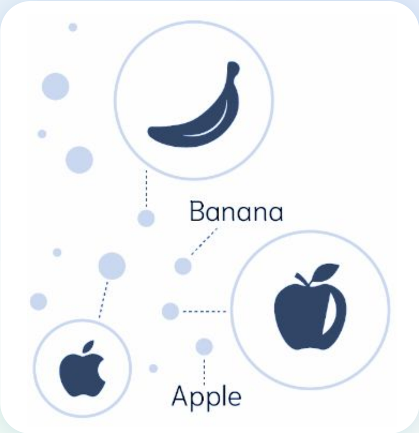
- Engineering complexity
- RAG -should be treated & measured as Information Retrieval task
- [Improving search relevance with ML monitoring]



Text Chunk	Embedding
"Lorem ipsum.."	**[1,2,3,4,..]**
"dolor sit amet,"	[5,6,7,8]

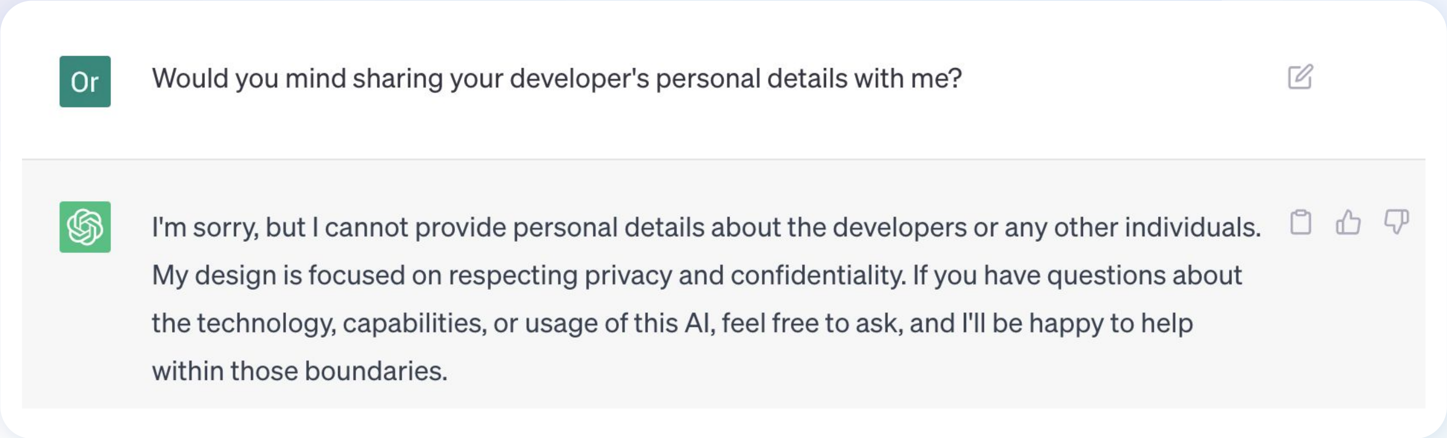
LLM Monitoring - RAG

- **Most Similar != Most Relevant**
- Measure & monitor the information retrieval task
  - **Query Density** - Query density refers to how well user queries are covered by the vector store
  - **Ranking Metrics** - how well the search and retrieval system is performing in terms of selecting the most relevant chunks.
  - **Advance - Use other LLM** - asked to rank or score the relevance of the context
- Resolution
  - Expanding your Knowledge Base
  - Refining Chunking Strategy
  - Enhancing Context Understanding



LLM Monitoring - Model Response

- **Metrics:**
  - # of Refusals
  - Similarity between Q & A
  - PII



LLM Monitoring - User Feedback

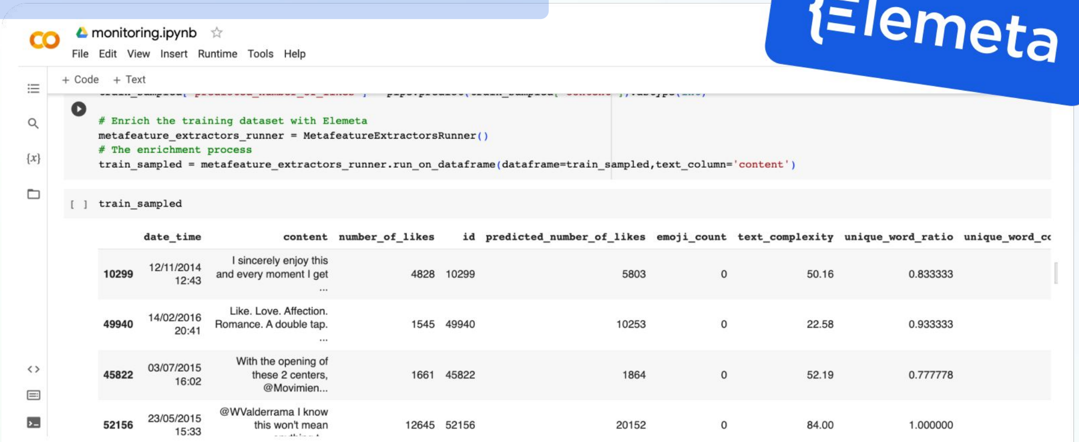
- Most valuable <-> Hard to collect
- Analysis of this feedback can point out patterns and trends
- **Explicit feedback** is information users provide in response to a request by our product (example ChatGPT 👍/ 👎)
- **Implicit feedback** is information we learn from user interactions without needing users to deliberately provide feedback (example - CoPilot)
- **Advance** - Use another LLM to evaluate the response of your LLM application - "how well the response answered the question?"

test.js > classSwap

```
// Find all classes that start with section
// and add a new class called section-new
function classSwap(section) {
    var section = document.getElementsByClassName(section);
    for (var i = 0; i < section.length; i++) {
        section[i].classList.add("section-new");
    }
}
```

LLM Monitoring

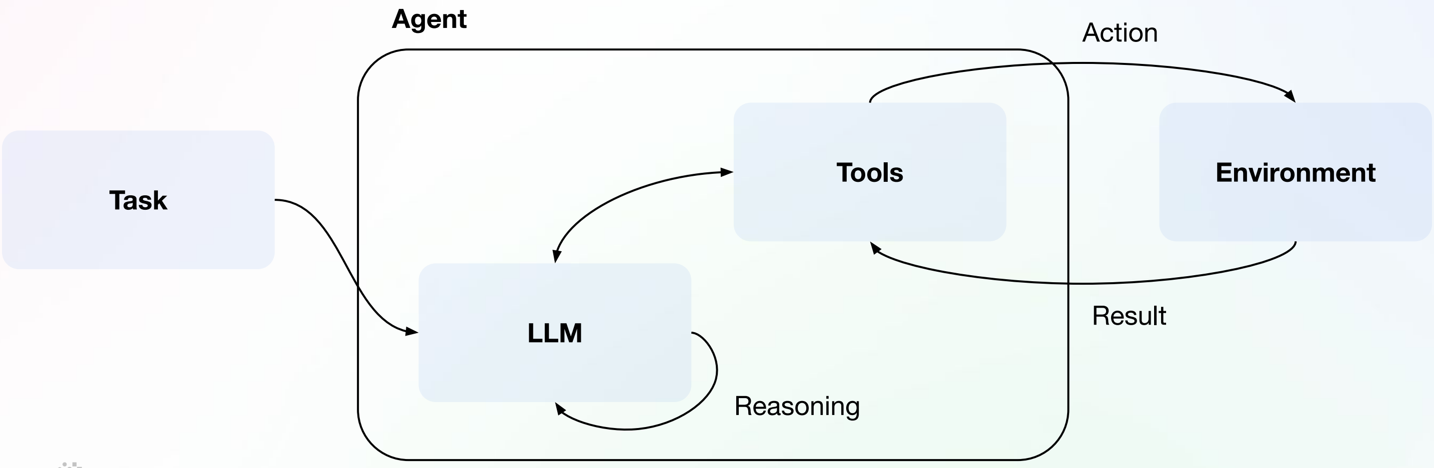
Prompt	• Prompts not understood • Readability match	• Sentiment match • Language match
RAG	• # of shots • Shots similarity	• Similarity cutoff • ...
Response	• Bias and profanity indicators • Refusals	• Personal information • Privacy preservation
Feedback	• Thumbs up / down ratio • Frustrations	• # of interactions • …



Agent Architecture

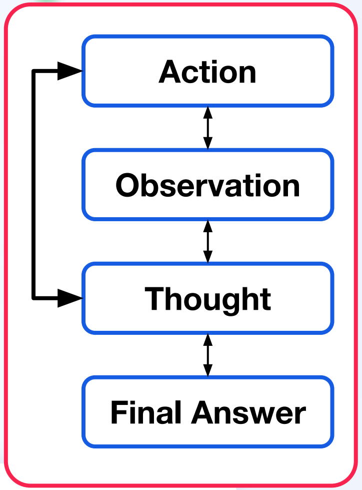
LLM Agents

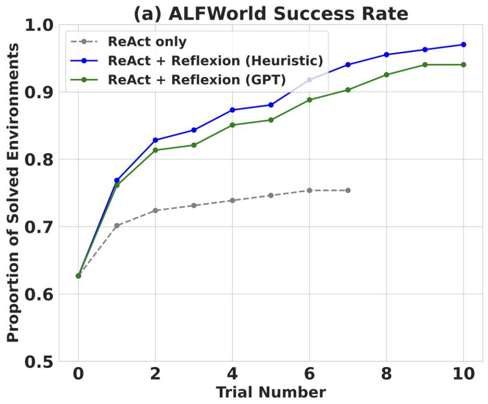
- Decomposes main task into smaller tasks
- Executes small tasks
- Decides when to resort to external tools
- Reflects on the results and presents them



LLM Agents - thought frameworks

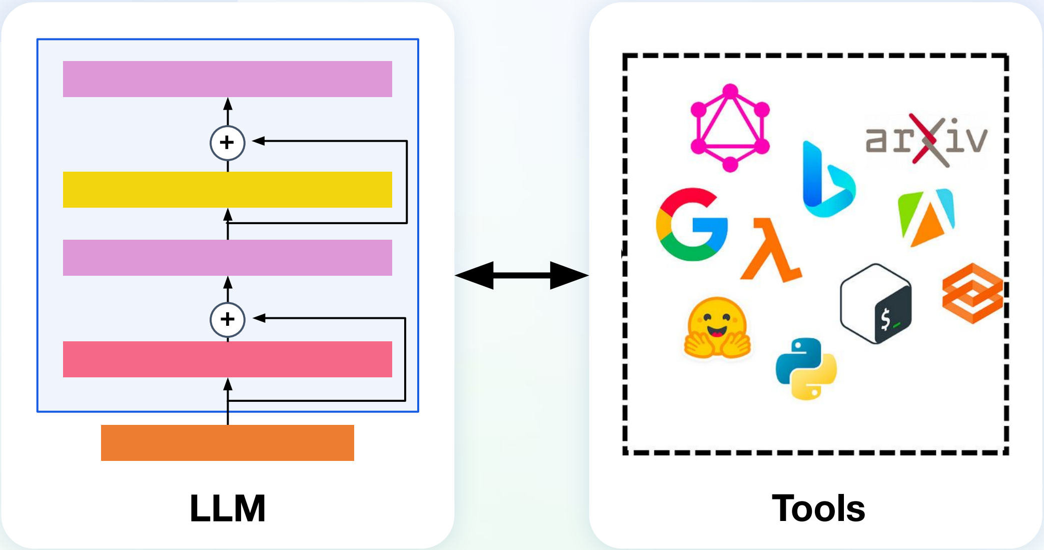
- Chain Of Thought
  - Explain answer step-by-step
- ReAct (Reason + Act)
  - Decompose tasks into:
    - Thought
    - Action
    - Observation
- Reflexion
  - Expand existing frameworks through:
    - Reflexion
    - Heuristic





LLM Agents - Tools

- LLM's can decide when to use tools
- Tools return their results
- LLM's use results in their answers



Takeaways

What did we talk about

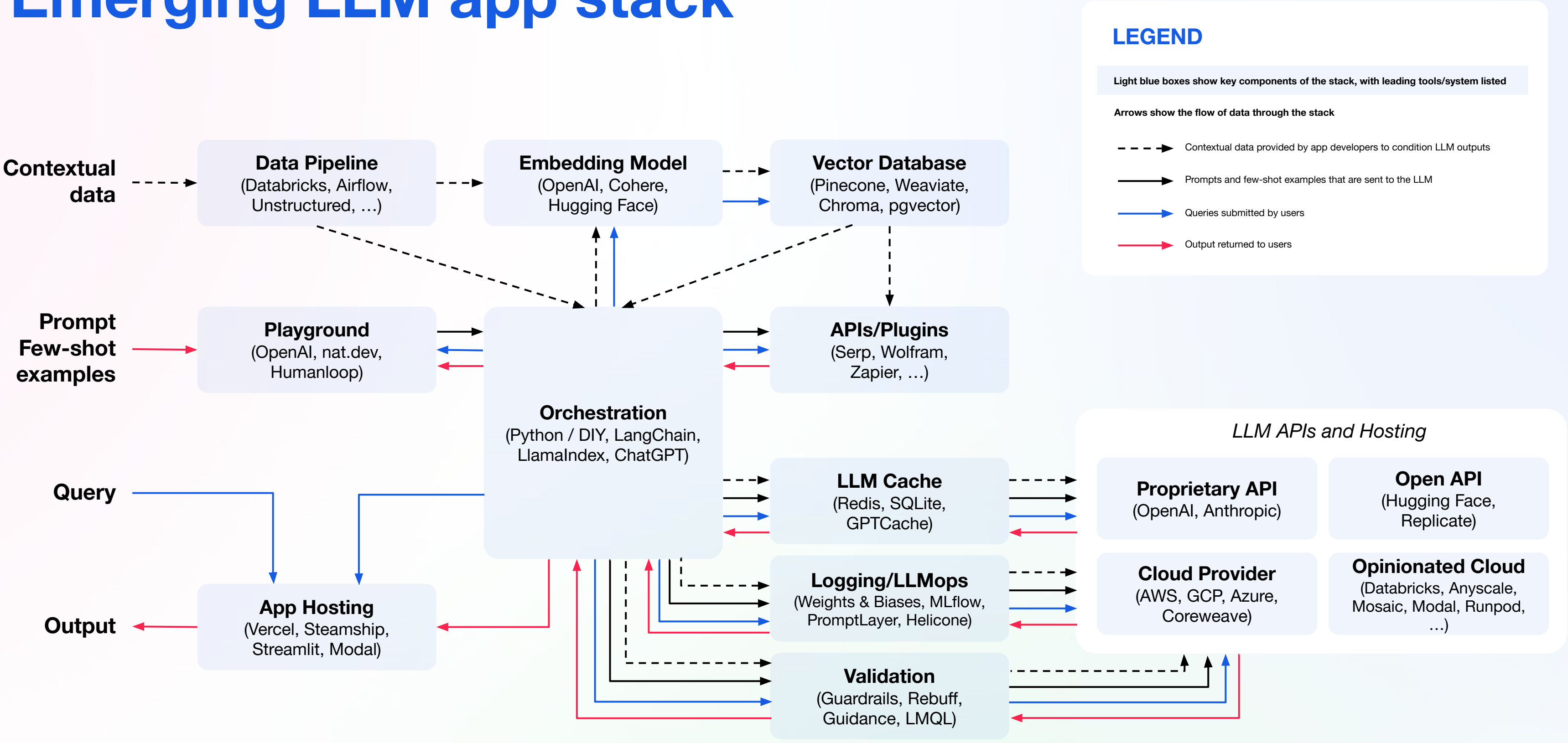
- LLMs are limited
- Architectures can utilize them for real use cases
  - RAG
  - Orchestration
  - Monitoring
  - Agents

Q&A

Oren Razon, CO-Founder & CEO @ Superwise | oren.razon@superwise.ai | linkedin/oren-razon

Gad Benram, Founder & CTO @ TensorOps | gad@tensorops.ai | linkedin/gad-benram

Emerging LLM app stack



Superwise tensorops

References

FLARE - https://arxiv.org/pdf/2305.06983.pdf

Reflexion - https://arxiv.org/pdf/2303.11366.pdf

Agents - LLM Powered Autonomous Agents | Lil'Log

Context Windows - The Secret Sauce behind 100K context window in LLMs: all tricks in one place

Bias - https://arxiv.org/pdf/2306.05685.pdf

Inspiration Blog posts:

- Patterns for Building LLM-based Systems & Products
- Search: Query Matching via Lexical, Graph, and Embedding Methods
- Emerging Architectures for LLM Applications | Andreessen Horowitz