



Asia's Largest

Cloud & AI

Conference 2023

17 – 18, November 2023
IIT Madras Research Park, Chennai



Considerations for LLMOps: Running LLMs in production



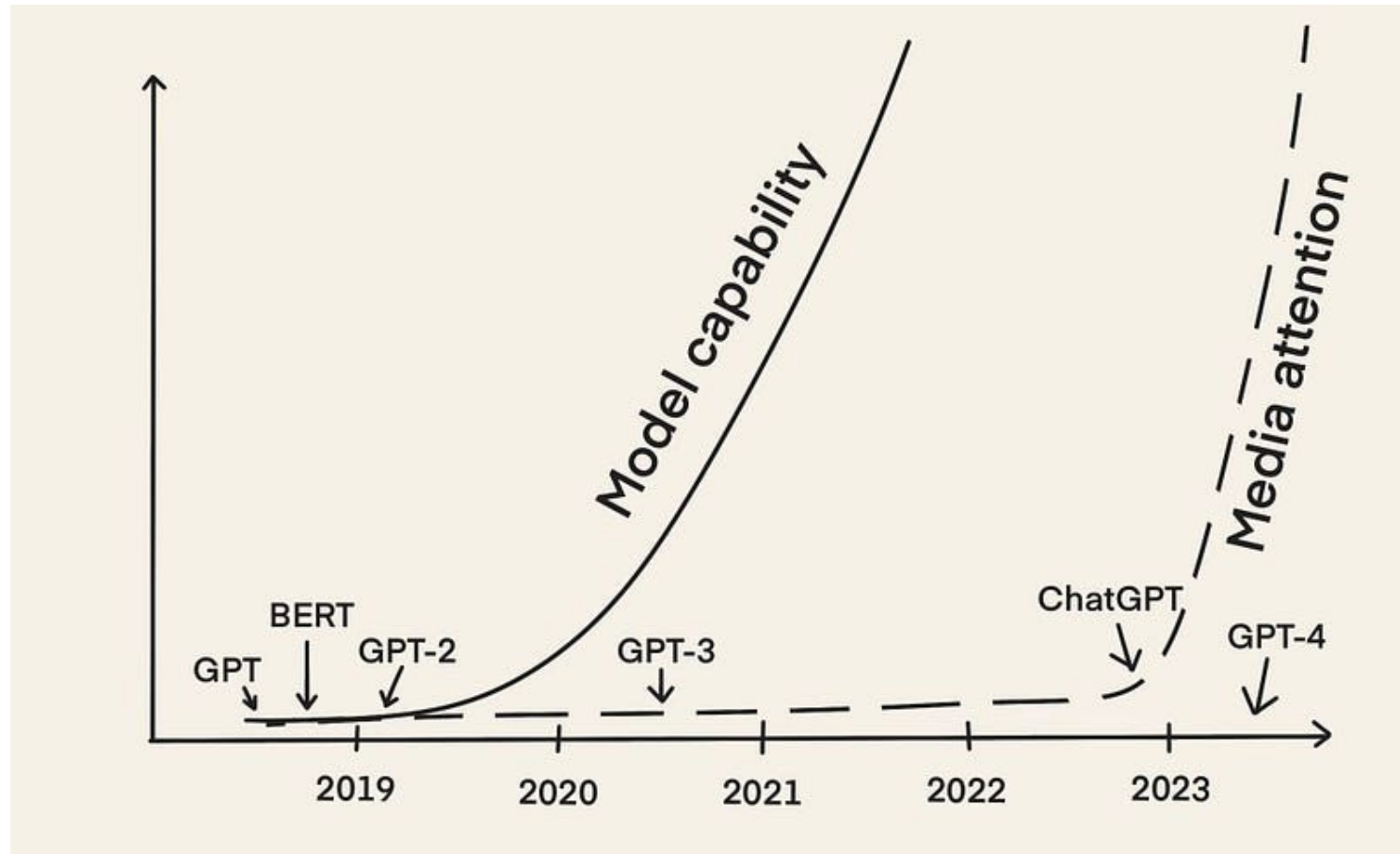
Saradindu Sengupta

Senior ML Engineer, Nunam



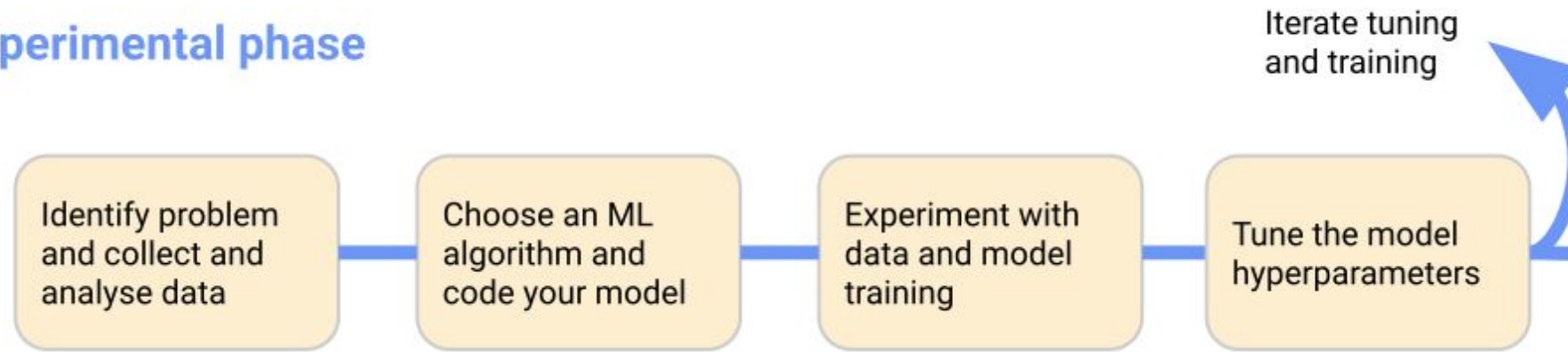
Why such a rush ?

Release of ChatGPT in 2022

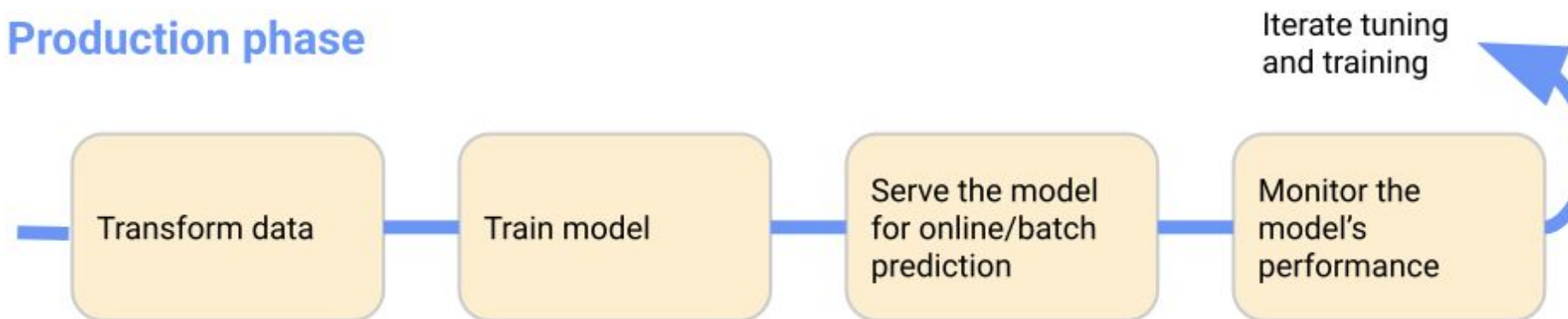


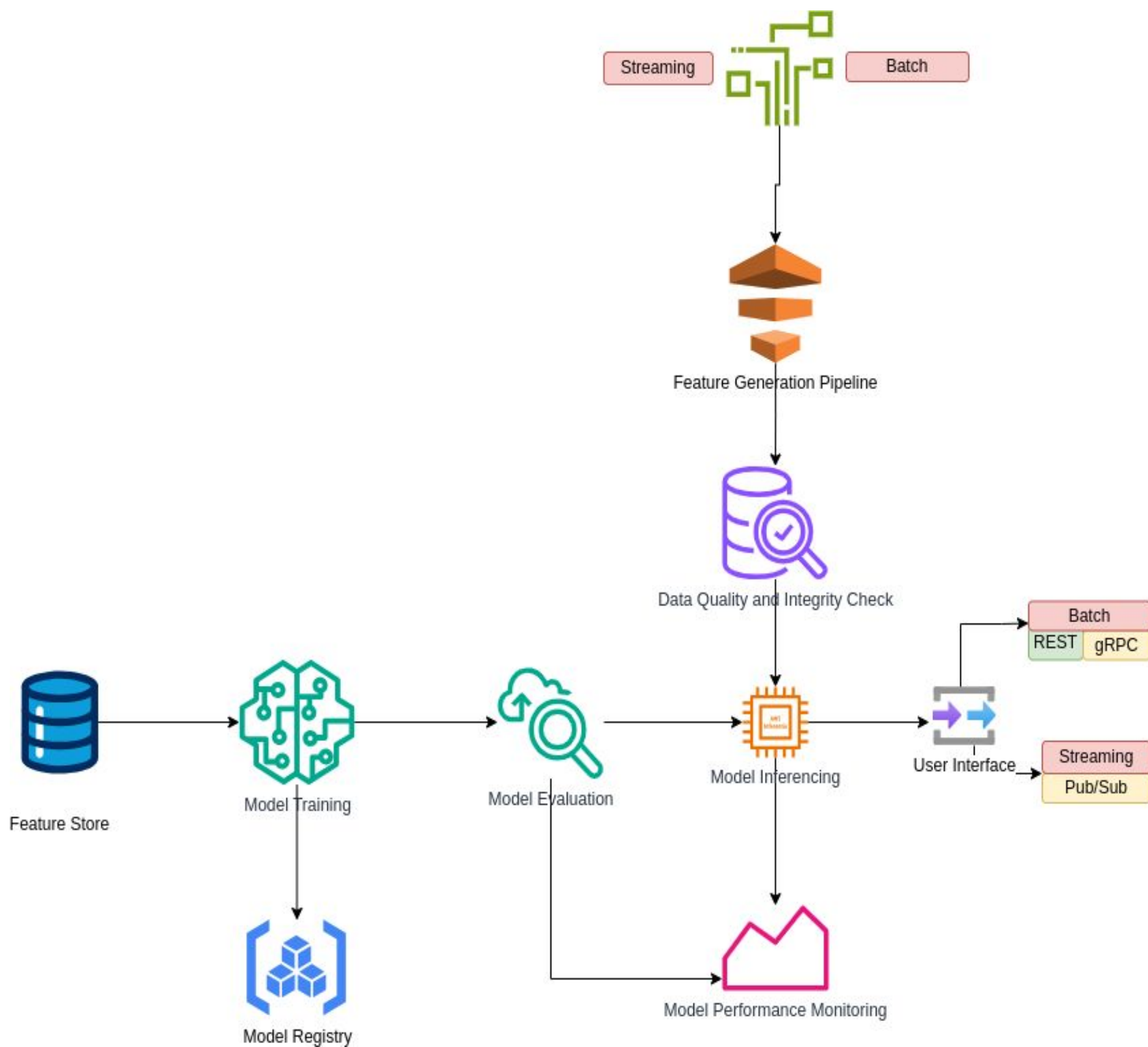
What is not there in MLOps ? Why Another Ops

Experimental phase

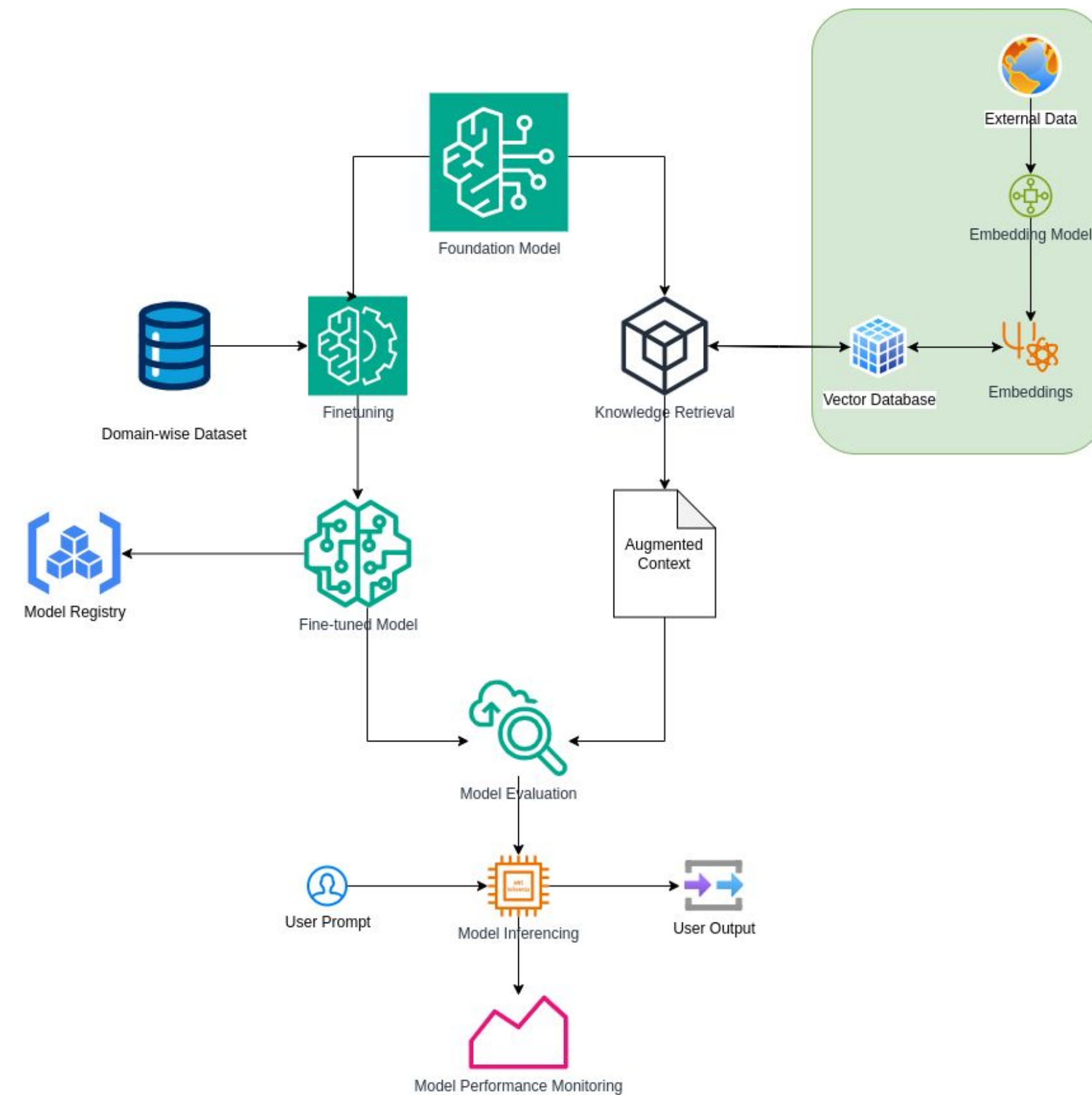


Production phase



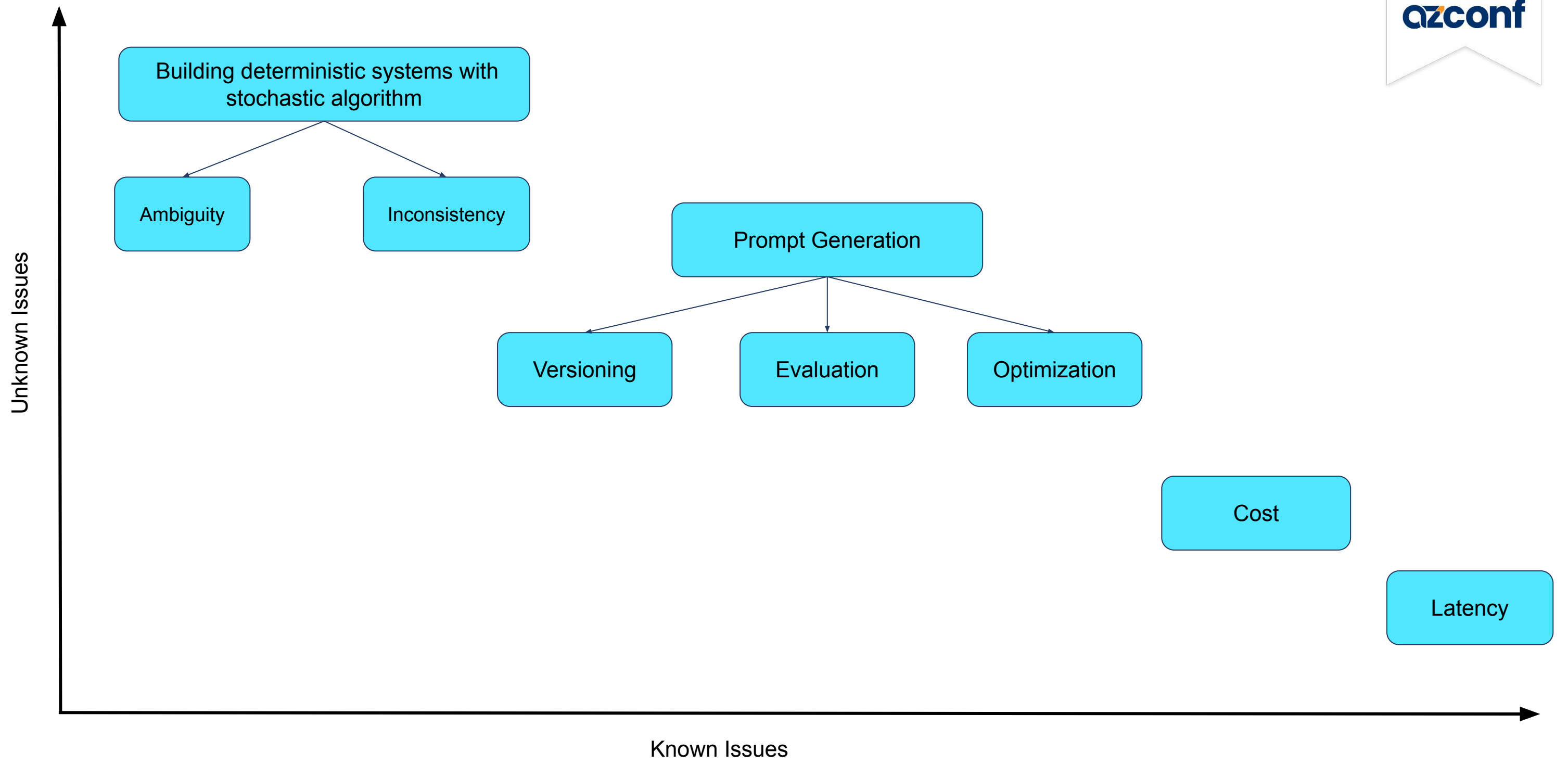


Overview of an ML System



Overview of an LLM System

What are the primary challenges ?



Ambiguity and Inconsistency

Any deterministic downstream application expects output in a certain format.

Consider the case for a weather application using weather API. The responses are non-ambiguous in nature such that the format of the response doesn't change with same given input.

On a different use-case, where scoring an essay is the primary task.

Solution

1. Engineering deterministic response;

[OpenAI Cookbook](#)

2. Changing design mindset to accommodate the stochastic nature



You

As a critical and unbiased English teacher, given an essay give it a score from 0 to 10 where 0 means the essay is bad and 10 means it is good.

Output in the following format:

Essay score: {score}/10 and nothing else.

Here is the essay:

The invention of Braille was a major turning point in the history of disability. The writing system of raised dots used by visually impaired people was developed by Louis Braille in nineteenth-century France.

In a society that did not value disabled people in general, blindness was particularly stigmatized, and lack of access to reading and writing was a significant barrier to social participation.

The idea of tactile reading was not entirely new, but existing methods based on sighted systems were difficult to learn and use. As the first writing system designed for blind people's needs, Braille was a groundbreaking new accessibility tool. It not only provided practical benefits but also helped change the cultural status of blindness. This essay begins by discussing the situation of blind people in nineteenth-century Europe. It then describes the invention of Braille and the gradual process of its acceptance within blind education. Subsequently, it explores the wide-ranging effects of this invention on blind people's social and cultural lives.



ChatGPT

Essay score: 9/10



Cost

Prompt Engineering

Prompt: 10k tokens (\$0.06/1k tokens)
Output: 200 tokens (\$0.12/1k tokens)
Evaluate on 20 examples
Experiment with 25 different versions of prompts
Cost: \$300 - One time

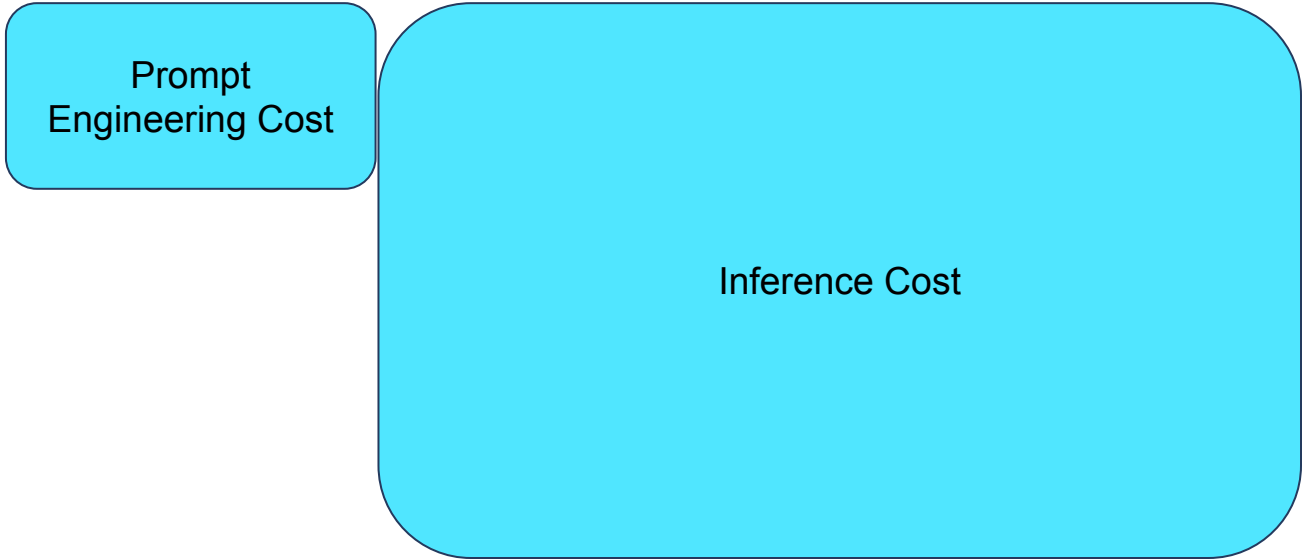
Inferencing

With GPT-4

Input Token: 10K
Output Token: 200
Cost: \$0.624 / inference
With 1 million inference / day: \$624,000 / day

With GPT-3.5-turbo

Input Token: 4K
Output Token: 4K
Cost: \$0.004 / inference
With 1 million inference / day: \$4,000 / day



Latency



- 1. No SLA [Ref: [Running into a Fire: OpenAI Has No SLA \(aragonresearch.com\)](#)]
 - a. Huge networking overhead
 - b. Exponentially volatile development speed
- 2. No clear indication of latency due to model size, networking or engineering overhead
 - a. With newer bigger models model size would be impact factor for latency
 - b. Networking and engineering overhead will be easier with newer releases
- 3. Difficult to estimate cost and latency for LLMs
 - a. Build vs buy estimates are tricky and outdated very quickly
 - b. Open-source vs Proprietary model comparison for enterprise use case are difficult

Latency for short token size:
Input token: 51
Output token: 1
Latency for GPT-3.5-turbo: 500 ms

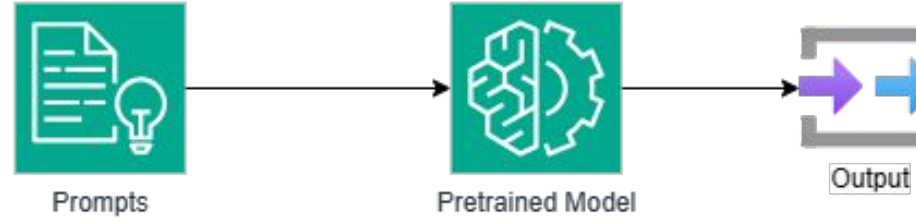
Input Token Size	Output Token Size	Latency (Sec) for 90th Percentile
51	1	0.75
232	1	0.64
228	26	1.62

Prompting vs Fine-tuning

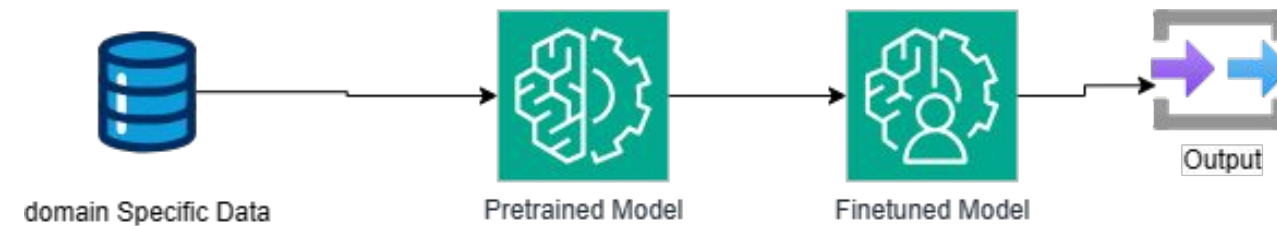
3 main Factor to consider

1. Availability of data
 - a. It is not straightforward to estimate
 - b. For a few it is better to stick to prompts
2. Performance increase
3. Cost reduction

Prompt Engineering



Finetuning



Prompting vs Fine-tuning

1. A prompt is approximately 100 examples.
2. As the number of examples are increased, fine-tuning will always give better result. Although the performance gain will saturate

Why finetune anyways:

1. With more data, model performance will always be better
2. Will be cheaper to run
 - a. Reducing 1K input token in GPT-3.5-turbo will save \$2000 on 1 million inferences

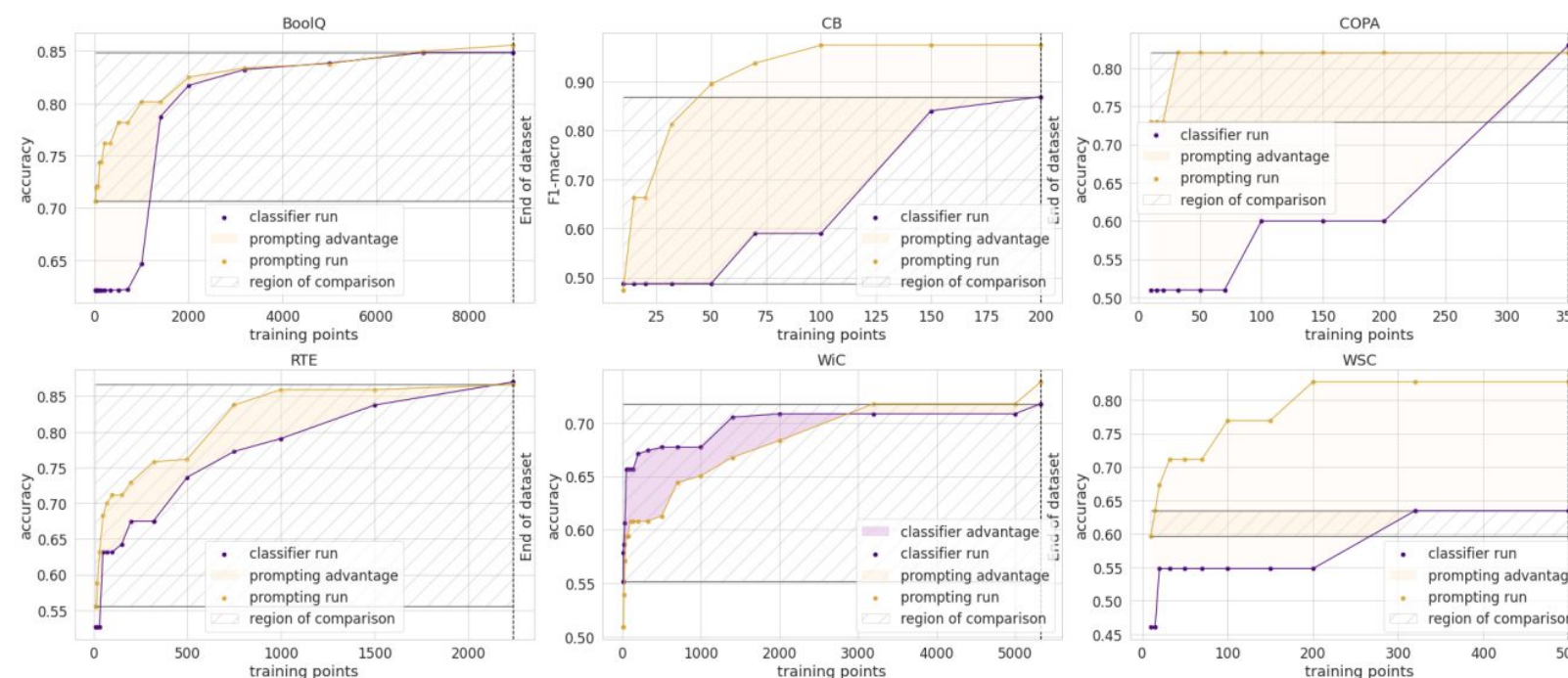


Figure 1: Prompting vs head (classifier) performance across data scales, up to the full dataset, for six SuperGLUE tasks. Compares the best prompt and head performance at each level of training data across 4 runs. Highlighted region shows the accuracy difference of the models. Cross-hatch region highlights the lowest- and highest- accuracy matched region in the curves. The highlighted area in this region is used to estimate the data advantage.

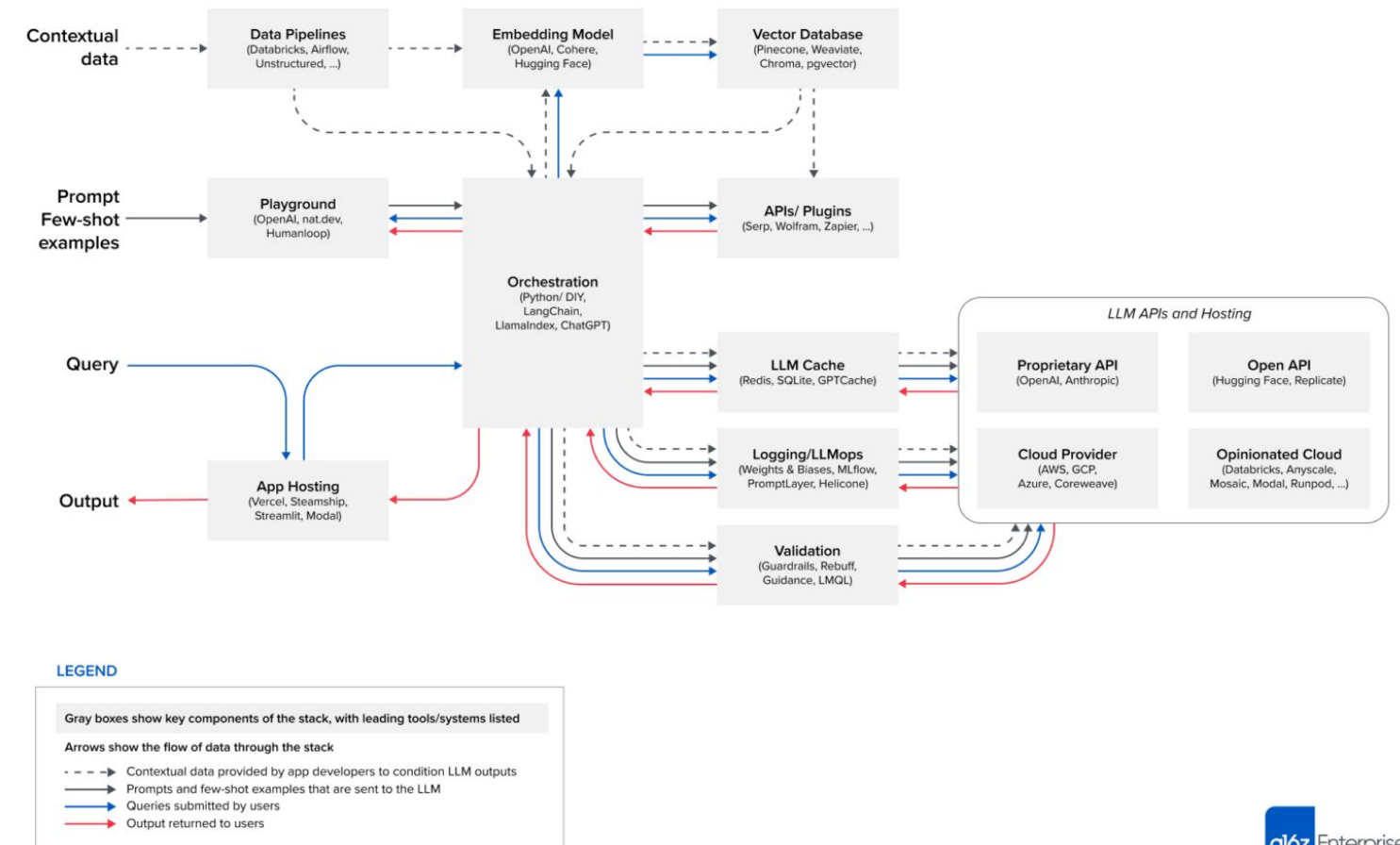
Embeddings and Vector Databases

The cost for embeddings using the smaller model *text-embedding-ada-002* is \$0.0004/1k tokens. If each item averages 250 tokens (187 words), this pricing means \$1 for every 10k items or \$100 for 1 million items.

It is still cheaper

1. Only need to generate once
2. Easy to generate embeddings in real-time for queries

Emerging LLM App Stack



References

1. [MLOps guide \(huyenchip.com\)](https://huyenchip.com)
2. How many data points is a prompt worth [[How Many Data Points is a Prompt Worth? | Abstract \(arxiv.org\)](#)]
3. [OpenAI GPT-3 Text Embeddings - Really a new state-of-the-art in dense text embeddings? | by Nils Reimers | Medium](#)
4. llama-chat [[randaller/llama-chat: Chat with Meta's LLaMA models at home made easy \(github.com\)](#)]
5. [Understanding LLMOps: Large Language Model Operations - Weights & Biases \(wandb.ai\)](#)
6. OpenAI Cookbook [[openai/openai-cookbook: Examples and guides for using the OpenAI API \(github.com\)](#)]



Thank You

Saradindu Sengupta

[LinkedIn](#)

[Twitter](#)

[GitHub](#)



Platinum Partner



Gold Partner



Silver Partner

