Crash Course: Open-Source LLMs

Mikiko Bazeley,
Head of Developer Relations
@Labelbox

Objectives

What we'll cover today:

- Main ways LLMs (both open-source & proprietary) are being using in applications
- Differences between OSS & closed-source models
- Introduce & define some key players & tools in the ecosystem
- Try to run some code samples (if not, we'll at least talk through them so you can run them outside the session)
- Answer Questions: What do you want to know about build LLM applications?

My Background

UCSD

AUTODESK. TELADOC.
Data Science

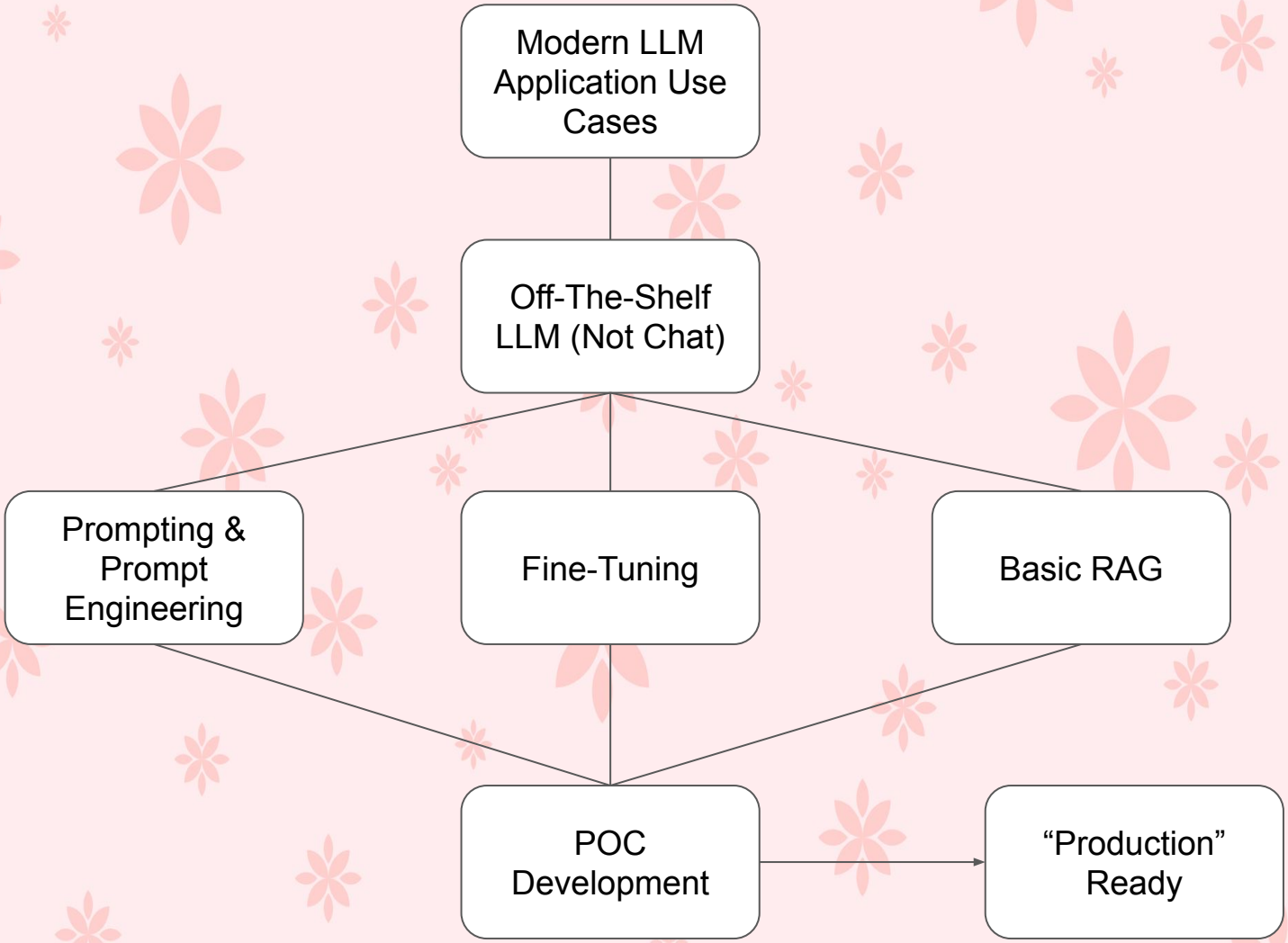
Head of MLOps
(Featureform)

Head of DevRel
(Labelbox)

sunrun walkme
Growth
& Analytics

mailchimp
MLOps

How We'll Proceed



How We'll Proceed

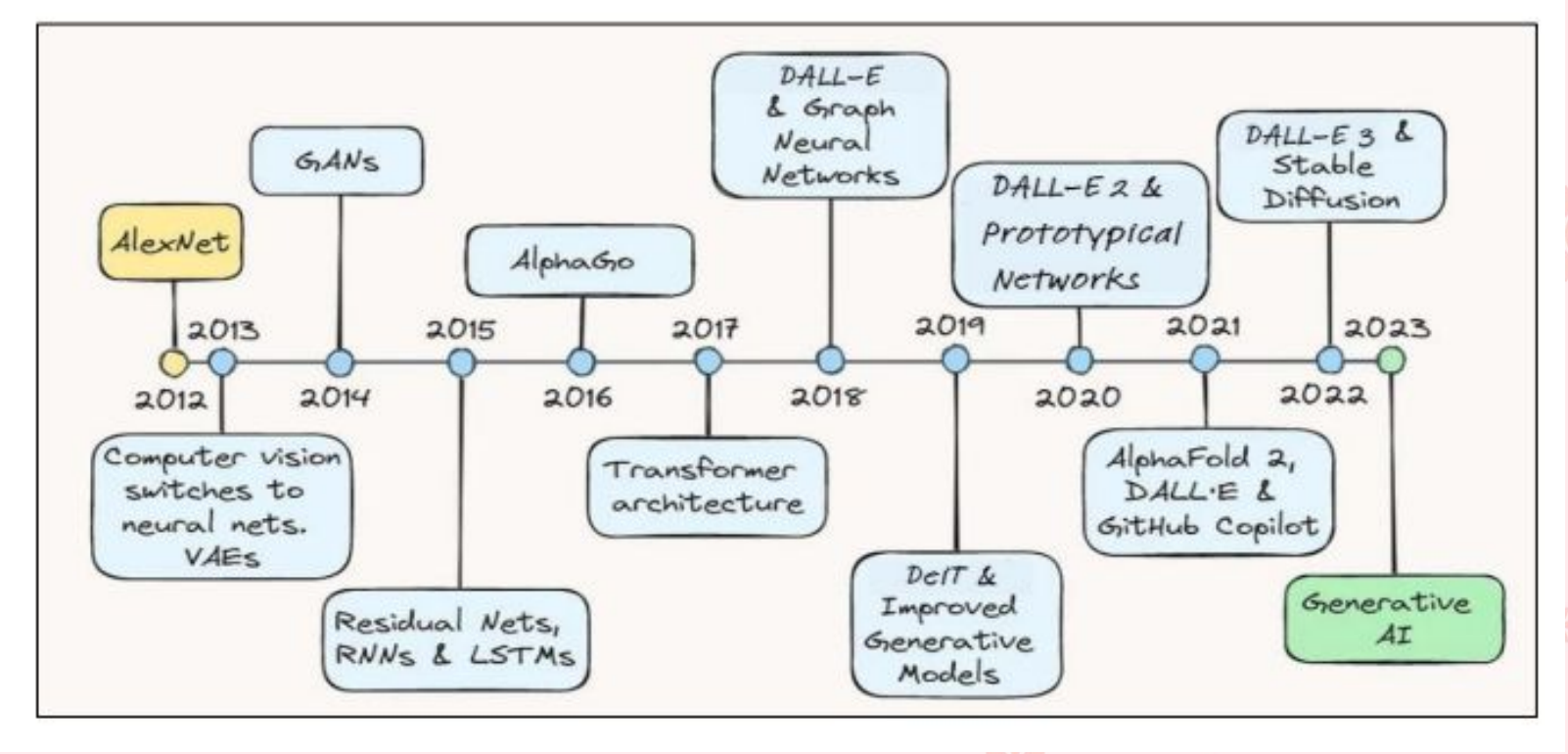
👇 Resources for all talks will be at this link 👇

https://github.com/MMBazel/LO_GenAI_Workshops

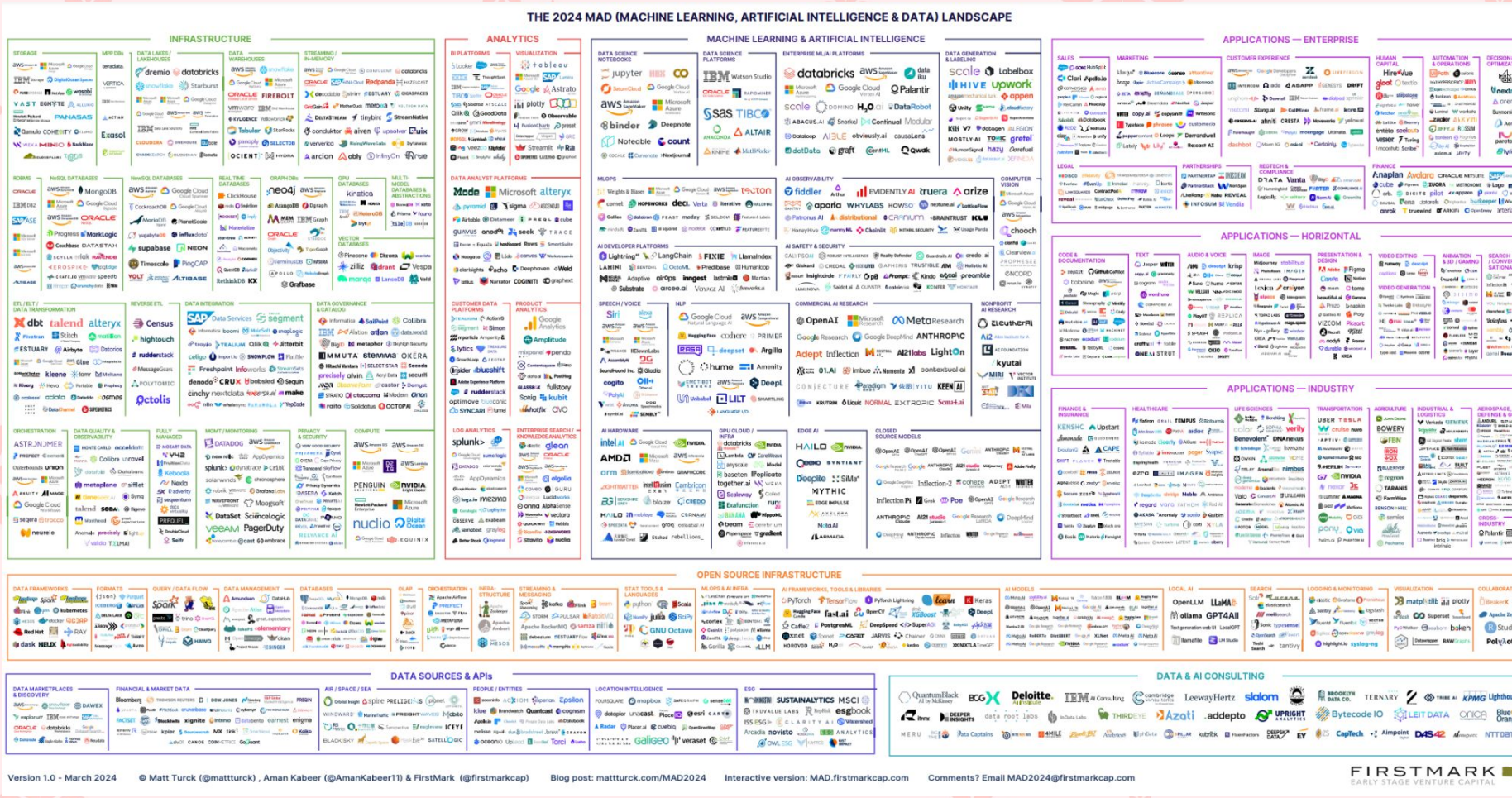
👇 Resources for OSS Models will be under /oss-models 👇

https://github.com/MMBazel/LO_GenAI_Workshops/tree/main/oss-models

We've Come A Long Way

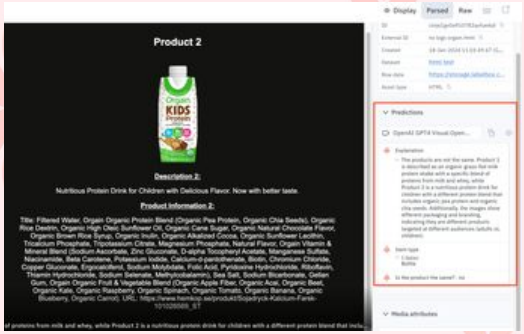


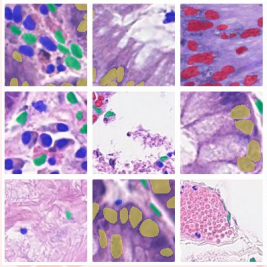
Almost Too Far???

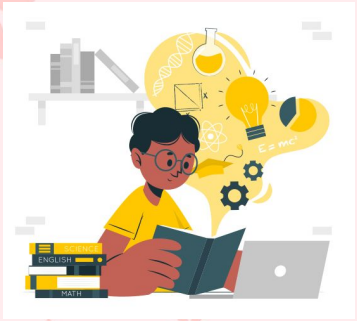


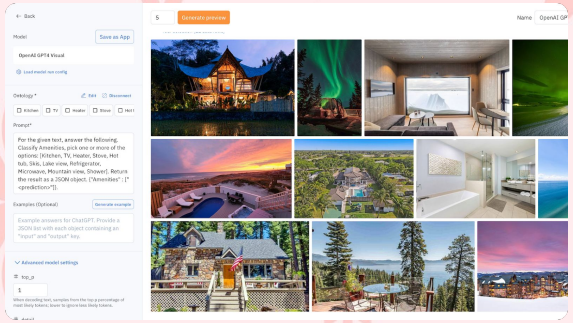
- Note: Diagram doesn't matter as much as the analysis does https://mattturck.com/mad2024/

Language as an interface is in every Gen-AI App









And is used in multiple modalities

Input/Output types:

- 📝 : Text
- 🏞️ : Image
- </> : Code
- 🛠️ : Software tool use (text, code generation & execution)
- 🎥 : Video
- 🎵 : Music
- 📦 : 3D
- 🤖 : Robot state

Model types:

- 📝 → 📝 : LLMs
- 📝 + 🏞️ → 📝 : Multimodal LLMs
- 📝 + 🏞️ + 🤖 → 📝 : Multimodal LLMs for Robotics
- 📝 → </> : Text to Code
- 📝 → 🛠️ : Text to Software tool use
- 📝 → 🏞️ : Text to Image
- 📝 → 🎥 : Text to Video
- 📝 → 🎵 : Text to Music
- 🏞️ → 📦 : Image to 3D
- 📝 → 📦 : Text to 3D

There are currently more than 18.9K Text-to-Image generative models listed on Hugging Face

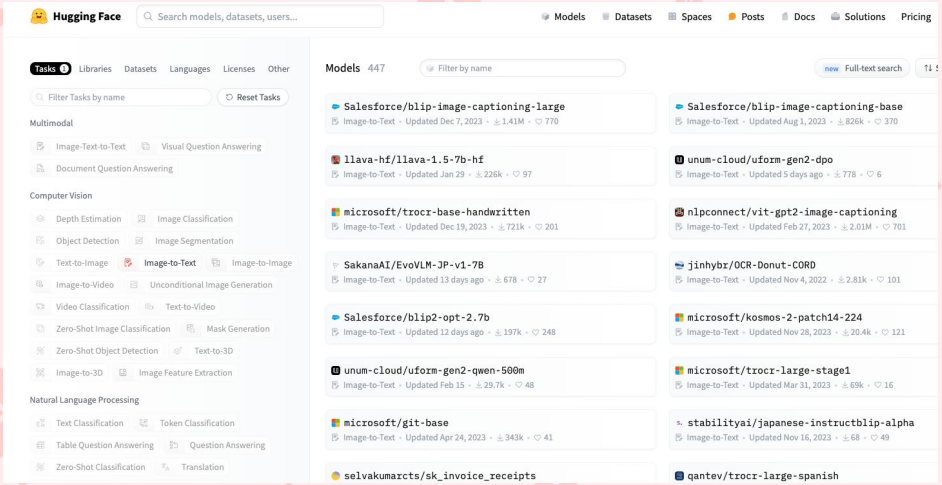


Table 1: Summary of Video Generation.

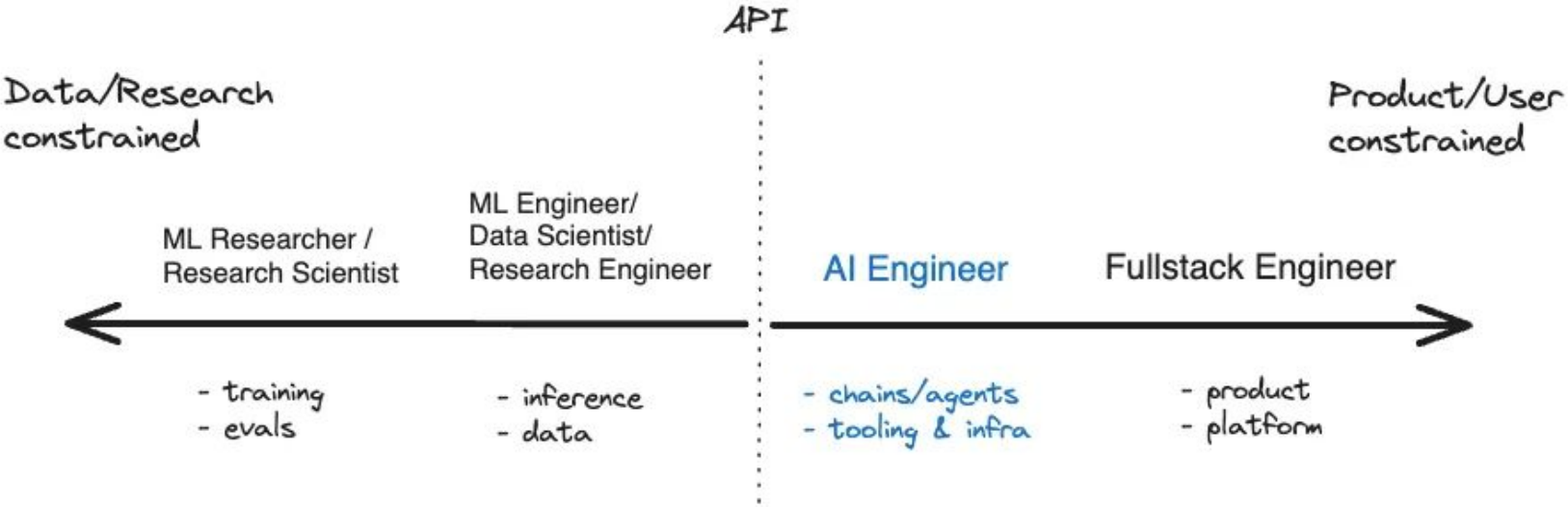
Model name	Year	Backbone	Task	Group
Imagen Video[29]	2022	Diffusion	Generation	Google
Pix2Seq-D[160]	2022	Diffusion	Segmentation	Google Deepmind
FDM[161]	2022	Diffusion	Prediction	UBC
MaskViT[162]	2022	Masked Vision Models	Prediction	Stanford, Salesforce
CogVideo[163]	2022	Auto-regressive	Generation	THU
Make-a-video[164]	2022	Diffusion	Generation	Meta
MagicVideo[165]	2022	Diffusion	Generation	ByteDance
TATS[166]	2022	Auto-regressive	Generation	University of Maryland, Meta
Phenaki[167]	2022	Masked Vision Models	Generation	Google Brain
Gen-1[168]	2023	Diffusion	Generation, Editing	RunwayML
LFDM[140]	2023	Diffusion	Generation	PSU, UCSD
Text2video-Zero[169]	2023	Diffusion	Generation	Picsart
Video Fusion[170]	2023	Diffusion	Generation	USAC, Alibaba
PYoCo[34]	2023	Diffusion	Generation	Nvidia
Video LDM[36]	2023	Diffusion	Generation	University of Maryland, Nvidia
RIN[171]	2023	Diffusion	Generation	Google Brain
LVD[172]	2023	Diffusion	Generation	UCB
Dreamix[173]	2023	Diffusion	Editing	Google
MagicEdit[174]	2023	Diffusion	Editing	ByteDance
Control-A-Video[175]	2023	Diffusion	Editing	Sun Yat-Sen University
StableVideo[176]	2023	Diffusion	Editing	ZJU, MSRA
Tune-A-Video[78]	2023	Diffusion	Editing	NUS
Rerender-A-Video[177]	2023	Diffusion	Editing	NTU
Pix2Video[178]	2023	Diffusion	Editing	Adobe, UCL
InstructVid2Vid[179]	2023	Diffusion	Editing	ZJU
DiffAct[180]	2023	Diffusion	Action Detection	University of Sydney
DiffPose[181]	2023	Diffusion	Pose Estimation	Jilin University
MAGVIT[182]	2023	Masked Vision Models	Generation	Google
AnimateDiff[138]	2023	Diffusion	Generation	CUHK
MAGVIT V2[47]	2023	Masked Vision Models	Generation	Google
Generative Dynamics[183]	2023	Diffusion	Generation	Google
VideoCrafter[81]	2023	Diffusion	Generation	Tencent
Zeroscope[184]	2023	-	Generation	EasyWithAI
ModelScope	2023	-	Generation	Damo
Gen-2[23]	2023	-	Generation	RunwayML
Pika[22]	2023	-	Generation	Pika Labs
Emu Video[185]	2023	Diffusion	Generation	Meta
PixelDance[186]	2023	Diffusion	Generation	ByteDance
Stable Video Diffusion[27]	2023	Diffusion	Generation	Stability AI
W.A.L.T[187]	2023	Diffusion	Generation	Stanford, Google
Fairy[188]	2023	Diffusion	Generation, Editing	Meta
VideoPoet[189]	2023	Auto-regressive	Generation, Editing	Google
LGVI[190]	2024	Diffusion	Editing	PKU, NTU
Lumiere[191]	2024	Diffusion	Generation	Google
Sora[3]	2024	Diffusion	Generation, Editing	OpenAI

In order to build a career as an AI Engineer, you need to understand the low-level “why”

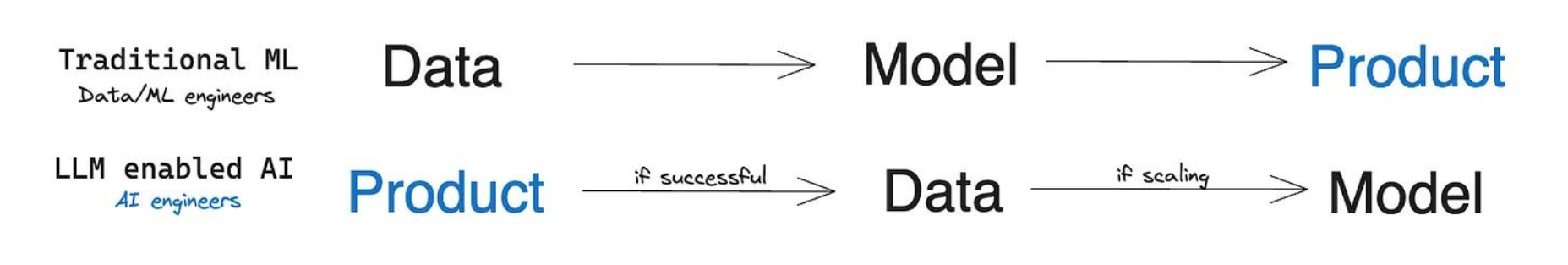
Someone else’s POC Gen-AI App on Twitter

Your results

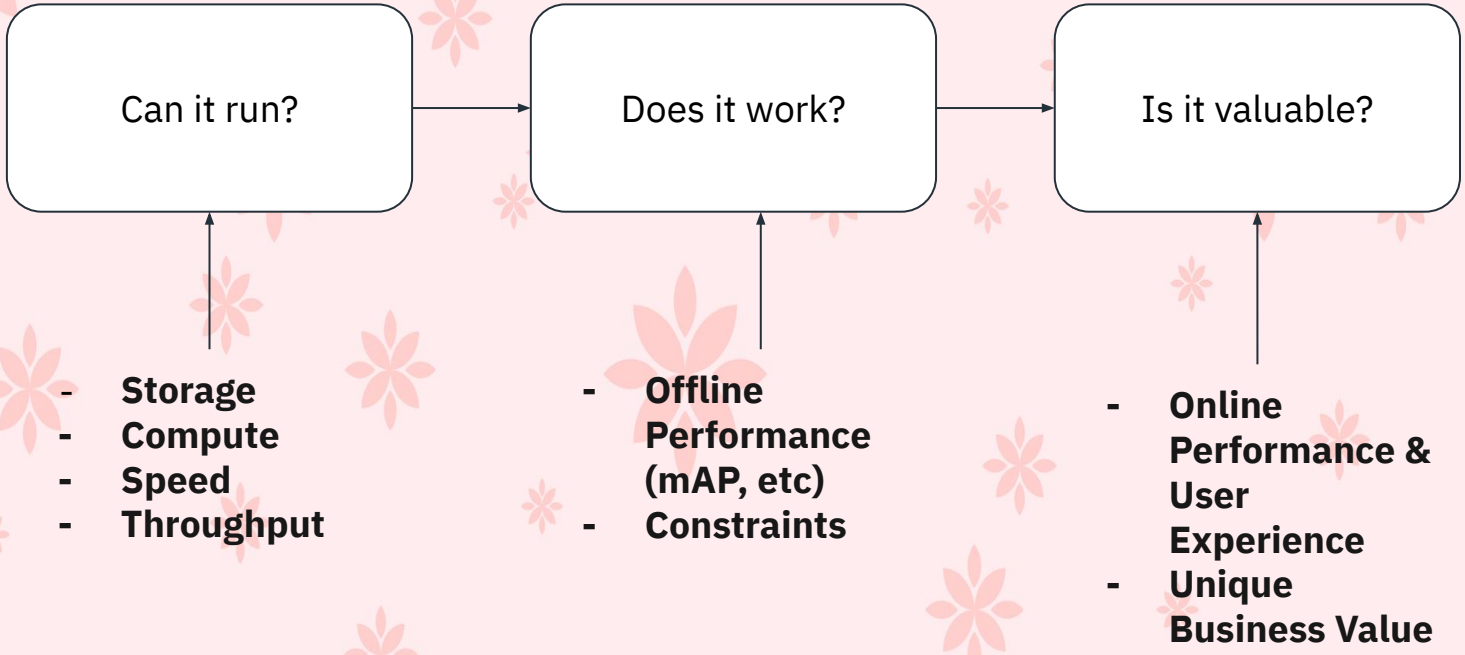
In order to build a career as an AI Engineer, you need to understand the low-level “why”



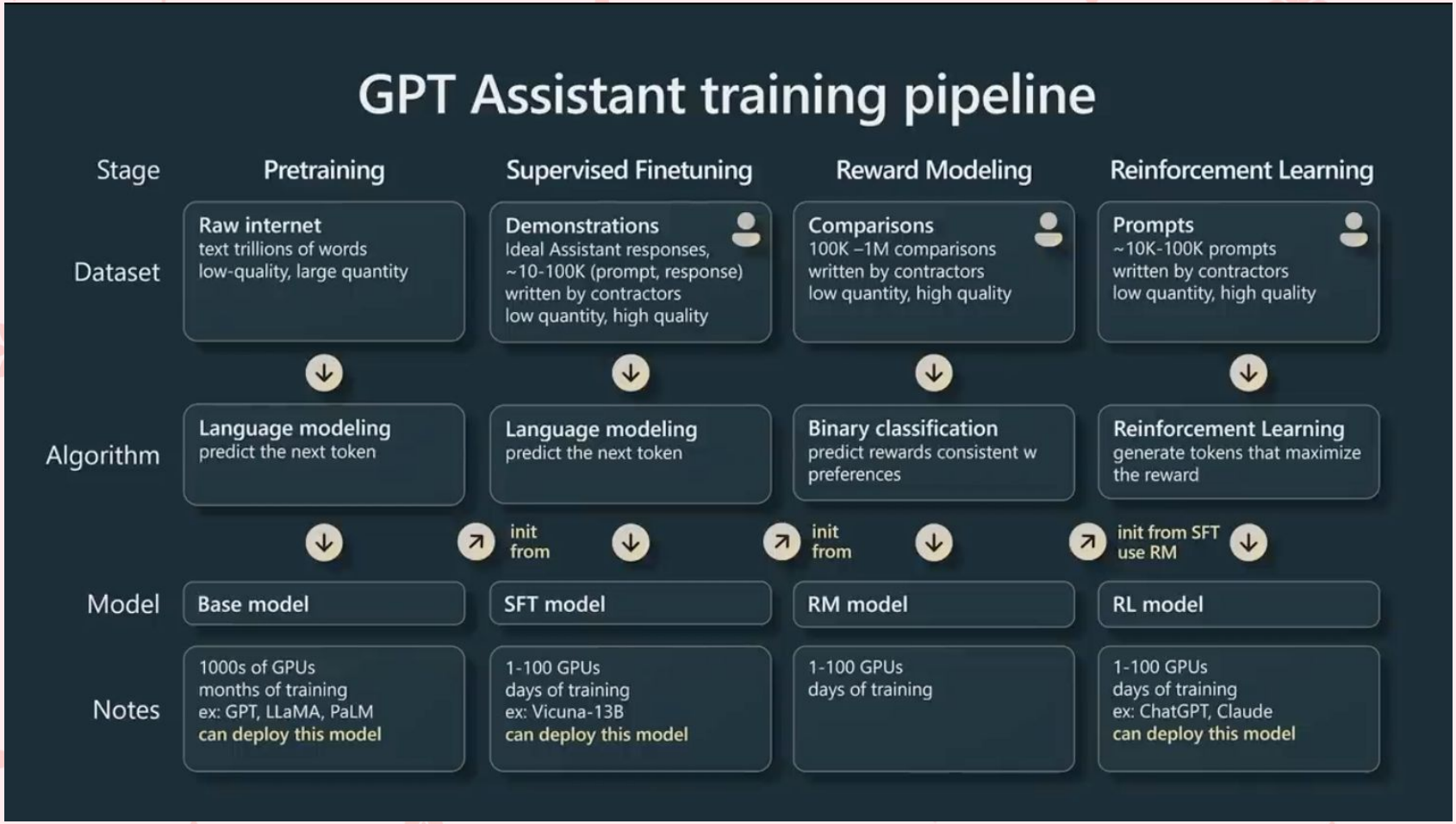
In order to build a career as an AI Engineer, you need to understand the low-level “why”



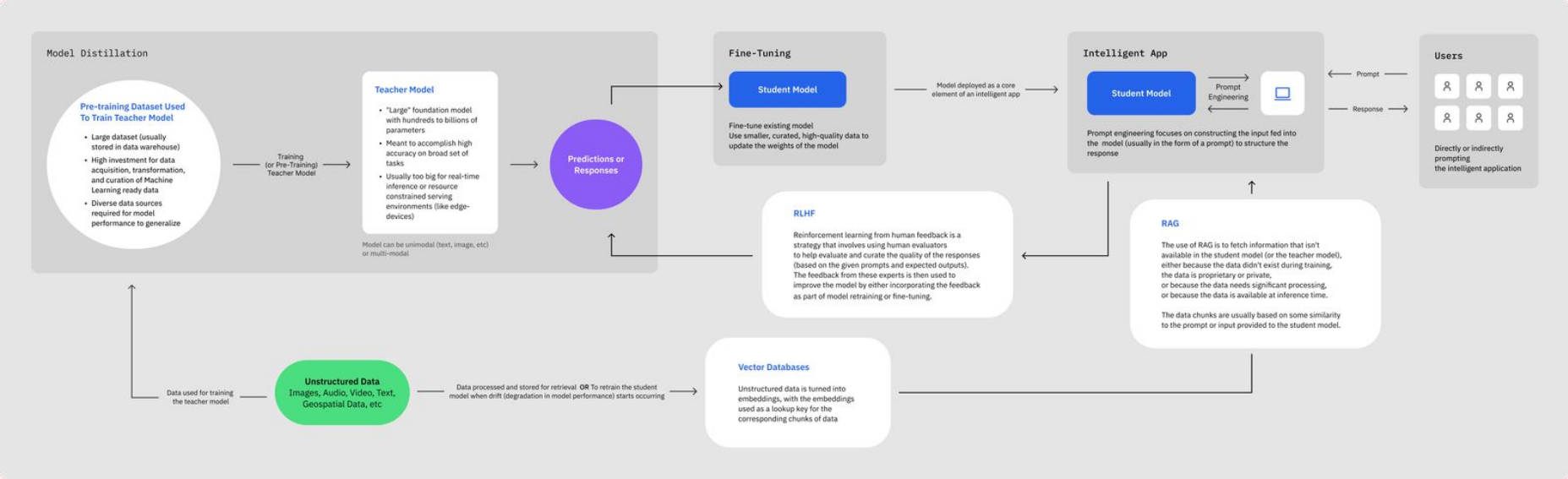
Production Gen-AI Applications are complicated & scaling requires considerations that are hard to address in the short-time we have



💬 Discussion Question: Is ChatGPT an LLM? 💬

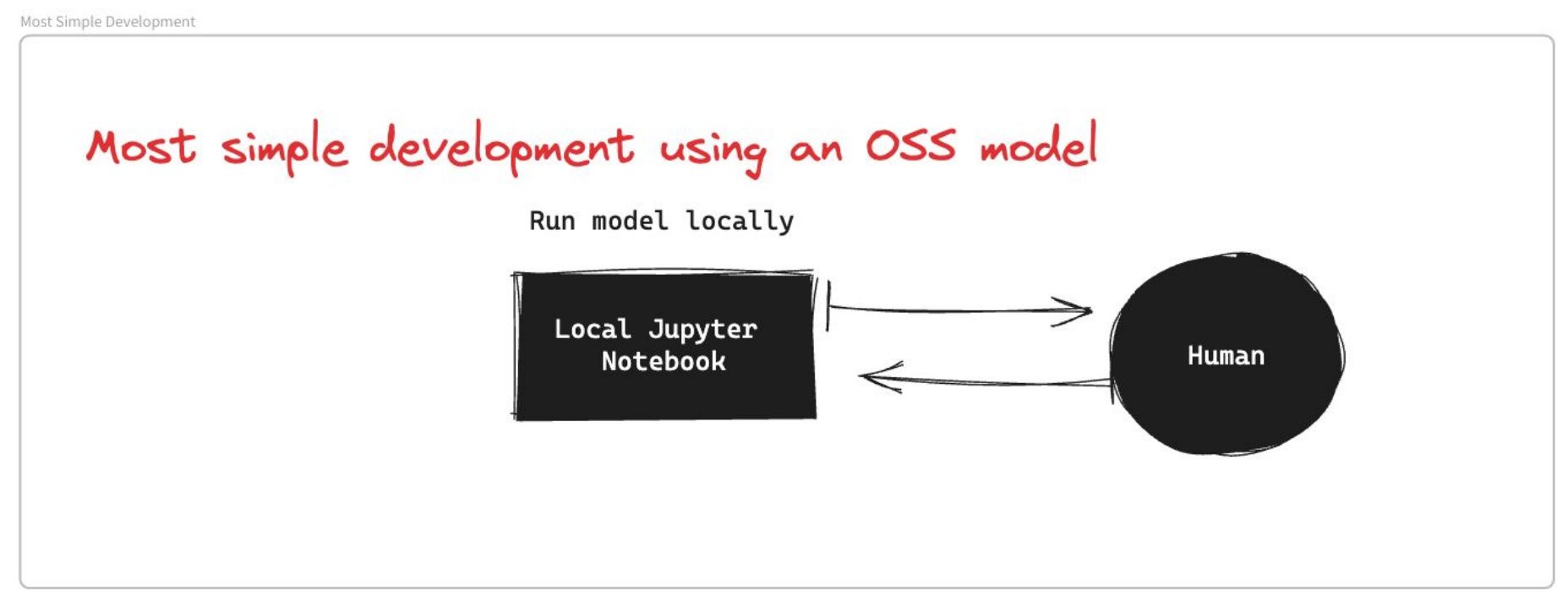


Production Gen-AI Apps:
(Model Distillation +) Fine-Tuning + RAG + Evaluation _ RLHF



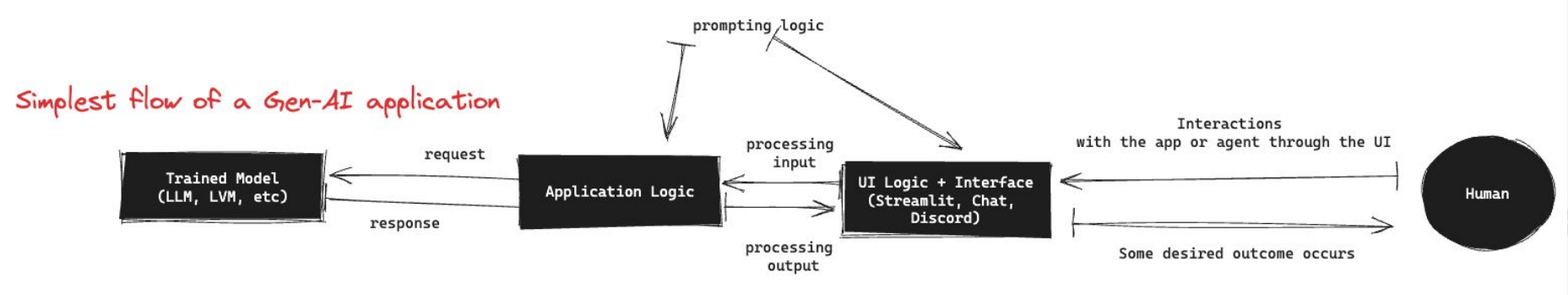
- Note: If you can’t see the diagram, the original one is here: https://labelbox.com/blog/a-pragmatic-introduction-to-model-distillation-for-ai-developers/

Easiest Way To Interact & Develop With An LLM



- You might use a tool like Ollama to run an LLM locally but for the most part you can be really slim on libraries like Langchain, etc
- UX will be pretty low-level & may need to specify “chat-like” prompts

Simplified Flow of Prompting

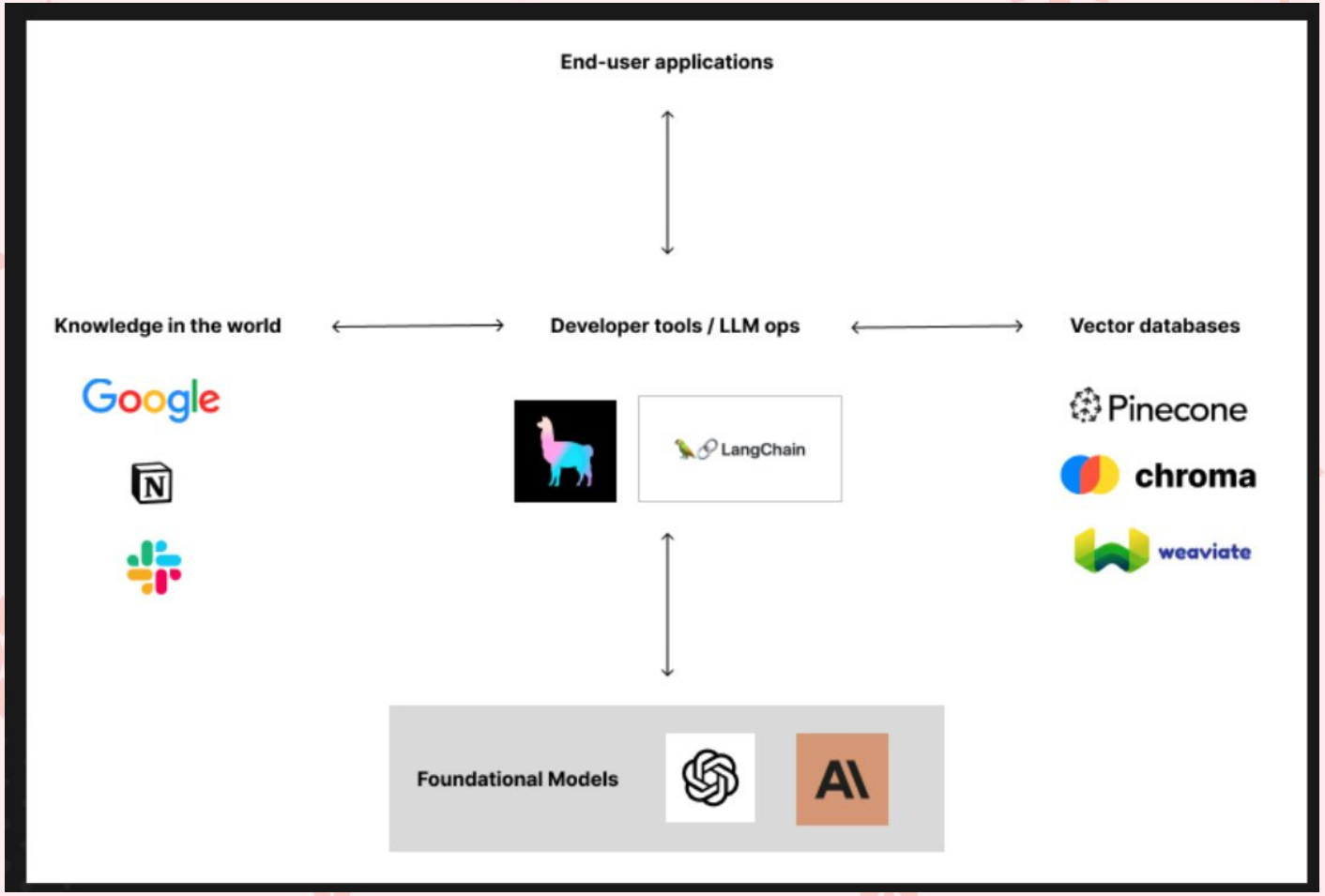


- OpenAI's Guide: https://platform.openai.com/docs/guides/prompt-engineering

LLM Stack

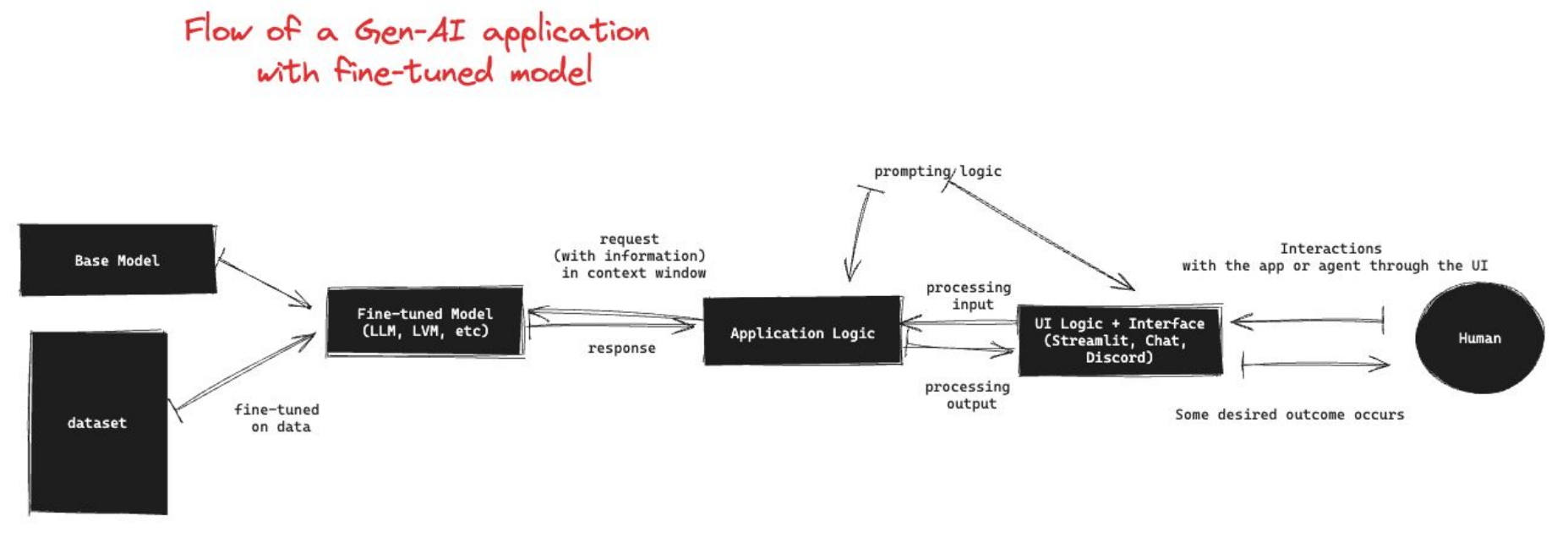
- Medium: https://medium.com/plain-simple-software/the-llm-app-stack-2024-eac28b9dc1e7

LLM Stack



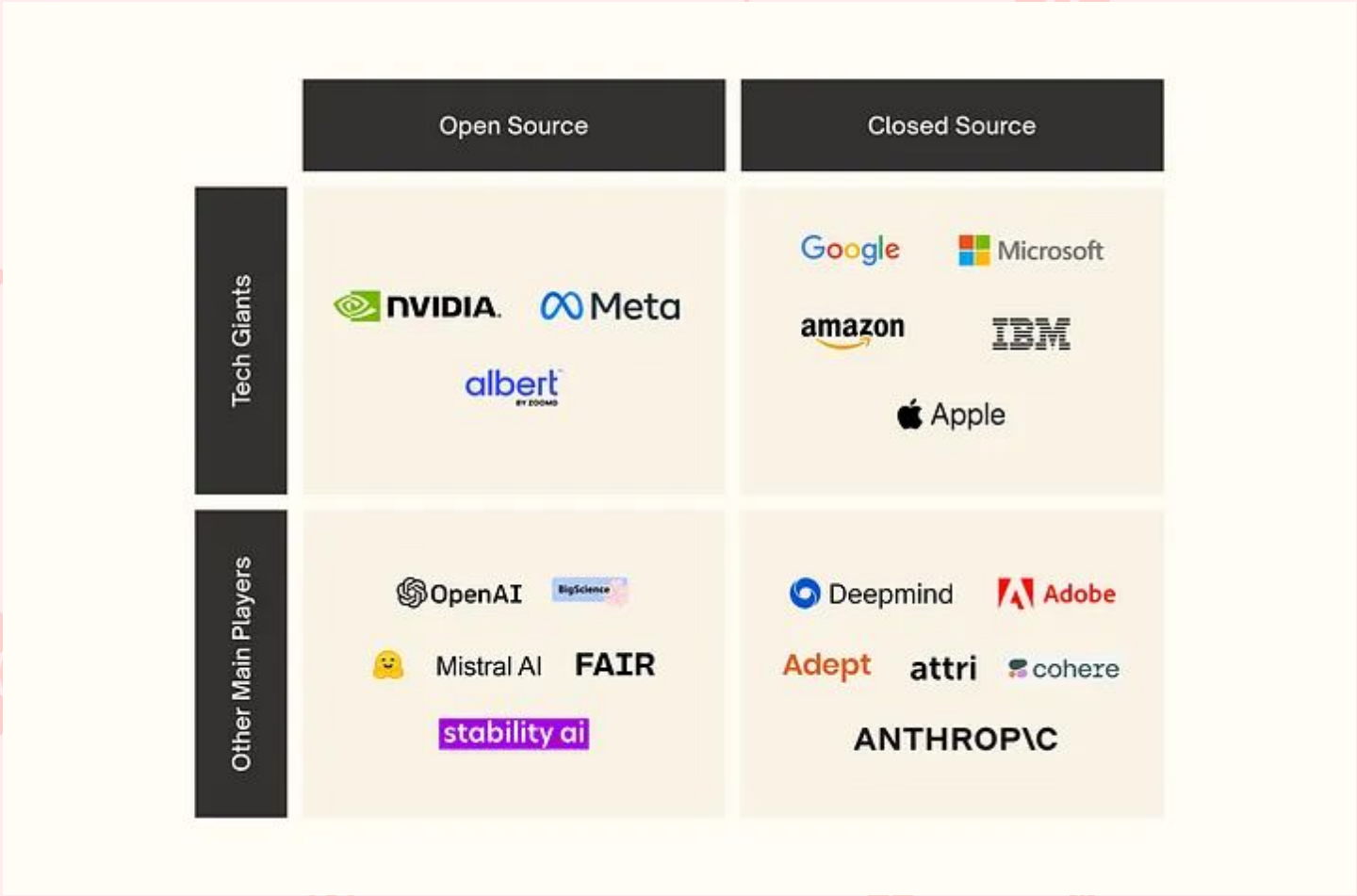
- AI Makerspace: https://www.canva.com/design/DAFsTCKwPGw/sxlXSvaU3C6HQCNQRftp9g/view

Simplified Flow of Fine-Tuned App



-

Why Open-Source? And who OS?



- Original link: [Choosing Your Path in Generative AI: Open-Source or Proprietary?]

Criteria	Commercial (Proprietary) Models	Open Source Models	Best Fit
Pricing	Usage-based, often with subscription models.	Generally free, potential costs for extra services.	Commercial: Ideal for businesses needing predictable costs. OSS: Fits budget-flexible or cost-sensitive projects.
Flexibility & Customization	Limited customization, standardized solutions.	High customization potential.	Commercial: Suited for standard solution requirements. OSS: Best for highly customized project needs.
Integration Capabilities	Better integration with existing ecosystems.	Requires more effort for integration.	Commercial: Fits businesses with existing infrastructure needing seamless integration. OSS: Suitable for environments where custom integration is practical.
Scalability	Designed for scalability and easier expansion.	Highly scalable but requires expertise.	Commercial: Ideal for rapidly growing businesses needing scalable solutions. OSS: Fits companies with the capability to manage scalability in-house.
Security & Compliance	Robust security and compliance with regulations.	Varies, may need additional work.	Commercial: Best for industries with stringent security and compliance needs. OSS: Suitable for scenarios, allowing for custom security adaptations.
Support & Maintenance	Structured support and maintenance.	Community-based, varies in quality.	Commercial: Suited for organizations needing consistent, dedicated support. OSS: Ideal for entities capable of self-maintenance and leveraging community support.
Community & Ecosystem	Certified partners and service providers.	Broad, collaborative community.	Commercial: Suitable for businesses seeking established service networks. OSS: Best for leveraging collaborative community input and development.
Long-term Viability	Predictable due to commercial backing.	Depends on community support.	Commercial: Ideal for organizations preferring stability and long-term support. OSS: Fits projects adaptable to community-driven changes and developments.

- Original link: [Choosing Your Path in Generative AI: Open-Source or Proprietary?]

Considerations In Choosing Models

Accuracy
vs
Speed

Customizability
vs
Ease

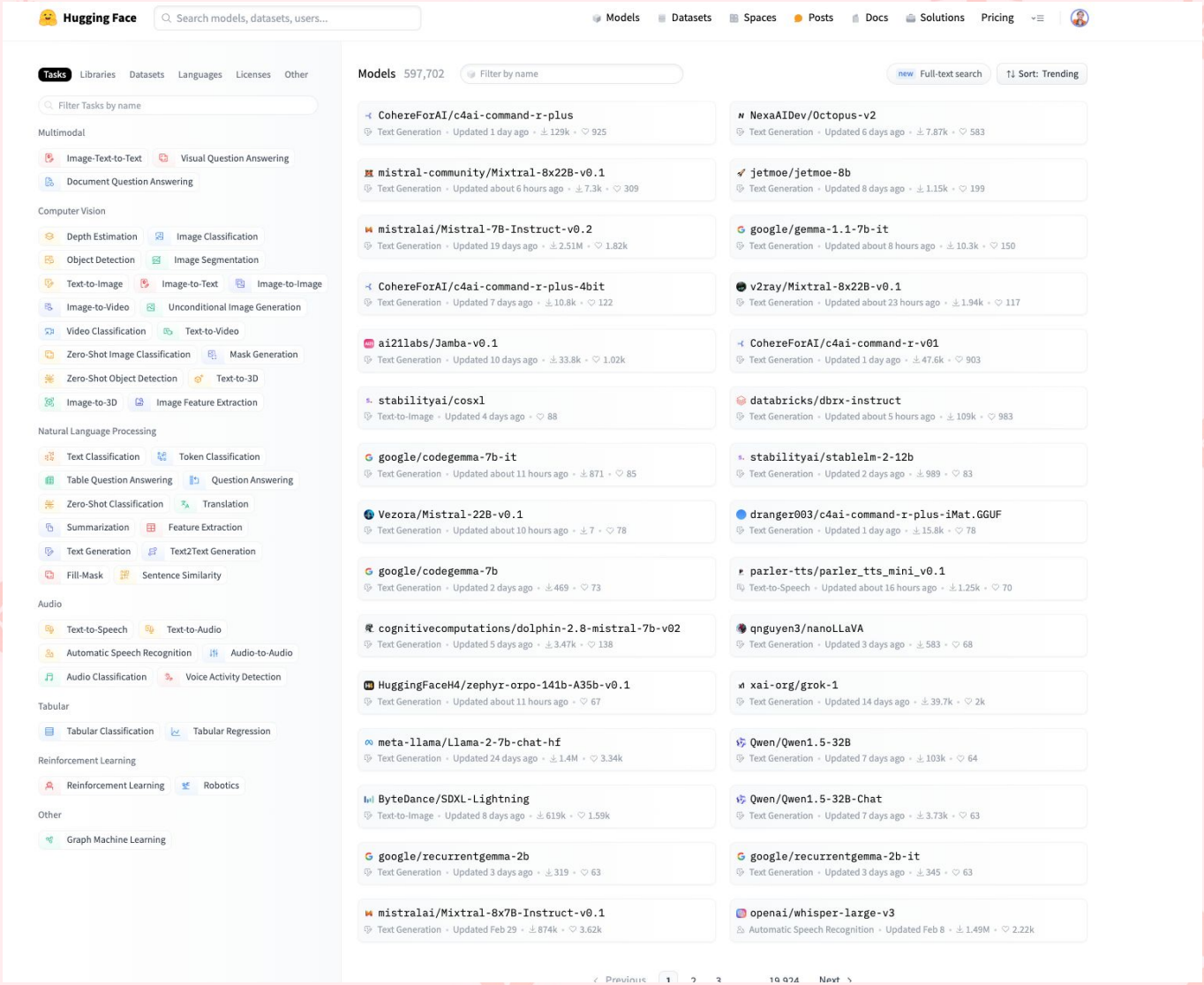
Product
Requirements

Memory
vs
Computation

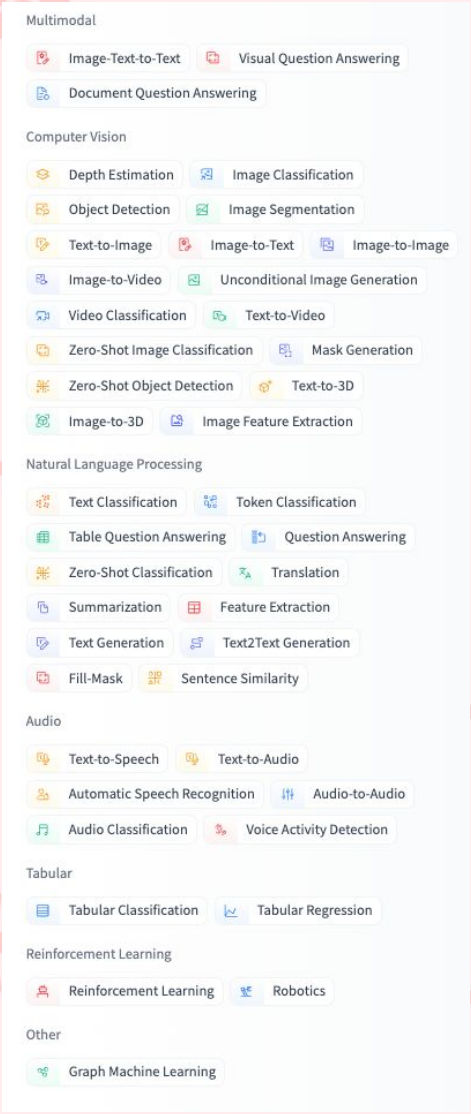
Generalized
vs
Specialized

Licensing

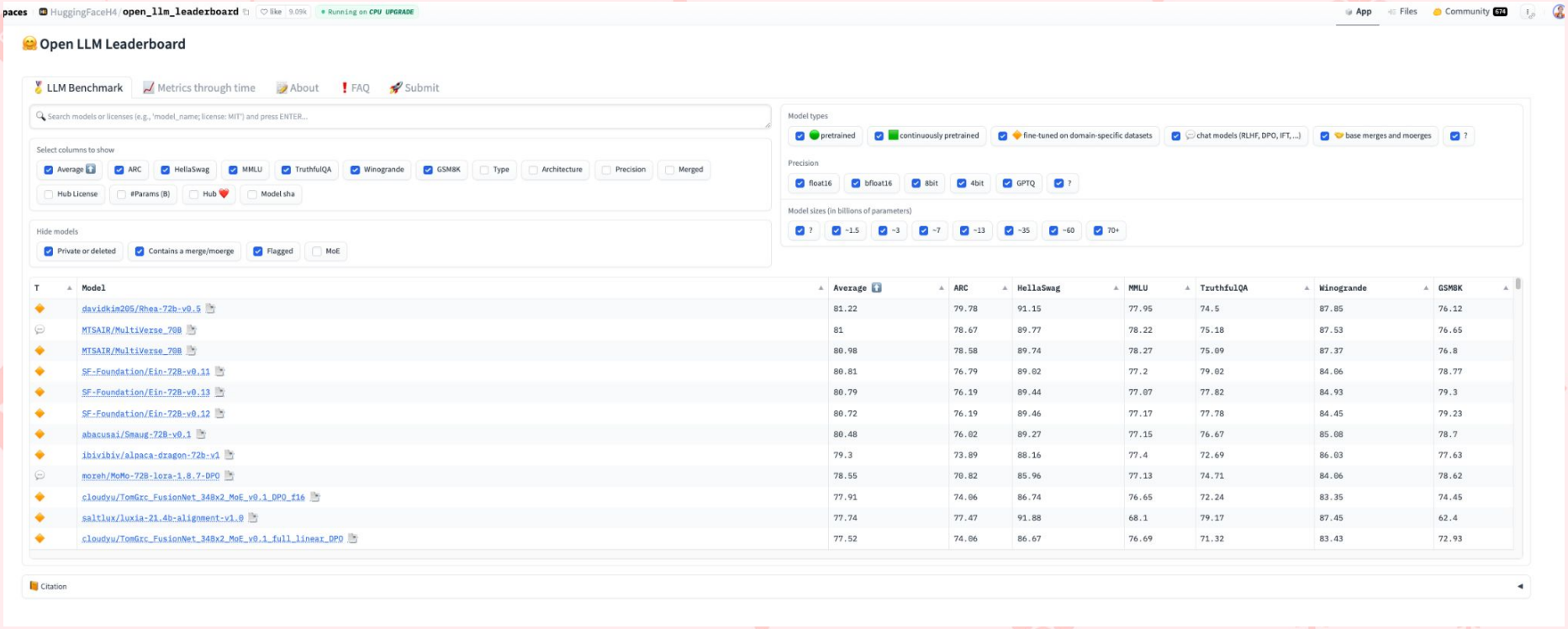
The De Facto Site For Models (Esp. OS)



The De Facto Site For Models (Esp. OS)

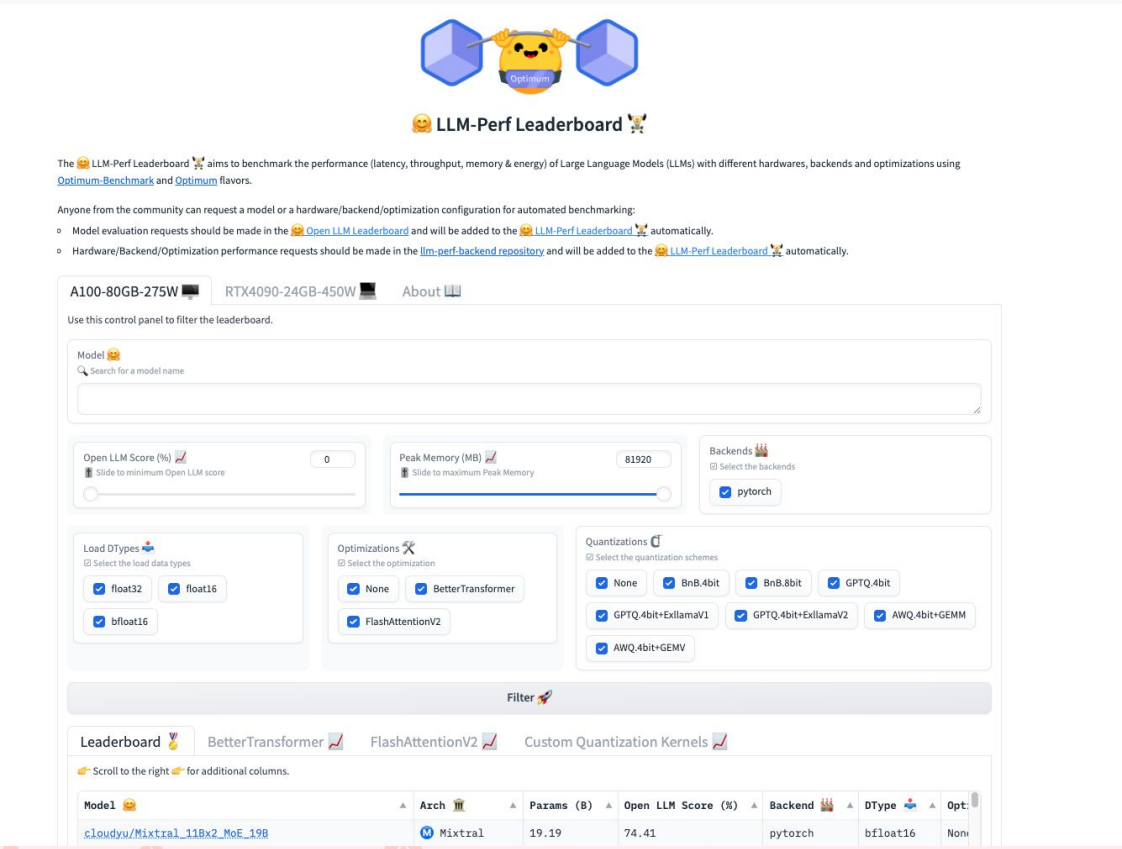


Also has leaderboards: Open LLM Leaderboard



- https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Also has leaderboards: LLM-Perf Leaderboard



- https://huggingface.co/spaces/optimum/llm-perf-leaderboard

Like TinyLlama

TinyLlama/**TinyLlama-1.1B-intermediate-step-1431k-3T** like 132

Text Generation | Transformers | PyTorch | Safetensors | cerebras/SlimPajama-627B | bigcode/starcoderdata | English | llama | Inference Endpoints | text-generation-inference

License: apache-2.0

Model card | Files and versions | Community 8 | Train | Deploy | Use in Transformers

Edit model card

TinyLlama-1.1B

https://github.com/jzhang38/TinyLlama

The TinyLlama project aims to **pretrain** a **1.1B Llama model on 3 trillion tokens**. With some proper optimization, we can achieve this within a span of "just" 90 days using 16 A100-40G GPUs 🚀🚀. The training has started on 2023-09-01.

We adopted exactly the same architecture and tokenizer as Llama 2. This means TinyLlama can be plugged and played in many open-source projects built upon Llama. Besides, TinyLlama is compact with only 1.1B parameters. This compactness allows it to cater to a multitude of applications demanding a restricted computation and memory footprint.

This Collection

This collection contains all checkpoints after the 1T fix. Branch name indicates the step and number of tokens seen.

Eval

Model	Pretrain Tokens	HellaSwag	Obqa	WinoGrande	ARC_c	ARC_e	boolq	piqa	avg
Pythia-1.0B	300B	47.16	31.40	53.43	27.05	48.99	60.83	69.21	48.30
TinyLlama-1.1B-intermediate-step-50K-104b	103B	43.50	29.80	53.28	24.32	44.91	59.66	67.30	46.11
TinyLlama-1.1B-intermediate-step-240k-503b	503B	49.56	31.40	55.80	26.54	48.32	56.91	69.42	48.28

Downloads last month
91,999

Safetensors | Model size 1.1B params | Tensor type F32

Inference API

Text Generation | Examples

My name is Mariama, my favorite

Compute ⌘+Enter 0.3

This model can be loaded on Inference API (serverless).

Model not loaded yet

JSON Output | Maximize

Datasets used to train TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T

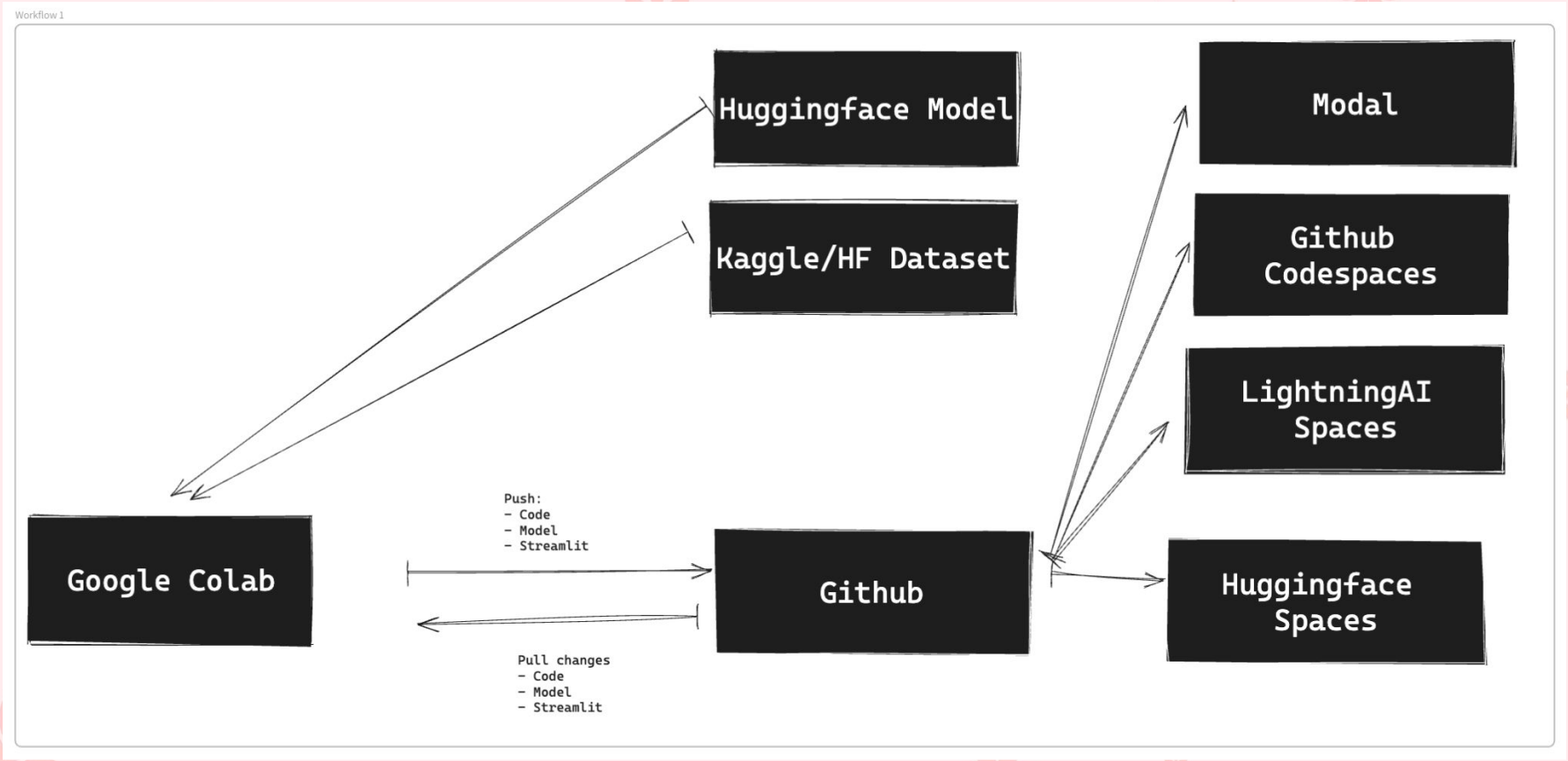
bigcode/starcoderdata
Updated May 16, 2023 · 4.85k · 316

cerebras/SlimPajama-627B
Updated Jul 7, 2023 · 1.77k · 336

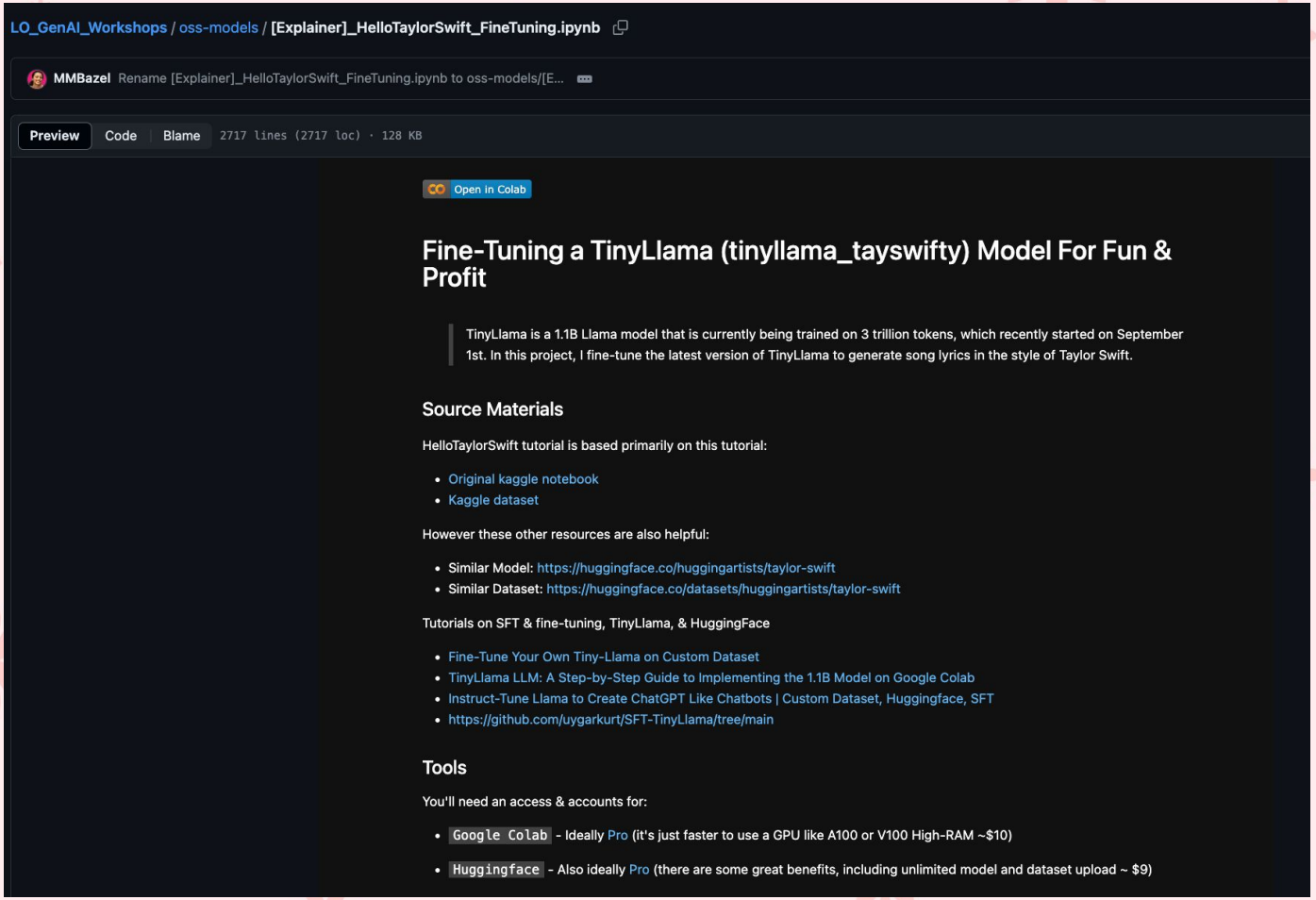
Spaces using TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T 5

eduagarcia/open_pt_llm_leaderboard | mhenrichsen/Axolotl_Launcher

Easiest Way To Interact & Develop With An LLM



Ex 1: Fine-Tuning a Taylor Swift TinyLlama Model



LO_GenAI_Workshops / oss-models / [Explainer]_HelloTaylorSwift_FineTuning.ipynb

MMBazel Rename [Explainer]_HelloTaylorSwift_FineTuning.ipynb to oss-models/[E...

Preview Code Blame 2717 lines (2717 loc) · 128 KB

Open in Colab

Fine-Tuning a TinyLlama (tinyllama_tayswifty) Model For Fun & Profit

> TinyLlama is a 1.1B Llama model that is currently being trained on 3 trillion tokens, which recently started on September 1st. In this project, I fine-tune the latest version of TinyLlama to generate song lyrics in the style of Taylor Swift.

Source Materials

HelloTaylorSwift tutorial is based primarily on this tutorial:

- Original kaggle notebook
- Kaggle dataset

However these other resources are also helpful:

- Similar Model: https://huggingface.co/huggingartists/taylor-swift
- Similar Dataset: https://huggingface.co/datasets/huggingartists/taylor-swift

Tutorials on SFT & fine-tuning, TinyLlama, & HuggingFace

- Fine-Tune Your Own Tiny-Llama on Custom Dataset
- TinyLlama LLM: A Step-by-Step Guide to Implementing the 1.1B Model on Google Colab
- Instruct-Tune Llama to Create ChatGPT Like Chatbots | Custom Dataset, Huggingface, SFT
- https://github.com/uygarkurt/SFT-TinyLlama/tree/main

Tools

You'll need an access & accounts for:

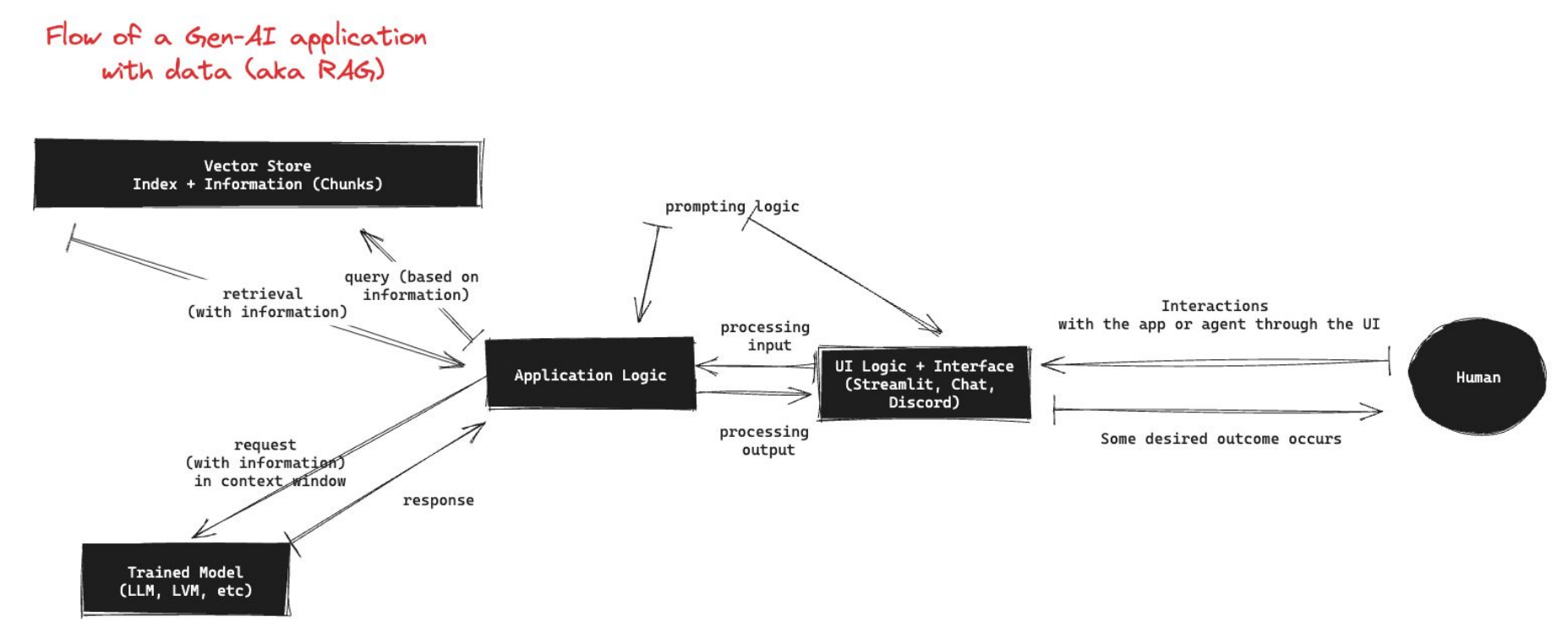
- `Google Colab` - Ideally Pro (it's just faster to use a GPU like A100 or V100 High-RAM ~$10)
- `Huggingface` - Also ideally Pro (there are some great benefits, including unlimited model and dataset upload ~ $9)

Ex 1: Fine-Tuning a Taylor Swift TinyLlama Model

- [Explainer]_HelloTaylorSwift_FineTuning.ipynb: https://colab.research.google.com/drive/1W3HQDpU2kOJSg2FF8Qts1m110XopiQa4?usp=sharing

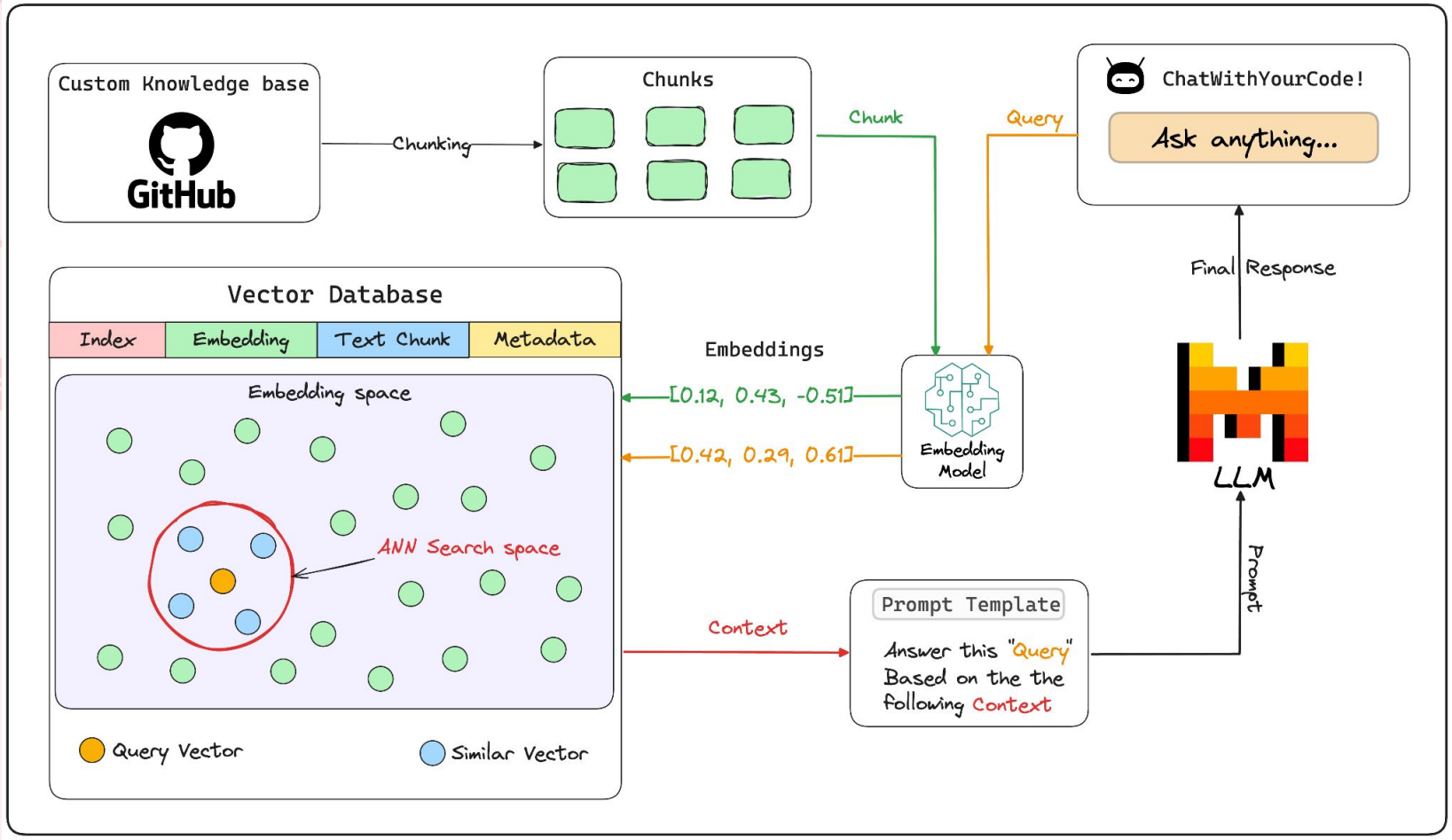
- [Mini]_HelloTaylorSwift_FineTuning.ipynb: https://colab.research.google.com/drive/1hGKgfb0SmFeJDC12dHb4X7FqAu0vt6Oh?usp=sharing

Simplified Flow of RAG



-

Ex: Simple RAG



- Link: https://lightning.ai/lightning-ai/studios/chat-with-your-code-using-rag?section=featured

Ex: Simple RAG

Chat with your code using RAG!

Akshay Pachaar
Published March 1, 2024

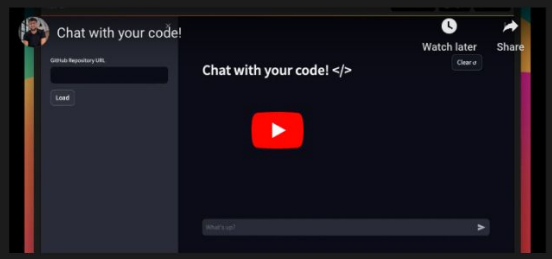
Open in Studio

Overview Files

Chat with your code using RAG!

Transform your interaction with GitHub repositories through a natural language interface. In this studio we are building a **"Chat with your code"** RAG application that simplifies code queries, making coding more intuitive and productive.

And here's a quick demo of what we're building:



Chat with your code demo!

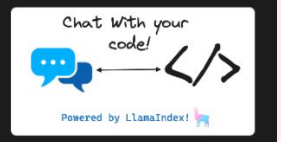
Try it yourself

At this point I would encourage you to take it for a spin by clicking on the Open Template & follow the steps below:

The studio launches in a new tab, you can always come back here to understand how things work!

Run main notebook

You can start by running the main.ipynb notebook, which contains the essential code to set up a query engine for interacting with the repository you provide.

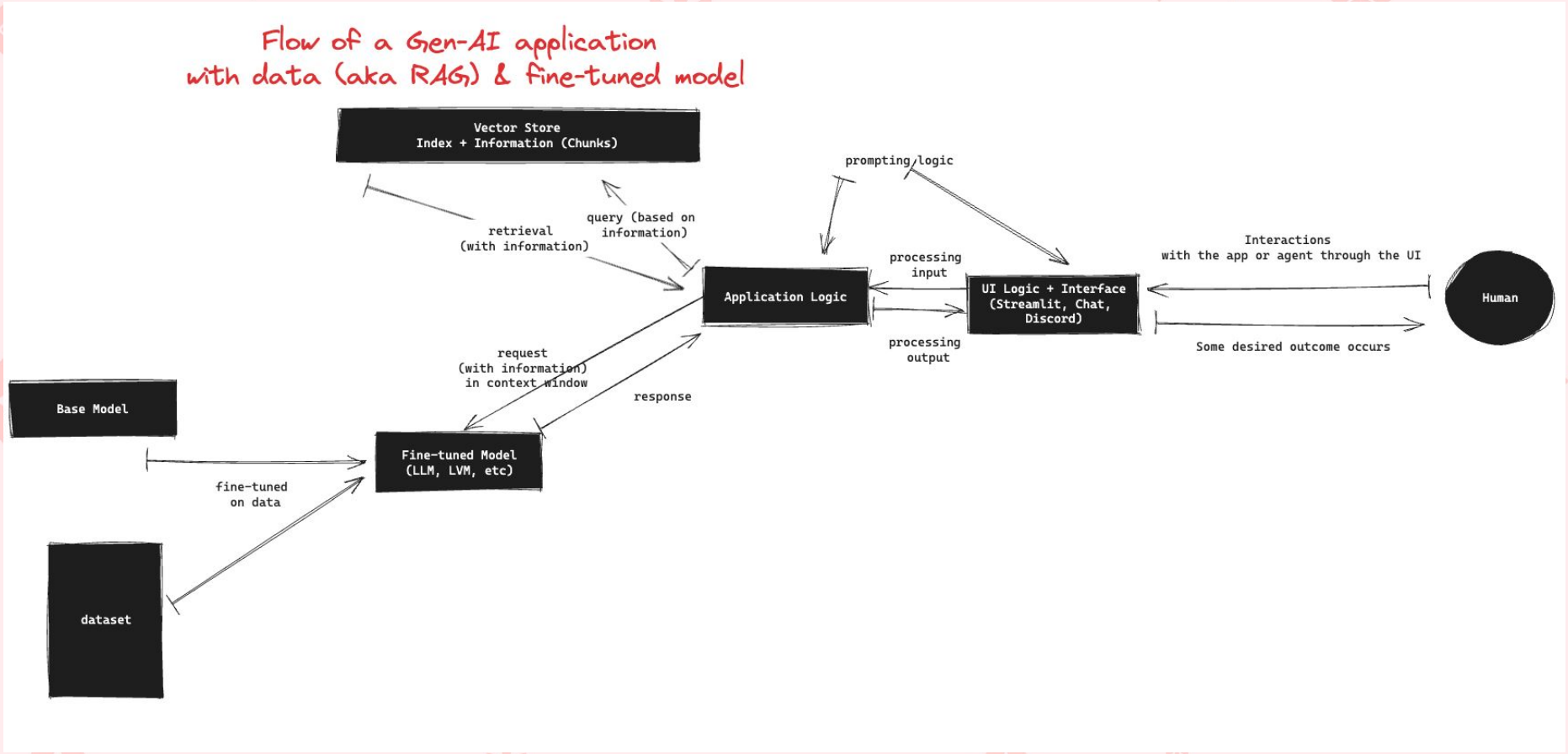


Machine: (1 × A10G) GPU
License: Apache-2.0
Get Studio badge

Text Tutorials Blogs

- Link: https://lightning.ai/lightning-ai/studios/chat-with-your-code-using-rag?section=featured

Simplified Flow of Rag + Fine-Tuned Based Application



-

Prompting vs Fine-Tuning vs RAG

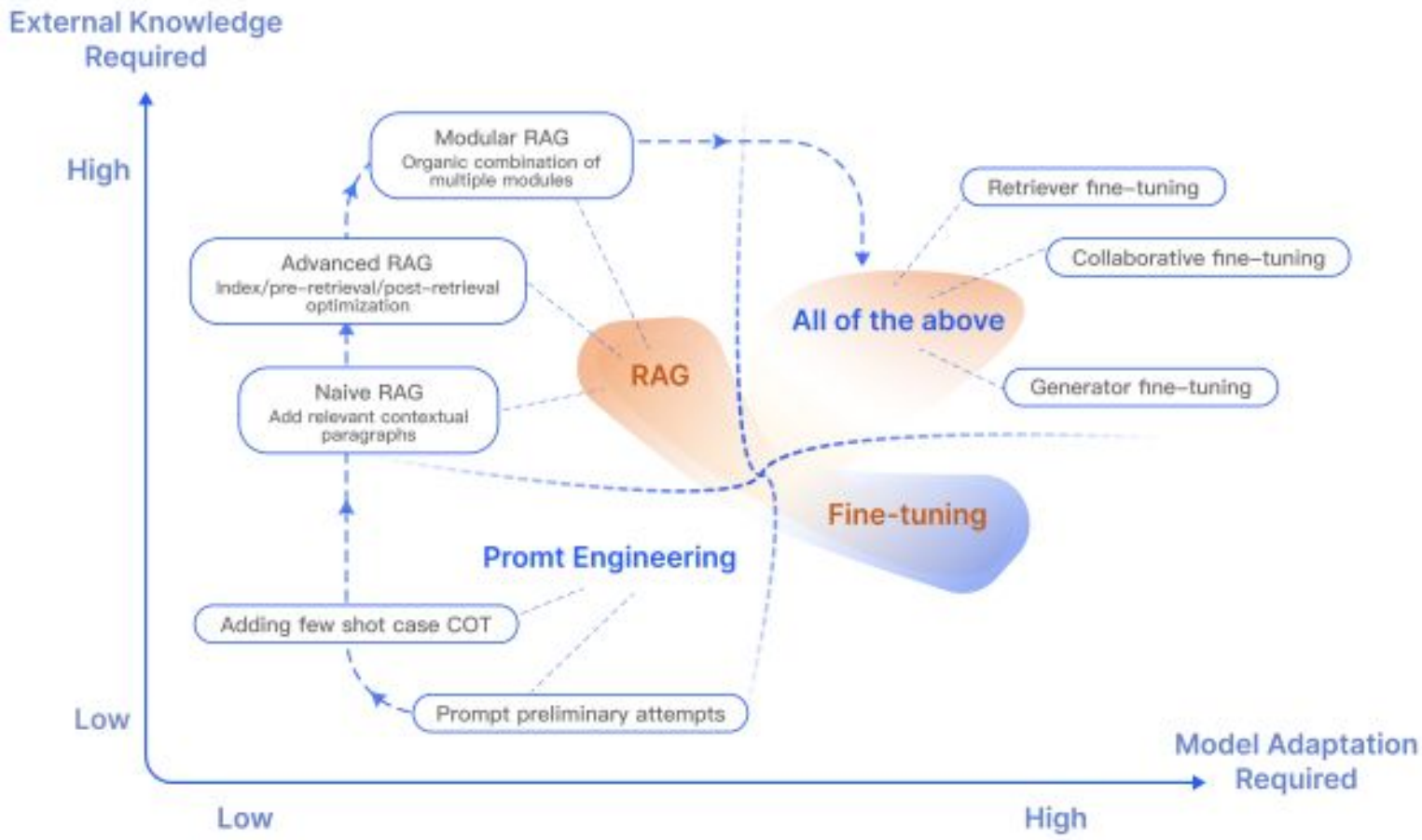


Figure 2: RAG compared with other model optimization methods

- Note: Comparison of RAG versus Fine-Tuning – Source: "Retrieval-Augmented Generation for Large Language Models: A Survey"

Ex 2: RAG With A Vector Database

Chat with your code: RAG with Weaviate and LlamaIndex

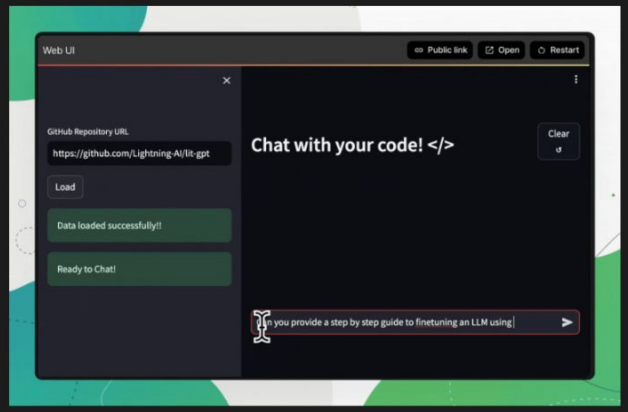
Leonie
Published April 5, 2024

Open in Studio

Overview Files

Chat with your code: RAG with Weaviate and LlamaIndex

This Studio template builds a simple application that lets you chat with code from a GitHub repository. Here's a quick demo of what we're building:

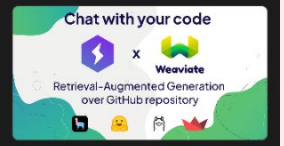


Demo of "Chat with your code" app.

Under the hood, you are building a Retrieval-Augmented Generation (RAG) application using the following components:

- **Embedding model:** BGE embedding model via Hugging Face
- **Vector database:** Weaviate
- **Large Language Model (LLM):** Mistral via Ollama
- **Orchestration framework:** LlamaIndex
- **App framework:** Streamlit

Note: This Studio template is based on the Studio template "Chat with your code using RAG!" by Akshay Pachaar.



Machine: (1 × A10G) GPU
License: Apache-2.0
Get Studio badge
Text Tutorials

Chat with your code: RAG with Weaviate and LlamaIndex
Prerequisites
Set up a Weaviate vector database
Set up environment variables
Build a Retrieval-Augmented Generation Pipeline
1. Setup LLM and embedding model
2. Load files from GitHub repository
3. Prepare Weaviate vector database
4. Setup query engine
5. Setup a simple user interface
Summary

- Link: https://lightning.ai/weaviate/studios/chat-with-your-code-rag-with-weaviate-and-llamaindex

The Ops Part of LLMOps

KEY PIECES, LLM OPS DEV

Is it just MLOps + LLMs?

- e.g., GitHub, Conda, UNIX Terminal, VS Code, Jupyter Notebook + OpenAI API

Kind of

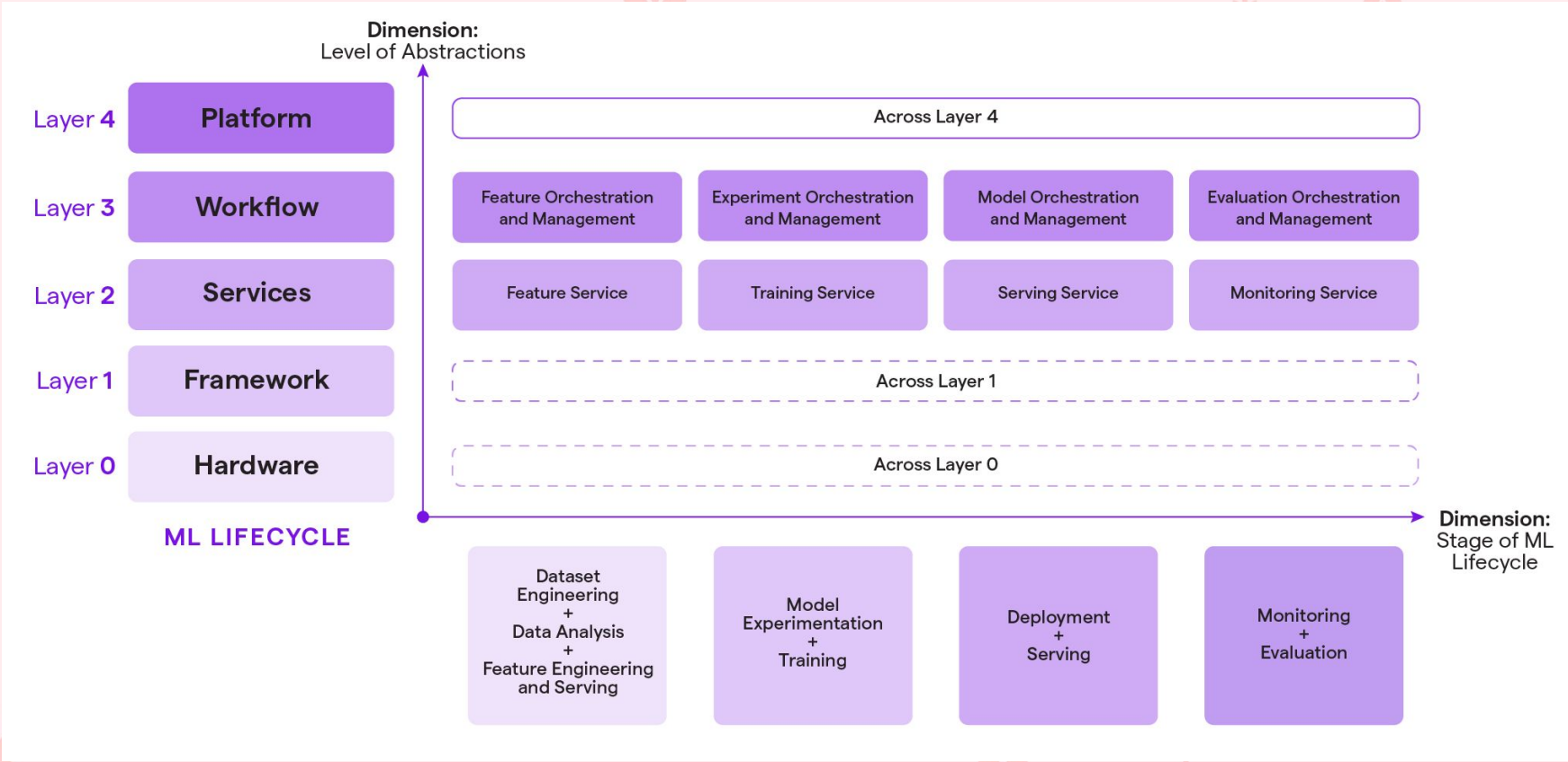
- But also, FastAPI + Docker => HF

And of course

- AWS, GCP, Azure, also Databricks...

- -

MLOps Stack (Pre-GenAI)



- Link: https://www.featureform.com/post/measuring-your-ml-platforms-north-star-metrics

MLOps Stack (Pre-GenAI)

The Four Wars of the AI Stack

The Quality Data Wars

Part of the AI Stack Wars

Content platforms defending from content raids

Belligerents	
Journalists	AI researchers
Writers	AI startups
Artists	Synthetic Data research
Leaders and flagbearers	
The New York Times	OpenAI
StackOverflow	Stability AI
Reddit	Microsoft
Getty Images	CommonCrawl
Sarah Silverman	Axel Springer
Greg Rutkowski	X (Elon Musk)
Sarah Andersen	Eleuther AI
George R. R. Martin	LAION / DataComp

The War of the GPU Rich/Poors

Part of the AI Stack Wars

Faster/Cheaper Inference, Finetuning & Training

Belligerents	
GPU Rich Clouds	GPU Poor AI Engineers
GPU Manufacturers	Edge/Local Compute
VC Funding	New Model Research
Leaders and flagbearers	
Nvidia	Modular / MLC
Google	Tinycorp
Microsoft	QLoRA (Tim Dettmers)
Amazon Bedrock	r/LocalLlama
Together.ai	Consistency Models
Fireworks.ai	Apple / MLX
Anyscale	RWKV / RetNet
Replicate	Mamba / StripedHyena

The Multimodality War

Part of the AI Stack Wars

Specialist models vs Everything models

Belligerents	
Text to Image Startups	OpenAI
Text to Audio Startups	Google Deepmind
Other modalities	
Leaders and flagbearers	
Midjourney	OpenAI ChatGPT/API
Playground.ai	Google Gemini
Lexica.art	
Eleven Labs	
Suno.ai	
Pika Labs	
HeyGen	
Assembly.ai	

The RAG/Ops War

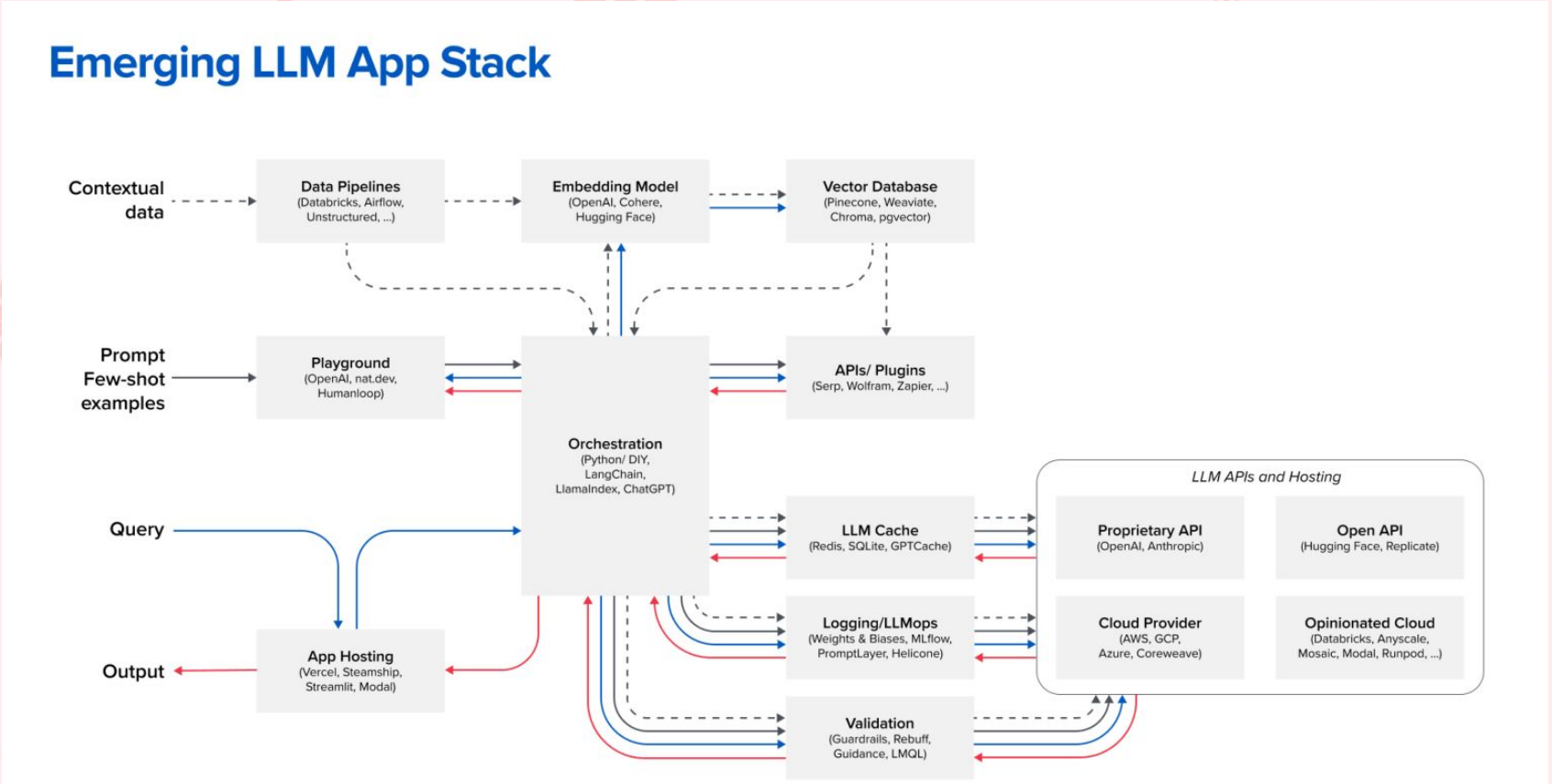
Part of the AI Stack Wars

Databases, Frameworks, vs Dev Tooling

Belligerents		
General DBs	LLM Frameworks	MLOps
Vector DBs	LLM Platforms	PromptOps
Search/Recsys	AIEF Standards	LLMOps
Leaders and flagbearers		
Postgres (pgvector)	LangChain	W&B
MongoDB Vector	LlamaIndex	Humanloop
Cassandra Vector	Guardrails	Arize.ai
Elasticsearch	Microsoft Autogen	Helicone
Redis Vector	Flowise	PromptLayer
Pinecone	BuildShip	Vellum
Chroma	LangServe	LangSmith
Qdrant	Agent Protocol	Gantry

- Link: Latent.Space: https://www.latent.space/p/dec-2023-audio

Gen-AI Stack???


Emerging LLM App Stack
Contextual data
Data Pipelines
(Databricks, Airflow, Unstructured, ...)
Embedding Model
(OpenAI, Cohere, Hugging Face)
Vector Database
(Pinecone, Weaviate, Chroma, pgvector)
Prompt
Few-shot examples
Playground
(OpenAI, nat.dev, Humanloop)
APIs/ Plugins
(Serp, Wolfram, Zapier, ...)
Orchestration
(Python/ DIY, LangChain, LlamaIndex, ChatGPT)
Query
LLM Cache
(Redis, SQLite, GPTCache)
LLM APIs and Hosting
Proprietary API
(OpenAI, Anthropic)
Open API
(Hugging Face, Replicate)
Logging/LLMops
(Weights & Biases, MLflow, PromptLayer, Helicone)
Cloud Provider
(AWS, GCP, Azure, Coreweave)
Opinionated Cloud
(Databricks, Anyscale, Mosaic, Modal, Runpod, ...)
Output
App Hosting
(Vercel, Steamship, Streamlit, Modal)
Validation
(Guardrails, Rebuff, Guidance, LMQL)

Gen-AI Stack???

Data pipelines	Embedding model	Vector database	Playground	Orchestration	APIs/plugins	LLM cache
Databricks	OpenAI	Pinecone	OpenAI	Langchain	Serp	Redis
Airflow	Cohere	Weaviate	nat.dev	LlamaIndex	Wolfram	SQLite
Unstructured	Hugging Face	ChromaDB	Humanloop	ChatGPT	Zapier	GPTCache
		pgvector				

Logging / LLMops	Validation	App hosting	LLM APIs (proprietary)	LLM APIs (open)	Cloud providers	Opinionated clouds
Weights & Biases	Guardrails	Vercel	OpenAI	Hugging Face	AWS	Databricks
MLflow	Rebuff	Steamship	Anthropic	Replicate	GCP	Anyscale
PromptLayer	Microsoft Guidance	Streamlit			Azure	Mosaic
Helicone	LMQL	Modal			CoreWeave	Modal
						RunPod

Have Additional Q's For Me About MLOps?

Find Me On

in/mikikobazeley

BazeleyMikiko

MikiBazeleyTheMLOps Engineer

Click Long Link