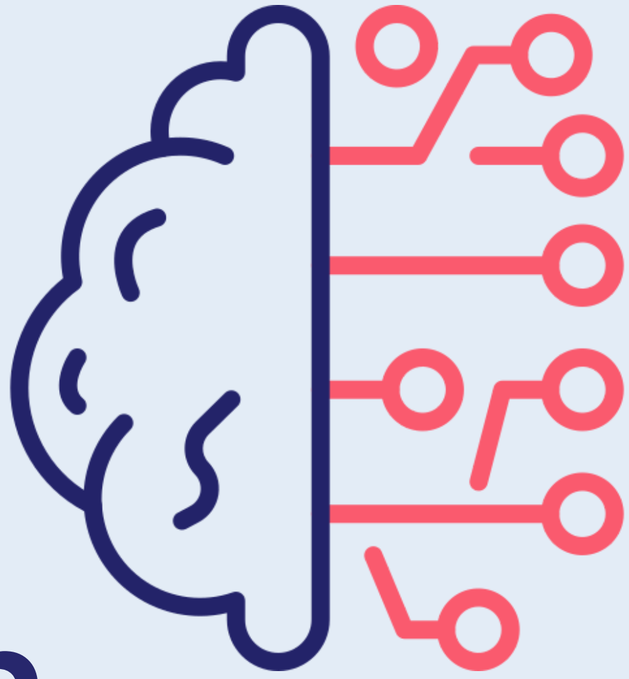
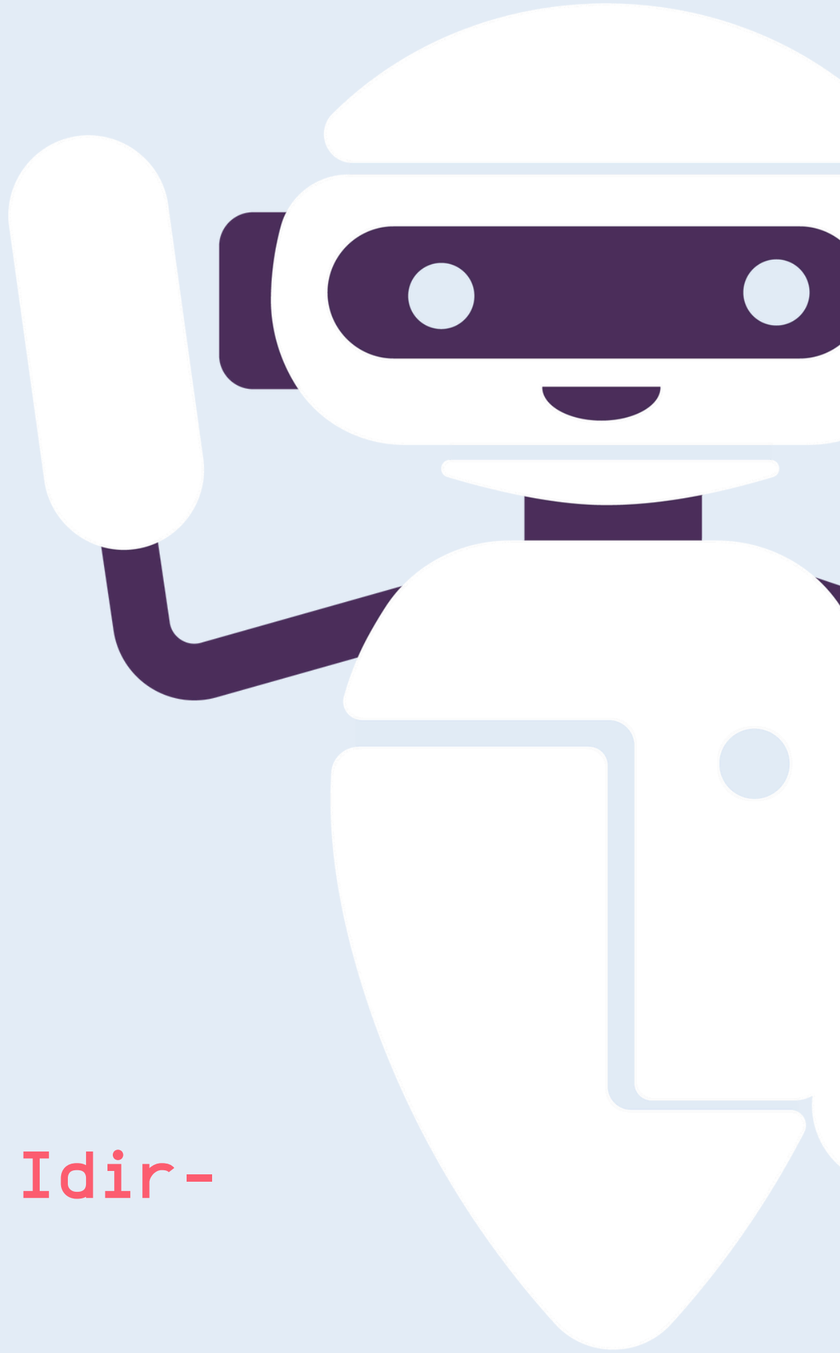




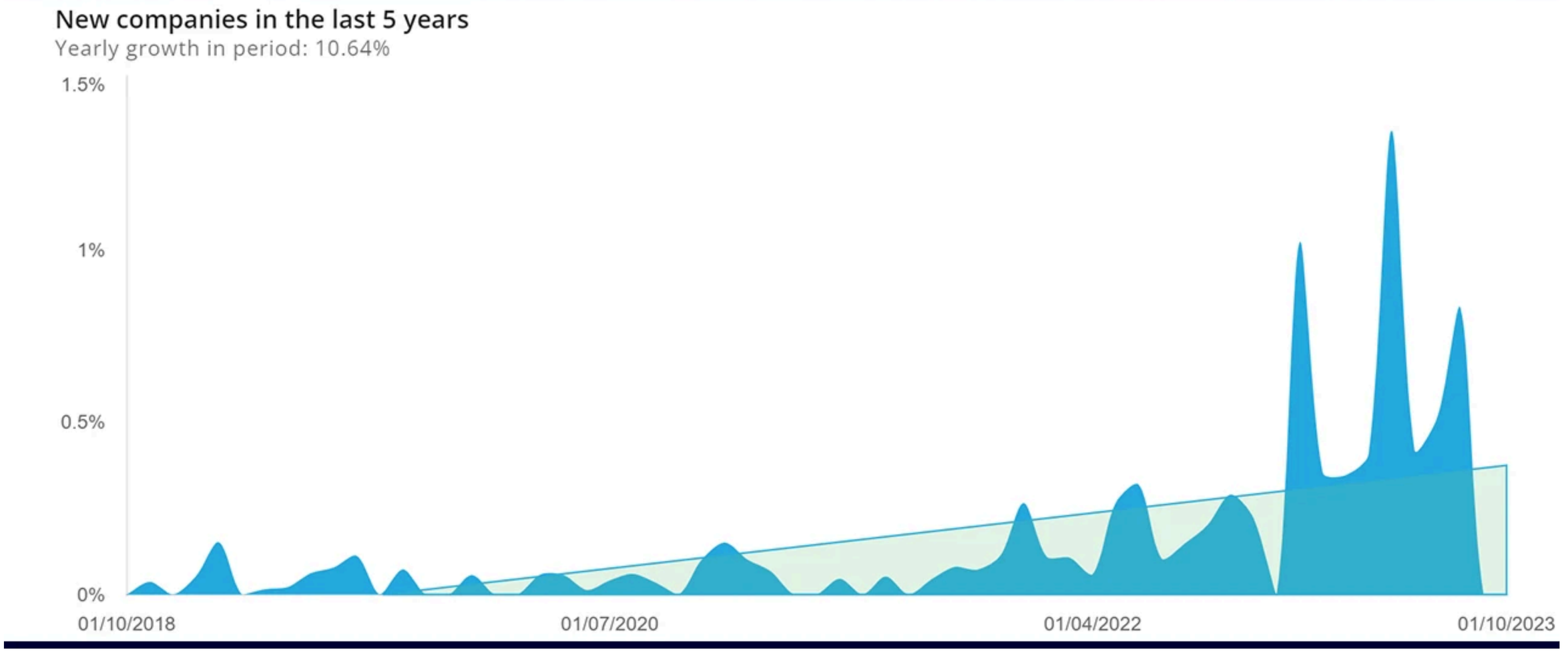
LLMs: from an engineering POV.

-With: AIT SAID Azzedine Idir-

About Me

- Computer engineer from ESI
- Graduation project: Anomaly detection in 5G networks using LLMs
- Actually in Ile-de-France, working on a project involving LLMs
- Passionate about replacing humans with LLMs

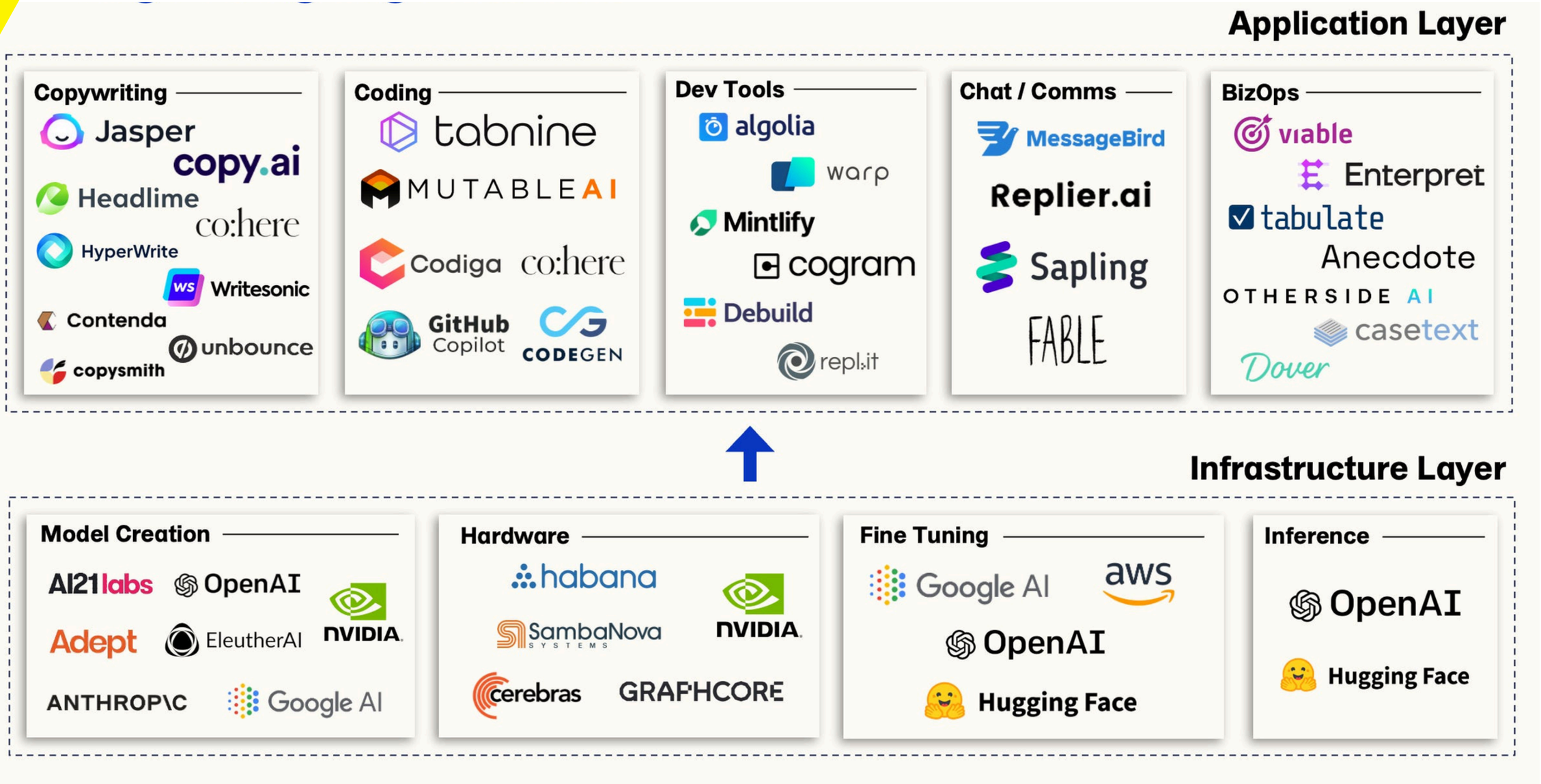
Large Language Models (Impact)



Growth of companies selling only LLM-based products

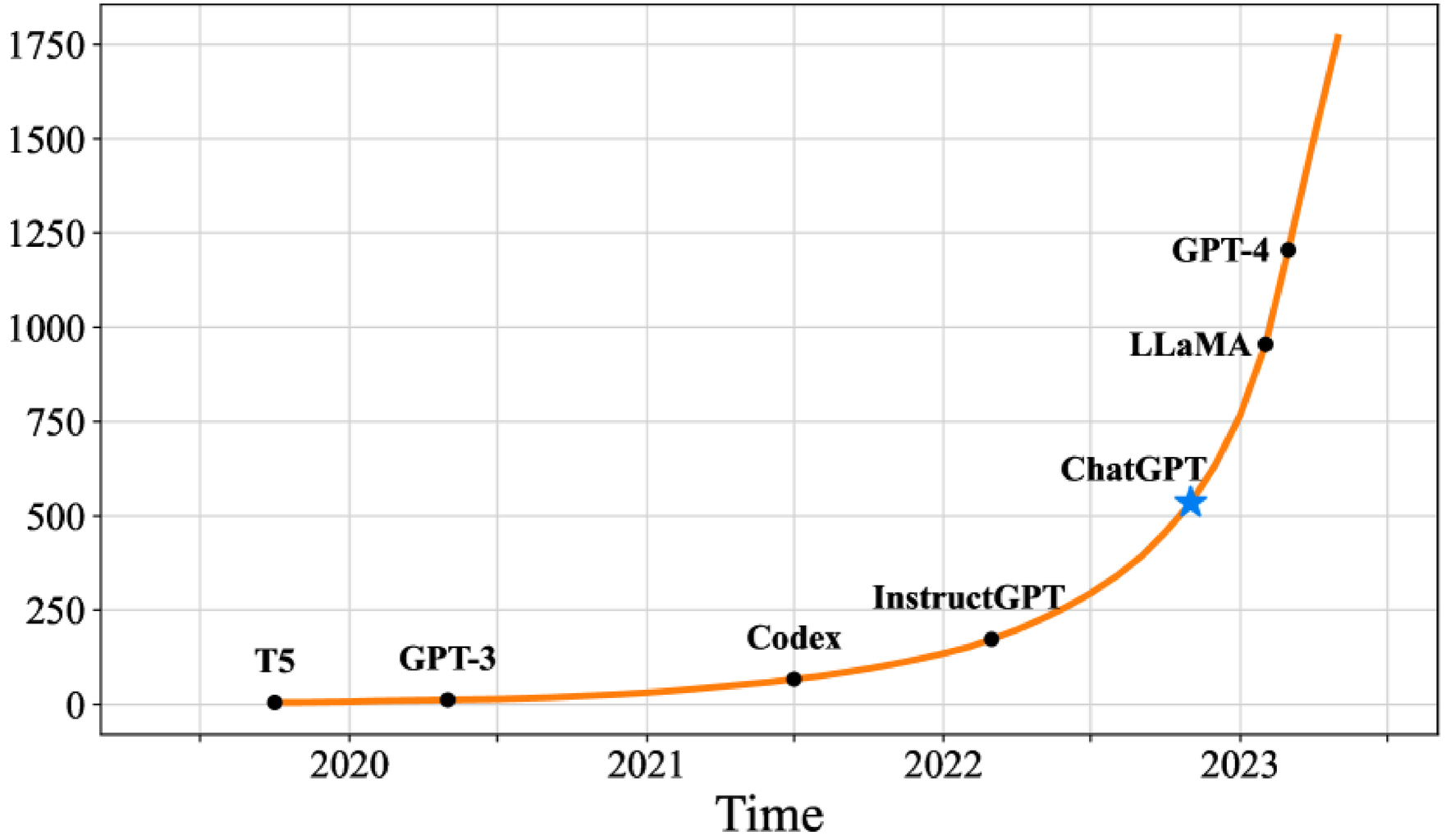
Source: Trendfeedr

Large Language Models (Impact)



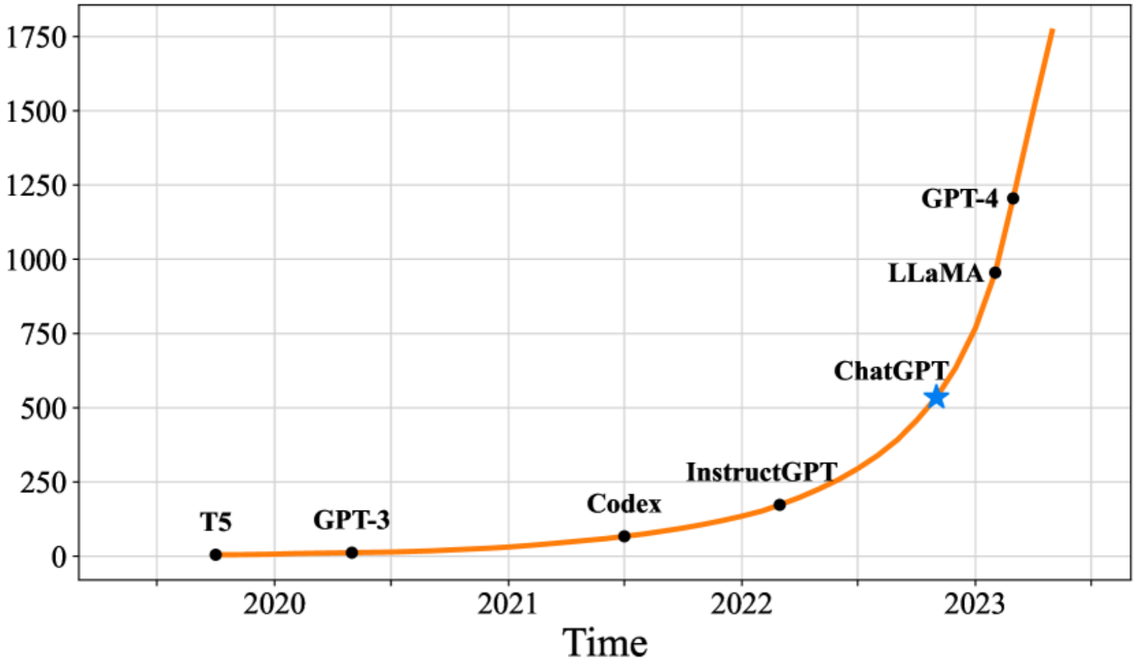
Sample of the LLM Market

Large Language Models (Research)



The trends of the cumulative numbers of arXiv papers that contain the keyphrase “Large Language Models”

Large Language Models (Research)



The trends of the cumulative numbers of arXiv papers that contain the keyphrase “Large Language Models”

the average number of published arXiv papers that contain “large language model” in title or abstract goes **from 0.40 per day to 8.58 per day**

-Source: A Survey of Large Language Models Wayne Xin Zhao et al

Large Language Models (Research)

A Survey of Large Language Models

Wayne Xin Zhao, Kun Zhou*, Junyi Li*, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie and Ji-Rong Wen

Abstract—Ever since the Turing Test was proposed in the 1950s, humans have explored the mastering of language intelligence by machine. Language is essentially a complex, intricate system of human expressions governed by grammatical rules. It poses a significant challenge to develop capable artificial intelligence (AI) algorithms for comprehending and grasping a language. As a major approach, *language modeling* has been widely studied for language understanding and generation in the past two decades, evolving from statistical language models to neural language models. Recently, pre-trained language models (PLMs) have been proposed by pre-training Transformer models over large-scale corpora, showing strong capabilities in solving various natural language processing (NLP) tasks. Since the researchers have found that model scaling can lead to an improved model capacity, they further investigate the scaling effect by increasing the parameter scale to an even larger size. Interestingly, when the parameter scale exceeds a certain level, these enlarged language models not only achieve a significant performance improvement, but also exhibit some special abilities (*e.g.*, in-context learning) that are not present in small-scale language models (*e.g.*, BERT). To discriminate the language models in different parameter scales, the research community has coined the term *large language models (LLM)* for the PLMs of significant size (*e.g.*, containing tens or hundreds of billions of parameters). Recently, the research on LLMs has been largely advanced by both academia and industry, and a remarkable progress is the launch of ChatGPT (a powerful AI chatbot developed based on LLMs), which has attracted widespread attention from society. The technical evolution of LLMs has been making an important impact on the entire AI community, which would revolutionize the way how we develop and use AI algorithms. Considering this rapid technical progress, in this survey, we review the recent advances of LLMs by introducing the background, key findings, and mainstream techniques. In particular, we focus on four major aspects of LLMs, namely pre-training, adaptation tuning, utilization, and capacity evaluation. Furthermore, we also summarize the available resources for developing LLMs and discuss the remaining issues for future directions. This survey provides an up-to-date review of the literature on LLMs, which can be a useful resource for both researchers and engineers.

Index Terms—Large Language Models; Emergent Abilities; Adaptation Tuning; Utilization; Alignment; Capacity Evaluation

1 Introduction

"The limits of my language mean the limits of my world."
—Ludwig Wittgenstein

Language is a prominent ability in human beings to express and communicate, which develops in early childhood and evolves over a lifetime [3, 4]. Machines, however, cannot naturally grasp the abilities of understanding and communicating in the form of human language, unless equipped with powerful artificial intelligence (AI) algorithms. It has been a longstanding research challenge to achieve this goal, to enable machines to read, write, and communicate like humans [5].

Technically, *language modeling (LM)* is one of the major approaches to advancing language intelligence of machines. In general, LM aims to model the generative likelihood of word sequences, so as to predict the probabilities of future (or missing) tokens. The research of LM has received extensive attention in the literature, which can be divided into four major development stages:

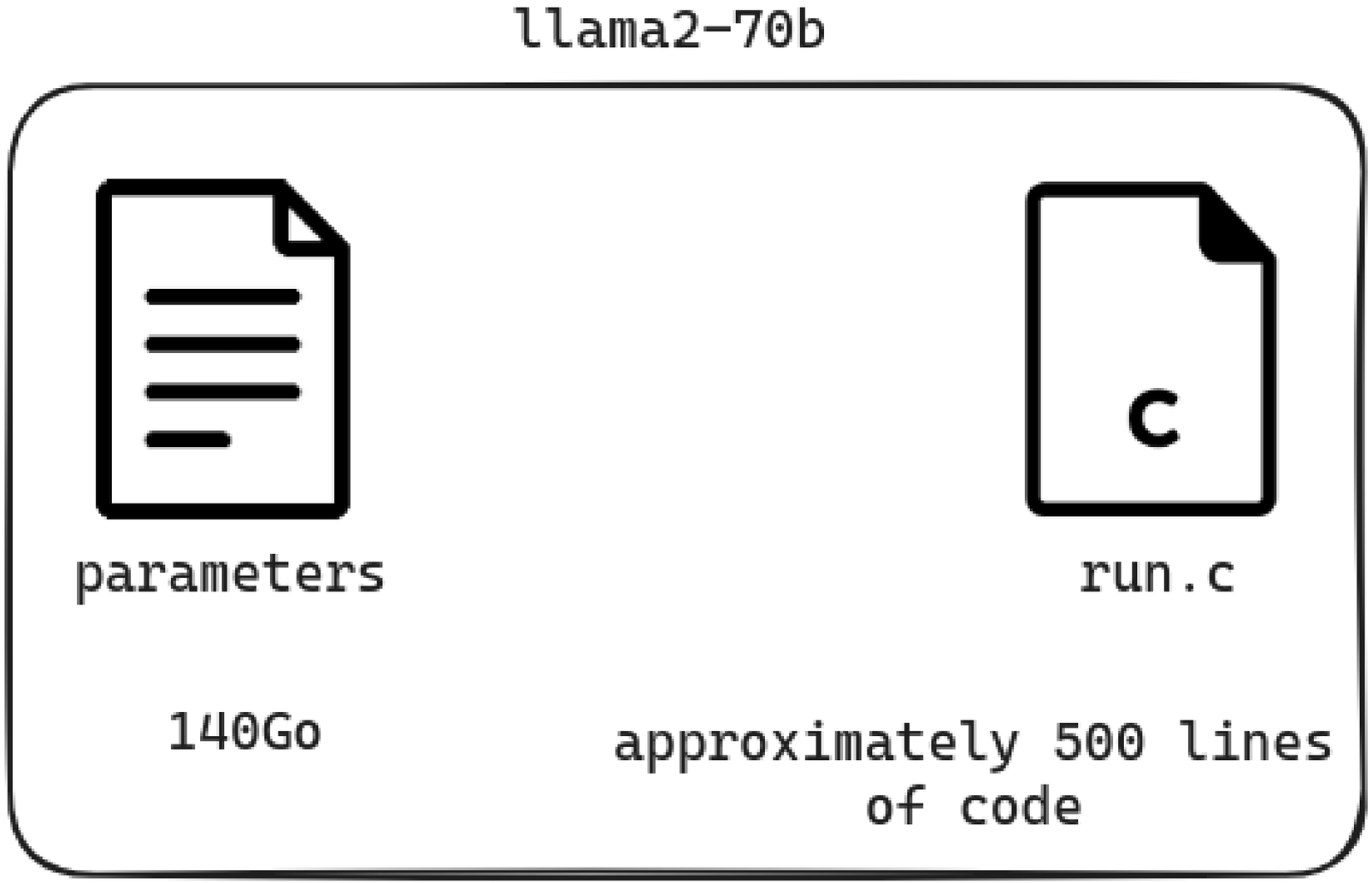
• *Statistical language models (SLM).* SLMs [6–9] are developed based on *statistical learning* methods that rose in the 1990s. The basic idea is to build the word prediction model based on the Markov assumption, *e.g.*, predicting the next word based on the most recent context. The SLMs with a fixed context length n are also called n-gram language models, *e.g.*, bigram and trigram language models. SLMs have been widely applied to enhance task performance in information retrieval (IR) [10, 11] and natural language processing (NLP) [12–14]. However, they often suffer from the curse of dimensionality: it is difficult to accurately estimate high-order language models since an exponential number of transition probabilities need to be estimated. Thus, specially designed smoothing strategies such as back-off estimation [15] and Good–Turing estimation [16] have been introduced to alleviate the data sparsity problem.

• *Neural language models (NLM).* NLMs [1, 17, 18] characterize the probability of word sequences by neural networks, *e.g.*, multi-layer perceptron (MLP) and recurrent neural networks (RNNs). As a remarkable contribution, the work in [1] introduced the concept of *distributed representation* of words and built the word prediction function conditioned on the aggregated context features (*i.e.*, the distributed word vectors). By extending the idea of learning effective features for text data, a general neural network approach was developed to build a unified, end-to-end solution for

• *Version: v14 (major update on September 25, 2024).*
• *GitHub link: https://github.com/RUCAIBox/LLMSurvey*
• *Chinese book link: lmbook-zh.github.io*
• ** K. Zhou and J. Li contribute equally to this work.*
• *The authors are mainly with Gaoling School of Artificial Intelligence and School of Information, Renmin University of China, Beijing, China; Jian-Yun Nie is with DIRO, Université de Montréal, Canada.*
Contact e-mail: batmanfly@gmail.com

[A Survey of Large Language Models](#)
[Wayne Xin Zhao et al](#)

Large language models

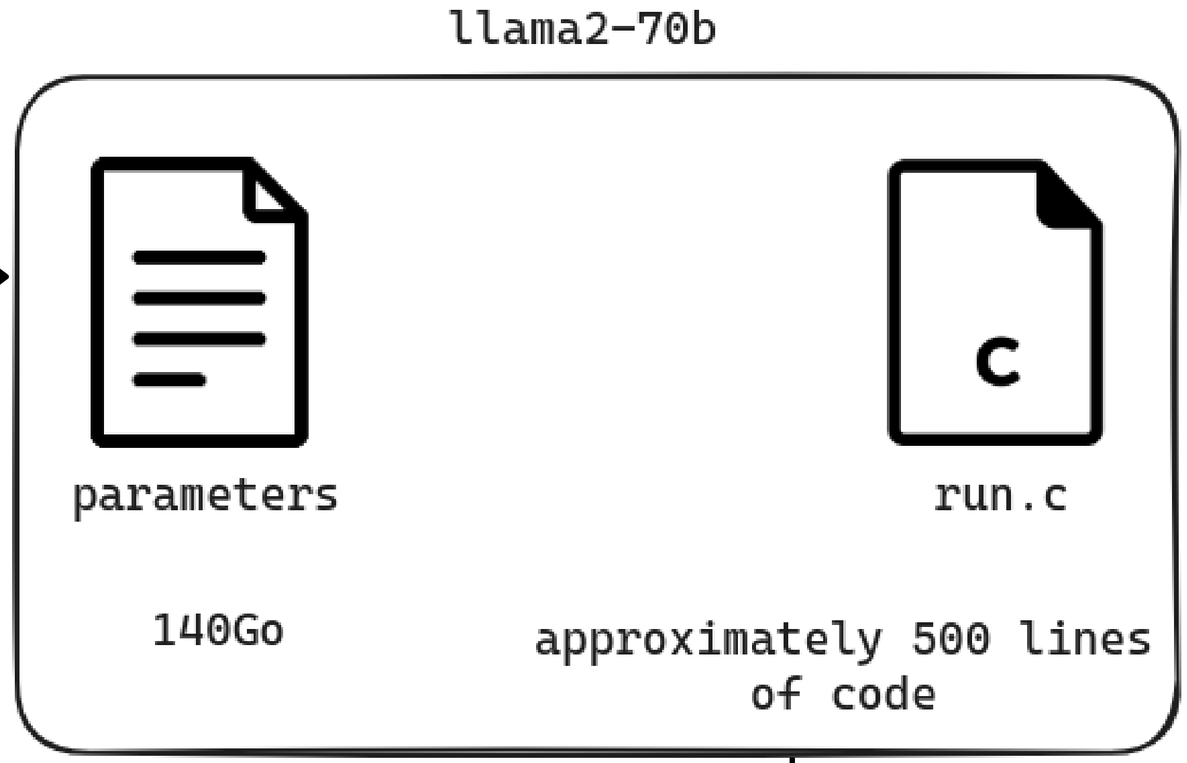


A very simplified scheme of an LLM

Large language models

What’s an LLM ?

Prompt

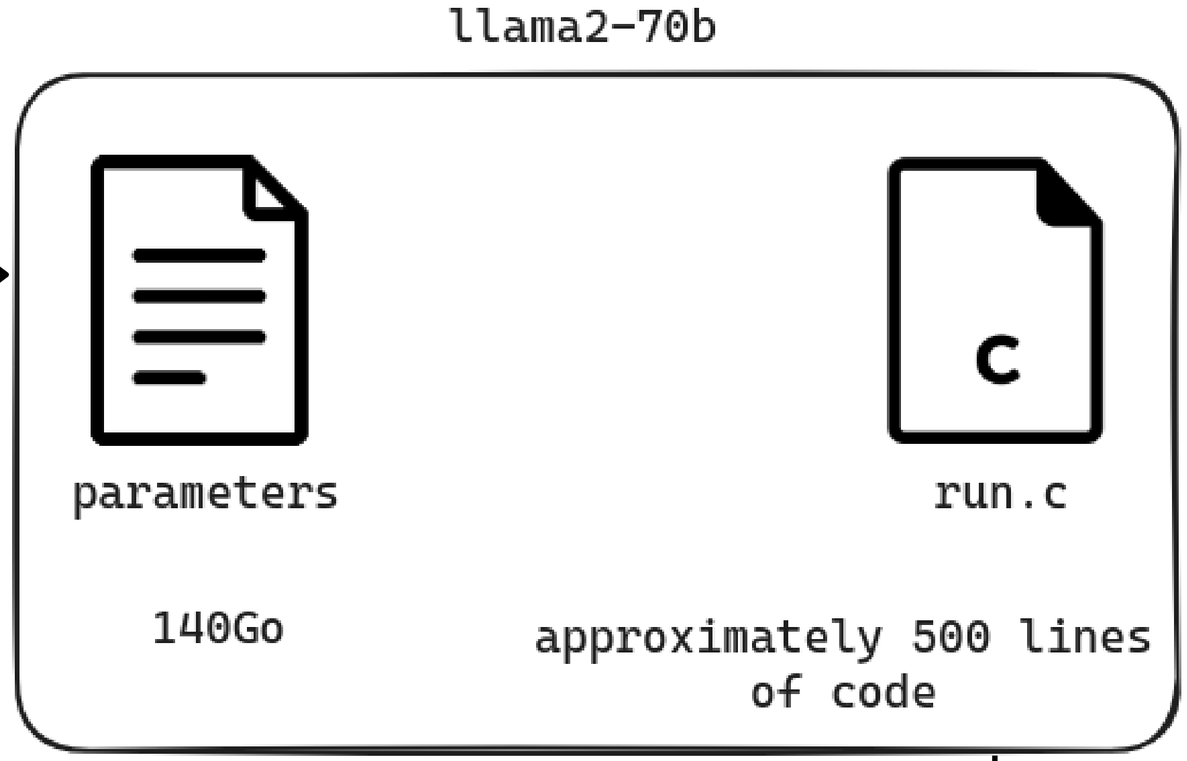


Response

A large language model (LLM) is a computational model capable of language generation or other natural language processing tasks.

Large language models

Answer the following question briefly and in a markup style:
Question: What’s an LLM ?
Answer:

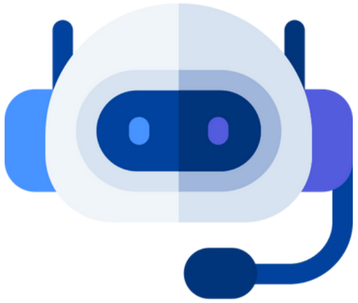


Better Prompt

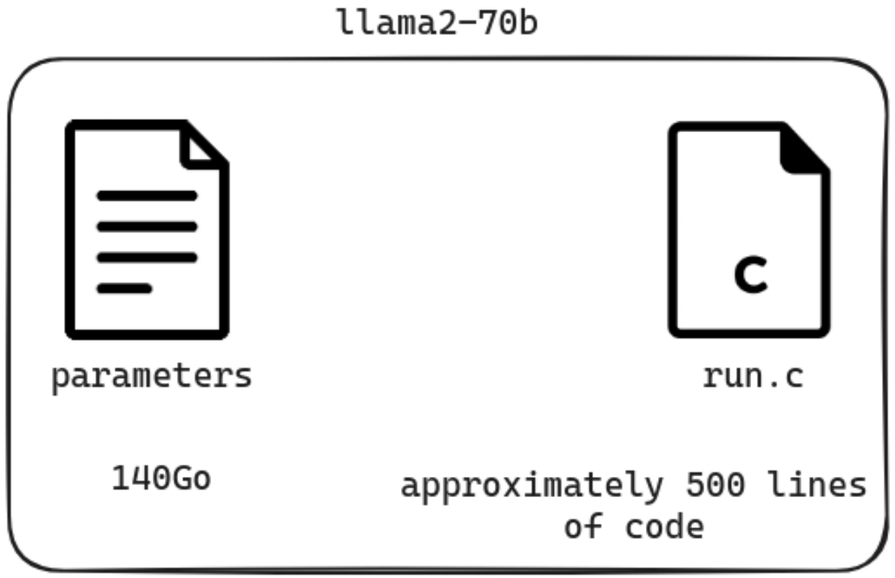
Better Response

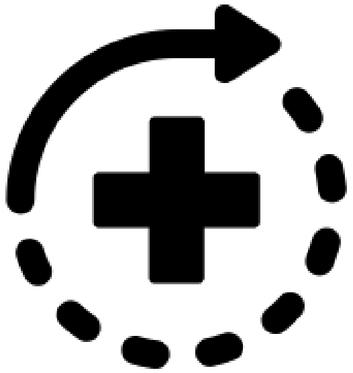
Question: What’s an LLM?
Answer: A Large Language Model (LLM) is an artificial intelligence model designed to understand and generate human-like text based on vast amounts of data.

Large language models



Chatbots



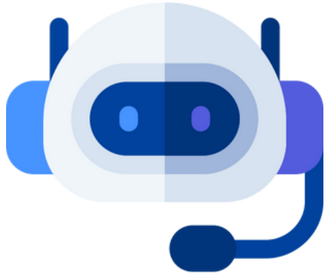


Self-healing systems

Automating routine tasks

Decision making

Large language models



Chatbots

Automating routine tasks

Decision making

Self-healing systems

Reasoning

The ability to combine internal and external knowledge in order to generate a desired text (perform a task)

In context learning

The ability to learn from the input context and generate desired text, performing a task that was unknown before

An engineering POV ?

What we will see ✅	What we will avoid ❌
LLM use cases,	**LLM Training**
LLM example applications	**LLM Fine-Tuning**
LLM-based applications design	**LLM Quantization details**
LLMOps	**/**

Famous LLM use cases

How big tech companies are using LLMs ?

Famous LLM use cases

How big tech companies are using LLMs ?

+1 bonus quick use case

Famous LLM use cases (1)

Source: Accelerating incident response using generative AI (21 April 2024)

Problem = Writing security and privacy incident reports is time and effort-consuming

Famous LLM use cases (1)

Source: Accelerating incident response using generative AI (21 April 2024)

Problem = Writing security and privacy incident reports is time and effort-consuming

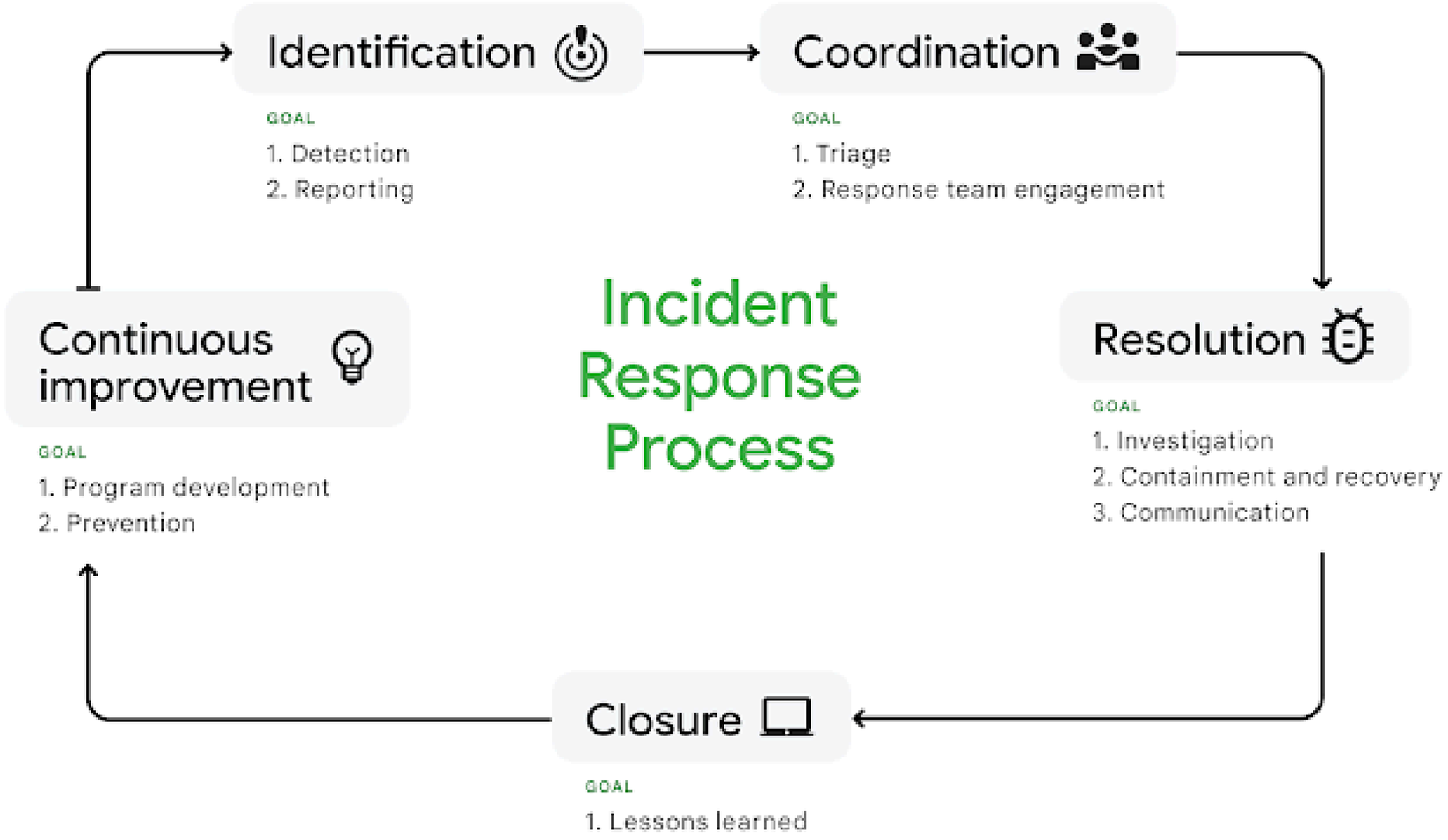
A writing task

Needs specific knowledge

Needs understanding and reasoning

Famous LLM use cases (1)

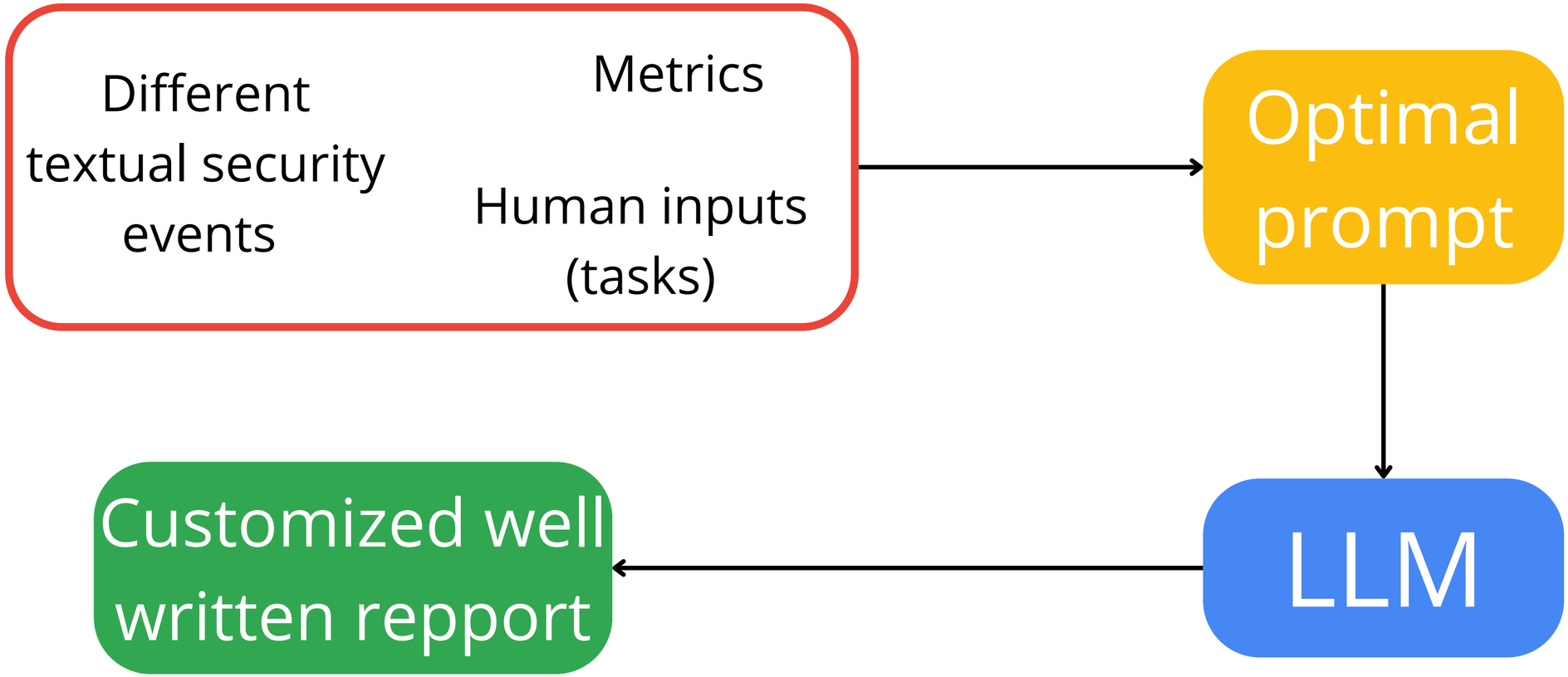
Source: Accelerating incident response using generative AI (21 April 2024)



Google Incident - Response - Process pipeline (a human existing workflow involving 0 LLMs)

Famous LLM use cases (1)

Source: Accelerating incident response using generative AI (21 April 2024)



Famous LLM use cases (1)

Source: Accelerating incident response using generative AI (21 April 2024)

Prompt engineering

Results : time consumption is reduced by 51%
stakeholders are satisfied
experts that were writing reports are doing other tasks

Famous LLM use cases (2)

Source: Applying Generative AI for CVE Analysis at an Enterprise Scale
(May 23, 2024)

Problem = Analyzing and remediating Common Vulnerabilities and Exposures (CVEs) at enterprise scale is challenging because of the volume and complexity of the data

An analysis task

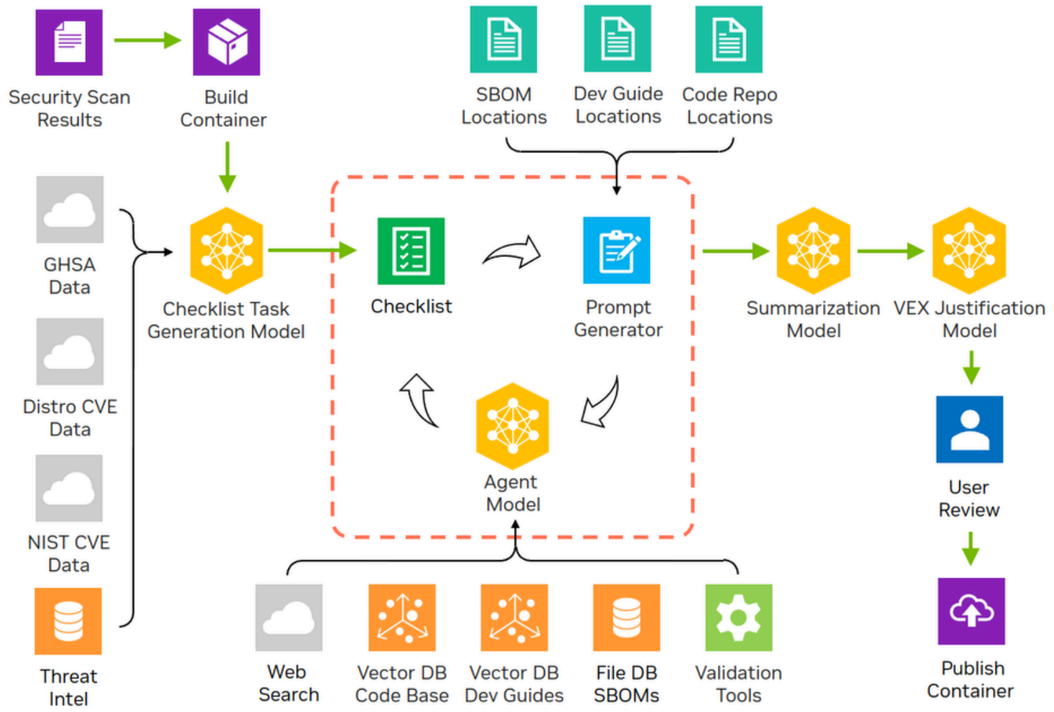
Needs specific knowledge

Needs understanding and reasoning

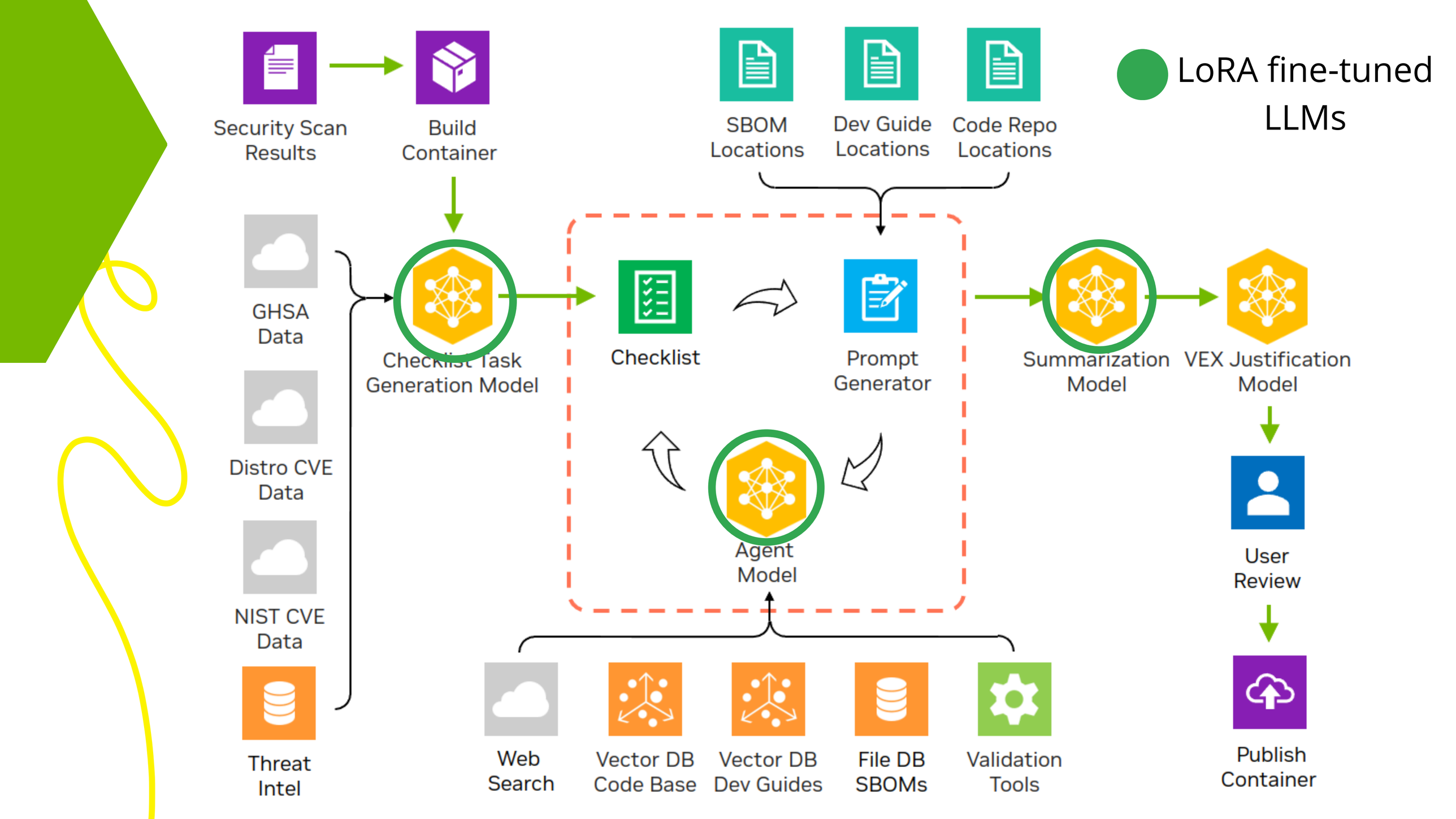
Famous LLM use cases (2)

Source: Applying Generative AI for CVE Analysis at an Enterprise Scale

(May 23, 2024)



Agent Morpheus solution design


LoRA fine-tuned LLMs
Security Scan Results
Build Container
SBOM Locations
Dev Guide Locations
Code Repo Locations
GHSA Data
Distro CVE Data
NIST CVE Data
Threat Intel
Checklist Task Generation Model
Checklist
Prompt Generator
Agent Model
Summarization Model
VEX Justification Model
User Review
Publish Container
Web Search
Vector DB Code Base
Vector DB Dev Guides
File DB SBOMs
Validation Tools

Famous LLM use cases (2)

Source: Applying Generative AI for CVE Analysis at an Enterprise Scale
(May 23, 2024)

Key techniques used:

- **Fine-tuning (Why ?)**
- **Retrieval augmented generation**
- **Agent modeling (Why ?)**

Famous LLM use cases (2)

Source: Applying Generative AI for CVE Analysis at an Enterprise Scale
(May 23, 2024)

Key techniques used:

- **Fine-tuning (Why ?)**
- **Retrieval augmented generation**
- **Agent modeling (Why ?)**

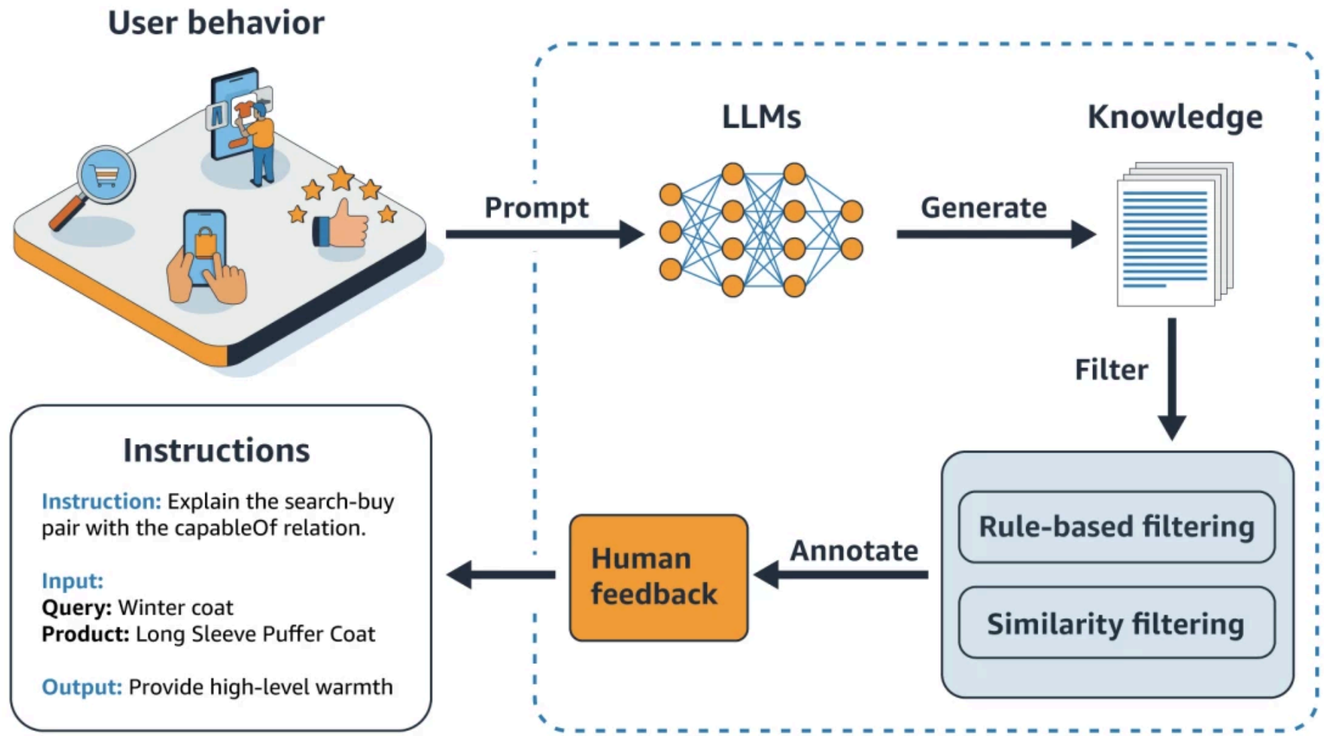
Results:

- **9.3x speedup**
- **Reduced analyst workload**
- **Improved efficiency by minimising Human Intervention**

Famous LLM use cases (3)

Source: Building commonsense knowledge graphs to aid product recommendation

(May 10, 2024)



Comprehensive summary

Solution Type	Need
Fine-tuning	Required when the task demands deep understanding and critical knowledge specific to a domain or topic.
Prompt Engineering	Used when tasks are straightforward and do not require external data or real-time information gathering.
RAG (Retrieval-Augmented Generation)	Necessary when real-time or new data is needed during the generation process.
Agents	Applied for automating workflows that involve multiple, distinct tasks or processes running independently.

Comprehensive summary

Solution Type	Difficulty	Beginner-Friendliness	Time to Set Up	Popularity
Fine-tuning	4	2	4	3
Prompt Engineering	2	5	1	5
RAG	3	3	3	4
Agents	5	2	5	3

Details of an LLM application

.pathway

LANGROID

LangChain

LlamaIndex

tigerlab.

Famous LLM apps building frameworks

Details of an LLM application

.pathway

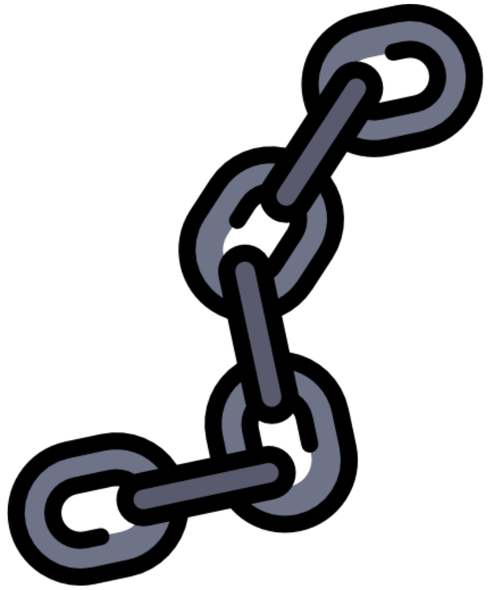
An example of a very simple LLM app built using Pathway package

Details of an LLM application

LangChain

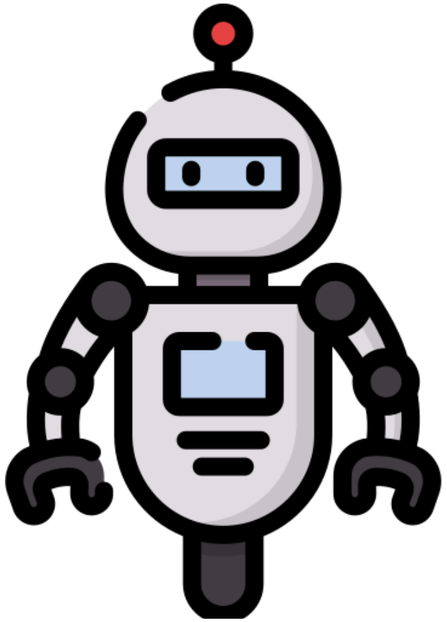
Build a Question/Answering system over SQL data

LLM app design patterns



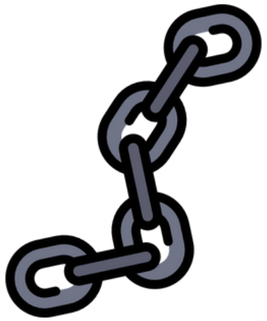
Chains

Tools



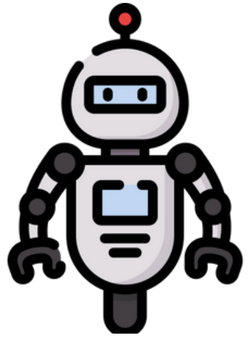
Agents

LLM app design patterns

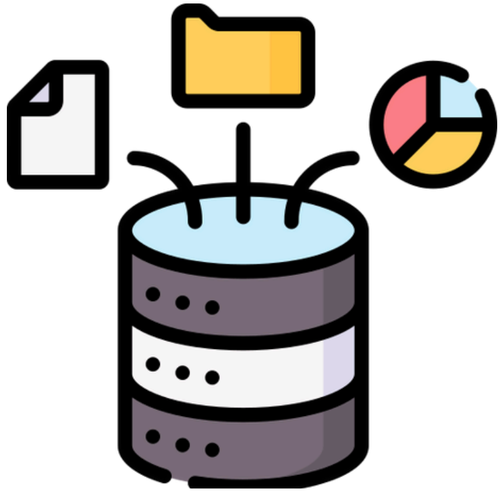


Chains

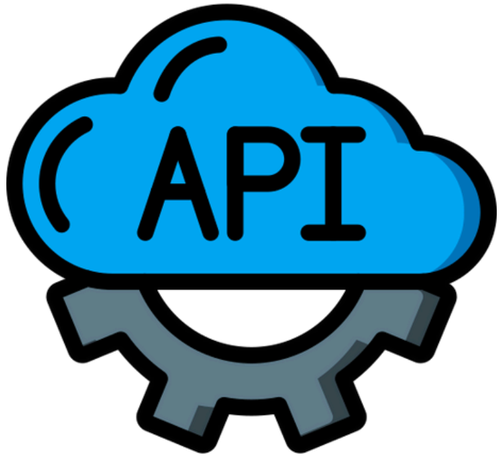
Tools



Agents



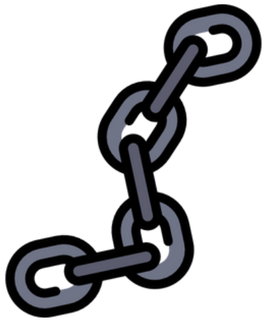
Databases



APIs

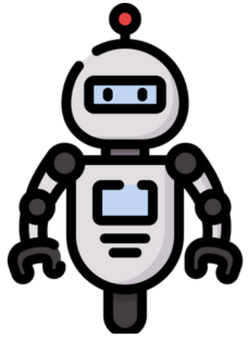
LLM app design patterns

Databases



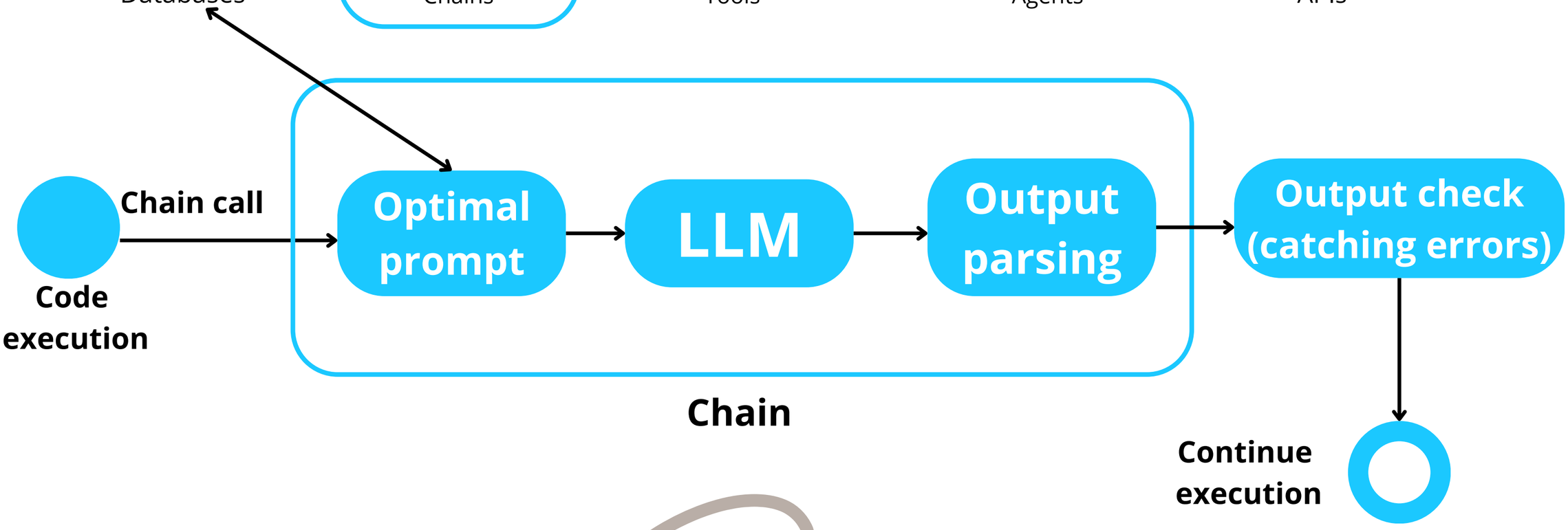
Chains

Tools



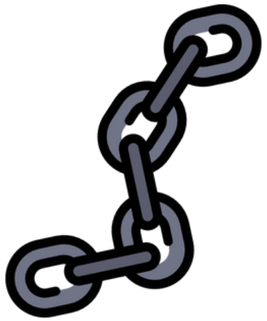
Agents

APIs



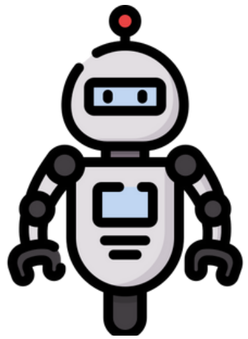
LLM app design patterns

Databases



Chains

Tools



Agents

APIs

Tool call

Code execution

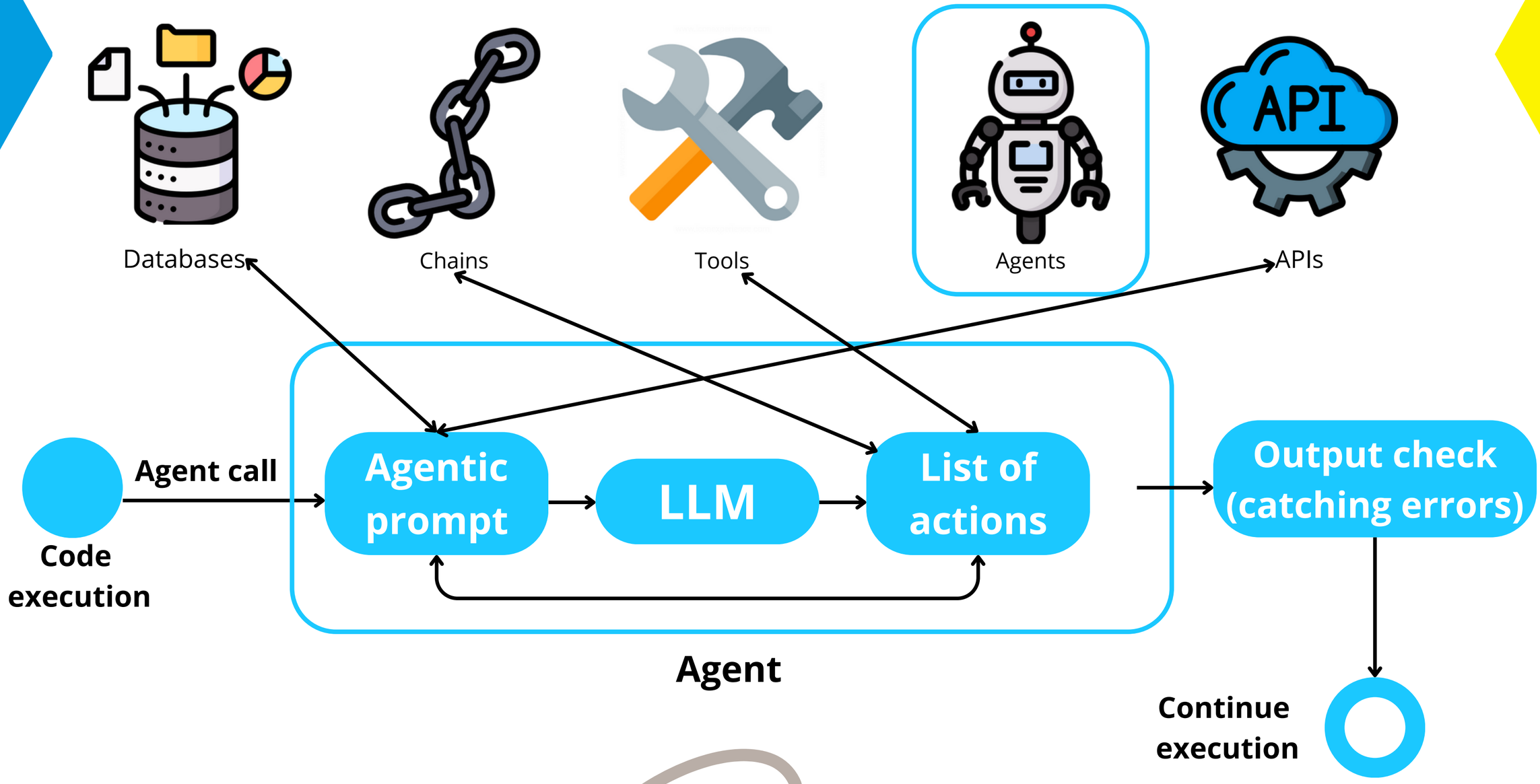
Standard python code executed without using any LLM (or not even an AI)

Output check (catching errors)

Tool

Continue execution

LLM app design patterns



LLM app design patterns

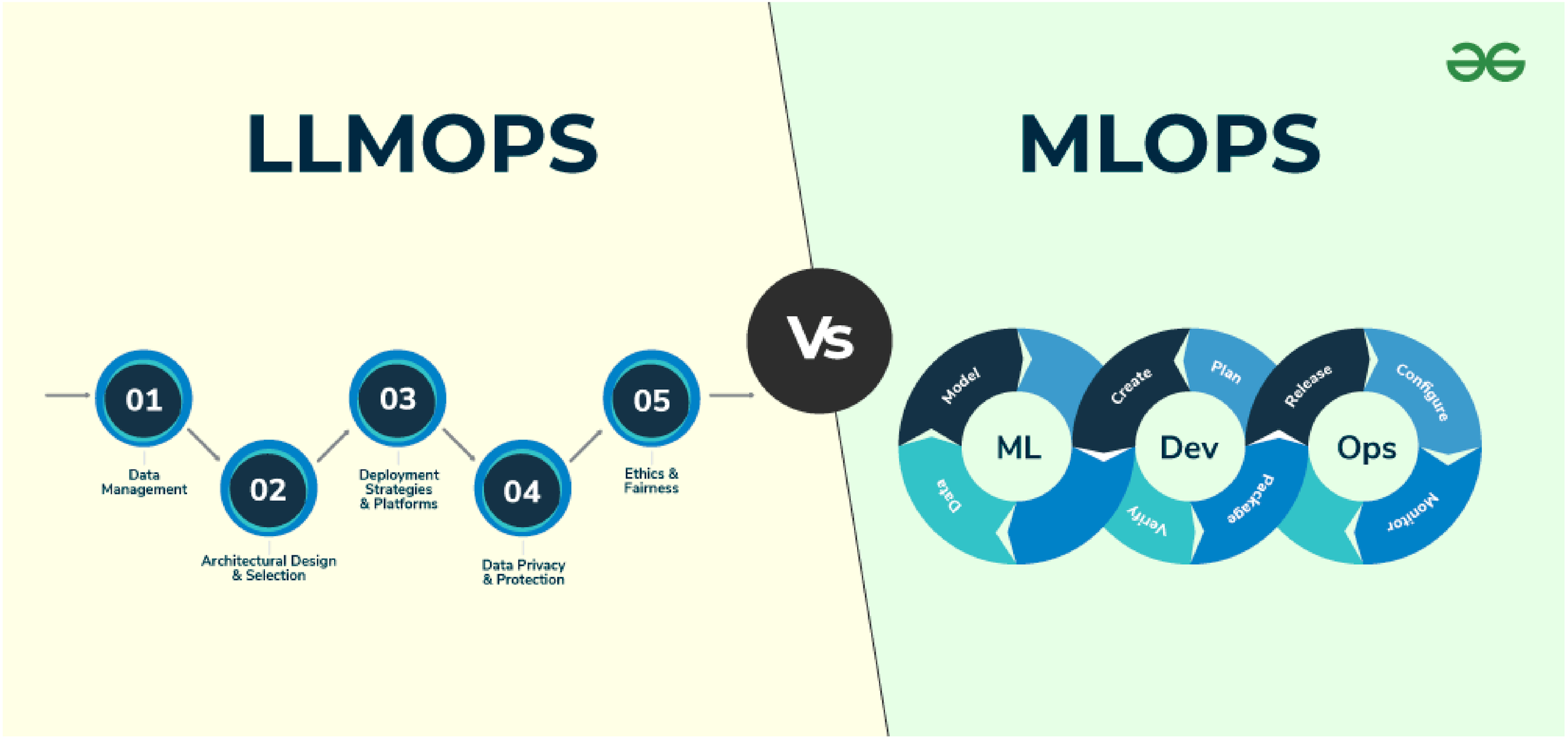
Framework	Key Characteristic (Why Choose It)
LangChain	Offers versatility for creating complex, multi-step LLM applications with support for memory, chains, agents, and prompt engineering. Best for projects needing dynamic workflows and contextual awareness in interactions, such as chatbots or custom pipelines. Supports various LLMs and is highly customizable.
LlamaIndex	Specializes in efficient data indexing and retrieval, making it ideal for RAG (retrieval-augmented generation) applications and handling large volumes of structured and unstructured data. Best for search, question-answering, or enterprise-level document retrieval tasks, with a focus on speed and query optimization.

Working with LLMs from the source:

HuggingFace

Hugging Face

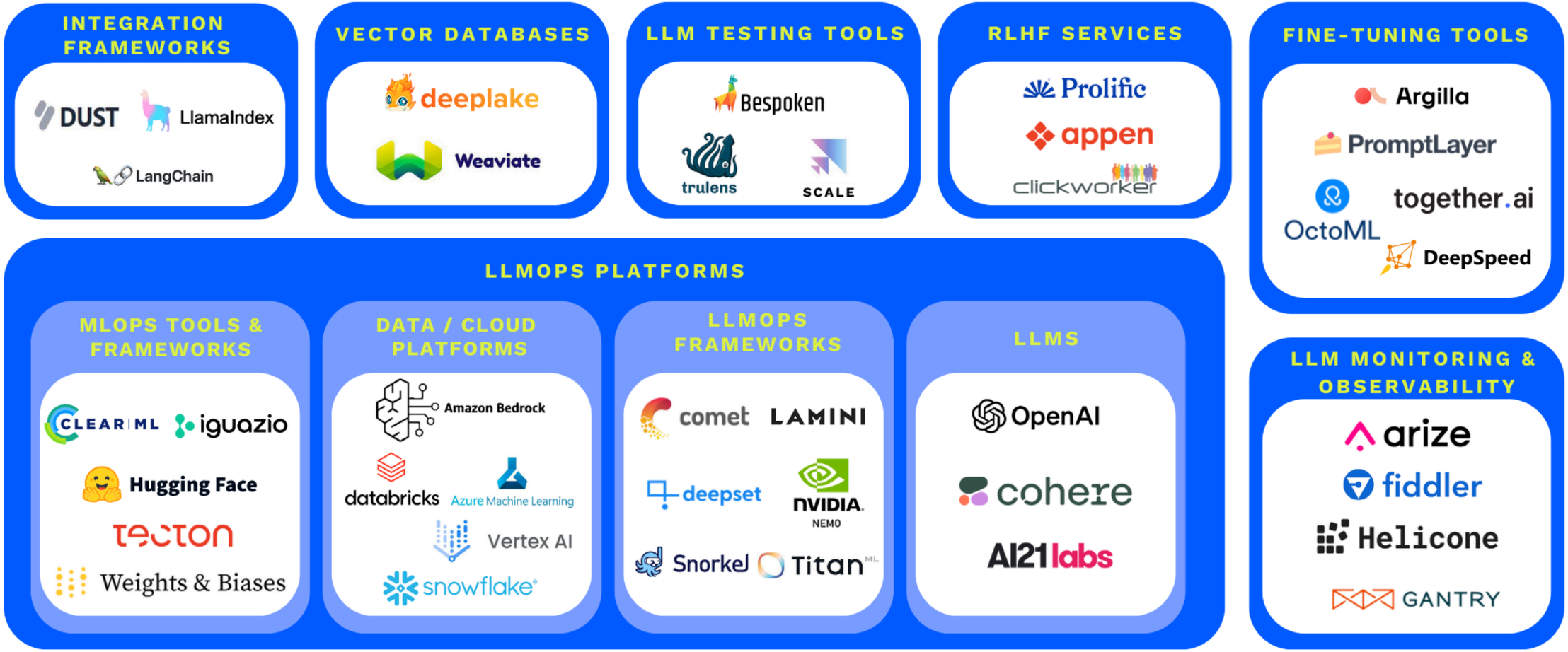
LLMOps



Comparison of MLOps and LLMOps

LLMOps

LLMOPS LANDSCAPE



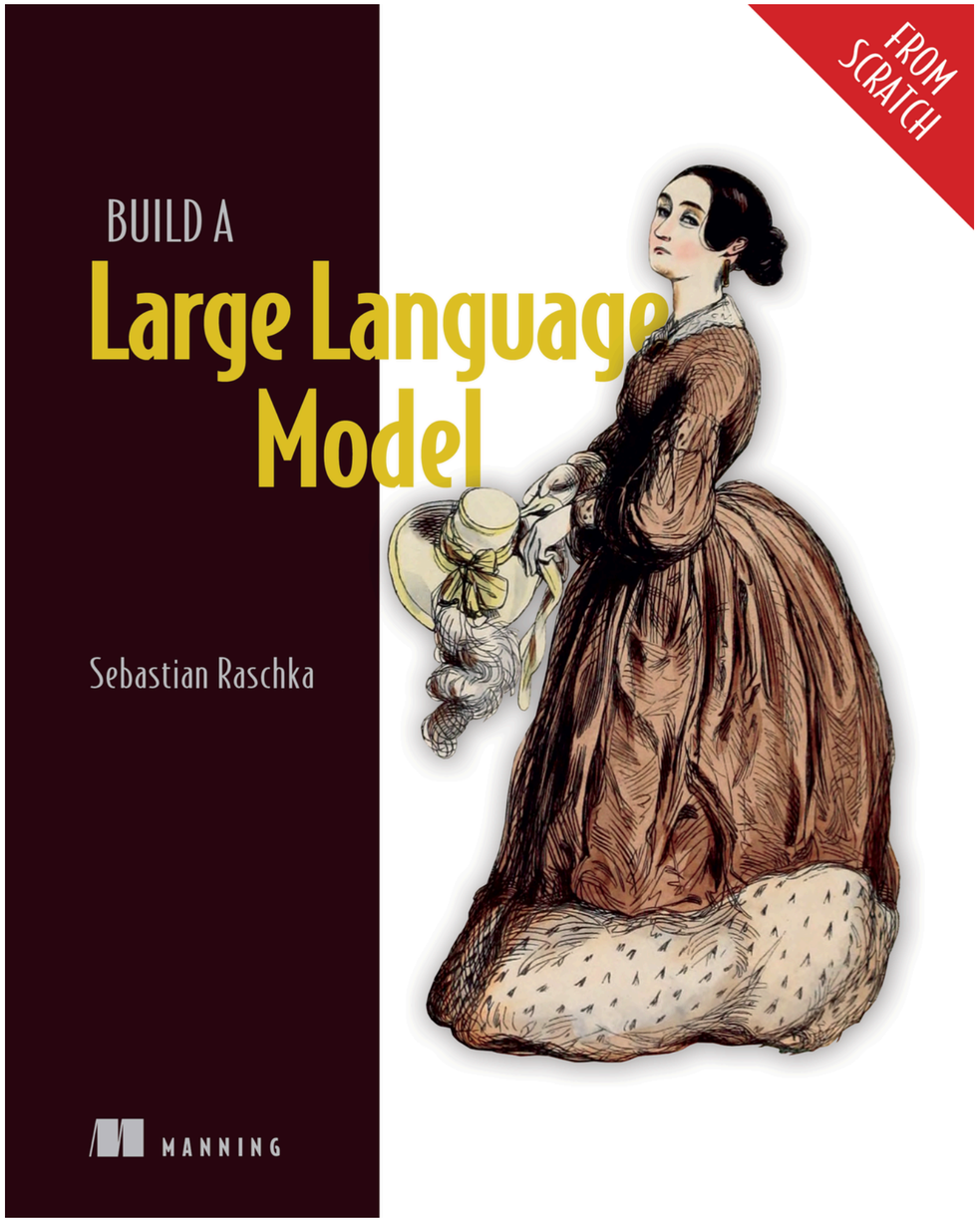
Note: The table shows vendor logos only for simplicity. Each vendor offers different pricing model and capabilities.

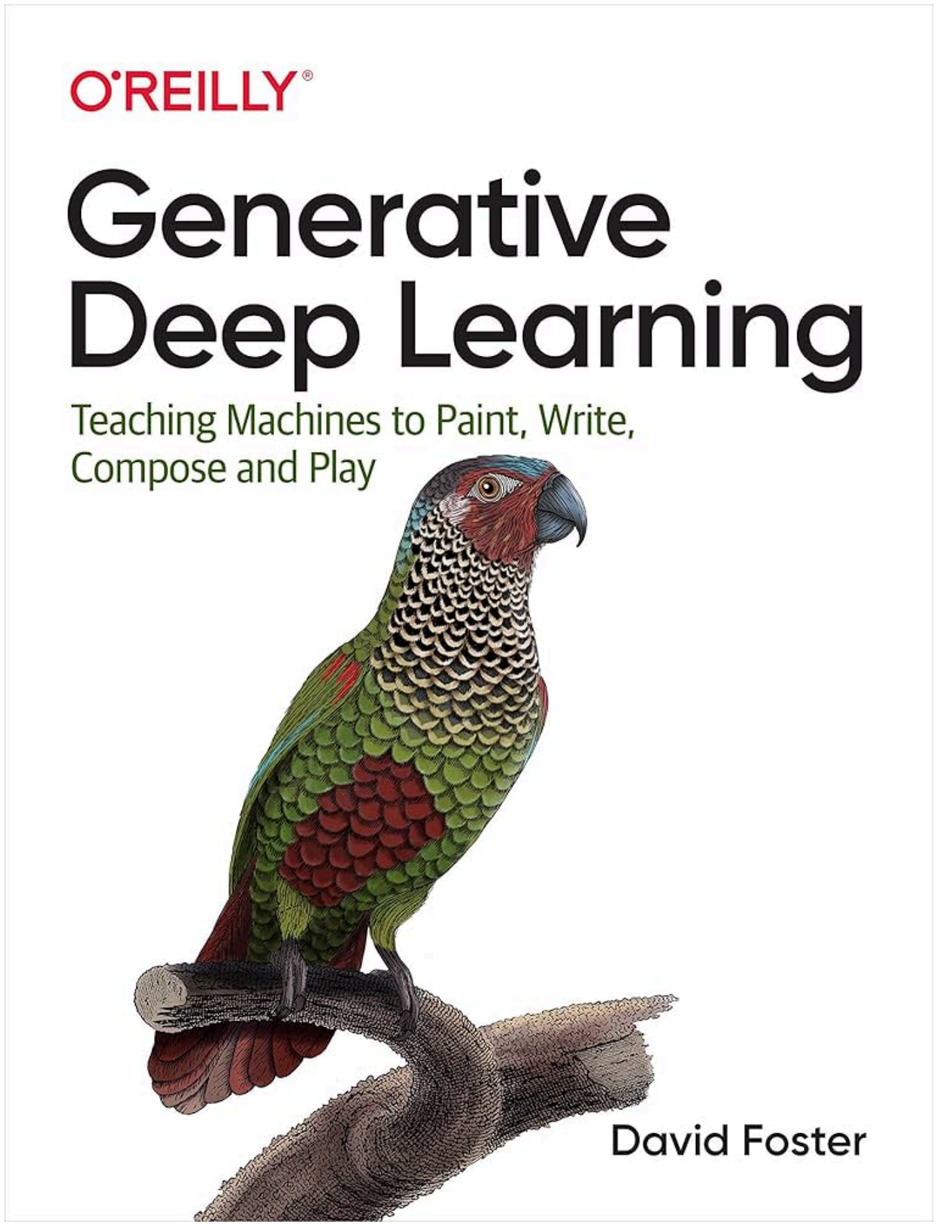
AIMultiple

LLMOps

Short Course about LLMOps in collaboration with google cloud

Other resources





Other resources

Check my other workshops here:

https://github.com/Azzedde/My-Workshops

Want to connect ?
Let’s do it on Linkedin !
My profile is [here]