



Build LLM-Powered Agentic Chatbots using LangGraph

Introduction to LangGraph and LanceDB
By Milad Mohammadi



About Me



- **MSc Student**, Information Technology Engineering, University of Tehran
- BSc Computer Engineering, University of Isfahan
- Graduate Researcher at
 - **NLP Lab**
Supervisor: Dr. Hesham Faili
 - **Computational Cognitive Psychology Lab**
Supervisor: Dr. Hadi Moradi
- LLM Researcher
- Research Field: LLM base **Language Agents** and Psychology
- **AI/Software Engineer**

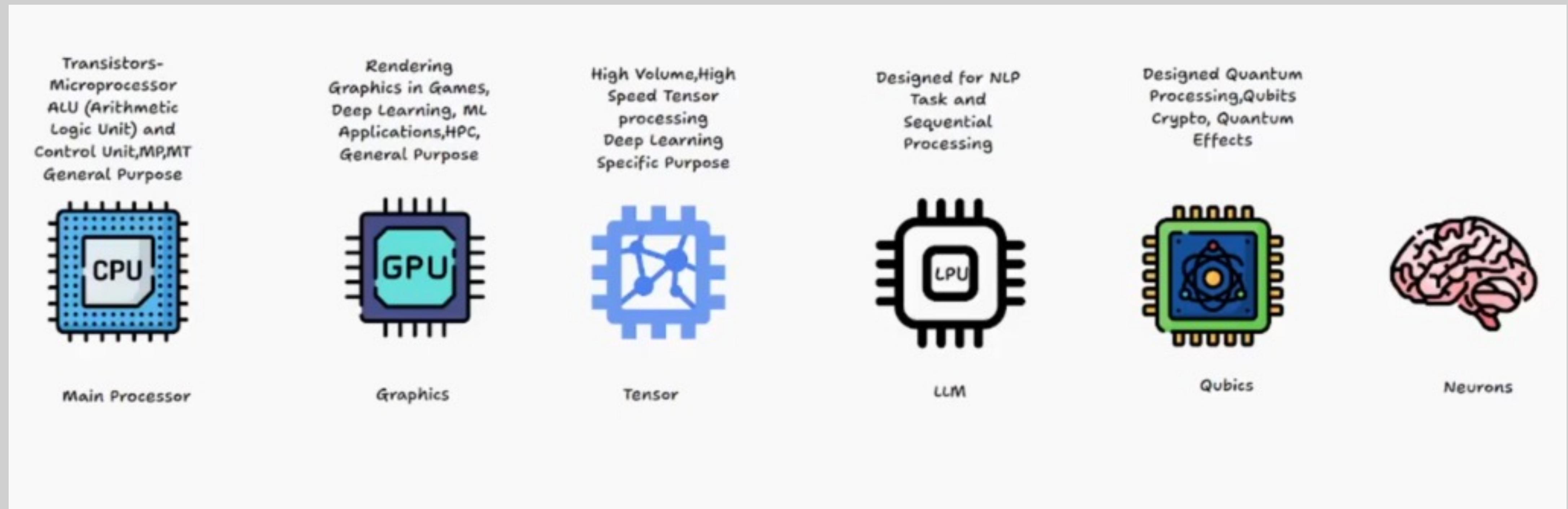


Workshop Plan

- . what is a language agent? / ReAct as an example agent design
- . a brief overview of tools and frameworks we are going to use
- . **CODING!**



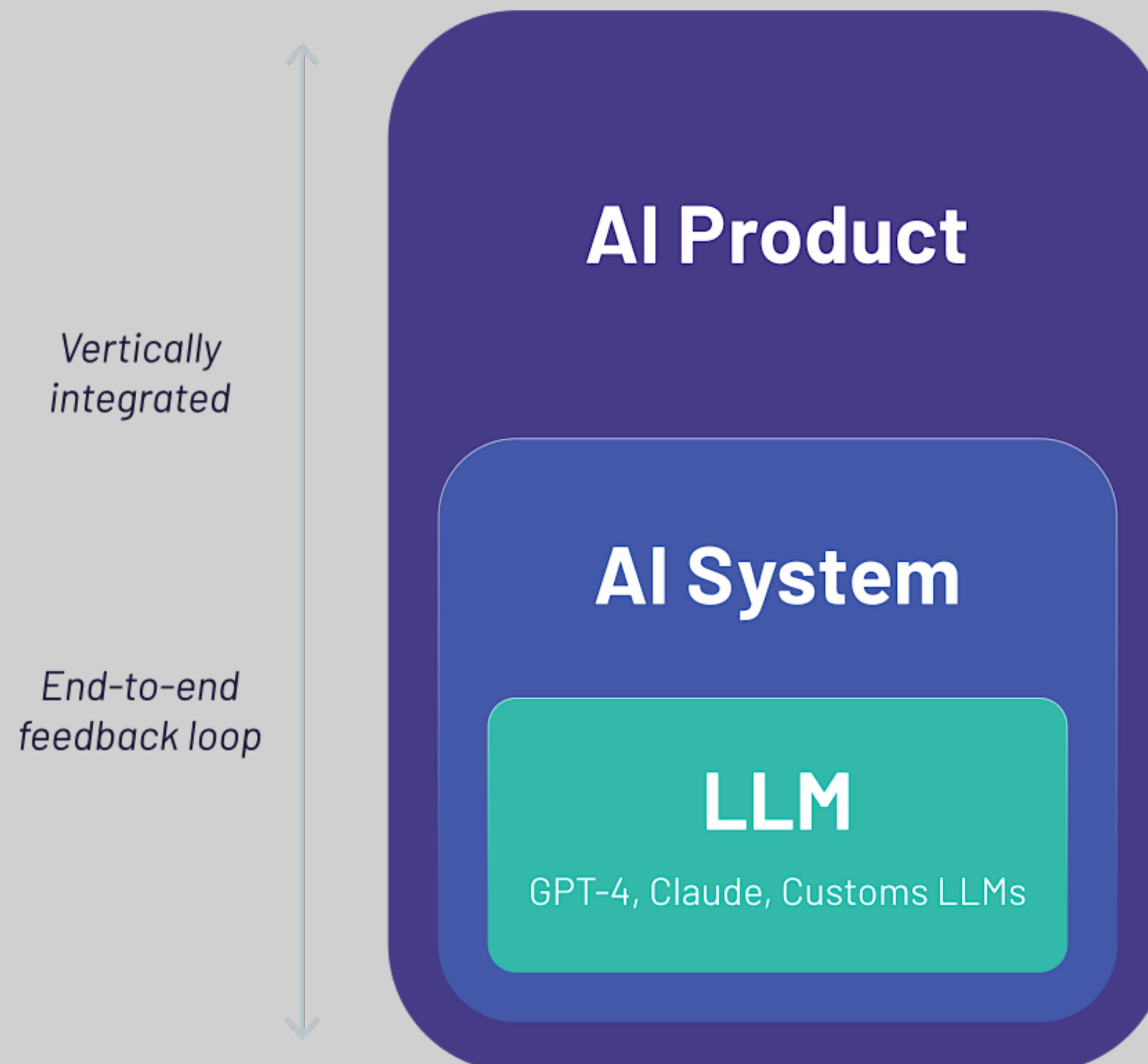
LLMs, but not just input/output



Source: JCharisTech



LLM based systems and products



AI Products re-bundle user workflow around AI systems (e.g. feedback to AI)

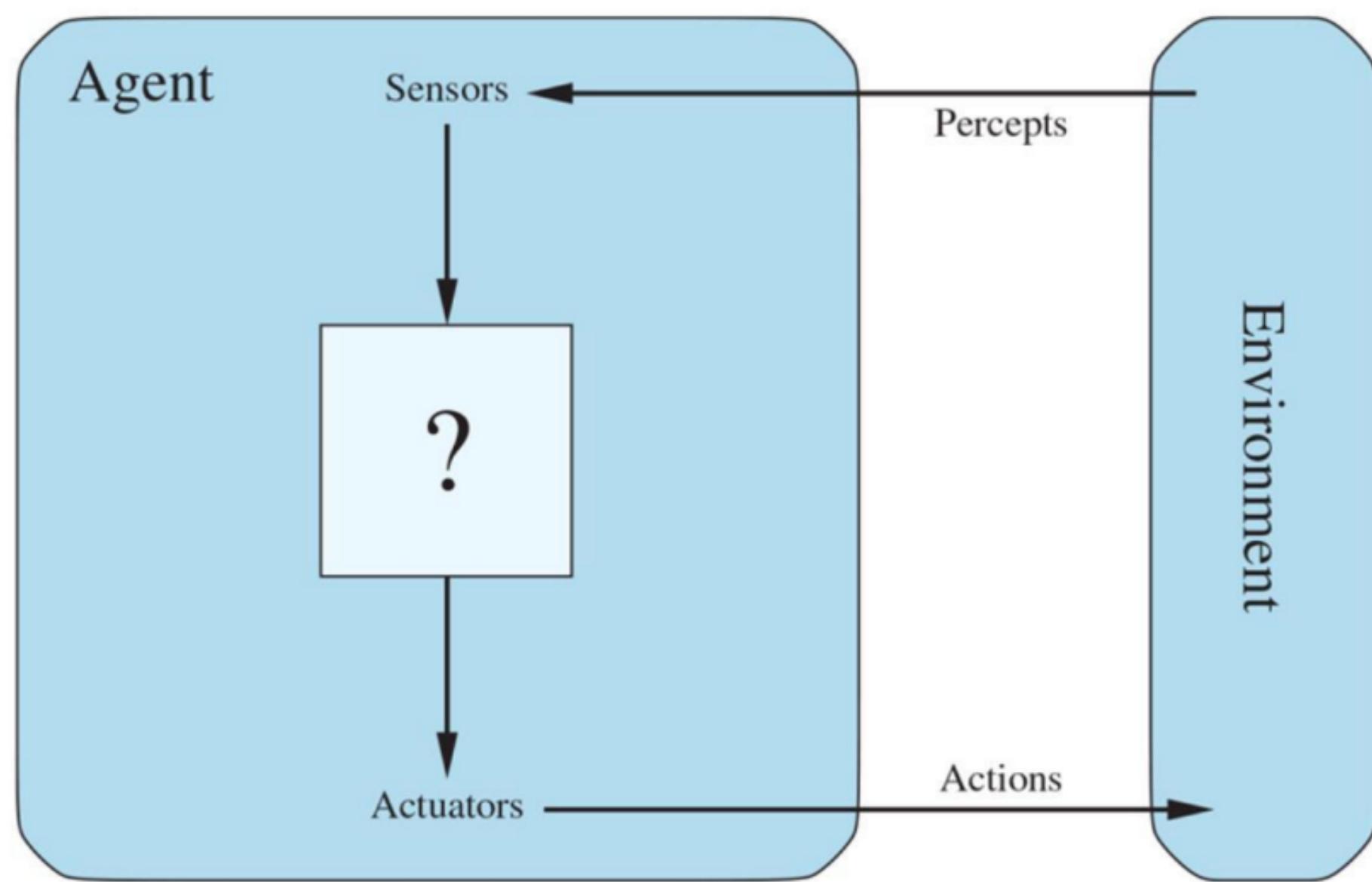
AI Systems combine multiple AI components (including LLMs) to do things like access private data, automate workflows, ...

LLMs are the core ("intel inside")

Source: BoxCars AI: Tabrez Syed



What is an Agent?

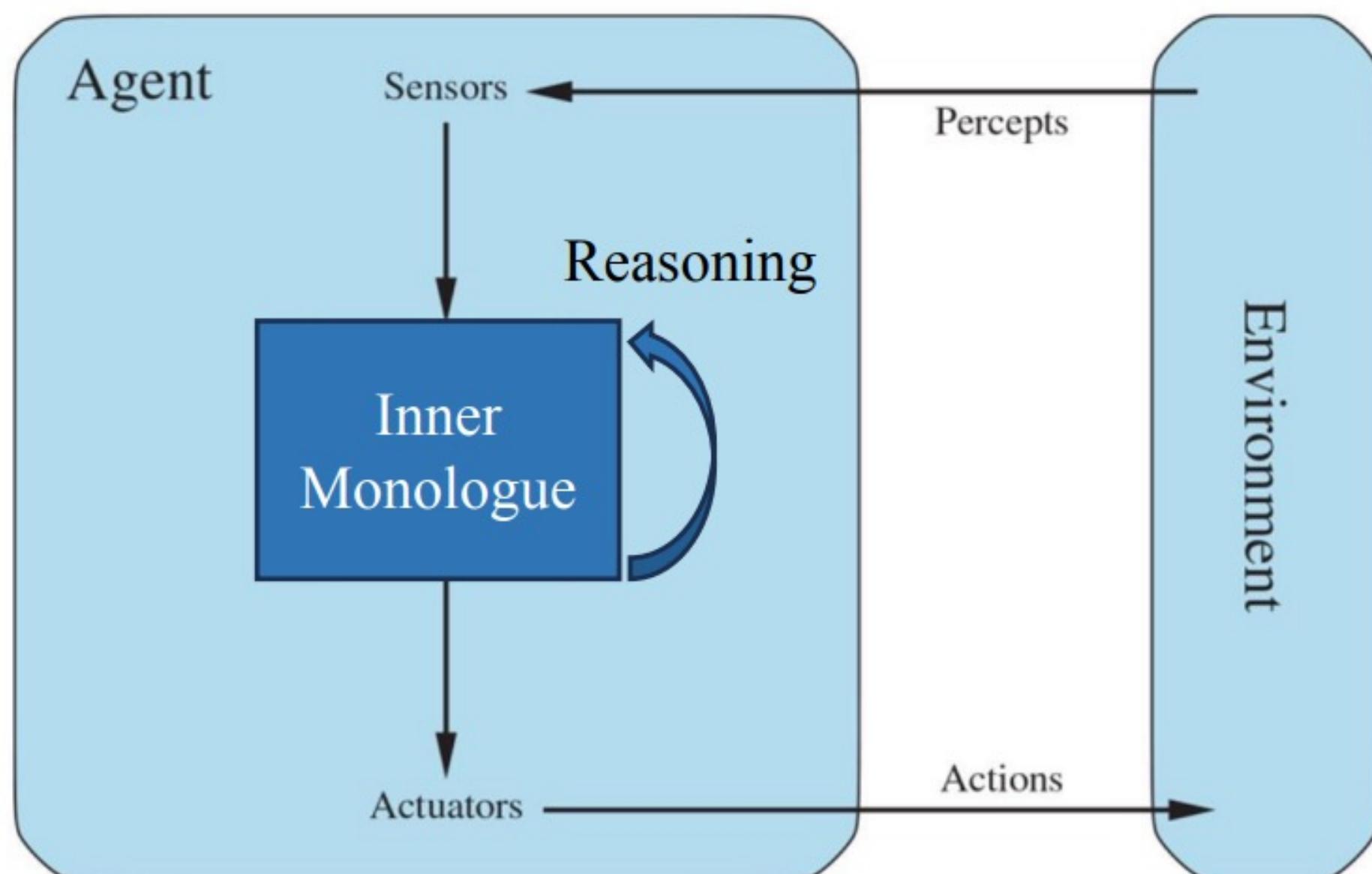


“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.”

-- Russell & Norvig, *AI: A Modern Approach* (2020)



What is a LLM Agent?



Adapted from Russell & Norvig (2020)

- **Reasoning by generating tokens** is a new type of action (vs. actions in external environments)
- **Internal environment**, where reasoning takes place in an inner monologue fashion
- **Self-reflection** is a ‘meta’ reasoning action (i.e., reasoning over the reasoning process), akin to metacognitive functions
- **Reasoning is for better acting**, by inferring environmental states, retrospection, dynamic replanning, etc.
- **Percept** and **external action spaces** are substantially expanded, thanks to using language for communication and multimodal perception

Source: EMNLP 2024 Tutorial: Language Agents: Foundations, Prospects, and Risks

Levels of autonomy in LLM applications

		Decide Output of Step	Decide Which Steps to Take	Decide What Steps are Available to Take
HUMAN-DRIVEN	1 Code			
	2 LLM Call			
	3 Chain			
	4 Router			
AGENT-EXECUTED	5 State Machine			
	6 Autonomous			







ReAct

Published as a conference paper at ICLR 2023

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao^{*1}, Jeffrey Zhao², Dian Yu², Nan Du², Izhak Shafran², Karthik Narasimhan¹, Yuan Cao²

¹Department of Computer Science, Princeton University

²Google Research, Brain team

¹ {shunyuy, karthikn}@princeton.edu

² {jeffreyzhao, dianyu, dunan, izhak, yuancao}@google.com

10 Mar 2023

ABSTRACT

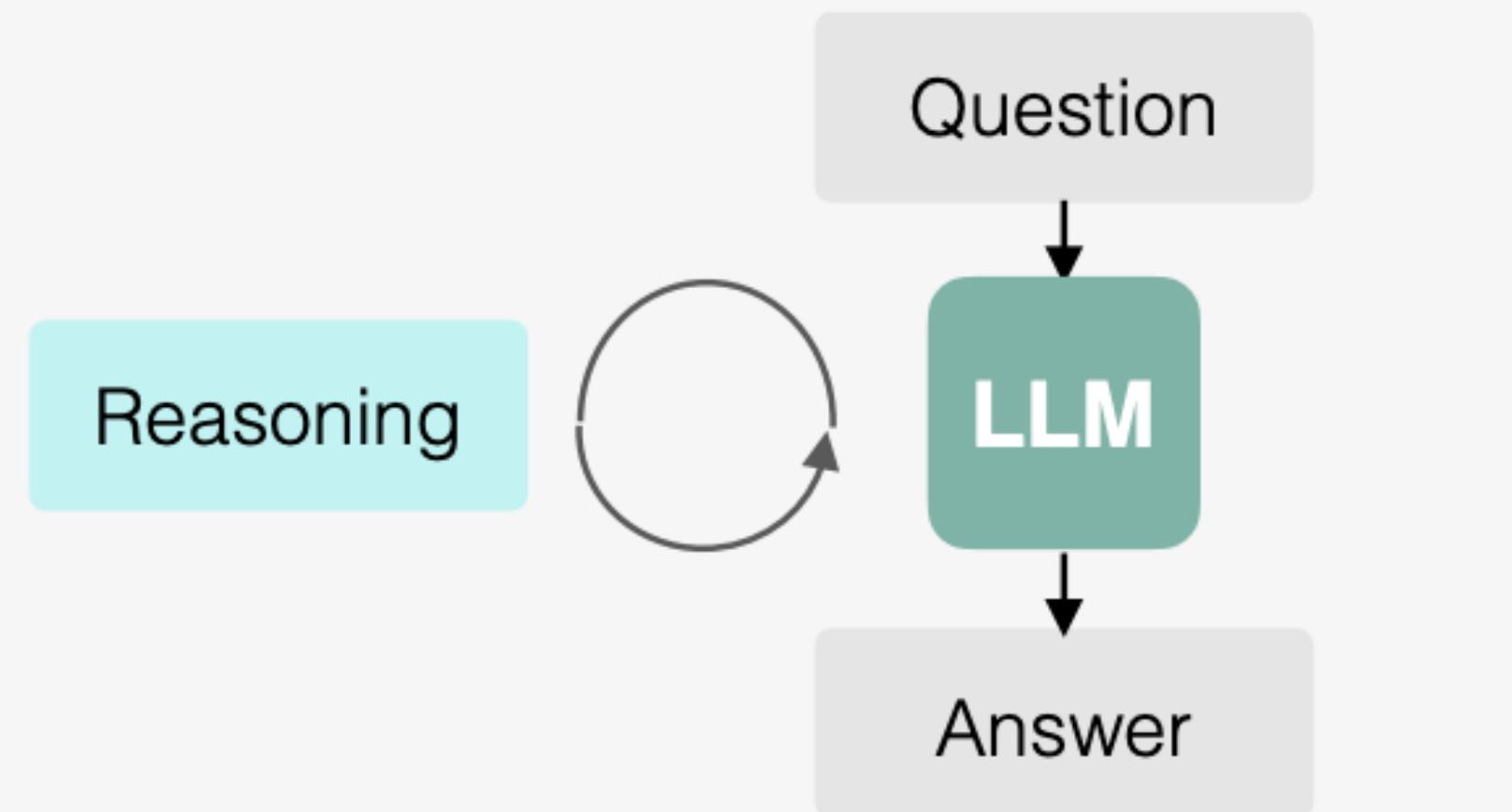
While large language models (LLMs) have demonstrated impressive performance across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation) have primarily been studied as separate topics. In this paper, we explore the use of LLMs to generate both reasoning traces and task-specific actions

Source: ReAct, Shunyu Yao

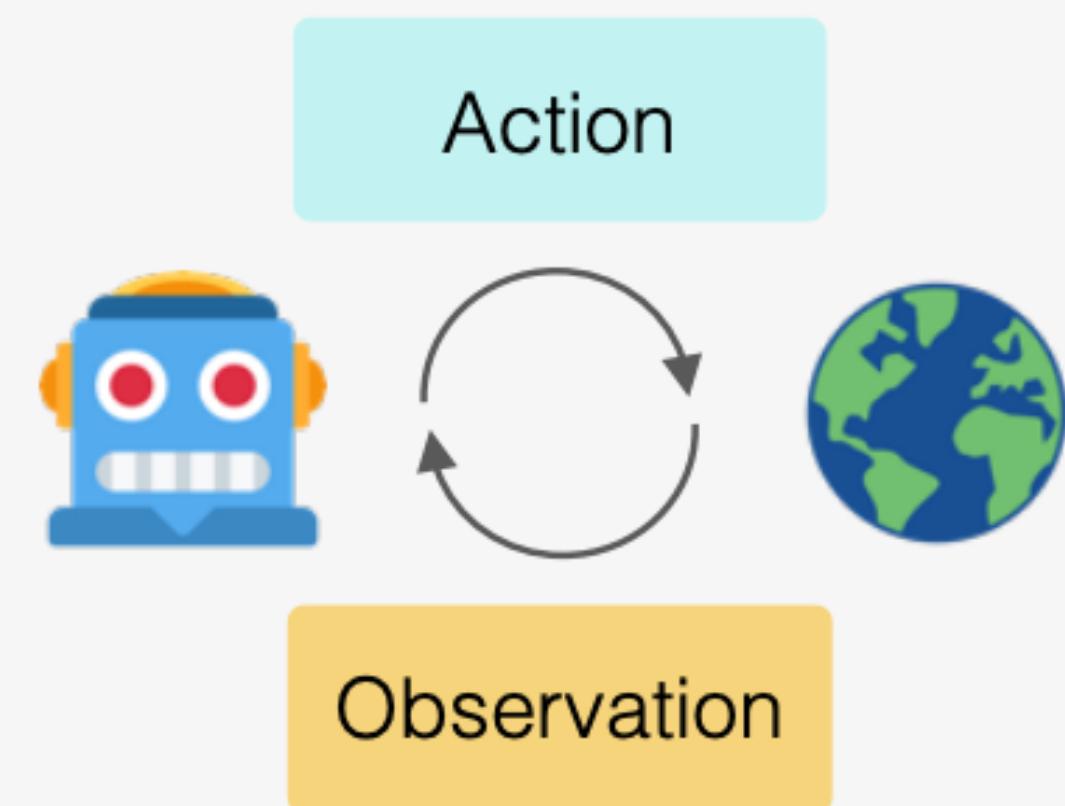


ReAct

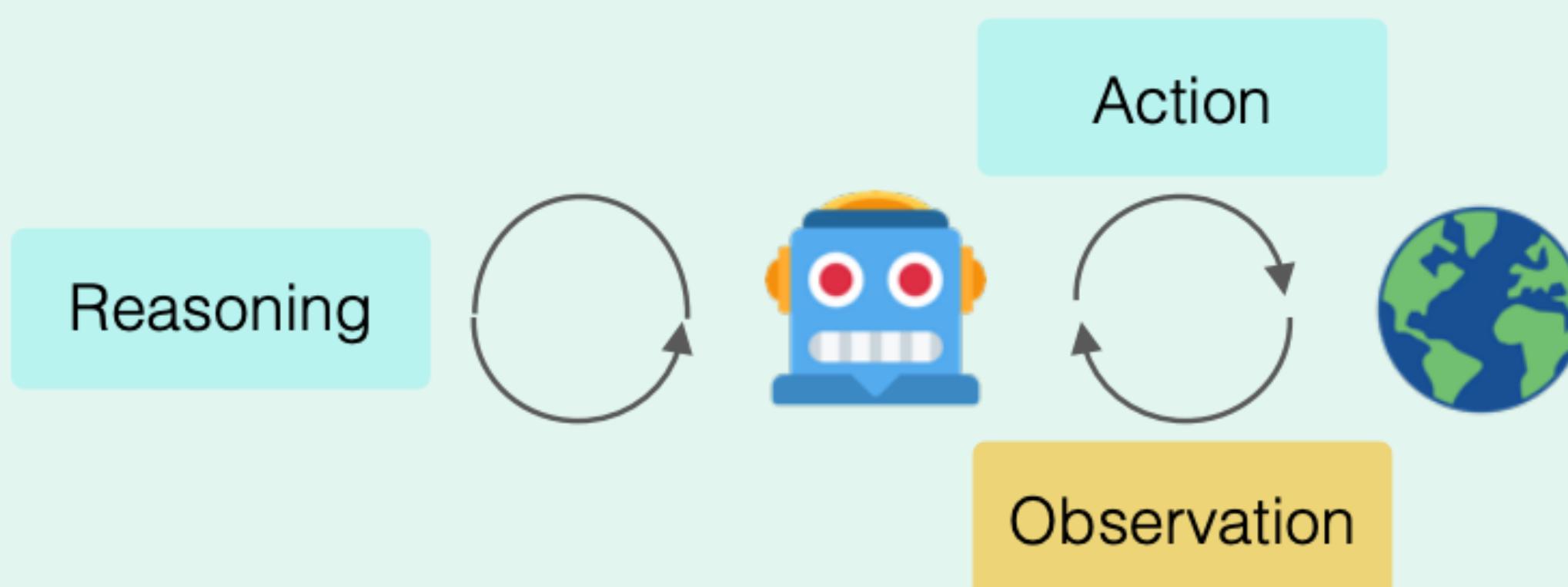
Reasoning (update internal belief)



Acting (obtain external feedback)



ReAct: a new paradigm of agents that **reason and act**



- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains

Source: ReAct, Shunyu Yao



What is a LLM Agent?

A zero-shot ReAct prompt

You are an agent that answers questions by using two actions:

- search[query]: Google search the query. You can also use it to calculate math.
- finish[answer]: return the answer

Your generation should have the following format:

Thought: to analyze the observation and inform the action.

Action: your action

Question: If I have seven trillion dollars today, can I buy Apple, Nvidia, and Microsoft? If not, how much more money do I need?

Source: ReAct, Shunyu Yao



What is a LLM Agent?

LLM

(GPT-4)

Thought: I need to find the current market capitalization of Apple, Nvidia, and Microsoft to understand how much it would cost to buy all of them.

Action: search["current market capitalization of Apple, Nvidia, and Microsoft"]



(Mar 2024)

Observation: "Nvidia's market cap stood at \$1.832 trillion as of about 10 a.m. ET Wednesday, while Alphabet's was at \$1.814 trillion, bringing Nvidia behind just Microsoft (MSFT) at \$3.015 trillion and Apple (AAPL) at \$2.836 trillion."

Thought: Now that I have the market cap for all companies, let's add them together and see if \$7 trillion is enough.

Action: search["1.832 trillion + 3.015 trillion + 2.836 trillion"]

Observation: 7683000000000

Thought: The amount of money needed in addition to the \$7 trillion is \$683 billion.

Action: finish["You need an additional \$683 billion to be able to buy Apple, Nvidia, and Microsoft."]



23

Source: ReAct, Shunyu Yao



What is a LLM Agent?

LLM

(GPT-4)

Thought: I need to find the current market capitalization of Apple, Nvidia, and Microsoft to understand how much it would cost to buy all of them.

Action: search["current market capitalization of Apple, Nvidia, and Microsoft"]

Observation: nothing found!

(My adversarial observation)

Thought: The search did not give any results. I will need to search for each company's market capitalization individually.

Action: search["current market capitalization of Apple"]

Observation: 'price': 180.75

(My adversarial observation)

Thought: This is the current price of Apple's stock, not its market capitalization. The market cap is calculated by multiplying the share price by the number of outstanding shares. Thus, I need to find the total number of Apple's outstanding shares to calculate its market cap.

Action: search["total number of Apple's outstanding shares"]



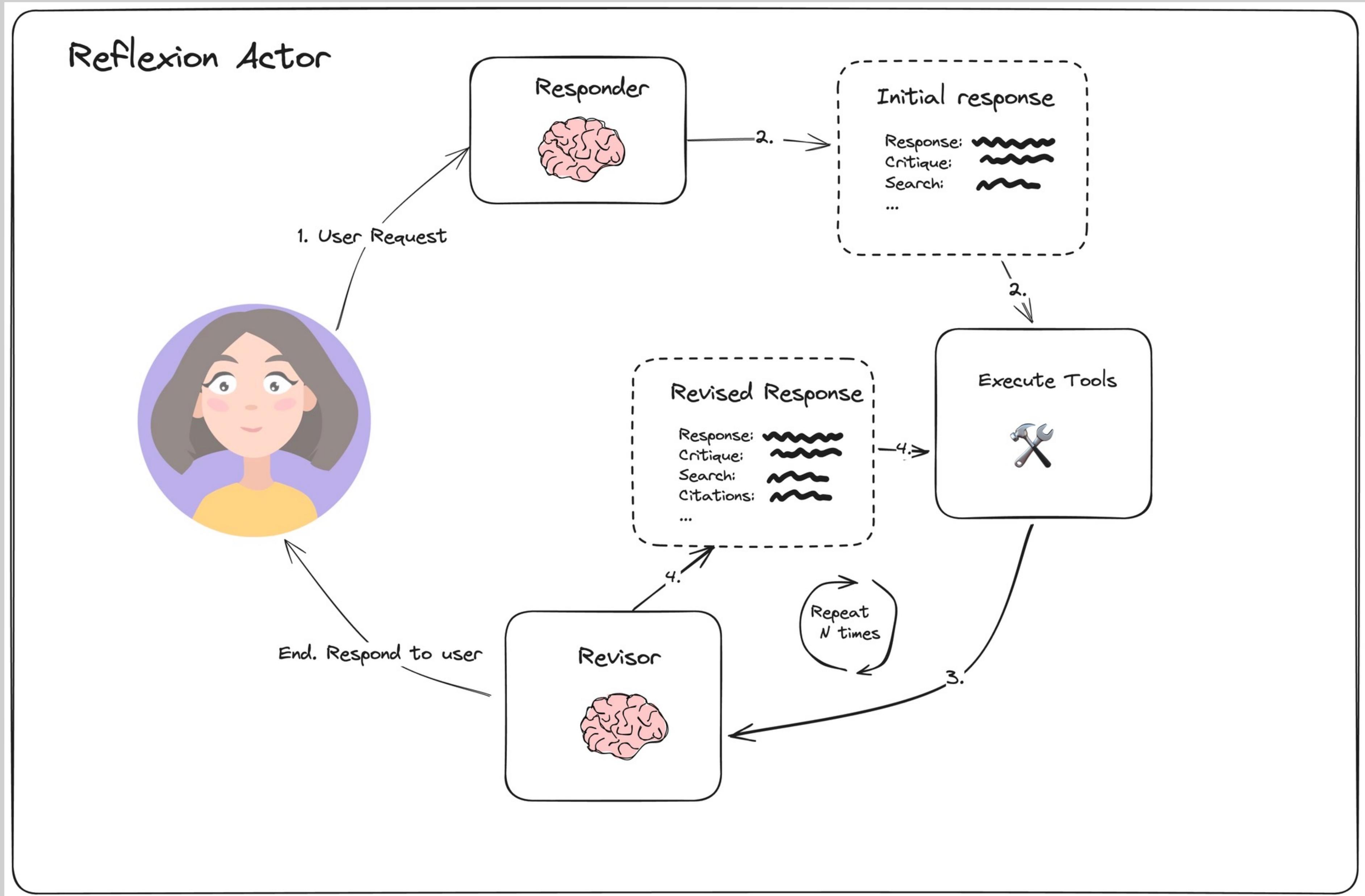
(Continues to solve the task)

Synergy: acting support reasoning, reasoning guides acting

Source: ReAct, Shunyu Yao



Other Language Agents Designs



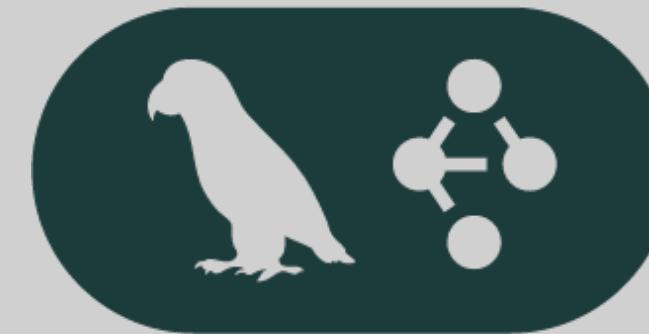
Source: Langchain



Tools and Frameworks!



Tools and Frameworks

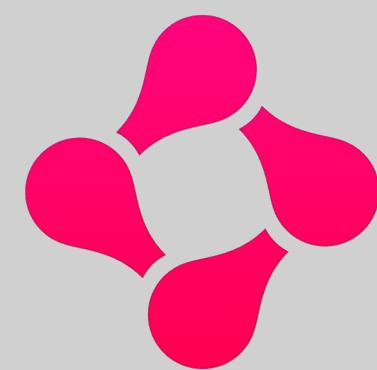


LangGraph

Gemini



LanceDB

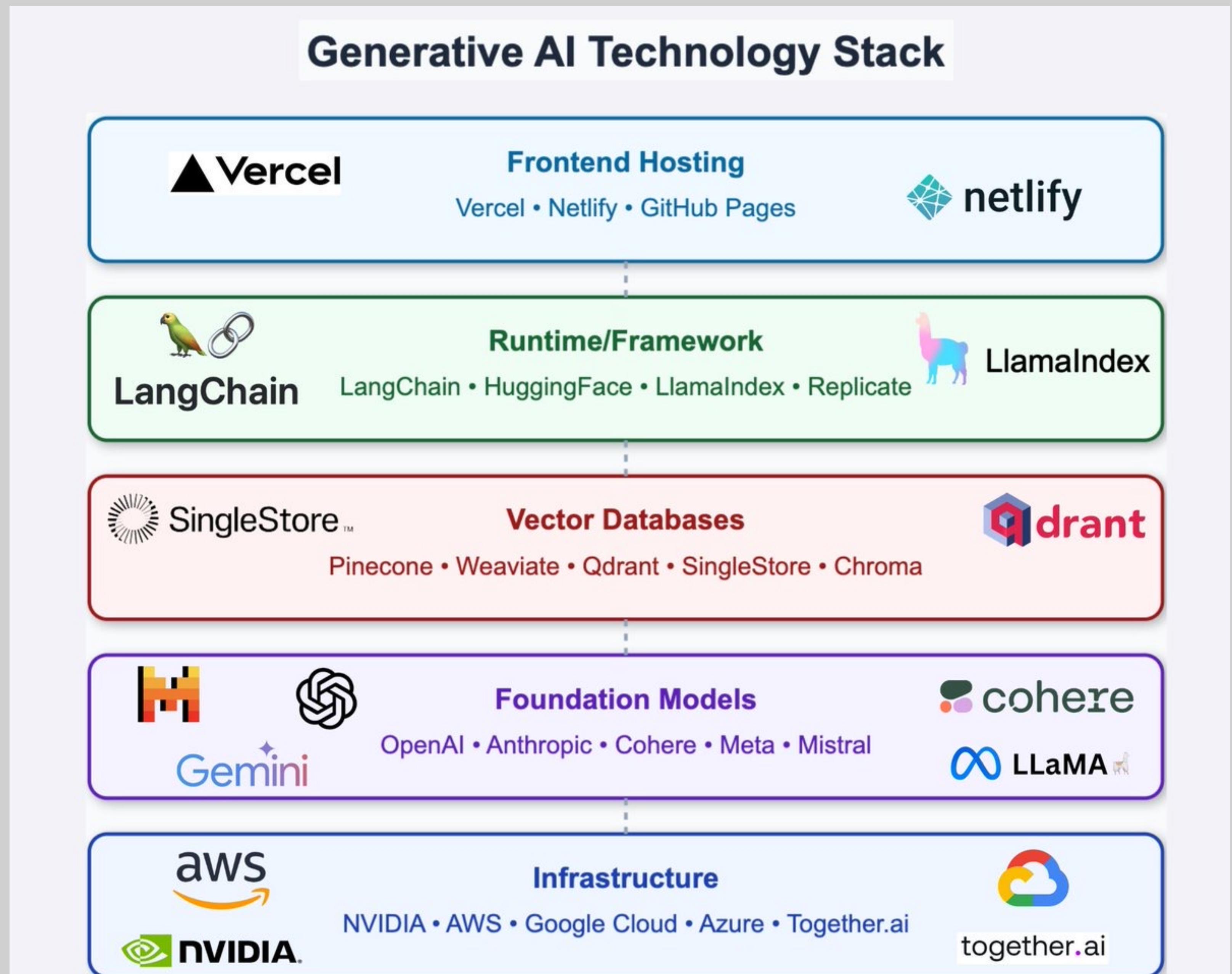


Chainlit





Stack



Source: Pavan
Belagatti™



Google Gemini

Gemini 1.5 Flash →
Gemini 1.5 Flash-8B
Gemini 1.5 Pro
Gemini 1.0 Pro
Text Embedding 004

Gemini 1.5 Flash AVAILABLE NOW

Our fastest multimodal model with great performance for diverse, repetitive tasks and a 1 million context window. Now generally available for production use.

Free of charge

The Gemini API “free tier” is offered through the API service with lower rate limits for testing purposes. Google AI Studio usage is completely free in all [available countries](#).

RATE LIMITS

15 RPM (requests per minute)

1 million TPM (tokens per minute)

1,500 RPD (requests per day)

INPUT PRICING

Free of charge

OUTPUT PRICING

Free of charge

CONTEXT CACHING

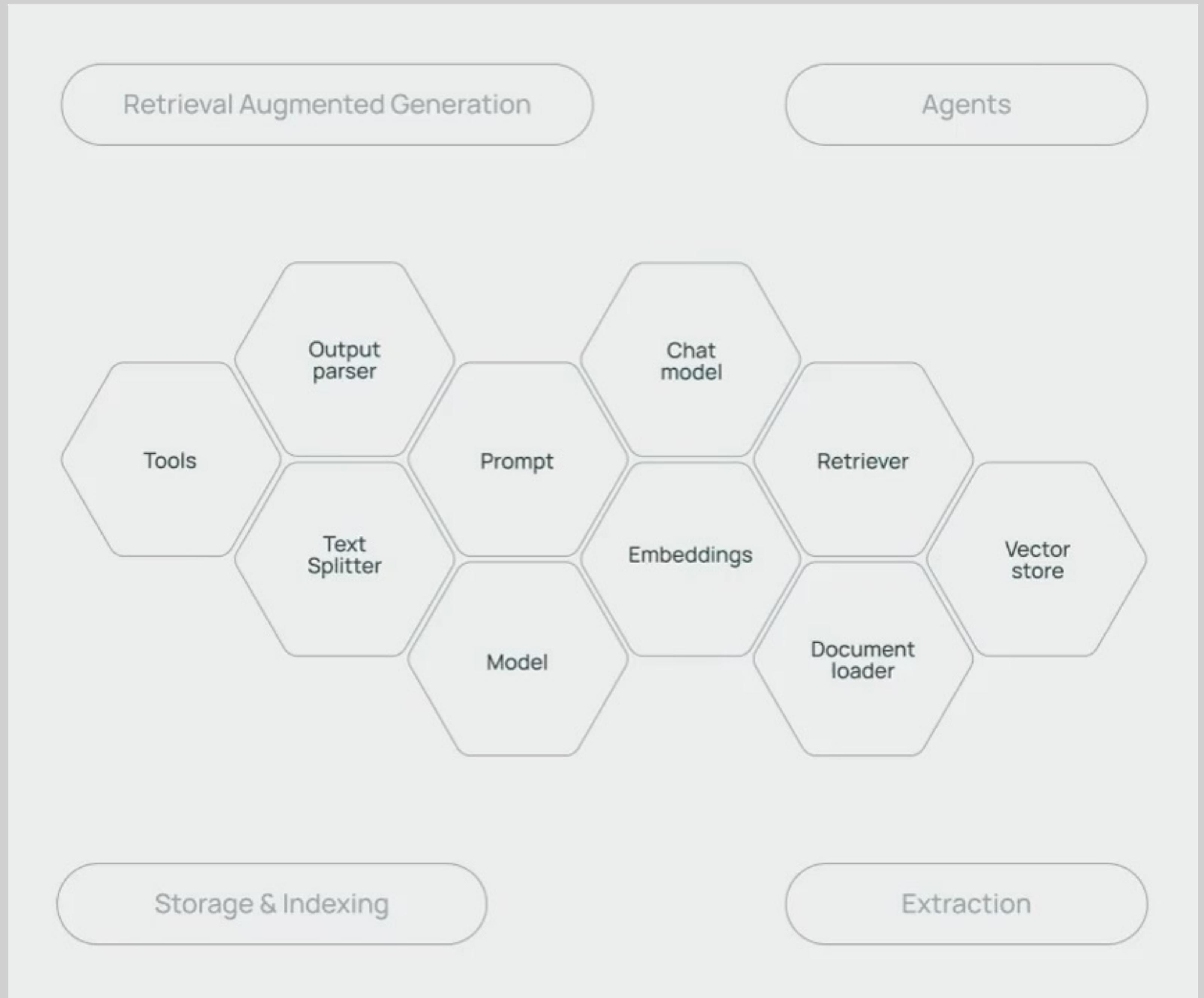
Free of charge, up to 1 million tokens of storage per hour

TUNING PRICE

Input/output prices are the same for tuned models. Tuning service is free of charge.

GROUNDING WITH GOOGLE SEARCH

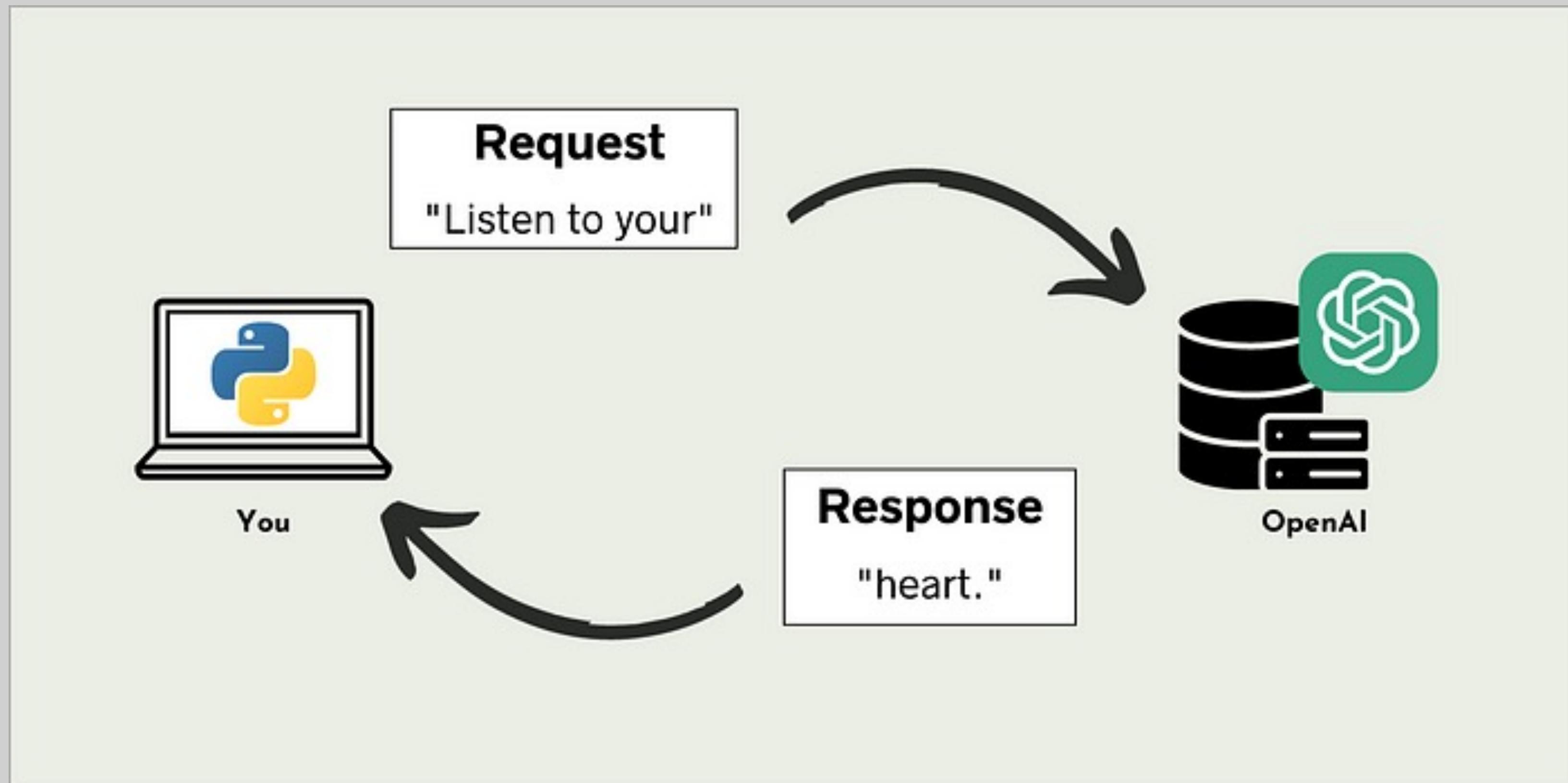
Not available



LangChain/LangGraph



OpenAI Client vs LangGraph



Source: TowardsDataScience: Shaw Talebi



OpenAI Client vs LangGraph

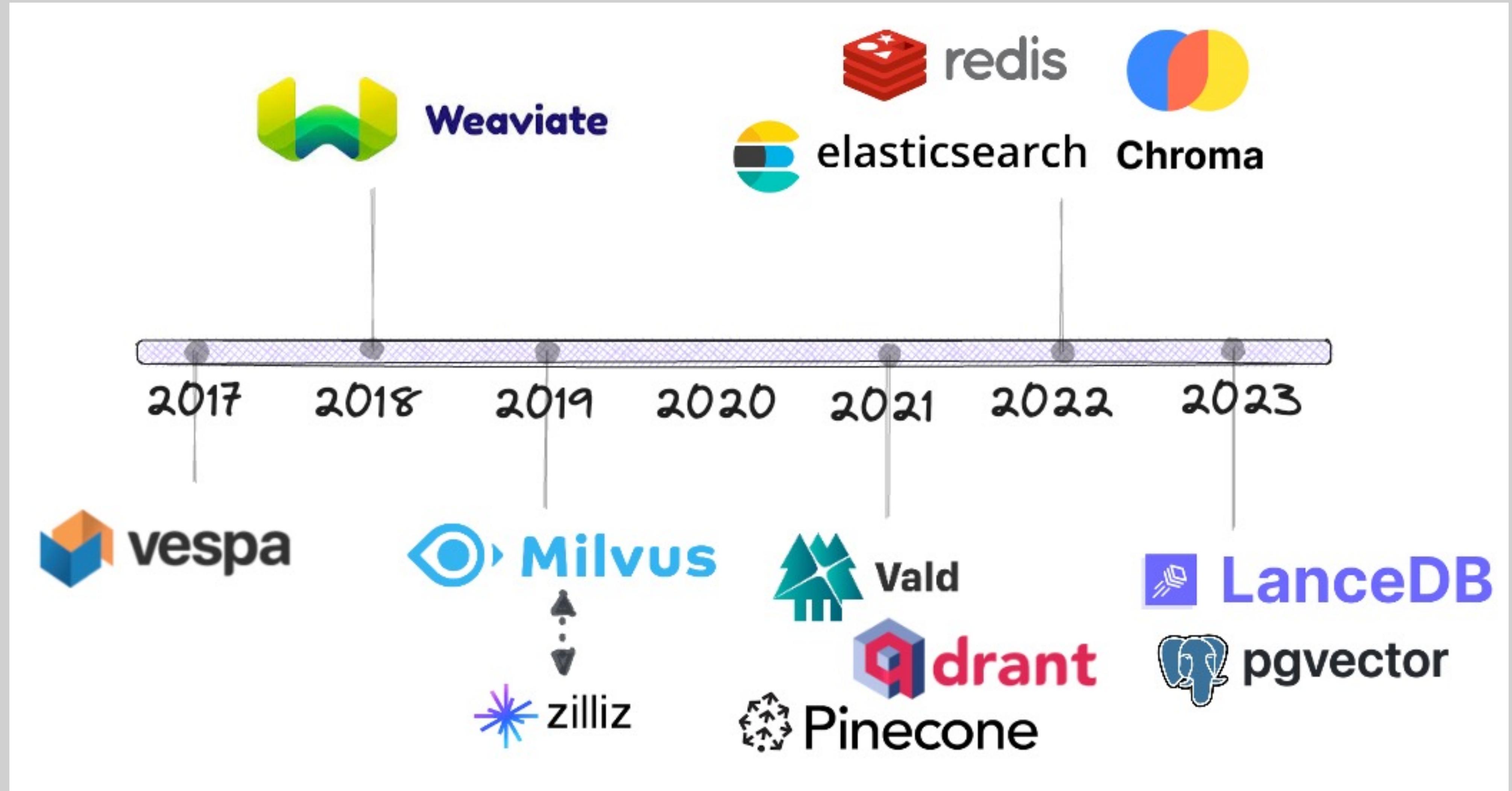
Agent Frameworks: Comparison of Abstractions



Source: Choosing Between LLM Agent Frameworks, TowardsDataScience, Aparna Dhinakaran



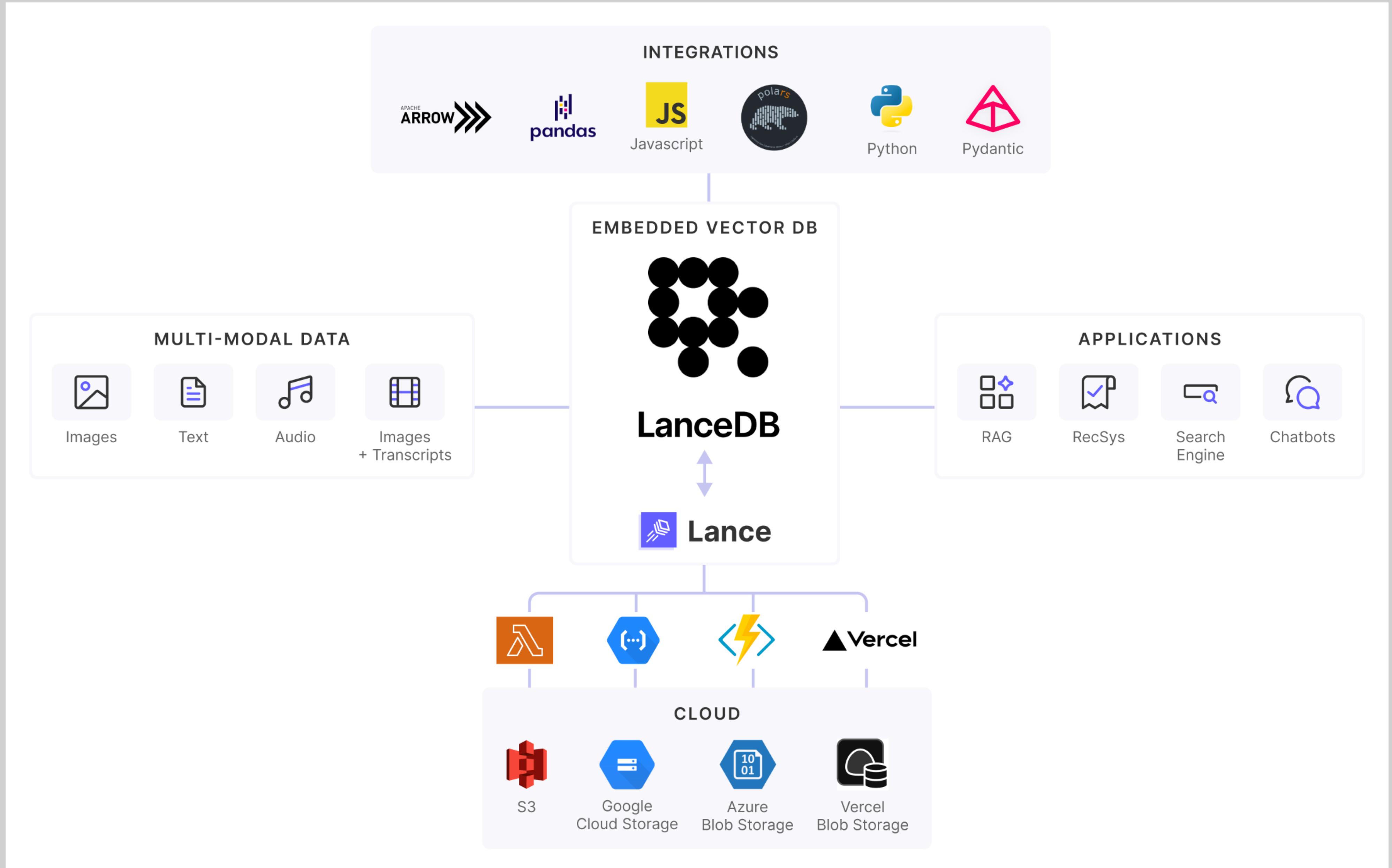
Vector Databased



Source: The Data Quarry blog



LanceDB



Source: LanceDB github



LlamaParse



Welcome	>
LlamaCloud	>
Getting Started	>
Usage Guides	>
Data Sources	>
Parsing & Transformation	>
Data Sinks	>
Embedding Models	>
Retrieval	>
Self-Hosting	>
LlamaParse	>
Getting Started	>
Using the UI	
Get an API Key	
Using in Python	
Using in TypeScript	
Using the REST API	
Input	
Output Modes	>
Examples	
Schemas	>
Features	>
Pricing and usage data	
Latency	
Privacy	

Getting Started

Overview

Welcome to LlamaParse, the world's best document parsing service, from [LlamalIndex](#). LlamaParse allows you to securely parse complex documents such as PDFs, PowerPoints, Word documents and spreadsheets into structured data using state-of-the-art AI.

LlamaParse is available as a standalone REST API, a Python package, a TypeScript SDK, and a web UI.

You can [sign up](#) to try it out or read the [onboarding documentation](#).

Original Document

Relationships Between CAP and Financial Performance
The following graphs illustrate how CAP for our NEOs aligns with the Company's financial performance measures as detailed in the Pay Versus Performance table above for each of Fiscal 2021, 2022, 2023, and 2024, as well as between the TSRs of NVIDIA and the Nasdaq100 Index, reflecting the value of a fixed \$100 investment beginning with the market close on January 24, 2020, the last trading day before our Fiscal 2021, through and including the end of the respective listed fiscal years.



NEO CAP versus Net Income & Non-GAAP Operating Income



Relationships Between CAP and Financial Performance

The following graphs illustrate how CAP for our NEOs aligns with the Company's financial performance measures as detailed in the Pay Versus Performance table above for each of Fiscal 2021, 2022, 2023, and 2024, as well as between the TSRs of NVIDIA and the Nasdaq100 Index, reflecting the value of a fixed \$100 investment beginning with the market close on January 24, 2020, the last trading day before our Fiscal 2021, through and including the end of the respective listed fiscal years.

NEO CAP versus TSR

Fiscal Year	CEO CAP	Other NEOs Average CAP	NVIDIA TSR	Nasdaq100 Index TSR
Fiscal 2021	\$79.6	\$27.9	\$207.79	\$141.39
Fiscal 2022	\$105.5	\$38.5	\$365.66	\$158.12
Fiscal 2023	\$(-4.1)	\$(-1.4)	\$326.34	\$133.09
Fiscal 2024	\$234.4	\$85.6	\$578.42	\$190.57

Note: Values on right y-axis range from (\$20) to \$1,120

NEO CAP versus Net Income & Non-GAAP Operating Income

Fiscal Year	CEO CAP	Other NEOs Average CAP	Net Income	Non-GAAP Operating Income
Fiscal 2021	\$79.6	\$27.9	\$4,332	\$6,803
Fiscal 2022	\$105.5	\$38.5	\$9,752	\$12,690
Fiscal 2023	\$(-4.1)	\$(-1.4)	\$4,368	\$9,368
Fiscal 2024	\$234.4	\$85.6	\$29,760	\$37,134

Note: Values on right y-axis range from (\$2,000) to \$40,000

All information provided above under the "Pay Versus Performance" heading will not be deemed to be incorporated by reference into any filing of the Company under the Securities Act of 1933, as amended, or the Securities Exchange Act of 1934, as amended, whether made before or after the date hereof and irrespective of any general incorporation language in any such filing, except to the extent the Company specifically incorporates such information by reference.

63

Source: LlamaParse



Chainlit

Chainlit

Search... ⌘K

Literal AI Github Join Discord Join Discord

Documentation Examples API Reference

Chainlit Help

Get Started

Overview

Installation

In Pure Python

Integrations >

Deploy

Overview

Platforms >

Basic Concepts

Chat Life Cycle

Starters

Message

Step

User Session

Element

Action

Advanced Features

Streaming

Ask User

Get Started

Overview

Chainlit is an open-source Python package to build production ready Conversational AI.

The screenshot shows a dark-themed user interface for the Chainlit application. On the left, there's a sidebar with navigation links like 'Get Started', 'Overview', 'Installation', etc. The main area has a large heading 'Overview' and a sub-section 'Chainlit is an open-source Python package to build production ready Conversational AI.' Below this is a large image of the Chainlit application interface. The interface features a dark background with a light-colored central panel. It shows a 'Today' section with a message 'Write a Chainlit hello world ...' and a question 'What kind of application can I create?'. To the right of this is a code editor window titled 'Readme' containing Python code for handling chat messages. Below the code editor is a terminal window showing the command 'chainlit run app.py -w'. A note below the terminal says: 'The -w flag enables auto-reloading, so you don't need to restart the server every time you make changes. Your Chainlit app should now be accessible at http://localhost:4000, and it will greet users with "Hello, World!" when they start a chat.' At the bottom of the interface is a text input field with placeholder 'Type your message here...' and a send button. The footer of the interface says 'Build Conversational AI with Chainlit'.

On this page

Key features

Integrations

3. Run the Application: Open a terminal, navigate to the directory containing `app.py`, and run the following command:

```
chainlit run app.py -w
```

The `-w` flag enables auto-reloading, so you don't need to restart the server every time you make changes. Your Chainlit app should now be accessible at <http://localhost:4000>, and it will greet users with "Hello, World!" when they start a chat.

Type your message here... ?

Build with Chainlit

Key features

1. **Build fast:** Get started in a couple lines of Python
2. **Authentication:** Integrate with corporate identity providers and existing authentication infrastructure
3. **Data persistence:** Collect, monitor and analyze data from your users



Let's code!



Good Luck with
LLMs!
you'll need it ;)

For further discussions or questions, feel free to reach out via email:
miladmohammadi@ut.ac.ir