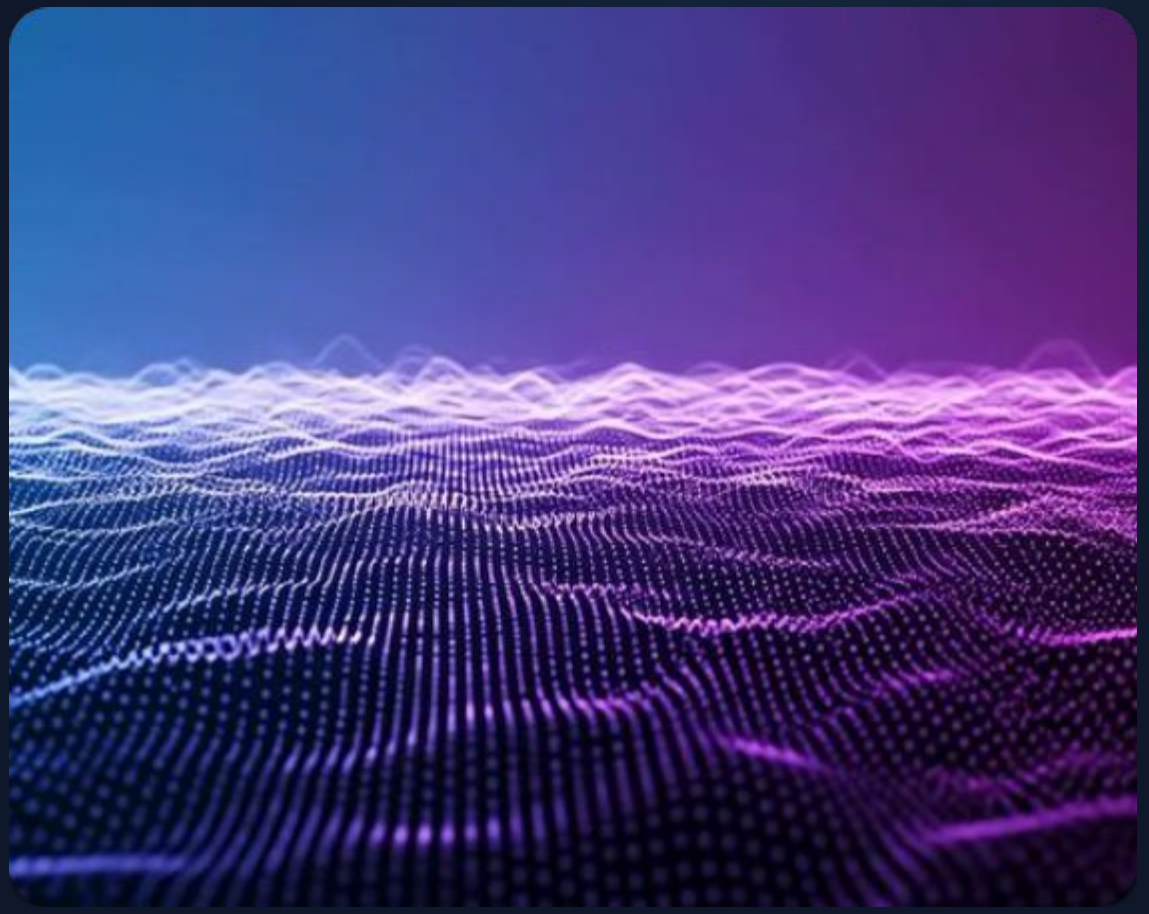
DAY 01 MODULE

GenAI & Agentic AI Bootcamp

Mastering the shift from Generative models to Autonomous Agents.



By

Shikha Tyagi

M.Tech IIT Delhi

Founder AIJamic, training & consulting

About me

- Total 12+ years of experience in AI/ML domain
- Since last 3 years I have been working on GenAI initiatives for productivity and automation
- Industry Experience: Amex, PayTM, Yahoo, Peoplestrong
- Training Experience: Conducted 20+ trainings with multiple clients

Guidelines

- Attendance is mandatory for all 5 sessions
- Hands on activity is mandatory
- 15 min break at 10:30PM
- QnA session at the end (10-15 min)
- Feel free to drop your questions in chat
- There will be quizzes in-between, drop your answers in chat

5 day roadmap

1

Shift

Agentic Thinking vs. Chatbots

2

Brain

LLMs & Prompting

3

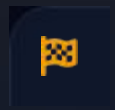
Hands

Function Calling & Tools

4

Memory

RAG & Vectors



Build

End to end pipeline & Capstone

Today’s Agenda

01 Foundations

History, Evolution, and Traditional vs. GenAI.

02 The Engine

LLMs, Transformers, and Tokenization.

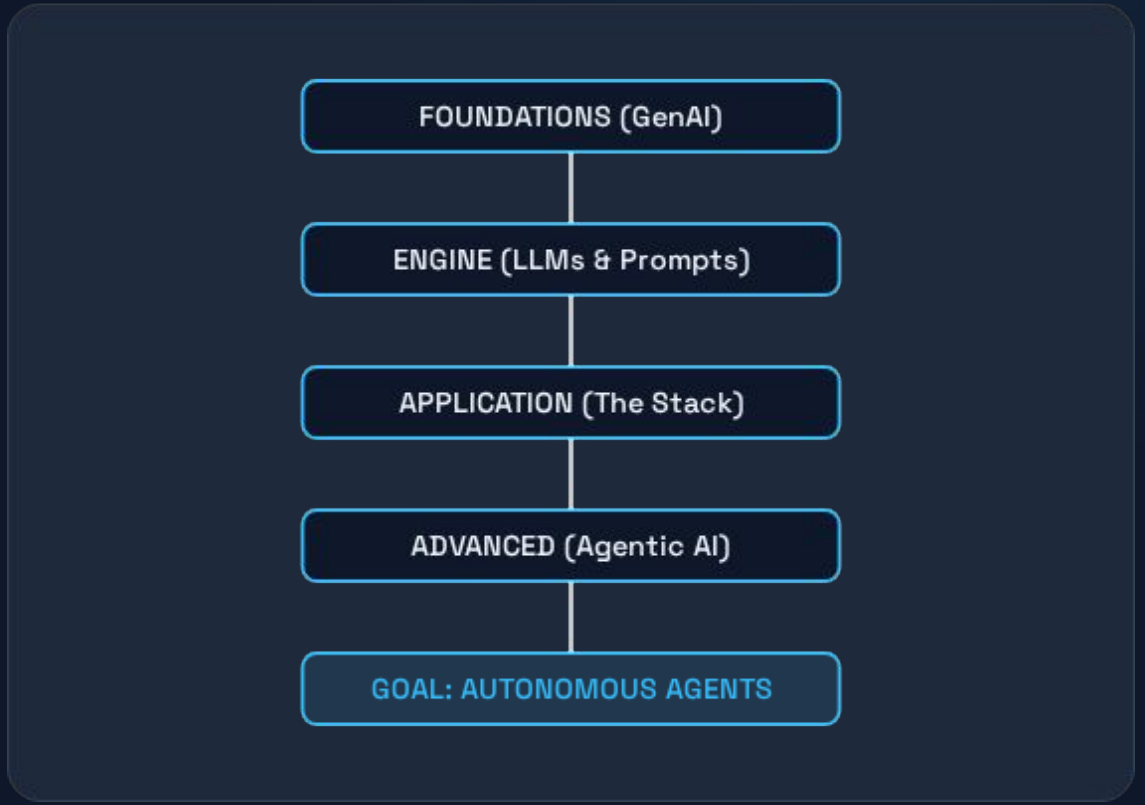
03 Agentic AI

Reasoning loops, Tools, and Autonomous Agents.

04 The Stack

LangChain, LlamaIndex, and Vector Databases.

05 Hands-on (1 hour)



Generative AI Foundations

From classification to creation.

Quiz: Core Concepts

Question 1:

AI has been there since?

A 1950

B 2022

C 2000

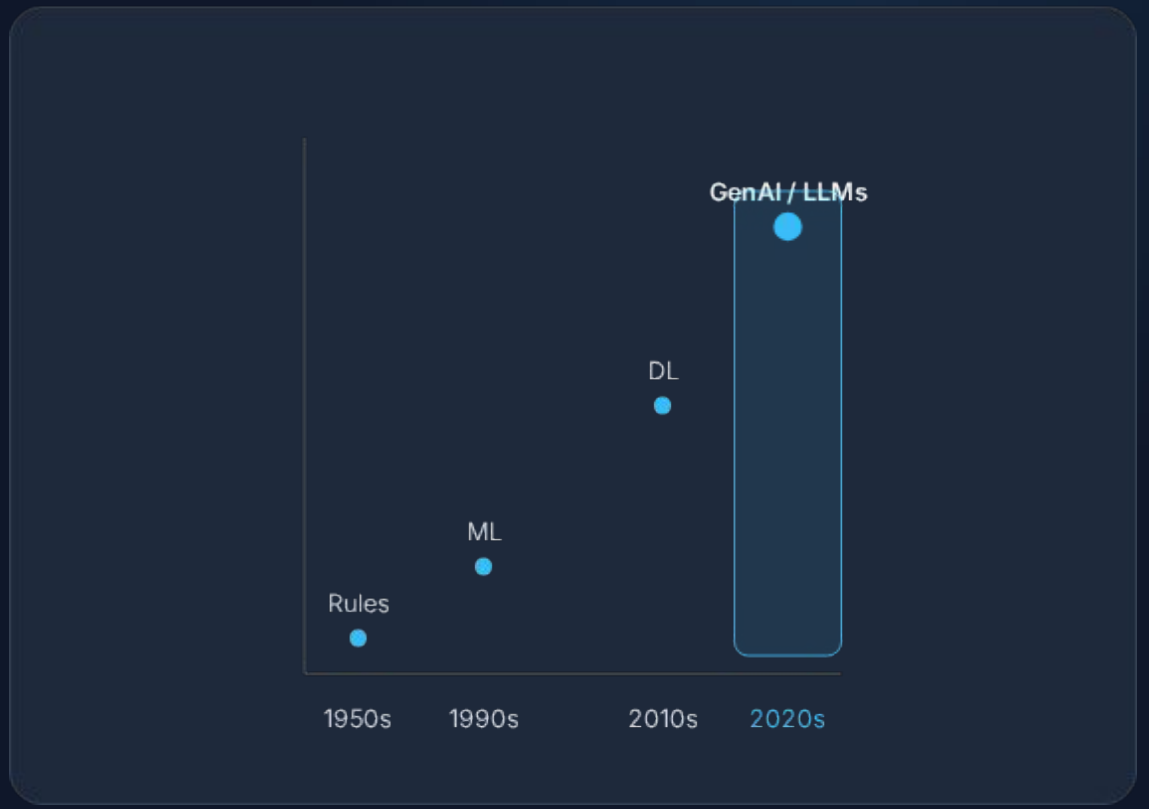
AI isn’t new : Why the Hype Now?

It’s ~70 years old.

- **1950s:** The Turing Test.
- **1990s:** Deep Blue beats Kasparov (Chess).
- **2010s:** Siri, Alexa, and AlphaGo.

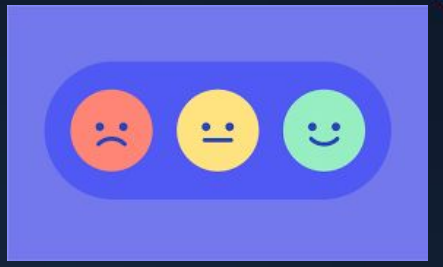
The Tipping Point: Accessibility

Before, you needed expert skill and Python code to use AI. **Now, you just need English.** The Natural Language Interface made AI accessible to everyone.



Example: Sentiment Analysis

Customer Review: The Hotel service was pathetic

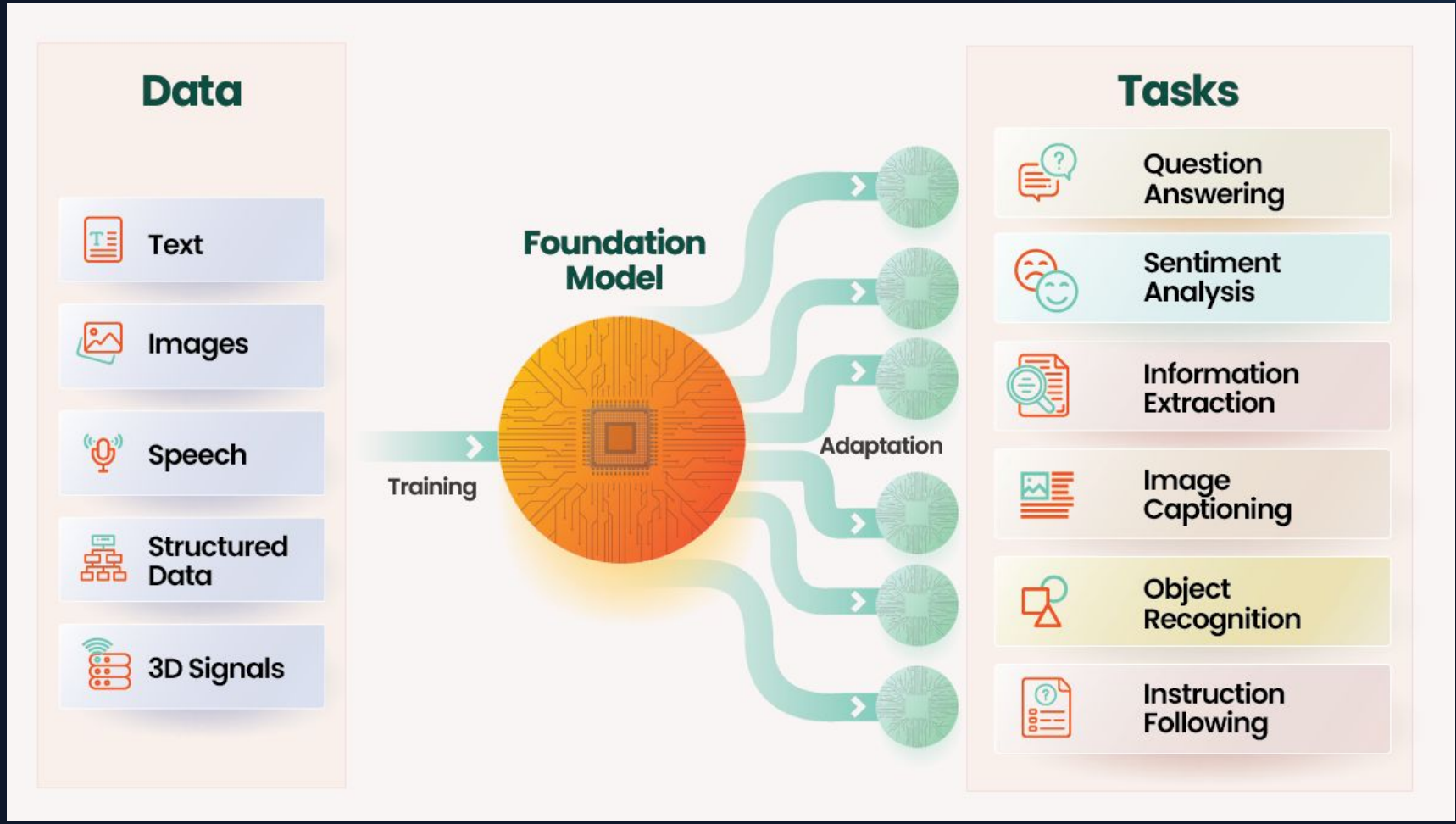


Before	Now
Collect Data Perform Data Pre Processing Train a Model Analyse Result Final Outcome	Classify this sentence into following sentiment categories: Positive, Neutral, Negative
Time Taken: (Few days to week)	**Time Taken: Instant**

Traditional vs. Generative AI

Aspect	Traditional AI (Discriminative)	Generative AI (Creative)
Goal	Classify, Predict, Cluster	Create new data (Text, Code, Images)
Function	Maps Input → Label	Maps Input → New Output Distribution
Example	Spam Filter (Yes/No)	Writing an email response
Training	Supervised (Labeled Data)	Self-Supervised (Massive Unlabeled Data)

How It Works?



https://www.tredence.com/generative-ai-101

How It Works? Is word == token

Training Data

Ingesting petabytes of text from the internet to learn language structure.

Neural Network

Learning probabilistic relationships between words (tokens).

Inference

Predicting the "next best token" to generate coherent sequences.

Popular GenAI models?

Quiz: Core Concepts

Question 2:

Which of the following is a primary task of Generative AI?

A. Classifying emails as spam or not spam.

B. Drafting a new marketing blog post.

C. predicting house prices based on historical data.

Large Language Models (LLMs)

Transformers, Tokens, and Context.

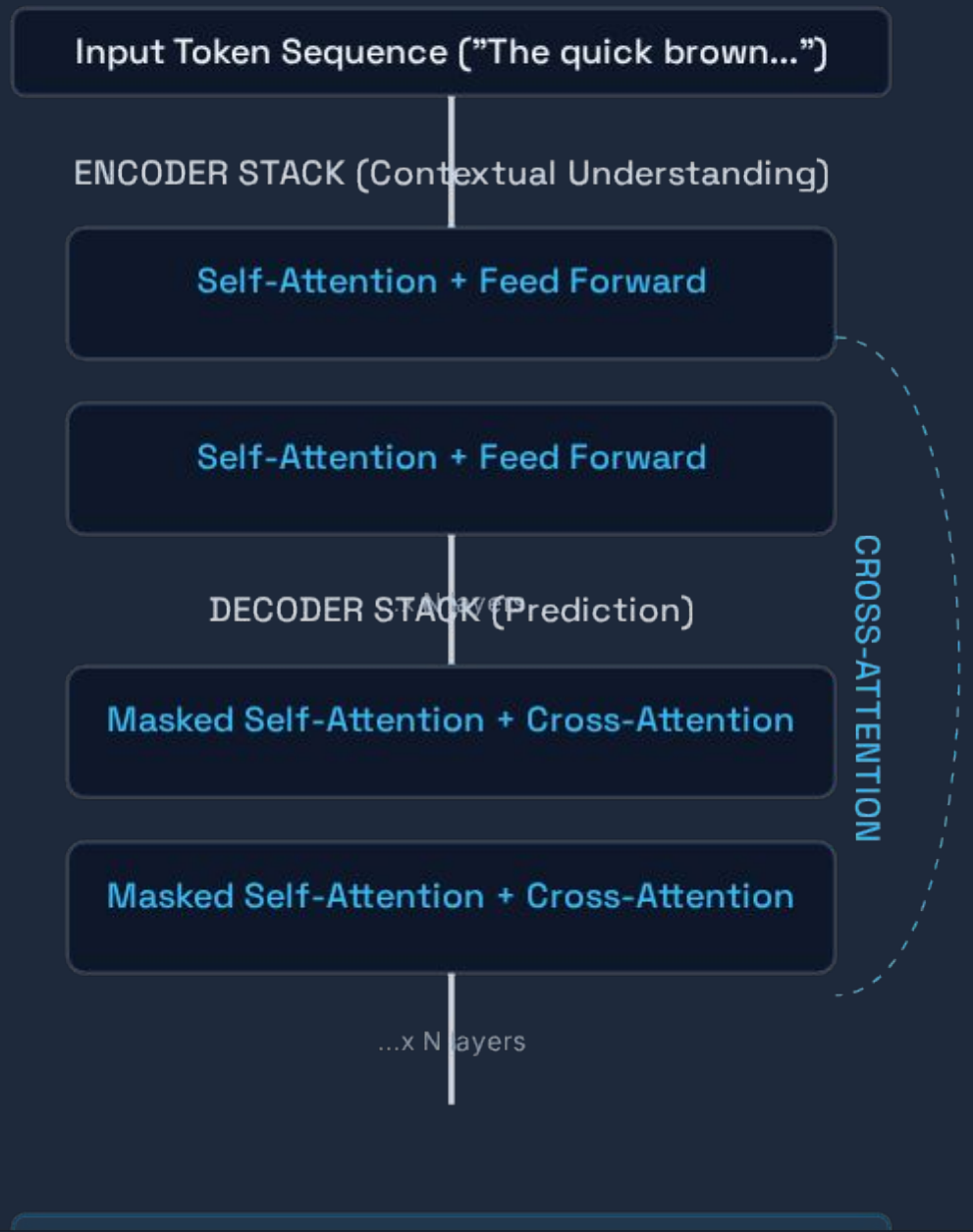
Understanding LLMs

LLMs are probabilistic engines trained to predict the next token.

They are powered by the **Transformer** architecture, which allows them to pay "Attention" to different parts of a sentence simultaneously.

- Billions of Parameters
- Contextual Awareness

Apple is a .....



Tokens: The Atomic Unit

LLMs process text as **Tokens**, not words.

- A token ~approx 0.75 words.
- 1,000 tokens ~approx 750 words.
- "Smart" → 1 Token
- "Ingenious" → 2-3 Tokens

> **Why it matters:** Costs and Context Windows are measured in tokens.

Input: ”Hello World”

Tokens: [15496, 2159]

Context Window

The limit of tokens the model can "see" at once.

Let us ask ChatGPT the context window

Try this: https://token-calculator.net/

Quiz: Architecture

Question 3:

What is the key innovation of the Transformer architecture?

A. Processing data sequentially word-by-word.

B. The "Self-Attention" mechanism allowing parallel processing. 

C. Using purely rule-based logic.

Agentic AI Autonomous Systems

The next evolution.

Standard LLM vs. AI Agent

Feature	Standard GenAI/LLM	AI Agent
Behavior	Reactive (Waits for input)	Proactive (Pursues goals)
Capabilities	Content Generation	Tool Use (Search, API, Code)
Loop	Input → Output	Observe → Think → Act → Loop

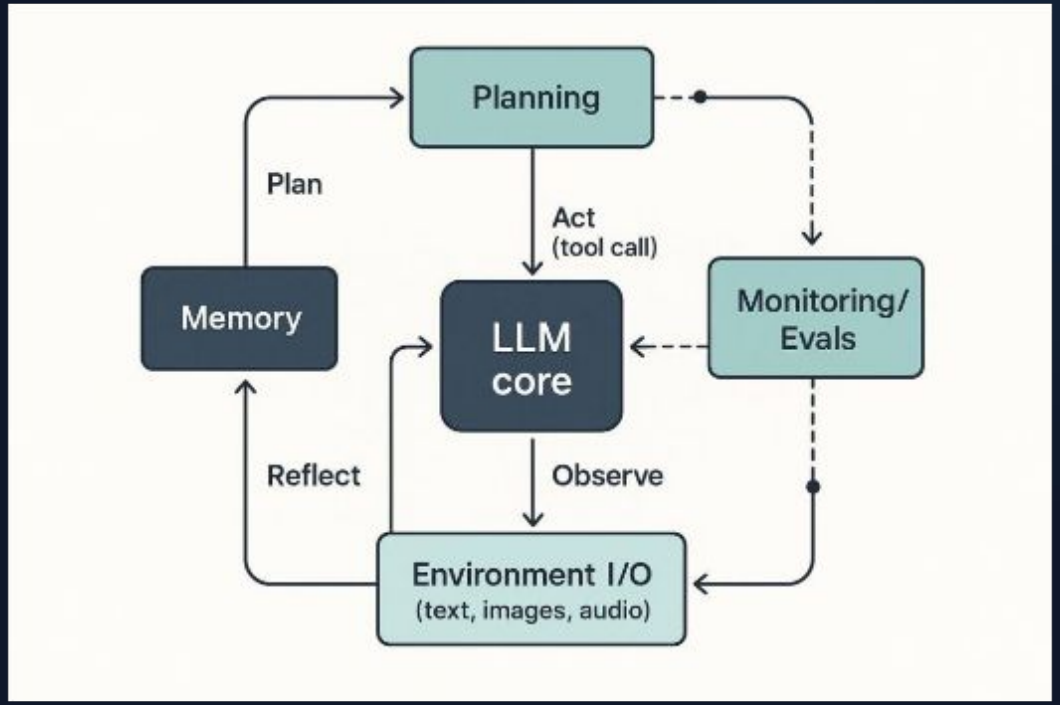
Anatomy of an Agent

An agent is a system, not just a model.

A. The Brain (LLM)

B. Memory (Short term and long term)

C. Tools (Increases Agent capability)



Quick Exercise

Step 1: Open ChatGPT
Step 2: Write
“who is the president of United States? Do not use any tool”

The ReAct Loop

Reasoning + Acting

1. Goal	2. Thought	3. Action	4. Observation
"Book me a flight."	"I need to check prices first."	**Call Tool:** Call_API	Read API result & Repeat.

Why are Agents "Advanced"?

Agents introduce three cognitive layers on top of the standard LLM:

1. Reasoning (Planning)

Chatbots answer immediately. Agents pause to plan: *"To solve X, I must first do Y, then Z."* They break complex goals into steps.

2. Tool Use

Chatbots hallucinate facts. Agents query databases, run code, search the web, and hit APIs to get **grounded truth**.

3. The Loop (Agency)

Chatbots stop after one reply. Agents enter a loop: **Think $\to$ Act $\to$ Observe**. They self-correct if an error occurs.

Chatbot vs. AI Agent

The Chatbot

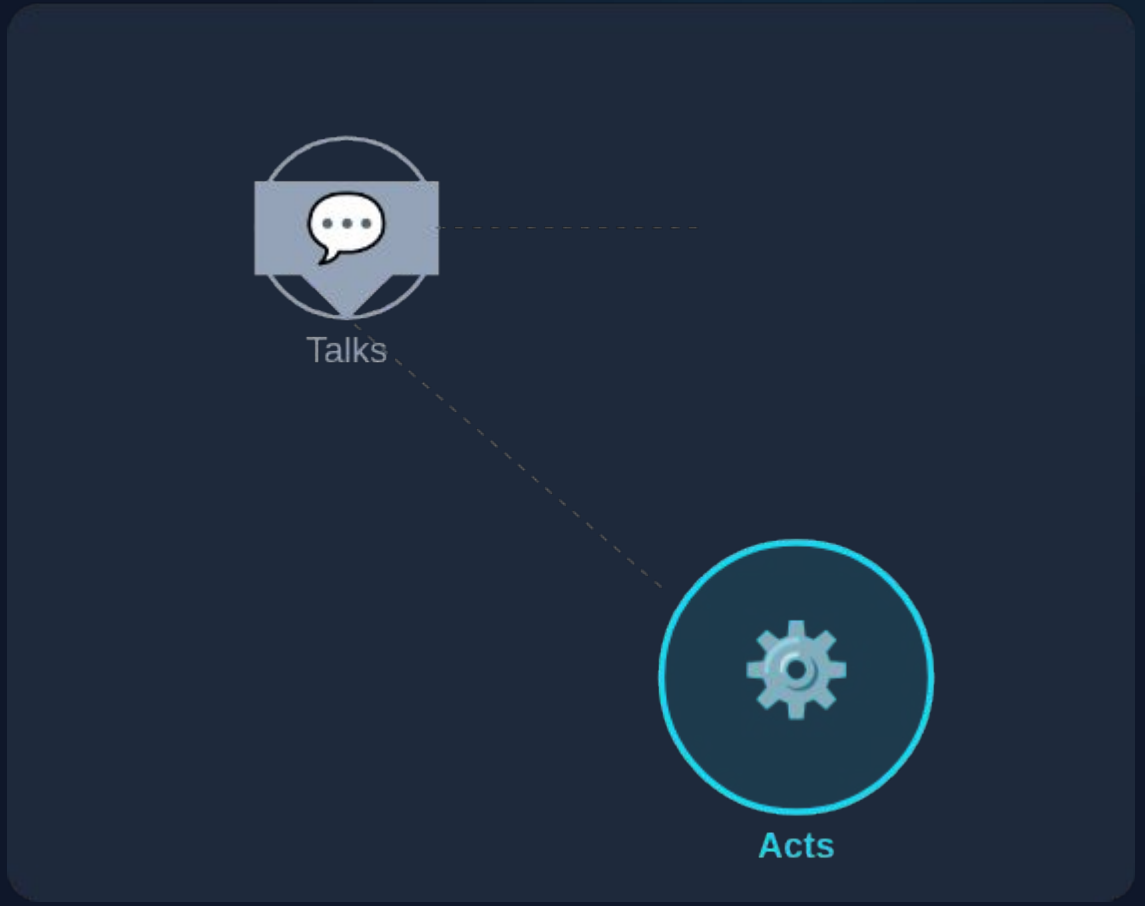
A passive conversationalist. It waits for input, processes text, and replies with text. It is **isolated** inside the chat window.

"I can tell you how to book a flight."

The Agent

An active worker. It has **agency** (the ability to act). It uses tools to change the environment and pursue goals autonomously.

"I have booked your flight and sent the receipt."



Feature Comparison

Feature	Standard Chatbot	AI Agent
Core Function	Text Generation / Conversation	Task Execution / Problem Solving
Environment	Isolated (The Chat Window)	Connected (APIs, Files, Web)
Interaction	Reactive (User prompts, Bot replies)	Proactive (Bot creates own sub-tasks)
Output	Words, Sentences, Paragraphs	Actions, API Calls, Database Changes

The Modern AI Tech Stack

LangChain, LlamaIndex, and Vector DBs.

The ”LLM App” Stack -

Structured data (Tabular data)
unstructured data (Text data, image, audio, video)

1. Model Layer

The "Brain". GPT-4, Claude, Gemini, Llama 3.

2. Orchestration

The "Logic". LangChain. Connecting chains and flows.

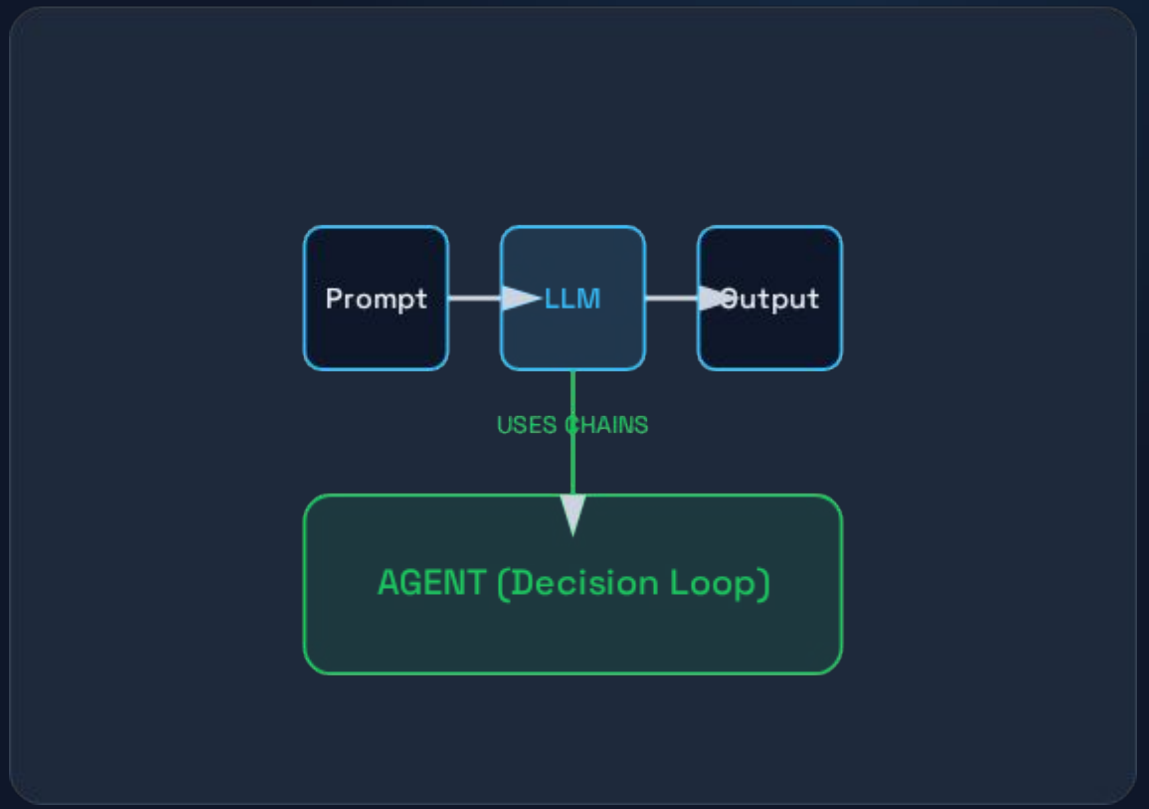
3. Data Layer

The "Memory". LlamaIndex & Vector Databases (Pinecone, Weaviate).

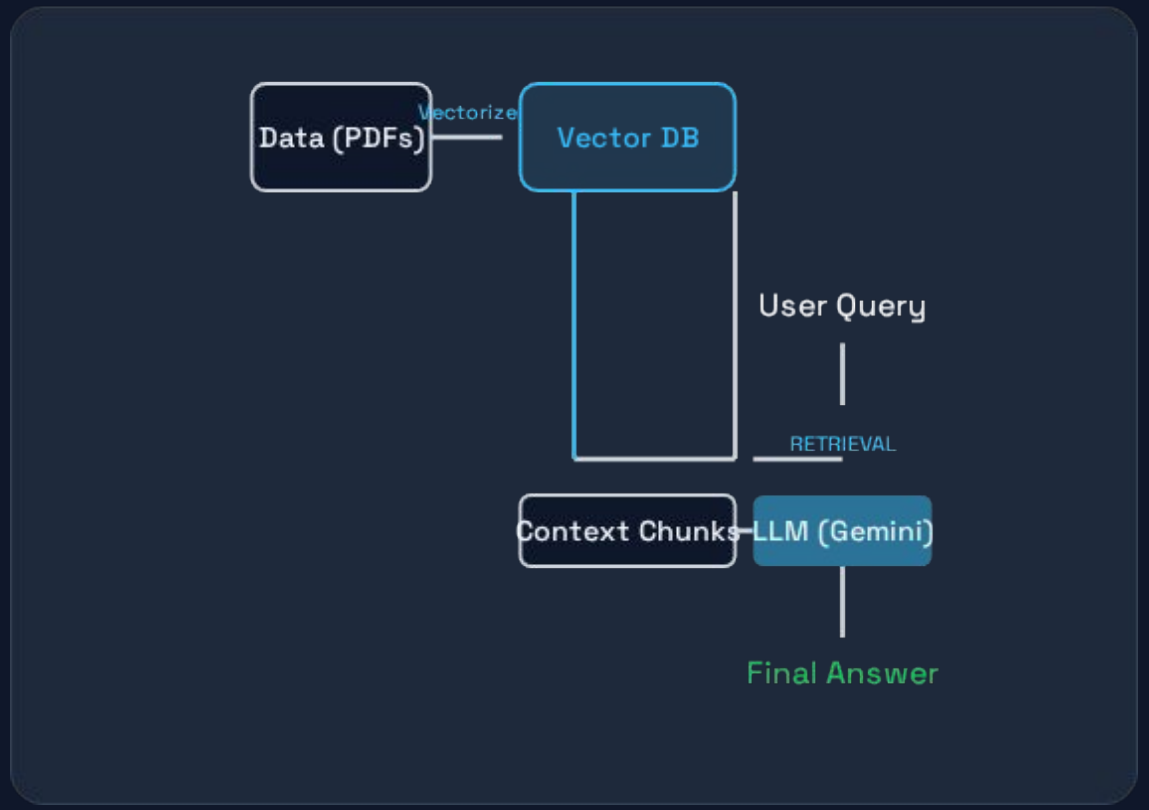
Orchestration: LangChain

LangChain is the industry standard framework for building LLM apps.

- **Chains:** Sequence of calls (Prompt →LLM → Output).
- **Agents:** LLMs that can use tools.
- **Memory:** Managing chat history state.



Data Framework: LlamaIndex



LlamaIndex specializes in **RAG (Retrieval Augmented Generation)**.

It solves the problem: *"How do I let the AI read my private PDFs and SQL data?"*

- Ingest Data
- Index (Vectorize)
- Retrieve Context

Quiz: Agents

Question 4:

What gives an AI Agent the ability to interact with the real world?

A. A larger parameter count.

B. Tools (APIs, Functions).

C. Faster GPUs.

Agent Tools Examples

Web Search

For retrieving real-time information (Stock prices, News).

Python REPL

For precise math and data visualization.

SQL Connector

For querying enterprise business databases.

Final Quiz

Question 5:

Which component is responsible for connecting an LLM to private data?

A. The Orchestrator

B. RAG (Retrieval Augmented Generation)

C. The Tokenizer

Hands-on

Step 1: Choose Your Setup

Local IDE (PyCharm)

Best for: Building real applications, managing complex files, and using `.env` for security.

- Install Python 3.12+
- Create Virtual Env
- Install Libs: pip install openai python-dotenv

Google Colab (Cloud)

Best for: Quick experiments, sharing notebooks, and free GPU access.

- No installation required.
- Runs in browser.
- Install Libs: !pip install openai

Step 2: The Keys to the Kingdom

Getting the Key

1. Go to **platform.openai.com** (or Google (Google AI Studio)/Groq/Anthropic).
2. Sign up / Log in.
3 Click "Create new secret key".
4 **Copy it immediately.** You won't see it again.

.

> ⚠**WARNING:**
> Never commit keys to GitHub. Use a .env file or Colab Secrets.

Step 3: ”Hello LLM”

Let us write code

Step 4: The Response Object

Parsing the Output

The LLM doesn't just return text; it returns a complex JSON object containing usage stats, finish reasons, and metadata.

- choices[0]: The first (and usually only) answer.
- message.content: The actual text reply.
- usage.total_tokens: How much this cost you.

The Future: Multi-Agent Systems

Beyond single agents. Swarms of specialized agents (Coder, Reviewer, Manager) collaborating to solve complex engineering problems.

Q & A

Prompt Engineering Fundamentals

Steering the model effectively.

Anatomy of a Prompt

Persona

"Act as a Senior Python Engineer..."

Task

"Write a script to automate data cleaning..."

Context

"The data is in CSV format, messy, and large."

Format

"Output the code in a markdown block."

Prompting Strategies

Strategy	Description	Use Case
Zero-Shot	No examples provided.	Simple facts, creative writing.
Few-Shot	Providing examples (shots) in prompt.	Complex formatting, specific style copying.
Chain of Thought	"Let's think step by step."	Math, Logic, Reasoning problems.

Quick Hands-on ChatGPT