INTRODUCTION

Agentic AI represents a transformative evolution in artificial intelligence, moving beyond traditional, reactive systems to create autonomous agents capable of independent decision-making, planning, and learning.

These intelligent agents operate through a cycle of perception, reasoning, action, and learning. They gather data from various sources, analyze it to make informed decisions, execute tasks, and continuously refine their strategies based on feedback and outcomes.

As we delve deeper into the capabilities and implications of agentic AI, it becomes clear that this technology is poised to redefine the relationship between humans and machines, offering unprecedented levels of efficiency, adaptability, and innovation.

WHAT IS AGENTIC AI?

Agentic AI is an advanced form of artificial intelligence that enables systems to operate autonomously, making decisions and executing tasks without continuous human guidance. Unlike traditional AI, which typically responds to specific prompts or follows

BASICS OF AGENTIC AI

Unlike traditional AI that passively responds to inputs, AGENTIC AI is proactive. It doesn't just process task – it takes initiative, adapt, and sometimes even improve.
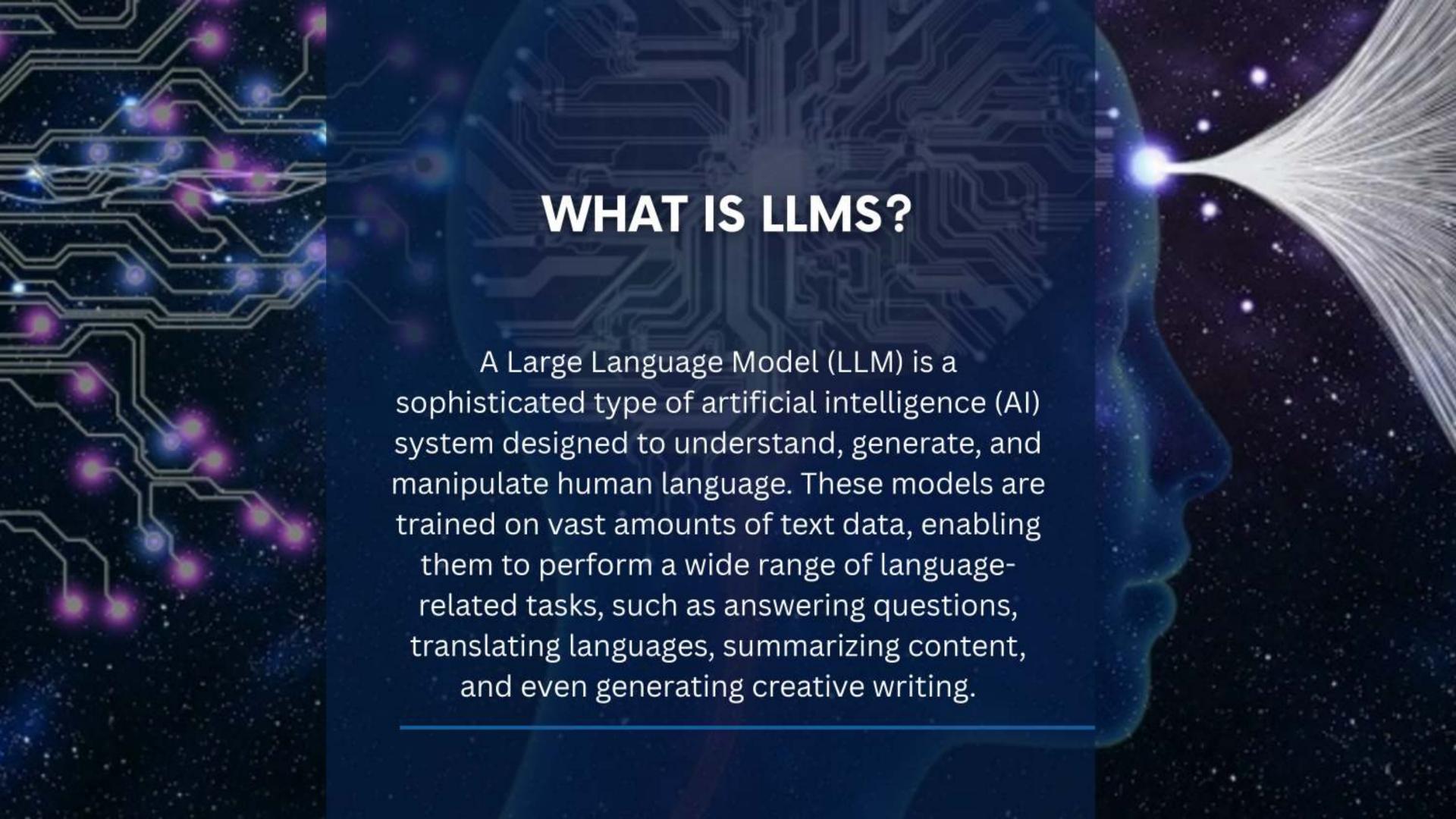
01 AUTONOMY : Acts independently without constant human instructions
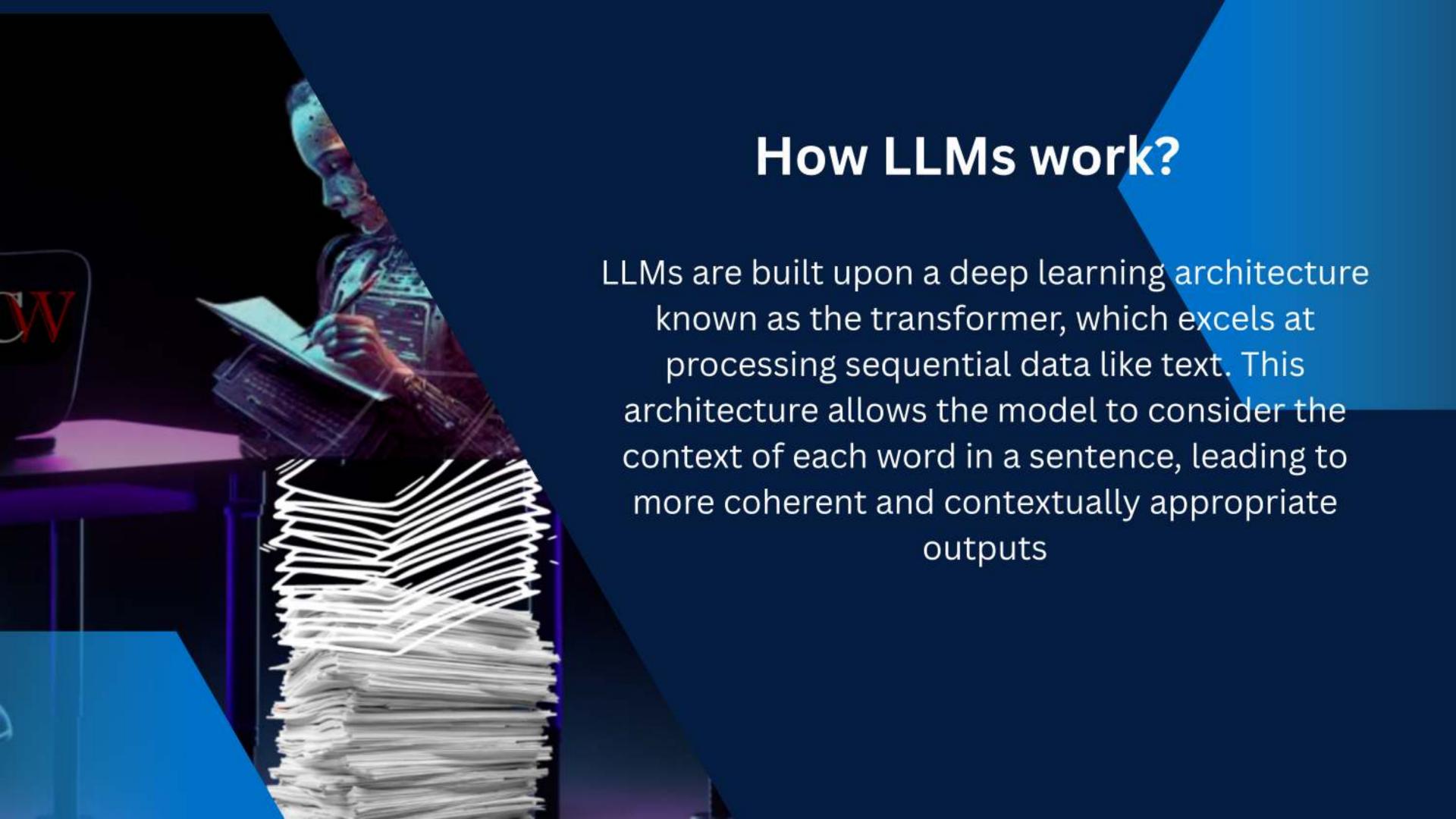
02 GOAL-ORIENTED : Designed to pursue objectives, not just follow programmed rules

03 DESICION- MAKING : Can choose between multiple actions or paths based on context.

04 CONTEXT-AWARENESS : Understands its environment and responds dynamically.

WHAT IS LLMS?

A Large Language Model (LLM) is a sophisticated type of artificial intelligence (AI) system designed to understand, generate, and manipulate human language. These models are trained on vast amounts of text data, enabling them to perform a wide range of language-related tasks, such as answering questions, translating languages, summarizing content, and even generating creative writing.

How LLMs work?

LLMs are built upon a deep learning architecture known as the transformer, which excels at processing sequential data like text. This architecture allows the model to consider the context of each word in a sentence, leading to more coherent and contextually appropriate outputs

Tokenization

LLMs begin by breaking down input text into smaller units called tokens, which can be words or subwords. This process enables the model to handle and analyze text efficiently.

Embedding

Each token is transformed into a numerical vector, capturing its semantic meaning. These embeddings allow the model to process and understand the context of words.

Positional Encoding

Since transformers process tokens simultaneously, positional encoding is added to embeddings to retain the order of words, ensuring the model understands the sequence of the text.

Transformer Architecture

LLMs utilize transformer models comprising encoder and decoder layers. The encoder processes the input sequence, while the decoder generates the output, enabling tasks like translation and text generation.

Self-Attention Mechanism

This mechanism allows the model to weigh the importance of different words in a sequence, capturing relationships and dependencies between words regardless of their position.

Multi-Head Mechanism

By employing multiple attention heads, the model can focus on different parts of the sequence simultaneously, capturing various linguistic features and enhancing understanding.

After attention mechanisms, the data passes through feed-forward networks that apply non-linear transformations, allowing the model to learn complex patterns..

Feed-Forward Neural Networks

LLMs are trained on vast amounts of text data from diverse sources. This extensive training enables them to learn grammar, facts, reasoning abilities, and even some level of world knowledge..

Training on Large Datasets

Post initial training, models can be fine-tuned on specific datasets to specialize in particular tasks, such as medical diagnosis or legal document analysis.

Fine-Tuning

Given a prompt, the LLM predicts the most probable next token iteratively, generating coherent and contextually relevant text until a complete response is formed

Output Generation

WHAT IS GENERATIVE AI

Generative AI is like a creative companion powered by artificial intelligence. It can craft original content it to be text, images, music, or code by learning patterns from existing data. Think of it as an imaginative machine that, when given a prompt, produces something new and unique.

Generative AI

O1: Generates new content

O2: It is cannot use AI tools

Agentic AI

Make decisions and takes actions on it own

Can use tools etc to do searching and complete the given task

Generative AI

Generative AI is a stateless which means it does not remember any previous interactions. It only responds based on the current prompt

Agentic AI

While Agentic AI is stateful they can store data using context and external memory to store it in long-term

What are OpenAI Agents SDK?

The agents SDK provide a tools and APIs that developers can use to build smart AI agents using LLMs autonomous or semi-autonomous systems that leverage OpenAI's models like (ChatGPT-O4, GPT-turbo)

It helps you to create your own AI agents that can :

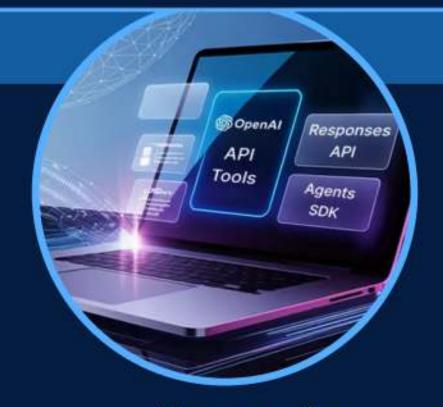**Talk to Users**

Use tools (like APIs , database, etc)

Remember things (Memory)

BENEFITS OF USING THE OPENAI AGENTS SDK

Autonomous Task Completion

Agents can call tools/APIs dynamically like calculators, search , database or custom functions

Memory & Personalization

The SDK includes built-in memory management, so agents can remember facts about users or sessions

Multi-Function Calling

Agents can reason through multiple steps and decide which tool to call and in what order

Modular & Extensible design

You can define your own tools, agents & workflows

Test, Optimized & Maintained by OpenAI

OpenAI's SDK ensures best practice in prompt design, tool calling, and state handling

Web + Local Execution Ready

The SDK is designed to work both locally and in production. Great for chatbots, copilots, automation bots , or internal business assistant