IEEE BANGALORE SECTION

भारतीय विज्ञान संस्थान

IEEE young professionals Bangalore Section

AGENTIC AI SUMMIT – 2025

DEMYSTIFYING AGENTIC AI

How an **AI Agent** Works: The 3-Step Process

1. Perceive

The agent gathers input and information from its digital or physical environment.

2. Think

It defines tasks, analyzes requirements, and proposes potential solutions.

3. Act

The agent executes the chosen solution to accomplish the defined task.

Identifying AI Agents: More Than Just Automation

SIMPLE AUTOMATION: PRE-PROGRAMMED & RIGID

FIXED-TIME SPRINKLER

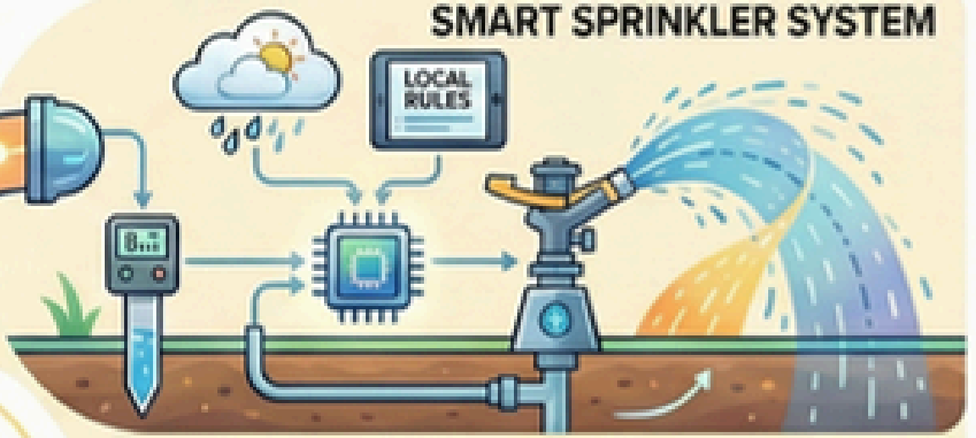
FIXED-TIME SPRINKLER
Follows a rigid, pre-programmed schedule without any external awareness.

MANUAL SPREADSHEET MACRO

MANUAL SPREADSHEET MACRO
Performs a specific, defined task only when directly commanded by a user.

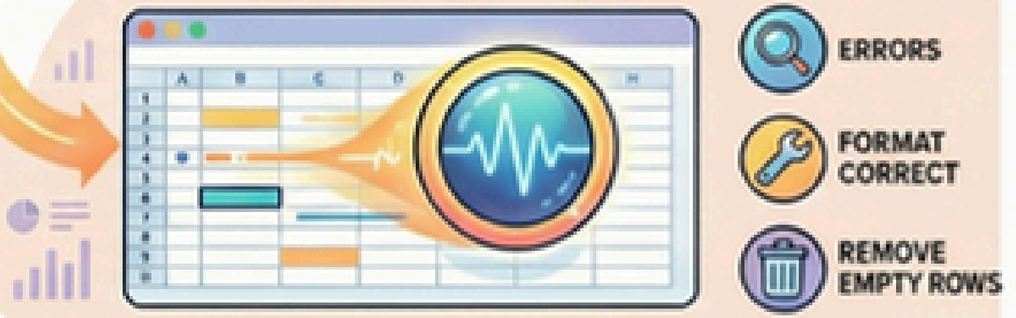
AI AGENTS: AUTONOMOUS & GOAL-ORIENTED

SMART SPRINKLER SYSTEM



SMART SPRINKLER SYSTEM
Uses soil sensors, weather forecasts, and local rules to hit a goal.

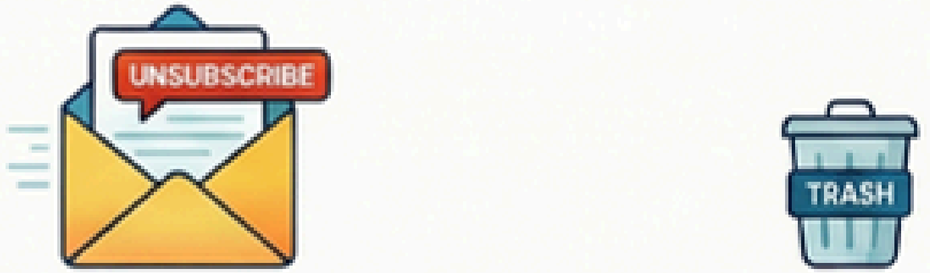
AUTONOMOUS SPREADSHEET AGENT



AUTONOMOUS SPREADSHEET AGENT
Independently scans for errors, corrects formats, and removes empty rows.

Identifying AI Agents: From Simple Automation to Complex Action

SIMPLE AGENTS:
SINGLE-TASK AUTOMATION



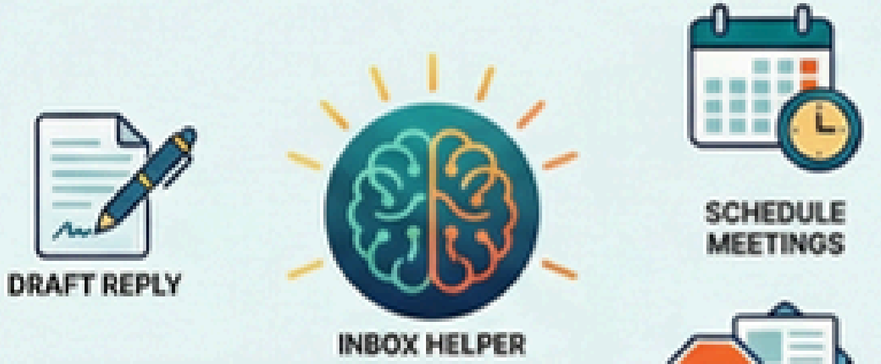
Rule-Based Email Filtering

An email trigger moves any email with "Unsubscribe" in the subject to the trash.

Basic Image Classification

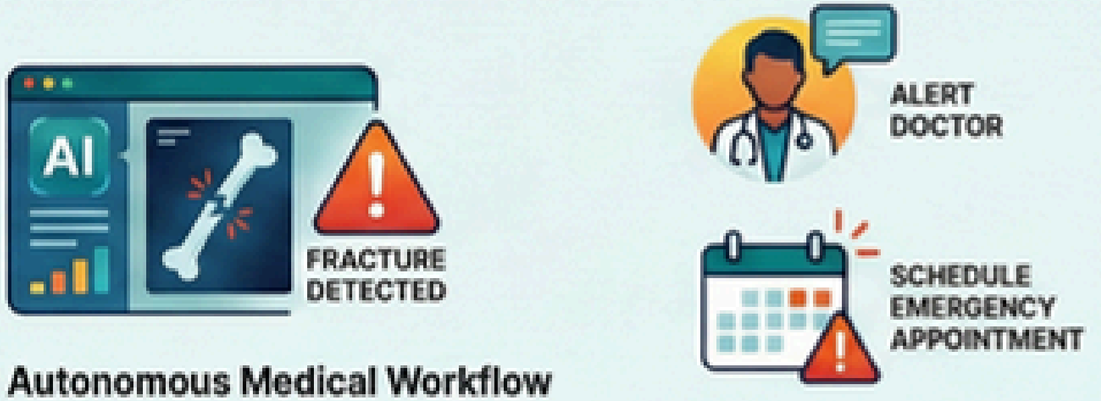
A vision model analyzes a medical image and assigns it a classification label.

ADVANCED AGENTS:
AUTONOMOUS SYSTEMS



Proactive Inbox Management

An "inbox helper" triages mail, drafts replies, schedules meetings, and unsubscribes.

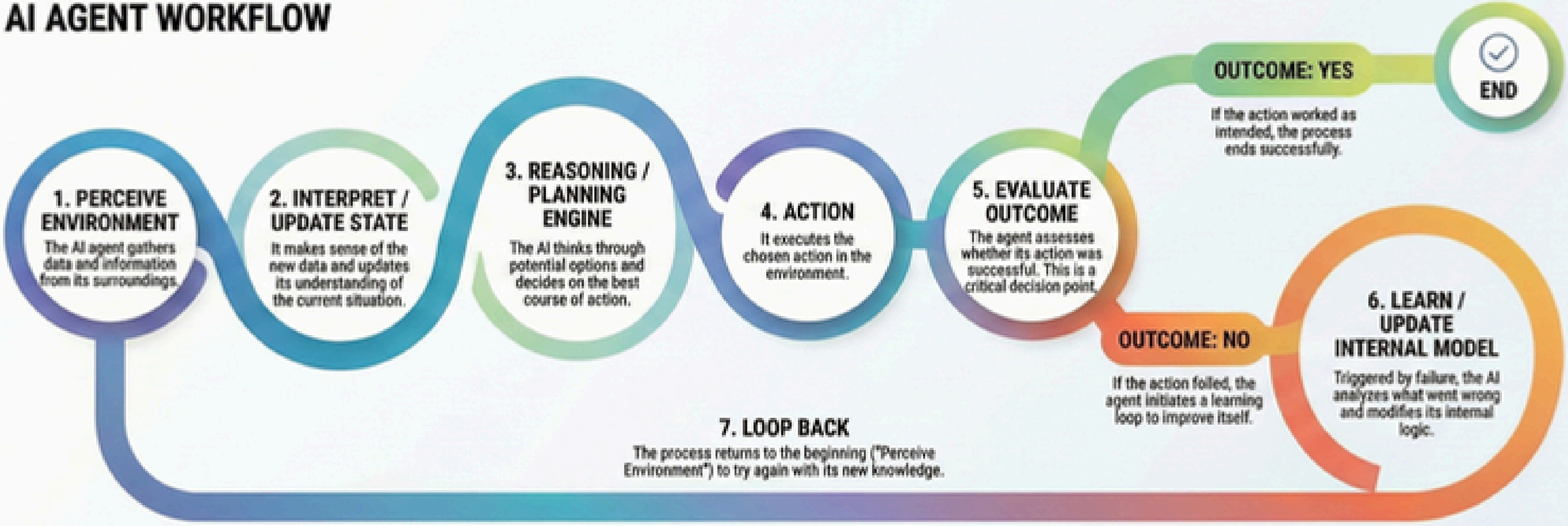


Autonomous Medical Workflow

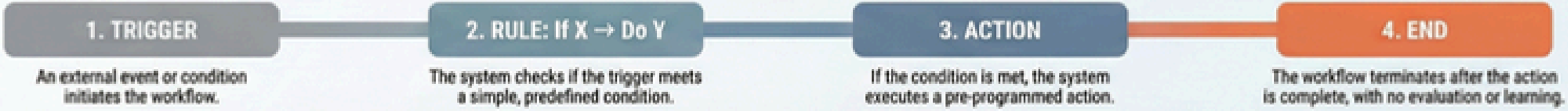
An agent detects a fracture, alerts a doctor, and schedules an emergency appointment.

AI Agent vs. Pure Automation: A Workflow Comparison

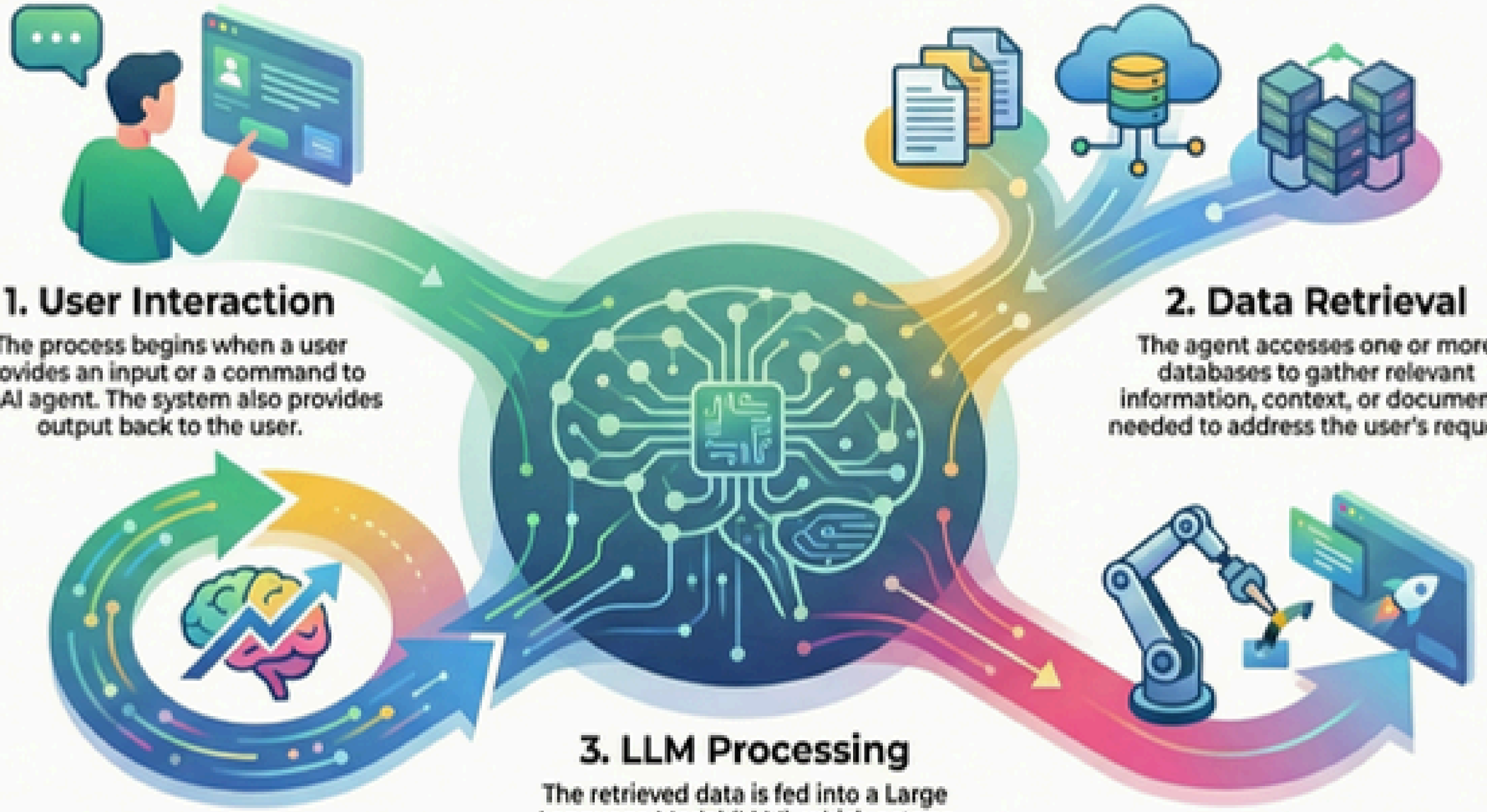
AI AGENT WORKFLOW



PURE AUTOMATION WORKFLOW



High-Level Architecture of an AI Agent



1. User Interaction

The process begins when a user provides an input or a command to the AI agent. The system also provides output back to the user.

2. Data Retrieval

The agent accesses one or more databases to gather relevant information, context, or documents needed to address the user's request.

3. LLM Processing

The retrieved data is fed into a Large Language Model (LLM), which acts as the core reasoning engine or "brain" of the agent.

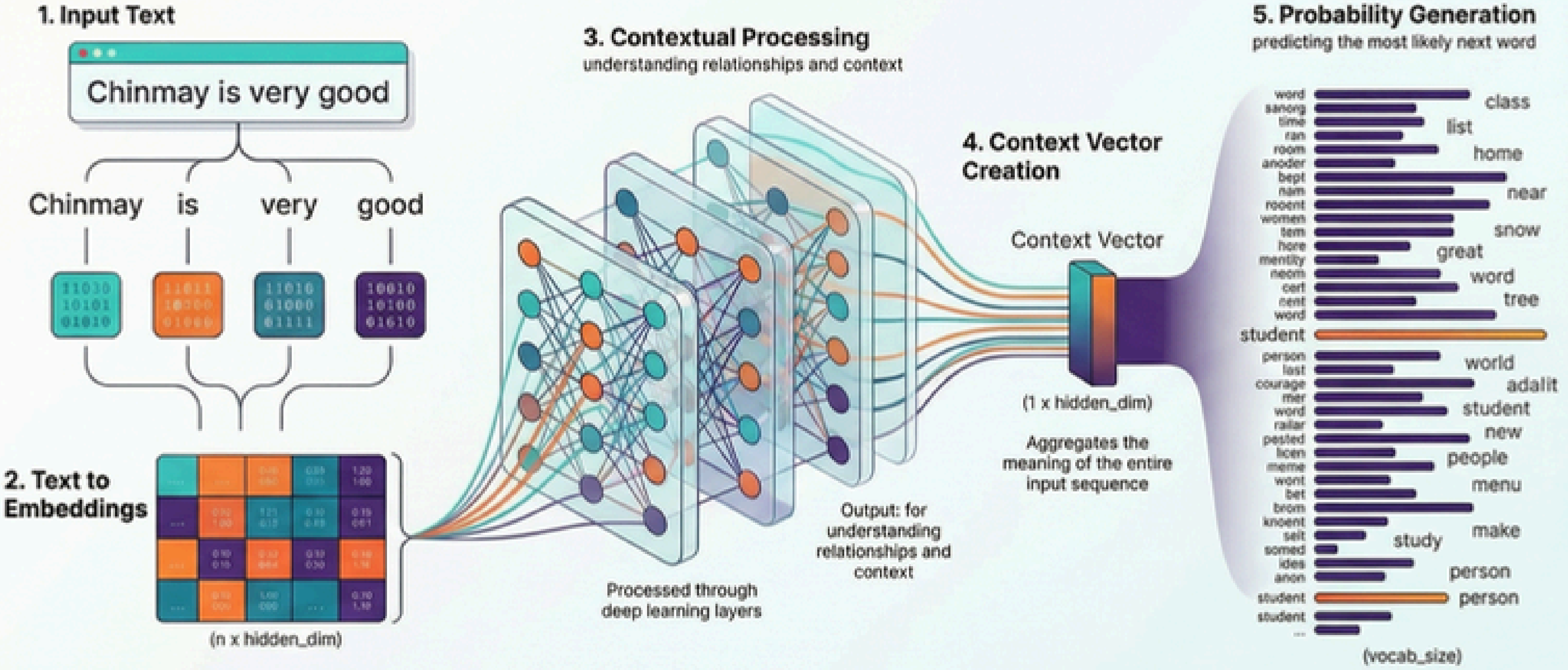
4. Action Execution

Based on the LLM's processing and decision-making, the agent triggers a specific action or task.

5. Feedback Loop

The outcome of the action is fed back to LLM, creating a continuous improvement cycle that allows the agent to learn and adapt.

How Do LLMs Work? A Simplified View



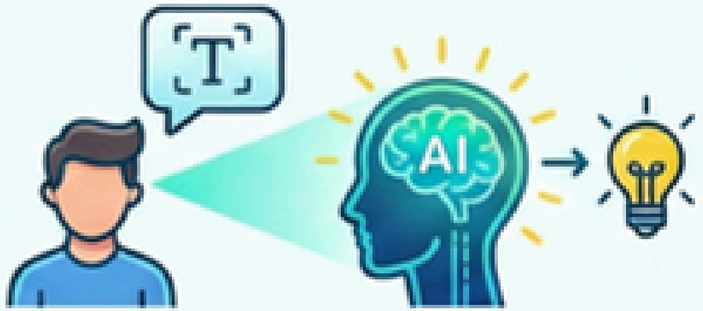
A Simplified Model: This representation has "many many flaws" and is a simplified view for conceptual understanding.

KEY CONCEPTS IN AI INTERACTION: PROMPTING, COT & REASONING

PROMPTING

What is Prompting?

The fundamental act of giving instructions or queries to an AI model to elicit a desired response.



How is it done?

Crafting clear, concise, and effective instructions for an AI.

Why is it important?

Guiding AI behavior and achieving accurate results.

COT PROMPTING

What is COT Prompting?

An advanced technique (Chain-of-Thought) encouraging AI to show its step-by-step thinking process.

How does it work?

Structuring a prompt to ask the AI to "think step-by-step" before providing a final answer.

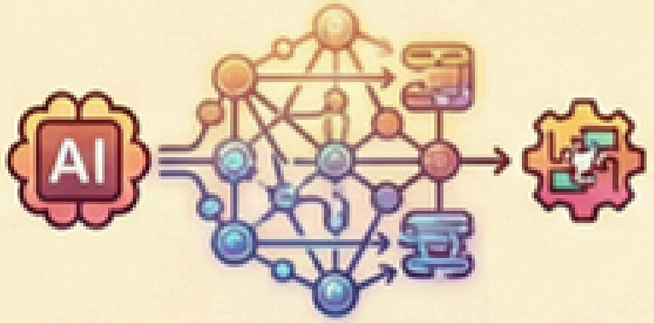
Why is it used?

Improving accuracy on complex problems.

REASONING

What is AI Reasoning?

The ability of an AI to infer, deduce, and make logical connections beyond simply pattern matching.



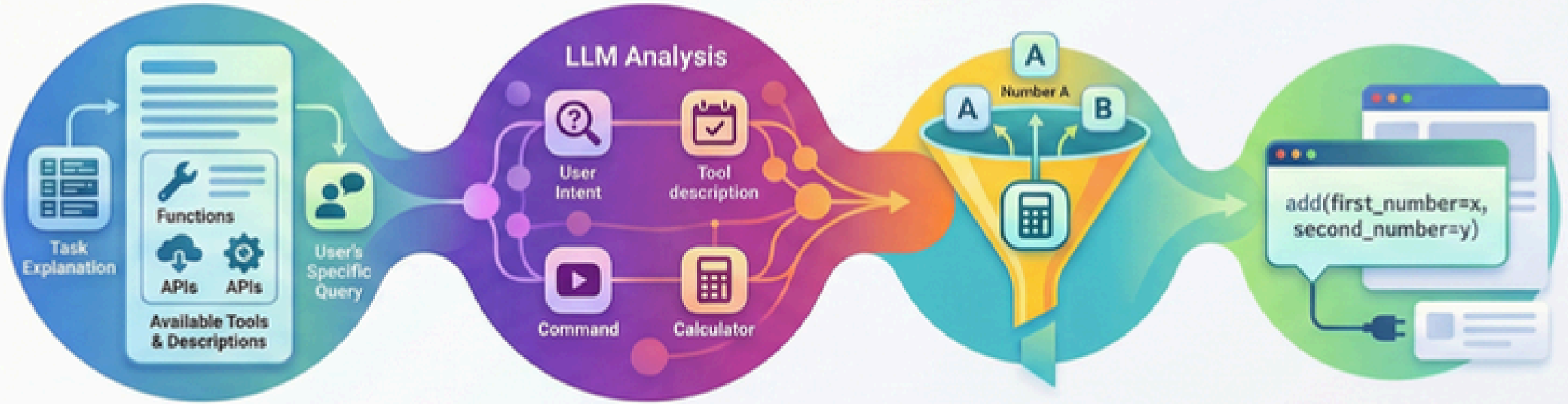
How is it achieved?

Enabled by model architectures and mechanisms that allow for demonstrating reasoning capabilities.

Why does it matter?

Significant for solving complex, multi-step problems and building more sophisticated AI systems.

Tool Determination: How an LLM Chooses the Right Tool



1. A detailed prompt is constructed for the LLM.

This prompt combines the task explanation, a list of available tools with their descriptions, and the user's specific query.

2. The LLM processes the input.

The model analyzes the user's intent and compares it against the descriptions of the provided tools to to find the best match.

3. The LLM makes a decision.

The model's output is the name of the chosen tool along with the necessary arguments extracted or inferred from the user's query.

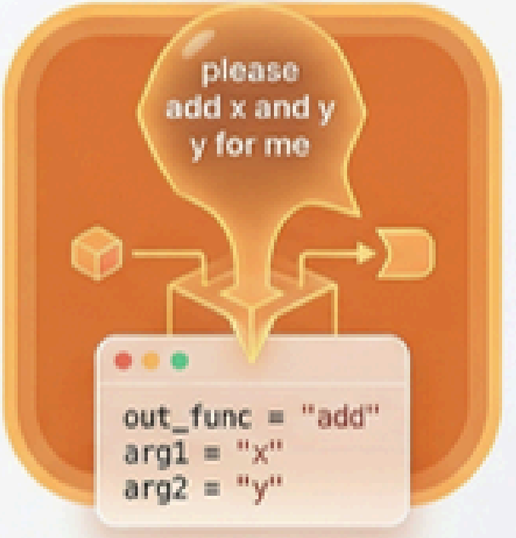
4. A formatted string output is generated.

The chosen tool and arguments are formatted into a final, executable string, such as add(first_number=x, second_number=y).

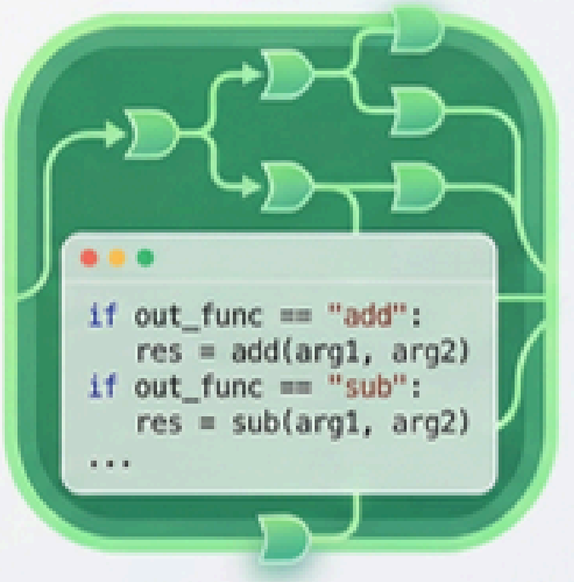
LLM TASK EXECUTION: FROM TEXT TO ACTION

1. GET LLM OUTPUT

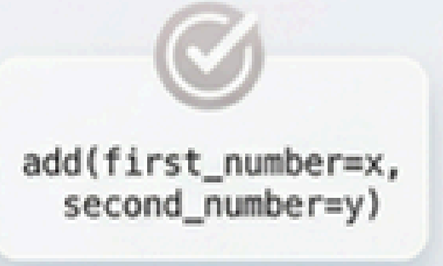
2. POST-PROCESS TO GET REQUIRED FORMAT



3. CALL THE FUNCTION BASED ON OUTPUT



ACHIEVE THE DESIRED OUTPUT!



1. GET LLM OUTPUT

The process starts by receiving the raw output from a Large Language Model, which is typically a text string.

2. POST-PROCESS TO GET REQUIRED FORMAT

The raw string is parsed to extract key information, such as the name of the function to be called and the arguments it requires.

3. CALL THE FUNCTION BASED ON OUTPUT

Conditional logic (e.g., if-statements) is used to match the extracted function name with the actual function, which is then executed with the extracted arguments.

ACHIEVE THE DESIRED OUTPUT!

Following this process successfully translates the LLM's text into a specific, executable function call.