[**The Complete Agentic AI Engineering Course (2025)**](https://tcsglobal.udemy.com/course/the-complete-agentic-ai-engineering-course/)

**Week 1 – FOUNDATION – 26/08/25**

**Day 1 – Make an Agentic Workflow - 26/08/25**

**Topics covered**

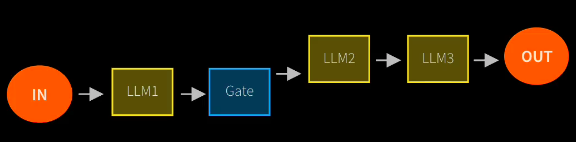
* Git Installation
* Cursor IDE (very similar to VSCode)
* UV (an extremely fast Python package and project manager written in Rust)
* Autonomous AI Agent Demo: Using N8n to control smart home devices
* Windows Setup
* Mac/Linux Setup
* Building your 1st Agentic AI Workflow with OpenAI
  + setting api\_key, endpoint, and version in the .env file
  + loading the values through load\_dotenv(override = True)
  + making an AzureOpenAI object by setting values – api\_version, azure\_endpoint, api\_key
  + creating a list of messages in the OpenAI format – messages = [{“role” : “user”, “content” : “What is 2+2?”}]
  + Finally creating a response by using OpenAI’s function – openai.chat.completions. create(model = “gpt-4o-mini”, messages = messages, stream = False, temperature = 0.3, seed = 24288, top\_p = 0.2, stream\_options = None)
  + response gets fetched – response.choices[0].message.content
  + Finally retrieving back the result in a normal format, as the response got from LLM will be in a Markdown format.

from IPython.display import Markdown, display  
display(Markdown(response))

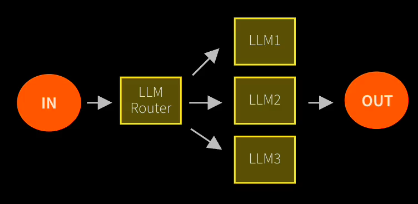
**Day 2 – Agents and Agentic Pattens - 26/08/25**

**Topics covered**

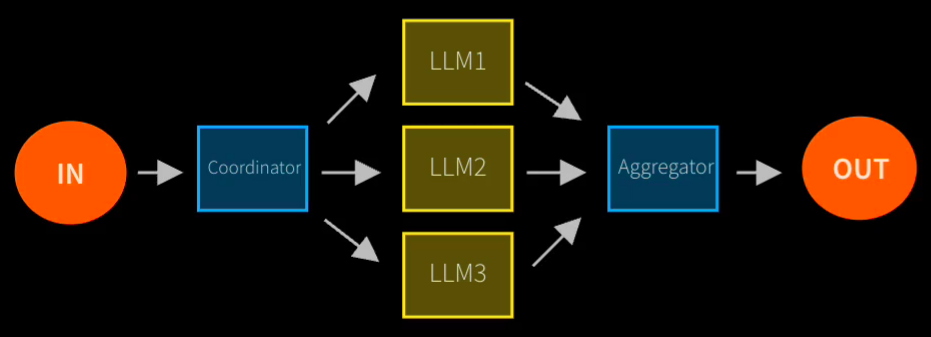
* Ambiguity on what Agents actually are – One simple definition came from Hugging Face – “AI agents are programs where LLM outputs control the workflow.”
* In practice, describe an AI solution that involves any or all of these –
  + Multiple LLM calls
  + LLMs with the ability to use tools
  + An environment where LLMs interact
  + A planner to coordinate activities
  + Autonomy
* Agentic Systems -
  + Anthropic distinguishes two types –
    - Workflows are systems where LLMs and tools are orchestrated through predefined code paths.
    - Agents are systems where LLMs dynamically direct their own processes and tool usage, maintaining control over how they accomplish tasks.
* Workflows – 5 workflow design patterns
  + PROMPT CHAINING – Decompose into fixed sub-tasks



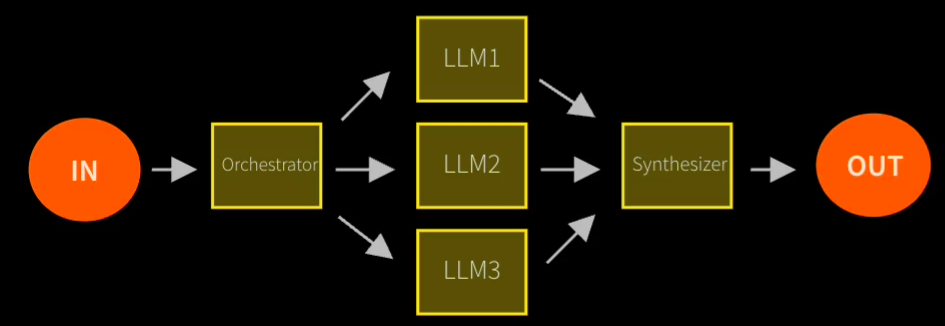
* + ROUTING – Direct an input into a specialized sub-task, ensuring separation of concerns.



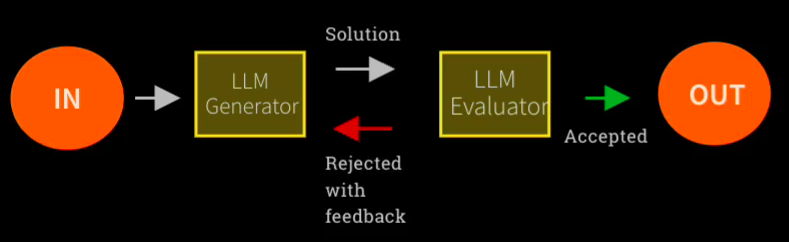
* + PARALLELIZATION – Breaking down tasks and running multiple subtasks consequently.



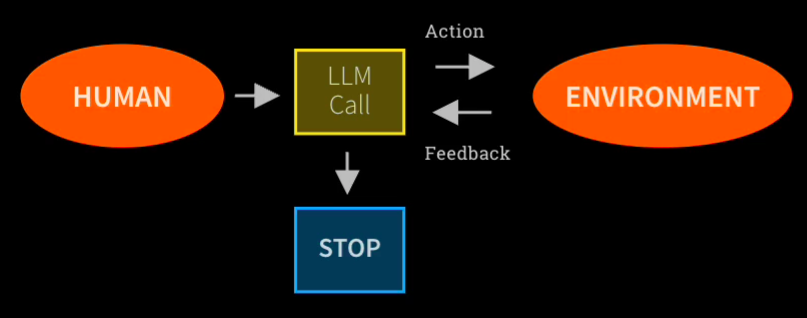
* + ORCHESTRATOR-WORKER – Complex tasks are broken down dynamically and combined.



* EVALUATOR-OPTIMIZER – LLM output is validated by another



* By contrast, Agents:
  + Open-ended
  + Feedback loops
  + No Fixed path



* Risks of Agent Frameworks
  + Unpredictable path
  + Unpredictable output
  + Unpredictable costs
  + Monitor
  + “Guardrails ensure your agents behave safely, consistently, and within your intended boundaries.”

**Day 3 – Orchestrating LLMs – 26/08/25**

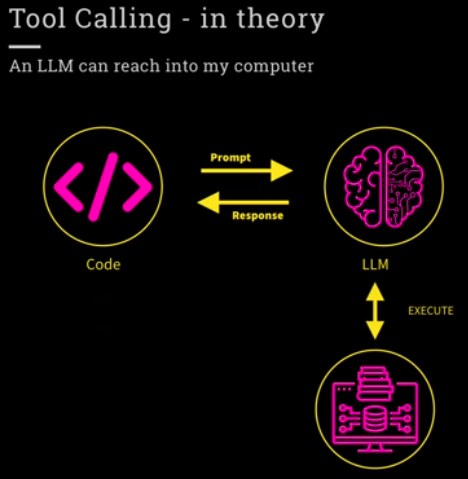
**Topics covered**

* Perform a comparative evaluation of multiple LLMs by issuing standardized test prompts. Examine the variation in their outputs, and subsequently, leverage an additional LLM call to generate an aggregated assessment and ranking of the models based on response quality.

**Day 4 – Autonomy & Tools – 27/08/25**

**Topics covered**

* **Basic overview of what kind of agentic frameworks are currently present in the industry.**
* **Anthropic MCP (Model Context Protocol)**
  + **Owner:** Anthropic, the company behind the Claude LLMs.
  + **What it offers:** MCP is an open standard and framework to connect AI assistants (like LLMs) with external data, tools, and systems smoothly. It acts as a “USB-C port for AI applications,” letting agents securely access files, execute functions, retrieve real-time information, and share contextual data with business or development tools without custom connectors. MCP solves complex integration problems and is becoming widely adopted, even by other major AI labs.
* **OpenAI Agents SDK**
  + **Owner:** OpenAI.
  + **What it offers:** The Agents SDK is a toolkit for building agentic AI applications, enabling developers to orchestrate agents (LLMs) with tools, maintain context/history, delegate tasks (handoffs), apply guardrails/validations, and handle multi-agent coordination. It is Python-centric, easy to learn, and production-ready, with built-in tracing and evaluation for debugging and monitoring agent flows. It exposes robust patterns for task planning, monitoring, error handling, and parallel execution.
* **CrewAI**
  + **Owner:** CrewAI, started by João Moura and a team with deep enterprise/automation experience.
  + **What it offers:** CrewAI is an independent, fast Python framework for building multi-agent teams (“crews”) that collaborate like company departments, each agent having specific roles, autonomy, goals, and tools. It supports high-level simplicity with low-level customization for autonomous, multi-step workflows and offers native support for cloud deployment and enterprise-scale automation. CrewAI emphasizes collaborative intelligence, agent delegation, granular workflow control, and precise event-driven orchestration.
* **LangGraph**
  + **Owner:** LangChain Inc., the creators of LangChain and related agent architectures.
  + **What it offers:** LangGraph is a low-level agent orchestration framework, supporting stateful, long-running agents and customizable cognitive architectures. It lets developers build single or multi-agent flows, add robust moderation/quality controls, integrate human review and oversight, and templatize agent logic for production. LangGraph is widely used for scalable development, debugging, human-in-the-loop collaboration, workflow memory, and deploying reliable agent systems.
* **AutoGen**
  + **Owner:** Microsoft Research.
  + **What it offers:** AutoGen is an open-source framework for designing and orchestrating teams of AI agents that work together to solve complex tasks. It boasts multi-agent architecture, customizable agents, code generation/execution, human-in-the-loop adaptability, flexible workflows, advanced observability (debugging/tracing), and modular extensionality. AutoGen supports asynchronous communication, dynamic collaboration, distributed setups, strong type safety, and cross-language agent interoperability.
* Resources vs. Tools: Two Ways to Enhance LLM Capabilities in Agentic AI
  + Resources
    - We can provide an LLM with resources to improve its expertise
    - Basically, this means shoving data relevant to the question into the prompt
    - There are techniques like RAG to get really smart at picking relevant content
  + Tools
    - Give an LLM the power to carry out actions like – query a database or message other LLMs
    - Tool calling - in theory



* + - Tool calling - in practice

