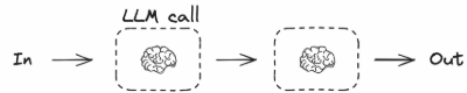
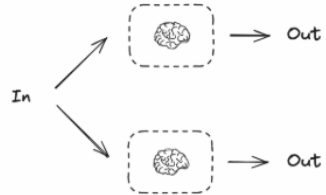


Workflows

Prompt Chaining

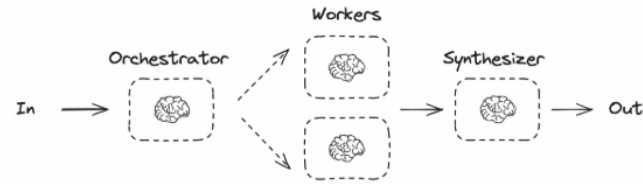


Parallelization

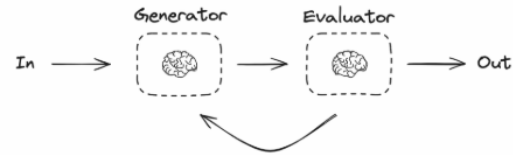


LLM is embedded in predefined code paths

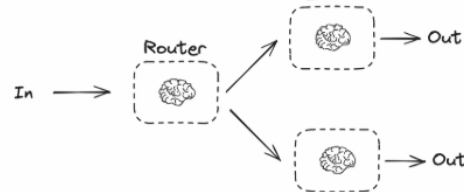
Orchestrator-Worker



Evaluator-optimizer

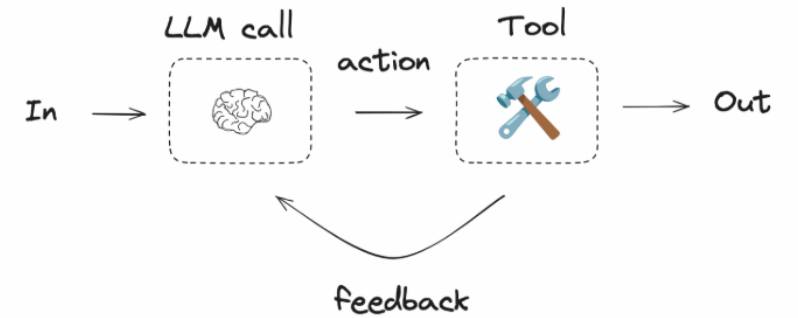


Routing



LLM directs control flow through predefined code paths

Agent



LLM directs its own actions based on environmental feedback

Workflow Type	Description	Ideal Use Cases
Prompt Chaining	Sequential execution where each LLM call processes the output of the previous.	Tasks that can be decomposed into fixed, linear steps.
Parallelization	Multiple LLMs operate concurrently on independent subtasks.	Tasks benefiting from speed or diverse perspectives, like content generation.
Routing	Input is classified and directed to specialized follow-up tasks.	Tasks requiring distinct handling based on input type, such as content categorization.
Orchestrator-Worker	A central orchestrator delegates dynamic tasks to multiple specialized workers.	Complex tasks with unpredictable subtasks, such as code generation or dynamic content creation.
Evaluator-Optimizer	An iterative loop where outputs are evaluated and refined based on feedback.	Tasks requiring iterative improvement, like joke generation or content refinement.
Agent	An autonomous LLM uses tools based on environmental feedback in a loop.	Open-ended problems where the number of steps can't be predetermined.
Multi-Agent Collaboration	Multiple agents share a common workspace, collaborating by reading and writing to it.	Scenarios requiring agents to build upon each other's outputs, like research and content creation.
Agent Supervisor	A supervisory agent manages and routes tasks to subordinate agents.	Hierarchical task management where oversight and coordination are essential.
Hierarchical Agent Teams	Nested agent structures where agents manage sub-agents, forming a hierarchy.	Large-scale, complex workflows requiring modularity and scalability.