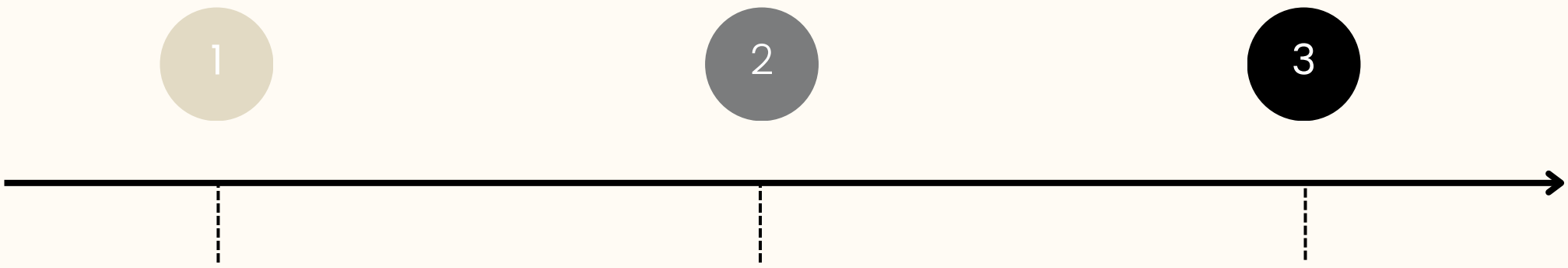
AGENTIC SYSTEMS DESCRIPTION

When and When not to use agents?

- Start with the simplest solution possible - This could be:
  -Optimizing signle LLM calls with retrieval and in-context learning
  -Not using any agentic system
- Only increasing complexity when needed.
- Avoid building agentic systems unless their tradeoff of higher latency and cost for better task performance is justified.
- Use workflows for predictable and consistent execution of well-defined tasks.
- Choose agents when flexibility and model-driven decision-making at scale are required.

Increasing Complexity



Building Block: The augmented LLM

This basic building block is an LLM enhanced with augmentations such as retrieval, tools, and memory

Workflows

Workflows are systems where LLMs and tools are orchestrated through predefined code paths.

Agents

Agents are systems where LLMs dynamically direct their own processes and tool usage, maintaining control over how they accomplish tasks.

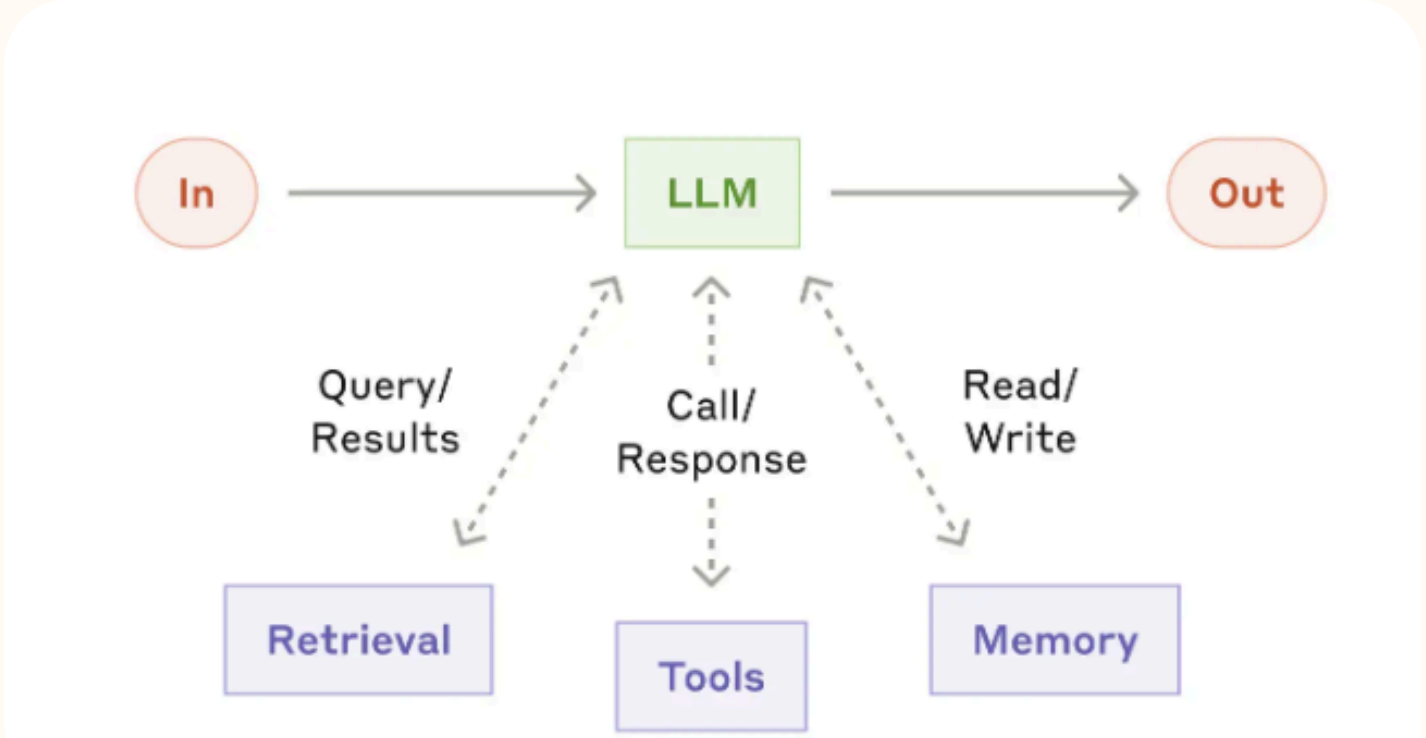
Source: Anthropic - Building Effective Agents

Hanane Dupouy

AGENTIC SYSTEMS DESCRIPTION

Building Block:
The augmented LLM

Where an LLM is augmented by various features such as Retrieval, Tools and Memory.



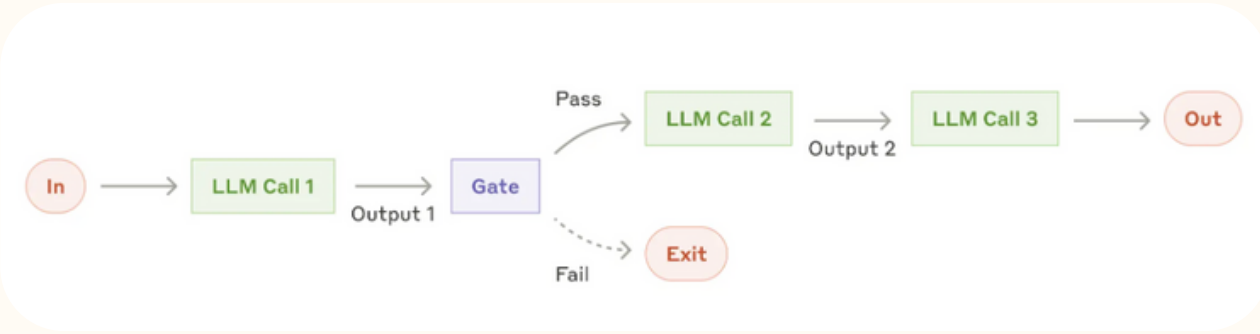
Source: Anthropic - Building Effective Agents

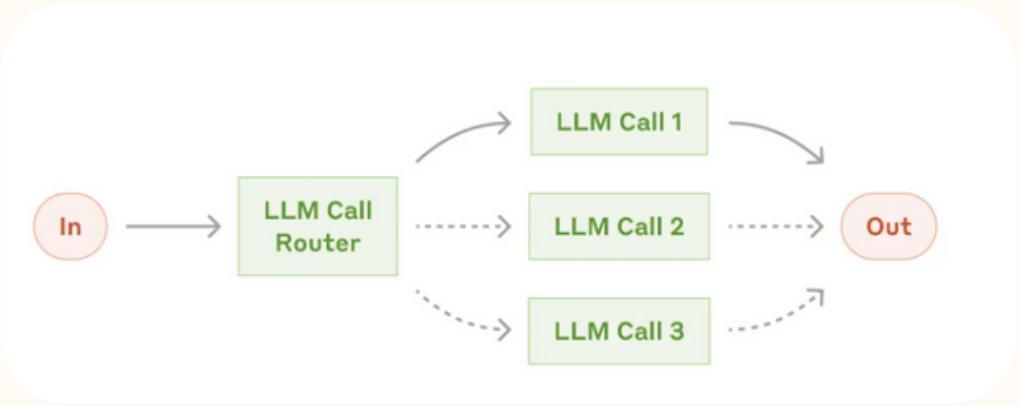
AGENTIC SYSTEMS DESCRIPTION

WORKFLOW

Prompt Chaining

- Prompt chaining breaks a task into a sequence of steps.
- Each LLM call processes the output of the previous step.
- Programmatic checks can be added to intermediate steps for validation.
- Use "gates" to ensure the process remains on track.
- This approach helps maintain accuracy and control in multi-step workflows.





Routing

- Routing identifies an input type and assigns it to a specific follow-up task.
- This approach ensures tasks are handled separately and efficiently.
- It facilitates the development of prompts tailored to specific needs.
- Without routing, optimizing for one input type can negatively impact performance on others.
- This approach ensures tailored handling for different input types, improving overall performance.

Source: Anthropic - Building Effective Agents

Hanane Dupouy

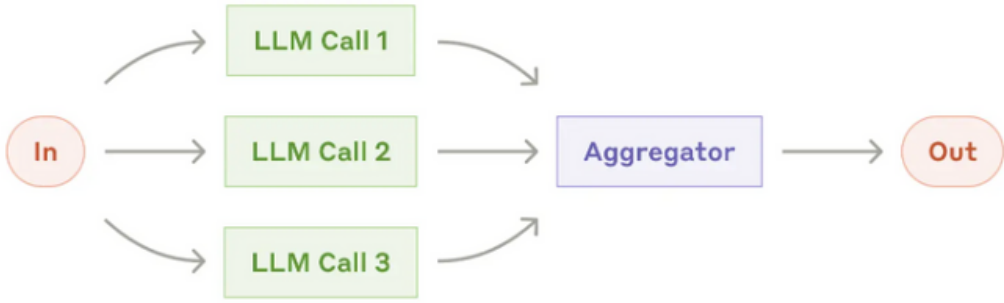
AGENTIC SYSTEMS DESCRIPTION

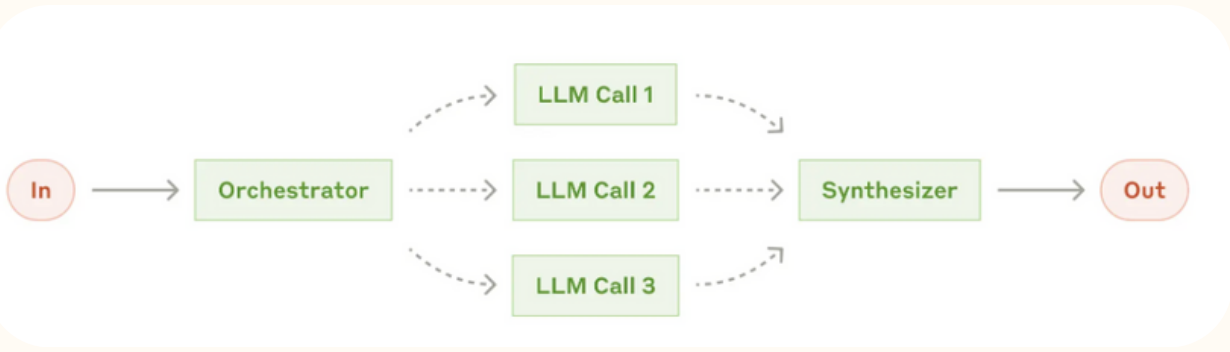
WORKFLOW

Parallelization

- LLMs can perform tasks simultaneously, with outputs aggregated programmatically.

- There are 2 key variations in parallelization:
  1. **Sectioning**: Splits a task into independent subtasks that are executed in parallel.
  2. **Voting**: Executes the same task multiple times to generate diverse outputs.





Orchestrator-Workers

- The orchestrator-workers workflow uses a central LLM to manage tasks dynamically.
- The central LLM breaks down tasks into smaller components.
- These components are delegated to worker LLMs for execution.
- The central LLM synthesizes the results from the worker LLMs.

Evaluator-optimizer

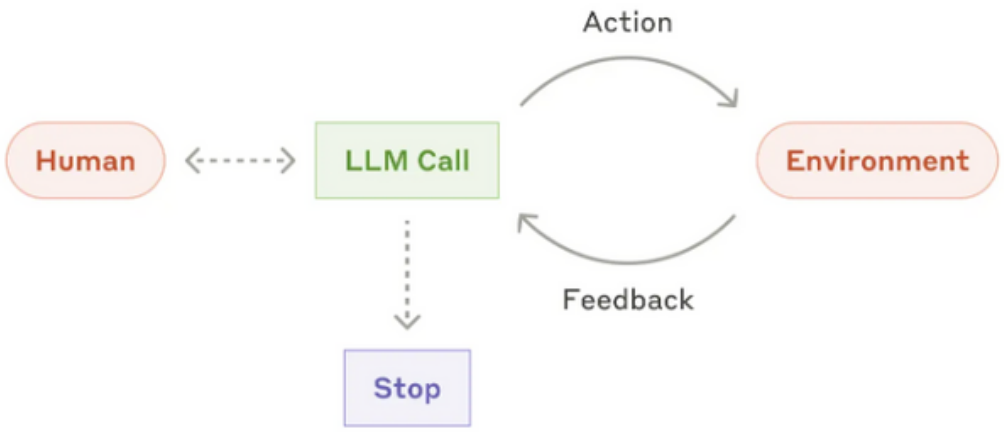
- One LLM call generates a response.
- Another LLM evaluates the response and provides feedback.
- This process operates in a continuous loop for iterative improvement.

Source: Anthropic - Building Effective Agents

Hanane Dupouy

AGENTIC SYSTEMS DESCRIPTION

Agents



Agents are becoming more prevalent as LLMs advance in understanding complex inputs, reasoning, planning, reliable tool usage, and error recovery.

Agents begin tasks with either a command or an interactive discussion with the user.

Once tasks are clear, agents operate independently, with the option to seek further user input when needed.

During execution, agents rely on “ground truth” (e.g., tool results or code execution) to evaluate progress.

Agents may pause for human feedback at checkpoints or when facing blockers, with tasks ending upon completion or when predefined stopping conditions are met.

While capable of handling complex tasks, agents are often implemented simply as LLMs using tools based on environmental feedback in a loop.

Clear and thoughtful design of toolsets and their documentation is essential for effective agent functionality.

- Use agents for open-ended tasks with unpredictable steps or no fixed path.
- Ideal for scaling tasks in trusted environments requiring autonomy.
- Higher costs and error risks demand sandbox testing and guardrails.

Source: Anthropic - Building Effective Agents

Hanane Dupouy

AGENTIC SYSTEMS DESCRIPTION

Sign up for the waitlist to get notified when the AI Agents in Finance course launches!

https://machinelearning-basics.com/ai-agent-in-finance/

Hanane Dupouy