What is an agent?

While conventional software enables users to streamline and automate workflows, agents are able to perform the same workflows on the users' behalf with a high degree of independence.

> Agents are systems that **independently** accomplish tasks on your behalf.

A workflow is a sequence of steps that must be executed to meet the user's goal, whether that's resolving a customer service issue, booking a restaurant reservation, committing a code change, or generating a report.

Applications that integrate LLMs but don't use them to control workflow execution—think simple chatbots, single-turn LLMs, or sentiment classifiers—are not agents.

More concretely, an agent possesses core characteristics that allow it to act reliably and consistently on behalf of a user:

01 It leverages an LLM to manage workflow execution and make decisions. It recognizes when a workflow is complete and can proactively correct its actions if needed. In case of failure, it can halt execution and transfer control back to the user.

---

02 It has access to various tools to interact with external systems—both to gather context and to take actions—and dynamically selects the appropriate tools depending on the workflow's current state, always operating within clearly defined guardrails.

When should you build an agent?

Building agents requires rethinking how your systems make decisions and handle complexity. Unlike conventional automation, agents are uniquely suited to workflows where traditional deterministic and rule-based approaches fall short.

Consider the example of payment fraud analysis. A traditional rules engine works like a checklist, flagging transactions based on preset criteria. In contrast, an LLM agent functions more like a seasoned investigator, evaluating context, considering subtle patterns, and identifying suspicious activity even when clear-cut rules aren't violated. This nuanced reasoning capability is exactly what enables agents to manage complex, ambiguous situations effectively.

As you evaluate where agents can add value, prioritize workflows that have previously resisted automation, especially where traditional methods encounter friction:

01	**Complex decision-making:**	Workflows involving nuanced judgment, exceptions, or context-sensitive decisions, for example refund approval in customer service workflows.
02	**Difficult-to-maintain rules:**	Systems that have become unwieldy due to extensive and intricate rulesets, making updates costly or error-prone, for example performing vendor security reviews.
03	**Heavy reliance on unstructured data:**	Scenarios that involve interpreting natural language, extracting meaning from documents, or interacting with users conversationally, for example processing a home insurance claim.

Before committing to building an agent, validate that your use case can meet these criteria clearly. Otherwise, a deterministic solution may suffice.