

4 Types of Agentic Systems

AND WHEN TO USE WHAT ?



Table of Contents

- 1. Introduction**
- 2. The Tool-Augmented LLM**
- 3. Rule-Based Systems/Agents**
- 4. Workflow Agents**
- 5. Semi-Autonomous Agents**
- 6. Autonomous Agents**
- 7. Examples**
- 8. How to Decide What to Build**

Introduction

Hi there,

Not all agents are built the same, and not all problems need highly autonomous systems.

In this lesson, we'll walk through [four types of agentic systems](#), using a simple but powerful lens:

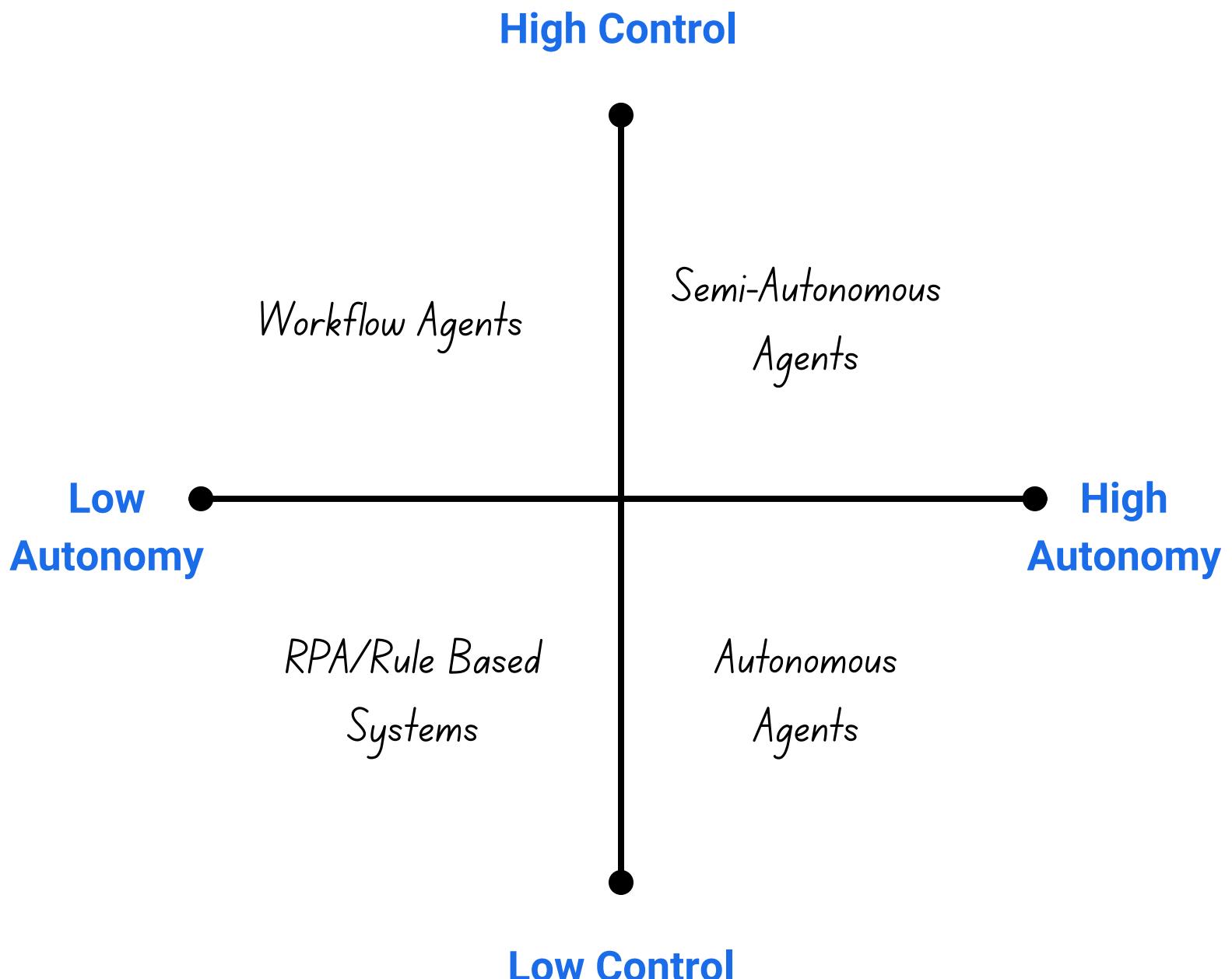
How much autonomy does the agent have?

How much control does the human or system retain?



This balance impacts how the system behaves, how you evaluate it, and what infrastructure you need to build.

Introduction



The Tool-Augmented LLM

At the core of most modern agents is an LLM acting as the brain of the system.

Throughout this course, we use the term LLM to refer broadly to generative AI models – not just text-only models.

On its own, it can generate content, but to turn it into an agent, you augment it with:

- **Tools** → APIs, functions, databases it can call
- **Planning** → The ability to break a goal into multiple steps
- **Memory** → So it can track past actions and outcomes
- **State and Control Logic** → To know what's done, what failed, and what to do next

When connected to these components, the LLM becomes more than a chatbot. It becomes a goal-driven system that can reason, take action, and adapt. But depending on how much you trust it to act without supervision, you end up with different types of agents.

1. Rule-Based Systems/Agents

Low Autonomy, Low Control

These systems don't use LLMs at all. They're built with traditional if-this-then-that logic. Every decision path is manually scripted. There's no reasoning or learning. Rule-based agents have existed long before the LLM era.

What problems do they solve?

Well-structured, repetitive tasks with fixed inputs and outputs.

Examples:

- Automatically approve reimbursements under a fixed amount
- Rename files in a folder based on filename patterns
- Copy data from Excel sheets into form fields

1. Rule-Based Systems/Agents

Pros:

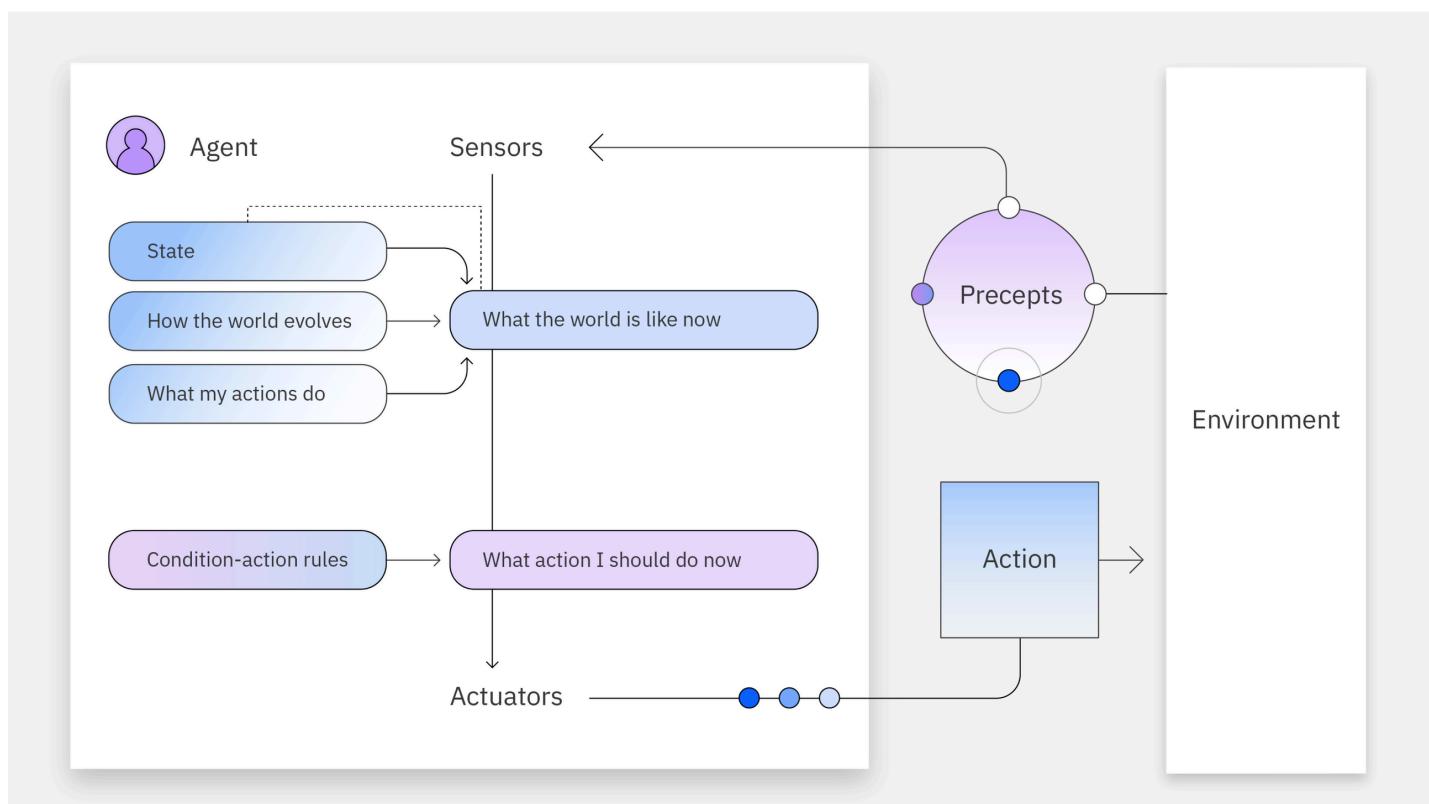
Fast, auditable, predictable

Cons:

Brittle to change, can't handle ambiguity

Best used when:

You know all the conditions ahead of time and there's no need for flexibility.



2. Workflow Agents

Low Autonomy, High Control

This is often the first step for enterprises introducing LLMs into their workflows.

Here, the LLM enhances an existing workflow but doesn't execute actions independently. A human stays in control.

What problems do they solve?

Repetitive tasks that benefit from natural language understanding, summarization, or generation, but still need human decision-making.

Examples:

- Suggesting first-draft responses in a support tool like Zendesk
- Generating summaries of meeting transcripts
- Translating natural language queries into structured search inputs for BI dashboards

2. Workflow Agents

Pros:

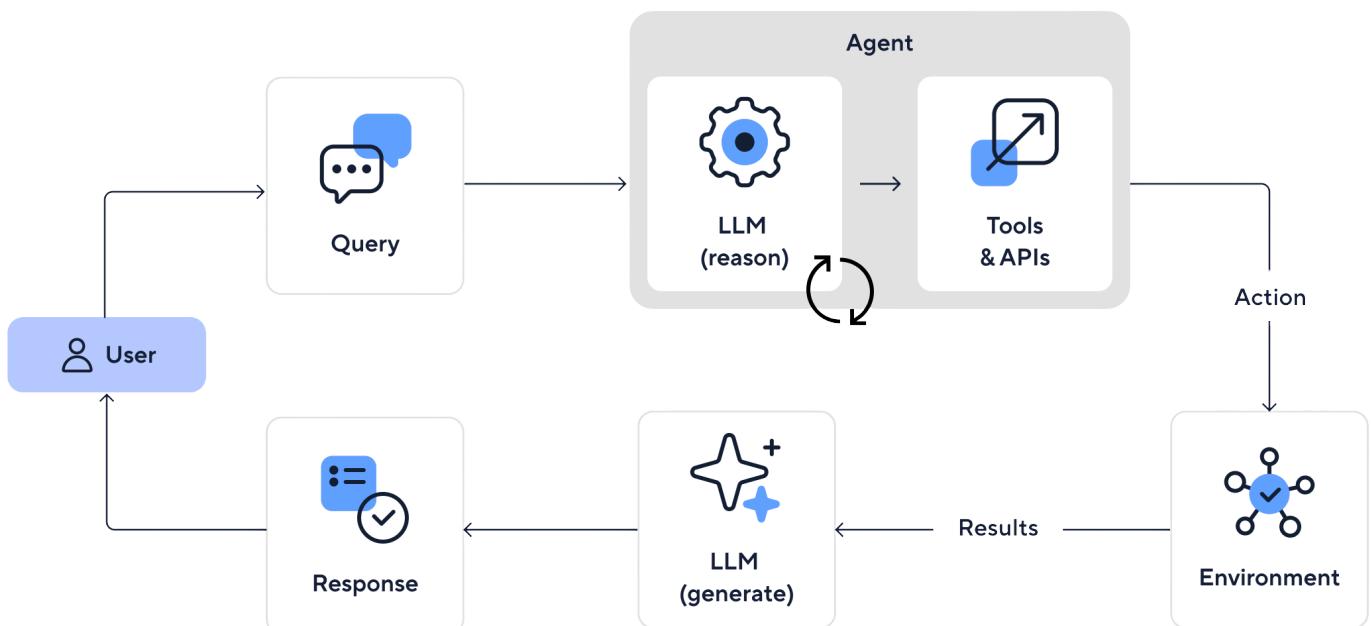
Easy to deploy, low risk, quick value

Cons:

Can't execute or plan, limited end-to-end value

Best used when:

You want to augment your team's productivity without giving up oversight.



3. Semi-Autonomous Agents

Moderate to High Autonomy, Moderate Control

These are true agentic systems. They not only understand tasks but can plan multi-step actions, invoke tools, and complete goals with minimal supervision. However, they often operate with some constraints or monitoring built in.

What problems do they solve?

Multi-step workflows that are well-understood but too tedious or time-consuming for humans.

Examples:

- A lead follow-up agent that drafts, personalizes, and sends emails based on CRM data, while logging results
- A document automation agent that extracts details from contracts and updates internal systems
- A research agent that pulls data from multiple sources, compares findings, and sends a structured report

3. Semi-Autonomous Agents

Pros:

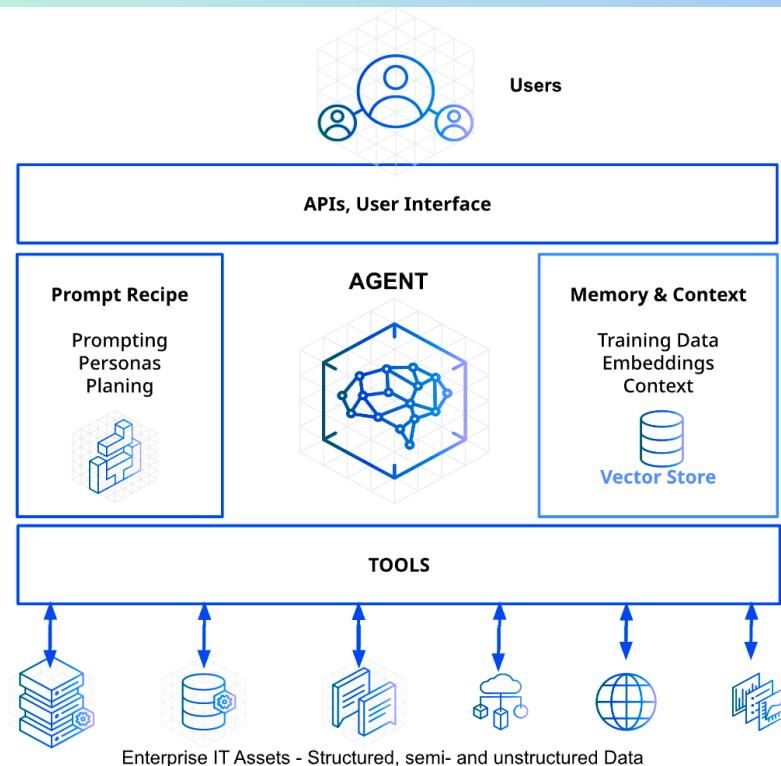
Automates complex workflows, saves time, higher ROI

Cons:

Needs infrastructure (planning, memory, tool calling), harder to test

Best used when:

You want to augment your team's productivity without giving up oversight.



4. Autonomous Agents

High Autonomy, Low Control

These agents are fully goal-driven. You give them a broad objective, and they figure out what to do, how to do it, when to retry, and when to escalate. They act independently, often across systems and over time.

What problems do they solve?

High-effort, async, or long-running tasks that span multiple systems or steps and don't need constant human input.

Examples:

- A competitive research agent that pulls data over days, summarizes updates, and generates weekly insight briefs
- An ops automation agent that detects issues in pipelines, diagnoses root causes, and files tickets with suggested fixes
- A testing agent that autonomously runs product flows, logs results, and suggests new edge-case scenarios

4. Autonomous Agents

Pros:

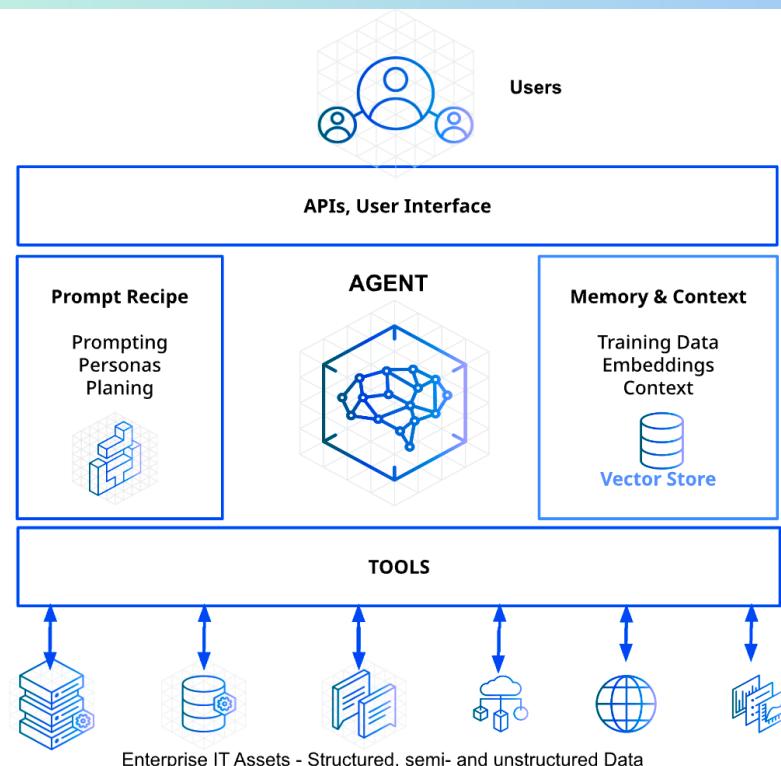
Extremely scalable, can handle complex tasks

Cons:

High risk if not monitored, hard to evaluate or trace, infra-heavy

Best used when:

The task is high-leverage, async, and doesn't require human feedback at every step.



Follow



OM NALINDE to learn more about AI Agents

Repost

Example

Problem Type	Suggested Agent Type	Enterprise Example
Structured, rules-based	Rule-Based System	Automatically approve reimbursements under \$100 or trigger invoice processing based on fixed thresholds
Human-led process, needs speed	Workflow Agent	Suggesting first-draft responses to support tickets in Zendesk, or summarizing meeting transcripts for Slack follow-up
Multi-step, repeatable, bounded	Semi-Autonomous Agent	Lead enrichment and follow-up: fetch CRM info → draft outreach → send email → log interaction
Complex, async, cross-system	Autonomous Agent	A competitive intelligence agent that monitors public news, pulls data from multiple APIs, clusters insights, and generates weekly market reports

How to Decide What to Build

Not by picking your favorite architecture.

You start with the problem.

[Ask yourself:](#)

- **Is it repetitive and structured?**
- **Does it involve language understanding or generation?**
- **Is it a multi-step task that needs decision-making?**
- **Do you trust an AI system to execute the entire task, or do you want a human in the loop?**

[Here's the key:](#)

- **These approaches aren't mutually exclusive.**
- **A single system can mix them — some parts might require high control, others can benefit from high autonomy.**
- **Each problem type can be tackled by either a single agent or a group of collaborating agents.**



**Interested in
more content like this?**

Follow :

OM NALINDE

