AGENTIC AI VS. GENERATIVE AI: WHAT ENTERPRISE ORCHESTRATION NEEDS TO HANDLE BOTH

Enterprises are adopting generative AI and agentic AI at the same time, often without a clear account of how the two differ or what each demands from the systems that actually run production work.

The distinction is operational, not just conceptual. Generative AI creates content: drafts, summaries, code. It changes how work gets designed. Agentic AI acts: it plans steps, takes actions across systems, and adjusts when conditions change. It changes how work gets executed. And the two also fail differently, need different kinds of oversight, and place different demands on the orchestration layer.

These differences shape what each form of AI requires from enterprise workflow orchestration. Agents moving into production are changing what that layer has to do.

What is generative AI?

Generative AI creates content. Given a prompt, it produces text, code, images, or summaries—new material shaped by patterns learned from the data it was trained on. It's the technology behind AI assistants that draft documents, answer questions, and write and explain code.

It's reactive, performing a standalone task with each prompt, and its output, such asa draft, an answer, or a report,typically goes to a person in the enterprise who reviews it and decides what happens next. In other workflows, generative outputs feed directly into automated pipelines without a human reviewing each one. In either case, the value is speed and accessibility: summarizing a

failure log in plain language, drafting a workflow structure from a description of what needs to run — work that once required specialist time now takes minutes.

What generative AI doesn't do is act. It changes how work gets designed and understood. It doesn't change how work gets executed.

What is agentic AI?

Agentic AI pursues goals. Given an objective, an agentic system makes a plan, carries it out, and adjusts it when conditions change. It draws on a language model for reasoning, with generative AI now functioning as a component rather than a standalone tool, and on external tools for action: it queries systems, triggers processes, calls APIs. Where generative AI responds only to prompts, agentic AI can also be set in motion by events: a threshold crossed, a job failed, a deadline approaching.

Instead of producing a draft, agentic AI changes the state of a system: a job is re-sequenced, a workload rerouted, a recovery initiated. A person still sets the goal and the boundaries, but the work between goal and outcome happens without step-by-step human direction.

In an operations context, an agentic system might detect that a critical data feed is running late, determine which downstream jobs depend on it, and re-sequence them to protect the delivery deadline—within the policies the enterprise has set.

How do agentic AI and generative AI compare?

In production, the differences that matter most come down to two questions: what happens when each system is wrong, and what each needs from the systems around it.

	Generative AI	Agentic AI
Core function	Creates content (text, code, images, summaries) in response to a prompt	Pursues a goal by planning the steps, taking actions across systems, and adjusting as conditions change
How it's engaged	Reactive. Each request stands alone: prompt in, output out	Goal-driven. Given an objective, it initiates and carries out a multi-step sequence, often triggered by events rather than people
What it produces	An artifact: a draft, an answer, a piece of code	An outcome: a completed task, a changed system state
How it adapts	Varies the style, format, and substance of what it creates	Revises its plan mid-course when a step fails or conditions shift
Human role	A person prompts it, reviews the output, and decides what to do with it	A person sets the goal and the boundaries; oversight shifts from checking each output to defining the policies actions must follow

Primary risk	Wrong content — an inaccurate answer, a flawed draft. The damage is contained until a person acts on it	Wrong actions — a step taken on a live system. The damage is direct, which is why autonomy requires runtime controls, not just review
What it needs from orchestration	Reliable workflows around it: pipelines that feed models the right data and move outputs where they need to go	Governed execution: enforced policies, full auditability, and visibility into what agents actually do in production
In an enterprise workflow	Summarizes a failure log. Drafts a job definition from a plain-language description. Generates a compliance report from structured data. Translates a legacy script into a modern format	Detects a delayed data feed and re-sequences dependent jobs within enterprise policies. Monitors an SLA threshold and reroutes a workload before a deadline is missed. Identifies a recurring failure pattern and opens a remediation workflow without waiting for a ticket

When AI is wrong: the risk asymmetry

When generative AI fails, it produces bad content, such as an inaccurate summary, a flawed piece of code, or a report that misstates the numbers. Whether a person reviews that output or it flows into an automated pipeline, the failure is informational: wrong content, not a wrong action. The systems downstream may act on bad information, but the AI itself hasn't changed a system state.

When agentic AI fails, it takes a wrong action. A job triggered against the wrong environment, a workload rerouted on a faulty premise, a remediation step that compounds the original problem. The AI has changed a system state directly, and there may be no pause between the mistake and its consequences.

This asymmetry is why oversight built for generative AI doesn't transfer to agentic AI. Review of outputs, whether human or automatic, can catch bad content before it spreads. But when the AI produces actions rather than content, governance has to move into the runtime itself: enforcing policies at the moment of execution, holding boundaries the agent cannot cross regardless of what it decides, and keeping a complete record of what it actually did.

What this means for enterprise AI workflow orchestration

Both kinds of AI place demands on the workflow orchestration layer: the systems enterprises already use to run workflows, enforce dependencies, and meet SLAs across applications, data pipelines, and infrastructure. But the demands are different.

Generative AI needs reliable workflows around it. Models are only as good as the pipelines that feed them current, correct data, and the processes that move their outputs to where decisions get made. That's what orchestration has always done—it makes sure the right work runs in the right order, every time.

Agentic AI needs something more: governed execution. If agents are going to act on production systems —triggering jobs, rerouting workloads, initiating recoveries— then something has to enforce the policies those actions must follow, maintain the audit trail of every action agents take, and make those actions visible enough for the enterprise to intervene when an agent's plan diverges from the enterprise's interests.

Some in the industry describe agentic AI as becoming the orchestration engine itself—the intelligence that coordinates jobs, adjusts schedules, and optimizes resources on its own. Agents will certainly participate more and more in how work gets sequenced and run. But collapsing the agent and the orchestration layer into a single thing gives up something enterprises should not give up: an independent layer that governs what agents do, sitting outside the agents themselves. The rise of agentic AI means that layer has to expand to do more than it ever has before—not that the layer can be replaced by agentic AI itself.

Generative AI creates the intent. Agents carry it out. The orchestration layer governs execution, applying the same policies, auditability, and controls to agents as to everything else it runs.

How Control-M approaches this

Control-M, BMC's orchestration platform, provides a unified orchestration layer for both generative AI and agentic AI by governing the execution of AI workflows, agents, applications, and data pipelines through a single operational framework.

On the generative side, Control-M orchestrates AI workflows that contain generative AI tasks, in addition to the data pipelines, application workflows, and dependencies that generative AI runs on. It also uses generative AI internally to help users: Control-M's built-in AI advisor, Jett, answers questions about workflows and their status in plain language; AI Workflow Creator builds workflow structures from natural-language descriptions.

On the agentic side, Control-M orchestrates AI agents and AI-powered tasks alongside the data pipelines, applications, and event-driven workflows it already runs—with the same reliability, visibility, and governance. Integrations with agent frameworks including CrewAI, LangGraph, and Snowflake Cortex bring agent-driven work into governed workflows. And through the optional Control-M MCP Server, built on the Model Context Protocol (an open standard for connecting AI agents to external systems), external AI agents can interact with Control-M itself— triggering jobs, checking workflow status, investigating failures—without bypassing enterprise controls.

FAQ

What's the main difference between agentic AI and generative AI?

Generative AI creates content (text, code, summaries) in response to a prompt. Agentic AI pursues a goal: it plans the steps, takes actions across systems, and adjusts when conditions change. The difference shows in the output: Generative AI produces an artifact that a person or an automated process reviews. Agentic AI produces an outcome—a changed system state.

Can generative AI and agentic AI work together in the same workflow?

Yes, and in practice they're deeply intertwined. Agentic systems are built on language models—they use them for reasoning, planning, and intermediate tasks like summarizing data or interpreting results. The orchestration layer coordinates both, making sure the generative steps get the right

inputs and the agentic steps run under the controls the enterprise sets.

Why can't enterprises govern agentic AI the same way they govern generative AI?

Because they fail differently. When generative AI is wrong, it produces bad content: an inaccurate draft, a flawed summary. The damage is contained until that output is acted on, and a review—by a person or an automated process—can intercept it. When agentic AI is wrong, it takes a wrong action on a live system, and the consequences can play out before anyone knows a mistake has been made. That's why oversight has to move from reviewing outputs to setting limits the agent can't exceed and enforcing them as it acts.

What role does workflow orchestration play in running AI in production?

Both kinds of AI depend on the orchestration layer, but differently. Generative AI needs reliable workflows around it. Agentic AI needs governed execution—policies enforced on its actions, a complete record of what it did, and visibility while it runs. The orchestration layer is the natural place to govern agents' actions, because it already enforces dependencies, SLAs, and recovery for the workloads around them.

Where to go from here

As agents enter production alongside data pipelines, applications, and human operators, the orchestration layer can't treat them as an exception. They need the same oversight as everything else that runs—governance that moves at the speed of the agents themselves, with the auditability and controls the enterprise already depends on.

Learn how Control-M orchestrates AI workflows and AI agents in production.

Explore BMC's approach to enterprise agentic orchestration.