

WHEN IT COMES TO AGENTIC FABRIC, WHAT'S INTERESTING, IMPORTANT, AND IMMEDIATE?



The industry has spent the last two years proving agents can “think.” The next few years will be about proving they can operate reliably.

Depending on who you ask, agentic fabric (sometimes called agent fabric, agentic control plane, or agent control fabric) is either a meaningful architectural idea or the latest label attached to AI agents.

The most useful conversations are happening among platform teams, IT operations leaders, architects, and automation professionals asking: *What happens after agents move from experiments to production? What does it take to operate hundreds or thousands of them reliably and securely?*

That's where the discussion quickly moves from what agents can do to how they'll be operated.

What Do People Mean by “Agentic Fabric”?

Agentic fabric is best understood as an architectural concept, not a product, platform, or framework. In practice, the term describes the layer that connects, coordinates, governs, and observes AI agents as they interact with systems, data, tools, and each other.

Strip away the terminology and it's an effort to create a control plane for agents. A way to:

- Connect agents to systems and data
- [Govern what they can and can't do](#)
- Coordinate work across multiple agents
- Monitor actions and outcomes

- Maintain accountability when agent-driven decisions affect business operations

Agentic fabric is giving a new name to a challenge IT teams have been dealing with for decades: coordinating increasingly complex systems in a way that's reliable, secure, observable, and scalable.

Whether the label lasts is beside the point. The underlying operational challenges are real and arriving regardless of what we call them.

Agentic Fabric vs. Orchestration

At this point, some of you may be thinking: *Isn't this just orchestration?*

It's a fair question and, in some cases, a fair criticism. Many products being marketed as agentic fabric are fundamentally [orchestration platforms](#), workflow engines, integration layers, or governance tools wrapped in AI terminology.

At the same time, it's hard to talk about agentic fabric without talking about [orchestration](#). The difference is that agent discussions tend to surface a broader set of operational questions around governance, identity, accountability, lifecycle management, and shared context, particularly as agent populations grow.

What's Interesting About Agentic Fabric?

Agentic fabric is getting attention because agents are increasingly interacting with tools, data, business systems, and each other, expanding the range of what's possible.

- **Specialized agents are becoming practical**

We can now realistically envision operations agents, customer service agents, finance agents, software delivery agents, and data analysis agents working alongside human teams, each focused on specific domains and responsibilities.

Agents are starting to mirror the environments they're entering, acting as specialized systems performing specialized functions as part of larger processes.

- **Multi-agent workflows are moving beyond theory**

Teams are implementing [workflows where agents coordinate work](#), access different tools, and participate in larger business processes. What's interesting is that we're beginning to structure business processes around that collaboration.

- **Agents are becoming platform consumers**

Platform teams are used to building services for developers, operators, SREs, and business users. Now they're designing for another consumer: agents.

As agents become participants in operational systems, familiar questions surface:

- How do they authenticate?
- What permissions should they have?
- How are policies enforced?
- How is their behavior observed?

The idea of agents as platform consumers may be one of the more underappreciated discussions happening now.

These three developments are interesting because they expand the range of work agents can participate in and the number of systems they can interact with. But they're still largely conversations about capability and possibility.

The next challenge is operational.

What's Important About Agentic Fabric?

What makes agentic fabric important is how the idea shows up in real operational requirements and solution capabilities that move agents from isolated interactions into business operations.

Powerful models and capable agents are increasingly accessible. As intelligence becomes commoditized, the differentiator is execution: the ability to deploy, coordinate, and govern agents across systems, workflows, and teams in a reliable, secure, and scalable way.

If control, not intelligence, is becoming the differentiator, agentic fabric solutions should:

1. Help agents contribute to outcomes, not just tasks

A customer issue might require data collection, analysis, approval, escalation, and action across multiple systems. A software delivery process might involve testing, validation, deployment, and notification workflows.

In practice, that means breaking larger objectives into smaller tasks, routing them to specialized agents or systems, and adapting when something doesn't go according to plan.

An agent's ability to perform a task is only the starting point. What's important is enabling multiple agents to contribute to a broader process without manual coordination between steps.

2. Give agents access without giving up governance

Agents need access to applications, APIs, databases, knowledge sources, and business systems to be useful. But we don't want them operating with uncontrolled access. Standards like MCP make connections easier to establish, but they don't replace [policy enforcement, monitoring, and accountability](#).

What's important is that those connections can happen inside existing security and governance frameworks rather than forcing teams to invent an entirely new operating model for AI.

3. Bring agents into the operating model

If an agent can take actions on behalf of the business, questions like ownership, authorization, and accountability become unavoidable:

- Does the agent have the right amount of resources available?
- Are we able to recover from transient failures?
- Can the agent handle an increase in request throughput?
- Can we pause the agent if we need to verify inputs?
- Can we rollback to a previous version of this agent?

Those questions need to be part of the operating model, not something we sort out after deployment.

4. Make agent behavior visible enough to trust

If an agent makes a recommendation, takes an action, accesses a system, or triggers a workflow, we need to understand what happened and why. That means being able to see prompts, decisions, tool calls, workflow execution, and outcomes. The more autonomous agents become, the more valuable that visibility becomes.

Taken together, these capabilities help move agents from experimentation to production.

What's Immediate About Agentic Fabric?

The immediate issue is that agents are starting to enter production environments faster than many teams can inventory and govern them as a population. That creates two practical challenges now: discovering where agents are showing up, and maintaining operational control as their number, scope, and interdependence increase. A call for preparation, not panic.

Agent sprawl is becoming harder to ignore

Your organization may already have more agents than you realize. Agents are arriving through Microsoft Copilot environments, SaaS platforms, internal development teams, departmental initiatives, third-party solutions, and embedded AI capabilities.

Before we can manage agents, we need to be able to find them. While most teams can inventory applications, infrastructure, and workflows, fewer can say how many agents are operating in the environment today.

Complexity is the real challenge

As agent populations grow, teams will need to manage conflicting actions, dependencies, permission drift, ownership ambiguity, tool contention, workflow failures, escalation paths, and audit requirements.

None of these are unique to AI. Enterprises have spent decades managing dependencies, permissions, failures, ownership, and coordination across applications and workflows. What's different is that those dependencies now involve software participants that can make recommendations, invoke tools, initiate actions, and interact with one another without direct human involvement.

Consider the progression: a single agent can be reviewed manually. Ten agents begin to affect workflows. Hundreds of agents interacting with systems, data, workflows, and other agents require a stronger operating model.

What's the Control Plane for Agents?

The industry has spent years building control planes capable of coordinating applications, infrastructure, workflows, and [data pipelines](#) at scale. It's hard to see most of us standing up a separate operating model just to manage agents.

A more likely outcome is that agents become another participant in the environments we're already managing, alongside applications, data, infrastructure, and business processes. The practical question is whether existing operational control planes can evolve to incorporate agent-driven work, rather than whether we should introduce yet another operational layer.

That's where solutions like Control-M come into play. Not as standalone agent platforms, but as a way to bring agent-driven work into the same operational framework teams use to coordinate, govern, and monitor business operations across the broader environment.

[See how Control-M operationalizes AI](#)

Evaluating Agentic Fabric: Questions That Matter

As the market fills with products claiming to be agentic fabrics, agent control planes, and multi-agent platforms, leaders should continue to focus on operational capability.

Discovery

- Can you identify all agents operating in the environment?
- Does every agent have a business owner?
- Can unmanaged or "[shadow](#)" agents be detected?
- Can you see where agents are interacting with critical systems and data?

Governance

- Are permissions consistently enforced across agents?
- Can agents operate within existing security and identity frameworks?
- Are policies centrally managed and auditable?
- Can access be reviewed and revoked when agents change roles or are retired?

Visibility

- Can actions, decisions, tool calls, and outcomes be observed?
- Can activity be traced across multiple agents, systems, and workflows?
- Is there [a complete audit trail](#) when investigators need to reconstruct events?
- Can teams understand why an agent took a particular action?

Coordination

- How does agent-driven work interact with existing workflows?
- Are dependencies managed across agents, applications, and data pipelines?
- What happens when an agent fails, stalls, or produces an unexpected outcome?
- Can human intervention be incorporated when needed?

Data and Context

- How do agents access trusted business data?
- Is access governed and monitored?
- Can context be shared consistently across multiple agents?
- Are business rules and policies applied consistently regardless of which agent is acting?

Scale and Control

- Would the current approach still work if the number of agents increased tenfold?
- What happens when agents begin interacting with other agents at scale?
- Can operational teams maintain visibility and governance as adoption grows?
- What becomes the control plane as agent populations expand?

Agent Lifecycle

- How are agents approved before deployment?
- How are changes reviewed and tested?
- When should an agent be retired?
- How do you verify inactive agents no longer retain access?
- Who is responsible for periodic review of permissions, behavior, and outcomes?

The strongest agentic fabric architectures won't simply help agents act. They'll help teams operate large populations of agents with the same discipline already applied to applications, workflows, infrastructure, and data.

Whether we call it agentic fabric, agent orchestration, or a [control plane for agents](#), the job remains the same: maintaining operational control as large populations of agents become part of how work gets done.