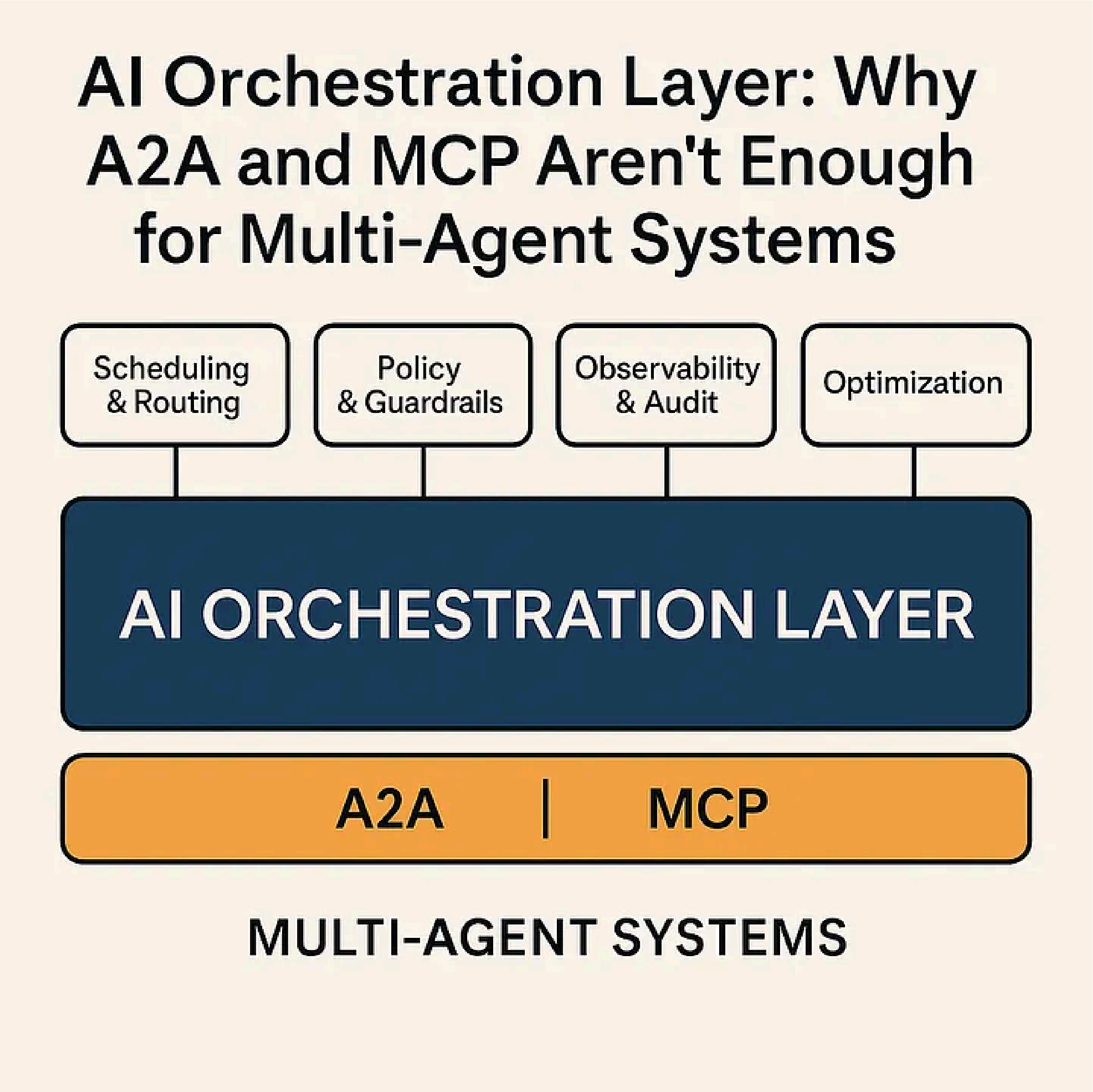
Stackademic · Follow publication

AI Orchestration Layer: Why A2A and MCP Aren't Enough for Multi-Agent Systems

6 min read · Sep 28, 2025

RAKTIM SINGH Follow

Discover why A2A and MCP protocols alone can't scale multi-agent AI systems, and how orchestration layers bring safety, governance, and efficiency.



Introduction: From Copilots to Fleets of Agents

If 2023–2024 were all about "adding an AI copilot(s)," 2025 is about managing fleets of them.

Companies are moving past single assistants to try out multi-agent systems: specialized AI agents who plan, retrieve, decide, act, and verify all in parallel.

Two major standards are helping to make this change:

- **A2A (Agent-to-Agent):** a protocol for agent-to-agent communication, which allows agents from different providers to establish secure handshakes.

- **MCP (Model Context Protocol):** an “AI USB-C” layer for standards based on how models discover tools, access data, and invoke outside systems.

These protocols provide connectivity. However, they do not provide coordination.

That’s what the AI orchestration layer does.

Protocols vs. Orchestration

Connectivity vs. Coordination

You can think of A2A/MCP as TCP/IP + USB-C for AI: they allow agents and tools the ability to talk to and plug into one another.

The orchestration layer is the **operating system & control tower** on top.

- **Scheduling & Routing:** Which agent should operate next? In what sequence? With what context?
- **Policy, Guardrails & RBAC:** Who is allowed to send email, move funds, or update a field in the CRM.
- **Observability & Audit:** Full traces across dozens of agents, that have replay, cost/latency budgets, or drift detection.
- **Optimisation:** Select smaller/cheaper SLMs when feasible, use larger models for edge cases, and safely parallelize.
- **Failure Handling:** Retries, fallbacks, hedged requests, degradation.

Without this “OS,” multi-agent pilots fail: Agents loop, conflict, or overspend, even if everything is “connected” according to the protocols.

Why A2A and MCP Were Necessary (Beyond REST APIs)

REST APIs Are Static, Agents Are Dynamic

REST assumes you know the endpoint and payload.
Agents do not. They need to discover what tools exist, learn each tool’s schemas, and integrate them dynamically.

MCP enables tool discovery, schema exchange, and secure operation instead of hard-coding API calls.

The API Sprawl Problem

Enterprises are drowning in thousands of APIs.
It would not scale to teach agents each custom spec and process.

MCP provides a universal *plug-and-play connector* like USB-C: regardless of the backend, the agent connects to the MCP server and immediately knows what it can do.

REST is Client/Server; Agents Need Peer-to-Peer

REST is essentially a client/server model.
Agents, however, need to operate in **peer-to-peer mode**: hand-offs, groupings, shared reasoning.

A2A provides an agent grammar to collaborate across frameworks and vendors.

Access Control and Governance

REST does not uniformly apply **access control, RBAC** (Role-Based Access Control), **or audit logging**.
Agents need strict policies, for example: *"This bot can read CRM records, but cannot delete them."*

MCP/A2A embed access and governance into their architecture.

Streaming vs. Multi-Modal Interaction

REST is request-response-based.
Agents require **streaming context, negotiation, and multi-turn interactivity.**

A2A enables this multi-turn collaboration, while **MCP** ensures tools provide context-specific data.

TL;DR

- REST = static, deterministic, request-response client/server
- MCP = dynamic, discoverable, secure tool/data access

- A2A = peer-to-peer agent coordination
- **Orchestration = governance + optimization layer on top**

Why Orchestration Is a New Challenge Now

From Copilots to Agent Teams

Vendors like Google, Salesforce, ServiceNow, and UiPath now build orchestration-first platforms, not just copilots.

Regulatory & Enterprise Risk

Once an agent can act — send emails, update records, or move money — **authorization, audit, and safety** become mandatory.

Security Wake-Up Call

A malicious MCP server package was discovered on npm, silently exfiltrating emails — proving protocols alone need orchestration guardrails.

The Agentic Internet Race

Standards like A2A and MCP are spreading fast.
But orchestration will define **policy, budgets, and safety** at runtime.

Core Capabilities of an Orchestration Layer

Planner–Router–Executor Cycle

- Planner decomposes the goals.
- Router selects the right agent/tool.
- Executor manages budgets, results, and controls.

Governance of Memory

Manage short-term and long-term memory with **PII rules, TTLs, and redaction.**

Policy and Permissions

- RBAC and human-in-the-loop gates.

- Signed tool calls with immutable audit logs.

Observability & SLOs

Dashboards that track cost, latency, loops, and compliance violations.

Cost/Latency Optimization

Default to smaller SLMs, escalate to LLMs only when necessary, and apply caching and GPU-aware routing.

Real Use Cases of Orchestration

1. Customer Service Automation

- **Agents:** triage, knowledge lookup, action agent (returns), QA.
- **Orchestration:** Enforces return policy, escalates if uncertain.
- **In practice:** Salesforce Agentforce, ServiceNow Orchestrator.

2. IT Operations & Employee Support

- **Agents:** access broker, catalog lookup, change executor, verifier.
- **Orchestration:** RBAC + SLA timers, rollback on failure.

3. Sales & Marketing Campaigns

- **Agents:** research, writer, compliance, CRM operator.
- **Orchestration:** Parallelizes tasks, adds compliance gates.
- **MVP:** Zapier Agent Pods orchestrating across 8k apps.

4. Financial Operations

- **Agents:** OCR extractors, risk checkers, compliance, payout.
- **Orchestration:** Deterministic routing with human approvals.

5. Multi-System Concierge (Banking, Telco, Gov)

- **Agents:** concierge, auth, domain specialists, payment handler.

- **Orchestration:** Delegation, retries, context preservation.

Industry Participants in Orchestration

Cloud Platforms

- Google Vertex AI (Agent Builder + A2A support)
- OpenAI (MCP + Swarm)
- Anthropic (Claude orchestration)
- NVIDIA (NIM microservices, Agent Blueprints)

Enterprise Automation

- Salesforce Agentforce (Command Center)
- ServiceNow (Control Tower)
- UiPath (multi-agent orchestration)
- Cognigy (customer service orchestration)

Open-Source Frameworks

- LangGraph (LangChain)
- Microsoft AutoGen
- LlamaIndex AgentWorkflow
- crewAI
- OpenAI Swarm

Security Lessons from MCP Incidents

A malicious MCP server highlighted **supply-chain risk.**

Key security features needed:

- Verified MCP servers (hashes, provenance)
- Sandbox agent actions

- Human approvals for irreversible steps
- Real-time monitoring for exfiltration and drift

Buyer's Checklist for Orchestration Platforms

- ✅ Native support for A2A + MCP
- ✅ RBAC, policy gates, and audit trails
- ✅ End-to-end observability dashboards
- ✅ Budget-aware routing (SLM-first, escalate to LLMs)
- ✅ Compliance-ready memory management
- ✅ Safety nets: retries, fallbacks, malicious detection
- ✅ Human-in-the-loop approvals

Final Note: The Road Ahead

A2A and MCP are the tracks of the agentic internet. But tracks don’t move trains.

The **AI orchestration layer** is where **strategy, safety, and scale** live: planning, routing, governing, observing, and optimizing entire teams of agents across your enterprise.

If you’re charting a course today:

- Standardize on A2A + MCP
- Select an orchestration layer (build or buy)

- Make governance the default
- Start with 2–3 target use cases

Done right, orchestration won't just make your agents work — it will make them **trustworthy, efficient, and enterprise-ready**.

A message from our Founder

Hey, Sunil here. I wanted to take a moment to thank you for reading until the end and for being a part of this community.

Did you know that our team run these publications as a volunteer effort to over 3.5m monthly readers? **We don't receive any funding, we do this to support the community.** ❤️

If you want to show some love, please take a moment to **follow me on LinkedIn, TikTok**, **Instagram**. You can also subscribe to our **weekly newsletter**.

And before you go, don't forget to **clap** and **follow** the writer!

Artificial Intelligence   Multi Agent Systems   Agentic Ai   Model Context Protocol

Enterprise Ai

Follow

Published in Stackademic

60K followers · Last published 14 hours ago

Stackademic is a learning hub for programmers, devs, coders, and engineers. Our goal is to democratize free coding education for the world.

Follow

Written by RAKTIM SINGH

30 followers · 42 following

I had done B.TECH from IIT BHU. Read more about Tech & Fintech at my website www.raktimsingh.com and my Youtube channel https://www.youtube.com/@raktim_hindi

Responses (1)

Bgerby

What are your thoughts?

Aurelije Zovko
2 days ago

Excellent article. My company is already building an AI intelligent orchestration framework.

Reply

More from RAKTIM SINGH and Stackademic

In Data And Beyond by RAKTIM SINGH

Business Forecasting 2025: Why Metrics, Models, and Methods Must Change

Oct 2 59 1

In Stackademic by Subodh Shetty

Practical Terminal Commands Every Developer Should Know

Most developers know the basics cd, ls, pwd, maybe even grep. But the terminal has a lot more under the hood. There are commands and…

Sep 19 704 12

In Stackademic by Shanvika Devi

I Removed a Single Annotation in Java and My API Became 50x Faster

Turns out, one tiny Spring Boot annotation was secretly slowing everything down — here's the full story of how I caught it and made our...

Sep 22 438 15

In Stackademic by RAKTIM SINGH

Federated Learning Explained: Privacy-Preserving AI Training for the Future

Federated learning (FL) is a distributed approach to training machine learning models on multiple devices or organizations without…

Sep 2 👏 22

See all from RAKTIM SINGH

See all from Stackademic

Recommended from Medium

In Bootcamp by Imran Shaik

The Hidden Friction Points in Modern Data Platforms—and How to Remove Them

When people talk about modern data platforms, the conversation usually revolves around scale, speed, and AI. How many rows can we handle...

Sep 13 1

Aman Raghuvanshi

Agentic AI #6—Multi-Agent Architectures Explained: How AI Agents Collaborate

We are at a critical point in the growth of artificial intelligence. As individual AI agents grow in power from personal assistants like…

Jul 3 120

Aruna Pattam

Agentic AI: Part 6—End-to-End Observability in Agentic AI

The Operational Scaffolding of Trust

Oct 6 1

In Agentic AI & GenAI Revolution by Yi Zhou

The Agentic Framework Battlefield: How to Escape Vendor Lock-In and Survive the Next AI War

Escape vendor lock-in with stack-driven AI: 5 rules, volatility insights, and system design for production-ready enterprise agents.

Sep 23 👏 31

In AgenticAI— The Autonomous Intelligence by Monoj Kanti Saha

Within the Context-Engineered Realm of Agentic AI, Can MCP Reinvent Enterprise Integration?

Exploring how the Model Context Protocol (MCP) could shift enterprise integration from API plumbing to context orchestration—building a…

5d ago 👏 2

In Towards AI by ravindu somawansa

The Real Reason Enterprise Agents Are So Hard (And It’s Not the AI)

Enterprise demos make agents look effortless. But the real pain starts after the pitch—when AI agents, workflows, legacy systems, and...

Oct 6 👏 34 💬 2

See more recommendations