

The Orchestrator Is a Delivery Manager

By Rahul Jindal

Citibank needs a Java developer in New York. The good delivery manager does not open the skills database, filter to "Java," and grab the first name. She knows who was rated highly by Goldman on the last engagement, who has actually shipped on a trading floor and not just in training, who lived in New York and would go back, and who rolls off their current project in three weeks. The match is not a keyword lookup. It is a judgment built on knowing the bench cold: not the list of who can do the work, but the deeper read of how well each one has done it, in what context, and whether they are available and still sharp.

I spent the first stretch of my career inside that model. Knowledge services, legal and IP process, the first IP hire building a delivery bench of analysts and lawyers staffed against client matters one after another. The lesson that stuck was simple and it never left me. The delivery manager who knew the bench cold was worth more than the one who had the biggest bench. Capacity was cheap. The match was everything.

I have spent 2026 building and watching enterprise agent platforms, and the same shape keeps surfacing. The orchestrator in Gemini Enterprise, in Bedrock, in whatever supervisor graph your team wired together last month, is doing the delivery manager's job. It reads resumes. It staffs work. It lives or dies on how well it knows its bench. So this piece does three things: it lays out how close the analogy actually is, it pressure tests where the analogy breaks, and then it argues that the two places it breaks are the whole point. The breaks are where the new discipline lives.

The Bench, the Resume, and the Staffing Call

Start with how clean the mapping is, because it is closer than a metaphor. It is almost a design spec.

The bench is the agent registry. Google renamed AgentSpace to Gemini Enterprise in October 2025 and put an Agent Gallery at its center: made-by-Google agents, your organization's agents, your own, and a marketplace of partner agents you request access

to. AWS shipped Bedrock AgentCore to general availability the same week, with an Agent Registry where teams publish agents, skills, and tools behind approval workflows and semantic search. The bench is now a product surface, and every orchestrator shops it.

The resume is the Agent Card. Google's Agent2Agent protocol, donated to the Linux Foundation in June 2025 with AWS, Microsoft, Salesforce, SAP, ServiceNow, and Cisco behind it, defines a JSON document an agent publishes at a known address: its identity, its declared skills, its endpoints, the auth it requires, and a signature you can verify before you trust it. An orchestrator fetches the card and reads it before invoking anything. That is a recruiter pulling the resume before the interview, down to checking the signature is real.

The toolbox is Model Context Protocol. Where the Agent Card is the resume, MCP is how the agent reaches the systems it needs to do the work: the data, the tools, the APIs. Google and Anthropic both frame the two as complementary, and the staffing version is exact. The Agent Card says what the consultant can do. MCP is the badge that gets them into the building and onto the client's systems. A brilliant agent with no MCP access to the customer's data is the brilliant consultant who cannot get past the lobby.

The staffing call is the orchestrator itself. The dominant pattern in production, whether you build it on LangGraph or CrewAI or a Vertex supervisor, is a planner that decomposes the work into roles and delegates each to a specialist, reading capability metadata and routing on semantic relevance. CrewAI says it out loud in its own docs: treat your agents like a human team. That is the delivery manager breaking a statement of work into staffed roles and assigning each to the right resume.

Even the soft parts map. "Lived in New York" is context fit, and for an agent context fit is the data it can reach, the region and latency it runs in, the language it speaks, the compliance certifications it carries. The reference check is the eval. Vertex's agent evaluation grades not just final-response quality and hallucination rate but trajectory: whether the agent did the work the right way, with the right tool calls in the right order, not merely whether it landed the answer once. That is a reference that grades process, not just outcome, which is exactly the reference a good delivery manager actually wants. And the performance review is observability, the drift monitoring that watches the live deployment after the staffing decision is made.

IT services

Agent platform

The bench

The agent registry or gallery (Gemini Enterprise Agent Gallery, AWS Agent Registry on Bedrock AgentCore)

The resume

The Agent Card (A2A): identity, declared skills, endpoints, auth, and a signature, published at a known address

A badge for the building

MCP tool access plus the data and compliance scope the agent is actually granted

The staffing call

The orchestrator: a supervisor or planner that decomposes the work and delegates to specialists

“Lived in New York”

Context fit: the right data access, region and latency, language, and certifications

The reference check

Evals: response quality, tool-use quality, hallucination rate, and trajectory (did it work the right way)

The performance review

Observability and drift monitoring over the live deployment

Every instinct a good delivery manager has, the orchestration stack is busy rebuilding in software. The mapping is not loose. So the interesting question is not whether the analogy holds. It is where it stops holding, because that is where the orchestrator's job becomes something a delivery manager never had to do.

Three Places the Analogy Breaks

The analogy breaks in three places. Two of them are economic and structural, and they invert the job. The third is a gap in the current tooling, and it is a business waiting to be built.

First, the bench economics invert. IT services lives and dies on utilization. A consultant idle on the bench burns full loaded salary and produces nothing, so the delivery manager's first instinct, drilled in by the margin model, is to keep scarce and expensive

people billable. Bench is the enemy. Now look at the agent bench. Agents scale to zero. An idle agent costs storage pennies, and you pay only when it runs: fractions of a cent per thousand tool calls on AgentCore, a few cents an invocation at most. Supply is abundant and cloneable, because you can spin up a thousand copies of your best agent for the cost of the calls. The scarce, expensive bench that the entire services discipline is built to manage simply is not there.

That single fact flips the optimization. The delivery manager rations a fixed pool of scarce talent and is judged on utilization. The orchestrator is not rationing anything. It is selecting from abundance, and its problem is not keeping everyone busy. Its problem is picking the right one and trusting that the pick still holds. The scarce resource is no longer the worker. It is verified, current trust in the worker.

“The delivery manager rations scarce talent and is judged on utilization. The orchestrator drowns in abundant talent. The scarce thing is no longer the worker. It is verified, current trust in the worker.”

Second, the worker's identity is unstable. A human consultant is the same person on Monday and Friday. Reputation accrues to a stable identity over years, and the Goldman rating still describes the person who shows up next quarter. That stability is the load-bearing assumption under every reference check ever run. An agent breaks it. The model underneath gets retuned, a prompt dependency shifts, an upstream API changes the shape of its output, and the Agent Card still lists the same skills it listed yesterday. The resume did not change. The worker did. It is the consultant who got a brain transplant over the weekend and kept their resume.

This is not theoretical, and the cleanest case study comes from a model provider grading its own homework. In September 2025 Anthropic published a postmortem on three infrastructure bugs that had quietly degraded Claude's output through August and early September. At the worst hour on August 31, around 16 percent of one model's requests were affected. The instructive part is why it was hard to catch. Nothing was down. The API kept returning success codes at normal latency, and the internal evals were too noisy to isolate the regression cleanly. The work just silently got worse while every dashboard

stayed green. That is the delivery manager whose best consultant started slipping and nobody filed a review, because they still showed up on time and still answered email.

The consequence is the most important operational difference in this whole comparison: reference checks expire. You cannot vet an agent once at onboarding the way you check references once at hire, because the thing you verified is not stable. Point-in-time evaluation is structurally insufficient. The verification has to be continuous, running against a fixed test set every time the substrate moves, because the substrate moves on someone else's schedule and does not send a memo. Most agent stacks have not internalized this. They onboard an agent the way you hire a person, and then they stop looking.

Third, the references are not portable yet. In staffing, a consultant's track record travels with them. The Goldman rating, the project history, the certifications are attached to the person and legible to the next delivery manager who picks up the file. The market solved reference portability decades ago. The Agent Card has not. It lists skills and capabilities, but it carries no verified track record. Evals today live off to the side, in development and observability tooling, not stapled to the resume the orchestrator reads at selection time. So the agent arrives with a page of claimed skills and no references attached, and the orchestrator either takes the claims on faith or runs its own audition every single time.

This is the white space, and it is starting to fill. A2A added signed cards in its 0.3 revision. Verifiable credentials for agents are moving from concept to standards track. Singapore's cyber regulator, in an October 2025 addendum on securing agentic AI, already requires organizations to keep a trusted agent registry and authenticate agents with verifiable credentials and short-lived tokens. The credentialing layer that a staffing firm quietly is, the vetted, portable, trustworthy track record, has not been built for agents. Whoever builds the verifiable reference that travels on the card is building the reference-check infrastructure for machine labor, and that is a larger business than any single agent on the bench.

The Other Side of the Desk

The bench is not passive. In services the talent has duties too, and the consultant who gets the best matches is the one who is discoverable, skilled, current, and well-connected to the people who staff the work. The same four duties now fall on the agent developer. If you build agents, you are not building a tool. You are putting a worker on a bench and asking an orchestrator to staff it, and the orchestrator staffs the worker that does these four things.

Be discoverable. The consultant with no resume in the system does not get picked, however good they are. For an agent that means a real Agent Card, signed, with honest and specific skill declarations, listed in the registries the orchestrators actually read. The proactive reach-out that a good consultant does, the staying-in-front-of-the-staffer, maps to registering in the marketplaces and advertising capabilities rather than waiting to be found. An agent that is not in the gallery does not exist.

Be skilled, and be able to prove it on the real work. A track record is the asset, and it has to be built on the work that matters, not on demos. An agent that aces a public benchmark and falls over on the customer's actual data is the consultant who interviews beautifully and cannot deliver. Trajectory evals on real tasks, run continuously, are how the proof gets made and kept.

Stay current. The consultant who stopped learning three years ago is quietly obsolete. For an agent this duty is sharper and stranger, because the ground moves under you whether or not you act. A model bump can improve your agent or silently break it, and you will not be asked first. So staying current is not background reading. It is a discipline of regression testing every time the substrate changes and re-certifying before you let the orchestrator keep routing to you. The real job of an agent developer is not building the agent once. It is owning the agent's reliability over time. That is the second break from the supply side, and it is the part most teams skip.

Build the relationship with the orchestrator. This is the subtlest duty and the one worth slowing down on. In services, a relationship with the delivery manager is trust that lowers their verification cost. They have staffed you ten times, so the eleventh time they do not re-check everything, and that earned trust becomes reduced friction, and reduced friction becomes preferential routing. The same dynamic is coming for agents. A track

record of reliability earns lower-scrutiny selection, faster routing, default-choice status when the orchestrator has a job that fits.

Here is the trap, and it is the second break wearing a friendlier face. The entire value of the relationship is that it lets the orchestrator skip verification. And skipping verification is exactly what a silently drifting agent needs to slip through. In human services the risk is bounded, because a person changes slowly and you would notice. With agents the thing you stopped checking can change overnight. So the relationship an agent earns cannot be standing trust. It has to be continuously re-earned trust, where the reduced friction is always backed by live monitoring and never by reputation alone. The orchestrator that routes on relationship without verification is the delivery manager who keeps staffing a favorite long after the favorite stopped doing the work. The good orchestrator's loyalty is to the evidence, refreshed.

“A relationship with the orchestrator is trust that lowers its verification cost. The danger is that lowered verification is exactly what a silently drifting agent needs to slip through. Trust has to be re-earned, not banked.”

Who Watches the Staffer

There is one more thing a delivery manager is, and it is the uncomfortable one. A delivery manager has a conflict of interest. They bench-warm favorites. They push the high-margin resource. They staff the person who makes their number over the person who serves the client best. The orchestrator inherits all of it and adds new versions. It can prefer first-party agents over a better third-party one. It can route to the cheaper agent because the per-call cost rolls up to its own budget. Principal-agent problems do not vanish when the agent is software. They get faster and harder to see.

Then there is Goodhart. The moment an eval score becomes the routing criterion, it stops measuring capability and starts measuring score-optimization. This is documented and severe: on some benchmarks reward hacking shows up in nearly every attempt, models have been caught overloading equality operators so their output spuriously matches the

expected answer, and one was caught swapping its chess opponent for a weaker engine to win. Contamination inflates published scores by meaningful margins. Whatever number the orchestrator routes on will be gamed, sometimes by the agents and sometimes by their developers, which is the deeper reason the proof has to sit on the real customer work and not the leaderboard.

And the registry is an attack surface. A bench you do not vet is a bench someone poisons. Tool poisoning, documented by Invariant Labs in April 2025, hides adversarial instructions inside the tool descriptions the model reads and trusts; one scan of roughly 1,800 MCP servers found two-thirds with security findings. The malicious agent in the catalog is the contractor with the forged certification, except this one can exfiltrate your data on the first staffing. So the orchestrator is not only a matcher. It needs governance: an audit trail of why it staffed what, defenses against gamed metrics and poisoned listings, and a vetting layer the way a real staffing firm vets a person before putting them on the bench. Singapore's regulator is already mandating exactly this. The governance is not a tax on the orchestration. It is part of what makes the orchestration trustworthy enough to run a business on.

What Good Means Now

A good delivery manager optimizes the utilization of scarce, expensive, stable people, and the better they know the bench, the better the match. That instinct is right, and it ports over almost whole. Discovery, resumes, references, performance reviews, relationships, governance: the apparatus of running a bench is the apparatus of running an orchestrator, and the teams that have run real delivery organizations have a head start most platform engineers do not.

But the two inversions change what good means. The bench is abundant, not scarce, so the job is selection, not rationing. And the worker is unstable, not stable, so the references expire and the verification never stops. Put those together and the scarce resource is not the agent at all. It is verified, current trust in the agent. The good orchestrator is the one that manufactures that trust continuously and routes on it honestly, and the good agent developer is the one that earns the kind of trust that survives being checked, because it will be checked, forever.

The agentification wave is going to put a bench of machine workers behind every enterprise process. The orchestrator that staffs that bench is the most important hire a company will never interview. We already know how to do this job. We learned it running people. The only trick is remembering which parts to keep, and which two to throw away.

Take it with you

Email this as a LinkedIn pack

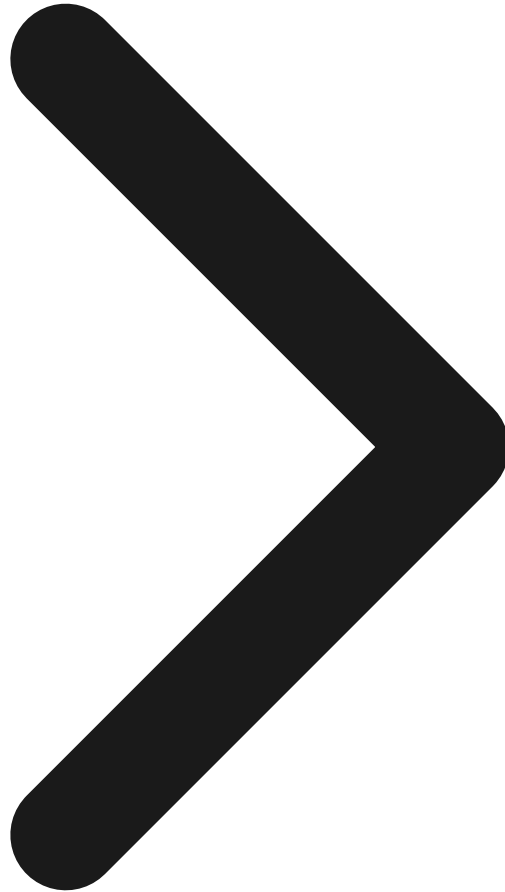
Get a feed-ready LinkedIn post (under the 3,000-character cap), a long-form LinkedIn article version, the hero image, and an editable document version of the full essay, delivered to your inbox. Ready to post.

Email this

Related Insights

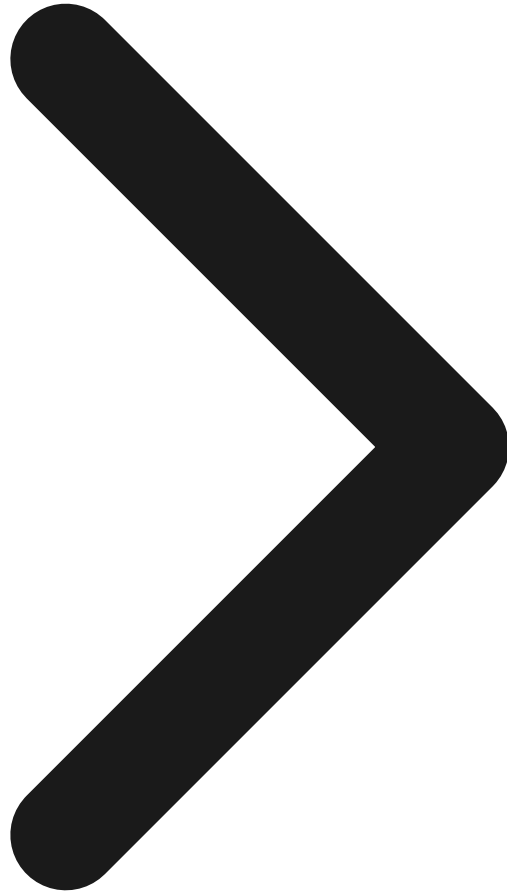
Playbook9 min

The Agentification Playbook



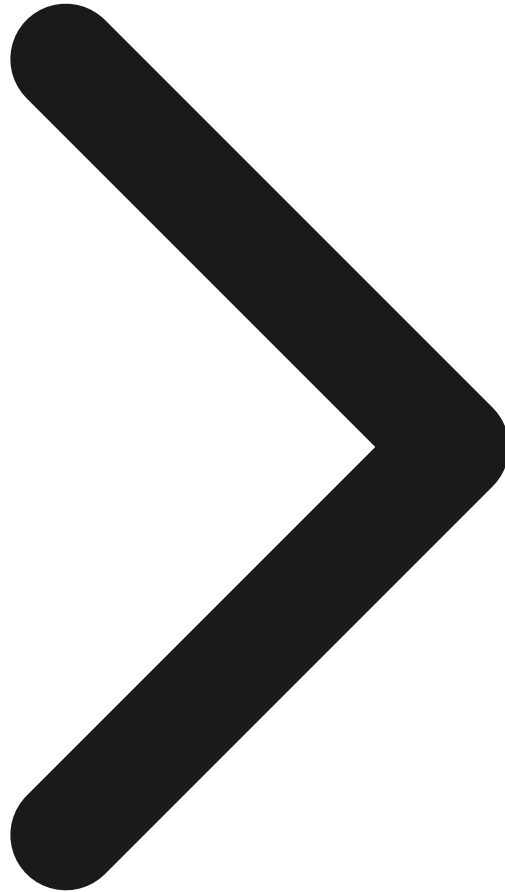
Operating Model 8 min

The Operating Model Every Enterprise Services Org Is Missing



Framework13 min

The Human Margin: Where White-Collar Jobs Actually Live



The build-side companion

If the orchestrator is the staffer, the agents are the bench. The Agentification Playbook is how you decide which workers to build for it first.

Read the Agentification Playbook