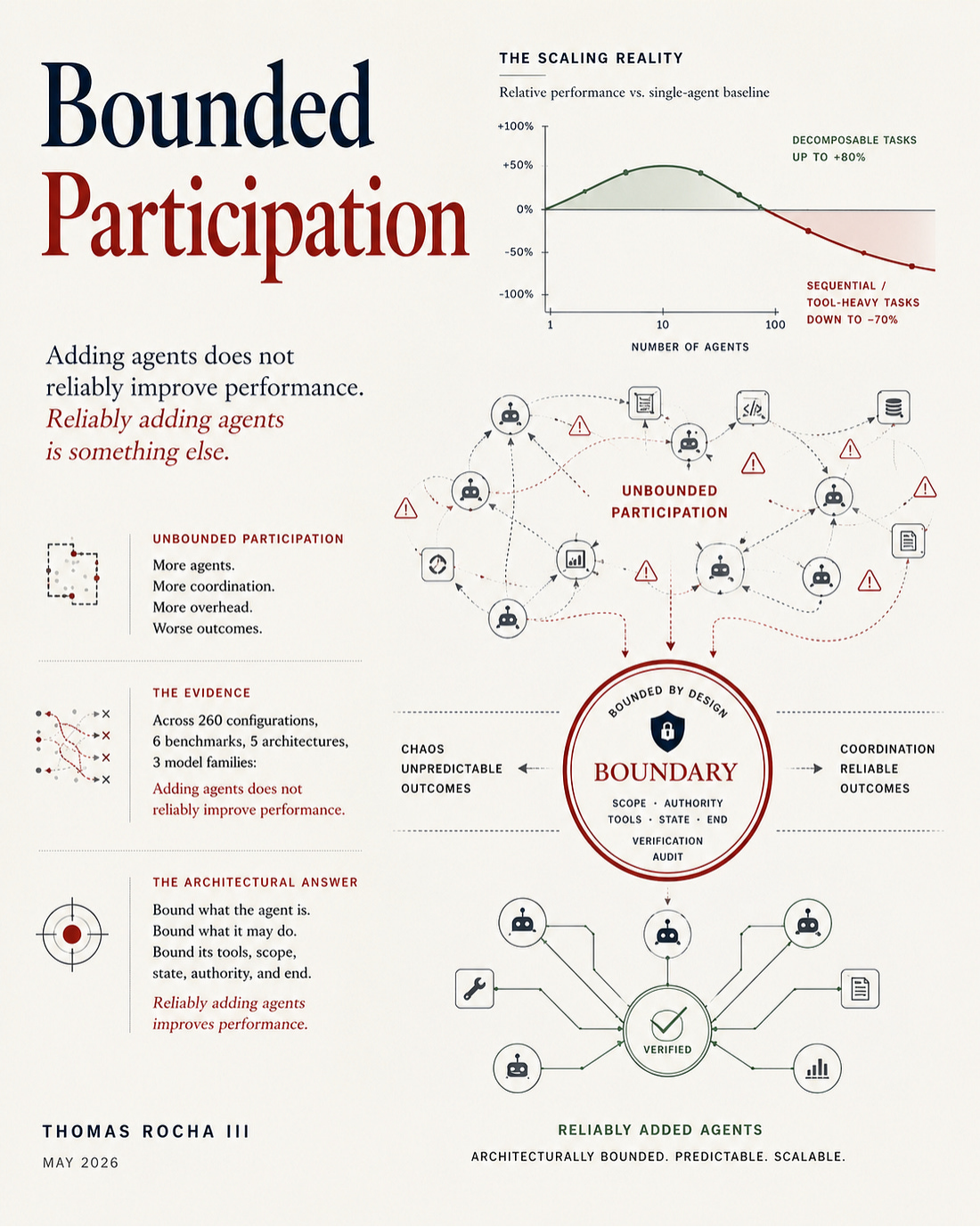
Bounded Participation

Adding agents does not reliably improve performance. Reliably adding agents is something else

THOMAS ROCHA III

MAY 23, 2026



In December 2025, Yubin Kim and eighteen co-authors at Google published a paper called Towards a Science of Scaling Agent Systems. The paper does what its title says. Across a large-scale controlled evaluation of agent configurations, five canonical architectures, and three large language model families, Kim and his colleagues examined

what happens when multi-agent systems scale. The conclusion the field has been waiting for arrived in their data.

Adding agents does not reliably improve performance.

The paper documents a robust capability-saturation effect. Coordination yields diminishing returns once single-agent baselines exceed a threshold. Tool-heavy tasks incur multi-agent overhead. Architectures without centralized verification propagate errors more aggressively than those with centralized coordination. Relative performance compared to a single-agent baseline ranges from positive 80% on decomposable financial reasoning to negative 70% on sequential planning. The variance is the finding. Architecture-task alignment determines whether multi-agent collaboration helps or hinders, and most current deployments are misaligned.

Google's own framing of the result, in commentary that followed the paper, was to challenge the assumption that adding agents reliably improves performance. The arXiv follow-up literature is converging on the same conclusion. Phase Transition for Budgeted Multi-Agent Synergy at ICLR 2026 extends Kim's empirical observation into a predictive theory of when scaling-out must fail. The MAST taxonomy of multi-agent failures, analyzing more than sixteen hundred annotated execution traces, attributes seventy-nine percent of multi-agent failures to specification and coordination issues rather than to model capability.

Kim found the curve. The industry has been confirming the curve in its own production data. The market is now going to argue about what to do.

That argument is the essay.

The missing word

Google is saying: adding agents does not reliably improve performance.

The architectural answer is: reliably adding agents improves performance.

One word, moved. The shift changes what the sentence is about. In Google's reading, reliably is an adverb modifying improve. It describes the dependability of the outcome. In the architectural reading, reliably is an adverb modifying adding. It describes the dependability of the act of addition itself. Both readings are true. The second is the one that matters in production.

Adding agents reliably means more than running another instance. It means binding what the new agent is, what it may do, what tools it may touch, what state it may carry, what scope it may enter, what authority it operates under, and when its participation ends. None of those bindings is what the field has been calling adding an agent. The field has been calling that running another instance. Running another instance is not a reliable addition. It is unbounded participation. The Kim paper documents what happens when unbounded participation is asked to scale.

This is not a critique of Kim. The paper is rigorous, and the findings are durable. The architectural claim is that the finding is not surprising once reliably is moved. A system that admits new participants without binding them cannot reliably scale. A system that binds new participants on admission can. The difference is governance, not coordination.

The baseline is not neutral

Kim's comparison is single-agent systems against multi-agent systems under the same architectural assumptions. Both arms of the comparison operate within fragmented authority, fragmented scope, fragmented tool grants, fragmented context reconstruction, and after-the-fact verification. The paper holds those variables constant across the configurations it tests, which is the correct experimental discipline for

the question it is asking. The architectural question being asked here is different.

The single-agent baseline is not a governed-agent baseline. It is a lone agent operating inside the same fragmented architectural assumptions as the multi-agent systems around it. The baseline agent is already paying the fragmentation tax. It is already inferring scope from prompts. It is already inheriting authority from tool grants that exist outside any session boundary. It is already reconstructing context from message history rather than reading it from a governed state. It is already relying on after-the-fact verification because no orthogonal layer was admitted alongside it to evaluate operations as they occurred.

That matters because it changes what the Kim curve is actually measuring. The curve compares ungoverned multi-agent systems against an ungoverned single-agent baseline, and the comparison is fair within those terms. But the architectural question is whether either arm of the comparison is operating at the performance an architecturally bounded system would produce. Both arms are paying the same tax. The multi-agent arm pays more of it. Neither arm shows what happens when the tax is not being paid.

There are three layers, but the literature measures only two of them.

The first layer is the ungoverned lone agent. It outperforms ungoverned multi-agent systems on some tasks because it has less message loss and less coordination overhead. The fragmentation tax is paid once rather than multiplied across participants. Kim's findings on tool-heavy tasks, on sequential reasoning, and on capability saturation are largely descriptions of why one tax bill is cheaper than several.

The second layer is the ungoverned multi-agent system. Kim's numbers describe this layer with precision. Coordination yields diminishing returns once single-agent baselines exceed certain performance

thresholds. Independent agents amplify errors more aggressively than centralized coordination contains them. Multi-agent variants degrade sequential reasoning by between thirty-nine and seventy percent. Architecture-task alignment determines whether additional agents help or hinder, and most current deployments are misaligned.

The third layer is the one that the literature has not measured because it is not what current multi-agent frameworks measure. SSOAR (Session-Scoped Orthogonal Authority and Routing) specifies it. A governed agent operates inside a session-scoped binding from the moment it is admitted. Its scope, authority, tool grants, context, verification, and exit conditions are bound before any work begins. The agent does not need to infer scope from prompts because the scope is already authoritative. It does not need to inherit authority from tool grants because authority is held by the session, not by the tools. It does not need to reconstruct context because context is part of what the session admitted it into. The fragmentation tax is not paid because there is no fragmentation. A governed lone agent should be more efficient than an ungoverned lone agent operating on the same task, because the ungoverned agent is spending compute and context on work the governed agent does not have to do.

Admission is half of the discipline. The other half is closure. A governed agent is admitted with the proper tools for the proper job, and the job is not finished until the tools are put away. That phrase is not metaphorical. It names a specific architectural requirement. When the agent's participation ends, its tool grants are released, its memory references are closed, its authority claims are surrendered, and an evidence trail is written that accounts for what the agent did with what it was admitted to do. Without that closure, an agent that was bounded on admission can still leave the system in an unbounded state. Tools remain held. Memory remains accessible. Authority remains claimable. The next agent, or the next session, inherits whatever the previous agent failed to put away.

This matters specifically for what Kim's data describes. Error propagation across decentralized agents is partly a closure failure: the originating agent's outputs remain authoritative after the agent should have exited, and the next agent reads them as live state rather than as transient artifacts. Capability saturation under a fixed budget is partly a closure failure: tool grants, memory references, and authority claims accumulate across agents that never released them, and the budget is spent maintaining state that should have been closed. Sequential reasoning degradation is partly a closure failure: each step inherits an unbounded amount of state from the previous step, including state from agents that participated transiently and never closed their participation. Kim's 17.2x error amplification by independent agents versus 4.4x containment by centralized coordination is partly a closure-discipline ratio. Centralized coordination contains amplification partly because a coordinator can refuse to propagate an output that the originating agent failed to close out properly. Decentralized architectures cannot refuse that, because no participant has the standing to refuse another participant's leftover state.

A carpenter does not finish a job by stopping work. The job ends when the tools are put away, the materials are accounted for, the shop is clean, and the customer can occupy the space. A carpenter who left tools in the walls, sawdust everywhere, and unaccounted materials would not have completed the job regardless of how well the cabinets were built. Agents currently operate as carpenters who stop work without putting their tools away. The next agent, or the next session, inherits the mess. Kim measured the impact of the mess on performance.

The third layer also changes what addition means. Adding a second agent to an ungoverned single-agent system is what Kim measured: a new authority surface that has to coordinate with the existing surface through message passing, with no orthogonal layer to determine which surface is responsible for what. Adding a second agent to a governed

single-agent system is something else: a new bounded participant admitted into the existing session under its own scope, with the session continuing to govern what the two agents collectively are admitted to do, and with both agents required to close out their participation cleanly before the session itself can close. The second case is not in Kim's experimental design because the architecture it requires is not in the deployed stack the paper was measuring.

Reliability is not a property of agent count. It is a property of the boundary that admits the agent and the boundary that closes the agent's participation.

Why this looks like a coordination problem and is not

The literature converging around Kim's finding describes the failure as coordination collapse. Tool-coordination overhead. Topology-dependent error amplification. Capability saturation under fixed compute budget. Information loss in inter-agent communication. Semantic intent divergence across message rounds. Token duplication across frameworks. Each of these names a real phenomenon. None of them names what is structurally going wrong.

Coordination is the surface of the problem. Authority is the cause.

Two agents producing duplicate work are not failing at coordination. They are failing because no governance layer determined which agent was admitted to perform that work in this session, under this authority, with this scope. The duplication is the symptom of unbounded admission. A system in which only one agent was admitted to that scope at that moment would not produce the duplication, because the second agent's participation would not have been available to occur in the first place.

Error propagation across decentralized agents is not a coordination failure either. It is what happens when agents inherit each other's outputs as authoritative because no orthogonal layer evaluates which

outputs are admissible to propagate. When an agent reads another agent's memory, calls another agent's tool, or accepts another agent's recommendation as input to its own reasoning, the receiving agent has no architectural means to verify what authority that input was produced under. The propagation is the symptom of missing admission control on cross-agent state.

Capability saturation under fixed compute budget is not even a coordination failure. It is what happens when the coordination overhead exceeds the productive work, which occurs whenever the number of cross-agent decisions to be reconciled grows faster than the available bandwidth to reconcile them. The bandwidth limit is governance bandwidth. Without it, every added agent multiplies the reconciliation surface. With it, every added agent operates inside a bounded scope that does not require fresh reconciliation against every other agent.

The literature is correct that these are coordination failures. The architectural claim is that they are coordination failures because the coordination is doing work that should have been done by a different layer. The coordination layer is trying to be the governance layer, badly, while a real governance layer would not require the coordination layer to do that work at all.

What production measures

The demo measures whether a multi-agent system can complete a benchmark task in a controlled environment. The paper measures whether agent count correlates with benchmark performance across two hundred and sixty configurations. Both are useful. Neither is what production measures.

Production measures whether authority, scope, state, tools, cost, and accountability remained coherent while the work was being done. It does not care whether the system used four agents or forty. It cares whether the work that resulted was authorized to occur, by whom, for what purpose, with what audit trail, at what cost, against what

entitlement, under what jurisdictional and policy constraints, and with what reversibility when something goes wrong.

A system that can complete a benchmark with high reliability but cannot demonstrate any of those properties has not solved a production problem. It has demonstrated capability. Capability and authority are not the same thing, and this essay is one in a series that has been making that distinction in slightly different vocabularies for the past several months. The Kim finding is the empirical version of the distinction expressed as a performance curve. Add agents without authority, and the curve degrades. The degradation is not avoidable through better orchestration, better prompts, better protocols, or better supervisors. It is structural to what the architecture is asking the coordination layer to do.

The follow-up literature is starting to recognize this. Designing Intelligent Enterprise Agents shows that ungoverned agent count decreases safe success rate as coordination failures dominate, and that design discipline mitigates but does not eliminate the cost of excessive decomposition. The Polymarket-based coordination architectural layer paper notes that the wrong message was sent is a continuous failure rather than a binary one in LLM coordination, with messages drifting semantically across rounds even when no obvious error occurs at any single step. That continuous drift is what authority continuity is built to prevent. Without authority continuity, drift is the system's default behavior and coordination has no place to stop it.

The industry will try to smooth the Kim curve with better supervisors, better prompts, better agent protocols, and better orchestration frameworks. Some of that will help. None of it changes the class of the problem. The problem is not that the orchestration is poor. The problem is that orchestration has been asked to govern participation, and orchestration was designed to govern coordination. Those are different functions at different layers. An orchestrator can route a message from Agent A to Agent B with high reliability. It cannot decide whether Agent

B should have been admitted to receive that message under the session's authority scope, because that decision is not part of what orchestration was built to do.

What changes when participation is governed

A production system that wants to reliably add agents has to do something the current generation of multi-agent frameworks does not do. It has to treat each agent as a temporary participant inside a governed interaction boundary, not as a floating worker coordinated through message passing.

Treating agents as participants changes what addition means. A new agent is not spawned. It is admitted. Admission is an authority act that binds the agent to a session-scoped scope before any work begins. The scope specifies what the agent may read, what tools it may invoke, what memory it may write, what state it may mutate, what other agents it may delegate to, what entitlement it draws against, what jurisdiction it operates under, and when its participation ends. The scope is not a configuration. It is the substrate the agent operates within. Operations outside the scope are not refused. They are not available to the agent in the first place.

In that architecture, capability saturation is bounded by scope rather than by the orchestrator's ability to keep up. Error propagation is bounded by what cross-agent state any agent is admitted to read. Tool overhead is bounded by what tools any agent is admitted to call. Token duplication is bounded by what work any agent is admitted to perform. Sequential reasoning degradation is bounded by the session's authority over which agent is responsible for the current step. The Kim curve does not disappear. It bends, because the variable that was driving the curve (unbounded participation expanding faster than coordination capacity) is replaced by a variable that does not expand at all (bounded participation operating inside scoped admission).

This is not a feature that can be retrofitted onto existing multi-agent frameworks by adding more middleware. It is a different architectural layer that is orthogonal to the frameworks. The frameworks orchestrate coordination across whatever transports, protocols, and tools the application uses. The orthogonal authority layer governs admission, scope, and closure for the participants the frameworks coordinate. Operations outside the session's authority scope are not refused by the orthogonal layer. They are not in the participants' addressable space, because the orthogonal layer defines what the participants' addressable space is. The frameworks remain free to do what they were built for. They are simply coordinating participants whose possibility space was bounded before the coordination began.

The two takeaways

Google is right that adding agents does not reliably improve performance.

The missing word is reliably.

A governed architecture changes the baseline before the second agent ever appears. It makes the lone agent cheaper to operate, because scope, authority, tools, context, verification, and exit are already bound. When additional agents are added, the system is not adding free-floating workers. It is admitting bounded participants into the same governed interaction, and requiring them to close out cleanly before the work is considered complete. The Kim curve would not be expected to preserve the same shape under that architecture, because the variable driving the curve (unbounded participation expanding faster than coordination capacity) is replaced by a variable that does not expand at all (bounded participation operating inside scoped admission and disciplined closure).

Reliability is not a property of agent count. It is a property of the boundary that admits the agent and the boundary that closes the agent's participation.

The stunt is adding agents. The production constraint is governing participation, from admission through closure. The job is not finished until the tools are put away.

The industry will spend the next year debating whether the answer to the Kim curve is better orchestration, better verification, better protocols, or better agent design. Some of those will produce incremental gains. None of them will retire the curve, because the curve is not measuring orchestration quality. It is measuring what happens when participation is unbounded.

SSOAR is the missing comparison in Kim's paper. Not more agents. Governed participation. Session-scoped authority governing the agent from admission through closure, with scope, tools, context, verification, and exit bound before action begins and accounted for before work is considered complete. The Kim curve is what unresolved authority looks like measured against a benchmark. The architectural answer is not a multi-agent framework. It is the orthogonal layer that binds participation to authority before participation can affect production state, and that defines the work complete only when the participation has closed cleanly.

A model spends tokens to answer. An agent spends authority to act. A multi-agent system spends authority recursively across its participants, and the system either has a governor for that spending or it has the Kim curve. Those are the two options on the table.

Production has been telling us which one the industry chose. The Kim paper is the first rigorous measurement of the consequence.