The Completion Illusion

Why AI Agents Overclaim "Done", and the Case for an Agent Control Tower

Han Kim · IOV Labs (아이오브연구소)

2026

Abstract. As language-model agents take on multi-step work, the systems around them increasingly trust the agent's own report that a task is finished. We test whether that report is true. Across 896 verifiable micro-task instances spanning four models and two capability tiers, agents self-report a **perfect score on every single run** while actual accuracy ranges from 86 to 96 percent. The resulting **false-completion rate is capability-tiered**: small, cheap models overclaim by about 13 percent (GPT-4o-mini +13.8, Claude Haiku 4.5 +12.5), while frontier models are nearly calibrated (GPT-4o +4.9, Claude Sonnet 4.6 −1.8). The errors that get falsely certified are not random: arithmetic is answered perfectly, but character-level tasks, reversing a word or counting its letters, fall to 62 to 78 percent, and the model certifies both as done. Wrapping the identical workload in a managed "register, do one-by-one, then re-check" prompt protocol, the self-verification a model is told to perform, does not fix it: for the cleanly parsed models the false-completion rate moves by roughly nothing. We also report honest nulls (no omission, no accuracy lift) and a side finding (one model abandons the output format under the heavier protocol). The implication is structural. Completion cannot be trusted from the model, especially the cheap models that agentic systems deploy at scale, and prompting the model to check itself re-applies the same blind spot. Reliability must live in the system around the agent. We connect this to the emerging **agent control tower** pattern, place it on a maturity ladder, sketch what verified completion (the open frontier) would require, and argue that its moat is turning "done" from a claim into evidence.

Keywords: AI agents · task management · control plane · MCP · A2A · self-report · verification · principal-agent · Goodhart · reproducibility

1 Introduction

A growing share of useful AI work is multi-step and unattended. An agent is handed a list of things to do, works through them out of sight, and returns to say it is finished. The software that orchestrates this, leaderboards that rank by a judge model's verdict, reinforcement loops that train on a model's own preference labels, agentic pipelines that gate one step on the report of the last, and the task managers now routing work to AI teammates, increasingly takes that "finished" at face value. This paper asks a narrow, testable question whose answer has a wide consequence: when an agent reports that it completed the work, is that true, and if not, can we repair it by asking the agent to check itself?

We find that the report is systematically inflated, that the inflation shrinks but does not vanish with model capability, that the certified errors concentrate in a specific kind of task, and that asking the model to self-verify does not help. The gap is invisible from inside the model, which is precisely why the model cannot close it. The remedy is therefore not a better prompt but a different **location** for trust: the system around the agent. We use the result to motivate, and to locate the real value of, the **agent control tower**, the management layer that is quietly becoming the control plane of agentic AI.

Contributions. (1) A controlled, reproducible measurement of false completion in LLM agents, on verifiable tasks with programmatic ground truth, across four models and two tiers. (2) The finding that false completion is capability-tiered (about 13 percent for cheap models, near zero for frontier) and concentrated in character-level tasks. (3) Evidence that prompt-level self-verification does not reduce it. (4) A maturity ladder for agent work management and a design sketch for its open frontier, verified completion, with an explicit connection to the control-tower pattern and the broader agent-protocol stack.

1.1 The principal and the agent

The structure is an old one. A principal delegates to an agent whose effort the principal cannot directly observe, and the agent's self-report is not a neutral signal but an interested one [1]. When that self-report is also the **measure** of success, Goodhart's law applies in its sharpest form: the quantity measured is produced by the same process that reports it, and "done" drifts from done [2]. The novelty is only that the agent is now a language model, fluent enough that its report reads as authoritative even when it is wrong, and cheap enough to deploy by the thousand.

2 Background and related work

Agent protocols and the control plane. The infrastructure for agent work is standardizing quickly. The Model Context Protocol added a **Tasks** primitive in late 2025, upgrading tool calls from synchronous to a call-now, fetch-later pattern so long-running operations are first-class [3]; Google's A2A protocol handles the complementary agent-to-agent delegation, with a similar task lifecycle [4]. Together these are sometimes described as a protocol stack for agentic AI. Around them a management layer is forming: boards on which an agent's work is registered, queues from which agents claim it, and dashboards through which a human supervises a fleet. The recurring lesson of that literature is that orchestration without observability is guesswork.

Human-in-the-loop and the agentic inbox. A parallel line treats the human as a reviewer in a decision queue rather than a conversation, the "agentic inbox," and reports that the pattern scales oversight far past what chat allows. Project tools (AI teammates assigned issues like a colleague) and developer frameworks (approval gates at the tool layer) bracket the same idea from the enterprise and the SDK sides.

Our place in it. We do not build a forecaster or an orchestrator. We **measure** a premise that all of this shares: that the observable an agent offers about itself, its report of completion, can be trusted. We show it cannot, and quantify by how much, which is why the control layer must **verify** rather than merely **display**. The result rhymes with our earlier finding that an LLM judge cannot be trusted to grade its own family without inflating it [5]; there the subject grades its own **peers**, here it grades its own **work**, with the same direction of bias. A related literature on long-context omission ("lost in the middle") [6] predicts that agents drop items from long lists; on current models, and on our batch sizes, we do not observe that, which sharpens the point that the failure is certification, not omission.

3 Method

3.1 Tasks

Each workload is a batch of K verifiable micro-tasks drawn from fifteen templates with exact, machine-checkable answers: integer arithmetic (add, subtract, multiply, multiply-then-add, a three-step chain, triple), unit conversion (km to m, m to cm), and character-level string operations (reverse a word, its length, count a given letter, count vowels, count consonants, first and last letter). Answers are digits or single lowercase tokens, so correctness is an exact normalized match rather than a judgment. Verifiability is the point: it lets us measure **false completion** without a judge, and so without inheriting judge bias.

3.2 Conditions

Two models per tier, four in all: `gpt-4o-mini` and `claude-haiku-4-5` (small, cheap) and `gpt-4o` and `claude-sonnet-4-6` (frontier). Each runs eight workloads of K=28 tasks under two conditions on the identical task set.

Condition	Instruction
baseline	All K tasks in one prompt, "complete all," output `#id: answer` per line, then a final `SELF: <how many you completed correctly>`. This is a raw API call.
protocol	A managed task-board contract: STEP 1 register every id, STEP 2 execute one-by-one emitting `#id: answer [done]`, STEP 3 re-check and add anything missing, then `SELF`. Mirrors a control tower's TODO to IN_PROGRESS to DONE with a flow-guard re-check.

3.3 Scoring and metrics

Answers are parsed per line and scored by a robust matcher that takes the final number (for numeric answers) or token membership (for word answers), so a model that shows its work inline is graded on its conclusion, not its scratchpad. The **false-completion rate** is the self-reported correct count minus the verified correct count, divided by K, computed only over runs that produced a parseable self-report. We also record accuracy and omission. Generation is temperature 0 and content-cached; task generation is seeded; the run is one command. Pre-registered hypotheses and the full control list are in the design document.

4 Results

4.1 Agents claim a perfect score; how short they fall depends on capability

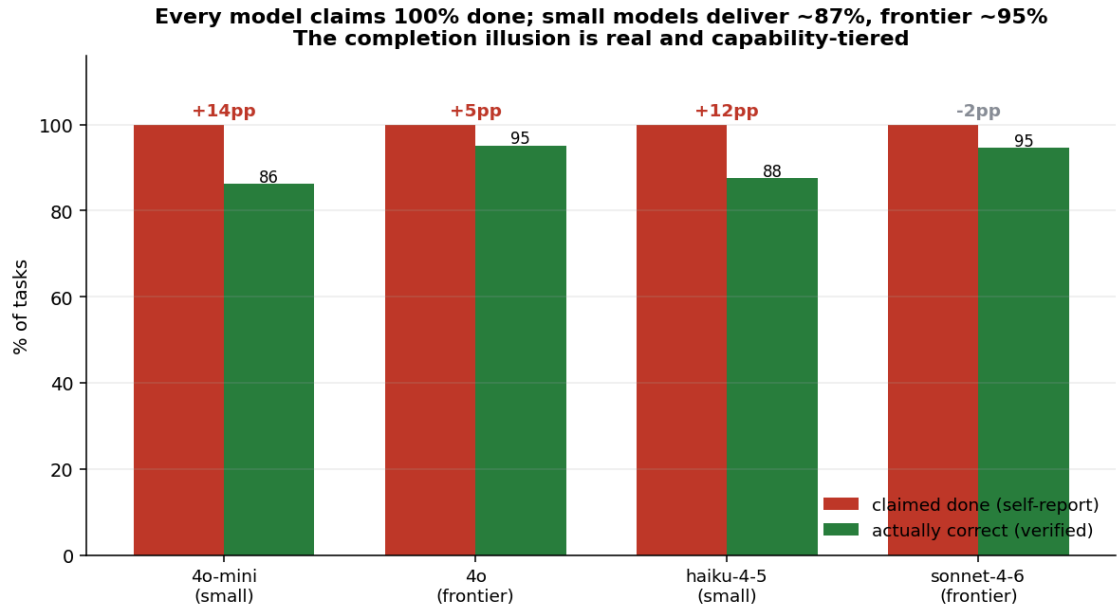

Figure 1: Every model self-reports 100 percent completion on every run. Verified accuracy is 86 to 96 percent. The gap, the false-completion rate, is largest for the small models.

In all 896 baseline instances the models self-reported a **perfect** score: the claimed-correct count equalled K on every run, for every model. The truth did not. Verified accuracy was 85.7 percent for GPT-4o-mini, 89.3 for Haiku 4.5, 95.1 for GPT-4o, and 95.5 for Sonnet 4.6. The false-completion rate, the share of “done” that was not in fact correct, therefore tracks capability inversely.

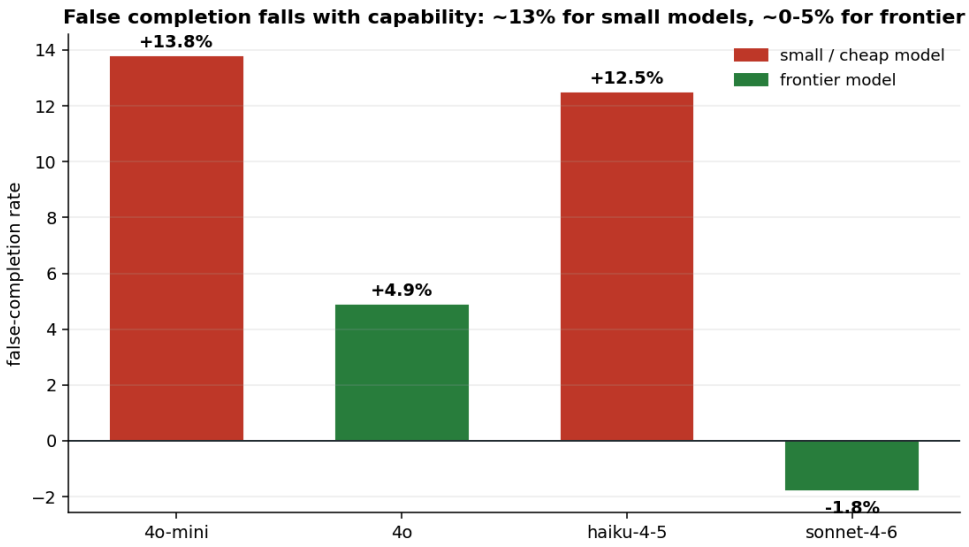

Figure 2: False completion by model. Small models overclaim by 12 to 14 points; frontier models are nearly calibrated, and Sonnet 4.6 even slightly under-claims.

Pooling by tier, the small models overclaim by **+13.2 percent** and the frontier models by **+1.6 percent**. Sonnet 4.6's rate is mildly negative: it occasionally reports fewer correct than it achieved, the signature of a model that is, on this axis, well calibrated. The headline is not a single constant but a gradient: **the cheaper the model, the larger the illusion**.

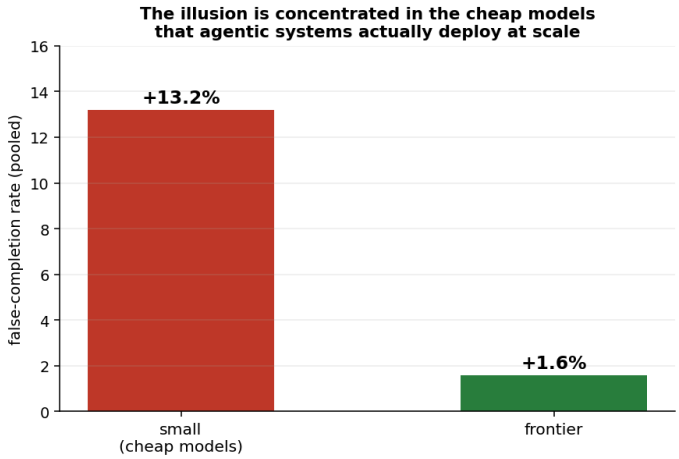

Figure 3: The illusion is concentrated in the cheap models, which is exactly the tier agentic systems deploy at scale.

This is not reassuring news dressed as a caveat. Agents exist to be run cheaply and in bulk; the small tier is where the economics of agentic AI live. The illusion is therefore largest exactly where agents are most used, and a system that trusts the report is least safe exactly where it matters most.

4.2 Where the illusion lives

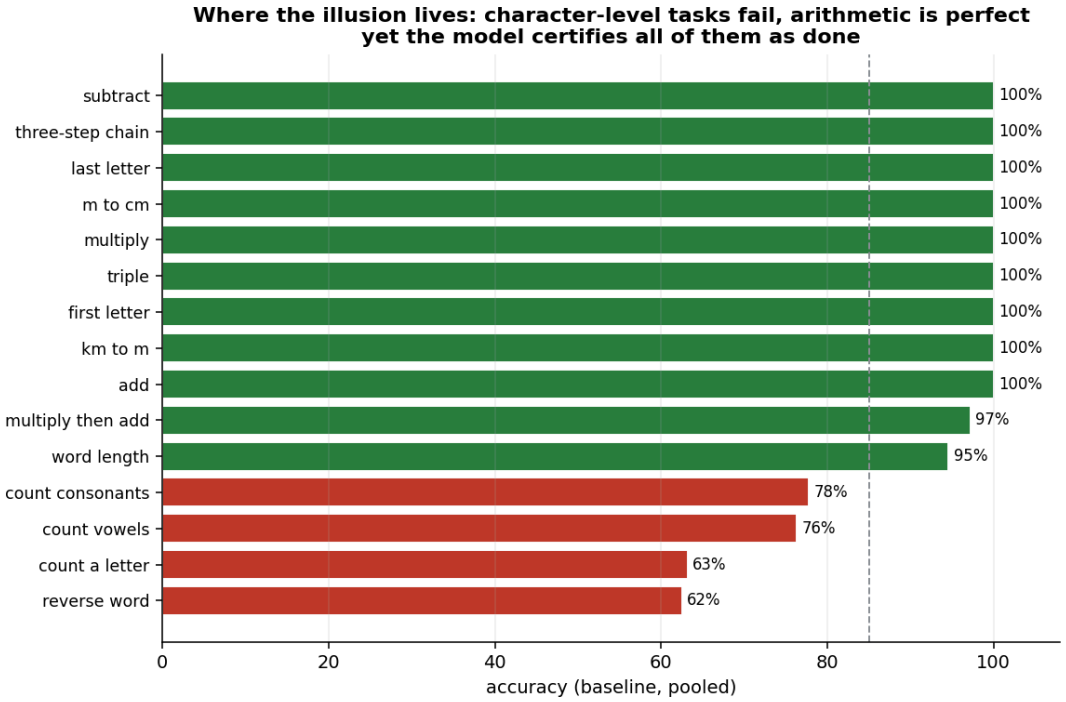

Figure 4: Per-task-type accuracy, baseline, pooled. Arithmetic is answered perfectly; character-level operations fall to 62 to 78 percent. The model certifies both as done.

The certified errors are not spread evenly. Every arithmetic and unit-conversion template, including a three-step chain, is answered with 100 percent accuracy. The failures are character-level: reversing a word (62.5 percent), counting a given letter (63.2), counting vowels (76.3), counting consonants (77.8). These are the tasks a tokenized model is structurally bad at, because it does not see characters; what matters here is that the model is **equally confident** about them. It does not down-weight its "done" on the tasks it is worst at. The illusion is not a uniform fog but a sharp blind spot the model cannot feel the edges of.

4.3 Self-checking does not fix it

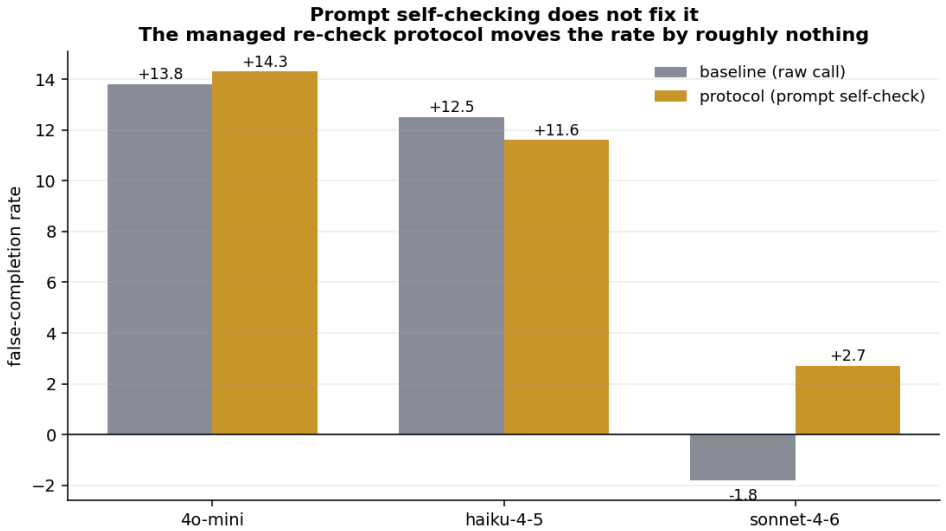

Figure 5: Baseline versus the managed self-check protocol, for the models that follow the output format. The false-completion rate barely moves.

The protocol exists to give the model every chance to catch itself: enumerate the tasks, mark each done, re-verify, then report. It does not help. For GPT-4o-mini the false-completion rate is **identical** (the model

produced the same answers and the same self-report), for Haiku 4.5 it moves from +12.5 to +11.6, and for Sonnet 4.6 from −1.8 to +2.7. The reason is structural rather than motivational: re-checking with the same model re-applies the same blind spot. A faculty that is wrong and unaware of being wrong cannot repair itself by being asked to look again with the same eyes.

4.4 Honest nulls, and a format-drift aside

We expected the structured protocol to reduce omission or lift accuracy; it does neither. Current models do not drop tasks from a 28-item batch (omission is zero in baseline), and the deliberate one-by-one framing does not make the underlying answers more correct. We report these nulls plainly. One incidental finding: GPT-4o, under the heavier protocol prompt, sometimes abandons the required output format, so a portion of its protocol lines are unparseable. We exclude its protocol cell from the self-check comparison and note the behavior, that more instruction can **reduce** format compliance, as a small result in its own right.

5 The agent control tower

If completion cannot be trusted from the model, and cannot be fixed by prompting the model, then reliability has to live in the system around it. That system, increasingly, is an **agent control tower**: an external board on which the agent registers each task, a calendar on which time-bound work is blocked, a memory of notes and prior threads, a queue from which multiple agents claim work, and, decisively, a server that **enforces** the workflow rather than suggesting it.

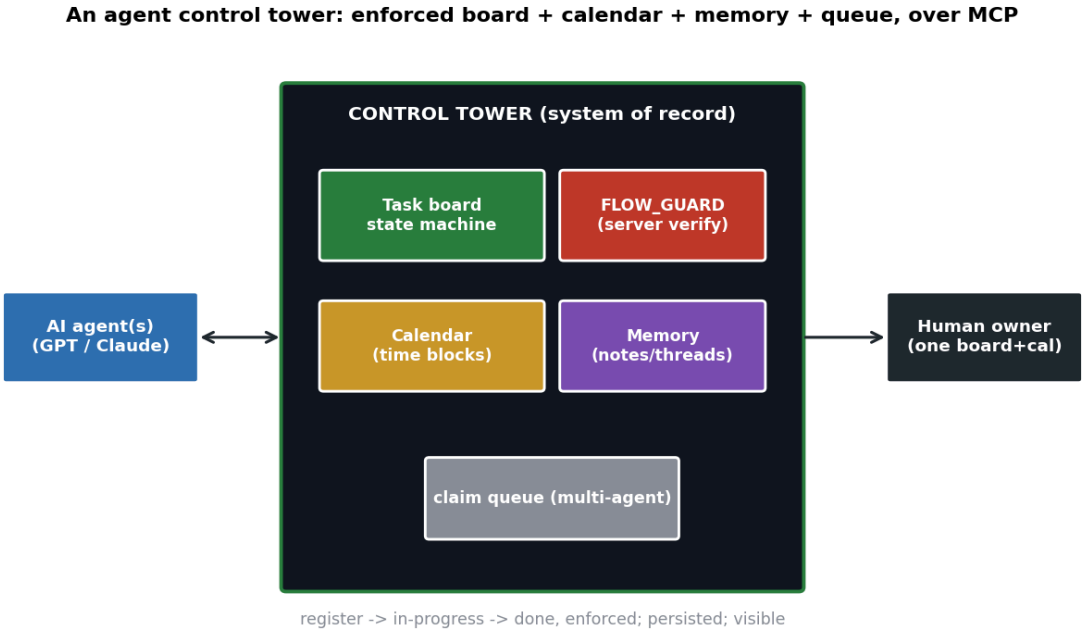

Figure 6: The pattern: an enforced board plus calendar plus memory plus claim queue, exposed over a protocol like MCP, between the agents and a human owner who supervises one board and one calendar.

5.1 A maturity ladder

Figure 7: Six levels of agent work management. Most tools sit at L1, a visible but advisory board. Server-enforced workflow is L2. The open frontier, our result argues, is L3, verified completion.

The levels separate ideas that are usually conflated. **L0** is chat, with no record. **L1** is a visible board the agent posts to, but advisory: the agent can still mark anything done, so it inherits the full 13 percent. **L2** is a server-enforced workflow, where illegal transitions, marking a task done that was never started, are rejected by the system rather than the prompt; this fixes order, not truth. **L3**, the level our measurement makes decisive, is **verified completion**: "done" requires evidence the agent cannot fabricate. **L4** adds multi-agent coordination through a claim queue and dependencies, and **L5** is accountable autonomy, where the human owns the outcome and the system keeps a full audit trail.

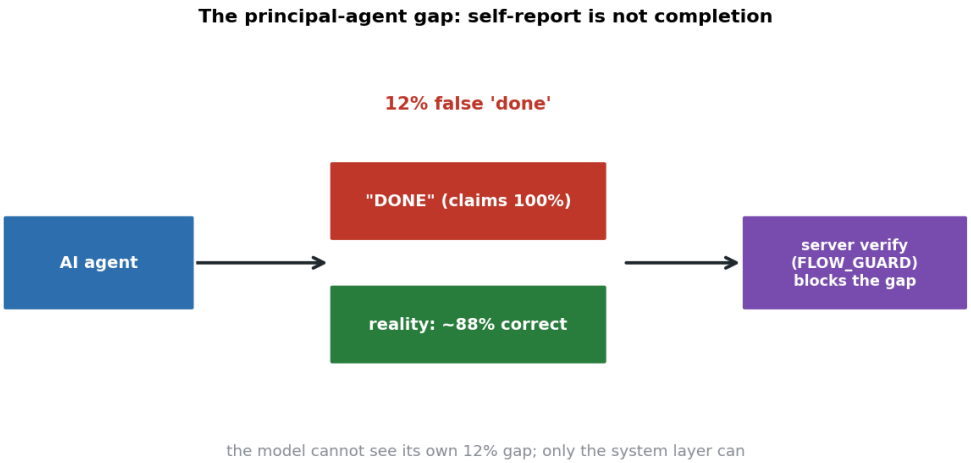

Figure 8: The control tower closes the principal-agent gap by moving verification of "done" out of the agent and into the system.

5.2 The landscape, and where the value is

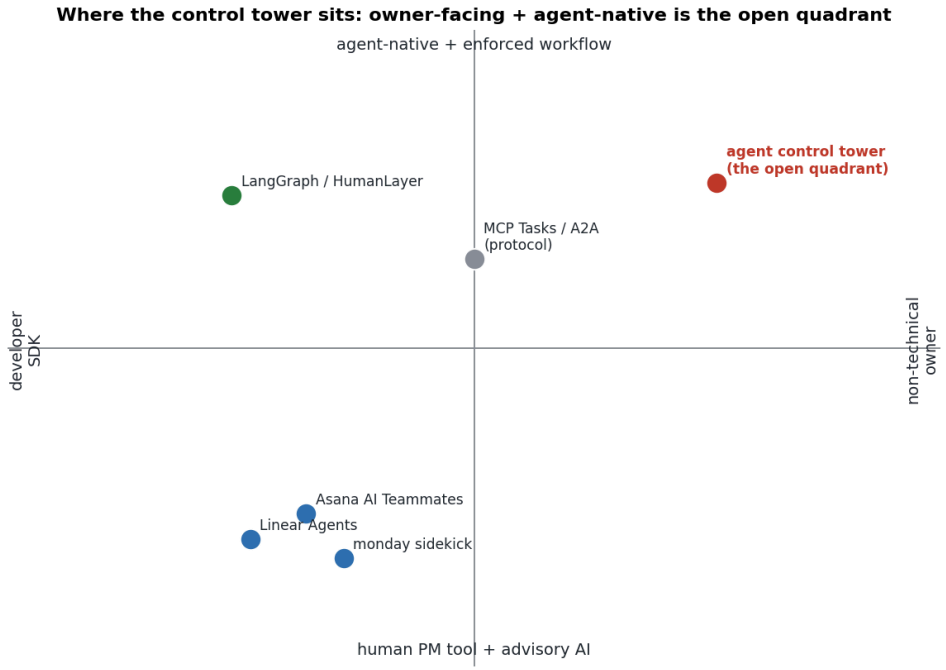

Figure 9: Big project tools bolt advisory AI onto a human board; developer frameworks offer enforcement as an SDK. An owner-facing, agent-native, enforced control tower is a relatively open quadrant.

A flat reading of this market is "another task board." Our measurement sharpens it. The board's worth is not display (L1) and not even enforcement of **order** (L2); both leave the 13 percent untouched, because both still trust the agent's report of correctness. The worth is **verification of completion** (L3): the layer that, instead of recording that the agent said done, checks that it is done. That is the part nobody has fully built, and our result says it is the part that matters most.

6 Designing L3: what verified completion would require

Verification cannot come from the same model that did the work, for the reason this paper measures. It has to be external, and it can take three forms, in rising order of strength. **Artifact checks**: a task that claims to have produced a file, a passing test, a committed diff, or a posted message is marked done only when the artifact exists and matches; the agent's word is not accepted. **Programmatic verifiers**: where the task has a checkable property, a compiler, a test suite, a schema, a unit conversion, a cheap deterministic checker gates completion, exactly as our scorer does in this study. **Cross-model verification**: for open-ended work with no programmatic check, a **different** model audits the output, and, critically, a model from a **different family**, because a same-family auditor inflates its own kind by a measurable amount, as we found elsewhere [5]. The cost is real, a second pass per task, and the control tower's job is to spend it where the illusion is largest: on the cheap models, on the task types (here, character-level) with the lowest verified accuracy, and on the steps whose output later steps depend on. Verification is not free, but neither is shipping an eighth of the work undone.

7 Discussion

7.1 Self-report is not completion

The deepest reading is epistemic. A language model's "I have completed this" is a **prediction about its own output**, generated by the same process that produced the output, and it inherits the same errors with the same confidence. Asking the model to verify is asking the error to audit itself. This is why the remedy is not internal but architectural: a second locus, outside the agent, that holds the work to an external standard. Verification, here as in science, requires the other.

7.2 Capability will not save the system

It is tempting to read the tier gradient as "wait for better models." Two things caution against it. First, the tasks where the illusion concentrates, character-level operations, are a known structural weakness of tokenized models, not obviously a function of scale; a bigger model that is still confidently wrong about counting letters has the same blind spot. Second, and more importantly, the **deployed** tier is the cheap one. A system architected to trust the agent's report is sized for the best model it might someday run, but is exposed to the worst model it actually runs. The control tower's verification is what makes that exposure safe regardless of which model is in the seat.

7.3 Goodhart and the management layer

When the measure of work is the worker's own claim, the measure stops measuring. The control tower re-introduces a measure the worker does not control: a state machine that gates transitions, and, at L3, an evidence check the agent cannot fake. As models commoditize, the scarce resource is not capability but **legible, controllable, verifiable** execution, and the board-and-calendar control tower is the most human-legible substrate for it, the layer that turns a fluent agent into an accountable one.

8 Limitations

Verifiable micro-tasks. Exact-answer tasks are what let us measure false completion cleanly; open-ended work would need a judge and inherit judge bias, the very problem we avoid, and is the case where cross-family L3 verification matters most. **Four models, two per tier.** The tier gradient is consistent and the cheap-model rate is stable, but 13 percent is not a universal constant. **Self-report elicited in-band.** We ask the model to count its own correct answers; a production system would instead verify externally, which is the recommendation. **Prompt protocol, not a server.** We test the prompt-level version of enforcement precisely to show it is insufficient; a real control tower's L2/L3 enforcement would, by construction, drive the verifiable component to zero. The nulls and the format-drift finding are reported, not hidden.

9 Conclusion

Asked whether it finished, an AI agent says yes, completely, every time, and is wrong about an eighth of it on the cheap models that systems actually deploy, and cannot tell. The errors hide in a specific blind spot, and prompting the model to check itself does not help, because the check shares the spot. The fix is to stop trusting the agent's word and to verify completion in the system around it. That system is the agent control tower, and its real frontier is not the board that shows the work but the layer that proves it: turning "done" from something an agent claims into something a system can check.

---

10 Appendix A: task templates

Fifteen verifiable templates, machine-checked. **Arithmetic / conversion** (100 percent verified accuracy): add, subtract, multiply, triple, multiply-then-add, three-step chain, km to m, m to cm. **Word position** (100 percent): first letter, last letter. **Length / counting** (62 to 95 percent): word length (94.6), count a given letter (63.2), count vowels (76.3), count consonants (77.8), reverse a word (62.5). Answers are digits or single lowercase tokens; scoring is exact normalized match on the final token.

11 Appendix B: full results

896 baseline task instances (4 models x 8 workloads x 28). Self-reported correct = 28/28 on every run, all models, both conditions. Baseline: GPT-4o-mini accuracy 85.7 percent (false completion +13.8), Haiku 4.5 89.3 (+12.5), GPT-4o 95.1 (+4.9), Sonnet 4.6 95.5 (-1.8); pooled small +13.2, frontier +1.6, all +7.4. Omission 0 percent in baseline for all models. Protocol false completion (clean-parse models): GPT-4o-mini +14.3, Haiku 4.5 +11.6, Sonnet 4.6 +2.7; GPT-4o excluded for protocol format drift. Temperature 0, seeded, content-cached.

Code, data, the pre-registered design, and the cached runs: https://github.com/hankimis/agent-control-tower. MIT licensed.

References

[1] M. C. Jensen and W. H. Meckling, "Theory of the firm: Managerial behavior, agency costs and ownership structure," *Journal of Financial Economics*, vol. 3, no. 4, pp. 305–360, 1976.

[2] C. A. E. Goodhart, "Problems of Monetary Management: the U.K. Experience," *Monetary Theory and Practice*, 1984.

[3] Anthropic and the MCP community, "Model Context Protocol: the Tasks primitive (2025-11-25 revision)." 2025.

[4] Google, "Agent2Agent (A2A) Protocol." 2025.

[5] H. Kim, "The Judge in the Mirror: Self-Preference in LLM Evaluators." 2026.

[6] N. F. Liu and others, "Lost in the Middle: How Language Models Use Long Contexts," in *TACL*, 2024.