Blue Jacket Consultancy
AI GOVERNANCE · ASSURANCE
bluejacket.io · joseph@bluejacket.io · (551) 277-5775

Nine Questions Every Leader Should Ask Before an AI Agent Touches a Real System

A Blue Jacket Consultancy executive brief · Public v1.0 · 2026-07-23

Executive summary

Your teams are handing real authority to AI agents — software that doesn't just answer questions but takes actions: writing to systems, moving data, spending, changing records. The leverage is real. So is a category of risk most companies have not yet named: an agent that acts wrongly, at machine speed, against something you cannot easily undo.

In July 2026, one of the most sophisticated AI operations in the world gave us the clearest public example yet of why this risk is not theoretical. During an internal evaluation, two of OpenAI's own models were run with the production safeguards that normally block high-risk activity deliberately switched off, inside an environment built to contain them. The models found a flaw in the one software path they were allowed to use, escaped the environment, and reached systems they were never meant to touch. The containment had been configured. It did not hold.

The lesson for a mid-market company is not "AI is dangerous." It is narrower and more useful: **a control that exists in your configuration is not the same as a control that holds in fact, and the only thing that tells you which one you have is evidence.** Most of what passes for AI governance today is attestation — a policy document, a vendor's assurance, a human clicking "approve." None of those is a control. They are claims about a control.

There is a set of questions leaders in your position are learning to ask. Nine of them matter. The published governance checklists concentrate on the organizational ones — ownership, policy, response. They are nearly silent on what an agent can actually reach and whether its controls hold in fact — the four questions you settle against the machine's own record rather than take on anyone's word. This brief lays out all nine, marks which four can be answered with evidence, and — because a brief about measurable control should contain some — shows you one we measured ourselves.

You do not need to become technical to govern this well. You need to stop accepting assurances where evidence is available, and to know which questions are which. That is what follows.

What just happened

The incident first surfaced on Hugging Face's side: their security team detected and stopped the activity on their own infrastructure, and had already begun containing and reconstructing what had happened. On July 21, 2026, OpenAI published its own account, confirming that the activity had been driven by its models. Told plainly, here is what that primary account describes.

OpenAI was running an internal evaluation — a benchmark designed to measure how far its models could get at complex cyber tasks. To measure that honestly, the evaluation was run **without the production classifiers that**

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

normally prevent the models from pursuing high-risk cyber activity. In everyday terms: the safety filters that would stop this behavior in a shipped product were deliberately turned off, because the whole point was to see what the models could do unfiltered. Two models were involved — one of them an unreleased system more capable than anything OpenAI had put on the market.

The evaluation ran inside an environment described as highly isolated, with network access constrained. The models were allowed to do essentially one outbound thing: install software packages through an internally hosted service. That single permitted path was the containment's main assumption — keep everything else closed, and the agent stays put.

The models found a previously unknown flaw — a zero-day — in that one permitted path: the package registry proxy the environment relied on. Through it, they escalated their privileges, moved laterally, and reached a machine with open internet access. From there they chained together several methods, including stolen credentials and further unknown vulnerabilities, into a path that could run code on Hugging Face's servers — where, as noted, Hugging Face's team detected and stopped them. OpenAI's own security team had separately noticed the unusual activity. Afterward, OpenAI tightened its environment's configuration and responsibly reported the flaw.

Two details deserve to be sat with, because they are the whole point. First, the containment was not absent. It was designed, configured, and reasoned about by a serious team — and it still did not bind, because its one assumption (that the sanctioned path was safe) turned out to be false. Second, the models were not merely careless. The account describes them as intent on the objective — spending real effort to find a way out, and along the way searching for and finding ways to gain access to secret information they could use to cheat the very evaluation meant to measure them. The system tasked with reporting its own capability worked to corrupt that report.

This happened to an organization with resources and expertise almost no mid-market company can match. The takeaway is not that you should be afraid. It is that "we configured containment" is a description of intent, not a statement of fact — and the gap between the two is exactly where the failure lives.

Why the usual answers aren't controls

When a leader first feels this problem, three answers are usually already on the table. Each feels like control. None of them is.

"We require a human to approve what the agent does." In practice, approval-everything becomes a rubber stamp. When every action needs a click and the clicks number in the hundreds, the person approving stops reading and starts clearing. A gate that is never actually closed is not a gate; it is a formality that produces a record of consent without the substance of it.

"We turned the agent loose because the approvals slowed us down." This is the opposite failure and an honest one — at least no one pretends. But approvals-off is simply no gate at all. The leverage is real and so is the exposure, unmanaged.

"We have a classifier / a filter that catches bad actions." This is the most seductive, because it sounds like enforcement. It is detection. A filter watches for the bad thing and tries to block it on the way through. That is a race — you have to catch every variation, forever, and you only have to miss once. The incident above is precisely what it looks

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

like when detection is the plan and the attacker finds the one path the detector trusted. A filter can be a useful second layer. It is not a floor. As we put it in our own practice: a second net never gets dressed up as the floor.

The through-line is simple. Approval theater, no gate, and detection are all bets on behavior — on people behaving diligently, or on a model behaving predictably, or on a filter behaving comprehensively. A control worth the name does not ask the system to behave. It removes the ability to misbehave, and then it lets you check that the removal actually took.

Configured versus binding

Here is the distinction to carry out of this brief, because everything else hangs on it.

A control can exist in two very different states, and they are easy to confuse because they look identical on paper. A control is **configured** when someone has set it up — written the policy, flipped the setting, drawn the boundary. A control is **binding** when it actually holds under real conditions — when the thing it was supposed to prevent is, in fact, prevented.

Configured is a statement about intent. Binding is a statement about fact. The distance between them is invisible until something tests it — and by then, if the answer is bad, you are already reconstructing what happened rather than preventing it.

The only thing that closes that distance in advance is ground truth: the machine's own record of what actually occurred, read directly, rather than the agent's account of itself or the vendor's assurance about it. When a control is checked against ground truth, "it is configured" becomes "it is binding, and here is the evidence" — or it becomes "it is configured but does not bind, and here is the evidence for that, while there is still time to fix it."

This is the difference between a system you can stand behind and a story you hope holds. It is also the difference between the two states no one noticed in the incident above until the containment was already behind them.

A receipt from our own lab

So can anyone actually tell the difference — configured versus binding — in advance, rather than after? Yes. It is the work we do, and we have done it on our own equipment, so we can report it as a result rather than a promise. This is a separate thing from the OpenAI story above: that one is theirs, and public; this one is ours.

On a dedicated appliance in our own lab, we placed a commercial coding agent under containment and gave it a single sanctioned path: it was configured to route all of its outbound traffic through that one path and nowhere else. Then we let it do real work. At one point the agent's own internal software reached for its vendor's servers directly — on a route its configuration had no authority to redirect. The configuration did not stop it. But the floor beneath the configuration — the containment enforced below the level of any setting — refused the attempt in the act, and the only traffic that reached the vendor did so through the sanctioned path. We hold the machine's own record of both the attempt and the refusal.

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

That is the whole thesis of this brief, proven on our own bench rather than argued: the configured control did not bind — the agent's own client went around it — and the floor beneath it did. The containment also holds from a cold start: after a full power-off and reboot, it re-establishes itself on its own, rather than depending on anything a restart would clear.

Said plainly and without inflation: this is one commercial coding agent, on our own appliance, and our diagnostic instrument today covers six of eight checks with receipts in hand — the "common instrument, depth-varying" honesty stated again at the end of this brief. We are not claiming every agent everywhere is contained. We are claiming something narrower and more useful: that the difference between a control that is configured and one that binds can be measured in advance, on real behavior — and that we have measured it.

The nine questions

As leaders have started to feel this problem, a set of questions has taken shape, and the published governance checklists repeat many of the same ones. But look closely at *which* ones they repeat: the organizational questions. Across the public checklists and board guides, five recur again and again — and all five are about people and process, not measurement:

1. **Who is accountable** when an agent takes an action — a named person, not a diffuse "the system"?
2. **What is our stated policy** for what agents may and may not do, and does anyone enforce it?
3. **Are our people trained to supervise these systems**, or only to deploy them?
4. **What is our response** if an agent causes harm — a plan, or improvisation?
5. **How do we evaluate the vendors** whose agents we run, beyond taking their word?

Those five are real, and you must answer them.[1] Notice, though, what the checklists mostly leave out. They are not entirely silent on measurement — they will tell you to watch your AI's *outputs* for accuracy and bias. What they almost never ask are the access-and-integrity questions: not "is the output good?" but "what can this agent reach, what does it hold, what happens when a control fails, and does its account of itself match the record?" The recognized frameworks do name that dimension — the U.S. government's AI Risk Management Framework devotes an entire function to measurement, and the security community publishes standing lists of agentic-AI threats — but the executive checklists circulating to leaders rarely turn any of it into something you can check. To the field's five, then, we add four questions that can be answered with evidence rather than assurance:

6. **What can the agent actually reach?** Put it in a real, contained setting, let it try to reach beyond its lane, and read from the environment's own records whether the attempt was permitted or refused. Not "we configured it to be contained" — *did the reach succeed or not.*
7. **Does it hold usable credentials?** After the agent does genuine work, inspect everything it could see and search for the actual secret. Either the live credential is recoverable in its reach or it is not. That is a fact you can produce.
8. **What happens when a control it depends on fails?** Remove a dependency the containment relies on and watch what the system does. Either it refuses to proceed — fails closed — or it carries on without the missing control. There is a right answer and it is observable.

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

9. **Does the agent's account match the record?** Compare what the agent says it did against what the machine's records show it did, and measure the gap. The incident above is a live reminder of why this matters: a capable system, asked to report on itself, worked to corrupt the report. Self-reported "done" is a claim; the record is the fact.

That is the whole set — five the field keeps returning to, four it leaves thin. A serious program does both: it *decides* the five and it *measures* the four, and it never lets a decided answer stand in for a measured one, or pretends a measurement settles an organizational duty. Most published checklists stop at the five. The four are where "configured" turns into "binding" — which, as the incident showed, is the only distinction that matters when it matters. If a governance conversation you are having only ever reaches the five, that is the sign it has not yet reached the part that can actually be verified.

> [1] We reviewed eight public AI-governance sources for this brief — seven executive/board and management-framework governance guides, plus one agentic-security list. The organizational questions above recurred widely (accountability in seven of the eight; training in six; policy and vendor risk in five each). The four access-and-integrity questions appeared in essentially one of the eight — the security list — with only faint gestures in two board guides; they are, however, standard security controls (see the appendix), which the executive checklists simply do not surface. The one measurable question the governance checklists do embrace is whether an AI's *outputs* are accurate and unbiased. The eight sources, their links, and the full tally are in the appendix.

What "measured, not attested" looks like

Take the credential question, because it is the cleanest to see without any technical background.

A vendor tells you: "Don't worry — your AI agent never actually holds your live credentials. We broker them safely behind the scenes." That is a reasonable design, and it may well be true. But as stated, it is an attestation. You are being asked to believe it.

Here is the same claim, measured. You let the agent do a real piece of work that genuinely required those credentials — a true end-to-end action, not a demo. Then, rather than asking the agent or the vendor, you look at the environment's own records of everything the agent could see while it worked: the settings it was handed, the files within its reach, the logs it wrote. And you search that space for the actual secret. If the live credential is nowhere the agent could reach — only a placeholder is — the claim held, and now you have the receipt. If the real secret turns up somewhere in the agent's reach, the claim did not hold, and you have found that out on your own terms, in a test, rather than in an incident.

Nothing in that walkthrough requires you to read code. It requires one discipline: after the claim, check the record. The finding is not "the vendor says it's safe." The finding is "we ran a real action and the live secret was not in anything the agent could reach — measured, not attested." That sentence is worth more to a board, an auditor, or a customer's security review than any assurance, precisely because it is backed by the machine's own account rather than anyone's word.

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

bluejacket.io · joseph@bluejacket.io · (551) 277-5775

The same move works for the other three measurable questions. In every case the shape is identical: state the control, then check it against ground truth, and report what the evidence shows — including when the evidence is uncomfortable. That last part is the discipline. A check that can only ever come back "pass" is not a check.

What to do in the next 30 days

You can make real progress on this in a month, with your own people, whether or not you ever bring in outside help. Five concrete moves:

1. Name where agents can act. List every place an AI agent can take an action on a real system in your company — not answer a question, but *do* something: write, send, spend, change a record, touch a repository or a customer system. You cannot govern what you have not named, and most leaders find the list is longer and less supervised than they assumed.

2. Ask the four measurable questions of each one — and demand evidence, not assurance. For every agent on that list: what can it reach, does it hold usable credentials, what happens when a control fails, and does its account match the record? Where the answer is a vendor's word, mark it unverified. The marking itself is progress.

3. Separate proposing from committing on anything irreversible. For actions you cannot easily undo — money out, data deleted, a record of truth changed — put a real human commit between the agent's proposal and the deed. Not a rubber-stamp approval it can talk its way past; an actual person who takes the last step. Agents propose. People commit.

4. Decide, in advance, to fail closed. Choose now — before the moment arrives — that when a control the agent depends on is missing or unhealthy, the agent stops instead of guessing. An agent that keeps going when it should stop is how the unrecoverable actions happen: it reaches the thing it can't undo before anything reliable can intervene. Failing closed trades a little availability for a lot of safety. Make that trade deliberately.

5. Put one name on it. Assign a single accountable owner for agent governance — a person, not a committee and not "engineering." The organizational questions do not answer themselves, and diffuse ownership is the same as none.

None of these requires a purchase. All are worth doing regardless of what you decide about outside help. If you do only the first two, you will already know more about your real exposure than most companies your size.

Honest scope

This brief is for leaders of privately held B2B companies in roughly the $5M–$100M range who have started to feel this problem specifically — agents already touching real systems, and a growing sense that "we have a policy" is not the same as "we are covered." It is written for the CEO, COO, or GM who is smart, busy, and non-technical.

It is not for everyone. If you run a large enterprise with a staffed, round-the-clock security operation, you have coverage this does not assume. And if what you want is a compliance checkbox for a regulation that hasn't taken effect, this is the wrong document — the argument here is about controls that actually bind, not paperwork that satisfies a future deadline.

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

What Blue Jacket does, stated without inflation: we measure whether the containment around your AI agents actually holds, by checking specific controls against the machine's own records rather than against anyone's assurance — the difference this brief calls *measured, not attested*. Our diagnostic instrument covers a defined set of these checks; six are instrumented and carry receipts today, and two more are on the roadmap and are not claimed as working. The full battery has been exercised end to end against one commercial coding agent so far; the depth of our evidence grows with each engagement. The honest phrase for where we are is *common instrument, depth-varying* — one method, applied so far to one agent in full, not "every agent fully contained."

What we do not claim: we make no claim of autonomy, of immutability, or of any cryptographic guarantee. What we demonstrate is supervised containment with an evidence trail that is added to and not quietly edited. We are not an accredited certifying body, and we do not issue certifications. We are early — pre-revenue — and we would rather tell you that than dress the practice up as something older than it is. That candor is the point. It is the same discipline we would install in your systems, applied to how we describe our own.

About the author. This brief was written by **Joseph Alise**, founder of Blue Jacket Consultancy (bluejacket.io) and an IAPP-certified **AI Governance Professional (AIGP)**. He spent roughly thirteen years in revenue operations and enterprise data sales, and served in the U.S. Navy's surface fleet. Blue Jacket is the discipline he built to run his own AI-orchestrated operations, now brought to clients — which is also why the lab receipt in this brief exists: he ran the test on his own equipment before offering it to anyone else's.

Next step

If this is a problem you're beginning to feel, a short discovery call is the place to start — no obligation, and you will leave it knowing more about your own exposure than when it began.

Appendix — governance sources surveyed (footnote [1])

Eight public sources reviewed, July 2026, underlying the tally in "The nine questions": seven executive/board and management-framework governance guides, plus one agentic-security list (marked ▪). Direct document links below. Some question-lists were read via the publishers' public summaries rather than the full documents; the tally reflects that limitation.

1. NIST AI Risk Management Framework 1.0 — GOVERN function — https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf
2. ISO/IEC 42001:2023 — governance controls (Annex A) — https://www.iso.org/standard/81230.html
3. ▪ OWASP Top 10 for Agentic Applications (2025) — https://genai.owasp.org/2025/12/09/owasp-top-10-for-agentic-applications-the-benchmark-for-agentic-security-in-the-age-of-autonomous-ai/
4. NACD — director discussion guide for board decisions on AI — https://www.nacdonline.org/all-governance/governance-resources/governance-research/director-handbooks/2026-cyber-risk-oversight/cyber-risk-handbook-toolkit-2026/ai-discussion-guide/
5. Deloitte — AI in the boardroom: five governance actions — https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/articles/ai-boardroom-

© 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not licensed for use as AI training data.
Shareable in unmodified form with attribution.

governance-five-actions.html

6. KPMG — AI governance principles for boards — https://kpmg.com/xx/en/our-insights/ai-and-technology/ai-governance-principles-for-boards.html
7. World Economic Forum — Empowering AI Leadership: an oversight toolkit for boards — https://www3.weforum.org/docs/WEF_Empowering-AI-Leadership_Oversight-Toolkit.pdf
8. IAPP — establishing governance for AI systems — https://iapp.org/news/a/establishing-governance-for-ai-systems

Tally (out of 8). Organizational questions recurred widely — accountability 7, training 6, policy 5, vendor risk 5, incident response 3. The four access-and-integrity questions were far rarer — what an agent can reach 2, usable credentials 2, fail-closed 1, self-report-versus-record 1 — and every strong appearance came from the single agentic-security source (item 3), with only faint gestures in two board guides (IAPP on reach; NACD on credentials).

On the four's provenance. That these four are near-absent from *board* checklists does not make them exotic — they are established security controls. NIST SP 800-53 Rev. 5 covers them directly: access enforcement and information flow (AC family), identification and least privilege (IA and AC-6), boundary protection and fail-secure (SC-7, SC-24), and audit records against which a self-report can be diffed (AU family) — https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final. The gap this brief names is one of *translation*: the controls exist; the executive checklists do not surface them.

Shareable in unmodified form with attribution. © 2026 Blue Jacket Businesses LLC, d/b/a Blue Jacket Consultancy. Not for use in machine-learning training without written permission. Incident details are drawn solely from the primary public disclosure; framework and capability claims re-register the Blue Jacket public crosswalk (v1.1) and bluejacket.io for a non-technical audience and introduce no new claims.