WHITEPAPER · THE AGENTIC ERA

Govern Before It Acts

The Infrastructure Operating Model for the Agentic Era

Operations tooling has learned to see, to correlate, and increasingly to act on its own. None of it governs what an action is **allowed** to be. This paper makes the case for the missing layer.

AUTHORIOM JUNE 2026 12 MIN READ

ABSTRACT

Event Intelligence Solutions have matured from hype into mainstream deployment, and the frontier is no longer detection and recommendation but action — AI that resolves on its own. Yet every one of these systems reasons against a model of what *is*: discovered topology, a configuration database, telemetry. None reasons against a model of what is *allowed*. That gap is structural, not a maturity problem — no volume of observed state yields authoritative intent. This paper defines the Infrastructure Operating Model as the authoritative model of intent and allowed states that validates every change — human, automated, or AI — before it executes, and argues that to adopt its premise is to accept how trustworthy AIOps must be implemented.

01 — THE AGENTIC INFLECTION

The day operations tooling started to act

For most of a decade, the artificial intelligence in IT operations did two things: it detected anomalies, and it recommended. A human remained in the loop, reading the recommendation and deciding whether to act on it.

That arrangement is ending. The leading platforms now ship agents that investigate and resolve incidents on their own. ServiceNow places generative agents inside IT operations that triage and remediate; BigPanda frames its direction as agentic IT operations; Vitria positions its platform around autonomous resolution across hybrid and network estates. The recommendation has become an action, and the human who used to stand between the two is being designed out of the path.

This changes the governing question. For a recommendation, the question was diagnostic: did the system predict the impact correctly? For an action taken autonomously, the question is one of permission: **is this change allowed to happen at all?** It must also be answered at the speed the action is taken: an autonomous operator does not wait for a change-advisory board, and a control that needs hours or days to return a verdict is no control at all. A probabilistic estimate of impact, however good, does not answer the permission question. Permission is not a prediction. It is a property of the change measured against intent — and intent is precisely what these systems do not hold.

---

02 — WHAT THE INCUMBENT ACTUALLY IS

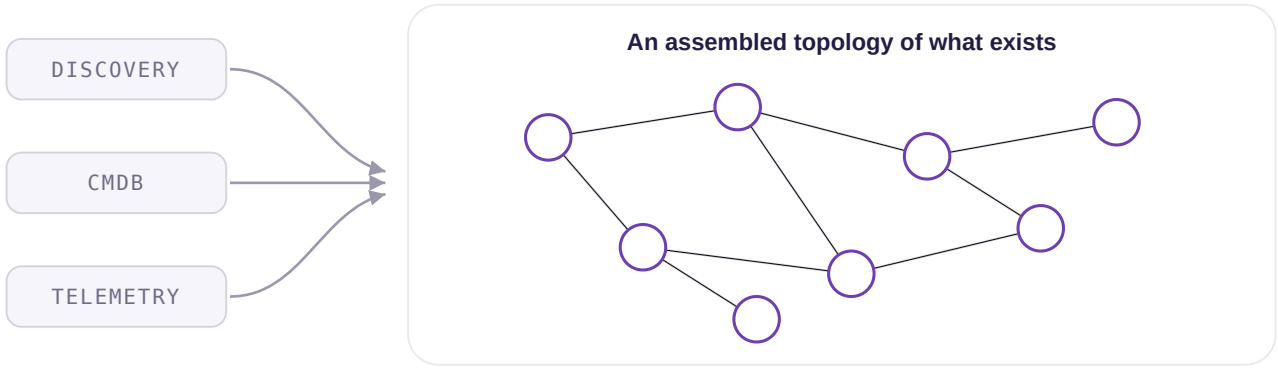
Event Intelligence, described on its own terms

It is worth being exact about the category, because the argument that follows depends on characterizing it fairly rather than caricaturing it. In 2025, Gartner retired the label "AIOps Platforms" and renamed the market "Event Intelligence Solutions," after years in which the term had been applied so broadly, and promised so much, that infrastructure and operations leaders had grown skeptical of it. The rename was a narrowing, not a dismissal: it refocused the category on a real and valuable domain.

That domain is the application of machine learning and analytics to cross-domain events drawn from monitoring and observability tools, in order to reduce noise,

correlate signals into incidents, and accelerate — and increasingly automate — the response. At this, the category is genuinely good. It collapses thousands of redundant alerts into a handful of actionable incidents. It surfaces problems earlier than static thresholds would. It is, for many operations teams, the difference between drowning and triaging.

Two facts about it matter here, and both come from the category's own definition rather than from any critique of it. First, Event Intelligence is designed to live inside a portfolio: Gartner is explicit that it depends on adjacent tooling — configuration databases, IT service management, automation — and that integration across that portfolio is essential to making it work. It does not own change control; it expects the surrounding stack to provide it. Second, the picture it holds of the estate is a *discovered* one: an assembled topology of services and dependencies, populated by automated discovery and a configuration database. That picture is a faithful record of what exists and how it connects.



WHAT EVENT INTELLIGENCE REASONS AGAINST — A MODEL OF WHAT *IS*: DESCRIPTIVE, DISCOVERED, AND DRIFTING.

None of this is a weakness. It is the category working exactly as designed. The question is what that design leaves unaddressed.

---

03 — THE AUTHORITY GAP

A model of what is, not what is allowed

Every model described in the previous section answers the same kind of question: what exists, what connects to what, what is happening right now. Not one of them answers a different kind of question entirely — what is *permitted* to change, by whom, and into what allowed state.

This is the authority gap, and it is easiest to see as a set of distinctions the existing tooling cannot make.

IT RECORDS

What is

→ What is allowed

IT LOOKS

Backward & present

→ Forward, at a proposed change

IT PRODUCES

A description

→ A decision

IT IS

A snapshot that drifts

→ A reference that governs

The market is beginning to sense that the model is the constraint. The clearest admission comes from inside the category itself: Vitria's chief technology officer has argued that conventional data structures give an AI statistical patterns where it needs genuine understanding, and that **you cannot automate what you cannot explain**. The diagnosis is correct. But the remedy on offer — a richer, more semantic model of what the estate *is* — does not close the gap. A better description of what exists is still a description of what exists. Understanding the estate more deeply is not the same as holding authority over change to it.

---

04 — WHY NO TOOL CLOSES IT

The gap is structural, not a question of maturity

It is tempting to assume the authority gap is temporary — that with enough discovery, a rich enough configuration database, and enough telemetry, the picture eventually becomes complete enough to govern from. It does not, and the reason is not a matter of fidelity.

Discovery enumerates what exists. A configuration database records how things are configured, whether discovered or declared. Telemetry streams what is happening from moment to moment. Each is a faithful empirical account — a statement of fact about the world as it is. Intent is not a fact about the world as it is. It is a statement of what *ought* to be permitted, which no quantity of observation contains. You cannot derive an "ought" from an "is," and stacking more "is," at higher resolution, never crosses into "ought." The missing layer is therefore not a better instrument in the existing stack. It is a different *kind* of artifact — a normative one, where everything in the Observe and Execute layers is empirical.

A concrete case makes the distinction tangible. Discovery can tell you that a security group permits inbound access from any address on a remote-administration port. That is a fact about the world as it is. It cannot tell you whether that configuration is permitted for this asset, in this environment, under this owner's intent — whether it is a sanctioned exception or a violation waiting to be exploited. The same observed fact is admissible in one place and forbidden in another, and only a model of intent can tell the two apart. Observation sees the rule; only authority knows whether it is allowed.

TWO REASONABLE OBJECTIONS

"Is this not just a configuration database?"

A configuration database records what is configured. It is description, it is populated after the fact, and it drifts from reality between discoveries. It does not hold a statement of what is permissible, and it does not validate a proposed change against intent before that change executes. The operating model is consulted before the change, not reconciled after it.

"Does policy-as-code not already do this?"

Policy-as-code checks a change against rules at a single gate — usually the infrastructure-as-code pipeline. It is a valuable input to governance, not a substitute for it. It has no authoritative, cross-estate model of allowed states to reason against, and it does not govern the changes that arrive outside its gate: the console, the vendor agent, the autonomous operator. Governance that only covers one path is not governance of the estate.

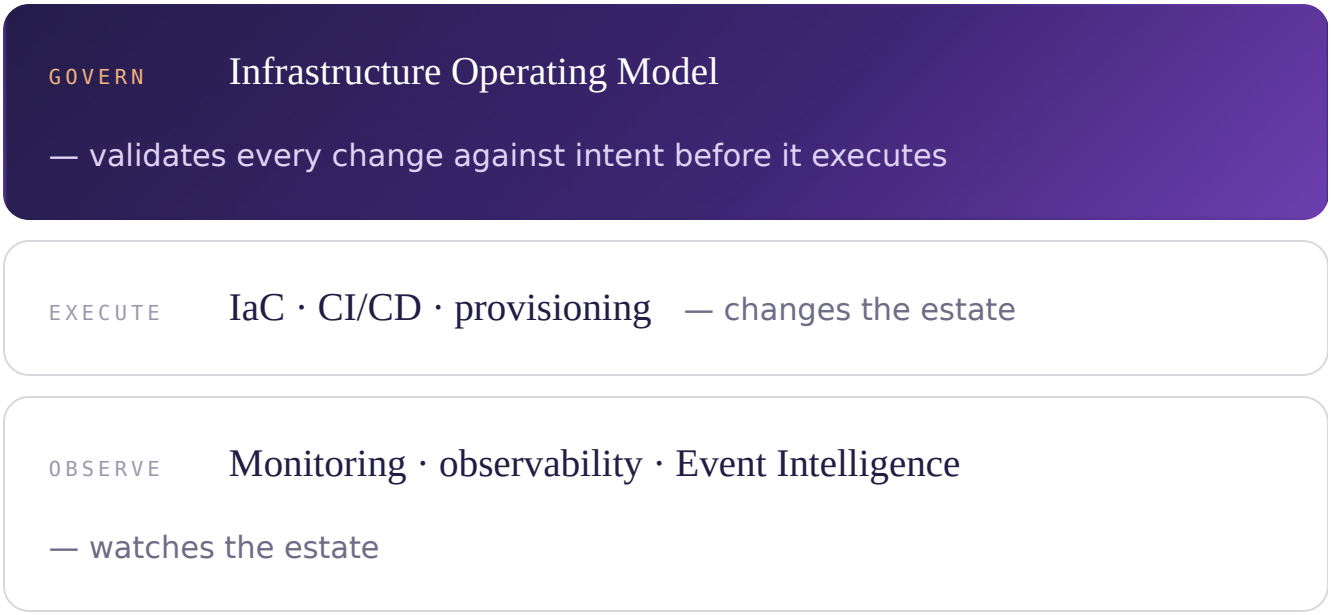
---

05 — THE MISSING LAYER, NAMED

The Infrastructure Operating Model

An Infrastructure Operating Model is the authoritative model of intent and allowed states for an enterprise's infrastructure. It is the layer that decides, rather than the layer that watches or the layer that acts.

It is best understood by where it sits. Most enterprises have already matured two layers beneath it. The **Execute** layer changes the estate: infrastructure-as-code, continuous delivery, provisioning. The **Observe** layer watches the estate: monitoring, observability, and Event Intelligence. Both are essential, and neither decides whether a given change should be permitted. Execute will faithfully apply whatever it is told to apply. Observe will faithfully report whatever happens as a result. The decision in between has had no home.



The Govern layer is the one most enterprises have never had.

The Govern layer is where the operating model lives, and it works on a single discipline: no actor changes infrastructure directly. A human, an automation, or an AI agent *proposes* a change. The model *validates* that proposal against what exists, what is intended, and what is permissible. Only then does a deterministic engine *execute* what has been approved. The actor never touches the infrastructure itself; it touches the model, and the model holds the authority. Governance lives in that model rather than in any one agent or pipeline, which is why it does not have to be re-established every time a new tool, a new vendor, or a new generation of AI enters the environment.

---

06 — THE CONSEQUENCE

How AIOps runs on it

Place the two side by side and the relationship is not rivalry but layering. Event Intelligence observes the estate and, increasingly, triggers action within it. The Infrastructure Operating Model governs what that action is allowed to be. They occupy different layers and do different work; neither replaces the other.

The analogy is the processor and the laptop. A processor is the component a laptop runs on without being the laptop. In the same way, the operating model is the layer trustworthy AIOps runs on without being AIOps. AuthorIOM does not run your operations; it governs what your operations are permitted to do.

And here is the consequence that is difficult to step around. If you accept that an autonomous system must validate its actions against a model of what is allowed before it executes — and once that system can act on its own, it is hard to accept anything less — then you have accepted that something must hold that model authoritatively. That something is an Infrastructure Operating Model. It cannot be the event-intelligence platform, because that platform reasons against a model of what is. It cannot be assembled from discovery, a configuration database, and telemetry, because those are empirical and intent is normative.

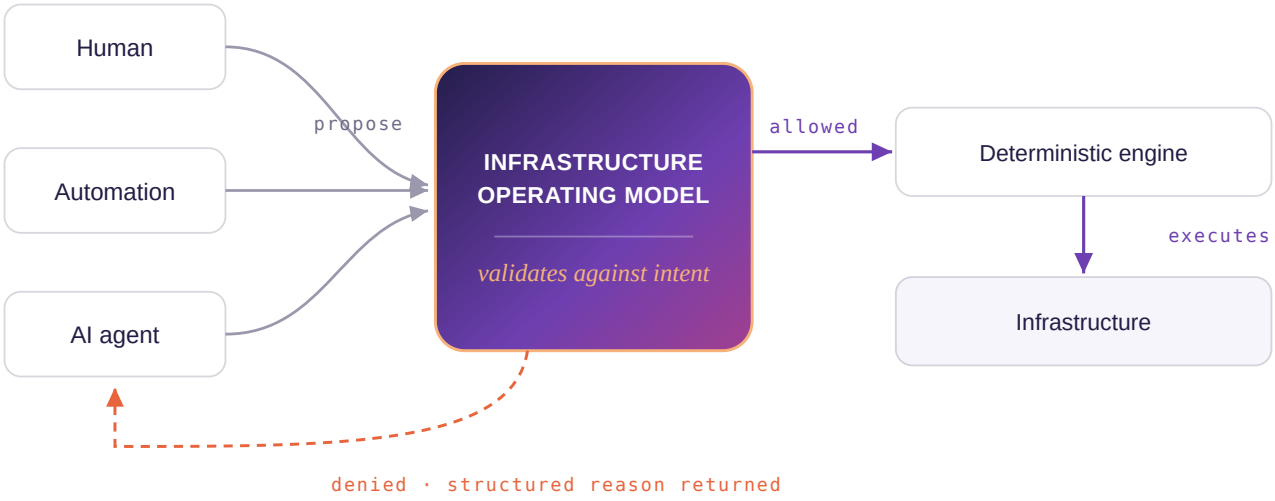
> *Adopt the premise of an operating model, and you have already accepted how trustworthy AIOps gets implemented — not as a feature inside the platform, but as a layer beneath it.*

---

07 — IN PRACTICE

What the layer looks like when it runs

The operating model presents itself to the rest of the stack as a Decision API: a synchronous, pre-execution check. A proposed change that conforms to intent proceeds at pipeline speed. A change that would violate intent is stopped before it executes, with a structured reason returned to whatever proposed it — the pipeline, the console, or the agent. Governance is therefore not a review meeting or a nightly reconciliation; it is a checkpoint in the path of every change, evaluated in line. A change that is stopped is not merely rejected; the structured reason travels back to its author, who can correct the proposal and resubmit, so the model shapes the change rather than only blocking it. Over time, that turns governance from a barrier into a contract the whole estate is built against.



EVERY CHANGE IS VALIDATED BEFORE IT EXECUTES — AND NO ACTOR TOUCHES INFRASTRUCTURE DIRECTLY.

The model is built from the systems that already hold pieces of the truth — cloud and on-premises infrastructure, infrastructure-as-code and continuous delivery, identity, IT service management and the configuration database, observability, and Event Intelligence itself. It connects through their interfaces rather than through agents installed on hosts, and it begins read-only, so that nothing is gated until governance is deliberately turned on. In that posture, Event Intelligence becomes one of the model's richest sources of signal: the events and correlations it produces inform the model, and the model decides what may be done in response.

The value is best stated as mechanism rather than as a promised number, because the mechanism is what is durable: every change, from every actor, validated against one authoritative model before it executes. What an organization saves, consolidates, or prevents follows from that, and is best modeled against its own environment rather than asserted in the abstract.

---

08 — THE PATH FORWARD

Make intent authoritative

Most enterprises can now see their infrastructure in extraordinary detail and still cannot say, with authority, what is allowed to change

it.

That is the authority gap, and it stops being abstract the moment an autonomous actor is handed a console. The response is not another instrument in the Observe layer or the Execute layer, however capable. It is to make intent authoritative — to give every actor, human or machine, a single model to validate against before anything runs. The era in which infrastructure can act on its own is the era in which that model stops being optional.

Sources & notes

1. Gartner, *Market Guide for Event Intelligence Solutions*, Matt Crossley and Gregg Siegfried, 10 March 2025 — for the renaming of the AIOps Platforms market to Event Intelligence Solutions, the category's defined domain, and its stated dependence on adjacent configuration-database, service-management, and automation tooling.
2. Vendor materials on agentic operations: ServiceNow (generative agents within IT operations management), BigPanda (agentic IT operations), and Vitria Technology (autonomous resolution) — for the shift from recommendation to autonomous action.

3. Vitria Technology, remarks by the chief technology officer, Appledore Research Podcast, episode 61, May 2026 — for the observation that conventional data structures provide statistical patterns rather than understanding, and that automation depends on explainability.
4. The discovered-topology and configuration-database characterization of Event Intelligence reflects publicly documented platform capabilities across the category (for example, ServiceNow IT Operations Management and BMC Helix).