

Trust by Design: A Case Study of Human-Approved AI Automation in Network Operations

Nikhil

Synetal Solutions Private Limited · nikhil@synetalsolutions.com

Abstract—Agentic artificial intelligence is reshaping network operations (NetOps), moving the field from scripted automation toward systems that can diagnose, converse about, and in some cases remediate infrastructure issues with limited human intervention. Yet capability alone does not explain adoption: network engineers remain cautious about granting autonomous systems write access to production infrastructure, and this caution, rather than any technical shortfall, is often the binding constraint on how far automation is actually used. Existing models of technology adoption and trust in automation were developed mainly for consumer and enterprise software, and they say little about how specific architectural choices build or erode trust in a safety-critical operations context. This paper addresses that gap through a qualitative single-case study of OrcaMonitor, an AI Ops platform for Cisco-first, multi-vendor network environments that combines a conversational query interface, autonomous remediation agents, and a mandatory human-approval workflow. Using thematic analysis, the platform's design features are mapped onto established trust-in-automation constructs, including ability, benevolence, and integrity, and the complementary dimensions of predictability, transparency, and control. The case analysis is triangulated against independent 2026 industry survey data ($n = 1,000$ IT and network operations leaders), which shows that 99% of organizations require at least one explicit guardrail before trusting AI to act, and that 56% require human approval specifically, corroborating the centrality of the approval-gate mechanism identified in the case. The paper contributes a practitioner-oriented framework, termed Trust-by-Design, that connects specific design decisions to trust and adoption outcomes, intended to inform both the designers of AI Ops platforms and the operations teams evaluating them.

Keywords—trust in automation, agentic AI, AI Ops, network operations, human-in-the-loop, technology adoption, case study, AIOps

1. Introduction

1.1 Background

Network operations centers have, for roughly two decades, relied on a layered automation model: rule-based monitoring systems raise alerts, scripted playbooks handle routine remediation, and engineers intervene for anything a script was not written to anticipate. This model scales poorly against two pressures that have intensified in recent years. The first is alert volume: as networks absorb more devices, more vendors, and more cloud-connected segments, the number of signals reaching a NOC queue grows faster than headcount. Independent survey data bears this out directly: the average organization now generates approximately 4,100 monitoring alerts and events per day, of which roughly half are network-related, while a typical practitioner can review and resolve only about 21 network alerts in a day — a gap that would require a team of nearly 100 specialists to clear by hand (Cisco and Omdia, 2026). The second pressure is heterogeneity: a modern enterprise network rarely runs a single vendor's stack end to end, so a script written against one platform's command syntax or API does not generalize to the next.

Agentic AI has been proposed as a response to both pressures. Rather than encoding a fixed set of remediation steps, an agentic system can interpret a novel alert, query live device state across multiple vendor APIs, propose a diagnosis in natural language, and in more advanced implementations draft or execute a fix (Bajpai, 2024). This is a qualitatively different mode of automation from the scripted era: it substitutes judgment-like reasoning for pre-written logic, which raises the stakes of the question addressed in this paper, namely whether and how operators come to trust it. The scale of this shift is already evident in practice: among organizations using AI for network operations, 51% report running agentic AI that acts rather than merely advises in production today, 84% expect to reach a fully AI-led operating model within twelve months, and 85% expect AI agents to autonomously manage most operational tasks within five years (Cisco and Omdia, 2026).

1.2 Problem Statement

Industry survey data and vendor-reported deployment experience consistently show a gap between the technical maturity of AI-driven network automation and its operational adoption. Tools capable of autonomous diagnosis and remediation are frequently deployed in a read-only or advisory capacity, with engineers unwilling to extend write access to production systems even where the underlying model performs reliably in testing. This gap is not well explained by capability alone. It is more plausibly explained by a trust deficit: operators are being asked to delegate actions with real operational consequences to a system whose reasoning is not fully transparent to them, in an environment where an error can cause an outage rather than a minor inconvenience.

1.3 Research Gap

Two bodies of literature bear on this problem without fully addressing it. The first, trust-in-automation research, offers general constructs for how humans come to trust automated systems, but was developed substantially from human factors work in aviation and process control, and has been applied to IT operations only unevenly. The second, technology adoption research such as the Technology Acceptance Model and the Unified Theory of Acceptance and Use of Technology, explains adoption primarily through perceived usefulness and ease of use, constructs that travel well to productivity software but map awkwardly onto a context where the central barrier is not usability but the willingness to cede operational control. Neither literature, on its own, offers a framework for evaluating how a specific NetOps tool's design choices, such as whether a conversational interface is read-only, whether remediation requires staged approval, or whether actions are logged for audit, function as trust-building mechanisms. This paper treats that intersection as the research gap.

1.4 Objectives

This study has three objectives: first, to identify, through a single-case examination of OrcaMonitor, the specific design mechanisms used to bridge the trust gap between AI capability and operator willingness to adopt; second, to map these mechanisms onto established trust-in-automation constructs in order to assess whether they correspond to recognized antecedents of trust, and to triangulate that mapping against independent industry survey evidence on what enterprises actually require before trusting agentic systems; and third, to synthesize the findings into a framework that platform designers and adopting organizations can use to evaluate or design trust-relevant features in agentic network automation tools.

2. Literature Review

2.1 Trust in Automation

The foundational framework for this study comes from human factors research on trust in automated systems, most notably the work of Lee and See (2004), who characterize trust as resting on three bases: the automation's performance (does it work reliably), its process (is its logic understandable), and its purpose (does it serve the operator's goals). A separate but complementary framework, drawn from organizational trust research (Mayer, Davis, and Schoorman, 1995), decomposes trust into judgments of a trustee's ability, benevolence, and integrity. Applied to automation rather than a person, ability maps to demonstrated technical competence, benevolence to evidence that the system acts in the operator's interest rather than against it, and integrity to consistency between what the system claims to do and what it actually does. A recurring theme across this literature is the problem of calibration (Parasuraman and Riley, 1997): both over-trust, where operators extend automation more authority than its reliability warrants, and under-trust, where capable automation goes underused, are treated as failures of calibration rather than simple deficits of trust.

2.2 Technology Adoption Models

The Technology Acceptance Model (Davis, 1989) and its successor, the Unified Theory of Acceptance and Use of Technology (Venkatesh et al., 2003), explain adoption primarily through perceived usefulness, perceived ease of use, and social or facilitating conditions. These models have been validated extensively across enterprise software contexts, but they were not built around scenarios where the adoption decision carries operational risk to infrastructure the adopter is personally accountable for. In such contexts, perceived usefulness may be high while adoption still lags, because the limiting factor is not whether the tool is useful but whether handing it operational authority is acceptable. This paper treats TAM and UTAUT as necessary but insufficient for explaining adoption in safety-relevant automation, and argues that trust-in-automation constructs do more explanatory work in this setting.

2.3 Human-in-the-Loop Design in AI Systems

A separate strand of literature, emerging from machine learning operations and responsible AI design, examines human-in-the-loop (HITL) patterns: approval gates before an automated action executes, confidence thresholds that route uncertain cases to a human reviewer, and staged autonomy models that expand a system's authority only after a track record is established. This literature is largely prescriptive, offering design patterns, but it is rarely connected empirically to trust outcomes in a specific operational domain, which is one of the connections this paper attempts to make.

2.4 AIOps and Agentic AI in Network Management

A growing applied literature documents the shift from monitoring-and-alerting tools toward AIOps platforms that apply machine learning to anomaly detection and root-cause correlation. Bajpai (2024) describes a representative architecture in which a data ingestion layer, anomaly detection models, and root-cause correlation feed an automated remediation stage, reporting that such pipelines meaningfully reduce mean time to resolution in telecommunications network operations while identifying data quality, model interpretability, and scalability as persistent implementation challenges. This literature establishes the technical feasibility of automated diagnosis and remediation, but, consistent with the gap identified in

Section 1.3, it emphasizes engineering challenges over the operator-side trust dynamics that determine whether these capabilities are actually used in production.

More recent industry research extends this picture to agentic AI specifically. A 2026 survey of 1,000 IT and network operations leaders found that 92% of organizations report that performance issues commonly span multiple domains, that teams rely on an average of nearly ten separate tools for end-to-end visibility, and that the mean time to resolve a network incident is 88 hours against a 12.5-hour median, a sevenfold gap attributed to a small number of outlier incidents that disproportionately consume resolution time (Cisco and Omdia, 2026). This literature tends to emphasize efficiency metrics, with comparatively less attention paid to the operator-side trust dynamics that determine adoption; this paper positions its case study, together with the triangulating survey evidence in Section 4.7, as a corrective to that imbalance.

2.5 Explainability and Transparency

Research on explainable AI argues that an automated system's perceived trustworthiness depends heavily on whether its reasoning is inspectable, not merely whether its output is correct (Ribeiro, Singh, and Guestrin, 2016). In a NetOps context, this translates into concrete design features: whether a conversational AI interface shows the data it queried to produce an answer, whether an audit trail records what an autonomous agent did and why, and whether the system distinguishes clearly between telling an operator something and doing something on the operator's behalf. These features are treated in this paper as operationalizations of transparency, one of the dimensions examined in the case analysis.

2.6 Parallels from Safety-Critical Automation

Finally, the paper draws on adoption patterns from other safety-critical domains, including flight deck automation and clinical decision support, where mandatory human oversight has remained a persistent design requirement even as underlying system capability has advanced. A consistent finding across these domains is that trust grows not from capability improvements alone but from the presence of a reliable override or approval mechanism that preserves the human's final authority. This finding motivates the attention this paper gives to OrcaMonitor's approval workflow as a central trust mechanism rather than a secondary feature, and is echoed directly in current industry data: 99% of surveyed organizations report that they will not trust AI to act without at least one explicit guardrail, most commonly human approval for actions (Cisco and Omdia, 2026).

Taken together, these themes are rarely integrated into a single framework specific to NetOps. This paper's contribution is to draw that integration through a case study that treats trust-in-automation constructs, adoption theory's limitations, HITL design patterns, and transparency research as jointly necessary to explain adoption behavior around one representative platform, triangulated against independent survey evidence.

3. Methodology

3.1 Research Design

This study uses a single-case, exploratory qualitative design, triangulated with secondary quantitative survey data. A case study design is appropriate here because the research question concerns how design mechanisms operate within a bounded, real system, a question better suited to in-depth examination of one richly documented case than to a large-N survey at this early stage of inquiry into agentic AI trust in NetOps. The secondary survey data (Cisco and Omdia, 2026) is used not to generalize the case statistically but to assess whether the trust mechanisms identified within it are consistent with what a broader, independent sample of practitioners reports requiring.

3.2 Case Selection Rationale

OrcaMonitor was selected as what qualitative case study methodology terms an information-rich case: a single platform in which the trust-relevant design choices under investigation, specifically a conversational query interface, autonomous remediation agents, and a mandatory approval workflow, coexist and are directly observable, rather than having to be inferred or reconstructed from fragmentary evidence across multiple systems. The platform's stated design philosophy, summarized in its own materials as pairing autonomous diagnosis with mandatory human approval, makes the trust mechanism under study explicit rather than incidental, which is the primary justification for treating it as the case rather than selecting a platform where such mechanisms are implicit or undocumented.

3.3 Data Sources

Four categories of data inform the analysis. Primary data consists of the platform's own architecture documentation, product walkthroughs, and design-rationale materials, which describe the function and intended purpose of its conversational chat interface, its autonomous agent system, and its ticket-approval workflow. Secondary data, where accessible, consists of practitioner-facing demonstrations and feature explanations that describe how operators are expected to interact with the system in practice. Tertiary data consists of a peer-reviewed applied literature on AIOps architecture and implementation (Bajpai, 2024), used to situate the case's technical design within an established research stream. A fourth, triangulating category consists of an independent 2026 industry survey of 1,000 IT and network operations leaders (Cisco and Omdia, 2026), used to assess whether the trust mechanisms identified in the case correspond to what practitioners outside the case report requiring. Where only vendor-originated material was available for a given feature, this is noted explicitly in the results rather than presented as independently verified.

3.4 Analytical Approach

The analysis proceeds through thematic coding. Each documented feature of the platform is coded against two coding frames drawn from the literature review: the ability-benevolence-integrity framework of trust, and the predictability-transparency-control framework associated with human-in-the-loop design. A feature is coded as relevant to a construct where the documented design rationale or function plausibly addresses that construct, for example a staged approval requirement coded against control, or an audit log coded against transparency. Codes are then grouped into higher-order themes, from which the Trust-by-

Design framework presented in the discussion section is derived, and each theme is cross-checked against the independent survey data for convergence or divergence.

3.5 Limitations

This design carries three limitations that should be weighed when interpreting the findings. First, single-case designs are not intended to produce statistically generalizable claims, and the framework proposed here should be read as a theoretically grounded hypothesis for further testing rather than a validated general model; the survey triangulation in Section 4.7 partially, but not fully, offsets this limitation, since it corroborates the general shape of the trust requirements rather than validating OrcaMonitor specifically. Second, because a substantial portion of the case-specific source material originates from the platform vendor, there is a risk of favorable framing bias that independent practitioner interviews or field observation would help to offset; this study treats its reliance on vendor-originated material as a disclosed limitation rather than a resolved one. Third, the study does not include direct, platform-specific quantitative adoption or trust-survey data from actual OrcaMonitor users, meaning its claims about trust formation in this particular case are inferred from design intent rather than measured from operator response, a gap identified explicitly as a direction for future work in the conclusion.

4. Results

4.1 Case Description

OrcaMonitor is an AI Ops platform built for Cisco-first, multi-vendor network environments. Its architecture separates three functions that are often merged in earlier-generation NetOps tooling: a device inventory and monitoring layer that maintains live state across the managed estate; a conversational interface, referred to in the platform as Orca AI Chat, through which an operator can query that inventory in natural language; and an automation layer, comprising configurable AI Agents and a ticketing workflow, through which proposed remediations are executed only after staged approval. This separation is the structural basis for the trust mechanisms examined below.

Figure 1 details this architecture as documented by the vendor. Three layers are kept deliberately separate. A cloud console (orcamonitor.ai) owns the customer's tenant account, billing, and the provisioning of secure connectivity (WireGuard or IPsec) back to the customer's own network. A cloud instance, reached at a per-customer subdomain, is the day-to-day operating surface: device inventory, Intelligent Tickets, AI Agents, and Orca AI Chat. An on-prem agent, deployed inside the customer's own network, is the only component that ever reaches the managed devices directly. Connectivity between the cloud instance and the on-prem agent is outbound-only from the customer's side, meaning no inbound firewall rule is required on the customer's network, which is itself a trust-relevant design choice: it removes the need for the customer to expose a listening port to a third-party cloud service.

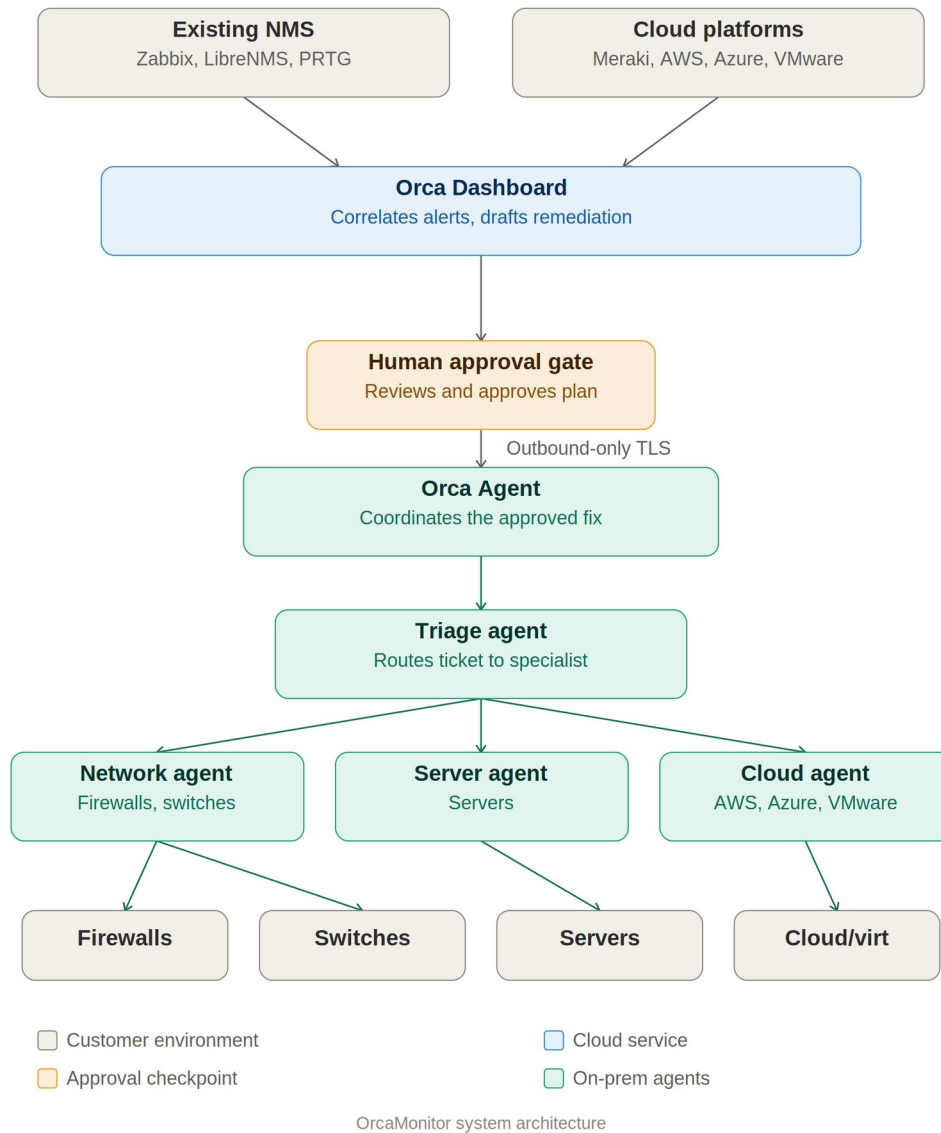


Figure 1. OrcaMonitor system architecture — cloud console, cloud instance, and on-prem agents across the approval chain. Source: platform documentation.

Alerts ingested from an existing network management system are correlated into a single Intelligent Ticket, for which an AI Agent drafts a remediation plan. That plan does not execute until it passes a staged supervisor chain, documented as an L1-to-Ln approval sequence, after which the on-prem agent executes the fix over SSH, Nornir, or a comparable mechanism, validates the result, and closes the ticket. Orca AI Chat, the platform's conversational interface, is documented as strictly read-only: it can query the same inventory and tooling to answer an operator's question, but any mutating action still has to go through the ticket-and-approval path described above, which is the specific mechanism discussed in Section 4.2.

4.2 Read-Only Conversational Access

Orca AI Chat is explicitly scoped as read-only: an operator can ask about a device's status, a routing path, or a VPN tunnel's health and receive an answer drawn from live inventory and monitoring data, but the chat interface itself cannot alter device configuration. Coded against the ability-benevolence-integrity framework, this design choice most directly addresses integrity, in that the system's stated scope (answering questions) matches its actual capability (it cannot act beyond that scope), removing the possibility that a conversational answer is mistaken for an executed action. Coded against the predictability-transparency-control framework, the same feature serves predictability, since the operator always knows in advance that a chat response is informational rather than consequential.

4.3 Staged Ticket Approval

Where OrcaMonitor's AI Agents do propose or execute remediation, this occurs through a ticketing workflow in which a proposed action is logged, routed for human approval, and only then executed. This maps most directly onto the control dimension of the trust framework: operational authority remains with the human reviewer at the point of consequence, even though diagnostic and preparatory work is automated. It also bears on benevolence, in the sense that an approval gate functions as evidence that the system is designed to work with the operator's judgment rather than to bypass it.

4.4 Audit Logging

Actions taken by AI Agents, along with the ticket history and approval record, are logged in a persistent, reviewable form. This corresponds to the transparency dimension: an operator or a reviewing manager can reconstruct what an agent observed, proposed, and did, after the fact, which is a documented mechanism for sustaining trust over time in both the aviation and clinical automation literatures discussed in Section 2.6.

4.5 Multi-Vendor Abstraction

The platform's capacity to operate across Cisco and non-Cisco device families, as well as cloud and virtualization platforms, addresses the ability dimension of trust: an operator's confidence in the system's competence is partly a function of whether it demonstrably understands the specific, heterogeneous environment it is deployed into, rather than a narrow subset of it.

4.6 Summary Mapping

Across the four mechanisms examined, each corresponds to a distinct combination of trust constructs: read-only chat access to integrity and predictability; staged approval to control and benevolence; audit logging to transparency; and multi-vendor coverage to ability. No single mechanism addresses every construct, which suggests that trust in this case is not produced by one feature but by the combination of several mechanisms, each addressing a different facet of the operator's trust judgment. This distributed pattern is taken up directly in the discussion that follows.

4.7 Corroborating Industry Evidence

To assess whether the mechanisms identified in the OrcaMonitor case reflect an idiosyncratic design choice or a broader pattern, this section triangulates the case findings against an independent 2026 survey of 1,000 IT and network operations leaders (Cisco and Omdia, 2026). The survey asked respondents

directly what level of autonomy they are comfortable granting AI systems in network operations. As shown in Figure 2, only 24% of organizations report comfort with fully autonomous action requiring no human review, while 56%, the largest single group, require a human to approve the action before it executes, and the remaining 20% permit no or only limited autonomy. This distribution corresponds closely to the control mechanism identified in Section 4.3: the modal organizational preference is not for full autonomy but for exactly the staged-approval pattern OrcaMonitor implements by design.

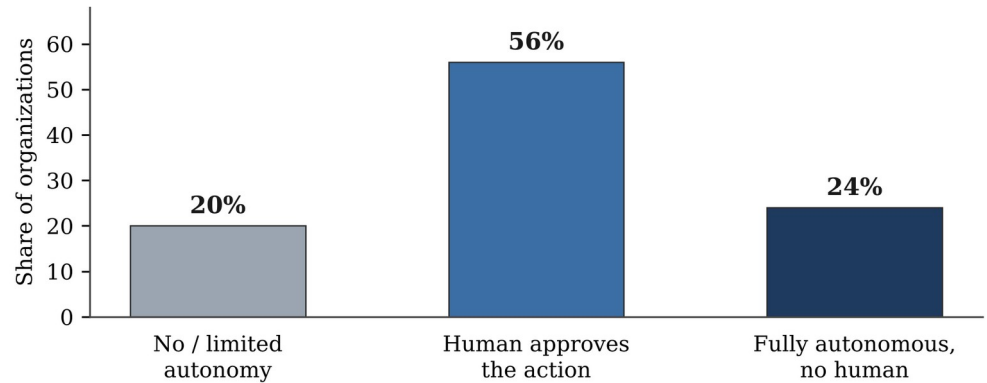


Figure 2. The autonomy spectrum: share of organizations comfortable with no/limited autonomy, human-approved action, and full autonomy for AI in network operations. Source: Cisco and Omdia (2026), n = 1,000.

The same survey also asked what conditions organizations require before trusting AI to act at all. As shown in Figure 3, 99% of organizations require at least one explicit guardrail, with the specific guardrail categories named in the underlying study, explainable AI actions, human approval for actions, policy-based operational limits, emergency override mechanisms, role-based access control, and immutable audit trails, mapping closely onto the mechanisms coded in Sections 4.2 through 4.4. In addition, 86% of respondents say a single integrated platform, rather than a collection of point tools, is the most effective path to trustworthy automation, and 69% require detailed explainability for agent-driven actions, with 36% requiring nothing less than full observability (detailed tracing, summarized rationale, and post-action audit) as the minimum acceptable standard.

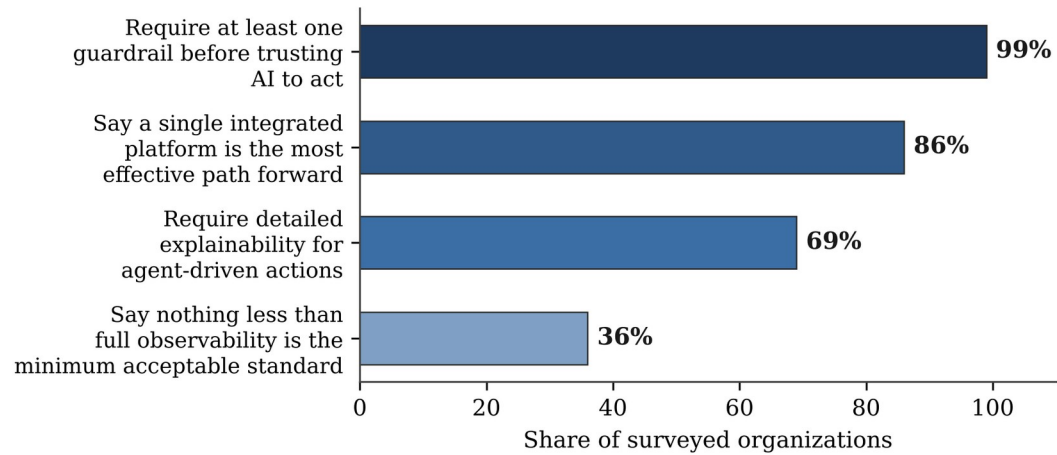


Figure 3. Trust and transparency conditions enterprises report requiring before adopting agentic AI in network operations. Source: Cisco and Omdia (2026), n = 1,000.

This convergence is notable because the survey data was not collected with reference to OrcaMonitor or any single platform; it reflects independent, aggregate practitioner preference. The fact that the guardrail categories enterprises report requiring map so closely onto the specific mechanisms observed in the case strengthens the claim, developed further in Section 5, that these mechanisms are not incidental product choices but responses to a widely shared trust requirement in the NetOps field.

4.8 Vendor-Reported Operational Outcomes

The platform's own materials report two outcome metrics associated with these mechanisms: a 94 percent reduction in alert volume reaching an operator, attributed to the correlation step that collapses related alerts into a single Intelligent Ticket, and a mean time to resolution of under five minutes, contrasted against a cited industry average above sixty minutes. These figures are reported by the vendor rather than independently measured in this study, and they are presented here as evidence of the claimed operational case for adoption rather than as validated findings; they are listed separately from the design-mechanism analysis above because they speak to perceived usefulness rather than to the trust constructs that are this paper's primary focus. Their relevance to the trust framework is indirect: a mechanism that is also associated with a large, favorable outcome metric may accelerate trust calibration by giving operators an early, legible signal that the system performs as claimed, which is consistent with the performance basis of trust in the Lee and See framework discussed in Section 2.1. It is also notable that these claimed outcomes are directionally consistent with the independent survey finding that the mean time to resolve a network incident across the broader industry is 88 hours (Cisco and Omdia, 2026), suggesting the problem the platform claims to address is, at minimum, independently documented at scale, even though the platform's own resolution-time claim is not independently verified here.

5. Discussion

5.1 The Trust-by-Design Framework

The results suggest a framework, termed Trust-by-Design, in which specific interface and workflow mechanisms are deliberately paired with the trust constructs they are intended to satisfy, rather than trust being treated as an emergent byproduct of extended use. The mapping observed in the case is summarized in Table 1.

Table 1. The Trust-by-Design framework: OrcaMonitor's design mechanisms mapped onto trust-in-automation constructs.

Design Mechanism	Trust Construct(s) Addressed	Function
Read-only conversational chat	Integrity, Predictability	Keeps stated scope and actual capability aligned; answers cannot be mistaken for actions
Staged ticket approval	Control, Benevolence	Preserves human authority at the point of operational consequence
Persistent audit logging	Transparency	Allows post-hoc reconstruction of what an agent observed, proposed, and did
Multi-vendor device coverage	Ability	Demonstrates competence across the operator's actual, heterogeneous environment

5.2 Relationship to Existing Theory and Survey Evidence

This framework both confirms and extends prior trust-in-automation theory. It confirms Lee and See's (2004) performance-process-purpose structure, in that each mechanism observed addresses either the system's reliability (performance), the inspectability of its reasoning (process), or its alignment with operator goals (purpose). It extends the ability-benevolence-integrity framework (Mayer, Davis, and Schoorman, 1995), developed originally for interpersonal and organizational trust, by showing that these constructs can be operationalized as concrete interface and workflow features in a software system rather than inferred only from repeated interpersonal interaction. This is a meaningful extension because it implies trust in agentic software need not be built solely through accumulated track record; it can be partially engineered into the system at design time.

The survey triangulation in Section 4.7 adds a form of external validity a single case cannot provide on its own. The six guardrail categories that 99% of surveyed organizations require before trusting AI to act, explainability, human approval, policy limits, override mechanisms, role-based access control, and audit trails, align closely with the four mechanisms identified in the OrcaMonitor case: read-only chat access operationalizes a form of policy-based limitation and explainability; staged approval directly operationalizes human approval for actions; and audit logging directly operationalizes immutable audit trails. The case does not appear to document an explicit emergency override mechanism or granular role-

based access control as distinct, named features, which is noted here as a gap between the surveyed guardrail expectations and the documented case rather than elided; it suggests either that these controls exist but were not emphasized in the materials reviewed, or that the case has room to extend its trust mechanisms further to fully match the guardrail set the broader field reports requiring.

The findings also clarify why TAM and UTAUT are limited explanatory tools in this context. Perceived usefulness and ease of use, the central constructs of those models, are largely orthogonal to the mechanisms identified here: a staged approval workflow, for instance, likely reduces ease of use in the narrow sense of adding a step, while simultaneously increasing adoption likelihood by preserving operator control. This suggests that in safety-relevant automation contexts, trust constructs and usability constructs can move in opposite directions, and adoption models built only around usability risk misreading a trust-building friction point as a usability defect.

5.3 Implications for Platform Designers

For vendors building agentic NetOps tools, the framework implies that trust-relevant features should be treated as first-class design requirements, specified and evaluated with the same rigor as functional requirements, rather than as secondary considerations layered onto a capable automation engine after the fact. In particular, the case suggests that separating advisory capability (what the system can tell an operator) from executive capability (what the system can do without approval) is itself a trust mechanism, independent of how capable the underlying model is. The survey evidence further suggests designers should treat the full six-category guardrail set, not a subset, as the practical target, since organizations report expecting these collectively rather than selecting among them.

5.4 Implications for Adopting Organizations

For organizations evaluating such platforms, the framework offers a practical checklist: does the tool's conversational or advisory layer have a scope that is clearly bounded and consistently honored; does any autonomous action pass through a human checkpoint before taking effect; is that action auditable after the fact; does the tool demonstrate competence across the organization's actual vendor mix rather than a narrow reference environment; and does the platform additionally provide emergency override and role-based access control as explicit, named capabilities. These questions translate the abstract trust constructs into criteria an evaluating team can apply directly.

5.5 Boundary Conditions

The framework's applicability is bounded by one condition evident in the case itself: it describes trust formation in systems that retain a human-in-the-loop approval step. Its extension to fully autonomous systems, where no such checkpoint exists, is not established by this study and would require separate investigation, since control and benevolence as observed here are constructed specifically around the presence of an approval gate. The survey data suggests this boundary condition currently applies to the clear majority of the field: only 24% of organizations report comfort with fully autonomous action requiring no human review (Cisco and Omdia, 2026). Whether comparable trust can be established in agentic systems that forgo that gate, through other mechanisms such as staged autonomy or confidence-based escalation, is left as an open question for future research.

6. Conclusion

This paper examined how trust and adoption challenges in agentic network operations tooling can be addressed through deliberate design, using a qualitative single-case study of OrcaMonitor triangulated against independent 2026 industry survey data. The analysis found that four mechanisms, specifically read-only conversational access, staged ticket approval, persistent audit logging, and multi-vendor device coverage, correspond closely to established trust-in-automation constructs, namely ability, benevolence, integrity, predictability, transparency, and control, and that this mapping converges with what a broader, independent sample of 1,000 NetOps leaders report requiring before trusting agentic AI to act. From this mapping, the paper proposed a Trust-by-Design framework, arguing that trust in agentic NetOps systems need not be left to accumulate through extended operational use but can be substantially engineered into a platform's interface and workflow at the design stage.

The study's contribution is primarily conceptual and practical rather than statistical: it offers a framework, grounded in established trust and adoption theory and corroborated by independent survey evidence, that both platform designers and evaluating organizations can apply to agentic automation tools in network operations. Its limitations, discussed in Section 3.5, stem chiefly from its single-case design and its reliance on vendor-originated material for the case-specific claims. Future work should pursue three extensions: first, validating the Trust-by-Design framework against practitioner interview or survey data collected directly from operators using OrcaMonitor or comparable platforms, to test whether the trust constructs identified here are actually experienced as intended; second, extending the framework through comparative case studies of other AI Ops platforms with differing approval and transparency architectures, including whether they implement the emergency-override and role-based-access-control guardrails this case did not clearly document, to assess how generalizable the proposed mechanisms are beyond a single vendor; and third, investigating whether and how comparable trust can be established in more fully autonomous systems that operate without a mandatory human approval gate, a boundary condition this study was not positioned to address.

References

- Bajpai, M. (2024). Automating troubleshooting network issues with AIOps and machine learning. *Journal of Artificial Intelligence & Cloud Computing*, 3(1), 1–3. [https://doi.org/10.47363/JAICC/2024\(3\)E149](https://doi.org/10.47363/JAICC/2024(3)E149)
- Cisco and Omdia. (2026). The impact of agentic AI on network operations [Research report]. Cisco Systems, Inc.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340.
- Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1), 50–80.
- Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of Management Review*, 20(3), 709–734.
- Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2), 230–253.

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425–478.

Note on sourcing: platform-specific claims regarding OrcaMonitor's architecture and features are drawn from the platform's own product documentation and demonstration materials; where independent, peer-reviewed sources specific to this platform were not available, this is disclosed in Section 3.5 as a limitation of the case study design. Industry survey statistics are drawn from Cisco and Omdia (2026), an independent survey of 1,000 IT and network operations leaders. Citations above should be verified against the original publications and reformatted to the exact style (e.g., IEEE, ACM, or APA numbered/author-date) required by the target venue before submission.