REDUCING TOTAL COST OF OWNERSHIP WITH TRUSTED AGENTIC IT OPERATIONS

A NEW OPERATING MODEL FOR MANAGED SERVICE PROVIDERS

Droma

EXECUTIVE SUMMARY

MSP margins are eroding in plain sight. Not because service quality has slipped, but because the cost of delivering it keeps climbing faster than revenue.

Modern customer environments generate enormous volumes of events, alarms, logs, metrics, incidents, and changes. Every customer brings a different technology stack, monitoring architecture, ITSM platform, cloud environment, and security tooling. Complexity doesn't grow with the customer count. It compounds against it.

Observability provides visibility. AIOps reduces noise and correlates signals. Automation executes predefined tasks. Generative AI summarizes information and answers questions. Each solves a piece of the operational problem. None closes the gap that matters most: the absence of a system that converts data into decisions, and decisions into action.

For MSPs, that gap has a direct economic cost. Engineers triage alerts, reconstruct context, investigate recurring incidents, search for knowledge, determine the right course of action, and coordinate execution across fragmented systems. As the customer base grows, this workload grows with it. Operational capacity ties itself to headcount. Growth gets harder to scale. Margin takes the hit.

The next evolution of AIOps is not more automation. It is agentic operations, systems that understand context, reason across evidence, determine what should happen next, and execute within defined boundaries.

Droma (formerly Grok), the Role-Based Agentic Operations Platform, is built for this shift. It is not a productivity layer bolted onto existing workflows. It is an operating model that lets MSPs absorb more operational work, so growth no longer requires proportional growth in headcount, and total cost of ownership becomes a controllable input rather than a fixed consequence of scale.

CHALLENGES FACED BY MANAGED SERVICE PROVIDERS

MSP economics are built on scale. Yet the very scale that drives growth can expose the limits of the traditional operating model.

Every new customer introduces a different environment, toolset, runbook, escalation policy, and level of operational maturity. Traditional MSP models absorb that variability with people. More customers mean more data, more alerts, more tickets, more investigation, more specialists, and ultimately, more cost.

The challenge is not the volume of data. It is the growing dependency on people to turn that data into decisions.

When human capacity remains the mechanism for converting operational signals into action, scale increases complexity faster than it increases efficiency.

Traditional AIOps promised to make operations more intelligent. But intelligence that requires significant human effort to prepare and maintain ultimately inherits the same scalability constraints it was meant to address.

The challenge is not configuration alone. It is the ongoing dependency on humans to create rules, tune models, maintain context, validate outcomes, and continuously adapt the system as customer environments change.

That dependency appears in lengthy onboarding, manual rule creation, ongoing tuning, and resources pulled away from customer operations to maintain the platform itself.

The result is intelligence that must be continuously maintained to remain effective—not intelligence that continuously adapts to the environment in which it operates.

Agent autonomy introduces a new economic reality for MSPs: the same capability that allows one agent to handle work across thousands of operational events can also amplify the impact of a single wrong decision.

In a traditional model, human intervention naturally limits the scale and speed at which mistakes propagate. Autonomous operations remove that constraint. A flawed decision can trigger actions across systems, workflows, and potentially multiple customer environments before a human has an opportunity to intervene.

For an MSP, unguided autonomy can turn scale from an advantage into an exposure. A poorly governed agent can turn a single error into an issue that propagates across accounts.

The question is no longer whether agents can do more work. It is how much authority they should have when the cost of being wrong scales with the business itself.

SCALE OPERATIONS WITHOUT SCALING HEADCOUNT

Enterprise AI still gets described as a productivity technology — something that automates tasks and speeds up existing workflows. That framing is useful, but incomplete. For an MSP, the real shift underway is cognitive, and it's the shift that finally lets service-delivery cost stop climbing in lockstep with customer growth.

MSPs aren't short on signal. Every customer environment generates alarms, dashboards, tickets, topology data, and telemetry by the truckload. The shortage is meaning — alarms with no context attached, dashboards that show symptoms but not causes, tickets with no memory of what came before. Closing that gap manually is exactly where MSP operator cost concentrates today.

Resolving an incident was never really one task. It's a chain of judgment calls: reading signal correctly, remembering what happened last time, reasoning through cause, coordinating across tools and teams, then acting — often across dozens of environments at once, faster than any engineer can hold in their head. Every one of those steps, done manually, is effort that scales with the customer base instead of shrinking against it.

> Agentic AI changes that equation by wrapping a persistent layer of intelligence around the complexity itself — one that correlates evidence, retains memory across incidents, explains likely cause, and acts within governed limits. For an MSP, that layer becomes something closer to a shared service-assurance brain: connecting alarms, topology, tickets, customer impact, and engineer expertise into one system built for understanding, not just visibility. What used to require a person reconstructing context from scratch now happens inside the system itself — which is precisely where the cost used to live.

That doesn't put the operator out of a job. It puts them in a different one, and a more economically efficient one for the business.

In the model MSPs run today, the operator is largely a manual correlator — watching dashboards, chasing alarms, checking tickets, searching logs, inspecting topology, tracking down who owns what, hunting for a similar incident in memory. That's headcount tied directly to ticket volume. In the agentic model, that same person becomes something closer to a supervisor, a teacher, and a governor of the system doing that work. They oversee the investigation, teach the system which patterns actually matter, validate or reject its hypotheses, and set the boundaries of what it's allowed to act on. They still own the calls that should never be automated — when uncertainty means the situation needs to escalate, when a specific customer's context changes the right answer, when an action is technically sound but operationally the wrong move.

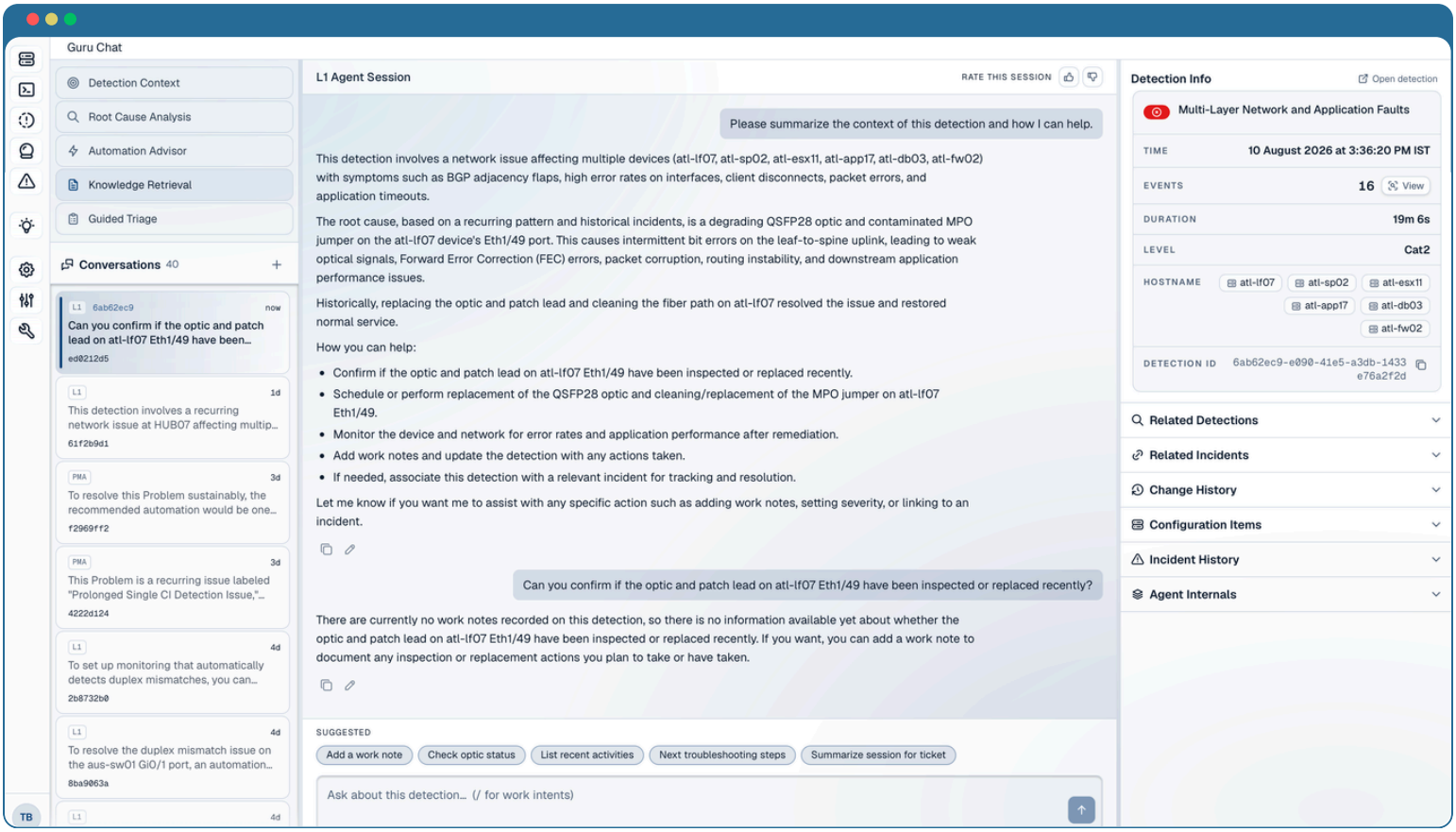
That's a higher-leverage role, not a lower-cost headcount to eliminate. It is not a lower-skill one either — if anything, it demands more: an operator who can tell when the agent's read is right, when it's incomplete, or when it's missing context only a human would catch. This is where agentic operations actually pays off on the TCO line: the same operator now oversees judgment across many accounts instead of executing tasks in just one, which means the ratio of human effort to customers served finally starts moving in the MSP's favor. The human hasn't left the loop. The loop has simply become more efficient to run, and more valuable to be part of.

HOW Droma HELPS

Droma L1/L2 Agent closes the gap between detecting a issue and knowing what to do about it. It evaluates every detection, investigates causally, and acts — gathering evidence, diagnosing likely causes, and retrieving relevant knowledge without waiting for a human to validate what the system already knows. Validated, previously resolved issues resolve autonomously. Everything else escalates with full context and a recommended next step already attached.

When an incident fires, the sequence runs without intervention:

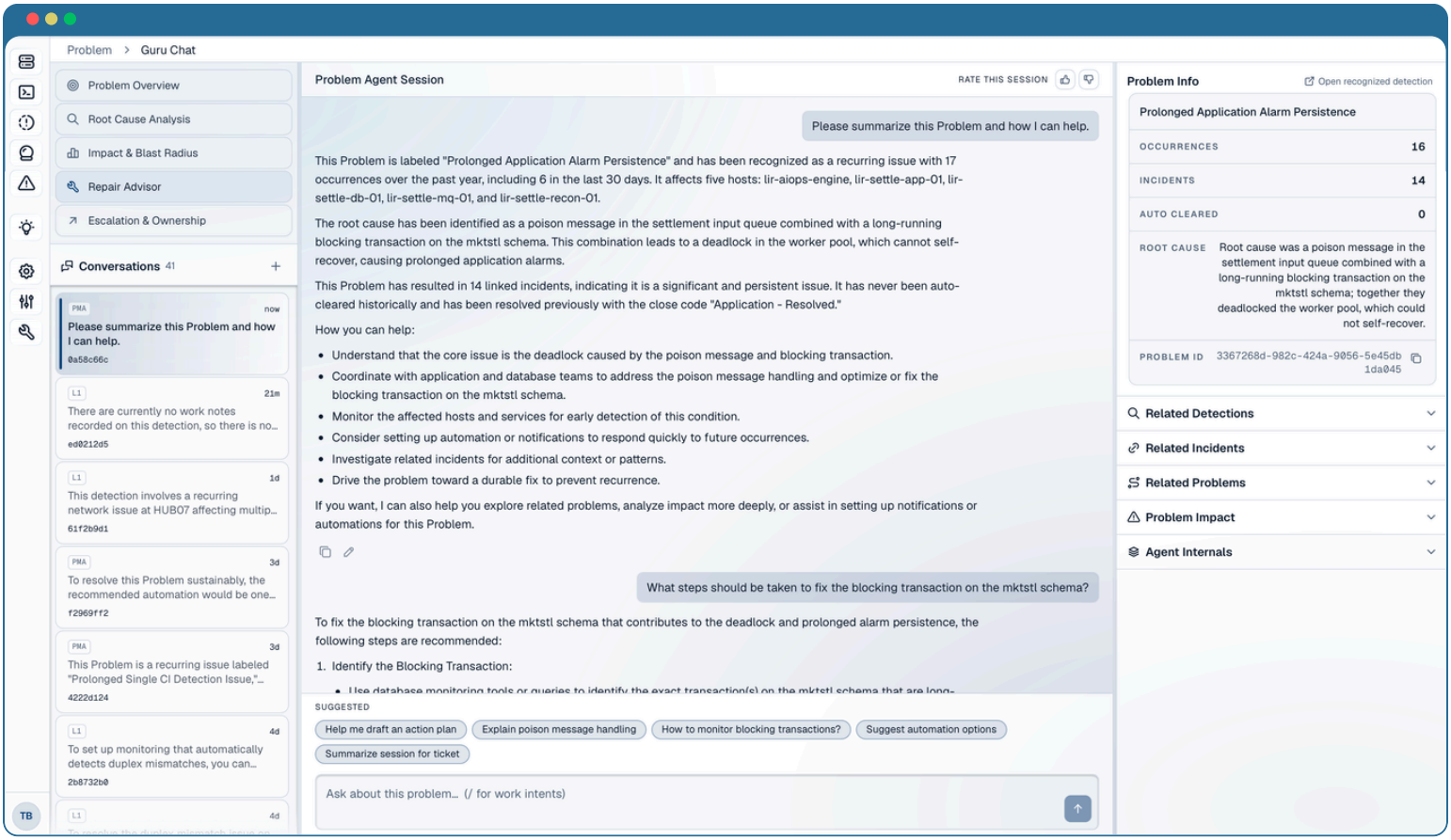
- **Ingest** — The agent pulls enriched context from Dynamic Data Fusion layer: historical patterns, recent changes, service dependencies, and cross-domain signals already correlated.
- **Reason** — Guru analyzes the enriched incident, identifies probable root cause, and determines whether the pattern is validated, novel, or ambiguous, grounded in your SOPs, runbooks, KEDB articles, and escalation procedures.
- **Act** — Recommends or autonomously executes governed actions, updates workflows, generates scripts, and orchestrates specialized operational agents and enterprise workflows in accordance with organizational policies and risk thresholds. Validated, previously resolved issues resolve autonomously. Novel or ambiguous ones route to the right team with causal context and recommended next steps already attached.
- **Learn** — Every outcome, automated or escalated, feeds back into the model — improving accuracy and expanding autonomous coverage over time.



While most tools treat problems as records to investigate, document, and eventually close, **Droma Problem Manager Agent** turns it into a continuously learning operational object, building context across recurrence, investigation, decisions, and outcomes. It orchestrates specialized operational agents for root cause analysis, impact and blast-radius assessment, repair guidance, and escalation and ownership — bringing investigation, reasoning, and governed action together through a single cognitive layer.

When a recurring pattern emerges, the sequence runs continuously:

- **Detect** — The agent identifies patterns across detections and incidents, correlating frequency, similarity, severity, and history to distinguish systemic problems from isolated events.
- **Investigate** — Specialized agents analyze the Problem across its history, identify the underlying root cause, and determine why it persists, its impact, and its blast radius, grounded in correlated incidents, detections, changes, service dependencies, operational knowledge, and prior remediation outcomes.
- **Recommend** — The agent translates that understanding into the next best course of action, evaluating available remediation paths by relevance, expected impact, and operational fit — from suppression and soak periods to notification or ticketing changes, automation, or escalation.
- **Learn** — Every intervention and outcome becomes operational memory, strengthening pattern recognition, sharpening recommendations, and reducing the effort required to understand and resolve future recurrences.



MAKE INTELLIGENCE ADAPTIVE, NOT MAINTENANCE-HEAVY

Intelligence that requires continuous human maintenance inherits the limitations of the people maintaining it. It may automate analysis, but it still relies on people to define what matters, establish the context in which decisions are made, and continually adjust the system as environments evolve.

This is the fundamental limitation of traditional AIOps. Intelligence is often built around conditions that humans define in advance: rules, thresholds, topology, correlations, classifications, and other representations of how the environment is expected to behave. When the environment changes, those representations need to change with it. The intelligence does not inherently adapt; people adapt it.

That model becomes increasingly difficult to sustain as operational environments grow more distributed and dynamic. Services are added and removed. Dependencies shift. Infrastructure changes. Traffic patterns evolve. New failure modes emerge. Customer environments behave differently from one another. The more variables an MSP operates across, the more effort is required to keep manually maintained intelligence aligned with reality.

Adaptive intelligence requires a different foundation. It must be able to learn from the environment rather than depend on humans to continually describe that environment to it. That starts with data.

Operational data is not simply an input to intelligence. It is the context from which intelligence is built and the feedback through which it evolves. The system must continuously transform raw signals into meaningful context, recognize patterns in that context, understand how those patterns change over time, and incorporate the outcomes of previous decisions into what it understands next.

This creates a fundamentally different operating model. Instead of continuously tuning intelligence to keep pace with the environment, the intelligence continuously learns from the environment itself.
For MSPs, this distinction is particularly important. Intelligence that requires more human effort as the customer base and operational complexity grow simply reproduces the economics of the model it was meant to improve. Adaptive intelligence creates the possibility of the opposite: intelligence that becomes more capable as it encounters more operational experience, without requiring a proportional increase in human maintenance.

HOW DROMA HELPS

Droma's Dynamic Data Fusion ingests telemetry, including but not limited to, events, alarms, logs, metrics, incidents, and changes from across your environment, then cleanses, normalizes, maps, and enriches them into structured inputs ready for real-time machine learning and automated action. Unlike platforms that treat ML as a bolt-on, Droma fuses data preparation and intelligence at the source — turning complexity into clarity, and alerts into outcomes.

The result is an AI-readiness layer that makes everything downstream measurably better: sharper detection, earlier prediction, more confident automation. As Droma's Cognitive AI engine learns before, during, and after every incident, Dynamic Data Fusion ensures it always learns from signal-rich, model-ready data.

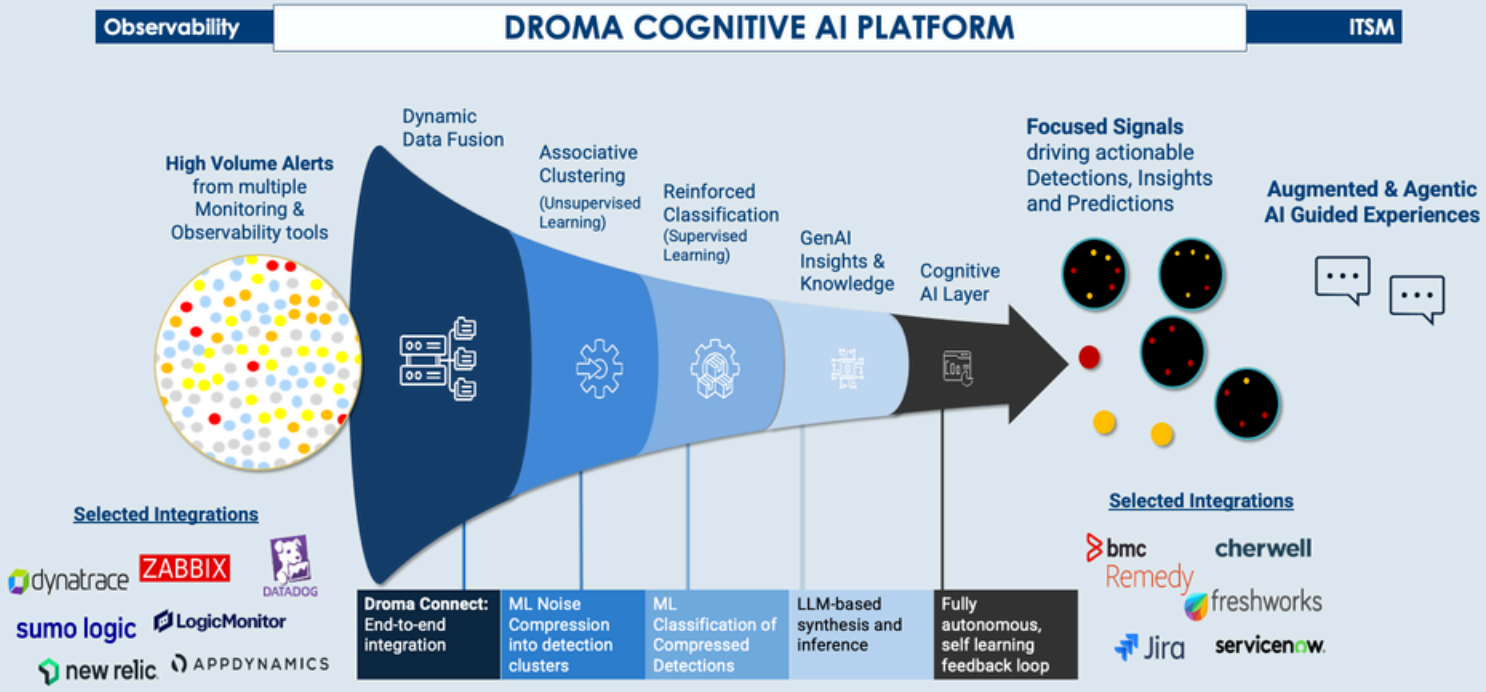
Above the data layer sits a coordinated set of learning functions that form the system's reasoning capability. These are not independent models operating in isolation. They are stages in a structured process that incrementally build understanding.

The system does not rely on a single learning model. It employs multiple approaches — anomaly detection, clustering, classification, and causal inference — applied dynamically based on the nature of the data and the problem at hand. This multi-model approach delivers greater accuracy and adaptability than systems constrained to a single algorithm.

The process is sequential and deliberate. Deviations from learned baseline behavior are identified first. Those deviations are grouped into clusters representing potentially related activity. Clusters are classified based on patterns learned from historical incidents, and their evolution is tracked over time. Finally, causal inference is applied to determine the most likely origin and propagation of the incident.

What distinguishes this approach from traditional AIOps is the emphasis on causality rather than correlation. Correlation identifies patterns that occur together; causality seeks to explain why. This distinction is critical for decision-making, as actions must be based on cause, not coincidence.

Over time, reinforcement mechanisms refine the system's behavior by incorporating the outcomes of previous decisions. Every resolved incident makes the next one easier to predict, faster to diagnose, and less likely to recur.



From dispersed alerts to decisive actions. Droma continuously learns, reasons, and acts to prevent issues faster, reduce noise, and drive operational outcomes.

ENABLE GOVERNED AGENTIC MSP OPERATIONS

Autonomy doesn't just multiply what an agent can do. It multiplies what a mistake can cost — across every customer environment it touches. For an MSP, that risk isn't theoretical. A flawed decision in one account doesn't just cost that account — it costs credibility with every account watching.

So the real question isn't whether an agent is capable enough. It's whether the MSP has the governance in place to see what the agent did, understand why, and step in when it matters — because that governance, not raw capability, determines how much work can safely move off a human's desk and onto an agent's.

That governance rests on three things: every action is visible, every action is explainable, and every action is reversible. Skip any one, and the MSP is left choosing between trusting the agent blindly or shadowing every action it takes.

This is where governance becomes a direct lever on total cost of ownership. Actions gated by a confidence threshold, scoped by policy, and checkpointed for a human only where the stakes genuinely call for it — that's what lets an MSP hand an agent more investigation, more remediation, more of the judgment calls that used to sit exclusively with a human operator. And every decision becomes a recorded outcome that sharpens the next one — so the agent doesn't just do more work, it gets measurably better at it.

The economics compound from there. Stronger guardrails let an MSP expand what an agent is trusted with. The more an agent is trusted with, the more human hours get freed from repetitive investigation and redirected to the accounts and escalations that actually need a person. That's not a productivity anecdote — it's a structural shift in how service-delivery cost scales against customer growth. Governance isn't what stands between an MSP and lower TCO. It's the mechanism that gets them there.

HOW DROMA HELPS

Trust and control are core to Droma. The platform ensures a secure architecture with role-based access controls, plus stronger capability governance through a registry and tenant-level controls. It also supports prompt, model, and action traceability along with an audit trail for support and operational review. Together, these capabilities make the agent more deployable in real customer environments without overextending into unsafe autonomy.

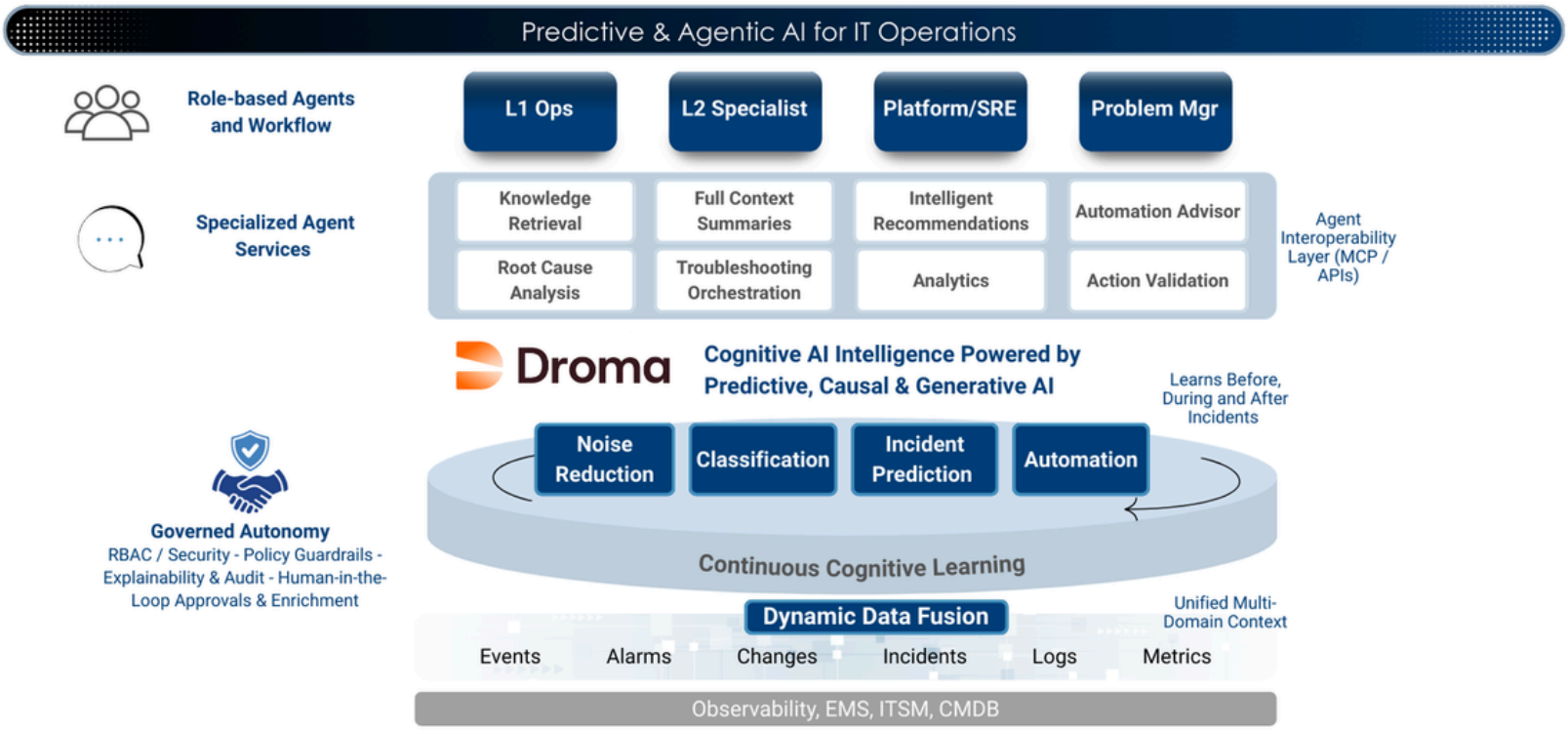
Droma puts governance into practice through three distinct layers:

- At the system level, organizations decide when human approval is required before an agent can act — the broadest control, set once and applied consistently across every agent operating underneath it.
- At the agent level, that control gets specific. Each role-based agent — L1/L2, Problem Manager — has its own defined scope: what it's responsible for, what systems it's allowed to touch, and where its authority stops.
- At the action level, governance goes granular. Pulling a log, opening a ticket, triggering an automation, calling another service — each capability carries its own permission, evaluated on its own terms, not inherited automatically from the agent performing it.

HOW DROMA WORKS

Droma operates as a cognitive AI layer across your environment, ingesting signals, learning continuously, and orchestrating agent-driven actions across IT Operations and ITSM.

- **Ingest & Unify Signals -** Droma ingests events, alarms, logs, metrics, incidents, and changes through Dynamic Data Fusion, creating a real-time, unified operational context across systems.
- **Learn & Reason Continuously -** Using predictive, causal, and generative AI, Droma learns before, during, and after incidents, improving detection accuracy, understanding relationships, and building a continuously evolving model of your environment.
- **Detect, Predict, and Decide** - Droma compresses noise into actionable detections, classifies issues, and identifies risks hours in advance, delivering context-rich insights, root cause analysis, and recommended actions.
- **Orchestrate Agent-Driven Workflows -** Role-based agents (L1, L2, SRE, and Problem Manager) leverage specialized agent services to triage, investigate, and resolve issues, coordinating actions across tools and workflows.
- **Execute with Governed Autonomy -** Actions are executed through automation and orchestration, with human-in-the-loop approvals, policy guardrails, and full explainability, progressing toward self-healing IT operations.



ABOUT DROMA

Droma delivers predictive, agentic, and self-healing IT operations through a cognitive AI platform that continuously learns from signals, incidents, actions, and outcomes. Built for rapid deployment and seamless integration, Droma enables organizations to move from reactive operations to autonomous, outcome-driven performance. We partner with high-growth and large enterprises across oil and gas, data center, fintech, and other industries, as well as CSPs and MSPs, supporting mission-critical, real-time environments where reliability, scale, and customer trust are non-negotiable.

Contact us at info@Droma.ai to learn how to modernize your IT Operations.