



ELIMINATE RECURRING ISSUES WITH AGENTIC PROBLEM MANAGEMENT: **GROK PROBLEM MANAGER AGENT** AI THAT REASONS, DECIDES, AND ACTS

Problem Management was created to stop IT from solving the same failure twice. Yet recurring failures continue to consume teams because the context behind them is scattered across incidents, detections, changes, knowledge, and operational history. Problem Managers are pulled into live incidents, leaving systemic issues to wait. Root cause analysis remains manual, and what teams learn from one recurrence rarely becomes intelligence for the next. Traditional AIOps tools can surface patterns and suggest causes, but insight alone does not close the loop. Grok changes that. As part of the broader Role-Based Agentic Operations Platform, powered by a cognitive AI engine, Grok's Problem Manager Agent orchestrates specialized operational agents to perform root cause analysis, assess impact and blast radius, recommend repair strategies, and determine escalation and ownership — turning Problem Management from a retrospective process into a continuously learning, actionable discipline.

Overview

Every recurring incident is evidence that the system has learned too little from what happened before. Problem management was designed to turn those failures into systemic improvement, yet most organizations still treat problems as records to investigate, document, and eventually close. The result is a costly cycle: the same patterns return, teams reconstruct the same context, and hard-won operational knowledge remains trapped in individual investigations.

Grok Problem Manager Agent changes this. As part of the Role-Based Agentic Operations Platform, it turns the Problem 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.

Every resolved problem strengthens the system's understanding of what to recognize, prioritize, and prevent next. The Problem Manager Agent is the next step beyond incident-level autonomy, extending Grok from resolving what is happening now to continuously learning from why it happens — advancing toward a broader, progressively autonomous system.

HOW THE GROK PROBLEM MANAGER AGENT WORKS

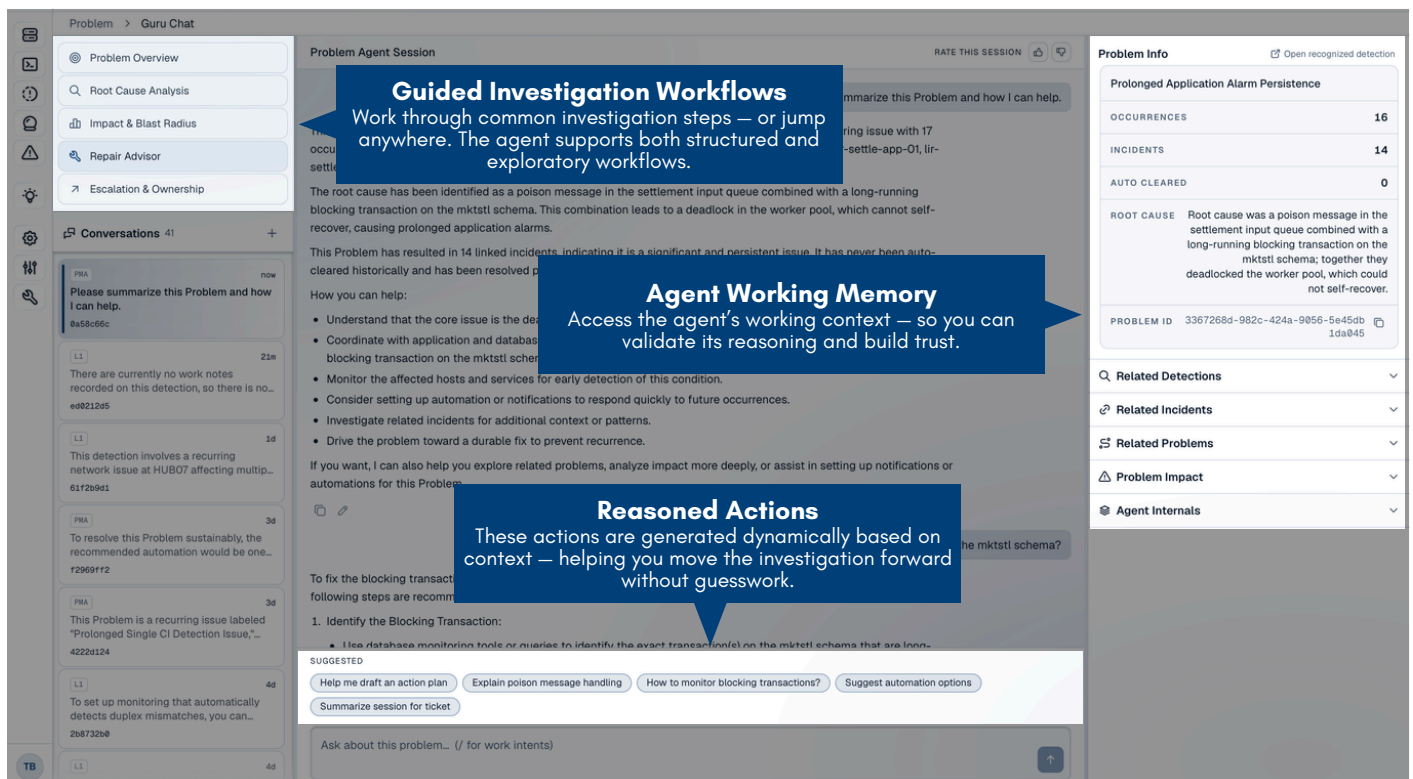
Problem management breaks down when every recurrence becomes a new investigation. The evidence exists, but it is fragmented across incidents, detections, changes, dependencies, and past remediation. Teams spend valuable time reconstructing what has already been learned instead of using that knowledge to prevent the next failure.

Grok Problem Manager Agent is built to break that cycle.

Dynamic Data Fusion connects operational evidence across domains, bringing recurring detections, incident history, changes, service dependencies, and prior interventions into a unified context. Guru, Grok's generative AI, turns that context into operational understanding — what is recurring, why it is happening, how far it reaches, and what has already been tried. The Problem Manager agent orchestrates specialized operational agents to determine the highest-value path forward, from repair guidance to escalation and ownership.

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.



The screenshot displays the Grok Problem Manager Agent interface, showing a problem session for "Prolonged Application Alarm Persistence". The interface includes a sidebar with navigation options, a main chat area with a conversation history, and a right-hand panel with problem details and related items.

Guided Investigation Workflows
Work through common investigation steps — or jump anywhere. The agent supports both structured and exploratory workflows.

Agent Working Memory
Access the agent's working context — so you can validate its reasoning and build trust.

Reasoned Actions
These actions are generated dynamically based on context — helping you move the investigation forward without guesswork.

Problem Info
Prolonged Application Alarm Persistence

Metric	Value
OCCURRENCES	16
INCIDENTS	14
AUTO CLEARED	0

ROOT CAUSE
Root cause was a poison message in the settlement input queue combined with a long-running blocking transaction on the mktst schema; together they deadlocked the worker pool, which could not self-recover.

PROBLEM ID
3367268d-982c-424a-9056-5e45db1da045

Related Detections
Related Incidents
Related Problems
Problem Impact
Agent Internals

Conversations
41

Problem Agent Session
The root cause has been identified as a poison message in the settlement input queue combined with a long-running blocking transaction on the mktst schema. This combination leads to a deadlock in the worker pool, which cannot self-recover, causing prolonged application alarms.

How you can help:

- Understand that the core issue is the deadlock in the worker pool.
- Coordinate with application and database teams to resolve the blocking transaction on the mktst schema.
- Monitor the affected hosts and services for early detection of this condition.
- Consider setting up automation or notifications to respond quickly to future occurrences.
- Investigate related incidents for additional context or patterns.
- Drive the problem toward a durable fix to prevent recurrence.

1. Identify the Blocking Transaction:
Use database monitoring tools or queries to identify the exact transaction(s) on the mktst schema that are locking the database.

SUGGESTED ACTIONS:

- Help me draft an action plan
- Explain poison message handling
- How to monitor blocking transactions?
- Suggest automation options
- Summarize session for ticket

WHAT MAKES GROK PROBLEM MANAGER AGENT DIFFERENT

- **Problem Becomes the Unit of Intelligence.** Grok doesn't treat a problem as a static record. It continuously builds context around the underlying pattern — connecting detections, incidents, changes, dependencies, prior interventions, and outcomes.
- **Cognitive Learning Foundation.** Classical ML identifies recurring patterns, clusters related detections, and surfaces problems based on frequency, impact, and historical context. Generative AI drives reasoning, explanation, and interaction. Both run on the same enriched signal Dynamic Data Fusion prepares, and both get sharper with every problem — learning from how patterns emerge, how they are investigated, and what ultimately resolves them.
- **Specialized Agents, Orchestrated for Depth.** The Problem Manager Agent orchestrates specialized agents for root cause analysis, impact and blast-radius assessment, knowledge retrieval, repair guidance, and escalation and ownership — each contributing focused expertise to the problem lifecycle. Depth without slowing down execution.
- **Governed Autonomy, Built In.** Grok governs agent behavior across three levels. System level sets whether human approval is required. Agent level defines what agents can do and which systems they reach. Action level assigns risk, approval rules, and eligible personas to every action, including operational limits like rate limits and timeouts.
- **Next Step Towards Governed Autonomous Operations.** Grok extends governed autonomy beyond individual incidents to the problems behind them. Each problem becomes a persistent unit of reasoning, learning, and action — continuously enriched by new evidence and outcomes.



GOVERNED AUTONOMY BY DESIGN

Grok is built to anticipate the specific risks that come with agentic reasoning, memory, and action — not patch them after the fact.

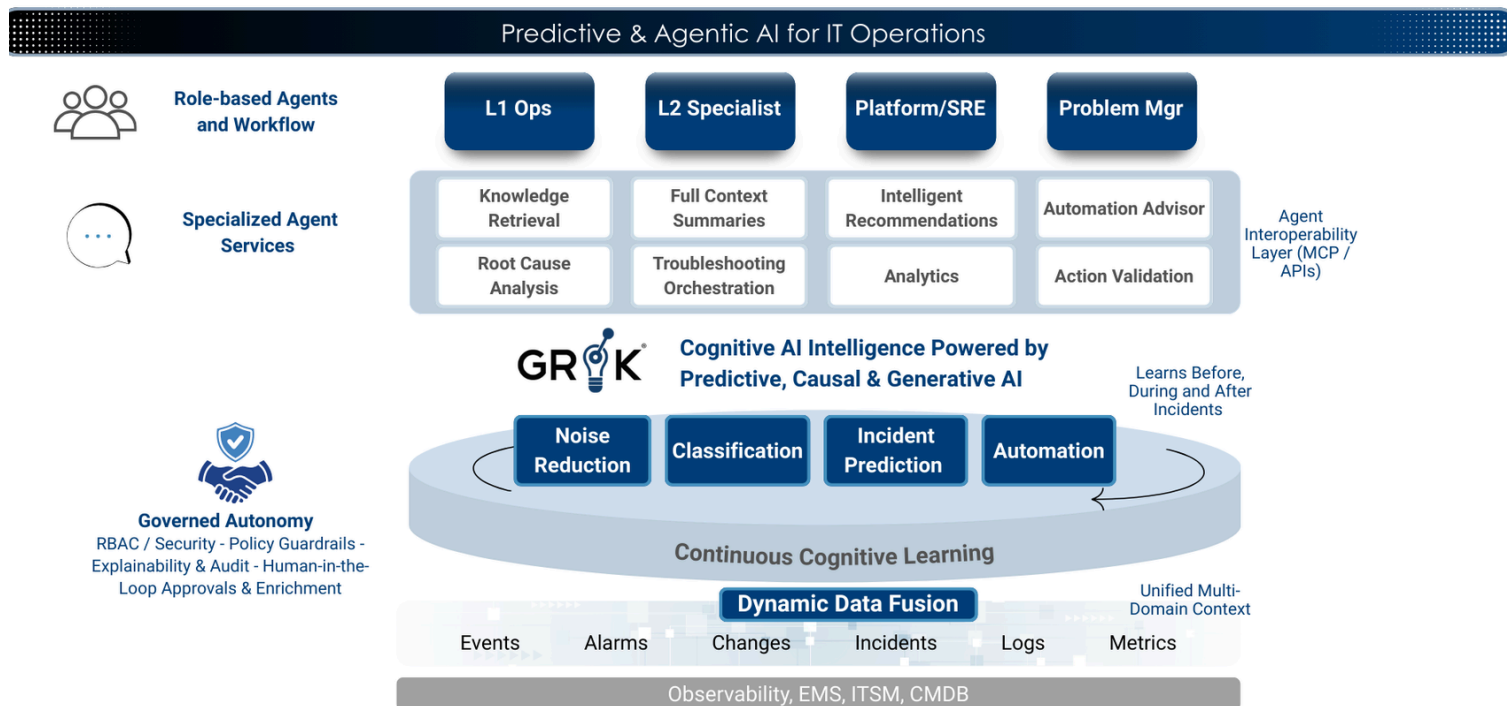
- **Cross-Domain Cascading Failures** — Federated oversight and simulation, dependency and impact modeling, and global state awareness prevent isolated actions from triggering unintended downstream effects.
- **Memory & Learning Corruption** — Versioned writes and checkpoints, reflect-before-commit validation, and belief graph validation protect the integrity of what the agent learns over time.
- **Hallucination & False Confidence** — Confidence gates, auto-rollback and fallback plans, and human-in-the-loop review for low-confidence cases ensure the agent never acts on certainty it hasn't earned.
- **Data Leakage & Boundary Violations** — Per-agent data boundaries, role-based access control from the ground up, and memory firewalls between agents keep context properly isolated.
- **Unsafe Actions & Tool Invocations** — Role- and risk-based execution controls and human approval for high-impact tasks govern what the agent can act on and when. Agent readiness state validation ensures correct configuration and grants are in place before any agent is permitted to act.
- **Prompt Manipulation** — A governance reasoner invoked on every turn assesses unsafe elements in the pre-generation prompt, failing closed or sanitizing before injection. System-level guardrails, intent filtering, and instruction hierarchy enforcement protect the integrity of the agent's reasoning.
- **Agent Identity & Role Management** — Defines each operational agent's identity, responsibilities, permissions, governance policies, and memory boundaries.



HOW GROK WORKS

Grok 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** - Grok 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, Grok learns before, during, and after incidents, improving detection accuracy, understanding relationships, and building a continuously evolving model of your environment.
- **Detect, Predict, and Decide** - Grok 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 GROKSTREAM

Grokstream 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, Grok 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@grokstream.com to learn how to modernize your IT Operations.