Optimizing NOC Operations Through an Agentic Approach: A Path Toward NOCless, White, and Dark NOCs

Contents

Abstract 3
Introduction 3
Architecture deep dive 5
Incident Co-Pilot detailed flow 8
Conclusion 13
Authors 13
Reviewers 13
References 13

Abstract

This white paper explores how an agentic approach can transform Network Operations Centers (NOCs) into fully autonomous models—NOCless, white, or dark NOCs—that minimize or eliminate manual intervention. By leveraging AI-driven automation platforms aligned with AI agents based on the TM Forum Incident Co-Pilot Catalyst project and the Cisco Crosswork® Multi-Agentic AI Framework, service providers can optimize business and technical processes traditionally managed by NOC personnel. The paper presents a comparative analysis of these solutions, focusing on architecture, use cases, and agent management strategies that collectively enable the journey toward autonomous NOC operations.

Introduction

NOCs are critical to ensuring continuous service availability and performance in service provider networks. Traditionally, NOC operations rely heavily on manual workflows, which can introduce delays, increase operational costs, and impact customer satisfaction. The vision of NOC automation aims to reduce or eliminate manual intervention, leading to operational models known as NOCless, white, or dark NOCs—where the NOC is fully autonomous or requires minimal human oversight. Our proposed approach here is AI agentic operation. Agentic automation uses intelligent agents to adapt and learn from complex, changing environments, while traditional automation relies on rigid rules for repetitive tasks. Choosing AI agentic automation enables more flexibility, resilience, and efficiency in handling dynamic business processes.

This paper explores how an agentic approach based on the TM Forum Incident Co-Pilot Catalyst project and built on Cisco's Crosswork Multi-Agentic AI Framework is enabling the transformation of network operations to dark/white NOCs.

The Crosswork Multi-Agentic AI Framework introduces a platform where AI agents, powered by both Large Language Models (LLMs) and Machine Learning (ML) techniques, collaborate to automate incident management and operational workflows. These agents bring together knowledge graph, advanced reasoning, prediction, best practices injected through Retrieval-Augmented Generation (RAG), and orchestration capabilities to streamline NOC processes, reduce Mean Time to Resolution (MTTR), and enhance overall service assurance.

How an agentic approach adds value for NOCless or dark/white NOCs

Traditional automation typically relies on static scripts or narrowly defined playbooks, which often lack adaptability when incidents deviate from expected patterns. In contrast, an agentic AI system:

- **Adapts dynamically** by using reasoning-driven agents that learn and improve over time.
- **Reduces operational silos** by enabling cross-domain coordination between specialized agents (IP, optical, security, etc.).
- **Delivers faster outcomes** by autonomously triaging, diagnosing, and recommending actions—while still allowing human-in-the-loop oversight.
- **Optimizes OpEx** by minimizing repetitive manual tasks and accelerating service recovery.

By shifting from rigid automation to **agentic intelligence**, operators gain a scalable and flexible model that not only improves today's operational efficiency but also builds the foundation for **autonomous NOCs of the future**.

The Incident Co-Pilot supports the end-to-end lifecycle of incident management in the NOC, going well beyond just business processes to include technical operations, real-time incident resolution, and continuous improvement of network reliability and security. By addressing both business and technical processes, the solution enables a gradual shift toward autonomous NOC operations, enhancing productivity and operational agility. Examples of business processes addressed include incident ticketing, compliance reporting, Root Cause Analysis (RCA), release approvals, and other workflow approvals, while technical processes involve network monitoring, fault detection, automated diagnostics, RCA, and system recovery. The Incident Co-Pilot streamlines both types of processes to help ensure efficient and reliable NOC operations.

- Business processes: Incident ticketing, compliance reporting, RCA documentation, release approvals, and workflow governance.
- Technical processes: Network monitoring, fault detection, automated diagnostics, RCA, and system recovery.

By streamlining both dimensions, the Incident Co-Pilot helps ensure more efficient, reliable, and resilient NOC operations.

Architecture deep dive

TM Forum Incident Co-Pilot architecture

The TM Forum Incident Co-Pilot is a multi-agentic system designed to enhance NOC engineers' capabilities by integrating advanced AI technologies, including LLMs and RAG knowledge bases enriched with telecom domain expertise. The architecture shown in Figure 1 uses a conversational interface or integration with case ticketing to feed an agentic platform that orchestrates specialized agents for incident detection, diagnosis, and resolution.

Key components include:

- **Incident Co-Pilot UI:** Simplifies complex NOC workflows through conversational interaction.
- **Incident agent:** Performs RCA using real-time alarms, telemetry, and topology data. In addition, develops and executes healing plans, including trouble ticket creation and repair recommendations.
- **Net-query agent:** An intelligent tool that automatically queries network devices to gather real-time data, diagnose issues, and deliver actionable insights for efficient network operations and troubleshooting.
- **RAG knowledge base:** Combines unstructured text, Q&A pairs, rules, and knowledge graphs to provide context-aware insights.
- **Integration with external systems:** Interfaces with OSS, field maintenance engineers, and other platforms via TM Forum open APIs provided by the agentic platform.

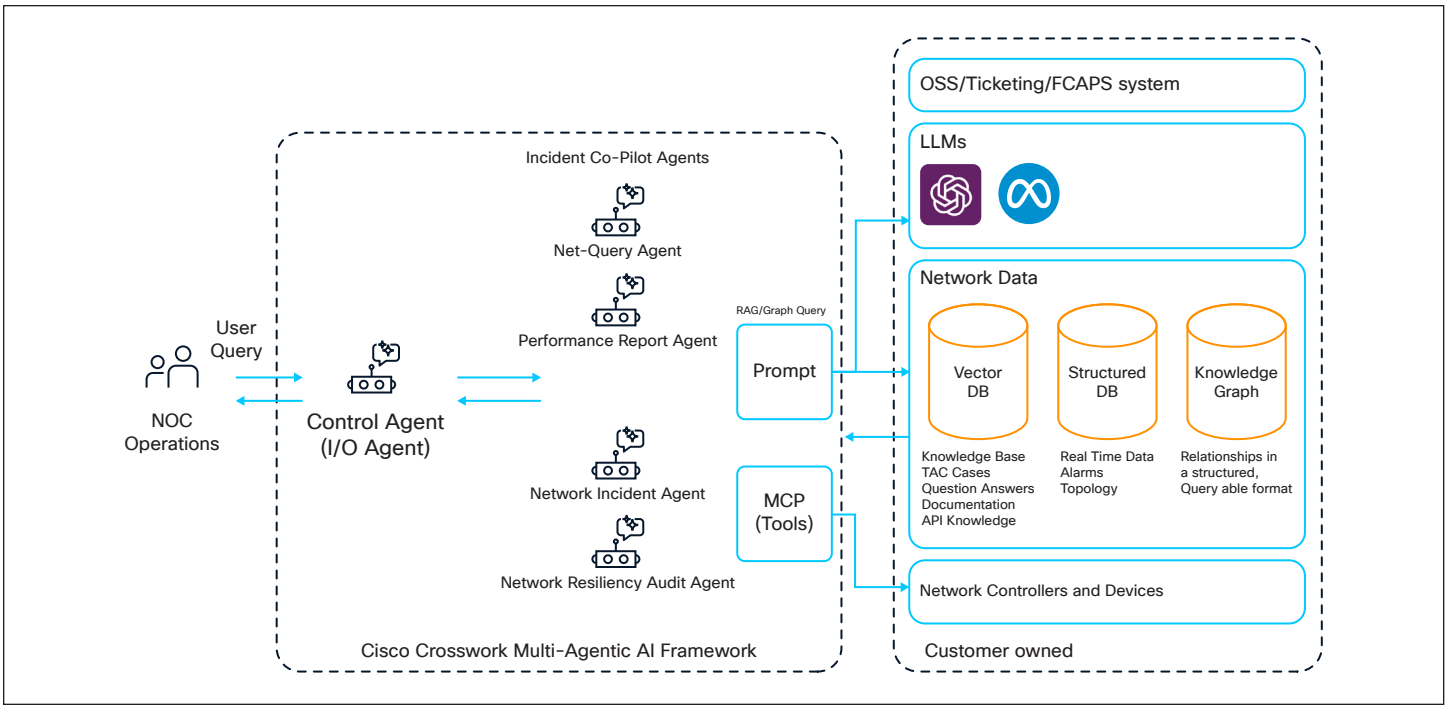

Figure 1. Incident Co-Pilot architecture

This architecture fosters transparency and trust by exposing agent decision-making processes to NOC engineers, supporting a collaborative human-AI environment.

Cisco Crosswork Multi-Agentic AI Framework

Cisco Crosswork Multi-Agentic AI Framework as shown in Figure 3 delivers a containerized and scalable platform for rapidly developing, customizing, and deploying AI agents. Supporting both LLM- and ML-based agents, it enables seamless agent-to-agent communication and orchestration, driving the shift toward autonomous NOC operations.

Key capabilities

The following capabilities enhance the outcome of the agentic approach.

- **Unified agentic platform:** Standardized APIs, catalogs, lifecycle management, and data retrieval services for AI agents.
- **Specialized AI agents:** Examples include configuration drift, toxic factor detection, net-query, performance reporting, and troubleshooting agents.
- **Knowledge graph:** Semantically aligns all agents through a digital network representation built on Crosswork Network Control Suite capabilities.
- **Data integration:** Combines telemetry, alarms, topology, and inventory data enhanced by RAG.
- **Toolchain integration:** Works with Model Context Protocol (MCP), LangChain, LangGraph, and other frameworks for intent-based automation and continuous learning.
- **Scalable architecture:** Micro services and containerization for on-demand deployment and scaling of NOC-specific agents.

With its modular, extensible design, the Crosswork Multi-Agentic AI Framework empowers service providers to progressively evolve into fully autonomous NOCs.

Use cases

This section covers a few use cases specifically built for NOC automation. While the focus has been on reactive AIOps, we can also build agents that enable proactive and prescriptive AIOps.

Net-query agent

Purpose: Streamlines access to network data by abstracting the complexities of vendor-specific Command-Line Interfaces (CLI) and APIs and provides a unified and simplified way for users to interact with network devices or inventory data, regardless of the underlying vendor technologies.

Capabilities: The net-query agent responds to natural language queries about device status, configurations, and metrics. Using AI/ML and RAG, it searches network inventory and command databases to identify the right information. When a user submits a query, the system converts it into the appropriate device-specific command and executes it on the device using integrations such as MCP servers to retrieve outputs. It supports requests from NOC staff and other AI agents, delivering timely and accurate network insights.

Impact: Reduces technical barriers, accelerates information retrieval, eliminates need for multiple tools, and supports faster decision making.

Example: A NOC operator can ask for the status of Intermediate System-to-Intermediate System (IS-IS) interfaces on a device in plain language, and the tool will return the result; there are no specific vendor commands that the operator needs to know.

Performance report agent

Purpose: Generates detailed performance reports from collected network data, producing both executive and technical reports with minimal coding effort by leveraging RAG techniques and predefined report templates.

Capabilities: The agent analyzes time-series network Key Performance Indicator (KPI) data such as interface throughput, device metrics, and overall network health to generate user-defined reports. It uses the AI agent-driven approach with prompt engineering to convert user queries into data retrieval commands, pulling information from various time-series data sources. The tool extracts key metrics and custom insights like interface names, device IDs, utilization stats, and descriptions from time-series databases or Comma-Separated Value (CSV) metric files and presents them in a clear report format. In this way reports can be generated without the need to explicitly code for each template, significantly reducing the time required to deliver network insights in a user-defined format.

Impact: Cost savings by eliminating repetitive human tasks. On-demand report generation for compliance.

Example: A NOC operator can request an interface utilization report for the entire network or a subset of devices, specifying key columns and metrics such as mean and maximum values. The AI agent then queries the time-series database, calculates the mean and maximum if they are not present in the original data, and generates a CSV report based on the user's requirements, which is made available for download.

Troubleshooting agent

Purpose: Plays a central role in incident management by using root cause analysis to troubleshoot networking issues based on Intellectual Capital (IC) stored as Methods of Procedures (MoPs). This IC represents trusted data that can deliver reliable and highly deterministic outcomes. The reliance on LLM capabilities is minimized, instead leveraging the expertise of Subject Matter Experts (SMEs). A more advanced use case leverages a knowledge graph–based network topology to identify service health and perform impact analysis. In addition, semantic correlation through context engineering will help achieve faster and more accurate resolution of complex troubleshooting issues. Our approach is focused on a deterministic behavior.

Capabilities: This agent can be triggered automatically by a network incident or manually when a user submits a troubleshooting query. Using RAG, it retrieves relevant troubleshooting guides and MoPs from its database, which serve as step-by-step guidelines for resolving issues on network devices. The agent executes these MoPs step by step through integration with MCP servers, retrieving information from the network. At each step, it observes outputs, applies reasoning, and suggests remediation for the issue. Alternatively, the agent can perform remediation actions in the network via MCP servers, requiring human approval when needed. This will evolve to more complex scenario with Knowedge Graph eventually replacing the MOP based technique and providing similar deterministic behavior.

Impact: Minimizes outages, improves MTTR, reduces dependency on Layer 2/Layer 3 human experts.

Example: If a wrong interface is shut down and causes a network issue resulting in error logs, the troubleshooting agent will retrieve the relevant MoP document and follow a step-by-step troubleshooting procedure to identify the root cause. It can then trigger a service to “unshut” the interface, using the Crosswork Network Services Orchestrator (NSO) MCP server as shown in Figure 2. MCP is a client/server protocol designed to enable AI applications and agents to interact with various tools, data sources, and APIs in a standardized way. It facilitates automation by acting as a centralized hub for managing diverse systems, including orchestration platforms like Crosswork NSO.

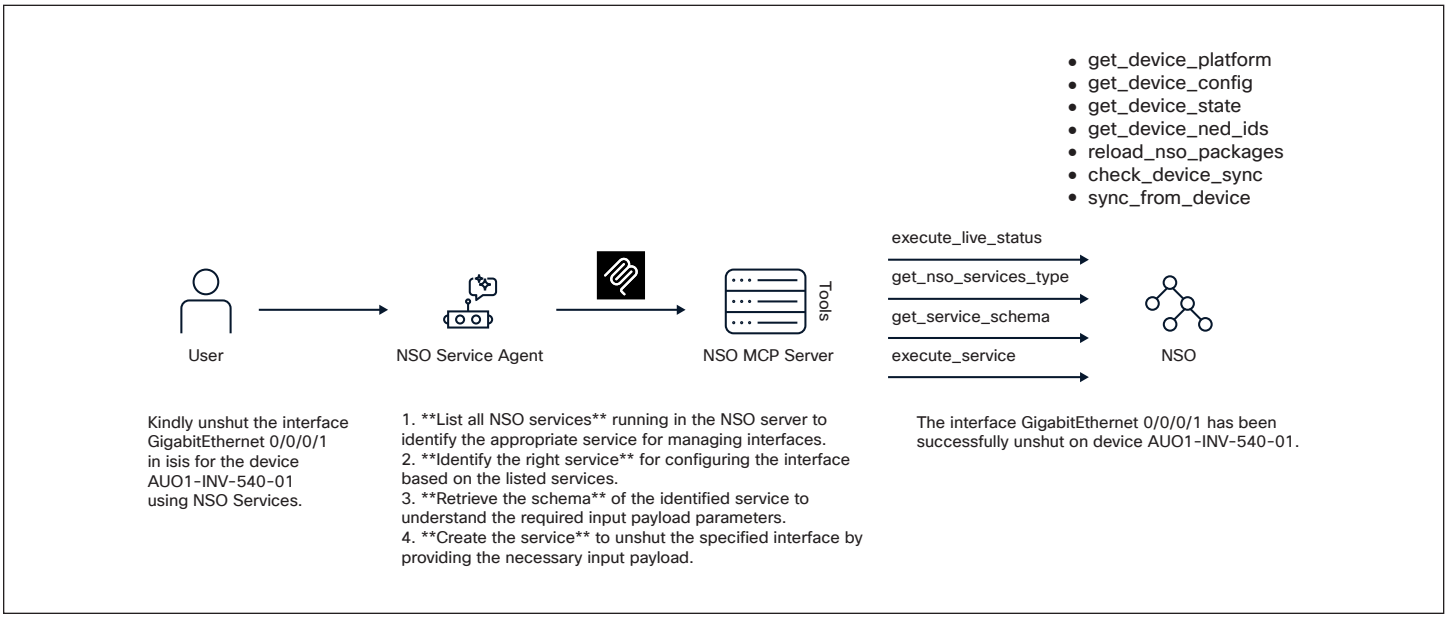

Figure 2. Using MCP services for remediation

Incident Co-Pilot detailed flow

As explained in previous sections, Incident Co-pilot agents automate and enhance network operations using AI-driven workflows. All agents are deployed and managed through the Cisco Crosswork Multi-Agentic AI Framework, as shown in Figure 3. This framework enables the registration, onboarding, and management of agents and supports advanced features such as intelligent routing of user queries to the appropriate agent, integration with data sources, and seamless connection with Cisco Crosswork suite products, as well as providing memory, scalability, and high availability. The Cisco Crosswork Multi-Agentic AI Framework also includes a Software Development Kit (SDK) for creating customized agents to meet specific customer requirements. Once registered and onboarded, agents can access data sources through standard integration.

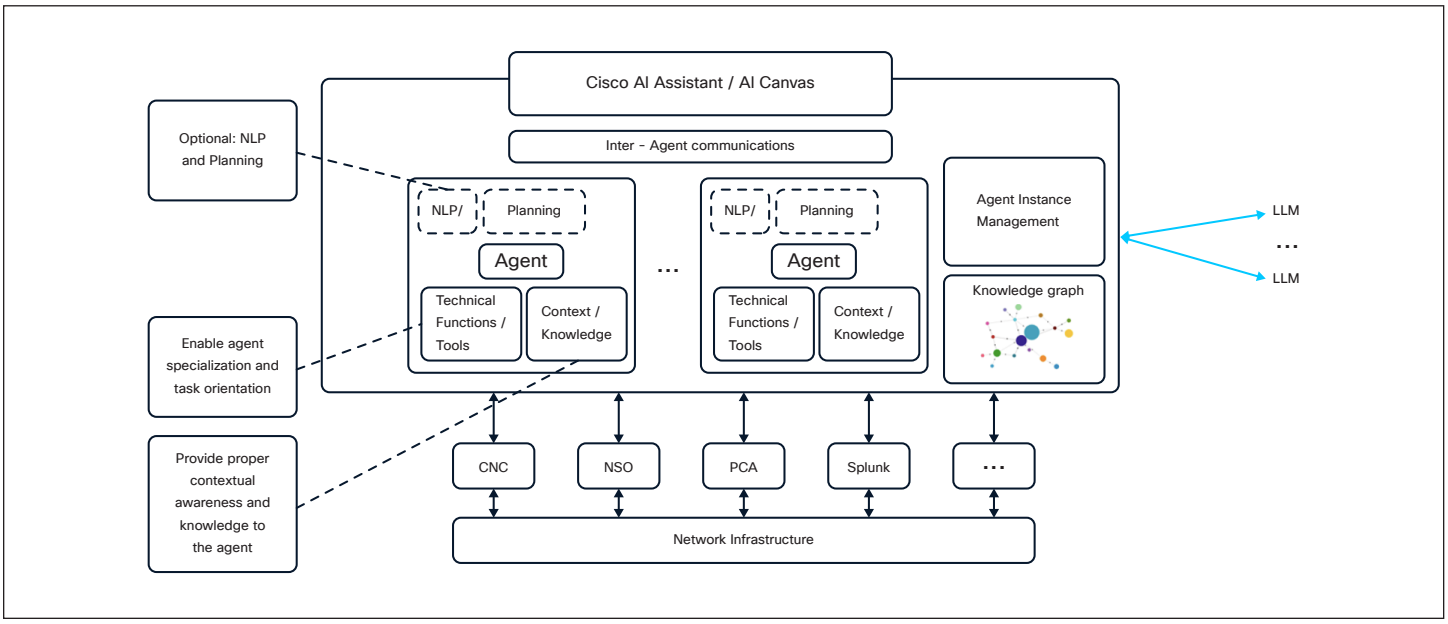

Figure 3. Crosswork Multi-Agentic AI Framework

Let's look at a detailed workflow in which the Incident Co-pilot agents deployed on the Crosswork Multi-Agentic AI Framework are used to troubleshoot and remediate a routing issue, as shown in the Figure 4.

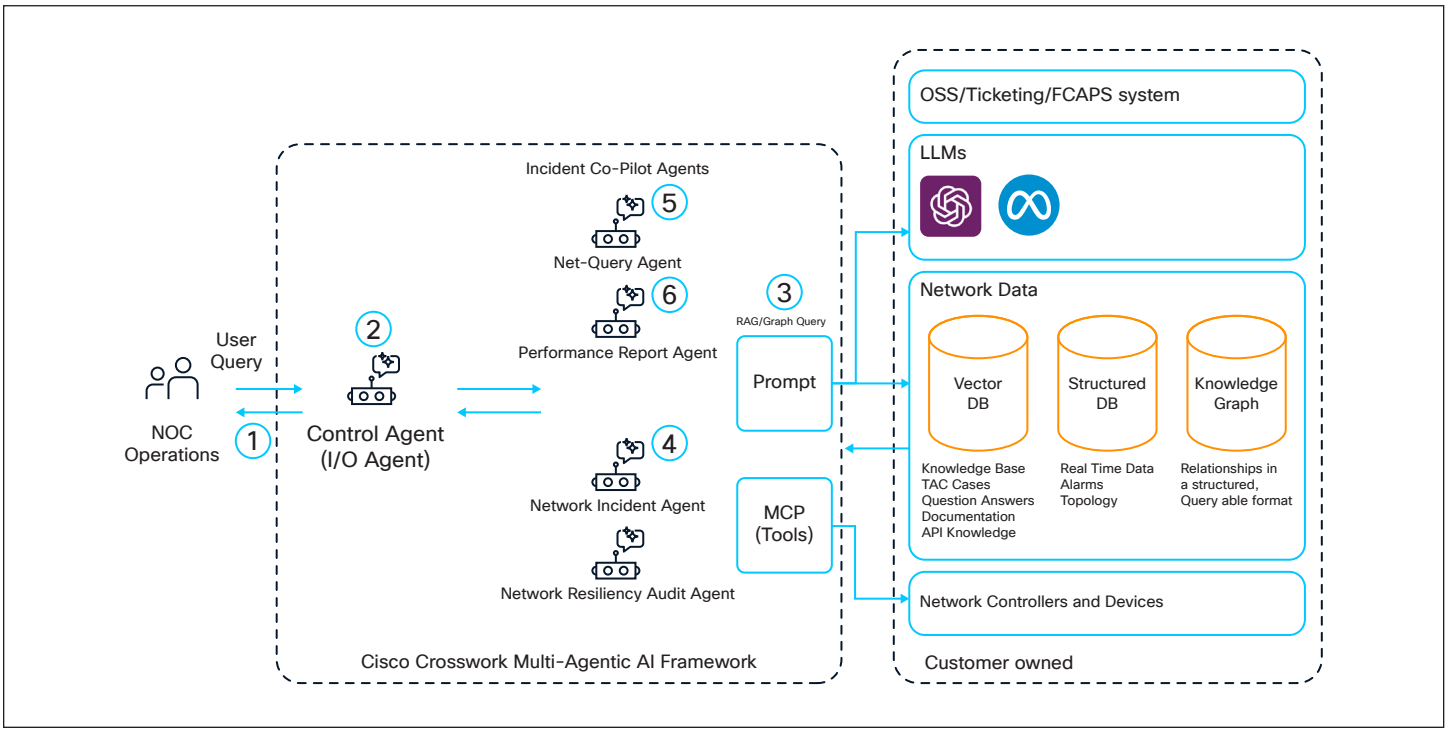

Figure 4. Detailed operational flow of Incident Co-Pilot

1. The Incident Co-Pilot workflow can be triggered either by network events or by a NOC operator's query through a centralized UI, which serves as the main interaction point between users and agents.
2. When a troubleshooting request is made, it is routed via the control agent (I/O agent) to the appropriate troubleshooting agent, which uses RAG databases and action execution tools and integrates with network controllers or devices through MCP servers and APIs.
3. Each agent is equipped to communicate with other agents and network devices and is integrated with LLM services.
4. The troubleshooting agent retrieves the relevant MoP, executes each step, and identifies the root cause; it can also suggest and, with approval, execute remediation steps through the Crosswork NSO MCP server for secure network changes. For devices outside Crosswork NSO, the architecture supports integration with other controllers using REST APIs.
5. After remediation, users can check network health via the net-query agent, which accesses devices through MCP server integrations.
6. The performance report agent can generate detailed network health reports, including interface utilization and traffic analysis.

This architecture supports reusability and customization through modular components such as prompts, retrieval logic, vector database methods, tools, extensions, and custom agents, which makes it flexible and robust.

Once the agent-to-agent capabilities are available on the Crosswork Multi-Agentic AI Framework platform, Incident Co-Pilot will be able to leverage the other available agents.

Agent lifecycle management

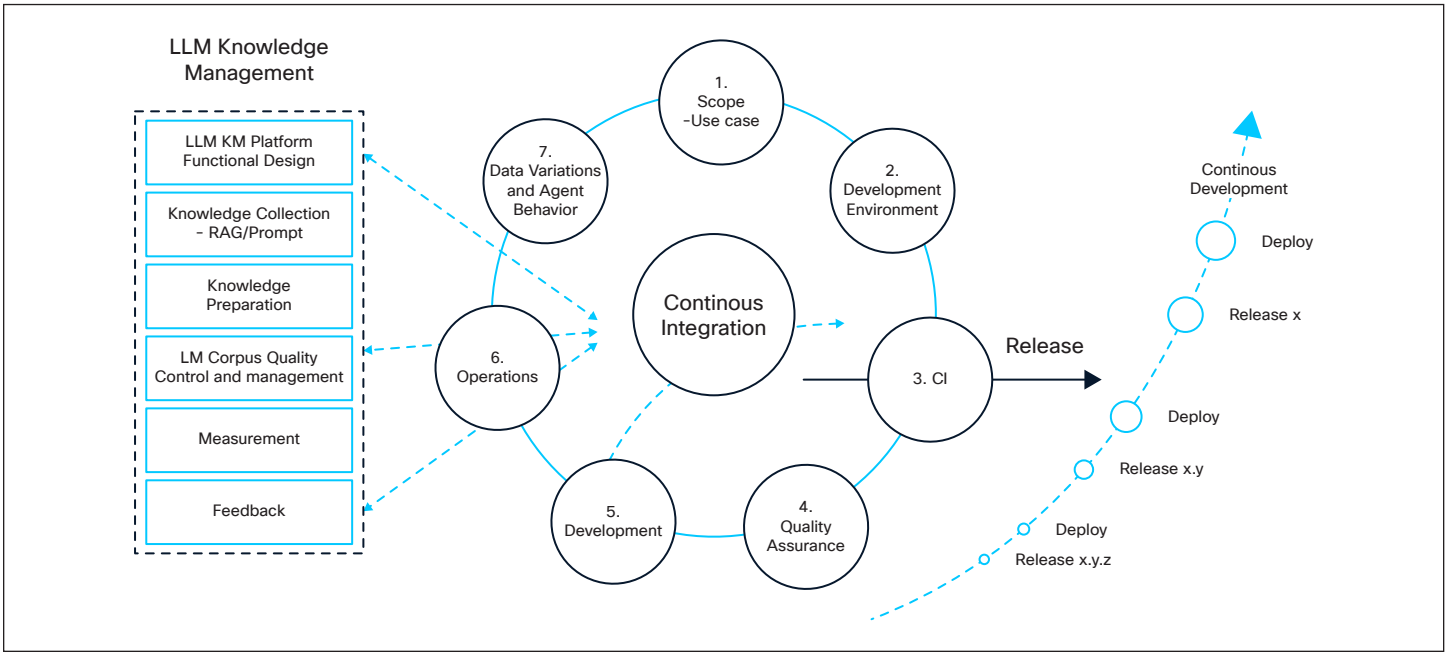

Figure 5. Lifecycle management of AI agents

Every agent plays a critical role in ensuring the overall effectiveness of the multi-agent ecosystem. For the ecosystem to function seamlessly as a whole, it is essential to continuously monitor and validate the performance and accuracy of each agent. Figure 5 shows the different stages of lifecycle management, similar to a Continuous Integration/Continuous Deployment (CI/CD) process in the software development lifecycle.

To achieve an efficient agent, multiple metrics, scoring mechanisms, and benchmarking techniques are applied to assess efficiency, reliability, and adaptability. These help identify optimization opportunities while helping ensure sustained alignment with business and technical objectives. Figure 6 shows a framework with different tools to accomplish various outcomes at different stages of lifecycle management.

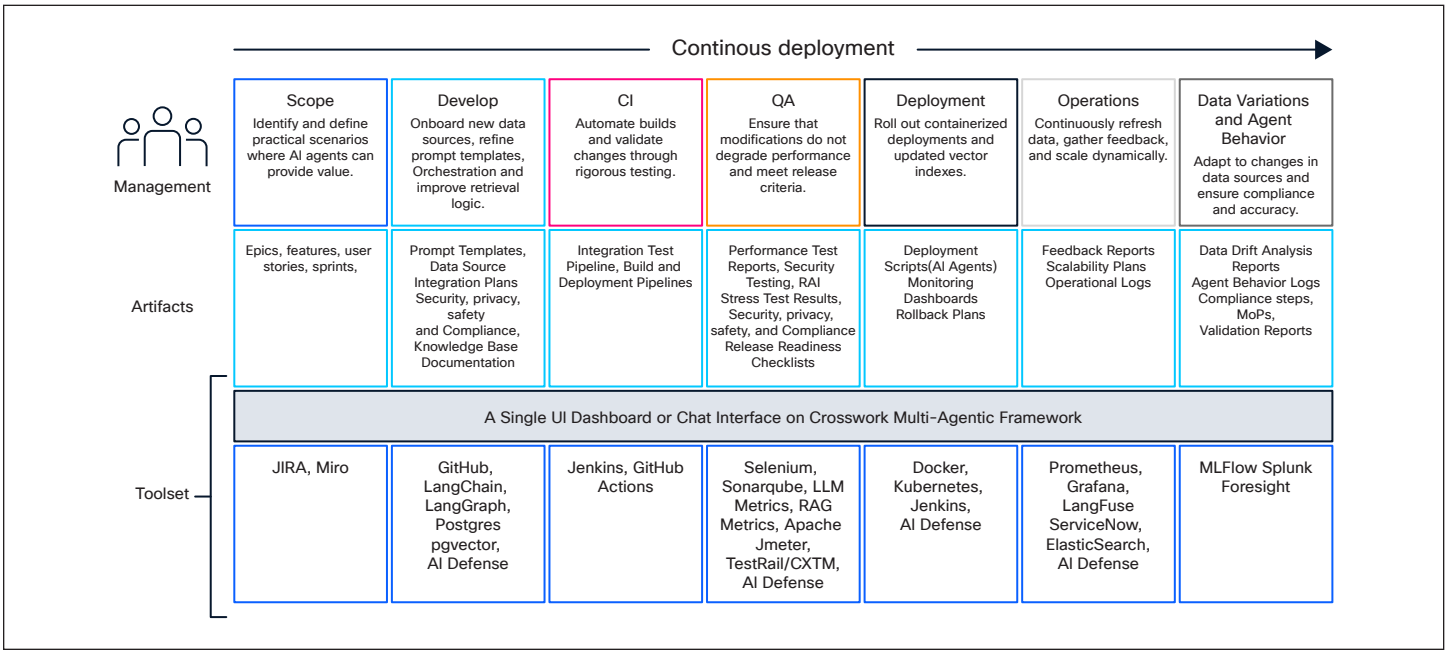

Figure 6. AI agents Ops methodology and components

When monitoring the lifecycle and performance of AI/LLM-based agents, a comprehensive set of metrics and techniques should be considered:

- **Foundational metrics:** Accuracy, precision, recall, and F1 score, supplemented with NLP/LLM evaluation metrics such as ROUGE, METEOR, and BLEU.
- **RAG-specific metrics:** Leveraging Retrieval Augment Generation Assessments (RAGAs) to evaluate the effectiveness of agents using RAG.
- **Resource utilization:** Tracking consumption of memory, CPU, tokens, API loads, LLM load, disk usage, context length, and availability.
- **Success rate and reliability:** Measuring tool invocation accuracy, tool outcomes, final response validity, and failed queries.
- **Logging and traceability:** Maintaining detailed execution traces, error logs, and user activity records for debugging and compliance.
- **User feedback:** Capturing structured and unstructured end-user feedback to continuously refine agent performance and features.

Journey toward autonomous NOCs

The agentic approach, as demonstrated by the TM Forum Incident Co-Pilot-based AI agents and Cisco's Crosswork Multi-Agentic AI platform, underscores the value of collaborative multi-agent systems in automating complex NOC workflows. Each agent is designed with a specialized function, yet they communicate and coordinate seamlessly to analyze data, reason about incidents, and recommend or execute actions, paving the way toward fully autonomous NOCs. The complete Crosswork Network Control Suite is leveraged to achieve closed-loop automation.

Key management considerations

When planning for the implantation of the agentic platform, keep the following in mind.

- AI agents will require a transformation in the NOC, not only in terms of technology, but also in people's skill sets and operational processes. Building additional trust in the system will be essential, along with human oversight, which will play a critical role in ensuring success.
- Agent collaboration: Specialized agents work in unison to deliver comprehensive incident management and enable proactive network optimization. Identify the task that needs to be replaced with agents. This will be an enhanced process that should be more optimal.
- Learning and adaptation: LLM-driven agents leverage RAG with diverse datasets, combining probabilistic and deterministic reasoning to enable intent-based learning and autonomous decision-making. High-quality datasets are required for efficient agents. Make sure the data acquired is from proven legacy and newer systems. You can use data translation techniques to integrate legacy systems. Red teaming will help achieve the desired efficacy of the system.
- Tooling support: Integration with MCP, LangChain, LangGraph, and AI orchestration frameworks supports prompt engineering, knowledge retrieval, and workflow automation. Well-supported and proven tools will help achieve a high level of automation in a scalable way. The testing and DevOps role will be critical to keep the accuracy of the system high.
- Human-AI interaction: Agents augment NOC engineers' expertise with transparent, explainable outputs, fostering trust and easing adoption. Set the expectation of achieving autonomous capabilities in a proven step. The accuracy of the system can be compared with legacy systems, harmonizing the outcomes.
- Scalability and flexibility: Lightweight and modular agent architectures simplify deployment, customization, and scaling to meet evolving operational demands. Make sure the architecture of agents follows the latest standards.
- Additionally, workforce readiness is essential, requiring an emphasis on skill development, new competency requirements, and adoption strategies within the NOC environment.

In summary, transformation of people and processes is just as crucial as the incorporation of technological enablers. Addressing these elements will enhance the credibility of the narrative and underscore change management as a key enabler of the autonomous NOC journey—potentially even more critical than the technology itself.

Conclusion

The evolution of NOC operations toward fully autonomous models is no longer optional—it is a strategic imperative for service providers aiming to enhance agility, reduce operational costs, and deliver superior service quality. The TM Forum Incident Co-Pilot Catalyst project and Cisco's Crosswork Multi-Agentic AI Framework exemplify complementary, agentic approaches that automate both business and technical processes within the NOC.

By harnessing LLM- and ML-powered AI agents deployed on scalable platforms and integrated with advanced tooling, these solutions enable intent-driven, adaptive, and autonomous workflows. Alongside the transformation of people's skill sets and process changes, the focused pursuit of NOCless, white, or dark NOCs through agentic architectures underscores the industry's clear trajectory—toward intelligent, efficient, and self-sustaining network operations.

Authors

Vijay Raghavendran, Distinguished Engineer

Javier Antich, Principal Product Management Engineer

Jiming Shen, Principal Architect

Rajesh Jain, Customer Delivery Architect

Karthik Raja, Customer Delivery Software Architect

Niraj Gadhe, Customer Delivery Software Architect

Reviewers

Prapanch Ramamoorthy, Principal Engineer

Vijay Chundury, Director Customer Delivery

Vijay Kataria, Leader, Customer Delivery

Varsha Virmani, Leader, Customer Delivery

References

- TM Forum
- Autonomous Networks Implementation Guide
- A Holistic View of AI-Driven Network Incident Management
- New-Generation Intelligent Operations: Agentic AI-Driven Transformation