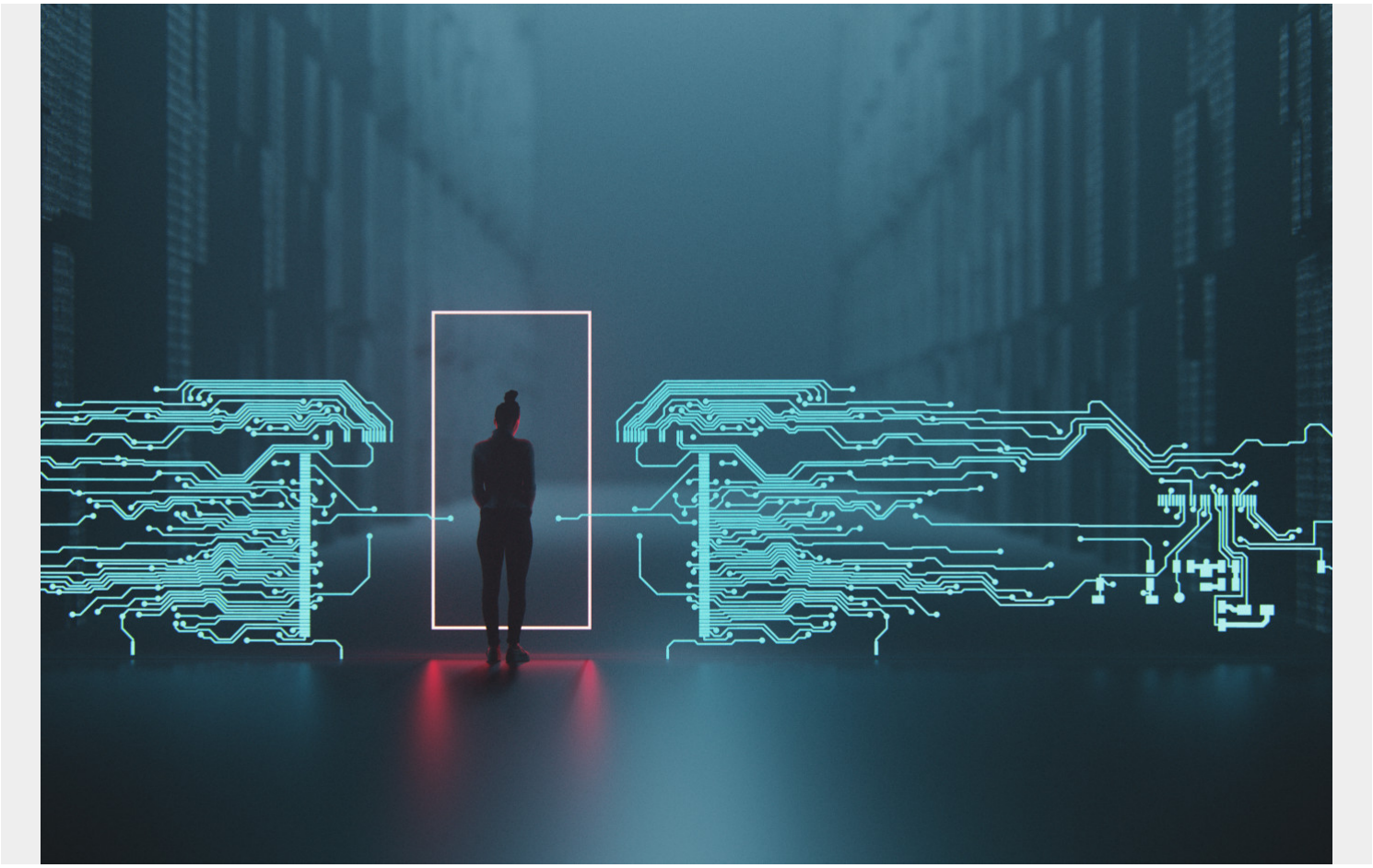
HOW BMC HELIXGPT-POWERED AIOPS CONNECTS OBSERVABILITY SILOS FOR FASTER PROBABLE ROOT CAUSE ISOLATION



In the news last year, a core network change at a large service provider resulted in a major outage that not only impacted end consumers and businesses, but also critical services like 911 and Interac. At home, phone and internet services were down for the workday as millions were without service.

Modern distributed and ephemeral systems are connecting us better than ever before, and the latest ChatGPT phenomenon has opened the possibility for new and mind-blowing innovations. However, at the same time, our dependency on this connected world, along with its nonstop innovations, challenges our ethos with important questions and concerns around privacy, ethics, and security, and challenges our IT teams with outages of often unknown origin.

When it comes to system outages, artificial intelligence for IT operations (AIOps) solutions with the right foundation can help reduce the blame game so the right teams can spend valuable time restoring the impacted services rather than improving their mean time to innocence (MTTI) score. In fact, much of today's innovation around ChatGPT-style algorithms can be used to significantly improve the triage process and user experience.

In the monitoring space, impact analysis for services spanning application to network or cloud to mainframe has known gaps that, if solved, can have a big impact on service availability. Today, these gaps require human intervention and result in never-ending bridge calls where each siloed team responsible for applications, infrastructure, network, and mainframe are in a race to improve their MTTI score. This, unfortunately, also has a direct impact on customer experience and brand quality.

The challenge faced by teams is a layering issue, or "layeritis." Figure 1 below shows the different layers that can contribute to a typical business service like mobile banking or voicemail:

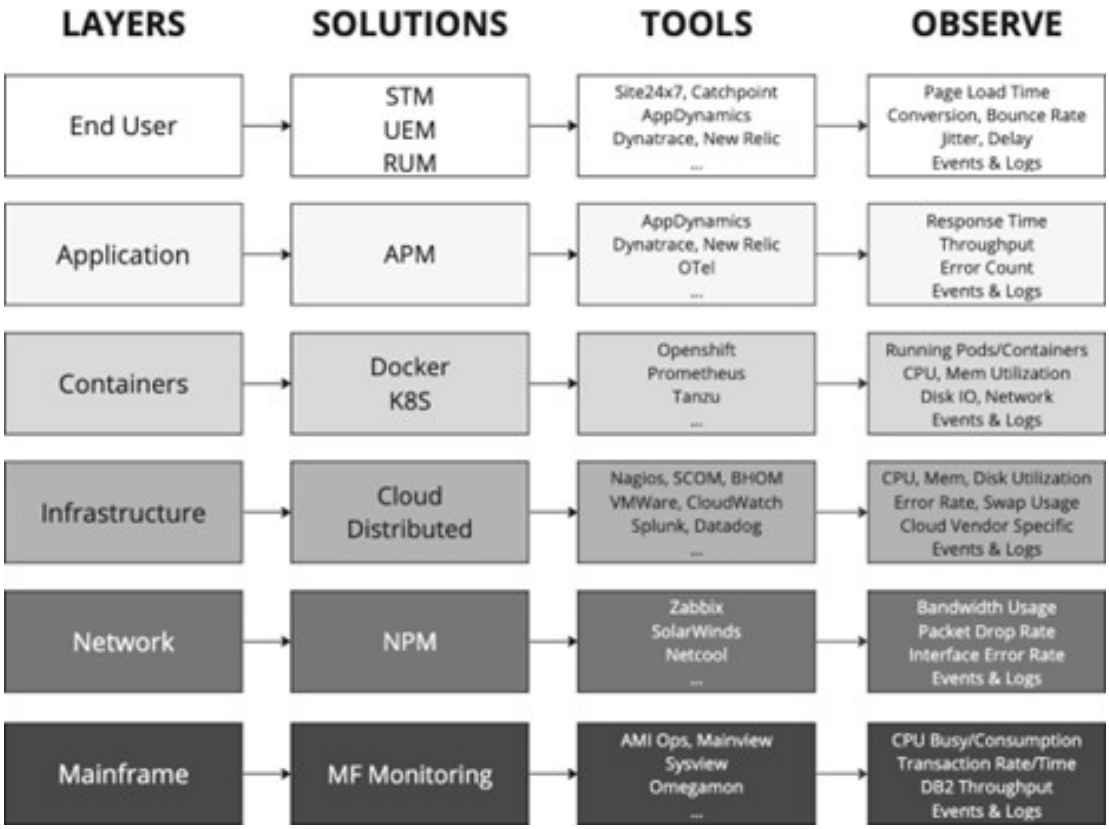


Figure 1: This diagram shows observability silos and the resulting challenge of reconciliation.

For each layer, different kinds of monitoring solutions are used. Each solution, in turn, has its own team and applies different techniques like code injection, polling, or network taps. This wide spectrum of monitoring techniques eventually generates key artifacts like metrics, events, logs, and topology that are unique and useful in the given solution but operate in silos and do not provide an end-to-end impact flow.

Tool spam leads to a noise reduction challenge, which many AIOps tools solve today with algorithmic event noise reduction using proven clustering algorithms. However, in practice, this has not been proven to reveal root cause. The hard problem is root cause isolation across the layers, which requires a connected topology (knowledge graph) that spans the multiple layers and can deterministically reconcile devices and configuration items (CIs) across the different layers.

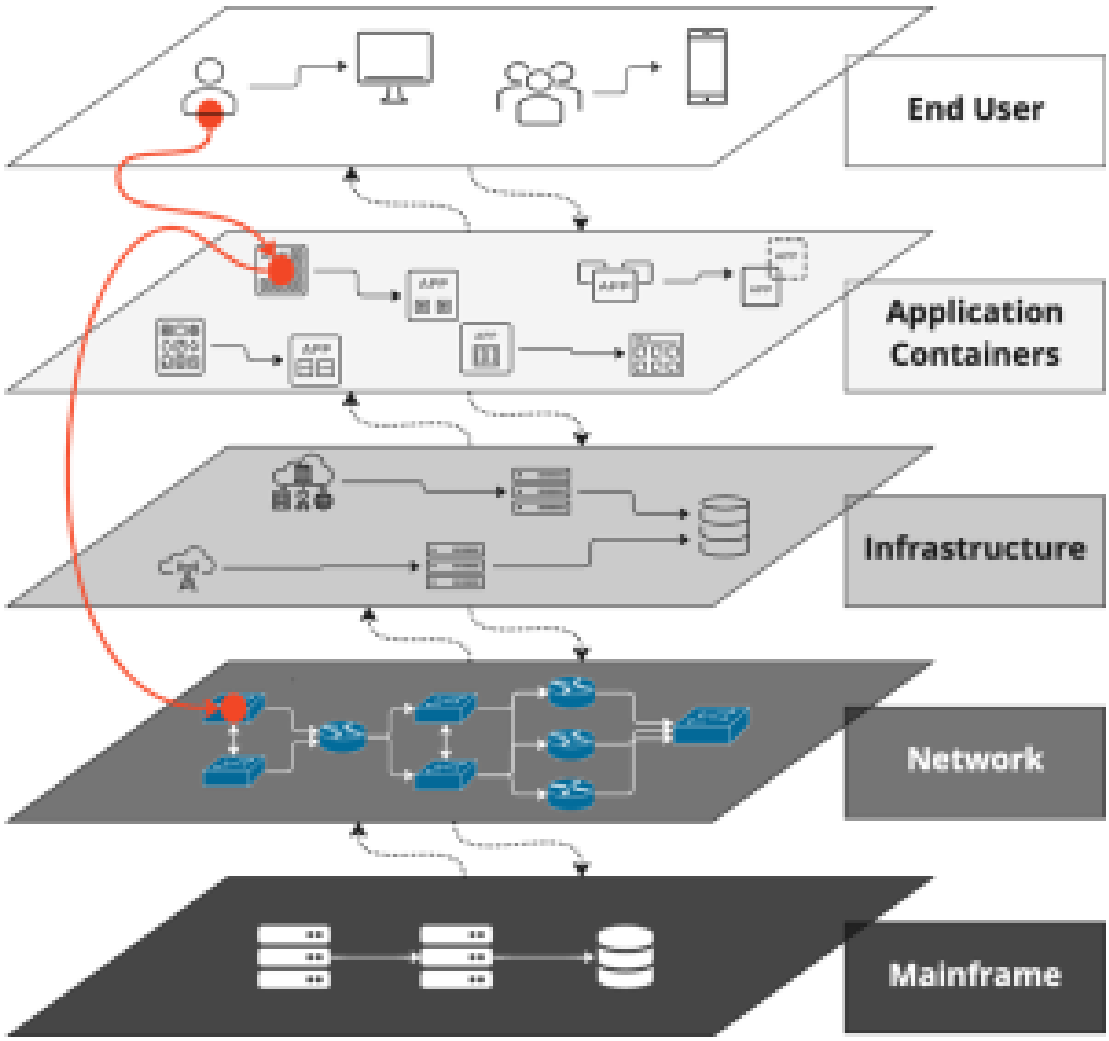


Figure 2: The challenge of root cause isolation across the siloed application, infrastructure, network, and mainframe layers.

Seven steps to cure layeritis:

The solution to an AIOps layeritis challenge requires planning and multiple iterations to get it right. Once steps 1–3 are in a good state, steps 4–7 are left to AI/machine learning (ML) algorithms to decipher the signal from the noise and provide actionable insights. The seven steps are as follows:

1. Data ingestion from monitoring tools representing the different layers to a common data lake that include metrics, events, topology, and logs.
2. Automatic reconciliation across the different layers to establish end-to-end connectivity.
   a. Since end user experience is tied to service health score, include key end-user performance metrics like browser response time or voice quality.

   b. Application topology to underlying virtual and physical infrastructure for cloud, containers, and private data centers (e.g., application performance monitoring (APM) tools may connect to the virtual host, but will not provide visibility to the underlying physical infrastructure used to run the virtual hosts).

   c. Infrastructure connectivity to the underlying virtual and physical network devices like switches, routers, firewalls, and load balancers.

d. Virtual and physical infrastructure connectivity to mainframe services like IBM® Db2®, IBM® MQ®, IBM® IMS™, and IBM® CICS®.

3. Dynamic service modeling to draw boundaries and build business services based on reconciled layers.
4. Clustering algorithm for noise reduction of events from metrics, logs, and events within a service boundary.
5. Page ranking and network centricity algorithms for root cause isolation using the connected topology and historical knowledge graph.
6. Large language model (LLM) and generative AI (GPT) algorithm to build human-readable problem summaries. This helps less technical help desk resources quickly understand the issue.
7. Knowledge graph updated with the causal series of events, a.k.a. a fingerprint, which is compared with historical occurrences to help make informed decisions on root cause, determine the next best action, or take proactive action on issues that could become major incidents.

For algorithms to give positive results with a high level of confidence, good data ingestion is required. Garbage data will always give bad results. For data, organizations rely on proven monitoring tools for the different layers to provide artifacts like topology, metrics, events, and logs. Additionally, with metrics and logs, it's possible to create meaningful events based on anomaly detection and advanced log processing.

Below are three use cases that focus on common issues today's IT teams face, all of which can be resolved using AIOps in a single consolidated view to identify the root cause and automate the next best action. Note in each use case that 1) the generative AI-based problem summary reduces event noise and makes it easy for the help desk to understand the issue, 2) the clustering of all events, event deduplication, and single incident creation help with noise reduction, and 3) root cause isolation eliminates the blame game and improves mean time to repair (MTTR).

The reconciliation engine for BMC Helix AIOps capabilities is key to automatically reconciling and building a connected topology from application to network and cloud to mainframe. The reconciled topology is based on CIs and their relationships from monitoring and discovery tools.

Use case 1: Application-Only Issue (No Infrastructure or Network Impact)

Isolate root cause to the application (application issue where infrastructure and application are not impacted), as shown in Figure 3.

In this example, the root cause was isolated to the application software components, monitored by an APM tool.

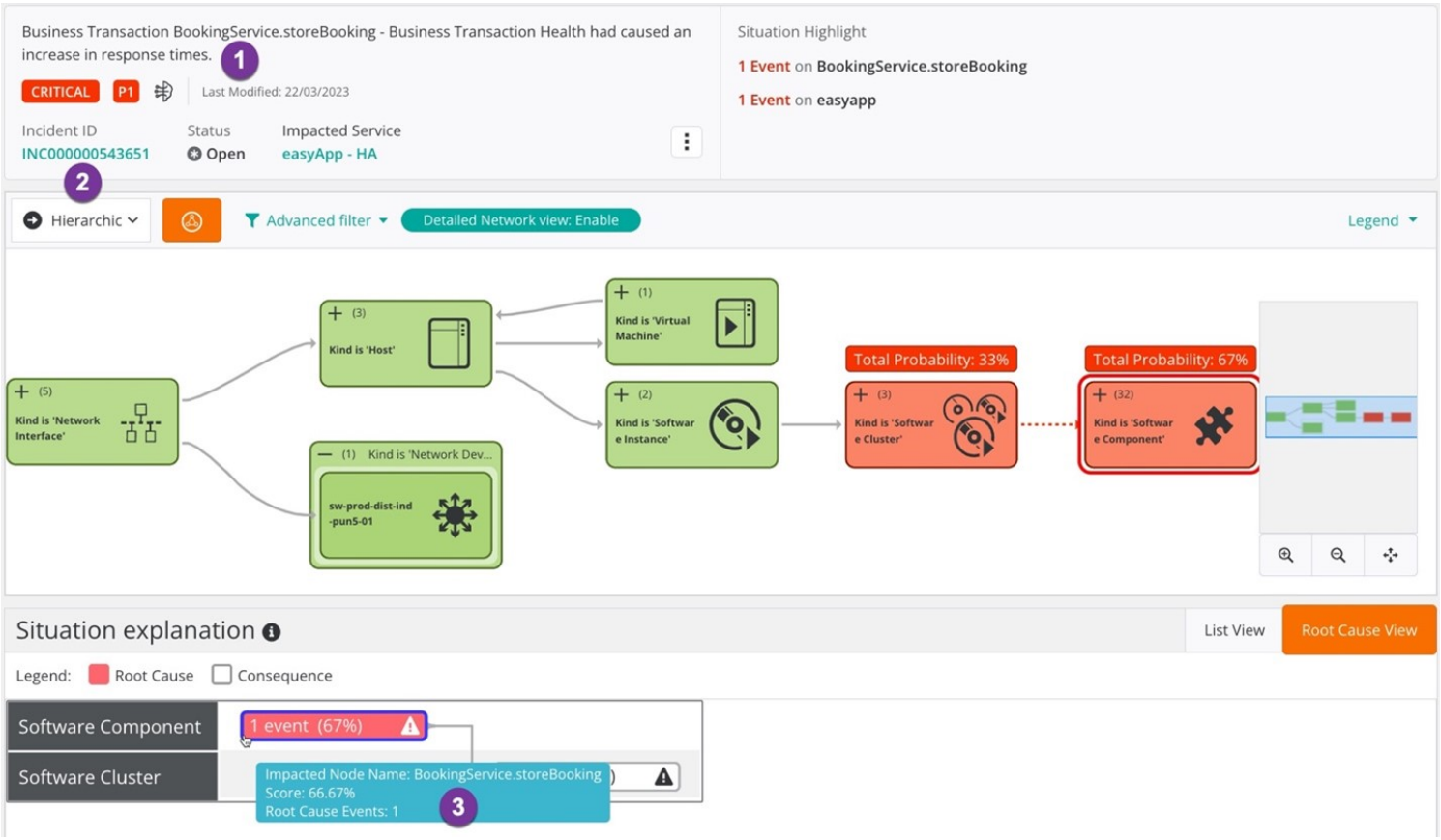


Figure 3: An application-only issue with no infrastructure or network impact.

In this scenario, AIOps will summarize the issues in Figure 3 as follows:

1. **Generative AI problem summary:** *"Business Transaction BookingService.storeBooking - Business Transaction Health had caused an increase in response times."*
2. **Noise reduction:** Clusters and deduplicates events for the impacted service based on time, text, and topology. Opens a single ITSM incident for the problem cluster (*"INC000000543651"*).
3. **Root cause isolation:** Root cause is the "*BookingService.storeBooking*" service. The situation explanation/fingerprint provides evidence on how the events started with the application software components and eventually impacted the software cluster.

Use case 2: Network issue impacting host and application

Isolate root cause from application to network (network issue where infrastructure and application are impacted, but not at fault), as shown in Figure 4.

In this example, the root cause was isolated to a network device, monitored by a network monitoring tool.

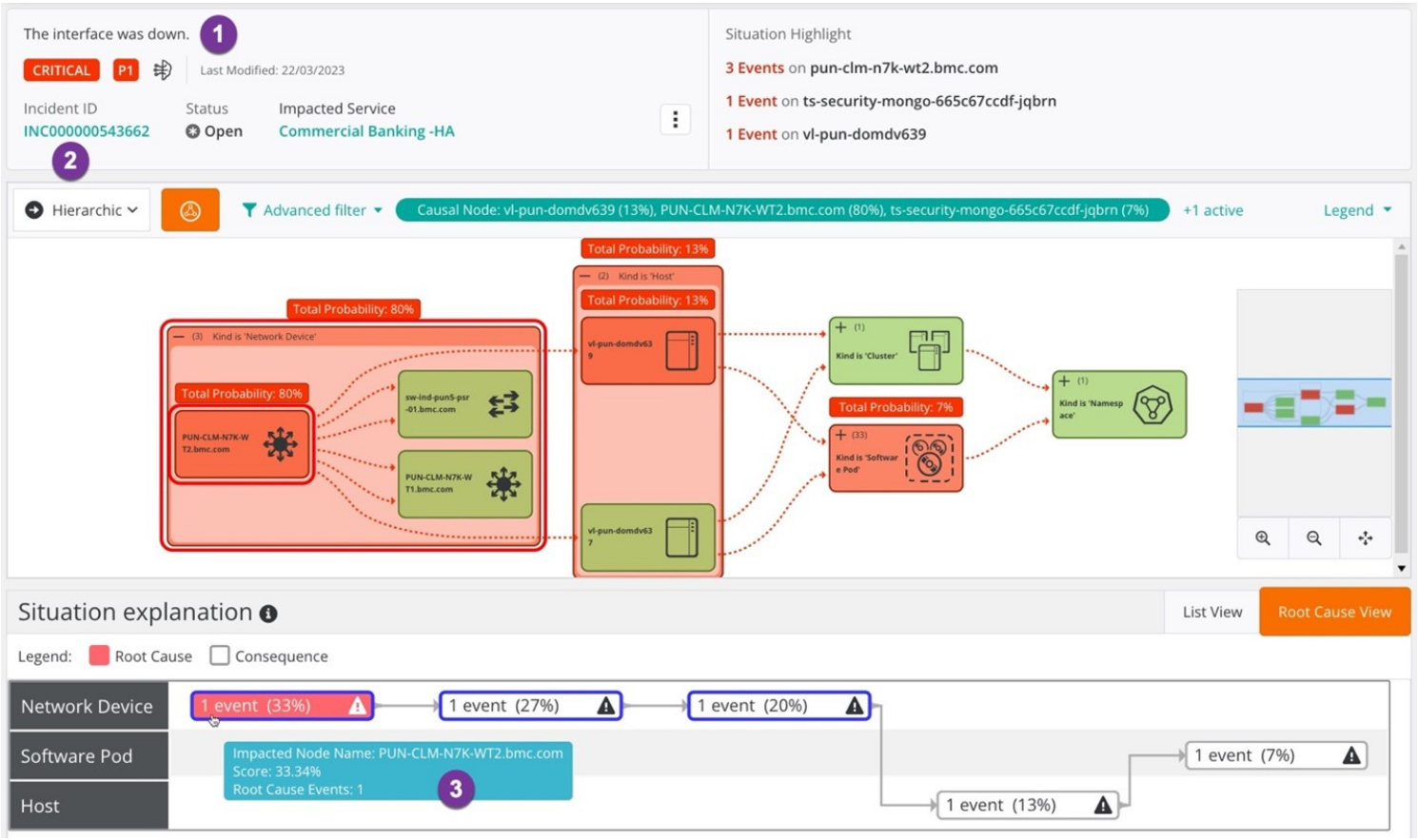


Figure 4: A network-only issue impacting the host and application.

In this scenario, AIOps will summarize the issues in Figure 4 as follows:

1. **Generative AI problem summary:** *"The Interface was down."*
2. **Noise reduction:** Clusters and deduplicates events for the impacted service based on time, text, and topology. Opens a single ITSM incident for the problem cluster (*"INC000000543662"*).
3. **Root cause isolation:** Root cause is the *"pun-clm-n7k-wt2.bmc.com"* network device. The situation explanation/fingerprint provides evidence on how the events started with the network device and eventually impacted the host and software pod.

Use case 3: Mainframe database issue impacting distributed applications

Isolate root cause from cloud to mainframe (mainframe database issue where distributed application impacted), as shown in Figure 5.

In this example, the root cause was isolated to the mainframe Db2 database, monitored by a mainframe monitoring tool.

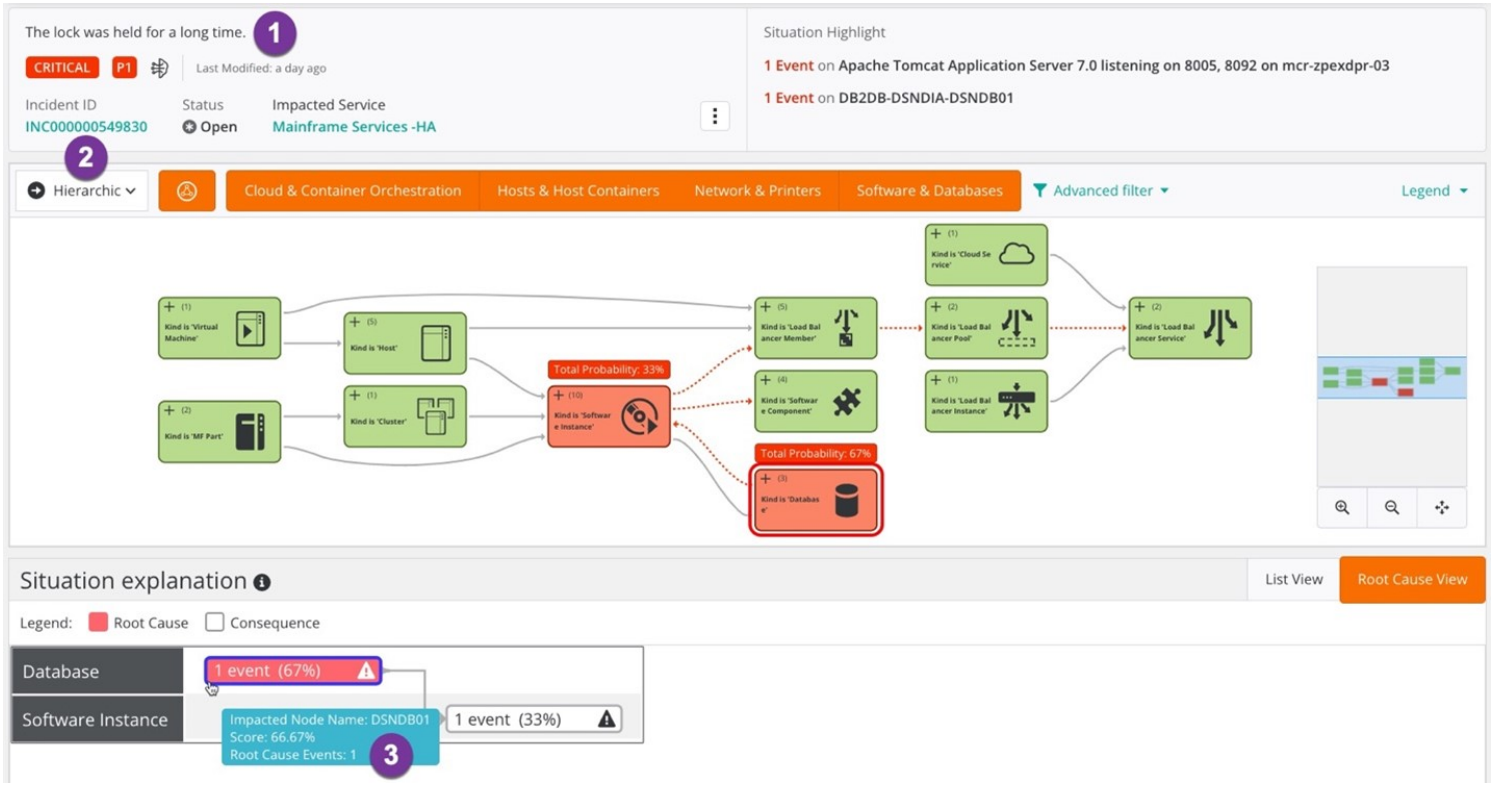


Figure 5: Mainframe database issue impacting upstream applications.

In this scenario, AIOps will summarize the issues in Figure 5 as follows:

1. **Generative problem summary:** *"The lock was held for a long time."*
2. **Noise reduction:** Clusters and deduplicates events for the impacted service based on time, text, and topology. Opens a single ITSM incident for the problem cluster (*"000000549830"*).
3. **Root cause isolation:** The database lock held by the *"DB2DB-DSNDIA-DIA1"* database, which is impacting dependent application services. The situation explanation/fingerprint provides evidence on how the events started with a database issue and impacted a software instance.

With a defined service model and reconciled topology, AI/ML algorithms are used to derive insights and root cause with a high level of confidence.

In each use case above, AIOps removes the need for time-intensive investigation and guesswork so your team can see and respond to issues before they affect the business and instead focus on higher-value projects.

With today's complex challenges across modern, distributed architectures, AIOps solutions can provide visibility and generate proactive insights across the entire application structure, from end user to cloud to data center to mainframe.