Tuskira™

White Paper

The SIEM Tax

Why centralized security operations create more alerts, higher costs, and longer delays, and how federated, agentic SecOps changes the economics of detection.

Contents

Short on time? Ask your AI assistant:

Copy the prompt below into your AI assistant to surface the strongest insights, unresolved questions, and practical implications.

Prompt:

Read this document carefully. Then do the following:

Identify the 3–5 non-obvious insights. Focus on what can be inferred from the argument, not just what the author states directly. Skip anything already presented as a key point.

Find the tensions, contradictions, or unresolved trade-offs. Where does the argument conflict with itself, with conventional wisdom, or with how security teams actually operate?

Extract the "so what." If a smart, busy executive could take away only one actionable implication, what should it be and why?

Name what is missing. What important question does the document raise but does not fully answer? What would you want to know next before acting on it?

Preface

For two decades, security operations have been organized around a single architectural assumption: centralize all telemetry into one platform, write detection rules against the aggregated data, and staff analysts to triage what comes out. This model was designed for environments with fewer tools, slower adversaries, and simpler infrastructure. It is no longer that world.

The centralization tax, the compounding cost of ingestion, storage, detection engineering, and analyst headcount required to operate a centralized SOC, has become the dominant budget driver in enterprise security. Each new tool added to the stack increases cost at every level without proportional improvement in outcomes. Detection latency is structural, not operational. Alert volume has outpaced human capacity. Analyst burnout drives turnover above 30 percent annually. The industry has spent a decade optimizing the queue. Outcomes have not improved proportionally.

This white paper examines why the centralized model fails, what replaces it, and how the economics of security operations fundamentally change when detection moves to the source and correlation happens across the stack in real time.

Five Structural Failures of the Centralized SOC

1. Centralization creates detection latency, not just cost. The log transport chain, endpoint to collector to pipeline to SIEM to rule match, introduces a structural delay that sophisticated attackers exploit. This is a physics problem, not a vendor problem.
2. The SOC operates as a queue, but attackers operate through paths. A coordinated attack crosses identity, endpoint, cloud, and network as a single progression. The centralized SOC sees it as four separate alerts in four separate queues. No single queue sees the whole path.
3. Detection engineering has become a maintenance bottleneck. Teams spend 60 percent of their time maintaining existing rules across heterogeneous schemas rather than building new coverage. Every new tool added deepens the debt.
4. Alert volume has surpassed human processing capacity. The average SOC handles 11,000+ alerts per day. Analysts spend 80 percent of their time on triage. The operational model relies on humans performing tasks that machines should handle.
5. Cost scales linearly with coverage, but effectiveness does not. Adding data sources increases ingestion, storage, rule counts, and increases analyst workload. The centralization tax compounds at every level while detection coverage improves marginally.

Five Required Shifts for Modern Security Operations

1. Detection must move to the source. Security tools already detect. The EDR detects endpoint threats. The identity provider detects authentication anomalies. Federated detection leverages native tool intelligence rather than duplicating it through centralized log analysis.
2. Correlation must happen across tools in real time, without data movement. The federation layer queries tools through native APIs, stitching context across 150+ integrations in seconds rather than waiting for log transport and human pivoting across consoles.
3. AI agents must operate as autonomous analysts, not assistants. L1 agents triage and enrich. L2 agents investigate and validate. Response agents execute containment. Human analysts shift from queue processors to oversight managers.
4. Shared context must replace siloed detection. The Security Context Graph maps relationships between identities, assets, vulnerabilities, and threat behaviors, transforming detection from pattern-matching to behavioral understanding.
5. The cost curve must invert. In a federated model, adding tools improves intelligence without increasing ingestion or storage cost. The marginal cost of coverage decreases while the marginal value increases.

Chapter I

The Librarian and the Fire

It was 2:47 AM on a Tuesday when Priya noticed the pattern. Staring at three monitors, running on her third energy drink, she scrolled through a queue of 847 security alerts. Sixty-three were flagged as critical. She had been at this for six hours. She knew, with the certainty that comes from five years in a security operations center, that roughly 840 of these alerts would turn out to be noise. The problem was that she had no idea which seven were real.

Priya wasn't bad at her job. She was excellent at it. She had been one of the best detection engineers at her previous company, the kind of person who could look at a SIEM query and see the attack before it fully formed. She took this SOC manager role because she wanted to build something better, to create a place where security felt less like whack-a-mole and more like actual investigation. Instead, she had become the world's most expensive spam filter.

She didn't start with 847 alerts a night. When she joined the company six months ago, it was 200. Clean. Manageable. Then they added CrowdStrike for endpoint detection. The alert count doubled. Then Okta for identity, AWS GuardDuty for cloud, Proofpoint for email, and Network GuardDuty for network traffic. Each tool was the right decision on its own. A company that wants to protect itself needs to know what's happening on its endpoints, in its identity system, and in its cloud. But collectively, they had created something nobody wanted: a system that generated more information than any human team could process.

Here is where the problem got interesting, in the way problems become interesting right before they destroy a company. All these alerts flowed into the SIEM, Splunk, in this case, where they were supposed to be correlated and interpreted. The idea was straightforward. Centralize all the information. Put one smart librarian in charge. Let that librarian spot the patterns that individual librarians in individual rooms would miss.

But imagine you actually have ten librarians, each running a room in a massive library.

The first librarian knows every book in the Reference section. She can spot a misplaced volume, an unusual patron, a fire starting in a wastebasket. The second librarian runs the Children's wing. The third manages History. Each one is an expert in their domain. Now imagine someone, say a consultant, probably, or an executive who has read too many LinkedIn articles, decides the better approach is to carry every single book from every room into one central library. Take the fifteen million volumes and stack them in one warehouse. Hire one librarian to watch all of them. That central librarian has more books, but less understanding. She can see everything, but she knows nothing. The context that made each room's librarian effective was proximity and familiarity, not volume. When you centralize, you lose the former and gain the latter.

That's the modern security operations center. It took the distributed intelligence of specialized tools like the EDR that understands endpoints, the identity provider that understands authentication patterns, the cloud security platform, and centralized it all into one system optimized for volume, not understanding. The SIEM became the god of all information. Everything flowed in. Somehow, from the chaos, detection was supposed to happen.

Priya understood the problem. She just couldn't do anything about it. The entire industry was built on the assumption that centralization was the answer. The SIEM is where the magic happens. The SIEM is where you write detection rules. The SIEM is where alerts are born. Every other tool is merely a data source feeding into the true center of gravity. This wasn't a decision anyone at her company had made. This was just how the world worked.

It should have been obvious by now. The architecture Priya was trapped inside was designed in a world that no longer existed. And the most dangerous thing about it wasn't that it failed. It was that it failed in a way that made it look like it was working.

Chapter II

The Centralization Tax Nobody Talks About

Three weeks later, Marcus Harwood, CFO, sat in his office reading something he wished he didn't have to read. It was the annual security tool renewal bill. It was enormous. It was $2.3 million larger than the previous year, and Marcus, who controlled the budget, was supposed to explain why.

Marcus didn't understand cybersecurity. He didn't need to. What he understood was money, and from the bill, he understood that his security team kept asking for more. They had Splunk, which was expensive. They had CrowdStrike, which was expensive. They had Okta, GuardDuty, Proofpoint, and 17 other tools, each one with a price tag that made his eye twitch. His question, the reasonable question any CFO would ask, was simple: why do we need all of this when we already have everything?

When the security director, Tom, tried to explain, Marcus learned something interesting. The bill wasn't just for the tools themselves. Splunk charged by the gigabyte. Every time security added a new data source, the Splunk bill went up. Every new tool that fed into the SIEM increased the ingestion cost. And because the SIEM couldn't process all the data fast enough with their current infrastructure, they had to pay for additional compute, additional storage, and additional network. The bill had exponentials.

But it got worse. The tools themselves generated data, which had to be stored, and someone had to analyze it. Tom's team had hired three new analysts in the past year, expensive people who demanded competitive salaries. Yet alert volume kept climbing. The three new analysts were drowning in the same way Priya was drowning. So either they hired more analysts (more budget), or they paid for smarter tools to filter alerts (more budget), or they paid for more computing infrastructure to speed up processing (more budget).

Marcus realized what was happening. He was caught in a paradox. Better security coverage meant higher costs at every level. Not just the new tools themselves, but the infrastructure to centralize their data, the storage to keep it, the compute to process it,

“He wasn’t paying for security. He was paying for the architecture’s inability to think.”

the retention policies that compliance demanded, and the skilled analysts to triage what it all meant. When Tom asked permission to add a new cloud security tool, Marcus’s instinct was to say no, because adding detection coverage had become synonymous with raising the entire budget.

This is what economists would call the “centralization tax.” It compounds silently. Most organizations discover it the same way Marcus did, when the renewal comes in significantly higher than last year, and nobody can point to what actually improved in security. The individual tools got cheaper. The infrastructure to centralize them got more expensive. The people tasked with managing the chaos grew in number and became more frustrated. The whole stack became a machine for generating expenses.

Consider the numbers in this industry. The average enterprise SIEM cost grows 25 to 40 percent annually as data sources expand. Detection engineering teams, the people who write the rules that make SIEMs useful, spend roughly 60 percent of their time maintaining rules that already exist instead of writing new ones. SOC analyst turnover exceeds 30 percent per year, which, in a tight labor market, means you spend a fortune recruiting and training people who quit in frustration. The average SOC is now processing more than 11,000 alerts per day, which is like asking someone to read a novel while riding a motorcycle.

Think of centralization as a highway system in which every road in the country leads to a single toll booth. The first year is fine. By year three, you have ten thousand cars a day trying to squeeze through a single gate. So what do you do? You hire more toll collectors. You build faster gates. You add lanes approaching the booth. The entire highway system’s economy becomes optimized around the toll booth rather than around getting people where they want to go. Meanwhile, the actual job of the toll booth, letting cars through, never improves, because the fundamental design is the bottleneck.

Marcus had a realization sitting in his office, reading a bill that shouldn’t have been so large. He wasn’t paying for security. He was paying for the architecture’s inability to think.

Chapter III

The Queue Problem

Priya started timing herself. From the moment an alert appeared in her queue to the moment she could render a verdict, suspicious or not, threat or noise, escalate or close. She bought a notebook and wrote down the time for each alert. Over two weeks, she collected data on 673 alerts. The average time from alert to decision was 47 minutes.

“Attackers don’t operate through queues. They operate through paths.”

Not because she was slow. Because the work required opening four different consoles, running three manual queries, cross-referencing data that refused to match up because the tools used different naming conventions for the same concepts, and then building a timeline by hand because none of the tools spoke to each other natively. An alert from Okta meant opening the Okta console. An endpoint alert meant opening the EDR. A cloud alert meant opening AWS. You were constantly bouncing between windows, copying identifiers, pasting them into other systems, waiting for queries to finish, trying to remember what you had learned five minutes ago when you were looking at a different tool.

Here is what nobody talks about when they build centralized SOCs: the system, at its heart, is a queue. Like a deli counter where customers take a number. The whole operational philosophy is optimized around how fast you process the queue. Mean Time to Detect and Mean Time to Respond are metrics that are the religion of modern security. They’re also measuring the wrong thing.

Because attackers don’t operate through queues. They operate through paths. Consider a real attack. Someone compromises a credential, maybe it comes from a third-party breach, maybe it comes from phishing. That’s an identity event. It appears to the identity provider as a failed authentication attempt. The attacker gets the right password and logs in successfully. Now they’re an endpoint. The EDR sees unusual behavior, lateral movement, reconnaissance. The attacker escalates privileges and moves to cloud resources. Now they’re a cloud event. Finally, they exfiltrate data. Now they’re a network event.

The attack is a coordinated path across four different tools’ territories. But because each tool feeds into the SIEM separately, the SOC sees it as four separate alerts in four separate queues. The identity team detects a failed login attempt from an unfamiliar

“You can’t optimize your way out of a fundamental architectural flaw.”

geographic location. Nothing unusual. The endpoint team sees some baseline reconnaissance. Baseline. The cloud team sees a read operation on S3 buckets. The user has permissions. The network team sees a data transfer that stays within the company VPN. Nothing suspicious. No single team sees the whole path. Each sees one chapter of the story and concludes that nothing interesting is happening. The thief walks through four rooms of a museum, each one guarded by an excellent security officer who watches their room perfectly. But they are watching their room, not watching for the thief. The thief walks out the back with a Vermeer

The museum’s response is predictable. It has been the response for two decades. Hire faster guards. Buy better radios. Process the visitor sighting queue more efficiently. Reduce mean time to response. Put guards in the hallways. Create an incident response playbook. Meanwhile, nobody asks the obvious question: what if the guards could see beyond their own room?

The industry has spent a decade making the queue faster. Time-to-alert has improved. Detection latency has dropped. But the percentage of attacks that reach their objective hasn’t dropped proportionally. That gap, between faster processing and unchanged outcomes, is the evidence that the model itself is broken. You can’t optimize your way out of a fundamental architectural flaw. You can only build faster, more elaborate versions of the broken thing.

Chapter IV

What “Agentic” Actually Means...
and What It Doesn’t

The word “agentic” in 2026 has achieved the status of every buzzword before it. The cloud. Multicloud. The blockchain. Every vendor had slapped it on something. A dashboard that auto-prioritizes alerts based on a machine-learning model, agentic. A chatbot that could understand natural language questions about security, agentic. A script that executed remediation steps automatically, agentic. If every product is agentic, the word means nothing. So let’s define it through what it actually changes.

The first era was the playbook era. SOAR (Security Orchestration Automation and Response) promised that you could write a series of steps: if this condition is true, then do this, then do that. Like a recipe. Execute reconnaissance, then check if the verdict is suspicious, then respond. The problem was obvious in retrospect. Most attacks don’t follow recipes. Attackers adapt. They change their tools, timing, and tactics because they are smart and motivated. By the time you wrote a playbook, the attack had already evolved past the playbook. SOAR was useful for routine tasks. But routine threat response is a tiny fraction of what a SOC actually does.

Then came the AI-assistant era. Machine learning models that helped human analysts think. These products prioritized alerts by learning which alerts had historically led to confirmed threats. They suggested the next investigation step. They summarized findings from multiple tools into a single brief. They were genuinely useful. Think of them as a very good research assistant who pulls relevant files and highlights important passages, but never writes the brief. The human analyst still does the thinking. The machine just retrieves information and suggests directions.

Agentic is different in kind. An agentic system has the authority and capability to reason across the full security context without human guidance. An AI agent that triages and enriches an alert in seconds, not by following a predefined script, but by actually investigating. An agent that queries multiple security tools, synthesizes the responses, builds a timeline, assesses whether the behavior represents a genuine threat, and renders

a verdict with confidence and supporting evidence. A response agent that can execute remediation within defined boundaries.

The human analyst doesn't disappear, but the role changes. The human becomes a plant manager instead of a line worker. Overseeing the system. Handling exceptions. Making the strategic decisions that only humans can make, about what risk the organization is willing to accept, about whether the evidence justifies a particular response, about what the attack tells you about your adversary and their intentions. The work becomes more interesting.

But agentic operations are only as good as the architecture underneath them. An AI agent that can only see one tool's data is just a faster version of the old model. For agents to reason across the full security context, they need access to it. Which means the centralized model, where data must be ingested, stored, and processed before it can be analyzed, is a bottleneck for AI just as much as it was for humans. You can't build genuine agentic SecOps on top of a centralized architecture. The architecture has to change first.

Chapter V

Stop Centralizing. Start Federating.

What if the librarians stayed in their rooms?

The central library idea failed because it lost something essential: the librarian's expertise. But what if you could keep the expertise and add coordination? Instead of carrying every book to one place, the librarians stayed in their rooms. The Reference librarian stayed in Reference. The History librarian stayed in History. Each one maintained their specialty, their understanding, their mastery of the material. But they gained the ability to talk to each other. When the Reference librarian spotted something unusual, she could instantly reach across to History and ask, **"Are you seeing anything related to this pattern?"** When the History librarian discovered a gap, she could ask Science if they had information that filled it.

That's federated detection. And here is what should make it obvious: every security tool in an enterprise already detects. That's literally what they are designed to do. CrowdStrike detects endpoint anomalies. It does this extremely well. Okta detects suspicious authentication. It does this extremely well. AWS GuardDuty detects cloud threats. It does this extremely well. The problem was never that these tools couldn't detect. The problem was that they couldn't talk to each other meaningfully, in real time, without first funneling everything through a centralized bottleneck.

Federated detection connects to these tools through their native APIs. Not by ingesting their logs, logs are an after-the-fact representation of what happened, but by speaking their language. When CrowdStrike fires a detection, the federation layer doesn't wait for that event to appear as a log line in a SIEM. It queries CrowdStrike directly. Is this endpoint showing signs of infection? Is there lateral movement? Has the attacker escalated privileges? Then it reaches across to Okta. Has this user been authenticating normally? Then AWS. Has this account been accessing resources it normally accesses? Then the network. Is there data leaving the network? In real time. Within seconds.

The correlation that Priya spent 47 minutes assembling by hand happens in seconds. The timeline that required jumping between four consoles is automatically stitched together.

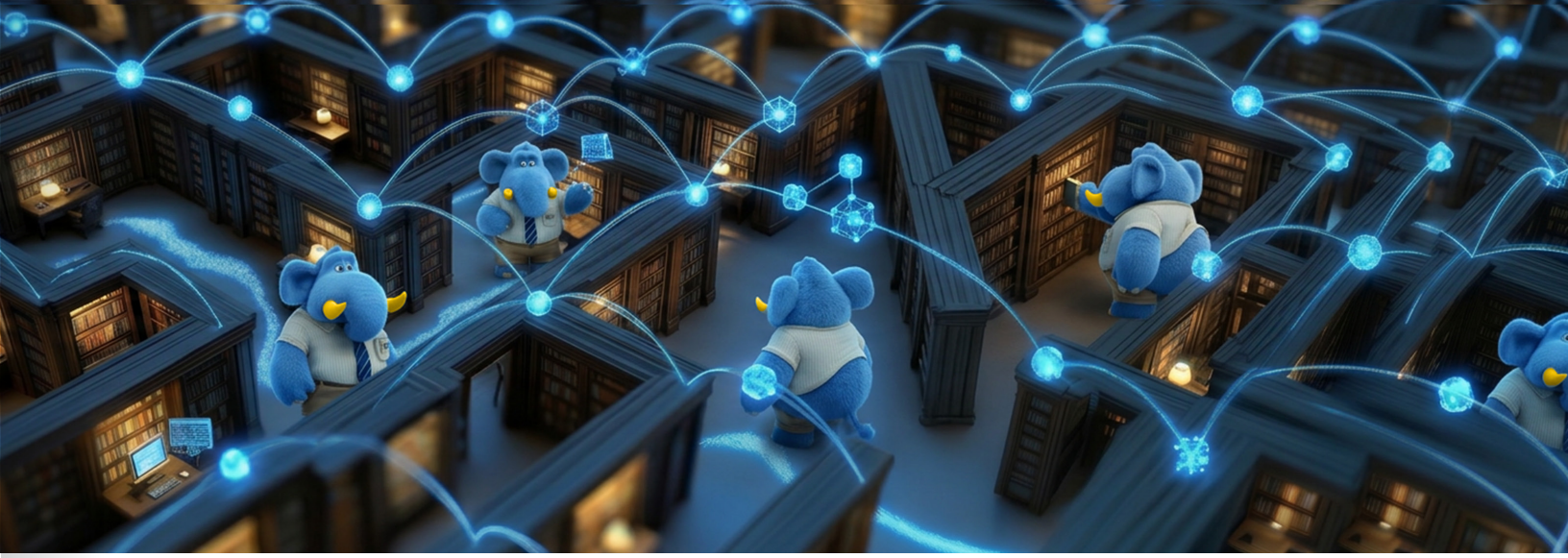
"The intelligence traveled. The data stayed put."

The context that required domain expertise and pattern recognition is computed.

But here's where it gets interesting, and where Marcus the CFO should pay close attention: no data moved. No ingestion bill. No storage costs for duplicate data. The intelligence traveled. The data stayed put. CrowdStrike's data stays in CrowdStrike. Okta's stays in Okta. AWS's stays in AWS. The federation layer only carries questions and answers.

The Federated Detection Engine sits in the middle. It understands what each tool can do and how to ask it meaningful questions. CrowdStrike speaks the language of endpoints and behavioral analytics. Okta speaks the language of authentication and identity risk. AWS speaks the language of cloud resource access and configuration. The engine normalizes across these differences. It generates detection logic that works across heterogeneous environments. It maintains that logic automatically, which is the part that should make security leaders sit up straight, because detection engineering, the act of writing rules that actually catch attacks, has been the bottleneck, consuming 60 percent of security teams' time. The engine eliminates most of that work.

And the scale of integration isn't academic. One hundred fifty or more security tools connected through native integrations. Not log parsers that flatten rich, structured data into generic fields. Native integrations that preserve each tool's full detection capability. The vendor ecosystem is large. The federation layer speaks their languages.



Chapter VI

The Mesh

Federation solves the coordination problem. But federation alone gives you ten librarians with walkie-talkies. Useful. But you are still limited by what each librarian knows about their own room. What is missing is shared understanding. The bigger picture. The map.

This is where the Security Context Graph comes in. Think of it as the shared brain across the federation. The librarians have walkie-talkies. Now they also have a map of the entire building: who is in it, where they are going, what they are carrying, which rooms connect to which, where the exits are, and which routes an intruder would take.

The Security Context Graph maps relationships between identities, assets, vulnerabilities, and threat behaviors in real time. It tracks who has permissions to what. It understands the data flows. It learns normal patterns. It becomes the connective tissue that transforms individual detections into understanding.

Walk through an example. A suspicious Azure AD sign-in from an unfamiliar geography might, by itself, be a password typo. A user traveling, logging in from their hotel. But the Security Context Graph knows more. It knows the IP address also probed the VPN yesterday. It knows the credentials match a set exposed in a recent breach, published three hours ago. It knows the targeted account has elevated permissions to cloud storage containing customer data. It knows this account normally authenticates from five specific IP ranges and never logs in at 2 AM. Suddenly, that suspicious sign-in isn't a password typo. It's the first step of a coordinated attack. Not because any single tool flagged it. Because the mesh connected what each tool saw into a coherent narrative.

The consequence is profound. Detection shifts from pattern-matching to behavioral understanding. The system doesn't just catch known attack signatures. It understands how attacks behave, the paths they take, the gaps they exploit, the timing they require, and the sequence of actions that real humans would never execute but compromised accounts do. The same attack doesn't land twice because the system understood the behavior and adapted across every connected tool.

"Four minutes versus four hours. The attacker accessed nothing."

Chapter VII

A Tale of Two SOCs

Return to Priya one more time. The same attack. A credential-stuffing campaign hitting Azure AD using credentials from a third-party breach. The attack is real, and the paths are the same, but the difference is the architecture.

In Priya's SOC (centralized, queue-based, dependent on human triage), the attack begins at 2:17 AM on Wednesday. By 2:32 AM, logs have reached the SIEM, and an alert fires at position 63 in the queue. Priya doesn't see it for forty minutes. By 3:12 AM, she's reading the alert. Suspicious sign-in from an unfamiliar location. She opens four different tools manually, copies data between them, checks threat intelligence with ambiguous results, and determines access scope by manually checking permissions. By 4:45 AM, she has enough to escalate. L2 validates by 5:30 AM. Account disabled. By then, the attacker had accessed two S3 buckets and downloaded customer contact data for three thousand people. Four hours from initial attack to response. Data breach complete.

In the alternative SOC, federated and agentic. Same attack. 2:17 AM. Azure AD native detection fires immediately. The L1 agent receives it within seconds and queries the Security Context Graph. IP associated with a credential-stuffing campaign. Credentials match recent breach. Account has elevated cloud permissions. This violates all baselines. Within 12 seconds, the attack is correlated across four intelligence domains. L2 agent maps blast radius. Which S3 buckets? No access yet. Response agent revokes the session, disables the account, and blocks the IP range. 2:19 AM. Human analyst reviews the investigation chain and approves. 2:21 AM.

The attacker accessed nothing.

Four minutes versus four hours.

The contrast speaks for itself.

Marcus reached a conclusion: he wasn't overspending on security. He was overspending on an architecture that couldn't think.

Chapter VIII

The Economics of Thinking Differently

Marcus discovered something interesting when he asked his security director to explain the cost structure of a federated model. The centralized model has a linear cost curve. Every new tool, every new data source, every new detection rule costs more at every level. More ingestion. More storage. More networking. More analysts. The cost of coverage scales linearly, but the effectiveness doesn't. More alerts don't mean better outcomes. It's the curve of an industry throwing resources at a problem instead of solving it.

The federated model inverts this curve. Adding a new tool doesn't increase ingestion or storage costs. The data stays at the source. Adding a tool increases the mesh's intelligence. More context. Better correlation. Higher-confidence detections. The marginal cost of coverage decreases while the marginal value increases. This is the curve of every technology shift that replaced centralized infrastructure with distributed intelligence.

KEY METRICS

80%+ reduction in SIEM costs

70x faster threat response

50%+ improvement in detection coverage

80% of analyst triage time reclaimed

The numbers tell the story. SIEM-related costs decrease by 80 percent or more. Mean time to response improves by 70x. Detection coverage improves by 50 percent. SOC teams reclaim 80 percent of the time previously spent on triage. Priya goes from processing 847 alerts a night to overseeing an intelligent system that escalates only those that require human judgment.

Chapter IX

What This Means for You

This isn't a rip-and-replace argument. The SIEM doesn't disappear overnight. It becomes a consumer of enriched, correlated findings rather than the engine that produces them. Its role gets smaller and more focused. Which, paradoxically, makes it more valuable.

Integration breadth matters significantly. Ask how many of your current security tools are natively supported. Partial integration delivers partial value. The system's intelligence is limited by the tools it can see. If you integrate 40 percent of your tools, you get benefits for 40 percent of your environment.

Organizational readiness matters more than technical readiness. The shift from "analyst processes queue" to "analyst oversees intelligent system" changes job descriptions and team structure. L1 analysts now review agent verdicts and hunt for novel threats. That's a better job. But it's a different job. L1 analysts need retraining, not replacement. L2 and L3 analysts find their roles elevated, handling more strategic work, improving the system, and dealing with sophisticated adversaries instead of noise.

"She still gets 847 alerts a night. The difference is, she doesn't see them."

Transparency is non-negotiable. Any AI agent making security decisions must produce an auditable reasoning chain. The evaluation question that cuts through vendor noise: show me the audit trail of an AI agent investigating a real alert. Include every data source consulted, every reasoning step, every correlation found, every action recommended. If the vendor can't produce that, the product isn't agentic.

Return to Priya one last time. It's six months after the shift. She still gets 847 alerts a night. The difference is, she doesn't see them. The alerts arrive, the agents investigate, and the agents render verdicts with confidence and supporting evidence. Priya reviews the verdicts. She overrides maybe 2 percent of cases where the agent missed context or the risk calculus was wrong. She spends three hours a week hunting for novel threats, the work she was originally hired to do. Her team is smaller but happier. Nobody has quit in four months. For the first time in years, she isn't exhausted.

The twenty-year-old architecture that centralized everything into one place created a system that could see everything but understand nothing, because it wasn't optimized for comprehension; it was optimized for volume. The future belongs to architectures that distribute intelligence to where the data lives and connect that intelligence through shared context. That future is agentic SecOps, and the organizations that get there first will spend less on security infrastructure and will actually be more secure.

See Agentic SecOps in Action

Request a personalized demo tailored to your security environment.

tuskira.ai/request-a-demo

Tuskira's Agentic SecOps platform connects to 150+ security tools, deploys AI analysts that triage, investigate, and respond to threats in real time, and eliminates the centralization bottleneck that has defined and limited security operations for two decades.

www.tuskira.ai
contact@tuskira.ai

Tuskira™
Full Stack Agentic SecOps