

AI-native, sovereign MDR.

New **Processes** for the AI-Driven SOC

Five shifts — and the toolkit to act on them — for security teams building an AI-driven SOC.

AI-NATIVE • SOVEREIGN-BY-DESIGN • EU-HEADQUARTERED



SOVERA SECURITY

sovera.security

New Processes for the AI-Driven SOC

Five Shifts — and the Toolkit to Act on Them — for Security Teams Building an AI-Driven SOC

Published by Sovera Security ApS, Copenhagen, Denmark.

© 2026 Sovera Security. All rights reserved.

These five shifts were first published individually on soverasecurity.com in 2026, and are collected here in a single edition for readers who want the full arc in one sitting.

No part of this publication may be reproduced or distributed without written permission, except for brief quotations in critical articles or reviews.

Learn more or request an assessment at sovera.security.

INTRODUCTION

The security operations centre has been running at a deficit for a decade. The average organisation now generates roughly 4,484 alerts a day and about half are never investigated at all; 40% of alerts go untouched, and 61% of SOC teams admit having ignored one that turned out to be critical. Seventy-one per cent of analysts report burnout. None of that is a detection failure. It is the arithmetic of an operating model built on one assumption: that human attention is the scarce resource, so volume must become a queue, the queue must become a priority order, and everything below the line must be closed and left closed.

AI removes that assumption without, by itself, removing any of the practices built on top of it. A model bolted onto a workflow designed for human analysts saves time; it does not change what the SOC treats as scarce, what it treats as final, or where it puts authority. The harder and more useful question is what has to change in the process, the architecture, and the accountability model once the operator can reason.

This book answers that question in five shifts, in the order they depend on each other. **Guidance** replaces playbooks: work gets defined by outcome, evidence, boundaries, escalation and record rather than by a fixed sequence. **Closed cases stay alive**: deprioritisation becomes provisional, and new evidence is allowed to change the significance of an old decision. **AI capacity is provisioned, not rationed**: a multi-tier pipeline keeps routine investigation running continuously instead of recreating analyst scarcity in tokens. **Governance moves outside the model**: the critical constraints are enforced by the platform, not hoped for in a prompt. And **humans move up the accountability stack**: from signing off individual verdicts to owning the processes, framework, model standards and strategy that produce them.

Each shift is argued first, then reduced to the decisions a director actually has to make. The **Toolkit** at the back turns those decisions into work: five worksheets, checklists and protocols — one per shift — designed to be filled in against your own environment rather than read. Use them in order if you are starting from scratch, or individually if you already know which part of the operating model is holding you back. The **Reference** section that follows condenses the whole argument into an executive summary, a 90-day roadmap, a maturity scorecard with benchmarks, and a self-audit timeline.

Nothing here has been rewritten to sound more finished than it was at the time. These are field notes from building the thing, not a retrospective.

— Sovera Security

The AI-native SOC is an operating-model shift.

Bolting a model onto a workflow built for human analysts can save time. It does not, by itself, change the way the SOC works.

The deeper change is where work is defined, where evidence is retained, where AI capacity is provisioned, where authority is enforced, and where human accountability sits. Taken together, the five shifts describe a different SOC topology.

The five shifts in this book are one continuous argument — not five disconnected ideas.

01	Guidance replaces playbooks Procedure → intent	Define outcomes, evidence, boundaries, escalation and record-keeping — not every step.
02	Closed cases stay alive Closure → reassessment	Keep history available to new evidence; let it change the significance of old decisions.
03	AI capacity is provisioned Analyst scarcity → machine capacity	Use AI capacity continuously and reserve expensive reasoning for the work that needs it.
04	Governance moves outside the model Prompt → platform	Enforce the security boundary independently of what the model remembers to do.
05	Humans move up the stack Human-in-the-loop → human accountability	Own the system that produces the verdicts, not every operational verdict.

CONTENTS

THE FIVE SHIFTS • ONE OPERATING MODEL, FIVE CHANGES

01	Guidance replaces playbooks	7
	<i>Playbooks vs. Guidance: How Work Gets Defined for an AI-Native SOC</i>	
02	Closed cases are not dead signal	13
	<i>Your Garbage Is Our Gold: Finding the Hidden Signal in the New AI Era</i>	
03	AI capacity should be provisioned, not rationed	20
	<i>Don't Fall Back Into the Same Loop in the AI SOC</i>	
04	Governance moves outside the model	27
	<i>Principles for Building an AI-Driven MDR</i>	
05	Humans move up the accountability stack	36
	<i>The Human Accountability Stack</i>	

THE TOOLKIT • PUT EACH SHIFT TO WORK

01	The Guidance-Writing Framework	44
02	The Deprioritized-Incident Revisit Protocol	46
03	The Multi-Tier AI Pipeline Worksheet	48
04	The Agent Governance Self-Audit	49
05	The Human Accountability Stack Mapper	51

REFERENCE • THE ARGUMENT, CONDENSED

01	The AI-Native SOC — executive summary	53
02	The Five Shifts at a glance	54
03	From Theory to Practice — worked example	55
04	The 90-Day AI-Native SOC Roadmap	56
05	AI-SOC Maturity Scorecard & benchmarks	57
06	The Self-Audit Implementation Timeline	58
07	Where to Start — the checklist	59

END MATTER

—	Source notes	60
—	About Sovera Security	61

Each of the three sections opens on a dark divider page.

The old SOC was designed around human scarcity.

Analyst attention is finite. Traditional workflows therefore turn volume into queues, queues into prioritization, and prioritization into permanent closure.

An AI-native SOC changes the economics of the operating model. It can examine more evidence, revisit more history, and perform more routine work — but only if the surrounding process is redesigned for it.

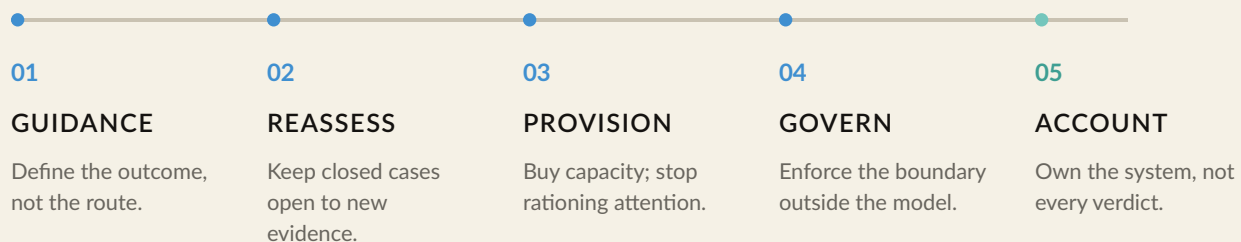


The objective is not to automate the old SOC faster. It is to change what the SOC treats as scarce, what it treats as final, and where it puts authority.

So the question is not what AI can do. It is what the SOC can stop assuming.

Every practice in the old operating model — the queue, the threshold, the permanent close, the rationed analyst hour — was a rational answer to a single constraint. That constraint has moved. The practices built on top of it have not.

What follows are the five places where the operating model has to change, in the order they depend on each other. Each shift is argued in full, then reduced to the decisions a director actually has to make. The Toolkit at the back turns those decisions into worksheets you can run against your own environment.



The five shifts begin on the next page. Each one opens on a dark divider, is argued in full, and closes with the decisions it forces.

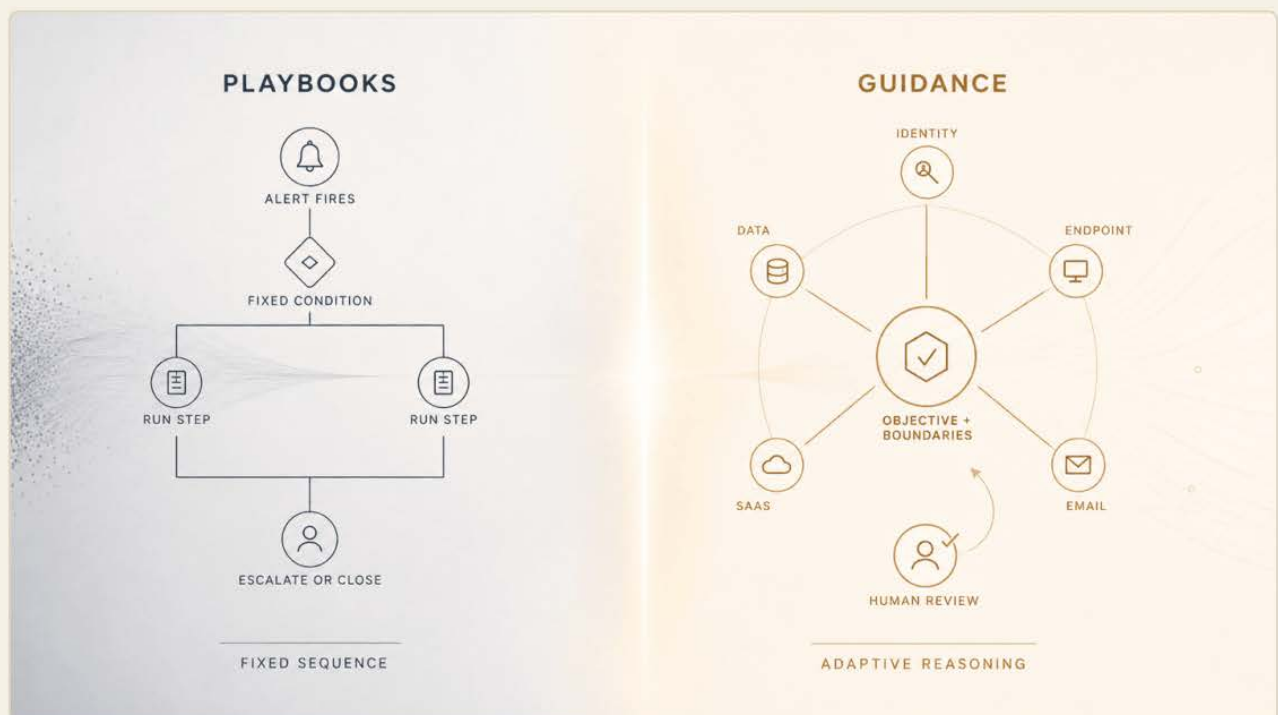
Guidance replaces playbooks.

The question is no longer how to write a better sequence. It is how to define work for a system that can reason.

Playbooks vs. Guidance: How Work Gets Defined for an AI-Native SOC

Security automation was built around playbooks for good reason: analysts forget steps under pressure, and traditional tools could only execute what they were explicitly told. So organizations documented every action and escalation path as a structured sequence, i.e. if condition A, check B; if evidence crosses threshold C, do D. That reduced variation and became the foundation for SOAR and most early SOC automation.

An AI-native SOC faces a different problem. LLMs hold deep technical knowledge and can reason through situations no playbook author anticipated. Forcing them through rigid runbooks anyway just turns them into an expensive version of traditional automation. The question isn't how to write a better playbook but how to define work for a system that can reason. The answer is simply **guidance**.



Why Playbooks Were Necessary

Playbooks solved two problems: human inconsistency, since analysts handle the same alert differently depending on experience, and machine inflexibility, since automation couldn't interpret

intent and needed every branch spelled out. A phishing playbook might tell the system to extract URLs, check domain reputation, and quarantine the email if certain indicators appear. This is only useful when the system can't judge what matters on its own. An LLM can instead understand the goal and adapt as findings emerge, though a rigid playbook still limits it to logic reflecting only what its authors knew at the time.

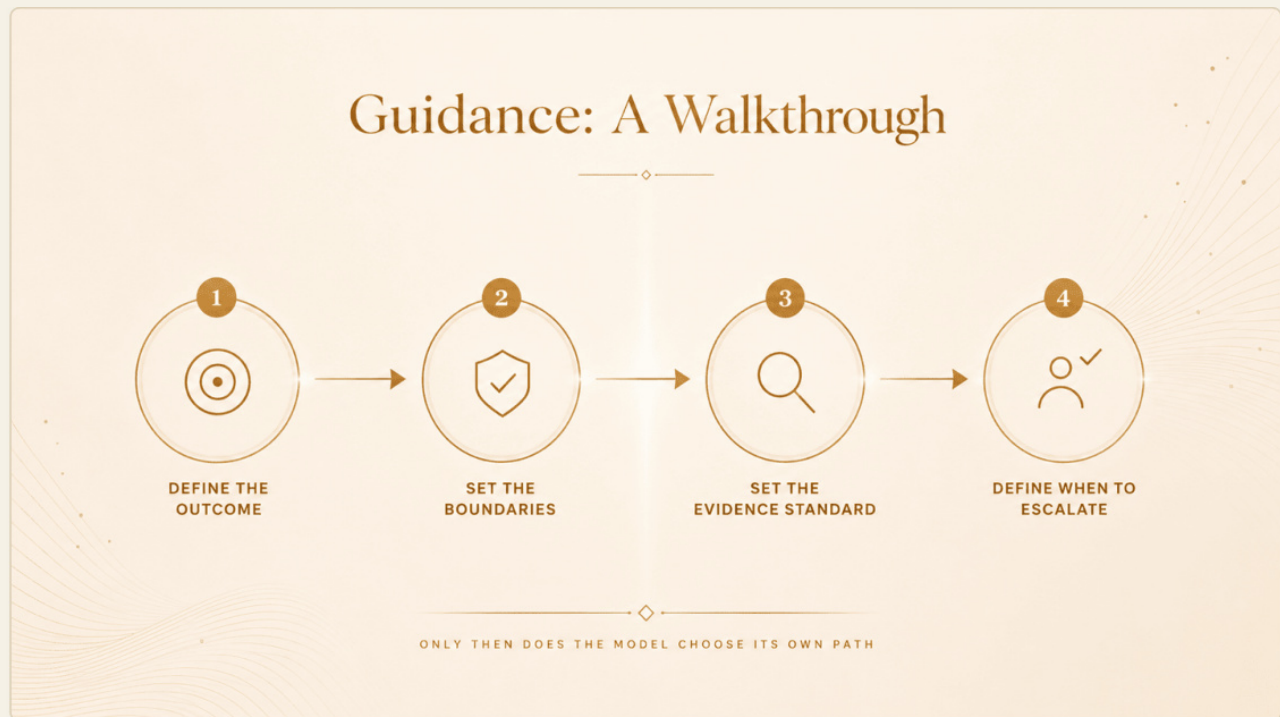
Consider an identity alert for a login from a new geography. A traditional playbook checks IP reputation, impossible travel, and MFA status, then closes the alert if those pass. An AI system may instead notice the same user recently received a suspicious OAuth consent request and accessed an unfamiliar SaaS app. No single signal clears the playbook's threshold, but together they suggest compromise: that a connection the AI would never make confined to the login playbook. The playbook doesn't make it safer, just blinder to context it already has.

Guidance Defines the Objective

Guidance isn't the absence of process but a different way of defining one. Instead of prescribing every step, it specifies the outcome, the boundaries, the evidence standard, and when human review is required. For an identity investigation, guidance might state:

- Determine whether the activity is consistent with the user's normal behavior and approved business activity.
- Consider authentication, endpoint, email, SaaS, and data-access evidence.
- Do not disable an account based on geolocation alone.
- Escalate before any action that could interrupt executive or production operations.
- Record the evidence that supports and contradicts the conclusion.

This defines what a good investigation must accomplish without forcing a fixed sequence and the model can order its own steps, request more evidence, and adjust as facts change.



Boundaries Matter More Than Instructions

The most important process artifact in an AI-native SOC may be a clear definition of what the system must not do. A capable model can generate many possible actions; the risk is letting it act without sufficient constraints. Investigating malware, it might be allowed to search telemetry and draft a summary, but not isolate a production server without approval, delete files, or close an incident while critical evidence is missing. These boundaries outlast a playbook because they stay useful even as the attack pattern changes.

Guidance Should Define Evidence Quality

AI-native investigations still need standards. A model shouldn't be rewarded for a confident answer alone, it should also show how the evidence supports it. For a suspected exfiltration incident, guidance might require identifying the user involved, the affected assets, the data's sensitivity, and whether the activity was authorized. That keeps the model from treating a plausible narrative as a confirmed incident and gives reviewers a consistent basis for evaluating the work.

Guidance Enables Adaptive Investigation

The strongest benefit of guidance is that the investigation can follow the evidence. An endpoint alert shows encoded PowerShell activity; a playbook requires decoding the command, checking the

file hash, and searching for persistence. An AI system under guidance may start there but expand once it finds the command pulled credentials from a browser store, pivoting to identity activity and data downloads. The path changes because the evidence changes for how an experienced analyst works, and the goal for AI too.

Playbooks Still Have a Role

Playbooks don't disappear. They remain useful for deterministic actions where consistency matters more than reasoning, like collecting a forensic package or blocking a confirmed malicious hash. What changes is where they sit in the operating model: they should govern repeatable execution, not the entire reasoning process. Guidance governs investigation and judgment; playbooks govern approved mechanical actions. That's ultimately a shift from procedure to intent. A legacy SOC tells the system exactly what steps to follow; an AI-native SOC defines the outcome, the evidence standard, the boundaries, and the points where accountability returns to a person. That isn't less control but a more appropriate form of control for a system that can reason.



Organizations that get this right won't just automate old runbooks faster. They'll build processes that let AI investigate broadly, adapt to new evidence, and operate safely within clearly defined limits.

Define the outcome. Not the route.

Guidance is not the absence of process. It is a different way of defining process: what good looks like, what evidence matters, what the system must not do, when authority returns to a person, and what must be recorded.

01	OUTCOME	What must the investigation accomplish?
02	EVIDENCE	What sources should be considered before a conclusion?
03	BOUNDARIES	What must never happen without approval?
04	ESCALATION	Which conditions require a human?
05	RECORD	What evidence and reasoning must remain reviewable?

Closed cases are not dead signal.

Deprioritization is a necessity. It should not become a statement of certainty.

Your Garbage Is Our Gold: Finding the Hidden Signal in the New AI Era

What keeps CISOs awake at night isn't always the incidents under active investigation. It's the ones already closed, downgraded, or deprioritized because they looked harmless at the time.

Every SOC runs on prioritization and there are always more alerts than analysts can investigate thoroughly. That's an unavoidable reality. But the incidents a SOC waves through as low-risk aren't necessarily noise. They may be its most valuable untapped threat intelligence.

The Triage Problem

An analyst reviews an incident tied to a software update, checks the process tree, validates the command-line arguments, and concludes the behavior looks like legitimate admin activity. Case closed as a false positive. Based on the evidence in front of them, that call is often perfectly reasonable.

What rarely happens next: checking whether the same executable showed up on other endpoints, whether similar commands ran outside the maintenance window, or whether unusual identity or network activity preceded it. That takes time, history, and data scattered across tools most SOCs don't have to spare.

The Cost of Deprioritizing

So deprioritization becomes a necessity, not a statement of certainty. Every closed case becomes part of an accumulating archive: weak signals, incomplete investigations, attacker behavior that never crossed a threshold.

What the SOC treats as garbage may be some of its most valuable threat intelligence if anyone ever goes back to look.

Traditional SOC workflows treat incidents as discrete units: enrich, group, assign severity, investigate, close. Once closed, an incident exits the workflow entirely.

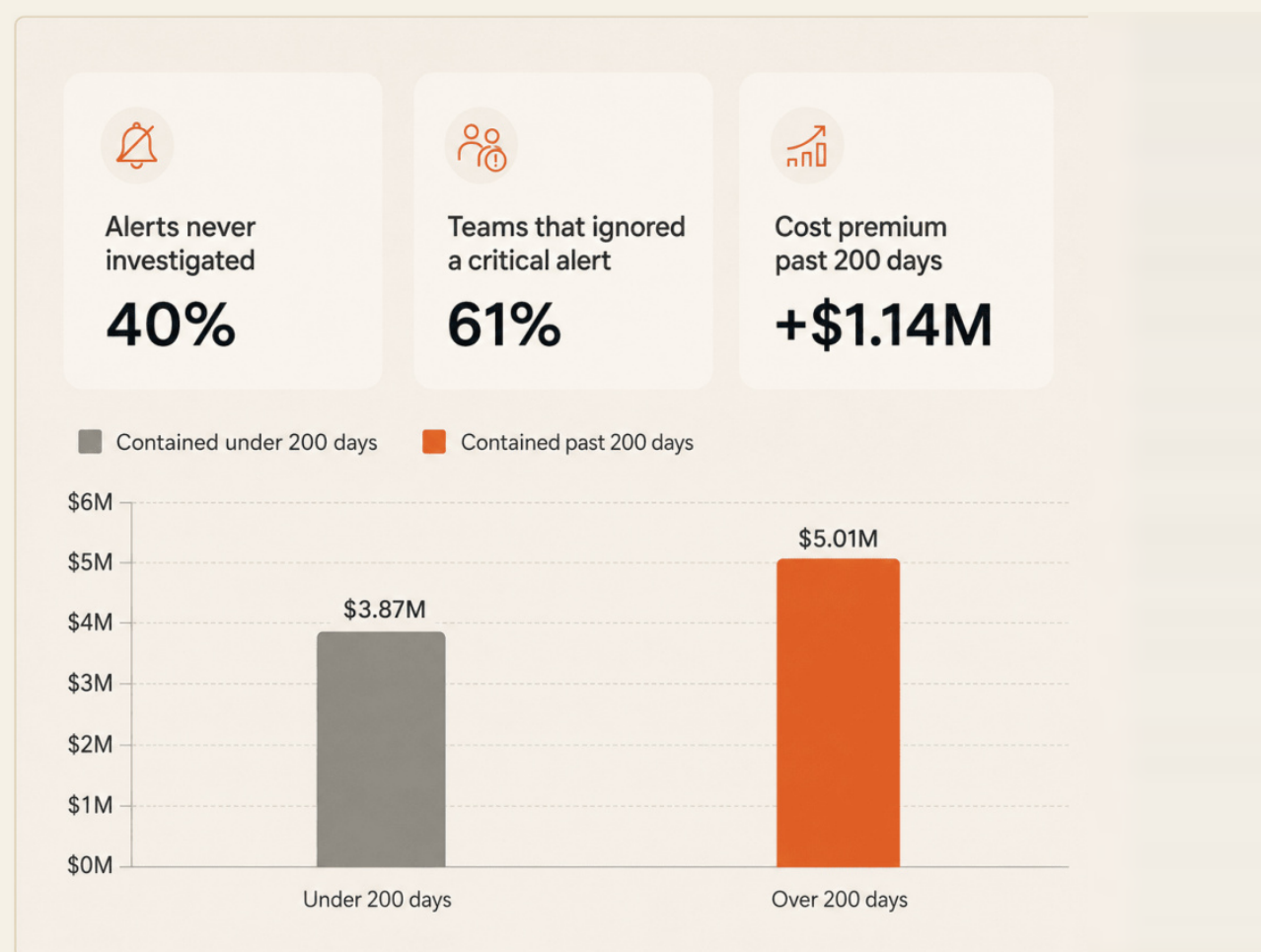
That works when malicious activity leaves a strong signature. It works far less well when an attacker spreads activity across time, identities, and endpoints. A suspicious PowerShell command,

a dormant login, a short-lived outbound connection and none of these justify escalation alone. Correlated over weeks, they can describe reconnaissance, credential testing, and command-and-control staging.

How Attackers Exploit the Gap

Attackers know how SOC prioritization works. They know analysts are time-constrained, that detection relies on thresholds, and that isolated anomalies get dismissed when they don't match a known pattern.

Many effective attacks aren't invisible; they're just insufficiently suspicious when viewed one incident at a time. The gap isn't detection coverage; it's the SOC's ability to retain, revisit, and reason across everything it has already seen.



According to [IBM's 2025 data breach report](#): 40% of alerts go uninvestigated and 61% of SOC teams admit ignoring one that turned out critical.

Why Earlier Correlation Tools Fell Short

Incident correlation isn't new. Earlier platforms linked alerts using statistical similarity, shared entities, timing, and risk scoring as well as clustering activity around users, endpoints, or IPs.

The problem was operational: correlation often produced more alerts and more cases, not fewer. A tool could surface a cluster, but a human still had to decide whether it was a coordinated attack or unrelated noise. Correlation became another queue in an already overloaded SOC, moving the burden downstream instead of reducing it.

What AI Changes

AI changes this in two ways. First, investigation becomes economically scalable because AI can examine process trees, authentication logs, and historical incidents across the environment without an analyst manually opening every case.

Second, modern language models add a reasoning layer correlation systems never had. Rather than flagging statistical similarity, AI can form and test hypotheses about how incidents relate, weigh conflicting evidence, and explain why a sequence is or isn't consistent with an attack.

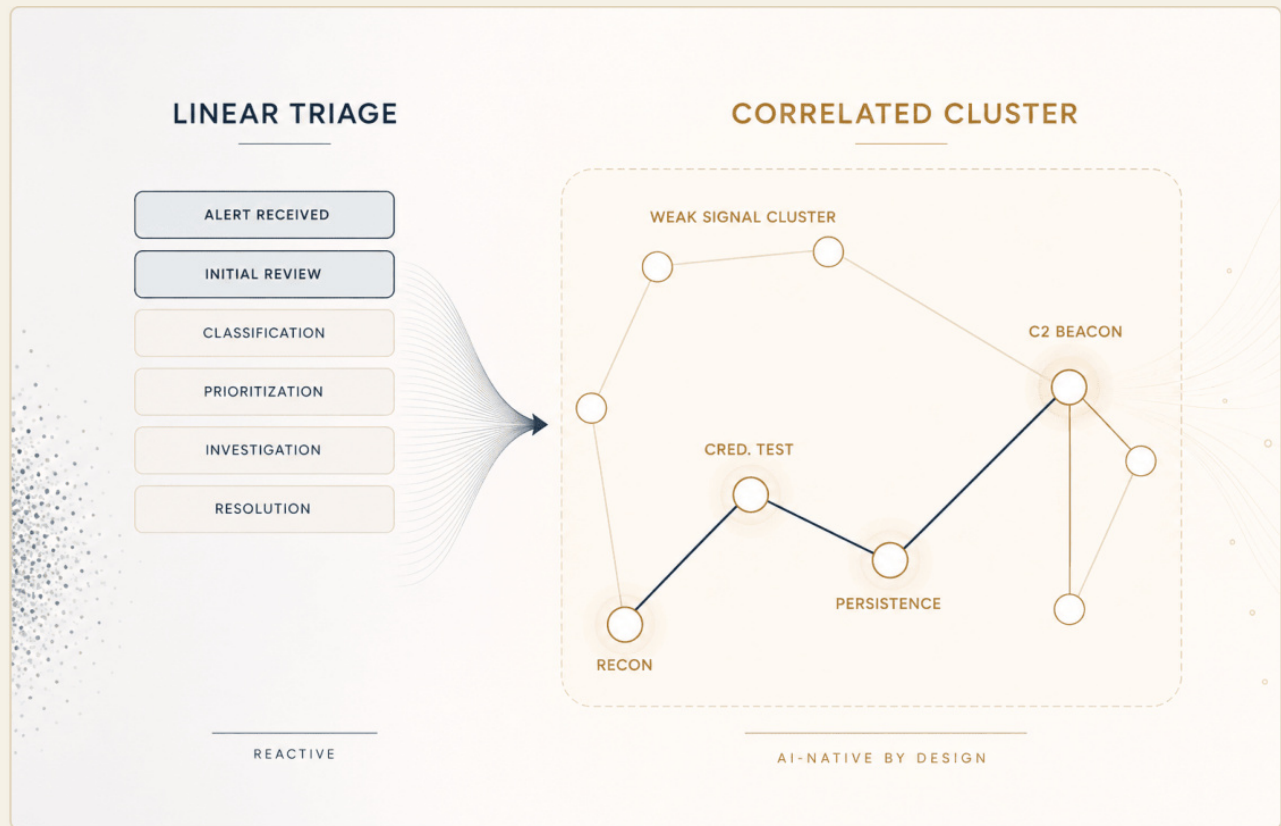
Reopening the Case

A low-severity incident no longer has to stay low severity forever. A process tied to an approved update can be re-evaluated if that binary later runs under a different parent process, appears on an unmanaged endpoint, or opens an unexpected connection.

The original call wasn't wrong, however, the surrounding evidence changed. An AI-native SOC can keep asking whether dismissed activity has become meaningful.

Nothing Is Truly Deprioritized

The promise here isn't that every incident gets immediate human attention (that just recreates the scaling problem under a new name). The promise is that nothing is permanently discarded. Every closed case and weak indicator stays available for reassessment against new evidence. For CISOs, that's a different kind of assurance: not a point-in-time judgment made under pressure, but continuous triage running across the full history of the environment.



The strongest signal rarely lives inside one incident. It emerges when hundreds of them (including the ones everyone wrote off as garbage) are read together as part of the same evolving story.

Next up: the economics of this shift, and whether an AI-native SOC is actually cheaper to run once investigative work starts moving from analysts to AI.

The signal often appears later.



Earlier correlation tools could link shared entities, timing and risk. Their operational problem was that they often created more alerts and more cases. AI adds a reasoning layer: it can form and test hypotheses across historical activity without requiring a person to open every case.

What can be reopened — and why?

Treat the revisit capability as part of the operating model, not as an analyst side project.

RETENTION	What context is retained on every low-risk close?
TRIGGERS	What new evidence should change the original significance?
CADENCE	How frequently does reassessment run across the archive?
MEASUREMENT	What is the reopen rate — and how do we know what should have been revisited?

AI capacity should be provisioned, not rationed.

Do not recreate the analyst queue with tokens as the new unit of scarcity.

Don't Fall Back Into the Same Loop in the AI SOC

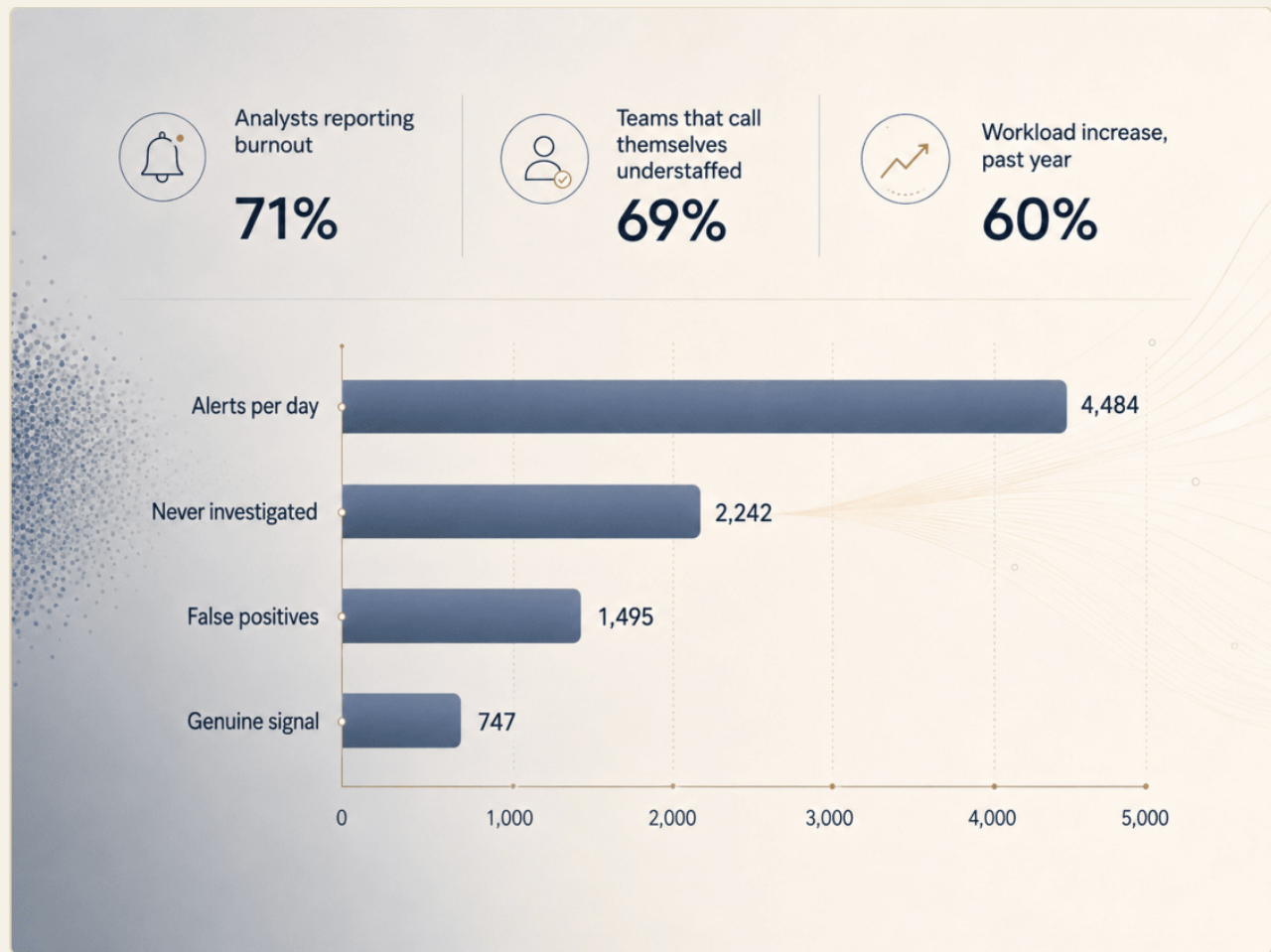
Token economics became a serious conversation the moment the industry realized we'd been getting an unreasonably generous discount from AI vendors. Now that free tiers are shrinking and usage caps are showing up everywhere, teams have started rationing tokens the way they once rationed analyst hours.

That instinct is understandable. It's also the same mistake that built the SOC staffing crisis in the first place.

The Ceiling Every SOC Already Hit

Traditional SOCs scale by hiring, and hiring has a hard ceiling: headcount costs money, so every team caps capacity somewhere below the alert volume it actually faces. The average organization now generates roughly 4,484 alerts a day, and about half go completely uninvestigated ([Netenrich, 2025](#)).

That gap isn't free. It shows up as burnout: 71% of SOC analysts report burnout, and 64% say they're likely to change jobs within a year ([Times, "Voice of the SOC Analyst"](#)). 42% of SOC leaders report that staff tenure is actively shrinking ([Netenrich, 2025](#)). Capacity limits didn't just create a coverage gap, they built a churn engine.



Repeating the Same Mistake with AI

Move to an AI-native SOC and the first question people ask is still a capacity question, just with tokens instead of headcount: if we send 10% of incidents to the model, what's the bill? Should we cut it to 5%? Frontier pricing makes that instinct rational in isolation, i.e. Claude Opus 4.6 runs \$5/\$25 per million input/output tokens, and top reasoning tiers elsewhere in the market run as high as \$30 per million input tokens ([CloudZero, LLM API Pricing Comparison 2026](#)).



But optimizing around that number recreates the exact ceiling the SOC just escaped. If AI capacity gets rationed the same way analyst hours were, incidents get triaged down to fit a budget rather than investigated because they matter.

Two Different Starting Questions

“How many tokens can we afford” and “how do we keep the model working around the clock on every weak signal” produce two different SOC. The first treats AI like an expensive analyst headcount line. The second treats it like infrastructure which is something provisioned for constant use, not metered scarcity.

That shift only works if cost isn’t the first constraint the design revolves around, which means the model layer underneath has to look nothing like a single frontier API call per incident.

Why the Pipeline Has to Be Multi-Tier

Pricing across the model market now spans a wide range, from \$0.10 per million input tokens for commodity models like GPT-4.1 Nano up to \$30 for the top reasoning tiers ([CloudZero, LLM API Pricing Comparison 2026](#)). That range is the point: a multi-tier pipeline routes routine triage to small, cheap, sometimes locally hosted models, and reserves frontier reasoning for the handful of cases that actually need it.

Quantization is what makes the cheap tier viable at scale. 4-bit quantization cuts GPU memory needs by roughly 75% with minimal accuracy loss. A 70B model that costs about \$24,000 in full-precision GPU hardware drops to around \$4,000 quantized, and monthly cloud inference for the same workload can fall from \$18,000 to under \$7,000 ([Latitude, “We Tested Quantized LLMs”](#)). Fixed, low-marginal-cost capacity at the bottom of the pipeline is what lets the top stay reserved for real escalations.

What AI Doesn't Burn Out On

Run a traditional SOC on “keep every analyst maximally busy at all times” and the first record broken is the attrition rate. No team survives that operating model. That’s precisely why the shift matters: AI doesn’t get tired, doesn’t quit, and doesn’t need a fifth of its headcount replaced every year. It just needs power, cooling, and a GPU.

That difference is the argument for flipping the pipeline. Once idle model capacity is nearly free, the constraint that forced deprioritization in the first place (a fixed number of tired humans) stops applying.

Where Cost Actually Belongs

None of this makes token cost irrelevant. It just moves it to the right place in the pipeline: a question asked about the handful of cases that escalate to frontier reasoning, not a gate applied to every incoming alert. Capable models like DeepSeek V3.2 already run at \$0.14/\$0.28 per million input/output tokens ([CloudZero, LLM API Pricing Comparison 2026](#)) which is cheap enough that “should we even look at this” stops being a real budget question.



The SOC industry spent a decade building a model where every analyst hit a hard capacity ceiling, and burnout is what that ceiling cost. Repeating the same design with tokens just moves where the ceiling sits. A multi-tier pipeline is cheap and quantized at the bottom and frontier reasoning at the top. This is what actually lets the constraint disappear instead of just changing currency.

In the next post, we'll dig into how we build that multi-tier pipeline and how we measure whether it's actually working.

Where is the budget constraint actually sitting?

INTAKE

Are we deciding cost on every incoming alert?

ROUTING

Which work earns a higher reasoning tier, and by what rule?

CAPACITY

What should be continuously available rather than metered?

PORTFOLIO

Which model classes are strategic infrastructure, and which are interchangeable?

Director output: a model portfolio decision — not a token budget spreadsheet.

Governance moves outside the model.

Security should not depend on the model remembering to behave securely.

Principles for Building an AI-Driven MDR

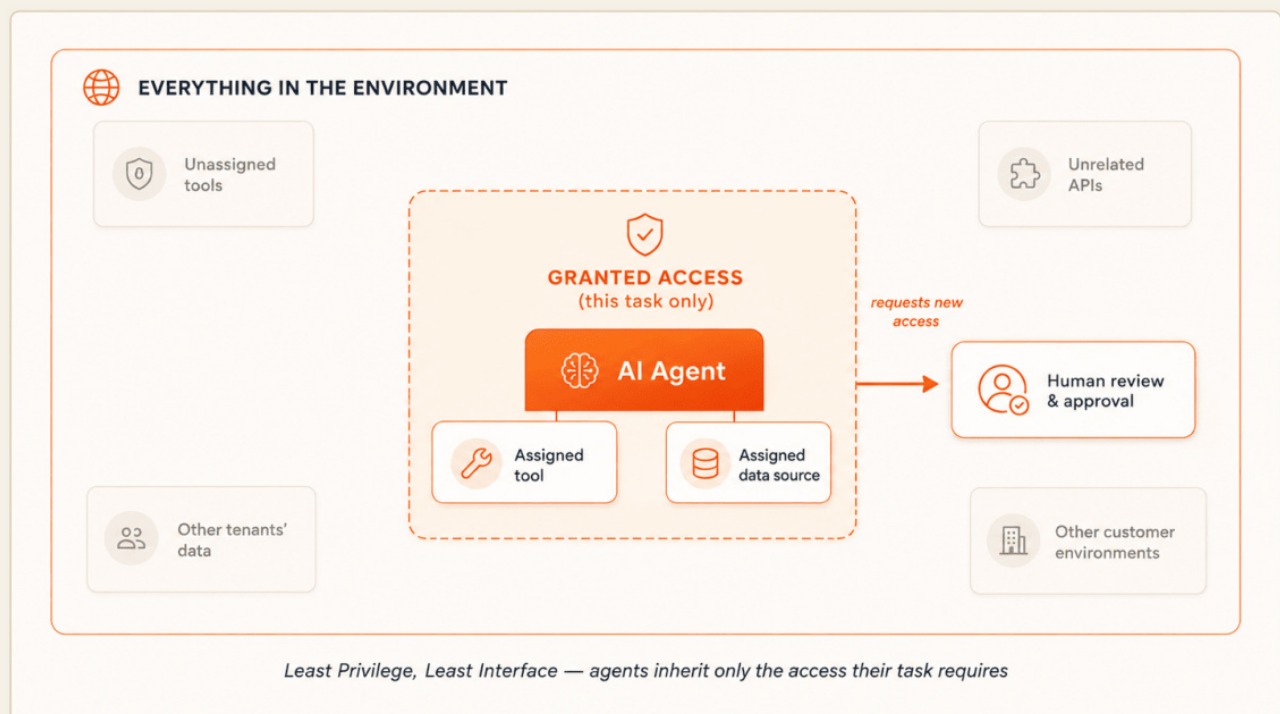
When cybersecurity experts set out to build an agentic pipeline at Sovera Security, one thing quickly becomes clear: the hardest problems are not really about AI. Zero trust, least privilege, alignment, continuous validation, optimization, and cost efficiency have to come first. AI comes after all of them.

These principles guide how we design an AI-driven MDR that can operate safely, efficiently, and at scale.

Least Privilege, Least Interface

We never set AI free inside a cybersecurity workflow. Security analysts operate with limited access to data sources, tools, and customer environments. AI agents should inherit those same constraints and receive access only to what their assigned task requires.

When an agent needs a new API, data source, or tool, it submits a request for human review and approval, much like a new employee requesting additional access. Capability should never imply unrestricted access.



Zero Trust

We treat AI agents the way we would treat a newly hired analyst: capable, but unproven and not yet trusted with the keys to everything.

Trust should be earned through a track record, not granted by default. An allowlist approach is the starting point. Key rotation is mandatory. Comprehensive auditing is built into the system. Requests for broader permissions require approval.

The result is an environment where agents have the access they need without accumulating unnecessary privilege.

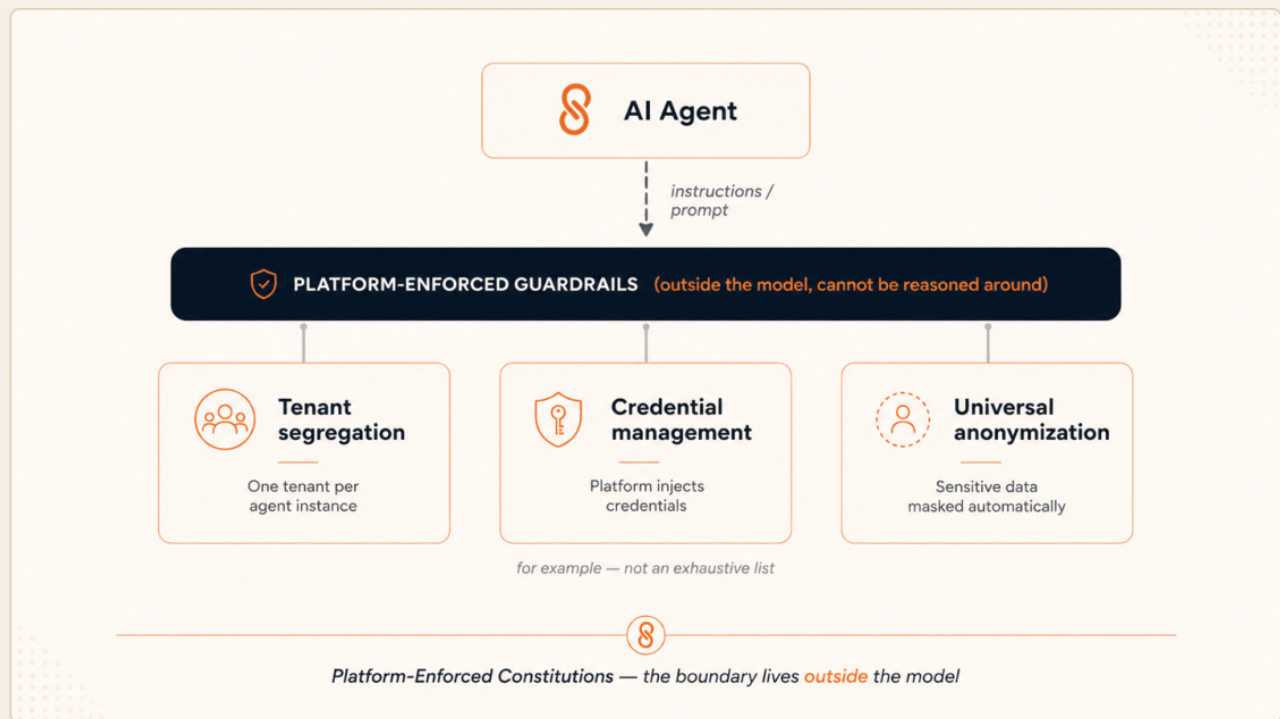
Platform-Enforced Constitutions

We do not give AI agents a set of rules and simply hope they follow them.

Prompt-based instructions are not a sufficient security boundary because the same model reading those instructions may also be processing adversarial input intended to manipulate its behavior. Critical constraints therefore need to live outside the model and be enforced by the platform.

For us, that includes principles such as:

1. **Tenant segregation.** Agents never specify a tenant ID in their requests. Each operates inside an isolated environment with access to only one tenant.
2. **Credential management.** Agents never see, select, or provide credentials. The platform injects the appropriate credentials when communicating with an external system.
3. **Universal anonymization.** When sensitive data leaves a tenant environment, the platform anonymizes it automatically without requiring agent involvement.



Security should not depend on the model remembering to behave securely.

Deterministic Orchestration and Evaluation

Workflows should remain deterministic whenever the underlying logic is already known. We do not ask an AI agent to calculate something that can be handled reliably by code. We write the loop or build the function, then allow the agent to operate within that structure.

AI should handle ambiguity, interpretation, investigation, and reasoning. Code should handle logic that is already well defined. This improves reliability while making agent behavior easier to test, evaluate, and audit.

Memory Is an Attack Surface

AI agents need to learn from failures, identify optimization opportunities, and apply established best practices. But memory itself introduces risk. The moment an agent learns to dismiss a pattern as a false positive may be the moment a threat actor starts using that same pattern in a real campaign.

Agent memory therefore cannot become an uncontrolled collection of historical conclusions. Every memory needs defined provenance, ownership, scope, confidence, and lifecycle management,

along with testing, validation, expiration, and cleanup. What an agent learns should be governed as carefully as what it can access.

Continuous Optimization

Defining the right boundaries for AI agents is an ongoing challenge. As models become more capable, overly specific prompts can sometimes make them less effective. Too much control limits their ability to reason through unfamiliar situations. Too little can make workflows unpredictable or unsafe.

The goal is to continuously find the right balance between deterministic and nondeterministic behavior. That balance will change as models improve. A task that requires tightly constrained orchestration today may be safely delegated to a more capable model tomorrow.

Continuous optimization is therefore part of the architecture, not a one-time tuning exercise.

Model Hierarchy

A cybersecurity agentic pipeline also has to be cost efficient. Otherwise, it inherits one of the most damaging bottlenecks of the traditional SOC: deprioritization.

When investigation is expensive, teams inevitably decide which alerts deserve attention and which can be ignored. AI should reduce that constraint, not reproduce it. A model hierarchy assigns the right model to the right task based on capability, complexity, and cost.

Lightweight models can handle high-volume routine work so incidents are not left uninvestigated simply because they appear insignificant at first glance. More capable models can step in when investigations require deeper reasoning, orchestration, or judgment. The most capable models should be reserved for the highest-leverage work, including improving the system itself rather than repeatedly processing routine cases.



The objective is not to use the most powerful model everywhere. It is to make advanced reasoning available across the operation without allowing cost to become another form of prioritization.

Model Integration Patterns

A model hierarchy does not mean every model operates independently.

Different models can contribute different strengths to the same investigation. One model might investigate while another evaluates its conclusion. Multiple models might produce independent verdicts that are reconciled before action is taken. In other cases, models may collaborate on different parts of the same investigation, with a more capable model coordinating their work.

The important question is not which single model should own the workflow. It is how models, deterministic systems, and human analysts can work together to produce a better outcome than any one component could achieve alone.

Building AI Around Security

AI can fundamentally change how managed detection and response operates, but only if the surrounding system is designed to support it.

The goal is not to bolt an LLM onto an existing SOC process and call it AI-driven MDR. It is to rethink how work should be divided across deterministic software, AI models, and human expertise. That means constraining agents before expanding autonomy, enforcing security at the platform layer, governing memory as infrastructure, matching model capability to the economics of the task, and continuously validating how the system behaves.

The models will keep improving. The architecture around them is what determines whether those improvements translate into a security operation that is safer, more scalable, and more effective.

Put the non-negotiables outside the model.

01	LEAST PRIVILEGE	Agents receive only the access their task requires.
02	ZERO TRUST	Trust is earned through a track record; allowlists, key rotation and auditing are foundational.
03	PLATFORM CONSTITUTIONS	Tenant segregation, credential handling and anonymization are enforced independently.
04	DETERMINISTIC ORCHESTRATION	Known logic stays in code; AI handles ambiguity, interpretation and reasoning.
05	GOVERNED MEMORY	Every memory has provenance, ownership, scope, confidence and lifecycle management.
06	CONTINUOUS OPTIMIZATION	The deterministic/AI boundary is revisited as models improve.
07	MODEL HIERARCHY	Lightweight models handle routine volume; capable models are reserved for higher-leverage work.

What changes require human authority?

ACCESS

Who can grant an agent a new API, data source or credential scope?

POLICY

Where are the critical constraints enforced outside the model?

MEMORY

What may an agent remember, for how long, and with what provenance?

MODEL

What evidence is required before replacing the model underneath a consequential workflow?

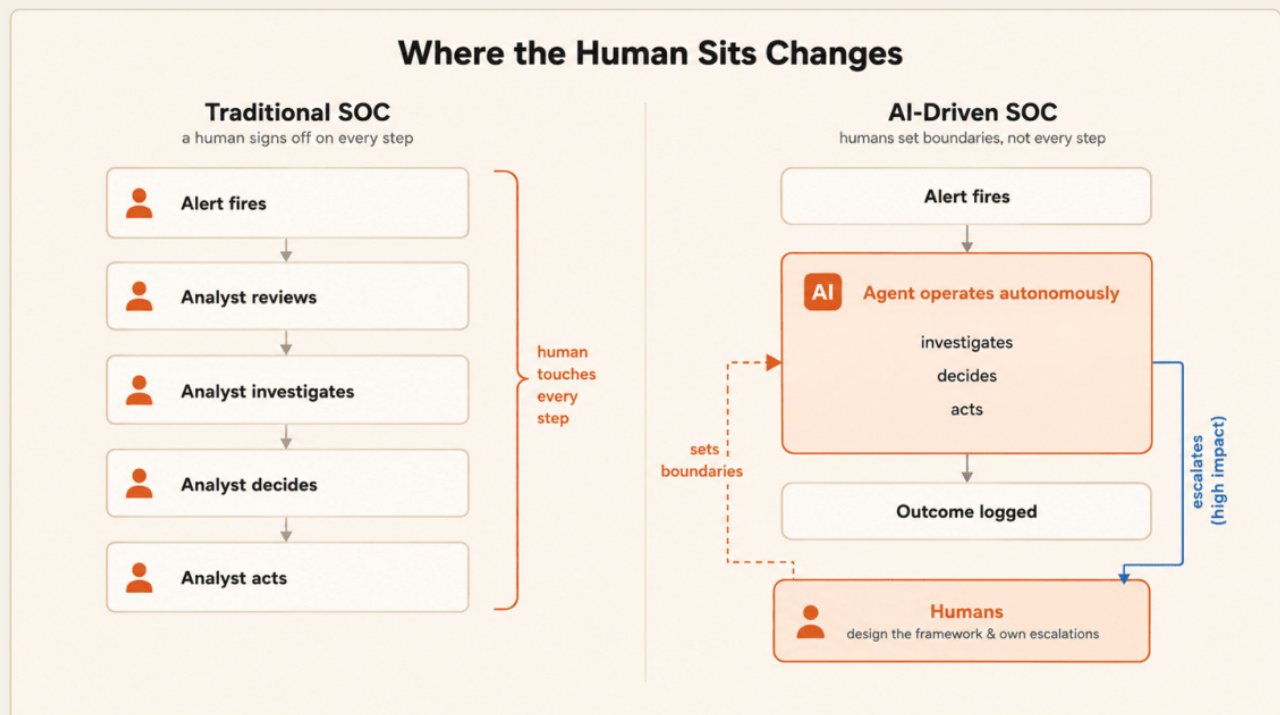
Humans move up the accountability stack.

The future SOC is better understood as a pyramid of accountability: AI at the broad operational base; humans progressively higher.

The Human Accountability Stack

At his [ICM 2026 public lecture](#), Terence Tao described mathematics entering an era of *proof abundance* rather than proof scarcity. If AI can generate correct proofs faster than mathematicians can absorb them, the bottleneck stops being generation and becomes digestion: deciding what is correct, what matters, and how it fits into the field's shared understanding.

Cybersecurity is heading toward the same inflection point. For decades, the SOC has operated under scarcity: too many alerts, too little correlation, and too few analyst-hours. Most modern security tooling has therefore been designed to help scarce human analysts move faster. AI changes that constraint. An agent can generate hypotheses, query systems, weigh evidence, and act at a speed no human team can match. The question is no longer simply how to produce more analysis. It becomes how to turn a flood of machine-generated analysis into decisions that can be trusted, governed, and defended.



This is why Anton Chuvakin's warning against "[building a 2003 SOC with AI](#)" matters. Bolting an agent onto every step of the existing analyst workflow misses the real shift underway, which is a redistribution of where humans sit in the system.

Humans will not disappear from the loop. They will move up it. Picture a pyramid: at the base is an enormous number of bounded, low-consequence decisions, while near the top there are far fewer decisions, each with a much greater blast radius. AI increasingly owns the base, while humans own the top and define how the base is allowed to operate.

Layer 1: AI Runs the Operational Floor

Most of what fills an analyst's day, including querying identity logs, checking a domain against threat intelligence, correlating events, or blocking a known malicious indicator, is repetitive and well bounded. Requiring human sign-off on every such action adds friction without necessarily adding meaningful control.

The human role therefore shifts earlier in the process. Humans decide what an agent can see, what credentials it holds, and which actions are consequential enough to require escalation. Removing a phishing email and isolating a production server are not equivalent decisions, even if a model classifies both as response actions. Humans define that distinction, while the agent operates autonomously within the boundaries they establish.

Layer 2: Humans Own the Process, Not Every Verdict

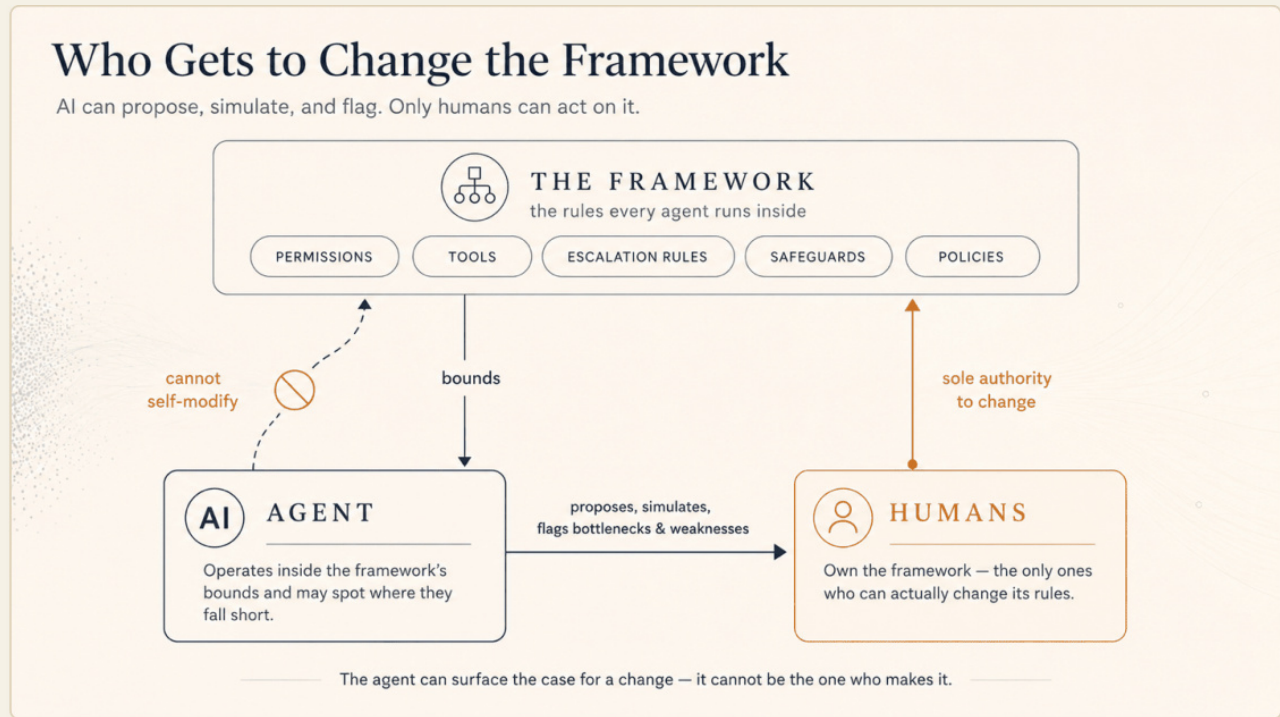
When an AI system gets something wrong, such as repeatedly missing account takeovers because it never considers credential stuffing, fixing that one case is not enough. The same blind spot can recur across thousands of future investigations unless someone identifies the source. The problem might be an instruction, a missing data feed, a retrieval gap, an integration issue, or a flawed branch in the workflow.

This creates a much more scalable form of oversight. One analyst catching one bad verdict improves one incident. An analyst who identifies and corrects a systemic weakness improves every future investigation that would otherwise have repeated the same mistake. As autonomous volume grows, this is where human judgment becomes most valuable.

Layer 3: Humans Govern the System That Governs the Agents

Every agent operates inside a framework of permissions, tools, escalation rules, safeguards, and policies. As AI systems become more capable, they will increasingly identify limitations in that framework themselves and may correctly conclude that broader credentials or fewer approval gates would make them more effective.

That does not mean they should be able to grant those permissions to themselves. A system that can expand its own authority, weaken its own safeguards, or redefine the rules governing its behavior creates a fundamental governance problem, regardless of how sound its reasoning may be.



AI can propose changes, simulate their effects, identify bottlenecks, and flag weaknesses. The authority to alter the framework itself, however, should remain independently owned by humans.

Layer 4: Model Choice Is a Governance Decision, Not Routine Maintenance

Different models can behave very differently when presented with identical evidence. Some may chase too many false leads, while others converge too quickly. Some may handle incomplete information well but be more vulnerable to manipulation. These differences matter because the underlying model influences how every workflow built on top of it reasons.

Replacing a conventional software component is often routine maintenance. Replacing the model underneath an agentic SOC can be far more consequential because it may change the behavior of the entire system without anyone modifying the surrounding process or policy layer.

AI can continuously benchmark models, compare performance, and route work toward the best option for specific tasks. For lower-risk workflows, that degree of automation may soon become routine. For high-impact decisions, however, allowing the system to choose its own successor

introduces a recursive governance problem. Humans should continue to own the criteria for adopting or replacing the models that influence consequential security decisions.

Layer 5: Humans Decide What the SOC Is Optimizing For

At the top of the pyramid, the question is no longer execution. It is objectives. Every SOC operates under real constraints, including budget, headcount, risk tolerance, regulatory exposure, infrastructure costs, and operational priorities. AI can help identify waste, model tradeoffs, and show where additional controls are likely to produce the greatest marginal benefit.

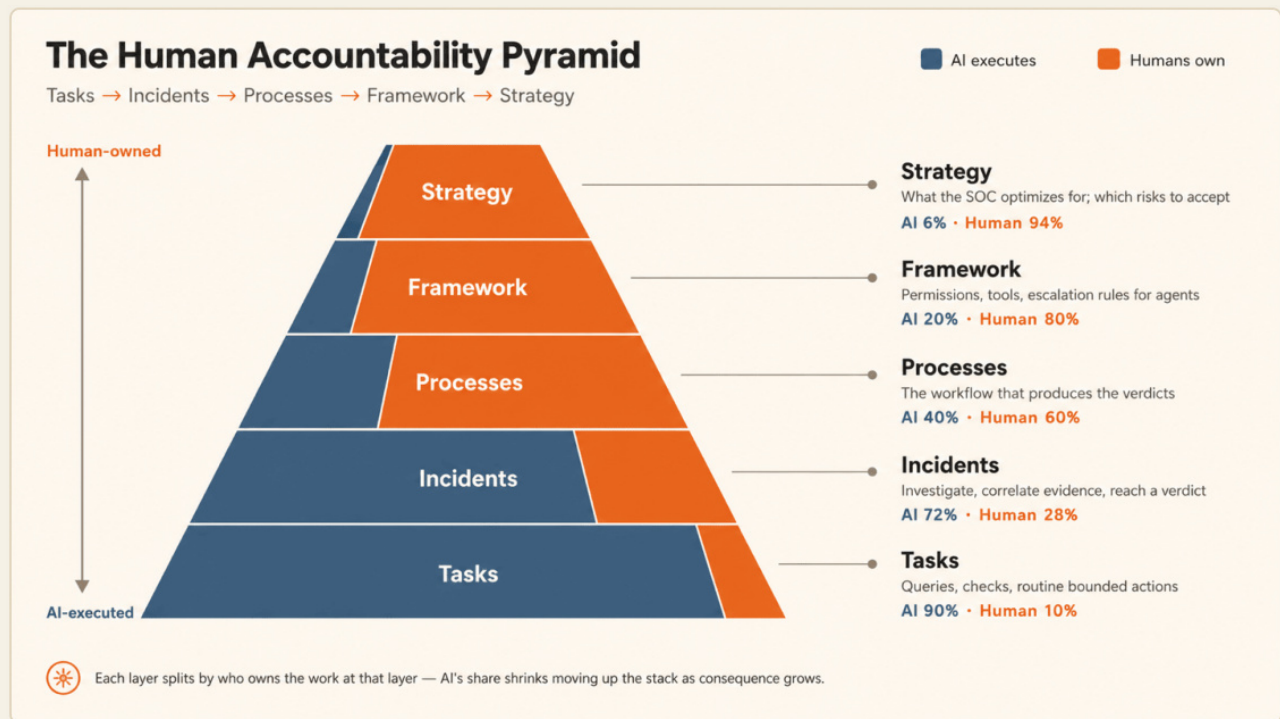
What AI should not determine independently is which risks are worth accepting. Two organizations with nearly identical technical environments may rationally choose very different security postures because the business consequences of failure are different for each of them. That is not simply a modeling problem. It is a strategy problem.

Security strategy reflects business priorities, regulatory obligations, customer expectations, financial constraints, and organizational risk appetite. Those decisions remain human because they determine not merely how the SOC operates, but what the organization is ultimately willing to protect, spend, and risk.

The Shift Is Upward, Not Away

As AI takes on more of the SOC's operational work, humans will become less visible at the level of individual queries, verdicts, and containment actions. That is not necessarily a problem. What matters is that accountability does not shrink along with that visibility. It relocates.

Instead of approving one query, humans define which categories of queries are safe to automate. Instead of correcting one verdict, they improve the process responsible for producing thousands of verdicts. Instead of supervising every response action, they govern the framework that determines when autonomous response is permitted. Instead of selecting every model manually, they define the standards under which model changes are acceptable. At the highest level, they determine the objectives and risk tradeoffs the entire system is meant to serve.



“Human in the loop” was a useful phrase for a world in which machines largely assisted human decision-makers. An AI-driven SOC requires a broader concept. Humans may no longer sit inside every operational loop, but they remain responsible for the structure around those loops, the authority granted to the systems operating within them, and the consequences that follow.

The future SOC is therefore better understood as a pyramid of accountability. AI occupies the broad base, executing the enormous volume of routine work that can be safely bounded and automated. Humans sit progressively higher in the structure, taking responsibility for process design, governance, model selection, and strategy. Their role becomes less frequent at the task level but more consequential at the system level, because the decisions made at the top determine how everything beneath them is allowed to operate.

Accountability relocates. It does not disappear.

01	OPERATIONAL FLOOR	AI runs bounded, repetitive work.
02	PROCESS	Humans improve the system when a recurring weakness appears.
03	GOVERNANCE	Humans own permissions, tools, escalation rules and safeguards.
04	MODEL CHOICE	Humans own the standards for consequential model changes.
05	STRATEGY	Humans decide objectives, risk tradeoffs and what the SOC is optimizing for.

The higher the layer, the less frequent the decision — and the greater its blast radius.

The toolkit is for decisions, not worksheets.

Use these five director-level tools to move the five shifts from ideas into operating decisions. Each asks what must change in your process, architecture, governance or accountability model.

01 Guidance-to-Governance

Turn one rigid workflow into an approved autonomous operating envelope.

02 Continuous Signal Reassessment

Design how low-risk cases remain available to future evidence.

03 AI Capacity & Model Portfolio

Decide how model tiers support continuous investigation.

04 Agent Governance & Control Plane

Test whether critical constraints are enforced outside the model.

05 Human Accountability Stack

Map decisions upward to the level that should own them.

The Guidance-Writing Framework

Adapted from “Playbooks vs. Guidance”

Pick one playbook your team still runs on rigid, step-by-step logic. Rewrite it as guidance using the same five elements Sovera uses for identity investigations — outcome, evidence, boundaries, escalation, and record-keeping.

Worked example • Identity investigation

FROM THE SHIFT

Outcome. Determine whether the activity is consistent with the user’s normal behavior and approved business activity.

Evidence to weigh. Authentication, endpoint, email, SaaS, and data-access evidence.

Boundaries. Do not disable an account based on geolocation alone.

Escalation trigger. Escalate before any action that could interrupt executive or production operations.

Evidence to record. The evidence that supports and contradicts the conclusion.

Your playbook, rewritten

PLAYBOOK BEING CONVERTED

OUTCOME

What does a good investigation have to accomplish, stated as a goal, not a sequence of steps?

EVIDENCE TO WEIGH

Which sources should the system draw on before concluding anything?

BOUNDARIES

What must the system never do without a human approving it first?

ESCALATION TRIGGER

What conditions require a human before any action is taken?

EVIDENCE TO RECORD

What has to be documented so a reviewer can evaluate the reasoning later?

The Deprioritized-Incident Revisit Protocol

Adapted from “Your Garbage Is Our Gold”

A closed incident isn’t a conclusion — it’s a bet that new evidence can still overturn it. Use this to check what your SOC keeps, what should trigger a second look, and how often that second look actually happens.

Part A • What to retain on every closed incident

If this doesn’t get kept, it can’t be reassessed later.

- ☐ Process tree and command-line arguments at the time of the original call
 - ☐ Identity and network activity in the surrounding window — not just the flagged event
 - ☐ Timing relative to other activity on the same endpoint, user, or network segment
 - ☐ The analyst’s own reasoning for closing it, in their words
-

Part B • What should reopen a case

- ☐ The same binary or script resurfaces on a different endpoint
- ☐ It runs under a different parent process than the one that made it look routine
- ☐ It appears on an unmanaged or newly discovered endpoint
- ☐ Unusual identity or network activity shows up before or after it, in hindsight
- ☐ A pattern it resembles gets confirmed malicious somewhere else in the environment

Part C • Cadence and ownership

SWEEP CADENCE

How often does something — a person or a system — re-check closed, low-severity incidents against new evidence?

OWNERSHIP

Who or what is responsible for running that sweep, and where does its output go?

CURRENT REOPEN RATE

Of everything closed as low-risk in the last quarter, how many were ever revisited? How many should have been?

The Multi-Tier AI Pipeline Worksheet

Adapted from “Don’t Fall Back Into the Same Loop”

Two different starting questions produce two different SOC’s. Check which one your organization is actually asking — then map what a multi-tier pipeline would look like in your own environment.

Part A • Which question are you actually asking?

Check the one that matches how your team talks about AI cost today.

- ☐ “How many tokens can we afford to spend on this?”
 - ☐ “How do we keep a model working around the clock on every weak signal?”
-

Part B • Map your own pipeline

TIER 1 — COMMODITY / QUANTIZED

Always-on, cheap enough to run against everything. What routine triage lives here today?

TIER 2 — MID-TIER REASONING

Handles what Tier 1 flags as ambiguous. What sits here today, if anything?

TIER 3 — FRONTIER REASONING

Reserved for the cases that actually need it. What earns escalation to this tier?

Part C • Where should the cost question apply?

TODAY, COST GETS QUESTIONED AT:

Intake (every incoming alert) or escalation (only the handful that reach the top tier)?

The Agent Governance Self-Audit

Adapted from “Principles for Building an AI-Driven MDR”

Seven engineering principles behind Sovera’s own agentic pipeline. Check the ones already true of your environment today, and use the note line for what would have to change to check the rest.

- ☐ **Least Privilege, Least Interface.** Agents inherit the same restricted access a human analyst would have — nothing more by default.
- ☐ **Zero Trust.** New agents are treated like a newly hired analyst: capable, but unproven, with trust earned through a track record.
- ☐ **Platform-Enforced Constitutions.** Critical constraints (tenant isolation, credential handling, anonymization) live outside the model, enforced by the platform — not hoped for in a prompt.
- ☐ **Deterministic Orchestration.** Logic that’s already well defined runs as code. AI is reserved for ambiguity, interpretation, and judgment.
- ☐ **Memory Is Governed.** Every memory an agent forms has defined provenance, ownership, scope, confidence, and an expiration — not an unmanaged pile of past conclusions.
- ☐ **Continuous Optimization.** The boundary between deterministic and AI-driven behavior is revisited as models improve, not set once and left alone.
- ☐ **Model Hierarchy.** Lightweight models handle routine volume; the most capable models are reserved for the highest-leverage work.

What would have to change

For each principle you could not check, note the specific change required, who owns it, and what evidence would prove it is in place.

PRINCIPLE	CHANGE REQUIRED	OWNER • EVIDENCE

The Human Accountability Stack Mapper

Adapted from “The Human Accountability Stack”

The pyramid, condensed — then a worksheet to place your own team’s decisions on it.

The five layers

LAYER 1 — AI RUNS THE OPERATIONAL FLOOR

Bounded, repetitive, well-defined actions — querying logs, correlating events, blocking a known indicator.

LAYER 2 — HUMANS OWN THE PROCESS

Not every verdict — the systemic fix when a blind spot recurs across many investigations.

LAYER 3 — HUMANS GOVERN THE FRAMEWORK

The permissions, tools, and escalation rules agents operate inside — which agents cannot expand for themselves.

LAYER 4 — MODEL CHOICE IS GOVERNANCE

Replacing the model underneath an agentic SOC can change system behavior everywhere at once.

LAYER 5 — HUMANS DECIDE WHAT TO OPTIMIZE FOR

Risk tolerance, regulatory exposure, and which business consequences are acceptable — a strategy question, not a modeling one.

Map three decisions your team makes today

DECISION	SITS AT TODAY	SHOULD SIT AT

The argument, condensed.

Seven pages for the reader who has to brief someone else: the summary, the worked example, the roadmap, the scorecard and the audit that keeps it honest.

The AI-Native SOC

Executive summary – the argument in ten minutes.

01 • GUIDANCE

Replace rigid playbooks for reasoning work with guidance that defines the outcome, evidence standard, boundaries, escalation points, and required record.

02 • CONTINUOUS REASSESSMENT

A closed incident is not permanently discarded. New evidence should be able to change the significance of an earlier low-risk decision.

03 • AI CAPACITY

Do not recreate analyst scarcity as token scarcity. Use a multi-tier model pipeline so routine work can run continuously while deeper reasoning is reserved for the cases that need it.

04 • PLATFORM GOVERNANCE

Critical constraints should be enforced outside the model: least privilege, tenant segregation, credential handling, anonymization, deterministic orchestration, and governed memory.

05 • HUMAN ACCOUNTABILITY

Humans move upward. AI increasingly owns bounded operational work; humans own the processes, framework, model standards, and strategic risk decisions that determine how that work may occur.

The Five Shifts

One operating-model change, expressed five ways.

PLAYBOOKS → GUIDANCE

Procedure becomes intent.

Outcome · Evidence · Boundaries · Escalation · Record

CLOSED → REVISITABLE

Deprioritization becomes provisional.

Retain context · Re-correlate · Reassess when evidence changes

ANALYST CAPACITY → AI CAPACITY

Scarcity moves out of the workflow.

Always-on routine triage · tiered reasoning · frontier when needed

MODEL TRUST → PLATFORM GOVERNANCE

Security cannot depend on prompt compliance.

Least privilege · zero trust · isolation · credential control · governed memory

HUMAN-IN-THE-LOOP → HUMAN ACCOUNTABILITY

Humans move from every verdict to system-level ownership.

Process · framework · model standards · strategy

From Theory to Practice

Worked example · identity investigation.

Scenario: a login appears from a new geography.

OLD SOC	Check IP reputation → impossible travel → MFA → close if thresholds pass.
AI-NATIVE SOC	Start with the objective: determine whether activity is consistent with the user and approved business activity.
EVIDENCE	Authentication, endpoint, email, SaaS, and data-access context. The agent may expand the investigation as findings change.
BOUNDARY	Do not disable an account based on geolocation alone.
ESCALATION	Escalate before an action that could interrupt executive or production operations.
RECORD	Capture evidence supporting and contradicting the conclusion so the work can be evaluated later.

The key change is not “more automation.” It is that the system is given a goal, an evidence standard, and boundaries, then allowed to adapt its investigation path as evidence changes.

The 90-Day AI-Native SOC Roadmap

A practical starting sequence built directly from the five shifts.

DAYS 1–30

Define

Select one high-volume investigation. Rewrite the existing playbook as guidance using outcome, evidence, boundaries, escalation, and record-keeping.

Deliverable: one approved guidance artifact and one deterministic action list.

DAYS 31–60

Reassess

Define what context is retained when incidents close and what new evidence should trigger a revisit. Assign ownership for the sweep.

Deliverable: a revisit protocol with cadence, triggers, and reporting.

DAYS 61–90

Govern

Map your model tiers, agent permissions, platform-enforced controls, memory lifecycle, and human escalation points.

Deliverable: an AI-SOC governance baseline and accountability map.

The roadmap covers the first three shifts end to end and opens the fourth. The self-audit timeline on the following pages is what keeps it honest after day 90.

AI-SOC Maturity Scorecard

Use 0–3 to establish a baseline: 0 = absent · 1 = ad hoc · 2 = defined · 3 = operational and measured.

AREA	0–3	EVIDENCE TO LOOK FOR
Guidance	--	Outcome, evidence, boundaries, escalation, record
Reassessment	--	Closed incidents retained and revisited against new evidence
Model hierarchy	--	Routine / mid-tier / frontier routing with clear triggers
Agent access	--	Least privilege, allowlists, key rotation, auditability
Platform controls	--	Tenant isolation, credential injection, anonymization
Deterministic orchestration	--	Known logic handled in code; AI handles ambiguity
Memory governance	--	Provenance, ownership, scope, confidence, lifecycle
Human accountability	--	Clear ownership of process, framework, model standards, strategy

BENCHMARK YOUR TOTAL · 8 AREAS × 0–3 = 24 POINTS

0–7 HUMAN-SCARCITY SOC The operating model still assumes analyst attention is the binding constraint. Procedure is rigid and closure is final. Start at shifts 01 and 02.	8–13 ASSISTED SOC AI is bolted onto the old workflow. Practice is documented in places and inconsistent in others. Start at shift 03: provision capacity instead of rationing it.	14–19 AI-NATIVE, GOVERNED Defined across most areas and enforced outside the model. The remaining work is measurement and ownership — shifts 04 and 05.	20–24 OPERATIONAL & MEASURED All five shifts are in operation and evidenced. The question changes from “are we doing this?” to “can we prove it, quarter after quarter?”
---	---	---	--

HOW TO READ THE SCORE

Any single 0 outranks the total. An 18 with no memory governance is more exposed than a 12 with nothing missing. Close every 0, then every 1, before pursuing a 3 anywhere.

Two is the target, not three. Most enterprise SOC's should aim for *defined* across all eight areas before pursuing *operational and measured* in any one of them. Uneven maturity is where incidents live.

Score against evidence, not intent. If a reviewer cannot see the artifact — the written guidance, the retention setting, the routing rule, the permission grant — the area scores a 1 at most. A high score is not the objective: the useful output is knowing where the operating model still depends on human scarcity, rigid procedure, implicit trust, or point-in-time decisions.

The Self-Audit Implementation Timeline

How to run the scorecard, act on what it shows, and keep it honest — twelve weeks, then a cadence.

WEEK 1 Scope it and own it

Name one accountable owner and one executive sponsor. Freeze the scope, then assemble the evidence you will score against: guidance documents, permission grants, routing rules, retention settings.

Deliverable: a one-page scope with a named owner and sponsor.

WEEK 2 Score the baseline

Two assessors score all eight areas independently, then reconcile — disagreement is itself a finding. Every score points at an artifact; an unevidenced 2 is a 1.

Deliverable: a baseline scorecard with an evidence reference per area.

WEEKS 3–4 Register the decisions

For each governance decision class, record the owner, the boundary or standard and the review cadence. Anything without a named owner is the audit's first finding.

Agent permissions · Escalation classes · Model replacement · Memory retention · Reassessment triggers · Optimization objective

Deliverable: a governance decision register with named owners.

WEEKS 5–8 Close the zeros

Remediate every area scoring 0 first, then every 1. Move critical constraints out of prompts and into the platform. Do not chase a 3 while a 0 remains.

Deliverable: shipped control changes, each mapped to the area it moves.

WEEKS 9–10 Test the boundary

Verify by exercise, not by document. Attempt an action outside the agent's envelope, confirm the escalation classes fire, and replay a closed low-risk incident against new evidence to see whether it reopens.

Deliverable: test results, exceptions, and accepted risks.

WEEKS 11–12 Re-score and publish

Re-run the scorecard with the same assessors and evidence standard. Publish baseline against current to the sponsor, with one target per area for next quarter.

Deliverable: a delta report and next-quarter targets.

THEN KEEP THE CADENCE

Quarterly — re-score, review the register, report the delta. **Annually** — full re-audit with sponsor sign-off. **On change** — re-assess whenever a model, data source or autonomy boundary changes. An audit that runs once is a document; one that runs on a cadence is a control.

The AI-Native SOC: Where to Start

The objective is not to automate the old SOC faster. It is to redesign where work, authority, evidence, and accountability sit.

☐ Start with one workflow

Choose a high-volume investigation where the existing playbook is mostly deterministic but still requires contextual judgment.

- ☐ Pick a single investigation type with enough daily volume to show a result inside one quarter
- ☐ Confirm the current playbook is written down — and that it is what the team actually follows
- ☐ Record today's baseline: volume, close rate, false-positive rate, mean time to close

☐ Define the boundaries

Make the safe operating envelope explicit before expanding autonomy.

- ☐ Write the outcome, evidence standard, boundaries, escalation trigger and record for that workflow
- ☐ List the actions the agent must never take without a human approving them first
- ☐ Move every critical constraint out of the prompt and into the platform

☐ Keep the history

Treat closed incidents as evidence that can become meaningful later, not as permanently discarded work.

- ☐ Define what context is retained on every low-risk close
- ☐ Define what new evidence should reopen a case, and what should not
- ☐ Set the sweep cadence and name who owns its output

☐ Provision the pipeline

Build model tiers so continuous investigation does not depend on using the most expensive model for every task.

- ☐ Map which routine triage can run on always-on commodity or quantized capacity
- ☐ Set the routing rule that earns a case a higher reasoning tier
- ☐ Move the cost question from intake to escalation

☐ Move accountability upward

Humans should own the system that produces decisions: its processes, permissions, models, and strategic objectives.

- ☐ Name the owner for each governance decision class
- ☐ Define the evidence required before replacing a consequential model
- ☐ Agree what the SOC is optimizing for — and who decides when that changes

Source notes.

The claims and examples in this edition are drawn from the original Sovera manuscript. The manuscript itself cites IBM, Netenrich, Tines, CloudZero, Latitude, Terence Tao and Anton Chuvakin in specific places.

Editorial note. Point-in-time statistics and model pricing should be rechecked against the underlying source immediately before publication. The original manuscript contains those attributions; this edition does not add new external research.

The design direction favours declarative headlines, short paragraphs, strong page-level ideas, selective dark dividers, generous white space, and practical material framed as decisions rather than dense worksheet grids.

ABOUT SOVERA SECURITY

Sovera Security is an AI-native, sovereign MDR platform built from the ground up for European enterprise environments. Headquartered in Copenhagen, Sovera combines agentic AI-driven detection and response with 24/7 analyst coverage, deep GDPR and NIS2 expertise, and European data residency by design. No retrofitted legacy tooling. No dependence on non-European infrastructure.

Ready to find out what your MDR is missing?

Sign up for a free assessment at **sovera.security**. Our founding team will map your current stack against modern threat patterns, identify where your detection-to-response loop breaks down, and show you exactly what needs to change.

**The objective is not to automate
the old SOC faster.**

**It is to redesign where work, authority,
evidence and accountability sit.**

Five shifts. One operating model.

SOVERA SECURITY

sovera.security

SOVERA SECURITY • SOVERA.SECURITY