



## **REVIEW OF SYSTEM DESIGN APPROACHES IN AGENTIC ARTIFICIAL INTELLIGENCE FOR SECURITY OPERATIONS CENTERS**

S.M. Zakiryanov<sup>1\*</sup>

<sup>1</sup>TSARKA, Astana, Kazakhstan

\*Corresponding author: sherkhan.z@tsarka.com

### **Abstract**

Security operations centers increasingly rely on agentic artificial intelligence systems, which combine large language model reasoning with tool invocation and multi-step planning, to automate alert triage, investigation, and response. Independent research groups have proposed structurally distinct architectures for this purpose, ranging from single-agent reasoning loops to multi-agent systems with shared context or deterministic safety constraints, but no publication has systematically compared these designs against a common set of criteria. This review addresses that gap by constructing a five-dimension classification framework covering agent cardinality, coordination mechanism, autonomy boundary, safety enforcement layer, and validation maturity. A corpus of published systems relevant to security operations was retrieved from academic preprint archives and conference proceedings and coded against this framework. Five recurring design patterns were identified, together with the available quantitative evidence comparing them, including a controlled trial showing that multi-agent orchestration produced consistently reliable recommendations while single-agent approaches did not. The analysis found that claimed autonomy consistently outpaced empirical validation across the reviewed designs, with most systems evaluated no further than a single organizational deployment despite permitting autonomous action. These findings indicate that validation maturity should be assessed independently from architectural sophistication when organizations evaluate agentic artificial intelligence for security operations, and they identify calibration and human oversight effectiveness as priority areas for future empirical study.

**Keywords:** *agentic artificial intelligence, security operations center, multi-agent systems, autonomy boundary, validation maturity.*

### **1. Introduction**

Security Operations Centers (SOCs) function as the primary human-machine interface between raw telemetry and organizational risk decisions. Alert volume, tool fragmentation, and multi-stage attack progression have historically constrained the throughput of rule-based correlation and human triage alike. Recent work reports that enterprise SOCs process alert volumes exceeding 100,000 per day, with analysts allocating two to four hours to manual triage while a substantial proportion of alerts remain uninvestigated [1]. This operational pressure has



motivated a distinct line of research proposing agentic artificial intelligence (systems combining large language model, or LLM, reasoning with tool invocation, memory, and multi-step planning) as a mechanism for automating triage, investigation, and response functions within the SOC.

Unlike deterministic Security Orchestration, Automation, and Response (SOAR) playbooks, agentic systems are characterized by context-dependent reasoning that adapts the system's investigative path to evidence encountered during execution rather than following a fixed decision tree. This adaptability has led to markedly different design choices across proposed systems, ranging from single-agent reasoning loops to multi-agent orchestration frameworks.

Existing proposals can be provisionally grouped by architectural structure. At the simplest end, Reason-Act (ReAct) agents unify previously disjoint analyst tooling, including SIEM queries, OSINT lookups, and internal enrichment services, into a single automated reasoning loop. One such system was evaluated directly against manual analyst workflows on real SOC tickets, representing one of the few designs validated against operational data rather than synthetic scenarios [2]. A second class of proposals introduces multi-layer internal structure within a single agent: Roy and Singh [1] describe a framework consolidating perception, anticipatory reasoning, and risk-based action planning into one operational loop, evaluated conceptually within an enterprise environment and via proof-of-concept on a public authentication dataset.

A third and increasingly prevalent class distributes reasoning across multiple cooperating agents. Design choices within this class vary considerably: some architectures decompose reasoning by role and pass a shared, mutable context object between agents rather than relying on strictly sequential handoff [6], while others impose a deterministic verification layer, combining static analysis with LLM refinement, to constrain autonomous action within a formally specified safety envelope, reporting zero false positives on clean controls in a seeded vulnerability benchmark [5]. A fourth class addresses institutional constraints directly, proposing organization-scoped runtime architectures in which agent actions are bound to role permissions, data classification, and regulatory obligation rather than governed through prompt instructions alone [4].

Despite these structural differences, the four classes described above share an underlying reasoning cycle, illustrated in Figure 1.

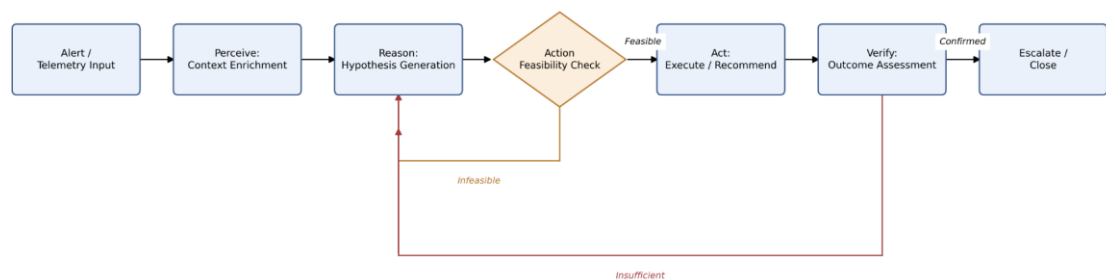


*Figure 1. Generic agentic reasoning loop*

Critically, comparative evidence between these classes remains sparse. One controlled study directly contrasting single-agent and multi-agent architectures for incident-response decision support, using 348 trials on identical scenarios, found multi-agent orchestration produced actionable recommendations in 100% of trials versus 1.7% for single-agent approaches, with zero variance in output quality, while both architectures exhibited comparable latency. This indicates that the observed benefit was decision consistency rather than processing speed [7]. Beyond SOC-specific contexts, broader surveys of multi-agent LLM systems catalog architectural approaches to task decomposition without converging on a standard classification scheme applicable to security operations specifically [9]. Separately, literature on automation bias in human-AI collaboration establishes that overreliance on automated recommendations is a recurring pattern across high-stakes domains, independent of the underlying model's measured accuracy [10], a finding with direct implications for autonomy-boundary design choices in SOC deployments that has not yet been discussed alongside the architectural literature reviewed above.

Two limitations recur across this literature. First, validation methodology is inconsistent: designs range from purely conceptual proposals to single-organization proofs-of-concept to the single controlled multi-trial comparison identified above, with no shared benchmark permitting direct cross-study comparison. Second, calibration, defined here as whether an agent's confidence in a classification corresponds to its actual reliability when granted execution authority, is addressed by only one identified study, which explicitly distinguishes benchmark capability from operational calibration under adversarial conditions [3].

No existing publication systematically classifies the architectural design space for agentic AI in security operations against a common set of dimensions, nor consolidates the fragmented empirical evidence comparing these designs. Practitioners evaluating architectural options for SOC deployment currently must independently locate and interpret findings distributed across a minimum of eight disparate sources spanning multiple venues and adjacent application domains.



This paper addresses that gap through the following objectives: (1) to construct a classification framework spanning agent cardinality, coordination mechanism, autonomy boundary, safety enforcement layer, and validation maturity; (2) to map the identified literature onto this framework and identify recurring design patterns; (3) to consolidate available comparative evidence, including the controlled trial identified above; and (4) to characterize open research challenges, with particular attention to the relationship between claimed autonomy and empirical validation maturity observed across the reviewed corpus.

## **2. Material and research methods.**

Records were retrieved from arXiv, restricted to the categories cs.CR, cs.AI, and cs.MA, and from the proceedings of the NDSS Symposium, covering the period from 2025 through the first half of 2026. Search queries combined the terms "agentic AI," "LLM agent," and "multi-agent" with "security operations," "incident response," and "SOC." Reference lists of the retrieved papers were examined in turn to identify additional relevant sources, a method known as backward snowballing. A source was included in the corpus if it described a system architecture in which an artificial intelligence component performed multi-step reasoning, invoked external tools, or coordinated with other autonomous components in a security-relevant context. Because published work specific to SOC deployment remained limited at the time of the search, the corpus was extended to closely adjacent domains, including regulated cybersecurity operations, embedded and automotive security, and incident response decision support. This extension reflected a deliberate scope decision rather than an oversight, since the same architectural patterns recurred across these adjacent domains. Sources limited to rule-based automation without adaptive reasoning were excluded from the corpus, as were sources without an identifiable technical architecture.

Each source in the corpus was coded against five dimensions, presented in Table 1. Agent cardinality distinguished single-agent designs from multi-agent designs. Coordination mechanism described how information moved between components, categorized as sequential handoff, shared persistent context, or hierarchical orchestration. Autonomy boundary described the extent of independent action permitted to the system, ranging from recommendation only to guarded autonomous action to unrestricted autonomous execution. Safety enforcement layer recorded whether the design relied on no explicit safety mechanism, a deterministic rule-based floor, or a runtime verification oracle. Validation maturity classified the strength of evidence reported for each system, ranging from a conceptual proposal to a single-organization proof of concept to a controlled comparative trial involving multiple runs and a defined evaluation metric.

*Table 1. Classification framework applied to the reviewed corpus*

| Dimension                | Categories                                                                              |
|--------------------------|-----------------------------------------------------------------------------------------|
| Agent cardinality        | Single-agent, multi-agent                                                               |
| Coordination mechanism   | Sequential handoff, shared persistent context, hierarchical orchestration               |
| Autonomy boundary        | Recommendation only, guarded autonomous action, unrestricted autonomous execution       |
| Safety enforcement layer | None, deterministic rule-based floor, runtime verification oracle                       |
| Validation maturity      | Conceptual proposal, single-organization proof of concept, controlled comparative trial |

Coded sources were grouped into design patterns where two or more sources shared the same combination of agent cardinality and coordination mechanism. Five such patterns emerged from the corpus, corresponding to the system designs reported by Roy and Singh [1], in the NDSS proceedings [2], by Fatouros et al. [4], by Petrovic et al. [5], and in the multi-agent risk management preprint [6]. Where a source reported quantitative outcomes from a controlled comparison, as in Drammeh [7], the reported figures were extracted directly rather than re-estimated. Findings on automation bias, drawn from Romeo and Conti [10], were treated as contextual evidence for the Discussion rather than coded within the five-dimension framework, since this source did not describe an agentic system architecture. No statistical aggregation was applied across the corpus given its limited size and the heterogeneity of reported metrics. Findings were instead organized according to the five dimensions and presented in tabular form in the Results and Discussion section.

### 3. The results and discussion

Application of the five-dimension framework to the corpus surfaced five recurring design patterns, which cluster naturally around a single distinction: whether reasoning is concentrated in one agent or distributed across several.

Among single-agent designs, complexity varies by how much internal structure is imposed on the reasoning loop. The tooling-unification system evaluated at NDSS [2] keeps this loop flat: a Reason-Act cycle that unifies previously separate analyst tools into one automated sequence, tested directly against real SOC tickets and giving it the strongest operational grounding of any design in the corpus. AgentSOC [1] instead breaks the loop into internal stages, separating perception, hypothesis generation, and action planning within the same agent rather than leaving them implicit in a single reasoning pass. The added structure did not extend to empirical rigor, however: AgentSOC's evaluation remained conceptual, supplemented only by a proof of



concept on a public authentication dataset, whereas the flatter NDSS design was tested against real operational tickets.

Multi-agent designs diverge along a different axis: how agents share information, and how tightly their actions are constrained. The risk management pipeline [6] lets agents read from and write to a shared, mutable context object rather than passing output sequentially, so that later agents can draw on conclusions several stages upstream. This richer context-sharing came at a measurable cost: the full six-agent pipeline failed all 30 attempts on hardware with a constrained context window, while a single fine-tuned model completed every run, a result that complicates any assumption that more agents and richer coordination straightforwardly improve reliability. CyberLLM [5] takes a more constrained approach to multi-agent coordination, pairing a fixed rule-based detection layer with an LLM refinement stage and adding a runtime oracle that checks every proposed action before execution, an architecture that trades the shared-context pipeline's flexibility for an explicit safety floor.

The fifth pattern addresses a concern the other four do not: institutional accountability rather than coordination mechanism. The regulated operations design proposed by Fatouros et al. [4] ties agent permissions to organizational role, data classification, and regulatory obligation through a structured security context, rather than relying on prompt instructions to constrain behavior. Unlike the other four patterns, its contribution lies less in how agents reason or coordinate and more in how their actions are authorized within an organization, a dimension the corpus's other designs leave largely unaddressed.

Direct quantitative evidence was available for three of the five patterns. Drammeh [7] reported 348 controlled trials comparing a single-agent copilot against a multi-agent system on identical incident response scenarios. Multi-agent orchestration produced an actionable recommendation in every trial, compared with 1.7 percent of trials for the single-agent condition, and showed no variation in output quality across trials where the single-agent condition did not. Both architectures completed their reasoning in approximately 40 seconds, indicating that the measured difference concerned the reliability of the output rather than processing speed. The safety envelope design in [5] was evaluated on a benchmark of 47 seeded vulnerabilities across nine automotive modules. A deterministic rule-based layer alone identified 34 percent of labeled vulnerabilities at full precision, and adding the LLM refinement stage raised coverage to approximately 70 percent with an F1 score of 0.83 while producing no false positives on clean control samples. AgentSOC [1] reported a processing time of 506 milliseconds on a public authentication dataset, a figure that described runtime performance under a single deployment condition without a comparison against an alternative architecture or a manual baseline. The remaining two patterns, the shared-context pipeline [6] and the organization-scoped runtime design [4], were reported without a comparable quantitative outcome, the former evaluated

through a cross-sector ablation study and the latter presented as an architectural proposal not yet tested empirically.

Across the five patterns, validation maturity concentrated at the lower end of the scale defined in Table 1. Four of the five representative sources reported evidence no stronger than a single-organization proof of concept, and one source offered only a conceptual proposal without empirical testing. Only the comparison reported by Drammeh [7] met the criteria for a controlled comparative trial. This distribution did not align with the autonomy boundary dimension in a favorable direction. Three of the five patterns permitted guarded autonomous action, meaning the system could execute a decision within a bounded safety mechanism rather than only recommend one, yet none of these three patterns had validation beyond a single organization. The pattern with the fewest restrictions on autonomy, single-agent reactive design, also carried the least formal safety mechanism of the five, relying on human review after the fact rather than a rule-based or verification layer.

A related finding came from OpenSec [3], which distinguished a system's classification accuracy on a benchmark from its behavior once given authority to act. The source reported that a model reaching high precision on alert classification in a controlled setting did not act with the same reliability once granted execution authority, in one case initiating containment on nearly all flagged alerts including a portion that should not have been escalated. This finding indicated that benchmark accuracy, the metric most commonly reported across the corpus, does not directly measure the property that matters once a system is permitted to act rather than only classify.

The concentration of low validation maturity alongside expanding autonomy claims raises a question addressed only indirectly within the corpus, concerning how human operators are expected to supervise these systems in practice. Literature on automation bias outside the cybersecurity domain offers relevant context. Romeo and Conti [10] found that overreliance on automated recommendations recurs across multiple high-stakes fields regardless of the underlying system's measured accuracy, and that this tendency persists even when users hold generally favorable views of the automated tool. Applied to the corpus reviewed here, this finding suggests that a security operations team supervising a guarded autonomous agent cannot assume that human review functions as an effective check simply because a human remains formally in the loop. None of the five identified patterns reported data on analyst behavior when reviewing agent-proposed actions, meaning the corpus currently offers no direct evidence on whether human oversight functions as intended once these systems are deployed.

Figure 2 summarizes the relationship between validation maturity and autonomy boundary across the five identified patterns. Four of the five patterns cluster at the lower end of validation maturity despite permitting guarded or greater autonomous action, while the single controlled



trial identified in the corpus, reported by Drammeh [7], occupies a separate position defined by its comparative design rather than by an autonomy claim.

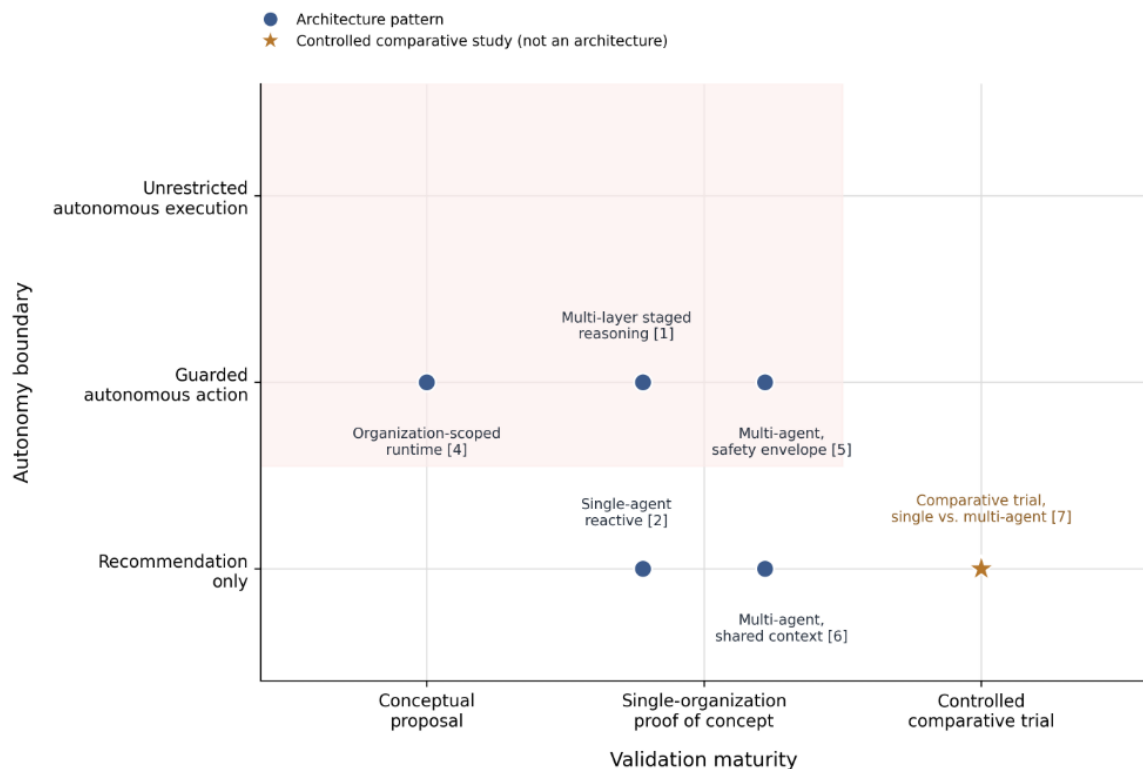


Figure 2. Validation maturity against autonomy boundary

#### 4. Conclusion

This review applied a five-dimension classification framework, covering agent cardinality, coordination mechanism, autonomy boundary, safety enforcement layer, and validation maturity, to a corpus of published agentic AI systems relevant to security operations. The analysis identified five recurring design patterns, ranging from single-agent reactive loops to multi-agent architectures with shared context or deterministic safety envelopes. Available quantitative evidence was consolidated, including a controlled trial comparing single-agent and multi-agent orchestration and a benchmark evaluation of a safety envelope design. The four objectives set out in the introduction were addressed through this process, and the resulting taxonomy and findings are presented in the results and discussion.

The main contribution of this work is the finding that claimed autonomy and empirical validation are not aligned across the reviewed corpus. Four of the five identified patterns permit or approach guarded autonomous action while reporting evidence no stronger than a single-organization proof of concept. This pattern had not been documented previously, since existing publications describe individual systems rather than compare designs against a shared set of dimensions. The classification framework introduced here offers a reusable structure for future



comparisons as the literature expands, and the identification of calibration, defined as the correspondence between a system's confidence and its actual reliability once given execution authority, as a distinct and underexamined property adds a consideration not captured by benchmark accuracy alone.

For practitioners evaluating agentic AI options for deployment in a security operations center, the findings indicate that validation maturity should be assessed independently from architectural sophistication or claimed capability. A system permitting autonomous action should not be treated as equivalent to one that has been tested under conditions resembling operational use. This review is limited by the size of the corpus, reflecting the recency of the literature, and by its reliance on published claims rather than independent replication of the reviewed systems. The scope extension to adjacent domains, disclosed in the methods section, means that conclusions drawn about security operations centers specifically rest partly on evidence from related but distinct application areas.

Two directions follow directly from the gaps identified. First, controlled comparative trials of the kind reported by Drammeh [7] remain rare, and additional studies of this design would allow the five identified patterns to be compared on a common basis rather than through separately reported metrics. Second, no reviewed source measured analyst behavior when reviewing or overriding agent-proposed actions, leaving the practical function of human oversight in guarded autonomous systems untested. Addressing this gap would connect the architectural literature reviewed here with the automation bias literature discussed in the results and discussion, and would supply the calibration evidence that current benchmark-based evaluations do not provide.

### References

1. Roy, J., & Singh, S.K. (2026). AgentSOC: A Multi-Layer Agentic AI Framework for Security Operations Automation. arXiv preprint arXiv:2604.20134.
2. Banstola, K., Al Faisal, F., & Ou, X. (2026). Experiences of Using Agentic AI to Fill Tooling Gaps in a Security Operations Center. NDSS Symposium, Workshop on Security Operations Center Operations and Construction (WOSOC).
3. Barnes, J. (2026). OpenSec: Measuring Incident Response Agent Calibration Under Adversarial Evidence. arXiv preprint arXiv:2601.21083.
4. Fatouros, G., Makridis, G., Kousiouris, G., Soldatos, J., & Kyriazis, D. (2026). An Organization-Scoped LLM Agent Runtime Architecture for Regulated Cybersecurity Operations. arXiv preprint arXiv:2605.30604.
5. Petrovic, N., Jeddou, O., Ben Fraj, F., Zolfaghari, V., Pan, F., Schamschurko, A., & Knoll, A. (2026). CyberLLM: A Multi-Agent LLM Framework for Autonomous Detection and Guarded Response in Automotive Cybersecurity. arXiv preprint arXiv:2608.06651.



6. Gupta, R., Kumar, S., Sharma, S., Dang, M., & Aggarwal, A. (2026). An Agentic Multi-Agent Architecture for Cybersecurity Risk Management. Preprint submitted to AICTC 2026 (Springer LNCS). arXiv preprint arXiv:2603.20131.
7. Drammeh, P. (2026). Multi-Agent LLM Orchestration Achieves Deterministic, High-Quality Decision Support for Incident Response. arXiv preprint arXiv:2511.15755.
8. Nguyen, T., Ndebugre, M., & Arremsetty, D. (2026). Security Considerations for Multi-agent Systems. arXiv preprint arXiv:2603.09002.
9. Vinay, V. (2025). The Evolution of Agentic AI in Cybersecurity: From Single LLM Reasoners to Multi-Agent Systems and Autonomous Pipelines. arXiv preprint arXiv:2512.06659.
10. Romeo, G., & Conti, D. (2025). Exploring automation bias in human-AI collaboration: a review and implications for explainable AI. AI & Society. <https://doi.org/10.1007/s00146-025-02422-7>

#### **Information about authors**

**Zakiryanov Sherkhan Merekeevich –**  
Higher Education, TSARKA GROUP, Astana,  
Kazakhstan  
e-mail: sherkhan.z@tsarka.com