



Autonomous Software License Intelligence: A Governance-Aware Agentic AI Framework for Enterprise Software Asset Management

Madhu Babu Chenna

Independent Researcher, USA. ORCID ID: <https://orcid.org/0009-0001-7030-8540>

Abstract

Enterprise software licensing practices are more complex when organizations mix on-premises and cloud-based applications and multiple Software as a Service (SaaS) subscriptions from multiple publishers and internal departments. Software Asset Management (SAM) best practices have traditionally relied on regular discovery, manual entitlement reconciliation, exception handling, and software management intervention at contract renewal. We propose the Autonomous Software License Intelligence (ASLI) framework: a governance-aware agentic AI framework for continuous discovery, use, compliance, optimization, and execution of software licenses. Anchored in Design Science Research (DSR), ASLI presents a prescriptive five-layer artifact including the layers of discovery and ingestion, license intelligence, agentic orchestration, compliance and risk, and executive reporting. ASLI employs retrieval-augmented generation techniques for contract and policy reasoning, multi-agent orchestration for complex disposition workflows, and a confidence-scored governance mechanism for escalations to humans, human approval/signoff, or autonomous execution. In order to quantify the artifact, five operational metrics have been defined: Software Utilization Variance Index (SVI), License Inefficiency Score (LIS), Estimated License Optimization Savings (ELOS), Autonomous Confidence Score (ACS), and License Drift and Fragmentation Index (LDFI). The paper also shows how third-party benchmarks of Flexera, Zylo, the FinOps Foundation, and Gartner can be used for index calibration, but does not report any production deployment experience with this approach. Thus, the paper's contribution to AI-enabled SAM is the evaluated architectural artifact and the measurement model. The framework additionally distinguishes between deterministic automation and agentic reasoning, and conditions under which AI systems may be safely shifted from recommendation to supervised or autonomous execution.

Keywords: Agentic AI, FinOps Governance, License Compliance Automation, Operational Intelligence, Software Asset Management

1. Introduction

Enterprise software license management has progressed from a knowledge domain of procurement and asset management into a world of governance decisions choosing a portfolio of on-premises or cloud-based products and services, Software as a Service (SaaS) subscriptions, enterprise agreements (EAs), and user or consumption-based licenses. Due to varying geographies and business units, entitlement records, usage telemetry, and contracts and procurement data, as well as financial management information may be stored in different technology tools. According to Flexera's 2024 State of ITAM report, 53% of the IT teams surveyed had difficulty being able to gain or maintain complete visibility into their technology investments. SaaS usage was less visible as compared to on-premises applications, hardware, and cloud instances.[1]

The economic impact of shadow SaaS is also meaningful. In its 2024 SaaS Management Index, SaaS and enterprise application management platform Zylo stated that the average organization used only 49% of its provisioned SaaS licenses, and that the average organization in its benchmark sample experienced approximately \$18 million in annual SaaS license waste [2]. While this is not an enterprise-wide baseline but rather an industry average, it indicates the size of the opportunity. The 2025 State of FinOps report found the FinOps practice continuing to expand from infrastructure to SaaS and licensing, with greater emphasis on cost predictability and value measurement across the technology portfolio [3].

The governance of applications is complicated by agentic AI. In 2026, Gartner predicted that by 2030, \$234 billion of enterprise application expenditure would be exposed to agentic arbitrage, where agents do the work across applications, undoing the traditional relationship between the number of software seats and value generated [4]. The upshot for license management is that if enterprise applications become increasingly accessed by autonomous agents on users' behalf, organizations will need to determine how many entitlements to procure and how to assign them, what they're used for, and how much they're worth to users and the business.

Standard SAM processes are discovery scanning, entitlement reconciliation, periodic utilization analysis, exception-handling, and renewal-cycle remediation. Required for compliance, these processes are often periodic and compartmentalized, which makes continuing and integrated process optimization difficult. Thus, the research question this paper addresses is not how to automate license provisioning, but how to

build enterprise AI to continuously sense license state, reason over contracts and organizational context, recommend or take disposition actions, and ensure human accountability when uncertainty and/or risk are present.

Its counterpart, Autonomous Software License Intelligence (ASLI), proposes a data-driven evaluation approach to software licenses as dynamic enterprise assets. To be classed as being in good (or poor) standing, a software license must be continually assessed against use, terms, business context, cost and governance. It comprises retrieval-augmented generation (RAG), multi-agent orchestration, deterministic workflow and confidence scored governance. It is more than a chatbot or a rules engine. It has separate sensing, reasoning, orchestrating, gating by risk and executive reporting components.

The paper presents four contributions. First, a five-layer agentic architecture for continuous software license intelligence. And second, it distinguishes agentic reasoning from classical rule-based automation. It describes where each type of reasoning is appropriate. Third, this paper identifies five operational indices with mathematical formulae for measuring SAM intelligence maturity. Fourth, it introduces a governance gate for autonomous AI actions where AI confidence is a necessary but not sufficient condition, with policy, business risk, and reversibility as additional requirements.

The paper is structured as follows: related work, Design Science Research methodology, ASLI architecture and decision model, evaluation against external benchmarks, and theoretical and practical implications and future lines of empirical research.

2. Literature Review

2.1. Software Asset Management and ITAM

Historically, SAM has focused on discovery, entitlement tracking, license reconciliation, preparing for an audit, and realizing savings. Industry research still shows that visibility remains an active challenge with respect to managing SaaS. In contrast, only 54% of businesses in the Flexera 2024 State of the Cloud report felt they had the required visibility across SaaS usage. This is important because organizations cannot effectively optimize a license portfolio without the ability to reconcile usage and entitlement states within a given technology group.

The relationship between SAM and FinOps, as well as technology value management, is becoming more recognized. According to the FinOps Foundation survey in 2025, most respondents manage SaaS expenditure and half manage licensing, indicating the connection between cloud finance and software asset governance has become a reality [3]. ASLI extends this convergence by reframing license optimization as an operational intelligence problem, addressing usage, contracts, workflows, and financial decisions across the entire domain.

2.2. Retrieval-Augmented Generation for Licensing Intelligence

Large language models appear to be good at language understanding but may not reliably provide information on current enterprise policy or contracts. Retrieval-augmented generation improves the reasoning of large language models with retrievable knowledge [5]. The original RAG proposed to combine non-parametric memory with generative models to achieve knowledge-intensive capabilities alongside an explicit retrieval mechanism [5].

RAG can act as a mechanism to reason over vendor specific licensing terms, a company's own licensing policies, exceptions, renewals, and past decisions. Within ASLI, RAG is considered a knowledge service and not a separate decision-making entity. Such evidence is one part of a decision process that includes evaluating telemetry, organizational context, and policy and risk.

2.3. Multi-Agent AI and Agentic Orchestration

Multi-agent LLM architectures can be used to decompose intricate enterprise tasks where capabilities are assigned to different agents. Another example is AutoGen, which shows agent communication, tool usage, human intervention, and management of complex tasks [6]. The model is useful to SAM because license governance is typically comprised of discovering and validating discovery data, reconciling entitlements, interpreting contracts and usage, managing exceptions, assessing financial impact, and orchestrating remediation.

ASLI therefore separates specialist responsibility rather than attributing all reasoning to a single agent to improve traceability and create governance boundaries around tool use. It also enables deterministic rules to be used when they are more reliable than generative reasoning, such as checking whether a known license count exceeds an entitlement quantity.

2.4. Explainability, Trust, and AI Governance

Arrieta et al. argue that explainability is important for the use of autonomous decision systems in enterprises; the field of explainability consists of varieties of transparency and interpretability enabling responsible AI development [7]. NIST AI RMF 1.0 also centers on governance, measurement, and management of AI risk throughout the system lifecycle [8].

ASLI achieves these properties through evidence logs, confidence scores, policy gates, action history logs, and human escalation paths. The primary difference is that model confidence in an action is not treated as a sufficient basis for autonomy, since an action could be predicted with high confidence but be high impact, irreversibly damaging, or inconsistent with policy, for example.

2.5. Intelligent Automation and RPA

Rule-based automation and RPA are applicable to deterministic SAM processes, such as vendor data extraction, reconciliation report production, and standard notification sending. A recent systematic review of RPA identifies process selection, technology integration, governance, and continuous improvement as important dimensions of enterprise automation [9]. As such, ASLI uses both deterministic automation (for formalized, repeatable tasks) and agentic AI (for tasks with less formalization that require interpretation, contextual reasoning, or multi-step planning).

2.6. Research Gap

The literature reviewed found that the technologies needed for ASLI (RAG, multi-agent orchestration, explainable AI, intelligent automation and AI risk governance) have been shown to be feasible. However, neither this study nor the literature searched have identified a software-license architecture for integrating these technologies within a governance-aware lifecycle model. ASLI's final question, the architectural and measurement gap, is how to combine continuous license telemetry, contract and policy intelligence, agentic reasoning, deterministic execution, and confidence-based governance into an evaluable SAM artifact.

3. Methodology and ASLI Framework

3.1. Design Science Research Methodology

The ASLI framework is developed following the Design Science Research (DSR) approach. According to Peffers et al., DSR is a research methodology for the development and evaluation of prescriptive information systems artifacts and it consists of six activities: problem identification and motivation, solution objectives definition, design and development, demonstration, evaluation, and communication [10]. This approach is appropriate for this paper, because the contribution is an artifact: specifically a reference architecture and measurement model, rather than a causal theory of naturally occurring behavior.

Table 1. Design Science Research mapping for the ASLI artifact

DSR activity	ASLI implementation	Evidence / output
Problem identification	Fragmented entitlement data, underutilization, audit exposure, manual reconciliation	Industry benchmarks; problem statement
Objectives definition	Continuous license intelligence with governed automation	Design requirements
Design and development	Five-layer architecture plus measurement indices	ASLI artifact and decision model
Demonstration	Illustrative license disposition and renewal workflows	Scenario-based walkthroughs
Evaluation	Benchmark calibration and measurement validity analysis	Index definitions; external anchors
Communication	Research manuscript and reusable framework	This article

3.2. ASLI Five-Layer Architecture

The structure of ASLI is as a pipeline of intelligence, where each layer of functionality offers a specific function while normalizing data. This separation allows an organization to replace or upgrade an individual step without having to overhaul the entire governance structure at once.

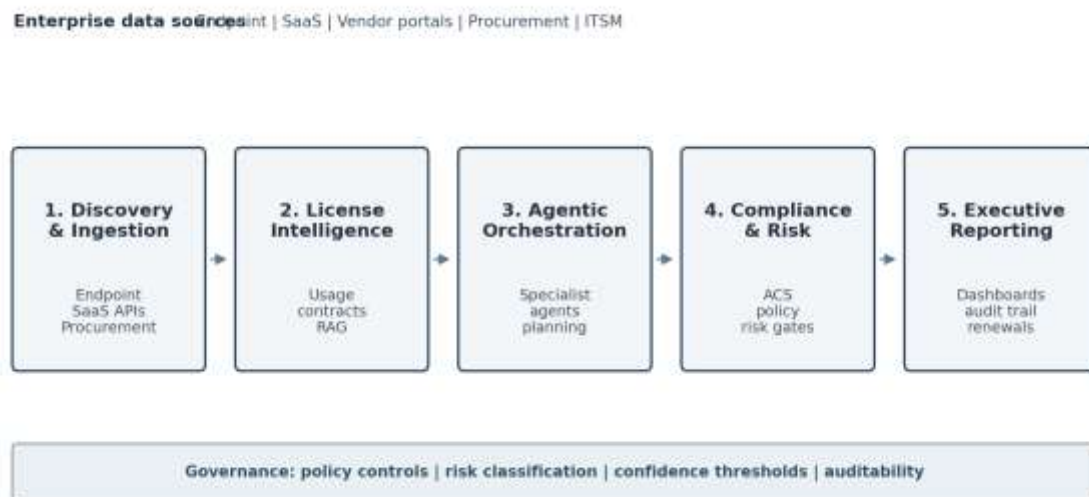


Fig. 1. ASLI five-layer agentic architecture. The governance layer evaluates each disposition recommendation before action.

Table 2. ASLI five-layer architecture summary

Layer	Primary function	Representative inputs	Representative outputs
1. Discovery and Ingestion	Continuously collect and normalize software and entitlement state	Endpoint telemetry; SaaS APIs; vendor portals; procurement records	Normalized entitlement and usage records
2. License Intelligence	Analyze utilization and retrieve contractual / policy context	Entitlements; usage data; contract terms; policies	Utilization signals; compliance flags; renewal signals
3. Agentic Orchestration	Coordinate specialist agents for reasoning and planning	Intelligence outputs; organizational context; policy rules	Disposition recommendations with rationale
4. Compliance and Risk	Evaluate risk, confidence, policy conformance, and autonomy eligibility	Recommendation; risk class; ACS; policy constraints	Escalation, approval, or autonomous execution path
5. Executive Reporting	Expose portfolio intelligence and maintain evidence history	Index values; outcomes; action logs	Dashboards; audit evidence; renewal briefings

3.3. Layer 1: Discovery and Ingestion

The Discovery and Ingestion layer forms the common data backbone of the rest of the architecture through the aggregation of endpoint discovery, application-launch or usage telemetry, SaaS provider APIs, enterprise identity, procurement, and contract metadata. The principal design requirement is normalization, whereby the disparate source data must be mapped to a common entitlement representation in terms of assignment, usage, cost, contractual quantity and lifecycle status.

3.4. Layer 2: License Intelligence

The License Intelligence layer supports analytics, such as utilization assessment, entitlement variance analysis, license tier alignment, and portfolio duplication analysis with RAG, and retrieves specific contract clauses, policy rules, exception conditions and past decisions, rather than providing a generic natural language answer from the knowledge base. Each recommendation should include the evidence on which it was based.

3.5. Layer 3: Agentic Orchestration

The Agentic Orchestration comprises specialized agents: the Discovery Validation Agent for data quality, the Entitlement Reconciliation Agent compares purchased and assigned objects, the Contract Intelligence

Agent extracts and understands applicable contractual language, the Optimization Agent identifies reclaim, reassignment, rightsizing or consolidation opportunities, and the Workflow Agent orchestrates downstream tools. An orchestrator agent divides portfolio level goals into sub-goals and combines resulting evidence from agents supporting the goals.

3.6. Layer 4: Compliance and Risk

The next layer then translates the recommendation into an action path, but ASLI uses AI confidence as an input rather than just autonomy. Autonomous action upon the recommendation is allowed if the confidence score exceeds an organizational threshold, the policy applies in the environment, the recommendation has an acceptable risk, and it is sufficiently reversible or is limited to the pre-determined level of risk.

3.7. Layer 5: Executive Reporting

The Executive Reporting layer describes the actual software portfolio situation and trend. It is also where utilization, inefficiency, optimization opportunity, confidence distribution, entitlement drift, and action history are exposed. It also provides renewal briefing on license demand, contractual due dates, optimization opportunities, evidence needed for negotiation plus audit trail of data sources, obtained knowledge, recommendation, governance path, action and outcome.

3.8. Deterministic Automation Versus Agentic Reasoning

Table 3. Automation capability split within ASLI

Activity	Best-fit mechanism	Rationale
License expiry notification	Deterministic workflow	Rule is explicit and repeatable
Entitlement count reconciliation	Deterministic analytics / rules	Structured comparison with known boundaries
Contract clause interpretation	RAG + LLM reasoning	Language and contextual interpretation required
Exception analysis	Agentic reasoning + policy retrieval	Context and prior decisions must be considered
License tier recommendation	Agentic reasoning + analytics	Usage must be interpreted against role and needs
Complex disposition planning	Multi-agent orchestration	Multiple systems and constraints must be coordinated
Autonomous execution	Workflow automation after governance gate	Execution should be deterministic once authorized

4. Operational Decision Model and Benchmark-Based Evaluation

4.1. Evaluation Boundary

As such, this evaluation does not report on the production rollout of ASLI over time, but rather its design and measurement approach, by (1) defining computable operational indices, (2) mapping them to externally specified enterprise benchmarks, and (3) showing how such measures could inform enterprise governance and prioritization structures. That said, external benchmarks are best thought of as calibration anchors rather than evidence that ASLI already reached these goals.

4.2. Software Utilization Variance Index

The Software Utilization Variance Index (SVI) is defined as the percentage of provisioned units not in use during the measurement window. Assuming P is the number of provisioned units, and U is the number of units satisfying the active-use criterion of the organization's usage policy. SVI is defined as:

$$SVI = \left(\frac{P - U}{P} \right) \times 100 \quad (1)$$

An SVI of 0% means no underutilization was found under the selected policy. The higher the SVI, the more potential for reclaiming or rightsizing unused capacity. Further, the active-use criterion should be application- and vendor-specific, so that, for example, a design application used quarterly may be considered active for that application.

4.3. License Inefficiency Score

The License Inefficiency Score (LIS) is the sum of redundancy, underutilization, and tier mismatch, each normalized to the range of 0 to 1. These structural inefficiencies are not explained by utilization alone. Weights can be configured per organization, totaling one:

The LIS can be expressed as a linear combination: $LIS = wR(R) + wU(U) + wM(M)$, where $wR + wU + wM = 1$ (2)

The higher the LIS, the more inefficient the portfolio is. Equal weighting may be used for an initial pilot, but the final weightings must reflect organizational priorities, vendor economics and the reversibility of each optimization action.

4.4. Estimated License Optimization Savings

ELOS (estimated license optimization savings) are the annualized financial opportunity provided that optimization actions are put in place. These should be based on action level, auditable estimates. They should not be based on industry averages. Thus, the annual avoided cost C_i for action i , after any transition or implementation cost, is:

$$\text{ELOS} = \text{Sum over } i \text{ of (Avoided annual license cost}_i \text{ - Action cost}_i \text{)} \quad (3)$$

The Zylo 2024 benchmark of approximately \$18m average annual SaaS license waste per organization is used only in this paper to provide understanding into external (i.e. third-party) magnitudes [2]. This should not be interpreted as a theoretical max or a predicted ASLI saving.

4.5. Autonomous Confidence Score

An Autonomous Confidence Score (ACS) quantifies how ready a recommendation is to be performed in a governed manner. It is based on the completeness (D), conformance (P) and adherence (A) to governance of the data. One calibrated version is:

$$\text{ACS} = wD(D) + wP(P) + wA(A), \text{ where } wD + wP + wA = 1 \quad (4)$$

The weights should be based on the historical decision and risk outcomes. The 60% and 85% thresholds in the illustrative routing model below are governance parameters, not empirically validated universal thresholds.

4.6. Autonomy Eligibility Gate

To each autonomous action, not just to actions with high AI confidence, we should consider Risk (the normalized business and contractual risk of the proposed action) and two binary variables called Policy and Rev (whether the proposed action complies with an organization's policy and with the organization's rules of reversibility/pre-authorization, respectively). Here is an illustrative autonomy gate:

$$\text{AE} = I[\text{ACS} \geq \tau \text{ and Risk} \leq \rho \text{ and Policy} = 1 \text{ and Rev} = 1] \quad (5)$$

AE equals 1 in all cases described above, while equaling 0 elsewhere, thereby separating absolute certainty in the model from permission to govern. The strength of the ASLI model, though, isn't the high confidence score but the ability to turn confidence and enterprise policy into a decision trail.

4.7. License Drift and Fragmentation Index

The License Drift and Fragmentation Index (LDFI) is the score indicating how far the observed entitlement posture is from its authorized baseline and how fragmented its underlying data sources are. Since drift and fragmentation are in different units, their values are normalized before being averaged together. A simple implementation is:

$$\text{LDFI} = wT(T) + wF(F) \text{ where } wT + wF = 1 \quad (6)$$

T is normalized entitlement drift for the observation period and F is normalized fragmentation across source systems. The index can provide a current-state measure and a descriptive measure of the trend.

Table 4. ASLI operational indices and evaluation interpretation

Index	What it measures	Primary use	External calibration evidence
SVI	Provisioned capacity not actively used	Reclamation / rightsizing	Zylo 2024 SaaS utilization benchmark [2]
LIS	Structural inefficiency across redundancy, utilization, and tiering	Portfolio optimization	Flexera 2024; Zylo 2024 [1], [2]
ELOS	Auditable annualized optimization opportunity	Financial prioritization	Zylo 2024; FinOps 2025 [2], [3]
ACS	Readiness for governed autonomous execution	Human vs. autonomous routing	Governance design; requires empirical calibration
LDFI	Entitlement drift and source fragmentation	Data quality and audit readiness	Flexera 2024 visibility evidence [1]

4.8. Benchmark-Based Evidence

In addition to the problem space described by ASLI, the 2023 Flexera State of the Cloud report found 53% of all surveyed IT teams had trouble maintaining complete visibility into technology investments, and 54%

had accurate visibility into their SaaS usage [1]. The 2023 Zylo State of SaaS report documented 49% usage of provisioned SaaS licenses, and an average annual enterprise waste of \$18 million [2]. FinOps Foundation reports show increasing organizational use of SaaS and licensing in FinOps practices [3]. These provide plausible external anchors for the ASLI indices but do not provide a direct and formal evaluation of the ASLI artifact.

As detailed in Gartner 2026's analysis of agentic arbitrage, up to \$234 billion of enterprise application spending could be exposed as agentic systems displace or bypass use of traditional application interactions by 2030 [4]. The implication is that seat-based license governance must evolve to outcome- and usage-aware governance, especially where agents do most of the work without human interaction with the interfaces.

5. Results Interpretation and Discussion

5.1. What the Benchmark Evidence Supports

Three conclusions can be drawn from the benchmark analysis. Firstly, because incomplete visibility happens often, it is worthwhile to keep discovery and entitlement records. Second, underutilized licenses and redundant software portfolios represent economic opportunities. Third, agentic AI requires governing software usage based on how work is actually done and how software is used, regardless of seats assigned or what licensing agreements dictate.

The evidence does not show, and ASLI cannot argue that they have reached a specific percentage of reduction for waste, audit exposure or administrative burden without the evidence being produced. A more academically accurate interpretation of ASLI is that it is a method that can be used to measure or drive such waste and burden reductions.

5.2. Theoretical Contribution: Confidence-Scored Governance

The strongest conceptual contribution is integrating confidence with policy and risk gates. Traditional workflow automation is suited for deterministic and known conditions and assumptions. Pure autonomous AI is impractical in case of ambiguity, irreversibility or importance. Under ASLI, an intermediate level of governance is used to recommend, ask for permission, or act based on calibrated evidence and organizational constraints.

Extending the practical concept of progressive automation, this becomes a model to quantify the decision architecture and applied to other domains beyond SAM, including cloud-cost optimization, vendor management, endpoint remediation and enterprise procurement and asset management. The contribution, therefore, is not whether 85% confidence is safe, but instead that the issues of model confidence and authorization to act should be viewed separately.

5.3. Theoretical Propositions

The model proposes four propositions for further empirical testing.

P1. Continuous entitlement and usage integration allows license optimization to happen sooner than periodic reconciliation.

P2. Using contracts and policies for contextual retrieval improves traceability of license disposition recommendations compared to using utilization-only rules.

P3. Multi-agent decomposition improves coverage of complex license lifecycle tasks requiring multiple enterprise systems and heterogeneous evidence across the license lifecycle.

P4. Confidence- and risk-gated autonomy can reduce manual governance effort, while allowing human oversight for materially uncertain or high-impact decisions.

5.4. Illustrative Autonomous Scenario

If, for example, a license for a premium application is assigned to an employee who has not used the application during the organization's defined grace period, the License Intelligence layer retrieves the utilization policy and contract terms, while the Business Context and Optimization agents determine whether that employee has a business role or active project to justify continued license access. Upon receiving the Notification Agent's proposal for license reclamation and exception path, the Workflow Agent, if the autonomy eligibility gate is satisfied, initiates the license reclamation, updates the entitlement record, records the estimated savings, and forwards the request to the measurement layer.

Thus, this example illustrates the need for a license management system to have both automation and reasoning. Removing a license is deterministic. Whether this is acceptable depends on the context of the situation.

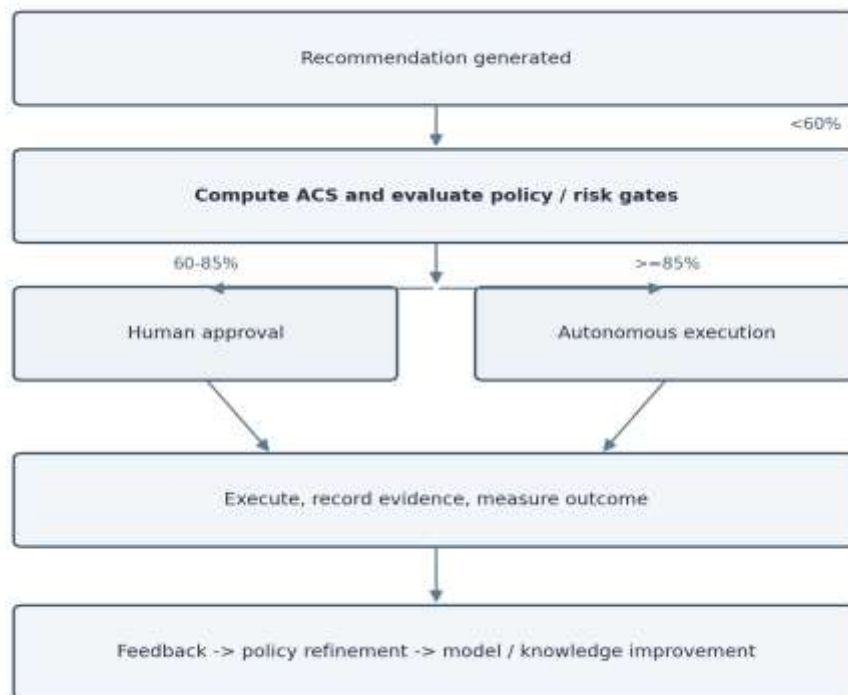


Fig. 2. Governance pathway from recommendation to autonomous execution.

5.5. Implications for Enterprise SAM

ASLI advocates continuous license intelligence instead of periodic compliance management so that SAM functions have more time to devote to policies, exceptions, measurement, vendor strategies, and AI-powered disposition, rather than manual license reconciliations. ASLI does not suggest that SAM professionals are replaced, but instead that their roles are redefined from transaction processor to governance and optimization expert.

6. Limitations and Future Research

6.1. Limitations

Some limitations of the study include that ASLI has not been experimentally tested in a longitudinal deployment, so the actual gains remain hypothetical. Second, the indices of active usage, redundancy, tier mismatch, risk, and reversibility must be defined by the organization using the indices. Third, licensing contracts differ considerably among vendors, which may limit the portability of an individual decision policy. Fourth, RAG quality depends on the quality of the current authoritative contract and policy documents. Fifth, multi-agent systems differ in terms of latency, cost, tool use, and reliability.

6.2. Future Research

Next, controlled empirical evaluation of a matched ASLI-enabled investment portfolio and a manual SAM control, across multiple renewal periods, should report on SVI, LIS, ELOS, ACS outcomes, LDFI, administrative effort and compliance exceptions in the overall investment portfolio over time. Experimental designs for evaluation should consider alternative thresholds for autonomy, and whether these threshold policies differ by the risk of the action.

A second research direction is adaptive policy calibration, using past approvals, rejections, reversals, and audits to learn organization-specific weights and thresholds while joining every policy to governance constraints. A third direction is multi-modal contract intelligence that combines text, tables, procurement data, and entitlement telemetry to comprehensively understand the value of contracts. A fourth direction is FinOps and technology-value-management practices that assess software, cloud, and agentic consumption through a common economic lens.

7. Conclusion

ASLI interprets software asset management as a continuous enterprise intelligence problem and seeks innovative and practical combinations of discovery, knowledge retrieval, multi-agent reasoning, governance, and workflow execution that move away from periodic license reconciliation towards continuous explainable optimization.

The five-layer framework has five computable operational indicators: SVI, LIS, ELOS, ACS, and LDFI. These help to evaluate whether the level of autonomy provided to an AI is appropriate to the circumstances. To determine eligibility, autonomy gates use inputs such as AI confidence, policy compliance, business impact and reversibility. This is an important distinction because safe enterprise autonomy is more about the organization authorizing its AI systems to act than about its confidence level.

The benchmark analysis shows the relevance of the problem, but it does not prove ASLI's empirical validity. The next step should be to experimentally evaluate and calibrate ASLI across different vendor, organizational, and regulatory environments. If ASLI's implementation is validated in practice, it could be a reusable governance framework for software licenses in a world where agentic AI is impacting the enterprise software market.

Ethics Statement

This research presents a design artifact and benchmark-based evaluation approach and does not involve human participants, animals, or collection of personal data for the purposes of this study.

Declarations

Conflict of Interest: The author declares no conflict of interest with respect to the research, authorship, or publication of this article.

Human and Animal Rights: Not applicable.

Informed Consent: Not applicable.

AI Tool Disclosure: AI-assisted writing and document-production tools were used during manuscript revision. The author retains responsibility for the conceptual framework, interpretation, factual verification, and final manuscript content.

Funding: No external funding was received for this research.

Data Availability: The benchmark figures discussed in this manuscript are derived from publicly available industry reports and academic sources cited below. No proprietary enterprise data are reported.

References

- [1] Flexera, 2024 State of ITAM Report, Flexera, 2024. Available: <https://www.flexera.com/about-us/press-center/flexera-2024-state-of-itam-report-finds-software-audit-costs-continue-to-rise>
- [2] Zylo, SaaS Management Index 2024: Utilization, Waste, and Optimization Benchmarks, Zylo, 2024. Available: <https://zylo.com/resources/saas-management-index>
- [3] FinOps Foundation, State of FinOps 2025 Report, FinOps Foundation, 2025. Available: <https://data.finops.org/2025-report/>
- [4] Gartner, "Gartner Says \$234 Billion in Enterprise Application Software Spend Is at Risk from Agentic AI," July 1, 2026. Available: <https://www.gartner.com/en/newsroom/press-releases/2026-07-01-gartner-says-us-dollars-234-billion-in-enterprise-application-software-spend-is-at-risk-from-agentic-artificial-intelligence>
- [5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [6] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," in *Proceedings of COLM*, 2024. arXiv:2308.08155, 2023.
- [7] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82-115, 2020, doi: 10.1016/j.inffus.2019.12.012.
- [8] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.
- [9] S. Khantong and P. Sriboonlue, "Robotic Process Automation in Business Process Management: A Systematic Literature Review and an Integrated Framework," *Technologies*, vol. 14, no. 4, p. 225, 2026, doi: 10.3390/technologies14040225.
- [10] K. Peppers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007, doi: 10.2753/MIS0742-1222240302.