

Governed Deployment of Enterprise Artificial Intelligence Control Planes for AI-Assisted Modernization Services: A Systematic Review

Nithesh Gudipuri

Submitted:01/10/2024

Revised:11/11/2024

Accepted:20/11/2024

Abstract: We have seen the growth of artificial intelligence (AI) assisted modernization services such as automated code modernization, legacy migration tooling and generative coding assistants, but the governance processes to put such services into practice safely and at scale have not kept up. The systematic review collates nineteen resources relating to cloud computing standards, zero trust security architecture, machine learning operations (MLOps), algorithmic accountability and AI risk management, to define the architecture and governance of an enterprise “AI control plane” – a single layer for policy enforcement, access control, and auditing, located between AI-assisted tooling and production systems. To uncover common constructs and tensions, thematic synthesis analysis was performed in four domains of infrastructure and access-control architecture, deployment automation, accountability and explainability, and risk-management frameworks. The results suggest that there is common consensus on the need for three elements: continuous verification, least-privilege access, and reproducible deployment pipelines, but no single element is enough to enable governed AI deployment. A composite governance-risk formulation is suggested to demonstrate how different risk signals found in the literature could be combined in a control-plane gating mechanism. There is also evidence that tools used for coding with AI have measurable productivity impacts, highlighting the operational implications of governance design. The review finds that none of the reviewed frameworks consider infrastructure, deployment and accountability issues simultaneously, creating an opportunity for an integrated control-plane architecture to fill this gap.

Keywords: *artificial intelligence governance, control plane, MLOps, zero trust architecture, algorithmic accountability, cloud computing, AI risk management, legacy modernization, explainability, DevOps*

1. Introduction

Artificial intelligence is being integrated into the workflows of enterprise modernization programs that previously only human engineers could manage, such as legacy code translation, infrastructure provisioning and continuous integration and delivery (CI/CD) pipelines. These AI-driven capabilities will move from pilot projects to production-ready enterprise services, bringing governance requirements that are remarkably similar to those of previous cloud and DevOps adoption experiences, but are also unique to probabilistic, data-dependent systems (Kreuzberger et al., 2023; Sculley et al., 2015). The idea of an enterprise “AI control plane” has been introduced

into practitioner and standards conversation as a potential architectural solution: a centralized orchestration and policy-enforcement layer between AI-enabled tools and the enterprise's downstream systems that provides authentication, authorization, auditing and risk gating.

There are three reasons for such a layer. Firstly, AI modernisation tools such as large language model (LLM) code generation are shown to be reliable though not foolproof, and require verification before their changes are deployed into production (Chen et al., 2021). Second, there are invisible technical debt incurred by machine learning systems (Sculley et al., 2015), such as coupling between components and changing data requirements. Third, expectations regarding accountability, explainability, and fairness are now making their way into regulatory and organizational requirements for AI-assisted decision support that is

*Independent Researcher - Enterprise Services,
Enterprise AI and Architecture, FinTech, Platform
Modernization*

integrated into enterprise workflows (Tabassi, 2023; Busuioc, 2021). The purpose of this review is to synthesize evidence from four streams of literature – cloud and zero trust infrastructure, MLOps and deployment automation, algorithmic accountability and explainability, and AI risk management – to describe what a governed AI control plane should offer, as well as the gaps that remain unfilled by the individual reviewed frameworks.

2. Method

2.1 Eligibility Criteria

Sources had to satisfy the following criteria: (a) published in a peer-reviewed research journal, an official standards publication or in a widely cited preprint reporting original empirical work, (b) related to the deployment of cloud computing infrastructure, DevOps or MLOps, zero trust or access-control architectures, algorithmic accountability/explainability, or AI risk management, and (c) published between 2011 and

2023. Sources that discussed an unrelated topic or that discussed AI governance but didn't refer to technical architecture were not included.

2.2 Search Strategy and Study Selection

The reviewed corpus consisted of nineteen sources from the computing and information systems venues, such as Association for Computing Machinery (ACM) digital library, IEEE Xplore, arXiv and National Institute of Standards and Technology (NIST) special publications. The venues were chosen to represent empirical software engineering research (such as migration surveys and DevOps and MLOps research) as well as normative, standards, or standards-oriented aspects, given the technical and governance nature of the research question. Screening was done by title and abstract review then full-text review based on the above-mentioned eligibility criteria. The identification, screening, and thematic allocation process is summarized in Figure 1.

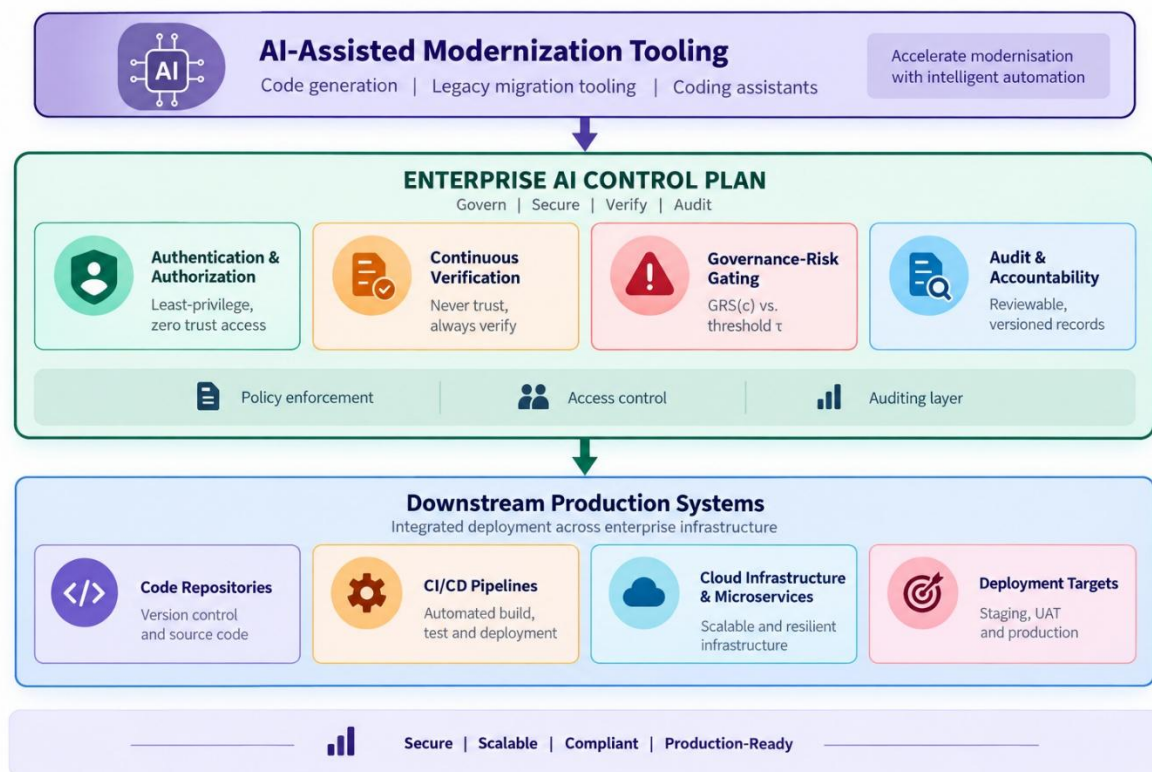


Figure 1. Position of the enterprise AI control plane as a governance layer between AI-assisted modernization tooling and downstream production systems, showing its four core functions.

2.3 Data Extraction and Synthesis

The following information was collected for each of the included sources, if applicable: research domain, methodological approach, main architectural or governance construct proposed, and

quantitative findings reported. Thematic synthesis was used, as commonly done in previous secondary studies in software engineering (Leite et al., 2019), with the constructs extracted and organized in a higher-order synthesis in terms of the four domains

presented in Section 1. The distribution of the corpus used in the study over these themes is summarized in Table 1 and displayed graphically in Figure 2.

Table 1 Thematic Distribution of Reviewed Sources

Theme	Representative Sources	n	Primary Construct
Infrastructure and access-control architecture	Mell & Grance (2011); Rose et al. (2020); Chandramouli & Butcher (2023)	3	Cloud service models; zero trust tenets; multi-location access control
Deployment automation (DevOps/MLOps)	Leite et al. (2019); Garg et al. (2021); Kreuzberger et al. (2023); Lwakatare et al. (2020); Sculley et al. (2015)	5	CI/CD pipelines; reference MLOps architecture; ML-specific technical debt
Legacy and cloud migration	Gholami et al. (2017); Di Francesco et al. (2018)	2	Migration drivers, barriers, and decomposition strategy
Accountability and explainability	Ribeiro et al. (2016); Doshi-Velez & Kim (2017); Busuioc (2021); Cobbe et al. (2021); Kordzadeh & Ghasemaghahi (2022); Konigstorfer & Thalmann (2022); Costanza-Chock et al. (2022)	7	Model-agnostic explanation; interpretability taxonomy; reviewability; documentation; audit ecosystems
Risk management and productivity evidence	Tabassi (2023); Chen et al. (2021); Peng et al. (2023)	3	AI risk management functions; code-generation benchmarking; developer productivity

Note. n represents the count of sources primarily associated with the theme; several sources contributed to more than one theme.

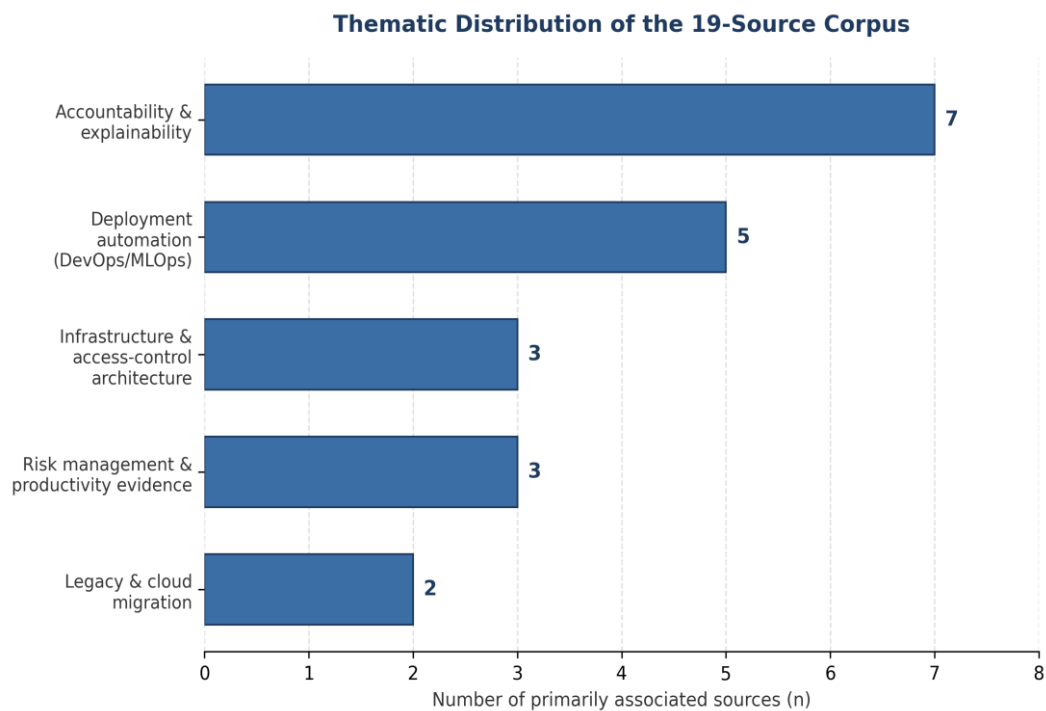


Figure 2. Thematic distribution of the 19-source corpus across the five constructs summarized in Table 1.

3. Results

3.1 Infrastructure and Access-Control Foundations

The fundamental definition of cloud computing has five key features: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service, as well as three service models and four deployment models, which continue to form the core of enterprise infrastructure classification (Mell & Grance, 2011). An AI control plane, therefore, is best thought of as a governance layer on top of this cloud service stack, or on top of a set of APIs and tools that abstract access to the service stack. From this point on, zero trust architecture shifts the focus of network security from “trust anyone on the network” to “never trust, always verify,” meaning that the network must

always be checked and resources must be micro-segmented and assigned the least privileges possible (Rose et al., 2020). The model was later extended to accommodate access-control models for cloud-native, multi-location architectures where access requests need to be evaluated dynamically across distributed microservice boundaries in the context of service meshes and container-orchestration contexts (Chandramouli & Butcher, 2023). For an AI control plane, this is especially salient, as AI-assisted modernization tooling often accesses numerous distributed repositories, build systems, and deployment targets, with every interaction with a production resource being an individual access event that needs to be assessed. Figure 3 presents the resulting reference architecture, positioning the control plane between AI-assisted tooling and the underlying cloud service stack.

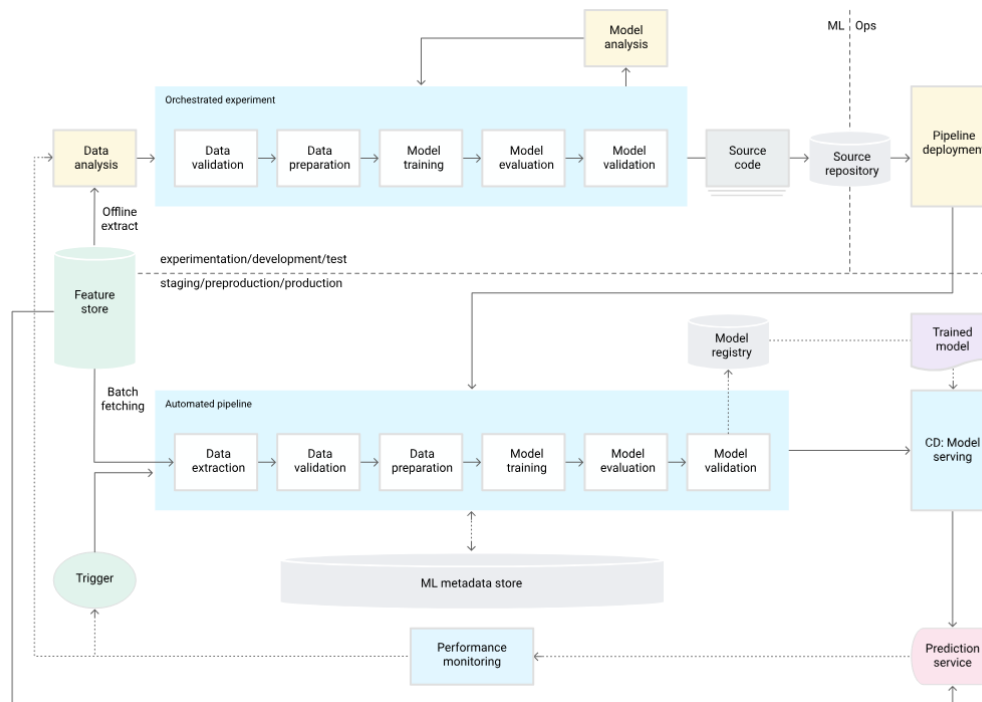


Figure 3. Continuous delivery and automation pipelines in machine learning (GoogleCloud,2021)

3.2 Deployment Automation, DevOps, and MLOps

Systematic review of DevOps literature has identified persistent organizational challenges of cultural resistance to DevOps, toolchain fragmentation and lack of ability at scale to operationalize continuous delivery (Leite et al., 2019), as well as the practice defined by collaborative culture, automation, measurement and continuous improvement. On top of these DevOps layers, machine learning adds another layer of

complexity. A popular analysis of technical debt in machine-learning systems found that software-related debt sources were less likely to occur than other types of debt, such as entanglement between model components (which can be summarized as the rule that "Changing anything changes everything"), unstable data reliance, feedback loops between deployed models and the data they feed upon, and configuration debt due to the growth of tunable parameters (Sculley et al., 2015). The results prompted specific MLOps architectures, which are

summarized in a reference architecture that merges existing MLOps literature (Kreuzberger et al., 2023): data management, experimentation, and production serving components must all be part of an end-to-end architecture that embraces principles such as versioning of data, code, and models; automated training and deployment pipelines; monitoring of deployed models; and experimentation reproducibility. There are also specific proposals for automated machine learning model deployment that define continuous integration and continuous delivery pipelines with steps of model validation, staged rollout and rollback (Garg et al., 2021).

Theory and architecture literature are supported by the industry evidence. When studying large-scale machine learning systems in real-world industrial environments, certain problems were found to be common in data management, deployment, monitoring, and organizational alignment, and a maturity series was suggested that would guide and propel organizations from ad hoc experimentation towards systematic and automated machine learning operations (Lwakatare et al., 2020). The constructs in the literature related to deployment automation that are relevant to governance are summarized in Table 2, which maps them to the control-plane functions they inform.

Table 2 Deployment-Automation Constructs Mapped to Control-Plane Functions

Source	Core Construct	Control-Plane Function Informed
Sculley et al. (2015)	ML-specific technical debt (entanglement, data dependency, configuration debt)	Change-impact risk scoring
Kreuzberger et al. (2023)	End-to-end MLOps reference architecture	Pipeline orchestration and versioning
Garg et al. (2021)	CI/CD pipeline for automated ML deployment	Staged rollout and rollback gating
Lwakatare et al. (2020)	Industrial ML maturity stages	Governance maturity benchmarking
Leite et al. (2019)	DevOps collaboration and automation dimensions	Cross-team policy coordination

3.3 Legacy and Cloud Migration Evidence

In the context of legacy system modernization, AI-based modernization services are often applied to the specific challenge of migration from legacy systems, thus directly relevant to control-plane design is empirical migration research. In a survey with industry practitioners that are moving towards microservice architectures, the key drivers found were modularity, independent scalability, and deployment flexibility, while the key challenges were service decomposition strategy, data consistency across service boundaries, and organizational readiness (Di Francesco et al., 2018). Another empirical study on legacy-to-cloud migration revealed that data migration integrity risk, vendor lock-in risk, lack of staff expertise in both legacy and cloud and accumulated technical debt in legacy codebases are the challenges encountered in

the migration process (Gholami et al., 2017). The results show that such tooling for system modernization using AI is generally used on the very systems that are most likely to cause unintended changes in the system, which serves to illustrate the need for an AI governance layer that can be used to limit automated changes to the riskiest systems, regardless of whether the individual AI-generated change is deemed correct or not.

3.4 Accountability, Explainability, and Auditability

Many of the reviewed texts focused on mechanisms to make algorithmic and AI-based decisions accountable once they were put into practice. The research on explainable machine learning identified a taxonomy of approaches to machine learning evaluation: application-grounded, human-grounded, and functionally-grounded, and

used these to formulate machine learning control-plane explainability requirements (Doshi-Velez & Kim, 2017). A popular model-agnostic explanation technique showed that explanations of misclassifications, which could be local and interpretable, could improve user trust in classifiers and enable debugging, offering a practically achievable capability which was essentially a theoretical requirement (Ribeiro et al., 2016).

Accountability, at the organizational and administrative level, has been defined as involving contestability and the ability of those impacted and overseers to contest the result of automation, in addition to providing explanation (Busuioc, 2021). In that spirit, a set of requirements for record-keeping and process were defined for reviewable automated decisions, in order to make a decision "reviewable after the fact": Cobbe et al. (2021) proposed that input data, version of the model, and decision rationale should be recorded. Documentation practices, such as the "model cards" and "dataset cards" that document models and datasets respectively, have also been proposed as a practical accountability mechanism that connects technical and organizational stakeholders (Konigstorfer & Thalmann, 2022). In addition to these accountability measures, a review of algorithmic bias literature identified the sources of biases at different stages of data collection, algorithm development, and feedback in algorithm use, and also identified the different mitigation strategies at each of these stages, ranging from pre-processing to in-processing and post-processing (Kordzadeh & Ghasemaghaei, 2022). Lastly, a field scan of the algorithmic auditing ecosystem revealed significant variation in the range of auditing techniques, and continued lack of standardization, access to audited systems, and enforcement mechanisms for the auditing ecosystem itself, as compared to the systems it audits, suggesting that the auditing ecosystem needed governance systems akin to those the systems lack, as well (Costanza-Chock et al., 2022). The sub-literature suggests that in order to be accountable, an AI control plane must include capabilities beyond explanation, such as independent audit access, contestability provisions, and record-keeping.

3.5 AI Risk Management and Empirical Productivity Evidence

The most direct normative template for control-plane risk governance is the AI Risk

Management Framework, which outlines AI risk management around four categories (Govern, Map, Measure, Manage), and describes Properties of Trustworthy AI (validity and reliability, safe, secure and resilient, accountable and transparent, explainable and interpretable, privacy enhancement, and harmful bias managed) (Tabassi, 2023). The Govern function maps to organizational policy definition, Map maps to risk identification for a given use case for an AI-assisted modernization, Measure maps to quantitative risk-scoring mechanisms (which are discussed in Section 4), and Manage maps to gating and rollback mechanisms (which are discussed in Section 3.2).

The stakes of the control-plane decision-making are further strengthened by the empirical research on the capability and productivity benefits of AI-powered coding tools. Evaluation of a large language model trained on source code illustrated that the model is capable of non-trivial functional correctness on standardized programming benchmarks, with an accuracy of about twenty-nine percent on a test set of programming problems that has not been seen in the training set, and an increase of tenfold when multiple candidate solutions were sampled and the best solution was selected (Chen et al., 2021). This discovery suggests that individual AI-generated code suggestions have a significant chance of producing a working error, further emphasizing the importance of verification gates over blindly implementing AI-generated changes.

4. Synthesis: A Composite Governance-Risk Formulation

We reviewed literature from across these four categories of risk signals that are relevant to governance: Change-impact risk – from system entanglement and technical debt (Sculley et al., 2015); Functional-correctness risk – from imperfect reliability of the AI modifications (Chen et al., 2021); Accountability risk – from lack of explainability, documentation, and/or reviewability (Doshi-Velez & Kim, 2017; Cobbe et al., 2021; Konigstorfer & Thalmann, 2022); and Access-control risk – from scope and sensitivity of the resources the AI agent can modify (Rose et al., 2020; Chandramouli & Butcher, 2023). Synthesis is offered as an illustrative set of constructs that can be aggregated into a single governance-risk score (GRS) that can be passed into a control plane policy engine: The point of this is to illustrate how the constructs identified above might be aggregated, but

not to claim that it is an actual tool that has been validated.

The proposed AI-assisted change c is represented by a composite score that is a weighted sum of four normalized sub scores ranging from zero to one which is represented in equation 1.

$$GRS(c) = w_1 \cdot R_{change(c)} + w_2 \cdot R_{correctness(c)} + w_3 \cdot R_{account(c)} + w_4 \cdot R_{access(c)}, \sum w_i = 1$$

In Equation 1, R_{change} is a normalized change-impact risk term based on the technical-debt taxonomy of Sculley et al. (2015); $R_{correctness}$ is a functional-correctness risk term based on

confidence in the AI system output as reported by Chen et al. (2021) using a benchmark-based evaluation paradigm; $R_{account}$ is an accountability-readiness risk term based on the completeness of document, version retention, and reviewability artifacts available for the change; R_{access} denotes access-control risk term reflecting the sensitivity and blast radius of resources targeted, consistent with the zero trust least-privilege principles (Rose et al., 2020). The weights w_1 , w_2 , w_3 , and w_4 are organization-specific and would be determined by policy (as per the Govern function of the AI Risk Management Framework) and not empirically (Tabassi, 2023). The structure of this aggregation is depicted in Figure 4.

Figure 4

Structure of the Composite Governance-Risk Score

Risk Input	Weight	Aggregation	Output
R_{change} — change-impact risk from entanglement, unstable data dependencies, and configuration debt (Sculley et al., 2015)	$\times w_1$	$\rightarrow$	GRS(c)
$R_{correctness}$ — functional-correctness risk from benchmark-derived confidence in AI output (Chen et al., 2021)	$\times w_2$	$\rightarrow$	Governance-Risk Score
$R_{account}$ — accountability-readiness risk from documentation, version retention, and reviewability completeness (Cobbe et al., 2021)	$\times w_3$	$\rightarrow$	Passed to control-plane policy engine
R_{access} — access-control risk from sensitivity and blast radius of targeted resources (Rose et al., 2020)	$\times w_4$	$\rightarrow$	$\sum w_i = 1$; each $R \in [0, 1]$

Note. The four risk terms are normalized to the unit interval and combined as a weighted sum, as expressed in Equation 1. Weights are organization-specific and set by policy rather than derived empirically. The structure is illustrative and has not been empirically validated.

A deployment-gating threshold τ may then be defined such that a proposed change is admitted to automated deployment without additional human review when $GRS(c)$ is less than τ , and is instead

$$Decision(c) = \{Automateddeployment, if GRS(c) < \tau; Humanreview, if GRS(c) \geq \tau\}$$

routed to human-in-the-loop review when $GRS(c)$ is greater than or equal to τ , as expressed in Equation 2 and illustrated in Figure 5.

This formulation operationalizes, within a single decision rule, the convergent requirements identified across the four reviewed themes: infrastructure-derived access constraints, deployment-automation risk signals, accountability artifacts, and correctness confidence. It is presented

as a structural illustration of aggregation logic rather than as a validated risk model, and the specific functional form of each R term, along with appropriate weighting coefficients, would require empirical calibration within a given organizational context prior to production use.

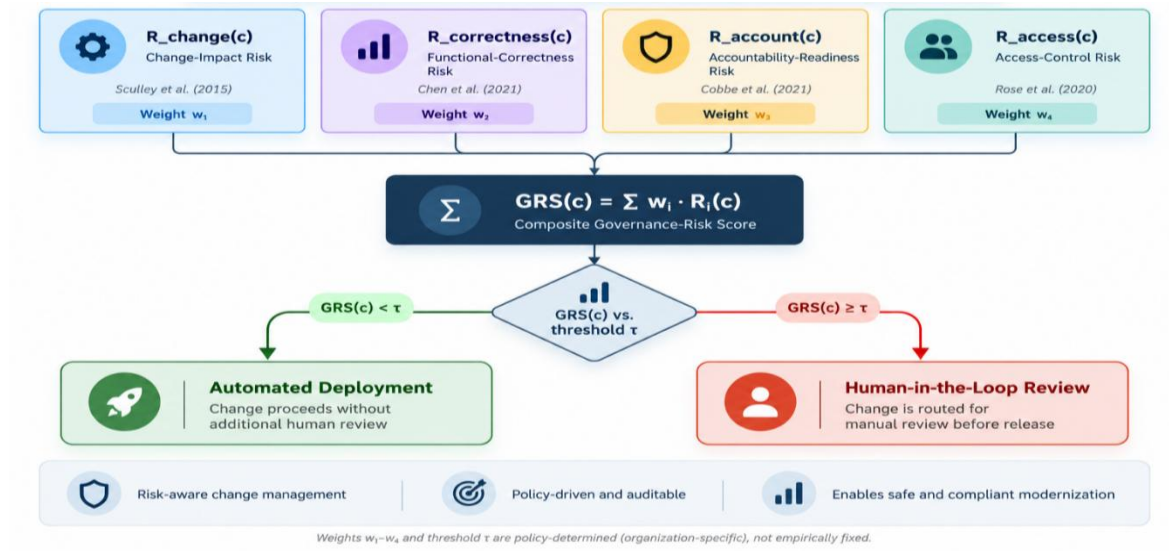


Figure 4. Composite Governance-Risk Score (GRS) formulation: four weighted risk sub-scores are aggregated and compared against a policy-defined threshold τ to determine whether a change is deployed automatically or routed to human review (Equations 1 and 2).

Table 3 Mapping of Composite Governance-Risk Score Components to Reviewed Constructs

GRS Component	Underlying Construct	Primary Source
R_change (change-impact risk)	Entanglement, CACE principle, configuration debt	Sculley et al. (2015)
R_correctness (functional-correctness risk)	Benchmark pass@k functional accuracy	Chen et al. (2021)
R_account (accountability-readiness risk)	Reviewability, documentation completeness	Cobbe et al. (2021); Konigstorfer & Thalmann (2022)
R_access (access-control risk)	Least privilege, continuous verification	Rose et al. (2020); Chandramouli & Butcher (2023)

5. Discussion

5.1 Convergent Findings

In all the four themes reviewed, convergence was identified that governance of AI-assisted systems must not be realized by a single static control, such as access credentials, model evaluation in one-off manner, or a single documentation artifact. Zero trust architecture is

built on continuous verification instead of trusting something in the first place (Rose et al., 2020); MLOps architecture is built on continuous monitoring of the model, rather than a single validation (Kreuzberger et al., 2023); and reviewable decision-making frameworks are built on persistent decision records that can be retrieved and reviewed at any time (Cobbe et al., 2021). This convergence provides a strong argument for

designing an enterprise AI control plane as a policy layer that is continuously evaluated, as opposed to a single pre-deployment point, which is reflected in only part of the current CI/CD-based MLOps pipelines (Garg et al., 2021).

5.2 Gaps and Tensions

There were three gaps identified. But the first, the literature on accountability and explainability, has been developed largely in the context of statistical and machine learning models predicting directly and does not fully account for the governance needs of code-creating AI systems, whose output is an executable artifact (Doshi-Velez & Kim, 2017; Ribeiro et al., 2016). Second, the literature on auditing ecosystems showed that the lack of standardized and independently verifiable audit mechanisms is a problem, even with a well-designed internal control plane, if not complemented by auxiliary third-party audit mechanisms (Costanza-Chock et al., 2022). Third, there is a trade-off between the verification overhead associated with strict governance and the significant productivity benefits derived from the use of AI-for-coding tools that have been shown in practice (Peng et al., 2023); a conservative governance control plane can jeopardize the productivity gains of AI-assisted modernization from the start.

5.3 Limitations

This review is restricted to the scope of the nineteen-source corpus, which covers both infrastructure and deployment, accountability and risk-management literatures, but is not an exhaustive catalogue of each domain per se. This composite governance-risk formulation proposed in section 4 is not empirically validated and the weights and definitions of components require to be calibrated and validated, depending on the specific organization, before being deployed in practice. Moreover, several of the sources reviewed prior to the widespread enterprise adoption of generative AI coding assistants are not specific to the generative AI context and so their applicability, although reasoned by analogy, has not been tested against deployment data specific to generative AI.

6. Conclusion

This systematic review brought together evidence from the cloud infrastructure, zero trust deployment automation, algorithmic accountability and AI risk management to describe the needs of an enterprise AI control plane to manage the

deployment of AI modernization services in a governed way. Results show that there is high agreement on the need for continuous verification, least privilege access, reproducible and monitored deployment pipelines, and documentation-based accountability, with empirical evidence that AI-assisted coding tools can be both useful for productivity and have a non-trivial functional-correctness risk. There is no single reviewed governance framework that treats Infrastructure, deployment automation and accountability requirements together and the composite governance-risk formulation proposed in this review is only one possible way to combine the two separate literatures into a single control-plane decision rule. Empirical studies to calibrate and validate such aggregation mechanisms with production deployment results are required in future.

References

- [1] Busuioc, M. (2021). Accountable artificial intelligence: Holding algorithms to account. *Public Administration Review*, 81(5), 825–836. <https://doi.org/10.1111/puar.13293>
- [2] Chandramouli, R., & Butcher, Z. (2023). A zero-trust architecture model for access control in cloud-native applications in multi-location environments (NIST Special Publication 800-207A). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-207A>
- [3] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., . . . Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv*. <https://doi.org/10.48550/arXiv.2107.03374>
- [4] Cobbe, J., Lee, M. S. A., & Singh, J. (2021). Reviewable automated decision-making: A framework for accountable algorithmic systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 598–609). Association for Computing Machinery. <https://doi.org/10.1145/3442188.3445921>
- [5] Costanza-Chock, S., Raji, I. D., & Buolamwini, J. (2022). Who audits the auditors? Recommendations from a field scan of the algorithmic auditing ecosystem. In *2022 ACM*

- Conference on Fairness, Accountability, and Transparency (pp. 1571–1583). Association for Computing Machinery.
<https://doi.org/10.1145/3531146.3533213>
- [6] Di Francesco, P., Lago, P., & Malavolta, I. (2018). Migrating towards microservice architectures: An industrial survey. In 2018 IEEE International Conference on Software Architecture (ICSA) (pp. 29–38). IEEE.
<https://doi.org/10.1109/ICSA.2018.00012>
- [7] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv.
<https://doi.org/10.48550/arXiv.1702.08608>
- [8] Garg, S., Pundir, P., Rathee, G., Gupta, P. K., Garg, S., & Ahlawat, S. (2021). On continuous integration/continuous delivery for automated deployment of machine learning models using MLOps. In 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 25–28). IEEE.
<https://doi.org/10.1109/AIKE52691.2021.00010>
- [9] Gholami, M. F., Daneshgar, F., Beydoun, G., & Rabhi, F. (2017). Challenges in migrating legacy software systems to the cloud—an empirical study. *Information Systems*, 67, 100–113. <https://doi.org/10.1016/j.is.2017.03.008>
- [10] Kordzadeh, N., & Ghasemaghaci, M. (2022). Algorithmic bias: Review, synthesis, and future research directions. *European Journal of Information Systems*, 31(3), 388–409. <https://doi.org/10.1080/0960085X.2021.1927212>
- [11] Königstorfer, F., & Thalmann, S. (2022). AI documentation: A path to accountability. *Journal of Responsible Technology*, 11, Article 100043.
<https://doi.org/10.1016/j.jrt.2022.100043>
- [12] Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11, 31866–31879.
<https://doi.org/10.1109/ACCESS.2023.3262138>
- [13] Leite, L., Rocha, C., Kon, F., Milojevic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys*, 52(6), Article 127.
<https://doi.org/10.1145/3359981>
- [14] Lwakatare, L. E., Raj, A., Crnkovic, I., Bosch, J., & Olsson, H. H. (2020). Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and Software Technology*, 127, Article 106368.
<https://doi.org/10.1016/j.infsof.2020.106368>
- [15] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (NIST Special Publication 800-145). National Institute of Standards and Technology.
<https://doi.org/10.6028/NIST.SP.800-145>
- [16] Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv.
<https://doi.org/10.48550/arXiv.2302.06590>
- [17] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). Association for Computing Machinery.
<https://doi.org/10.1145/2939672.2939778>
- [18] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology.
<https://doi.org/10.6028/NIST.SP.800-207>
- [19] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (pp. 2503–2511). Curran Associates.
<https://doi.org/10.5555/2969442.2969519>
- [20] Tabassi, E. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). National Institute of Standards and Technology.
<https://doi.org/10.6028/NIST.AI.100-1>