International Journal of Cyber Threat Intelligence and Secure Networking (IJCTISN)
eISSN: 3087-4289
Volume. 03, Issue. 04, pp. 01-09, April 2026"

Automation of Compliance Control Processes According to PCI DSS Standards in Hybrid Cloud Environments

Aghasi Gevorgyan
Head of Network Infrastructure, Armenian Card CJSC

Article Received: 15/01/2026, Article Accepted: 18/03/2026, Article Published: 02/04/2026
DOI: https://doi.org/10.55640/ijctisn-v03i04-01

ABSTRACT

The article addresses the problem of transforming compliance control with the PCI DSS 4.0.1 standard in hybrid cloud environments from an episodic audit practice into a continuous risk management function. It is shown that increasing infrastructure complexity, the deferred mandatory status of certain measures, and the accelerating pace of change render manual compliance operationally untenable. Meanwhile, the distribution of control points across hybrid/multi-cloud leads to the blurring of assessment scope, accountability boundaries, and control verifiability, which determines the high relevance of the study. The purpose of this work is to formalize a framework for automating PCI DSS compliance processes in a hybrid architecture, combining normative analysis with an engineering representation of controls. Scientific novelty consists in interpreting scoping and segmentation as a verifiable hypothesis. In projecting the principles of automated continuous compliance (policy-as-code, shifting left of checks, and formalization of the evidence base as a managed artifact) onto the specifics of PCI DSS, and in proposing a reference architecture and a phased automation roadmap that integrates management planes, telemetry, response processes, and an immutable evidence perimeter. The main conclusions indicate that PCI DSS compliance in a hybrid cloud can be maintained as a system property, dependent on continuous dependency inventory, a disciplined segmentation approach, standardized identity governance, a formalized shared-responsibility model, and machine-executable policies embedded into the change lifecycle. The article will be useful to hybrid infrastructure architects, information security specialists, payment service owners, and auditors involved in assessing and building PCI DSS-compatible solutions.

KEYWORDS

hybrid cloud environments, continuous compliance, compliance control automation, policy-as-code

Introduction

The PCI DSS standard remains one of the few industry regimes in which requirements for protecting payment data are formalized to such an extent that they become a mandatory framework for designing infrastructure, access processes, and monitoring. The current version of the requirements and testing procedures is 4.0.1 (June 2024). The standard specifies the period during which its provisions are applicable and the date on which any previous version ceases to be applicable. Because of its lifecycle, a one-time clean-up is always temporary (PCI, 2024). Additional pressure on operational resilience is also increasing: a number of measures previously marked as recommended have deferred mandatory status, and now they must be treated in assessments as full requirements. That is, the organization is compelled to design controls in advance so that they function continuously rather than only during report preparation. Under this logic, compliance turns from an audit event into an ongoing risk management function, where the quality of substantiation is determined by the reproducibility and completeness of evidence rather than by the volume of manual documentation.

Manual compliance has ceased to withstand the pace of change for two reasons. First, infrastructure and applications are updated substantially more frequently than internal regulations are revised. Therefore, divergence between the described and the actual state arises naturally and rapidly. Second, the effort required for manual evidence collection grows nonlinearly with the number of components and the frequency of changes. Contemporary research on continuous compliance control characterizes this shift as a transition from annual checks to continuous assurance, where standardization and automation are identified as critical factors, together with representing compliance artifacts and verification logic in a formalized, machine-processable form (Angermeir et al., 2024).

Hybrid clouds complicate compliance attainment not by adding another site, but by introducing a heterogeneous environment in which data and control points are distributed across different management planes, network domains, and access models. A survey of security in multi-cloud and hybrid architectures highlights that distributed storage, infrastructure heterogeneity, and inter-cloud communications pose significant challenges to confidentiality, access governance, secure interaction, and meeting regulatory requirements, and also necessitate scalable and adaptive compliance-oriented solutions (Ali et al., 2025). A practical implication for PCI DSS follows: even with formally identical requirements for protecting payment data, the verifiability of controls is blurred by accountability boundaries and differences in telemetry. Therefore, without centralized visibility, continuous auditing of changes, and regular policy updates, an organization loses the ability to reliably demonstrate compliance under dynamic conditions.

Materials and Methodology

The research materials were developed on a normative and scientific-practical basis, reflecting the evolution of PCI DSS requirements and the shift in compliance practices toward continuous control. The core of the corpus comprised: the current version of the PCI DSS 4.0.1 standard with a defined lifecycle of applicability and the deferred mandatory status of certain measures (PCI, 2024), as well as the PCI SSC glossary to operationalize the terms of scope, CDE, CHD, and SAD as the foundation for correct scoping and control provability (PCI, n.d.). To refine the hybrid-environment problem statement and mechanisms that erode verifiability, works were used on risks of distribution and heterogeneity in multi-/hybrid-cloud (Ali et al., 2025), on automating continuous compliance assurance as a transition from periodic checks to formalizable and executable rules (Angermeir et al., 2024), and on shared responsibility and its influence on control ownership and the evidence base in the cloud (Vanga, 2025). Additional sources were included that expose applied bottlenecks critical for PCI DSS in a hybrid architecture, specifically scoping and segmentation as the primary lever for reducing the hidden perimeter and inadvertent scope expansion (PCI Security Standards Council, 2016). Centralized observability and event-correlation architectures based on SIEM as a condition for reproducible monitoring (Tuyishime et al., 2023); identity management specifics in cross-cloud scenarios as a complexity factor for uniform least privilege (Haj et al., 2025); and approaches to automating compliance processes in cloud environments with an emphasis on machine verifiability and reducing manual effort (Wang & Yang, 2025).

The methodology is constructed as a linkage of normatively oriented analysis and engineering formalization of controls, where the goal is not to describe compliance, but to show how it can be continuously substantiated in a dynamic hybrid infrastructure. First, a conceptual decomposition of PCI DSS 4.0.1 requirements into verifiable entities was performed: assessment boundaries were defined through the CDE and related components, and CHD/SAD processing regimes and their implications for evidence automation were clarified (PCI, n.d.; PCI, 2024). Second, a comparative analysis of control-enforcement mechanisms in a heterogeneous environment was conducted: typical sources of configuration drift, telemetry gaps, and management-plane differences characteristic of hybrid/multi-cloud environments were compared, with an emphasis on how these properties undermine evidence reproducibility (Ali et al., 2025). Third, a content analysis of studies on automated continuous compliance was conducted to extract recurring principles (policy-as-code, shifting left of checks within the change lifecycle, and automated artifact collection) and their projection onto PCI DSS requirements (Angermeir et al., 2024; Wang & Yang, 2025). Finally, a practical automation framework was synthesized: scoping/segmentation as a verifiable hypothesis and an object of continuous validation (PCI Security Standards Council, 2016), centralized observability and event correlation as a prerequisite for monitorability and provability (Tuyishime et al., 2023), and formalization of shared responsibility as a necessary

precondition for fixing control ownership (Vanga, 2025) under increased IAM complexity in cross-cloud domains (Haj et al., 2025).

Results and Discussion

Within the logic of continuous compliance assurance outlined in the introduction, the first step is to fix the terms that define automation boundaries: the PCI DSS standard operates with the concept of assessment scope and the cardholder data environment (CDE), and the CDE includes not only technical components but also people and processes that store, process, or transmit cardholder data (CHD) and/or sensitive authentication data (SAD). A material detail is that the CDE also includes components that do not themselves process CHD/SAD, but have unrestricted connectivity to CDE components or can affect their security. This category typically creates the hidden perimeter of assessment. For practical formalization, it is important to preserve the distinction between CHD (e.g., primary account number and associated attributes) and SAD (e.g., validation values and magnetic-stripe/chip data), because their storage and permissible processing regimes differ fundamentally, and classification errors turn a technical defect into systemic non-compliance (PCI, n.d.).

The hybrid cloud environment in this work is understood as a composition of an organization's on-premises computing environment and one or more external cloud perimeters, supplemented by provider application services used as ready-made functions and platforms. Typical models differ less by the set of technologies than by the topology of data flows and connectivity channels: some CDE components may reside in the on-premises segment, while others are in a public cloud, and certain business functions are implemented through external application services, where access control and logging are defined differently. In practice, assessment boundaries begin to be determined not by deployment location but by how inter-segment connections are organized and what dependencies form between components. Therefore, connectivity and inter-perimeter integration become equivalent in importance to storage and processing. This view aligns with the characterization of contemporary multi-cloud and hybrid architectures as distributed and heterogeneous ecosystems, where heterogeneity and inter-cloud interactions elevate the complexity of ensuring security and meeting regulatory requirements (Ali et al., 2025).

The shared-responsibility model is critical for PCI DSS because technical safeguards are implemented at different layers of the stack, and the owners of these layers differ between cloud and on-premises environments. Therefore, the presence of a control and the ability to prove its operation cannot be automatically transferred from provider to customer or vice versa. At the normative level, this implies the need to decompose PCI DSS requirements across layers in advance and to assign responsibility for configuration, operation, and substantiation of each control. Otherwise, an assessment inevitably turns into a dispute about obligation boundaries. Moreover, access governance in a hybrid architecture is further complicated because different cloud service models govern different access objects and rely on different trust assumptions. Accordingly, control mechanisms also change, including cross-cloud operations and their auditing (Vanga, 2025).

From these frames, typical PCI DSS compliance problems in a hybrid cloud follow directly. Scope-definition errors and boundary creep occur when an organization fixes only the obvious payment-data processing nodes but underestimates the influence of connected components and integration channels; therefore, even formally correct segmentation may leave within the risk domain systems that do not touch the data but alter the conditions of its protection (PCI Security Standards Council, 2016). Configuration drift occurs when the actual state of environments diverges from the intended state, particularly under manual changes outside a controlled process. Fragmented logs and the difficulty of event correlation become a systemic barrier because observability is distributed across multiple management planes, while security events require normalization and consolidation for reproducible analysis, which is especially evident in works on cloud security event-management architectures (Tuyishime et al., 2023). Divergent identity and access management models hinder the uniform enforcement of least privilege, and empirical evidence shows that organizations in multi-cloud environments often maintain multiple identity providers simultaneously, thereby increasing the number of failure points and raising the cost of access rights errors (Haj et al., 2025). Finally, audit evidence collection becomes a distinct burden when checks are performed manually and asynchronously in response to operational changes, as evidence loses completeness and reproducibility. Research on automating compliance processes emphasizes that automated data collection and analysis reduce cycle time and manual effort, thereby making continuous assurance practically attainable (Wang & Yang, 2025). Table 1 illustrates Typical PCI

DSS Compliance Challenges and Mitigation Measures in Hybrid Cloud Environments.

Table 1. Typical PCI DSS Compliance Challenges and Mitigation Measures in Hybrid Cloud Environments (Ali et al., 2025; Haj et al., 2025; PCI, n.d.; PCI Security Standards Council, 2016; Tuyishime et al., 2023; Vanga, 2025; Wang & Yang, 2025)

Typical PCI DSS compliance issue	One-line essence	Risk/impact	What helps
Scope errors / hidden CDE perimeter	Overlooks connected components/channels with unrestricted connectivity to the CDE	Unexpected scope expansion and audit non-compliance	Connectivity & dependency inventory, network path control, continuous segmentation validation
CHD vs. SAD misclassification	Incorrect data classification leads to improper handling/storage	A technical defect becomes a systemic non-compliance	Data discovery & classification, masking/filtering, preventing sensitive data in logs/traces
Unclear shared-responsibility split	Control ownership (configure/operate/evidence) is not assigned across layers	Gaps in controls and disputable evidence	RACI mapping, control decomposition by layers (on-prem/IaaS/PaaS/SaaS), evidence playbooks
Configuration drift	Actual configurations diverge from documented/approved state due to manual changes	Loss of reproducibility; audit findings	IaC + policy-as-code, change control, automated drift detection/remediation
Fragmented logging / weak event correlation	Logs are dispersed across planes and hard to normalize/correlate	Weak monitoring/response and limited provability	Centralized SIEM, normalization, correlation rules, unified log retention/access policy
Heterogeneous IAM	Multiple IdPs/models make least privilege hard to enforce consistently	Excess privileges, access errors, higher audit burden	Federation/SSO, periodic access reviews, JIT access, automated policy checks
Manual evidence collection	Evidence is gathered out of sync with operational changes	Incomplete/non-reproducible audit trail	Automated evidence collection, integrity controls, continuous control reporting

The results of the study indicate that the primary determinants of PCI DSS compliance failure in hybrid cloud environments are boundary ambiguity (assessment scope and shared responsibility) and operational volatility (configuration drift and fragmented observability). These findings are derived from the systematic decomposition of PCI DSS requirements and the comparative analysis of hybrid control enforcement mechanisms.

The two key obstacles to PCI DSS compliance in hybrid clouds are boundary ambiguity (scope and shared responsibility) and operational volatility (configuration drift and fragmented observability): PCI DSS compliance evolves from a static attestable state to an observed property continuously maintained by systematic dependency mapping, standardized identity governance, and automated evidence generation. This framing supports the view that continuous compliance requires

integrating security controls with change management and telemetry pipelines rather than treating audits as periodic, manual checkpoints.

The transition from manual compliance assurance to automated control begins with abandoning descriptions that exist separately from the infrastructure and with translating requirements into formal rules executable by computing systems. Instead of checklists, a set of policies is formed that defines permissible states of resources and relationships between them, and that constrains hazardous changes before they are introduced. Policies also need to be authored for what is probably true: which configuration parameters are required, which events are to be logged, which component relationships are prohibited, and which divergences should be remediated. Versions of policies require consistency checking: in a hybrid environment, network constraints affect observability, which in turn affects investigability, and investigability impacts the admissibility of operational compromises.

Automation loses meaning if it merely detects deviations but does not create reproducible evidence. Therefore, the second principle involves formalizing the evidence base as a managed artifact. Evidence collection must be performed regularly or triggered by events, and its storage must prevent undetectable alteration or removal. The snapshot itself does not matter most, but someone should reconstruct the chain of causation: someone changed something, someone checked integrity, someone made a decision, and someone took action. The evidence that resulted would continuously form history, able to be verified and re-interpreted during the changes to the requirements or architecture over time.

A further requirement is measurability: a control must not merely be enabled, but must demonstrate effectiveness. Continuous observation of controls implies that, for each protection mechanism, indicators of operability and coverage are defined, along with thresholds beyond which the system deems the control degraded. This shifts compliance from binary logic to probabilistic logic: instead of answering 'compliant' or 'not', the answer becomes 'with what confidence it can be asserted that the control operates and covers the assessment scope'. In a hybrid environment, this is especially important because some signals reside on-premises, some externally, and some with application service providers; losing one observability source can silently render a control nominal. Therefore, measurement must account for telemetry completeness, event delivery latency, the proportion of untagged resources, and the frequency of changes that can invalidate the initial assumptions about scope boundaries.

Another principle is to shift control earlier to cheaper and reversible stages in the change lifecycle. Starting post-deployment checks means that every issue is a crisis: the deployment needs to be rolled back, a process needs to be reapproved, and an unplanned maintenance window and temporary exceptions need to be created. This exposes additional sources of risk. Hence, checks and policies must be embedded into change preparation, including infrastructure and configuration descriptions, and must function as a quality gate that blocks patently non-compliant solutions. As a result, speed and control cease to be rivals: speed is achieved through standardization and reuse of validated templates, while control is achieved by making any deviation explicit and requiring a deliberate decision that leaves a trace in the evidence base.

These principles naturally crystallize into a reference architecture in which different data and action types are linked into a single loop. At the foundation lie sources of truth about resources and their relationships: asset inventory, configuration data from registries, cloud management-plane data, network flows, change delivery data, and centralized security telemetry. Above them lies a verification loop that translates policies into concrete configuration checks, analyzes infrastructure descriptions, and verifies runtime environments, including container platforms. In parallel, a monitoring loop operates that aggregates logs and events, normalizes them, and binds them to resource models, so that deviation detection does not depend on manually crafted queries. When non-compliance is detected, a response loop is activated, where a deviation becomes managed work: an owner is assigned, a remediation scenario is selected, deadlines are recorded, and automated or semi-automated actions are executed. All of this is closed by an evidence loop in which state snapshots, check results, change chains, and exception decisions are preserved so that they can be re-verified and mapped to the assessment scope, including components connected to the cardholder data environment. At the requirements level, this enables automation of key measure groups: maintaining segmentation and network policies through verifiable connectivity rules; governing access through strict roles and mandatory additional factors for critical operations; ensuring encryption and key management through controlled settings and verifiable rotation cycles; sustaining vulnerability management through regular

checks and disciplined remediation with formal exceptions; maintaining logging and response through end-to-end event correlation; and change management through strict change-admission rules and mandatory decision capture, so that every system state is not only secure, but also provable. The transition to Automated Compliance Control is illustrated in Figure 1.

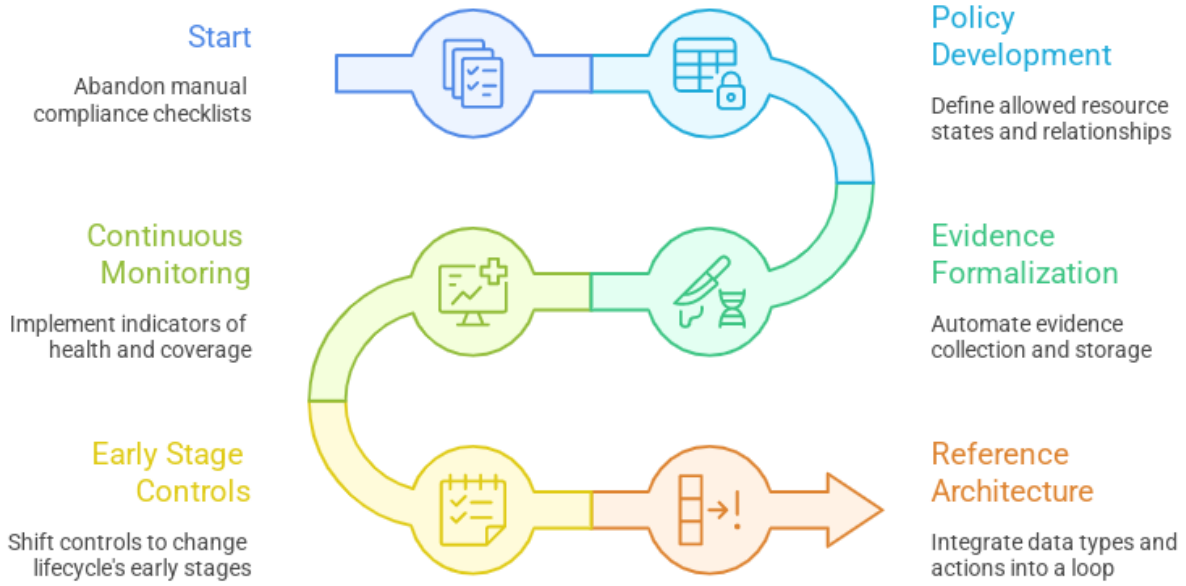

Fig. 1. Transition to Automated Compliance Control

Reducing the applicability radius of requirements begins with preventing payment data from circulating freely across shared infrastructure and anchoring it within an isolated perimeter where every ingress and egress point is known and controlled. Tokenization and cryptographic protection serve here not only as confidentiality mechanisms but also as instruments of scope management: if application systems receive a surrogate instead of a primary account number while the original value remains within a narrow perimeter, the number of components that can potentially affect payment data security is reduced. However, this effect is achieved only with strict isolation of the services performing the transformation, with a governed key lifecycle, and with controlled exchange channels with external parties. Particular attention is required for storage minimization, as any redundant copy of data in intermediate stores, reports, logged request parameters, or backups expands the assessment scope and significantly increases the cost of control automation, because evidence must be collected for a larger number of systems and scenarios.

Practical scope management in a hybrid environment ultimately reduces to a discipline of segmentation and the verifiability of the segmentation itself. Separation into the cardholder-data perimeter and other zones must be expressed in explicit connectivity rules rather than in architectural assumptions, because inter-segment dependencies arise not only at the network level but also through shared identity services, logging pipelines, orchestration platforms, and remote administration tools. Therefore, segmentation must be verified as a hypothesis rather than asserted as a declaration: automated control must detect unplanned routes, overly broad access rules, bidirectional bridges between zones, and components that are formally placed outside the payment perimeter but can modify their configuration or gain access to it. Importantly, segmentation verifiability relies on continuous resource inventory and on matching actual flows against the model of permissible interactions. Without this, even an initially correct scheme rapidly degrades due to configuration drift and localized manual changes.

The operating model links technical automation loops with accountable responsibility and predictable decision-making in non-standard situations. A responsibility matrix fixes which control elements are performed by the organization, which are provided by the cloud infrastructure provider, and which depend on third parties supplying services and components; without such fixation, it is impossible to sustainably collect evidence, because some artifacts will be out of reach and will require contractual mechanisms for retrieval. To sustain

continuous compliance, formal processes for handling exceptions and accepting residual risk are necessary. Otherwise, the system will either block business changes or create unmanaged bypasses. Each exception must have an expiration, a rationale, and observable conditions under which it remains acceptable. Change management serves as the binding mechanism between policies and reality, defining the rules under which changes are admitted, verified, and recorded in the evidence base. The quality of this model is measured by metrics of control coverage, time to remediate non-compliance, the share of drift relative to defined baselines, and the level of audit readiness at any given time. These metrics must be automatically computable and consistently interpretable for both on-premises and cloud perimeters. The Hybrid Compliance Automation Roadmap is shown in Figure 2.

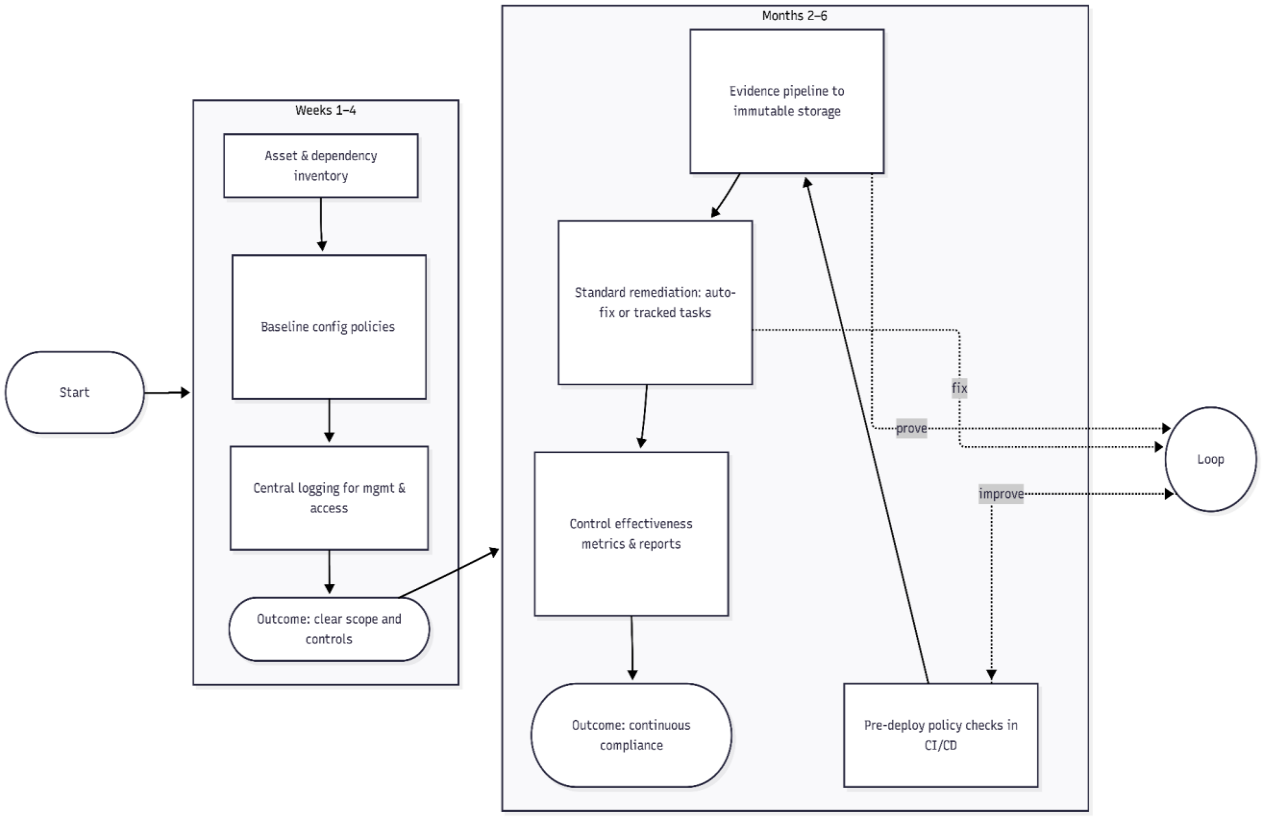

Fig. 2. Hybrid Compliance Automation Roadmap

The diagram formalizes a staged roadmap for compliance automation in a hybrid infrastructure, prioritizing early uncertainty reduction via asset/relationship inventory, baseline configuration policies, and centralized observability. It then operationalizes controls by integrating pre-deployment policy checks into the change lifecycle, establishing an immutable evidence pipeline, and standardizing remediation workflows. Finally, it advances toward continuous compliance by introducing quantitative control-effectiveness metrics and feedback loops that support longitudinal monitoring and improvement.

Conclusion

The work demonstrates that under PCI DSS 4.0.1 conditions, compliance ceases to be a one-time audit campaign and is compelled to transform into a continuous risk management function, because the standard defines a lifecycle of version applicability. Against this background, manual assurance practices tend to degrade predictably: the frequency of infrastructure and application changes exceeds the rate of procedural updates, while the effort required for evidence collection grows nonlinearly as the system becomes increasingly complex. A hybrid cloud amplifies the problem not by adding one more site, but by producing a heterogeneous environment with distributed control points, distinct management planes, and differing access models. As a result, control verifiability is blurred by responsibility boundaries and telemetry differences, and the ability to convincingly demonstrate compliance becomes derivative of centralized observability, continuous change auditing, and regular policy updates.

A key conclusion is that automation must rely on the rigorous formalization of the assessment scope and the PCI DSS conceptual framework, distinguishing between

CHD and SAD. Correctly understanding the CDE as a set comprising not only technical components but also people and processes, and accounting for the hidden perimeter of connected components with unrestricted connectivity or influence over CDE security. In a hybrid architecture, the shared-responsibility model becomes decisive: without decomposing requirements by layers and fixing control ownership (at the levels of configuration, operation, and evidence), assessment inevitably collapses into disputing obligation boundaries rather than verifying protection operability. Typical failures follow logically: scope errors, configuration drift, fragmented observability, and weak event correlation, as well as heterogeneous IAM, resulting in an incomplete and non-reproducible evidence base under manual artifact collection.

The proposed logic of transitioning to automated control reduces to the fact that requirements must exist not as descriptions adjacent to the infrastructure, but as versioned, machine-executable policies that define provably correct states of resources, connectivity, and logging, and shift checks to early stages of the change lifecycle. Automation retains meaning only when it generates a managed evidentiary history: immutable state snapshots, check results, change chains, exception decisions, and the causal linkage of who/what/when/why, enabling re-interpretation when requirements or architecture change. Finally, continuity requires measurability: a control must demonstrate effectiveness and coverage through computable indicators and degradation thresholds that account for telemetry completeness, event latency, and change frequency. Combined with segmentation discipline, verifiable segmentation itself, storage minimization, and exception governance, this model transforms compliance from a binary status into a maintainable system property that can be substantiated at any time, rather than only at the audit point.

Thus, the contribution of this study lies in demonstrating that PCI DSS compliance in hybrid cloud environments can be maintained as a continuous, verifiable system property rather than a periodic audit outcome. By formalizing scope, segmentation, shared responsibility, and evidence generation as machine-executable and continuously observable constructs, the proposed approach enables sustainable compliance under high change velocity without sacrificing auditability or operational resilience.

References

1. Ali, S., Talpur, D. B., Abro, A., Alshudukhi, K. S. S., Alwakid, G. N., Humayun, M., Bashir, F., Wadho, S. A., & Shah, A. (2025). Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions. *Computers & Security*, *157*, 104599. https://doi.org/10.1016/j.cose.2025.104599

2. Angermeir, F., Fischbach, J., Moyón, F., & Méndez, D. (2024). Towards Automated Continuous Security Compliance. *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 440–446. https://doi.org/10.1145/3674805.3690748

3. Haj, B., Laborde, R., Benzekri, A., Kandi, M. A., & Ferreira, A. (2025). Identity Management in Cross-Cloud Environments: Towards Self-Sovereign Identities Using Current Solutions. *Lecture Notes in Computer Science*, *15456*, 56–71. https://doi.org/10.1007/978-3-031-89350-6_4

4. PCI. (n.d.). *Glossary*. PCI Security Standards Council. Retrieved December 1, 2025, from https://www.pcisecuritystandards.org/glossary/

5. PCI. (2024). *Payment Card Industry Data Security Standard*. PCI. https://www.middlebury.edu/sites/default/files/2025-01/PCI-DSS-v4_0_1.pdf

6. PCI Security Standards Council. (2016). *Guidance for PCI DSS Scoping and Network Segmentation*. PCI Security Standards Council. https://listings.pcisecuritystandards.org/documents/Guidance-PCI-DSS-Scoping-and-Segmentation_v1.pdf

7. Tuyishime, E., Balan, T. C., Cotfas, P. A., Cotfas, D. T., & Rekeraho, A. (2023). Enhancing Cloud Security, Proactive Threat Monitoring and Detection Using a SIEM-Based Approach. *Applied Sciences*, *13*(22), 12359. https://doi.org/10.3390/app132212359

8. Vanga, P. R. (2025). Demystifying Cloud Security: Understanding Shared Responsibility Models. *International Journal of Information Technology and Management Information Systems*, *16*(1), 347–357. https://doi.org/10.34218/ijitmis_16_01_026

9. Wang, Y., & Yang, X. (2025). Machine Learning-Based Cloud Computing Compliance Process Automation. *Automation and Machine Learning*, *6*(1). https://doi.org/10.23977/autml.2025.060105