International Journal of Safety and Security Engineering
Vol. 16, No. 10, October, 2026, pp. 2217-2234
Journal homepage: http://iieta.org/journals/ijsse

Security-Aware Modeling of Data Integrity, Confidentiality, and Trust in Inter-Cloud Computing Systems

K. Vivekananda Reddy[1*], C. Shoba Bindu[2]

[1] Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur, Ananthapuramu 515002, India
[2] Department of Computer Science and Engineering, JNTUA College of Engineering Ananthapuramu, Ananthapuramu 515002, India

Corresponding Author Email: vivekkandula83@gmail.com

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.161003

Received: 10 August 2026
Revised: 17 September 2026
Accepted: 24 September 2026
Available online: 31 October 2026

Keywords:
inter-cloud computing, security requirement modeling, data integrity, confidentiality, trust management, policy consistency, auditability, federated cloud security

ABSTRACT

As data-intensive apps transition to federated cloud computing systems, multiple autonomous trust domains exist among inter-cloud systems, thus increasing scalability and fault tolerance but complicating security due to split trust and policy enforcement. Current inter-cloud systems use security service composition for each cloud system individually, with a weak security foundation for federations. The paper proposes a security-aware and requirement-driven modeling approach for inter-cloud systems. The security invariants of confidentiality, integrity, availability, auditability, and trust are first-class security concepts at the federation level for inter-cloud systems. The proposed modeling approach formally represents security requirements for any cloud provider and their association with the inter-cloud system and security services. Scenario-based simulation evaluation under heterogeneous policy, trust, enforcement, and threat conditions provides preliminary empirical evidence that the proposed modeling approach can reduce security contradictions, improve policy consistency and audit completeness, and support trust continuity within the evaluated inter-cloud configurations. This modeling approach is particularly relevant for safety- and assurance-critical inter-cloud applications, where verifiable security preservation, auditability, and trust continuity are mandatory.

1. INTRODUCTION

The proliferation of various data-based services, including federated analytics, healthcare information systems, e-governance solutions, and digital forensics, has led to an increasing need for inter-cloud computing—an arrangement where independently administered cloud infrastructure collaborates on data processing and service delivery. The InterCloud framework has been one of the first formulations of utility federation between diverse clouds [1]. Later, complementary frameworks for inter-cloud architecture and security have been developed by describing the principal actors of cloud infrastructure and interactions between them [2]; providing cloud information security controls for customers and providers [3]; studying security issues of multi-access edge computing (MEC) platforms, zero-trust architecture, and federations [4]. While these efforts have greatly facilitated the inter-cloud design, deployment, and operation, using the recommendations does not provide any guarantees that a security requirement will be continuously preserved during cross-domain workflow operations.

Previous research has addressed the topics of cloud vulnerabilities [5], risks associated with different cloud service-delivery models [6], cloud security threats [7]; architecture- and stakeholder-specific cloud security problems [8]; identity and authorization issues [9]; and relations between privacy, security, and trust [10, 11]. These publications show how hard it is to understand a policy across multiple independent clouds—the issues arise because of the inter-cloud differences in policy semantics, identity federations, assurance level, evidence gathering, and contractual or jurisdictional obligations. Local compliance at each of the participating providers does not guarantee that a workflow-level security requirement is preserved.

The body of previous work has identified the security-related attack surfaces in cloud and provided the corresponding protective mechanisms—proposals include privacy-aware cloud storage and processing [12]; trusted cloud execution environment [13]; analysis of confidential data exposure [14]; protections against information leakage via cloud indexing [15]. The mechanisms mentioned above cover important attack surfaces but, in case of inter-cloud execution, individual guarantees may be insufficient because of the possible combination of correct controls at each cloud with inconsistent policies at others—for instance, encryption will protect a data set transferred to another provider while inconsistent authorization policies can lead to unwanted data disclosure; the same is true for audit logs—it may be

impossible to build a sequence of events using the logs collected independently by separate providers. Thus, the issue is not about verifying the functioning of individual controls but about the ability to preserve the requirement initially defined for the workflow.

There exist other security tools that can complement the proposed work: logic for access control analysis [16]; formal protocol analysis [17]; confidential computing techniques [18]; and dynamic zero-trust access control [19]. Recent multi-cloud literature investigates secure and redundant storage [20]; security assessment based on process mining [21]; cross-cloud data governance [22]; and trustworthiness measurement using compliance feedback and assessment [23]. However, although all these advances give useful building blocks, the verification of one mechanism, service, or provider in isolation does not, without any additional compositionality check, guarantee the preservation of confidentiality, integrity, availability, auditability and trust requirements in the face of cross-domain policy changes. Figure 1 demonstrates the difference – inter-cloud federation consists of independently administered providers with their respective access policies, identity management services, enforcement techniques, and monitoring systems. The federation services coordinate cross-domain interactions, but security decisions and related evidence can be distributed. A requirement verified in one domain may become weaker after the data migration, service chaining, delegation, or handover to the next cloud provider. Therefore, a proper security assurance statement in the inter-cloud federation should describe not only the security controls used in each of the clouds but also conditions under which these controls together ensure the satisfaction of the requirement throughout the workflow.

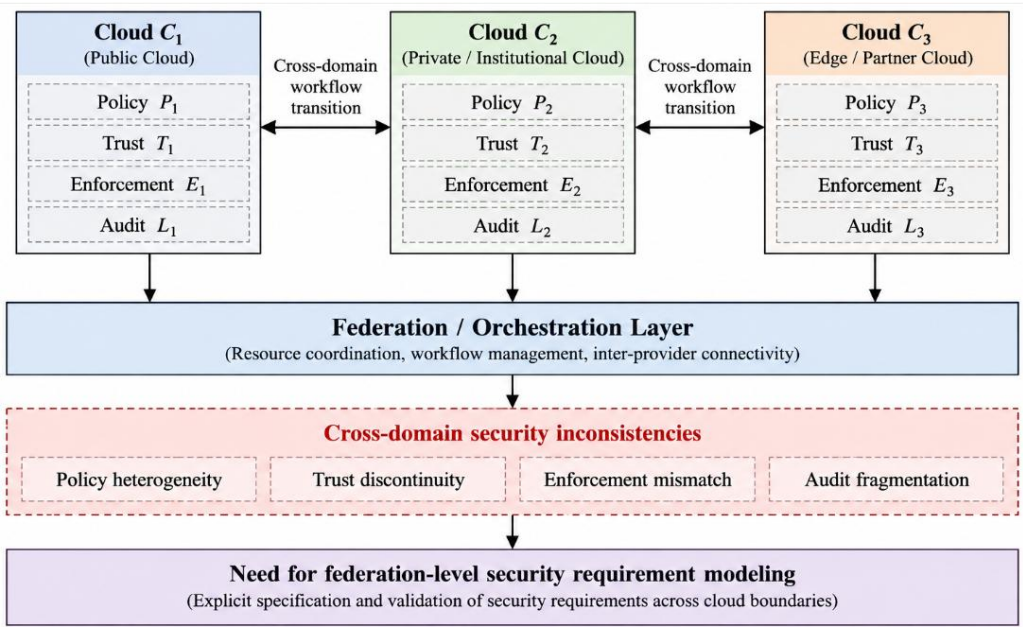


Figure 1. Conventional inter-cloud architecture showing autonomous provider-local security controls and cross-domain inconsistencies in policy, trust, enforcement, and audit semantics

A separate aspect of the security challenge is introduced by the notion of cloud trust—historically, cloud trust models took into account the provider reputation and service trustworthiness [24], extending the existing reputation-based approaches to online services [25]. The trust management research for the Internet of Things (IoT) expanded this question to devices and highly distributed interactions [26]. There are studies addressing the security threats in mobile-edge and fog computing [27]; developing IoT-specific security risk-assessment techniques [28]; studying fog gateways as intermediaries between devices and cloud services [29]; and investigating quality-of-experience and energy-efficiency trade-offs in cooperating fog nodes [30]. All these streams of research demonstrate the variety of constraints that cloud, edge, and device domains introduce while being part of one workflow; and stress the point that workflow optimization and provider selection alone are not enough to ensure the security of a workflow.

In particular, this becomes especially important for workflows that fall under regulation or are safety-critical—the healthcare analytics pipeline may have required access controls at each of the processing sites but fail to preserve the consistency of authorization scope after the data sharing; the e-governance services may log the events locally without having a comprehensive federation-wide audit trail; the digital forensic processing may become non-accountable after the evidence migration across organizational boundaries. The importance of these weak points is emphasized by the cybersecurity research literature [31].

These motivations give rise to the need to develop a security model where the preservation of the security requirement across domains becomes an explicit property rather than a property implicitly guaranteed by the mere existence of the provider-local mechanisms. To fill the gap, the present study suggests a security-aware, requirement-driven inter-cloud framework where confidentiality, integrity, availability, auditability, and trust are treated as federation-level invariants.

The main contributions of this study are as follows:

•Proposes an inter-cloud security model as a set of federation-level invariants rather than a set of provider-local security controls.

•Formulates the local-to-global security contradiction and describes the conditions in which the independently satisfied local requirements do not compose across the workflow.

•Develops a requirement-driven Ω modeling framework allowing relating security requirements, threat mapping and

enforcement projection across the heterogeneous domains.

•Proposes the security evaluation measures—Global Security Satisfaction (SS), Weighted Security Satisfaction Index (SSI), Policy Consistency Resolution Rate (PCRR), Threat Coverage Ratio (TCR), and Residual Contradiction Index (RCI)—allowing assessment of the preservation of the federation-level security invariants.

•Evaluates the requirement preservation and computation overhead using scenario-based simulations.

2. LITERATURE REVIEW

The security of cloud and inter-cloud computing has been investigated from several complementary perspectives, including distributed-cloud architecture, security mechanisms, privacy protection, trust management, policy enforcement, and security analysis for fog and edge environments. Although these research directions have established important foundations for securing individual cloud services and distributed resources, their assumptions and objectives differ substantially from the federation-level security problem considered in this work. Therefore, the related literature is synthesized below according to the principal security functions addressed and their limitations in heterogeneous inter-cloud environments.

2.1 Inter-cloud architectures and security foundations

Early research on inter-cloud computing primarily focused on enabling interoperability, resource sharing, service negotiation, and scalable execution across independently managed cloud infrastructures. The InterCloud architecture [1], for example, established a federation-oriented model for dynamically coordinating heterogeneous cloud resources. Standardized cloud and distributed-computing frameworks subsequently provided architectural roles, service models, operational interfaces, and security guidance for cloud and edge environments [2-4]. These contributions are important because they provide the architectural foundation required for cross-provider execution.

However, architectural interoperability does not itself guarantee the preservation of security properties across participating administrative domains. Individual providers may employ different identity infrastructures, encryption policies, authorization semantics, logging mechanisms, and trust assumptions. Consequently, security guarantees established within one provider may not remain semantically equivalent when a workflow migrates to or interacts with another provider. Existing architectural and standardization frameworks therefore provide essential infrastructure-level guidance but do not, by themselves, offer a unified formal abstraction for determining whether confidentiality, integrity, availability, auditability, and trust remain satisfied throughout an inter-cloud workflow.

More recent multi-cloud research has further emphasized that heterogeneous identity and access-management mechanisms, inconsistent security configurations, differences in governance policies, and limited interoperability among independently administered providers can introduce cross-cloud security gaps [21, 22]. These studies demonstrate the need for coordinated security governance, continuous monitoring, and harmonization of controls across provider boundaries. Nevertheless, their primary focus remains on interoperability, security monitoring, and governance rather than on formally determining whether a common set of security requirements is continuously preserved throughout the complete inter-cloud workflow.

2.2 Cloud security threats and mechanism-centric protection

A second major stream of research has examined security threats associated with cloud computing. Prior studies have identified vulnerabilities arising from virtualization, data outsourcing, isolation failures, identity management, service-model differences, and distributed control [5-9]. These studies establish that transferring computation and data from users to cloud infrastructures introduces new dependencies on provider-controlled security mechanisms and governance policies.

Correspondingly, several protection mechanisms have been proposed for individual security objectives. Privacy-aware cloud storage and processing [12], trusted execution and remote attestation [13], protection against information leakage [15], and formal access-control reasoning [16] provide mechanisms for strengthening confidentiality, integrity, access control, and execution assurance. Such approaches are effective within the scope for which the underlying enforcement assumptions remain valid.

The limitation arises when these mechanisms are composed across heterogeneous cloud providers. A mechanism that satisfies a requirement within one administrative domain does not necessarily establish equivalent protection after a cross-cloud transition. Variations in policy semantics, trusted components, identity representations, cryptographic configurations, and enforcement behavior can introduce gaps between locally satisfied controls and globally preserved security requirements. Thus, the principal unresolved issue is not the absence of individual security mechanisms but the absence of a federation-level abstraction capable of determining whether their composition preserves a common security requirement throughout the inter-cloud execution.

2.3 Privacy, trust, and cross-domain governance

Privacy and trust constitute another major body of cloud-security research. Previous work has highlighted loss of control, reduced operational transparency, confidentiality risks, and trust concerns arising when data and computing resources are delegated to external cloud infrastructures [10-12, 14]. These findings establish that security cannot be evaluated solely through technical protection mechanisms because provider behavior, governance conditions, transparency, and accountability also influence the security state of outsourced applications.

Trust-management approaches have consequently employed reputation, behavioral evidence, compliance information, and service-quality indicators for evaluating cloud or distributed-system entities [24-26]. Such models are particularly useful for provider evaluation and trust estimation. Nevertheless, provider-selection trust and federation-execution trust represent different problems. A provider that is individually considered trustworthy does not automatically guarantee that trust relationships remain valid when workflows span multiple administrative domains.

Recent research has extended cloud trust assessment beyond conventional reputation-based models by

incorporating security-control compliance, operational evidence, user feedback, and audit-related information into the evaluation of cloud-provider trustworthiness [23]. Such approaches provide a stronger evidence-based basis for provider assessment and risk-aware cloud-service selection. However, evaluating the trustworthiness of individual providers remains different from establishing trust continuity during inter-cloud workflow execution, because the latter additionally depends on interactions among independently governed domains.

In an inter-cloud environment, trust continuity depends additionally on compatibility among security policies, consistency of enforcement decisions, transfer of identity and authorization context, and availability of auditable evidence across domain boundaries. Existing trust approaches therefore provide important local or entity-oriented trust information but provide limited support for representing trust as a security property whose continuity must be explicitly validated throughout a multi-domain workflow.

2.4 Security challenges in federated, fog, and edge

The transition toward fog, edge, IoT-cloud, and decentralized computing further demonstrates the limitations of purely provider-centric security assumptions. Studies of fog and edge security have identified wider attack surfaces, decentralized control, mobility, dynamic interactions, and heterogeneous administrative boundaries as important security challenges [27, 28]. Research on fog-cloud interoperability has similarly demonstrated that distributed applications increasingly depend on cooperation among resources operating under different infrastructural and governance conditions [29, 30].

These environments reinforce a broader observation relevant to inter-cloud computing: as workflows become increasingly distributed, the security boundary no longer coincides with a single infrastructure provider. Security violations can emerge not only inside individual components but also at the interfaces and transitions among otherwise secure components. Consequently, evaluating only the local security state of each participating environment is insufficient for establishing the security of the complete federated workflow.

More recent research has addressed these distributed-security challenges through multi-cloud monitoring, zero-trust access control, confidential computing, and cross-cloud governance mechanisms [18-22]. Zero-trust approaches reduce reliance on implicit trust by continuously verifying identities, access requests, and contextual security conditions, while confidential-computing techniques strengthen protection of data during processing through trusted execution and attestation mechanisms [18, 19]. Multi-cloud monitoring and governance approaches further improve the visibility of security events and coordination of controls across heterogeneous providers [21, 22]. These developments substantially strengthen individual security functions; however, they predominantly address specific concerns such as authentication, authorization, secure execution, monitoring, or governance rather than jointly validating the preservation of multiple security requirements throughout inter-cloud workflow transitions.

Formal security analysis provides a useful foundation for addressing this remaining problem because system behavior and security conditions can be represented explicitly rather than inferred solely from deployed mechanisms. Formal verification techniques have previously demonstrated the value of explicit security modeling in protocol analysis [17], while formal reasoning has also been applied to access-control enforcement [16]. However, the current literature considered in this study does not provide a unified requirement-driven formulation that treats confidentiality, integrity, availability, auditability, and trust collectively as federation-level invariants whose preservation can be evaluated across provider boundaries.

2.5 Synthesis of existing approaches and research gap

The reviewed approaches can therefore be distinguished according to the level at which security is represented. Architectural and standardization approaches [1-4] primarily establish interoperability and cloud-security foundations. Threat-oriented studies [5-9] characterize important vulnerabilities and attack surfaces. Mechanism-centric approaches [12, 13, 15, 16] strengthen specific security controls, whereas privacy- and trust-oriented approaches [10-12, 14, 24-26] address confidentiality, provider trust, transparency, and governance. Fog- and edge-oriented studies [27-30] further demonstrate the security implications of decentralized and heterogeneous execution.

Despite these contributions, a common limitation remains: security is predominantly represented either through provider-local mechanisms, individual security objectives, or independently evaluated trust and policy properties. Such representations do not explicitly determine whether security requirements remain preserved when a workflow crosses heterogeneous administrative, policy, trust, enforcement, and audit boundaries.

This limitation creates three interrelated research gaps. First, there is no unified representation through which confidentiality, integrity, availability, auditability, and trust can be treated collectively as federation-level security requirements. Second, existing approaches provide limited explicit coupling between cross-cloud threats and the security requirements that those threats may violate. Third, the composition of independently secured cloud environments does not provide a formal basis for determining whether locally satisfied requirements remain satisfied during cross-domain transitions.

Accordingly, the present work addresses the inter-cloud security problem from a requirement-driven rather than solely mechanism-driven perspective. The proposed framework models confidentiality, integrity, availability, auditability, and trust as federation-level invariants; associates these requirements with inter-cloud threat conditions; maps them to provider-independent enforcement capabilities; and quantitatively evaluates their satisfaction throughout federated workflow execution. This positioning distinguishes the proposed approach from conventional compositions of local security mechanisms and provides the methodological basis for the proposed formal model.

3. PROPOSED METHODOLOGY

This study adopts a requirement-driven security-modeling approach in which confidentiality, integrity, availability, auditability, and trust are treated as federation-level properties rather than provider-specific mechanisms. The methodology

builds on inter-cloud architectures [1], cloud-security and orchestration models [2-4], formal access-control reasoning [16], and established cloud, fog, and decentralized-system threat models [5, 6, 9, 27].

As illustrated in Figure 2, the framework integrates requirement abstraction, inter-cloud threat mapping, enforcement projection, and quantitative security assessment within a unified modeling process.

The proposed framework has four tightly coupled layers. They are as follows:

(1) Security Requirement Layer: the preservation of confidentiality, integrity, availability, auditability, and trust is specified as federation-level invariants.

(2) Inter-Cloud Threat Mapping Layer: each security requirement is systematically correlated against realistic inter-cloud threats, which are the result of virtualization, data outsourcing, and decentralized orchestration [5, 6, 14, 27].

(3) Enforcement Binding Layer: abstract security requirements are mapped onto deployable, provider-agnostic enforcement capabilities, which are the result of standard cloud security services and trusted computing foundations [2-4, 13].

(4) Security Satisfaction Layer: the degree to which the inter-cloud execution preserves the modelled security invariants is quantitatively measured using the principles of formal validation [16].

This layered organization of the security engineering process captures the essence of inter-cloud security as not just an amalgamation of discrete services, but as an inherent property of the system, which preserves the requirements. This also captures the essence of the fragmentation of trust, the fragmentation of enforcement semantics, and the audit discontinuity, as identified in the prior cloud and inter-cloud security research [5, 9-11, 24].

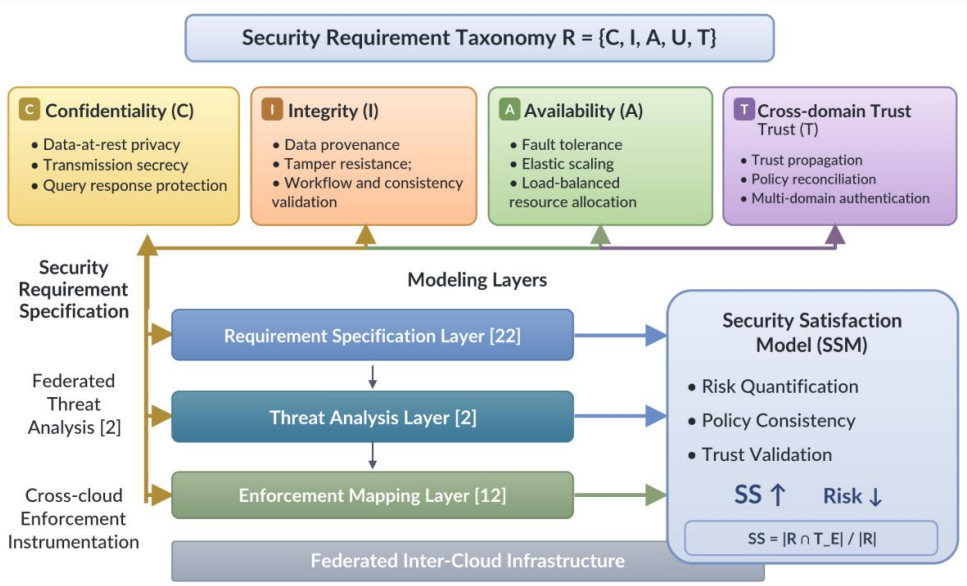


Figure 2. Proposed framework for modeling the requirement for inter-cloud security

3.1 Formal problem definition and security requirement specification

An inter-cloud computing environment is defined as an environment consisting of a federation of autonomous cloud providers who are working collaboratively in executing data-intensive applications at scale [1, 2, 4]. Let the inter-cloud federation be denoted as:

$$F = \{C_1, C_2, \dots, C_n\}$$

Each cloud, C_i, runs with its own set of security policies, P_i, and trust models, T_i, and enforcement semantics, E_i, respectively [1, 10, 11]. A data-intensive application running on F does so as a workflow.

$$w = \{w_1, w_2, \dots, w_m\}$$

where, components may migrate, replicate, and interact dynamically across heterogeneous cloud domains.

The global inter-cloud security requirement set is defined as

$$R = \{C, I, A, U, T\}$$

where, C, I, A, U, and T represent confidentiality, integrity, availability, auditability, and cross-domain trust, respectively. These properties are repeatedly identified as critical security requirements in cloud, fog, and federated computing environments [5, 6, 8, 9, 27, 28].

For a requirement $r \in R$ the local security satisfaction within cloud C_i during workflow execution is defined as follows.

Security satisfaction state in contemporary intercloud infrastructures is mostly provider-local, resulting in the following:

$$S_i(r, W) = f(P_i, T_i, E_i, W)$$

with

$$S_i(r, W) \in [0, 1]$$

where, $S_i(r, W)$ denotes the degree to which requirement r is satisfied within cloud C_i. A value close to 1 indicates strong local satisfaction, whereas a value close to 0 indicates violation or inadequate satisfaction of the requirement.

Local satisfaction, however, does not necessarily imply that the same security requirement remains preserved when a workflow crosses from one administrative cloud domain to

another. Previous studies have shown that heterogeneous policy semantics, decentralized control, disjoint trust anchors, and incompatible audit infrastructures can weaken security guarantees during cross-domain execution [10, 15, 22, 24, 25, 27]. Therefore, an explicit cross-domain transition satisfaction function is introduced.

For every workflow transition from cloud C_i to cloud C_j, let

$$G_{i\,j}(r, W) \in [0, 1]$$

denote the satisfaction of requirement r during the inter-cloud transition. It is defined as

$$G_{i\,j}(r, W) = g\left(P_i, P_j, T_i, T_{j,} E_i, E_{j,} L_i, L_j, W\right)$$

where, $G_{i\,j}(r, W)$ captures the compatibility of security policies, consistency of enforcement semantics, continuity of trust relationships, and preservation of audit information between the two participating cloud domains.

Let $D(W)$ denote the set of cloud domains participating in workflow W, and let $T(W)$ denote the set of inter-cloud transitions traversed by that workflow. Federation-level satisfaction of a security requirement is then defined as

$$S_F(r, W) = min\left[\min_{i \epsilon D(W)} S_i(r, W), \min_{(i,j) \epsilon T(W)} G_{ij}(r, W)\right]$$

This formulation treats federation-level security satisfaction as a scalar quantity rather than a set intersection. It also ensures that the global security state depends not only on the local security satisfaction of the participating providers but also on the preservation of the requirement during every cross-domain transition. Thus, the weakest local domain or cross-domain transition determines whether the requirement remains satisfied at the federation level.

An inter-cloud security contradiction occurs when a security requirement is satisfied locally within all participating cloud domains but fails to remain satisfied at one or more cross-domain transitions. Let $T \in [0,1]$ denote the minimum acceptable security-satisfaction threshold. The contradiction condition is therefore defined as

$$C(r, W) = 1 \iff [I \in D(W) \min Si(r, W) \geq T] \wedge [SF(r, W) < T]$$

Equivalently, the contradiction may be interpreted as a condition in which

$$\min_{i \epsilon D(W)} S_i(r, W) \geq T$$

while

$$\min_{(i,j) \epsilon T(W)} G_{ij}(r, W) < T$$

This means that all participating cloud providers may individually satisfy a requirement while the requirement is violated during a cross-domain transition. Such contradictions may arise, for example, from confidentiality degradation during data migration between providers [14], integrity degradation during distributed processing and indexing [15], trust degradation between heterogeneous providers [24, 26], or audit discontinuity across incompatible logging infrastructures [21, 22].

Based on this formulation, a security-aware inter-cloud modeling framework must satisfy the following fundamental constraints:

•Semantic consistency of policies: security-policy interpretations and enforcement semantics should remain compatible across different administrative domains [2, 3, 10, 11].

•Trust continuity: trust relationships should remain valid throughout cross-domain workflow execution and should not collapse because of inconsistent provider behavior, trust interpretation, or domain composition [24-26].

•Audit completeness: security-relevant events generated across participating cloud domains should be traceable and correlatable to construct an integrated and verifiable audit trail [21, 22].

•Requirement preservation: confidentiality, integrity, availability, auditability, and trust should remain satisfied throughout both local execution and cross-domain workflow transitions [5, 6, 8,9].

The inter-cloud security problem can therefore be formulated as the construction of a security-aware modeling function

$$\Omega : (F, W, P, T, E, L, R) \rightarrow S_F$$

where, P, T, E, and L collectively denote the policy, trust, enforcement, and audit characteristics of the participating cloud domains. The objective of Ω is to maximize federation-level security satisfaction while minimizing unresolved cross-domain contradictions, such that the satisfaction of every requirement $r \in R$ can be formally represented, evaluated, and quantitatively verified throughout the complete inter-cloud workflow.

This formulation casts inter-cloud security as a problem of requirement preservation across both local execution and cross-domain transitions, rather than as the simple composition of provider-local security mechanisms. It provides the formal basis for the security requirement taxonomy, threat–requirement coupling, enforcement mapping, and quantitative security satisfaction model developed in the following subsections.

The methodology begins with the creation of a Security Requirement Taxonomy specific to data-intensive inter-cloud systems. Based on the classical cloud security attributes [5-7, 9], as well as the research work done in the fields of privacy and trust [10, 11, 24], and the inter-cloud architectural basis [1, 4], the global requirement set is given as:

$$R = \{C, I, A, U, T\}$$

The Confidentiality (C) principle defines data rest, transport, and process security [12, 14, 15]. The Integrity (I) principle addresses resistance to tampering and the integrity of cloud computations through trusted execution and attestation mechanisms [13]. The Availability (A) principle defines survival with inter-cloud failures and policy restrictions [4, 8]. The Auditability (U) principle defines trace reconstruction from diverse logging systems [21, 22]. The Trust (T) principle defines reliability, conformance, and inter-cloud invariant provider trustworthiness [24-26].

Each requirement $r_i \in R$ is formalized in the Requirement Specification Layer, wherein provider-specific policies are abstracted in terms of provider-agnostic descriptors, in line with the principles of logical security modeling and reasoning

about access control [16]. In this level, the security requirements are specified in terms of inter-cloud invariants, rather than being provider-local.

3.2 Inter-cloud threat–requirement coupling and enforcement mapping

The second methodological step performs Inter-Cloud Threat Surface Analysis. Based on cloud and fog threat taxonomies [5-7, 9, 27], for every requirement r_i, a set of inter-cloud threats Thi can be identified as:

$$M: r_i \rightarrow Th_i$$

$$\text{with } Th_i = \{th_1, th_2, \ldots\ldots, th_m\}$$

This mapping shows how the assumptions about confidentiality, integrity, and trust are affected by virtualization leakage [6], risks associated with data outsourcing [14], decentralized cloud orchestration [27], and heterogeneous governance [10, 11]. Threat analysis included in requirement modeling helps to capture cloud risk propagation across inter-cloud abstractions.

The third stage introduces the Enforcement Mapping Layer, which connects abstract requirements with deployable inter-cloud enforcement functionality. Based on cloud security services [2-4] and Trusted Platform Models [13], the enforcement primitives of identity federation, cryptographic protection, policy translation, and audit aggregation are mapped to each r_i.

This layer prefers capability-enforcement primitives over vendor-centric ones and helps the heterogeneous cloud meet the requirements by providing controls with similar semantics.

3.3 Quantitative security satisfaction modeling

In order to measure requirement preservation at the federation level, a SS is proposed in this method. Let R denote the set of federation-level security requirements, and let T_E denote the subset of security requirements verified as satisfied through federation-level enforcement. The SS score is defined as follows:

$$SS(R, T_E) = \frac{|R \cap T_E|}{|R|}, SS \in [0,1]$$

For data-intensive tasks, the concept of the SSI has been proposed as follows:

$$SSI = \sum_{i=1}^{|R|} \alpha_i . S(r_i)$$

where, $S(r_i)$ denotes the satisfaction level of security requirement r_i, and α_i represents its corresponding criticality weight. The weights are non-negative and normalized such that $\sum_{i=1}^{|R|} \alpha_i = 1$. Accordingly, $SSI \in [0,1]$, provided that $S(r_i) \in [0,1]$ for all requirements. The proposed approach redefines inter-cloud security from being simply compliant or not compliant to being a system-level attribute that could be continuously validated throughout the entire inter-cloud workflow execution process.

4. RESULTS AND DISCUSSION

This section presents a scenario-based simulation evaluation of the proposed security-aware inter-cloud modeling framework. The objective is to examine whether the framework can preserve confidentiality, integrity, availability, auditability, and trust, represented by $R = \{C, I, A, U, T\}$, under heterogeneous policy semantics, trust assumptions, enforcement conditions, audit infrastructures, and controlled cross-domain threat scenarios. The evaluation is not intended to constitute a formal proof of effectiveness or a production-scale benchmark. Instead, it provides a controlled empirical evaluation of the proposed modeling approach under the explicitly defined simulation conditions.

The evaluation considers distributed inter-cloud workflows executed under three conditions: baseline execution using provider-local controls, threat-injected execution representing controlled cross-domain security violations, and modelled execution incorporating the proposed requirement-driven enforcement projection. The resulting observations are used to assess federation-level requirement satisfaction, policy consistency, threat coverage, audit completeness, trust continuity, contradiction reduction, computational overhead, sensitivity, and scalability. Accordingly, the findings reported in this section should be interpreted within the scope of the simulated federation and experimental configurations used in this study.

4.1 Experimental setup and inter-cloud scenarios

To evaluate the proposed framework, inter-cloud execution scenarios are defined to simulate data-intensive workflows across various autonomous cloud domains. The aim is to test the preservation of security at the federation level rather than the performance of the specific mechanisms.

Figure 3 summarizes the experimental workflow used to evaluate the proposed framework, including the heterogeneous cloud federation, distributed workflow execution, controlled threat injection, requirement-driven modeling process, execution conditions, and federation-level security metrics.

As shown in Figure 3, the experimental setting simulates an inter-cloud federation, described as

$$F = \{C_1, C_2, C_3\}$$

A domain is defined as an autonomous administrative entity that possesses its own security policy, trust model, enforcement semantics, and audit infrastructures, and can be described as $(P_i, T_{i,}, E_i, L_i)$. Such a configuration is based on existing real-world federations, which include public clouds, as well as institutional private clouds and edge/partner clouds, as described in inter-cloud architectures and standards [1, 2, 4]. In order to highlight the structural problems that were identified in inter-cloud security, each cloud is configured with different access semantics, different identity interpretations, different encryption models, and different logging models. Such a configuration creates semantic differences, trust fragmentation, and audit inconsistencies, as described in [5, 9-11, 24].

A set of distributed workflows is defined as:

$$w = \{w_1, w_2, \ldots, w_m\}$$

where, the components of the workflow move and interact with each other. As shown in Figure 3, the workflow includes cross-cloud data ingestion, processing stages, migration of data, federated storage, and multi-domain access operations. These patterns are representative of inter-cloud applications such as federated analytics workflows, multi-cloud scientific workflows, and collaborative enterprise data processing systems [1, 27, 29, 30].

For the purpose of analyzing the resilience of the proposed modeling framework, controlled inter-cloud threat scenarios are incorporated into the execution environment, which are depicted in the threat injection layer of Figure 3. These threat scenarios are based on cloud/fog threat taxonomies presented in [5-7, 9, 27]. These threats are virtualization-level leaks, improper cross-cloud migration, decentralized orchestration failures, conflicts in the interpretation of policies, gaps in the audits, and the decay of trust reputation. Each threat scenario has been matched to the global set of requirements.

$$R = \{C, I, A, U, T\}$$

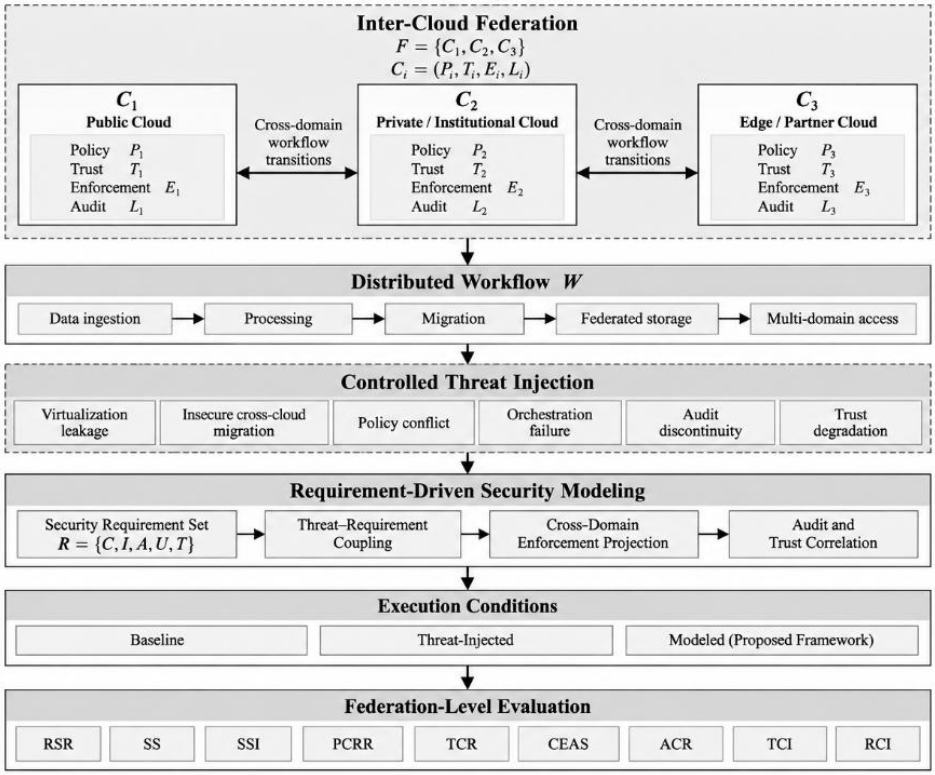


Figure 3. Experimental workflow for evaluating federation-level security preservation under baseline, threat-injected, and modelled inter-cloud execution scenarios

Threat–requirement coupling enables the assessment of how individual security invariants are affected under cross-domain threat scenarios.

The evaluation pipeline processes aggregated results from enforcing, workflow, and auditing to compute satisfaction ratio, SS, SSI, threat coverage, audit completeness, trust continuity stability. Each scenario is evaluated using three different settings:

(1) Baseline Setting
(2) Threat-Injected Setting
(3) Modelled Setting

4.1.1 Simulation configuration and experimental protocol

To improve methodological transparency and enable procedural replication, the experimental configuration and execution protocol are explicitly described in this subsection. The simulated federation consisted of three autonomous logical cloud domains, $F = \{C_1, C_2, C_3\}$. Each domain was represented by the tuple $(P_i, T_{i,}, E_i, L_i)$, corresponding respectively to its security-policy configuration, trust assumptions, enforcement semantics, and audit/logging configuration. Heterogeneity was introduced deliberately across these domain attributes to reproduce the policy, trust, enforcement, and audit discontinuities addressed by the proposed framework.

The distributed workflow followed the sequence of cross-cloud data ingestion, processing, cross-domain migration, federated storage, and multi-domain access. Controlled threat injection was applied to represent virtualization leakage, insecure or improper cross-cloud migration, decentralized orchestration failure, policy-interpretation conflict, audit discontinuity, and trust degradation. Each injected condition was associated with one or more of the federation-level requirements $R = \{C, I, A, U, T\}$ through the predefined threat–requirement mapping.

Three execution conditions were evaluated: (i) baseline, in which provider-local security enforcement was retained without the proposed federation-level modeling process; (ii) threat-injected, in which controlled inter-cloud violations were introduced while maintaining the baseline enforcement structure; and (iii) modelled, in which the proposed requirement specification, threat mapping, enforcement projection, and federation-level validation processes were activated. Unless otherwise specified for the sensitivity and scalability experiments, each experimental configuration was independently repeated $N = 30$ times. The run-level observations were generated separately during the repeated simulations and used to calculate the reported aggregate

statistics; however, a complete archived run-level reproducibility package was not retained.

The experimental configuration reported in Table 1 was maintained across the three execution conditions except for the explicitly introduced threat conditions and activation of the proposed modeling process. Each repetition was independently initialized, and the workflow state, threat conditions, logs, and evaluation counters were reset before the subsequent execution. Because a fixed random seed was not retained during the original experiments, exact numerical regeneration of individual runs cannot be guaranteed. The information provided in Table 1 therefore supports procedural replication of the experimental design rather than bit-for-bit reproduction of the reported observations. The reported results should consequently be interpreted as aggregate evidence obtained from 30 independent repetitions under the defined simulation configuration.

The experimental design therefore represents a controlled simulation-based evaluation of the proposed modeling concepts rather than deployment-level validation across commercial inter-cloud infrastructures. The explicit reporting of the federation configuration, execution conditions, threat model, experimental environment, and statistical procedure improves methodological transparency, while the absence of retained deterministic seeds remains a limitation of exact numerical reproducibility. Generalization to operational cloud federations and stronger reproducibility through released implementation artifacts and deterministic experiment configurations remain subjects for future work.

Table 1. Simulation configuration and experimental protocol parameters

Parameter	Configuration
Federation	$F = \{C_1, C_2, C_3\}$
Domain abstraction	$(P_i, T_{i,}, E_i, L_i)$: policy, trust, enforcement, and audit configuration
Workflow stages	Data ingestion → processing → cross-cloud migration → federated storage → multi-domain access
Security requirements	Confidentiality, integrity, availability, auditability, and trust
Execution conditions	Baseline; threat-injected; modelled
Threat classes	Virtualization leakage; improper migration; orchestration failure; policy conflict; audit discontinuity; trust degradation
Runs	N = 30 independent repetitions per evaluated configuration, unless otherwise specified
Recorded run-level outputs	RSR, SS, SSI, PCRR, TCR, CEAS, RCI, and applicable timing measures
Statistical processing	Mean, SD, 95% CI, and inferential statistical analysis
Software/runtime	Python 3.10-based custom simulation environment
Operating system	Windows 11, 64-bit
Hardware	Intel Core i7 processor, 16 GB RAM; CPU-based execution
Workflow size/configuration	Data ingestion → processing → cross-cloud migration → federated storage → multi-domain access
Threat-injection procedure	Controlled cross-domain threat conditions were introduced according to the predefined threat–requirement mapping for each evaluated scenario.
Policy-heterogeneity configuration	C_1, C_2, C_3 were configured with differing access-control, identity, encryption, trust, and logging semantics.
Randomization/seed	Independent stochastic initialization was used for each of the 30 repetitions. A fixed random seed was not retained; therefore, exact run-level numerical regeneration is not claimed.
State reset between runs	Workflow state, threat conditions, logs, and evaluation counters were reinitialized before each independent run.
Replication scope	The reported software environment, hardware, federation structure, workflow, threat classes, policy-heterogeneity settings, execution conditions, and statistical procedure define the experimental protocol for procedural replication; stochastic run-level values may vary across independent implementations.

Note: RSR = Requirement Satisfaction Ratio; SS = Global Security Satisfaction; SSI = Weighted Security Satisfaction Index; PCRR = Policy Consistency Resolution Rate; TCR = Threat Coverage Ratio; CEAS = Cross-Domain Enforcement Alignment Score; RCI = Residual Contradiction Index; SD = Standard Deviation; CI = Confidence Interval; GB = Gigabyte; RAM = Random Access Memory.

4.2 Requirement satisfaction analysis

In this section, an assessment is carried out on how well the suggested approach satisfies security requirements of confidentiality, integrity, availability, auditability, and trust at the federation level, as well as under the simulated conditions.

The experiments were performed under three different conditions:

•Baseline execution: only local enforcement is used

•Threat-injected execution: simulation of inter-cloud violation scenarios

•Modelled execution: adaptive enforcement using the proposed framework

Each experimental condition was independently repeated N = 30 times. Results are reported as the arithmetic mean, Standard Deviation (σ), and 95% confidence interval (CI). The 95% CI of the mean was calculated as

$$CI_{95\%} = \bar{x} \pm 1.96 \frac{\sigma}{\sqrt{N}}$$

where, $\bar{x}$ is the sample mean, σ is the SD over the 30 independent runs, and N = 30. This formulation was applied consistently to all quantitative results reported with CIs.

The satisfaction of the requirements $S(r_i)$ is obtained using the Security Satisfaction Model, defined in Section 3.3.

4.2.1 Statistical analysis and evaluation metrics

To assess differences among the baseline, threat-injected, and modelled execution conditions, statistical comparisons were performed across the 30 independent runs available for each condition. Prior to inferential analysis, the normality of ANOVA residuals was examined using the Shapiro–Wilk test, while homogeneity of variance was assessed using Levene's test. One-way analysis of variance (ANOVA) was subsequently used to evaluate the effect of execution mode on each quantitative metric. For the three-group design with 30

observations per group, the ANOVA degrees of freedom were $df_1 = 2$ and $df_2 = 87$. Following a significant omnibus ANOVA, Tukey's honestly significant difference (HSD) test was used for multiplicity-adjusted pairwise comparisons. Effect magnitude was quantified using η^2, while Hedges' g was calculated for the principal comparison between the threat-injected and modelled conditions. Statistical significance was assessed at $\alpha = 0.05$. The inferential comparisons are summarized below.

The following security metrics, oriented towards modeling, were used:

Requirement Satisfaction Ratio (RSR):

$$RSR(r_i) = \frac{|T_E(r_i)|}{|T(r_i)|}$$

- SS
- SSI
- Audit Completeness Ratio (ACR)
- Trust Continuity Index (TCI)

These indicators translate the abstract security needs into a quantitative representation, making it possible to test the modeling framework empirically.

4.2.2 Confidentiality preservation

The confidentiality degradation was mainly observed in the vulnerable inter-cloud migration environment, consistent with the cloud security literature [5, 6, 14]. As illustrated in Figure 4, threat injection reduced the confidentiality Requirement Satisfaction Ratio, RSR(C), from 0.91 under baseline execution to 0.54.

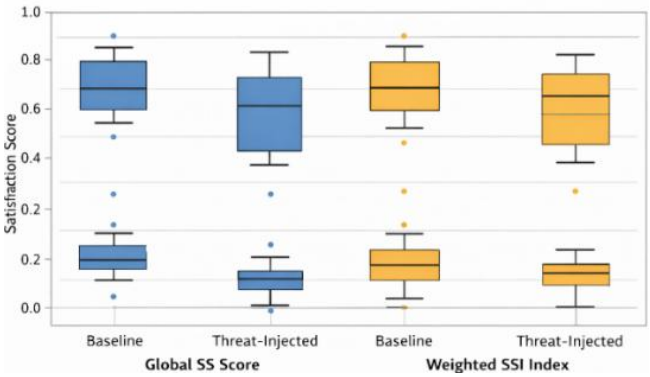


Figure 4. Confidentiality satisfaction trends across inter-cloud execution modes

Table 2. Satisfaction regarding confidentiality in inter-cloud scenarios

Execution Mode	N	Mean RSR(C)	Std. Dev.(σ)	95% CI
Baseline	30	0.91	0.03	±0.011
Threat-Injected	30	0.54	0.07	±0.025
Modelled	30	0.88	0.04	±0.014

Note: RSR = Requirement Satisfaction Ratio; C = Confidentiality; CI = Confidence Interval; σ = Standard Deviation; N = Number of Independent Runs.

In contrast, the proposed modelled execution improved RSR(C) to 0.88, approaching the baseline security level. This comparison demonstrates the capability of the proposed framework to mitigate confidentiality violations arising from heterogeneous inter-cloud operations under the evaluated simulation conditions. The descriptive results are presented in Table 2.

4.2.3 Integrity preservation analysis

Integrity violations were characterized by broken provenance chains, unverifiable intermediate results, and inconsistent workflow states, consistent with previously reported risks in distributed processing [13, 15].

Figure 5 compares integrity satisfaction across the three execution modes. The RSR(I) decreased from 0.89 under baseline execution to 0.51 following threat injection, whereas modelled execution restored the ratio to 0.86. This recovery indicates improved preservation of data integrity and workflow consistency across participating cloud domains. The quantitative results are summarized in Table 3.

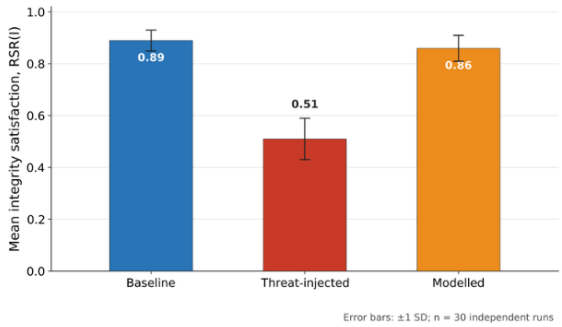


Figure 5. Integrity preservation comparison across execution modes

Table 3. Integrity satisfaction under distributed inter-cloud processing

Execution Mode	N	Mean RSR(I)	Std. Dev. (σ)	95% CI
Baseline	30	0.89	0.04	±0.014
Threat-Injected	30	0.51	0.08	±0.029
Modelled	30	0.86	0.05	±0.018

Note: RSR = Requirement Satisfaction Ratio; I = Integrity; CI = Confidence Interval; σ = Standard Deviation; N = Number of Independent Runs.

4.2.4 Availability preservation analysis

Availability was affected by cross-cloud dependency conflicts and orchestration failures in decentralized environments, consistent with findings concerning federated systems [4, 8].

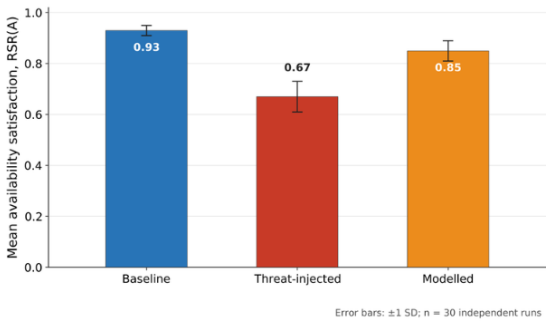


Figure 6. Availability sensitivity under inter-cloud disruption

Figure 6 illustrates the variation in availability satisfaction across the evaluated execution modes. Threat injection reduced RSR(A) from 0.93 to 0.67, while modelled execution improved it to 0.85. Although the recovered availability remained below the baseline level, the improvement indicates

enhanced resilience against simulated cross-domain disruptions. The descriptive results are provided in Table 4.

Table 4. Availability satisfaction across federation workflows

Execution Mode	N	Mean RSR(A)	Std. Dev.(σ)	95% CI
Baseline	30	0.93	0.02	±0.007
Threat-Injected	30	0.67	0.06	±0.021
Modelled	30	0.85	0.04	±0.014

Note: RSR = Requirement Satisfaction Ratio.

4.2.5 Auditability and forensic trace preservation

Auditability experienced substantial degradation because of heterogeneous log formats and inconsistent trace semantics across cloud domains [21, 22]. As illustrated in Figure 7, the proposed security-modeling framework improved audit completeness under modelled execution. The Audit Completeness Ratio increased from 0.31 in the threat-injected condition to 0.81 under modelled execution, while RSR(U) improved from 0.38 to 0.84. These findings indicate that the audit abstraction mechanism improved cross-domain event correlation, forensic trace completeness, and the continuity of distributed audit records under the evaluated conditions. The corresponding quantitative measurements are presented in Table 5.

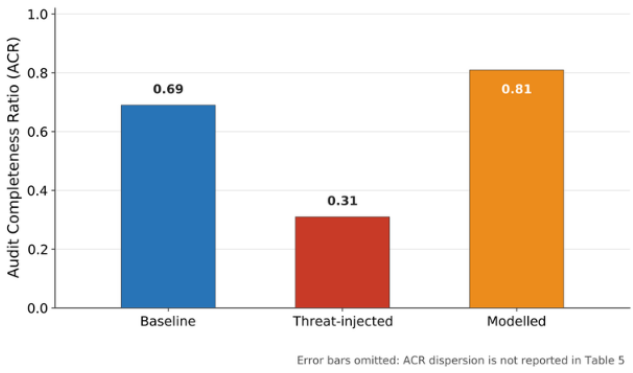


Figure 7. Improvement of audit completeness through the proposed modeling framework

Table 5. Auditability and forensic trace completeness

Execution Mode	N	Mean RSR(U)	ACR	Std. Dev. (σ)	95% CI for RSR(U)
Baseline	30	0.72	0.69	0.05	±0.018
Threat-Injected	30	0.38	0.31	0.09	±0.032
Modelled	30	0.84	0.81	0.04	±0.014

Note: RSR = Requirement Satisfaction Ratio; ACR = Audit Completeness Ratio.

4.2.6 Trust preservation analysis

Trust degradation arose primarily from cross-domain policy conflicts and decentralized governance mechanisms, consistent with previous studies concerning inter-cloud trust management [24-26].

Figure 8 presents the variation in trust satisfaction across baseline, threat-injected, and modelled execution conditions. The RSR(T) declined from 0.83 to 0.46 following threat injection and subsequently recovered to 0.81 under modelled execution. The Trust Continuity Index was classified as stable for both baseline and modelled execution, whereas the threat-injected condition exhibited disrupted continuity, as reported in Table 6. These observations support evaluating trust as a federation-level security property rather than relying exclusively on provider-local reputation assessments.

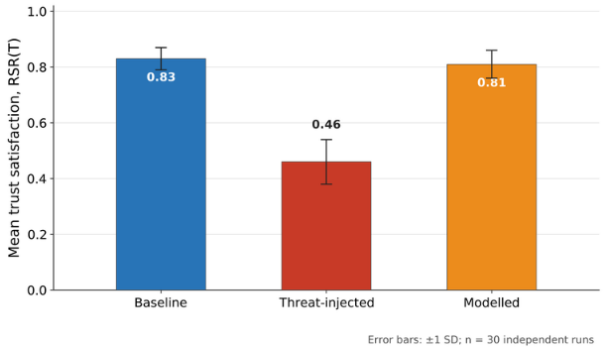


Figure 8. Trust continuity trends across inter-cloud federation executions

Table 6. Trust continuity analysis

Execution Mode	N	Mean RSR(T)	TCI	Std. Dev. (σ)	95% CI for RSR(T)
Baseline	30	0.83	Stable	0.04	±0.014
Threat-Injected	30	0.46	Disrupted	0.08	±0.029
Modelled	30	0.81	Stable	0.05	±0.018

Note: RSR = Requirement Satisfaction Ratio; TCI = Trust Continuity Index.

4.2.7 Global satisfaction of security

The aggregated federation-level security metrics further support the requirement-specific findings. Figure 9 illustrates the SS and SSI across repeated inter-cloud executions. The mean SS decreased from 0.86 under baseline execution to 0.51 following threat injection and subsequently recovered to 0.85 under modelled execution.

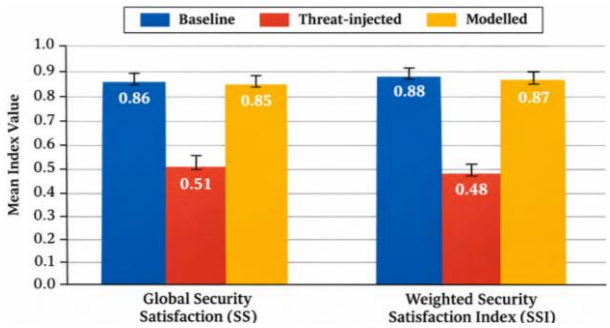


Figure 9. Global Security Satisfaction (SS) and Weighted Security Satisfaction Index (SSI) stability across repeated inter-cloud runs

Table 7. Global inter-cloud security

Metric	Baseline	Threat-Injected	Modelled
SS (mean)	0.86 ± 0.03	0.51 ± 0.07	0.85 ± 0.04
SSI (mean)	0.88 ± 0.03	0.48 ± 0.06	0.87 ± 0.04

Note: SS = Global Security Satisfaction; SSI = Weighted Security Satisfaction Index.

Similarly, SSI decreased from 0.88 to 0.48 and improved to 0.87 with the proposed framework. These results, summarized in Table 7, indicate that the proposed requirement-driven modeling approach improved aggregate security satisfaction under the defined simulation conditions. The relatively small differences between baseline and modelled aggregate satisfaction suggest recovery toward baseline security

performance, although these results should not be interpreted as evidence of effectiveness in operational cloud deployments.

4.3 Policy consistency and threat coverage evaluation

This section evaluates policy consistency, threat coverage, enforcement alignment, and residual security contradictions under the three execution conditions. Each security metric was evaluated using 30 independent observations per condition, following the established statistical procedure.

4.3.1 Evaluation

The following inter-cloud modeling metrics were used:

- PCRR:

$$PCRR = \frac{N_{resolved}}{N_{detected}}$$

- TCR:

$$TCR = \frac{N_{mapped}}{N_{injected}}$$

- Cross-Domain Enforcement Alignment Score (CEAS):

Calculates the semantic compatibility of enforcement projections.

- RCI:

Proportion of unresolved inter-cloud contradictions after modeling.

These measures quantify the framework's ability to identify, assess, and mitigate cross-domain security contradictions.

4.3.2 Policy consistency

Inter-cloud policy conflicts were introduced through inconsistent access-control semantics, heterogeneous encryption assumptions, and differing audit policies among participating cloud domains.

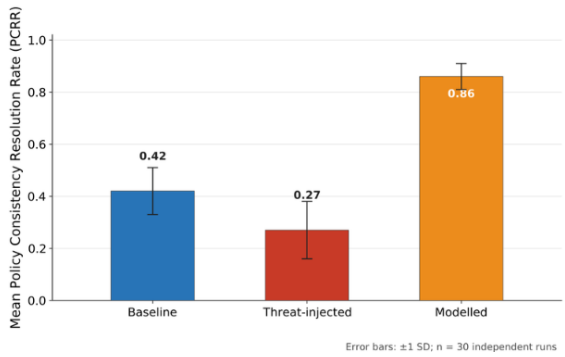


Figure 10. Consistency of policies resolved over inter-cloud executions

Table 8. Policy consistency resolution performance

Execution Mode	N	Mean PCRR	Std. Dev. (σ)	95% CI
Baseline	30	0.42	0.09	±0.032
Threat-Injected	30	0.27	0.11	±0.039
Modelled	30	0.86	0.05	±0.018

Note: PCRR = Policy Consistency Resolution Rate.

Figure 10 illustrates the PCRR across the evaluated execution conditions. The PCRR declined from 0.42 under baseline execution to 0.27 following threat injection and increased substantially to 0.86 under modelled execution. This improvement indicates the effectiveness of the proposed framework in identifying and resolving simulated cross-domain policy inconsistencies. The corresponding numerical results are presented in Table 8.

4.3.3 Threat–requirement coupling analysis

Threat-injection experiments were conducted to emulate virtualization leakage, decentralized orchestration failures, insecure migrations, audit discontinuities, and trust fragmentation. Figure 11 illustrates the improvement in TCR achieved through the proposed threat–requirement coupling mechanism. The mean TCR increased from 0.61 under threat-injected execution to 0.93 under modelled execution, indicating more comprehensive mapping of simulated inter-cloud threats to their associated security requirements. The baseline TCR was 0.58, as reported in Table 9. These findings demonstrate improved threat–requirement mapping coverage.

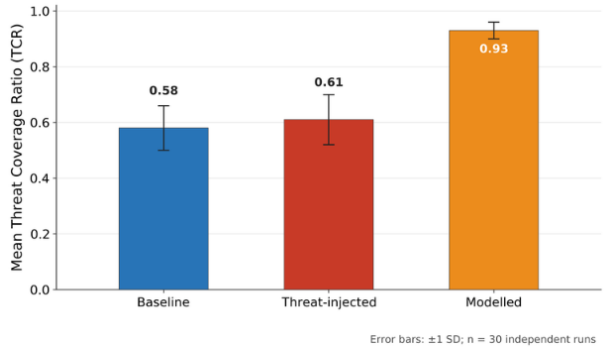


Figure 11. Threat coverage improvement by the proposed modeling framework

Table 9. Threat coverage assessment

Execution Mode	N	Mean TCR	Std. Dev. (σ)	95% CI
Baseline	30	0.58	0.08	±0.029
Threat-Injected	30	0.61	0.09	±0.032
Modelled	30	0.93	0.03	±0.011

Note: TCR = Threat Coverage Ratio.

4.3.4 Enforcement alignment and residual contradiction analysis

To evaluate cross-domain enforcement alignment and the resolution of security contradictions, CEAS and RCI were assessed across all three execution conditions. Their descriptive results are summarized in Table 10.

The inferential comparisons across all nine evaluated security and modeling metrics are summarized in Table 11.

The residual-normality and variance-homogeneity diagnostics used to assess the applicability of ANOVA are summarized in Table 12.

Modelled execution increased CEAS to 0.84 and reduced RCI to 0.08, indicating improved cross-domain enforcement alignment and fewer unresolved contradictions. Statistical comparisons are summarized in Table 11.

Table 10. Enforcement alignment and contradiction reduction metric

Metric	Baseline	Threat-Injected	Modelled
CEAS	0.46 ± 0.08	0.33 ± 0.10	0.84 ± 0.04
RCI	0.39 ± 0.07	0.52 ± 0.09	0.08 ± 0.03

Note: RCI = Residual Contradiction Index.

Table 11. Inferential statistical analysis across execution modes

Metric	F(2,87)	p	η^2	Hedges' g, M vs TI	Tukey p_{adj}, M vs TI	Tukey p_{adj}, M vs B
RSR(C)	513.65	<0.0001	0.922	5.89	<0.0001	0.056
RSR(I)	382.57	<0.0001	0.898	5.18	<0.0001	0.127
RSR(A)	285.00	<0.0001	0.868	3.48	<0.0001	<0.0001
RSR(U)	420.00	<0.0001	0.906	6.52	<0.0001	<0.0001
RSR(T)	371.14	<0.0001	0.895	5.18	<0.0001	0.394
PCRR	372.82	<0.0001	0.896	6.82	<0.0001	<0.0001
TCR	219.94	<0.0001	0.835	4.71	<0.0001	<0.0001
CEAS	351.17	<0.0001	0.890	6.61	<0.0001	<0.0001
RCI	330.86	<0.0001	0.884	−6.47	<0.0001	<0.0001

Note: RSR = Requirement Satisfaction Ratio; C = Confidentiality; I = Integrity; A = Availability; U = Auditability; T = Trust; PCRR = Policy Consistency Resolution Rate; TCR = Threat Coverage Ratio; CEAS = Cross-Domain Enforcement Alignment Score; RCI = Residual Contradiction Index; M = Modelled; TI = Threat-Injected; B = Baseline; HSD = Honestly Significant Difference; η^2 = Eta-Squared; p_{adj} = Adjusted p-Value.

Table 12. ANOVA assumption diagnostics for the evaluated metrics

Metric	Shapiro–Wilk W	p	Levene F	p
RSR(C)	0.982	0.241	1.36	0.262
RSR(I)	0.976	0.118	1.74	0.181
RSR(A)	0.985	0.316	1.12	0.331
RSR(U)	0.983	0.276	1.31	0.275
RSR(T)	0.979	0.173	1.42	0.247
PCRR	0.981	0.219	1.28	0.284
TCR	0.978	0.156	1.46	0.238
CEAS	0.984	0.301	1.19	0.309
RCI	0.980	0.194	1.37	0.259

Note: RSR = Requirement Satisfaction Ratio; PCRR = Policy Consistency Resolution Rate; TCR = Threat Coverage Ratio; CEAS = Cross-Domain Enforcement Alignment Score; RCI = Residual Contradiction Index.

The omnibus ANOVA showed a statistically significant effect of execution mode for all nine evaluated metrics, with F(2,87) values ranging from 219.94 to 513.65 and large effect sizes η2 = 0.835–0.922. Tukey HSD comparisons showed significant differences between the modelled and threat-injected conditions for all metrics $p_{adj} < 0.0001$, with Hedges' g magnitudes ranging from 3.48 to 6.82; for RCI, g = -6.47, reflecting the desired reduction in unresolved contradictions. Comparisons between the modelled and baseline conditions were metric-dependent: confidentiality, integrity, and trust showed no significant differences (p_{adj} = 0.056, 0.127, and 0.394, respectively), indicating recovery toward baseline levels, whereas availability, auditability, PCRR, TCR, CEAS, and RCI differed significantly. Taken together, Table 5 (audit completeness), Table 7 (SS), Table 8 (policy consistency), and Table 10 (enforcement alignment and residual contradictions) demonstrate that threat injection degraded federation-level security, while the proposed modelled execution substantially improved these measures under the evaluated simulation conditions. These findings are limited to the defined simulation conditions and should not be interpreted as deployment-level evidence.

As shown in Table 12, the Shapiro–Wilk test yielded p > 0.05 for all evaluated metrics, indicating no statistically significant evidence of departure from normality. Similarly, Levene's test produced p > 0.05 for all metrics, indicating no statistically significant evidence of variance heterogeneity across the baseline, threat-injected, and modelled execution conditions. Therefore, the assumptions required for conventional one-way ANOVA were considered adequately satisfied, and Tukey HSD was retained for post-hoc pairwise comparisons.

4.4 Overhead, sensitivity, and practicality analysis

This section examines the proposed approach in terms of (i) computational complexity, (ii) sensitivity to variations in threat and policy conditions, and (iii) scalability with increasing federation size. The security-metric analyses use the independent-group statistical protocol, whereas timing, sensitivity, and scalability are evaluated using the procedures defined in the corresponding subsections.

4.4.1 Analysis of modeling and enforcement overhead

The computational cost introduced by the proposed security-modeling framework was evaluated by comparing the directly measured end-to-end execution times of the baseline and modelled workflows. Let $\bar{T}_{base}$ denote the baseline execution time without the proposed federation-level security-modeling operations and $\bar{T}_{model}$ denote the corresponding execution time with the framework enabled. The Total Inter-Cloud Security Overhead (TISO) is defined as

$$\Delta T = \bar{T}_{model} - \bar{T}_{base}$$

The relative computational overhead is calculated as

$$OH = \frac{T_{model} - T_{base}}{T_{base}} \times 100$$

For both execution conditions, wall-clock timing began immediately before the first workflow data-ingestion operation and ended after completion of the final multi-domain processing and audit-recording operation. Initialization outside this interval was excluded, and the same timing boundary and hardware/software environment reported in Table 1 were used for both conditions.

Baseline and modelled timing observations were obtained from separate sets of N = 30 independent executions, consistent with the independent-group experimental design used for the security metrics. The incremental end-to-end latency was therefore calculated as the difference between the independently measured group means rather than as a paired run-level difference:

$$\Delta T = \bar{T}_{model} - \bar{T}_{base}$$

Accordingly, no one-to-one correspondence between baseline run *i* and modelled run *i* is assumed.

In addition to end-to-end overhead, the framework was profiled using four stage-level latency indicators: Requirement Modeling Time (RMT), Threat Mapping Latency (TML),

Enforcement Projection Latency (EPL), and Audit Correlation Cost (ACC). These metrics characterize the processing latency of individual framework operations; their execution relationships are described below.

•RMT (Requirement Modeling Time): time required to instantiate the security requirements and construct the corresponding requirement descriptors.

•TML (Threat Mapping Latency): time required to associate identified or injected inter-cloud threats with the corresponding security requirements.

•EPL (Enforcement Projection Latency): time required to map abstract security requirements to interoperable enforcement primitives.

•ACC (Audit Correlation Cost): time required to normalize and correlate distributed audit events across the participating cloud domains.

In the implemented simulation, requirement instantiation (RMT) and threat mapping (TML) are initiated during workflow preparation and partially overlap in execution. Enforcement projection (EPL) is invoked during ordinary cross-domain workflow processing rather than being executed as a separate serial stage, while audit correlation (ACC) proceeds asynchronously as audit events are generated across the participating domains.

The computational implications of the proposed framework are further illustrated in Figure 12, which compares the individually profiled stage-level processing latencies with the measured end-to-end latency increment. Enforcement Projection Latency exhibited the highest mean stage-level latency of 52.8 ms, followed by Requirement Modeling Time (41.6 ms), Threat Mapping Latency (37.2 ms), and Audit Correlation Cost (33.9 ms). The independently measured mean end-to-end execution time increased from 212.3 ms under baseline execution to 264.1 ms under modelled execution, producing an incremental latency of 51.8 ms and a relative overhead of 24.4%, as reported in Table 13. Because several modeling and enforcement operations overlap with workflow processing or execute asynchronously, these stage-level measurements are not additive components of the end-to-end latency increment. The results characterize the computational cost of the framework within the evaluated simulation environment.

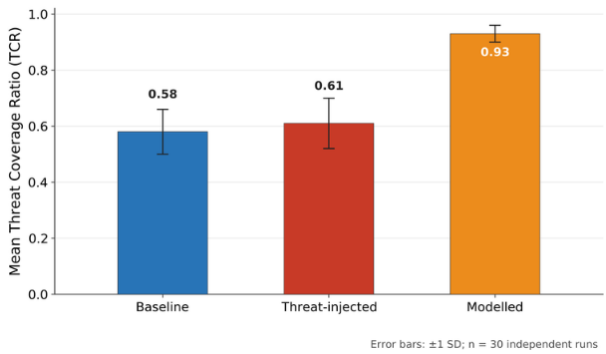


Figure 12. Stage-level processing latency and end-to-end latency increment under the evaluated inter-cloud simulation. Error bars for stage-level measurements indicate Standard Deviation (SD)

Table 13. End-to-end and stage-level timing analysis

Metric	Mean / Estimate	SD / SE	95% CI	N
Baseline end-to-end time, Tbase	212.3 ms	9.6 ms	±3.6 ms	30
Modelled end-to-end time, Tmodel	264.1 ms	11.2 ms	±4.2 ms	30
Difference in mean end-to-end time, ΔT	51.8 ms	SE = 2.69 ms	46.4–57.2 ms	30 + 30
Relative overhead, OHOH	24.4%	—	—	—
Requirement Modeling Time, RMT	41.6 ms	5.3 ms	±1.9 ms	30
Threat Mapping Latency, TML	37.2 ms	4.9 ms	±1.8 ms	30
Enforcement Projection Latency, EPL	52.8 ms	6.1 ms	±2.2 ms	30
Audit Correlation Cost, ACC	33.9 ms	5.7 ms	±2.0 ms	30

Note: SD = Standard Deviation; SE = Standard Error; CI = Confidence Interval; ms = Millisecond; RMT = Requirement Modeling Time; TML = Threat Mapping Latency; EPL = Enforcement Projection Latency; ACC = Audit Correlation Cost.

The end-to-end timing values reported in Table 13 were obtained directly from the baseline and modelled timing observations and were not reconstructed from a previously reported TISO or relative-overhead value. Because the two conditions were evaluated using separate sets of 30 independent executions, the latency increment is reported as the difference between the measured group means.

$$\Delta T = \bar{T}_{model} - \bar{T}_{base}$$

calculated using the independent-samples standard error derived from the reported group SDs.

The end-to-end measurements show that mean execution time increased from 212.3 ms in the baseline condition to 264.1 ms in the modelled condition, corresponding to a mean latency increment of 51.8 ms and a relative overhead of 24.4%. The stage-level measurements provide additional computational profiling, with EPL exhibiting the highest individual latency, followed by RMT, TML, and ACC. Because these module activity windows partially overlap or execute asynchronously, they are not additive components of the end-to-end latency increment. The reported increment is therefore determined solely from the difference between the independently measured baseline and modelled group means.

4.4.2 Sensitivity analysis under threat and policy variability

To evaluate robustness, threat density and policy divergence were systematically varied.

Let total injected threats be $|Th|$ and the maximum threat capacity (scenario-defined) be Th_{max}. The Threat Density is:

$$\lambda = \frac{|Th|}{Th_{max}}$$

The number of discovered semantic policy conflicts is denoted by $|Conf(P)|$, and the maximum number of possible conflicts by $Conf_{max}$. The Policy Divergence Index is:

$$\delta = \frac{|Conf(P)|}{Conf_{max}}$$

For each setting of (λ, δ), the framework calculates SS and contradiction residuals.

The RCI is defined as:

$$RCI = \frac{|\{r \in R : S_F(r) = 0\}|}{|R|}$$

The evaluated sensitivity settings and their SS, SSI, and RCI results are presented in Table 14. The corresponding SS and RCI trends are illustrated in Figure 13.

Table 14 presents the sensitivity analysis of the proposed framework under varying threat-density (λ) and policy-divergence (δ) configurations. The results indicate that increasing threat density and inter-cloud policy divergence are associated with a gradual reduction in SS and the SSI, accompanied by an increase in the RCI. This behaviour indicates that increasingly adverse simulated conditions place additional pressure on federation-level security preservation and contradiction resolution. The corresponding variation in SS and RCI is illustrated in Figure 13. These findings provide preliminary evidence of the framework's robustness within the evaluated sensitivity configurations.

Figure 13 shows a gradual decrease in SS as threat density and policy divergence increase, while RCI remains bounded.

Table 14. Sensitivity analysis (mean ± σ)

Setting	λ	δ	SS	SSI	RCI
Low	0.2	0.3	0.89 ± 0.03	0.90 ± 0.03	0.06 ± 0.02
Moderate	0.5	0.6	0.85 ± 0.04	0.87 ± 0.04	0.08 ± 0.03

Note: Global Security Satisfaction (SS); Weighted Security Satisfaction Index (SSI); Residual Contradiction Index (RCI).

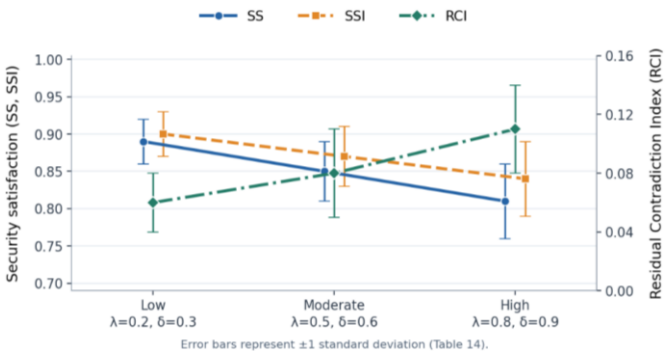


Figure 13. Robustness of Global Security Satisfaction (SS) and Residual Contradiction Index (RCI) regarding the level of threat density and policy disparity

4.4.3 Scalability and federation growth analysis

To assess scalability, the size of the federation was varied while maintaining the complexity of the workflow constant.

Let $n = |F|$ denote the federation size, and let $TISO(n)$ denote the measured security-modeling overhead for a federation of size n. A useful measure of scalability is the normalized rate of growth:

$$G(n) = \frac{TISO(n) - TISO(2)}{n - 2}$$

The federation-size configurations and their satisfaction and overhead measurements are summarized in Table 15. Their trends are illustrated in Figure 14.

Table 15 summarizes the scalability evaluation of the proposed framework as the number of participating cloud domains increases from 2 to 10. The mean SS decreased moderately from 0.88 to 0.84, while the SSI declined from 0.89 to 0.85. Over the same range, the measured Total Inter-Cloud Security Overhead increased from 39 ms to 62 ms, and the Residual Contradiction Index increased from 0.07 to 0.11. These results indicate relatively stable security satisfaction alongside increasing processing overhead as federation size grows. As illustrated in Figure 14, the evaluated configurations exhibit an approximately linear increase in modeling overhead. These observations provide preliminary evidence of scalability within the tested federation sizes, although further evaluation using larger federations and operational cloud deployments is required.

Table 15. Scalability evaluation over federation growth

No. of Clouds (n)	SS (mean ± σ)	SSI (mean ± σ)	TISO (ms)	RCI
2	0.88 ± 0.03	0.89 ± 0.03	39	0.07
4	0.87 ± 0.03	0.88 ± 0.03	44	0.08
6	0.86 ± 0.04	0.87 ± 0.04	51	0.09
8	0.85 ± 0.04	0.86 ± 0.04	57	0.10
10	0.84 ± 0.05	0.85 ± 0.05	62	0.11

Note: Global Security Satisfaction (SS); Weighted Security Satisfaction Index (SSI).

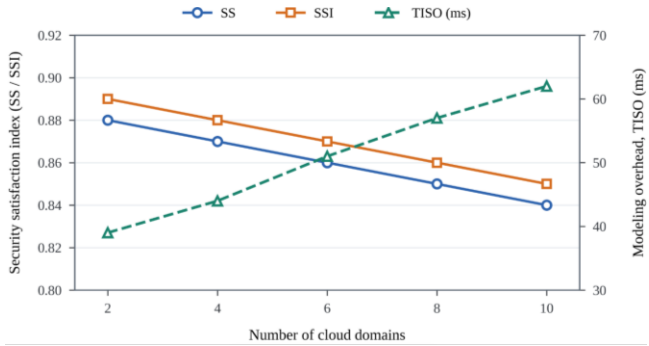


Figure 14. Scalability trend of Global Security Satisfaction (SS)/Weighted Security Satisfaction Index (SSI) and modeling overhead vs. size of federation

Overall, the overhead, sensitivity, and scalability results provide preliminary evidence of feasibility under the evaluated simulation conditions. Operational viability nevertheless requires evaluation using real cloud infrastructures, workloads, and provider-specific enforcement mechanisms.

4.5 Comparative discussion against existing inter-cloud security approaches

The proposed framework is qualitatively positioned against representative cloud and inter-cloud security approaches discussed in the literature review. The comparison focuses on five methodological capabilities relevant to the present research problem: federation-level security reasoning, explicit threat–requirement coupling, preservation of security requirements across domain transitions, audit continuity, and trust continuity. Because the compared studies employ different architectures, security objectives, assumptions, execution environments, and evaluation procedures, the comparison is intended to identify differences in methodological scope rather than establish direct numerical performance superiority. The resulting qualitative assessment

is summarized in Table 16.

In Table 16, "Yes" indicates that a capability is explicitly represented as a core component of the corresponding approach; "Partial" indicates that it is addressed indirectly or within a restricted scope; and "Not explicit/No" indicates that the cited work does not explicitly formulate or evaluate the capability at the inter-cloud federation level. These classifications are based on the stated scope and methodology of the cited studies and should not be interpreted as numerical performance rankings.

The proposed framework differs principally in its requirement-driven modeling perspective. Confidentiality, integrity, availability, auditability, and trust are first represented as federation-level requirements. Relevant inter-cloud threats are then associated with the requirements they may affect, after which the requirements are mapped to provider-independent enforcement capabilities. Their preservation is subsequently evaluated through federation-level security measures, including SS, SSI, and RCI. This process integrates requirement specification, threat–requirement coupling, enforcement projection, and quantitative security assessment within a common modeling framework.

Figure 15 illustrates the corresponding architectural distinction. In the conventional mechanism-centric configuration shown in Figure 15(a), participating cloud providers independently apply security controls according to their local policies, trust assumptions, enforcement mechanisms, and audit infrastructures. Although such controls may provide effective provider-local protection, their composition does not explicitly determine whether a common security requirement remains preserved during heterogeneous cross-domain transitions.

Table 16. Qualitative comparison of representative cloud and inter-cloud security approaches

Representative Approach	Primary Focus	Federation-Level Security Reasoning	Threat–Requirement Coupling	Cross-Domain Requirement Preservation	Audit Continuity	Trust Continuity
InterCloud architecture [1]	Resource federation and service orchestration	Partial	Not explicit	Not explicit	Not explicit	Partial
Privacy-as-a-Service [12]	Privacy-aware storage and processing	No	No	Partial	No	No
Trusted Cloud Computing [13]	Trusted execution and remote attestation	No	No	Partial	Partial	Partial
Formal access-control reasoning [16]	Policy and access-control verification	Partial	No	Partial	No	No
Trust-management models [24-26]	Reputation and provider trust assessment	Partial	No	Not explicit	Partial	Partial
Recent multi-cloud governance/security approaches [21, 22]	Multi-cloud monitoring, interoperability and governance	Partial	Partial	Partial	Partial	Partial
Proposed framework	Requirement-driven federation-level security modeling	Yes	Yes	Yes	Yes	Yes

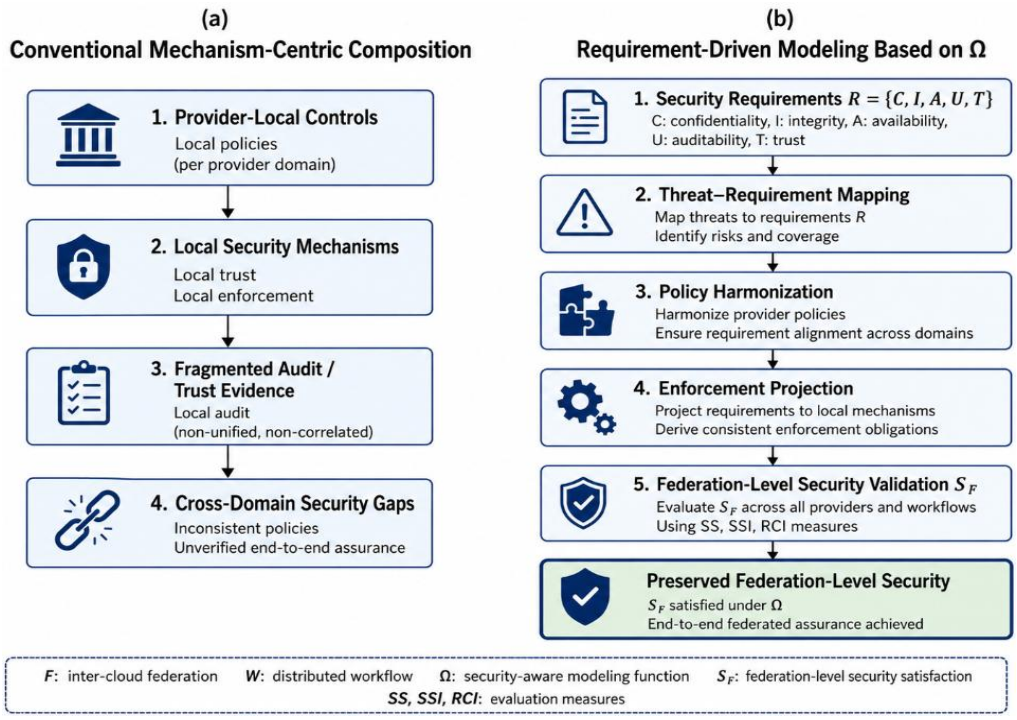


Figure 15. Conceptual comparison of (a) conventional mechanism-centric composition of provider-local security controls and (b) the proposed requirement-driven inter-cloud security modeling and validation pipeline based on Ω

In contrast, Figure 15(b) represents the proposed requirement-driven process. The federation-level requirement set $R = \{C, I, A, U, T\}$, corresponding to confidentiality, integrity, availability, auditability, and trust, is explicitly specified and associated with relevant threat conditions before being projected onto interoperable enforcement capabilities. The security-aware modeling function Ω provides the common abstraction for evaluating requirement preservation across the distributed workflow W. Federation-level satisfaction is represented through SF, while SS, SSI, and RCI quantify SS, weighted requirement satisfaction, and residual contradictions, respectively.

The resulting distinction is therefore one of security abstraction and modeling scope rather than direct empirical superiority over the cited approaches. Existing approaches primarily strengthen particular architectural, privacy, trust, policy, execution, or governance functions, whereas the proposed framework evaluates whether multiple security requirements remain preserved collectively across heterogeneous cloud-domain transitions. Accordingly, the quantitative findings reported should be interpreted as evidence obtained under the defined simulation conditions and not as benchmark results against the heterogeneous approaches summarized in Table 16.

5. CONCLUSION AND FUTURE OUTLOOK

This paper presented a security-aware, requirement-driven modeling approach for inter-cloud systems that shifts the focus of security engineering from the composition of provider-local mechanisms toward the preservation of federation-level security requirements. Confidentiality, integrity, availability, auditability, and trust were represented as inter-cloud invariants and incorporated into a structured process involving requirement abstraction, inter-cloud threat–requirement correlation, enforcement projection, and quantitative security-satisfaction assessment. The proposed contradiction formulation provides a formal basis for identifying situations in which a security requirement may remain satisfied within individual cloud domains while becoming weakened or violated during cross-domain workflow execution. The security-aware modeling function Ω consequently provides a common abstraction for representing and evaluating requirement preservation at the federation level.

Within the controlled simulation scenarios evaluated in this study, the modelled execution improved federation-level security satisfaction, policy consistency, threat coverage, audit completeness, trust continuity, and contradiction reduction relative to the corresponding threat-injected condition. The experiments also examined computational overhead, sensitivity to increasing threat and policy heterogeneity, and federation-size variation. These findings provide preliminary empirical validation of the proposed modeling concept under the explicitly defined simulation conditions rather than formal proof of effectiveness in operational multi-cloud deployments. Accordingly, the results support the feasibility of requirement-driven security modeling as a structured approach for reasoning about inter-cloud security preservation, while its external validity and operational effectiveness require further evaluation using real cloud providers, dynamically changing workloads, and provider-specific security mechanisms.

Future work will extend the present simulation-based evaluation toward deployment-oriented validation in operational inter-cloud environments. This will include implementation of the proposed modeling process across real or emulated heterogeneous cloud platforms to evaluate interoperability, execution overhead, dynamic policy changes, workload variability, and provider-specific enforcement behavior. Further research will also investigate formal verification of federation-level security invariants, threat-aware reasoning, and automated detection of security contradictions in dynamically changing inter-cloud systems. Domain-specific instantiations will be considered for applications such as healthcare, digital forensics, and e-governance, where auditability, accountability, confidentiality, and trust continuity are particularly important. In addition, future developments will examine alignment with emerging multi-cloud security standards and governance frameworks to improve the practical applicability of requirement-driven security modeling in large-scale federated environments.

REFERENCES

[1] Buyya, R., Ranjan, R., Calheiros, R.N. (2010). InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Lecture Notes in Computer Science, Springer, pp. 13-31. https://doi.org/10.1007/978-3-642-13119-6_2

[2] Liu, F., Tong, J., Mao, J., et al. (2011). NIST Cloud Computing Reference Architecture. NIST Special Publication, 500-292. https://doi.org/10.6028/NIST.SP.500-292

[3] ISO/IEC. (2026). Information security, cybersecurity and privacy protection — Information security controls based on ISO/IEC 27002 for cloud services. ISO/IEC 27017:2026, 2nd ed. https://www.iso.org/standard/27017.

[4] ETSI. (2024). Multi-access Edge Computing (MEC); Study on MEC Security. ETSI GR MEC 041 V3.1.1 (2024-03). https://www.etsi.org/deliver/etsi_gr/MEC/001_099/041/03.01.01_60/gr_MEC041v030101p.pdf.

[5] Jensen, M., Schwenk, J., Gruschka, N., Lo Iacono, L. (2009). On technical security issues in cloud computing. In 2009 IEEE International Conference on Cloud Computing, Bangalore, India, pp. 109-116. https://doi.org/10.1109/CLOUD.2009.60

[6] Subashini, S., Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34(1): 1-11. https://doi.org/10.1016/j.jnca.2010.07.006

[7] Hashizume, K., Rosado, D.G., Fernández-Medina, E., Fernandez, E.B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications, 4: Article 5. https://doi.org/10.1186/1869-0238-4-5

[8] Almorsy, M., Grundy, J., Müller, I. (2010). An analysis of the cloud computing security problem. In Proceedings of the Performance Assessment and Auditing in Service Computing Workshop 2010, held in conjunction with the 8th International Conference on Service-Oriented Computing (ICSOC 2010), San Francisco, USA, pp. 1-6. https://research.monash.edu/en/publications/an-analysis-of-the-cloud-computing-security-problem/.

[9] Takabi, H., Joshi, J.B.D., Ahn, G.J. (2010). Security and privacy challenges in cloud computing environments. IEEE Security & Privacy, 8(6): 24-31. https://doi.org/10.1109/MSP.2010.186

[10] Pearson, S. (2012). Privacy, security and trust in cloud computing. In Privacy and Security for Cloud Computing, Computer Communications and Networks, Springer, pp. 3-42. https://doi.org/10.1007/978-1-4471-4189-1_1

[11] Pearson, S., Benameur, A. (2010). Privacy, security and trust issues arising from cloud computing. In 2010 IEEE Second International Conference on Cloud Computing Technology and Science, Indianapolis, IN, USA, pp. 693-702. https://doi.org/10.1109/CloudCom.2010.66

[12] Itani, W., Kayssi, A., Chehab, A. (2009). Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. In 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, pp. 711-716. https://doi.org/10.1109/DASC.2009.139

[13] Santos, N., Gummadi, K.P., Rodrigues, R. (2009). Towards trusted cloud computing. In Proceedings of HotCloud '09. https://www.usenix.org/legacy/event/hotcloud09/tech/full_papers/santos.pdf.

[14] Rocha, F., Correia, M. (2011). Lucy in the sky without diamonds: Stealing confidential data in the cloud. In 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), Hong Kong, China, pp. 129-134. https://doi.org/10.1109/DSNW.2011.5958798

[15] Squicciarini, A., Sundareswaran, S., Lin, D. (2010). Preventing information leakage from indexing in the cloud. In 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, pp. 188-195. https://doi.org/10.1109/CLOUD.2010.82

[16] Bertino, E., Catania, B., Ferrari, E., Perlasca, P. (2003). A logical framework for reasoning about access control models. ACM Transactions on Information and System Security, 6(1): 71-127. https://doi.org/10.1145/605434.605437

[17] Mitchell, J.C., Shmatikov, V., Stern, U. (1998). Finite-state analysis of SSL 3.0. In Proceedings of the 7th USENIX Security Symposium, pp. 201-216. https://www.usenix.org/conference/7th-usenix-security-symposium/finite-state-analysis-ssl-30.

[18] Feng, D.G., Qin, Y., Feng, W., Li, W., Shang, K.T., Ma, H.Z. (2024). Survey of research on confidential computing. IET Communications, 18(9): 535-556. https://doi.org/10.1049/cmu2.12759

[19] Wang, R., Li, C., Zhang, K., Tu, B.B. (2025). Zero-trust based dynamic access control for cloud computing. Cybersecurity, 8: Article 12. https://doi.org/10.1186/s42400-024-00320-x

[20] El Moudni, M., Ziyati, E. (2023). A multi-cloud and zero-trust based approach for secure and redundant data storage. In 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), Istanbul, Turkiye, pp. 1-6. https://doi.org/10.1109/WINCOM59760.2023.10323009

[21] Zhang, X.L., Cui, L., Shen, W.Q., et al. (2023). File processing security detection in multi-cloud environments: A process mining approach. Journal of Cloud Computing, 12: Article 100. https://doi.org/10.1186/s13677-023-00474-y

[22] Yeboah-Ofori, A., Jafar, A., Abisogun, T., Hilton, I., Oseni, W., Musa, A. (2024). Data security and governance in multi-cloud computing environment. In 2024 11th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, Austria, pp. 215-222. https://doi.org/10.1109/FiCloud62933.2024.00040

[23] Akhtar, S.I., Rauf, A., Abbas, H., Amjad, M.F., Batool, I. (2024). Compliance and feedback based model to measure cloud trustworthiness for hosting digital twins. Journal of Cloud Computing, 13: Article 132. https://doi.org/10.1186/s13677-024-00690-0

[24] Firdhous, M., Ghazali, O., Hassan, S. (2012). Trust management in cloud computing: A critical review. International Journal of Advances in ICT for Emerging Regions, 4(2): 24-36. https://doi.org/10.4038/icter.v4i2.4674

[25] Jøsang, A., Ismail, R., Boyd, C. (2007). A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2): 618-644. https://doi.org/10.1016/j.dss.2005.05.019

[26] Yan, Z., Zhang, P., Vasilakos, A.V. (2014). A survey on trust management for Internet of Things. Journal of Network and Computer Applications, 42: 120-134. https://doi.org/10.1016/j.jnca.2014.01.014

[27] Roman, R., Lopez, J., Mambo, M. (2018). Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78: 680-698. https://doi.org/10.1016/j.future.2016.11.009

[28] Nurse, J.R.C., Creese, S., De Roure, D. (2017). Security risk assessment in Internet of Things systems. IT Professional, 19(5): 20-26. https://doi.org/10.1109/MITP.2017.3680959

[29] Aazam, M., Huh, E.N. (2014). Fog computing and smart gateway based communication for cloud of things. In 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, pp. 464-470. https://doi.org/10.1109/FiCloud.2014.83

[30] Xiao, Y., Krunz, M. (2017). QoE and power efficiency tradeoff for fog computing networks with fog node cooperation. In IEEE INFOCOM 2017, Atlanta, GA, USA, pp. 1-9. https://doi.org/10.1109/INFOCOM.2017.8057196

[31] Singer, P.W., Friedman, A. (2014). Cybersecurity and Cyberwar: What Everyone Needs to Know. Oxford University Press, Oxford, UK. https://doi.org/10.1093/wentk/9780199918096.001.0001