AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

Preventing Cross-Tenant Data Exposure in Financial Platforms Through AI-Assisted Security and Enterprise Architecture Controls

[1]*Kannan Meiappan
[1]Department of Architecture, Ascendion, Inc, USA.

Abstract

Modern financial platforms are increasingly offering banking, payments, wealth management, and other financial services through shared multi-tenant architectures. However, with the sharing of data among tenants, significant security concerns arise that may not be completely addressed by traditional controls. This paper introduces a framework for the enterprise architecture which combines tenant isolation, identity management, API gateway enforcement, secure identifiers, immutable audit logs, and AI-driven risk detection. Unlike previous approaches, which mainly focused on AI model creation, this framework views AI as an assistive overlay that enhances rather than substitutes for deterministic architectural controls. We have enumerated nine cross-tenant exposure vectors, outlined a layered architecture with eight levels of control, and demonstrated, using scenario-based metrics, the added security value in the use of AI-driven anomaly detection and policy violation analysis. The paper also provides implementation guidance, data privacy considerations, and adoption planning for financial platform architectures. The proposed framework brings together enterprise security architecture and AI-driven risk intelligence to enable financial systems to be resilient across multiple tenants.

Keywords: Multi-tenant security, Cross-tenant data exposure, AI-assisted security, Tenant isolation, RBAC, API gateway enforcement.

1. Introduction

Multi-tenant architectures are becoming a more common approach for financial institutions for software-as-a-service (SaaS) and platform-as-a-service (PaaS) in banking, payments, investment management, and insurance sectors. Hundreds or thousands of tenants (financial entities, corporate customers, partitions, etc.) may use a single platform. This will offer many economic and operational advantages through common infrastructure with lower per-tenant costs, faster feature rollout, centralized security management and logical separation between tenants (Eboseremen et al., 2022; Singh, 2019).

Multi-tenant systems are inherently shared, which presents a key challenge: the risk that a data breach can spread from one tenant to another (Hashim & Hussein, 2024; Ren et al., 2025). The risk of cross-tenant exposure is present in multi-tenant systems, and the high-profile incidents where attackers exploited subtle flaws in the architecture of systems, with missing tenant filters, broken object-level authorization, cache leakage, and misconfigured API policies, which cross-tenant boundaries and access sensitive financial data across tenant partitions, demonstrate the severity of this exposure risk (Manukonda, 2023).

The traditional approach to multi-tenant isolation is based on role-based access control and tenant filtering, which are inherently limited by the lack of real-time visibility into access attempts violating the tenancy rules, subtle tenant access drift, vulnerability to insider and credential risks, and fragility of implementation, as user access changes in one system can break isolation in others if the rules are not incorporated into every database query, API endpoint, and background job correctly (Reddy & Ratnam, 2021). When integrated into enterprise architecture controls, AI is not a one-size-fits-all solution but rather an assistant to those controls (Sharma, 2025). While deterministic architectural controls continue to be the most important way to enforce policy, AI-assisted controls could help mitigate some of these limitations by detecting risks and analyzing for policy violations. (Kumar, 2020).

This paper introduces an all-encompassing risk model comprising nine distinct types of cross-tenant exposures, an enterprise architecture framework featuring eight layers of business controls that integrate AI-assisted controls from tenant context to identity, API enforcement, data access,

Kannan Meiappan
Department of Architecture, Ascendion, Inc, USA.
Email: kannan.mei@outlook.com
ORCID ID: 0009-0002-2668-6082

Received: 6-Jul-2026 Revised: 29-Jul-2026 Accepted: 17-Aug-2026

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

secure identifiers, auditability, AI-assisted intelligence, and governance, a governance model that includes human-in-the-loop review, handling of false-positive alerts, explainability requirements, and compliance reporting, and summarizes how financial multi-tenant systems architects can apply AI-assisted controls in regulated financial environments (Bala, 2025; Teegala, 2017).

The remainder of this paper follows Section 2 reviews the background and related works, followed by the description of the cross-tenant exposure risk model in Section 3, the proposed enterprise architecture framework in Section 4, the description of the AI-assisted security controls in the form of architectural assistants in Section 5, the scenario-based evaluation in Section 6, the governance model in Section 7, the discussion of practical implications, limitations and adoption considerations in Section 8, and the conclusion and further directions in Section 9.

2. Background and Related Work

2.1 Multi-Tenant Platform Architecture

Financial platforms can achieve multi-tenancy by isolating at the schema level (multiple schemas in a single database), row-level (tenant identifier column with mandatory filtering), service-level (dedicated microservice instances per high-value tenant) or database level (dedicated databases per tenant, which is the most isolating option but also the most operationally costly) (Narlawar, 2025). With the increasing adoption of microservices, the context of a tenant needs to be passed through service boundaries, API gateways, message queues and background job processors, and consistency of tenant identification and context propagation must be maintained to ensure isolation integrity (Singh, 2019).

2.2 Tenant Isolation Models

There are different types of tenant isolation models that vary in the degree of isolation and the operational overhead they incur: A separate database provides the strongest isolation, but has the highest overhead and is commonly used when the tenant has high value. Table 1 shows that the separate schema provides high isolation with medium overhead, and is often used in regional deployments. Row-level, with a mandatory filter, provides medium isolation with low overhead and is widely adopted in the financial industry, whereas Service-level provides high isolation with high overhead and is typically used to isolate critical workloads (Pippal et al., 2024). The choice of which model to implement depends on the trade-offs among security assurance, cost, performance, and regulatory compliance. Many financial platforms implement a combination of models in which high-sensitivity tenants are isolated at the database level, while others are isolated at the row level with compensating controls (Kumar, 2020).

Table 1. Comparison of Tenant Isolation Models in Financial Multi-Tenant Platforms

Isolation Model	Isolation Strength	Operational Overhead	Common in Financial Platforms
Separate database	Highest	High	High-value tenants
Separate schema	High	Medium	Regional deployments
Row-level with mandatory filters	Medium	Low	Widespread
Service-level	High	High	Critical workloads

2.3 Traditional Access Control and Enforcement

While role-based access control (RBAC) and attribute-based access control (ABAC) are used extensively in conventional access control and enforcement systems, they still come with the burden of an accurate mapping of entitlements that is often a challenge in a changing financial environment (Oluoha et al., 2022). API gateways provide tenant isolation by recognizing the tenant via the authentication token, checking the policies according to a framework, like Open Policy Agent, validating the requests based on the tenant context, and applying rate limiting and anomaly detection at the API edge (Chode, 2025; Manukonda, 2023). Regulations require observability and auditability, which are provided by immutable logs that record access to tenant data, including who, when, and from what context. The logs, however, are predominantly used for post-mortem analysis and not as a real-time detection mechanism, leaving a gap for AI-driven controls to fill.

2.4 AI-Assisted Cybersecurity and Anomaly Detection

Evaluated recent developments in AI for cybersecurity include the use of supervised anomaly

detection with classification models trained on labeled datasets of attacks, unsupervised methods such as Isolation Forest, autoencoders, and clustering for emerging attack patterns, behavior analytics based on user and entity behavior analytics (UEBA) for creating normal behavior baselines, and sequence models including LSTMs and Transformers for analyzing API call sequences and access patterns (Harrou et al., 2024). The current research, however, addresses the detection through AI as the main tool rather than the use of AI as part of a larger architecture, and financial platforms have special needs to be explainable, to have low false-positive rates, and to have clear handoffs to human analysts, some of which are lacking in pure AI research (Desetty, 2024). This emphasizes the need for an architectural framework that puts AI in a supportive control position, not as a solution in its own right.

2.5 Regulatory Context

Regulatory expectations highly focus on continuous monitoring, access control verification, data protection by design, operational resilience, and incident reporting. When integrated into a governed enterprise architecture framework, AI-powered anomaly detection and policy enforcement can help meet these expectations. Financial platforms are under strict regulations, such as PCI-DSS v4.0, which demands cardholder data environment isolation and segmentation testing, GDPR, which requires data protection by design, breach notification, cross-border transfer controls, MAS TRM (Singapore) specifying technology risk management and access control verification, and DORA (EU) which demands digital operational resilience, incident reporting, and threat-led testing (Clausmeier, 2022). Table 2 presents the implications for tenant isolation of these regulations.

Table 2. Regulatory Requirements and Their Tenant Isolation Implications for Financial Platforms

Regulation	Tenant Isolation Implication
PCI-DSS v4.0	Cardholder data environment isolation, segmentation testing
GDPR	Data protection by design, breach notification, cross-border transfer controls
CCPA	Data inventory, access logging, and deletion verification
MAS TRM (Singapore)	Technology risk management, access control verification
DORA (EU)	Digital operational resilience, incident reporting, threat-led testing

3. Cross-Tenant Exposure Risk Model

3.1 Risk Vector 1: Missing Tenant Filter

Missing tenant filters represent the most common form of cross-tenant exposure, which occurs due to an absence of tenant identifier filtering on database queries and API access methods. Consequently, a tenant can view records from all tenants. For example, using a query for SELECT * FROM transactions WHERE account_id = ? without including AND tenant_id = current_tenant (). Traditional methods of detecting such issues include code reviews, ORM tenant-aware scoping, and automated testing, all of which are limited by human error and by gaps in test coverage (Alobaywi et al., 2026). While those methods rely on the human ability to detect abnormal patterns, AI-assisted methods can identify abnormal cross-tenant access to data by analyzing query logs. For example, AI-assisted methods will identify queries that return rows from more than one tenant ID, and/or those that have a result set count that deviates from the expected tenant-scoped count.

3.2 Risk Vector 2: Broken Object-Level Authorization

Broken object-level authorization happens when API endpoints accept identifiers like transaction IDs or account numbers without checking if they belong to the tenant. This allows attackers to cycle through identifiers across tenants, as seen with endpoints like GET /api/transactions/{transaction_id}, where an attacker can request transaction IDs from other tenants. The standard controls include adding middleware to add authorizations and ownership, but it is difficult to maintain because these are required for all endpoints (Kumar, 2020). When the implementation controls fail, AI-assisted techniques can identify ID access attempts in sequence or patterns by the same user and can also recognize API calls which do not have the correct resource ownership in the requester's context.

3.3 Risk Vector 3: Incorrect Enterprise/Entity Entitlement

When there is an incorrect entitlement in the enterprise or entity (e.g. tenant members are incorrectly assigned to resources), then you can get incorrect tenant

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

data, as a result of role inheritance issues, overlapping group names, misconfiguration of identity federation, etc. For example, a Tenant A user may inadvertently have access to Tenant B's reporting dashboard due to conflicts with naming groups and/or incorrect role assignments. Conventional approaches rely on entitlement review boards and manual access certifications, typically done quarterly or annually, while AI-powered approaches use the ability to recognize access anomalies by correlating users with their peers. This helps you to identify users with access to resources that other users in the same tenant don't have, which typically marks resource entitlement mapping errors and can be detected within days, not months, or even years.

3.4 Risk Vector 4: Shared Cache Leakage

Caching layers (such as Redis, Memcached or CDN services) cache the data for each tenant without tenant-aware cache keys. As a result, if two tenants make a request for user data using the same cache key (user_profile: {user_id} the cache can provide an answer for another tenant. Controlled environments are founded on policies for key generation for the cache, such as prefixes for namespaces and separate caches per tenant. However, such policies are not necessarily implemented and enforced. We use an AI analysis to monitor tenant mismatches according to the tenant's cache usage patterns and can provide an alert when the cached data is used by the tenant without the knowledge of that tenant.

3.5 Risk Vector 5: Exposed Identifiers in URLs and Logs

A subtle risk associated with publicly exposed IDs is that URLs, browser history or application logs maintain tenant IDs together as well as cross-tenant resource IDs, where unauthorized parties can gain access to this. An example of this would be where the tenant ID and account number are logged to error logs that support personnel from multiple tenants can access. While traditional security controls use log sanitization rules and design guidelines for creating URL formats that do not expose sensitive IDs, these methods are often not complete or circumvented. AI-supported detection provides the ability to view log streams for patterns of IDs in order to detect when logs have mixtures of IDs of Tenant A, which show access attempts by Tenant B (which may create targeted access to steal data across tenants).

3.6 Risk Vector 6: Misconfigured API Gateway Policy

A misconfigured API Gateway policy is an example of a failed control plane; routing and authentication/authorization policies may allow cross-tenant requests as a result of incorrect API gateway policy configuration, such as a gateway policy configured only to allow GET /tenant/data, but will permit any authenticated user to provide any value (Oluoha et al.). While traditional controls utilize policy-as-code validation, pre-deployment testing, and rule review, these controls only check a given moment in time and may miss certain configuration errors. In contrast, AI-enhanced improvements measure gateway telemetry performance to monitor policy compliance and detect successful cross-tenant requests that cannot be validated by expected tenant constraints, thereby enabling continuous monitoring of the policy's effectiveness throughout the API Gateway's production lifecycle.

3.7 Risk Vector 7: Background Job Processing Wrong Tenant Data

Background job processing tenant data in an incorrect manner happens when asynchronously processed jobs, such as batch processes, report generation or data export jobs, do not have the tenant context properly set, and therefore process data from multiple tenants through to the one tenant, creating the risk that the results will be routed incorrectly. One common error of this type is within a nightly statement generation job that processes all tenants but routes output files to the incorrect SFTP folder for each individual tenant that generated the report. Traditionally, controls to ensure tenant context is propagated, idempotency keyed, and staging validated are applied within the job environment. However, controls are not always consistently applied across the various job types. In contrast, implementing an AI-assisted improved monitoring of the input/output relationship between jobs allows for the identification of when the job results are routed to a different tenant than the one that generated the job input, thus reducing the risk of sensitive financial documents being routed to unintentional recipients.

3.8 Risk Vector 8: Reporting and Export Leakage

Reports and export leakage can occur as a result of a reporting engine's, data warehouse's or export function's inability to enforce tenant isolation in how it distributes data once they combine tenant information. A common example of this could be a BI dashboard which has a default filter of "all tenants" for users with administrator

rights, and subsequently, an administrator accidentally generates an export that contains consolidated reports (more than one tenant) of tenant information. Row-level security is implemented by most BI tools as a traditional means of controlling who has access to what when exporting information; however, once the report export has been approved by the appropriate user, it is ultimately up to that user(s) to follow the appropriate export process for that report. Therefore, should an administrator accidentally generate an export containing confidential information due to a misconfiguration, there may be no means to recover that exported information, resulting in either accidental or malicious leakage of tenant information.* Additionally, enhancements via artificial intelligence (AI) augmentation) Assist in preventing data exports that have a higher-than-expected volume of data exported from a BI system by flagging any export that has an unusually large volume and/or exports containing data from multiple tenants that do not match the expected historical trend of data from previous exports.

3.9 Risk Vector 9: Support and Admin Impersonation Risk

A risk of impersonating either support or admin arises when authorized personnel are given access to impersonate tenants, as well as bypassing isolation for the purpose of troubleshooting. The capabilities create new areas of insider threats and, therefore, are the result of administrator (support) identity verification. Such tools permit users to impersonate tenants through a "login as tenant" capability. A compromised account may be able to log in to a tenant and impersonate when using a tool with that feature. In general, traditional controls in place to help prevent impersonation would include logging, approval workflow, and time-based elevation of access rights. These methods are reactive and only investigated for a period of time (e.g., at the time of an audit). However, using assistance from artificial intelligence, it will be possible to identify abnormal impersonation behavior through the use of any of the three factors: timing, accessing tenants outside of the support/administrator's normal job duties, or taking any administrative action after impersonation. By identifying abnormal impersonation patterns, it should also be able to detect policy violations immediately and identify potential insider threats.

3.10 Risk Model Summary

The nine key risk vectors are identified in this section, such as missing tenant filters, broken object-level authorization, incorrect enterprise entitlement, shared cache leakage, exposed identifiers in URLs and logs, misconfigured API gateway policies, background job processing wrong tenant data, reporting and export leakage, and support admin impersonation, which mainly represent the primary failure points of multi-tenant isolation in financial platforms. Table 3 provides an overview of each vector, showing the traditional control gaps and the improvements that can be made using AI. This hierarchical taxonomy helps architects to threat model and assess threats. Besides, AI improvements for every vector act as the basis for the AI-assisted risk intelligence layer explained in Section 4 and discussed in Section 5.

4. Proposed Enterprise Architecture Framework

4.1 Architectural Principles

The proposed framework is founded on the principles of defense in depth, tenant-context integrity, least privilege, policy-based enforcement, auditability and AI-

Table 3. Summary of Cross-Tenant Exposure Risk Vectors with Traditional Control Gaps and AI-Assisted Enhancements

Vector	Traditional Control Gap	AI-Assisted Enhancement
Missing tenant filter	Relies on static testing	Real-time query pattern anomaly detection
Broken object-level auth	Per-endpoint implementation	Sequential ID access pattern detection
Incorrect entitlement	Periodic manual review	Peer-inconsistent access detection
Shared cache leakage	Cache key policy compliance	Tenant mismatch telemetry analysis
Exposed identifiers	Log sanitization rules	Identifier pattern scanning
Misconfigured API policy	Pre-deployment validation	Gateway telemetry deviation analysis
Background job leakage	Job context propagation	Input-output relationship monitoring
Report/export leakage	Export approvals	Volume and cross-tenant result detection
Support impersonation	Impersonation logging	Behavioral anomaly detection

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

assisted risk detection. AI is seen as an assistive capability to enhance detection and prioritization; deterministic controls are responsible for enforcement. The framework includes nine cross-tenant exposure vectors and an eight-layer control architecture. It is designed to facilitate governance via human oversight, compliance reporting, and secure SDLC integration. Additionally, the paper discusses implementation aspects for financial platform architectures, such as data privacy and adoption strategies. This comprehensive offering will provide enterprise security architecture and AI-driven operations, resulting in robust financial systems that enable multiple tenants.

4.2 Core Layers

The framework is divided into eight integrated control layers, each of which covers a particular facet of tenant isolation and security enforcement.

Layer 1: Tenant Context Layer registers and manages tenants' identities reliably across the platform, processes tenant identifiers and makes sure they are always presented in the same format across services, defines propagation protocols for context, such as through HTTP headers, gRPC metadata, or message queue headers and validates the context at service boundaries, with tenant contexts required in all authenticated requests, missing or malformed tenant contexts rejected, and cross-service context integrity verified.

Layer 2: Identity and Entitlement Layer authenticates principals and defines who is granted access to which tenants via an identity provider capable of multi-tenant awareness, RBAC with tenant-scoped roles, ABAC for dynamic conditions, and an entitlement cache with tenant partitioning, and includes controls, such as tenant membership assertion for every principal who is authenticated, role assignments that are scoped to a specific tenant, and regular entitlement attestation workflows.

Layer 3: API Policy Enforcement Layer enforces tenant isolation at the platform edge and between services via an API gateway with policy-as-code (Open Policy Agent or AWS Cedar), tenant-aware rate limiting, tenant validation of requests, and inspection of responses for whether a tenant has leaked into another tenant's query. Controls include a policy requiring a match in tenant ID in the authentication token to the tenant ID in the request target, automatic rejection of cross-tenant request patterns, and tenant ID sanitization of response headers.

Layer 4: Data Access Control Layer enforces tenant isolation on the database, cache and storage level using ORM-level tenant filtering with automatic query augmentation, database row-level security policies, and tenant-partitioned cache keys and storage namespaces; controls include mandatory tenant columns with NOT NULL constraints, views that enforce tenant filtering, and stored procedure validation.

Layer 5: The Secure Identifier Layer safeguards against identifier leaks and guarantees that identifiers reflect tenant context through opaque, non-sequential, tenant-specific identifiers. These identifiers are either cryptographically signed or internally mapped, preventing exposure of raw tenant details. Key controls involve signing identifiers to avoid tampering, verifying their binding, and automatically scanning logs for exposed identifiers.

Layer 6: Audit and Observability Layer captures logs that are structured and include tenant context, immutable storage (WORM or blockchain hash chains), integrity monitoring, and security information and event management (SIEM) integration, while offering controls such as logging all tenant activity, both authorized and denied, regular log integrity verification, and retention of logs to satisfy regulatory requirements.

Layer 7: The AI-Assisted Risk Intelligence Layer monitors access patterns for anomalies, tenant boundaries for violations, entitlement drift analysis, log and telemetry correlation, and has a risk scoring engine with controls such as real-time access pattern anomaly detection with configurable thresholds, security operations alerts, and integration with enforcement layers for automated response to anomalies.

Layer 8: Governance and Response Layer layers human oversight, compliance and continuous improvement with measures like an escalation path for confirmed violations, regular reviews of the effectiveness of the framework, and security integration requirements during the SDLC process.

4.3 Graphical Representation of Framework

Figure 1 shows a diagram of the framework and the flow of the requests through the eight layers. The requests are first validated, and the policy checks are performed at the Layer 3 API Policy Enforcement. Validation of the request and whether the requester is entitled to access it is the next layer, Identity and Entitlement. Layer 1 is Tenant Context, to make sure the identity of a tenant is kept consistent over the life of a request across an application; Layer 4 is Data Access Control, which gives isolation at the data layer. Layer 5: Secure Identifiers – Leverage a

resource identifier to carry tenant context around; Layer 6: Audit and Observability – Immutable log events and integrate with SIEM. Layer 7 is AI-Assisted Risk Intelligence, where the AI layer continuously analyzes telemetry from all the other layers to detect anomalies and generate alerts that are fed into Layer 8, which performs review and response and ensures compliance reporting with false-positive management. The result is a closed loop of detection informing response, and response data improving detection models.

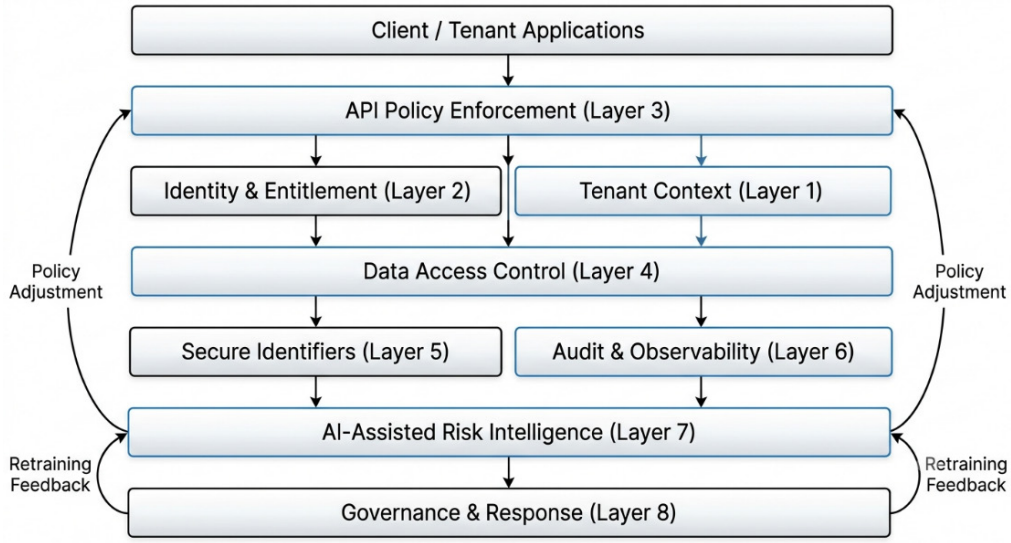

Figure 1. High-Level Architecture of the Proposed Eight-Layer Enterprise Framework for Cross-Tenant Data Exposure Prevention in Financial Multi-Tenant Platforms

4.4 Inter-Layer Integration

The layers work together in the context of key integration points. The Layer 3 API telemetry provides information to the Layer 7 AI models for real-time anomaly detection and risk scoring. These AI-generated outputs can trigger policy-approved, high-confidence, low-ambiguity blocking, temporary rate limiting, analyst reviews, or step-up authentication, allowing the system to be used for prompt and effective enforcement while also achieving human oversight. Layer 6 audit logs are used as training data and as an input to real-time detection for Layer 7. In the meantime, the Layer 8 governance determines the control requirements, evaluates the effectiveness of controls and provides false-positive and confirmed incident classifications for model retraining.

5. AI-Assisted Security Controls

5.1 Design Philosophy

The design principles for AI-assisted security controls are as follows: AI should be an assistant and complement deterministic controls, and not replace them; The rationale or why for the detection action should be human-readable and understandable to security analysts for any action with a significant impact; The false-positive burden should be low and the baseline should be tuned carefully, as financial operations simply cannot afford alert fatigue or they will simply get it wrong; The AI system should be continuously improved and updated through regular retraining on confirmed incidents and false positives, which creates a feedback loop that enables the AI system to become more accurate over time, while continuously reminding the human employee of the need to maintain high level of human oversight in regulated business environments.

5.2 AI-Assisted Detection Use Cases

The AI-assisted detection framework covers seven use cases, all of which cover the nine risk vectors identified in Section 3:

Use Case 1: Access Anomaly Detection uses input features such as the following to generate risk scores for each access: request timestamp, access frequency, resources accessed (tenants, object types, etc), API endpoints called, geographic context, network context, authentication method, and scores that, if they exceed a dynamic threshold, are flagged for review, rather than for blocking, thereby suggesting investigations.

Use Case 2: Tenant Boundary Violation Detection detects

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

tenant boundary violation attempts or accesses by seeing if the principal is a member of a specific tenant, the requested resource is a member of a specific tenant, and there is a history of access pattern violations, and it logs all tenant boundary violations while providing real-time comparison of principal tenant versus resource tenant, and generating alerts for possible violations to be reviewed by humans.

Use Case 3: API Response-Size Anomaly Detection detects API responses returning abnormally large amounts of data that suggest data exfiltration and missing pagination and filtering, based on input features such as response payload size by endpoint and tenant, user's historical response sizes, and peer average response sizes, flagging responses that are greater than expected by configurable thresholds like three standard deviations from baseline and potentially triggering automated pagination enforcement or even request blocking for extreme anomalies.

Use Case 4: Suspicious Data Export Detection monitors anomalies in reports, data extracts, and batch jobs from exports of data based on the volume of exports (rows, bytes), sensitivity classification of the tenant, and the user's export history, and provides risk scores for each export; gives human approval to deliver high-risk exports; logs all exports with risk metadata.

Use Case 5: Entitlement Drift Detection analyzes user entitlement to access consistency using input features like role assignments and membership changes, access patterns before and after entitlement changes, peer group access profiles in the same tenant, and time since last entitlement review, and periodically identifies users with high levels of drift between what they are entitled to do and what they can do, generating entitlement certification tasks for governance.

Use Case 6: Log and Audit Correlation detects complex attack sequences or policy violations across multiple services, time windows, or tenants based on input features such as: audit log sequences across services, temporal relationships between events, graph-based access patterns, alerts comprise an aggregated list of all the events, and the aggregated alerts can support forensic investigation such as timeline reconstruction, the events that were individually benign are considered as part of the attack when they collectively reveal that they are part of cross-tenant exposure attempts like reconnaissance followed by attempted boundary traversal.

Use Case 7: Risk Scoring and Alert Prioritization uses input from all the detection use cases and features, which includes outputs from Use Cases 1-6, tenant criticality classification, regulatory context (PCI data vs. non-sensitive data) and current threat intelligence, to aggregate signals into normalized risk scores and prioritize alerts by severity, confidence, and impact, preventing alert fatigue by hiding low-confidence or low-impact alerts, while allowing high-confidence and high-severity alerts to initiate automated policy-approved actions or prompt human review based on the risk level and confidence.

5.3 Model Selection and Positioning

The contribution to AI/architecture control is not the AUC-ROC of the model, but understanding how AI complements the architecture controls. When temporal dependencies exist, sequence models can be used for API access patterns. The framework does not require a particular algorithm, such as LSTM or Transformer, and the evaluation of the models is part of the implementation phase, instead of the definition phase. New architectures are not always developed, but architectures are chosen based on the characteristics of the data, the response time and the requirement to understand the security decision; and explainability is more important than small gains in accuracy, as there are regulatory requirements to understand security decisions.

5.4 Data Privacy and Model Training

In financial multi-tenant environments, there are privacy considerations like training data without PII where possible, feature extraction without sharing tenant data, and even future options such as federated learning, a distributed approach to model improvements without data sharing that can be addressed, along with model governance, to ensure models continue to detect threats over time without losing accountability and transparency, in tandem with the need for data retention and consistency, training frequency, and the ability to monitor model performance.

6. Scenario-Based Evaluation

6.1 Evaluation Approach

Instead of reporting the experimental results on synthetic datasets or comparing architectures, this section examines realistic architecture scenarios where traditional controls are used without AI assistance and then with AI-assisted controls, and for each scenario describes the exposure vector from section 3, how effective the traditional controls are, how AI-assisted controls improve the situation, and the comparison of the outcome with the traditional control scenario.

6.2 Scenario 1: Missing Tenant Filter in Payment Query

In this scenario, omitting a tenant filter in a payment query led to an error with a third-party payment processing service. The payment processing service added a new query endpoint to get the status of a payment (GET /payments/status/{reference}), but an attacker with credentials associated with Tenant A could have iterated through the reference numbers and issued a request for payment information from Tenant B, due to the omission of the tenant filter. Using only traditional controls, the code review during the usual process did not detect that an essential part of the query's join condition was missing, so the flaw would have gone undetected until the payment data was released to production and became available to the attacker. This likely occurred a couple of weeks after the code was reviewed, and no other review process was in place to catch the flaw in production. The flaw would only have been detected when a Tenant B user contacted the payment processing service about the breach. With the assistance of AI, Layer 7 was able to detect the query returning rows with multiple tenant IDs in its logs and alerted users of these results within minutes of the first cross-tenant transaction; subsequently, after a review and confirmation of the violation by security operations on Layer 8, temporary blocking was enforced by the API gateway on Layer 3 until the flaw was fixed. In this case, the exposure window might be shortened from weeks to just hours.

6.3 Scenario 2: Excessive Data Export by Compromised Account

The scenario looks at the issue of excessive data export as a consequence of a compromised account. In this case, the account of a financial analyst was compromised, and an attacker had legitimate access to the API, allowing them to export all transaction records from the analyst's tenant. This amounts to 10,000 rows exported, compared with typical exports of approximately 50 rows. Based upon traditional access controls, the use of role-based access control (RBAC) allows for exporting of transaction records since the analyst role provides for that permission. There are also no rate or volume limits associated with this export, and it was completed without issues. There was no detection that data exfiltration had occurred until the periodic access review occurred several months later. The use of artificial intelligence (AI) with Layer 7 response size detection would have detected the export based on the volume being 200 times the normal export volume; the risk score for the export would have calculated the volume anomaly together with the unusual time of 2 AM and geographical mismatch based upon the export originating from an unexpected country; and the combined high-risk score would have triggered a human review at Layer 8. As a result, the export would have been paused, and the analyst would have been contacted to verify the export prior to any detection of a compromise, and credential rotation would have taken place to prevent exfiltration instead of detection after the fact.

6.4 Scenario 3: Wrong Entitlement Mapping During Onboarding

In this scenario, we examine the effects of incorrect entitlement mapping within an onboarding process, which results in a user being wrongfully assigned to the entitlement group "Finance_Readers_TenantB," rather than the correct group, "Finance_Readers_TenantA." As a result of this erroneous assignment, the user was able to access Tenant B's financial reporting for a period of one week before it was discovered. Without traditional control measures in place, manual attestation would occur every three months, allowing for the only detected inaccuracy to occur after three months or not at all. Due to the introduction of AI technology for entitlement drift detection on the Layer 7, this process allows for a user's access pattern to be compared against that of their peers within Tenant A, indicating that the user has accessed resource(s) within Tenant B that no other member of Tenant A had ever accessed, generating an alert to security operations (SO) within the first 24 hours of registering access to a resource(s) in Tenant B. Upon investigation, the security team was able to detect that the user was assigned the wrong group and correct the entitlement within 24 hours of identification of the discrepancy. In this case, the duration of unauthorized access might be shortened from several months to around 2 days.

6.5 Scenario 4: Shared Cache Leakage

In this scenario, a shared cache leak occurs; in this case, where Redis is configured to use a key pattern user:{userId}:profile without considering a tenant prefix, a user in Tenant A (user ID 123) has their profile data returned when another user in Tenant B requests the profile of user ID 123. Because traditional controls were used, the cache design review failed to identify the problem. Six months after penetration testing began, an issue was identified. However, exposing the tenant's profile was done for the

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

entire duration. With AI assistance, Layer 7 tracks cache response patterns, identifies tenant mismatches where the cache key user:123:profile returns data with a different tenant ID than the requester, issues real-time alerts, and updates the cache key policy within hours. In this case, the exposure time might decrease from months to just hours.

6.6 Scenario 5: Support Admin Impersonation Abuse

In this case, impersonation abuse of support admin is mitigated, with a support engineer from a platform using the "impersonate tenant" functionality to access the CEO's tenant account for curiosity's sake, which is a high-value tenant. While it is against the policy, it is not malicious. Impersonation is only reviewed quarterly, relying only on traditional controls, without pattern detection. Violations are usually discovered on random audits, or they are not known. Supported by AI, Layer 7 can detect normal impersonation patterns such as who the support staff impersonate, how long they do so for, and when they do it. It identifies when impersonation is by staff members who have not accessed the tenant before, at times outside of business hours and where there are no active support tickets. Anomalies are flagged for review, and the security team is asked to review the anomalies. A policy reminder is then issued, facilitating detection and resolution of violations before escalation.

6.7 Comparative Summary

Table 4 shows a comparative overview of the five scenarios, with the traditional controls requiring weeks to months to detect and react to problems, including the fact that the tenant filters were not installed, excessive data export was detected after it had occurred, entitlement mappings were verified once a quarter, the next penetration test to discover leaking cache information, and quarterly log reviews to identify support impersonation. The scenario-based evaluation indicates that controls supported by AI can significantly shorten detection and response times from weeks or months to just hours or days in typical exposure scenarios.

Table 4. Comparative Analysis of Traditional versus AI-Assisted Controls Across Five Cross-Tenant Exposure Scenarios

Scenario	Traditional Only	AI-Assisted	Key Improvement
Missing tenant filter	Weeks–months detection	Hours detection	Real-time log analysis
Excessive data export	Post-exfiltration detection	Pre-exfiltration prevention	Volume + context anomaly
Wrong entitlement mapping	Quarterly attestation	24-hour detection	Peer inconsistency detection
Shared cache leakage	Next penetration test	Hours detection	Telemetry correlation
Support impersonation	Quarterly log review	Real-time anomaly flag	Behavioral baseline

7. Governance Model

7.1 Human-in-the-Loop Review

Governance establishes accountability, compliance and continuous improvement for effective AI-assisted security and the human-in-the-loop principle is applied, with humans making decisions on how to handle high-impact alerts, automated responses limited to low-risk anomaly flagging, alert generation, logging and rate limiting for extreme anomalies, human review required for account suspension, blocking data export, revoking entitlements and terminating impersonation, and escalation paths defined for each type of alert, including service level agreements to review high-risk cross-tenant alerts after 15 minutes; roles are clearly detailed, with a security analyst being responsible for triage, investigation and true or false-positive classifications, a security engineer responsible for responding to confirmed incidents and implementing containment and a compliance officer reviewing audit evidence and certifying controls.

7.2 Security Operations Workflow

The security operations workflow is organized and alert generation is done at Layer 7. A security analyst processes these alerts and filters them by false positive or real incident alerts. There are different responses to the different risk levels of alerts: low-risk alerts are only logged, medium-risk alerts are reviewed and remediated by an analyst, and high-risk alerts require immediate response, containment and formal incident response. This strategy helps prioritize alerts and response, matching response to impact. In addition, the workflow also keeps audit trails of all decisions and actions made.

7.3 False Positive Handling

In financial platforms, handling false positives is important because alert fatigue can result in critical alerts being ignored if the rate of false positives is too high. The framework is designed to dynamically set the thresholds per tenant, based on the risk profile as well as the history of false positives. It contains a feedback loop that can be used for retraining the model based on analysts' true or false classification, and it provides the detail that helps the analyst make a decision, such as contributing features, confidence score, and similar past cases, to make it explainable. Also, it permits for documented exemptions, such as valid unusual designs, such as quarterly reporting exports. The framework includes target metrics, including less than 1% false positives for automated blocks and less than 5% for alerts that require investigations, to provide clear performance standards.

7.4 Model Explainability

Security analysts should be aware of the reason for an anomaly to decide whether to take any action. The framework provides several options for model explainability, such as features that are important or have values that are near or above thresholds, using measures derived from the SHAP or LIME values. It also offers counter-factual explanations such as: "This access would be allowed normally during business hours or if there were fewer than 100 rows accessed. It also mentions comparable incidents, such as "flagged because of similarity to 3 previous data exfiltration incidents," and keeps audit logs for regulators to review, and for continuous system improvements.

7.5 Audit Evidence and Compliance Reporting

Having audit evidence and compliance reports is essential to verify that all AI-driven decisions affecting tenant isolation are connected to specific model versions, training data sources, and inference times. The results of human reviews should be recorded, and the reasons for them. Automated reports are mandatory weekly, which include a summary of detections, false positive rate and alert types; monthly reports are required, including information on entitlement drift and unresolved issues; and quarterly reports are necessary, including attestations of control effectiveness and model performance evaluations. These reports support gathering evidence for compliance alignment, such as PCI-DSS requirement 10 – Logging, GDPR Article 32 – Security, and MAS TRM 7.2 – Access Control Verification. A summary of reporting schedules and other regulatory mappings is shown in Table 5.

Table 5. Compliance Reporting Schedule with Frequency, Content, and Regulatory Mappings

Frequency	Report Content	Regulatory Mapping
Weekly	Detection summary, false positive rate, alert volume by type	PCI-DSS 10 (logging)
Monthly	Entitlement drift findings, outstanding exceptions	GDPR Article 32 (security)
Quarterly	Control effectiveness attestation, model performance review	MAS TRM 7.2 (access control)

7.6 Architecture Review Checklist

The architecture review checklist calls out mandatory reviews prior to deployment, including verifying tenant context propagation for each service, adding tenant filtering or row-level security to all database queries, documenting tenant-context matching in all API gateway policies, documenting where tenant ID is captured in audit logs for all access events, documenting AI coverage for all nine risk vectors, defining false positive handling and escalation paths, and testing the human-in-the-loop review workflow to ensure that all layers of the framework are implemented correctly before new services or features are deployed, and that security controls are not inadvertently weakened during the evaluation of the system.

7.7 Secure SDLC Integration

Secure SDLC integration comprises threat modelling during the design phase, which covers threat modelling of cross-tenant exposure vectors and completion of the architecture review checklist, tenant-aware ORM patterns and tenant filtering unit tests during development phase, automated tests using a multi-tenant scenario in testing phase, canary analysis during deployment phase, whereby the volume of anomalies flagged by AI tools is monitored for any increase, which could a sign of misconfiguration, and regular red-team testing throughout the operations phase to test the boundaries of cross-tenant exposures, ensuring that security is built into the platform from the ground up, as opposed to bolted on after deployment, and that the framework continues to be effective as the platform evolves.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

8. Discussion

8.1 Interpretation of Results

The scenario-based evaluation indicates that AI-assisted controls can significantly shorten detection and response times from weeks or months to just hours or days in typical exposure scenarios. Pre-exfiltration prevention has been driven by excessive data exfiltration problems, moving security from reactive to proactive. Also, from quarterly attestation cycles (up to 90 days), the entitlement drift detection time has been decreased to 24 hours. Impersonation abuse monitoring was also enhanced from a quarterly log review to real-time anomaly detection. These advances are achieved through the combination of AI-assisted detection with deterministic architectural controls to achieve overall telemetry correlation and baseline behavior analysis. With Human-in-the-loop review and explanations, a governance layer holds operational accountability in place as detection improves. These findings suggest that AI should be used as an assistant for tenants' isolation controls, and not replace those controls, as an architectural aid.

8.2 Practical Benefits

It offers a full risk model with 9 vectors of exposure to help uncover and minimize blind spots when conducting threat modeling and architecture assessments for financial platform architects. It also includes an eight-layer framework that includes a robust control inventory that can help prioritize implementation activities. Security operations teams benefit from ease of proactive detection because they are no longer discovering breaches reactively and from a decrease in alert fatigue, as anomaly suppression and prioritization are streamlined. The platform provides audit trails for compliance officers, enabling them to document and access audit evidence that supports compliance reviews. Continuous monitoring allows continuous control monitoring and provides evidence to support internal audit and regulatory review (Eboseremen et al., 2022).

8.3 Limitations

There are a number of constraints as well that will need to be addressed, such as architectural constraints (assuming that the platform can propagate tenant context consistently across all services, which is a luxury that legacy systems can't always afford to provide and necessitates a high degree of refactoring), high-velocity transaction systems, where real-time AI inference latency raises unacceptable performance overhead , multi-cloud deployments, where the ability to implement the same layer consistently across different infrastructure environments is complex, technical constraints, which includes AI-specific data privacy challenges where access to tenant access patterns requires for training, is sometimes prohibited by the confidentiality obligations of those financial institutions, explainability vs accuracy trade-offs where the most accurate deep learning models are also the least explainable (Ribeiro et al., 2016), adversarial robustness issues where an attacker can learn to evade anomaly detection through pattern drift over time (Goodfellow et al., 2014), which the platform can't detect because this is outside of the training data and introduces operational overhead, and concept drift where normal access patterns continuously evolve over time necessitating ongoing retraining to keep up with these changes and adding to the operational overhead).

8.4 Implementation Complexity

While the framework is accessible to organizations that have mature security programs, it will take a substantial investment for organizations with less mature programs, as it involves the following prerequisites: standardized identification of tenants across all services; structured audit logs with a consistent schema; access to historical access data for model training; and a security operations team with data science support.

8.5 Data Privacy Concerns

Data privacy concerns must be considered, since training data considerations indicate that access patterns to current data on a tenant may contain commercially sensitive information, training may be done in memory, and some financial tenants, such as banks with bank secrecy laws, cannot share their data for model training, while pattern analysis happens in memory and hence is not stored beyond the retention period set by a defined policy; inference privacy is more complicated, because it requires detecting anomalies in real time as they occur while using it, but the analysis is done in memory and not stored, therefore, mitigation involves defining data retention policies which set the period for which data will be stored, encrypting any data stored as telemetry, and regularly assessing the impact of the data on security and privacy standards to ensure adherence to evolving data protection laws.

8.6 Adoption Roadmap

The framework's adoption is divided into four phases over 12 months. Phase 1 (months 0-3) focuses on foundational controls like implementing tenant context propagation across all services, deploying an API gateway with tenant-aware policies, and standardizing audit logs across the platform. Phase 2 (months 3-6) involves initial deployment of AI assistance, such as access anomaly detection from Use Case 1 integrated into security operations, and setting up a false-positive feedback loop. Phase 3 (months 6-9) broadens detection capabilities to include tenant boundary violations, entitlement drift, and response-size anomalies, covering more risk vectors. Phase 4 (months nine to twelve) is about maturity, including full 9-vector coverage, automated response for high-confidence detections with appropriate human oversight, and regular red-team exercises that test the framework against realistic attack scenarios (Ren et al., 2025), while providing a clear progression from foundational controls to mature AI-assisted security operations (Hundeyin, 2025).

8.7 Comparison with State-of-the-Art

While pure AI research for intrusion detection is based on LSTM or Transformer models for intrusion detection, AI is treated as the main component of the system, and attention is paid to the architecture and accuracy of the model, the proposed framework treats AI as an assistant rather than the main part of the system, and focuses on the cross-tenant detection risk identification model with integrated AI assistance, while state-of-the-art approaches opt for cloud provider native controls (AWS IAM, Azure RBAC) that provide deterministic isolation with limited or no AI component, SIEM with UEBA puts emphasis on log analysis and alerting with anomaly detection but lacks integrated architecture and governance layers (Desetty, 2024), and zero-trust architectures emphasize the verification of every request, with often no AI component and this framework includes an integrated cross-tenant detection risk identification model, eight-layer architecture controls, the integration of AI assistance, and a regulatory governance model tailored for regulated financial environments, (Mushtaq et al., 2025). Few existing frameworks explicitly combine all these elements, such as comprehensive risk identification, multi-layered architecture controls, AI assistance, and regulatory governance into a single, practitioner-focused enterprise architecture model designed for regulated financial platforms.

9. Conclusion and Future Work

This paper has explored the cross-tenant data exposure risk and introduced an enterprise architecture framework that combines the principles of tenant isolation, identity management, API enforcement, secure identifiers, auditability, and AI-driven risk detection. The paper features a nine-vector risk taxonomy to offer a thorough understanding and approach to threat assessment; an eight-layer control architecture for defense in depth; an "assistant" approach to using AI for anomaly detection and telemetry correlation, not as an end-to-end security solution; a governance model that incorporates human review, explainability, and compliance reporting; and implementation and adoption guidance.

In the near term (3-6 months), prioritize auditing the current state by mapping existing controls to each of the nine risk vectors. Standardize the propagation of tenant context across all services. Implement structured audit logs with immutable storage. Begin AI-assisted anomaly detection, focusing initially on access anomalies and response size, with low-risk alerts. Long-term investments (6-12 months) should include deploying all nine vectors with risk scoring and prioritization, establishing a governance program with regular model reviews, false-positive analysis, compliance reporting, and launching AI-assisted anomaly detection on access anomalies and response size, with low-risk alerts. Additionally, a red-team exercise involving scenario cross-tenancy testing should be conducted to test the framework's effectiveness with real attack patterns over the next year, ensuring it evolves with the platform and combats new threats.

Additionally, the proposed framework will find various promising avenues to explore, such as federated learning for privacy-preserving model improvement without sharing tenant data; privacy-preserving telemetry using differential privacy to ensure that tenant access patterns cannot be reconstructed; automated policy validation using formal methods to ensure collective enforcement of tenant isolation; real-world deployment studies for empirical evidence on framework effectiveness; and adversarial robustness studies to ensure that detection models are effective against adaptive attackers.

Declarations

Ethics approval and consent to participate

This study did not involve any human participants, animal subjects, or sensitive personal data. Therefore, no ethical approval or informed consent was required for this research.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

Consent for publication

The author confirms that this work is original and has not been published elsewhere. No individual person's data, images, or other identifiable information are presented in this manuscript. Hence, consent for publication is not applicable.

Conflicts of Interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions

Kannan Meiappan (Corresponding Author): Conceptualization, methodology, formal analysis, investigation, writing—original draft preparation, writing—review and editing, visualization, project administration.

Acknowledgements

The author would like to thank the reviewers for their constructive feedback, which significantly improved the quality and clarity of this manuscript.

References

Alobaywi, B., Almutairi, M. G., & Sheldon, F. T. (2026). Performance Trade-Offs in Multi-Tenant IoT–Cloud Security: A Systematic Review of Emerging Technologies. IoT, 7(1), 21.

Bala, V. (2025). Secure Multi-Tenant FinTech Architecture: Real-Time AI-Powered Fraud Detection Pipeline with Encrypted Data Streams. Journal of Computer Science and Technology Studies. https://doi.org/10.32996/jcsts.2025.7.10.24

Chode, B. (2025). AI-Driven Risk-Adaptive Authorization for Multi-Tenant Cloud APIs - Microsoft Azure. International Journal of Computer Science and Data Engineering. https://doi.org/10.55124/csdb.v2i3.254

Clausmeier, D. (2022). Regulation of the European Parliament and the Council on digital operational resilience for the financial sector (DORA). International Cybersecurity Law Review, 4, 79-90. https://doi.org/10.1365/s43439-022-00076-5

Desetty, A. G. (2024). Unveiling Hidden Threats with ML-Powered User and Entity Behavior Analytics (UEBA). Turkish Journal of Computer and Mathematics Education (TURCOMAT). https://doi.org/10.61841/turcomat.v15i1.14394

Eboseremen, B. O., Ogedengbe, A. O., Obuse, E., Oladimeji, O., Ajayi, J. O., Akindemowo, A. O., Ayodeji, D. C., & Erigha, E. D. (2022). Secure Data Integration in Multi-Tenant Cloud Environments: Architecture for Financial Services Providers. Journal of Frontiers in Multidisciplinary Research. https://doi.org/10.54660/.jfmr.2022.3.1.579-592

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Courville, A. (2014). Generative Adversarial Networks. Advances in Neural Information Processing Systems, 3. https://doi.org/10.1145/3422622

Harrou, F., Bouyeddou, B., Dairi, A., & Sun, Y. (2024). Exploiting Autoencoder-Based Anomaly Detection to Enhance Cybersecurity in Power Grids. Future Internet, 16, 184. https://doi.org/10.3390/fi16060184

Hashim, W., & Hussein, N. A.-H. (2024). Securing Cloud Computing Environments: An Analysis of Multi-Tenancy Vulnerabilities and Countermeasures. SHIFRA. https://doi.org/10.70470/shifra/2024/002

Hundeyin, W. (2025). Modernizing IT Audit Function to Support to Support Zero Trust and Cloud Security – A Systematic Review. International Journal of Innovative Science and Research Technology. https://doi.org/10.38124/ijisrt/25oct056

Kumar, R. K. R. (2020). Multi-Tenant SaaS Architectures: Design Principles and Security Considerations. Journal of Software Engineering and Simulation. https://doi.org/10.35629/3795-06052841

Manukonda, A. K. (2023). Secure Multi-Tenant Application Deployments in Kubernetes. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences. https://doi.org/10.37082/ijirmps.v11.i4.232600

Mushtaq, S., Mohsin, M., & Mushtaq, M. (2025). A Systematic Literature Review on the Implementation and Challenges of Zero Trust Architecture Across Domains. Sensors, 25, 6118. https://doi.org/10.3390/s25196118

Narlawar, S. (2025). Enterprise Data Integration: A Case Study Analysis of Multi-Tenant Platforms Enabling Cross-Domain Analytics. Journal of Computer Science and Technology Studies. https://doi.org/10.32996/jcsts.2025.7.4.122

Oluoha, O., Odeshina, A., Reis, O., Okpeke, F., Attipoe, V., & Orieno, O. H. (2022). A Unified Framework for Risk-Based Access Control and Identity Management in Compliance-Critical Environments. Journal of Frontiers in Multidisciplinary Research. https://doi.org/10.54660/.ijfmr.2022.3.1.23-34

Pippal, S., Kumar, S., & Rani, R. (2024). Optimizing multi-tenant database architecture for efficient software as a service delivery. TELKOMNIKA (Telecommunication Computing Electronics and Control). https://doi.org/10.12928/telkomnika.v22i5.26385

Reddy, R., & Ratnam, K. V. (2021). Multi-tenant data isolation techniques in public clouds assessing the effectiveness of isolation mechanisms. World Journal of Advanced Research and Reviews. https://doi.org/10.30574/wjarr.2021.12.1.0402

Ren, Y., Wang, Z., Sharma, P. K., Alqahtani, F., Tolba, A., & Wang, J. (2025). Zero Trust Networks: Evolution and Application from Concept to Practice. Computers, Materials and Continua, 82(2), 1593-1613. https://doi.org/ https://doi.org/10.32604/cmc.2025.059170

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. https://doi.org/10.1145/2939672.2939778

Sharma, R. K. (2025). Multi-Tenant Architectures in Modern Cloud Computing: A Technical Deep Dive. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. https://doi.org/10.32628/cseit25111236

Singh, P. (2019). Design Patterns for Secure Multi-Tenant Architecture in Financial Services. International Journal on Science and Technology. https://doi.org/10.71097/ijsat.v10.i4.5507

Teegala, R. T. R. (2017). Architectural Design and Governance of Multi-Tenant Microservices for Regulated Financial Institutions. International Journal of Scientific Research in Science Engineering and Technology. https://doi.org/10.32628/ijsrset1735152