A Structured Review of SaaS Security Architecture: Challenges, Models, and Research Gaps

Dr. Sonal Sagar Boda

University of the Cumberlands
sboda59203@ucumberlands.edu

Abstract

The growing adoption of Software as a Service (SaaS) has introduced significant architectural security challenges that impact both technical implementation and organizational management. This paper presents a structured review of over 60 peer-reviewed academic sources, emphasizing publications from 2019 to 2024 while incorporating foundational studies essential to the development of secure and scalable SaaS platforms. The review focused on identifying and synthesizing security concerns associated with SaaS environments. The review applied a structured keyword-based search across multiple academic databases and selected literature based on relevance to SaaS security, architecture, and information security management. Key topics identified include customization, scalability, multitenancy, integration complexity, encryption, and third-party risks. To frame the analysis, the review draws on theoretical models including Model-Driven Architecture (MDA), Service-Oriented Architecture (SOA), and the Information Security Management (ISM) framework. These models support an examination of how confidentiality, integrity, availability, access control, and risk management are conceptualized in SaaS environments. The literature identified concerns related to cross-domain interoperability, limitations in controlling hosted platforms, evolving data privacy standards, and the organizational need for adaptable security practices to support SaaS environments. The findings indicate that, although SaaS provides recognized benefits in scalability and operational efficiency, the existing literature presents a fragmented and uneven treatment of its architectural security challenges. This review contributes a consolidated theoretical perspective on SaaS security architecture and highlights gaps in integrating organizational governance with architectural decision-making. The study concludes by recommending future research focused on empirically validating security frameworks, developing standardized assessment criteria, and exploring interdisciplinary approaches to secure SaaS system design.

Keywords: SaaS security architecture, cloud computing, Information Security Management (ISM), multitenancy, integration security, customization, scalability, data privacy, Model-Driven Architecture (MDA), Service-Oriented Architecture (SOA)

I. INTRODUCTION

The increasing adoption of Software as a Service (SaaS) has transformed how organizations deliver and consume software, offering benefits such as cost-efficiency, scalability, and ease of integration [1], [2]. SaaS platforms operate on a subscription-based model, providing flexible cloud-native infrastructure and operational efficiency across industries [2], [3]. This delivery model has become integral to digital transformation initiatives, especially as enterprises shift from traditional on-premises systems to scalable,

user-centered platforms that support cross-domain collaborations and automation [1], [4], [12]. However, this transition has also introduced a series of architectural and operational risks [1], [5].

The architectural design of SaaS platforms presents a range of challenges that include customization complexity, limited control over hosted services, third-party integration risks, and vulnerabilities introduced through multi-tenancy [2], [5], [6]. Security concerns in particular remain at the forefront, encompassing risks to data privacy, user authentication, encryption standards, and regulatory compliance [1], [5]. Díaz de León Guillén et al. [6] identified key SaaS threats such as insecure APIs, insider attacks, identity theft, and compliance violations, highlighting the pressing need for robust, theory-driven frameworks to manage these threats. Scholars have emphasized that despite the integration of built-in security features by cloud providers, client organizations face difficulties maintaining secure configurations during platform scaling and integration [2], [6], [7].

Multiple frameworks and models have been proposed to address SaaS security challenges. The models include Service-Oriented Architecture (SOA), which emphasizes service-level abstraction, and Model-Driven Architecture (MDA), which enhances system flexibility [8], [9]. More recently, the Information Security Management (ISM) framework has gained traction as a comprehensive security governance model, covering confidentiality, integrity, availability, access control, and risk management [10]. While these frameworks contribute essential theoretical underpinnings, prior research remains fragmented in integrating them systematically to assess SaaS architecture security [1], [7]. Some studies evaluated individual components (e.g., access control or customization) without addressing the full architectural ecosystem [8], [9]. As such, only few literature studies have identified security-centric architectural synthesis aligned with current industry demands and scholarly expectations [6], [10].

To address this gap, the study conducted a systematic literature review of academic sources published between 2019 and 2024, focusing on SaaS architecture security challenges. Guided by the ISM framework, the review analyzes how peer-reviewed research conceptualizes and responds to security constructs within SaaS design and implementation. Sources were identified using the University of the Cumberlands Library system and Google Scholar, based on selection criteria that emphasized peer-reviewed journals and conference proceedings relevant to information systems, cybersecurity, and cloud architecture [1], [2], [7].

This study contributes to the literature in three specific ways. First, it consolidates fragmented discussions of SaaS security architecture into a coherent synthesis informed by ISM theory [6], [10]. Second, it maps the alignment and disconnects between theoretical frameworks and practical applications across selected studies [7]–[9]. Third, it identifies gaps in the current literature and suggests future research opportunities, such as exploring cross-domain privacy mechanisms and adaptive access control strategies [6], [9], [10].

The remainder of the article is structured as follows: Section II outlines the review methodology, including database selection, inclusion and exclusion criteria, and analytical procedures. Section III discusses the theoretical background of SaaS and the ISM framework. Section IV presents the results of the literature review. Section V synthesizes the findings using the ISM framework. Section VI discusses implications for research and practice, and Section VII concludes the study.

II. METHODOLOGY

This study employed a structured literature review approach to examine SaaS security architecture concerns through the lens of the Information Security Management (ISM) framework. The design of this review, including its conceptual framing and inclusion criteria, builds on prior work conducted in the author's doctoral dissertation [11], which provided a structured lens to examine how confidentiality, integrity, availability, access control, and risk management are represented in the literature on SaaS architecture.

A. Data Sources

Relevant academic literature was retrieved from the University of the Cumberlands Library database system and Google Scholar. These platforms provided access to journal articles, conference proceedings, and doctoral dissertations within the domains of information systems, cybersecurity, cloud computing, and software architecture. Grey literature, vendor whitepapers, and non-peer-reviewed content were excluded.

B. Search Strategy

The search strategy was developed using Boolean operators (AND, OR) to combine targeted keywords such as “SaaS security architecture,” “multi-tenancy risk,” “API integration,” “cloud system customization,” “identity and access management,” and “information security governance.” The query was restricted to English-language publications from 2019 to 2024. The initial search yielded approximately 250 documents. Titles and abstracts were screened for alignment with architectural security concerns. Duplicates and inaccessible records were removed prior to full-text screening.

C. Inclusion and Exclusion Criteria

Studies were included if they met the following criteria:

1. Peer-reviewed journal articles, conference papers, and doctoral dissertations.
2. Published primarily between 2019 and 2024, with the inclusion of foundational works where relevant.
3. Focused on SaaS-related architectural issues such as customization, integration, multitenancy, scalability, or access control.
4. Provided theoretical, technical, or governance-oriented insight into SaaS security.

Studies were excluded if they:

1. Focused solely on Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS).
2. Addressed only user-level behavior or adoption patterns without architectural relevance.
3. Lacked discussion of security-related constructs or theoretical frameworks.
4. Were inaccessible in full text or published in a non-academic outlet.

D. Review and Synthesis Approach

The review followed a structured reading protocol informed by best practices in integrative literature synthesis [1], [2]. Each eligible study was analyzed for its alignment with ISM framework dimensions. A matrix-based synthesis approach was applied to extract and organize architectural risks, security design considerations, and conceptual contributions across the selected studies. Although no software-assisted coding tool was used, the review maintained internal consistency through theory-based classification and analytical rigor.

The synthesis aimed to consolidate theoretical perspectives on SaaS security architecture rather than assess empirical results. Special attention was given to how architectural design interacts with security governance, system integration, and regulatory expectations across cloud deployments.

E. Limitations

This review is limited to English-language publications indexed through the University of the Cumberlands Library and Google Scholar. The exclusion of grey literature and vendor-authored technical reports may omit some applied insights. While the ISM framework structured the analytical process, the review did not apply formal inter-rater reliability checks or a critical appraisal tool, as the focus was on conceptual synthesis rather than quality-weighted scoring. This design choice reflects the study’s aim to

integrate architectural security concerns through a theoretical lens rather than to evaluate individual study validity.

III. Literature Review

The evolution of Software as a Service (SaaS) is situated within broader advancements in cloud computing. As enterprises transition from traditional on-premises systems to distributed, cloud-native infrastructures, SaaS has emerged as a dominant delivery model characterized by subscription-based access, rapid deployment, and elastic scalability [1], [3]. The SaaS delivery model offers cost and operational benefits but introduces architectural and security challenges not present in conventional environments [1], [6].

Security within SaaS ecosystems remains a focal point of scholarly concern, especially as services increasingly rely on third-party integrations, identity management protocols, and cross-border data flows. Numerous studies have examined security implications of various SaaS architectural features, yet the discourse remains fragmented across isolated architectural concerns such as customization, multitenancy, API security, and regulatory compliance [1], [6], [8]. The need for an integrative framework to analyze these concerns holistically has led to increased interest in theory-grounded evaluations of SaaS security architecture.

The Information Security Management (ISM) framework provides a comprehensive lens for evaluating organizational and technical controls across five core dimensions: confidentiality, integrity, availability, access control, and risk management [10]. Recent literature has applied this framework to diverse security contexts; however, its systematic application to SaaS architecture remains limited. This review employs the ISM framework to synthesize architectural concerns and assess the depth and distribution of security emphasis across the existing body of knowledge.

This section presents a structured review of peer-reviewed academic literature published between 2019 and 2024. It begins by contextualizing SaaS within the broader field of cloud computing and examining its architectural foundations. A dedicated subsection outlines the ISM framework and its relevance to SaaS security. Subsequent subsections organize the literature around five major architectural concerns—customization and scalability, multitenancy, integration and API security, identity and access management, and regulatory compliance. Each is analyzed for its theoretical framing, reported risks, and alignment with ISM security dimensions. The section concludes with a synthesis of findings and identification of literature gaps that inform subsequent implications for research and practice.

A. *Cloud Computing and SaaS Architecture Overview*

Cloud computing is characterized by its ability to deliver shared computing resources over the internet, allowing organizations to access infrastructure, platforms, and software services on demand. Among the three principal service models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)—SaaS has become the most prominent for enterprise software consumption [1], [3]. SaaS provides fully managed software applications delivered through the web, eliminating the need for organizations to maintain local infrastructure or install client-side software [4].

SaaS platforms are typically built on top of multitenant architectures that allow multiple clients to share a single software instance while preserving logical data separation and user-specific configurations. This architecture offers economies of scale and centralized management, but it also introduces security concerns related to resource isolation, data leakage, and shared vulnerability exposure [6], [8]. Virtualization and containerization technologies are frequently employed to facilitate resource abstraction and workload segregation across tenants.

Customization is a defining feature of SaaS offerings, enabling clients to tailor functionalities, interfaces, and workflows to match organizational needs. However, this flexibility can lead to inconsistent security baselines when tenant-specific configurations bypass vendor-imposed safeguards. Similarly, elastic scalability—a key advantage of cloud-native design—requires dynamic resource provisioning, which, if unmanaged, may compromise availability or lead to overexposed interfaces.

The SaaS delivery model relies heavily on web-based interfaces, APIs, and middleware integrations, enabling seamless user access and third-party interoperability. While these attributes enhance extensibility and user experience, they also expand the system's attack surface. Access management, encryption, and auditability become essential components of architectural design, especially when SaaS applications span multiple geographic and regulatory zones [3], [5].

Understanding the foundational architecture of SaaS platforms is essential for interpreting the security risks explored in the subsequent sections. By outlining the operational model, service dependencies, and deployment characteristics, this subsection provides a baseline for analyzing how specific architectural concerns—such as customization, multitenancy, and identity management—align with the security principles defined by the Information Security Management (ISM) framework.

B. *Information Security Management (ISM) Framework in SaaS Context*

The Information Security Management (ISM) framework has emerged as a central reference point in evaluating organizational security posture, particularly in cloud-based environments. Originally formulated to assess how organizations govern information security through structured controls, ISM provides a set of five interrelated dimensions: confidentiality, integrity, availability, access control, and risk management [10].

In the context of Software as a Service (SaaS), ISM offers a structured lens for assessing architectural and operational decisions that influence system security. SaaS environments operate across shared infrastructure and expose services through web interfaces and APIs, creating complex interdependencies between user access, data storage, and multi-tenant processing. Traditional perimeter-based security models are often insufficient in these settings, necessitating a layered and theory-driven approach to evaluate exposure and mitigation strategies [6], [10].

Each ISM dimension addresses a distinct aspect of security. Confidentiality refers to the protection of sensitive information from unauthorized access, particularly during data transmission and storage. Integrity ensures that system data and configurations are not modified maliciously or inadvertently. Availability focuses on the resilience of systems to ensure authorized access is not disrupted, especially under peak load or attack. Access control involves managing user permissions, authentication protocols, and identity governance mechanisms. Risk management serves as the integrating layer, evaluating the likelihood and impact of threats, and informing mitigation strategies across the other dimensions.

The literature on SaaS security increasingly references these constructs—explicitly or implicitly—as part of architectural analysis. For instance, multitenancy concerns are frequently discussed in terms of integrity and confidentiality risks, while API exposure is framed as a threat to access control and data leakage. Customization and scalability, when unmanaged, can lead to weakened availability and increased attack surfaces, highlighting the role of risk evaluation in architectural decision-making [1], [6], [10].

Although several frameworks have been proposed to model SaaS security, including Service-Oriented Architecture (SOA) and Model-Driven Architecture (MDA), the ISM framework uniquely integrates both technical and governance considerations, making it particularly suitable for holistic assessments. It allows researchers to classify concerns across multiple layers—ranging from interface logic to storage architecture—while retaining alignment with compliance and audit frameworks such as ISO 27001 and NIST [10].

As applied in this review, the ISM framework provides a reference model to analyze how the literature has conceptualized, addressed, or overlooked security dimensions within SaaS architecture. Subsequent sections examine the degree to which each architectural concern maps to one or more ISM domains, providing a cohesive basis for evaluating scholarly treatment of security in the SaaS landscape.

C. *Customization and Scalability*

Customization in SaaS environments refers to the ability of client organizations to tailor platform components, such as interfaces, workflows, and system logic, to meet specific operational needs. Scalability refers to a system's capacity to dynamically allocate computing resources—such as processing power, memory, and storage—in response to changes in workload demand, without compromising performance, reliability, or security. These architectural features have become core differentiators in SaaS adoption, but they also introduce significant security and configuration challenges [1], [3].

Ali et al. [8] highlight that customized deployments can weaken security by circumventing vendor-imposed safeguards, thereby producing heterogeneous system configurations across tenants. This "configuration drift"—a deviation from secure default settings over time—has been linked to increased attack surfaces and weakened enforcement of organizational policies [1], [6]. Additional complexity arises when customization is layered across UI, middleware, and access control mechanisms, often without unified governance [3]. Such divergences create fragmented enforcement of baseline security controls and increase the difficulty of maintaining consistent compliance across deployments [6].

Scalability amplifies these concerns. Aleem et al. [1] observe that horizontal scaling—particularly during demand spikes—can degrade the responsiveness of access management and real-time encryption controls. Dynamic scaling may also introduce automated provisioning of unsecured APIs or temporary storage nodes, increasing exposure to unauthorized access [5]. Several studies suggest that scalability often relies on third-party services or plugins, which may not conform to baseline security policies or data residency requirements, thereby introducing inconsistencies across operational environments [6], [8]. These issues intersect directly with the ISM framework.

Customization affects integrity when changes produce unpredictable system behavior or conflict with shared logic [10]. Scalability affects availability, as rapid resource shifts may destabilize access, load balancing, or session persistence [6]. Both customization and scalability introduce elevated risk, especially when controls are not centrally validated or when performance is prioritized over policy enforcement [1], [10].

Despite the recognized challenges, few studies provide comprehensive frameworks that address these dimensions in tandem. The literature remains divided—some focusing on customization for user experience and others on scalability for infrastructure performance—without integrating the security implications of these dynamics. This fragmentation underscores the need for research that holistically evaluates customizable, scalable SaaS systems through a security architecture lens [1], [6].

D. *Multitenancy and Shared Architecture*

Multitenancy is a foundational characteristic of SaaS platforms, enabling multiple client organizations (tenants) to share a single application instance and underlying infrastructure while maintaining logical separation of data and configurations. This architectural strategy allows for operational efficiency, centralized updates, and economies of scale, but it introduces substantial security challenges due to the co-residency of data and processes from different tenants [1], [6].

One of the primary risks associated with multitenancy is insufficient tenant isolation. In environments where virtualization or containerization is improperly configured or inconsistently applied, one tenant's access to shared memory, CPU cycles, or network interfaces may result in unauthorized data exposure or

leakage [6], [8]. Researchers have noted that shared components such as cache, session tokens, and logging infrastructure can become indirect vectors for lateral movement between tenant environments [1], [5]. These risks are exacerbated when tenants are permitted to deploy customized modules or middleware without rigorous validation [3].

Shared architecture also creates vulnerabilities related to resource contention and cross-tenant performance degradation. In high-load conditions, poorly enforced isolation boundaries may allow one tenant's activity to impact the availability or response time of others, which can be exploited to infer usage patterns or to trigger denial-of-service-like behaviors [5], [6]. Additionally, multitenancy increases the complexity of enforcing granular access controls, as identity federation, session management, and authorization must scale securely across all participating tenants [4].

Within the ISM framework, multitenancy directly implicates concerns around integrity, confidentiality, and access control. Integrity may be compromised if shared code execution or memory use leads to data overwriting or process manipulation. Confidentiality is threatened when boundaries between tenants are weak, whether due to insecure APIs, flawed isolation logic, or inadequate encryption [6], [10]. Access control becomes increasingly difficult to manage in multitenant environments with diverse access policies and federated identities, making role-based access control insufficient on its own [4], [10].

While various techniques such as tenant tagging, namespace segregation, and virtual machine isolation have been proposed, few implementations address the full spectrum of ISM-aligned security concerns. The literature tends to treat multitenancy risks in silos, without integrating performance, compliance, and governance dimensions into a unified security model [1], [6]. As SaaS platforms scale and integrate with other cloud services, the need for secure-by-design multitenant architectures becomes increasingly urgent.

E. *Integration and API Security*

Integration with external services is a defining characteristic of SaaS platforms, enabling extensibility, interoperability, and modularity across enterprise applications. These integrations are often facilitated through RESTful APIs, middleware components, and third-party connectors. While these tools enhance service orchestration and data mobility, they also expand the system's attack surface by exposing internal functions to the external environment [1], [3].

API endpoints that lack proper access validation or encryption may allow unauthorized parties to extract or manipulate data. Studies have shown that insufficient authentication mechanisms—particularly when relying on weak API keys, static tokens, or unaudited service accounts—can enable lateral movement within SaaS environments [5], [6]. Additionally, integration points often bypass traditional user interfaces, allowing automated interactions that are more difficult to monitor or constrain through standard security controls [4].

Middleware plays a critical role in managing integration logic, but it can introduce its own vulnerabilities when deployed with misconfigured permissions, outdated libraries, or unsecured gateways. The risk is magnified when middleware supports bidirectional data flows between internal applications and third-party platforms, especially when data transformation processes lack input validation or output encoding [6], [8]. In such configurations, even indirect exposure of system metadata or error messages can facilitate reconnaissance and exploit chaining.

From an ISM perspective, API integration concerns primarily map to confidentiality and access control. Confidentiality is at risk when APIs transmit sensitive information without encryption or expose data structures beyond what is necessary for service interaction. Access control is jeopardized when endpoint permissions are not granular, or when federated access models fail to enforce tenant-level separation [4], [10]. Some literature also flags integrity risks when API transactions are accepted without schema validation or when business logic is embedded in client-side operations [1], [6].

Despite growing attention to API threats, the literature remains fragmented in proposing standardized frameworks or validation protocols tailored to SaaS environments. Existing guidelines often mirror generic web security models, which may not account for the unique service composition and multi-tenant architecture of SaaS platforms [6]. This gap underscores the need for integration-aware security models that reflect the operational realities and shared responsibility inherent in SaaS ecosystems.

F. *Data Privacy and Regulatory Compliance*

Data privacy is a critical concern in SaaS environments due to the centralized storage, processing, and transmission of client information across geographically distributed infrastructure. SaaS providers often host data in multi-tenant environments or rely on third-party data centers, making it difficult for clients to control where and how their sensitive information is stored. This raises significant concerns over data sovereignty, unauthorized access, and regulatory non-compliance [1], [6].

One of the most cited regulatory frameworks in the SaaS context is the General Data Protection Regulation (GDPR), which mandates strict controls over personal data collection, usage, and cross-border transfers. SaaS applications that operate across multiple jurisdictions must align with country-specific data residency requirements, impose role-based access restrictions, and maintain audit logs to demonstrate compliance [3], [5]. The complexity of these legal obligations is compounded by the fact that many SaaS platforms integrate with external services, some of which may be located outside compliant jurisdictions [5], [6].

Auditability and data classification are also key components of compliance. SaaS platforms must support logging, monitoring, and forensics to ensure that data access and modification events are recorded and attributable. When audit trails are incomplete or decentralized across services, accountability becomes diluted. Studies have emphasized the need for metadata management and encryption-at-rest mechanisms to maintain compliance with both technical and governance requirements [6], [8].

From an ISM perspective, data privacy and regulatory compliance primarily map to confidentiality and risk management. Confidentiality is compromised when sensitive data is exposed to unauthorized personnel due to weak encryption, misconfigured permissions, or lack of tenant isolation. Risk management is essential to evaluate the legal and operational impact of non-compliance, especially in cloud-native ecosystems where responsibility for security is often shared between providers and clients [1], [10]. These concerns demand security strategies that are not only technically robust but also auditable and transparent across regulatory domains. Although regulatory frameworks provide extensive guidance, existing academic literature tends to focus on individual controls rather than integrated architectural strategies that ensure end-to-end compliance [6].

G. *ISM-Based Synthesis and Literature Gaps*

A review of the selected literature reveals significant variability in how architectural security concerns are addressed across SaaS environments. While individual risks—such as multitenancy isolation, unsecured APIs, and scalability-induced misconfigurations—are frequently discussed, the literature tends to treat these concerns in isolation rather than as interconnected components of a unified security architecture [1], [6]. This siloed treatment undermines efforts to develop comprehensive mitigation strategies and obscures the broader implications of architectural trade-offs in SaaS design [5].

From a theoretical standpoint, the ISM framework offers a valuable structure for assessing security across confidentiality, integrity, availability, access control, and risk. However, few studies explicitly apply ISM or similar comprehensive frameworks to evaluate SaaS architecture holistically. For example, confidentiality is commonly emphasized in the context of data privacy and encryption, yet access control and availability are often only partially explored, particularly in relation to integration and identity

management mechanisms [6], [10]. This inconsistency suggests an underutilization of multi-dimensional models that could otherwise guide balanced and scalable security design.

Another recurring limitation is the lack of security-oriented architectural models that are validated against regulatory or operational benchmarks. Studies frequently highlight vulnerabilities without offering design principles or implementation roadmaps that align with real-world compliance frameworks such as ISO 27001 or GDPR [3], [5]. Additionally, while risk is acknowledged across various domains—from API exposure to customization drift—quantitative or scenario-based risk assessments are largely absent, limiting the applicability of findings to enterprise settings [6].

The review also indicates that many studies rely on high-level conceptual discussions or static analyses of platform configurations. There is a notable absence of empirical validation, simulation, or comparative evaluation across SaaS vendors or deployment models [1], [6]. As SaaS ecosystems continue to evolve through microservices, serverless architectures, and AI-enabled automation, the security assumptions embedded in earlier literature may no longer hold. This creates an urgent need for continuous reassessment using structured frameworks like ISM to ensure ongoing relevance and resilience.

Overall, the literature reflects an increasing awareness of SaaS-specific architectural security concerns, but it lacks theoretical consistency, cross-dimensional analysis, and actionable guidance for practitioners. Addressing these gaps requires integration of technical, operational, and governance perspectives within a unified security architecture model grounded in frameworks such as ISM [10].

IV. Synthesis Of Findings

The analysis of academic literature across five key SaaS architectural concerns—customization, multitenancy, API integration, identity management, and regulatory compliance—highlights both depth and variation in how security is addressed. Each concern presents distinct technical and governance challenges, yet they frequently intersect at common pressure points, such as configuration consistency, access enforcement, and auditability [1], [6], [10]. These recurring intersections suggest the need for security evaluation approaches that consider not only individual components but also their interactions within a shared environment [5].

Customization and scalability raise concerns around maintaining consistent configurations, particularly when platform flexibility allows deviations from secure defaults. These concerns often emerge in multi-tenant contexts where shared logic and infrastructure magnify the impact of tenant-level modifications [1], [8]. Multitenancy itself introduces challenges related to data segregation and resource isolation, particularly under dynamic workloads and federated access models [6]. Integration via APIs further expands the exposure surface, especially when authentication and input validation mechanisms vary across endpoints [3], [5]. Identity and access management must account for decentralized permissions, multi-factor requirements, and the interplay between role-based models and federated identity systems [4], [6]. Data privacy and regulatory compliance, while more extensively addressed in the literature, also require systems-level alignment between architecture, encryption, data flows, and logging [1], [5], [10].

The ISM framework provides a unifying perspective to assess how these architectural concerns align with five core security dimensions. Confidentiality and access control are central to most discussions, especially in the context of API usage, data protection, and tenant segmentation [6], [10]. Availability and integrity are often implicated in scalability and middleware decisions, though they are less frequently discussed as explicit design criteria [1], [6]. Risk management appears across all domains, often as a high-level concern, underscoring the relevance of structured methodologies to evaluate and prioritize threats systematically.

Notably, some studies emphasize technical solutions, such as encryption or containerization, while others focus on policy-level mechanisms like compliance checklists or audit trails. This variety reflects the multidimensional nature of SaaS security rather than a weakness in the literature. However, it also illustrates

the value of frameworks like ISM that can connect these perspectives and help organize security efforts across both technical and operational layers [5], [10].

The findings demonstrate that effective SaaS security architecture requires an integrated strategy—one that acknowledges the interdependence of architectural components, aligns with ISM security dimensions, and supports consistent, scalable implementation. Continued refinement of theory-guided models, such as ISM, may help bridge existing gaps and guide future research and practice toward more resilient and accountable SaaS ecosystems [10].

V. Implications For Future Research And Practice

The literature reviewed in this study offers multiple points of entry for advancing both scholarly inquiry and practical implementation of secure SaaS architectures. For researchers, one implication is the need to more systematically evaluate the relationship between architectural design and multi-dimensional security outcomes. While various studies have examined individual concerns—such as API vulnerabilities or multitenancy isolation—few have leveraged theory-driven models like ISM to assess how these issues interact or compound within live systems [1], [6], [10]. Future research could benefit from empirical methods, including case studies or simulations, to validate security design patterns across different SaaS deployment models [5].

From a technical standpoint, there is a need for studies that explore dynamic configuration management, secure extensibility, and scalable access governance frameworks. These are especially relevant as SaaS platforms increasingly rely on microservices, serverless components, and third-party integration layers, each of which introduces new security trade-offs. Additionally, research that maps architectural decisions to specific ISM dimensions could provide clearer metrics for comparing platform security readiness or benchmarking vendor practices [6], [10].

For practitioners—including cloud architects, security engineers, and IT governance professionals—the findings underscore the importance of aligning technical design with strategic security models. Customization features should be evaluated not only for usability but also for how they alter the system's security baseline. Similarly, multitenancy and API integration require architecture-level planning to ensure tenant isolation, encrypted communication, and federated identity enforcement [1], [3], [6]. Incorporating the ISM framework into system development lifecycles may help ensure that each dimension—confidentiality, integrity, availability, access control, and risk—is explicitly addressed from early design through deployment.

Audit professionals and compliance officers may also benefit from using ISM-aligned checklists to evaluate SaaS solutions. The increasing complexity of regulatory requirements—especially in multi-jurisdictional deployments—makes it essential to operationalize auditability, data classification, and access logs within architectural blueprints [5], [10]. SaaS vendors and clients should collaborate to define shared responsibility models that account for platform configuration, data storage practices, and third-party service chains.

Overall, these implications highlight the necessity of bridging research, technical design, and governance practices. Applying the ISM framework not only as an analytical tool but also as a development and assessment guide could support more resilient, auditable, and scalable SaaS ecosystems.

VI. Conclusion

This study presented a structured review of security challenges in Software as a Service (SaaS) architecture, synthesizing academic literature across five core areas: customization and scalability, multitenancy, integration and API security, identity and access management, and regulatory compliance. The review was guided by the Information Security Management (ISM) framework, which enabled a

multidimensional analysis of security concerns aligned with confidentiality, integrity, availability, access control, and risk management [10]. Findings highlighted that while significant progress has been made in identifying vulnerabilities and proposing safeguards, architectural concerns are often addressed in isolation rather than as interconnected components of a secure SaaS environment [6].

Customization introduces complexity in maintaining consistent security baselines, especially when tenant-specific configurations are permitted without centralized validation. Multitenancy raises concerns around tenant isolation, shared logic, and resource contention, particularly in horizontally scalable environments [1], [6]. API integrations, though essential for interoperability, expose platforms to risks stemming from inconsistent authentication, inadequate input validation, and federated access models [3], [5]. Identity management is challenged by fragmented policy enforcement, especially when distributed across cloud components. Meanwhile, data privacy and regulatory compliance demand encryption, auditability, and data sovereignty—attributes that are difficult to enforce in multi-jurisdictional deployments [1], [5], [10].

The ISM framework proved effective in identifying the overlaps and inconsistencies within the literature. While confidentiality and access control are frequently discussed, dimensions such as availability and risk are addressed less consistently, particularly in the context of emerging deployment models and third-party service chains [6]. This reinforces the value of structured theoretical models for guiding both academic analysis and operational design in SaaS security architecture.

Future research can build on this foundation by operationalizing ISM into actionable evaluation criteria, integrating it with system development lifecycles, and applying it in empirical settings such as case studies or controlled experiments. Practitioners, vendors, and auditors may also benefit from adopting ISM-informed checklists and governance templates to strengthen architectural security and align with compliance frameworks [5], [10]. By framing SaaS security through the lens of architectural interdependence and theoretical rigor, this study contributes to a growing body of knowledge that seeks to bridge research, practice, and governance in cloud-native environments.

ACKNOWLEDGMENT

This paper builds upon research conducted as part of the author's doctoral dissertation at the University of the Cumberlands. I extend my heartfelt appreciation to Dr. Oludotun Oni, my dissertation chair, for his guidance during the development of the research design and literature framework. I also acknowledge the University of the Cumberlands Writing Center for their academic support throughout the manuscript preparation process. This manuscript also benefited from the use of the Grammarly tool, which supported academic clarity and grammatical precision during the writing process. Special thanks to my family and my dogs for their unwavering support during this research journey. I am also grateful to the broader scholarly community whose prior work helped inform and shape this review.

REFERENCES

[1] M. Aleem, S. A. K. R. Naqvi, and S. R. S. Kumari, "A review of SaaS security issues and solutions," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, pp. 315–321, 2020.

[2] A. Sill, "The design and architecture of microservice-based software-as-a-service infrastructure," IEEE Cloud Computing, vol. 3, no. 5, pp. 76–80, Sep./Oct. 2016.

[3] Y. S. Al-Dhuraibi, M. Paraiso, N. Djarallah, and P. Merle, "Elasticity in cloud computing: State of the art and research challenges," IEEE Transactions on Services Computing, vol. 11, no. 2, pp. 430–447, Mar./Apr. 2018.

[4] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24–31, Nov./Dec. 2010.

[5] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Information Sciences, vol. 305, pp. 357–383, 2015.

[6] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[7] A. Fernandes, L. Soares, J. Gomes, M. Freire, and P. Inácio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, pp. 113–170, 2014.

[8] R. Ranchal, Y. J. Ko, D. Sandhu, and L. B. Othmane, "Policy-driven security management for SaaS applications," in Proc. IEEE 10th Int. Conf. on Cloud Computing, Honolulu, HI, USA, 2017, pp. 780–787.

[9] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST Special Publication 800-145, Gaithersburg, MD, USA, Sep. 2011.

[10] M. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: From single to multi-clouds," in Proc. 45th Hawaii Int. Conf. on System Sciences, Maui, HI, USA, 2012, pp. 5490–5499.

[11] S. S. Boda, *Exploring Software as a Service Security Architecture Challenges and Considerations*, Ph.D. dissertation, University of the Cumberlands, Williamsburg, KY, USA, 2024. [Online]. Available: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=Y097WfQAAAAJ&citation_for_view=Y097WfQAAAAJ:u-x6o8ySG0sC

[12] H. Hassan, R. Alenezi, and F. Khan, "Security challenges in SaaS architecture: An evolving landscape," Journal of Cloud Computing, vol. 11, no. 1, pp. 22–36, 2022.

[13] M. Humayun, M. Jhanjhi, S. A. Ahmed, and A. M. Noor, "A multivocal literature review on SaaS security architecture: Trends, risks, and control mechanisms," Computers, vol. 11, no. 4, pp. 55–72, 2022.

[14] A. Koli, B. Upadhyay, and P. Gupta, "A security framework for SaaS platform adoption: An Indian SME case," Journal of Information Security, vol. 14, no. 2, pp. 113–126, 2023.

[15] J. Litmala, "Organizational collaboration and data ownership in SaaS environments," ACM Transactions on Internet Technology, vol. 23, no. 1, pp. 1–21, 2023.