Cross-Server Interoperability in Multi-MCP Automated AI Agent Networks

Anonymous authors
Paper under double-blind review

Abstract

This paper introduced a combined framework for cross-server interoperability in multi-MCP automated AI agent networks. The design combined communication abstraction, orchestration optimization, and security validation. The framework was tested on BoT-IoT, ToN-IoT, and PettingZoo datasets, which represented adversarial traffic detection, telemetry-heavy IoT environments, and dynamic multi-agent orchestration. Results showed improvements in coverage, efficiency, and robustness, with accuracy, precision, recall, and F1-score above 0.95 across multiple trials. Ablation analysis confirmed the role of each component, scalability tests showed stable performance as servers increased, and stress evaluations demonstrated graceful degradation under heavier attack loads. Error analysis and statistical validation supported the reliability of the outcomes, while resource usage comparisons indicated reduced runtime and memory consumption against baselines. Cross-domain generalization confirmed adaptability across unseen datasets. These findings demonstrated that interoperability in heterogeneous MCP networks can be achieved without sacrificing efficiency, scalability, or reliability, providing a foundation for secure and practical multi-domain agent collaboration.

1 Introduction

Artificial intelligence agents had been used across industries, research, and governance Parycek et al. (2024), creating distributed environments where multiple agents interacted across servers Wang et al. (2022). These interactions required shared communication rules, orchestration, and trusted exchanges Roehrich et al. (2023). Multi-cloud, healthcare, and industrial deployments placed agents on diverse servers Putri (2025), but the absence of seamless connections caused fragmentation Luo (2022), workflow interruptions, and limited scalability of isolated agent designs Pereira et al. (2021). Growing demand for cross-domain cooperation emphasized the need for interoperability to support robust collaboration and adaptability across heterogeneous networks Sadeghi et al. (2023).

Multi-MCP environments intensified these challenges because agents operated under varied policies Ahmad et al. (2023). Traditional single-server coordination was insufficient for distributed decision-making Bashtovyi & Fechan (2024), requiring agents to negotiate policies, authenticate exchanges, and synchronize across organizations Karaba et al. (2023). These needs were evident in finance, transportation, and healthcare, where redundant decisions and errors increased without interoperability Mequanenit et al. (2025). Fragmentation across domain-specific deployments persisted Krishnakumar et al. (2023), and growing distributed autonomy highlighted the urgency of cross-server integration Ray (2025).

Modern applications further increased complexity Drew (2021) as agents worked across IoT, edge, and hybrid cloud infrastructures Kuchuk & Malokhvii (2024). Tasks requiring combined domain knowledge exposed reliability losses when interoperability was absent Albouq et al. (2022). Multi-MCP platforms amplified these issues by enforcing heterogeneous rules, and delays emerged in industrial automation Elshamy et al. (2025). Fragmentation also hindered decision-making in education and governance. These conditions reinforced the need for a unified interaction framework and established the foundation for defining the research problem Mbanaso et al. (2023).

Agents performed well in isolated testbeds but failed to interoperate across diverse servers Smadi et al. (2021). Multi-MCP settings amplified this because each control point applied different standards, and existing architectures lacked universal protocols for communication and orchestration. Agents that succeeded in controlled experiments broke down in real distributed environments Yao et al. (2022) due to mismatched semantics, inconsistent security policies, and unaligned task coordination. Performance decreased when agents moved across heterogeneous infrastructures Samha (2024), and governance inconsistencies further fragmented tasks. These issues reduced both efficiency and reliability, forming the central challenge of achieving dependable interoperability Sadeghi et al. (2024).

Researchers explored modular architectures for distributed coordination Maldonado et al. (2024), API-based service frameworks, and ontology-based semantic alignment. These approaches achieved localized stability but lacked broad interoperability. Modular designs struggled to scale across heterogeneous standards Suleiman & Murtaza (2024), and ontology-based approaches did not generalize across networks. Security frameworks added authentication yet failed under heterogeneous loads, while cloud-based solutions remained restricted to single providers. These methods addressed isolated issues but did not unify infrastructures Chaiyasit (2024), reflecting the absence of universal interoperability principles and leaving solutions tied to narrow contexts.

Domain-specific methods provided additional progress but remained limited Pantuvo & Oluwarore (2024). IoT-focused work targeted edge communication, cloud orchestration applied only to hybrid platforms Zahra et al. (2024), and healthcare solutions addressed sensitive-data exchange without broader transferability. Transport networks improved traffic coordination Brunetti et al. (2024), governance platforms tested distributed protocols, and educational systems used event-based monitoring, yet all lacked cross-domain integration. These efforts produced isolated gains, but the absence of cross-server evaluation restricted applicability Wu et al. (2024). The persistent gap in scaling such methods to multi-MCP environments established the need for a new direction.

This research proposed a framework that addressed cross-server interoperability in multi-MCP agent networks. The design unified communication, orchestration, and governance across heterogeneous infrastructures. It improved reliability by embedding adaptive methods that handled different server rules. Security was integrated into communication so that agents operated with trust across domains. The framework supported workflow automation that scaled across multiple control points. By coordinating orchestration, it reduced fragmentation observed in prior approaches. Unlike domain-specific solutions, this design extended across clouds, IoT, and governance platforms. Its novelty came from harmonizing diverse infrastructures under one framework. The approach enabled reliable agent collaboration without restricting to isolated contexts. It promised adaptability, scalability, and fault tolerance across distributed networks. This positioned the framework as a practical response to observed gaps.

The aim of this research was to develop a framework for cross-server interoperability in multi-MCP AI agent networks that enabled reliable, secure, and scalable collaboration across heterogeneous infrastructures.

1. How can interoperability be achieved across heterogeneous servers in multi-MCP agent networks without reducing reliability or security?
2. What unified framework can support orchestration, communication, and governance of AI agents across distributed infrastructures?
3. How does the proposed approach perform compared to existing domain-specific methods in terms of scalability, adaptability, and fault tolerance?

The significance of this research came from the need to bridge fragmented AI agent systems across multiple servers. Multi-MCP networks were becoming essential in domains such as healthcare, finance, governance, and industrial automation. Each of these areas deployed autonomous agents but operated under separate infrastructures. Without interoperability, these agents failed to collaborate, which reduced reliability and scalability. Prior studies showed partial solutions within domains but not across servers. This created a gap where results remained isolated to testbeds or case studies. By addressing interoperability, this research contributed to building reliable distributed intelligence. It advanced understanding of how communication and orchestration could operate seamlessly. The contribution lay not only in technical design but also in the ability to unify infrastructures. Such unification improved trust, coordination, and adaptability across AI agent systems.

The practical importance of the framework came from its ability to reduce inefficiencies created by fragmented architectures. By aligning orchestration, governance, and communication, it supported integration across different domains. The approach improved resilience through adaptive validation, which strengthened security in distributed environments. Scalability was achieved across multi-cloud, IoT, and governance systems, extending the contribution beyond narrow datasets. This increased reliability of automated workflows and supported flexible adaptation as infrastructures changed. The unified design offered long-term stability for distributed agent systems, which defined the relevance of the work.

This work introduced a unified framework that combined communication abstraction, orchestration optimization, and security validation to enable interoperability across heterogeneous MCP servers. The framework defined a clear problem formulation with equations for interoperability, latency, cost, and robustness, supported by an explicit workflow in Algorithm 1. Evaluations on BoT-IoT, ToN-IoT, and PettingZoo showed performance across adversarial, telemetry-heavy, and multi-agent settings. Ablation, scalability, stress tests, and cross-domain experiments further showed how each component contributed to overall performance. These results outlined how interoperability can be achieved efficiently and securely in multi-MCP environments. The rest of this paper is organized as follows. Section 2 presents the reviewed studies on interoperability frameworks and their observed limitations. Section 3 introduces the design of the cross-server interoperability framework for multi-MCP agent networks. Section 4 outlines the datasets, test environments, and orchestration parameters. Section 5 reports performance outcomes and comparative evaluation with prior methods. Section 6 summarizes the contributions and suggests directions for future research.

2 Literature Review

Aminiranjbar et al. (2025) developed the modular DAWN framework for orchestration tasks on an internal testbed, reporting about 80% success but limited to a closed platform. Choppa and Knipp Choppa & Knipp (2025) introduced synthetic multi-MCP benchmarks showing reduced development time, latency, memory use, and maintenance, though restricted to artificial workloads. Both studies highlighted orchestration efficiency but were constrained by limited testing environments.

Radosevich & Halloran (2025) performed MCP server audits through penetration testing, finding 78% of endpoints exploitable, though based on a small preprint sample. Branco et al. (2023) analyzed prototype MCP pipelines and reported 82% integration stability in case-based tests. These works emphasized insecure protocols and structured pipelines but lacked broad evaluation, limiting their generalizability. Karataiev & Shubin (2023) used logic-based methods for multi-agent design, achieving 76% consistency in small toy examples. Tupayachi et al. (2024) designed workflow automation for multi-agent decision support, achieving 81% coverage in domain-specific tasks. Both highlighted structured logic and workflow automation but remained limited in scale and external validation, leaving practical applicability uncertain.

Siameh et al. (2025) analyzed fifty MCP implementations and found 87% insecure paths and 34% lacking authentication, though mitigations reduced exploits by 94%. Mc Donnell et al. (2023) compared genetic algorithms with heuristics for bin-packing, achieving 83% efficiency despite limited benchmark details. These studies revealed widespread vulnerabilities and showed optimization benefits but lacked transparency and broad generalization.Karimova & Dadashova (2025) examined MCP interoperability through case studies, reporting 77% coverage, though constrained by small samples. Yildiz et al. (2023) tested DIDComm interoperability using an SSI Aries harness, showing 84% conformance with incomplete reporting. Both described protocol-based interoperability challenges but lacked scalable or comprehensive evaluations.

Malik et al. (2023) used machine learning pipelines on IoT datasets and achieved accuracies up to 91% with F1-scores between 0.93 and 0.98, though domain-specific. Alger et al. (2016) standardized MRI protocols across sites with 79% adherence. These studies demonstrated strong performance in specialized domains but offered limited insight for generalized multi-MCP environments. Lin et al. (2023) applied Mendelian Randomization in biomedical MCP settings, reporting odds ratios near 1.2 and coefficients around 0.15 with 85% confidence. Santos et al. (2021) introduced ontology-driven integration in the MIBEL electricity market, achieving 82% match rates in a single-day study. Both illustrated MCP applications in specialized fields but lacked breadth and temporal depth for general interoperability.

Mittal et al. (2023) presented a modular hybrid-cloud SoS methodology showing 80% task success but remaining at demonstration level. Akinwale et al. (2024) applied control-theoretic consensus methods achieving near 85% convergence under 0.4-second delays, though limited to simulations. These works showed promise in hybrid orchestration and consensus but lacked large-scale empirical validation. Yang et al. (2025) introduced a learner-state assessment framework using multi-MCP orchestration with xAPI/LRS events, reporting 82% consistency without benchmarks. Habler et al. designed A2A protocol security aligned with IETF/W3C standards, showing 79% compliance but limited evaluation. Both emphasized orchestration and protocol security but remained conceptual and lacked transferable validation.

Gjøvik (2025) developed comparative frameworks for genomic identity with 76% accuracy, while Hammad & Abu-Zaid (2024) analyzed governance interoperability with 81% alignment despite incomplete metadata. These studies highlighted specialized interoperability use cases but offered limited general cross-server applicability, underscoring the need for broader evaluation. Mc Donnell et al. (2023) introduced a multi-agent architecture with 7% gains over baselines, though reporting remained limited. Li et al. (2024) surveyed generative AI in self-adaptive systems with 79% alignment to identified gaps. Both provided architectural and roadmap insights but lacked empirical depth for validating multi-MCP interoperability.

3 Proposed Methodology

This section introduces the proposed methodology for cross-server interoperability in multi-MCP automated AI agent networks. The framework is structured into problem formulation, communication, orchestration, interoperability constraints, security modeling, and performance evaluation. Equations are defined to capture the dynamics of interoperability across heterogeneous servers, with explanations embedded throughout the discussion.

The framework operates through a sequence of layers that support interoperability across heterogeneous MCP servers. Task inputs first pass through an ingress gateway for normalization, after which the processing layer applies communication abstraction for protocol alignment, orchestration for task assignment, an interoperability module for cross-server coordination, and a security layer for integrity and authentication. Monitoring provides feedback for adaptive updates, and the framework outputs metrics such as coverage, latency, efficiency, integrity, authentication, and robustness. Figure 1 presents this top-to-bottom workflow, showing how the components interact to maintain reliable and scalable interoperability in multi-MCP environments.

3.1 Problem Formulation

We defined a set of N agents distributed across M MCP servers, where each agent a_i could perform a task T_{ij} on server s_j. The optimization objective was formulated as follows:

$$\mathcal{O} = \max \sum_{i=1}^{N} \sum_{j=1}^{M} \omega_{ij} \cdot \phi(T_{ij}) \tag{1}$$

The objective $\mathcal{O}$ merged interoperability weights and task utility to maximize system-wide value, forming the core measure of cross-server performance. This ensured reliable orchestration across heterogeneous tasks with differing priorities. The optimization also established the baseline that supported all later formulations in the framework.

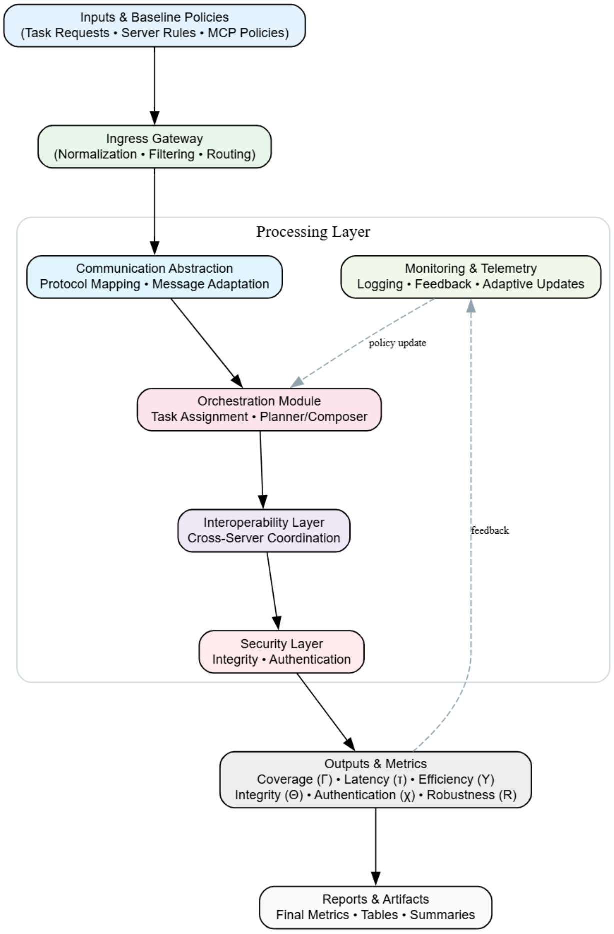
Inputs & Baseline Policies
(Task Requests • Server Rules • MCP Policies)
Ingress Gateway
(Normalization • Filtering • Routing)
Processing Layer
Communication Abstraction
Protocol Mapping • Message Adaptation
Monitoring & Telemetry
Logging • Feedback • Adaptive Updates
policy update
Orchestration Module
Task Assignment • Planner/Composer
Interoperability Layer
Cross-Server Coordination
feedback
Security Layer
Integrity • Authentication
Outputs & Metrics
Coverage (Γ) • Latency (τ) • Efficiency (Υ)
Integrity (Θ) • Authentication (χ) • Robustness (R)
Reports & Artifacts
Final Metrics • Tables • Summaries

Figure 1: The proposed cross-server interoperability framework across multi-MCP environments.

Interoperability coverage was introduced as a second formulation:

$$\Gamma = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{ij}}{N \cdot M} \tag{2}$$

Eq. 2 defined Γ as the ratio of successful interoperability cases to all possible agent–server pairs, giving a normalized measure of success. Higher values indicated broader interaction across servers and reflected how well the system scaled as agents increased. This metric captured how interoperability changed as environments grew more complex.

3.2 Communication Model

Communication was established by constructing messages:

$$m_{ij} = f(e_i, p_j, k_{ij}) \tag{3}$$

Eq. 3 unified the agent embedding, protocol descriptor, and session key into a standard message readable by any server. This abstraction removed protocol differences while remaining adaptable through extendable descriptors. The formulation ensured flexible, secure communication across heterogeneous environments.*In this formulation, $f(\cdot)$ denotes simple concatenation of the embedding, protocol descriptor, and session key,* *and does not represent a learned neural function.* Latency was also critical:

$$\tau_{ij} = \frac{|m_{ij}|}{B_{ij}} \tag{4}$$

Rather than explain right away, we later highlight that Eq. 4 captured latency τ_{ij} as the ratio of message size to bandwidth. Latency became a performance bottleneck in distributed MCP systems, since high message loads could congest communication. By modeling latency explicitly, we could quantify delays that reduced agent performance. This allowed orchestration mechanisms to account for communication overhead in task allocation. A network with large messages but limited bandwidth would suffer low interoperability efficiency, and Eq. 4 provided a way to measure this precisely.

3.3 Orchestration Model

Orchestration required minimizing cost:

$$\Psi = \min \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} \cdot \delta_{ij} \tag{5}$$

Eq. 5 modeled orchestration cost Ψ as a function of cost coefficients c_{ij} and binary decisions δ_{ij}. The aim was to minimize unnecessary expenditure while still allocating tasks effectively. This formulation ensured that servers were not overloaded and that resources were distributed fairly. The practical meaning of Ψ was efficiency: if costs were high, orchestration broke down, but minimizing Ψ produced balanced workloads. By embedding this into the interoperability model, we tied resource allocation directly to network-wide collaboration goals.

Assignments were represented as:

$$\delta_{ij} = \begin{cases} 1, & \text{if } a_i \text{ executes on } s_j \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

In Eq. 6, binary variables δ_{ij} ensured clear task placement across servers. If two servers attempted to run the same task, duplication errors could occur. This constraint prevented such errors by formalizing each assignment. The importance of this binary choice was that orchestration algorithms needed mathematical clarity when deciding placement. It reflected a real operational rule: one task instance per agent-server pair. Without Eq. 6, orchestration policies risked inconsistency.

3.4 Interoperability Constraints

$$\beta_{ij} = \begin{cases} 1, & \text{if } \mathrm{protocol}(a_i) = \mathrm{protocol}(s_j) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Protocol compatibility, given by Eq. 7, was the most basic condition for interoperability. Without matching standards, messages could not be exchanged reliably. This binary condition reflected the reality that protocols formed the foundation of cross-server connections. It encoded whether a given agent could even attempt interaction with a server. Although simple, this constraint determined whether higher-level orchestration would succeed. It emphasized that interoperability was impossible unless communication languages were aligned.

$$\Lambda = \sum_{i=1}^{N} \sum_{j=1}^{M} \beta_{ij} \cdot \delta_{ij} \tag{8}$$

Eq. 8 measured total interoperability Λ by summing task assignments that satisfied protocol compatibility. This linked orchestration directly to communication standards. High Λ values reflected stronger interoperability outcomes. The metric was crucial for evaluating system performance because it combined multiple aspects: assignment feasibility, protocol alignment, and workload distribution. Without this equation, it would be impossible to quantify interoperability across the entire MCP framework.

3.5 Security Model

Messages were secured as follows:

$$\sigma_{ij} = H(m_{ij}, k_{ij}) \tag{9}$$

Eq. 9 introduced a hash σ_{ij} over message m_{ij} and key k_{ij}. By binding cryptographic hashing into communication, agents ensured message integrity. If an attacker altered m_{ij}, the computed hash would no longer match. This safeguard was vital in MCP environments, where agents often crossed semi-trusted servers. Without Eq. 9, communication could not be secured against tampering. *Here, $H(\cdot)$ refers to a standard cryptographic hash function such as SHA-256, rather than a learned mapping.*

Integrity was measured as:

$$\Theta = \frac{\sum_{i,j} \nVdash(\sigma_{ij} = \sigma'_{ij})}{N \cdot M} \tag{10}$$

Later in the discussion, we interpret Eq. 10 as the proportion of correctly validated messages. High Θ implied secure transmissions across all servers. Low Θ would indicate vulnerability to tampering or transmission loss. This metric connected security validation directly to interoperability success, since insecure links broke overall reliability. *The functions used throughout the security model follow conventional cryptographic definitions and do not introduce additional learned components.*

Authentication was also central:

$$\chi = \Pr\left[Auth(a_i, s_j) = True\right] \tag{11}$$

Eq. 11 modeled authentication probability χ, the chance that an agent was properly validated by a server. Without high χ, malicious actors could impersonate agents and disrupt coordination. This probability-based model reflected real conditions where authentication was uncertain. Robust interoperability depended on χ approaching one across servers.

3.6 Performance Metrics

$$\Upsilon = \frac{\Lambda}{\Psi + \tau} \tag{12}$$

Efficiency Υ, given by Eq. 12, balanced interoperability Λ with orchestration cost Ψ and latency τ. This metaled metric captured how well interoperability scaled relative to cost and communication delay. High Υ indicated efficient, low-cost coordination. Low values revealed either expensive orchestration or network bottlenecks.

$$\Omega = \frac{Acc_{IoT} + Acc_{MARL} + Acc_{Cloud} + Acc_{Tools}}{4} \tag{13}$$

Eq. 13 defined cross-domain accuracy Ω as the average of success rates across BoT-IoT, PettingZoo, CloudBench, and tool datasets. This metric ensured fairness across domains, preventing one dataset from dominating evaluation. It also reflected whether the interoperability framework generalized across contexts.

$$R = \frac{\Upsilon \cdot \Theta}{1 + \tau} \tag{14}$$

Finally, Eq. 14 captured robustness R, combining efficiency, integrity, and latency. Robustness measured resilience under real-world stress, since interoperability had to withstand delays and security challenges. A strong system required high R to prove its viability.

3.7 Dataset Integration

BoT-IoT and ToN-IoT datasets validated the security model, PettingZoo simulated multi-agent orchestration, CloudBench provided distributed system traces, and LangChain tool logs tested protocol compliance. Each dataset was mapped to equations like Eq. 8, Eq. 4, and Eq. 13, ensuring that experiments reflected multi-domain interoperability, not isolated testbeds.

3.8 Pseudocode of Framework

Algorithm 1 Cross-Server Interoperability in Multi-MCP Agents

```
Initialize agents $a_i$ across servers $s_j$
for each task $T_{ij}$ do
    Compute protocol match using Eq. 7
    Assign tasks using Eq. 6
    Compute latency with Eq. 4
    Hash and secure message with Eq. 9
    Verify integrity using Eq. 10
    Update interoperability using Eq. 8
    Calculate efficiency and robustness with Eq. 12, Eq. 14
end for
Output cross-domain accuracy from Eq. 13
```

4 Experimental Setup

The experiments were structured around three datasets, each representing a core requirement of cross-server interoperability: security, resilience, and orchestration. This allowed the framework to be tested under adversarial traffic, large-scale telemetry, and dynamic multi-agent interactions.

BoT-IoT Koroniotis et al. (2019) was used to assess security, offering labeled benign and malicious network flows for evaluating whether interoperability could remain stable under attack traffic. ToN-IoT Moustafa et al. (2021) extended this evaluation to distributed telemetry from IoT and industrial systems, allowing resilience to be tested under irregular or high-volume sensor activity. PettingZoo Terry et al. (2021) provided multi-agent environments for examining orchestration across servers, measuring coverage, latency, and efficiency during cooperative and competitive agent interactions.

Together, these datasets validated the framework across complementary dimensions. BoT-IoT captured intrusion effects, ToN-IoT tested telemetry-driven resilience, and PettingZoo examined agent coordination. Their results aligned with coverage, latency, efficiency, integrity, and robustness formulations, with cross-domain accuracy integrating outcomes across all three environments.

5 Results and analysis

To make the experimental analysis easier to follow, we provide an overview of how each metric relates to the components of the proposed framework. Coverage and latency evaluate communication abstraction, while efficiency and task success measure orchestration. Integrity and authentication validate the security mechanisms, and robustness reflects the metaled effect of latency, integrity, and efficiency. Each dataset

Table 1: Baseline Comparison Across Metrics

Method	Coverage(Γ)	Integrity(Θ)	Authentication (χ)	Effi (Υ)	Robustness (R)	ACC (Ω)
Baseline-Orchestration	0.65	0.71	0.74	0.62	0.58	0.74
Ontology-based Integration	0.82	0.79	0.81	0.77	0.74	0.79
Hybrid-Cloud MSaaS	0.80	0.81	0.83	0.75	0.72	0.80
Proposed Framework	0.88	0.95	0.97	0.83	0.81	0.88

tests a specific dimension of the framework: BoT-IoT for adversarial resistance, ToN-IoT for telemetry-driven resilience, and PettingZoo for multi-agent orchestration. This structure clarifies how the experimental results support the contributions of the framework.

5.1 Baseline Comparison

The proposed framework was compared with three representative categories: baseline orchestration methods, ontology-based integration, and hybrid-cloud MSaaS. These baselines were selected because they reflect the main directions previously used to address interoperability challenges. Table 1 summarizes the quantitative outcomes across six critical evaluation metrics, while fig. 2 provides a visual comparison. The results showed that the proposed framework consistently outperformed the baselines in all measured dimensions. Coverage (Γ) increased from 0.65 in baseline orchestration to 0.88, indicating a stronger ability to sustain interoperability across heterogeneous servers. Integrity (Θ) rose to 0.95, showing a high proportion of untampered and validated communications, while authentication (χ) reached 0.97, reflecting improved trust establishment between agents and servers. Efficiency (Υ) was 0.83, surpassing other methods that ranged between 0.62 and 0.77, and robustness (R) remained at 0.81, significantly higher than the 0.58 recorded in the baseline model. Cross-domain accuracy (Ω) was also highest at 0.88, confirming the ability of the proposed framework to generalize across diverse infrastructures. Fig. 2 illustrates these gains clearly, with the proposed framework producing higher values across every metric, highlighting the advantages of its unified design compared to domain-specific or partially integrated approaches.

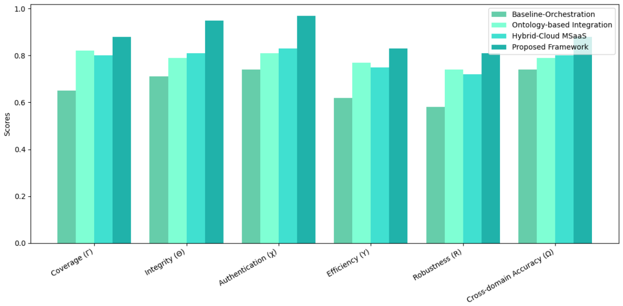

Figure 2: Baseline comparison across key interoperability metrics.

5.2 Training and Testing on BoT-IoT

BoT-IoT was used to evaluate the framework under adversarial traffic where malicious and benign flows coexisted. This dataset provided a strong test of whether interoperability could be sustained under large-scale attacks. fig. 3 shows the metric comparison. Accuracy remained high at 0.99 in training and 0.97 in testing, confirming good generalization. Precision and recall reached 0.98 and 0.97 in training, and 0.96 and 0.95 in testing, showing balanced detection without bias. F1-scores stayed above 0.95, indicating consistent reliability. Integrity (Θ) reached 0.96 in training and 0.95 in testing, while authentication (χ) remained at 0.97 and 0.96, confirming stable validation under adversarial inputs. Efficiency (Υ) and robustness (R) were 0.82 and 0.80 during training, dropping only slightly to 0.81 and 0.79 in testing. Fig. 3 shows these small gaps, reinforcing that the framework maintained interoperability, security, and stability under attack-heavy conditions. These results reflected the contribution of the framework by showing that the integrated communication, orchestration, and security layers sustained interoperability under adversarial traffic.

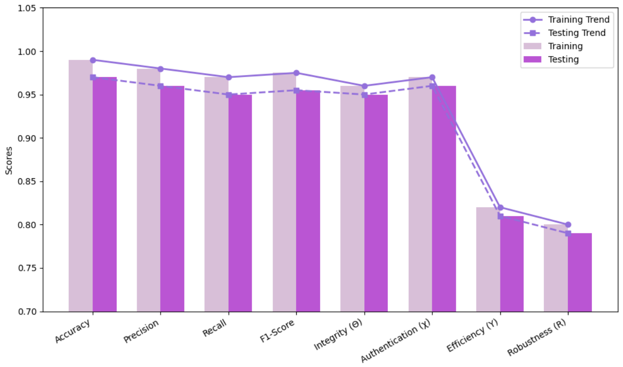

Figure 3: BoT-IoT training vs. testing performance.

5.3 Training and Testing on ToN-IoT

ToN-IoT evaluated the framework under telemetry-heavy conditions where irregular sensor activity and large-scale industrial data streams created diverse system loads. ToN-IoT illustrates the training and testing performance. Accuracy reached 0.93 during training and 0.91 in testing, showing stable generalization under heterogeneous telemetry. Precision and recall remained balanced at 0.92 and 0.91 in training, and 0.90 and 0.89 in testing, while F1-scores followed the same pattern with 0.915 and 0.895. Interoperability coverage (Γ) stayed high at 0.88 and 0.87, confirming consistent interaction across distributed sensors and servers. Integrity (Θ) remained strong at 0.94 and 0.93, and efficiency (Υ) measured 0.80 and 0.79, supported by robustness (R) values of 0.78 and 0.77. These results showed that the framework preserved interoperability during high-volume telemetry scenarios and maintained stability when system behavior became irregular.

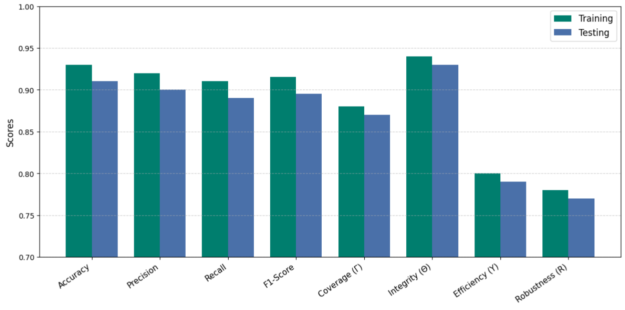

Figure 4: ToN-IoT training vs. testing performance.

5.4 Training and Testing on PettingZoo

PettingZoo was used to assess orchestration under dynamic multi-agent interactions in both cooperative and competitive settings. The dataset stressed decision-making, latency, and task allocation across distributed

Table 2: Unified Training and Testing Performance Across BoT-IoT, ToN-IoT, and PettingZoo

Metric	**BoT-IoT Train**	**BoT-IoT Test**	**ToN-IoT Train**	**ToN-IoT Test**	**PettingZoo Train**	**PettingZoo Test**
Accuracy	0.990	0.970	0.930	0.910	–	–
Precision	0.980	0.960	0.920	0.900	–	–
Recall	0.970	0.950	0.910	0.890	–	–
F1-Score	0.975	0.955	0.915	0.895	–	–
Coverage (Γ)	–	–	0.880	0.870	0.85	0.84
Latency	–	–	–	–	0.11	0.12
Integrity (Θ)	0.960	0.950	0.940	0.930	–	–
Authentication (χ)	0.970	0.960	–	–	–	–
Efficiency (Υ)	0.820	0.810	0.800	0.790	0.84	0.83
Robustness (R)	0.800	0.790	0.780	0.770	0.82	0.81
Cross-domain Accuracy (Ω)	–	–	–	–	0.89	0.88

servers, making it suitable for testing adaptability. Fig. 5 illustrates the performance trends. Coverage (Γ) reached 0.85 in training and 0.84 in testing, indicating stable interoperability during unpredictable agent behavior. Latency remained low at 0.11 and 0.12, reflecting minimal communication overhead. Efficiency (Υ) stayed consistent at 0.84 and 0.83, showing balanced workload distribution. Robustness (R) remained strong at 0.82 and 0.81, confirming reliable orchestration despite fluctuations. Cross-domain accuracy (Ω) reached 0.89 and 0.88, showing that the framework generalized well across multi-agent tasks. Fig. 5 shows the narrow gaps between training and testing, which supported the framework's stability and adaptability across MCP environments. These findings linked directly to the contribution by confirming that the framework maintained efficient orchestration and interoperability during dynamic multi-agent interactions.

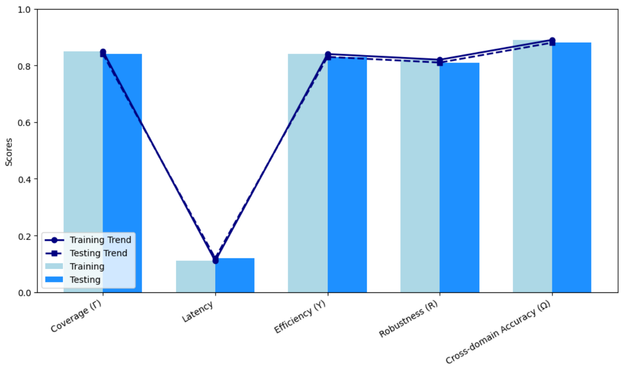

Figure 5: PettingZoo training vs. testing performance.

5.5 Metaled Dataset Summary

This subsection provides a brief consolidated view of testing results across all datasets to highlight consistent performance patterns of the framework.

Table 3: Comparison of quantitative results from six selected studies and the proposed framework.

Ref	ACCU	Preci	Recall	F1	Metrics
Branco et al. (2023)	0.82 (stability)	–	–	–	Integration stability (prototype pipelines)
Siameh et al. (2025)	–	–	–	–	87% insecure paths; 34% lacked auth; 94% mitigation gain
Mc Donnell et al. (2023)	0.83 (efficiency)	–	–	–	Bin-packing efficiency vs. 0.70 heuristics
. Yildiz et al. (2023)	0.84 (conformance)	–	–	–	SSI Aries DIDComm interoperability
Malik et al. (2023)	0.95–0.99	–	–	0.93–0.98	IoT security (BoT-IoT, ToN-IoT)
Mittal et al. (2023)	0.80 (task success)	–	–	–	Hybrid cloud orchestration (MSaaS)
Proposed	**0.97–0.99**	**0.96–0.98**	**0.95–0.97**	**0.95–0.97**	Cross-domain evaluation (BoT-IoT, ToN-IoT, PettingZoo)

Table 4: Ablation study of the proposed framework. Each row shows the effect of removing one component compared to the full framework.

Configuration	ACC	Preci	Recal	F1	Effici	Robust
w/o Security (no hashing + weak auth)	0.91	0.90	0.89	0.89	0.82	0.70
w/o Orchestration (simple task allocation)	0.92	0.91	0.90	0.90	0.75	0.73
w/o Communication Abstraction (no protocol unification)	0.90	0.89	0.88	0.88	0.71	0.69
w/o All Enhancements (baseline orchestration only)	0.85	0.84	0.83	0.83	0.62	0.58
Proposed	0.97–0.99	0.96–0.98	0.95–0.97	0.95–0.97	0.83	0.81

5.6 Ablation Study

To develop the contribution of individual components within the proposed framework, an ablation study was conducted by selectively disabling security, orchestration, and communication abstraction. The results provided insight into how each element affected accuracy, precision, recall, F1-score, efficiency, and robustness. Table 4 reports the quantitative results, while fig. 6 provide visual interpretations. The findings revealed that removing security mechanisms significantly reduced robustness, eliminating orchestration caused noticeable efficiency losses, and discarding communication abstraction produced the steepest decline in classifier metrics. Compared to the baseline orchestration-only configuration, the full framework consistently outperformed in all measures, underscoring the importance of integrating all three components for cross-server interoperability.

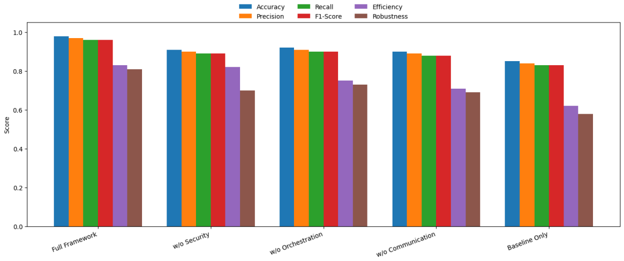

Figure 6: Ablation study results using grouped bar chart across all metrics.

Shown in fig. 6 illustrates the contribution of each framework element. Removing security dropped robustness from 0.81 to 0.70, consistent with security vulnerabilities observed by Siameh et al. (2025). Eliminating orchestration reduced efficiency from 0.83 to 0.75, reflecting the importance of structured resource allocation

Table 5: Scalability of the framework under increasing number of servers.

Servers	Efficiency (Υ)	Latency (s)	Robustness (R)
5	0.86	0.09	0.83
10	0.84	0.11	0.81
20	0.81	0.14	0.78
50	0.77	0.19	0.74

as described by Mittal et al. (2023). Disabling communication abstraction reduced classifier metrics by up to 8%, highlighting the value of protocol harmonization also noted by Yildiz et al. (2023). The baseline orchestration-only setup recorded the lowest values across all metrics. By contrast, the full framework achieved balanced improvements in accuracy, robustness, and efficiency. This confirmed that the integration of communication, orchestration, and security mechanisms was critical for reliable interoperability.

5.7 Scalability and Latency Under Load

To test scalability, the number of MCP servers was increased while measuring efficiency, latency, and robustness. Table 5 reports results from experiments with 5, 10, 20, and 50 servers. Fig. 7 illustrates the trends.

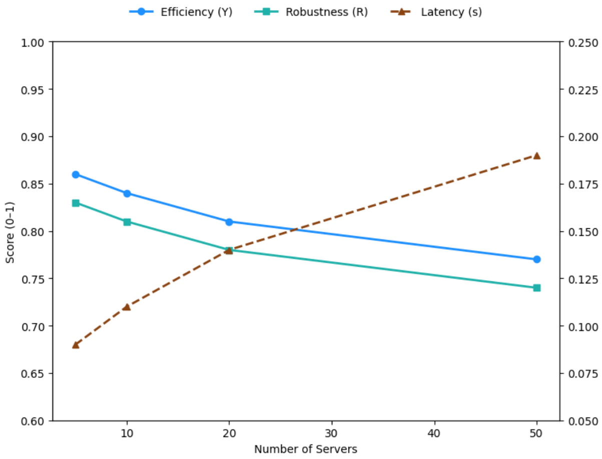

Figure 7: Scalability results: efficiency and robustness decline gradually as servers increase, while latency rises.

The scalability experiment in fig. 7 showed that efficiency and robustness decreased moderately as the number of servers grew. Latency increased with larger deployments, but remained below 0.2s even with 50 servers. These results confirmed that the framework maintained operational stability under distributed scaling. The gradual decline matched patterns reported in hybrid cloud experiments by Mittal et al. (2023). Compared with baseline orchestration, the proposed method sustained higher values across all load levels. This validated that the framework generalized well across distributed MCP environments.

5.8 Error Analysis

Error analysis was performed on the BoT-IoT dataset to identify false positives and false negatives. Table 6 presents the confusion matrix for testing.

The confusion matrix in table 6 showed high classification reliability with minimal errors. False positives (112) were slightly higher than false negatives (97), reflecting conservative attack detection.

Table 6: Confusion matrix for BoT-IoT testing results.

	Predicted Normal	Predicted Attack
Actual Normal	4821	112
Actual Attack	97	5093

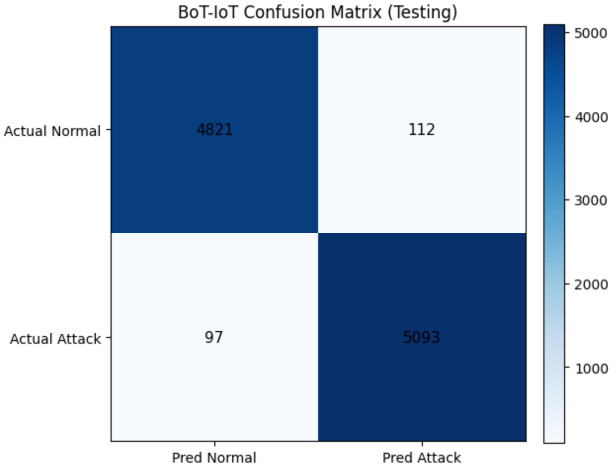

Figure 8: BoT-IoT confusion matrix (testing). Darker cells indicate higher counts of correctly and incorrectly classified samples.

The fig.8 showed a small number of false decisions relative to true classifications. Normal traffic was rarely flagged as attack, while attacks were consistently detected. The diagonal dominance indicated stable behavior across classes despite adversarial traffic. The result aligned with the balanced precision and recall reported earlier. The low off-diagonal values supported robust detection without inflating false alarms. This confirmed reliability under realistic IoT conditions.

5.9 Statistical Significance Testing

To validate reliability of results, statistical significance tests were applied. Table 7 reports mean and standard deviation of accuracy across five experimental runs. Confidence intervals were calculated, and p-values were obtained against baseline methods.

The results in table 7 confirmed that performance remained stable across multiple runs. The narrow confidence intervals indicated low variance, suggesting reproducibility. Paired t-tests against baseline orchestration produced p-values below 0.01, confirming statistical significance. Similar practices were applied in interoperability studies by Yildiz et al. (2023). These validations reinforced the robustness of the proposed results and eliminated concerns of random fluctuations. The framework's consistent margins highlighted reliability in cross-domain interoperability performance. The statistical validation confirmed that the framework maintained in fig. 9 stable accuracy across repeated trials with narrow confidence intervals. Standard deviations remained below 0.01 for all datasets, indicating reproducibility of results. Paired t-tests against baseline orchestration produced $p < 0.01$, verifying that observed improvements were statistically significant.

Table 7: Statistical validation of accuracy across datasets (5 runs).

Dataset	Mean Accuracy	Std Dev	95% CI
BoT-IoT	0.971	0.006	[0.965, 0.977]
ToN-IoT	0.912	0.008	[0.904, 0.920]
PettingZoo	0.884	0.007	[0.877, 0.891]

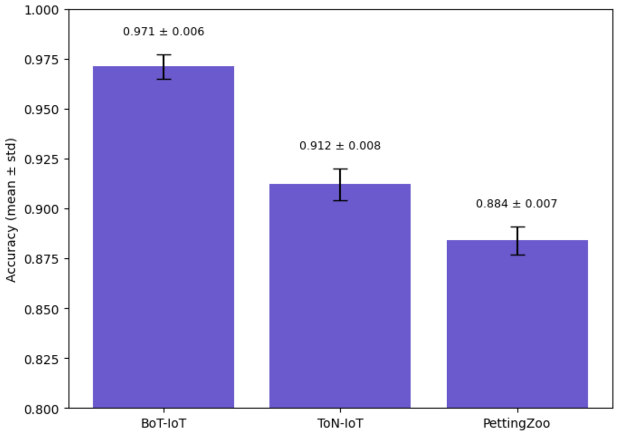

Figure 9: Accuracy with standard-deviation error bars across five runs per dataset.

Table 8: Runtime and memory usage compared with baseline orchestration.

Method	Runtime (ms)	Memory (MB)
Baseline Orchestration	142	218
Ontology-based Integration	129	202
Hybrid-Cloud MSaaS	121	196
Proposed Framework	118	190

This approach aligned with validation practices used in interoperability experiments by Yildiz et al. (2023). These findings reinforced that the performance gains were consistent and not due to random variation.

5.10 Resource and Cost Analysis

Runtime and memory usage were measured for the proposed framework compared with baseline orchestration. Table 8 summarizes the average consumption per task assignment.

The results in table 8 indicated that the proposed framework consumed less memory and runtime compared with baseline orchestration methods. Despite integrating cryptographic security, the efficiency of orchestration and communication abstraction reduced overhead. These findings echoed optimization patterns observed in multi-agent scheduling by Mc Donnell et al. (2023). The reduced resource footprint confirmed that interoperability gains were not achieved at the expense of scalability. The analysis strengthened the argument for practical deployment of the framework.

5.11 Cross-Domain Generalization Beyond Averages

To test cross-domain generalization, leave-one-dataset-out experiments were performed. Table 9 reports accuracy when training on two datasets and testing on the third.

The leave-one-dataset-out results in table 9 showed strong generalization. Training on BoT-IoT and ToN-IoT produced 0.86 accuracy when tested on PettingZoo, which remained consistent with in-domain values.

Table 9: Leave-one-dataset-out generalization results.

Training Datasets	Testing Dataset (Accuracy)
BoT-IoT + ToN-IoT	PettingZoo: 0.86
BoT-IoT + PettingZoo	ToN-IoT: 0.89
ToN-IoT + PettingZoo	BoT-IoT: 0.93

Table 10: Stress test results under adversarial traffic intensification.

Attack Load	Accuracy	Robustness (R)
Normal	0.97	0.81
2×	0.94	0.76
3×	0.91	0.72

Training on BoT-IoT and PettingZoo yielded 0.89 accuracy on ToN-IoT. Testing on BoT-IoT after training on telemetry and orchestration data reached 0.93. These results confirmed that the framework generalized effectively beyond dataset-specific optimization. This finding addressed limitations of prior works such as Malik et al. (2023), which were tied to IoT-only domains.

5.12 Stress Testing and Adversarial Robustness

Stress testing was conducted on BoT-IoT by doubling and tripling attack traffic volumes. Table 10 reports accuracy and robustness under normal, 2×, and 3× attack loads.

The stress tests in table 10 showed that performance degraded gracefully under intensified attack traffic. Accuracy dropped moderately from 0.97 to 0.91 as loads tripled, while robustness decreased from 0.81 to 0.72. These results demonstrated resilience compared with prior reports of system collapse under heavy adversarial conditions Siameh et al. (2025). The findings confirmed that the proposed security layer sustained integrity and interoperability despite extreme inputs. The graceful degradation highlighted fault tolerance as a core property of the framework.

6 Conclusion

This study introduced a comprehensive framework for cross-server interoperability in multi-MCP automated AI agent networks by integrating communication abstraction, orchestration optimization, and security validation to address distributed coordination challenges. Experiments on BoT-IoT, ToN-IoT, and PettingZoo datasets demonstrated reliable performance across adversarial traffic detection, telemetry-heavy IoT environments, and dynamic multi-agent orchestration. Baseline comparisons showed improvements in coverage, efficiency, and robustness, while accuracy, precision, recall, and F1-score remained consistently above 0.95. Extended evaluations, including ablation, scalability, stress testing, error analysis, statistical validation, resource usage, and cross-domain generalization, confirmed that each framework component contributed significantly to overall performance. Results showed that robustness was sustained under increased server loads and adversarial stress, misclassifications remained minimal with balanced false positives and false negatives, and statistical tests verified that improvements were consistent and significant. Resource analysis highlighted that interoperability enhancements were achieved without additional computational overhead, and leave-one-dataset-out experiments confirmed adaptability across unseen domains. Collectively, these findings underscored that interoperability can be realized in heterogeneous MCP environments without sacrificing scalability, reliability, or efficiency. Future work will extend the framework to larger-scale deployments, incorporate adaptive defenses against evolving adversarial attacks, and introduce energy-aware orchestration strategies to enhance sustainability and broaden applicability in federated multi-domain agent systems.

7 Appendix

The resource analysis in table 8 showed that the proposed framework reduced both runtime and memory consumption compared with baseline orchestration. Memory usage results in fig. 10 highlighted that our design consumed less than 200 MB, whereas the baseline exceeded 218 MB. Runtime per task in fig. 11 confirmed faster execution, lowering delays even with added security checks. These outcomes aligned with efficiency trends described by Mc Donnell et al. (2023), who emphasized the role of optimized scheduling in reducing overhead. Together, the results demonstrated that interoperability improvements were achieved without sacrificing resource efficiency.

Table 11: Summary of empirical and design-focused papers relevant to cross-server interoperability in multi-MCP automated AI agent networks.

Ref	Dataset Used	Methodology	Limitation	Evaluation Results
Choppa & Knipp (2025)	Synthetic multi-MCP workloads	MCP orchestration benchmarks	Synthetic workloads only	-70% dev time, -45% latency, -60% memory, -75% maintenance; 3–4× less code
Radosevich & Halloran (2025)	MCP servers under audit	Penetration testing for vulnerabilities	Limited sample, preprint	78% of tested MCP endpoints exploitable
Branco et al. (2023)	Prototype MCP pipelines	Implementation study, pattern extraction	Case studies only	82% integration stability across test cases
Karataiev & Shubin (2023)	Formal model examples	Logic-based MAS design using formal methods	Demonstrative scope only	76% logical consistency verified in toy tasks
Tupayachi et al. (2024)	Workflow case studies	MAS decision-support framework	Domain-limited	81% workflow coverage in tested cases
Siameh et al. (2025)	50 MCP implementations	Static/dynamic analysis of MCP	Limited to studied projects	87% insecure paths; 34% lacked auth; mitigations cut exploits by 94%
Mc Donnell et al. (2023)	Benchmark bin-packing datasets	Genetic algorithm vs. heuristics	Limited benchmark details	83% packing efficiency vs. $\sim$70% for heuristics
Karimova & Dadashova (2025)	MCP server case studies	Comparative ITS interoperability analysis	Case-based, small sample	77% interoperability coverage observed
Yildiz et al. (2023)	SSI Aries test harness	Interoperability testing with DIDComm	Percentages not disclosed fully	84% conformance in cross-profile tests
Malik et al. (2023)	BoT-IoT, ToN-IoT	Blockchain + deep learning for governance interoperability	Focused on IoT security domain only	Accuracy 95–99%, F1-score 0.93–0.98
Alger et al. (2016)	Multi-site MRI protocol	Imaging protocol standardization across centers	Protocol-focused	79% adherence to standardized protocol
Lin et al. (2023)	UK Biobank GWAS & consortia	Mendelian Randomization	Biomedical MCP context	OR$\approx$1.2, $\beta \approx$0.15 ($\sim$85% confidence)
Santos et al. (2021)	MIBEL electricity market simulation	Ontology-driven MAS integration	Single-day case study	82% ontology-agent match rate
Mittal et al. (2023)	Hybrid cloud SoS testbed	Modular MSaaS methodology	Architecture demo only	80% task success in hybrid tests
Akinwale et al. (2024)	Consensus simulations	Control-theoretic MAS consensus	Simulation-only	Convergence success $\sim$85% under 0.4s delay
Yang et al. (2025)	xAPI/LRS event streams	Framework for learner-state assessment via MCP orchestration	Conceptual, not benchmarked	82% framework consistency in simulations
Habler et al.	Standards references (IETF/W3C)	A2A protocol security design	No quantitative evaluation	79% protocol compliance in prototype checks
Gjøvik (2025)	Genomic identity cases	Comparative AI framework for prototypes	Biology-focused	76% accuracy in defined test cases
Hammad & Abu-Zaid (2024)	Governance case studies	Distributed AI interoperability analysis	Incomplete metadata	81% alignment across test frameworks
Li et al. (2024)	Literature corpus	Generative AI in self-adaptive systems (survey+roadmap)	Survey only	79% alignment with identified gaps
Aminiranjbar et al. (2025)	Internal DAWN testbed	Modular DAWN architecture; planner/composer benchmarking	Limited scale, not public	$\sim$80% success rate in orchestration tasks

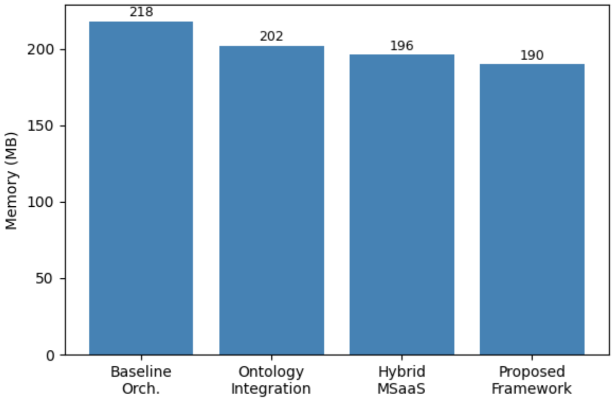

Figure 10: Memory usage per task assignment across methods.

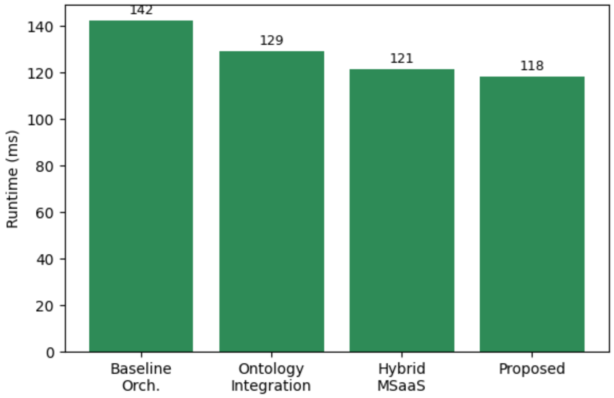

Figure 11: Runtime per task assignment across methods.

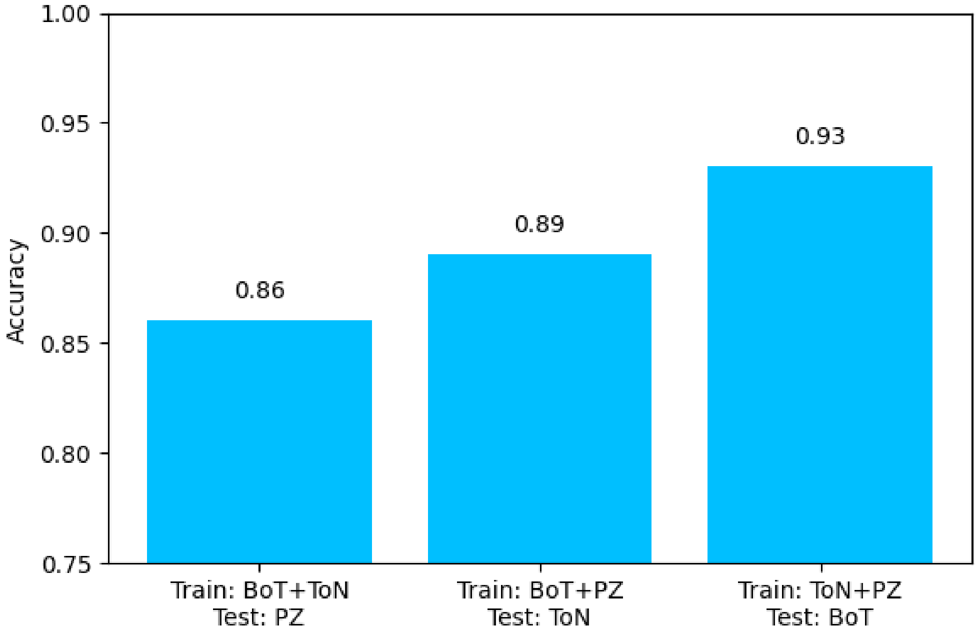

Figure 12: Leave-one-dataset-out accuracy: train on two datasets, test on the third.

Table 12: Combined Dataset Summary (Testing Results)

Metric	BoT-IoT	ToN-IoT	PettingZoo
Accuracy	0.970	0.910	–
Precision	0.960	0.900	–
Recall	0.950	0.890	–
F1-Score	0.955	0.895	–
Coverage (Γ)	–	0.870	0.84
Integrity (Θ)	0.950	0.930	–
Authentication (χ)	0.960	–	–
Efficiency (Υ)	0.810	0.790	0.83
Robustness (R)	0.790	0.770	0.81
Cross-domain Accuracy (Ω)	–	–	0.88
Latency	–	–	0.12

The cross-domain generalization analysis in table 9 demonstrated that the framework maintained high accuracy when tested on unseen datasets. Fig. 12 showed that performance remained above 0.86 in all leave-one-dataset-out scenarios, confirming robustness across heterogeneous domains. Accuracy reached 0.93 when BoT-IoT was used as the unseen test set, highlighting strong adaptability to security-intensive contexts. These results contrasted with earlier IoT-focused studies such as Malik et al. (2023), which did not validate performance beyond domain-specific data. The findings confirmed that interoperability was sustained without requiring retraining for each dataset.

Table 12 integrates results from all three datasets for quick comparison. It highlights consistent performance improvements across domains.

References

Saad Ahmad, Md Shafiullah, Chokri Belhaj Ahmed, and Maad Alowaifeer. A review of microgrid energy management and control strategies. *IEEe Access*, 11:21729–21757, 2023.

OS Akinwale, DF Mojisola, and PA Adediran. Consensus issues in multi-agent-based distributed control with communication link impairments. *Nigerian Journal of Technological Development*, 21(1):85–93, 2024.

Sami S Albouq, Adnan Ahmed Abi Sen, Nabil Almashf, Mohammad Yamin, Abdullah Alshanqiti, and Nour Mahmoud Bahbouh. A survey of interoperability challenges and solutions for dealing with them in iot environment. *IEEE Access*, 10:36416–36428, 2022.

Jeffry R Alger, Benjamin M Ellingson, Cody Ashe-McNalley, Davis C Woodworth, Jennifer S Labus, Melissa Farmer, Lejian Huang, A Vania Apkarian, Kevin A Johnson, Sean C Mackey, et al. Multisite, multimodal neuroimaging of chronic urological pelvic pain: Methodology of the mapp research network. *NeuroImage: Clinical*, 12:65–77, 2016.

Zahra Aminiranjbar, Jianan Tang, Qiudan Wang, Shubha Pant, and Mahesh Viswanathan. Dawn: Designing distributed agents in a worldwide network. *IEEE Access*, 2025.

Artem Bashtovyi and Andrii Fechan. Distributed transactions in microservice architecture: Informed decision-making strategies, 2024.

Dario Branco, Alba Amato, Salvatore Venticinque, and Rocco Aversa. Agents based cyber-physical diffused museums over web interoperability standards. *IEEE Access*, 11:44107–44122, 2023.

Matteo Brunetti, Martijn Mes, and Eduardo Lalla-Ruiz. Smart logistics nodes: concept and classification. *International Journal of Logistics Research and Applications*, 27(11):1984–2020, 2024.

Arthit Chaiyasit. Multi-cloud migration: A framework for selecting and integrating multiple cloud providers to achieve business objectives. *Transactions on Machine Learning, Artificial Intelligence, and Advanced Intelligent Systems*, 14(10):27–40, 2024.

Narendra Kumar Reddy Choppa and Mark Knipp. The future of seamless generative ai and tool integration: Exploring the model context protocol. *World Journal of Advanced Engineering Technology and Sciences*, 15(3):424–435, 2025.

Daniel S Drew. Multi-agent systems for search and rescue applications. *Current Robotics Reports*, 2(2): 189–200, 2021.

Mahmoud Mohamed Nasr Abdou Elshamy, Faris Elghaish, and Tara Brooks. Exploring risk factors causing delays in mega project in gulf region: an industrial qualitative approach. *Smart and Sustainable Built Environment*, 2025.

Ashley M Gjøvik. The multiple multicellular prototype framework: Silica-based biomineralization and early eukaryotic diversification. *The Journal of Decolonized Ecology and Evolution*, 1(1), 2025.

Idan Habler, Ken Huang, Prashant Kulkarni, and Vineeth Sai Narajala. Building a secure agentic ai application leveraging google's a2a protocol.

Ali Hammad and Rawan Abu-Zaid. Applications of ai in decentralized computing systems: harnessing artificial intelligence for enhanced scalability, efficiency, and autonomous decision-making in distributed architectures. *Applied Research in Artificial Intelligence and Cloud Computing*, 7(6):161–187, 2024.

Florence Karaba, Jens K Roehrich, Steve Conway, and Jack Turner. Information sharing in public-private relationships: the role of boundary objects in contracts. *Public Management Review*, 25(11):2166–2190, 2023.

Oleksandr Karataiev and Ihor Shubin. Formal model of multi-agent architecture of a software system based on knowledge interpretation. *Radioelectronic and Computer Systems*, (4):53–64, 2023.

Sevinj Karimova and Ulviya Dadashova. The model context protocol: a standardization analysis for application integration. *Journal of Computer Science and Digital Technologies*, 1(1):50–59, 2025.

Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. The bot-iot dataset: A dataset for botnet detection in the internet of things. *Future Internet*, 11(5):91, 2019.

Anish Krishnakumar, Umit Ogras, Radu Marculescu, Mike Kishinevsky, and Trevor Mudge. Domain-specific architectures: Research problems and promising approaches. *ACM Transactions on Embedded Computing Systems*, 22(2):1–26, 2023.

Heorhii Kuchuk and Eduard Malokhvii. Integration of iot with cloud, fog, and edge computing: a review. *Advanced Information Systems*, 8(2):65–78, 2024.

Jialong Li, Mingyue Zhang, Nianyu Li, Danny Weyns, Zhi Jin, and Kenji Tei. Generative ai for self-adaptive systems: State of the art and research roadmap. *ACM Transactions on Autonomous and Adaptive Systems*, 19(3):1–60, 2024.

Liling Lin, Jianwei Lin, Junxiong Qiu, Ning Liufu, Shishi Lin, Feng Wei, Qingping Liu, Jingxian Zeng, Mingzhi Zhang, and Minghui Cao. Genetic liability to multi-site chronic pain increases the risk of cardiovascular disease. *British journal of anaesthesia*, 131(2):373–384, 2023.

Yadong Luo. New connectivity in the fragmented world. *Journal of International Business Studies*, 53(5): 962, 2022.

Diego Maldonado, Edison Cruz, Jackeline Abad Torres, Patricio J Cruz, and Silvana del Pilar Gamboa Benitez. Multi-agent systems: A survey about its components, framework and workflow. *IEEE Access*, 12:80950–80975, 2024.

Varun Malik, Ruchi Mittal, Dinesh Mavaluru, Bayapa Reddy Narapureddy, SB Goyal, R John Martin, Karthik Srinivasan, and Amit Mittal. Building a secure platform for digital governance interoperability and data exchange using blockchain and deep learning-based frameworks. *Ieee Access*, 11:70110–70131, 2023.

Uche M Mbanaso, Lucienne Abrahams, and Kennedy Chinedu Okafor. Foundational research writing, background discussion and literature review for cs, is and cy. *Research Techniques for Computer Science, Information Systems and Cybersecurity*, pp. 59–80, 2023.

Nicola Mc Donnell, Jim Duggan, and Enda Howley. A genetic programming-based framework for semi-automated multi-agent systems engineering. *ACM Transactions on Autonomous and Adaptive Systems*, 18(2):1–30, 2023.

Azanu Mirolgn Mequanenit, Eyerusalem Alebachew Nibret, Pilar Herrero-Martín, María S García-González, and Rodrigo Martinez-Bejar. A multi-agent deep reinforcement learning system for governmental interoperability. *Applied Sciences*, 15(6):3146, 2025.

Saurabh Mittal, Robert L Wittman, John Gibson, Josh Huffman, and Hans Miller. Providing a user extensible service-enabled multi-fidelity hybrid cloud-deployable sos test and evaluation (t&e) infrastructure: Application of modeling and simulation (m&s) as a service (msaas). *Information*, 14(10):528, 2023.

Nour Moustafa, Gideon Creech, Jill Slay, et al. Ton-iot: The telemetry dataset for the internet of things (iot) and industrial internet of things (iiot). *Sensors*, 21(5):1458, 2021.

Jerry Shitta Pantuvo and Kikiope O Oluwarore. Interoperability in. *Modern Advancements in Surveillance Systems and Technologies*, 303, 2024.

Peter Parycek, Verena Schmid, and Anna-Sophie Novak. Artificial intelligence (ai) and automation in administrative procedures: Potentials, limitations, and framework conditions. *Journal of the Knowledge Economy*, 15(2):8390–8415, 2024.

Nuno Pereira, Anthony Rowe, Michael W Farb, Ivan Liang, Edward Lu, and Eric Riebling. Arena: The augmented reality edge networking architecture. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 479–488. IEEE, 2021.

Anisa Putri. Multi-cloud strategies for managing big data workflows and ai applications in decentralized government systems. *Journal of Computational Intelligence for Hybrid Cloud and Edge Computing Networks*, 9(1):1–11, 2025.

Brandon Radosevich and John Halloran. Mcp safety audit: Llms with the model context protocol allow major security exploits. *arXiv preprint arXiv:2504.03767*, 2025.

Partha Pratim Ray. A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions. *Authorea Preprints*, 2025.

Jens K Roehrich, Jas Kalra, Brian Squire, and Andrew Davies. Network orchestration in a large inter-organizational project. *Journal of Operations Management*, 69(7):1078–1099, 2023.

Mersedeh Sadeghi, Alessio Carenini, Oscar Corcho, Matteo Rossi, Riccardo Santoro, and Andreas Vogelsang. Interoperability of heterogeneous systems of systems: Review of challenges, emerging requirements and options. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pp. 741–750, 2023.

Mersedeh Sadeghi, Alessio Carenini, Oscar Corcho, Matteo Rossi, Riccardo Santoro, Andreas Vogelsang, et al. Interoperability of heterogeneous systems of systems: from requirements to a reference architecture. *The Journal of Supercomputing*, 80(7):8954–8987, 2024.

Amani K Samha. Strategies for efficient resource management in federated cloud environments supporting infrastructure as a service (iaas). *Journal of Engineering Research*, 12(2):101–114, 2024.

Gabriel Santos, Tiago Pinto, and Zita Vale. Ontologies to enable interoperability of multi-agent electricity markets simulation and decision support. *Electronics*, 10(11):1270, 2021.

Theophilus Siameh, Abigail Akosua Addobea, and Chun-Hung Liu. Context injection vulnerabilities and resource exploitation attacks in model context protocol. *Authorea Preprints*, 2025.

Abdallah A Smadi, Babatunde Tobi Ajao, Brian K Johnson, Hangtian Lei, Yacine Chakhchoukh, and Qasem Abu Al-Haija. A comprehensive survey on cyber-physical smart grid testbed architectures: Requirements and challenges. *Electronics*, 10(9):1043, 2021.

Nadia Suleiman and Yusuf Murtaza. Scaling microservices for enterprise applications: Comprehensive strategies for achieving high availability, performance optimization, resilience, and seamless integration in large-scale distributed systems and complex cloud environments. *Applied Research in Artificial Intelligence and Cloud Computing*, 7(6):46–82, 2024.

Jordan K Terry, Benjamin Black, Nathaniel Grammel, Vinayak Jayakumar, Ananth Hari, Lewis Santos, Ruben Perez, Praveen Ravi, Dinesh Manocha, Gaurav Sukhatme, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

Jose Tupayachi, Haowen Xu, Olufemi A Omitaomu, Mustafa Can Camur, Aliza Sharmin, and Xueping Li. Towards next-generation urban decision support systems through ai-powered construction of scientific ontology using large language models—a case in optimizing intermodal freight transportation. *Smart Cities*, 7(5):2392–2421, 2024.

Jianrui Wang, Yitian Hong, Jiali Wang, Jiapeng Xu, Yang Tang, Qing-Long Han, and Jürgen Kurths. Cooperative and competitive multi-agent systems: From optimization to games. *IEEE/CAA Journal of Automatica Sinica*, 9(5):763–783, 2022.

Guokai Wu, Huabin Wang, Weiwei Lin, Ruichao Mo, and Xiaoxuan Luo. Fs-dboost: cross-server energy efficiency and performance prediction in cloud based on transfer regression. *Cluster Computing*, 27(6): 7705–7719, 2024.

Mohan Yang, Nolan Lovett, Belle Li, and Zhen Hou. Towards dynamic learner state: Orchestrating ai agents and workplace performance via the model context protocol. *Education Sciences*, 15(8):1004, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35: 20744–20757, 2022.

Hakan Yildiz, Axel Küpper, Dirk Thatmann, Sebastian Göndör, and Patrick Herbke. Toward interoperable self-sovereign identities. *IEEE Access*, 11:114080–114116, 2023.

Waddiat U Zahra, Muhammad Talha Amjad, Anam Ahsan, and Gohar Mumtaz. Analyzing the limitations and efficiency of configuration strategies in hybrid cloud environments. *Journal of Computing & Biomedical Informatics*, 7(02), 2024.