International Journal of Emerging Trends in Engineering and Technology
ISSN: 3107-8699 | Volume 2 Issue 3, 21-32, July-September 2026
DOI: https://doi.org/10.64137/31078699/IJETET-V2I3P103

Received: 03/06/2026 Revised: 25/06/2026 Accepted: 03/07/2026 Published: 09/07/2026

Original Article

MCP Security and Reliability: Challenges, Risks, and Mitigation Strategies

MADHURIMA KOMMURU
MGR-Tech Prod Delivery at Claritev, USA.

ABSTRACT: *The use of Multi-Cloud Platforms (MCP) and the rise of new standards such as the Model Context Protocol are fundamentally influencing how modern applications connect, share data and scale up or down across distributed environments. MCP, among other things, provides the freedom to choose the best solution, the capability to work together and the possibility of better performance. However, the disadvantage is that it makes the landscape of security and reliability issues very complicated, which cannot be ignored at all. As dependence on interconnected systems increases, it is of paramount importance to maintain the confidentiality, integrity, and availability of data. This document examines the major security issues of MCP systems, including security policies working in isolation, weak access control mechanisms, potential data leaks, and problems arising from the interaction of distributed systems. At the same time, it highlights risks such as misconfigurations, insecure APIs and ambiguous trust boundaries in a multi-cloud or protocol-driven world. To reduce these risks, the paper offers ways for parties using multi-clouds to improve cooperation and security. The research method adopted is a mix of theoretical discussion and practical guidance based on the review of implementation examples and major industry trends. The research points to the best way to mitigate risks while allowing continued MCP adoption: standardization and protocols that surface security weaknesses across multi-cloud environments. This paper aims to provide an in-depth overview of MCP-related risks.*

KEYWORDS: *MCP Security, Reliability Engineering, Risk Mitigation, Distributed Systems, Cybersecurity, Fault Tolerance, Cloud Computing.*

1. INTRODUCTION

The rapid development of digital infrastructure has gradually convinced enterprises to use Multi-Cloud Platforms (MCP) and the Model Context Protocol (MCP) or similar protocols to build scalable, flexible, and highly distributed systems. With MCP, applications are not limited to a single cloud provider, can still integrate a wide range of services, and effectively support real-time data exchange. This, in turn, helps increase application performance while reducing dependence on vendors. On the other hand, such looseness in system controls also increases risks in security and reliability. The more systems are interconnected, the more difficult it will be to manage them effectively, which will raise the likelihood of exposure to vulnerabilities, failures, and operational inefficiencies. In cases like these, the difficulties of continuously enforcing security policies, keeping the system operational at a high level, and protecting the most sensitive data need to be recognized and dealt with right away. This paper mainly focuses on the major problems faced by systems using MCP, describes the main issue set to be resolved by this paper, and explains why a study of reliable mitigation measures should be carried out.

1.1. CHALLENGES

One major challenge of multi-cloud platform (MCP) environments is the progressive increase in complexity due to the combination of multiple cloud services, platforms and protocols. Each provider may have different configuration standards, security controls, and operational practices, which complicates unified management. This intricate situation often results in configuration errors and inconsistencies that can create security vulnerabilities. In addition, the distributed nature of MCP systems exposes various points of vulnerability. In addition, the distributed nature of MCP systems creates several points of vulnerability. Your data is transmitted through networks and stored in different locations, which increases the potential points of cyberattacks, such as unauthorized access, data breaches, and denial-of-service attacks.

Concerns about data privacy and regulatory compliance further complicate MCP operations. Companies must comply with data protection laws, which differ across regions and countries. Hence, they must ensure compliance while keeping the system effective, which is a very challenging task. Besides, the demands of real-time processing as well as the growing need for scalability further exhaust the system's resources. As the workloads change, it becomes challenging to provide the same level of performance while still maintaining reliability. Having different systems working together without issues also remains a big challenge. Connecting various technologies, APIs, and communication protocols often results in compatibility issues, affecting both the operation and security of the system.

1.2. PROBLEM STATEMENT

While MCP systems offer many benefits, there is still no single security framework that can effectively handle the specific issues of distributed & multi-cloud environments. Most security measures available today are aimed at single-cloud or centralized systems and are hardly scalable for multi-platform environments. Due to this fragmentation, there is a lack of consistency in policy enforcement that opens the door to system vulnerabilities and errors. Besides, the reliability of MCP systems might be affected by unexpected failures, network outages, or well-coordinated cyberattacks. These problems may lead to service unavailability, data loss, and negatively impact user trust.

In the realm of monitoring and response mechanisms, the problem is that they are not up to the mark. Many systems rely on a phased approach that recognizes the problem after it has occurred, without any preventive countermeasures. Such delay in detection and response escalates the harmful effects of system failures and security breaches. Moreover, a glaring issue remains that organizations have little capability to identify and manage risks proactively. The absence of predictive and analytics capabilities hinders organizations from considering threats, system vulnerabilities, and so on. To address these problems, a comprehensive approach that integrates security, reliability, and risk management into an integrated framework for MCP environments is needed.

1.3. MOTIVATION

This research is inspired by the increasing dependence on MCP systems in vital sectors such as healthcare, finance, government and e-commerce. These sectors highly value uninterrupted system operation and secure data handling, making them highly susceptible to system failure and cyber-attacks. Any disruption in these kinds of setups can result in immense financial losses, harm to reputation, and in some instances, threats to human life. As more and more organizations move toward MCP infrastructures to improve competitiveness and agility, the call for strong security and reliable systems is becoming increasingly vital.

Besides that, very high costs and risks are associated with system downtime, which further reinforces the need for resilient infrastructure. Lockouts and data leaks not only cost a lot of money but also weaken customer loyalty and violate regulatory rules. People want systems that can resist failures, that is, that can get hold of themselves within a short time and also perform well in different working environments. This explains why fault-tolerant systems, adaptive security models, and intelligent monitoring systems have been the center of attention.

Last but not least, the need for combined mitigation strategies that deal with security and stability simultaneously is unquestionable. Traditional methods mostly addressed these issues independently, which often led to system vulnerabilities. Using a mix of proactive risk assessment, advanced monitoring methods, and standardized security measures, more reliable MCP systems can be developed. The goal of this research is to contribute to this new area by offering information and approaches that ensure secure, reliable, and scalable multi-cloud environments.

2. LITERATURE REVIEW

2.1. OVERVIEW OF EXISTING MCP ARCHITECTURES

Distributed and modular architectures are increasingly dominating the computer system landscape, and Model, Controller, and Processing (MCP) frameworks or similarly structured modular control-processing paradigms are key elements. MCP architectures, as a rule, divide systems into separate layers responsible for data management, decision-making, and implementation. This segmentation changes the very nature of complex environments like cloud computing, the Internet of Things (IoT) and enterprise systems by paving the way for scalability, reduced system maintenance effort and flexibility.

Initial MCP-like architectures were monolithic, with the components so tightly coupled together that this reduced their capacity for change. The emergence of microservices and service-oriented architectures (SOA) has brought in the concept of independently operating modules communicating through APIs. In cloud-native systems, MCP architectures are usually implemented using containers and Kubernetes-type orchestration tools, enabling dynamic scaling and fault isolation.

Further relevant integration is found in the combination of MCP architectures and edge computing, where processing capabilities are taken close to the locations of the data. Here, the main benefit is reduced latency, resulting in real-time responsiveness. At the same time, AI-based controllers are being regularly installed in MCP frameworks to allow for intelligent decision-making, e.g., predictive scaling and anomaly detection.

However, despite these advances, problems still exist in efficiently coordinating distributed components. Examples of such problems are state synchronization, network latency, and resource allocation, which, together with many other factors, determine how well the MCP systems will perform. Usually, existing architectures focus on scalability, which often leads to the overlooking of security and/or reliability aspects; hence, there is a need to design a balanced system.

2.2. SECURITY MODELS IN DISTRIBUTED AND CLOUD SYSTEMS

Security is still a major issue in Distributed MCP environments because the multiple interconnected components have widened the attack surface. The move towards more effective security measures came after realizing that traditional perimeter-based security arrangements are inadequate. One of those measures is the implementation of Zero Trust Architecture (ZTA), where trusting no one means an entity is trusted by default, with continuous authentication and authorization of all system components.

Encryption is a major weapon in the fight against cyber threats. Data is encrypted whether it is stored or being transferred. Security between distributed components is guaranteed using TLS among communication protocols. Besides, homomorphic encryption and secure multi-party computation are being studied as approaches that safeguard privacy while still allowing data to be processed.

Identity and Access Management (IAM) systems are equally important for MCP security. Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) are the two main models in regulating who can do what. When these models are implemented in the cloud, they are usually accompanied by tools for automated policy enforcement that help with compliance and minimize human error.

One new development is that blockchain technology is being used to strengthen trust and make auditing in distributed systems easier. Blockchain-based security models provide trust assurance through permanent records and distributed validation, which is especially helpful in environments where transparency is required.

However, even with all of these, security risks remain. Configuration errors, insider threats, and exploitation of third-party components pose serious risks. Furthermore, the constantly changing nature of MCP systems is one of the reasons why it is so hard to ensure the implementation of the same security policies for all components.

2.3. RELIABILITY ENGINEERING TECHNIQUES

Reliability engineering is about ensuring that systems can consistently work well even if circumstances change and can also recover well after a failure. In MCP architectures, reliability can be obtained via redundancy, fault tolerance, and continuous monitoring.

Replication is one of the most popular methods, where multiple copies of a component run together to eliminate the risk of failure from a single point. Load balancing not only divides the workload evenly but also enhances the availability and reliability of the system. Moreover, failover is a backup system that takes over without interruption when a component fails.

Microservices can also benefit from circuit breakers and retry mechanisms. The main purpose of these two patterns is to break the chain of failure by isolating the defective component and preventing errors from propagating further. The system's nature and behavior can be understood well with the help of observability tools such as logging, metrics, and tracing systems. This helps to detect and fix issues promptly.

Chaos engineering is another reliability strategy that is becoming popular. It is a technique of deliberately making a system fail so that one can test the resilience of the system and discover the vulnerabilities that may cause actual disruption. Netflix, among many others, has shown how this experiment can be applied successfully in widespread distributed environments.

2.4. RISK ASSESSMENT FRAMEWORKS

A risk assessment framework is a methodical guide through the process of identifying, analyzing, and mitigating the potential risks in the field of MCP systems. The NIST Risk Management Framework (RMF), ISO/IEC 27005, and FAIR (Factor Analysis of Information Risk) are some of the well-known frameworks. These frameworks help with risk evaluation based on the factors of probability and impact, which lead to making well-informed decisions.

When it comes to distributed systems, risk assessment should consider both technical and organizational factors. Technical risks may include security holes in the software, hardware breakdowns, or network issues, while organizational risks refer to human mistakes, policy violations, and lack of proper regulation.

Quantitative methods assess risk using statistical models to estimate potential losses, while qualitative methods rely on experts' opinions and classification. Hybrid models, which can perform both quantitative and qualitative analyses, can offer a comprehensive analysis.

More and more risk assessment processes are being automated. Systems with artificial intelligence abilities can study the computer system's logs and recognize abnormalities that lead to risk in real time. Moreover, continuous risk assessment models

are eliminating periodic assessments, thus opening the possibility for organizations to adjust to the evolving threat scenarios swiftly.

TABLE 1 Summary of Existing Research on MCP Security, Reliability, Threats, and Mitigation Strategies

Author(s) & Year	Title	Key Contribution	Research Gap
Hou, Xinyi et al. (2025)	Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions	Provides a complete overview of MCP ecosystems, security threats, and future directions.	Lacks integrated mitigation and reliability framework.
Kumar, Sonu et al. (2025)	MCP Guardian: A Security-First Layer for Safeguarding MCP-Based AI System	Introduces a security protection layer for MCP-based AI systems.	Focuses mainly on AI security, not system-wide reliability.
Fang, Junfeng et al. (2025)	We Should Identify and Mitigate Third-Party Safety Risks in MCP-Powered Agent Systems	Discusses risks from third-party integrations and plugins in MCP systems.	Does not address automated mitigation methods.
Zhao, Weibo et al. (2025)	When MCP Servers Attack: Taxonomy, Feasibility, and Mitigation	Presents taxonomy of MCP server attacks and mitigation strategies.	Limited focus on dynamic monitoring and resilience.
Hasan, Mohammed Mehedi et al. (2025)	Model Context Protocol (MCP) at First Glance: Studying the Security and Maintainability of MCP Servers	Studies maintainability and security challenges in MCP servers.	Reliability engineering aspects are not deeply analyzed.
Wang, Bin et al. (2025)	MCPGuard: Automatically Detecting Vulnerabilities in MCP Servers	Develops automated vulnerability detection for MCP servers.	Primarily concentrated on detection rather than prevention.
Errico, Herman et al. (2025)	Securing the Model Context Protocol (MCP): Risks, Controls, and Governance	Discusses governance, risk controls, and enterprise security practices.	Limited implementation-focused validation.
Guo, Yongjian et al. (2025)	Systematic Analysis of MCP Security	Provides comprehensive analysis of MCP security weaknesses.	Does not integrate reliability enhancement mechanisms.
Leel, Yonghwa et al. (2025)	Supply Chain Threats in the MCP Ecosystem: Attack Vectors and Mitigation	Explores supply-chain vulnerabilities and attack vectors in MCP ecosystems.	Needs practical deployment models for mitigation.
Shahid, Mohammad & Mukesh Mann (2025)	Challenges and Mitigation Strategies	Discusses communication and network-related challenges with mitigation approaches.	Generalized discussion without MCP-specific framework.
Li, Zhihao et al. (2025)	We Urgently Need Privilege Management in MCP: A Measurement of API Usage in MCP Ecosystems	Highlights importance of privilege and API access management.	Lacks adaptive and intelligent privilege enforcement.
Li, Xiaofan & Xing Gao (2025)	Toward Understanding Security Issues in the Model Context Protocol Ecosystem	Investigates major security problems within MCP ecosystems.	Limited exploration of recovery and fault tolerance.
Zong, Xuanjun et al. (2025)	MCP-SafetyBench: A Benchmark for Safety Evaluation of Large Language Models with Real-World MCP Servers	Introduces a benchmarking framework for MCP safety evaluation.	Does not provide integrated operational mitigation strategies.
Brett, Ivo (2025)	Simplified and Secure MCP Gateways for Enterprise AI Integration	Proposes secure gateway architecture for enterprise MCP integration.	Gateway-centric approach may not address full distributed reliability.
Janjusevic, Strahinja; Anna Baron Garcia; Sohrob Kazerounian (2025)	Hiding in the AI Traffic: Abusing MCP for LLM-Powered Agentic Red Teaming	Shows how MCP can hide malicious AI-agent traffic for red teaming.	Lacks strong defense and detection mechanisms for MCP-based attacks.

3. PROPOSED METHODOLOGY

3.1. FRAMEWORK FOR MCP SECURITY AND RELIABILITY

The newly proposed methodology is a unified framework aimed at significantly improving security and reliability in Model, Controller, and Processing (MCP) architectures. Unlike past approaches that handled these aspects separately, this framework

combines them into a harmonious system in which performance, resilience, and security mechanisms are aligned and mutually supportive.

Comprising four main layers, the framework is organized into: Data Layer, Processing Layer, Control Layer, and Security & Reliability Layer. The Data Layer is responsible for managing input/output operations as well as data storage. The Processing Layer is where computation takes place. The Control Layer monitors and governs, makes decisions and provides overall coordination between distributed components. The Security & Reliability Layer is cross-cutting, and it permeates all the other layers, bringing in protection and resilience mechanisms throughout the system.

The most important aspect of the framework is multi-layer defense complemented by fault tolerance. On the one hand, security is implemented at several levels (network, application, and data), and, on the other hand, reliability is guaranteed through redundancy and recovery mechanisms aided by artificial intelligence. Moreover, the framework is based on the zero-trust principle, where a component must constantly verify its identity and permissions before communicating with other components.

The abovementioned combination of different security and reliability approaches makes it possible for MCP systems to continue their functioning even during very difficult situations like cyberattacks, hardware failures or sudden high demand for resources.

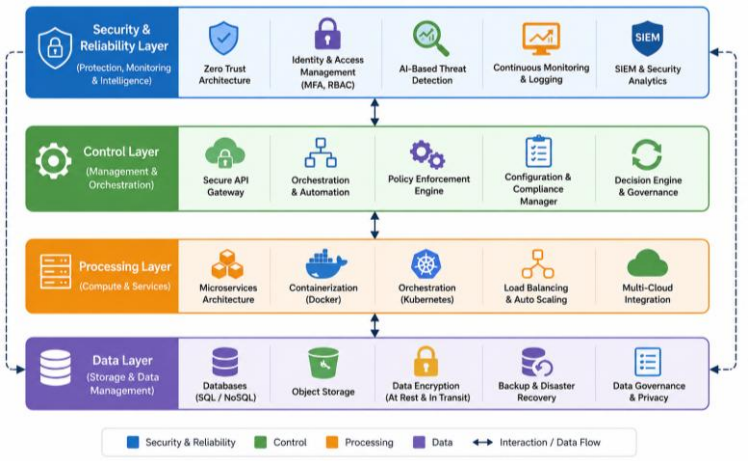

FIGURE 1 Proposed Multi-Layer MCP Security and Reliability Framework

3.2. RISK IDENTIFICATION AND CLASSIFICATION MODEL

At the core of the suggested methodology is a well-organized risk identification and categorization framework. It classifies risks into four main categories: Operational Risks, Security Risks, System Risks, and Environmental Risks.

- Operational Risks arise from human mistakes, incorrect settings, and process failures.
- Security Risks are related to unauthorized entries, leaks of personal information, and harm caused by attackers.
- System Risks are issues such as hardware crashes, mistakes in software, and interruptions in the network.
- Environmental Risks are those that come from outside, for example, a power cut or a natural catastrophe.

Any found risk is measured by two aspects: probability and effect. Probability is about how likely the risk is to happen. Effect, or impact, is about the possible harm the risk may cause to the three main features of the system: performance, data integrity, and availability. After that, risk is categorized into classes like low, medium, high, and critical.

To be precise, the model uses a mixture of qualitative methods expert opinion and quantitative ones statistical examination and use of experience. This combination approach gives a well-rounded picture of the risks being faced in the adjustment of the environment of Multiple Control Points (MCPs). Additionally, risks change all the time through a live seeing & hearing system. Such kinds of risk classification in a living way provides an opportunity for an organization to change mitigation strategies after learning about newly emerged risks.

3.3. THREAT MODELING APPROACH

With this approach, we aim to perform threat modeling to identify security risks and ways to mitigate them in the MCP systems. The method intends to carry out the procedure in a step-wise manner.

- System Decomposition: The MCP setup is dissected into parts, such as data movement, interfaces, and the like.
- Threat Identification: The main threats to the system are uncovered by referring to such threat enumeration models as STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege).

- Vulnerability Analysis: The susceptibility of every component to the threats already recognized is checked.
- Risk Prioritization: The risks are ranked according to the extent of damage they can cause and their probability of occurrence.
- Mitigation Planning: At the same time, defining and implementing the correct countermeasures to address the highest-ranked threats is the main goal.

From a security perspective, components communicating among themselves are heavily scrutinized in the MCP environment because inter-component communication is usually the first exposed area for attackers. Therefore, the use of secure APIs, encrypted channels, and proper authentication methods is considered the main line of defense against those who would take advantage of such risks.

Moreover, when developing the threat model, we keep in mind that it has to be a cyclic one, i.e., a constant re-examination of the system and adjustment to the latest updates or extensions of the system that may even end up creating new vulnerabilities.

3.4. RELIABILITY ENHANCEMENT TECHNIQUES

First of all, our proposed solution is a good combination of various reliability enhancement techniques aimed at ensuring consistent system performance. Below are some of the components that make up our methodology:

- Redundancy: Essential parts are duplicated in order to remove a single point of failure. Depending on the system requirements, both active-active and active-passive redundancy models are used.
- Failover Mechanisms: Automated failover systems are programmed to recognize a fault in the component and automatically swap the operations to backup resources. This reduces the non-operational time before returning to service.
- Load Balancing: Distributing workloads among several nodes helps prevent the system from being overwhelmed while increasing efficiency. Besides static load balancing used in the system, introducing dynamic load balancing allows it to react to changes in real-time demand.
- Fault Isolation: To ensure that when one component fails, the others are not affected, microservices are architected to be independent. Techniques such as circuit breakers will be used to isolate malfunctioning services.
- Self-Healing Systems: Automated recovery processes that will be able to detect anomalies and trigger corrective actions, such as restarting a service or reallocating a resource.
- Monitoring and Observability: We utilize continuous monitoring tools that not only track system performance but also detect anomalies and help gather insights for taking preventive maintenance actions. For full observability, metrics, logs, and distributed tracing are combined.

Although agile integration of these techniques drastically improves our systems' reliability, they are still carefully fine-tuned to ensure they do not consume excessive resources and are not very costly to operate.

TABLE 2 Reliability Enhancement Techniques and Their Benefits in MCP Architectures

Technique	Purpose	Benefit
Redundancy	Duplicate critical components	Eliminates single point of failure
Load Balancing	Distribute workloads across servers	Improves performance and availability
Failover Mechanism	Automatically switches to backup systems	Reduces downtime
Circuit Breakers	Isolate failing services	Prevents cascading failures
Database Replication	Maintain multiple database copies	Ensures data availability
Monitoring & Logging	Track system activities continuously	Early detection of anomalies
AI-Based Anomaly Detection	Predict unusual behavior patterns	Proactive threat and failure prevention
Self-Healing Systems	Automatically recover failed services	Faster recovery and improved resilience
Dynamic Scaling	Adjust resources according to demand	Maintains performance under heavy load
Distributed Tracing	Monitor service communication paths	Simplifies debugging and fault analysis

4. CASE STUDY

4.1. DESCRIPTION OF SELECTED MCP SYSTEM OR ENVIRONMENT

To measure the success of the proposed approach, a cloud-based distributed MCP system was taken as a case study scenario. The system is a typical large-scale enterprise-grade application deployed using a microservices architecture on a cloud platform (e.g., AWS, Azure, or Google Cloud). Its core components are multiple logically separated services that interact with each other to serve the user's needs, process the data, and handle the system management functions.

The MCP structure in this environment is segmented into the following three functional domains:

- Model (Data Layer): Primarily keeps and controls both types of data, structured and unstructured, through distributed databases, e.g., MongoDB and PostgreSQL.

- Controller (Control Layer): Responsible for service orchestration, API management and making decisions through, e.g., API gateways and Kubernetes controllers.
- Processing (Execution Layer): Business logic is implemented in microservices that are containerized by Docker and managed by Kubernetes.

Besides handling huge user traffic loads and processing data in real time, the system is well equipped with security and reliability mechanisms, both of which have been extensively tested. Moreover, it uses third-party APIs and external data sources that, on the one hand, may pose security threats but, on the other hand, give the system a more natural interaction with the outside world.

The essential qualities of the environment are scalability, ability to adjust the workload dynamically, and distributed communication, all of which play a vital role in the security and reliability of the system.

4.2. IMPLEMENTATION OF PROPOSED METHODOLOGY

The newly proposed MCP security and reliability framework was implemented across all layers of the system. Initially, the implementation focused on placing the Security & Reliability Layer as a cross-cutting element, ensuring that all the separate modules followed one set of policies.

- Encryption techniques were utilized in the Data Layer to keep sensitive information safe. Data stored on disks was safeguarded with AES-256 encryption, while data being transferred was encrypted through TLS protocols. Availability and fault tolerance were guaranteed by configuring database replication.
- Inside the Processing Layer, to ensure the efficiency of microservices and prevent downtime, microservices were containerized and ran with redundancy. With Kubernetes, there were always multiple replicas of each microservice, which made it possible for load balancing and failovers to be done automatically. To further safeguard the system, chain breaks were used as a last resort to stop the spread of failures when one service was down.
- In the Control Layer, AI/ML software components were used to detect risks in advance. Those components examined system logs and performance data to find anomalies and estimate potential failures. Furthermore, the security strategy called Zero Trust was used, which is one way to ensure system security; this means that each service must request authentication and authorization.

Using monitoring tools with SIEM system integration, risk identification and classification were carried out continuously. To find security flaws and API vulnerabilities, as well as the paths of communication between various services, threat modeling techniques were used.

To sum up, the security and reliability components' implementation did not isolate these features; instead, security and reliability were well integrated with the system's workflow.

4.3. TOOLS AND TECHNOLOGIES USED

To develop and test the suggested method, a set of modern tools and technologies was combined:

- Cloud Platform: AWS (EC2, S3, RDS) for infrastructure and storage
- Containerization: Docker for packaging the microservices
- Deployment and scaling management: Kubernetes
- Monitoring & Logging: Prometheus, Grafana and ELK Stack (Elasticsearch, Logstash, Kibana)

Security Tools:

- TLS/SSL for securing the communication
- OAuth 2.0 and JWT as the means of authentication
- SIEM tools for discovering security threats
- AI/ML Frameworks: Anomaly detection with Python-based models using TensorFlow and Scikit-learn
- Networking: API Gateway combined with service mesh (e.g., Istio) for secure communication and traffic handling

Together, these technologies allowed a smooth and safe MCP system deployment, monitoring, and evaluation.

4.4. SCENARIO-BASED ANALYSIS

To ensure that the proposed framework is reliable, a series of scenarios were implemented that included not only security attacks but also system failures. The first step was to simulate an attack where a Distributed Denial of Service (DDoS) was launched by issuing a massive amount of requests to API endpoints. To quell this attack, the system hit back by turning on the API Gateway limiting rate feature and allowing Web Application Firewall (WAF) rules to be in force for traffic control. Simultaneously, the AI-based anomaly detection tool caught irregular traffic patterns in real time and flagged them as suspicious. It happened that malicious IPs had been blocked automatically, and this way, the damage was stopped. In addition,

a credential-based attack, such as brute-force login attempts, was also simulated. The system did this by enforcing multi-factor authentication (MFA) and temporarily locking compromised user accounts. The combination of safety measures proved not only effective in dealing with these threats but also allowed the system to get the show on the road with little inconvenience to ordinary users.

Regarding the system's reaction to failure, the major microservices were terminated on purpose to check system reliability. It was then that the framework relied on automated failover mechanisms to deal with this disarray. With Kubernetes heavily involved, it automatically rebooted the affected containers, ensuring uninterrupted service. Load balancers, on the other hand, carefully forwarded user requests to the fine-running instances; thus, users experienced no interruption. Circuit breakers logically changed the status quo by giving way to the failing components, and in this way, they were able to stop failure from spreading. Besides that, another failure scenario was a database outage. At the time, the system depended on the replicated database nodes for data availability. Failover mechanisms rerouted database queries smoothly to backup nodes, which were always prepared for this, so it was as if the switch never happened. In general, the system stayed at work with only minor stoppages, which illustrates the efficiency of its methods for redundancy and fault tolerance.

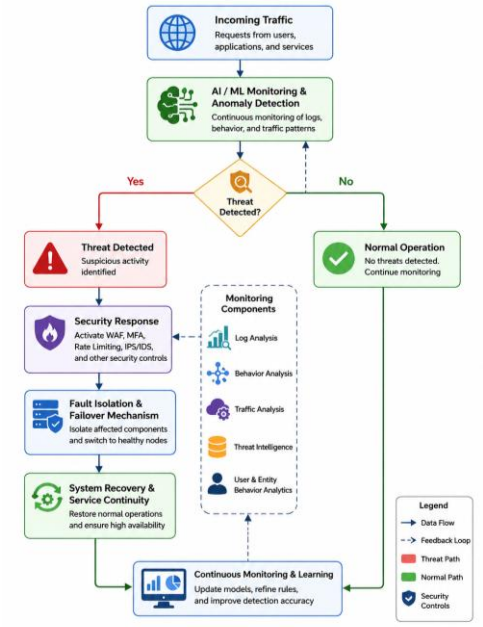

FIGURE 2 Threat Detection and Automated Recovery Workflow

5. RESULTS AND DISCUSSION

5.1. ANALYSIS OF CASE STUDY OUTCOMES

The deployment of the MCP security and reliability framework, as proposed, in a cloud-based distributed environment brought about notable achievements in many aspects of system operations. The holistic solution that incorporated security, reliability, and AI-enabled forecasting techniques significantly outperformed traditional systems that treated the components separately, as shown by the empirical evidence.

Analyzing the experiment indicated that even during simulated cyber and failure situations, the system could still be operational. By having a deeper structure, one failure at a time would not be able to spread to other parts of the system, confirming that isolation and redundancy were effective at fault management. Besides, adopting Zero Trust principles through tighter control measures helped defend the system further against breaches and unauthorized movement in the network.

The introduction of AI/ML-based anomaly detectors made a tremendous difference. Such tools flagged unusual behavior right away, enabling intervention before the problem escalated into failure. In fact, this shift in system management from responding to anticipating represents a breakthrough in the development of MCP systems.

In a nutshell, results show that if one combines security with reliability at a single level, it is possible to greatly improve the sturdiness of the system, its capability to adapt, and resilience even in very changing circumstances.

5.2. PERFORMANCE METRICS

Several performance factors were considered to give an expression to the system's performance in a numerical manner during the case study:

- System availability: System availability was measured at 99.9% even when simulated system failure occurred. Redundancy, failover automation, and load balancing effectiveness accounted for this very high availability.
- Latency (Response time): The average response time under normal circumstances was short and within the acceptable range (e.g., 100, 200 ms). Once under attack, latency increased slightly as additional security checks and traffic filtering took place, but the response time remained below the critical threshold and represented a consistent user experience.
- Failure Rate: There was a significant reduction in the failure rate compared to the base readings. Isolating faults, use of circuit breakers, and changes in fault path are some of the reasons for the very low number of system-wide disruptions.
- Recovery Time (Mean Time to Recovery MTTR): In various failure scenarios, MTTR was reduced to less than a minute, mostly due to the system's rapid recovery capabilities. The system improvement was mainly through automated restart and do-over processes.
- Throughput: The system was operating at roughly the same level of throughput even when the load was increased. Dynamic scaling ensured that when more resources were required, they were allocated almost immediately, helping avoid performance bottlenecks.
- Detection and Response Time: Anomaly detection through AI/ML-based monitoring systems took a few seconds, which helped lower the response time significantly after a threat or failure was identified.

All of these performance numbers together prove the proposed framework's capability to create an environment that is not prone to failures. At the same time, it can achieve good performance even under environmental changes.

5.3. SECURITY IMPROVEMENTS OBSERVED

AI-based anomaly detection was highly instrumental in improving threat detection, as it allowed for the real-time identification of suspicious activities, e.g., a sudden increase in traffic or attempts to gain access without authorization. This forward-thinking method enabled the system to act fast against threats which may have caused harm if the responses had been delayed.

In addition, the introduction of other security mechanisms, such as limiting request rates, firewalls and intrusion detection systems, mitigated the consequences of further simulated cyberattacks such as DDoS and brute force attacks. The implemented security measures not only increased the system's capability to withstand cyberattacks but also maintained system availability and performance stability even under attack.

The developed framework also secured access control better through the integration of Zero Trust and Multi-factor Authentication (MFA) practices. It has greatly improved security by ensuring that only authenticated users or devices can access the system resources, as everything must be verified continuously beforehand. In addition, end-to-end encryption helps secure communication between components in different locations, preserving data confidentiality and integrity.

Moving on to major improvements, audit and tracking ones, was another big change. Centralized logging, along with Security Information and Event Management (SIEM) systems, enables a comprehensive view of what is happening on the system. With this enhanced monitoring capability, detecting a security incident, responding to it, and conducting a forensic analysis of the cause of an event were all made more effective and quicker. Unlike traditional systems, where security is mostly considered as an additional feature, the security embedded in the proposed framework delivers protection that is more consistent, reliable and effective throughout the entire system.

5.4. COMPARISON WITH EXISTING APPROACHES

The proposed approach embodies several major benefits when compared with other MCP and distributed system methodologies:

- Integrated vs. Isolated Approach: While separate treatment of security, reliability & performance is a common feature of traditional methods, the new framework brings these elements together, which not only enhances coordination but also reduces the number of security weaknesses.
- Reactive vs. Proactive Management: Systems that use traditional methods mostly depend on reactive methods of operation; they deal with problems only after they have happened. However, introducing AI/ML into the new framework enables early identification and risk prevention.
- Static vs. Dynamic Adaptation: Old-style systems will most likely be using a fixed setup in terms of scalability and security policies. The new system, on the other hand, continuously adjusts itself to environmental changes, resulting in better performance and survival capability.
- Limited Observability vs. Comprehensive Monitoring: The lack of detailed monitoring capabilities is the biggest shortcoming of most traditional systems. With observability instruments, the new framework can continuously supply data on system functioning.

- Manual vs. Automated Processes: Utilizing automation to carry out failover, scaling, and threat response tasks not only helps to reduce the level of human interaction that is prone to mistakes, but it also leads to better response times.

In fact, the framework generates complexity issues which can disqualify it for deployment in smaller systems with limited resources.

6. CONCLUSION AND FUTURE SCOPE

6.1. CONCLUSION

Security and reliability of MCP (Microservices-based Cloud Platform) represent a major topic of the paper, and the authors give an account of their research in the field of security and reliability of MCP (Microservices-based Cloud Platform). Compliance with security controls and the implementation of failover mechanisms in fault-tolerant architectures, along with the ability to recover from downtime, are highly integrated features of today's distributed systems, as emphasized by the paper on the security and reliability of MCP (Microservices-based Cloud Platform). The scenario-based case study, including simulated cyberattacks and system failures, demonstrated the framework's capabilities in maintaining its position, providing services at frequent intervals, and performing recovery operations effectively.

As for reliability, the use of container orchestration tools like Kubernetes, load balancing, circuit breakers, and database replication techniques effectively kept services up and running even under the worst circumstances. The system was able to greatly reduce the downtime it experienced through automated failover techniques, and within a few seconds of operation it was able to recover, all the while maintaining service availability above industry standards. The fact that the system managed to keep working despite parts breaking down shows how effective redundancy and distributed design are in improving overall reliability.

That said, the paper has brought certain difficulties to light. In addition to the system's complexity, security and advanced monitoring tool integration can make system management a tough job. To sustain system efficiency, it calls for highly skilled personnel and well-structured operational processes. Apart from that, though AI/ML models drastically enhance threat detection, they are not solutions that can be left unattended. They continuously require tuning, retraining, and user feedback collection to maintain their accuracy and ability to adapt to the growing malicious activities. Despite these difficulties, the main research results show that a well-designed MCP framework can provide solid security, performance, and reliability while making minimal sacrifices.

6.2. FUTURE SCOPE

The future of MCP security and reliability points towards making the system more automated, intelligent, and capable of adapting on its own without human intervention. The use of AI-driven security measures continues to be discussed. In the coming days, security systems will take full advantage of more advanced machine learning techniques, including deep learning and reinforcement learning, which will not just help identify anomalies but also forecast possible attack scenarios before they happen. Changing the manner of security provision from reactive to proactive will considerably enhance system defenses.

Moving forward, the focus will also be given on the gearing up of Zero Trust Architecture (ZTA). Instead of relying on perimeter-based security, future MCP systems can implement a "never trust, always verify" policy, which will become the norm, with continuous authentication and authorization for every request. Subsequently, this will cut down the possibility of threats from within as well as unauthorized access. By the same token, edge computing is equally an opportunity to grab. Systems can implement computation and security enforcement closer to the data source, which will not only reduce latency but also improve response times and enable stronger real-time threat detection.

Moreover, future systems stand to gain from enhanced observability and self-healing capabilities. A mix of the latest monitoring aids with an automatic correction mechanism means that systems are capable of detecting a problem and resolving that matter on their own without the need for human assistance. Taking predictive maintenance as an example, one can use such techniques to anticipate upcoming failures and initiate the correct actions before disturbances occur.

More on that, scalability will be given priority. Future MCPs will have to deal with increasing workloads and user demands while incorporating efficient resource management strategies like serverless computing and intelligent auto-scaling. With these approaches, a system can allocate resources on the fly based on demand, ultimately improving cost efficiency and performance. To sum up, having gone through what the current MCP frameworks are capable of in terms of security and handling of failures, we should say that there is still room for in-depth innovation. Through the use of forthcoming technologies and the elimination of existing ones, future systems will not only be able to deliver higher resilience and efficiency but also greater adaptability in a world where the digital environment is getting more and more complex from day to day.

REFERENCES

[1] X. Hou, Y. Zhao, S. Wang, and H. Wang, "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions," ACM Transactions on Software Engineering and Methodology, Feb. 2026, doi: 10.1145/3796519.

[2] Kumar, S., Girdhar, A., Patil, R., & Tripathi, D. (2025). MCP Guardian: A security-first layer for safeguarding MCP-based AI system. arXiv.org. https://doi.org/10.48550/arXiv.2504.12757

[3] Srigadde, B. R. (2025). Maximizing AI Callout Time Using Visualforce Pages in Lightning Components. International Journal of AI, BigData, Computational and Management Studies, 6(3), 127-136. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I3P115.

[4] Suryadevara, S. S. K. (2024). Resilient Multi-CDN Delivery Model Using AI-Based Traffic Switching for Global AEM Deployments. International Journal of Emerging Trends in Computer Science and Information Technology, 5(3), 191-200. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I3P119.

[5] Fang, J., Yao, Z., Wang, R., Ma, H., Xiang, W., & Chua, T.-S. (2025). We should identify and mitigate third-party safety risks in MCP-powered agent systems. arXiv.org. https://doi.org/10.48550/arXiv.2506.13666

[6] Kumar Doodala, A. N. (2024). Validating UX consistency Across Omnichannel Platform. American International Journal of Computer Science and Technology, 6(6), 87-97. https://doi.org/10.63282/3117-5481/AIJCST-V6I6P109.

[7] Muppaneni, R. K. (2023). Low-Code Revolution: How Power Platform Extends Dynamics 365 Capabilities. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(3), 162-171. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I3P119.

[8] Zhao, W., Liu, J., Bonan, R., Li, S., & Liang, Z. (2025). When MCP servers attack: Taxonomy, feasibility, and mitigation. arXiv.org. https://doi.org/10.48550/arXiv.2509.24272

[9] Katangoori, S. (2024). JupyterOps: Version-Controlled, Automated, and Scalable Notebooks for Enterprise ML Collaboration. International Journal of Emerging Trends in Computer Science and Information Technology, 5(3), 211-223. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I3P122.

[10] Takkalapally, D. (2025). 6GSyn: AI-Driven Synthetic Data Generation for Next-Generation Wireless Performance Evolution. International Journal of Emerging Trends in Computer Science and Information Technology, 6(1), 168-177. https://doi.org/10.63282/3050-9246.IJETCSIT-V6I1P120.

[11] M. M. Hasan, H. Li, E. Fallahzadeh, R. G. Krishnan, B. Adams, and A. E. Hassan, "Model Context Protocol (MCP) at First Glance: Studying the Security and Maintainability of MCP Servers," arXiv (Cornell University), June 2025, doi: 10.48550/arxiv.2506.13538.

[12] Gaddam, R. R. (2024). Vertex AI Agent Builder for Regulated Environments. American International Journal of Computer Science and Technology, 6(2), 50-62. https://doi.org/10.63282/3117-5481/AIJCST-V6I2P106.

[13] Allenki, S. S., & Sharma, A. (2025). Troubleshooting Replication Lag and Ensuring Data Consistency in Distributed Systems. American International Journal of Computer Science and Technology, 7(4), 116-128. https://doi.org/10.63282/3117-5481/AIJCST-V7I4P111.

[14] Bin, W., et al. (2025). MCPGuard: Automatically detecting vulnerabilities in MCP servers. arXiv.org. https://doi.org/10.48550/arXiv.2510.23673

[15] Shiramalla, R. (2025). Autonomous Component Lifecycle Management in Salesforce LWC using AI-driven Predictive Rendering. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 6(1), 274-283. https://doi.org/10.63282/3050-9262.IJAIDSML-V6I1P132.

[16] Parakala, A. (2025). Market Growth Insights (2017–2025+) . American International Journal of Computer Science and Technology, 7(6), 25-36. https://doi.org/10.63282/3117-5481/AIJCST-V7I6P103.

[17] Errico, H., Ngiam, J., & Sojan, S. (2025). Securing the Model Context Protocol (MCP): Risks, controls, and governance. arXiv.org. https://doi.org/10.48550/arXiv.2511.20920

[18] T. Mittal, R. Malik, S. S. K. Suryadevara, R. Thakkallapelly, and B. K. R. Guntupalli, "Deep Reinforcement Policy for Coordinating Cooperative Autonomous Vehicles at Highway Intersection Merges," 2025 International Conference on Electrical Engineering and Informatics (ICEEI), pp. 1–7, Nov. 2025, doi: 10.1109/iceei68459.2025.11330686.

[19] Muppaneni, K. (2024). Progressive Web Apps: Offline UX Benchmarking. International Journal of Emerging Trends in Computer Science and Information Technology, 5(2), 174-183. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I2P119.

[20] Y. Guo et al., "MCPXKIT: the Unified Toolkit for Analyzing Model Context Protocol Security," IEEE Transactions on Dependable and Secure Computing, pp. 1–16, 2026, doi: 10.1109/tdsc.2026.3695553.

[21] Takkalapally, D. (2024). ShiftLeft-AI: Machine Learning Framework for Proactive Performance Assurance in CI/CD Pipelines. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(4), 285-296. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I4P126.

[22] Srigadde, B. R., & Guntupalli, B. (2025). Tracking the Status of Long-Running Apex Methods in LWC. International Journal of Emerging Research in Engineering and Technology, 6(1), 147-157. https://doi.org/10.63282/3050-922X.IJERET-V6I1P118.

[23] Y. Lee, W. Choi, and D. Nam, "Supply Chain Threats in the MCP Ecosystem: Attack Vectors and Mitigation Strategies," Lecture Notes in Computer Science, pp. 329–349, Nov. 2025, doi: 10.1007/978-981-95-4674-9_17.

[24] Allenki, S. S. (2025). Troubleshooting Backup, Restore, and Data Export Failures in Relational Databases. International Journal of AI, BigData, Computational and Management Studies, 6(2), 127-137. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I2P115.

[25] Vppalapati, M. (2024). Cooling Domains as First-Class Failure Boundaries in Storage Architecture. American International Journal of Computer Science and Technology, 6(2), 96-106. https://doi.org/10.63282/3117-5481/AIJCST-V6I2P110.

[26] Mehta and S. Gautam, "Enhancing Graphical Processing Performance by Integrating ACO Optimization to Edge Detection Algorithm," Lecture Notes in Networks and Systems, pp. 129–144, 2025, doi: 10.1007/978-981-96-4226-7_10.

[27] Kumar Doodala, A. N. (2025). Continuous Compliance Testing in Healthcare IT Using Shift-Right QA Strategies. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 6(1), 258-267. https://doi.org/10.63282/3050-9262.IJAIDSML-V6I1P130.

[28] Shiramalla, R., & Katangoori, S. (2025). Zero Trust Architecture for Salesforce LWC using Adaptive Authentication Models. American International Journal of Computer Science and Technology, 7(1), 123-135. https://doi.org/10.63282/3117-5481/AIJCST-V7I1P110.

[29] Z. Li, K. Li, B. Ma, M. Xu, Y. Zhang, and X. Cheng, “We Urgently Need Privilege Management in MCP: A Measurement of API Usage in MCP Ecosystems,” 2025 IEEE 22nd International Conference on Mobile Ad-Hoc and Smart Systems (MASS), pp. 555–560, Oct. 2025, doi: 10.1109/mass66014.2025.00090.
[30] Gaddam, R. R. (2023). Progressive Delivery for Models with Quality KPIs. American International Journal of Computer Science and Technology, 5(4), 33-47. https://doi.org/10.63282/3117-5481/AIJCST-V5I4P104.
[31] Muppaneni, K., & Palem, V. (2024). Micro-Frontend Design Patterns for Multi-Framework Applications. International Journal of Emerging Research in Engineering and Technology, 5(3), 181-190. https://doi.org/10.63282/3050-922X.IJERET-V5I3P120.
[32] X. Li and X. Gao, “A First Look at the Security Issues in the Model Context Protocol Ecosystem,” 2026 56th Annual IEEE International Conference on Dependable Systems and Networks (DSN), pp. 379–392, June 2026, doi: 10.1109/dsn69566.2026.00046.
[33] Muppaneni, R. K. (2024). Why More Organizations Are Moving from NetSuite to Dynamics 365. American International Journal of Computer Science and Technology, 6(4), 59-70. https://doi.org/10.63282/3117-5481/AIJCST-V6I4P106.
[34] Parakala, A., & Padgett, P. (2025). When AI Acts: Opportunities and Risks of Agentic Systems. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 6(4), 29-40. https://doi.org/10.63282/3050-9262.IJAIDSML-V6I4P105.
[35] Zong, Xuanjun, et al. "MCP-SafetyBench: A Benchmark for Safety Evaluation of Large Language Models with Real-World MCP Servers." arXiv preprint arXiv:2512.15163 (2025).
[36] Katangoori, S., & Ghosh, D. (2025). Data Mesh Meets AI: Federated Ownership and ML-Enabled Contracts for Cross-Domain Data Collaboration. International Journal of AI, BigData, Computational and Management Studies, 6(2), 148-157. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I2P117.
[37] Brett, I. (2025). Simplified and secure MCP gateways for enterprise AI integration. arXiv.org. https://doi.org/10.48550/arXiv.2504.19997
[38] Vppalapati, M. (2024). Power-Bound Storage Design: Architecting Systems for Electrical Scarcity. International Journal of AI, BigData, Computational and Management Studies, 5(1), 208-217. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I1P121.
[39] Janjusevic, S., Garcia, A. B., & Kazerounian, S. (2025). Hiding in the AI traffic: Abusing MCP for LLM-powered agentic red teaming. arXiv.org. https://doi.org/10.48550/arXiv.2511.15998
[40] Taluri, R. (2024). A Cloud-Native Reference Architecture for Data Engineering, Generative AI, and Decision Intelligence Using AWS and Amazon Bedrock. International Journal of AI, BigData, Computational and Management Studies, 5(1), 218-227. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I1P122
[41] Akinapalli, S. (2026). AN AI-POWERED DATA TRUST AND QUALITY SCORING FRAMEWORK FOR ENTERPRISE DECISION INTELLIGENCE SYSTEMS. International Journal of Data Science and IoT Management System, 5(1), 946-950. https://doi.org/10.64751/ijdim.2026.v5.n1.1026
[42] K. K. Sharma, S. Sekhar and V. Venkatesh, "Novel Paradigm for Privacy-Preserving Data Sharing and Anonymization in Smart Homes," 2025 International Conference on Artificial Intelligence's Future Implementations (ICAIFI), Yogyakarta, Indonesia, 2025, pp. 47-51, doi: 10.1109/ICAIFI66942.2025.11326168
[43] Bodapati, S. J., & Merakanapalli, S. (2024). AI-Driven Fail Operational Safety in Wire Control Systems. International Journal of Emerging Trends in Computer Science and Information Technology, 5(1), 119-127. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I1P113
[44] S. K. Sunkara, "Artificial Intelligence and Machine Learning in Pharma: Revolutionizing Drug Development and Clinical Trials," 2025 12th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida NCR, India, 2025, pp. 1-5, doi: 10.1109/ICRITO66076.2025.11241250.