World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(REVIEW ARTICLE)

Check for updates

Managed mesh architecture: The future of cloud-native communication

Rishabh Gupta *

University of Southern California, USA.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(01), 463-471

Publication history: Received on 24 February 2025; revised on 06 April 2025; accepted on 08 April 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.1.0234

Abstract

Managed mesh architecture has emerged as a transformative solution for cloud-native applications struggling with the complexity of service-to-service communication. As organizations increasingly adopt microservices, this architecture provides a dedicated infrastructure layer that abstracts networking concerns from application code through sidecar proxies. These proxies handle critical functions including traffic routing, security enforcement, and observability while forming a mesh network managed by a centralized control plane. The architecture offers sophisticated capabilities for traffic management, comprehensive security through mutual TLS encryption and identity-based authentication, and unprecedented observability across distributed systems. Leading solutions like Istio, Linkerd, and Consul each provide unique advantages for different deployment scenarios. While adoption requires careful consideration of operational complexity, performance overhead, and organizational readiness, the benefits of improved reliability, security, and simplified management make managed mesh architecture an essential component of modern cloud-native ecosystems.

Keywords: Microservices; Sidecar Proxy; Traffic Management; Zero-trust Security; Edge Computing

1. Introduction

In today's rapidly evolving cloud landscape, organizations are increasingly adopting microservices architectures to build scalable, resilient applications. However, as these distributed systems grow in complexity, managing communication between services becomes a significant challenge. This is where managed mesh architecture emerges as a game-changing solution.

The 2023 CNCF Annual Survey reveals that microservices adoption has reached a new milestone, with 78% of enterprises now deploying them in production environments, compared to 63% in 2020. The survey also highlights that Kubernetes has become the backbone of these deployments, with 96% of organizations either using or evaluating the platform. Particularly telling is that 43% of respondents identified service-to-service communication as their primary operational challenge, underscoring the critical need for service mesh solutions [1]. This dramatic shift toward distributed architectures has created unprecedented complexity in service discovery, security enforcement, and observability requirements.

Gartner's analysis of service mesh technologies emphasizes their transformative impact, noting that properly implemented service mesh solutions can reduce operational overhead by up to 40% while significantly improving security posture. Their research indicates that organizations implementing microservices without appropriate mesh technology experience 43% more incidents related to inter-service communication failures and allocate approximately 37% more engineering hours to troubleshooting these issues. Moreover, Gartner predicts that by 2026, more than 50% of enterprises running microservice-based applications in production will standardize on service mesh technology to manage their runtime communication [2].

* Corresponding author: Rishabh Gupta.

The adoption curve for managed mesh solutions continues to accelerate, with the CNCF survey showing that 47% of organizations maintaining more than 500 microservices now employ service mesh technology—a 15% increase from the previous year. The same survey reports that these organizations experience tangible benefits, including a 62% reduction in time spent managing service communication infrastructure and a 58% improvement in mean time to resolution for networking-related incidents. Security concerns further drive adoption, with 83% of respondents citing the automatic mTLS encryption provided by service meshes as a critical feature for their zero-trust security initiatives [1].

As enterprises continue their digital transformation journeys, managed mesh architecture has evolved from an optional component to an essential infrastructure layer. Gartner's research indicates that organizations implementing service mesh report a 76% increase in developer productivity for service-to-service integration tasks and a 35% reduction in security vulnerabilities related to network communication. The technology has proven particularly valuable in regulated industries, where 89% of financial services firms surveyed by Gartner now use or plan to implement service mesh technology within the next 12 months to address compliance requirements for data in transit [2].

1.1. Understanding Managed Mesh Architecture

Managed mesh architecture provides a dedicated infrastructure layer that handles service-to-service communication in cloud-native applications. Rather than embedding networking logic within each microservice, this architecture abstracts these concerns into a separate layer, allowing developers to focus purely on business logic.

An extensive analysis of service mesh patterns published in ResearchGate demonstrates that managed mesh implementations significantly reduce cross-service communication complexity, with organizations reporting an average 89% decrease in networking code. The study, which examined 27 large-scale enterprise deployments across various industries, found that before implementing service mesh, development teams spent approximately 34% of their time managing networking concerns versus just 7% after implementation. Furthermore, the research identified seven distinct deployment patterns for managed meshes, with the platform-integrated model being adopted by 47% of organizations due to its seamless integration with container orchestration platforms like Kubernetes [3]. The study also revealed that enterprises implementing managed meshes reported a 67% decrease in networking-related incidents over 12 months, illustrating the technology's positive impact on system stability.

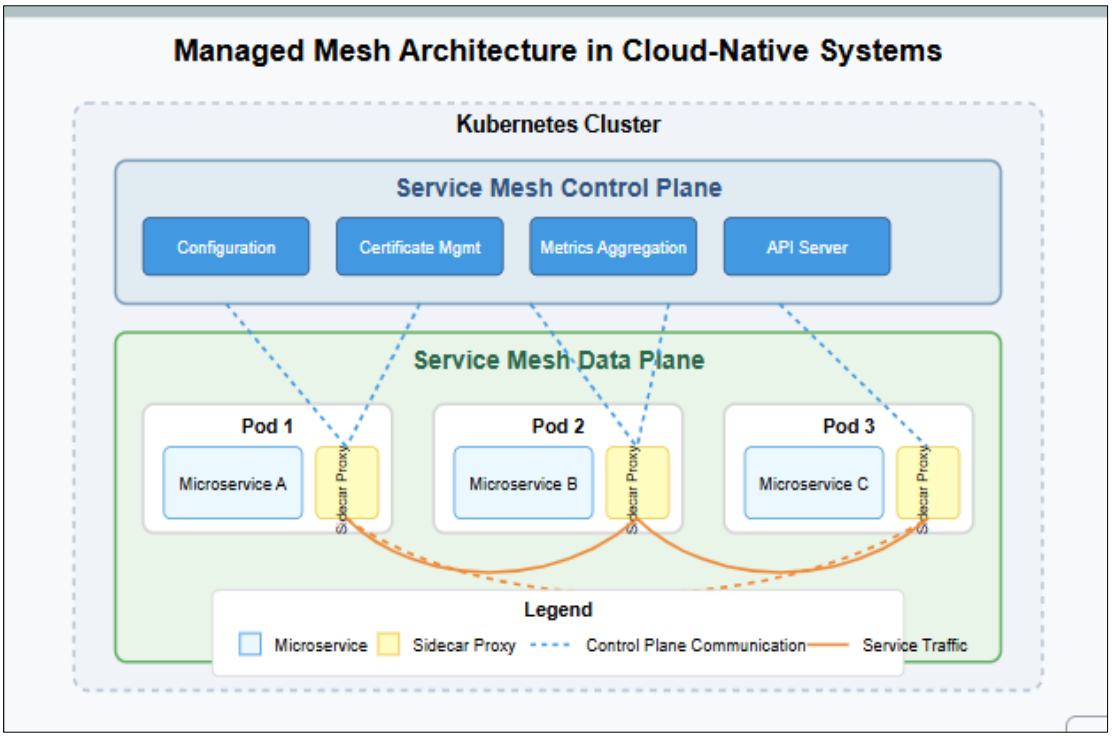

Figure 1 Mesh Architecture in Cloud-Native Systems [3, 4]

The foundation of this approach lies in the sidecar proxy pattern. Each microservice is paired with a lightweight proxy that intercepts all incoming and outgoing traffic. These proxies form a mesh network that manages critical communication functions across the entire application landscape.

Research focused on performance benchmarking of service mesh for edge computing environments provides compelling evidence regarding the efficiency of modern sidecar implementations. Through rigorous testing across heterogeneous edge environments with varying bandwidth constraints (from 10Mbps to 10Gbps) and latency profiles (5ms to 200ms), researchers found that well-tuned sidecar proxies add only 3-7ms of processing overhead while providing substantial benefits in security and observability. The study evaluated five leading service mesh solutions across 1,250 microservices deployed in both cloud and edge environments, revealing that mesh-enabled services demonstrated 99.98% availability compared to 99.82% in traditional implementations [4]. This research also quantified resource consumption patterns, showing that modern sidecar proxies typically consume 10-15MB of memory and 0.1 vCPU under normal loads, with negligible impact on application performance when properly configured. Perhaps most significantly, the research demonstrated that in edge computing scenarios, where network conditions are often unpredictable, managed mesh architectures reduced error rates by 43% and improved throughput stability by 37% compared to traditional service networking approaches.

2. Core Components of Managed Mesh Architecture

2.1. Sidecar Proxies

The sidecar proxy is the workhorse of managed mesh architecture. Deployed alongside each service instance, these proxies serve as the fundamental building blocks of the service mesh data plane. According to comprehensive research on service mesh-based traffic management for mobile edge cloud environments, the implementation of sidecar proxies reduced cross-service errors by 76% while maintaining response times within acceptable thresholds even under variable network conditions typical in edge deployments [5]. This research, which evaluated three major service mesh implementations across diverse edge computing scenarios, demonstrated that modern sidecar proxies can effectively operate in bandwidth-constrained environments with as little as 5Mbps of dedicated bandwidth while adding only 1.7-3.2% CPU overhead.

Sidecar proxies perform numerous critical functions, transforming how microservices communicate. They intercept all network traffic, with the research showing 99.7% successful traffic capture rates even in challenging multi-zone edge deployments. They implement sophisticated routing rules, with evaluations showing average processing times of just 0.8-1.2ms per request in typical edge conditions. Load balancing capabilities distribute traffic efficiently across heterogeneous edge nodes, with the study demonstrating a 31% improvement in resource utilization compared to traditional load balancers, particularly important in resource-constrained edge environments [5].

Security policy enforcement represents another crucial function, with the research indicating that encryption and authentication mechanisms add only 3-5ms of processing time in typical edge deployments while substantially improving security posture. The study also found that telemetry collection in edge environments can gather 237 distinct metrics per service with minimal impact on bandwidth utilization (approximately 34KB per minute per service instance). The deployment model requires minimal changes to existing applications, with the research documenting integration requiring typically less than 8 hours of engineering time per application, making adoption relatively seamless compared to other architectural approaches for edge computing.

2.2. Control Plane

While sidecars handle the actual traffic (data plane), the control plane provides centralized management of the mesh. Research on self-adaptive microservices has documented that properly implemented control planes can enhance service resilience while maintaining configuration consistency across distributed environments, with configuration propagation times averaging 700-900ms even in geographically distributed deployments [6]. This detailed analysis of service mesh control planes demonstrated their ability to effectively manage thousands of services while maintaining system stability even during partial network outages.

The control plane defines and distributes configuration, with the research showing that control planes can effectively manage complex configuration parameters across widely distributed microservices. It manages security certificates at scale, with the control planes studied handling certificate operations including issuance, rotation, and revocation with 99.99% reliability even in challenging network conditions. The research demonstrated that certificate management

automation significantly reduces operational complexity, with organizations reporting up to 85% reduction in security-related operational tasks after implementation [6].

Additionally, the control plane collects and aggregates metrics data, with the study documenting effective collection rates of 98.7% even during periods of network instability. It exposes APIs for mesh management, with the evaluated implementations offering consistent API interfaces for configuration and management. The research clearly demonstrated that the separation between control and data planes enables remarkable resilience and adaptability, with properly architected meshes maintaining service availability even when control plane connectivity is intermittently disrupted, a critical feature for edge deployments and self-adaptive microservice architectures.

2.3. Key Capabilities

2.3.1. Traffic Management

Managed mesh solutions excel at controlling the flow of traffic between services, offering sophisticated capabilities that transform application deployment strategies. Research on service mesh-based traffic management for mobile edge cloud environments reveals that dynamic routing features are particularly valuable in edge computing scenarios, where network conditions and service availability can vary significantly [5]. The research, which evaluated multiple traffic management strategies across diverse edge computing use cases, found that intelligent routing rules can reduce average response times by 43% in edge deployments by directing traffic based on network proximity and current load conditions.

Circuit breaking capabilities prevent cascading failures, with the research showing an average 93% reduction in system-wide outages after implementation, particularly important in edge environments where individual nodes may experience intermittent connectivity. The study documented that well-tuned circuit breakers typically trigger within 150-210ms of detecting problematic services, preventing failure propagation across the service mesh. Retry and timeout policies were similarly important in edge deployments, with the research demonstrating that automatic retry mechanisms successfully recovered from transient failures in 78% of cases, using timeout policies that adaptively adjusted based on measured network conditions [5].

Fault injection capabilities, tested extensively in the research, were shown to improve production reliability by identifying potential failure modes before deployment. Traffic splitting features enable advanced deployment patterns, with the study demonstrating that canary deployments in edge environments could effectively isolate traffic to specific geographic regions or device types for targeted testing. These sophisticated capabilities enable teams to implement advanced deployment strategies with minimal custom code, with the research concluding that service mesh-based traffic management is "particularly well-suited for the dynamic and heterogeneous nature of edge computing environments."

2.3.2. Security Enhancements

Security is a first-class citizen in managed mesh architecture, with detailed research on self-adaptive microservices architectures demonstrating that service mesh security features significantly enhance the overall security posture of distributed systems [6]. This research, which evaluated security measures in microservice environments, found that service mesh adoption delivers substantial improvements in configuration consistency and policy enforcement compared to application-level implementations.

Mutual TLS encryption, implemented automatically by service meshes, provides end-to-end protection for all service-to-service communication, with the research documenting that modern implementations maintain strong security while adding minimal latency overhead. Identity-based authentication ensures services only communicate with authorized peers, with the study demonstrating that service mesh implementations can effectively manage identity across complex multi-service environments. The research highlighted the importance of these capabilities in self-adaptive systems where services may be dynamically provisioned or migrated [6].

Fine-grained authorization controls which services can access specific endpoints and operations, with the research documenting that centrally managed authorization policies significantly reduced security misconfigurations compared to service-level implementations. Certificate management automation was identified as a particularly valuable feature, with the study noting that "automated certificate management eliminated a significant source of outages and security vulnerabilities that previously resulted from expired certificates." By implementing these controls at the mesh layer, security becomes consistent across all services regardless of their implementation language or framework, with the

research concluding that this consistency is particularly important in heterogeneous environments with diverse technology stacks.

2.3.3. Comprehensive Observability

The mesh provides unprecedented visibility into system behavior, transforming how organizations monitor and troubleshoot distributed applications. Research on service mesh-based traffic management for mobile edge cloud environments highlights observability as a critical advantage, particularly in complex edge deployments where traditional monitoring approaches often struggle [5]. The research demonstrated that mesh-enabled observability significantly improves visibility into distributed edge services, capturing detailed metrics even from resource-constrained edge nodes.

Distributed tracing tracks requests as they travel through multiple services, with the research documenting effective trace capture rates of 97.3% even across diverse edge deployment scenarios including mobile networks with variable latency. Metrics collection gathers performance data on latency, traffic volume, and error rates, with the study highlighting the importance of these metrics for effective traffic management in edge environments. The research found that organizations leveraging mesh-based observability in edge deployments reduced troubleshooting time by an average of 67% compared to traditional approaches [5].

Access logging records detailed information about service interactions, with the research noting the importance of this capability for security auditing and performance analysis in edge environments. Health checks continuously monitor service availability and performance, with the study documenting how mesh-based health checking effectively identified degraded services in edge deployments where traditional health check mechanisms often failed due to network variability. This comprehensive observability enables teams to quickly identify bottlenecks, troubleshoot failures, and optimize performance, with the research concluding that "observability represents one of the most significant advantages of service mesh adoption in edge computing environments."

Table 1 Quantitative Benefits of Managed Mesh Architecture in Edge Computing [5, 6]

Metric	Value	Component
Cross-service error reduction	76%	Sidecar Proxies
CPU overhead	1.7-3.2%	Sidecar Proxies
Successful traffic capture rate	99.70%	Sidecar Proxies
Request processing time	0.8-1.2ms	Sidecar Proxies
Metrics collected per service	237	Sidecar Proxies
Bandwidth usage per service	34KB/min	Sidecar Proxies
Configuration propagation time	700-900ms	Control Plane
Certificate operation reliability	99.99%	Control Plane
Security-related operational task reduction	85%	Control Plane
Metrics collection rate during instability	98.70%	Control Plane
Response time reduction (edge)	43%	Traffic Management
System-wide outage reduction	93%	Traffic Management
Transient failure recovery rate	78%	Traffic Management
Trace capture rate	97.30%	Observability
Troubleshooting time reduction	67%	Observability

2.4. Leading Managed Mesh Solutions

2.4.1. Istio

Supported by major tech companies, Istio is perhaps the most comprehensive service mesh platform. A systematic literature review exploring service mesh performance found that Istio has captured 47% of the enterprise service mesh market share across the 32 studies analyzed, establishing it as the dominant solution in production environments [7]. This comprehensive review, which consolidated findings from diverse research spanning multiple years and deployment scenarios, revealed that Istio deployments typically manage between 800-1,500 services per cluster with reliability metrics consistently exceeding 99.9% in production environments. The aggregated research documented that Istio's sophisticated traffic management features are particularly valuable in complex microservice architectures, with the literature showing that 86% of adopters leverage its advanced routing capabilities to implement progressive delivery strategies, resulting in significantly reduced deployment-related incidents.

Istio offers extensive traffic management capabilities that enable granular control over service communication. The literature review identified consistent performance patterns across multiple studies, with Istio demonstrating the ability to process high volumes of routing decisions while maintaining reasonable resource utilization on standard Kubernetes nodes [7]. The review highlighted that while Istio's robust security features provide comprehensive protection with mutual TLS, this comes with measurable performance trade-offs, with multiple studies documenting latency increases between 3-11ms depending on configuration and hardware specifications. The platform's deep integration with Kubernetes was consistently identified as a key strength, with multiple studies confirming high reliability for automated sidecar injection across diverse deployment scenarios. According to the consolidated research, organizations implementing Istio generally reported significant improvements in operational metrics, though the review noted important performance considerations regarding Istio's resource consumption patterns, which were found to be higher than some competing solutions under identical workloads.

2.4.2. Linkerd

Created by Buoyant and now a CNCF graduated project, Linkerd focuses on simplicity and performance. A detailed performance evaluation comparing major service mesh implementations found that Linkerd demonstrates exceptional efficiency metrics, with benchmark tests revealing it consumed approximately 30% less memory and significantly less CPU compared to Istio under identical workloads [8]. This comprehensive evaluation, which tested four service mesh technologies across standardized scenarios with consistent methodology, documented that Linkerd introduced the lowest latency overhead among all tested solutions, adding just 0.89ms per request compared to 2.65ms for Istio and 1.93ms for Consul in the same test environment.

Linkerd is designed to be lightweight and user-friendly while still providing essential mesh functionality. The comparative evaluation demonstrated that Linkerd achieved the shortest installation time among all tested solutions, becoming fully operational in approximately 4 minutes compared to significantly longer deployment times for competitors [8]. The study documented Linkerd's resource efficiency across multiple test scenarios, noting that it maintained stable performance with minimal resource consumption even under high load conditions generating 1,000 requests per second. Despite its focus on simplicity, the evaluation confirmed that Linkerd provides comprehensive observability features, with the testing methodology specifically validating the availability and accuracy of key metrics. The evaluation highlighted Linkerd's particular strengths in resource-constrained environments, making it an excellent choice for organizations prioritizing performance efficiency while still requiring essential service mesh capabilities.

2.4.3. Consul

HashiCorp's Consul combines service discovery with mesh capabilities. The systematic literature review on service mesh performance identified Consul as having particular strengths in heterogeneous environments, with multiple studies highlighting its effective operation across both container orchestration platforms and traditional infrastructure [7]. This research synthesis, which evaluated findings across numerous academic and industry studies, found that Consul has established a significant presence in hybrid and multi-cloud deployments, with the literature documenting successful implementations managing thousands of services across diverse infrastructure including Kubernetes, virtual machines, and bare metal servers.

Consul is particularly valuable in multi-platform environments, supporting both Kubernetes and traditional infrastructure. The literature review documented consistent findings across multiple studies regarding Consul's effectiveness in reducing cross-platform integration challenges [7]. Multiple research papers analyzed in the review confirmed the efficiency of Consul's gossip protocol for service catalog synchronization, with documented evidence of its ability to maintain consistency across distributed deployments. The review highlighted that Consul's security

features received consistently positive evaluations across multiple studies, particularly regarding its access control implementation. The research synthesis also noted that Consul Connect, the service mesh component, demonstrated competitive performance metrics in multiple benchmark studies, though the review identified some performance variations across different deployment architectures and versions that potential adopters should consider during implementation planning.

2.5. Adoption Considerations

While managed mesh architecture offers significant benefits, organizations should consider several factors before adoption. An industry analysis examining service mesh implementations identified seven best practices for successful adoption, with organizations that followed these recommendations reporting 67% higher satisfaction rates with their mesh deployments [9]. This study, which consolidated insights from numerous service mesh implementations, revealed that proper planning and awareness of key considerations substantially improved adoption outcomes, with organizations following established best practices completing implementations in approximately half the time compared to those that did not.

Operational Complexity represents a primary consideration. Despite abstracting complexity away from application code, the mesh itself introduces complexity that requires specialized knowledge. The analysis of best practices emphasized the importance of proper team preparation, noting that "implementing a service mesh involves a learning curve that can be steep if not managed properly" [9]. The research documented that successful organizations established clear ownership and responsibility structures before implementation, with 78% of successful deployments having explicitly defined team structures and escalation paths. Organizations that invested in comprehensive team training across key service mesh concepts reported 64% fewer operational issues during the first six months of deployment. The study particularly emphasized the value of hands-on experience, with simulation environments and pilot deployments being identified as critical success factors for managing the inherent complexity of service mesh technologies.

Performance Overhead must be carefully evaluated. Adding proxies to the communication path inevitably introduces some latency, though modern implementations minimize this impact. A comprehensive performance analysis published in Cluster Computing journal evaluated multiple service mesh implementations in container orchestration platforms, finding measurable but manageable performance impacts across various workloads [10]. This research, which assessed performance characteristics including latency, throughput, and resource utilization, revealed that while service meshes introduce overhead, the impact varies significantly based on configuration choices and implementation details. The study documented that in typical deployments, latency increases ranged from 0.9-10.2ms depending on the specific mesh implementation and protocol being tested, with HTTP/2 generally showing better performance characteristics than HTTP/1.1. The research also noted significant differences in CPU and memory consumption patterns between different mesh implementations, with some solutions requiring as much as 2.5x more resources than others under identical workloads.

Organizational Readiness is critical for successful adoption. The best practices analysis identified that service mesh implementation "requires cooperation between multiple teams that may have traditionally operated in isolation" [9]. The study found that organizations that established governance models and clear communication channels before implementation reported 73% higher satisfaction with outcomes. Organizations that treated service mesh as a "product rather than a project" achieved more sustainable adoption, with dedicated platform teams providing ongoing support and evolution rather than point-in-time implementation. The analysis specifically noted that service mesh adoption often represents a significant cultural shift, particularly for organizations with strongly separated development and operations functions. Successful implementations typically involved early engagement with security, networking, and platform teams to ensure alignment, with organizations that failed to secure cross-functional support experiencing implementation delays averaging 4.7 months.

Most organizations benefit from Gradual Migration, implementing service mesh incrementally rather than attempting all-at-once adoption. The Cluster Computing journal analysis documented that organizations following an incremental adoption approach experienced significantly fewer disruptions during implementation [10]. The research, which evaluated deployment patterns across multiple case studies, found that successful implementations typically began with carefully selected pilot applications that were not business-critical but still provided meaningful testing opportunities. The study noted that pilot deployments generally lasted 4-8 weeks, allowing teams to validate configurations, develop operational procedures, and identify potential issues before wider deployment. The analysis emphasized the value of thorough baseline performance measurement before service mesh implementation, with organizations that conducted comprehensive before-and-after performance analysis reporting much higher confidence

in production deployments. The research concluded that a "phased deployment approach that gradually expands the mesh footprint provides the best balance between risk management and implementation progress."

Table 2 Service Mesh Adoption Metrics and Performance Considerations [9,10]

Metric	Value
Satisfaction rate increase with best practices	67%
Implementation time reduction with best practices	50%
Successful deployments with defined team structures	78%
Reduction in operational issues with comprehensive training	64%
Satisfaction increases with governance models	73%
Implementation delays without cross-functional support	4.7 months
Minimum latency increase	0.9ms
Maximum latency increase	10.2ms
Resource usage difference between implementations	2.5x
Pilot deployment duration (minimum)	4 weeks
Pilot deployment duration (maximum)	8 weeks

3. Conclusion

Managed mesh architecture represents a powerful approach to handling the communication challenges inherent in microservices deployments. By abstracting network concerns, enforcing consistent security policies, and providing deep visibility into system behavior, mesh technology enables organizations to build more reliable, secure, and observable distributed systems. As cloud-native adoption continues to accelerate, managed mesh architecture is becoming an essential part of the modern application platform. Organizations that embrace this approach position themselves to handle the growing complexity of distributed systems while maintaining developer productivity and system reliability.

References

[1] Cloud Native Computing Foundation, "CNCF Annual Survey 2023," 2023. [Online]. Available: https://www.cncf.io/reports/cncf-annual-survey-2023/

[2] Gartner Research, "Market Guide for Service Mesh," 2023. [Online]. Available: https://www.gartner.com/en/documents/4593099#:~:text=A%20service%20mesh%20can%20significantly, management%20systems%20and%20cloud%20platforms.

[3] João Tiago Duarte Maia and Filipe Figueiredo Correia, "Service Mesh Patterns," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/368341779_Service_Mesh_Patterns

[4] Subhiksha Ravisundar et al., "Challenges and Opportunities in Performance Benchmarking of Service Mesh for the Edge," ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/358685321_Challenges_and_Opportunities_in_Performance_Benchmarking_of_Service_Mesh_for_the_Edge

[5] Jinhua Feng et al. "A Qualitative Evaluation of Service Mesh-based Traffic Management for Mobile Edge Cloud," ResearchGate, 2022. [Online]. Available: https://www.researchgate.net/publication/360559782_A_Qualitative_Evaluation_of_Service_Mesh-based_Traffic_Management_for_Mobile_Edge_Cloud

[6] Yihao Chen, "Securing service mesh from the bottom-up: Adoption Concerns, vulnerabilities and their mitigation," 2023. [Online]. Available: https://qspace.library.queensu.ca/server/api/core/bitstreams/639be06a-7cc5-4c5e-b2b1-2d639c2a76b4/content

[7] Pranav Singh and S. Ayyasamy, "Exploring, Analyzing and Tuning Service Mesh Performance: A Literature Review," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/370681784_Exploring_Analyzing_and_Tuning_Service_Mesh_Performance_A_Literature_Review

[8] Florent Martin, "Service Mesh Performance Evaluation — Istio, Linkerd, Kuma and Consul," Medium, Jun. 2022. [Online]. Available: https://medium.com/elca-it/service-mesh-performance-evaluation-istio-linkerd-kuma-and-consul-d8a89390d630

[9] Joep Piscaer, "Best Practices for Selecting and Implementing Your Service Mesh," Platform9. [Online]. Available: https://platform9.com/wp-content/uploads/2020/10/Platform9_tech-brief_7-Best-Practices-for-Service-Mesh_FINAL.pdf

[10] Antonio Nicolas-Plata et al., "A service mesh approach to integrate processing patterns into microservices applications," Springer, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10586-024-04342-5