International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 5, Issue 2, 144-149, 2024
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V5I2P116

Original Article

Design and Evaluation of Secure Microservices Architecture for HIPAA-Compliant Prescription Processing on AWS and OpenShift

Srikanth Reddy Gudi
Cigna Evernorth Health Services Inc., Charlotte, North Carolina, USA.

Abstract - *With evermore challenges in balancing compliance with innovation, the healthcare industry is struggling to update its prescription processing systems. This Research Analyses the Secure Microservices Architecture Design and Evaluation of secure micro services architecture for Hipaa compliant prescription processing deployed on AWS and OpenShift Platforms. The main goals are to analyze security protocols, measure performances and evaluate compliance approaches for containers-based healthcare systems. We conducted a mixed-methods research methodology that quantitatively analyzed performance and then qualitatively assessed security-based reasons across multiple implanting microservices-based prescription inside a wide range of healthcare organizations. Hypothesis: Microservices architecture can offer better HIPAA compliance rates, as compared to monolithic systems, if it is properly set up with appropriate defense-in-depth security controls. The research shows that organizations running microservices on hybrid cloud platforms were 94.7% more likely to achieve compliance, with large gains in audit logging, access control, and data encryption metrics. In the article, we discuss how service mesh and container orchestration give you the per pod security controls required for handlign protected health information. In summary, secure microservices architecture shows a possible route for modernizing healthcare organizations while moving to awareness of the strong HIPAA requirements still in place.*

Keywords - *Microservices Architecture, HIPAA Compliance, Prescription Processing, AWS Healthcare, OpenShift Container Platform.*

1. Introduction

The end-to-end digital transformation of all healthcare systems has been greatly accelerated, with prescription processing a key area where operational efficiency will be demanded in conjunction with high security. As noted by Kratzke & Quint (2017), modern demands of interoperability, scalability and regulatory compliance goes way beyond what legacy monolithic systems can offer, and healthcare organizations have increasingly acknowledged this limitation. The Health Insurance Portability and Accountability Act outlines extensive expectations for safeguarding electronic Protected Health Information and sets forth complicated compliance hurdles for system architects (McGraw 2019). A Microservices architecture is a new promising paradigm of service-based architecture for health care applications comprising decomposed, independently deployable, scalable, and secure services (Newman, 2021). That architectural style is much in tune with the latest Dev Sec Ops method where the continuous integration and deployment goes hand in hand with security implementations during the development life cycle of a software (Balalaie et al., 2016). Infrastructure-Level capabilities provided by cloud platforms (like Amazon Web Services) and container orchestration systems (like Red Hat OpenShift) required for enterprise-grade microservices stacks.

Challenges include the need to know, score and track controlled substances, contend with privacy practices, implement pharmacist verification workflows, and manage pharmacy benefit management system (supply chain) integration, when the patient is treated on the prescription processing domain. However, traditional architectures often find it hard to meet these needs, while keeping the agility needed to quickly react to changing regulatory needs (Dragoni et al., 2017). Addressing these ongoing challenges often requires moving away from established paradigms, however, and the combination of microservices principles with cloud-native security mechanisms has the potential to do just that. However, limited research exists on microservices applications in healthcare settings, and limited research has evaluated the outcome of HIPAA compliance on such systems empirically. Earlier works approach the topic of microservices architecture separately or focus on healthcare compliance without considering the intersection between the two (Jamshidi et al. This research work fills the identified research gap by addressing secure microservices architecture that provides comprehensive analysis specifically for applications to process prescribed medicinal items. The importance of this study goes beyond academic research because growing demand for healthcare organizations to upgrade legacy prescription systems is a major concern to avoid becoming non-compliant — which comes with serious financial consequences. Practitioners are presented a myriad of technology choices and use

recommendations, both in terms of specific use cases: this is where well-understood architectural patterns (and security configurations) enable informed technology investment and deployment decisions (Esposito et al.

2. Literature Review

A lot has been written about how microservices architecture has evolved in the domain of enterprise contexts starting from foundational work on design principles such as bounded contexts, API-first design and choreographed service communication (Lewis & Fowler, 2014). Follow-up studies have advanced the challenges facing microservices into categories, and focus areas, including service decomposition strategies, distributed data management, and operational complexity, as the first and foremost areas of concern in need of systematic solutions (Di Francesco et al., 2019). There are more and more academic publications dealing with microservices in the healthcare domain. Alonso et al. Microservices applications (2018) investigated the use of microservices for clinical decision support systems, resulting in higher maintainability of the system and lower risk of deployment. They noticed that after migration organizations could deliver features 40% faster, but security implementation patterns were only sparsely covered.

Cloud environments HIPAA compliance has been reviewed from various angles. Chen et al. AWS HIPAA Compliance 3rd party evaluation: (2020) Compared the security controls for top cloud service providers and stated that, AWS Healthcare-specific services provides a high level of compliance tooling but needs diligent configuration to meet HIPAA compliance end to end. The research highlights encryption key management and audit logging as areas especially in need of attention when implemented. Container security within regulated industries has thereby become its own area of scholarship. Shu et al. (2017) conducted an extensive vulnerability analysis of container images that found that 30% of all official Docker Hub images had severe vulnerabilities which are not fit for healthcare deployments. This places emphasizes the need for HIPAA-compliant systems to scan their container images and select a secure base image. Enterprise contexts have been tested for OpenShift platform security measures. Sultan et al. Container orchestration; role-based access control; network policies; and secrets management were among the security mechanisms identified by & Buller (2019). In their comparative analysis, they placed OpenShift well for regulated workloads as it comes with better security defaults and integrated compliance scanning functionalities.

We have seen microservices deployment taking great advantage of security through service mesh technologies. In a study of Istio and other service mesh implementations, Calcote and Butcher (2019) noted features available to support mutual TLS, traffic encryption, and fine-grained authorization policies. These features mirror HIPAA requirements & rules for secure data transmission and storage and limit access to only what is necessary. Microservices Design Patterns influenced by Healthcare Interoperability Standards (notably, HL7 FHIR) Mandel et al. (2016) described FHIR-based capabilities that are inherently interoperable and microservices-based, enabling the development of systems that can share standardized health data while also being designed to support modular functionality.

3. Objectives

1. To analyze security architecture patterns enabling HIPAA compliance within microservices-based prescription processing systems deployed on AWS and OpenShift platforms.
2. To evaluate performance metrics and scalability characteristics of containerized prescription processing applications under varying workload conditions.
3. To assess the effectiveness of defense-in-depth security controls including encryption, access management, and audit logging in achieving regulatory compliance.
4. To develop implementation recommendations for healthcare organizations seeking to modernize prescription processing systems while maintaining HIPAA compliance.

4. Methodology

This study used a cross-sectional research design that included both a quantitative assessment of performance and compliance metrics, and qualitative assessment of security architecture. Study setting This study was conducted between January 2022 and December 2023 to analyse microservices implementations among healthcare organizations within India and the United States which had deployed prescription processing systems via cloud and container platforms. We induced our sample: We purposively sampled 12 healthcare organizations that utilized microservices-based prescription processing systems (including small to medium clinics to very large hospital networks) on a wide variety of deployments in AWS-native, OpenShift on-premises and hybrid deployments. Selection criteria included minimum of 6 months in production, documented HIPAA compliance assessments, and ability to share anonymized performance and compliance data.

Methods Data collection tools consisted of structured compliance assessment questionnaires based on the HIPAA Security Rule administrative, physical, and technical safeguard requirements. Performance metrics collected using application performance monitoring tools such as AWS CloudWatch, Prometheus, Prometheus and Grafana dashboards deployed over participant organizations We used common assessment frameworks for security architecture evaluations: NIST Cybersecurity Framework mappings and CIS Benchmarks for container platforms. Methods of Analysis Descriptive statistical analysis of

compliance achievement rates, performance metric distributions, and security control implement frequencies. Differences between deployment configurations and organization types were analyzed comparatively. We performed a thematic analysis of our qualitative data based on our security assessments to identify common architectural patterns and implementation challenges. All procedures of data collection and analysis ensured confidentiality of the participants through anonymisation protocols.

5. Results

Table 1: HIPAA Compliance Achievement Rates by Deployment Configuration (n=12)

Deployment Configuration	Organizations	Administrative Safeguards (%)	Physical Safeguards (%)	Technical Safeguards (%)	Overall Compliance (%)
AWS Native	4	96.2	91.5	94.8	94.2
OpenShift On-Premises	3	93.7	97.2	92.1	94.3
Hybrid (AWS + OpenShift)	5	97.1	95.8	96.4	96.4
Mean	-	95.7	94.8	94.4	94.9

Table 1, HIPAA compliance levels reached over the different deployment configurations studied. According to the data, hybrid deployments with OpenShift platforms on AWS showed the greatest overall compliance rates at 96.4%, surpassing both the single AWS and OpenShift deployments. Amongst all configurations, administrative safeguards performed the best while hybrid deployments exhibited the best compliance levels at 97.1% level of compliance. In particular, OpenShift on-premises implementations had a generation positive score of 97.2% for physical safeguards due to organizations seeing substantial control over dedicated infrastructure (Chen et al., 2020).

Table 2: Microservices Security Control Implementation Frequency (n=12)

Security Control	Fully Implemented (%)	Partially Implemented (%)	Not Implemented (%)
Mutual TLS (mTLS)	83.3	16.7	0.0
API Gateway Authentication	100.0	0.0	0.0
Container Image Scanning	75.0	25.0	0.0
Secrets Management	91.7	8.3	0.0
Network Policies	66.7	33.3	0.0
Audit Logging	100.0	0.0	0.0

The Table 2 shows the frequent implementation of essential security controls found within each of the microservices architectures analyzed. Integration of API gateway authentication and complete audit logging attained ubiquity, signifying that these controls are now viewed as fundamental table stakes across the organizations. Implementation of Mutual TLS (mTLS) was solid, achieving 83.3% full implementation and remaining organizations still using some form of partial encryption configurations. Full implementation was only achieved by 66.7% of respondents for network policy, exhibiting the most variability and indicating likely challenges in the configuration of micro segmentation in complicated service topologies (Calcote & Butcher, 2019).

Table 3: Prescription Processing System Performance Metrics (Mean Values)

Performance Metric	AWS Native	OpenShift On-Premises	Hybrid	Industry Benchmark
API Response Time (ms)	142	168	151	<200
Transaction Throughput (TPS)	847	712	923	>500
System Availability (%)	99.94	99.91	99.97	>99.9
Encryption Overhead (%)	8.2	7.1	7.8	<15
Auto-scaling Response (sec)	34	58	41	<60

Performance metrics from application deployments of a prescription processing system across the three configuration types are shown in Table 3. Across measured dimensions, all configurations met or exceeded industry benchmark requirements. The deployments of Hybrid saw better transaction throughput at 923 TPS, a feature of distributing the workload across platforms. Thanks to the elasticity of the cloud, the AWS native deployments had the fastest auto-scaling response time at just 34 seconds. We show that the encryption overhead did not exceed 8.2% across the configurations and thus a full security implementation does not have to come at such a cost to performance (Dragoni et al., 2017).

Table 4: Security Incident and Vulnerability Metrics (12-Month Period)

Metric	AWS Native	OpenShift On-Premises	Hybrid	Total
Critical Vulnerabilities Detected	12	8	15	35

Mean Time to Remediation (days)	3.2	4.7	2.9	3.6
Security Incidents Reported	2	1	2	5
Successful Breach Attempts	0	0	0	0
Compliance Audit Findings	7	5	4	16

Table 4 records security incidents and vulnerability management during a 12-month observation period. Overall, all orgs had 35 high severity weaknesses identified, and none of those lead to a successful breach attempt, proving the efficacy of defense-in-depth implementations. Hybrid deployments delivered the fastest mean remediation time at 2.9 days while managing the highest volume of vulnerabilities to remediate, likely reflecting a higher degree of maturity in security operations processes. As multi-platform expertise in hybrid configurations relates to a better maintenance of compliance, the hybrid configurations had the lowest compliance audit findings 4 findings (Sultan et al., 2019).

Table 5: Protected Health Information Access Control Metrics

Access Control Dimension	Implementation Rate (%)	Audit Coverage (%)	Policy Violation Rate (%)
Role-Based Access Control	100.0	100.0	1.2
Multi-Factor Authentication	91.7	100.0	0.4
Privileged Access Management	83.3	91.7	2.1
Break-Glass Procedures	75.0	100.0	0.8
Third-Party Access Controls	66.7	83.3	3.4

Table 5 fully implemented role-based access control, it is the bedrock of HIPAA compliance with 100% of user actions covered by audit trails. Policy violations were low, with strong adoption of multi-factor authentication at 91.7%. The weakest implementation at 66.7%, and highest policy violation rate at 3.4%, were for third-party access controls, which reflect difficulties in encrypting external integrations common in prescription processing workflows (McGraw, 2019).

Table 6: Container Orchestration Security Configuration Analysis

Security Configuration	OpenShift Implementation (%)	AWS EKS Implementation (%)	Compliance Requirement Mapping
Pod Security Policies	88.9	71.4	Technical Safeguards
Network Segmentation	77.8	85.7	Technical Safeguards
Runtime Security Monitoring	66.7	57.1	Administrative Safeguards
Immutable Infrastructure	100.0	85.7	Technical Safeguards
Encrypted Persistent Volumes	100.0	100.0	Technical Safeguards

Table 6 Container Orchestration Security Settings in OpenShift and AWS EKS Implementations Both platforms realized ubiquitous encrypted persistent volumes, meeting a HIPAA specification for data-at-rest encryption. OpenShift proved to have higher pod security policy implementation at 88.9% than AWS EKS at 71.4%, likely due to OpenShift's default security configurations being less permissive. Over three-quarters of groups have implemented immutable infrastructure practices, with OpenShift attaining full implementation to aid compliance demands to configuration management and change regulate requirements (Shu et al., 2017).

5. Discussion

The results from this study strongly support the use of microservices architecture in HIPAA-compliant prescription processing systems. This means that microservices can meet the strict requirements of compliance in healthcare as represented by the overall compliance achievement rate of 94.9percent across all organizations examined. This result is consistent with and expands the work of Alonso et al. (2018), who reported the benefits of microservices for clinical systems but did not assess compliance comprehensively. This is particularly important when you consider the fact that hybrid deployments beat the average across all of these metrics. The highest compliance rates, shortest time-to-remediation, and strongest transaction throughput were achieved by organizations that used both the AWS and OpenShift platforms. This behavior bolsters the multi-platform strategies that organizations take, using the best of both worlds, taking advantage of AWS managed services for scalability and OpenShift appliance + enhanced security defaults for sensitive workloads. From a practical perspective for healthcare organizations, this suggests that hybrid approaches might yield the best results for critical, high-stakes applications of prescription processing, despite increasing architectural complexity.

Patterns of security control implementation indicated aspects of promising adoption, but also areas where challenges remain for attention. In other words, it shows that proper API gateway authentication and audit logging has become a necessity everywhere. Yet, while network policy implementation peaks at 66.7% full adoption, third-party access controls are

at similar levels indicating challenges translating security needs into the operational sphere of the organization. These results parallel fears made by Di Francesco et al. Source: Surge/Figure: from (2019) concern identifying the operational complexity of microservices expanded to security con figuration management Analysis of performance metrics targets one of the main challenges in any security-compliant system, which is security overhead. The end-to-end encryption overhead not exceeding 8.5% in any configuration and response times within benchmarks, shows that deep security doesn´t have to compromise usability or responsiveness. This discovery has notable implications for health systems where full encryption cannot be deployed due to performance concerns. Where data protection has historically impacted the performance of applications, modern cryptographic implementations and hardware acceleration have materially reduced that overhead.

Container security configurations exposed the strengths of various platforms, helping drive decisions on where to deploy them. The adoption of stricter pod security policies in OpenShift speaks to the opinionated nature of the platform when it comes to container security secure defaults set in OpenShift require explicit modification not explicit configuration. Meanwhile, there were areas where AWS EKS was more advantageous, like network segmentation implementation, allowing for tighter network segmentation due to integration with AWS VPC and security group capabilities. Healthcare Organizations should gauge these characteristics of a platform against their particular culture of security and operational models. And, in the case of 35 critical vulnerabilities detected, zero successful breaches is proof of solid defense-in-depth. Redundant layers of security from segmentation, to mutual TLS, to runtime monitoring, to access management that protected the system over so many layers against any potential exploitation of those vulnerabilities. This reinforces an architectural principle that microservices security should not rely on any individual control complete layered defenses should provide holistic security. This applies most directly to prescription processing domain since PHI and controlled substance data are High-value data.

Practitioners should pay particular attention to third-party access control challenges identified in Table 5. Because prescription processing systems have to talk to a whole slew of other external entities (pharmacy benefit managers, drug databases, state prescription monitoring programs, and pharmacy networks, among others), many of their security challenges will sound familiar. These integrations need to be managed in a HIPAA-compliant manner through careful API design, management of credentials, and monitoring capabilities. In the end, organizations should treat third-party access governance as a separate security domain and not as an extension of internal access control frameworks for external integrations. The limitations of this research include the small sample size of 12 organizations, which limits the statistical generalizability. Further, self-report data on compliance may reflect a half-full glass version of reality, but external audit findings confirm that self-reporting of compliance is occurring with some frequency. Future studies should involve larger samples from different healthcare settings and use the assessments from a third party to quantify compliance more objectively.

6. Conclusion

The research here presented shows that the proposed architecture of secure microservices can be a suitable and efficient option to be adopted on AWS and OpenShift for prescription processing systems to be HIPAA compliant. AMOS data showed organizations can achieve 94.9% overall HIPAA compliance rates with properly configured microservices, making it clear that architectural modernization does not have to come at the expense of regulatory compliance. Hybrid deployments featuring a combination of both cloud and container platforms proved especially potent, achieving the lowest compliance rates with the best performance attributes. Continuous monitoring alone cannot ensure security; however, the detection of vulnerabilities was mitigated by the presence of many levels of defense-in-depth security implementations (e.g., mutual TLS, rigorous access controls, and continuous monitoring itself). However, microservices adoption in a healthcare organization should be done with systematic attention to security architecture using built-in capabilities offered by such platforms while implementing access control level as well as audit logging at the microservice level. This study adds empirical backing for healthcare technology chiefs rationalizing strategic choices to modernize prescription processing systems by demonstrating cloud-native microservices architecture to be operationally viable and regulatory compliant.

References

[1] Kumar, R. P. (2023). Smart healthcare monitoring system for elderly care (US Patent No. 11223344). Registered with the United States Patent and Trademark Office, Class 10-02, granted in June 2023.

[2] Alonso, S. G., de la Torre Díez, I., Rodrigues, J. J., Hamrioui, S., & López-Coronado, M. (2018). A systematic review of microservice architectures in healthcare. Journal of Medical Systems, 42(8), 149. https://doi.org/10.1007/s10916-018-0997-0

[3] Gunda, S. K. (2023). Machine learning approaches for software fault diagnosis: Evaluating decision tree and KNN models. 2023 Global Conference on Communications and Information Technologies (GCCIT), Bangalore, India, 1–5. https://doi.org/10.1109/GCCIT63234.2023.10861953

[4] Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. IEEE Software, 33(3), 42-52. https://doi.org/10.1109/MS.2016.64

[5] Calcote, L., & Butcher, Z. (2019). Istio: Up and running: Using a service mesh to connect, secure, control, and observe. O'Reilly Media.

[6] Chen, Y., Paxson, V., & Katz, R. H. (2020). What's new about cloud computing security? IEEE Security & Privacy, 18(4), 15-23. https://doi.org/10.1109/MSEC.2020.2981897

[7] Gunda, S. K. (2023). Analyzing machine learning techniques for software defect prediction: A comprehensive performance comparison. 2023 Asian Conference on Intelligent Technologies (ACOIT), Kolar, India, 1–5. https://doi.org/10.1109/ACOIT62457.2023.10939610

[8] Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150, 77-97. https://doi.org/10.1016/j.jss.2019.01.001

[9] Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. In Present and Ulterior Software Engineering (pp. 195-216). Springer. https://doi.org/10.1007/978-3-319-67425-4_12

[10] Esposito, C., Castiglione, A., Choo, K. K. R., & Martini, B. (2016). Cloud manufacturing: Security, privacy, and forensic concerns. IEEE Cloud Computing, 3(4), 16-22. https://doi.org/10.1109/MCC.2016.79

[11] Gunda, S. K. (2023). Enhancing software fault prediction with machine learning: A comparative study on the PC1 dataset. 2023 Global Conference on Communications and Information Technologies (GCCIT), Bangalore, India, 1–4. https://doi.org/10.1109/GCCIT63234.2023.10862351

[12] Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., & Tilkov, S. (2018). Microservices: The journey so far and challenges ahead. IEEE Software, 35(3), 24-35. https://doi.org/10.1109/MS.2018.2141039

[13] Kratzke, N., & Quint, P. C. (2017). Understanding cloud-native applications after 10 years of cloud computing. Journal of Systems and Software, 126, 1-16. https://doi.org/10.1016/j.jss.2017.01.001

[14] Gunda, S. K. (2023). Fault prediction unveiled: Analyzing the effectiveness of random forest, logistic regression, and k-neighbors. 2023 2nd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 107–113. https://doi.org/10.1109/ICSSAS64001.2023.10760620

[15] Lewis, J., & Fowler, M. (2014). Microservices: A definition of this new architectural term. MartinFowler.com. https://martinfowler.com/articles/microservices.html

[16] Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., & Ramoni, R. B. (2016). SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. Journal of the American Medical Informatics Association, 23(5), 899-908. https://doi.org/10.1093/jamia/ocv189

[17] Gunda, S. K. (2023). Software defect prediction using advanced ensemble techniques: A focus on boosting and voting methods. 2023 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 157–161. https://doi.org/10.1109/ICESIC61777.2023.10846550

[18] McGraw, D. (2019). Building public trust in uses of Health Insurance Portability and Accountability Act de-identified data. Journal of the American Medical Informatics Association, 20(1), 29-34. https://doi.org/10.1136/amiajnl-2012-000936

[19] Newman, S. (2021). Building microservices: Designing fine-grained systems (2nd ed.). O'Reilly Media.

[20] Gunda SK, Yettapu SDR, Bodakunti S, Bikki SB. Decision Intelligence Methodology for AI-Driven Agile Software Lifecycle Governance and Architecture-Centered Project Management, 2023 Mar. 30;4(1):102-8. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P112

[21] Shu, R., Gu, X., & Enck, W. (2017). A study of security vulnerabilities on Docker Hub. In Proceedings of the Seventh ACM Conference on Data and Application Security and Privacy (pp. 269-280). ACM. https://doi.org/10.1145/3029806.3029832

[22] Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container security: Issues, challenges, and the road ahead. IEEE Access, 7, 52976-52996. https://doi.org/10.1109/ACCESS.2019.2911732

[23] Gunda, S. K. G. (2023). The Future of Software Development and the Expanding Role of ML Models. International Journal of Emerging Research in Engineering and Technology, 4(2), 126-129. https://doi.org/10.63282/3050-922X.IJERET-V4I2P113