**Library Progress International**
*Vol.44 No. 3, July-December 2024:*
*P. 25718-25723*

*Print version ISSN 0970 1052*
*Online version ISSN 2320 317X*

*Original Article*

*Available online at www.bpasjournals.com*

# Cloud-Native Architecture Observability and Compliance Challenges: A Comprehensive Reference Architecture Approach

**Arun Pandiyan Perumal**

Illinois Institute of Technology
Fremont, California, USA.
Email: apandiyan@hawk.iit.edu

**How to cite this article:** Arun Pandiyan Perumal (2024) Cloud-Native Architecture Observability and Compliance Challenges: A Comprehensive Reference Architecture Approach. *Library Progress International,* 44(3), 25718-25723

**Abstract**
In the context of contemporary software development, the desire for scalability and resilience has become of the utmost importance, particularly considering the explosion in the number of online services and applications. In the context of cloud computing, the supply of hosted services, which may include software, hardware, and storage, is referred to as cloud computing. As a result of the benefits that cloud computing offers, which include rapid deployment, flexibility, minimal upfront costs, and scalability, numerous enterprises of varying sizes have adopted cloud computing. Cloud-native applications (CNAs) have emerged as a potential game-changer in the fight against these difficulties. These applications provide dynamic scalability and robust resilience by utilizing unique architectural techniques. Observing and monitoring these applications might be difficult, particularly for a certified nursing assistant who is constrained by compliance regulations. The investigation's chief goal is to research a comprehensive reference architecture approach for CNA observability along with compliance challenges from a research perspective. This research indicates that it is of the utmost importance to implement a multi-layered observability strategy that incorporates metrics, logs, and traces. This will guarantee that all micro-services and components are adequately visible to the user. This study provides a systematic method for improving observability and assuring compliance in cloud-native architectures. This technique enables businesses to achieve operational efficiency and regulatory compliance in environments that are complex and scattered throughout the world.

*Keywords— Cloud Computing, Cloud-native applications, Observability, Cloud Monitoring, Cloud Compliance, Reference Architecture.*

## Introduction

CNA refers to a contemporary methodology for designing and constructing applications that effectively utilize the complete functionalities of cloud computing environments [1]. In contrast to conventional monolithic architectures, cloud-native systems consist of loosely interconnected microservices that are implemented in containers and controlled using orchestration tools such as Kubernetes. This architectural design facilitates fast development, scalability, and resilience, but it also brings intricate challenges in the management, monitoring, and assurance of compliance across distributed systems [1].

Cloud-native technologies include a collection of algorithms, procedures, and tools that enhance the efficiency of developing, deploying, and managing applications in cloud settings [2]. Core to CNA is the construction of applications as collections of loosely connected, independently deployable services that make use of cloud-native infrastructure and services [3]. According to Alnafessah et al. [4], the concepts encompassed in this context are containerization, microservices architecture, declarative APIs, continuous integration and delivery (CI/CD), and infrastructure as code (IaC). Cloud-native technologies empower enterprises to leverage the scalability, agility, and resilience provided by cloud platforms to offer cutting-edge and dependable software solutions [1].

Comprehensive consideration of observability and compliance is essential in cloud-native systems. Observability refers to the capacity to observe, track, and record actions throughout the architecture, so offering a profound understanding of system operational efficiency, malfunctions, and user engagements. Compliance, conversely, guarantees that the cloud-native systems conform to regulatory obligations, security criteria, and optimal methodologies. The reconciliation of observability and compliance in a cloud-native environment is a challenging task that necessitates a well-designed reference architecture to tackle these intricacies [5] successfully.

The following section elaborates on previous research concerning the observability and compliance challenges of cloud-native architecture. The study by Marie-Magdelaine (2021) [6] noted that Cloud Computing and Cloud-Native technologies underpin the Internet. Cloud apps are currently used daily by users and enterprises. Outages and degradation of the quality of service can devastate society. Web applications are now complicated distributed systems that are hard to understand and administer, making them more prone to failure if not appropriately managed. Therefore, it's crucial to understand, observe, avoid, detect, and fix failure-causing factors. This cited study presented an Observability framework for cloud-native systems in this thesis. An autonomic computing-based architecture for Observability-driven auto-scaling in Cloud-Native settings is also proposed. This approach lets us link auto-scaling to the application workload. This cited study also demonstrates the benefits and potential of cloud-native design and concepts for the IoT. This cited research proposed using Software-Defined Networking and Radio to create generic Internet of Things devices that are adaptable and reconfigurable.

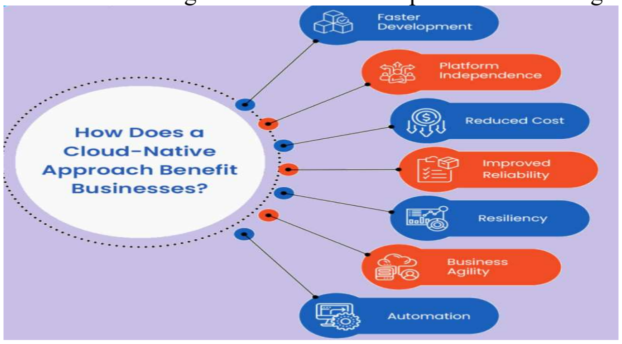

Fig. 1. An overview of the benefits of Cloud Native Applications

As per Olabanji (2022) [7], the study combined the Theory of Planned Behaviour (TPB) and the Technology Adoption Model (TAM) to assess the enterprise's intention to adopt cloud-native architecture. A questionnaire is utilized to collect data, which is analyzed using descriptive and multivariate statistical techniques. This investigation revealed that enterprise decisions to port or migrate to CNA are influenced by design complexity, technical competence, and attitude toward new ideas and technology. The data analysis was utilized to enhance the decision-support framework by adding essential elements for enterprises to consider before, during, and after CNA implementation. The framework includes cloud deployment models as a decision or task for enterprises to consider, as they are crucial for cloud-native adoption and often overlooked in decision-supporting tools.

Cybersecurity relies on identity and access management, according to Subburaman & Chandrasekaran (2022) [8]. Techniques, methods, and regulations manage identity access to digital resources and the extent of permission. Every week, a new cybercrime or data breach is reported. Poor security, software faults, human error, malicious insiders, and privilege misuse cause many data breaches. AI can improve access control. Artificial intelligence in IAM research is needed to strengthen authentication and access control to reduce cyber risks and other IAM issues. This study examines the relationship between AMIS and AI in identity and access management, including privilege monitoring, administration, and control. A binary categorization approach for security access control transforms the PDP problem into a yes/no question. Supervised machine learning is used to build a vector decision classifier for a distributed, effective, and accurate policy decision point (PDP). The Kaggle-Amazon access control policy dataset compared the proposed method to existing research standards for performance, duration, and flexibility. The suggested technique achieves excellent access control secrecy because the PDP is not in direct touch with the Point Administration Point (PAP). Finally, PDP-based ML can handle enormous access requests, execute multiple main rules simultaneously, and have a 95% accuracy rate without policy conflicts in 0.15 s. Access control can be more responsive, flexible, dynamic, and scattered to improve security.

The authors Kosińska et al. (2023) [9] described the Cloud-native paradigm as a way to design and deliver applications using DevOps concepts, Continuous Integration/Continuous Delivery, containers, and microservices, enhancing the Twelve-Factor patterns. Such applications are complicated, but observability can help. A Systematic Mapping Study (SMS) on Cloud-native application observability is presented. This referenced study compiled research conducted between the years 2018 and 2022. The comparison criteria were used to examine, compare, and classify the selected studies. The SMS assessed engineering approaches, maturity, and efficiency of observability by considering four research points: motivations for equipping Cloud-native applications with observability capabilities, literature research areas, implemented observability approaches, and future trends.

The authors Lichtenthäler & Wirtz (2024) [10] observed that cloud computing interest is rising, and services are changing owing to technical advancement. Therefore, cloud-native refers to creating apps that maximize the benefits of current cloud computing. However, cloud-native is a vast topic, and cloud computing solutions vary. This cited study needs to enable developers and software architects who want to use cloud-native concepts. This cited study presented a quality model for cloud-native software architectures that shows how architectural factors affect quality. The goal is to evaluate application architecture models based on cloud-native characteristics and quality throughout design time. This paper describes the methodology for creating and validating the quality model for cloud-native software architectures and its current state. This talk draws from earlier research, particularly a validation survey on architectural traits and quality. This paper integrated methodology into a larger philosophical and methodological context. The findings give a qualitative overview of cloud-native software architectures and lay the groundwork for quantitative quality evaluations using architectural models of applications.

Research Gap: Although cloud-native architectures are increasingly being adopted, there is still a notable research gap in creating a comprehensive and scalable reference architecture that effectively tackles both observability and compliance demands. Prior research frequently examines these elements separately, disregarding the intricacies of combining observability tools (monitoring, logging, tracing) with compliance concerns (security, regulatory conformance) in dynamic, microservices-based setups. In highly regulated industries where data security and privacy are of utmost importance, this gap underscores the need for a comprehensive framework that not only enables real-time insights and visibility into cloud-native systems but also guarantees automated, consistent enforcement of compliance policies.

## *METHODOLOGY*

Using secondary data collecting, this study applies a qualitative research technique to investigate the observability and compliance issues in cloud-native architecture. The study entails a thorough examination and evaluation of scholarly articles, industry reports, and case studies published during the last six to seven years. The main areas of attention include cloud-native architectures, monitoring services, distributed tracing, security compliance, and regulatory frameworks.

The study intends to identify current gaps, emerging trends, and best practices in integrating observability and compliance within cloud-native settings by synthesizing findings from several research sources. Systematic analysis of the gathered data is conducted to create a comprehensive reference architectural strategy that effectively tackles these two difficulties holistically, guaranteeing both operational visibility and compliance with regulations.

## *RESULTS AND DISCUSSIONS*

Cloud Native Applications are highly suitable for development and deployment as enterprise applications. Its adaptable features render it a viable choice for extensive software development. Functional and non-functional requirements, including Service Level Agreements (SLAs) and compliance requirements, govern the operation of enterprise systems. In sectors subject to regulation, such as healthcare and finance, applications are required to fulfil specific expectations and comply with audit criteria [11, 5]. CNAs working in regulated industries must have a proactive tact to monitor. Conventional reactive monitoring methods often implemented in traditional utilisations are unable to provide the desired amount of transparency that regulated businesses anticipate from CNAs. Therefore, a novel method for monitoring computational neural networks is emerging [5].

Implementing observability in CNA is a comprehensive strategy that includes monitoring, logging, and tracing to gain a profound understanding of system activity. Cloud-native systems, being inherently dispersed, present increased complexity in terms of observability owing to the dynamic characteristics of microservices, containers, and orchestrators [12]. The primary objective of monitoring is to gather and analyse metrics pertaining to application performance, resource usage, and network traffic in real time. Effective identification of performance bottlenecks, outages, and anomalies is of utmost importance. Web applications such as Prometheus and Grafana are widely used in cloud-native systems for the purpose of collecting and visualizing metrics [9]. Logging is a crucial element of observability, as it allows for the comprehensive documentation of events that take place within the system. Logs from several microservices in cloud-native settings must be consolidated to offer a cohesive perspective, making solutions such as the Elasticsearch, Logstash, and Kibana (ELK) stack indispensable. Effective tracing is crucial for comprehending the sequence of transactions among several microservices [14]. Distributed tracing technologies like Jaeger or Zipkin facilitate the mapping of requests as they move through different services, enabling the identification of latency problems and the precise determination of the underlying cause of errors. Nevertheless, the task of attaining thorough observability in cloud-native architectures is difficult because of the highly dynamic and transient characteristics of microservices, container orchestration, and continuous upgrades [12].

1.1 *Challenges in Observability*:

Although observability is crucial in cloud-native design, the intricate and distributed nature of these environments gives rise to various problems. An inherent obstacle lies in the decentralized structure of microservices, which poses difficulties in tracking and overseeing transactions as they traverse several services, containers, and nodes [9]. The intricate nature of this intricacy often leads to a limited understanding of the data flow inside the system, therefore complicating the identification, diagnosis, and resolution of problems in real time. Scalability poses a notable obstacle since the amount of telemetry data, including metrics, logs, and traces, produced in a cloud-native system of considerable scale can be daunting. Scaling observability tools to manage this data without compromising performance or incurring exorbitant expenses is a challenging endeavor [15]. Tool integration is a significant obstacle in cloud-native systems, which often encompass a wide range of tools and technologies for observability. Coordinating the integration of various instruments to function harmoniously while guaranteeing uniform data gathering, storage, and analysis can be exceedingly tricky. Furthermore, the security issues pertaining to confidential information in logs and traces need meticulous attention to prevent possible breaches of compliance. Cloud components in a CNA are of such magnitude that they amount to hundreds of components. The observed observability statistics for these components indicate significant levels of volume, velocity, value, and diversity. The classification of observability data as a Big Data problem implies that a Big Data solution should be explored for data observability's storage and analysis [5].

*1.2 Compliance Requirements in Cloud-Native Architecture*

Compliance in cloud-native architectures refers to the process of verifying that apps adhere to regulatory obligations, security standards, and industry-leading methodologies. These criteria are particularly crucial in regulated sectors such as finance, healthcare, and government entities, where data protection and security are of the utmost importance. Security compliance encompasses several aspects, such as safeguarding data, implementing encryption, specifying access rules, and ensuring network security. In cloud-native systems, it is imperative to guarantee the encryption of data both when stored and during transmission, establish appropriate access controls, and prevent the exposure of sensitive information in logs or monitoring technology [16]. Software applications such as Amazon Web Services based identity and access management (AWS IAM), Azure Active Directory, and Google Cloud IAM are essential for establishing robust IAM systems that effectively enforce security standards. Regulatory compliance encompasses the strict observance of frameworks such as the General Data Protection Regulation (GDPR), General Data Protection Regulation (HIPAA), Payment Card Industry Data Security Standard (PCI – DSS), and Sarbanes-Oxley Act (SOX), which mandate firms to handle data protection, security, and information transparency proficiently [5]. A CNA must include systems for producing audit logs, implementing data retention policies, and guaranteeing compliance of all microservices with applicable regulations [17]. Furthermore, the task of attaining compliance in cloud-native systems is made more complex by the shared responsibility model between cloud service providers and clients. This necessitates meticulous deliberation on the party accountable for various elements of security and compliance [9].

*1.3 Reference Architecture Approach and Implementation Approach*

Implementing a thorough reference architectural strategy is crucial for tackling the issues of observability and compliance in cloud-native systems. The conceptualization of this reference architecture should be grounded on fundamental design concepts such as modularity, automation, and robustness. The importance of modularity in cloud-native design lies in its ability to separate observability and compliance obligations into distinct modules, therefore facilitating the independent management and scaling of these features. Another fundamental aspect is automation, which utilizes Infrastructure as Code (IaC) technologies like Terraform, AWS Cloud Formation, or Azure Resource Manager to automate the deployment and setup of components related to observability and compliance [6]. The implementation of automation in monitoring, logging, and security regulations guarantees their consistent application in all settings, therefore minimizing human error and boosting reliability. The concept of resilience encompasses the implementation of architectural patterns, such as circuit breakers, retries, and failover mechanisms, which serve to enhance the robustness of a system and facilitate observability by establishing distinct points of failure and subsequent recovery. Through the integration of these concepts, businesses can construct a reference architecture that offers a comprehensive structure for observability and compliance [5].

The design and construction of CNAs are grounded in Cloud technology. They employ micro-service designs and APIs for internal interconnections, as well as CI/CD pipelines for the release and deployment processes. CNAs depend on several computing environments, including containers, serverless, and virtual machines (VMs), together with automated deployment methods that are based on continuous integration/continuous delivery (CI/CD) pipelines. By leveraging the most sophisticated capabilities of Cloud services, CNAs are highly suitable for developing enterprise applications and large-scale software systems. As the non-functional requirements of software dictate, performance, reliability, and stability are crucial criteria that CSCs must monitor to guarantee that the quality of service provided to the end-users aligns with the ones specified in SLAs. Likewise, professionals in regulated sectors must determine if their CNAs can maintain compliance throughout an audit.

The suggested reference architecture incorporates observability and compliance as integral elements of its components. Also, the reference architecture serves as a tool for Cloud practitioners to develop and deploy CNAs, particularly in industries subject to regulations. This study asserts that incorporating observability along with compliance as inherent elements of our reference architecture dramatically decreases the time required by Cloud practitioners to incorporate external observability along with compliance components into their solutions. The reference architecture suggested is developed by certified Cloud practitioners and researchers who possess extensive knowledge and experience in monitoring large-scale systems and dealing with them on a regular basis. Finally, a specific example of this design, with some adjustments, is central to Observability procedures at IBM Cloud and utilizes artificial intelligence to identify anomalies in almost real-time [5].

The successful implementation of a resilient observability and compliance strategy in cloud-native systems necessitates the integration of optimal methodologies, technologies, and approaches. Infrastructure as Code (IaC) is an essential strategic approach that guarantees the consistent definition, deployment, and management of all infrastructure components, including observability and compliance tools, across all environments. Implementing this approach not only mitigates the possibility of misconfiguration but also facilitates version control and auditing of infrastructure modifications, both of which are crucial for ensuring compliance. Container orchestration platforms like Kubernetes have a vital function in overseeing the operations of microservices and containers. They improve both the ability to observe and comply with regulations by offering inherent features for monitoring, logging, and enforcing policies. Integration of Kubernetes' inherent tools, like Prometheus for monitoring and Open Policy Agent (OPA) for policy enforcement, into the standard design, enables the establishment of a cohesive framework for observability and compliance. Moreover, the implementation of DevSecOps (Development, Security, Operations) methodologies, which involve including security and compliance checks into the CI/CD pipeline, guarantees the early detection and resolution of security vulnerabilities and compliance breaches during the development process.

To prove the efficacy of a complete reference architecture for compliance along with observability, evaluate how organizations have implemented these approaches. A large financial institution that switched to a CNA using Prometheus, ELK stack, and Jaeger for traceability and HashiCorp Vault and Kubernetes-native solutions for compliance and security. The company achieved high transparency in its microservices and met strict financial regulations like PCI-DSS and SOX using this method. Second, a healthcare provider created a CNA with integrated observability and compliance frameworks to meet HIPAA regulations . Automated monitoring, logging, and encryption would help the provider comply and improve system performance.

According to this research, organizations should follow best practices to increase cloud-native deployment observability and compliance. First, establish a multi-faceted observability method using metrics, logs, and traces to ensure complete visibility over all microservices and components. To maintain uniformity and compliance across operations, organizations should use IaC and automation technology. DevSecOps-integrated security and compliance tests in CI/CD pipelines ensure early vulnerability and non-compliance identification. Finally, organizations should regularly evaluate and adopt tools and technology that improve visibility and compliance, such as AI-powered monitoring solutions and advanced policy enforcement approaches.

## CONCLUSION

Through this study, it has been concluded that cloud-native architectures face significant but manageable observability and compliance issues. Operational efficiency and regulatory compliance require a complete reference architectural strategy that blends observability and compliance at every tier. Organizations must deliberately create modular, automated, and resilient cloud-native infrastructures to solve observability and compliance issues holistically. With the correct strategies, tools, and practices, cloud-native architectures can be highly observable and compliant, allowing enterprises to use cloud computing without compromising security or regulations.

## REFERENCES

[1] O. C. Oyeniran, O. T. Modupe, A. A. Otitoola, O. O. Abiona, A. O. Adewusi, and O. J. Oladapo, “A comprehensive review of leveraging cloud-native technologies for scalability and resilience in software development,” Int. J. Sci. Res. Arch., vol. 11, no. 2, pp. 330–337, 2024.

[2] A. Tundo, M. Mobilio, O. Riganelli, and L. Mariani, “Monitoring probe deployment patterns for cloud-native applications: Definition and empirical assessment,” IEEE Trans. Serv. Comput., vol. 17, no. 4, pp. 1636–1654, 2024.

[3] J. Alonso et al., “Understanding the challenges and novel architectural models of multi-

cloud native applications-a systematic literature review," Journal of Cloud Computing, vol. 12, no. 1, 2023.

[4] A. Alnafessah, A. U. Gias, R. Wang, L. Zhu, G. Casale, and A. Filieri, "Quality-aware DevOps research: Where do we stand?" IEEE Access, vol. 9, pp. 44476–44489, 2021.

[5] W. Pourmajidi, L. Zhang, J. Steinbacher, T. Erwin, and A. Miranskyy, "A reference architecture for observability and compliance of cloud Native Applications," arXiv [cs.SE], 2023.

[6] N. Marie-Magdelaine, Observability and resources managements in cloud-native environnements (Doctoral dissertation). 2021.

[7] D. O. Olabanji, Towards the development of a decision framework for portability in cloud-native architecture deployment (Doctoral dissertation). 2022.

[8] S. P. Subburaman and S. Chandrasekaran, "Traditional Techniques and Emerging Technologies in Observability," Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-255, vol. 238, no. 1, pp. 2–4, 2022.

[9] J. Kosińska, B. Baliś, M. Konieczny, M. Malawski, and S. Zieliński, "Toward the observability of cloud-native applications: The overview of the state-of-the-art," IEEE Access, vol. 11, pp. 73036–73052, 2023.

[10] R. Lichtenthäler and G. Wirtz, "Formulating a quality model for cloud-native software architectures: conceptual and methodological considerations," Cluster Comput., vol. 27, no. 4, pp. 4077–4093, 2024.

[11] T. Laszewski, K. Arora, E. Farr, and P. Zonooz, Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud. Packt Publishing Ltd, 2018.

[12] M. C. Borges, J. Bauer, S. Werner, M. Gebauer, and S. Tai, "Informed and assessable observability design decisions in cloud-native microservice applications," in 2024 IEEE 21st International Conference on Software Architecture (ICSA), 2024.

[13] A. Widerberg and E. Johansson, Observability of Cloud Native Systems: An industrial case study of system comprehension with Prometheus & knowledge transfer. 2021.

[14] N. Kratzke, "Cloud-native observability: The many-faceted benefits of structured and unified logging—A multi-case study," Future Internet, vol. 14, no. 10, p. 274, 2022.

[15] B. Sharma and D. Nadig, "EBPF-enhanced complete observability solution for cloud-native microservices," in ICC 2024 - IEEE International Conference on Communications, 2024, vol. 46, pp. 1980–1985.

[16] B. Nascimento, R. Santos, J. Henriques, M. V. Bernardo, and F. Caldeira, "Availability, scalability, and security in the migration from container-based to cloud-native applications," Computers, vol. 13, no. 8, p. 192, 2024.

[17] T. Theodoropoulos et al., "Security in cloud-native services: A survey," J. Cybersecur. Priv., vol. 3, no. 4, pp. 758–793, 2023.