Journal of Distributed Computing and Systems(JDCS)
Vol.7, Issue 2, 2025, Pages 86-95

A Novel Architecture for Monitoring and Evaluating Cloud Services Based on Indicator Extraction

Davood Maleki[1*], Neda Ghorbani[2], Ehsan Arianyan[3], Masoud Beiklaryan[4]
Information Technology faculty, ICT Research Institute, Tehran, Iran
[1]*dmaleki@itrc.ac.ir*, [2]*n.ghorbani@itrc.ac.ir*, [3]*ehsan_arianyan@itrc.ac.ir*, [4]biklaryan@ito.gov.ir

Article History:
Received: 09 September 2024
Received in revised form: 20 February 2025
Accepted: 11 March 2025
Available online: 19 March 2025

Abstract

With the rapid expansion of cloud services and the increasing complexity of its various layers, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), monitoring and evaluating the performance of these services has become a significant challenge. This paper introduces a novel architecture for cloud service monitoring, leveraging multi-layer service monitoring through index extraction and performance analysis. The proposed architecture, utilizing a hybrid approach of data extraction and index analysis, can simulate the performance status, scalability, and service quality at each layer. This approach enables cloud service administrators to identify performance issues, potential threats, and scalability deficiencies, thereby effectively improving service quality. The proposed architecture specifically addresses scalability and performance requirements in large-scale cloud environments. Furthermore, this paper discusses the potential challenges and barriers in implementing large-scale monitoring architectures and provides practical and actionable solutions to tackle these issues within the proposed architecture. Additionally, a comparison with existing traditional architectures reveals that the proposed architecture is significantly superior, particularly in terms of security, scalability, integrity, and adaptability. Finally, this study highlights the importance of future research on the impact of distributed and decentralized storage systems, such as Blockchain technology, on the security and scalability of data warehouses in global cloud monitoring environments.

Keywords: Indicator Extraction, Service Performance Improvement, Service Quality, Cloud Service Monitoring, Scalability, IaaS, PaaS, SaaS.

I. Introduction

With the rapid expansion of cloud services at various organizational and commercial levels, challenges related to monitoring, managing, and improving the performance of these services have become key issues in the field of information technology.

Cloud services are categorized into three main models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), each of which has its own unique features and requirements [1]. These models provide scalable and flexible resources to users, offering numerous benefits while introducing challenges in monitoring and security [2]. Precise and effective monitoring of these services, especially across different cloud layers, is essential to properly evaluate performance, scalability, and service quality.

Various methods have been proposed for monitoring cloud services, most of which focus on specific aspects such as security [3] or performance analysis [4]. However, there remains a lack of comprehensive and integrated approaches for simultaneously monitoring all layers of cloud services at scale. This paper introduces a proposed architecture for a Cloud Services Monitoring Center, aiming to optimize the supervision of IaaS, PaaS, and SaaS layers. The architecture leverages indicator extraction methods to evaluate performance, scalability, and service quality.

This research particularly emphasizes extracting performance indicators across each cloud layer, facilitating the simulation and identification of performance issues and potential threats. The primary goal of this study is to provide a comprehensive and scalable framework that enables cloud service managers to identify and manage issues related to efficiency, scalability, and potential threats. The paper also explores the challenges and barriers to implement this architecture in real-world cloud environments and proposes innovative solutions to enhance the quality and efficiency of cloud services.

The remainder of the paper is structured as follows:

- Section 2 discusses related work.
- Section 3 introduces the methodology and presents the monitoring indicators for the IaaS, PaaS, and SaaS layers across three domains: technical, security, and business.

- Section 4 examines the proposed monitoring center architecture.
- Section 5 reviews the challenges of standard monitoring architectures and the solutions and advantages of the proposed architecture.
- Finally, Section 6 provides conclusions, evaluates the final results, and suggests directions for future research.

II. RELATED WORK

Cloud services, as one of the key pillars of information technology in recent decades, have rapidly expanded. Numerous studies have been conducted to monitor and evaluate the performance of these services, particularly at the infrastructure, platform, and software layers.

One of the early studies in this field, "A Framework for Resource Management in Dynamic Cloud Environments," focused on the importance of scalability and reliability in cloud infrastructures. While this framework addressed the need for scalability and adaptability to changing user demands, it did not provide sufficient attention to monitoring the various service layers in detail [5].

Another study examined fundamental challenges in cloud services, including latency, security, and service quality. It emphasized that performance evaluation should be based on key quality of service (QoS) indicators. However, the study did not propose adequate tools for continuous and automated monitoring of these indicators [6].

In the domain of monitoring architectures, a multi-layered monitoring system for cloud services was proposed, utilizing machine learning techniques to analyze collected data. Although this approach proved efficient, its complexity in large-scale environments posed challenges and required further optimization [7].

Another study focused on developing a data-driven monitoring architecture for cloud services. This research suggested that data monitoring should be conducted through distributed and scalable architectures. While this approach was successfully applied to SaaS services, its capabilities for PaaS and IaaS layers were limited [8].

These studies highlight the need for a comprehensive architecture capable of simultaneously monitoring all layers of cloud services and analyzing performance indicators. This need is particularly pronounced in large-scale cloud environments, where scalability and flexibility are critical. The proposed architecture in this paper addresses these challenges by emphasizing intelligent and hybrid monitoring capabilities.

III. METHODOLOGY

To develop a cloud services monitoring center, the process begins with identifying data sources, primarily the cloud service providers. Next, for monitoring cloud services across the three layers of IaaS, PaaS, and SaaS, key indicators that can be monitored online in three domains—technical, security, and business—were identified for each layer, as shown in Table 1. Various methods for extracting these monitoring indicators were examined, including agent-based, API-based, event-based, AI and machine learning-based, log-based, telemetry-based, container-based, and network traffic-based methods.

In a hybrid approach, the extraction methods were mapped to the respective cloud service layers:

- SaaS layer: agent-based and container-based methods.
- PaaS layer: API-based, event-based, and AI/machine learning-based methods.
- IaaS layer: network traffic-based, telemetry-based, and log-based methods [9].

For each indicator, the appropriate extraction method was determined. Based on the selected extraction methods, tools for data extraction from cloud service layers were identified. The extracted data, which includes time-series data, log and text data, and analytical data, was then categorized for storage in three types of databases.

An ETL (Extract, Transform Load) unit was designed to process the extracted data by transforming, cleansing, and reformatting it for storage in the designated databases. Once stored, the data is transferred to a data warehouse for backup and preparation for further analysis and processing.

Analytical tools, such as OLAP Engines, were used for data analysis, enabling the transition of data from the data warehouse to the analysis layer. After processing, the results are utilized for reporting or generating alerts.

Following the data analysis layer, an alerting layer was incorporated. This layer sends alerts to notification systems when issues or specific conditions are detected during data analysis.

Finally, the reporting and dashboard layer displays the analyzed results as reports and dashboards for end-users.

The proposed architecture is illustrated in Figure 1, outlining the flow from data extraction to analysis, alerting, and reporting.

A. Monitoring Indicators:

Key monitoring indicators for each of the cloud service layers (IaaS, PaaS, and SaaS) have been identified across three domains: technical, security, and business. These indicators are designed to be monitored online and provide real-time insights into the performance, scalability, and quality of services which is shown in Table 1.

B. Extraction Methods

Table 2 summarizes methods reported in previous studies in terms of data accuracy and response time indicators and various service layers [34]. The accuracy of the data includes the accuracy of the extracted data and the absence of errors. Response Time includes delay time and real time. Real time is the ability of the method to provide data simultaneously and without delay [35].

Table 1. Monitoring indicators for different layers of cloud services

Service layer	Area	Indicators	index definition	Ext-method
IaaS	Technical	Utilization of Processing Resources [10]	1. CPU and Memory Utilization: The extent of processor and memory usage for running applications or processing data.	Telemetry-based
		Scalability of Resources [11]	2. System Scalability: The system's ability to scale resources (e.g., CPU and memory) up or down based on demand	Log-based
		System Load Level [12]	3. System Load and User Concurrency: The level of simultaneous pressure and usage by users or applications on the system	Network traffic-Based
	Business	Compliance with Security Policies [13]	4. Compliance with Security Policies: Assessment of whether the system adheres to security policies and regulations	Log-based
		Effort for DDoS Attacks Mitigation [14]	5. DDoS Attack Mitigation Efforts: The number or severity of identified attempts to execute Distributed Denial of Service attacks.	Network traffic-Based
		Number of Failed Resource Requests [15]	6. Failed Requests: The frequency of user requests not fulfilled due to technical issues or resource limitations.	Log-based
	Security	Revenue from Services [16]	7. Revenue from Services: The amount of income generated by providing services to customers.	Log-based
		Number of Active Users [17]	8. Active Users: The number of users utilizing services within a specified time frame.	Telemetry-based
		Trend in Service Usage [18]	9. Service Usage Trends: Changes and patterns in service usage over time.	Log-based
PaaS	Technical	API Performance [19]	10. API Performance: Speed and quality of performance for APIs facilitating system communication	API-based
		Processing Time in PaaS Services [20]	11. Processing Time in PaaS Services: The time taken to execute processes or deliver services in a Platform-as-a-Service environment	Event-based
		Service Compatibility with Technical Standards [21]	12. Standards Compliance: The capability of services to meet industry requirements and standards.	API-based
	Business	Number of Incidents Detected via API [22]	13. API Incident Reporting: The number of recorded incidents or errors detected through APIs	API-based
		Data Encryption Level [23]	14. Data Encryption Levels: The percentage or volume of data encrypted during transmission or storage	API-based
		Performance of Security Alert Systems [24]	15. Security Alert Systems: The accuracy and speed of systems in identifying and reporting security threats	Artificial intelligence-based
	Security	Cost Analysis of Services [25]	16. Cost Analysis: Comparison of service provision costs against revenue or performance.	API-based
		Impact of PaaS Services on Business Models [26]	17. Impact on Business Models: The effect of using PaaS on revenue and business processes.	API-based
		Customer Satisfaction Level [27]	18. User Satisfaction: The level of customer satisfaction measured through surveys or feedback	Artificial intelligence-based
SaaS	Technical	Service Quality in SaaS [28]	19. SaaS Reliability and Performance: The reliability, performance, and speed of Software-as-a-Service offerings	Container-based
		Scalability in SaaS Services [11]	20. SaaS Scalability: The ability of SaaS services to handle increases or decreases in user numbers.	Agent-based
		Quality of Communication Between SaaS Services [29]	21. SaaS Collaboration Efficiency: The efficiency and synchronization of SaaS services for collaborative work	Agent-based
	Business	Number of Detected Incidents at SaaS Level [30]	22. SaaS Security Threats: The number of identified issues or security threats within SaaS solutions.	Agent-based
		Compliance with Security Standards [31]	23. Security Compliance: Adherence to specified security requirements across various domains	Agent-based

		Use of Encryption in Data Transmission [32]	24. Secure Data Transmission: The percentage or volume of data securely transmitted.	Container-based
	Security	Customer Needs Analysis [33]	25. User Needs Analysis: Evaluation of customer needs and expectations from services	Agent-based
		Return on Investment Analysis in SaaS [33]	26. SaaS ROI Analysis: Assessment of profitability or success of investments in SaaS services.	Container-based
		Market Competitiveness in SaaS [33]	27. Market Competitiveness: The level of competition among SaaS providers in the market.	Agent-based

According to the results obtained in Table 2, the accuracy and response time, network traffic-Based and log-based and telemetry-based methods is suitable in the infrastructure layer. The accuracy and response time, API-Based, event-based and AI/ML-Based methods are suitable for the substrate in platform layer. The accuracy and response time agent-Based extraction and container-based Extraction Methods are suitable for the software layer.

Table 2. Data Accuracy and Response Time Indicators for Various Methods in Service Layers

Other reasons for choosing the method	Response Time	Data Accuracy	Service layer	Methods for Extracting Cloud Service Metrics
By analyzing network traffic and data flows, indicators such as bandwidth usage, latency, and Quality of Service (QoS) can be extracted.	3 s	*92%*	IaaS	Network Traffic-Based Methods
	7 s	80%	PaaS	
	10 s	75%	SaaS	
System and device logs can provide indicators such as CPU and memory usage, as well as system errors.	3 s	92%	IaaS	Log-Based Methods
	10s	88%	PaaS	
	15 s	86%	SaaS	
Utilizing telemetry data to monitor resource performance and identify indicators such as resource utilization rates and anomaly detection.	1 s	95%	IaaS	Telemetry-Based Methods
	3.5 s	89%	PaaS	
	5 s	88%	SaaS	
Using APIs provided by platforms to directly retrieve data from services and extract indicators such as service usage, response times, and the success rate of requests.	5s	88%	IaaS	API-Based Methods
	2s	94%	PaaS	
	7s	90%	SaaS	
By collecting and analyzing system-recorded events, indicators like the number of error events, successful events, and request response times can be identified.	10 s	80%	IaaS	Event-Based Methods
	1 s	91%	PaaS	
	5s	83%	SaaS	
Analyzing log data and network traffic using machine learning algorithms to identify patterns and predict performance	20 s	85%	IaaS	AI/ML-Based Methods
	4s	93%	PaaS	
	10 s	89%	SaaS	
Using software agents to monitor and collect data related to application performance, such as user response times and the usage of various features.	7s	87%	IaaS	Agent-Based Extraction Methods
	5s	89%	PaaS	
	1 s	96%	SaaS	
Monitoring containers and the services running within them to extract indicators like container setup times, resource usage, and service scalability.	7s	85%	IaaS	Container-Based Extraction Methods
	5 s	88%	PaaS	
	1s	95%	SaaS	

The extraction methods used in this study include the following:

- Agent-Based Methods: A common approach for monitoring cloud services which uses agent-based software installed directly on servers, virtual machines, or containers. These agents collect detailed information about system status, resource utilization, and application performance.
- API-Based Methods: These methods allow IT administrators to retrieve necessary data from cloud services without installing additional software. They leverage APIs provided by cloud service providers to gather information.
- Log-Based Methods: Logs contain detailed records of events and operations across cloud systems and services. Collecting and analyzing logs provides valuable insights into system issues, security concerns, and performance metrics.
- Network Traffic-Based Methods: Another effective way to monitor cloud services is through analyzing network traffic. This method involves examining data flows within the network, offering insights such as bandwidth usage, network latency, and packet loss rates.
- Telemetry-Based Methods: Telemetry-based methods involve collecting and transmitting performance data from distributed cloud services and applications. This approach is particularly useful in microservices environments, providing an overarching view of system status.
- Event-Based Methods: Event-based methods focus on analyzing and monitoring specific events in the cloud

infrastructure. These events may include service outages, sudden traffic spikes, or configuration changes.

- Container-Based Methods: With the increasing adoption of containers and orchestration platforms like Kubernetes, specialized methods for monitoring and managing these environments have become essential. These methods help identify container-related issues and manage their performance effectively.
- AI-Based Methods: Artificial intelligence and machine learning-based methods are increasingly used for monitoring cloud services. These approaches utilize machine learning algorithms to analyze existing data, identify complex patterns, predict failures, and optimize service performance [9].

Each of these methods are mapped to specific layers of cloud services (IaaS, PaaS, and SaaS) based on their suitability and functionality within the proposed architecture.

IV. Proposed Architecture

The design of the proposed cloud service monitoring architecture is illustrated in Figure 1. This architecture is designed to collect, process, analyze, and store data from various sources effectively and in a scalable manner. It is structured into the following layers:

A. Data Sources Layer

This layer collects data from multiple sources, including the user panels of cloud service providers and their financial reports. Based on the monitoring indicators defined for each cloud service layer (IaaS, PaaS, and SaaS) and the extraction methods mapped to these indicators (as shown in Table 1), the data is categorized into three types:

- Time-series data
- Log data (primarily textual)
- Analytical data

B. Processing and Preprocessing Layer (ETL Layer)

After extraction, the data undergoes transformation, cleaning, and formatting in this layer to prepare it for storage in the databases within the data warehouse.

- ETL tools process the data and convert it into formats suitable for storage.

C. Data Storage Layer

Different types of data of time-series data, log data, and analytical data, are stored in this layer. These are stored in specialized databases based on their type:

- Time-series data: Stored in databases like InfluxDB.
- Log data: Stored in databases like Elasticsearch.
- Analytical data: Stored in SQL or NoSQL databases, depending on requirements.

D. Data Warehouse Layer

This layer is designed for storing and managing analytical and historical data. It aggregates data from various storage sources after preprocessing. Data from different storage systems is consolidated here for further analysis and historical tracking.

E. Data Processing and Analysis Layer

In this layer, the stored data is analyzed using advanced analytical models and algorithms. Artificial intelligence and machine learning techniques are employed to extract valuable insights, identify patterns, and conduct predictive analysis. Specifically, LSTM (Long Short-Term Memory) algorithms are used to identify anomalies in time-series data, and K-means clustering algorithms are used for clustering and identifying similar patterns in the data. Additionally, Regression and Random Forest models are employed for analyzing complex trends and predicting future behaviors. The choice of these algorithms is due to their ability to process large and complex data and identify hidden patterns. For example, LSTM is particularly useful for analyzing time-series data and predicting potential anomalies. The K-means clustering algorithm allows us to group the data into similar clusters, enabling the identification of underlying trends within each group.

This layered architecture ensures efficient monitoring and analysis of cloud services across various layers, facilitating proactive decision-making and optimization. By utilizing these algorithms, we can detect potential issues in a timely manner and take appropriate actions to prevent them. Selecting the right algorithms and evaluating their accuracy through simulations and real-world environments will help improve the efficiency and accuracy of predictions.

F. Alerting Layer

This layer is responsible for detecting potential issues or anomalies following the data analysis process and sending appropriate alerts. These alerts play a critical role in the early identification of problems and enabling timely responses. To enhance operational reliability—especially in AI-based monitoring environments—this layer incorporates mechanisms for managing false alarms. These include machine learning models that differentiate between true anomalies and noise, prioritization of alerts based on severity and likelihood, and continuous model updates through human feedback. Such approaches reduce unnecessary alerts and improve the accuracy and effectiveness of the alerting system.

G. Dashboard and Reporting Layer

In this layer, the results of data analysis are visualized through reports and graphical dashboards. These tools allow teams and managers to make quick, informed decisions.

- Reporting Tools: Tableau, Power BI, Grafana

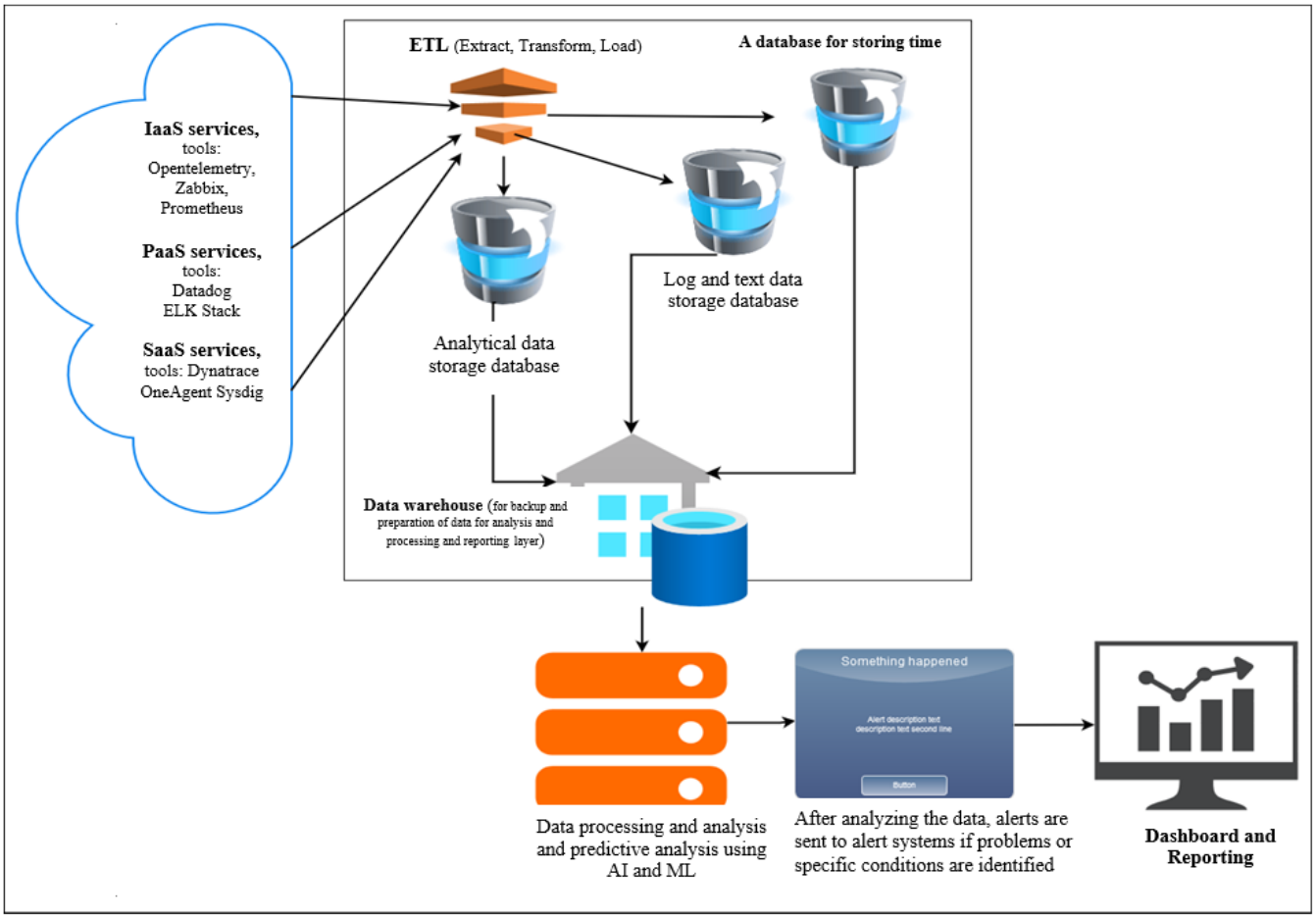

Figure 1. Proposed Architecture for Monitoring Center in Different Cloud Service Layers: IaaS, PaaS and SaaS

In the proposed architecture, the data warehouse acts as the central component for storing and managing analytical and historical data. Data is extracted and processed from various sources and stored in the data warehouse for advanced analysis, reporting, and alert generation.
The interaction between the components is efficiently designed to ensure a smooth flow of data from sources to the processing and storage layers, ultimately leading to actionable insights and reports. This integrated approach facilitates effective monitoring and decision-making.

V. EXAMINING THE PROPOSED ARCHITECTURAL SOLUTIONS FOR THE EXISTING CHALLENGES OF A CLOUD SERVICE MONITORING CENTER ARCHITECTUR

This section, examines potential challenges and obstacles in implementing monitoring architectures on a global scale and provides practical and applicable solutions to address these issues in the proposed architecture which is shown in Table 3.

Table 3 . Advantages of the Proposed Architecture Compared to a Standard Architecture

Advantages	Solutions in Proposed Architecture
1- Scalability	Utilizing distributed and scalable systems like Apache Kafka, Hadoop, and Spark
2- Management of Diverse Data	Advanced ETL tools such as Apache NiFi or Talend
3- Data Security and Privacy	The use of encryption during data transfer and storage, along with strict access control and advanced security tools
4- Data Integrity	Protocols like Eventual Consistency or Strong Consistency in distributed databases
5- Performance and Latency	Real-time processing tools like Apache Kafka or Apache Flink
6- Complexity in Data Management	Tools such as Apache Atlas for metadata management
7- Cost Optimization	Utilizing pay-as-you-go models and adopting cold storage f
8- Testing and Quality Assurance	Automated testing processes and data quality tools

A. Scalability

- Challenge: In every cloud service monitoring center architecture, the increasing volume of data on a global scale can create scalability issues. Specifically, in the ETL and data storage layers, processing and storing large amounts of data may lead to delays and performance bottlenecks.
- Solution in Proposed Architecture: Utilizing distributed and scalable systems like Apache Kafka, Hadoop, and Spark enables parallel data processing and improves scalability.

B. Management of Diverse Data

- Challenge: In every cloud service monitoring center architecture, different data sources have various formats and structures (such as time-series data, logs, textual data, and images). This diversity can complicate data processing and analysis.
- Solution in Proposed Architecture: Advanced ETL tools such as Apache NiFi or Talend ensure seamless processing and harmonization of diverse data types.

C. Data Security and Privacy

- Challenge: In every cloud service monitoring center architecture at a global scale, ensuring data security and compliance with privacy regulations (such as GDPR or CCPA) for both stored and processed data is a serious challenge.
- Solution in Proposed Architecture: The use of encryption during data transfer and storage, along with strict access control and advanced security tools, ensures robust data protection.

D. Data Integrity

- Challenge: In every cloud service monitoring center architecture within distributed systems, synchronizing data across different sources and ensuring data integrity in the event of failures or network issues is difficult.
- Solution in Proposed Architecture: Protocols like Eventual Consistency or Strong Consistency in distributed databases help preserve data integrity.

E. Performance and Latency

- Challenge: In every cloud service monitoring center architecture, processing and analyzing data, especially in real-time scenarios, can experience delays—particularly when data is collected from multiple geographic regifons.
- Solution in Proposed Architecture: The use of real-time processing tools like Apache Kafka and Apache Flink can significantly reduce latency and improve system performance across various layers of cloud services. These tools are particularly effective in the architecture of a cloud services monitoring center, where real-time data processing and analysis are crucial. For instance, in a cloud services monitoring center, data from multiple sources such as servers, devices, and monitoring systems are continuously transmitted. These data could include information on network traffic, service status, security issues, and other related variables. In such scenarios, Apache Kafka can be used to collect and transfer this real-time data across different layers of the system. For example, data regarding the status of services and servers is collected from various points and sent via Apache Kafka to other components for analysis. Moreover, Apache Flink can be employed in the data analysis layer to process the received data in real-time and identify any anomalies or issues. For example, if an overload is detected on one of the servers, Apache Flink can immediately analyze this data and send alerts to the system for prompt action.

F. Complexity in Data Management

- Challenge: In every cloud service monitoring center architecture, managing data at a global scale requires precise strategies for data transparency, data quality, and resource management.
- Solution in Proposed Architecture: Tools such as Apache Atlas for metadata management and data lineage efficiently track and manage data from source to consumption.

G. Cost Optimization

- Challenge: In every cloud service monitoring center architecture, implementing and maintaining such a complex architecture on a global scale may involve significant costs, particularly in infrastructure, data processing, and storage.
- Solution in Proposed Architecture: Utilizing pay-as-you-go models and adopting cold storage for historical data significantly reduces costs. Total Cost of Ownership (TCO) includes hardware costs and software, operational costs, long-term costs including scalability and service downtime. Pay-per-use and cold storage reduce TCO costs. For Example, in a monitoring center that generates around 5 terabytes of data per day, storing all data in hot storage can cost more than \$2,000 per month. However, by transferring 80% of historical data to cold storage solutions such as Amazon Glacier, storage costs can be reduced by approximately 70%. Also, by combining pay-as-you-go models for compute resources with cold storage for infrequently accessed data, the Total Cost of Ownership (TCO) can be reduced by 30% to 50% compared to traditional models. This approach not only lowers costs but also improves scalability by ensuring that resources are consumed only when needed.

H. Testing and Quality Assurance

- Challenge: In every cloud service monitoring center architecture at a global scale, ensuring the accuracy of operations and data quality at every architectural layer, especially in processing and analysis layers, is challenging.
- Solution in Proposed Architecture: Automated testing processes and data quality tools like Great Expectations assess and monitor data quality effectively.

I. Integration with Existing Systems

- Challenge: In a cloud service monitoring center architecture, integrating a new architecture with existing systems in organizations and various cloud services can introduce complexities.
- Solution in Proposed Architecture: Leveraging standard APIs facilitates the integration process, ensuring smooth compatibility with existing systems.

J. Change and Version Management

- Challenge: In a cloud service monitoring center architecture, within large-scale and globally distributed systems, managing changes in data and processing methods—especially when different versions of software or databases are used—can be difficult.
- Solution in Proposed Architecture: Utilizing version control systems like Git for code and tools such as Apache Hive or Data Lake so that data ensures effective management of changes and versioning.

VI. Comparison Criteria of Previous Traditional Architectures with the Proposed Architecture

If we consider the lower score as 1, the medium score as 3, and the high score as 5 on a scale from 1 to 5, the results of comparison criteria of previous traditional architectures with the proposed architecture are shown in Table 4. The scoring in Table 4 is based on expert opinion and professional judgment.

Table 4. Scores of the Proposed the Architecture with Traditional Architectures

Index \ Architectures	Traditional	Proposed
Scalability	Medium (3)	High (5)
Security and Data Integrity	Low (1)	High (5)
Accuracy in Predicting Potential Events	Medium (3)	Medium (3)
Impact of Decentralized Technologies	Low (1)	Medium (3)
Multi-layer Monitoring Support	Medium (3)	High (5)
Integration with Other Systems	Low (1)	High (5)
Adaptability to Changing Requirements	Low (1)	Medium (3)
Average Scores	**1.86**	**4.43**

1. Scalability:
Proposed Architecture: Due to the use of distributed and scalable systems such as Apache Kafka, Hadoop, and Spark, the proposed architecture offers high scalability.
Traditional Architecture: It is a framework for resource management in dynamic cloud environments that can adapt to changing user demands. While it addresses scalability needs, it has not paid sufficient attention to monitoring different service layers [5]. Therefore, it has moderate scalability.

2. Security and Data Integrity:
Proposed Architecture: The architecture employs data encryption during transmission and storage, enforces strict access control policies, and utilizes advanced security tools. Therefore, it ensures high security and data integrity.
Traditional Architectures: One of the main challenges in traditional architectures is security. As a result, they provide low security and data integrity.

3. Accuracy in Predicting Potential Events:
Proposed Architecture: By utilizing artificial intelligence and machine learning algorithms to extract useful information and patterns, as well as predictive analytics in the data processing and analysis layer, the proposed architecture achieves moderate accuracy in predicting potential events.
Traditional Architectures: This feature is available to a limited extent in the SaaS layer. Thus, the accuracy in predicting potential events is also moderate.

4. Impact of Decentralized Technologies:
Proposed Architecture: The role of Blockchain in enhancing security and scalability has been explored. Therefore, the architecture moderately addresses the impact of decentralized technologies.
Traditional Architectures: They lack this feature.

5. Multi-layer Monitoring Support:
Proposed Architecture: It supports multi-layer monitoring, making its qualitative rating for this criterion high.
Traditional Architectures: In cloud monitoring architectures, a multi-layer monitoring system has been proposed, utilizing machine learning techniques to analyze collected data [7]. This capability is moderately fulfilled in traditional architectures.

6. Integration with Other Systems:
Proposed Architecture: The use of standard APIs facilitates integration, while version control systems like Git for code and Apache Hive or Delta Lake for data help manage changes. Thus, high integration capability with other systems is achieved.
Traditional Architectures: They lack this feature, resulting in low integration capability with other systems.

7. Adaptability to Changing Requirements:

Proposed Architecture: Standard APIs facilitate integration, and version control systems like Git for code and Apache Hive or Delta Lake for data assist in managing changes. Therefore, the architecture achieves high adaptability to changing requirements.
Traditional Architectures: They lack this feature, leading to low adaptability to changing requirements.

As shown in Figure 2. The proposed architecture is significantly superior to traditional architectures, especially in terms of security, scalability, integrity, and adaptability.

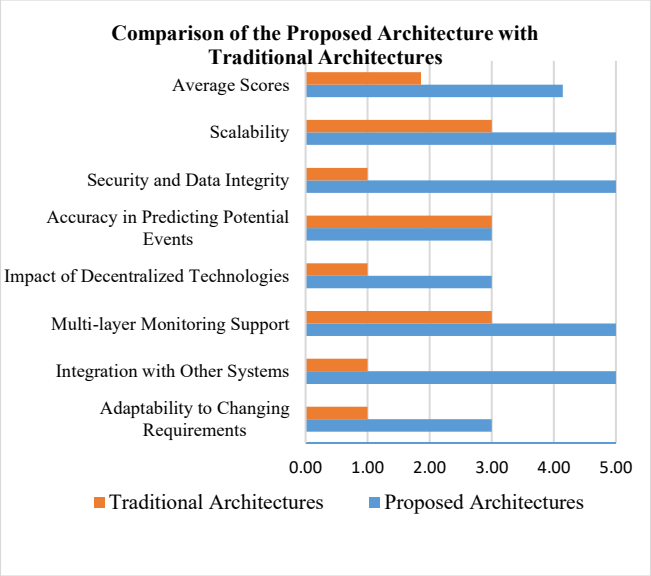

Figure 2. Comparison of Proposed Architecture with Traditional Architectures

VII. CONCLUSION

A review of existing literature reveals that technical, security, and business indicators across all layers of cloud services have not been comprehensively and interactively analyzed. The architecture proposed in this paper leverages artificial intelligence and machine learning algorithms in the data analysis layer, not only addressing these challenges but also enabling the prediction of potential events. This architecture incorporates technologies such as blockchain, multi-layer monitoring, and high integration capabilities. Evaluation results indicate that it significantly outperforms traditional architectures, particularly in scalability, security, data integrity, and adaptability.
In this context, key challenges such as limited scalability, data security, data integrity, high processing and storage costs, dependence on central entities, and the need for increased transparency and trust in cloud systems have been identified. To address some of these challenges, blockchain and decentralized systems have been explored as potential solutions. These technologies can:

- Enhance data security and integrity: By using encryption and consensus mechanisms, they prevent data alteration and manipulation, improving security.
- Eliminate dependence on central entities: By distributing data across multiple nodes, reliance on centralized servers is reduced, making the system more resilient.
- Improve transparency and trust: By recording changes in an immutable ledger, direct monitoring and validation of data become possible.
- Manage access and digital identity: Using decentralized technologies, user authentication and data access control can be done without the need for intermediary entities.

However, a key gap observed in prior studies—as well as in the current proposed architecture—is the lack of adequate attention to the potential of edge computing and fog computing. Given the rapid growth of the Internet of Things (IoT) and edge cloud ecosystems, the combined use of these technologies alongside blockchain can play a crucial role in achieving decentralized monitoring, real-time processing, and enhanced infrastructure resilience.
In future research, exploring Blockchain hybrid models, optimizing consensus algorithms to enhance efficiency, developing smart contracts for automatic data management, and decentralized storage solutions could help solve more challenges in cloud environments.
Therefore, it is recommended that future research focus on designing hybrid architectures that integrate blockchain, edge, and fog computing. This should be accompanied by the development of smart contracts, optimization of consensus algorithms, and adoption of decentralized storage solutions. Such a strategic approach can enhance the security, efficiency, and transparency of large-scale cloud services, and provide a flexible, intelligent, and robust infrastructure for emerging ecosystems such as IoT, smart cities, and critical cloud-based applications.

REFERENCES:

[1] *Mell, P., The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology, 2011.*

[2] *Armbrust, M., et al., A view of cloud computing. Communications of the ACM, 2010. 53(4): p. 50-58.*

[3] *Zissis, D. and D. Lekkas, Addressing cloud computing security issues. Future Generation computer systems, 2012. 28(3): p. 583-592*

[4] *Zhou, W., Leung, V. C., & Lu, H. (2013). Cloud computing: A perspective study of research trends Proceedings of the International Conference on Cloud Computing.,*

[5] *Buyya, R., C.S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. in 2008 10th IEEE international conference on high performance computing and communications. 2008. Ieee.*

[6] *Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2011). A view of cloud computing. Communications of the ACM, 53(4), 50-58.*

[7] *Chen, H., Wang, F., Helian, N., & Akanmu, G. (2014). User-prioritized scheduling policy for hybrid IaaS cloud architecture. Future Generation Computer Systems, 37, 93-105.*

[8] *Zhang, Q., L. Cheng, and R. Boutaba, Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 2010. 1: p. 7-18.*

[9] *D. maleki, N.Ghorbani, E. Arianyan, A. Mansouri, Proposing a Comprehensive Method for Extracting Monitoring Indicators for Cloud Service Layers, Tehran, IST 2024.*

[10] *Sinha, N. and L. Khreisat. Cloud computing security, data, and performance issues. in 2014 23rd Wireless and Optical Communication Conference (WOCC). 2014. IEEE.*

[11] *Sehgal, N.K., P.C.P. Bhatt, and J.M. Acken, Cloud computing with security and scalability. 2020: Springer.*

[12] *Khare, S., U. Chourasia, and A.J. Deen. Load balancing in cloud computing. in Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 1. 2022. Springer.*

[13] *Kuyoro, S.O., F. Ibikunle, and O. Awodele, Cloud computing security issues and challenges. International Journal of Computer Networks (IJCN), 2011. 3(5): p. 247-255.*

[14] *Yang, L., et al. Defense of DDoS attack for cloud computing. in 2012 IEEE international conference on computer science and automation engineering (CSAE). 2012. IEEE.*

[15] *Alhenaki, L., et al., Security in cloud computing: a survey. International Journal of Computer Science and Information Security (IJCSIS), 2019. 17(4): p. 67-90.*

[16] *Mishra, D., et al., Intelligent and cloud computing. Proceedings of ICICC, 2019. 1*

[17] *Li, Y., et al., Big data and cloud computing. Manual of digital earth, 2020: p. 325-355.*

[18] *Masiyev, K.H., et al. Cloud computing for business. in 2012 6th International Conference on Application of Information and Communication Technologies (AICT). 2012. IEEE.*

[19] *Stantchev, V. Performance evaluation of cloud computing offerings. in 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences. 2009. IEEE*

[20] *Barcelo, M., et al., IoT-cloud service optimization in next generation smart environments. IEEE Journal on Selected Areas in Communications, 2016. 34(12): p. 4077-4090.*

[21] *bakh, M., et al., Cloud Computing and Big Data: Technologies, Applications and Security. 2019: Springer.*

[22] *Siriwardena, P., Advanced API Security. Apress: New York, NY, USA, 2014.*

[23] *Krutz, R.L. and R.D. Vines, Cloud security: A comprehensive guide to secure cloud computing. 2010: Wiley Publishing.*

[24] *Stamp, M., et al., Artificial Intelligence for Cybersecurity. 2022: Springer*

[25] *Kratzke, N. Cloud Computing Costs and Benefits: An IT Management Point of View. in Cloud Computing and Services Science. 2012. Springer.*

[26] *Weinhardt, C., et al., Cloud computing–a classification, business models, and research directions. Business & Information Systems Engineering, 2009. 1: p. 391-399*

[27] *Asadi, S., et al., Customers perspectives on adoption of cloud computing in banking sector. Information Technology and Management, 2017. 18: p. 305-330*

[28] *Boniface, M., et al. Platform-as-a-service architecture for real-time quality of service management in clouds. in 2010 fifth international conference on internet and web applications and services. 2010. IEEE.*

[29] *Liu, F ,.L. Li, and W. Chou. Communications enablement of software-as-a-service (SaaS) applications. in GLOBECOM 2009-2009 IEEE Global Telecommunications Conference. 2009. IEEE.*

[30] *Fatima, E., I.A. Sumra, and R. Naveed, A comprehensive survey on security threats and challenges in cloud computing models (SaaS, PaaS and IaaS). Journal of Computing & Biomedical Informatics, 2024. 7(01): p. 537-544*

[31] *Yimam, D. and E.B. Fernandez, A survey of compliance issues in cloud computing. Journal of Internet Services and Applications, 2016. 7: p. 1-12.*

[32] *Chenthara, S., et al., Security and privacy-preserving challenges of e-health solutions in cloud computing. IEEE access, 2019. 7: p. 74361-74382.*

[33] *Kumar, V. and W. Reinartz, Customer relationship management. 2018: Springer.*

[34] *Pipino, L.L., Y.W. Lee, and R.Y. Wang, Data quality assessment. Communications of the ACM, 2002. 45(4): p. 211-218.*

Silberschatz, A., H.F. Korth, and S. Sudarshan, Database system concepts. 2011.

How to cite: D. Maleki, N. Ghorbani, E. Arianyan, M. Beiklaryan, **A Novel Architecture for Monitoring and Evaluating Cloud Services Based on Indicator Extraction ,** Journal of Distributed Computing and Systems (JDCS), Vol 7, Issue 2, Pages 86-95, 2025.

Davood Maleki is a faculty member at the Research Institute for ICT (Information and Communication Technology) of Iran. He holds a Master's degree in Computer Engineering from Ferdowsi University of Mashhad and completed his Bachelor's degree in Computer Engineering as well. His areas of interest include virtualization, cloud computing, big data, and data centers.

Email: dmaleki@itrc.ac.ir

Neda Ghorbani is a collaborator at the Research Institute for ICT (Information and Communication Technology) of Iran. She received her Master's degree in Software Engineering from Al-Taha University. She has experience working with the NeuroGame research team and the Baqiyatallah Medical Sciences Research Center.

Email: n.ghorbani@itrc.ac.ir

Ehsan Aryanian holds a Ph.D. in Electrical and Electronics Engineering from Amirkabir University of Technology. He is the Head of the Information Technology Research Institute and an Assistant Professor at the Research Institute for ICT (Information and Communication Technology) of Iran. His areas of interest include cloud computing, big data, parallel processing, and data centers.

Email: ehsan_arianyan@itrc.ac.ir

Masoud Beik Larian received his Bachelor's degree in Electronic Engineering from Noshirvani University of Technology in Babol and his Master's degree in Power Electrical Engineering from Islamic Azad University, South Tehran Branch. He is the Director General of Cloud Management and Development at the E-Government Department of the Information Technology Organization of Iran. His interests lie in the fields of cloud computing, artificial intelligence, and electronics.

Email: biklaryan@ito.gov.ir