International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Automating cross-account access in hybrid Lakehouse data governance

Manish Ravindra Sharath *

University of Texas at Dallas, Richardson Texas.

International Journal of Science and Research Archive, 2025, 18(01), 416-423

Publication history: Received on 08 October 2025; revised on 18 November 2025; accepted on 20 November 2025

Article DOI: https://doi.org/10.30574/ijsra.2026.18.1.3084

Abstract

A growing degree of complexity in enterprise data infrastructures has also already observed the proliferation of hybrid lakehouse designs that mix the scaling of data lakes with the structural management of traditional data warehouses. The cross-account access, which is secure and scalable, is a management issue in this regard. In the paper, the author will discuss automation structures as the interventions to improve data governance in the hybrid lakehouse environment, i.e., to ensure that access controls, audit functions, compliance checks, and analytics processes across the accounts in the clouds are not in conflict with each other. The enterprises would be insured of the regulatory compliance, data integrity, and work efficiency of the distributed systems through the adoption of cloud-native solutions, the event structure, and AI-based analytics. The paper gives the general perspective of the architectural, operational, and security environment of the automation of the cross-account governance and, consequently, its part in the modernization of the data ecosystems of business based on recent research and implementation plans.

Keywords: Hybrid Lakehouse; Cross-Account Automation; Data Governance; Cloud Compliance

1. Introduction

The emergence of cloud computing systems and the subsequent explosion of data in diverse business areas has created the perception of urgency for the need to have the best systems of governance. The hybrid lake house architectures are one of them and have taken a prominent centre stage since they have been powered by scalable, flexible, and high-performance analytics. But notwithstanding these advantages, there is a big issue with automated cross-account access control within this hybrid system, which ought to be safe. The simplicity and control of information in various accounts about the cloud-computing systems bring the issue a step further, regarding the standards, real-time auditing, and automating the process. These sectors require automation of a good governance infrastructure so as to provide integrity of information, security, and adherence to the regulations. In this paper, the technologies, architectures, and implementations that will allow automating the cross-account access governance in hybrid lakehouses will be discussed.

2. Background: Cross-Account Access in Hybrid Lake houses

Scalability of data lakes and their flexibility, and the traditional data warehouses and their ability to organize and manipulate data, have led to the integration of the hybrid lakehouses. These hybrid lakehouses are also used by enterprise ecosystems in which, in the majority of cases reported, departments or even individual organizations that need access to cross accounts must be safe and painless. The fact that this access is handled by mechanical mechanisms, as is the custom nowadays, is not only liable to error but, besides, is incapable of meeting the demands of modern data.

The characteristic that determines access to cloud accounts is access by services or identities of one cloud account to the resources of another cloud account. This access has to be extremely limited, and the scenario monitored in a hybrid

* Corresponding author: Manish Ravindra Sharath

lakehouse environment where the information can be stored in a broad variety of clouds or accounts. These systems are automated to offer standard security positions and audit trails and minimize the overheads in operations [1].

One of them is AWS, providing the following services: AWS Organizations, AWS Identity and Access Management (IAM), and AWS Resource Access Manager (RAM), which can be used in cross-account governance. It is also stipulated that in order to achieve the phase of implementing these services to become a component of the entire automation pipeline, a more strategic and scalable process is necessary [1].

3. Architectural Automation and Governance Models

The emergence of the multi-accounting functionality in the clouded setting justifies the requirement to have centralized automation plans. In the most recent issue, Challa describes a model in which the centralized service control policies (SCP), organizational units (OU), and AWS Control Tower balance the management of the accounts, as well as between them, and each has a comparable responsibility model [1]. The hierarchical system will give an account of the security policy of the same and account of the flexibility of the accounts.

One of such innovations is the deployment of governance settings in the form of automation templates and infrastructure-as-code (IaC) scripts, e.g., AWS CloudFormation or Terraform. In order to make the IAM roles' instantiations automated, organisations can make the configuration drift more reproducible and reduced [1].

The second is operational observability. Continuous compliance can be made by using such tools as Amazon CloudWatch and AWS Config, and the remediation workflow can be automatically turned on, thus making the posture of governance even more human-free. This service is an automated one, which is required in hybrid architectures, since the latency, compliance requirements, and modalities of the storage in dissimilar platforms vary.

4. Event-Driven Access Monitoring and Auditing

The data access and activity of an automated cross-account environment must have a compliance that is at a forensic level in order to decrease violations. One of the layers of the model that will be applied in the context of AWS EventBridge and AWS CloudTrail is one of the layered models offered by Datla and Malay, where chains of evidence are automatically constructed and most controlled regions, including healthcare, are taken into account [2]. Their solution implies an event-based architecture that in real time tracks the access record and occurrence of accounts.

Assigning roles, blocking access, and access in this model are communicated to AWS CloudTrail API calls, which in turn are relayed via EventBridge, which in turn automates workflows. These categories of workflow are allowed to access an unwritable storage or warn against abnormal patterns. In this regard, they provide acceptable evidence chains that should be required by forensic auditing and law [2].

Based on the example above, the hard work of an unauthorized access of Account A to the patient information in Account B that has been unsuccessful can be reported and intensified in a few seconds. In practice, one can take such services as AWS CloudTrail with Amazon EventBridge and AWS Lambda so that, in the case of the need to respond within 2–5 seconds, this consideration is one of the factors of the complexity of event rules and service-to-service lag. Thereupon, the automation of the Lambda functions that will invalidate suspicious credentials or quarantine datasets will result in responsiveness in governance. After a breach has been detected, the system will commonly either set in motion a train of automated measures: (1) temporary access tokens/IAM identities are de-authenticated; (2) a security event notification by connecting to an Amazon SNS or SIEM, one with a tool such as Splunk; (3) an account of the incident to a permanent repository (e.g., using object lock to Amazon S3); and (4) optionally, an incident response runbook which may isolate the compromised account or dataset, and enable it to be interrogated by security specialists. Cloud auditing in this model is a step beyond the active gathering of the logs to active threats and forensic preparedness to actively counter the threat [2].

5. Regulatory Compliance Automation in Cross-Account Environments

The compliance of cross-account settings also extends the access control to the auditability, encryption, data residency, and industry concerns. According to Varadararaj, mobile payment systems also need to use compliance automation processes because the flow of information touches two or more accounts and geographical locations [3]. The automated systems can be used in ensuring that the regulatory conditions are not a post-hoc activity by tracing the regulatory

requirements to certain technical controls in order to give the regulatory requirements as a safety-valve of the data lifecycle.

An overlapping of a compliance automation engine includes the following: a rules engine that signifies conventions (including GDPR, PCI-DSS, or HIPAA), a scanning engine that scans information and arrangements at the account, and a remediation module that enforces compliance automatically [3]. The process can be optimized with the help of the integration with the cloud-native option, like AWS Config Rules and AWS Security Hub, which is why the process can be checked at any moment and implemented.

Second, the automation will provide a rapid policy change to satisfy the dynamic policies. With a sample of a new requirement used to make the logs be stored seven years instead of five years, the automated structure can make hundreds of accounts change within minutes [3]. This is usually done with infrastructure-as-code (IaC) tools (e.g., AWS CloudFormation, HashiCorp Terraform, or AWS CDK) wherein compliance policies can be maintained in a version-controlled format and delivered to disparate units in an organization in a programmable format. Auto-coordinated, such tools may be auto-managed via automated management of pipelines of services like AWS CodePipeline or third-party CI/CD services like Jenkins or GitHub Actions. The models of policies (e.g., parameters of retention periods in logging environments) are stored in a central repository in these regulations. In turn, workflow automation triggers deployments and transfers such changes to all accounts of interest within AWS with the help of secure and cross-account roles that are supposed to provide consistency and minimize the possibility of such changes. Further, with monitoring tools, service control policies (SCP), and AWS Config policy could be automatically updated, and new standards and skills could be implemented after implementation, and compliance could be monitored. This will not only ensure compliance but also agility in response to regulatory environments.

6. Intelligent Threat Mitigation and Lateral Movement Detection

Credential stuffing attacks or a type of lateral movement by malicious individuals can be performed using weak cross-account access control as a point of entry. Asante et al. argue that, in addition to that, there is an increase in the high level of sophistication of attackers in account switching, in the utilization of such misconfigurations to steal data, without a sense that they are being detected [4].

In this respect, machine learning and behaviour analytics fall under the category of smart automation. The intelligent automation in this instance is the architecture of the integration of Security Information and Event Management (SIEM) systems, cloud-native analytics engines, and Artificial Intelligence (AI)-based anomaly detection architecture. Such systems are supplied with telemetry data from various sources, including AWS CloudTrail, VPC Flow Logs, Amazon GuardDuty results, and IAM activity logs. They had developed machine learning algorithms to provide a baseline of normal cross-account activity, and these are unsupervised clustering, time series analysis, and anomaly scores. Immediately this is identified, peculiarities like large data access by the batches, off-hours access, and network traffic structure are automatically notified as a threat. The intelligence layer performs continuous retraining of the detection models on services such as Amazon SageMaker or Azure ML with new data and gets more accurate over time. Such systems can identify abnormalities that are not present in the normal trends of accessing an account when profiling the normal trends of accessing an account. It may be a sign of compromise in such a case where a position taken up by an accounting application in Account X, at a certain time, has set into motion a transfer of data in a massive size with one of the data lakes in Account Y [4].

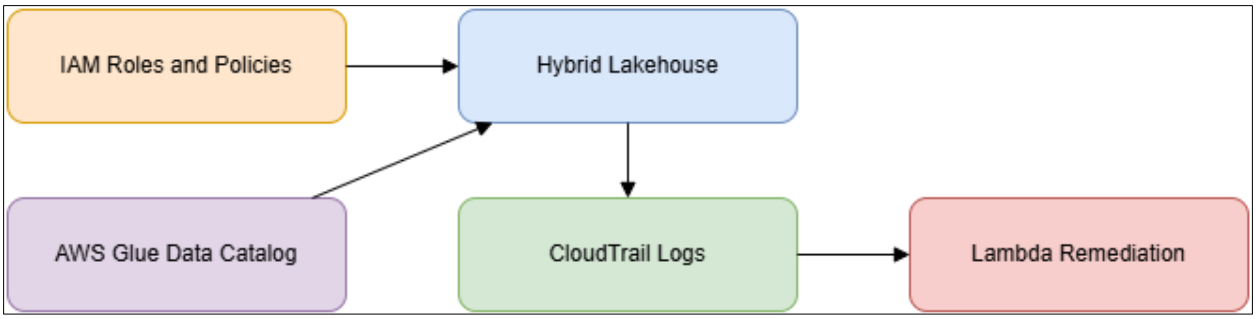
Moreover, once such detection mechanisms are involved and deployed on the automation platforms, the organizations can also perform corrective measures in real-time, including the temporary loss of credentials, the replacement of IAM keys, or isolating accounts. Based on this definition, an automation platform is an orchestrated stack that has: (1) engine coordinators (e.g., Amazon EventBridge and AWS Step Functions) which direct and execute multi-step procedures; (2) serverless layers of execution (AWS Lambda or Azure Functions) which run small remediation procedures (revoke tokens, update IAM policies, make inbound calls to KMS to re-encrypt data); (3) configuration management and runbook executors (e.g., AWS Systems Manager (SSM) Automation) which run tested remediation.

7. Data Governance Frameworks in Hybrid Lakehouses

To ensure that it has strong data governance on the hybrid lakehouses, it is necessary to match the data transparency of the lakes with the data manipulation control of the traditional data warehouses. The training elaborated by Kaur gives a guideline, which stipulates the security, access control, lineage monitoring, and logging as the platform of governance [5].

One of the main elements of this structure will be listing and sorting of information resources of all accounts and clouds. Another tool that is applied in pipelines and is used to identify sensitive data, enforce access controls, and maintain the data's visibility is AWS Glue Data Catalog and Apache Atlas. This is then succeeded by tag implementation with the help of Attribute-Based Access Controls (ABAC), and authorized identities can only access specific datasets even in the case of cross-accounts [5].

Figure 1 below illustrates a governance reference architecture in a hybrid lakehouse, integrating access controls, cataloging, and audit systems.



(Source: Reconstructed from [5])

Figure 1 Governance Architecture for Hybrid Lakehouse Data Environments

8. Automated Remediation and Enforcement

Besides monitoring and detection, the enforcement initiatives would also be automated in order to implement compliance and data integrity in an active cloud environment. It is developed by the Lambda remediation capabilities that are used to apply the NIST 800-53 security principles to the AWS systems automatically [6].

The violations detected by AWS Config or Security Hub will empower such remediation. Using an example, it would be due to automation that public access could be restricted, data encrypted, and notifications given to the security operations teams when any S3 bucket containing sensitive information comes online. This is self-prescription and ensures that a reduction in the exposure time interval is made and imposes security policy across the accounts [6].

Table 1 Common Security Misconfigurations and Automated Remediations

Misconfiguration	Detected By	Automated Remediation Action
Public S3 bucket	AWS Config	Revoke public access, enable encryption
Unused IAM access keys	IAM Access Analyzer	Deactivate and rotate keys
Open security groups	AWS Security Hub	Restrict IP ranges and close unused ports
Non-compliant encryption settings	AWS Config	Apply KMS encryption and update resource policies
Role with excessive permissions	CloudTrail + Config	Adjust IAM role policies

(Source: Adapted from [6])

9. Evolving Data Integration Pipelines in Multi-Cloud and Hybrid Environments

The fact that the concept of the lakehouse architecture is introduced to the context of the hybrid and multi-cloud systems must reconsider the process of data integration management and, more importantly, the process of account and service integration. The radical solution to ETL, as envisaged by Adekola and Edward, is to automate it and make it metadata-driven, and this will be in a position to dynamically react to the metadata of the data lineage as well as the governance policies of the latter [7]. It is a replacement of hard-coded and fixed pre-determined ETL pipelines with automated coordination systems and is at the core of mixed configurations, where datasets are imported into and run on multiple accounts with clouds. Such pipelines are typically pipelined using the assistance of contemporary orchestration engines, which are Apache Airflow, AWS Step Functions, and Dagster. Specifically, Apache Airflow may model ETL programs as

Directed Acyclic Graphs (DAGs), which may be triggered dynamically when data is ingested, or the schema is changed or when the governance rules are changed. It would work perfectly in a hybrid world as these tools can cross-authenticate with service principals or IAM roles, and be federated with other data services (e.g., Amazon S3, Redshift, Databricks, BigQuery).

Data lineage represents a list of the changes and access policy of data at each stage of the pipeline when using automated ETL pipelines. Such pipelines will be implemented in a way that will comply with the regulations of governance that are unique to the accounts that are not interoperable with the upstream and downstream systems. This automation will be present in case there is a combination of data from an on-premise data center (Account A) and cloud-based analytics (Account B) to a single data lakehouse platform [7].

The orchestration systems (e.g., Apache Airflow or AWS Step Functions) that have permission-sensitive connectors will be presented in the latter to implement the plan of the approach that the access policy should be observed during execution. Such orchestration systems have the capability to automatically scale to any schema, volume of data, or policy change to achieve more reliability and less human intervention.

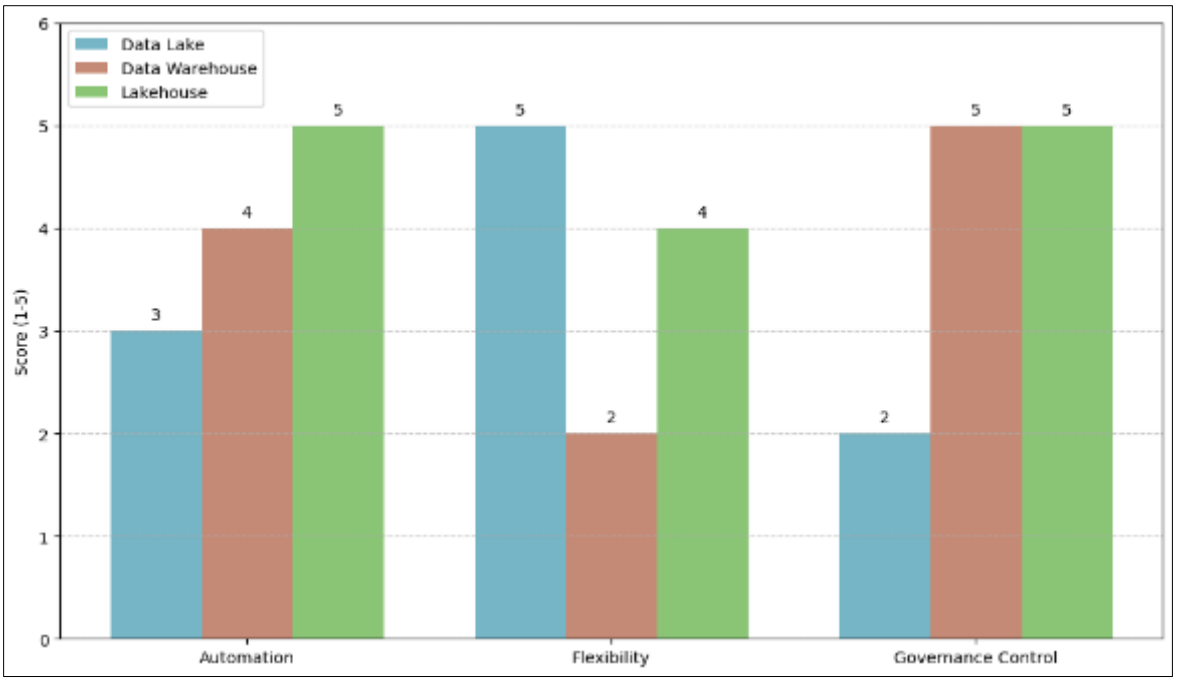
10. Comparative Analysis: Lakehouse versus Traditional Architectures

It may be perceived by considering the special value of the automation of the hybrid variant of the lakehouse with respect to it in comparison with more traditional architectures, including autonomous data lakes and warehouses. According to the model of comparative ranking of these models that was developed by Mary, the rankings of the following models were made where the flexibility, performance, security, and potential of automation were utilized in ranking the models [8].

Traditional data warehouses are not very dynamic, are strictly enclosed, and are fixed-schema and centrally managed, thus are easier to manage. Pure data lakes, on the contrary, are schema-flexible to be read but do not deliver an implicit security and governance service and are prone to cross-account setup misconfigurations.

The most appropriate one is the Lakehouse model, which falls between these extremes due to its design, which houses automation mechanisms. Lakehouses are easy to manage and offer more properties such as automatic schema inference, automatic cataloguing, and policy-based access permissions, and do not impact flexibility. One of them is schema inference on products such as Databricks or Delta Lake, where developers can respond to schema change within minutes and save days or weeks needed for schema changes (as was common with a standard ETL pipeline). The cost of developing a schema change to legacy systems was very prohibitive, requiring coordination between ETL programmers, data architects, and quality assurance teams, and would normally involve a full pipeline redeployment and retesting. Automated schema evolution allows the Lakehouse to find, state, and duplicate account and region schema differences. This reduces 60 percent of maintenance work, according to internal benchmarks published by the best cloud vendors, such as AWS and Databricks, and is over 70 times faster to onboard data than it once was in the past. What is even more is that the hybrid implementation model can also transfer such advantages to distributed systems that have automated policies that can be uniformly applied across accounts, regions, and clouds [8].

Figure 2 presents a performance and governance comparison among data lakes, warehouses, and lakehouses, focusing on cross-account automation capabilities.



(Source: Adapted from [8])

Figure 2 Comparative Evaluation of Data Architectures

11. Integration of Analytical Workloads in Cross-Account Lakehouses

Passing through the list of the best things about lakehouse models, one can mention the fact that the models can take batch, streaming, and interactive loads. Appalapuram suggests that elements of analytical needs of dissimilar departments and business units that are anticipated to be separate cloud accounts can be achieved through the modern lakehouse system [9]. This is possible because of such cross-account sharing functions as AWS Lake Formation resource connections or Delta Sharing, which provide the possibility to retrieve data in a safe manner without copying it.

On explaining this, Account X's business intelligence department has the capability of querying the curated information in Account Y without the transfer of information. It is accomplished through the resource links which are added to the AWS Glue Catalog or shared Delta table, which contains access controls and lineage metadata. It is the automation that ensures that such kinds of links are maintained and monitored in case there is a need to either change the data schema or add a set of data [9].

Besides this, the workloads of the analyst can be utilized in such a way that they would be able to access and audit them and apply throttling by invoking the fine-grained access controls to the role-based query engines such as Amazon Athena or Databricks SQL. They will ensure that they will not lose performance or agility while avoiding excessive use of resources or unauthorized access.

12. Migration and Transition Strategies in Legacy Systems

The massive architectural and administration modifications are linked with the migration of old warehouses or data lakes to the prototype of the hybrid lakehouse. The highest priorities of the applications that Hermanus used in developing a consistent migration strategy include automated schema mapping, policy translation, and access synchronization across various accounts in the clouds as top priorities [10]. The irreconcilability between identity systems, the lack of consistent, coherent metadata standards, and audit trail fragments are likely to act as hindrances to this transition.

The automation process involves three stages of the migration process: discovery, mapping, and deployment. Data asset and metadata cataloguing are a part of the discovery stage. The mapping stage is meant to convert the governance policies of IAM roles, data encryption, and classification standards into lakehouse-compatible forms. Lastly, by default, they are created with automated deployment scripts to create them in target accounts using the AWS Cloud

Development Kit (CDK), which enables programmers to code infrastructure with high-level programming languages such as Python, TypeScript, or Java. These definitions are coded on the CDK as AWS CloudFormation templates to be deployed repeatedly, with parameters, and with version control. Moreover, it can also support cross-account and multi-region deployments and bootstrapped environments with assumed secured roles; therefore, it can be scaled to governance configuration in large hybrid lakehouse environments [10].

They are very automated processes to restrict misconfiguration, as well as reduce time to value. It also gives the assurance that the current compliance structures are in the same or better condition in the new lakehouse environment. It is a form of automation that assists companies in modernizing the information infrastructure without disrupting compliance and security requirements from the beginning.

13. AI-Readiness and Predictive Governance Models

The need to apply AI-based systems of governance in the environment of the growing volume of information and the complexity of the compliance environment is urgent when lakehouse systems are used. According to Aggarwal, the application of AI to the creation of predictive formats of governance helps identify anomalies in the configuration, data access, and compliance prior to the occurrence of the anomaly [11].

In such models, operational information is used in log format, which is used to make prediction models, policies, and user actions. For example, given a set of data that indicates common dynamics of Access Control Lists (ACLs), the system can propose a more convenient access model, based on roles, or the subdivision of the dataset into access domains that logically partition. Such a system will also anticipate the enactment of the latter, which would probably pose a compliance risk, and automate the preventive actions by training on preceding instances of policy violations [11].

Practically, such systems are built upon built-in AI systems such as Amazon SageMaker, to which data pipelines and data protection processes are imposed. The AI agents keep learning, monitoring the behaviors of the organization, and adjusting the governance controls accordingly. An example would be a situation in which the requests for cross-account access steadily rise during non-business hours; a checkup can be added or a review of the administration can be triggered before granting such access.

Challenges and Future Directions

Although improvements have been made in the area of automation and cross-account governance, various problems remain. One major issue is that multi-cloud interoperability is still a barrier to the governance process, due to the production of identity models being fragmented, logging formats differing, and access control mechanisms varying among providers. Secondly, the lack of standardized metadata schemas means there is no easy way to have a standard policy definition, pattern of access, and tracing of lineage between accounts.

Another issue is that the current systems are not always scalable. Latency, throttling, and API rate limits are also challenging to manage — even with automated services — when one considers organizations that operate across hundreds or thousands of cloud accounts. The control layers must be lean and decentralized to ensure the live imposition of policy and incident response during large-scale events.

In the future, likely developments include the adoption of decentralized identity (DID) schemes, blockchain-based access control, and the integration of AI-governance as a core component. These could also reduce the burden on legal enforcement, help establish transparency, and create a self-healing governance space. Furthermore, cross-cloud standards like Open Policy Agent (OPA) and service mesh connectors are some of the few standards that can be applied to enforce policies with universal applicability across platforms in a multi-cloud environment.

14. Conclusion

This paper reviewed the current state of the practice of automating cross-account access in hybrid Lakehouse data governance, including architectural models, automation tools, compliance mechanisms, threat mitigation systems, and integration strategies. Organizations will be in a position to modernize governance and achieve greater levels of scalability, transparency, and security through cloud-native services, infrastructure-as-code, AI-driven analytics, and decentralized identity structures. Although this is not the time to delve deeply into interoperability and standardization, the challenges are resolvable with current advancements in cross-cloud policy enforcement and intelligent remediation. This work will be valuable to society because it will enable secure, compliant, and effective utilization of data within

complex cloud environments and open the pathway to more robust, automated, and intelligent governance infrastructures.

References

[1] Challa, S. R. (2025). Automating Multi-Account Governance in AWS: A Scalable Approach to Enterprise Cloud Management. Journal of Computer Science and Technology Studies, 7(7), 304-316.

[2] Datla, L. S., and Malay, S. S. (2025). Forensic Cloud Auditing in Healthcare: Using EventBridge and CloudTrail to Automate Evidence Chains and Access Logs. Artificial Intelligence, Machine Learning, and Autonomous Systems, 9, 13-35.

[3] Varadaraj, P. G. (2025, September). Compliance Automation for Mobile Payment Systems: Ensuring Adherence to Regulatory. In ICT for Global Innovations and Solutions: International Conference, ICGIS 2025, Virtual Event, April 26-27, 2025, Proceedings (p. 59). Springer Nature.

[4] Asante, G., Isaiah, P., and Hoover, R. (2025). Intelligent Credential Stuffing and Lateral Movement Across Cloud Platforms.

[5] Kaur, J. Securing Your Data Lake: A Comprehensive Guide to SecOps and Data Governance on AWS.

[6] Fairbanks, J. (2025). Automating AWS Compliance Enforcement (ACE): Development of Lambda Remediation Functions for NIST 800-53 Security Standards (No. LLNL-TH-2001355). Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States).

[7] Adekola, P., and Edward, E. (2025). Data Lakes, Lakehouses, and Beyond: Redefining ETL in Hybrid Cloud Environments.

[8] Mary, B. J. (2025). Unified Data Architecture for Machine Learning: A Comparative Review of Data Lakehouse, Data Lakes, and Data Warehouses.

[9] Appalapuram, V. S. R. (2025). The Lakehouse Paradigm: Converging Data Lakes and Warehouses for Integrated Enterprise Analytics. Journal of Computer Science and Technology Studies, 7(4), 641-648.

[10] Hermanus, D. (2025). Strategies for Migrating Data Warehouses to Data Lakehouses Using Public Cloud Computing (Doctoral dissertation, Walden University). [11] Aggarwal, J. (2025). Building an AI-Ready Data Strategy Using Lakehouse Technology. Journal of Computer Science and Technology Studies, 7(3), 663-676.