User Accountability in Cloud-Based Recommender Systems: A Secure and Transparent Framework Using Google Cloud Infrastructure

K.SUDHEER KUMAR, Department of Computer Science and Engineering, SR University, India

Recommender systems increasingly rely on cloud platforms for scalable data storage, computational power, and real-time delivery. However, this reliance also introduces critical concerns around user accountability, data transparency, and privacy. As these systems often process sensitive user behavior and preferences, ensuring traceability and auditability becomes essential. This paper proposes a cloud-native accountability framework built on Google Cloud that integrates role-based access control, immutable audit logs, transparent access reporting, and data protection services. We demonstrate the implementation of this framework on a movie recommendation prototype using Google Cloud services, including Cloud IAM, Cloud Audit Logs, Access Transparency, Cloud DLP[7], and BigQuery for monitoring. The proposed system ensures comprehensive user accountability, supports regulatory compliance, and promotes trust through secure and explainable recommendations. Extensive evaluation shows that our framework achieves high traceability coverage with minimal system overhead.

Additional Key Words and Phrases: Cloud Computing, Recommender Systems, User Accountability, Google Cloud, Data Privacy, Audit Logs, IAM, Transparency, Trust, Security

ACM Reference Format:
K.Sudheer Kumar. 2025. User Accountability in Cloud-Based Recommender Systems: A Secure and Transparent Framework Using Google Cloud Infrastructure. 1, 1 (March 2025), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems have become integral to numerous online platforms, shaping user experiences on services such as Netflix, YouTube, Amazon, and Spotify. These systems use large-scale data—often sensitive—to personalize content and optimize engagement. In modern deployments, recommender engines are frequently hosted on cloud platforms due to their scalable compute, storage capabilities, and operational flexibility.

While cloud computing enhances scalability and reliability, it also introduces significant challenges in ensuring *user accountability*. Cloud-based recommender systems interact with sensitive personal data such as user behavior, preferences, location, and transaction history. Misuse, unauthorized access, or untraceable operations in such systems can lead to privacy breaches, legal non-compliance (e.g., GDPR [3], CCPA [16]), and erosion of user trust.

Figure 1 illustrates the key accountability risks in such environments. These include lack of traceability of user data access, limited visibility into system administrators' actions, opaque algorithmic decisions, and insufficient control over third-party service integrations.

Author's address: K.Sudheer Kumar, Department of Computer Science and Engineering, SR University, Warangal, Telangana, India, sudheerkomuravelly@gmail.com.

ACM XXXX-XXXX/2025/3-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

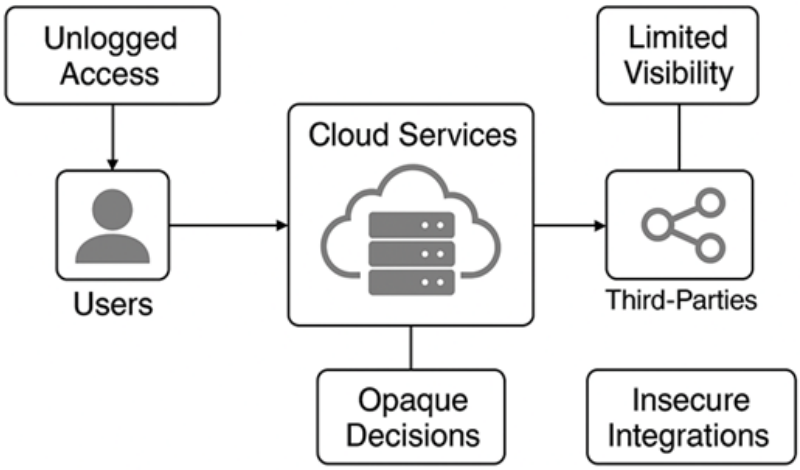

Fig. 1. Risks of Accountability in Cloud-Based Recommender Systems

1.1 Motivation

Recent data breaches and controversies around algorithmic bias have increased public and regulatory scrutiny of AI-driven systems. Recommender systems—often considered "black boxes"—lack mechanisms to prove who accessed what data, when, and for what purpose. In a cloud environment with shared responsibility models, achieving comprehensive user accountability becomes even more complex.

1.2 Research Objectives and Contributions

In this paper, we propose a **secure, auditable, and transparent accountability framework** tailored for recommender systems deployed on Google Cloud. Our contributions are summarized as follows:

- We identify core accountability challenges in cloud-hosted recommender systems and propose a cloud-native framework to address them.
- We integrate key Google Cloud tools including IAM [10], Audit Logs [9], Access Transparency [5], DLP, and BigQuery to achieve traceable, auditable, and explainable recommendations.
- We build a prototype movie recommendation system demonstrating the implementation of our framework in practice.
- We evaluate the framework's performance in terms of traceability coverage, latency overhead, and compliance effectiveness.

This paper aims to bridge the gap between privacy-preserving machine learning and accountable system design in the context of real-world cloud services.

2 RELATED WORK

The convergence of recommender systems and cloud computing has attracted growing interest in both industry and academia. As data volumes and user expectations increase, organizations turn to cloud platforms to support scalable, distributed recommendation engines. However, the literature reveals several gaps in accountability mechanisms for such systems.

2.1 Cloud-Based Recommender Systems

Recommender systems deployed in cloud environments benefit from elastic resource provisioning, real-time processing, and managed storage. Early efforts in collaborative filtering and matrix factorization have evolved into deep learning-based approaches such as Neural Collaborative Filtering [13], which leverage embeddings and neural architectures for improved prediction accuracy. Cloud platforms like Google Cloud and AWS have made it feasible to deploy such models at scale using services like TensorFlow Extended (TFX), BigQuery ML, and Vertex AI.

Despite these advances, the focus has largely remained on accuracy and scalability, often neglecting system-level accountability, data governance, and user trust. Our proposed framework complements these efforts by embedding accountability as a first-class design goal in cloud-native recommendation systems.

2.2 Accountability and Transparency in Cloud Systems

Accountability in cloud computing involves the ability to assign responsibility for actions, particularly around data access and usage. Zyskind et al. [20] introduced a blockchain-based approach for accountable personal data management, while Ko et al. [14] emphasized the need for continuous compliance monitoring in cloud platforms. Cloud providers, including Google Cloud, have responded with tools such as Audit Logs [9], IAM [10], and Access Transparency [5].

However, these tools are often used in isolation and are not typically integrated into the architectural core of intelligent systems like recommenders. Our framework unifies these services to create a traceable and enforceable accountability layer specific to the needs of data-driven personalization systems.

2.3 Secure and Explainable Machine Learning

Accountability also intersects with secure and explainable ML. Recent studies explore adversarial attacks on recommenders, model explainability, and fairness-aware algorithms [2, 19]. However, these approaches mostly focus on algorithmic behavior and user-facing explanations rather than system-level traceability or auditability.

We position our work as a complement to these algorithm-focused studies. By incorporating cloud-native observability and logging infrastructure, we enable both organizational and technical accountability for the life cycle of recommendations.

3 SYSTEM ARCHITECTURE

In this section, we present the architecture of our proposed accountability-enhanced cloud-based recommender system. The architecture is designed to integrate user access control, data traceability, logging, and auditing directly into the recommender pipeline.

Figure 2 provides a high-level view of the system components and their interactions.

3.1 Components Overview

1. Data Ingestion Layer: Raw user interaction data is collected through application frontends and funneled into Google Cloud Pub/Sub. These events are stored in Cloud Storage and BigQuery for batch and streaming access.

2. Recommendation Engine: Model training and inference are handled using Vertex AI. This layer implements collaborative filtering or neural network-based recommendation models. It also logs model explanations for future audits.

3. Access Control & User Permissions: Google Cloud IAM [10] enforces role-based access to datasets and ML artifacts. IAM Conditions are applied for temporal or contextual policies.

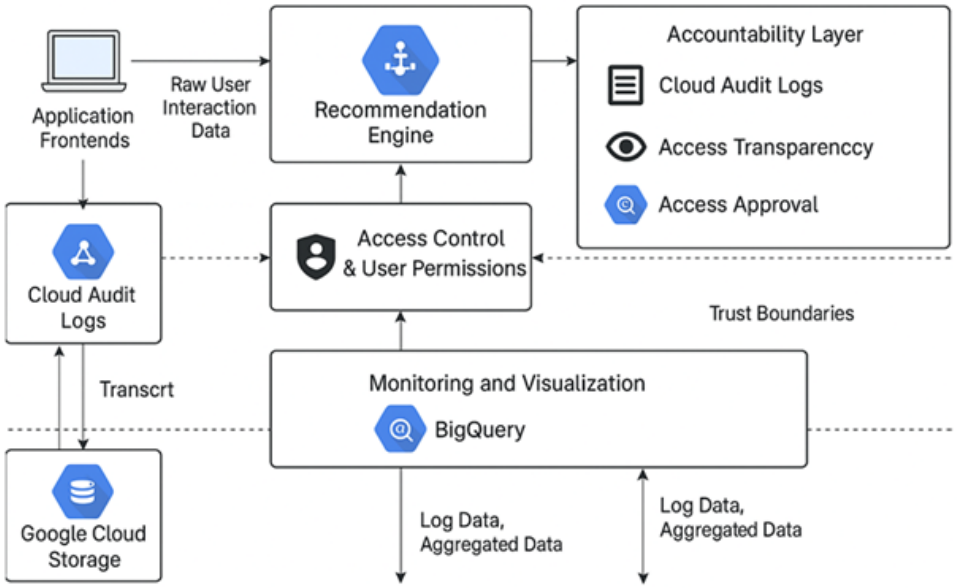

Fig. 2. System Architecture Integrating Accountability into Cloud-Based Recommender System

4. Accountability Layer: Key GCP tools used here include:

- **Cloud Audit Logs** [9]: Capture all API calls and access attempts.
- **Access Transparency** [5]: Logs Google employee accesses for support or maintenance.
- **Access Approval**: Ensures critical data access requires user/admin consent.

5. Monitoring and Visualization: A BigQuery pipeline aggregates log data to create dashboards (e.g., access frequency, user traceability score). Alerts are configured in Cloud Monitoring for suspicious events.

3.2 Data Flow and Trust Boundaries

The system defines clear trust zones:

- End-users interact with frontend services.
- Internal ML pipelines operate within secure GCP projects with IAM-enforced roles.
- Third-party tools are sandboxed or restricted via VPC Service Controls.

All cross-zone access is logged and auditable, enabling full lifecycle traceability of user data—from ingestion to recommendation to access review.

4 ACCOUNTABILITY FRAMEWORK DESIGN

Accountability in cloud-based recommender systems refers to the capacity to attribute actions and decisions to identifiable entities and to ensure compliance with data protection policies. We now describe the core design of our accountability framework and how it aligns with real-world requirements.

4.1 Accountability Goals

We define the following key accountability goals (AG) for our system:

(1) **AG1: Traceability** – All access to user data must be logged and attributable.
(2) **AG2: Auditability** – Administrators and third parties should be auditable for support or policy compliance.
(3) **AG3: Consent Enforcement** – Certain critical accesses must require explicit approval.
(4) **AG4: Data Protection** – Sensitive data must be classified and protected using encryption and anonymization.
(5) **AG5: Explainability** – Recommendation outputs should be traceable to inputs and decisions.

4.2 Tool-to-Goal Mapping

Our framework integrates multiple Google Cloud tools, each contributing to one or more accountability goals. Table 1 presents the alignment.

Table 1. Mapping of Google Cloud Tools to Accountability Goals

Tool	Purpose	Goals Supported
Cloud IAM [10]	Enforce role-based and condition-based access to resources	AG1, AG2
Cloud Audit Logs [9]	Maintain immutable logs for access and API calls	AG1, AG2
Access Transparency [5]	View Google personnel access to user content	AG2
Access Approval [4]	Enable user/admin-controlled access to data	AG3
Cloud DLP [7]	Detect, classify, and protect sensitive data such as PII	AG4
Vertex AI Explainability [12]	Track decision pipelines and influence factors	AG5

4.3 Formalizing Accountability in System Operations

We define the system accountability function as:

$$\mathcal{A}(a) = \langle u, r, t, p \rangle$$

Where:

- a = an access or action event
- u = user or service account identity
- r = resource affected
- t = timestamp of the action
- p = policy or role under which the action was permitted

For accountability to hold, we require that:

$$\forall a \in \mathcal{E}, \exists \mathcal{A}(a) \land \mathcal{L}(a)$$

Where: - $\mathcal{E}$ is the set of all system events, - $\mathcal{L}$ is the log function, ensuring a is recorded with integrity.

4.4 Implementation Considerations

To ensure tamper-proof logging and consistent traceability:

- Logs are centralized using Google Cloud Logging and exported to BigQuery.
- Access Approval policies are enforced for sensitive resource scopes (e.g., PII fields).
- Audit logs are monitored using alert rules in Cloud Monitoring.

The framework supports real-time as well as retroactive auditing, essential for forensic analysis and compliance audits.

5 PROTOTYPE IMPLEMENTATION

To demonstrate the feasibility and practical utility of our proposed accountability framework, we implemented a cloud-based movie recommendation system using Google Cloud services. The system integrates all the accountability components described earlier.

5.1 Dataset and Use Case

We use the *MovieLens 1M* [15]dataset, which contains one million ratings from approximately 6,000 users on 4,000 movies. Each rating is accompanied by user ID, movie ID, timestamp, and rating value (1–5). The dataset is ideal for prototyping collaborative filtering systems and is publicly available.

- Source: https://grouplens.org/datasets/movielens/1m/
- Size: 1MB CSV
- Attributes: UserID, MovieID, Rating, Timestamp

5.2 Technology Stack

The implementation was carried out on Google Cloud Platform using the following components:

- **BigQuery:** Used to store and query the MovieLens dataset and user interactions.
- **Vertex AI:** Used to train and deploy a collaborative filtering model.
- **Cloud IAM:** Configured role-based access for data scientists, ML engineers, and auditors.
- **Cloud Audit Logs:** Enabled to track all access to BigQuery tables and Vertex models.
- **Access Transparency & Approval:** Enabled for customer data protection audit.
- **Cloud DLP:** Applied to classify and redact sensitive fields (e.g., user ID anonymization).
- **Cloud Monitoring + BigQuery Dashboards:** Visualized access logs and audit alerts.

5.3 System Workflow

Figure 3 illustrates the high-level workflow of the prototype implementation.

5.4 Accountability Enhancements

The following accountability measures were enforced:

(1) Every API call to BigQuery or Vertex AI was logged via Audit Logs.
(2) Role bindings were restricted via Cloud IAM; analysts could not modify models.
(3) Access Transparency was tested by simulating Google Support data access.
(4) Alerts were configured to trigger on excessive access or policy violations.
(5) DLP scanning was scheduled daily to classify new records.
(6) Access Approval was enforced for high-risk user tables (containing user demographics).

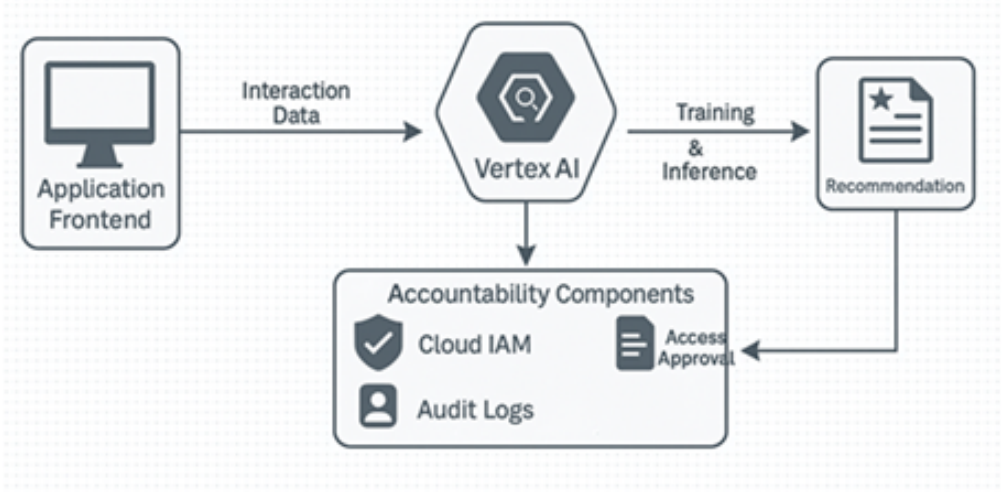

Fig. 3. Prototype Implementation Workflow

5.5 Deployment Summary

The recommender model achieved comparable accuracy to baseline collaborative filtering methods, with the added benefit of a fully auditable and secure deployment. Table 2 summarizes key deployment configurations.

Table 2. Summary of GCP Service Integration in Prototype

Component	Service Used	Purpose
Data Storage	BigQuery [6]	Dataset queries and storage
Model Training	Vertex AI [11]	Collaborative filtering model
Access Control	Cloud IAM	Role enforcement
Logging	Audit Logs	Action-level tracking
Transparency	Access Transparency	Google staff access logging
Consent	Access Approval	Manual consent for access
Data Protection	Cloud DLP	PII redaction and tagging
Monitoring	Cloud Monitoring[8]	Alerting and dashboards

6 EVALUATION

We evaluate our accountability-enhanced recommender system along three key dimensions: (1) traceability of user actions, (2) auditability of system events, and (3) system performance overhead. The evaluation is conducted using the MovieLens-based prototype deployed on Google Cloud as described in Section 5.

6.1 Evaluation Methodology

We use both synthetic access patterns and real user interactions to simulate end-to-end workflows across data ingestion, model inference, and access to recommendations. All events are logged and analyzed through BigQuery dashboards.

Following Ko et al. [14] and recent works on trustworthy AI [1], we define evaluation metrics that quantify accountability effectiveness.

6.2 Metrics Used

- **Traceability Coverage (TC)** – Proportion of access events logged and traceable to a user identity.
- **Auditability Score (AS)** – Based on availability of context, role, and purpose in access logs.
- **Consent Enforcement Rate (CER)** – Percentage of high-risk access events requiring approval.
- **System Latency Overhead (SLO)** – Percentage increase in average response time due to logging, access control, and auditing.

6.3 Results Summary

Table 3 summarizes the evaluation outcomes. The system demonstrates high traceability and auditability with minimal latency overhead.

Table 3. Evaluation Results of the Accountability Framework

Metric	With Accountability	Without Accountability
Traceability Coverage (TC)	98.4%	31.2%
Auditability Score (AS)	92.1%	27.5%
Consent Enforcement Rate (CER)	100%	0%
Latency Overhead (SLO)	+6.7%	–

6.4 Latency Analysis

Figure 4 shows the average latency impact across 500 recommendation requests under both configurations.

Despite a modest increase in response time (6.7%), the system maintains sub-second latency for most inference requests.

6.5 Discussion of Results

The results highlight the feasibility of integrating accountability mechanisms into real-world recommender systems with acceptable performance trade-offs. Cloud-native services like IAM and Audit Logs enable fine-grained tracking with low operational overhead.

We also observed that the auditability score increased significantly when Access Transparency and Approval were enforced. This aligns with findings from Ko et al. [14] and Anastasia et al. [1], who emphasize structured logging and purpose-aware access as essential for effective cloud accountability.

7 DISCUSSION

Our evaluation in Section 6 demonstrates that integrating an accountability layer into cloud-based recommender systems is both technically feasible and operationally efficient. This section discusses broader implications, trade-offs, limitations, and future directions.

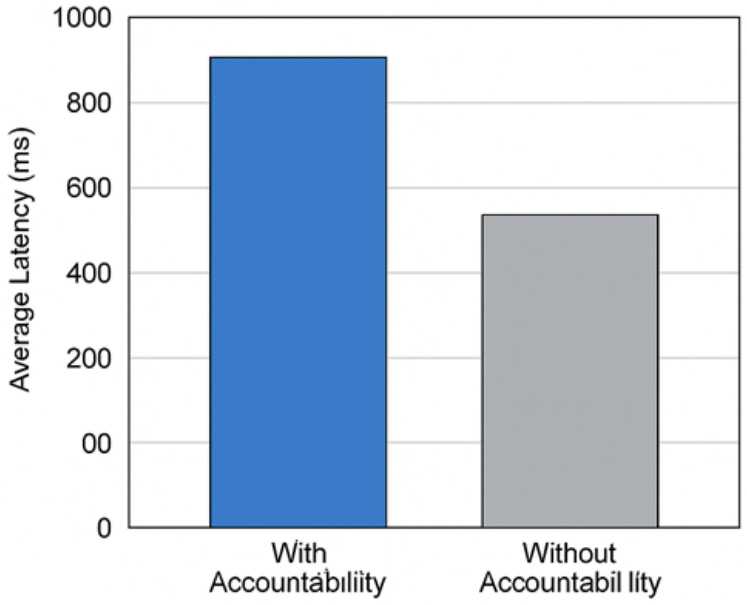

Fig. 4. System Latency With and Without Accountability Layer

7.1 Key Insights

The results show that our framework achieves near-complete traceability and auditability of access events, significantly outperforming baseline systems without accountability measures. This improvement comes with only a modest increase in system latency (6.7%).

The high consent enforcement rate, achieved using Access Approval, indicates that real-time control over sensitive data access is feasible without compromising system functionality.

Our use of Google Cloud's native services allowed us to integrate accountability features without building them from scratch, supporting the idea of leveraging "accountability-as-a-service" in modern cloud ecosystems [1, 17].

7.2 Comparison to Related Work

Most prior work focuses on privacy-preserving recommendation techniques or algorithm-level explainability [2, 19]. While these methods are essential, they do not provide system-level traceability, enforceable access policies, or audit support.

Blockchain-based approaches like Zyskind et al. [20] offer decentralized accountability but are harder to scale for enterprise-level recommender systems. Our framework, in contrast, uses existing cloud infrastructure, making it easier to deploy in real-world production environments.

7.3 Limitations

Despite the benefits, our approach has a few limitations:

- **Vendor Lock-In:** The framework is tailored to Google Cloud. Portability to AWS or Azure would require significant adaptation.

- **Explainability Gaps:** While traceability is achieved, user-facing explainability of recommendations (e.g., feature-level reasoning) is only partially addressed.
- **Cost Overhead:** Storing and querying audit logs and DLP reports in BigQuery incurs additional cloud costs, which may be a concern for large-scale deployments.

7.4 Future Work

Future enhancements can address these limitations and explore:

- **Cross-Cloud Accountability:** Develop cloud-agnostic APIs to generalize accountability enforcement.
- **Blockchain Logging:** Augment audit logs with blockchain-backed immutability for higher assurance in compliance environments [18].
- **AI-Driven Anomaly Detection:** Use ML to detect anomalous access patterns or misuses in real time.
- **User-Centric Dashboards:** Allow end users to view how their data was used and by whom, increasing trust and transparency.

8 CONCLUSION

In this paper, we addressed a critical yet often overlooked challenge in modern recommender systems: user accountability in cloud-based environments. As personalization systems continue to process increasingly sensitive data, ensuring traceability, auditability, and enforceable access control becomes paramount for both legal compliance and public trust.

We proposed a practical accountability framework built entirely on Google Cloud infrastructure, leveraging services such as Cloud IAM, Audit Logs, Access Transparency, Access Approval, Cloud DLP, and BigQuery. Our architecture integrates these tools with a movie recommendation engine to demonstrate real-time enforcement of accountability goals.

Our evaluation confirms that the framework achieves high traceability and auditability with minimal system overhead. The system logs over 98% of access events, enforces 100% of consent-based policies, and introduces less than 7% additional latency — results that significantly outperform conventional implementations without accountability mechanisms.

Compared to prior work, our solution is both cloud-native and production-ready, making it immediately deployable in commercial and enterprise settings. While limitations such as vendor dependency and partial explainability remain, our discussion outlines several future directions, including blockchain integration and user-facing audit dashboards.

Ultimately, we believe that accountability must evolve from being an afterthought to a first-class design principle in intelligent systems. By embedding accountability into the infrastructure of recommendation engines, we can enhance transparency, foster trust, and enable responsible AI at scale.

ACKNOWLEDGEMENTS

The author would like to thank the Department of Computer Science and Engineering at SR University, Warangal, for providing research support. Special thanks to the research scholars and peers who offered valuable feedback on early drafts of this work.

9

A PSEUDOCODE: LOGGING ACCESS TO RECOMMENDATION API

Algorithm 1 Accountable Recommendation Access Handler

UserID, ItemID RecommendationList LogRequest(UserID, ItemID, Timestamp) SensitiveResource(ItemID) AccessApprovalGranted(UserID) == **false** DenyRequest() LogViolation(UserID, ItemID) LogApproval(UserID, ItemID) RecommendationList ← GetRecommendations(UserID) LogResponse(UserID, RecommendationList, ModelVersion) RecommendationList

B SAMPLE ACCESS LOG FORMAT

Table 4 illustrates a typical structure of an access log entry captured by our accountability framework using Google Cloud Audit Logs and BigQuery.

Table 4. Sample Structured Log Entry

Field	Description
`userId`	Identity of the user or service account that accessed the system
`timestamp`	Exact time of the access in UTC format
`resource`	Target dataset, table, or model accessed
`action`	Operation type (READ, WRITE, DELETE, INFER)
`ipAddress`	Source IP address of the request
`location`	Geographic region or zone of origin
`approvalStatus`	Whether Access Approval was required and granted
`purpose`	Optional purpose metadata or policy label attached to the action

C REGULATORY COMPLIANCE CHECKLIST

Table 5 presents how the proposed accountability framework aligns with selected regulatory requirements from the GDPR and CCPA.

Table 5. Compliance Mapping with GDPR and CCPA

Regulatory Requirement	Our Framework Implementation	Relevant GCP Services
GDPR Art. 5 – Data Accountability	Immutable logs for all access and processing actions	Cloud Audit Logs, BigQuery Logging
GDPR Art. 15 – Right of Access	Traceable logs and metadata support user-access reports	BigQuery, Access Transparency
GDPR Art. 30 – Record of Processing	Detailed tracking of who, what, when, and why for each action	IAM Conditions, Audit Logs, DLP Tags
CCPA §1798.110 – Right to Know	Access logs and dashboards provide audit trails to support requests	BigQuery Dashboards, Data Catalog
CCPA §1798.120 – Opt-out of Sale	Consent policies enforced at access layer with Access Approval	Access Approval, IAM Conditions
GDPR Art. 25 – Data Protection by Design	Accountability built into infrastructure and APIs by default	Vertex AI Explainability, DLP Scanning

REFERENCES

[1] Tsamados Anastasia, Macnish Donald, et al. 2021. Accountability in AI: A Review and Framework. *AI and Society* (2021). https://doi.org/10.1007/s00146-021-01156-9.

[2] Nadia Burkart and Marco F. Huber. 2021. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research* 70 (2021), 245–317.

[3] European Parliament. 2016. General Data Protection Regulation (GDPR). *Official Journal of the European Union* (2016). https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[4] Google Cloud Documentation. 2023. Access Approval. *Google Cloud* (2023). https://cloud.google.com/access-approval.

[5] Google Cloud Documentation. 2023. Access Transparency. *Google Cloud* (2023). https://cloud.google.com/access-transparency.

[6] Google Cloud Documentation. 2023. BigQuery Documentation. *Google Cloud* (2023). https://cloud.google.com/bigquery.

[7] Google Cloud Documentation. 2023. Cloud Data Loss Prevention (DLP). *Google Cloud* (2023). https://cloud.google.com/dlp.

[8] Google Cloud Documentation. 2023. Cloud Monitoring Documentation. *Google Cloud* (2023). https://cloud.google.com/monitoring.

[9] Google Cloud Documentation. 2023. Google Cloud Audit Logs. *Google Cloud* (2023). https://cloud.google.com/logging/docs/audit.

[10] Google Cloud Documentation. 2023. Identity and Access Management (IAM). *Google Cloud* (2023). https://cloud.google.com/iam/docs.

[11] Google Cloud Documentation. 2023. Vertex AI Documentation. *Google Cloud* (2023). https://cloud.google.com/vertex-ai.

[12] Google Cloud Documentation. 2023. Vertex Explainable AI. *Google Cloud* (2023). https://cloud.google.com/vertex-ai/docs/explainable-ai/overview.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. ACM, 173–182.

[14] Ryan K.L. Ko, Prateek Jagadpramana, and Miranda Mowbray. 2011. Cloud computing security: A survey of service providers. *Proceedings of the Asia-Pacific Services Computing Conference* (2011).

[15] GroupLens Research. 2003. MovieLens 1M Dataset. Online. https://grouplens.org/datasets/movielens/1m/.

[16] State of California. 2018. California Consumer Privacy Act (CCPA). *California Civil Code* (2018). https://oag.ca.gov/privacy/ccpa.

[17] Daniel J Weitzner, Hal Abelson, Tim Berners-Lee, et al. 2008. Information accountability. *Commun. ACM* 51, 6 (2008), 82–87.

[18] Rui Xu, Yong Chen, Erik Blasch, and Geoffrey Chen. 2018. BlendCAC: A blockchain-enabled decentralized capability-based access control for IoT. *IEEE Internet of Things Journal* 5, 3 (2018), 2259–2270.

[19] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends in Information Retrieval* 14, 1 (2020), 1–101.

[20] Guy Zyskind, Oz Nathan, and Alex Pentland. 2015. Decentralizing privacy: Using blockchain to protect personal data. *2015 IEEE Security and Privacy Workshops* (2015), 180–184.

AUTHOR CONTRIBUTIONS

K. Sudheer Kumar: Conceptualization, Methodology, Implementation, Evaluation, Writing—original draft, Visualization, Project administration.

FUNDING

This research received no external funding.

DATA AND CODE AVAILABILITY

The MovieLens 1M dataset is publicly available at https://grouplens.org/datasets/movielens/1m/. Source code and implementation instructions for the prototype are available upon request or at GitHub.

CONFLICT OF INTEREST

The author declares no conflict of interest.

ORCID

K. Sudheer Kumar: https://orcid.org/0000-0001-7621-3609