Addressing Metadata Synchronization Challenges: Iceberg Rest Catalog as an Interoperability Enabler

Pravin Kumar Dwiwedi

Abstract—**Apache Iceberg [1] is a high-performance open-source table format for large analytic tables. I was developed at Netflix [2] and donated to the Apache Foundation. This research examines the importance of Iceberg REST Catalog (IRC) [3] for Iceberg Open Table format. This supports implementation for achieving true cross-platform interoperability in modern data lake architectures. The study demonstrates that IRC provides cross-platform interoperability without creating data copies at consumer end. This supports near real-time data consumption. This also enables the significant cost reductions in storage, operational, and engineering expenses. Based on OpenAPI specification, IRC offers centralized security policies across analytics platforms and tools. Through comparative analysis of integration approaches between AWS Data Lake and Snowflake, this paper illustrates how IRC eliminates data duplication, maintains single source of truth for the data. It provides unified access policy enforcement while avoiding vendor lock-in.**

Index Terms—**Apache Iceberg, IRC, REST Catalog, Cross-Platform interoperability, Data Lake, Metadata Management, Open Table Format, Analytical Data Platforms, Modern Data Lake**

I. INTRODUCTION

A. Background and Motivation

Apache Iceberg is a high performance open-source table format for large analytic tables. It was developed by Netflix and then donated to Apache Foundation as Open Source. It has the core features like multi query engine support, expressive SQL, full schema evolution, hidden partitioning, time travel and rollback and compaction. Apart from these core features, Apache Iceberg provides cross-platform interoperability through Iceberg REST Catalog (IRC). Unlike Hive Table Format [4], Apache Iceberg manages the metadata in the files on the storage. These metadata files make Iceberg so special.

In recent years industry has been modernizing the data lake using Apache Iceberg for the data lake table format. Key business drivers behind Iceberg adoption include multi-platform strategy, unified and scalable data governance and compliance support, scalability for large scale analytical data workloads, schema evolution without downtime, operational support with ACID transactions [5], vendor agnostic ecosystem support, and performance and cost optimization.

IRC using practical use cases, demonstrate opportunities to save huge storage, operational and engineering costs, and discuss business propositions for IRC adoption. The study also examines the availability of IRC in the market with associated pros and cons.

B. Problem Statement

IRC is comparatively a very new feature from Apache Iceberg and still evolving. Many vendors are still implementing it in their catalog service. There are limited resources on the Internet which explain the IRC and how it works under the hood. Understanding IRC implementation is crucial for organizations seeking to optimize their data lake architectures while maintaining cross-platform interoperability.

C. Research Objectives

This article focuses on how Iceberg REST Catalog works for cross platform interoperability. The research aims to illustrate

II. LITERATURE REVIEW AND TECHNOLOGY BACKGROUND

A. Apache Iceberg REST Catalog Overview

IRC is a REST-based Catalog API for managing table metadata and performing catalog operations. This has the endpoints as per OpenAPI Specification [6] available for implementation. These endpoints are called by the clients (platforms/query-engines) to retrieve the credentials and metadata. Apart from retrieving the required details, there are certain APIs which support Create/Delete/Drop operations on the

Iceberg catalog. The REST Catalog provides the interface through the available API endpoints on top of existing underlying technical Apache Iceberg Catalogs like AWS Glue [7], Polaris [8], Unity [9] etc.

B. Security and Access Control Features

IRC provides a robust security mechanism through Multi-Layer Authentication, Token-Based Access, and OAuth2.0 [10]. It supports Fine Grain Access Control and can integrate Enterprise Security Systems to support RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control).

For example, if AWS Lake Formation is implemented, IRC integrates very well to respect all the access policies and grants. It also provides audit and logs capabilities for Schema Change Auditing, Data Access Patterns, and API Request Auditing. In AWS [11] implementations, CloudWatch [12] and CloudTrail [13] logging services capture the audit logs.

III. METHODOLOGY AND SYSTEM DESIGN

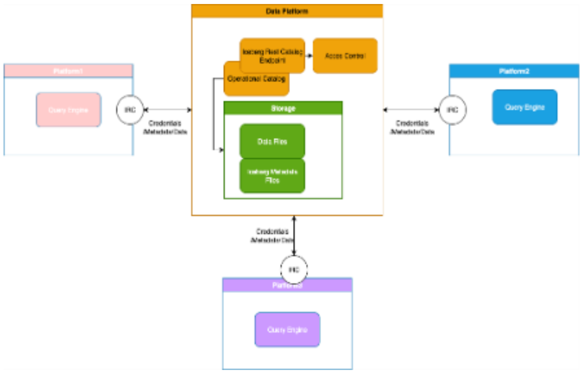
A. Illustration Scenario

The research methodology involves analyzing a practical scenario where two analytical platforms (P1, P2) require data sharing with unified access controls. Platform P1 contains data, metadata and required access controls, while Platform P2 needs to access data from P1 following enforced access controls. For this illustration, P1 represents AWS Data Lake and P2 represents Snowflake [14] analytical platform.

The system design includes a Data Lake with Apache Iceberg Open Table Format in AWS Environment, cataloged through AWS Glue Catalog with S3 [15] as storage. I have used AWS Lake Formation [16] to enforce the access policies. Snowflake is example of cross platform analytics platform which accesses AWS data lake data stored in Iceberg Table Format.

B. Hub and Spokes Architecture

Though the illustration focuses on two platforms, real-world implementations can accommodate multiple platforms using a "hub and spoke" model. The platform containing data in Iceberg table format, operational catalog, and access control policy enforcement serves as the hub. All other platforms requiring read/write operations to/from the hub function as spokes.



IV. IMPLEMENTATION APPROACHES AND ANALYSIS

A. Prerequisites

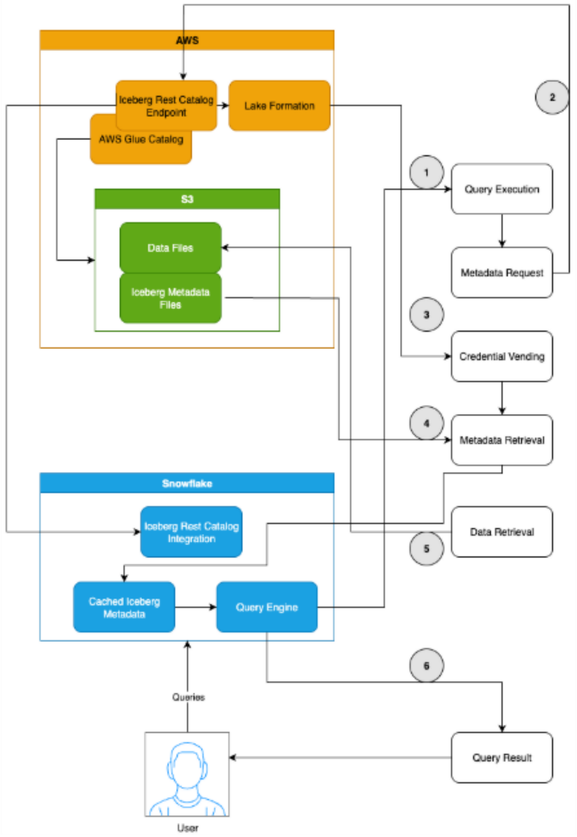
Required implementation prerequisites include:

- Required IAM role and permissions at AWS end in place
- Create trust relationship between AWS and Snowflake
- Create catalog integration for AWS Glue Iceberg REST endpoint at Snowflake end

B. Credentials, Data, Metadata Flow Process

Once prerequisites are completed, when analytical data platform (Snowflake) users write queries to fetch data from AWS Data Lake, the following steps occur in sequence:

1. Query Execution: Query engine (Snowflake) attempts to execute user-provided query to fetch data from source data platforms (AWS Data Lake) Iceberg Tables.
2. Metadata Request: The system requires metadata to fetch data, sending request to IRC endpoint to retrieve Iceberg metadata.
3. Credential Vending: IRC endpoint communicates with policy enforcement engine (Lake Formation) and vends credentials (for accessing both data and metadata) to the query engine.
4. Metadata Retrieval: Query engine uses the endpoint to retrieve required/authorized metadata from Iceberg metadata files (rather than from Glue Catalog) and maintains it in cache.
5. Data Retrieval: Query engine uses retrieved Iceberg metadata and reads authorized data from storage location.
6. Query Result: Results are provided back to users.



If you closely see the above first diagram you can see the consumption platform (Snowflake) does not hold the data nor copy the Iceberg Metadata/Catalog. Access control policies are enforced at the data lake data platform (AWS).

The consumer data platform does not talk to the vendor catalog (AWS Glue Catalog) to get the metadata or data. This solves the challenges we discussed in the above sections.

V. BUSINESS VALUE PROPOSITION

Most of the time you need to convince your business team about the IRC for compliance and regulation. Few specific business domains like banking & finance, healthcare have very strict guidelines before adopting any tools/services/offerings. These companies evaluate each tool before adopting it. These business value propositions will help you to get the approval for the IRC.

- Cost Reduction
  - Eliminate data duplication and reduce storage costs by 60-80%.
  - A single source of truth reduces infrastructure & engineering overhead.
  - Faster analytics deployment cycles.
  - ROI within 3-6 months through above.
- Faster Time-to-Insights
  - Analysts access data immediately without ETL delays.
  - Real-time data availability across all platforms.
- Enhanced Security
  - Centralized security policies enforced consistently across all analytics platforms and tools.
- Business Agility
  - Deploy new analytics tools without data migration.
  - Support changing business requirements rapidly.
- Risk Mitigation
  - Avoid vendor lock-in.
  - Maintain data portability and reduce dependency on single platforms.
- Future-Proof Architecture
  - Open standards ensure compatibility with emerging technologies and analytics platforms.

VI. STRATEGIC POSITIONING

- Enables data democratization across the organization through centralized data and metadata and open interoperability through OpenAPI specifications.
- Supports rapid innovation and new use case deployment.
- Provides future-proof architecture for emerging technologies as it is based on fast growing open-source contributions.
- Provides quantifiable business results and impacts and achieves systematic implementation.

A. IRC Providers

Current market providers include:

- Tabular [17]: Commercial Catalog service from Tabular
- Starburst [18]: Commercial Enterprise Trino
- Dremio [19]: Commercial Data Lakehouse platform from Dremio
- Databricks [20]: Commercial Unity Catalog
- Project Nessie [21]: Open-Source catalog with data versioning
- Apache Polaris: Open catalog from Snowflake

B. Detailed Feature Comparison (See Last Page)

VII. FUTURE SCOPE

A. AI/ML Use Case

Iceberg REST Catalog is an open source-based specification which works with Iceberg Open Table Format. This design can be extended to the AI/ML feature platforms to store the data for training and inferences as well as cross platform consumptions. AI models required to train in another platform can use this feature to consume the data. This will remove the data copy requirement. REST end points can be well integrated with any of the guardrail's solution for ML governances.

B. Implementation Considerations

Evaluation of the vendors' products are required to validate the support of IRC features and capabilities. Validation should be based on the organization use case and domain not only technical. Spoke & Hub model can be extended to any level based on the use cases and number of platforms.

C. Security and Compliance

For the research and illustration purpose I have explained very high level about security and compliances. For production, you should be exploring this aspect into deep. Several tools and algorithms can be evaluated which can support the IRC.

VIII. CONCLUSION

In this article we talked about Apache Iceberg REST Catalog. IRC is crucial for cross platform interoperability as well as storage, operational, engineering cost optimizations security, audits and logs are supported by IRC so it can be implemented in those domains where strict regulatory compliances are required. IRC is an open-source project, so it is scalable and fast growing. All the major cloud providers support the IRC so implementations should not be a big challenge.

I have illustrated the integration between AWS Data Lake and Snowflake, but core remains the same for other platforms interoperability.

REFERENCES

[1] Apache Software Foundation, "Apache Iceberg - Open Table Format," Apache Iceberg Documentation, 2024. [Online]. Available: https://iceberg.apache.org/

[2] Netflix Technology Blog, "Apache Iceberg: An Architectural Look Under the Hood," Netflix Tech Blog, 2021. https://github.com/Netflix/iceberg

[3] Apache Software Foundation. Apache Iceberg REST Catalog Specification. Apache Iceberg Project. Available: https://iceberg.apache.org/rest-catalog-spec/. 2024.

[4] Apache Software Foundation. Apache Hive. Apache Hive Project. Available: https://hive.apache.org/. 2024.

[5] MinIO Inc. ACID Transactions in Apache Iceberg. MinIO Blog. Available: https://blog.min.io/iceberg-acid-transactions/. 2024.

[6] OpenAPI Initiative, "OpenAPI Specification," OpenAPI Specification Documentation, 2024. [Online]. Available: https://spec.openapis.org/

[7] Amazon Web Services, "AWS Glue Data Catalog," AWS Documentation, 2024. [Online]. Available: https://aws.amazon.com/glue/

[8] Apache Software Foundation. Apache Polaris. Apache Polaris Project. Available: https://polaris.apache.org/. 2024.

[9] Databricks Inc. Unity Catalog. Databricks Product Documentation. Available: https://www.databricks.com/product/unity-catalog. 2024.

[10] Apache Software Foundation. Apache Polaris REST Catalog OpenAPI Specification v0.9.0-incubating. Apache Polaris Project GitHub Repository. Available: https://raw.githubusercontent.com/apache/polaris/refs/tags/apache-polaris-0.9.0-incubating/spec/rest-catalog-open-api.yaml. 2024.

[11] Amazon Web Services Inc. Amazon Web Services. AWS Official Website. Available: https://aws.amazon.com/. 2024.

[12] Amazon Web Services Inc. Amazon CloudWatch. AWS CloudWatch Service

Documentation. Available: https://aws.amazon.com/cloudwatch/. 2024.

[13] Amazon Web Services Inc. AWS CloudTrail. AWS CloudTrail Service Documentation. Available: https://aws.amazon.com/cloudtrail/. 2024.

[14] Snowflake Inc., "Snowflake Data Platform," Snowflake Documentation, 2024. [Online]. Available: https://docs.snowflake.com/

[15] Amazon Web Services Inc. Amazon Simple Storage Service (S3). AWS S3 Service Documentation. Available: https://aws.amazon.com/s3/. 2024.

[16] Amazon Web Services Inc. AWS Lake Formation. AWS Lake Formation Service Documentation. Available: https://aws.amazon.com/lake-formation/. 2024.

[17] Tabular Technologies Inc. Tabular Catalog. Tabular Platform Documentation. Available: https://www.tabular.io/. 2024.

[18] Starburst Data Inc. Starburst Enterprise Platform. Starburst Data Lakehouse Platform. Available: https://www.starburst.io/. 2024.

[19] Dremio Corporation, "Iceberg REST Catalog," Dremio Documentation, 2024. Available online: https://docs.dremio.com/current/data-sources/lakehouse-catalogs/iceberg-rest-catalog/.

[20] Databricks Inc., "Databricks Data Intelligence Platform," Databricks Official Website, 2024. Available online: https://www.databricks.com/,

[21] Project Nessie Community, "Configure Nessie with Iceberg REST," Project Nessie Guides, 2024. Available online: https://projectnessie.org/guides/iceberg-rest/

B. Detailed Feature Comparison

Feature	Tabular	Starburst	Dremio	Databricks Unity Catalog	Project Nessie	Apache Polaris
Catalog Type	Commercial	Commercial	Commercial	Commercial	Open Source	Open Source
Multi-Engine Support	Yes	Yes (with Trino)	Yes	Yes (Iceberg REST API)	Yes (Spark, Flink, Trino)	Yes (Spark, Flink, Trino, Doris, StarRocks)
Primary Focus	Managed Iceberg service	Enterprise Trino	Data lakehouse platform	Unified data governance, Delta Lake & Iceberg	Git-like version control	Open, vendor-neutral catalog
Key Features	Scalable, fully managed Iceberg catalog.	Trino-optimized Iceberg connector.	Automated maintenance, built-in Iceberg catalog.	UniForm (Delta & Iceberg interoperability), credential vending.	Catalog-level versioning, multi-table transactions, branching, and merging.	Unified catalog for Iceberg and Delta Lake, cross-cloud support.
Data Governance	Centralized, secure access control.	Integration with security frameworks.	Centralized, secure access control.	Fine-grained access control with SQL, auditing.	Version control for auditing, rollback.	Fine-grained access control with SQL, auditing.
Deployment	SaaS	Self-managed or SaaS	Self-managed or SaaS	SaaS	Self-managed	Self-managed
Read/Write Access	Full read/write support.	Full read/write support.	Full read/write support.	Read/write to managed tables; read-only to foreign tables.	Full read/write support.	Full read/write support.
Notable Limitations	N/A	N/A	N/A	No row-level deletes on foreign tables, limited partition evolution.	Primarily focused on versioning.	Still an incubating project.