databricks

Governance with Unity Catalog

for data and AI

Gopala Raju
Sr. Solutions Architect

Gopala Raju

Sr. Solutions Architect - APJ Governance Lead

- Nearly 2 years with Databricks
- 19+ years of experience
- 10+ years of experience in Data and Analytics (at AWS and other companies)
- I am the Governance Practice Lead for APJ, working closely with field teams on all aspects related to Governance

Agenda

- Introduction
- Governance with Unity Catalog
- Key capabilities
- Unity Catalog and cloud providers
- Upgrading to Unity Catalog
- Demo

Housekeeping

- This presentation will be recorded and we will share these materials after the session, within 48 hours
- There are no hands-on components so you only need to take notes
- Use the Q&A function to ask questions
- If we do not answer your question during the event, we will follow-up with you afterwards to get you the information you need!
- Please fill out the survey at the end of the session so that we can improve our future sessions

Data and AI governance drives business value

"Organizations are finally realizing the value of data as an asset that needs to be protected, managed and maintained to increase asset value"

—

IDC

"Organizations seeing the highest returns from AI have a framework for AI governance to cover every step of the model development process"

—

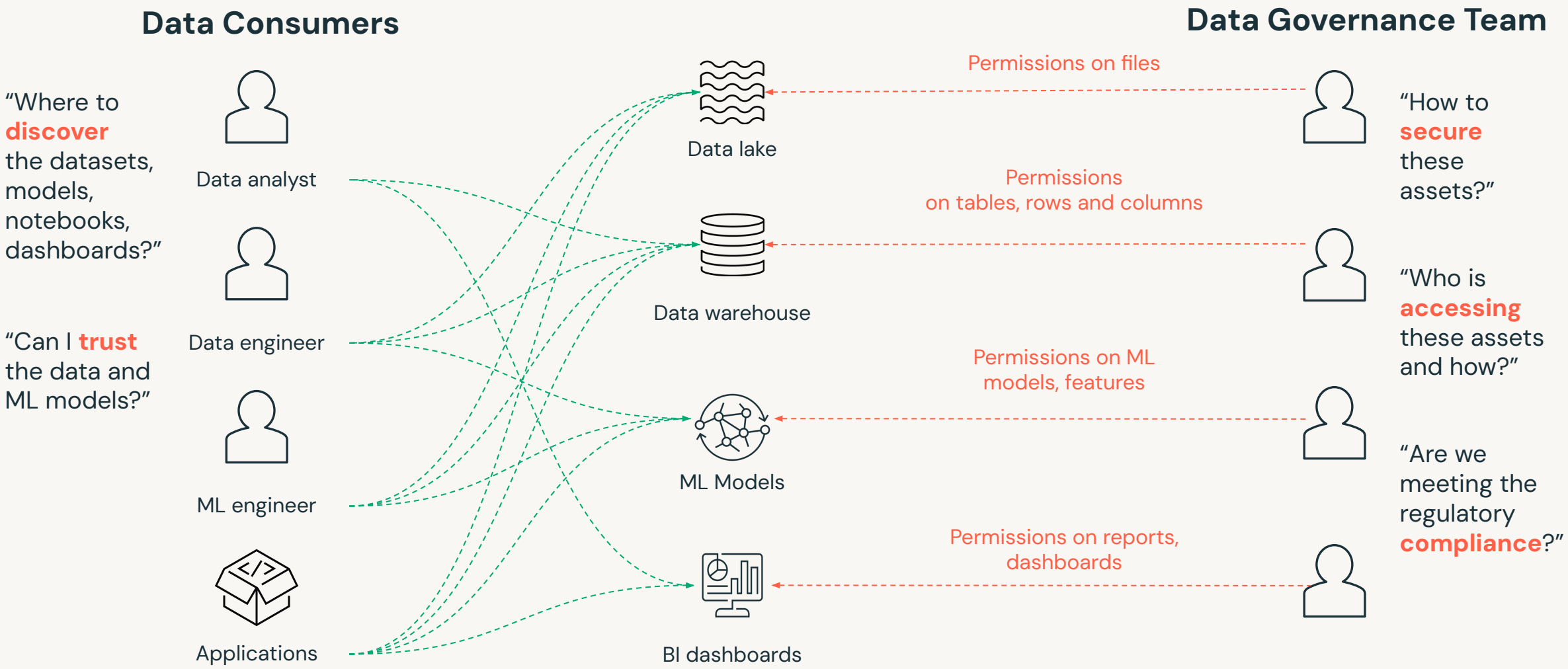
The State of AI in 2022, McKinsey & Co

"AI is now an enterprise essential, and as such, AI governance will join cybersecurity and compliance as a board-level topic"

—

Forrester, 2023 AI Predictions report

Today, data and AI governance is complex



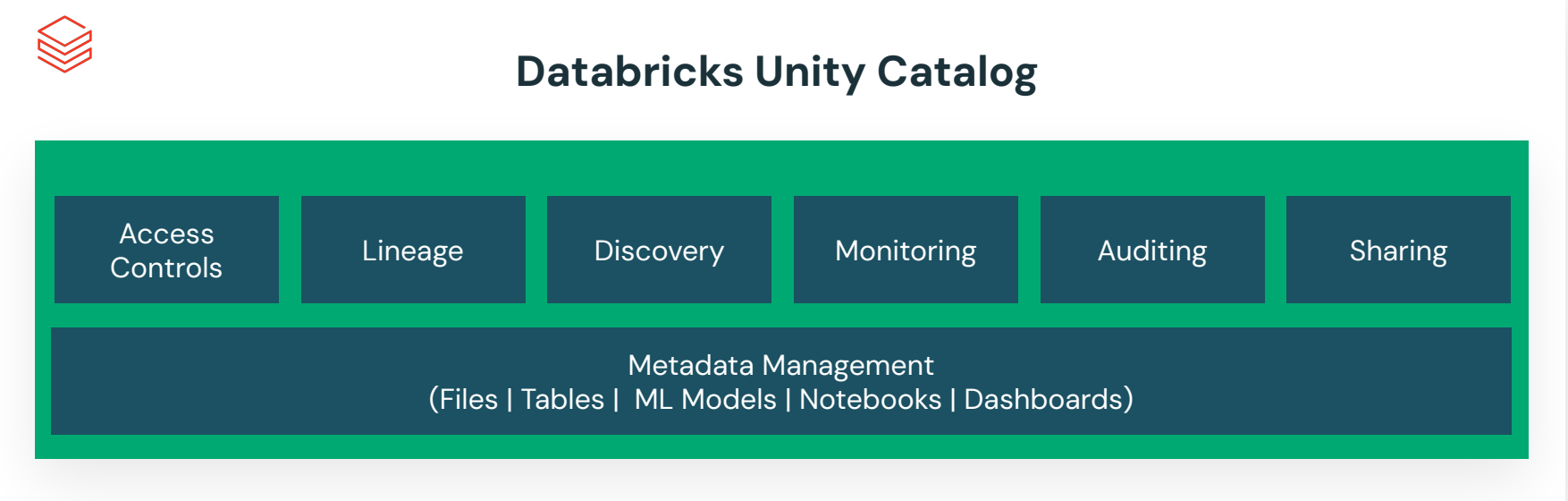
Databricks Unity Catalog
Unified governance for data and AI

Unified visibility into data and AI

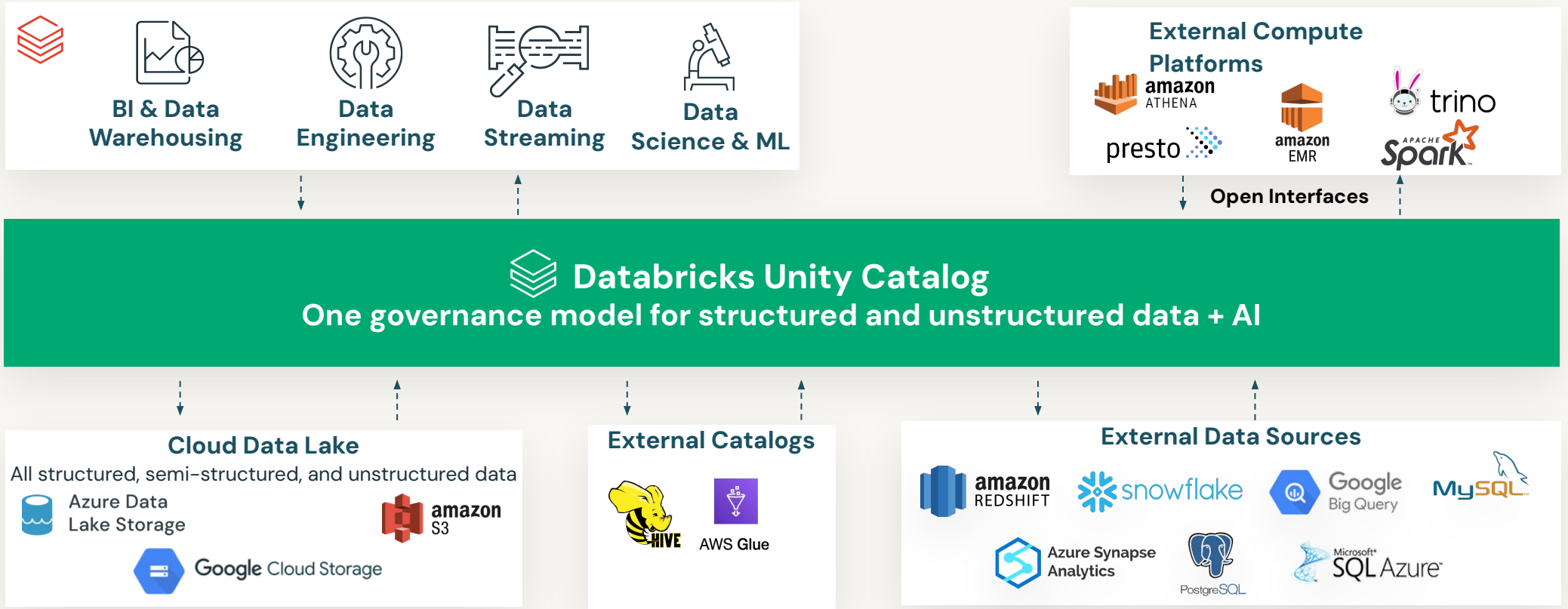
Single permission model for data and AI

AI-powered monitoring and observability

Open data sharing

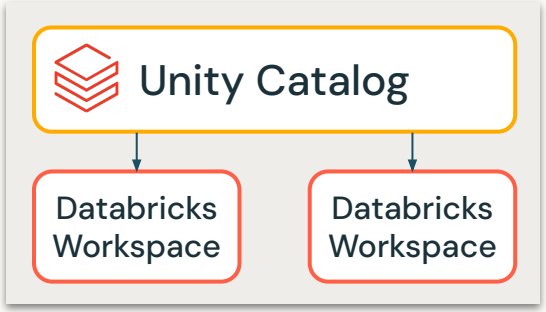


Databricks Lakehouse unifies data and AI governance



Unity Catalog – Key Capabilities

- Centralized metadata and user management
- Centralized access controls
- Lakehouse federation for external data sources
- Data search, discovery through tags
- Lineage
- AI powered detection and classification (through Okera acquisition)
- Data access auditing
- Observability through system tables
- Secure data sharing with Delta Sharing
- Marketplace for data, analytics and AI




```
GRANT ... ON ... TO ...
REVOKE ... ON ... FROM ...
```

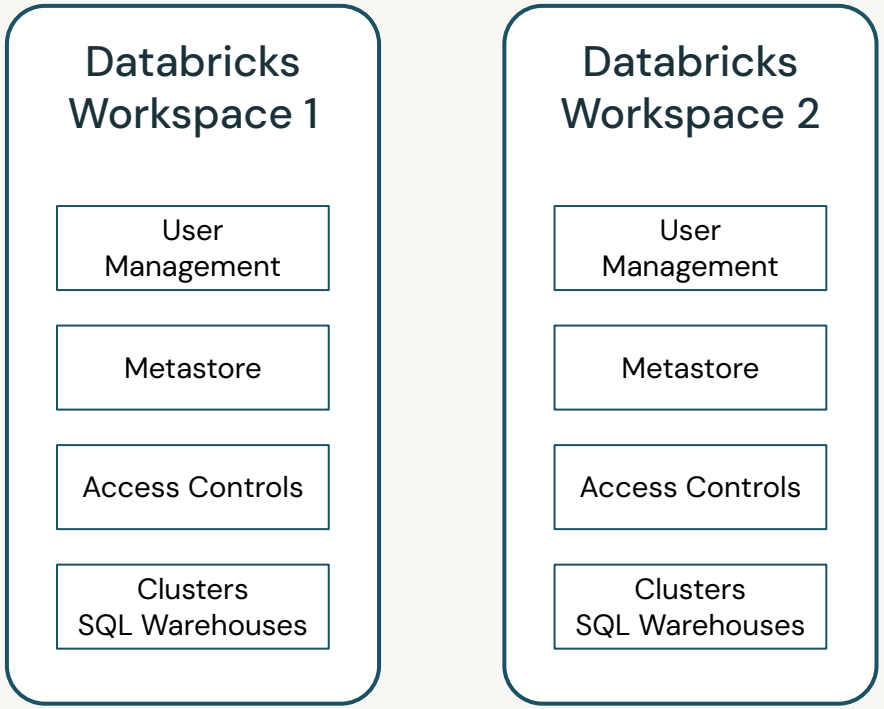
Catalogs, Databases (schemas), Tables, Views, Storage credentials, External locations

Centralized metadata and controls

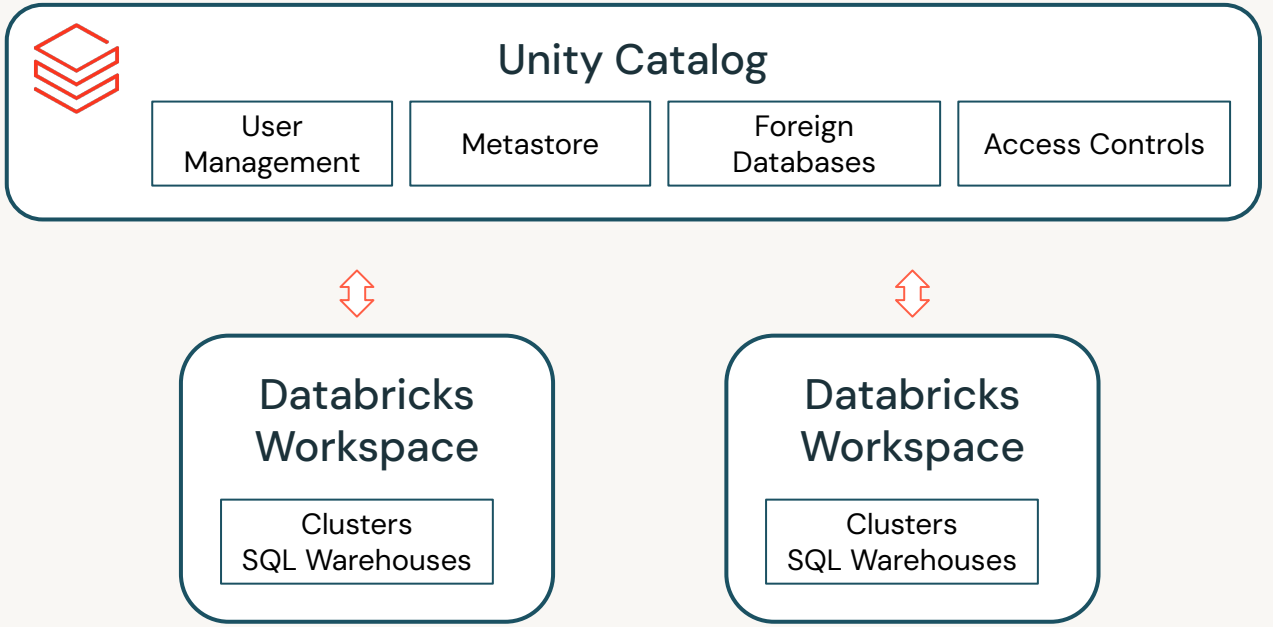
All your metadata, in one place

One metadata layer across file and database sources **superpowers** governance

Without Unity Catalog



With Unity Catalog



Centralized Access Controls

Centrally grant and manage access permissions across workloads and foreign databases

Using ANSI SQL DCL

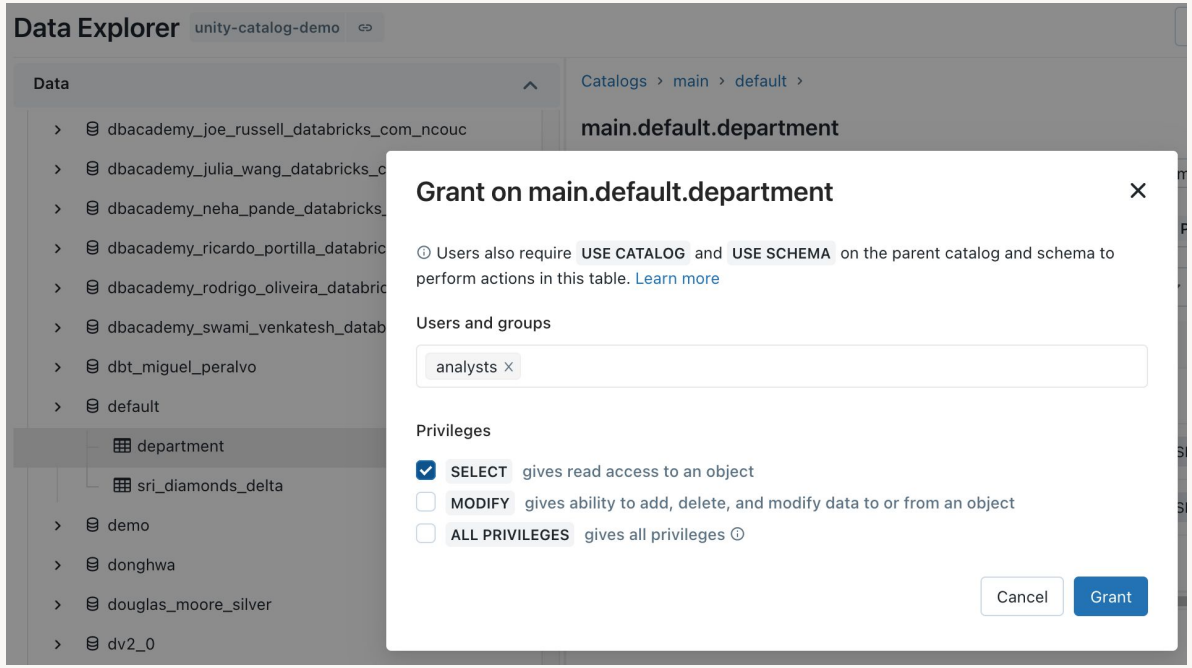
```
GRANT <privilege> ON <securable_type>
<securable_name> TO `<principal>`

GRANT SELECT ON iot.events TO engineers
```

Choose permission level

'Table'= collection of files in S3/ADLS

Sync groups from your identity provider

Using UI



Row Level Security and Column Level Masking

Provide differential fine grained access to file based datasets and foreign tables

In Preview

Only show specific rows

```
CREATE FUNCTION <name> ( <parameter_name >
<parameter_type> .. )
RETURN {filter clause whose output must be a boolean}


CREATE FUNCTION us_filter(region STRING)
RETURN IF(IS_MEMBER('admin'), true, region="US");

ALTER TABLE sales SET ROW FILTER us_filter ON region;
```

Test for group membership

Assign reusable filter to table

Specify filter predicates

Mask or redact sensitive columns

```
CREATE FUNCTION <name> (<parameter_name>,
<parameter_type>, [, <column>...])
RETURN {expression with the same type as the first
parameter}

CREATE FUNCTION ssn_mask(ssn STRING)
RETURN IF(IS_MEMBER('admin'), ssn, "****");

ALTER TABLE users ALTER COLUMN table_ssn SET MASK
ssn_mask;
```

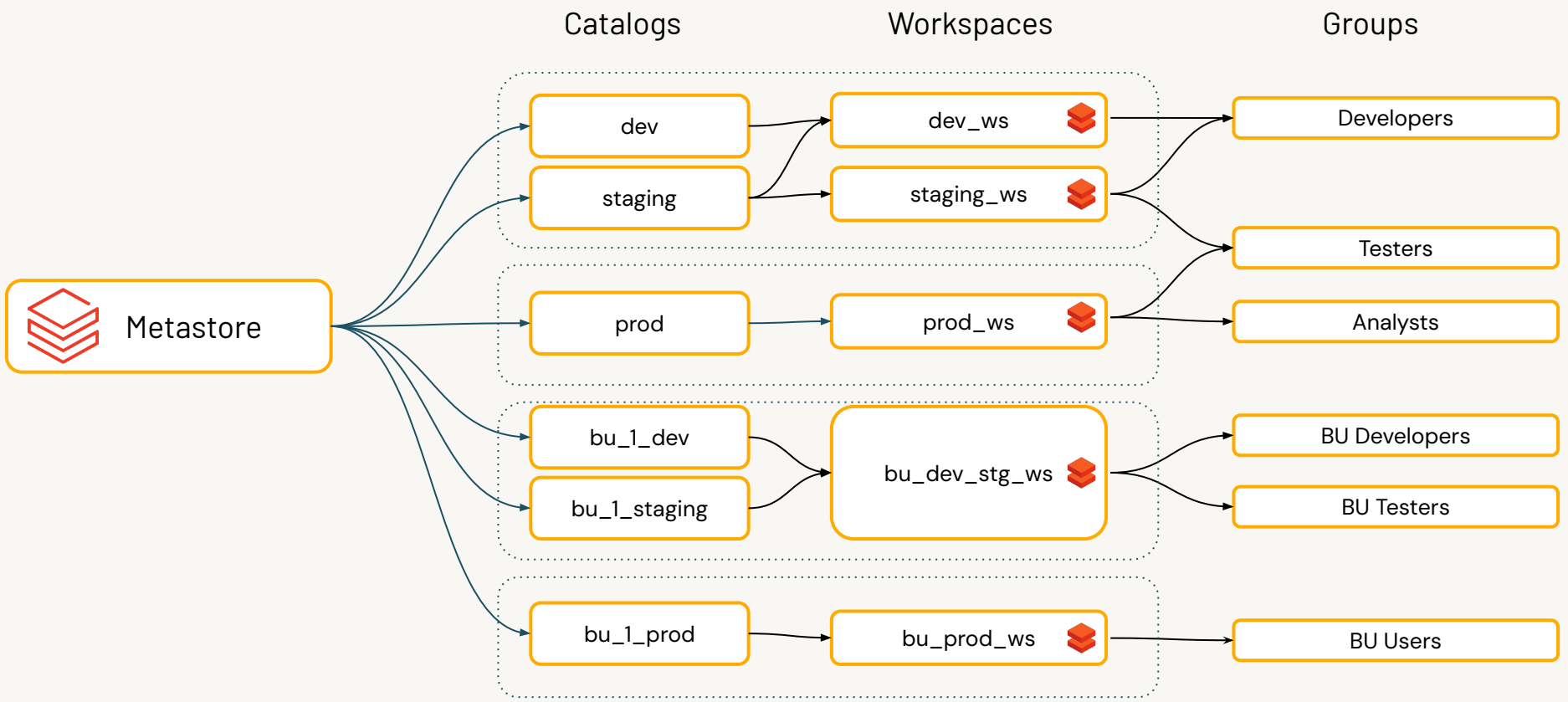
Test for group membership

Assign reusable mask to column

Specify mask or function to mask

Access data from specified environments only

Restrict catalog access by environment or purpose



Access to data and availability of data can be isolated across workspaces and groups

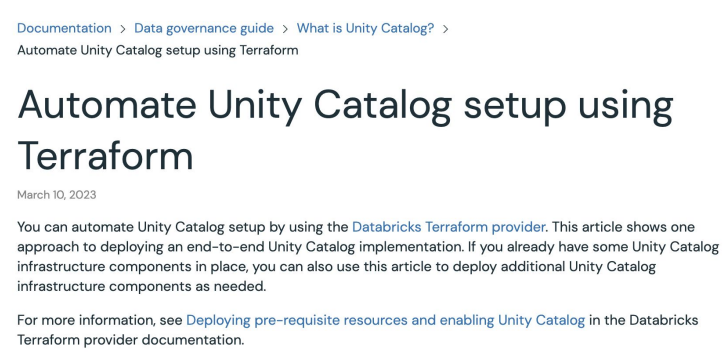
In Preview

High Leverage Governance with Terraform & APIs

Use data-sec-ops, policies as code patterns to scale your efforts

- Privileges for UC objects can be managed programmatically using our Terraform provider, especially for teams already using Terraform
- This will pair naturally with the management of the UC objects (Metastore, Catalog, Assignments etc.) themselves.

(If not already using Terraform, maybe now is a good time!)


Documentation > Data governance guide > What is Unity Catalog? >
Automate Unity Catalog setup using Terraform

Automate Unity Catalog setup using Terraform

March 10, 2023

You can automate Unity Catalog setup by using the Databricks Terraform provider. This article shows one approach to deploying an end-to-end Unity Catalog implementation. If you already have some Unity Catalog infrastructure components in place, you can also use this article to deploy additional Unity Catalog infrastructure components as needed.

For more information, see Deploying pre-requisite resources and enabling Unity Catalog in the Databricks Terraform provider documentation.

```
resource "databricks_grants" "sandbox" {
  provider = databricks.workspace
  catalog = databricks_catalog.sandbox.name
  grant {
    principal = "Data Scientists"
    privileges = ["USAGE", "CREATE"]
  }
  grant {
    principal = "Data Engineers"
    privileges = ["USAGE"]
  }
}
```

Governance for file-based and database sources

Query Federation

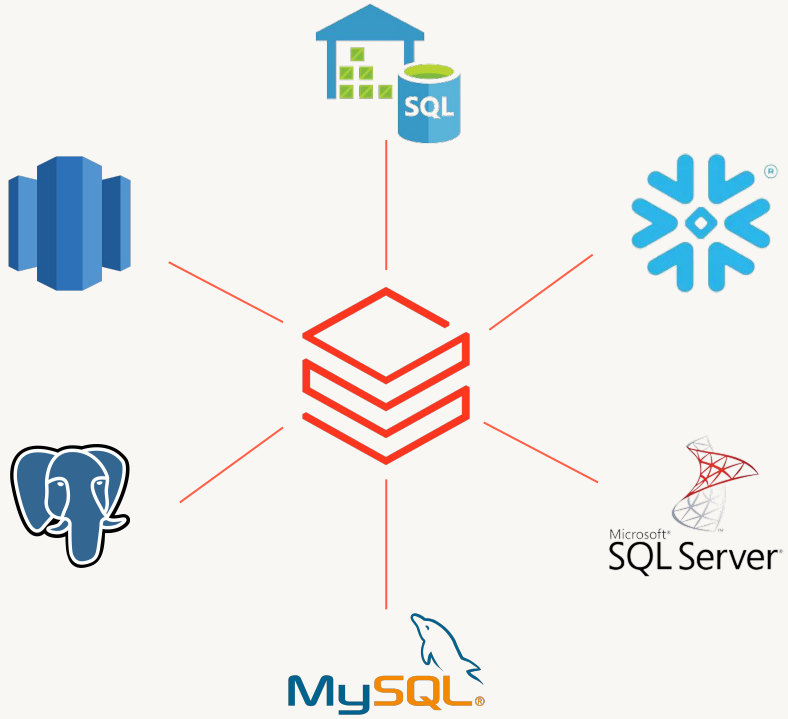
Unify your entire data estate with lakehouse

In Preview

Query Federation provides **one single point of secure access to all your data** – no matter where it lives – and one way to access, catalog, govern, and query all your data – **no ingestion required**.

- **Unified permission controls**
- Intelligent pushdown optimizations
- Accelerated query performance with Materialized Views
- Support for R/O operations today

```
CREATE FOREIGN CATALOG <catalog_name>
USING CONNECTION <connection_name>
OPTIONS (database '<remote_database>')

SELECT * FROM <catalog_name>.<schema_name>.<table_name>
```



In Preview

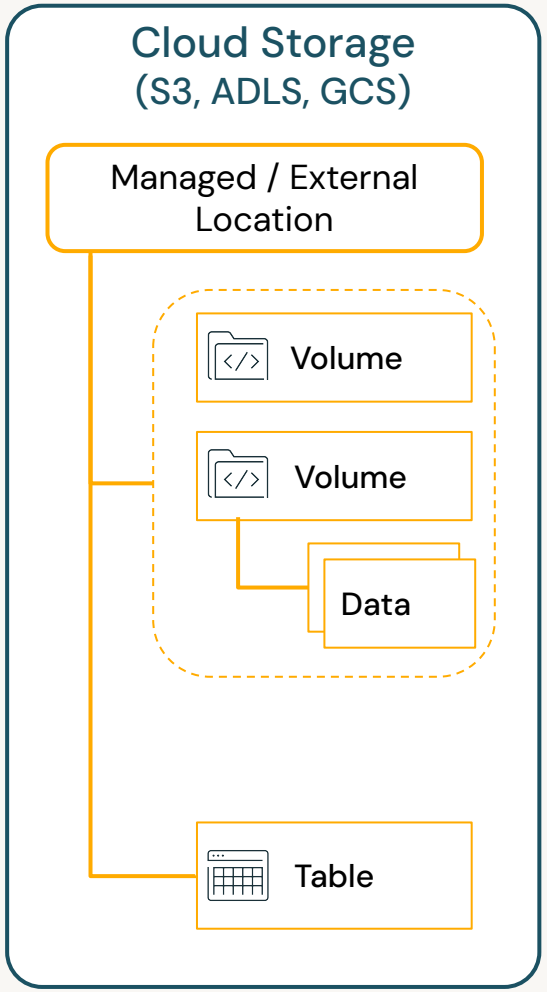
Volumes in Unity Catalog

Access, store, organize and process files with Unity Catalog governance

- Volumes can be accessed by some POSIX commands

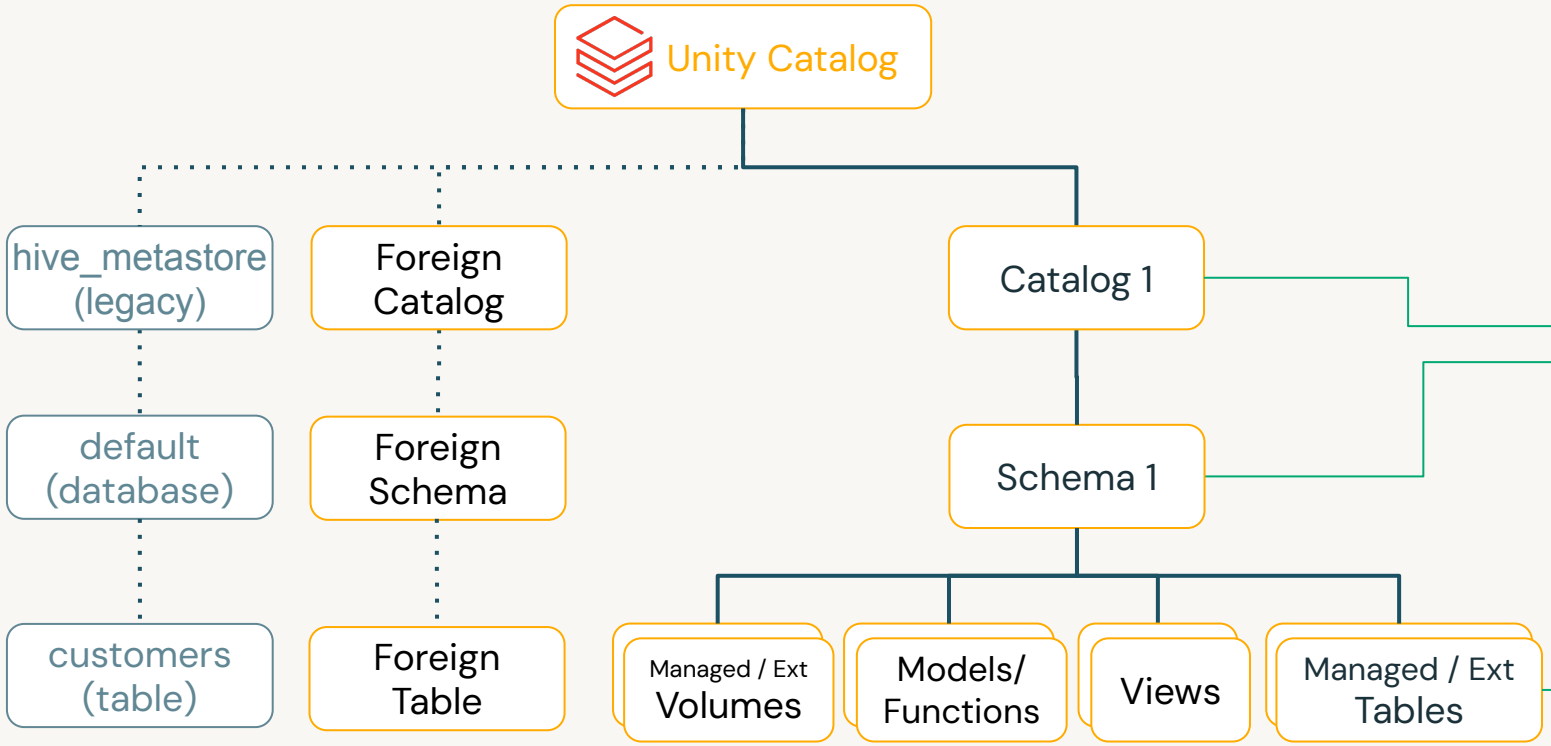
```
dbutils.fs.ls("s3://my_external_location/Volumes/catalog/schema/volume123")

ls /Volumes/catalog/schema/volume123
```

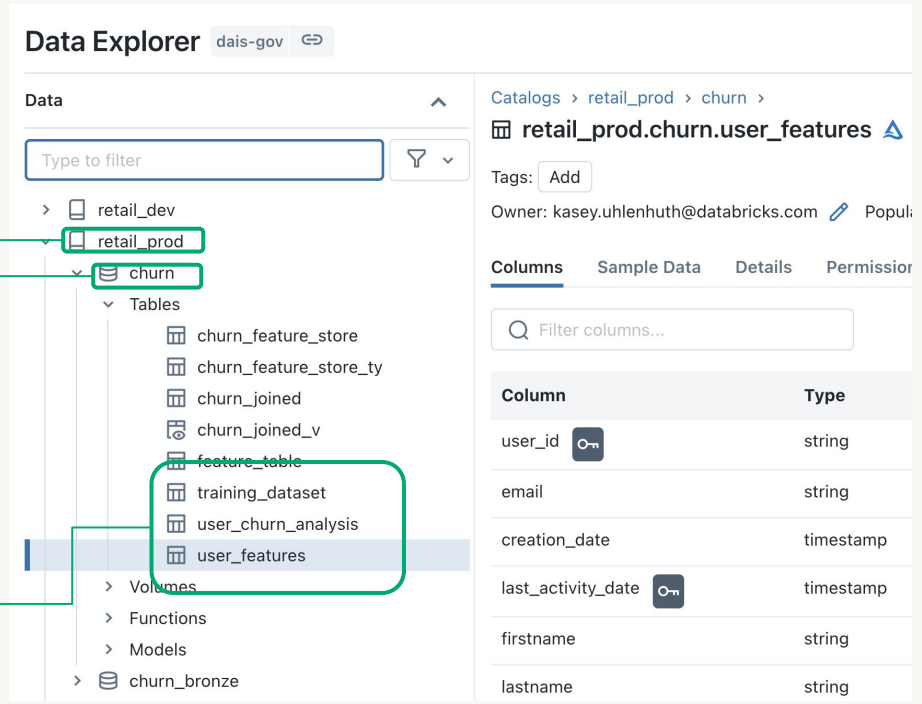
- Volumes are created under Managed or External Locations and show up in UC Lineage
- Volumes add governance over non-tabular data sets
  - Unstructured data, e.g., image, audio, video, or PDF files, used for ML
  - Semi-structured training, validation, test data sets, used in ML model training
  - Raw data files used for ad-hoc or early stage data exploration, or saved outputs
  - Library or config files used across workspaces
  - Operational data, e.g., logging or checkpointing output files
- Tables are registered in Managed / External Locations, not in Volumes



Governed namespace across file and database sources

Access legacy metastore and foreign databases powered by Query Federation `






```
SELECT * FROM main.paul.red_wine;  -- <catalog>.<database>.<table>

SELECT * FROM hive_metastore.default.customers;

SELECT * FROM snowflake_warehouse.some_schema.some_table;
```

Fundamental Concepts

Working with file based data sources

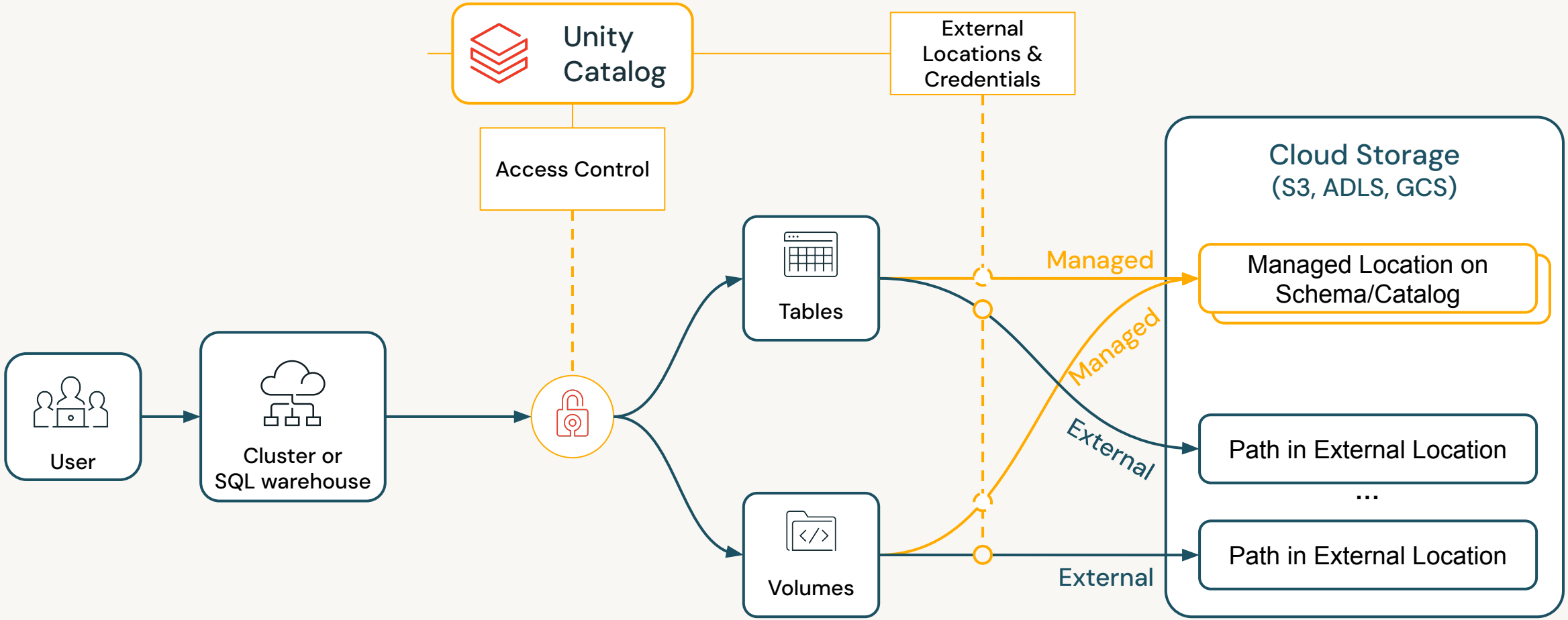
- Credentials
  - Cloud provider credential to connect to storage
- External Locations
  - Storage location used for external tables, external volumes, or arbitrary files, or default managed location for a catalog or schema
- Managed / External Tables
  - Tabular data stored in managed or external locations
- Managed / External Volumes
  - Arbitrary file container inside a managed or external location

Working with databases

- Connections
  - Credential and connection information to connect to an external database
- Foreign Catalogs
  - A catalog that represents an external database in UC and can be queried alongside managed data sources and file sources

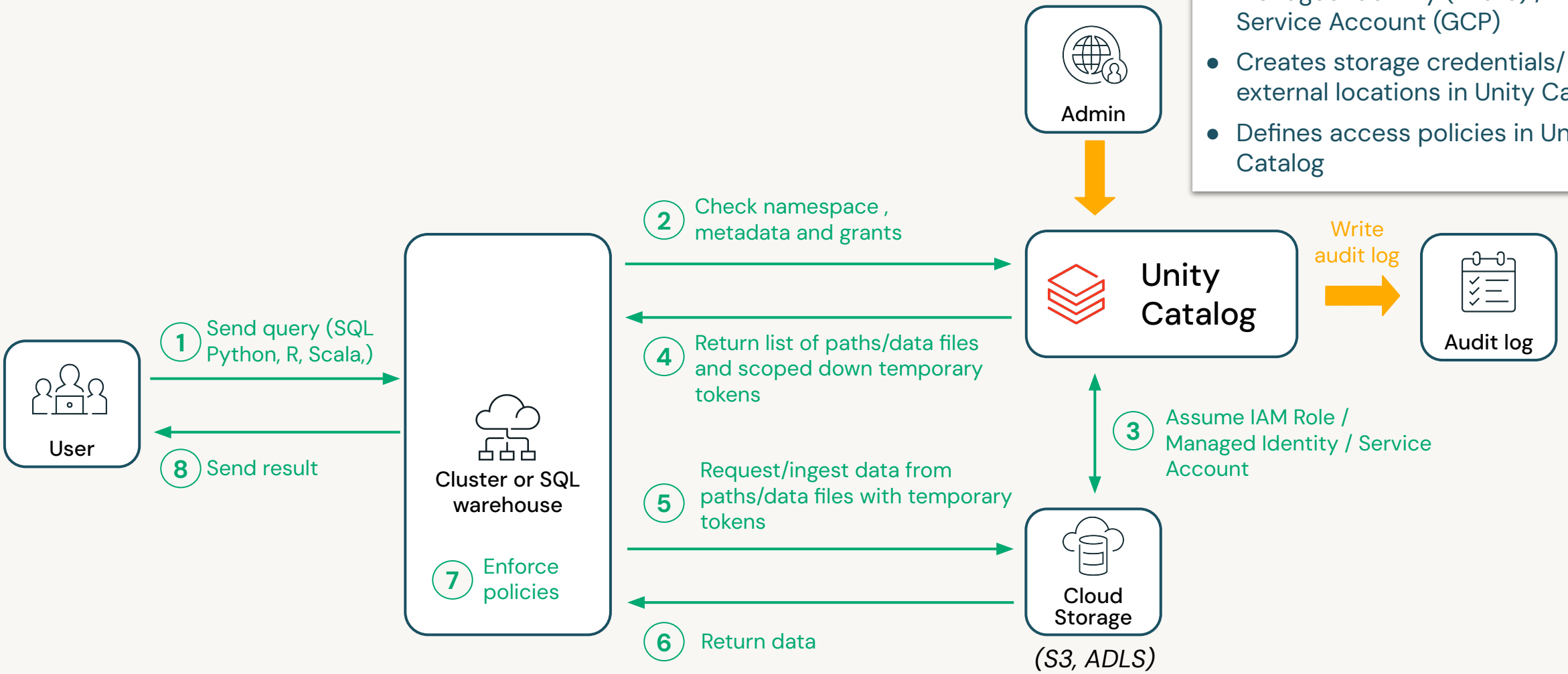
Defining file based data sources in Unity

Simplify data access management across clouds

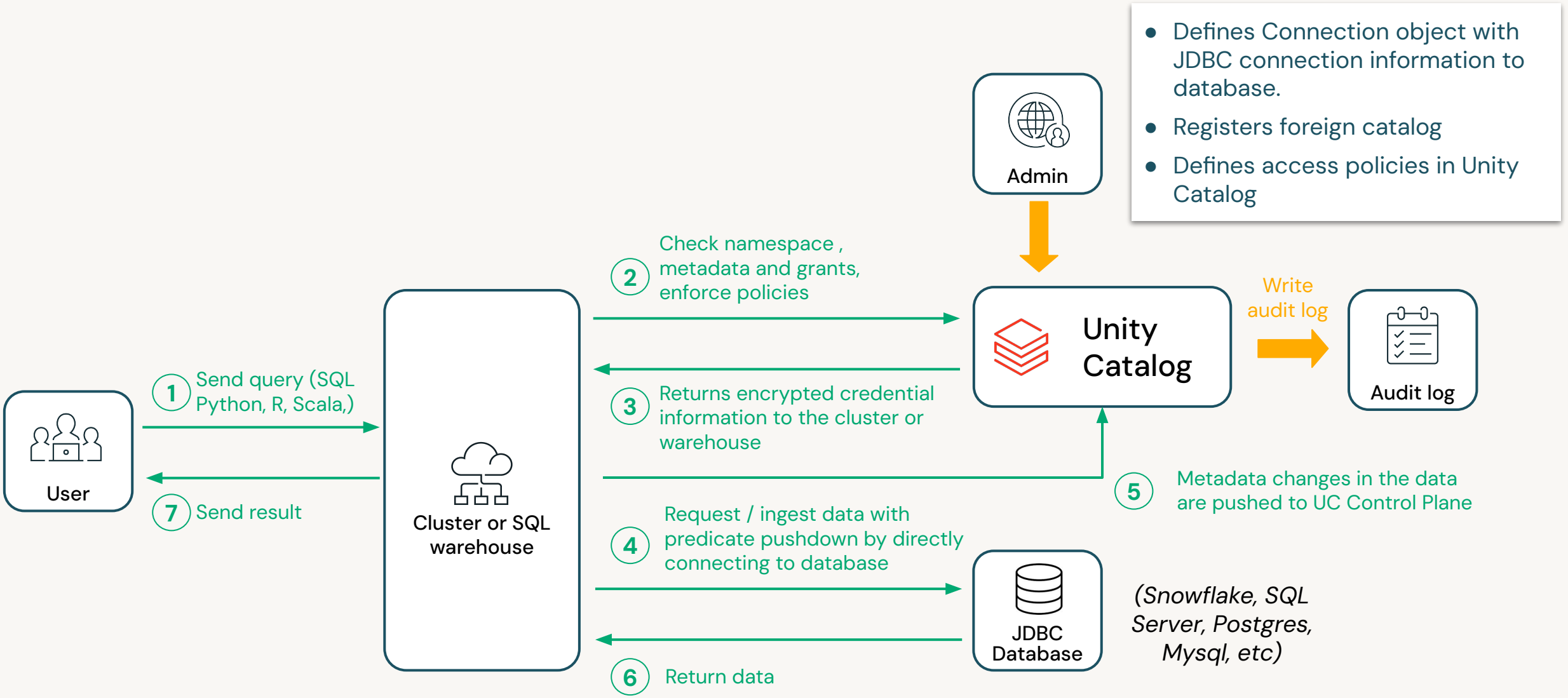


Querying file based data sources with Unity

- Creates an IAM role (AWS)/ Managed Identity (Azure) / Service Account (GCP)
- Creates storage credentials/ external locations in Unity Catalog
- Defines access policies in Unity Catalog



Querying database sources with Unity



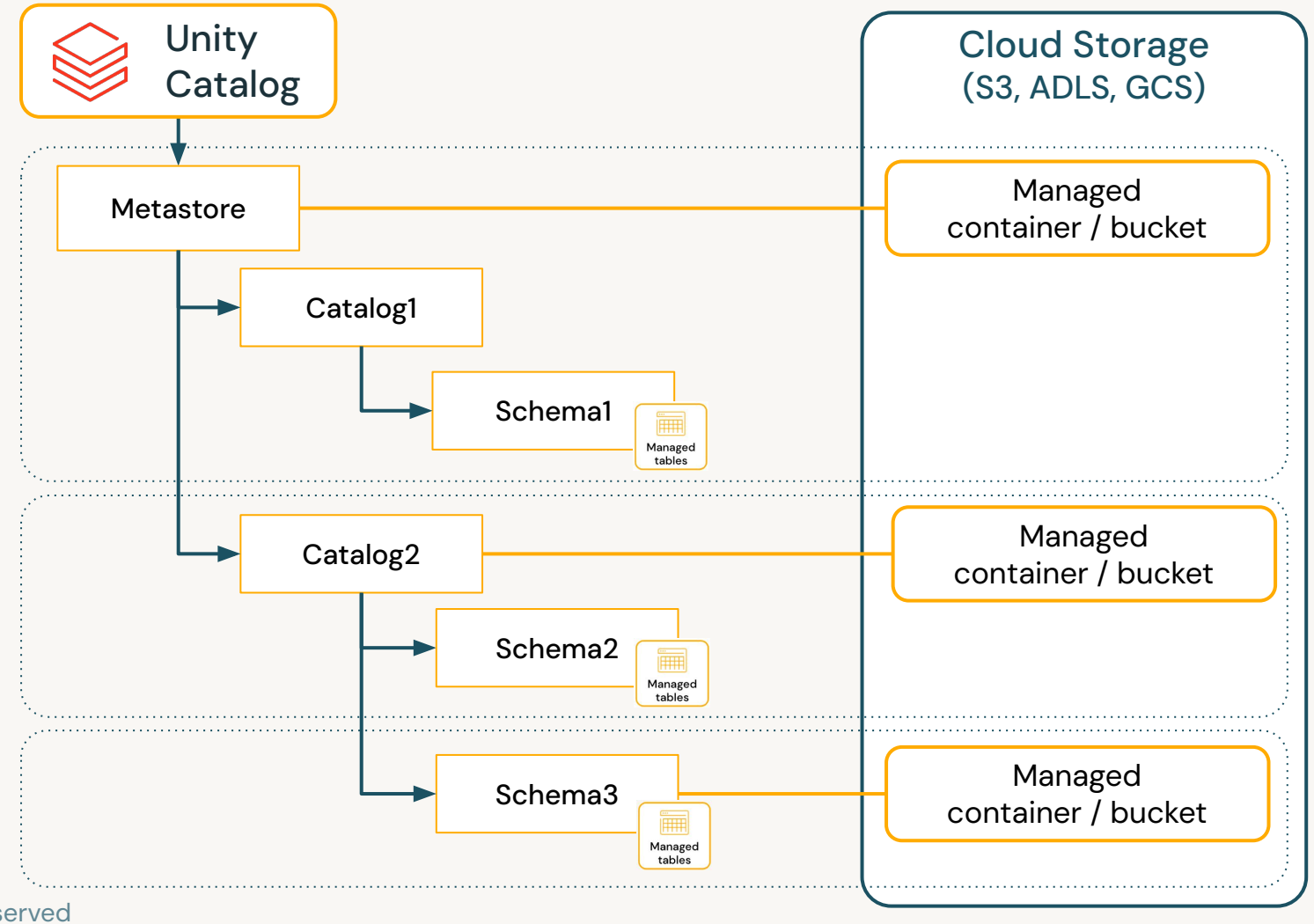
Isolation between file based data sources

Use managed data sources for data isolation or cost allocation

1) Store at the **metastore**

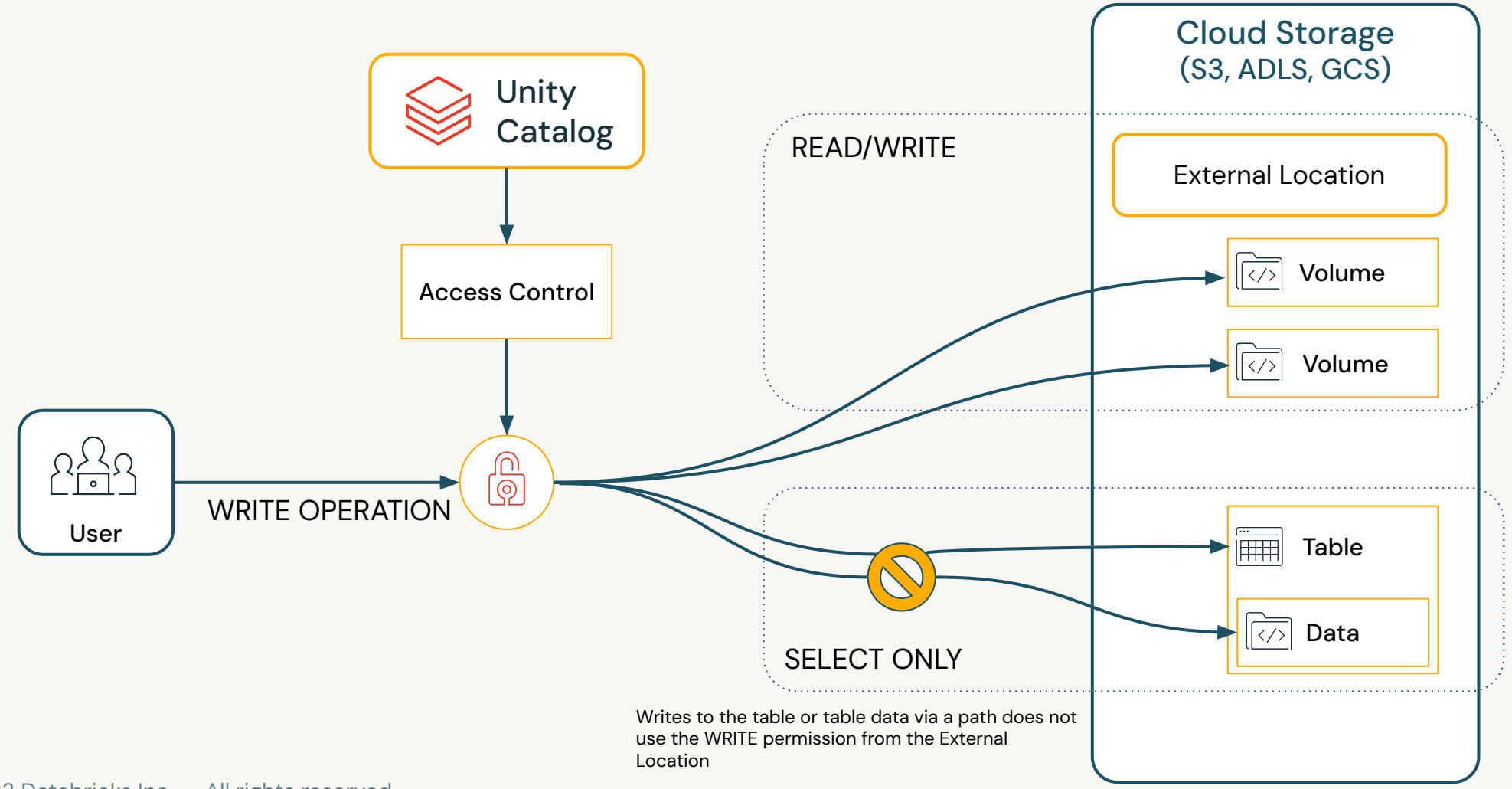
2) Store at the **catalog**

3) Store at the **schema**



Multiple ACL trees for flexible governance

Govern external tables and file based data source access separately



Discover your data with search and lineage

Why is data lineage important?

Compliance

- **Regulatory** requirements to verify data lineage
- Track the **spread of sensitive data** across datasets

Discovery

- Understand **context** and **trustworthiness** of data before using it in analytics
- **Prevent duplicative** work and data

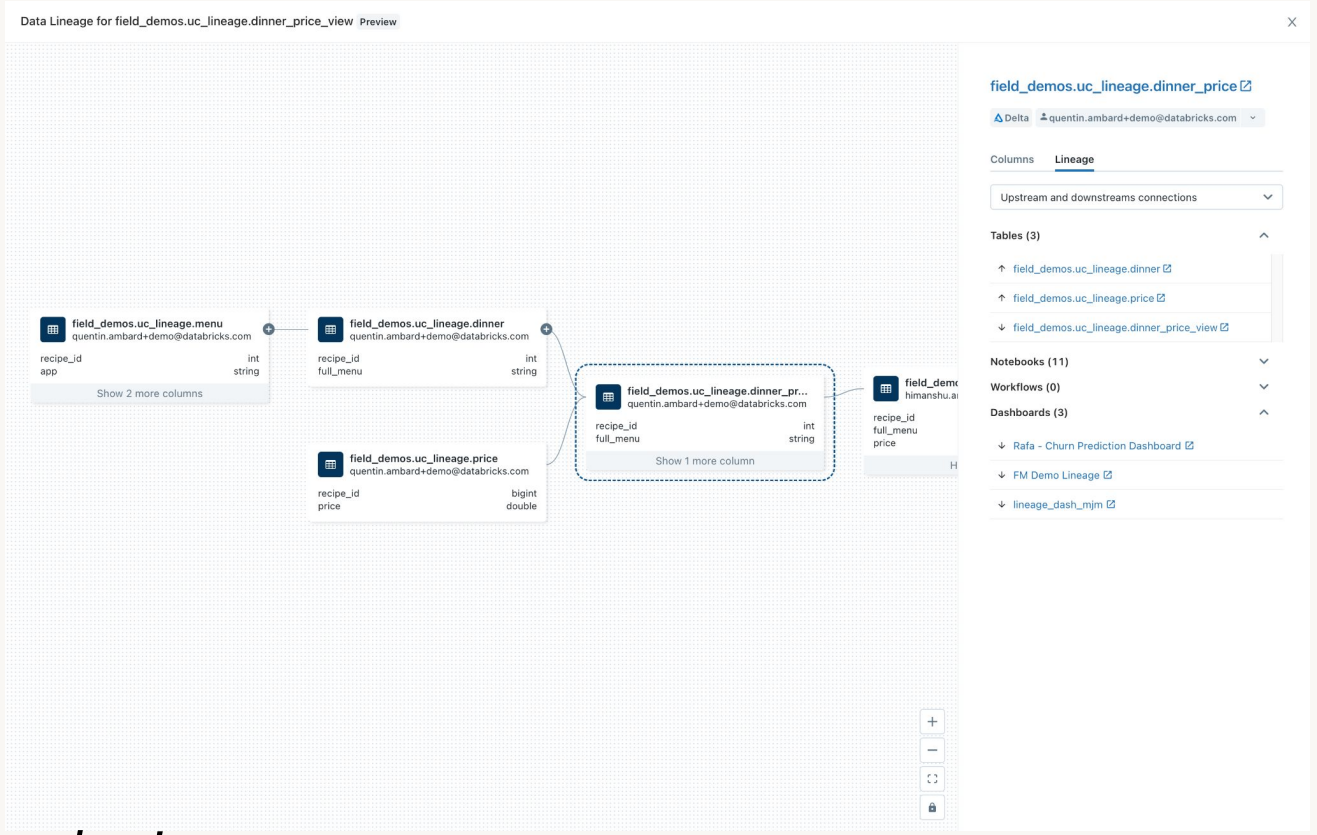
Observability

- Track down **issues / discrepancies** in reports by tracing back the data
- Analyze **impact of proposed changes** to downstream reports e.g. column deprecation

Automated lineage for all workloads

End-to-end visibility into how data flows and consumed in your organization

- Auto-capture runtime data lineage on a Databricks cluster or SQL warehouse
- Leverage common permission model from Unity Catalog
- Lineage across tables, columns, dashboards, workflows, notebooks, files, external sources, and models



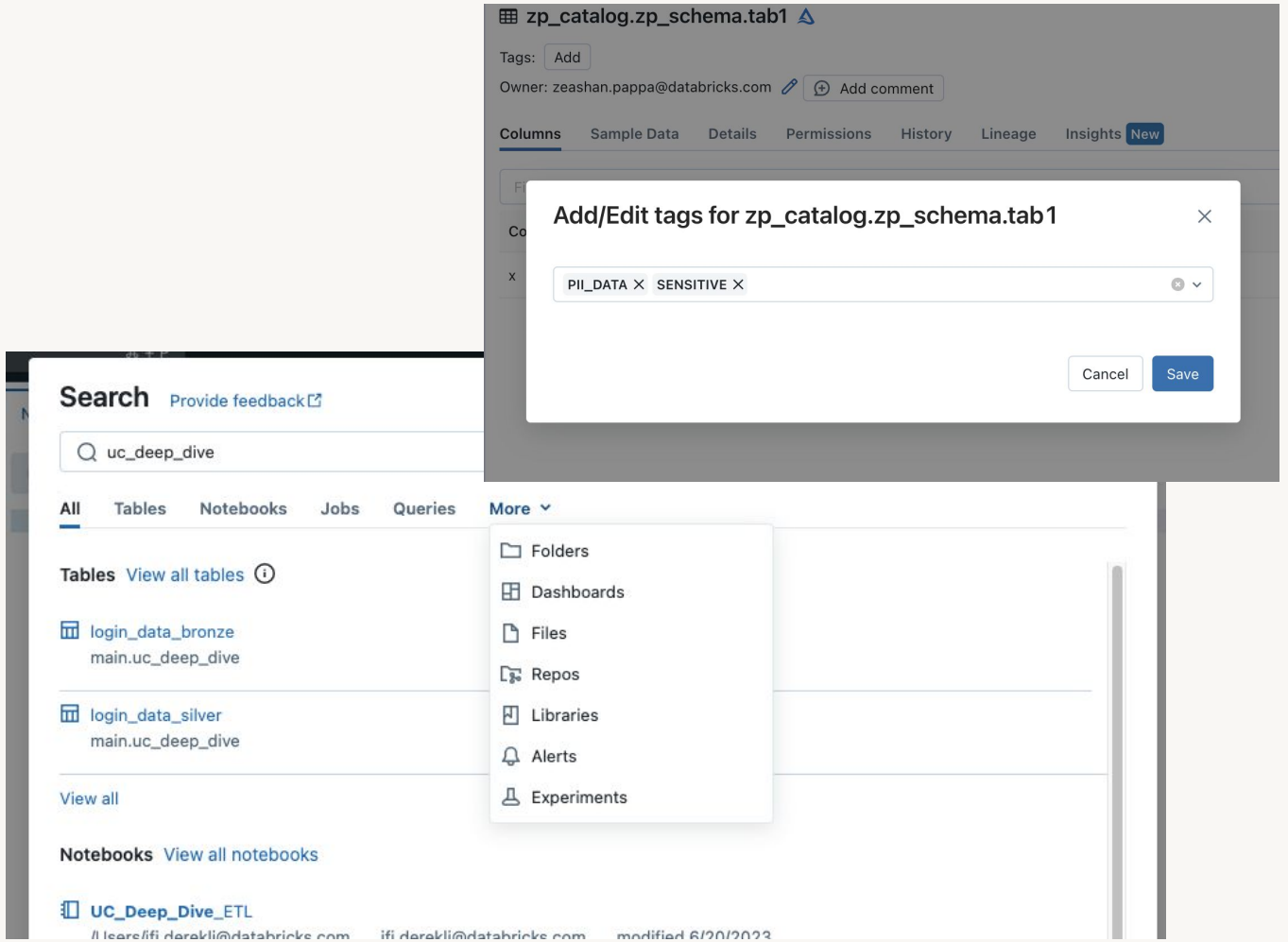
- *Recommendation: Upgrade to Unity Catalog!*

Built-in search and discovery

Accelerate time to value with low latency data discovery

- Unified UI to search for data assets stored in Unity Catalog
- Leverage common permission model from Unity Catalog
- Tag Column, Table, Schema, Catalog objects in UC
- Search for objects on tags

- *Recommendation: Use comments and Tag your Data Assets on Ingest*



Audit your data

System Tables: Object Metadata

Answer questions about the state of objects in the catalog

What tables are in the sales catalog?

```
SELECT table_name
FROM system.information_schema.tables
WHERE table_catalog="sales"
AND table_schema!="information_schema";
```

Who has access to this table?

```
SELECT grantee, table_name, privilege_type
FROM system.information_schema.table_privileges
WHERE table_name = "login_data_silver";
```

`

Who last updated the gold tables and when?

```
SELECT table_name, last_altered_by, last_altered
FROM system.information_schema.tables
WHERE table_schema = "churn_gold"
ORDER BY 1, 3 DESC;
```

Who owns this gold table?

```
SELECT table_owner
FROM system.information_schema.tables
WHERE table_catalog = "retail_prod" AND table_schema =
"churn_gold" AND table_name = "churn_features";
```


System Tables: Audit Logs

Near-realtime, see who accessed what, and when

In Preview

Who accesses this table the most?

```
SELECT user_identity.email, count(*)
FROM system.operational_data.audit_logs
WHERE request_params.table_full_name = "main.uc_deep_dive.login_data_silver"
AND service_name = "unityCatalog"
AND action_name = "generateTemporaryTableCredential"
GROUP BY 1 ORDER BY 2 DESC LIMIT 1;
```

Who deleted this table?

```
SELECT user_identity.email
FROM system.operational_data.audit_logs
WHERE request_params.full_name_arg =
"main.uc_deep_dive.login_data_silver"
AND service_name = "unityCatalog"
AND action_name = "deleteTable";
```

What has this user accessed in the last 24 hours?

```
SELECT request_params.table_full_name
FROM system.operational_data.audit_logs
WHERE user_identity.email = "ifi.derekli@databricks.com"
AND service_name = "unityCatalog"
AND action_name = "generateTemporaryTableCredential"
AND datediff(now(), created_at) < 1;
```

What tables does this user access most frequently?

```
SELECT request_params.table_full_name, count(*)
FROM system.operational_data.audit_logs
WHERE user_identity.email = "ifi.derekli@databricks.com"
AND service_name = "unityCatalog"
AND action_name = "generateTemporaryTableCredential"
GROUP BY 1 ORDER BY 2 DESC LIMIT 1;
```


In Preview

System Tables: Billing Logs

Understand cost allocation across your data estate

What is the daily trend in DBU consumption?

```
SELECT date(created_on) as `Date`, sum(dbus) as `DBUs Consumed`
   FROM system.operational_data.billing_logs
GROUP BY date(created_on)
ORDER BY date(created_on) ASC;
```

How many DBUs of each SKU have been used so far this month?

```
SELECT sku as `SKU`, sum(dbus) as `DBUs`
   FROM system.operational_data.billing_logs
WHERE
   month(created_on) = month(CURRENT_DATE)
GROUP BY sku
ORDER BY `DBUs` DESC;
```

Which 10 users consumed the most DBUs?

```
SELECT tags.creator as `User`, sum(dbus) as `DBUs`
   FROM system.operational_data.billing_logs
GROUP BY tags.creator
ORDER BY `DBUs` DESC
LIMIT 10;
```

Which Jobs consumed the most DBUs?

```
SELECT tags.JobId as `Job ID`, sum(dbus) as `DBUs`
   FROM system.operational_data.billing_logs
GROUP BY `Job ID`;
```

System Tables: Lineage Data

Query upstream and downstream sources in one place

In Preview

What tables are sourced from this table?

```
SELECT DISTINCT target_table_full_name
FROM system.access.table_lineage
WHERE source_table_name = "login_data_bronze";
```

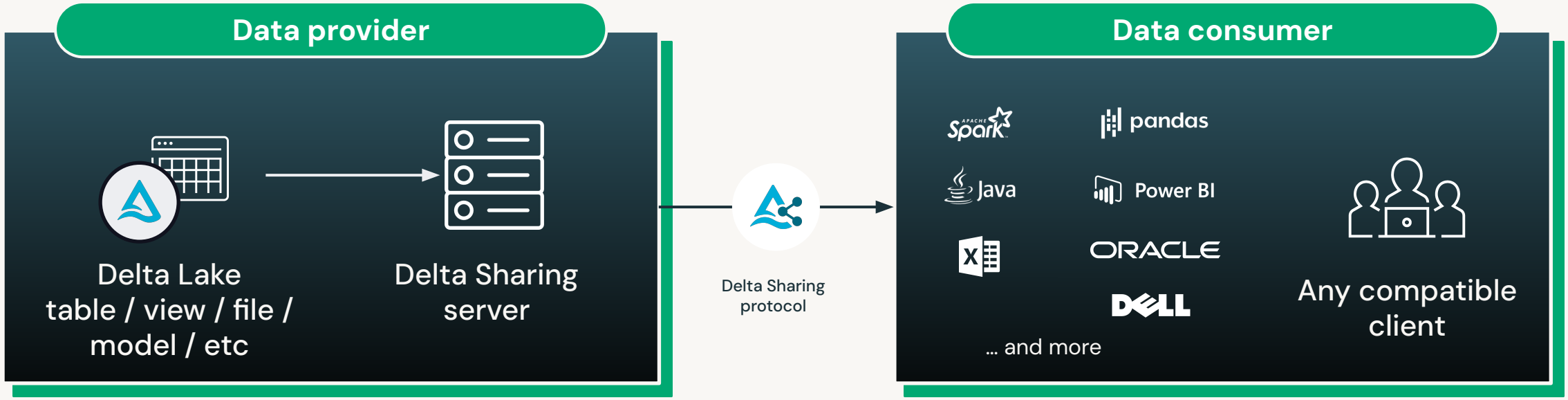
What user queries read from this table?

```
SELECT DISTINCT entity_type, entity_id,
source_table_full_name
FROM system.access.table_lineage
WHERE source_table_name = "login_data_silver";
```

Open data sharing powered by Unity Catalog

Delta Sharing

An open standard for secure sharing of tables, views, files, models, and more



Share cross-platform w/ open protocol

Share data with no replication

Databricks Marketplace

Open marketplace for data, analytics, & AI.

Datasets

Notebooks

Dashboards

ML models

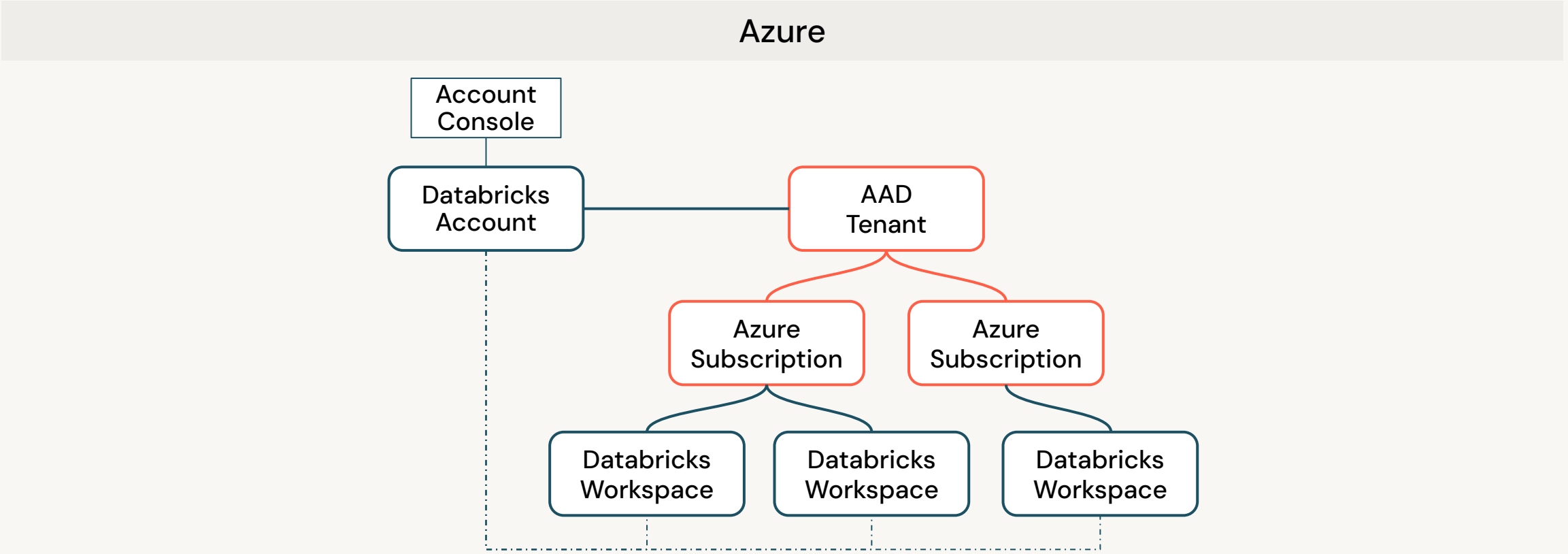
Solutions accelerators

Data applications

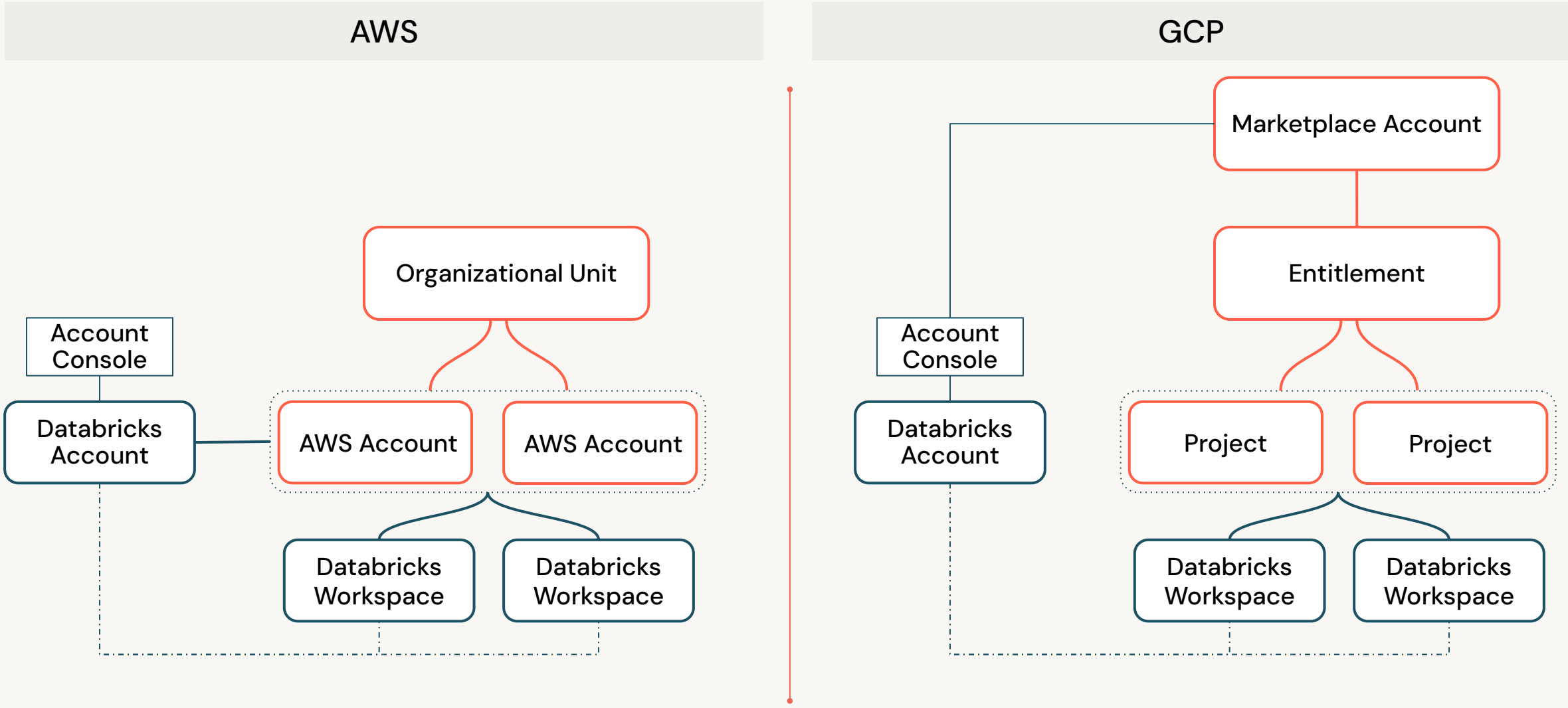
Powered by Delta Sharing

Unity Catalog and cloud providers

Databricks Accounts and Cloud Providers



Databricks Accounts and Cloud Providers



Unity Catalog and Cloud Constructs

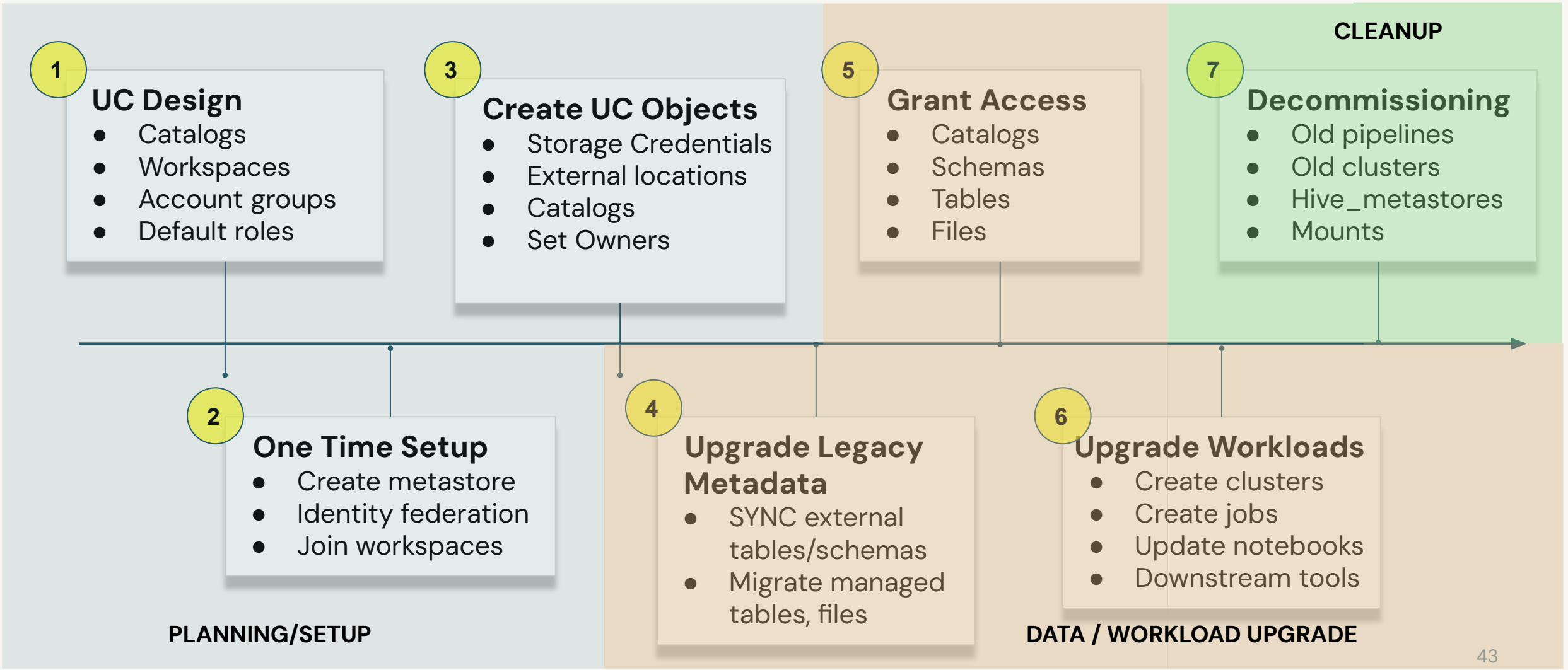
	AWS	Azure	GCP
Databricks Account	Accounts	Tenant	Marketplace Account
Metastore	Region	Region	Region
Catalog	*Account**	*Subscription**	*Project**
Storage Location	S3 Bucket	ADLS Account	GCS Bucket
Credential	IAM Role	Managed Identity	Service Account

* *Minimum one, more are optional*

Upgrade to Unity Catalog

High Level Roadmap to Unity Catalog

Steps to consider for a full upgrade



Best Practices for working with Cloud Constructs

Organizational patterns dictate usage

Centralized

- Data and compute is typically stored in **central subscriptions / accounts / projects** in the the cloud, by SDLC scope (e.g. DEV vs PROD).
- One team controls cloud **infrastructure** → One team controls creation of **credentials and external locations.**
- **One** team **produces data** → **One** team manages **production pipelines**.
- **One** team owns **governance and access** → **One** team will administer and **own Metastore and Catalogs** and manage permissions.
- ***Central team* carries administrative responsibilities both in the cloud and in Databricks.**

Distributed

- **Each team has their own subscription(s) / account(s) / project(s)** in the cloud with their own storage and compute for data isolation and cost allocation → each team owns their own **catalog(s) and workspace(s).**
- **Many or one** team controls cloud **infrastructure** → **Many or one** team controls creation of **credentials and external locations.**
- **Many** teams **produce data** → **each** team manages their **own workspaces / pipelines.**
- ***Each team* carries *shared* administrative responsibilities both in the cloud and in Databricks.**

Key Roles in Unity Catalog

Assign roles to groups

Account Admin Group

- Can create workspaces *(*in Azure any contributor can)*
- Can create & configure metastores
- Can create users, groups, & service principals
- Can create credentials
- Can grant users access to workspaces
- *Recommended: Platform Ops or Central Gov Team*

Metastore Admin Group

- Can create CATALOG, EXTERNAL LOCATION
- Can create SHARE, RECIPIENT
- Can change OWNER of any securable object
- *Recommended: Platform Ops or Central Gov Team*
- *Recommended: Delegate to Data Owners*

Workspace Admin Group

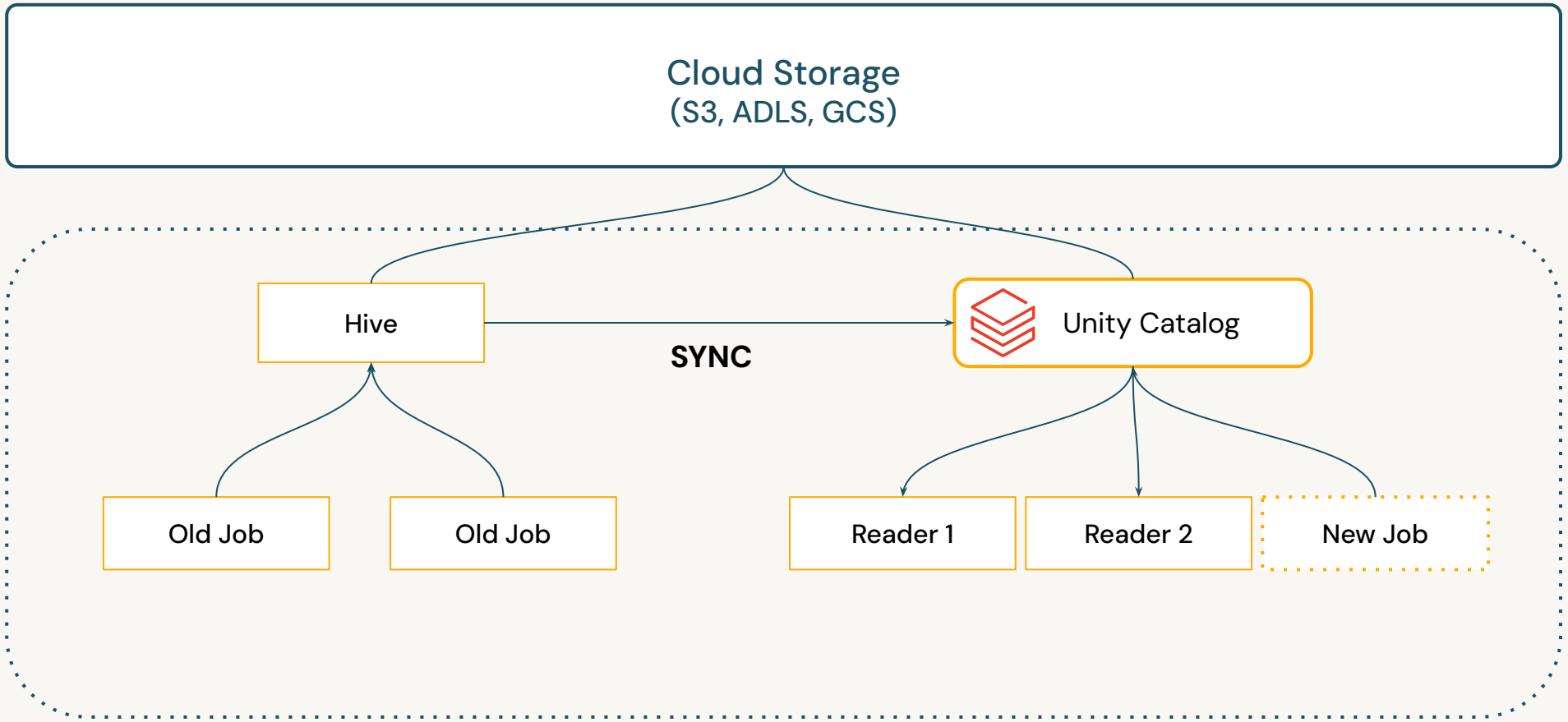
- Can add users and groups to workspace
- Can create clusters & cluster policies
- Can change OWNER of clusters, workflows, notebooks, queries, dashboards
- *Recommended: IT/Platform/DevOps team*
- *Recommended: Regular Service Principal Audits*

Data Owner Group

- Can change ownership of securable object (CATALOG, SCHEMA, TABLE, VIEW, etc.)
- Can grant any privilege to any principal
- *Recommended: BU Gov Team or Central Gov Team*

Keep your jobs

Bring your readers



Upgrading Hive tables to Unity

Managed & External tables - use SYNC command

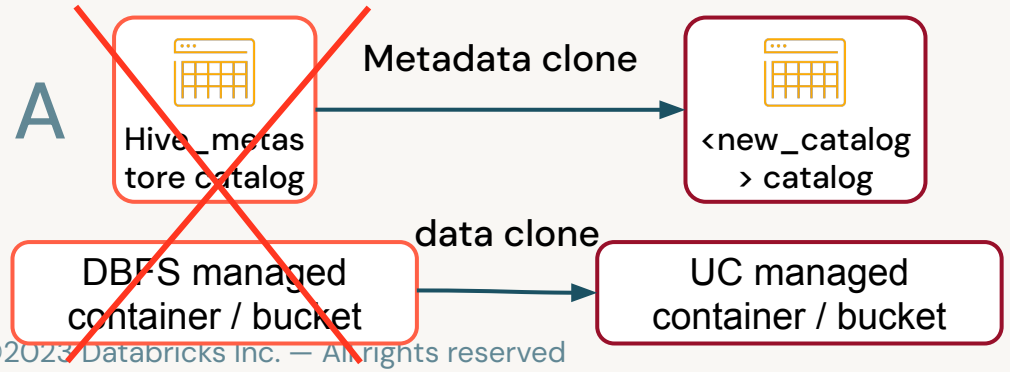
- Run multiple times to pull changes from the hive/glue database into Unity over time
  - Use a job for long term synchronization
- Use the DRY RUN option to test the sync without making any changes to the target table.
- Works on Hive Managed Tables where schema locations are defined.

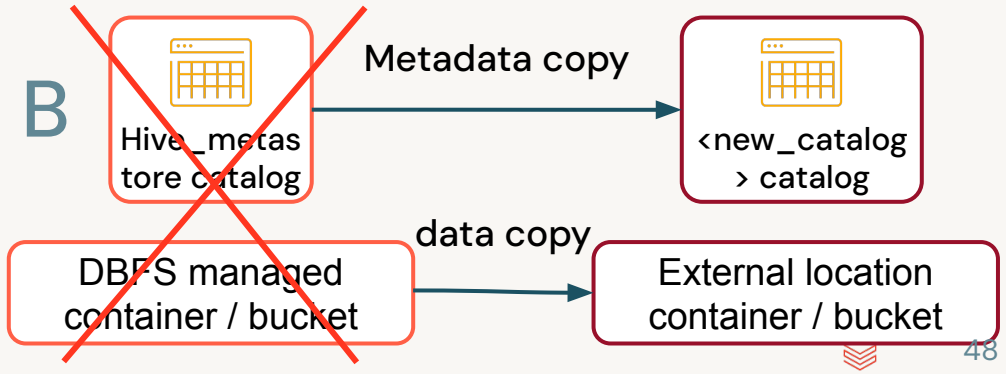
```
SYNC SCHEMA hive_metastore.my_db TO SCHEMA main.my_db_uc DRY RUN

SYNC TABLE hive_metastore.my_db.my_tbl TO TABLE main.my_db_uc.my_tbl
```

Moving Managed Hive tables to Unity

Optional or if in DBFS root - CTAS/CLONE

```
1  // A. Managed Delta -> Managed Delta
2  CREATE TABLE <new_catalog>.<new_schema>.<new_table> CLONE
3  hive_metastore.<old_schema>.<old_table>;
4  // B. Managed non-Delta -> External non-Delta
5  CREATE TABLE <new_catalog>.<new_schema>.<new_table> LOCATION <..> AS SELECT * FROM
6  hive_metastore.<old_schema>.<old_table>;
7  // A+B. Once fully upgraded and tested, drop hive table
8  DROP TABLE hive_metastore.<old_schema>.<old_table>;
```





Demo

Thank you!