databricks

Governance with Unity Catalog

for data and AI

Gopala Raju

Governance - Product Specialist - APJ

- 2+ years with Databricks, 19+ years of experience
- 10+ years of experience in Data and Analytics
- Governance lead for APJ

Agenda

- Introduction
- Unity Catalog overview
- Key capabilities & Lakehouse Federation
- Sharing and Collaboration
- Unity Catalog on different clouds
- 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"

—

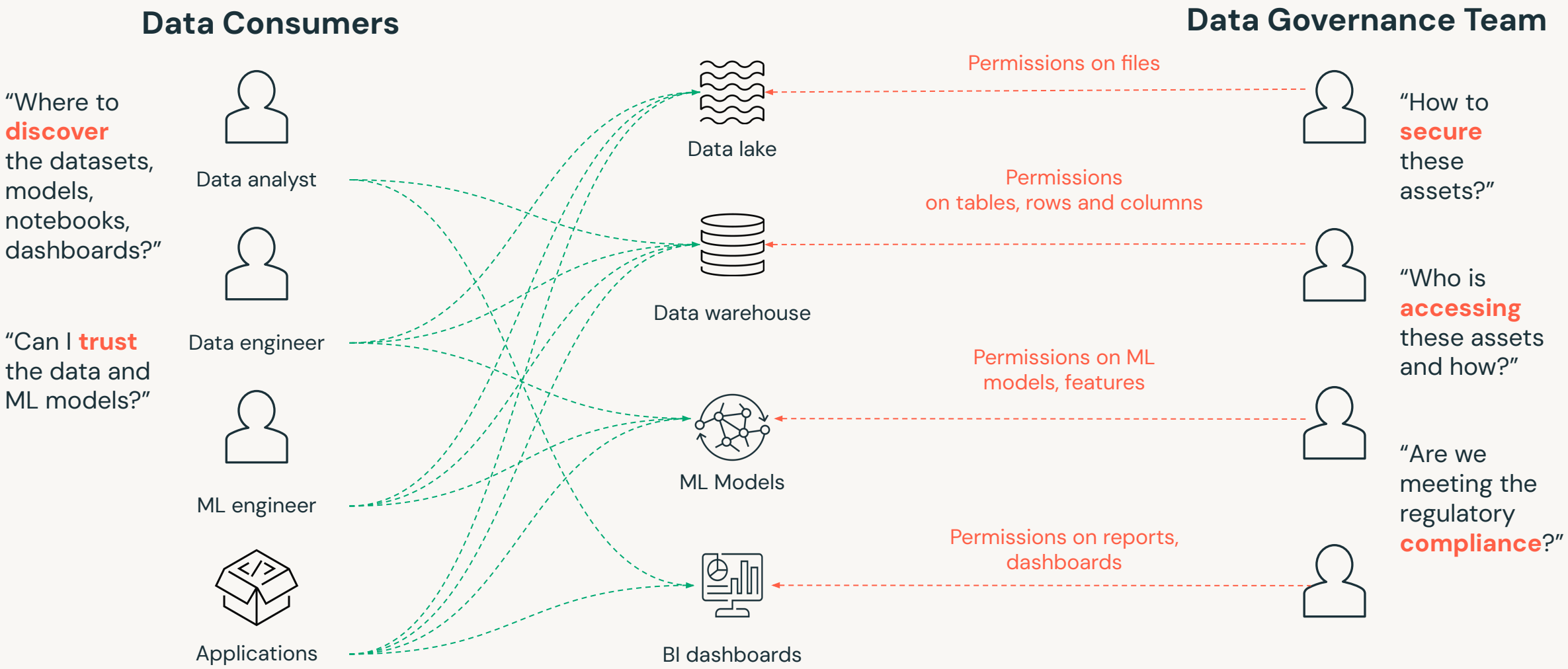
The State of AI in 2022, McKinsey & Co

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

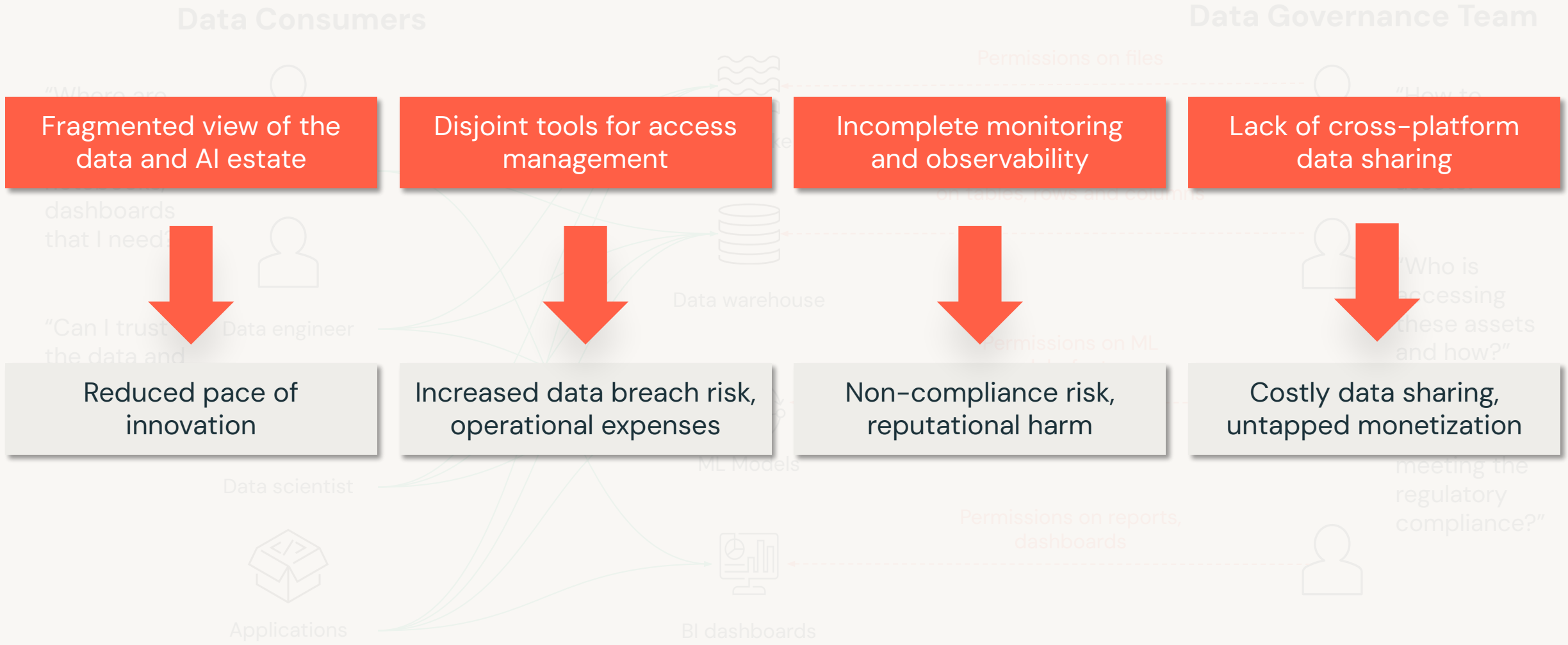
—

Forrester, 2023 AI Predictions report

Today, data and AI governance is complex



Today, data and AI governance is complex



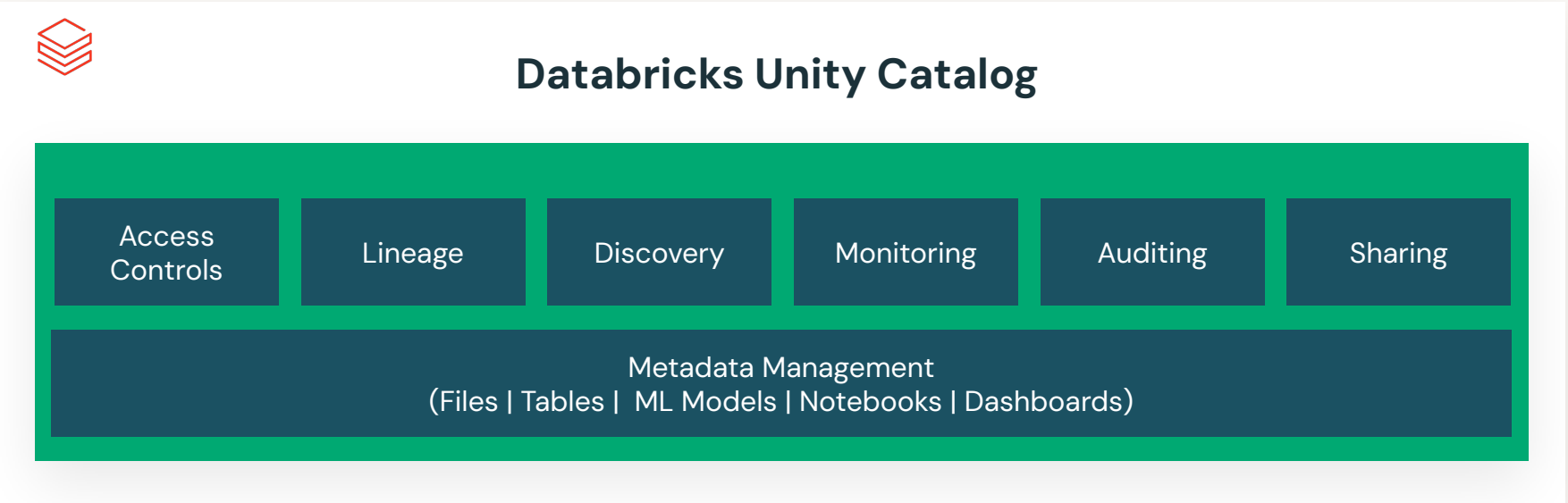
Databricks Unity Catalog
Unified governance for data and AI

Unified visibility into data and AI

Single permission model for data and AI

AI-powered monitoring and observability

Open data sharing



Lakehouse Federation

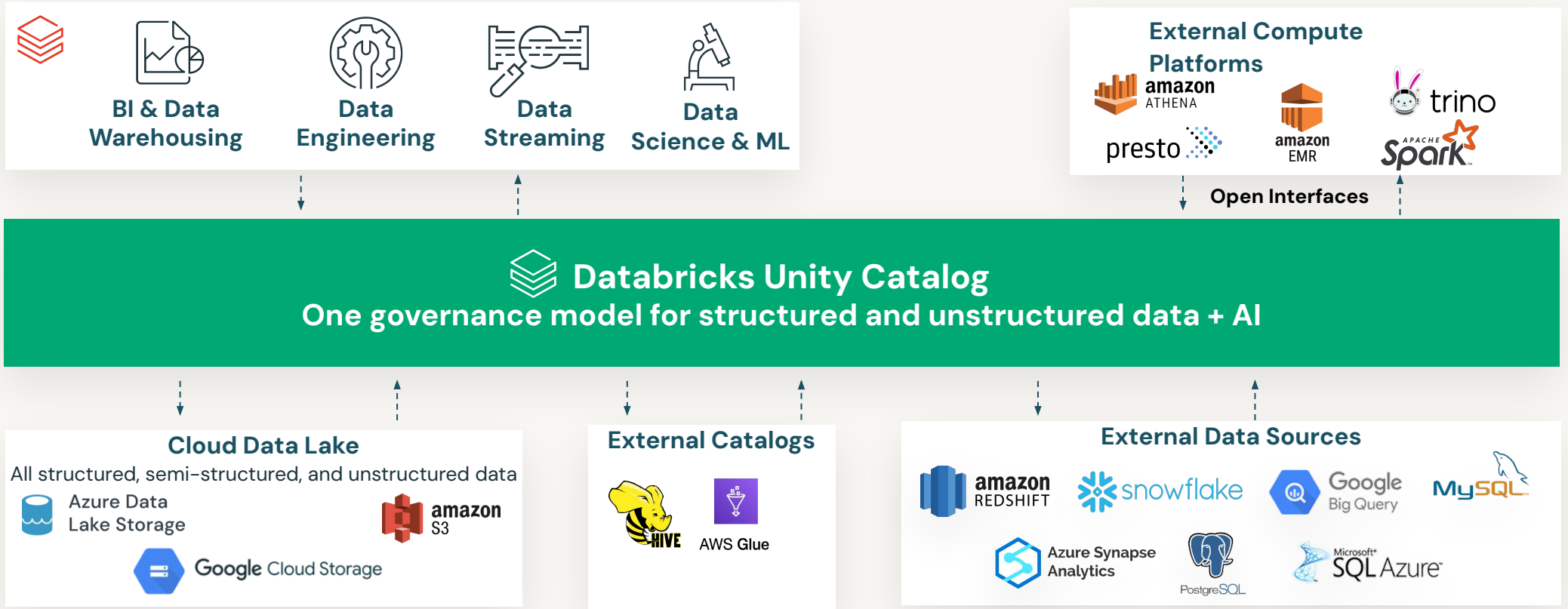
Discover, query, and govern all your data - no matter where it lives

Build a unified view of your data estate

Query and combine all data efficiently with a single engine

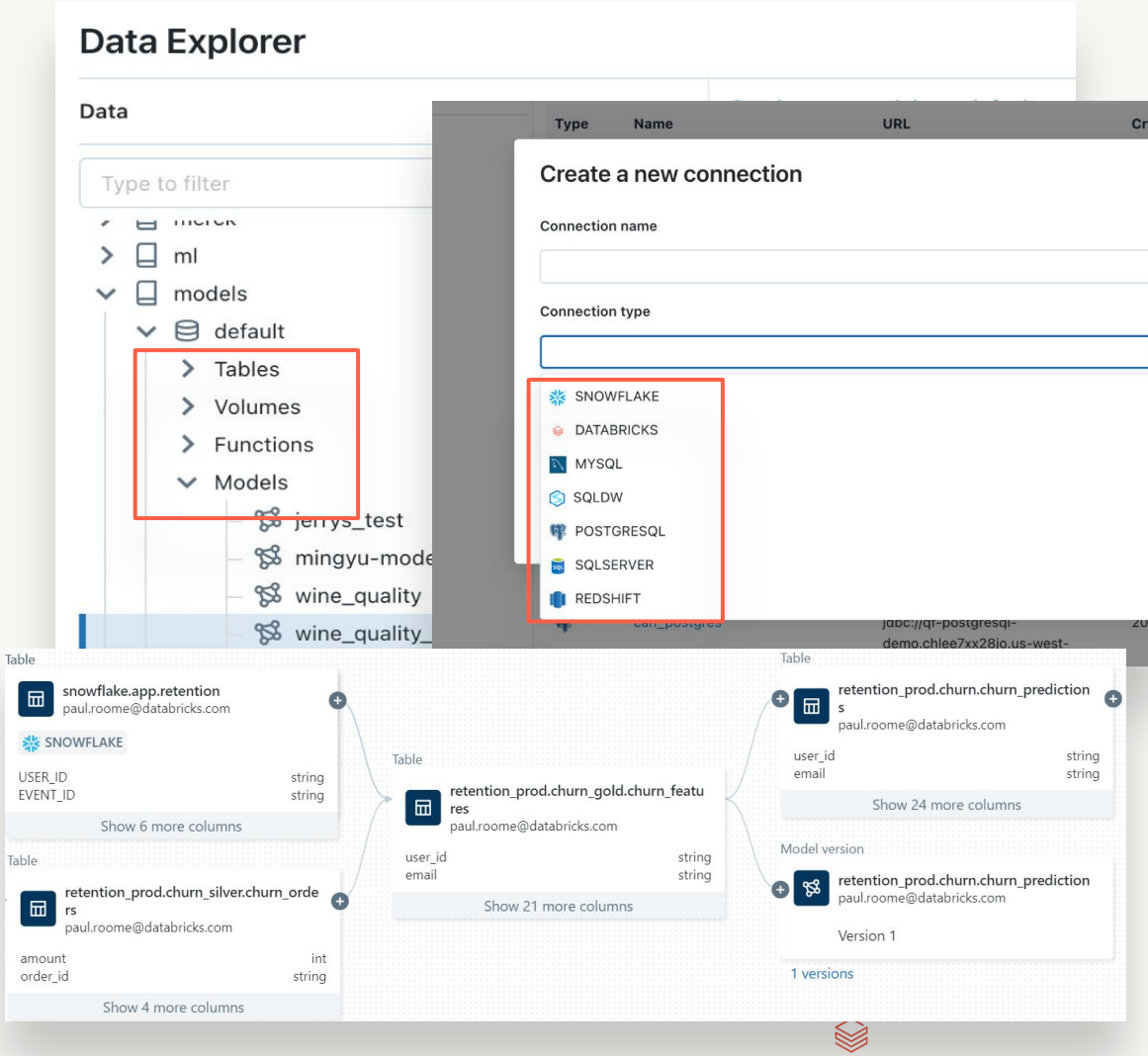
Safeguard data across data sources

Databricks Lakehouse unifies data and AI governance



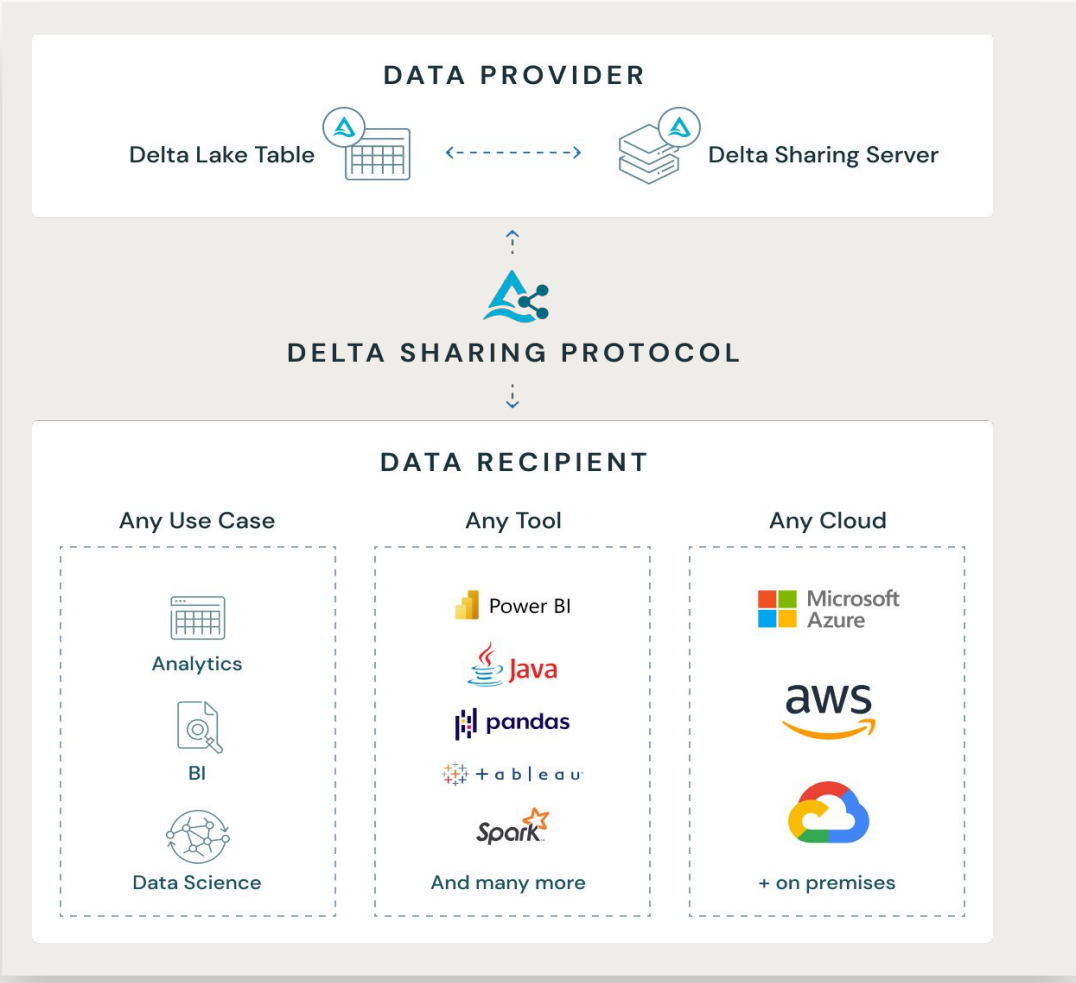
Unified visibility into data and AI

- **Discover and classify** structured and unstructured data, files, notebooks, ML models, and dashboards at one place
- Consolidate and query data from **other databases and data warehouses** using a **single point of access,** without moving or copying the data
- Build better **understanding of your data estate** with automated lineage, tags and auto-generated data insights
- Boost productivity by searching, understanding and gaining insights from your data and AI assets, using **natural language**

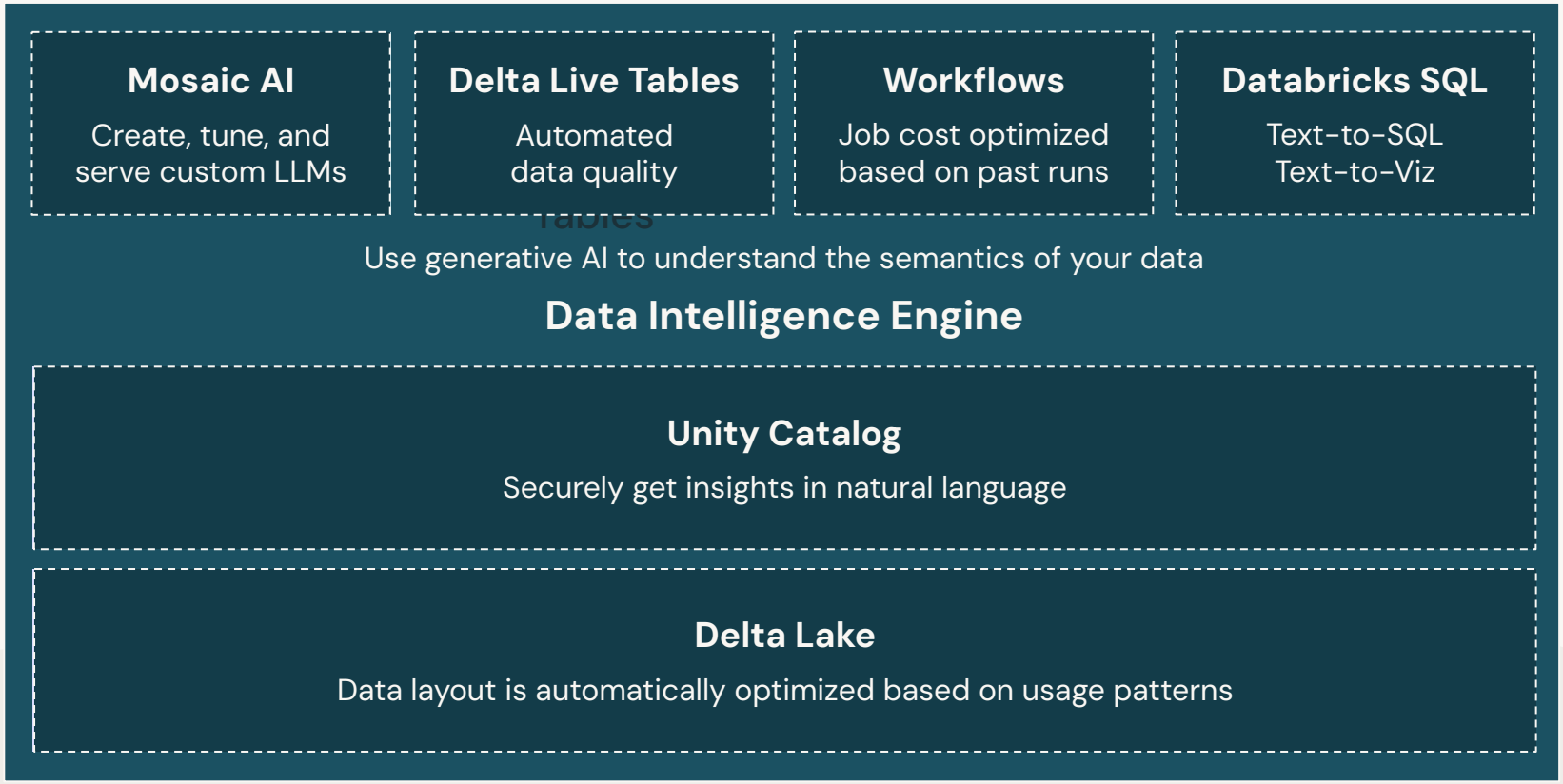


Open data sharing

- **Avoid vendor lock-in** with open source Delta Sharing for seamless data sharing across clouds, regions, and platforms, without replication
- Share **more than just data** - Notebooks, ML models, dashboards, applications
- Explore and monetize data products through an **open marketplace**
- Collaborate securely on sensitive data with **scalable data clean rooms**



Unity Catalog Powers Databricks Data Intelligence Platform



Open Data Lake

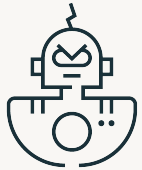
All Raw Data
(Logs, Texts, Audio, Video, Images)

Unlock full databricks experience

Unity Catalog is Central to Data Intelligence Platform

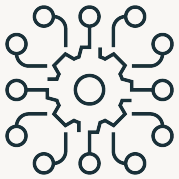
Lakehouse Monitoring

End-to-end monitoring of Data and ML Models for quality and drift



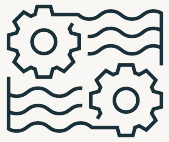
Databricks Assistant

A context-aware AI assistant, that integrates throughout the platform to improve productivity via conversational interface



Lakehouse IQ

A knowledge engine that learns unique nuances of your business and data to power natural language access



AI Accelerated Performance

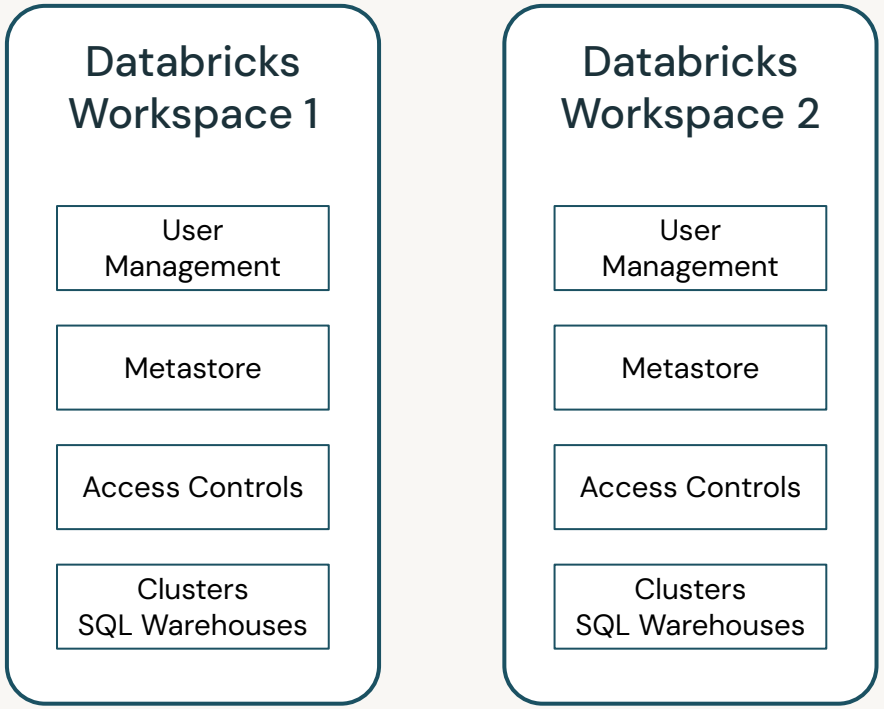
Automated organization of data, performance improvements, predictive IO and serverless

Unity Catalog Overview

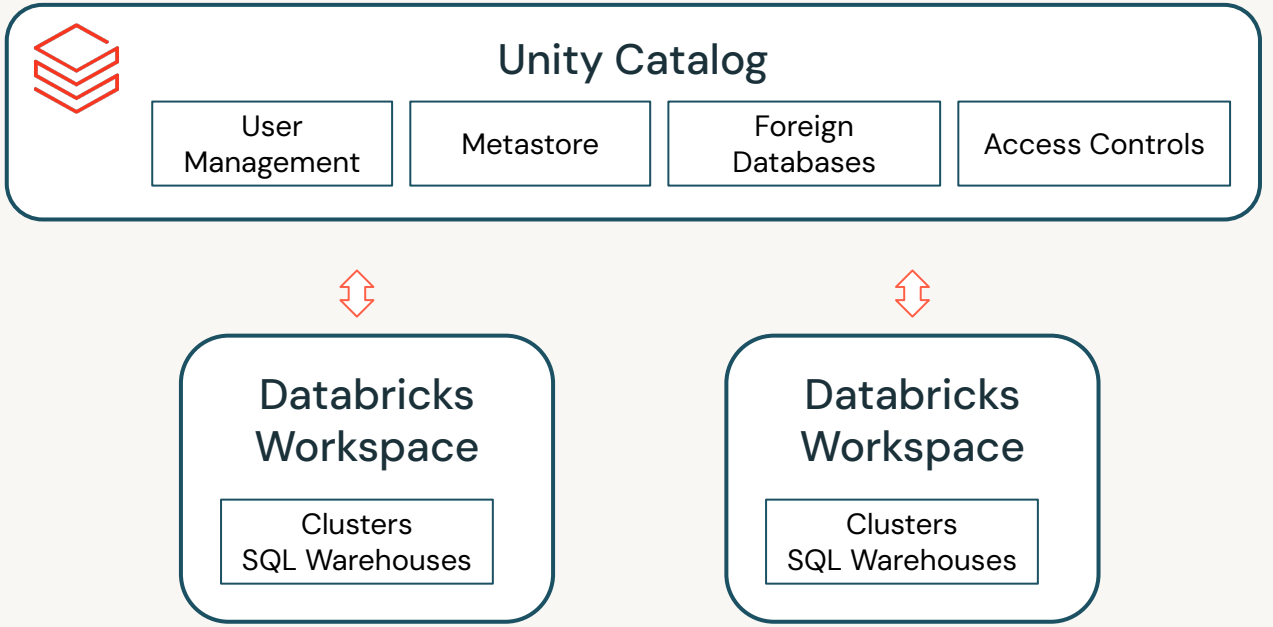
Centralized metadata and controls

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

Without Unity Catalog



With Unity Catalog



Fundamental Concepts

Working with file based data sources

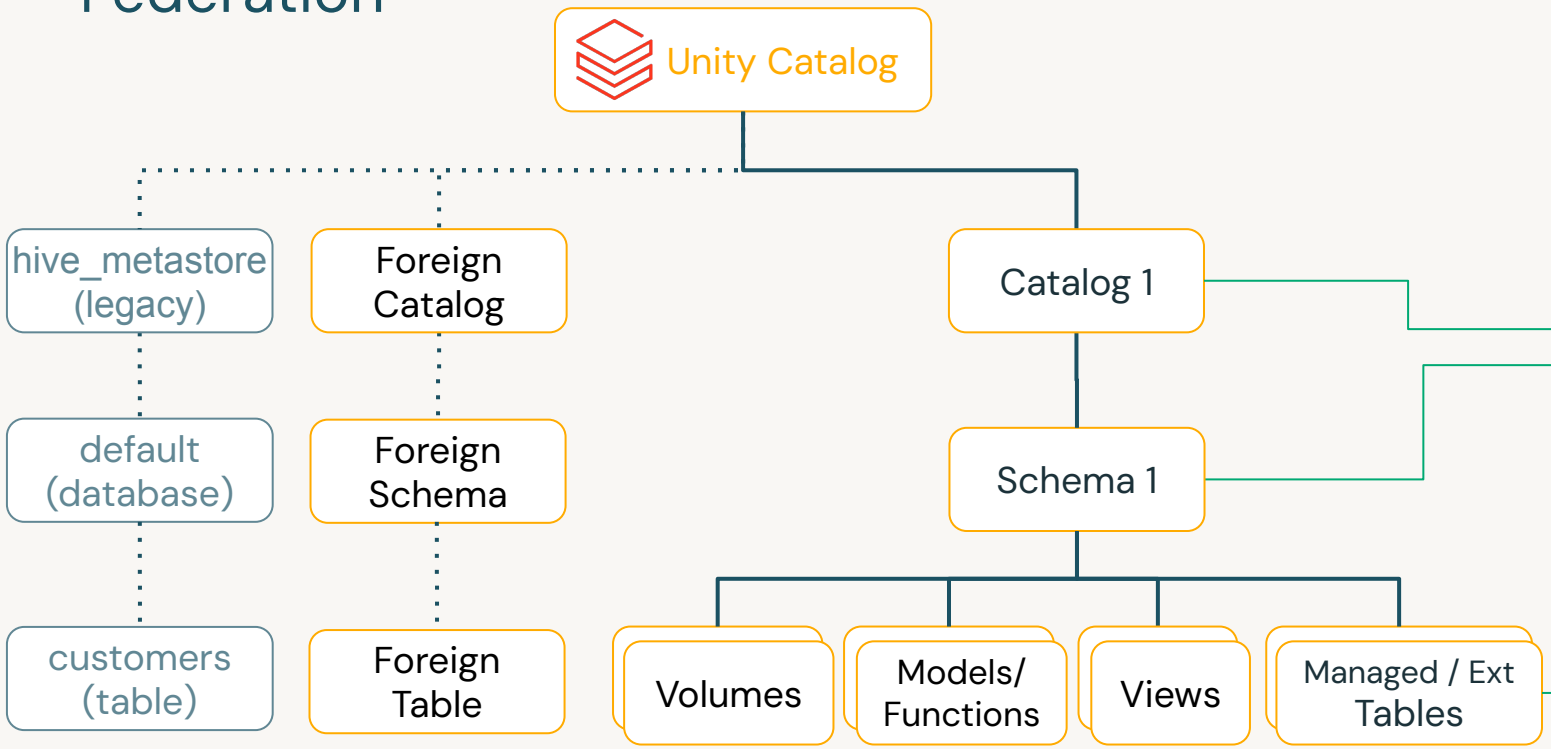
- Credentials
  - Cloud provider credential to connect to storage
- External Locations
  - Storage location used for external tables, external volumes, or arbitrary files, or default managed location for a catalog or schema
- Managed / External Tables
  - Tabular data stored in managed or external locations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

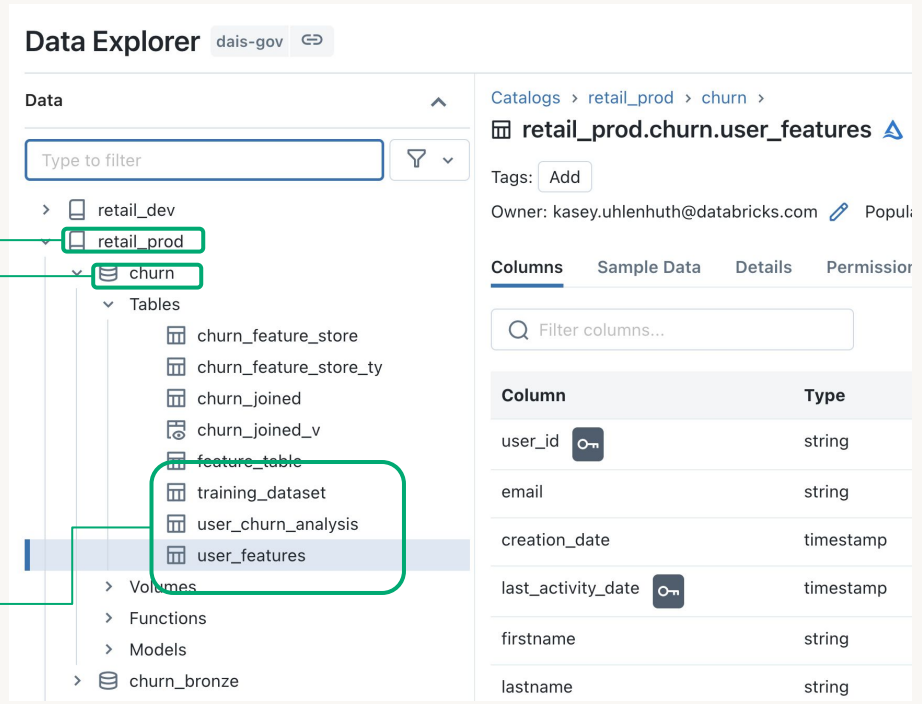
Working with databases

- 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

Governed namespace across file and database sources

Access legacy metastore and foreign databases powered by Lakehouse Federation






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

SELECT * FROM hive_metastore.default.customers;

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

Centralized Access Controls

Centrally grant and manage access permissions across workloads and foreign databases

Using ANSI SQL DCL

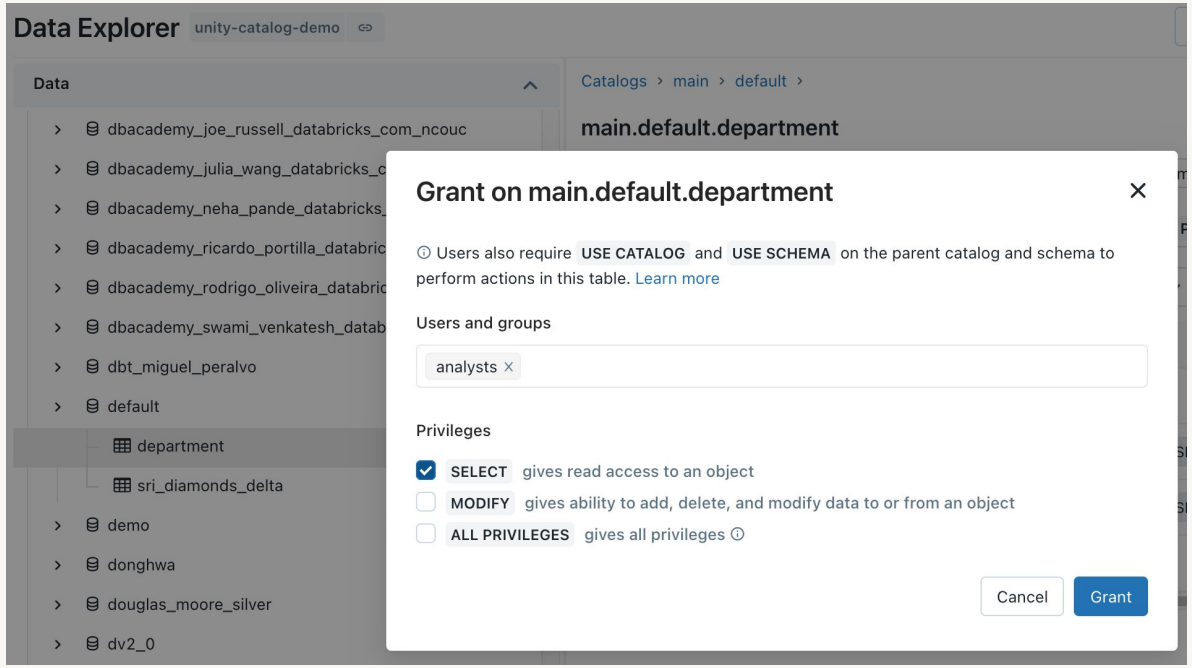
```
GRANT <privilege> ON <securable_type>
<securable_name> TO `<principal>`

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

In Preview

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

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;
```

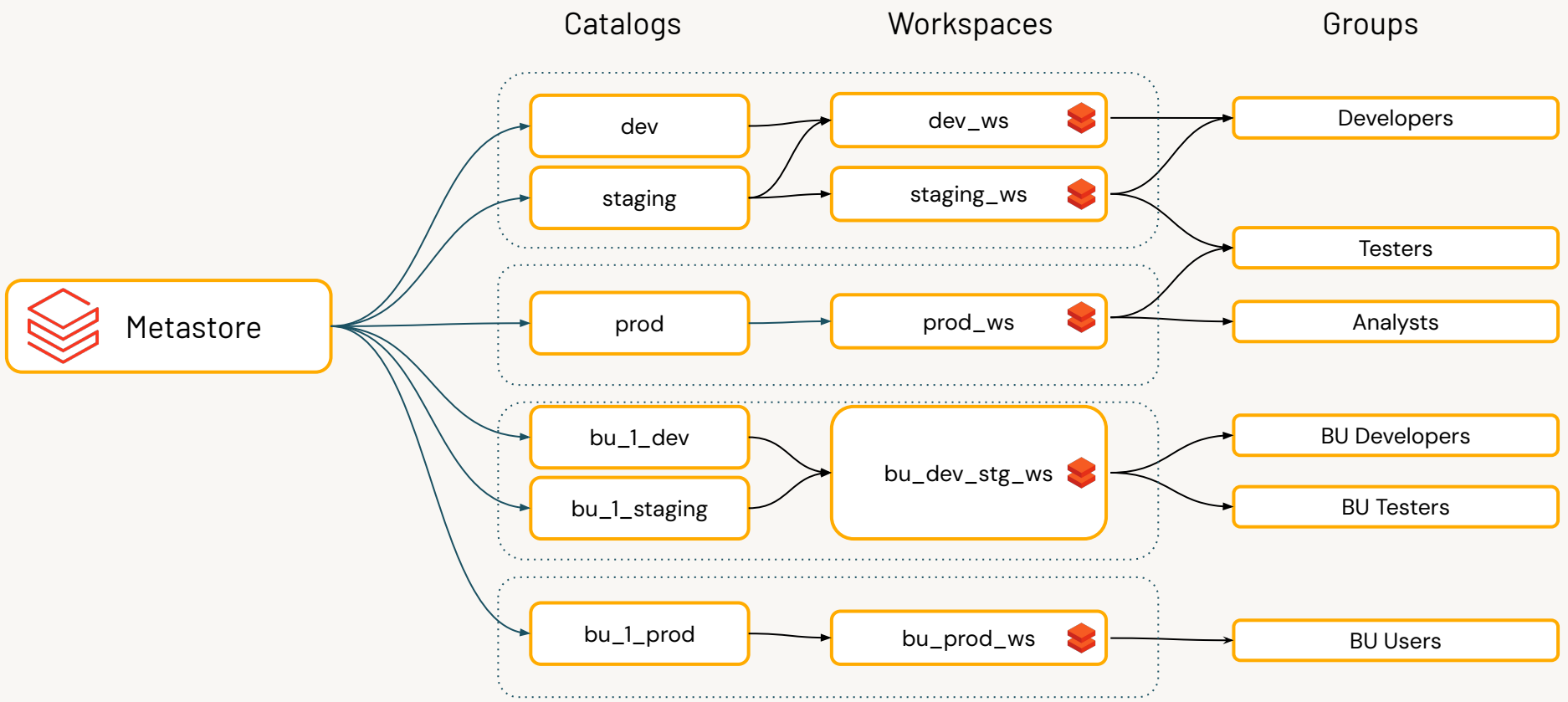
Test for group membership

Assign reusable mask to column

Specify mask or function to mask

Access data from specified environments only

Restrict catalog access by environment or purpose



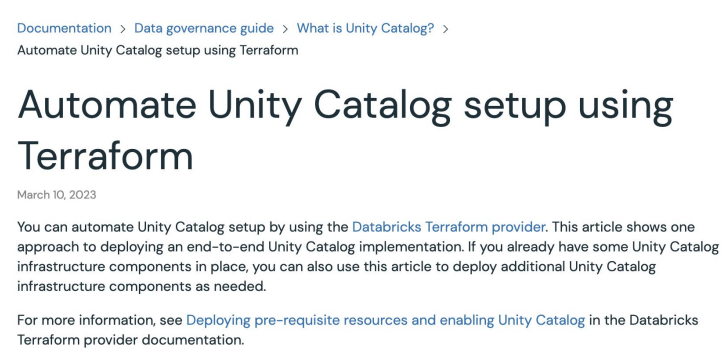
Access to data and availability of data can be isolated across workspaces and groups

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

In Preview

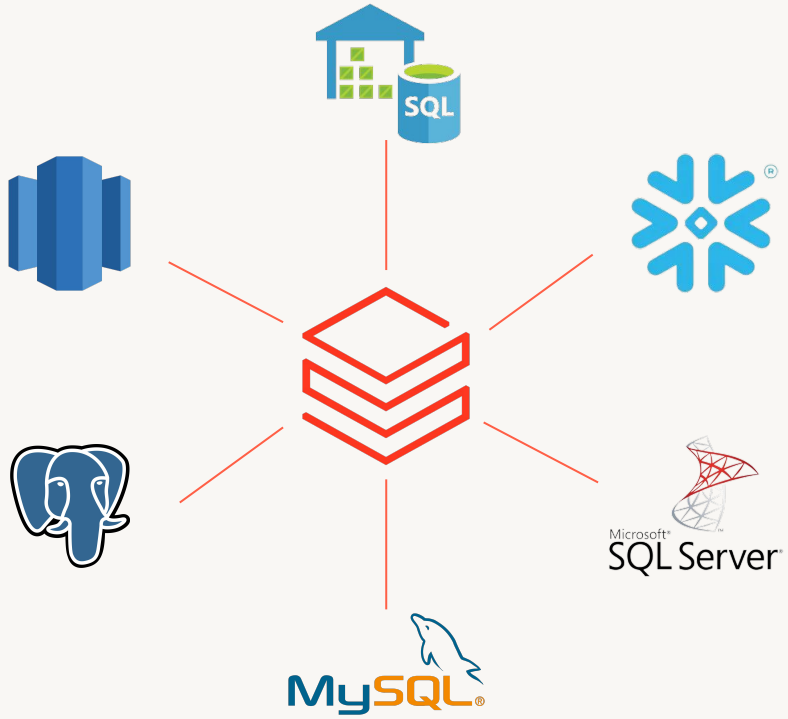
Query Federation

Unify your entire data estate with lakehouse

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

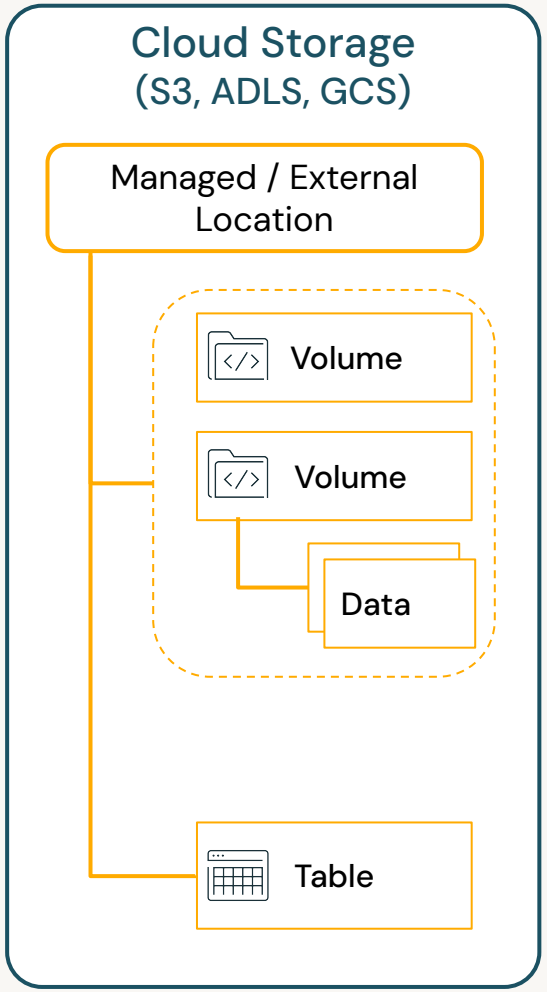
Volumes in Unity Catalog

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

- Volumes can be accessed by some POSIX commands

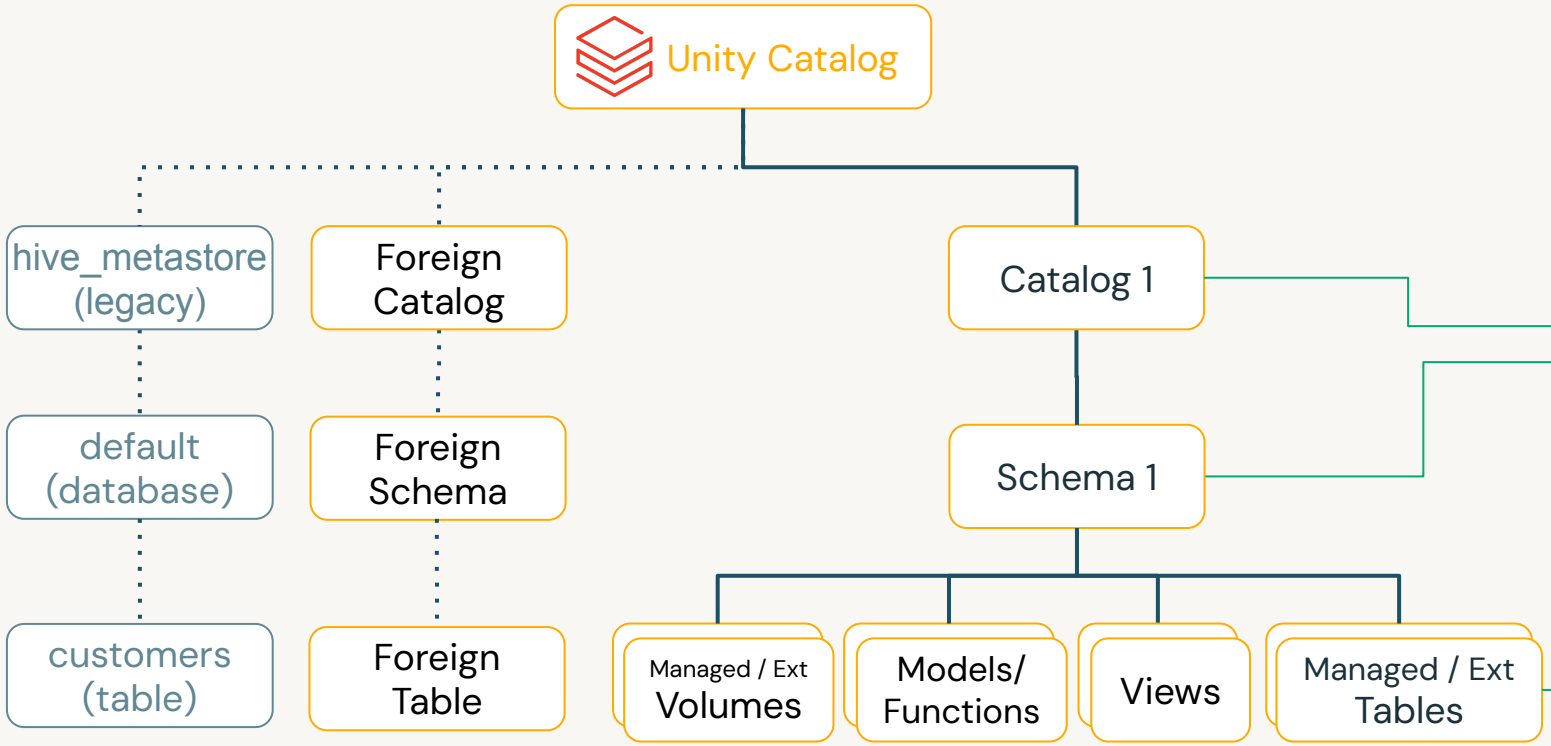
```
dbutils.fs.ls("s3://my_external_location/Volumes/catalog/schema/volume123")

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

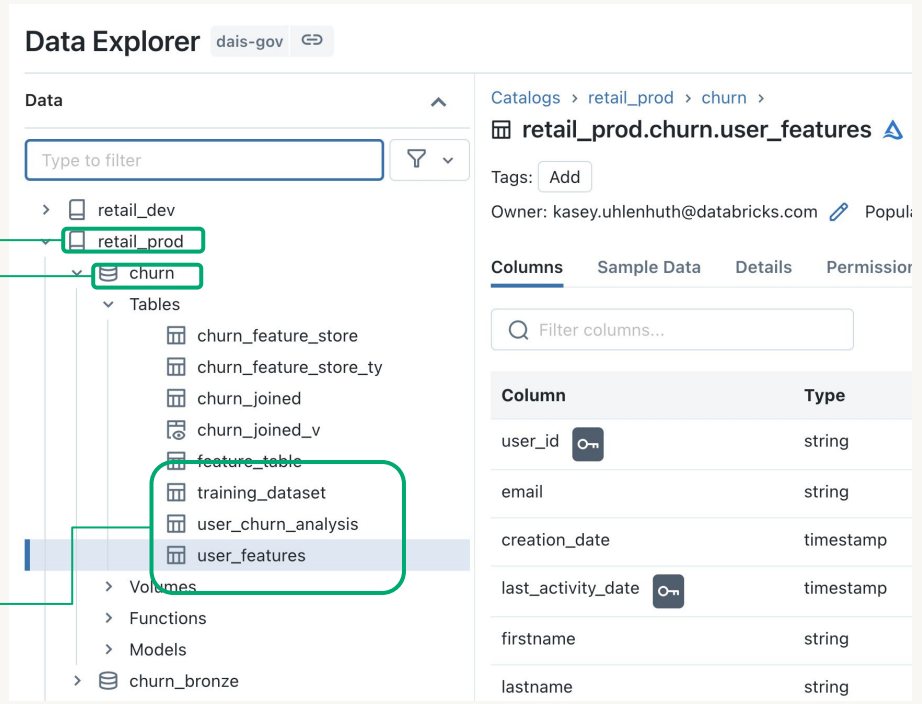
- Volumes are created under Managed or External Locations and show up in UC Lineage
- Volumes add governance over non-tabular data sets
  - Unstructured data, e.g., image, audio, video, or PDF files, used for ML
  - Semi-structured training, validation, test data sets, used in ML model training
  - Raw 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 `

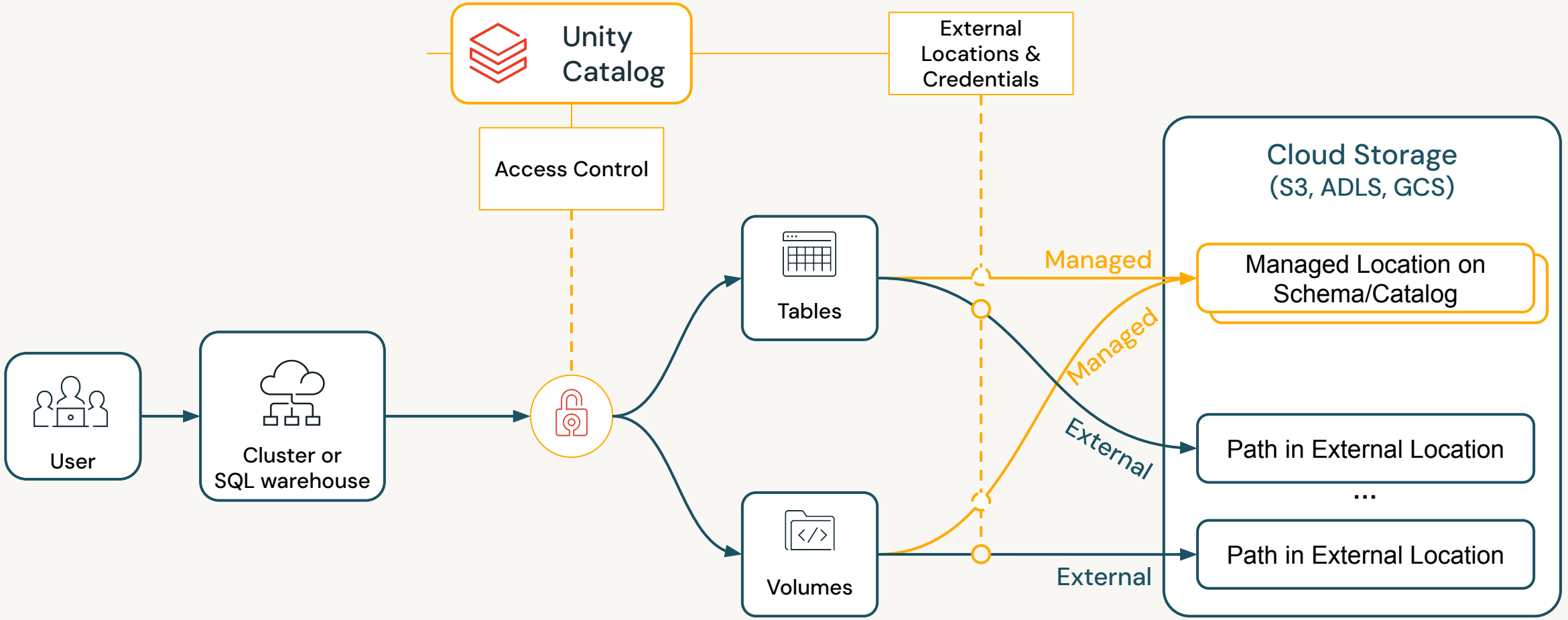





```
SELECT * FROM main.paul.red_wine;  -- <catalog>.<database>.<table>

SELECT * FROM hive_metastore.default.customers;

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

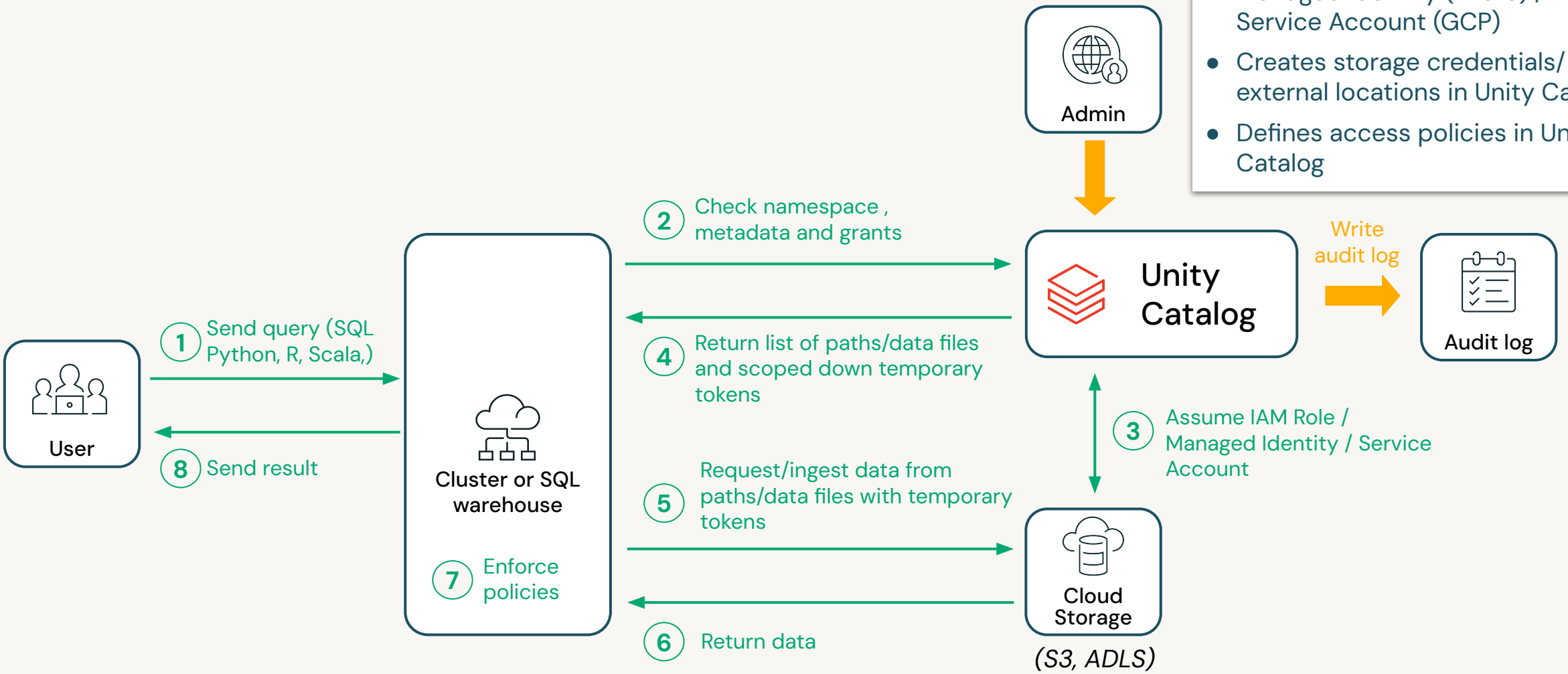
Defining file based data sources in Unity

Simplify data access management across clouds



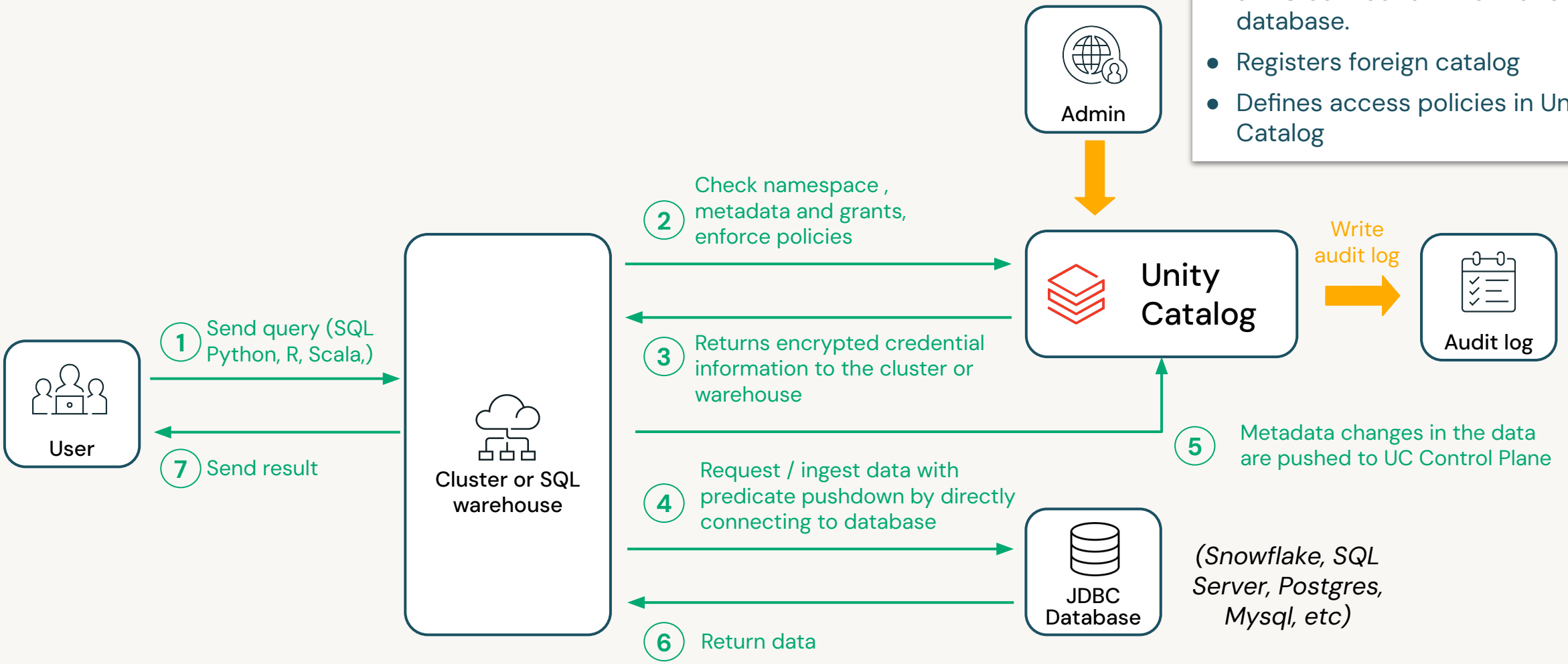
Querying file based data sources with Unity

- 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

- Defines Connection object with JDBC connection information to database.
- Registers foreign catalog
- Defines access policies in Unity Catalog



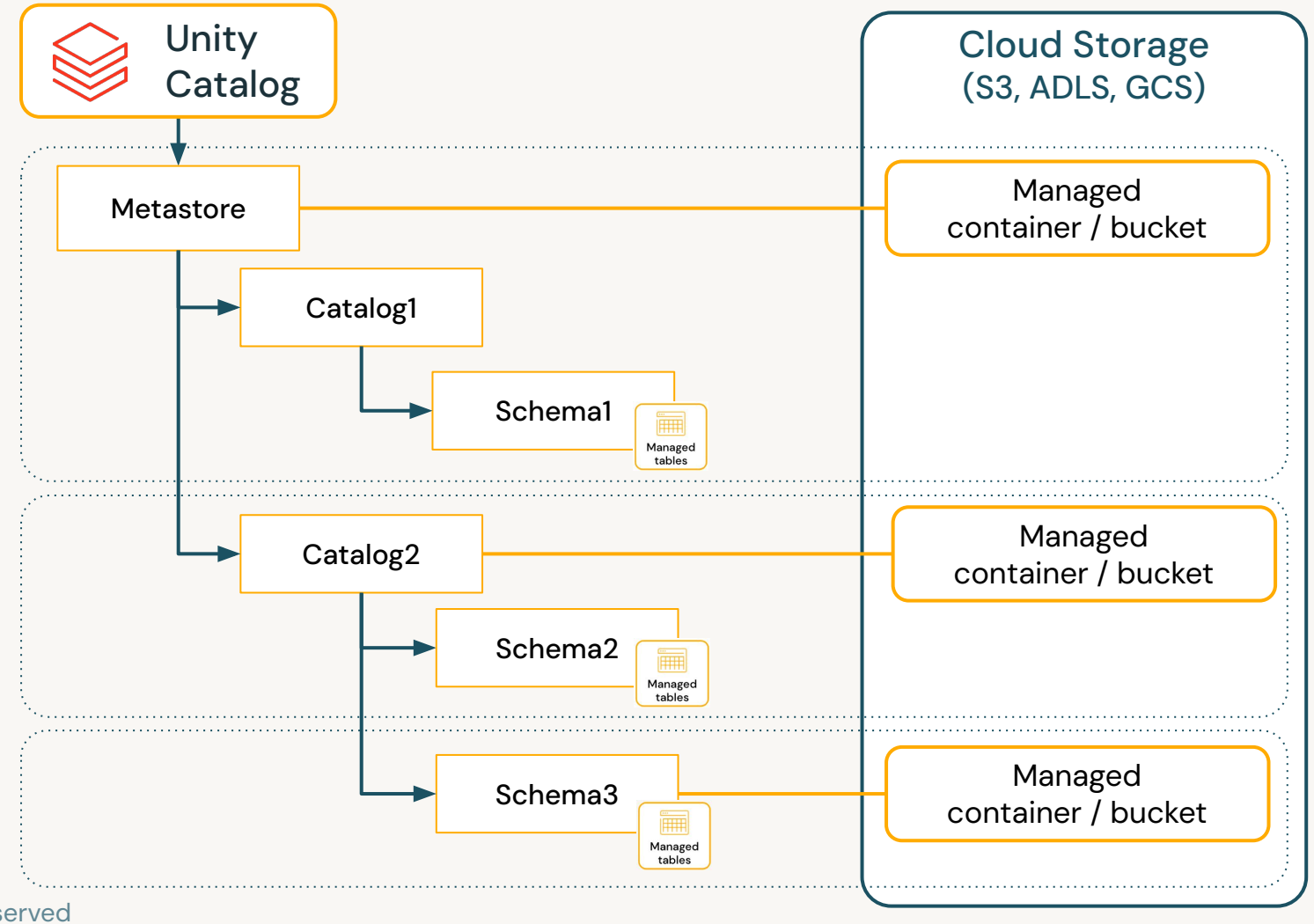
Isolation between file based data sources

Use managed data sources for data isolation or cost allocation

1) Store at the **metastore**

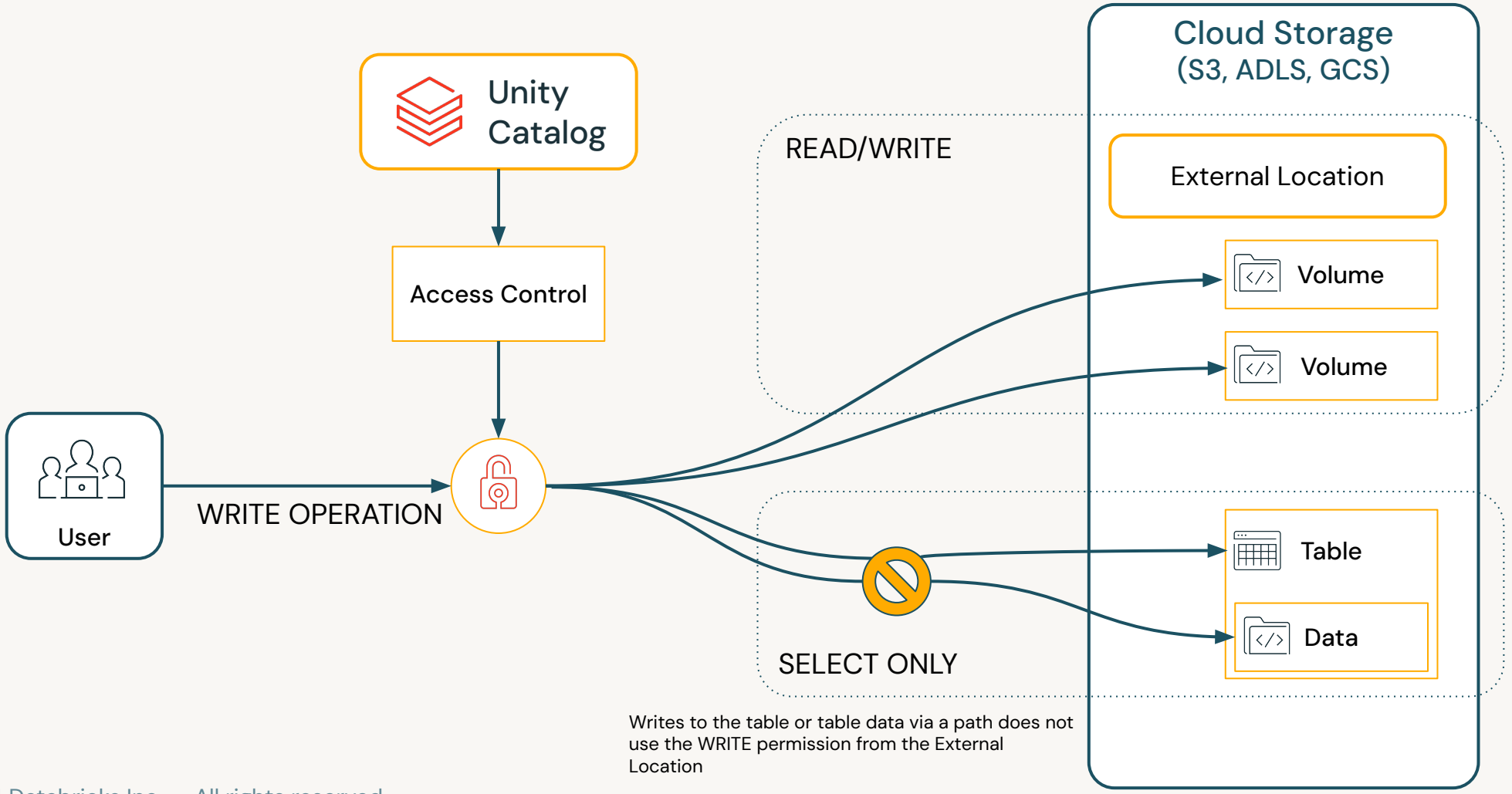
2) Store at the **catalog**

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

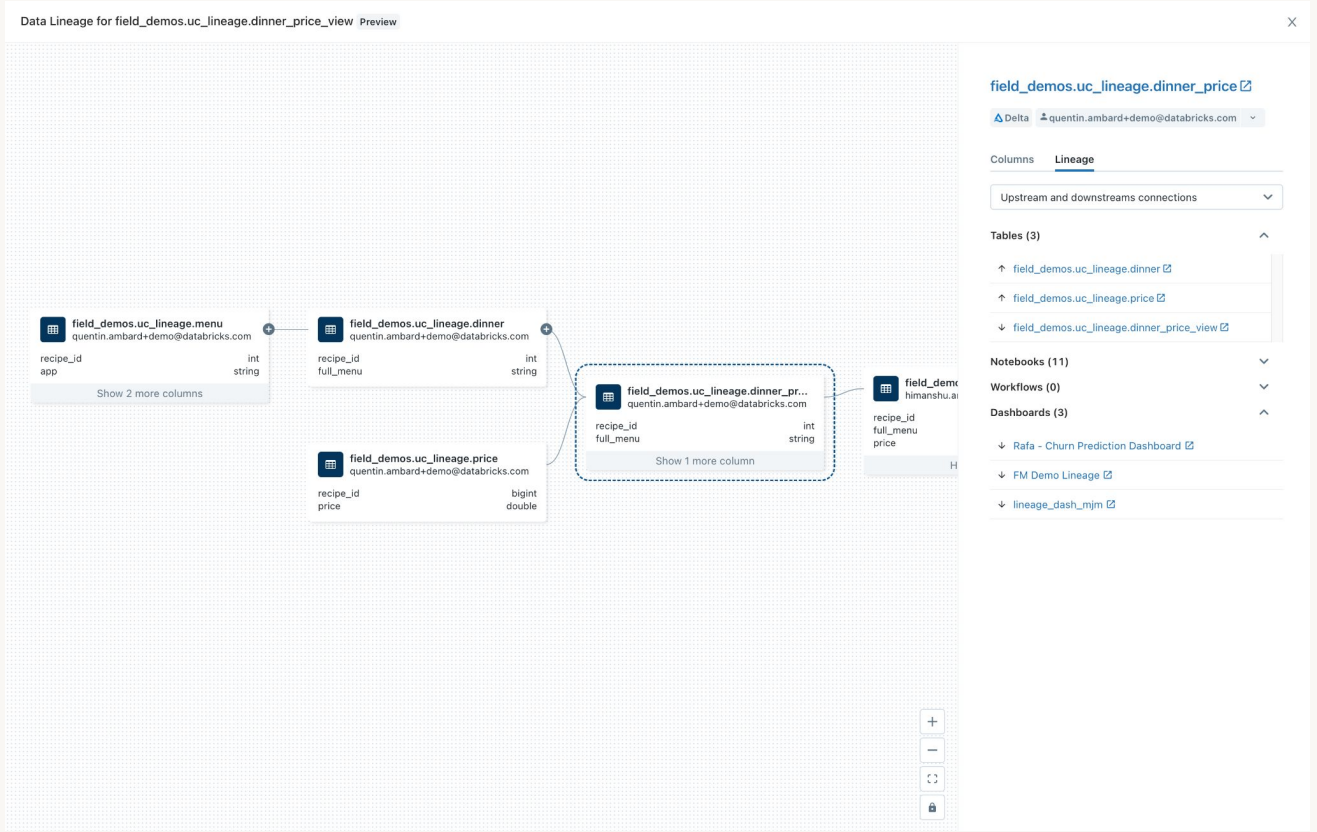
Observability

- 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

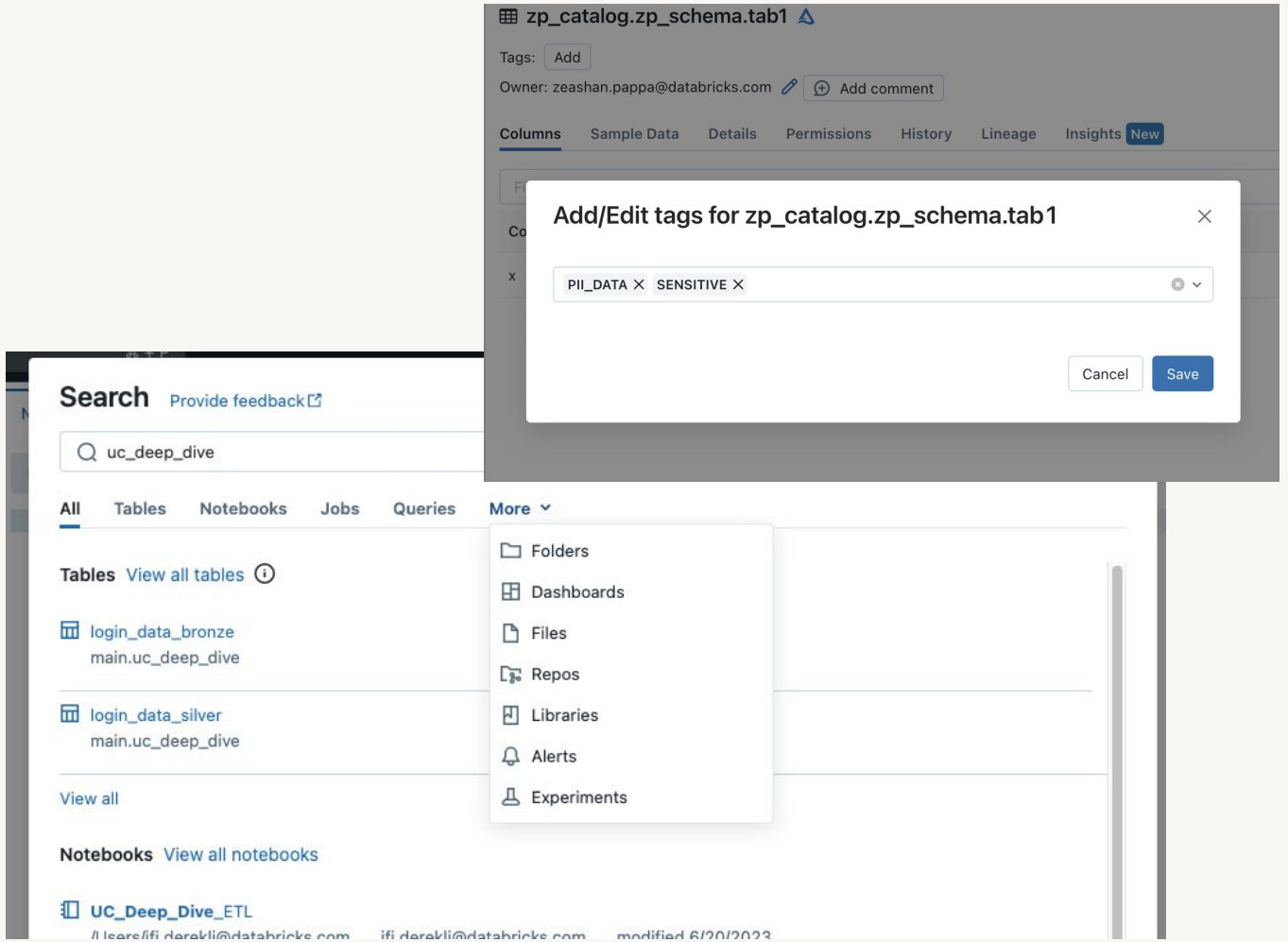
- Auto-capture runtime data lineage on a Databricks cluster or SQL warehouse
- Leverage common permission model from Unity Catalog
- Lineage across tables, columns, dashboards, workflows, notebooks, files, external sources, and models



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-real time, 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 = "unity Catalog"
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 = "unity Catalog"
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 = "unity Catalog"
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 = "unity Catalog"
AND action_name = "generateTemporaryTableCredential"
GROUP BY 1 ORDER BY 2 DESC LIMIT 1;
```

System Tables: Billing Logs

Understand cost allocation across your data estate

In Preview

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";
```

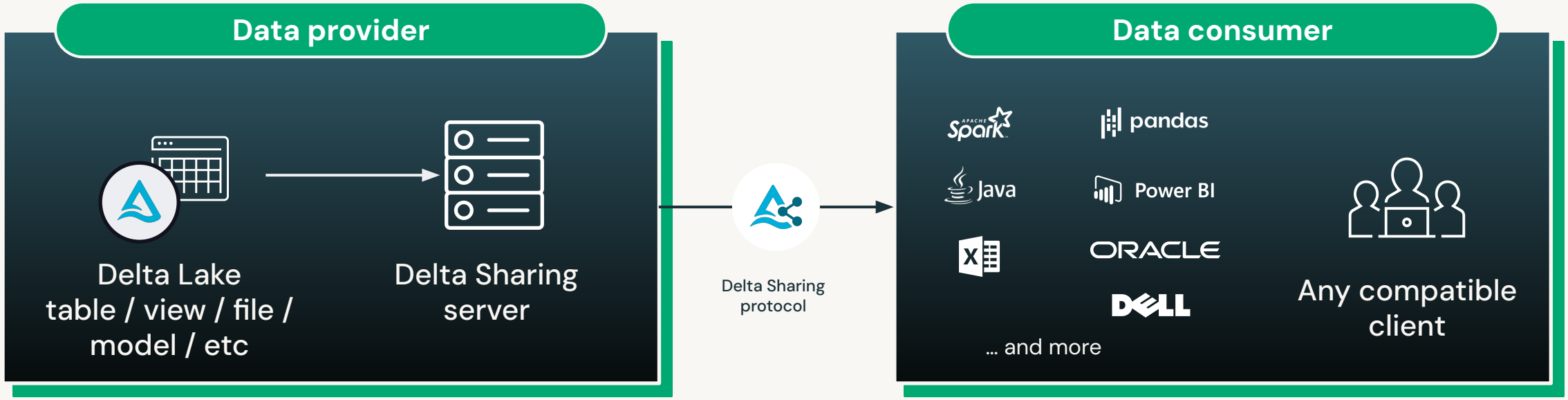
What user queries read from this table?

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

Open Collaboration 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

An open marketplace for data, analytics, and AI

Data sets, Notebooks, ML models and applications from top data & solution providers

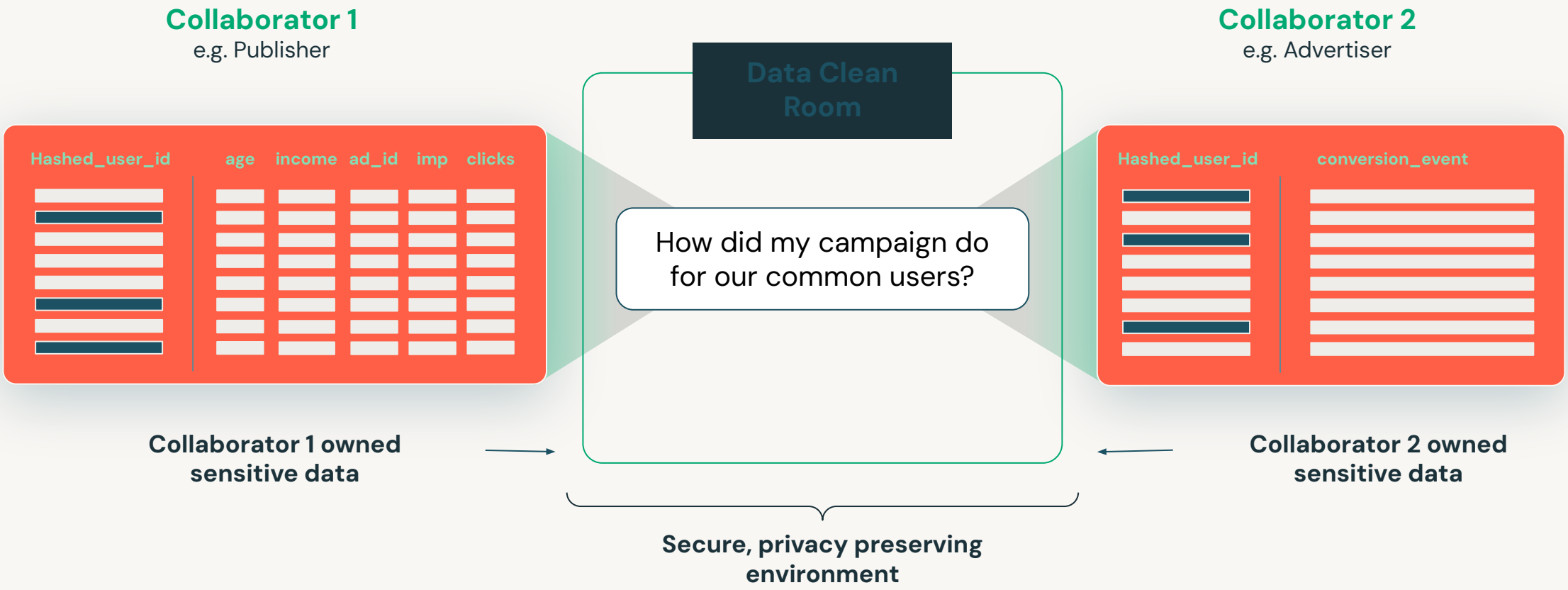
---

Public marketplace, private exchanges

Open for Databricks & non-Databricks users

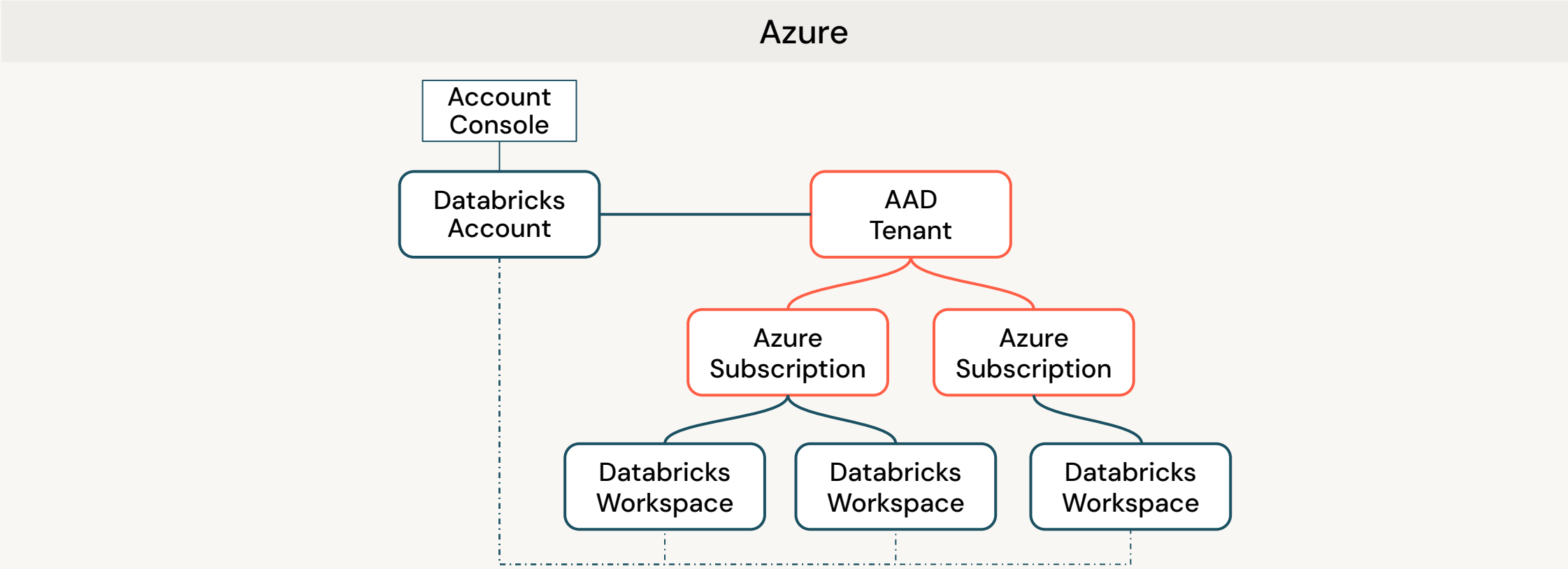
Clean rooms

Secure environments to run computations on joint data

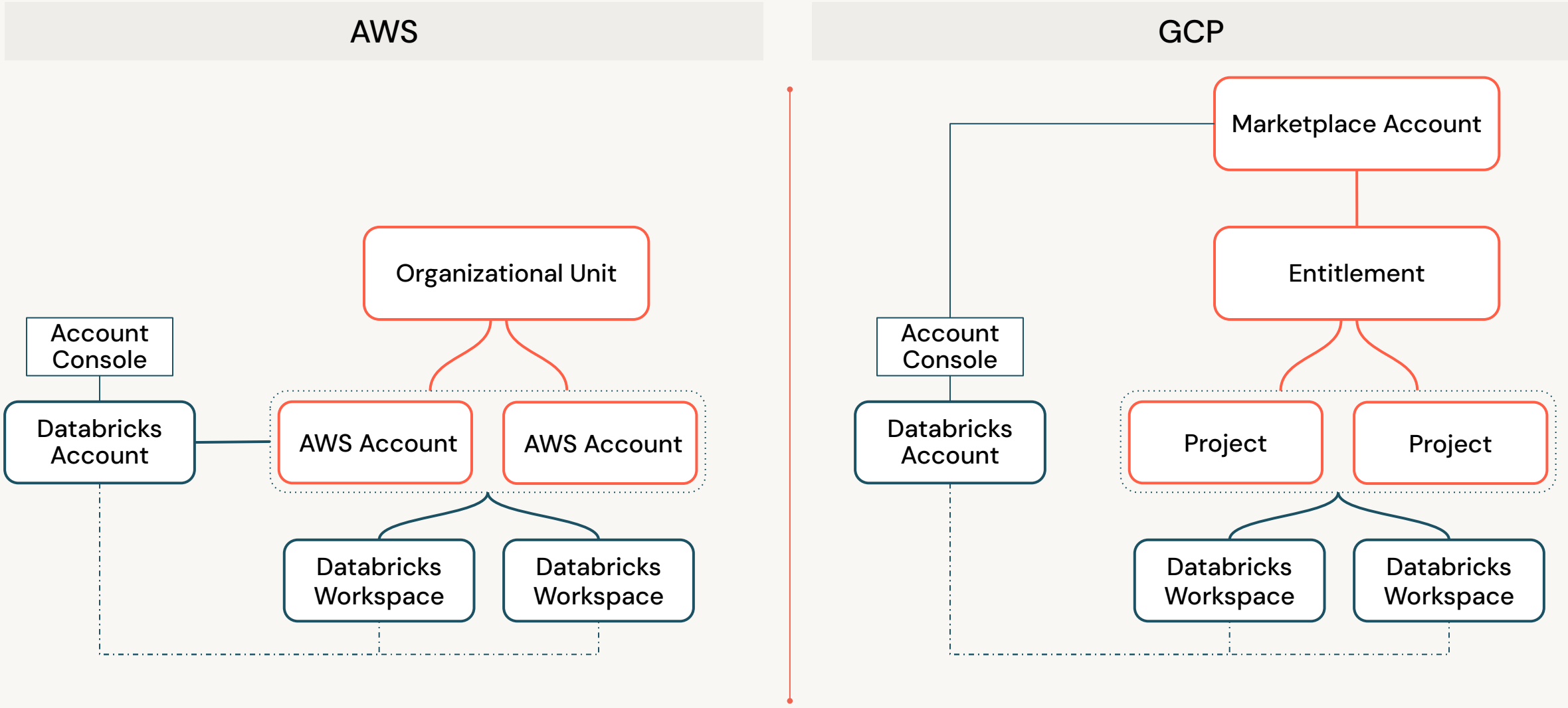


Unity Catalog and cloud providers

Databricks Accounts and Cloud Providers



Databricks Accounts and Cloud Providers



Unity Catalog and Cloud Constructs

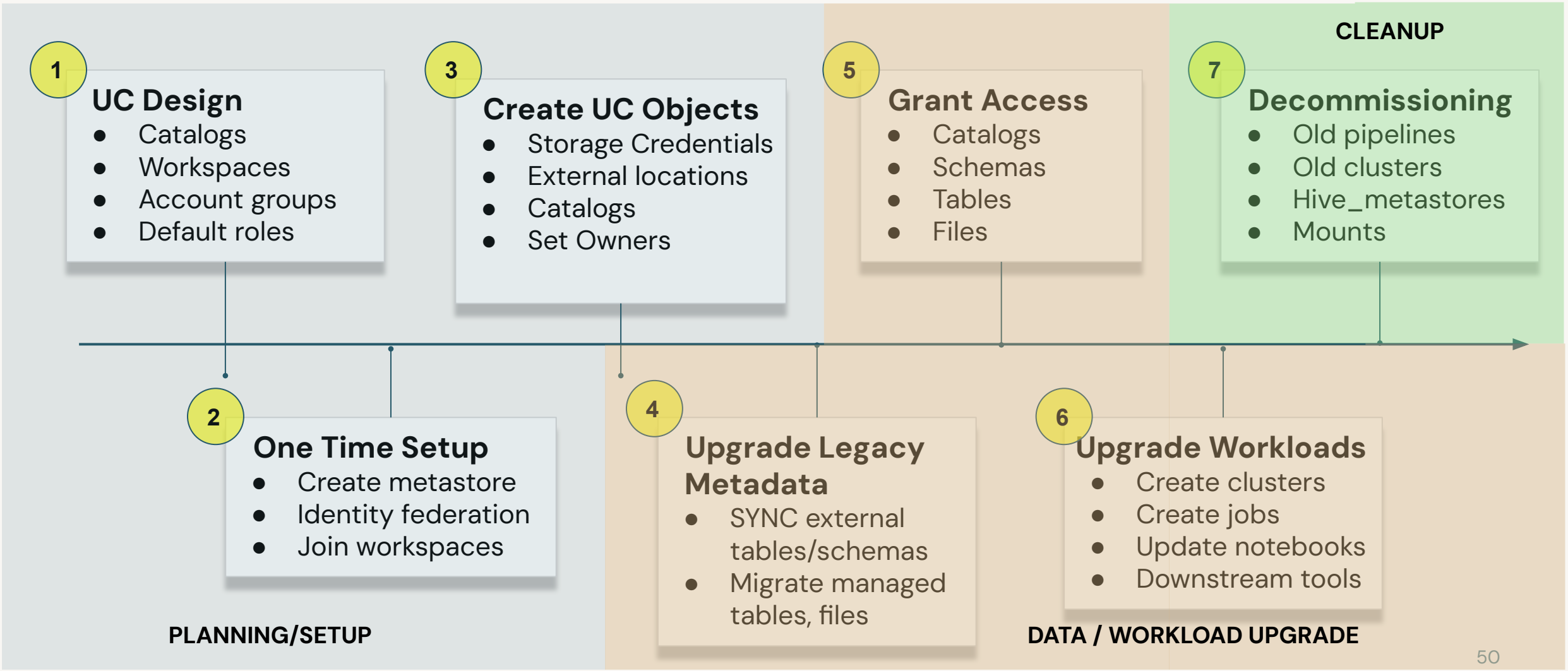
	AWS	Azure	GCP
Databricks Account	Accounts	Tenant	Marketplace Account
Metastore	Region	Region	Region
Catalog	*Account**	*Subscription**	*Project**
Storage Location	S3 Bucket	ADLS Account	GCS Bucket
Credential	IAM Role	Managed Identity	Service Account

* *Minimum one, more are optional*

Upgrade to Unity Catalog

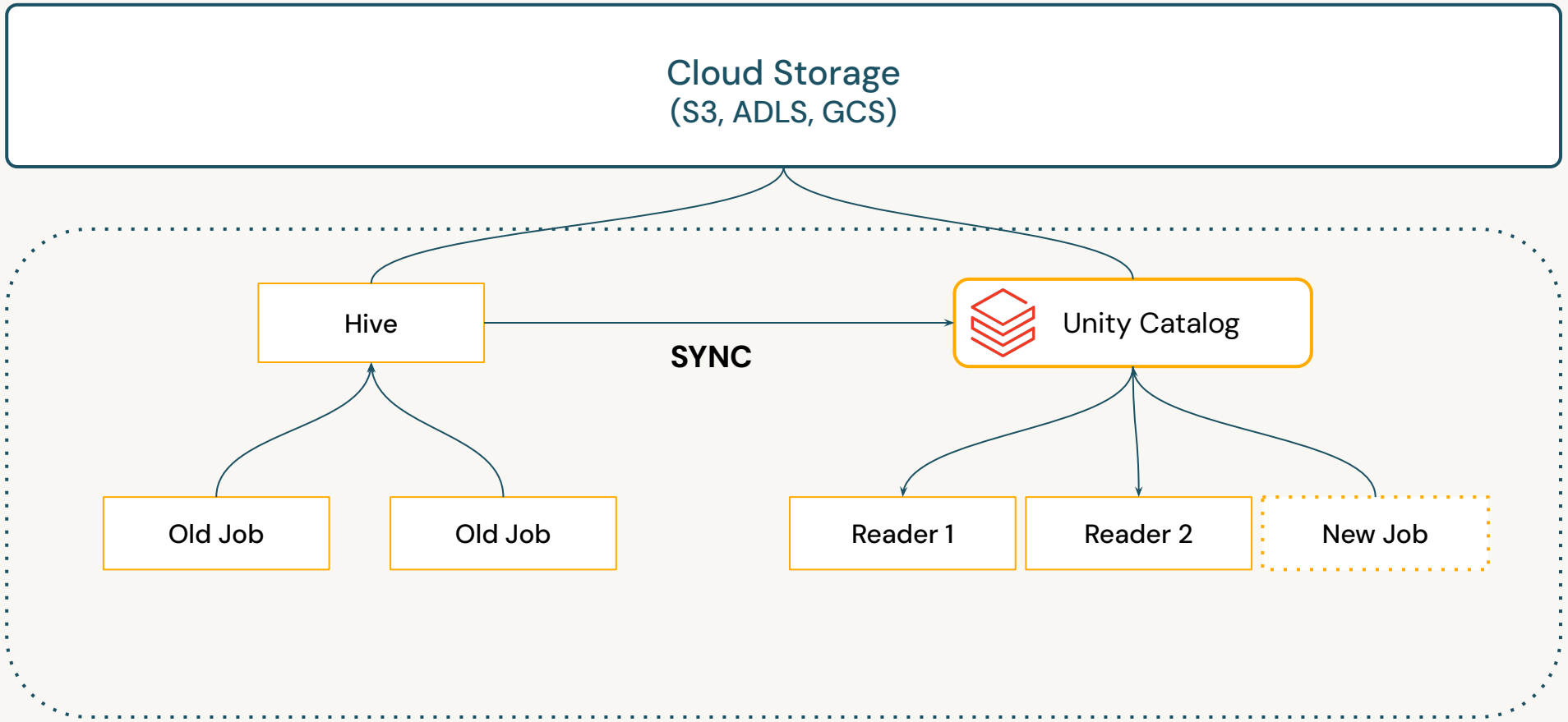
How to upgrade to Unity Catalog

Steps to consider for a full upgrade



An example transition

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
```

Demo

Thank you!