



Databricks Unity Catalog

Unified governance for data and AI

Date

Name



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

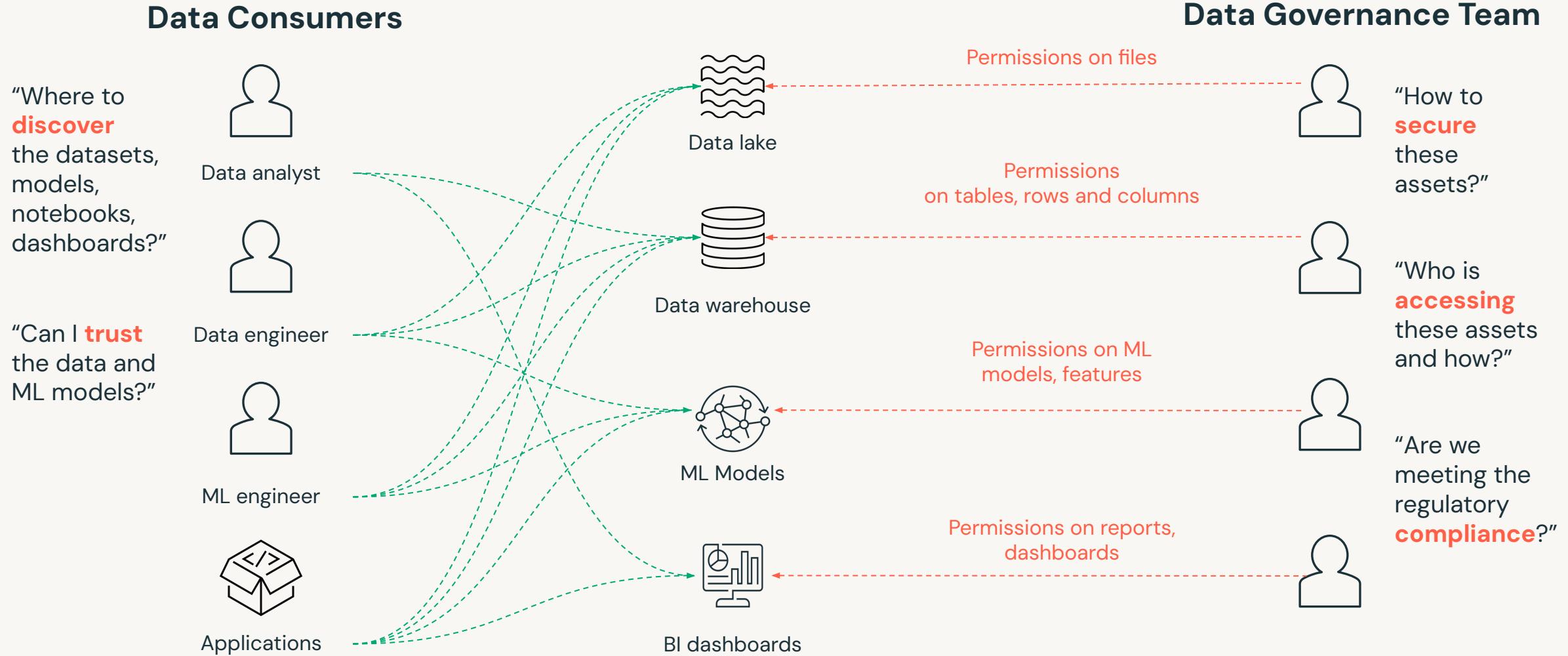
“Organizations seeing the **highest returns** from AI have a framework for **AI governance** to cover every step of the model development process”

—
The State of AI in 2022, McKinsey & Co

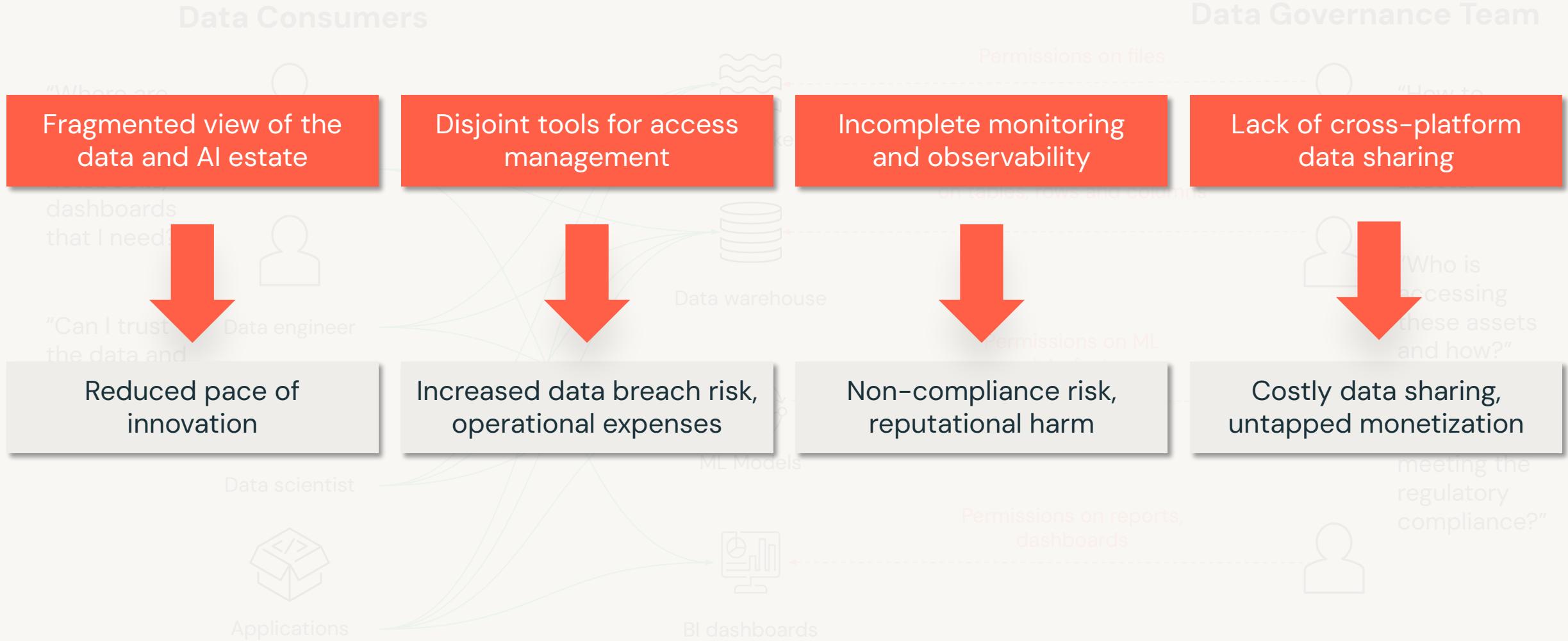
“AI is now an enterprise essential, and as such, **AI governance** will join cybersecurity and compliance as a **board-level topic**”

—
Forrester, 2023 AI Predictions report

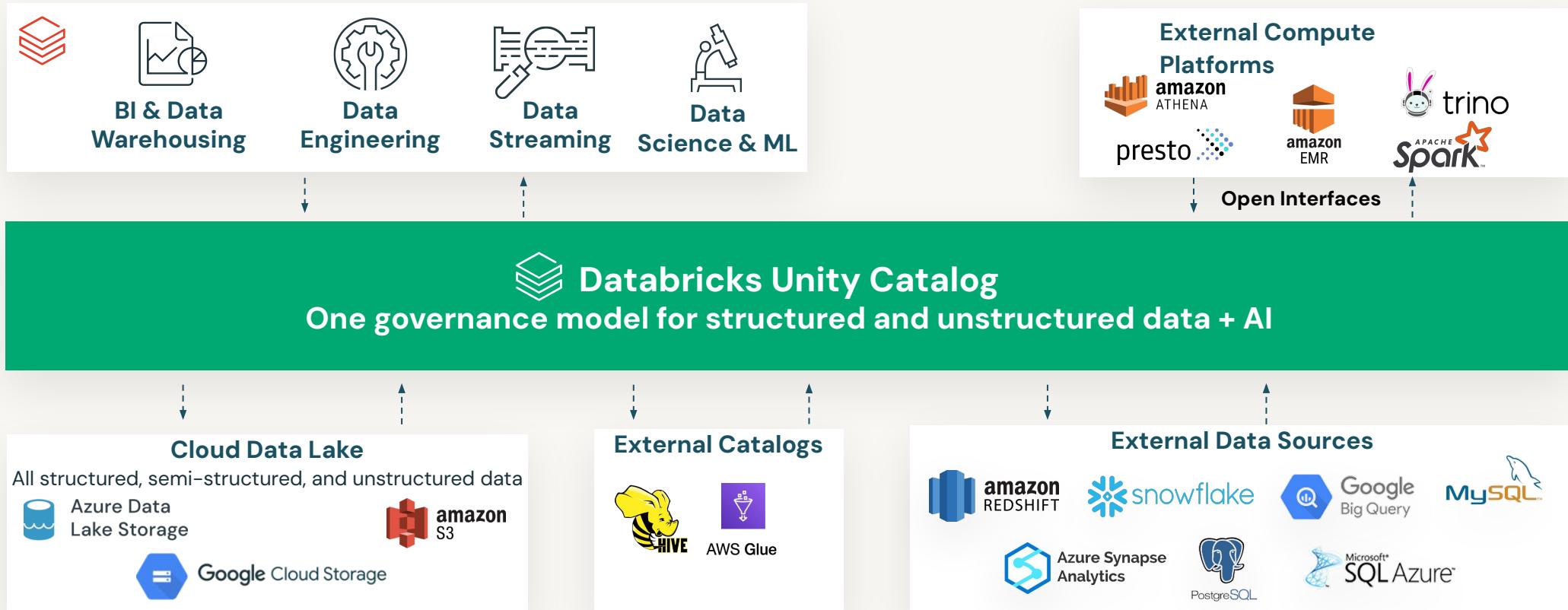
Today, data and AI governance is complex



Today, data and AI governance is complex



Databricks Lakehouse unifies data and AI governance



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 Unity Catalog

Discovery

Access
Controls

Lineage

Monitoring

Auditing

Data
Sharing

Tables



Files



Models



Notebooks

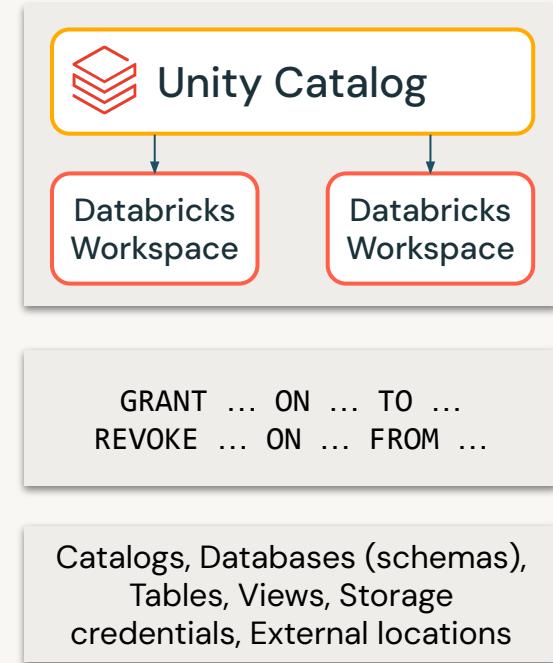


Dashboards



Unity Catalog – Key Capabilities

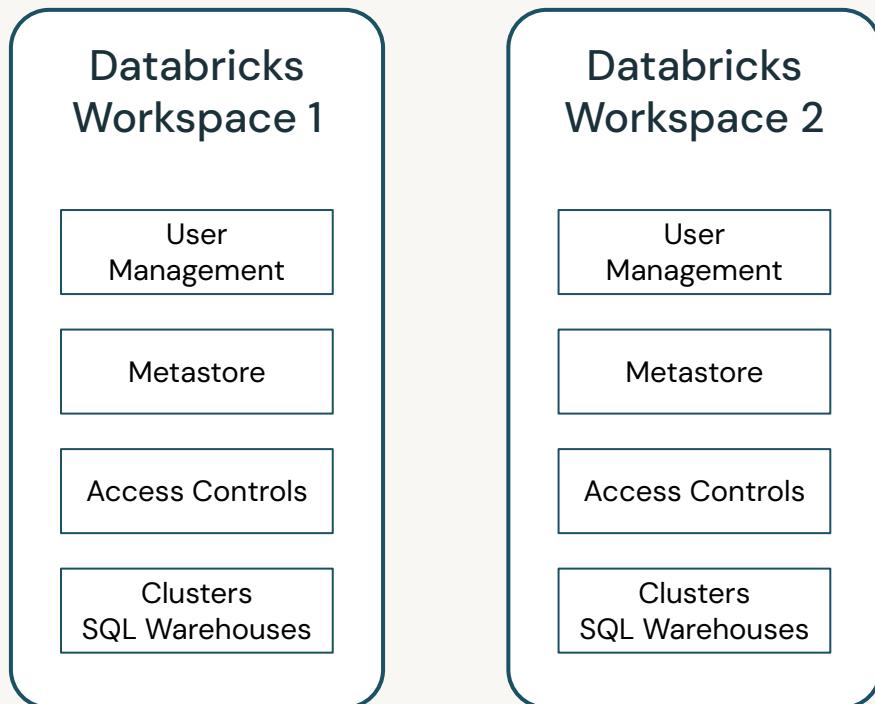
- Centralized metadata and user management
- Centralized access controls
- Data lineage
- Data access auditing
- Data search and discovery
- Secure data sharing with Delta Sharing



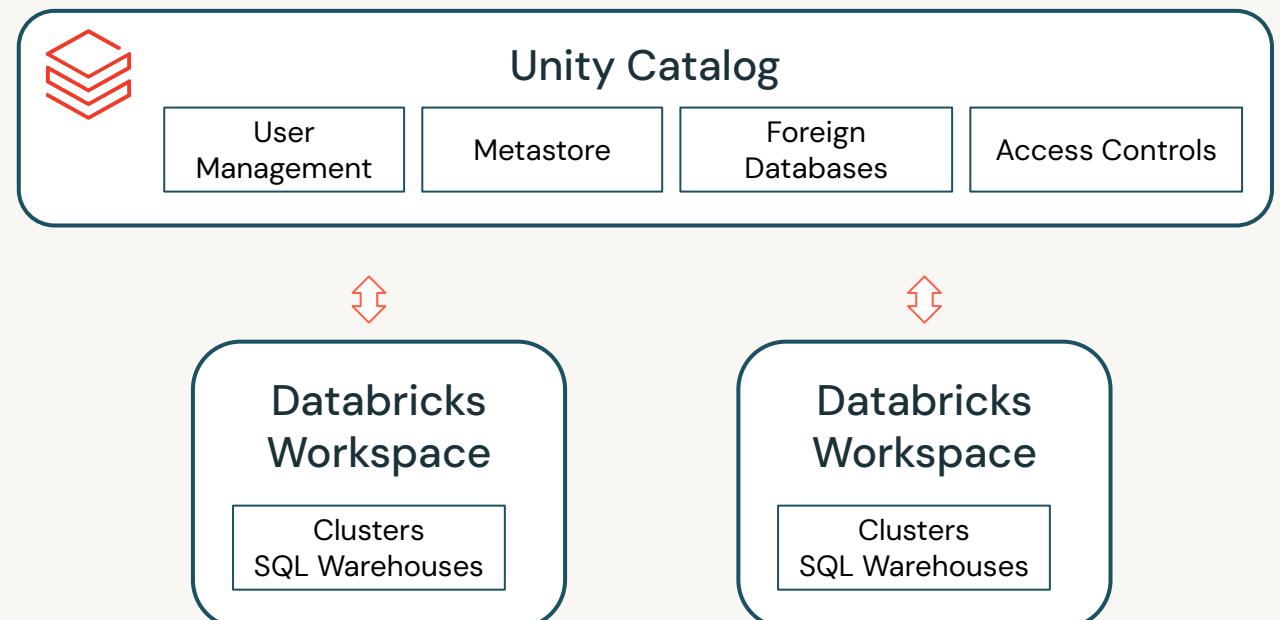
All your metadata, in one place

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

Without Unity Catalog



With Unity Catalog



Fundamental Concepts

Working with file based data sources

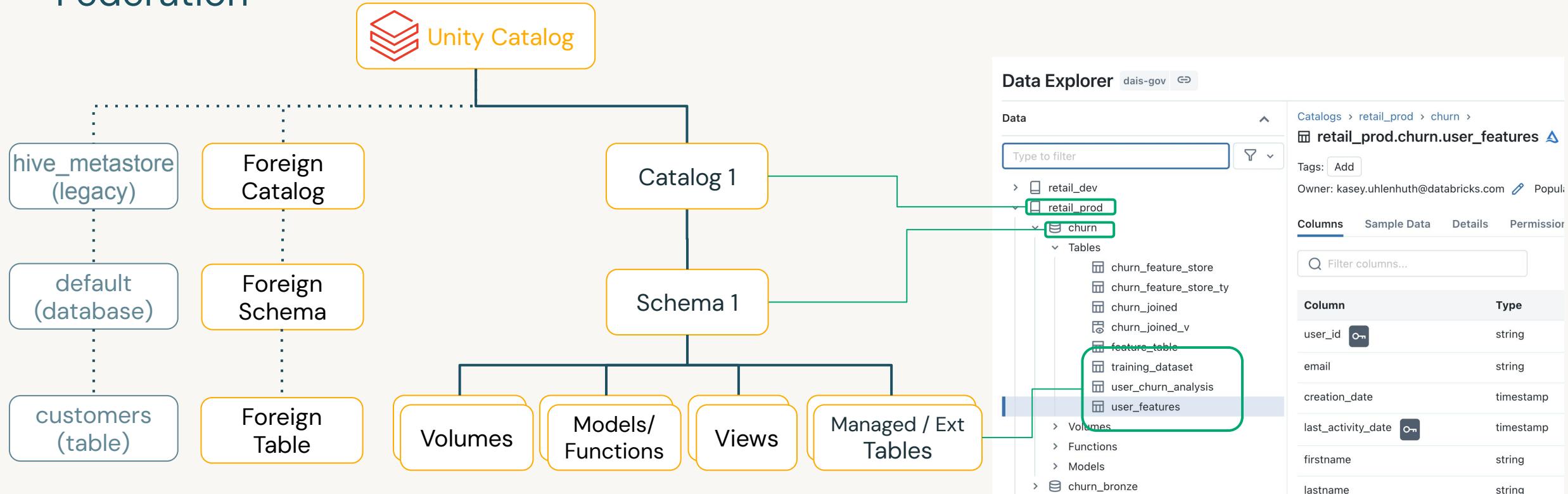
- Credentials
 - Cloud provider credential to connect to storage
- External Locations
 - Storage location used for external tables or arbitrary files
- Managed Data Sources
 - External Location that is used exclusively for tabular data
- Volumes
 - Arbitrary file container inside an external location

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

Using ANSI SQL DCL

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

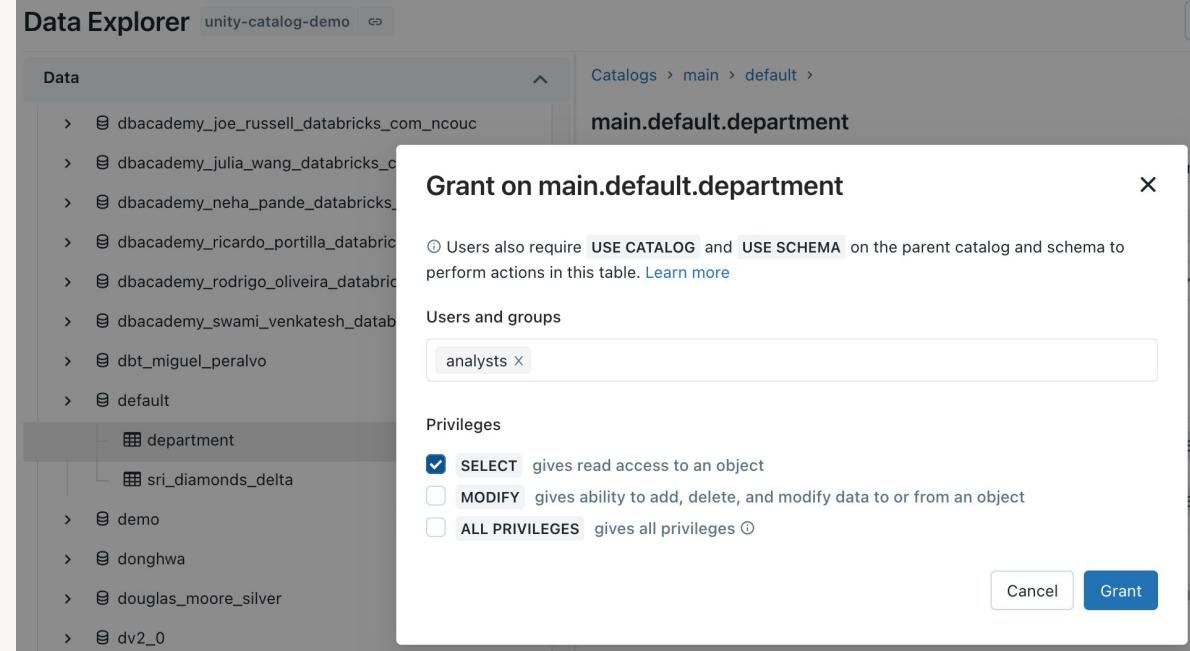
```
GRANT SELECT ON iot.events TO engineers
```

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 datasets

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

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.

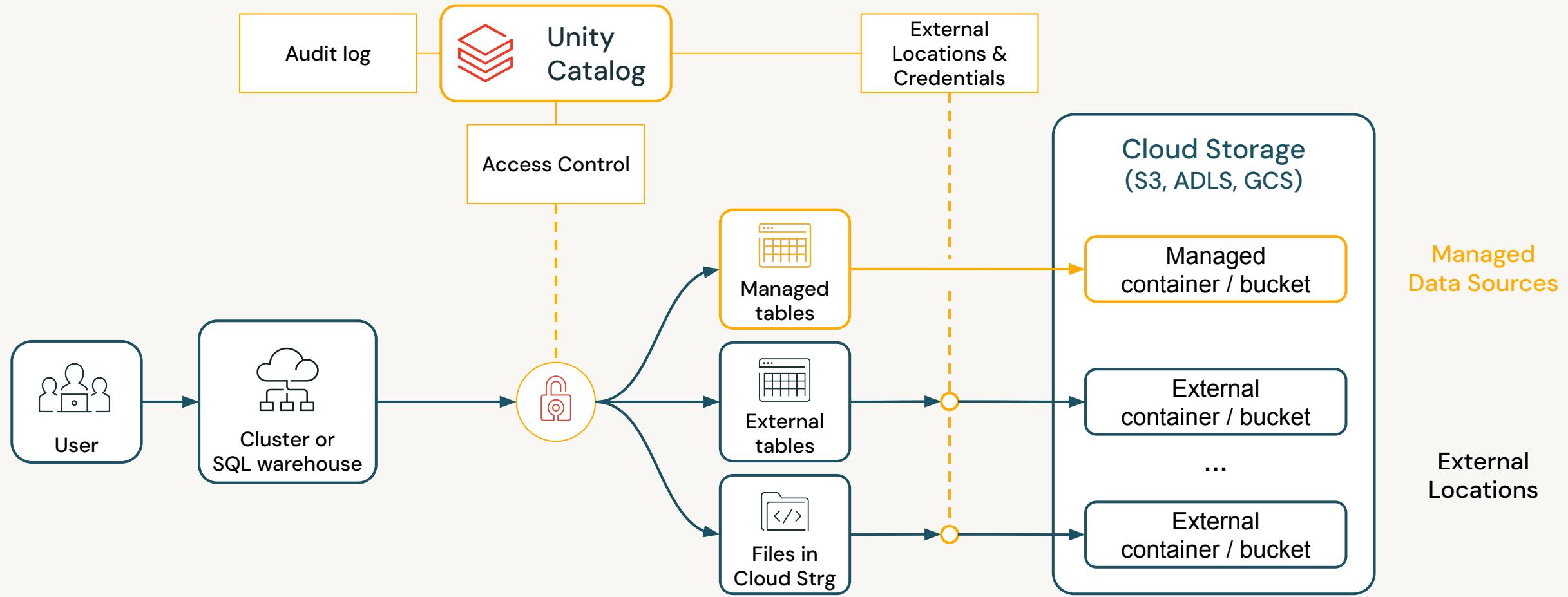
For more information, see [Deploying pre-requisite resources and enabling Unity Catalog](#) in the Databricks Terraform provider documentation.

```
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"]  
  }  
}
```



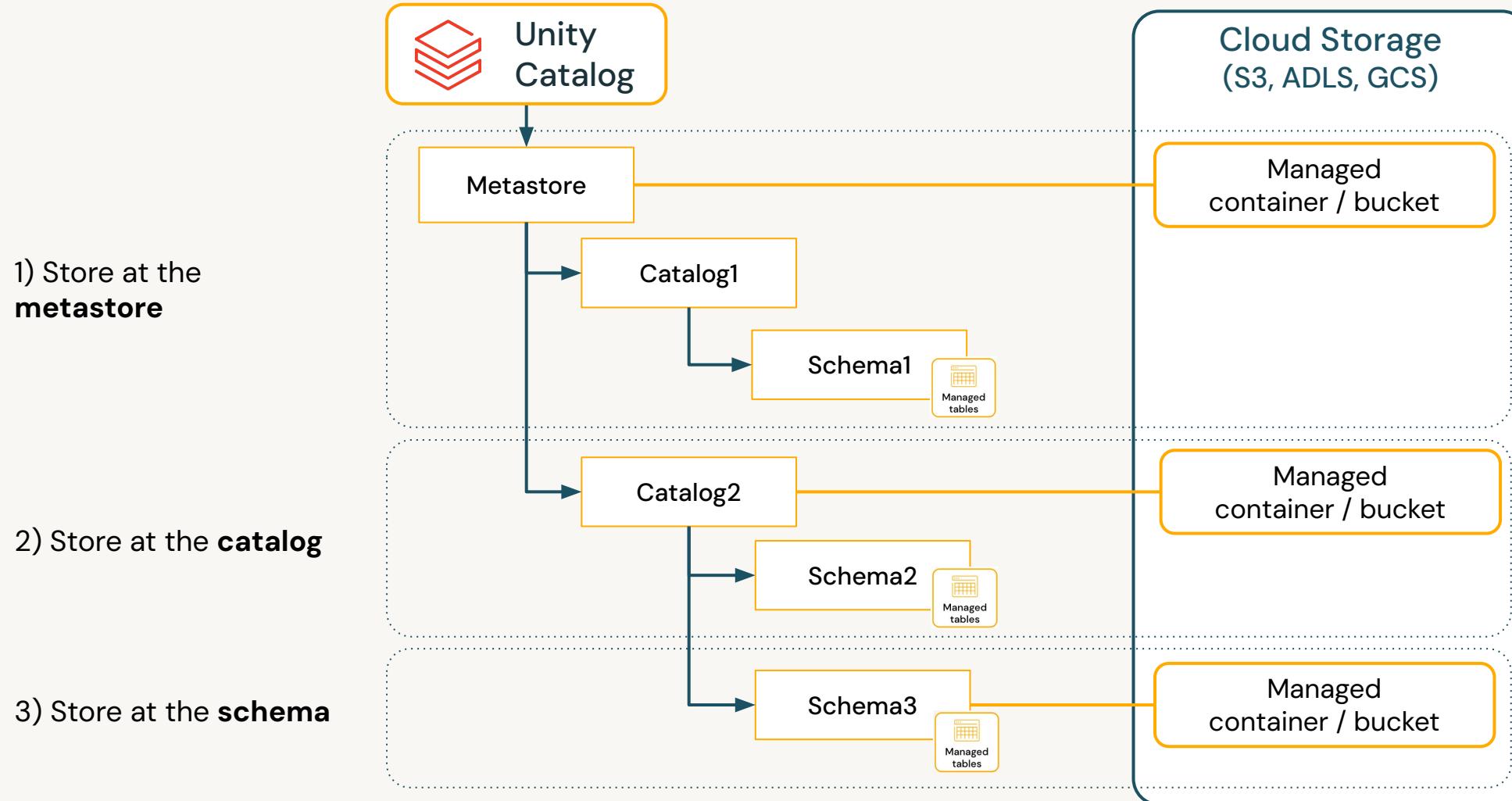
Managed Data Sources & External Locations

Simplify data access management across clouds



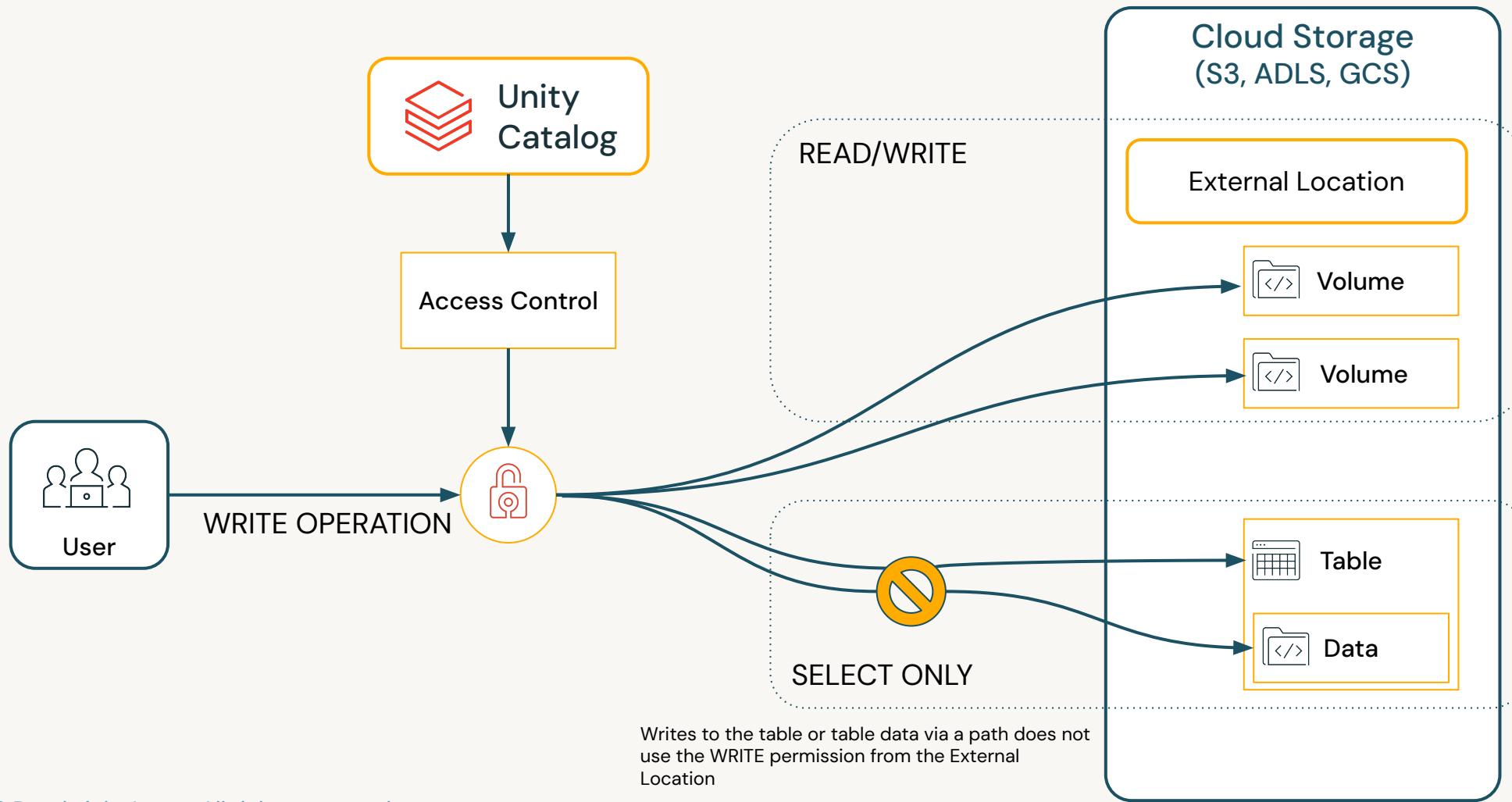
Default access to storage by catalog or schema

Use managed data sources for data isolation or cost allocation



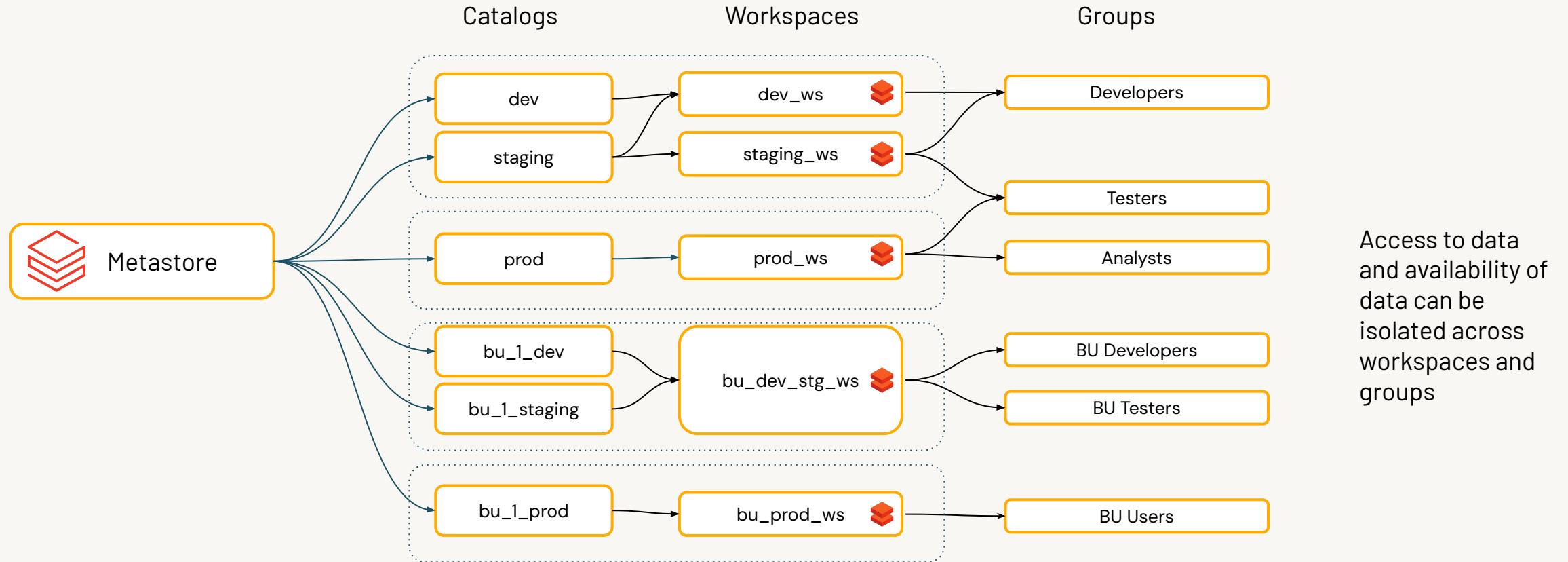
Govern filesystems and objects distinctly

Govern external tables and filesystem access separately

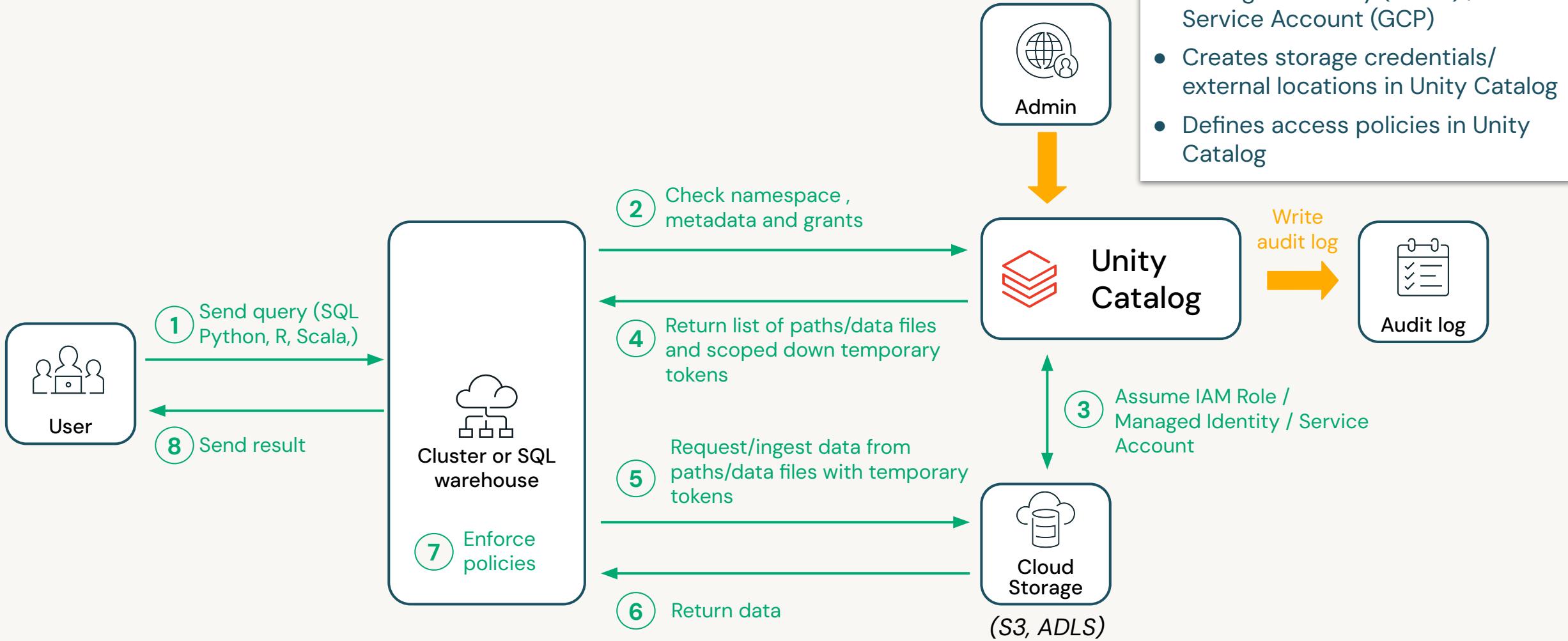


Access data from specified environments only

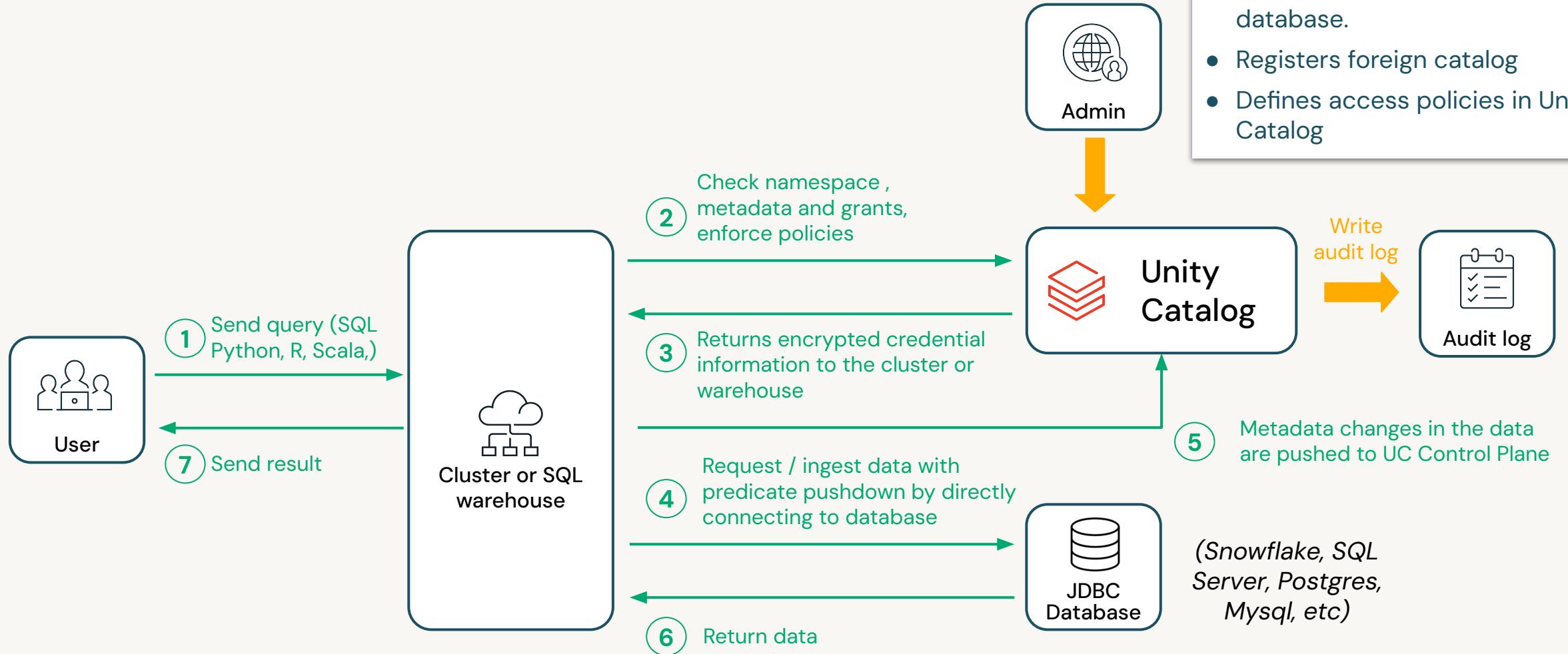
Restrict data access by environment or purpose



Querying file based data sources with Unity



Querying database sources with Unity

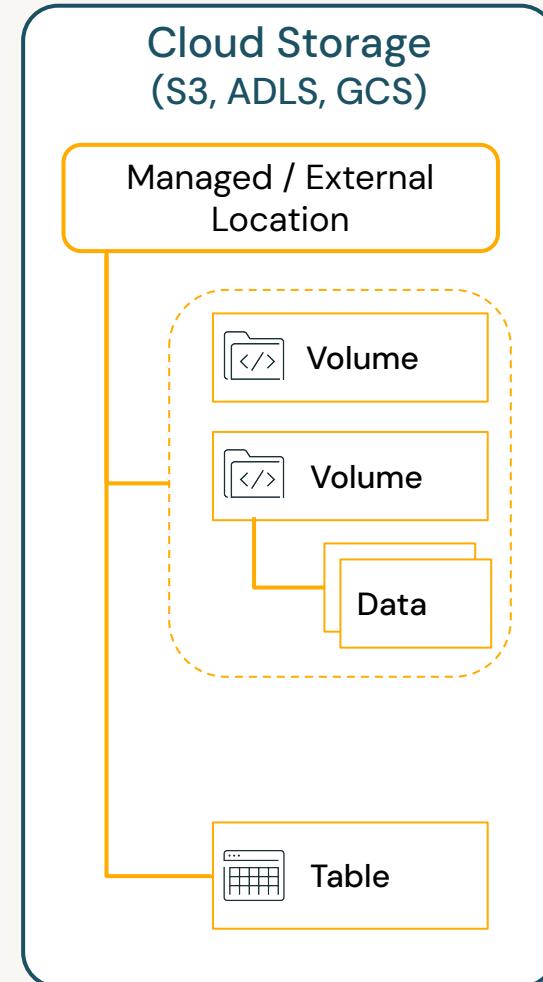


Volumes in Unity Catalog

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

- Volumes can be accessed by POSIX commands

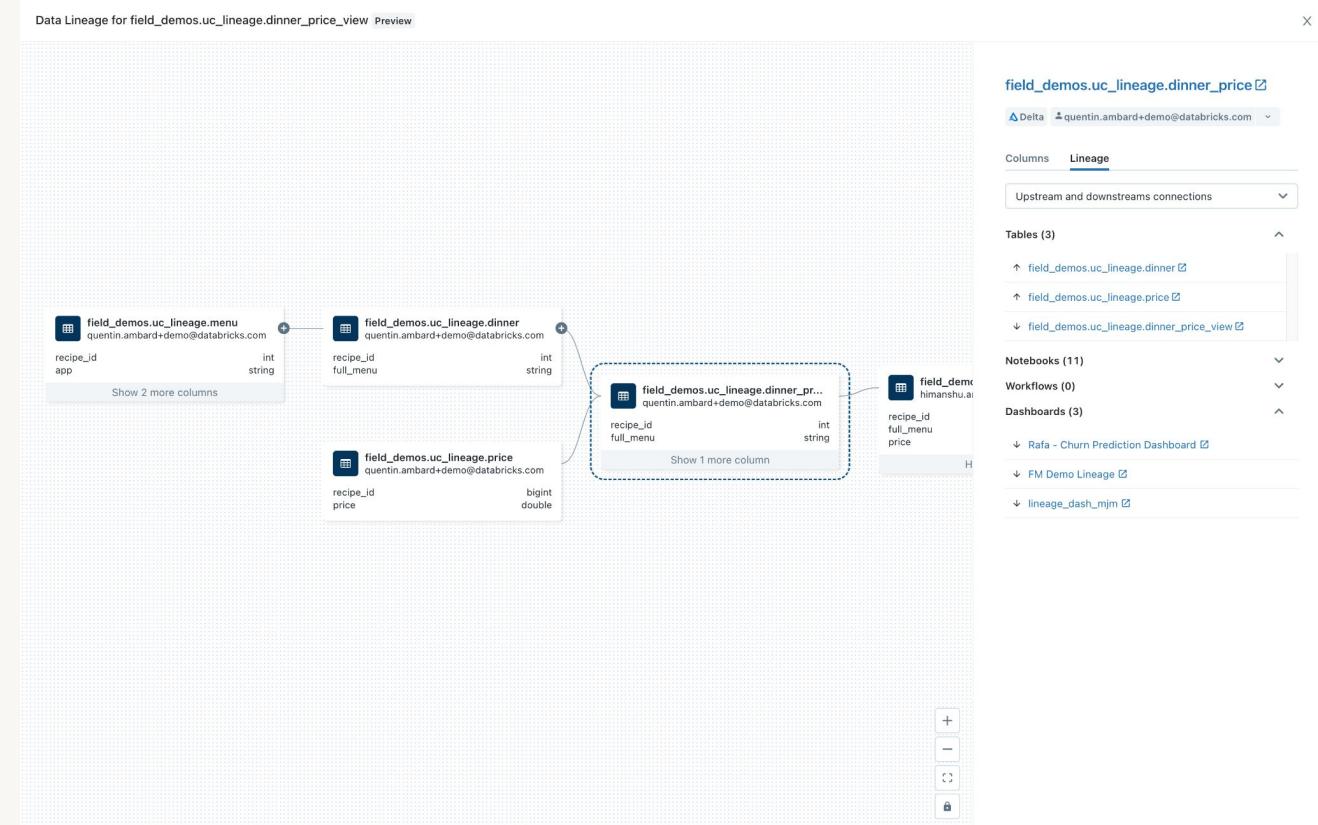
```
dbutils.fs.ls("s3://my_external_location/Volumes/volume123")  
ls /Volumes/volume123
```
- Volumes are created under Managed or External Locations and show up in UC Lineage
- Volumes add governance over non-tabular data sets
 - Unstructured data, e.g., image, audio, video, or PDF files, used for 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



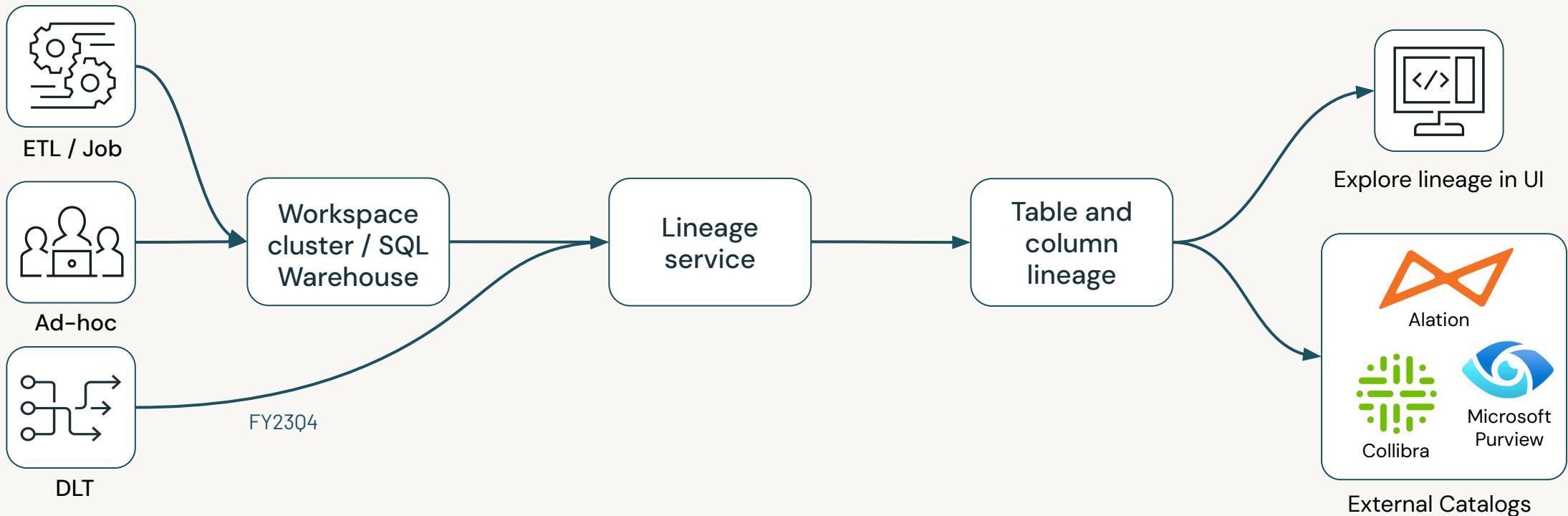
Automated lineage for all workloads

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
- Track lineage down to the table and column level
- Leverage common permission model from Unity Catalog
- Lineage across tables, dashboards, workflows, notebooks, feature tables, files, and DLT



Lineage flow – How it works

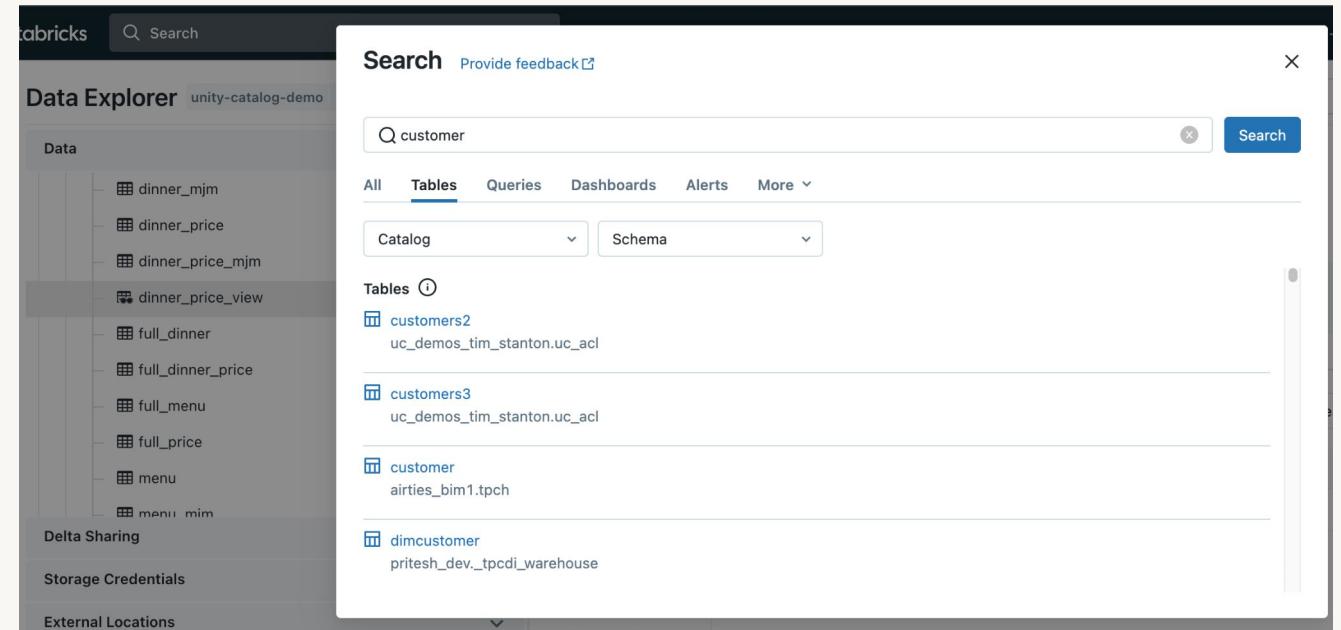


- Code (any language) is submitted to a cluster or SQL warehouse or DLT* executes data flow
- Lineage service analyzes logs emitted from the cluster, and pulls metadata from DLT
- Assembles column and table level lineage
- Presented to the end user graphically in Databricks
- Lineage can be exported via API and imported into other tool

Built-in search and discovery

Accelerate time to value with low latency data discovery

- UI to search for data assets stored in Unity Catalog
- Unified UI across DSML + DBSQL
- Leverage common permission model from Unity Catalog
- Apply semantic tags to data and search across tags



Discovery Tags

Semantic layer for your lakehouse

Problem

Searching for data assets in business terms or generally agreed upon taxonomies usually requires additional catalog tools.

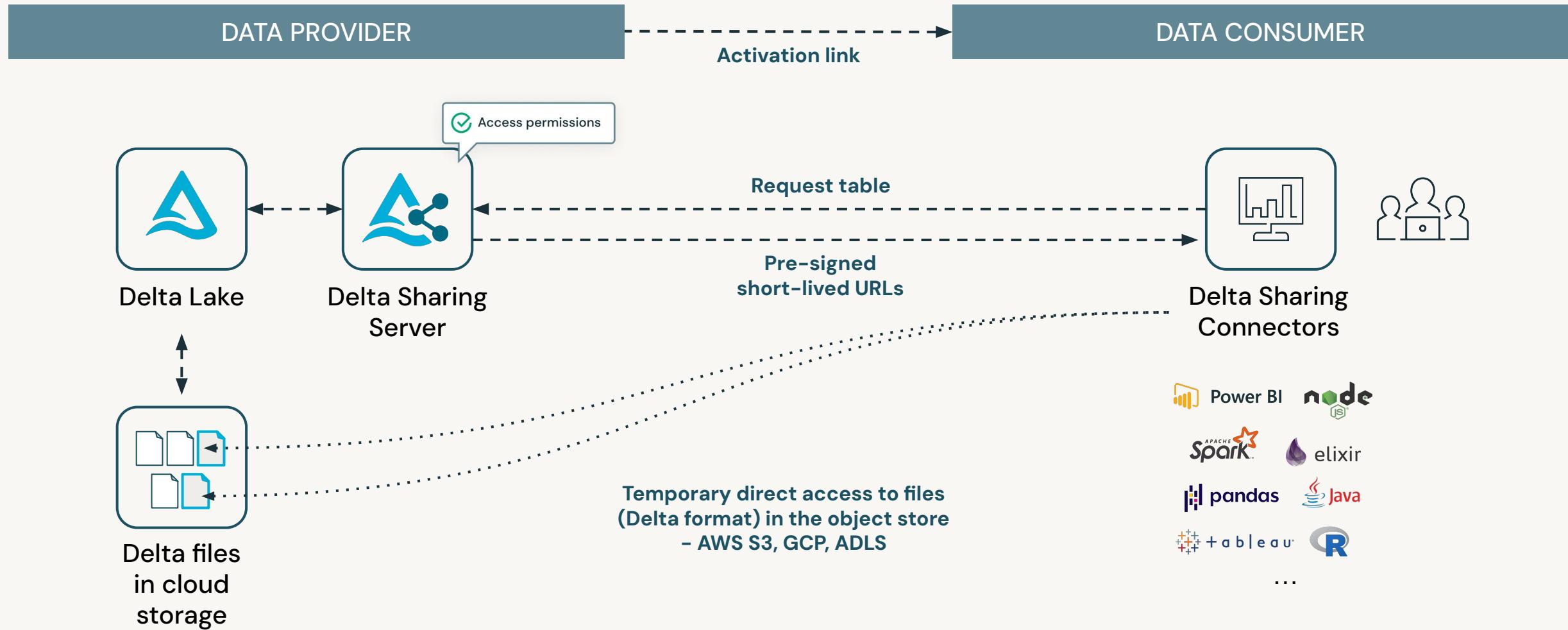
Solution

- Discovery Tags allow you to tag Column, Table, Schema, Catalog objects in UC
- **Integrated search mechanism in UC allows you to search for objects by tag.**

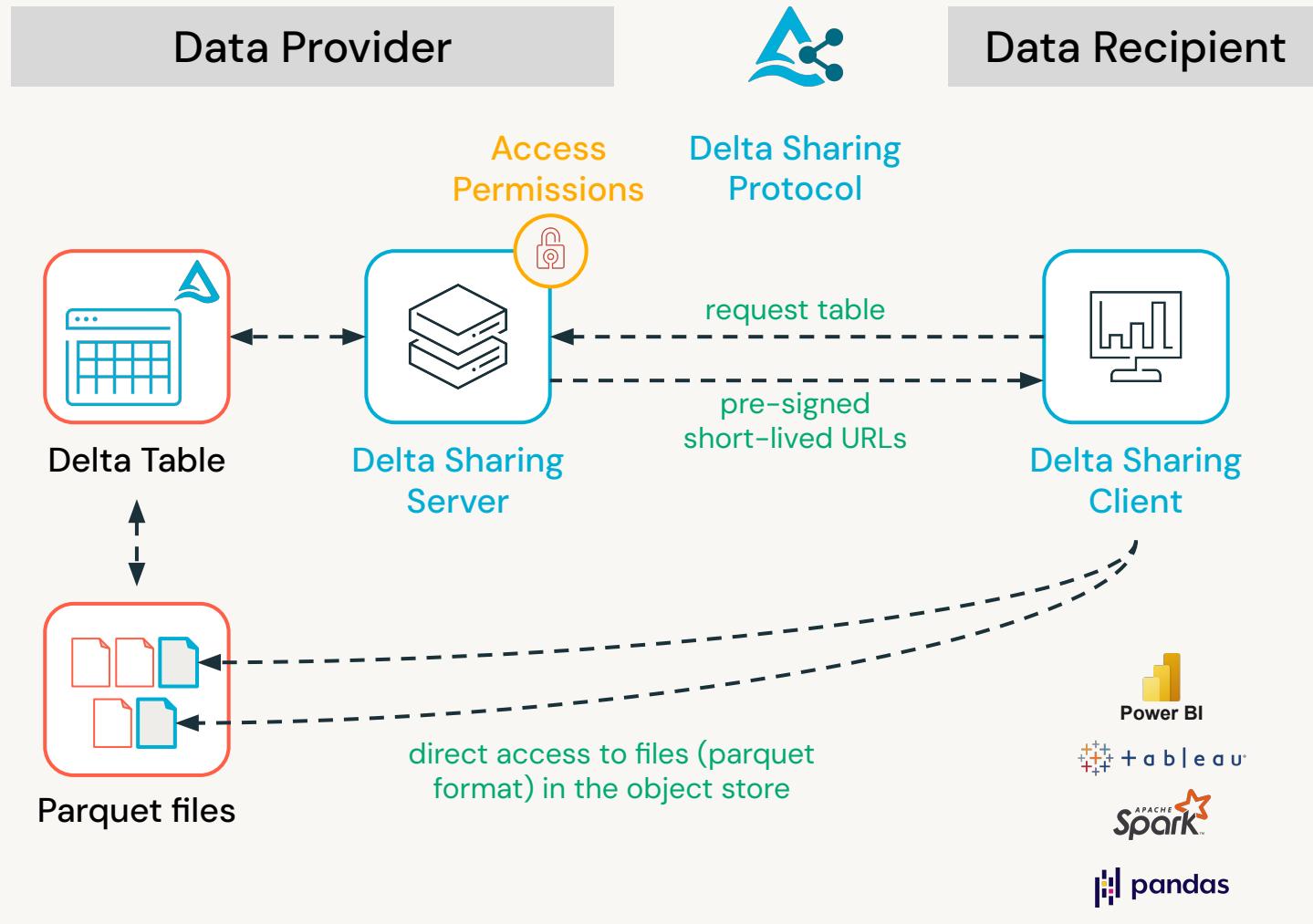
Search based on tags

The screenshot shows the Databricks User Catalog interface. At the top, there's a navigation bar with 'Catalogs > zp_catalog > zp_schema >'. Below it, a table object named 'zp_catalog.zp_schema.tab1' is selected. The 'Columns' tab is active. A modal window titled 'Add/Edit tags for zp_catalog.zp_schema.tab1' contains two tags: 'PII_DATA' and 'SENSITIVE'. Below the modal, the main catalog view shows a search bar with 'Search for tables' and a 'Tables' tab selected. Under the 'Tables' section, there is one result: 'tab1' under 'zp_catalog.zp_schema'. A note at the bottom says, 'Not the results you expected? Try using different keywords, checking for typos, or adjusting filters.' On the right side of the interface, there are 'Cancel' and 'Save' buttons.

Delta Sharing



Delta Sharing - Under the hood



Delta Sharing Protocol:

- Client authenticates to Sharing Server
- Client requests a table (including filters)
- Server checks access permissions
- Server generates and returns pre-signed short-lived URLs
- Client uses URLs to directly read files from object storage

Notes:

- Sharing happens on Delta part files, supporting full tables, partitions, delta versions, ...
- Client is system independent, just needs to be able to read parquet files
- In Databricks Sharing Server and ACL checks are integrated with Unity Catalog





Thank you