databricks

Security and Governance

for data and AI

Gopala Raju

Governance Product Specialist - APJ

- ~3 years with Databricks
- Many years of experience in cloud, data and analytics

Agenda

- Data and AI security overview
- Key considerations and resources available
- Data Governance and Introduction to Unity Catalog
- Key concepts and capabilities
  - Centralised metadata and access controls
  - Query federation & Volumes
  - Discover your data with search and lineage
  - Audit your data
- Sharing and Collaboration
- Upgrading to Unity Catalog
- Demo

Housekeeping

- This presentation is being recorded, we will share the recording and other materials after the session, within 48 hours
- There is no hands-on component 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

Overview of Data and AI Security

Data Security

Data Security is a set of practices and procedures

Secure Data

- Unauthorized access and Theft
- Corruption, Poisoning or Accidental loss to preserve data privacy
- Confidentiality, Integrity and Availability

Protect Sensitive Data

- Personally Identifiable Information
- Financial Information
- Health Information
- Intellectual Property

Encompass Everything

- Physical security of hardware and storage devices
- Administrative and access controls
- Security of software applications
- Data governance policies

Why is data security important?

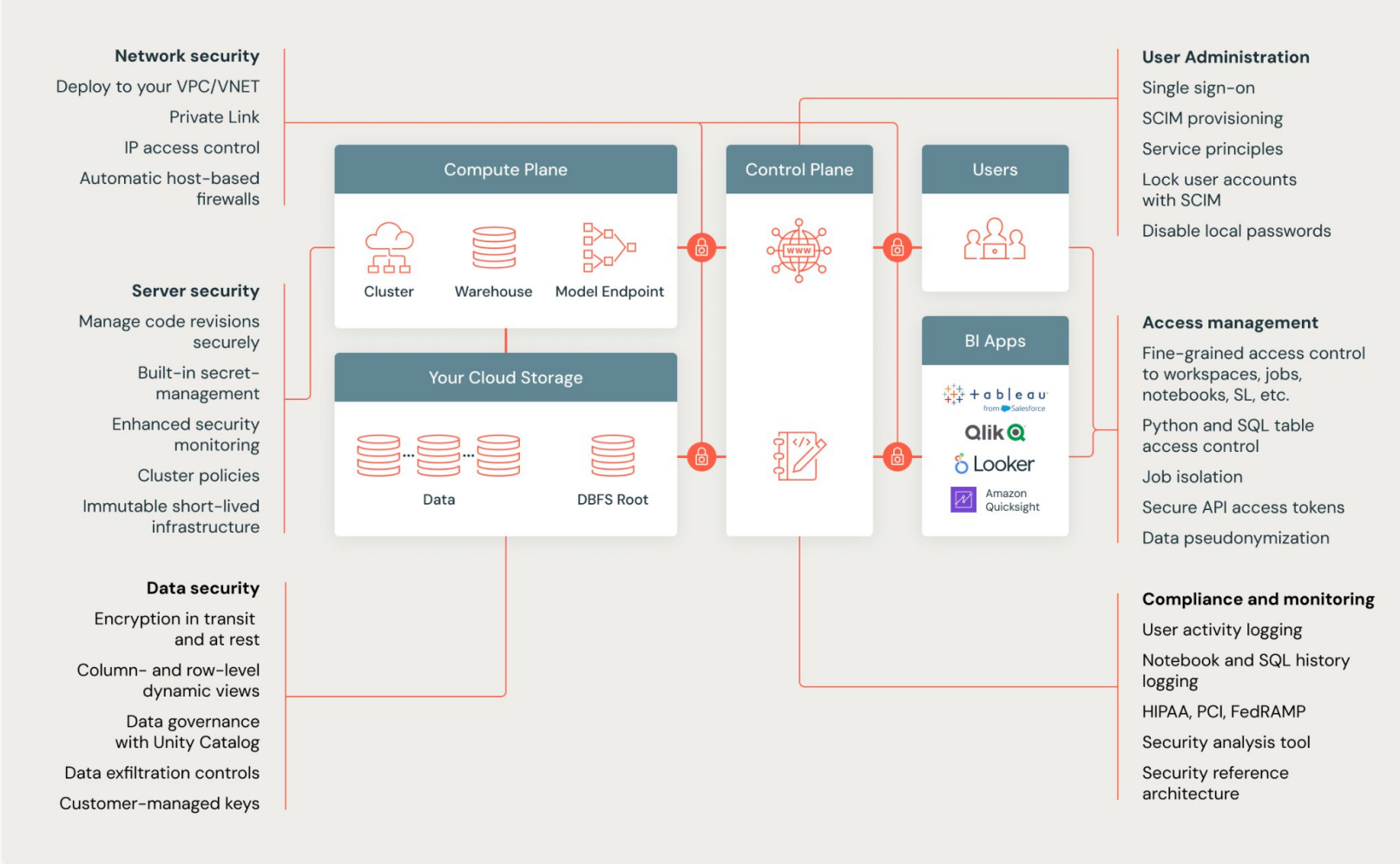
Data is the most critical asset today, When it is protected, governed and maintained it increases in value

Data loss or misuse can have severe consequences like reputational and financial damage

Databricks security overview

Data security is implemented at various levels

It's a shared responsibility



Security Best Practices at Databricks

Making security easy to implement

Best Practice Whitepaper

Cloud-specific checklists and documentation, based on the threats faced by your organization

Security Analysis Tool

Automated comparisons of your deployed security configurations against best practices

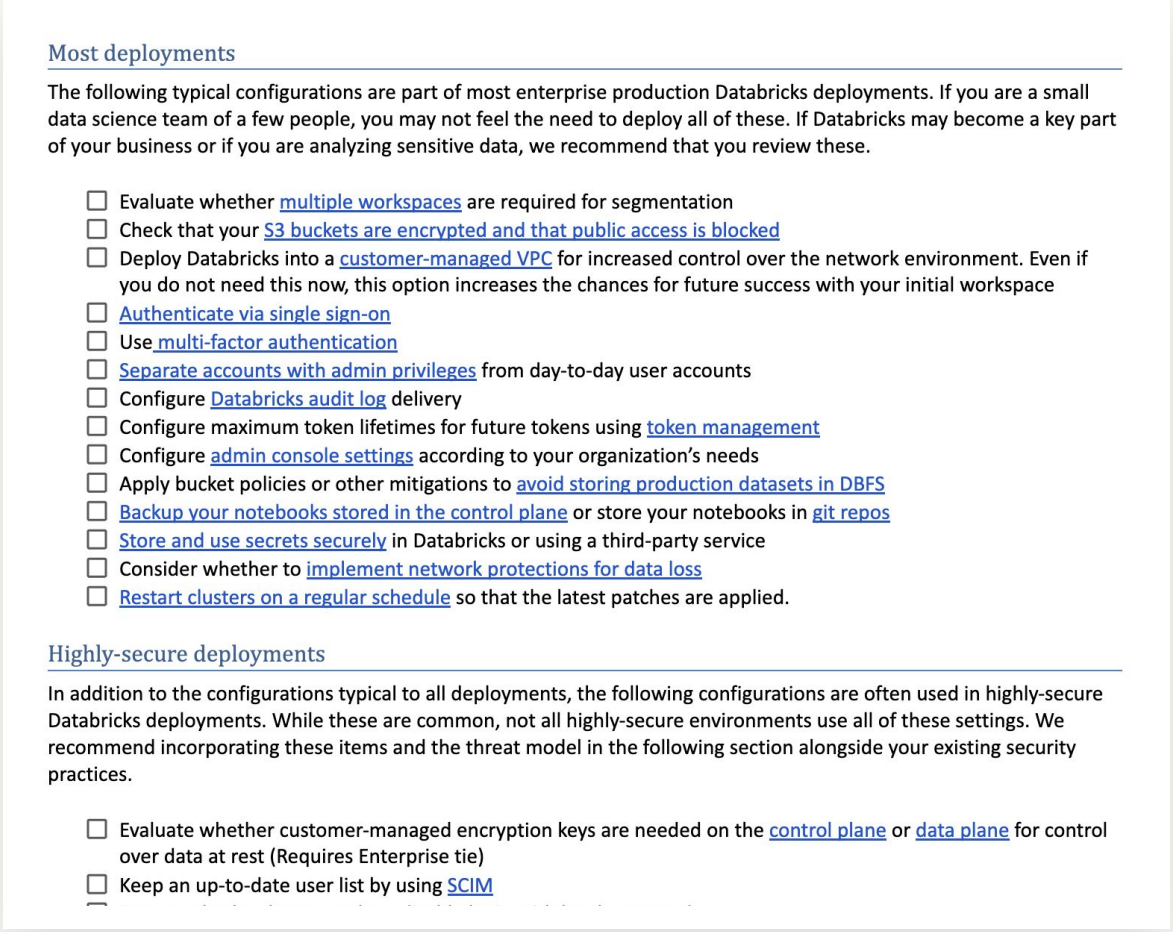
Terraform Templates

Integrate with your infrastructure automation tools

Define: Security Best Practices White Papers

Harden your workspaces with our list of best practices

- Provides best practices collected from our most security-conscious deployments
- Both Databricks configurations and related services like AWS, SSO
- Includes general guidance along with configurations to address specific threats
- Available on databricks.com/trust



Most deployments

The following typical configurations are part of most enterprise production Databricks deployments. If you are a small data science team of a few people, you may not feel the need to deploy all of these. If Databricks may become a key part of your business or if you are analyzing sensitive data, we recommend that you review these.

- ☐ Evaluate whether multiple workspaces are required for segmentation
- ☐ Check that your S3 buckets are encrypted and that public access is blocked
- ☐ Deploy Databricks into a customer-managed VPC for increased control over the network environment. Even if you do not need this now, this option increases the chances for future success with your initial workspace
- ☐ Authenticate via single sign-on
- ☐ Use multi-factor authentication
- ☐ Separate accounts with admin privileges from day-to-day user accounts
- ☐ Configure Databricks audit log delivery
- ☐ Configure maximum token lifetimes for future tokens using token management
- ☐ Configure admin console settings according to your organization's needs
- ☐ Apply bucket policies or other mitigations to avoid storing production datasets in DBFS
- ☐ Backup your notebooks stored in the control plane or store your notebooks in git repos
- ☐ Store and use secrets securely in Databricks or using a third-party service
- ☐ Consider whether to implement network protections for data loss
- ☐ Restart clusters on a regular schedule so that the latest patches are applied.

Highly-secure deployments

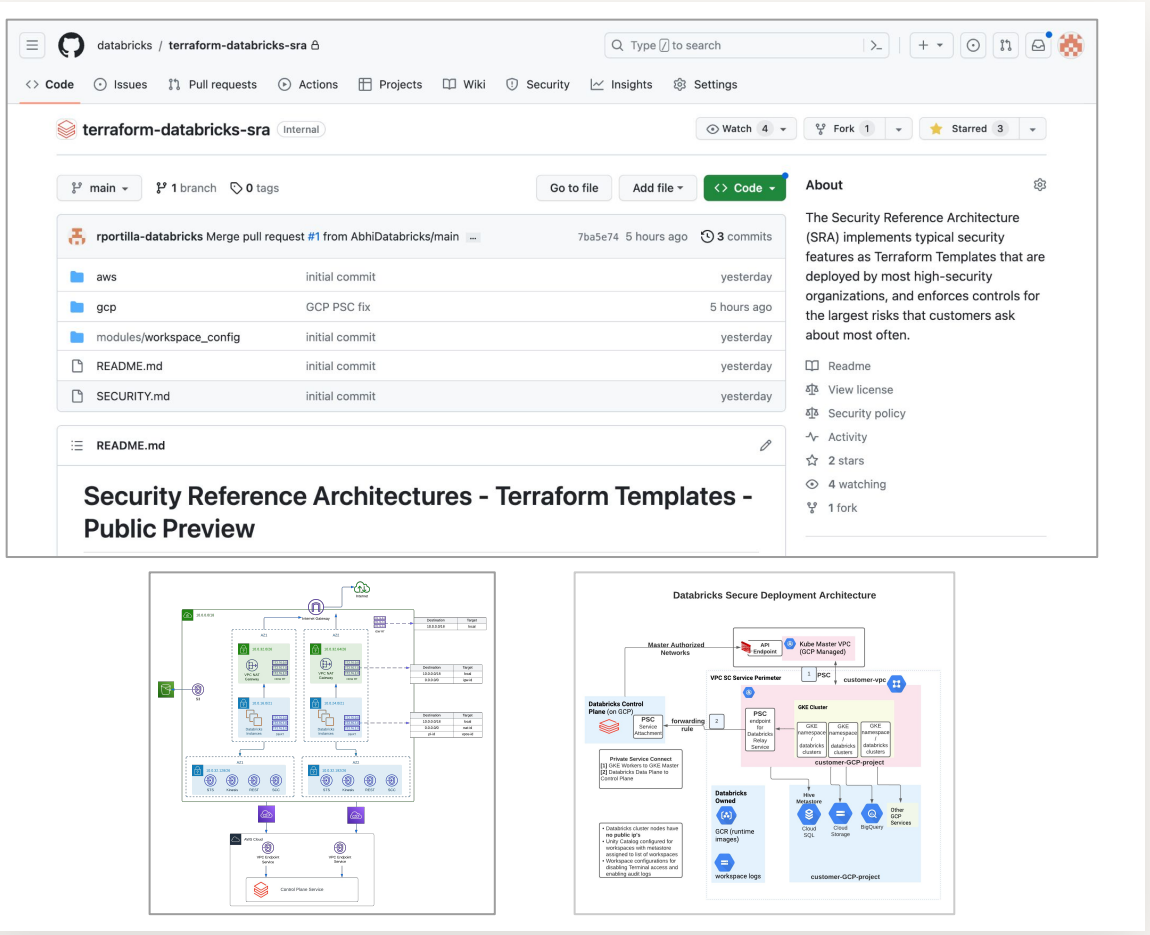
In addition to the configurations typical to all deployments, the following configurations are often used in highly-secure Databricks deployments. While these are common, not all highly-secure environments use all of these settings. We recommend incorporating these items and the threat model in the following section alongside your existing security practices.

- ☐ Evaluate whether customer-managed encryption keys are needed on the control plane or data plane for control over data at rest (Requires Enterprise tie)
- ☐ Keep an up-to-date user list by using SCIM

Deploy: Terraform Templates

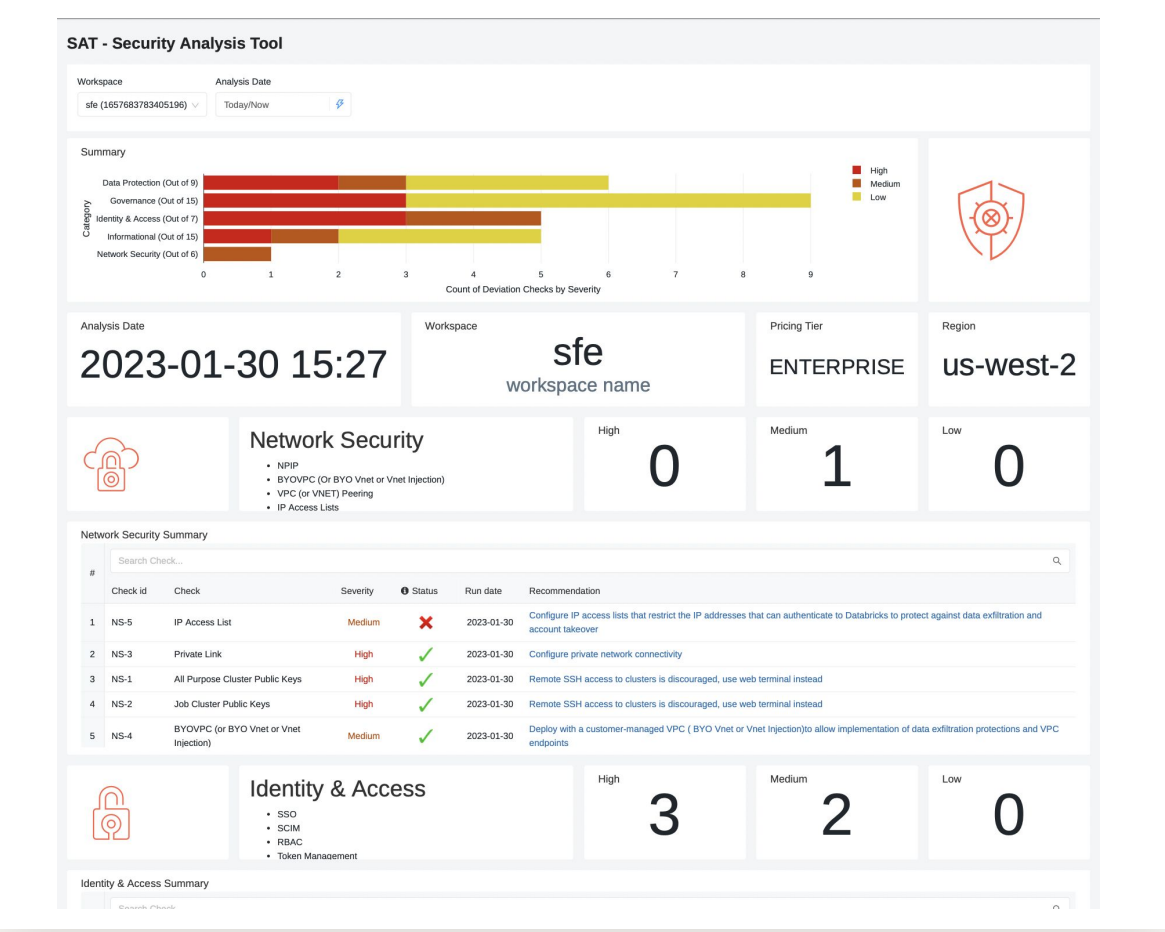
Run secure-by-default deployments via infrastructure-as-code

- The most successful Databricks deployments are often managed using Terraform
- Begin with a secure template to default secure configurations
- Initial templates based on Databricks Security Best Practices
- Customer feedback/input is collected through git issues and pull requests and iterated



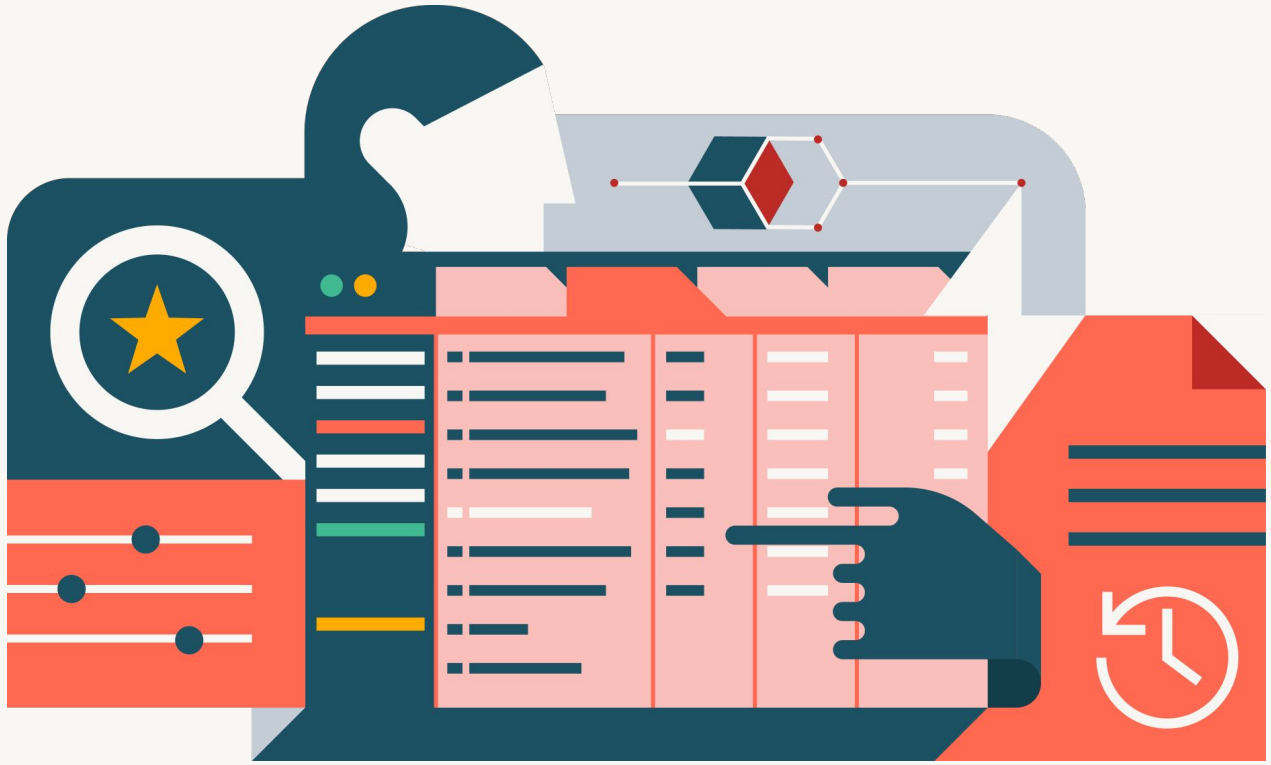
Monitor: Security Analysis Tool

Monitor the security health of your account workspaces over time



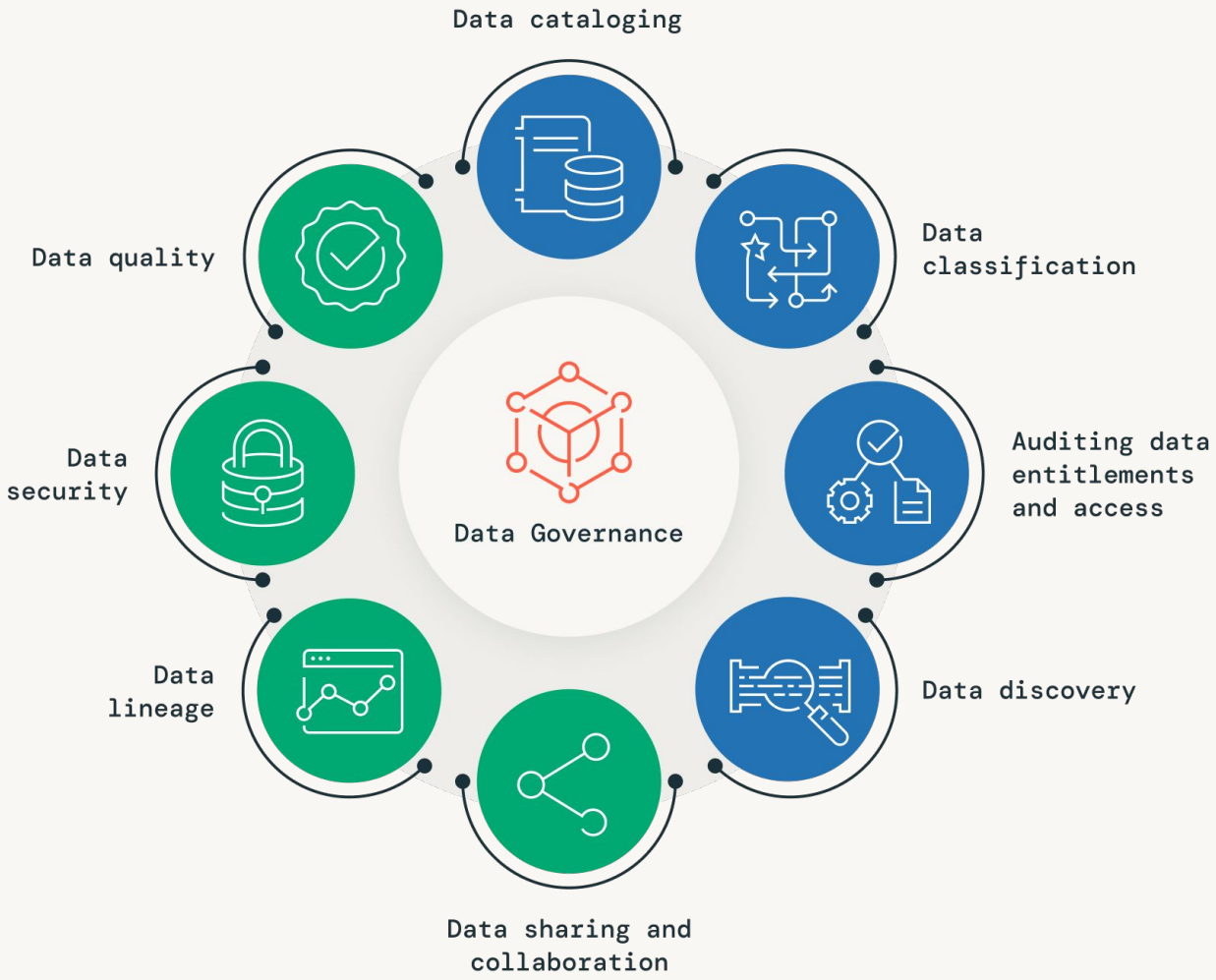
- Compare workspace configurations against specific best practices
- Flag deviations for your workspaces over a period of time
- Easily identify mitigation references
- Available for AWS, Azure and GCP (including Terraform deployments)

Data and AI Security & Governance



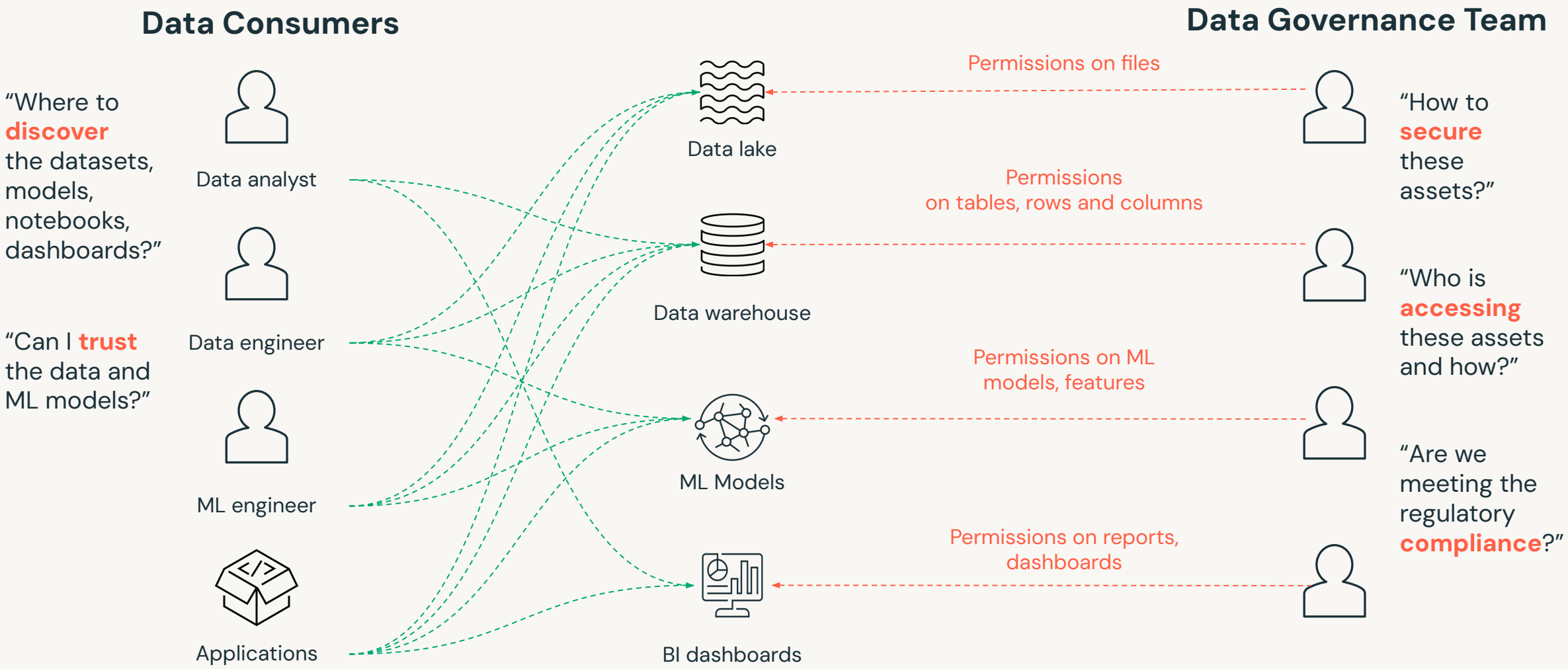
Governance > Security

Governance enhances value of data and AI

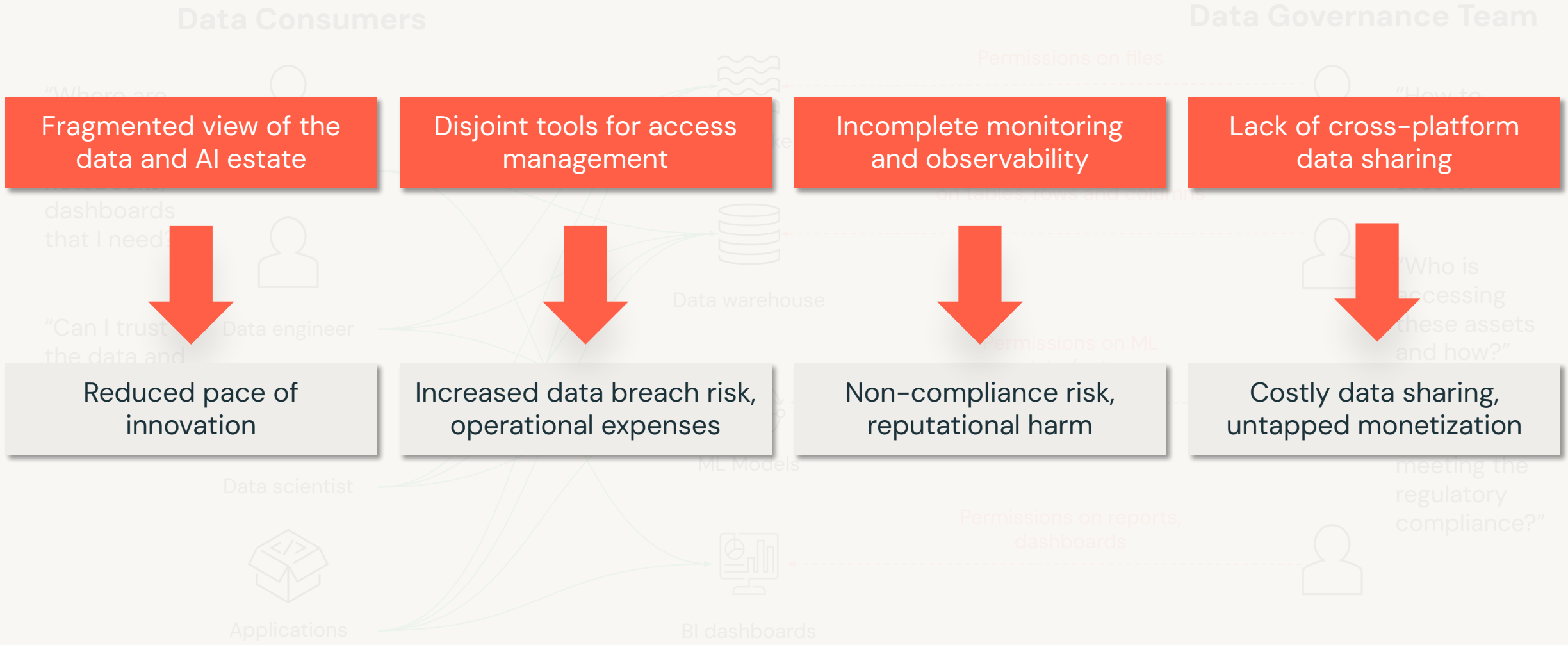


Key dimensions of enterprise data governance

Today, data and AI governance is complex



Today, data and AI governance is complex



Databricks Unity Catalog

Unified governance for data and AI

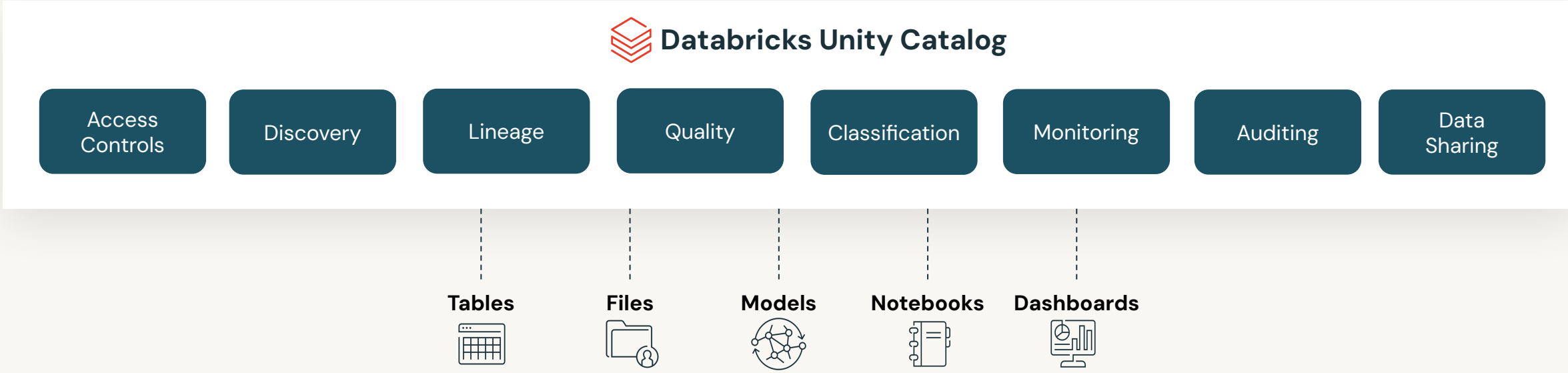
Unified visibility and single permission model for data and AI

AI-powered monitoring and observability

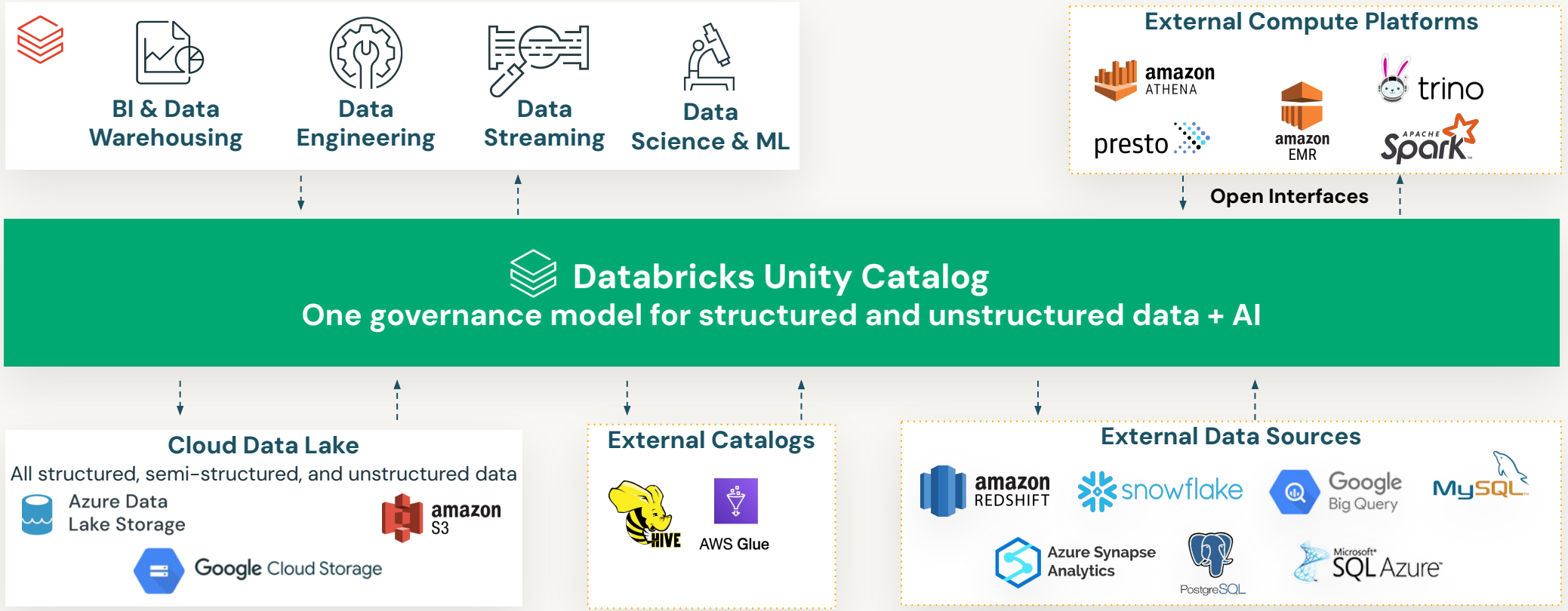
Discovery based on metadata

Open source & Open interfaces

Open data sharing

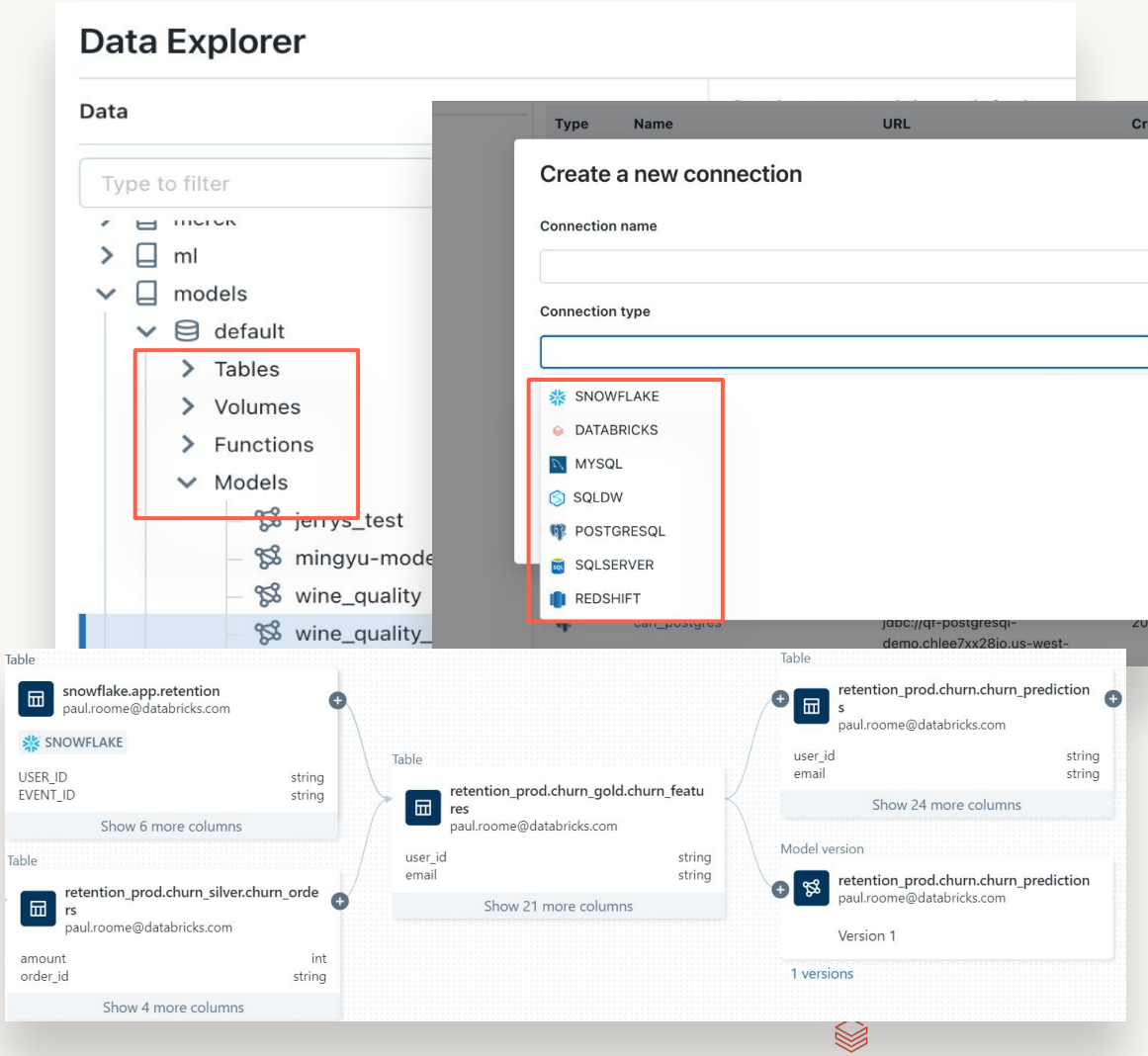


Databricks Lakehouse unifies data and AI governance



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**



Lakehouse Federation

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

Build a unified view of your data estate

Safeguard data across data sources

Efficient execution and caching

Available Soon

Metastore Federation

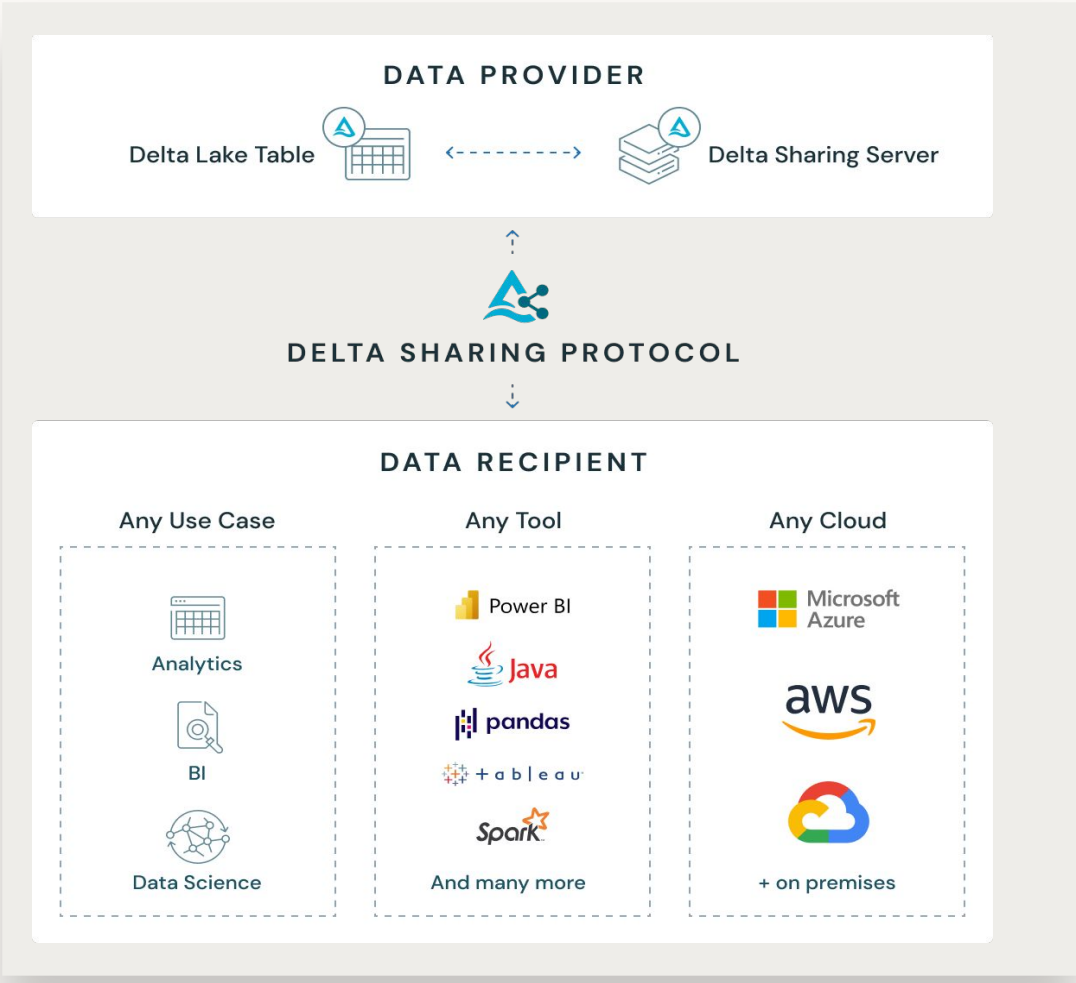
Bring external catalogs to Unity Catalog

Bring external catalogs for seamless transition

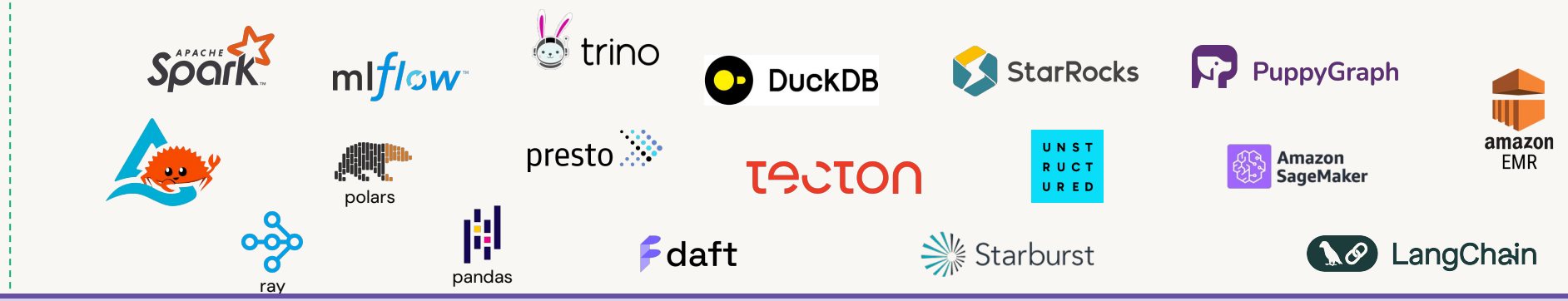
Overlay access controls

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**



OPEN DATA LAKEHOUSE



Goal: any customer asset from Databricks should be accessible from any external engine

Tables	Objects	AI / ML
Delta	Volumes	Functions
Iceberg		ML Models
Hudi		Vector DBs
Views		

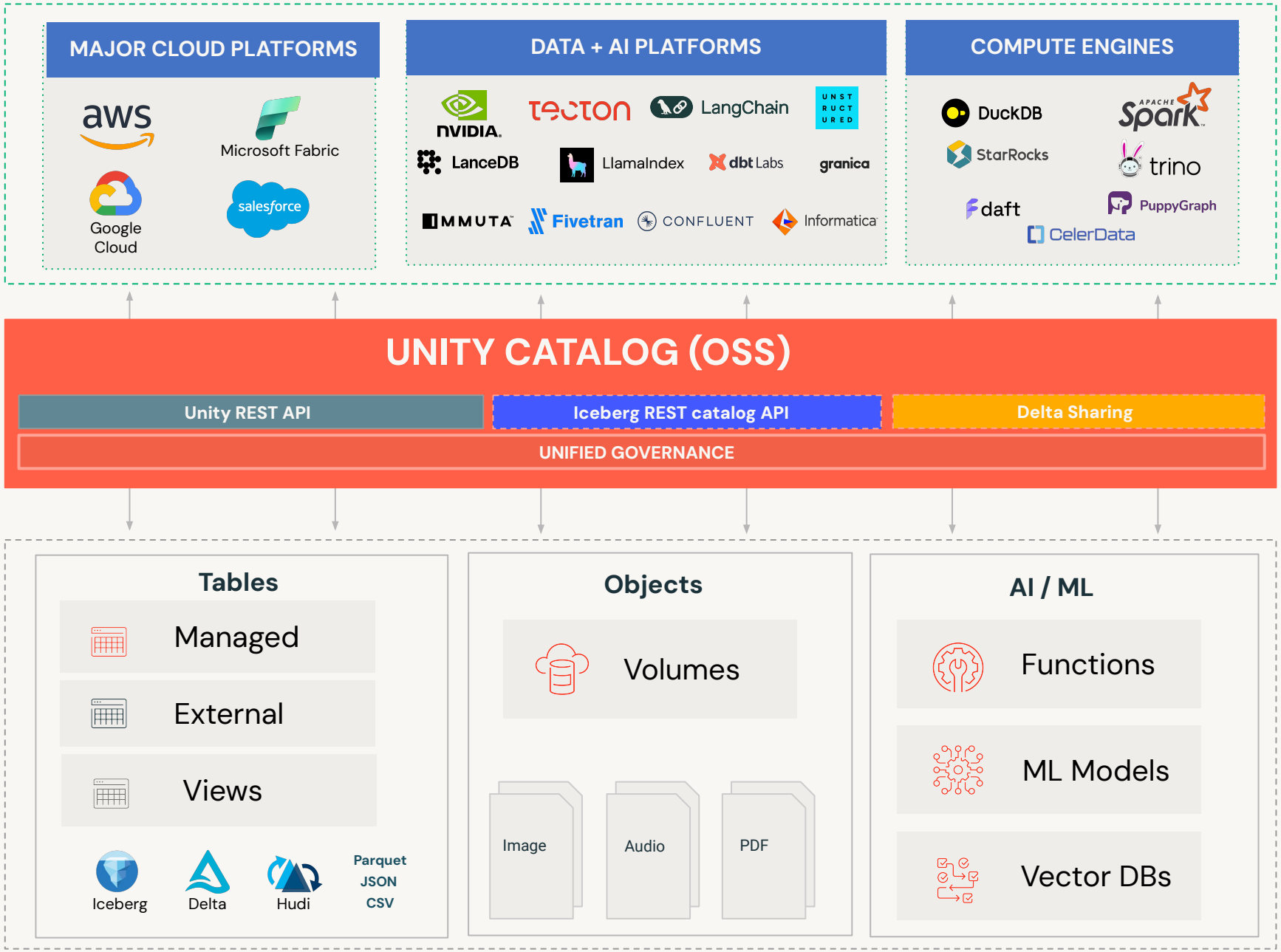
Unity Catalog: The industry's only universal catalog for Data and AI

Any engine
Client ecosystem

Any client
Universal standard

Any asset
Data + AI assets

Any format
UniForm



UC OSS interoperability

LF AI & DATA
SANDBOX PROJECT

{Iceberg REST catalog APIs }

ICEBERG ECOSYSTEM

- Amazon Web Services
- Microsoft Azure
- Google Cloud
- Snowflake
- Salesforce
- Apache Impala
- Clickhouse
- Daft
- Doris
- Dremio
- Flink
- Hive
- Presto
- Apache Spark
- StarRocks
- Trino

Unity Catalog

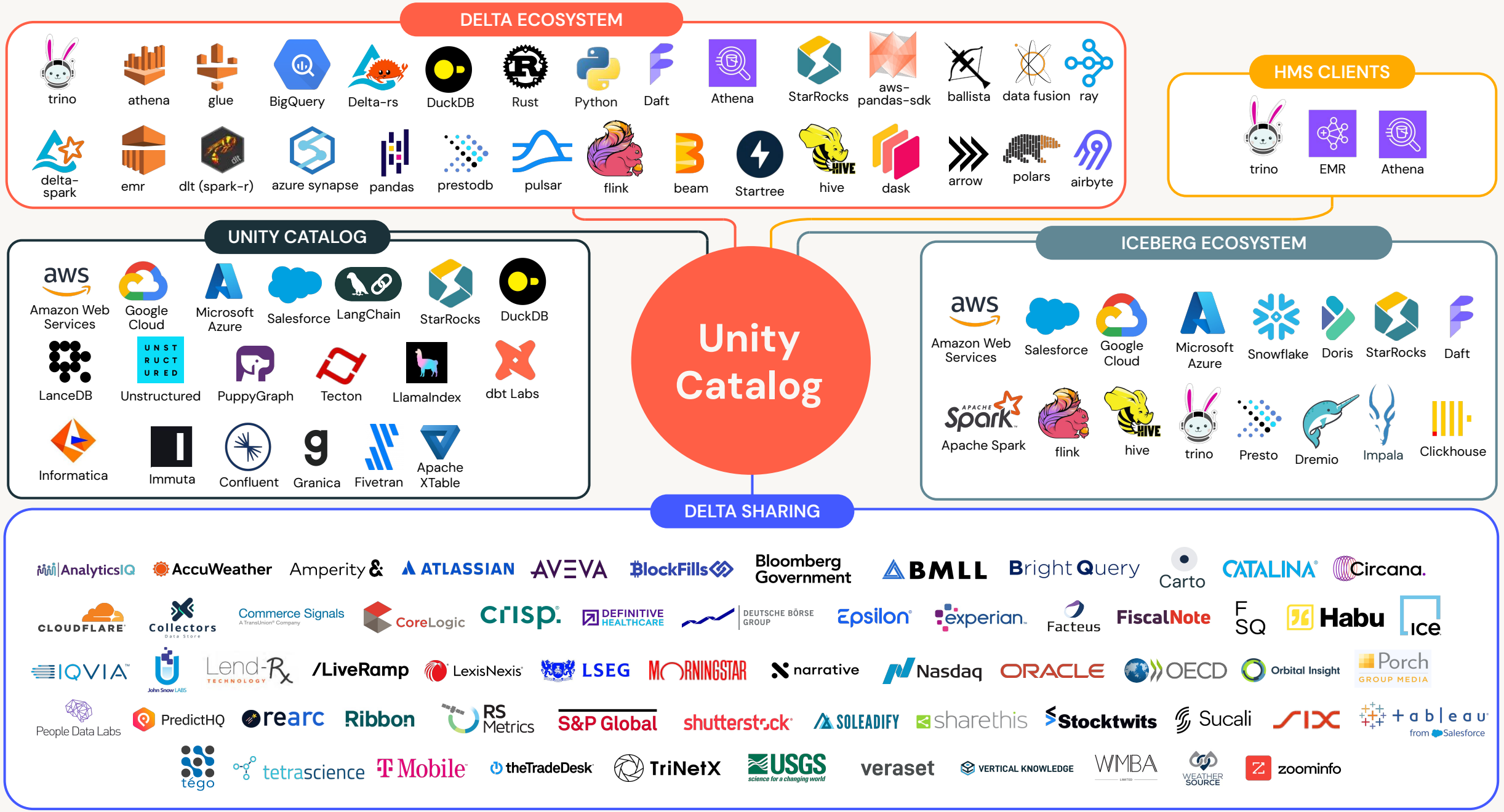
DELTA ECOSYSTEM

- Amazon Web Services
- Azure
- Google Cloud
- Airbyte
- Arrow
- Athena
- Apache Ballista
- aws-pandas-sdk
- beam
- Daft
- Dask
- Apache DataFusion
- Delta-RS
- DLT (spark-r)
- DuckDB
- EMR
- Flink
- Glue
- Hive
- Pandas
- Polars
- PrestoDB
- Pulsar
- PuppyGraph
- Python
- Ray
- Rust
- Apache Spark
- StarRocks
- Startree (pinot)
- Trino

UNITY CATALOG PARTNERS

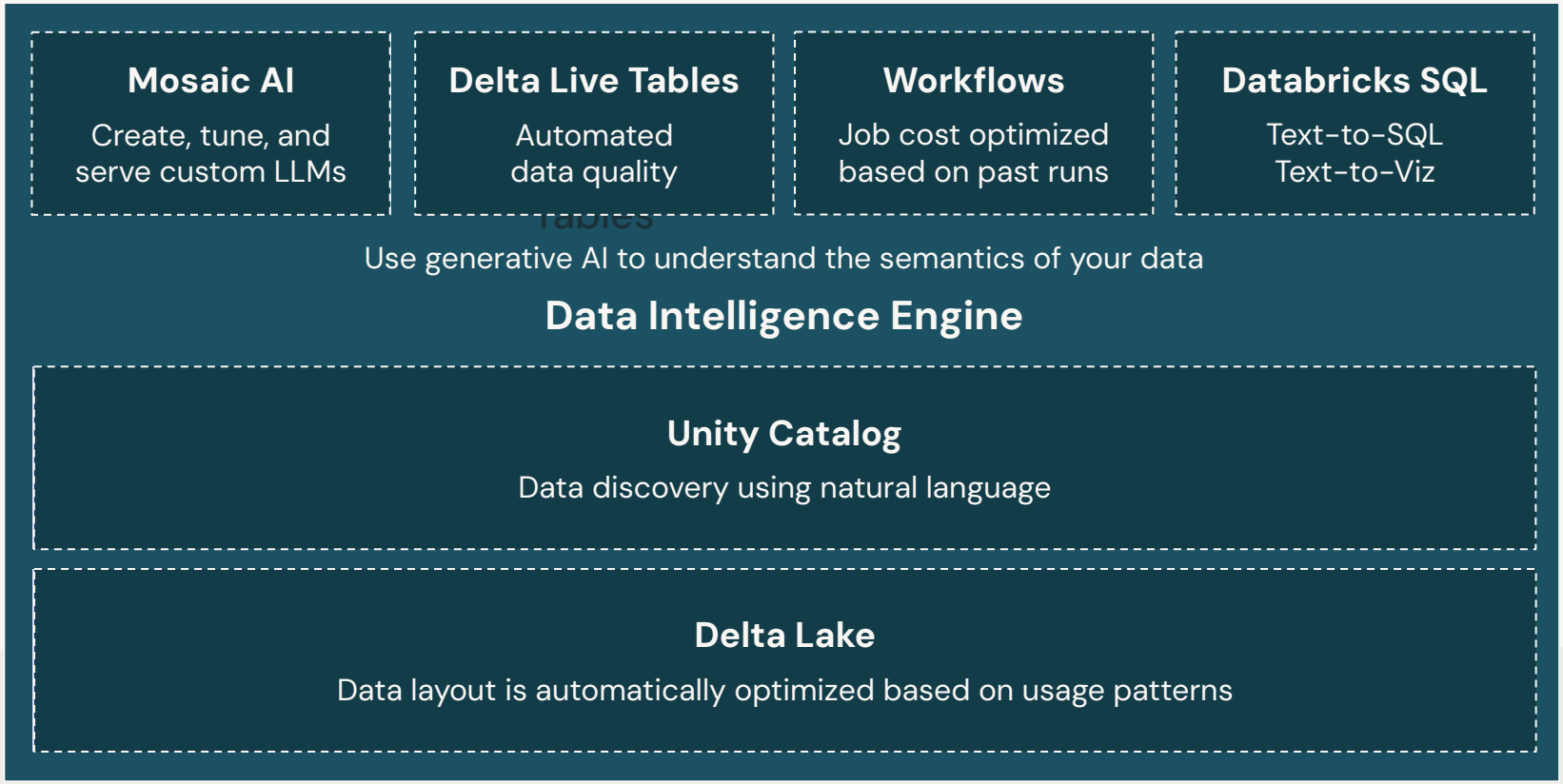
- Amazon Web Services
- Azure
- Google Cloud
- Salesforce
- NVIDIA
- CONFLUENT
- daft
- databricks
- dbt Labs
- DuckDB
- Fivetran
- granica
- IMMUTA
- Informatica
- LanceDB
- LangChain
- LlamaIndex
- ONEHOUSE
- PuppyGraph
- StarRocks
- tecton
- UNSTRUCTURED
- Apache XTable

Most open and interoperable catalog for data and AI



However, it is not just a catalog … in databricks

Unity Catalog Powers Databricks Data Intelligence Platform



Open Data Lake

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

Key concepts and capabilities

Key concepts

Centralised metadata and access controls

Query federation & Volumes

Discover your data with search and lineage

Audit your data

Open collaboration

Key Concepts

Working with file based data sources

- Metastore
  - A high-level entity, one per region, that holds multiple catalogs
- Catalogs
  - Entities to organize multiple schemas together
- 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

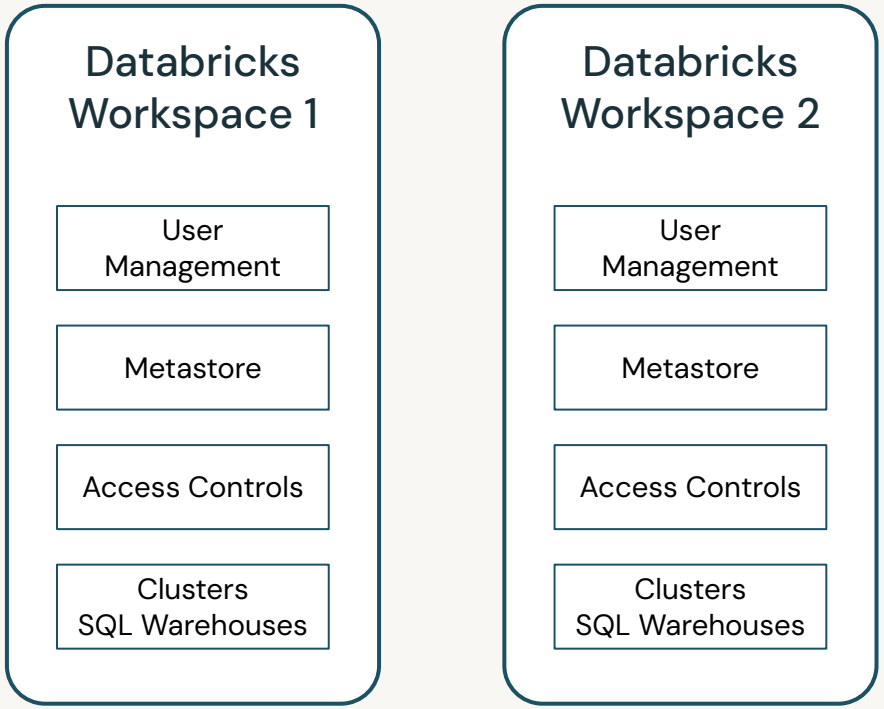
Working with databases

- 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

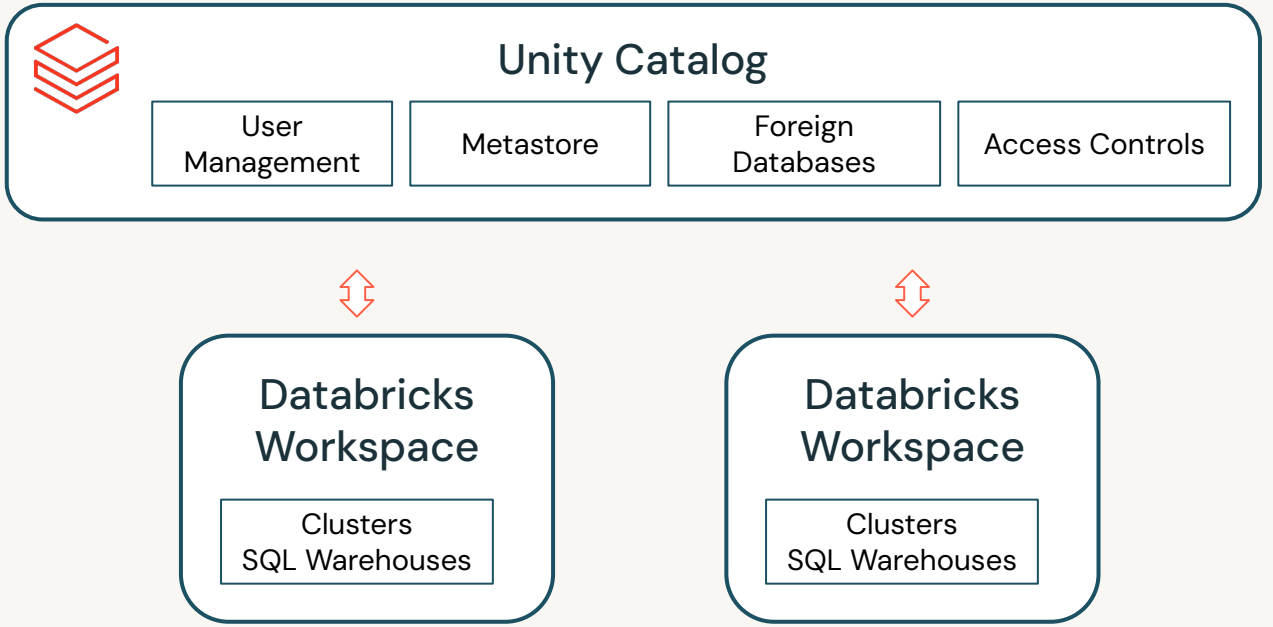
Centralized metadata and controls

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

Without Unity Catalog

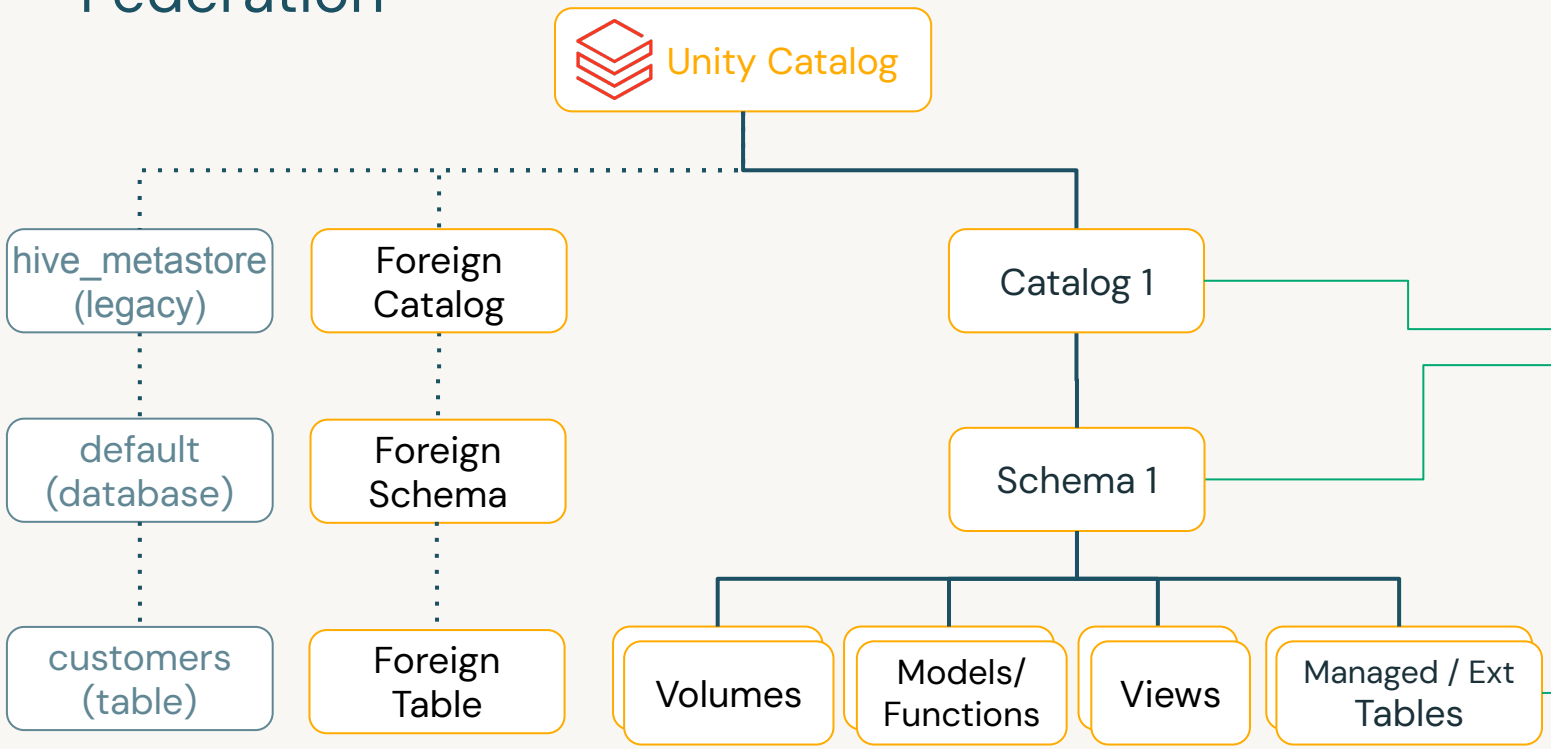


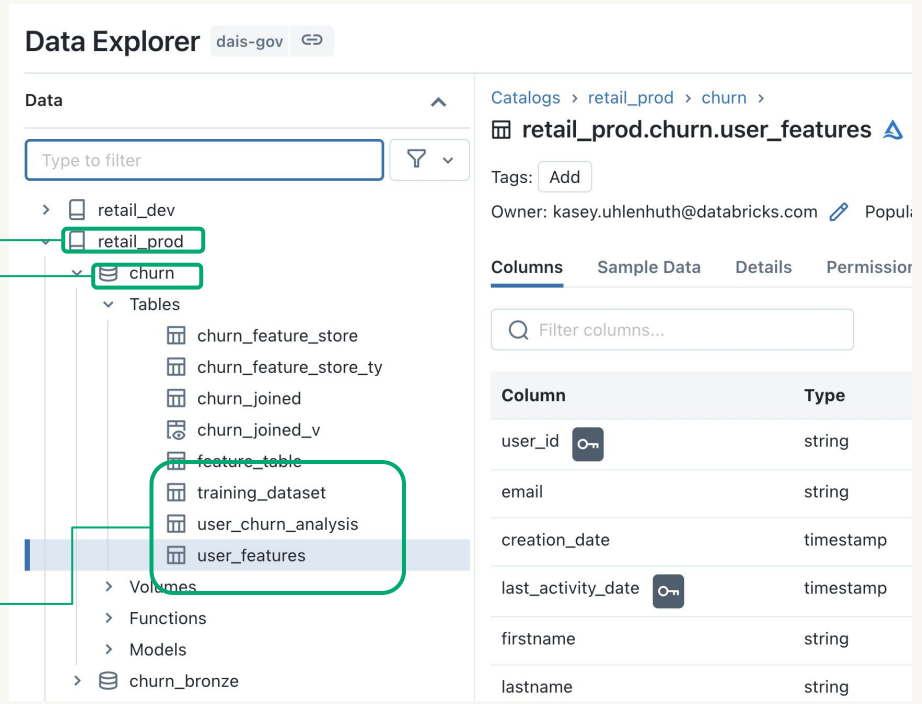
With Unity Catalog



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

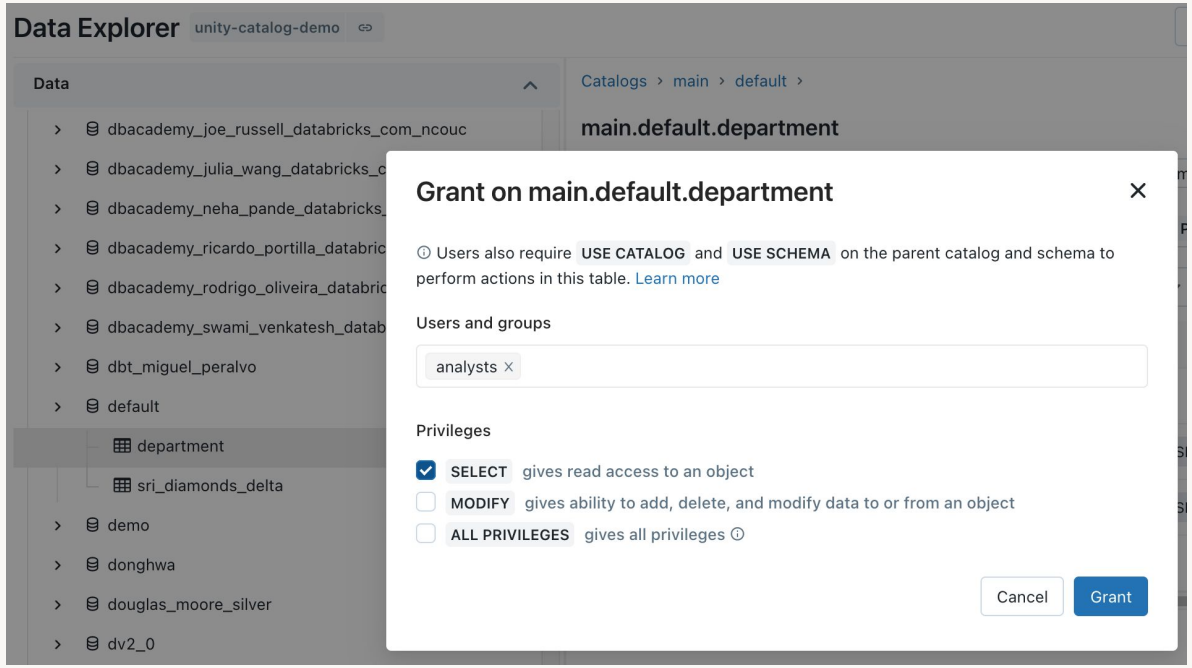
```
GRANT <privilege> ON <securable_type>
<securable_name> TO `<principal>`

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

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;
```

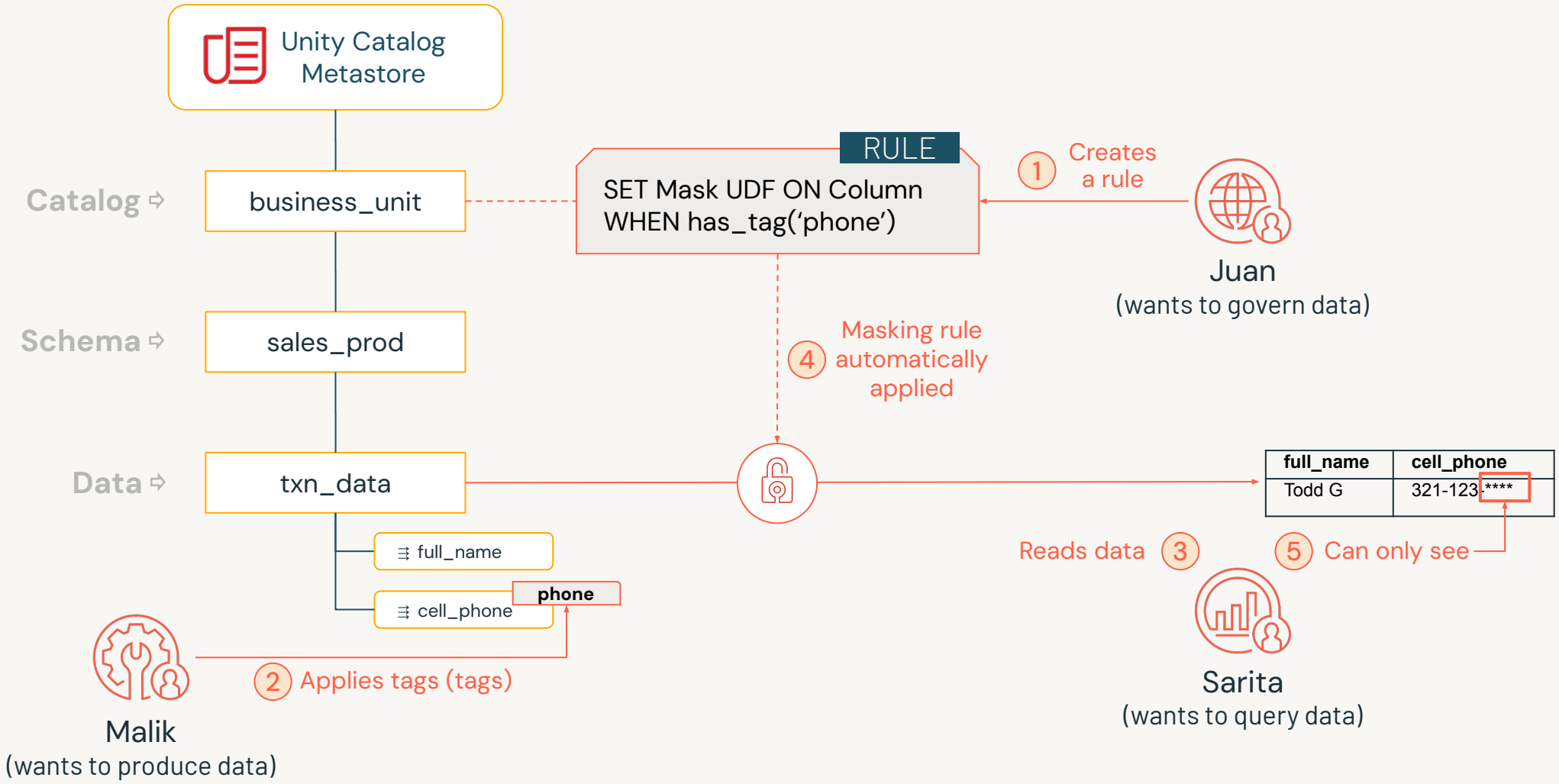
Test for group membership

Assign reusable mask to column

Specify mask or function to mask

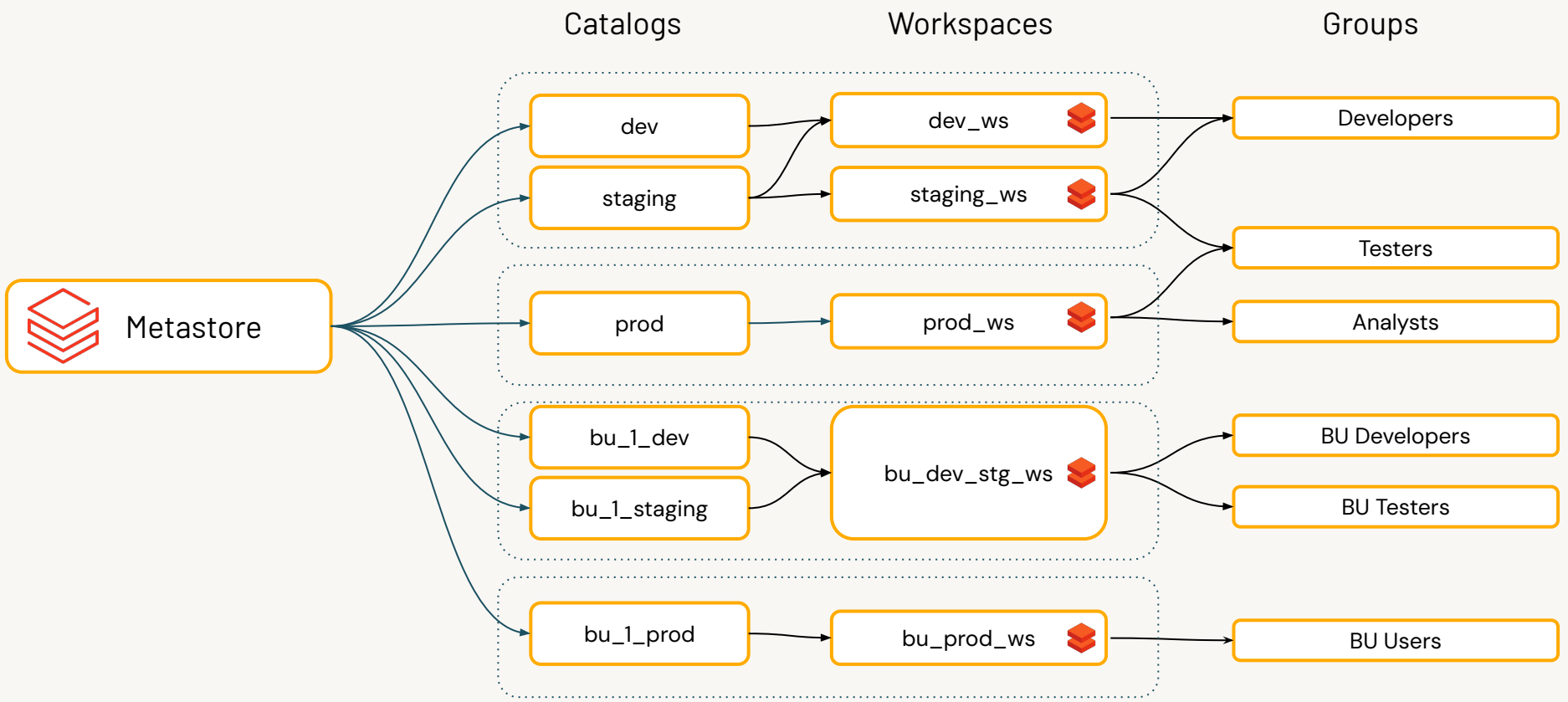
Scalable governance with ABAC

Available Soon



Catalog binding

Restrict catalog access by environment or purpose



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

Key concepts and capabilities

Key concepts

Centralised metadata and access controls

Query federation & Volumes

Discover your data with search and lineage

Audit your data

Open collaboration

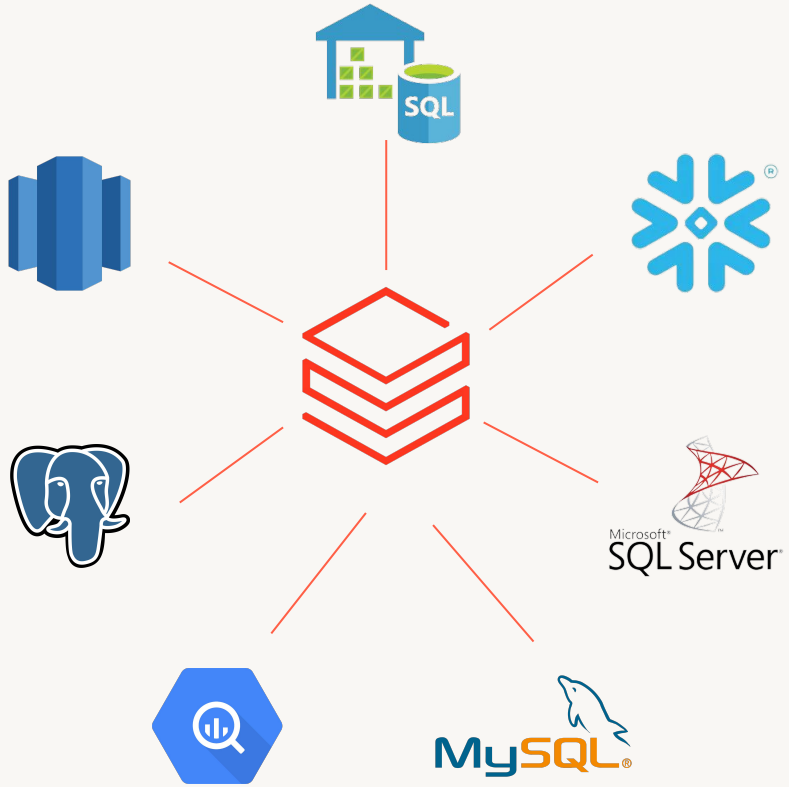
Query Federation

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>
```



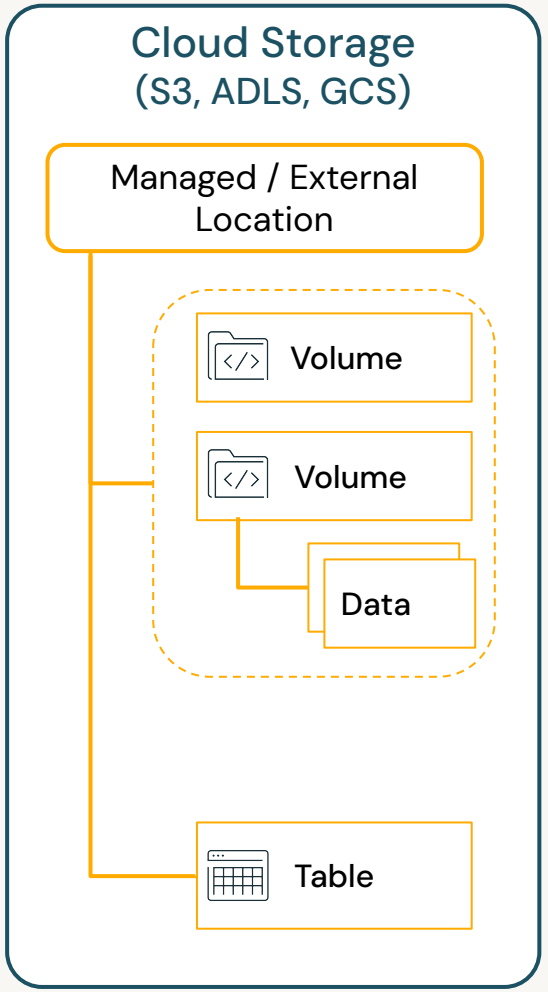
Volumes in Unity Catalog

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

- Volumes can be accessed by some POSIX commands

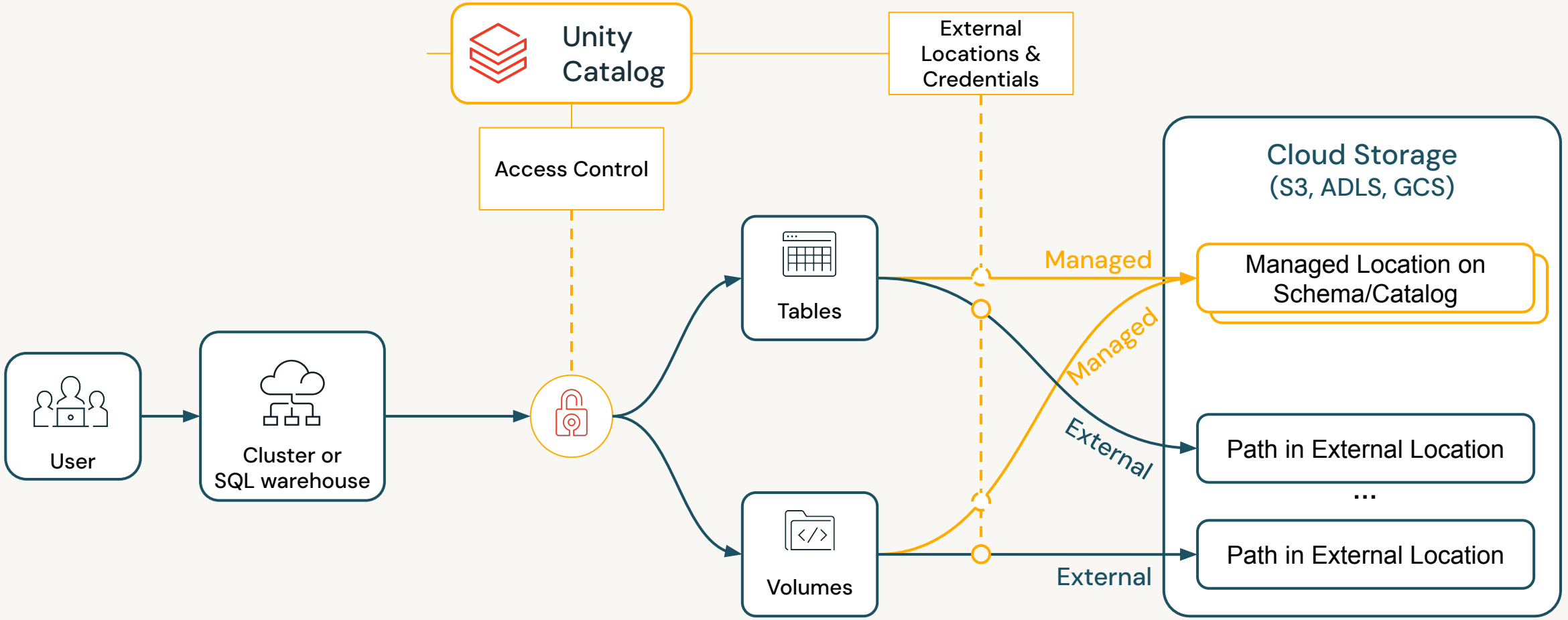
```
dbutils.fs.ls("s3://my_external_location/Volumes/catalog/schema/volume123")

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



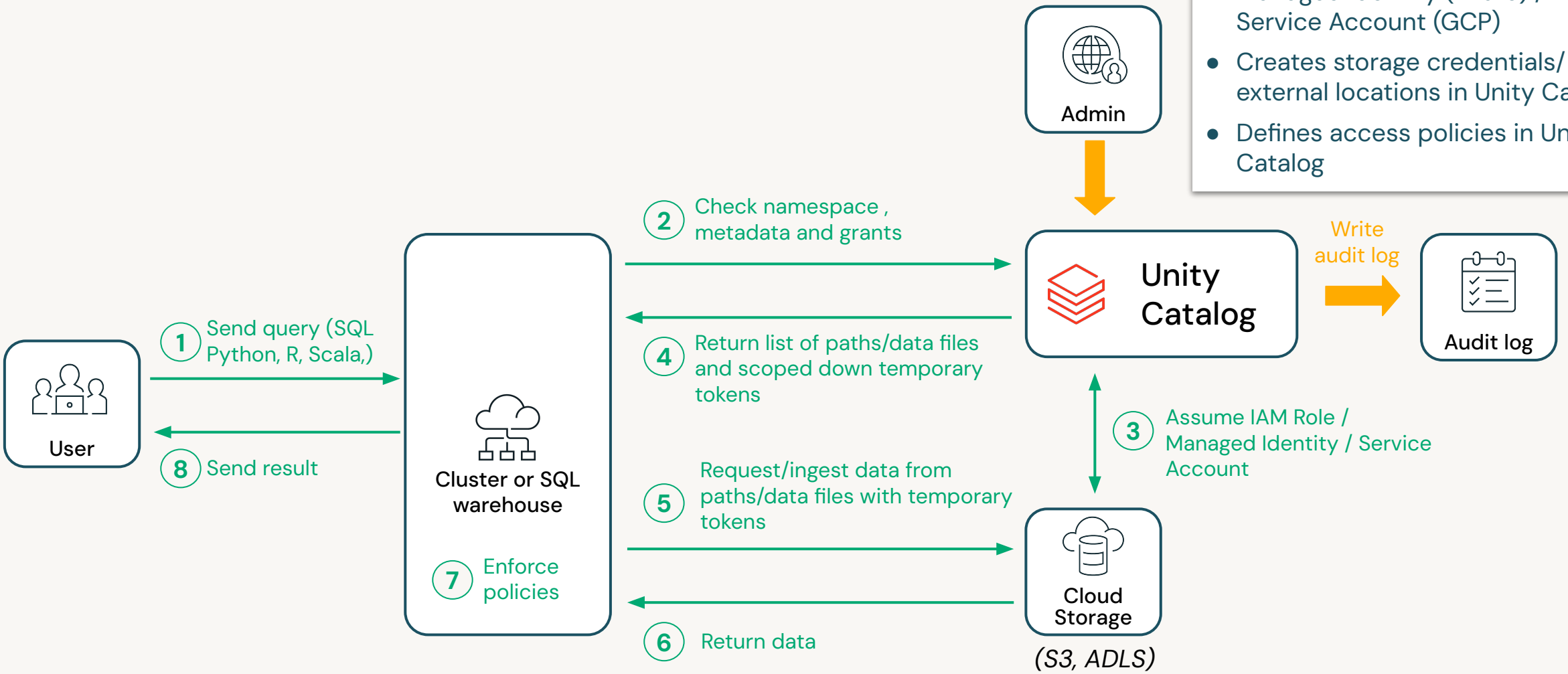
Defining file based data sources in Unity

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



Key concepts and capabilities

Key concepts

Centralised metadata and access controls

Query federation & Volumes

Discover your data with search and lineage

Audit your data

Open collaboration

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

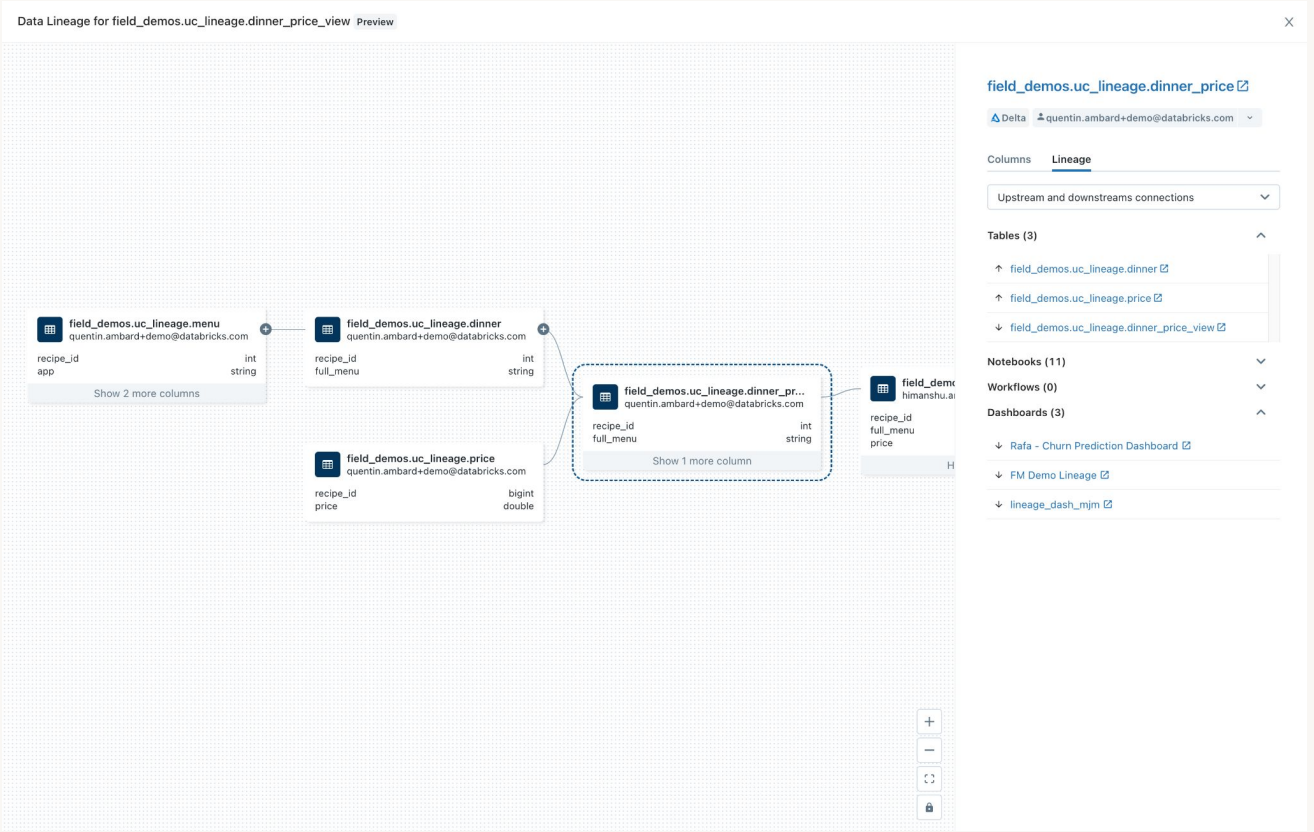
Observability

- 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

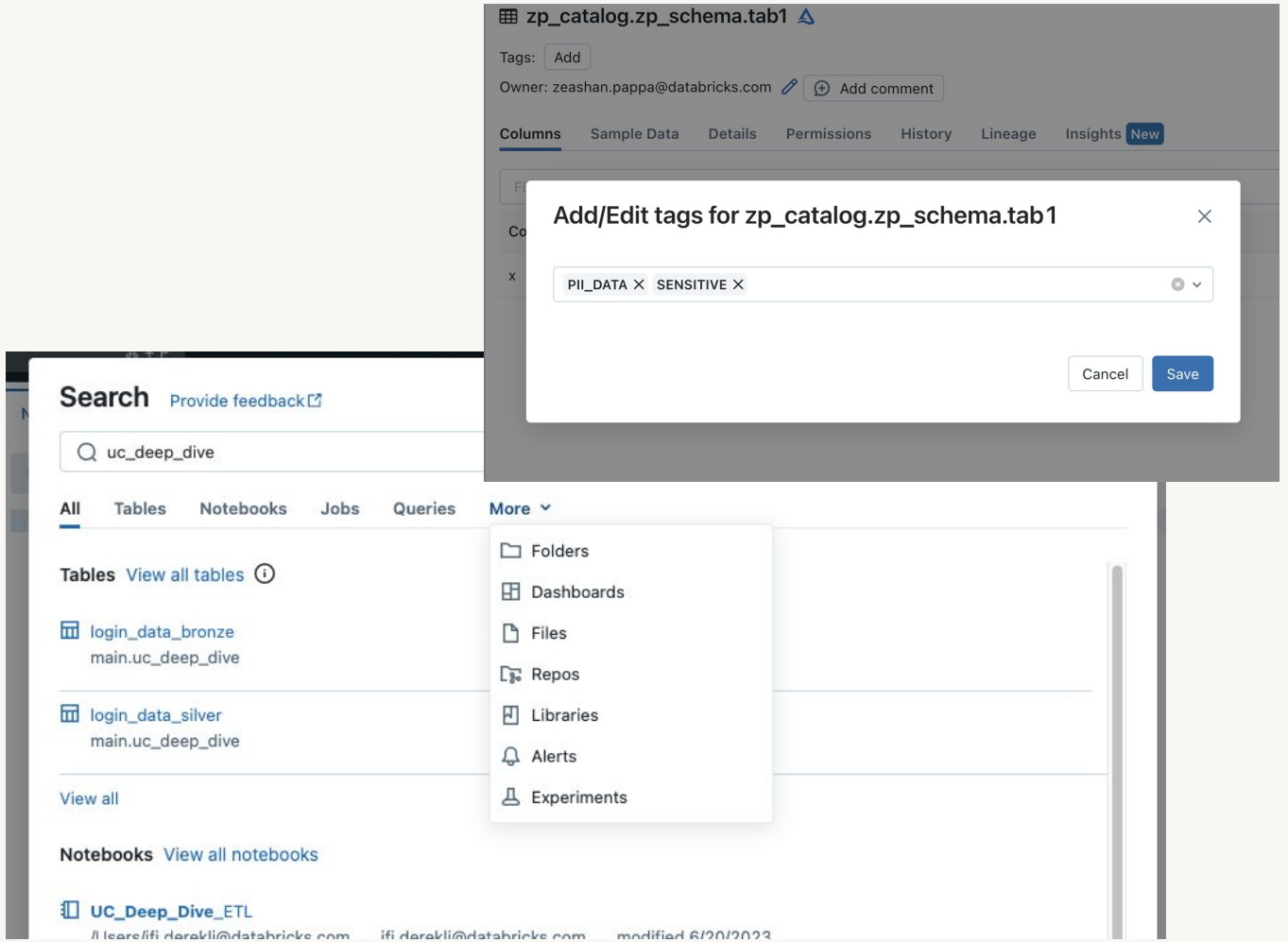
- Auto-capture runtime data lineage on a Databricks cluster or SQL warehouse
- Leverage common permission model from Unity Catalog
- Lineage across tables, columns, dashboards, workflows, notebooks, files, external sources, and models



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

Key concepts and capabilities

Key concepts

Centralised metadata and access controls

Query federation & Volumes

Discover your data with search and lineage

Audit your data

Open collaboration

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

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

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

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";
```

Key concepts and capabilities

Key concepts

Centralised metadata and access controls

Query federation & Volumes

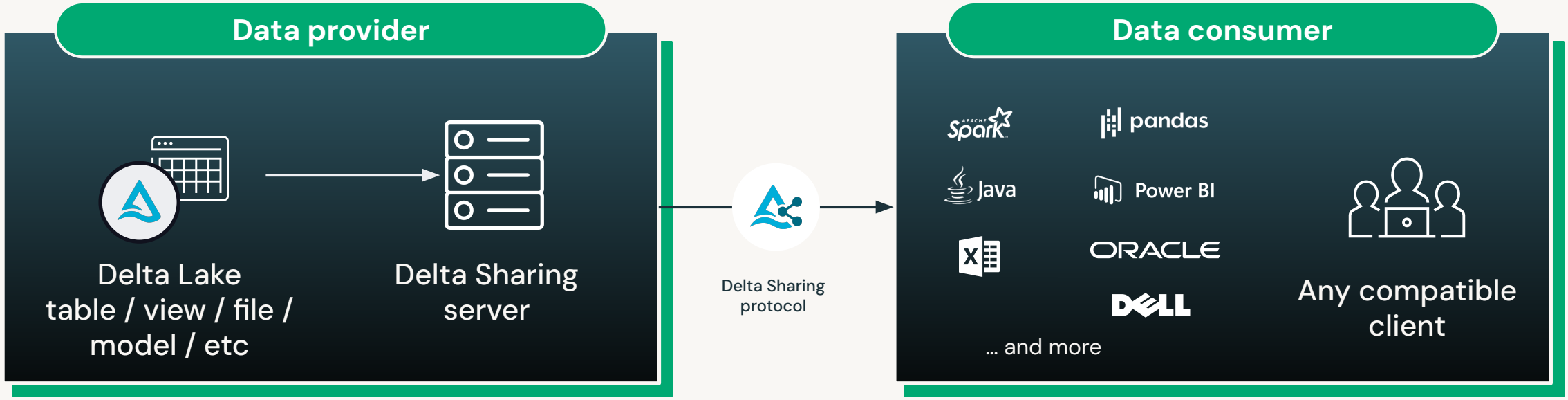
Discover your data with search and lineage

Audit your data

Open collaboration

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

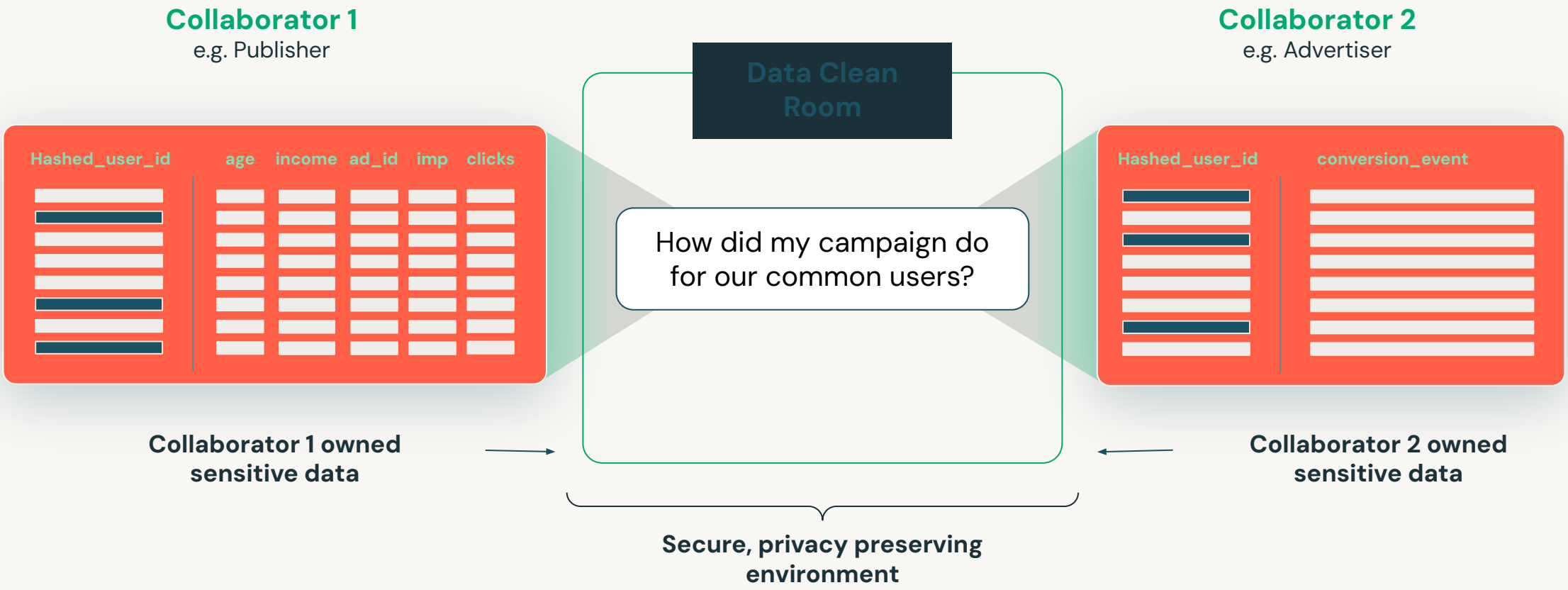
---

Public marketplace

Open for Databricks & non-Databricks users

Clean rooms

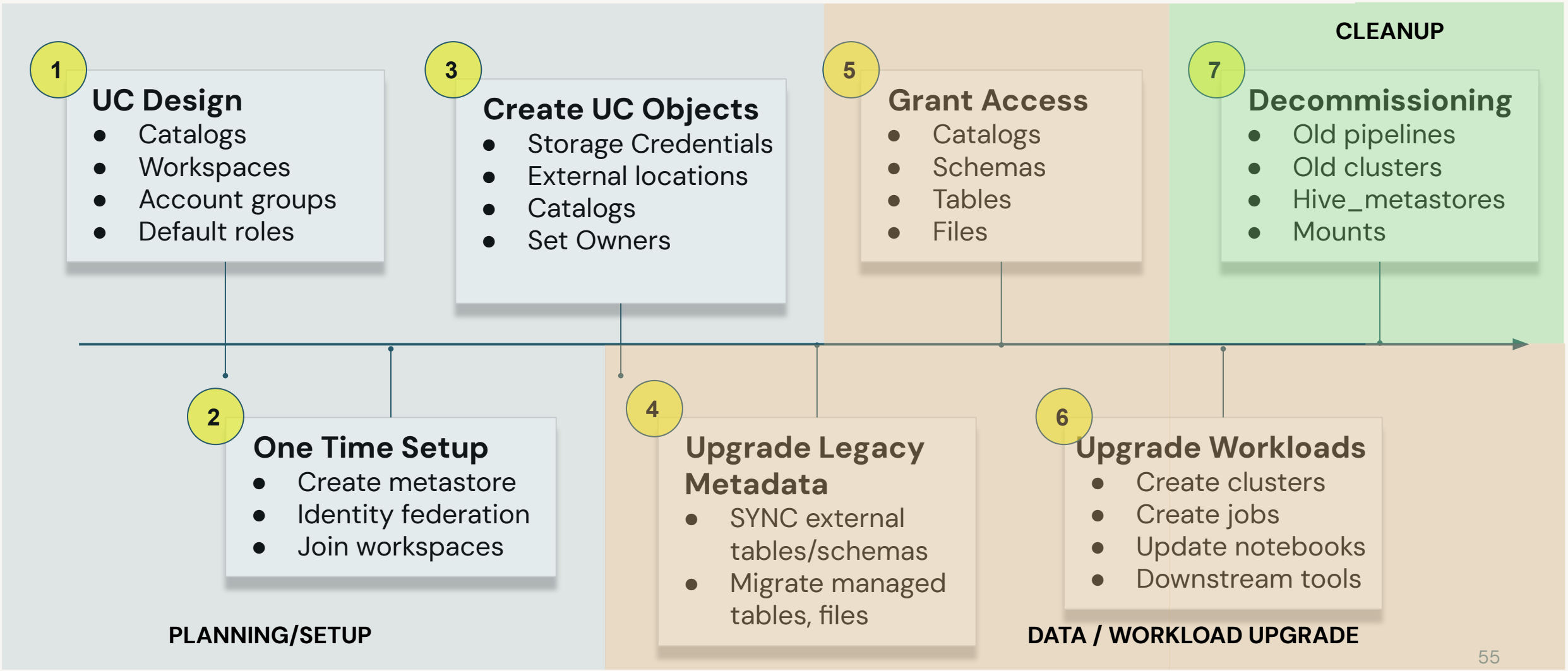
Secure environments to run computations on joint data



Upgrade to Unity Catalog

How to upgrade to Unity Catalog

Steps to consider for a full upgrade



UCX – your best resource for UC upgrades

github.com/databrickslabs/ucx

UCX is a Databricks lab project aimed at streamlining UC Adoption. It is a public source project by Databricks UC field team

Capabilities

- ❑ Assessment
- ❑ Group Upgrade
- ❑ Table Upgrade
- ❑ ACLs
- ❑ Workflows/ Notebook Upgrade (Soon)

Demo

https://www.databricks.com/resources/demos/tutorials/governance/data-lineage-with-unity-catalog?itm_data=demo_center

Thank you!