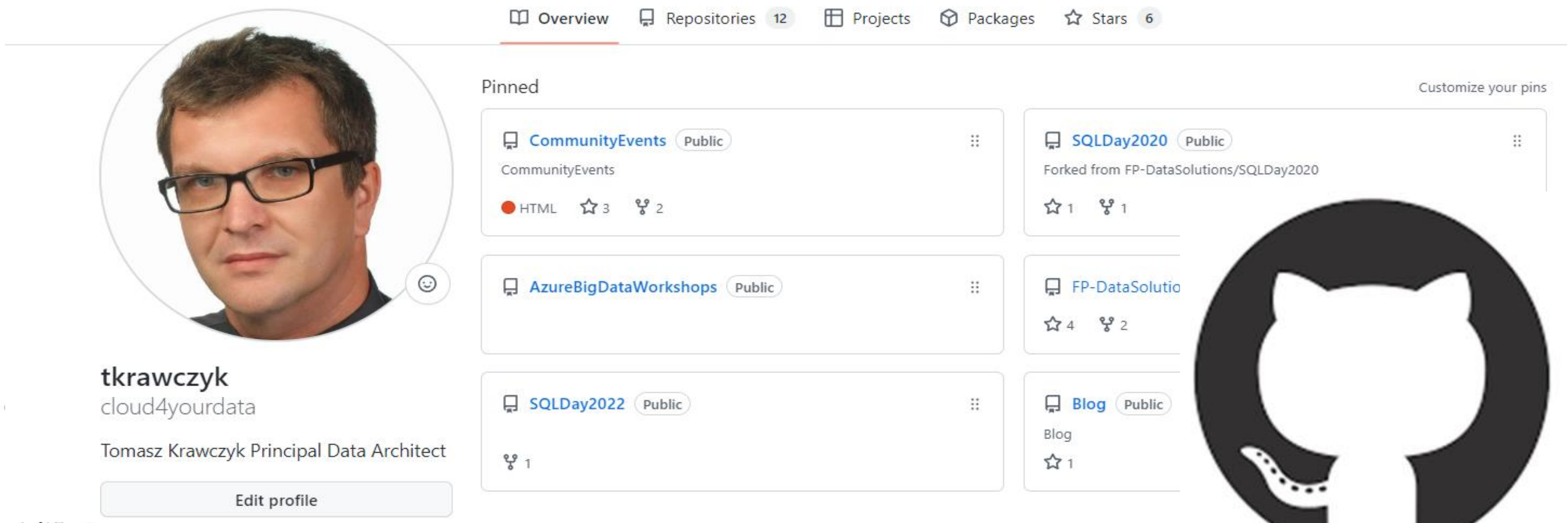
Future Processing

Databricks Unity Catalog - do we really need it?

Tomasz Krawczyk
Future Processing - Data Solutions

Future Processing



tkrawczyk
cloud4yourdata

Tomasz Krawczyk Principal Data Architect

Pinned

CommunityEvents Public
CommunityEvents
HTML ☆ 3 ⑂ 2

SQLDay2020 Public
Forked from FP-DataSolutions/SQLDay2020
☆ 1 ⑂ 1

AzureBigDataWorkshops Public

FP-DataSolutio
☆ 4 ⑂ 2

SQLDay2022 Public
⑂ 1

Blog Public
Blog
☆ 1

https://github.com/fp-datasolutions

https://github.com/cloud4yourdata/CommunityEvents

PLAN

- Hive/Spark/Databricks Metastore
- Unity Catalog
  - Setup (on Azure)
  - Structure (Unity Catalog Objects)
  - Access Control
  - Data Lineage
  - Data Security (Row and Column)
  - Data Lakehouse Federation
- Demo(s)
- Q&A

Hive and Spark Metastore

Apache Hive is a distributed, fault-tolerant **data warehouse** system that enables analytics at a **massive scale**.

Hive Metastore is a repository containing metadata (databases, tables, column names, data types, comments, etc.) about objects we create in Hive. By default, Hive uses a built-in **Derby SQL server** to store its metadata, but in production solutions usually RDBS solution are used (MySQL, MariaDB, PostgreSQL, SQL Server …) .

Apache Spark™ is a multi-language engine for executing **data engineering**, **data science**, and **machine learning** on single-node machines or clusters.

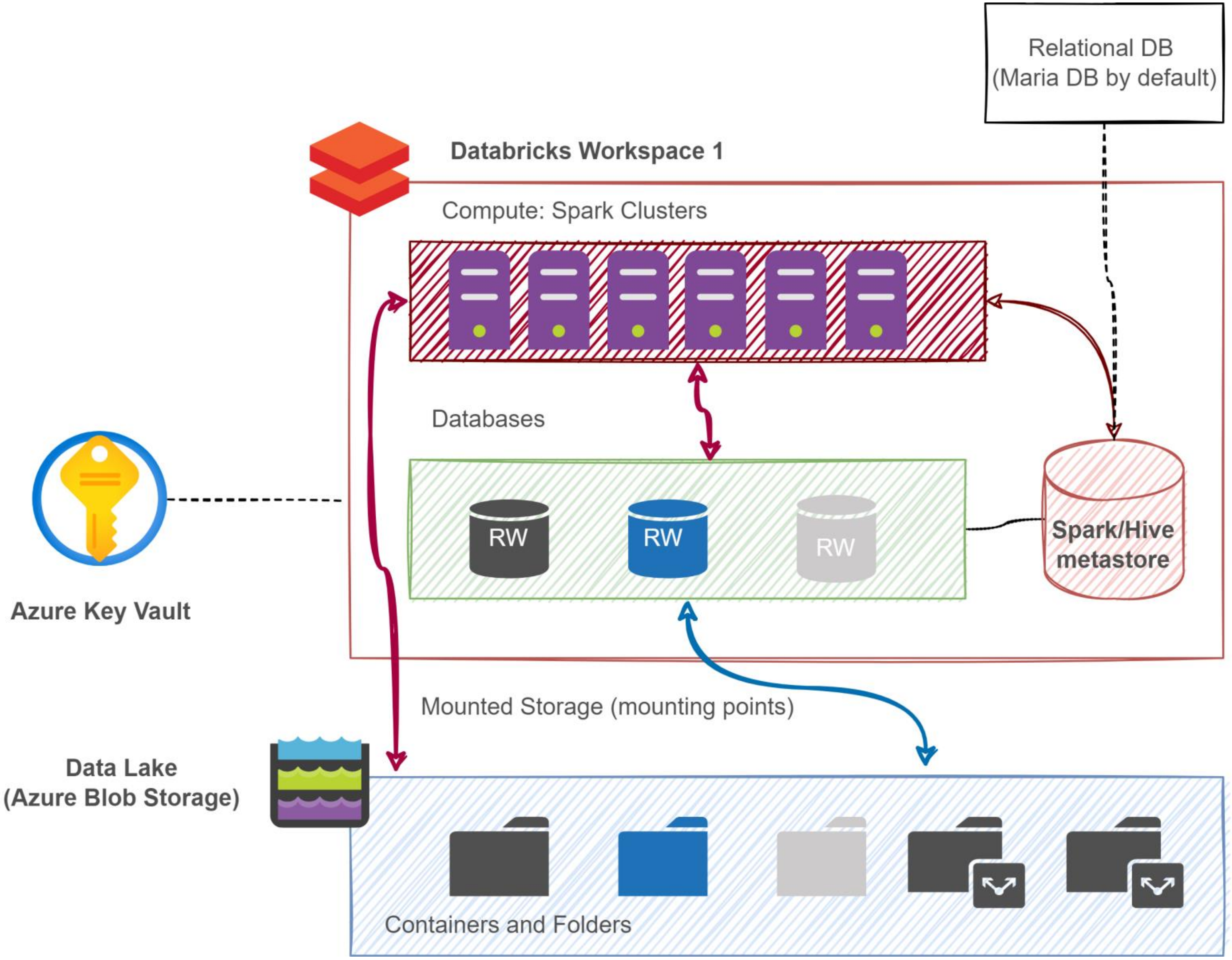
Spark SQL was released in May 2014 as an enhancement to Shark, which was principally a SQL front end to Hive. Spark SQL provides a programming abstraction called DataFrame that can act as distributed **SQL query engine**.

Databricks

Databricks is a unified, open **analytics platform** for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and **AI solutions** at **scale**.

The **Databricks Lakehouse Platform** combines the best elements of **data lakes** and **data warehouses** to help you reduce costs and deliver on your data and AI initiatives faster.

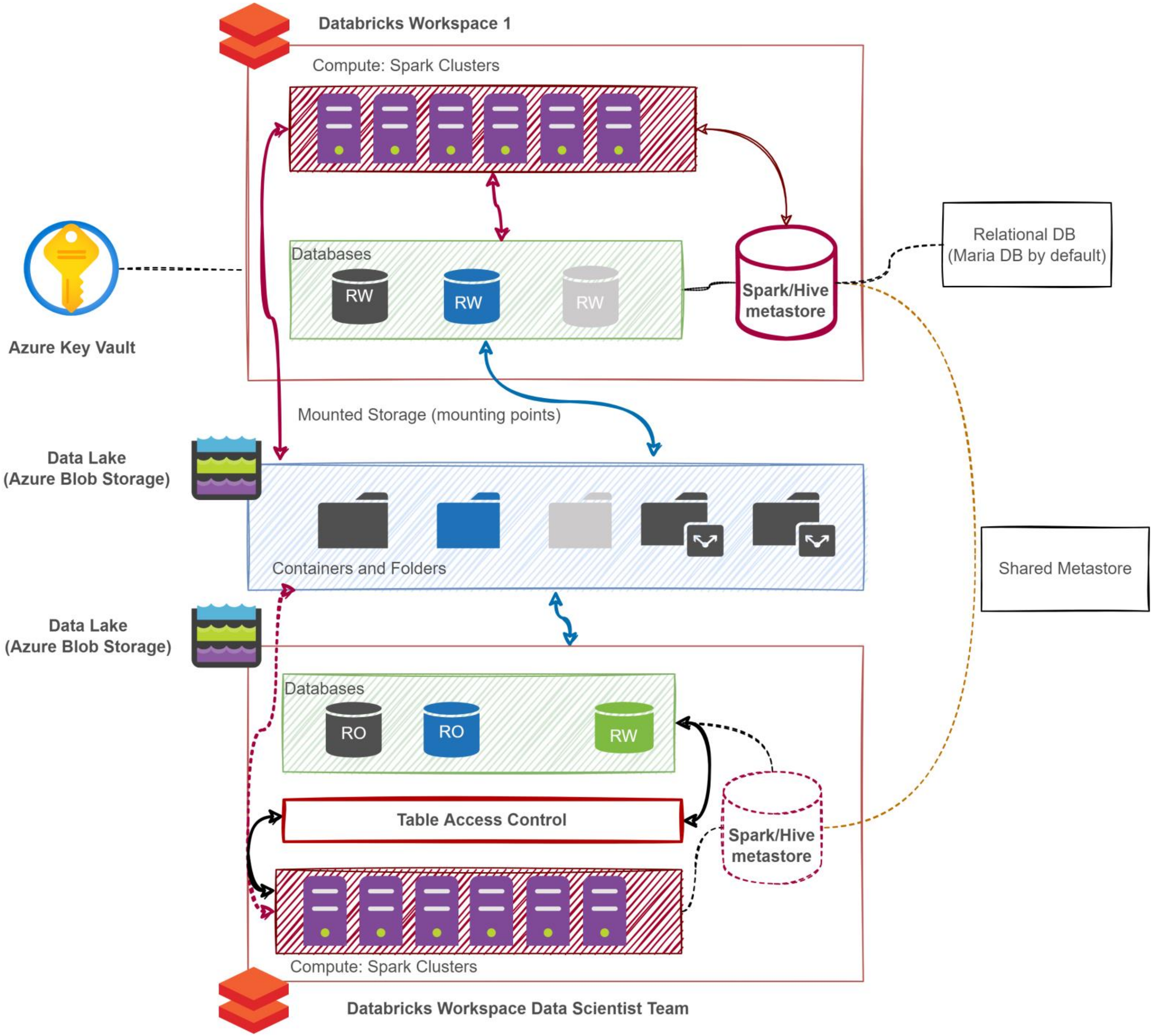
Built on **open source** and **open standards**, a lakehouse simplifies your data estate by eliminating the silos that historically complicate data and AI.



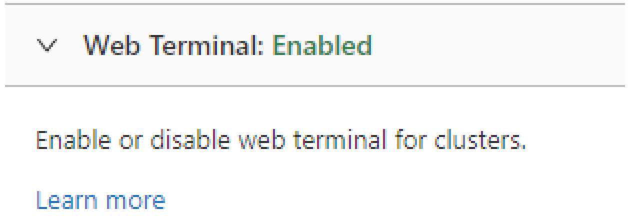
Databricks – "Data Mesh"

- **OUR CASE**
  - One Workspace for ETL processes
  - Additional Workspaces for Data Scientist's teams
    - Read Only Access

- **ETL Workspace**
  - Internal Metastore
  - Mounted ADLS Gen2 Storage
  - Databases, tables …



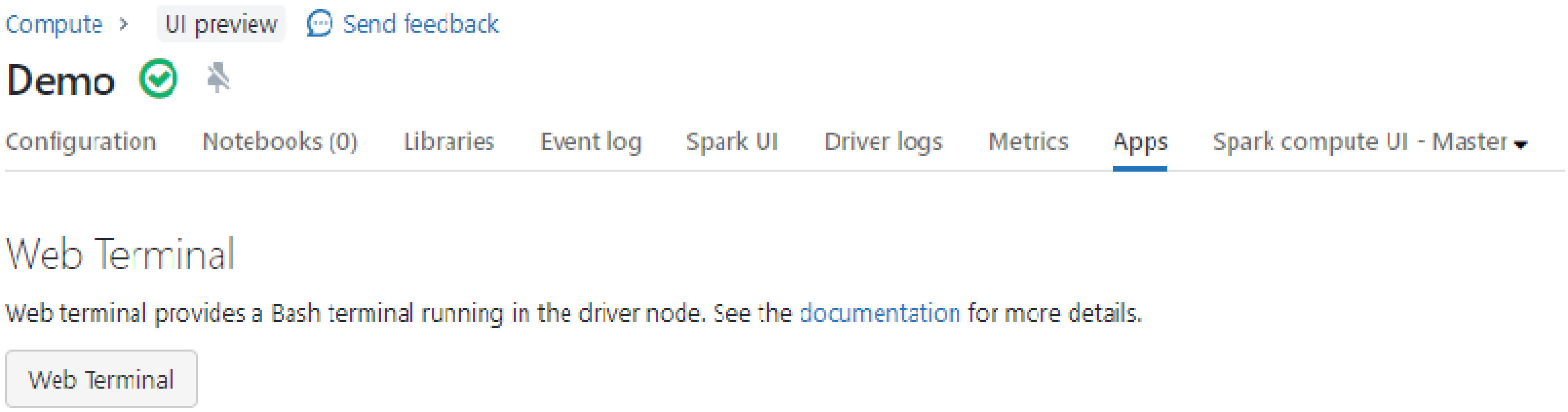
Databricks Access to Internal Metastore



Web Terminal: Enabled

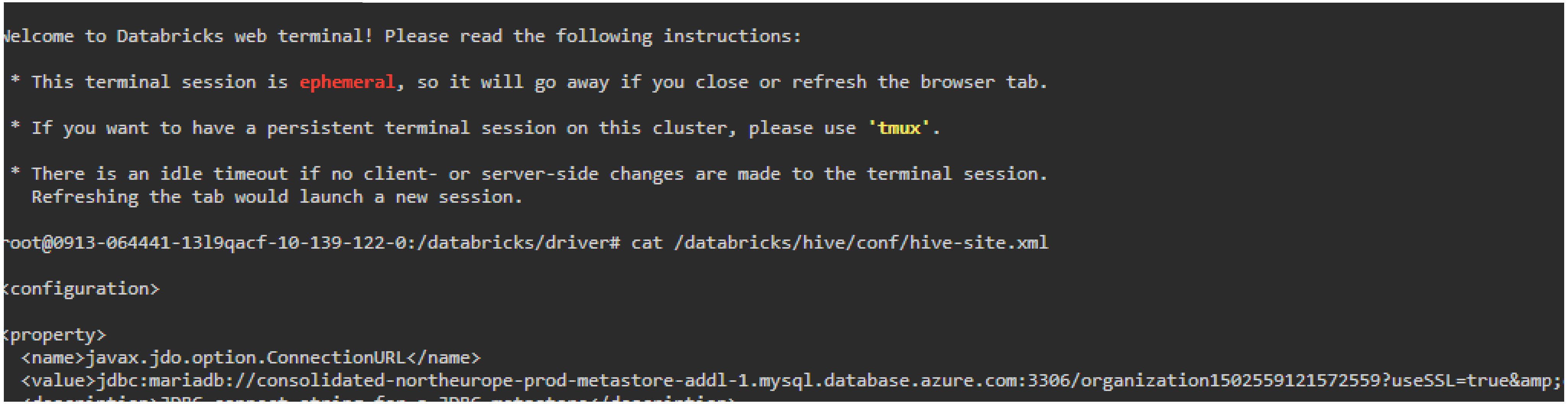
Enable or disable web terminal for clusters.

Learn more



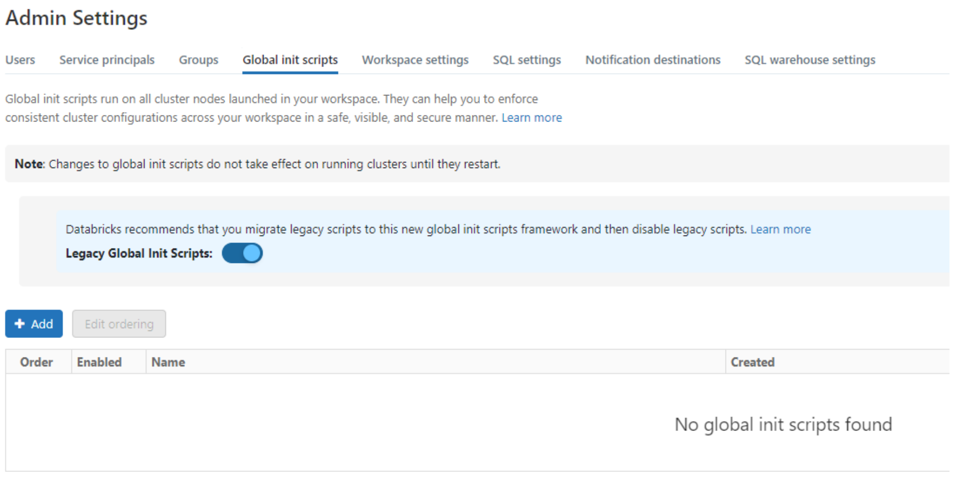
Web Terminal

Web terminal provides a Bash terminal running in the driver node. See the documentation for more details.



```
Welcome to Databricks web terminal! Please read the following instructions:

 * This terminal session is ephemeral, so it will go away if you close or refresh the browser tab.

 * If you want to have a persistent terminal session on this cluster, please use 'tmux'.

 * There is an idle timeout if no client- or server-side changes are made to the terminal session.
   Refreshing the tab would launch a new session.

root@0913-064441-1319qacf-10-139-122-0:/databricks/driver# cat /databricks/hive/conf/hive-site.xml

<configuration>

<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mariadb://consolidated-northeurope-prod-metastore-addl-1.mysql.database.azure.com:3306/organization1502559121572559?useSSL=true&amp;
  <description>JDBC connect string for a JDBC metastore</description>
```

Databricks External Metastore



Admin Settings

Global init scripts run on all cluster nodes launched in your workspace. They can help you to enforce consistent cluster configurations across your workspace in a safe, visible, and secure manner. Learn more

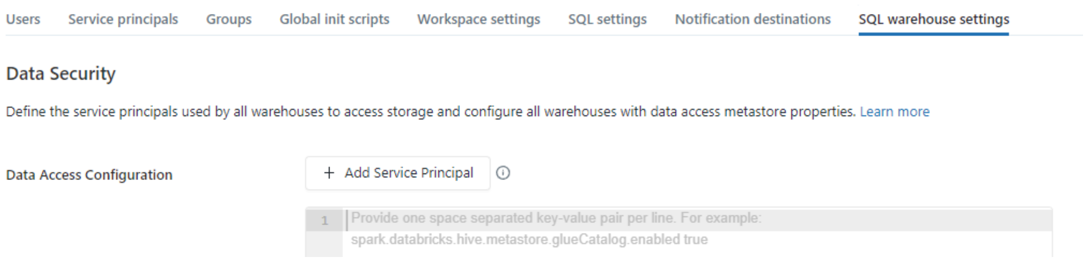
Note: Changes to global init scripts do not take effect on running clusters until they restart.

Databricks recommends that you migrate legacy scripts to this new global init scripts framework and then disable legacy scripts. Learn more

Legacy Global Init Scripts: ●

Order	Enabled	Name	Created

No global init scripts found

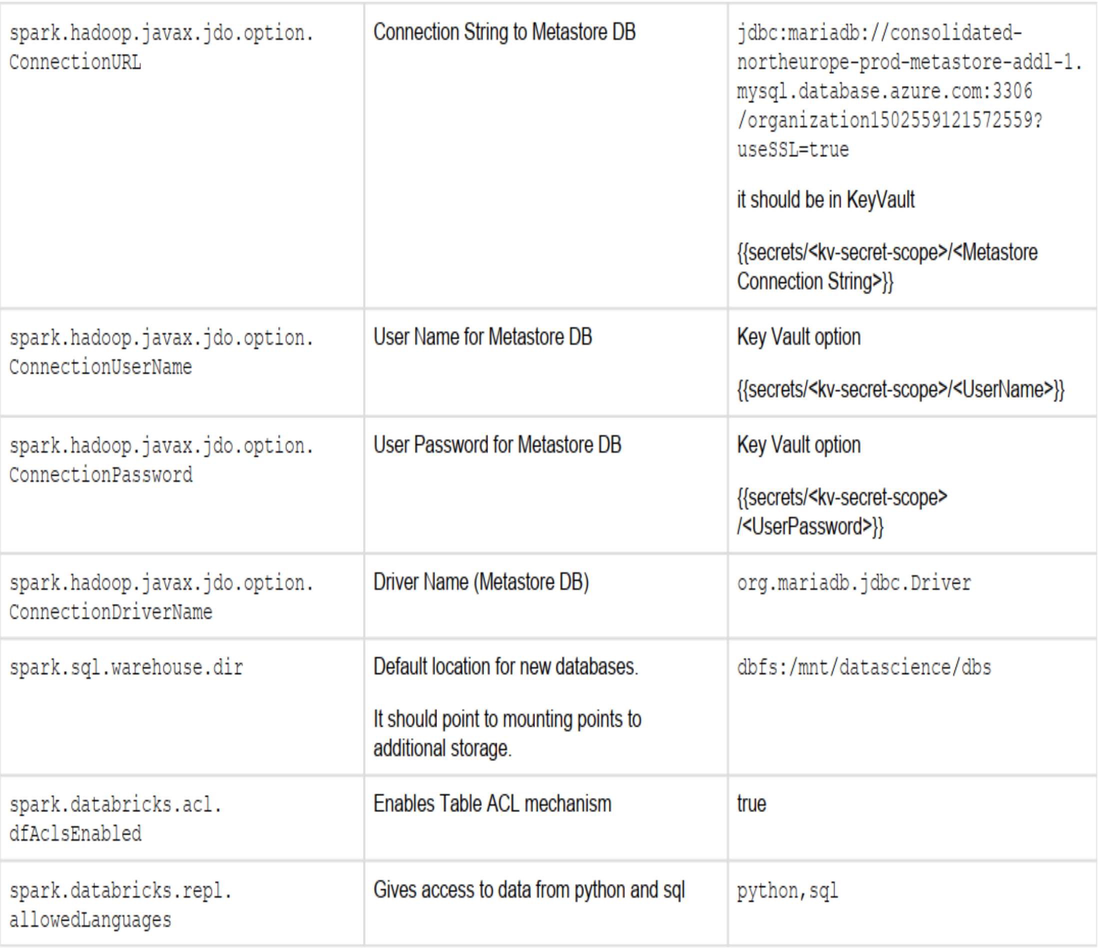


Data Security

Define the service principals used by all warehouses to access storage and configure all warehouses with data access metastore properties. Learn more

Data Access Configuration

Provide one space separated key-value pair per line. For example:
spark.databricks.hive.metastore.glueCatalog.enabled true



spark.hadoop.javax.jdo.option.ConnectionURL	Connection String to Metastore DB	jdbc:mariadb://consolidated-northeurope-prod-metastore-addl-1.mysql.database.azure.com:3306/organization1502559121572559?useSSL=true it should be in KeyVault {{secrets/<kv-secret-scope>/<Metastore Connection String>}}
spark.hadoop.javax.jdo.option.ConnectionUserName	User Name for Metastore DB	Key Vault option {{secrets/<kv-secret-scope>/<UserName>}}
spark.hadoop.javax.jdo.option.ConnectionPassword	User Password for Metastore DB	Key Vault option {{secrets/<kv-secret-scope>/<UserPassword>}}
spark.hadoop.javax.jdo.option.ConnectionDriverName	Driver Name (Metastore DB)	org.mariadb.jdbc.Driver
spark.sql.warehouse.dir	Default location for new databases. It should point to mounting points to additional storage.	dbfs:/mnt/datascience/dbs
spark.databricks.acl.dfAclsEnabled	Enables Table ACL mechanism	true
spark.databricks.repl.allowedLanguages	Gives access to data from python and sql	python,sql

Global Init Script

Docs: https://docs.microsoft.com/en-us/azure/databricks/clusters/init-scripts#global-init-scrip

Databrick Access Control - TAC

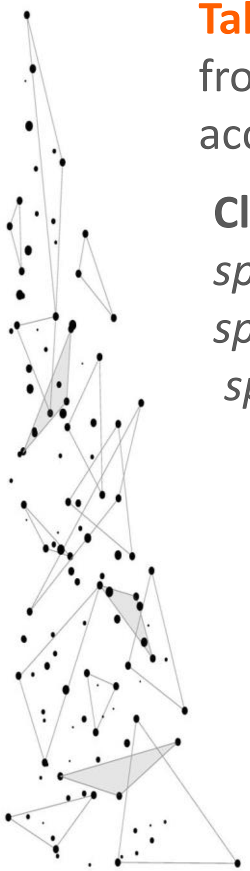


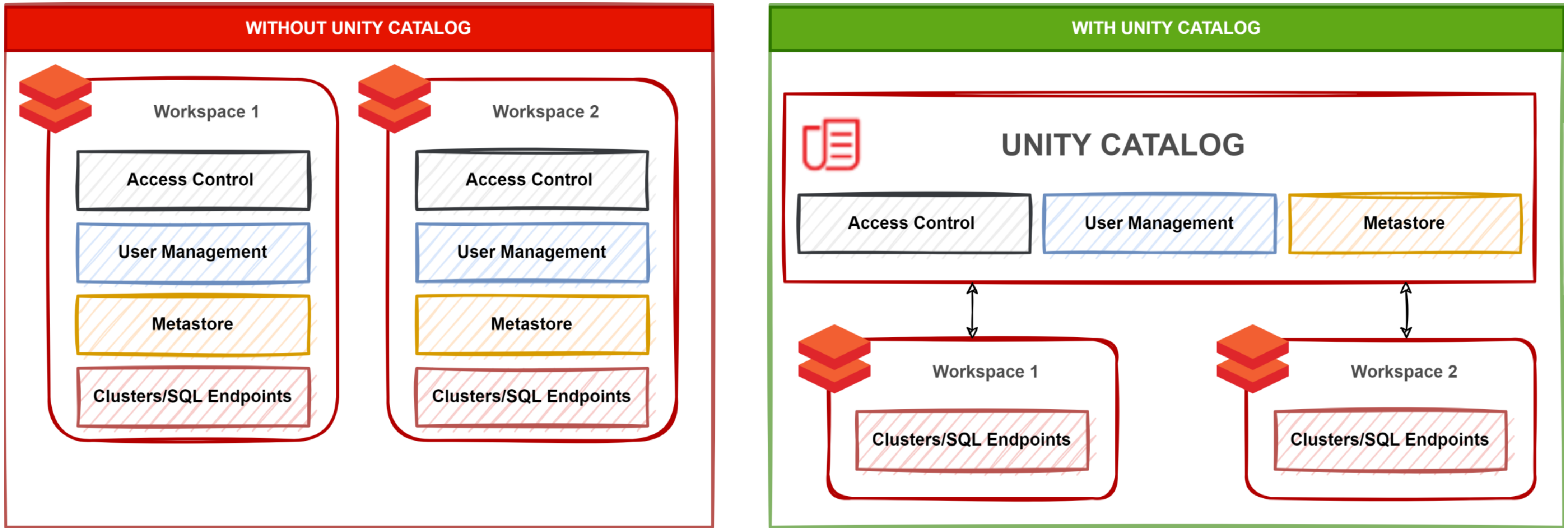
Table access control lets you programmatically **grant** and **revoke access** to objects in your workspace's Hive metastore from **Python** and **SQL**. When table access control is enabled, **users can set permissions for data objects** that are accessed using that cluster.

Cluster settings:
spark.databricks.acl.sqlOnly true
spark.databricks.repl.allowedLanguages python,sql
spark.databricks.acl.dfAclsEnabled true

- **GRANT**/**REVOKE** *privilege_types* ON *securable_object* TO *principal*
  - Privilege Types:
    - SELECT, CREATE, MODIFY, USAGE, READ_METADATA, ALL PRIVILEGES
  - Securable objects
    - DATABASE, TABLE, VIEW, FUNCTION, **ANY FILES**

Databrick Unity Catalog

Unity Catalog provides **centralized access control**, **auditing**, **lineage**, and **data discovery** capabilities across Databricks workspaces.



Databricks Unity Catalog on Azure

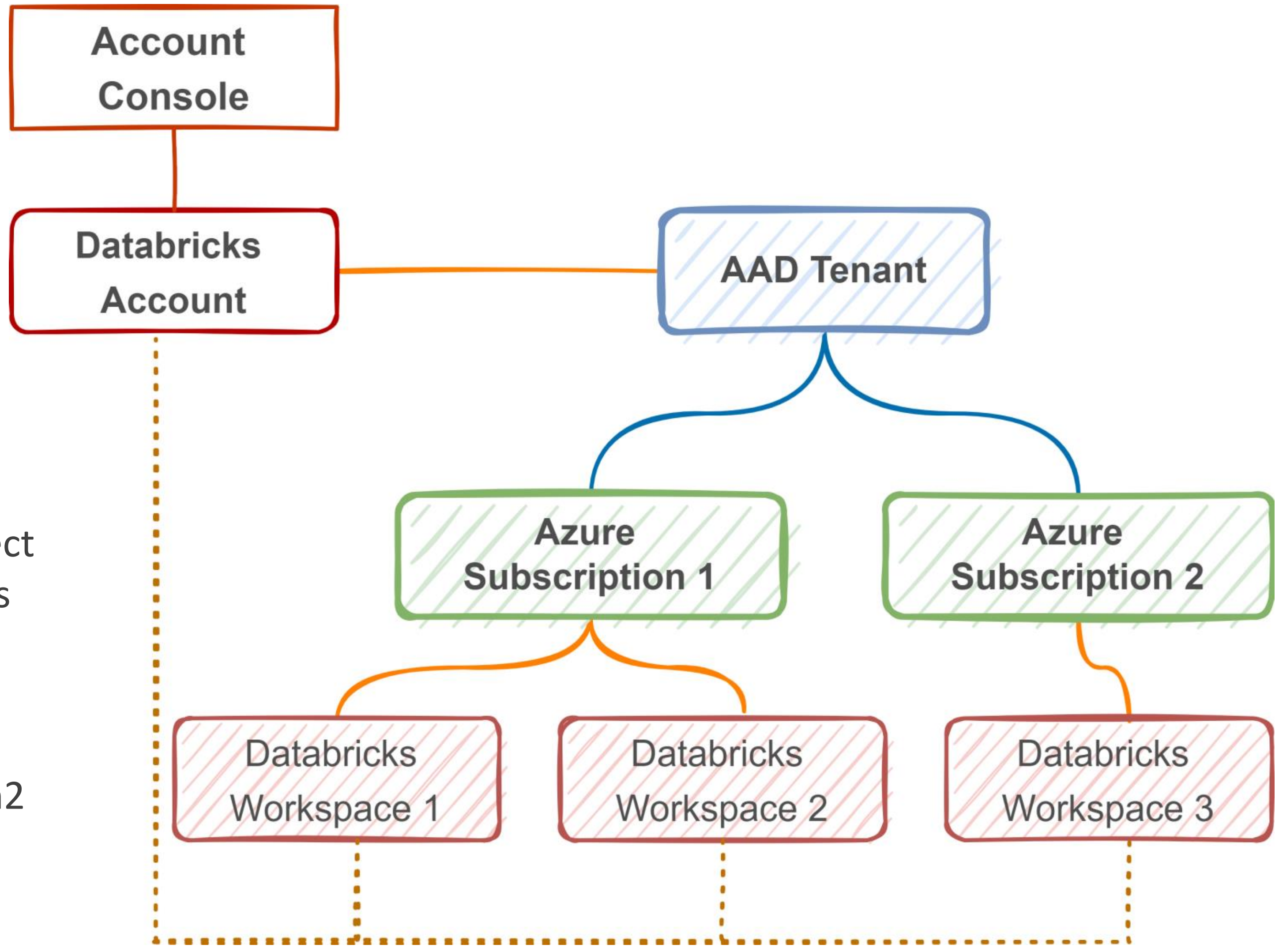
Account Console:

https://accounts.azuredatabricks.net/

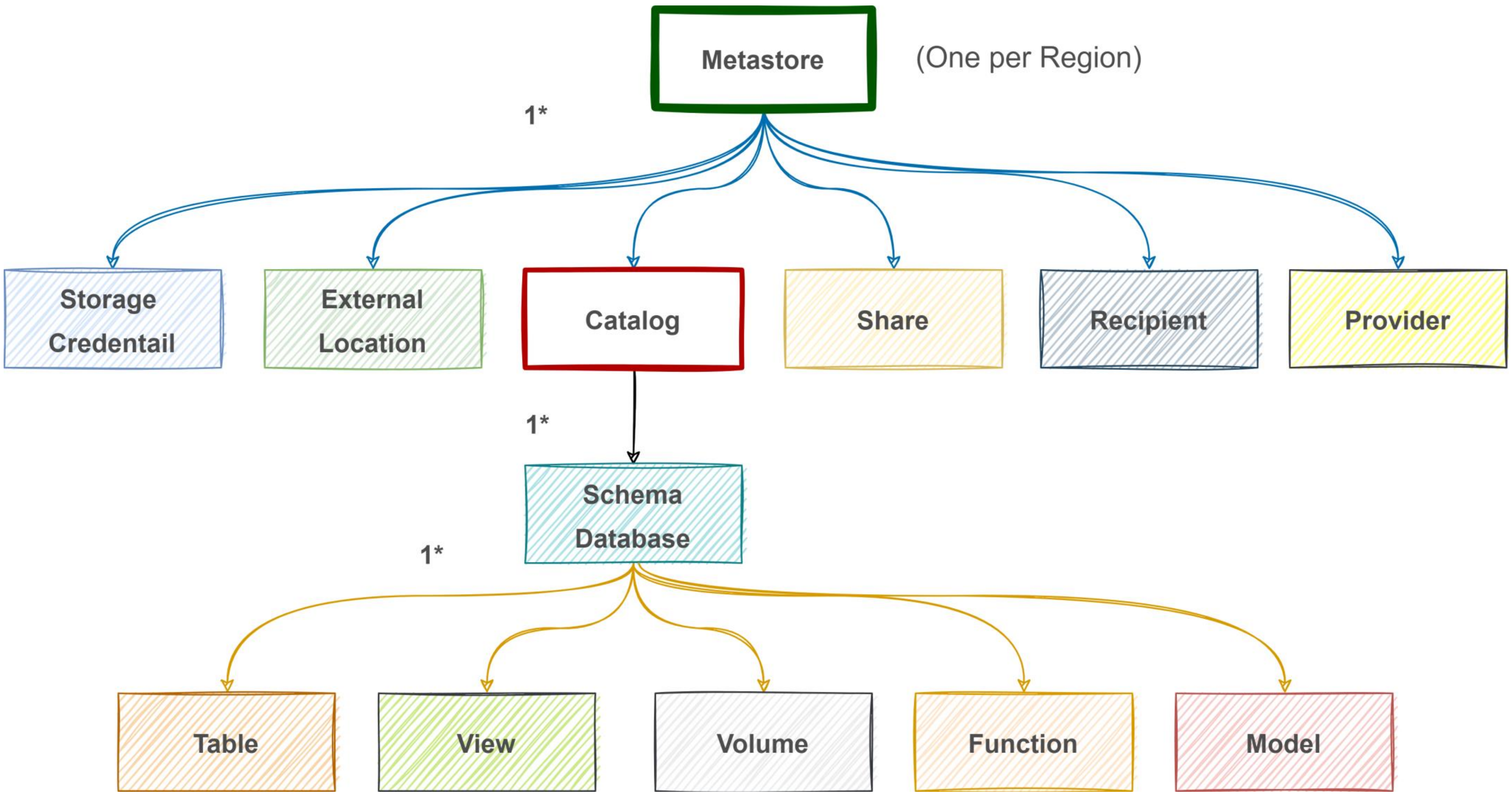
Azure AD Global Administrator role

Access Connector for Azure Databricks
- is an Azure resource that lets you connect managed identities to an Azure Databricks account.

Managed storage
- location in an Azure Data Lake Storage Gen2 container to store data and metadata

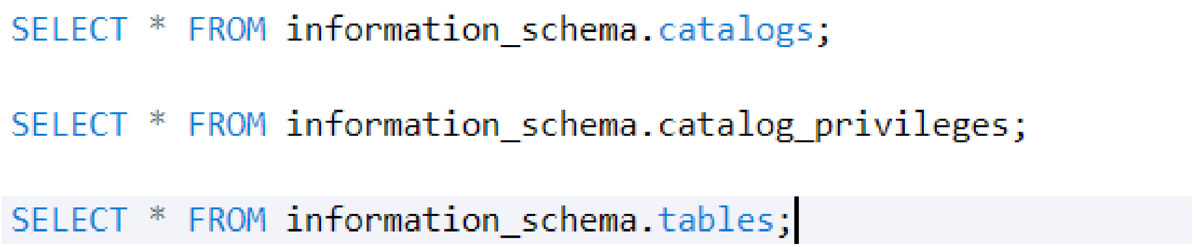


Unity Catalog



Unity Catalog – Objects Metadata

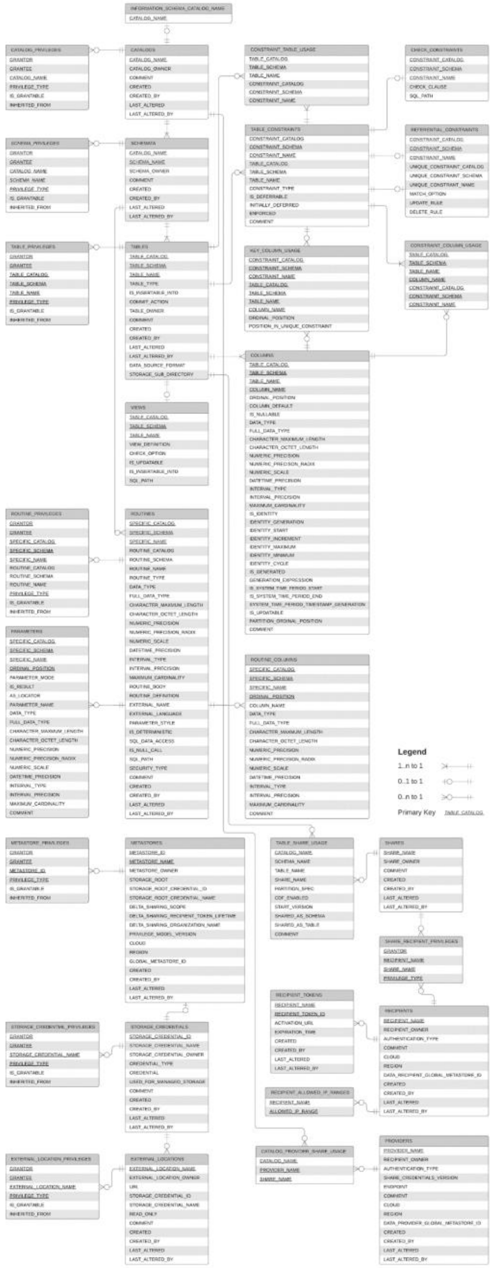
The *INFORMATION_SCHEMA* is a SQL standard based schema, provided in every catalog created on **Unity Catalog**.



```
SELECT * FROM information_schema.catalogs;

SELECT * FROM information_schema.catalog_privileges;

SELECT * FROM information_schema.tables;
```

System tables are an Databricks-hosted analytical store of your account’s operational data

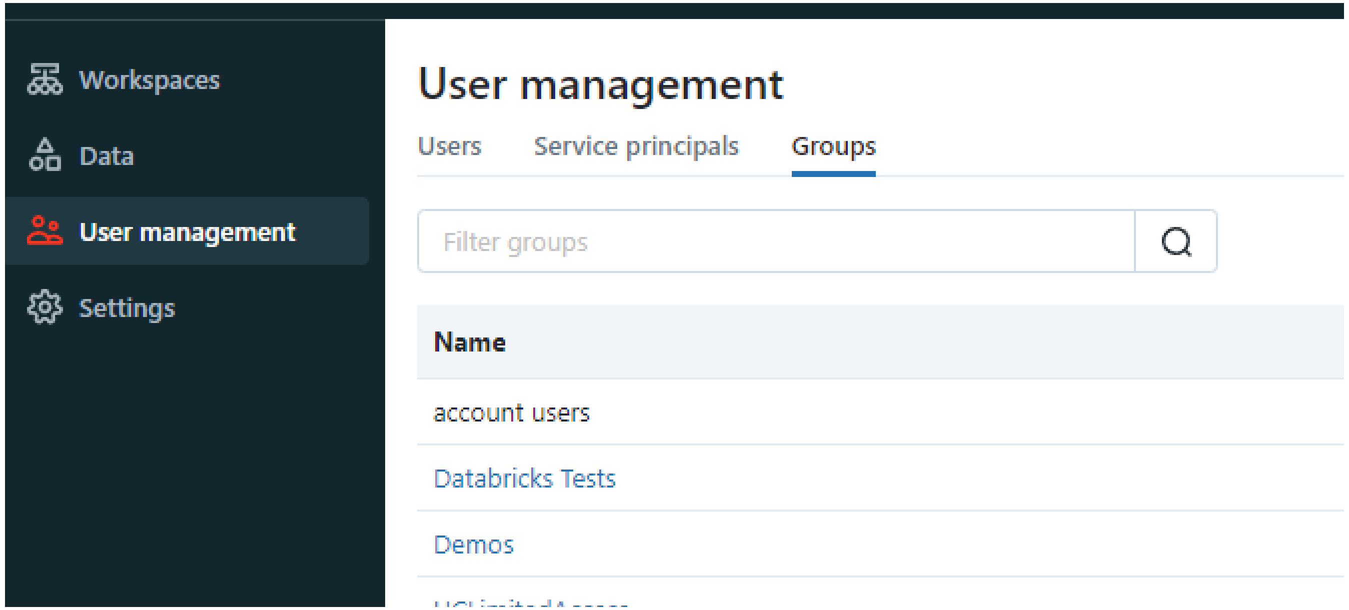
- **Audit logs**: Located at system.access.audit.
- **Billable usage logs**: Located at system.billing.usage.
- **Pricing table**: Located at system.billing.list_prices.
- **Table and column lineage**: Both tables located under system.access.
- **Marketplace listing access**: Located at system.marketplace.listing_access_events.



Unity Catalog

Managing Users and Access Control

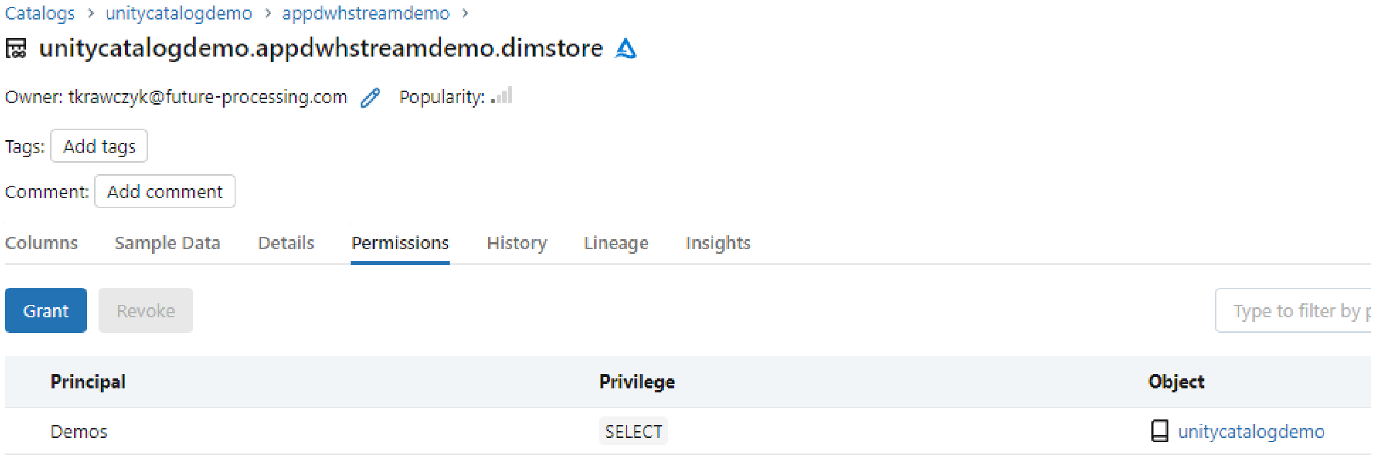
- Azure Databricks SCIM Provisioning Connector
  - synchronizes users and groups from AD to Azure Databricks (Docs)
- **GRANT/REVOKE** *privilege_types* ON *securable_object* TO *principal*
  - Privilege Types:
    - SELECT, CREATE, MODIFY, USAGE, READ_METADATA, ALL PRIVILEGES
  - Securable objects
    - DATABASE, TABLE, VIEW, FUNCTION ..



User management

Users | Service principals | Groups

Name
account users
Databricks Tests
Demos



unitycatalogdemo.appdwhstreamdemo.dimstore

Owner: tkrawczyk@future-processing.com Popularity:

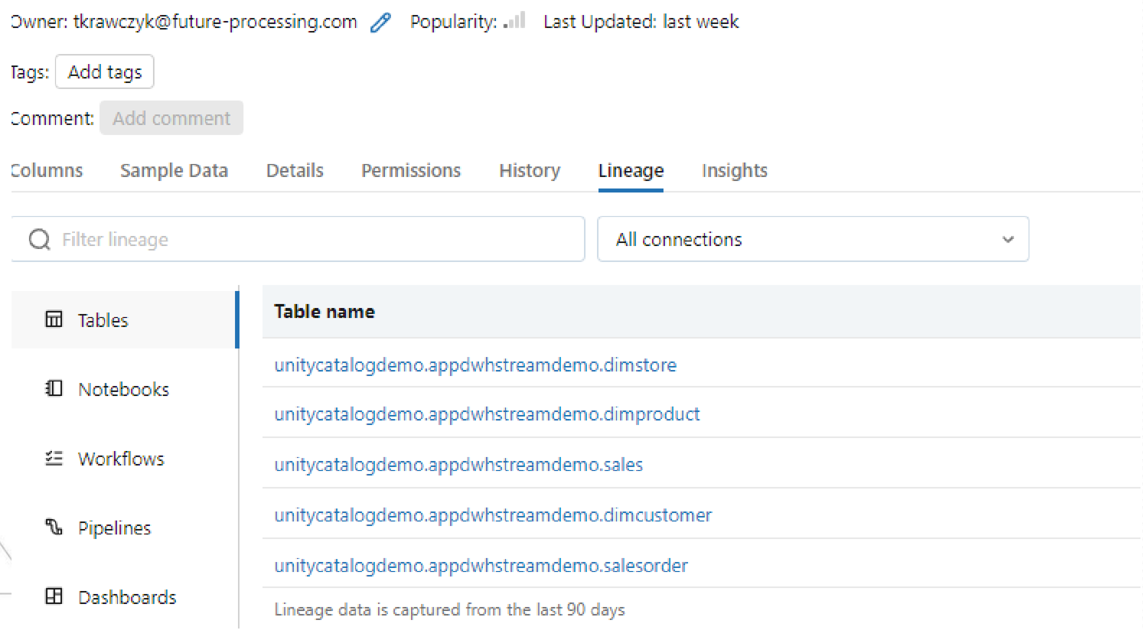
Tags: Add tags

Comment: Add comment

Principal	Privilege	Object
Demos	SELECT	unitycatalogdemo

Unity Catalog Data Lineage

Data Lineage is supported for all languages and is captured down to the column level. Lineage data includes notebooks, workflows, and dashboards related to the query.



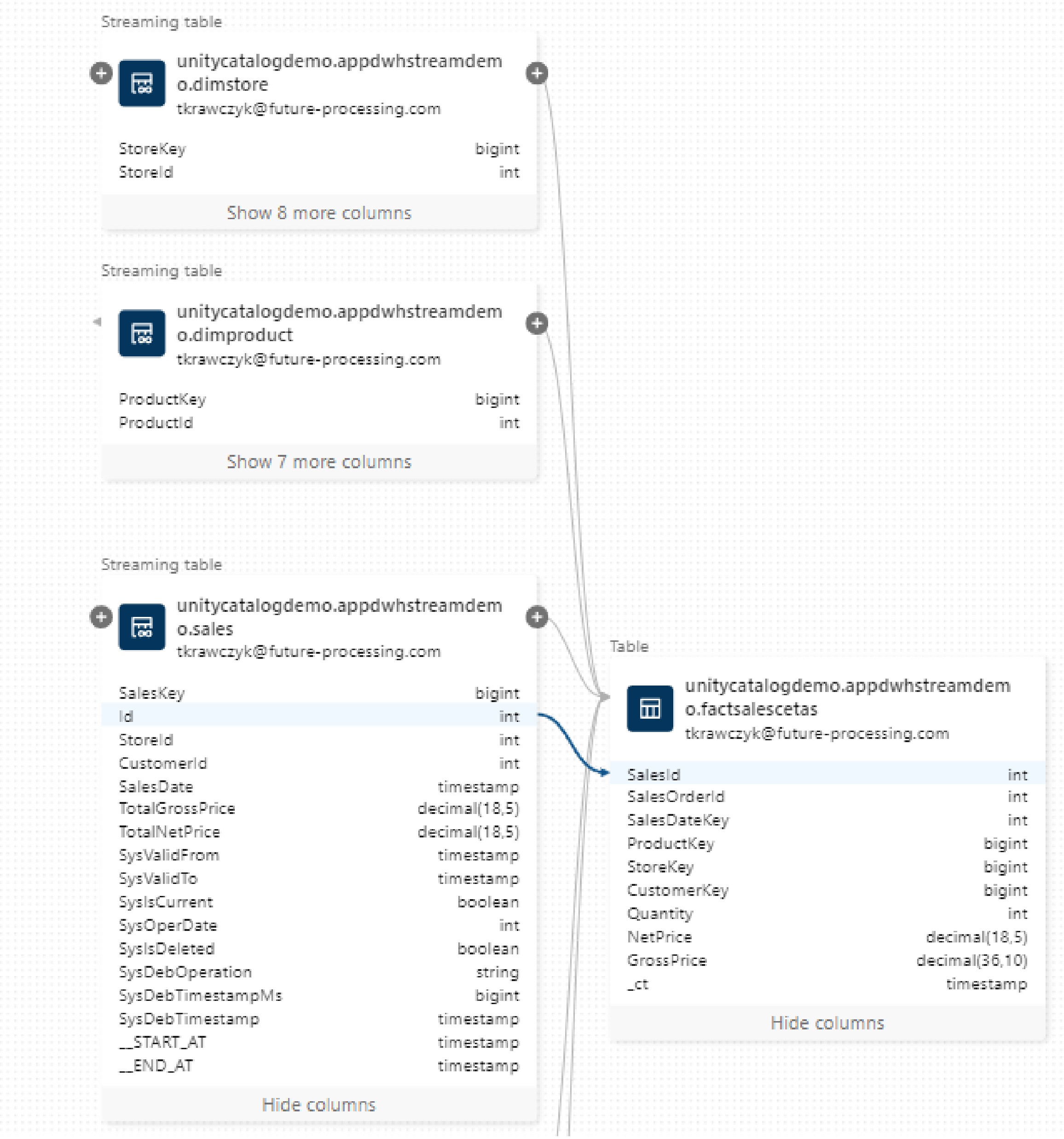
Owner: tkrawczyk@future-processing.com Popularity: Last Updated: last week

Tags: Add tags

Comment: Add comment

Table name
unitycatalogdemo.appdwhstreamdemo.dimstore
unitycatalogdemo.appdwhstreamdemo.dimproduct
unitycatalogdemo.appdwhstreamdemo.sales
unitycatalogdemo.appdwhstreamdemo.dimcustomer
unitycatalogdemo.appdwhstreamdemo.salesorder

Lineage data is captured from the last 90 days



Streaming table

unitycatalogdemo.appdwhstreamdemo.dimstore
tkrawczyk@future-processing.com

StoreKey	bigint
StoreId	int

Show 8 more columns

Streaming table

unitycatalogdemo.appdwhstreamdemo.dimproduct
tkrawczyk@future-processing.com

ProductKey	bigint
ProductId	int

Show 7 more columns

Streaming table

unitycatalogdemo.appdwhstreamdemo.sales
tkrawczyk@future-processing.com

SalesKey	bigint
Id	int
StoreId	int
CustomerId	int
SalesDate	timestamp
TotalGrossPrice	decimal(18,5)
TotalNetPrice	decimal(18,5)
SysValidFrom	timestamp
SysValidTo	timestamp
SysIsCurrent	boolean
SysOperDate	int
SysIsDeleted	boolean
SysDebOperation	string
SysDebTimestampMs	bigint
SysDebTimestamp	timestamp
__START_AT	timestamp
__END_AT	timestamp

Hide columns

Table

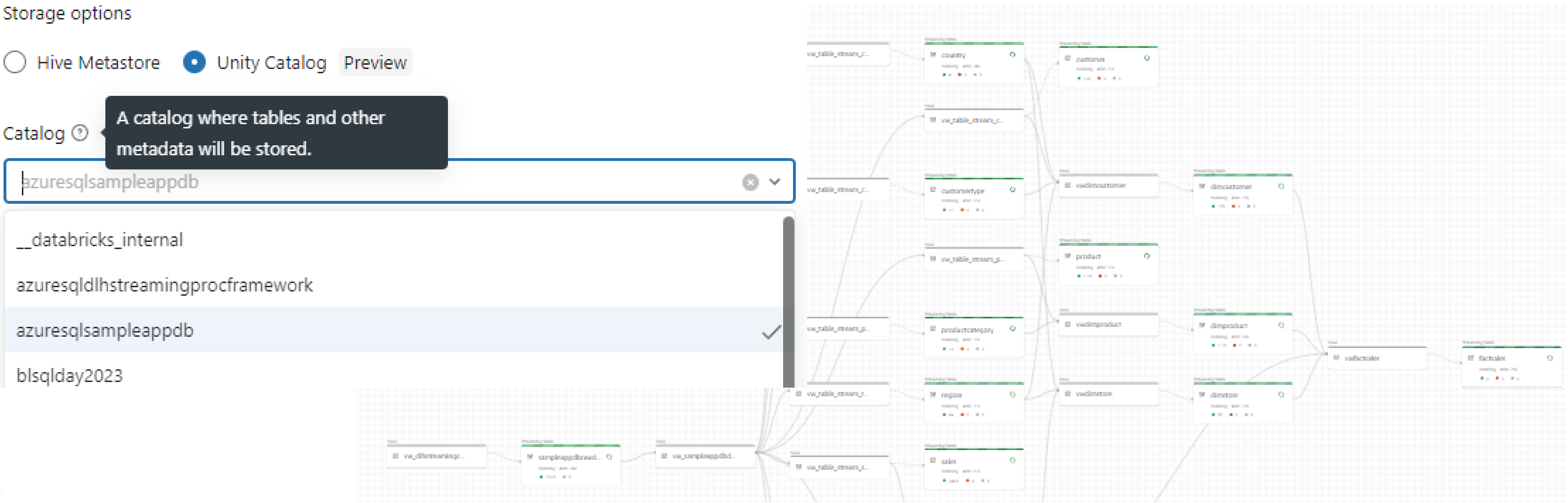
unitycatalogdemo.appdwhstreamdemo.factsalescetas
tkrawczyk@future-processing.com

SalesId	int
SalesOrderId	int
SalesDateKey	int
ProductKey	bigint
StoreKey	bigint
CustomerKey	bigint
Quantity	int
NetPrice	decimal(18,5)
GrossPrice	decimal(36,10)
_ct	timestamp

Hide columns

Unity Catalog Data Lineage and Delta Live Tables

Delta Live Tables is a declarative framework for building reliable, maintainable, and testable data processing pipelines.



Storage options

○ Hive Metastore ● Unity Catalog Preview

Catalog

A catalog where tables and other metadata will be stored.

azuresqlsampleappdb

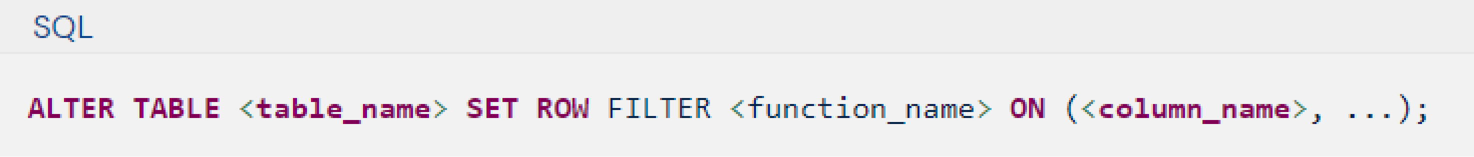
- __databricks_internal
- azuresqldlhstreamingprocframework
- azuresqlsampleappdb ✓
- blsqlday2023

Unity Catalog - Row filters and column masking

- **IS_ACCOUNT_GROUP_MEMBER (** account level)
- **IS_MEMBER** (workspace local group)

- **ROW FILTER FUNCTION**

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



SQL

```
ALTER TABLE <table_name> SET ROW FILTER <function_name> ON (<column_name>, ...);
```

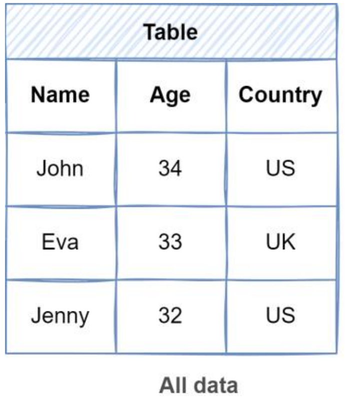
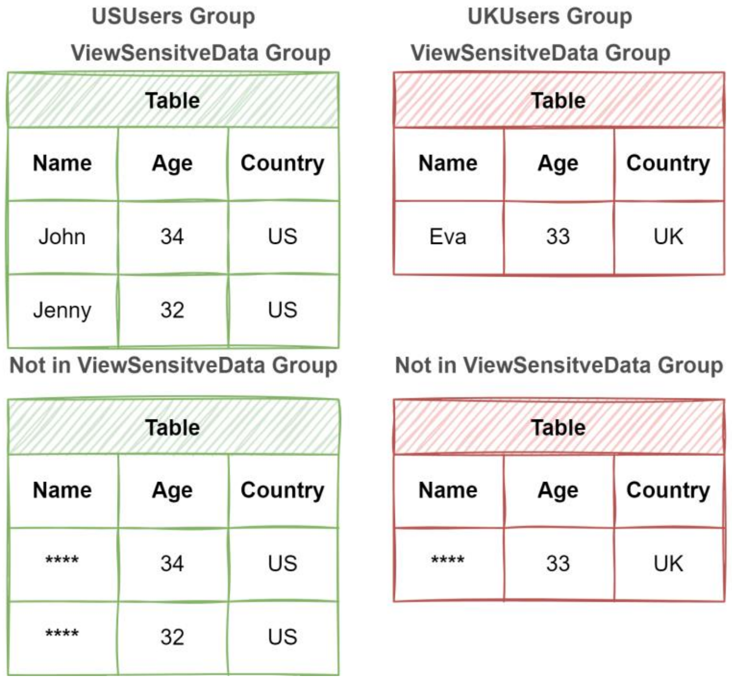


Table		
Name	Age	Country
John	34	US
Eva	33	UK
Jenny	32	US

All data



USUsers Group

ViewSensitveData Group

Table		
Name	Age	Country
John	34	US
Jenny	32	US

Not in ViewSensitveData Group

Table		
Name	Age	Country
****	34	US
****	32	US

UKUsers Group

ViewSensitveData Group

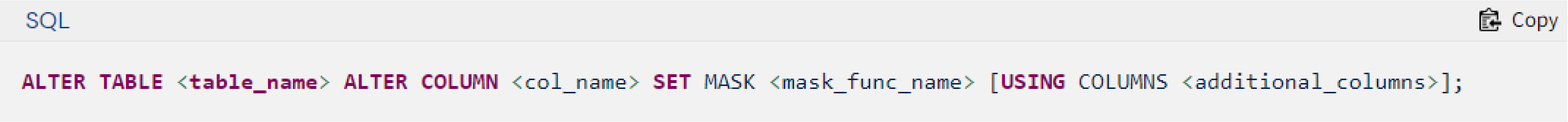
Table		
Name	Age	Country
Eva	33	UK

Not in ViewSensitveData Group

Table		
Name	Age	Country
****	33	UK

- **COLUMN MASKING FUNCTION**

```
CREATE FUNCTION ssn_mask(ssn STRING)
  RETURN IF(IS_ACCOUNT_GROUP_MEMBER('admin'), ssn, '****');
```



SQL

```
ALTER TABLE <table_name> ALTER COLUMN <col_name> SET MASK <mask_func_name> [USING COLUMNS <additional_columns>];
```

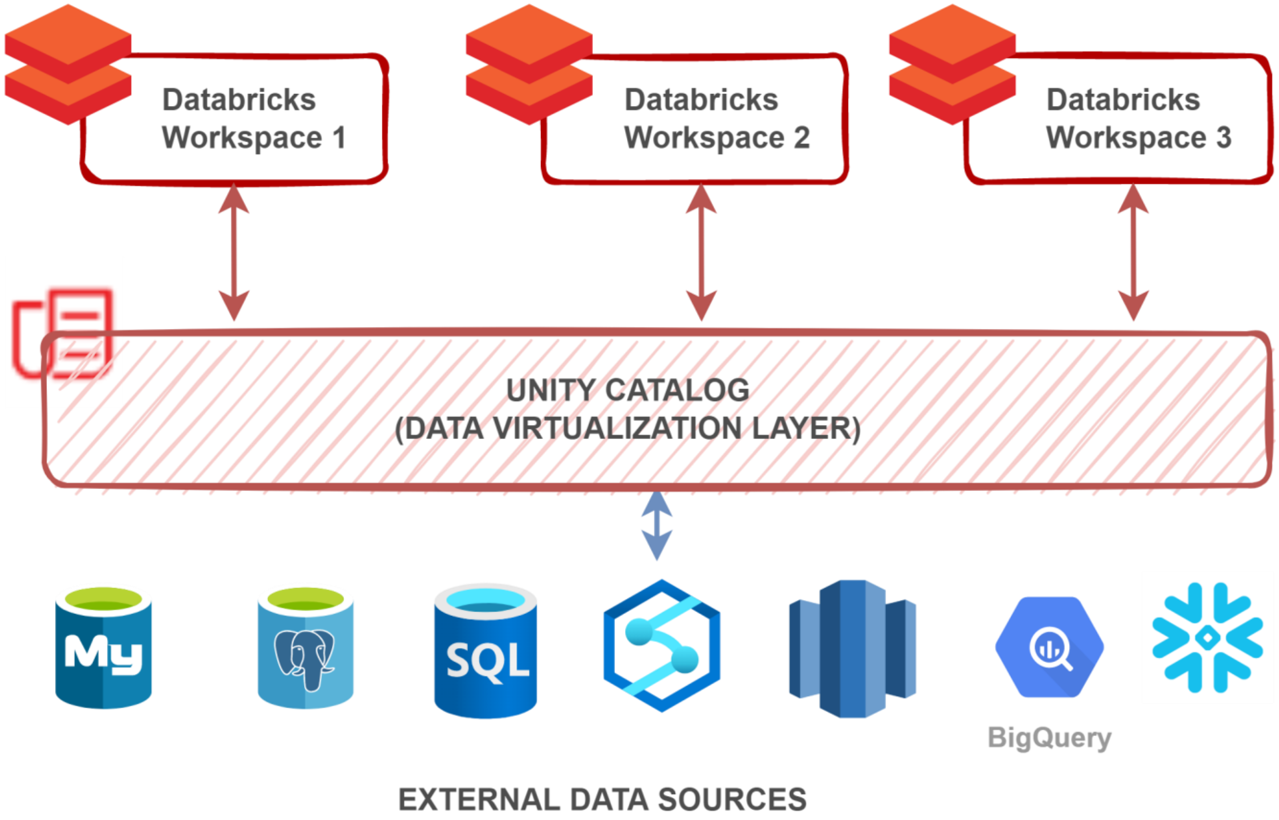
Unity Catalog – Lakehouse Query Federation

Lakehouse Query Federation provides one single secure access to all your data.
Supported data sources:

- **MySQL ,PostgreSQL,Amazon Redshift, Snowflake, Azure SQL Database, Azure Synapse, Google's BigQuery …**

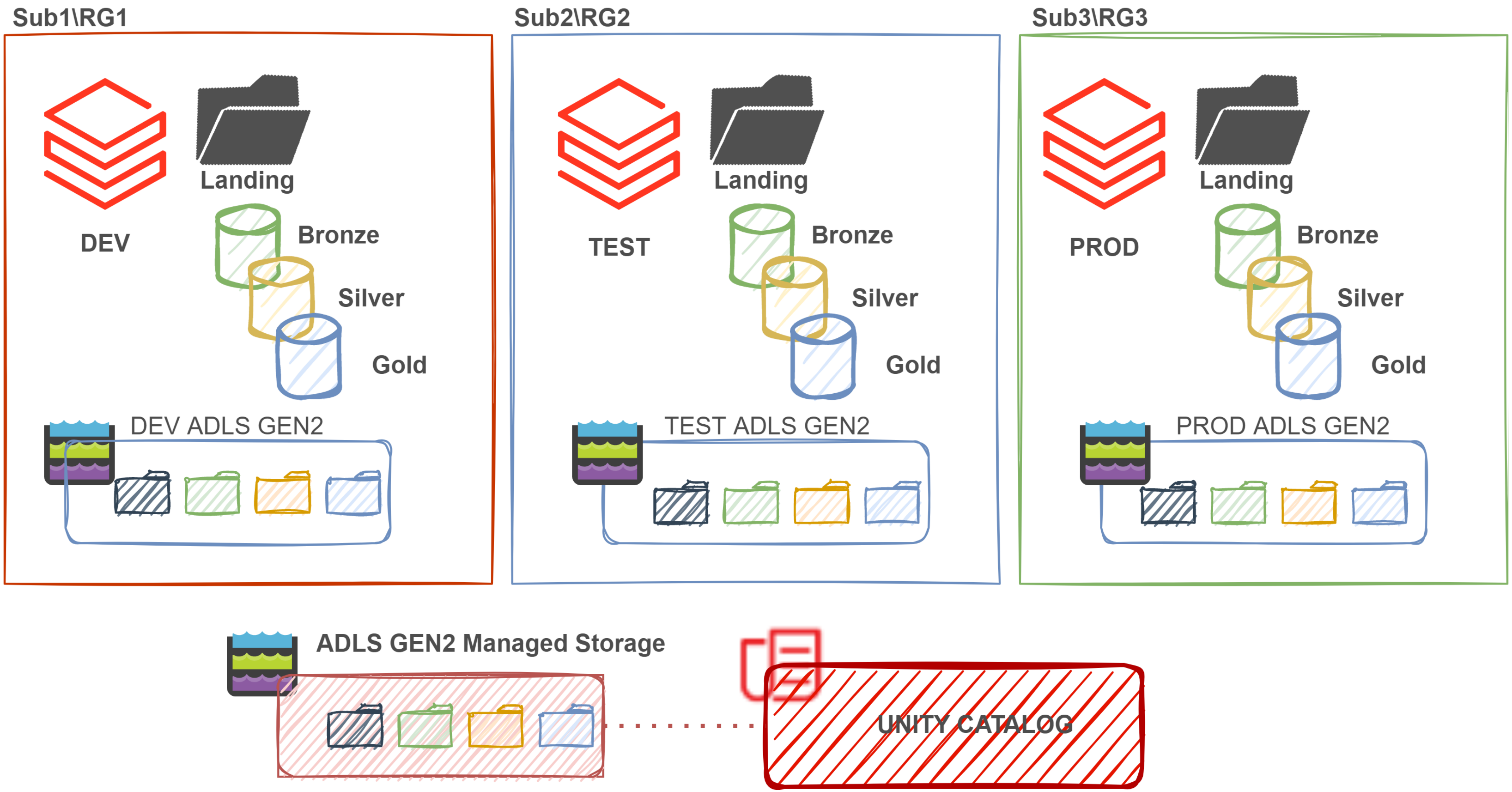
Unity Catalog provides:

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



```
CREATE FOREIGN CATALOG [IF NOT EXISTS] <catalog-name> USING CONNECTION <connection-name>
OPTIONS (database '<database-name>');
```

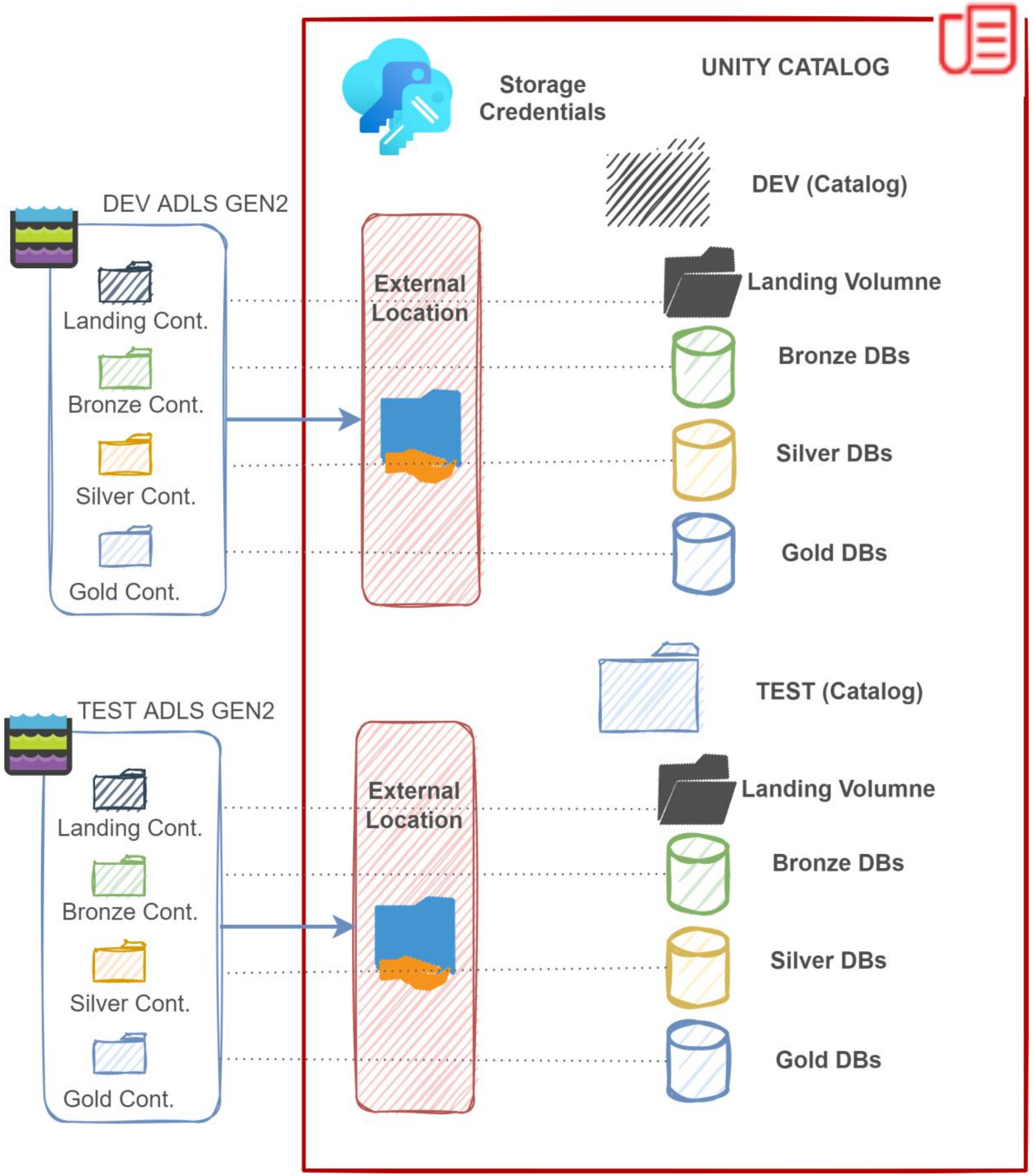
Unity Catalog –Environments



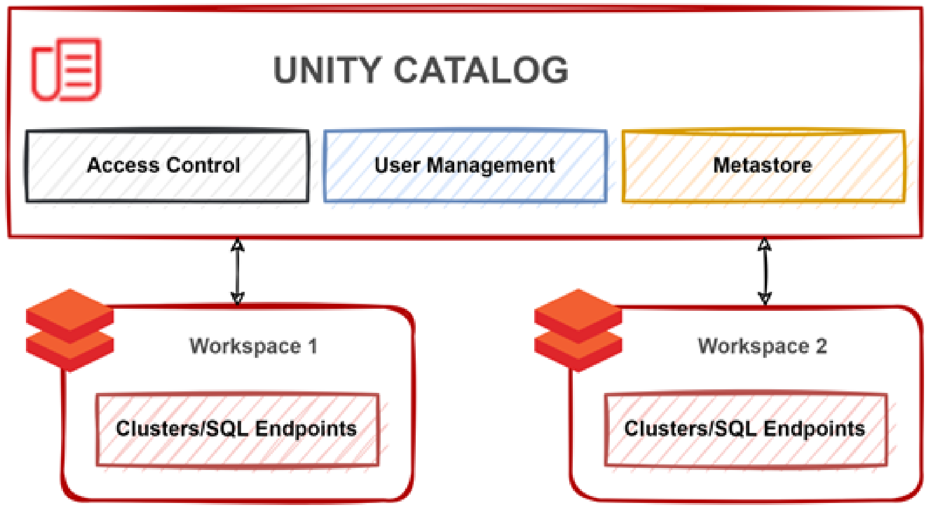
Unity Catalog –Environments

A **storage credential** represents an authentication and authorization mechanism for **accessing data stored on your cloud tenant**, using an Azure **managed identity** (strongly recommended) or **service principal**.

An **external location** is an object that combines a **cloud storage path** with a storage credential that authorizes access to the cloud storage path.



Data Mesh



UNITY CATALOG

Access Control

User Management

Metastore

Workspace 1

Clusters/SQL Endpoints

Workspace 2

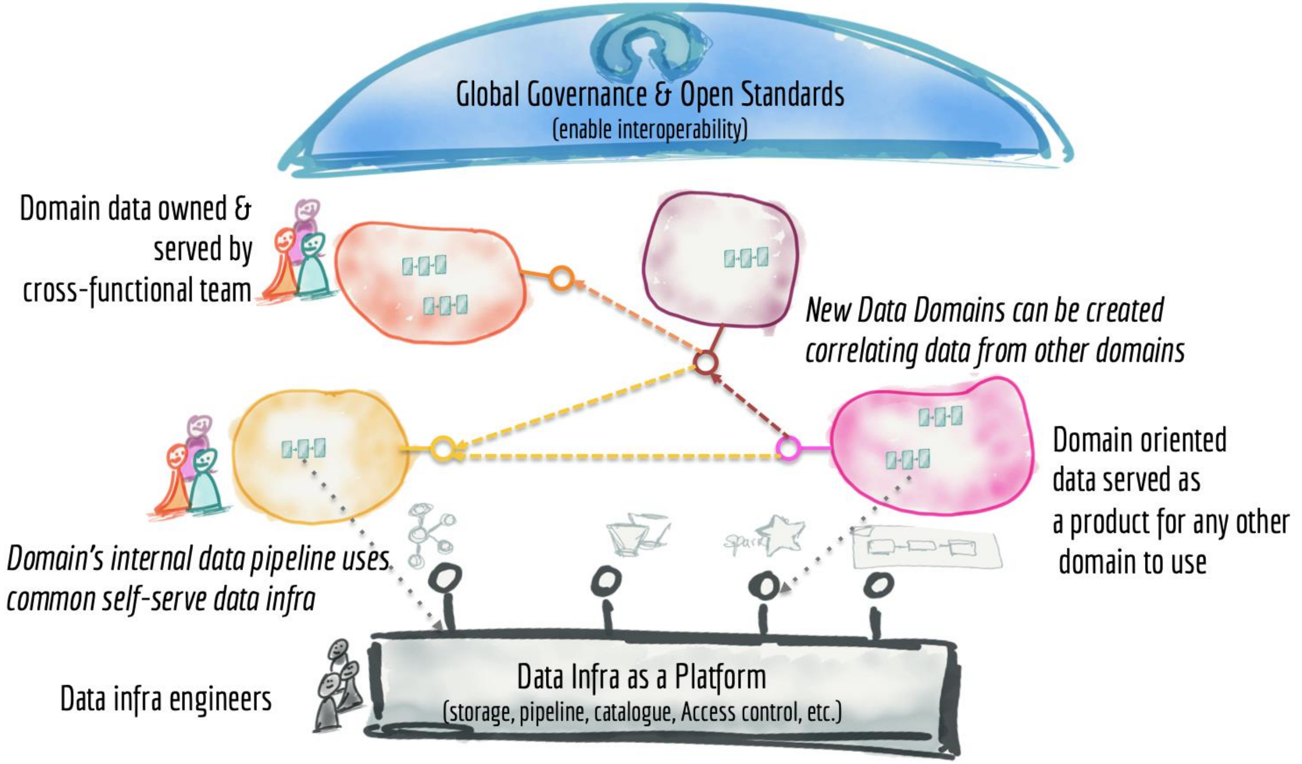
Clusters/SQL Endpoints

Centralized:

- Governance policies applied by a central team
- Production of data artifacts managed by a central team

Distributed:

- Domain driven production of data artifacts
- Entitlements on data owned by domain teams


Global Governance & Open Standards
(enable interoperability)
Domain data owned & served by cross-functional team
New Data Domains can be created correlating data from other domains
Domain oriented data served as a product for any other domain to use
Domain's internal data pipeline uses common self-serve data infra
Data infra engineers
Data Infra as a Platform
(storage, pipeline, catalogue, Access control, etc.)

Source:

https://martinfowler.com/articles/data-monolith-to-mesh.html#TheParadigmShiftTowardsADataMesh

DEMO TIME

SUMMARY

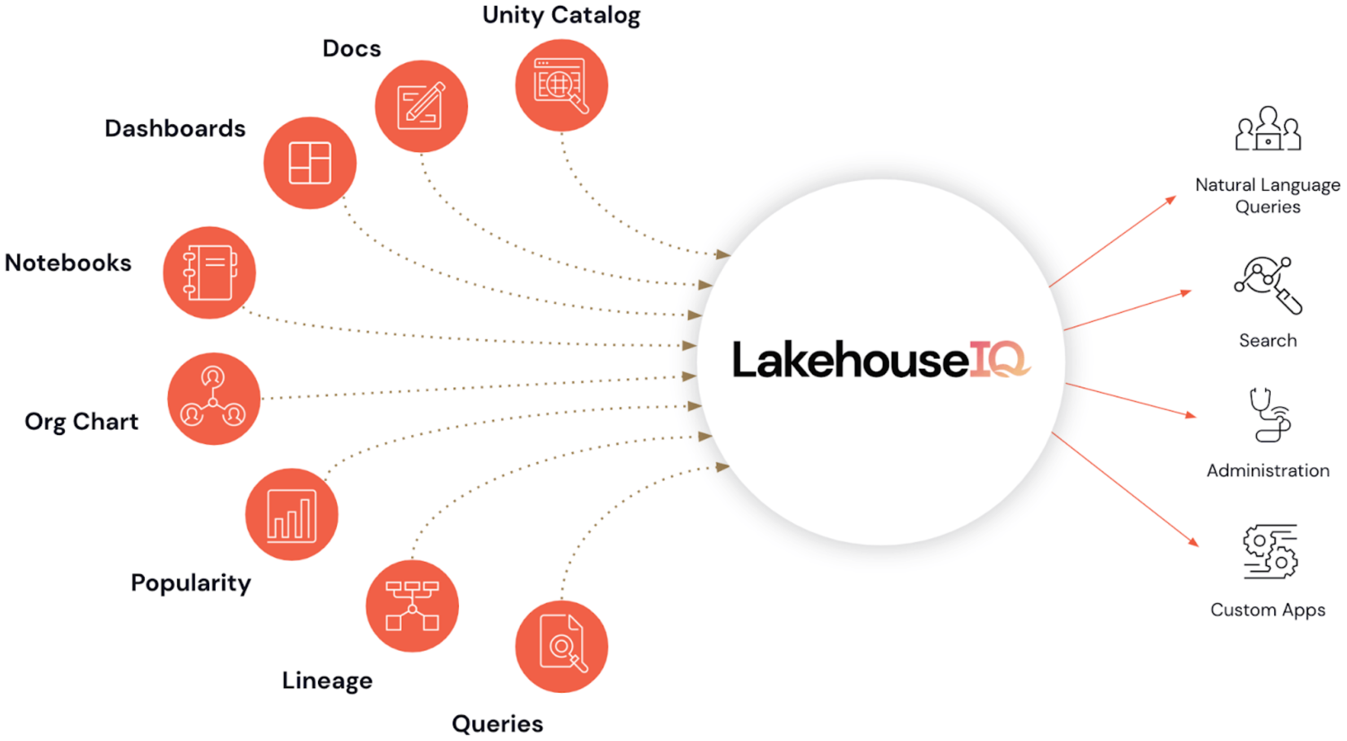
Do we really need Databricks Unity Catalog?

IMO : Yes

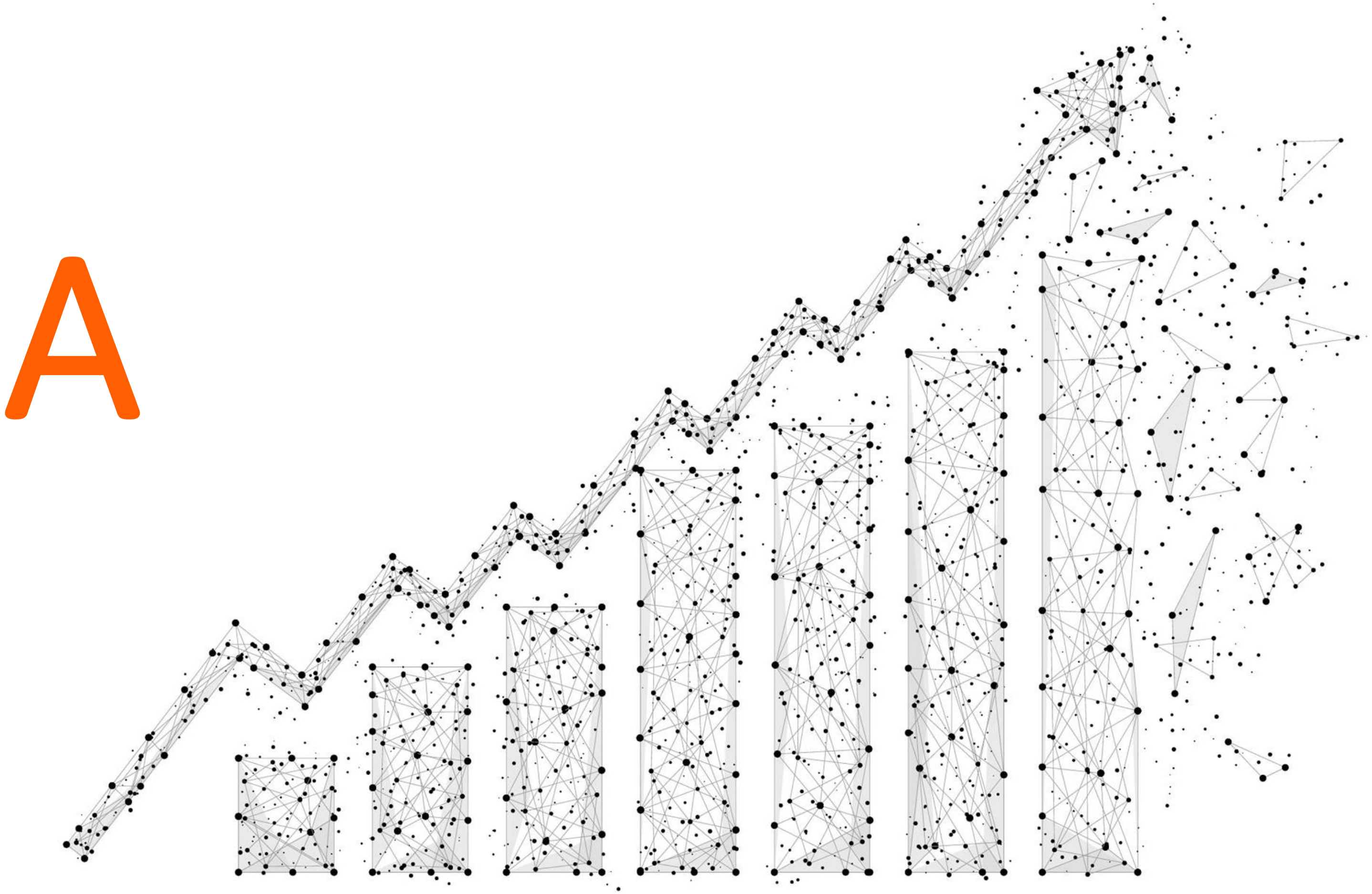
Why:

- Centralized access control
- Single permission model for our data (tables, files, models)
- Auditing
- Data Lineage
- Data Discovery
- Data Sharing
- Data Lakehouse Federation

- **What next:**
  - **Lakehouse IQ**
    - **Docs**
  - **Enzyme Engine**



Q & A



Resources

- https://www.databricks.com/product/unity-catalog
- https://www.databricks.com/resources/demos/tutorials?itm_data=demo_center
- https://learn.microsoft.com/en-us/azure/databricks/sql/language-manual/sql-ref-information-schema
- https://learn.microsoft.com/en-us/azure/databricks/administration-guide/system-tables/

THANK YOU!

tkrawczyk@future-processing.com
tomasz.k.krawczyk@gmail.com