

Unity Catalog Deep Dive

A practitioner's guide



Ifi Derekli

Field Eng. Manager / UC Specialist

ifi@databricks.com



Liran Bareket

Senior Specialist Solutions Architect

liran.bareket@databricks.com



Zeashan Pappa

Senior Product Specialist - Data Governance

zeashan@databricks.com

Product Safe Harbor Statement

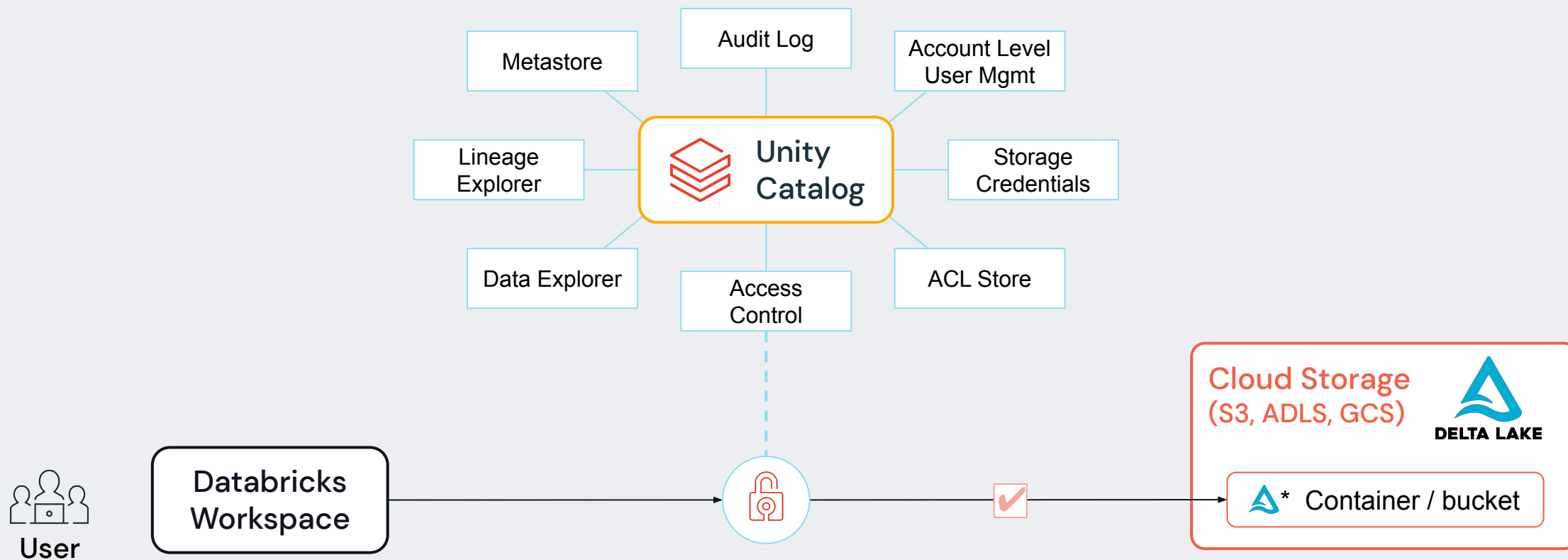
This information is provided to outline Databricks' general product direction and is for informational purposes only. Customers who purchase Databricks services should make their purchase decisions relying solely upon services, features, and functions that are currently available. Unreleased features or functionality described in forward-looking statements are subject to change at Databricks discretion and may not be delivered as planned or at all.

Agenda

- Upgrading **Users/Groups** to UC
 - Identity Federation
 - Roles & RACI Chart
- Upgrading **Metastores** to UC
 - Metastore Topologies
 - Managed/External Data Sources
- Upgrading **Workloads** to UC
 - Cluster policies
 - Job execution
- **Integrating** with UC
 - Using the REST API
 - Lessons from our Partner – Privacera

Unity Catalog - Recap

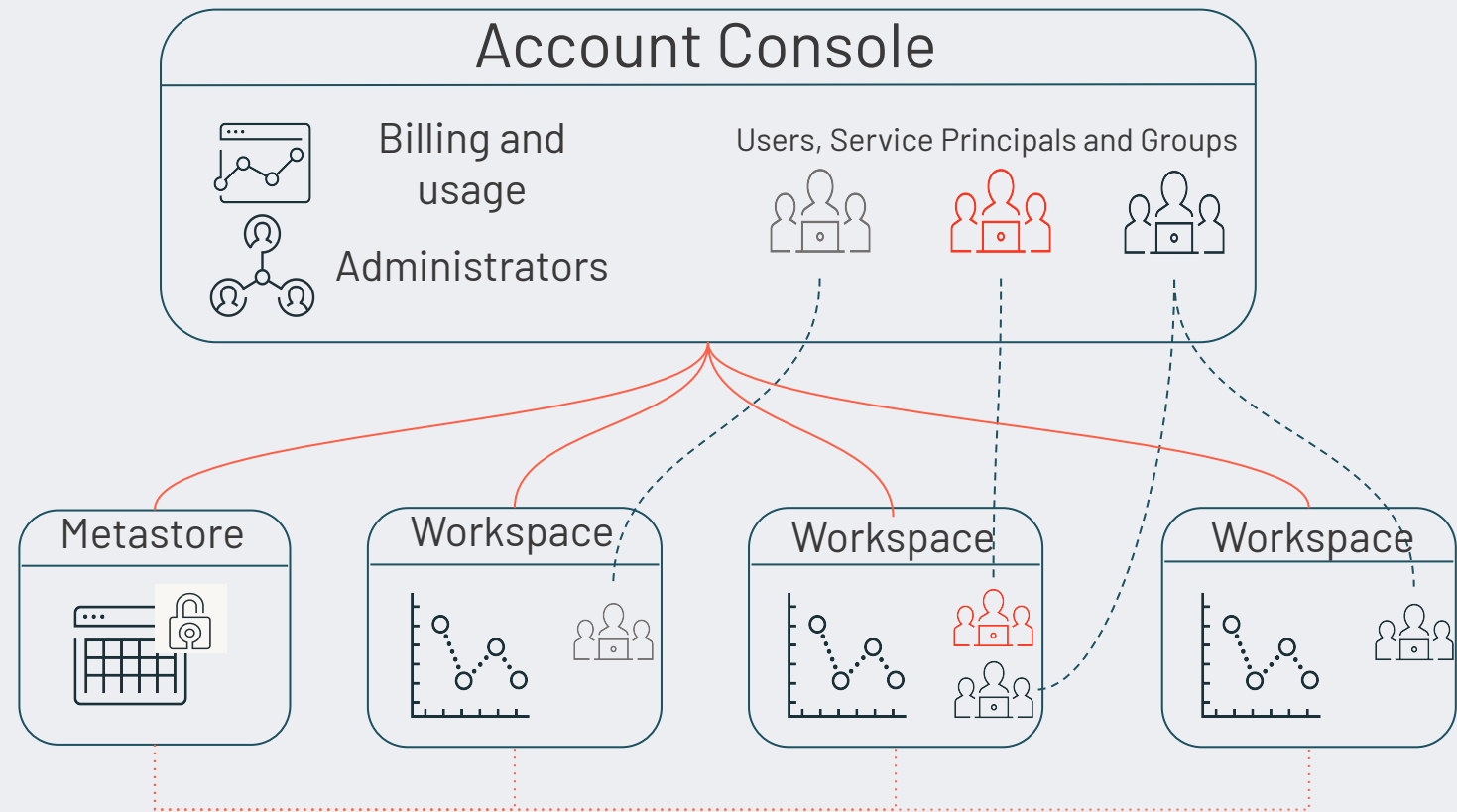
Unity Catalog – Architecture



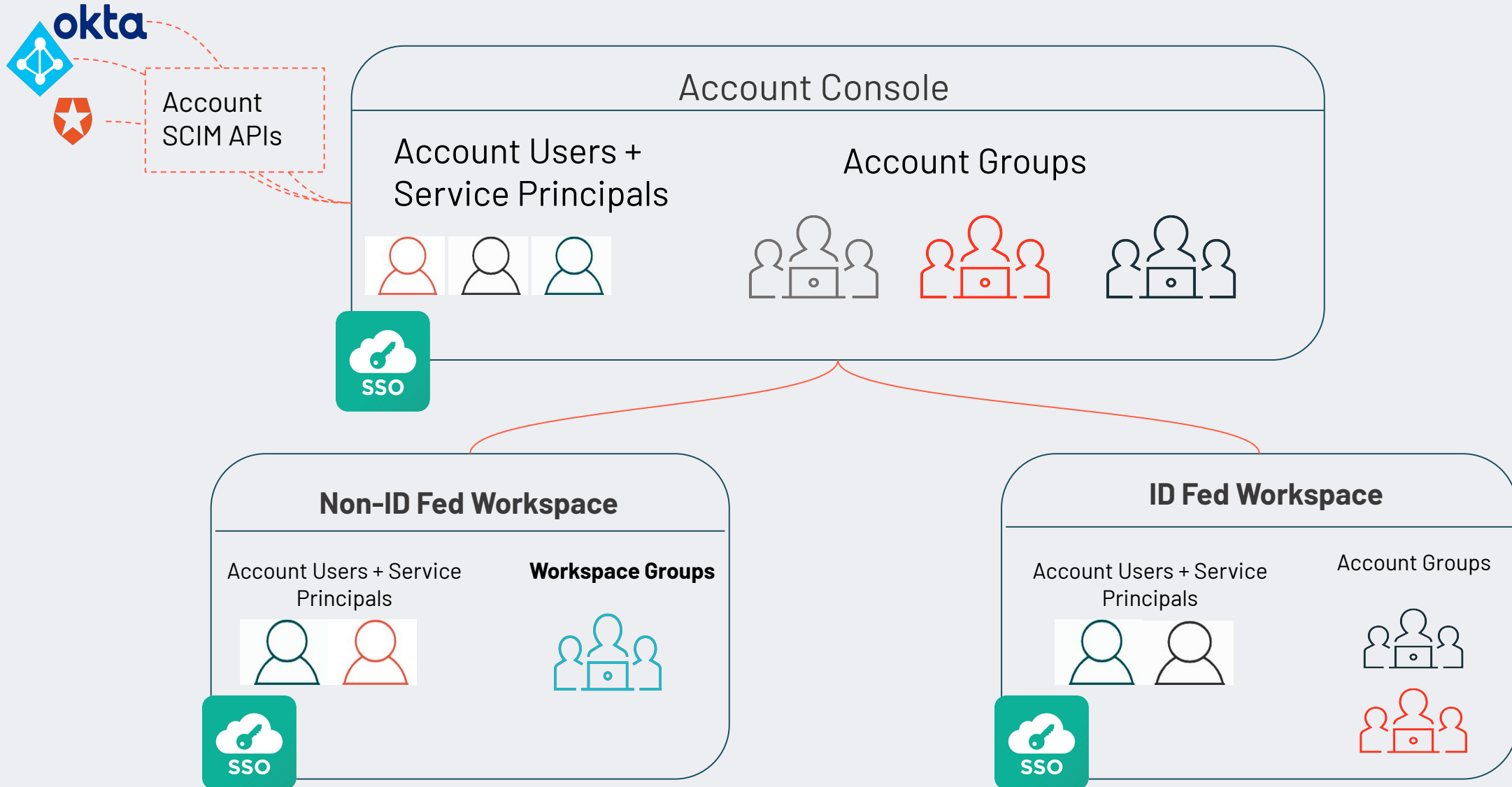
Upgrading Identity Management

Account and Workspaces

- Account
 - Typically, one per customer/cloud
 - Metastores (Metadata/ACL/Lineage)
 - Principals/Groups
- Workspaces
 - Multiple
 - Compute
 - Clusters
 - Endpoints
 - Workflows
 - Jobs
 - DLT



Identity Federation



Who can do what?

- **Account Admin** – Create Metastores, Workspaces, Manage Users
 - *NOTE: can effectively access all data*
- **Metastore Admin** – Can create catalogs
 - *NOTE: can effectively access all data in the metastore*
- **Workspace Admin** – Can create clusters, endpoints, manage users and groups within the workspace
- **Catalog/Database/Table Owner** – Can Assign access to other users
- **Account User** – Can Access a workspace, if assigned

Capabilities Chart



Data



Compute



Data and Compute

	Account Admin	Metastore Admin	Workspace Admin	Catalog, DB, TBL Owner	Account User
Create Metastores	Y	N	N	N	N
Manage Users and Groups, Assign Groups to Workspaces	Y	N	N	N	N
Create Workspaces, Assign Metastores To Workspace	Y	N	N	N	N
Create Clusters, Workflows, Delegate Access to compute	Y	N	Y	N	N
Create Catalog	Y	Y	N	N	N
Delegate Access to Data (Can Manage)	Y	Y	N	Y	N
Access Workspaces and Data	Y	Y	Y	Y	Y

Identity Onboarding Steps

- All UC workspaces use Identity Federation
- Identify Account Administrator (Azure)
- Enable SSO at the account console (OIDC/SAML)
 - Workspace SSO is still required
- Identify Business Groups for SCIM
- Enable SCIM for the Account Console
- Set up service principals for workflows (SPN/MI/Profiles)
- Assign users and groups to workspaces
 - Existing relationships will be maintained
- Test federation.

Upgrading Metastores

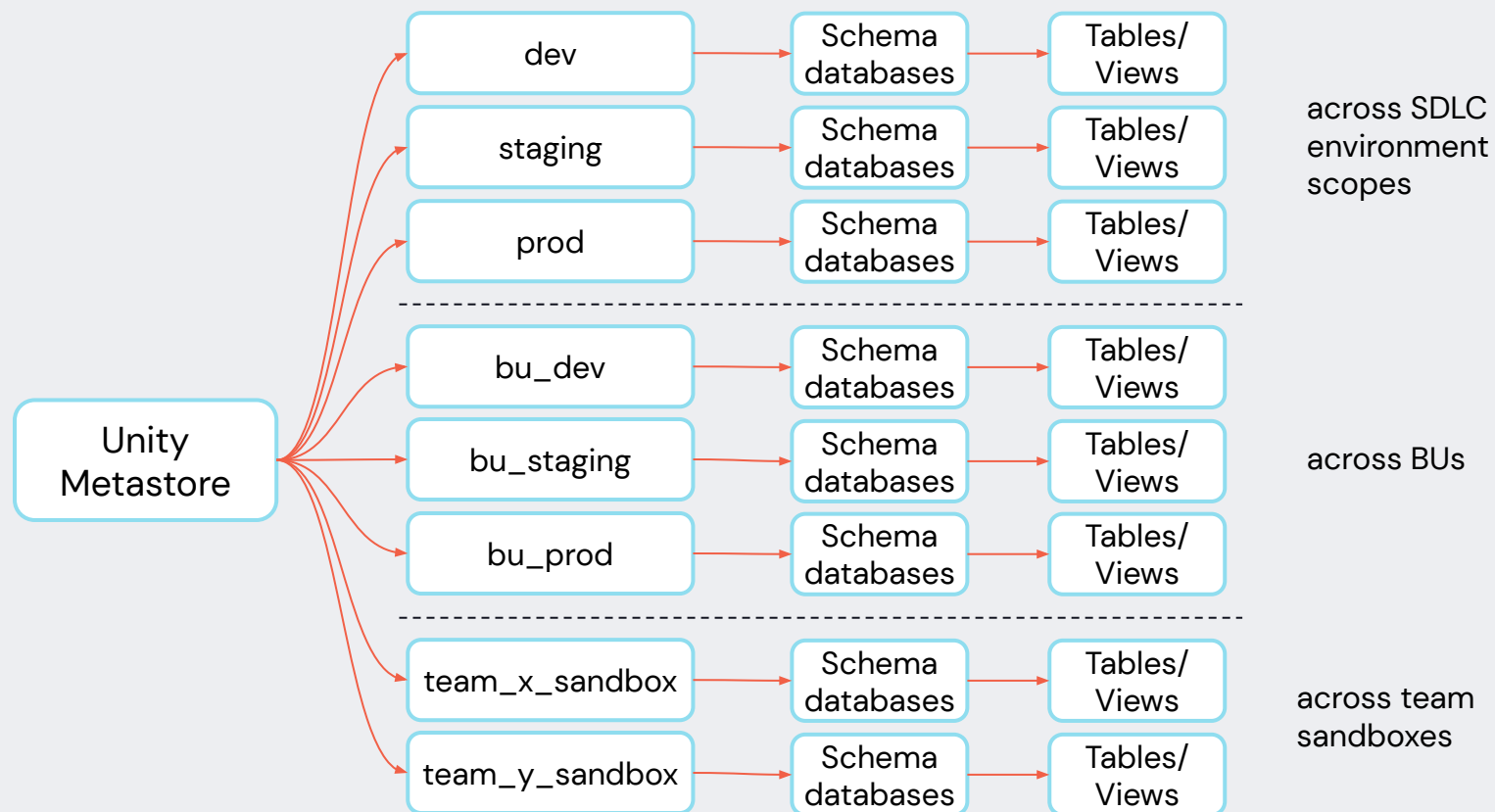
Definitions

- **Table / View** = collection of data, consists of columns & rows. LOGICAL
 - **Schema / Database** = collection of tables & views.
 - **Catalog** = collection of databases.
-
- **Metastore** = Physical implementation of metadata service. Collection of catalogs.
 - **Unity Catalog** = centralized security & governance service for your Lakehouse. Collection of metastores + ACLs + lineage + ... PHYSICAL

3-level namespace: `<catalog>.<schema>.<table>;`
E.g. `select * from dev.marketing.contacts;`

Catalog / schema / table setup

The catalog level of the 3-level namespace allows to structure databases and tables / views according to technical or business needs.

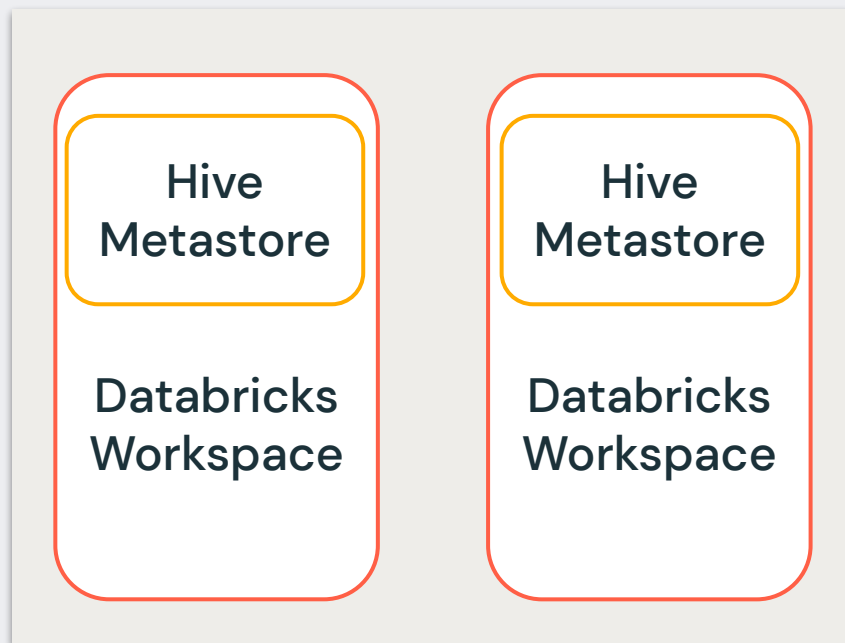


*Catalog+Schema owned by central team.
Usage Grants performed by central team
GRANT USAGE on <catalog>
GRANT USAGE, CREATE on <schema>*

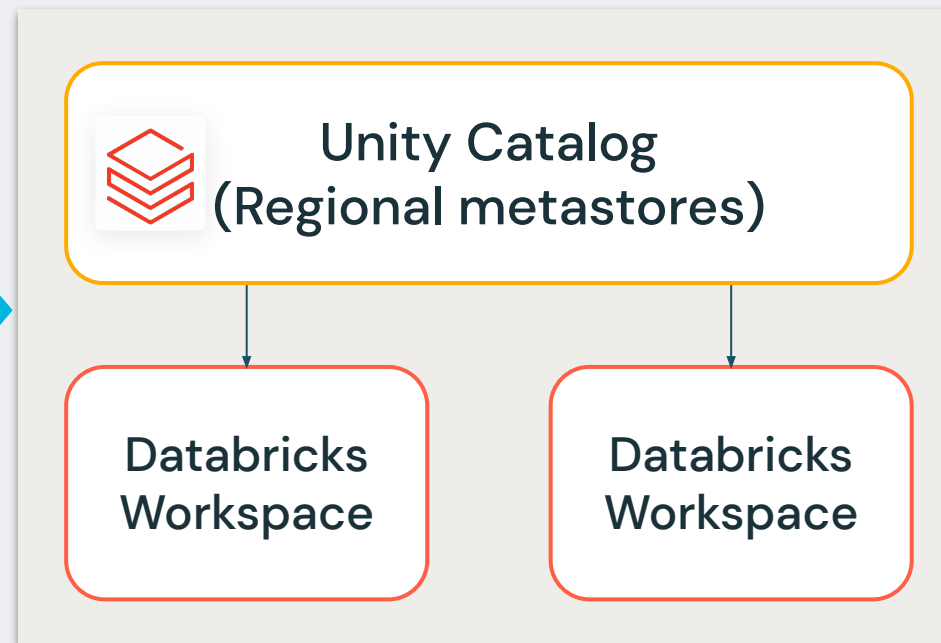
*Tables owned by team. Grants performed by teams X/Y.
Teams X, Y cannot share outside of team*

Topology: from Hive to Unity

Before UC



With UC

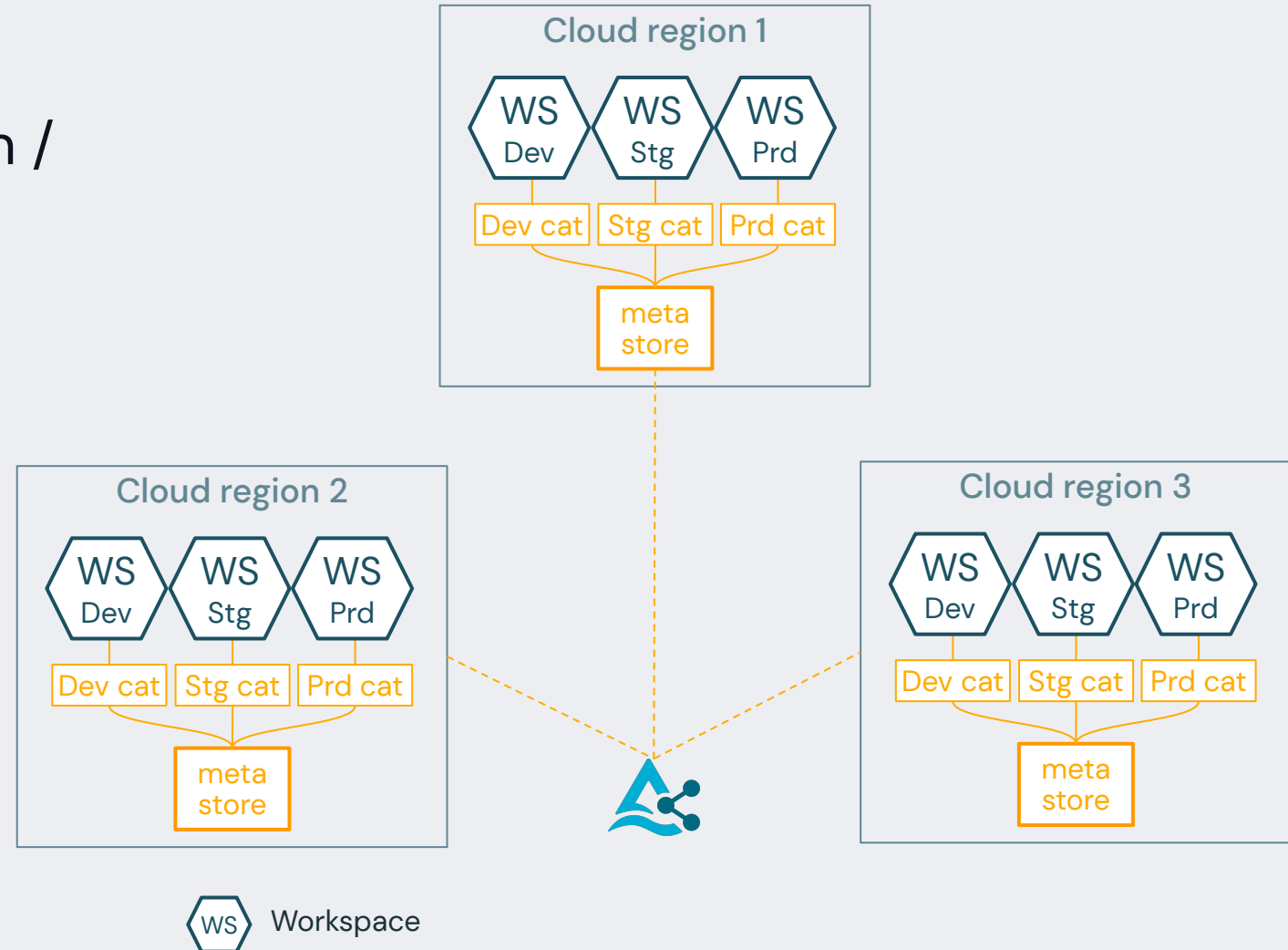


*How do I upgrade the metastore? Simply attach a Workspace to a Unity Metastore in the Account Console.
Hive_metastore becomes a catalog in the 3-level namespace.*

Topology: multi-region / multi-cloud UC

Powered by Delta Sharing

- Metastore boundary = region / cloud (due to latency, cost)
- *Use single region Metastore for all SDLC scopes and business units*
- *Use Databricks-to-Databricks Delta Sharing between cloud regions and cloud providers*



Let's talk about tables and cloud storage

What's the difference between *Managed* and *External* tables again?

	Managed	External
DROP TABLE	Deletes data	Does NOT delete data
Data location	Metastore's default S3/ADLS location	Custom S3 / ADLS location
Performance Optimizations	YES	NO
Management	Much simpler	More complex
Best For	Delta tables RECOMMENDED	1) R/W to data outside DB 2) Requirements of data isolation on infra-level 3) Non-Delta tables

Configuring your objectstore

- For your Metastore's Managed Location use a dedicated bucket / container that no other service/group/user has access to.
- For External Locations **do NOT mount** them on DBFS.

(otherwise...)



Upgrading Hive tables to Unity

External tables - use wizard

hive_metastore > vuong_nguyen_audit >

Upgrade tables to Unity Catalog

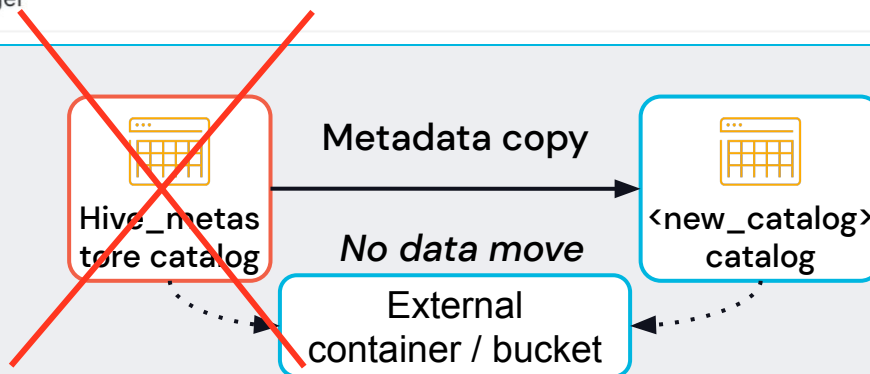
Shared Endpoint - Photon (M) ✓

1 Choose tables — 2 Choose destination — 3 Upgrade

Select the tables that you want to upgrade from Hive Metastore to Unity Catalog. Note that we currently only support the upgrade of external tables. Also, upgraded tables will still be available in the Hive Metastore.

i Make sure that you or your administrator has already created a storage credential and external location. [Learn more.](#) ×

<input type="checkbox"/> Table	Upgraded ⓘ	Type ⓘ	Format ⓘ
<input type="checkbox"/> vuong_nguyen_audit.accountbillableusage		EXTERNAL	delta
<input type="checkbox"/> vuong_nguyen_audit.accounts		EXTERNAL	delta
<input type="checkbox"/> vuong_nguyen_audit.accountsmanager		EXTERNAL	delta



Upgrading Hive tables to Unity

External tables - use SYNC command (coming soon)

- 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.
- Run multiple times idempotently

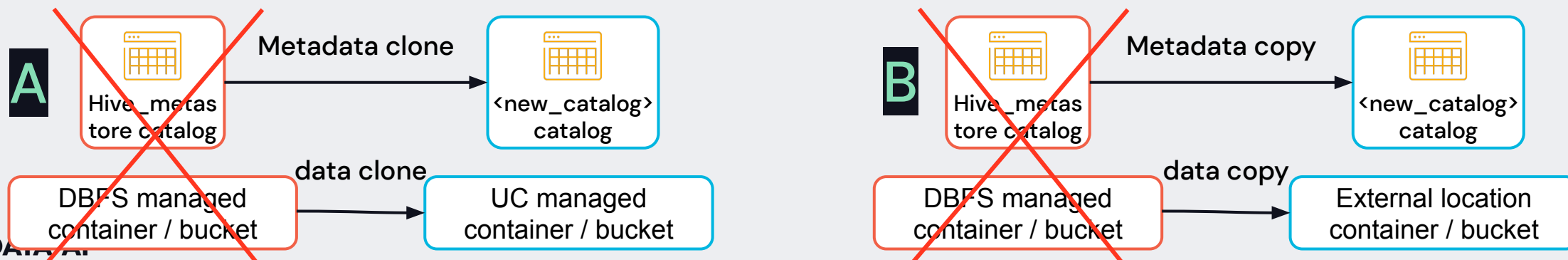
```
SYNC SCHEMA hive_metastore.my_db TO SCHEMA main.my_db_uc DRY RUN
```

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

Upgrading Hive tables to Unity

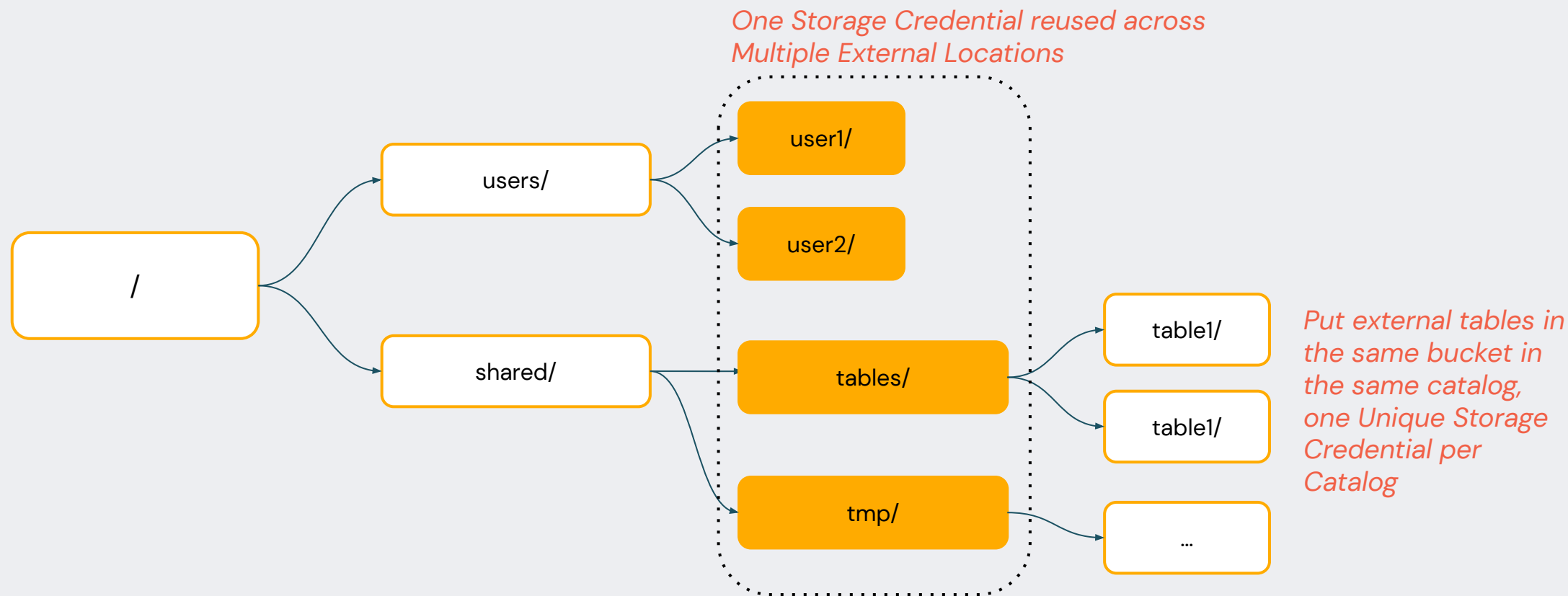
Managed tables - CTAS / CLONE (in the future, wizard)

```
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>;
```



Suggested external location structure

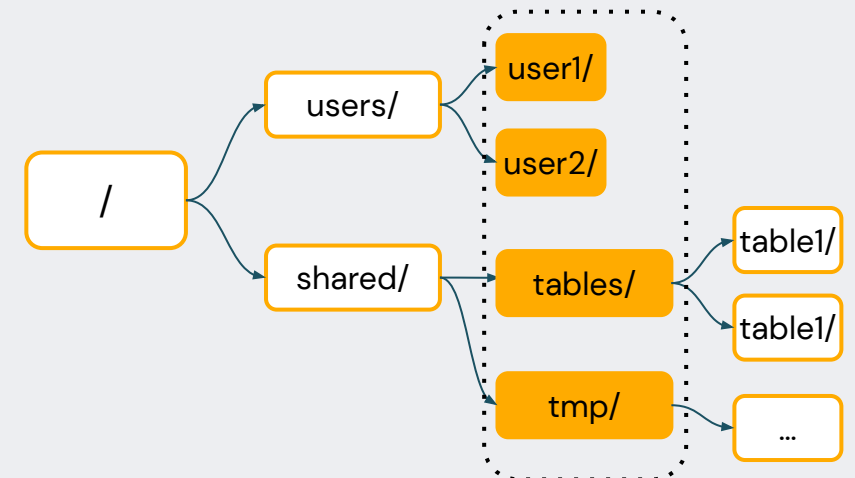
How to store and secure external data



Suggested external location structure

How to store and secure external data

- Personal directory for each user. Only user has access via UC
 - `CREATE EXTERNAL LOCATION user1loc URL 'abfss://cont@acct.dfs.core.windows.net/users/user1'`
`WITH (CREDENTIAL team_x_cred);`
 - `GRANT READ FILES, WRITE FILES ON EXTERNAL LOCATION user1loc TO `user1@company.com`;`
 - `GRANT CREATE TABLE ON EXTERNAL LOCATION user1loc TO `user1@company.com`;`
- Shared tmp directory for all users
 - `GRANT READ FILES, WRITE FILES ON EXTERNAL LOCATION tmp TO `team_x`;`
 - `GRANT CREATE TABLE ON EXTERNAL LOCATION tmp TO `team_x`;`
- **Best Practice:** Minimize # of Credentials and External Locations:
 - 1 cred / team or bucket
 - 1 location / team or user



Metastore recommendations: summary

- Metastore
 - Create single UC metastore per region per cloud.
 - Leverage Delta Sharing between regions and clouds.
 - Use catalogs to structure schemas & tables per business and technical needs (e.g. sandbox, dev/prod, BU)
- Tables
 - Use managed delta tables when possible
 - Use the Upgrade Wizard, SYNC, CTAS / CLONE, to upgrade tables
- Object Store
 - Configure managed & external object store locations securely
 - Do a role/access audit to ensure good governance
 - Structure your external locations smartly to minimize credential and location management

Upgrading Workloads

Unity-enabled clusters

Pre-create or leverage cluster policies

```
1 // Example Single-User Cluster Policy
2 { "spark_version": {
3   "type": "regex",
4   "pattern": "1[0-1]\\.[0-9].*",
5   "defaultValue": "10.4.x-scala2.12"
6 },
7 "data_security_mode": {
8   "type": "fixed", "value": "SINGLE_USER",
9   "hidden": true
10 },
11 "single_user_name": {
12   "type": "regex", "pattern": "(.*)",
13   "hidden": true
14 },
15 "Spark_conf.spark.databricks.
16   dataLineage.enabled": {
17   "type": "fixed",
18   "value": "true"
19 },
20 "Spark_conf.spark.databricks.sql.
21   initial.catalog.name": {
22   "type": "fixed",
23   "value": "hive_metastore"
24 }}
```

```
1 // Example Multi-User (user isolation) Policy
2 { "spark_version": {
3   "type": "regex",
4   "pattern": "1[0-1]\\.[0-9].*",
5   "defaultValue": "10.4.x-scala2.12"
6 },
7 "data_security_mode": {
8   "type": "fixed", "value": "USER_ISOLATION",
9   "hidden": true
10 },
11 "spark_conf.spark.databricks.unityCatalog.
12   userIsolation.python.preview": {
13   "type": "fixed", "value": "true"
14 },
15 "Spark_conf.spark.databricks.
16   dataLineage.enabled": {
17   "type": "fixed",
18   "value": "true"
19 },
20 "Spark_conf.spark.databricks.sql.
21   initial.catalog.name": {
22   "type": "fixed",
23   "value": "hive_metastore"
24 }}
```


Unity-enabled jobs

- Use SINGLE USER policy for JOB CLUSTERS
- Set a SERVICE PRINCIPAL as the OWNER of prod jobs and RUN as that SP
 - NOTE: Workspace Admins can change job ownership and by extension access data that service principals of the workspace can access
 - *Limit Workspace Admin role to required Dev Ops or IT Ops groups only*

Demo

Using the REST API

Automating access control management

- REST API provides full operational coverage for Unity Catalog CRUD Metastore/Catalog/Schema/ACL/Lineage
- Ability to integrate access control management to existing processes (jira, ServiceNow tickets, jenkins, etc)
- Case in point:  The Privacera logo, featuring a stylized 'P' icon in orange and blue followed by the word 'privacera' in a blue sans-serif font.

Integrating with Unity



Don Bosco Durai

Co-founder & CTO, Privacera

Privacera Integration with Unity Catalog

Translate Ranger Policy from YAML format to Unity Catalog JSON format



Apache Ranger

```
service: databricks_unity_catalog
resources:
  catalog:
    values:
      - sales_catalog
  schema:
    values:
      - sales_schema
  table:
    values:
      - sales_table
policyItems:
  - accesses:
      - type: Select
        isAllowed: true
  users:
    - emily.hope
```



Unity Catalog

```
https://your-uc-workspace.cloud.databricks.com/api/2.0/unity-catalog//permissions/table/sales_catalog.sales_schema.sales_table

{
  "privilege_assignments": [
    {
      "principal": "emily.hope@acme.com",
      "privileges": [
        "SELECT"
      ]
    }
  ]
}
```


Privacera and Unity Catalog Better Together

Privacera with Unity Catalog brings simpler governance across any data, any cloud

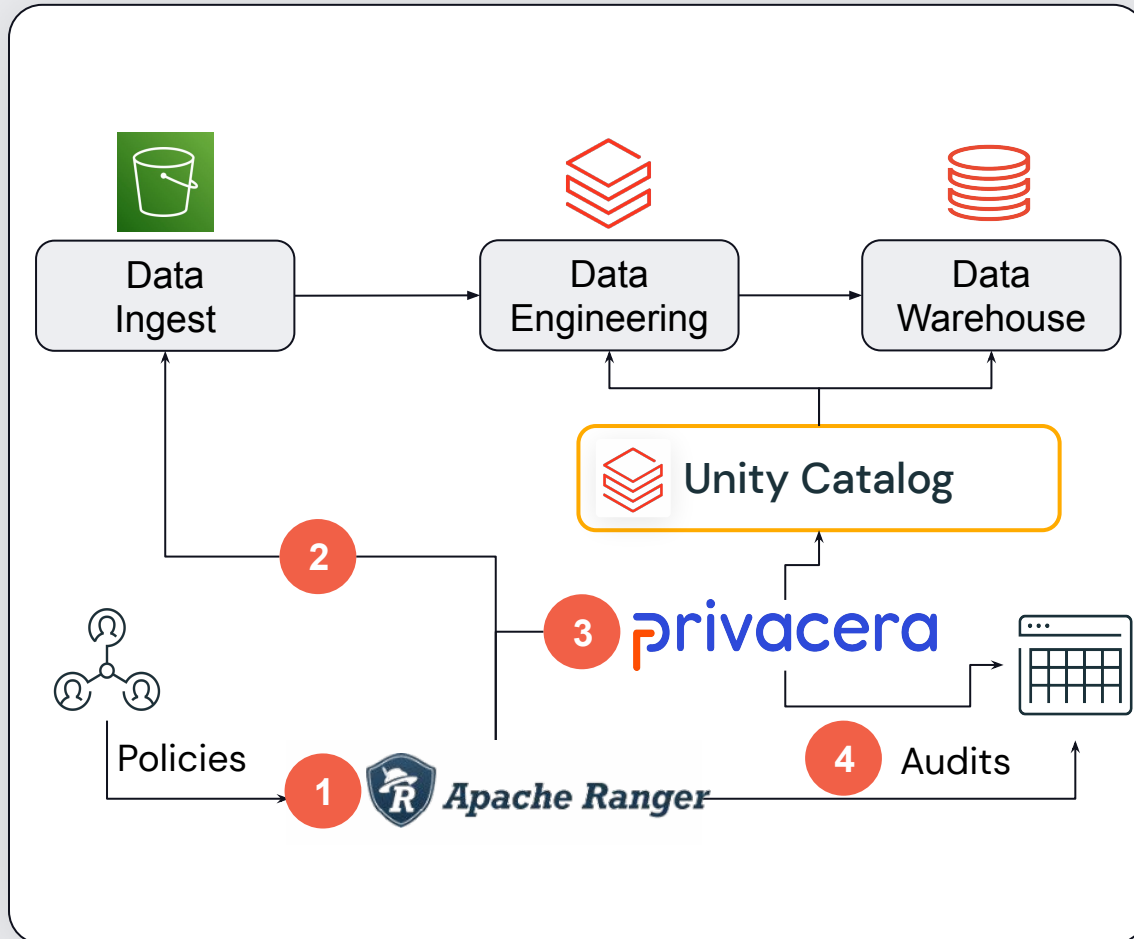
Privacera + Unity Catalog

- ✓ Data Governance across hybrid and multi-cloud
- ✓ Sensitive Data discovery, fine grained access management and encryption across any data
- ✓ Automated workflows to reduce data and user onboarding time
- ✓ Centralized auditing and canned reports for security and compliance

Unity Catalog

- ✓ Metadata and user management for lakehouse
- ✓ Access control and auditing for the lakehouse
- ✓ APIs to integrate with partner solutions

Flow – Unity Catalog/Ranger/Privacera



Steps

1. Pre-create Tag and Attribute based policies in Apache Ranger
2. Data is scanned and tagged during ingest and while tables are created
3. Privacera translates Ranger policies into native policies by calling Unity Catalog APIs
4. Privacera reads audit records generated by Unity Catalog and pushes it into Apache Ranger

Demo

For an integration deep
dive, please attend
tomorrow's session with
Bosco and Zeashan at 4pm -
MOSCONE SOUTH | UPPER
MEZZANINE | 152

How to Build a Complete Security and Governance Solution Using Unity Catalog

Wednesday, June 29 @4:00 PM

DATA+AI SUMMIT 2022

Thank you



Ifi Derekli

Field Eng. Manager / UC Specialist

ifi@databricks.com



Zeashan Pappa

Senior Product Specialist - Data Governance



Liran Bareket

Senior Specialist Solutions Architect

liran.bareket@databricks.com