DATA+AI SUMMIT BY databricks

LEARNING NEW TRICKS: UNILEVER'S DIVE INTO UNITY CATALOG

Roberto Flores, Head of Data Engineering, Unilever Europe

Athulya Ramamoorthy, Sr. Solution Architect, Databricks

NICE TO MEET YOU

INTRODUCTION

Athulya Ramamoorthy

Senior Solutions Architect @Databricks

INTRODUCTION

OUR SUPERSTARS

Unilever

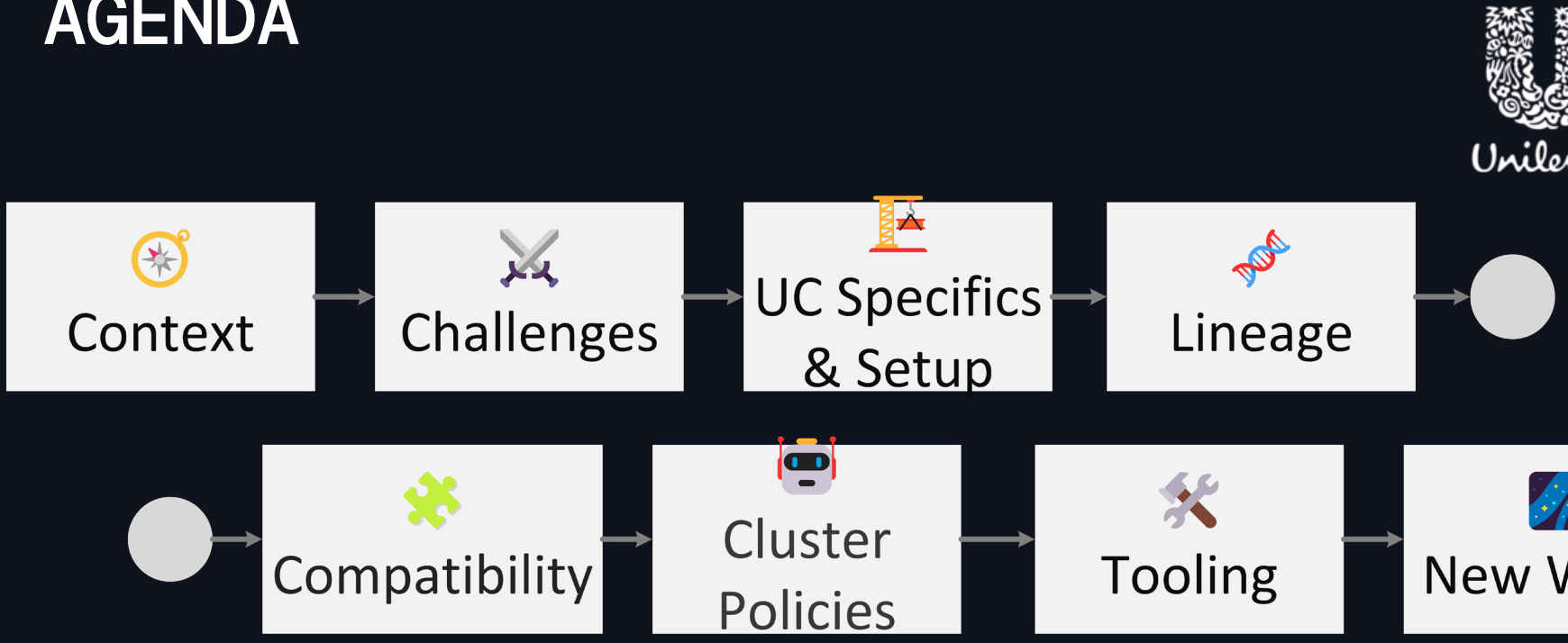
INTRODUCTION

Roberto Flores Meregote

Europe Head of Data Engineering @Unilever

AGENDA

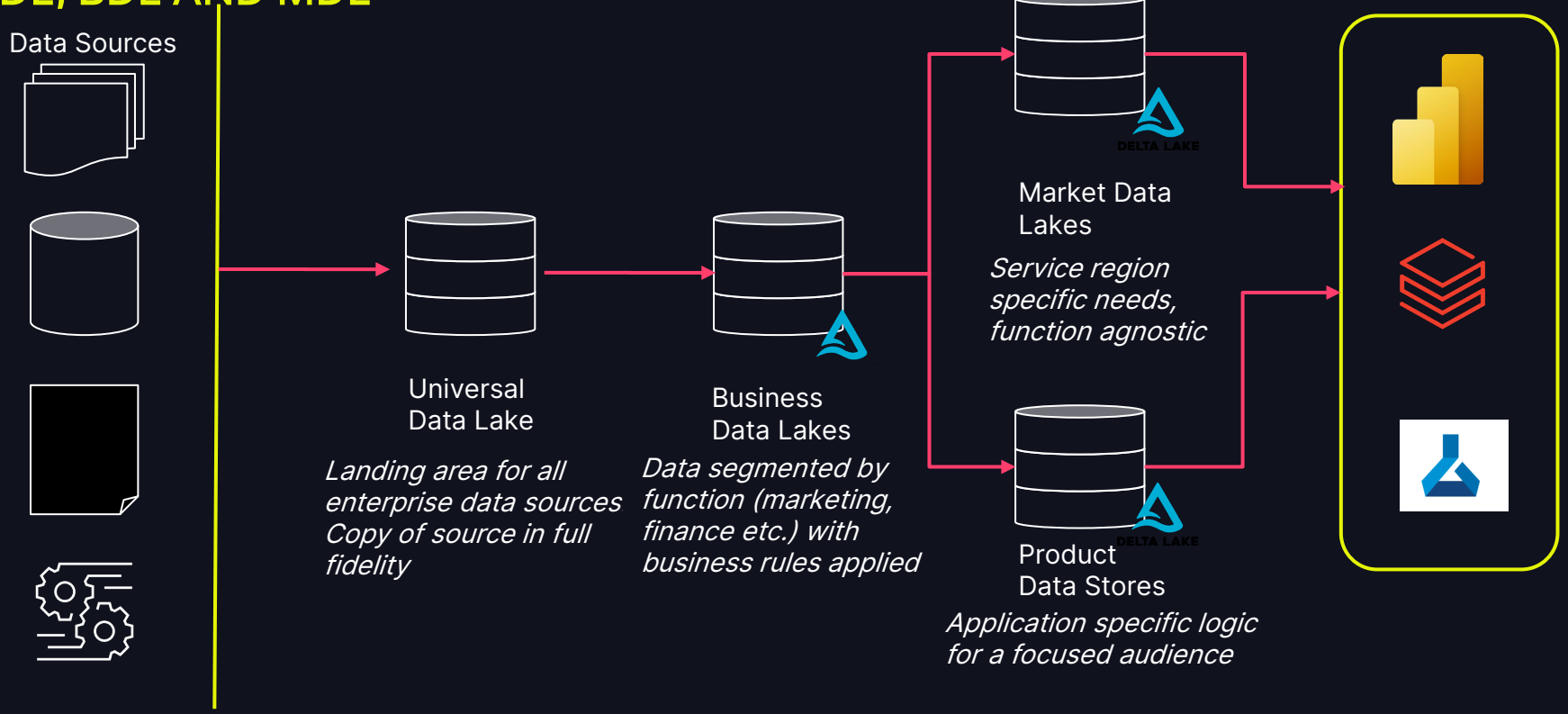
AGENDA



SOME BACKGROUND...

THE DATA LAKES

UDL, BDL AND MDL



UNILEVER'S DATA ESTATE

Sources

Internal
- SAP ECC
- SAP BW
- Teradata
- 100+ sources

Retailer/Purchased Data
- POS
- Product
- Store

Market Local Data Sources
- POS
- Product

Ingestion

Azure Databricks

Azure Data Factory

3rd Party Data Acquisition Platforms e.g; TradeEdge

Foundational Data Lake Systems

Universal Data Lake → Business Data Lakes

Raw → Staging → Curated

DELTA LAKE

Azure Databricks

Raw → Staging → Curated

DELTA LAKE

Market Data Lake

Azure Databricks

Market Data Lake

Semantic Layer

Databricks SQL Serverless

Azure Synapse Analytics

Global Products

Internal

Power BI

Machine Learning & Self Serve

Databricks Machine Learning

Azure ML

Azure Synapse Analytics

Data Distribution Layer

Unity Catalog

Azure Purview

DELTA LAKE

Market(s) Products

Power BI

Azure Analysis Services

Unity Catalog

Azure Active Directory

Azure Purview

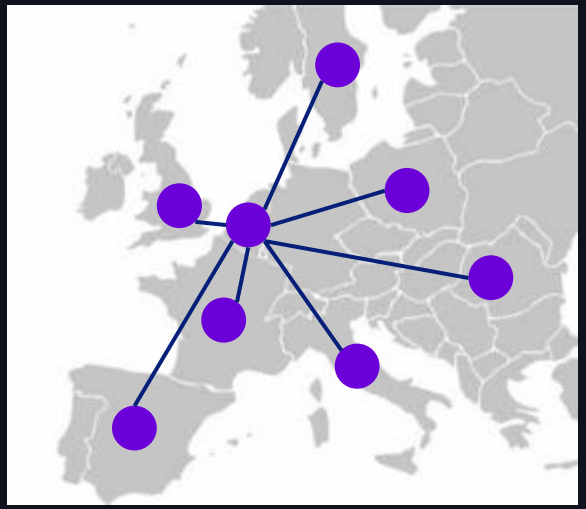
EUROPE MDL IS UNIQUE

SIMILAR IN SIZE TO NORTH AMERICA BUT WITH HIGH DATA COMPLEXITY

DATA COMPLEXITY		
		
UL Markets	2	**38**
UL BG Cells	10	**129**
Official Languages	1	**24**
Databases (external)	20	**316**

EUROPE MDL IS UNIQUE

PRIOR TO MDL THE REGION HAD DIFFERENT MATURITY LEVELS



Harmonized Data
Agility, Quality & Trust

- ✓ **One Source of Truth** for EUANZ enabling **pan-European performance** analysis and decision trade offs
- ✓ Processing **8bn+ rows of data** daily, with continuous quality checks
- ✓ **Shadow IT decommissioned**

WHAT DID THAT MEAN FOR OUR WAYS OF WORKING?

WHO'S DOING WHAT?

The Problem with Mount Points:

- Unauthorised access
- Limited auditing
- Data deletion risk

THE MYTH OF SECURITY

Access management on the Data Lake

- Dealing with folders and files
- Fine grained.. what?

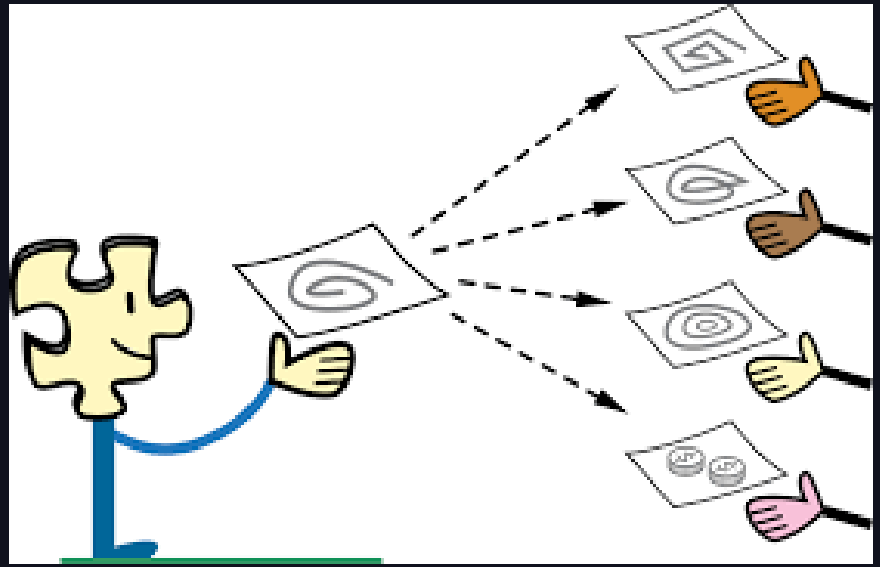
WE HAVE THE DATA, DON'T WE?

- Who owns what?
- What does the asset mean?
- How does it relate to other assets?

THE SHARING CONUNDRUM

Just how many copies of data is too many?

- Securing your data
- Maintaining the copies
- Ensuring Compliance



COMPLIANCE

Are all the copies secure?

Is all usage compliant?

Where is the exposure?

Is data usage tracked?

HOW WE APPROACHED UC ADOPTION

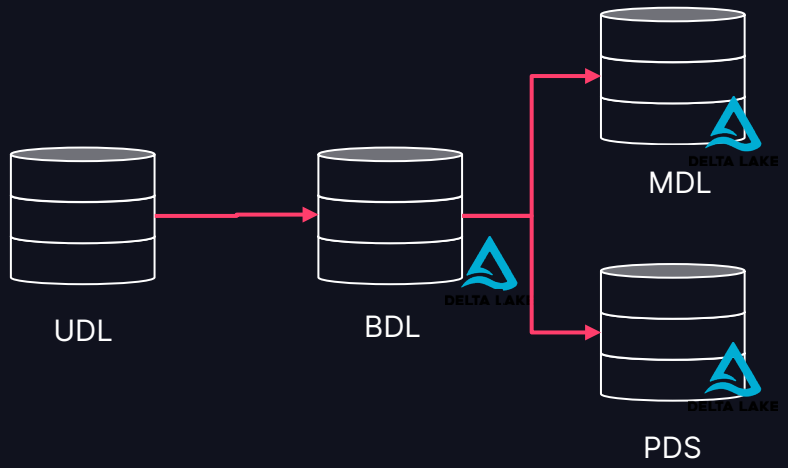
FAREWELL MOUNT POINTS

TABLES, VOLUMES & EXTERNAL LOCATIONS

- Catalogs and Schemas, data sits in External Locations:
  - EXTERNAL TABLES
    - Tabular datasets
    - CREATE EXTERNAL TABLE, READ FILES and WRITE FILES
  - EXTERNAL VOLUMES
    - Files in any format including structured, semi-structured or unstructured datasets
    - CREATE EXTERNAL VOLUME

CATALOG LAYOUT

CATALOGS (REMINDER OF LAYOUT)



CATALOG LAYOUT

CATALOG NOMENCLATURE

UDL

- UDL_dev
- UDL_qa
- UDL

BDL

- BDL_{topic}_dev
- BDL_{topic}_qa
- BDL_{topic}

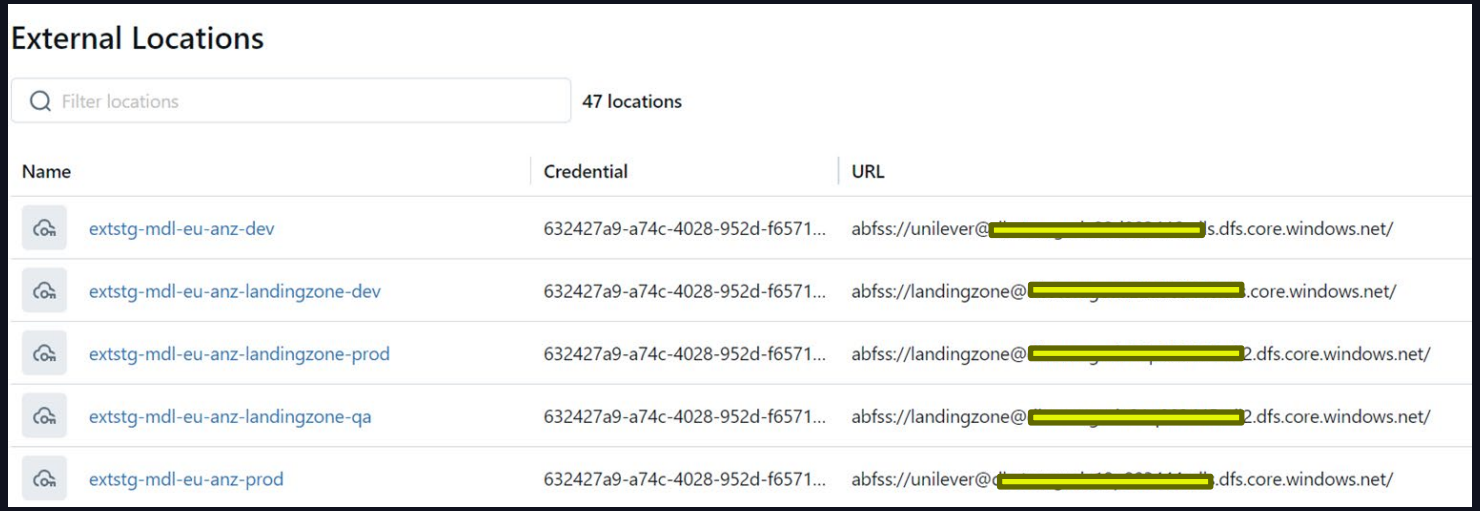
MDL

- MDL_{mkt}_dev
- MDL_{mkt}_qa
- MDL_{mkt}

FAREWELL MOUNT POINTS

EXTERNAL LOCATIONS

- Managed Storage Locations for Catalogs and Schemas

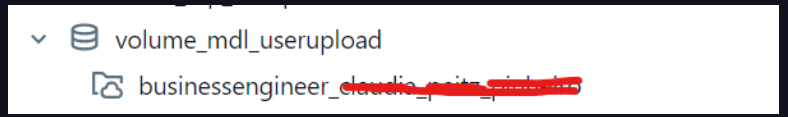


FAREWELL MOUNT POINTS

VOLUMES

- Managed Storage Locations for Catalogs and Schemas
  - In Unilever, a Catalog owner is able to create external volumes
  - Very useful for Self Service Business Analysts

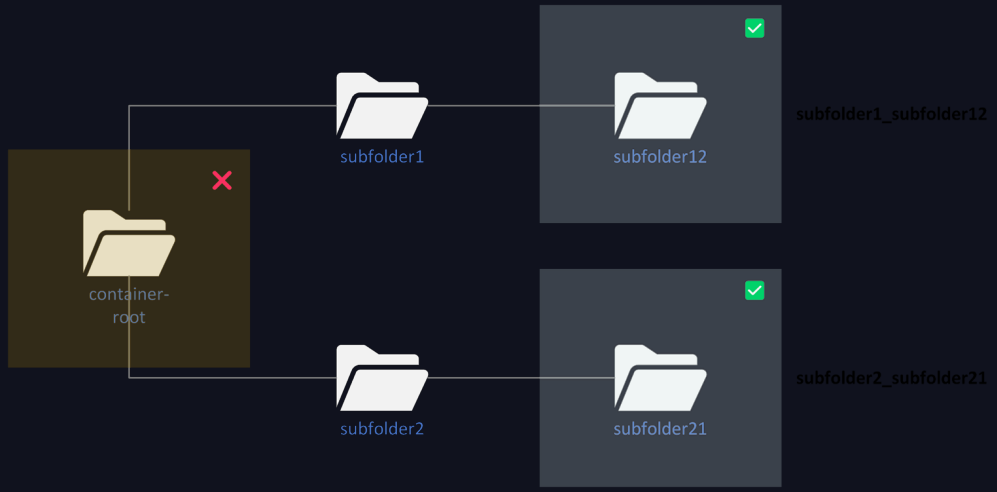
```
CREATE EXTERNAL VOLUME <catalog>.<schema>.<external-volume-name> LOCATION
'abfss://<container-name>@<storage-account>.dfs.core.windows.net/<path>/<directory>';
```



FAREWELL MOUNT POINTS

VOLUMES

- Volumes scopes must be mutually exclusive
  - For each specific location, pick a consistent level at which to declare volumes




```
> AnalysisException: [RequestId=cd416395-cb3d-49d0-a339-6f1c01ec2d45 ErrorClass=INVALID_PARAMETER_VALUE.LOCATION_OVERLAP] Input path url 'abfss://user-upload-area@dbstorageda22d903456adl2.dfs.core.windows.net/france/systemeu' overlaps with other external tables or volumes within 'CreateVolume' call. Conflicting tables/volumes: mdl_europe_anz_dev.volume_mdl_user upload.user_upload_area
```

PERMISSIONS APPROACH

CREATE

	DEV	QA	PROD
CATALOG	🌎 TDA	🌎 TDA	🌎 TDA
EXTERNAL LOCATION	🌎 TDA	🌎 TDA	🌎 TDA
EXTERNAL TABLES	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
EXTERNAL VOLUMES	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI

PERMISSIONS APPROACH

CREATE

	DEV	QA	PROD
SHARES	🌎 TDA	🌎 TDA	🌎 TDA
EXTERNAL RECIPIENTS	🌎 TDA	🌎 TDA	🌎 TDA
MODELS	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
FUNCTIONS	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI

PERMISSIONS APPROACH

APPLY TAGS, GRANT ACCESS

	DEV	QA	PROD
CATALOG	ENG / MSI	MSI	MSI
EXTERNAL LOCATION	ENG / MSI	MSI	MSI
EXTERNAL TABLES	ENG / MSI	MSI	MSI
EXTERNAL VOLUMES	ENG / MSI	MSI	MSI

PERMISSIONS APPROACH

WRITE

	DEV	QA	PROD
CATALOG	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
EXTERNAL LOCATION	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
EXTERNAL TABLES	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
EXTERNAL VOLUMES	😺 ENG / 🤖 MSI	🐶 USER / 🤖 MSI	🐶 USER / 🤖 MSI

PERMISSIONS APPROACH

READ, BROWSE

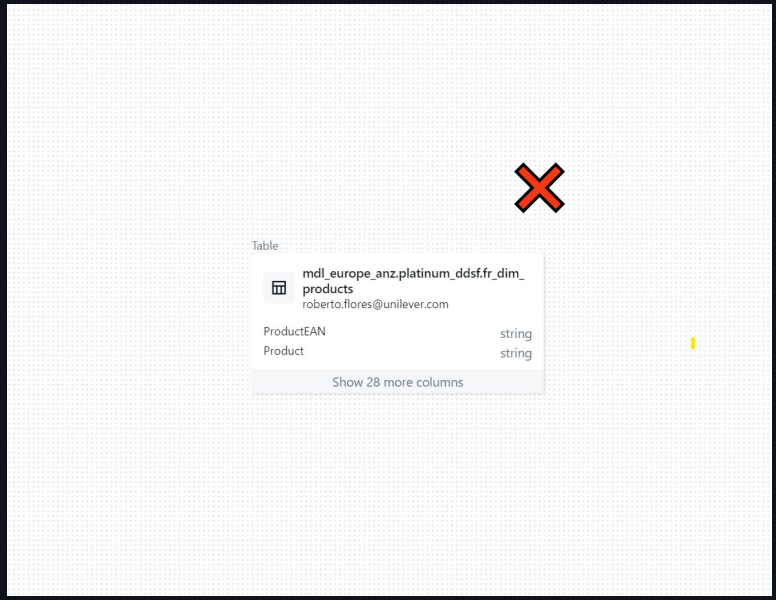
	DEV	QA	PROD
CATALOG	😺 ENG / 🤖 MSI	🤖 MSI	🐶 USER / 🤖 MSI
EXTERNAL LOCATION	😺 ENG / 🤖 MSI	🤖 MSI	🤖 MSI
EXTERNAL TABLES	😺 ENG / 🤖 MSI	🤖 MSI	🐶 USER / 🤖 MSI
EXTERNAL VOLUMES	😺 ENG / 🤖 MSI	🐶 USER / 🤖 MSI	🐶 USER / 🤖 MSI

MOVING OBJECTS INTO UC

TACTICAL CODE WILL MISS LINEAGE

Tactical declarations will get your tables into UC but will miss lineage:

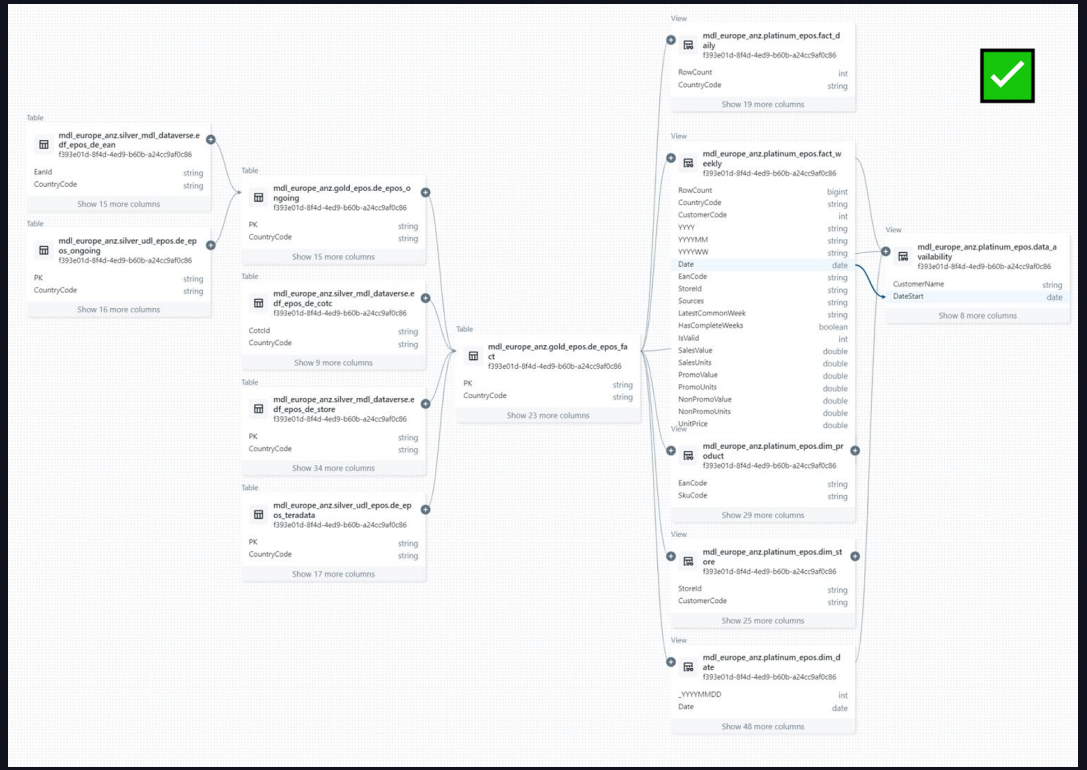
```
spark.sql(f"""CREATE TABLE IF NOT EXISTS
mdl_europe_anz{catalog_suffix}.platinum_ddsf.fr_dim_products USING
DELTA LOCATION
'abfss://unilever@{storage_name}.dfs.core.windows.net/MarketDataLa
ke/Platinum/SFO_bdl/platinum_fr_products'""")
```



MOVING OBJECTS INTO UC

REFACTORING CODE IS THE WAY TO GO

- Refactoring code is the way to go

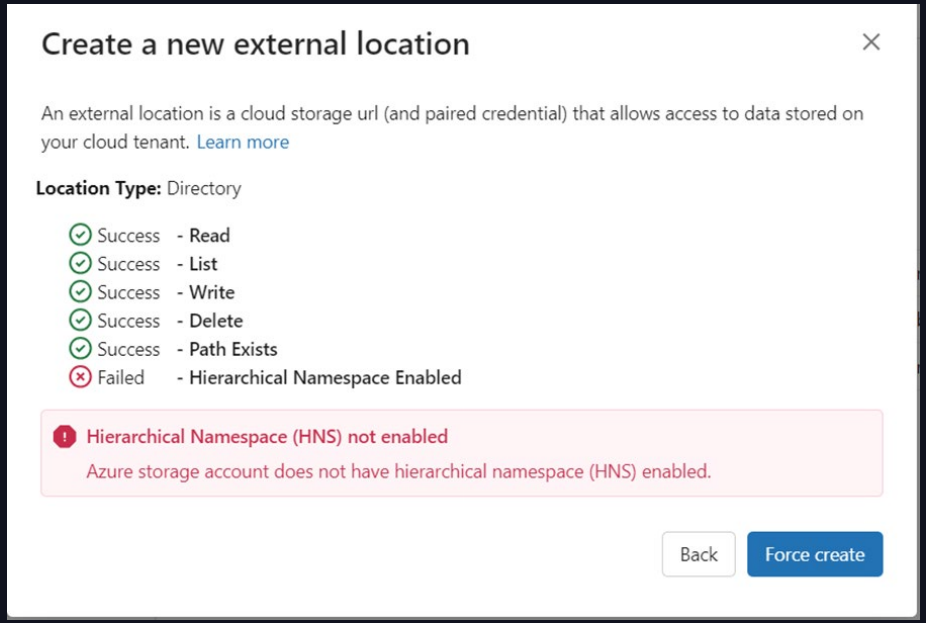


COMPATIBILITY - BLOB

BLOB STORAGE DOES NOT HAVE HIERARCHICAL NAMESPACE ENABLED WHICH PREVENTS EXTERNAL LOCATION CREATION

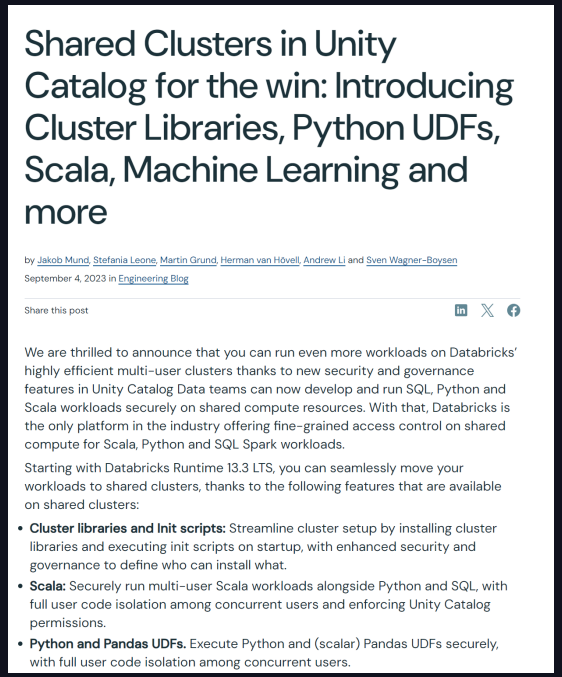
We had to switch from Traditional Blob Storage fully towards ADLS Gen2

Note: Process and context



COMPATIBILITY - SCALA

NOT ALL UC ENABLED CLUSTERS SUPPORTED SCALA
✅ FIXED AS PER SEPTEMBER 2023



Shared Clusters in Unity Catalog for the win: Introducing Cluster Libraries, Python UDFs, Scala, Machine Learning and more

by Jakob Mund, Stefania Leone, Martin Grund, Herman van Hövell, Andrew Li and Sven Wagner-Boysen

September 4, 2023 in Engineering Blog

Share this post

We are thrilled to announce that you can run even more workloads on Databricks' highly efficient multi-user clusters thanks to new security and governance features in Unity Catalog Data teams can now develop and run SQL, Python and Scala workloads securely on shared compute resources. With that, Databricks is the only platform in the industry offering fine-grained access control on shared compute for Scala, Python and SQL Spark workloads.

Starting with Databricks Runtime 13.3 LTS, you can seamlessly move your workloads to shared clusters, thanks to the following features that are available on shared clusters:

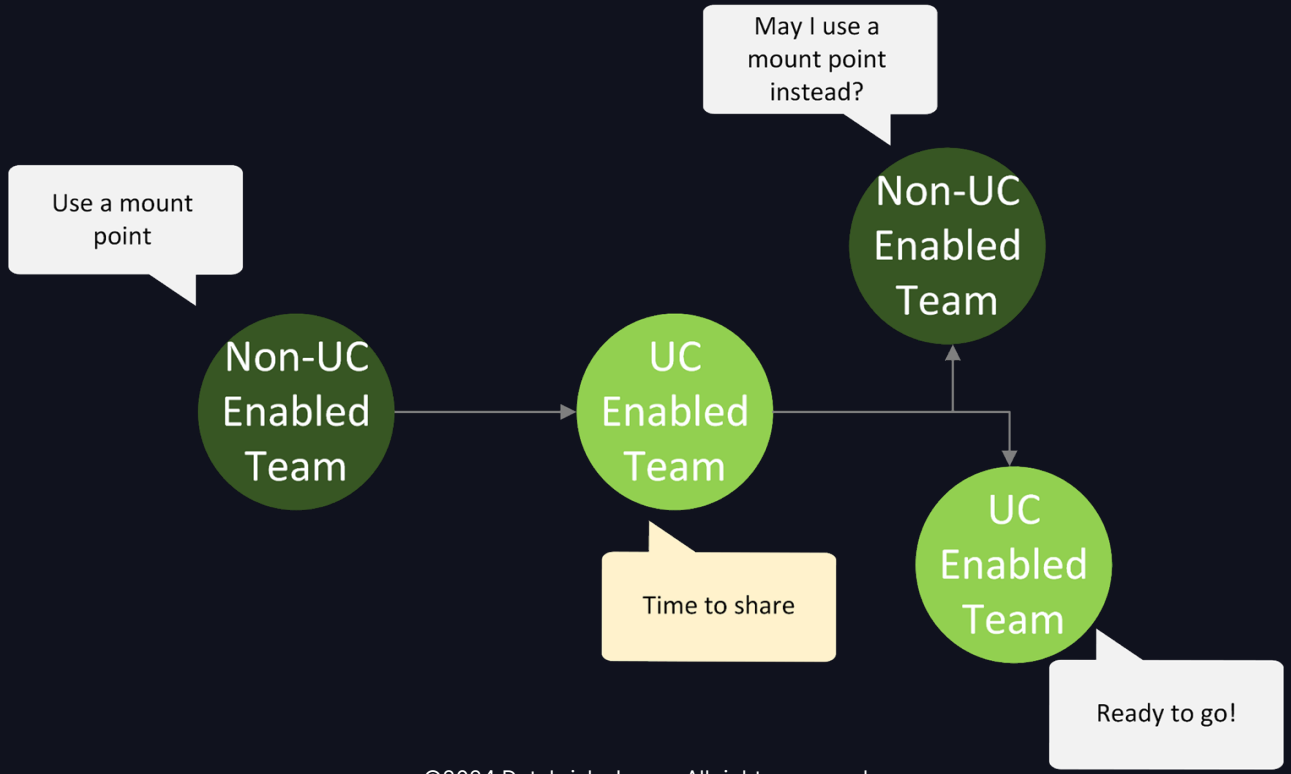
- **Cluster libraries and Init scripts:** Streamline cluster setup by installing cluster libraries and executing init scripts on startup, with enhanced security and governance to define who can install what.
- **Scala:** Securely run multi-user Scala workloads alongside Python and SQL, with full user code isolation among concurrent users and enforcing Unity Catalog permissions.
- **Python and Pandas UDFs.** Execute Python and (scalar) Pandas UDFs securely, with full user code isolation among concurrent users.

COMPATIBILITY - TEAMS

ONE EUROPE TEAM ONE CATALOG - FORCES US TO BE CONSISTENT

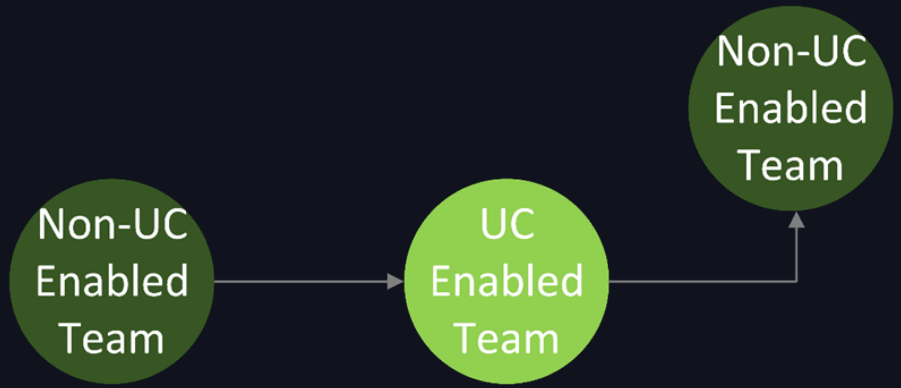
COMPATIBILITY - TEAMS

UC ADOPTION MIGHT NOT BE UNIFORM ACROSS A LARGE BUSINESS



COMPATIBILITY - TEAMS

UC ADOPTION MIGHT NOT BE UNIFORM ACROSS A LARGE BUSINESS



Upstream approach:
- Continue to consume objects as possible
- Offer help with DB Premium upgrade and transition

Downstream approach:
- Push to UC if they want access to your data
- Help with DB Premium upgrade and transition

COMPATIBILITY - CODE

RETIRING OLD CODE OR WAITING FOR UPSTREAM DEPENDENCIES

UC-enabled cluster on non-UC data

```
AnalysisException:
[UC_COMMAND_NOT_SUPPORTED.WITHOUT_RECOMMEN
DATION] The command(s): Creating a
persistent view that references both Unity
Catalog and Hive Metastore objects are not
supported in Unity Catalog.
```

Non-enabled cluster on UC data

```
AnalysisException: [UC_NOT_ENABLED] Unity
Catalog is not enabled on this cluster.
```

Diagnose error

COMPATIBILITY - CLUSTERS

PYTHON NOTEBOOKS

Non-UC Enabled Team → UC Enabled Team

❗: succeeds, not best practice
✅: succeeds

COMPATIBILITY - VIEWS

VIEWS ON DB-SQL: QUERYING NOT AN ISSUE, PERSISTING THE VIEW IS

Non-UC Enabled Team → UC Enabled Team

COMPATIBILITY - REGIONS

SPEAD ACROSS TWO AZURE REGIONS

CLUSTER POLICIES

EUROPE HAS 11 RESOURCE GROUPS, SPREAD BETWEEN 2 AZURE REGIONS

MDL

North Europe

- MDL_{mkt}_dev
- MDL_{mkt}_qa
- MDL_{mkt}

West Europe

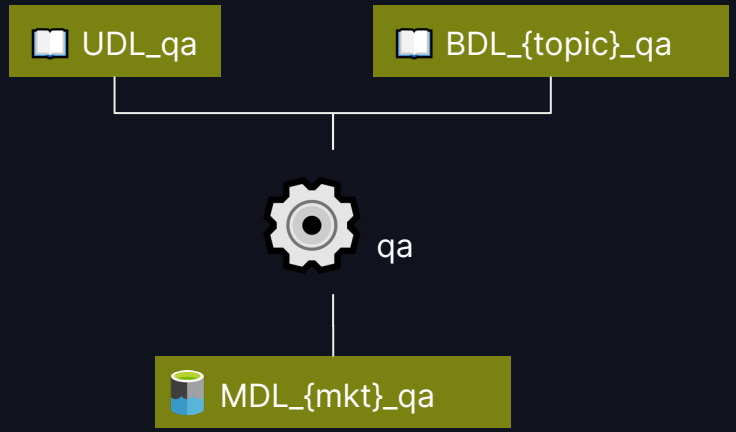
- MDL_{mkt}_dev
- MDL_{mkt}_qa
- MDL_{mkt}

What a script in any environment needs to do well:

1. Read from the right catalog
2. Write into the right external location
3. Declare into the right catalog

CLUSTER POLICIES

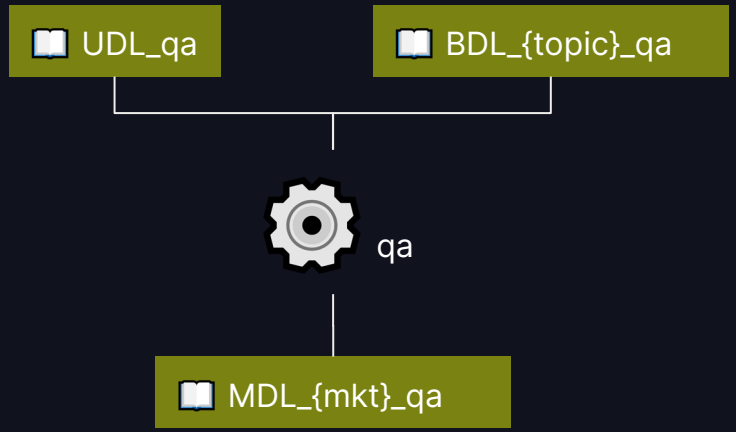
ENVIRONMENT VARIABLES




```
"spark_env_vars.AZ_DEVOPS_ORG_NAME": {
  "type": "fixed",
  "value": "[redacted]"
},
"spark_env_vars.AZ_TENANT_ID": {
  "type": "fixed",
  "value": "[redacted]"
},
"spark_env_vars.AZ_DEVOPS_FEED_NAME": {
  "type": "fixed",
  "value": "[redacted]"
},
"spark_env_vars.AZ_DEVOPS_SP_SECRET": {
  "type": "fixed",
  "value": "{{secrets/[redacted]}}"
},
"spark_env_vars.AZ_DEVOPS_SP_APP_ID": {
  "type": "fixed",
  "value": "{{secrets/[redacted]}}"
},
"spark_env_vars.ADLS_ROOT_PATH": {
  "type": "fixed",
  "value": "abfss://unilever@[redacted]/MarketDataLake"
},
"spark_env_vars.UNITY_CATALOGUE": {
  "type": "fixed",
  "value": "[redacted]_dev"
}
```

CLUSTER POLICIES

ENVIRONMENT VARIABLES

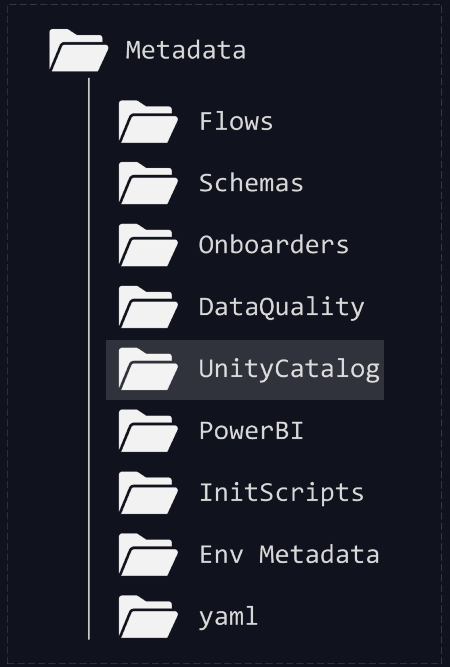


Python: reading cluster env variables

```
META_SECRET_SCOPE = os.environ.get('META_SECRET_SCOPE')
UNITY_CATALOGUE = os.environ.get('UNITY_CATALOGUE')
ADLS_ROOT_PATH = os.environ.get('ADLS_ROOT_PATH')
UDL_ROOT_PATH = os.environ.get('UDL_ROOT_PATH')
BDL_{topic}_ROOT_PATH = os.environ.get('BDL_{topic}_ROOT_PATH')
```

TOOLING

UNITY CATALOG METADATA/ACCESS FOR SCHEMAS, TABLES, VOLUMES

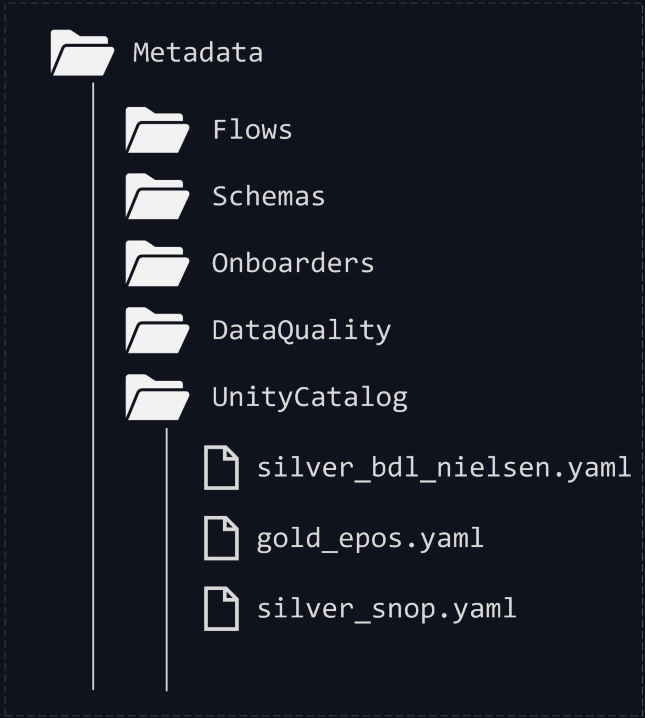


Sample: UnityCatalogDefinition_dev.yaml

```yaml
UC_Def:
- flow_name: Init_schema
  schemas:
  - name: init_schema
    comment: "This is just a test file to init the correct schema"
    tags:
    - Test
    access:
      - acls_group_name: "SEC-ES-DA-p-903444-europe-analyst"
        acls_access_type: "SELECT, MODIFY"
        acls_access_catalogue: "mdl_europe_anz_dev"
    tables:
    - name: init_she
      comment: "### Supports Markdown 1. First item 2. Second item"
      tags:
      - PII
      columns:
      - name: "Country_Code"
        comment: "Country code key as of 07/03/2024"
        tags:
          - PK
(...)
```

TOOLING

ALLOWS MARKDOWN ENTRY FOR METADATA AT DIFFERENT LEVELS



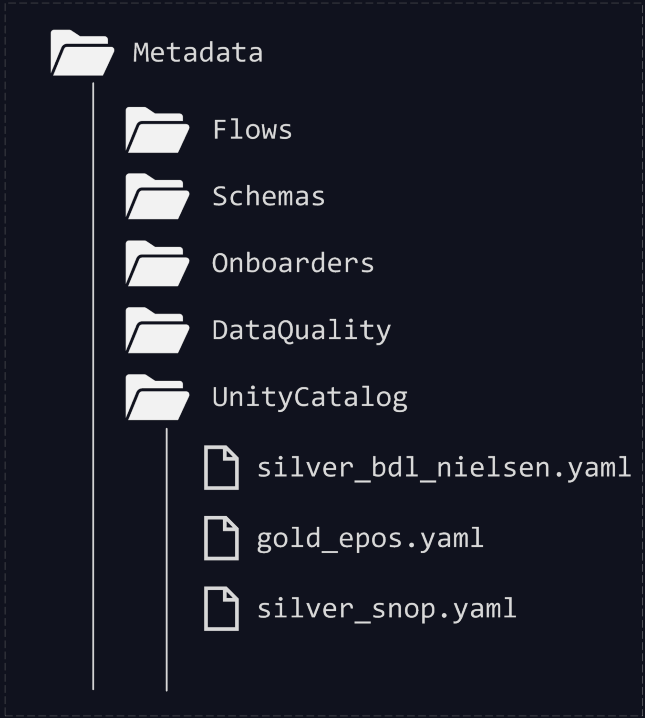
Access:

Allows us to set access policy for each AD group

1. Can differ across environments (DEV, QA, PROD)
2. Can be at Schema, Table, Volume level

TOOLING

ALLOWS MARKDOWN ENTRY FOR METADATA AT DIFFERENT LEVELS

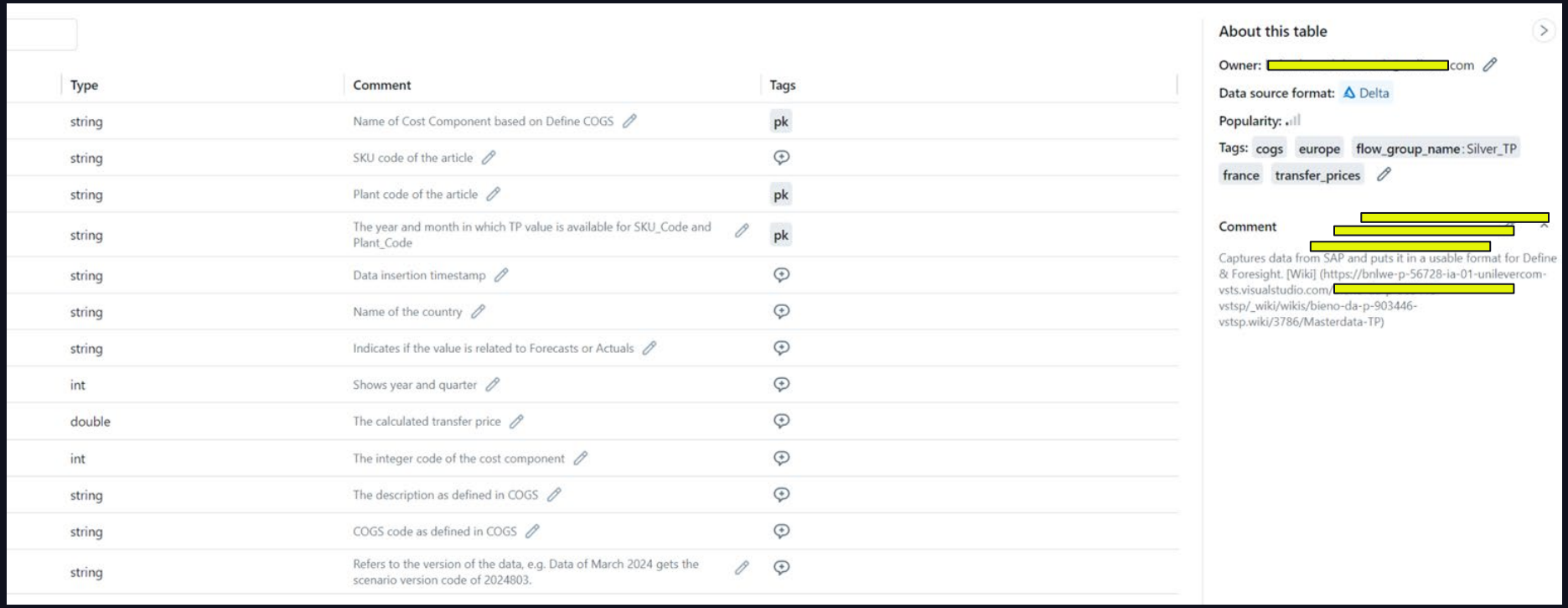


Comments and Tags:

1. Markdown enabled where supported
2. Can applied at Schema, Table, Volume level

TOOLING

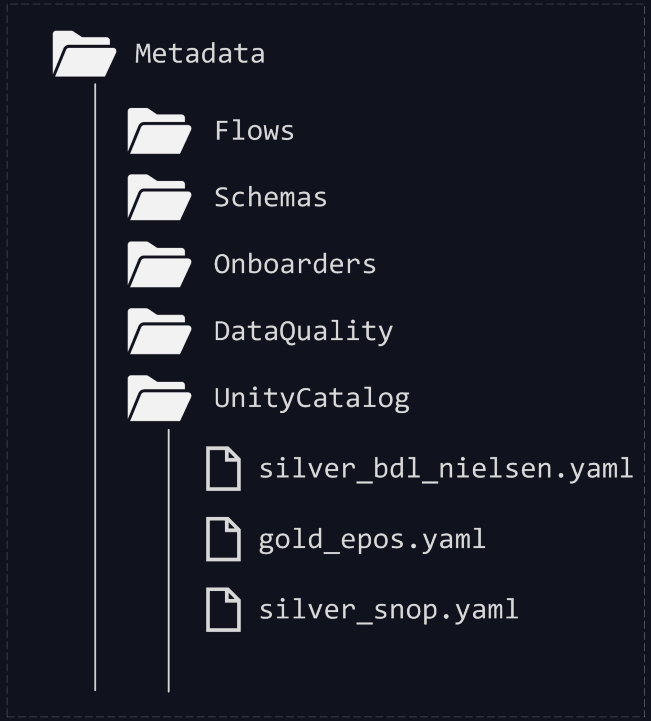
BIG PICTURE: THIS ENABLES LAKEHOUSE IQ



Type	Comment	Tags
string	Name of Cost Component based on Define COGS	pk
string	SKU code of the article	
string	Plant code of the article	pk
string	The year and month in which TP value is available for SKU_Code and Plant_Code	pk
string	Data insertion timestamp	
string	Name of the country	
string	Indicates if the value is related to Forecasts or Actuals	
int	Shows year and quarter	
double	The calculated transfer price	
int	The integer code of the cost component	
string	The description as defined in COGS	
string	COGS code as defined in COGS	
string	Refers to the version of the data, e.g. Data of March 2024 gets the scenario version code of 2024803.	

TOOLING

WIP: MASKING AND AES, PROPER REGION MIRRORING



WIP:

1. Field Masking
2. AES_Encryption views

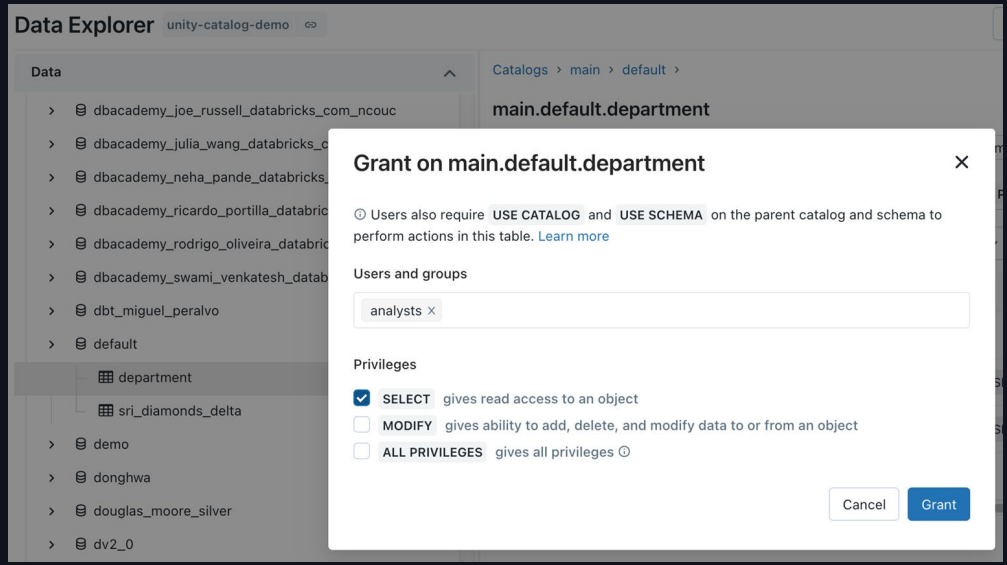
HOW WE'RE NOW LEARNING NEW TRICKS

A UNIFIED TEAM

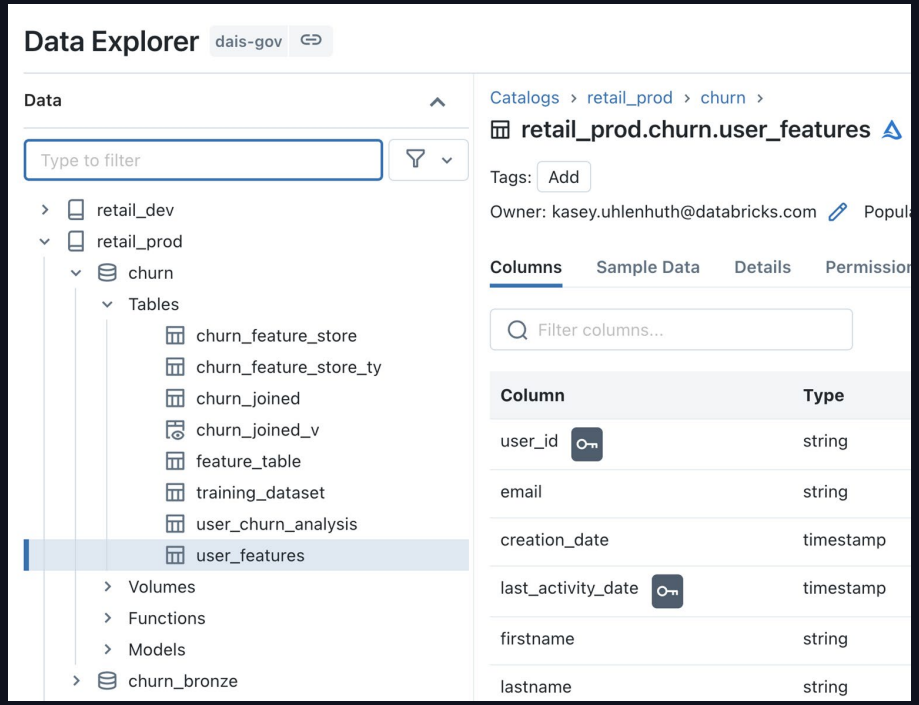
- Silos broken down
- Environment unification
- Centralised data assets for collaboration

IMPROVED ACCESS MANAGEMENT

- Access via a single permission model
- Fine Grained Access
- Workspace Bindings for segregation



GETTING TO DATA QUICKER



- Discover and explore data centrally
- Single point for permissions and management of all data
- Streamline resources & cost

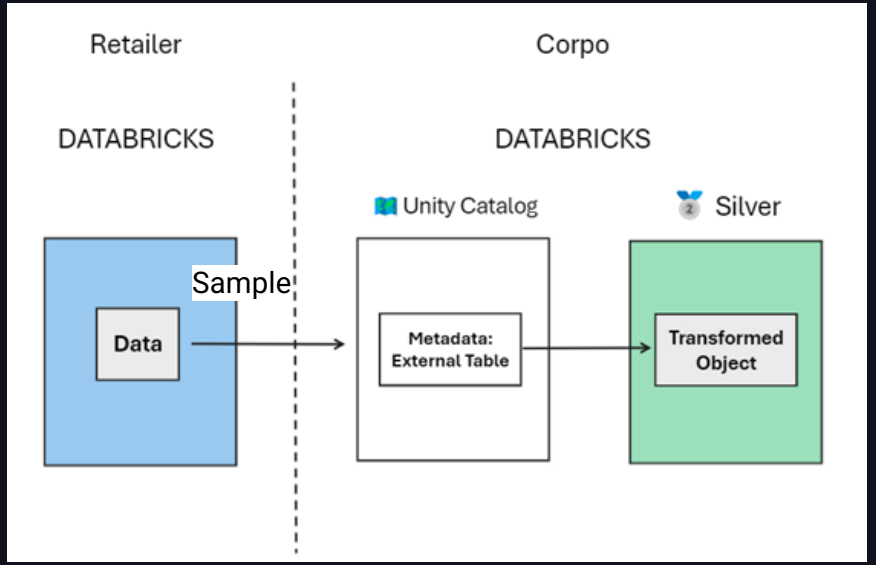
DATA FEDERATION

- Access to data at source
- Access instantly vs. weeks

DATA SHARING

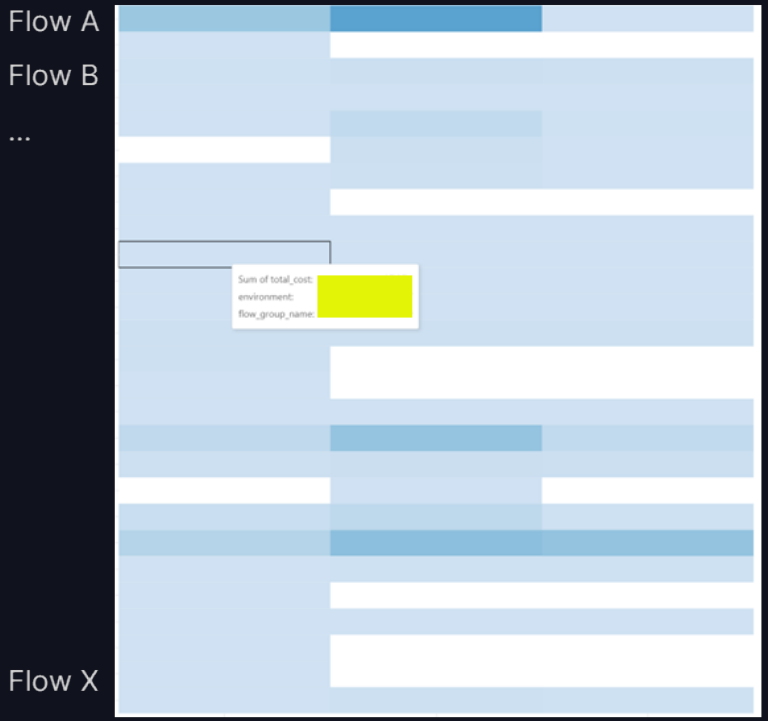
Mutual benefits driven by

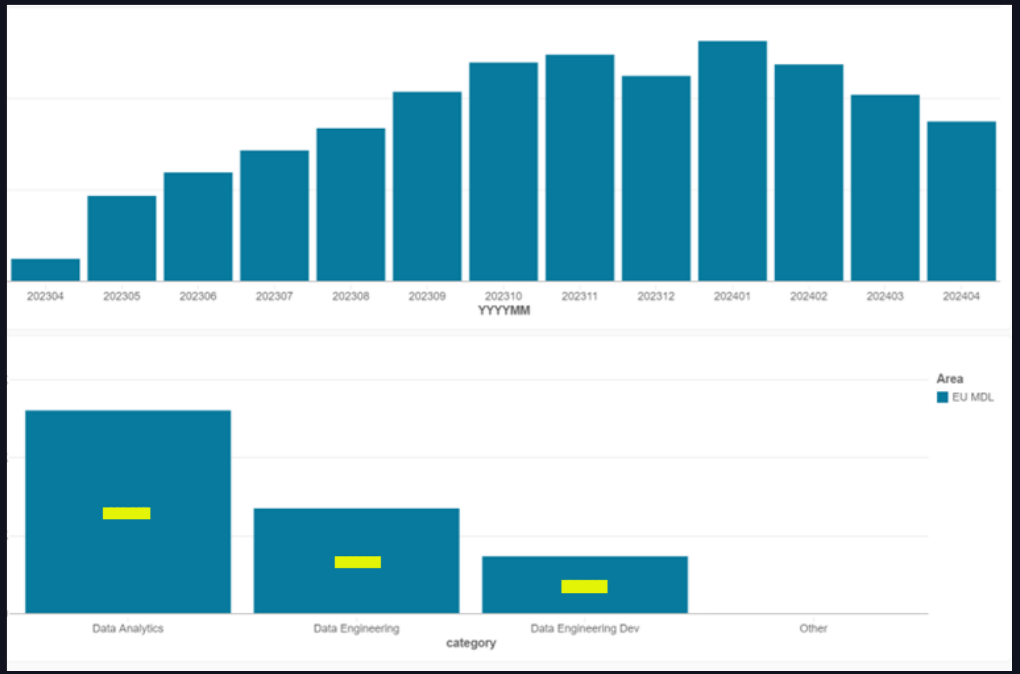
- Closer collaboration
- Agility
- Clearly audited access
- Instant integration of 74 tables



VISIBILITY INTO COSTS

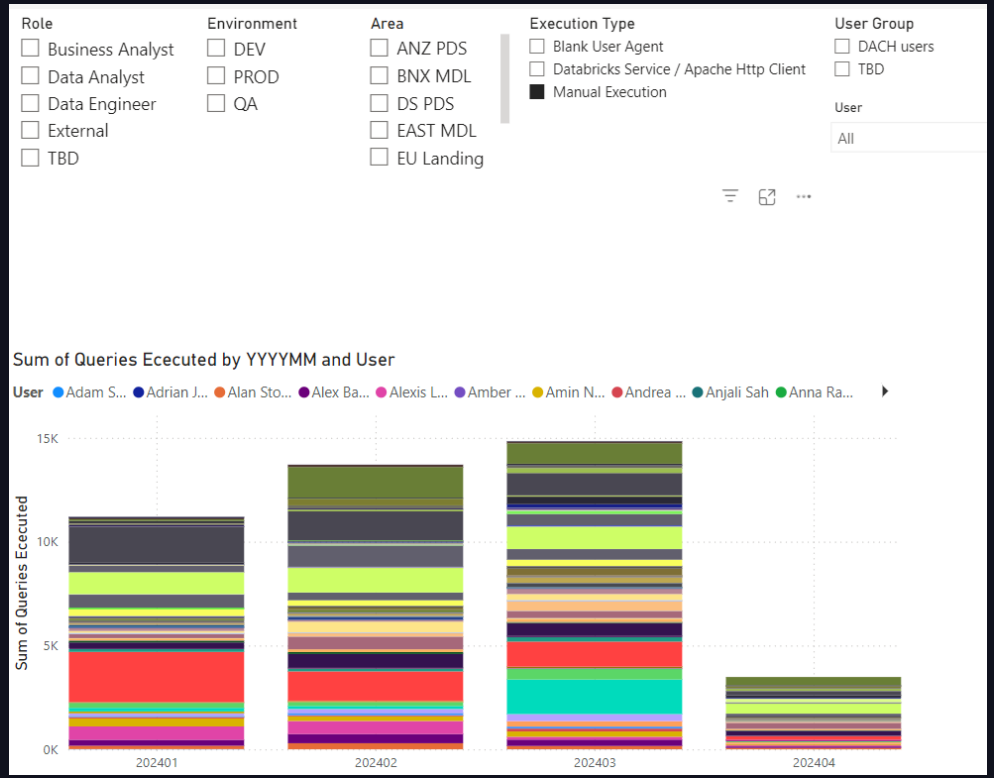
ACTIVITY BASED COSTING



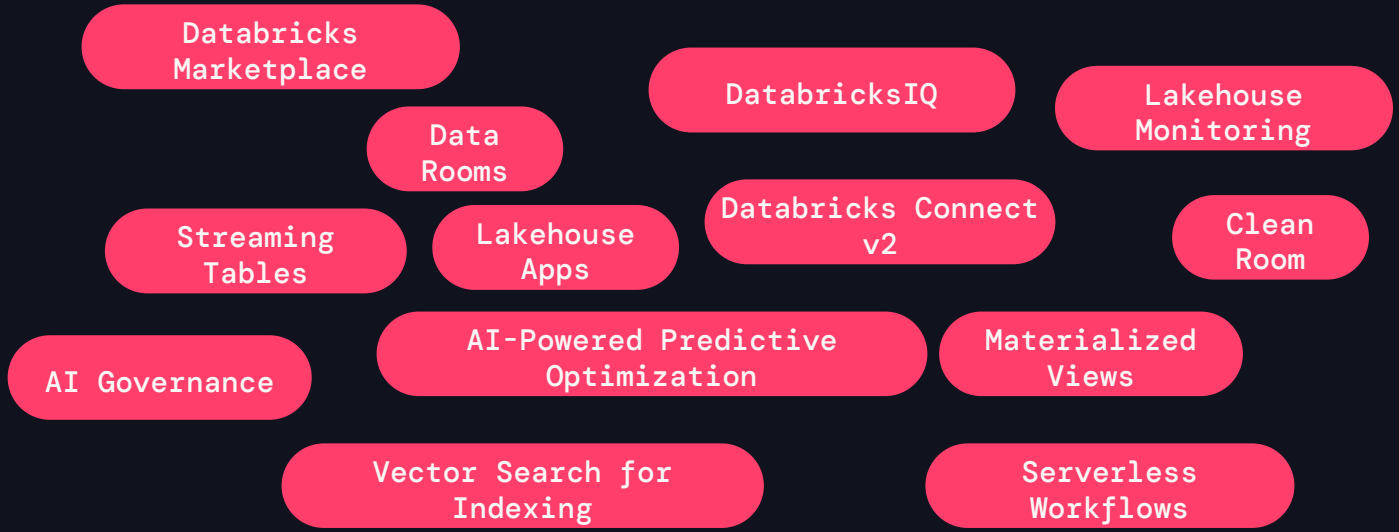


VISIBILITY INTO USAGE

- Track usage by persona & specific users
- Identify patterns of usage
- Identify areas for optimisation



FUTURE PROOFING



WE'VE LOVED COLLABORATING

OUR PARTNERSHIP

ALL ABOUT HABITS

1. Weekly question hours
2. Quarterly Community Sessions
3. Join on-site events (Data AI World Tour, Data Intelligence Days)
4. Solution architecture for our coolest projects

QUESTIONS?