DATA+AI SUMMIT BY databricks

Mastering Unity Catalog

Lessons from Zalando's Adoption Journey

Sebastian Herold
2024-06-11

WHO AM I?

Sebastian Herold

Sr. Principal Engineer @ Zalando

6y @ Zalando

ML Productivity

Streaming DWH

Data Platform

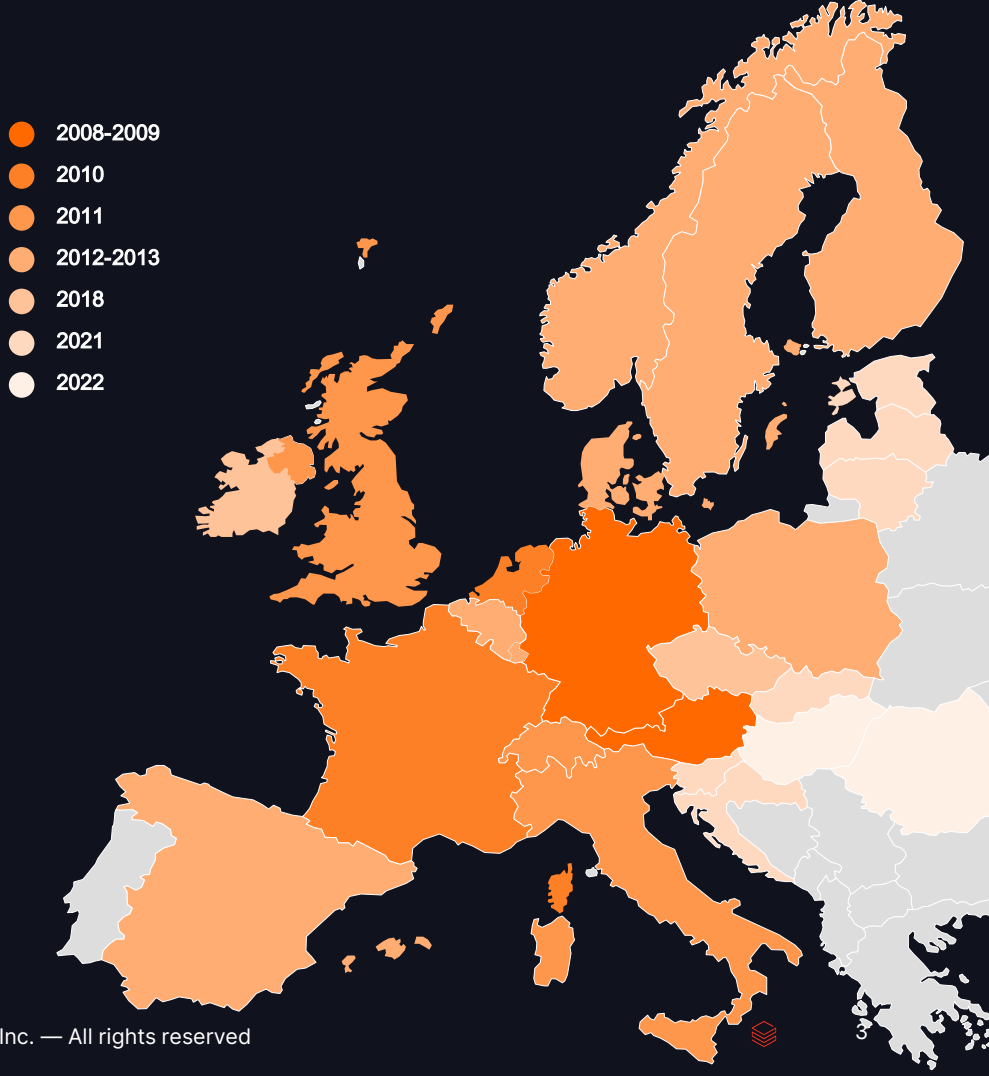
7y @ (Immo-)Scout24

DataDevOps Manifesto

Data Platform

Linkedin

We are the leading multi-brand fashion destination in Europe



>14bn

Euro GMV

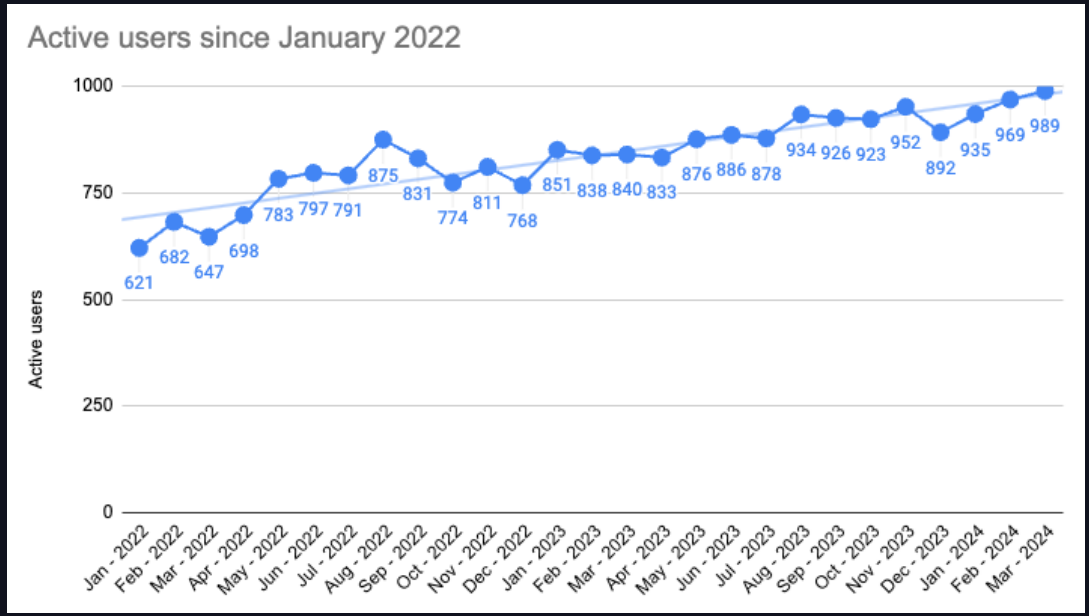
around 50m

active customers

DATABRICKS ADOPTION

@Zalando

- >950 Monthly Active Users
- >300 Teams
- ~15,000 Jobs per day



UNITY CATALOG - Our Motivation

Center of Innovation @ Databricks

- LakehouseIQ
- LLM Training
- Description generation
- Serverless
- Databricks SQL Warehouse
- System Tables

UNITY CATALOG - Our Motivation

One Central Technical Data Catalog

- Instead of >200 (see later)
- More transparency across teams
- Better data discovery
- Data sharing based on tables,

  not S3 paths

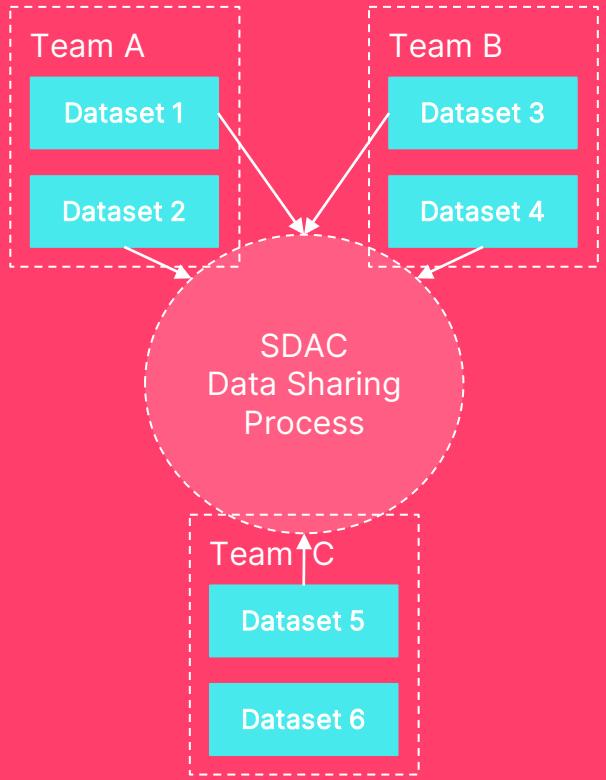
UNITY CATALOG - Our Motivation

Fine-grained Data Access Control

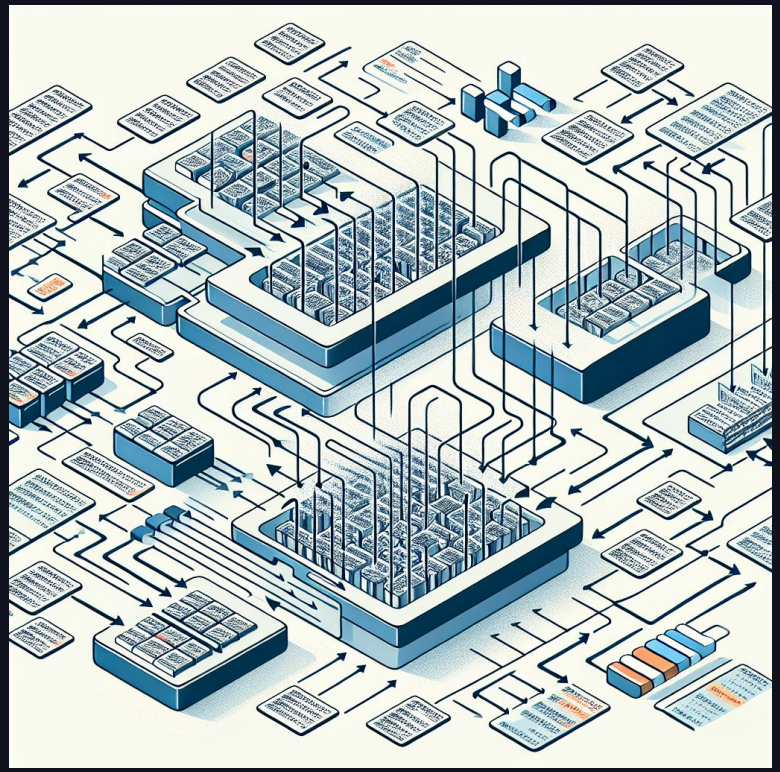
- Column-level Security
- Row-level Security
- Automatic Access Provisioning
- Integration with our Scalable Data Access Control program

EXCURSION - Scalable Data Access Control (SDAC)

- **The team is the security boundary!**
- Fine-grained access control across all major data systems
- Column-level security e.g. for PII columns
- Row-level security e.g. for antitrust or consent
- Automatic data classification
- Simplified process for non-critical use cases



UNITY CATALOG - Our Motivation



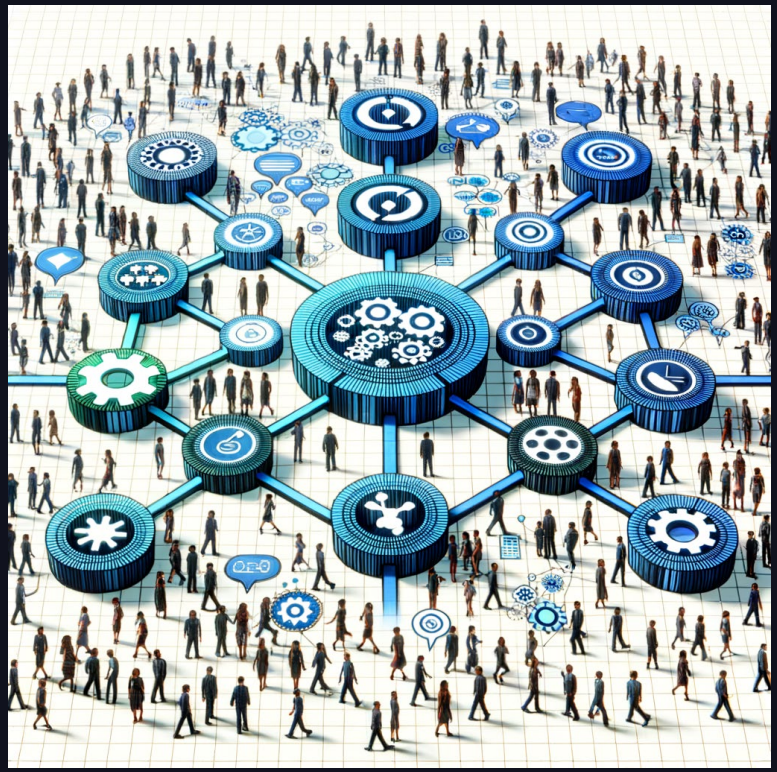
Data Lineage

- Automatic detection
- Relationships between tables
- Relationships between columns
- System table for export (in preview)

UNITY CATALOG - Our Motivation

Access Control at the Catalog

- Not at the cluster anymore
- People can easily switch clusters
- Centralising clusters for efficiency



BASIC DESIGN PRINCIPLES

For Unity Catalog Adoption

Full Self-Service

Common actions can be done in autonomy without interacting with e.g. central teams.

Pre-defined Templates

All major functions are available as pre-defined templates or components and can be deployed via CI/CD in your team environment.

Security by Design

The design respects teams as security boundary. Data sharing between teams is only possible via SDAC processes.

UNITY CATALOG

Guiding Questions

1. How do we structure Unity Catalog?

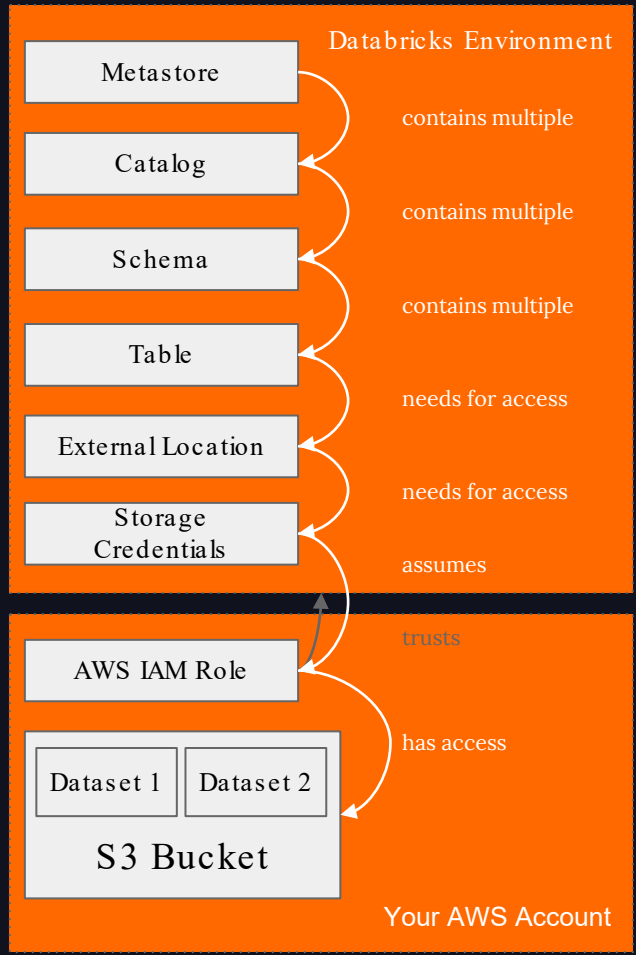
1. How do we map our security model (SDAC) to Unity Catalog?

1. How do we migrate our current model to the new model?

UNITY CATALOG

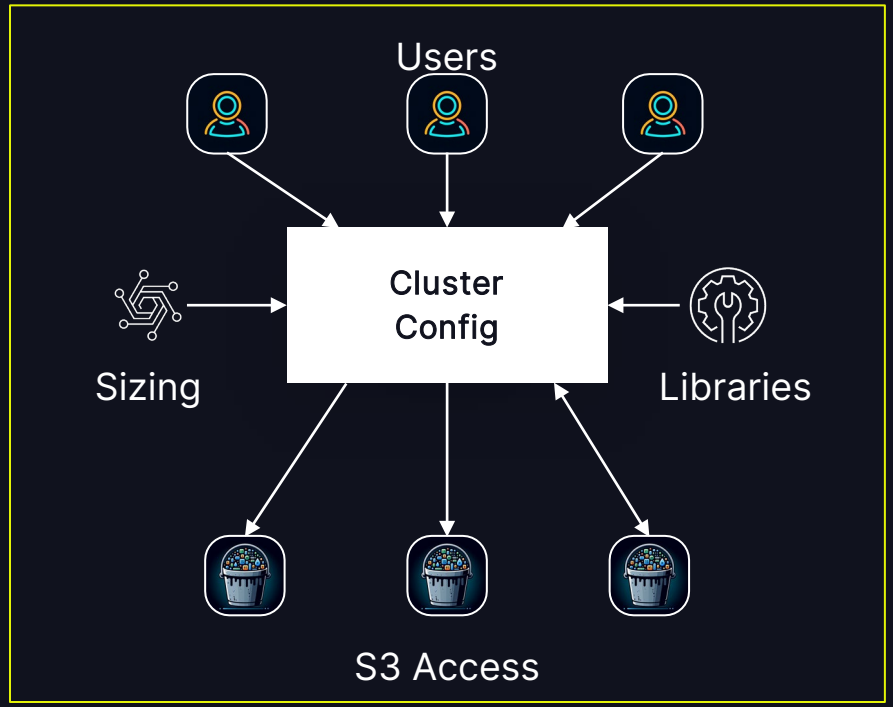
Recap: UC Structures

1. One Metastore per Region
2. Multiple Catalogs per Metastore
3. Schemas, Tables and Views inside Catalogs
4. Tables pointing to S3 paths via External Locations
5. External Locations managing S3 paths via Storage Credentials
6. Storage Credentials using AWS IAM roles to access S3 data
7. AWS IAM roles trust Databricks AWS account to access S3



CURRENT DATABRICKS SETUP

Cluster-centric Approach via Cluster Definitions



- Central Git Repo with JSON files
- Self-service through pull requests
- Cluster definition contains:
  - Owning team
  - Users having access to the cluster
  - Sizing (instance types, # nodes, ...)
  - Libraries
  - S3 Paths with read/write access
  - Access to AWS resources like SQS, IAM roles
  - Linked Data Processing Requests

CURRENT DATABRICKS SETUP

Cluster-central Approach via Cluster Definitions

What works well?

- Whole cluster config in one place
- Teams have independent clusters -> no noisy neighbors
- People trigger changes via PRs
- People can start clusters or jobs on their own
- Individual Hive Metastores

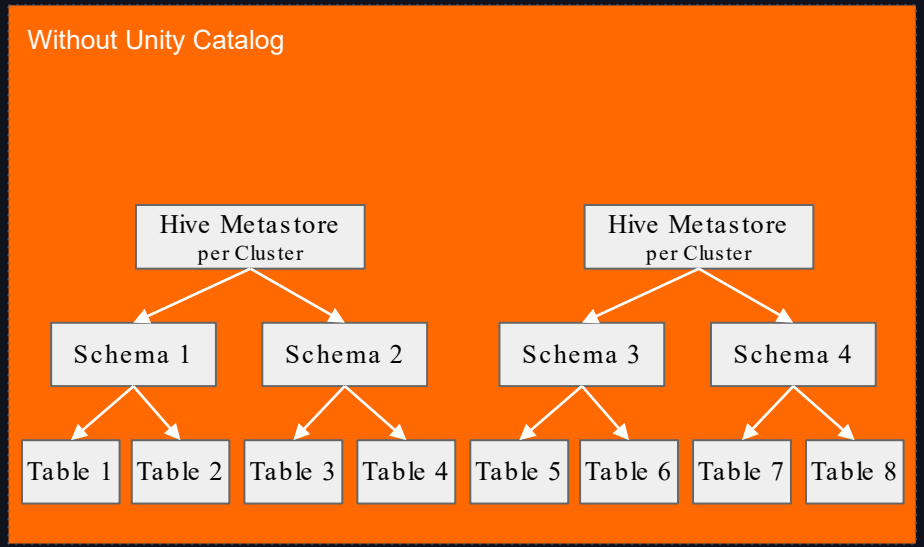
What do we lack?

- All changes need to be approved by central team -> wait time
- No CLS/RLS due to direct S3 access
- No central clusters
  - Less efficient
  - Long cluster startup times
- >200 Hive Metastores

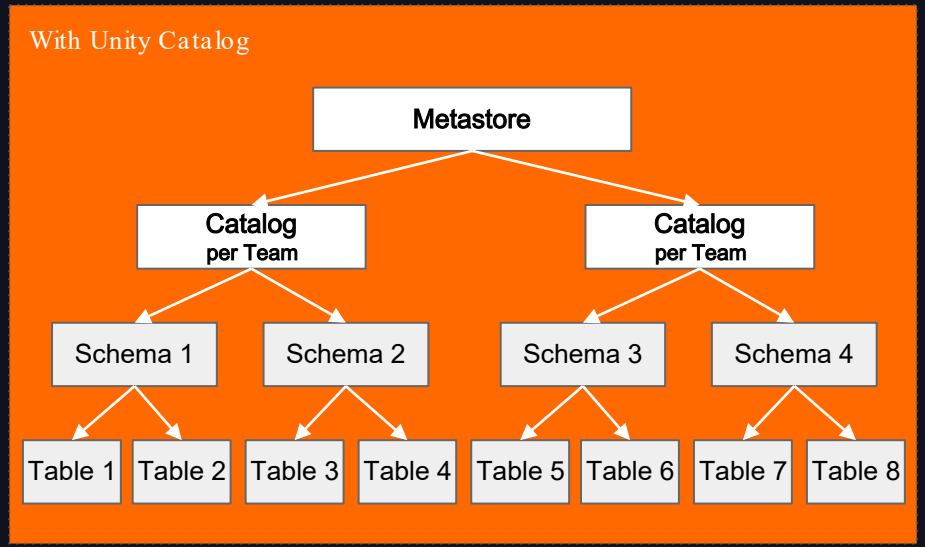
UNITY CATALOG

Catalog Model

Before



After



Why are we not done yet?

CHALLENGES FOR OUR ADOPTION

Table Owners Can GRANT Access to Others

```
# create table
CREATE TABLE my_table1
# owners can share table with other people
GRANT SELECT ON my_table1 TO my-best-friend-user
```

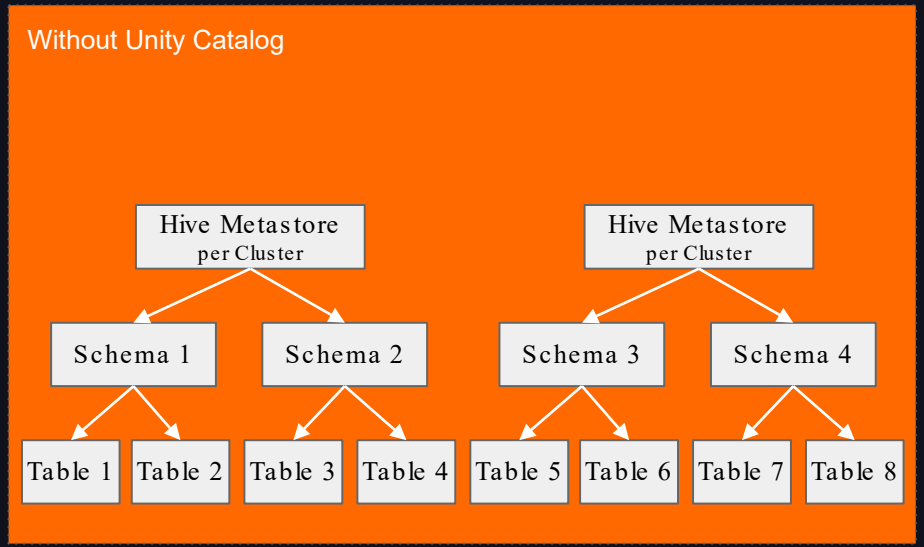
BUT: SDAC should be the only process to grant access!

Option 1 - Reactive:	Option 2 - By design:	Option 3 - The future:
Monitoring of audit log if unauthorized GRANTs have been executed	Introduce Dynamic Views in a shared catalog under control of central team	Wait for new policy framework (2025).

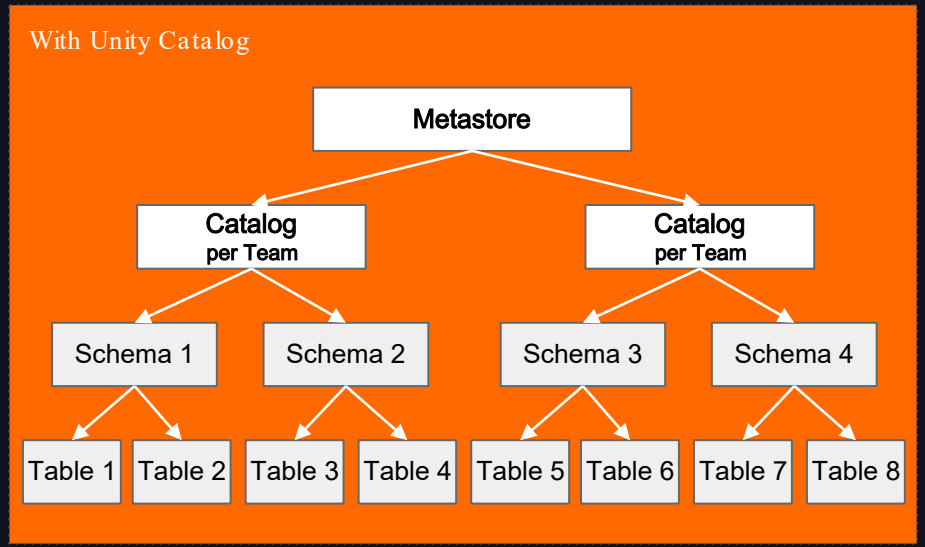
NEW CATALOG MODEL

Team Catalogs + Shared Catalog

Before



After

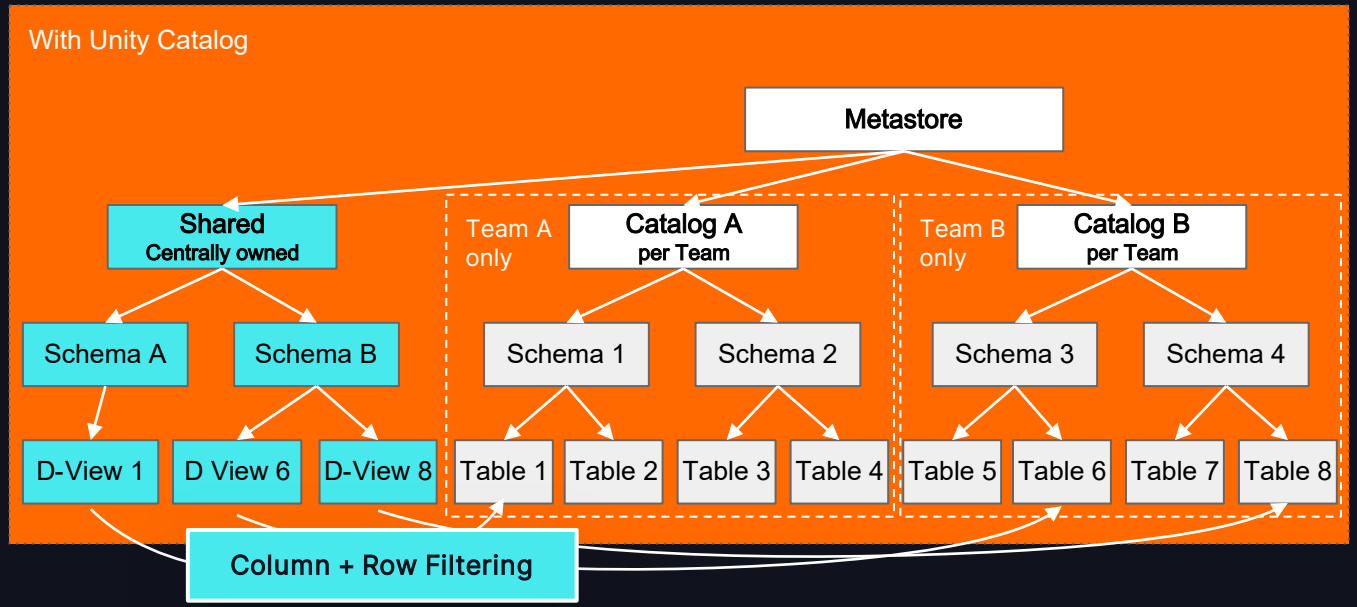


NEW CATALOG MODEL

Team Catalogs + Shared Catalog

- "Shared" Catalog
- Schema per Team
- Contains Dyn. Views
- RLS / CLS in Views
- SDAC Integration

After



NEW CATALOG MODEL

Centrally Managed Dynamic View

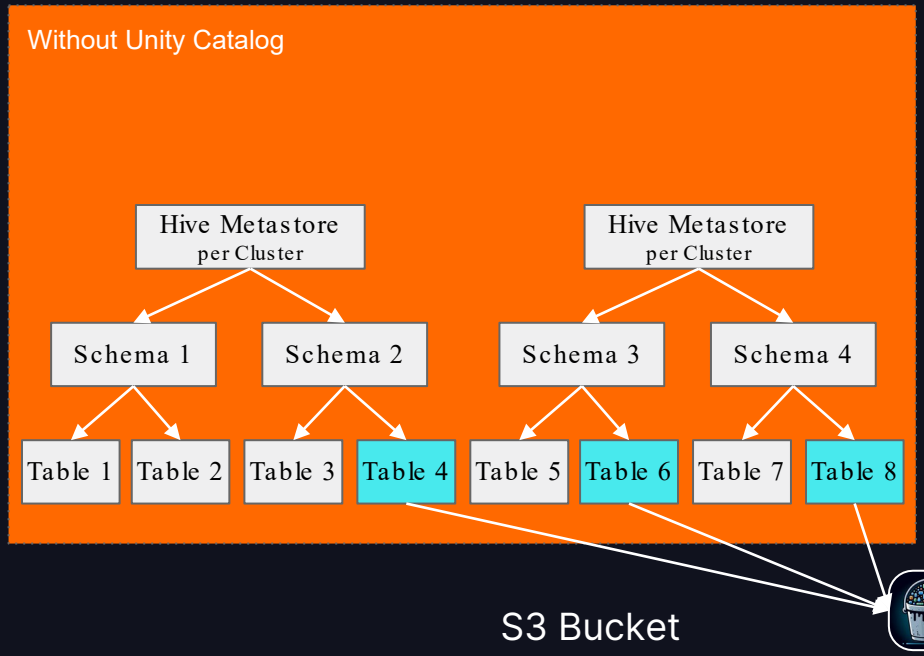
```
# created by publishing workflow
CREATE TABLE my_table1

CREATE OR REPLACE VIEW shared.sales.orders AS
SELECT
  non_pii_col1,
  non_pii_col2,
  CASE WHEN is_account_group_member('sales.orders_pii')
    THEN email ELSE NULL END AS email
FROM
  sales_team_cat.int_schema.sales
WHERE
  is_account_group_member('sales.orders_all')
  OR merchant_id = 'ZALANDO'
```

- Owner by central team
- Based on data classification
- Created after publishing
- Access managed by SDAC

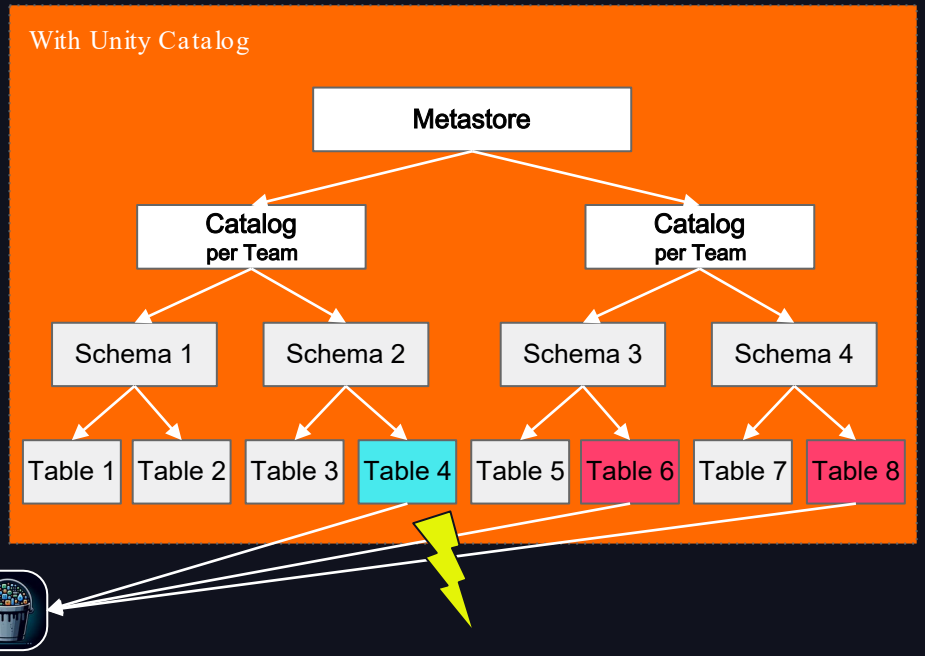
CHALLENGES FOR OUR ADOPTION

Only One External Table Can Point To a Specific S3 Path

Before



After



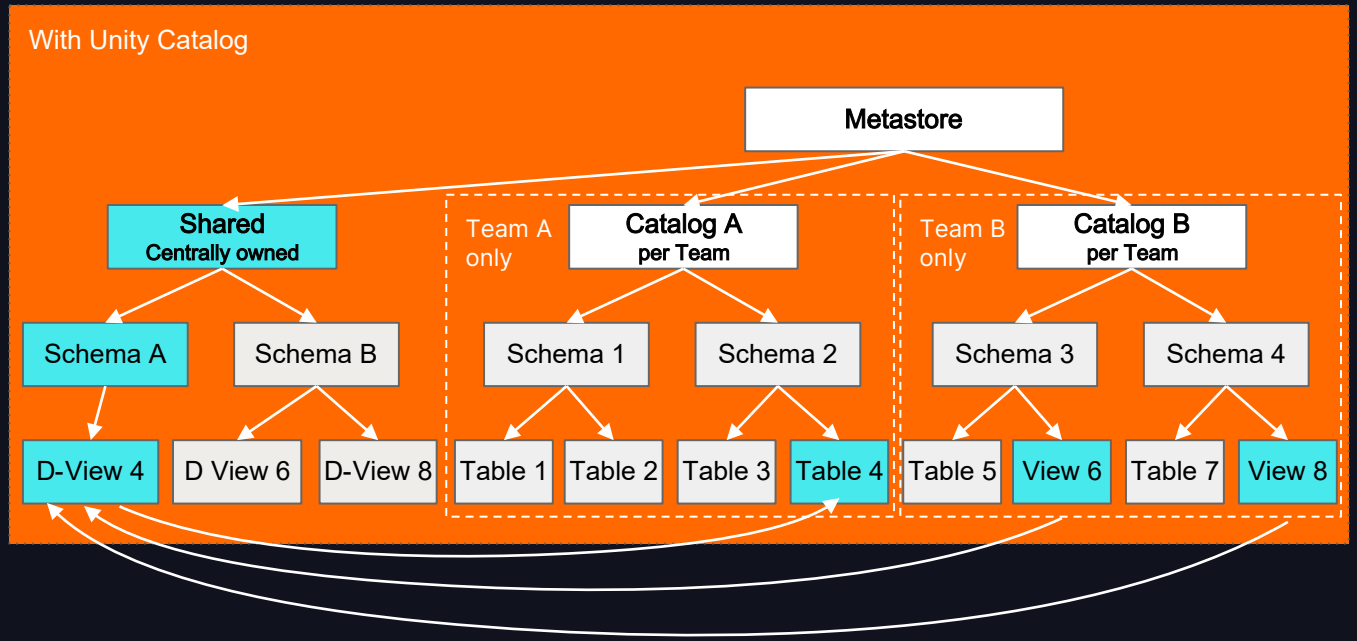
S3 Bucket

NEW CATALOG MODEL

Unique Table per S3 Path

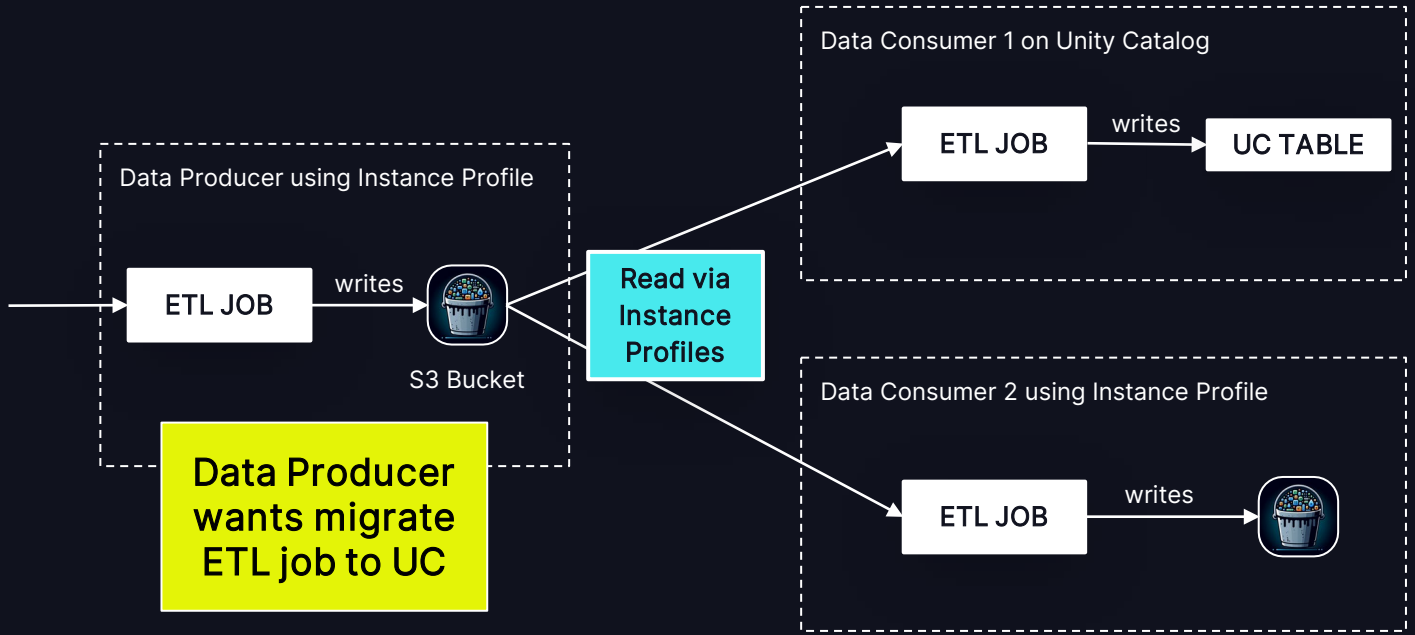
- All data shared via Shared Catalog, not via S3 path
- Inform teams during migration
- Dependency to owner to create table first
- Push owners of major datasets

After



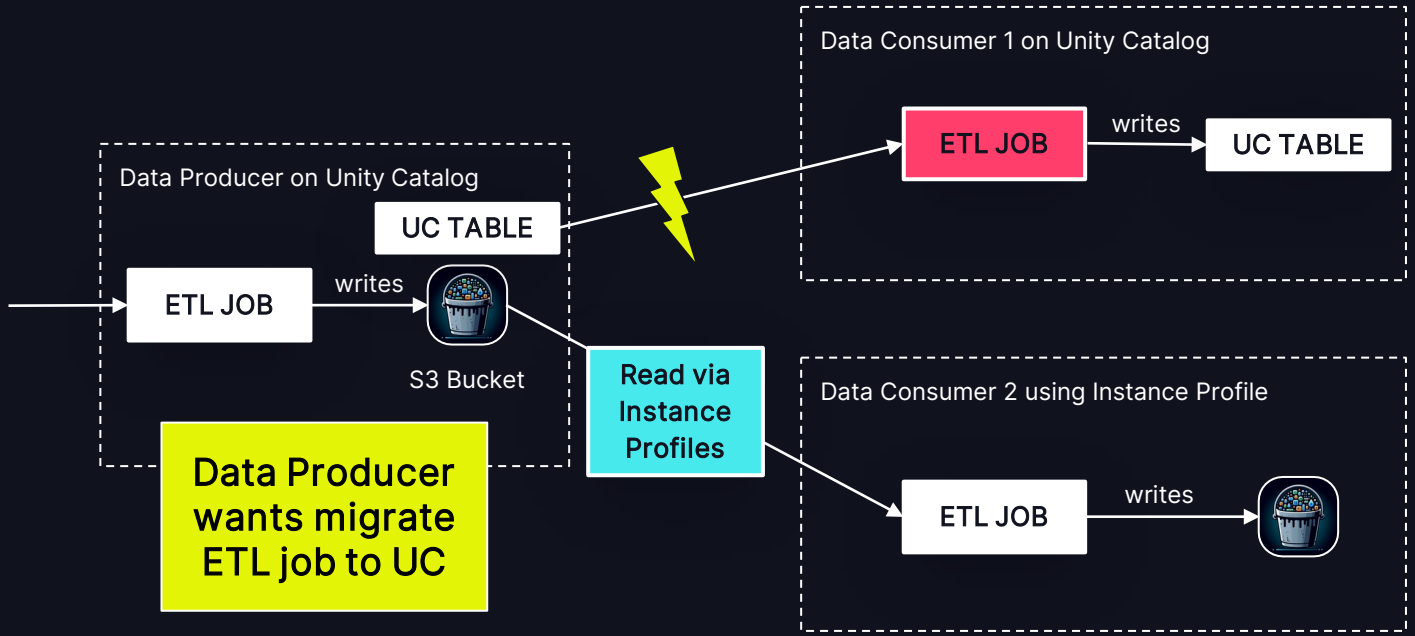
CHALLENGES FOR OUR ADOPTION

Unity Catalog Permissions Overwrite Instance Profile Permissions



CHALLENGES FOR OUR ADOPTION

Unity Catalog Permissions Overwrite Instance Profile Permissions



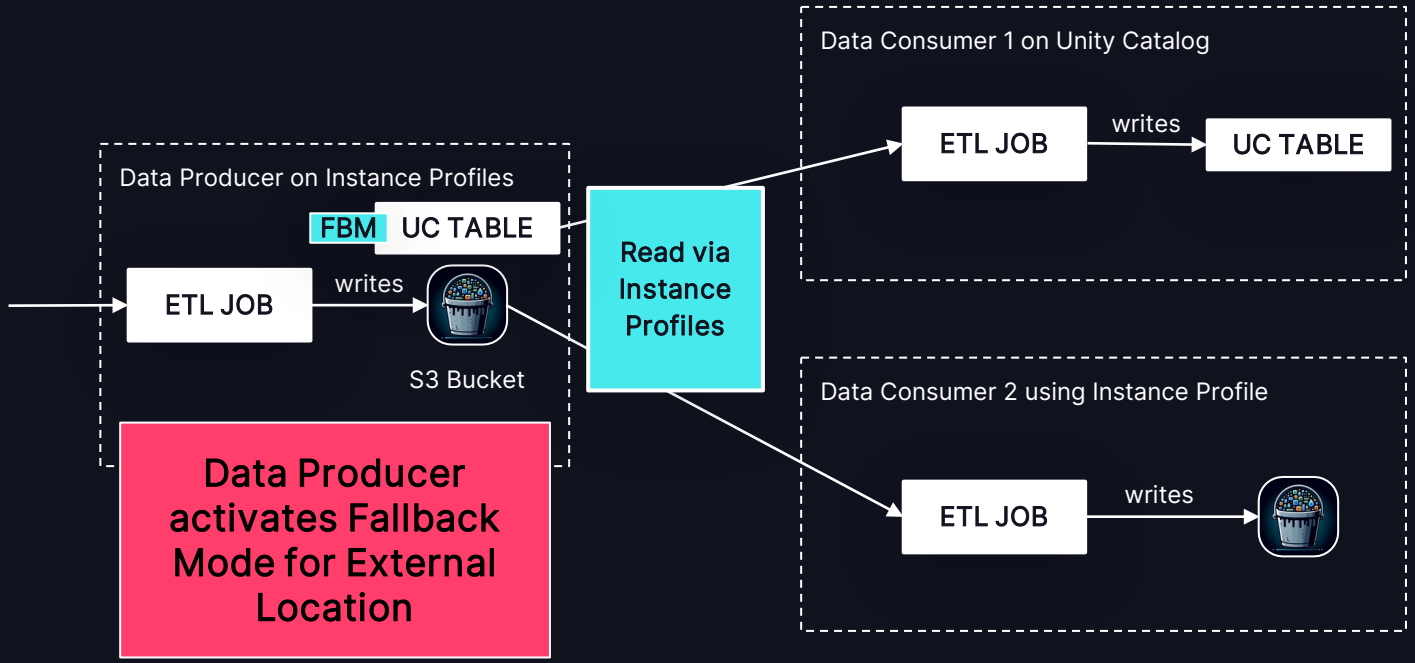
Existing instance profile permissions are ignored for UC consumer 1.

Unity Catalog permissions immediately apply.

Creates inter-team dependencies for migration.

NEW CATALOG MODEL

External Locations with Fallback Mode



New Fallback Mode to the rescue

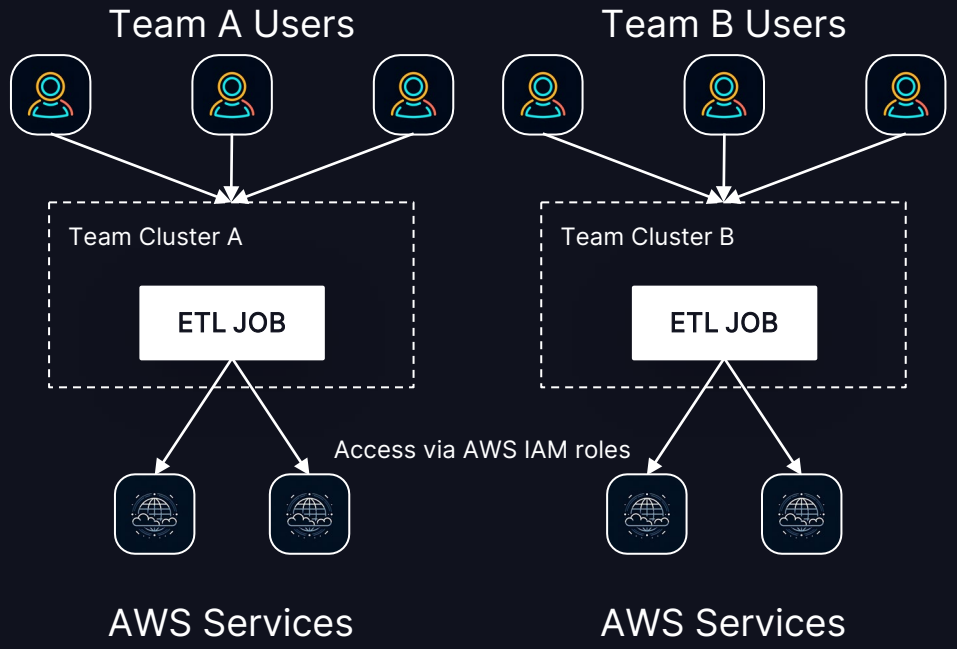
Falls back to instance profile permissions if UC permissions fail

Removes dependencies during migration

Expected delivery in Q2 2024

CHALLENGES FOR OUR ADOPTION

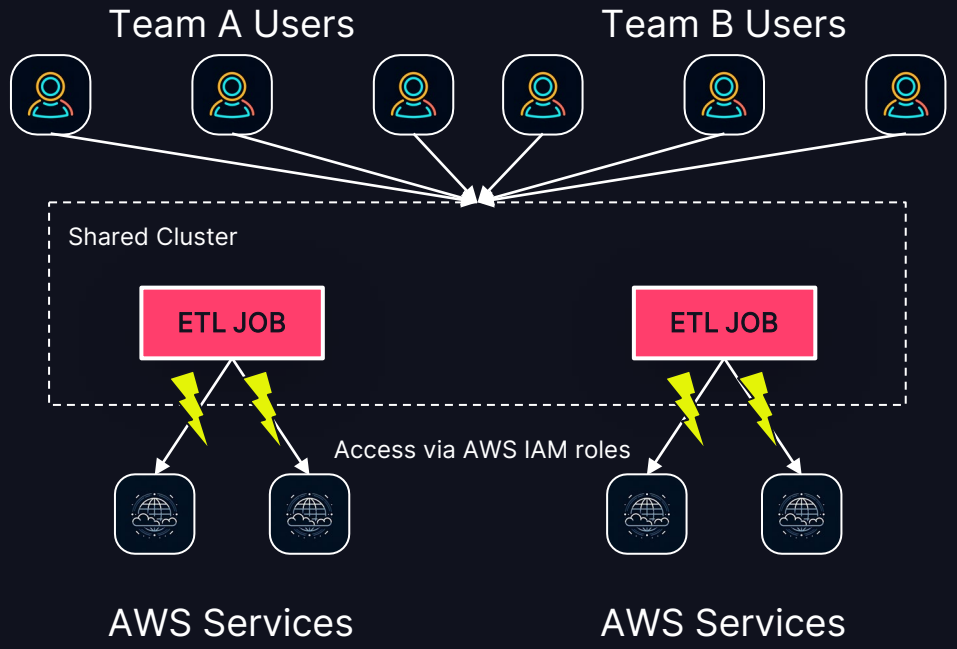
No Individual AWS Credentials on Shared Clusters



Team clusters can access individual AWS services by assuming AWS IAM roles

CHALLENGES FOR OUR ADOPTION

No Individual AWS Credentials on Shared Clusters

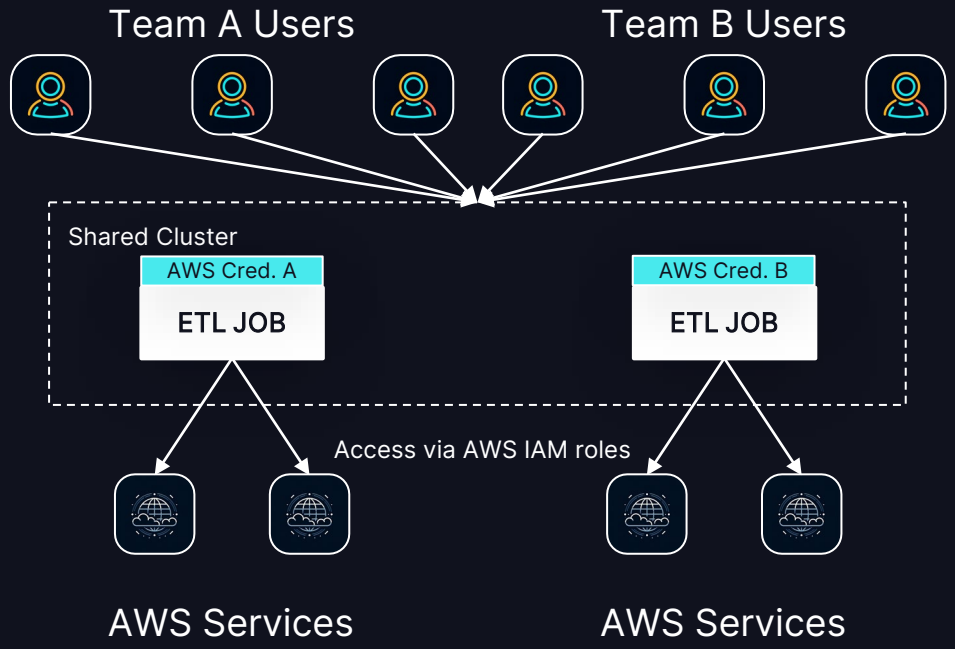


No individual AWS credentials on shared clusters

Access to AWS services either shared or denied

NEW CATALOG MODEL

New Cloud Service Credentials



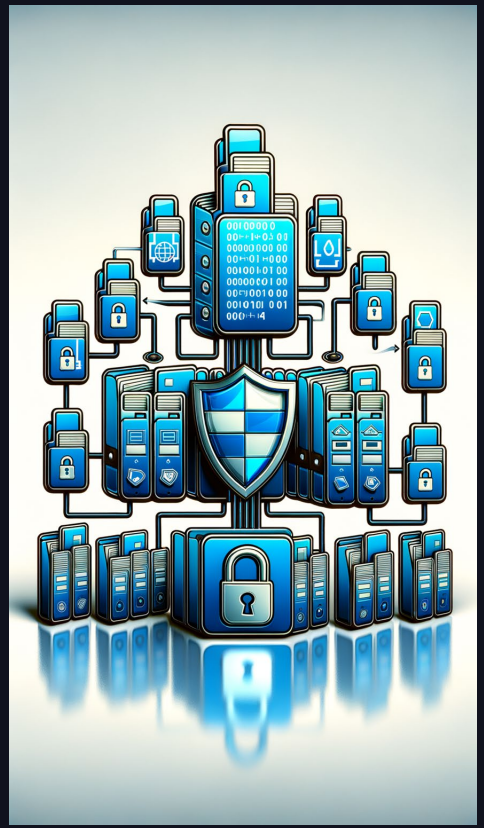
Individual AWS credentials on shared clusters

Expected in private preview soon

NEW CATALOG MODEL

Summary

- Users can create resources (catalogs, external locations, service princials) via AWS Service Catalog
- A catalog is always owned by one team
- Special “shared” catalog for shared data
- Inside shared catalog, Dynamic Views pointing to source tables in team catalogs
- Dynamic Views do CLS and RLS
- Cloud Service Credentials for AWS access on shared clusters



WHAT IS THE MIGRATION PATH ?

MIGRATION PATH

Basic Principles

Leverage Unity Catalog

Try to benefit from Unity Catalog features as early as possible to boost productivity and security.

Migration in Isolation

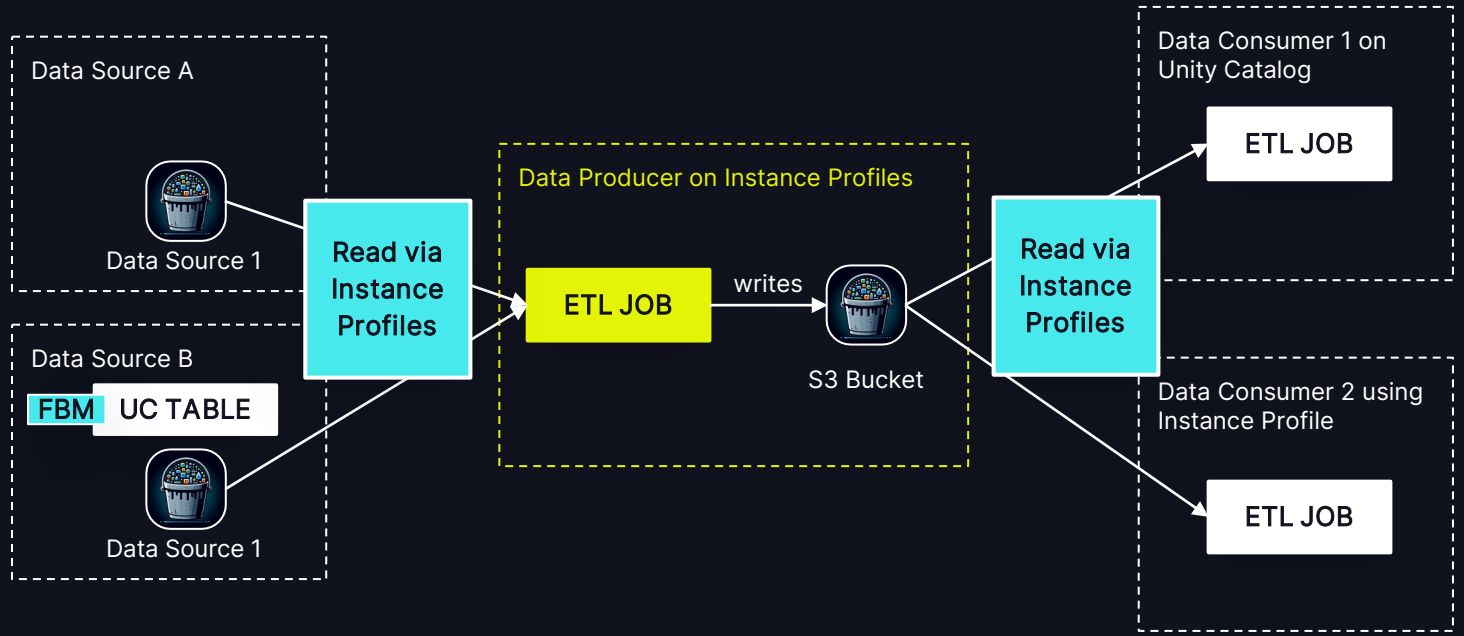
Teams can migrate to Unity Catalog in isolation with little to no dependencies to other teams and their migration speed.

Automation

We prefer central automation over decentral manual processes to keep migration effort for teams low.

MIGRATION PATH

For Data Producers- Typical Case

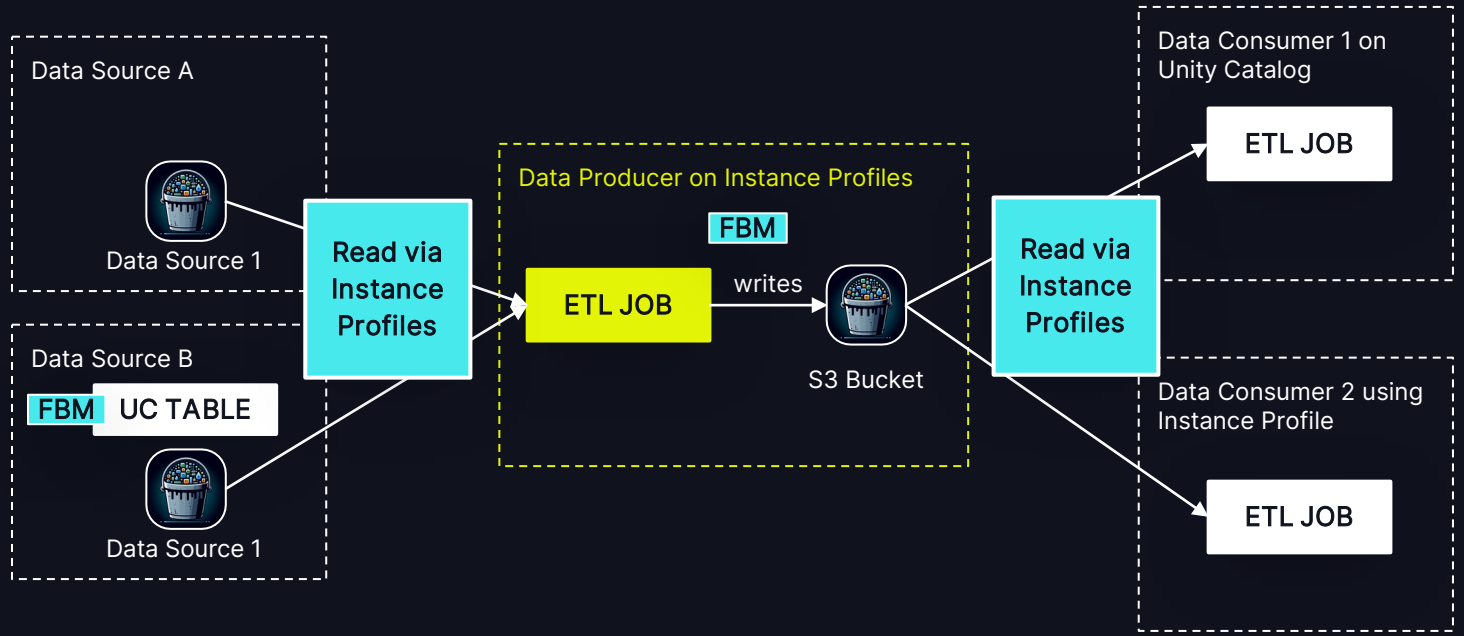


All combinations possible

Data sources and data consumers can be using Instance Profiles or Unity Catalog

MIGRATION PATH

Step 1 - Activate UC and Fallback Mode

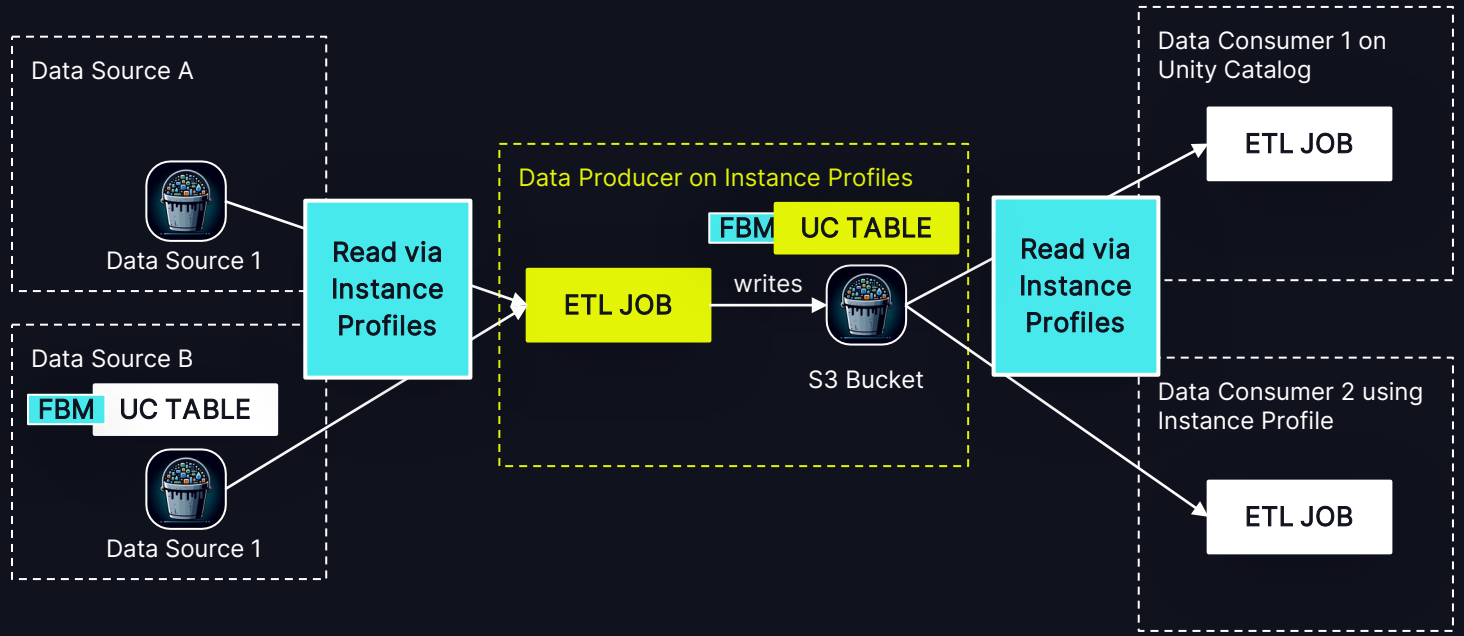


Data Producer activates Unity Catalog for the job.

Create External Location with Fallback Mode activated.

MIGRATION PATH

Step 2 - Create Table

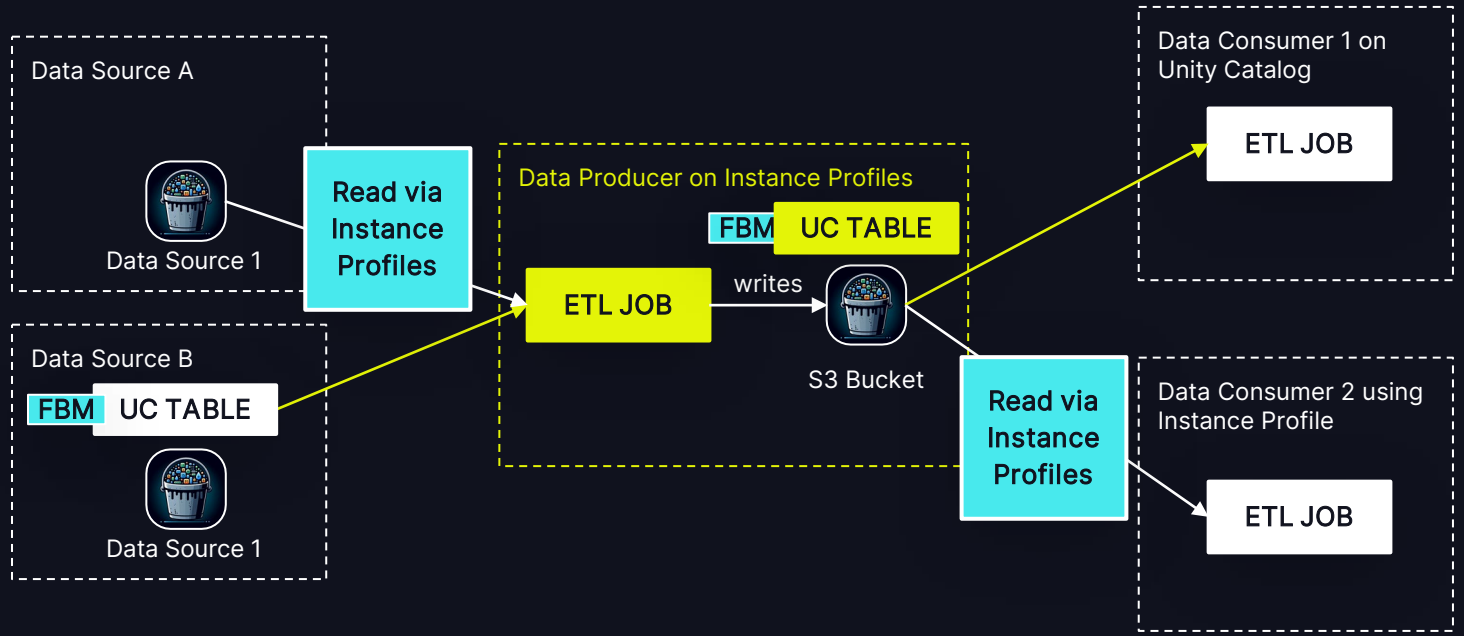


Data Producer creates a table in Unity Catalog

Or use SYNC command to sync from existing HiveMetastore

MIGRATION PATH

Step 3 - Get Unity Catalog Permissions

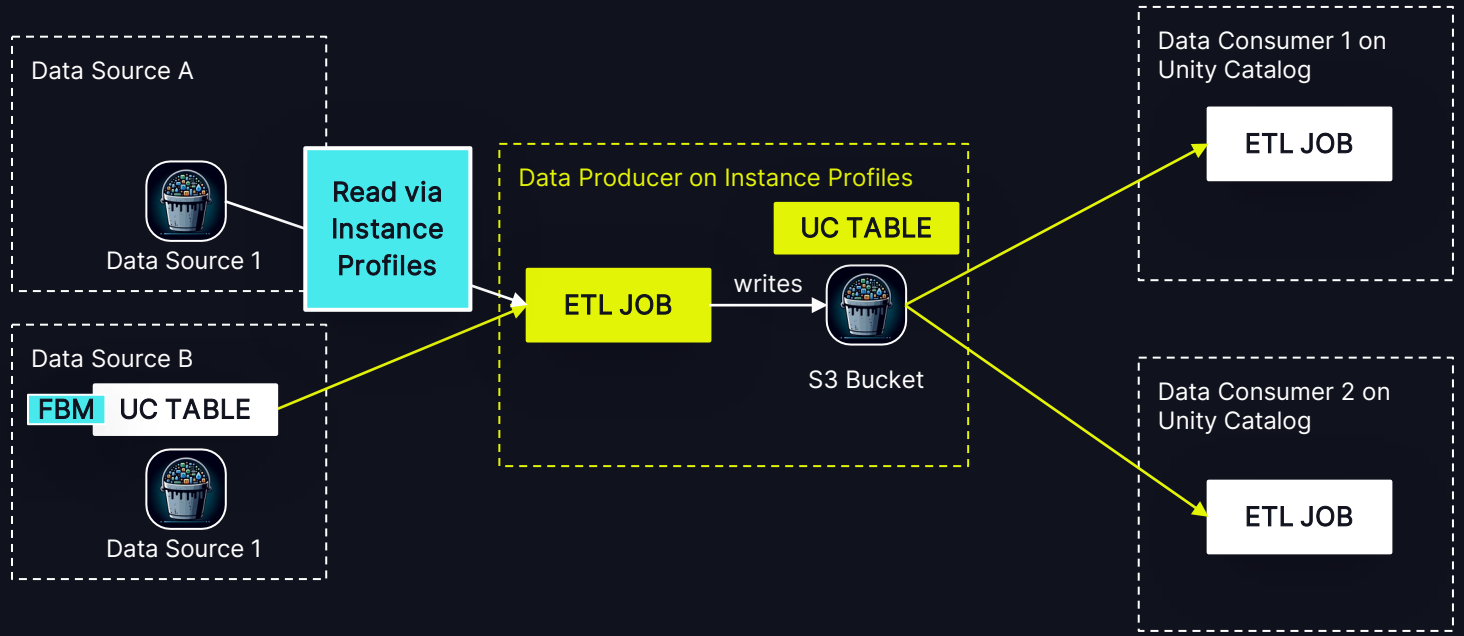


Use SDAC process to get access to data sources

Change jobs to directly consume from Unity Catalog tables

MIGRATION PATH

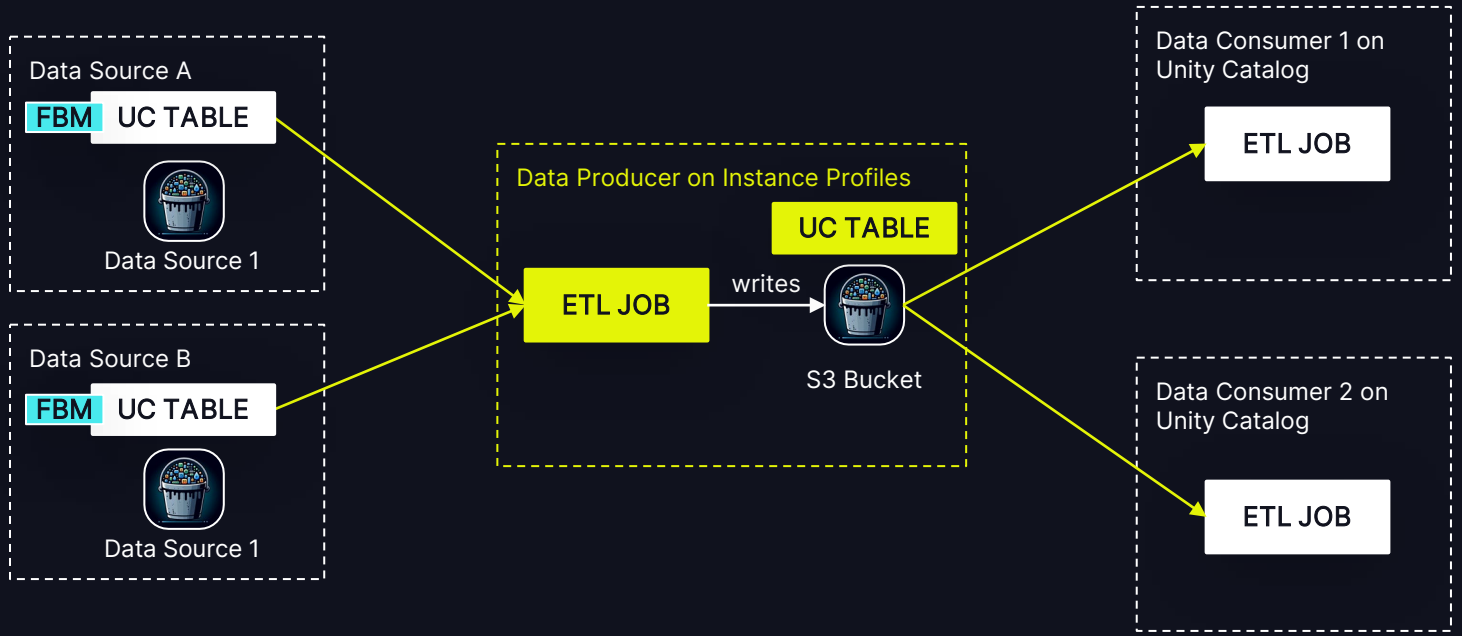
Step 4 - Deactivate Fallback Mode



When all consumers migrated to Unity Catalog, fallback mode can be deactivated

MIGRATION PATH

Step 5 - Finish Migration



When all data sources migrated to Unity Catalog, Instance Profiles are not needed anymore

MIGRATION STATUS

Will Be a Long Way

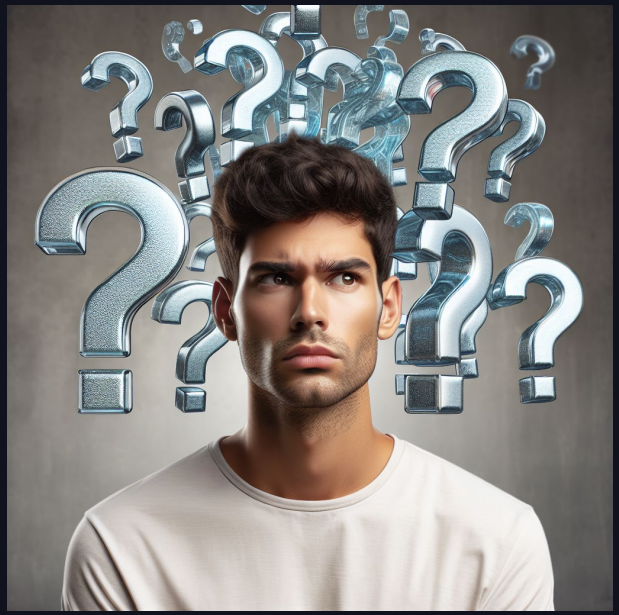
- Started with pilot team
- Self-service deployment is ready
- SDAC integration to be finished
- Pushing major data producers
- Whole migration will take multiple quarters

Big Thanks to

Zalando Data Transformation Team:
Hiroki Yonetani, Dejan Simic, Pavel Knapek (EPAM), Aruba Khan, Stefan Bojarowski, Aykut Akin

Databricks (for endless support sessions):
Marcin Wojtyczka, Ivan Trusov, Lars George, Todd Greenstein, Paul Roome, Mareike Kretschmer, Luba Bohad

Thank you



Questions?