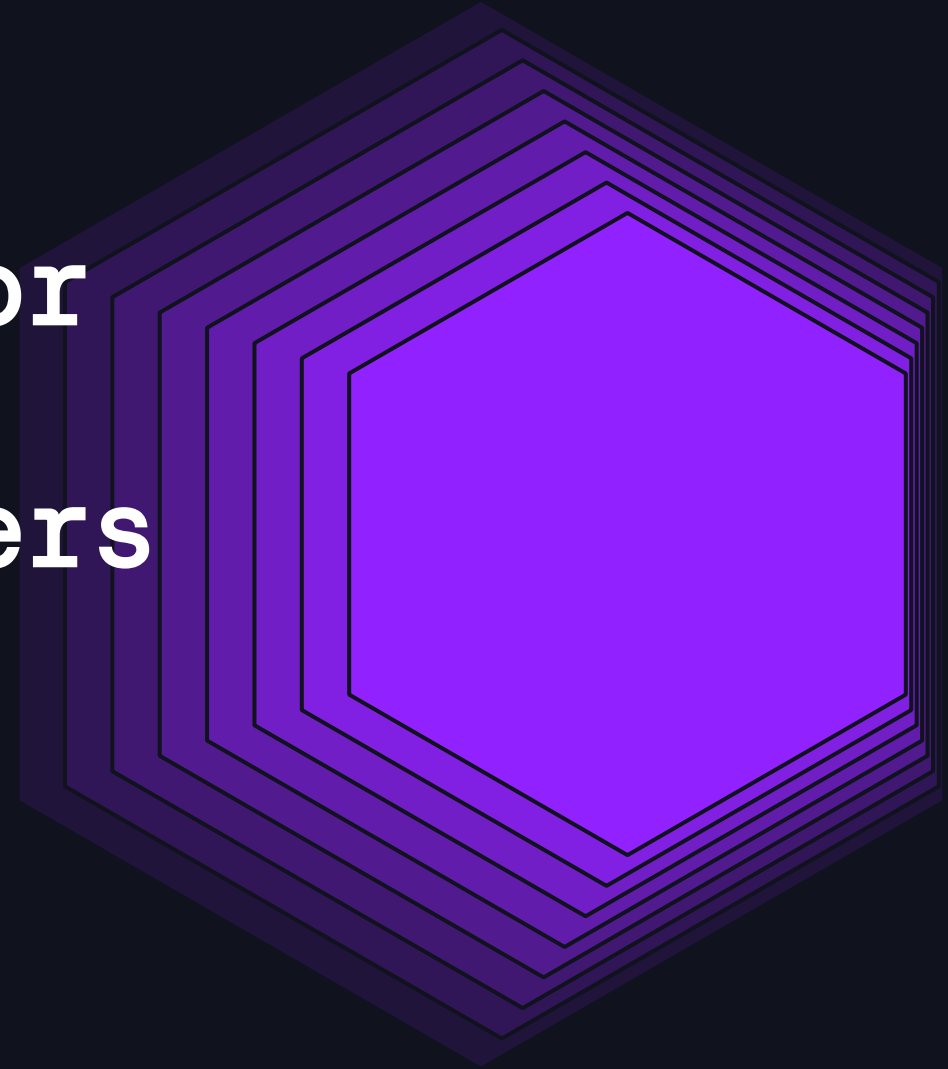


Limitless Scaling for the Enterprise: Onboarding 500k+ Users to Databricks

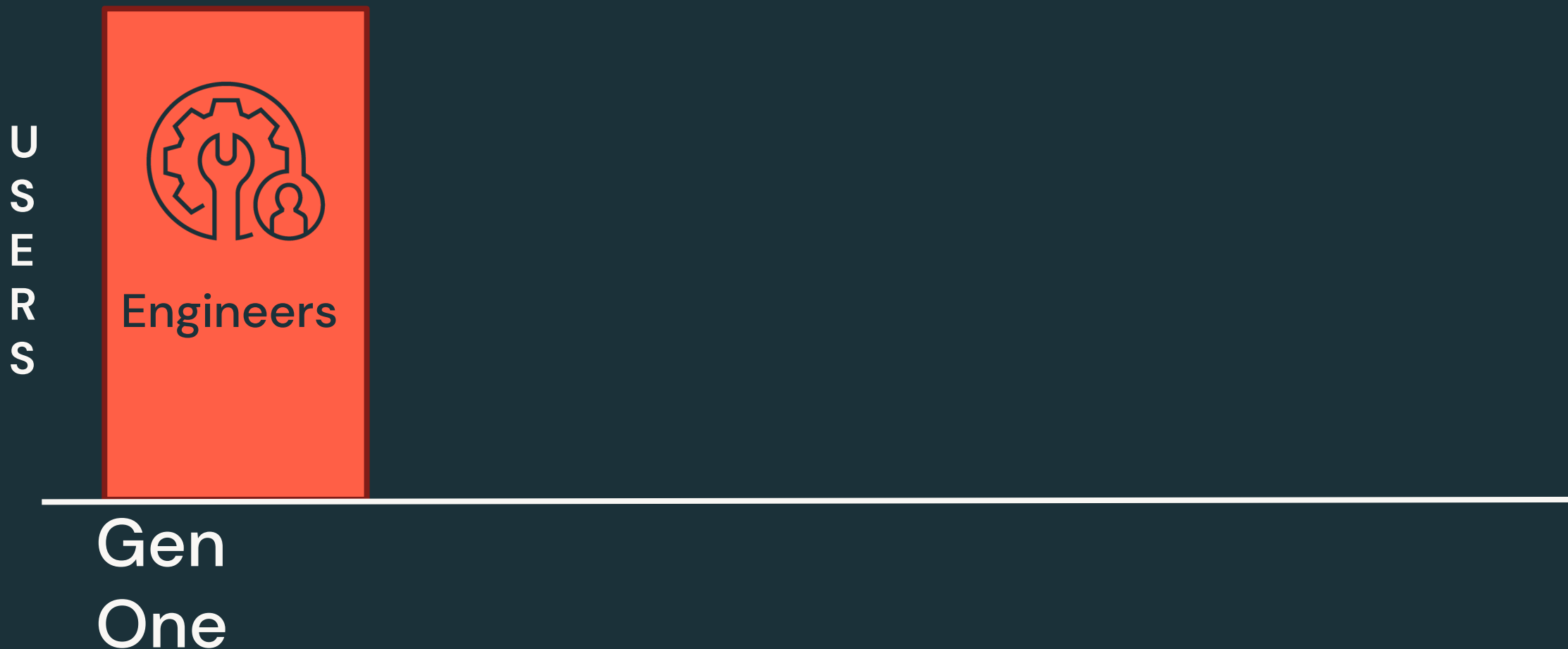


Siddharth Bhai, Silviu Tofan
13 June 2024

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

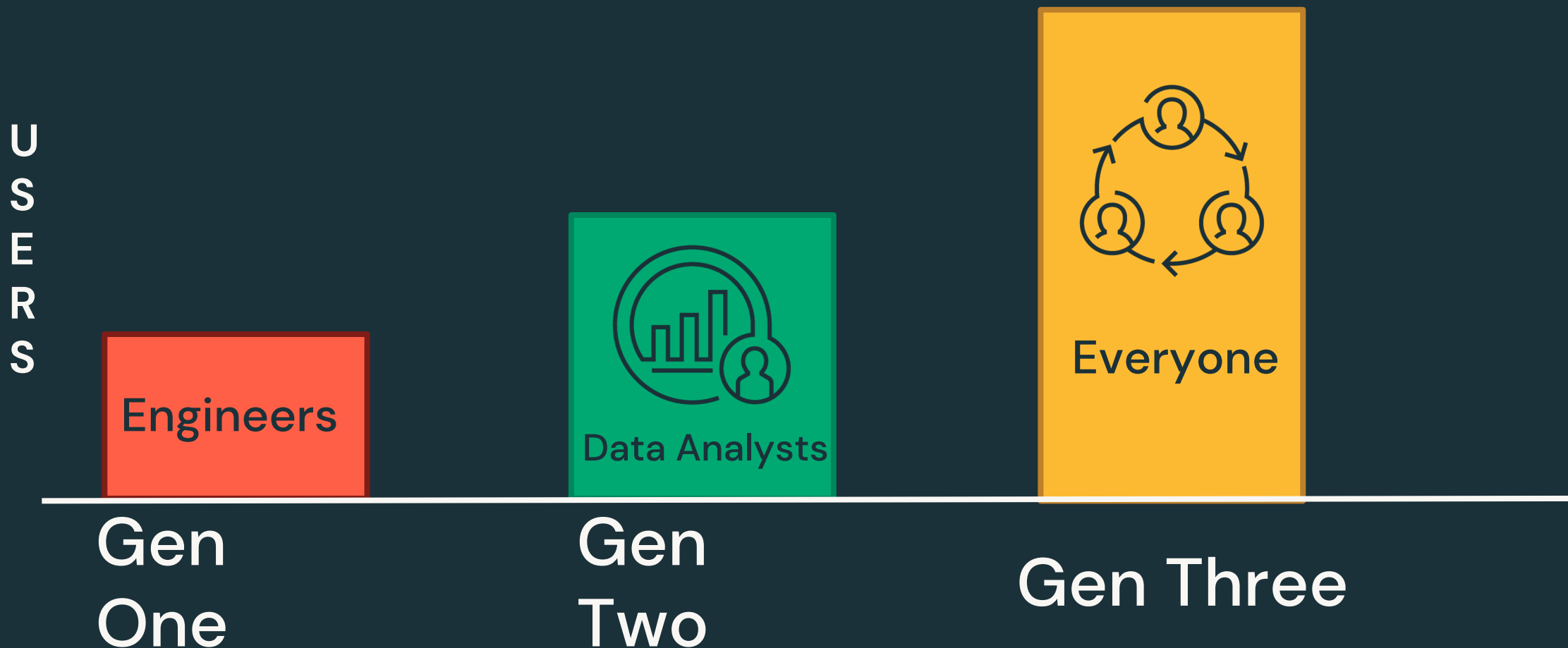
Democratizing access to Data+AI



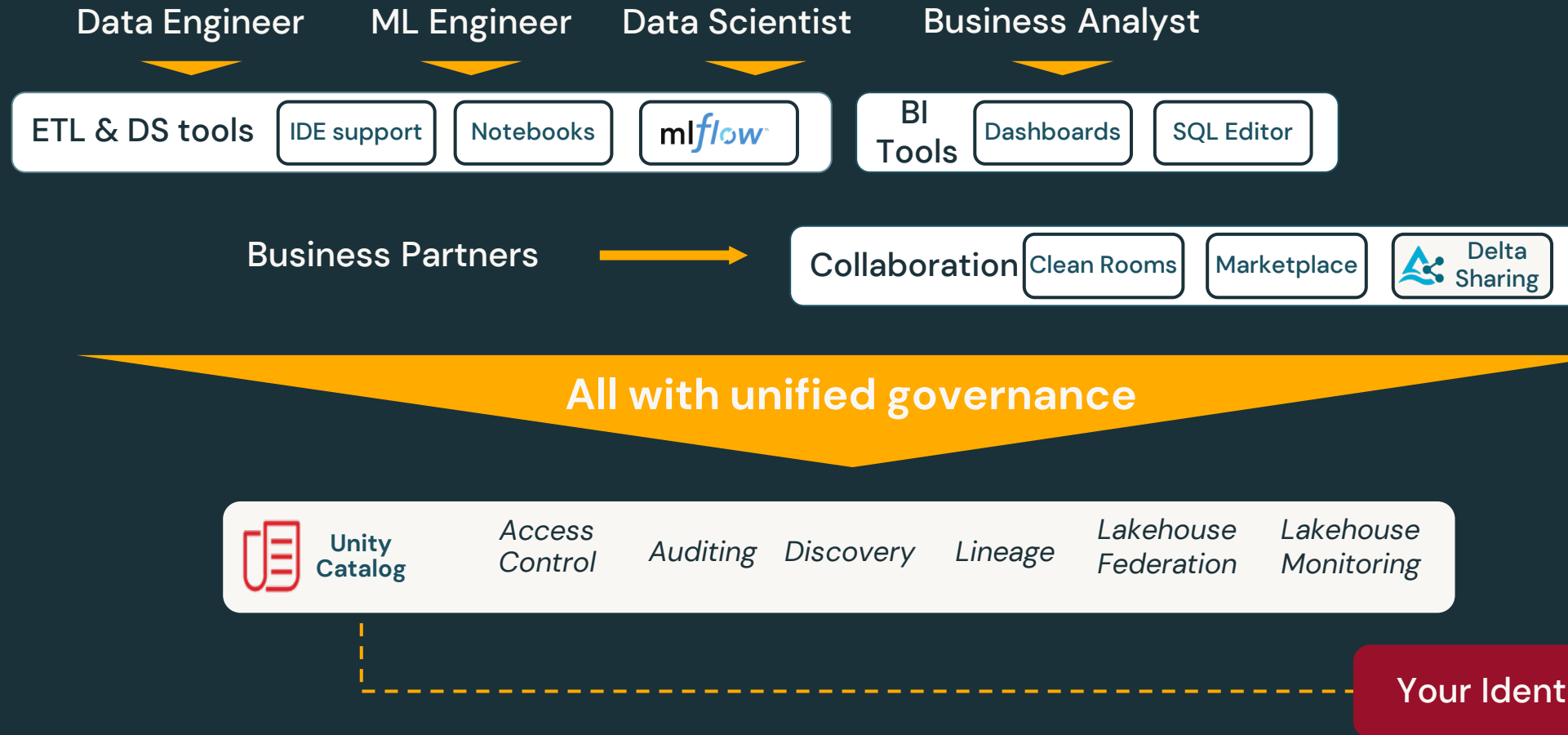
Democratizing access to Data+AI



Democratizing access to Data+AI



Unified governance should work for all users



“Databricks has taken a giant leap in simplifying account-level identity management, making it more intuitive and efficient. The recent additions to configure sync only once, **simplified group-based administration, and massively increased scale for users to the account console elevate our administrative capabilities, providing **a seamless experience** in orchestrating the intricacies of our data platform needs.”**

– Alexander Summa, Lead Solution Architect, Mercedes-Benz



Session outcomes

Limitless scaling for the enterprise – to 500K+ users and beyond



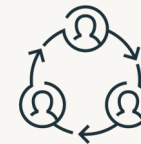
Provisioning & user logins

Best practices to provision users and have them login securely



Data isolation & external tools

Leverage proven strategies to configure data access correctly and from 3P tools



Automation & Delegation

Learn how to automate and delegate to scale Databricks usage for the long-term securely



Provisioning & user logins

Best practices to provision users and have them login securely



Data isolation & external tools

Leverage proven strategies to configure data access correctly and from 3P tools

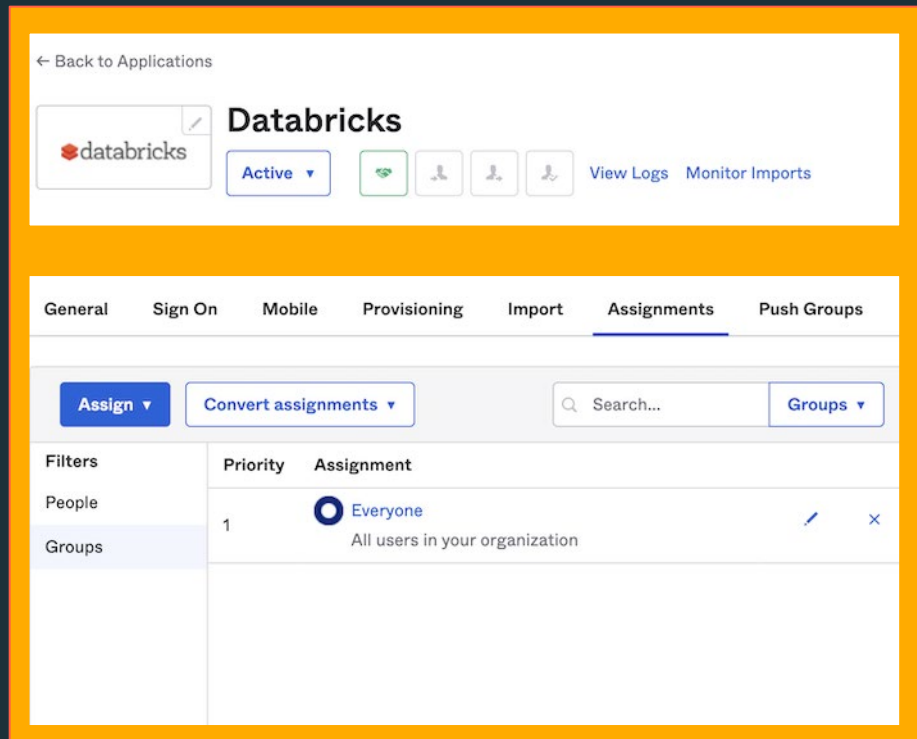


Automation & Delegation

Learn how to automate and delegate to scale Databricks usage for the long-term securely

Onboard first team

Enable proof of concept



Set up **SSO**

Security-compliant logins

Onboard first **users**

The proof of concept team

Use one **workspace**

Keep things simple!

Onboard the division

Deploy the first use case to production

Create workspaces

DEV – from before

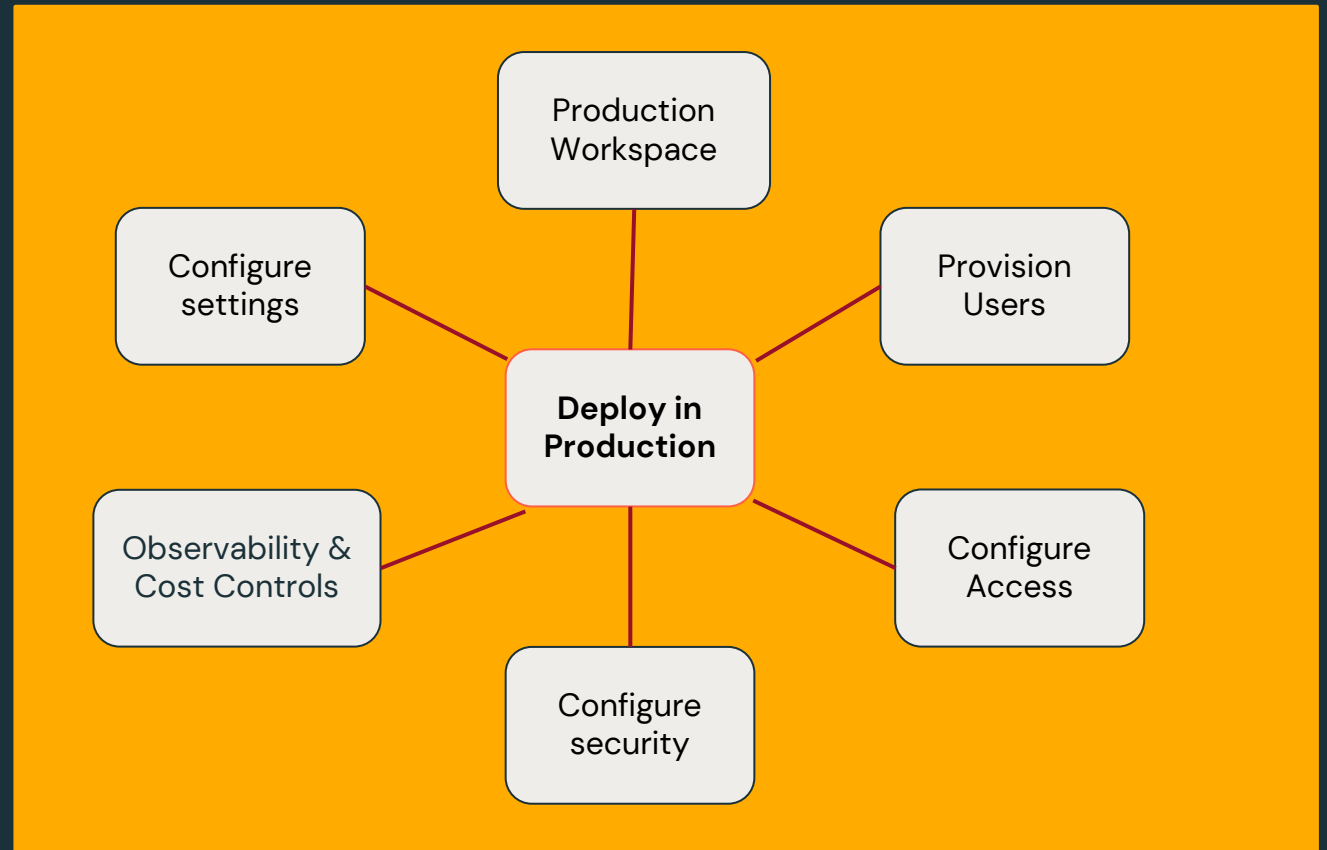
STAGING – **new**

PRODUCTION – **new**

Configure access control

Secure workspaces

Select right **admins**



Onboard next divisions

Standardize process to scale faster



Secure



Audit



Processes



Standardize

User groups
To configure access

Integrate with processes
For new business units

Standardize approvals
From
stakeholders

Data + AI and your organization

Core personas



Databricks admins



Data stewards



Data scientists



Data engineers



Data analysts



Business users

Stakeholder personas



Cloud Ops
Admin



Identity
Admins



Billing &
procurement



Security &
compliance

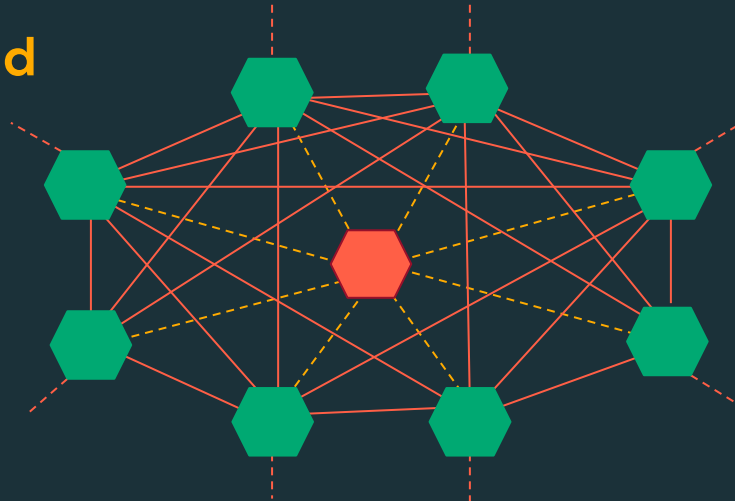
Databricks Administration Models

Balancing autonomy with complexity

Decentralized

- Business Unit
- Platform team

publish metadata/
discover data
consume data
external sharing

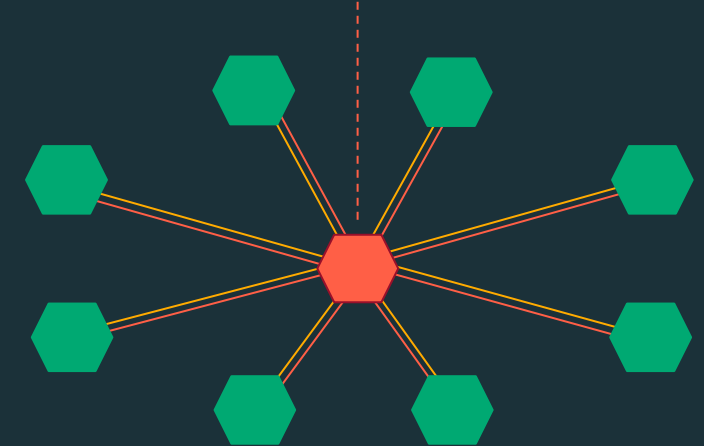


Responsible for design recommendations, but does not handle centralized implementation

Centralized

- Business Unit
- Platform team

publish and discover data
products
consume data
external sharing



Responsible not just for design, but also organizational-wide implementation

Best Practices for Provisioning and user logins

- 1 — Onboard all your users and groups to the Databricks account
- 2 — Leverage groups and use them to configure access across Databricks
- 3 — Audit using system tables to see account and workspace level events
- 4 — (AWS) Enable Unified Login and configure emergency access with MFA
- 5 — (Azure) Use Seamless Onboarding to simplify and speed up onboarding

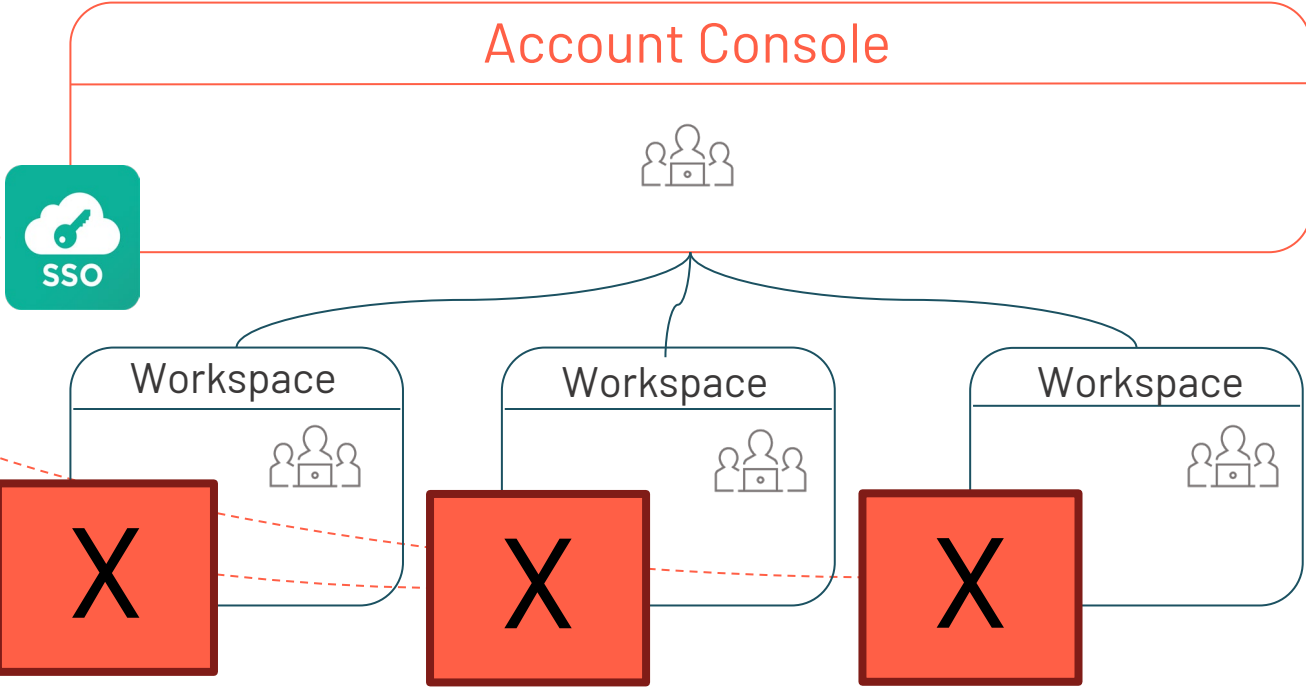


Challenge #1

How do we scale login mechanisms?



Single Sign on is still provisioned **repeatedly** with your Identity Provider



Don't do this per workspace!



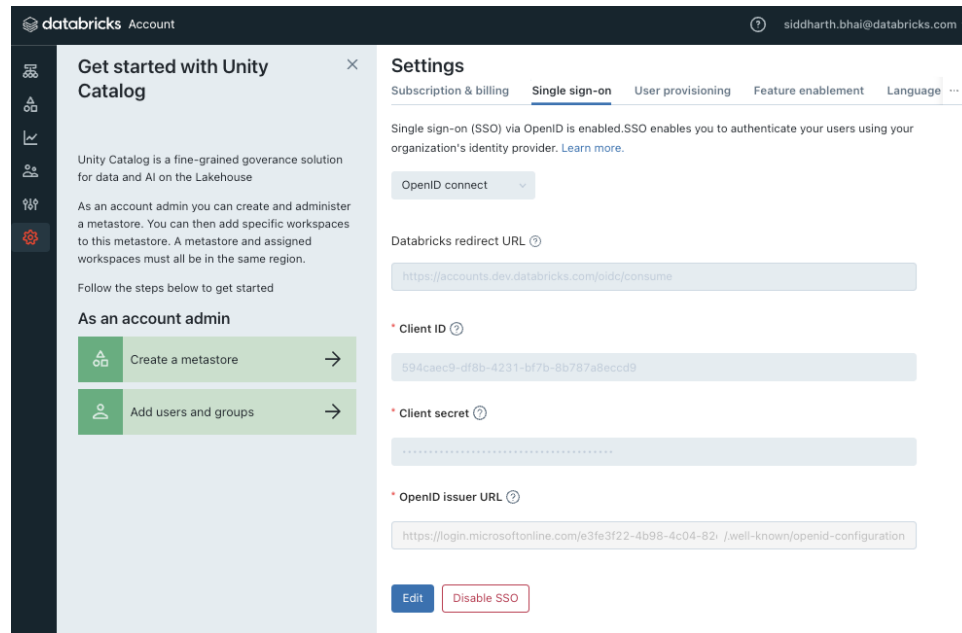
Configure SSO at account level

Unified Login makes it easier to troubleshoot

Single Sign on is provisioned **once** with your Identity Provider



Account Console



The screenshot shows the Databricks Account Console interface. On the left, there is a sidebar with navigation options. The main content area is split into two panels. The left panel is titled "Get started with Unity Catalog" and contains instructions for account admins, including buttons for "Create a metastore" and "Add users and groups". The right panel is titled "Settings" and has tabs for "Subscription & billing", "Single sign-on", "User provisioning", "Feature enablement", and "Language". The "Single sign-on" tab is active, showing a message: "Single sign-on (SSO) via OpenID is enabled. SSO enables you to authenticate your users using your organization's identity provider. Learn more." Below this, there are configuration fields: "OpenID connect" (a dropdown menu), "Databricks redirect URL" (a text input field with the value "https://accounts.dev.databricks.com/oidc/consume"), "Client ID" (a text input field with the value "594caec9-df8b-4231-bf7b-8b787a8eccd9"), "Client secret" (a text input field with a masked value), and "OpenID issuer URL" (a text input field with the value "https://login.microsoftonline.com/e3fe3f22-4b98-4c04-82- /well-known/openid-configuration"). At the bottom of the settings panel, there are two buttons: "Edit" and "Disable SSO".

And automatically applies to all **new workspaces created** in these accounts



Demo #1

Secure account and integrate with company practices

Context

AWS

Your Identity Provider

Your hardware key vendor

Technologies

- Databricks
 - Unified Login
 - User onboarding via SCIM sync
 - Emergency access with MFA

For added security, protect the account administrator with multi-factor authentication.





Settings

- Subscription & billing
- Single sign-on**
- User provisioning
- App connections
- Feature enablement
- Language settings

Single sign-on (SSO) is disabled. SSO enables you to authenticate your users using your organization's identity provider. [Learn more.](#)

OpenID connect ▾

Databricks redirect URL ⓘ

https://accounts.cloud.databricks.com/oidc/consume

* Client ID ⓘ

2902460d-7c0c-47c0-b85d-c7dcf004840d

* Client secret ⓘ

.....

* OpenID issuer URL ⓘ

https://login.microsoftonline.com/46648983-e5d5-.../.well-known/openid-configuration

Save Cancel



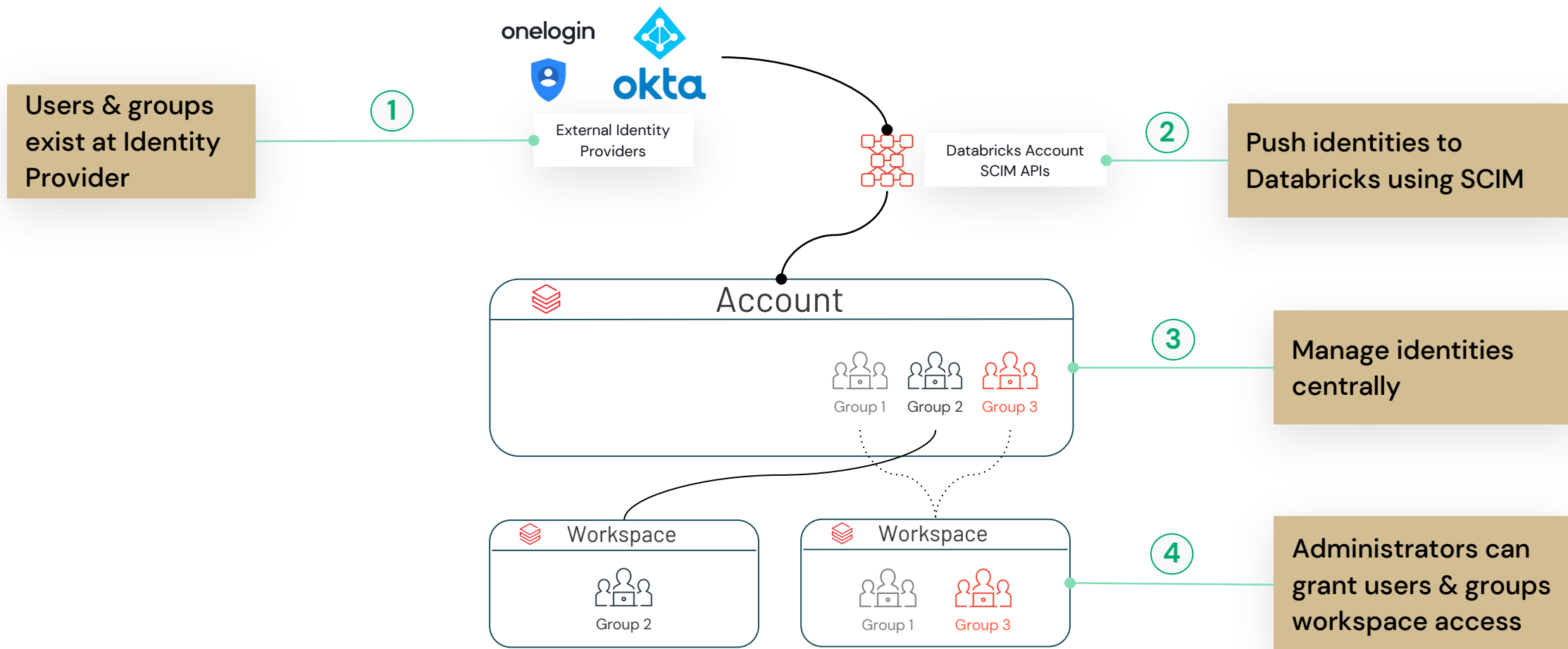
Challenge #2

How do we scale user provisioning?



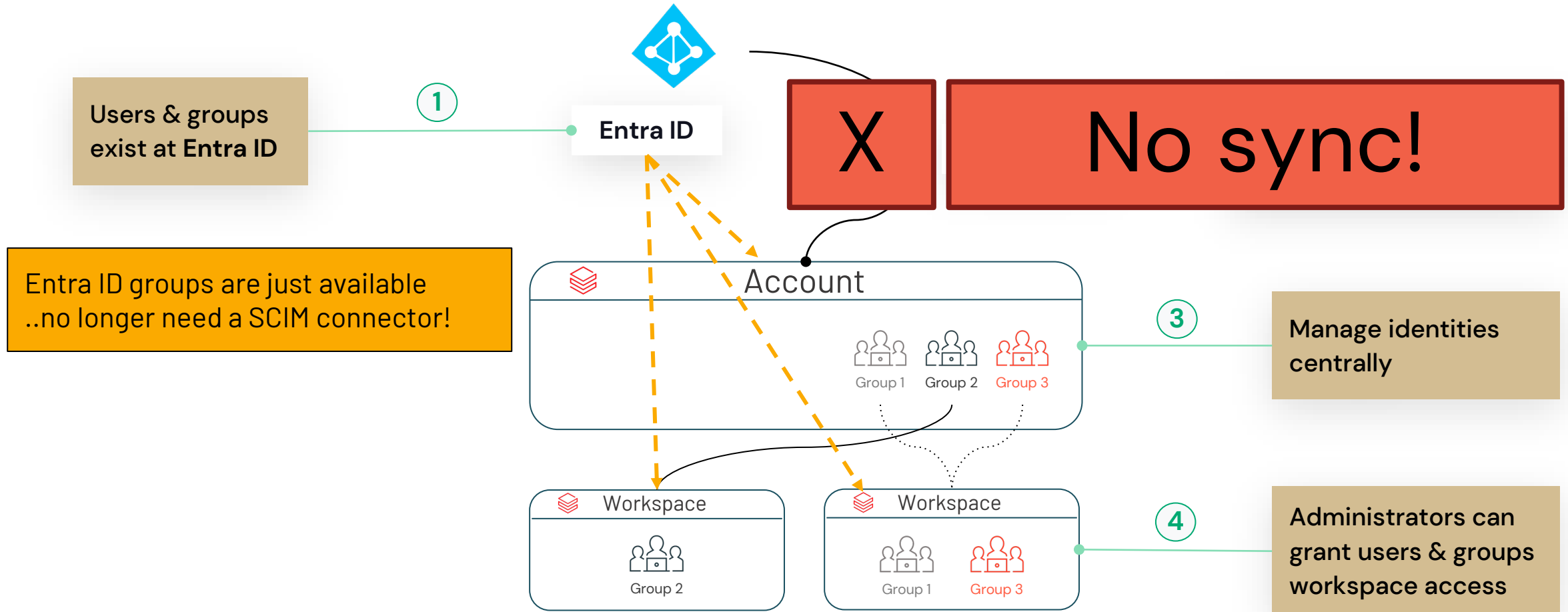
User provisioning to the account

Simplify sync pipelines to one per account



User provisioning to the account

Simplify sync pipelines to none per account

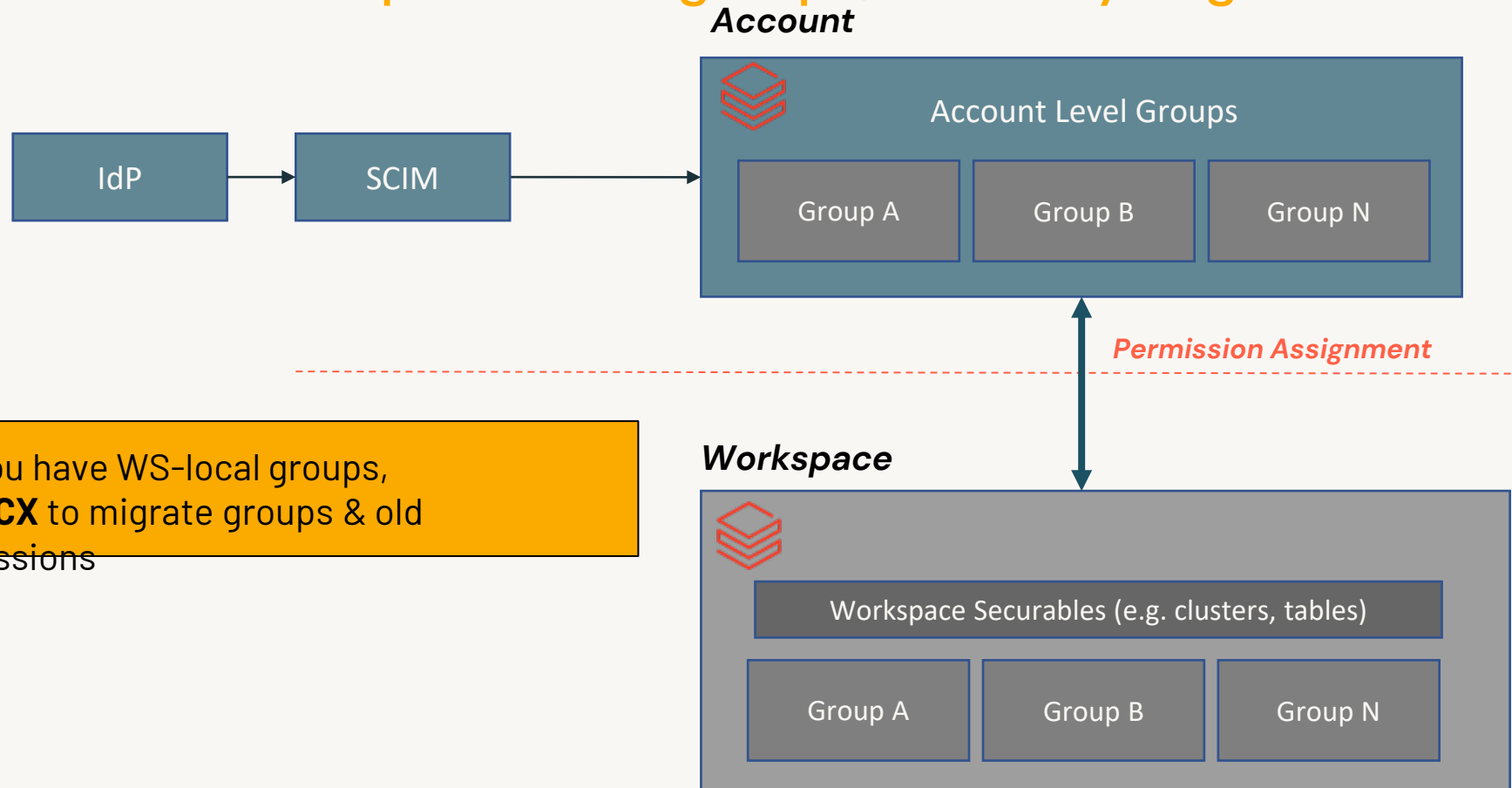


Entra ID groups are just available ..no longer need a SCIM connector!



Account level identity is the ideal

If you have workspace-local groups, how do you get there?

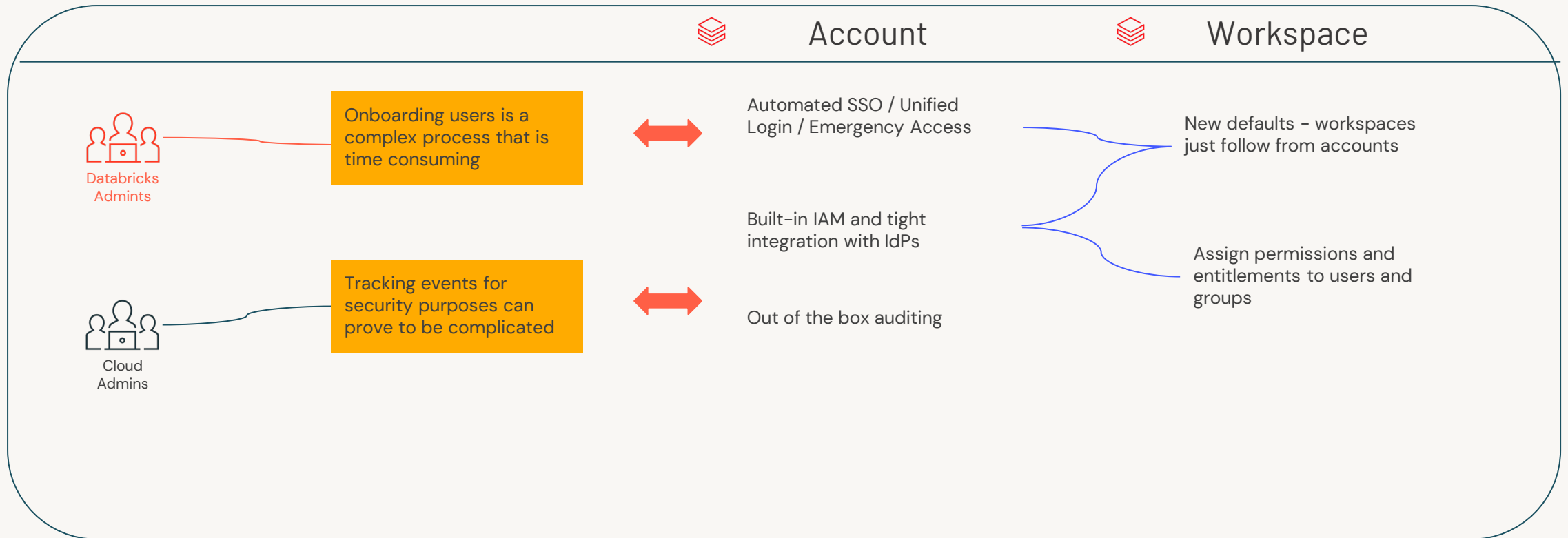


** If you have WS-local groups,
Use UCX to migrate groups & old
permissions



IAM & Administration

How Databricks and Cloud admins can partner to scale administration?





Provisioning & user logins

Best practices to provision users and have them login securely



Data isolation & external tools

Leverage proven strategies to configure data access correctly and from 3P tools



Automation & Delegation

Learn how to automate and delegate to scale Databricks usage for the long-term securely

Best Practices for Data Isolation & External Tools

- 1 • Plan your data isolation model
- 2 • Assign UC securables to specific workspaces
- 3 • Implement fine-grained access control
- 4 • Leverage Serverless networking configs and Lakehouse Federation
- 5 • Configure access via OAuth in a centralized with native integrations



Challenge #3

How do we scale user access to data?



Plan Isolation, Assign Securables, Implement ACLs

Delegation of Management (admin isolation)

Each SDLC environment has its own admin

Workspace to catalog binding

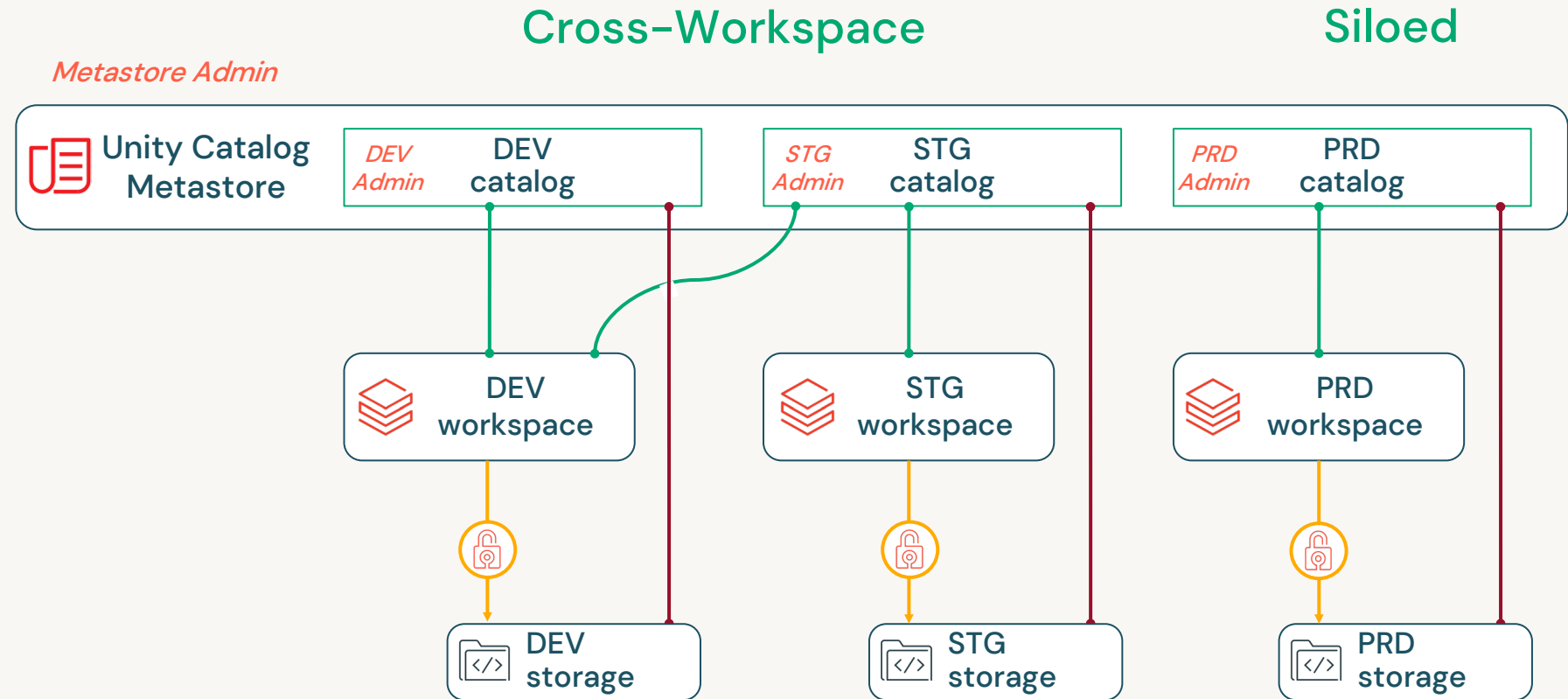
The PRD workspace is fully isolated and has its own catalog. DEV workspace has access to DEV and STG catalog

Storage isolation

This example separates the storage locations on catalog level (typically sufficient)

UC Access Control

Users should only gain access to data/metadata based on agreed access rules

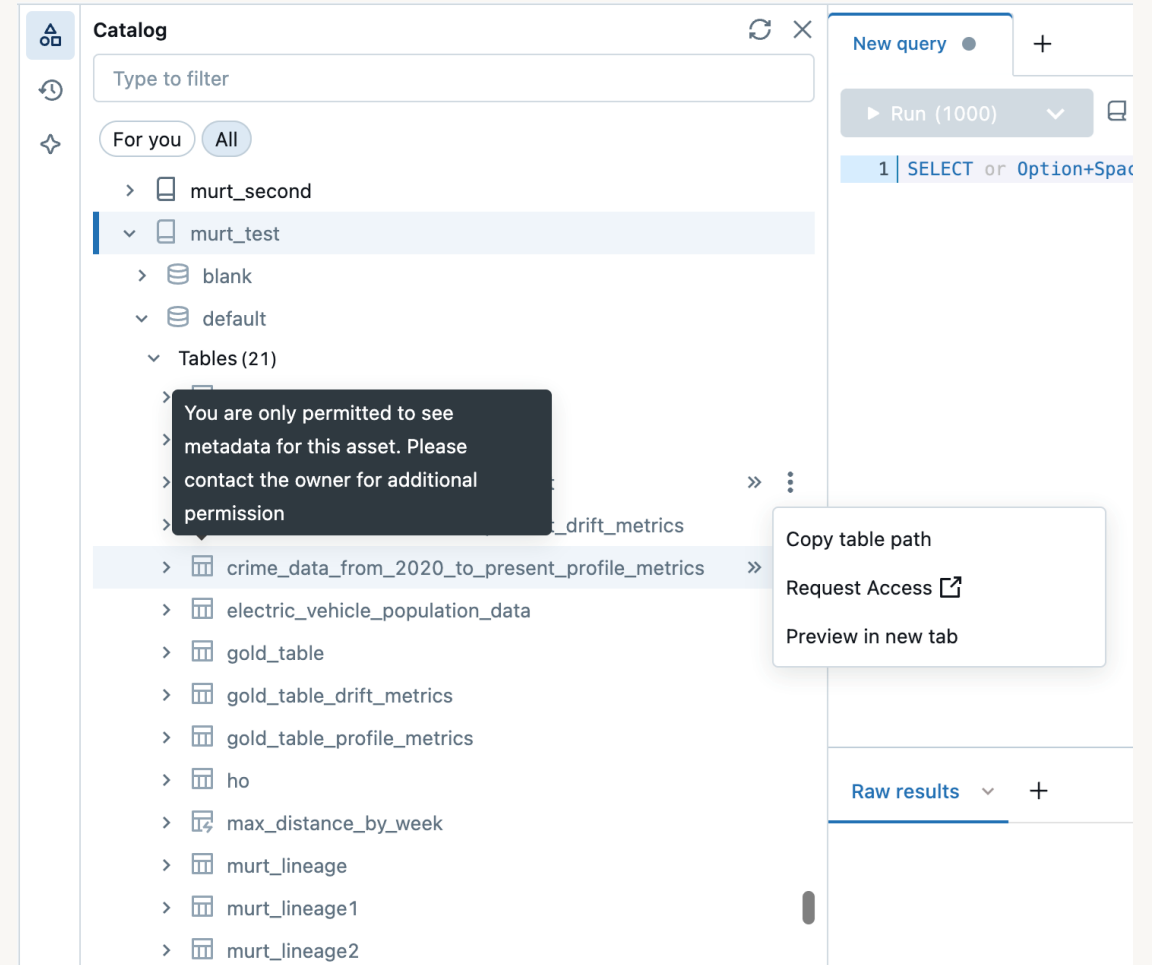


Share Metadata for Discoverability

Make **ALL** Metadata Discoverable

Grant **BROWSE** broadly across all catalogs

Allows users to view metadata, search, see lineage




Expand Use Cases via Lakehouse Federation

Use Case 1: No ETL

Use Case 2: Data migrations

Use Case 3: Distributed Analysis

Other DBs



Catalog Explorer field-eng-east

Send feedback

Add data Browse DBFS Shared Endpoint Serverless M

Catalog

Delta Sharing

External Data

Storage Credentials

External Locations

Connections

Create a new catalog

A catalog is the first layer of Unity Catalog's three-level namespace and is used to organize your data assets. [Learn more](#)

General

Authentication

* Catalog name

* Type

Foreign

Connection det:

Advanced optio

- abe_snowflake
- adb-to-bq ✓
- adrian_metastore_test
- adrian_sql
- ahc_test_sqlserver

- Pus
- -
 - Scan + project

Silob

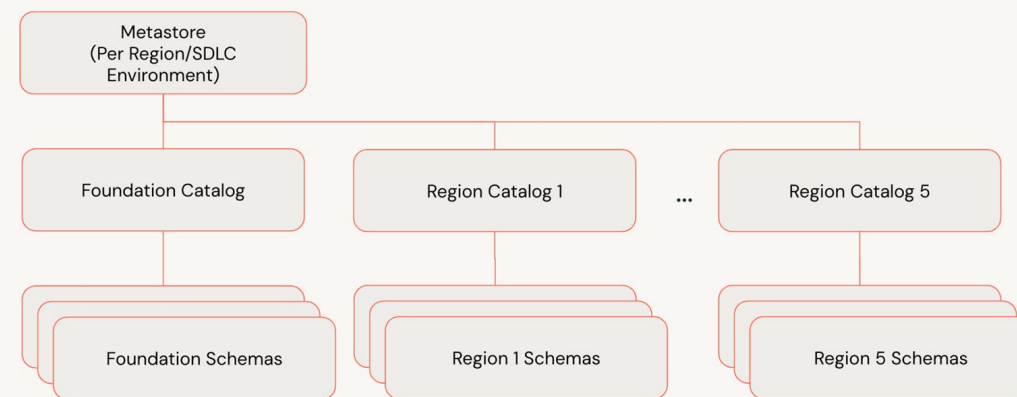


Customer Case Study

How do you start initial design and implementation?

- Single metastore per region & SDLC environment, ensuring isolation between each phase
- Each Catalog will require its own “Owner” group to act as the administrators for that Catalog – with specified default storage locations
- Workspace bindings planner for the future
- All objects within Catalogs, including Schemas, Tables, Views, Functions, etc., have their own access control policies

Each Catalog will need an identified group of owners, technically implemented via EntraID groups, with appropriate skills for managing their data assets.



```
GRANT USE ON CATALOG catalog;  
GRANT SELECT ON SCHEMA schema1;  
GRANT BROWSE ON SCHEMA schema2;
```



Object	User Can See	User Can Browse	User Can Query
catalog	✓	✗	✗
schema1	✓	✓	✓
schema2	✓	✓	✗
schema3	✗	✗	✗



Challenge #4

How do we scale users' access from their favourite tools?



Configure Access from Users' Tools

Connect apps with Databricks to allow your users to use their data in other applications with a simple sign in. [Learn more.](#)

Application	Created by	Creation date	Client ID	
 Tableau Cloud	Databricks ⓘ	-	7de584d0-b7ad-4850-b915-...	⋮
 Databricks SQL Connector	Databricks ⓘ	-	databricks-sql-connector	⋮
 dbt adapter for Databricks	Databricks ⓘ	-	dbt-databricks	⋮
 Databricks SQL ODBC	Databricks ⓘ	-	databricks-sql-odbc	⋮
 Databricks SQL JDBC	Databricks ⓘ	-	databricks-sql-jdbc	⋮
 Databricks SQL Python	Databricks ⓘ	-	databricks-sql-python	⋮
 Databricks CLI	Databricks ⓘ	-	databricks-cli	⋮
 Tableau	Databricks ⓘ	-	0464ea90-c12f-42a7-b347-c...	⋮
 Power BI	Databricks ⓘ	-	power-bi	⋮



Users Just Log In

Query the freshest data in SQL, and build **apps** and **dashboards** with **any tools** powered by the lakehouse



The screenshot shows the Tableau Desktop interface with the 'Connect' dialog box open. The dialog box is titled 'Connect' and has a 'Databricks' tab selected. The 'General' sub-tab is active, showing the 'Server Hostname' field with the value 'e2-demo-field-eng.cloud.databricks.com'. Below this, there is a field for '9d50b' and a dropdown menu. The 'Advanced' sub-tab is also visible. The background shows the Tableau 'Open' dialog box with 'Open a Workbook' and 'More Accelerators' options.



Sign in to continue to Databricks

moe.derakhshani@databricks.com

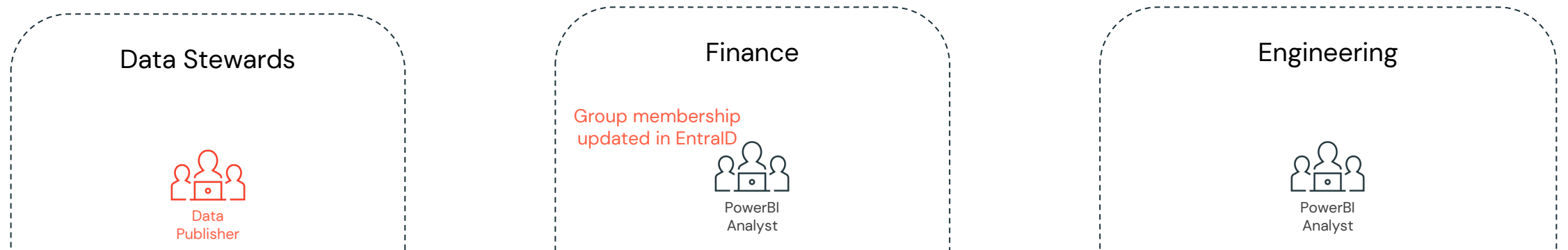
.....

Sign In



Demo #2

Databricks Seamless – updating EntraID flows through automatically, updating UC grants, and reflected in the BI layer



Data access policies based on group membership seamlessly flow through to PowerBI

Grant access to column masked by policy

Grant access to the table

Column-level policy: only show sales price to groups Data Stewards and Finance



- New
- Workspace
- Recents
- Catalog
- Workflows
- Compute
- SQL
- SQL Editor
- Queries
- Dashboards
- Genie
- Alerts
- Query History
- SQL Warehouses
- Data Engineering
- Job Runs
- Data Ingestion
- Delta Live Tables
- Machine Learning
- Playground
- Experiments
- Features
- Models
- Serving
- Marketplace
- Partner Connect
- Collapse menu

Unity Catalog : Support for Identity Columns, Primary + Foreign Key Constraints

To simplify SQL operations and support migrations from on-prem and alternative warehouse, Databricks Lakehouse now give customers convenient ways to build Entity Relationship Diagrams that are simple to maintain and evolve.

These features offer:

- The ability to automatically generate auto-incrementing identity columns. Just insert data and the engine will automatically increment the ID.
- Support for defining primary key
- Support for defining foreign key constraints

IDENTITY COLUMNS

- Define **IDENTITY** column on a table
- Delta can automatically generate unique integer values when new rows are added to the table with **IDENTITY** columns
- Users can also explicitly insert values for **IDENTITY** columns



PRIMARY + FOREIGN KEY CONSTRAINTS

- Declare unenforced Primary and Foreign keys with **ALTER TABLE**
- Visible in **INFORMATION_SCHEMA** and **DESCRIBE TABLE**
- Allow end users to understand **relationships between tables**

GOAL: Enable data quality and easy table relationship discovery for tools and users that are not familiar with the data model

Note that as of now, Primary Key and Foreign Key are informational only and then won't be enforced.

Use case

Defining PK & FK helps the BI analyst to understand the entity relationships and how to join tables. It also offers more information to BI tools who can leverage this to perform further optimisation.

We'll define the following star schema:

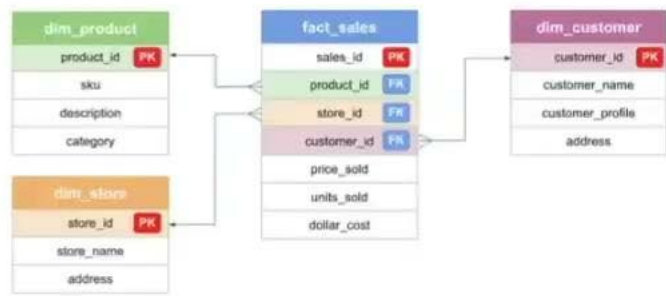
- dim_store
- dim_product
- dim_customer

And the fact table containing our sales information pointing to our dimension tables:

- fact_sales

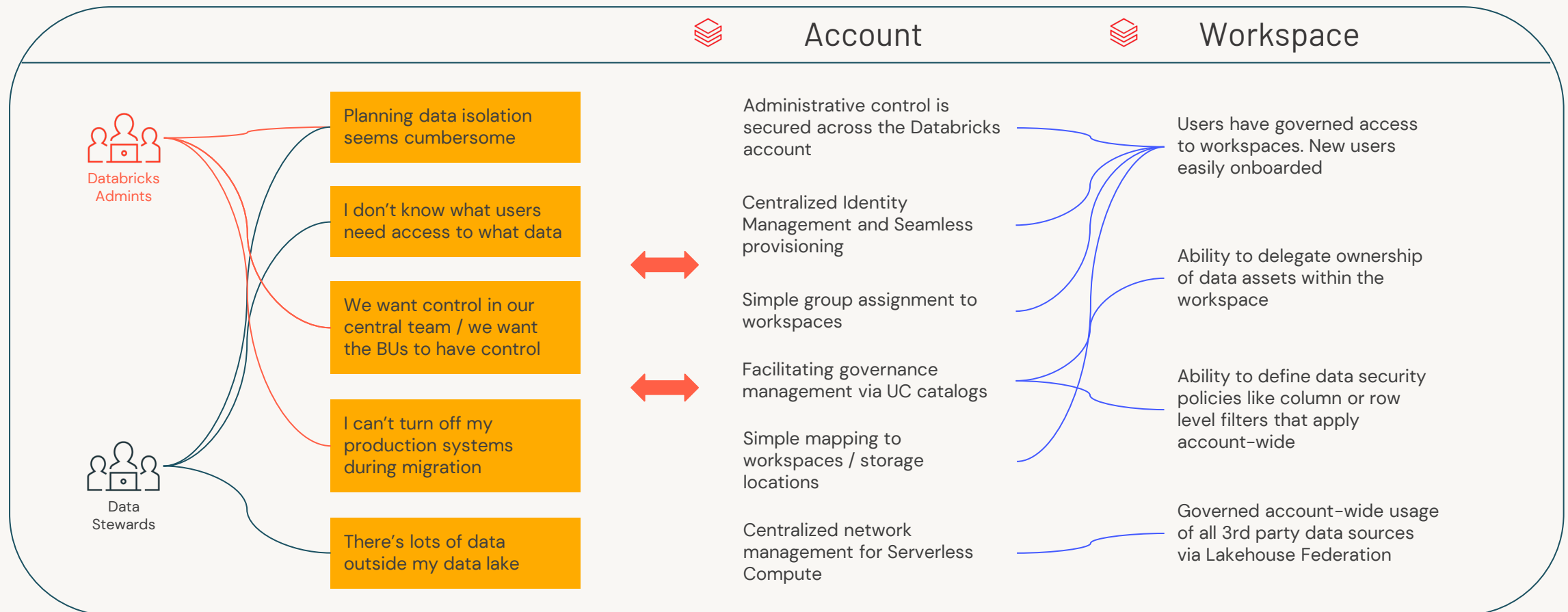
Requirements:

- PK/FK requires Unity Catalog enabled (Hive Metastore is not supported for FK/PK)
- DBR 11.1



Data Isolation and External Tools

Personas' roles in setting up governance with Databricks





Provisioning & user logins

Best practices to provision users and have them login securely



Data isolation & external tools

Leverage proven strategies to configure data access correctly and from 3P tools



Automation & Delegation

Learn how to automate and delegate to scale Databricks usage for the long-term securely

Challenge #5

How do we scale ongoing Databricks usage?



Best Practices for Automation & Delegation

- 1 • Design workspaces as collaboration boundaries
- 2 • Trust and limit the number of workspace administrators
- 3 • Use Service Principals to run administrative tasks and production workloads
- 4 • Leverage the account console to configure settings that apply to all workspaces
- 5 • Apply Terraform modules to standardize Databricks deployments



Use Service Principals

Choose the right type of SP and select who may use them

Databricks workflows

- Run as service principals
- SP:Manager and SP:User

Databricks automation

- Generate OAuth secret
- Use OAuth to get DB-token

(Azure) multiple services

- (Optional) Azure-backed SP
- (Optional) Use AAD tokens

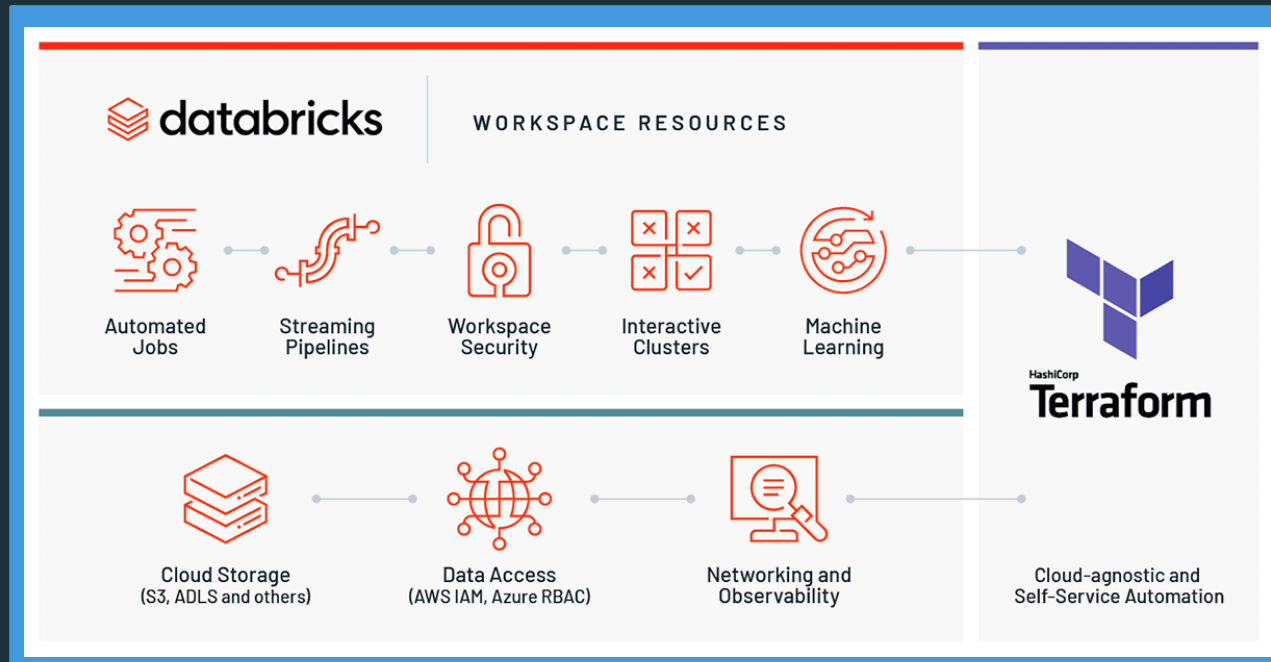
The image displays two screenshots of the Databricks interface, both for a service principal named "Ingestion SP - June 2024".

The top screenshot shows the "Secrets" tab. It features a "Delete" button in the top right corner. Below the title, there are tabs for "Configurations", "Permissions", and "Secrets". The "Secrets" section is titled "OAuth secrets" and includes a link to "Learn more". Below this, there is a table with two columns: "ID" and "Created at". A single row is visible with the ID "93693db58e7d39ed9a5bd2a0209d0fd49f8d51f36e417886b752386cc8369825" and the creation time "today at 11:14 PM". A trash icon is located to the right of the row.

The bottom screenshot shows the "Permissions" tab. It also has a "Delete" button in the top right corner. Below the title, there are tabs for "Configurations", "Permissions", and "Secrets". A search bar labeled "Search principa..." is present. A "Grant access" button is located on the right side. Below the search bar, there is a table with two columns: "Principal" and "Roles". A single row is visible with the principal "Siddharth Bhai (siddharth.bhai@databricks.com)" and the role "Service principal: User". A vertical ellipsis menu icon is located to the right of the row.

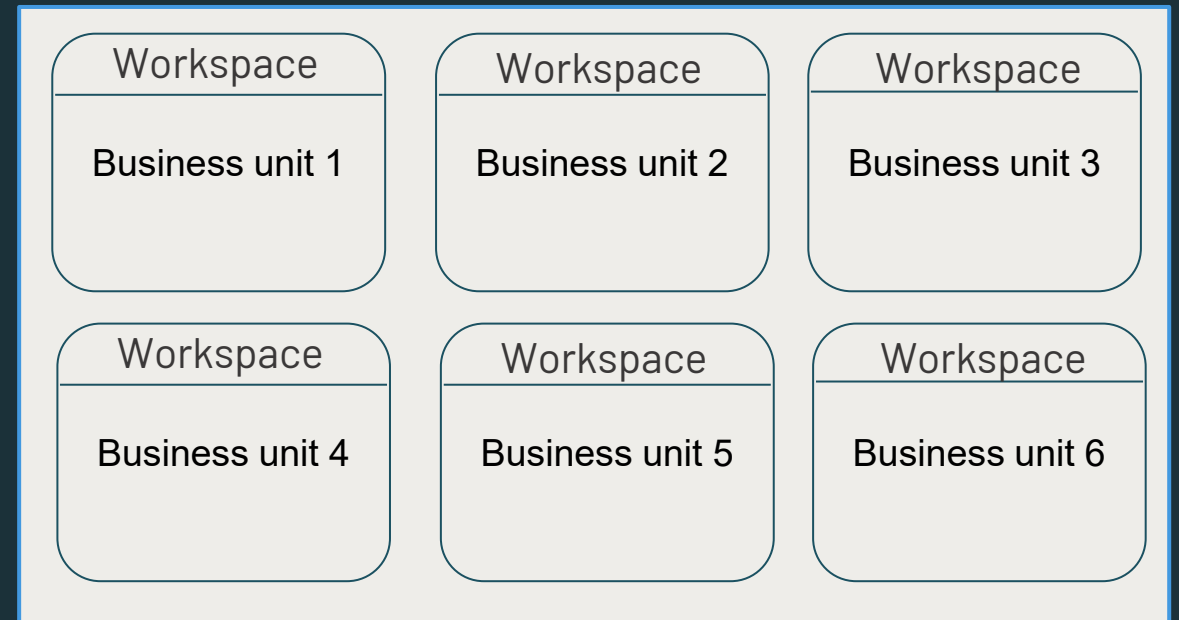
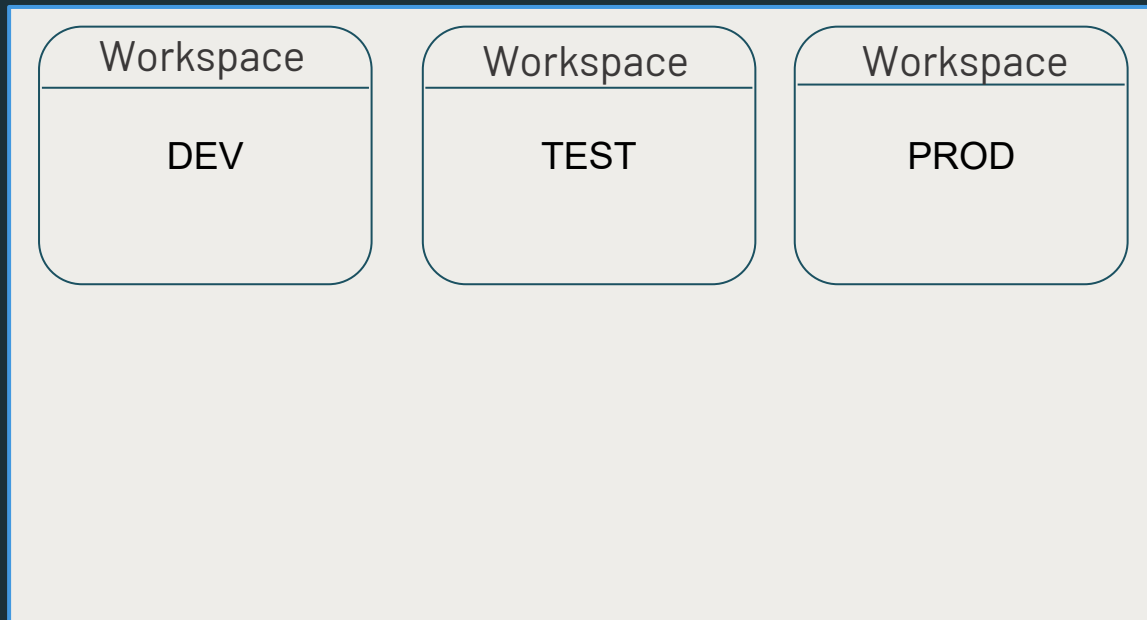
Apply Terraform modules

Enforce a standardized approach and adopt best practices



- Easily maintain, manage and scale your infrastructure
 - DevOps loves Terraform
- 30 Million+ installations
 - Generally Available
- Terraform Registry Module
 - 30+ reusable examples

Workspaces as collaboration boundaries



Demo #3

Scale usage and share data insights faster

Context

2 stakeholders:

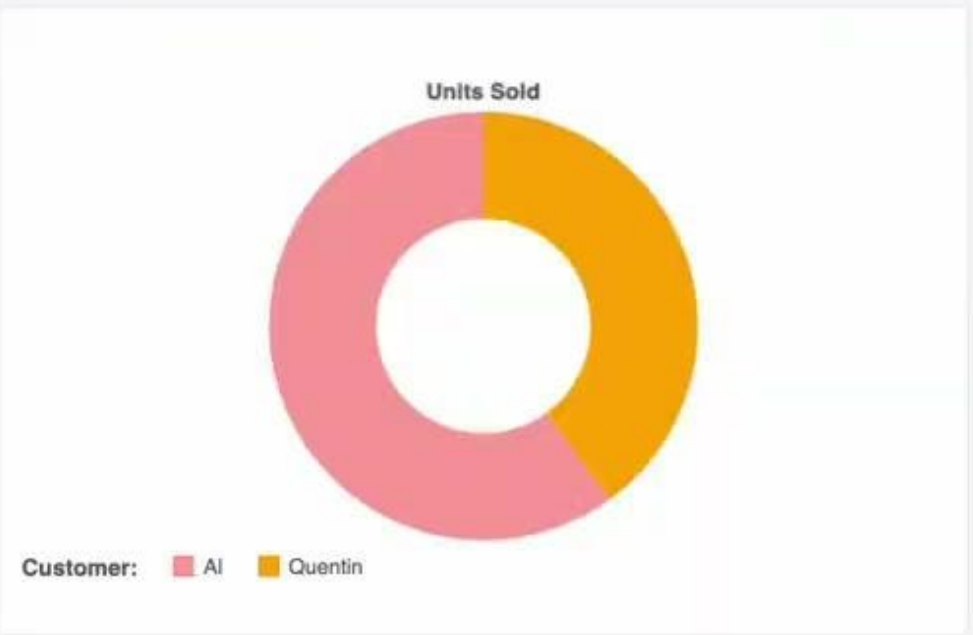
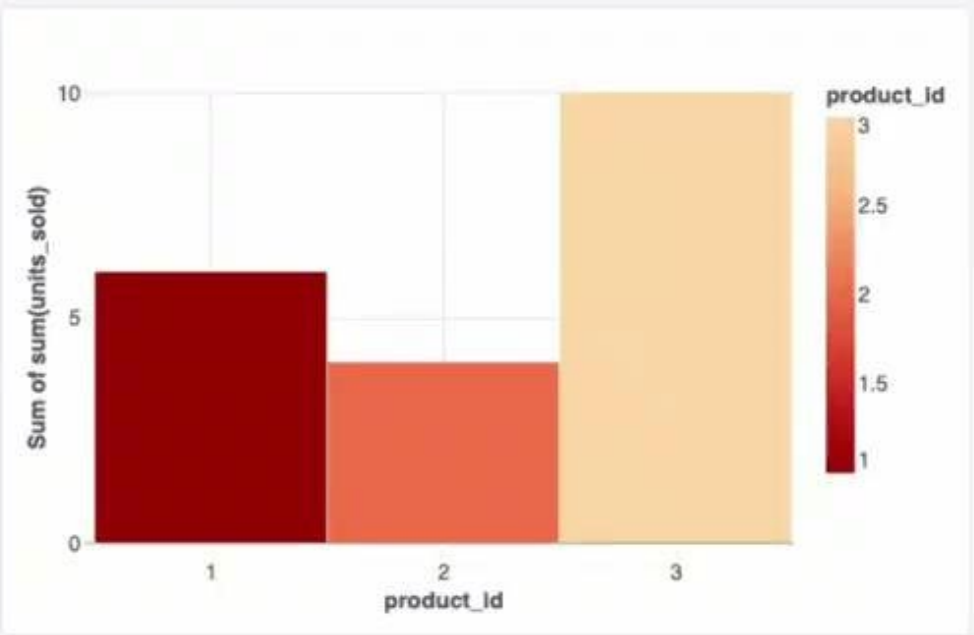
- Data.publisher ([Data Stewards](#) group)
- Sales associates ([Sales - Americas - US - WA](#) group)

Technologies

- Databricks
 - Dashboards
 - Sharing
- Identity Provider
 - User login

Our Data.publisher will publish a Dashboard and share it with a sales group. No one from sales uses Databricks yet. A sales associate will log in with their company login credentials and just access the dashboard with no waiting.





Call to action

How you can scale – to 500K+ users and beyond



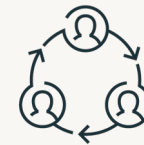
Provisioning & user logins

AWS: Set SSO open to all
Azure: Seamlessly onboard



Data isolation & external tools

Plan isolation and sharing
Use OAuth, not passwords



Automation & Delegation

Use Service Principals well
Encourage sharing artifacts

