OneLake Security - Centralized Data Security for Microsoft Fabric



Gabi Muenster
Jochen Schröer

Intros



Gabi Muenster
Microsoft
Principal Program Manager
Azure Data CAT - Fabric



Jochen Schröer
Microsoft
Principal Architect
Industry Solutions

Agenda

1

Setting the scope

2

OneLake Security Concepts

3

End-to-End Demo

4

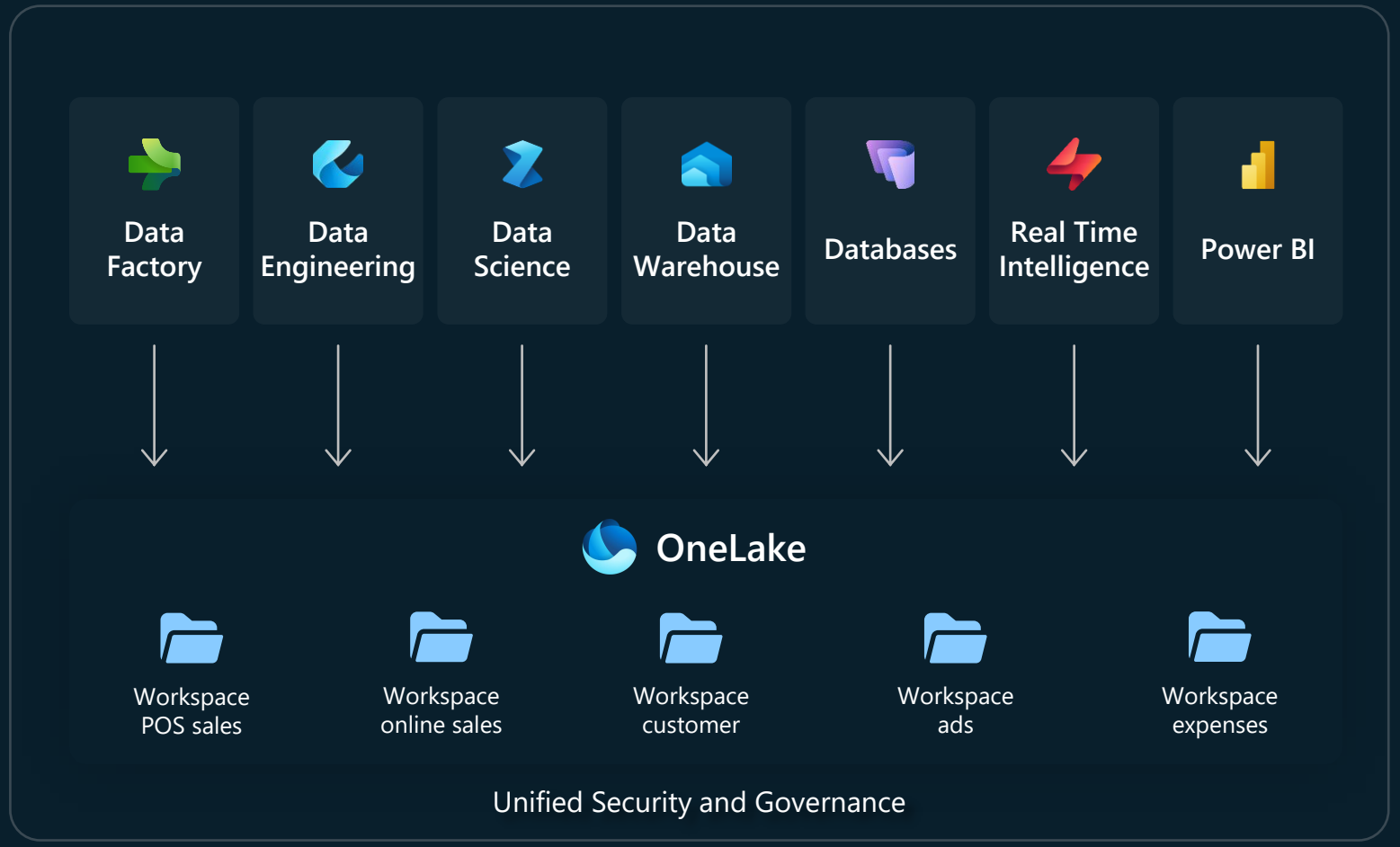
OneLake Security Features Recap

5

What's next?

A single unified SaaS data lake

"No Silos"



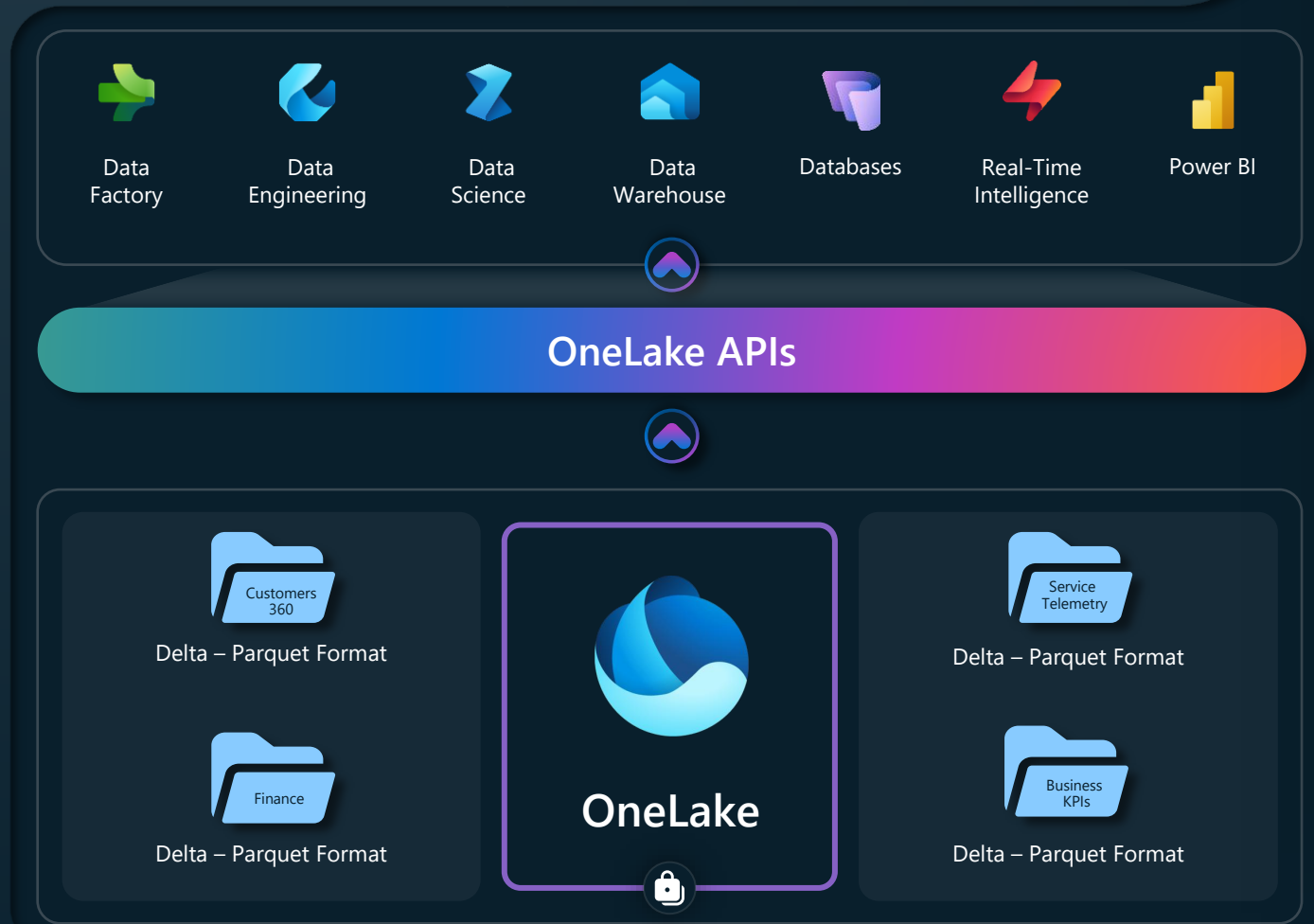
Provisioned automatically with the tenant.

Any data in OneLake works with out-of-the-box governance such as data lineage, data protection, certification, catalog integration, etc. All data is ultimately under the control of a tenant admin.

OneLake enables distributed ownership. Different workspaces allow different parts of the organization to work independently while still contributing to the same data lake. Each workspace can have its own administrator, access control, region and capacity for billing.

OneLake Security

Defined once, enforced everywhere



Define security roles in OneLake using powerful features like table, row, or column level security.

Data is secured consistently across experiences inside and outside of Fabric.

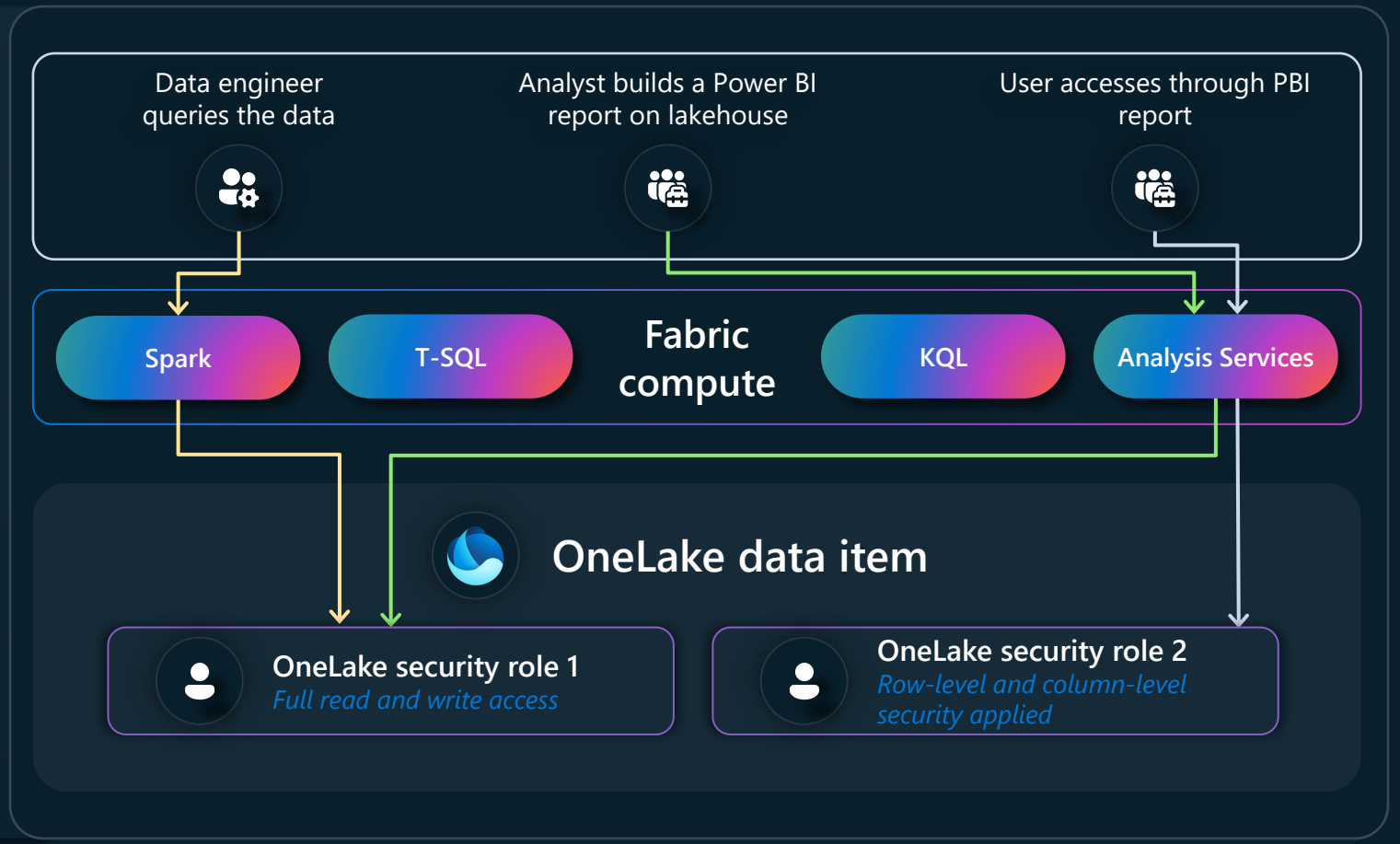
OneLake security roles can be managed in a single place, providing end to end coverage for data access for your entire data estate.

OneLake Security

03 Users access data through different Fabric engines

02 The permissions are checked when specific users are accessing data from the data item

01 Data owner sets up roles with defined permissions and assigns user groups to each role



Industry Needs for OneLake Security

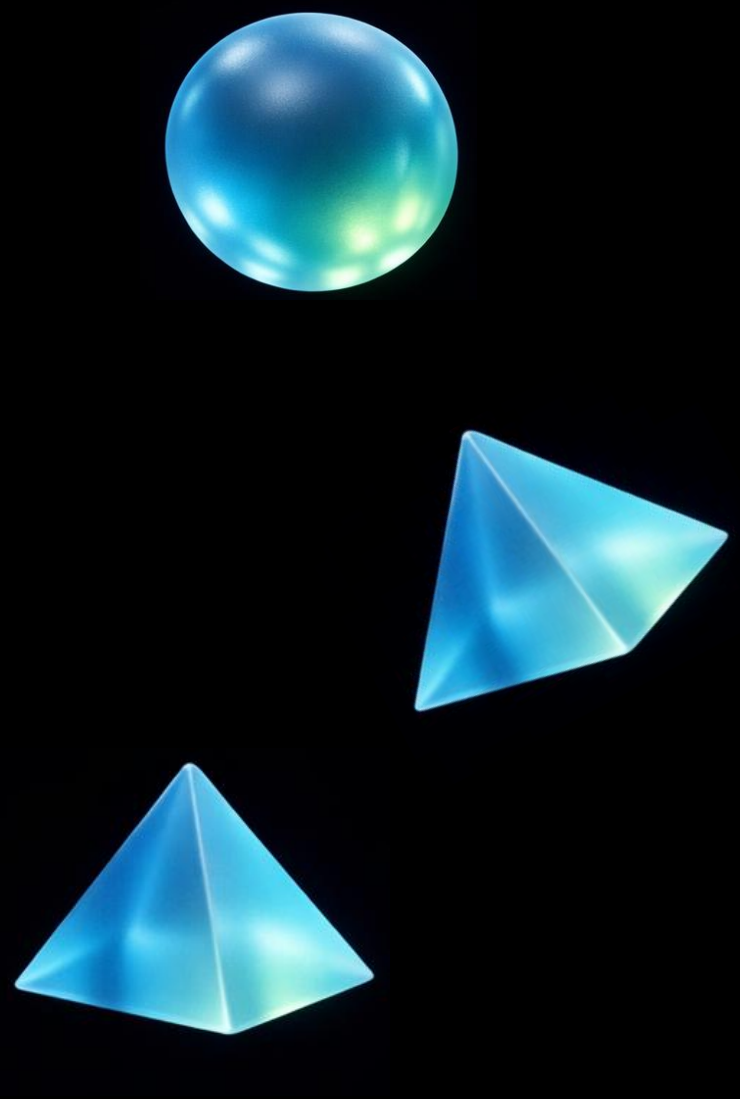
Why industries — like finance, insurance, healthcare — cannot compromise on data platform security

01 DORA - ICT risk & resilience (EU)

02 GDPR - data minimization, lawful access

03 NIS2 - critical-entity obligations

A large European insurer's OneLake adoption confirmed that enterprise rollout is gated on private endpoints, OneLake Security roles, customer-managed keys - all in place from day one, not retrofitted.



Fabric Security Concepts

Fabric Security Model

Control Plane Permissions & Introducing OneLake Security

Control Plane (Workspace + Items)

Workspace roles & item permissions from previous slides

Regulates:

- Which endpoints users can access
- What operations users can perform on items
- Administrative actions (create, delete, share)

Data Plane (OneLake)

NEW: OneLake Security

Permissions stored at the lake level

Regulates:

- Access to tables, columns, rows, folders
- **Security is in the data, not the engine**

Lakehouse

Warehouse

Eventhouse

Semantic Model

OneLake

`Delta / Parquet`

`Iceberg / Parquet`

Security enforced here

Fabric Security Model

__Workspace__ Roles & Membership

WORKSPACE



Admin

Full control





Member

Manage & build





Contributor

Create & edit





Viewer

Read only





Fabric UX Portal

Workspace Settings → Manage Access



REST API

`POST /workspaces/{id}/roleAssignments`

Fabric Security Model

__Workspace__ Role Privileges

 **Admin**

 **Member**

 **Contributor**

 **Viewer**

Privilege	Admin	Member	Contrib	Viewer
Add admins	✓	✗	✗	✗
Add members	✓	✓	✗	✗
Edit OneLake security	✓	✓	✗	✗
Write data & create items	✓	✓	✓	✗
Read data in OneLake	✓	✓	✓	via roles*
Update/delete workspace	✓	✗	✗	✗

** Viewers can access data through OneLake security roles*



Fabric UX Portal

Workspace Settings → Manage Access



REST API

`POST /workspaces/{id}/roleAssignments`

Fabric Security Model

*Roles Grant Permissions on **Items***

WORKSPACE ROLES

- Admin
- Member
- Contributor
- Viewer

PERMISSION BUNDLES

		A	M	C	V
Read	View item metadata	●	●	●	●
ReadData	See data in SQL	●	●	●	●
ReadAll	See data in OneLake	●	●	●	○
Write	Create & edit items	●	●	●	○
Reshare	Share with others	●	●	○	○

DATA ITEMS


Lakehouse


Warehouse


Eventhouse


Semantic Model

Key Concepts

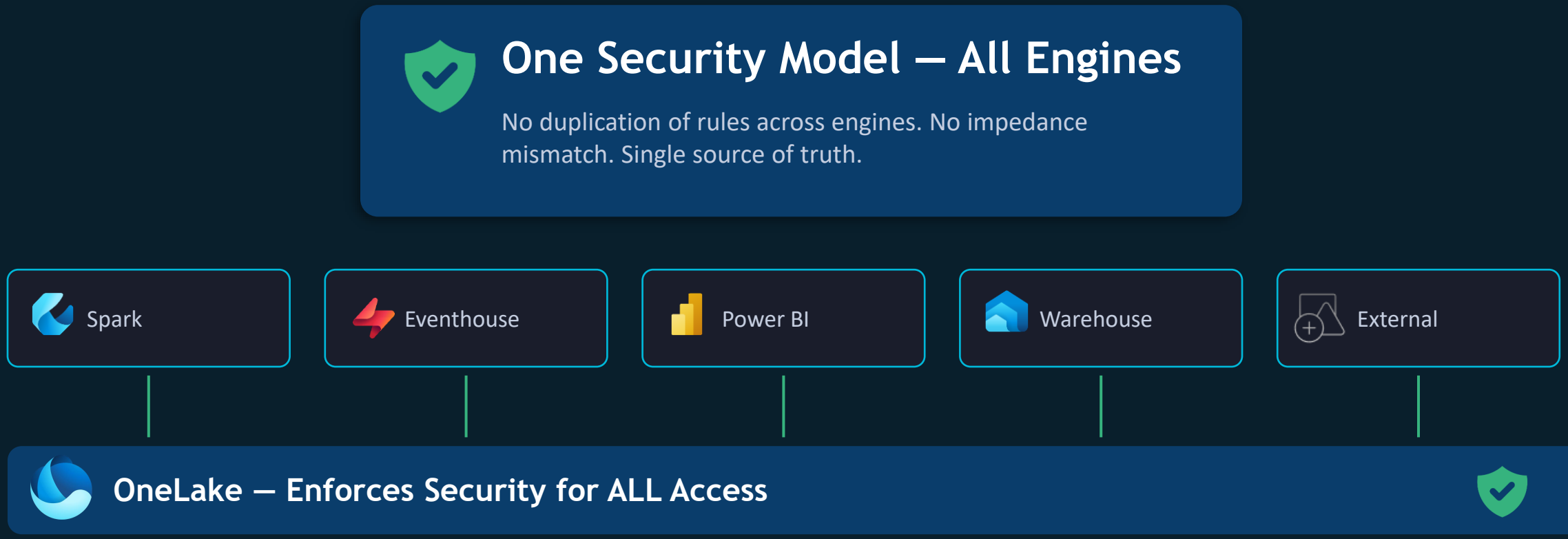
Cannot explicitly remove role-entangled permissions

✚ Explicit Item Permissions

Add **additional** permissions to specific items for specific principals.
Can be workspace members or external.
These CAN be removed.

Fabric Security Model

OneLake Security — Security in the Data, Not the Engine



Fabric Security Model

OneLake Security — Security in the Data, Not the Engine

One Security Model — All Engines

No duplication of rules across engines. No impedance mismatch. Single source of truth.

Spark | Eventhouse | Power BI | Warehouse | External

OneLake — Enforces Security for ALL Access

- OneLake security is the ONLY security for data access
- Control plane security is NOT required for data access
- Configurable via UX or REST API, per item

Fabric Security Model

OneLake Security Role Anatomy

Each item has its own OneLake security roles (different from workspace roles)

OneLake Security Role

WHO

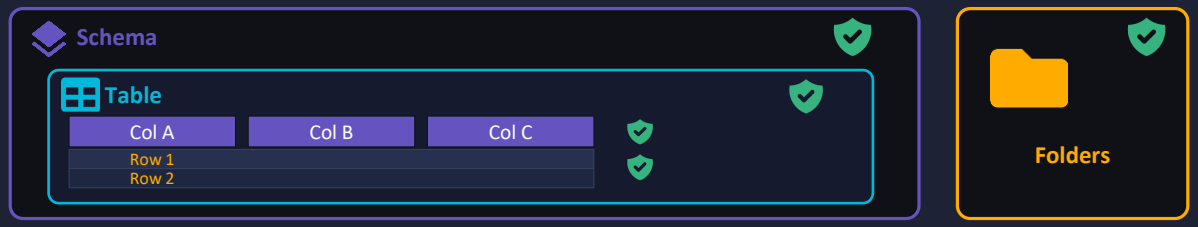
Security principals (users, service principals, groups)
Virtualized members — e.g., "anyone with Read permission"

WHAT

Permission type applied (e.g., Read)

TARGET

Sub-objects the rules apply to — each level can be independently secured:



Multiple Roles?



= UNION of all permissions

If a principal is in multiple roles, they get the combined access of all roles.

Configuration

UX REST API *Per item*

Fabric Security Model

DefaultReader Role — Minimizing Security Maintenance

When an item is created, a default role is automatically provisioned to streamline access:

DefaultReader Role

Auto-created on item creation

WHO: Virtualized — principals with **ReadAll** control plane permission

WHAT: Read permission

TARGET: The entire artifact (all tables, folders, data)

Net Effect

- Users with Read permission can read all data automatically
- Adding/removing Read permission keeps this role in sync (virtualized membership)
- Role is NOT entangled — admins can edit or delete it
- If unchanged, acts as seamless bridge between control plane & data plane access

Goal: Minimize maintenance. By defaulting to a virtualized membership based on Read permission, the system stays in sync without manual intervention. Customize when needed — the role is fully editable.

Fabric Security Model

Complete Security Architecture

CONTROL PLANE

Admin | Member | Contrib | Viewer

UX + API

Item Permissions

Role-entangled (locked) + Explicit (removable)

Endpoints, operations, admin actions

 Lakehouse

 Warehouse

 Eventhouse

 Semantic Model

DATA PLANE — OneLake Security

OneLake Roles

WHO
WHAT (Read)
TARGET (tables, cols, rows)

DefaultReader

Auto-created
Virtualized membership
Editable / deletable

All Engines Comply

Fabric engines
External engines
Enforced by the lake

Delta/Parquet | Iceberg/Parquet

UX + API | Per item | Union of roles

Fabric Security Model

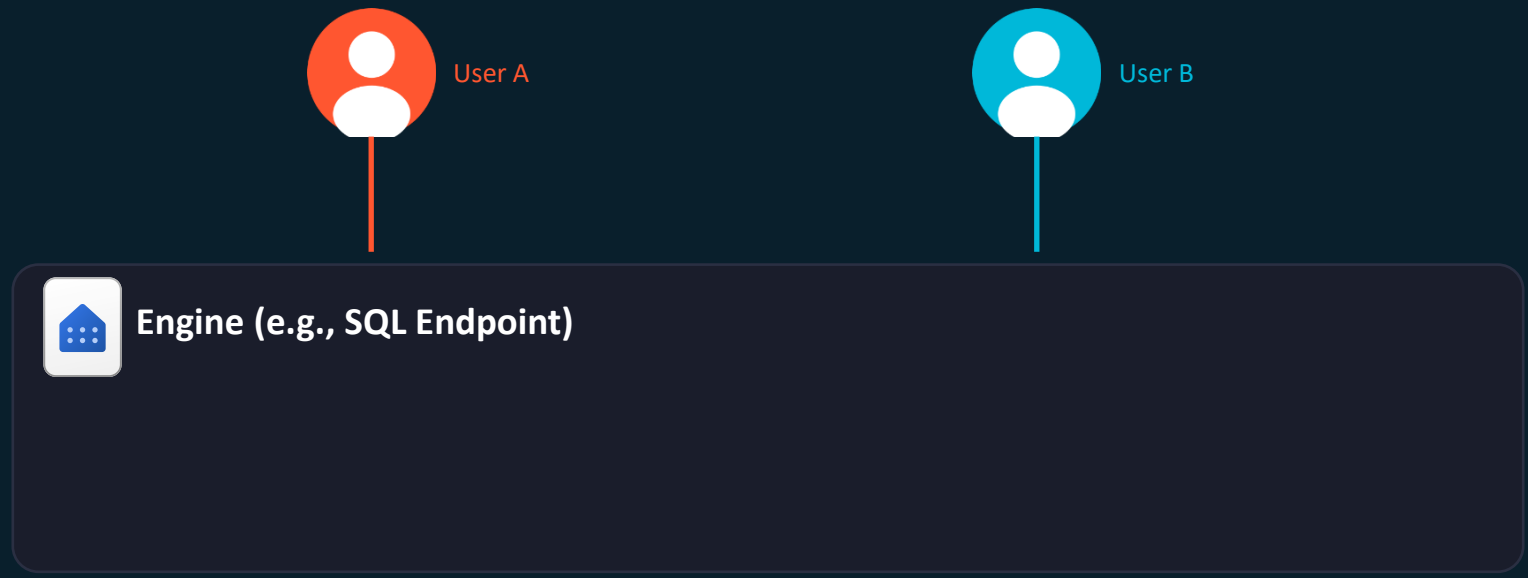
Engine Integration — Passthrough Mode

Passthrough

Control plane access — as the user

Data plane access — as the user

User identity flows straight through the engine to OneLake



Fabric Security Model

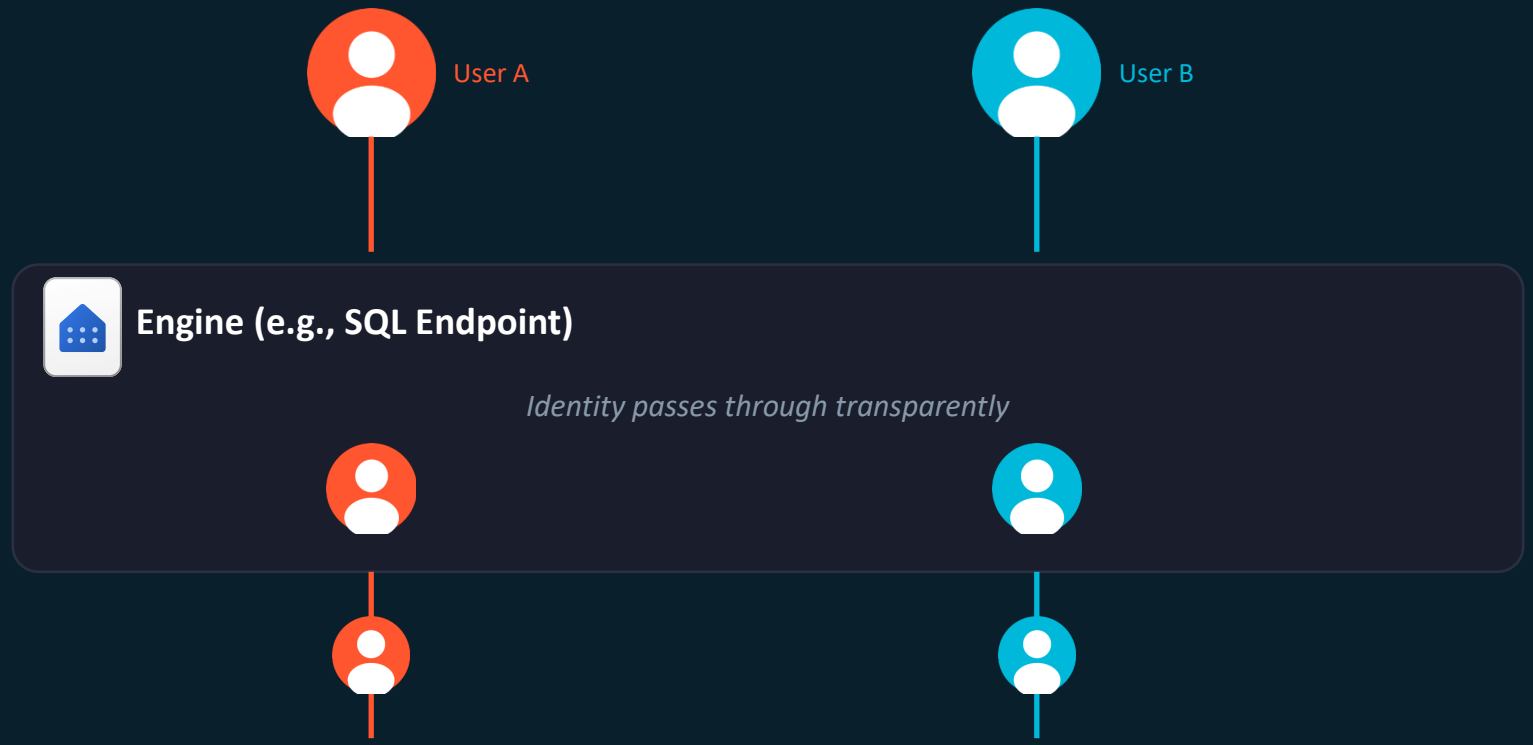
Engine Integration — Passthrough Mode

Passthrough

Control plane access — as the user

Data plane access — as the user

User identity flows straight through the engine to OneLake



Fabric Security Model

Engine Integration — Passthrough Mode

Passthrough

Control plane access — as the user

Data plane access — as the user

User identity flows straight through the engine to OneLake

Key Properties

Each user's identity reaches the lake directly

OneLake security evaluates each user individually

No engine-level security needed

Consistent access across all engines



Fabric Security Model

Engine Integration — Delegated Mode

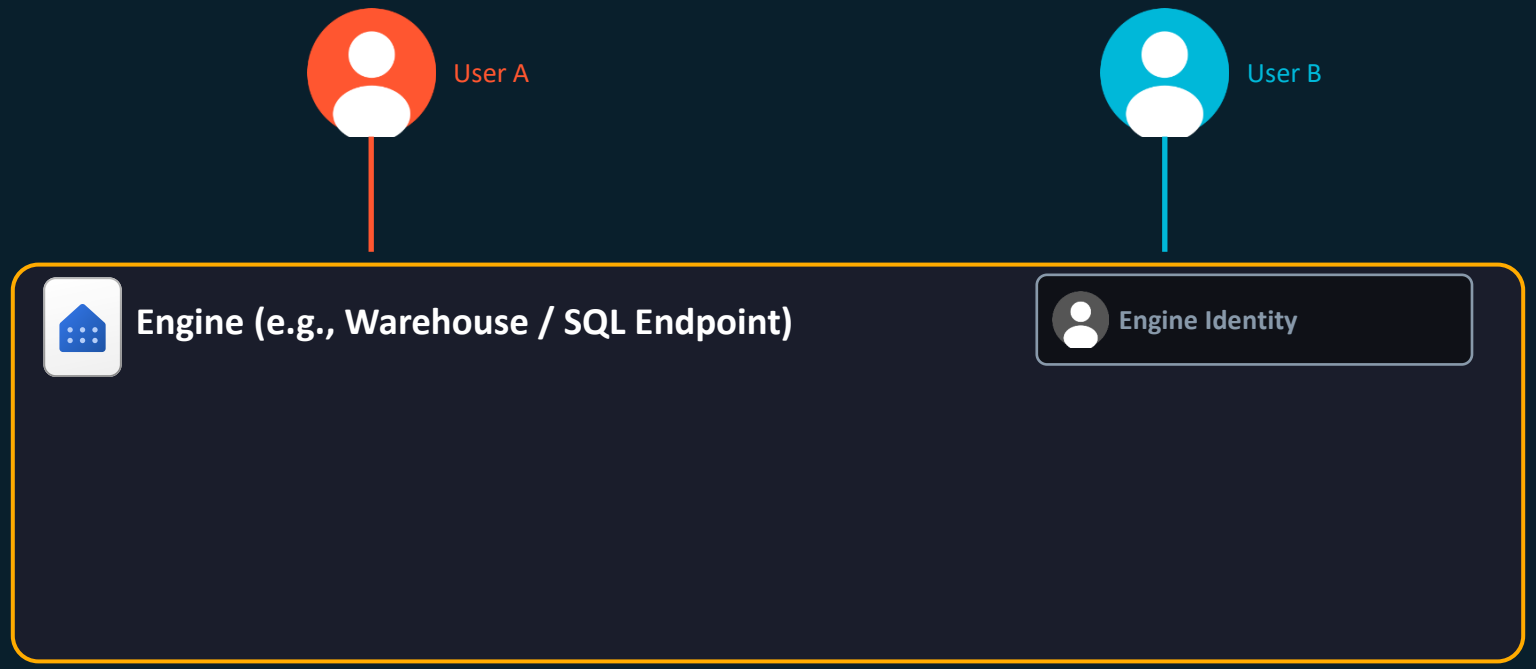
Delegated

Control plane access — as the user

Data plane access — as the engine's fixed identity

Engine identity must have permissions on the data

Engine adds its own security layer (e.g., GRANT/DENY) for per-user discrimination



Fabric Security Model

Engine Integration — Delegated Mode

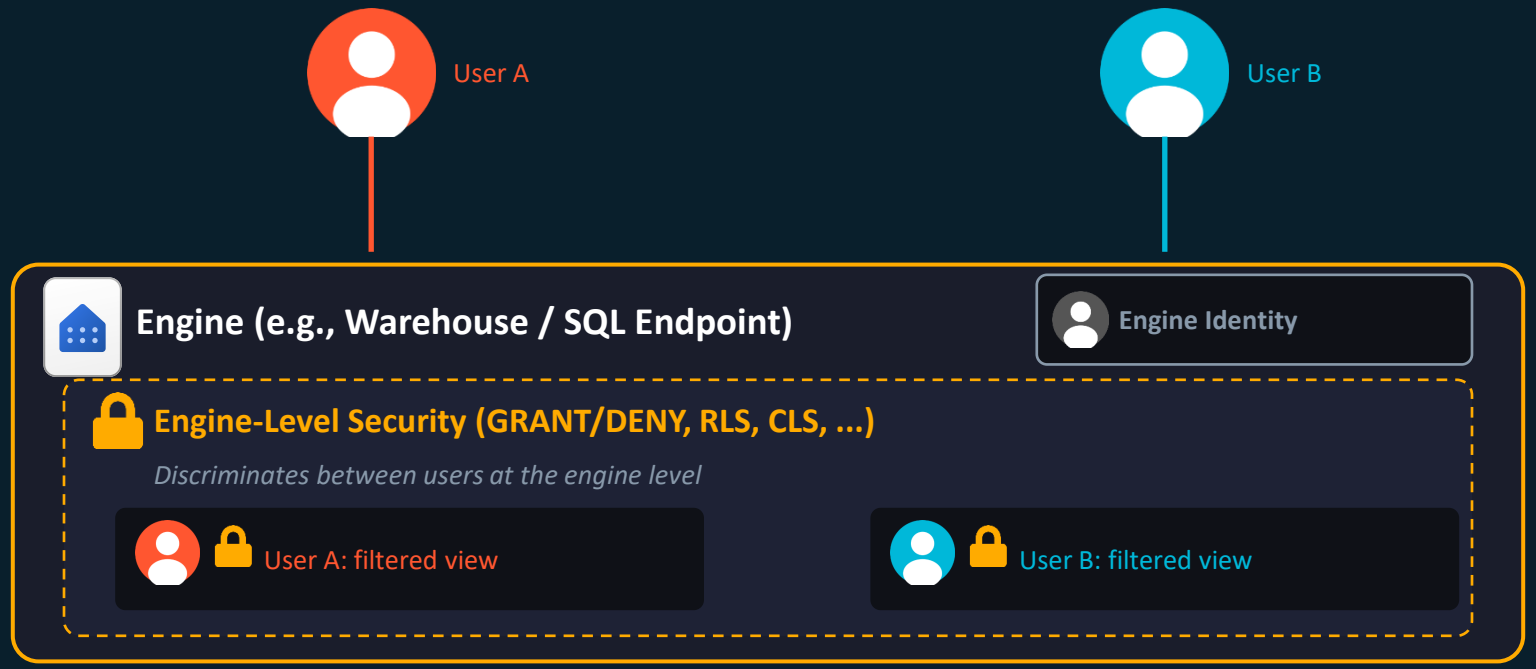
Delegated

Control plane access — as the user

Data plane access — as the engine's fixed identity

Engine identity must have permissions on the data

Engine adds its own security layer (e.g., GRANT/DENY) for per-user discrimination



Fabric Security Model

Engine Integration — Delegated Mode

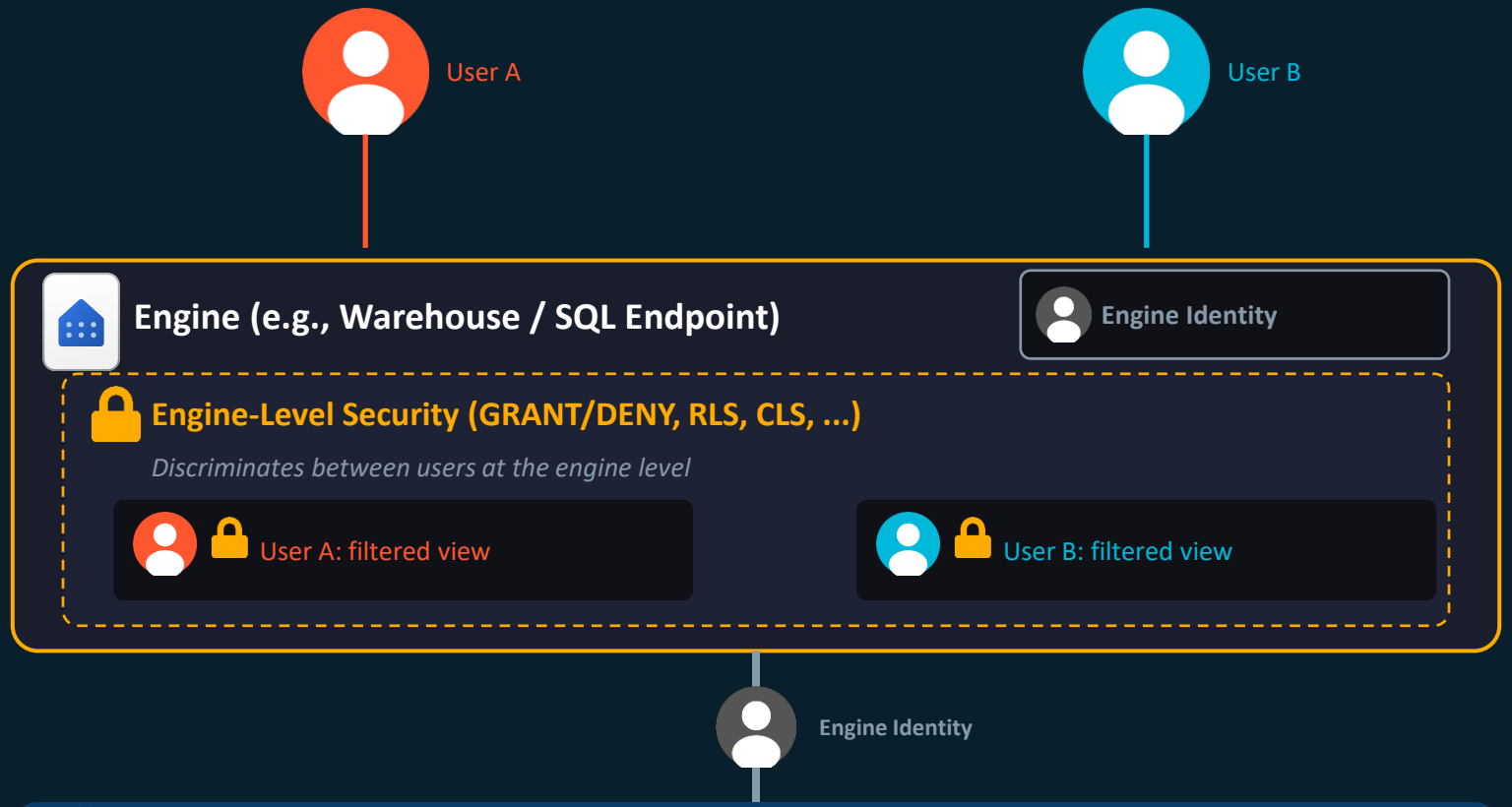
Delegated

Control plane access — as the user

Data plane access — as the engine's fixed identity

Engine identity must have permissions on the data

Engine adds its own security layer (e.g., GRANT/DENY) for per-user discrimination



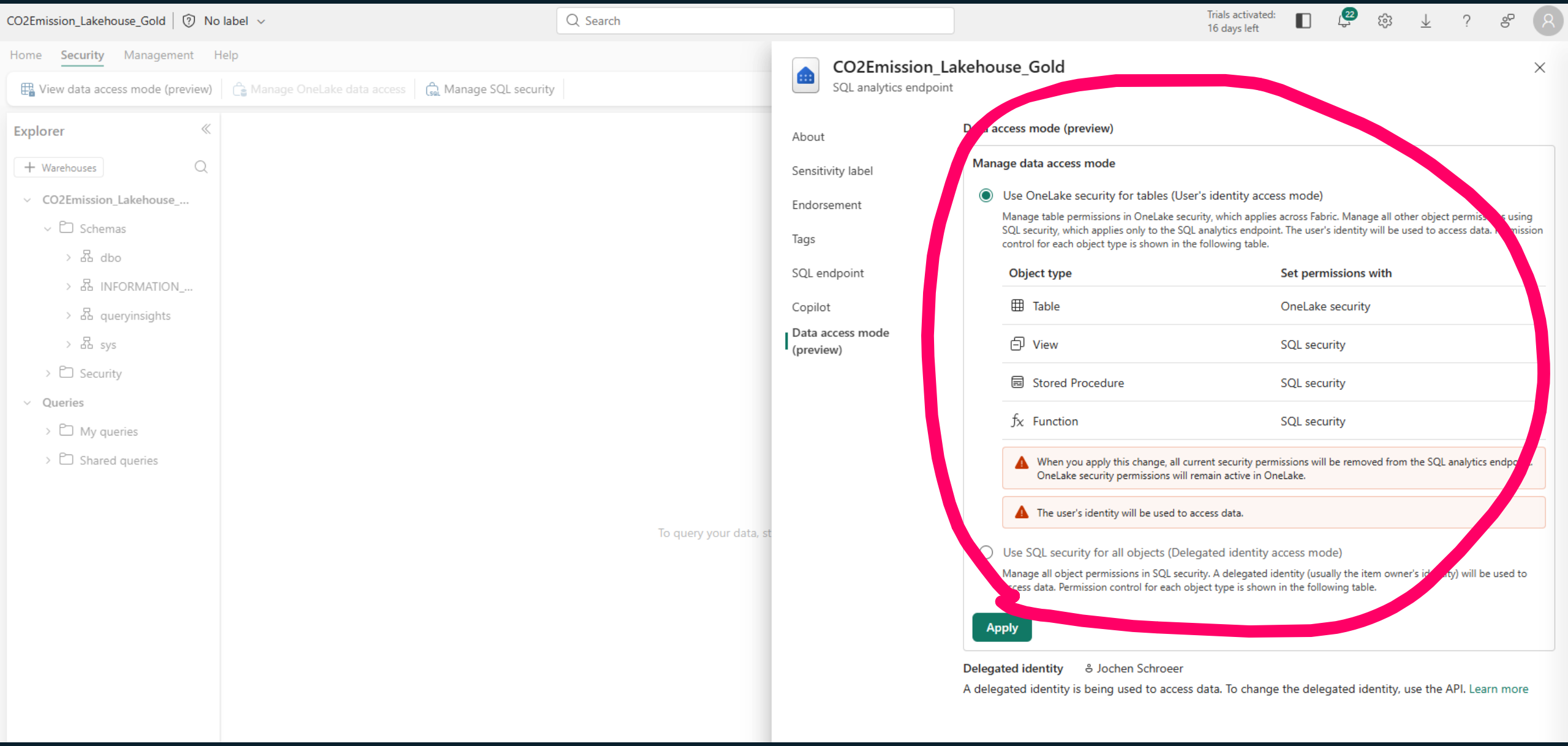
Why Delegated?

Advanced security scenarios

not yet covered by OneLake security

Backward compatibility

with existing engine-level policies


CO2Emission_Lakehouse_Gold
No label
Search
Trials activated:
16 days left
22
Home
Security
Management
Help
View data access mode (preview)
Manage OneLake data access
Manage SQL security
Explorer
Warehouses
CO2Emission_Lakehouse_...
Schemas
dbo
INFORMATION_...
queryinsights
sys
Security
Queries
My queries
Shared queries
To query your data, st
CO2Emission_Lakehouse_Gold
SQL analytics endpoint
About
Sensitivity label
Endorsement
Tags
SQL endpoint
Copilot
Data access mode (preview)
access mode (preview)
Manage data access mode
Use OneLake security for tables (User's identity access mode)
Manage table permissions in OneLake security, which applies across Fabric. Manage all other object permiss using SQL security, which applies only to the SQL analytics endpoint. The user's identity will be used to access data. mission control for each object type is shown in the following table.
Object type
Set permissions with
Table
OneLake security
View
SQL security
Stored Procedure
SQL security
Function
SQL security
When you apply this change, all current security permissions will be removed from the SQL analytics endpo OneLake security permissions will remain active in OneLake.
The user's identity will be used to access data.
Use SQL security for all objects (Delegated identity access mode)
Manage all object permissions in SQL security. A delegated identity (usually the item owner's id ty) will be used to cess data. Permission control for each object type is shown in the following table.
Apply
Delegated identity
Jochen Schroeer
A delegated identity is being used to access data. To change the delegated identity, use the API. Learn more

Fabric Security Model

Shortcut Security — Passthrough

Passthrough

OneLake shortcuts (internal)

User identity passes through the shortcut to the target

OneLake Security evaluates the user at the target

Same identity, same security — transparent

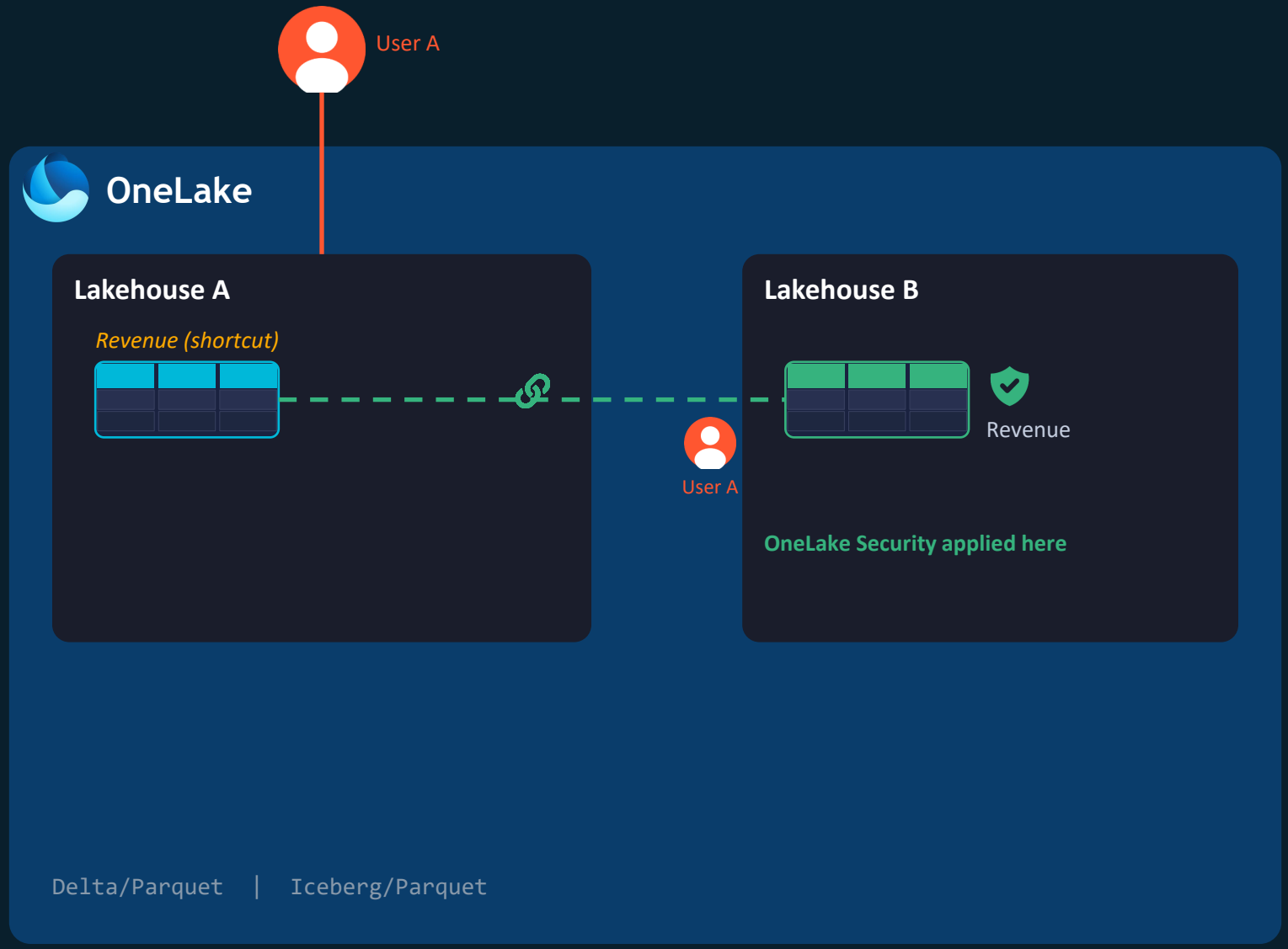
Key Properties

Each user's identity reaches the target

OneLake Security evaluates the user at the target

Applies to: OneLake shortcuts

Transparent — same as direct access



Fabric Security Model

Shortcut Security — Delegated

Delegated

Applies to:

External shortcuts (AWS S3, GCS, ADLS Gen2)

OneLake internal shortcuts as well (future)

A fixed (delegated) identity accesses the target data

OneLake roles on the shortcut side discriminate between users

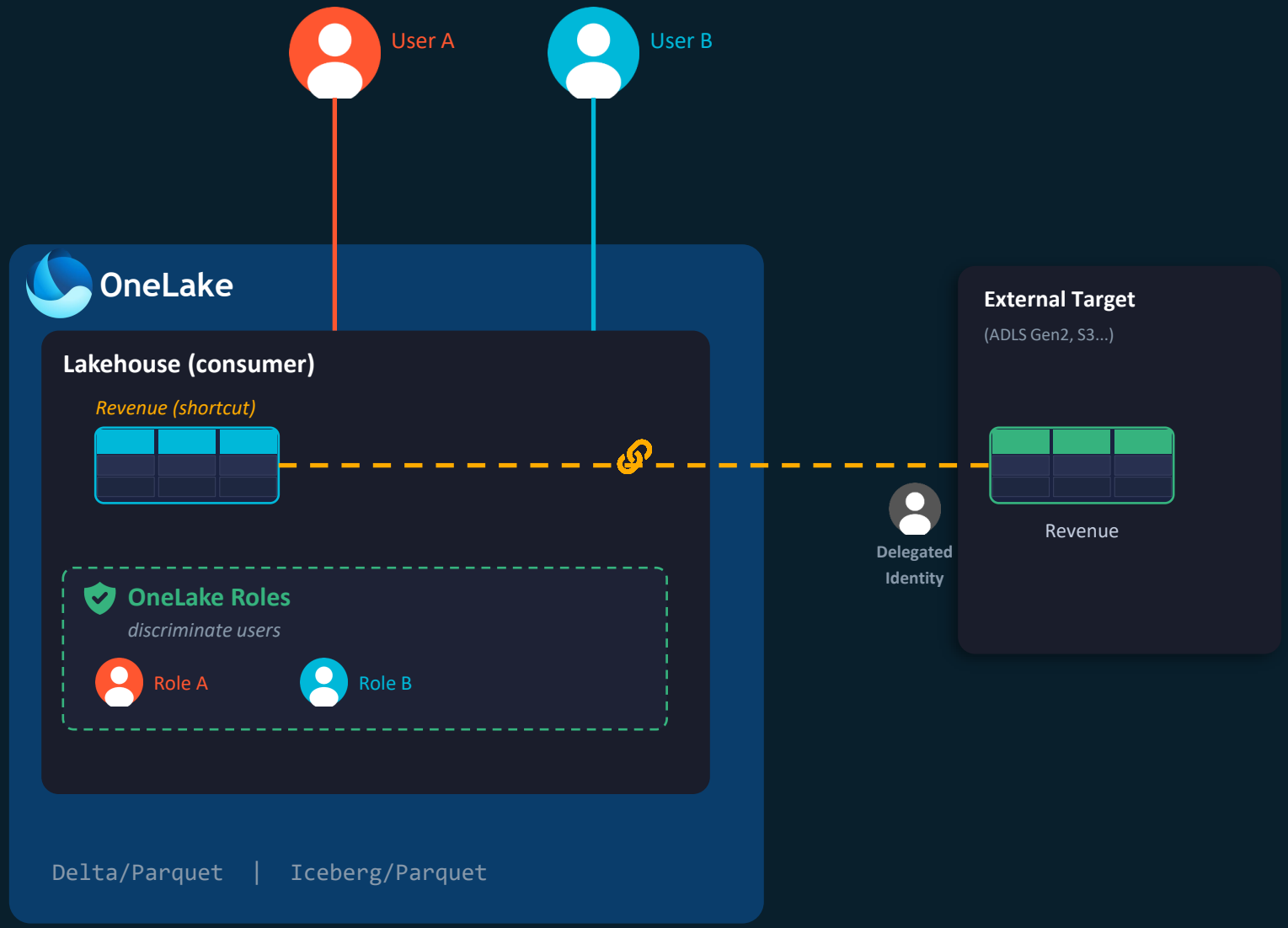
Typical use: dissemination (like import mode in Power BI)

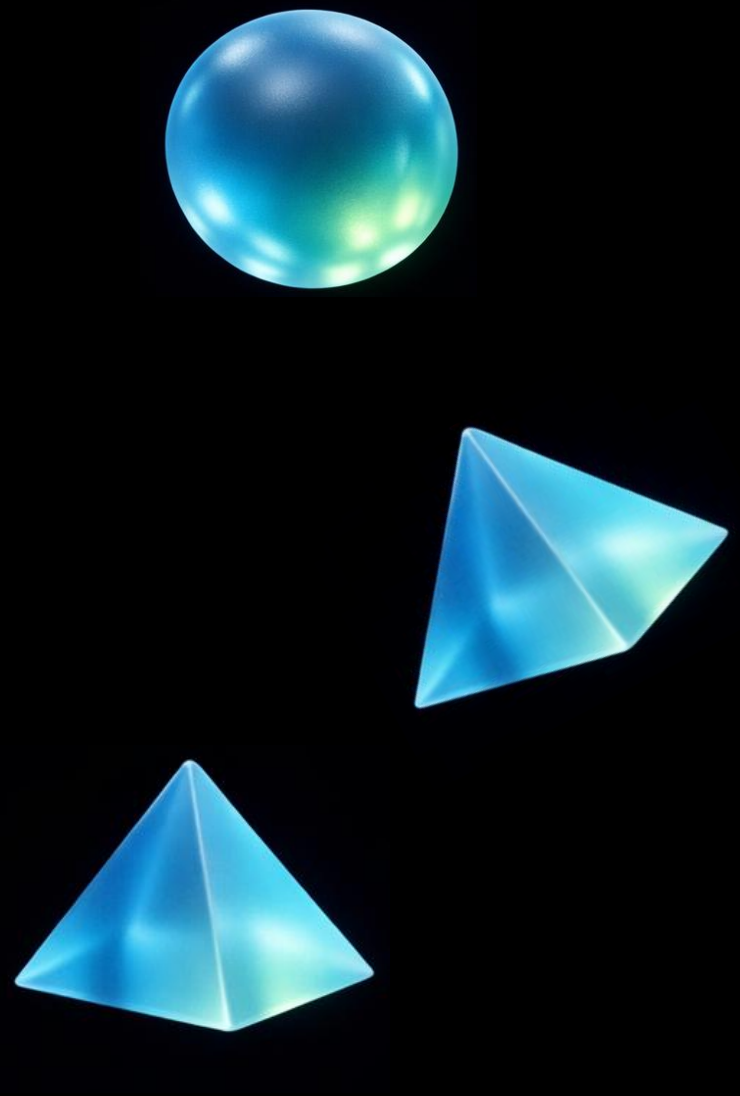
Why Delegated?

External sources can't evaluate Entra identities

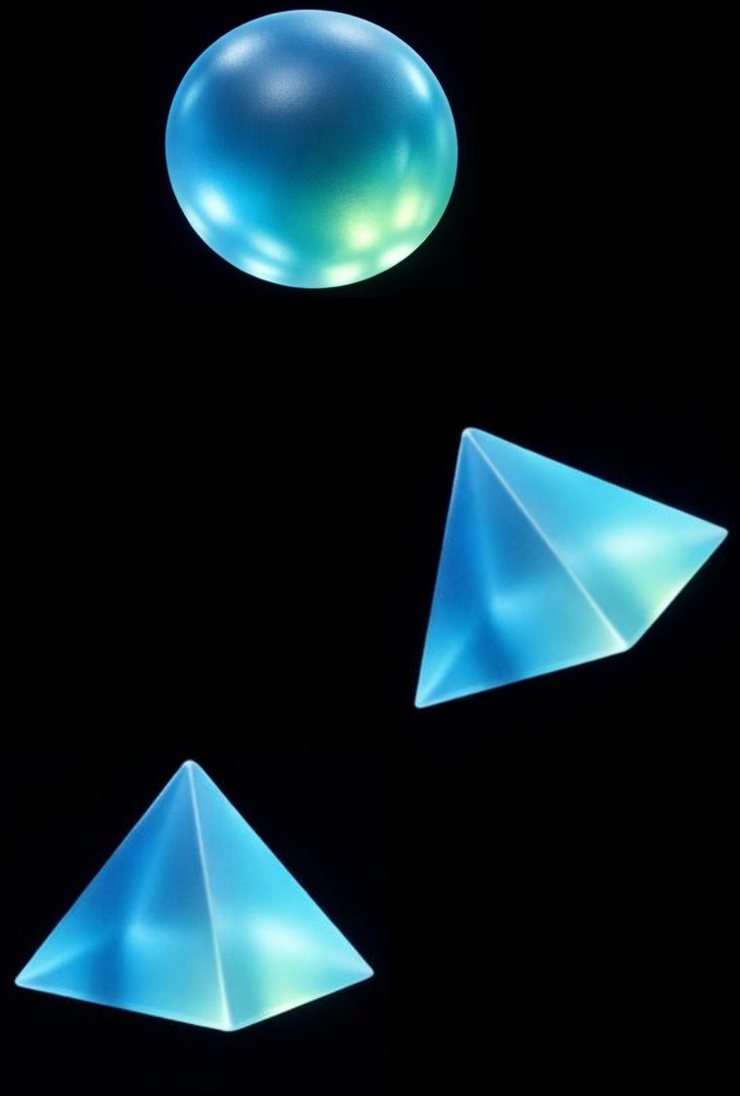
Dissemination pattern — like Power BI import mode

OneLake roles on consumer side control per-user access





End-to-End Demo



OneLake Security Features Recap

Folder Level Security

Grant access to folders in OneLake

Access inherits to all subfolders and files

Listing access on parent folders is maintained

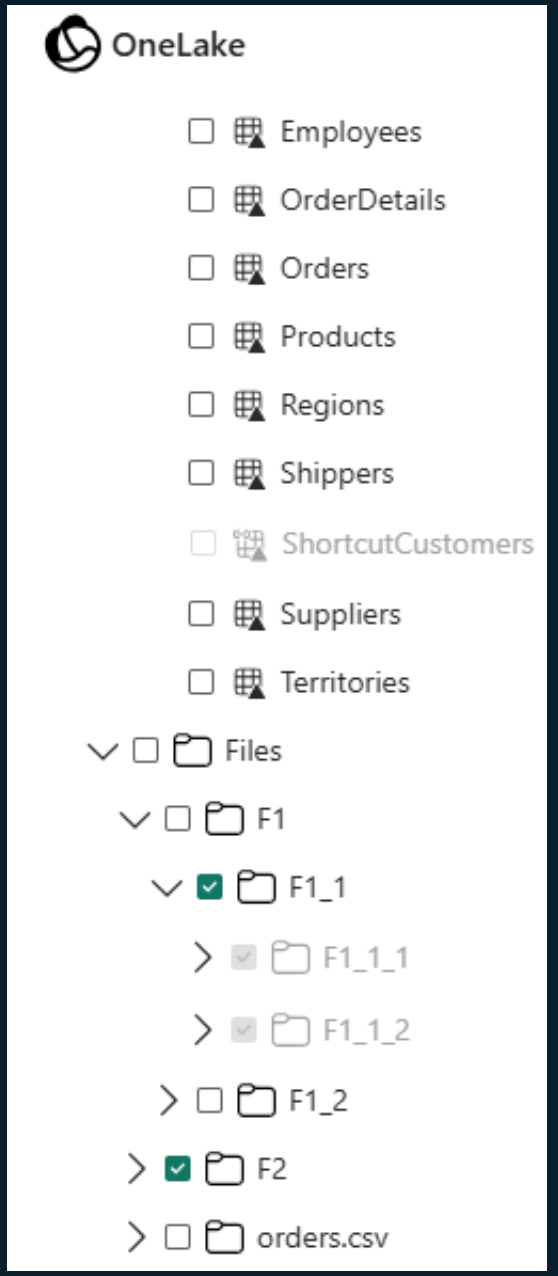
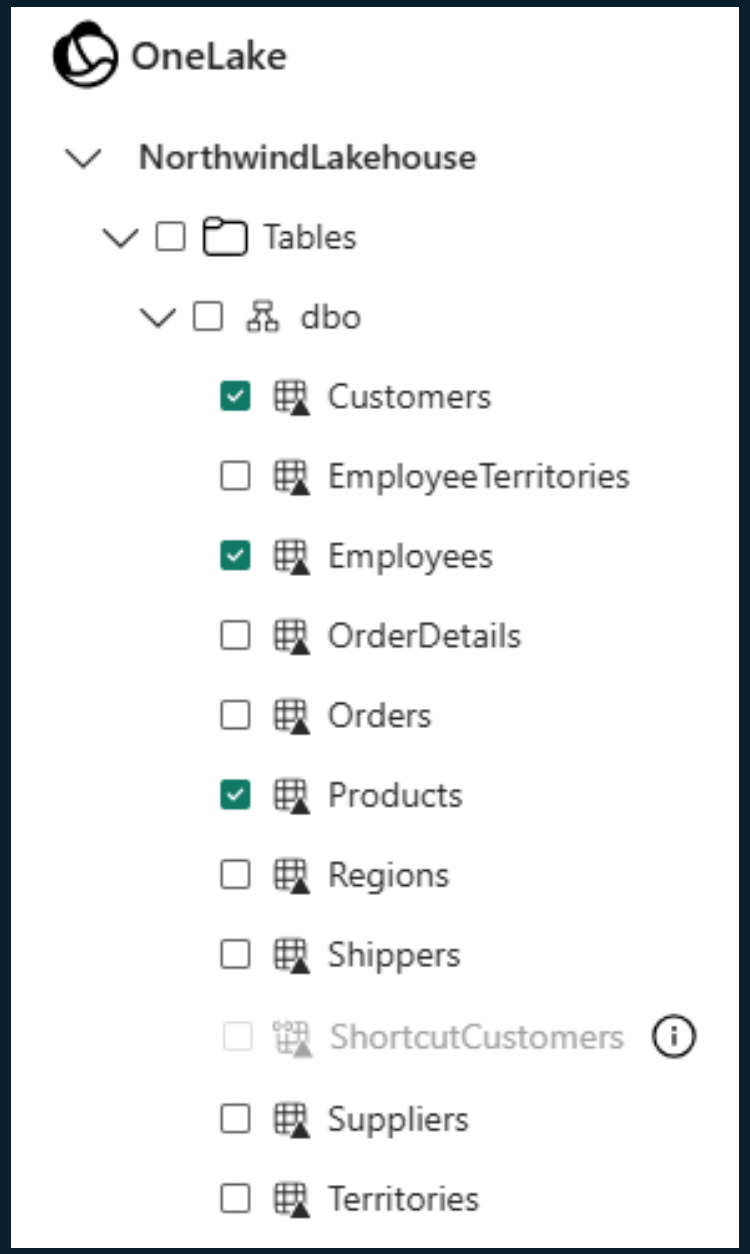


Table Level Security

Grant access to tables in OneLake.

Access inherits from schemas to all sub-tables

Listing access on parent schemas is maintained

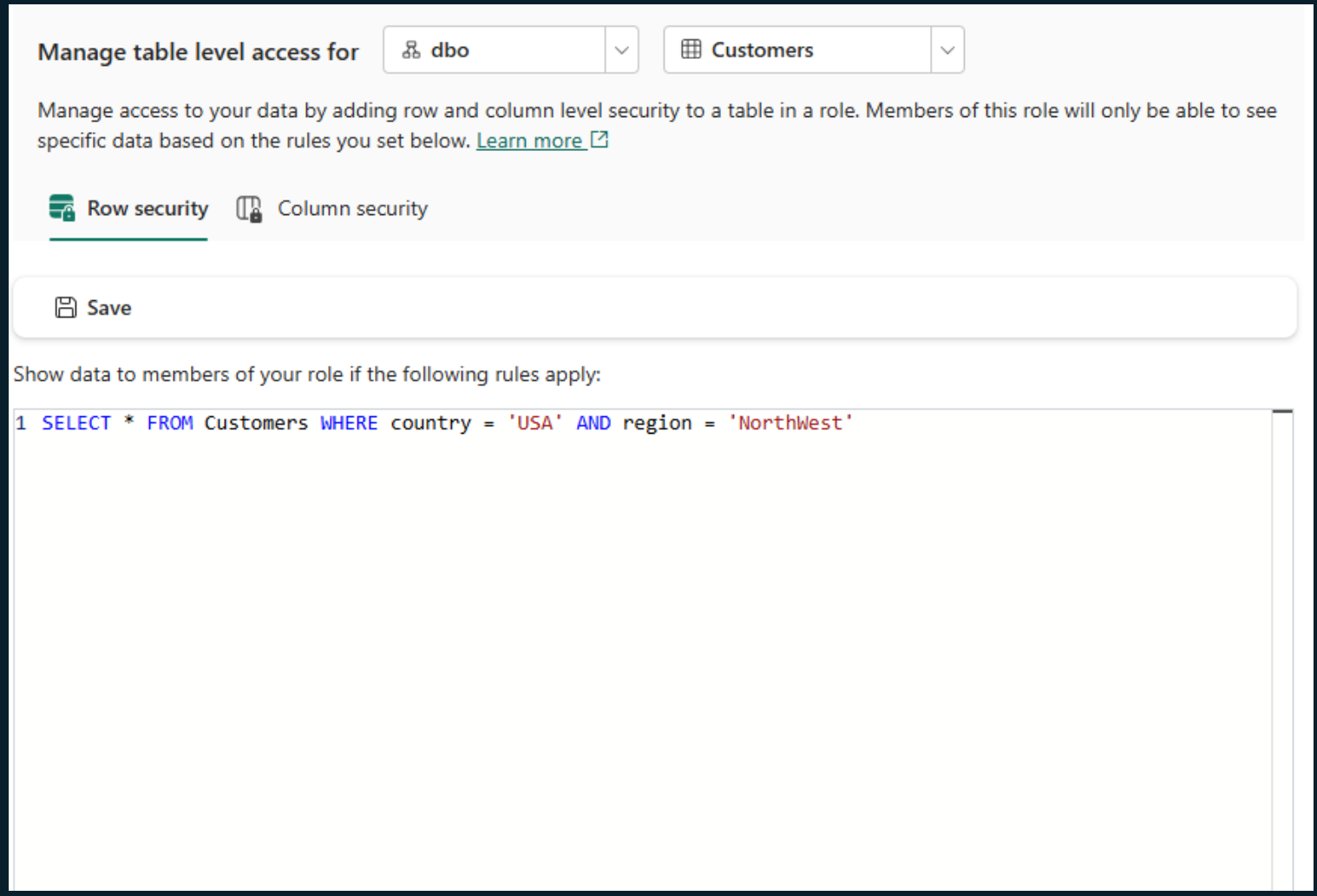


Row Level Security

Restrict access to specific rows for a delta table

Set row conditions by writing SQL

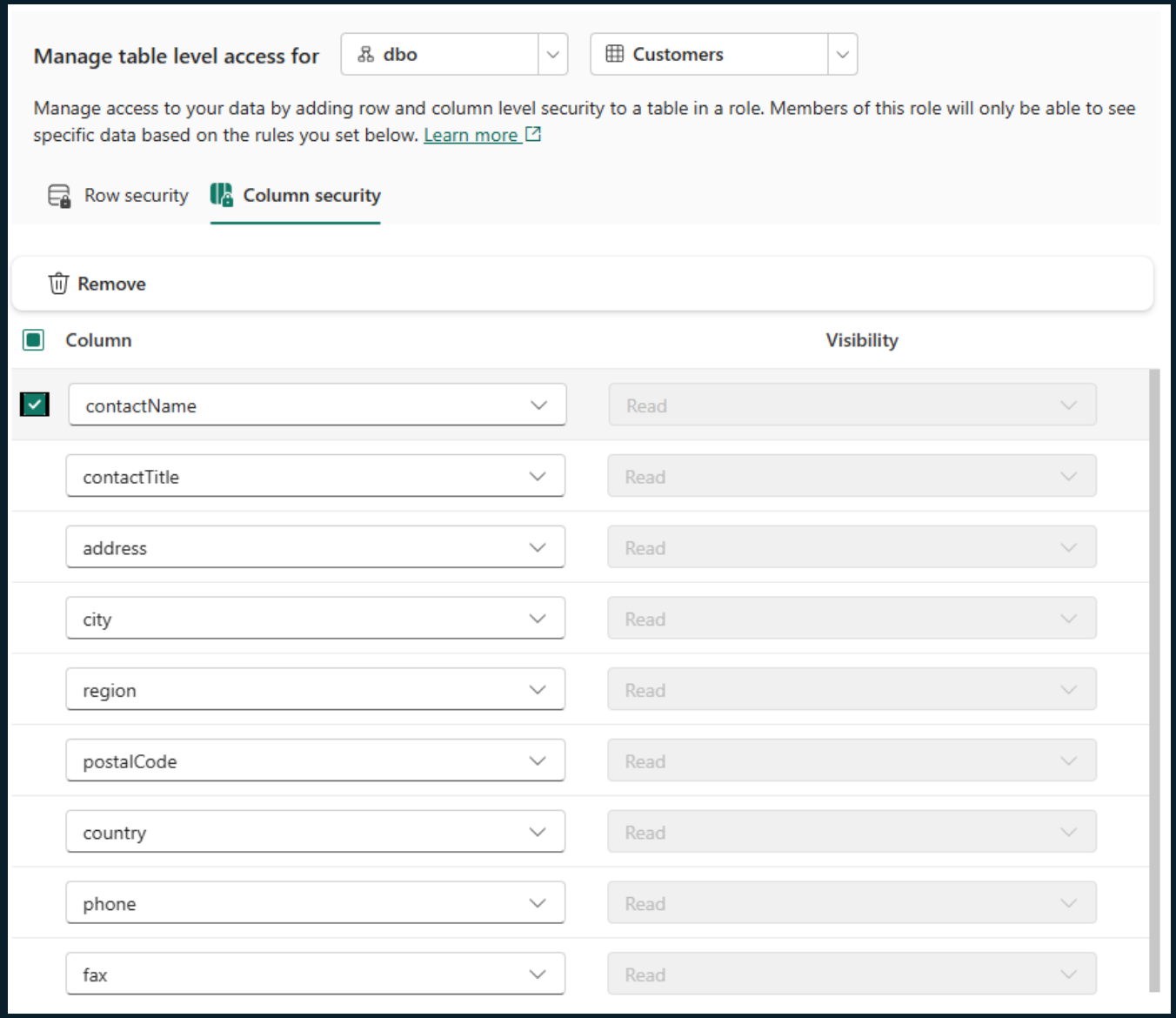
Currently only single-table RLS is supported



Column Level Security

Restrict access to specific columns in a delta-parquet table

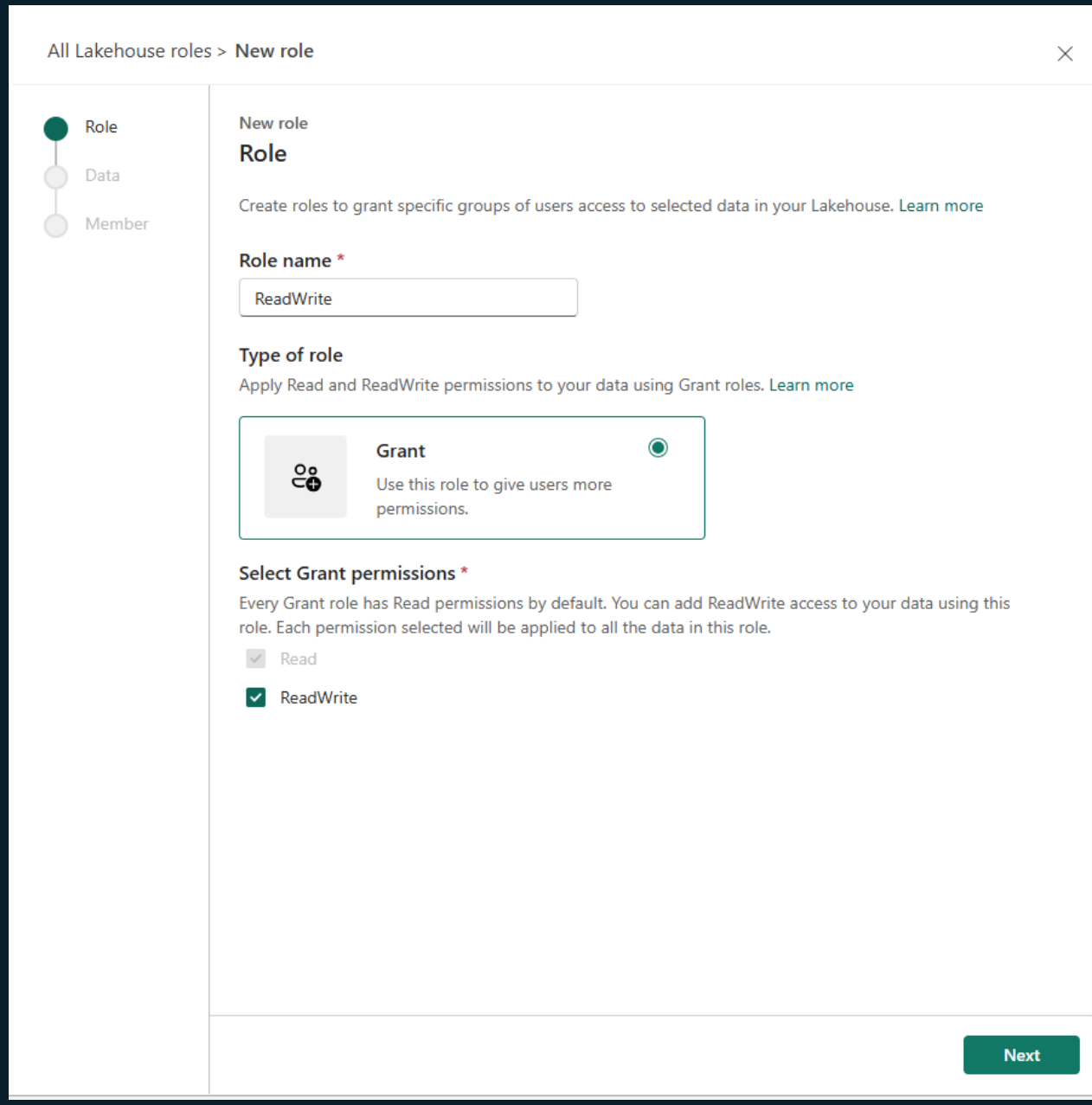
Hides both the column name and data

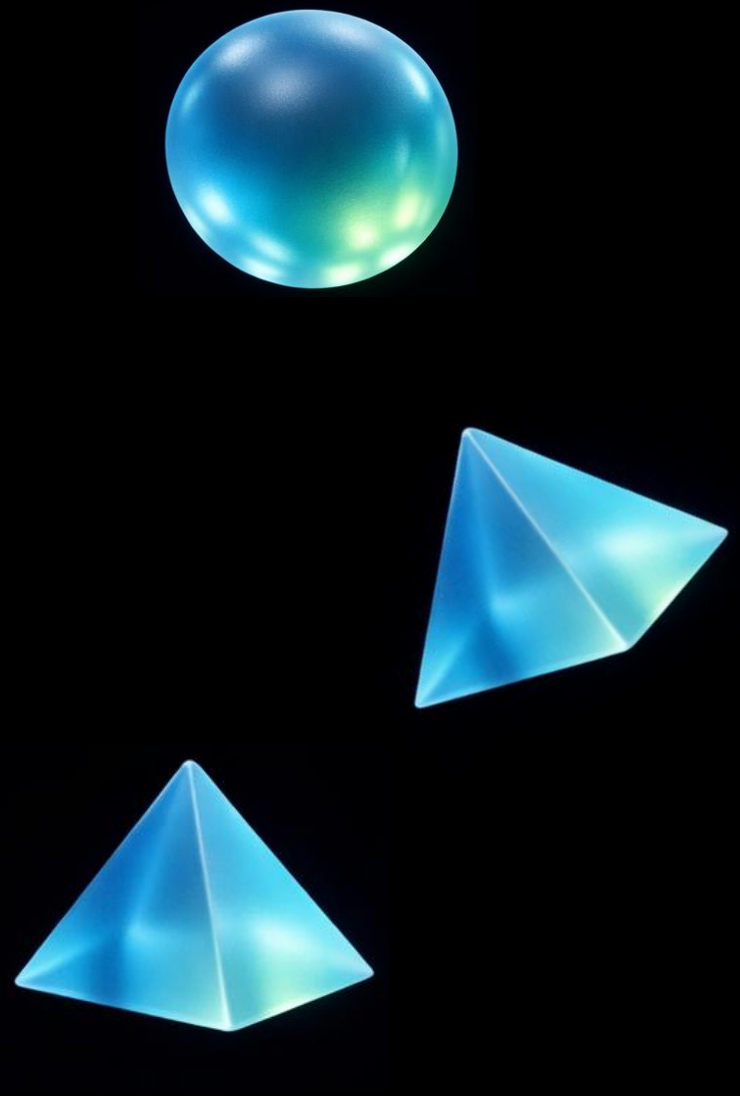


ReadWrite Permissions

Elevate access to Viewers by allowing write operations on specific tables or folders.

Use notebooks, APIs, and now the Lakehouse UX to write data.





What’s next?

OneLake security - now

OneLake security features:

GA

- Object level security
- Column level security
- Row level security
- ReadWrite permissions

• Define OneLake security on:

GA

- Lakehouse
- Azure Mirrored Databrick Catalog
- All mirrored databases

OneLake security filtering:

GA

- Lakehouse
- Spark notebooks
- SQL Analytics Endpoint
- Semantic Models

Preview

- External apps and engines

OneLake security - now

OneLake security features:

GA

- Object level security
- Column level security
- Row level security
- ReadWrite permissions

+ Strong deny
+ Multi-table RLS
+ Dynamic RLS

• Define OneLake security on:

GA

- Lakehouse
- Azure Mirrored Databrick Catalog
- All mirrored databases

+ All Fabric items

OneLake security filtering:

GA

- Lakehouse
- Spark notebooks
- SQL Analytics Endpoint
- Semantic Models

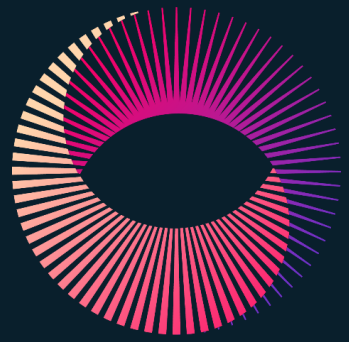
Preview

- External apps and engines

+ Eventhouse
+ Graph DB

Announcing: Omni-directional policy sync

In partnership with Onehouse.ai



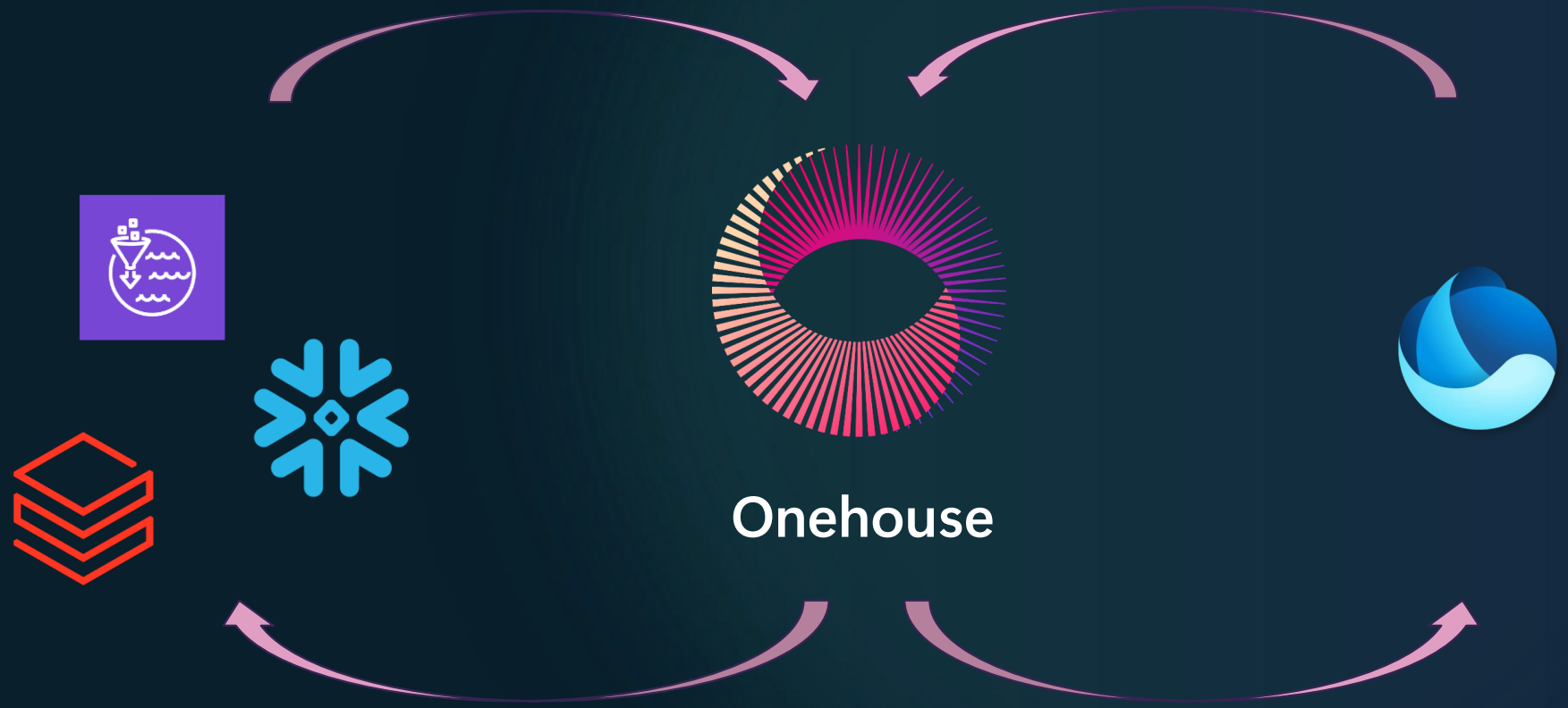
Onehouse



OneLake

Announcing: Omni-directional policy sync

In partnership with Onehouse.ai



OneLake Security Whitepaper



Download now at:

https://aka.ms/OneLake-Security-Whitepaper

We appreciate your feedback

