


# Fabric - Storage account (ADLS gen2)

 [SYSM-362](#) - Jira project doesn't exist or you don't have permission to view it.


 Assess shortcut from Azure (ADLSgen2) to Fabric

- Compare access control mechanisms between Azure and Fabric for ADLS shortcuts
- Evaluate security implications of ADLS Gen2 shortcuts in Fabric
- Latency consideration

| Version | Date        | Description      | Contributor    |
|---------|-------------|------------------|----------------|
| V0.1    | 21 Apr 2026 | Initial document | COLOMBANI Théo |
|         |             |                  |                |

- [SYSM-389](#) - Compare access control mechanisms between Azure and Fabric for ADLS shortcuts
  - [Key message](#)
  - [Description](#)
  - [Azure vs Fabric comparison](#)
- [SYSM-388](#) - Evaluate security implications of ADLS Gen2 shortcuts in Fabric
  - [Shortcut-specific access model](#)
  - [Access flow](#)
  - [Shortcut authentication models](#)
- [Latency consideration](#)
  - [Latency view](#)
  - [Cache Solution for Shortcuts](#)

## SYSM-389 - Compare access control mechanisms between Azure and Fabric for ADLS shortcuts

 [SYSM-389](#) - Jira project doesn't exist or you don't have permission to view it.

### Key message

*Access to ADLS Gen2 data through Fabric shortcuts is governed by **two distinct control planes**: Azure controls access to the storage target, while Fabric controls access to the shortcutted data experience. The design question is not only “who can connect”, but also “which layer authorizes what, with which identity, and at which granularity.”*

### Description

This section compares how access to ADLS Gen2 data is managed in **Azure** versus **Fabric**, focused on three dimensions:

| Dimension      | Azure   | Fabric   |
|----------------|---|--|
| Authentication | Microsoft Entra identity, service principal, managed identity, SAS, Shared Key depending on access mode | Shortcut credential such as Workspace Identity, Service Principal, Organizational account, SAS, or Account Key |

|               |   |  |
|---------------|---|--|
| Authorization | Azure RBAC plus POSIX-style ACLs on folders/files | Workspace roles, item permissions, and OneLake security roles on folders /tables |
| Access scope  | Storage account, container, directory, file       | Workspace, item, shortcut path, folder, table                                    |

- Azure Data Lake Storage uses **RBAC for coarse-grained access** and **ACLs for fine-grained access**.
- Fabric uses workspace and item permissions, and OneLake security adds fine-grained access at folder or table level for supported items.

## Azure vs Fabric comparison

| Topic                    | Azure ADLS Gen2   | Fabric  |
|--------------------------|---|---|
| Primary purpose          | Protect the storage resource itself                         | Protect access to data through Fabric items and experiences                         |
| Identity model           | Entra users, groups, service principals, managed identities | Fabric users plus shortcut credential / workspace identity                          |
| Main authorization model | Azure RBAC + ACL  | Workspace roles + item permissions + OneLake security                               |
| Granularity              | RBAC = broad, ACL = file/folder level                       | Workspace/item = broad, OneLake security = folder/table level                       |
| Default security posture | Depends on RBAC/ACL assignments                             | OneLake security follows deny-by-default once enabled on the item                   |
| Non-Entra access         | SAS and Shared Key supported                                | SAS and Account Key can be used for shortcuts, but reduce identity-based governance |
| Operational owner        | Azure / platform / infra team                               | Fabric / analytics / data platform team   |

- Azure RBAC can grant broad access to a storage account or container, while ACLs secure individual directories and files.
- In Fabric, OneLake security roles can grant access only to specific folders or tables, while Admins, Members, and Contributors generally retain broad access within the item

## SYSM-388 - Evaluate security implications of ADLS Gen2 shortcuts in Fabric



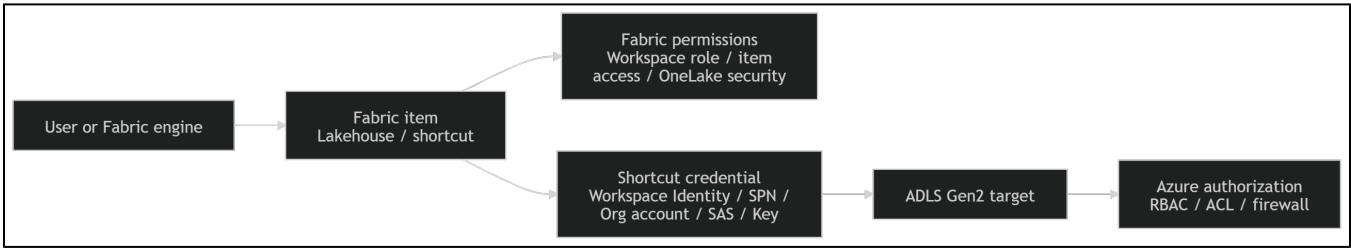
**SYSM-389 - Jira project doesn't exist or you don't have permission to view**

it.

## Shortcut-specific access model

| Question                              | Practical answer   |
|---------------------------------------|--|
| Who authenticates to ADLS?            | The identity configured on the shortcut  |
| Who authorizes storage access?        | Azure ADLS   |
| Who authorizes visibility in Fabric?  | Fabric workspace/item permissions and, when enabled, OneLake security  |
| What happens if both layers apply?    | The effective access is constrained by both layers; for shortcuts, Fabric documents a most-restrictive logic between shortcut path and target path |
| Is behavior identical across engines? | No; some scenarios use delegated identity differently, including owner-based access patterns in specific engines                                   |

## Access flow



| Layer          | What it controls                                     | Examples  |
|----------------|--|---|
| Fabric layer   | Who can see and use the shortcut inside Fabric       | Workspace role, item access, OneLake security                                   |
| Shortcut layer | Which identity is used to reach ADLS                 | Workspace Identity, Service Principal, Organizational account, SAS, Account Key |
| Azure layer    | Whether the target storage path can actually be read | RBAC, ACL, firewall / trusted access  |

## Shortcut authentication models

*Delegated shortcuts access data by using some intermediate credential, such as another user or an account key.*

*These shortcuts allow for permission management to be separated or 'delegated' to another team or downstream user to manage.*

*Delegated shortcuts always break the flow of security from one system to another.*

[blocked URL](#)

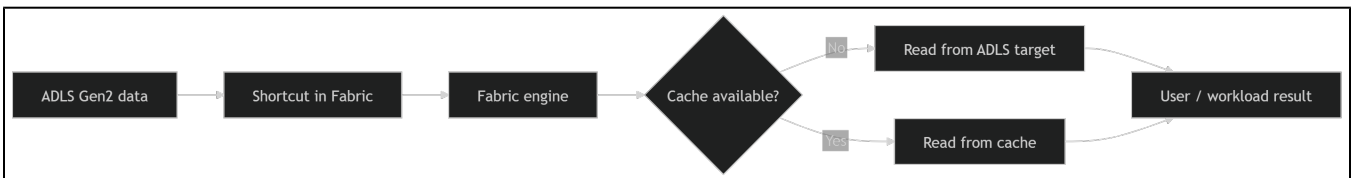
## Latency consideration

### Latency view

| Dimension             | What it means                                     | Shortcut impact  |
|-----------------------|---|--|
| Exposure latency      | Time to make data available in Fabric             | Low, because no ingestion copy is required. ( <a href="#">Microsoft Learn</a> )  |
| First-read latency    | Time for the first query/read to access ADLS data | Can be higher than fully ingested local data because Fabric still reads the external target. ( <a href="#">Microsoft Learn</a> )                         |
| Repeated-read latency | Time for subsequent reads of the same data        | Often improved when cache is used. ( <a href="#">Microsoft Learn</a> )   |
| Refresh latency       | Delay before changes in ADLS are reflected        | Depends on engine and cache refresh behavior; Spark intelligent cache automatically detects underlying file changes. ( <a href="#">Microsoft Learn</a> ) |

## Cache Solution for Shortcuts

### Mecanism



## Settings



Shortcut caching currently supports Google Cloud Storage (GCS), S3, S3 compatible, and on-premises data gateway shortcuts.

[blocked URL](#)