BCBSRI-logoA blue circle with black text

Description automatically generated

DataHub Architecture Decision Record -1

Version # : 0.1

Released on : 11/02/2023.

Author : Rajesh Chinthada

Blue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode IslandBlue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode IslandBlue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode IslandBlue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

Blue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

Blue Cross Blue Shield of Rhode Island

500 Exchange St

Providence, RI 02903

BCBSRI Confidential

**Document Information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | **DataHub DataStage to Matillion Migration** | **Version No** | **0.1** |
| **Date** |  |
| **Project Owner:** |  | **Telephone No:** |  |
| **Executive Sponsor** |  | **Telephone No:** |  |
| **Program Manager** |  | **Telephone No:** |  |
| **Target Implementation date** |  |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Revision Date** | **Summary of Changes** | **Changes marked** |
| **0.1** | **02-Oct-2023** | **First Draft** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## **Glossary**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **ADR** | An architecture decision record (ADR) is a document that captures an important architectural decision made along with its context and consequences. |
| Azure Data Lake Storage Gen2 | *Azure Data Lake Storage Gen2* refers to the current implementation of Azure's Data Lake Storage solution, built on [Azure Blob Storage](https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction). Data Lake Storage Gen2 converges the capabilities of [Azure Data Lake Storage Gen1](https://learn.microsoft.com/en-us/azure/data-lake-store/) with Azure Blob Storage.  *Note: Azure Data Lake Storage Gen1* will be retired on February 29, 2024. |
|  |  |
|  |  |

# **ADR-1-Azure Data Lake Storage Gen2 Container structure**

**Context**

We are designing DataHub Azure Data Lake storage structure. We need to decide how to structure our Datahub Azure data lake storage to optimize for cost, performance, security and usability.

**Decision**

We decided to go with **Option 1 as a pattern** to define Azure Data Lake storage structure in DataHub in each storage account. Gen 2 folders are based on Azure Data Lake Storage Gen2, which is a scalable and cost-effective cloud storage solution that supports hierarchical namespaces.

Landing/Inbound folders are used to store the raw data from the sources/vendors, publish/outbound folders are used to store the processed data for the vendors/destinations, and archive folders are used to store the historical data for backup and auditing purposes.

Also decided to leverage one storage account per environment for supply and demand workflows. Rationale being storage account has no storage limit, massively scalable and we don’t anticipate any storage account scalability limitations applicable for our batch operations([storage-account-scalability](https://learn.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account)).

**Option 1:**

**Supply:**

<<storage-account>>/<<datahub>>/<<supply >>/**landing**/<<vendor-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<supply >>/**publish**/<<vendor-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<supply >>/**archive**/<<vendor-name>>/<<feed-name>>/<<frequency>>

**Demand:**

<<storage-account>>/<<datahub>>/<<demand >>/**landing**/<<application-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<demand >>/**publish**/<<application-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<demand >>/**archive**/<<application-name>>/<<feed-name>>/<<frequency>>

**Option 2:**

**Supply:**

<<storage-account>>/<<datahub>>/<<supply >>/**inbound**/<<vendor-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<supply >>/**outbound**/<<vendor-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<supply >>/**archive**/<<vendor-name>>/<<feed-name>>/<<frequency>>

**Demand:**

<<storage-account>>/<<datahub>>/<<demand >>/**landing**/<<application-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<demand >>/**publish**/<<application-name>>/<<feed-name>>/<<frequency>>

<<storage-account>>/<<datahub>>/<<demand >>/**archive**/<<application-name>>/<<feed-name>>/<<frequency>>

Example: [Blob-Structure-Options.xlsx](https://bcbsri.sharepoint.com/sites/CloudNativeDataProject-365copy-365/Shared%20Documents/Track%201%20-%20Datastage%20Netezza%20Migration/Workstreams/Architecture/Architecture-Decision-Records/Blob-Structure-Options.xlsx?d=w38e3d8392c904c0b86f3ae610c656a2d&csf=1&web=1&e=Bb2puD)

**Consequences**

By using above zones to separate data by its stage in the workflow/pipeline, we expect to achieve following benefits.

Improved performance: We can optimize file formats, compression, partitioning for each zone.

Enhance security: We can apply granular permissions and access control policies for each zone according to its sensitivity and governance level.

Increase usability: We can provide a clear and consistent view of our data from landing and publish stages perspectives. Metadata and cataloging will be much efficient.

- Enhanced data traceability: We can use the hierarchical namespaces and metadata of Azure Data Lake Storage Gen2 to organize our data in a logical and consistent way. We can also use the archive folders to keep track of the data lineage and provenance.

- Optimized performance and cost: We can leverage the performance advantages of Azure Data Lake Storage Gen2, such as parallelism, concurrency, and caching, to speed up our data processing. We can also reduce our storage costs by using tiered storage policies and lifecycle management features.

**Next Steps**

Based on above initial pattern approval, team will create supply and demand structure for all applications. Any additional changes will be brought back to this forum and this decision will be updated.

**References**

[Azure Data Lake Storage Gen2 Introduction](https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)

<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/cloud-scale-analytics/best-practices/data-lake-zones>

<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming>

<https://learn.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account>

[Compare Azure Data Lake Store to Azure Blob storage - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/introduction-to-azure-data-lake-storage/4-azure-data-lake-and-blob-storage)

919