# Multiple S3 Archive – Software Design Document

At its current state HPC-DME allows configuring a single archive per DOC. That archive can be either POSIX filesystem or S3. The system is designed to support multiple S3 providers (Cleversafe, Cloudian, AWS, etc), but at its current state a single provider can be used.

A new requirement is given to support multiple S3 providers concurrently. Each DOC will be configured with a default S3 provider to upload data, but data that was uploaded in the past to other S3 providers will be available for download.

The actual S3 archive and provider details shall remain abstracted from the users as it is in current state.

There will be no API changes associated with this change, so no impact on UI or CLI.

## Detailed design

The following changes/additions to the code base are needed to implement multiple S3 Archive support:

#### HpcMetadata.xsd

* HpcSystemGeneratedMetadata – add a new system metadata ‘s3\_archive\_configuration\_id’ of type string

#### HpcMetadataServiceImpl.java

* Modify addSystemGeneratedMetadata() to add a new parameter to record the new ‘s3\_archive\_configuration\_id’ system metadata
* Modify getDataObjectSystemGeneratedMetadata() to set the ‘s3\_archive\_configuration\_id’.

#### HpcUserTypes.xsd / HpcUser.xsd

* HpcIntegratedSystem - Add value for each S3 provider added to this enum, e.g CLOUDIAN, AWS
* HpcDataTransferAuthenticatedToken – Add s3ArchiveConfigurationId to this complex type

#### HpcDataTransferServiceImpl.java

* Modify getAuthenticatedToken(HpcDataTransferType dataTransferType, String configurationId) to accept s3ArchiveConfigurationId parameter as optional. This additional parameter will be used only if the system account to be looked up for is a ‘S3 data transfer’ – so we will need to get the Cleversafe vs. Cloudian system account.
* Change the logic to search for existing token based on integrated system (i.e. get a S3 system account to access Cleversafe vs. Cloudian etc)
* If token not found - Provide the integrated system to the system locator to return the appropriate system account (Cleversafe or Cloudian or AWS etc)
* Change all places in this class that call getAuthenticatedToken and provide s3ArchiveConfigurationId if it’s S3 – Note: there are 15 places

#### HpcSystemAccountLocator.java

* Modify getSystemAccount(HpcDataTransferType dataTransferType, String hpcDataMgmtConfigId) to accept s3ArchiveConfigurationId for S3 accounts.

#### HpcDataManagementConfigurationLocator.java

* Update getDataTransferConfiguration(String configurationId, HpcDataTransferType dataTransferType) to accept s3ArchiveConfigurationId as an optional parameter to be used in case we look for S3 config.
* Change all places in the code to send the ‘s3ArchiveConfigurationId’ – for upload, used the value configured in the HPC\_DATA\_MANAGEMENT\_CONFIGUATION table for the DOC, and for download use the value received from the system generated metadata. For files uploaded before this change, there will be no s3 config id metadata, so use the default value configured in HPC\_DATA\_MANAGEMENT\_CONFIGUATION

#### HpcDataManagementConfiguration.xsd

* Add a 2 new attributes to capture default upload and download s3 archive configuration IDs.

#### HpcDataManagementConfigurationDAOImpl.java

* Current HPC\_DATA\_MANAGEMENT\_CONFIGUATION table keep a single set of S3 config per DOC. Create a new table HPC\_S3\_ARCHIVE\_CONFIGURATION which will hold a list of S3 configurations for each DOC and provider
* Change the DAO to load the configurations based on the schema change – Note: this is a fairly big change
* Add a 2 new columns to HPC\_DATA\_MANAGEMENT\_CONFIGUATION which will capture the default upload and download s3-archive-configuration-id for each config