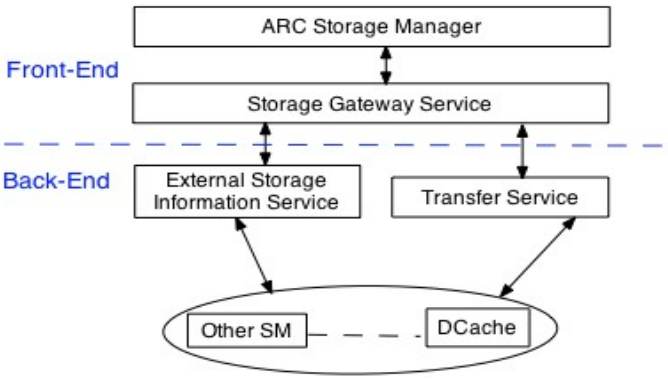
Architecture for ARC1 Storage Gateway Component

The proposed architecture for accessing data from third party storage system using ARC storage component is based on following sections.

Front-End: Higher level of standard interfaces to access the third party store provided through “Storage Gateway Service”.

Back-End: Based on “External Storage Information Service” and “Transfer Service” to access the underlying storage systems.



Description of the Services

Storage Gateway Service

This service provides a transparent view of external storage system to the ARC Storage Manager. A higher level service that is based on External Storage Information Service and Transfer Service, provide a reliable and consistent access to the underlying storage system.

External Storage Information Service

The goal of this service is to check the status of the external storage system in some regular intervals and get the information required by the Storage Gateway Service by creating minimal communication overhead.

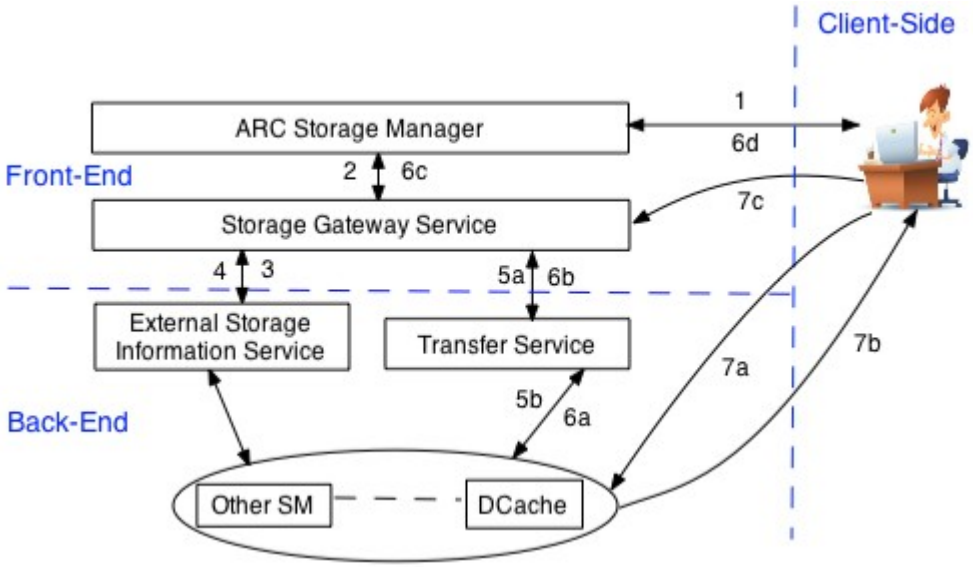
Comment: May be in the initial phase this service contains static information about each underlying system. Like what protocols are available etc, which makes this service less important in the system but I think, as the work progresses this service becomes an important building block of the proposed architecture.

Transfer Service

Core service responsible for establishing a connection with the underlying storage system using its native API to initiate the process of transferring data. This service provides the actual transparency towards different storage systems by implementing there interfaces.

The concept behind the proposed architecture is to provide a reliable and consistent environment for the ARC users to accessing the third party storage system without actually participating in the transfer of data. The proposed hierarchical architecture also allows the ARC Storage Manager to have a standard interfaces for all the different storage systems.

Download Scenario from Third Party Store



1 – User requests for downloading a file. Path of the file is

/ExternalStore/DCache/store-1/FileName

1: ARC SM related — 2:Third party store

2 – By using the path of the requested file, Storage Manager determine that the request is for external store and pass the request to Storage Gateway Service .

3 – Storage Gateway Service first determine which external store can fulfill this request and then generates a request to the External Storage Information Service to get the information about that storage system.

4 – External Storage Information Service time to time connects the external storage systems to check the status of the system and also gather the information required by the Storage Gateway Service.

5a – Once the Storage Gateway Service gets the required information, according to the defined priorities in the Storage Gateway Service, the download request will be transfer in the proper formate of the external system to the Transfer Service.

5b – Transfer Service connects the underlying system using its native API or command line interface and delivers that request. For example, In case of DCache as external store and GridFTP as protocol used, DCache will generate a transfer URL (TURL).

6a – 6b – 6c – 6d – ARC Storage Manger transfers the TURL to the client.

7a – 7b – 7c – Using the TURL, client directly get the file from the DCache pool and in the end of the transfer, the client tool will notify the Storage Gateway Service about the status of the transfer. In case of failure the process can be restarted at the level of the Storage Gateway Service.