What is an SRM

SRM's provide an API to manage storage resources on the Grid. They provide a consistent service orientated API which is common to Mass Storage Systems and single platter disk file servers.

Needed abilities/capabilities:

Managing TOTAL Storage capacity

*UseCase 0 Interoperability

Clients should be able to inter-operate with all SRM's supporting the same version of the SRM. This top level requirement leads to some other requirements.

*UseCase 0.1 Interoperability Dependancy Latency

Storage latency is quite variable. Off line storage and near line storage requests have considerably higher latency than on line storage. The SRM API must be asynchronous for all operation that will be affected by data retrieval/storage latency to preserve interoperability.

*UseCase 0.1 Interoperability Dependancy Naming of files

The API must be able to work with a variety of different back end storage systems. Some of these systems may support a directory structure behind the API others may only provide and list of request ID's. Both these naming conventions should be supported, for this reason file addressing should be of type string.

*UseCase 0.1 Interoperability Dependancy SOA

Storage elements must be able to fully define their API specification using a interface specification language that can be specified to check for compliance, examples include Corba's IDL and WSDL. The SRM API interface must at the least be a support of the interface specified in the interface description language.

UseCase1.1 Putting Data In Storage.

A client wants to be able to place a certain amount of data/number of files into the storage. It needs a guarantee that this operation can be successfully carried out within a given timeframe, in which the storage is not allowed to run out of resources for this purpose. The storage should be able negotiate the duration of the guarantee with the client. The storage should be able to extend the guarantee duration in negotiation with the client.

UseCase1.2 Retrive Data From Storage

A client wants to be able to retrive a certain amount of data/number of files into the storage. It needs a guarantee that this operation can be successfully carried out within a given timeframe, in which the storage cache is not allowed to run out of resources for this purpose. The storage should be able negotiate the duration of the guarantee with the client. The storage should be able to extend the guarantee duration in negotiation with the client.

UseCase1.3 Remove Data From Storage

The client needs to be able notify the storage that certain data/files or a given reserved storage capacity is not needed anymore and that it can be given back to the storage manager.

UseCase1.4 Temporary Data Storage

What is an SRM
The client wants to store some data only temporarily in the storage. It will want to have a guarantee that during a negotiable time, the data will be kept on storage but after the time has expired, the client expects the storage to be freed and the data removed.

**UseCase1.5 Checking Data Storage**

The client wants to check some data or files are stored with an external reference known to the client without having to get the data.

**Managing ONLINE Storage**

**UseCase2.1 Data Movement**

In order to accommodate a large number of data movement requests, the storage needs to manage which data are available online at a given moment so that they can be transferred.

**UseCase2.2 Low-Latency Access**

In order to accommodate applications that have a need for 'fast' high-performance access, the storage needs to be able to keep requested data online for a certain amount of time. The client needs to have the guarantee that during this time the data is accessible with very low latency. The client should be able to negotiate the duration of this guarantee with the storage, and should be able to notify the storage also if the data is not needed online any longer.

**Additional Use Cases**

**UseCase3.1 Hierarchical Namespace**

A client would like to address the files stored in the SRM by a hierarchical file namespace. For this purpose the Storage URLs that identify the files for the client need to have some additional semantics (i.e., it's not just a string identifier). The client would like to manage the hierarchical namespace through the SRM.

**UseCase3.2 Access Control**

A client would like to secure the data (with or without the hierarchical namespace). Only specific users should be able to read, write, and delete the data from the SRM.

**UseCase3.3 First-party Transfer**

In order to avoid networking and firewalling problems, the SRM should be able to initiate a transfer as a primary participant (as opposed to being controlled as a 3rd party transfer client from the exterior). In order to achieve maximal performance, the SRM would need to expose the parameters of supported transfer protocols.

*UseCase3.4 Bulk operations

The Client should be able to minimise the cost of authentication per operation by bulking multiple operations together.

*UseCase3.4.1 Bulk operations dependancy

The Client should be able to establish the status of each operation within an bulk operation.
.. and there are more use cases concerning

- quotas
- metadata
- monitoring
- 'other spaces'

-- PeterKunszt - 28 Sep 2005 -- OwenSynge - 12 Oct 2005

- that will do for now