Proposal for GLUE 1.3 for Storage

S. Andreozzi, INFN
P. Badino, CERN
J.-P. Baud, CERN
S. Burke, RAL
E. Corso, ICTP
F. Donno, CERN
A. Frohner, CERN
A. Hanushevsky, SLAC
T. Hesselroth, FNAL
J. Jensen (editor), RAL
M. Litmaath, CERN
L. Magnoni, INFN
T. Perelmutov, FNAL
A. Shoshani, LBNL
A. Sim, LBNL
O. Synge, DESY
S. de Witt, RAL
R. Zappi, INFN

2006-Nov-15

1 Introduction

This document is the result of the SRM working group meeting at CERN 29 Aug to 1 Sep 2006, updated with feedback from the GLUE meeting at Imperial College 30-31 Oct 2006. It is a GLUE schema proposal for SEs, updated from version 1.2 to support SRM versions 2.2 and 3.0.

Note on the authors: the list of authors include SRM implementers, SRM protocol designers, other middleware people, and some GLUE group members who have contributed ideas and feedback to this proposal (including L Field who also contributed but asked that his name be removed from the list). The GLUE members should not be seen as endorsing this proposal—only the final 1.3 version should be seen as endorsed by the GLUE group.

This document was typeset with LaTeX.

2 Document status

This document is the result of combining the input from the SRM working group and that of the LCG group, with the feedback from the GLUE schema group.

3 Definitions

- *Accounting* is defined as processing metrics from an SE. The only metrics considered in this document are space used and space available. Processing includes collecting, storing, and visualizing. Note the SE is only responsible for publishing instantaneous values of metrics, not historical data.
- (More definitions here)

4 Use Cases

Brief description of use cases. Assume we are given a VO name, a location of an information service, and minimally, a list of acceptable transfer and control protocols.

- Service Discovery 1: Client (person or software) on User Interface (UI) or Worker Node (WN) wishes to locate an SE to store a file.
- Service Discovery 2: FTS wishes to transfer a file from one SE to another and needs to discover the appropriate paths and endpoints (web services endpoints and SURL paths) for the transfer.
- Service Discovery 3: A Resource Broker wishes to send a job to a site and needs to check if there is an appropriate SE at or near the site.
- Accounting 1: The LCG GDB wishes to get an overview of the resources available to, and used by, a specific VO in a given SE.

5 Summary of proposed changes for 1.3, compared to 1.2

Overview of changes:

- The SA needs improvement to enable more than one VO to use the same SA without duplication; also, the SA may itself cover several StorageComponent classes.
- Many entities now have a Capability property, allowing the SE to publish implementation specific, or common but optional, key/value pairs. Some proposed values are described in this document.

Detailed summary of changes for 1.3, by class.

- StorageElement:

  - Added an ImplementationName property (enum) and a ImplementationVersion property (string). The ImplementationVersion will be version-specific (i.e. the format is not intended to make sense between versions; also, it should never be used for service discovery).
  - Clarified sizes are published in GB (not GiB).
  - Deprecated SizeTotal and SizeFree.
  - Added TotalOnlineSize and TotalNearlineSize.
  - Added Status enum.

- StorageArea:
  - Updated with VO-specific spacetokens ("default" for SEs that don't support spaces) and paths.
  - Sizes are now all GB (10^9 bytes), to be consistent.
  - Added Capability for publishing optional or implementation specific information.
  - StorageType (volatile, durable, permanent) is now deprecated because the meaning was overloaded. We now propose RetentionPolicy (`custodial`, `output`, `replica`) and AccessLatency (online, nearline, (offline)) as clearer replacements. The StorageType with the old values should still be published in the Policy class for backward compatibility. There is now also an ExpirationMode with values releaseWhenExpired, warnWhenExpired, neverExpire which also covers some of the previous meaning of the StorageType.

- New VOInfo class, containing the path and the a Tag attribute allowing space tokens (etc) to be published. The VOInfo class is equivalent to the VOView for the CE (roughly).

- Policy: The FileLifetime attribute deprecated; it is now published inside the SA.

- ControlProtocol: unchanged.
  - We note that although Port is deprecated (in 1.2), there exist software that use it.

- AccessProtocol: unchanged except we propose to add one attribute, `MaxStreams`.

6 Summary of postponed changes for 2.0, compared to 1.2/1.3

An incomplete list of postponed changes (between version 0.8 and 0.9 of this document, after the discussion with the GLUE group).

- We proposed a StorageComponent (SC) class which can be used to describe different storage implementations within a single SA: An SA may have more than one physical space, each of which can have different access latencies or retention policies, or types of network access. Access to the SC has also been split off to a separate AccessType class, because each SC can support one or more access protocols (e.g. LAN and WAN).

Detailed summary of postponed changes, by class.

- StorageComponent: new entity (class).
- NetworkAccess: new entity (class).

7 GLUE SE Schema

In the following section, we give an overview of the GLUE storage entities (classes). The reader is advised to refer to the accompanying UML diagram, which, for reasons of space, are not included in this document.

7.1 Entities, or Classes

Entity	Inherits From	Brief Description
StorageElement		The storage element type—top level class for each SE.
StorageArea		An abstraction of physical storage.
ControlProtocol		A description of the control protocol access points
AccessProtocol		A description of the supported protocols for file transfer and access.
AccessControlBase		Access control rule for coarse-grained selection.
State		Summary of used sizes for the SA.
Policy		All attributes optional, Quota and FileLifetime deprecated.
VOInfo		VO (or group) information for an SA (like VOView for CE)

In version 0.8 of this proposal, sizes had moved to a StorageComponent class; but since that was postponed for 2.0, we have undeprecated State.

7.2 Notes

Certain attributes such as Capability are meant to publish implementation specific hints between to the client. Such attributes thus hold key/value pairs, and the client may in turn pass them in via the SRM methods, with unknown pairs being ignored by the implementation (or possibly make the implementation issue a warning).

It has been suggested that such pairs be published as multi-valued strings of the form $k_i = v_i$ where k_i is a key string (which obviously may not contain the character '='), and v_i is the value string. The order of the key/value strings thus does not matter.

Example:

```
Capability: ServerTCPBufferSize=10240
Capability: NetworkType=OPN
Capability: RFC1323Support=WindowScale,RTTM,PAWS
```

An implementation or a Grid can define its (their) own such capabilities, but they should be documented to enhance interoperability and prevent collision (i.e., same key used with different semantics).

Another proposed way to obtain the sizes of SAs, a potentially more detailed view than can be published for the SA, is to call srmGetSpaceMetadata().

7.3 StorageElement

StorageElement

Property	Type	Mult.	Unit	Description
UniqueID	string	1		Global unique identifier for the SE
InformationServiceURL	string	1		URL to the information service for this SE.
ImplementationName	enum	1		The name of the implementation.
Version	string	1		The version of this implementation, as a string.
Architecture	SEArch_t	1		Underlying architecture: disk, tape, multidisk, other.
SizeTotal	int	1	GB	Total size of storage space for this SE. Deprecated.
SizeFree	int	1	GB	Total size available for this SE (sum over all SAs). Deprecated.
TotalOnlineSize	int	1	GB	Total size of online storage space for this SE
TotalNearlineSize	int	1	GB	Total size available for this SE (sum over all SAs)
Port	int	1		Deprecated.
StateCurrentIOLoad	string	1		Deprecated.
Status	enum	1		Production, Draining, Closed, Queueing

Notes:

- Note, storage unit is GB, not GiB (GiB is 2^{30}, GB is 10^9). The conversion factor $k \mapsto 2^{10k}/10^{3k}$ increases with k.
- TotalOnlineSize and TotalNearlineSize are meant to publish estimates of the total space (across all SAs) with the specific AccessLatency.
- ImplementationName is an open-ended enum, with names being agreed by GLUE. Examples: “dCache”, “DPM”, “StoRM”, “CASTOR”.

7.4 StorageArea

StorageArea

Property	Type	Mult.	Unit	Description
LocalID	string	1		Identifier, unique within the SE
Root	string	1		Deprecated
Path	string	1		Path for clients not using VOInfo.Path (or if no VOInfo available)
Type	StorageType	1		Deprecated; now using ExpirationMode instead.
Capability	string	*		Other policies and optional implementation-dependent parameters
RetentionPolicy	RetentionPolicy_t	1		Custodial, Output, Replica
AccessLatency	AccessLatency_t	1		Online, Nearline
ExpirationMode	ExpirationMode_t	*		neverExpire, warnWhenExpired, releaseWhenExpired

7.4.1 Supporting multiple VOs

There is a facility for letting multiple VOs share an SA. The facility also exists in 1.2 (AccessControlBase) but was not used in practice. It works by publishing multi-valued FQAN within a single SA; in practice with 1.2, each SA was published multiple times, once for each VO.

Multiple VOs are supported via the VOInfo class. Note that a client that discovers an SA it wishes to use must then discover the space token or path it will use to write data into the space by looking up in the VOInfos associated with the SA, by matching them with FQAN. The SA must continue to publish a multivalued AccessControlBase for backward compatibility.

VOInfo

Property	Type	Mult.	Unit	Description
AccessControlBaseRule	string	1		Rule for access control, see AccessControlBase class.
Path	string	1		Path used by VO for writing into SA.
Tag	string	1		A string associated with this VO's use of the SA.
Name	string	1		Short well-defined name of VO
LocalID	string	1		Id, unique within this SE

- VO doesn't have to be a full VO, but can be defined by its name or the access control base, so could be a subgroup of a VO. It is recommended that even if the access control is published, the "Name" is also published, for clients that don't know about access control base, or don't care.
- Some SRM implementations use a space token description to select the SA, but it may be different for different VOs. The Tag attribute can be used to publish this.

7.4.2 Space support

- Version 1.2 had the problem that when VOs shared an SA, the available space was sometimes published as available for all the VOs, leading some clients to incorrectly add all the available space. In this version, this problem is "fixed" by publishing multiple VOs within the same SA (this was also possible in 1.2 but only via AccessControlBase).

- Capability describes implementation-specific key/value pairs that the client can use to ask for hints when using the control protocol. The intention is that these can be optional, and an implementation can add further key-value pairs.

7.4.3 Publishing Quality of storage

- Note on the file lifetime: files have a lifetime, that is, a time by when they "expire". Here, however, "file lifetime" describes the *space's policy for removing files that have expired*. The original SRM 2.1 definition defined three values, volatile, durable, and permanent. Since then, these values have had their semantics redefined, so we need to make this attribute deprecated and replace it with the ExpirationMode, AccessLatency, and RetentionPolicy attributes.

- As a replacement for the *original* definition of lifetime, the ExpirationMode attribute can take the following values:

  - releaseWhenExpired: file is released silently when it expires; this was known as volatile in the SRM 2.1 spec.
  - warnWhenExpired: a warning is issued in an implementation-defined way when it expires; this was known as durable in the SRM 2.1 spec.
  - neverExpire: the file never expires; its lifetime is effectively infinite, and will never be deleted by the SE. This was known as permanent in the SRM 2.1 spec.

- Note that ExpirationMode is an attribute of the space, the SA. The files themselves may have lifetimes.

- SEs SHOULD publish the file lifetime enum for backward compatibility. It should be consistent with the ExpirationMode, but FileLifetime is single valued so should publish the "least" supported ExpirationMode, using the SRM 1 enum. Example: an SA that supports both releaseWhenExpired and warnWhenExpired should publish type=volatile.

- Note that a released file is not necessarily deleted. A typical implementation will flag it for garbage collection.

- Note that RetentionPolicy has a linear order: `custodial` is better than `output` which in turn is better than `replica`.

- AccessLatency similarly is ordered: `Online` is "better" than `nearline`. Note that, like RetentionPolicy, it is an attribute of the *space*: the fact that some files are online (temporarily or not) is not sufficient to say the space is `online`. For a space to be `online`, it must guarantee that *all* files are online *all* the time.

7.4.4 State

Since there is a $1-1$ mapping between SA and State, State is usually subsumed within the SA (it is documented separately for backward compatibility reasons with 1.2). Likewise for the link from VOInfo (q.v.) to State.

State

Property	Type	Mult.	Unit	Description
UsedSpace	int	1	KB	Space used within SA
AvailableSpace	int	1	KB	Unused and reclaimable ("free") space
UsedOnlineSpace	int	1	KB	Used online space
UsedNearlineSpace	int	1	KB	Used nearline space
TotalOnlineSpace	int	1	KB	The total online space in this SA.
TotalNearlineSpace	int	1	KB	The total nearline space in this SA.
ReservedOnlineSpace	int	1	KB	The reserved online space in this SA.
ReservedNearlineSpace	int	1	KB	The reserved nearline space in this SA.

7.4.5 AccessControlBase

Since there is a $1-1$ mapping between SA and AccessControlBase, AccessControlBase is usually subsumed within the SA as an AccessControlBaseRule attribute (it is documented separately for backward compatibility reasons with 1.2). The same is true for VOInfo's relation to AccessControlBase.

AccessControlBase

Property	Type	Mult.	Unit	Description
Rule	string	*		Authorization scheme (*scheme*':'*auth*)

- Each Rule is of a scheme (like URI) followed by the ':' character, and then the authorization string. Schemes can be "VO" (followed by VO name), or "VOMS" followed by a VOMS FQAN.

7.5 ControlProtocol

The Control Protocol is the primary interface to the SE. The SE may provide multiple instances of the control protocols typically by supporting multiple control versions or interfaces on different networks.

ControlProtocol

Property	Type	Mult.	Unit	Description
LocalId	string	1		Local Identifier, unique within a specific Storage Element instance
Type	enum	1		Type of control protocol
Endpoint	string	1		Web services endpoint, a URI
Version	string	1		Protocol Version
Capability	string	*		Additional properties supported by this access protocol

7.6 AccessProtocol

An access protocol describes the allowable ways to transfer files to and from an SE. It is published as a URL scheme, cf. RFC 2396, section 3.1. Examples include gsiftp (not GridFTP or gridftp), rfio, dcap, and file. The protocol is defined by Type, which is an enumerated list of allowed values. The main value is the protocol version as entire URL should be obtained through negotiation with the control protocol.

The SE may have multiple instances of every access protocol but providing the attributes remain constant only one instance may occur within the schema.

In addition to there is a multi-valued string called "Capability" which can be used to identify particular features. For example a GridFTP server may support a maximum of 10 concurrent streams or is limited in its features. These values are not bound by the schema and are left to provide extensible hooks for Grid projects to extend, they may in practice be agreed between projects and be elevated to the status of real attributes in later versions of the Glue project. The usage is similar to the RunTimeEnviroment attribute for the CE.

Note that endpoint and port should be optional and will not necessarily be specified for SE's providing the Control protocol.

AccessProtocol

Property	Type	Mult.	Unit	Description
LocalId	string	1		Local Identifier, unique within a specific Storage Element instance
Type	string	1		Type of access protocol
Endpoint	string	1		Web services endpoint, a URI
Version	string	1		Protocol Version
Capability	string	*		Additional properties supported by this access protocol
AccessTime	int	1		Deprecated
SupportedSecurity	string	*		The security feature supported by this protocol, deprecated
Port	int	1		Deprecated
MaxStreams	int	1		Number of streams for protocols that support this

Notes:

- Capability can be used to publish MaxBufferSize, MinBufferSize, MaxBandwidth, etc.

7.7 Policy

Policy

Property	Type	Mult.	Unit	Description
Quota	int	1	**KB**	Quota
MinFileSize	int	1	**bytes**	Smallest file that may be written in the SA
MaxFileSize	int	1	**bytes**	Largest file that can be written by policy
MaxData	int	1	**bytes**	Max data that can be written by a single client
MaxNumFiles	int	1		Max number of files that can be written by a single client
MaxPinDuration	int	1	s	Maximum permitted duration of a PIN
FileLifeTime	StorageType_t	1		Deprecated in favor of SA.Quality

- Note that units are *not* GB, for backward compatibility.

8 Document History

Version	Date	Editors	Description
0.1	2006-09-15	J Jensen, O Synge	First writeup of schema agreed at the meeting
0.2	2006-09-29	J Jensen, O Synge	Second writeup
0.3	2006-09-31	J Jensen	Sent to SRM group for discussion
0.4	2006-10-03	J Jensen	Updates following discussion.
0.5	2006-10-04	J Jensen	Addressed bugfixes from M Litmaath
0.6	2006-10-15	J Jensen	Added SC and AccessType
0.7	2006-10-26	J Jensen	Updated, including all deprecated features; moved VO→VoSaAssociation
0.8	2006-10-29	J Jensen	Changed SRM1Lifetime to StorageType; clarified description of latency and retention for SC. Added a few points regarding the discussion of multiplicities.
0.9	2006-11-06	J Jensen	Major revision updated with suggestions from GLUE group.
0.10	2006-11-06	J Jensen	Restricted VoSaAssociation back in doc, simplified SA following phone mtg. Also put State and AccessControlBase back in.
0.11	2006-11-09	J Jensen	Moved VoSaAssociation→VOInfo, made LocalID explicit in * classes, lots of minor updates
1.0=0.12	2006-11-11	J Jensen	Added link back from SA to AccessProtocol. Renamed NumberOfStreams.
1.01=0.13	2006-11-14	J Jensen	Removed SA→AccessProtocol link. Added reserved and total space to State to be consistent with Sergio's diagram.
1.02=0.14	2006-11-15	J Jensen	Addressed further comments from srm-devel (mainly Maarten and Stephen) on 0.12. Fixed a few naming inconsistencies.
1.03=0.15	2006-11-15	J Jensen	Corrected origin of volatile et al, and addressed bugfixes from Stephen.