Convergence of data solutions within JRA1 in EMI

Formerly known as
Structure of the EMI harmonization effort in storage

Editor : Patrick Fuhrmann
External Reviewer : Gerd Behrmann

ARC : Jon Kerr Nilsen
dCache : Patrick Fuhrmann
gLite : Ákos Frohner and Riccardo Zappi for StoRM
Unicore : Ralf Müller-Pfefferkorn

The purpose of this document is to describe possible improvements in the area of data handling in EMI based on existing storage middleware services in gLite, ARC and UNICORE following the four EMI objectives. The high level goals should be to reduce the cost of maintaining existing storage middleware and to make the EMI distribution sustainable beyond the end of the EMI project. These goals should be achieved by only providing a minimum of development efforts funded through EMI.

1 Current state of the art

1.1 Overall picture

Various components construct the building block of storage handling in the three middlewares. While in gLite, ARC and dCache the storage management software itself plays a significant role, Unicore only focuses on seamless access to data in EMI, even though it provides its own data repository. Although, ARC, gLite and dCache share basic ideas on storage management and control, methodology and deployment differ significantly. While gLite is composed of individual services, covering data storage endpoints, file catalogue and file transfer services; ARC provides a rather tight integration of all those services in one high-level component. dCache purely provides the storage element service including certain client libraries and tools. In terms of maturity and deployment, gLite and dCache share the WCLG/EGEE production infrastructure for some years in a row, while the ARC solution is not deployed in production yet.

	Clients	**Metadata**	**Storage Elements**
ARC	*Chelonia*		
gLite	*FTS, GFAL, posix-io libs*	*LFC*	*DPM, StoRM, CASTOR*
Unicore	*To be integrated*	-	-
dCache	*Posix-io lib, SRM*	-	*dCache*

1.2 Description of components

A **Storage Element** is the atomic unit in the storage Grid infrastructure. It supports a variety of protocol families. Control protocols (SRM), wide area data transfer protocols (gridFtp, http), local posix (like) protocols (dCap rfio,xrood), genuine posix protocols (NFS4.1) and information provider protocols (GLUE). The high-level transfer service **FTS** manages grid wide transfers of datasets between storage elements. Applications on worker nodes access data from the local storage elements by either copying the data to local disks, shared file systems or directly through network protocols. The access can be done through specialized client libraries as well as directly through a mounted file-system if supported by the storage element.
All accesses of the storage element either by users on the worker-nodes or from the FTS service have to be authenticated and authorized. While the GSI mechanism is the currently preferred way of authentication, a more standard way would be appreciated. Alternatives are available as described in the remaining

document.

Although the driving force behind EGEE1/2/3 project(s) and the ARC project has been the LCG community, EMI tries to avoid LCG specific dependencies wherever possible especially where they would prevent other communities from using the EMI middleware bundle.

1.3 Overlapping services

Parts of the middleware stack, described above, provide overlapping services. The ARC Chelonia storage solution offers functionality similar to the gLite Storage Element, File transfer and file catalogues service but integrates all the components into a unified system. The largest overlap can be observed in the area of the storage elements itself. Within gLite, three implementations are provided competing with dCache.

- The **CASTOR** (gLite) Storage Element is targeting the area of very large storage. It needs to cope with storage and access requirements of the LCH Tier 0. It is the only SE supporting tape libraries directly, not requiring intermediate tertiary storage software. It is deployed at CERN, as well as at the Tier I's at RAL, CNAF and in Taipee, however support requires prior agreement with CERN. It is not planed to integrate CASTOR in EMI.
- **dCache** aims for the middle to very large storage sizes. It directly supports disk storage and can drive tertiary storage back-ends (e.g. HPSS, TSM, etc) if connected. dCache manages the largest share of LHC data outside CERN as it is in production at eight out of eleven WLCG Tier I centers and more than 40 Tier II's. The largest installations have been already growing beyond 3 Petabytes on disk and on tape. The most interesting single installation spans 4 countries.
- **StoRM** (gLite) is an SRM 2.2 implementation on top of file systems supporting POSIX file permissions and Access Control Lists (ACL), allowing genuine posix file access. GPFS and Lustre are currently supported. StoRM can driving tertiary storage, if supported by the underlying file system (as with GPFS). It supports small to middle size storage and is aiming for large storage capacities, as it is supposed to be used at the Italian Tier I.
- **DPM** (gLite) is designed to be useful for small to middle storage sizes. Large implementations are currently not envisioned, as it doesn't provide a tape backend support.
- In the US, dCache and DPM are in use in addition to **BeStMan**, a small to medium size storage element. dCache and Bestman are distributed in the framework of the OSG Virtual Data Toolkit. It is not planed to integrate BeStMan in EMI.
- Chelonia targets relatively small communities that are not necessarily familiar to the Grid environment, but are in need of an expandable, distributed storage solution for data sharing and collaboration.

Technical details	**HSM**	**Request Scheduling**	**Wide Area I/O**	**Posix like access**	**Real Posix**
dCache	Any	Yes	gridFtp, http, WebDAV	dCap, xrood	NFS 4.1
StoRM	If supported by file system backend. E.g : HPSS and TSM through GPFS	If supported by file system backend	gridFtp	Xroot	Provided by GPFS or Lustre
DPM	None	None	gridFtp, http	Xroot, rfio, DICOM	None

2 Issues and requirements

This section lists topics being identified as possible problems or as issues that could enable or improve interoperability between the three middlewares or increase the acceptance by the user community. Please refer to a later section on how EMI would like to tackle those items.

2.1 Authentication

Accessing or controlling storage requires proper authentication. Currently the Grid infrastructure is based on X509 certificates and protocols are using the Grid Security Infrastructure (GSI) to do authentication. There is the clear goal of replacing this by the standard SSL/TLS framework as this had becoming an industry standard in the World Wide Web. There is however a GSI functionality which can't be mapped to SSL/TLS one to one. This is the ability to delegate an operation. Delegation allows a service to do something on behave of a certificate owner. In the context of the data grid, this is required to drive HSM back-ends and to delegate transfer services between two or more SE's. The issue can be overcome by advanced techniques like '*one time tokens*' or separate delegation services. gLite and Grid Site provide such services. Those approaches have to be coordinated with Grid partners outside of Europe (OSG) to ensure future interoperability. This topic is covered by the 'security group' within this work package (JRA1).

2.1 Synchronization of name spaces

In gLite, file catalogues provide the mapping between global filenames to *Storage URL*'s, pointing to the actual location of the data within *Storage Elements*. As the File Catalogue and the Storage Element service are independent, references may get out of sync, with the consequence that either Storage Elements may contain datasets that are no longer referenced by the File Catalogue or the File Catalogue entries point to non-existing data in the Storage Element. Although with Chelonia (ARC) this shouldn't happen by design, the problem remains when the ARC service has to interact with gLite. There have been attempts in the past to solve this issue. A general agreement couldn't be achieved yet.

2.2 Storage Resource Management consolidations

The Storage Resource Manager protocol has been introduced to allow remote Space and File Storage Attribute Management. All Storage Elements implement the WLCG interpretation of the SRM 2.2 specification. However, painless interoperability is still not achieved. A clear interpretation of the specification is essential for Unicore as they plan to us the SRM protocol for accessing data from outside the Unicore domain. ARC offers an SRM client but there are no plans to implement SRM on the server side.

2.3 Native Posix Data Access and access control

To allow client applications to access data directly within the storage systems without copying it to local storage first, all Storage Element implementations provide mechanisms for direct random access. While StoRM leaves it to the underlying file system, DPM provides two proprietary protocols (rfio,xroot) together with the corresponding client libraries; applications have to be link against. dCache offers both. It supports dCap and xrood with the corresponding client libraries as well as NFS4.1, a standard network file system protocol. Chelonia uses FUSE, a third party open source component, running on the client node. DPM and dCache provide fine-grained (posix) access control (ACLs). StoRM leaves the access control to the underlying file system while FUSE doesn't provide access control. However in the context of Chelonia, a Command Line Tool can be used to set ACLs. As the use of FUSE is limited to Unix flavors, the Chelonia team would prefer to implement NFS 4.1 (pNFS) instead. It is essential for the acceptance of an EMI distribution that a common industry standard is offered to remotely access data in a posix like fashion.

2.4 Wide area protocols and firewall issues

Currently gridFtp (including version 2) is required to allow wide area data transfers between Storage Elements and partially between storage elements and worker nodes. The protocol is well specified (IETF) and established in the HEP grid world. However it is a burden in terms of firewall settings and not in use by generic web applications. HTTP(S) would be a solution for both issues. In addition, clients tools are freely available in all operating systems and browsers to access data through HTTP(s). Most Storage Elements

support this protocol already to some extends. There are however issues to solve before it could possibly replace gridFtp. These are authentication, proxy delegation and 3rd party transfers.

2.5 Client access to storage (Unicore)

In order to allow Unicore Jobs to seamlessly access data in the gLite and ARC world, storage management and data access protocols would still need to be integrated. SRM is envisioned for storage control. Some more evaluation is needed to decide on the optimal remote file access to Storage Elements.