EUROPEAN MIDDLEWARE INITIATIVE

EMI DATA LIBRARY CONSOLIDATION PLAN

EU TECHNICAL NOTE: DXN.M

Document identifier:	**EMI_datalib-consolidation-EMI-tech-doc-v1.doc**
Date:	**17/08/2011**
Activity:	**JRA1**
Lead Partner:	**UiO**
Document status:	**DRAFT**
Document link:	

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Abstract:
In this document a plan for consolidating the gLite and ARC data libraries into one EMI data library is described.

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Delivery Slip

	Name	Partner / Activity	Date	Signature
From	Jon Kerr Nilsen	UiO/JRA1		
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author / Partner
1	9/8/2011	First draft	Jon Kerr Nilsen/UiO, Zsombor Nagy/NIIFI
2	17/8/2011	Update based on meeting 15/8/2011	Jon Kerr Nilsen/UiO
3			
4			
5			
6			
7			
8			

Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

TABLE OF CONTENTS

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *8*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: 9

Jon Kerr Nilsen 16.8.11 10:53
Deleted: *libarcdata*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *9*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *9*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *9*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *9*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *9*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *10*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *10*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: *10*

Jon Kerr Nilsen 16.8.11 09:29
Deleted: *10*

Jon Kerr Nilsen 16.8.11 09:29
Deleted: *10*

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

1. INTRODUCTION

1.1. PURPOSE

The goal of the EMI Data Library Consolidation activity is to find duplicate functionality within the ARC and gLite data libraries and to merge them in order to reduce the amount of code to maintain.

To achieve this we have to examine the current status of the gLite and ARC data libraries.

1.2. ABOUT THIS DOCUMENT

This document describes the current gLite and ARC data libraries and the plan on how to merge them.

Disclaimers, background, etc.

The document is made by the EMI Data Library Consolidation TF in the EMI Data group, Zsolt Molnar, Zsombor Nagy, Jon Kerr Nilsen and Michail Salichos and David Cameron from NDGF.

1.3. REFERENCES

R1	
R2	
R3	

1.4. DOCUMENT AMENDMENT PROCEDURE

This document can be amended by the authors further to any feedback from other teams or people. Minor changes, such as spelling corrections, content formatting or minor text re-organisation not affecting the content and meaning of the document can be applied by the authors without peer review. Other changes must be submitted to peer review and to the EMI PEB for approval.

When the document is modified for any reason, its version number shall be incremented accordingly. The document version number shall follow the standard EMI conventions for document versioning. The document shall be maintained in the CERN CDS repository and be made accessible through the OpenAIRE portal.

1.5. TERMINOLOGY

2. TECHNICAL DESCRIPTION

2.1. DESCRIPTION OF THE CURRENT DATA LIBRARIES

2.1.1 gLite Data Library

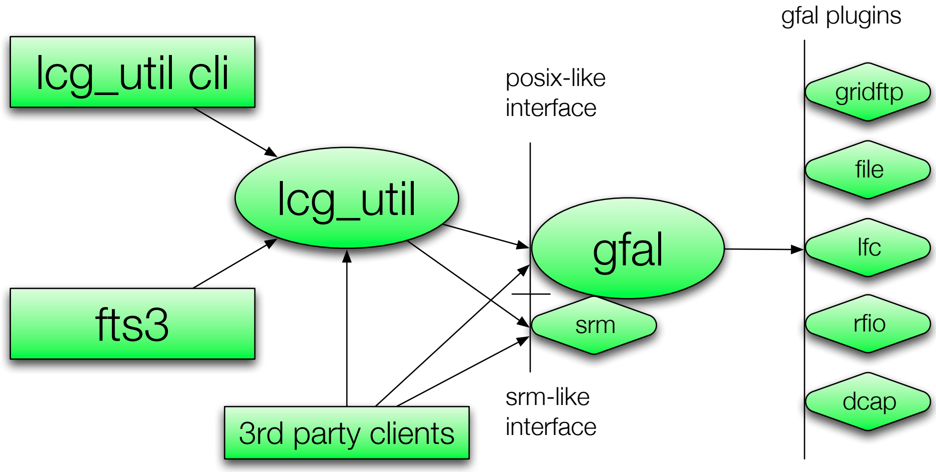

Figure 1: gLite Data Library Architecture

The GFAL library provides a posix-like interface for data access. It has a pluggable architecture, where different plugins provide support for different data transfer protocols. It also has a plugin for LFC (file catalog) interaction, which can resolve logical names to transfer URLs.

Besides the posix-like interface, GFAL has an SRM interface which also supports usage of space tokens.

Also part of the gLite data library is the lcg_util library, which provides higher-level file operations and which uses GFAL for the actual file access. The lcg_util command line tools are thin wrappers around the lcg_util library.

The newest version of FTS (file transfer service) also uses the lcg_util library.

There are several third party clients (integrated into experiments' software) using the GFAL library or the lcg_util library (some through the python API).

The GFAL and lcg_util libraries are written in C.

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 15.8.11 14:07
Deleted: -

Jon Kerr Nilsen 15.8.11 14:07
Deleted: s

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

2.1.2 ARC Data Library

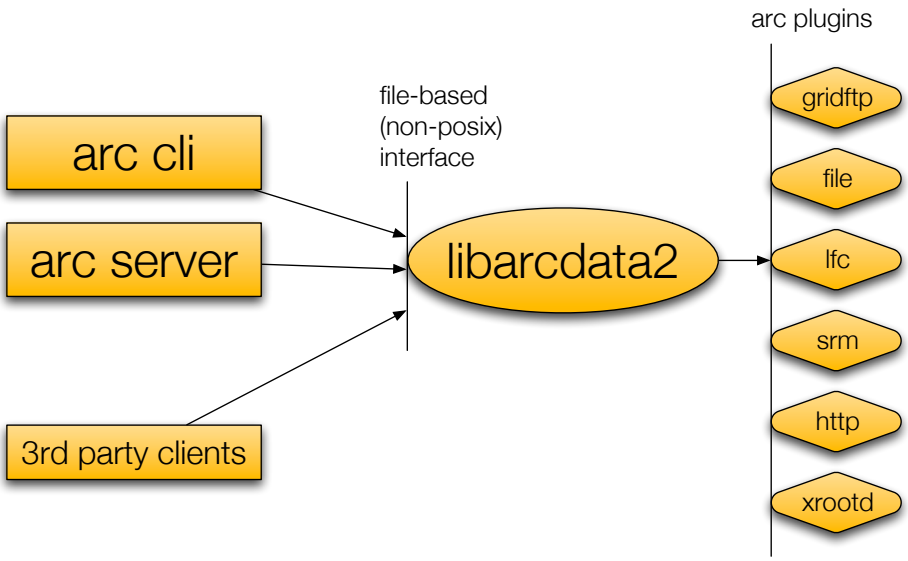

Figure 2: ARC Data Library Architecture

The ARC data library called libarcdata2 has a pluggable architecture, different plugins providing support for different data transfer protocols. It is a file-based (non-posix) interface providing higher-level operations to read and write files. It has plugins for LFC and SRM operations (also supports usage of space tokens).

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

The arc command tools and also the A-REX service on the computing element are using this library to upload and download files. Third-party clients also exist (some of them using the python API).

The libarcdata2 is written in C++.

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

2.2. CONSOLIDATION PLAN

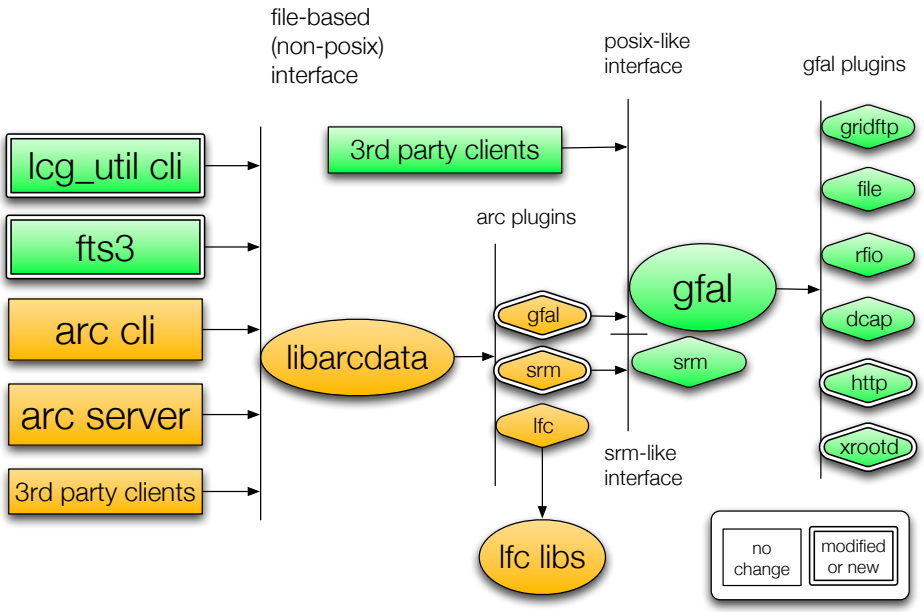

Figure 3: Proposed arcitecture of EMI_datalib

The proposed architecture (Figure 3) will keep both the posix-like interface (provided by the GFAL library) and the higher-level file-based interface (provided by the libarcdata2 library).

Two main connection points: libarcdata2 would use GFAL for the data access, and the lcg_util command line tools and the FTS would use libarcdata2 for the file transfers. The lcg_util library would be removed (a compatibility layer would be provided for existing clients of the lcg_util library - this layer would translate the lcg_util calls to libarcdata2 calls). Also most of the libarcdata2 plugins would be removed or recreated as GFAL plugins.

2.2.1 Benefits

- One set of file transfer plugins instead of two
- One set of control plugins instead of two
- One file-based library and one posix-like library
- GFAL gets support for http and xrootd transfer
- ARC gets support for third party transfer
- No change for existing users
- A clear division of components, with one file-based layer and one underlying posix layer with most of the transfer plugins should simplify support.

2.2.2 Disadvantages

- As development of GFAL will need to focus on the consolidated data library, no new features will be added to the current lcg_util/GFAL implementation. This means that users need to do more radical changes to get new features.

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 15.8.11 14:17
Deleted: two

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 17.8.11 10:27
Deleted: o

Jon Kerr Nilsen 17.8.11 10:27
Deleted: o

Jon Kerr Nilsen 17.8.11 10:27
Deleted: o

Jon Kerr Nilsen 17.8.11 10:27
Deleted: n

Jon Kerr Nilsen 17.8.11 10:20
Formatted: Heading 3;l3;H3;Level 2 Heading;Level 2;h2;h3;1.2.3.;T3;H31;T31;l31;Level 2 Heading1;Level 21;h21;h31;1.2.3.1;H32;T32;l32;Level 2 Heading2;Level 22;h22;h32;1.2.3.2;H33;T33;l33;Level 2 Heading3;Level 23;h23;h33;1.2.3.3;H34;T34;l34;Level 2 Heading4, No bullets or numbering

Jon Kerr Nilsen 17.8.11 10:20
Formatted: Bulleted + Level: 1 + Aligned at: 0,63 cm + Indent at: 1,27 cm

- One may argue that the consolidated library does not lead to a huge removal of lines of code when compared to the effort needed.

2.2.3 Components to be removed

- lcg_util library
- libarcdata2 plugins
  - file
  - gridftp
  - srm
  - http
  - xrootd

2.2.4 Components to be created

- GFAL plugin for ARC
- http plugin for GFAL
- xrootd plugin for GFAL

2.2.5 Components to be modified

- FTS (to use libarcdata2)
- lcg_util command line tools (to use libarcdata2)
- the SRM plugin of ARC (to use GFAL)

2.2.6 Estimated amount of work needed

gLite:

1. need to finalize GFAL2
   - Work already shows promising results
   - Already more stable than GFAL
   - Will definitely be ready for EMI2
2. removing lcg_util is unproblematic
3. start with FTS built on arclibraries

ARC

1. ARC will start creating a GFAL2 plugin
   - Focus on SRM and third-party transfer part
2. The removal of plugins should be unproblematic
   - May require some internal repackaging of ARC components

2.2.7 Estimated SLOC change

- lfc_util:
- Removed ARC plugins (not counting http and xrootd):
- ARC GFAL plugin:

The modified components are not foreseen to change significantly in terms of SLOC. The http and xrootd plugins removed from ARC will be recreated as GFAL plugins and are not foreseen to change the overall SLOC.

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Jon Kerr Nilsen 17.8.11 10:20
Formatted: English (UK)

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:30
Deleted: <#>lcg-utils compatibility C and python API

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:30
Deleted: lcg-utils

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 17.8.11 09:57
Formatted: Heading 3;l3;H3;Level 2 Heading;Level 2;h2;h3;1.2.3.;T3;H31;T31;l31;Level 2 Heading1;Level 21;h21;h31;1.2.3.1;H32;T32;l32;Level 2 Heading2;Level 22;h22;h32;1.2.3.2;H33;T33;l33;Level 2 Heading3;Level 23;h23;h33;1.2.3.3;H34;T34;l34;Level 2 Heading4, No bullets or numbering

Jon Kerr Nilsen 17.8.11 10:05
Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0,63 cm + Indent at: 1,27 cm

Jon Kerr Nilsen 17.8.11 10:05
Formatted: Bulleted + Level: 2 + Aligned at: 1,9 cm + Indent at: 2,54 cm

Jon Kerr Nilsen 17.8.11 10:05
Formatted ... [1]

Jon Kerr Nilsen 17.8.11 10:05
Formatted ... [2]

Jon Kerr Nilsen 17.8.11 10:09
Formatted ... [3]

Jon Kerr Nilsen 17.8.11 10:10
Formatted ... [4]

Jon Kerr Nilsen 17.8.11 09:57
Formatted ... [5]

Jon Kerr Nilsen 17.8.11 10:13
Formatted ... [6]

Jon Kerr Nilsen 17.8.11 10:13
Formatted: List Paragraph

Jon Kerr Nilsen 17.8.11 10:16
Formatted: No bullets or numbering

2.3. QUESTIONS AND ANSWERS

2.3.1 Could GFAL use libarcdata2 as a plugin instead?

GFAL is written in C and has a POSIX-like interface, libarcdata2 is written in C++ and has a file-based interface. It doesn't seem feasible to transform posix-like C queries into file-based C++ queries.

2.3.2 How to support the transition period?

The libarcdata2 library can keep all the plugins until the functionality of the GFAL plugin (in libarcdata2) is not at the expected level. The existing users of GFAL will be able to use it as before. An lcg_util compatibility layer will be created which transforms the lcg_util calls to libarcdata2 calls. If this layer works the same way as the current lcg_util library, only then can the lcg_util library be dropped. The lcg_util command line tools and FTS can use the compatibility layer first, then later libarcdata2 directly. If any change to the interface of libarcdata2 is needed, it should be done in a backward compatible way, this way the existing clients of libarcdata2 would function the same way as previously.

2.3.3 What about platforms outside EMI?

Until the platform support (different linuxes, Mac, Windows) of GFAL is not sufficient to maintain the current level in ARC, libarcdata2 would have to keep the existing plugins, and support them outside EMI. Within EMI there could be no effort to make GFAL more platform independent.

2.3.4 What about external dependencies?

GFAL itself has only internal dependencies, only the plugins have their own dependencies. However one of the dependencies is an internalized version of gSOAP and a security layer on top if it, called cgsi. The new GFAL implementation does not depend on cgsi and will function without it.

2.3.5 What about the SRM support?

Currently both GFAL and libarcdata2 has a native implementation of the SRM protocol with roughly the same capabilities. The one in GFAL cannot be removed without affecting existing clients and removing the possibility of one-step posix-like opening of SRM URLs. The GFAL plugin within libarcdata2 will have a posix-like interface where the SRM commands would not fit. The proposed solution is to rewrite the current SRM plugin of libarcdata2 to call the corresponding methods of the SRM interface of GFAL, which makes GFAL do the actual SRM calls.

2.3.6 What about the http and xrootd support?

The libarcdata2 currently have strong http support and experimental xrootd support, although the xrootd plugin is just a thin wrapper around the xrootd libraries. There is a request for GFAL to support http and xrootd. If these plugins would stay in the libarcdata2 and would be still absent from GFAL, then the new user would be able to do transfer only through the file-based interface. This may make sense in the http case, but the xrootd is much more a posix-like protocol. If a http and an xrootd plugin would be created within GFAL, then the user of the posix-like GFAL interface would also benefit from these new protocols, and the file-based support would also remain, even after the http and xrootd plugins of libarcdata2 would be removed. Although the http plugin can especially cause concerns about unneeded dependencies and platform-independence, because the current libarcdata2 version has no dependencies and runs on every platform, this makes the keeping of the current libarcdata2 plugin (outside EMI) even more likely.

There is an other important point of having the http and xrootd plugins ported to GFAL: because the SRM would be handled by GFAL, if the SRM request would return a http or xrootd transfer URL and GFAL would not have these plugins, it would be complicated to hand back the TURL to libarcdata2 which knows how to deal with those protocols.

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata…ibarcdata2 library car… [7]

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 11:26
Deleted:

Jon Kerr Nilsen 17.8.11 10:33
Deleted: Question: is the GFAL library itself able to work without the globus libraries?

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata…ibarcdata2 has a nativ… [8]

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata…ibarcdata2 currently h… [9]

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

2.3.7 What about LFC support?

The Jibarcdata2 LFC plugin uses the official LFC libraries, the same is true for GFAL. The LFC reuqest coming through the file-based Jibarcdata2 interface would be resolved by the currently existing Jibarcdata2 LFC plugin. The same is true for the LFC registration request. Any further namespace management is not supported, and it won't be supported. The new GFAL library supports posix-like support for LFC (e.g., it is possible to mount an LFC site using GFAL).

As both the GFAL and ARC plugins for LFC are just thin wrappers to the official LFC libraries and their usage is slightly different, it is currently not clear how much can be saved by to removing any of these plugins.

2.3.8 What about VOMS support?

The ARC data clients have no VOMS dependencies as proxies are not handled explicitly but handed over to other components. GFAL2 will keep supporting VOMS and, as such, has a dependency on the VOMS library. Additionally cgsi depends on depends on VOMS, but cgsi will probably not be supported in GFAL2.

2.3.9 What about support for third party transfer?

GFAL currently does support third party file transfers (between two servers). The Jibarcdata2 does not support this, but with the GFAL plugin it should only pass through the options, and GFAL would provide it. It should be noted that a (trivial) extension of the Jibarcdata2 interface may be needed for this.

2.3.10 What about information system support?

Currently GFAL is able to query some information from the information system. The Jibarcdata2 doesn't have any information system support. GFAL needs some parameters for the queries which it gets from environment variables, so in theory it could keep doing this even if a request came through Jibarcdata2. But this may involve changeing slightly the URL handling of Jibarcdata2 to let through shortened/invalid URLs which would be resolved by GFAL from the information system.

2.3.11 What about naming and packaging?

We think that it is not necessary right now to rename the libraries (e.g. removing the name "arc"). The packaging is not within the scope of this document, but it may be natural that the name EMI_datalib is is introduced as a (meta)package.

2.3.12 Who will use this outside/after EMI?

Naturally, the ARC CE depends on libarcdata2, and will need to move over to the new EMI_datalib. Since libarcdata2 is already plugin-based, this is not foreseen to have much practical implication but should of course be properly tested. Additionally, FTS, which now has started a redesign to use GFAL for data transfer, will need to use EMI_datalib. Since FTS is file-based, the proposed data library should not be too problematic for FTS.

Jon Kerr Nilsen 17.8.11 12:25
Deleted: EMI_datalib-consolidation-EMI-tech-doc-v1.docx

Jon Kerr Nilsen 15.8.11 10:40
Deleted: 09/08/2011

Jon Kerr Nilsen 16.8.11 11:29
Deleted: Becauce currently GFAL doesn't provide posix-like support for LFC, this functionality could be removed from GFAL, and kept intact within libarcdata.

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 16:03
Deleted: Question: do third-party clients use any kind of (non-posix) LFC operation through the current GFAL library?

Jon Kerr Nilsen 16.8.11 16:05
Deleted: Question: what is the current VOMS support within GFAL, within ARC, what dependencies it has?

Jon Kerr Nilsen 16.8.11 16:11
Deleted:

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 16:13
Deleted: It is possible that a

Jon Kerr Nilsen 16.8.11 16:13
Deleted: slight

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 16:13
Deleted: is

Jon Kerr Nilsen 16.8.11 16:13
Deleted: Question: How much extenstion is needed of the libarcdata for this?

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 10:53
Deleted: libarcdata

Jon Kerr Nilsen 16.8.11 16:14
Deleted: ... [10]

Jon Kerr Nilsen 17.8.11 09:37
Formatted ... [11]