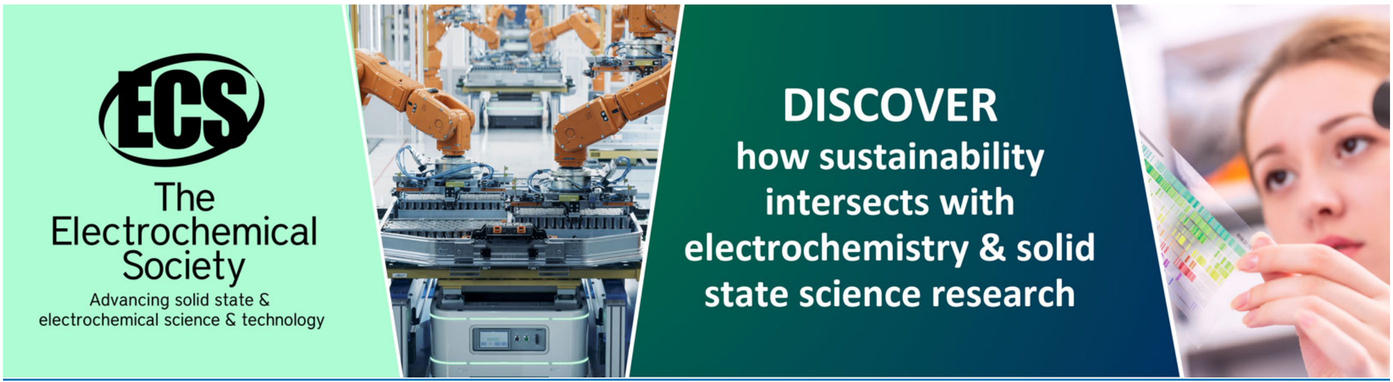
Journal of Physics: Conference Series

Securing the AliEn File Catalogue - Enforcing authorization with accountable file operations

To cite this article: Steffen Schreiner *et al* 2011 *J. Phys.: Conf. Ser.* **331** 062044

View the article online for updates and enhancements.



ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

Securing the AliEn File Catalogue - Enforcing authorization with accountable file operations

Steffen Schreiner[1,2], Stefano Bagnasco[3], Subho Sankar Banerjee[1], Latchezar Betev[1], Federico Carminati[1], Olga Vladimirovna Datskova[1], Fabrizio Furano[1], Alina Grigoras[1], Costin Grigoras[1], Patricia Mendez Lorenzo[1], Andreas Joachim Peters[1], Pablo Saiz[1], Jianlin Zhu[4]

[1] CERN, Geneva, Switzerland
[2] CASED, Darmstadt, Germany
[3] Istituto Nazionale di Fisica Nucleare, Torino, Italy
[4] Huazhong Normal University, Wuhan, China

E-mail: steffen.schreiner@cern.ch

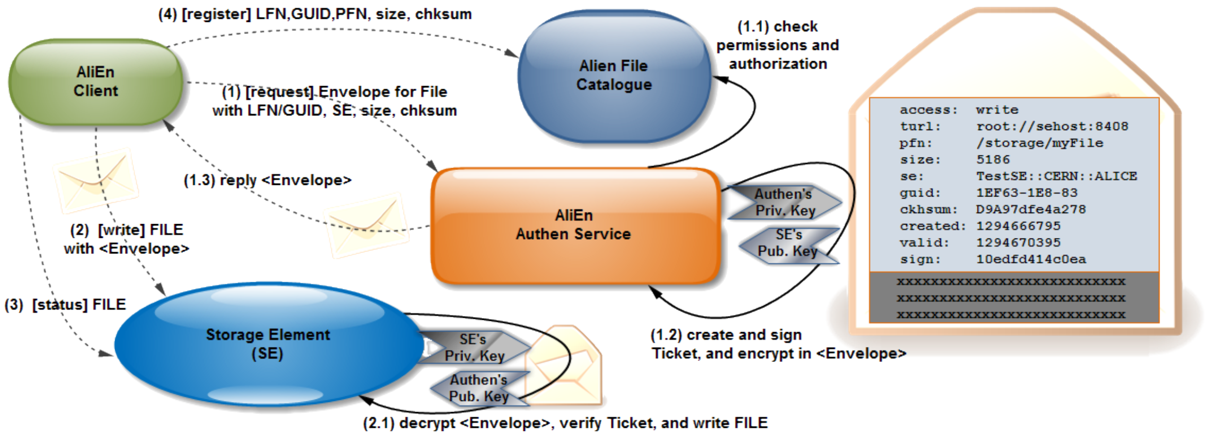
Abstract. The AliEn Grid Services, as operated by the ALICE Collaboration in its global physics analysis grid framework, is based on a central File Catalogue together with a distributed set of storage systems and the possibility to register links to external data resources. This paper describes several identified vulnerabilities in the AliEn File Catalogue access protocol regarding fraud and unauthorized file alteration and presents a more secure and revised design: a new mechanism, called LFN Booking Table, is introduced in order to keep track of access authorization in the transient state of files entering or leaving the File Catalogue. Due to a simplification of the original Access Envelope mechanism for xrootd-protocol-based storage systems, fundamental computational improvements of the mechanism were achieved as well as an up to 50% reduction of the credential's size. By extending the access protocol with signed status messages from the underlying storage system, the File Catalogue receives trusted information about a file's size and checksum and the protocol is no longer dependent on client trust. Altogether, the revised design complies with atomic and consistent transactions and allows for accountable, authentic, and traceable file operations. This paper describes these changes as part and beyond the development of AliEn version 2.19.

1. Introduction

The AliEn [1, 2] Grid Services, developed and operated by the ALICE Collaboration [3, 4] as a global physics analysis grid framework, are built upon one central File Catalogue [5]. Based on a globally distributed set of trusted storage systems, provided by computing sites within the ALICE Collaboration as so called Storage Elements (SE), it constitutes one global grid file system. An entry in the File Catalogue is represented by a global unique identifier (GUID) linked to one or more logical file names (LFN) within the Catalogue's file system hierarchy. Each, GUID and LFN entries have their own user and group ownership and read and write permissions, while the GUID's ownership and permissions are overruling. Physical file names (PFN) point to the actual data files as resource locators, which are linked to GUID entries, while one GUID entry can be linked to several PFNs as replicas of the same file. A PFN can point to external resources, e.g. HTTP and SOAP addresses. Nevertheless, all experiment or crucial data is stored on the trusted, xrootd [6] protocol based SEs, which describes the main scenario discussed throughout this paper. Thereby, SEs are unaware of the user identities and per-user access control settings on the Catalogue level. The AliEn Grid Services are granted full access to SEs as one single user, which is why additional mechanisms are necessary in order to allow for fine-grained access control.

2. The AliEn File Catalogue access protocol in v2.18

In AliEn version 2.18 and lower, the clients have a direct database access to the File Catalogue as a central service, authenticated and authorized based on their grid certificates. In order to access a PFN on a SE over the xrootd protocol, a client has to connect to an AliEn central service, called Authen, to retrieve a so called Access Envelope, which contains a capability-based access ticket. Figure 1 shows how a client retrieves and uses an Access Envelope during a write operation: (1) The client sends a request specifying the LFN and/or GUID, SE, file size, and checksum to the Authen service. (1.1) Upon successful authentication and authorization of the request, (1.2, 1.3) the Authen service replies with a digitally signed ticket inside an encrypted Access Envelope (see section 4). (2) The client sends a write request to the SE over the xrootd protocol while passing on the Envelope and the file's data. (2.1) The SE decrypts the Envelope and verifies the ticket in order to authenticate and authorize the access and executes the write operation if applicable. After a successful write attempt, the client (3) requests the status of the written file in order to confirm the file's existence and size. (4) Finally, the client connects to the File Catalogue and registers the file while specifying LFN, GUID, PFN, size and checksum. The decoupling of the logical and the physical layer connected only by the user, respectively the communication layout of one SE access embraced by two central service invocations, is being seen as a key concern of scaling and performance.


(4) [register] LFN,GUID,PFN, size, chksum
AliEn Client
Alien File Catalogue
(1.1) check permissions and authorization
(1) [request] Envelope for File with LFN/GUID, SE, size, chksum
(1.3) reply <Envelope>
(2) [write] FILE with <Envelope>
AliEn Authen Service
Authen's Priv. Key
SE's Pub. Key
(3) [status] FILE
Storage Element (SE)
SE's Priv. Key
Authen's Pub. Key
(1.2) create and sign Ticket, and encrypt in <Envelope>
(2.1) decrypt <Envelope>, verify Ticket, and write FILE

```
access:  write
turl:    root://sehost:8408
pfn:     /storage/myFile
size:    5186
se:      TestSE::CERN::ALICE
guid:    1EF63-1E8-83
ckhsum:  D9A97dfe4a278
created: 1294666795
valid:   1294670395
sign:    10edfd414c0ea
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figure 1. Writing a file in AliEn v2.18 over the xrootd protocol

2.1. *Vulnerability assessment*

Throughout an in-depth security analysis of the access protocol of the AliEn File Catalogue, several security flaws have been identified, summarized briefly as follows:

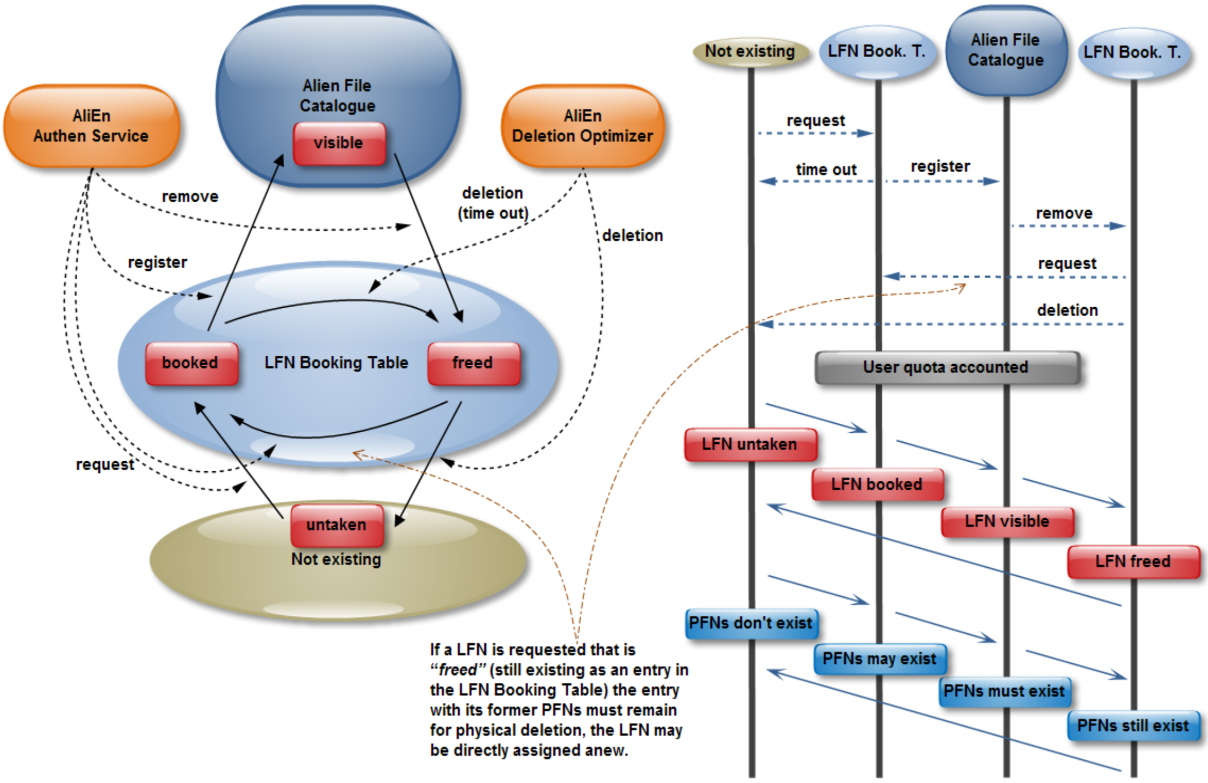
1. While grid file access is limited by the permissions within the grid file system hierarchy and both LFNs and GUIDs are regulated by ownership, the file registration is open and uncontrolled regarding the PFN. The read access on both the actual files and their meta information in the File Catalogue is not restricted, as the grid file system is not designed to hold confidential data.

By retrieving the PFN value of any existing LFN, a user can register the PFN as a new file entry with different LFN and GUID. This will bypass the access control and the user will have write and delete access on the PFN, although the client might not be entitled with write or delete access on original LFN and GUID. Since the PFN for a write access is always generated using SE specific parameters and the GUID of the concerned request, the user will not be able to overwrite the PFN. Yet, the user will be able to retrieve an Access Envelope for deletion from the Authen service and can use the Envelope to physically delete the file. By exploitation of this vulnerability it is even possible for any user to delete all physical files in the file system.
2. Since the authorization during the Access Envelope generation is independent of the later file registration, race conditions regarding LFN and GUID entries may occur. If e.g. two clients attempt to write to the same LFN concurrently, only the first client registering the file will be successful, even if the other client requested an Access Envelope for the corresponding LFN as the first. Within limitations, it is also possible to overwrite physical files owned by other users. A feature of the protocol is the possibility to request write access on any non-existent GUID. Within the time frame of another client's successful upload to a SE and the final registration, it is possible to request write access to the same GUID and SE, and to overwrite the physical file. This can be exploited e.g. to overwrite the output files of grid jobs, as the final registration is delayed with respect to the upload to an SE.
3. As the central Authen service never gets in touch with a physical file in the course of a client's file operation, the service relies on client trust regarding the file's meta information of existence, size, and checksum. As a consequence, it is not possible to distinguish if a file is broken or corrupted at a later state. Accordingly, it is possible to commit fraud against the file system's accounting, respectively a user's quota, and regarding the identity of a file stated by the checksum. The problem persists even if a SE could enforce the correct size and checksum according to the foregoing authorization due to the open file registration and the missing tracking of the authorization.

3. Ensuring consistency with a LFN Booking Table

In order to keep track of Catalogue changing write or delete operations, a new database table, called the LFN Booking Table, is introduced next to the File Catalogue. Whenever a PFN enters or leaves the File Catalogue, an entry is added to the Booking Table and thereby representing the PFN and LFN during its transient state. Figure 2 shows the state modelling of LFN entries and the according conditions for PFNs. In the course of a write request from the client, the Authen service adds a corresponding entry to the Booking Table for each PFN, containing the LFN, GUID, PFN, size, checksum, owner, and further information. This sets the LFN from *"untaken"* to *"booked"*, if it is unknown to the system, or from *"freed"* to *"booked"*, if it was removed or a foregoing request on it timed out (see below). During the registration phase of a write operation, a file entry is removed from the Booking Table and registered to the Catalogue. This sets the LFN state from *"booked"* to *"visible"* once the first PFN is registered. Due to the Booking Table, the registration requests can be based on only the PFN as a parameter, as the PFN is sufficient to identify the corresponding entry, which contains all necessary parameters in order to process the registration. If a user deletes a LFN or PFNs of a LFN in the Catalogue, the physical removal is postponed, only the entries in the File Catalogue are deleted, and the corresponding entry is written to the Booking Table for later physical removal. If a LFN is deleted completely from the Catalogue, this describes the state transition from *"visible"* to *"freed"*. A central process, called the Deletion Optimizer, is checking periodically the Booking Table. If an entry is not moved to the File Catalogue within a certain time frame, the Deletion Optimizer issues a physical removal of the PFN on the SE and removes the entry. The Booking Table has a lifetime value for each entry, describing the distinction of the LFN states *"booked"* and *"freed"* and providing a time out for unregistered write access requests. Once all entries

regarding a LFN in state *"freed"* are physically removed, it is in state *"untaken"*.
By introducing the Booking Table, LFNs and PFNs can be represented in all their respective states. File Catalogue altering operations can be assured to be atomic and consistent with respect to both the physical and the logical file level, and the authorization can persistently enforced.



If a LFN is requested that is *"freed"* (still existing as an entry in the LFN Booking Table) the entry with its former PFNs must remain for physical deletion, the LFN may be directly assigned anew.

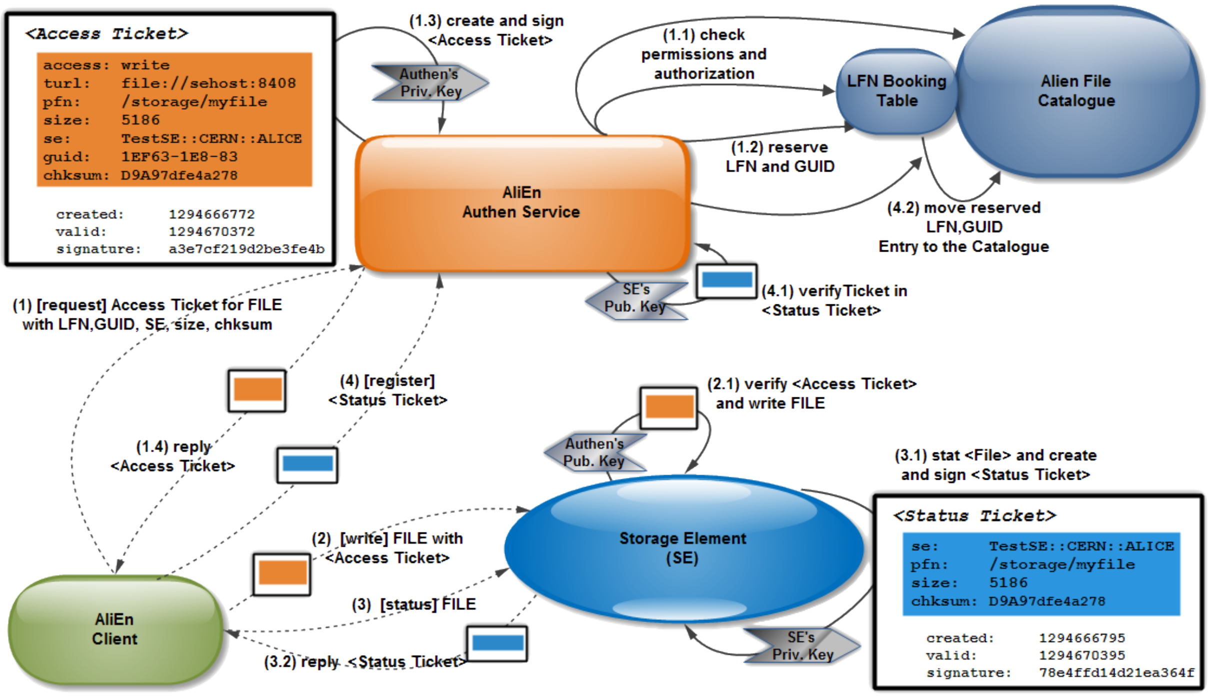
Figure 2. The states and transitions of a LFN utilizing the LFN Booking Table

4. Simplifying the Access Envelope mechanism with Access Tickets

In the original implementation of Feichtinger and Peters [7], the Access Envelope for xrootd-protocol-enabled storage systems is an encrypted access ticket with a RSA public-key signature of a checksum of the ticket. Its validity is limited by a creation and an expiration timestamp. The encryption allows for a secret transmission of information, though this functionality is actually avoided in AliEn. The ticket contains no secret information and in order to allow for informed decisions [8] and detailed logging on the client side, all information is additionally passed on as plain text.
In a new and simplified design, the Envelope mechanism and its encryption is discarded leaving only a signed Access Ticket (see figure 3). The Access Ticket encloses all necessary information and is based on a RSA public-key signature using a SHA384 checksum. Due to the simplification, both creation and verification of the Ticket receive a fundamental computational improvement and the size of the final Ticket is reduced up to 50%.

5. Enforcing file authenticity with replied Status Tickets

Even by adopting the LFN Booking Table, the issue of client trust regarding a file's existence, size, and checksum cannot be solved. The decoupling of data and meta data makes it necessary to introduce additional functionality to enforce the authenticity of a file's meta data with respect to the storage level. Any callback from the resource (SE) to the authorization authority (central services), as utilized e.g. in GridShib [9], would break up the existing communication layout (see section 2) and is therefore disqualified. By introducing a size and checksum verification on SE level together with a second ticket, the Status Ticket, which is created and signed by the SE, the original communication layout of the protocol can be preserved. By replying the Status Ticket from the SE over the client to the Authen service, the existence, size, and checksum for a given PFN can be proven during the file registration.



```
<Access Ticket>
  access: write
  turl:   file://sehost:8408
  pfn:    /storage/myfile
  size:   5186
  se:     TestSE::CERN::ALICE
  guid:   1EF63-1E8-83
  chksum: D9A97dfe4a278

    created:     1294666772
    valid:       1294670372
    signature:   a3e7cf219d2be3fe4b
```

```
<Status Ticket>
  se:     TestSE::CERN::ALICE
  pfn:    /storage/myfile
  size:   5186
  chksum: D9A97dfe4a278

    created:     1294666795
    valid:       1294670395
    signature:   78e4ffd14d21ea364f
```

(1.3) create and sign <Access Ticket>
Authen's Priv. Key
(1.1) check permissions and authorization
LFN Booking Table
Alien File Catalogue
AliEn Authen Service
(1.2) reserve LFN and GUID
(4.2) move reserved LFN,GUID Entry to the Catalogue
SE's Pub. Key
(4.1) verifyTicket in <Status Ticket>
(1) [request] Access Ticket for FILE with LFN,GUID, SE, size, chksum
(4) [register] <Status Ticket>
(2.1) verify <Access Ticket> and write FILE
Authen's Pub. Key
(1.4) reply <Access Ticket>
(3.1) stat <File> and create and sign <Status Ticket>
Storage Element (SE)
(2) [write] FILE with <Access Ticket>
(3) [status] FILE
AliEn Client
(3.2) reply <Status Ticket>
SE's Priv. Key

Figure 3. Status Ticket during a write operation

Figure 3 shows the adoption of the Status Ticket based on the AliEn File Catalogue access protocol with the LFN Booking Table and the new Access Ticket mechanism: (1) The client requests an Access Ticket for writing, optionally specifying size and checksum of the future file. (1.1) After successful authentication and authorization of the request, (1.2) the Authen service writes an entry into the Booking Table. (1.3) It creates and digitally signs the Ticket with its private key and (1.4) sends it back to the client. (2) The client uses the Ticket to request write access at the SE. (2.1) The SE grants the access based on the verification of the Ticket with the Authen service's public key and if successful, (2.2) executes the write operation. (3) The client requests the status of the file. (3.1) The SE calculates size and checksum of the file, creates and digitally signs a Status Ticket and (3.2) sends it back to the client. After checking size and checksum of the written file, (4) the client sends the Status Ticket to the Authen service requesting the registration of the corresponding entry. (4.1) The registration is granted based on the verification of the Status Ticket with the SE's public key and the PFN's consistence with an entry in the Booking Table. (4.2) The file entry is removed from the Booking Table and registered in the File Catalogue, using the size and checksum information from the Status Ticket. Due to the Status Tickets, the authenticity of the file's meta information can be assured, as the SE is considered to be a trusted location and the signature of the Status Ticket can prove its origin to the Authen service.

6. Registration of untrusted data resources

Due to the LFN Booking Table, the possibility of open registration of PFN entries on xrootd-protocol-based SEs becomes unnecessary for normal file operations. The registration of untrusted data resources, as e.g. over HTTP or SOAP, describes a remaining weakness, as it is neither possible to get trustable information on size or checksums, nor to ensure a file content's integrity over a link's lifetime. Files that are registered in the File Catalogue with these protocols should be treated in a different manner than files on a controlled storage systems. The according system changes would be straightforward, as they are either directly registered or follow a different file protocol. Another possibility is to restrict the registration of untrusted data resources to privileged or trusted users, or to generally limit the registration to trusted resources.

7. AliEn v2.19

As of AliEn version 2.19, any access to the file system is authorized by the Authen service and the client has no more direct connection to the File Catalogue. The authentication to the Authen service is based on the HTTPS protocol using grid and derived proxy certificates [10]. The LFN Booking Table and the Access Ticket mechanism are part of AliEn v2.19 as described above. The development of the Status Ticket is expected to be finalized by the publication of this document.

8. Summary

The LFN Booking Table assures atomicity and consistency of operations by modelling the transient state of entering or leaving file entries in the File Catalogue. Moreover, it allows to enforce authorization throughout the two phases of write access to SEs. Signed Status Tickets enable the protocol to have trustable size and checksum information of files, which not only ensure the authenticity of the File Catalogue entry's meta information, but also the correct transmission and storage on a SE. The revised design is fully compliant with the decoupled flow of data and meta data and preserves the initial communication layout. Yet, only by also restricting the direct registration of entries in the File Catalogue as described, a circumvention of the authorization mechanism and malicious exploitation can be prevented.

The presented work is the necessary first step to enable authentic grid jobs in AliEn, where it would be necessary to verify the authenticity of a grid job's executable and data input files.

References

[1] *Alien2* URL `http://alien2.cern.ch/`

[2] Bagnasco S, Betev L, Buncic P, Carminati F, Cirstoiu C, Grigoras C, Hayrapetyan A A, Harutyunyan A, Peter A J and Saiz P 2003 Alien: Alice environment on the grid *Journal of Physics: Conference Series, Volume 119, Part 6*

[3] The ALICE Collaboration 2005 *ALICE: Technical Design Report of the Computing* ISBN 92-9083-247-9

[4] *ALICE Collaboration* URL `http://aliceinfo.cern.ch/Collaboration/`

[5] Buncic P, Peters A J and Saiz P 2003 Alienfs - a linux file system for the alien grid services *Computing Research Repository C0303241:THAT005, e-Print Archive: cs.dc/0306071*

[6] Dorigo A, Elmer P, Furano F and Hanushevsky A 2005 Xrootd/txnetfile: a highly scalable architecture for data access in the root environment *Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics* (World Scientific and Engineering Academy and Society (WSEAS))

[7] Feichtinger D and Peters A J 2005 Authorization of data access in distributed storage systems *The 6th IEEE/ACM International Workshop on Grid Computing*

[8] Grigoras C, Betev L, Saiz P and Schreiner S 2010 Optimization of grid resources utilization: Qos-aware client to storage connection in alien *Journal of Physics: Conference Series*

[9] Welch V, Barton T, Keahey K and Siebenlist F 2005 Attributes, anonymity, and access: Shibboleth and globus. integration to facilitate grid collaboration *4th Annual PKI R&D Workshop*

[10] Zhu J, Saiz P, Carminati F, Betev L, Zhou D, Lorenzo P M, Grigoras A G, Grigoras C, Furano F, Schreiner S, Datskova O V, Banerjee S S and Zhang G 2011 Enhancing the alien web service authentication *Proceedings of the 18th Int. Conference on Computing in High Energy and Nuclear Physics (CHEP 2010)*