PROCEEDINGS OF SCIENCE

A security architecture for the ALICE Grid Services

Steffen Schreiner[*ab], **Costin Grigoras**[b], **Alina Grigoras**[b], **Latchezar Betev**[b], **and Johannes Buchmann**[ac]

[a] *CASED - Center for Advanced Security Research Darmstadt, Mornewegstrasse 32, 64293 Darmstadt, Germany*

[b] *CERN - European Organization for Nuclear Research, CH-1211 Genève 23, Switzerland*

[c] *Technische Universität Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany*

E-mail: steffen.schreiner@cased.de

Globally distributed research cyberinfrastructures, like the ALICE Grid Services, need to provide traceability and accountability of operations and internal interactions. This document presents a new security architecture for the ALICE Grid Services, allowing to establish non-repudiation with respect to creatorship and ownership of Grid files and jobs. It is based on mutually authenticated and encrypted communication using X.509 Public Key Infrastructure and the Transport Layer Security (TLS) protocol. Introducing certified Grid file entries and signed Grid jobs by implementing a model of *Mediated Definite Delegation* it allows to establish long-term accountability concerning Grid jobs and files. Initial submissions as well as any alteration of Grid jobs are becoming verifiable and can be traced back to the originator. The architecture has been implemented as a prototype along with the development of a new central Grid middleware, called jAliEn.

The International Symposium on Grids and Clouds (ISGC) 2012,
February 26 - March 2, 2012
Academia Sinica, Taipei, Taiwan

[*] Speaker.

PoS(ISGC 2012)027

1. Introduction

The ALICE ("A Large Ion Collider Experiment") Grid Services [1, 2], a distributed storage and computation Grid framework, are developed and operated by the ALICE Collaboration [3] as a global research cyberinfrastructure. It provides the computing environment for simulation, reconstruction and analysis of the physics data collected by the ALICE detector at CERN, one of the four large experiments within the Large Hadron Collider (LHC). The Virtual Organization (VO) [4] represented by ALICE is currently composed of more than 70 computing centres (hereafter referred to as sites) located in over 30 countries, combining up to 45k CPU cores and 50PB of storage and serving approximately 1000 users within the international ALICE Collaboration.
The ALICE Grid Services are based on a central workload management system and a file catalogue established by the open source Grid middleware AliEn [5] ("ALICE Environment") [6]. From a technical perspective, users are free to provide and execute arbitrary data and program code while being allowed to utilize a multitude of facilities from organizations all over the world. Accordingly, accountability and traceability are key security concerns. This document presents a new Grid security architecture, pursuing to achieve a high level of accountability and traceability of the interactions within a eScience Grid, while instantiating non-repudiation of creatorship and ownership of Grid files and jobs. Our approach is based on enhancements of signed storage tickets [7] and the model of *Mediated Definite Delegation* [8]. Along with the development of a new central Grid middleware for the ALICE Grid services, called jAliEn [9], the security architecture was implemented as a prototype and proof of concept.
The remainder of this document is structured as follows: Section 2 provides a brief overview on related work. Section 3 summarizes an analysis of the ALICE Grid Services and the ALICE VO and specifies security requirements. Throughout section 4, the key aspects of the new security architecture are presented. Finally, section 5 concludes with a short discussion.

2. Related Work

The Globus Security Infrastructure (GSI) based on X.509 proxy certificates [10] [11] allowing for dynamic delegation implies major security limitations to a Grid infrastructure. Delegation restrictions in terms of concerned delegatees or other context-sensitive qualifications of a delegation are difficult to establish. Unrestricted X.509 proxy certificates even allow for full impersonation of delegators while not providing any other controls and can thereby render authentication and authorization meaningless[8].
Benedyczak et al. [12] describe the security model of the UNICORE 6 Grid framework based on assertions. Using only standard X.509 Public Key Infrastructure [13] and the Transport Layer Security (TLS) [14] protocol it allows to create chains of signed assertions, whereby non-repudiation can be achieved concerning the assignment of an assertion. Nevertheless an assertion depends on logical references, binding e.g. a delegation to the ID of a certain Grid job. The relevance of an assertion relies therefore on the correctness of the logical reference.
In [15, 16] a security framework based on signed task descriptions, so called *Signed ClassAds*, for the Condor distributed batch computing system is presented. A *Signed ClassAd* is placed inside a X.509 proxy certificate as a policy information together with so called *Action authorization ex-*

pressions. These are rules expressing which entity is allowed to use the proxy credential for what purposes. The mechanism allows e.g. to specify file checksum as conditions for executables and input files of a task. There is though no explanation how to establish a dynamic delegation or allow for derivatives of a *Signed ClassAd* and the framework seems to presume all conditions to be expressed explicitly upon the initial submission.

3. Security analysis and requirements

The ALICE VO is based on the interactions of three categories of players (see figure 1): First, the VO's central management, operating the so called *central services*. Second, the *sites*, operated by research institutions as a contribution to the collaboration, providing the storage and computing resources. Third, the VO's users, acting as clients to the Grid services while connecting to it from various systems including their personal computers or mobile devices. Central management, users and sites represent together one organizational entity or group of interest as the VO, though they describe different entities with respect to domains of control from a technical perspective (see [17]). The ALICE VO is embedded in an environment consisting of international Grid organi-

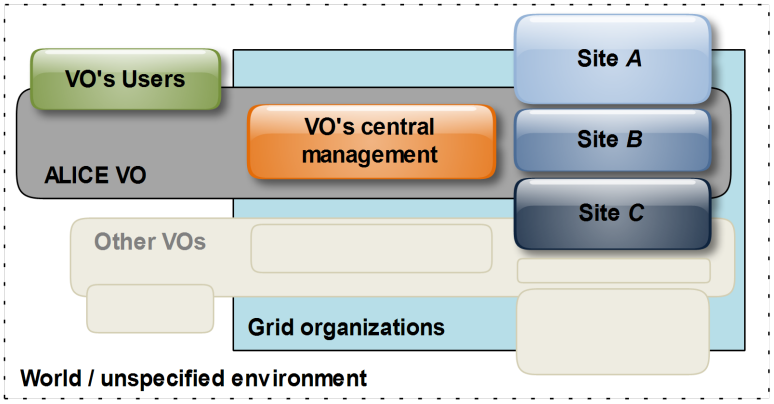

Figure 1: ALICE VO, environment, entities and relations

zations, most importantly the Worldwide LHC Computing Grid (WLCG) [18] which spans more than a dozen VOs from different areas of science. Sites can be part of a Grid organization or share their resources with the ALICE VO directly and provide access to their computing resources by an entry point, a so called *VOBox*. The actual computing facilities are so called *Worker Nodes* (WNs), accessible to the VOBox through a Computing Element, which acts as an interface to a Local Resource Management System e.g. a batch system. Further, sites provide disc and tape storage devices aggregated as so called *Storage Elements* (SEs), which compose the physical storage subsystem for the central Grid file catalogue.
From a VO's perspective, a VOBox can be seen as a virtual aggregation or pool of computing resources consisting of a site's WNs. The physical control of site systems as well as local scheduling decisions remain at the site level and a site is free to share its facilities also with other parties (see figure 1). Moreover, a site's infrastructure is masked by so called *Pilot Jobs*. Rather than submitting

a Grid job directly to a site, a Grid framework's *Pilot Job* is submitted beforehand, which turns into a service agent once executed on a WN. A *Pilot Job* receives a fraction of a WN's computational capabilities for a limited time, establishes a connection to its Grid framework and advertises the provided capabilities. Due to the *Pilot Job* model, previously unknown WNs are dynamically integrated into a virtual Grid layer for the lifetime of a corresponding *Pilot Job*. Figure 2 illustrates the trust relations of this Grid layer. Thereby, a user's system and a site's WN are the two most distant entities in terms of trust within the ALICE Grid Services, as well the two least trusted entities. The former is illustrated in horizontal distance, the latter by the size of the circles.

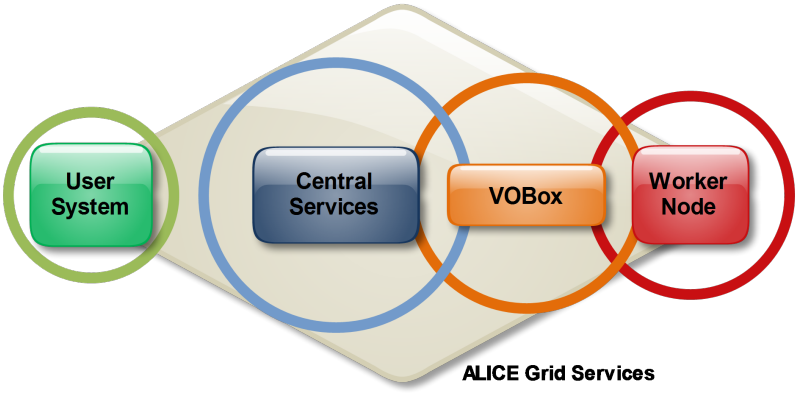

Figure 2: Trust relations in the ALICE Grid Services, illustrated in circles.

The data processed within the ALICE Grid Services origins from both the ALICE detector and as well as the users, e.g. due to simulations. The analysis software is provided centrally as software packages using an automated installation of the packages on the WNs. Users are though allowed to upload any files into the Grid file catalogue and use them both as data and executable code.
The framework's security characteristics are summarized as follows.

Security characteristics of the ALICE Grid Services

C1 Distributed services hosted in different domains of control which are communicating within a public and insecure environment, the Internet.

C2 Computation based on non-exclusive or shared resources on sites.

- *C2.1* Sharing follows a site's decisions and scheduling.
- *C2.2* Sites have full physical access to their resources.

C3 Uncontrolled client environment, e.g users' personal devices.

C4 The framework connects users and computing sites as legal entities from dozens of countries.

C5 Multiple origins of (including user-supplied) data and source and program code.

In particular with respect to characteristic *C1* the access to all Grid services must be authenticated and authorized, their integrity and availability must be ensured and breaches or errors need to be traceable. Data stored and processed within the ALICE Grid services are understood to be non-confidential as such due to the experiments character of public research. In favor of I/O performance it is therefore tolerated to send the data through unencrypted channels over the Internet. Since SEs as the storage subsystem are unaware of Grid users, authorized users must be provided with access credentials [19, 7] upon request. Accordingly, we define the following security requirements.

Security requirements for Grid service communication

R1 **Authentication:** Mutually authenticated communication parties

R2 **Integrity:** Assured integrity control

R3 **Confidentiality:** Possible confidential transmission of credentials

The ALICE Grid Services are based on a central task queue using textual Grid job descriptions, so called JDL (Job Description Language) strings. After submission, central services validate and preprocess a job, which produces derivatives of the initial submission and can result e.g. in the splitting of an initial job into sub jobs. All jobs are executed on WNs of sites using the VO's identity and the job accounting is based on a VO internal user name appended to a job's JDL string. As such, there is no actual delegation mechanism in place and user's are held accountable for any consequences of their initial submission. Since the transformations and their origin of an initially submitted Grid job are not verifiable, a user has to fully trust a VO without any further controls.
The *Grid Security Traceability and Logging Policy* [20], one of the regulating policies of the operations within the WLCG, declares the following traceability requirement: *"The minimum level of traceability for Grid usage is to be able to identify the source of all actions [. . .] and the individual who initiated them."* According to the policy and as a response to the characteristics *C1 - C6* we define three more requirements.

Security requirements for Grid entities

R4 **Verifiable Grid files:** Certified identity and originator of a Grid file entry, providing non-repudiation

R5 **Verifiable origin of Grid job submission:** Certified Grid job submission, including originator, providing non-repudiation

R6 **Verifiable processing of Grid job submission:** Certified Grid job processing, modification and delegation, including originator, providing non-repudiation

4. A new security architecture

A security architecture was developed and implemented as a prototype in line with the development of a new Java based central Grid middleware for the ALICE Grid services, called jAliEn.

The following sections describe different components as the key aspects of the architecture with respect to the defined security requirements *R1 - R6*.

4.1 Mutual authentication and secure communication

The system's communication is established via a mutually authenticated and encrypted connection. This connection is negotiated between a central service, called jCentral, and all its clients, respectively both users and sites use the same connection schema and address. The communication is based on the Transport Layer Security (TLS) [14] protocol and X.509 client and server authentication using the Bouncy Castle Crypto API [21], satisfying the security requirements *R1 - R3*. While users authenticate using VO acknowledged X.509 user certificates, standard X.509 host certificates are utilized for the authentication of central services and the sites. After a successful authentication of a client connection, the Distinguished Name of the corresponding client certificate is mapped to an internal Grid user name by a central directory service. This Grid user name is used for later authorization of a request. Accordingly, a site's VOBox is mapped to a site entry based on its host name after authentication.

4.2 Site resource aggregation and request forwarding

The connection of a site and the jCentral service is established by a service on the site's VOBox called *jSite*, which acts as a communication router and as such virtually aggregates a site's computing resources, respectively WNs. Along the *Pilot Job* model, the *jSite* service advertises idle resources to *jCentral*. Upon request it submits so called *jAgents* as *Pilot Jobs* to a local Computing Element. For every *Pilot Job* submission the *jSite* service generates a private key and X.509 certificate, places the private key into the *Pilot Job* and stores the certificate to be able to authenticate later on the according *jAgent* instance. Once a *jAgent* is started on a WN as a service it uses the private key to establish a persistent connection to its *jSite* service as described above and requests Grid jobs for execution. Accordingly, *jCentral* has only direct connections to the *jSite* instances of each of its sites while all communications with the *jAgent* instances running on WNs are aggregated.

4.3 A local string query service over serialized Java object streams

The service offered via the described connection by the central service *jCentral* allows to execute serialized Java object code, as such e.g. to request the Java object of a Grid file or job by a string identifier. The protocol is limited to exchange only objects of Java classes known to both communicating peers beforehand, while only object values and no class definitions or code is exchanged. Therefore it is possible to strictly enforce authorization using fine-grained controls on the access of objects and their attributes. The approach is based on a Java class with defined local and remote execution methods, a so called *Request*. Another Java class, a so called *Dispatcher* serializes and sends an object to the remote entity once a remote method is called locally. On the remote entity the object is deserialized and the method is executed, while typically changing the state of the object itself as well as the state of the remote entity. Consecutively, it is serialized again and sent back to the caller. There, the object replaces its original version and represents its final state after the execution of the requested methods. The functionality is fully encapsulated into the respective *Request* and *Dispatcher* classes and is implicitly invoked over inheritance of any class

extending *Request*. The operations provided as such *Requests* are e.g. to get, remove or delete Grid file entries from the Grid file catalogue by specified file names, to get a Grid job by its job number or to submit a new Grid Job.

4.4 Grid user interfaces

The ALICE Grid Services are required to provide two user interfaces: First, a command-line interface (CLI) used both manually by users and within plugins and script engines. Second, a connection library for the ROOT [22] and AliRoot [23] frameworks utilized by users locally as well as within Grid jobs on the WNs. In order to provide a user CLI and to enable the use of the ROOT library together with the described Java object serialization protocol, a local client service called *jBox* was developed (see figure 3): A *jBox* instance is started and dispatched as a process daemon

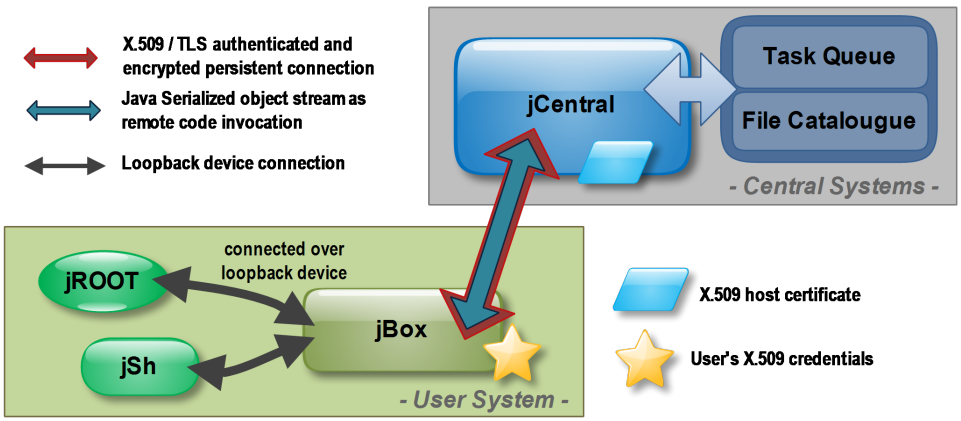

Figure 3: Client connection schema

after a user's credentials, private key and X.509 certificate, are loaded. The credentials remain in a protected Java *Keystore* for the lifetime of the *jBox* process, allowing for a single sign-on behavior of the system. Once initialized, *jBox* immediately opens a persistent TLS connection to the VO's *jCentral* service, whereas the connection is reestablished automatically whenever interrupted. After successful authentication and authorization at the *jCentral* service, a local CLI service is started, which listens on the operating system's loopback interface. The service's address and a randomly generated string token for local authentication is placed in a specified temporary file protected from other system user's access. An adapted version of the ROOT library called *jRoot* and a newly developed user CLI client called *jSh* are then able to connect to the local *jBox* service while using the token previously placed in the temporary file for authentication.

4.5 Certified Grid files

In [7], a mechanism is presented to ensure authentic file information and consistent meta information in the Grid file catalogue with respect to the physical files on SEs as the storage devices. For both read and write accesses, a client receives a signed *Access Ticket* from a central service after successful authorization and is then able to use the ticket to access a SE and perform the operation.

In case of a successful write operation, the client requests a signed *Status Ticket* (see figure 4) from the SE, containing at least the file's location on the storage and its checksum and size, which allows to prove a SE's acknowledgement of a file to central services. The adoption of the *Status Ticket* as a mandatory requirement for file registration allows to ensure consistent information on the files listed in the Grid file catalogue with respect to the physical storage level.
A remaining problem regarding accountability and non-repudiation of a file's ownership is the necessary full trust in the Grid file catalogue. Consequently, being able to arbitrarily change Grid file catalogue information will empower to forge file meta data, including any file's identity and ownership without trace. Introducing another digitally signed ticket called the *File Certificate*, that is created and signed by the client upon registration and kept on a long term basis, this can be mitigated (see figure 4). The *File Certificate* contains the logical file name as listed in the Grid

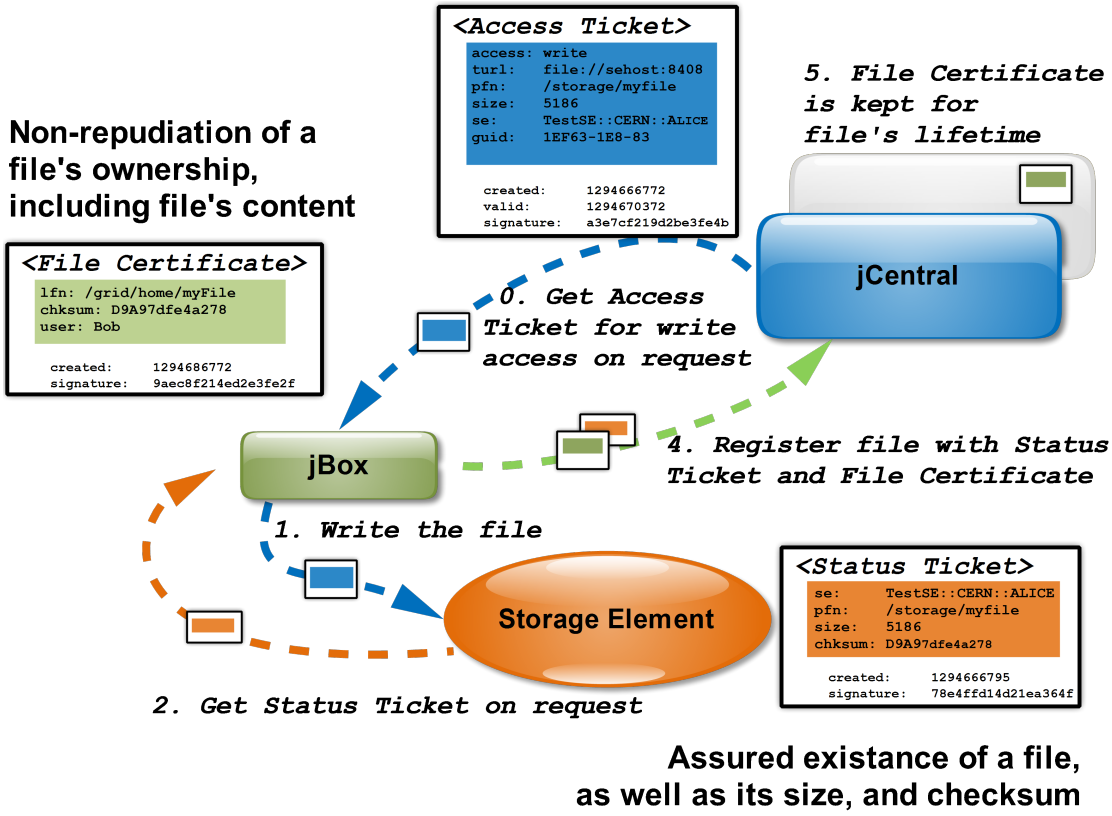

Figure 4: Certified Grid files

file catalogue, the file's checksum, user name and time-stamp and is sent together with the corresponding user's X.509 certificate and the according *Status Ticket* to central services in order to register the file. In case of successful ticket verifications and consecutive file registration, the *File Certificate* is stored in a designated database next to the Grid file catalogue, as well as the user's certificate if not available already. Kept for the lifetime of the file, the *File Certificate* proves the owner's accountability for the file and its content. As it is thereby possible to identify any illegitimate alteration of the file's meta information in the Grid file catalogue, the mechanism provides non-repudiation and thereby fulfills security requirement *R4*.

4.6 Certified Grid jobs with Mediated Definite Delegation

In [8] a model called *Mediated Definite Delegation* is specified to allow for certified Grid jobs. The model's approach is to require the client to sign the JDL of a later Grid job before its submission, while only relying on X.509 certificates and abandoning any usage of X.509 proxy certificates. Every broker processing the job thereafter appends information to the signed JDL and signs it another time. This model was implemented as follows: Upon Grid job submission a user's local *jBox* service digitally signs a job's JDL string using the user's private key before the created Grid job object is sent together with the user's public X.509 certificate to the *jCentral* service. There, the user's signature is verified before the job is placed into the central task queue. If not available already, the user's certificate is stored in a central certificate database. Before a Grid job is sent out to a site for execution the user-signed JDL is signed another time by *jCentral*, using the VO's central private key, and the Grid job object is sent together with the user's X.509 certificate to the site's *jSite* service. The VO's central private key corresponds to a X.509 certificate that is known beforehand and trusted by all VO entities.
After its startup as a *Pilot Job*, a *jAgent* on a WN is persistently connected to its corresponding *jSite* service as described above (see figure 5). A *jAgent* instance is implicitly authorized to retrieve Grid jobs matching its capabilities through the site based authentication of its *jSite* service. Once a *jAgent* instance receives a Grid job object from *jCentral* trough its request to *jSite* it verifies all signatures of the corresponding JDL. Upon successful verification, necessary Grid files are downloaded to the WN using the signed JDL as a reference for authorization and the Grid job is executed. The authentication of a Grid file access request is based on the combination of the authenticated *jSite* connection to *jCentral* and the corresponding job identification. The authorization is based on a legitimation of the file access request within the JDL, e.g. as an explicit input or output file statement. In order to enable jobs to directly access Grid files a local *jBox* service is started by the

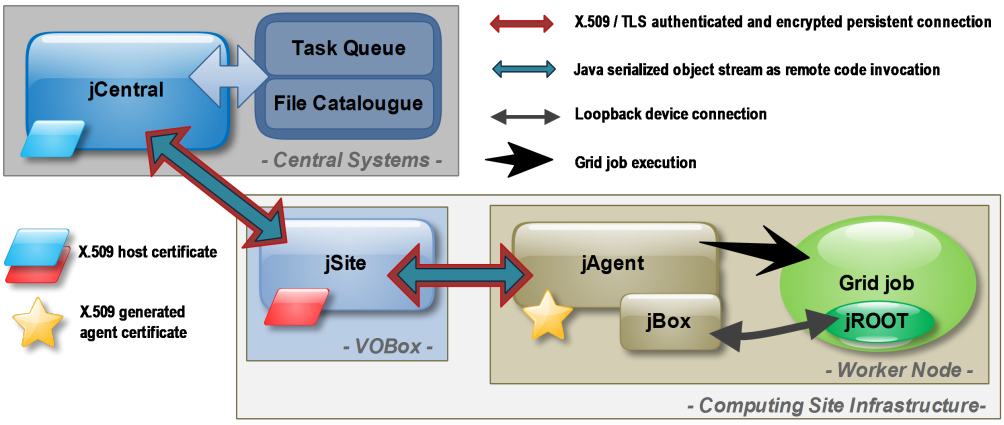

Figure 5: jSite, *jAgent* and *jBox* on a Worker Node

jAgent. In contrast to a user instance of the *jBox* service, a local authentication token is created for every Grid job a *jAgent* is processing. By providing each Grid job with an appropriate local token it is possible to transparently handle requests from different jobs concurrently. Forwarded by the *jAgent Pilot Job* as *Requests*, every Grid file access is authenticated and authorized based on a

legitimation in the JDL as explained above.
The implementation of the model of *Mediated Definite Delegation* allows to prove a user's Grid job delegation at a WN without requiring further information or callbacks and provides an actual warrant of the requested delegation. Furthermore, a user's X.509 certificate corresponding to a JDL's first signature provides a verifiable attestation of the delegators identity. As such, the mechanism is able to satisfy security requirement *R5*.
Concerning the registration of files in the Grid file catalogue by a Grid job, neither the job nor the corresponding *jAgent* instance have a recognized X.509 certificate at their disposal. In order to allow for certified Grid files, the *File Certificate* is created by the *jAgent* and signed by the *jSite* service after a successful write operation. The site as the delegatee of a user for a given Grid job assures thereby its responsibility for the files created due to the job's execution. The *File Certificate* and the site's X.509 certificate are stored centrally as specified above.
Remaining issues with respect to security requirement *R6* are a Grid job's access to the *jAgent's* process on a WN and the concurrent or consecutive execution of several Grid jobs by one *jAgent* instance. In practice, a Grid job can tamper arbitrarily with its corresponding *jAgent* or another Grid job running with the same local user account on the WN's operating system (see figure 6). These problems can be specified as missing *protection* of the *Pilot Job* and missing *isolation* of Grid jobs [8] and can be mitigated by introducing a identity-switching module, like e.g. gLExec [24] or a virtualization mechanism. The decoupling of Grid jobs from their corresponding *jAgent* instances

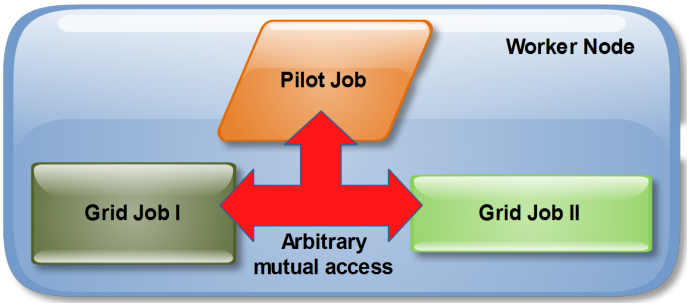

Figure 6: Pilot Job and Grid job processes running on the same local user account on a WN.

in the jAliEn prototype, while using authenticated connections over the loopback interface, is fully prepared for such mechanisms. Presuming though, the identity-switching mechanism to be able to authenticate a Grid job based on its signed JDL and the corresponding X.509 certificate. Under these conditions, a fulfillment of security requirement *R6* can be achieved.

5. Conclusion

The presented security architecture is based on mutually authenticated and encrypted communication using X.509 Public Key Infrastructure and the TLS protocol. The introduction of *File Certificates* in the Grid file catalogue minimizes the necessary trust in the file catalogue and prevents forgery of ownership. The adoption of the model of *Mediated Definite Delegation* allows to establish certified Grid jobs and thereby to protect a user's identity and privileges along the Grid

job processing. The initial submission as well as any alteration of a Grid job throughout its lifetime is becoming verifiable and can be traced back to its originator. Following the trust relations of the three players of the VO's central management, its users and its sites, the presented Grid security architecture allows to establish long-term accountability due to non-repudiation of creator- and ownership of Grid jobs and files.

References

[1] P. Cortese et al., *ALICE computing: Technical Design Report*, *Technical Report ALICE-TDR-012 ; CERN-LHCC-2005-018*, CERN, 2005.

[2] *ALICE Grid Monitoring with MonALISA*, `http://pcalimonitor.cern.ch/` .

[3] *ALICE Collaboration*, `http://aliceinfo.cern.ch/` .

[4] I. Foster, C. Kesselman and S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, *International Journal of High Performance Computing Applications* 15(3), 2001.

[5] *MonALISA Repository for ALICE*, `http://alien.cern.ch/` .

[6] S. Bagnasco et al., *AliEn: ALICE environment on the GRID*, *Journal of Physics: Conference Series* 119, 2008.

[7] S. Schreiner et al., *Securing the AliEn File Catalogue - Enforcing authorization with accountable file operations*, *Journal of Physics: Conference Series* 331(6), 2011.

[8] S. Schreiner, L. Betev, C. Grigoras and M. Litmaath, *A Mediated Definite Delegation Model allowing for Certified Grid Job Submission*, *CoRR* 2011, [`abs/1112.2444`].

[9] *jAliEn*, `http://jalien.cern.ch/` .

[10] V. Welch et al., *X.509 proxy certificates for dynamic delegation*, in proceedings of the *3rd Annual PKI R&D Workshop*, 2004.

[11] S. Tuecke et al., *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*, RFC 3820 (Proposed Standard), 2004.

[12] K. Benedyczak et al., *Key aspects of the UNICORE 6 security model*, *Future Generation Computer Systems* 27(2), 2011.

[13] D. Cooper et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 5280 (Proposed Standard), 2008.

[14] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246 (Proposed Standard), 2008.

[15] I. D. Alderman and M. Livny, *Task-specific restricted delegation*, in proceedings of the *16th International Symposium on High Performance Distributed Computing*, HPDC '07, 2007.

[16] I. D. Alderman, *A Security Framework for Distributed Batch Computing*, Ph.D. thesis, University of Wisconsin-Madison, 2010.

[17] I. Foster, *What is the Grid? - a three point checklist*, *GRIDtoday* 1(6), 2002.

[18] *Worldwide LHC Computing Grid (WLCG)*, `http://lcg.web.cern.ch/lcg/` .

[19] D. Feichtinger and A. J. Peters, *Authorization of Data Access in Distributed Storage Systems*, in *6th IEEE/ACM International Workshop on Grid Computing*, Grid'05, 2005.

[20] Joint Security Policy Group, *Grid Security Traceability and Logging Policy*, *Technical Report CERN-EDMS-428037*, LCG EGEE Joint Security Policy Group, 2008.

[21] The Legion of the Bouncy Castle, *Bouncy Castle API*, `http://www.bouncycastle.org/` .

[22] *ROOT*, `http://root.cern.ch/` .

[23] *AliRoot*, `http://aliceinfo.cern.ch/offline/` .

[24] D. Groep, O. Koeroo and G. Venekamp, *gLExec: gluing grid computing to the Unix world*, *Journal of Physics: Conference Series* 119, 2008.