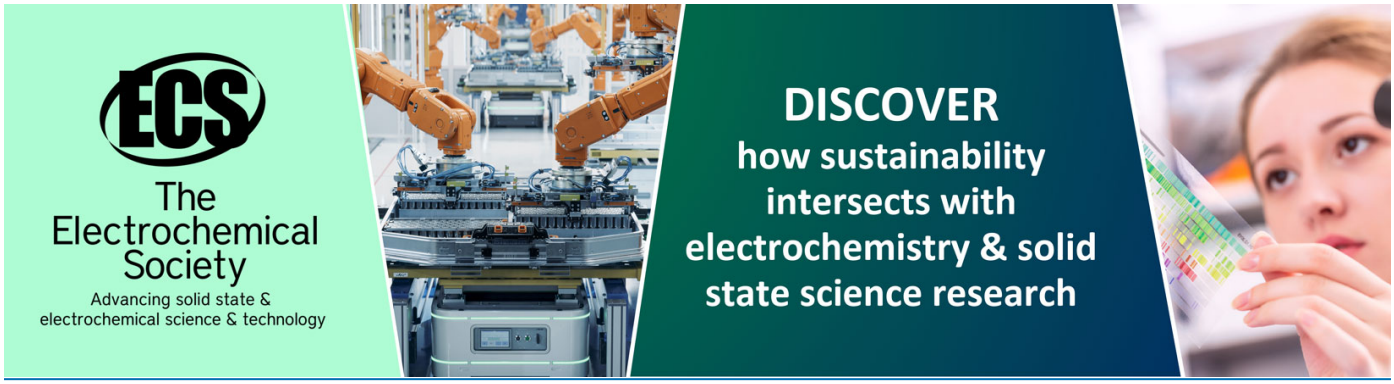
Mediated definite delegation - Certified Grid jobs in ALICE and beyond



Journal of Physics: Conference Series **396** (2012) 032096 doi:10.1088/1742-6596/396/3/032096

Mediated definite delegation - Certified Grid jobs in ALICE and beyond

Steffen Schreiner[1,2,3], Costin Grigoras[2], Maarten Litmaath[2], Latchezar Betev[2] and Johannes Buchmann[1,3]

[1] CASED - Center for Advanced Security Research Darmstadt
Mornewegstrasse 32, 64293 Darmstadt, Germany
[2] CERN - European Organization for Nuclear Research
CH-1211 Genève 23, Switzerland
[3] Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany

E-mail: steffen.schreiner@cased.de

Abstract. Grid computing infrastructures need to provide traceability and accounting of their users' activity and protection against misuse and privilege escalation, where the delegation of privileges in the course of a job submission is a key concern. This work describes an improved handling of Multi-user Grid Jobs in the ALICE Grid Services.
A security analysis of the ALICE Grid job model is presented with derived security objectives, followed by a discussion of existing approaches of unrestricted delegation based on X.509 proxy certificates and the Grid middleware gLExec. Unrestricted delegation has severe security consequences and limitations, most importantly allowing for identity theft and forgery of jobs and data. These limitations are discussed and formulated, both in general and with respect to an adoption in line with Multi-user Grid Jobs. A new general model of mediated definite delegation is developed, allowing a broker to dynamically process and assign Grid jobs to agents while providing strong accountability and long-term traceability. A prototype implementation allowing for fully certified Grid jobs is presented as well as a potential interaction with gLExec. The achieved improvements regarding system security, malicious job exploitation, identity protection, and accountability are emphasized, including a discussion of non-repudiation in the face of malicious Grid jobs.

1. Introduction

Global eScience Grid environments provide researchers with unified access to computing and storage services across national borders, jurisdictions, and domains of responsibility. Accordingly, and beyond the existence of operational and usage policies, accountability needs to be ensured for actions occurring due to the operation of such infrastructures and in the course of its users' activities. Protection against misuse and privilege escalation needs to be established and any violations need to be traceable. These baseline security concerns form the general background and motivation of this work.
The ALICE Grid Services [1], a globally distributed Storage and Computation Grid, are developed and operated by the ALICE Collaboration [2] as a research cyberinfrastructure. Its central workload management system and its File Catalogue are provided by the open source Grid framework AliEn [3, 4]. The system provides the environment for simulation,

reconstruction and analysis of physics data collected by the ALICE detector at CERN, one of the four large experiments within the Large Hadron Collider (LHC). As such, it is embedded within the WLCG [5, 6], a tiered infrastructure of Grid services for the large LHC experiments. The ALICE Grid Services constitute a Virtual Organization (VO) and are based more than 70 computing centres, hereafter referred to as *Sites*, located in over 30 countries, combining up to 50k CPU cores and 30 PB of storage, and serving almost 1000 users within the collaboration.
The ALICE File Catalogue establishes a central logical layer on top of a globally distributed set of storage servers provided by the Sites, which constitutes one virtual Grid file system [7]. The computation is based on Worker Nodes (WNs) aggregated on Sites within batch farms, receiving Grid jobs from a central task queue. Grid jobs are specified and represented by a textual description, listing e.g. the job reference number, the submitter's user name, the file to be executed, execution arguments, and input and output files. Users are free to deliberately upload program code and data to the system and request it to be executed as Grid jobs on WNs. This freedom within a globally distributed cyberinfrastructure creates security challenges, in particular with respect to accountability and liability.
In the remainder of this introduction (section 1.1), user credentials, Pilot Jobs and implicit delegation of privileges in the ALICE Grid Service are explained. Chapter 2 presents a security analysis of the ALICE Single-User Pilot Jobs approach and identifies according security objectives. Within chapter 3, two existing approaches of Multi-User Pilot Jobs with gLExec based on X.509 proxy credentials are shortly described. General limitations of proxy credentials are identified (section 3.1), followed by a discussion of problems arising from proxy credentials in Multi-User Pilot Job scenarios (section 3.2). In chapter 4, a new model of delegation is defined and explained, potential further extensions are demonstrated (section 4.1) followed by the illustration of an existing prototype implementation (section 4.2) and the model is discussed with respect to the security objectives (section 4.3). Finally, the security objectives and the model are reconsidered with respect to related work (section 5).

1.1. Pilot Jobs in the ALICE Grid Services

Throughout the WLCG, X.509 certificates [8] are used as an authentication mechanism for Grid users and operators, e.g. via the Globus Toolkit [9, 10]. Such a user certificate is hereafter referred to as a *Grid Certificate*. Based on the concept of X.509 proxy certificates [11, 12], proxy credentials are used in order to allow for delegation and single sign-on [13]. An X.509 proxy credential, hereafter denoted as *PrxCrd*, consists of a private/public key pair and a Grid Certificate, where the public key is signed with the corresponding private key.
In the ALICE Grid Services user PrxCrds are only utilized upon client logon for authentication and once a PrxCrd is validated its Grid Certificate's *Subject* entry is mapped to an ALICE internal user name. Throughout the system users are only represented and authorized by their user names. There is no actual mechanism for delegation of privileges in place and Grid jobs as well as any other interactions with the system are simply associated to the internal user names.
Pilot jobs are a mechanism in Grid environments to optimize resource utilization by establishing a virtual layer on top of a Site's resources. The basic principle is to request a WN to run a service instead of a job or payload and to let this service handle the execution of actual Grid jobs later on.
The ALICE Pilot Job uses a PrxCrd in order to authenticate itself to the VO's central services. This PrxCrd, hereafter referred to as *Pilot PrxCrd*, represents a Site or a Grid operator and is submitted to a WN by a Site service in combination with the Pilot Job request. Once placed on the Site service, it is renewed automatically by periodical requests to a MyProxy [14] service. MyProxy is a credential management service that holds long-term valid PrxCrds (e.g. weeks) and provides authorized entities on demand with derived PrxCrds [15] with a shorter time of validity (e.g. one day). As the ALICE Pilot Job executes Grid jobs directly, all jobs are executed

on WNs using the same local user account as the respective Pilot Job and accounting is based only on the job submitter's user name. As both the Pilot Job and all Grid jobs are executed by the person identified by the Pilot PrxCrd, this approach is hereinafter referred to as *Single-User Pilot Jobs*.

2. Security Analysis of Single-User Pilot Jobs in ALICE

The ALICE VO can be considered as a service provider and as an intermediary between its users and the Sites as platform providers. As this intermediary the VO receives Grid jobs as *task assignments* from its users, decides which Grid jobs are to be executed by which Site and propagates the jobs to Pilot Jobs on WNs. Accordingly a Grid user performs an implicit *delegation* of privileges to a VO and thereby to a respective Site and its WN.

Supplying software packages, which are installed on demand on a WN by the Pilot Job, the ALICE Grid Services provide a Software as a Service (SaaS) layer to its users, itself based on the Platform as a Service (PaaS) based usage of its Sites' WNs. The PaaS layer can as well be used by users directly, though, as any program code and data can be deliberately requested to be executed within a Grid job. In doing so, users are merely obliged by policy to utilize the infrastructure only in line with their research in the ALICE experiment. The program code and data executed along with a Grid job can be classified according to four internal and one external origin, as specified in table 1.

Once a user has submitted a job to the Grid, the job as a task assignment is completely in the

Internal I: File Catalogue entries	Raw and processed data of the ALICE detector or any data or program code supplied by users.
Internal II: Software Packages	Program code downloaded and supplied to the job by the Pilot Job, provided by the VO.
Internal III: Middleware	Program code provided by the VO and third parties.
Internal IV: Worker Node	Program code provided within the operating system of the WN, e.g. system commands and libraries.
External	Any data and program code retrieved within the job from external resources, e.g. from the Internet.

Table 1. Grid job data and program code origins in the ALICE Grid Services

sovereignty of the VO central management system. As the relation between a Grid user and a job is provided by the internal user name, this relation is controlled within the VO. Similarly, the central management system is able to deliberately alter a user's submission, which is processed and can e.g. be split into sub-jobs. On a WN there is no assurance of a correct execution of a job beyond simple run time and resource utilization monitoring. While the administrative sovereignty of the WN resides at the Site, the sovereignty of the Pilot Job program code is with the VO. Consequently, the approach of Single-User Pilot Jobs has drastic limitations to security and user accountability, as it may virtually be impossible to state the origin of potential security incidents, attacks or misbehaviour arising along or from a Grid job execution. Accordingly we define four security objectives as necessary criteria for accountability and non-repudiation of both a Grid user's job submission and the job's processing within the VO's control.

Objective 1, *Provable authenticity of assignment*: The original submission of a Grid job must be verifiable at any later stage, including the submitter's identity.

Objective 2, *Provable authenticity of assignment processing*: The processing of a Grid job as an assignment must be verifiable at any later stage, provably resulting from a set of sound transformations.

Objective 3, *Protection against forgery of assignment*: Forgery of Grid job assignments by Grid users, the VO, the Sites or any third party must be impeded.

Objective 4, *Protection against misuse of delegation*: The delegation of privileges along with a Grid user's job submission must be protected from being misused.

According to the Single-User Pilot Job model, Grid jobs are executed by the Pilot Job process and therefore run within the same local user environment on a WN. A Pilot Job runs only one job at a time and a job's working directory is scratched after the execution. Nevertheless, a job can fork sub-processes that will remain on the system after its execution and are able to tamper with jobs executed later on. Hence Grid jobs are not strictly isolated from each other. Further, Grid jobs are neither encapsulated nor isolated with respect to their Pilot Job and are able to alter the Pilot Job or get hold of the Pilot PrxCrd. This introduces a crucial security threat, as a Grid job might e.g. use the Pilot PrxCrd to deliberately submit new jobs or exploit the Pilot PrxCrd's escalated privileges. These escalated privileges entitle the holder e.g. to impersonate any user within the ALICE Grid Services in order to register File Catalogue entries in the name of the job submitter. Accordingly we further formulate the following security objectives:

Objective 5, *Grid job isolation*: Grid jobs should be mutually isolated and must be prevented from potential interference, both concurrently and consecutively in time.

Objective 6, *Pilot Job protection*: A Pilot Job, as well as its credentials, must be protected from alteration, interference or disruption by any of its Grid jobs.

Objective 7, *Pilot credential limitation*: Pilot Job credentials must be limited in power, not to allow any escalated privileges, in particular with regard to Grid users' identities.

Objective 8, *Pilot platform integrity*: A WN and its Pilot Jobs must provide an environment of integrity and be protected from any non-conforming Site access or access by third parties.

In the model being discussed, Grid jobs of different users are not visible to a WN's operating system, and thereby to Sites, in a transparent way and it is not possible to enforce a per-user Grid job control. In case of security incidents or attacks it is not possible to revoke access of single users and potential counter-measures can only affect a VO's entire set of jobs on a Site or WN. Pilot Job mechanisms allowing for such a transparent identification and control of Grid jobs on a WN based on the job submitter's identity are hereinafter referred to as *Multi-user Pilot Jobs*. Accordingly we formulate another objective:

Objective 9, *On-Site Grid job user accounting*: Grid jobs need to be authenticated and authorized in a transparent way on a WN.

Meaningful Grid job accountability would require both a submitter's responsibility to be proven for a certain malicious or illegal behaviour that was observed, as well as to ensure a submitter can rightfully disclaim responsibility, in case the submitter's actions were appropriate. All Grid job related data and program code must therefore be verifiable in order to prove their integrity and authorship. This would permit differentiating a user's submission from interferences by other

users or third parties, or by erroneous system behaviour (see e.g. objective 8). We therefore define one last objective:

Objective 10, *Non-repudiation of responsibility for Grid jobs*: The authenticity of all entities referred to within a Grid job submission and its processing must be verifiable at any later stage, including the originator's identity.

3. Multi-user Pilot Jobs based on X.509 proxy credentials

In order to allow for a secure handling of Multi-User Pilot Jobs, the Grid middleware gLExec [16] was developed. Instead of a direct execution of a job, gLExec is invoked by a Pilot Job in order to authenticate and authorize a job request based on a X.509 PrxCrd beforehand. Using gLExec, mutual Grid job isolation (objective 5) and Pilot Job protection (objective 6) can furthermore be achieved by a local user and environment switch on WN. This functionality is similar to the UNIX `sudo` command, while using mapping mechanisms to determine a local user account based on a PrxCrd's *Subject* entry. The application of gLExec relies on the propagation of a job submitter's PrxCrd to the Pilot Job on a WN. We outline therefore briefly two different approaches taken by the LHC experiments *ATLAS* [17] and *LHCb* [18]:
In [19, 20], an integration of gLExec into the ATLAS Grid middleware is specified, using a MyProxy service into which user PrxCrds are uploaded and protected with random keys. The keys are then propagated to and kept within the VO's central management system. The key is sent to a Pilot Job together with a corresponding Grid job and is then used by the Pilot Job to retrieve the corresponding user PrxCrd from the MyProxy service. Such an approach, in which a VO holds keys with a one-to-one relation to its users' PrxCrds, is hereafter referred to as an *indirect user PrxCrd propagation*.
An integration [21] of gLExec into the LHCb middlware is based on a direct storage of user PrxCrds within the VO's central management system. Accordingly a Pilot Job immediately receives a user's PrxCrd together with the corresponding Grid job. This approach of a storage within the VO's sovereignty and implicit transfer of user PrxCrds is hereafter referred to as a *direct user PrxCrd propagation*.

3.1. Security limitations of unrestricted Proxy Credentials

X.509 PrxCrds as used throughout the WLCG are based on unrestricted delegation. The mechanism has long-known cardinal security limitations [11], which have already been considered upon its definition [13]. A delegation provided by unrestricted PrxCrds has conceptually no dependencies other than in the dimension of time, given by a PrxCrd's validity. While disregarding potential auxiliary restriction mechanisms, we specify three essential limitations of PrxCrds:

Limitation 1, *Unconditional delegation*: A PrxCrd has neither any binding to a particular delegate nor any context-sensitivity of its usage. Any privilege is held as such and any limitation or binding would require additional external mechanisms.

Limitation 2, *Unrestricted delegation*: Except in time, a PrxCrd allows only for an unrestricted delegation to the delegate, which thereby holds all privileges of the delegator.

Limitation 3, *Exposure to theft*: A PrxCrd is by itself completely unprotected while being passed on within a distributed environment. Regarding this aspect it is comparable to a plain security token. Without additional protection a PrxCrd can be stolen at any of its locations and must be expected to be accessible at least by persons with administrator privileges. Moreover, a PrxCrd has a validity of typically several hours or days, which cannot be considered too little for a successful exploitation by potential attackers.

The Virtual Organization Membership Service (VOMS) [22] provides the functionality to add attributes to a PrxCrd: upon authentication with a valid PrxCrd it is possible to request a service to apply the values of additional attributes present in the PrxCrd. These attributes can define e.g. VO membership or roles and the mechanism prevents an attribute's value being altered once set. As such it can be utilized to limit or restrict the delegation of privileges, though the mechanism alone is not sufficient to fully condition a PrxCrd in a context- or delegate-sensitive manner (see limitation 1) without resulting in a trivial fully static delegation.

3.2. Security analysis of Multi-user Pilot Jobs based on X.509 proxy credentials
Without comprehensive additional mechanisms the described limitations of PrxCrds based on unrestricted delegation lead to fundamental weaknesses concerning user accountability. Their adoption as credentials in a Multi-user Pilot Job architecture introduces severe security threats, which are subsequently discussed as security problems:

Problem 1, *Unprovable correlation of assignment and delegation*: A PrxCrd on a WN does not have any binding to any actual Grid job. The availability or presence of a valid user PrxCrd is no binding statement to prove the authenticity of a job or its sound processing.

As a consequence the use of PrxCrds is not able to fulfill the requirements for accountability of users with respect to job submissions. PrxCrds could be potentially stolen, misused or mixed up without notice at various points between the user's job submission and a WN. Similarly a job's description or payload could be altered or exchanged. In [23] this concern was raised as the necessity to trust a VO to provide flawless correlations between PrxCrds and jobs.

Problem 2, *Fuzzy validity and expiration*: The validity of a PrxCrd is by itself independent of the validity or lifetime of a Grid job.

A PrxCrd must be assumed to be still valid once a corresponding Grid job has terminated. In case of an indirect user PrxCrd propagation using a MyProxy service the relation between PrxCrds and Grid jobs cannot be assumed to be bijective, viz. the same PrxCrd can and will be used for several Grid jobs, e.g. to reduce the credential management overhead. In any case a PrxCrd can be renewed until the corresponding first-order PrxCrd in the MyProxy service expires. Also the latter credential could be renewed periodically, which would lengthen the potential validity of all PrxCrds derived from it. In the worst case this could cause PrxCrds to be valid up to many months.

Problem 3, *Unlimited access of VO and Sites*: Even if PrxCrds are never stored or processed within a VO's services, as in case of the indirect user PrxCrd propagation, a user or attacker holding certain privileges within the VO must still be considered to be able to retrieve any active user's PrxCrd. Since e.g. a job submitter's PrxCrd must be readable to the Pilot Job on the WN, everybody with access to the Pilot Job would be able to retrieve PrxCrds.

Problem 4, *Challenge of storage*: For both the direct and the indirect user PrxCrd propagation the main storage entity of PrxCrds becomes a critical security concern. The storage must be instantiated and maintained securely and attacks on the storage must be considered severe security threats.

Problem 5, *Drawback of additional service invocation*: A scenario utilizing remote service callbacks, e.g. the indirect user PrxCrd propagation, introduces additional risks of reduced availability due to failures or attacks. Moreover, any additional invocation amounts to additional dependencies, additional load in matters of scalability and the introduction of delays.

As a consequence the application of gLExec based on the two discussed alternatives for the propagation of PrxCrds amounts to no change or improvement regarding the significance in accountability. In both the ALICE Single-user Pilot Jobs approach and the two Multi-user Pilot Jobs alternatives a VO is e.g. able to submit jobs to Sites in the name of a user with neither the user's nor the Site's notice. Consequently the presence of a user PrxCrd cannot be considered at all as proof or anchor for accountability and an implementation would not be able to fulfill any criteria reflected by the objectives 1-4, 7, 9 and 10.

4. Mediated definite delegation

The ALICE Grid Services as an eScience Grid infrastructure can be described by two fundamental types of entities visible to its users, characterized by their behaviour within the system and with respect to the overall state of the system. Any data or program code in the system represents an entity at rest, as any alteration of the content of these entities would change their identity. This implies e.g. to refer to any change in an existing file entry to result in a new file entry, while this characterization conforms with the definition of checksums.

Definition 4.1: Resources, like e.g. data and program code files and software packages are defined as *stationary entities*.

Grid Jobs can be seen as modifiers of stationary entities, as non-trivial jobs are based on data and program code provided in the system, while their results are stored in new data entries. As such they are active within the system, yet transient as their benefit or utility is established only by their successful termination. Another example of a transient entity would be a Transfer Request, which is a request delegated to an agent to relocate or replicate physical copies of a stationary entity.

Definition 4.2: Processes or tasks, like e.g. Grid jobs or Transfer Requests, operate on stationary entities and are defined as *transient entities*.

In order to allow for strong accountability and non-repudiation, the authorship of any influence on a system's state must be verifiable. These influences can be classified according to the two classes of stationary and transient entities. As such all interactions with a system resulting in an alteration regarding stationary or transient entities must be verifiable, including their authorship.
With respect to stationary entities we first define a mapping to express authorship. Let $C = \{c : \text{creator of a stationary entity}\}$, $I = \{i : \text{identity of a stationary entity as a checksum}\}$, $T = \{t : \text{time stamp}\}$ and $S = \{s : \text{statement of authorship}\}$, wherein each element declares the authorship by a creator $c \in C$ with respect to a stationary entity $i \in I$ issued at a moment in time $t \in T$.

Definition 4.3: A *declaration of authorship* concerning a stationary entity is expressed by the mapping

$$auth : C \times I \times T \to S \ . \tag{4.1}$$

The ALICE Grid Services are currently based upon on a three-tier architecture of service consumers (Grid users), a service broker and processor (VO), and back-end service or platform providers (Sites). Further, Grid jobs or Transfer Requests as transient entities are represented throughout their whole lifetime in an explicit textual description.
With respect to these requirements we define a new model of mediated delegation, stating delegations to operate on explicitly specified system entities, while permitting verifiable transformations and dynamic delegations to agents via a broker:

Let $U = \{u : \text{user able to submit transient entities}\}$, $O = \{o : \text{delegable operation}\}$, $E = \{e : \text{reference to a stationary entity}\}$, $A = \{a : \text{agent, able to execute transient entities}\}$, and $\dot{T} = \{\dot{t} : \text{time period of validity}\}$. Further let $D = \{d : \text{delegation}\}$, wherein each element describes the delegation of an operation $o \in O$ with respect to a set of stationary entities E by a user $u \in U$ to an agent $a \in A$ within a certain period of validity $\dot{t} \in \dot{T}$. And $\mathcal{P}(X)$ representing the power set of a certain set X.

Definition 4.4: A simple *definite delegation* can be expressed by the mapping

$$deleg : U \times O \times \mathcal{P}(E) \times A \times \dot{T} \to D \ . \tag{4.2}$$

Let $B = \{b : \text{a broker}\}$ and $\tilde{D} = \{\tilde{d} : \text{unmediated delegation}\}$ describing delegations of $o \in O$ in the name of $u \in U$ with respect to a set of stationary entities E, to be mediated by a broker $b \in B$ and $Z = \{z : \text{verifiable transformation}\}$ describing derivatives or transformations the broker can apply to a $\tilde{d} \in \tilde{D}$, all valid within a certain period of time $\dot{t} \in \dot{T}$.

Definition 4.5: An *unmediated delegation* as element of $\tilde{D}$ is defined as a mapping

$$udeleg : U \times O \times \mathcal{P}(E) \times B \times \dot{T} \to \tilde{D} \ . \tag{4.3}$$

Definition 4.6: A *mediation of delegation* is defined as the derivative or transformation of a $\tilde{d} \in \tilde{D}$ according to a set Z and the assignment to an agent $a \in A$, valid for a certain period of time $\dot{t} \in \dot{T}$, with the result being an element of $\bar{D} = \{\bar{d} : \text{mediated delegation}\}$. It is expressed by the mapping

$$med : \tilde{D} \times \mathcal{P}(Z) \times A \times \dot{T} \to \bar{D} \ . \tag{4.4}$$

A *mediated definite delegation* can be now expressed by the composition of the two mappings

$$deleg_{\text{med}}(u, o, E, b, \dot{t}_s, Z, a, \dot{t}_m) = med(\ udeleg(u, o, E, b, \dot{t}_s)\ , Z, a, \dot{t}_m) \tag{4.5}$$

The mapping $deleg_{\text{med}}$ then describes a mediated definite delegation of an operation with respect to a set of stationary entities, while allowing verifiable transformations and the dynamic election of an agent.

4.1. Mediated definite delegation with multiple brokers

The model can be further extended beyond the original requirements to allow multiple brokers. Let $W = \{w : \text{witness of authorship}\}$, wherein each element declares the responsibility of a broker $b \in B$ for a certain statement of authorship $s \in S$ or witness of authorship $w \in W$, issued at a moment in time $t \in T$.

Definition 4.7: A *witness of authorship* concerning the authorship of a stationary entity is expressed by either of the mappings

$$witn : B \times S \times T \to W \ \ and \tag{4.6}$$

$$witn_* : B \times W \times T \to W \ . \tag{4.7}$$

Definition 4.8: A *propagation of unmediated delegation* is defined as the propagation of a $\tilde{d} \in \tilde{D}$ to another $b \in B$ while applying transformations according to a set Z, valid for a certain period of time $\dot{t} \in \dot{T}$. It is expressed by the mapping

$$prop : \tilde{D} \times \mathcal{P}(Z) \times B \times \dot{T} \to \tilde{D} \ . \tag{4.8}$$

Then a mediated definite delegation with n propagations between brokers can be expressed as a mapping

$$deleg^{\mathrm{n}}_{\mathrm{med}} = \mathrm{med}(\ \mathrm{prop}^{\mathrm{n}}(\ \mathrm{udeleg}(\ldots)\ ,\ldots)\ ,\ldots)\ \ , \tag{4.9}$$

where the case $n = 0$ would need to represent the mapping $deleg_{\mathrm{med}}$ as defined above.

Figure 1 illustrates the concept of the model of definite delegation with the defined mappings

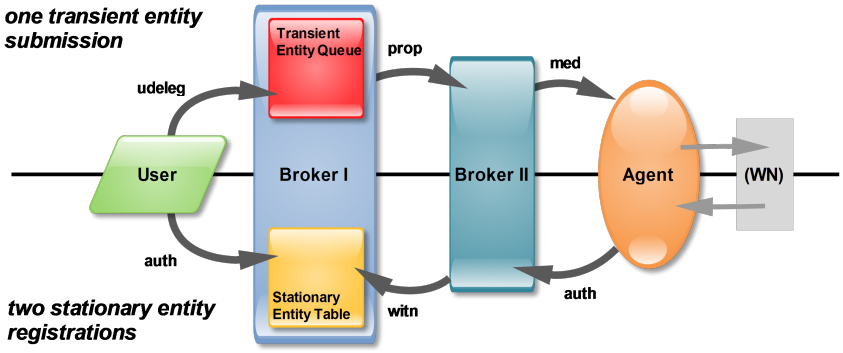


Figure 1. The model of mediated definite delegation with two brokers

as interaction requests. The submission, processing and assignment of a Grid job as a transient entity is demonstrated by the requests *udeleg*, *prop* and *med* from the user over the two brokers to the Site as the agent. A Transfer Request as another transient entity could be processed accordingly. The registration of a Grid file entry as a stationary entity by a user is shown as an *auth* request, and e.g. the registration of a software package by a Grid operator would be equivalent. The registration of a Grid job output file by a WN is represented by the statement of authorship from the corresponding Site in an *auth* request and the consecutive witness of authorship by the second broker in a *witn* request. The WN in the figure was only added for better understanding, it has no direct representation in the model and is fully abstracted by the Site as the agent.
The figure shows a transient entity queue and a stationary entity catalogue, viz. a Grid job queue and a File Catalogue, within the first broker. With corresponding adjustments of the respective *witn* and *prop* requests, the model allows having one or both of them located at any other broker.

4.2. Implementation
The model of mediated definite delegation, as described above with a single broker, is implemented as a prototype within the Grid middleware project *jAlien* [24, 25], using digital signatures to assure the authenticity of the interaction requests defined as mappings in the model. A user signs e.g. the textual representation of a statement of authorship (*auth*) during a file registration or a Grid job description as the unmediated delegation (*udeleg*), using the private key corresponding to her or his Grid Certificate. The VO's central management system as the broker and Sites as agents are each provided with X.509 host certificates and keys, in order to be able to digitally sign the respective mediation of delegations (*med*) or statements of authorship (*auth*). The mediation of delegation is designed as a cascaded signature of the corresponding unmediated delegation. As such, a signed mediated delegation includes the original user-signed unmediated delegation, thereby allowing the verification of any applied transformations and thus establishing actual certified Grid jobs. Certificates are implicitly exchanged upon the

mutual authentication of communicating peers. The signature verification of any interaction request is mandatory before the request's authorization and consecutive processing. Moreover, certificates, statements of authorship and mediated delegations are stored within the VO's central management system, and are provided on demand in order to allow verifying any stationary or transient entity's authenticity.

4.3. Discussion

Using digital signatures based on a public key infrastructure, the model of definite delegation allows instantiating a Grid infrastructure that satisfies the defined objectives 1-4, as forgery or alteration of a job submission or processing can be fully detected.

Objective 10, non-repudiation concerning Grid jobs, can be fulfilled with the restriction of a potential deletion of stationary or transient entities and the respective statements of authorship and mediated delegations. This limitation can be relaxed by the application of deferred deletion, e.g. using history or shadow data structures [26] in order to ensure these entries to be available for verification for a certain time. In order to hold a user responsible for illegal behaviour of a submitted Grid job it would be required to identify evidence of inappropriate or malicious instructions. These would need to be encountered either in the job description or, with respect to the Grid job data and program code origins defined in table 1, directly in File Catalogue entries or software packages, or indirectly as links to external sources. The absence of such evidence would suggest an error or malicious interference on a WN, or a problem in the middleware itself, and indicate an appropriate behaviour of the job submitter.

Objective 7 can be achieved since a signed mediated delegation can be utilized to authenticate and authorize a Pilot job to act only in the name of a job submitter in a least-privilege and predefined mode. Accordingly a Pilot job's credential would no longer require escalated privileges, thereby impeding the proliferation of potential attacks and the possibility of covering tracks.

Presuming a middleware like gLExec to be able to authenticate and authorize a Grid job based on a signed mediated delegation, objectives 5, 6 and 9 could be fulfilled and Grid jobs could be fully validated before their execution. Accordingly the model of mediated definite delegation provides all necessary requirements to allow for a secure implementation of Multi-user Pilot Jobs. The model further requires no additional remote callbacks or service invocations and thereby introduces none of the drawbacks discussed as problem 5.

Concerning objective 8, the integrity of the pilot platform cannot be directly influenced by the presented mechanisms and the objective was introduced as an auxiliary criterion. Nevertheless, the model of mediated definite delegation allows simplifying the identification of flawed behaviour on a WN and ensures the detection of File Catalogue entries changed by a Grid job.

5. Related Work

Beyond the fulfillment of the objectives 1 to 10, a functional concern and boundary condition was to identify approaches allowing the least invasive integration into the current architecture model of the ALICE Grid Services, by implication not involving any additional remote callbacks or service invocations as stated as problem 5. Accordingly, for example GridShib-based [27] implementations or dynamic restricted delegation [28], both based on callback mechanisms, were disqualified.

Snelling et al. [29] proposed a model called *Explicit Trust Delegation (ETD)* to digitally sign job requests in the UNICORE Grid framework allowing for static delegation. In comparison to our work, ETD uses only one signature, either by the user or a trusted Grid portal, which in the latter case is consequently based on unrestricted delegation to the portal. ETD does not distinguish actors such as brokers and agents, and gives no explicit information on intermediate processing, validation or the delegation's consequences for accountability at the execution endpoint. In

version 6 the UNICORE framework's security model was redesigned [30]. Using only a standard X.509 Public Key Infrastructure [8] the new model allows creating chains of signed assertions and thereby for non-repudiation of an assertion's assignment. However, the assertions simply depend on the correctness of logical references in order to bind a delegation to entities, as e.g. Grid jobs.
In [31, 32], a security framework for the Condor distributed batch computing system is presented, based on signed task descriptions, so called *Signed ClassAds*. A *Signed ClassAd* is placed inside an X.509 proxy certificate as a policy information, together with so called *action authorization expressions*. These are rules, expressing which entity is allowed to use the proxy credential for what purposes. The mechanism enables e.g. specifying file checksums as conditions for executables and input files of a task. However, there is no explanation how to establish a dynamically assigned delegation or how to allow transformations of a *Signed ClassAd*, and the framework's design seems to presume all conditions to be expressed explicitly upon the initial submission.

6. Conclusion

We presented an in-depth security analysis of the Single-user Pilot Job model in the ALICE Grid Services. Along the discussion of the shortcomings and problems of the current model, in particular with respect to user accountability and Grid job security, we derived security objectives as necessary criteria for potential solutions. Two existing approaches of Multi-user Pilot Jobs based on X.509 proxy certificates and the Grid middleware gLExec were briefly described. The delegation mechanism provided by X.509 proxy certificates was examined both in general and in detail with respect to their functionality within a Multi-user Pilot Job scenario. Utilizing the derived security goals, we identified and specified fundamental deficiencies and severe security problems in the delegation of privileges, especially concerning user accountability and protection against misuse of a delegator's identity.
As a solution, we defined and illustrated a new delegation model, named mediated definite delegation, which fully complies with the specified objectives in matters of strong accountability, long-term traceability and non-repudiation of responsibility for Grid entities. The model allows a broker to process and transform user-submitted Grid jobs and to dynamically assign them to agents, while preserving the capability to verify the original submission and any later transformations. A prototype implementation using digital signatures based on a public key infrastructure was described and explained, which allows proving a Grid job or file originator's identity transparently. The prototype design foresees a potential interaction with the gLExec middleware and constitutes a necessary framework to achieve full on-site user accounting and protection of Grid jobs and their environment on a WN.

Acknowledgments

We would like to thank the gLExec development group for their valuable input and suggestions and their positive feedback. In addition, we would like to thank Olga Vladimirovna Datskova and Arsen Hayrapetyan for proofreading and their efforts, help and suggestions regarding the formal presentation and illustration of this work.

References

[1] The ALICE Computing Group 2005 ALICE computing technical design report Tech. Rep. CERN-LHCC-2005-018, ALICE-TDR-012 CERN
[2] ALICE Collaboration URL `http://aliceinfo.cern.ch/Collaboration/`
[3] AliEn URL `http://alien2.cern.ch/`
[4] Bagnasco S, Betev L, Buncic P, Carminati F, Cirstoiu C, Grigoras C, Hayrapetyan A A, Harutyunyan A, Peter A J and Saiz P 2008 *Journal of Physics: Conference Series* **119** 062012 URL `http://stacks.iop.org/1742-6596/119/i=6/a=062012`

[5] Worldwide LHC Computing Grid (WLCG) URL http://lcg.web.cern.ch/lcg/
[6] The LCG TDR Editorial Board 2005 LHC computing Grid technical design report Tech. Rep. CERN-LHCC-2005-024 ; LCG-TDR-001 CERN
[7] Grigoras C, Betev L, Saiz P and Schreiner S 2010 *Proceedings of Science: 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research* **ACAT2010** 032 URL http://pos.sissa.it/archive/conferences/093/032/ACAT2010_032.pdf
[8] Cooper D, Santesson S, Farrell S, Boeyen S, Housley R and Polk W 2008 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile RFC 5280 (Proposed Standard)
[9] Foster I 2005 *IFIP International Conference on Network and Parallel Computing* (*LNCS* no 3779) (Springer-Verlag) pp 2–13
[10] Foster I and Kesselman C 1999 *The Grid: Blueprint for a New Computing Infrastructure* ed Foster I and Kesselman C (Morgan Kaufmann) pp 259–278
[11] Neuman B C 1993 *Proceedings of the 13th International Conference on Distributed Computing Systems* pp 283–291
[12] Tuecke S, Welch V, Engert D, Pearlman L and Thompson M 2004 Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile RFC 3820 (Proposed Standard)
[13] Welch V, Foster I, Kesselman C, Mulmo O, Pearlman L, Gawor J, Meder S and Siebenlist F 2004 *Proceedings of the 3rd Annual PKI R&D Workshop*
[14] Kouril D and Kouril D 2005 *6th IEEE/ACM International Workshop on Grid Computing* 63–68
[15] Basney J, Humphrey M and Welch V 2005 *Software: Practice and Experience* **35** 801–816
[16] Groep D, Koeroo O and Venekamp G 2008 *Journal of Physics: Conference Series* **119** 062032 URL http://stacks.iop.org/1742-6596/119/i=6/a=062032
[17] The ATLAS Computing Group 2005 ATLAS computing technical design report Tech. Rep. CERN-LHCC-2005-022, ATLAS-TDR-017 CERN
[18] The LHCb Computing Group 2005 LHCb computing technical design report Tech. Rep. CERN-LHCC-2005-019, LHCb-TDR-011 CERN
[19] Caballero J, Hover J, Litmaath M, Maeno T, Nilsson P, Potekhin M, Wenaus T and Zhao X 2010 *Journal of Physics: Conference Series* **219** 072028 URL http://stacks.iop.org/1742-6596/219/i=7/a=072028
[20] Caballero J, Maeno T, Nilsson P, Stewart G, Potekhin M and Wenaus T 2011 *Journal of Physics: Conference Series* **331** 062005 URL http://stacks.iop.org/1742-6596/331/i=6/a=062005
[21] Paterson S K 2008 *CERN GDB Meeting Geneva* URL http://indico.cern.ch/getFile.py/access?contribId=6&sessionId=4&resId=0&materialId=slides&confId=20235
[22] Ceccanti A 2007 A VOMS overview Tech. Rep. EGEE-II INFSO-RI-031688 NRENS and Grids Workshop Malaga URL http://www.terena.org/activities/nrens-n-grids/workshop-06/slides/ceccanti-voms-overview.pdf
[23] Groep D 2006 glexec deployment models - local credentials and grid identity mapping in the presence of complex schedulers Tech. Rep. INFSO-RI-508833 Joint OSG EGEE Operations Workshop CERN Geneva URL http://indico.cern.ch/materialDisplay.py?contribId=s4t2&sessionId=s4&materialId=0&confId=a062031
[24] jAliEn URL http://jalien.cern.ch/
[25] Schreiner S, Grigoras C, Grigoras A, Betev L and Buchmann J 2012 *Proceedings of Science: The International Symposium on Grids and Clouds 2012* **ISGC2012** 027
[26] Schreiner S, Bagnasco S, Banerjee S S, Betev L, Carminati F, Datskova O V, Furano F, Grigoras A, Grigoras C, Lorenzo P M, Peters A J, Saiz P and Zhu J 2011 *Journal of Physics: Conference Series* **331** 062044 URL http://stacks.iop.org/1742-6596/331/i=6/a=062044
[27] Barton T, Basney J, Freeman T, Scavo T, Siebenlist F, Welch V, Ananthakrishnan R, Baker B, Goode M and Keahey K 2006 *Proceedings of the 5th Annual PKI R&D Workshop*
[28] Ahsant M, Basney J, Mulmo O, Lee A and Johnsson L 2006 *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing* GRID '06 pp 152–159
[29] Snelling D F, van den Berghe S and Quian Li V 2004 *Fujitsu Scientific and Technical Journal* **40**
[30] Benedyczak K, Bala P, van den Berghe S, Menday R and Schuller B 2011 *Future Generation Computer Systems* **27**
[31] Alderman I D and Livny M 2007 *Proceedings of the 16th International Symposium on High Performance Distributed Computing* HPDC '07 (ACM) pp 243–244
[32] Alderman I D 2010 *A Security Framework for Distributed Batch Computing* Ph.D. thesis University of Wisconsin-Madison