A Specification for Security Services on Computational Grids

Franco Arcieri, Fabio Fioravanti, Enrico Nardelli, Maurizio Talamo

Abstract: In this paper we present a computational infrastructure, the Security Backbone, which is able to satisfy security requirements arising from resource sharing and services interoperability in Grid-like environments, without having to rely on a Public-Key Infrastructure (PKI). Motivation of our approach is rooted in the well-known difficulties encountered to show that interoperability of PKIs is effective or efficient in real-world environments.

The proposed solution uses a security layer, lying between the communication and the application level, which provides confidentiality, integrity and authentication services in a fully transparent way from the application point of view, thus enabling the deployment of distributed network applications satisfying the highest security constraints, at a very low organizational and financial cost.

Moreover, we have designed a service for scalable and flexible management of authorization policies governing access to resources shared by members of a Virtual Organization, by improving on the Community Authorization Service distributed with the Globus Toolkit.

Keywords: Grid security, network security, organizational aspects of network security, inter-organizational services, authorization management.

Corresponding author: Fabio Fioravanti <fioravan@di.univaq.it>

A Specification for Security Services on Computational Grids

Franco Arcieri[1], Fabio Fioravanti[2], Enrico Nardelli[1], Maurizio Talamo[1]

(1) NESTOR - Laboratorio Sperimentale per la Sicurezza e la Certificazione di Servizi
Telematici Multimediali - Univ. of Roma "Tor Vergata", Roma, Italia
(2) Dipartimento di Informatica, Univ. of L'Aquila, L'Aquila, Italia.

Abstract. In this paper we present a computational infrastructure, the Security Backbone, which is able to satisfy security requirements arising from resource sharing and services interoperability in Grid-like environments, without having to rely on a Public-Key Infrastructure (PKI). Motivation of our approach is rooted in the well-known difficulties encountered to show that interoperability of PKIs is effective or efficient in real-world environments.

The proposed solution uses a security layer, lying between the communication and the application level, which provides confidentiality, integrity and authentication services in a fully transparent way from the application point of view, thus enabling the deployment of distributed network applications satisfying the highest security constraints, at a very low organizational and financial cost.

Moreover, we have designed a service for scalable and flexible management of authorization policies governing access to resources shared by members of a Virtual Organization, by improving on the Community Authorization Service distributed with the Globus Toolkit.[1]

Computational resources sharing between different organizations in an untrusted environment arises several issues related to information security. This is especially true on *computational grids* [26] where members of different organizations join a *Virtual Organization* (VO) for performing collaborative tasks, and users and resources can be dynamically added to or removed from a VO.

In this paper we address the problem of managing certification and security-related operations on grid infrastructures, with a particular focus on specific needs arising from inter-organizational cooperation.

We have studied how to protect interactions between computational entities belonging to different organizations when such interactions take place over unsecure public networks. Typical examples of critical interaction where security is a primary concern are: (i) transactions involving transfer of funds, (ii) transactions where parties commit to action or contracts that may give rise to financial or legal liability, and (iii) transactions involving information protected under privacy regulations, or information with national security sensitivity.

[1] This work has been partially supported by the Grant MIUR L.449/97, CNR Project "P1 - INTERNET networks: efficiency, integration and security", Research Action "Security Services" and by the Grant MIUR PNR 2001-2003, FIRB Project RBNE01KNFP "GRID.IT: Enabling Platforms for High-Performance Computational Grids Oriented to Scalable Virtual Organizations"

In order to enable secure interactions between network applications in multi-organizational environments with large and rapidly evolving communities, the following standard requirements have to be met:

Confidentiality:nobody but the intended recipient can read the content of a message travelling over an insecure communication network. **Integrity:** unauthorized alteration of messages must be detected and traced. **Authentication:** subjects (i.e. persons or processes) participating in a communication must be sure about the identity of all involved parties. **Authorization:** resources and services can only be accessed by authorized subjects. **Auditing:** the information flow associated with an interaction must be audited either for certifying correct service provision or for identifying who is responsible for failure. Timestamped audits can also be used for a-posteriori monitoring the performance of service provision. **Single sign-on:**users should be able to authenticate themselves only once, at the beginning of the work session. Notice that this behavior may require a mechanism for delegating credentials to remote processes running on behalf of the user.

In this paper we present an architecture which is able to satisfy security requirements arising from resource sharing and service interoperability in inter-organizational cooperation, without having to rely on the existence of a Public Key Infrastructure (PKI) shared by all involved organizations. Indeed, the adoption of a single PKI by different and autonomous organizations would be the source for many technical and organizational difficulties, like certificate usage and management of certificate validity (see Section 1 for details).

The proposed solution uses an infrastructural layer (called *Security Backbone*) for managing security-related functions. The Security Backbone provides services which are needed for secure service interoperability in a completely transparent way from the application point of view, thus allowing for deployment of network applications which satisfy strict security requirements with very low financial and organizational costs.

Moreover, we propose a scalable and flexible system for the management of authorization policies governing coordinated resource sharing in Virtual Organizations, which allows one to specify authorization rights granted by Virtual Organizations to their members, as well as authorization rights granted by resource owners to Virtual Organizations. The proposed solution does not depend on the existence of a PKI shared by all "real world" organizations for performing the signing of authorization credentials or for verifying the identity of a subject. Instead, the system for authorization management we have devised leverages the security services provided by the Security Backbone, thereby avoiding the PKI-specific problems which are present in the Globus Toolkit [25] version of the Community Authorization Service [36].

The Globus Toolkit (GT), by relying on a general-purpose security API (GSS-API) [29], allows security services to be implemented by using different security protocols (like Kerberos, for example). However, in its current implementation, GT's security services heavily rely on the availability of PKIs and using different security mechanisms still requires huge implementation efforts [30,3]. Moreover, GSSAPI does not remove the need for credential translation when enabling interoperability between subjects using different security technologies.

In Section 1 we survey on PKI features and shortcomings. In Section 2 we present our solution for infrastructural security services provision. In Section 3 we describe the authorization model used by the Security Backbone and we present some example scenarios. Moreover, we give a description of the authorization service as a Web Service by using WSDL [18]. We designed our solution dealing with security issues in grids within a very large (8.1 milion Euro) Italian national research project on High-Performance Computational Grids [1] involving the Italian National Research Council (CNR), the Italian Institute for Nuclear Physics (INFN), the Italian Space Agency (ASI), the Photonic Networks Laboratory (CNIT) and many Italian universities. Our group is responsible for research related to grid security issues.

1 Problems with Current PKI Technology

The PKI-based approach is suitable for use in well-structured and trusted environments like scientific communities, but it has demonstrated to be unable to effectively or efficiently support secure interactions when deployed in an open and dynamic environment like the Internet, both for technical and organizational reasons. For a detailed survey on the shortcomings of current PKI technology we refer the interested reader to [28].

1.1 PKI Technical Problems

The most important technical obstacles to the success of the PKI approach in real-world inter-organizational environments are the following:
(i) there are still several open issues about interoperability between PKIs operated by different organizations, and
(ii) the predominant use of Certificate Revocation Lists (CRLs) for handling the (in)validity of certificates makes the PKI not scalable,
(iii) PKI technology is too hard for end-users [27].

In the real world, when members of different organizations join to form a Virtual Organization, they establish a network of relations enjoying a structure which is much richer than the tree-like schema of *hierarchical* PKI. In this scenario *mesh* PKIs seem to be more appropriate, but their adoption dramatically increases the complexity of certificate *path discovery* (the process of constructing the chain of certificates leading to a subject) and *path validation* (the process of checking the validity of each certificate in the path). It is also natural to assume that if a PKI-based grid is to be deployed in a large, world-wide scale, there can be no single top-level CA. Instead several independently managed PKIs will be created, just like it has happened on the Internet for many other services.

In this more general and realistic setting, interoperability can only be enabled by using cross-certification techniques between independent PKIs. However, achieving cross-certification is very hard because of (i) the organizational obstacles deriving from the fact that two or more organizations are forced to trust each other, (ii) the increased computational effort needed for verifying longer chains of certificates, and (iii) the lack of scalability of this approach which requires each CA to recognize each CA it wants to interoperate with.

We want to remark the problems and the risks associated with the existence of a single PKI by quoting P. Alterman, member of the U.S. Federal PKI Steering Committee and Federal Bridge Certification Authority: "There are strong arguments against fielding a U.S. Federal PKI, especially under a single root".

The most relevant of these problems is that a single nation-wide Certification Authority represents an actual threat to individual privacy, as it will enable the government and security agencies to collect personal information of any kind.

Such a single CA would also violate organizational autonomy, as most organizations are reluctant to participate in a single PKI run by an entity other than themselves.

Moreover, the existence of a single supplier of PKI services would generate disastrous consequences to other suppliers: it is easy to imagine the lobbying activity which will be performed by suppliers for winning such a competition.

Also, the overall deployment and operational cost of this approach would be an obstacle to the wide adoption of security services in inter-organizational cooperation. The cheapest solution will be the most popular, and PKI is not by any means the cheapest solution.

We should not forget that "The purpose of deploying a PKI is to provide a secure electronic government utilizing Internet technology, not only to satisfy the little hearts of a dedicated cadre of techno-nerds and paranoiac security gurus but to serve the citizenry", as Alterman states.

Recently proposed solutions try to mitigate scalability and interoperability issues of PKIs by using *bridge certification authorities* [16,37] and *validation authorities* [38]. Bridge CAs do not issue certificates to users, but they are used for creating *bridges of trust* between PKIs operated by different organizations and for translating between different security policies. Validation authorities are entities which are responsible for performing resource consuming tasks like path construction and path validation on behalf of users, possibly by interacting with PKIs using different technologies. Although use of the above solutions can enable better interoperability of PKIs on a large scale, they are currently supported only by very few applications, thus the benefits which can be obtained in the short term by following this approach are minimal. Moreover, the feasability of the approach based on Bridge CAs is currently being tested by the U.S. Federal Bridge Certification Authority [11], but it is still not clear which would be its performance when deployed to support applications' needs on a large scale.

In a PKI each Certification Authority (CA) manages the validity of certificates it releases by making Certificate Revocation Lists (CRLs) available for download on the network. CRLs are large documents signed by the issuing CA containing a list of certificates released by the CA itself which are not to be considered valid anymore. Unfortunately, CRLs suffer from the following serious problems: (i) they do not provide real-time information about the validity of certificates, (ii) their distribution and checking is expensive, and (iii) they are extremely vulnerable to denial-of-service attacks.

The intrinsic problem with the CRL-based approach is that only negative information is provided: if a certificate is in the CRL then it must not be considered valid, but if it is not listed therein then no warranty is given about its validity as, for example, the list may simply be not recent enough. However maintaining CRLs fresh generates very high loads for servers distributing them, due to the simultaneous requests for CRL update by PKI-enabled clients.

Since there is no real economic advantage for CAs which update their CRLs most frequently (except for having a good reputation, of course), currently deployed attempts to solve this problem try to reduce the size of CRLs by grouping certificates in classes or by publishing only changes with respect to previously issued CRLs. However, in the real-world scenario many high-value transactions rely on the validity of certificates and the need for real-time validity assertions is ever increasing.

The CRL-based approach is also exposed to paradoxical situations like the existence of a CRL containing the certificate which was used to sign the CRL itself. Moreover, non-standard situations like this are not handled uniformly by applications.

A more radical solution to manage certificate validity would be not to use CRLs at all, and adopt a protocol which can provide real-time information about the validity of a certificate, like the Online Certificate Status Protocol (OCSP) [23]. This approach, which is encountering an increasing support by vendors, is anyhow not yet a standard component of certificate validation client software installed in the more common applications. Also, OCSP server may be subject to "denial of service" attacks and must satisfy the same strict security requirements of a Certification Authority.

Another obstacle to the adoption of PKI technology is that it is too complex for use by average end-users [27]. Indeed, for example, there is no procedure which allows the end-user to obtain a PKI certificate in an automated, transparent way, like DHCP does for configuration of networking parameters on workstations.

1.2 PKI Organizational Problems

A pure PKI-based approach also suffers from an important organizational problem which is rarely addressed in the literature but is often the culprit of unsuccessful secure service interoperability: Trust in a CA can be established unilaterally.

Any entity in an organization can indeed decide to trust any CA, independently of the organization security policies and without necessarily asking for authorization (see Figure 1).

This behavior is clearly only acceptable in no-profit scientific communities where reliance on security service is non-critical. Indeed, when we focus on the reality of business cooperation it becomes evident that security services can be established only after some kind of agreement among involved organizations is formally in place, that is, *trust between members of different institutions always requires a bilateral agreement at the organizational level*.

This aspect was a further motivation for our choice of putting security services in a layer fully independent from the application one.

A notable exemplification of this organizational requirement is mobile phone roaming in a foreign country, where access to local resources is granted only if an agreement exists between the local company and the home-base one, and it becomes impossible for the user to by-pass the local infrastructure.

In conclusion, PKI technology, despite of considerable recent developments, is not yet to be considered mature for deployment in large and dynamic environments like the

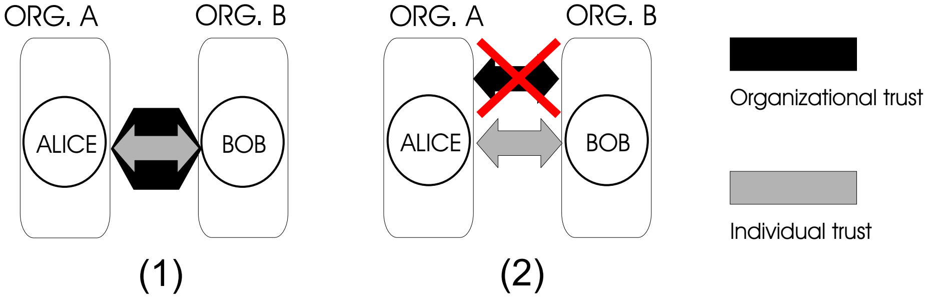

Fig. 1. Organizational and individual trust: (1) the reality of business cooperation, and (2) the approach allowed by PKIs.

grids. An alternative solution to PKI infrastructures for providing security services on a grid is presented in Section 2.

2 The Security Backbone

In this section we present the Security Backbone, an alternative approach for easy and transparent provision of security services at the infrastructure level, independently from locally deployed network technology and topology. In the proposed architecture security services are provided by a layer lying between the application and the communication layers (see Figure 2.B), which is in charge of monitoring network connections directed to or originating from the application level and securing them according to the policies of the Security Backbone.

In our view *security is an infrastructural service of inter-organizational communication, not an add-on service*. Notice how our position is similar to the requirement expressed in the WS-Security roadmap document [19]: "What is needed in a comprehensive Web service security architecture is a mechanism that provides end-to-end security".

Our approach, by making security services readily available to applications in a completely transparent, infrastructural way, allows for separation of issues related to security services and business logic, relieving developers of the burden of managing security-related mechanisms and thereby reducing the risks of introducing security flaws. This is in contrast with the standard approach, where security services are usually provided at different levels in the protocol stack (see Figure 2.A).

Moreover, our approach also solves the organizational problems of PKIs by allowing cooperation between members of different organizations only on top of the Security Backbone layer, which is set up only after a bilateral agreement is formally in place between their organizations. This represents a reasonable trade-off between freedom granted to users by PKI technology and the functionalities needed by business-to-business cooperation.

The Security Backbone also provides auditing services, thus making it possible to certify successful e-services interaction and composition, to identify culprits of bad service provision, as well as monitoring the actual performance of the service as perceived

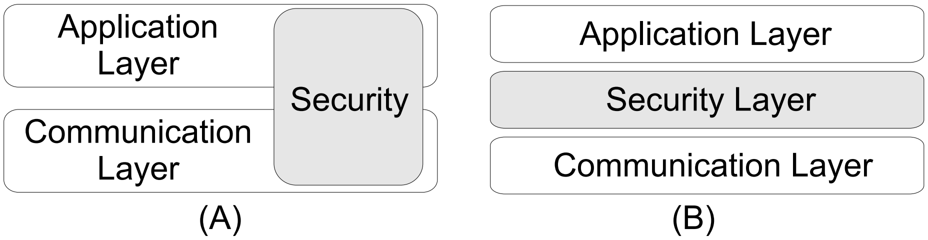

Fig. 2. Provision of security services: the standard approach (A) and the Security Backbone approach (B).

by end-users. There is an increasing interest in techniques which are able to certify correct service execution [31] and this is especially important for composite services, which result from the composition of simpler subservices provided by different organizations [17].

We also want to point out that our solution for security services provision is currently in use within large Italian e-government projects [6,4].

2.1 The Security Backbone Technical Details

The Security Backbone contains the following functional subsystems: (i) confidentiality and integrity services, (ii) authorization service, (iii) authentication service, (iv) documentation subsystem, (v) access policy management, (vi) quality of service monitoring.

We now give some detail on the functions executed by the subsystems and how they have been realized.

Confidentiality and integrity services A mechanism similar to SSL/TLS is used for guaranteeing integrity and confidentiality of exchanged messages: before being transmitted over an insecure communication channel, TCP packets are encrypted by using symmetric cryptography based on session keys which are generated anew for each session and exchanged between communicating parties using asymmetric cryptography.
A part of each subject's private key is distributed by out-of-band methods. Once this part of a subject's private key is arrived at the destination site, the confidentiality and integrity subsystem at the site has to be activated, as described in the paragraph below on the authorization subsystem. After activation, local and remote modules of the confidentiality and integrity subsystem are fully operational.

Authorization service This subsystem takes care of the initial set-up of functions in the security layer. On the basis of the part of the private key obtained by out-of-band methods, an exchange of encrypted messages between the local subsystem and a central control server happens, aiming at registering the local subsystem at the central control server. Hardware identifiers of the communicating machines are exchanged during this phase, so that it is possible to uniquely identify physical sites having the right to access the communication network. After successful completion of this registration procedure the site is activated, its private key is complete and bound both to registered end-user(s) and registered machine(s), and the client is authorized to securely exchange messages.

Authentication service Guarantee of the identification of message source and destination is implemented by having local and remote modules of the authentication subsystem exchange messages over an encrypted tunnel: TCP packets are encrypted and transmitted as payload of IP packets addressed to the other endpoint of the tunnel. Again, encryption uses symmetric cryptography based on session keys, securely exchanged using private keys. In this way, whenever IP packets arrive at the destination endpoint, only those originating from authenticated sources are accepted, while the other ones get discarded.

Documentation subsystem A dedicated subsystem of the Security Backbone records application-level messages exchanged between authorized access points of the communication network, so that documentation can be produced on actually exchanged data. In fact, since service provision is often bound to contractual or legal obligations, it becomes extremely important to certify, when a problem is later found, if and when data were sent and received.

The documentation subsystem is based on an architecture using network probes at network access points for recording exchanged application-level messages. This solution has been extensively described elsewhere [8,9,10]. Here we just want to recall that it works without any change to existing applications, it performs filtering of selected IP packets and reconstructs messages exchanged at the application-level, using highly efficient algorithmic solutions [32], which make the solution scalable and with a very low overhead.

Access policy management It is also possible to define and enforce the desired policy for access management at a central control point. In fact, both authorization and documentation services are fully parameterized, making it possible to implement various access control policies.

For example, users or groups of users can be given different rights (e.g. read-only, write, publish, query) to different resources in a dynamic and flexible way, without requiring any modifications at the application level. After the initial set-up and registration phase of the access point, end-users' access rights can be dynamically established by means of a communication between the local and the central modules of the access policy management subsystem.

Quality of service monitoring Quality of service measuring and monitoring in a business cooperation scenario needs techniques which measure and certify *actual* application level performance of service flows spreading on a network in consequence of a service request. To obtain precise measurements, it is then needed to record the actual behaviour in the network of IP packets corresponding to service flows, while it is not possible to use estimation based approaches, where sophisticate techniques have been proposed for accounting and billing [21,22]. The same reasons prevent the use of flow statistics like those being provided by Cisco NetFlow [40].

To the best of our knowledge no solution for the problem of actual performance measurement of distributed e-services is known in the literature beyond ours: our solution is based on the same technique used to provide documentation services (see paragraph above) and is described in more detail in [7,5].

3 Authorization Management on a Computational Grid with the Security Backbone

The Security Backbone can be easily deployed for creating a computational grid which allows secure utilization of resources shared by members of a Virtual Organization. In this scenario, the configuration of the Security Bacbone is managed by the VO administrator and each non-virtual organization which wants to allow access to the grid to some of its members, or make some of its resources available, will have to join the Security Backbone infrastructure which transparently provides, among other services, mutual authentication, integrity and confidentiality for network communication.

In this section we present a model for authorization management and we show it in action in two different usage scenarios: in the first scenario, the set of authorization rights granted to VO users does not change over time, while in the second authorization rights can be dynamically managed.

3.1 An Authorization Model

In the following we use a simple yet flexible authorization model, inspired by the requirements which led to the development of languages [34,33,24] and models [39,2] for management of authorization rights in distributed network environments.

In the considered authorization model we can identify three main entities: *subjects*, *resources* and *actions*. Entities are specified by a set of *attributes* of the form $\langle n, v \rangle$, where n is the attribute name and v is the attribute value.

A *subject* is an entity which wants to perform an action on a resource. It can be specified by using a name and, optionally, a host or a role. For example, a valid subject may be the attribute $\langle$"*name*", "*John Smith*"$\rangle$, and $\langle$"*host*", "*jsmith.employees.mycompany.com*"$\rangle$.

A *resource* is a computational entity which is available for use to members of a VO. A resource is typically specified by using the following information: the name of the resource, the host where it is located, and the application protocol and the network port which must be used for performing actions on the resource. Example of resources are FTP directories, filesystems and web applications.

An *action* is an operation which a subject wants to perform on a resource. Actions can be specific to a particular application protocol and thus, not all actions can be performed on a given resource. The complete set of actions which can potentially be performed on a resource must be explicitly agreed upon by the organizations involved and stated in formal agreement documents. Example of actions are the following: read, write, execute, HTTP GET, HTTP POST.

Authorization *policies* are sets of authorization *rules* which specify if and how subjects can perform actions on resources by using constraints on resource and action attributes (f.e. "read-write access is granted to FTP directories below /pub/incoming on host A"), or time related constraints (f.e. "access is only granted between 9 AM and 7 PM", or "access is granted for a maximum time period of two hours").

In order to ease the definition of authorization policies, authorization rules need not refer to every particular instance of subjects, resources or actions but can refer to classes of subjects. When evaluating an authorization policy, rules can be combined in different ways. We assume that the authorization rights granted by an authorization policy consists of the union of the authorization rights granted by each applicable authorization rule herein contained.

We now illustrate two different usage scenarios: a scenario where authorization rights are statically defined and cannot be changed, and a scenario where authorization rights can be dynamically modified. In both cases, users are authenticated by the Security Backbone at the beginning of the session, by performing the single sign-on procedure described above. Notice that, by following this approach, each organization still retains full control of the hosts operating on the grid, and, as already mentioned, no changes to applications or to intra-organizational architectures are required.

3.2 A Static Authorization Scenario

In a simple scenario, the set of authorization rights owned by users of a VO is statically determined by the configuration of the Security Backbone, and does not change during the lifecycle of a VO unless the Backbone is externally reconfigured by manual intervention.

The authorization rights owned by the user are not limited to those which are explicitly created according to the Backbone configuration. Indeed, a user process running at a remote site can access resources located at other sites if the configuration of the Security Backbone which controls communication between the host where the process is running and the host where the resource is located permits so. The newly requested resource can be a process itself which may require access to further resources, and so on (see Figure 3).

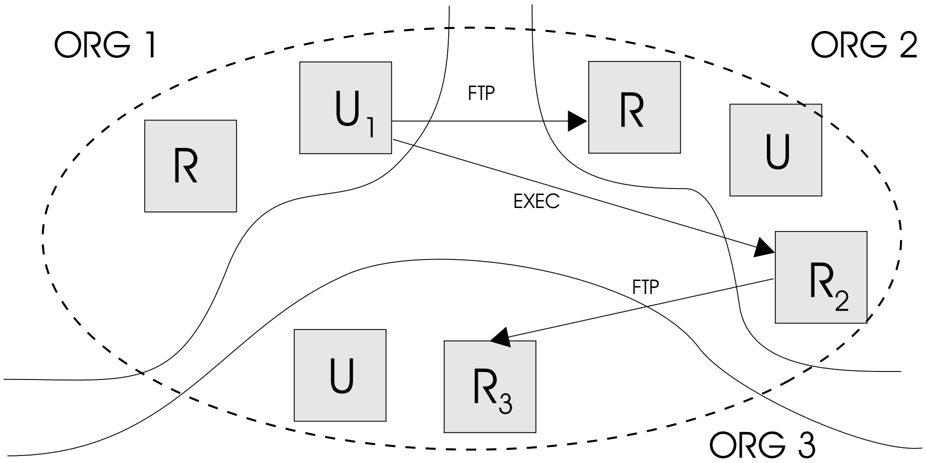

Fig. 3. A Static Authorization Scenario: although there is no direct agreement between Org1 and Org3, U_1 can leverage authorization rights owned by R_2 for accessing R_3.

From a mathematical model point of view we can represent this authorization relation as a labeled directed graph where nodes can be labeled by subjects and resources and edeges are labeled by actions. Then, the set of authorization rights owned by each subject can be thought of as the transitive closure of the peer-to-peer authorization relation constructed by starting with the authorizations for the resources which can be directly accessed by the subject. a subject s can perform an action a on a resource r iff there exists a path from s to r labeled $a_0, \ldots, a_n, a$ where, for all $i = 1, \ldots, n$, if action a_i is performed then action a_{i+1} can also be performed.

In this scenario, differently from what happens on Globus grids, requests for access to resources are not mediated by a user-proxy. This enables enhanced scalability, as there is no single point of failure, while retaining control of authorization policies. Moreover, interactions between hosts can be audited and documented by the Security Backbone in real-time, thus providing a useful tool for detecting possible anomalies and for providing legal evidence in case of judicial dispute.

3.3 A Dynamic Authorization Scenario

In this section we illustrate a scenario where authorization rights can be dynamically managed by interacting with the Security Backbone.

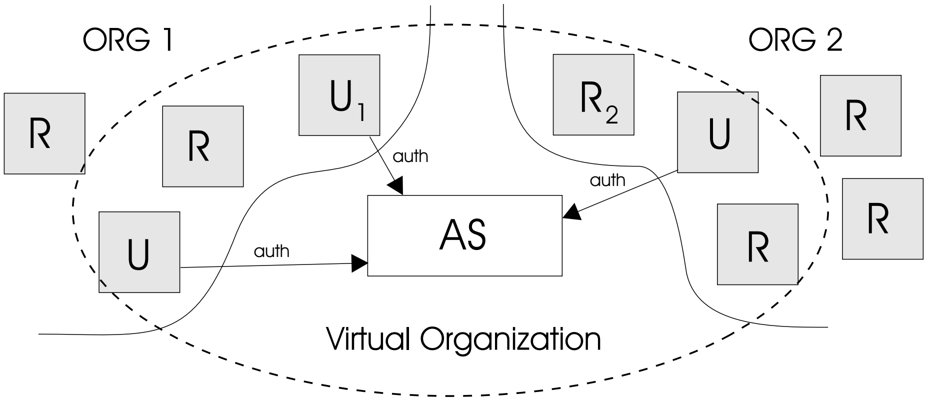

Fig. 4. The configuration of the Security Backbone in the initial state of the Virtual Organization.

In the initial state of the considered scenario, i.e. when a Virtual Organization is created, the only interactions allowed by the Security Backbone are authorization requests from members of the VO to the VO's authorization server, also referred to as the Policy Decision Point (PDP), that is the machine which accepts authorization requests, evaluates authorization policies and replies with authorization decisions (see Figure 4). The VO's administrator is the unique responsible for management of the VO's authorization server.

For issues of reliability, the VO authorization server may be replicated on different machines. In this case, members should be allowed to send authorization requests to all authorization servers used by the VO and standard replication techniques should be used for ensuring overall consistency of the authorization servers.

Notice that the set-up phase of a Virtual Organization, which involves configuring the PDP as well as user and resource sites, cannot be completely automated as it relies on the existence of credentials which must be obtained by out-of-band methods (see Section 2.1). Offline procedures must also be performed when users or resources belonging to new organizations want to join an existing VO. However, apart from the cases mentioned above, by following our approach one can dynamically modify the set of authorization rights granted to users of a VO, as described below.

When a subject (a member, or a process running on a member's behalf) wants to perform an action on a resource, it sends an authorization request containing attributes of the resource, action, and other relevant information to the authorization server. The authorization server, upon receiving the request for authorization, examines it and retrieves policies which are relevant for determining whether and how to satisfy the request.

As a result of the decision process, the authorization server sends back to the requesting subject a response containing the authorization decision (which f.e. can be one of *Permit*, *Deny*, *Don't Know*, *Error*). If the authorization request is accepted the authorization server proceeds in activating a procedure which reconfigures some software components affecting the behavior of the Security Backbone. Only after this reconfiguration process, the subject is allowed to establish a secure channel with the resource over the Security Backbone and to perform operations which are compliant with the authorization policies defined by the VO and the resource provider (see Figure 5). The secure channel is then destroyed at the end of the work session or after a timeout occurs.

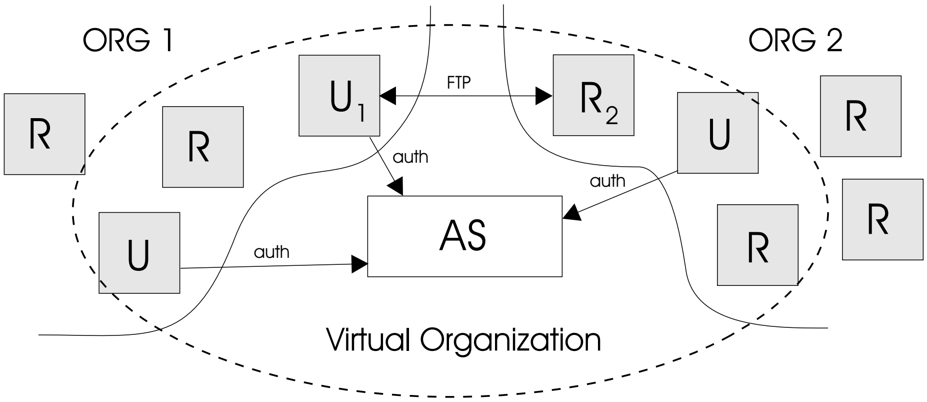

Fig. 5. The configuration of the Security Backbone after creation of a secure FTP channel between member U_1 and resource R_2.

Notice that in our framework it is not needed to include information about the requester's identity as this can be obtained by the Security Backbone itself when the request is evaluated (recall that authorization requests are performed over secure channels). In some cases however, in order to manage requests performed by processes with delegated rights, it might also be useful to specify the identity of the requesting subject.

Our authorization model enjoys the same flexibility of the CAS system [35,36], while allowing non grid-aware applications to securely access and share resources on the grid.

Indeed, management of a VO authorization policies can be performed at the authorization server by extending the model described above to consider management services as resources. Thus, in the initial state of the VO, the VO administrator is granted access to the services for managing the VO itself (adding, removing users and groups) and its authorization policies. On the other hand, each resource owner must only recognize those VO's which will be using its resources and need not be concerned about their dynamics.

Moreover, by following our approach, security services can be provided both to applications which are able to interact with the Security Backbone, as well as to applications which were not developed to manage security related issues. In the former case, for example, applications can perform authorization requests by interacting with the authorization server as a Web Service (see Section 4 for details), while in the latter case authorization requests can be performed by using special purpose client application similar in concept to those distributed with the CAS system.

The architectural solution we propose uses techniques and technologies which are well-known in network security, but is novel as it makes security services available at the infrastructure level, thus enabling secure interoperability between legacy network applications in a non-intrusive and cost-effective manner.

Other systems provide security services at the network level (like IPSec or IPv6) or at the transport level (like TLS [20]) but they require changes at the application level. Thus, they do not represent an effective solution for providing security to existing applications which are expensive, and often impossible, to modify. On the contrary, the Security Backbone, does not require any change to existing applications and can coexist with locally deployed security solutions.

4 Specification of the Authorization Service as a Web Service

In this Section we introduce some basic concepts about web services and we present a simple WSDL document describing the authorization service used for creating a secure channel for accessing a resource by using the FTP protocol.

The problem of enabling interoperability between e-business applications on the World Wide Web is currently being addressed by using XML-based [15] standards like the Web Service Definition Language (WSDL) [18] and the Simple Object Access Protocol (SOAP) [14]. These technologies provide a framework within which it is possible to expose existing network applications in a uniform and abstract manner.

A WSDL document contains the definition of the message exchange pattern between a service requester and a service provider (one-way, request/response, or publish/subscribe), together with the definition of the structure of the messages, the message data types and the bindings to concrete network protocols (HTTP GET/POST, SOAP, MIME) to be used for communication of messages. Messages exchanged by the service requester and provider are typically formatted according to the SOAP protocol. SOAP messages are XML documents consisting of three parts: (*i*) an envelope describing the message content and the rules for processing it, (*ii*) an optional header for extending a message with new features like authentication and transaction management, and (*iii*) a body containing data related to service requests or responses. Although HTTP is used as the main network protocol, SOAP can potentially be used in combination with a variety of other protocols, like FTP, SMTP or RPC.

In a Web Service architecture [13], if a service requester wants to use a web service, it must first obtain the WSDL document containing the service description, either from the service provider itself or from a network-accessible catalog where service providers publish their service descriptions, like UDDI [12]. In order to successfully complete the service invocation, interaction between requester and provider must adhere to the service specifications contained in the WSDL document. As long as the parties involved in service provision adhere to the same service description, the software systems actually providing the Web services can be implemented by using any technical solution, ranging from Java Servlets to legacy applications.

In the case of the FTP protocol, resources can be described by using the name or the address of a host and the path of a file or directory on the host filesystem, while actions are described by a single attribute which can take one of the following values: READ, STOR, WRITE, MKD, DELE.

Below, we present a simple WSDL document describing how interaction takes place between a subject requesting access to a FTP resource and the FTP authorization service.

```
<?xml version="1.0" encoding="UTF-8"?> <wsdl:definitions
targetNamespace="http://DefaultNamespace"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://DefaultNamespace"
xmlns:intf="http://DefaultNamespace"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:types>
  <schema elementFormDefault="qualified" targetNamespace="http://DefaultNamespace"
  xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="createSecureChannel">
    <complexType>
     <sequence>
      <element name="host" nillable="true" type="xsd:string"/>
      <element name="resource" nillable="true" type="xsd:string"/>
      <element name="action" nillable="true" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="createSecureChannelResponse">
```

```
    <complexType>
     <sequence>
     <element name="createSecureChannelReturn" nillable="true" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
  </schema>
 </wsdl:types>


  <wsdl:message name="createSecureChannelResponse">
     <wsdl:part element="intf:createSecureChannelResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="createSecureChannelRequest">
     <wsdl:part element="intf:createSecureChannel" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="gridBackbone">
     <wsdl:operation name="createSecureChannel">
        <wsdl:input message="intf:createSecureChannelRequest"
        name="createSecureChannelRequest"/>
        <wsdl:output message="intf:createSecureChannelResponse"
        name="createSecureChannelResponse"/>
     </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="gridBackboneSoapBinding" type="intf:gridBackbone">
     <wsdlsoap:binding style="document"
     transport="http://schemas.xmlsoap.org/soap/http"/>
     <wsdl:operation name="createSecureChannel">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="createSecureChannelRequest">
           <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="createSecureChannelResponse">
           <wsdlsoap:body use="literal"/>
        </wsdl:output>
     </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="gridBackboneService">
    <wsdl:port binding="intf:gridBackboneSoapBinding" name="gridBackbone">
    <wsdlsoap:address
    location="http://backbone-auth-server:6080/gridBackbone/services/gridBackbone"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

References

1. Grid.it: Enabling platforms for high-performance computational grids oriented to scalable virtual organizations. http://grid.it:8080/InFlow.

2. Gail-Joon Ahn. Specification and Classification of Role-based Authorization Policies. In *Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* June 09 - 11, 2003 Linz, Austria, 2003.
3. Edgardo Ambrosi. Creazione di un sistema plug-in di AA in Globus ed aggregazione dinamica di porzioni di griglie computazionali attraverso CAS: Analisi di fattibilita'. Master's thesis, Advanced Master Thesis in Network Security, Univ. Roma "Tor Vergata" and INFN - Frascati, 2004. submitted for partial fullfilment of the Master Degree.
4. F. Arcieri, F. Fioravanti, E. Nardelli, and M. Talamo. The italian electronic identity card: a short introduction. In *The National Conference on Digital Government Research (dg.o2004), May 24-26, 2004, Seattle, Washington, USA*.
5. F. Arcieri, F. Fioravanti, E. Nardelli, and M. Talamo. Inter-organizational e-services accounting management. In *3rd IFIP conference on e-Commerce, e-Business, and e-Government (I3E-03)* Sao Paolo, Brasil. Kluwer Academic Publishers, September 2003.
6. F. Arcieri, F. Fioravanti, E. Nardelli, and M. Talamo. A layered it infrastructure for secure interoperability in personal data registry digital government services. In *14th Int. Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE'04), March 28-29, 2004, Boston, USA*. IEEE Computer Society, 2004.
7. Fabio Arcieri, Fabio Fioravanti, Enrico Nardelli, and Maurizio Talamo. Certifying performance of cooperative services in a digital government framework. In *3rd International Symposium on Applications and the Internet (SAINT'03)*, pages 249–256, Orlando, Florida, USA, January 2003. IEEE Computer Society Press.
8. Franco Arcieri, Elettra Cappadozzi, Enrico Nardelli, and Maurizio Talamo. SIM: a working example of an e-government service infrastructure for mountain communities. In *Workshop Electronic Government (DEXA-eGov'01), associated to the 2001 Conference on Databases and Expert System Applications (DEXA'01)*, pages 407–411, Munich, Germany, September 2001. IEEE Computer Society Press.
9. Franco Arcieri, Giovanna Melideo, Enrico Nardelli, and Maurizio Talamo. Experiences and issues in the realization of e-government services. In *12th Int. Workshop on Research Issues on Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE'02)*, pages 143–150, San Jose, California, USA, February 2002. IEEE Computer Society Press. An extended version is published in the journal "Distributed and Parallel Databases".
10. Franco Arcieri, Giovanna Melideo, Enrico Nardelli, and Maurizio Talamo. A reference architecture for the certification of e-services in a digital government infrastructure. *Distributed and Parallel Databases*, 12:217–234, 2002. A preliminary version was published in the proceedings of the 12th Int. Workshop on Research Issues on Data Engineering (RIDE'02).
11. U.S. Federal Bridge Certification Authority. http://csrc.nist.gov/pki/fbca/welcome.html.
12. T. Bellwood, L. Clement, D. Ehnebuske, A. Hately, M. Hondo, Y. Husband, K. Januszewski, S. Lee, B. McKee, J. Munter, and C. von Riegen. Universal description, discovery and integration of web services (UDDI) version 3. http://uddi.org/pubs/uddi_v3.htm, 2002.
13. D. Boot, M. Champion, C. Ferris, F. McCabe, E. Newcomer, and D. Orchard. Web services architecture. http://www.w3.org/TR/ws-arch, 2002.
14. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S. Thatte, and D. Winer. Simple object access protocol (soap) 1.1. http://www.w3.org/TR/SOAP, 2000.
15. T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. eXtensible Markup Language (XML) 1.0 (Second Edition). http://www.w3.org/TR/REC-xml, 2000.
16. W. E. Burr. Public key infrastructure (PKI) technical specifications: Part a - technical concepts of operations. US Federal Public Key Infrastructure Tech. working group, September 1998.
17. Fabio Casati, Mehmet Sayal, and Ming-Chien Shan. Developing e-services for composing e-services. In *Proceedings of CAISE 2001, Interlaken, Switzerland, June 2001*, 2001.

18. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl, 2001.
19. IBM Corporation and Microsoft Corporation. Security in a web services world: A proposed architecture and roadmap. ftp://www6.software.ibm.com/software/developer/library/ws-secmap.pdf, 2002.
20. T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246, January 1999.
21. N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *ACM-SIGCOMM Internet Measurement Workshop (IMW'01),* San Francisco, Ca., USA, Nov.01.
22. C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *ACM-SIGCOMM Internet Measurement Workshop (IMW'01),* San Francisco, Ca., USA, Nov.01.
23. M. Myers et al. Online Certificate Status Protocol (OCSP). RFC 2560, June 1999.
24. P. Ashley et al. Enterprise Privacy Authorization Language (EPAL). http://www.zurich.ibm.com/security/enterprise-privacy/epal/.
25. Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 2(11):115–129, 1998.
26. Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organization. *International Journal of Supercomputer Applications*, 15(3):200–222, 2001.
27. Peter Gutmann. Plug-and-Play PKI: A PKI your Mother can Use. In *Proceedings of the 12th USENIX Security Symposium*, pages 45–58, 2003.
28. Peter Guttman. PKI: It's Not Dead, Just Resting. *IEEE Computer*, pages 41–49, August 2002.
29. J. Linn. Generic Security Service Application Programming Interface (GSSAPI). RFC 2743, January 2000.
30. Patrick Moore, wilbur Johnson, and Richard Detry. Adapting Globus and Kerberos for a Secure ASCI Grid. In *Proceedings of the 2001 ACM/IEEE conference on Supercomputing* Denver, Colorado, 2001.
31. E. Nardelli and M. Talamo editors. *Proceedings of the First International Workshop on Certification and Security in E-Services (CSES 2002), August 28-29, 2002, Montreal, Canada*. Kluwer Academic.
32. Enrico Nardelli, Maurizio Talamo, and Paola Vocca. Efficient searching for multidimensional data made simple. In Jaroslav Nešetřil, editor, *7th Annual European Symposium on Algorithms (ESA'99)*, pages 339–353, Prague, Czech Republic, July 1999. Lecture Notes in Computer Science vol.1643, Springer-Verlag.
33. OASIS. eXtensible Access Control Markup Language (XACML). http://www.oasis-open.org/committees/xacml/.
34. OASIS. Security Assertion Markup Language (SAML). http://www.oasis-open.org/.
35. L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. 2002.
36. L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. The community authorization service: Status and future. In *CHEP03*, La Jolla, California, March 24-28 2003.
37. William Polk and Nelson Hastings. Bridge certification authorities: Connecting b2b public key infrastructures. US National Institute of Standards and Technology, 2001.
38. William Polk, Nelson Hastings, and Ambarish Malpani. Public key infrastructures that satisfy security goals. *IEEE Internet Computing*, pages 60–67, August 2003.
39. Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.
40. Cisco Systems. Netflow. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/.