A Federated Framework for Secure Collaborative Systems

Daniel Kouřil[1,2], Luděk Matyska[1,2], Michal Procházka[1,2], Tomáš Kubina[1], Lukáš Spurný[1]

[1]Masaryk University, Botanická 68a, 602 01 Brno, Czech Republic, [2]CESNET z. s. p. o., Zikova 4, 160 00 Praha 6, Czech Republic

{kouril,ludek, michalp}@ics.muni.cz, xkubina@fi.muni.cz, luspu@mail.muni.cz.

Keywords
Identity federations, videoconferences, PKI, OpenVPN.

1. EXECUTIVE SUMMARY

Videoconferences can play an important role in contemporary e-learning systems since they allow students to attend courses that are given by teachers from a remote location. Videoconferences also make life easier for researchers from many different institutions who collaborate on a joint project, since they can easily coordinate their work. Many collaborative and videoconferencing systems have been proposed in the past, however, these systems usually do not offer a proper level of security and/or user friendliness. They either do not provide any security functionality at all or they require users to obtain and maintain a new set of authentication credentials, such as username/passwords, which is annoying and complicates the use of the systems. Such an arrangement is also difficult to administrate as the authentication credentials must be managed by the administrators of each collaborative system independently, because the systems are usually not integrated with any user management system.

Strong support for authentication and authorization in e-learning and videoconferencing systems is necessary in deployments where sensitive information is communicated between the parties. The emerging model of *identity federations* can be utilized to solve the problem by introducing decentralized user account management and providing a layer that makes authentication and authorization transparent for end applications. A federation is an infrastructure connecting user management systems from different institutions to provide standardized access to information about users maintained by their systems. An *Identity Provider* (IdP) service is built on the top of each local user management system, providing a standardized interface for authentication information and attributes about the users. End services (*Service Providers* – SP) process the attributes returned by the user's home IdP and use them to make access control decisions.

The standard federation model is suitable for web applications, however, the functionality cannot be directly utilized by other systems that are not based on the web technology, such as videoconferencing systems. Recently, we have introduced a design that describes how non-web applications can be integrated into federations using the Public Key Infrastructure technology and an infrastructure of VPN channels. It utilizes digital certificates issued by an on-line CA to users authenticated using the federation model. Apart from usual information, the certificates also contain a set of attributes assigned by the IdP to the user, which can be used by end services to make access control decisions.

In this paper we will present an updated version of the design and a pilot implementation and deployment. The fundamental part of the framework is the OpenVPN software, which is able to make secure and pervasive VPN tunnels between clients and servers and represents an underlying layer. OpenVPN uses PKI for authentication of clients. We have also enhanced OpenVPN to be able to process certificates issued by the federated on-line CA and make authorization decision afterwards. Inside the tunnels, standard MBone videoconferencing tools are used, which are very common collaborative applications but they do not provide sufficient level of security. Authorization policies are specified by the administrator of a videoconference, who defines simple rules defining a set of authorized users, based on their attributes managed by their home organization (such as the identification of courses they are enrolled to, or their affiliation with a faculty, etc.). The system is also used as part of a collaborative environment within e-Health projects in which we participate.

2. INTRODUCTION

More and more users start to use videoconferences and other collaborative tools as another medium for communication since they allow users to easily exchange information despite the physical location of the participants. Despite indisputable advantages of collaborative applications, they also present a new security issues that must be dealt with properly. For example, physicians discussing their cases will not be willing to communicate details about their patients over an unsecured channels because they could be intercepted and misused by unauthorized people. Similarly, many research groups try to hide their current activities from competing groups. An analog scenario could be applied to e-learning courses servicing licensed contents that should be only made available to authorized recipients. From the security point of view there are three requirements that videoconference applications should implement: *authentication* and *authorization* of the users and *data protection*.

Unfortunately, virtually all current videoconferencing systems either does not address security issues at all or employ proprietary security measures. Such measure usually do not provide sufficient easiness of use because they require independent user registration and some type of custom user credentials for accessing the systems. On the server side of the application it is required to have an own management system for user accounts and roles. If a videoconferencing system supports authorization for accessing virtual rooms, it is in most cases implemented as a shared password per virtual rooms. This approach for achieving security is hard to maintain for administrators and also users are required to have another set of credentials. Such an arrangement introduces new challenges to solve to provide users with a comfortable solution.

Today we witness a big progress in deploying of identity federations, which allows to leave the management of the users to theirs home organizations. Therefore, applications do not need to implement an own authentication system, instead they delegate authentication to the user's home institution. Unfortunately todays' federations are focused only on web environment, which is why we introduced the design of the framework that is able to integrate non-web videoconference applications into the federations adopting the federation approach to security. The framework is designed as a general solution with the collaborative systems being just one possible environment supported.

The paper is organized as follows. In chapter three we will discuss how federations work, than we will describe technologies which are part of our framework. Our framework will be described in chapter 5.

3. BASIC TECHNOLOGIES

Our framework is designed as general as possible using well-tested and widely-used technologies. The framework employs a public-key infrastructure (PKI) and identity federations to provide basic security features. Communication between the parties is transparently encapsulated in an additional channel provided by our infrastructure. This chapter describes identity federations, PKI which is used for authentication and also as a container for transfer the authorization data from user to the server and OpenVPN which provides secured channel.

3.1.IDENTITY FEDERATIONS

An identity federation is an infrastructure connecting user management systems from different institutions to provide standardized access to information about users maintained by their systems. Federations provide a virtual bus layer to which systems for user management and end applications can connect and share authentication and authorization data. Every organization participating in a federation manages its users by a local user management system. An *Identity Provider* (IdP) service is built on the top of each local user management system, providing a standardized interface to access authentication information and other attributes about the users. Any party in the federation can get this information by calling the IdP service using a standardized protocol. End services (*Service Providers — SP*) are able to process the data returned by the user's home IdP and use them to make access control decisions. Before users are allowed to use a service, they have to present a

set of *attributes* issued by their home IdP. These attributes are provided to users or to a service working on their behalf upon proper authentication of the user with the IdP.

The major advantage of using the federation model lies in the fact that users authenticate to arbitrary services with their home institution's credentials (which may be a username and password, a digital certificate, a hardware token, or something else.). Every SP in the federation can use this mechanism transparently from the user's point of view. There is no need to introduce new credentials for every new service or to synchronize existing credentials (like passwords) among different services. Having no additional credentials also means there is no need to distribute them among the users. Such an arrangement not only eases credential management but also makes it more secure, as users are only required to maintain one piece of authentication information and no user's credential are exposed to the SP. Similar functionality is offered by PKI (see chapter 4.1). However, it is too difficult and time consuming for most users, especially if their home institution does not already have an extensive network of registration authorities and properly trained user support. The federation model is undoubtedly more acceptable to the users, as it is not tied to any particular authentication method and institutions can decide the most appropriate method for their users.

Upon proper user authentication, the IdP provides a set of attributes that represent additional information about the user. The attributes are very often encoded using the Security Assertion Markup Language (Security Assertion Markup Language 2 (SAML) V2. Technical Overview, 2007). In this way the home institution provides information that can be used, e.g. for specification of a group of users without explicitly naming them. For example, it is possible to create a collaborative session for students enrolled on particular course at different institutions (provided that information about the courses a person is enrolled on is made available as part of the attributes). The whole communication can be logged for future auditing, so the session administrator can learn who participated in a session without needing to be given this information in advance, in order to authorize access.

Enhanced privacy is a potential side effect of the above-mentioned use of attributes. The IdP could provide only attributes, not a precise user identity (i.e., it could provide the information that a person is enrolled on course identified as CS102 without revealing his or her name or other unique identification). The attributes are sufficient for a service to make an authorization decision, however the precision of audit trace is lost (while the privacy of the user is enhanced). In the event of abuse, the home institution is still able to identify the user from their local logs. This approach is interesting if we do not want to reveal the individual user's identity to a collaborative session administrator, e.g., in some semi-anonymous survey.

3.2.PKI

PKI plays a significant role in our framework since it is used as a secondary authentication mechanism and as a container carrying additional user attributes from their IdP. The whole architecture of PKI is based on three pillars. The first one is represented by a key-pair (consisting of a public and private key) that is bound to its owner using a *public key certificate*, most often encoded in the X.509 format (X.509, 2005). The second component is a *Certification Authority* that signs the certificates with its signing key. The last pillar is formed by the *relaying parties* who trust the CA and accepts certificates it signed. All the three parts are tightly interconnected.

In PKI every entity holds its key pair that is used for asymmetric cryptography. That means the data encrypted by a public key can be decrypted only with the corresponding private key and vice versa. A personal digital certificate binds the key pair and its owner and provides information about the owner identity. Each certificate contains a public key and information about the person such as his or her name, institution and location. The certificate along with all necessary information is signed with the private key of a CA, whose identification is also included in the certificate. Together with these mandatory information the certificate can also contains additional information called *extensions*. There are some well known extensions but other can be defined to carry any type of data. In order to make the CA operation scalable, the model of PKI introduced the concept of *Registration Authorities* (RA) that are responsible for proper authentication of applicants who ask for certificates. In this model the CA signs certificates requests that are validated by authorized RAs.

Concept of PKI becomes more scalable with the RA but not enough, since the process of obtaining a certificate is time-consuming and hard to understand and follow for ordinary users. The latter problems can be mitigated by using on-line CA's that can be accessed by common web browsers and allows to hide the process of key-pair generation and certificate signing. However, standard RA's still require the users to visit them personally, which is not comfortable for the end users. In order to make life for the user easier and still retain a sufficient level of trust, we introduced an on-line CA integrated with the federation model. The CA relies on the federation to provide proper authentication of the users. Attributes issued by the user's home institution are used to determine a profile to be used to generate the certificate. The profiles define how the certificate will look like, e.g., the size of the private and public key, content of mandatory fields and extensions. Attributes issued by the IdP are also included in the certificate as an extension for further use by end applications.

3.3.SECURED TUNNELS

Secured tunnels and virtual private networks (VPN) can be used to transparently protect payload transported by protocols that do not provide security functionality. We have tested and routinely use OpenVPN as an application providing a secured tunnel between an arbitrary client and server. OpenVPN provides secure communication together with ability of firewall and NAT penetration. It makes a VPN on the application layer from the ISO/OSI perspective which means on top of the UDP or TCP protocol. There is no need to modify configuration of network elements on the path from the client to the VPN server. Also whole communication between client can be encrypted and peers authenticated.

The client side needs OpenVPN software to be able to connect to the OpenVPN server. This software makes a virtual network adapter and sets appropriate routing table records for the client. Clients must authenticate either using their certificate or username and password. Once a VPN connection is established, all network communication to the videoconferecing server is automatically routed via the VPN and is secured by OpenVPN.

In some cases we do not need to make a tunnel on the IP layer, instead, depending on the type of application used, we would rather create a port to port channel connecting two particular machines. There is plenty of software components providing such a functionality, such as stunnel, whose benefit over OpenVPN is simpler installation since it can be done by an ordinary user without administrative permissions. To our best knowledge, however, none of them supports UDP, which is a crucial transport mechanism for videoconferencing.

4. SECURING COLLABORATIVE SYSTEMS

Several types of videoconferencing tools exist nowadays such as AccessGrid, VRVS, VRVS-EVO, Skype, Adobe Connect. However, they either do not tackle with security issues or their approach is not convenient for use.

Adobe Connect provides a very convenient mechanism for videoconferecing since it is implemented in Flash that is supported by virtually every current web browser and users do not need to install any software on their desktop. Adobe Connect enables to lock videoconferencing rooms using a password that users must enter before accessing the room. This mechanism is simple however, not convenient for the users and also do not provide any possibility to mount sophisticated access control mechanisms. Moreover, even if the Adobe Connect server were modified to support authentication via an identity federation, the payload would not be secured and even could be subject of an man-in-the-middle attack or session hijacking. A more detailed analysis of security parameters of remaining products mentioned above was published in our design paper (Transparent Security for Collaborative Environments, 2007) describing the basic framework for secure collaborative systems. In this section we provide an updated version of the framework and describe current state of implementation.

5. COMMON ACCESS TOOLKIT FOR FEDERATIONS

All parts of our framework were discussed and in this chapter we put these parts together. We called the framework "COMMON ACCESS TOOLKIT FOR FEDERATIONS (CAT)" to reflect universality of the tools since the goal is to provide a general layer to transparently secure a wide range of applications. Our primary target are collaborative tools but other kind of applications can be supported as well. In the first stage of the project we have focused only on Microsoft Windows operating systems for client side, but implementation for Linux and Mac OS will follow.

5.1.CAT architecture

The basic idea of CAT is to establish a universal network channel providing basic security features, which is combined with easiness to use by the users. CAT aims at applications communicate only through the secured channel, which means that the server side listens only on IP addresses belonging to the secured channel. Authentication and authorization is provided by the CAT infrastructure without changing either client or server side of the application.

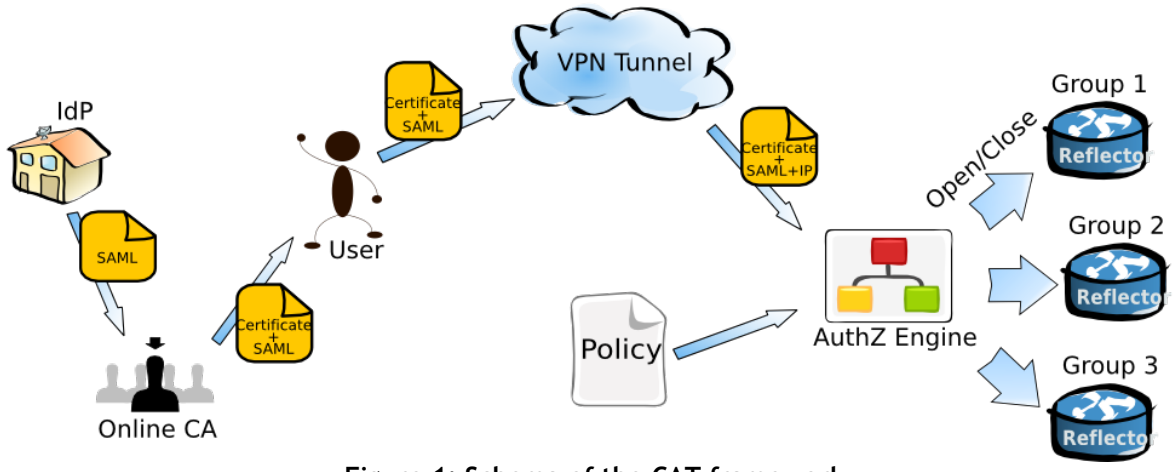

Figure 1: Schema of the CAT framework

The design of the framework (see Figure 1) is split into three parts. First part involves the process users' entrance to the infrastructure. In the architecture, we have introduced a concept of explicit logging into a collaborative environment. Since we have chosen PKI as the primary security mechanism, in order to log in the environment, the users have to obtain their public-key certificates. Since we do not want to complicate the user's perception of the system we provide a convenient way of getting certificates based on the use of an on-line CA.

The next part of the architecture deals with establishment of secured channels between the user and server. We based our solution on OpenVPN, which provides a general, open-source solution to build an VPN. Authentication of users is used on their certificates.

The third part is responsible for access control to the application server based on clients' certificate and/or attributes assigned. The authorization engine is plugged into the OpenVPN server and is called after a connection from a client is initiated. The engine is either able to directly communicate with the application server (if such an operation is supported by the server) or use dynamically set rules for local firewall controlling access to the server.

After passing these three parts, a user can start his or her application, which is now able to communicate with the server side using an encrypted channel.

5.2.LOGGING INTO FEDERATION

Since we do not expect users to already have certificates we operate an on-line CA that allows user to obtain certificates easily. We utilize the federated on-line CA from the GridShib (GridShib Project website, 2008) project, which implements the CA functionality as a common federate Service

Provider. The federative approach makes it possible to employ the ordinary federative strong authentication approach and allow user to receive certificate based on their home credentials. Moreover, the CA add additional attributes about the users to the certificate issued. These attributes are stored as X.509 extensions that are available to every end service that the certificate owner contact. In order to retain privacy and we do not store real attributes but rather a SAML artifact, which is a reference to a real values maintained by the IdP.

In our first implementation of the framework user has to manually get the certificate from the on-line CA and then save the certificate in an appropriate place on his or her computer. We have decided to get this process automated and provide better certificate management, covering contacting the CA, automatic lookup of certificate by the OpenVPN client and certificate renewal.

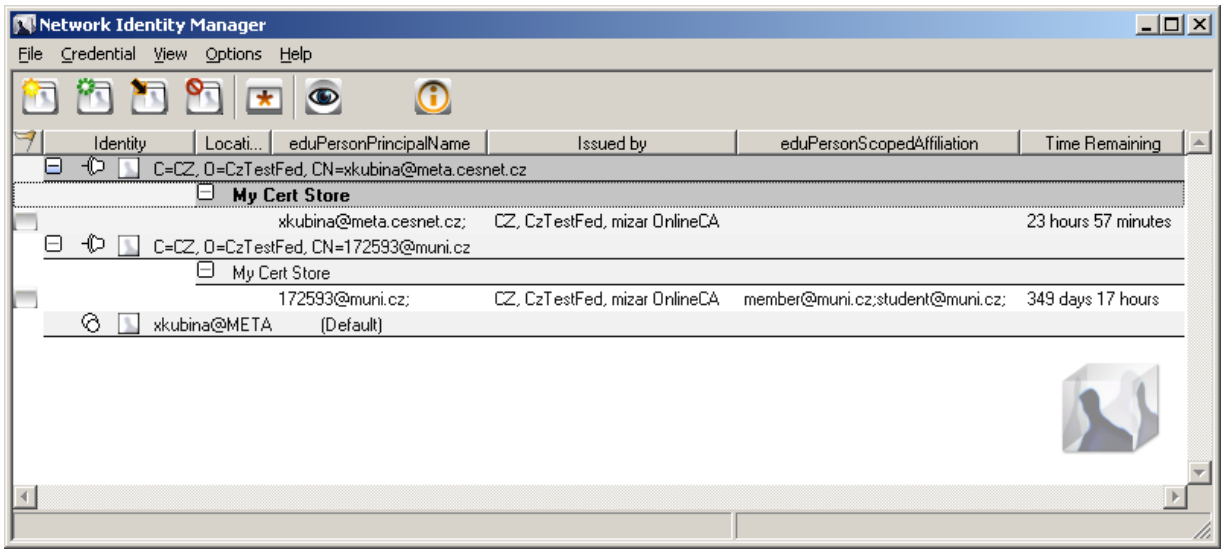

Figure 2: Main window of Network Identity Manager

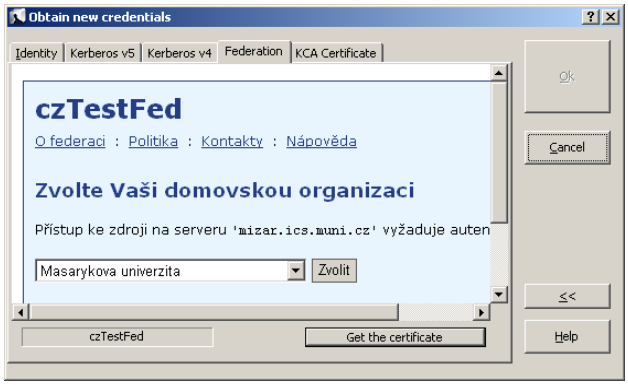

Figure 3: Using the embedded browser

We have experiences with software Network Identity Manager (NIM) (Network Identity Manager, 2008), which is a GUI application used to manage user credentials. Originally, NIM was developed to manage user's Kerberos tickets but it can be extended by various plugins to support different types of credential as well. We have develop a plugin to help users to obtain the certificate from the on-line CA. When a user wants to get a certificate, NIM launches embedded Internet Explorer, which tries to access the on-line CA (Figure 3). If the user is not authenticated yet, they are redirected to their home institution IdP as usual. Because they access the IdP from the embedded browser everything looks like a common authentication process in the federation. After successful authentication the on-line CA issues the certificate and the embedded browser saves it in the standard Windows certificate store, which is shared among all Windows applications. NIM is able to

scan the certificate store periodically and show to the user only the certificates issued by the on-line CA. Certificates and their status are shown in the NIM main window (Figure 2). Users can see information about the certificate, its lifetime and also attributes placed in the certificate (Figure 4).

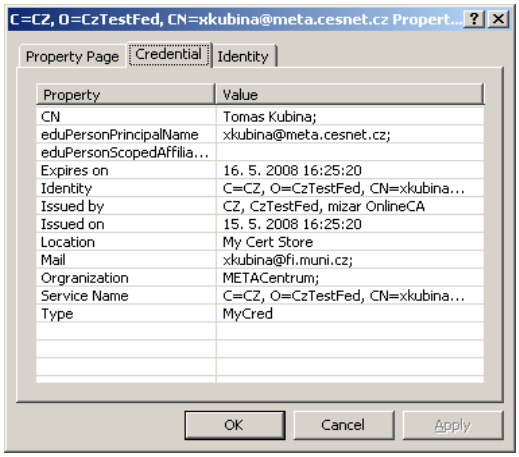

Figure 4: Details about a certificate

The OpenVPN client can easily access the certificates stored in Windows certificate store and use it for authentication with OpenVPN server. The certificates managed by the NIM can be used not only by OpenVPN but also to access web pages using https authentication or by the Putty SSH client to access the Linux/Unix machines by the SSH protocol.

5.3.VPN

We decided to employ OpenVPN since it provides a viable and well-tested solution for VPN connection. Besides good networking features it also supports clients authentication using digital certificates and modular access control mechanism.

All client workstations are connected to a VPN server in a point-to-point mode. The set up of the VPN network guarantees that only the traffic belonging to the videoconference's services is sent into the VPN tunnel. In stage of early testing, we wanted to use one of the private IP address ranges as defined by RFC1918 for internal VPN addressing. After the survey at selected institutions we have found it very complicated to avoid conflicts with internal address ranges used at various institutions, especially as new institutions may join. The whole overlay network is therefore addressed using a public IP address range assigned by RIPE, but the addresses are treated as internal address and not routed outside the VPN overlay network. The OpenVPN server could run in two modes: either over UDP or TCP. The UDP mode is preferred due to better performance. The TCP mode can also run through an HTTP or SOCKS proxy.

We employed an OpenVPN based environment to provide an infrastructure for collaboration between the participants of the Ithanet project. As part of this work a set of measurement of the characteristics of the OpenVPN-based infrastructure was also performed (Secure and Pervasive Collaborative Platform for Medical Applications, 2007). The results showed that OpenVPN in UDP mode has minimal influence on the latency and jitter and provides enough capacity of the tunnel, therefore it is suitable to be used to transfer videoconference data. Even the plain TCP and TCP with HTTP proxy modes do not degrade he performance much, especially in latency, which has main influence on the quality of a videoconference.

In order to make the framework as much transparent as possible we have to tie the process of creating the secured channel and starting the application. This can be done by *profiles* that define requirements of the application on the network such as IP address of the server and range of used ports. A profile also contains startup parameters for the application. In fact, the profiles act as a wrapper around the client's application and OpenVPN client. The result of the profile is a startup script, which the user uses to start the application. The process of authentication, authorization and creating of the tunnel is hidden from the user.

The OpenVPN server was extended by an access control hook that implements an authorization engine. The engine evaluates a policy specified on the server against the client's certificate and SAML attributes embedded in the certificate. If the user is not authorized to access the application, the authorization engine replies to the OpenVPN server and the tunnel is immediately closed. If the user meets the policy requirement the engine either notify the application server or set configuration rules on a firewall controlling access to the server.

5.4.SECURING VIDEOCONFERENCES USING CAT

The CAT framework was originally designed for use with videoconferencing applications based on the Mbone Tools (Sumover Project website, 2008) and a reflector (User Empovered Collaborative Environment: Active Network Support, 2004). The Mbone Tools contain client's applications VIC for video transmission and RAT for audio transmission. These tools communicate with the reflector that duplicates all received data to all connected clients and simulates a multicast connection. A videoconference based on these tools has special requirements on the network, it cannot be run behind NAT and must use UDP. The RTP protocol used does not provide any any encryption or authentication. The reflector can make only IP based authentication and completely lacks of authorization.

CAT fulfill network requirements and also provides a higher level of security without making any change to the Mbone Tools or reflector. The usage of CAT and this type of videoconference was tested by the users from Ithanet project (Ithanet - developing en infrastructure of electronic communication for Thalassaemia research, 2007). In order to make CAT user friendly we made an installer, which contains the OpenVPN client, Mbone Tools and a corresponding profile. A new installer is being prepared that also contains the Network Identity Manager to allow logging into the framework.

6. CONCLUSION

We have designed and implemented a framework that is able to set up authenticated and authorized secured data tunnels between user's application and server-side of this application. We have used the PKI technology and OpenVPN to implement the framework. Because initial authentication of the user is made through the federation applications the users have an easy access to the infrastructure. Users do not to maintain new passwords and all security operations are performed transparently by the framework without user's intervention. Users' attributes carried by the certificates allows to implement complex access control policies. The framework was successfully tested with videoconference applications based on MBone Tools and Reflector on real users.

7. ACKNOWLEDGEMENT

The work has been supported by the research intent *Optical Network of National Research and Its New Applications* (MSM 6383917201) of the Ministry of Education of the Czech Republic, and by project *Common Access Toolkit for Federations* (253R1/2007) funded by the CESNET Development Fund and Masaryk University.

8. COPYRIGHT NOTICE

9. REFERENCES

GridShib Project (2008). GridShib Project website. Retrieved April 15, 2008, from: http://gridshib.globus.org/.

Hladka, E. (2004). *User Empovered Collaborative Environment: Active Network Support*. Brno, Czech Republic: Masaryk University.

Holub, P., Hladka, E., Prochazka, M., Liska, M. (2007). Secure and Pervasive Collaborative Platform for Medical Applications. *Studies in Health Technology and Informatics*. Amsterdam, The Netherlands: IOS Press, 229-238.

Lockhart, H., Wisniewski, T., Cantor, S., Mishra, P., Lien, J. (2007). *Security Assertion Markup Language 2 (SAML) V2.0 Technical Overview.* Retrieved April 15, 2008, from: http://www.oasis-open.org/committees/download.php/22553/sstc-saml-tech-overview-2%200-draft-13.pdf.

Network Identity Manager (2008). MIT Kerberos website. Retrieved April 15, 2008, from: http://web.mit.edu/kerberos/.

Prochazka, M., Matyska, L., Hladka, E., Kouril, D., Holub, P. (2007). Transparent Security for Collaborative Environments. White Plains, New York, ICST. *The 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, ISBN 1-4244-1317-6

Sumover Project website (2008). VIC and RAT Tools. Retrieved April 15, 2008, from: http://www.cs.ucl.ac.uk/research/sumover/.

Vejvalka, J., Holub, P., Prochazka, M., Liska, M., Hladka, E. (2007). Ithanet - developing en infrastructure of electronic communication for Thalassaemia research. *Conference Proceedings of EMMIT 2007, Euro-Mediterranean Medical Informatics and Telemedicine, 3rd International Conference*. Mangalia. ISBN 978-973-739-423-1. 181-186.

X.509 (2005), *ITU-T Recommendation X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*. Retrieved April 15, 2008, from: http://www.itu.int/rec/T-REC-X.509/e.