Resources Sharing and Access Control in Group-oriented Networks: Fednet and Related Paradigms

Malohat Ibrohimovna
Technical University of Delft
The Netherlands

K.M.Ibrohimovna@ewi.tudelft.nl

Sonia Heemstra de Groot
Twente Institute for Wireless and Mobile Communications and
Technical University of Delft
The Netherlands

sonia.heemstra.de.groot@ti-wmc.nl

Abstract

A Personal Network (PN) is a network composed of devices of a person that can communicate with each other independently from their geographical location. Extra functionality in PNs enables the cooperation amongst different persons forming a group-oriented network called a Federation of Personal Networks (Fednet). A Fednet is a secure, opportunity or purpose driven ad-hoc network for sharing personal resources. A Fednet can be composed for applications in different areas, e.g. education, entertainment, business, emergency, etc.

A number of group-oriented resource-sharing technologies for distributed environments have been reported in the literature, such as grids, Virtual Organizations, Secure Virtual Enclaves and P2P networks. All these technologies for sharing resources have their own peculiarity in the architecture, their implementation, and in the ways they control the access to shared resources. This paper provides a comparative overview of these technologies with our Fednet concept. In addition, a special attention is given to various approaches for controlling the access to shared resources in cooperative distributed environments, in particular grid environments. We discuss the details of these access control architectures, advantages and disadvantages of these approaches.

Keywords: sharing resources, group-oriented networks, personal networks, federation of personal networks, access control.

1. INTRODUCTION

Personal devices with networking capabilities have become an integral part of daily activities, business and entertainment. Examples of such personal devices are mobile phones, PDAs, digital cameras, laptops, desktops, MP3 players, printers, home appliances, gadgets, etc. It is exciting and useful, when these personal devices and appliances could communicate with each other and provide meaningful services to their owners independently of their geographic location. This is the idea behind the concept of a Personal Network (PN) [2].

The personal devices in a PN are organized into clusters. A cluster is a networked group of personal devices located in the vicinity of each other. A simple PN consists of a local cluster around the user. Figure 1 illustrates an example of a PN. In this PN a local cluster is extended with other remote clusters, i.e. office cluster, home cluster and car cluster with the help of interconnecting infrastructures. This way, personal devices can form a distributed personal environment of a user.

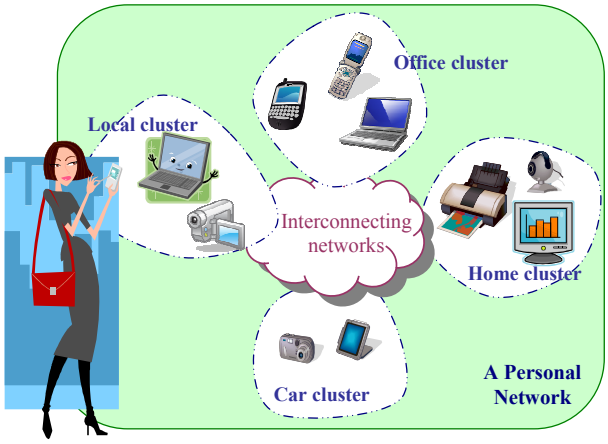

Figure 1. Example of a Personal Network

By adding extra functionality, PNs can form group-oriented networks called Fednets. The concept of a Fednet was introduced in [3] and defined as a temporal, ad hoc, opportunity- or purpose-driven secure cooperation of independent PNs. In the core of the Fednets is sharing personal resources and services to achieve a common objective.

A number of technologies and paradigms for sharing resources have been reported in the literature. But in each of them the concept of 'sharing' appears with a new flavor: in grid computing [4] it is sharing the spare CPU resources, processing power and storage facilities; in P2P networks it is sharing data and multimedia content, such as music, clips and video; in Wireless community networks [5] it is sharing services and facilities such as Internet access; and in Fednets it is a broad range of sharing personal resources and services among the users on demand. All these technologies for sharing resources differ in their architecture, their implementation, and in the ways they control the access to shared resources.

The contribution of this paper is a comparative overview of several group-oriented resource-sharing technologies for distributed environments. In addition, a special attention is given to various approaches for controlling the access to shared resources in cooperative distributed environments, in particular grid environments. We discuss the details of these access control architectures, advantages and disadvantages of these approaches. In this sense, this paper extends the survey on resource sharing technologies presented in [1] (UBICOMM 2008).

The organization of the paper is the following. In Section 2, we explain the motivation for PNs to federate, and briefly describe the basic component-level architecture of a Fednet and access control in Fednets. Further in this section, we explain our approach to analyze the system based on functional modules. In Section 3, we discuss some of the related technologies for group-oriented communication reported in the literature. In Section 4, we analyze the access control mechanisms used in grid environments based on the generic authorization framework for Internet resources and services [24]. Finally, in Section 5, we summarize the survey and draw conclusions.

2. FEDNETS

In this section, we describe Fednets, present their architecture and the access control to its resources.

2.1 Motivation to federate PNs

Persons usually communicate with each other, carry out common tasks and cooperate with each other in order to reach a common goal. They might encounter many situations, when it is desirable and beneficial to enrich their cooperation by connecting their Personal networks for raising the efficacy of their communication towards reaching a common goal. A network that is created by connecting independent PNs is called a Federation of Personal networks (Fednet). The concept of a Fednet was introduced in [2] as a temporal, ad hoc, secure cooperation of independent PNs. PNs, driven by a certain purpose or triggered by opportunity, can form a Fednet to achieve a common objective by means of sharing personal resources and services. Figure 2 depicts an example of a Fednet formed of four PNs to share their resources and services. In this Fednet the PN owners can run different applications and benefit from sharing personal resources (e.g. data and multimedia) and personal services (e.g. printing, displaying, storing, connectivity to the infrastructure, routing and Internet access).

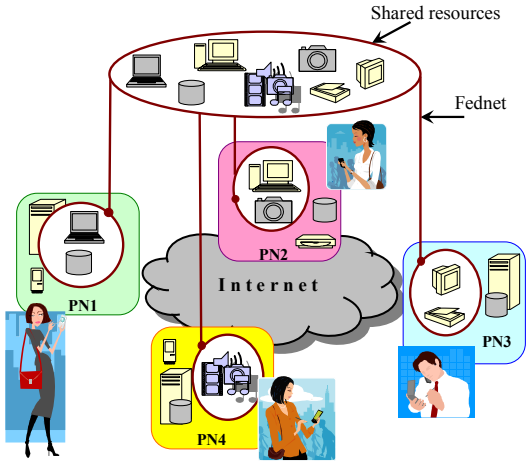

Figure 2. A Fednet and its shared resources

The main objective in Fednets is to facilitate reaching a common task. For example, consider the following scenarios:

a) *File sharing*. Colleagues attend a conference together. They meet after the conference and form a Fednet with the goal to share photos and videos from the conference. The Fednet here is purpose driven and is formed between laptops, photo cameras and PDAs of the colleagues. This example is elaborated in [6].

b) *Camera view and sensor information sharing*. People with wearable cameras and sensors can form a Fednet with the goal to exchange valuable information (e.g. images, temperature and location) in disaster relief situation. The Fednet here is purpose driven and is composed by wearable cameras and sensors of different people.

c) *Facility sharing*. Friends meet at home of one of them. They show each other photos in their iPods. The host has a big screen. Using this opportunity, they form a Fednet with the goal to display the pictures on a big screen. The Fednet here is opportunity-driven and is formed between the iPods of friends and the screen.

These examples show the wide applicability of Fednets in various situations for ad hoc occasional sharing of personal resources. It is important to note, that Fednets can have a large scale involving a large number of distributed

PNs. Examples of such Fednets are a secure network between patients and doctors for remote healthcare services or a virtual classroom environment formed for a distance learning course scenario.

2.2 Architecture of Fednets

A Fednet is composed of interconnected PNs. PNs belong to different owners and represent independent security domains; therefore the architecture of a Fednet should take into account the following considerations:

• The resource and service owners might want to keep the control over their resources and services themselves;

• The internal structure of a PN is not to be revealed to other PNs.

Two approaches [7], [8] have been taken so far to build the architecture of the Fednets: using overlays between PNs [7] and using service proxies at the gateways of the PNs [8]. The difference between the two approaches is in the way service access control and service provisioning are carried out. In the overlay approach, each personal device in the Fednet carries out the service access control, while providing a service to others.

In the proxy-based approach, the services of a PN are accessed by other PNs not directly at the service providing device, but at the gateway of a PN by means of service proxies. Besides, the access control to the PN services is carried out not in every device in a PN, but at the border of the PN (i.e. the gateway of a PN), so other personal devices inside the PN do not need to have access control capabilities. Having the access control at the borders, allows each PN to have a separate security domain in a Fednet and to keep its autonomy. This way, the proxy-based approach meets better the above-mentioned considerations. This advantage in comparison to the overlay-based approach has been our main motivation for choosing for the proxy-based architecture in our work. Figure 3 illustrates the basic proxy-based architecture of a Fednet.

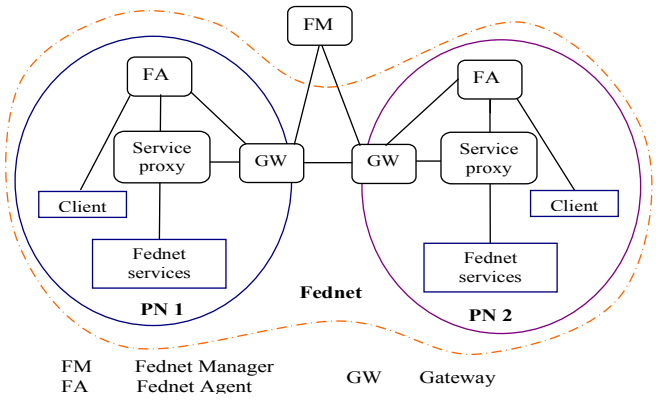

Figure 3. Basic proxy-based architecture of a Fednet

The main components of a Fednet are the Fednet manager (FM) and the Fednet agent (FA). The Fednet manager is responsible for management and control of the Fednet, such as creating and dissolving a Fednet, and accepting and removing Fednet members. The Fednet agent is responsible for the management and control functions of the PN when operating within a Fednet, such as joining and leaving a Fednet, controlling access to its personal resources and services. The Gateway (GW) is a device with multiple network interfaces. A PN communicates with other PNs of the Fednet through this gateway, by making one of its interfaces publicly addressable. The Service Proxy is a functional component that is located at the gateway of the PN. Its role is to prevent direct access of other Fednet members to the personal devices (services) of a PN, by making the services available at the gateway of a PN. The services offered by the PN to the Fednet are called Fednet services. A client is an application or a personal device within a PN requesting a Fednet service. The proxy-based architecture of a Fednet is given in more details in [6].

2.3 Access control in Fednets

In Fednets two or more PNs share their resources and services with each other. When people share their personal resources and services, an important issue is providing a proper access control to them. We took a two-level approach for the access control in Fednets. We consider that *becoming a member* of the Fednet (i.e. access to the community) and *using the services* and resources of the Fednet (i.e. access to the community services and resources) are two different issues. Therefore, we distinguish between these two levels of the access control. The two-level approach gives a separation of concerns in the access control.

The first-level access control takes place when a new member joins a Fednet. The first-level access control in a Fednet is carried out by the Fednet manager. Having a centralized entity (i.e. the Fednet Manager) for this task facilitates the management of the Fednet with dynamically joining and leaving members. The accepted Fednet members receive a membership credential which is used to prove their membership within the Fednet. Membership credential also indicates a PN's *membership class*, which is the ranking of the member within the Fednet based on its contributions and reputation.

The second-level access control is the access control to the Fednet services. It takes place when a Fednet member requests a Fednet service. The second-level access control in a Fednet is carried out by the Fednet agent of a PN. This allows a PN to keep the control over its personal resources and services. This meets the preferences of the PN owners, who usually prefer not to delegate the access control rights over their personal resources to a third party.

Fednet services are distinguished between common and specific. Common services are accessible by all members of the Fednet upon presenting their membership credential. Special services require the second-level fine-grained access control at the PNs.

2.4 Functional modules in the architecture of Fednets

In this paper, we focus on the *concept of sharing resources*, in particular on the following questions:

- Who is sharing?
- What is shared?
- How is the sharing done?

To analyze the system based on these questions we introduce a system architecture decomposed into functional modules, as is shown in Figure 4. By *module* we mean a collection of functional components.

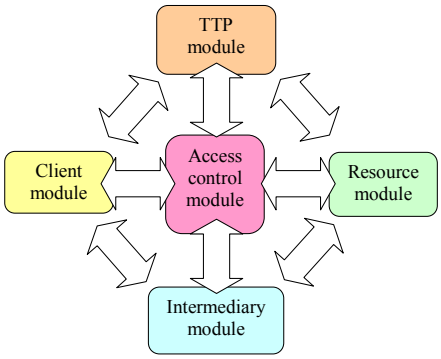

Figure 4. Functional modules of system architecture.

The following functional modules are relevant when sharing resources between different owners:

- *Client module,* which contains the administrative domains and the users of shared resources.
- *Trusted Third Party* (TTP) module that contains a trusted authority between all administrative domains.
- *Intermediary module* that contains technology-specific components of the architecture.
- *Access control module* that contains the mechanisms or methods used in the access control to shared resources.
- *Resource module* that contains types of shared resources and services.

We took this approach to analyze the system, because it gives us better understanding of how the system works and how the sharing is accomplished. It shows explicitly the interrelation of functionalities in the sharing process. It also helps us to compare Fednets and other related technologies with respect to the concept of sharing resources.

Figure 5 shows the mapping of the Fednet architecture, depicted in Figure 3 into functional modules grouped according to the above mentioned criterion.

- In Fednets *the client module* consists of PNs that belong to different owners.
- *The TTP module* can contain a Certification Authority (CA), who issues digital certificates for PNs to certify their identity in the authentication process (see Figure 5, arrow 1).
- *The intermediary module* contains a service directory (SD), which stores the list of Fednet services; a gateway (GW) through which all external communication of a PN takes place and a Service proxy, which makes a service available at the gateway of a PN.

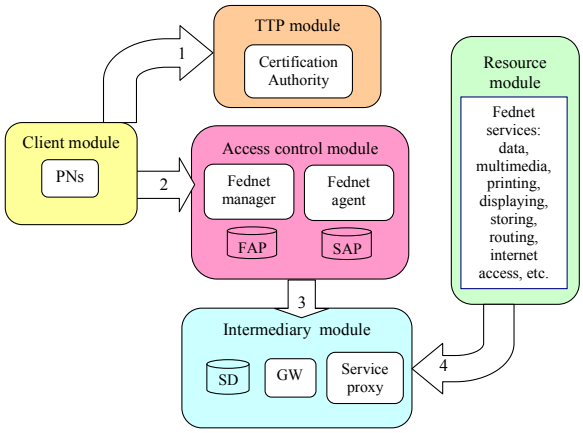

Figure 5. Functional modules in the Fednet architecture.

- *The access control module* contains the Fednet manager; Fednet access control policies (FAP), which are the rules about how a new member is accepted to a Fednet; the Fednet agent and service access control policies (SAP), which are the rules about how the access to the PN services is controlled.
- *The resource module* contains Fednet resources and services that are shared between the PNs. The access to the Fednet and its services is achieved through the access control module (Figure 5, arrow 2), which produces the access control decisions that are enforced by the intermediary module (arrow 3), where a service proxy located at the gateway of the PN acts as a delegate of a PN service (arrow 4).

In Section 3 we describe some of the related paradigms proposed in literature. In order to analyze them, we use our approach of decomposition of the system architecture into functional modules, presented in Figure 4.

3. RELATED PARADIGMS FOR SHARING RESOURCES

The main purpose of Fednets is sharing the resources and services which belong to different persons. A number of technologies and paradigms for sharing resources and

services have been proposed in the literature [3], [9-11], [19-23]. While some of them focus on sharing a particular service, other systems are designed for a group of various applications.

To place the Fednets amongst related technologies we discuss some of them in this Section. We provide the definition of a technology, discuss its differences and similarities with Fednets, and analyze the functional architecture focusing on the access control to shared resources in each technology.

3.1 Grids

A grid [3] is a hardware and software infrastructure that allows resources to be shared across organizational boundaries. Grid computing was started with the idea of sharing spare processing power and storage facilities to carry out big scale computations that were not possible by using single machines. Later, organizations using grid networking started cooperation based on mutually agreed rules to form so-called Virtual Organizations (VO) [9]. There are many grid projects all over the world, such as the project CrossGrid [10], which addresses realistic problems in medicine, environmental protection, flood prediction, and physics analysis; the project AccessGrid [11], which enables connecting people using remote video, visualization techniques, microphones and cameras. An impressive amount of examples of grid projects and applications is given in [12] and [13].

Functional architecture of grids

Figure 6 depicts the functional architecture of a grid network.

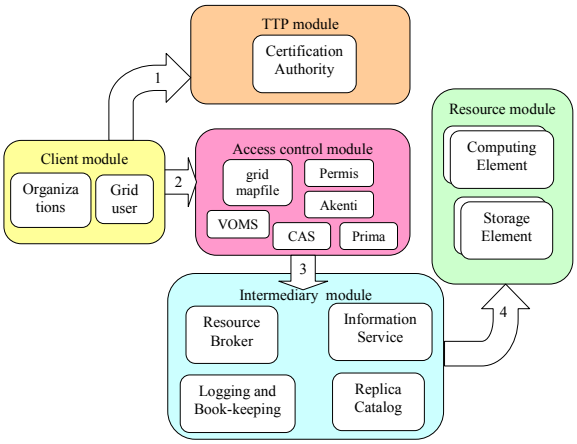

Figure 6. Functional modules in Grid architecture.

The *client module* contains the grid users, and organizations, which are the members of a VO.

The *TTP module* contains a Certification Authority (CA), who issues Grid digital certificates after certifying user's identity, for example showing staff ID.

The *resource module* contains computing and storage elements.

The *intermediary module* is responsible for managing the job allocation and execution. Grid computation shares resources online through the Internet, so anyone may access shared resources. In order to use a grid facility, the grid user first gets a certificate from the CA (see Figure 6, arrow 1) and submits a job (arrow 2) to the grid facility via the user interface. The application control mechanisms carry out the access control by checking the grid map file. This file holds a mapping list of the authenticated grid users to their local account names. When the user is found in the grid map file, the resource broker uses information service and replica catalog (arrow 3) to find a suitable computing element and a storage element to execute the job (arrow 4). When the job is done, the resource broker returns the result to the user. Logging and book-keeping service maintains the records on the job execution procedure, which are purged when the job is completed.

The *access control module* of grid networks contains several mechanisms, which are shown in Figure 6. They are grid map files, Community Authorization Service (CAS) [14], Virtual Organization Membership Service (VOMS) [15], PERMIS [16], AKENTI [17] and PRIMA [18].

The access control in grids

Grid computing provides not unrestricted sharing, but controlled sharing of resources. Resource owners typically put restrictions on the access to their resources based on the membership, payability, etc. The basic idea of controlling access to shared resources is through authentication. The simplest authentication design is to set up a username and password for the user to join a grid and to keep this information in a *grid map file*. The username is verified with a digital certificate issued by the CA. The drawback of using grid map files is that it is difficult to maintain them for a large number of grid users. More sophisticated mechanisms developed for the access control in grid environments are: CAS, VOMS, PERMIS, AKENTI and PRIMA. Section 4 provides detailed discussion on these mechanisms.

Similarities and differences with Fednets

Although the idea of sharing in grids and VOs is similar to the idea behind Fednets, there are major differences between them. First of all, the administrative domains in Fednets are personal networks, with personal resources and services. It is an overlay network between personal networks of individuals. In grids, the administrative domains are organizations and therefore it is

an overlay network formed between organizations. Second, personal resources are mostly portable and battery-powered, while in grids the resources are big scale computing and storage facilities. Third, Fednets are formed on demand for temporal situations. On the contrary grid applications and projects are set up on a long-term basis to solve complex problems with long-term goals. Forth, Fednets have a dynamic nature, i.e. its constitution dynamically changes over the time, while grid networks have a static nature, with static constituent parts. Fifth, the applications of a Fednet have relatively smaller scope in comparison with grids and VOs. For example, Fednet can be formed for the Internet access sharing, file sharing, printing, display, storage, games, and entertainment, while grids and VOs are formed to solve country-wide or international problems, such as weather forecasting, air pollutions, human genome studies, etc. And sixth, is the way how the management is done. In VOs there is one (or more) professional system administrators to manage the VO, while in Fednets, the user is not a professional and should preferably not be bothered with any management/configuration task.

3.2 Secure Virtual Enclaves

SVE [19] is a middleware infrastructure that allows multiple organizations to share their distributed application objects respecting organizational autonomy. The goal of the SVE is to provide a restricted access to the resources and information databases of organizations. Controlled collaborative computing in SVE is based on using open networks and distributed application technologies, such as WWW, CORBA, Java, Active X and combinations with legacy applications.

The SVE was meant to be used in collaborative computing scenarios, such as:

• In military environments, joint task forces might share selected information and applications for distributed collaborative planning.

• In disaster or incident response teams, various government organizations and corporate units rapidly form a team. They share information in a limited way that is beyond sharing in ordinary settings.

• In business environments, corporate units share information with outside organizations without allowing general access to sensitive corporate data, only allowing authenticated controlled access to a subset of data.

Functional architecture of Secure Virtual Enclaves

An *enclave* is a set of resources (computers and networks) of an organization, which belongs to the same security domain. One or more enclaves form a SVE by joining with a subset of their resources. SVE identifies a distributed collection of selected resources, along with the principals that are authorized to access those resources. Principals are the persons, servers or programs. Figure 7 illustrates a functional architecture of SVE.

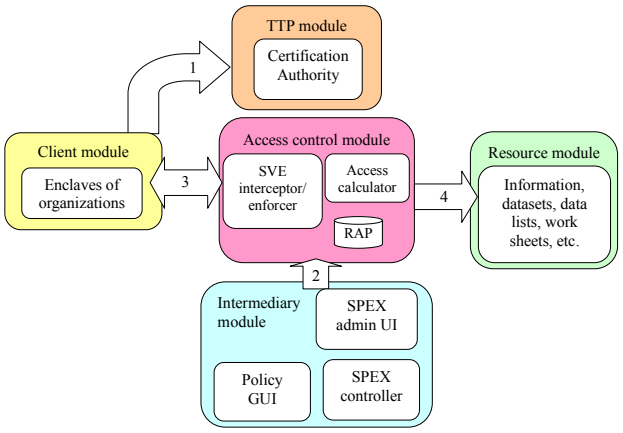

Figure 7. Functional modules in SVE architecture

The *client module* contains the enclaves of different organizations, which are the members of SVE.

The *TTP module* contains a Certification authority that issues X.509 certificates for enclaves.

The *resource module* contains application objects of enclaves, e.g. data lists, work sheets, corporate data and information.

The *intermediary module* is responsible in management and operation of SVE. It contains SVE Policy EXchange administration graphical user interface (SPEX GUI), SPEX controller and Policy GUI. The management of an enclave is carried out by the enclave administrator who administers SVE operation, e.g. initiates joining, leaving and creating the SVE. For this task, the administrator gives commands to the SPEX controller through SPEX Administration GUI. The Enclave can join several SVEs. The administrator of the enclave is also responsible for defining and maintaining the access policies for local enclave resources via the Policy GUI. SPEX controller in its turn propagates these policies within the local enclave to the SVE policy enforcement components (i.e. SVE interceptor/enforcer) and to other SVE member enclaves.

The *access control module* contains SVE interceptor/enforcer, Access Calculator and resource access policies (RAP).

Access control in Secure Virtual Enclaves

While sharing it is important for enclaves to keep autonomy, so that the control over the resources is kept in each enclave locally. Access control policies are not propagated among enclaves. The enclave has the full control over its resource access policy. Therefore, the access control to the resources is done within each enclave

locally, while the administration and maintenance within the SVE is done by all enclaves together.

Enclaves authenticate each other using certificates of a Certification authority (Figure 7, arrow 1). Each local administrator determines the access to the local resources granted to the community by defining the enclave's resource access control policies (arrow 2). A request of the user from another enclave is received at the SVE interceptor (arrow 3), which queries a local Access Calculator for an access decision. Then the Access Calculator evaluates the resource access policies to grant or deny the access to the resources of the enclave. The SVE enforcer then enforces the decision by either allowing the request to proceed as usual (arrow 4), or dropping the request and returning an error message to the client. The access rights are derived by the Access Calculator in four steps: domain derivation, type definition, access matrix check and constraint check. The SPEX controller provides asynchronous policy updates to local access calculators.

In SVE access authorization is role-based and the access is granted equally to all local and foreign principals, which are represented by a domain. For example, if an individual acts in the SVE as an engineer, then he belongs to an engineer's domain. He has the rights assigned to this role to access the SVE resources regardless of his location and the location of the resources he is accessing. The autonomy of the enclave is provided by having its local policies to its resources and having the opportunity to withdraw any resources any time from the SVE. If any of the collaboration partners is found untrustworthy, the enclave can immediately modify its local policy components and update its Access Calculators.

Similarities and differences with Fednets

We can see the following differences among Fednets and SVE. First, the administrative domains in SVE are organizations. Second, the resources in SVE are application objects, the information and datasets e.g. worksheets, data lists that belong to distributed applications such as WWW, CORBA, Java, Active X. Third, the SVE applications are set up for a relatively long time to support inter-organizational activities, e.g. working on the common data lists. Forth, the applications of SVE have a bigger scope, they are meant for inter-organizational communication, while Fednets are meant for inter-personal communication. The idea behind Fednets is to enable a broad range of sharing personal services among the users on demand.

In addition, there are differences in implementation. Interceptor and enforcer in SVE are functionalities that act as intermediaries between external clients and internal servers of the organization. They are implemented at the enclave gateways or as server modifications. In Fednets, these functionalities are implemented as Fednet agents and service proxies at the gateways of the PNs. Furthermore, in Secure Virtual Enclaves (SVE), the SVE administrator defines the resource access policies of the enclave, initiates joining, leaving and creating the SVE. Its functionality is comparable with the PN owner, with the difference that the PN owner manages his/her own resources, while the SVE administrator manages the resources of the enclave that belongs to one organization. Moreover, in SVE a new enclave can join the SVE through voting if only the majority of the enclaves agree on that. Each enclave maintains the list of trusted collaborators, i.e. enclaves of other organizations. Consequently, no anonymity is supported in SVE. Fednets, on the contrary, depending on its goal and the type of its applications can have also anonymous nature, in which the members do not need to know who is in the Fednet.

3.3 Peer-to-peer file-sharing networks

A P2P network is a collection of distributed computers where each computer is called 'a peer' and shares resources and services with other peers. Peers have equal responsibilities and capabilities in providing/consuming the services.

The examples of most popular P2P applications are Napster, Gnutella, Fasttrack, Morpheus, Freenet and Kazaa.

The functional architecture

Figure 8 shows the functional architecture for a typical P2P file-sharing network.

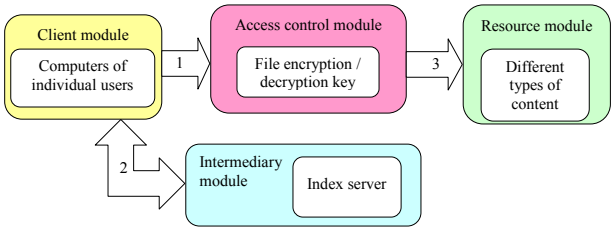

Figure 8. Functional modules of P2P architecture

The *client module* contains distributed computers of users, which are called peers.

The *resource module* contains different types of content, such as data, multimedia and other types of information. Each client computer stores the content that it shares with the rest of the P2P network.

The *intermediary module* represents the index server. The architecture of a P2P network can be decentralized, i.e. without a central index server or hybrid, i.e. with a central index server that maintains an index of the metadata for all files in the network. More specifically, a central index server maintains:

• A table of registered user connection information (IP address, connection bandwidth, etc.)

• A table listing the files that each user holds and shares in the network, along with metadata descriptions of the files (e.g. filename, time of creation, etc.)

In pure P2P network, peers contact each other directly (see Figure 7, arrow 1). In hybrid P2P network, a computer that wishes to join the network contacts the central server and reports the files it maintains (arrow 2).

The *access control module* contains the mechanism that uses file encryption and decryption keys. Having obtained the necessary information about the location of the required file, a peer requests the access to the file and using the file decryption key accesses the file (arrow 3).

Access control in P2P file-sharing networks

Typical P2P file-sharing systems do not emphasize on the access control. The primary objective in P2P networking was enabling free sharing between peers. Therefore they apply a simple access control mechanism, which is illustrated in Figure 8 as a file encryption and decryption mechanism. The authorized readers have the decryption key and the authorized writers have the signing (encryption) key. In Plutus [20] the reader receives a file-signature verification key, while the writers have a file-signing key. When the user wants to access the file to read or to modify it, he must have a key from the file owner. When the user wants to write, he must obtain the write token from the file owner. Using this token the writer can authenticate himself to the file server. The major drawback of this approach is the lack of efficient user revocation system. This brings the problem of re-encryption of large amount of data with a new key, when the reader leaves.

In Freenet [21] the files are encrypted with a random encryption key and the key is stored together with the file's identifier. This implies that any reader can access the file.

Similarities and differences with Fednets

PNs in a Fednet cooperate and share resources in a peer-to-peer manner and therefore, a Fednet is a peer-to-peer network of PNs. Consequently, there are similarities between Fednets and P2P file-sharing networks. First, the types of resources that are shared in Fednets and P2P networks are personal. Second, Fednets and P2P networks have high dynamism, i.e. the participants join and leave the network dynamically. Third, Fednets and P2P networks are both formed for a temporal sharing of resources.

However, there are also some differences between Fednets and P2P file-sharing networks. First, the administrative domains in Fednets are personal networks. Second, the scope of Fednets is broader than a file-sharing. Fednets can be created for a variety of applications for different purposes: emergency networks, learning environments, entertainment and business applications.

3.4 Wireless Community Networks

One of the possible applications of a Fednet is sharing the Internet access. Here we briefly compare Fednets with several other technologies for sharing the Internet access, such as Wireless community networks (WCN), P2P wireless networks confederation (P2PWNC) [22] and FON [23].

WCN is a development of interlinked community networks using wireless technologies. The goal of the WCN is to provide Internet access in areas where the conventional connection services are expensive or not available. WCN was developed by the Center for Neighborhood Technologies [4] to deliver low-cost, high-speed broadband access to homes, small businesses and community-based institutions. To join the WCN, a wireless networking equipment in a water-proof enclosure is installed on rooftops of the community, homes, apartments and other community buildings. This equipment is a wireless router running the mesh routing software. When the computer is connected to the router, it allows accessing the wireless community network. WCN is a mesh network, the wireless access points are interlinked to each other providing multiple and redundant paths, which makes the network robust to failures and damages.

The functional architecture

Figure 9 depicts the functional modules of WCN.

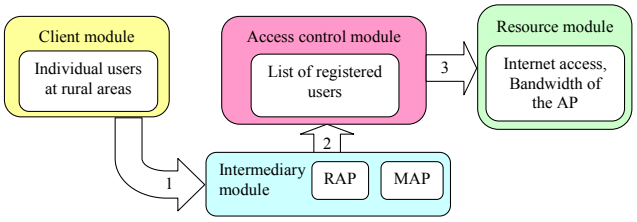

Figure 9. Functional modules of WCN architecture

WCN uses a mesh network to provide high-speed internet access to members of local communities. WCN consists of distributed WAP that belong to different owners. The *client module* contains individual computers that belong to different users.

The *resource module* contains the bandwidth of the Access Point to provide the Internet access.

The *intermediary module* contains Wireless access points: Main access points (MAP), with the direct connection to the Internet and Repeater access points (RAP), which pass the signal from a user until it reaches the Main access point. All access points are connected with each other in a mesh network and have the ability to wirelessly associate with each other without a landline

connection between them. To connect to the Internet at least one Main access point is needed. The rest of the access points need to be within the signal range of another access point.

The *access control module* contains the filtering mechanisms based on MAC addresses at the access points or on higher levels based on the list of registered users.

Access control in Wireless Community Networks

WCN uses a simple access control mechanism, which is done at the wireless routers by configuring and MAC address filtering. A client attempting access must have its MAC address listed on an internal table of the wireless router (Figure 8, arrows 1 and 2). If so, it can be permitted to associate with the access point (arrow 3). In case the access point is a repeater, the traffic of the client will be forwarded to the next access point, in case the access point is the main access point (i.e. directly connected to the Internet), the client will get connected to the Internet.

Similarities and differences with Fednets

There are the following differences between WCN and Fednets. First, in WCN the administrative domains can be heterogeneous, e.g. individuals, institutions and organizations, while in Fednets the administrative domains are personal networks. Second, WCN have a relatively static backbone network, with static access points to the Internet. Participants join the network forming a mesh network on top of it. While a Fednet is a dynamic network, its composition, topology and the point of attachment of its components to the Internet can change over the time. Third, WCN are tailored for a specific application, i.e. sharing the Internet access, while sharing the Internet access is one of the possible applications of Fednets. Therefore WCN can be seen as a special case of Fednets.

3.5 P2P Wireless Networks Confederation

P2PWNC is a community of administrative domains that offer wireless internet access to each others registered users. It is a system that is built on WCNs and enables roaming of the users between WCNs based on incentive techniques. Reference [22] proposes a P2PWNC protocol. The goal is to simulate the participation in the WCN and the provision of 'free' Internet access to mobile users in order to enjoy the same benefit when mobile.

The functional architecture

Figure 10 illustrates the functional modules of P2PWNC. The *client module* contains institutions, service providers and operators.

There is no *TTP module* in P2PWNC. The system uses a reciprocity scheme, which does not require registration with authorities, and relies only on uncertified free identities and public/private key pairs.

The *resource module* contains the Internet access, bandwidth of the AP to access the Internet.

The *intermediary module* consists of the key entities in the P2PWNC, i.e. Domain Agents. Each independent domain maintains one Domain Agent, which has a unique logical name within the P2PWNC system. Domain Agents form a P2P network with each other. The Domain Agent has several functions, such as: name-service, authentication, accounting, consumer and provider strategy, service provisioning.

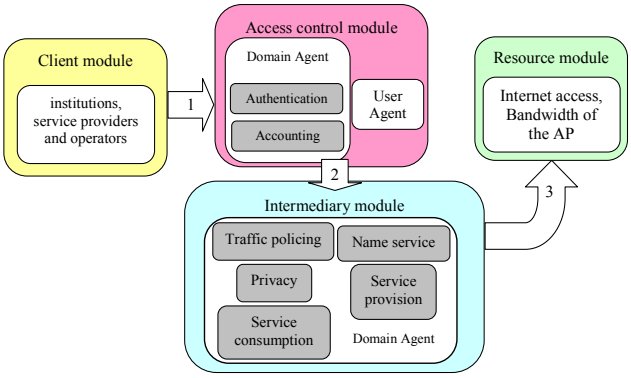

Figure 10. Functional modules of P2PWNC architecture

The main purpose of the Domain Agents is to eliminate the administrative overhead of roaming agreements. Instead of the roaming agreements, the Domain Agents use a token-exchange accounting mechanism. According to this mechanism a consumer Domain Agent transfers tokens to the visited Domain Agent in compensation for the used resources. So the central design goal is to build into the system incentive mechanism based on the reciprocal behavior: consumption and provision should be balanced.

The *access control module* contains Domain Agents and User Agents, which are explained in the next sub-section.

Access control in P2PWNC

The P2PWNC users can be registered with several domains, but they should have a unique identifier, in the form of '*user_at_domain'* for each account. For identity privacy, the users are allowed to have pseudonyms for each account.

The system uses a reciprocity scheme. Users sign digital receipts when they consume service. The receipts form a graph, which is used as input to a reciprocity algorithm that identifies the contributing users. Although the users can easily get free identities, the new users must first contribute to the system before using the services. The

users are divided into teams. The contribution and consumption is evaluated on a team base. Therefore the scheme has a free-riders problem.

In order to use the services the users input their user identifiers and associated security credentials to user agents (Figure 10, arrow1). The user agents carry out the authentication procedure in cooperation with the Domain Agent. The users may use different identifiers, choosing from the Domain Agent who has a higher token level. When the access is granted (arrow 2), the Domain Agent coordinates the wireless service provisioning and consumption for its domain.

In every domain (e.g. institution, service provider) there is an associated group of registered users. The Domain Agents maintain the list of its own registered users. The Domain Agent is an economic agent within the P2PWNC, it is responsible for the coordination of bandwidth consumption by the registered users of the domain in a roaming scenario and for the coordination of bandwidth provisioning by the domain itself.

Similarities and differences with Fednets

There are the following differences between P2PWNC and Fednets. First, in P2PWNC the administrative domains can be heterogeneous, e.g. individuals, institutions, service providers and operators. Second, since it is built on top of WCN, P2PWNC have a relatively static backbone network, with static access points to the Internet. Third, P2PWNC are tailored for a specific application, i.e. sharing the Internet access. Therefore similar to WCN, P2PWNC can be seen as a special case of Fednets.

3.6 FON

FON [23] is a system of shared wireless networks. The FON's members share their WiFi with others, in return they can freely access all other FON wireless access points that are available all over the world. This is achieved by sharing the bandwidth of their special routers, called La Fonera routers.

The functional architecture

Figure 11 illustrates the functional modules of FON.

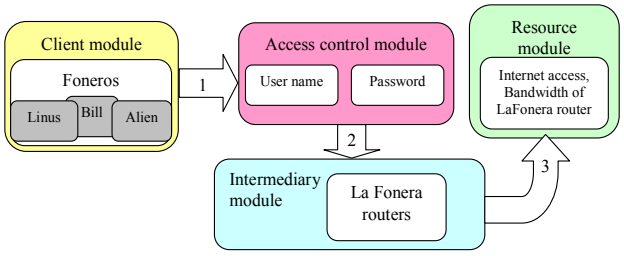

Figure 11. Functional modules of FON architecture

The *client module* contains the users of the FON network, called Foneros. Foneros are distinguished into three types based on their membership: Linuses, Bills and Aliens, as shown in Figure 10. Linuses and Bills are registered users of the FON network. They share their home WiFi hotspot with the FON network and can use any FON hotspot for free, can roam the FON network for free. Aliens do not share their bandwidth but they can use the FON network by purchasing daily passes. FON passes are similar to prepaid cards. Aliens can purchase FON passes by detecting a FON signal and connecting to FON or by sending SMS through their mobile phones. Aliens can also get 15 minutes of free WiFi access to any FON spot per day.

The *resource module* contains the bandwidth of La Fonera routers, which is shared between Foneros. FON consists of distributed La Fonera routers that belong to different owners. While roaming, the user can connect to internet by means of these routers through the WiFi available in the vicinity. Using laptops or WiFi enabled devices, such as phones, cameras, Foneros can access any FON spot around the world.

The *intermediary module* consists of La Fonera routers of Foneros.

The *access control module* in FON contains the authentication mechanism based on FON usernames and passwords.

Access control in FON

FON software includes a level of access control, which could be beneficial for WiFi in open network with little or no security, also beneficial for service providers. All registered members of the FON network have FON username and password. Once an Alien has registered with FON (Figure 11, arrow 1), using its user name and password, it can be granted the access to the Internet through La Fonera router (arrow 2), which provides with the part of its bandwidth for the traffic of the Alien (arrow 3).

Aliens can also use their FON username and password to access their own personal User Zone. In the User Zone, the Alien can retrace her WiFi activities through the FON Community. Including seeing how many FON Passes they have purchased, used and how many they still have remaining.

Similarities and differences with Fednets

There are the following differences between FON and Fednets. First, in FON the administrative domains are individuals with *La Fonera* routers. Second, FON have a relatively static backbone network, with static access points to the Internet. Participants join the network forming a mesh network on top of it. Third, FON are tailored for a

specific application, i.e. sharing the Internet access. Since sharing the Internet access is one of the possible applications of Fednets, FON can be seen as a special case of Fednets.

3.7 Comparison of related technologies

In this section we summarize our survey on resource sharing technologies and paradigms in Table 1.

Table 1. Comparison of Fednets and related technologies

Technology	Definition	Typical applications	Administrative domains	Scale of the system	Shared resources	Access control mechanisms
Grids	Hardware and software infrastructure to allow coordinated resource sharing and problem solving	• Medical/Healthcare • Bioinformatics • Nanotechnology • Engineering • Natural Resources and the Environment	Individual users, organizations, Virtual Organizations	Number of services, participants and geographic scale is **large**	Hardware, software, computer processing power, big scale computing and data storage facilities	**Centralized** access control (grid mapfile, CAS, VOMS, PERMIS, PRIMA), **distributed** access control at stakeholders (AKENTI)
Secure Virtual Enclaves	An infrastructure implemented in middleware to allow multiple organizations to share their distributed data and application objects	• Military environments • Disaster or incident response teams • Business environments	Organizations	Number of services, participants and geographic scale is **small**	Distributed application objects, information and data of organizations	**Distributed** access control to the SVE, **distributed**, local access control to the SVE resources
P2P file-sharing networks	A collection of networking nodes where each node has equal responsibilities and capabilities in providing and consuming the services.	File and content sharing (e.g. Napster, Gnutella, Fasttrack, Morpheus and Kazaa)	Individuals users, organizations	Number of participants and geographic scale is **large**. Number of services is **small**.	Files, information, media and entertainment	**Distributed** access control with encryption/decryption keys
WCN	Interlinked community network using wireless technologies	Low-cost broadband connectivity and related opportunities such as job searching capability and skill development, to underserved households, community groups, and small businesses.	Individuals, organizations	Number of participants is **large.** Number of services and geographic scale is **small**.	Wireless Internet access is shared, by means of sharing access point repeaters for traffic forwarding between neighbors	**Distributed** access control at Wireless access repeaters and Wireless access points by MAC address filtering, or on higher levels with the list of registered users
P2PWNC	System that is built on WCNs and enables roaming of the users between different WCNs based on incentive techniques	Universities, residential hotspots, private companies that provide WLAN access to employees, mobile operators offer wireless internet access to each others registered users.	Individuals, organizations, service providers, operators	Number of participants and geographic scale is **large.** Number of services is **small**.	Wireless Internet access is shared on the basis of reciprocity algorithm.	**Centralized** access control to the domain carried out by the Domain agents

Table 1 (continued). Comparison of Fednets and related technologies

Technology	Definition	Typical applications	Administrative domains	Scale of the system	Shared resources	Access control mechanisms
FON	A system of shared wireless networks	Sharing personal WiFi with others, in return to the possibility to freely access all other FON wireless access points.	Individuals, called 'Foneros' with their home WiFi and La Fonera routers	Number of participants and geographic scale is **large.** Number of services is **small.**	Part of the bandwidth is shared to give the Internet access to others	**Centralized** access control with password and user name submitted to the FON special site
Fednets	Ad-hoc, temporal, secure cooperation of independent Personal networks	• Family networks for entertainment and remote file sharing • Ad hoc network during the Project-meetings • Inter-vehicle networks to share information on the road conditions • Emergency, disaster relief and rescue/recovery networks to rescue people • Health-care and hospital networks • Distance learning networks and virtual classrooms • Commercial resource sharing • Online gaming • Networks for information services	Individuals with their personal networks	Number of services, participants and geographic scale is **large**	Personal resources and services (audio-video, storage, printing, processing, routing, internet access etc.)	**Centralized** access control to the Fednet, **distributed**, local access control to the Fednet services at PNs

We observe from the table, that all systems, except SVE, may have administrative domains composed of individuals participating with various types of resources. This observation suggests that the group networking formed between individuals is of particular interest.

As can be seen from Table 1, all systems differ in their scalability. Among all others, Fednets and grids can have a large number of service types, participants and their geographic scale is large. In P2P file-sharing networks, P2PWNC, and FON despite their large number of participants and large geographic span, the number of shared service types is small. In WCN, the geographic scale is also small as well as the number of shared services. Furthermore, in SVE, in addition to the geographical scale and number of shared services, the number of participants is also limited, since the SVE is a closed, controlled collaborative network.

Furthermore, our survey reveals that each system is designed for a particular application for sharing specific types of resources. Access control is the most essential component in service provisioning in a cooperative environment. The type of application and shared resources are important when choosing for specific access control architecture. The majority of systems deploy distributed access control to shared resources and services.

From the surveyed technologies grids are of particular interest, because based on the structure, Fednets can be seen as '*a grid of personal networks*'. In addition, grids and Fednets have similarities with regard to their scalability in geographical span, number of service types and number of participants. Moreover, grids and Fednets combine both centralized and distributed approach in their access control architectures. This determined our motivation to survey a number of access control mechanisms used in grids environments. Section 4 is devoted to this topic.

4. ACCESS CONTROL TO SHARED RESOURCES

Further in our survey we focus on different approaches for access control in grid environments. We analyze them

based on the IETF Authentication Authorization and Accounting framework [24] (AAA), which is the authorization framework for Internet resources and services.

4.1 Generic AAA Framework

Basic conceptual entities

The basic conceptual entities that may take part in the authorization process are illustrated in Figure 12. They are:

- Users,
- User home organizations, with its AAA Server,
- Service providers, with its AAA Server and Service Equipment.

AAA server is a network server used for access control. The user home organization based on the user agreement checks whether the user's request for a service should be permitted. This task is performed by the *AAA server of the user home organization*.

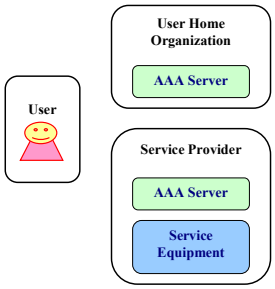

Figure 12. The Basic Authorization Entities

When the user's service request gets to the service provider, the *AAA server of the service provider* authorizes the user access to its service based on the agreement with the user home organization. The service equipment of the service provider is the one that provides the service to the user.

The framework defines several authorization message sequences to achieve trust between the user and the service provider. There are two cases: a single domain case and a roaming case. In a *single domain case* the user, the service provider's AAA server and the service provider's service equipment take part. No user home organization is involved. The *roaming case* explores the situation where the organization that authenticates and authorizes the user is different from the organization providing the services. This means that in this case, both user home organization and service provider are involved with their AAA servers.

In group-oriented networks individual users communicate with each other without involving the user home organization. Therefore, further we describe the sequences of authentication message flow for a single domain case. There are three message exchange sequences defined in AAA framework between the user and the service provider. They are push, pull and agent sequences. We briefly describe them here.

The Push sequence

Figure 13 depicts the push sequence. The user gets a ticket or a certificate from the service provider's AAA server (arrows 1 and 2) and then presents it to the service provider's service equipment together with the request (3). The service equipment uses the ticket to verify that the request is approved by the service providers AAA server. By the successful verification, it grants the access (4).

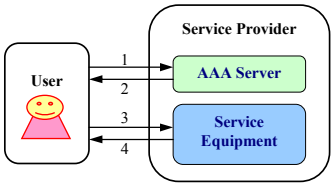

Figure 13. Push sequence of authorization message flow

The Pull sequence

This sequence is typically used in dial-in applications. The user sends the request to the service equipment (arrow 1), which forwards it to the service provider's AAA server (2), this is illustrated in Figure 14. The AAA server evaluates the request and returns the response to the service equipment (3). The service equipment sets up the service and notifies the user that it is ready to serve (4).

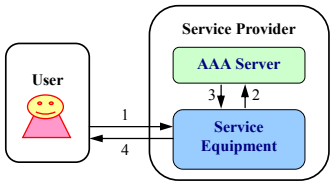

Figure 14. Pull sequence of authorization message flow

The Agent sequence

In this sequence the service provider's AAA server acts as an agent between the user and the service equipment, as is depicted in Figure 15. It receives the request from the user and sends authorization and configuration information to the service equipment.

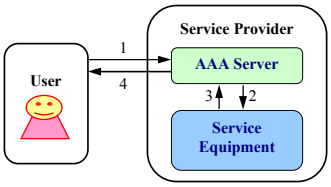

Figure 15. Agent sequence

AAA framework and policy framework

IETF RFC 2904 [24] also describes the relationship of authorization and policy. It extends the policy framework presented in IETF RFC 2753 [25] to support policy across multiple domains. RFC 2904 introduces the components such as the *Policy Decision Point (PDP)*, which makes access control decisions based on policies; and the *Policy Enforcement Point (PEP)*, which enforces the decisions made by the PDP.

When mapped into the policy framework, the AAA server locates the PDP function. The PEP function is located at the Service Equipment of the Service Provider, as is shown in Figure 16.

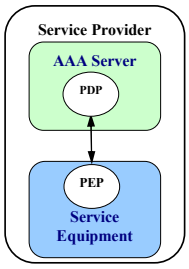

Figure 16. Mapping of policy framework components to AAA framework

With policy extension, the above-mentioned sequences will be as follows:

1. *Push sequence*. The user calls the PDP, the PDP returns the authorization decision and the user submits it to the PEP.

2. *Pull sequence.* The user calls the PEP. The PEP pulls the authorization decision from the PDP and based on this decision it grants or denies the access.

3. *Agent sequence*. The user calls the PDP. The PDP sends the user request along with the authorization decision to the PEP. This way the PDP acts as an agent on behalf of the user.

4.2 Access control architectures for grid environments

A number of different authorization architectures are reported in literature for access control in distributed environments, such as grids. They are based on different approaches, such as Certificates (CAS [14], VOMS [15], AKENTI [17], and PERMIS [16]), Signed assertions (CAS), Capabilities (CAS), Roles (PERMIS and PRIMA [18]) and Policy statements. In this section, we describe these authorization architectures and the control over the access to shared resources based on the message exchange patterns described in Section 4.1.

4.2.1 CAS

To address the scalability of the access to the distributed virtual community's resources and improve the manageability of user authorization, a trusted third party - a community authorization service (CAS) was proposed [14] in 2002. It minimizes the burden of maintaining grid map files (discussed in Section 3.1) by the administrators.

Reference [26] discusses a number of challenges imposed by Virtual Organizations (VO), such as scalability (the cost of administering a VO, adding and removing participants, changing community policy increases by growing of the VO) and complex policy hierarchies. The CAS architecture is built on Public Key Infrastructure (PKI) [37] and Globus Toolkit Grid Security Infrastructure (GSI) [34] and addresses the issues of single sign on, delegation and scalability that arise in Virtual organizations (VO). According to the CAS principles, the community delegates the access granting rights to a subset of its resources to the central authority, i.e. the CAS server. When the client requests to use the resource of the community, the CAS server, based on the policies defined by the communities, decides about the access rights that should be granted to the user. Having taken the decision, the CAS server produces self-signed certificates with permissions to access the resources. The access to the resource will be granted, if the validity of the certificate and the CAS service is proved.

Access control

CAS is an authorization service developed within the Globus project for Grid environments. CAS server acts as a trusted intermediary between the VO users and resources. The resource owners grant the access to the subset of their resources to the VO. The CAS in this VO is a trusted intermediary between the users and resources, which decides who can use the resources. This means that the resource owner delegates the allocation of authorization rights to the CAS server.

First, the user becomes a member of the community. This process corresponds to the first-level access control. Afterwards, the user can request a service by contacting the CAS server, which delegates the rights to use the requested service that belongs to the community. The rights are in the form of capabilities. They are embedded in GSI proxy credentials as policy assertions written in SAML [36] and signed by the CAS server. Having obtained the proxy credentials, the user presents them to the resource to access the resource on behalf of the community. This process corresponds to the second-level access control.

Authorization message flow

Figure 20 illustrates the resource access using CAS. The CAS server stores the policies which contain the list of objects and their rights. This information is included in the extension of the delegated proxy certificate.

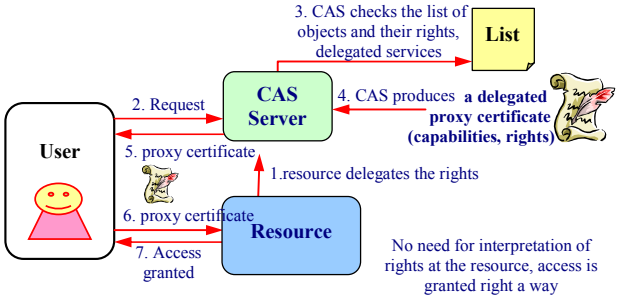

Figure 20. Resource access using CAS. Push sequence

The requestor contacts the CAS server to get a delegated proxy certificate that includes the information about what resources can be accessed and to what extent. The delegated proxy certificate is a short-lived X.509 certificate. To access the resource the user submits this proxy certificate to the gatekeeper of the resource. This certificate is enough to access the resource, there is no need to submit the attribute certificate (attribute value bindings to a user) to the gatekeeper. This means that the granting the access rights to the community resources is done in advance, before the user contacts the gatekeeper. This offers some relief to the resources from interpreting the rights of the users. This approach corresponds to a push sequence of the AAA framework.

The CAS administrator is responsible for adding each user to the appropriate group of the community. The CAS administrator can delegate to others the administration of subsets of objects. Here note, that the member administration is centralized, with the delegation possibilities.

Observations

The CAS approach has the following drawbacks. CAS issues a proxy certificate instead of the attribute certificate and the authorization information is included in the extension of the proxy certificate. The extension includes the restriction on the access rights, placing specific limits on the rights of the user. When the service receives the certificate it should check the extension to know the restrictions to the access rights. This approach is not efficient, because it requires modification at the service side. Furthermore, CAS does not support roles, but permissions to do actions. Consequently, the CAS server records permissions and does not record the roles, because the roles of the users or the groups of the users are not defined.

The drawbacks of the CAS approach also include the requirement for enforcement within the application code, Policy Enforcement function is built into the grid service application, so there is a need for a trusted application code. Moreover, a group owned infrastructure component – CAS server and a community administrator are required. This also raises scalability problems.

4.2.2 VOMS

VOMS (Virtual Organization Membership Service) [15] is another implementation of the access control to the grid resources. It provides the authorization information about members and an authentication and authorization service within the VO. It is developed in the European Data Grid project [27].

VOMS delegates the authorization of the users to the managers of the VO and allows managing user roles and capabilities centrally. The main difference between the approaches in CAS and VOMS, is that in VOMS the resources should carry out the interpretation of rights based on the membership certificates of the users, while in CAS the resources do not need such interpretation, since the certificate is enough to access the resource.

Access control

VOMS provides the authentication and authorization services, we can map its architecture to the AAA framework. The architecture of the VOMS uses the authentication and delegation mechanisms provided by Grid Security Infrastructure (GSI). In VOMS the authorization to the resources is based on policies, which are written by the VOs representing their agreements with the resource providers. VOMS embeds attribute certificates in GSI proxy credentials that specify group and VO membership information for access to community resources. The requestor contacts the VOMS server to get a credential and submits this credential to access the resource.

The authorization information is separated into two types, because this information controls the resource access in VO from different perspectives, with different roles.

First, the information regarding the relationship of the resource user to the VO, for example its membership, belonging to which group and etc, this information is stored at the server managed by the VO;

In addition, the information regarding the relationship of the resource user to the resource provider, for example, what the user can do at the resource and etc., this information is stored locally at the resources.

Here we encounter two different types of access control, which reminds the two-level-access control that we

defined for Fednets. The first is the access control to the community as a member. The second is the access control to the resources of the community.

Authorization message flow

In the first version, VOMS was a system for dynamically creating grid map files from LDAP directories containing the details about the VO users. A grid map file contains the list of authenticated distinguished names of the grid users mapped into the corresponding local user accounts names. The resources could periodically retrieve them to make authorization decisions, as is shown in Figure 18. This approach corresponds to a pull sequence of the AAA framework.

This approach maximizes the work of the resource administrator, because he must first pre-configure the grid application with the names of every VO user, if the user is allowed to access the grid resource. This approach is not scalable and not flexible, and the administrative task can not be distributed throughout the VO.

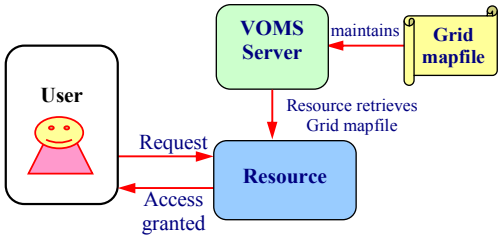

Figure 18. Resource access using VOMS. Pull sequence

Later version of the VOMS issues a short-lived X.509 Attribute Certificate [28] for the VO users which they can submit to the resource. The certificates are signed by the VOMS server. The certificate contains the information about the users, such as local account name, to which group does the user belong to, what roles the user is assigned within this group and some other privileges and capabilities. Therefore the resources do not need to retrieve a grid map file, since all necessary information to verify the identity is included in the certificate. However, the resource needs software to interpret the attribute certificate. This approach corresponds to a push sequence of the AAA framework and is illustrated in Figure 19.

Observations

VOMS has a community centric attribute server that issues authorization attributes to members of the community, similar to CAS server. But in CAS the subjects have a group credential, while in VOMS subjects authenticate with their own credentials.

The drawback of the VOMS approach is that the resources should carry out the interpretation of rights based on the membership certificates of the users. This puts a burden to the resources, since the resources should know how to do the interpretation. Furthermore, VO administrator maintains a centralized database to add each VO user and gives users appropriate attributes needed to access the VO resources. This approach has scalability problems, in managing joining and leaving members of the VO, their access rights and roles within the VO, since it is based on centralized model in user management.

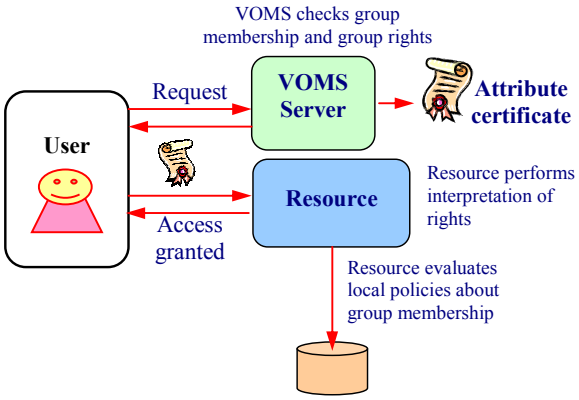

Figure 19. Resource access using VOMS. Push sequence

4.2.3 AKENTI

AKENTI [17] is a distributed policy-based authorization system for grid environments and is designed for authorizing the access on web resources, such as web sites. AKENTI addresses the issues of providing restricted access to resources that are controlled by multiple stakeholders. The Stakeholders in AKENTI are the parties with authority to grant access to the resource.

Access control

AKENTI does not require any central authority to enforce the access control to the resources. AKENTI uses distributed policy certificates in XML format. These certificates are signed by the stakeholders from different domains, who decide on the access control to a resource and place its own restrictions to the usage of the resource. AKENTI makes a dynamic authorization decisions based on supplied credentials and applicable usage policy statements defined in AKENTI policy language. For expressing policies and certificates AKENTI uses XML, although the first version of AKENTI used a simple keyword language.

Digitally signed certificates specified in AKENTI can contain the following information:

• identity authentication information,
• attribute certificates,
• use condition certificates (the list of users owning the attributes and the explanation of which attributes are needed for which access rights),
• policy certificates (include the list of trusted CA and stakeholders and the links from where the use-conditions and attribute certificates can be retrieved).

Authorization message flow

AKENTI can use both pull and push models of authorization information flow. Figure 22 illustrates the push model. In AKENTI there can be multiple stakeholders participating and administering one resource. A new user should contact the stakeholder to be added to the list of resources and appropriate policy files. This process corresponds to the first-level access control.

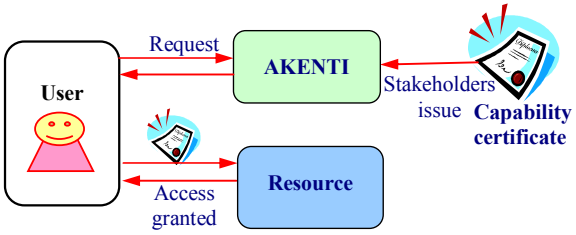

Figure 22. Resource access in AKENTI. Push sequence

To access the resources the requestor is required to present some credentials to AKENTI authorization system. First, AKENTI authenticates the user based on the X.509 identity certificates, then it uses the attribute certificates belonging to the user and use-condition certificates (regarding the resources) and makes a decision on the access rights of the user. The system then provides the client with a capability certificate. The client contacts the gatekeeper of the resources by presenting this capability certificate.

Observations

In AKENTI, the resources are accessed based on the resource access policies. The evaluation of policies and granting the access rights are done after the gatekeeper of the resource is contacted, whereas in CAS, the rights are already included in the capabilities issued by the CAS server.

AKENTI does not require a centralized authority to run the access control policies. Although the resources are owned and controlled by multiple stakeholders, the access to the resources is controlled by means of distributed policy certificates, without a central authority.

The drawback of AKENTI is that it gives the user the total access, not a fine-grained access to the resources. As a consequence, the user's access can not be limited during the sessions. Moreover, AKENTI does not link the identities with groups or roles but with permissions, therefore the user can not specify the role that he wants to use during the access. As a result, the attributes in AKENTI cannot form a role hierarchy.

AKENTI specifies separately the authorities for performing authentication and for creating and signing attribute certificates. This introduces another drawback that the resources must know about the CA of each user, which causes scalability problems. In contrast, in CAS the resources must know only the CA of the CAS server.

4.2.4 PRIMA

PRIMA [18], [29] is a system for Privilege Management, Authorization and Enforcement in grid environments to support dynamic, spontaneous, short-term collaborations of small groups of grid users. While CAS and VOMS are systems that rely on central servers for the authorization service in grids, PRIMA is a fully decentralized system that enables direct trust establishment among participants. It supports dynamic authorization policies for grid resources. PRIMA distinguishes from other authorization systems by its support for the creation, configuration and management of user accounts on demand. Other grid security services support only static accounts, which limit the scalability, hinder collaboration and creates security holes through static accounts.

Access control

PRIMA focuses on access control for small and dynamic working groups. The system uses fine-grained privileges as fine-grained access rights. It uses privileges to enforce policy statements. The subject privileges are issued by resource owners and administrators, or group and project leaders. Both privilege statements and policy statements are expressed in XACML and are embedded in X.509 Attribute Certificates [28], [30], so the X.509 Attribute Certificate carries privilege and policy statements.

In PRIMA the privilege attributes are issued by individual attribute authorities, such as project leaders, resource owners, administrators, but not community servers like in VOMS or CAS.

Regarding the levels of the access control, the first-level is carried out when the entity becomes a group member. The PRIMA is used for the second-level access control.

Authorization message flow

PRIMA implements a hybrid model of authorization message flow as is shown in Figure 21. Although the user pushes the acquired privileges to the resource (i.e. a push

sequence), still the resource requests the access control decision from a PDP function based on these privileges. Therefore this part of the message exchange corresponds to the pull sequence.

The privileges are collected by the Policy Enforcement Point (PEP) and are checked against the access control policies at the Policy Decision Point (PDP). PDP returns an authorization decision to the PEP and a set of recommendations on the actions, for example, setting up a local account based on the valid privileges, file access permissions, network access and etc.

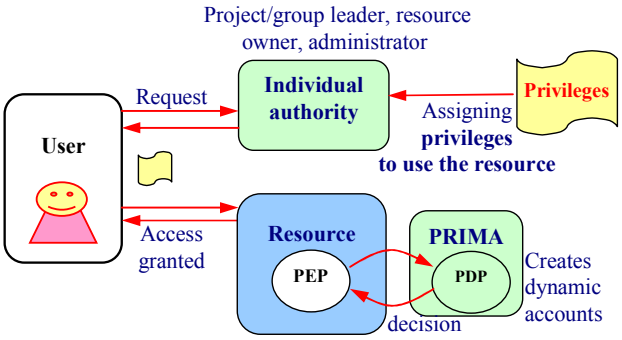

Figure 21. Resource access in PRIMA. Hybrid model

Observations

PRIMA is distinguished from other authorization systems with its support for the creation, configuration and management of user accounts on demand [18], [29]. Dynamic accounts are like dynamic IP addresses. They are taken from the pool of available addresses and returned into the pool when released. The pool of dynamic accounts is created by the system administrator. These accounts do not allow direct login and have minimal rights. When a dynamic account privilege is presented, the system first checks for an existing account matching the distinguished name and optional project identifier, then maps the user to the existing dynamic account. If the user is not found in the map, a new dynamic account is assigned from the pool of available accounts. Before expiring the holder is notified by the privilege revocator. Once the account is expired, it is reset and returned to the pool.

In PRIMA individual privileges can be grouped to form a group, the group of privileges can be applied to a set of users – holders of the roles. PRIMA is different from the Role-Based Access control (RBAC) system, because RBAC system creates the roles and binds the access rights to roles. Besides RBAC focuses on the administrators and resources, while PRIMA focuses on the resource users.

For the implementation PRIMA module is integrated with the Globus toolkit as an authorization component. PRIMA module acts as a PDP and makes fine-grained authorization decisions based on the privileges of the user.

Creating dynamic user accounts on demand allows the users to utilize the resources on temporal basis. The accounts are created based on the privileges of the users assigned by the authorities, such as resource owners, project leaders or administrators. Dynamic accounts can be replaced by static accounts on demand. This gives flexibility in managing the user accounts.

The drawback is that the resources need extra functionality to implement the PDP and dynamic account creation.

4.2.5 PERMIS

PERMIS [16], [31] is an authorization system that implements a Role Based Access Control mechanism for different role-oriented scenarios. A user is granted rights to access a resource based on the authorization policy for the resource, and a set of role attributes that the user possesses.

A user's attributes are stored in digitally signed X.509 Attribute Certificates [28]. Given the name of the user, PERMIS retrieves the user's attributes/roles and makes decisions based on them. The authorization policy, written in XML, expresses which users can be assigned what roles by whom, and what privileges are bound to each of the roles. The XML policy is then inserted in an X.509 Attribute Certificate, signed by the manager who wrote it, and stored in an entry in an LDAP server.

Access control

When an application starts up, its PEP passes to the PERMIS PDP the name of the manager, the location of the LDAP directory, and the unique number of the policy to be used. Each policy is assigned a globally unique number, so that a manager can create different policies to be used in different contexts. Then the PERMIS PDP retrieves the policy X.509 Attribute Certificate from the LDAP directory, checks the signature and the policy number. If both are correct, PERMIS makes the authorization decision evaluating this policy based on the attribute/roles of the user retrieved from the X.509 Attribute Certificate. Therefore, PERMIS is considered as a strong policy engine to control the resource access.

PERMIS consists of two subsystems: privilege allocation and privilege verification. The first issues the attribute certificates and stores them in LDAP, the second retrieves the attribute certificates and the policies on the user roles from a pre-configured list of LDAP directories. In PERMIS the entity that creates policies is called a Source of Authority.

Similar to PRIMA, PERMIS does not include the first-level access control. PERMIS grants the access to the resources based on the roles of the users and the resource

policies. This process corresponds to the second-level access control.

Authorization message flow

PERMIS does not provide an authentication services, but it can work with any authentication system, such as Shibboleth [32], Kerberos [33], PKI [37] or username/password. Given a username, a target and actions, the PERMIS, based on the policy, decides whether the user is granted or denied the access to the resource.

Figure 23 depicts the resource access in PERMIS, which corresponds to the pull sequence of the authorization message flow. However, the users can also push the certificates to the system for verification. Therefore, PERMIS uses a hybrid model of authorization.

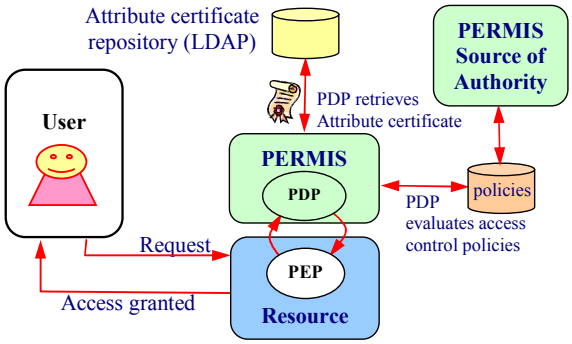

Figure 23. Resource access in PERMIS. Pull sequence

Observations

PERMIS is an alternative to VOMS. Users are given roles and attributes that belong to these roles. The roles and attributes are assigned permissions to access the resources. It is also called a privilege management infrastructure that uses X.509 certificates.

Similar to CAS and VOMS, PERMIS also uses the attribute certificates, which are stored in the repository for attribute certificates. After the user is authenticated successfully, the system retrieves the attribute certificate of the user from the repository. To make decisions, PERMIS processes the content of the policy file and the content of the attribute certificate of the user.

4.2.6 My Proxy

Most Grid Portals (gateways) require that the user delegates the rights to the server to act on its behalf. Normally the Grid resources are protected by GSI [34], which supports such delegation. But web security protocols do not support the delegation function, so this leads to incompatibility between Grid security and Web security. To address this problem the reference [35] proposes an online credential repository system called 'MyProxy', which allows smooth operation of grid portals that use GSI to interact with grid resources. MyProxy is open source software for managing X.509 PKI security credentials, such as certificates and private keys. MyProxy combines an online credential repository with an online certificate authority to allow the users to securely obtain credentials when and where needed.

Authorization message flow

Figure 24 shows the process of accessing the grid resources through web portals using MyProxy credential repository.

Resource access using MyProxy corresponds to the pull sequence of the authorization message flow. By the request of the user to access a grid resource, web portal retrieves user credentials from MyProxy repository. Using these delegated credentials web portal authenticates to the grid resources and provides the user with the access to this resource.

The first-level access control is the process when the user becomes a grid user or a VO member. MyProxy provides credentials that are used for the second-level access control, i.e. to access the grid resources.

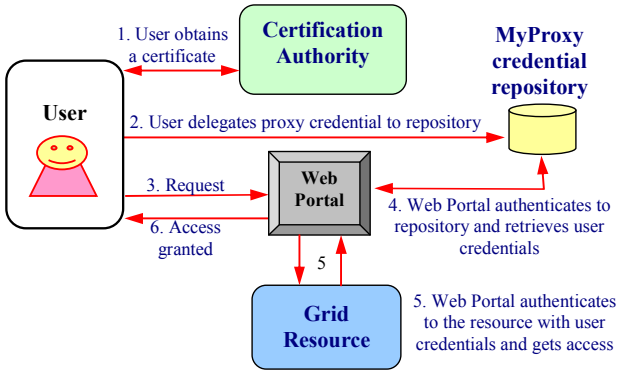

Figure 24. Resource access using MyProxy. Pull sequence

Observations

MyProxy is a system to provide online short-lived credentials to access grid resources. MyProxy supports multiple authentication mechanisms, including passphrase, certificate, Kerberos, VOMS, LDAP and One Time Passwords. The MyProxy CA issues short-lived session credentials to authenticated users. The repository and CA functionality can be combined into one service or can be used separately.

4.3 Summary of access control architectures

In this section we summarize the discussed access control architectures. Table 2 provides an overview of each of the systems, in terms of the entities of their access control architecture, what information these entities use to carry out the access control and how this information is conveyed between these entities.

Table 2. Comparison of access control architectures discussed in this paper.

Access control architecture	Description	entities of the access control arcthitecture	Access control information	Authorization sequence
CAS	An **authorization** system for distributed virtual community resources	CAS server Organizational resources Users	Delegated proxy certificate, roles, policies, capabilities, group credential. Access rights are in the proxy certificates in a form of SAML policy assertions	**Push** The user gets the attribute certificate from the CAS server and then presents this certificate to the resource
VOMS	An **authentication** and **authorization** system for virtual organizations	VOMS server Organizational resources Users	Attribute certificates, policies, group membership, roles in the group	**Pull** When user requests a service, the resource retrieves a grid map file to make authorization decisions **Push** The VOMS server issues a short-lived X.509 Attribute Certificate for the VO users which they can submit to the resource.
PRIMA	Privilege management and **authorization** system for dynamic small groups of grid users	Project leaders, administrators Grid resources Small group of grid users	Privileges, policies, X.509 Attribute Certificates, dynamic accounts. Privileges are embedded in X.509 Attribute Certificates	**Hybrid** The user pushes the acquired privileges from the PRIMA system to the resource (i.e. a push sequence). The resource requests the access control decision from a PDP function based on these privileges (i.e. a pull sequence).
AKENTI	Distributed policy-based **authorization** system for web resources, grids	Stakeholders Resources of multiple stakeholders Users	Capability certificates, policies, use conditions for resources, Mutual authentication using X.509 certificates	**Pull** The user contacts the resource, which calls AKENTI with the user name and the resource name. Then the resources obtain an access control decisions. **Push** AKENTI gives a capability certificate to the user. To access the resource the user presents it to the gatekeeper of the resources.
PERMIS	An **authorization** system with role-based access control system	Source of authority (entity that creates policies) Resources Roles	Roles, privileges, policies, X.509 Attribute Certificates	**Pull** The PDP contacts the attribute certificate repository to retrieve the appropriate certificate and then runs the access control policies. Decision is a Boolean grant/deny response. **Push** Users can push the certificates to the PDP for verification and access rights.
MyProxy	An online **credential repository** to bridge the incompatibility between web- and grid-portals	MyProxy online credential repository Web portal Web resources Users	X.509 Attribute Certificates, delegated proxy credentials, MyProxy credential repository	**Pull** Web portal retrieves user credentials from MyProxy credential repository and grants the access based on them.

As can be seen from the table, most of the architectures use X.509 Attribute Certificates to store the authorization information. Several mechanisms use both pull and push models of authorization message flow (e.g. VOMS, AKENTI and PERMIS). PRIMA deploys hybrid model, since the authorization contains consecutive push and pull sequences. Moreover, all systems use access control policies based on different conditions (e.g. roles, attributes, privileges, capabilities). PERMIS among them is a strong policy engine to control the resource access based on roles and privileges. Access control decisions can be in a simple 'grant/deny' form (e.g. AKENTI), as well as fine-grained access differentiating from total till restricted access to the resource based on various criteria (e.g. PERMIS).

We can recognize some similarities between our approach and other approaches. For example, the Fednet manager issues a *membership credential* which in case of VOMS is a 'VOMS certificate' issued by the VOMS server. To request the service, the client PN presents this membership credential to the service providing PN. Based on the *access control policies defined by the PN owner*, the access control decision is made.

Furthermore, *common services* in a Fednet remind the CAS principles. As was explained in Section 2.3, the common services of a Fednet are accessible to all members of the Fednet upon presenting their 'membership credentials' issued by the Fednet manager. In the case of CAS it is a 'proxy certificate' issued by the CAS server, which grants the access rights to the community resources.

Finally, dynamic access control. For the second-level access control the PN owners define their own *policies* and *access privileges* to allow the access to their personal resources. The membership class is assigned by the Fednet manager based on the previous experiences, the contributions, the reputation or the role of the Fednet member. Different membership class corresponds to different privileges to access the service. Privileges together with the membership class dynamically change the access rights to the Fednet services. This approach reminds the dynamic authorization policies provided in the PRIMA system.

All approaches have their attractive points. For example, the approach taken in VOMS facilitates the management, since there are dedicated VOMS administrators for this task. The administrators provide the user with the authorization credentials that are interpreted by the resource. Furthermore, the approach taken in CAS is attractive with its proxy certificates, which give the access to the resources, so that the resources do not need the interpretation of the credentials. An interesting part of MyProxy approach is that online credential repository acts as a trusted intermediary between the web-portals and grid users. PRIMA approach is interesting with its dynamic on-demand creation and management of user accounts for small groups of grid users. PERMIS might be attractive with the creation of role-hierarchies and fine-grained access control based on roles and policies.

5. SUMMARY

In this paper, we described Fednets, which are group-oriented networks to share personal resources and services to achieve a common objective. We introduced its architecture in functional modules, which provide explicit information about what is shared, who is sharing and how the sharing is done. Further in our survey we compared Fednets with the related technologies in order to place Fednets amongst them. We summarize our observations as follows:

The access to the system, i.e. first-level access control. In SVE a new enclave can join the SVE through voting if only the majority of the enclaves agree on that. Each enclave maintains the list of trusted collaborators, i.e. enclaves of other organizations. The same principle works for VOs. In P2PWNC the first-level access control is carried out by Domain agents and in FON it is carried out by a special registration site. In Fednets, the access to the Fednet is controlled by the Fednet manager functionality. Similar to P2P file-sharing network, Fednets can have also an anonymous nature, in which the members do not need to know about other Fednet members.

The access to the resources, i.e. the second-level access control. In our survey, we encountered centralized and distributed access control to the shared resources. In grid networks that use grid map files, the CAS server, the VOMS server or the PERMIS policy engine the access control to the shared resources is centralized.

In the following cases the access control is distributed:

- In Fednets it is carried out at each PN by the PN agent;
- In grid networks that use AKENTI, it is carried out by each stakeholder;
- In grid networks that use PRIMA, it is carried out by each participating working group;
- In SVE it is carried out at each enclave by the enclave administrator;
- In P2P networks it is carried out by each peer.
- WCN, which is carried out at each Repeater AP and Wireless AP.

Complexity. The access control mechanisms and their complexity differ from technology-to-technology. The simplest mechanism is used in P2P networks, i.e. file encryption-decryption keys. The P2PWNC, FON and Grid map files use the mapping of the registered usernames to local accounts. Grid networks use access control policies based on privileges, capabilities or roles. The SVE uses

role-based resource access control policies, defined by the enclave administrator. In our design of Fednets we use access control policies based on criteria such as the contribution of the PNs to the Fednet and the behavior of the PNs in the resource sharing.

Management scope. In PNs, the PN owner manages his/her own resources, while the SVE administrator manages the resources of the enclave that belongs to one organization. The domain agent in P2PWNC manages the access control to the resources of a Wireless Community Network with many users. In grids the scope is even larger, i.e. CAS and VOMS administrators manage the resources that belong to several organizations.

Being one of the most important issues in sharing resources, in this survey, we have focused on the access control mechanisms of group-oriented networking systems reported in the literature. Moreover, we discussed the advantages and disadvantages of each approach. We showed that although, the concept of sharing resources in these technologies that belong to different owners is similar to the concept of Fednets, the implementation of the access control mechanisms is different. Similar functionalities such as managing and controlling the group cooperation, the access control over the community resources are implemented differently. Our final remarks are the followings:

• There are different solutions for the access control in group-oriented communications. However, there is no universal solution. Each of the proposed solutions counters a particular problem and thus has its own tradeoffs. The access control can be carried out at the resource itself; in this case, the resources should have access control capabilities to interpret the user's attributes and make an authorization decision. This brings overhead and complexity at the resource side. Another solution is having a centralized entity, such as VOMS or CAS servers, who decide on behalf of the community how to grant the access to the resources. This approach releases the resources from extra task of controlling the access, and the complexity will move to a centralized entity. But this creates extra overhead for the whole system, since a centralized entity should be maintained. In addition, it is prone to a single point of failure.

• Fednets stand close to grids with a number of their characteristics. The scale of their geographic span, the number and the variety of resources and services shared between PNs, as well as the number of participants can vary based on the goal of the Fednet and type of its applications. Fednets are, in fact, a grid of personal networks cooperating in a P2P manner.

• Based on the survey we conclude that Fednets are distinguishable from the discussed related paradigms and technologies in the types of the participating resources (which are personal) and devices (which are mostly portable and battery powered). Fednets are enabled by the collaboration of individual Personal Networks, thus the administrative domains are PNs. The uniqueness of the Fednets among the existing related technologies is that Fednets are temporal, opportunity or purpose driven ad-hoc sharing of personal resources and services.

Fednets enable users to share their personal resources in a seamless, secure and flexible way. Fednets have a potential to cover a variety of P2P application categories, such as communication and collaboration (instant messaging), distributed computation (sharing available processing power), internet service support (sharing internet connection, multicasting services) and content distribution (digital media sharing). PNs and Fednets can be seen as a next generation networking concept that allows organizing personal devices in order to make them cooperate in an effective way.

References

[1] M. Ibrohimovna and S.M. Heemstra de Groot, "Sharing resources in group-oriented networks: Fednet and related paradigms" in Proceedings of UBICOMM 2008, Spain, Valencia, October 2008.

[2] I.G. Niemegeers and S.M. Heemstra de Groot, "Research Issues in Ad-Hoc Distributed Personal Networking", Kluwer International Journal of Wireless and Personal Communications, Vol. 26, No.2-3, 2003, pp.149-167.

[3] I.G. Niemegeers and S.M. Heemstra de Groot, "FEDNETS: Context-Aware Ad-Hoc Network Federations", Wireless Personal Communications Vol. 33, N.3-4, pp. 305-318, Springer 2005.

[4] I. Foster, C. Kesselman (editors). "The Grid: Blueprint for a New Computing Infrastructure". Morgan Kaufmann Publishers, USA, 1999.

[5] Wireless Community Networks, http://wcn.cnt.org, 12.05.2009.

[6] M. Ibrohimovna and S.M. Heemstra de Groot, "Proxy-based Fednets for sharing personal services in distributed environments," in Proceedings of ICWMC 2008, Greece, Athens, July 2008.

[7] IST 6FP MAGNET, http://www.telecom.ece.ntua.gr/magnet/, 12.05.2009.

[8] The Dutch Freeband Communications Project PNP2008, http://www.freeband.nl. 12.05.2009.

[9] I.Foster, C.Kesselman, S.Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International J. Supercomputer Applications, 15(3), 2001.

[10] CrossGrid project. http://www.eu-crossgrid.org/, 12.05.2009.

[11] AccessGrid project. http://www.accessgrid.org/, 12.05.2009.

[12] Wilkinson, Grid Computing, Lecture notes. http://www.it.uom.gr/teaching/unc_charlottePPG/grid.htm, 12.05.2009.

[13] I. Foster, presentation "The Grid: Beyond the Hype," Argonne National Laboratory and University of Chicago, September 14th, 2004.

[14] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration", in Proc. of the IEEE 3rd Int. Workshop on Policies for Distributed Systems and Networks, 2002.

[15] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, K. L″orentey, and F. Spataro. Voms: An authorization system for virtual organizations. Proc. of the 1st European Across Grids Conference, Santiago de Compostela, February 2003.

[16] D. Chadwick and O. Otenko. The permis x.509 role based privilege management infrastructure. The 7th ACM Symp.on Access Control Models and Technologies, 2002.

[17] M. R. Thompson, A. Essiari, and S. Mudumbai. Certificate-based authorization policy in a pki environment. ACM Trans. Information Systems. Security, 6(4):566–588, 2003.

[18] M. Lorch and D. G. Kafura. The prima grid authorization system. Journal on Grid Computing, 2(3):279–298, 2004.

[19] D. Shands, R. Yee, J. Jacobs and E.J.Sebes, "Secure Virtual Enclaves: Supporting Coalition Use of Distributed Application Technologies", in Proc. of NDSS 2000, San Diego, California, February 2000.

[20] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: scalable secure file sharing on untrusted storage, Proc. of FAST '03:2nd USENIX Conference on File and Storage Technologies, San Francisco, CA, USA 2003.

[21] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: Distributed Anonymous Information Storage and Retrieval System. Lecture Notes in Computer Science, 2009:46, 2001

[22] E. C. Efstathiou at al., "Stimulating Participation in Wireless Community Networks", in Proc. of IEEE INFOCOM 2006, Barcelona, Spain, April 2006.

[23] FON. www.fon.com, 12.05.2009.

[24] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege and D. Spence, IETF RFC 2904, AAA Authorization Framework, August 2000.

[25] R. Yavatkar, D. Pendarakis, R. Guerin, A Framework for Policy-based Admission Control, IETF RFC 2753, January 2000.

[26] L. Pearlman, C. Kesselman, V. Welch, I. Foster and S. Tuecke, The community authorization service: status and future, CHEP03. March 24-28, 2003, La Jolla, California.

[27] The DataGrid Project. http://www.edg.org/, 12.05.2009.

[28] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure Proxy Certificate Profile. RFC 3820, June 2004.

[29] M. Lorch, D. Adams, D. Kafura, M. Koneni, A. Rathi, and S. Shah. The prima system for privilege management, authorization and enforcement in grid environments. In Proceedings of the 4th Int. Workshop on Grid Computing - Grid 2003, Phoenix, AZ, USA, November 2003.

[30] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist. X.509 proxy certificates for dynamic delegation. In 3rd Annual PKI R&D Workshop, April 2004.

[31] David Chadwick, Sassa Otenko, Von Welch, Using SAML to Link the Globus Toolkit to the Permis Authorisation Infrastructure, Springer Boston ISSN 1571-5736 (Print) 1861-2288 (Online) Volume Volume 175/2005.

[32] Shibboleth. A Project of the Internet2 Middleware Initiative. http://shibboleth.internet2.edu/, 12.05.2009.

[33] C. Neuman, S. Hartman, K. Raeburn, RFC 4120, The Kerberos Network Authentication Service (V5), July 2005.

[34] Grid Security Infrastructure, GSI. http://www.globus.org/security/overview.html, 12.05.2009.

[35] J. Novotny, S. Tuecke, and V. Welch. An online credential repository for the grid: MyProxy. In Symposium on High Performance Distributed Computing, San Francisco, August 2001.

[36] Security assertion markup language. http://saml.xml.org/saml-specifications, 12.05.2009.

[37] S. Chokhani and W. Ford, Internet X.509 Public Key Infrastructure. Certificate Policy and Certification Practices Framework. RFC 2527, March 1999.