D3.1 – POPEYE Security Design Document

Abstract

This report presents an overview of the architecture of POPEYE security (WP3) and the relationships with the other modules of the POPEYE framework.

Keywords List

Security, access control, trust, intrusion detection systems (IDS), Ad Hoc networks, Mesh networks
Executive Summary

This deliverable document entitled “POPEYE Security Design Document” is part of the design phase of the POPEYE project. Its purpose is to provide a first architectural view of the security functionalities addressed by WP3, and the relationships with the other modules of the POPEYE framework.

This document was developed taking the following documents as references:

- Milestone M3.1, entitled “POPEYE Security challenge and state of the art report”, that provides a presentation of the different security approaches used to allow securing the functionality of exchanges within mobile ad hoc networks.
- Internal document entitled “POPEYE Module Interface Description” written with the cooperation of all the partners of POPEYE and which lists all the modules that are going to be developed organized by Work Packages and which gives details of every module like the functions or methods provided by each module, a brief definition of the method and the complements that need to be developed.

The first section of the document provides a brief introduction, considering the conclusions of the previous documents and explaining the motivation behind the security architecture design.

The second section of this document presents the functional description of each module within the scope of WP3, namely: “Security Management”, “Privacy Services”, “Access Control” and “Super Peer Trust”.

The document then presents an architectural view of the different modules, describing both their static architecture and their dynamic behaviour. UML class diagrams and sequence diagrams are included to describe the internal architecture and the dynamic behaviour, respectively.

The last section of the document provides a description of the relationships between WP3 modules and other POPEYE modules.

Furthermore, this document describes and illustrates the achievements made in the first year of the project. The first two phases have concluded “specification and state of the art analysis phase” and “design research phase” and the key achievements of these phases are the following:

- Active participation in the End-Users Requirements Workshop – Security session
  We presented the key concepts of the POPEYE scalable security, that can rely on a PKI when a fixed infrastructure is available and switch towards lighter mechanisms when only pure P2P infrastructure remains.
  We asked for and received feedbacks and recommendations from the attendance.
• Definition of POPEYE Security challenge and state of the art (documented in M3.1):
  Several complementary mechanisms (trust computation, security architecture, group
  management, and intrusion detection) are proposed in the literature to secure the
  functionality of and exchanges within a wireless ad-hoc network. The main challenge of
  POPEYE security will be to couple this ad hoc approach with legacy infrastructure.

  In a first part, we presented our security analysis of the scenarios described in the WP2 –
  System Architecture, with a special focus on the issues raised by the behaviour of a
  hacker within the POPEYE infrastructure (the so called “bad guy” scenario). The
  security challenge POPEYE should face is summarized in a short list of security
  requirements.

  In a second part, we identified and evaluated some security mechanisms suitable to fulfil
  the security needs of the POPEYE users. Our investigation was specifically performed
  on the following topics:
  • Group Management and Access Control Service
  • Trust Support Services & Privacy
  • Intrusion Detection Systems for MANETs

• Draft POPEYE security architecture description (documented in M3.2):
  We presented the security functions required to support the key scenarios:
  • Join a group
  • Join a cluster
  • Create a group
  • Super-Peer trust

  The security services in POPEYE support collaborative working application for small
  spontaneous communities with sporadic connectivity. The security mechanisms are
  suitable for to ad hoc networks constraints.
  We defined the functionalities required to secure the POPEYE framework :
  • Trust and Privacy (Credentials and Certificates)
  • Access Control (Join the cluster, Create a group, Join a group)
  • Trust Peer and Overlay Network
  • Security Policies Services
  • Security Management

  We outlined the static internal architecture of each module (in which UML diagrams
  such as Component and Class Diagrams are used to illustrate that architecture), and their
  dynamic behaviour.

  We identified the relationships between the security modules and the other modules of
  the POPEYE architecture:
  • The certification process requires strong interactions with the Context layer and the
    module User Management to retrieve the relevant public data.
  • The computed trust can be used in super peer election at the network level (Network
    Abstraction layer). At the community level (Middleware layer), the trust history can
    help the POPEYE user to decide if she/he can accept a new member or if she/he
    should leave a group with unreliable members.
  • Comprehensive User Interfaces raises end-user awareness of the security level of any
    workspace members.
# Revision history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description, Editors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>22.05.2007</td>
<td>First Version, Lionel Besson and Frédérique Tastet Cherel (THC)</td>
</tr>
</tbody>
</table>
# Contributors

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Company</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lionel</td>
<td>Besson</td>
<td>THC</td>
<td><a href="mailto:lionel.besson@fr.thalesgroup.com">lionel.besson@fr.thalesgroup.com</a></td>
</tr>
<tr>
<td>Frédérique</td>
<td>Tastet-Cherel</td>
<td>THC</td>
<td><a href="mailto:frederique.tastet@fr.thalesgroup.com">frederique.tastet@fr.thalesgroup.com</a></td>
</tr>
<tr>
<td>Juan Antonio</td>
<td>Martínez Navarro</td>
<td>UMU</td>
<td><a href="mailto:JuanAntonio@dif.um.es">JuanAntonio@dif.um.es</a></td>
</tr>
<tr>
<td>Antonio F.</td>
<td>Gómez Skarmeta</td>
<td>UMU</td>
<td><a href="mailto:skarmeta@dif.um.es">skarmeta@dif.um.es</a></td>
</tr>
<tr>
<td>Paola</td>
<td>Inverardi</td>
<td>UDA</td>
<td><a href="mailto:inverard@di.univaq.it">inverard@di.univaq.it</a></td>
</tr>
</tbody>
</table>
Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE</td>
<td>Collaborative Working Environments</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc Network</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer to Peer</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
</tbody>
</table>

The present document has been produced in consistence with the definition of terms described in the POPEYE Glossary v1.0 accessible on the POPEYE web site.
# Table of Contents

Executive Summary ................................................................................................................................. 2  
Revision history ........................................................................................................................................ 4  
Contributors .......................................................................................................................................... 5  
Acronyms ............................................................................................................................................... 6  
Table of Contents ................................................................................................................................ 7  

1 Introduction ......................................................................................................................................... 9  

2 Security Functional Description ....................................................................................................... 11  
  2.1 Functionalities Overview ............................................................................................................... 11  
  2.1.1 Module functionalities ............................................................................................................. 11  
  2.1.2 End user awareness ............................................................................................................... 11  
  2.2 Security Management Functional Description ........................................................................... 13  
  2.2.1 Security Toolbox ................................................................................................................. 13  
  2.2.2 Global Security Configuration ............................................................................................ 13  
  2.3 Security Policies Services Functional Description ................................................................. 14  
  2.4 Privacy Services Functional Description ................................................................................. 14  
  2.4.1 POPEYE profile .................................................................................................................. 14  
  2.4.2 Private Information Database ............................................................................................. 15  
  2.4.3 POPEYE Certificates Process ............................................................................................. 16  
  2.4.4 Managing POPEYE profiles ............................................................................................... 17  
  2.5 Access Control Functional Description .................................................................................... 17  
  2.5.1 Access Control (Join the cluster): ...................................................................................... 17  
  2.5.2 Access Control (Create a group): ...................................................................................... 18  
  2.5.3 Access Control (Join a group): ......................................................................................... 18  
  2.6 Super Peer Trust Functional Description ................................................................................ 20  
  2.6.1 Trust Computation ............................................................................................................. 20  
  2.6.2 Superpeer Trust .................................................................................................................. 23  

3 Security Architecture .......................................................................................................................... 25  
  3.1 Security Management Architecture ............................................................................................ 25  
  3.1.1 Security Management Static View ....................................................................................... 25  
  3.1.2 Security Management Behaviour ......................................................................................... 26  
  3.2 Security Policies Services Architecture ..................................................................................... 27  
  3.2.1 Security Policies Services Static View ................................................................................ 27  
  3.2.2 Security Policies Services Behaviour ............................................................................... 28  
  3.3 Privacy Services Architecture ................................................................................................... 30  
  3.3.1 Privacy Services Static View ............................................................................................... 30  
  3.3.2 Privacy Services Behaviour ............................................................................................... 31  
  3.4 Access Control Architecture ....................................................................................................... 34  
  3.4.1 Access Control Static View ................................................................................................ 34  
  3.4.2 Access Control Behaviour .................................................................................................. 35  
  3.5 Super Peer Trust Architecture ................................................................................................... 41  
  3.5.1 Super Peer Trust Static View ............................................................................................... 41  
  3.5.2 Super Peer Trust Behaviour ............................................................................................... 42  

4 Security relationship with other modules ......................................................................................... 44  
  4.1 Security Management Relationships .......................................................................................... 44  
  4.2 Security Policies Services Relationships .................................................................................. 44  
  4.3 Privacy Services Relationships .................................................................................................. 44  
  4.4 Access Control Relationships .................................................................................................... 44  
  4.5 Super Peer Trust Relationships .................................................................................................. 44  

5 Conclusion .......................................................................................................................................... 46
1 Introduction

The POPEYE objectives are to provide a simple and reliable computing environment for collaborative group working.

An integrated overlay networking architecture is designed to benefit both from the stability and the performance of infrastructure as the Internet or LAN when available and from the flexibility and spontaneous character of mobile ad hoc networks. The POPEYE infrastructure integrates a communication platform and context-aware, secure and personalised core services to enable the design and the usability of collaborative applications in such mobile environments.

The main goal of this document is to define the architecture of the security modules within POPEYE framework and to establish their relationship with the other POPEYE modules. It will be used by WP3 members as a first draft of the security architecture and also by other POPEYE members to understand the functionalities provided by WP3 modules and to properly interface with the other modules.

This document was developed taking the following documents as inputs:

- Milestone M2.1 "POPEYE Architecture Description", that gave a first general overview of POPEYE architecture
- Deliverable D2.2 "Description of functional, non functional and technical requirements", that defines the requirements of the POPEYE framework and deliverable
- Milestone M3.1, entitled “POPEYE Security challenge and state of the art report”, that provides a presentation of the different security approaches used to allow securing the functionality of exchanges within mobile ad hoc networks.
- Internal working document “POPEYE Module Interface Description”, that describes the functions provided and required by each module and is used to define the module relationships
- Milestone M4.1 “Draft POPEYE Lower Level Architecture Description Report” that provides a first architectural view of POPEYE network abstraction and middleware services
- Milestone M5.2 “Draft POPEYE Core Services Architecture Description Report” that provides a first architectural view of POPEYE core services to support collaborative working applications (Data Management and Sharing Services, Workspace Management, User Profile Service, Context Collection Services, Context Delivery Services, Context Management Services, Plugin Management, Plugin API)

The POPEYE architecture, as defined in M2.1 (POPEYE architecture description report) was used as a starting point to define the WP3 modules, and their relationships. The identified WP3 modules are:

- Security Management
- Privacy Services
- Access Control
- Super Peer Trust

These modules will be linked with WP4, WP5 and WP6 modules, as WP3 acts as a transversal package.
The document is organized through the following sections:

- Introduction
- Functional Description
- Architecture
- Relationship with other modules

In Functional Description, we review each of the module and explain its functionalities. Section 3 (Architecture) describes the static internal architecture of each module (Component diagrams and other schemas are sued to illustrate that architecture) and their dynamic behaviour (Sequence diagrams are then used). Finally, section 4 (Relationships with other modules) introduces the relationships between WP3 modules and other POPEYE modules. An appendix is also provided, with the specific functions provided and needed by each of the modules.
2 Security Functional Description

This section describes separately the functionality of each of the modules in which WP3 is divided into, i.e. Security Management, Privacy Services, Access Control and Super Peer Trust. It explains in details what these modules do without taking into account architectural details.

2.1 Functionalities Overview

2.1.1 Module functionalities

This section provides a brief overview of each WP3 module focusing on the module functionality.

Firstly, the Security Management is aimed at providing all the basic cryptographic functionalities (sign, cipher, generate keys...) and to manage the global security configuration. This means every configuration that is not specific to a group or a workspace.

The Privacy Services are in charge of all functionalities that are related to the certified profiles. Thus, this module is in charge of managing private information (information that must only be delivered to specific users, via the use of certificates), generating and validating profile certification requests. It is also responsible for managing every certified profile that the user deals with (that is, the user own certified profiles, and the certified profiles of other users).

Access Control module is related to the different security aspects related to the stages to become a member of the POPEYE network, and the activities related to collaborative work with other peers (like creating or joining a group).

Finally, the Super Peer Trust module deals with the trust computation and provides security in the communication between super peers of the network.

2.1.2 End user awareness

Apart from providing functionalities for securing the POPEYE network, the WP3 modules are also in charge of raising the end-user awareness towards security. This is achieved by providing simple and representative graphical user interfaces concerning the trust and security issues about a user or a group.

A good graphical user interface must first present pertinent information. When looking at one user’s security page, we would expect to have access to its trust level, but also to useful information like the groups we are both belonging to, the recent common history, or the credentials provided.

The presentation is also very important, and POPEYE aims at providing accurate icons that allow the user to easily and quickly identify the attached signification.

Representing the trust level of a user is achieved by modifying the user icon, and attaching it symbols like a gauge or an arrow.
The colour of the icon representing the peer reflects the trust that we have in it. For example, a completely un-trusted peer will appear red. A trusted peer will appear green, and mixed green-red colours will be used to represent intermediary values. If we use an unbounded trust value, the correspondence between trust value and colour must be set arbitrarily. The trust level can also be represented by using a gauge that can also be coloured, or simply by adding the value of the trust next to the peer icon. This solution is more explicit, but may require more attention from the user to decrypt it. Furthermore, an interesting feedback to the user could be the recent evolution of the trust in a peer. For example, a green rising arrow could be an indication that in the last 15 minutes, the trust that we have in a peer as been increasing.

The user could also use some information that is not directly linked to the security, but worth having. For example if a user is the creator or the manager of a group, if he’s a super peer…

The difficulty is to clearly make the distinction between the different attributes. Thus it is necessary to use very role-centric symbols, like an S (or a Superman logo) to represent a super peer. If a group is secured, if a communication is encrypted, a lock should be added to the icon. Every certified entity should also be represented with the symbol. Finally, a yellow triangle is the most common way to represent a warning.

All this different approaches can naturally be combined, depending on what we want to put stress on. The next figure shows some example of graphical representations of the security.

Figure 1: Security graphical representations

The experience of the user will be enhanced with the possibility to easily browse between the different security information panels. Whenever it is possible, elements presented will be clickable to access to their own security information panel. Thus, if presented with the groups that you share with an other user, the POPEYE user will be able to click on the group’s name to access the panel presenting the security information on this group. By this means, he can easily have an overview of the security issues he is facing and react more accurately.

Figure 1 is a mock-up of such a graphical user interface.
2.2 Security Management Functional Description

This section describes the functionality of the Security Management module. The goal of this module is to manage the global configuration of POPEYE security and to provide the basic security functionalities like sign or cipher. Thus this module is composed of two sub-modules.

2.2.1 Security Toolbox

This sub-module provides the basic security functionalities related to cryptography, like sign, cipher and generate keys. Those functions may be used by any of the different modules of the POPEYE framework, and aren’t specific to WP3.

2.2.2 Global Security Configuration

The global security configuration represents all the security parameters that aren’t specific to a user or a workspace. For example, the parameters used to compute the trust will be managed by this module, such as the different coefficients that must be applied for the trust computation criteria (this includes the coefficient attached to the common membership...
criteria, or the evolution of an event influence with the time when computing the experience associated with a user).

In order to achieve the goal of raising the awareness of users, it is necessary to present only two or three default configuration sets (with explicit names, such as ‘paranoiac’ for a configuration that would be very strict with users performing suspicious operations). The user will still be able to have a full control on those parameters but will need to switch to an advanced mode for this.

### 2.3 Security Policies Services Functional Description

This section is dedicated to the Security Policies Services module. The POPEYE security policies are specific to Workspace and Group Management. This module is used to manage and retrieve the set of rules that regulate how security is implemented and protects and distributes sensitive information.

The goal of POPEYE is to provide flexible security that can be easily managed by the end user. Thus, it is necessary to allow non-technical users to make accurate but simple changes with respect to the governance of their system, without requiring specific knowledge or technical support. This implies that a POPEYE user must be able to write and edit policy in terms that have meaning to them; that means that the user will specify what he wants the system to do (with limited knowledge on how it is achieved) and without having to use formal languages or modelling.

Developing tools that would translate the user intentions into formal security policy language is out of the scope of the POPEYE project. Thus, the first version of POPEYE will only present configurations panels with a limited set of rules to raise awareness of security issues (3 default security levels). Advanced users will still be able to modify the policies with the finest granularity.

### 2.4 Privacy Services Functional Description

This section describes the functionality of the Private Services module. The goals of this module are to manage a database with private information used for creating a POPEYE profile, to issue POPEYE certified profile and to manage and store those certified profiles.

#### 2.4.1 POPEYE profile

A POPEYE profile is a public profile or the claimed identity of a user within a POPEYE Workspace. The main field of the certificate is composed of three main parts:
Figure 3: POPEYE Profile Issuing

- The personal key of the user.

- Personal data about the user identity. This field contains business data, personal data, social data and other information about the user.

- Contributions. Contains data about the user’s behaviour. In one hand, this field contains keywords extracted from user interest and, on the other hand, user’s skills and references. This data is supposed to be included in the scope of the workspace.

Whereas the first field is a cryptographic key represented by a unique identifier, the other two fields refers to data that can be obtained using the Context Services, the Privacy Services Information Database or entered directly by the user when creating the profile.

Concretely, when creating a profile, the user will be presented a series of pre-filled fields (from Context Services or Private Information Database) that he will or not enable. He will also have the possibility to add personalized information by editing empty fields, and the possibility to provide credentials or certificates with the profile.

2.4.2 Private Information Database
A user may want to add private information to a profile, that means information that will only be disclosed to the members of the group associated to the profile. This information may for example be required for the validation of the process for joining a specific group, but should not be disclosed to other people.

As the information provided by the Context Services is publicly available, this module cannot be in charge of managing private information. This is the role of the Private Information Database of the Privacy Services module.

This database will use a simplified version of the ontological model from WP5 so that the data will be presented as a list of pairs <name, value>. This is due to the necessity of obtaining this information directly from the user, by means of a form dialog dedicated to the management of the Private Information Database.

2.4.3 POPEYE Certificates Process

A POPEYE profile needs to be certified by a trusted third party or one of the group members. The Privacy Services module handles the associated functions of the process (create a POPEYE profile certification request and certify a POPEYE profile).

The POPEYE certificates process is presented in the following figure:

![Figure 4: POPEYE Certification Process](image)

A POPEYE profile certification request consists simply of sending the POPEYE profile to the certifying authority, signed by the personal private key of this POPEYE profile.

When receiving a certification request, will extract the information stored in the profile. He will thus have available all the data (group identity, personal data, user skills and interests,
credentials and certificates) necessary to decide whether or not he should certify the profile. The decision is based on the group security policy.

If the validation process succeeds, the POPEYE certified profile is returned to the POPEYE user, and stored in the certifying authority Profile Database. Otherwise, the requester is informed that his request has been rejected.

2.4.4 Managing POPEYE profiles

The Privacy Services module is in charge of storing and managing certified POPEYE profiles (own certificates and members certificates). Only certified profiles are stored.

User’s certified profiles are added when a POPEYE certification process occurred. When a user leaves definitely a workspace, he will be asked if he wants or not the corresponding profile to be removed from the database.

A second database is in charge of collecting all the certified profiles that we encountered, associated with the corresponding groups and profiles. This is useful for example in order to raise the user attention when members already members of a group we belong to becomes very suspect and cannot be trusted anymore. The user could then decide to quit the group or eventually to create a new group, but that will explicitly rejects the un-trusted user.

2.5 Access Control Functional Description

In this section we are going to show different security aspects depending on the scenarios caused by the stages to become a member of the network and the activities related to collaborative work with other peers. These actions require a bootstrapping process to generate the POPEYE certificate, which will be the credential presented to the corresponding authority in every authentication process. This bootstrapping process will be described in the section 3.5.

2.5.1 Access Control (Join the cluster):

The first step for a peer to enter in the network is the discovering of the nearest super peer to join his cluster. In order to secure this process is needed certain credentials to assure the identity of each entity that will take part in this process, in our scenario, a peer and a super peer.

The process is described in the next figure as follows: Once the peer receives the announcement of the super peer presence he replies with a message, signed with his private key, in which is included a certificate used as his credentials. Since the super peer is the responsible of the cluster, he will receive this message in order to validate the identity of the new peer. If the validation is successful then he will store the peer certificate in his database and will confirm the decision to the peer. In the reply the super peer will also include the list of groups and services of the network to allow the peer to join them.
Access Request (signed by the personal key used for this POPEYE Profile)

Validation Process (based on Workspace/Group security policy)

List of known groups and services

Members Certificates Database

If access granted, the certificate is stored in the Group Membership database

Figure 5: Access Control – Join the cluster

2.5.2 Access Control (Create a group):

In order to collaborate with other peers it is necessary the creation of groups and services to group peers with same interests. However only certain privileged peers can do this action. Integrating security in this stage is the only way to assure that unauthorized peers could not create groups with malicious intention.

The super peer, as a cluster manager, has to authorize these actions. The process carried out by the entities of the network is the same as in the former scenario. A peer sends a signed message to request the creation of the group to the super peer to get his authorization. If the validation process is successful then the super peer will send a reply to the peer and will update the information of the services of his cluster with the information of the new group to notify the other clusters.

Depending on the security policy implemented he will be the manager of the group having the responsibility of admitting new members to the group or this labour will be delegated to all the members using a voting scheme to authorize the new members.

2.5.3 Access Control (Join a group):

This activity is the more complex one and it can be implemented in two different manners as we said above: the first one with a group manager in charge of the group and the second one using a voting scheme to do this activity.

Using the same scheme as described in the above sections a peer that wants to collaborate with other peers accessing a group has to request the group manager, instead of the super peer, the authorization to enter the group. The peer will sign this request and will accompany it with his credential. If the group manager validates his identity and check that the peer owns the profile and permissions necessary to be accepted in the group, he will send a reply and will cipher with the public key of the new peer the group key used to cipher all the traffic of the group.

The credential presented to a group manager or to the members of the group is a wallet with:
The POPEYE profile presented by the requester
The associated public key
The validation of above attributes (by a third party or by the requester – self signed certificate)

POPEYE User

<table>
<thead>
<tr>
<th>Personal Key</th>
<th>Personal Data</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>* from a TTP</td>
<td>* a Group member</td>
<td>* or the User (self-signed certificate)</td>
</tr>
</tbody>
</table>

Access Request
(signed by the personal key used for this POPEYE Profile)

POPEYE Certified Profile

<WP3> Privacy Services

Validation Process (based on Workspace/Group security policy)

List of known groups and services

Members Certificates DataBase

If access granted, the certificate is stored in the Group Membership database

Request + Credential

Figure 6: Access Control - Join a group

Other possibility is delegating this process to the current members of the group using a voting scheme like shown in the next figure. This scheme is based on Threshold Cryptography in which a secret is shared among a number of members, and can be regenerated if at least you can recover X parts of the secret.

Figure 7: Voting scheme

This process begins with a bootstrapping process in which the group creator distributes the group partial keys among a limited number N of members previously decided. This process will warranty the authorization process even if the creator leaves the group. Thus a new peer could be accepted by the members of the group if the majority of the votes sent to the new peer are for the authorization of entering the group.
The next paragraphs will describe the whole process to admit a new peer to a group using this scheme:
The new peer will start this process by sending a signed request to the group multicast address to become a member of the group. This request includes the name of the group and the usual fields, such as issuance time and the validity interval. It may reference the guest identity Public Key Certificate (PKC).

After receiving the request, a group member first extracts the sender’s PKC and verifies the signature. If the member approves of admission, he will replies with a signed vote and its group membership certificate (GMC).

Finally, on receiving $K (\geq N+1)$ messages in the above step, the guest randomly selects $N+1$ votes out of $K$ of these sponsoring members, validates the membership of the selected members by asserting the correctness of the corresponding GMCs and using these votes he composes his own GMC will allow him to receive the symmetric group key to cipher all the traffic and participate into the group activities.

2.6 Super Peer Trust Functional Description

This section describes the functionality of the Super Peer Trust module. The first goal of this module is to automatically compute a trust level for each members of the network. This trust is one essential component of the flexible and MENET-adapted security used in POPEYE; it will be used in every security decision, particularly when no other trust mechanisms (like PKI) can be used.

This module will also be in charge of securing the super peers from the network abstraction layer (see WP4), which are nodes responsible for each cluster of the network. This will be achieved by the integration of trust in super peers election mechanism, and by the ciphering of all the messages transmitted among them.

2.6.1 Trust Computation

The identifier of each POPEYE profile is a public key (or a hash of this key). A trust level is computed for each POPEYE profile. Even if a user has different profiles (different identities), trust will be computed in a completely independent way for each profile.

An ‘a priori’ trust value is given to each POPEYE new profile. This value can be increased if an associated certificate issued by a trusted third party is provided.

Depending of the behaviour of the POPEYE user (e.g. the POPEYE user is actively participating to several groups), this value is updated.

In POPEYE, we intend to take in account the following trust computation criteria:
- Alerts and Distributed IDS: users detecting anomalous behaviours send the alerts. The management of the alert depends of the trust in the emitter.
- Certificates or Credentials provided: these recommendations are used to compute initial trust, and also to update trust live.
- Group Membership: the trust is updated when users are members of the same groups
- Experience: this criteria represents the memory of the interactions with other users
- Reputation: the objective is to shared opinion of the community on a user
- Paranoia and balance between different trust sources: the importance of each criterion differs for different users. We propose a few default levels.
Figure 8 summarizes the different criteria used to compute trust in POPEYE. Their role is developed in the next paragraphs.

![POPEYE Trust Criteria Diagram](image)

**Figure 8: POPEYE Trust Criteria**

2.6.1.1 **Alerts and Distributed Intrusion Detection System**

Users that detect suspicious or abnormal behaviours raise alerts. This process can be either automatic (for example raised by the distributed IDS) or initiated by the end user itself (through the graphical user interface).

Once a risk has been assessed, and depending on the criticality and nature of the event, the alert is spread either to all the nodes of the network, or to the node’s neighbours, or only to trusted nodes.

On receiving an alert, a node should check the trust level associated to the source of the alert and decide whether or not to handle the alert, and how the trust of the denunciated node should evolve. If an alert has been proven wrong or right afterwards, the trust level of the source should be modified accordingly.

The distributed IDS is one mean for evaluating the trust of a peer. Furthermore, it is possible to implement other alerts based on other components, e.g. if a data is corrupted or infected by a virus, or if an identification process failed.

Diversity in the systems that can raise an alert (every node could implement different alert thresholds with the same IDS) might increase the global security, because it is more difficult for an attacker to bypass those protections without the knowledge of their nature.

2.6.1.2 **Certificates or credentials provided**

POPEYE is able to work with an infrastructure or in a complete peer-to-peer mode. A certification authorities database is kept in memory, so that the security provided by the certification mechanism is kept even if no infrastructure is available.
A user providing a certificate or some kind of credentials signed by a trusted third party or an other user can be used to modify the initial trust. This mechanism can also happen anytime during a POPEYE session, and trust should be modified accordingly.

If the authority of the certificate is a POPEYE user, the trust increase should be correlated with the trust in the user. If the authority is a trusted third party, the trust is raised with a fix value.

2.6.1.3 Membership of groups

A user that belongs to a group we’re also a member of is trustworthy because he passed the process of entering the group.

Depending of the security level associated with the group (for example if it is a completely open group or if every member passed a very strict acceptance test), the trust associated with the group membership should be refined.

The past has no impact on this mechanism. That means that even if two nodes belonged to the same group in the past, it won’t affect the trust associated to group membership. Furthermore, we only deal with one level of common membership (there is no transitivity). In fact, all this is taken into account by other trust computation mechanisms like reputation or time-evolution.

2.6.1.4 Experience

Experience corresponds to the memory of the interactions that a user had with each it peers. Although it is relatively simple to assess criterions linked with the possibility of a user (like bandwidth, energy level or available disk space), it is far more complicated to assess a security criterion.

One database will be used to keep track of the past interactions. Only events related to security will be tracked. That means that criteria concerning the dependability of the network won’t be taken into account.

Different experiences:

- Successful or unsuccessful transfers (checksum and availability)
- Virus detection (if the antivirus allows feedback)
- Small automatic tests (like sending a small file and testing the result)
- Feedback provided either by the end user (is the concerned profile active or not...) or through the context awareness module. When feedback is provided by the end user through the graphical interface, the information is broadcasted to the other peers.

The database can bring some scalability problems on small portable devices. Thus every event should be associated with a timeout, and there should be a maximum number of events reported. When the database is full, the older events should be discarded first.

Trust of subject $i$ toward $s$ can be expressed as follows:

$$ C_{es} = \sum_t (\alpha_t \cdot \sum (Elk_{i, s} / n)) $$
Where $0 < E < 1$. $Elk_i$ represents the evaluation by $i$ of the criterion $l$ during an experience with the subject $s$. $k$ is present because multiple events of same kind can occur.

The value of an experience decreases in time, so the $\alpha$ coefficients are time dependant.

2.6.1.5 Reputation

Reputation corresponds to the shared opinion of a community. The goal is to share one’s comprehension and knowledge of the environment with the other members of the community in order to improve one’s knowledge using other’s information.

We will use the transient trust model in POPEYE: reputation is computed along all possible paths in a trust graph, and results in an average trust value. Figure 9 is an example of such a computation; the reputation of the green node for the purple one is 0.241.

![Figure 9: Transient trust model](image)

2.6.1.6 Paranoia and balance between the different trust elements

Ideally, each node should have it’s own parameters, depending of the trust he wants to put in every criteria. For example, very paranoiac nodes should increase the trust of a node very slowly, and decrease it after only a few mistakes.

However, the complexity introduced by this very many parameters should not be rejected to the user’s choice, hence the security management for the user might be as complicated (maybe more) than without the automatic trust computation. Ideally, we should provide 2 to 3 default levels, and only allow the end user to modify the different parameters in an ‘expert’ mode.

2.6.2 Superpeer Trust

As described above, the network is organized in clusters and the responsible of each cluster will be the super peer. In order to provide security in the communication between super peers is necessary to cipher all the messages transmitted among them. This process will be done in the following manner:

Every super peer will have the symmetric key group in order to cipher their communications. They will also have to sign every message they send in order to assure their identity. In order to provide the availability of the POPEYE certificates, the certificate repository will also be connected to the super peers, like a super peer more.
When a super peer or other even peer needs a certificate it will request it to a well-known multicast address (Security Group Multicast Address) aimed at this group of entities. This request will be received by super peers and the certificate repository. And they will reply directly to him by a unicast message.

The next figure, Figure 10, shows this secure channel between super peers and the certificate repository.

The trust computed by the Super Peer trust module will also be used as a key criteria for the Super Peer Election algorithm. A node that isn’t trusted by the existing super peers won’t be able to become one.
3 Security Architecture

In this chapter we focus on the architecture of WP3 modules. The architectural design, the structure and the organization of each module are described in the corresponding subsection. Sub-components of each module are identified and described as well. For each module, we describe the static internal architecture and its dynamic behaviour.

Figure 11: POPEYE layered architecture

3.1 Security Management Architecture

3.1.1 Security Management Static View

The Security Management module is aimed at providing the functions needed to deliver the information concerning a POPEYE user or a POPEYE group.
The next figure shows an abstraction of the Security Management module architectural overview. It is not intended as a hard-wired division into sub-modules, but as a view of the architectural design of the module. The main sub-modules are:

- **Global Security Manager** is in charge of providing to the GUI all the functionalities to manage the security configuration that is not specific to a Workspace, a Group or a User. It serves as a wrapper between the GUI and the Super Peer Trust module configuration.

- **User and Group Security Feedback** is in charge of retrieving the security information concerning the different groups, workspaces and users from the corresponding modules. It can also relay to the Super Peer Trust module the intention of a user to manually raise or lower the trust that he has in an other user.

- **Security Toolbox** provides the security functionalities to sign, cipher, generate cryptographic keys...

![Security Management Module Architecture](image)

**Figure 12: Security Management Module Architecture**

### 3.1.2 Security Management Behaviour

The behaviour of this module is pretty simple, since it consists almost in forwarding the requests to the appropriate modules (e.g. Group Management Services), and forwarding back the answers to the graphical user interface.

The following figure illustrates the exchanges happening when the end-user decides to manually raise the trust he has in a particular profile, for example because he met the corresponding physical person and was convinced of its identity.
It is possible that other global configuration than the one concerning Super Peer Trust might be used, although none where identified at this stage. The Security Manager would then have to manage them directly, without relying on other modules.

### 3.2 Security Policies Services Architecture

#### 3.2.1 Security Policies Services Static View

The Security Policies Services module is in charge of editing and storing the security policies associated with the group and workspaces.

The next figure shows an abstraction of the Security Policies Services module architectural overview. It is not intended as a hard-wired division into sub-modules, but as a view of the architectural design of the module. The main sub-modules are:

- **Security Policy Editor** is in charge of creating and editing security policies for workspaces and workgroups.
- **Security Policies Manager** is responsible for the storage of the policies associated with the different workspaces and workgroups, and for the communication with the Workspace and Workgroup Management Services.
3.2.2 Security Policies Services Behaviour

The behaviour of this module is better understood by the description of its main interactions, developed below.

3.2.2.1 Creating a security policy for a group or a workspace

When a Workspace or a Group is created, a security policy needs to be created and attached to it.

When requesting the creation of the security policy of a Group, the Security Policies Manager is asked for it, as it handles the communication with the other modules. The Security Policy Editor is used to interact with the user and create fill the different field of the policy with the appropriate values. By default, the user will be given a set of 3 default policies with predefined parameters suitable for the most common situations. He will also be able to switch to an advanced mode, and can then modify each parameter individually. The policy is then stored in the database of the Security Policies Manager, and its identifier is returned to the requester.

A workspace comprises a Group, shared data and tools (Plug-ins associated to sessions). Thus, the security policy attached to a workspace is composed of the rules that are specific for the workspace, and of a link to the policy of the corresponding group. The figure below describes the interactions that occur.
3.2.2.2 Requesting a policy parameter

The Workspace and Group Management Services need to be aware of the security policies of their Workspaces and Groups in order to know how to handle for example the communication encryption or how to perform the validation process of a newcomer to a group.

This task is achieved by the Security Policy Manager. The requester can either ask for a list of all the parameters available or for the value of one particular parameter. The following figure shows an example where the Group Management Services first obtains the list of the parameters of a security policy, and then retrieves the value of the parameter “Default policy” (which is in this case “Paranoiac” but could be “Low” or “Personalized”).

![Figure 15: Creating a Security Policy for a Workspace](image-url)
3.2.2.3 Editing a policy

If the policy of a Group or of a Workspace needs to be changed, the request (with the id of the policy that needs to be changed) will be forwarded from the Workspace or Group Management Service to the Security Policy Editor. The user will then be able to chose a new default configuration, or to edit manually the parameters (advanced mode).

3.3 Privacy Services Architecture

3.3.1 Privacy Services Static View

The Privacy Services module is in charge of managing the private information. It also aims at certifying profiles, and managing them.

The private information consists in all data that cannot be managed by the Context Services (all the information managed by the Context Services is public) and that will be delivered only to the members of common groups, via the profile used in the group.

The next figure shows an abstraction of the Privacy Services module architectural overview. It is not intended as a hard-wired division into sub-modules, but as a view of the architectural design of the module. The main sub-modules are:

- **Private Information Manager** is the entity that is used to store and retrieve all personal information.
- **Private Information Editor** is used by the POPEYE end user to create, modify or delete private information.
- **Profile Manager** is in charge of storing every certified profiles (User own certified profiles and the certified profiles that where presented to the user by other POPEYE users)
- **Profile Certification Centre** is in charge of certifying presented profiles, and requesting trusted third parties or other POPEYE users to certify the user own profiles.

![Diagram of Privacy Services Architecture](image)

**Figure 17: Privacy Services Architecture**

### 3.3.2 Privacy Services Behaviour

The behaviour of this module is better understood by the description of its main interactions, developed below.

#### 3.3.2.1 Editing Private Information

The following figure explains the interactions occurring when the end-user edits its private information.
3.3.2.2 Creating and certifying a profile (see also WP5 – User Management)

Figure 19: Creating and certifying a new profile
The end-user initiates the creation of a new POPEYE profile, and forwards this request to the User Management module (WP5). The User Management module is in charge of creating the new User Profile.

A new cryptographic key is retrieved from the Security Toolbox, and is incorporated in the profile. Useful data is then retrieved from the Context Management Services (WP5) and from the Private Information Manager.

The POPEYE user is then presented a graphical interface that let him decide which fields he wants to use, and lets him add any extra information needed like certificates and credentials. The User Management module is now able to generate the profile file.

This profile is forwarded to the Profile Certification Center, who creates the certification request. Once the request has been validated by the third party, the certified profile is stored by the Profile Manager of WP3 (in the section concerning the POPEYE user’s certified profile), and the user is acknowledged that the operation succeeded. The user is also notified if an error occurred during the whole process.

### 3.3.2.3 Profile Certification Request

![Figure 20: Certification Process](image)

The previous figure shows the sequence diagram corresponding to the process of certifying a profile presented by an other POPEYE user.

When the certification request is received by the Profile Certification Center, its information is retrieved and the Security Management and Privacy Services modules are requested to validate the request, comparing the data presented and the different security policies.
If the request is validated, the profile is certified, and stored by the Profile Manager (in the section concerning other user’s profiles). The certified profile is then sent back to the requester.

### 3.3.2.4 A user of a group cannot be trusted anymore

In this section we consider a user that belongs to a group we also are member of, and that becomes un-trusted for any reason.

The Profile Manager becomes aware that we have a group in common with a very suspect, and decides to raise the attention of the user and to ask him to take an appropriate decision, that is, purposely staying in the group, or leaving the group (and maybe alerting the other members), or even create a new group with an explicit policy rule that will prevent the untrusted user to enter it. In the figure below, the second choice is made by the user.

![Figure 21: User awareness and decision about group trust](image)

#### 3.4 Access Control Architecture

### 3.4.1 Access Control Static View

As shown in Figure 10, the network will be formed by:

- A register node with two network interfaces, the first one in infrastructure network to register new users to the POPEYE network and the other one in ad-hoc mode to participate in POPEYE network providing certificates as a certificate repository.

- In this network there will be super peers who will be in charge of their clusters taking decisions about allowing new peers to join their clusters, and notifying updated information about the network to the other super peers.

- And peers that will be the main users that participate and collaborate in POPEYE network.
Super peers and the certificate repository will form a group called Security Group with its own multicast address that will provide the certificates to the ones who request them. This group is necessary in order to not overload the network sending the certificate with every message along the network.

3.4.2 Access Control Behaviour

In order to provide the POPEYE certificate used as a credential in every security process, a bootstrapping process is needed.

First of all a user must configure his mobile device wireless card in infrastructure mode to detect the Registration POPEYE wireless network (Figure 22).

![Figure 22: POPEYE wireless network detected](image)

Once he selects this network a web browser will be run in his device showing a web portal asking for him to log in it. This process is carried out by a captive portal which will not allow the user to establish other communications through this network until he has registered in the network and received his POPEYE certificate.
Figure 23: POPEYE captive portal

Afterwards the user will fulfil a form with his personal data and also his interests (Figure 24) in order to receive a certificate (Figure 25) signed by a certification authority to use in the main activities of POPEYE network.
Welcome to 2nd world conference of ....

Please, fill in this form to obtain a certificate that will allow you to enter Popeye network:

<table>
<thead>
<tr>
<th>Personal and contact information:</th>
</tr>
</thead>
<tbody>
<tr>
<td>First name: <em>Required</em></td>
</tr>
<tr>
<td>Last name: <em>Required</em></td>
</tr>
<tr>
<td>Contact information is for:</td>
</tr>
<tr>
<td>☐ Home ☐ Work/business</td>
</tr>
<tr>
<td>Best times to contact me:</td>
</tr>
<tr>
<td>Address: <em>Required</em></td>
</tr>
<tr>
<td>City: <em>Required</em></td>
</tr>
<tr>
<td>Provincial/state: <em>Required</em></td>
</tr>
<tr>
<td>Postal code/zip: <em>Required</em></td>
</tr>
<tr>
<td>Country:</td>
</tr>
<tr>
<td>Phone: <em>Required</em></td>
</tr>
<tr>
<td>Email:</td>
</tr>
<tr>
<td>Interests ...</td>
</tr>
</tbody>
</table>

Figure 24: Web form
Once this bootstrapping process is achieved the user only has to change the configuration of his wireless device in ad hoc mode and establish the right SSID and channel.

This configuration will allow him to interact and collaborate with other members of POPEYE network. In the following diagrams that we are showing below several access control activities are described.
In Figure 26, it is shown how a peer must connect to its closest cluster. After receiving the periodic SP_Info_Cluster of the nearest super peer, he will send a signed reply, which will include his identity. The super peer will receive this message and will check the identity of the new peer in his temporally certificate repository, it he does not have it, he will send a multicast message aimed at the Security Group to request the certificate of the peer identity. Once he has received it he will check the identity of the new peer. If the check process is successful then he will authorize the peer to join his cluster and will send him the list of services and groups of the POPEYE network.
Figure 27: Access control - group creation

Only some privileged peers can create or delete groups, in order to allow only these privileged peers to carry out this operation it is needed to secure these operations. When a peer wants to create a group, he will send a SP_New_Group message to the super peer, this message must be signed with the private key of the peer and also must include the identity of the peer. The super peer, will do the same aforementioned process, checking the identity of the group creator. Once he is available to make a decision he will send to the group creator, if it is successful the super peer will update his service list and will update the information of the other super peers sending them a SP_Info message.
Finally, one of the more common operations in POPEYE is the operation of joining a group. Applying an access control operation to this activity only the peers with the same interests will collaborate in the same group. The process described in Figure 28 is very similar to the previous diagrams. In this case the group manager will be one who asks for the certificate to the Security Group. Depending on the certificate will decide if the peer can be a member of the group or not.

### 3.5 Super Peer Trust Architecture

#### 3.5.1 Super Peer Trust Static View

The next figure shows an abstraction of the Super Peer Trust module architectural overview. It is not intended as a hard-wired division into sub-modules, but as a view of the architectural design of the module. The main sub-modules are:

- **Distributed Intrusion Detection System and Alerts Manager** is in charge of the interaction with the distributed IDS and to raise other kind of alerts
- **Reputation Manager** implements the transient trust model and computes the reputation of the different POPEYE users
- **Experience Manager** manages the events database, and computes the experience feedback accordingly
- **Certificates and Credentials Manager** handles the certificates and credentials if any.
- **Group Membership Manager** interacts with Group Management module from WP4, and computes the associated trust.
- **Peer Trust Computation** computes the global trust with the previously mentioned sub-modules, and is the interface for every other POPEYE modules.
- **Super Peer Manager** is in charge of managing the super peers trusted channel.

![Diagram](image)

**Figure 29: Super Peer Trust Module Architecture**

### 3.5.2 Super Peer Trust Behaviour

#### 3.5.2.1 Peer Trust Computation

The behaviour of the sub-modules in charge of the peer trust computation is simple, the Peer Trust Computation sub-module gathers the different inputs from the other sub-modules and computes the resulting trust levels. He also acts as an interface for every communication with other POPEYE modules.

#### 3.5.2.2 Super Peer Secure Channel

This super peer trusted channel will be formed by the super peers of every cluster establishing among them a secure channel, as shown in Figure 10: Super Peer secured network. Signing and ciphering every message aimed at the super peer multicast address will guarantee that no other peer will be able to listen or participate in these communications.
The interactions of the elements of the aforementioned architecture are shown in this sequence diagram.

First of all super peers and the certificate repository uses the previously established group key to cipher their messages. They also must sign the messages in order to allow the others super peers to authenticate them.

Once a peer creates a group the corresponding super peer will update its service list and will notify the others super peers through a ciphered and signed SP_Info message. This process is shown in Figure 30.

![Sequence diagram of super peer collaboration](image_url)

**Figure 30: Super peer collaboration**
4 Security relationship with other modules

In this section we focus on the interactions between WP3 modules and other POPEYE modules.

4.1 Security Management Relationships

The Security Management provides a number of basic cryptographic functionalities like sign, cipher, generate keys… These functions can be used by any other POPEYE module, and even by plug-ins if they need such tools.

The global security configuration management isn’t accessed by other POPEYE modules.

4.2 Security Policies Services Relationships

Security policies are specific to a group or a workspace. These policies will be used by different POPEYE modules to check whether or not an action can be performed, if they have security rules related to them:

-   Workspace Management (WP5)
-   Group Management (WP4)
-   Framework Manager (WP6) will also have access to the security policies, because it acts as an entry point to the POPEYE Group, Workspace and Plug-in Management.

4.3 Privacy Services Relationships

The POPEYE profile certification process will imply both the Privacy Services module and the User Management Module from WP5. The User Profile generation is left to the User Management module, but all the certification process (Request and validation) is done by the Privacy Services. The User Management module also employs the Security macro-component to create new user accounts on security level.

The Context Management doesn’t interact directly with the WP3, but via the User Management module.

4.4 Access Control Relationships

Group management module will need several functionalities from Access Control module. In first place, access control lists (ACL) will be needed when new users connect to a group. Thus, group management module will verify if the user has the corresponding rights to access the selected group. Furthermore, group creation rights (ACL and security policies) must also be checked before users can create new groups.

4.5 Super Peer Trust Relationships

Peer trust provided by the Super Peer Trust module will be used potentially used everywhere when trust is the only way to decide whether an action or not should be performed.
It will for example be used by the replication mechanisms to decide where to place replicated data, because certain data are not to be stored in un-trusted peers. Super Peer Trust will also be used by the Group Management module, for example when one or more member of a group become too un-trustable, or indirectly via the group membership validation process. It is also a component of the super peer election algorithm.
5 Conclusion

This deliverable has evolved over the last months and reflects the current state of WP3. We have provided an architectural overview of the different modules of WP3, in terms of functional description, internal architecture design, and relationship with other POPEYE modules. UML diagrams have been used to complete descriptions: component diagrams to describe the internal architecture, and sequence diagrams to describe the dynamic behaviour and relationships between modules.

This document stands as the first POPEYE deliverable where all security modules are described and will be very useful for the architecture definition and implementation process of WP3 services.
APPENDIX A – MODULE INTERFACE DESCRIPTION

WP3 modules interface

**Module name:** Security Management  
**Description:** This module provides the security functionalities like sign, cipher.

**Provided functions:**

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Consuming modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign</td>
<td>Signs with your private key the content of the second parameter (data)</td>
<td>Private key, data</td>
<td>The sign of the second parameter</td>
<td></td>
</tr>
<tr>
<td>VerifySign</td>
<td>Verifies the sign of the data</td>
<td>Key, signed data</td>
<td>Boolean (verified or not)</td>
<td></td>
</tr>
<tr>
<td>Encrypt</td>
<td>Encrypts the data using the key</td>
<td>Key, data</td>
<td>The encrypted data</td>
<td></td>
</tr>
<tr>
<td>Decrypt</td>
<td>Decrypts the data using the key</td>
<td>Key, data</td>
<td>The decrypted data</td>
<td></td>
</tr>
<tr>
<td>GenerateKey</td>
<td>Generates a key or a pair of keys</td>
<td>Key type</td>
<td>The key(s)</td>
<td></td>
</tr>
<tr>
<td>GenerateCertificate</td>
<td>Generates a certificate</td>
<td>Public Key, owner, authority</td>
<td>The certificate</td>
<td></td>
</tr>
<tr>
<td>GenerateCredential</td>
<td>Generates a Credential through peer trust or certification mechanisms</td>
<td>Credential data</td>
<td>The credential</td>
<td></td>
</tr>
<tr>
<td>CreateVote</td>
<td>Creates a vote</td>
<td>Workspace, Question, timer</td>
<td>The result of the vote</td>
<td></td>
</tr>
</tbody>
</table>

**Required functions:**

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Providing module</th>
</tr>
</thead>
</table>
**Module name:** Security Policies Services  
**Description:** This module provides security policies management.

### Provided functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Consuming modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>listSecurityPolicies</td>
<td>Lists all the available security policies</td>
<td></td>
<td>List of security policy identifiers</td>
<td></td>
</tr>
<tr>
<td>setSecurityPolicy</td>
<td>Adds a new security policy, or modifies an existing one</td>
<td>Security policy Identifier</td>
<td>Boolean (ok or not)</td>
<td></td>
</tr>
<tr>
<td>removeSecurityPolicy</td>
<td>Removes an existing security policy</td>
<td>Security policy Identifier</td>
<td>Boolean (ok or not)</td>
<td></td>
</tr>
<tr>
<td>applySecurityPolicy</td>
<td>Sets up the system according to the right security policy</td>
<td>Security Policy Identifier</td>
<td>Boolean (applied or inapplicable)</td>
<td></td>
</tr>
</tbody>
</table>

### Required functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Providing module</th>
</tr>
</thead>
</table>

---

**Module name:** Privacy Services  
**Description:**

### Provided functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Consuming modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>getProfile</td>
<td>Gets the user profile</td>
<td>Credential, POPEYE user</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Required functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Providing module</th>
</tr>
</thead>
</table>
Module name: Access Control
Description: Provides the authentication functionality to verify the identity of the peer.

Provided functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Consuming modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Authenticates a peer against the correspondent architecture (AAA server or verify the Credential of the peer)</td>
<td>Credential</td>
<td>Boolean (authenticated or not)</td>
<td></td>
</tr>
<tr>
<td>grantAccessRight</td>
<td></td>
<td>Credential, user, Group or SharedSpace, Claimed Rights</td>
<td>Boolean</td>
<td>Group Management Services</td>
</tr>
</tbody>
</table>

Required functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Providing module</th>
</tr>
</thead>
</table>

Module name: Super Peer Trust
Description: Provides a trust mechanism that allows the user to compute the trust level for each of its peers

Provided functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Consuming modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>notifyBehavior</td>
<td>Asks the module to take into account one good (or bad) behaviour of a node</td>
<td>POPEYE user, behavior</td>
<td>Boolean (accepted or not)</td>
<td></td>
</tr>
<tr>
<td>getTrust</td>
<td>Gets the trust level computed for the peer</td>
<td>POPEYE user</td>
<td>Trust level</td>
<td></td>
</tr>
</tbody>
</table>

Required functions:

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
<th>Parameters</th>
<th>Returns</th>
<th>Providing module</th>
</tr>
</thead>
</table>