Filling the Gap with GAAA-P

Gap Analysis of Authorisation technologies and solutions for Optical Light Path Provisioning.

Authors:

Yuri Demchenko <demch@science.uva.nl>

Bas van Oudenaarde <oudernaarde@science.uva.nl>

Leon Gommans <lgommans@science.uva.nl>

Abstract

This technical report will analyse the current state of art regarding authorisation- and policy enforcement technologies. With a special focus on the RFC2904 GAAA Authorisation framework, it will consider applications for network resource provisioning and on-demand Grid- and Web Services Security technologies for dynamic security association management and resource virtualisation. Additionally this report will consider the use of a VO model, which allows the creation dynamic security associations in complex resource provisioning situations.

The report continues with identifying gaps between desired functionalities and the current state of art. It will attempt to define functional requirements and components of an authorisation- and policy enforcement services infrastructure for Optical Light Path Provisioning (OLPP). This definition will be used as case to illustrate complex, multi-component resources provisioning.

Special attention is given towards adding functionalities and components to the GAAA Authorisation framework and the corresponding UvA GAAA toolkit developments to address complex network resources provisioning problems. A major addition is the definition of a special GAAA profile for provisioning (GAAA-P). The development of the GAAA toolkit is part of the GigaPort NG Research on Networks project.

Table of Content

1 1. Introduction

The optical network-provisioning model differs from the model used in traditional IP networks. The traditional model uses connection-less packet switching using routers between shared links carrying IP traffic. Optical networks, using technologies such as (D)WDM, Sonet/SDH, have been evolving around different provisioning and bandwidth allocation models that are optimised for delivering dedicated light paths between data-intensive applications such as found in the Grid. In order to efficiently use the available link capacity across long distances, applications may need to use non-stand transmission protocols. Although typically based on the Internet Protocol, the different and more aggressive (re-) transmission behaviour does not mix well with standard IP traffic. Applications may need congestion free, reserved bandwidth channels, which are not shared with other applications. These kinds of applications will require a different network-provisioning model from those built around the principle of sharing the same communication channel using technologies such as ATM, differential services based QoS, or multiprotocol label switching where the network carries the intelligence. The provisioning model should allow user- or application control over the underlying resource- and service characteristics. In provisioning network resources, rather than network services (typically run by the provider on the resources), there is need of adding business logic and context to the provisioning process, which should be capable to offer the flexibility to manage the formal user-provider relationship.

This document proposes a set of functionalities, distributed over a number of major components, which handle authorisation and policy enforcement of a complex services infrastructure for Optical Light Path Provisioning (OLPP).

The analysis starts with an overview of a multi-domain OLPP provisioning model that is currently being developed in the SURFnet GigaPort6 project. This model is used to specify general requirements for a number of security services (authorisation services in particular) that supports dynamic user-controlled resource provisioning.

Section 3 provides overview and analysis of the basic Generic AAA Authorisation framework functionality and existing attribute management frameworks and solutions. This overview includes the Internet2 Shibboleth based Attribute Authority Service (SAAS) and Virtual Organisation (VO) management service for Grids. Design and implementation suggestions are provided for each of discussed frameworks.

Section 4 introduces new functionalities and associated components that need to be added to the GAAA Authorisation framework and the GAAA toolkit to address complex network resources provisioning requirements and specifics. These extensions are proposed as a definition of a GAAA profile for provisioning (GAAA-P). Additionally, suggestions about using VO model for creating dynamic security associations in complex resource provisioning are provided.

This document is part of Deliverable 3.3.2. of UvA's RoN project plan for 2005.

2 OLP provisioning operational model in GigaPort NG RoN

This section provides an overview and analysis of a multi-domain OLP provisioning model currently being researched for the Research on Networking part of the SURFnet GigaPort6 project [1]. This work is coordinated with other European projects, in particular GEANT2 (GN2) user-controlled network resource provisioning and AA activity [2, 3, 4] and the European NREN's AAI Initiative. This work may be used to specify general requirements towards security related services, in particular and authorisation services, in order to support dynamic multi-domain user-controlled resource provisioning.

2.1 Multidomain reservation operation in GigaPort NG RoN

2.1.1 Components

A proposed model for inter-domain OLPP is shown on Fig. 1 [1]. The inter-domain provisioning process can be split into the following components. These components are designed to interact in different ways in order to implement different provisioning models and OLPP algorithms[1]:

- **A User,** represented by two components:
- 1) A user terminal system (identified in network connection by a network port)
- 2) A user client, which acts on behalf of the user in all interactions with the provisioning system.
- A user may contain a Broker, which represents a user based on delegated rights. A user may also act independently. All interactions are based on a predefined agreement.
- **A Network domain**, consisting of a meshed network of optical links. A connection is defined by input port X and output port Y. (The shown index indicates: 1 - user/originator domain; i – intermediate network; r – resource domain)[2].
- **A local domain management and control system**. The control part contains the state of a particular connection whereas the management part is capable of providing information about- and manipulating the state of a particular connection.
- **The Inter-domain Connection Creator (ICC).** Its function is to create and manage connections between two separate domain ports, depending on the used interconnection route optimisation algorithm, for example:
- 1) output-port Y1 of the originating domains (the user's home domain) and input port X3 of the target/resource[3]
- 2) input-port X2 of the controlled domain and input port of neighbour domain X3, (X2<=>X3)
- 3) output port Y1 of the controlled domain and input port of neighbour domain Xj, (Xi<=>Xj)

[1] It assumed that connection is always initiated by User (unless "call-back" model is implemented). However, OLP building may start from both sides of the connection initiator/consumer and resource provider. For the initial model we assume that Interdomain connection creator from User connection provider domain acts as an OLP connection requestor.

[2] There may be models that allows multiple interdomain connections/links, what will be defined by implemented optimisation and routing model.

[3] For simplification we will use word "input" and "output" in relation to connection/path from the user side in direction to the resource. In respect to network traffic they both can accept bi-directional traffic and in physical respect can be bi-directional link. Terms "ingress" and "egress" can be considered as an alternative.

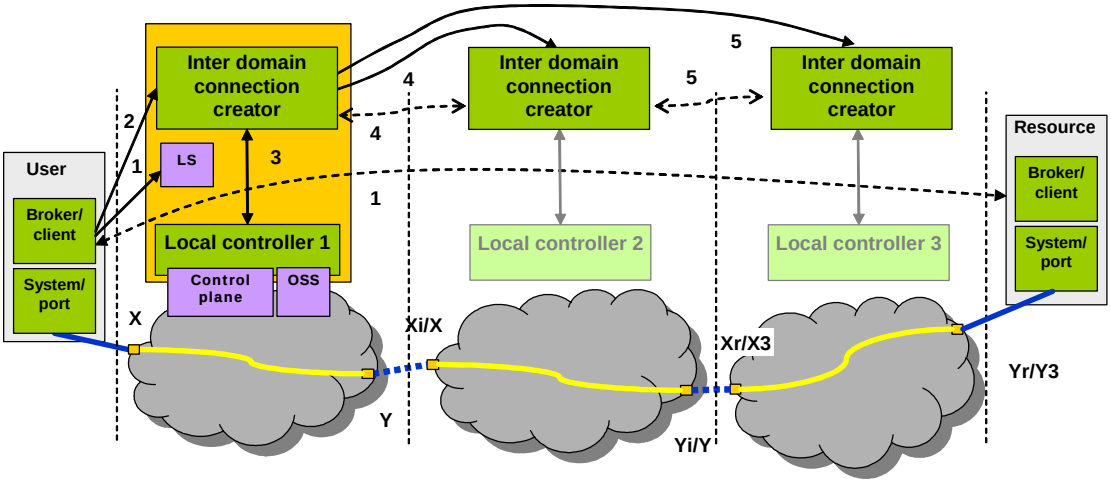

Fig.1. OLP provisioning operation

2.1.2 Interactions

In OLP provisioning models, the control plane is typically separated from the physical network elements (switches and optical links). Two OLP provisioning models can be distinguished which differ from the discovery and path creation/reservation perspective. A path can either be created in:

1) a "hop by hop" fashion - where each domain understands which domain is next in the path towards reaching the destination. There is no central awareness of the path.

2) an "agent based" fashion - where an agent in the originating domain or a central agent has total awareness of the path and establishes contact with all involved domains along the path.

During both types of OLPP process, the following interaction phases can be considered:

	Phase	Description
1	**Route Discovery**	A Lookup Service (LS) will provide a list of one or more domains that can be used and contacted to reach a particular destination
2	**Inter-domain connection creation**	Inter-domain Connection Creators inside each domain will exchange information to determine an optimal path based on request and information available from the individual domains.
3	**Path reservation**	Upon successful establishment of an optimal path across the domains, the path will now be committed by to a notion of a reservation.
4	**Reserved path provisioning**	The reservation will be implemented within the individual networks making the path available for usage.

2.1.2.1 Route discovery phase

The application broker or user client requests (1 fig. 1.) a path via a Lookup Service (LS) to a target system- or resource. The LS returns one or more networks or OXPs (Optical eXchange Points) leading to the resource connected to the network. The LS may either provide detailed routing information about the involved domains or provide minimal information, just indicating the availability of a path that meets with User requirements. The amount of information that can be provided will be dependant on how much information each individual domain is willing to share with other domains at this point of time. Autonomous network domains typically do not want to share details regarding their current state of their topology. LS may or may not use user credentials at this phase to obtain different levels of information.

For further analysis, we will use a simple model containing three domains related to User provider/entry point, Resource provider/entry point, and intermediate optical network/cloud.

2.1.2.2 Inter-domain connection creation phase

A Request inter-domain connection (2 fig. 1) is send by the User. This will initiate a route creation process between the User and the Resource with specific set of parameters. A metadata model for this information exchange will need to exist. We assume that a User is connected to a known port X1 of the provider Domain 1. Domain controller 1 provides internal optimisation from user port X1 to the Domain 1 output/egress port (Y1).

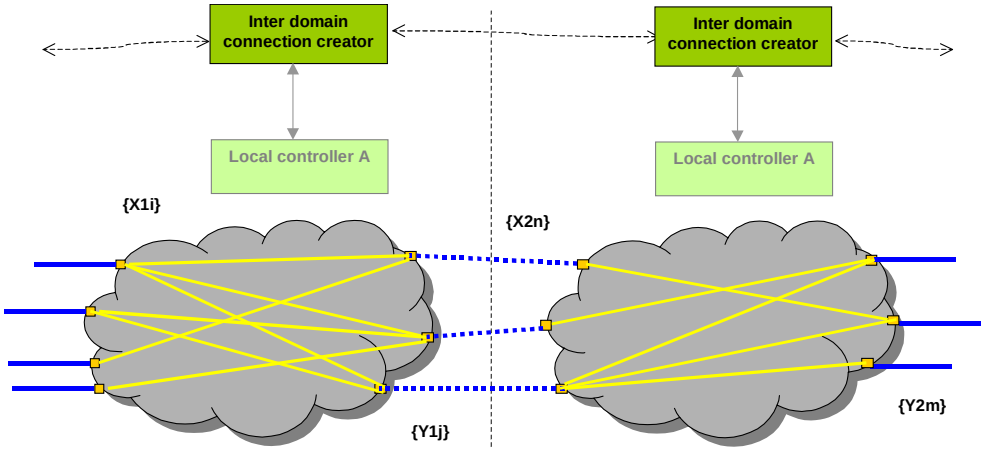

Figure 2. Interdomain-intradomain trespassing graph

The Inter-domain Connection Creator (ICC) calculates an optimal inter-domain path through the interconnected domains via the available connections between two domains by:

- Requesting path cost, link capacity parameters, link numbers and other relevant operational details. This information is expected to be inferable from combining the port information from both the ingress and egress ports of the connected domains. We assume that every individual ingress port is permanently connected a corresponding egress port at the corresponding domain.
- Request any additional information that is required from the User Client or Broker in order to create an optimal path, including the type of credentials that must be provided.

Note. Because of the possible different costs of partial links and different QoS requirements for possible temporal reservations, the ICC may require user credentials and attributes which user must be obtained in advance.

Phase 2 will resulted in two types of information:

- one or few optimal paths that correspond to user/requestor defined criteria to points/ports that is defined/required for suggested reservation algorithms that will be used in Step3 – Reservation, e.g. full path or just path to the next domain;
- set of descriptive attributes and credentials from User in order to get authorisation to use (have access to) particular calculated path.
- For example, such set of credentials may contain:
- AuthN credentials from a trusted CA or Identity Providers (IDP),
- Attributes describing a membership of a virtual organization or federation
- AuthZ tickets obtained from trusted parties by means of a domains AuthZ service
- Any other attributes confirming user credibility or capability issues by mutually trusted authorities.
- A common working model for managing such credentials for well known users can be a VO concept. Anonymous or users that want to provide only few attributes may use some payment infrastructure based credentials based on real monetary units.

2.1.2.3 Path Reservation phase.

This part may be more or less predefined depending on what path calculation algorithm was used in previous stage. Reservation process may use the same sequence as in previous step, however in this stage two different interaction models can be defined:

1) **Agent based allocation:** User ISP Domain ICC1, acting as an agent to the user, contacts all intermediate ICC's to reserve requested intra-domain connection(s) between pre-defined input/ingress and output/egress ports Xi <=> Yi;

2) **Hop by hop allocation:** User ISP Domain ICC1 contacts only the neighbour ICC to request connection to the next suggested domain to create a intra-domain connection between pre-defined input/ingress and output/egress ports Xi <=> Yi;

The following issues are considered as common security related elements/components in the reservation process:

- **Pre-requisite credentials to initiate a reservation:** To initiate reservation process, user must have and (optionally) provide to (a trusted) broker the appropriate credentials that may include:
- 1) The user identity.
- 2) Any attributes confirming permission, capabilities, roles.
- 3) The level of credibility in a form of personal credit, usage limit or membership to a brand or federation that will vouch for consumption of resources. Note that these credentials can be obtained by a user in advance or may be requested by a broker on behalf of the user.
- **Initial domain path request credentials:** When requesting a reservation from an Inter-domain Connection Creator (ICC), a user client/broker must provide the agreed credentials. This may be a comprehensive document or just a few attributes but always enough information about how additional (not included) credentials can be obtained. Note that part of credentials location information can be provided explicitly by referencing AuthN service of issuing CA or user IdP, or implicitly by including information about user membership in one of associations securing user request/operations (e.g., credit card number) or having an agreement with a interconnecting domain (represented by an ICC, VO or federation).
- **Inter-domain path request credentials:** ICC when requesting path reservation from other domains may use:

- user original credentials (bound with ICC's own credentials)
- impersonate the user, using one of the delegation procedures (limited delegation or full impersonation).
- use only own credentials securing this process by existing agreement between user and domain provider, depending on level of cooperation or agreement between user and OLP provider
-
- **Initial trust relationships:** Inter-domain path reservation requires existence some form of initial trust between these domains that can be provided by one of interdomain trust management mechanisms like:
- A commonly used payment system (credit-, debit-, charge card, micropayment system, e-purse),
- VO or any other virtual group association
- DNSSEC mechanism for public key distribution
- AAI mechanism based on secure attribute providing systems like Shibboleth.

  Note. As it will be discussed later (see section 3), VO and DNSSEC mechanisms can be combined by assigning DN to registered VO and storing VO public key in DNSSEC record.

- **Absence of a trust relation:** It is assumed that an ICC can request reservation only from domains with which it has a trust relation. In case of absence of a direct trust relation, an ICC can request the reservation from other ICC's that may have trust relations with required segments of the whole calculated path. A particular case of this situation is when ICC from user provider domain has relations and knows only about connection to neighbour domain and relies on it for destination path provision.
- **Delegating path reservation:** Transferring or delegating further path or resource reservations to another domain implies explicitly (via requestor credentials or Identity delegation) or implicitly (via pre-existing agreement) a trust relation between cooperating domains. Identity or credentials delegation procedures may be related to general Identity management functionality and should be governed by a Delegation Policy. Delegated identity may use initial requestor credentials and name (or even namespace in general) together with domain ICC credentials, or new credentials issued by local domain Identity Provider.
- **Authorisation within a domain:** When processing path reservation request, remote domain/ICC will evaluate requested resource/path and user credentials (in case of direct reservation by user domain ICC), or requestor credentials (in case reservation delegation) against available (network) resources (pre-reservation may be considered but should be proved in some additional provided information) and reservation and provisioning policy, part of which is security related and concerns about issues related to user identity, user membership on associations/federation defining network resources management or consumption. During this evaluation, ICC may request:

  1) additional information or credentials from user/requestor,

  2) additional credentials and/or presented credentials confirmation/verification from user/requestor related security services providers and authorities.

  Note: This means that final reserved path will depend on results of user request authorisation in domains and therefore from applied access control policies in domain.

- **Confirmation of the reservation:** In case of positive decision, the ICC makes reservation of the requested resource/path and commits it with user/requestor or broker credentials (e.g., by marking them with requestor ID and linking it to billing attributes). As a confirmation of the reservation, ICC returns a confirmation ticket in

the form of secure assertion signed by the ICC's own credentials. The reservation confirmation ticket could contain below information, which identifies this reservation:

1) Reservation ID number.

2) Initial request or its ID reference, which may contain any additional information that must be provided.

3) Reservation conditions, details and limitations: e.g. the scheduled time-slot, additional credentials or tokens required to use the reserved path/resource, operational details describing location of resource, billing information, etc.

4) The obligations of the requestor in respect to using the resource: e.g. logging requirements, cost, payment details for usage, etc.,

5) Audit related or historic information that may include initial request, original credentials and/or the used delegation chain, etc.

- **The reservation ticket** returned to the requestor represents an authorisation, which can be used to access the resource or path. To ensure authenticity and integrity, reservation ticket returned to user or user broker must contain all tickets and assertions confirming the entire path reservation, including directly reserved segments (referring to segment reserved by user domain ICC) and reference to reservation tickets made by other ICC in behalf of user's ICC.

Note. The shown reservation algorithm has similarities with the Source-Routing Algorithm (SRA). The algorithm is different in the sense that in the inherently loosely coupled environment the resulted path will contain not only a suggested route, but also returns policy criteria to the requestor. For discovering possible inter-domain connections it use BGP style algorithms. This means a reserved end-to-end path will depend on the results of a user request for path authorisation in all involved domains. During the reservation process, the ICC can request required security operation related to user AuthN, AuthZ, identity and attributes verification from domain AAA/security services that may constitute inter-domain AAA infrastructure. Defining requirements to AAA services/infrastructure and API to AA(A) services is a subject of this report.

2.1.2.4 Reserved path provisioning phase

The provisioning process is based on the reservations made at the previous step and uses information about confirmed reservation in particular domains. When requesting a particular path via a domain, the user must provide the necessary credentials and any additional information requested in the reservation confirmation response. One may refer to a given reservation ID, and maybe required to provide the original reservation ticket response. The following operations may take place during actual provisioning process:

- **Simple referral:** In a simple case, the provisioning request and provided credentials will confirm the previous reservation and OPL can be simply provided to user based on stored reservation information and a local request verification at the ingress point of the network(s).
- **Evaluation of additional details:** In the case that provisioning requires the evaluation of new information or credentials, additionally requested in reservation confirmation (e.g., equipment certification, scheduled traffic, etc.), this may require additional evaluation of new information and contacting AAA services, etc.
- **Fallback conditions**: It may happen that a particular domain ICC cannot provide reserved resources, whilst accepting the request. In such case, ICC may offer another option to user and initiate additional negotiation cycle if such service is contained within the Service Level Agreement between the user and provider. A renewed reservation ticket should be sent back to user or requestor.

Note. The provisioning process and negotiation process during a fallback scenario, can be managed completely in the control plane. It could also be combined with the traffic/data tokenisation. The obvious benefit of tokenised data flow is higher dynamic of possible network re-configuration inside single

managed domain. In this case, tokenised traffic uses tokens created/generated at the reservation stage and consequently cached by related ICC's or their AAA services.

2.2 Security requirements to enable Inter-domain OLPP Operation

This paragraph will summarise the various security requirements, which are needed to allow inter-domain OLPP to happen. The use of terms MUST, SHOULD and MAY are in accordance with RFC2119.

Authentication	Authentication (AuthN) is the first stage in access control. It is performed to establish a trusted electronic identity of the requesting user. The user MUST present credentials, which has been issued by a person or organisation which MUST be trusted to check a persons identity according to pre-established procedures (e.g. check identity based on a government issued photo ID and/or credentials from other recognised and trusted registries).
R1.1	An AuthN SHOULD yield a result in the form of: 1) An explicitly provided AuthN ticket or token. 2) An implicit allowed access to the protected resource or system. In the latter case the AuthN is confirmed by the start of a session under a users personal- or group ID.
R1.2	In a multi-domain scenario, the (initial) user authentication in a User Home Organisation (UHO) SHOULD be allowed to used a user-centric Trust Anchor (TA), with the user as a root of trust for all following identity translation and attribute management operations. This SHOULD therefore be considered as the most sensitive procedure/operation. However, IdM or Authorisation (access control) services MAY also verify and request confirmation of the initial user AuthN.
R1.3	In a multi-domain scenario, only the User Home Organisation SHOULD provide the authentication service. In such case, additional security services MUST provide inter-domain user identity, credentials and attributes translation.
R1.4	In case of using more extended functionality with Identity management, AuthN SHOULD be allowed to be a basis for issuing user identity credentials and/or user attributes

Identity Management	Identity management MAY be used as an additional step in Access Control after Authentication and before Authorisation to provide: 1) Single Sign-On (SSO) service in environment with multiple identities (of the same user/requestor) and also multiple domains. 2) flexible user attributes management bound to his/her identity. 3) manage and provide context to user federations and associations, 4) enable user identity delegation both in single domain and multiple domains. Note. A Virtual Organisation (VO) management system MAY be considered as a part of the general Identity Management.
R2.1	Within multi-domain scenario's, each domain MAY contain an Identity Management service (IdM) as to provide: 1) inter-domain or inter-organisational identity translation. 2) independent management of domain's users and resources membership, i.e. associations and federations 3) (user-centric) inter-domain trust management. Note. This functionality can be abstracted to the Security Token Service (STS) as a generic service.
R2.2	An IdM service SHOULD be allowed to issue user credentials (that can be both a user Id and attributes) based on user AuthN or other form of identity credentials. The IdM SHOULD rely on existing trust relationship with AuthN service or other IdM services. There MAY be different models for trust management when issuing identity credentials. 1) The IdM service in a UHO domain MAY rely on existing trust relations between AuhtN services and IdM, e.g. having the same root CA. 2) An IdM service in a remote domain MAY use a direct or indirect trust relationship between UHO AuthN or IdM. Special (business/provisioning) agreements between interacting domains SHOULD define the acceptance policies for remote AuthN or Id credentials. In particular, the acceptable strength of AuthN, or the acceptable chain of trust/credentials, and the Identity delegation conditions (e.g., limited delegation, or full impersonation). 3) Federations or associations in which a user has a proven membership, that are supported by special a membership services such as the VO Membership Service (VOMS), MAY be used for inter-domain attribute- and trust management.

Authorisation	The Authorisation function protects a resource by defining and enforcing access control policies. Authorisation is based on the identity of an authenticated user or requestor. The identity is represented explicitly in a form of AuthN or Id credentials, that are issued by a trusted AuthN or IdM service. An authorisation service evaluates a request for a resource or path containing user or requestor credentials according to the resource domain's AuthZ policy, which defines access control rules based on user attributes (group membership, roles or other capabilities).
R3.1	User attributes MAY include user membership attributes from an association or federation that governs network usage, security or imposes resource consumption constraints. During the policy evaluation, an AuthZ service MAY therefore request additional information such as: 1) budget related or accounting type of information 2) more user specific credentials, 3) confirmation information from a users security services provider, authorities, resource managers etc.
R3.2	An AuthZ service MAY include one or more of the following functional modules: 1) A PEP – Policy Enforcement Point 2) A PDP – Policy Decision Point 3) A PAP – Policy Authority Point
R3.3	When operating in an inter-domain, multi-domain provisioning scenario, an AuthZ service MAY request evaluation of some part of a request by a different AuthZ service, possibly located in another domain. However, in order to protect the integrity of an AuthZ decision, the final composition of the decision MUST be performed by the PDP that received the original request.
R 3.4	Based on a successful authorisation, the AuthZ service MAY issue an AuthZ ticket that MAY be used in subsequent AuthZ requests or MAY be used by the ICC as a base for issuing a reservation ticket. It is essential that,, when presenting AuthZ tickets (or tokens), the ticket or tokens authenticity and integrity within subsequent requests MUST be evaluated by a resource's PEP. For this, the PEP MUST have a secure trust relationship with the PDP in order to exchange the corresponding key material.
R 3.5	An AuthZ service MAY either operate in pull or push mode Note. One of the push model implementations MAY be based on using AuthZ tickets obtained in advance from the resource's AuthZ service or other trusted AuthZ service, e.g. belonging to a VO or other user and resource federation.

R 3.6	An AuthZ service MAY issue provisional authorizations during the reservation stage. Authorizations MAY be altered or made more specific during the provisioning stage. This requirement MAY also imply evaluation of different criteria and applied policies during the reservation and provisioning stage. E.g. a reservation request may specify only basic requirements towards the resource. Only during the resource allocation phase, a user/application will expect confirmation from the particular resource, which MAY also imply that a different set of user attributes are required to be offered.
R 3.7	Mutual AuthZ MAY be required, E.g. the receiver first asks the sender to receive certain information. Subsequently, when ready, the sender explicitly asks permission from the receiver to send. Applications within the medical- or banking area, are likely to pose such requirements.
Attribute management	User (and resource) attributes MAY be managed separately by Attribute Authorities (AA) but still in conjunction with user or the identity (resource). Attribute management MAY be delegated to an association or federation membership service, such as a VO in Grid applications or InCommon Federation in Internet2 Shibboleth infrastructure.
R 4.1	One of the AA infrastructure specific functions is the management of attribute namespaces that are shared between interacting members or domains, or can be mapped/translated by IdM services. For this purpose, the AAI SHOULD provide potentially mapped attributes/namespaces that are directly understood by IdM services or can be mapped (based on known/pre-established relations).
R 4.2	The validity and trustworthiness of attributes will have effect on an AuthZ decision's trustworthiness and MUST therefore be considered in the overall trust-relationship analysis.
R 4.3	A two stage reservation and provisioning sequence MAY require different strength of user ID and attribute confirmation.
Trust management	All security related operations and resource allocation operations MUST be based on established and traceable trust relations based on mechanisms such as PKI, SPKI, shared secrets, etc.
R5.1	Trust relations, being instant for any particular service invocation, can be invoked dynamically, however SHOULD rely on more static pre-established relations that can be used for initial trust introduction. For example, use published service public key to initiate session to exchange more secure credentials, etc.

R 5.2	A VO MAY be used for inter-domain/inter-organisational trust management by providing trust anchor for inter-domain credential management.
R 5.3	DNSSEC MAY contain a VO's or Federation's public key bound to the domain name and MAY be used for user/originator attributes verification and/or initial trust introduction.
R 5.4	All security valid decisions, e.g. delegation, AuthZ or reservation, and credentials MUST have an unbroken and auditable chain of trust.
Federation management	Inter-domain/multi-domain scenarios require some form of federation to be established for user identity-, attribute- and trust management.
R 6.1	Federations that MAY be used for OLPP are inter-university federations like Internet2 InCommon, or VO's originated from various Grid projects such as DutchGRID, LCG etc. In the particular case of inter-domain trust management, such federations SHOULD be useable for attribute management and/or trust management.
R 6.2	Federations, such as a Grid VO, SHOULD be allowed to provide a communication context for services and applications interacting through (enterprise) firewalls.
AuthN/AuthZ service API	AuthN/AuthZ services API is required to flexibly and dynamically request AuthN, AuthZ and Attribute services from network services and applications.
R 7.1	AuthN/AuthZ services API SHOULD define protocols, request- and response message formats, basic commands and extensibility procedure, basic configuration profiles, namespace resolution/management and enumerated attribute values assignment.

Conceptual issues	A OLPP management structure MUST fit into a broader framework within a federative environment. Certain concepts SHOULD be clear before a OLPP service and control structure can be established.
R 8.1	A VO infrastructure organisation- and management architecture and model SHOULD be established before defining the framework, architecture and implementation of a user/application controlled OLP provisioning environment. Legal, Economic and Administrative responsibilities and interactions between federative elements MUST be clear.
R 8.2	The VO concept used for multi-domain and inter-domain AA services operation and trust management SHOULD be investigated.

3 Existing solutions and gap analysis for Authorisation infrastructure and related services

3.1 Generic AAA Authorisation Framework as a basis for an OLPP AAA Authorisation infrastructure

This chapter will describe the basic Generic Authentication, Authorization and Accounting (GAAA) Framework with a focus on Authorization. This GAAA framework is used to describe authorization sequences enabling the access and usage of a Lightpath.

3.1.1 Basic GAAA Authorisation framework operational models

Generic AAA Authorisation Framework [5] and its specific implementations for network provisioning [6, 7] and define three basic operational models that describe interaction (in sense of request/response sequences) between a user, a service or resource provider and AAA Authorisation service acting as an Authority. These sequences have also been used as basis for the Conceptual Grid Authorization Framework and Classification document [8].

The push authorization sequence.	Within the push (or token-) sequence, the User first requests an authorization from a trusted Authorisation service that may or may not honor the User's request. It then may issue and return some kind of Authorisation assertion (a secured ticket or token) that acts as a proof of right or as asserted list of requestor capabilities. Typically such an assertion has an associated validity time window. The assertion may subsequently be used by the User to request a specific service by contacting the Resource. The Resource will accept or reject the authorization assertion and will report this back to the requesting Subject. The Resource must have been provisioned with the appropriate key material to recognize the appropriate assertions.
The pull authorization sequence.	Within the pull (or outsource-) sequence, the User will contact the Resource with a request. Before admitting the service request, the Resource must contact its Authorization service. The Authorization service will evaluate the request against a specific authorization policy and will return an authorization decision. The Resource will subsequently grant or deny the service to the User by returning a result message. The Resource, which enforces a policy, effectively out-sources a policy decision.
The agent authorization sequence.	Using the agent (or provision-) sequence, the User will contact an Agent, which will handle the User's request for the particular Resource. The Agent is trusted both by the User and the Resource. The Agent will make an authorization decision and, using its own or User-delegated credentials, it will contact the Resource to provision the requested service. The Agent will provide the User with details on how to contact and use the Service.

The three basic authorisation sequences described above are elementary abstractions of more complex real world examples that normally combine the basic sequences. It may use various protocols and message formats to handle and secure user credentials and requests.
Although more functions can be found in both an Authority and a Resource, an Authority typically acts as a Policy Decision Point (PDP) and a resource acts as a Policy Enforcement Point. In the subsequent discussion we may use the term PDP and PEP to represent functions inside the corresponding entities.
Some examples of combining basic authorisation models to achieve performance or security benefits are discussed below in relation to available two major GAAA implementations for Bandwidth-on-Demand (BoD) reservation [6, 7] and for RBAC-based authorisation service for collaborative applications [9, 10].

3.1.2 Abstract GAAA operational models for complex resources

To provide examples, this chapter describes some cases based on current research performed at the UvA. The research is aimed at the development of operational models based on the GAAA tools to provide access and usage control of a complex set of resources in a distributed heterogeneous environment. Such an environment can be characterized by:

- Access control- and usage policies are defined by **multiple policy instances**, governed by different authorities and captured in different formats. Such environment can however be structured and ordered as a combined policy.
- Multiple PDPs and PEPs may **interact in sequences**, which can either be flexibly configured or pre-defined. The sequences can be described using elements of the GAAA authorization framework.
- A **network of PDPs and PEPs** can operate in the push-, pull and agent modes. An ordinary RBAC may require the agent mode to be supported by push functionalities. A basic provisioning model that can be split into the discovery and reservation stage, which operates both in agent mode where the actual service delivery is supported by pushing an authorisation credentials/ticket/token.
- PDPs and PEPs **elements can be part of a Resource, User or a Service**. A set of PEPs and PDPs can together create a control plane (like in OLPP).

UvA aims to extend existing high-level GAAA Authorisation models [8] with components to define and handle workflows in complex resource provisioning scenarios.

This work is based in initial research and development done by the UVA SNE Group in the framework of various projects such as GigaPort-NG, STARplane, VL-e, Collaboratory.nl (CNL) and NextGRID. Use cases drive the requirements such as the CNL RBAC use case [9, 10].

3.1.2.1 Use cases illustrating the GAAA framework.

Two basic use cases/models are discussed in this section:

1) combined agent-push (provisioning) model for complex resources

2) combined pull-push (RBAC) model for multi-layer resource protection.

An important component of both combined models is the use of authorisation tickets and tokens for security context handling and performance optimisation.

Figure 3 below illustrates an abstract access control model that combines two generic AAA Authorisation sequences: the agent sequence and the push sequence. Such model is typically found within Bandwidth-on-Demand (BoD) use cases. The type of complex service that is collected and provisioned is less relevant and can therefore be applied more generally.

In the agent model, the PDP orchestrates a (complex) service request on behalf of the Requestor. The policy, in such case, can be considered as a “driving policy” and as such represents elements of the total workflow of the system. In case of complex resource/service request, a sequence of PDP’s may create a flow of recursive policy evaluation chains. The PDPs may use a set of PEPs to enforce the policy at different resources and services. It is assumed that each PDP can request other PDP’s for evaluating some of the policy components for the specific resource. In more details PDP and PEP interaction is discussed below for the combined pull-push model.

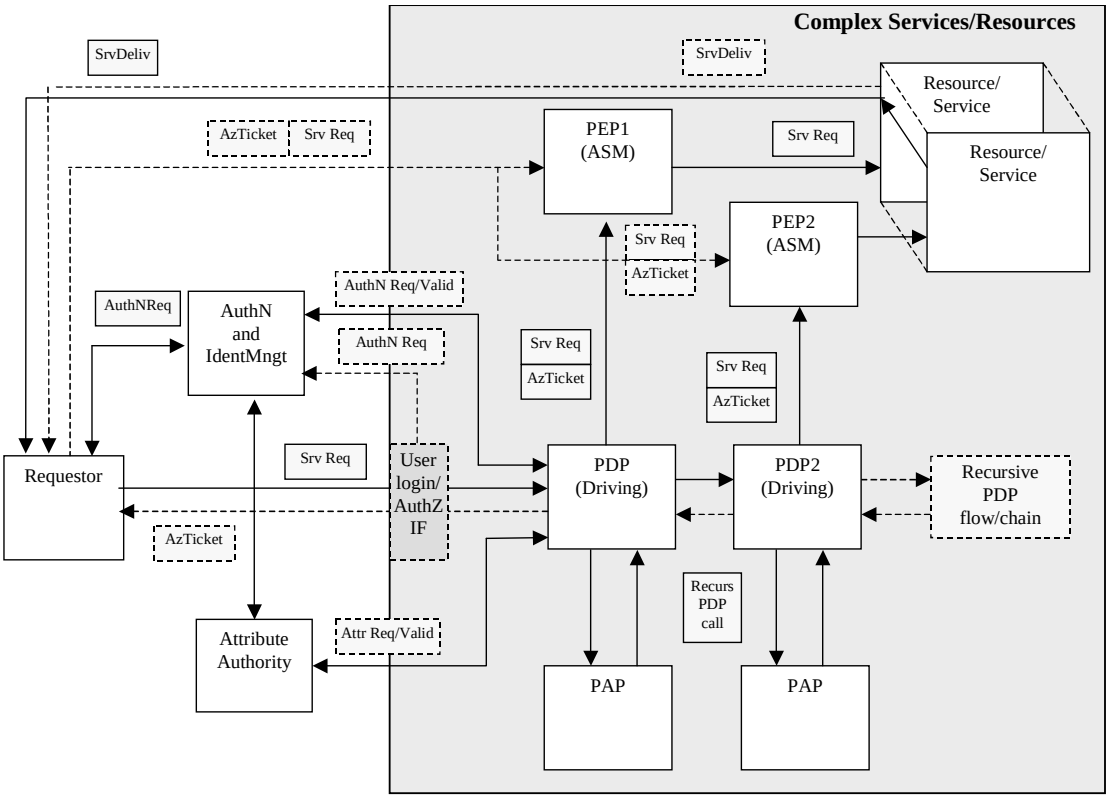

Figure 3. Major components of the complex Resource/Service Authorisation service (combined push and agent model, complex/multi-component resource)

Figure 4 below illustrates a typical RBAC authorisation model that implements pull model of the generic AAA Authorisation framework and may also use the authorisation ticket “push” functionality to optimise performance. The picture also explains how the policy combination can be done via PEP chaining/sequencing and/or PDP nesting/recursion as a common component for all GAAA operational models.

A detailed policy enforcement process analysis must formulate security constrains for a use case that involves multiple policies evaluation with a combination of multiple PEP’s and PDP’s. The aim of such analyses is to preserve a site or resource access control integrity.

The proposed approach retains integrity of the combined policy based decision. Although the PDP, when evaluating a request from the PEP, may call for external evaluation of some other policy components, it will make its own final decision and only it will return a reply to the calling PEP, which acts as a gateway to the initial request.

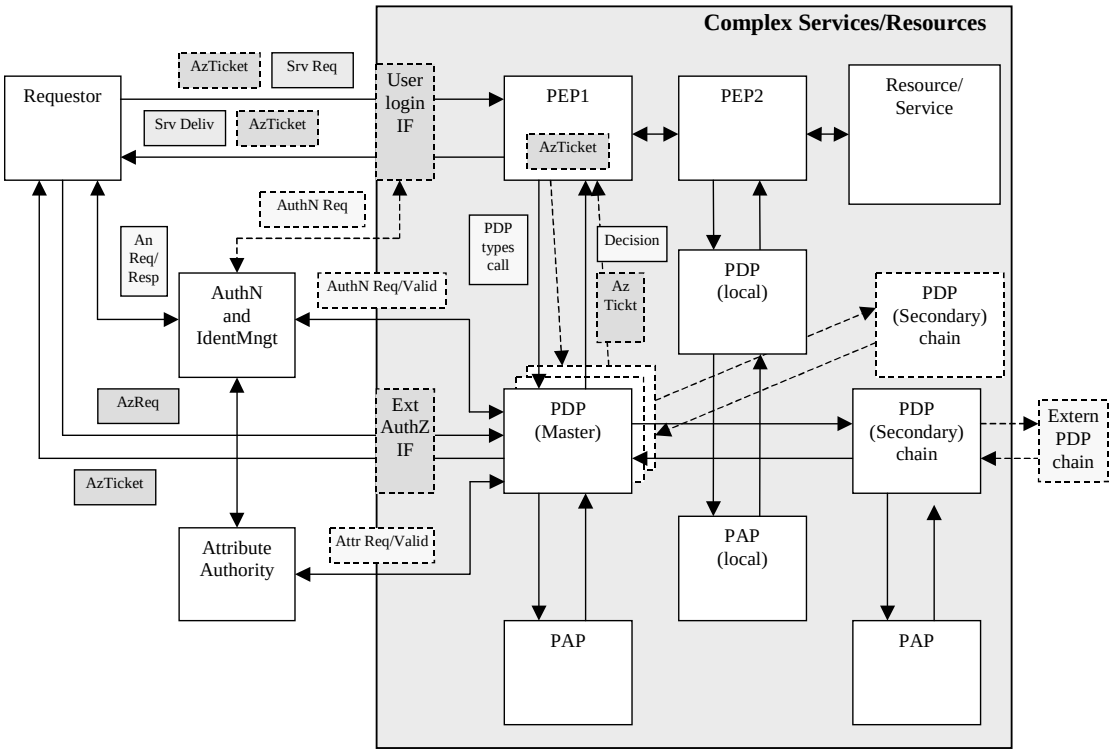

Figure 4. Multiple/multi-domain policies combination in complex resource/service Authorisation service (combined pull-push model)

The Requestor requests a service by sending a service request ServReq to the Resource's PEP providing as much or as little information about Subject/Requestor, Resource, Action, and additionally Environment as it decides necessary according to used authorisation model and (known to the Requestor) local policies.

In a simple scenario, the PEP sends the decision request to the (designated) PDP and after receiving a permissive reply from PDP, it relays a service request to the Resource. The PDP identifies the applicable policy instance, retrieves required context information and evaluates the request against the policy. During this process it may need to validate the presented credentials locally based on pre-established/shared trust relations, or call external Authentication and Attribute Authorities.

Described above process represents a basic scenario. However, in a more complex and open environment, the PEP may receive requests that have different formats and semantics (namespaces) or may refer to policies stored in other policy repositories. In such case, the PEP should have a possibility to relay a decision request to an appropriate type of PDP, capable of handling the entire decision request. It is essential that a request is evaluated as a whole and an ultimate decision is made by a single PDP. This PDP may however make calls to external PDP's to evaluate some request components and process their decisions as components of the general policy evaluation process. The PDP that makes a final combined decision can be defined as a master PDP and it needs to have mechanisms in place to preserve integrity of the final combined decision. In case when an integral request evaluation is not possible, a fallback with possible roll-back scenario's should be suggested or executed. Responses such as "not applicable" or running through a "deny-override" evaluation chain for the partial request components should be possible.

Existing (open) policy expression formats, such as XACML and our AAA driving policy language provides mechanisms that allow a particular policy instance to refer to another policy instance. Complex combined policies can be created by a PAP on a PDP policy request. or processed by the PDP by requesting required policy components during the request evaluation.

As a trade-off of being open by using separate access control components and open standards, the solution above has known performance concerns. The resolution of this problem is seen in combining pull and push operation models. Since the decision is made by the PDP, an AuthZ ticket can be issued and used in the next similar or repetitive actions requests for the duration of a ticket's validity period. An AuthZ ticket can be obtained via PEP during the first access request or it can be requested from the PDP via external AuthZ interface priory to sending a service request.

In the push model the Requestor first requests an Authorisation decision to obtain an AuthZ ticket, which it will attach to one or more subsequent service requests. The PEP will evaluate the authenticity, integrity and validity of the presented ticket and maybe some additional security credentials that proves correctness of for example the ownership, billing information, service level etc.. However, no other access decision functions should be given to the PEP as a functional component. If there is a need to enforce other components of the site or resource control, like "blacklist", it should be done via separate (local) PEP-PDP chain.

General implementation suggestions.

Described above scenarios are simple ones, but they require that both Requestor and Resource services know explicitly or implicitly the policy, semantics and know or can access the context information. Requestor and Resource should have established trust relation via common PKI or via preliminary shared public and secret keys.

When implementing an authorization sequence, the following issues should be considered (using RFC2119 terminology for the words MUST, SHOULD, MAY etc.):

1. PDP and PAP MUST share a common namespace

2. Policy and respectively PAP SHOULD be referenced in the request message explicitly or known to PEP and PDP a priory

3. Every PEP in the chain of policy enforcement MUST take care of the whole request evaluation/enforcement by calling to a single (master) PDP. A PEP MUST not do multiple decision combination.

4. Only one PDP MUST provide a final decision on the whole request

5. However, PEP MAY have a possibility to request different PDP types based on request semantics/namespace and referred policy. By definition, PEP MUST have an ability to recognise request's context semantics/namespace and convert the initial request format to those accepted by a particular PDP that will handle a particular request.

6. It is suggested, that in general (and to have a possibility to combine pull and push AuthZ models for the performance optimisation) a PEP SHOULD understand and have a possibility to validate an AuthZ ticket issued by a trusted PDP or AuthZ system in general.

7. For this purpose the Requestor MAY request and the PDP MAY issue the AuthZ ticket which the PEP MAY relay back to the Requestor. The AuthZ ticket issued by the PDP SHOULD have validity and usage restrictions and MUST contain all information about the decision and the resource. Depending on the used security context management model, the AuthZ ticket MAY also include all context information about Requestor, its capability/attributes, its Identity credentials (in a form of AuthN or Identity provider token).

8. In the particular case of a dynamic access control policy operational model (so-called "push-policy"), an AuthZ ticket MAY be provided in the form of a (serialised) policy instance that defines exact matching conditions for the Request evaluation. In this case, the request processing SHOULD require only simple operations that can be executed by a PEP with some extended functionality.

9. For future validation of the AuthZ tickets, the PEP MAY cache the ticket locally to speed-up the validation procedure.

10. When using AuthZ tokens, which uniquely reference AuthZ tickets but are smaller and simpler, AuthZ tickets SHOULD be cached by a PEP for future token resolution (or retrieval by token reference).

Specific implementation suggestions for OLPP.

Because the OLPP operation includes at least three stages (lookup, reservation and provisioning/delivering) the following specific issues SHOULD be considered:

- User/requestor credentials and consequently the trust model MAY be different at the reservation stage and at the provisioning stage
- A reservation ticket, used at the resource/service consumption stage, MUST include all reservation tickets for the whole OLP (or complex resource).
- Multidomain OLPP requires inter-domain trust management that SHOULD be solved by establishing a general/common security federation or managed via delegation between inter-operating domains.
- Interdomain trust management MAY be implemented by using an open trust introduction model, for example DNSSEC or Shibboleth.

3.1.3 Tickets and tokens handling with the GAAAPI package

This section provides information about example/prototype implementation of the discussed functionality for AuthZ tickets and tokens handling in the GAAAPI (Generic AAA Programming Interface) package developed as a part of CNL project [10].

Tickets and tokens handling in combined agent-pull-push operation models requires a specific functionality which is not explicitly defined in the generic RBAC and PIM (Privilege Management Infrastructure) models. This functionality can be defined as intermediate between PEP and PDP functionalities but can not be instantiated to just Request the context handling because of its operation may be resulted in definite decision based on local request evaluation (without calling to PDP) against provided AuthZ ticket.

This specific functionality is defined as a **Triage** that provides the following functionality:

- Evaluate the request against provided AuthZ ticket and provide a decision on the requested action or resource.

  Note. In fact, Triage confirms or denies a decision contained in the ticket, although in most cases the ticket will only be issued to positive decision.

- Underlying Triage operations may include: request validation, ticket validation, request classification (to define candidate PDP for processing), etc.

  Note. Such functions in the request (pre-) processing as attributes validation and request should be better attributed to the general context handling functionality that may be related to PEP or PDP.

Although the Triage function provides initial request evaluation, it should be considered as a function called from the PEP (and optionally from PDP). The justification for this is that from the design viewpoint, the Triage should be separated from converting application specific request format/context to those that corresponds to the ticket or pushed policy format. Such conversion is a generic function of PEP.

Under some considerations, the Triage functionality can be attributed to PDP (or PEP) but as it is discussed above its specific functionality is different from the generic PDP and PEP

functionality. Actually, the Triage implementation in GAAAPI allows calling Triage function from PDP or PEP.

Picture 5 below illustrates how the Triage interacts with the PEP, the PDP and other generic RBAC and major GAAAPI components.

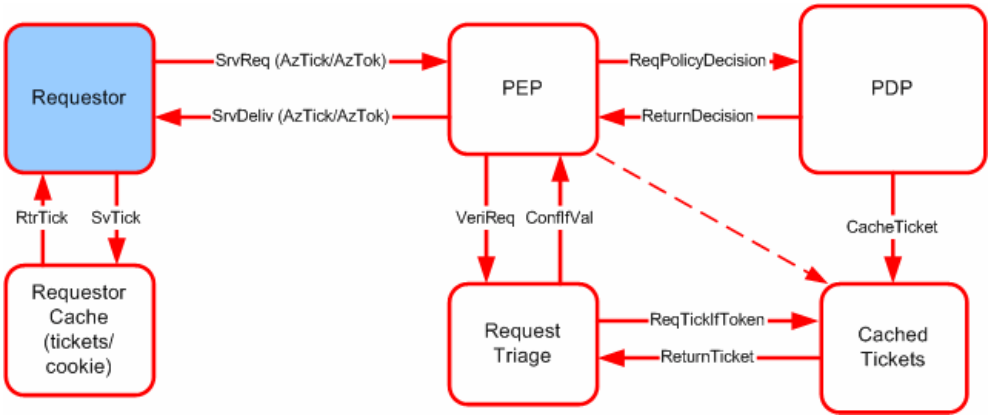

Figure 5. Triage operation in handling AuthZ tickets and tokens.

The following summarises the Triage operation on AuthZ tickets and token handling/evaluation:

- AuthzTicket is issued by PDP and MAY be issued by PEP
- AuthzTicket MUST be signed to ensure authenticity and integrity
- AuthzTicket MUST contain all necessary information to make a local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets MUST be cached; Resolution mechanism from token to ticket must be provided

GAAAPI supports AuthZ and AuthN tickets generation in a proprietary XML format and by using the SAML assertion format, which example implementation/design is discussed below.

3.1.4 Summary on GAAA Authorisation framework functionality

The UvA GAAA Toolkit and its GAAAPI functionality can provide a basis for building an Authorisation infrastructure for OLPP, however it will require some specific developments to enable multidomain distributed security infrastructure for user controlled resource provisioning. These GAP's are further discussed in section 4.

3.2 Existing Membership Management Services

This sections provides an overview of existing solutions and technologies for managing inter-organisational federations and/or associations for trust, policy and identities/attributes management. Their principal need for interdomain service provisioning was explained in the previous sections. The practical implementation may take a form of inter-organisational agreement, a coordinating or policy management authority, a managed registry, and a trusted service in general.

Experience and experimental implementations show that inter-organisational and inter-domain federations require some kind of inter-organisational agreements that is used to

establish trust relations. Trust relations can either be hierarchically organised or established in a meshed fashion. Trust relations may differ in the way they manage security associations. Federations can provide tightly or loosely coupled trust relations that can be subsequently used directly in inter-domain interaction or just used for initial trusted introduction. For example, DNSSEC [11, 12] can be considered as such a loosely coupled federation that can be used for initiating direct trust relations between a service advertising its public key via DNSSEC and having trust relations with the DNSSEC maintainer and a user that can trust provided in DNSSEC binding between the service identity (i.e., domain name) and its public key.

3.2.1 Internet2/US Federations and Supporting Middleware Tools

The Internet2 Middleware initiative and infrastructure is based on the following key projects [13]:

eduPerson/eduOrg [14]. The EDUCAUSE/Internet2 eduPerson task force has the mission of defining an LDAP object class that includes widely-used person attributes in higher education.

Shibboleth is developing architectures, policy structures, practical technologies, and an open source implementation to support inter-institutional sharing of web resources subject to access controls [15].

Grouper. An open source toolkit for managing groups [16]. It is designed to function as the core element of a common infrastructure for managing group information across integrated applications and repositories.

Signet [17]. A privilege management service is a component of campus middleware that provides centralized management of user privileges across a range of applications.

The **InQueue test federation**, operated by Internet2, is designed for organizations that are becoming familiar with the Shibboleth software package and the federated trust model [19]. Participating in InQueue permits an organization to learn about Shibboleth via the experience of multi-party federated access, whilst integrating its services into the organization's procedures and policies. It is also available as a temporary alternative to sites for which no suitable production-level federation exists.

The **InCommon federation** (http://www.incommonfederation.org [20]) supports user access to protected resources by allowing organizations to make access decisions based on the user's home institution exchanging agreed upon traits with the resource provider. InCommon eliminates the need for researchers, students, and educators to maintain multiple, password-protected accounts. Built using Shibboleth authentication and authorization technology, InCommon enables cost-effective, privacy-preserving collaboration among InCommon participants.

Although Internet2 middleware initiative provides a full set of tools to manage inter-university federations and also proposes a good business model to extend the number of adopters, the following factors should be taken into account when considering a Shibboleth based InCommon federation:

- Shibboleth requires the LDAP based EduPerson format for defining Identity and attributes. Although Shibboleth provides a well developed and well defined architecture, its implementation requires significant efforts as:

a) There are four primary components to the origin side in Shibboleth: the Attribute Authority (AA), the Handle Service (HS), the directory service, and the local sign-on system (SSO).

b) There are three primary components to the target side in Shibboleth: the Shibboleth Indexical Reference Establisher (SHIRE), the Shibboleth Attribute Requester (SHAR), and the resource manager (RM)

- Using Shibboleth for attributes management doesn't solve the whole access control problem as:

c) Current Shibboleth implementations have only examples for web-based access to electronic resources/information for humans. Both AuthN and AuthZ services in these examples are provided by sites or resources.

d) There is no good example for the whole access control bundle, in particular for the support of an AuthN service and a policy based AuthZ solution.

- Although currently SAAS (Shibboleth Attribute Authority Service) infrastructure is quite large, there is no special IdP/ServP directory or resolution service. Trusted providers are preconfigured manually and maintained by the files sites.xml and trust.xml
- Shibboleth's AA/IdP use its own namespace "`urn:mace`" which is preconfigured in both IdP Service Provider. If Shibboleth is to accept external calls from other systems and is required to send responses back, it is a task of the external system to understand and map Shibboleth attributes to its own namespace and presentation.

In summary, InCommon together with Shibboleth establishes an important landmark and provides a good framework for establishing compatibility with other associations and frameworks based on common attribute format and attribute management practice. The following ongoing development and works will ensure wider Shibboleth acceptance in the future:

- The currently recommended Shibboleth version 1.3 still uses SAML 1.1 however implements SAML 2.0 attribute namespace definitions and identifier formats.
- Currently ongoing project GridShib will provide a special Shib profile for Grids and potentially will allow a User Home Organization to manage VO membership information (see for me information about GridShib below). Additionally GridShib will add WS-based interface to SAAS.

3.2.2 European Federations

There is not yet a single European inter-university federation. However, there are ongoing coordination activities on Authentication and Authorisation services deployment among European NREN's. According to information provided by TERENA's TF-EMC2, currently in Europe only 2 NREN's support Shibboleth for application access (SWICHaai and Funet HAKA) and 7 NREN's are members of the EduRoam federation that provides access to a network using IEEEs 802.1X remote authentication protocol [21].

There is an intention to build common European Authentication, Authorisation Infrastructure (AAI) for European NREN's in the framework of the GEANT2 development [2, 3, 4]. This is an ongoing work where leading European NREN's participate, including SURFnet.

3.3 Virtual Organisations in Grid Applications

This section briefly presents the Virtual Organisation (VO) concept in Grid/OGSA and describes widely used VO management tool Virtual Organisations Membership Service (VOMS).

3.3.1 Virtualisation and Virtual Organisations in Grid

Grid resource and service virtualisation, together with provisioning, are two key concepts in the OGSA [22]. OGSA Security is built around the Virtual Organisation (VO) concept and targeted for the enforcement of the security policies within a VO as an association of users

and resources. VO provides a framework for inter-organisational and inter-domain trust management. When registered with a VO, an external user will be able to access to the enterprise/provider internal network based on his/her VO membership and relationship between the VO and the enterprise or provider. Access is typically enforced by a firewall, VPN gateway or Application Level gateway.

Typically, the VOs security services are created on the basis of the VO members' security services and may interact with them. A VO may run its own security services. Examples of such services are: credential validation services, trust services, authorisation services, and attributes services. But still many other services will remain in member domains and their authority need to be translated into VO domain through established trust relations and shared/translated semantics.

Although presenting a basic approach to understanding security services interaction in a virtualised Grid environment, the model above needs to be extended with basic operational models describing such use cases like project based collaboration, members' resource sharing or OLPP (or dynamic provisioning of complex multidomain distributed resources in general). At least, those VO operational models should describe existing and prospective use cases. Such attempt is undertaken in section 4.2. Conceptual VO model and basic operational models.

3.3.2 The Virtual Organization Membership Service (VOMS)

The Virtual Organization Membership Service (VOMS) has been developed in the framework of EU project EDG and DataTAG and currently being developed in the framework of the EGEE project [23, 24, 25, 26, 27]. VOMS goal is to solve the problems of granting users authorization to access the resources at VO level, providing support for group membership, roles and capabilities.

In the VOMS design, a VO is represented as a complex, hierarchical structure with groups and subgroups [26] what is required to clearly separate VO users according to their tasks and home institutions. From an administrative point of view, the management of each group can be independently delegated to different administrators. The administrators of each group can create subgroups and grant administration rights to these subgroups; they cannot modify memberships in any other subgroup. A group is basically a set of users, which may also contain other groups. In general a user can be a member of any number of groups contained in the VO hierarchy.

Every user in a VO is characterized by a set of attributes defining his/her group membership, role and capabilities in the scope of the VO, which can be expressed in a form of 3-tuples (group, role, capability). The combination of all 3-tuple values forms unique attribute, the so-called "Fully Qualified Attribute Name" (FQAN). In general an FQAN has the following form [26, 27]:

```
/VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]
```

For example, the FQAN corresponding to the role Administrator in the group Nerds of the VO campus.example.org is:

```
/campus.example.org/Nerds/Role=Administrator
```

The VOMS system consists of the following parts [24-26]:

- **User server:** receives requests from client and returns information about the user.
- **User client:** contacts the server presenting a user's certificate and obtains a list of groups, roles and capabilities of the user.
- **Administration client:** used by VO administrator to add users, create new groups, change roles.

- **Administration server:** accept the request from the admin client and updates the database.

In the authorisation sequence, the user obtains a VOMS Certificate via the User (VOMS) client. The VOMS Certificate is embedded into the Proxy Certificate (ProxyCert), which is send together with a request to the Resource to authorize user access.

The VOMS server returns a user X.509 Attribute Certificate (AC) that contains information about the user VO, and optionally information about the user group association and its role [27]. Future versions of the VOMS server claim to support the SAML Attribute assertion format. At the Resource, the authorization information provided by VOMS needs to be extracted from the user's proxy certificate and evaluated against the local access control policies in order to make the authorization decision.

The Administration Server communicates using the SOAP protocol, which can be easily integrated into WS-based Globus version 4 Toolkit. It consists of five sets of routines grouped into services:

1. **the Core,** which provides the basic functionality for the clients

2. **the Admin**, which provides the methods to administrate the VOMS database

3. **the History**, which provides the logging and auditing functionality (the database scheme provides full audit records for every changes)

4. **the Request**, which that provides an integrated request handling mechanism for new users and for other changes

5. **the Compatibility**, which provides a simple access to the user list for the mkgridmap utility. Two administrative interfaces (web and command line) are available.

The VOMS infrastructure suggests that a VO may have a few VOMS servers with synchronised membership databases, however one VOMS server can serve multiple VO's.

A central membership database is maintained by a VO and must contain information/attributes for all registered VO members. Currently, only user attributes are stored in VOMS database. There is ongoing discussion about providing VO credentials to the resources as well.

User Server and Clients (Core VOMS System) is developed by INFN, Administration Server and Client (Admin Interface) is developed at CERN. VOMS is available as open-source software under an EGEE/EDG license.

3.3.3 GridShib profile for VO attributes management

GridShib is an NMI (NSF Middleware Initiative) project that intends to integrate GT/Grid security infrastructure and Shibboleth to form a robust attribute infrastructure for campus environments to enable secure verification of user attributes for inter-institutional Grid users [28]. This project will deliver over 2005-2006 a framework that allows participants in multi-organizational collaborations to control the attribute information that they want to publish, share, and reveal to other parties. Those parties will be also able to determine whether they possess the capabilities to access a service by matching their capabilities with the service's shared policy of required attributes. Pseudonymous interactions will be supported through the use of anonymous public key credentials that are mapped to the client's identity at the client's own discretion.

The project substantially leverages on and extends existing technologies of primarily Internet2's Shibboleth, the Globus Alliance's Globus Toolkit[4], and the MyProxy[5] based GridLogon Service. The framework will use Shibboleth's Attribute Authority service (SAAS)[6] and its attribute release policies to restrict the attributes communicated to other parties. GridShib will enable Web Services access to Shibboleth services by using GT4 application integration tools. To enable pseudonymous deployment, a module will be developed for the GridLogon service to allow authenticated users to obtain public key credentials that do not reveal their identity, yet are fully compatible with the Grid Security Infrastructure. Formats and protocols will be developed and implemented to express, publish, share, and match attribute-related policies and capabilities.

In a summary, GridShib will produce a Shibboleth implementation for non-web-based applications, so-called GridShib profile. GT and Shibboleth integration will be based on Shibboleth attributes management/access model and will focus on the following attributes handling/providing/requesting models:

1. Basic Globus-Shibboleth integration without anonymity using attributes request/pull by the resource from the trusted SAAS

2. Basic Globus-Shibboleth integration without anonymity using attributes provided by the requestor which are previously obtained from the trusted SAAS

3. Globus-Shibboleth integration with anonymity and attributes requested by the resource from the trusted SAAS that is can release attributes based on user pseudonym or authentication confirmation credentials.

4. Globus-Shibboleth integration with anonymity using attributes provided by the requestor which are previously obtained from the trusted SAAS for the user pseudonym or anonymous authentication confirmation credentials (Authentication/identity token).

Interaction between the Shibboleth enabled client and the resource in the GridShib profile will consists of four major steps:

1. The Grid Client POSTs a SOAP request to the Grid Service together with user credential in the form of user ProxyCert.

2. The Grid Service, if user authentication is passed, POSTs a SAML SOAP message to the Attribute Authority (AA) at the Identity Provider (IdP). Information about AA may be included by the requestor into its proxy credential, or the service may use preconfigured list of trusted AA's.

3. The AA returns an attribute assertion to the Grid Service based on provided user identity (both real and pseudonymous providing identity mapping if necessary).

4. The Grid Service performs request evaluation based on received attributes and access control policy and proceeds with the requested operation and returns a response to the Grid Client.

3.3.4 VO Management in LCG and EGEE

The current VO management practice in the LCG and EGEE projects, provide a good example of the instant implementation of the VO concept. The approach is however project

[4] Globus Toolkit. - http://www.globus.org/toolkit/
[5] MyProxy Online Credential Repository - http://grid.ncsa.uiuc.edu/myproxy/
[6] Shibboleth Project. - http://shibboleth.internet2.edu/

based and project oriented. This means that they have a well-defined VO registration procedure, a basic Security Policy, and a simple Acceptable Use Policy. The Major VO membership management tool is the VOMS, which supports user registration procedures with the VOMS Admin server automated workflow.

The following documents define VO management framework in LCG/EGEE:

Virtual Organisation Registration Procedure [29]. This document lists the necessary steps a Virtual Organisation (VO) should take in order to get registered, configured and integrated in the LCG2/EGEE infrastructure. Before following this procedure, the VO managers should follow the instructions of the Virtual Organisation Security Policy document and prepare their VO Acceptable Use Policy (AUP) (see below).

Note. The complete life-cycle of a VO, including its wrap-up procedure is not discussed in this document. The operational responsibilities during the life of the VO, e.g. regular membership expiration and re-registration, non-replication of Personal user data across sites etc. are defined in the User Registration and Virtual Organisation Membership Management Requirements document [30].

Several decisions and steps need to be taken in the process of a VO creation and registration:

1. **Naming the VO.** Recommended VO naming style suggests that VO name should resemble project and/or team. It also includes appropriate DNS host aliases (or even dedicated domain name) and host certificates, when necessary, in order to prepare a properly managed system environment for VO-related data, scripts, web pages and transactions.

2. **Request VO integration into existing EGEE infrastructure** from one of designated bodies EGEE Generic Applications Advisory Panel (EGAAP) or NA4/SA1 Joint Group. During the request processing NA4/SA1 JG will estimate required resources (computing power and load, storage size, etc.) and propose possible VO applications hosting and resources allocation between candidate hosting sites and Grid Regional Centers (RC) and also fix requirement to RC to participate in the VO. As a result of this stage (but not limited to) a VO manager is appointed and a CIC (Core Infrastructure Centre) appointed to provide VO user management service to the new VO.

3. **Setting-up a VO.** The VO management selects a site where to run the VO database (VODB) server and the Registration service/database (where the acceptance of the Grid Usage Rules by the user is registered). There can be few options for particular implementations of the VO services.

4. **Populating a VO.** Candidate entries in the VODB are passed through successful Registration process and Registration database additions. Suggested mechanisms to bootstrap and update a VODB depends on the selected technology and may be use LDAP based solution or integrated Registration and VODB solution based on VOMS

5. **Integrating VO into existing infrastructure.** As soon as a VO is configured, the VODB contents must be propagated to the Grid sites in order to be matched to the users' credentials at job submission time. This is done currently with the grid-map file or LCMAPS that reside on resource side and are supported by RC Mapping System. In addition to the VO Registration server and VODB, two other Grid infrastructure components must be VO-aware: a Resource Broker Service, that is at least a Resource Broker (RB) and its associated BDII, and a Replica Manager Service, that is a Replica Manager (RM) and a Replica Catalog.

6. **Organising support structure for the VO.** This requires designated group of people to manage VO procedures both registration and user support, including VO-wide Security Incident response. A VO Support Manager is responsible for building this structure and becomes a member of the EGEE Support Task Force.

There are many different valid options for some of these steps. They depend on many parameters like the technology (LDAP[7] or VOMS[8]) and the location where the VO database (VODB) resides.

LCG/EGEE Virtual Organisation Security Policy [31]. This policy defines a set of responsibilities placed on the members of the VO and the VO as a whole through its managers. It aims to ensure that all Grid participants have sufficient information to properly fulfil their roles with respect to interactions with a Virtual Organisation (VO).

3.3.5 Summary on VO functionality for multidomain resource provisioning

Current VO concepts and existing practices lack a common theoretical foundation. As a result, it causes different understandings of the VO concepts and functionalities by different groups of potential adopters and users. The following can be considered as a reason of this confusion and misunderstanding:

1) **Support for details:** OGSA's vision of the VO and virtualisation is not supported by more detailed description of the VO functionality and operation;
   a) first of these confusions is relations between virtualisation and VO which presumably could be resolved with the definition of the VO management functionality including VO foundation/agreement and life cycle;
   b) second issue to be clarified is relation between VO and dynamic associations: which part of the VO concept is static (like CA/PMA and AttrAuth) and which can support dynamic associations (and dynamic trust management).
2) **Definitions:** Current VO implementation in LCG/EGEE needs more conceptual/higher-level definition to be aligned with (yet to be developed) OGSA VO concept.
   a) There is still no clear definition of the VO Agreement and VO policy in LCG/EGEE. Current use of the VO is directly association with two projects and therefore VO is managed under the project administration. (Using in this case generic word VO adds to the confusion around VO concept itself.)

The following issues should be taken into account when considering VO use for dynamic resource provisioning:

1) VO setup is a complex long-time procedure; therefore a VO cannot be used at the first row for the global ad-hoc dynamic trust establishment.
2) VO management and VOMS infrastructure is rather designed for long-term collaborative projects. However, VOMS provides all necessary functionality for creating ad-hoc dynamic VO association. The issue remains how to consistently manage trust and authority in such a dynamic VO. One of possible solution is to combine/add attribute management functionality being developed in the framework of Internet2 Grouper and Signet projects. This is in addition to suggested use of the GridShib profile for SAAS integration into the VO management.
3) VOMS server Attribute Certificate is based on X.509 AC for Authorisation and currently well defined. However, its use for Gird authorisation (with GT) suggests using Proxy Certificates.
4) The VOMS client-server protocol is not clearly defined. Formalisation of the VOMS client-server protocol will facilitate wider VOMS adoption and better understanding

[7] Instructions for setting up a LDAP-based VO: http://cern.ch/grid-deployment/cgi-bin/index.cgi?var=gis/vo-setup
[8] Instructions for full deployment of a VOMS-based VO: http://cern.ch/grid-deployment/cgi-bin/index.cgi?var=gis/voms-deploy

5) The current VOMS implementation does not have a flexible attribute namespace management (and corresponding procedure and policy)

6) VOMS requires a user ID and therefore doesn't provide (user) controlled privacy protection (in contrary to Shibboleth).

    a) It is expected that the currently developed GridShib profile will provide a framework for combing well developed Shibboleth attribute management solutions and VOMS functionality currently a standard-de-facto for VO management in Grid

7) There is obvious benefit in interoperability between VOMS and SAAS and presumably will be achieved with the GridShib profile which targets for providing SAAS integration into Grid/GT environment/infrastructure. Although VOMS and SAAS both serve as Attribute Authorities there are minor differences in their operation on the user/client and service/resource sides:

    a) In VOMS the user first needs to obtain VOMS AC by requesting particular VOMS server, and next include it into newly generated Proxy Cert and send request to the service

    b) In SAAS the user sends request to the Shib-aware service and may include a particular IdP reference, otherwise service will poll trusted AA/IdP's based on preconfigured list of trusted providers.

8) Existing LCG/EGEE VO registration procedures allow the use of DNSSEC for populating a VO together with its (secondary) public key that can be used for initial trusted introduction of the VO and secure session request by the requestor.

Note. DNSSEC has limited space for putting the key information because of DNS/DNSSEC response message allows only one non-fragmented package of size 1220 bytes for standard DNS message and 4000 bytes for special DNSSEC extension [11].

Note. In DNSSEC, it is suggested that domain's (in our case VO's) record and key is signed by upper layer domain's key, and therefore DNSSEC trust tree must be compatible with the application oriented trust domain.

4 Filling the Gap - Required new GAAA Components for Complex Resources Provisioning

4.1 Required new GAAA Functionality and Components

4.1.1 Extending GAAA Authorisation Framework

This section discuses further development of the GAAA Authorisation Framework with new conceptual and architectural components that will target resource provisioning area and OLPP in particular. This discussion will be based on the requirements that came out of the OLPP use case analysis in section 2.1 and overview of the current GAAA Authorisation Framework and GAAA Toolkit development provided in section 3.1.

First suggested step can be on defining two GAAA Authorisation Framework profiles:

1) GAAA Authorisation Framework for Provisioning profile (GAAA-P) that targets specifics in providing AuthZ service in provisioning complex resources that requires preliminary reservation and combination of multi-component resources that may span over multiple administrative and trust domains, and therefore requires multiple independent access control decisions to be combined in a workflow controlled by the provisioning policy.
2) GAAA Authorisation Framework for Role Based Access Control profile (GAAA-RBAC) which major task is to provide instant access control decision and doesn't require (policy-based) decision flow control. This profile may also require evaluation of complex user credentials originated from multiple administrative and trust domains but access control policy doesn't require evaluation flow control.

GAAA-RBAC has its major use in controlling access to the multifunctional and/or hierarchical resources that provide/reveal their service depending on presented attributes, roles, and credentials, like in case of accessing data storage, allocating computer resources or controlling complex equipment.

Example of GAAA-P is the OLPP which is the main topic of this gap analysis. The GAAA-P operational model may also be related to the general complex resource provisioning like in the classic GAAA scenario of ordering party components like movie, pizza and music. Theoretically, GAAA-P should also describe such examples like travel itinerary or vacation package reservation but in these more business oriented cases the separation of business logic and access control decision should be done clearly.

4.1.1.1 Adding workflow control to the GAAA based provisioning model

Current GAAA-P implementation for Bandwidth-on-Demand provisioning uses driving policy for combined bandwidth request authorisation and network equipment control. This functionality became possible because of flexibility of the generic AAA policy format that allows building complex IF-THEN-ELSE statements and compiling then into executive object. However, such approach has manageability problems, and one of such problems can be in combining external policy components and/or making calls to external decision making points depending on master driving policy flow.

One of suggested solution for this issue is to separate policy evaluation and flow control and making flow control interpretable at the runtime:

1) policy is a static set of rules that in general can be defined by existing framework agreement between user and provider;

2) workflow is an instant dynamic process (although may be create based on existing business process template) that orchestrates interaction of multiple services and processes to deliver final service to the requestor.

Standard business workflow also contains decision points that are driven by events and process status context but in basic cases doesn't require initial attributes re-evaluation at the decision points. This means that all interacting in the workflow resources must share explicitly or implicitly a common access control policy. Adding policy based control to the workflow will allow to combine decision points with independently manageable policies. However, this adds additional requirements to track full security context of the request.

In two basic provisioning scenarios described in chapter 2, workflow management has the following differences between them:

1) when reservation (and additionally provisioning) is controlled by one of domain ICC (interdomain Connection Controller), in particular case from user domain, all workflow is managed by a single ICC that the most probable will do the component policies evaluation centrally.
2) when reservation (and additionally provisioning) is chained, the workflow object may need to be transferred between participating domains and policy application can be done locally in each domain, without populating policy between all participating domains.

In summary, above discussion provides a motivation and use case for separating workflow management from the policy evaluation and combining them in the workflow decision points. This approach actually uses workflow as the upper layer abstraction of the overall provisioning process/model. Workflow description standardisation is currently ongoing in the framework of the OASIS Web Services Business Process Execution Language (WSBPEL) TC with the initial input from earlier IBM specifications WSFL and BPEL4WS.

Note, discussed above workflow and policy combination model suggests that user/request credentials are not changed between workflow decision points and service state is not exposed to the overall/master workflow process but can be implicitly applied in the local policy evaluation. In other words, from the requestor's and provider's point of view policy remains static in this provisioning model, however it can account service or resource state by referring to their state via environment/status information request.

4.1.1.2 Dynamic trust management

Security operations such as AuthN and AuthZ must always rely on established trust relations between communicating parties. This trust can be established directly or indirectly via trusted intermediaries. Also in exchanging information between AuthN/AuthZ system components, e.g. IdP/AuthN, PDP, PEP, PAP and PIP, etc., a secure context must be present explicitly in a form of evidence as part of a request/response, or implicitly if a related processes runs on a single system and using one runtime environment (under protected privileged account).

The security context should be present explicitly or implicitly in any session on the protected resource. Such security context is established during session start based on the positive AuthZ decision, which in its own turn must rely on positive AuthN result. Therefore from the session viewpoint there is no difference how this security context is established. Trust relations on which this security context is based can be established at the service configuration stage by configuring trusted sites and key/certificates, or established dynamically in a special trust negotiation and key exchange session between interacting services.

During dynamic trust negotiation to establish a security context, negotiating parties must present initial credentials that must have a verifiable trust chain to the mutually trusted authority. Keys and credentials and all chaining certificates may be cached for future re-use, but every time at invocation time the whole path needs to be checked against validity period and possible presence within revocation lists. In some cases, the use of online certificate status (using OCSP protocol) checking can be considered.

The framework for (dynamic or session based) trust and credentials negotiation for Web Services is defined in two complimentary specifications WS-Trust (WST) [32] and WS-SecureConversation (WSSC) [33]. Additionally, WS-Federation (WSF) specification [34] proposes a framework for flexible Identity Management and leverages both WS-Trust and WS-SecureConversation specifications. WSF can add more flexible requestor identity management including pseudonymous services, identity and attributes mapping, single sign-on.

WST defines SOAP based mechanisms for brokering trust relationships, requesting and returning security tokens. Requests for security tokens are made by sending a Request Security Token (RST) to the Security Token Service (STS). WST specification defines three possible actions that can be performed: issue a new token, renew a token, or validate a token. It is essential that all these requests must provide initial secure credential or token as a base for issuing a new token.

Another solution for dynamic key and trust management can be provided by the W3C XML Key Management Specification (XKMS) that comprises of two parts: the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS) [35].

However, it is important to stress that all these specifications don't deal with the initial trust establishing. Trust relations must be established in one or another way and presented in all WS-* interactions in a form of trust anchor or business anchor (which is in its own turn should be cryptographically proven).

WS-based specifications use SOAP header for communicating security context, i.e. initial security token or credential, what is considered to be a solution transparent for applications as SOAP header is processed automatically in most WS/SOAP applications.

So, even when considering to use well-defined solutions for session/instant security context establishing with WST or XKMS we still need to solve the problem of initial trust relations or establish an initial trust anchor. In currently used solutions and implementation for inter-domain access control the problem is split in two parts – federated trust for the attribute services/management (which is rather static) and confirmed/verifiable trust for the identity (which is dynamically established or invoked). This means that based on explicitly existing and presented trusted attribute credentials the identity credential confirmation/verification can be requested in a separated request to the identity origination site. This model is actually based on the separation of Authentication and Authorisation.

Existing solutions for federated trust management are represented by user and organisation federations, VO's, identity services and also can be based on banking or credit card clearing services.

It is quite important for user controlled service provisioning that based on existing federated trust relations (static) the dynamic user and resource association can be established for the particular service provisioning request. Practically, this operation should be supported by the dynamic trust management system that can use the VO technology or Job-centric security model [9].

4.1.1.3 Policy combination and aggregation

Multidomain and/or multilevel access control may require multiple policy decisions to be combined, which may use the same or different semantics and languages depending on locally used access control systems. This situation occurs when reservations and resource provisioning traverses domains or when the a set of requested resources expres policies in different formats, which uses domain specific semantics and metadata. Another example is when access control system have different policies for different tiers and protection levels, (e.g. a site firewall, service/application authorisation, system access control list or ban-list, etc.) In addition, some of these policies can be of legacy type and should be accepted as is or trustworthily converted to GAAA runtime policies. In such cases central authorisation service should have a possibility either to aggregate all component policies or combine them

at runtime by chaining related PEP or calling to related PDP for partial policy evaluation. In both cases the major issues is to ensure request evaluation integrity if it needs to be evaluated by multiple PDP's and/or against multiple policies (refer to section 3.1.2 for the abstract GAAA Authorisation model).

Another reason for having policy mapping functionality is the need to audit/evaluate suggested combined policy against component/contributing policies to reveal possible conflicts at different levels and in different domains. This may happen in at the time of a VO or other association creation when resource and member policies need to be evaluated and in some cases audited to find contradictions and bypasses. In more dynamic scenario mutual evaluation of the security policies need to be done in the process of establishing relation between participating services and/or parties.

Generally, GAAA can create an infrastructure of communicating GAAA nodes that can be configured for combine domain specific policies defined by domain specific policy languages. In multilevel protection, multiple PDP's and PEP's can be nested or chained to combine different policies during the evaluation of a service request. GAAA implementations should be aware and able to process different request formats and policy expressions. Application Specific Models, part of the GAAA architecture [36], can be constructed specifically to perform translations between different request formats, whilst performing semantic mapping [36]. Infrastructures using the same policy language and a uniform, well-defined set of semantics may not need such functions. However, the more complex cases, the higher the likelihood to encounter non-uniform cases.

4.1.1.4 Attributes and metadata resolution and mapping

Correct policy evaluation and combination in multidomain scenario requires either use of common attributes and metadata format and namespace or mapping between used formats and namespaces.

Actual attribute and metadata mapping can be provided by authoritative/trusted IdP and AA services but still GAAA functionality should have a possibility to understand or resolve known namespaces to make consequently necessary requests to authoritative services. This can be provided by general context management function of the policy evaluation engine or separately by so-called Policy Information Point (PIP). Locally PDP/PEP should have a possibility to securely cache known namespaces and enumerated attributes and metadata but globally they must be supported by respective registries and resolvers.

Currently maintained registries include IETF/IANA namespace registry, Internet2/MACE OID and URN Registries. In particular, the Namespace Identifier (NID) of the namespace "urn:mace" is specified by RFC3613. There is no known namespace and enumerated types registries in Europe[9].

Individual IdP and VOMS can act as ad-hoc attribute and metadata registries but without common naming schema and common registries they can serve only as stand-alone registries and their mapping with other IdP and VOMS will require establishing mutual relations.

4.1.2 Extending GAAA Toolkit

This section suggests a list of developments and extensions to the GAAA Toolkits to support authorisation infrastructure for multidomain user controlled service provisioning.

[9] Therefore, this is of interest for both GAAA-P and GAAA-RBAC profiles development to facilitate and contribute to such activity currently ongoing in the framework of TERENA TF-EMC2 and EGEE Policy Coordination activities.

1) Integrating GAAA Toolkit and GAAAPI

As it was described in the section 3.1 and discussed in the section 4.1.1 above current GAAA Toolkit has two major implementations for providing BoD AuthZ service and for RBAC in collaborative applications. Both of these applications are built around Rule Based Engine (RBE) using a AAA driving policy language that have may call different sets of ASM's and API's.

Integration suggests, first of all, adding the following functionality to the RBE ASM's:

a) AuthZ and AuthN tickets support both proprietary and SAML

b) XACML messaging

c) XML Signature and Encryption

The following development should be related to the integrated GAAA Toolkit.

2) Adding SAML 2.0 assertion and protocol support, including SAML XACML profile that will simplify AuthZ tickets management

3) Adding XACML to describe policies and as a policy meta-format and exchange format

4) Developing simple policy management tools supporting multiple policy formats, first of all, AAA-format and XACML

5) Adding support for different types of secure credentials, in particular, X.509 PKI Certificate and Attribute Certificate, SAML assertions (currently available in GAAAPI), and related callouts to issuing authorities, in particular VOMS and Shibboleth.

6) Adding WS-Trust secure tokens support and Secure Token Service (STS) functionality, first of all, for credentials mapping that later can be organised as a separate service/component

7) Integration with the GT4 and the EGEE gLite Authorisation Framework

   This can be done in three ways:

   a) Using GT4 WS/messaging firmware to provide WS-based access to GAAA_tk authorisation service; this will allows easy GAAA_tk integration into different applications.

   b) Adding GAAA AuthZ callouts to GT4/gLite AuthZ framework; this will allow using GAAA RBE as one of regular services for GT4 and gLite

   c) Integrating GAAA AuthZ/RBE into GT4 AuthZ framework as one of PDP's.

Suggested GAAA-P and GAAA-RBAC structure is shown on the picture below. It contains the following main functional sub-systems:

- GAAAPI that provides all necessary functionality for the communication between PEP and PDP and providing security context for service request evaluation against service (access) policy and includes
  - i) namespace resolver to define and resolve what policy and what attributes should be used for the request evaluation
  - ii) a triage and cache used to provide initial evaluation of the request including validity of provided credentials

iii) another targeted triage functionality is to provide AuthZ tickets/tokens handling functionality that in the first row includes service request evaluation against provided AuthZ ticket/token claims (what can be also forward policy supplied together with the request);

iv) attribute resolver and Policy Information Point (PIP) provide resolution and call-outs to related authoritative Policy Authority Points (PAP) and Attribute Authority Service (AAS) which can be a part of general Identity Provider service (IdP);

- GAAA-RBAC subsystem that provides GAAA-RBAC profile functionality and basically includes PEP, PDP and GAAAPI with related Application Specific Modules (ASM);
- GAAA-P subsystem includes GAAA-RBAC subsystem used for general policy evaluation and adds flow control with the Flow Control Engine (FCE) and Flow Repository modules;
- Rule Based Engine (RBE) is represented by combination of PDP used for individual policies evaluation and FCE that control multiple policies evaluation or other sequence of policy evaluation for the complex resource;
- A set of ASM's that provide interfaces to application specific functions of the requestor (requesting service) and the resource/service.

Technically, two defined GAAA profiles use the same set of functional components but have different configuration of modules/components related to security context (including key, trust relations, external call-outs configuration), internal components interaction and also required ASM functionality.

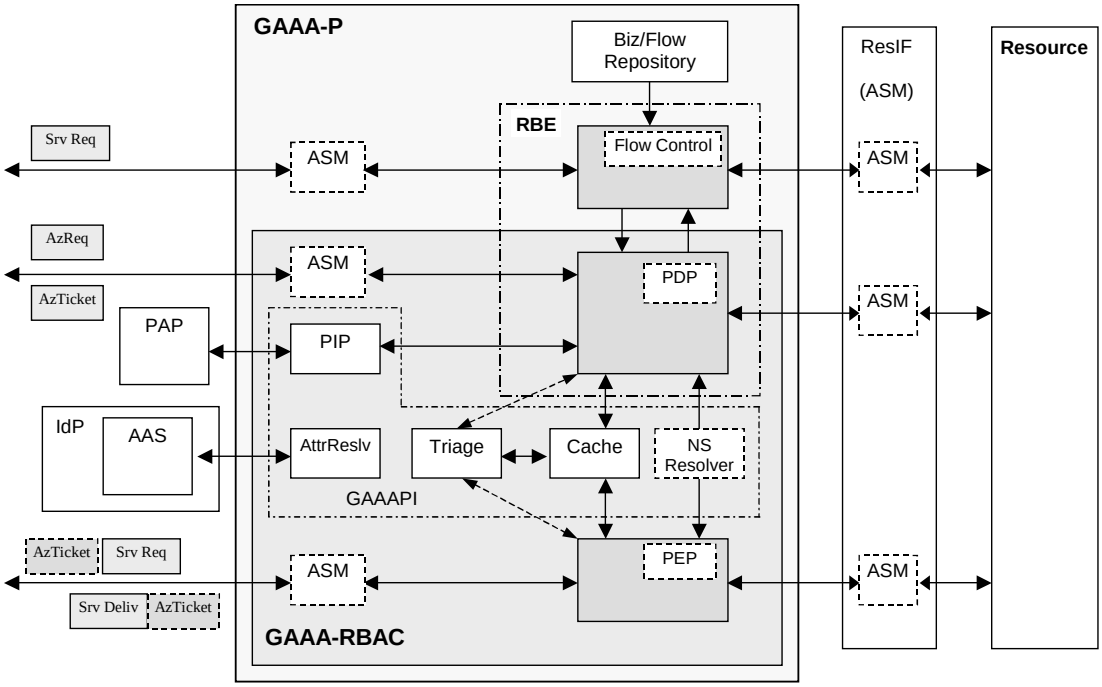

Figure 6. GAAA-P and GAAA-RBAC structure and main functional components

Separation of the flow management/processing and individual resources' policy evaluation will allow to separate business related part of the provisioning process that is normally related to the general/complex user request and policies applied to some component services/resources. Service/resource policies are more static and managed by owners/providers. Provider of the complex services/resources can apply it's own provisioning (business) model that can be described in the form of (work)flow and can

contain different options for that provisioning and consequently different sequence of individual policies evaluation and also some other conditions related to overall provisioning process.

Workflow and (resource) policy separation doesn't affect individual policies evaluation that can also have some sequence of evaluation of the request against the related/referenced policy. In this relation there can be defined three levels/steps of the service request evaluation against the provisioning or individual policy:

- one step (or instant) request evaluation by Triage that simply checks (instant) request matching against provided AuthZ ticket/token or instant push-policy;
- resource/service policy evaluation by the PDP that does request evaluation according to the policy that itself describes a sequence of provided attributes/information evaluation, e.g. in XACML evaluation sequence includes first target (subject, resource, action) matching, next rules evaluation and finally rules combination to make overall policy based decision;
- complex request evaluation that requires multiple policies evaluation in the sequence described by provider or request specific (business) flow; in this case the FCE take care about driving the evaluation and provisioning process.

Outsourcing combination of individual policies evaluation to upper layer element/functionality of FCE will simplify multiple policies management in sense that there will not be a need for the overall policy validation to avoid possible conflicts and attributes conversion.

4.2 Using VO model for dynamic security associations in complex resource provisioning

This section attempts to review current VO concept [22, 37, 38] and provide a multilevel approach and model to understand how the VO, as an abstract concept and as a practical implementation can be used for federated and/or dynamic trust management. In other words, we will discuss relations between VO and dynamic associations: which part of the VO organisation and operation is static (like CA/PMA and AttrAuth) and which can support dynamic associations (and dynamic trust management).

First of all we need to clarify one of widely used misunderstanding between VO as virtual entity and dynamic processes and associations. To do it consistently we need to look at different types of security associations and their dynamics (or lifetime characteristics). In relation to this we can build the following list:

1) **Session** – establishes a security context in the form of session key, which can be a security token or a simple UID bound to secure credential or session ticket. Session may associate/federate users, resources and actions/processes.
2) **Job/workflow** – this may be more long-lived association and include a few sessions. Job or workflow is built around specific task that is defined either as contract to perform some work or deliver product, or business process unit that also deliver some service and provides orchestration of many other processes. They may need to associate a more distributed collection of users and resources for longer time required to deliver a final product or service. Job and workflow may contain decision points that switch alternative flows/processes. The security context may change during workflow execution or Job lifetime. Job description, as it is used in the Job-centric security model [9], may contain both user and resource lists. It may also provide trust anchor(s) (TA) and security policies. Job TA is derived from the requestor and the service trust relations established on the base of the contract to perform some job. Workflow TA can be implicitly derived from the parent process.
3) **Project or mission oriented cooperation** – this type of association is established for long time cooperation (involving people and resources) to do some research, development or production but it still has some well-defined goals and area of activity

and often criteria of mission fulfilment. This is actually the area of currently existing VO associations.

4) **Inter-organisational association or federation** – this type of association is built on long-term (often indefinite) cooperation agreements and may have a wide scope of cooperative areas. This is the area of inter-university associations which examples are InCommon or InQueue, and Shibboleth is specially designed to support this kind of federations.

Comparing two last types of associations, we can suggest that for the VO type of federation the common membership service is typical and essential. However, its implementation can be either centralised like in VOMS or distributed like it is intended in the GridShib profile.

Proposed above classification allows us to assume that all identified types of associations will have its place and use in the future responding to different goals and tasks. Another suggestion that can be done from the above discussion in the context of user controlled service provisioning (UCSP) is that Job-centric/VO-based associations may scale to each other and consequently use each other's technical infrastructure and tools by adopting the dynamics to their specific tasks.

Now we will try to identify possible VO operational models depending on more detailed analysis of the major service provisioning use cases. Introducing VO concept/functionality into dynamic service provisioning will bring flexibility to the problem of dynamic trust management

When considering the use of VO for trust and attributes management, we should refer to the conclusion made in the VO overview section (section 3.3) that VO creation is quite complicated and bureaucratic/formal procedure. VO creation is normally initiated by one of organisational or business/project entity and has a specific goal and mission. VO can be created for the project based collaboration, members' resource sharing or dynamic provisioning of complex multidomain distributed resources in general. VO concept can be also used for general purpose user association.

VO attribute or membership service is used for trusted attributes brokering between (member) organisations when requesting resources or services from the VO members or their associates. However, VO operation will differ depending on what are the VO associated members and how the VO membership service is used in VO related activities or services.

In this context three basic and one additional VO operational models can be defined:

1) User-centric VO (VO-U) that manages user federation and provide attribute assertions on user (client) request.
2) Resource/Provider centric VO (VO-R) that supports provider federation and allows SSO/access control decision sharing between resource providers.
3) Agent centric VO (VO-A) that provides a context for inter-domain agents operation, which process a request on behalf of the user and provide required trust context to interaction with the resource or service.
4) Project centric VO (VO-G) that combines User centric and Provider centric features what actually corresponds to current VO use in Grid projects.

Although in different applications and use cases VO operations will differ in sense of providing primary association of users, resource providers or services providers the VO management infrastructure will need to have almost the same set of services. The above classification should help to understand how major security services will operate in each of the different types of VO.

User-centric VO-U manages user federation and provides attribute assertions on user (client) request. For this purpose, VO-U maintains VOMS or user Attribute Authority that receives requests from user clients and provides VO member attribute certificates or other type of attribute assertion. VOMS/AA can also validate user credentials on request from services. However, this is the user who presents attribute credentials to the service in order

to obtain access control permission. In this sense, VO-U actually implements pull model for the access control decision. VO Attribute service is the central service for this type of VO. This can be considered as current operational model for the VOMS in Grid application. GridShib profile will allow decentralisation of attributes management.

Resource/Provider centric VO-R supports provider federation and allows SSO and access control decision sharing between VO members, i.e. resource providers. In this respect, VO-R may run own VO-wide AuthN and AuthZ services and correspondently VO-wide access control policy. It is logically that all services in the VO-R association can accept the VO AuthZ service decision once issued for the user on their request. If the user wants to access multiple services in the VO-R s/he can use obtained access granting ticket as a SSO credential, however services may need to validate presented credentials/ticket with the VO AuthZ and AuthZ services.

Agent centric VO-A provides a context for inter-domain agent operation. In this model/profile agent acts as a representative and a broker of the trust and other services for the specific domain. Agents are considered more independent in the VO-A than users or providers in other models VO-U and VO-R. Agents may have central attribute or certificate service but in more specific for the VO-A model case they will maintain mutual trust relations (which initial establishment for a time being is out of scope for this study).

Project centric VO-G (as originated from Grid projects) can be introduced to reflect typical use case when a VO is established to support user cooperation in the framework of the long-running project and to overcome existing/legacy organisational boundaries. VO-G associates both users and resources and actually combines two identified earlier models VO-U and VO-R. It maintains central VO membership/attribute service and may run also VO-wide security services such as AuthN/IdP/SSO and AuthZ.

There may not be clear difference in real life VO implementations to which operational model they adhere but proposed abstraction will help to more flexibly design supporting security services. For example, it can be suggested that current VOMS based VO in Grid will evolve from currently used VO-U model to more appropriate VO-G model.

One of open issues that should be resolved by practice in ongoing implementations is to which operational model we should ascribe a resource/service attributes assignment/management if we need to provide mutual user/requestor and resource/service AuthN or AuthZ.

The major motivation behind defining basic VO operation models is to define possible profiles for the VO security services as well as suggested gateway services to interact with different/external security models.

Benefit of using VO based trust and attribute managing/brokering is that VO can be created and used as a dynamic association for wide range of duration given the VO as a concept that can potentially combine virtualisation and dynamic.

Proposed above classification and definitions can also help in achieving better understanding between Grid originated customers and traditional infrastructure providers (in particular, network/OLP providers) in situation when attempting to match their traditional operational security models. For example, Grid customer comes to network/LP provider on behalf of the VO and wants to order LP connectivity on-demand. The question for the customer is how it can present its VOMS credential normally used inside VO to the external service; the question for the provider is how it must handle VOMS credentials to consistently adhere to its corporate security model and policy.

5 Summary

This technical report provides extended analysis of the currently available authorisation and policy enforcement technologies. The analysis is based on the RFC2904 Generic AAA Authorisation framework (GAAA-AuthZ) approach and access control model applied to on-demand network resource provisioning.

Based on detailed analysis of the typical usecase of on-demand Optical Lightpath Provisioning, the requirements to major components of the supporting access control services and infrastructure are specified: authentication, identity management, authorisation, attribute and federation management, trust management, and authentication/authorisation services API.

The requirements were used as a framework for detailed analysis of currently available solutions and tool to support required functionality and services. Special attention was given to GAAA-AuthZ operational models for complex resource provisioning that allow dynamic AuthZ service configuration and consistent service/resource request evaluation in multi-domain environment.

The report also provided extended overview of currently available solutions for managing user membership services and federations which are considered as an important component in multidomain service provisioning used for inter-organisational attribute and trust management. Particular attention was given to Shibboleth based inter-university federations and Virtual Organisations membership service currently used in Grids.

Suggestions are given what available technologies and solutions can be used for on-demand network resource provisioning and what functionality is still missing and requires development.

Proposed solutions are based on further development of the GAAA Authorisation framework and the corresponding UvA GAAA toolkit to address complex network resources provisioning. This special set of functionalities is defined as a GAAA profile for provisioning (GAAA-P). The GAAA-P development is part of the GigaPort NG Research on Networks project.

Additionally suggestions are provided how the VO concept can be used for dynamic security associations management in complex/inter-organisational service provisioning and trust management.

6 References

[1] Interdomain OLPP in Gigaport 6 RoN, by Bram Peeters

[2] GEANT2 Deliverable DJ5.2.1: Documentation on GÉANT2 AAI Requirements. - http://www.geant2.net/upload/pdf/GN2-05-026v6.pdf

[3] AA aspects in some GN2 activities, by Maurizio Molina. - http://www.terena.nl/tech/task-forces/tf-emc2/meetings/feb05/ppt/gn2-aai-for-JRA1-SA3-JRA3.ppt

[4] The Authentication/Authorisation Initiative in GN2. First Steps towards an Integrated Infrastructure, by Diego Lopez, Jürgen Rauschenbach, Klaas Wierenga. - http://www.terena.nl/conferences/tnc2005/programme/presentations/show.php?pres_id=78

[5] RFC 2904 - "AAA Authorization Framework" J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, August 2000 - ftp://ftp.isi.edu/in-notes/rfc2904.txt

[6] B.Oudenaarde, et al. “Grid Network Services: Lessons and proposed solutions from Super Computing 2004 demonstration”, GGF Draft. - https://forge.gridforum.org/projects/ghpn-rg/document/Grid_Network_Services_in_the_SC04_Demonstrator/en/1

[7] Bas van Oudenaarde, Leon Gommans, Cees de Laat, Tal Lavian, Inder Monga, Arie Taal, Franco Travostino, Fred Wan, "Applications Drive Secure Lightpath Creation across Heterogeneous Domains", Submitted for IEEE Communications Magazine, Feature topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision, scheduled to be published in the March 2006 issue.

[8] GFD.38 Conceptual Grid Authorization Framework and Classification. M. Lorch, B. Cowles, R. Baker, L. Gommans, P. Madsen, A. McNab, L. Ramakrishnan, K. Sankar, D. Skow, M. Thompson - http://www.ggf.org/documents/GWD-I-E/GFD-I.038.pdf

[9] Job-centric Security model for Open Collaborative Environment, by Yuri Demchenko, Leon Gommans, Cees de Laat, Bas Oudenaarde, Andrew Tokmakoff, Martin Snijders. - Proceedings 2005 International Simposium on Collaborative Technologies and Systems (CTS2005). - May 15-19, 2005, Saint Louis, USA. - IEEE Computer Society, ISBN: 0-7695-2387-0. - Pp. 69-77.

[10] Distributed security infrastructure and services: Authorisation and Policy Enforcement, CNL2 D3.5 Deliverable. - https://doc.telin.nl/dscgi/ds.py/Get/File-55051/D3.5_-_Distributed_Security-03.doc

[11] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, “Resource Records for the DNS Security Extensions”, RFC4034. - http://www.rfc-archive.org/getrfc.php?rfc=4034

[12] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, “Protocol Modifications for the DNS Security Extensions”, RFC4035. - http://www.rfc-archive.org/getrfc.php?rfc=4035

[13] Internet2 Middleware Initiative. - http://middleware.internet2.edu/

[14] The eduPerson object class. - http://www.educause.edu/eduperson/

[15] Shibboleth Project. - http://shibboleth.internet2.edu/

[16] MACE-Dir-Groups: Grouper - http://middleware.internet2.edu/dir/groups/grouper/

[17] MACE – Signet. - http://middleware.internet2.edu/signet/

[18] R. Morgan and K. Hazelton, "A URN Namespace for MACE", RFC 3613, October 2003. - http://www.faqs.org/rfc/rfc3613.txt

[19] InQueue Federation. - http://inqueue.internet2.edu/

[20] InCommon Federation - http://www.incommonfederation.org/

[21] TERENA TF-EMC2 AA update page - http://www.terena.nl/tech/task-forces/tf-emc2/aai.html

[22] The Open Grid Services Architecture, Version 1.0 – 29 January 2005. - - http://www.gridforum.org/documents/GFD.30.pdf

[23] EGEE Global Security Architecture (GSA) rev. 2 - DJRA3.3 - https://edms.cern.ch/document/487004/1.1

[24] Virtual Organization Membership Service (VOMS) project homepage - http://infnforge.cnaf.infn.it/voms/

[25] VOMS Admin - http://edg-wp2.web.cern.ch/edg-wp2/security/voms/

[26] R. Alfieri, R. Cecchini, V. Ciaschini, F. Spataro, L. dell'Agnello, A. Frohner, K. Loorentey, "From gridmap-file to VOMS: managing Authorization in a Grid environment". - http://infnforge.cnaf.infn.it/voms/voms-FGCS.pdf

[27] VOMS Attribute Certificate for Authorisation. - http://infnforge.cnaf.infn.it/voms/AC-RFC.pdf

[28] GridShib - A Policy Controlled Attribute Framework - http://grid.ncsa.uiuc.edu/GridShib/

[29] Virtual Organisation Registration Procedure. By Maria Dimou, Ian Neilson. - https://edms.cern.ch/document/503245/

[30] User Registration and VO Membership Management Requirements document: https://edms.cern.ch/document/428034

[31] LCG/EGEE Virtual Organisation Security Policy. Version 1.1, by Ian Neilson - https://edms.cern.ch/document/573348/

[32] Web Services Trust Language (WS-Trust) - ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf

[33] Web Services Secure Conversation Language (WS-SecureConversation) - http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-secureconversation.asp

[34] Web Services Federation Language (WS-Federation) Version 1.0 - July 8 2003 – http://msdn.microsoft.com/ws/2003/07/ws-federation/

[35] XML Key Management Specification (XKMS 2.0), Version 2.0. W3C Recommendation 28 June 2005. - http://www.w3.org/TR/xkms2/

[36] RFC2903 – “Generic AAA Architecture”, C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, IETF Aug 2000, ftp://ftp.isi.edu/in-notes/rfc2903.txt

[37] Demchenko Yu. Virtual Organisations in Computer Grids and Identity Management. – Elsevier Information Security Technical Report - Volume 9, Issue 1, January-March 2004, Pages 59-76.

[38] Using VO concept for managing dynamic security associations, by Yuri Demchenko, Leon Gommans, Cees de Laat. - Accepted paper for the 21st IFIP International Information Security Conference "Security and Privacy in Dynamic Environments". May 22 - May 24, 2006, Karlstad University, Karlstad, Sweden.