Lightpath AAA – Gap Analysis

or

Filling the Gap with GAAA-P

SURFnet RoN Meeting

17-18 October 2005, Utrecht

Yuri Demchenko <demch@science.uva.nl>
Advanced Internet Research Group
University of Amsterdam

Outline

- Gap analysis – document structure
- Operation and requirements
  - Multidomain Optical LightPath (OLP) reservation operation
  - Requirements to AAA/Security services
- Extending GAAA Authorisation framework for OLP provisioning
- Using VO concept for dynamic security associations management
- Additional information

Approach and Conventions for the analysis and presentation

- Work in progress
- Technically detailed
- Abstraction from optical networking subject area to generic AAA and general security domain
  - Not a solution for network management but supporting AAA Authorisation functionality
- Attempted bridging between AAA, NREN/Internet2 and Grid domains
- OGSA and WSA architectural frameworks
- XML and Web Services development platform

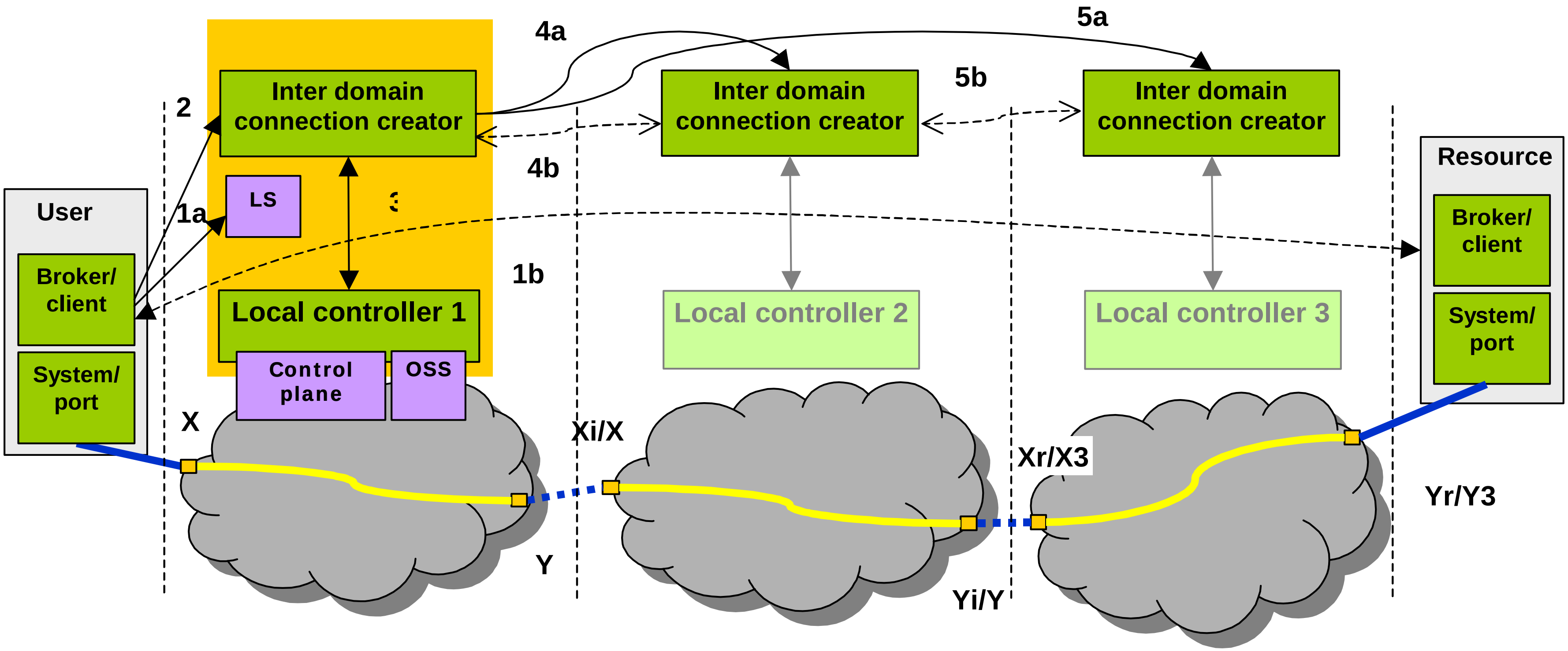
Gap analysis – Document structure

- Chapter 1 - OLPP and AAA
  - OLP provisioning process and sequences
  - Required functionality for AAA/Security services
- Chapter 2 – Existing solutions and gap analysis for AAA Authorisation infrastructure and related services
  - GAAA Authorisation framework
  - Shibboleth Attribute Authority Service infrastructure
  - VO management in Grid
- Chapter 3 – Extending GAAA Authorisation framework components for complex resource provisioning
  - Required new GAAA Authorisation framework functionality
  - Using VO model for dynamic security associations management

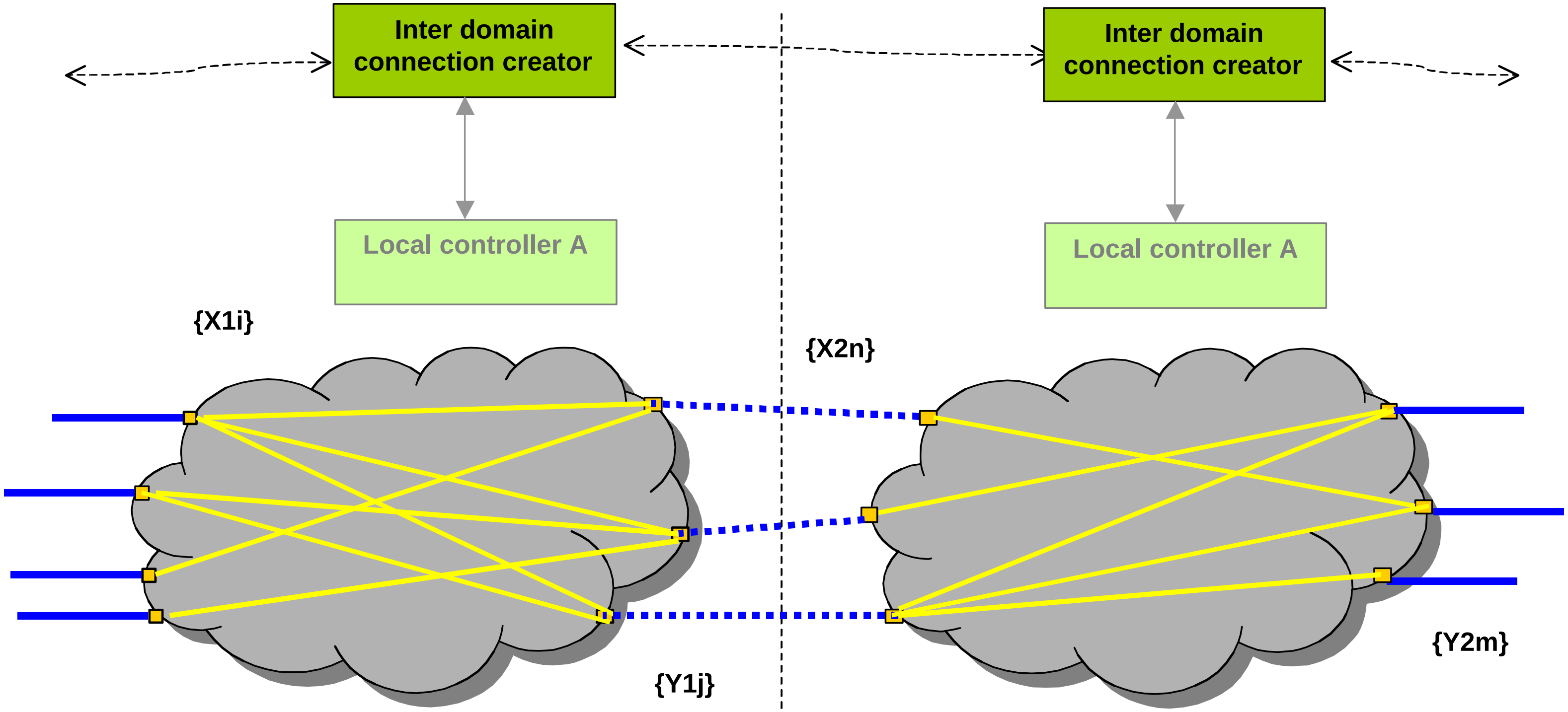
OLP provisioning operation – Basic Steps

- **Step 1.** The application broker or user client requests from the lookup services (LS) a path to a target system or resource
- **Step 2.** Building/calculation of the interdomain connection between User and Resource domains with specific parameters
- **Step 3.** Reservation of calculated path
  - Agent based allocation
  - Hop-by-hop allocation
- **Step 4.** Provision reserved OLP
  - Reservation ticket is used
  - Fall-back conditions

OLP provisioning operation – Simple multidomain model



Interdomain-intradomain trespassing graph


Inter domain connection creator
Local controller A
{X1i}
{Y1j}
{X2n}
Inter domain connection creator
Local controller A
{Y2m}

Binding policy and AAA service to XML/RDF based network description (example):

<Network aaa:Policy=“http://aaa-server/policy” aaa:Authority=“http://aaa-server/”>

Required AAA/Security services functionality for OLPP

- Authentication and Identity management
- Authorisation
- Attribute management
- Federation
- Trust management
- AuthN/Z services API

- Conceptual issues and models

Authentication and Identity management

- ✓ Authentication service must provide (to authenticated) user the Authentication credentials/confirmation in the form of
  - ◆ authentication ticket or token, or process/sessionID
  - ◆ <in contrary to browser based cookie>
- ✓ Any AuthN credentials must be traced back to the requestor's home organisation (HO) and common requestor-resource Trust Anchor (TA)
- Additional Identity management functionality by Identity provider (IdP) – for authenticated users
  - ◆ Single-sign-on (SSO)
  - ◆ Attribute Authority service (AAS)
  - ◆ Delegation and Federation
- Intra-/interdomain Identity and attribute translation/mapping
  - ◆ Based on established federations or bilateral trust relations

Authorisation and Policy management

- Authorisation relies on Authenticated or confirmed Identity
- Authorisation decision is made by evaluating service/resource request against access control policy
- Authorisation service may request additional confirmation or information about requestor’s Identity, Attributes to complete or verify required security context

✓ Based on successful AuthZ decision, AuthZ service may issue AuthZ ticket or token

✓ AuthZ service may operate in push, pull, or agent model

✓ Multi-domain Multiple policies evaluation may be required

  - ◆ via multiple policies or AuthZ decisions combination

- Operation may differ at reservation and provisioning stage
- Mutual authorisation may be required for some cases and applications

Attribute management and Federations

- User (and resource) attributes may be managed by Attribute Authorities Service (AAS) but in conjunction with the user/resource identity
- In inter-domain scenario, Attribute management may be delegated to association or federation membership services
  - ◆ E.g., VO in Grid applications or InCommon Federation in Internet2 Shibboleth infrastructure
- ✓ One of AAS specific function is management of *attribute namespaces* that is shared between interacting domains or mapped/translated by IdP
  - ◆ In this case, AA should provide potentially mapped attributes/namespaces, and interacting domains' IdP may provide trust management
- It is considered that two stages reservation and provisioning may require different strength of user ID and attributes confirmation

Note: Validity and trustworthiness of attributes will have effect on AuthZ decision trustworthiness and must be considered in overall trust relations analysis

Trust management

- ✓ All (*instant*) security related operations and resource allocation operations must be based on established and traceable trust relations based on PKI, SPKI or other models
  - ◆ These 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.
- ✓ All security valid decisions, e.g. delegation, AuthZ or reservation, and credentials must have unbroken/verifiable chain of trust
- • VO can be used for interdomain/inter-organisational trust management by providing trust anchor for interdomain credential management
  - ◆ DNSSEC may contain VO's/Federation's public key bound to the domain name and can be used for user/originator attributes verification and/or ***initial trust introduction***

AuthN/AuthZ API

- AuthN/AuthZ services API is required to flexibly and dynamically request AuthN, AuthZ and Attribute services from network services and applications
- AuthN/AuthZ services API should define and support
  - protocol(s)
  - request and response messages format
  - basic commands and extensibility procedure
  - namespace resolution/management
  - (enumerated) attribute values assignment/resolution
  - basic configuration profiles

Conceptual issues and models

- OLP provisioning model and process must be defined in details
  - It is used as a basis for defining AAA/Security functionality and operation
- GAAA Authorisation framework for complex resource provisioning
  - Multiple resources and multiple domains
  - Multiple policies combination and evaluation
- Driving policy re-factoring/implementation by separating flow management and policy enforcement
- Dynamic security context and trust management model
- VO infrastructure and management for dynamic user controlled service provisioning

Extending GAAA Authorisation Framework for OLPP

- GAAA AuthZ framework – two basic profiles are defined
  - GAAA-RBAC for Collaborative Environment
  - GAAA-P for interdomain network/resource provisioning
- Major GAAA-P components/extensions
  - Workflow control in the GAAA based provisioning model
    - WSFL and WSBPEL as upper layer to (stateless) WS/WS-Security
  - Dynamic trust management using federated trust model
    - Based on dynamic VO federation model
    - Compatibility with GridShib-SAAS
  - Attributes and metadata resolution and mapping
    - Support of common naming scheme and resolution
  - Policy combination and aggregation
    - For complex multi-component and multidomain resources
    - For combined policy audit/evaluation

Workflow management

- Separate policy evaluation and flow control and make flow control interpretable at runtime
  - Policy is a static set of rules that in general can be defined by the agreement between user and provider
  - Workflow is an instant dynamic process that orchestrates interaction of multiple services and processes to deliver final service to the requestor
- Workflow management for two basic provisioning scenarios
  - ***Centralised:*** Reservation (and provisioning) is controlled by one of domain Interdomain Connection Controller (ICC), e.g. from user domain, and the workflow is managed by a single ICC
    - individual policies are evaluated centrally and published into central repository
  - ***Distributed:*** Reservation (and provisioning) is chained and the workflow object may need to be transferred between participating domains
    - individual policies are evaluated locally in each domain, without populating policy between all participating domains
- Available technologies and tools
  - OASIS BPEL and IBM's WSFL
  - Oracle and Apache plugins for Eclipse

Dynamic trust management

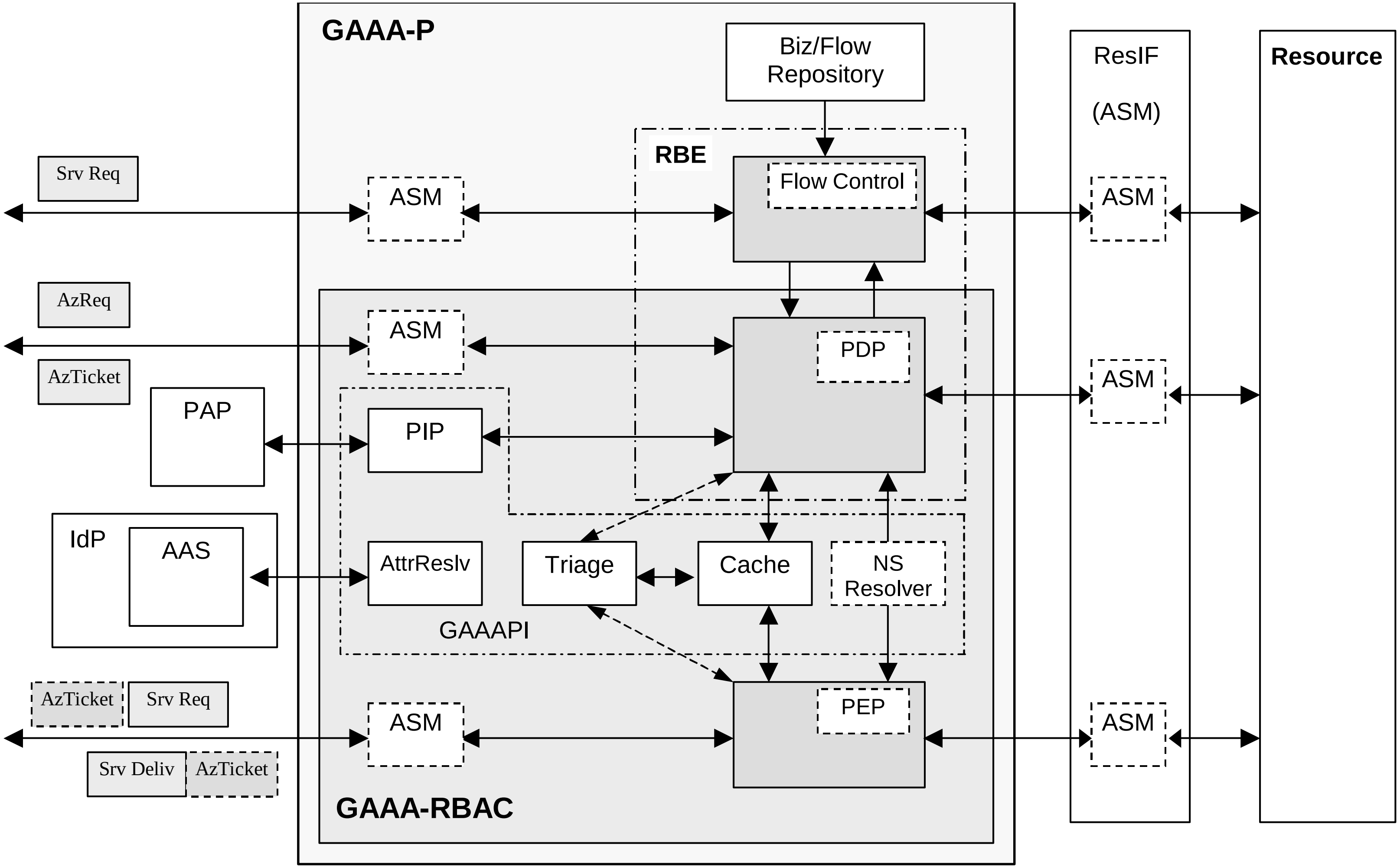
Dynamic trust management is a foundation for secure user controlled service provisioning

- 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
- During dynamic trust negotiation, in general, or security context establishing, in particular, negotiating parties must present initial credentials that must have verifiable trust chain to the mutually trusted authority
- The framework for (dynamic or session based) trust and credentials negotiation for Web Services is defined in two complimentary specifications WS-Trust (WST) and WS-SecureConversation (WSSC) and WS-Federation (WSF)
  - 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)
- Initial trust relations (or security context) establishment is considered outside of the WS-* scope and must be presented in all WS-* interactions in a form of trust anchor (TA) or business anchor (BA)

Extending GAAA Toolkit to support new functionality

- SAML 2.0 assertion and protocol support, including SAML XACML profile that will simplify AuthZ tickets management
- XACML policy support as a policy meta-format and exchange format
- Simple policy management tools supporting multiple policy formats, first of all, AAA-format and XACML
- Support for different types of secure credentials, in particular, X.509 PKI Certificate and Attribute Certificate, SAML assertions, and related callouts to issuing authorities, in particular VOMS and Shibboleth
- WS-Trust Secure token support and Secure Token Service (STS) functionality for credentials mapping and dynamic trust management
- Integration with GT4 and gLite Authorisation Framework
  - Using GT4 WS/messaging firmware to provide WS-based access to GAAA_tk authorisation service, to allow easy GAAA_tk integration into different applications
  - Adding GAAA AuthZ callouts to GT4/gLite AuthZ framework; this will allow using GAAA RBE as one of regular services for GT4 and gLite
  - Integrating GAAA AuthZ/RBE into GT4 AuthZ framework as one of PDP’s

Extended GAAA Toolkit structure



Using VO for Dynamic Resource Provisioning

Issues to be taken into account when considering VO for dynamic resource provisioning:

- Current VO management and VOMS (VO Membership Service) infrastructure are rather designed for long-term collaborative projects
  - ◆ VO setup is a complex long-time procedure and cannot be used as a solution for the global ad-hoc dynamic trust establishment
  - ◆ VOMS server Attribute Certificate is based on X.509 AC for Authorisation, its use for Grid authorisation (with GT) suggests using Proxy Certificate
  - ◆ VOMS client-server protocol is not clearly defined
  - ◆ Current VOMS implementation has no flexible attribute namespace management (and corresponding procedure and policy)
- Dynamic VO infrastructure must provide a solution for dynamic distributed trust management and attribute authority
  - ◆ VOMS provides all necessary functionality for creating ad-hoc dynamic VO associations
  - ◆ GridShib (GT4/WS-enabled) profile can be used for VO with distributed membership management
    - – (Use Grouper and Signet tools for Attribute/group/roles management/editing)

Dynamic Security Associations

- **Session** – establishes security context in the form of session key that can be a security token or simple UID bound to secure credential/context
  - Session may associate/federate users, resources and actions/processes
- **Job/workflow** – more long-lived association and may include few sessions
  - May need to associate 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
  - Security context may change during workflow execution or Job lifetime
  - Job description may contain both user and resource lists and also provide security policy and trust anchor(s) (TA)
- **Project or mission oriented cooperation** – established for longer time cooperation (involving people and resources) to do conduct some activity
  - This is actually the area of currently existing VO associations
- **Inter-organisational association or federation** – established for long-term cooperation, may have a wide scope of cooperative areas
  - This is the area of inter-university associations
    - Shibboleth is specially designed to support this kind of federations (e.g. InCommon or InQueue)

VO Operational Models

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

VO and DNSSEC – Feasibility Analysis

- Existing LCG/EGEE VO registration procedure allows actually using DNSSEC for populating 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
  - VO registry problem can be solved
- DNSSEC limitations
  - 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 [RFC4034]
  - DNSSEC domain record (in our case VO domain name) and key must be signed by upper layer domain’s key, and therefore DNSSEC trust tree must be compatible with the application oriented trust domain

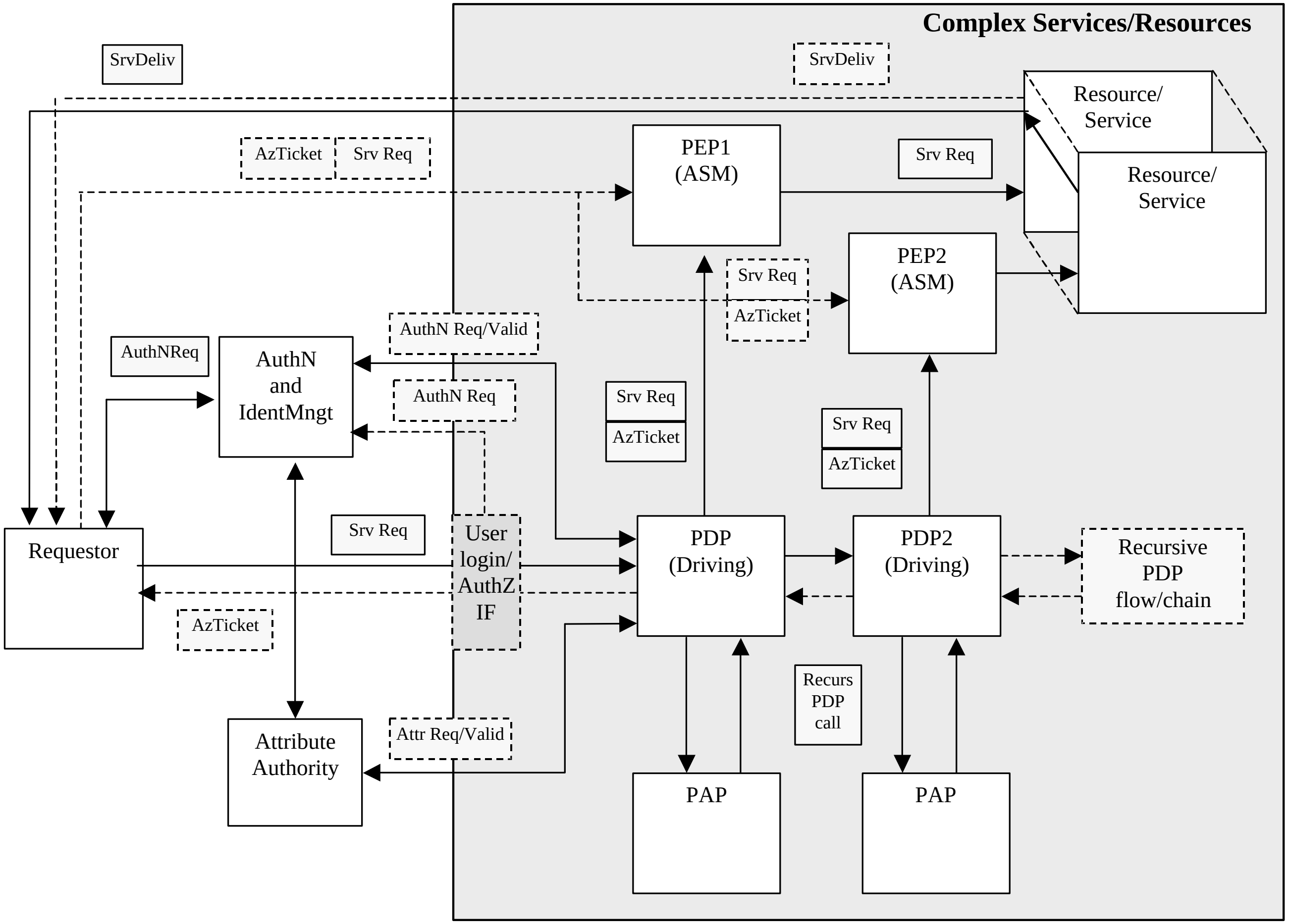
Additional information

- GAAA AuthZ service/infrastructure operation in handling multiple policies and complex resource access
- Internet2 Shibboleth Attribute Authority System
- GridShib Profile
- VO practice in EGEE/LCG

Basic GAAA Authorisation operational models

- The push authorization model
- The pull authorization model
- The agent authorization model

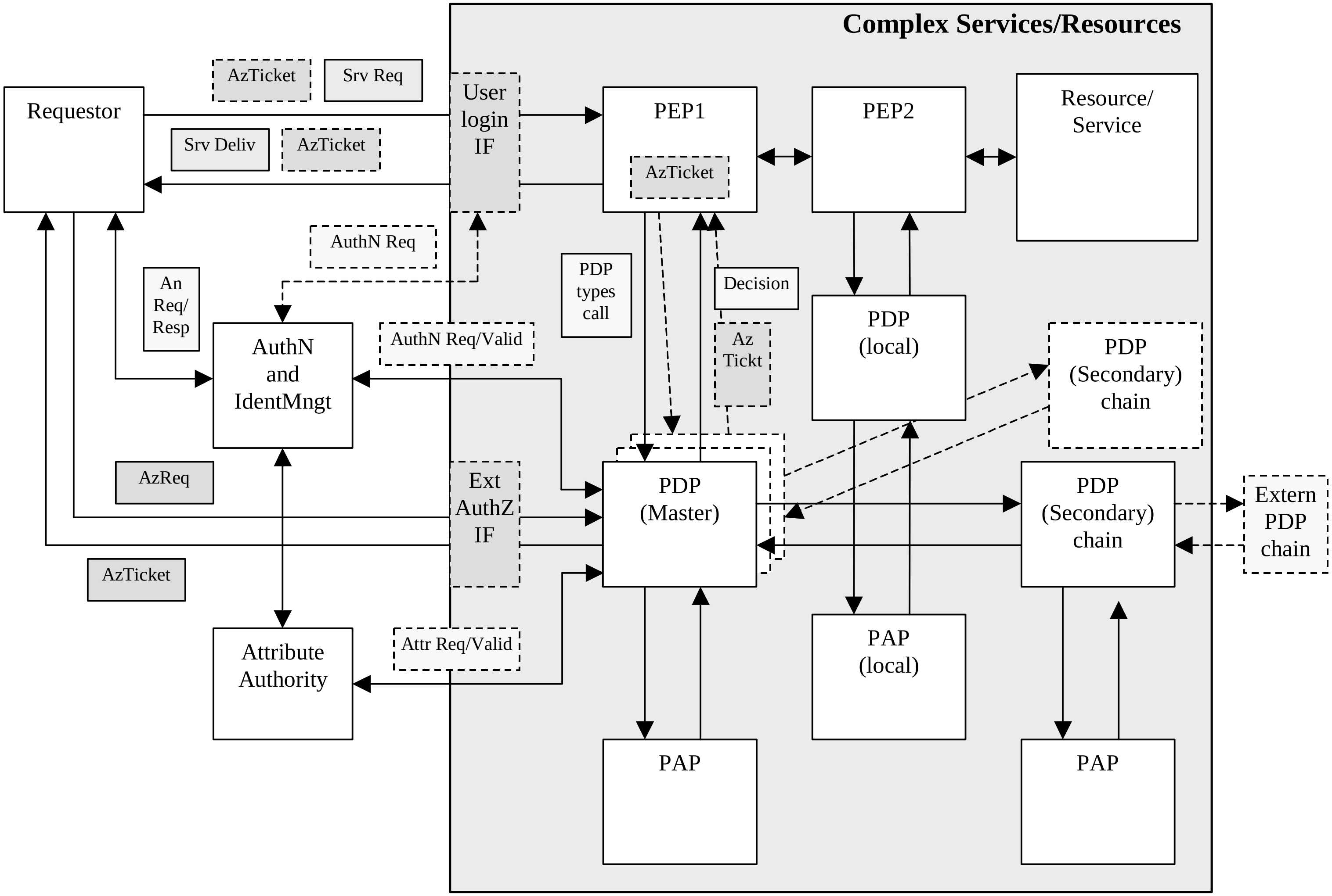
Authorisation in complex Resource/Service



Complex/multi-component resource

Combined push and agent model

Multiple/multi-domain policies combination



Multiple policies and/or multiple PDP's

General implementation suggestions

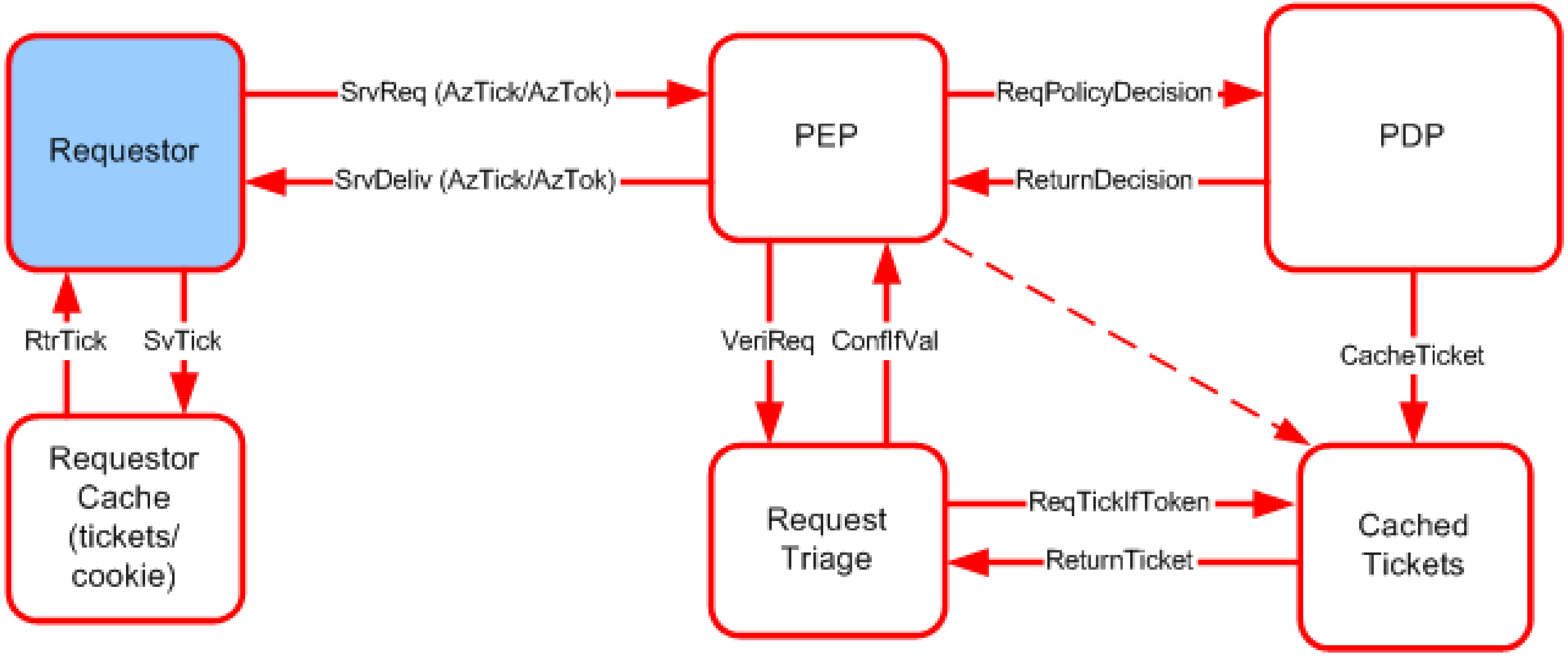
- PDP and PAP must share common namespace
- Policy and respectively PAP should be referenced in the request message explicitly or known to PEP and PDP a priory
- Every PEP in the chain of policy enforcement should take care of the whole request evaluation/enforcement by calling to a single (master) PDP.
  - PEP should not do multiple decision combination.
- Only one PDP should provide a final decision on the whole request
  - However, PEP may have a possibility to request different PDP types based on request semantics/namespace and referred policy
- When using ticket/token based access control model, the PEP should understand and have a possibility to validate the AuthZ ticket issued by trusted PDP
  - The AuthZ ticket should have validity and usage restriction and contain information about the decision and the resource.
- For the further validation of the AuthZ tickets/token, the PEP may cache the ticket locally to speed-up the validation procedure.

Specific implementation suggestions for OLPP

OLPP operation includes at least three stages: lookup, reservation and provisioning/delivering.
This brings the following specifics:

- User/requestor credentials and consequently trust model may be different at the reservation and provisioning stages
- Reservation ticket used at the resource/service consumption stage must include all reservation tickets for the whole OLP (or complex resource)
- Multidomain OLPP requires interdomain trust management that can be solved by establishing general/common security federation or managed via delegation between inter-operating domains
- Interdomain trust management can be solved by using open trust introduction model, e.g. using DNSSEC

Tickets/Tokens handling in AuthZ system



- AuthzTicket is issued by PDP and may be issued by PEP
- AuthzTicket must be signed
- AuthzTicket contains all necessary information to make local PEP-Triage Request verification
- When using AuthzTokens, AuthzTickets must be cached; Resolution mechanism from token to ticket must be provided

Before deploying GAAA Authorisation infrastructure

Design conventions and agreements

- Key distribution and trust establishing
  - *Currently, in search of simple consistent model*
- Policy definition and format including subject, attributes/roles, actions semantics and namespaces
  - Compatibility with existing formats, e.g. SAML, XACML
  - Policy format defines/defined by the PDP implementation
- Secure credentials/ticket format
  - Standard vs proprietary
- Protocols and Messages format
  - SOAP + XACML Request/Response
  - SOAP + SAMLP + XACML

Internet2 Federations and Supporting Middleware Tools

- eduPerson/eduOrg (LDAP Directory Schema)
- SAAS (Shibboleth Attribute Authority Service) includes architectures, policy structures, practical technologies to support inter-institutional sharing of web resources subject to access controls
- Grouper - an open source toolkit for managing groups
- core element of a common infrastructure for managing group information across integrated applications and repositories.
- Signet - a privilege management service for centralized management of user privileges across a range of applications
- The InCommon federation (http://www.incommonfederation.org/
- InQueue test federation

Implementation suggestions for SAAS

- SAAS requires LDAP based EduPerson format for Identity and attributes
- 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)
- 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
  - 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.
- There is no special IdP/ServP directory or resolution service for SAAS
  - Trusted providers are preconfigured by manually maintained files sites.xml and trust.xml
- Shibboleth’s AA/IdP use and understand (only) own namespace “urn:mace” which are preconfigured in both IdP Service Provider

GridShib Attribute Handling Models

- Basic Globus-Shibboleth integration without anonymity using attributes request/pull by the resource from the trusted SAAS
- Basic Globus-Shibboleth integration without anonymity using attributes provided by the requestor which are previously obtained from the trusted SAAS
- 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.
- 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)

Establishing Security context for GridShib

1. The Grid User and the Grid Service each possess an X.509 credential that uniquely identifies them.
2. The Grid User is enrolled with a Shibboleth Identity Provider (IdP), and correspondently with IdP’s AA.
3. The IdP is able to map the Grid User's X.509 Subject DN to one and only one user in its security domain.
4. The IdP and the Grid Service each have been assigned a unique identifier called a providerId.
5. The Grid Client application has access to the Grid User's X.509 certificate and the IdP providerId. This information is used to create Proxy Cert that will contain IdP providerId and signed by the User private key.
6. The Grid Service has a set of certificates identifying IdP/AAs that it trusts to provide attributes suitable for use in authorization decisions.
7. The Grid Service and the IdP rely on the same metadata format and exchange this metadata out-of-band.
8. It is assumed that all X.509 End-Entity Certificates (EEC) are issued by CAs that are trusted by all parties mentioned in this document.

VOMS and SAAS interoperation and integration

- GridShib profile targets for SAAS integration into Grid/GT environment
  - Expected to provide a framework for combing well developed Shibboleth attribute management solutions and VOMS functionality
- Differences in VOMS and SAAS operation on the user/client and service/resource sides
  - 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
  - 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.
  - VOMS requires user ID and therefore doesn’t provide (user) controlled privacy protection (in contrary to Shibboleth).

EGEE/LCG Practice: VO Registration Procedure

1. Naming the VO
2. Request VO integration into existing EGEE infrastructure from one of designated bodies EGEE Generic Applications Advisory Panel (EGAAP) or NA4/SA1 Joint Group
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).
4. Populating a VO. Candidate entries in the VODB are passed through successful Registration process and Registration database additions.
5. Integrating VO into existing infrastructure
6. Organising support structure for the VO

EGEE/LCG Practice: VO Security Policy

VO enrolment process MUST capture and maintain at least the following information:

1. VO Name
2. VO Acceptable Use Policy
3. Contact details and certificates for the VO Manager and at least one Alternate
4. Email address of a generic VO contact point for the VO managers
5. A single email address of the security contact point
6. URL of one or more VO Membership Servers