Enabling Grids for E-sciencE

Use of VOMS Attributes: semantics and suggestions

Vincenzo Ciaschini

MWSG 12

Stockholm 12-13/06/07

www.eu-egee.org

Index

Introduction

- **What are VOMS attributes?**
- **Three broad types:**
  - Groups.
  - Roles.
  - Generic Attributes.

- **Different semantics.**
- **Usually, not (easily) interchangeable.**

GROUPS

Group Attributes

- **Group Attributes are meant to represent an organizational structure:**
  - They are hierarchical.
  - Membership in a subgroup $\Rightarrow$ Membership in the parent group.
- **Groups are not deniable.**
  - All group membership information will always be returned.
- **Group membership in at least the root group is mandatory.**
  - By convention, the root group has the same name as the VO.
  - Implies that all users will be in the root group.

Group Attributes

- **Group naming requirements:**
  - A group name may use the following characters:
    - [a-zA-Z0-9-_.]
    - '/' is special and should not be used in a group name.
      - *See next slide on why.*
- **There are no limitations on the length of group names.**
- **There are no limitations on the depth of the group tree.**
- **Groups are returned in no particular order, except:**
  - The root group will be the first group.
  - Users may request a specific ordering.
- **No hard-coded requirement on naming standards.**

Group Attributes

- **Group representation:**
  - Groups a represented in a filesystem-like way:
    - `/dteam/ce/PL`
    - Means: member of the group *PL*, which is a subgroup of *ce*, which is a subgroup of *dteam*, which is the root group.

- **`/dteam/ce/PL` implies that the following group will also be returned:**
  - `/dteam/ce`
  - `/dteam`

- **Remember:**
  - Membership in a subgroup ⇒ Membership in the parent group.
  - Group membership is not deniable.

ROLES

Role Attributes

- **Role Attributes are meant to be used when additional privileges are needed.**
  - They are unstructured.
- **Roles are always granted in the context of a specific group.**
  - There are no freestanding roles.
    - Though you could just assign a role in the main group.
- **Roles are assigned to users, not to groups.**
- **I.e: Roles are assigned to users as members of a specified group.**

Role Attributes

- **Role naming conventions:**
  - A role name may use the following characters:
    - [a-zA-Z0-9-_.]
  - A role name is represented as: /Role=<role name>

- **Roles are not normally granted:**
  - Besides having the right to receive them, the user must also explicitly request them.
    - Normally, no roles are present in the credentials.

- **Special name: NULL**
  - Implies no specific role.
    - Will be phased out. (see later)

FQAN

FQANs

- **Compact way of representing groups and roles.**

- **Syntax:**
  - <group name>[/Role=<rolename>][/Capability=<cap name>]
  - Example:
    - /atlas/Role=Production/Capability=NULL

- **Notes:**
  - Parts within [ ] will soon be optional (Voms 1.8). Specifically:
    - [/Capability=<cap name>] is deprecated.
      - *Ignore it.*
    - <rolename> == NULL means no specific role.
      - *It is deprecated.*
    - [/Role=<rolename>] may only be present when <rolename> != NULL (voms 1.8)
      - *Implementations should be prepared.*

FQANs

- **One FQAN for every group.**
- **One FQAN for every role.**
- **FQANs are returned without any predefined order, except:**
  - The user may specify a preferred order.
  - If the user does not specify anything, this FQAN will be the first:
    - <voname>/Role=NULL/Capability=NULL

- **Services are not obliged to consider all FQANs for authorization decisions.**
  - But if they do not, then they should consider the first <n> at least, with <n> chosen by the service.

Generic Attributes

Generic Attributes

- **Generic Attributes (GAs) are couples (name, value)**
  - Name and Value are both chosen by the VO admin.
- **They are non deniable.**
  - All of them will always be present in the credentials.
- **There is no structure among them.**
  - Each one is independent of all the others.
- **There may be any number of GAs per user.**

Generic Attributes

- **GAs conventions:**
  - Name: ASCII Printable
  - Value: ASCII Printable

- **GAs may be associated to:**
  - Directly to users.
    - Only the specific User will receive that specific GA.
  - To Groups:
    - All Users of the specified group will receive that specific GA.
  - To Groups and Roles:
    - Only Users holding the specified role inside the specified group will receive that specific GA.

Generic Attributes

- **GA Representation:**
  - <name>=<value> (/some/group)
  - Examples:
    - userid=vciaschi (/vo)
    - HLR=hlr.to.infn.it (/vo)
    - Guarantor=JohnSmith (/vo/group)

Example

Example

- **Authorize all users except a subset:**
  - Solution 1: (using groups)
    - Put the users that should not be authorized in a group.
    - Explicitly authorize all groups but that.
    - Pro: No Deny.
    - Con: May be daunting if a lot of groups must be specified.
  - Solution 2: (using groups)
    - Put the users that should be authorized in a new group.
    - Authorize only that group.
    - Pro: Easy configuration.
    - Con: Group proliferation. Prone to errors

Example

- **Authorize all users except a subset:**
  - Solution 3: (using roles)
    - Give the users the same role.
    - Authorize the role.
    - Pro: Easy configuration
    - Con: Extra step for users, difficult setup
  - Solution 4: (using GAs)
    - Give the same GA to the group.
    - Authorize only the GA
    - Pro: Easy to setup.
    - Con: Difficult if the union of the sibling groups is not the parent group or if the intersection with the blocked group is not null.

Contacts

- Vincenzo Ciashini (vincenzo.ciaschini@cnaf.infn.it)
- Valerio Venturi (valerio.venturi@cnaf.infn.it)
- Andrea Ceccanti (andrea.ceccanti@cnaf.infn.it)