Open Apereo 2016

100% Open for Education

Exploring Internet2 Grouper & NIST RBAC/ABAC

Misagh Moayyed, IAM Architect, Unicon
William G. Thompson, Jr,. CISSP, Lafayette College

Exploring Internet2 Grouper & NIST RBAC/ABAC

Introduction to NIST RBAC/ABAC models and standards

- INCITS 359-2012 Role Based Access Control
- INCITS 494-2012 RBAC - Policy-Enhanced
- NIST Special Publication 800-162 Guide to Attribute Based Access Control Definition and Considerations

How do these models and standards apply to Grouper and Grouper based access management systems?

Shout out to Shawn McKinney, symas.

https://www.linkedin.com/in/shawn-mckinney-5238672

https://symas.com/

- OpenLdap
- Apache Fortress
- JavaOne Open Source IAM Expert Panel

“Good artists copy, great artists steal” - Steve Jobs quoting Picasso quoting Igor Stravinksy quoting T.S. Elliot quoting ...” - http://quoteinvestigator.com/2013/03/06/artists-steal/

INCITS 359-2012 Role Based Access Control

INCITS 494-2012 RBAC - Policy-Enhanced

NIST Special Publication 800-162 Guide to Attribute Based Access Control Definition and Considerations

Introduction to NIST RBAC/ABAC models and standards

NIST: Role Based Access Control (RBAC) and Role Based Security - http://csrc.nist.gov/groups/SNS/rbac/

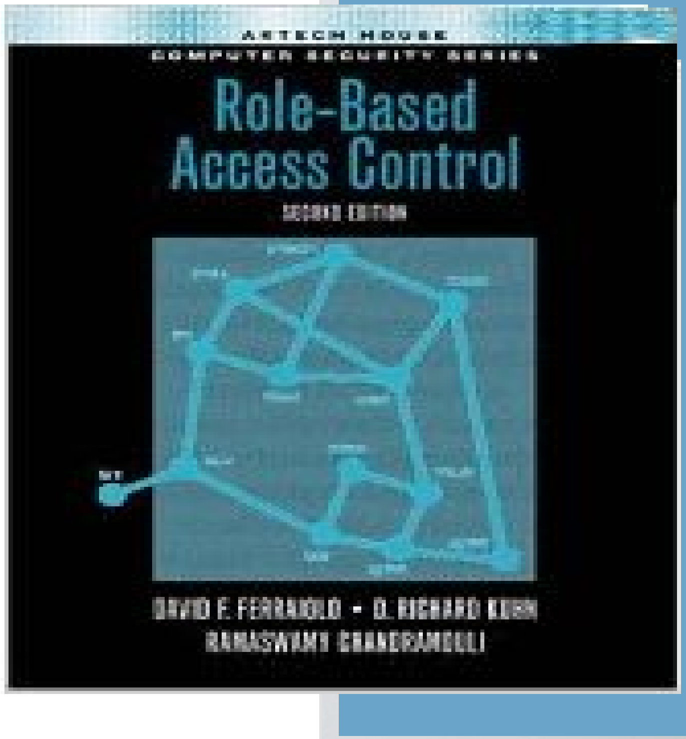
Formalized by David Ferraiolo and Rick Kuhn in Role-Based Access Controls (1992)

NIST RBAC model (Sundhu, Ferraiolo, and Kuhn, 2000)

Initially released as ANSI INCITS standard in 2004

Updated and expanded in 2012 as

- INCITS 359-2012 Role Based Access Control
- INCITS 359-2012 Role Based Access Control - Policy Enhanced



INCITS 359-2012 Role Based Access Control (RBAC)

“...initiated by National Institute of Standards (NIST) in recognition of a need...for **consistent and uniform definition of role based access control (RBAC) features.**”

“...lack of widely accepted model resulted in uncertainty and confusion about RBAC’s utility and meaning. This standard seeks to resolve this situation by using a **reference model to define RBAC features** and then describing the functional specifications for those features.”

Developed by InterNational Committee for Information Technology Standards (INCITS) and approved by American National Standards Institute (ANSI). INCITS Committee members: Apple, EMC, IBM, IEEE, Intel, NIST, Oracle, Microsoft, Purdue University, US DOD, US DHS, VMWare,...

INCITS 359-2012 Role Based Access Control (RBAC)

RBAC Reference Model & Functional Specification

Reference Model has four model components:

Core RBAC - Users, Roles, Perms, Session (aka Role Activation)

Role is a set of permissions

Users assigned to Roles (creates effective permission sets for users)

Users activate one or more Roles in an application Session

Hierarchical Roles

Roles can inherit privilege sets

Static Separation of Duties (SSD)

Can be assigned a subset of roles in a particular collection

Dynamic Separation of Duties (DSD)

Can activate a subset of roles in a particular collection

ANSI INCITS 359 RBAC has three standards interfaces:

1. Administrative - CRUD permission, role, hierarchy assignments, etc.
2. Review - policy interrogation (grouper audit/report/etc)
3. System - policy enforcement (authN/authZ CAS and Spring/Shiro/.Net))

Map RBAC Reference Model to Grouper terminology

(users, roles, permissions, operations, and objects) -> (users/groups, roles, permissions, actions, resources)

Map RBAC functional specification to Grouper functions

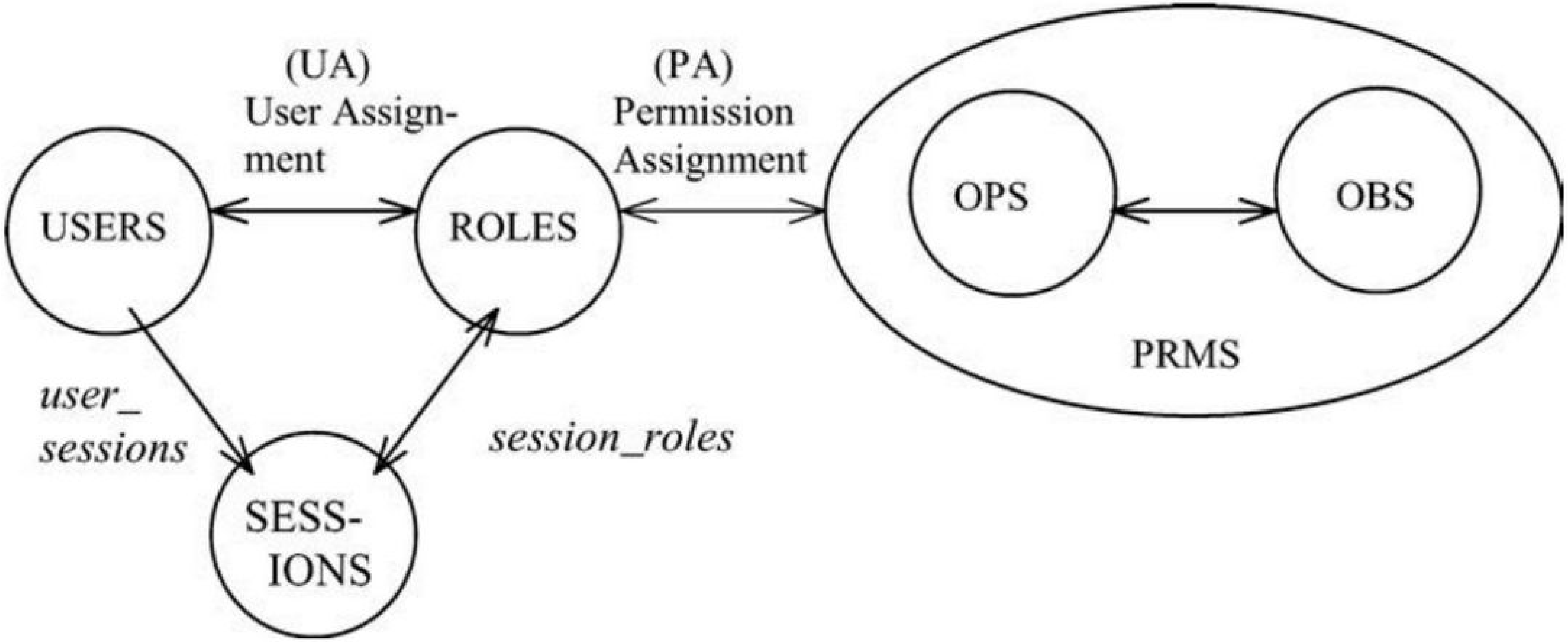


Figure 1: Core RBAC

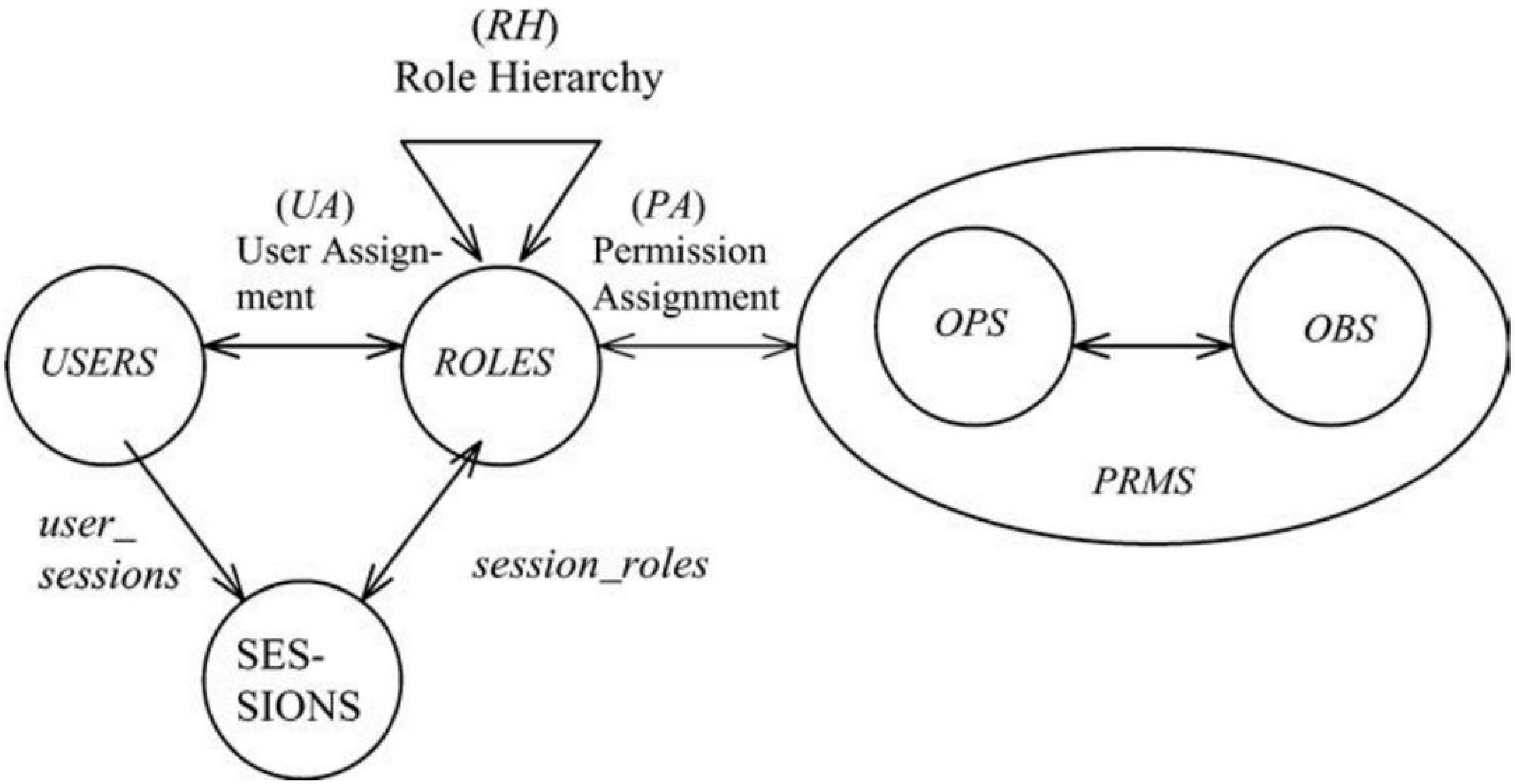


Figure 2: Hierarchical RBAC

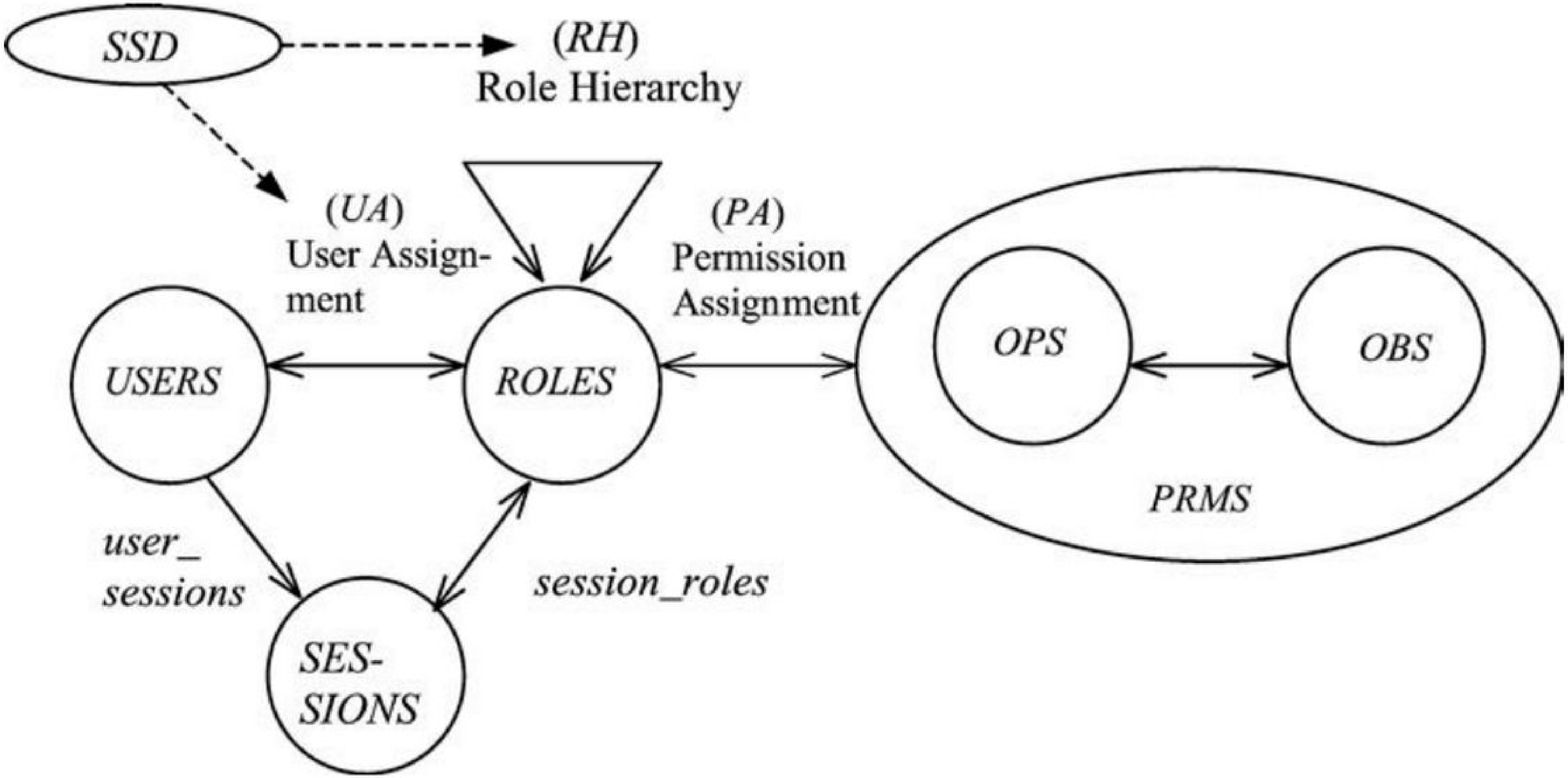


Figure 3: SSD within Hierarchical RBAC

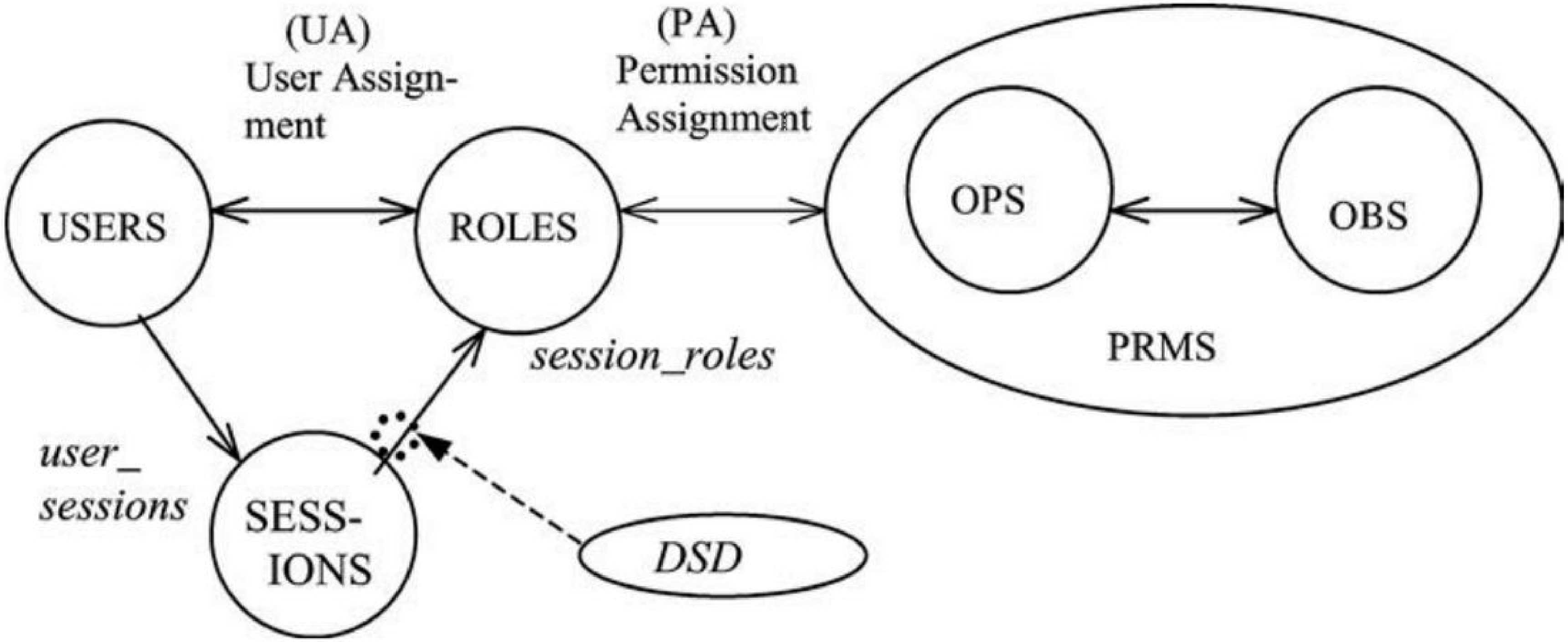
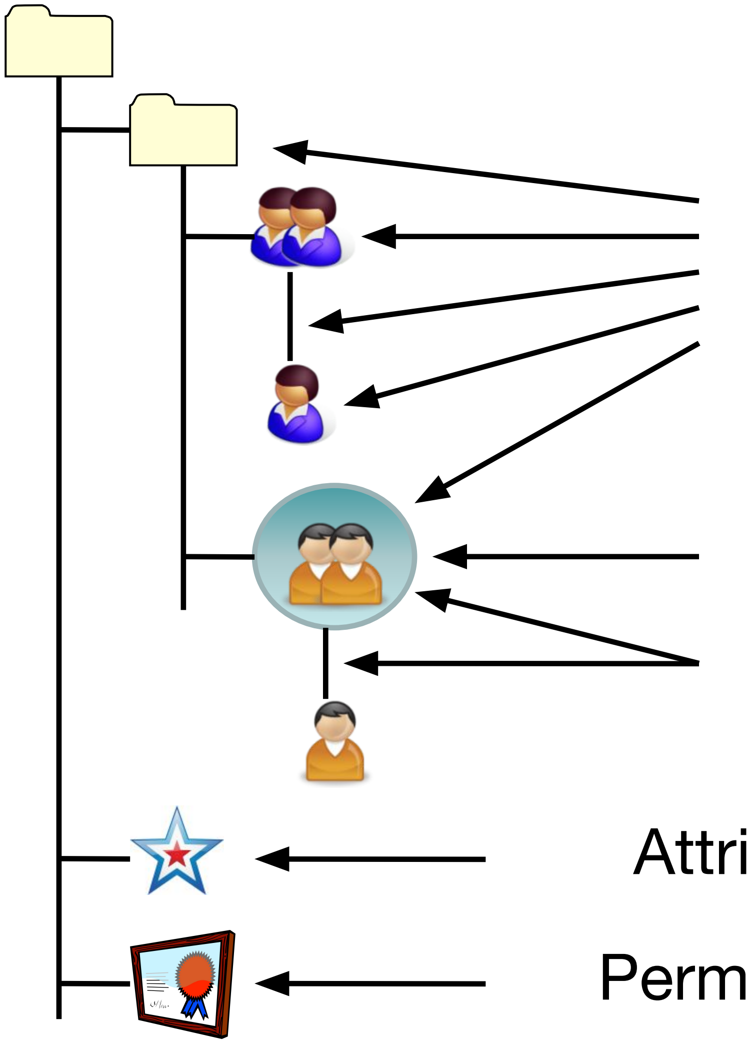


Figure 4: Dynamic Separation of Duty Relations

Grouper RBAC



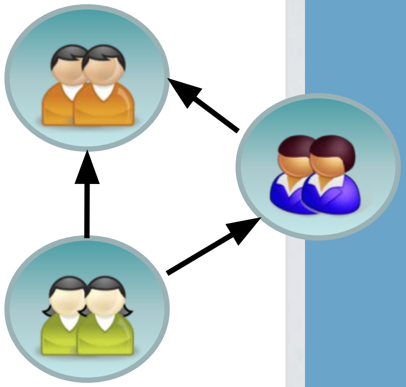
Attributes

Roles

Permissions

Attribute definition

Permission definition

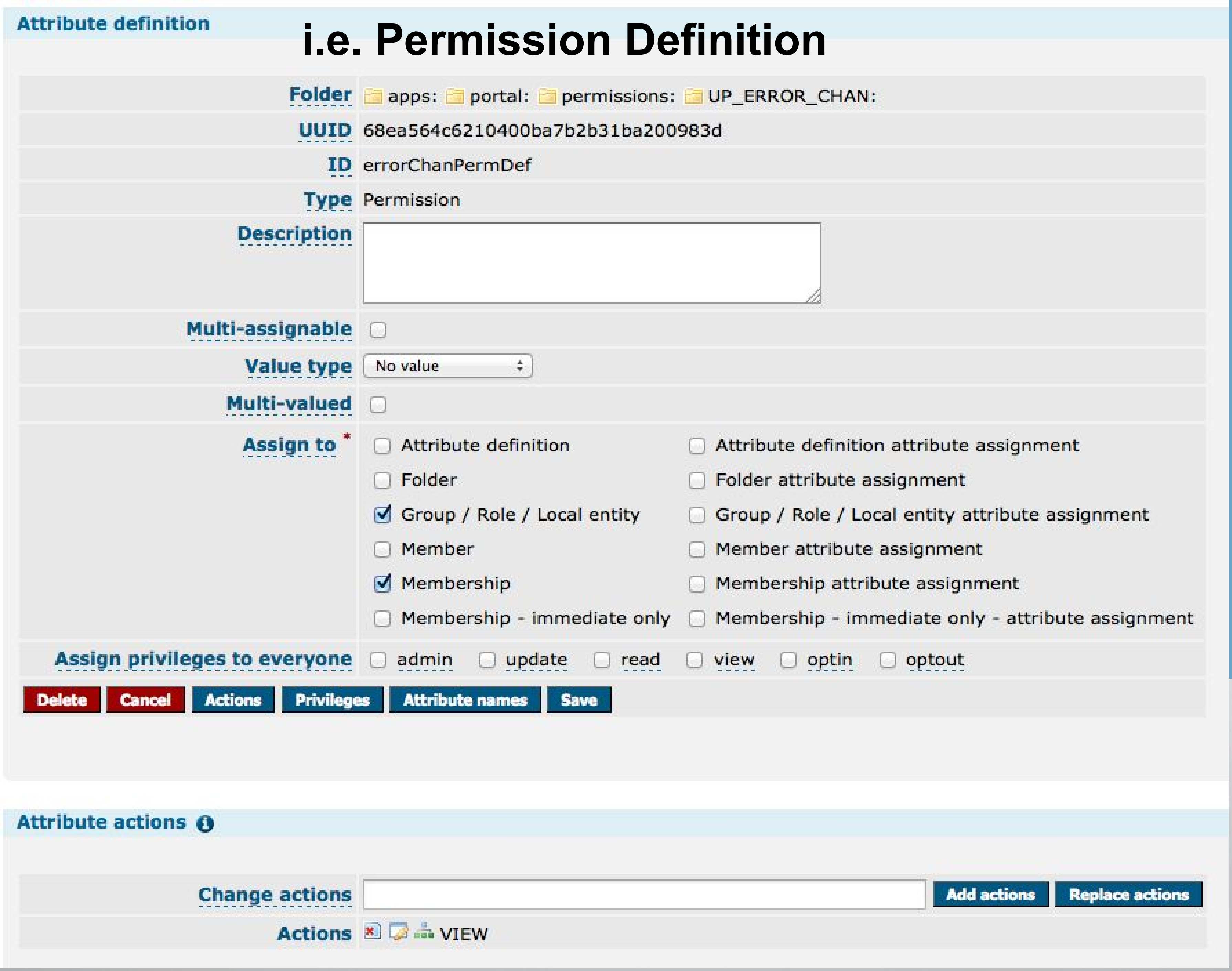


Role inheritance

Delegation model extends that for Groups

RBAC	Grouper	uPortal
User	Subject	Principal
Operation	Action	Activity
Object	Resource	Target

i.e. Permission Definition

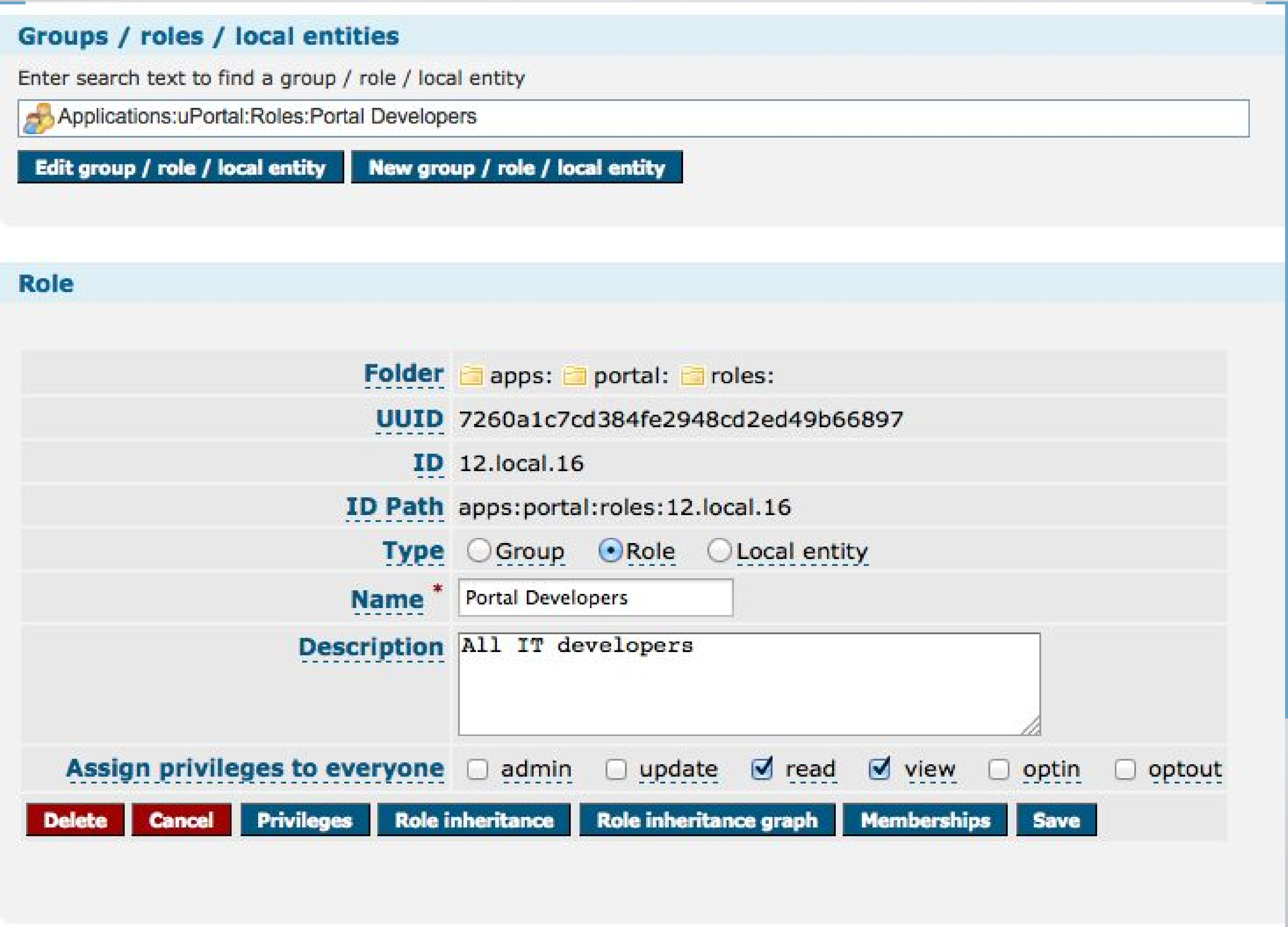


Attribute definition

Folder	apps: portal: permissions: UP_ERROR_CHAN:
UUID	68ea564c6210400ba7b2b31ba200983d
ID	errorChanPermDef
Type	Permission
Description	
Multi-assignable	□
Value type	No value
Multi-valued	□
Assign to *	□ Attribute definition □ Folder ☑ Group / Role / Local entity □ Member ☑ Membership □ Membership - immediate only
	□ Attribute definition attribute assignment □ Folder attribute assignment □ Group / Role / Local entity attribute assignment □ Member attribute assignment □ Membership attribute assignment □ Membership - immediate only - attribute assignment
Assign privileges to everyone	□ admin □ update □ read □ view □ optin □ optout

Attribute actions

Change actions	
Actions	VIEW



Groups / roles / local entities

Enter search text to find a group / role / local entity

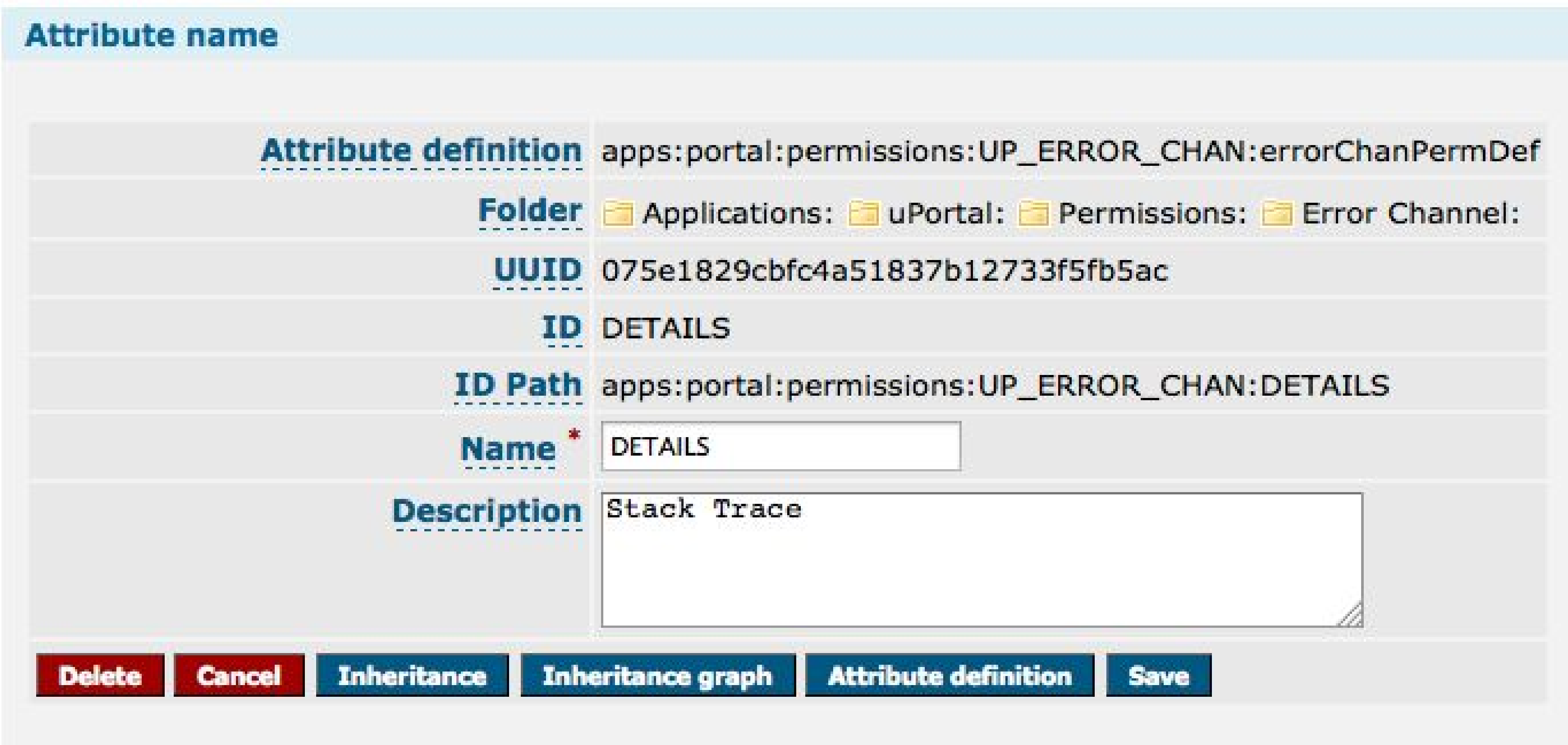
Applications:uPortal:Roles:Portal Developers

Edit group / role / local entity
New group / role / local entity

Role

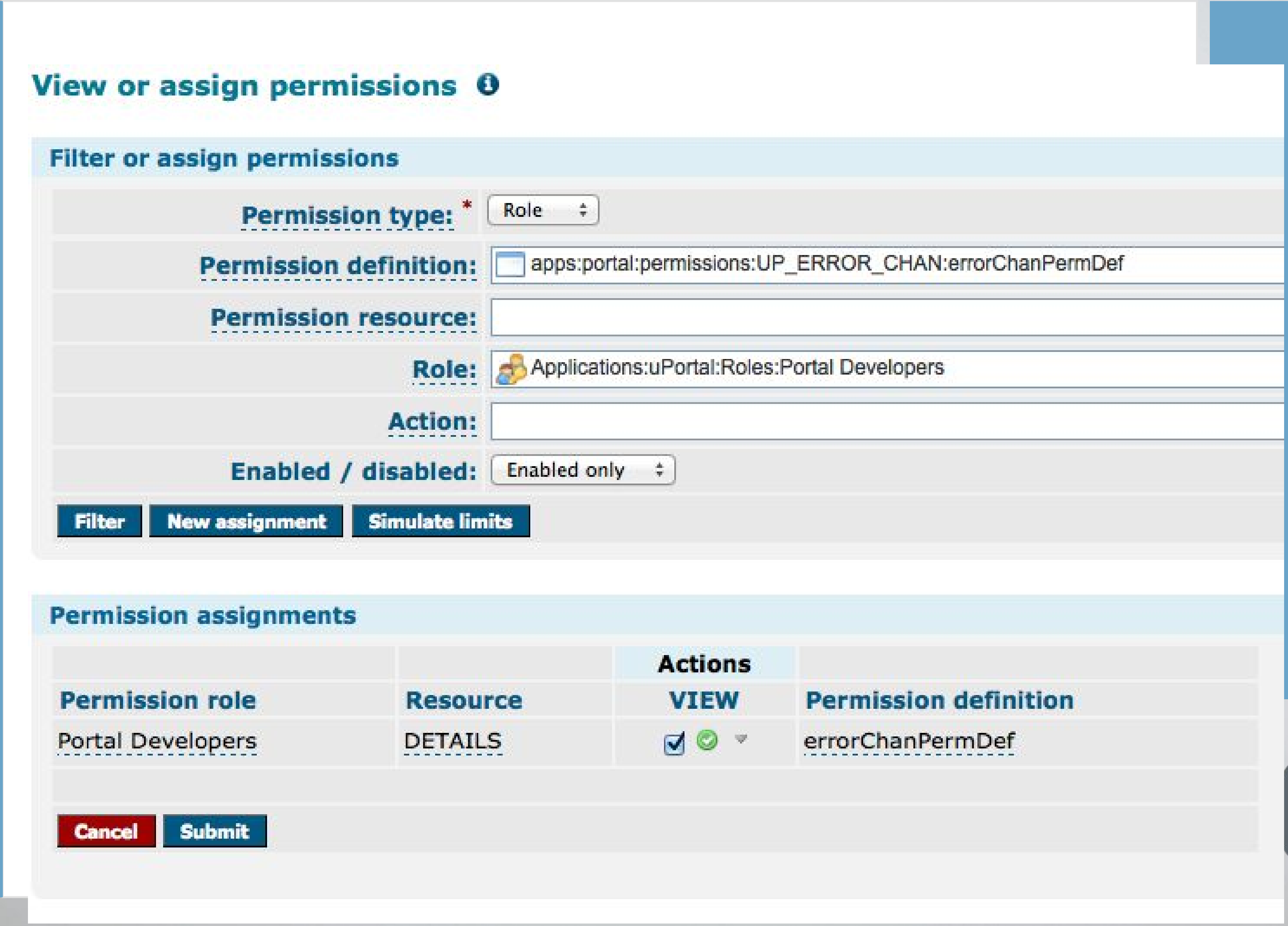
Folder	apps: portal: roles:
UUID	7260a1c7cd384fe2948cd2ed49b66897
ID	12.local.16
ID Path	apps:portal:roles:12.local.16
Type	○ Group ● Role ○ Local entity
Name *	Portal Developers
Description	All IT developers
Assign privileges to everyone	□ admin □ update ☑ read ☑ view □ optin □ optout

Resource/Object - Error Channel Details



Attribute name

Attribute definition	apps:portal:permissions:UP_ERROR_CHAN:errorChanPermDef
Folder	Applications: uPortal: Permissions: Error Channel:
UUID	075e1829cbfc4a51837b12733f5fb5ac
ID	DETAILS
ID Path	apps:portal:permissions:UP_ERROR_CHAN:DETAILS
Name *	DETAILS
Description	Stack Trace



View or assign permissions

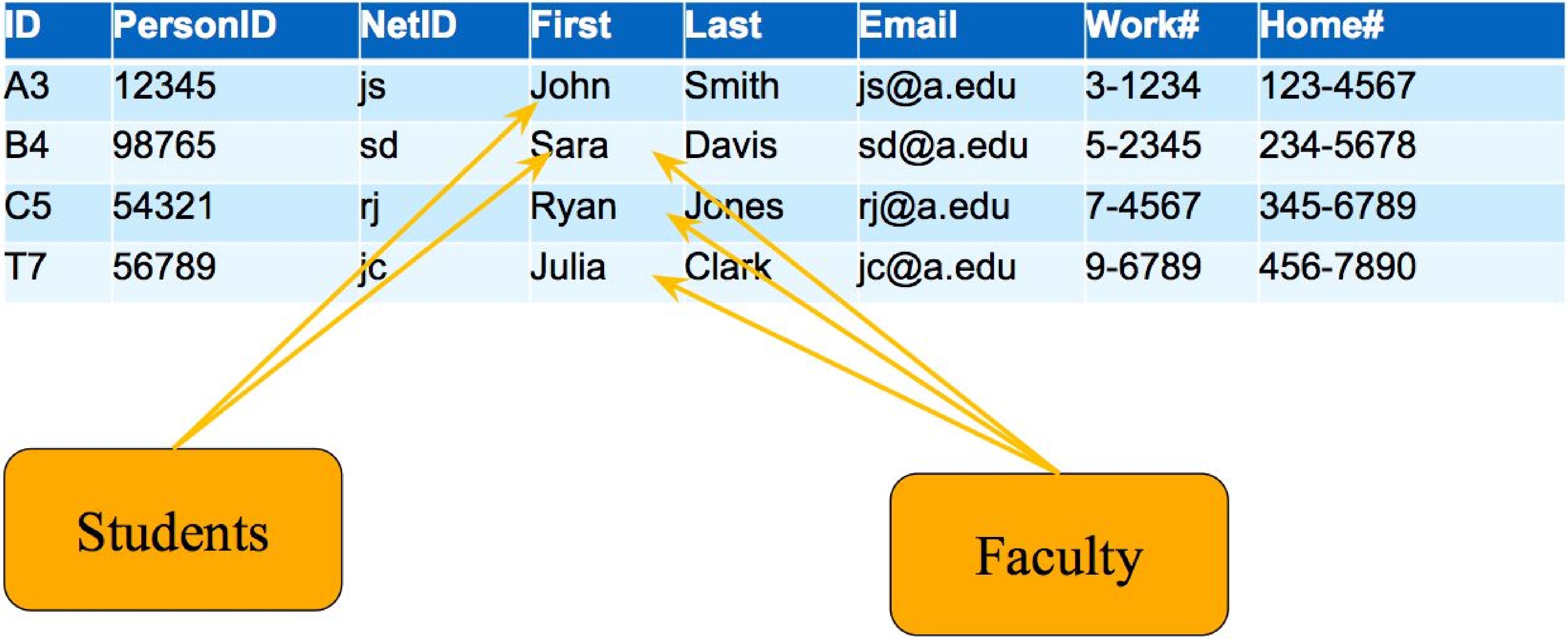
Filter or assign permissions

Permission type: *	Role
Permission definition:	apps:portal:permissions:UP_ERROR_CHAN:errorChanPermDef
Permission resource:	
Role:	Applications:uPortal:Roles:Portal Developers
Action:	
Enabled / disabled:	Enabled only

Permission assignments

		Actions	
Permission role	Resource	VIEW	Permission definition
Portal Developers	DETAILS	☑ ✓	errorChanPermDef

Database table rows and columns (i.e. target resource/object)

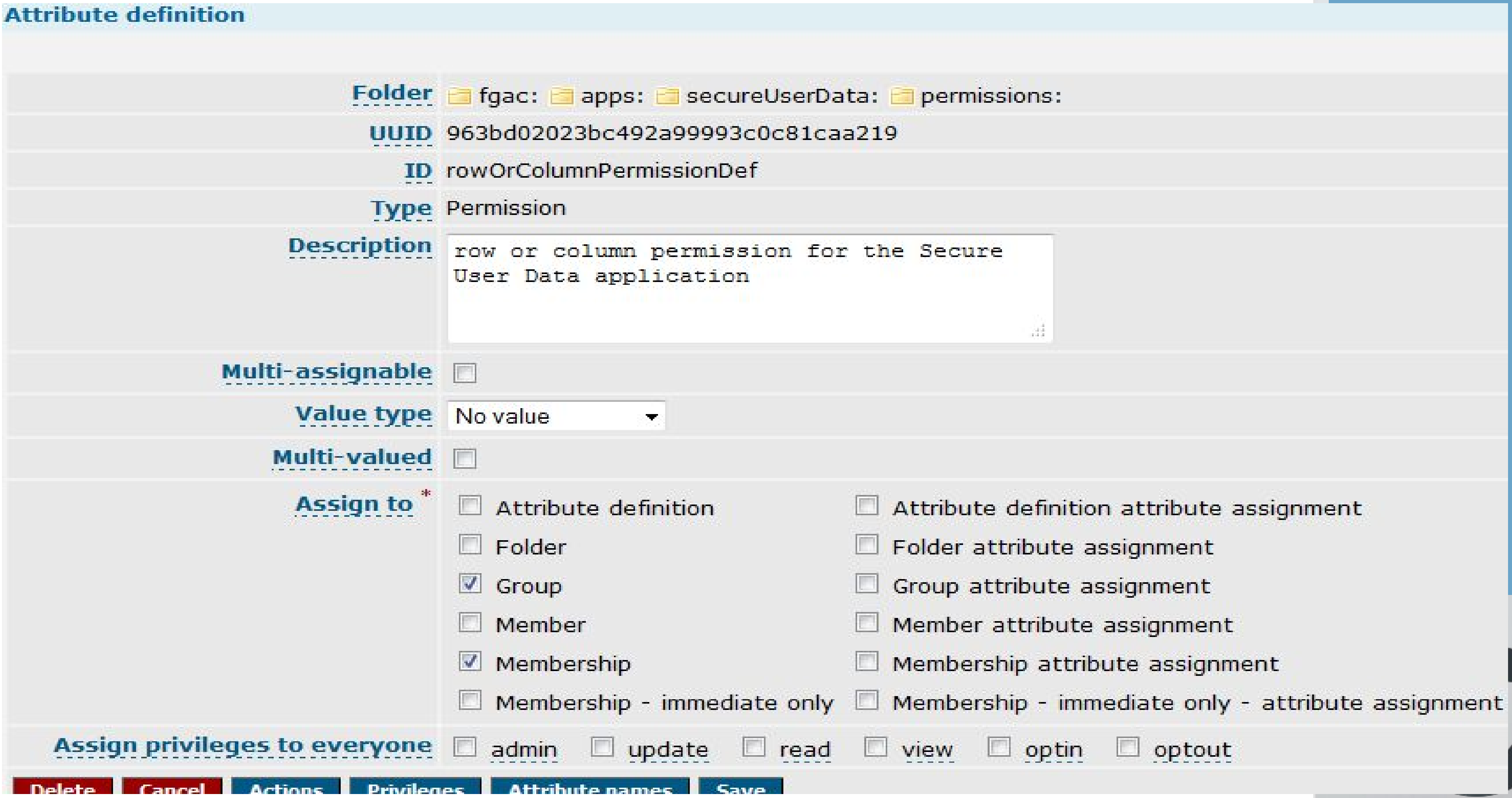


ID	PersonID	NetID	First	Last	Email	Work#	Home#
A3	12345	js	John	Smith	js@a.edu	3-1234	123-4567
B4	98765	sd	Sara	Davis	sd@a.edu	5-2345	234-5678
C5	54321	rj	Ryan	Jones	rj@a.edu	7-4567	345-6789
T7	56789	jc	Julia	Clark	jc@a.edu	9-6789	456-7890

Students

Faculty

Permission definition has configuration and security



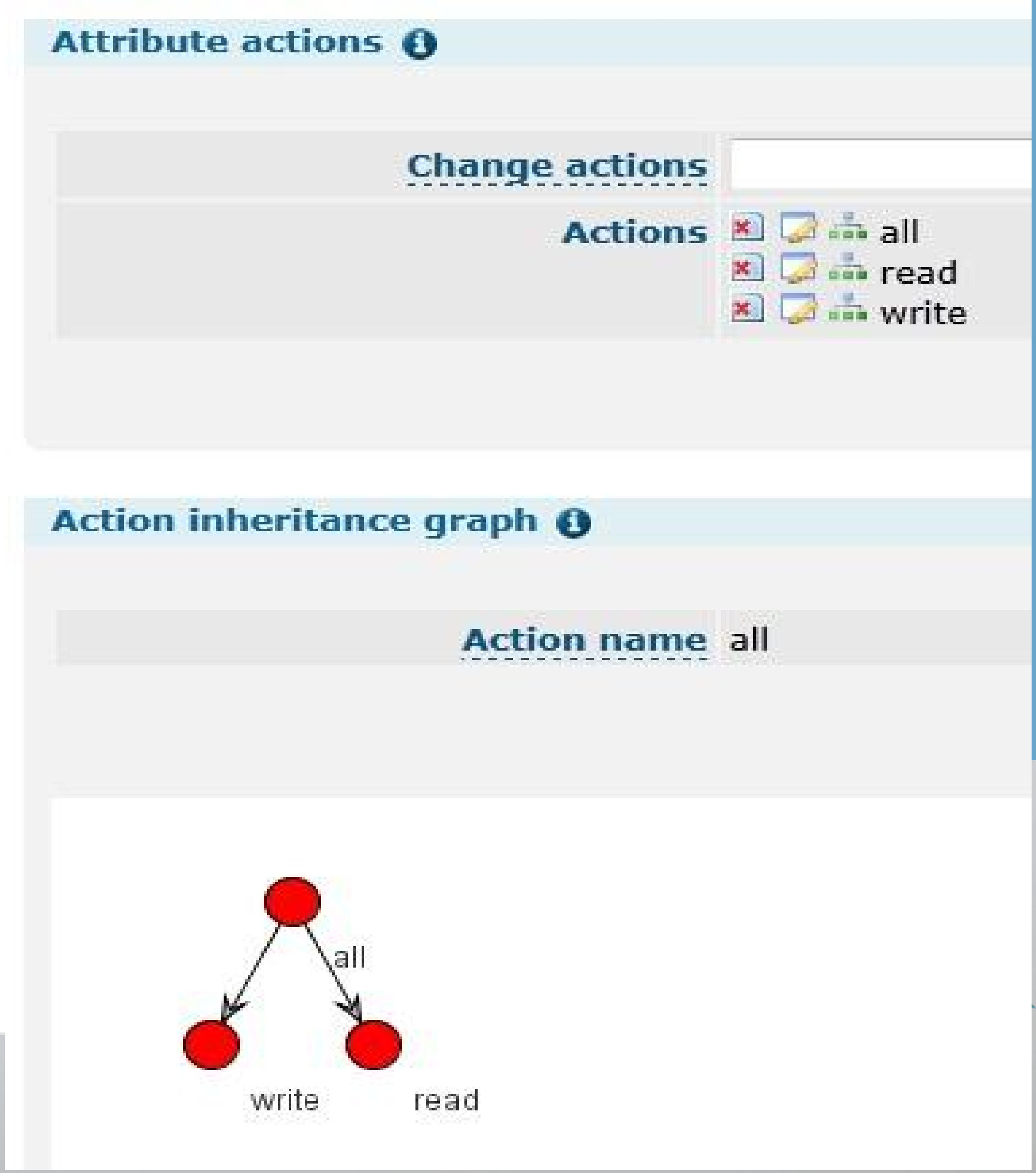
Attribute definition

Folder	fgac: apps: secureUserData: permissions:
UUID	963bd02023bc492a99993c0c81caa219
ID	rowOrColumnPermissionDef
Type	Permission
Description	row or column permission for the Secure User Data application
Multi-assignable	□
Value type	No value
Multi-valued	□
Assign to *	□ Attribute definition □ Folder ☑ Group □ Member ☑ Membership □ Membership - immediate only □ Attribute definition attribute assignment □ Folder attribute assignment □ Group attribute assignment □ Member attribute assignment □ Membership attribute assignment □ Membership - immediate only - attribute assignment
Assign privileges to everyone	□ admin □ update □ read □ view □ optin □ optout

Read/write action for this permission def

Include an “all” which implies read and write

Note: this is specific to this one permission definition, and does not affect other permissions in Grouper



Attribute actions

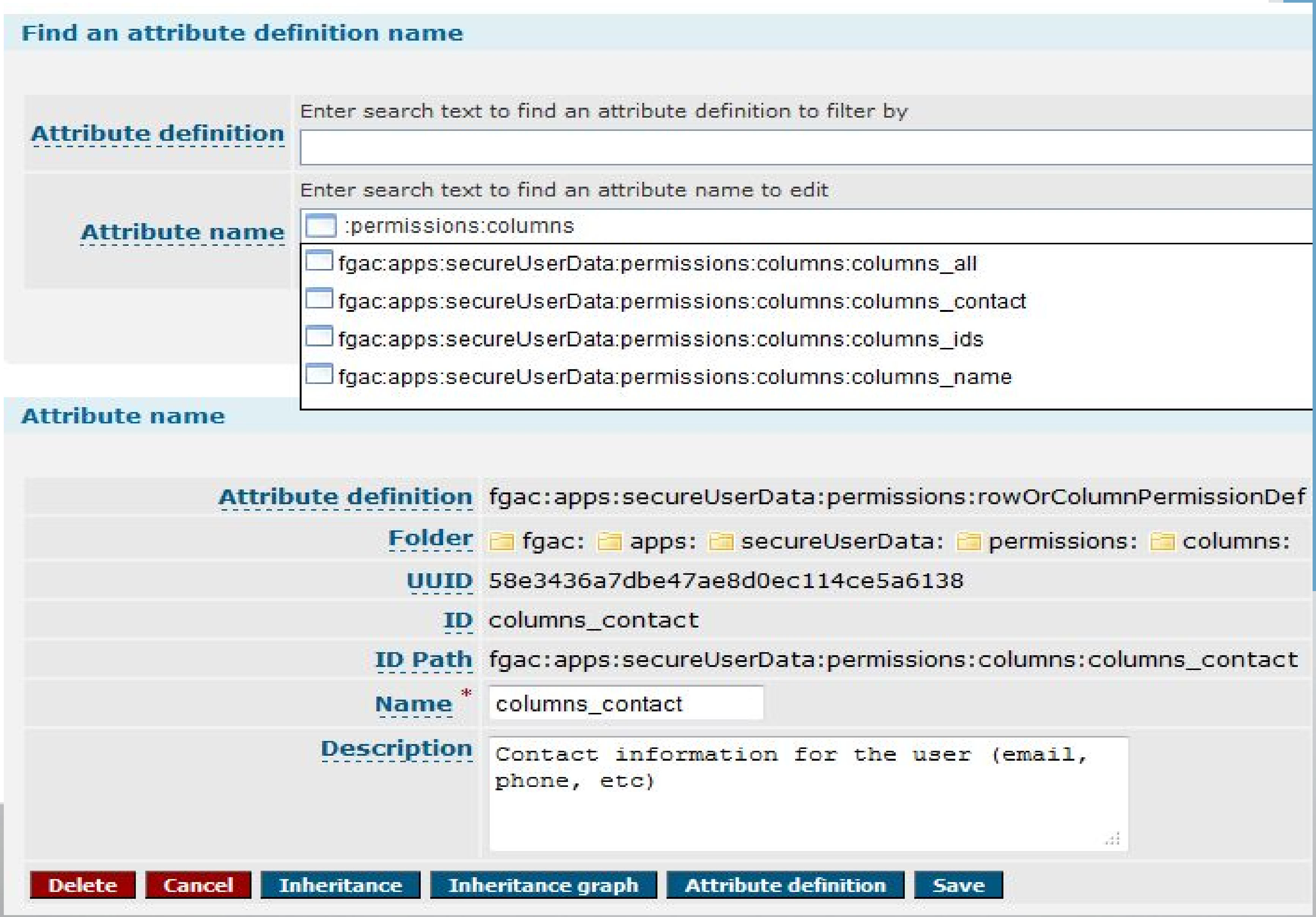
Change actions

Actions ☒ all
☒ read
☒ write

Action inheritance graph

Action name all

Resource/Object name for each set of columns



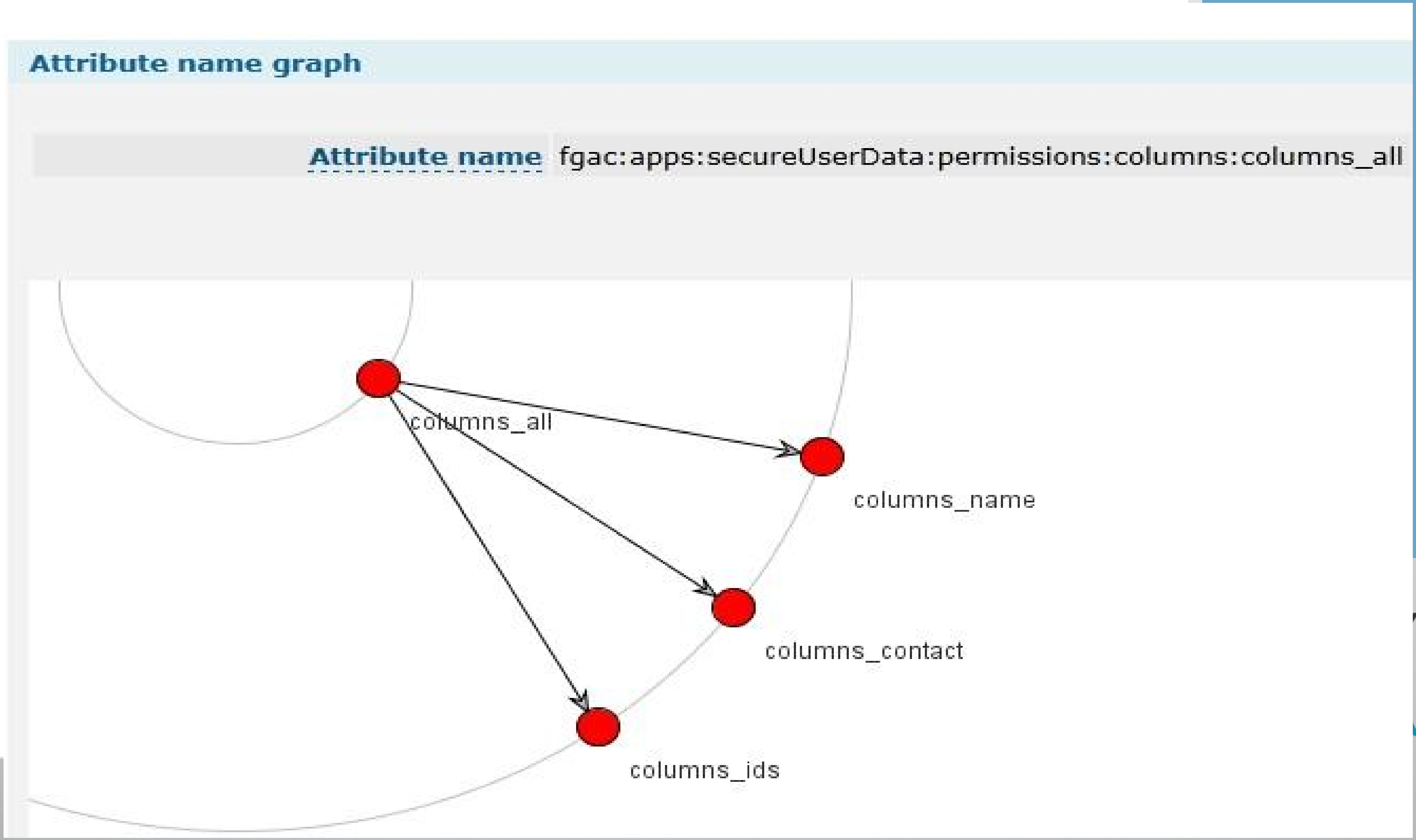
Find an attribute definition name

Attribute definition	Enter search text to find an attribute definition to filter by
Attribute name	Enter search text to find an attribute name to edit :permissions:columns fgac:apps:secureUserData:permissions:columns:columns_all fgac:apps:secureUserData:permissions:columns:columns_contact fgac:apps:secureUserData:permissions:columns:columns_ids fgac:apps:secureUserData:permissions:columns:columns_name

Attribute name

Attribute definition	fgac:apps:secureUserData:permissions:rowOrColumnPermissionDef
Folder	fgac: apps: secureUserData: permissions: columns:
UUID	58e3436a7dbe47ae8d0ec114ce5a6138
ID	columns_contact
ID Path	fgac:apps:secureUserData:permissions:columns:columns_contact
Name *	columns_contact
Description	Contact information for the user (email, phone, etc)

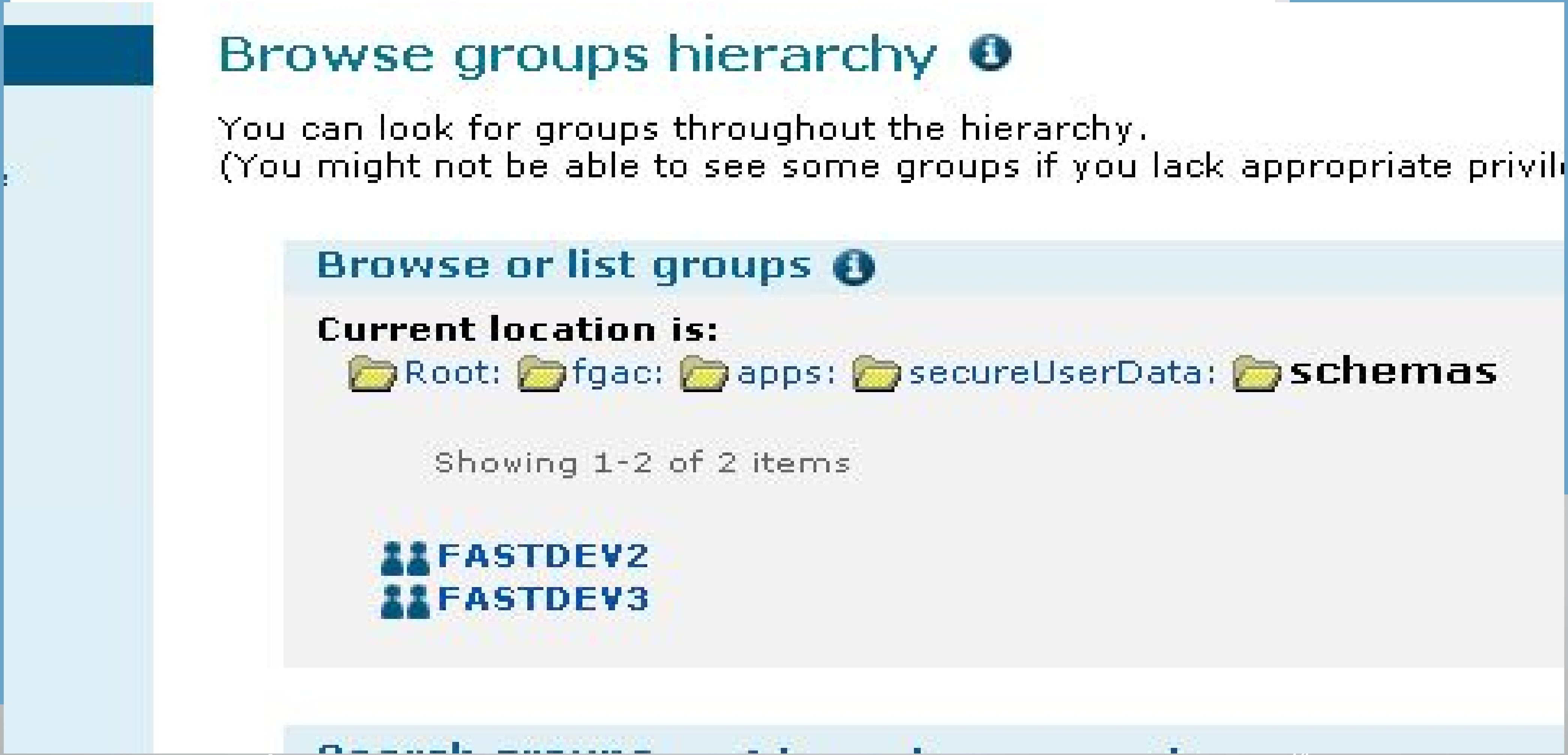
Column resource inheritance



Attribute name graph

Attribute name fgac:apps:secureUserData:permissions:columns:columns_all

Subjects will get connect as a specific database schema.



Browse groups hierarchy

You can look for groups throughout the hierarchy.
(You might not be able to see some groups if you lack appropriate privil

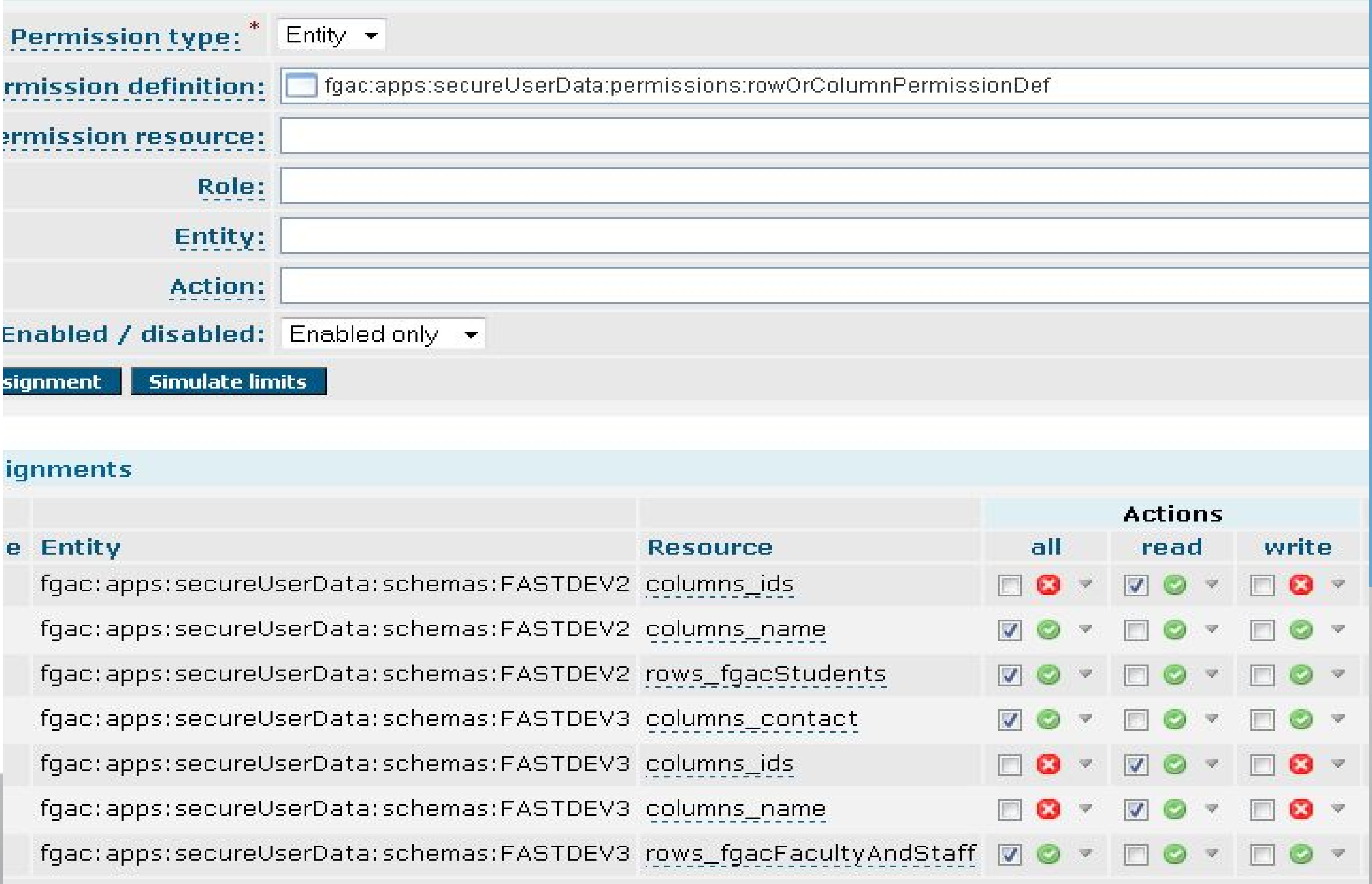
Browse or list groups

Current location is:
Root: fgac: apps: secureUserData: **schemas**

Showing 1-2 of 2 items

FASTDEV2
FASTDEV3

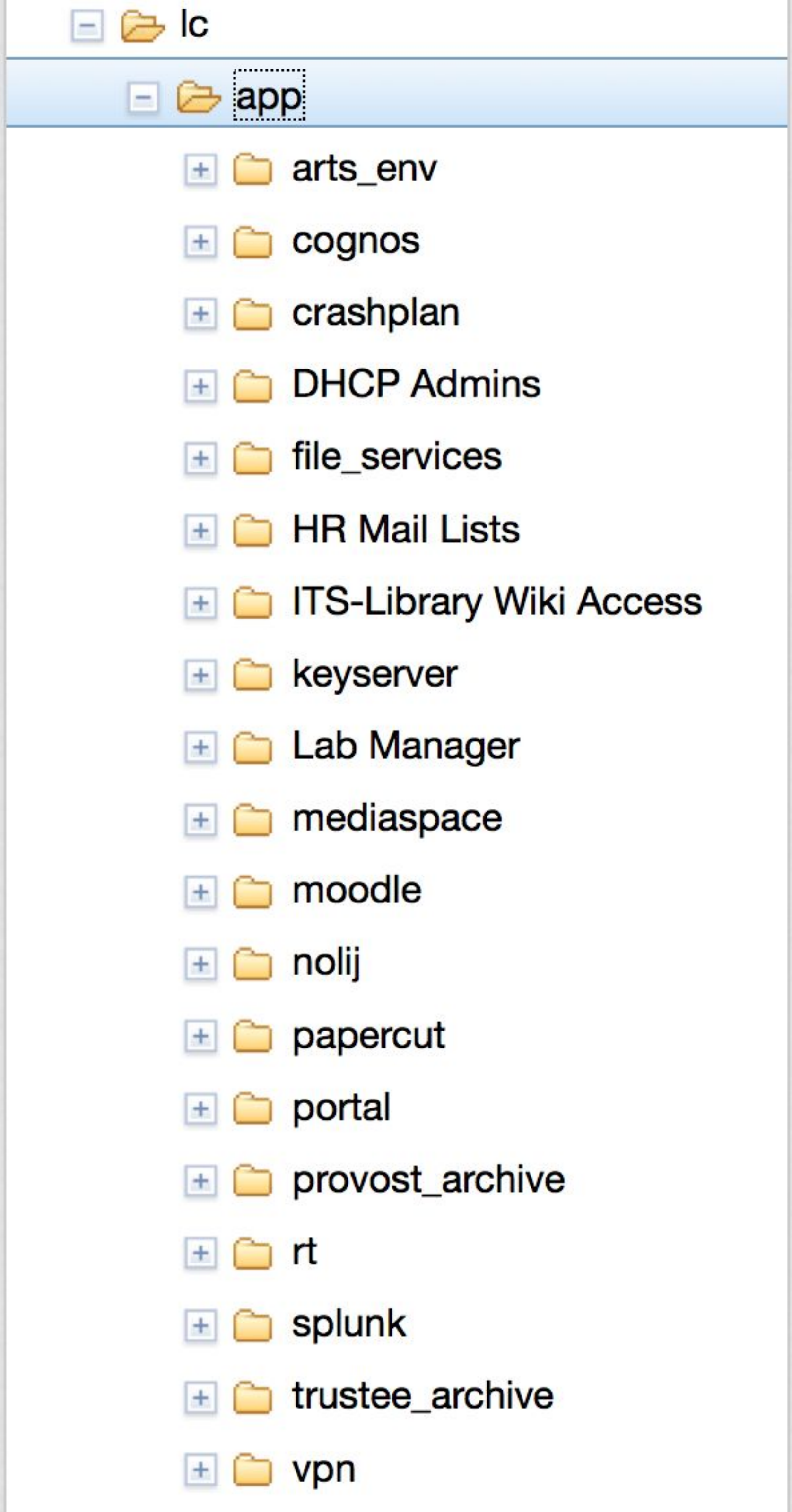
Assign the permissions



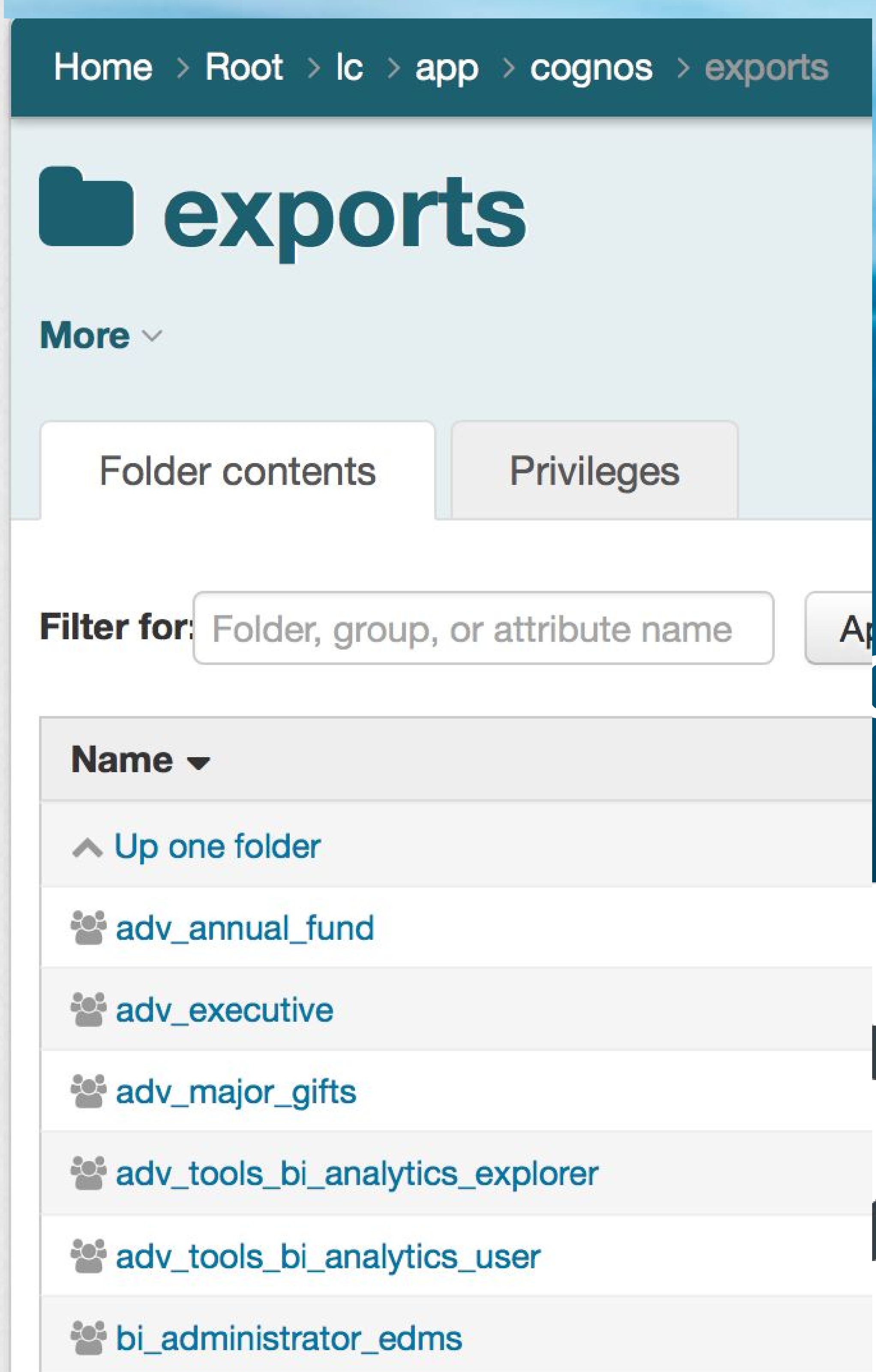
Permission type: *	Entity ▾
rmission definition:	fgac:apps:secureUserData:permissions:rowOrColumnPermissionDef
ermission resource:	
Role:	
Entity:	
Action:	
Enabled / disabled:	Enabled only ▾

ignments

			Actions		
e	Entity	Resource	all	read	write
	fgac:apps:secureUserData:schemas:FASTDEV2	columns_ids	☐ ✗ ▾	☑ ✓ ▾	☐ ✗ ▾
	fgac:apps:secureUserData:schemas:FASTDEV2	columns_name	☑ ✓ ▾	☐ ✓ ▾	☐ ✓ ▾
	fgac:apps:secureUserData:schemas:FASTDEV2	rows_fgacStudents	☑ ✓ ▾	☐ ✓ ▾	☐ ✓ ▾
	fgac:apps:secureUserData:schemas:FASTDEV3	columns_contact	☑ ✓ ▾	☐ ✓ ▾	☐ ✓ ▾
	fgac:apps:secureUserData:schemas:FASTDEV3	columns_ids	☐ ✗ ▾	☑ ✓ ▾	☐ ✗ ▾
	fgac:apps:secureUserData:schemas:FASTDEV3	columns_name	☐ ✗ ▾	☑ ✓ ▾	☐ ✗ ▾
	fgac:apps:secureUserData:schemas:FASTDEV3	rows_fgacFacultyAndStaff	☑ ✓ ▾	☐ ✓ ▾	☐ ✓ ▾



- lc
  - app
    - arts_env
    - cognos
    - crashplan
    - DHCP Admins
    - file_services
    - HR Mail Lists
    - ITS-Library Wiki Access
    - keyserver
    - Lab Manager
    - mediaspace
    - moodle
    - nolij
    - papercut
    - portal
    - provost_archive
    - rt
    - splunk
    - trustee_archive
    - vpn



Home › Root › lc › app › cognos › exports

exports

Filter for: Folder, group, or attribute name

Name ▾
^ Up one folder
adv_annual_fund
adv_executive
adv_major_gifts
adv_tools_bi_analytics_explorer
adv_tools_bi_analytics_user
bi_administrator_edms

016

ducation

INCITS 494-2012 RBAC - Policy-Enhanced

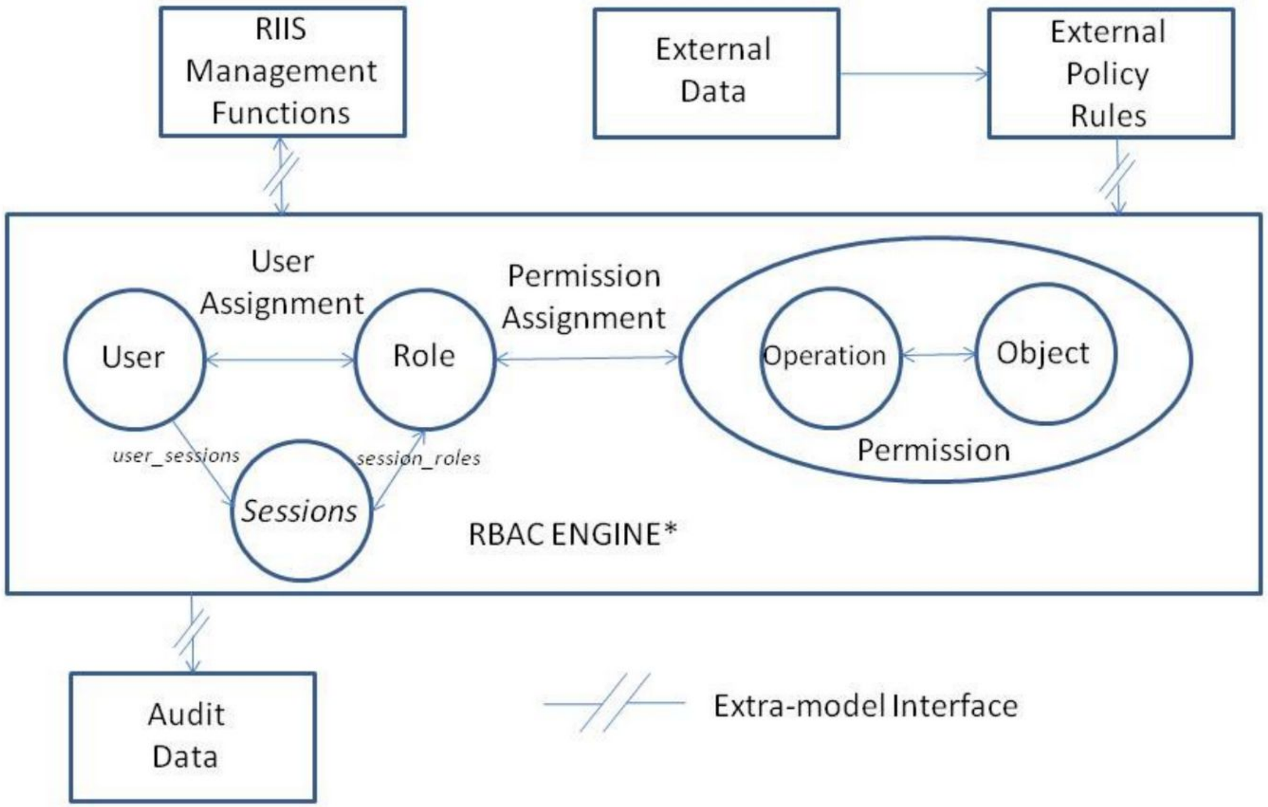
attribute: RBAC session attributes as used in this document are a characteristic of a subject, resource, action, or environment that may be referenced in a predicate or target.

constraint: A constraint is a relation among role features that acts as a restriction. This standard describes both static constraints (administratively controlled) and dynamic constraints (run time)

external policy rules: Imported constraints and data values for use in making role-base access control decisions.

INCITS 494-2012 RBAC - Policy-Enhanced

All constraints are determined only by External Policy Rules and External Data



*The RBAC engine controls all access control decisions

Figure 1 – RBAC Policy-Enhanced Reference Model

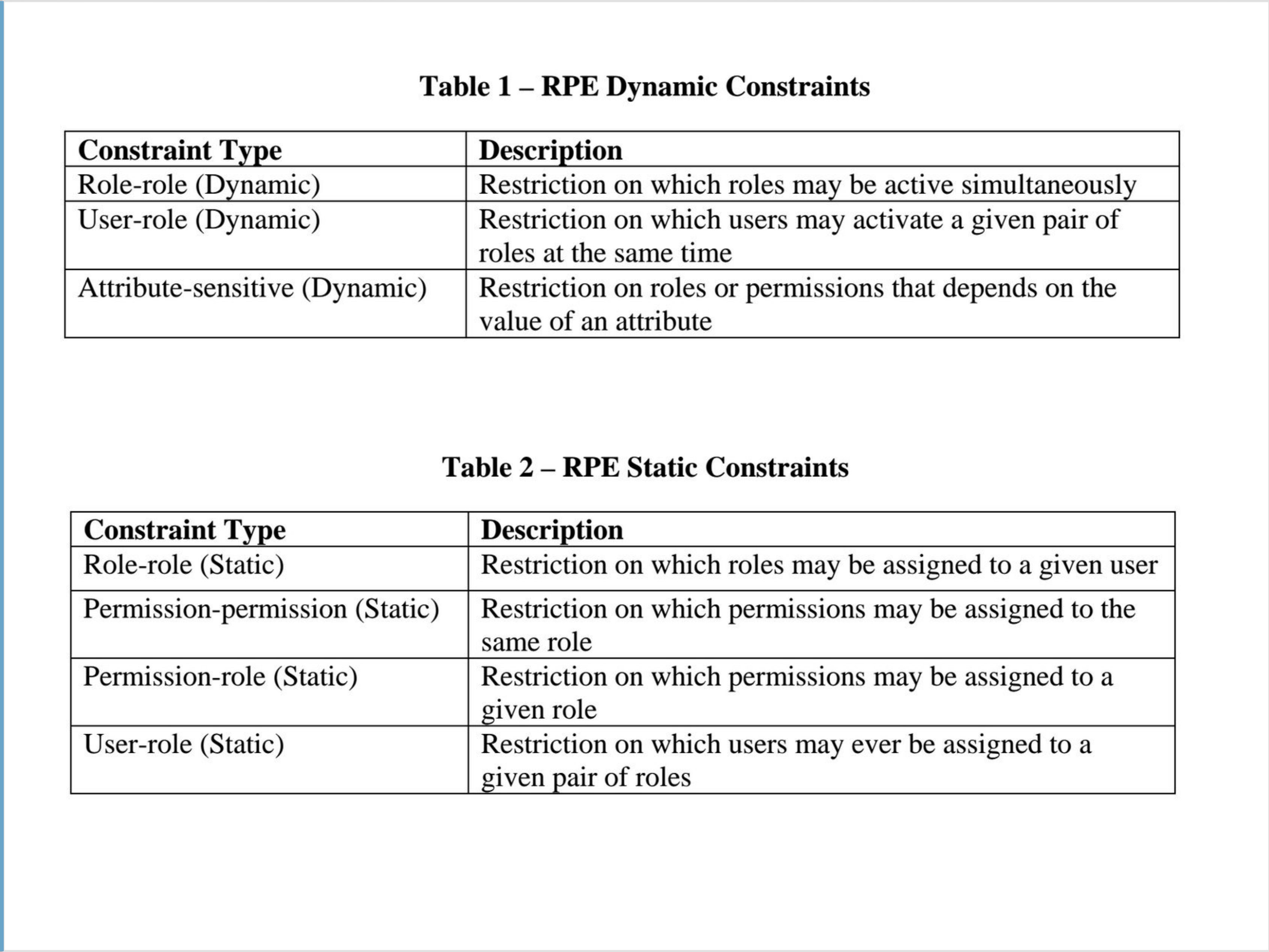


Table 1 – RPE Dynamic Constraints

Constraint Type	**Description**
Role-role (Dynamic)	Restriction on which roles may be active simultaneously
User-role (Dynamic)	Restriction on which users may activate a given pair of roles at the same time
Attribute-sensitive (Dynamic)	Restriction on roles or permissions that depends on the value of an attribute

Table 2 – RPE Static Constraints

Constraint Type	**Description**
Role-role (Static)	Restriction on which roles may be assigned to a given user
Permission-permission (Static)	Restriction on which permissions may be assigned to the same role
Permission-role (Static)	Restriction on which permissions may be assigned to a given role
User-role (Static)	Restriction on which users may ever be assigned to a given pair of roles

Grouper RPE

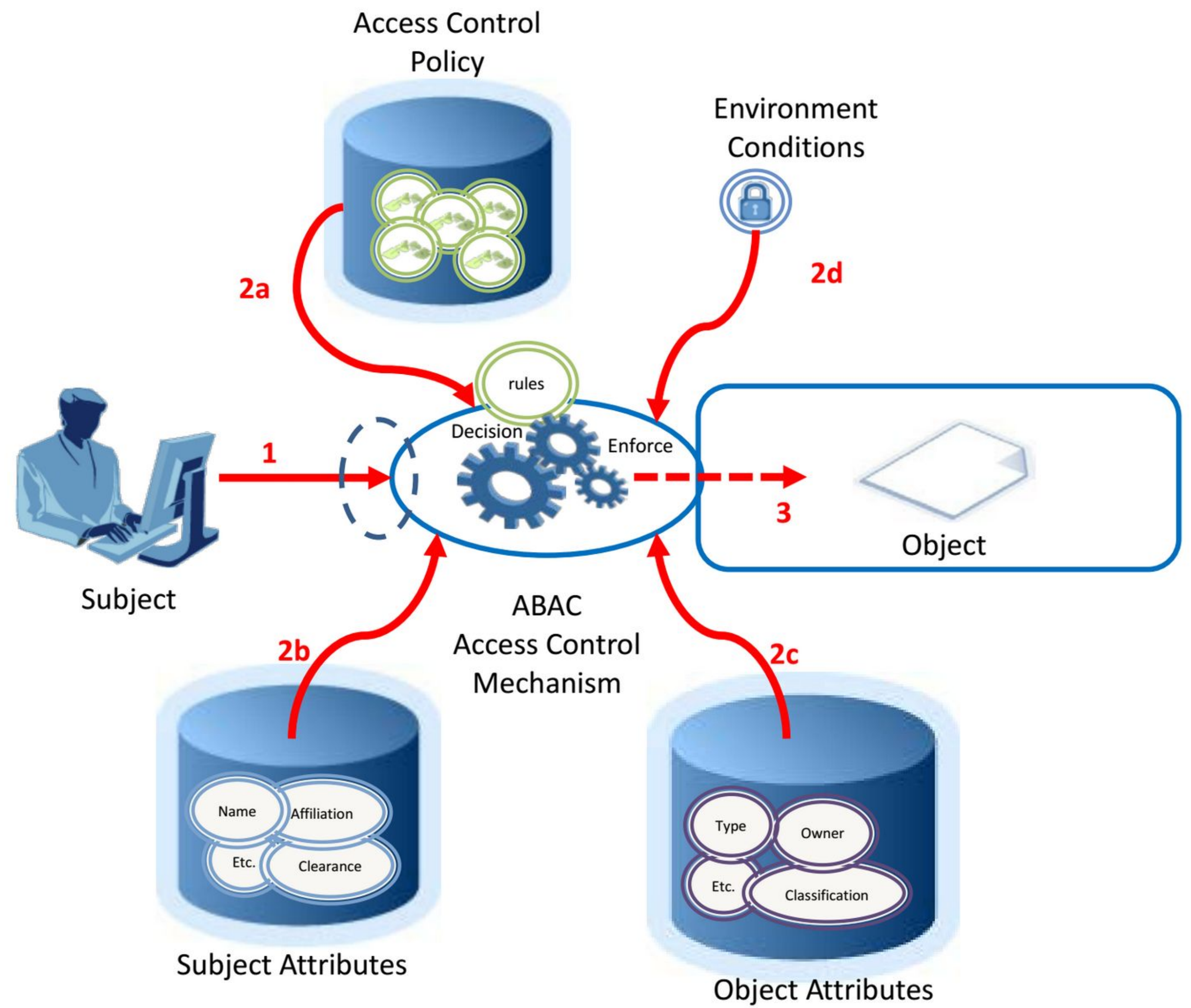
Grouper Permission Limit Built-In Implementations are:

- Weekday 9 to 5 limit
- Amount less than limit
- Amount less than or equal limit
- Labels contain limit
- IP address on networks limit
- IP address on network realm limit
- Expression language (EL) limit

NIST Special Publication 800-162 Guide to Attribute Based Access Control Definition and Considerations

Attribute Based Access Control (ABAC): An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment attributes and conditions.

Access Control Mechanism (ACM): The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.


Access Control Policy
Environment Conditions
2a
2d
rules
Decision
Enforce
1
3
Object
Subject
ABAC Access Control Mechanism
2b
2c
Name
Affiliation
Etc.
Clearance
Type
Owner
Etc.
Classification
Subject Attributes
Object Attributes

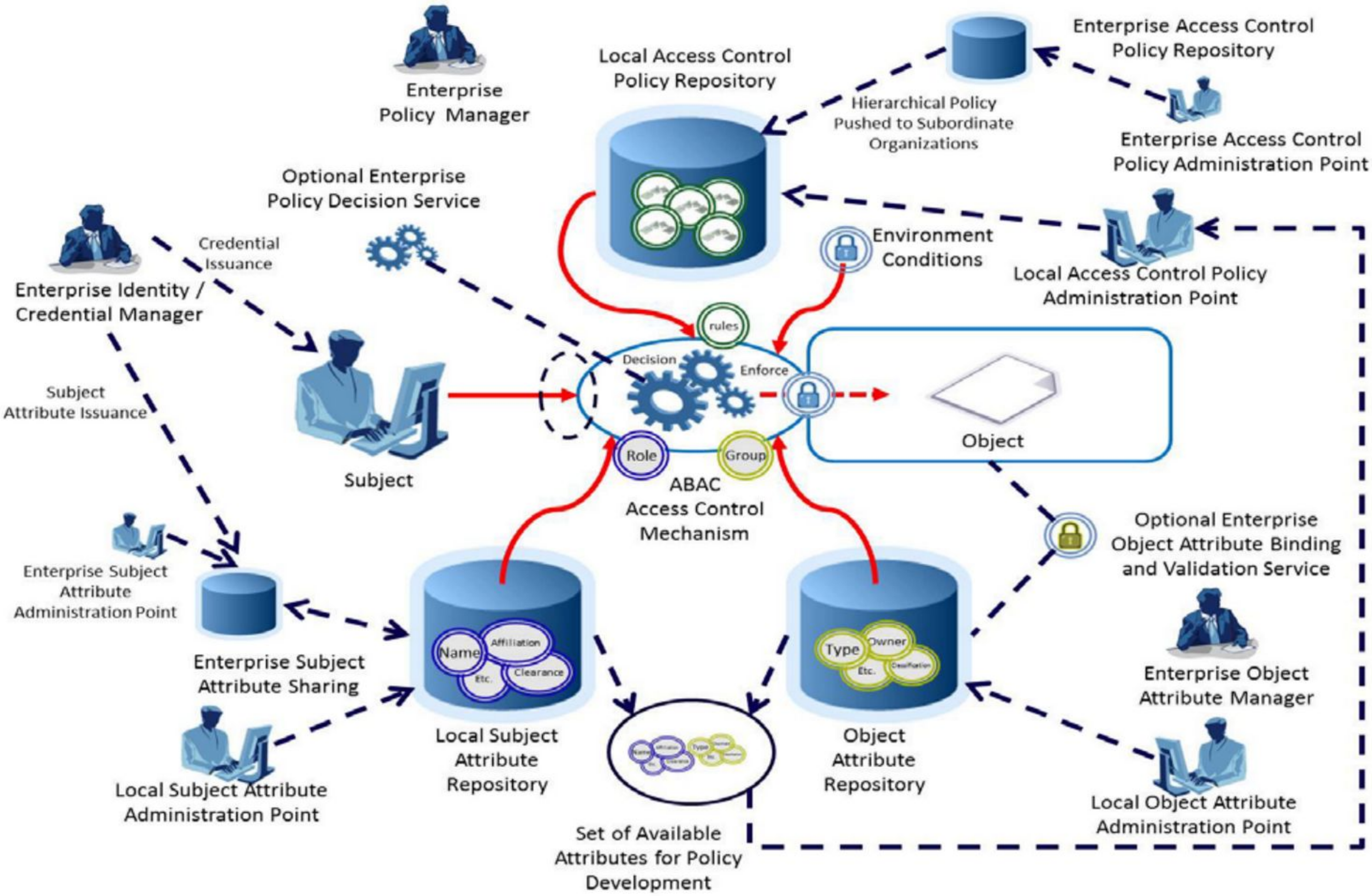
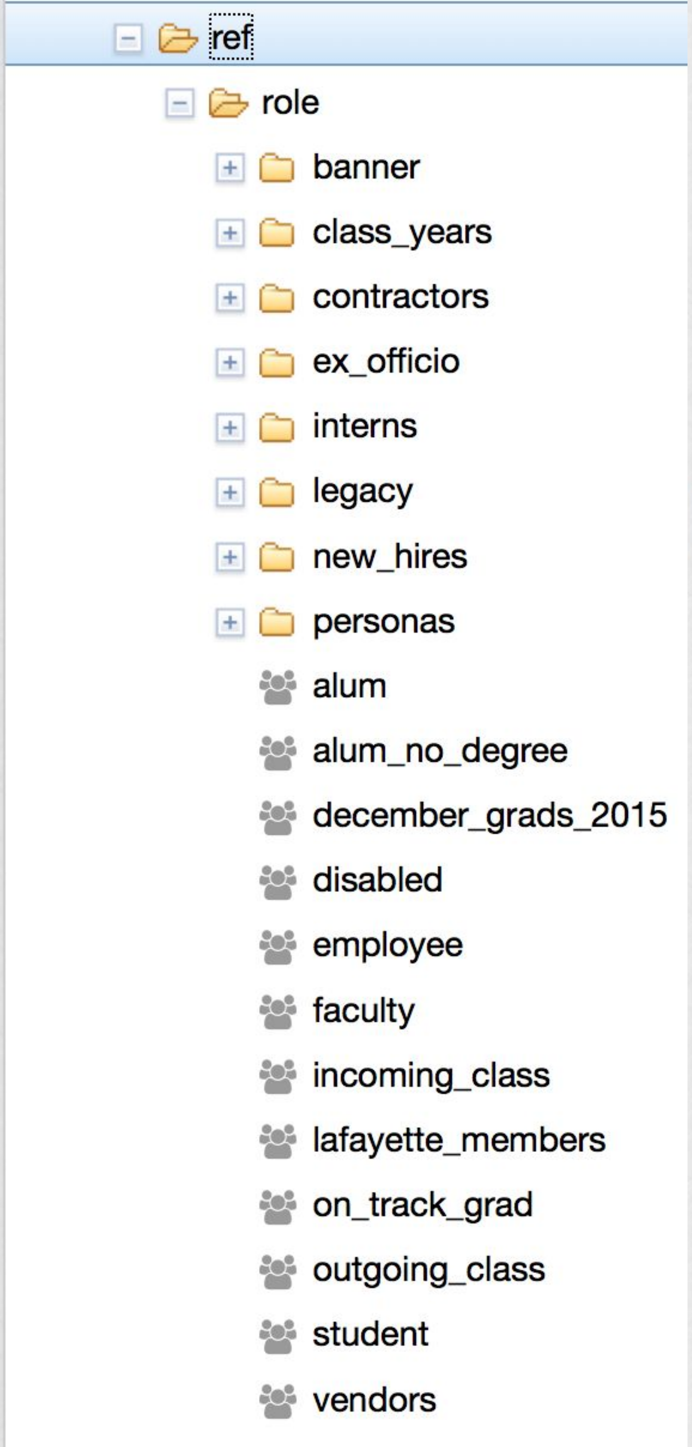
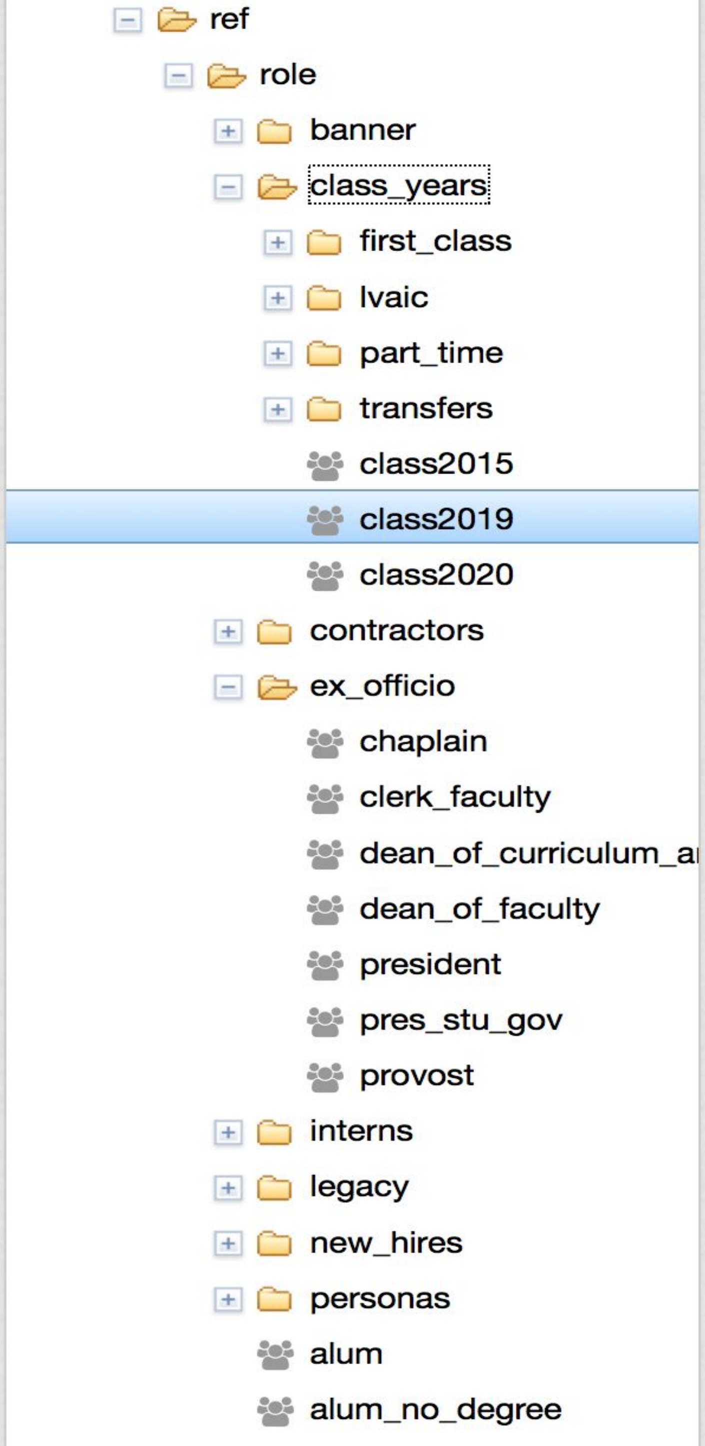

Enterprise Policy Manager
Local Access Control Policy Repository
Enterprise Access Control Policy Repository
Hierarchical Policy Pushed to Subordinate Organizations
Enterprise Access Control Policy Administration Point
Optional Enterprise Policy Decision Service
Credential Issuance
Enterprise Identity / Credential Manager
Environment Conditions
Local Access Control Policy Administration Point
rules
Decision
Enforce
Subject Attribute Issuance
Subject
Object
Role
Group
ABAC Access Control Mechanism
Optional Enterprise Object Attribute Binding and Validation Service
Enterprise Subject Attribute Administration Point
Enterprise Subject Attribute Sharing
Name
Affiliation
Etc.
Clearance
Type
Owner
Etc.
Classification
Enterprise Object Attribute Manager
Local Subject Attribute Administration Point
Local Subject Attribute Repository
Object Attribute Repository
Local Object Attribute Administration Point
Set of Available Attributes for Policy Development

Figure 4: Enterprise ABAC Scenario Example

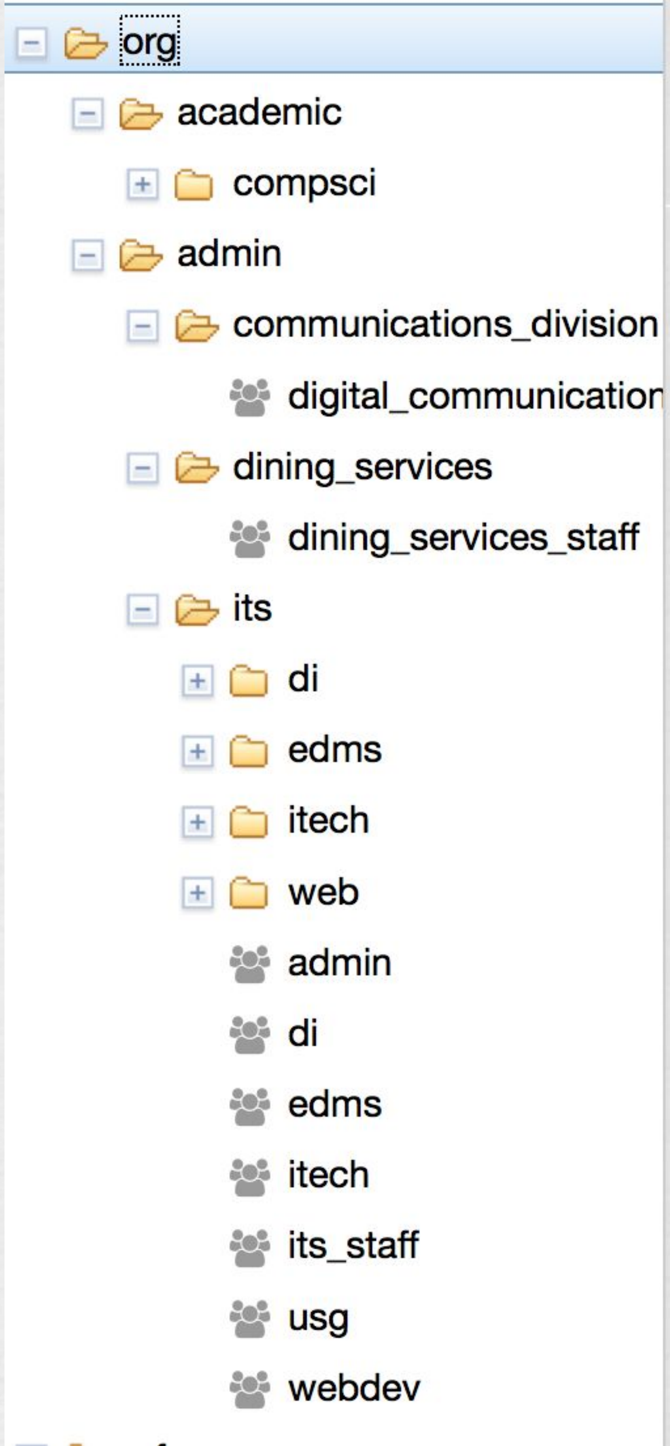
Reference Groups



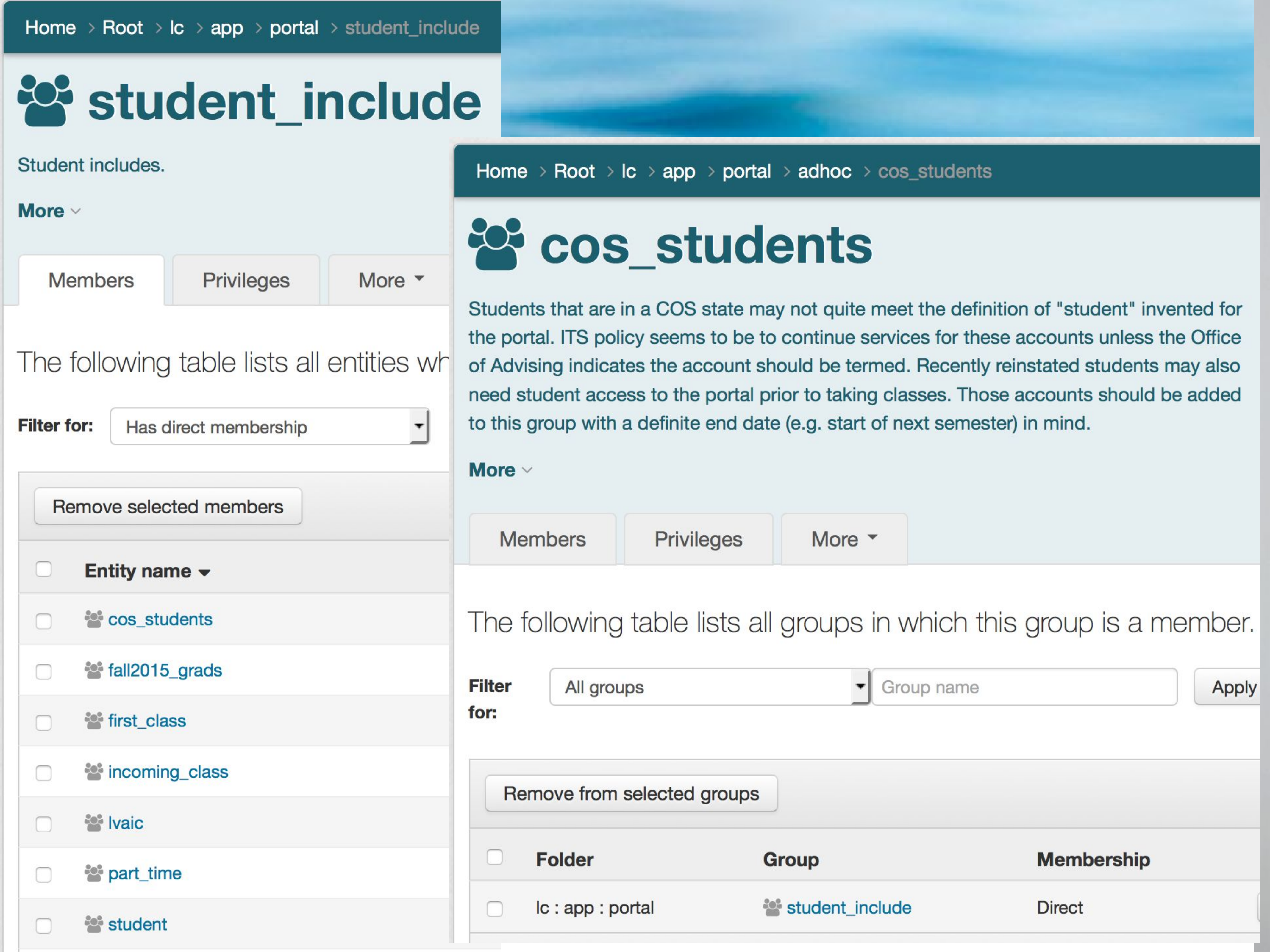
- ref
  - role
    - banner
    - class_years
    - contractors
    - ex_officio
    - interns
    - legacy
    - new_hires
    - personas
    - alum
    - alum_no_degree
    - december_grads_2015
    - disabled
    - employee
    - faculty
    - incoming_class
    - lafayette_members
    - on_track_grad
    - outgoing_class
    - student
    - vendors



- ref
  - role
    - banner
    - class_years
      - first_class
      - lvaic
      - part_time
      - transfers
      - class2015
      - class2019
      - class2020
    - contractors
    - ex_officio
      - chaplain
      - clerk_faculty
      - dean_of_curriculum_a
      - dean_of_faculty
      - president
      - pres_stu_gov
      - provost
    - interns
    - legacy
    - new_hires
    - personas
    - alum
    - alum_no_degree



- org
  - academic
    - compsci
  - admin
    - communications_division
      - digital_communication
    - dining_services
      - dining_services_staff
    - its
      - di
      - edms
      - itech
      - web
      - admin
      - di
      - edms
      - itech
      - its_staff
      - usg
      - webdev



Home › Root › lc › app › portal › student_include

student_include

Student includes.

The following table lists all entities wh

Filter for: Has direct membership

□	Entity name
□	cos_students
□	fall2015_grads
□	first_class
□	incoming_class
□	lvaic
□	part_time
□	student

Home › Root › lc › app › portal › adhoc › cos_students

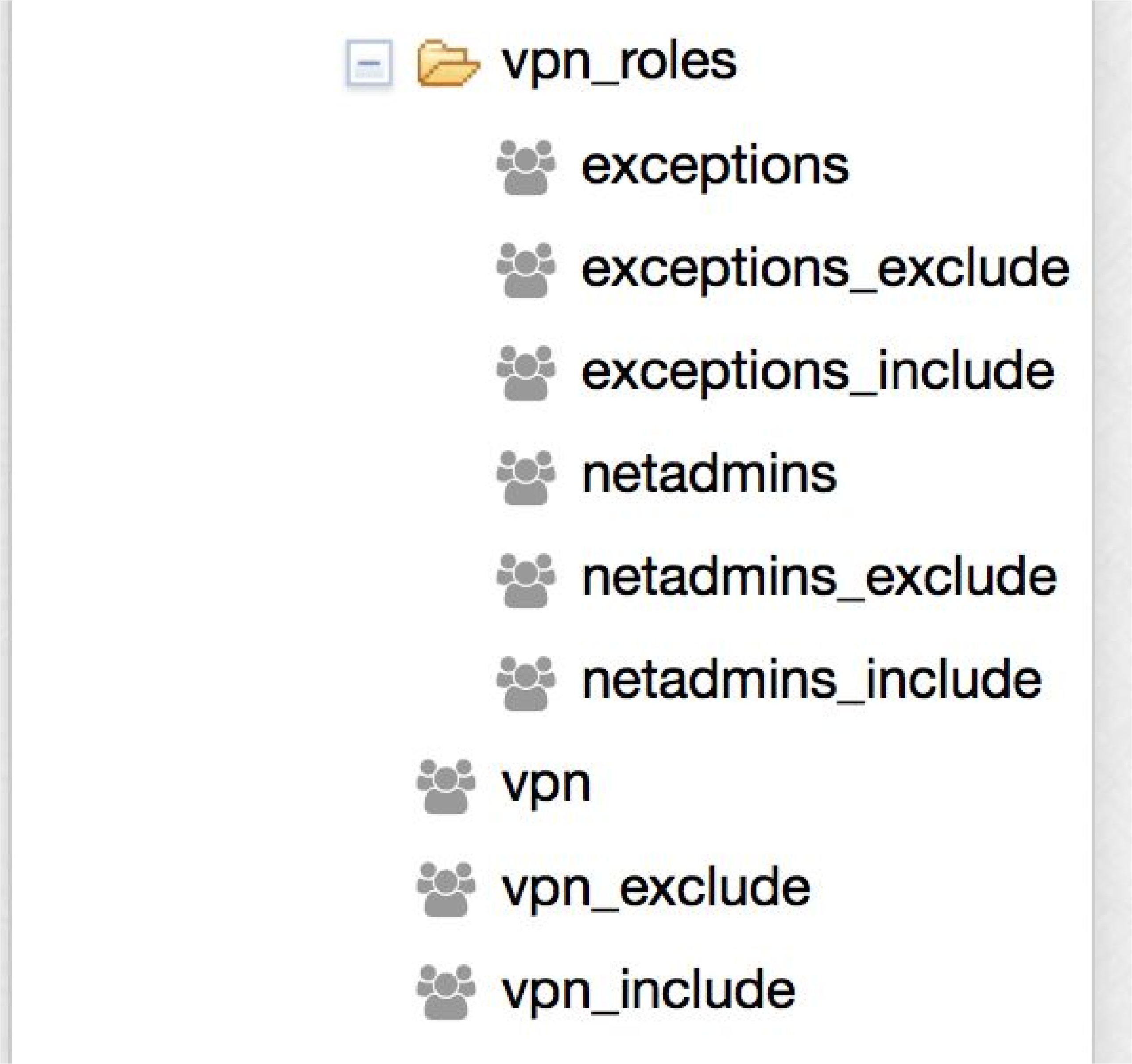
cos_students

Students that are in a COS state may not quite meet the definition of "student" invented for the portal. ITS policy seems to be to continue services for these accounts unless the Office of Advising indicates the account should be termed. Recently reinstated students may also need student access to the portal prior to taking classes. Those accounts should be added to this group with a definite end date (e.g. start of next semester) in mind.

The following table lists all groups in which this group is a member.

Filter for: All groups | Group name

□	Folder	Group	Membership
□	lc : app : portal	student_include	Direct



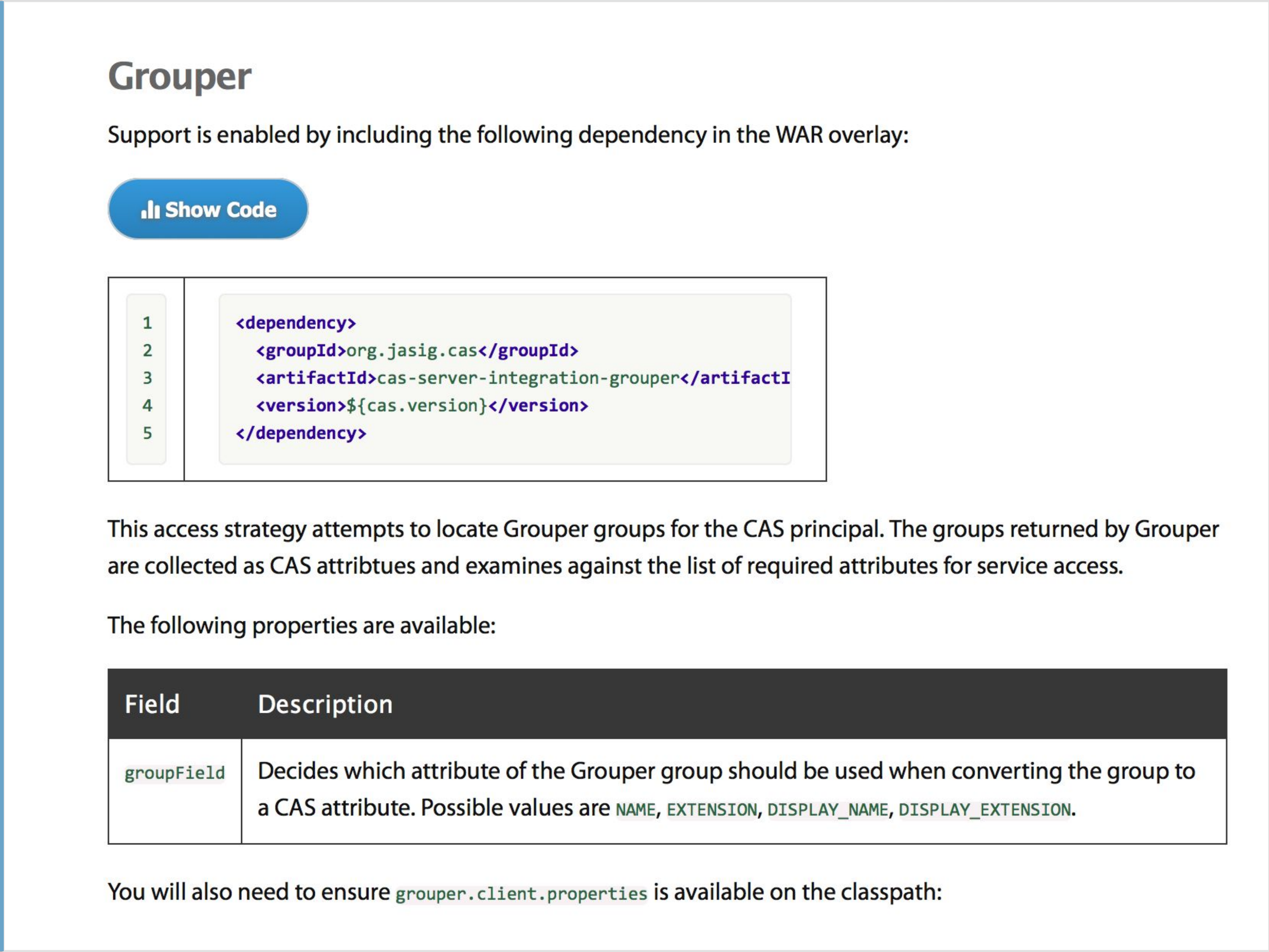
vpn_roles

- exceptions
- exceptions_exclude
- exceptions_include
- netadmins
- netadmins_exclude
- netadmins_include

vpn

vpn_exclude

vpn_include



Grouper

Support is enabled by including the following dependency in the WAR overlay:

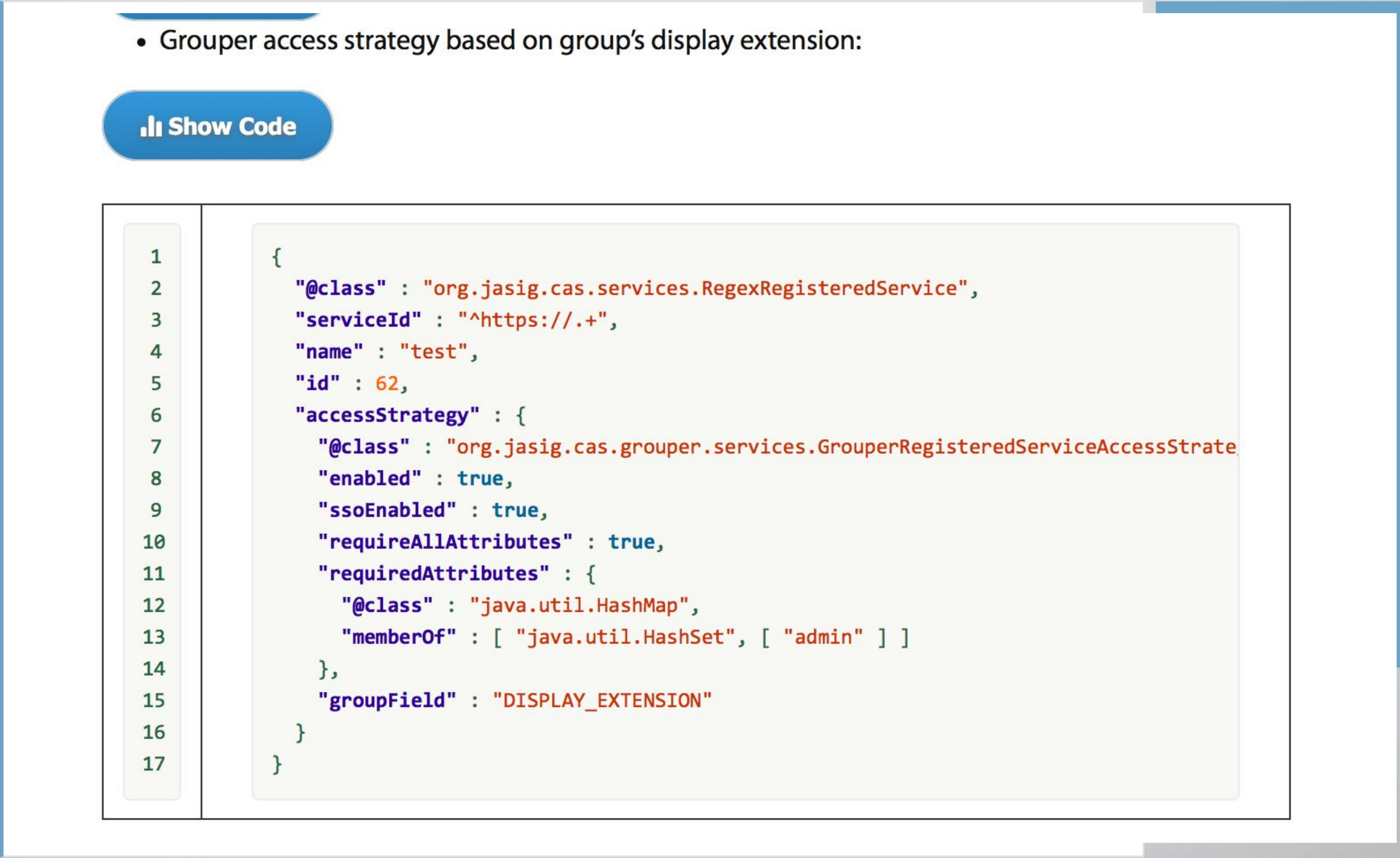
```
<dependency>
  <groupId>org.jasig.cas</groupId>
  <artifactId>cas-server-integration-grouper</artifactI
  <version>${cas.version}</version>
</dependency>
```

This access strategy attempts to locate Grouper groups for the CAS principal. The groups returned by Grouper are collected as CAS attribtues and examines against the list of required attributes for service access.

The following properties are available:

Field	Description
`groupField`	Decides which attribute of the Grouper group should be used when converting the group to a CAS attribute. Possible values are `NAME`, `EXTENSION`, `DISPLAY_NAME`, `DISPLAY_EXTENSION`.

You will also need to ensure `grouper.client.properties` is available on the classpath:



- Grouper access strategy based on group’s display extension:

```
{
  "@class" : "org.jasig.cas.services.RegexRegisteredService",
  "serviceId" : "^https://.+",
  "name" : "test",
  "id" : 62,
  "accessStrategy" : {
    "@class" : "org.jasig.cas.grouper.services.GrouperRegisteredServiceAccessStrate
    "enabled" : true,
    "ssoEnabled" : true,
    "requireAllAttributes" : true,
    "requiredAttributes" : {
      "@class" : "java.util.HashMap",
      "memberOf" : [ "java.util.HashSet", [ "admin" ] ]
    },
    "groupField" : "DISPLAY_EXTENSION"
  }
}
```

Policy Enforcement Points

https://github.com/UniconLabs/CASGrouperWebServicesWebApplication
ASP .NET web application with a custom implementation of a RoleProvider that uses Grouper Web Services to determine roles and permissions.

https://github.com/UniconLabs/cas-spring-security-grouper
A proof of concept Spring Security adapter implementation on top of Grouper data store

https://github.com/UniconLabs/cas-shiro-grouper
proof of concept Apache Shiro adapter implementation on top of Grouper data store