Folder and Group Design

Overview

> *"Just having a plan or standard has been quite helpful, as it allows implementers to get on with real work without having to stumble on how to name things or where to stick them."*
> *- Tom Barton, University of Chicago*

Once Grouper is initially deployed it is up to the Identity and Access Management (IAM) analyst to construct and organize the appropriate folders and groups necessary to achieve the desired access management capabilities. The folder and group design provides institutional-level and application-specific group definition and management, and supports the campus-wide scope of the service. Such a plan enables organized service growth and promotes effective reuse of common objects.

This section first defines a variety of group types and purposes and then describes a recommended initial folder and group organization.

An efficient way to set up groups that adhere to the recommended structure is using the Grouper Template Wizard.

Group Definitions

Basis Groups

Often the best source of data for building meaningful cohorts is a combination of arcane codes representing various types and states of employees or students, and often sourced from multiple systems. To leverage the power of Grouper, these groups should be brought in as “raw” **basis groups**.

Basis groups are used by the IAM analyst to construct the cohorts that are required for access policy. Access policy does not use basis groups directly, rather the basis groups are used to build up reference groups. This indirection provides the IAM analyst the ability to adjust to changing source systems and business practices while keeping reference groups and access policy relatively stable. Basis groups are typically only visible to the IAM analyst, and would not normally be reflected out to applications and directories.

Reference Groups

Reference groups tend to be organized in particular folder locations for convenience and ease of use, but what makes a group a reference group is not its name or folder location, but rather its intended use, definition and scope, and data management expectations.

A **reference group** is a set of subjects that is largely intended to be used by reference within access policy. **Reference groups can be thought of as labels or tags that identify meaningful cohorts**. In this way, they can also be viewed as subject attributes from an ABAC perspective. Access policies often require cohorts organized via institutional affiliation (faculty, staff, student), a particular office or department (president’s office, finance division, chaplain), program (chemistry students), and even residence or class year. All of these are good examples of reference groups.

Reference groups represent the best possible source of “truth” about any particular subject at a given time for the purposes of access control. Therefore, the rules that define the various cohorts must be well understood and known. Reference groups may have institutional scope (e.g. student, faculty, staff) where the definition is expected to apply globally. Reference groups can also have application or organizational scope in which case the definition only applies to limited set of applications or policy definitions.

Reference groups are intended to support effective and efficient day-to-day operations by providing timely, accurate groups representing various cohorts required for access control and collaboration. Data for placing subjects into a particular cohort is often available in source systems or operational data stores. However, in cases where a source system is not available an authoritative office may be responsible for maintaining membership directly via the Grouper UI.

Ideally, manually managed reference groups should only be for small cohorts that lack sufficient institutional data. If you find yourself manually managing large reference groups, look for good sources of data for a loader job or other basis groups. Sources of data, timeliness of updates, reliability, and administrative access control are expected to be well known since they will directly affect access to a wide variety of services and resources.

When viewing a group in the Grouper UI, under the “More tab” click “this group’s membership in other groups” to show where the reference group is used in access policy.

Access Policy Groups

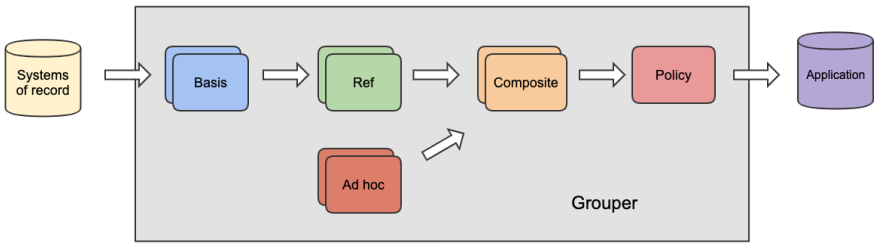
Access to services is often controlled by membership in a particular group or by having a certain subject attribute value (e.g. eduPersonEntitlement). This is sometimes called coarse-graining access control to distinguish it from the fine-grained permission management that is often found in RBAC like systems. Access policy groups can be used to deny or allow access to a resource (e.g. access to VPN) based on policy or to place a subject in an application specific role.

ABAC natural language policy, that is access policy stated in common language, must be converted to digital policy for any access control mechanism to effectively operate. Digital policy is manifest in Grouper via access policy groups. Subject membership in an access policy group must be indirect and represents a precomputed access policy decision based on subject attributes (i.e. the subject's membership in various reference groups).

An **access policy group** is a composite group whose membership is composed of an **include group** (i.e. the **allow** group) and an **exclude group** (i.e. the **deny** group). Subject membership in both the allow group and the deny group should be indirect (i.e. through reference groups) and have a clear mapping to the natural language policy. When exceptions to policy are necessary, locally scoped reference groups should be added.

It is good practice to **limit access policy groups to indirect membership assignments via reference groups**. This ensures that as subject attributes change, effective membership are kept up to date and access control decisions are correct. It also enables the direct mapping from natural language policy to digital policy and vice versa. Individual exceptions to policy, while not expressly recommended, can be accommodated by adding subjects directly to the allow/deny groups.

Membership in an access policy group is often kept in sync directly with a target service or an intermediary like an LDAP-based enterprise directory service. Services can also query Grouper directly for membership assignment.



Account Policy Groups

Services almost always require some type of identity record to be maintained at the service as a first step in granting access. This could be as simple as a single identifier and a few subject attributes. Membership within an account policy group signals that a suitable identity record (i.e. an account) should be created and kept in sync at the target service.

An **account policy group** is a composite group whose membership is composed of a **single include group** (i.e. the allow group) and a **single exclude group** (i.e. the deny group). Effective membership in an account policy group represents a precomputed account policy decision.

Manual Groups

Groups designated as manual indicate that they are intended to be managed by a person. Examples of manual groups are:

1. Teams that cannot be derived from a query to another system
2. Includes or excludes for exceptions to a policy driven by automatically loaded groups
3. Fine grained assignments that do not follow a broad collaboration group structure

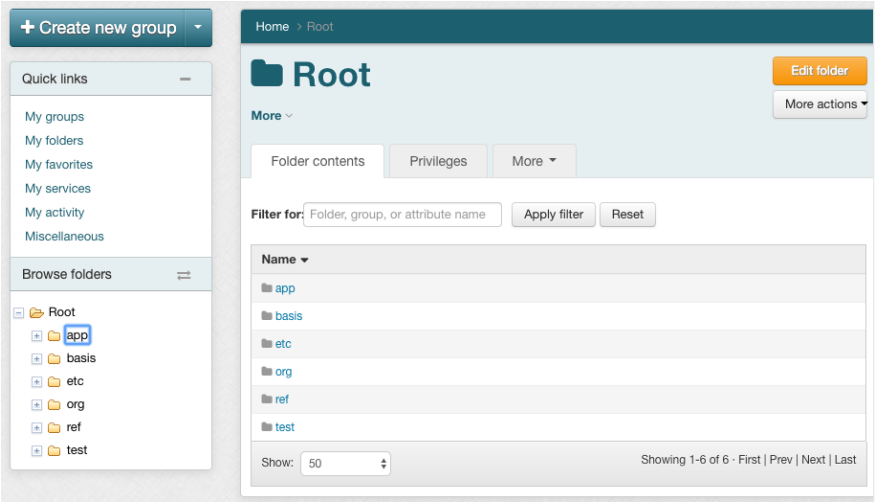
Intermediate Groups

Groups that do not fall in to another category might be "intermediate" groups. These are considered internal, or they exist to make a policy work but the average Grouper user does not need to be concerned with them. While building a policy out of composites, you might have some groups that are marked as "intermediate" to indicate that they are necessary but can be generally ignored.

Standard Folder Set and Pattern

The recommended standard folder set and pattern consists of **six folders** that are generally organized by group type, intended use, and visibility.

- `etc:` - Grouper configuration, administrative access control groups, and loader jobs. Note: loader jobs can be configured anywhere
- `basis:` - groups used exclusively by the IAM team to build reference groups
- `ref`: - reference groups (i.e. institutional meaningful cohorts) - "truth" about subjects
- `app:` - enterprise applications access control policies - specific policies for services
- `org:` - delegated authority, ad-hoc groups, org "owned" apps or reference groups
- `test:` - test folder for system verification, and group math sandbox



A main reason for a folder structure is to delegate access to sub-organizations and teams. Grouper privileges are inherited from folders. Also some groups belong together but it is unclear if they are a certain type or another. For example some people might consider a group to be a basis group, and others might think it is a reference group. "Institutional meaning" can be vague. Therefore you will have conflicting requirements when trying to follow the folder structure but also organize the groups so that they make sense and have inherited privileges. In Grouper you can identify folders or groups as a certain type so that groups can have a type but live in a non-standard location. i.e. most of your reference groups might be in the "ref" folder, but you will also have reference groups peppered throughout your Grouper registry. Pay attention when assigning types to folders since that will inherit to all groups in the folder. For example, a "policy" folder in an app might container non-policy intermediate groups. For this reason you should avoid assigning the "policy" type to a folder.

etc folder

The `etc:` folder holds various Grouper system specific configuration, loader jobs (though they can be configured anywhere), and the Grouper system access control groups.

- `etc:grouper_ui` - members can login and use the Grouper User Interface
- `etc:grouper_ws` - members can call and use the Grouper Web Services
- `etc:grouper_admin` - members have root-like privileges to the Grouper system
- `etc:loader:` - folder for various loader jobs. Note: loader jobs can be configured anywhere

> Tip: Initializing administration of privileges provides further details on setting up and configuring Grouper system access control groups.

basis folder

The `basis:` folder is strictly for use by the IAM team and provides a place to pull in and manage “raw” basis groups. **Basis groups are used to build up institutionally meaningful reference groups**. Basis groups provide a layer of abstraction between reference groups and source systems, and shields access policy from low level details of any particular source system. This layer provides the IAM analyst the flexibility to build appropriate reference groups and keep access policy groups focused on institutionally meaningful concepts. The folder structure under `basis:` should help the IAM analyst understand, find, and manage various basis groups. This folder might be further organized by source system.

`basis:hris:{employee_codes}` - types of employees, used to build reference groups

`basis:sis:{student_codes}` - types of students, used to build reference groups

ref folder

The `ref:` folder is for institutionally meaningful reference groups. These may be built from basis groups, inline loader jobs, or manual maintained with the Grouper UI. Reference groups are intended to be used in access policy and help provide a direct mapping to the natural language policy. The folder structure under ref: should help the access policy manager understand and find the appropriate reference groups. Generally, sub folders are used to organize various kinds of cohorts.

`ref:role:` - institutional scope roles (e.g. president, provost, chaplain...)

`ref:employee:` - types of employees (faculty, staff, part-time, full-time...)

`ref:student:` - types of students (class year, on-track-grad, incoming-class...)

`ref:alum:` - types of alumni

`ref:course:` - course rosters including instructors, TAs, etc

`ref:dept:` - organization hierarchies

app folder

The `app:` folder is used to organize the groups and folders required to effectively manage access policy for a particular application. This may come in the form of access policy groups, account groups, and application specific reference groups. Each application has its own folder and a similar sub folder structure. Grouper templates can be used to build the application folder structure and to add a policy group.

`app:foo:` - root folder for the “foo” application

`app:foo:security:` - folder for administrative security groups for this application folder and group set

`app:foo:security:fooAdmin` - members have root-like privileges for the app:foo: folder and group set

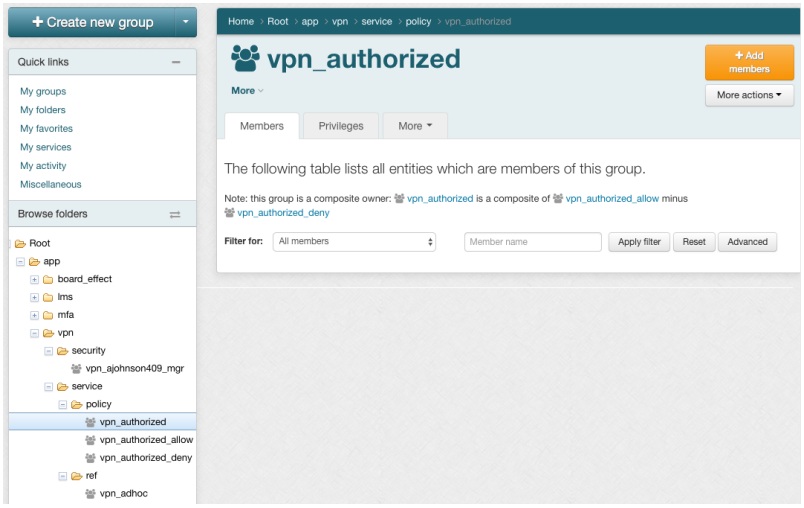
`app:foo:service:ref:` - folder for “foo” application specific reference group if needed

`app:foo:service:policy:foo_user` - access policy group, composite group (foo_users_allow - foo_users_deny)

`app:foo:service:policy:foo_user_allow` - generally this contains reference groups, composites, or manual groups. This is an 'intermediate' group type.

`app:foo:service:policy:foo_user_allow_manual` - only direct members people who should be included in the policy as exceptions, and are not automatically included. This is a 'manual' group type

`app:foo:service:policy:foo_user_deny` - may include `ref:iam:global_deny`. This is an 'intermediate' group type.



org folder

The top-level folder structure and overall Grouper deployment is typically managed by IAM architects within a central team IT department. This can be a bottleneck to adoption in large institutions. The `org:` folder provides a namespace for distributed IAM management. A distributed IT organization would be given root-like administrative privileges on a particular org: folder and could managed the namespace independently without the help of central IT. The org: folder is also sometimes used to delineate ownership of applications, and may be used for organizational scoped or maintained reference groups. The folder structure may replicate the top-level folder structure, but will be scoped to the particular organization.

`org:compsci:etc:compsci_admin` - members have root-like access to `org:compsci:`

`org:compsci:ref:` - Computer Sciences Department managed reference groups

`org:compsci:app:` - Computer Sciences Department applications

test folder

The `test:` folder is generally used by the IAM analyst for isolated verification testing of new production provisioning components and as a sandbox for group analysis and exploration.

Previous: Understanding Grouper

Next: Access Control Models