Kuali Service Summaries

Authentication Service

Description

This service establishes identity only. It may also provide a mapping from the identity "Principal" to the Person_Id.

Assumptions

* A Person may not have a Principal.
* A Principal may not be a Person in all cases; it may be another system or service.
* A person may have multiple principalIds each valid in different contexts, e.g. a person may be a staff member and a student. Each context would have different access levels.
* The current principal will always be available through the infrastructure/framework so there is no requirement for a fetchCurrentPrincipal() operation

Key Concepts

* The Person Service is the SOR for personIds
* The SOR for systemIds is TBD

Authorization Service

Description

The Authorization Service manages the maintenance, auditing, and checking of authorizations. The authorization will support process based security (e.g. ability to force add a student to a restricted section) and value based security (e.g. access to students in the Bachelor of Arts program) for individual principals and groups of principals. Access may be granted for specific periods of time.

Assumptions

* A principal can be a person, but it can also be a non-human entity such as an application.
* Some references are not updatable through this service, since the "core" information should poke through from the service of record.
* All authorizations are explicit.
* All authorizations are positive.
* Finding all permissions delegated by a principal is handled through a search operation.
    - Andy Bucior This may need to be restated. There's an expectation that you'll be able to determine the principal who granted the authorization, but since

connections between authorizations aren't explicitly visible through the service (at the moment), you may not be able to directly distinguish a "grant" from a "delegate" operation.

* Set up of roles, with associated permissions, etc. are handled in configuration.
* Set up of role categories and qualifier types are handled in configuration.
* Set up of qualifier hierarchies, including creation of "root" qualifier nodes, are handled in configuration.

Authorization

* An authorization is the composite of the roleId, qualifierId, qualifierType, and principalId/azGroupId.
* While permissions could be thought of as meta-data describing the functional role, authorization checks will be handled at the permission level; however, granting authorizations will occur at the role level.
* Most authorizations are scoped to a particular context. Very few entities can perform the same function in all cases.
    o Ex. An instructor can view the roster for a class. A department administrator can see the rosters for the classes taught by the department. A person in the Registrar's office can see the rosters for all classes taught at the university.
* There are dominant pieces of information in the context which control authorization. These dominant pieces can be arranged in a tree form to allow inherited permissions.
    o Ex. An instructor can view the roster for a class (qualifier = class). A department administrator can see the rosters for the classes taught by the department (qualifier = department). A person in the Registrar's office can see the rosters for all classes taught at the university (qualifier = institution).
* The caller checking an authorization shouldn't need to know necessarily how an authorization was granted, just that the principal has the appropriate authorization.
* The caller should be able to check authorizations in a consistent way with a minimum amount of secondary lookups.
    o The caller is assumed to know at the minimum
        + "who or what is attempting to perform the action" = principal
        + "what action is being requested" = permission (could also be the role depending on approach/naming)
        + "in what context is the action being performed" = qualifier - this usually maps to the object id being worked on

Delegating Authorizations

* When allowed, authorizations can be delegated to another principal within the constraints of the initial authorization. In other words, you shouldn't be able to delegate more authority than you actually possess (this is a different concept than having authority to grant authorizations in general).
    o Ex. An instructor is heading out of town for a week or two and needs to allow his TA to submit grades for his courses.

    o Ex. If the "primary" authorization has an end date of 9/1/2008, the delegated authorization can't have an end date beyond 9/1/2008.
  * Authorizations can be delegated repeatedly.

Qualifiers and Qualifier Hierarchies

  * The combination of qualifierId and qualifierType is assumed to be unique enough to disallow collisions between qualifiers.
  * Qualifiers will have one to many parent qualifiers; the parent can be of the same or different qualifier type.
  * The root of one qualifier hierarchy can be registered as the child in another hierarchy.
  * Linking a qualifier does not include any child nodes that have been registered in other hierarchies.
  * Qualifier types can be restricted to disallow updates to associated qualifiers.
  * The same qualifier hierarchy may supply the context for multiple permissions and roles.
    o Ex. A hierarchy of class is a child of department is a child of institution may support both viewing the roster as well as submitting grades.

AZ Group Service

Description

The AZ Group Service manages groups consisting of principals and/or other groups. This service is narrow in scope and is primarily used by the Authorization Service to grant and revoke authorizations by group rather than individual principals.

Notes

  * May be merged with the Authorization Service at some point.

Assumptions

  * Service allows group members to be managed, e.g. retrieved, updated, added or removed.
  * Service is aware of direct and indirect (i.e. through hierarchies) group memberships.
  * Service does not currently support group types.
  * Group members may not be able to see the membership of their own group(s).

Key Concepts

  * Groups may contain principals. An organization or person may not be a member of a group - but the principals of the members of that organization may be members of a group.
  * Groups may contain other groups.

* Groups have a single membership option, e.g. member. Distinct from the Organization Service which contains both people and role information, e.g., membership type of Chair, President, Secretary etc.
* Organizations cannot be part of a group, e.g. like magnet high schools. A group is defined only to be used in an authorization context.
* There are no named hierarchies - there is only one hierarchy here that is not mandatory (allows orphans), thus it does not have the same issues as you would find in learning objective hierarchies.

Communication Service

Description

This service supports sending and posting messages to people and organizations. Messages may be sent using a number of different methods including email, mail, worklists, portal messages, etc. Future releases of the service will support Word-like mail merges. This will allow personal information (e.g. names) to be merged with fixed content.

Assumptions

* This service is not the SOR for persisting contact information, e.g. preferred email address, etc. This is handled by the Contact Service.
* Some message types may be able to be converted, e.g. voicemail to email, speech to text translation.
* Initially this service is not covering the following concepts:
    - Subscription
    - Threading
    - Message retrieval
    - Delivery tracking
    - Message boards
    - Complex message structures
    - Attachments
* There will be no database persistence for this release; functionality such as persisting messages will be addressed in a later release

Key Concepts

* The initial focus of this service is limited to facilitating communications, there is no persistence of messages, recipients and related usage and media types
* Media Type refers to the format for the communication, e.g. email, street address, IM, phone, etc.
* Usage Type refers to the premise of the communication, e.g. billing, home, campus, emergency, etc.

Contact Service

Description

This service manages contact information for communication to people, groups of people or organizations. Contact information may include (but is not limited to) mailing addresses, e-mail addresses, phone and fax numbers, etc. The individual's or organization's preferences around which contact record should be used in predefined contexts is also retained here.

Assumptions

* Media types and usage types will be managed using a config utility and not through service operations
* Constraints between media type and contact type are also managed via a config utility

Key Concepts

* Media Type is the format of the contact information, e.g., phone, mailing address, email.
* Usage Type represents the focus or reason for the communication, e.g., student billing, academics
* Preferences can be defined for people and organizations in support of multiple prioritized contact methods (media type, e.g., text message, cell phone, fax) for each usage type. Effective dates support expected movement of a student throughout the academic year, e.g., different phone and mailing address for Summer.
* A contact record may only be a single media type at a time. The information contained within the contact record is structures by the media type. In other words, a phone number will have a different structure than a street address.
* The same contact record may be used in multiple contexts (usage types).
    o E.g., the same street address may be used for both class-centric communications and billing communications.
* Certain usages may constrain the allowed media types associated with them.
    o E.g., an alert usage may be restricted to synchronous methods only, which restricts the contact records to media types associated with those methods: no street addresses, e-mail addresses, etc.
* An individual or organization may interleave media types in their preferences for a given usage.
    o E.g., "Contact me by this e-mail address first, then try this phone number, then try this other e-mail address, then try this other phone number."

Organization Service

Description

This service manages organizational units that have a relationship to the institution. The organizations may be internal and include officially recognized organizations (e.g. Departments, Faculties, Schools) or unofficial organizations (e.g. clubs or student groups), or they may be external organizations (e.g. companies, other

institutions, government, associations). This service also manages the relationships between people and organizations.

Assumptions

* Most organizations have "parent" organization(s) within a given context.
    o E.g., The School of Arts and Sciences exists within the institution as a whole. Parent:child::institution:School of Arts and Sciences
* The "parent" organization(s) of an existing organization may shift depending on the context. This leads to the need to capture multiple relationships for a given organization.
    o E.g., A department may report to a particular institution for administrative purposes, but report to another institution for financial purposes.
* Organization to organization relationships can be grouped into hierarchies based upon the type of relationship.
* Organizations may place additional constraints on the types of relationships a person may have with the organization.

Key Concepts

* Organizations are distinguished from authorization groups in that organizations deal directly with people while groups deal directly with principals. In other words, organizations may be comprised of individuals who have no way to authenticate themselves (and thus have no unique permissions) and AZ groups may have principals which are linked to non-human entities (such as batch jobs, other services, etc.).
* Organizations and groups may be related, in that a member of an organization may have one of their principals associated with an AZ group, but this is not required.

Person Service

Description

The Person services supports the management of people related to Kuali Student. This includes managing person to person relationships, but does not include relationships of a person to a non-person entity (like an organization). Although generalized information such as academic or contact information may be available through this service, manipulation of this information is handled through the appropriate service of record. The concept of attribute sets allows for the grouping of person attributes. Once defined, attribute sets can be associated with the person types to identify the relevant information for a given person. A person will always have one or more person types assigned.

Assumptions

* the service exposes attribute sets directly at this level
* the service may know about more personTypes than it can create people for

* attributes aren't created for a given person directly through the service (so treated as if created when person is)
* deletes occur at the person level
* person information must be complete and valid before an addition/change of personType is allowed
* person deletion may indicate that the identifier is retired at first, so the identifier may be found, but no longer valid. While we talked about the error and look-up as a group, we hadn't come to a consensus on if we should have functionality like this exposed here. As a result, all the DISABLED_IDENTIFIER errors and the fetchReplacementPersonId operation might go away.
* isValidPersonInfoForPersonType is not listed since validatePersonInfoForPersonType has similar functionality.