Kuali Identity Management

Goal/Problem Space

The Kuali Identity Management (KIM) module of Rice is intended to provide Identity and Access Management services to Kuali and other applications. It also provides graphical interfaces for managing identity and access management data.

Features

- Principals and Entities
  - numerous built-in entity attributes for name, phone, address, affiliations, external ids, employment info, email, and privacy preferences (FERPA)
  - support for different types of entities, such as Persons, Systems or custom types
- Groups
  - supports group nesting
  - custom attributes
- Permissions
  - evaluated by the KIM PermissionService
  - applications can plug in custom logic for complex or data-dependent permission evaluation
- Roles
  - used to aggregate permissions
  - principals, groups and even other roles can be assigned and will be granted the permissions on the role
- Responsibilities
  - defines actions that a principal is being asked to take (i.e. approve this document)
  - provides integration with the Kuali Enterprise Workflow (KEW) engine
- Integration with the Kuali Service Bus, including:
  - services published as SOAP web services
  - services published using Java serialization over HTTP
- Designed to allow for the various service reference implementations to be customized or overridden on a service-by-service basis

Technology Stack

Java, Spring Framework, OJB, JPA, Struts, Apache CXF, MySQL, Oracle, XML, JTA, plus more

Identity Services

Please indicate which of the following identity services/transports you consume, produce, or define.

Managed Information	**Consume?**	**Produce?**	**Broker /Convey?**
Privileges	X	X	
Roles	X	X	X
Groups	X	X	X
Attributes	X	X	X
Identification	X	X	X
Defined Interfaces	**Consume?**	**Produce?**	**Broker /Convey?**
Authentication	X		
Attributes	X	X	X
Permissions	X	X	X
Provisioning	X	X	X
Authorization	X	X	X
Subjects	X	X	X
Other	**Consume?**	**Produce?**	**Broker /Convey?**
Responsibilities	X	X	X

Standards and Interfaces

The KIM API provides 6 standard services:

- IdentityService
- GroupService
- PermissionService

- RoleService
- ResponsibilityService
- AuthenticationService

All interaction with KIM happens through these services. If accessed remotely over the service bus, then the communication is authorized by digital signatures between the two endpoints. In the case of SOAP web services this is done using WS-Security.

The default implementation of the AuthenticationService just uses the REMOTE_USER on the incoming request to identify the incoming principal. Out of the box, KIM provides integration with CAS.

Issues and Challenges

1. Currently, KIM hasn't done much testing with federated identity which is an area we want to be sure we have good support for.
2. KIM doesn't currently provide good out-of-the-box connectors for things like LDAP or other identity services. The goal of KIM is to allow for easy plugability so it would be good to have these.
3. As stated, plugability is one of the goals but there hasn't been much work done yet to try plugging in alternate IDM systems behind the KIM services. It's likely that once we have more implementations using KIM, we will identify some gaps in the APIs.
4. Creating a central permission system that is both usable and performant is a challenge. Steps have been made to try and make KIM as efficient as possible but this is still a challenging issue.

More Information

Rice Homepage - http://rice.kuali.org
Internet2 Presentation on KIM - https://test.kuali.org/confluence/x/CwDGCQ
Rice API Javadocs - https://test.kuali.org/rice/rice-api-1.0-javadocs/
Draft Documentation for KIM - AG KIM draft.doc