Kuali Response To Registry Questionnaire

IAM Registry questions to evaluate features and functionality against standard business requirements.

Category	Description or Question for solution provider	Response	Link(s) to Documentation
General architecture	Describe how ID match capability is provided by the registry solution. For example, is it (a) an integral part of the solution as provided or (b) must it be integrated with an external ID match engine or (c) can it be provided in some other way?	KIM does not presently provide built-in ID match functionality. Currently, most implementers are not managing identity directly via KIM but rather provisioning identity data into the KIM registry. So the ID match process is typically institution-specific based on what other products and tools they have in place. However, recent project work within Kuali is surfacing requirements to have better support for this. Specifically, the Kuali People Management for the Enterprise (KPME) project (which is HR/Payroll/Time/Attendence) as well as the Kuali Student project. Both of these projects use KIM as their master repository of identity data and will need to have good support for maintaining identity therein, including appropriate ID match functionality. Additionally, these projects are implementing systems which are typically the primary sources of identity within institutions of higher education.	• Kuali Student: Duplicate /Matching Logic • Kuali Student: Merging Duplicates • Kuali HR: Life Long ID and Person Registry
	Describe how groups management (for use with authZ controls and other purposes) is provided. For example, is it (a) handled internally by the solution or (b) integrated with an external group management engine such as Grouper or (c) provided in some other way?	KIM supports both of the following: 1) An out-of-the box reference implementation of roles and groups, including user interfaces for management of roles and groups. 2) Integration with external implementations of groups and roles through a standard set of service contracts defined in KIM. There is a connector that was developed by the Grouper team which allows for usage of Grouper via the KIM service contracts. Additionally, members of the community have integrated with LDAP groups (such as Microsoft's Active Directory) as well as other sources for group and role data. KIM has the concepts of both groups and roles and draws the distinction between them with groups simply being a collection of principals or other groups and roles being similar but also allowing for permissions to be granted to them. The group contract is defined by the `GroupService`: * http://site.kuali.org/rice/2.0.0/apidocs/org/kuali/rice/kim/api/group/GroupService.html * http://maven.kuali.org/release/org/kuali/rice/rice-kim-api/2.0.0/rice-kim-api-2.0.0-GroupService.wsdl The role contract is defined by the `RoleService`: * http://site.kuali.org/rice/2.0.0/apidocs/org/kuali/rice/kim/api/role/RoleService.html * http://maven.kuali.org/release/org/kuali/rice/rice-kim-api/2.0.0/rice-kim-api-2.0.0-RoleService.wsdl KIM also includes an api for performing authorization checks, we call this our `PermissionService`: * http://site.kuali.org/rice/2.0.0/apidocs/org/kuali/rice/kim/api/permission/PermissionService.html * http://maven.kuali.org/release/org/kuali/rice/rice-kim-api/2.0.0/rice-kim-api-2.0.0-PermissionService.wsdl Note, however, that we don't really consider these as parts of the "identity" portion of KIM. Generally speaking, KIM has 5 sub-modules: 1) Identity 2) Groups 3) Roles 4) Permissions 5) Responsibilities	• http://kim.kuali.org
Data model	Describe how the registry solution supports an extensible set of attributes about (a) persons, (b) applications or other external resources, and (c) other, arbitrary entities?	This is something that KIM does not do a good job of currently for identity data. The main way that someone would extend the current schema is to do so manually via modification to the database (which is a traditional relational database) as well as modify the associated service API layer. There are a few places in the identity data model where extension is supported: * There is a concept of "external identifiers" which can be used to associated the identity with any number of desired identifiers. * KIM supports the concept of "Entity Types". There are two default types supported out of the box: PERSON and SYSTEM. Through this mechanism it's possible to extend KIM to support different types of entities. * KIM has support for an arbitrary number of addresses, phone numbers, names, etc. for a given identity record.	• KIM Data Model
AuthZ support	Describe how the registry data model supports defining arbitrary user roles in support of authZ functions.	This ties into the response to the earlier question about groups management. But KIM has support for the concept of "Roles". In KIM a role is essentially a group of identities which can have permissions granted to it. So a role in KIM that might be used in something like the financial system would be an "Account Manager". Account managers can then be granted certain permissions within the system. In KIM, permissions work off the concept of "Permission Templates". So you might define a set of templates like the following: * Administer Routing for Document * Perform Custom Maintenance Document Function * Manually Execute Batch Job * Upload Batch Input File(s) * Maintain System Parameter You then create permissions from these templates which provide additional details to help qualify the permission such as: * Administer Routing for **Purchasing** Documents * Manually Execute **General Ledger** Batch Job * etc. KIM also has a concept of storing affiliations which can be used for very course-grained roles such as: * Student * Faculty * Alumni * Staff * etc.	

Features	Describe how the registry solution supports audit logging of sensitive transactions, including support for the recording of historical changes made to sensitive data. Describe how this log includes the requester and authorizer identities, and transaction timestamps.	KIM has partial support for this. Specifically, it uses Kuali Enterprise Workflow (KEW) in order to route changes for possible approval. KEW records the following information about a particular transaction: * Who initiated it * It's current status * Who action was requested from * Who took action * Timestamps on all of the above However KIM does not currently make a full copy of each record and store it for historical purposes before updating the existing record. So it's not currently possible to do effective date reporting on records. For example, you can't ask it what someone's name was 3 years ago, or similar historical reporting. This is a feature which has been requested by the community but not yet implemented. Additionally, KEW routing is typically only performed whenever updates are made from the user interface adminstration screens. Using the service API to update records does not currently kick off workflow processing. One thing that KIM does support however on nearly all records is an "active" indicator which services as a form of logical deletion of the record. And all data has a "last update" timestamp which is stored in the database.	
	Describe how the registry solution supports the secure storage of security questions and answers for use in password recovery.	KIM was not originally designed to store passwords or security questions and there has been no push from the community as of yet to add support for that.	
	Is there support for multiple name and address types as well as history? If yes, please describe.	Yes, there is support for multiple name and address types. KIM has a few built in ones: **Name Type:** * Preferred * Primary * Other **Address Type:** * Home * Work * Other The available name and address (as well as other) types can be extended and there is also a user interface which can be used to maintain these or add new types. See the earlier section on Audit Logging recording KIM's support for tracking of historical data. In the case of addresses and names, if these are ever changed the old name/address will be marked as inactive and the new record will be created as active. So, in this case, name and address has better capability for recording of historical changes than some of the other parts of KIM.	
Identity Assurance	Are registration events captured as they occur? Do these events automatically trigger assignment /deassignment of an IAP	KIM does not currently have support for Identity Assurance Profiles.	
	Is there support for real time provisioning of Identities /services	KIM has the apis that allow for creation and updating of identity data. There's not much additional infrastructure provided out of the box that sits on top of this however. So if someone wanted to provision realtime into KIM they would need to invoke the service themselves. The main service in KIM through which this would be done is called the `IdentityService`: • http://site.kuali.org/rice/2.0.0/apidocs/org/kuali/rice/kim/api/identity/IdentityService.html • http://maven.kuali.org/release/org/kuali/rice/rice-kim-api/2.0.0/rice-kim-api-2.0.0-IdentityService.wsdl	
	Describe how data is processed (batch, web services)	Data could be processed through either batch or web services. Our current web services use SOAP, but there are plans in the future to provide RESTful interfaces to these. As mentioned previously, the data model is a traditional relational data model and therefore fairly straightforward to work with. One of the caveats with bypassing the service layer however is that KIM has a fairly sophisticated caching infrastucture that allows applications which are consuming information from the registry to cache that data and get notified about updates to it via a message queue which will automatically flush their local cache. This allows for increased performance at the application layer but does open the possibility of clients having stale data in their cache. Appropriate caching configuration can alleviate this. We use Ehcache (http://ehcache.org/) for caching and take advantage of the Kuali Service Bus (http://ksb/kuali.org) messaging layer to distribute and route notification messages.	
	Is registry dependent on other open source or vendor products? If yes, please provide details.	Yes, all of Kuali Rice (of which Kuali Identity Management is a module) is licensed under the Educational Community License and uses many other open source libraries with compatible licenses. It is designed to run inside of a standard Java servlet container (such as Tomcat) and takes advantage of the Spring Framework, Apache CXF, various Apache commons libraries, JAX-WS, Quartz, wss4j (which implements WS-Security), and JTA (the Java Transaction API) among others.	
	Where is the business logic stored? Is there support for delegation to maintain these rules?	Business logic is currently stored behind service implementations which are overridable and customizable for those who implement. Additionally, with version 2.0 of Kuali Rice a new module has been introduced called Kuali Rule Management System (http://kuali.org/rice/modules/krms) which is a business rule management system which can be used to maintain and execute business rules for routing, validation, and various other purposes. So there is the possibility for integration with that module at some point in the future.	• KRMS
	How does the registry notify external entities of data changes? (for example name is changed)	Notifications about data changes are done via the reliable messaging component of the Kuali Service Bus. However, not all events that occur within KIM trigger outbound messages at the present time and the ones that do are not very granular. For example, they are more along the lines of "entity changed" as opposed to "the entity's name changed".	
	Is code located in public repository	Yes, see: http://svn.kuali.org/repos/rice/	
	How are changes, marketing, etc communicated to public? (wiki, lists, web presence)	The Kuali Foundation has various channels that can be used for communication and the Kuali Rice project itself has some of it's own. This includes the following: 1. The Kuali News Feed where release announcements and other information can be shared. This then goes out to an RSS feed as well as via email to those who are subscribed. This is hosted from http://www.kuali.org 2. The rice.collab@kuali.org mailing list which people can use to get help or ask questions. We also send out announcements about upcoming work and opportunities to this mailing list. 3. The Kuali Rice website (http://rice.kuali.org) 4. The Kuali wiki: https://wiki.kuali.org 5. The release notes, change log, and documentation for each release which are written and generated using Docbook. Example here: http://site.kuali.org/rice/2.0.0/reference/html/portal.html 6. Bi-Monthly gatherings of the collaboration group. This is an open forum and anyone is permitted to attend these and ask questions or provide feedback for the group. We also discuss project status updates and project activity at this meetings as well. 7. The annual "Kuali Days" conference. Once a year we have a conference where there are many sessions and presentations given on the various activities going on within the Kuali Community: http://kuali.org/kd	• http://kuali.org • http://rice.kuali.org • http://wiki.kuali.org • Kuali Days

	Is there proper OSS license?	Yes, Kuali Rice is licensed under the Educational Community License version 2.0. We have occasional code audits using 3rd party vendors (most recently Black Duck Software to ensure license compliance. We additionally utilize automated tools as part of our build environment to ensure proper license attribution and acknowledgements in source code.	• ECL 2.0
	Is there a clear project lead?	Yes, the Kuali Rice project actually has a few different leadership positions. The project team is structured as follows: * Project Manager * Lead Technical Architect * User Experience Architect * Business Analyst * Configuration Managers * Development Managers * Developers The project manager handles the schedule, resourcing, and the budget. While the technical architect is responsible for ensuring conceptual integrity and architecture for the product as well as doing technical design and analysis with the other technical leads. The user experience architect and business analyst work as functional leads on requirements and ensuring that the system works the way it's supposed to and is easy to use. The business analyst also functions as the Quality Assurance lead for the product as well. Each of the individual sub-teams on the project are lead by a development manager who is essentially the technical lead for that team (they do development as well), and then developers develop the majority of the code. At the time of this writing the Kuali Rice project currently has about 19 FTE working on it. Most of these are directly involved in development. However, it's important to note that Kuali Rice includes more than just KIM and is rather large in scope. So these resources are working on numerous different projects and modules.	• Kuali Rice Project Organization • Kuali Rice Project Team
	Is there an existing project steering committee /governance?	The Rice project has the following governance bodies: 1) Kuali Rice Board 2) Application Roadmap Committee (ARC) 3) Technology Roadmap Committee (TRC) All governance bodies are represented by a group of voting members. Each institution which has invested in Kuali Rice has a vote as well as each other Kuali project which has invested in Kuali Rice has a vote. Each of the groups has a chair and vice-chair who are appointed via an election process and serve a 1-year term. Once the current chair's term is completed, the vice-chair assumes the chair role and an election is held for a new vice chair. All groups meet on a bi-weekly basis. The board helps the project with resource and budgetary decisions as well as high-level strategy and advisement. The ARC defines the product roadmap and desired schedule via a formal roadmap prioritization process. The ARC has a standing working group called the Kuali Application Integration working group (KAI). This group is responsible for functional governance and change management for Kuali Rice. The TRC defines the product's technical roadmap and strategy. The TRC has a standing working group called the Kuali Technical Integration working group (KTI). This group is responsible for technical governance and change management for Kuali Rice. It's important to note that while the TRC defines technical roadmap, the ARC is the group which makes the final decision on the roadmap for the product based on input from the ARC, TRC, and the Kuali community.	• Kuali Rice Project Organization • Rice Project Charter • Application Roadmap Committee • Technology Roadmap Committee