Functional Requirements

General Description

Below is an overview of the functionality Project will enable through deployment of the identity management system in Phase 1.

|  |  |
| --- | --- |
| **Capability** | **Requirements** |
| **Automation** | The system must support multiple systems of record; be able to filter just changes of interest; be able to transform changes to multiple formats, suitable for multiple target systems.  Automation logic:   * The business logic for monitoring changes to user profile information in systems of record (e.g., HRMS), and for filtering, transforming and pushing out updates to target systems must be highly configurable. |
| **People and Accounts** | Every employee and contractor who has access to the SRP network and applications will have identity records in the identity management system. An enterprise login ID will be configured to match the user’s primary Active Directory login ID, using a template such as first initial, middle initial and the first six letters of the last name (e.g., ‘jssimpso’). The enterprise login will be the unique naming attribute (e.g., uid, RDN) for the user in a single ‘person’ object in the identity management system.  The system must also support the configuration of a single ‘person’ object having one or more accounts in target systems, such as Active Directory and RACF initially. Eventually, other target systems like Oracle, UNIX, and SRP applications must be supported. The accounts must be created through the identity management system, subject to workflow approvals, linked to a person object, entered into groups (e.g., RACF and/or AD) based on the user’s cost center, job code and other attributes, then configured accordingly on the target systems. RACF accounts must be created using a rule such as:   * If employee, prefix a letter A-Z corresponding to a table lookup (cost center-related) to the employee’s 5-digit employee number (e.g., ‘X09535’) * If contractor, prefix a letter A-Z corresponding to a table lookup (cost center-related) to the contractor’s 3-digit employee number, then suffix the result with the letters ‘TP’ (e.g., ‘G372TP’)   Note that the identity management system must also be capable of reconciling accounts on AD and RACF. For example, if an account is created directly on RACF, the identity management system must be able to reconcile the difference, notify security administration, and allow synchronization with the identity management system upon approval. |
| **Groups** | The system must support the assignment of user accounts to existing AD and RACF groups initially. Additionally, the system must allow identity management administrators to create new groups in AD and RACF through the IdM interface. Conversely, the system must also discover new AD and RACF groups defined on these systems, assign them to users in IdM who are in them on target systems, and make the new groups available for assignment to other user accounts via the identity management system. |
| **Object Relationships** | The figure below represents the relationships to be supported between a person object, one or more account objects, groups, cost centers and non-person accounts (“generic” accounts).  SRP DIT 2-23-05  Figure 2-1: Object Relationships in the Identity Management Structure  Note the relationship between people and accounts: each account has an owner, which is the Distinguished Name of a specific person object. This relationship is crucial for tracking multiple person accounts and non-person accounts (“generic” accounts) on managed systems (initially AD, RACF). A more detailed definition of the proposed schema is provided as an attachment to this RFP. |
| **Workflow** (provisioning and de-provisioning, etc.) | The system must allow:   * any user to   + submit specific security requests (such as change physicalDeliveryOfficeName (i.e., mail stop))   + submit self-service password reset request   + track the status of their request * any delegated administrator to   + validate security requests   + authorize initial security based on requested resources, identity of the requester, and other request attributes   + track the progress of a workflow   + cancel open requests if appropriate     IdM provisioning and deprovisioning high level process flow  Figure 2-2: Typical IdM Workflow Lifecycle |
| **Workflow Cont.** | **Security Request Approval Workflow Example**  One key capability of the IdM system is to accept and validate security requests, to route them to the appropriate authorizers, and to act on security requests once sufficient authority has been received.  The system must be capable of fulfilling the following workflow approval scenario:     * A requester (typically an HR Rep, IT Security Admin, delegated administrator, or User) logs into the identity management system, and requests new or changed security access. * The security request may include information about a new person, or identification of an existing one. * Security requests may include multiple SRP-defined attributes. * Security requests are routed to appropriate authorizers, based on requested login id access and membership in groups. Potentially cost center, location and job code are considered. Cost center and group owners are cross-referenced to specific authorizers. * Authorizers are asked to review open requests by e-mail, where the message contains a URL linking them to a secure web form. * Authorizers sign into a secure web form to approve or reject requests that await their review. * An authorizer is sent a reminder, and then delegation and escalation is used to ensure that authorizers respond in a timely manner. * As the appropriate authorizers have approved their part of a request, the identity management system makes the changes to the target systems.   This process is illustrated in the Figure below.  Figure: parreq-3  Figure 2-3: Workflow to Route Security Requests |
| **Workflow Cont.** | Additional key IdM life-cycle process scenarios (including but not limited to the following):  • Create new and delete existing accounts for a user.  • Change active/de-activated status of existing accounts.  • Change the login ID of an existing account (due to user name or cost center changes).  • Read attributes of existing users and their accounts.  • Modify the membership of existing accounts in RACF and AD security groups.   * Receive triggered change events from outside sources, including name, position and status changes.   • Update other user profile information.  The system must be capable of supporting these workflow tasks and be flexible enough to support future IdM life-cycle workflow needs. |
| **Workflow System** | Additional Workflow System Requirements:   * The workflow system must leverage existing data to identify primary authorizers, and to find alternative authorizers to receive escalated security requests. * The workflow system must be able to limit what users can request. |
| **Central user management / administration** | The system must allow security administrators to manage any user on any system from one point; implement access control lists to control what each administrator can see and do; be fully accessible from a web browser. |
| **Delegated user management** | The systems must allow organization-specific logic to leverage existing information to make dynamic delegation decisions (i.e., can this security administrator sign in, and if so what can they update?). |
| **Authentication** | The system must allow access to authenticated users only.  Any authentication failures must be tracked. The system must be able to lock users if the authentication failure threshold has been reached. |
| **Auto-discovery** | The system must be able to efficiently extract full lists of users and groups from each managed system; be able to extract specific user attributes and group memberships from managed systems, without adversely impacting run-time performance. |
| **Password management** | The system must be capable of password synchronization, self-service password reset, assisted password reset, global policy enforcement, early warning of upcoming password expiration, etc.; be accessible from a web browser and workstation login prompts.  Password policy:  The system must be able to enforce a global password policy for accounts. |
| **Password initialization** | The system must generate one-time passwords for users, using configurable rules. |
| **Audit logs** | The system must be capable of tracking user access rights on the managed systems, and enable full change history. |
| **Reporting engine** | The system must support both built-in reports (e.g., who has access to what), and allow for custom reports. |
| **Security** | An identity management system must meet the most stringent security requirements:  Encryption   * User access to the system must be encrypted (this typically means only HTTPS must be supported as a user interface). * Any sensitive data stored in the system must likewise be encrypted or hashed, as appropriate. This includes administrative passwords of people authorized to manage the system, as well as passwords used by the system to manage target systems. This also includes any sensitive user profile data. * The system must support encrypted communication with all managed systems -- including those that do not natively implement an encrypted client/server protocol (e.g., most DBMS servers, mainframes, etc.). * Encryption must rely on well-known implementations of well-known, trusted encryption and hashing algorithms. * Encryption keys must be managed effectively. * The amount and type of information available in e-mail or any other non-secure notification must be configurable (e.g., the ability to exclude any sensitive and/or private information that was used during the provisioning process).   Platform   * The system must be able to operate on a locked down operating system. * The system must support a diversity of web servers, so that if a given web server is deemed to have an unacceptable history of vulnerabilities, it can be avoided. * The system must be accessible across web proxies, so that it can be installed in a protected subnet, and accessed across a firewall without opening non-HTTPS ports. * The system must not require the installation of (possibly insecure or vulnerable) client software. |
| **Flexibility/ Scalability** | The system must cope with both current and possible future requirements for (including but not limited to):  User interface:   * The user interface must be customizable, and support different appearances for different users (such as basing it on user groups).   Login ID assignment:   * The mechanism for assigning login IDs to new users must be customizable to allow organizations to implement their own logic. |
| **Target Systems** | The identity management system must be able to manage login IDs on most if not all of the systems to which users log in. Managed systems must work `out of the box' in as many cases as possible. Where this is not feasible (e.g., home-grown applications, legacy applications), the product must be open enough to make it possible to easily integrate with these other applications. |

**TECHNOLOGY questions**

## Architecture

Below is a diagram of a potential architecture to support approximately 20,000 end users and accounts. Please use this as a guide to describe your proposed architecture in detail. Clearly label each component in the diagram, and identify whether each component is required or optional. Identify any third party components and whether they are integrated into your product, bundled with your product, or purchased separately. Include options to provide fault tolerance, load balancing and disaster recovery.

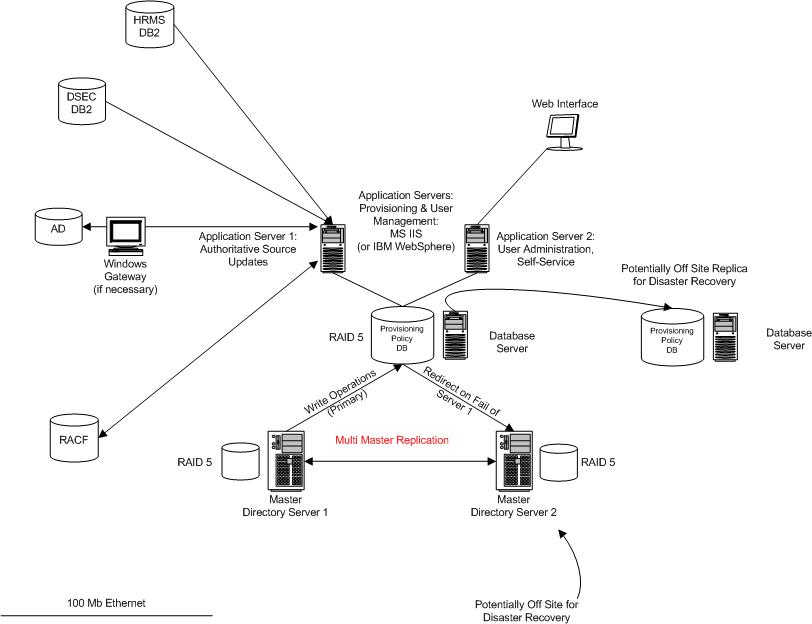


Figure 3-1: SRP Identity Management Architecture Template

Provide an appendix with term definitions.

**CA’S RESPONSE:** There is insufficient information in this RFP to architect the final solution. CA Technology Services will provide the architecture and best practices when more details are known.

## Platforms

Specify the required platform for each architectural component identified above to support no more than two-second response times to system operations initiated by administrative interfaces or integration interfaces. Include both minimum and recommended system requirements. If the component can be deployed on more than one platform, provide this information for all platform options and indicate which platform you recommend based on the requirements provided in this RFP. Include the percentage of your entire customer base currently using each platform and percentages for your customers in the energy/utility sector. Also, indicate which platform you use to perform your primary development activities.

## Security

Describe the application’s security model in detail. Indicate whether authentication and authorization are performed by the application or if an external method is used. If external, indicate what that method is. If the system is capable of integrating with a third party directory service, list the products your system supports, including version numbers. If a DBMS is utilized, provide details on how the data is secured. If different levels of security are provided for different types of users (such as those with view only and those with update permissions) describe how your product ensures correct access levels.

Describe the methodology or tools you recommend for auditing/validating the system’s security. Indicate whether the tools are integrated in your product, bundled with your product, or purchased separately.

## Data

Describe how your system manages persistent data. Include a logical data model with definitions of entities, elements and relationships. If XML is used, provide schemas and identify any standard vocabulary used, such as DSML, SPML or others. Identify any processes or tools required for data conversion and archival. Indicate whether each tool is integrated with your product, bundled with your product, or purchased separately.

## Development/customization

Describe how customizations to your system are developed. Include a list of tools and environments used and whether they are required or optional. Indicate whether each tool is included with your product or purchased separately.

## Interfaces

Describe the available methods for interfacing to your system and indicate the recommended method. Include information on the technology involved, such as an API or Web Service, as well as supported formats, such as flat files or the OAG Interface Specification. Indicate whether the interface options are proprietary, third party, or open standards. Identify any additional products available to interface with your solution. Indicate whether they are required or optional and whether they are bundled with your product or purchased separately.

## Reporting

Describe the available technology for reporting from your system and indicate the recommended method. Indicate whether the reporting technologies are proprietary or third party and whether they are integrated into your product, bundled with your product, or purchased separately.

## Upgrades

Describe your overall approach to product upgrades. Include frequency of releases and upgrades and limits on support for prior versions. If your product is available on multiple platforms (including DBMSs), indicate any lag between releases for the first platform and subsequent platforms. Provide the general availability date and identification numbers for the last three releases of your product.

Describe the process of installing a new release. Include a description of any conversion or migration tools and whether they are integrated into your product, bundled with your product, or purchased separately. Also, describe how customizations are handled during a new release.

## Support

Describe your overall ability and approach to fulfilling customer service needs. Include descriptions of your support for the functional, feature, and technical aspects of the system. Give specific product and service descriptions, including contact information, hours of operation, response times and support personnel qualifications. Identify any limitations to the support described, such as unauthorized modifications or operation in a shared environment.

Describe your problem resolution process, including how problems are reported and problem escalation procedures. Describe the environment you maintain to recreate customer problems and to test problem resolutions. Be specific about the technical architecture of this environment, including hardware, operating systems, tools and DBMSs.

Describe any methodologies or tools you recommend for monitoring and troubleshooting your system. Indicate whether the tools are integrated into your product, bundled with your product or purchased separately.