Common Fishery Policy

DG-MARE - Information System

USM Component – Design Document

[12-May-15]

Contents

[Introduction 3](#_Toc419210786)

[Purpose and Scope 3](#_Toc419210787)

[References 3](#_Toc419210788)

[Terminology 3](#_Toc419210789)

[Requirements Overview 4](#_Toc419210790)

[Use Cases 4](#_Toc419210791)

[Actors & Packages 4](#_Toc419210792)

[Application Actor 6](#_Toc419210793)

[Application User Actor 7](#_Toc419210794)

[User Manager Actor 8](#_Toc419210795)

[Supplementary Requirements 9](#_Toc419210796)

[Component Design 10](#_Toc419210797)

[Offered Interfaces 10](#_Toc419210798)

[Authentication Service Interface 11](#_Toc419210799)

[Authorisation Service Interface 12](#_Toc419210800)

[Information Service Interface 13](#_Toc419210801)

[Administration (User) Interface 14](#_Toc419210802)

[Self-Service (User) Interface 15](#_Toc419210803)

[Consumed Interfaces 15](#_Toc419210804)

[Domain Model 16](#_Toc419210805)

[Data Model 18](#_Toc419210806)

[Tables and Relationships 18](#_Toc419210807)

[Application Domain 20](#_Toc419210808)

[Organisation Domain 21](#_Toc419210809)

[User Domain 22](#_Toc419210810)

[Role Domain 23](#_Toc419210811)

[Scope Domain 23](#_Toc419210812)

[Authentication & Authorisation Views 24](#_Toc419210813)

[Component Model 25](#_Toc419210814)

[Deployment Model 26](#_Toc419210815)

[Integration with ECAS / CAS + IDM 26](#_Toc419210816)

[Integration with CAS 27](#_Toc419210817)

# Introduction

## Purpose and Scope

The purpose of this document is to document the design of a User Management (USM) component that provides services for the authentication, authorisation and notification of end users of the DG-MARE - Information System for the support of the Common Fishery Policy.

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Reference** | **Title** | **Version/Date** |
| R1 | N/A | DG-MARE - Information System, Software Architecture | 17-Feb-15 |
| R2 | N/A | Union VMS (USER MANAGEMENT) Requirements | 0.2  17-Feb-15 |

## Terminology

|  |  |
| --- | --- |
| **Term/Acronym** | **Description** |
| CAS | Central Authentication Service |
| ECAS | Single Sign On solution based on CAS used by the European Commission |
| HTTPS | HyperText Transfer protocol Secure |
| IDM | Identity Management |
| LDAP | Lightweight Directory Access Protocol |
| MD5 | Message Digest cryptographic algorithm |
| SMS | Short Message System |
| SSO | Single Sign On |
| USM | User Management Component |

# Requirements Overview

In a nutshell, the User Management (USM) component holds and provides information about users that may be used by business applications for both authentication and authorisation purposes as well as for user notification (e.g. sending an e-mail, or SMS when a business event occurs) and for providing an improved user experience (by e.g. using the user preferred language or number of results per page).

## Use Cases

This section summarises the architecturally significant Use Cases applicable to the User Management component.

### Actors & Packages



Figure 1: Requirements Overview - Actors & Packages

The actors interacting with User Management component are the following:

| **Actor** | **Type** | **Description** |
| --- | --- | --- |
| Application | System | A business application that consumes the services offered by the User Management component for user authentication, user authorisation, user notification and/or improved user experience. |
| Application User | Person, System | Either an end-user of a business application (man-machine interface) or a system interacting with a business application through a system-to-system interface. |
| User Manager | Person | An individual responsible for the management of the user related information maintained by the User Management component. |

The Use Cases applicable to the User Management component may be grouped in the following packages:

| **Package** | **Description** |
| --- | --- |
| Administration | The set of Use Cases related to the administration of information maintained by the User Management component. This information covers:   * **Users**: account, contact details, roles, profile (or preferences); * **Organisations**: name, description, contact persons; * **Roles**: sets of permissions (or rights) that may be assigned to users to grant them the right to use application features; * **Scopes (or scenarios)**: sets of data (held by applications) that may be assigned to users together with a role to grant them the right to access application data; * **Applications**: sets of features offered and data held to which access must be restricted, set of preferences a user may customise for an improved experience. |
| Authentication | The set of Use Cases related to the assertion of the identity of end-users willing to access an application. |
| Authorisation | The set of Use Cases related to granting (or denying) the usage of specific application features (on a specific data set) to an (authenticated) end-user. |
| Information | The set of Use Cases related to the provision of (user related) information to a business application. |
| Self-Service | The set of Use Cases that allow an Application User to access and modify his personal information. |

The following sections briefly describe the most significant Use Cases applicable to each actor.

### Application Actor

The *Application* actor is the primary actor of the User Management component. The *Application* actor is a system representing a business application that consumes one or more of the services offered by the User Management component for user authentication, user authorisation, user notification and/or improved user experience.



Figure 2: Use Cases - Application Actor

| **Use Case** | **Description** |
| --- | --- |
| authenticateUser | Asserts the identity of an end-user requesting access to the business application. Identify assertion is typically based on userName/password. |
| checkUserRight | Asserts that the user is authorised to use a specific application feature (on a specific data item). |
| confirmUserIdentify | Asserts (again) the identity of an end-user requesting to perform a (security) sensitive operation. Identify confirmation is typically based on a challenge/response. |
| getOrganisationInformation | Retrieves information about organisations and their respective end-points. |
| getUserProfile | Retrieves the user profile (or preferences) in order to customise the user experience. |
| getUserRights | Retrieves the set of features to which an end-user has been granted a right to use. Typically used to disable/hide application features for which the user is authorised. |
| getUserScopes | Retrieves the scopes (or scenarios) to which an end-user has been granted access. Typically used to disable/hide application features or data to which the user does not have access. |
| updateUserProfile | Updates (stores) the profile (or user preferences) of a user |

### Application User Actor

The *Application User* actor may be a) a person representing an end-user of a business application (man-machine interface) or b) a system interacting with a business application through a system-to-system interface.



Figure 3: Use Cases - Application User Actor

| **Use Case** | **Description** |
| --- | --- |
| changePassword | Changes the end-user password |
| isRemindedOnPasswordExpiry | The application user receives notification (via e-mail/SMS or at login time) that his/her password has (is about to) expired. |
| resetPassword | Triggers a reset of the user password (in case the password was lost, forgotten or stolen) |
| updateContactDetails | Updated the user contact details (phone, fax, e-mail) |
| updateSecurityQuestion | Updates the security question (challenge/response) used to confirm the user identity |
| updateUserProfile | Updates then user preferences for a specific application |

### User Manager Actor

The *User Manager* actor is a person representing an individual responsible for the management of the user related information maintained by the User Management component.



Figure 4: Use Cases - User Manager Actor

| **Use Case** | **Description** |
| --- | --- |
| copyUserProfile | Assigns a profile to a user by copying an existing profile. |
| manageApplication | Manages (create, update, delete) the (business) applications that rely upon the component, defining their features, data-sets and user preferences. |
| manageOrganisation | Manages the organisations to which users belong or relate. |
| manageRole | Manages (create, update, delete) the roles and their respective permissions. Assigns (or remove) roles to users. |
| managerUserAccount | Manages (create, update, enable, disable) user accounts, possibly importing user details from an existing user directory (e.g. LDAP).  Also manages the various policies applicable to user accounts (e.g. password strength, password expiry/history, account lockout) |
| manageScope | Manages (create, update, delete) the scopes (or scenarios) and assigns them to specific users. |

## Supplementary Requirements

This section summarises the architecturally significant non-functional requirements applicable to the User Management component.

|  |  |
| --- | --- |
| **ID** | **Description** |
| SR1 | The storage and transmission of user credentials (e.g. passwords) must be encrypted. |
| SR2 | It must be impossible for an administrator to recover the password of another user |
| SR3 | The communication with service consumers and end-user terminals must be based on the HTTPS protocol. |
| SR4 | It must be possible to integrate the User Management component with a Single-Sign-On (SSO) solutions like CAS (Central Authentication Service) or ECAS for user authentication. |
| SR5 | Changes made to user privileges (or permissions) must become effective immediately (i.e. without requiring a new user logon). |
| SR6 | The User Management component must be deployable both on an open-source platform (e.g. PostgreSQL RDBMS & JBoss Application Server) and on the platform used by the European Commission (Oracle RDBMS and WebLogic application server) . |

Table 1: Supplementary Requirements

# Component Design

## Offered Interfaces

The User Management component offers three service-level interfaces to business applications: an [**AuthenticationService**](#_Authentication_Service_Interface) provides operations for user authentication and identity confirmation; an [**AuthorisationService**](#_Authorisation_Service_Interface) provides operations for asserting the rights of an application end-user to use specific features; and an [**InformationService**](#_Information_Service_Interface) provides access to “personal” user information.



Figure 5: Component Design - Offered Interfaces

### Authentication Service Interface

The Authentication Service provides operations for user authentication and identity confirmation.



Figure 6: Offered Interfaces - Authentication Service

| **Operation** | **Description** |
| --- | --- |
| authenticateUser | Asserts the identity of a user based on the provided user identifier and password. |
| authenticateUser | Confirms the identity of a user based on the provided challenge and user response. |
| getUserChallenge | Retrieves a challenge suitable for confirming the identity of the user with the provided identifier. |

### Authorisation Service Interface

TheAuthorisation Service provides operations for asserting the rights of an application end-user to use specific features (on specific data items).



Figure 7: Offered Interfaces - Authorisation Service

| **Operation** | **Description** |
| --- | --- |
| chechUserRight | Asserts the right of an end-user to use a specific business application feature on a specific data item. |
| getUserRights | Retrieves the list of application features to which a user has been granted a right-to-use. Typically used to show/hide, enable/disable application features/options based on user rights. |
| getUserRoles | Retrieves the list of roles assigned to a user. Typically used to show/hide, enable/disable application features/options for applications using role-based authorisation. |
| getUserScopes | Retrieves the list of scopes (or scenarios) with which the user may interact. Typically used to show/hide, enable/disable application features/options based on user rights. |

Note: to address supplementary requirement [[SR5](#SR5)] the Authorisation Service will query the database at every service consumer request. Service consumers for which requirement [[SR5](#SR5)] does not apply should/could hence cache the information on the client to achieve a better throughput.

### Information Service Interface

The Information Service provides access to “personal” user information that may be used for providing an improved user experience or for out-of-band communication (via e.g. e-mail or SMS) with the user.



Figure 8: Offered Interfaces - Information Service

| **Operation** | **Description** |
| --- | --- |
| getContactDetails | Retrieves contact information about a specific application end-user. |
| getOrganisation | Retrieves information about an organisation and its associated end-points. |
| getUserProfile | Retrieves an end-user profile for a specific application. A user profile is a set of user preferences expressed as key-value pairs (e.g. language=en, linesPerPage=33, etc.) |
| updateUserProfile | Stores the (modified) user profile for a specific application. |

### Administration (User) Interface

The User Management component offers an (optional) web-based administration user interface for the management of the different entities managed by the component.



Figure 9: Administration (User) Interface

| **Component** | **Description** |
| --- | --- |
| Account Administration | Web Component for the consultation and administration of user accounts and their associated roles and scopes. |
| Application Administration | Web component for the definition and maintenance of the applications that rely on the component, their features, data sets and (user customisable) preferences. |
| Organisation Administration | Web Component for the consultation and administration of organisations and their associated end-points (or contact persons). |
| Policies Administration | Web Component for the definition and maintenance of security policies. |
| Profile Administration | Web Component that allows consulting user profiles and copying a profile from one user to another. |
| Role Administration | Web Component for the definition, consultation and maintenance of security roles and their associated permissions. |
| Scope Administration | Web Component for the definition, consultation and maintenance of scopes (or scenarios) and their association to Users. |

### Self-Service (User) Interface

The User Management component offers an (optional) web-based user interface that provides features to application end-users for the maintenance of their personal information.



Figure 10: Self-Service (User) Interface

| **Component** | **Description** |
| --- | --- |
| User Self Service | Web Component allowing the Application User actor to maintain his/her personal information (passwords, profile, contact details). |

## Consumed Interfaces

The User Management component will use the interface offered by the Auditing component to record actions (e.g. create, update or delete) performed or attempted by its users.

## Domain Model

This section presents the domain objects used by the User Management component.



Figure 11: Domain Model

| **Object** | **Description** |
| --- | --- |
| AccountPolicies | Holds the account related policies (lockout, concurrent usage) |
| Application | Holds identification information about a business application that uses the services of the User Management component |
| ContactDetails | Holds contact information about a user |
| DataSet | Holds identification information about a set of data managed by a business application. |
| EndPoint | Holds interconnection information about an information system operated by an organisation. An endpoint has the following attributes: name, URI, priority (main, fall-back, etc.), data source, contact details, status (enabled/disabled). |
| Feature | Holds identification information about a specific feature offered by a business application |
| Organisation | Holds identification information about an organisation. An organisation has the following attributes: name, nation, branch, status(enabled/disabled) |
| PasswordPolicies | Holds the password related policies (strength, history, renewal) . |
| Role | Holds identification information about a security role, the permissions to which the role grants a right-to-use. |
| Scope | Holds a right to use a specific set of the data (or scenario) managed by a business application |
| User | Holds identification and security information (password, roles, permissions) about a user |
| UserOption | Holds information about a user customisable property of a business application |
| UserPreference | Holds user preference for a specific customisable property of a business application |
| UserProfile | Holds a set of user preferences for a specific application. |

## Data Model

This section presents the persistent entities managed by the User Management component.

### Tables and Relationships

The below class diagram depicts the main database tables and their relationships.



Figure 12: Data Model - Overview

Note: To address supplementary requirements [[SR1](#SR1)] and [[SR2](#SR2)] passwords will not be persisted in the database; only a one-way hash (e.g. MD5) of the password will be stored in the database.

| **Table** | **Description** |
| --- | --- |
| APPLICATION\_T | Holds identification information about a business application (or technical component) that relies on the USM component for security purposes. An application has a name and a description. |
| CHALLENGE\_T | Holds challenge/response pairs used to confirm the identity of an authenticated user |
| DATASET\_T | Holds identification information for a specific data-set managed by a business application and to which access must be controlled. A data-set has a name (application unique) a free-ext description and a discriminator that identifies specific data items. |
| END\_POINT\_T | Holds interconnection information about an (external) information system. An endpoint has the following attributes: name, URI, priority (main, fall-back, etc.), data source, contact details, status (enabled/disabled). |
| FEATURE\_T | Holds identification for a business application feature. A feature has a name (application unique) and a (free-text) description. |
| OPTION\_T | Holds information about a user customisable option defined and supported by a business application. An option has a name (application unique), a type (e.g., number, string, date, y/n) a (free-text) description and a default value. |
| ORGANISATION\_T | Holds identification information about an organisation. An organisation has the following attributes: name, nation, branch, status(enabled/disabled) |
| PASSWORD\_HIST\_T | Holds the history user passwords (stored as MD5 hashes) |
| POLICY\_T | Holds the account and password policies. A policy has a subject (password or account), a name, a (free-text) description and a value. |
| PREFERENCE\_T | Holds a user preference for an application specific customisable option. |
| ROLE\_T | Holds identification information about a security role. A role has a name (application unique), a free-text description, and a status (enabled/disabled). |
| SCOPE\_DATASET\_T | Holds an association between a scope and a data-set. |
| SCOPE\_T | Holds a business application (data) scope (or scenario). A scope has a name (application unique) a free-text description, a validity period, a data period, a status (enabled/disabled) and a set of associated data-sets. |
| USER\_ROLE\_T | Holds an association between a user, a role and an optional scope. |
| USER\_T | Holds identification, authentication and contact information about a user. |

### Application Domain



Figure 13: Data Model - Application Domain

### Organisation Domain



Figure 14: Data Model - Organisation Domain

### User Domain



Figure 15: Data Model - User Domain

### Role Domain



Figure 16: Data Model - Role domain

### Scope Domain



Figure 17: Data Model - Scope Domain

### Authentication & Authorisation Views

For authentication and authorisation purposes, the below database views provide read-only access to users, roles, permissions and scopes that are currently active i.e. the user, role and scope entities are enabled and the validity period of the user (account) and the scope encompass the current system date.



Figure 18: Data Model - Authentication & Authorisation Views

## Component Model

The User Management component will itself use a component-based (internal) architecture, organising and distributing its features in a set of loosely coupled sub components.



Figure 19: Component Model

## Deployment Model

In this section we analyse the possible deployment models that address supplementary requirement [[SR4](#SR4)]: *It must be possible to integrate the User Management component with a Single-Sign-On (SSO) solutions like CAS (Central Authentication Service) or ECAS for user authentication.*

### Integration with ECAS / CAS + IDM

CAS is an open-source Single Sign On solution targeted primarily at web applications. CAS per-se does not provide Identity Management features and must hence be configured to query an existing (LDAP or database) user repository.

ECAS is the Single Sign On solution (based on CAS) used by the European Commission. It includes both an authentication server and an Identity Management repository.

In this configuration, a client application relies on (E)CAS for enforcing user authentication (on initial access) , the application recovers the user identifier from (E)CAS and uses this identifier to query the USM authorisation and information services.



Figure 20: Deployment Model - Integration with ECAS / CAS + IDM

Note: In this configuration, the USM DB does not need to store any authentication related information (login name, password). Furthermore it will not be possible for the User Management component to enforce password or account related policies.

### Integration with CAS

In environments where no Identity Management repository is readily available, CAS may be configured to use the USM DB using a (CAS provided) database authentication component.

In this configuration, a client application relies on CAS for enforcing user authentication (on initial access), the application recovers the user identifier from CAS and uses this identifier to query the USM authorisation and information services.



Figure 21: Deployment Model - Integration with CAS

Note: In this configuration, the USM DB must store authentication related information (login name, password).