Appendix 4
to Tender Specifications

Access and Identity Management Guide (abridged version)

EMSA

Table of Contents

Table of Figures

Definitions, acronyms and abbreviations

Definition	Description
AD	Microsoft Active Directory
BCF	Business Continuity Framework
CSN2	Clean Sea Net 2 Maritime application – version 2
EMSA	European Maritime Safety Agency
IdM	Identity Management which comprises Access and User Identity Management
IMDatE	Integrated Maritime Data Environment Maritime application
LDAP	Lightweight Directory Access Protocol
LRITDC	Long-Range Identification and Tracking Data Centre Maritime application
MarApps	Abbreviated form of referring to EMSA Maritime Applications
OAM	Oracle Access Management
OIM	Oracle Identity Management
OSB	Oracle Service Bus
OVD	Oracle Virtual Directory
RAC	Oracle Real time Application Cluster
RuleCheck	Application providing EU and International legislation regarding Port State Control
SAP	Webgate Specific Access Point configuration
SSN	Safe Sea Net Maritime application
SSO	Single Sign-On
STCW	Standards of Training Certification and Watchkeeping Maritime application
THETIS	The Hybrid European Targeting and Inspection System Maritime application
WebGate	Secured access entry point for applications

1. Introduction and objectives

This document describes EMSA Access and Identity Management. Its main purpose is to document the technical solutions used by EMSA to implement Access Control and User Identity Management throughout EMSA systems and applications.

It should be noted that this is an abridged version of the original document intended only for obtaining a high level perception of EMSA Access and Identity Management.

The document is organized in several chapters:

- Chapter 1: Introduction and objectives. This chapter;
- Chapter 2: EMSA Requirements for IdM. A quick specification of the requirements for implementing the IdM at EMSA.
- Chapter 3: Oracle's Overview of IdM Solution. Herein is defined Oracle Corporation's vision of the IdM implemented at EMSA.
- Chapter 4: Protecting Applications. An explanation on how applications can be protected by Oracle Access Management through the use of a WebGate and Access Policies.
- Chapter 5: Provisioning Applications. An explanation on how application users can be provisioned by Oracle Identity Management.
- Chapter 6: EMSA OIM Custom Interface. Brief description of the adopted custom interface.
- Chapter 7: Password Management. A description on the mechanisms behind the self-service functionality of changing/recovering lost passwords.
- Chapter 8: Security Model. An explanation on how privileges are established within IdM for such actions as user creation/modification.

2. EMSA Requirements for IdM

The present text aims at documenting the initial requirements for EMSA Identity Management / Single Sign-On (SSO) process.

Infrastructure

- High Availability
  - The final solution has to be compatible with the High Availability (HA) infrastructure at EMSA, and be run in an Active-Active Weblogic Cluster. This is true for both the OAM and OIM applications (as well as any solutions upon which these systems may depend – as described below).
  - The solution should also be compatible with BCF (Business Continuity Framework) to be implemented at EMSA.
  - Any database needed by the OAM and/or OIM should be RAC compatible as this is the HA solution adopted at EMSA for a database infrastructure.
  - Any other servers (Apache, LDAP, etc.) also have to be "cluster capable" or at least provide an HA solution.

- Hardware architecture – The complete hardware architecture will be mentioned at a later time.

- Oracle Service Bus (OSB) is also a part of the overall software architecture at EMSA and can also be considered in the final solution.

- In order for applications to be able to see all LDAP servers (AD, Open LDAP, etc.) in a similar coherent way, Oracle Virtual Directory (OVD) should be used to create a virtualization layer.

- Load Balancing
  - The final solution has to be compatible with the existing EMSA infrastructure as far as load balancing is concerned. This is presently done with F5 load balancers for external accesses and Apache Servers for internal accesses.

- The solution should be integrated with an SMTP Server to handle all mail related issues.

- The solution should be prepared to integrate cleanly with a Syslog Server for log auditing.

IdM

- OAM
  - This software has to provide all the functionalities for duly managing the authentication of users accessing EMSA applications (those which are covered under the IdM/SSO "umbrella").
  - Due to the fact that EMSA has applications that are not directly deployed under Liferay as well as Liferay deployed applications, there is a need for three types of login interception/treatment:
    - Applications that are not deployed under Liferay should be intercepted by the Apache Reverse Proxy and the authentication process should correspond to the creation of an SSO session token that then needs to be validated inside the application.
    - Anonymous (or Guest) access to public pages in Liferay should not be authenticated. However, there should be a link available in the guests' home page allowing a login to the system. This login should then be serviced by the SSO code in the normal manner.

        - Users directly accessing an application that is directly deployed in Liferay should be presented a login page (go through the regular authentication process) if they have not already established a session. This will probably be the most common case and corresponds to the base tests made during the PoC.
    - At EMSA, there are two main classes of applications, those which run under Weblogic and those that do not. For those that do not run under Weblogic, a filter for treating the SSO session token must be developed for inclusion in such applications. Depending on the technology involved, this may or may not have to be done by the application contractor. For those applications that do run under Weblogic, a new Realm Provider has to be built that will obtain its information directly from the SSO session token instead of connecting to LDAP to obtain such information.

- OIM
    - This software has to provide all the necessary functionalities for provisioning users/user data in all EMSA applications (where applicable). As such, each specific "connection" has to be analysed and a specific connector has to be used / implemented (i.e. Liferay provisioning is done through a Web Service, LDAP is done through a specific connector, etc.).
    - Provisioning
        - Due to the fact that there are multiple applications to be deployed in the IdM / SSO scenario, there is no one standard way to provision users / user data to all the applications. To try to obtain maximum uniformity and maximum benefits from the tool, the provisioning models should follow a Role Base Access Control approach (RBAC).[1] Each application will have to indicate the exact necessities for provisioning as well as specify the technological means through which this provisioning is to be done. The reason why such an approach is taken also has to be documented (for example, provisioning should be done through a Web Service instead of a direct access to a database table because of second level caching in the data access layer of the application).
    - Self Service Interface
        - Work flow approvals
        - Delegated Administration
    - Auditing
        - Logs
        - Reporting
    - Reconciliation with Authoritive sources

Integration

- Liferay
    - General user management should be blocked from Liferay because user provisioning is to be done by OIM. Specific user management details may still be done directly in Liferay but only by users with Liferay administration rights.
    - User first access challenge has to be disabled or alternatively the provision for this challenge has to be done in OAM / OIM.
    - Automatic importing of users / user data from LDAP has to be disabled as the user provisioning is solely to be done through OIM.
    - Even though Liferay is a "special" application (because it is a portal offering services to other applications), it should be considered as a separate application and have a specific LDAP group. This will allow a more uniform view of user provisioning amongst all applications.

[1] The first noticeable exception to the RBAC model is with the SSN (Safe Sea Net) application. They do not use an RBAC model so a "high level" access to SSN "Applications" is to be done instead.

- Up until now the Community Version of Liferay has been used but in the production environment the Enterprise version will be used. This means that the IdM / SSO will need to be compatible with this latter referred version.[2]

- Other
  - Each application / system is going to have to be analysed on a case-by-case basis to see what should be changed / removed / added. Without further study of each application / system to be included it is not possible to indicate which changes should be made.

The following diagram provides a "block view" of the possible architecture. In this diagram it is possible to see that all accesses are made through the Apache Server and WebGate module (acting as a reverse proxy). From here, if users are already authenticated, they may be permitted to access the web applications (Apache + WebGate -> Web Applications). If the users are not yet authenticated, they will be shown a Login Form from OAM for authenticating (Apache + WebGate -> OAM). After the users submit their credentials, these will be verified by OAM on the LDAP virtualization layer (OAM -> OVD) and if they are correct, a Session Token will be generated and returned to Apache for inclusion in all subsequent requests. Apache then redirects the user to the original URL requested. This authentication mechanism is used for all accesses that go through the Apache reverse proxy.

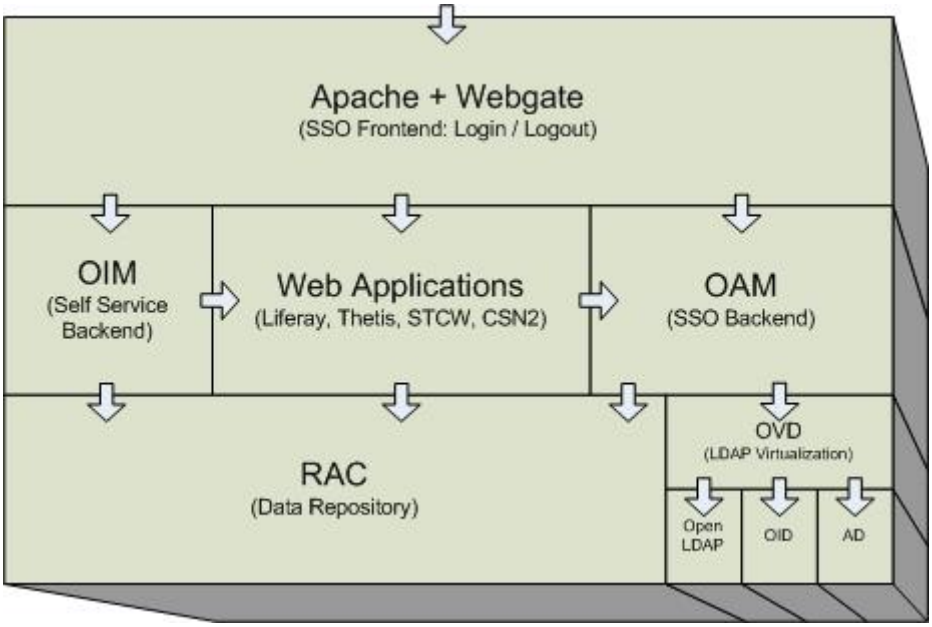

Figure 1: Block Diagram

In the event that the URL requested is part of the OIM self-service (Apache + WebGate -> OIM), there is a guarantee that users have already been authenticated and the corresponding functionality will be accessed. Depending on the action requested, OIM may do provisioning work through a service interface (OIM -> Web Application) or directly to an application's database (OIM -> RAC).[3]
If the URL requested corresponded to a web application (Apache + WebGate -> Web Applications), then the respective application may request Authorization information from OAM (Web Applications -> OAM). The exact process through which this is done will depend upon the application being deployed on Weblogic (in which case the request should be done

[2] Liferay Enterprise Edition is being used in almost all environments at EMSA (Training is still using the Community Edition).
[3] Until the present moment, all provisioning is done through Web Services. There is no direct access to any application database.

through JAAS) or, if the application is not deployed on Weblogic (non-java application); a call to the OAM API through custom code will need to be done. Please note that it may be possible for applications to access the Virtual LDAP layer instead of the OAM API but this is still an open issue.

The following diagram depicts the same information as the previous diagram, but through a more logical viewpoint. The machines depicted are purely "logical" and may not correspond to actual physical machines (they may be single, clustered or joined together depending on actual implementation constraints).

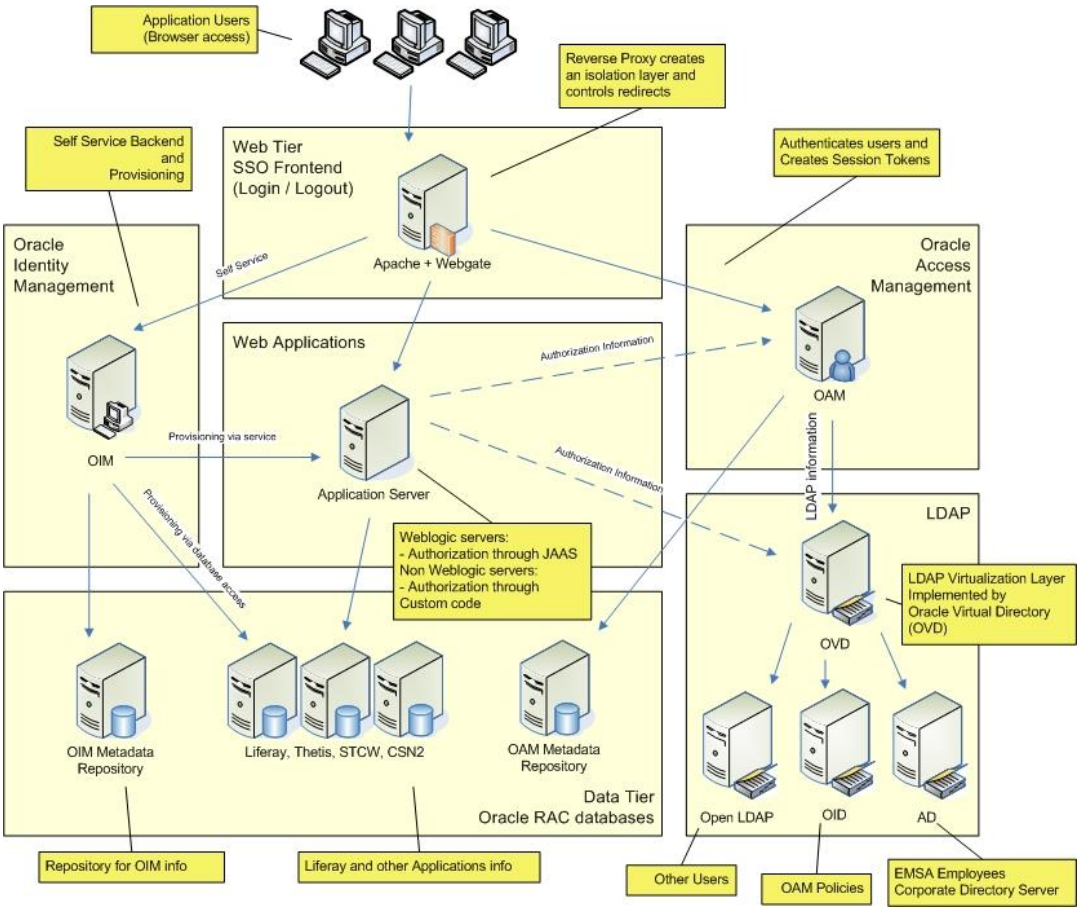

Figure 2: Logical Overview

3. Oracle's Overview of IdM Solution

This chapter documents Oracle's view of how the IdM solution could be implemented at EMSA. Please note that the information provided here may not correspond exactly to what was implemented. Oracle proposed solution was taken as the basis for the IdM design and architecture but other technical solutions were also adopted.

3.1. AUTHENTICATION AND AUTHORIZATION

EMSA applications that require user authentication and authorization should rely on a directory to store user credentials, roles and access privileges.

User directory technologies
• OpenLDAP • Application Server embedded LDAP

Although the use of a database schema to cope with these functions is a common practice, it has several disadvantages and should be avoided.

It is envisaged that over time all existing applications will be using this new SSO mechanism. For new application development, developers should focus on:

- Relying on SSO for authentication
- Using JAAS for in-app authorization
- Weblogic App Server needs to be configured accordingly (JAAS + OAM agent)
- Use an RBAC model
- All administration of security principals will be handled through the Oracle Identity Manager.

The following schemas give an overview of the current SSO implementation.

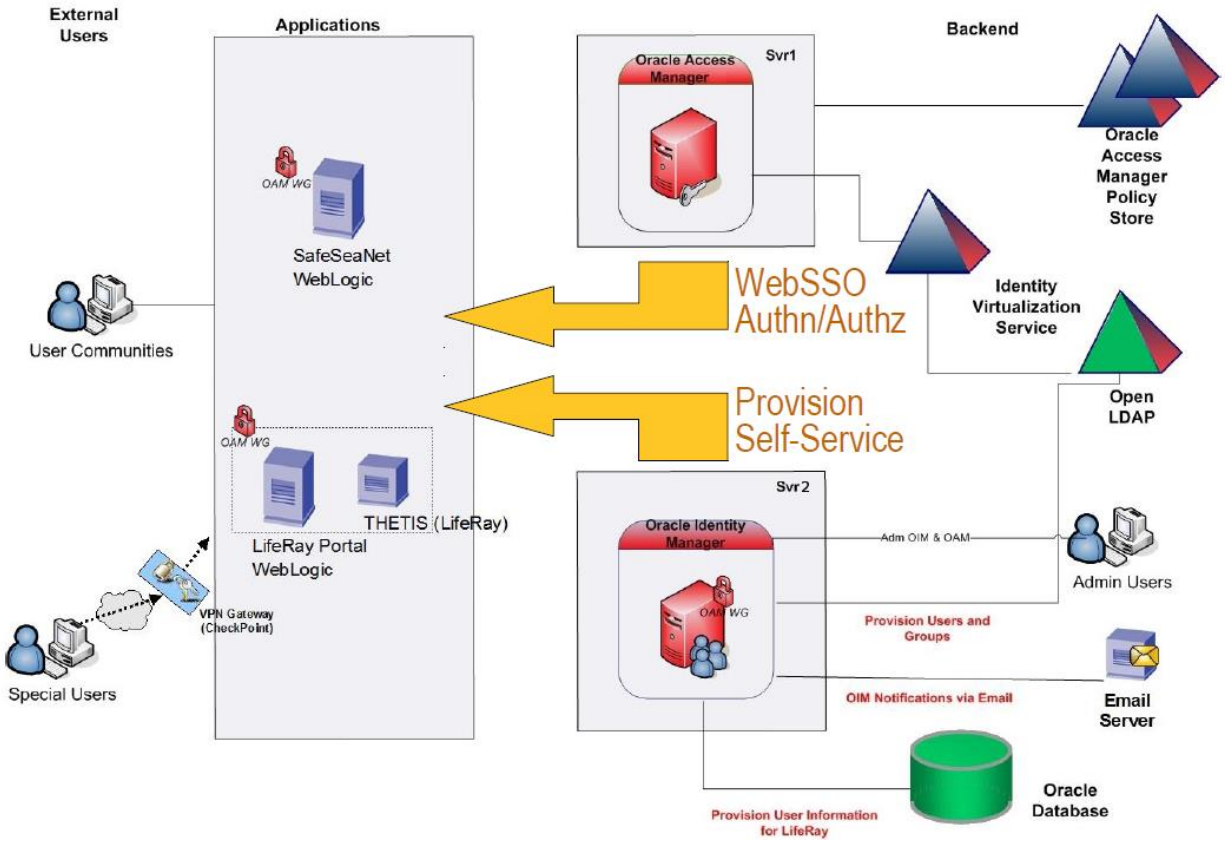

Figure 3: SSO high level diagram

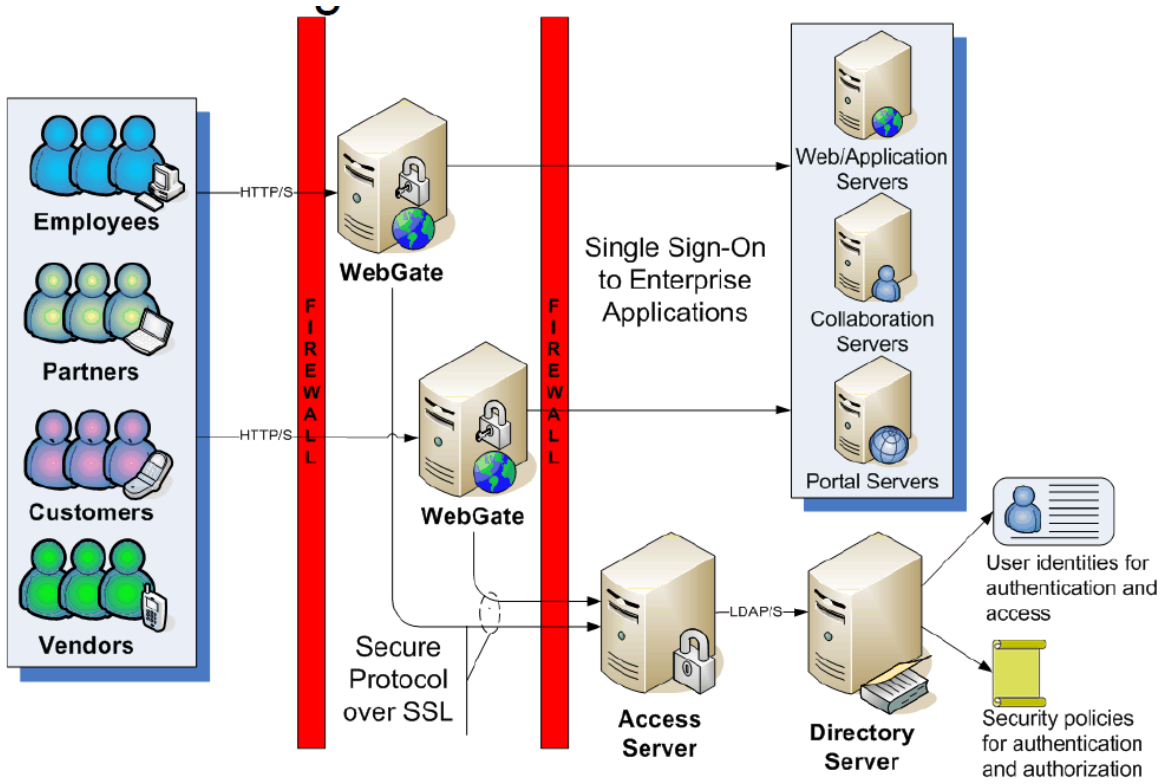

Figure 4: Access Manager Architecture

3.2. ADDITIONAL INFORMATION ON THE ORACLE PRODUCTS

The EMSA IdM solution uses a series of Oracle Products: Oracle Access manager, Oracle Identity Management, Oracle Internet Directory and Oracle Virtual Directory. Further information on any of these products may be found by following their respective links.

Oracle Access Manager 10gR3 (10.1.4.3.0)
http://www.oracle.com/technetwork/middleware/id-mgmt/index-090417.html?ssSourceSiteId=ocomen

Oracle Identity Management 10gR3
http://www.oracle.com/technetwork/middleware/id-mgmt/overview/index-098451.html?ssSourceSiteId=ocomen

Oracle Internet Directory 11g R1 (11.1.1.3 – 11.1.1.5)
http://www.oracle.com/technetwork/middleware/id-mgmt/overview/index-082035.html?ssSourceSiteId=ocomen

Oracle Virtual Directory 11g R1 (11.1.1.3 - 11.1.1.5)
http://www.oracle.com/technetwork/middleware/id-mgmt/index-093158.html?ssSourceSiteId=ocomen

4. Protecting Applications

EMSA hosts a number of Maritime Applications (MarApps), most of which deal with sensitive information that needs to be protected and or restricted. To reach this goal the MarApps have a series of protective layers.

The first layer of protection is provided through the IdM Single Sign-On (SSO) mechanism which only allows access to pre identified persons.

A second layer could be implemented through the OAM Access Policies only allowing access to specific URL's when users belong to specific LDAP groups.

Any layers from this point onward can be considered as application dependent and must be implemented inside the respective applications (i.e. application roles and/or specific business functionality access permissions).

This document only considers the first two layers leaving the other layers to each individual MarApp. It is worth mentioning that the second layer is not currently used to its full potential.

4.1. AUTHENTICATION

The general concept of Authentication can be defined as "the process of determining whether someone or something is, in fact, who or what it is declared to be". Whilst other definitions are possible, this is the one that most relates to EMSA's first layer of protection to the MarApps.

The process of authenticating a given person (henceforth referred to as a "user" of the MarApps) is achieved by presenting a place for the user to present his credentials (providing a "user identity" and a password) and then validating the information provided against a repository of known and allowed credentials. This process is achieved in EMSA by Oracle Access Manager (OAM) validating the credentials against EMSA's LDAP.

Correctly authenticated users are allowed access to the next layers of protection while unauthenticated users remain "stuck" at this first level or layer.

At EMSA, due to the SSO implementation, the user will only be confronted to give his credentials once per session though he will have to pass through the authentication / authorisation process on each request, albeit transparent to him.

4.2. AUTHORISATION

Once a user passes the first layer of protection, i.e. was authenticated, he is subject to the second layer of protection which will only allow the user to access resources (URL's) associated to LDAP groups to which he belongs (see Access Policies in the following Oracle Access Manager chapter). At this point any attempt to access a resource to which the user has not been granted permission will result in an error page being shown indicating that the user does not have permission to access the resource (see following Figure).

Through this extended use of OAM (namely the ability to restrict access to predefined resources (URLs) based upon membership of different LDAP groups), access rights similar to application roles could be enforced without the need for the actual MarApp to implement anything. This mechanism provides a very flexible way of implementing application roles because there is no need to change the application whenever specific access rules change.

There is however the need to update configurations inside OAM but this is always much simpler and cheaper time-wise than updating code.

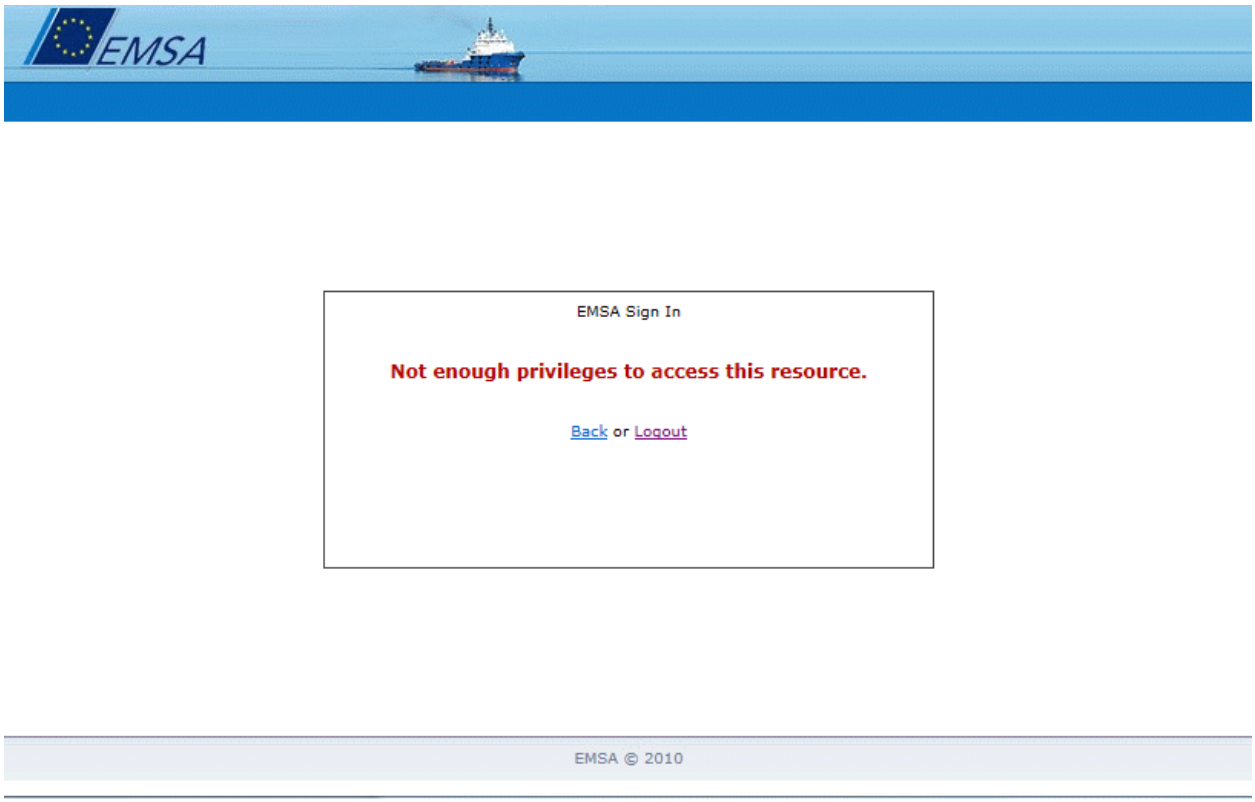

Figure 5: Authorisation denied

Attempts to access resources to which the user has been authorised to do so will result in a transparent intervention from OAM, i.e. nothing specific to OAM will be seen, so the user will not even be aware of existence of the protection layer.

4.3. WEBGATE

Important Note: One very important aspect in EMSA's SSO solution is that only web accesses are considered, i.e. http(s) requests. All other means of access to the EMSA MarApps infrastructure (T3, RMI, etc.) are effectively not protected by this particular solution.

To enforce the previously mentioned access technology restriction all protected communication from the MarApps interface (typically a web browser) must go through a proxy/reverse proxy that enforces the first two layers of protection.

In the Oracle technology stack used at EMSA, the proxy/reverse proxy component is called a WebGate (sometimes also referred to as an AccessGate) and is composed of an Apache HTTP Server with, amongst others, Oracle specific modules for communicating/interacting with OAM (obWebgateModule). To obtain a higher degree of service availability various Apache HTTP server instances are running at the same time. We call each instance a SAP (Specific Access Point) and will return to this subject shortly, but in the meantime, and before going any further, we would like to remind you that EMSA has three environments that are subject to SSO, i.e. Test environment, Pre-Prod environment and Production environment. Basically this means that any/all configuration done in one environment will eventually be rolled out to all other environments. If we add the fact that there are currently nine MarApps being accessed through SSO, the total number of configurations needed makes maintenance a head-ache. To ease this problem the following architecture has been devised.

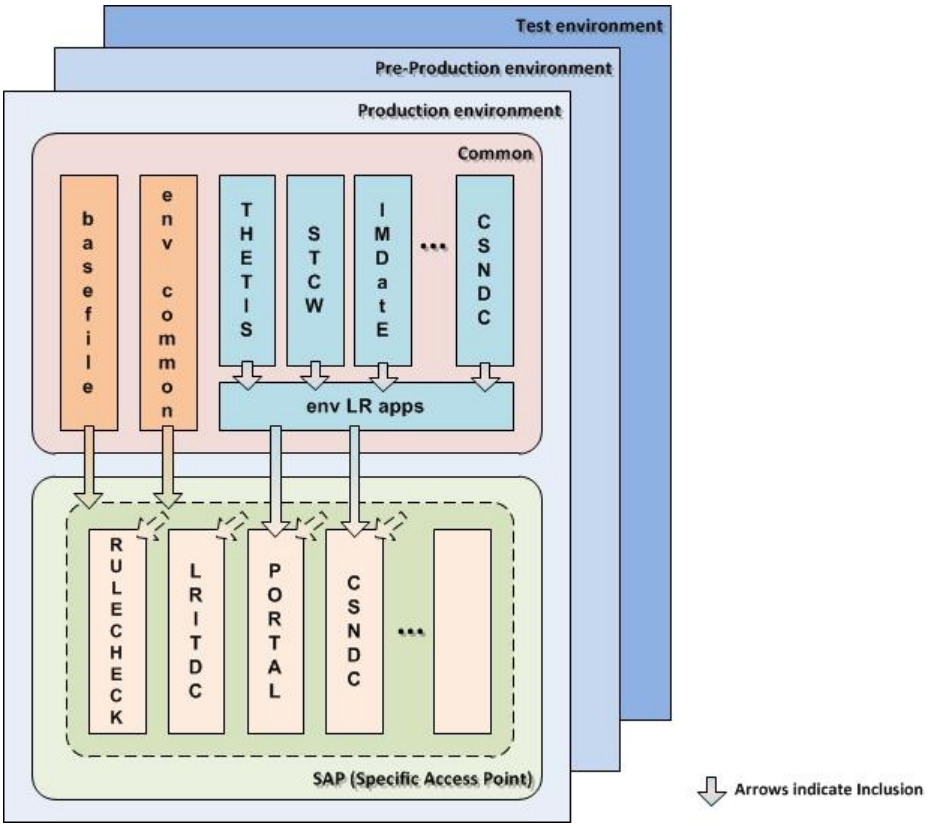

Figure 6: WebGate Configuration Architecture

From observing the previous figure we can see that in each of the three environments, there are two separate sections: the common configurations section and an SAP (Specific Access Point) configurations section.

The common configuration section is defined only once per environment whilst there may be (are) multiple SAPs per environment (not necessarily the same ones in all environments).

4.3.1. SAP Configurations

There have already been a few mentions to a SAP (Specific Access Point) in previous sections of this document but, exactly what is a SAP?

EMSA provides various MarApps to the user community. Some of these are stand-alone apps and some are integrated inside an enterprise portal (Liferay Portal), but all MarApps are web based and thus have a specific URL for being accessed. The unique URL base is what we call a SAP.

Currently in the production environment there are four SAP (excluding SSN). Each SAP has its own instance of an Apache HTTP server running (on each physical machine node because they can be/are clustered). This means that at any given time maintenance can be performed on one SAP while all others are still available/running. Whenever applications share a common access point, i.e. MarApps that are deployed in the Liferay Portal, interventions done to that SAP will obviously affect all of those other applications.

The advantages of having SAP are:

- As already mentioned, avoiding unavailability of non-related access points;
- Greatly reducing the amount of work necessary to maintain WebGate configurations by maintaining logical aggregations.

4.3.2. Common Configurations

After having extensively analysed all of the configuration files for all MarApps in all environments, a common set of attributes/definitions was identified. To ease the maintenance burden, all of the common values were brought together into a single file and explicitly included in each SAP configuration file. Furthermore, each SAP file sets various "*variables*" that are referred to in the common files. This mechanism allows for the maximum re-use of configurations not only across different SAP but also across different environments as well.

Further details can be found in the complete un-abridged version of this document.

4.4. ORACLE ACCESS MANAGER

Earlier in this chapter mention has been made to authentication of users and authorisation of accessing resources (URLs). The Webgate has been mentioned as being the filtering point for both authentication and authorisation. While this is true, the Webgate is not the system component that actually implements both of these functionalities. What it really does is, for each request, question the Oracle Access Manager (OAM) to see if the user is correctly authenticated and if he is authorised to access the resource. If so the proxy/reverse proxy rules are applied. If not the user is redirected to a specific page indicating that access rights are denied (if not authorised) or to the login page (if not yet authenticated).

4.4.1. Access Policies

At EMSA, we use the term "Access Policy" to describe the set of configurations needed by OAM to validate access to a specific resource.

Policy Domains

A top-down view of OAM shows the Policy Domains to be the highest level of the configuration structure. Each Policy Domain is a logical aggregator of a set of rules that can be applied to a set of resources (definitions on each of these terms follows). It facilitates management by allowing us to focus on a specific set of logically related rules/resources while permitting the high level operations of Enabling and Disabling the rules/resources, all at the same time.

At EMSA, the typical policy domains are the MarApps related policy domains (CSNDC, THETIS, STCW, IMDatE, LRITDC, etc.), a Liferay Portal related policy domain and IdM related (both OIM and OAM) policy domains.

Resources

The word resource has come up a lot in this document and it has always been associated with URLs. It is not too farfetched to say that there is an (almost) direct relation between the Resources configured in OAM and the proxy pass rules defined in the Webgate.

Authorisation Rules

An authorisation rule is, as the name implies, a set of rules that define the conditions under which authorisation is granted.

Policies

This is where everything previously mentioned comes together (and is the inspiration for EMSA's nomenclature of "Access Policies"). In a nutshell, this is where the Resources for the policy domain are grouped together with specific authorisation rules.

Examples of policies for a given MarApp can be "Public URLs" and "Private URLs". The resources associated with the Public policy are typically a welcome page in non-portal applications or public portlets in Liferay portal supported applications. Access grants for these types of policies are typically "Allow All".

Associated to a Private policy, we will find resources that are of a more sensitive nature therefore needing protection. With these policies an authorisation scheme is normally used as is an authentication rule.

4.5. RBAC IMPLEMENTATION IN THE EMSA INFRASTRUCTURE

In the scope of the Single Sign On / Identity Management project, we can state that all the applications to be considered are Web Applications. Furthermore, we can also state that all these web applications are to be (ideally) run under a common "umbrella" which is a Portal environment which will run on Weblogic JEE (Java Enterprise Edition) Application Servers. The Portal environment used at EMSA is based upon the Liferay Enterprise Portal implementation. An LDAP Server supports both the Portal as well as the web applications.

We will now describe how each piece of infrastructure implements/uses the previously mentioned RBAC concepts (basic definitions and relations).

4.5.1. LDAP

An LDAP server allows for the creation of a tree structure of Distinguished Names; DN's in LDAP terminology. It does not directly implement the notion of User Groups or Roles (or even Users for that matter). However, through the use of the DN syntax, one can just about map anything inside the LDAP tree structure. Roles and User Groups can be obtained by associating specific attributes to a DN (whose direct meanings can be interpreted as a Role or User Group) or they can be obtained by answering questions like "in what groups X is a member of" for Roles or "who are the members of that group" for User Groups.

The semantics of use of LDAP at EMSA are:

- The "top level" of the structure having beneath it:
  - The **groups** concept, under which will exist the representation of specific applications (or parts of and extensions to applications).
    - Inside (or underneath) a specific application group should come the actual names of meaningful groups.
  - The **users** concept, under which the **users** branch, two organizational units are possible:
    - Inside the **people** branch are all the physical application users
    - Inside the **system** branch are the system administrators or external systems

- Due to the fact that the concept of a role is not directly implemented in the EMSA semantics, such a concept should be achieved by associating users to groups through the **member** attribute. By using the first question previously described ("in what groups X is a member of"), one can conclude that in this way it is possible to infer **roles** from this structure (assuming that the name of the role is the same as the name of the group for ease of use). The only "restriction" applied here is that the name of the role be the same as the name of the LDAP group supporting the role.
- Applications that require only global authentication should create a group named **members** under the applications own group name and then associate the actual users with this group.
- Applications that need to implement role authorizations should associate the users with the name of the group that represents the desired role.

4.5.2. Liferay Enterprise Portal

The Liferay portal implements the following concepts: Communities, User Groups, Roles and Users. Likewise, the portal implements the concept of a page which we will consider as a resource in our RBAC model (or Functionality if you like). We will now have a look at each individual concept and discuss it in more detail.

- Users – In Liferay, a User represents a person and has a set of attributes. While it is possible to directly associate Permissions to Users, it is highly recommended not to do so as there are other ways to allow access to resources. There is a "one-to-one" relation between the users in Liferay and the users created in LDAP (even though it is possible for users to exist on only one of either side of the relation).
- User Groups – As the name suggests, this is an aggregator for joining Users. It allows a means for performing some operations on a variable number of users without having to do the same actions on each user individually. Whilst it is possible to assign Permissions to User Groups, as it was for users, this should also not be done. Like the relation between Liferay Users and LDAP users, there is also a "one-to-one" relation between Liferay User Groups and LDAP groups.
- Roles – A role is a way through which Liferay will grant user access to certain resources. A role is logically connected to a User Group (by associating the User Group to the Role) and should maintain a similar name to facilitate human reading/interpretation. This means that any User belonging to the User Group associated with the Role will have access to the resource protected by the Role. In this particular case, there is no direct connection between a Liferay Role and LDAP even though a logical association may be made through the similarity in the names.
- Communities – In Liferay, a Community is created to allow various Pages (we have called them resources in previous bullets and they are the Functionalities in the RBAC model) to be joined together thus providing a single point of configuration for a specific interest. Whenever access restrictions need to be applied (such as in the private pages of a community), Roles can be associated to a Functionality (Page) in a Community.

We have defined some basic concepts on the RBAC model. We have also explained how this model fits into the EMSA infrastructure. The next section will be about defining the requirements for provisioning users in the EMSA infrastructure for the Maritime applications.

4.6. DEPLOYING APPLICATIONS WITH SINGLE SIGN-ON

Most of the EMSA applications are not prepared to be integrated with IdM and may need some changes. This chapter documents how these changes can be done by using a "generic" application such as the Java Pet Store reference application as a "Guiney pig".

4.6.1. jPetStore

In the EMSA test environment, a well-known reference application – the Java Pet Store – has been deployed that allows for investigation and development of the Single Sign-On solution. One of the goals of deploying such an application in this environment was to assess the difficulties involved in adapting a web application to the Single Sign-On system.

Before going into the details of the necessary changes, we will first explain how the "normal" (unchanged) application works. The Java Pet Store application simulates an on-line shop for selling animals. There is "public" access to the application in which you can browse the existing information and you can even put items into a "shopping cart". If you decide to checkout your order, containing items in the shopping cart, you will have to log-in to the application to be identified. Only users that have been previously registered (provisioned) to the application may checkout orders. Likewise, if you wish to change your user attributes (password, address, phone, etc.) you must also be logged in.

Pre-emptive Authentication

A first interesting approach, while still not the desired one because of not fulfilling the previous "public user" functional requirements, will allow us to demonstrate how to perform authentication through Single Sign-On with minimum changes to the application. In this first approach, the whole application has been registered as "protected" in OAM (Oracle Access Management). This has the effect of the user/password being requested even before the first screen of the application is shown. After the initial logging in to OAM, there is no further need for identifying the user. If a user had already been authenticated in OAM prior to accessing any application screen, he will not be prompted to do so again (Single Sign-On). Please note that the only noticeable change in the application is the fact that the login form is never shown to the user.

Technical Considerations

We have indicated that the jPetStore application is now performing Single Sign-On with minimal changes to the application. We will now proceed to explain the actual changes made.

Three URLs were intercepted (the signonForm, the checkout and the editAccountForm). All three of these URLs have now been internally (internal to the server) redirected to the sign-on URL with additional parameters for the username and password. There are two comments to be made about this URL: first – it is always just internal to the server so there is no problem in sending the username and password as http GET parameters because the internal redirection can never be intercepted, and second – due to the fact that the user's password is never known outside of OAM, we need either to pass the username twice (serving as password) or pass a constant dummy password. This has to be consistent with the provisioning process followed.

Public and Private access to the application

As we have previously stated, the pre-emptive authentication scheme is not our target. As such, we now need to make some changes to the OAM to be able to comply completely with the full functional requirements. It is important to point out that there will not be the need to make any more changes to the jPetStore application, but the previously performed changes are still necessary for this stage.

Technical considerations for granting public access

As a result of the previous section, the jPetStore application is a protected resource which will require user authentication to be accessed. However, the functional requirements state that there is a part of the application that has public access.

In Oracle Access Manager, access the Policy Manager Application. Under the "Private URLs" policy domain, we will add another policy to the ones already existing in this domain. We have called this new Policy "JPetStore Public" and it consists of an http policy on GETs and POSTs, for all resources and all host identifiers, with the "/jpetstore/…/*" URL pattern.

In order for this policy to work correctly, the "Authentication Rule" associated to it must be that of "Anonymous Authentication" without any specific "Actions".

Once these changes are made no more user authentication is needed to access the application. There should now be no Authentication Form presented to the user whenever he accesses the jPetStore application, whatever the operation performed within. This,

however, is not what is intended as the user will now have to perform an application login (answering to an application login form – not the OAM one) whenever we tries to access the "private" area of the application (accessing the user account or checking out an order).

Final notes on configuration

Due to our current limited knowledge of the Oracle products, we have had the need to configure another policy, exactly the same as the previously mentioned "JPetStore Public", associated to the public URL for the application "/jpetstore".

4.7. LOGGING OUT OF SINGLE SIGN-ON

A first hand premise of SSO is that once a user is authenticated (in any given session), he will be able to access any EMSA Maritime Application to which he is authorised to do so, this without having to re-authenticate himself. The EMSA MarApps have to be prepared for the integration with OAM to allow automatically signing in a user and thus achieving an SSO solution.

One often overlooked aspect of an SSO system is that of logging out. Under the assumption that a valid session is in place, a user accessing an application that he has access rights to, will be automatically able to see the respective application (without having to present his credentials again – remember there is a valid session). When a user decides that he does not want to continue accessing a given application, he would normally "logout" from that given application and continues to use any other application that he so wishes. However, due to the automatic nature of SSO, whenever the user re-accesses the original application from which he previously logged out, he will be automatically logged in (due to SSO) and will be given the perception that he effectively never logged out. In practical terms this means that the operation of logging out is superfluous unless it is applied to ALL applications that the user was accessing under the current session.

If a global logout solution is not applied, a user can, at any time, simply close a browser tab without actually logging out of an application as the end result is the same as actually logging out and being automatically logged in again. Please note however that if closing a browser tab results in the end of a browser session (if the tab is the only one open on the browser and no other browser windows are open, for example), then the user will have to log in again if not for any other reason that the browser will not use the same session again when it's re-opened. This is a situation which the user should avoid as the session may still be active in the applications and be subject to session hijacking.

EMSA has chosen to implement a "Single Sign-Out" precisely for the previously mentioned reason of logouts, on their own, being superfluous.

4.7.1. Technical implementation of a global Logout

The implementation done at EMSA is that of once a logout URL is selected (from any of the SSO integrated applications), OAM will intercept the call and start a process of invoking the logout URLs of all the applications to which the user has accessed (been logged in to). After all of the application logout URLs have been invoked, OAM will proceed to terminate its own session, thus effectively logging the user out in a safe way.

Each Maritime application that has been integrated with SSO should be prepared to logout correctly upon request.

One final consideration associated to each application needs to be assessed and that is the existence of a logout URL for the application.

5. Provisioning Applications

This chapter aims at providing an overview of the user provisioning process for applications deployed at EMSA. It describes a generic tier model that shows how parts of provisioning can be done at different levels and it also describes two possible work-flows that can be applied to the provisioning process.

5.1. PROVISIONING TIER MODEL

One of the goals of having an Identity Management solution in EMSA is to have a common way (as much as possible) of provisioning users to applications. This essentially means that whatever can be found common to all possible applications should be done in a single point in Identity Manager leaving any particularities up to the specific applications that need these particularities. These particularities can be done in custom forms inside Identity Manager and/or directly inside the applications. This approach leads us to a three tier provisioning model that is depicted in the following diagram.

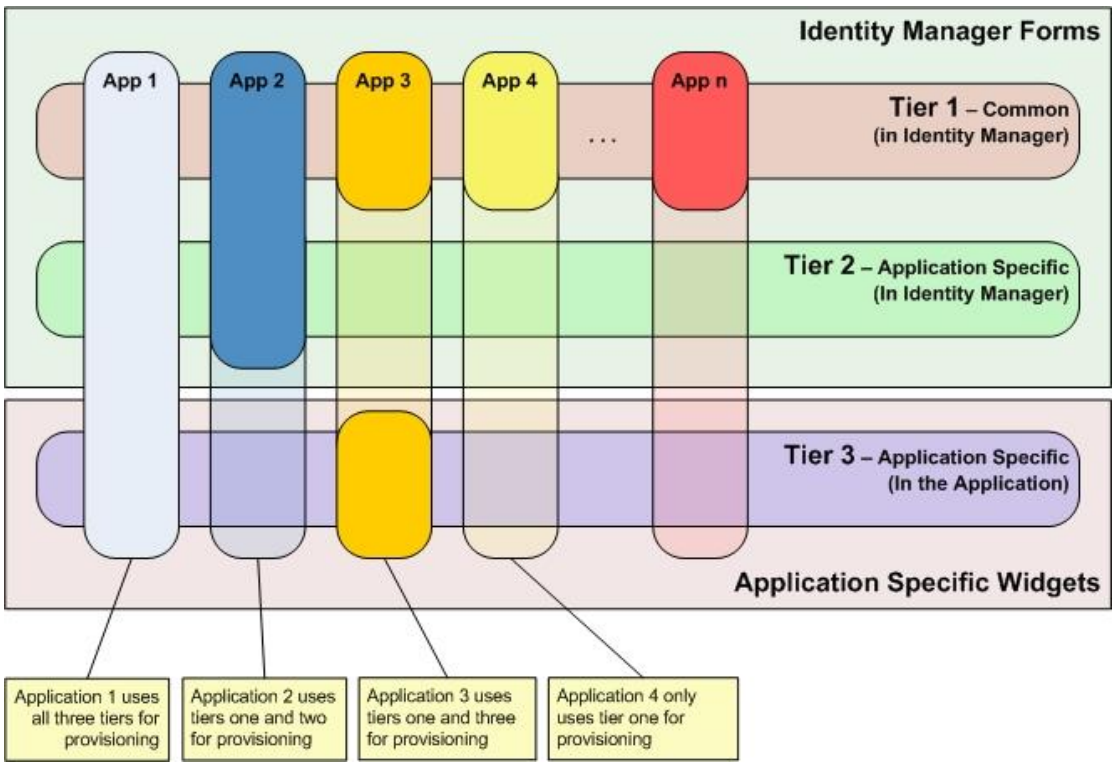

Figure 7: Tier Model for Provisioning

In the previous figure, the first tier corresponds to the Identity Manager forms that will be filled with the data that is common to (most of) the applications. Due to the fact that all users accessing EMSA applications need to exist in an LDAP server (or any other form of data storage that can be seen as such), the basic LDAP provisioning will be done automatically by this first tier. Depending on the work-flow involved (work-flows are seen later in this document), and also on the various applications' specific needs, parts of this data may be replicated to other forms in Identity Manager that are application specific. Likewise, further information can be added to LDAP by the following tiers if such is needed. This leads up the next tier in the model.

The second tier might be, in some cases, an optional tier depending on each specific application. Whenever an application needs certain information that is not part of the common layer (first tier in this model) but is still sufficiently application independent as far as the user interface is concerned (i.e. does not need a specific custom user interface to

obtain the data), then this information can be filled in through custom forms still inside the Identity Manager. At this point all of the common data is available to the forms from the previously filled in first layer.

It is convenient to point out that for the data from both tier 1 (common) and tier 2 (application specific), to be passed on to the applications, these need to provide specific services to be invoked from Identity Manager. These services should be as much as possible based upon normalized, secure standards (for example Web Services) that can use existing standard Identity Manager Adapters. Any form of service that does not comply with a standard adapter may have the added overhead of having to develop an adapter for use in Identity Manager.

The third tier, which like the second tier, may or may not be necessary, is completely application specific. In this tier, the user interface for gathering the provisioning information is to be done with native application widgets (there can be no use of Identity Manager – otherwise it would still be tier two and not three). This tier, if needed, will most probably correspond to the most complex user interface and most complex business logic for the provisioning of a user in the application (once again, otherwise it could be done in one of the other two tiers). The implementation of this tier should be done in such a manner that it could be seamlessly (at least apparently from the users' point of view) called from the provisioning process done inside Identity Manager (for example by providing a URL link that could be embedded in the Identity Manager User interface).

Having described the Tier Model, we will now concentrate on possible Work-flows using these various tiers.

5.2. PROVISIONING WORK FLOWS

The previously mentioned three tier model can be associated to various work-flows for provisioning. The two which are foreseen to be the most common ones will be described in some level of detail. These are a *Role Oriented Work-flow* in which one user will be provisioned to multiple Applications[4] according to a *Policy* (most probably to be used by administrators managing applications), and also a *System Oriented Work-flow* in which a user is provisioned in a single System[5] at a time (most probably to be used for and by EMSA users).

System Provisioning

[4] For the purpose of this document, an Application is interpreted as something that provides a contextualized set of goals for users, for example THETIS or STCW or CSN. Any other software that provides a more generic set of features is considered a System (see next footnote on this subject).

[5] In this particular context, the use of the word System can be analogous to Application. It is being used instead of the word Application because we tend to think of an LDAP server more as a System than as an actual application, for example. Another such example would be the Liferay Portal. Even though it is an application, due to its nature of supporting portlets and hence "applications" inside it, it can be seen as a system.

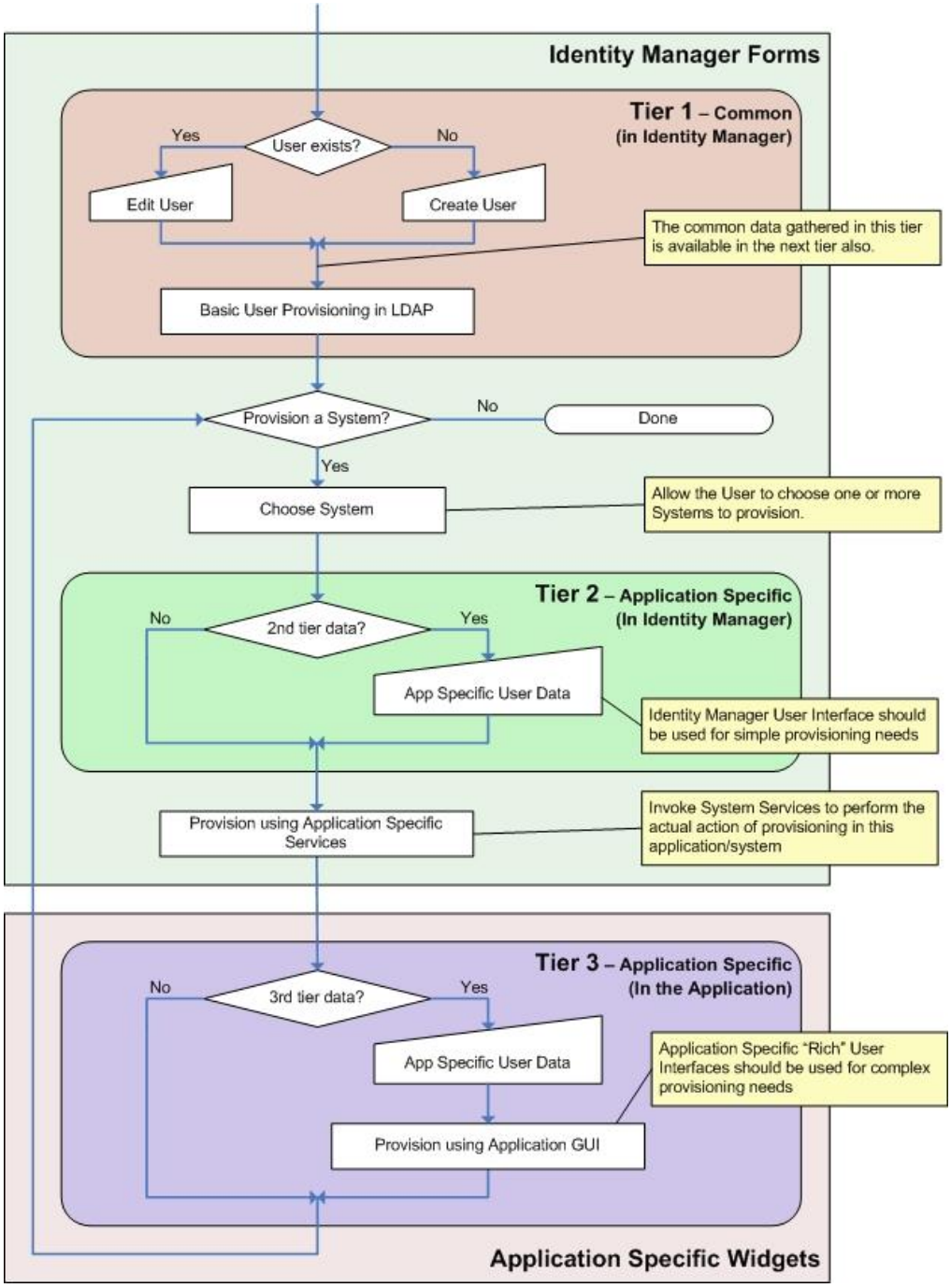

Figure 8: System Provisioning Work-flow

The previous figure shows a possible work-flow associated with provisioning a user in one System at a time. This provisioning can/will typically be done by an administrator of the system or it can be delegated to another user, permissions allowing. This will probably not be the preferred way to provision a user to an Application as the best way to do so would be to associate him with a given Role in the Application (that will be seen later on in this document).

The flow starts by performing a lookup of the user by way of some unique identifier. If the user is found (if he already exists), then the form shown will include this users' data and this data can be changed. If the user does not exist yet, then a clean form will be presented where all the relevant information should be filled in. This first step of providing the relevant "general" information about the user to be provisioned is done inside the first tier. Once the information is complete, the user can then be provisioned to LDAP (the reason for this has already been explained previously and is related to the fact that all EMSA applications use some form or another of an LDAP server to identify the users). If the user who is being

provisioned just needs to be provisioned in the LDAP system, then the flow can end here. If more than this simple LDAP provisioning is needed then the following steps are also used.

At this point it is possible to choose any of the existing systems (subject to the permissions of the user performing the tasks of provisioning) and perform a specific provisioning for this system. This may correspond to having either an Identity Manager form or an application (system) specific interface for gathering more information for executing the actual provisioning work (corresponding to tiers two and three respectively). The actual process of provisioning the user in the selected system may use (re-use) information gathered in the common tier one form and it may also imply changing or adding information in the LDAP system. This is totally dependent on the system being provisioned.

Having completed the provisioning of one system by this process, the user can choose to provision other systems to which he may have access, or he can choose to end the provisioning session in which case all the changes done will be "committed" to the various systems.

This work-flow can be repeated as many times as wanted for the same user performing corrections on data in already provisioned systems or provisioning new systems for the user.

Role Oriented Provisioning

The Role Oriented Provisioning is based upon the concept that a single user may perform functions in one or more applications. In order for him to do so, it may be necessary to give him permissions in more than one system. The logical grouping of such permissions (provisioning) in various systems is called a *Policy*. An example of one possible policy could be the role of a Thetis Inspector. In this case, the user has to first be provisioned in the LDAP system, then he has to be further provisioned in the Liferay system (to be made a Liferay User) and finally he has to be given the correct permission set in the Thetis Application to be able to act as an Inspector. It might also be possible for this Inspector to be given further permissions at the application level to allow him to only see a certain subset of data, for example according to his country.

Similar to the System Provisioning Work-flow, the first step is to identify the existence of the user (in which case a data edit can be performed), or to create a new user (by submitting new data). This first step is exactly the same as in the previously described work-flow (Role Oriented), having been already explained.

From this point on, the difference between this work-flow and the previous one becomes apparent. In this flow there is the possibility to decide upon the Role Policy to apply to the user (instead of applying permissions to actual systems) followed by a tight loop performing the provisioning on the various systems/applications associated with the selected policy.

After the provisioning for a specific Role has been applied, it is possible to select another Role Policy to apply to the user (if this user being provisioned is to have access to various applications), or terminate the process for that user and proceed to provision another user.

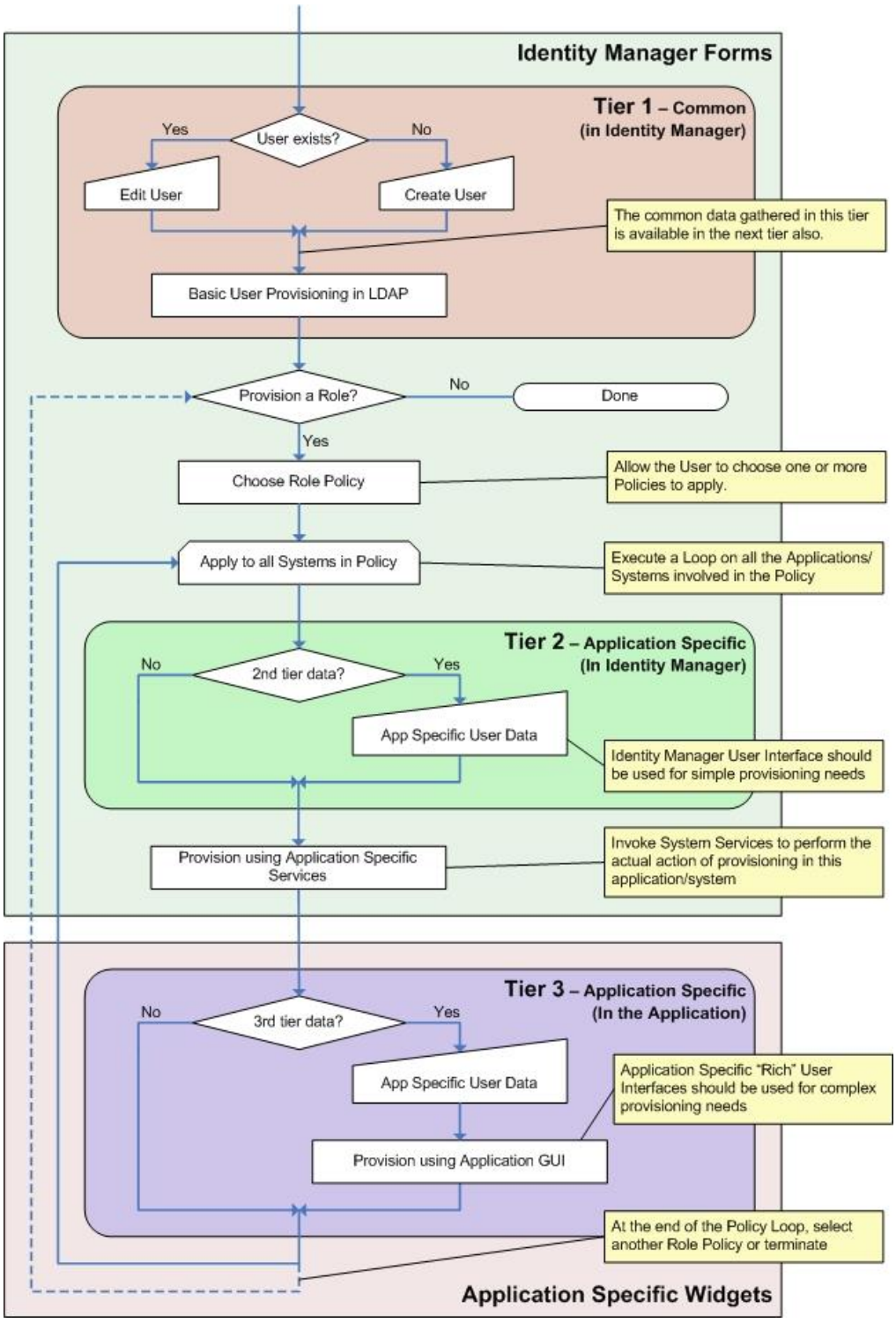

Figure 9: Role Oriented Provisioning Work-flow

5.3. IMPORTANT DEFINITIONS

Now that the provisioning model has been defined, it is time to define how actual EMSA applications can use this model; but before doing so, we will state a few concepts by providing their definitions so that there are no doubts about what is being said and how the existing infrastructure at EMSA supports these concepts.

Basic Definitions

Following is a list of simple definitions that need be fully comprehended.

- **Role-Based Access Control** – Role-Based Access Control (RBAC) is a logical architecture commonly used to implement control of accesses to protected resources (functionality, services, systems, …);
- **User** – An individual member having identification credentials to access a protected resource;
- **Permission** – A privilege that grants access rights to one specific resource;
- **Role** – A set of permissions granting access rights across their resource scope (functionality, service, system, …);
- **Functionality** – A specific resource to be protected;
- **User Group** – A collection of users;
- **Community** – Collections of Users or User Groups who have a common interest.

Relationships

Having defined the basic concepts, following is a list of how these concepts are related to each other.

- A **User** may belong to one or more **User Group**s;
- One **Functionality** has only one **Permission**;
- A **Role** is composed by **Permissions** and one **Permission** may belong to more than one **Role**;
- A **Role** can be assigned to one or more **User Groups**;
- A **Community** is composed by **Users** and/or **User Groups**.

As a side note, we will also state that the following relationships are also possible to implement in the RBAC model, but these complicate a lot the management of the model. Best practices and experience show that these relationships must be considered **only in very exceptional circumstances**; they are possible relations but not desirable ones. It is highly recommended that no EMSA application make use of these concepts.

- A **Permission** may be assigned to a **User** and a **User** can have more than one **Permission**;
- A **Role** can be assigned to a **User** and a **User** can have more than one **Role**.

Besides not applying the extended relations, further restrictions will be imposed (suggested but not enforced) limiting the cardinality of Roles and User Groups. These further restrictions will be discussed in due time.

5.4. PROVISIONING OF EMSA APPLICATIONS AND SYSTEMS

This sub-chapter provides generic information on the provisioning of the EMSA applications.

5.4.1. Provisioning of jPetStore Application

JPetStore has an internal representation of users. This consists of various database tables where information on the users is kept. In the jPetStore application deployed in the EMSA test environment, the provisioning process has not been touched, which goes to say "users" have not been provisioned through OIM (Oracle Identity Management). A full integration with the complete suite of Oracle products deployed for this solution would require deeper changes to the application in order to support the provisioning through OIM, namely:

- Removal of the "Register Now" link from the Login page;
- Change in the user Account form to only allow viewing of the attribute values (no edit allowed);
- Creation of new services to allow "remote" calls to the user provisioning methods (preferably through web services).

An implication of the provisioning process not being integrated is that it is now necessary to put the password exactly the same as the username.

5.4.2. Provisioning of RuleCheck Application

RuleCheck "users" do not have a specific provisioning process so there was no work done in OIM (Oracle Identity Management) for this application, at least at the present moment. In the future there may be a provisioning process for RuleCheck users but this will most certainly only be the creation of users/association to the correct groups in LDAP.

5.4.3. Provisioning of STCW Application

STCW "users" do not have a specific provisioning process so there was no work done in Oracle Identity Management.

5.4.4. Provisioning of THETIS Application

The generic process for provisioning applications has been discussed. We will now proceed to discuss the intended process for specific applications starting with THETIS.

Provisioning a new user

For a user (with the correct credentials) to be able to create a new user, he will have to start by accessing a link to OIM available in the Thetis theme. This link will contain a parameter that indicates that it is the Thetis application that is performing the request, and will open an OIM form where the user can search for the existence of the new user to be created. Even though this step seems to be redundant, after all we do want to create a new user, it is necessary because the user may be "new" in Thetis but already registered (already exists) in another context (for example CSN or STCW). If the user is not found in this initial search, then it effectively is a new user and must be provisioned in LDAP, Liferay, etc. followed by the actual Thetis provisioning. On the other hand, if the user is found in the initial search, then all that is needed is the Thetis provisioning.

Provisioning for an existing user

For a user (once again with the correct credentials) to be able to update an existing user, he will have to start by accessing a link to OIM available in the Thetis theme. This is similar to the creation process but differs in that the link provided has not only a parameter indicating the Thetis application, but also a parameter indicating the ID of the user to be provisioned. This will lead to an OIM form that contains the result of the search for the indicated user ID. It is thus possible to confirm the existence of the user, and proceed to the actual Thetis provisioning, or to discover that the indicated user does not exist after all and take the appropriate steps to correct the situation.
An alternative option is to access the link without the use of the ID parameter. This case results in something similar to the initial creation process in which a search form is presented where the user can apply a search criterion. After having found the desired user, further provisioning can be done on this user as if his ID had been passed initially.

Provisioning Thetis

Independently of being a new or existing user, after having passed the previously mentioned search form, the provisioning user will be allowed to change/fill in (update and create respectively) the new users' common data in a first form (by common data we mean data that is common to all EMSA applications like userId, name, etc.). Once the common data has been filled in, the next step is to define more information about the user (for example, roles).
Due to the fact that the provisioning user arrived at the OIM screens through a link in Thetis, the application is already known at this time. If it were not known, then a screen showing the possible applications to provision would be presented (the actual applications available would depend on the permissions held by the provisioning user) and one would be chosen.
Since we know that we are provisioning Thetis, a "second" form will be presented where the user can register specific information only present in Thetis along with the association of application "roles" to the user.

Provisioning Services

Typically provisioning of users is not as trivial as filling in the first level form in OIM and choosing an application to associate to the user. Normally there will be a need for further work to be done inside the actual application to which the user is being granted access. The way to promote the interaction between OIM and the applications will be through Web Services. Each and every application will have to implement at least one Web Service which will allow OIM to execute CRUD services in the application. The exact format for this service is still to be defined at the present moment, but should look something like:

- A first section which will contain all of the mandatory parameters for the creation/edition of a "generic" user;
- A field indicating what type of CRUD operation is to be performed by the application (still belonging to the first section);
- A second section containing a Hashmap of "key – value" pairs that is meaningful to the application (that contains the application specific data for the user). Please note that a possible Hashmap entry can be another Hashmap whenever complex structures are needed (such as for defining roles).

Whilst the first section is mandatory and will be defined by EMSA/OIM, the second section is optional and it will be the responsibility of the contractors of each application to provide relevant information as to the keys needed for their application. A complete specification of the previously mentioned Web Service can be found in the annexes.

Provisioning Permissions

It has been mentioned various times previously that the user performing the provisioning needs to have the privileges necessary to do so. The permission set for provisioning Thetis is as follows:

- ThetisSystemAdministrator: Can create/update/delete any user inside Thetis;
- ThetisNationalAdministrator: Can create/update/delete users who belong to the same Country as this user;

User self-service is to be done by the actual user and allows for changing of personal information (as long as this information is changeable, i.e. userId is not changeable).

It should be possible to revoke (delete) a user in one of two ways:

- Globally. The users' access is removed from all applications;
- Per application. The user loses access to the particular specified application.

Please note that a revocation is permanent and cannot be undone.

If there is a need to temporarily disable access for a user to a given application (or all at once), this can be done in OIM by issuing a Disable/Enable command. Either of the two operations can be undone by performing the complementary operation.

6. EMSA OIM Custom Interface

EMSA needs to access OIM directly from links placed in some third party applications. The implementation of an OIM custom interface has been created in order to provide a full User Management functionality in OIM - Create and Edit User fully adapted to the customer needs.

Create User

The procedure followed by EMSA to create a new OIM user and provision the necessary resources involves, apart from filling up the OIM user information, selecting a group that will trigger the corresponding Access Policy that automates the resource provisioning to the new user. All this is done in one single step.

In order to avoid creating more than one account in OIM for the same user, a search process is launched before creating the user, using the **First Name** and **Last Name** as search criteria. If some results are returned, they are shown in a table (User ID, First Name, Last Name and Email) so the user can decide whether the user already exists or creates a new one. If no results are returned by the search process, the user is created automatically and the resources provisioned.

The groups that a user is able to manage must comply with the security model defined in OIM defined for EMSA.

Edit User

The third party applications are responsible to display the list of users that can be edited by the logged user. By clicking on the link of the user to be edited, the View Details window will be shown, including, as well as the OIM User information the groups assigned to it.

The user information and the group assignation can be edited at one single step. When modifying the group the resources associated to the old group will be revoked and the new resources will be provisioned.

Apart from editing, a user is also able to enable/disable and delete OIM users.

6.1. APPLICATION GENERIC APPROACH

For every application that accesses the Custom Interface there may be different fields for the User Forms. This means that, apart from the system fields common to all the applications, there may be some OIM User Defined Fields that may appear in the user forms for one application but not in the others. These will be referred to as application-specific fields.

Note: <u>One field can only belong to one application.</u>

6.2. ACCESSING THE CUSTOM INTERFACE

The Custom Interface will be accessed through links placed in the third party applications.

Create User

The URL to access the Create User Form window will look as follows:

http://host.domain/xlWebApp/createUserCustom.do?method=**New+User**&application=***Thetis+User***&userAction=new

In the previous link, "*Thetis+User"* has to be changed to the correct resource name for the application in which the link is published.

Edit user

The URL to access the Edit User Form window will look as follows:

```
http://host.domain/xlWebApp/createUserCustom.do?method=viewUser
Details&loginID=ORACLEUSER1&application=Thetis+User&userAction
=new
```

In the previous link, *ORACLEUSER1* has to be changed to the correct user identification per call to the edit method. See also the comment made on the "Create User" link about the application name (*Thetis+User*).

Edit my account (only fields common to all applications)

The URL to access the Edit my account User Form window will look as follows:

```
http://host.domain/xlWebApp/createUserCustom.do?method=viewUser
Details&loginID=&application=&userAction=new
```

Please note the *loginID* and *application* variables must exist in the link and be empty for this to work.

7. Password Management

Besides granting access to resources, a Single Sign-On solution has one other major task, that of managing users credentials or passwords. The managed credentials obey to certain conditions set out by a password policy. We'll explain how credentials are managed at EMSA and also the password politics adopted at EMSA in the following sections.

7.1. CHANGE PASSWORD / LOST PASSWORD MANAGEMENT

The EMSA IdM platform is currently responsible for the Password Management actions encompassing several different functionalities. This document only refers to two specific functionalities, change password and lost password.

The SSO solution for managing passwords adopted at EMSA started with an out-of-the-box solution proposed by Oracle but after some time this was deemed as inadequate and a new bespoke solution was developed by Oracle. Both of these solutions are described hereafter.

7.1.1. Original Situation

The original solution proposed and implemented by Oracle was:

1. "Change Password", allowing a user to change his own password when he still knows his current password. This was achieved by entering the current password and the new password twice.
2. "Lost Password", allowing a user to change his password when he doesn't know the current one. This was achieved by correctly answering a set of challenging questions and introducing the new password twice.

Although the original implementations of these functionalities respected the requirements, some drawbacks were observed regarding the usage of such functionalities:

1. The original implementation of "Lost Password" used the Challenge Questions principle to decide if the requester was who he said to be. Answers to the Challenge Questions were sometimes very easy to know or find through Social Engineering.
2. Users were not properly educated about the importance and criticality of their answers to the Challenge Questions. From the *End User* point of view it was much easier to ask EMSA to reset the password than remember what the answers were.
3. As a consequence of the previous point, *End Users* tended to answer Challenge Questions in a very light-hearted way and almost immediately forget their answers.
4. Another drawback discovered was that the original implementation of "Change Password" and "Lost Password" allowed *userId* enumeration under certain circumstances.

Taking the previous points into consideration, EMSA recognized a need for a change in the implementation of the aforementioned functionalities.

7.1.2. Current Situation

Change Password
The original implementation allowed *userId* enumeration because the "Change Password" required the user to insert a valid *userId* before going to the actual page to change the password. The navigation was done using a link that was available in the Login screen before the user was authenticated.

Placing this link in a private area has solved the problem. As private areas are only accessible after user authentication, it assures that the user meets the conditions to change his password.

Therefore:

1. The "Change Password" link was removed from the original Login screen;
2. A Link to the "Change Password" functionality is now available in the "My Account" page provided by IdM to all Maritime Applications. Using this common IdM page avoids the need of changing the Maritime Applications that aren't deployed under EMSA Portal.

Lost Password
A 2 step procedure based on a One-Time generated URL replaced the original Challenge Questions mechanism for the "Lost Password" functionality. This way, both drawbacks identified in the original "Lost Password" implementation were fixed.

The "Lost Password" function is also able to unlock an account (if previously locked) and provides a detailed logging mechanism to allow an easy diagnosis of faulty or doubtful situations and/or audits.

The current process is aligned with the requirements for the "Lost Password" functionality, replacing the original "Challenge Questions" mechanism and presenting a solution for the original identified constraints.

However, it should be noted that currently:
1. E-Mails are not unique. Usage of shared e-mails might be problematic from the *End User* point of view.
2. E-Mails are not mandatory. A user without an e-mail address configured will not be able to recover his password by himself. He will always have to request EMSA intervention to obtain a new password (i.e. "reset" user account).

Maritime Applications are strongly encouraged to take measures in order to address these two constraints.

8. Security Model

Up until now this document has been focused on the technical side of IdM. Nothing has been said of the business requirements associated with user management. It has not been by chance that this is such as each Maritime Application imposes its own "set of rules", which are conveniently documented under each applications respective scope. There is however one important general rule to which all applications have forcibly to adhere and that is the so called **EMSA Security Model**. This model defines the necessary levels of user permissions that allow any user to edit any other or, said in another way: in terms of Identity Management, the Security Model defines who can do what in a hierarchical way.

The EMSA Security Model has 5 hierarchical levels. From the most privileged level to the least, these are:

1. **EMSA Administrator**
   Identity Manager *super users*. Users belonging to this level are entitled to manage ***all user accounts without restrictions*** and they also have privileges to access some normally restricted IdM functionalities. "EMSA Administrator" level can only be assigned to a person belonging to EMSA and is normally limited to a very small number (no more than 3).

2. **EMSA Application Administrator**
   Identity Managers for a specific Application. Users belonging to this level are entitled to manage ***user accounts related with a specific application*** (i.e. the Administrator´s). "EMSA Application Administrator" can only be assigned to a person belonging to EMSA and should be limited to a small number (it's a business decision to define how many administrators exist for any given application). It should be noted that a single person can be associated (i.e. have this level) with more than one application.

3. **National Administrator**
   Identity Managers for a specific Country/Institution inside a specific application. Users belonging to this level are entitled to manage ***user accounts that are simultaneously related with the Administrator´s application and the Administrator's Country/Institution***. "National Administrator" level can be assigned to any user of a specific Country/Institution even though at the business level there is normally a very limited set of people that possess this privilege.

4. **Local Administrator**
   Identity Managers for a specific Local Authority inside a specific application and Country/Institution. Users belonging to this level are entitled to manage ***user accounts that are simultaneously related with the Administrator´s application, Country/Institution and Local Authority***. "Local Administrator" level can be assigned to any user of a specific Country/Institution for a given Local Authority.

5. **End-user**
   End-Users have the most limited set of Identity Management privileges. They are only entitled to modify a limited set of their own personal attributes (i.e. the ones which are common to all applications).

It should be noted that not all Maritime Applications contemplate the use all of the levels. Most notably the **Local Administrator** is rarely (if not at all) used by most applications.

8.1. ACCUMULATION OF LEVELS

IdM allows any given user to accumulate more than one level or even various instances of the same level. There are certain restrictions that may exist while applying the accumulation (enumerated below). Whenever there is an accumulation of levels, the privileges available to the user always correspond to those of the highest level (when accumulation of different levels) and the sum of privileges whenever the same level is given more than once.

Why would a person accumulate levels? Normally there is no reason for this to be done as the privileges will correspond to the highest level assigned to the user therefor making the other (lower) levels irrelevant. This said, there are occasions in which the user momentarily (for a given event or time period) needs more privileges then those normally assigned to him. In this case the higher level is assigned during the known period of time and later on it is removed returning the user to his normal privilege level. This "vertical" accumulation corresponds to a vertical escalation of privileges.

Another reason for accumulation is the case of an EMSA user that is an administrator of more than one application. This "horizontal" accumulation corresponds to the sum of privileges (all at the same level). Even though this is rare, it may occur due to changes in business teams, etc.

As previously stated, there are certain restrictions to the accumulation of levels. Namely any accumulation that implies the existence of more than one Country/Institution for any given user is forbidden (as the user can have only one Country/Institution assigned to him). Likewise any combination that may conflict with the cardinality of attributes is forbidden (for example trying to be a Local Administrator for two different Local Authorities for the same application, etc.).

8.2. SECURITY MODEL LEVEL CORRESPONDENCE TO APPLICATION ROLES

One common misconception that occurs relating to IdM is the assumption of an implicit relationship between the EMSA Security Model and the Maritime Application roles. **This implicit relationship does not exist**. Any given user can be, for example, an end-user within an application and simultaneously be an Administrator (EMSA or National level) within IdM (for that same application). There is no mechanism imposing any limitation whatsoever. However, it is common for applications to request the establishment of a relationship of their internal application roles to certain security model levels **explicitly**.

One form of explicit relationship establishment is through role mappings. This means that whenever a given application role is assigned to a user, he will "inherit" (be automatically assigned) a certain security model level. One example of such a mapping is whenever a user is assigned the "Thetis System Administrator" role he will also be given the "EMSA Application Administrator" level as well. This goes without saying that only EMSA personnel can have this role/level.

One other form of explicit relationship is the existence of a specific field in the applications custom form in which the actual security model level can be chosen during user creation/editing. This is the way SSN chose to establish the relationship. Of course not all security model levels can be assigned depending mainly on the actual security model level of the user doing the creation/editing so as not to allow permission escalation.

In the end, it is important to retain that management inside IdM is (or can be) completely independent of any form of management within any given Maritime Application.