

Appendix B to the Tender Specification of procurement  
procedure EMSA/NEG/30/2019 "Alternative Learning  
Management System - Proof of Concept"

**IdM Guide**

**Identity and Access Management Guide**

(Abridged Version)

## Document History

Title

**Identity and Access Management Guide** (abridged version)

Version

2.1 from 01/02/2019

---

---

## Table of Contents

Definitions, acronyms and abbreviations.....	5
1. Introduction and objectives .....	6
2. EMSA IAM technical overview .....	8
3. Access Management .....	11
3.1. Protecting Applications .....	11
3.2. Authentication .....	11
3.3. Authorisation.....	11
3.4. Webgate .....	12
3.4.1. SAP Configurations.....	13
3.4.2. Common Configurations.....	14
3.5. Oracle Access Manager .....	14
3.5.1. Access Policies .....	14
3.6. RBAC Implementation in the EMSA Infrastructure .....	15
3.6.1. LDAP .....	15
3.6.2. Liferay Enterprise Portal.....	15
3.7. Deploying Applications with Single Sign-On.....	16
3.7.1. Portal integration .....	16
3.7.2. jPetStore.....	17
3.8. Logging out of Single Sign-On .....	19
3.8.1. Technical implementation of a global Logout.....	19
3.9. Password Management .....	19
3.9.1. Change Password / Lost Password Management.....	19
3.10. MAP Integration.....	20
3.10.1. MAP login Process .....	21
3.10.2. MAP Access Policies .....	21
3.11. JSON Login .....	21
3.11.1. User Authentication .....	22
3.11.2. User Authorization .....	22
4. Identity Management.....	23
4.1. EMSA Business View on Identity Management .....	23
4.1.1. Service .....	23
4.1.2. Profile .....	23
4.1.3. Role .....	23
4.1.4. Country/Institution.....	24
4.1.5. Organization .....	24
4.1.6. Operation.....	24
4.2. Security Model.....	24
4.2.1. Security Model Level Correspondence to Application Roles.....	25
4.2.2. Accumulation of Levels .....	25
4.3. Identity Management Functionalities .....	25
4.3.1. Reconciliation .....	26
4.3.2. Account Management .....	26
4.3.3. Provisioning.....	26
4.3.4. Other Administrative Functions .....	26
4.4. Identity Management Integrations .....	26
4.4.1. Provisioning Applications or "PUSH" Model.....	26

---

4.4.2. User Information Web Service (or “PULL” Model).....	27
4.4.3. Accessing IdM Functionalities via direct URL .....	27

## Table of Figures

Figure 1: Context Diagram .....	8
Figure 2: Technical Components .....	9
Figure 3: Authorisation denied .....	12
Figure 4: WebGate Configuration Architecture.....	13
Figure 5: Integration Sequence Diagram .....	17
Figure 6: Non-integrated Login.....	<b>Error! Bookmark not defined.</b>
Figure 7: MAP integrated Login .....	21

## Definitions, acronyms and abbreviations

Definition	Description
AccMng	Access Management
AD	Microsoft Active Directory
BCF	Business Continuity Framework
CMC	Common Management Console
Country/Institution	Defines the Nationality of a User and the area of control of a National Administrator.
CSN2	Clean Sea Net 2 Maritime application – version 2
EMSA	European Maritime Safety Agency
IAM	Identity and Access Management
IdM	Identity Management which comprises Access and User Identity Management
IdM V2	Identity and Access Management, version 2
IMDatE	Integrated Maritime Data Environment Maritime application
JAAS	Java Authentication and Authorization Service
LDAP	Lightweight Directory Access Protocol
LRITDC	Long-Range Identification and Tracking Data Centre Maritime application
JSON	JavaScript Object Notation. Lightweight data-interchange format
MAP	Maritime Application Portal (Liferay customisation of entry page to act as an “access point” for all of EMSA’s Maritime Applications)
MarApps	Abbreviated form of referring to EMSA Maritime Applications
MSS	EMSA’s Maritime Support Services
OAM	Oracle Access Management
OIM	Oracle Identity Management
Operation	Defines an Action that is available to a User.
Organization	Defines the Organization a User belongs to and the area of control of a Local Administrator.
OSB	Oracle Service Bus
OVD	Oracle Virtual Directory
RAC	Oracle Real time Application Cluster
REST	Representational State Transfer. Web Services that conform to the REST architectural style
RuleCheck	Application providing EU and International legislation regarding Port State Control
SAP	Webgate Specific Access Point configuration
SEG	SafeSeaNet Eco-system GUI
Service	Represents a set of (one or more) Business Functions implemented by an application (MarApp).
SOA	Service Oriented Architecture
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
UMC	User Management Console
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.**

During the past years, EMSA has developed a common infrastructure to provide Identity and Access Management (IAM) services to the EMSA Maritime Applications.

IAM suggests that each user assume a unique digital identity across applications and systems, which enables access control to be assigned and evaluated against this identity at a central place as well as centralized management of user attributes. Thus, the IAM concept encompasses two major areas:

- **Access Management** managing authentication and authorization to resources and Single Sign-On (SSO) which is a mechanism whereby a single action of user authentication and authorization can permit a user to access all applications and systems where he has access permission, without the need to enter multiple passwords.

Currently at EMSA, **Oracle Access Manager (OAM)** 10gR3 (10.1.4.3.0) is used to provide base Access Management and Single Sign-on functionalities.

- **Identity Management** is the management of the unique digital identity, associated attributes, security model and permission behind it. The set of user attributes varies from application to application and includes, among others, First Name, Last Name, Email, Groups and Roles. The security model establishes the management relationships (e.g. who is entitled to create/edit other users) and the permission rules (e.g. a Service Administrator can create users inside his own Service (application) and a National Service Administrator can create users belonging only to his own country for the service he manages). In addition, Identity Management also provides user's attributes and roles assignments to all applications that the user has access through a background provisioning process or through dedicated services.

Currently at EMSA, **Oracle Identity Manager (OIM)** 11gR2 is used to provide base Identity Management functionalities.

The IAM service conveys benefits to an enterprise through:

- Central user repository for all applications and systems;
  - Central User Management avoiding different implementations and rules across the enterprise;
  - Reduction of human errors, a major component of systems failure, therefore highly desirable but difficult to implement;
  - Reduction in the time taken by users in sign-on operations to individual domains, including reducing the possibility of such sign-on operations failing;
  - Improved security due to the reduced need for a user to handle and remember multiple sets of authentication information;
-

- Reduction in the time taken, and improved response, by system administrators in adding and removing users to the system or modifying their access rights;
- Improved security through the enhanced ability of system administrators to maintain the integrity of user account configuration including the ability to inhibit or remove an individual user's access to all system resources in a coordinated and consistent manner;
- Significantly reduce the User Management maintenance and operation effort.

The document is organized in several chapters:

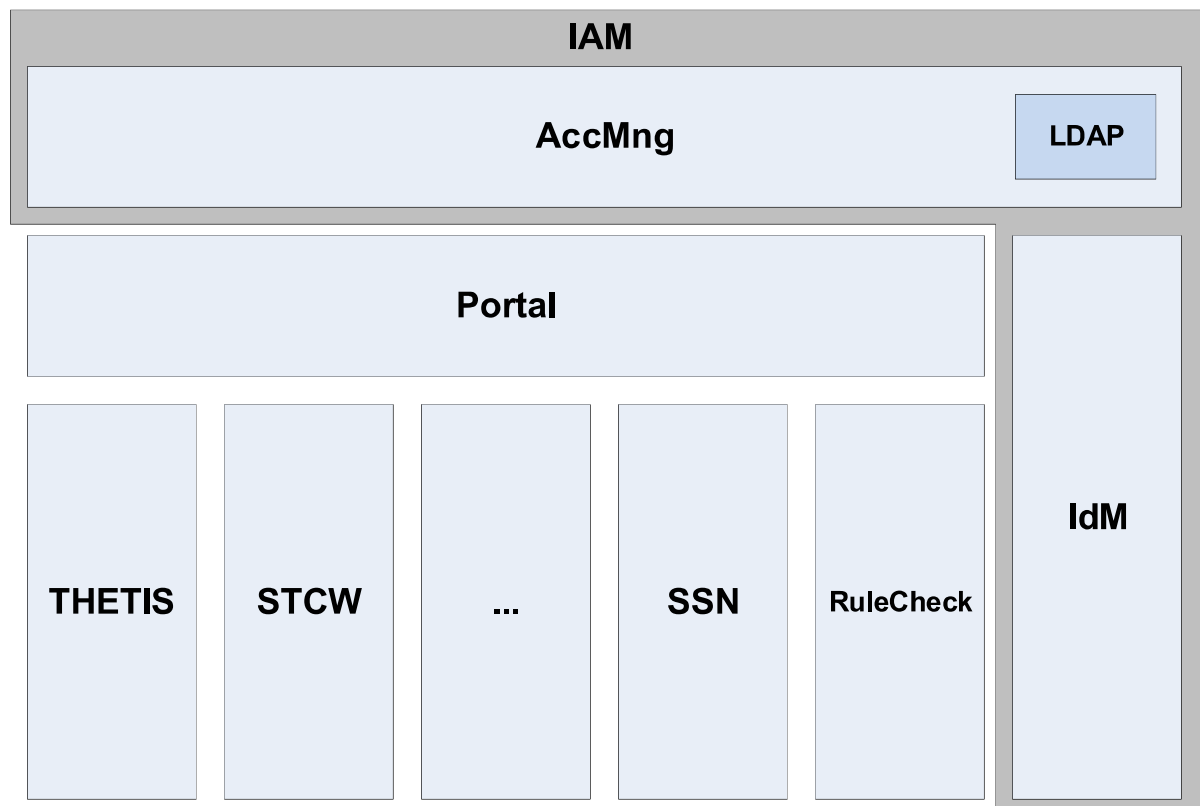
- Chapter 1: Introduction and objectives. This chapter;
- Chapter 2: EMSA IAM technical overview. A quick description of the architecture and components support EMSA IAM.
- Chapter 3: Access Management. Focus is given to the principles and implementation of EMSA's Access Management infrastructure.
- Chapter 4: Identity Management. Focus is given to the principles and implementation of EMSA's Identity Management infrastructure.

One final note about this document, as it is intended to be a guide used for presenting the information on reasons, implementations, etc. it is not necessarily supposed to be read "as a book", i.e. from the beginning to the end in a sequential manner. This document is more of a look-up to certain details and consequently may repeat information or "state the obvious" in some parts which have already been spoken about in other parts or in other documents.



## 2. EMSA IAM technical overview

The Introduction and objectives chapter presented the concept of IAM as implemented at EMSA. The next figure shows the context diagram of EMSA's IAM framework:



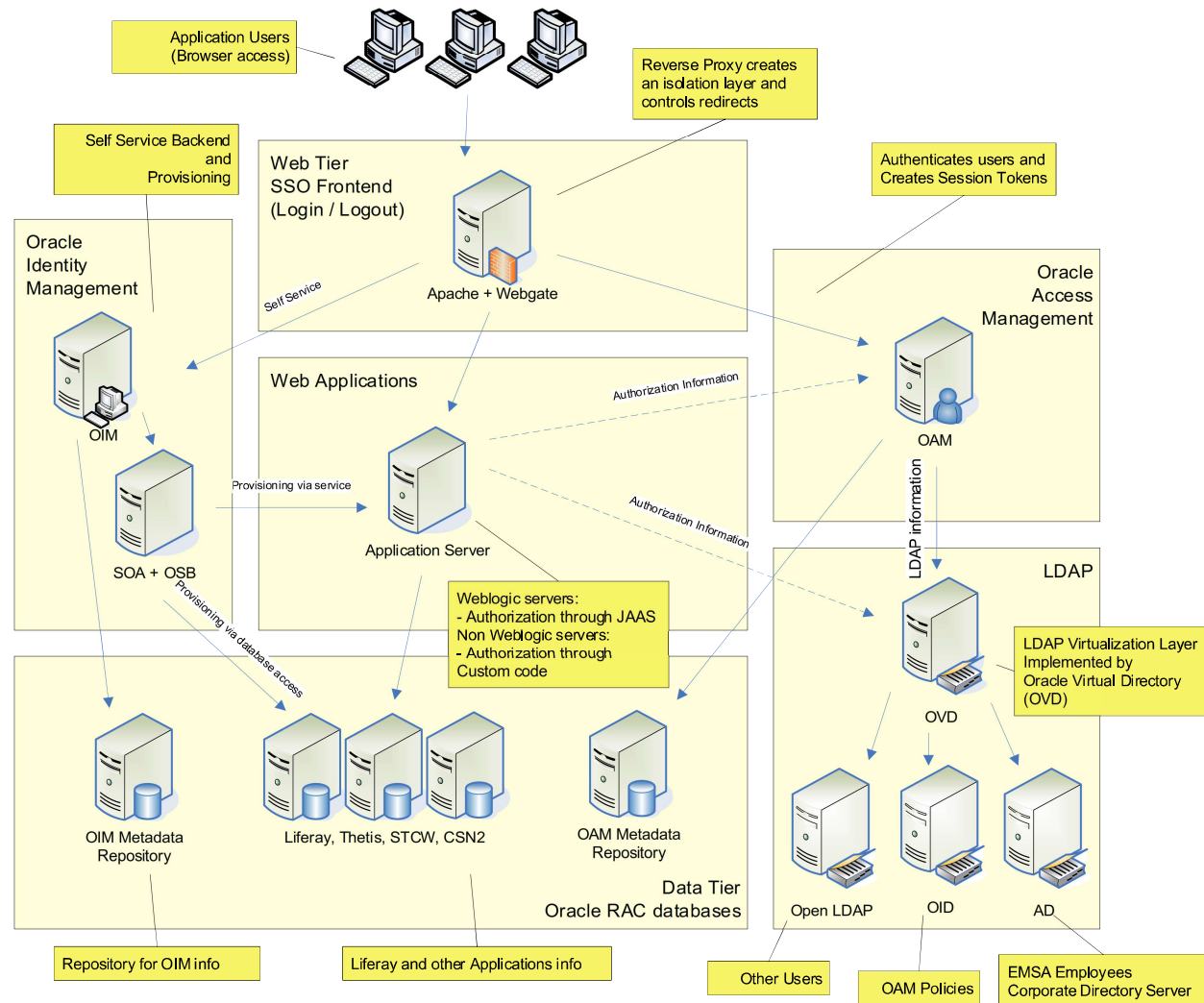
**Figure 1: Context Diagram**

The EMSA IAM framework provides governance for the accesses to EMSA applications. In a very simple way, we can say that:

- AccMng (Access Management) grants access to EMSA applications, providing Single Sign-On capabilities across those applications;
- IdM (Identity Management) manages the accounts (entities) that are entitled to use EMSA applications, providing functionalities like creation of new accounts or modification of existent accounts (changes of account attributes or Roles). It should be noted that AccMng also provides SSO functionality to IdM;
- LDAP is the central repository for access management maintaining information on accounts, roles and associations between accounts and their assigned roles;
- EMSA's Portal solution is built upon Liferay Portal. The Portal provides a single point of entry for several EMSA Maritime Applications (STCW-IS, THETIS, ...) and, for those applications, also takes care of the authentication process by interacting with AccMng;
- EMSA applications (STCW-IS, THETIS, ...), commonly referred to as Maritime Applications (MarApps), implement and provide EMSA core business functionalities.

The following diagram depicts the same information as the previous diagram, providing a deeper view of the different technical components used and includes the basic flow of requests. However, the machines depicted are purely "logical" and may not correspond to

actual physical machines (these may be single, clustered or joined together depending on actual implementation constraints).



**Figure 2: Technical Components**

The components of the IAM high level blocks depicted above are identified below:

- AccMng, Access Management, is composed of:
  - SSO Frontend (Apache + Webgate) + OAM + LDAP virtualization
  - Data repository
- IdM, Identity Management, is composed of:
  - OIM
  - SOA Suite + OSB
  - Data repository
- Web Applications
  - Please note that this block aggregates Portal and Maritime Applications (THETIS, STCW, SSN, ....)

From the Access Management point of view, 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<sup>1</sup> (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<sup>2</sup> (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.

If 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 -> SOA+OSB -> Web Application) or just store information inside its own database to be accessed through specific Web Services.

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 servers used.

If WebLogic is used, a JAAS integration might be best option; if not, a call to the OAM API through custom code will need to be done.

Note that, although not represented in the diagram (for clarity reasons), LDAP is usually provisioned (OIM -> SOA+OSB -> openLDAP) with the accounts information to serve as the base for the Authentication and Authorization process described above.

---

<sup>1</sup> "Web Applications" refers to Portal, THETIS, STCW, ....

<sup>2</sup> Although shown in the diagram, corporate AD is not integrated

### 3. Access Management

---

#### 3.1. PROTECTING APPLICATIONS

---

EMSA hosts several 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.

---

#### 3.2. 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 are never allowed past 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.

---

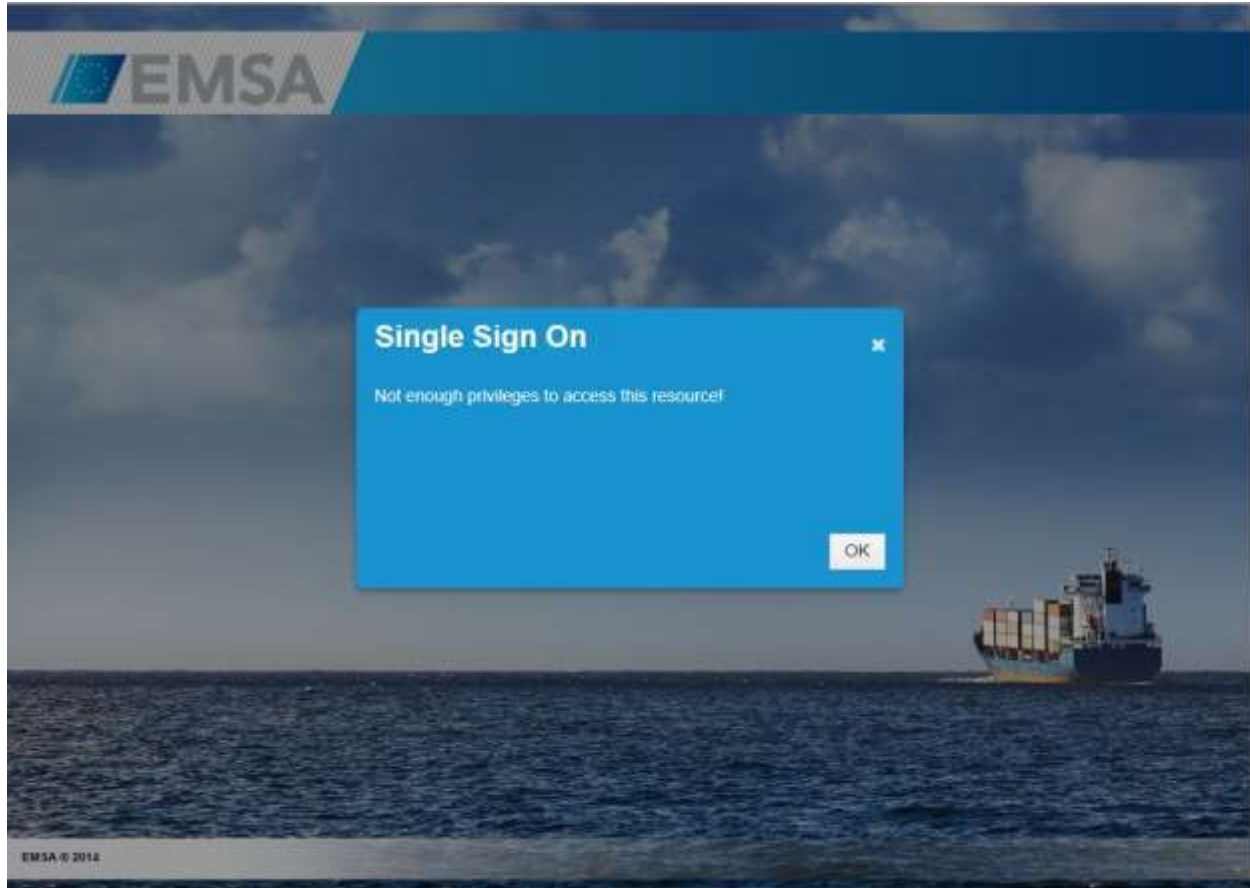
#### 3.3. AUTHORISATION

---

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. 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 the 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 of OAM but this is always much simpler and cheaper time-wise than updating code. This mechanism is extensively used for protecting access to the RuleCheck MarApp.



**Figure 3: 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.

---

### 3.4. WEBGATE

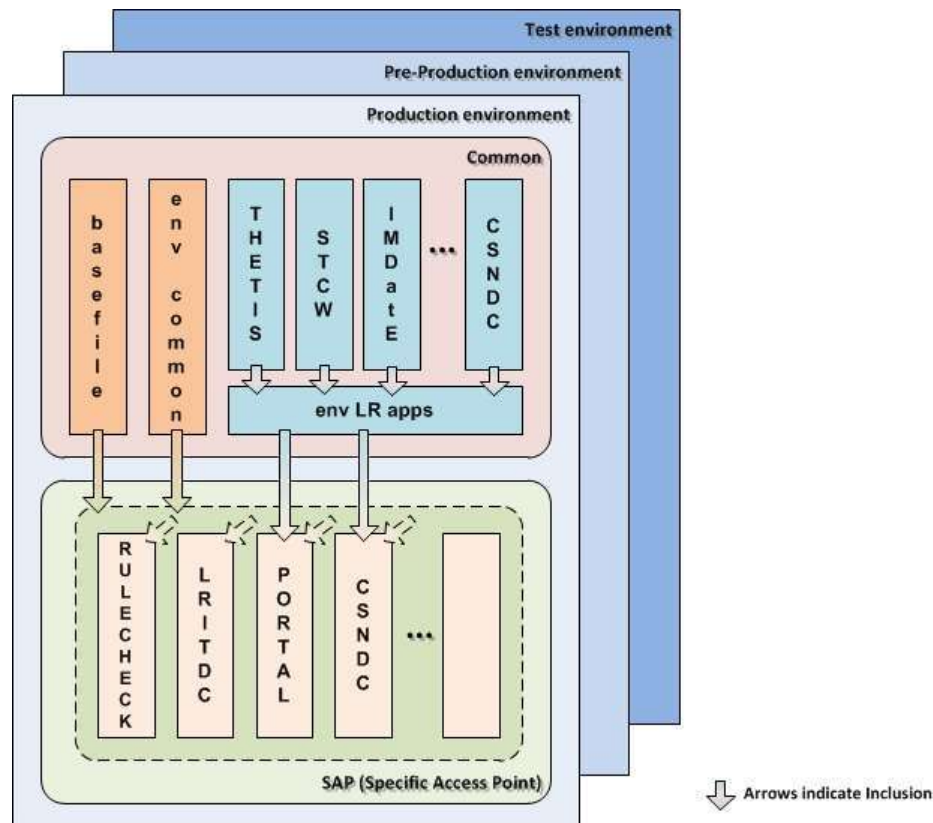
---

**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 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 an SAP (Specific Access Point). Given that EMSA has three environments that are subject to SSO, and various 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 4: 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 are multiple SAPs per environment (not necessarily the same ones in all environments).

### 3.4.1. SAP Configurations

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

EMSA's production environment contains various SAP, each having its own instance of an Apache HTTP server running. 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 those other applications.

The advantages of having SAP are:

- Avoiding unavailability of non-related access points;
- Greatly reducing the amount of work necessary to maintain WebGate configurations by maintaining logical aggregations.



### 3.4.2. Common Configurations

After having extensively analysed all the configuration files for all MarApps in all environments, a common set of attributes/definitions was identified. To ease the maintenance burden, all 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.

---

## 3.5. ORACLE ACCESS MANAGER

---

Earlier in this section 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 implements both 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).

### 3.5.1. 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.

#### **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.

---

### 3.6. 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 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).

#### 3.6.1. 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, using 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
- Since 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 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.

#### 3.6.2. 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 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.
- Sites – In Liferay, a Site 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 site), Roles can be associated to a Functionality (Page) in a Site.

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.

---

### 3.7. DEPLOYING APPLICATIONS WITH SINGLE SIGN-ON

---

Integration with AccMng and SSO can be a simple or a complex task, depending on the type of applications to be integrated.

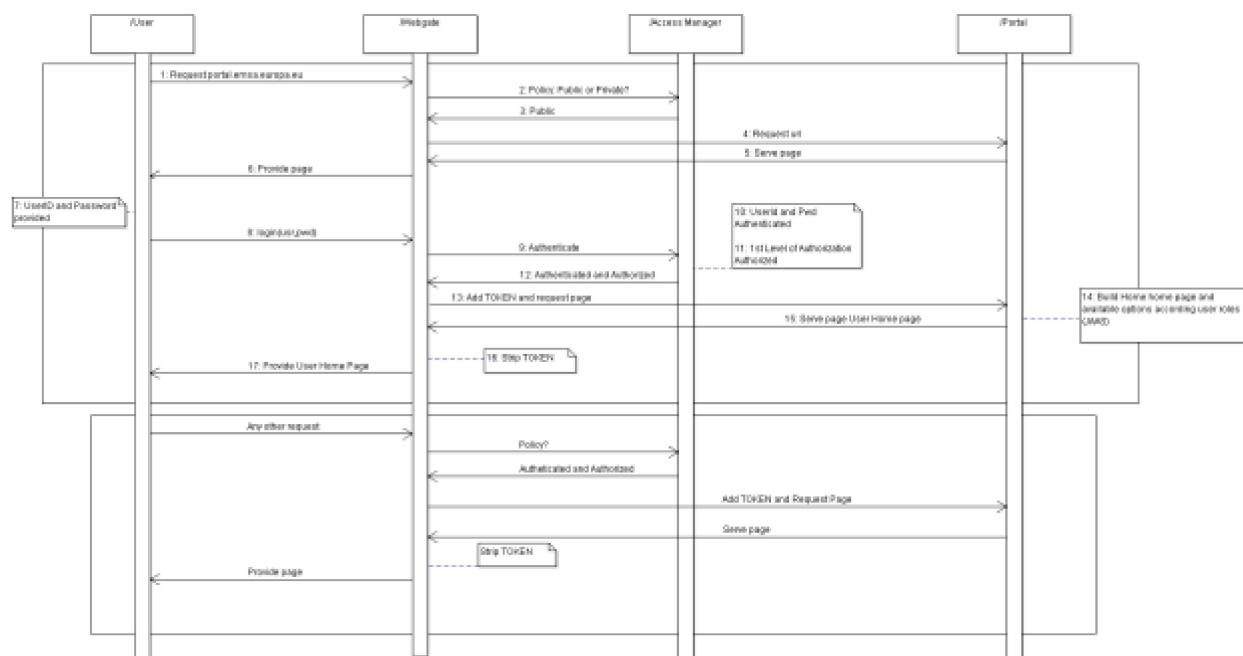
Simple applications may be integrated without any changes. In this case, AccMng only grants or denies access based on the application URL.

For more complex applications or applications with more demanding access rules:

- Some applications will never need to directly interact with AccMng (such as Thetis, STCW, etc.) because they are executed under the EMSA Portal and/or will obtain the necessary information by using JAAS. The integration sequence diagram in chapter 3.7.1 details the EMSA’s Portal integration with AccMng.
- It is likely that other applications may need to be modified to be integrated with AccMng. Chapter 3.7.2 documents how these changes can be done by using a “generic” application such as the Java Pet Store reference application as a “Guinea pig”.

#### 3.7.1. Portal integration

The next figure shows a sequence diagram representing the EMSA’s Portal integration with AccMng:



**Figure 5: Integration Sequence Diagram**

### 3.7.2. 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 must 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**

Because 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.

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 he tries to access the "private" area of the application (accessing the user account or checking out an order).

## **Final notes on configuration**

We have had also the need to configure another policy, the same as the previously mentioned "JPetStore Public", associated to the public URL for the application "/jpetstore".

One other important aspect to consider is the need to change the default session identifier. If the application is implemented using Java technology, change the default session identifier from JSESSIONID to something different (unique to the application), i.e. JSESSIONID\_petStore. If you do not make this change, there is a high probability that there will be "session corruption" if more than one Java application is protected by the same Access Manager.

---

### 3.8. 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 must 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 the auto-login capability of 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 logging out of an application as the result is the same as 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.

#### 3.8.1. 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 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.

---

### 3.9. PASSWORD MANAGEMENT

---

Besides granting access to resources, a Single Sign-On solution has one other major task, that of managing user's 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 the password politics adopted at EMSA in the following sections.

#### 3.9.1. 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 was deemed as inadequate and a new bespoke solution was developed by Oracle.

## **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 “Reset Password” functionality is now available in the “My Information” 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.

## **Reset Password**

IdM V2 now supports a concept of resetting a user’s password. This functionality can be accessed when editing an account by selecting the “Reset Password” link. Correct authorization is executed in the implementing code to verify if indeed a user can or not change the password for the account requested (for example, at this time a National Service Administrator – or lower – cannot change passwords for accounts). The basis for this functionality is the “Lost Password” implementation (described next) with the difference that the password introduced can only be used once (to effectively enter the system and change the password to something only known by the end-user) and also the auditing information registered is very clear that the action was done by an administrator and not upon request of the end-user.

This functionality is mainly useful when the “Lost Password” cannot be executed because an invalid email is defined for an account, or when the email is generic for multiple accounts.

## **Lost Password**

A 2-step procedure based on a One-Time generated URL replaced the original Challenge Questions mechanism for the “Lost Password” functionality.

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.

However, it should be noted that currently e-mails are not unique. Usage of shared e-mails might be problematic from the *End User* point of view. Maritime Applications are strongly encouraged to take measures to address this constraint.

---

### **3.10. MAP INTEGRATION**

---

A considerable effort has been made to integrate all EMSA Maritime Applications under the same entry point – known as MAP (Maritime Application Portal).



### 3.10.1. MAP login Process

MAP has the login screen being directly integrated inside the Portal. With MAP, while the user is still not authenticated he will see a login form on the first page of the Portal where he can directly insert his credentials and proceed to authenticate. All subsequent messages (invalid credentials, etc.) will be displayed in the same space giving the user the impression that he never leaves the screen. For compatibility purposes, for those applications still not integrated into MAP (LRITDC for example), a login screen similar to the MAP layout will be displayed. This will be discussed in the next sub-section.



**Figure 6: MAP integrated Login**

### 3.10.2. MAP Access Policies

To cope with this integration, in OAM an URL resource was created **"/mapLogin"** that is associated to a Policy named **"MAP Login"**. This policy continues to have *Form Based Authentication*, redirecting to the following URL in case of failure:

- /web/guest/home?p\_p\_id=login\_WAR\_emsamaploginportlet&p\_p\_lifecycle=0&\_login\_WAR\_emsamaploginportlet\_failed\_login=true

---

## 3.11. JSON LOGIN

---

Originally all EMSA MarApps were web applications that were accessed via internet browser. Over time, due to business and technological advances, some of the MarApps are (also) accessed via dedicated applications running in mobile devices. Two such examples are *Thetis Mobile Application* and *IMS Mobile*. Even though the underlying technology is different in both cases, what we describe next is valid not only for these two cases but also for any other application that wishes to use the same strategy.

### 3.11.1. User Authentication

In at least one of the cases described previously, i.e. *IMS Mobile*, despite being created as a stand-alone application; it still has the need to access business services supplied by EMSA's infrastructure. Basically, this means that a person using the application will have to identify himself as a recognized user, both to allow actual access to the application as well as to provide boundaries for what information the user can access. As such, the user must be authenticated against EMSA's infrastructure and later authorized to access certain functionality or view determined data. The easiest way to have a clear perception of the user and identify his access rights is using EMSA's SSO infrastructure.

To achieve this goal, EMSA provides a very simple "pseudo web service" that accepts as input the user's identification and his credentials. The information is posted to a URL that processes the information and effectively logs the user in returning success or not logging the user in and returning error. The return information is in the JSON format.

#### Login URL

The URI to access pseudo web service and attempt a user login is `/mobileLogin`. Please note that this URI only accepts POST requests and should contain two variables: *userid* that effectively contains the user's id and *password* that will contain the password for the user's account.

#### Return values

Under normal circumstances, once a POST has been executed to the aforementioned login URI, one of two things can happen:

- The user is authenticated correctly in which case a JSON message consisting of { **Status**: "success" } is returned;
- The user is not authenticated correctly, or the actual *userId* does not exist, in which case a JSON message consisting of { **Status**: "error" } is returned;

The reason this service has been labelled as a "pseudo web service" is because there are certain conditions that will trigger an HTML response instead of a JSON response thus defying compliance to the definition of a web service. The causes of return of such HTML pages are enumerated below. Please note that all these responses should be treated as the user not being logged in. The possible causes are:

- The user is locked out due to having failed his password too many times;
- The user's password is about to expire so a warning of such is sent from OAM;
- The user's password has expired, and a new password should be set;

These cases should be dealt with/resolved by normal access to the SSO login page via a browser.

### 3.11.2. User Authorization

Once the user is correctly authenticated, the MarApp using the JSON services can obtain information on the user by invocation of the corresponding services described in 4.4.2 User Information Web Service (or "PULL" Model).

## 4. Identity Management

---

### 4.1. EMSA BUSINESS VIEW ON IDENTITY MANAGEMENT

---

The first version of Identity Management implemented at EMSA was mainly based on an RBAC model (Role Based Access Control) with the user's attributes being spread out in vertical silos (i.e. the MarApps). As more and more MarApps became integrated with IdM, it became evident that there was a set of common attributes that should be the same (instead of the existing value per MarApp model). It was also apparent that an alternative structure to the RBAC model would be beneficial for some MarApps. When it became an absolute necessity to upgrade the underlying platform to a newer version, the opportunity was seized to execute these changes. The latest version of EMSA's IdM now has a common set of attributes per user account as well as providing support for a set of Business entities as described in the following sub-chapters.

#### 4.1.1. Service

A **Service** is a logical entity that represents a set of (one or more) Business Functions typically implemented by an application<sup>3</sup> (MarApp). In the context of the account management, it facilitates the logical discovery of a list of Profiles by filtering those visible or available for choosing. One example is the Thetis Service which has all the Thetis Roles mapped as Profiles and subsequently associated to the Thetis Service.

#### 4.1.2. Profile

A **Profile** is a group of one or more Roles logically combined or aggregated together such that they can be assigned/de-assigned to a User Account, all at the same time. It should be considered as a very high-level logical abstraction of a job function executed by a user.

#### 4.1.3. Role

In the context of EMSA's IdM, a **Role** is a low-level entity that is interpreted in one of two ways, depending on the MarApp or system supporting the role.

For some MarApps (such as Thetis or STCW) a Role is a logical definition of the function assumed or part played by a person or thing in a particular situation. An example of this is a THETIS\_INSPECTOR that is a person that has the function of performing inspections of ships according to the PSC regulations. One other example is an LRITDC\_ADMIN that is a person that manages the LRITDC MarApp.

Another possible interpretation for the Role is to consider it as a group of permissions that grant or deny access to specific resources. Roles facilitate the assignment of multiple permissions to a User Account. Please note that Permissions themselves are out of scope of IdM. An example of this interpretation is the role VIEW\_ABM whose description is "View ABM Alerts". The intent behind this role is to allow a person to view ABM alerts and not that of having a function of spending the time viewing ABM alerts.

---

<sup>3</sup> Please note that a Service can be implemented by more than one MarApp but in those cases there is always a principal MarApp providing the base functionalities. Please also note that a Service can be implemented via a horizontal platform or system such as Liferay Portal or LDAP



#### 4.1.4. Country/Institution

In the context of EMSA's IdM, the **Country/Institution** defines the "Nationality" (in a broad sense) of a User thus allowing the establishment of an area of control for a *National Administrator* (see 4.2 Security Model). Please note that in EMSA's context, an Institution - such as EFCA for example, is also considered at the same level as a Country.

#### 4.1.5. Organization

At EMSA, the concept of an **Organization** is a sub-entity of a Country or Institution. The Organization a user belongs to is used to establish not only an area of control of a *Local Administrator* (see 4.2 Security Model), but also to filter Profiles and Operations available to be assigned to accounts.

#### 4.1.6. Operation

In EMSA's IdM it's possible to assign **Operations** to an account. In Business terms, an Operation defines an Action that is available to a User in a given context (MarApp). Not all MarApps support Operations, and IdM is completely agnostic to their values and meaning. The list of Operations available to a given user is dependent on that user's Organization.

---

### 4.2. SECURITY MODEL

---

EMSA's IdM is the repository for the account information for users, as well as accumulating as a repository for generic access information to be used by MarApps. It also provides services to access these sets of information. As such, IdM is itself an Application and needs to have its own set of business rules to regulate who can do what in the IdM application. In IdM, the foundation for this regulation is the Security Model which establishes the management relationships (who is entitled to create/edit other users) and the permission rules (which serve as filters for limiting who a user can administer) or, said in another way: 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 of people as it implies knowledge of a specific skill-set.

2. **EMSA Service Administrator**

Identity Managers for a specific Service. Users belonging to this level are entitled to manage ***user accounts related with a specific Service*** (i.e. the services defined as those he is administrating). "EMSA Service Administrator" can only be assigned to a person belonging to EMSA and should be limited to a small number. It should be noted that a single person can be associated (i.e. have this level) with more than one service.

3. **National Service Administrator**

Identity Managers for a specific Country/Institution relating to a specific service. Users belonging to this level are entitled to manage ***user accounts that are simultaneously related with the Administrator's Service 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 Service Administrator

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

#### 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 of all levels. Most notably the **Local Administrator** is rarely used by most applications.

#### 4.2.1. 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 functional 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**.

The explicit relationship establishment is done through role mappings, i.e. each role is assigned a security level value. 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 the "Thetis System Administrator" role has a security level value of "EMSA Service Administrator". This means that whenever a user is assigned that Role, he will become an "EMSA Service Administrator" for Service Thetis (as this Role is associated to this Service).

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

#### 4.2.2. Accumulation of Levels

An important misconception is that an account has one (and only one) security level. While it's true that if a user has various Roles assigned to him (via Profiles), he will have the highest security level of all those Roles, this must be seen in the context of the Service to which the roles belong. It is possible for a user to be, for example, a National Service Administrator for one given Service while still being an End-User for another distinct Service.

---

### 4.3. IDENTITY MANAGEMENT FUNCTIONALITIES

---

For a User to have access to a MarApp with the appropriate permissions, IdM must handle User Account Management, including Provisioning of User Attributes.

While some operations of the application are performed following an automated process, most require either the intervention of or the initiation by a user. The following chapters demonstrate the various aspects of Identity Management.

#### 4.3.1. Reconciliation

The Reconciliation functionality of IdM is responsible for importing data from external systems, necessary for configuring a User Account. Examples of reconciliation of data is the list of Countries/Institutions from CBR (Country Base Registry), Organizations from COD (Central Organization Database), and low-level information used for provisioning from the Staging Area Database.

It should be noted that IdM is the authoritative source for User information.

#### 4.3.2. Account Management

Account Management is the name given to the set of actions that may be performed on a User Account to Create, Remove, Update/modify, Delete/disable. Besides the typical CRUD functionalities available, as part of IdM it is also possible to Search for Accounts, view the relationships between Services / Profiles and Roles, view the auditing information on changes made to accounts as well as recovery of failed provisioning attempts.

#### 4.3.3. Provisioning

The act of provisioning is the process by which IdM provides the updated User Account data to all the MarApps/Systems the affected User has access to (possesses a Profile/Role for). Please note that this effectively corresponds to the "PUSH" model described in 4.4.1 Provisioning Applications or "PUSH" Model.

#### 4.3.4. Other Administrative Functions

In this category are other operations not included in the other groups and are available only to the highest Administrator level, such as Reporting, Exports and Bulk Load.

---

### 4.4. IDENTITY MANAGEMENT INTEGRATIONS

---

EMSA's Identity Management system (IdM) is fully integrated within EMSA's applicational infrastructure. This effectively means that it can be accessed via Single Sign-On like any other MarApp/System. It can send information (i.e. invoke services or return data via invoked services) to other existing MarApps/Systems and finally, it can also receive requests to moderate its behaviour (i.e. provide functionalities upon request). The following chapters describe some of these aspects.

#### 4.4.1. Provisioning Applications or "PUSH" Model

One of the goals of having an Identity Management solution in EMSA is to have a common way of provisioning users to applications. This essentially means that whatever can be found common to all possible applications should be stored locally in IdM and thereafter provisioned to each MarApp that the user is effectively a member of (i.e. having a relevant Role in that MarApp).

EMSA's Maritime Applications are provisioned by IdM to contain user information. This mechanism can be roughly described as being a "PUSH" mechanism in that EMSA's Identity Management system effectively sends information on new users (or changes made to existing users) to the appropriate systems/MarApps for which this information is relevant. The way this is done is through invocation of a series of established Web Services available

in the MarApps (or eventually using another form of API such as an LDAP connection). IdM attempts to guarantee that every system/MarApp is kept up-to-date with the latest information on a user, be it personal attributes such as first name, etc. or authorization information such as Profiles (via Roles).

#### 4.4.2. User Information Web Service (or “PULL” Model)

It should be noted that in the EMSA eco-system of Maritime Applications and horizontal platforms, the “PUSH” mechanism may not always be the best solution for user management. There are cases in which the actual MarApp is significantly (or even completely) agnostic about users. One such example is STCW which has the need for “knowing” users and their personal attributes (such as their “Country” for example) but does not actually keep any information about them. Another much more radical example is RuleCheck that, in its current form, has absolutely no concept or knowledge of users. RuleCheck’s content is served to users in a differentiated model by strategic use of the Access Management component of IdM (the OAM) allowing or denying users to see certain content. Still another example is the SEG (SafeSeaNet Eco-system GUI) that only needs to know what the accessing user can or cannot do and see (i.e. his Profiles/Roles and Operations). This necessity led to the establishment of a new architectural model, the “PULL” model. The “PULL” model is none other than a service supplied by IdM allowing any MarApp to request information about a given user.

#### UserInfo REST Web Service

Currently the “PULL” model is implemented through a REST Web Service that is invoked via a normal HTTP GET call passing the user’s identification and returning information in the JSON format.

The UserInfo REST Web Service can only be called from within EMSA’s infrastructure so no data leakage can occur to entities outside of EMSA. The context path of the service is not available through any URL that can have public access. There is currently no assumption made about the user invoking the service, so no authentication is done.

The UserInfo Web Service will typically return one of two possible sets: a null message if the user ID passed as the last parameter does not exist in IdM, or a JSON message containing information on the user ID passed.

#### 4.4.3. Accessing IdM Functionalities via direct URL

EMSA needs to access OIM directly from links placed in some applications, namely MarApps and Liferay Portal. A bespoke module has been developed to allow such direct access.

#### Search User

The URL for “jumping” into IdM directly in the Search Users Functionality is

`/identity/faces/home?tf=SEARCH_USERS`

Through this URL authorized users can execute a search for one or more accounts and then proceed to execute other actions (view, edit, etc.)

#### Create User

The URL to access the Create User Form is

`/identity/faces/home?tf=CREATE_USER`

### **Edit User**

The URL to access the Edit User Form window is

`/identity/faces/home?tf=MODIFY_USER&userLogin=<User Login>`

In the previous link, *<User Login>* must be changed to the correct user identification per call to the edit method.

### **Edit my account (only fields common to all applications)**

The URL to access the Edit my account User Form window is

`/identity/faces/home?tf=MODIFY_USER`

You might notice that this link is similar to the Edit User link except for the fact that no UserLogin identification is provided and as such the account of the user accessing is displayed.