

System and Application Technical Landscape

APPENDIX A

Document History

Version	Date	Names	Comments
36	23/11/2023	D 3	

TABLE OF CONTENTS

1. Introduction and Objectives.....	7
2. System Landscape.....	8
2.1 High Level Network Schema.....	8
2.2 Data Links	8
2.3 Network Security	9
2.4 Proxy Policy.....	9
2.5 Network Load Balancing	10
2.6 High Level Virtual Infrastructure Schema	10
2.7 Virtual Infrastructure Services.....	11
2.8 Application Requirements For Virtual Infrastructure.....	11
2.9 Environments	11
2.10 Disaster Recovery	16
3. Application Landscape.....	19
3.1 Architecture Overview	19
3.2 Client Environment and Client Tier	20
3.2.1 Web Browser Environment	20
3.2.2 Client Application	21
3.2.3 External Systems	22
3.3 Application Environment	22
3.3.1 Application Server	22
A. Web Tier.....	22
B. Business Tier	23
3.3.2 EIS Tier	24
A. Database	24
B. Message Oriented Middleware	25
C. Other Information Systems	25
D. Authentication and Authorization	25
3.4 Security	25
3.5 Reporting Platform	26
3.6 Geographic Information System AND OGC (Open Geospatial Consortium) standards	26
3.7 Logging.....	27
3.8 Storing Times and Dates.....	27
3.9 Others.....	27
4. Service Oriented Architecture	29
4.1 Service Consumers.....	30
4.2 Shared Service Infrastructure	30
5. Software Versioning Scheme.....	31
6. Summary	32

List of Abbreviations

Acronyms	
AJAX	Asynchronous JavaScript and XML
BCF	Business Continuity Facility
BMP	Bean-Managed Persistence
CMP	Container-Managed Persistence
DAO	Data Access Object
DTO	Data Transfer Object
DB	Database
DC	Data Centre
DHTML	Dynamic HTML
DMZ	Demilitarized zone
DNS	Domain Name System
EIS	Enterprise Information System
EJB	Enterprise Java Bean
EMSA	European Maritime Safety Agency
ESB	Enterprise Service Bus
FTP	File Transfer Protocol
GIS	Geographic Information System
GUI	Graphical user interface
HA	High Availability
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IPSEC	Internet Protocol Security
ISP	Internet Service Provider
JCA	JAVA EE Connector Architecture
JDBC	Java Database Connectivity
JDK	Java Development Kit
JEE	Java Enterprise Edition
JMS	Java Message Service
JSF	Java Server Faces
JSP	Java Server Pages
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
Mbps	Megabit per second

MOM	Message Oriented Middleware
NAT	Network Address Translation
OAM	Oracle Access Management
OIM	Oracle Identity Management
OES	Oracle Entitlement Server
OS	Operating System
OSB	Oracle Service Bus
OWASP	Open Web Application Security Project
POJO	Plain Old Java Objects
R. Proxy	Reverse Proxy
RAC	Real Application Clusters
REST	Representational State Transfer
RIA	Rich Internet Applications
RMI	Remote Method of Invocation
SAN	Storage Area Network
SANS	SysAdmin, Audit, Network, Security Institute
sFTP	Secure File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
SRM	Site Recovery Manager
SOA	Service Oriented Architecture
SSL	Secure Socket Layer
TB	Tera Bytes (i.e. 10^{12} bytes or 1 million mega bytes)
UDDI	Universal Description Discovery and Integration
VLAN	Virtual Local Area Network
VM	Virtual Machine
WLI	WebLogic Integrator
WLS	WebLogic Server
XHTML	Extensible Hypertext Mark-up Language
XWS	WS Security implementation from Sun Microsystems

List of Annexes

List of Annexes	
Annex 1	“IAM Guide_abridged”
Annex 2	“EMSA secure development requirements v01”
Annex 3	“EMSA secure development recommendation guide v01”
Annex 4	“EMSA_JASPER_Technical_Document”
Annex 5	“EMSA SOA Guidelines & Rules”

1. Introduction and Objectives

This document describes EMSA System and Application landscape. Its main objective is to document the technical solutions used by EMSA at System level and to provide directions on options and preferable technologies to be considered at Application Level.

Although the System and Application Landscape described in this document are EMSA guiding lines, this does not mean that no deviations are allowed.

Exceptions can be proposed and they will be considered on a case by case basis; if it is found that is the best technical implementation for the requirement or there is no other way of doing it, this exception will be accepted.

Also suggestions for innovation are welcome and if they bring added value to the landscape, they will be included.

The document is organized in several chapters:

- Chapter 1: Introduction and Objectives.
- Chapter 2: Describes the System Landscape and the Technical solutions implements at systems and network levels.
- Chapter 3: Describes the Application Landscape and preferable options to be used at the Application level.
- Chapter 4: Describes the conceptual Service Oriented Architecture (SOA) to which the applications should comply
- Chapter 5: Describes the software versioning scheme
- Chapter 6: A summary of the Software versions

2. System Landscape

2.1 High Level Network Schema

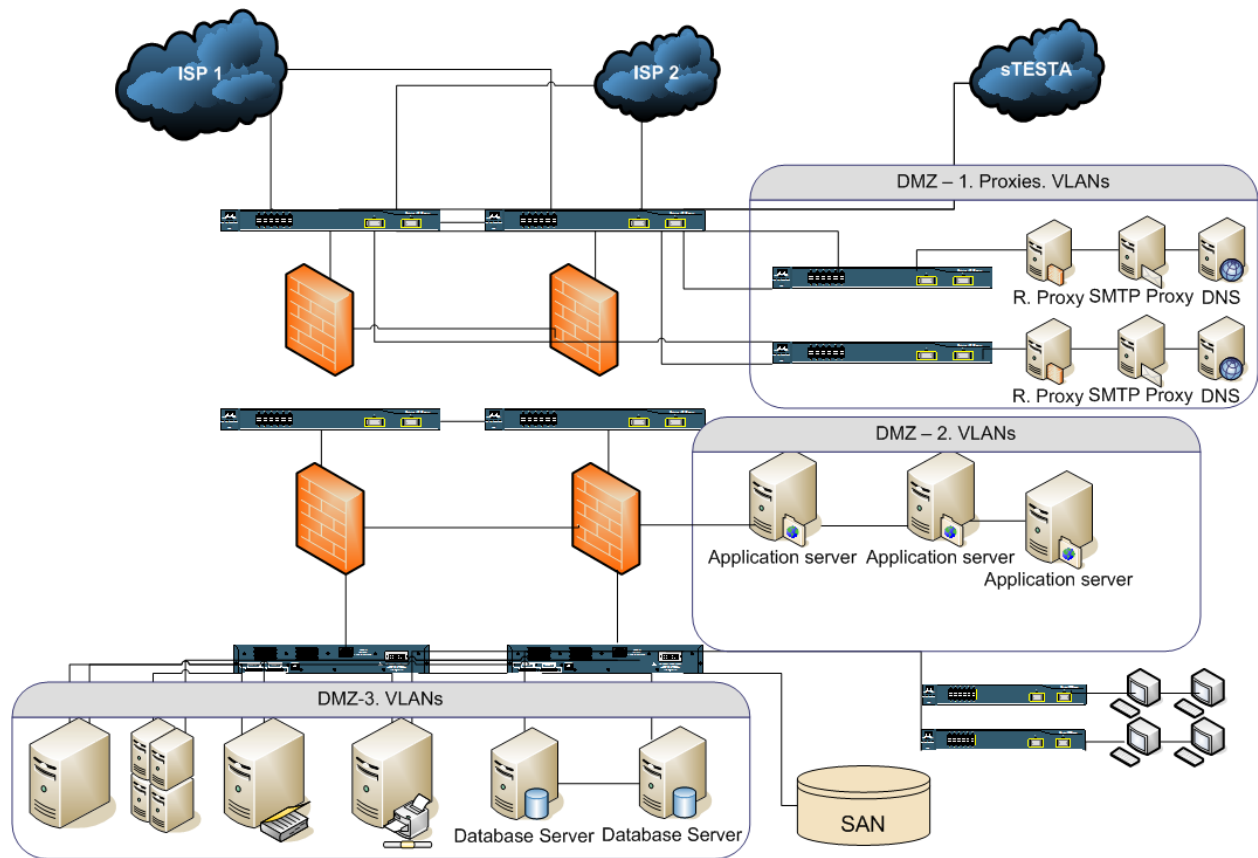


Figure 1 EMSA Primary site. High level network schema

2.2 Data Links

Table 1 Data Links

Data Links	
■	2 Internet ISP <ul style="list-style-type: none">– active/active using BGP– BGP autonomous system and routing fully managed by EMSA– 100 Mbps each– 256 Provided independent IP addresses
■	1 sTESTA link <ul style="list-style-type: none">– EU private network– 2 Mbps
■	1 GEANT link <ul style="list-style-type: none">– Reserved to the CleanSeaNet project for high speed image transfer– 1 Gbps

2.3 Network Security

Two layers of firewall protection:

- Checkpoint R75.40 2-nodes clusters;
- Cisco ASA;

Reverse proxies for incoming connections (currently handling the following protocols: HTTP, HTTPS and SFTP).
The network is segmented using VLAN's.

Table 2 DMZs

DMZs
<ul style="list-style-type: none">■ DMZ-1: reverse proxies, DNS servers, other services exposed to Internet■ DMZ-2: application servers and database servers (Front/Back End VLANs)

Monitoring of security events is currently achieved through a SIEM (Security Information Event Management) system including Suricata, Splunk, F5 ASM module on top of EMSA F5 reverse proxy.

2.4 Proxy Policy

The following rules should be followed:

- Accessing EMSA web applications should be always through HTTPS;
- Reverse proxies are used for all incoming connections from outside networks (Internet and sTESTA);
- All incoming connections shall pass through our reverse proxies;
- All incoming SSL connections are terminated in the reverse proxies;
- Proxies are always responsible for the SSL encryption and decryption;
- Proxies are always responsible for creation of the SSL connections;
- 1-way SSL is used for human to system interfaces while 2-way SSL should be used for system to system interfaces;
- All SSL outgoing connections shall use the proxy. Any outgoing SSL connection shall be initiated as plain HTTP by the applications to the proxy, where the SSL will be initiated for the outgoing SSL connection. The protocol used to request the proxy the creation of an outgoing HTTPS connection, involve the usage of an EMSA URL naming convention (<standard_URL>.f5 URL's) and some F5 configurations.

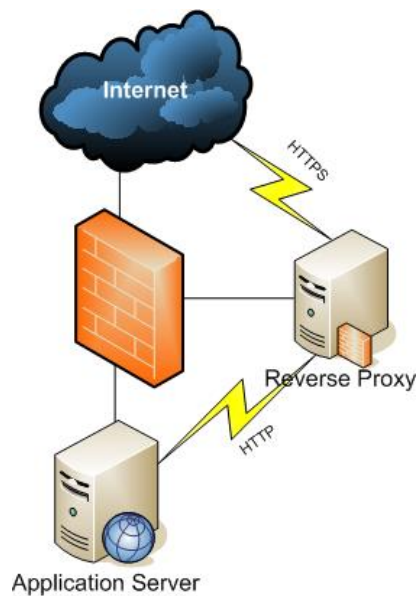


Figure 2 Proxy policy

Table 3 Proxy Devices

Proxy Devices	
	■ 2 x F5 Big IP v5000 Series

2.5 Network Load Balancing

The F5 appliances form a redundant cluster that can perform load balancing for web applications in any VLAN on EMSA network. The design of any new system or application should preferably implement load balancing with node fail detection on this equipment.

2.6 High Level Virtual Infrastructure Schema

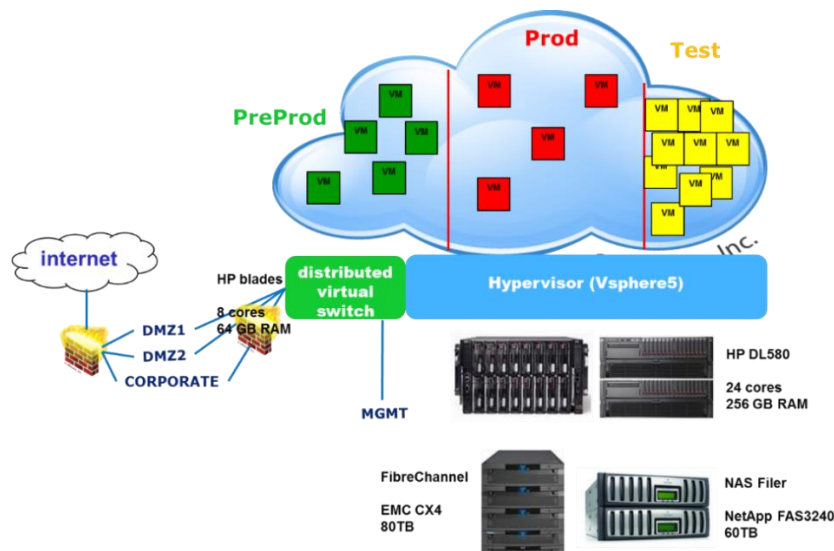


Figure 3 High Level infrastructure

2.7 Virtual Infrastructure Services

The following services are offered to VMs and application environments:

- Basic monitoring with Nagios;
- Performance monitoring with vCenter Operations;
- VM-level backup with Networker or Netapp SnapMgr for Virtual Infrastructure. Exceptionally also Networker agent-based backup can be implemented.
- Deployment of a VM or environment¹;
- Cloning of a VM or environment;
- Snapshotting of a VM or environment²;
- Exporting as OVF a VM or environment;
- Hardware resource allocation changes³;
- Upgrade of VMware tools and virtual hardware;
- Troubleshooting.

2.8 Application Requirements For Virtual Infrastructure

Applications and systems hosted in the EMSA Virtual Datacentre must respect the following requirements:

- Base OS must be chosen out of the current EMSA template catalogue⁴;
- Compatibility with the latest VMware virtual hardware specifications (currently version 8);
- Hardware provisioning done according to a principle of fit-for-purpose;
- Compatibility with vMotion.

2.9 Environments

EMSA has defined 6 possible different types of environments for the Maritime Applications. The following picture presents an overview of them.

¹ Subject to being included in the EMSA Template catalogue, currently including:

- Linux Red Hat Enterprise Server or CentOS in version 7;
- As above, with WebLogic or with Oracle DBMS;
- Latest Microsoft Windows servers.

² Subject to the following policy: the snapshot must be rolled back, or removed, in one week time to avoid performance penalties;

³ Subject to the following policy: CPU, Memory, disk and network for any VM should be fit for purpose, and oversized VMs should be avoided to reduce contention issues and overhead. Granting more resources is subject to a trend analysis of the use of current resources also looking at vCenter Operations performance indicators, and takes into account its recommendation. VMs oversized are reported on a regular basis and are subject to downsizing.

⁴ See note 1 on the previous page.

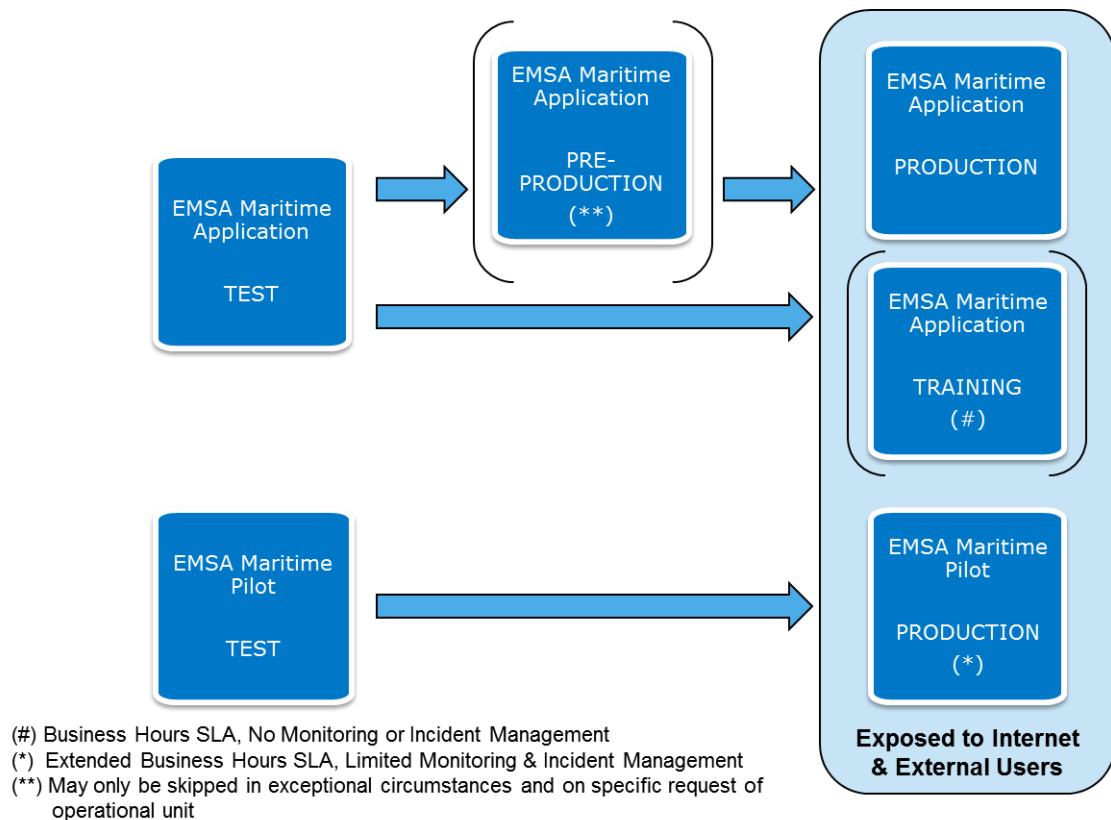


Figure 4 Types of Environments

The following figure shows detailed information related to each type of environment.

Type	InfraX grade	A.3 Monitoring	A.3 Incident Management	External access	Backup / Restore services (1)	Decommission Date	Ownership	VLANS
Test (t)	Test	No	No	No	Non Prod	No	Ops. Unit / Contractor - A.3 provides infraX + grant privileges - Ops.Unit handles it afterwards	Test
Pre-Production (pp)	Pre-Prod	No	Yes (Business Hours)	No or with IP Filtering	Non Prod	No	A.3	Pre-Prod
Training (I)	Pre-Prod	No	No	Yes	Non Prod	No	Ops. Unit - A.3 provides infraX + grant privileges - Ops.Unit handles it afterwards - A.3 act as support if needed	Pre-Prod
Production (p)	Prod	Yes (infraX and application outputs)	Yes (24 x 7)	Yes	Prod	No	A.3	Prod
Test Pilot (t)	Test	No	No	No	Non Prod	Yes (max. 1 year)	Ops. Unit / Contractor - A.3 provides infraX grant privileges - Ops.Unit handles it afterwards	Test
Pilot (i)	Prod (2)	Limited (max 5 checks per env, based on app outputs)	Yes (Extended Business Hours) MSS – normal A3 – major	Yes	Non Prod	Yes (max. 1 year)	Ops. Unit - A.3 provides infraX + grant privileges - Ops.Unit handles it afterwards - A.3 act as support if needed	Pilot

(1) - Backup/Restore services:

- Non-Prod: VM: weekly + DB (RAC): constant / DB (non-RAC): weekly + File System (data): weekly
- Prod: VM: weekly + DB (RAC): constant + File System (data): daily

(2) – Production for Pilot Projects:

- only uses VMs with standard A.3 templates
- No clustering / load balancing
- Single instance DB (no RAC)

Figure 5 Characteristics per Type of Environments

The basic infrastructure that supports the environments is as follows:

Table 4 Environments

Environments
<ul style="list-style-type: none"> ■ Production ■ Training: ideally 50% of the production capacity ■ Pilot Production: ideally 50% of the production capacity ■ Pre-Production: ideally 50% of the production capacity ■ Test/Quality: ideally 25% of the production capacity

Table 5 Server Infrastructure

Server Infrastructure
<ul style="list-style-type: none"> ■ EMSA Datacenter is fully virtualised with VMWare technologies ■ Those include: <ul style="list-style-type: none"> – VMware ESXi VSphere 5 – VMware HA, DRS and Failover

Table 6 High availability technologies

High availability technologies
<ul style="list-style-type: none"> ■ Service fail-over: Weblogic Active-Active, Oracle EXADATA ■ Server fail-over: VMware FailOver and VMware HA ■ Site fail-over: VMWare Site Recovery Manager ■ Data replication: Asynchronous data replication via FCIP; backup storing off-site

Table 7 Service Clustering

Service Clustering
<ul style="list-style-type: none"> ■ Weblogic Active/Active clustering ■ Oracle EXADATA

Table 8 SAN Storage

SAN Storage
<ul style="list-style-type: none"> ■ Brocade fabric based on Sanswitch DS5300 ■ EMC Clariion CX4-240 ■ Netapp filer FAS3240 (only CIFS/NFSv3)

Critical applications and services must be mandatorily designed following High availability techniques (e.g. clustering) without any Single Point of Failure.

Table 9 Environment characteristics

Environment	Test / Test Pilot	Pre-Production	Training	Pilot Production	Production
Purpose	This environment allows software contractors to perform testing and integration of their applications in the EMSA environment.	This environment offers a chance for EMSA application users to review and test applications in development or having past SAT.	This environment is used to perform training sessions with the end-users and MS commissioning tests.	This environment is used to implement new applications to validate new concepts before implementing a full-production system.	<p>Shall only be provided for applications whose deliveries have been formally accepted.</p> <p>When an application is no longer in use, the application owner shall inform the ICT team of this change in status.</p>
Infrastructure performance & scaling	Equivalent to 25% of production capacity	Equivalent to 50% of production capacity	Equivalent to 50% of production capacity	Equivalent to 50% of production capacity	
Responsibility and installation	In test environment the contractor will have the necessary privileges (limited to areas directly related to the development) in order to be able to deploy the application under development without help from ICT teams. On request, ICT may make available staff to support the contractor.	The environment shall also be used to test installation procedures. Before any applications are installed or before configuration changes, data fixes, etc are performed, the contractor will deliver to EMSA all source code, installation scripts, installation procedures, release notes, etc, as described in the release management procedure. ICT will be responsible for installation and therefore the	In training environment the Operational Units will have the necessary privileges (limited to areas directly related to the development) in order to be able to deploy the application under development without help from ICT staff. On request ICT may make available staff to support the contractor.	In Pilot Production environment the Operational Units will have the necessary privileges (limited to areas directly related to the development) in order to be able to deploy the application under development without help from ICT staff. On request ICT may make available staff to support the contractor.	<p>All software or scripts being run in the production environment shall first be installed in pre-production environment. Both EMSA business responsible and EMSA IT responsible shall have formally accepted the software in accordance with Software Release Management Procedure.</p> <p>Installation and maintenance will be performed solely by ICT or its contractors.</p>

		contractor or EMSA project officer will need to arrange with ICT, sufficiently beforehand, a date for installation.			
--	--	---	--	--	--

2.10 Disaster Recovery

EMSA's Business Continuity Facility (BCF) is hosted in the premises of a commercial hosting provider. The BCF is a fully equipped replica of the main site in terms of servers, network equipment, internet connectivity, storage and middleware, and as such it may function as either the main production site for an application, or as back-up site. This choice may be made on a per application basis and depends on the EMSA needs, the application's replication design and capabilities, and the desired SL.

Any new system or application must conform by design to one of the business continuity approaches foreseen so far:

1. ON/OFF model:

The servers and services that constitute the system or application are active and visible on the network only in the main site. They are kept in sync in the secondary site with some middleware or low level replica technology like Dataguard for backends, or virtual machine cloning or storage array based replication for front ends. But the replicated systems are always inactive on the secondary site in an off-state and not visible on the network unless the recovery procedure is executed. Taking over in that case means executing a procedure to stop the systems in the main site (if possible), execute a last synchronisation (if possible), stop the synchronisation flows, then restart the replicated systems in the secondary site changing all the parameters that differ in the two sites like network configuration, internal DNS entries, pointers to database or cartographic servers or to any other horizontal service platform always available in both sites like LDAP, Single Sign On, DNS etc.... Eventually, the external DNS entry should be changed to point external Internet users to the public IP of the system or application in the new site.

According to this model, it is still possible to have the same internal FQDN for the application servers in both sites, as servers are active and visible on the network only in one site at a time, and when taking over, the A records of the internal DNS can be changed to reflect the different IP address space in the new site.

2. ON/ON model:

The servers and services that constitute the system or application are active and ready to take over at any time in both sites. Synchronisation relies on the features of the application or middleware used rather than on a low-level cloning and transferring of the virtual machines, offering either a fully multi-master active/active approach like Active Directory, or some type of distributed geo-cluster, or anyway an autonomous system which keeps data and configuration in sync between the two legs in the two sites. Taking over in that case is a simpler procedure like activating some built-in system or application feature to switch to the other site, possibly requiring some internal and external DNS changes, or can be even fully transparent.

According to this model, different FQDNs and IPs for the application servers in the two sites must be chosen, as servers are active and visible on the network in both sites at any time.

Note: it is not accepted to design ON/ON systems where the virtual machines on the two sides have the same internal DNS FQDN.

The ON/ON model, when supported by the application or middleware, might guarantee faster and seamless fail-over procedure, hence it is the preferred approach.

The following figure exemplifies how the interconnection of current EMSA's production environment with the BCF is envisaged and points to the use of several replication/back-up systems at different levels of the infrastructure:

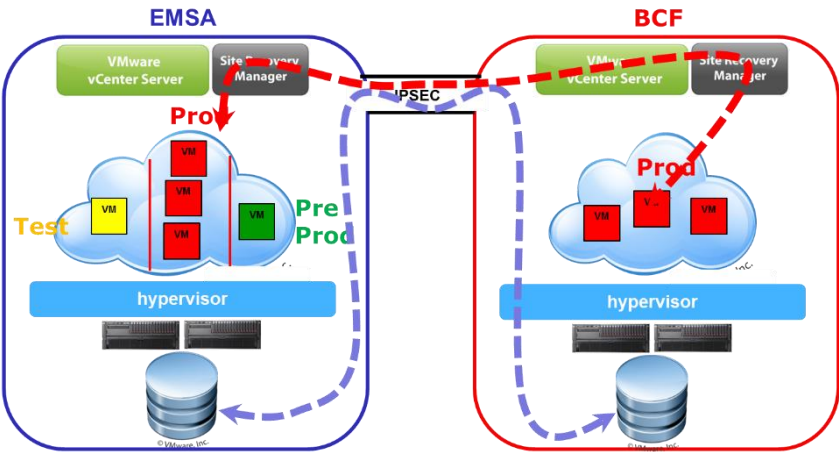


Figure 6 EMSA DC connection with BCF

Key elements of the actual BCF architecture are:

1. the two sites are connected through an IPSEC tunnel over an high performance link
3. the two sites use different private and public IP address ranges
4. the internal DNS zone emsa.local, containing server's FQDN, is shared between the two sites;
5. the external IP address space in each of the two sites is a different C-class of Provider Independent IPs whose routing advertisements is managed directly by EMSA routers
6. the external DNS zone "emsa.europa.eu" is unique across the sites, it is delegated to EMSA, and it is kept in sync between the two sites with master-slave DNS replication;
7. data and systems are kept in sync through either:
 - A. Oracle Dataguard for backend;
 - B. Storage array replication for most of the front end virtual machines;
 - C. Ad hoc application built-in replication technologies, like active directory replication, or Microsoft continuous cluster replication for Exchange and SQL.
 - D. Ad hoc scripts for data transfer.
8. Rerouting of Internet users to the BCF is done with DNS technologies

Applications development should always be *BCF friendly* by being compliant with the following requirements:

- Application shall never use IPs in any configuration or dependency.
- All relevant configurations must be externalized from the application; this can be achieved with properties files in the filesystem (never inside the application war, ear or deployment directory) or using a well identified table in the database.
- Application shall use FQDN in their configurations or references to any dependency.
- Bandwidth required for data and system alignment should be kept to a manageable amount to allow continuous replication over a non-dedicated medium bandwidth link. A bandwidth estimation for data synchronization between EMSA DC and BCF, through Oracle Data Guard and other technologies, shall be provided;
- A fail-over procedure to BCF shall be provided together with one to fail back to EMSA;
- A list of all the application configurations and dependencies which need to be resolved in the BCF and main production site for the application to run shall be provided:
 - Web services
 - Data sources
 - Other application(s)
 - Security constraints
 - Infrastructural services
 - Etc...
- Connections to other machines should always be configured by referring to the machine name, never by referring to the IP address directly.
- For critical system, BCF certification is mandatory

3. Application Landscape

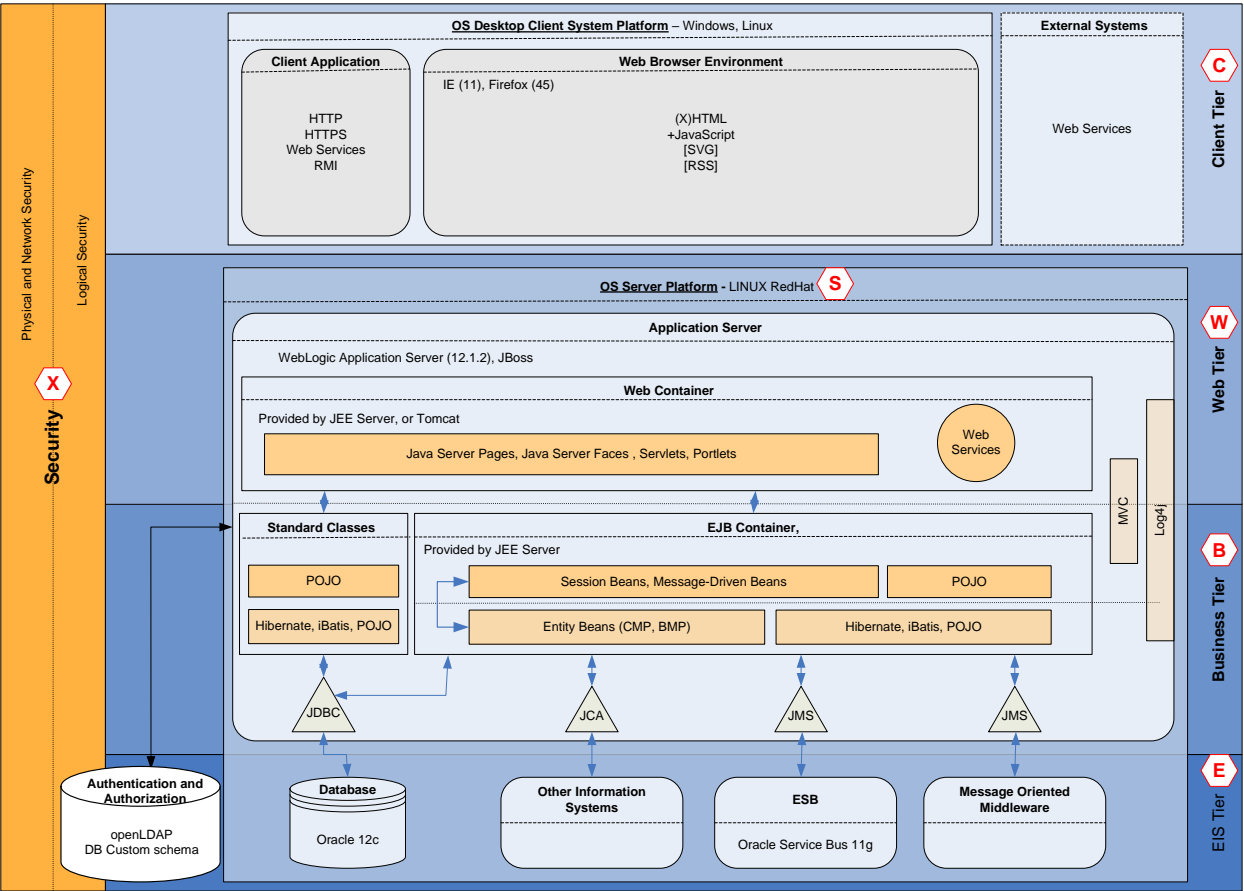


Figure 7 Application landscape

3.1 Architecture Overview

EMSA IT systems should follow state of the art JAVA PLATFORM, ENTERPRISE EDITION n-tier architecture. Error! Reference source not found. represents the preferable EMSA IT architecture where the major tiers are:

Table 10 Client Environment

Client Environment
Client Tier: Client Tier is a JEE application front-end that provides communication with human users or with others external systems. For details, refer to chapter 3.2

Table 11 Server environment

Server Environment
Web Tier: Web Tier connects user interface on a Client Tier with business logic on a Business Tier.

For details, refer to chapter 3.3.1, (a)
Business Tier: Business Tier provides transaction processing logic (business logic) and data processing logic (data management). Business processes and business components should not be implemented outside this tier. For details, refer to chapter 3.3.1, (b)
EIS Tier: EIS (Enterprise Information System) Tier consists of all enterprise information systems, such as databases or other information systems. ESB and Message Oriented Middleware are also included in this tier. For details, refer to chapter 3.3.2

Client Tier is the only tier of the Client Environment and it's by definition a distributed and separated tier.

Web Tier, Business Tier and EIS Tier are part of the Server Environment hosted at EMSA; EIS Tier (and its components) is usually a separated tier implemented on top of a separated server environment and depending on the complexity, the system architect may decide between a complete distributed architecture where all tiers are distributed in separated server environments or a mixed architecture where some tiers may share one server environment.

Operation systems options for the different environments are:

Table 12 Client Environment

Client Environment
<ul style="list-style-type: none">■ Windows 10■ LINUX distribution desktop

Table 13 Server Environment

Server Environment
<ul style="list-style-type: none">■ LINUX Redhat server 7 (64 bits)■ Windows Server 2019

3.2 Client Environment and Client Tier

3.2.1 Web Browser Environment

The majority of EMSA applications are delivered to the final user via a browser based interface. A Web UI's advantage is that no additional software needs to be installed on client side and minimal demands are placed on the client platform.

Because a HTML Thin Client GUI is limited by markup language / JavaScript capabilities, others resources can add to build Rich Clients providing better user experience through the Web Browser. Applications must be 100% compatible with, at least, the following browsers or higher versions:

Table 14 Web Browsers

Web Browsers
<ul style="list-style-type: none">■ Microsoft Edge (latest versions)■ Mozilla Firefox 70 and later

HTML page serves as a host for Rich Clients built with different technologies:

Table 15 Client Tier Technologies

Client Tier Technologies
<ul style="list-style-type: none">■ HTML 5■ Plain Javascript and Tag Libraries■ Single-Page Application, e.g. AngularJS, ExtJS,... (latest versions)■ WebGL

Technologies used to implement Rich Internet Applications in the Client Tier can also have strong relationships with the technologies used in the Web Tier (e.g. Tag Libraries) described in chapter 0.

Usage of Java Applets should be limited to very particular situations and the decision to allow this will be taken on a case by case basis.

3.2.2 Client Application

Due to some business requirements (e.g. operation in disconnected mode, access to the local file system, ...), some applications may require a Fat Client.

In order to create a unified technology platform, and to support all operating platforms in use at EMSA or EMSA clients, preference will be for using the Java language.

A mechanism for deploying and updating the client application at the remote PC will be needed (Java Webstart will be preferred). Dependencies on runtime components not already part of standard EMSA PC configurations will be regarded as negative.

Because EMSA needs to support other organisations within the Member States, any application to be installed on a client will need to be cross-platform, covering at least the platforms listed earlier in this document⁵.

Usually, a client application will need also to connect to the server side of the system in order to perform business actions (e.g. data synchronization). Several technologies can be used to address this client-server connection; please refer to Annex 5 “EMSA SOA Guidelines & Rules” for details.

Communications to servers shall be done using web services, exceptions may be granted on request. Exposed Web Services shall always be protected with Authentication and Authorization. Important business data should always be stored on servers managed by ICT, if this requirement cannot be met (due to business requirements, impossibility to connect, ...) a procedure for providing data back-ups needs to be foreseen.

⁵ If the application is to be used only by EMSA this requirement can be reduced to supporting Windows 10. An application installer compatible with EMSA's MS System Center needs to be provided.

In case development of a fat client is proposed, this needs to be discussed with ICT and agreements on installation requirements, connection technology and data back-up need to be reached before starting development.

Table 16 Mobile application platforms

Mobile application platforms
<ul style="list-style-type: none">■ iOS, latest versions■ Android, latest versions

Increasingly mobile devices are used for accessing web based information systems. Where possible, in order to avoid creating multiple platform dependent solutions, such developments should be based on simple website access, with appropriate changes applied to the UI to take into account the smaller screen size, reduced bandwidth and touch based controls used by mobile devices. In cases where business requirements cannot be reached using a mobile optimised website, at least the application platforms and version mentioned above need to be supported.

3.2.3 External Systems

External systems will also act as clients to EMSA systems creating the need of integrating different software systems used by different organizations (business partners). The system integration helps to automate collaboration processes and improve business performance. De-facto standard technologies should be used to inter-connect external systems with EMSA systems; please refer to Annex 5 “EMSA SOA Guidelines & Rules” for details.

3.3 Application Environment

3.3.1 Application Server

EMSA architecture is based on the standard JEE version 7. The following Application Servers should be used as the base Web and EJB containers:

Table 17 Application Servers

Application Servers
<ul style="list-style-type: none">■ Weblogic Application Server (latest version)■ Wildfly/JBoss (latest version)

New development or ‘significant’⁶ changes to existing applications should always target the latest version of the application server in use at EMSA. For existing applications, EMSA will assess the desirability vs the risks of upgrading the underlying application server on a case by case basis.

Simple applications, where distribution is not foreseen, the EJB container is not needed; see below for details.

A. Web Tier

⁶ Significant shall be understood as any change resulting in a change of either major or minor versioning number (see further for a description of the version numbering scheme in use at EMSA)

The delivery of Rich GUI based on Web Browsers is achieved by a set of components located in this tier and in close relationship with the Client Tier. Those components may vary depending on the technical solution adopted and level of complexity required for the Rich GUI; major technologies are presented in the next table:

Table 18 Web Tier Technologies

Web Tier Technologies
<ul style="list-style-type: none">■ JSP – Java Server Pages■ JSF – Java Server Faces■ Portlets■ Rich server side components

Table 19 Portal technology

Portal technology
<ul style="list-style-type: none">■ Liferay Enterprise Edition

Simple applications, that only require a Web Container can use:

Table 20 Web Container

Web Container
<ul style="list-style-type: none">■ Tomcat (latest stable version)

Web Services are used to provide communication between loosely connected system components and are the preferable mechanism to expose services to external systems/applications. Several technologies could be adopted; please refer to Annex 5 “EMSA SOA Guidelines & Rules” for details.

B. Business Tier

System functionalities are always implemented in the Business Tier and several technical options can be used to implement the Business components.

A software layer approach must be followed, implementing at least, two layers:

Business Layer: Responsible for the delivery of the business functionalities and orchestration of the business processes.

Data Access Layer: Responsible for isolation of data access and actions executed over the persistent data storage (typically a relational database). Usually, Data Access Object (DAO) design pattern is mapped into this layer.

To support data transfer between layers and even between tiers a complete set of objects according to the Data Transfer Objects design pattern must be implemented.

For simple applications where an EJB container is not required:

Table 21 Business Layer technologies

Business Layer technologies
<ul style="list-style-type: none">■ POJO (Plain Old Java Objects)

Table 22 Data Access Layer technologies

Data Access Layer technologies
<ul style="list-style-type: none">■ JPA■ JDBC■ Hibernate■ springJDBC

For systems requiring an EJB container (that will be provided by the selected Application Server):

Table 23 Business Layer technologies

Business Layer technologies
<ul style="list-style-type: none">■ Session EJBs■ Message Driven EJBs■ POJO (Plain Old Java Objects)

Table 24 Data Access Layer technologies

Data Access Layer technologies
<ul style="list-style-type: none">■ Hibernate■ springJDBC■ Entity EJBs

3.3.2 EIS Tier

A. Database

EMSA stores data in relational databases.

Table 25 Relational Database Management System

Relational Database Management System
<ul style="list-style-type: none">■ ORACLE 12c■ PostgreSQL 12

New development or significant upgrades should enable the application to use the latest RDBMS version in use at EMSA.

B. Message Oriented Middleware

To provide messaging services for integrated systems or asynchronous operations, EMSA relies on a Message-Oriented Middleware that increases the interoperability, portability, and flexibility by isolating the exposed services from the internal implementation and allowing distribution over multiple platforms (among other advantages).

Asynchronous messaging is the preferred method for exchanging data between internal applications. JMS will be the preferred manner for consuming and producing messages. The use of asynchronous message should enable better decoupling between applications (compared to web services), allow a more up-to-date system state (compared to batch processing), increased scalability (due to MOM underpinnings) and improved configurability and oversight of the system integrations (through use of the ESB). Asynchronous messaging over JMS will also be the preferred method for request/reply messaging paradigm.

Table 26 Message Oriented Middleware

Message Oriented Middleware
<ul style="list-style-type: none">■ WebLogic JMS

C. Other Information Systems

Any other Information Systems inside EMSA is considered to be in the EIS tier. Integration can be done using several techniques; preferable methods of integration are:

Table 27 Internal systems integration technologies

Internal systems integration technologies
<ul style="list-style-type: none">■ JCA – JAVA EE Connector Architecture■ Web Services; please refer to Annex 5 “EMSA SOA Guidelines & Rules” for details.

Asynchronous communication (based on call backs) should be used where possible.

Compared to the JMS based integration described above, more effort will be required to ensure the consumers / producers deal with service unavailability, scalability or reliability issues, therefore integration using asynchronous JMS is encouraged.

D. Authentication and Authorization

EMSA owns a centralized system for Identity and Access Management; for details on this system, please refer to Annex 1, “IAM Guide_abridged”.

3.4 Security

Implementation of EMSA applications shall follow and be compliant with the best practices for secure programming. The standards detailed in Annex 2, “EMSA secure development requirements v01” are mandatory and recommendations described in Annex 3, “EMSA secure development recommendation guide v01” must always be taken into consideration

All applications shall be assessed against those recommendations and standards. These security assessments will be conducted by EMSA together with an independent external partner, at least once before entering PRODUCTION and whenever there is a EMSA’s decision to carried out a new assessment. Vulnerabilities found shall be addressed by the application implementing partner in agreement with EMSA.

3.5 Reporting Platform

Table 28 Reporting Platform

Reporting Platform
<ul style="list-style-type: none">JasperReportsJasper BI

EMSA reporting platform is based on JASPER BI Enterprise Editition; details on this platform can be found in Annex 4, “EMSA_JASPER_Technical_Document”.

3.6 Geographic Information System AND OGC (Open Geospatial Consortium) standards

EMSA Maritime Applications geospatial services are fully based on the OGC (Open Geospatial Consortium) standards, which have become key standards in use at EMSA. Some practical usage, but not limited, of these standards are:

- Electronic Nautical Charts:

EMSA is currently using an Electronic Nautical Charts distribution system for usage on the EMSA Maritime Applications. This system is providing ENCs, using a standard WMS interface, that are used as the base layer on the EMSA Maritime Applications.

- OGC standards used for vessel detection and correlation:

EMSA is using OGC standards to provide Vessel Detection and Correlation services to other EU agencies. The standards being used are WMS and WFS. It is also intended to use WPS to generate VDS (Vessel Detection System) correlations.

- Creation of traffic density maps:

EMSA is now using OGC standards (mainly WMS) to provide traffic density maps to end-users. EMSA will develop further this functionality to include more detailed TDMs with a higher definition than the current maps, on smaller areas), comparative maps (which show the differences between two maps) and vector maps (which show individual ship routes in polylines).

The GIS technologies in use at EMSA are:

Table 29 GIS Platform

GIS Platform
<ul style="list-style-type: none">ESRI Arc GISJeppesen C-Map Professional +GeoServerLuciad

3.7 Logging

Log4J shall be the preferred library for generating application logs. All application logs should use the same log message format, as described below:

```
<param name="ConversionPattern" value="%d{yyyy-MM-dd/HH:mm:ss.SSS/zzz} %-5p [%-t] [%l] %x - %m%n" />
```

Mandatory fields and format:

- %d – date in the specified format.
- %-5p - Priority of the logging event.
- %m - application supplied message associated with the logging event.
- %-t - name of the thread that generated the logging event.
- %l - location information of the caller which generated the logging event.
- %x - NDC (nested diagnostic context) associated with the thread that generated the logging event.

The following conversion patterns should be avoided as much as possible for Production environments, due to increased processing needs:

- C
- F
- 1, L
- M

The logging level should be changeable without requiring a restart of either the application or the application server. As for all configuration files, the log configuration file must reside outside of the packaged application.

Definition and implementation of log rotation and clean-up rules/processes is mandatory for every single logfile generated by the systems and its components.

EMSA makes use of Splunk for logging centralization and visualization. Applications must make sure that the logging patterns used are compatible with Splunk.

3.8 Storing Times and Dates

All EMSA servers, regardless of their function, shall use NTP to maintain accurate and aligned system clocks.

In order to prevent mismatches between data stored in different applications, all data shall in all cases be stored in Coordinated Universal Time (UTC). It is important to note that UTC, as opposed to local time, does not change with a change of seasons.

When a time is displayed to a user, used for triggering workflows or generating reports, it shall be the responsibility of the application to convert, if so desired, the stored UTC time to local time for the user. The final decision on if, or how the conversion shall happen, depends on the business requirements and will be an application decision. It is recommended for the user to be informed whether UTC time, user local time or source local time is displayed.

3.9 Others

The following points are generic mandatory requirements that shall be respected:

- Root or rooted administration accounts shall not be used.
- All system components shall be used by the same OS user.
- Software distribution cannot be done using rpm or any other solution that requires root privileges.
- In case it is necessary to have authentication on middleware components (e.g. application server, JMS) a dedicated user must be used. This user cannot be administration user of the components.
- When using non-compiled languages (e.g. php, perl) the versions of these languages shall be aligned with the version distributed bundled in OS version.
- Configuration files shall not include passwords in clear text. Solution to cope with this requirement may vary and must be agreed with EMSA.

If any deviation is foreseen, it shall be detailed and justified. EMSA has the last word in the decision process.

4. Service Oriented Architecture

EMSA applications should be compliant with the Enterprise Service Oriented Architecture with the objective of providing business and data services to others applications and being flexible and agile in order to easily adapt to change in short time.

EMSA Service Oriented Architecture is supported by a state of the art Service Oriented Infrastructure that follows the architectural best practices of the SOA metamodel.

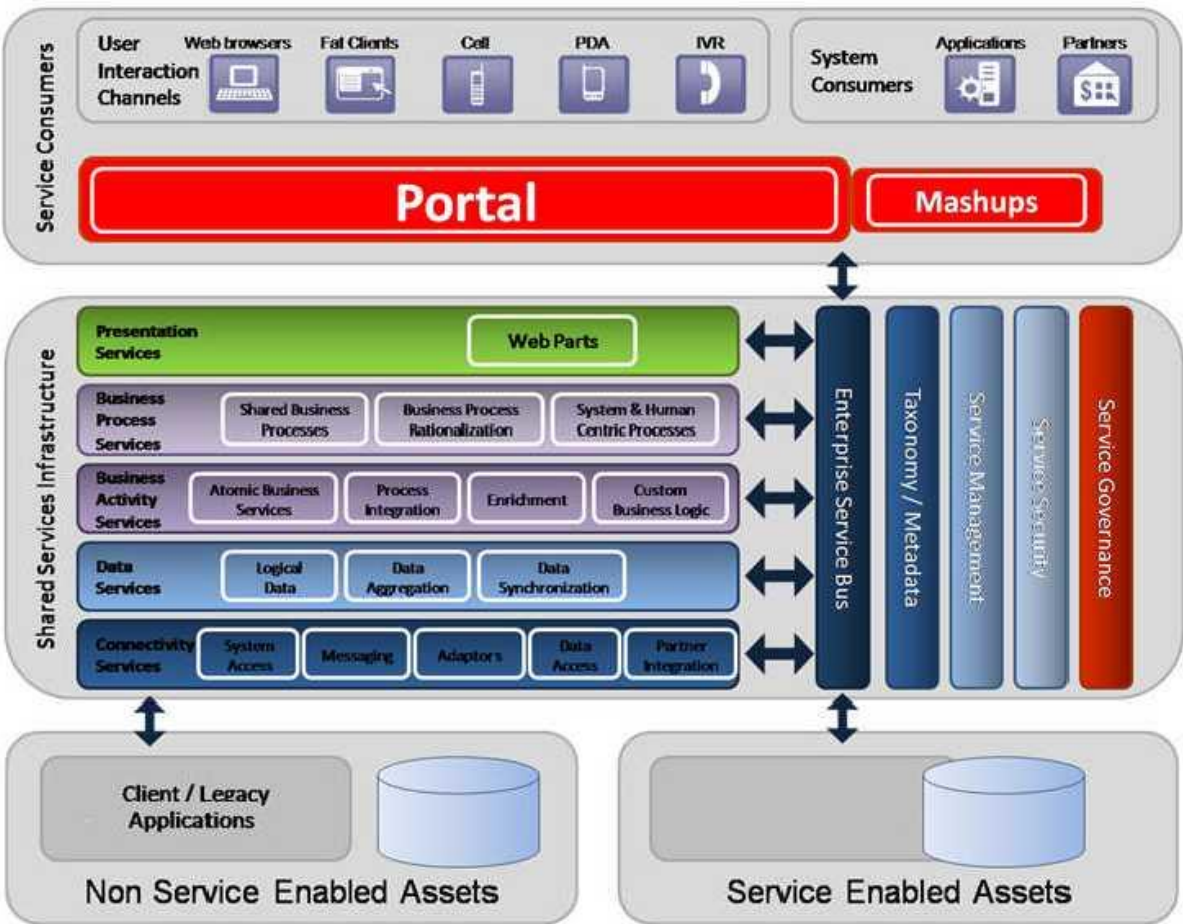


Figure 8 SOA architecture

The two major components supporting EMSA Service Oriented Architecture are:

Table 30 EMSA SOA key components

EMSA SOA key components	
■	Liferay Portal, version 7.1 Enterprise Edition
■	Oracle SOA Suite 12c

The fundamental building block of Service Oriented Architecture is a service. A service is a component that can be interacted with through well-defined interfaces or message exchanges. Services must be designed to perform

simple, granular functions with limited knowledge of how messages are passed to or retrieved from and for flexibility, agility, availability and stability.

EMSA principles of service orientation, which must be followed while designing services, are:

1. Services are loosely coupled components
2. Services are independent components
3. Services are self-contained
4. Services boundaries are explicit
5. Services are autonomous
6. Services share schema and contract
7. Services are independent deployable (logical aggregation can be considered)

Services designed based on these principles are much more likely to be reused within EMSA growing SOA infrastructure.

Please refer also to Annex 5 “EMSA SOA Guidelines & Rules”.

4.1 Service Consumers

Service consumers or composite applications are the applications that are developed to handle business actions or events initiated by business initiators. Business event initiators are entities that initiate business actions or events (either human users or other systems).

4.2 Shared Service Infrastructure

Shared service infrastructure defines the framework to shared services. It is based on Validate, Enrich, Transform, Route, and Operate or invokes (VETRO) patterns

Shared services are shared and reusable services that are used in service orchestration while creating business processes. Examples of shared services types are:

- Presentation services that present the data to the user.
- Business services that represent core business capabilities. Business services can range from relatively simple to very complex cross-functional, inter-enterprise business process.
- Data services that are entity services which provide access to enterprise data. Simple data services have a Validate, Create, Retrieve, Update, and Delete (CRUD) interface but more complex data services could be responsible for data aggregation or data synchronization.

5. Software Versioning Scheme

All applications being developed for or by EMSA shall use the following versioning scheme:

- [major].[minor].[revision]<.internal number>

Follows a description of the fields:

- Major will start 0 and will be increased by 1 every time significant new functionality is added to the application, or when significant changes to the implementation and/or organisation of the code have happened, such as:
 - When delivery of a new application or a major new version has been accepted, the major number will be increased by 1, other version numbers will be reset to 0;
 - Development of the next major version starts by increasing major version number by 1 and resetting all other version numbers to 0;
 - The above rules mean that all even numbered versions (+0) will be development releases for major new versions, whilst all odd numbered versions will be stable, production releases. E.g. if a software with version number 0.2.65 has been accepted for use in production environment, its version number will be 1.0.0. Development for the next major release will start at 2.0.0 and the production accepted release of this will carry a 3.0.0 version;
- Minor will be increased by 1 whenever less important new functionality or user interface changes are introduced;
- Revision will be increased by 1 whenever a new application version containing only bug fixes is delivered for deployment in EMSA pre-production environment;
- The internal number is an optional element that may be used by the contractor.

6. Summary

Minimum SW versions

Table 31 Minimum software versions

SW or Technology	SW Version	Comment
Oracle WebLogic	12.2.1	Active / Active Weblogic clustering is foreseen for critical applications
Wildfly	10.1	
Tomcat	9.0	
Oracle IdM Suite	ORACLE IAM Suite 11gR3 IAM 11.1.2.3.0 SOA 11.1.1.9.0	
Oracle Access Manager	OAM suite 10gR3 OAM 10.1.4.3.0	
Oracle SOA Suite	12.2.1	
Oracle OSB	12.2.1	
openLDAP	2.4	
Jasper BI Jasper Reports	7.1	
Liferay	DXP 7.1	
ORACLE EXADATA database	12.1.0.2	
ORACLE standalone database	12.1.0.2	TEST environment only
PostgresQL	12	
ESRI ARCGIS	10	
Geoserver	2.14	
LuciadLightspeed	2016.1.53	
LuciadFusion	2016.1.53	
LuciadRia	2017.1	

Please note that, based on EMSA's official patching policies, the mentioned versions can be changed in specific cycles and without notice. Therefore, the above versions shall be considered as the minimum versions and never as "the only version"

Some additional information, can also be found below:

Table 32 Software additional information

Area	Description	Technology	SW Version	Comment
Backup	SW	VMware VM backup; Legato Networker	7.6 SP3	
	HW	HP MSL8096 and Dell PVT Tape Libraries	N/A	

Business Continuity	HW/SW systems to guarantee different degrees of service availability	Local scale: VMware HA and FailOver Geographical scale: Asynchronous data replication through the Storage Array; VMWare Site Recovery Manager;	ESXi V 5	
Clustering	Service fail-over	Front-end: Weblogic Active/Active Back-end: Oracle EXADATA	12c 12c	
Data Links	Internet connectivity	2 Internet circuits Internet IP connections	N/A	Each link: 100 Mbps, 256 Provided independent IP addresses
GIS		ESRI ArcGIS Geoserver,	10 2.14	
HW Servers	VM hardware	VMware Hardware revision 8 (vSphere 5)		Only production database is not virtualised and runs on blades as well.
	VM Host hardware	HP Blade and DL series servers	N/A	
Monitoring System		Nagios	N/A	
Network Security	Security DMZ	Checkpoint blades	R75.40	2 node clustered configuration with Mobile Access VPN
Operating Systems		Linux and MS Windows	RedHat Enterprise Linux 7 Windows Server 2008	
Proxy	Security DMZ	F5 Big IP v5000 series proxies	11.4.0	Clustered configuration with 2 nodes
SAN Storage	Storage Area Network	Brocade Fabric; EMC Clariion Model CX4-240; Netapp FAS3240		
Virtualisation		VMWare	vSphere 5	
Electronica Nautical Charts		Jeppesen C-Map Professional +	V360	For redundancy purposes: 2 nodes load-balanced in the F5

European Maritime Safety Agency

Praça Europa 4
1249-206 Lisbon, Portugal
Tel +351 21 1209 200
Fax +351 21 1209 210
emsa.europa.eu

