Prepared for EMSA

Technical Design Specification

ENCWS

Reference:	ENC_WS
Date:	07-03-2012
Version:	3.0.0

WIDE SCOPE™

ENCWS Technical Design Specification

Wide Scope

Authors

Name	Area	Email
Ricardo Almeida	Business Dev	ricardo.almeida@widescope.pt

Referenced Documentation

Reference	Title	Author	Date

Reviews

Version	Date	Description	Author
1	16/01/2012	Initial Version	Ricardo Almeida
2	13/02/2012	Document reviewed	Ricardo Almeida
3	07/02/2012	Document review, WFS removed	Ricardo Almeida

Table of Contents

1 Technical overview 5

2 Conceptual System Architecture 6

2.1 Web Browser for Administration Purposes 7

2.2 Presentation 7

2.3 Core business functional components 7

2.4 GeoServer Catalog Data 8

2.5 Chart Integration Layer 8

2.6 Chart Database 9

2.7 LDAP 18

3 Software design and layering 19

3.1 Conceptual software architecture 19

3.2 Layers in the software architecture 20

3.2.1 Presentation 21

3.2.1.1 Apache Wicket 21

3.2.1.2 Spring Framework 22

3.2.1.3 Servlets 24

3.2.2 Services 24

3.2.2.1 WMS 24

3.2.2.1.1 OGC Standard 25

3.2.2.1.1.1 OGC GetCapabilities Operation 25

3.2.2.1.1.2 OGC GetMap Operation 26

3.2.2.1.1.3 OGC GetFeatureInfo Operation 29

3.2.2.1.2 Inspire Standards 30

3.2.2.1.2.1 Inspire View service operations 31

3.2.2.1.2.2 Inspire GetCapabilities Operation 33

3.2.2.1.2.3 Inspire GetMap Operation 33

3.2.3 Integration Layer 36

3.2.3.1 Rest Web Services 36

3.3 Transversal system features 37

3.3.1 Logging 37
3.3.1.1 Apache Log4j 37
3.3.1.2 Apache Log4net 37
3.3.2 Security and Authentication 37

4 Modules 39

4.1 Administration Module 39

4.2 The CMAP Web Service 40
4.2.1 Projections 40

4.3 Backend Layer Components 41

5 Object models and interactions between objects 42

5.1 Sequence diagram 43

6 Physical system architecture 46

6.1 Conceptual hardware architecture 46

6.2 Physical hardware architecture 48
6.2.1 Production environment 48
6.2.2 Pre-production environment 50
6.2.3 Test/Development environment 51

7 Annex I – Software tools description 53

7.1 Jeppesen SDK 53

7.2 GeoServer 53

7.3 Tomcat 54

7.4 Microsoft Internet Information Services (IIS) 55
7.4.1 IIS 7 Components 55
7.4.2 Protocol Listeners 55
7.4.3 World Wide Web Publishing Service (WWW service) 56
7.4.4 Windows Process Activation Service (WAS) 56

1 Technical overview

The ENCWS objective is to implement an ENC Web Services, following OWASP Security Level 2a , system that provides capabilities to distribute nautical charts dataset, using the EMSA Maritime Applications through Web Services. This system will use as source for the Electronic Nautical Charts the Jeppesen C-Map 'Professional +' Global Chart database CM 93/3.

ENCWS is a Geospatial Solution

The distribution of the Nautical Charts must follow the OGC (Open GIS Consortium) standard WMS (Web Map Service) from version 1.0.0 to 1.3.0 and must be in line with the INSPIRE directive.

The ENCWS technical architecture proposed is therefore based on a **Geospatial Solution** featuring Jeppesen SDK 4.5, Rest Web Services (.Net Framework 3.5) and GeoServer technologies (Oracle Java SE 6).These three tools implement a robust and strongly WMS Architecture that enables **ENCWS** integration with external systems.

Jeppesen C-Map will be connected to GeoServer thought a GeoServer Data Store, allowing the push of related layer information from C-Map. The C-Map Layer information is stored in GeoServer and is manly used to build WMS Services Metadata. GeoServer, using an OWS service will map the WMS request parameters into C-Map SDK logic.

OWS is the different types of services in GeoServer include WFS, WMS, and WCS, commonly referred to as "OWS" services. These services are global in that each service publishes ever layer configured on the server.

The C-Map Layers are managed inside GeoServer using GeoServer UI Extension.

2 Conceptual System Architecture

The system architecture can be depicted in the following high-level concept:

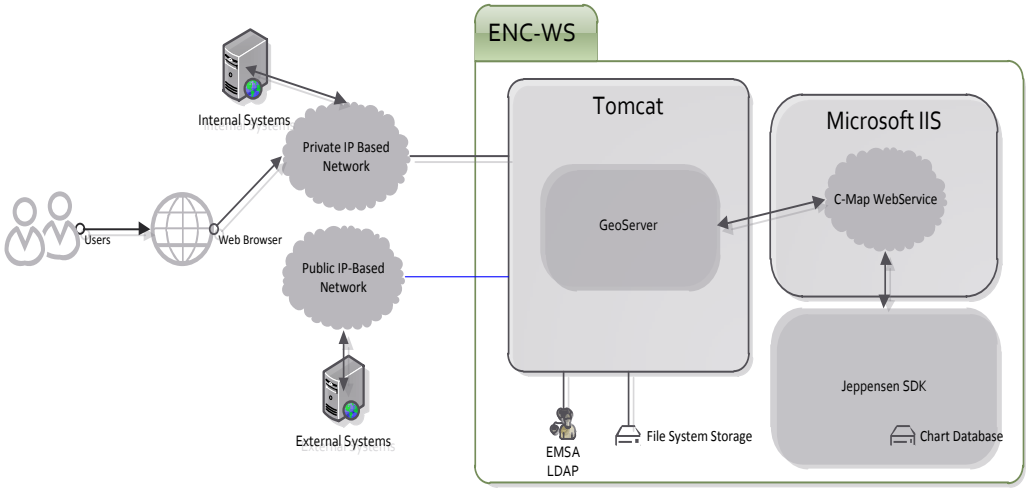


Figure 1 – Conceptual System architecture

The system is deployed on a **non-clustered Tomcat Application Server**, hosting the central component of ENCWS: GeoServer

The WMS uses the integration layer of **C-Map WebServices deployed** on Microsoft IIS for the chart operational data.

The GeoServer is protected under one common security mechanism whose authentication and authorisation is **managed in an LDAP server**.

The following topics describe superficially the core system components:

2.1 Web Browser for Administration Purposes

Administrator users access ENCWS using a web-browser which provides an intuitive **Web Interface.** It is composed of HTML, CSS and images in a Web 2.0 environment.

2.2 Presentation

GeoServer Portal is the component that interfaces with human end-users.

The portal provides users with the access to business functionalities wrapped in servlets under an intuitive and friendly Graphical User Interface. This human interface is built upon a **Web 2.0** oriented environment which improves user experience by making it feel like interacting with the web-based interface as if it would be interacting with a standalone application.
GeoServer requires **Oracle Java SE 6** or newer is strongly recommended, although Oracle JRE 5 will work. A Java Development Kit (JDK) is not required to run GeoServer.

2.3 Core business functional components

Tomcat Application Server is the core system of the application architecture. It is the application that support and serves GeoServer WMS Services to end-users and external systems.

Such components are resources under the Application Server perspective that are subject to access restrictions, performance monitoring, management and deployment.

2.4 GeoServer Catalog Data

The configuration of the system should be recorded in order to be reloaded by the system at any time and in possibly different installations. This configuration cannot be dependent on the specific details of the underlying infrastructure

The GeoServer data directory is the location on the file system where GeoServer stores its entire configuration.

The way to copy a data directory is to copy one that exists in an existing GeoServer instance installation.

2.5 Chart Integration Layer

Microsoft Information Server is the core system of the integration layer. It is the layer that supports and serves ENCWS Web Services to GeoServer.

Rest Web Services will be developed **using .Net Framework 3.5** and composes the layer of integration in ENCWS. It enables the interaction with the ENCWS Chart Database

2.6 Chart Database

Chart database is provided C-MAP Professional+ in vector format.

This product is distributed in the Jeppesen's DNV type approved SENC format, CM-93/3. SENC distribution makes the chart data available and secure in a data format ready to be used on the ECDIS/ECS, and is supported by the majority of system brands on the marked.

Maps presentation will be displayed using C-Map or S-52 according with the IHO S-52 standard.

The next table presents the possible parameters for C-Map and S-52.

Parameter description	Default	Input
Sets the value of the shallow contour, m.	2 meters	S52, C–Map
Sets the value of the safety contour, m.	30 meters	S52, C–Map
Sets the value of the deep contour, m.	30 meters	S52, C–Map
The two shades scheme is used for the depth areas presentation if set to '1'.	0	S52
If set to '1', shallow pattern is used to present shallow areas more prominently.	0	S52
If set to '1', a simplified POINTS presentation is used.	0	S52
If set to '1', Areas with plain boundaries presentation is used.	0	S52
Sets the value of the safety depth, all soundings with the depth value more than safety depth are shown in gray color, m.	30 meters	S52, C–Map

If set to '1', important for navigation text is displayed (e.g. bridge clearance).	0	S52, C–Map
If set to '1', other text (e. g. objects' names and information) is shown	0	S52, C–Map
IMO Display Category: 0 — DisplayBase 1 — Standard 2 — Other Mandatory parameter.	Should be set to the display category used in the application at a given moment	S52, C–Map
If set to '1', a string with the short light description is displayed.	0	S52, C–Map
If set to '1', the real lengths of lights sectors are shown	0	S52, C–Map
Sets the local time to process periodic attributes (DATSTA, DATEND), days (OLE time format).	0	S52, C–Map
If set to '1', obstructions (OBSTRN) or isolated underwater dangers (UWTROC, WRECKS) with depths less than the safety contour and which lie within the safety water are presented by the Isolated Danger symbol as hazardous objects. The category is set to the IMO category DISPLAYBASE.	0	S52, C–Map
If set to '1', correspondent M_COVR objects from overscaled dataset are presented according to the OVERSC01 pattern. See the correspondent value for the	0	S52
If set to '1', no DEPCNT objects are presented as safety contours.	0	S52, C–Map

If set to '1', all soundings with depths more than the GP_PARAM_ID_SAFETYDEPTH value are not shown. In C-Map presentation, VALSOU attributes for UWTROC, OBSTRN, WRECKS objects are not presented either.	0	S52, C–Map
If set to '1', no numeric labels of depth contours are shown.	0	S-52, C–Map
If set to '1', only the safety contour is shown.	0	S52
If set to '1', the object Quality of data is not shown.	0	S52
If set to '1', no $TEXTS objects are shown.	1	S52, C–Map
The Scale/ScalePres quotient : Scale — original scale of the dataset, ScalePres — scale of the presentation quotient. Should be more than one, the default value is two. In case the Scale/ScalePres is more than two, the correspondent pattern is drawn.	0	S52, C–Map
If set to '1', the coverage of datasets is not shown.	0	S52, C–Map
If set to '1', the soundings with depth values less then the value that is set in the GP_PARAM_ID_SHALLOWCONTOUR are not present. In the C-Map presentation, VALSOU attributes for the UWTROC, OBSTRN, and WRECKS objects is not presented either.	0	S52, C–Map
Permits to choose the light sectors to be highlighted. ALL_SECTORS 0x7	0	S52, C–Map

WHITE_SECTORS_ONLY 0x1 RED_SECTORS_ONLY 0x2 GREEN_SECTORS_ONLY 0x4 WHITE_RED_SECTORS 0x3 WHITE_GREEN_SECTORS 0x5 GREEN_RED_SECTORS 0x6		
The radius for the search range, NM (should be less than or equal to 40 NM). If the property is set to '0', the default radius value is 30 NM.	15 NM	S52, C–Map
Depth measurement units: 0 — Meters 1 — Feet as integer part and inches as fractional part 2 — Fathoms as integer part and feet as fractional part 3 — DecFathoms 4 — Rounded feet	0	S52, C–Map
Is valid if the GP_PARAM_ID_LIGHTS_LEG_LENT equals '1'. 0 — Light legs are shown according to the VALNMR attribute, (default); 1 - All Light legs are shown according to the GP_PARAM_ID_NOMINAL_RANGE;	0	S52, C–Map

2 - if the GP_PARAM_ID_NOMINAL_RANGE value is more than the VALNMR attribute value, Light legs are shown according to the GP_PARAM_ID_NOMINAL_RANGE, else Light legs are shown according to the VALNMR attribute value.		
If set to '1', the correspondent text is not shown for Mariner Objects.	0	S52, C–Map
If set to '1', symbols marking object with INFORM and/or NINFORM attributes are displayed.	0	S52
If set to '1', symbols marking objects with TXTDSC and/or NTXTDS attributes are displayed.	0	S52
If set to '1', symbols marking objects with PICREP attribute are shown.	0	S52
If set to '1', text of user remarks for Mariner Objects is not shown.	0	S-52, C–Map
If set to '1', all depth contours are shown in all modes	0	S-52, C–Map
If set to '1', the color for Safety contour is changed	0	S-52, C–Map
If set to '1', an index from S52Colors.h can be chosen.	obsolete, is not used in current presentation	S-52, C–Map
Sets the color priority as follows: 1 – WHITE lights have the highest priority 2 – RED lights have the highest priority 3 – GREEN lights have the highest priority	2: Red lights have the highest priority	S-52, C–Map

If set to '1', the radar conspicuous objects have special presentation. These objects are: bcnwtw, bridge, boywtw, cblohd, canbnk, FLODOC, PYLONS, rivbnc, SLCONS.	0	S-52 Inland waters
If set to '1', the objects with DATSTA ≥ GP_PARAM_ID_VOYAGE_START and DATEND ≤ GP_PARAM_ID_VOYAGE_END for route planning are shown. GP_PARAM_ID_VOYAGE_START and GP_PARAM_ID_VOYAGE_END are mandatory.	0	S-52, C–Map
If set to '1', unknown objects are present.	0	S52, C–Map
If set to '1', objects with unknown attributes are present.	0	S52, C–Map
Height units: 0 — meters 1 — feet	0	S52, C–Map
Range measuremen units: 0 — NM 1 — meters (or kilometers) 2 — yards (or kiloyards) 3 — statute miles All other values are prohibited	0	S52, C–Map
0 — only OBJNAM is present 1 — only NOBJNM is present	0	C–Map

2 — both OBJNAM and NOBJNM are present		
If set to '1', buoys and beacons (BOYXXX and BCNXXX) are presented with NOAA symbols.	0	C–Map
If set to '1', SOUNDG objects are not displayed.	0	S52, C–Map
If the parameter value is other than '0', the time used for Tides presentation is the UTC time, and as the time value the parameter value is used (the time is set in the OLE time format); If set to '0', the time used for Tides presentation is the system time.	0	C–Map
If the parameter is set to '1' and the IMO display category is OTHER, non–dangerous objects OBSTRN, UWTROC and WRECKS are not displayed.	0	S52, C–Map
If set to '1', the SCAMIN attribute is not processed.	0	S52, C–Map
If set to '1', lights whose COLOUR attribute is "White" are shown in the original color (white, rather than yellow).	0	C–Map
If set to '1', limits of the ENC coverage are displayed according to the conditional DATCVR02 procedure.	0	S52, C–Map (enabled from the 3.6.2.1 version of the SDK)
If set to '1', chart scale boundaries where the navigational purpose of data changes are displayed according to the conditional DATCVR02 procedure.	0	S52, C–Map (enabled from the 3.6.2.1 version of the SDK)

If set to '1', overscale pattern is displayed according to the DATCVR02 conditional procedure. GP_PARAM_ID_OVERSCALE_PATTEN is switched off automatically if this parameter is set to '1'.	0	S52, C–Map (enabled from the 3.6.2.1 version of the SDK)
If set to '1', all hazards (OBSTRN, WRECKS, UWTROC) are shown independently of the set Display page, Safety contour, etc. The danger symbol is not present.	0	C–Map
If set to '1', information about NATSUR, NATQUA and COLOUR attributes of the DBDARE object is shown.	0	S52, C–Map
If set to '1', isolated dangers (OBSTRN, WRECKS, UWTROC) in shallow water are shown at the IMO display categories: BASE, STANDARD, and OTHER for C–Map presentation; STANDARD and OTHER for S52 presentation.	1	S52, C–Map
If set to '1', DEPCNT objects are labeled in the INT 1 style.	0	S52, C–Map
If set to '1', labeling of LNDELV objects is off.	0	S52, C–Map
If set to '1', LNDELV objects are labeled in the INT 1 style.	1	S52, C–Map
Switches anticluttering on/off	0	C–Map, Tides, C–Mariner
If set to '0', the anti-cluttering engines are switched off when the current display scale is less than the original scale of the owner . Otherwise, the relation between the current display scale and the original scale of the owner are ignored. If the GP_PARAM_ID_ANTICLUTTERING_OFF parameter is set	0	C–Map, Tides, C–Mariner

to '1', the parameter is not considered.		
If set to '1', antialiasing is switched on for line graphic objects	0	S52, C–Map
If set to '1', C-Map, S52, and CSI presentations are switched to a special mode. In this case GMDSS and SAR objects are shown only on DEPARE objects and are not shown on LNDARE objects.	0	S52, C–Map, CSI
Used for switching ON/OFF anticluttering for hazard objects (OBSTRN, WRECKS and UWTROC). 0 – anticluttering is switched off 1 – anticluttering is used for "side texts" (e.g. 'Obstr') and values of soundings (VALSOU attribute) 2 – full anticluttering	1	C–Map
If set to '1', generalization for LIGHTS, CBLSUB, FOGSIG, OFSPLF objects is switched on. For lights sectors there is an additional condition: even if the parameter is set to '1', generalization is switched off if the ratio of Presentation Scale / Compilation Scale is less than 0.3.	1	C–Map
Width of the chart window in logical units. This parameter is mandatory for correct setting of the Window width for presentation of the ENC coverage limits. The default value is the value of the clipping rectangle set in presentation.		S-52, C–Map
Height of the chart window in logical units. This parameter is mandatory for correct setting of the Window height for presentation of the ENC coverage limits. The default value is the value of the clipping rectangle set in presentation.		S-52, C–Map

If set to '1', space image will be drawn over DEPARE.	1	S-52, C–Map
If set to '0', cluttering for SOUNDFG objects will be switched off.	1	C-Map
If set to '1', OBSTRN, WRECKS, and UWTROC objects will be presented at the Display Base and Standard modes, but only in case they have a valid VALSOU attribute.	0	C-Map, S-52

C-Map Symbolization is more close to standard INT1 Presentation Model.

2.7 LDAP

The **LDAP** server hosts and organizes the data for user **authentication.** This authentication is only used to guarantee the user access to GeoServer web site.

3 Software design and layering

3.1 Conceptual software architecture

The following figure illustrates the main concepts in Architecture proposed:

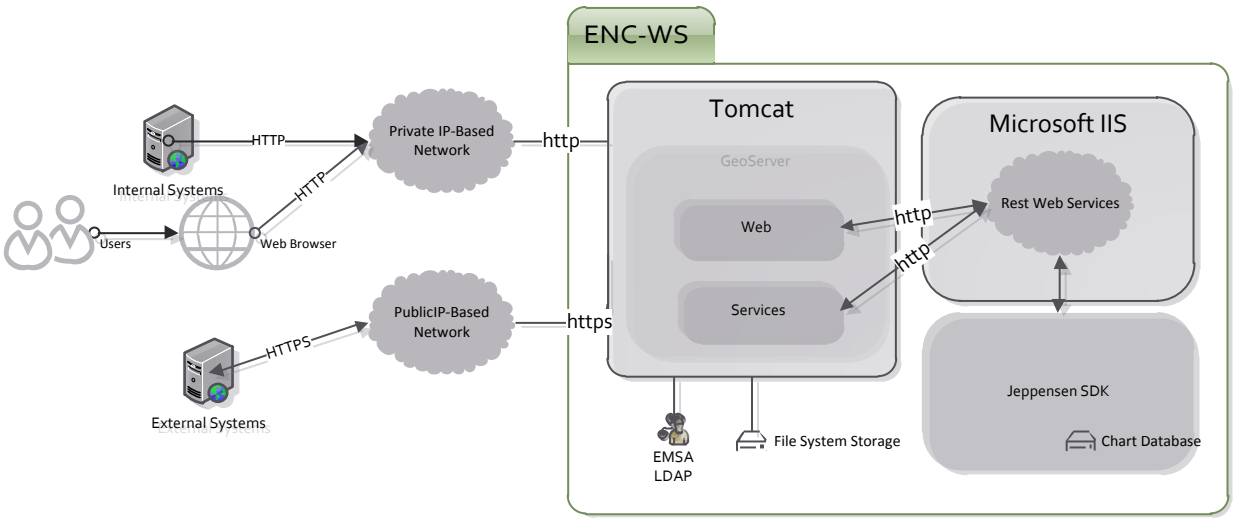


Figure 2 - Conceptual Software architecture

Complementing the figure in the previous chapter, this figure adds the conceptual design of the main software tools interaction and composition.

The picture exposes 3 main development components**: Web, Services and the Rest Web Services.** There will be two protocols accepted by GeoServer: HTTP for unsecured requests and HTTPS for secured communications. HTTPS is manly used for External Systems.

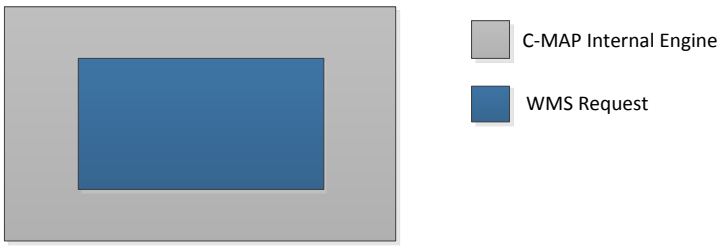
All requests to Rest Web Services use HTTP Protocol.

Rest Services implement the standard specification for OGC standard WMS.

Web Map Caching will use one of the primary potential benefits of REST that is HTTP caching. Using the WCF REST Starter Kit that provides a simpler model for controlling caching through we can apply different cache profiles to various GET operations. The Cache can be configured to vary

by a certain parameter of Get Operation within a time to expire.

There's also a mechanism used by C-Map Internal Engine that can improve also the map queries. This native mechanism of C-Map picks the bounding box for query a region and internally increases it for internal cache, allowing a better response by example on pan operations. Although the map query region being internally increased the response is always the original bounding box.



File System is used for storing GeoServer configurations. **Chart Database** is also stored on file system.

3.2 Layers in the software architecture

This section describes in detail the software layers (presented in the next figure) in the architecture with special relevance to the main 3 identified in the previous topic (Web, Service and Rest Web Services/C-Map Web Services) in a top-down approach.

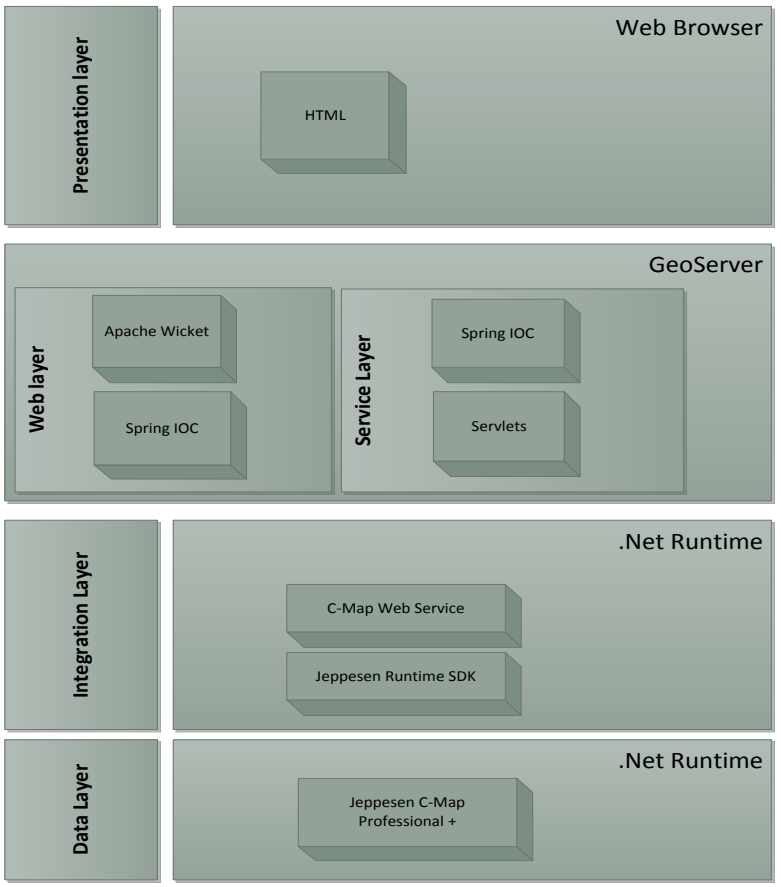


Figure 3 - Application layers

3.2.1 Presentation

GeoServer Portal is the component that interfaces with human end-users.

The portal provides users with the access to business functionalities wrapped in servlets under an intuitive and friendly Graphical User Interface. This human interface is built upon a **Web 2.0** oriented environment which improves user experience by making it feel like interacting with the web-based interface as if it would be interacting with a standalone application.

3.2.1.1 Apache Wicket

Wicket is a Java Web development framework. It's an open source, lightweight, component-based framework, which puts it in an altogether different league from some of the earlier ways of developing Web-based applications. Wicket strives for a clean separation between the roles of HTML page designer and Java developer by supporting plain HTML-based templates that can be built using any WYSIWYG HTML design tools. The templates can then be made dynamic with little modification.

Like other frameworks, Wicket is built on top of Sun Microsystems' servlet API. However, unlike frameworks based on the Model-View-Controller (MVC) model (such as Struts), Wicket takes away from you the task of handling request/response objects, which is inherent with technologies such as servlets.

With Wicket it’s possible to build reusable components that are statefull, instead of building controllers that handle request/response objects. Instead of creating a controller or action class, using a single page that holds components on it and define how each components reacts to user input.

3.2.1.2 Spring Framework

The Spring framework provides a common and useful design pattern which is the Inversion of Control (IoC).

Control flow is expressed in imperative programming in the form of a series of instructions or procedure calls. Instead of specifying a sequence of decisions and procedures to occur during the lifetime of a process, the user of an IoC framework writes the desired responses linked to particular events or data requests. External entities then take control over the precise calling order

and additional maintenance that are to be carried out to execute the process. In a sense, inversion of control follows what has been referred to as the "Hollywood Principle"—"don't call us, we will call you".

In practice, Inversion of Control is a style of software construction where reusable, generic code controls the execution of problem-specific code with the strong connotation that the reusable code and the problem-specific code are developed independently and the result often being a single integrated application.

The following picture displays the integration between MVC concepts in servlet development under the scope of the ENCWS architecture.

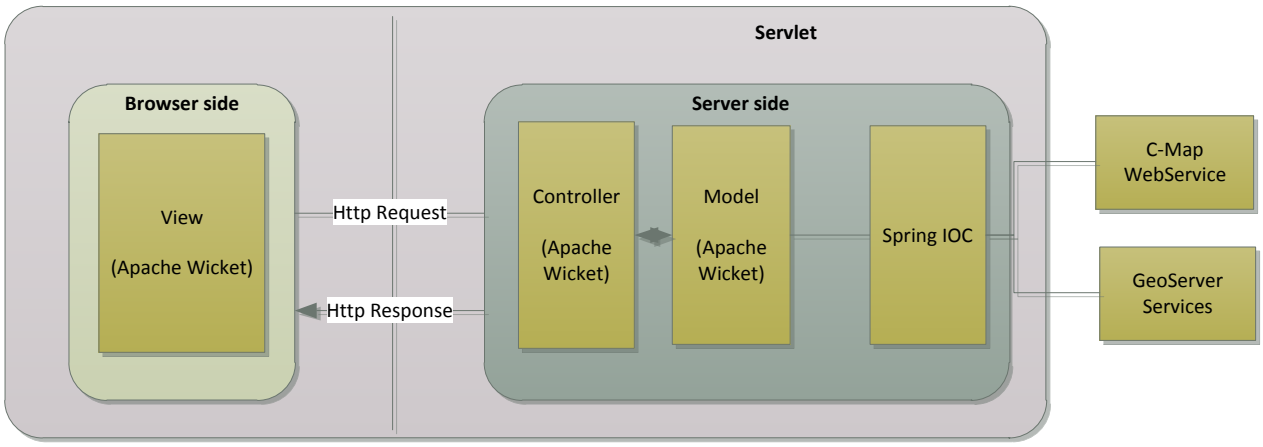


Figure 4 - Servlet architecture

- **Controller:** this component is an entry-point for all web requests. Spring already provides a rather complete implementation with features such as monitoring access to resources, roles, etc.

- **Model:** Spring Beans that are invoked by the controller based on requests received.

These beans are the integration point with the business logic, namely the **GeoServer Services and C-Map Web Services.**

View: The view is composed of CSS, JavaScript, images and HTML. It also includes a strong Web 2.0 environment enabled by Apache Wicket Framework (see previous topic).

3.2.1.3 Servlets

A servlet is a JavaTM technology-based Web component, managed by a container, that generates dynamic content. Like other Java technology-based components, servlets are platform-independent Java classes that are compiled to platform-neutral byte code that can be loaded dynamically into and run by a Java technology-enabled Web server. Containers, sometimes called servlet engines, are Web server extensions that provide servlet functionality. Servlets interact with Web clients via a request/response paradigm implemented by the servlet container.

3.2.2 Services

GeoServer is a server of geospatial data through the web. GeoServer implements standard open web protocols established by the Open Geospatial Consortium (OGC), a standards organization. GeoServer contains as well a high performance certified compliant Web Map Service (WMS). It is through this protocol that GeoServer can serve data and maps in an efficient and powerful way

3.2.2.1 WMS

The Web Map Service (WMS) is a standard protocol for serving georeferenced map images generated by a map server.

The WMS versions covered are from 1.0.0 to 1.3.0 .

WMS provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.

WMS Operations will allow the linking context of each layer to any tool that accepts a WMS standard call, e.g. this can be achieved by configuring a WMS Server on Arc-Map.

3.2.2.1.1 OGC Standard

The Open Geospatial Consortium, Inc (OGC) is an international industry consortium in a consensus process to develop publicly available interface specifications. OpenGIS® Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT.

3.2.2.1.1.1 OGC GetCapabilities Operation

GetCapabilities – this operation delivers service metadata, which is a machine readable (and human-readable) description of the server's information content and acceptable request parameter values bellow.

The next table presents the parameters for OGC.

Request Parameter	Mandatory / Optional	Description	Observations

VERSION	Optional	Request version	Supported
SERVICE=WMS	Mandatory	Service type	Supported
REQUEST=GetCapabilities	Mandatory	Request name	Supported
FORMAT=MIME_type	Optional	Output format of service metadata	Supported
UPDATESEQUENCE=string	Optional	Sequence number or string for cache control	Supported

3.2.2.1.1.2 OGC GetMap Operation

GetMap – this operation returns a map. Upon receiving a GetMap request, a WMS shall either satisfy the request or issue a service exception.

The projection requested (CRS parameter) shall be specified in each request but the Mercator projection is the default one.

Request Parameter	Mandatory / Optional	Description	Observations
VERSION	Mandatory	Request version	Supported
REQUEST	Mandatory	Request name	Supported
LAYERS	Mandatory	Comma-separated list of one or	Supported

		more map layers	
STYLES	Mandatory	Comma-separated list of one rendering style per requested layer	SDK doesn't follow SLD Standards. It will be created two dummy SLD for distinguish the S52 and C-Map Symbolization. This will be used on a GetMap Request
CRS	Mandatory	Coordinate reference system	Supported
BBOX	Mandatory	Bounding box corners (lower left, upper right) in CRS units	Supported
WIDTH	Mandatory	Width in pixels of map picture	Supported
HEIGHT	Mandatory	Height in pixels of map picture	Supported
FORMAT	Mandatory	Output format of map	Supported
TRANSPARENT	Optional	Background transparency of map (default=FALSE).	Supported, it Will be implemented using GDAL. The transparency will be applied to the full image.
BGCOLOR	Optional	Hexadecimal red-green-blue colour value for the background color	Only parameter White or Black are supported. Theses colors covers map

		(default=0xFFFFFF)	presentation Day | Night. The system will accept other parameters but it will assume white as default.
EXCEPTIONS	Optional	The format in which exceptions are to be reported by the WMS (default=XML).	Supported
TIME	Optional	Time value of layer desired.	Not Supported By CMAP, because is not part of IHO Standard and as such is not part of the CMAP 93/3 data. The system will accept this parameter but it will not take any effect on the request output.
ELEVATION	Optional	Elevation of layer desired	Not Supported By CMAP because is not part of IHO Standard and as such is not part of the CMAP 93/3 data. The system will accept this parameter but it will not take any effect on the request output.

3.2.2.1.1.3 OGC GetFeatureInfo Operation

GetFeatureInfo is an optional operation. It is only supported for those Layers for which the attribute queryable="1" (true) has been defined or inherited.

Request Parameter	Mandatory / Optional	Description	Observations
VERSION=1.3.0	Mandatory	Request version	Supported
REQUEST=GetFeatureInfo	Mandatory	Request Name	Supported
map request part	Mandatory	Partial copy of the Map request parameters that generated the map for which information is desired.	Supported
QUERY_LAYERS=layer_list	Mandatory	Comma-separated list of one or more layers to be queried.	Supported
INFO_FORMAT=output_format	Mandatory	Return format of feature information (MIME type).	Supported
FEATURE_COUNT=number	Optional	Number of features	Supported

		about which to return information (default=1).	
I=pixel_column M	Mandatory	i coordinate in pixels of feature in Map CS.	Supported
J=pixel_row	Mandatory	j coordinate in pixels of feature in Map CS.	Supported
EXCEPTIONS=exception_format	Optional	The format in which exceptions are to be reported by the WMS (default= XML).	Supported

3.2.2.1.2 Inspire Standards

The INSPIRE implementing directive 2007/2/EC aims to create a European Union (EU) spatial data infrastructure.

INSPIRE is based on a number of common principles:

- Data should be collected only once and kept where it can be maintained most effectively.
- It should be possible to combine seamless spatial information from different sources across Europe and share it with many users and applications.

- It should be possible for information collected at one level/scale to be shared with all levels/scales; detailed for thorough investigations, general for strategic purposes.

- Geographic information needed for good governance at all levels should be readily and transparently available.

- Easy to find what geographic information is available, how it can be used to meet a particular need, and under which conditions it can be acquired and used.

GeoServer has a community extension called Inspire. After using this extension it should be available at WMS Configuration Page.

There will be a block titled INSPIRE. This section will have three settings:

- Language combo box, for setting the Supported, Default, and Response languages
- ISO 19139 Service Metadata URL field, a URL containing the location of the metadata associated with the WMS
- Service Metadata Type combo box, for detailing whether the metadata came from a CSW (Catalog Service) or a standalone metadata file

3.2.2.1.2.1 Inspire View service operations

INSPIRE metadata are mapped to the <WMS_Capabilities> element of the GetCapabilities Response

Next table presents the mapping between INSPIRE metadata elements and [ISO 19128] WMS elements:

INSPIRE Metadata elements (Mandatory - Conditional)	ISO 19128 elements of <WMS_Capabilities>
Resource Title (M)	wms:Title

Resource Abstract (M)	wms:Abstract
Resource Type (M)	inspire_common:ResourceType (ExtendedCapabilities)
Resource Locator (C)	inspire_common:ResourceLocator (ExtendedCapabilities)
Coupled Resource (C)	wms:MetadataURL (Layer property)
Spatial Data Service Type (M)	inspire_common:SpatialDataServiceType (ExtendedCapabilities)
Keyword (M)	wms:Keyword; inspire_common:Keyword
Geographic Bounding Box (M)	wms:EX_GeographicBoundingBox (Layer property)
Temporal Reference (M)	inspire_common:TemporalReference (ExtendedCapabilities)
Spatial Resolution (C)	wms:Abstract
Conformity (M)	inspire_common:Conformity (ExtendedCapabilities)
Conditions for Access and Use (M)	wms:Fees
Limitations on Public Access (M)	wms:AccessConstraints
Responsible Organisation (M)	wms:ContactInformation
Metadata Point of Contact (M)	inspire_common:MetadataPointOfContact (ExtendedCapabilities)
Metadata Date (M)	inspire_common:MetadataDate (ExtendedCapabilities)
Metadata Language (M)	inspire_common:SupportedLanguages (ExtendedCapabilities)

3.2.2.1.2.2 Inspire GetView Service Metadata

Request parameter	Mandatory/ optional	Description	Observations
VERSION=version	Optional	Request version: 1.3.0	Supported
SERVICE=WMS	Mandatory	Service type. Fixed value: WMS.	Supported
REQUEST=GetCapabilities	Mandatory	Request name. Fixed value: GetCapabilities	Supported
LANGUAGE=code	Optional	Request language (INSPIRE extension).	Supported
FORMAT=MIME_type	Optional	Output format of service metadata. Defaults to text/xml.	Supported

3.2.2.1.2.3 Inspire GetMap Operation

Request parameter	Mandatory/ optional	Description	**Observations**
VERSION=1.3.0	Mandatory	Request version	Supported
REQUEST=GetMap	Mandatory	Request name	Supported
LAYERS=name,name	Mandatory	Comma-separated list of one or more map layers names. Names are	Supported

		harmonized INSPIRE layers names.	
STYLES=name,name	Mandatory	Comma-separated list of one rendering style per layer requested. When the STYLES parameter is left blank in the GetMap request, the INSPIRE default styling applies in the GetMap response to all layers (inspire_common:DEFAULT)	SDK doesn't follow SLD Standards. It will be created two dummy SLD for distinguish the S52 and C-Map Symbolization. This will be used on a GetMap Request
CRS=namespace:identifier	Mandatory	Coordinate reference system	Supported
BBOX=minx,miny,maxx,maxy	Mandatory	Bounding box corners (lower left, upper right) in CRS units and in the axis order of the CRS	Supported
WIDTH=output_width	Mandatory	Width in pixels of map picture	Supported
HEIGHT=output_height	Mandatory	Height in pixels of map picture	Supported
FORMAT=output_format	Mandatory	Output format of map.	Supported
TRANSPARENT=TRUE|FALSE	Optional	Background transparency of map (default=FALSE)	Will be supported with DGAL
LANGUAGE	Optional	Language	Supported

BGCOLOR=color_value	Optional	Hexadecimal red-green-blue colour value for the background color	Only parameter White or Black are supported. Theses colors covers map presentation Day | Night. The system will accept other parameters but it will assume white as default.
EXCEPTIONS=error_format	Optional	The format in which exceptions are to be reported by the WMS (default=XML)	Supported
TIME=time	Constraint	Time value of layer desired	Not Supported By CMAP because is not part of IHO Standard and as such is not part of the CMAP 93/3 data. The system will accept this parameter but it will not take any effect on the request output.
ELEVATION=elevation	Constraint	Elevation of layer desired	Not Supported By CMAP because is not part of IHO Standard and as such is not part of the CMAP 93/3 data. The system will accept this parameter but it

	will not take any effect on the request output.

3.2.3 Integration Layer

3.2.3.1 Rest Web Services

RESTful web services are designed to expose APIs on the web. REST stands for Representational State Transfer. It aims to provide better performance, scalability, and flexibility than traditional web services, by allowing clients to access data and resources using predictable URLs. Many well-known public web services expose RESTful APIs.

In the REST model, the server exposes APIs through specific URIs (typically URLs), and clients access those URIs to query or modify data. REST uses a stateless communication protocol. Typically, this is HTTP.

The following is a summary of RESTful design principles:

- A URL is tied to a resource. Clients access the resource using the URL.
- Create, Read, Update, and Delete (CRUD) operations are accessed via PUT, GET, POST, and DELETE requests in the HTTP protocol.
    - PUT creates a new resource.
    - DELETE deletes a resource.
    - GET retrieves the current state of a resource.
    - POST updates a resource's state.
- Resources are decoupled from their representation, so that clients can request the data in a variety of different formats.
- Stateful interactions require explicit state transfer, in the form of URL rewriting, cookies,

and hidden form fields. State can also be embedded in response messages.

3.3 Transversal system features

3.3.1 Logging

All actions from any class in the application, or any class in the open-source frameworks mentioned provide logging to plain-text files. The Log level can be changed thought property files using Log4j Standard Configurations. The log can be changed thought property files, also in runtime.

3.3.1.1 Apache Log4j

The framework **Log4j** (http://logging.apache.org) is a “de facto” standard in this chapter. Notice that other logging mechanisms based on saving errors and warnings in DB are also proposed, based on concepts analogous to those described in the “Auditing” section.

3.3.1.2 Apache Log4net

Apache log4net library is a tool to help the programmer output log statements to a variety of output targets. log4net is a port of the excellent Apache log4j™ framework to the Microsoft® .NET runtime.

3.3.2 Security and Authentication

According to the proposed ENCWS architecture there are only two entry points: the portal (GeoServer Portal) for end users and the services layer (GeoServer Rest Services) for external systems.

The GeoServer should provide services through 2 separate interfaces (ports).

- Internal port - in HTTP and without user/password authentication.
- External port - In HTTP but supporting the HTTPS offload to the F5 reverse proxy. Will be protected by login and password.

GeoServer Portal runs on the Tomcat Application Server.

Security Constraints will be configured in web application xml (web.xml).

The user information is stored in a Directory Service. **The Lightweight Directory Access Protocol (LDAP)** is the proposed directory to house user ids, passwords, and security policies including rules and access control lists. The directory will follow the current EMSA LDAP schema (TBC).

It also maintains User Access Policy information and contains access control lists and associated policies.

Administration Users are also required to be authenticated and authorised in order to interact with ENCWS Administration Area.

4 Modules

4.1 Administration Module

The Administration module contains the pages that allow configuring new Layers.

This module will allow customizing layer definitions:

1. Presentation objects of each layer

   A Layer is mapped to an object on SDK. There will be a list of two pre-defined Styles: Standard and Base. On each layer can be configured the spatial objects, called features, Danger, Ship Routing System, Navigation Aids, Nautical Background, Administration Boundaries, Harbour, Cautionary Areas and Installation and Wrecks
2. Create new Presentation Styles

   The creation of new presentation styles will be made using SDK Clone function. This allows to clone an existing syle, by ie S52 Standard, into o a new one.

This module will allow configuring the presentation styles according to S52 and C-Map symbolization, by ie: shallow contour, deep contour. These configurations will be stored on XML file and will be used upon a GetMap request.

Each layer that is configured using this module will be available, thought Data Store, to be published using GeoServer. A DataStore is essentially a source of data for rendering of feature; in this specific system the DataStore will be configured to use the CMAP Web Services.

Entire module is developed following GeoServer UI Extensions. The core functionality of this module is built on top of Apache Wicket and Spring IOC and is responsible to create new Layer Definitions.

The Apache Wicket supports the following browsers:

1. Internet Explorer 8+
2. Firefox 3+

4.2 The CMAP Web Service

C-Map Web Services exposes the Jeppesen SDK functionality over the network. Once it is exposed on the network, other application can use the functionality of your program.
C-Map Web Service will use the Microsoft WAS. The Windows Process Activation Service (WAS) manages the activation and lifetime of the worker processes that contain applications that host Windows Communication Foundation (WCF) services. The WAS process model generalizes the IIS 6.0 process model for the HTTP server by removing the dependency on HTTP. This allows WCF services to use both HTTP and non-HTTP protocols, such as Net.TCP, in a hosting environment that supports message-based activation and offers the ability to host a large number of applications on a given machine

4.2.1 Projections

Currently Jeppesen SDK, has implemented two projections components: the MercatorWGS84 and CMapUTM components.

The Latitude Longitude and Polar Projection will be implemented following Jeppesen SDK. This can be accomplish by using the **IGeoProjection** interface, this initializes the projection component and perform conversions of geographical coordinates on the earth ellipsoid to corresponding plane coordinates on a virtual map and vice versa.
Text distortion is handled using internal SDK mechanism.
The UTM projection will have the possibility to choose the UTM zone. When this value is not

defined in the request, the UTM zone to be used will be specified in a default parameter to be defined as part of the Administration of the system.

4.3 Backend Layer Components

The interaction of components in the backend is displayed in the following picture:

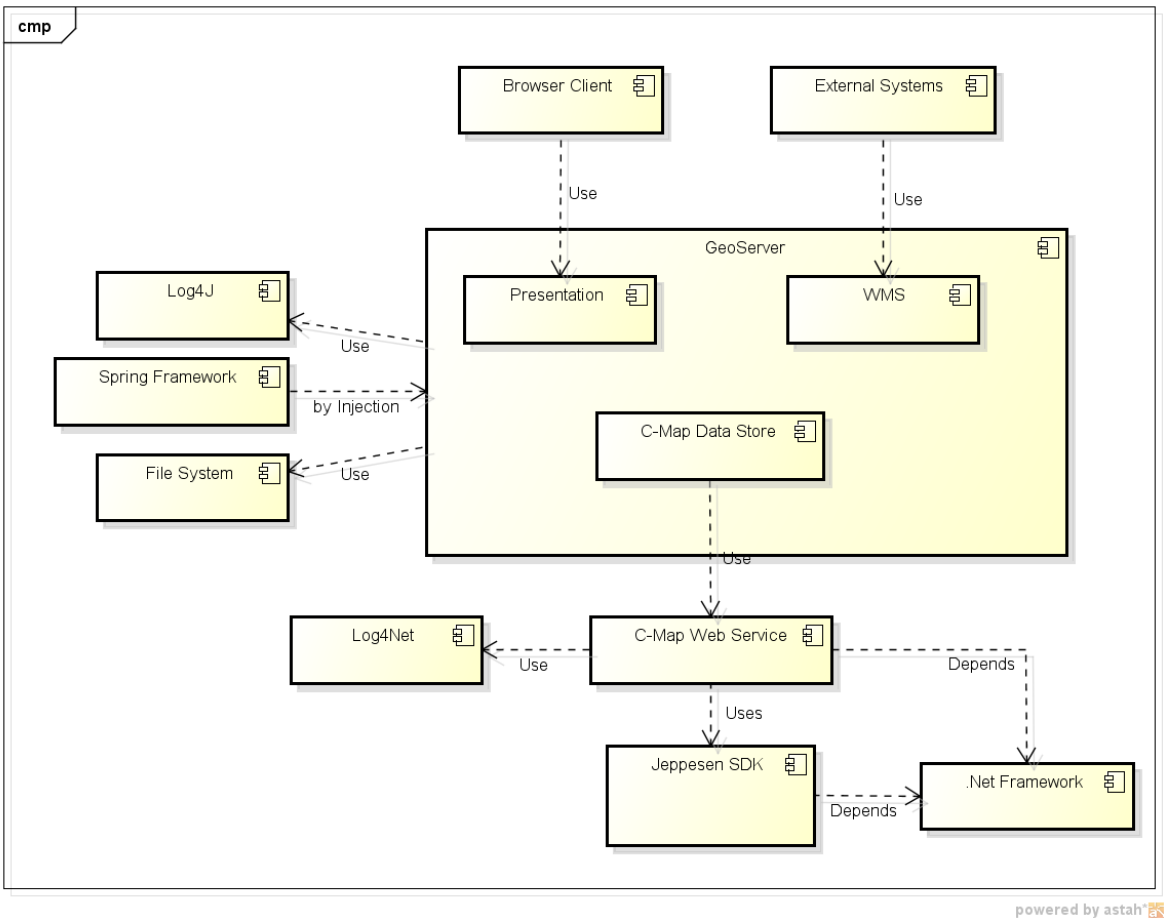


Figure 5 - Backend Components

5 Object models and interactions between objects

The object model maps the layers identified very closely. In fact, for each layer there is a corresponding package with objects and their interaction occurs as described for layers.

Consider GeoServer as Spring Framework + Apache Wicket + Servlet

The following picture illustrates the interaction between objects, frameworks.

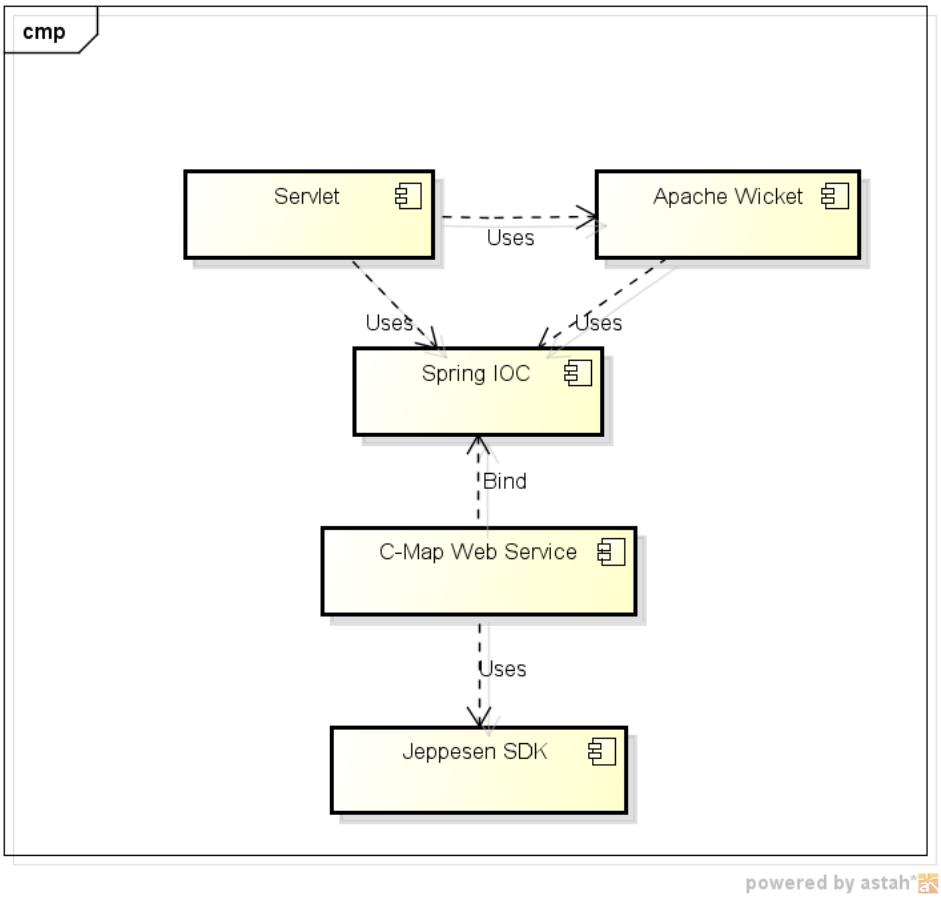


Figure 6 - Interactions between objects

The servlet object acquires through Apache Wicket a reference to a Web service through the injection of control provided by the Spring IoC sub-system.

The web service maps Jeppesen C-Map SDK implementation. This web service is a provider of some business logic, i.e. it wraps some Jeppesen SDK objects that encapsulate map producer logic.

In order to illustrate the flow of events going through the several system layers, we present a generic sequence diagram.

5.1 Sequence diagram

The following sequence diagram is a generic representation of the most complex use case, i.e. the one that traverses all application layers.

Let's consider the example request of an external system requesting a map. This request (which is a Servlet request) is going to request a map from the server and display it back to the system.

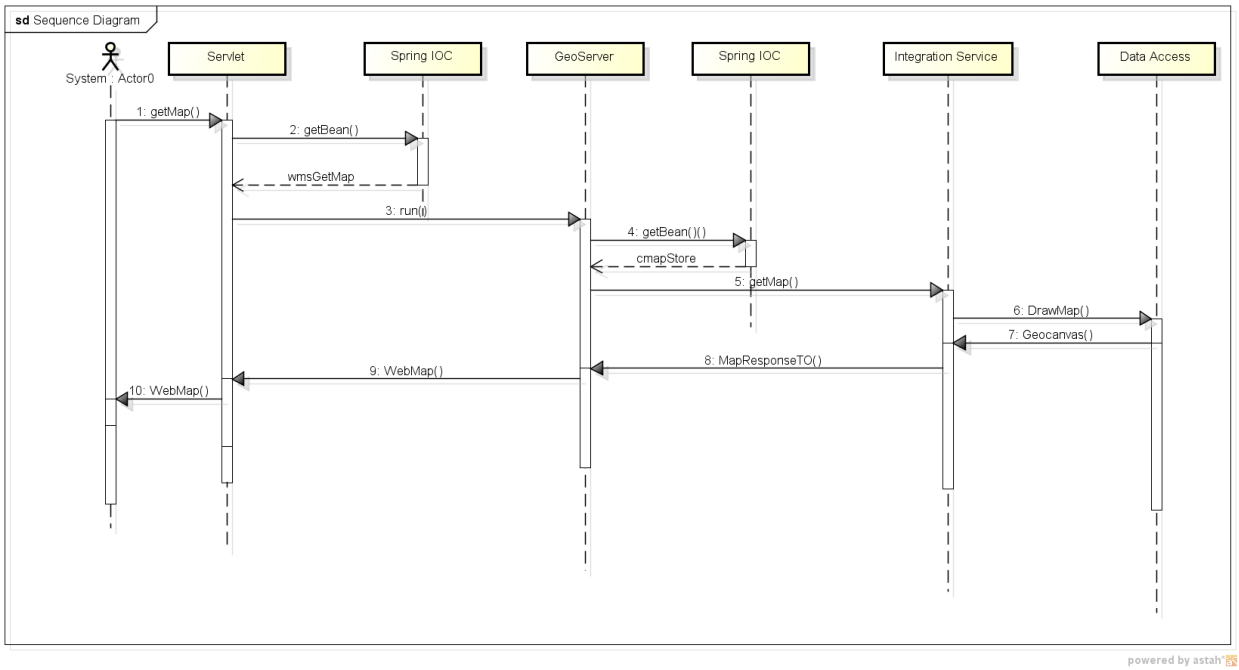


Figure 7 - Sequence diagram

The sequence diagram illustrates the events for such request:

1. getMap(): the external system calls an http url– its servlet – requesting to produce a map;
2. getBean(): the servlet needs to post the request to the WMS Bean. The control of the wms implementation to invoke is injected by a Spring IoC configuration.
3. wmsGetMap: Spring IoC injects the adequate WMS Implementation into the servlet in order for it to make it the appropriate request;
4. run(): with the WMS Implementation identified, the servlet sends an request request to produce a map;
5. getBean ():needs to post the request to the Store Bean. The control of the Data Store implementation to invoke is injected by a Spring IoC configuration.
6. cmapStore: Spring IoC injects the adequate Data Store Implementation into the servlet in order for it to make it the appropriate request;

7. getMap(): the Data Store needs to post the request to a web service;
8. drawMap: the Jeppesen SDK is invoked to produce the map ;
9. GeoCanvas(): the Jeppesen SDK returns a GeoCanvas Object containing the produced map
10. MapResponseTO(): the manager transforms the rich domain object entity into a DTO that is available to cross integration layers.
11. WebMap: the data store transforms de DTO into a WebMap that is available to cross all geoserver layers ;
12. WebMapp(): WebMap object ready for browser rendering;

6 Physical system architecture

6.1 Conceptual hardware architecture

Considering our understanding of EMSA's current hardware configuration the following picture illustrates the integration of the proposed ENCWS nodes in such scope.

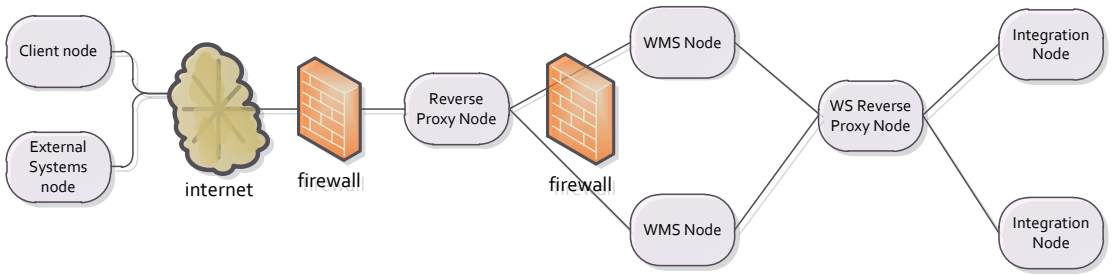


Figure 8 - ENCWS conceptual HW architecture

Client Node: The Client node provides the touch point for ENCWS users to interface with ENCWS services. This is typically a workstation running a Web browser.

External Systems Node: An External Systems node is a logical node hosting applications and services external to the enterprise.

The External System Node applications interface with the ENCWS services via Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS).

Reverse Proxy Node

The Reverse Proxy Node forwards all requests to the web server as well as the requests from the integration server. For the submitter of requests, the Reverse Proxy Node appears as the Web Server Node.

As request arrives in Web Node, the Web Service Request is made to Reverse Proxy that balances the load between each integration node.

The reverse proxy server provides security; access and data traffic redirection associated with the ENCWS applications.

WMS Node: The web node hosts the GeoServer itself.

GeoServer include:

- Connectivity and integration to allow access to ENCWS WMS data and services;

Integration Node: The integration node hosts the central repository of web services provided by the ENCWS information system. It is the placeholder for integration with C-Map Professional SDK.

Directory & Security Node: The LDAP for user authentication and authorization plays a central role in security. It is part of EMSA's current infrastructure and is accessible only for the web node.

6.2 Physical hardware architecture

6.2.1 Production environment

The conceptual hardware architecture maps into the physical architecture as the following picture illustrates.

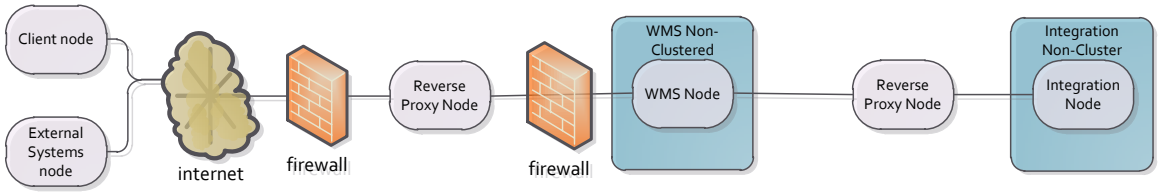


Figure 9 - ENCWS physical HW architecture

The squared blue boxes in the picture materialize the conceptual hardware nodes into physical resources. **Please notice that only conceptual nodes proposed in the scope of this document are materialized since all other nodes are part of EMSA's current infrastructure and therefore accepted as such**.

In detail, the 2 hardware clusters proposed can be illustrated in the following picture:

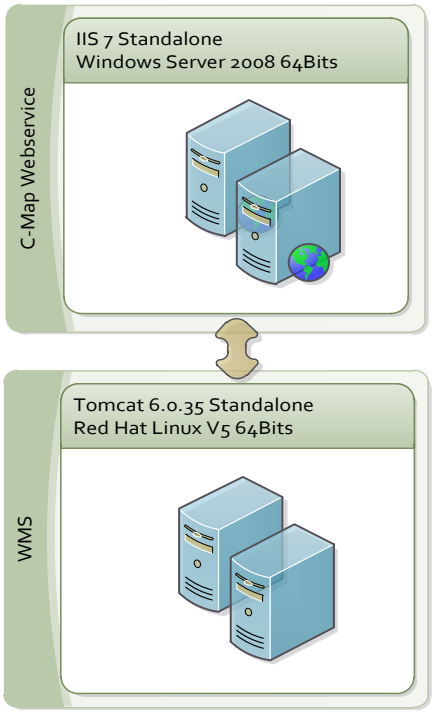


Figure 10 - ENCWS HW environment

WMS Non-Clustered: The WMS is composed of two independent servers running Tomcat Application Server 6.0.35 environment with a GeoServer 2.1.3 installation, on top of a Red Hat Linux V5 64 bits operating system.

Load balancing and transparent fail-over are provided by F5 Reverse Proxy.

Minimum hardware recommendation (identical for each of the 2 servers):

- 1 CPU with 2 cores at 2GHz;
- 8 GB RAM;

Integration Non-Clustered: The Integration Non-Clustered is composed of two independent servers running IIS 7 environment with a C-Map Web Services installation, on top of a Windows Server 2008 64bits operating system.

Load balancing and transparent fail-over are provided by F5 Reverse Proxy.

Minimum hardware recommendation (identical for each of the 2 servers):

- 2 CPU with 2 cores at 2GHz;
- 8 GB RAM;

6.2.2 Pre-production environment

The pre-production environment requires having the same computing capability of the production environment.

The following picture illustrates the pre-production environment.

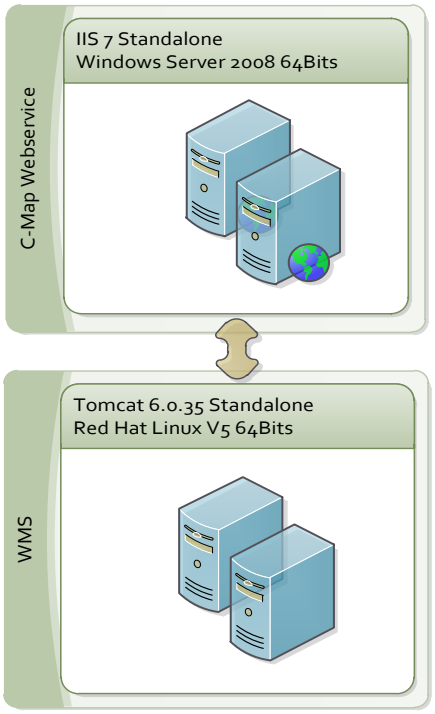


Figure 11 - ENCWS HW environment

WMS Non-Clustered: The WMS is composed of two independent servers running Tomcat Application Server 6.0.35 environment with a GeoServer 2.1.3 installation, on top of a Red Hat Linux V5 64 bits operating system.

Load balancing and transparent fail-over are provided by Reverse Proxy.

Minimum hardware recommendation (identical for each of the 2 servers):

- 1 CPU with 2 cores at 2GHz;
- 8 GB RAM;

Integration Non-Clustered: The Integration Non-Clustered is composed of two independent servers running IIS 7 environment with a C-Map Web Services installation, on top of a Windows Server 2008 64bits operating system.

Load balancing and transparent fail-over are provided by F5 Reverse Proxy.

Minimum hardware recommendation (identical for each of the 2 servers):

- 2 CPU with 2 cores at 2GHz;
- 8 GB RAM;

6.2.3 Test/Development environment

The testing/development environment is identical to the pre-production environment once this environment is already very minimalistic.

The testing/development is not required to have the same computing capability of the pre-production environment and it actually follows the 50% rule-of-thumb indication relating it to the production capacity.

The following picture illustrates the test/development environment.

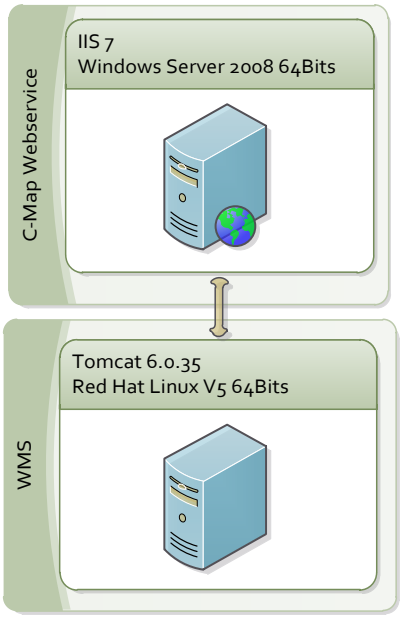


Figure 12 - Test/development HW environment

WMS Server: The WMS Server is running a Tomcat 6.0.35 instance with a GeoServer 2.1.3 installation, on top of a Red Hat Linux V5 64 bits operating system.

Minimum hardware recommendation:

- 1 CPU with 2 cores at 2GHz;
- 4 GB RAM;

Integration Server: The Integration Server is running an IIS 7 with C-Map WebServices installation, on top of a Windows Server 2008 64 bits operating system.

Minimum hardware recommendation:

- 1 CPU with 2 cores at 2GHz;
- 4 GB RAM;

7 Annex I – Software tools description

7.1 Jeppesen SDK

The technology has been developed by C-MAP to provide ***complete electronic chart services*** for mariners and electronic chart systems manufacturers

C-MAP Professional SDK is designed in full compliance with IMO Performance standard for ECDIS, IEC 61174, IHO S-57 edition 3, and S-52 edition 3 standards, covering a large part of requirements pertaining to the data processing and presentation of electronic charts. Nevertheless, the library provides the developer with a set of powerful tools for the rapid development of reliable applications ensuring correct implementation of CM-93/3 distribution conventions and electronic chart presentation.

One of the key elements and advantages of CM-93/3 electronic chart technology is automatic real time provision of licenses to the mariners which guarantees access to the new electronic charts and updates. This is provided by C-MAP Distribution Center, and should be supported by all electronic charting systems (ECDIS and ECS), which have been subscribed to and use the CM-93/3 service.

C-MAP provides complete cartographic service to ECDIS/ECS users, and is certified for correct implementation of CM-93/3 distribution conventions.

7.2 GeoServer

GeoServer is an open source software server written in Java that allows users to share and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source

using open standards.

Being a community-driven project, GeoServer is developed, tested, and supported by a diverse group of individuals and organizations from around the world.

GeoServer is the reference implementation of the Open Geospatial Consortium (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS). GeoServer forms a core component of the Geospatial Web.

7.3 Tomcat

Apache Tomcat is a widely used implementation of the Java Servlet Specification, which has been developed as an open-source project by the Apache Software Foundation since 1999, when the project source was donated to the ASF by Sun Microsystems.

Tomcat is actually composed of a number of components, including a Tomcat JSP engine and a variety of different connectors, but its core component is called Catalina. Catalina provides Tomcat's actual implementation of the servlet specification;

The name "Catalina," according to Craig McClanahan, who designed the original architecture of the servlet container, can be attributed to three things: his love for Catalina Island (despite never having visited it), his cat's habit of hanging around the computer while he was writing the code, and the consideration, at an early stage of development, of building Tomcat on a server framework called Avalon, which is the name of a town on Catalina island.

The Avalon framework was eventually abandoned, but the name stuck, and the rest is history.

7.4 Microsoft Internet Information Services (IIS)

Internet Information Services (IIS) 7 provides a new request-processing architecture that includes:

- A new service, Windows Process Activation Service (WAS), which enables sites to use protocols other than HTTP and HTTPS.
- A Web server engine that can be customized by adding or removing modules.
- A new approach to processing requests, integrating the request-processing pipelines from IIS and ASP.NET.

7.4.1 IIS 7 Components

IIS 7 contains several components that perform important functions for the application and Web server roles in Windows Server® 2008 (IIS 7.0) and Windows Server 2008 R2 (IIS 7.5). Each component has responsibilities, such as listening for requests made to the server, managing processes, and reading configuration files. These components include protocol listeners, such as HTTP.sys, and services, such as World Wide Web Publishing Service (WWW service) and Windows Process Activation Service (WAS).

7.4.2 Protocol Listeners

Protocol listeners receive protocol-specific requests, send them to IIS for processing, and then return responses to requestors. For example, when a client browser requests a Web page from the Internet, the HTTP listener, HTTP.sys, picks up the request and sends it to IIS for processing. Once IIS processes the request, HTTP.sys returns a response to the client browser.

By default, IIS 7 provides HTTP.sys as the protocol listener that listens for HTTP and HTTPS requests. HTTP.sys was introduced in IIS 6.0 as an HTTP-specific protocol listener for HTTP requests. HTTP.sys remains the HTTP listener in IIS 7, but includes support for Secure Sockets Layer (SSL).

7.4.3 World Wide Web Publishing Service (WWW service)

In IIS 7, functionality that was previously handled by the World Wide Web Publishing Service (WWW Service) alone is now split between two services: WWW Service and a new service, Windows Process Activation Service (WAS). These two services run as LocalSystem in the same Svchost.exe process, and share the same binaries.
In IIS 7, the WWW service no longer manages worker processes. Instead, the WWW Service is the listener adapter for the HTTP listener, HTTP.sys. As the listener adapter, the WWW Service is primarily responsible for configuring HTTP.sys, updating HTTP.sys when configuration changes and notifying WAS when a request enters the request queue.

7.4.4 Windows Process Activation Service (WAS)

Windows Process Activation Service (WAS) manages application pool configuration and worker processes instead of the WWW Service. This enables you to use the same configuration and process model for HTTP and non-HTTP sites.