Analysis of the Free GIS Software Applications in respect to INSPIRE services and OGC standards

Authors:

Francesco D'Alesio

Peter Hopfgartner

Paolo Viskanic

June 2011

Kanton Graubünden
Chantun Grischun
Cantone dei Grigioni

AUTONOME
PROVINZ
BOZEN
SÜDTIROL

PROVINCIA
AUTONOMA
DI BOLZANO
ALTO ADIGE

Executive Summary

This document analyses the main aspects related to Free Open Source Software GIS applications, the use of multilingualism in international standards and directives and the proposed architecture for the reference implementation to be developed by the FreeGIS.net project.

Chapter 1 introduces the FreeGIS.net project and its main components.

Chapter 2 contains an analysis of the Open Geospatial Consortium (OGC) standards and the INSPIRE (Infrastructure for spatial information in Europe) specifications to be addressed by the project and to be implemented in the Reference Implementation.

Chapter 3 analyzes the OGC standards and the INSPIRE specifications regarding the multilingual issue.

Chapter 4 analyzes the Free Open Source Software components that have been considered for the Reference Implementation. Paragraph 4.4 explains the software selection rationale for each layer.

Chapter 5 is the proposal for the Reference Implementation. Paragraph 5.1 illustrates the relationships between software and standards used for integration of the various services, while paragraph 5.2 contains a description of the Reference Implementation architecture.

Table of content

1. Introduction

This chapter introduce the FreeGIS Project, describing its goals, sponsors, partners and this report.

1.1. The FreeGIS Project

The Free GIS Project aims at satisfying the handling, the analysis and the publication of geographic information through the use of open data formats and Free Open Source Software and open standards.

FreeGIS is an Interreg Italy-Switzerland project financed by the Program of superregional cooperation Italy-Switzerland 2007-2013 of the ERDF (European Regional Development Fund) and the European Union. It is an in-depth analysis of the Interreg Italy-Switzerland project called CoCOS and it is consolidated through the Software Centre South Tyrol of the TIS innovation park in Bolzano

The aims of FreeGIS are:

- to identify the characteristics of the development of multilingual capabilities of GIS software;
- to define the details of a Reference Implementation composed exclusively by Free Software components;
- to identify a FreeGIS data licence, which is appropriate for the community and national requirements.

All actions are preceded by an analysis of the current situation and will result in the publication of guidelines dealing primarily with the needs of Public Administrations.

This report deals with the analysis of existing standards in the field of Geographic Information Systems and analyses their suitability for multilingual environments. Furthermore it analyses existing Free Open Source GIS Projects and their compliance with these standards. Finally a basic structure of the Reference Implementation is proposed, in order to meet the aims of the project: a complete stack of free open source applications, which communicate between each other on the basis of OGC Standards and which are compliant to the INSPIRE Directive.

2. INSPIRE and OGC services

This chapter introduce both OGC standards and the INSPIRE Directive. It describes the services required by INSPIRE, mapped to the relative OGC standards.

In the last paragraph, it explains the INSPIRE Data Specification guidelines for publishing compliant data themes.

2.1. Introduction

The INSPIRE Directive establishes an infrastructure for spatial information in Europe to support Community environmental policies, and policies and activities which may have an impact on the environment.

To ensure that the spatial data infrastructures of the Member States are compatible and usable in a Community and trans-boundary context, the Directive requires that common Implementing Rules (IR) are adopted in a number of specific areas (Metadata, Data Specifications, Network Services, Data and Service Sharing and Monitoring and Reporting).

The Open Geospatial Consortium (OGC®) is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services.

The INSPIRE RoadMap requires that Discovery and View services are operational by November 2011, while Transformation and Download services need to be operational by December 2012.

Therefore, this early analysis is focused on Discovery and View services.

Implementing Rules are mainly based on OGC and ISO standards. However, OGC standards need to be expanded with additional parameters and functionalities in order to match the INSPIRE needs.

2.2. Discovery Service

The Discovery service is based on the OGC CSW (Catalogue Services for the Web Specification, version 2.0.2) standard. It is intended to create a comprehensive resource where users can find all the metadata of the information published through INSPIRE services. It will also be harvested by the INSPIRE spider to collect and aggregate data in the Community level geo-portal.

The Discovery service INSPIRE standard operations are mapped to CSW standard operations, with additional parameters and functionalities, listed in the table below.

INSPIRE Operation	OGC Operation	Additional Parameters
Get Discovery Service Metadata	OGC_Service.GetCapabilities	Language
Discover Metadata	CSW.GetRecords	Language Query
Publish Metadata	CSWT Manager.Transaction CSWT Manager.Harvest	
Link Discovery Service	OGC_Service.GetCapabilities + CSW.GetRecords	

Part of the Discovery service must be implemented in the referenced OGC Service. For example, the "Get Discovery Service Metadata" for a WMS service, must be implemented in the WMS itself, extending the GetCapabilities request with the INSPIRE Discovery Service needs. This logic also applies to the Link Discovery Service operation.

2.3. View Service

The View service is based on the OGC WMS (Web Map Server Specification, version 1.3.0) standard and the OGC WMTS (Web Map Tile Service, version 1.0.0). It is mainly intended to serve map images and metadata.

While the WMS standard is intended to provide an “on the fly” map image generation, the WMTS provides map images starting from images (tiles) stored in the filesystem (cached). The server time needed to serve images from tiles is markedly less than to create images on the fly.

The View service INSPIRE standard operations are mapped to WMS standard operations, with additional parameters and functionalities, listed in the table below.

INSPIRE Operation	OGC Operation	Additional Parameters
Get View Service Metadata	WMS.GetCapabilities	Language
Get Map	WMS.GetMap	Language
Link View Service		

The GetCapabilities response must be extended with the necessary information to redirect users to the right Discover Metadata request.

This behaviour is intended to redirect users looking for WMS INSPIRE extended metadata to the corresponding catalog service.

The WMS.GetCapabilities response must also be extended with the following elements: Spatial Data Service Type, INSPIRE keywords, Temporal Reference, Conformity, Metadata Point of Contact, Date and Language.

As operating a WMS can be very demanding for performance reasons, INSPIRE accepts also a tiling service, based on OGC WMTS standard.

The mapping between WMTS and the INSPIRE View service operations is the same as above.

Tiles are squared images, cached from a WMS server, composing a mosaic of the bounding box at the different defined scales. Tiles are organized in a TileMatrix, that defines the position of the different images.

Every TileMatrix is part of a TileMatrixSet. The coordinate reference system as well as the scale set is defined at the TileMatrixSet.

To keep interoperability between different WMTS services, INSPIRE recommends to use the InspireCRS84Quad MatrixSet only.

The Implementing Rules also specifies how layers have to be organized and named.

2.4. Providing INSPIRE Services

INSPIRE provides Data Specification guidelines for various themes (Administrative Units, Cadastral Parcels etc.).

These specifications shall be matched by the data providers, so the available data must be extracted, transformed to the schema ordained by INSPIRE and loaded to the services software.

This process (ETL: Extract, Transform, Load) depends on the chosen software for the publication of the INSPIRE services.

For example, the DeeGree INSPIRE node requires an INSPIRE compliant GML, while GeoServer or MapServer could use database views or shapefiles.

Also, INSPIRE Data Specifications establishes that every published feature shall have an INSPIRE unique ID. This means that the transformation process needs to keep track of the relationships between the INSPIRE ID and the source database feature ID.

The ETL process is probably the most complex part of the efforts needed to setup INSPIRE services, so the transformation complexity should be considered in the choice of the publication software.

3. Multilingualism in Web Mapping standards

This chapter analyzes the differences between the OGC Standards and the INSPIRE specification regarding to multilingual issues.

3.1. OGC Web Service Common

The OGC specifies rules for multilingual web services in the "OGC Web Services Common Standard" document.

This applies to all the OGC web services specifications (WMS, WFS, WCS, catalog, etc.).

These rules establish that every OGC web service shall report the list of the accepted language in the capabilities document. Also, it shall accept an "acceptLanguages" parameter for every request.

The optional acceptLanguages parameter lets the user specify a list of desired languages, in order of preference.

The server shall serve the response in one of the specified languages. If the server cannot serve the response in any of the specified languages, than it shall raise an InvalidParameterValue exception.

There are two methods of defining a language: predetermined or best-effort.

In the first case, the client get the languages list from the capabilities document, let the user choose from one of them and use that parameter for the subsequent requests.

In the second case, the client ignore the list of languages reported in the capabilities document, and instead use the AcceptLanguages parameter. In this case, there is no guarantee that the response will uniformly contain the same language since each text string will be returned in the best available language as specified in the AcceptLanguages list.

The language tags shall be RFC 4646 5 character codes either, complete (e.g. "en-CA"), or abbreviated 2 character codes (e.g. "en"). In addition to the RFC 4646 codes, the server shall support the single special value "*" which is used to indicate "any language".

If the client specifies an abbreviated code (e.g. "en"), the server could use a language whose the abbreviated code is a prefix to (e.g. "en-CA").

Viceversa, if the user specifies a complete code (e.g. "en-CA"), while the server expose an abbreviated code (e.g. "en"), the response shall be an exception.

3.2. INSPIRE Implementing Rules

Regarding the multilingual handling, the INSPIRE Implementing Rules and the OGC standards do not match.

First of all, INSPIRE needs that the language list is inserted in the inspire:common ExtendedCapabilities section, while OGC consider it as part of the core specification for every web service.

Also, INSPIRE uses the "language" parameter, while OGC uses the "acceptedLanguages" parameter.

Then, INSPIRE requires the translation of a specific set of attribute response. OGC Common standard asserts that "For each human language text string in the server's response, including strings plotted into graphic images (such as in a WMS GetMap response), the server shall return that text string in most preferred language it has available". So OGC is much more demanding than INSPIRE.

Finally, INSPIRE uses language codes based on alpha-3 ISO 639-2/B, while OGC uses the RFC 4646 tags for identifying languages.

There aren't conflicts between these two specifications. However, they are mostly overlapping defining two different ways to address the same feature.

OGC asserts that "Servers that ignore the AcceptLanguages parameter entirely are trivially compliant", so if an INSPIRE service is needed, OGC multilingual constraints could be totally ignored. However, to be both INSPIRE and OGC compliant, a server shall listen to both acceptLanguages and language parameter.

It is so hard to establish the server behaviour for such a case.

Surely, the best solution is to drop INSPIRE multilingual constraints and use the OGC ones. In fact, the INSPIRE implementing specifications assert that "..with the availability of future versions of the OGC base standards the recommended approach to support multilingualism may need to be revisited.

4. Free WebGIS Software Analysis

This chapter explains the structure of the Reference Implementation WebGIS application, illustrating the two different usages and the software that has been considered for this analysis (paragraph 4.1) for the different layers.

Then, in paragraph 4.2, it shows how software can work together and the constrains for each server-side choice.

The last paragraph (4.3) explains the software selection logic for the different layers.

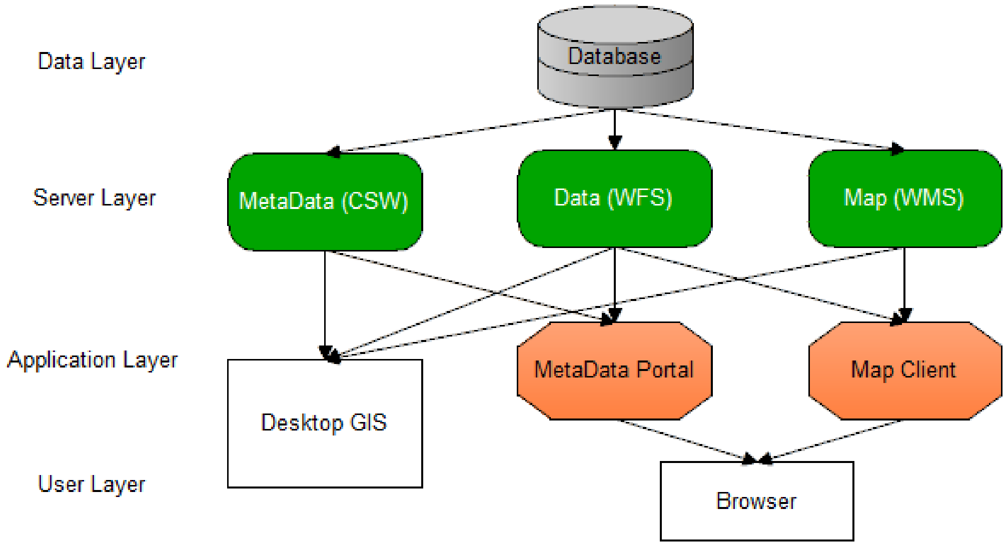
4.1. WebGIS applications structure

A WebGIS application consists in a stack of software able to serve geographical information over the web.

The following WebGIS structures considers the full components involved, from Data to Users, satisfying the needs specified by the FreeGIS Reference Implementation features.

4.1.1. Get Data

A (GET) WebGIS is structured as shown in the diagram below.



The Data Layer stores data: this operation can be handled with a Spatial enabled DBMS (like PostgreSQL or MySQL), with files (like ESRI Shapefiles, XML or JSON files) or using a WFS service.

The Server Layer gets data from the Data Layer, performs operations on data and publish them in mainly three OGC services: CSW (for catalog information), WFS (for raw data) and WMS (for map images).

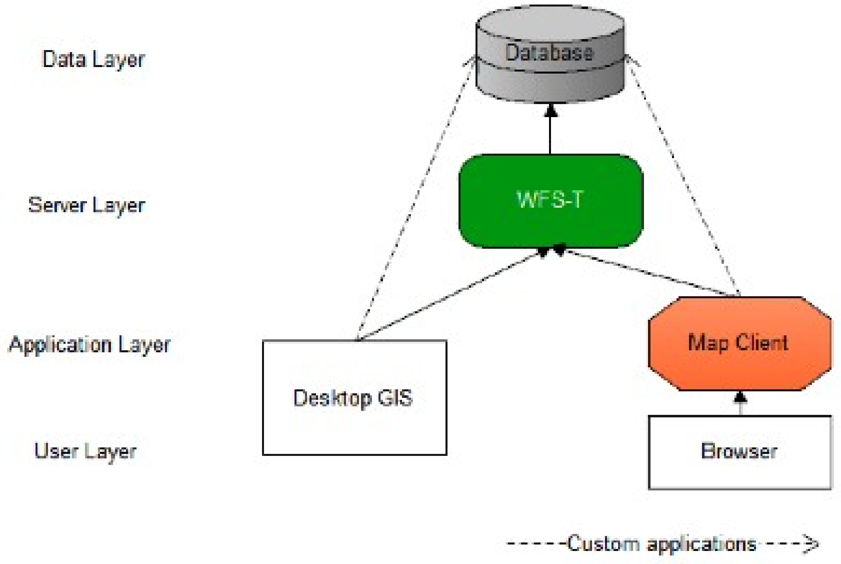
These services can be used both with a Desktop GIS application (like gvSIG or GRASS) or through web applications.

Finally, users can use WebGIS data using a browser (like Chrome or Firefox), with a Desktop GIS application or with mobile software (like MIXARE).

The stack above represents an OGC compliant web application. It is although possible to publish data in other formats.

4.1.2. Edit Data

An Edit WebGIS is structured as shown in the diagram below.



Users can edit data with a Desktop GIS application or with an internet browser through a web application. Data is then submitted to a WFS-T service that pass edits to the Data Layer.

Outside OGC standards, data can be edited from a Desktop GIS application that directly save edits in a server DBMS. Also, web applications can offer the possibility to edit data using custom systems.

4.2. WebGIS software

4.2.1. Data Layer

PostgreSQL + PostGIS

PostgreSQL is an open source RDBMS, able to provide spatial functionalities through its extension PostGIS.

Actually, PostGIS is the best SpatialDBMS: it is largely used, it has a big and active community, it is following the Simple Features SQL OGC standards and it implements lots of additional and useful functions.

Its next version 2.0, will make headway in reaching PostgreSQL proprietary competitors, supporting rasters and topology.

MySQL

MySQL provides a subset of the Simple Features SQL OGC standard functionalities.

Shapefiles

Shapefile is probably the most used format for spatial data. It is developed and regulated by Esri as an open specification for data interoperability among Esri and other software products.

It surely has the advantage of being totally portable and easier to approach with for end users. The great part of GIS applications are supporting it.

Text files

There are different file formats able to store geographical features. OGC provides specifications for GML and KML, both XML-based language.

The other text file format often used in Web Applications is the GeoJSON format, lighter than XML.

GPX is another text file for geographical information, mainly used by GPS devices.

Data can be also used directly from a WFS service, using the GML format. This option is actually hardly used in a production environment because of performance issues.

4.2.2. Server Layer

MapServer

MapServer is a geodata rendering engine, implementing WMS, WFS and WCS standards from several open and proprietary data source providers.

Actually, MapServer is the most used WMS server software. It has active developers and users community and a prompt mailing list.

It is written in C, it runs on various operating systems (Windows, GNU/Linux, Mac OS X, etc) and it supports many programming languages.

It is able to get features from many proprietary and open spatial DBMSs, shapefiles, GML (also cascading WFS), KML and GPX.

It is released under an MIT-style license.

GeoServer

GeoServer is an easy-to-use software that allows users to view and edit geospatial data.

It provides WMS, WFS (-T), and WCS services.

It is fully featured, it includes an author module to publish data and a client module.

It is written in Java, so it runs on every operating system supporting it (Windows, GNU/Linux, Mac OS X, etc).

It is able to get and edit features from both proprietary and open spatial DBMSs, shapefiles and cascading WFSs.

It is released under the GNU GPLv2 license.

Deegree

Deegree is a comprehensive geospatial software package with implementations of OGC Web Service, a geoportal, a desktop application, security mechanism, and various tools for geospatial data processing and management.

It claims to be INSPIRE ready for View and Download services, and it provides an example package configured to serve all Annex I Data Themes.

It can import data from a GML source (WFS or text file) and provide WFS, WMS and CSW services.

It is written in Java, so it runs on every operating system supporting it.

It is released under the LGPLv3 license.

QGIS Mapserver

QGIS mapserver is an open source WMS (1.3.0 and 1.1.1) implementation. In addition, it implements advanced cartographic features as specified in the Map and Diagram Service specifications.

Its main goal is to provide an easy-to-configure way to publish WebMaps. It gives users the possibility to publish the project edited in QGIS Desktop application.

It is written in C++ and it runs on GNU/Linux, Windows and MacOSX.

It is released under the GPLv2 license.

GeoNetwork

GeoNetwork is a catalog application to manage spatially referenced resources.

It provides metadata editing and search functions as well as an interactive map viewer.

It is written in Java, so it runs on every operating system supporting it.

It can import WMS and WFS metadata, and there are plans to integrate it with GeoServer.

It is released under the GPLv2 license.

TinyOWS

TinyOWS server implements the latest WFS-T standard. It is intended to complete MapServer with WFS-T capabilities and there are plans to integrate it MapServer.

PyWPS

PyWPS is an implementation of the WPS standard. It is intended to add geoprocessing functionalities to web and desktop applications from centralized software.

It is written in Python, it runs on GNU/Linux and Windows and it can use functions from GRASS, R as well as custom defined functions. It provides also an OpenLayers class to use it. It is released under the GPL license, version 2 or laters.

North52° WPS

North52° WPS is another implementation of the WPS standard. It is written in Java, so it runs on every operating system supporting Java. It has client plugins for OpenLayers, uDig and OpenJump to ease its usage, and connectors to GRASS, Sextante, FME and ArcGIS. It is released under the GPLv2 license.

Zoo Project

Zoo Project is another WPS implementation. I'ts main goal is to give developers the ability to write processes in various programming languages, using all the libraries available. It is released under the MIT/X-11 license.

4.2.3. Web Application Layer

OpenLayers

OpenLayers is a Javascript library with no server-side dependencies. It is the basis for several Web Mapping projects (some of them are listed below). It is able to serve maps from OGC standard services, query features through WFS and edit data through WFS-T servers.

It is surely the most used map library, with a very active community. It is well documented, fully customizable and easily extensible.

GeoEXT

GeoEXT is a javascript library combining OpenLayers with ExtJs, a toolkit for rich web application.

It provides a set of predefined common functionalities (such as legend panel, layer tree and search forms) to fasten the development of a rich web mapping application.

OpenScales

OpenScales is a mapping framework designed for building Rich Web Mapping Applications easily.

It is written in ActionScript3 and Flex. It initially started from as a Flex port for OpenLayers so its API and functionalities are very similar to the OpenLayers ones.

It can be used both for browser-based applications and for desktop-based applications, providing the ability for users to work offline with local resources, in the same environment as the online version.

MapFish

MapFish is a complete framework for building rich web-mapping applications.

It is written in Python, so it runs on every platform supporting it.

It has a "studio" component that aid users to configure and style web maps.

Also, it provides a Javascript toolbox based on OpenLayers and GeoEXT, able to serve data from different Map server software (MapServer, Geoserver etc).

It is released under the BSD license.

Gisclient

Gisclient is a complete framework for building rich web-mapping applications.

It has an “author” component with a GUI interface that helps users configuring MapServer. It also provides configurations for the “client” component.

The “client” component is a rich web mapping application with common features such as printing, editing, querying etc.

It is written in PHP for the server side and Javascript for the client side.

It is released under the GNU/GPLv3 license.

P.mapper

P.mapper is a framework intended to facilitate the setup of a MapServer application based on PHP/MapScript.

It includes both server and client functionalities.

It is released under the GPLv2 license.

MapBender

Mapbender is the back office software and client framework for spatial data infrastructures.

It is written in PHP for the server side and Javascript for the client side.

It is released under both the GPLv3 license and the Simplified BSD license.

Geomajas

Geomajas is a complete web mapping framework written in Java, with both a server component and a rich web mapping application. It provides lots of built-in functionalities to easily create a web GIS application.

It is released under the GNU Affero general public license (AGPL) v3.

4.2.4. User Layer

GRASS

GRASS is a fully featured desktop GIS, written in C and running on many operating systems.

It provides a wide range of functions for spatial analysis on both vector and raster data.

It can be linked to most of the spatial DBMS (both open and proprietary) with native support or ODBC driver. It is also able to get data from WMS and WFS services.

GRASS functions are also often used in other desktop GIS applications like QuantumGIS, gvSIG and uDig (described below).

QuantumGIS

Quantum GIS (QGIS) is a user friendly desktop GIS. It runs on GNU/Linux, Unix, Mac OSX, and Windows and supports numerous vector, raster, and database formats and functionalities.

It is able to get data from WMS and WFS services as well as editing data through WFS-T services.

It is also able to get and edit data from PostGIS and SpatiaLite.

It is released under the GNU license.

GvSIG

GvSIG is very similar to QGIS, although it is written in Java and support some additional functionalities (such as a 3d viewer and topology support).

It is released under the GPLv2 license.

Udig

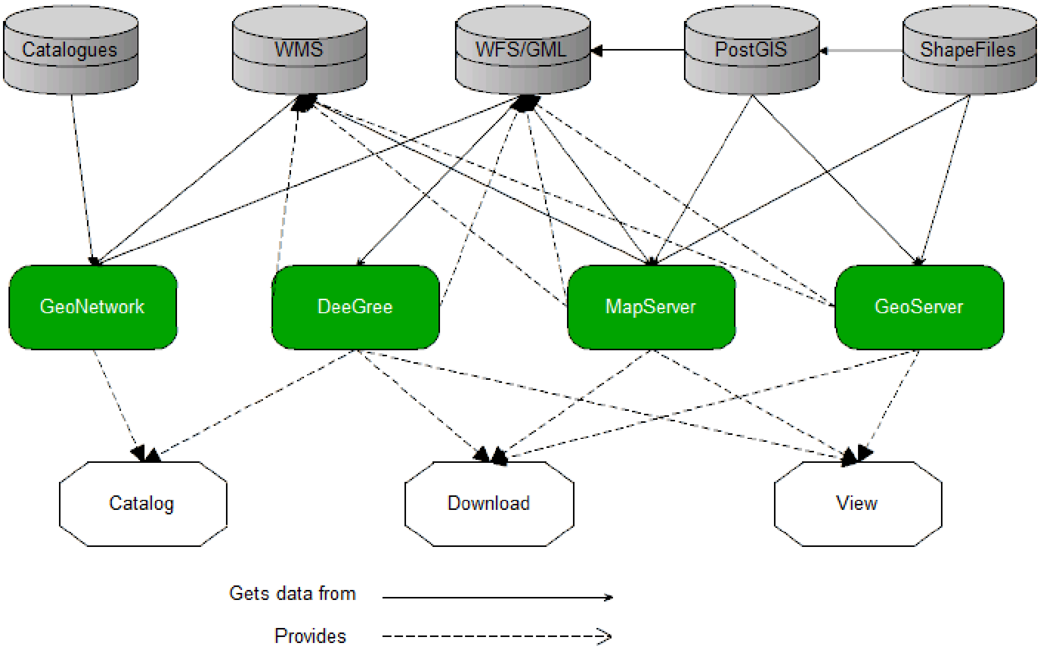
uDig is a desktop application framework for GIS. It is written in Java.

Its main goal is to provide an user friendly interface for spatial webservices and it is mainly intended to build custom desktop applications on it.

Udig is released under the LGPLv2.1 license.

4.3. Software Relationships

The implementation of the INSPIRE services involves several software. The following diagram exposes the possible relationships between the software analyzed starting from data sources and ending with the three required INSPIRE services.



As shown in the diagram above, thanks to standard formats and interoperability, there are many ways to get from a spatial database to an INSPIRE service.

However, none of these software are able to provide an exact compliance with the INSPIRE requirements. Even if all the communities involved in the development of these software are aware of the INSPIRE needs, and some of them (especially DeeGree) are addressing their efforts mainly on INSPIRE implementing rules, every choice will entail some developments.

4.3.1. GeoServer

GeoServer is now working to support WMS 1.3.0 and WFS 2.0, required by INSPIRE.

At the moment, they are not addressing the multilingualism and additional metadata problems.

As shown in the diagram above, GeoServer is able to get data from various sources. It can handle the catalog services and there are plans to integrate GeoNetwork, so that linkage between View/Download and Discovery services can be kept automatically.

The ETL process here is less demanding, while there is no needs to change the data format. So starting from a Spatial DB (like PostGIS), can results in the creation of a view.

4.3.2. MapServer

MapServer is already compliant with WMS 1.3.0, but they are not working on WFS 2.0.

Actually, MapServer can not serve additional INSPIRE metadata, nor is supporting multilingual capabilities.

However, the MapServer development team is actively working on this: they already coded patches for additional INSPIRE capabilities and a basic multilingual support.

They should also add flexibility to the layer definition: actually it is not possible to create layers with different geometry types, while the INSPIRE Data Specification requires this in different themes.

There is already a feature request for this problem, already approved by the Project Steering Committee.

The multilingual support is limited at the moment to the getcapabilities request/response (as required by INSPIRE). There are not any plans to extend this to other operations.

As shown in the diagram above, MapServer (as GeoServer) can get data from different sources. It can also handle a catalog CSW compliant service but, unlink GeoServer, there is not the possibility to link View and Download service to an external Discovery service (like GeoNetwork).
Like GeoServer, the ETL process is less demanding.

4.3.3. Deegree

Deegree is already compliant with WMS 1.3.0 and WFS 2.0. It also has an INSPIRE ready implementation, with extended metadata support and basic multilingual capabilities.

As shown in the diagram above, Deegree can only get features from an INSPIRE ready GML file (or WFS service). Starting from this, Deegree can store features in various DBMS through JDBC connection.

Unlike MapServer and GeoServer the ETL process is more demanding: it needs an INSPIRE compliant GML file (or WFS service), so the transformation process shall handle a format transformation and data reprojection.

Deegree can directly serve a Discovery service CSW compliant. However, who implemented an INSPIRE service using Deegree, chose GeoNetwork for the Discovery service. The reason besides this needs additional investigations. However, unlike GeoServer, Deegree is not integrated with GeoNetwork, so linkage between View/Download and Discovery services shall be kept manually.

4.4. Software selection

Unlike with proprietary software, the choice of a free open source software should consider some additional parameters.

Free Open source software are nearly often distributed without support. This means that the ability to easily find an experienced partner that could eventually support the development of the project is an important parameter.

Also, in case that a bug is blocking the project or a new functionality is needed, there should be the possibility to contact someone able to workout the problem. This requires an active development group and a large user community.

Again, most of the time, free open source software are tested by users. This means that newborn or barely used software are likely buggy.

As shown in the diagram above, the software considered in this report are nearly completely interoperable, so every choice could be made almost independently from each other.

That said, the following paragraphs, will compare the different software for each application layer.

4.4.1. Data Layer

For the data layer PostgreSQL with PostGIS is surely the best choice: it is mature, it has an active community and it is largely used.
PostGIS can also directly and easily import Shapefiles. It can create geometries from GML, KML and WKT formats, providing the ability to import almost every geographical text format.
Shapefiles can also be used directly (without PostGIS) from MapServer, Geoserver and so on.

4.4.2. Server Layer

The choice of the map server software is probably the most critical: this is the core component of a geoportal and it is desirable to use the less possible number of components to serve all the required services.

For example, using Qgis MapServer for WMS and MapServer for WFS means double configuration of the same features, and nearly double effort to maintain and update the whole project.

That said, supposing that WMS, WFS and catalog services shall be implemented, Qgis Mapserver, although tests acclaim it as the fastest (and probably easiest) mapserver, it should be excluded as it can just serve WMS.

TinyOWS shall be considered only if it will be integrated with MapServer, as it will cover its lack of WFS-T support.

Finally, the choice is between MapServer, GeoServer and Deegree.

MapServer is the most used and oldest mapserver software. It has a very large community of users and developers and many companies are working and supporting it.

Also, there are lots of applications and clients supporting and enhancing MapServer aiding the user in setting up a WebGIS application.

The worst about MapServer is the not-so-friendly configuration: the user shall know the mapfile syntax in order to configure layers and styles. However, there are a few GUI mapfile editors.

Geoserver is probably more user-friendly, with a web based, well documented GUI. It supports WFS-T natively and it will probably be linked to GeoServer for catalog administration.

However, Geoserver is less used than Mapserver.

Deegree is an almost newborn software, with a small user community and with just a subset of the functionalities of MapServer and Geoserver. It needs data in GML format so probably data should be transformed almost "manually" and updates can be more difficult.

However, Deegree seems to be the only INSPIRE-ready open source mapserver. This will probably increase rapidly its community and its functionalities.

Regarding the processing functionalities, PyWPS seems to be the most advanced and more standard compliant implementation for WPS. PyWPS is able to use GRASS modules or statistical functionalities with R, as well as to define custom processing functions.

4.4.3. Web Application Layer

The choice of an application software depends on the functionalities required and the ability to develop: OpenLayers, GeoEXT and OpenScales are libraries or frameworks, while the other are complete applications (sometimes based on that libraries).

Complete applications just need some configurations, but they offer a fixed structure. It's often hardy to add new functionalities or to modify things.

Libraries and frameworks need developments: they speed up development, but the whole structure of the portal shall be developed "ad hoc". On the other hand, this choice allows full customisation and less constrains.

On the **libraries** side, OpenLayers is almost a "must": actually, there aren't valid competitors and a growing number of applications are using or are migrating to OpenLayers.

GeoEXT is built on OpenLayers, adding common GIS functionalities that could speed up the development of a complete WebGIS application.

OpenScales is just a branch of OpenLayers based on another browser technology (Flash instead of Javascript). The only reason why one should choose OpenScales instead of OpenLayers is its ability to work offline. If this functionality is not needed, OpenLayers is surely the best option.

On the **applications** side, MapFish and GisClient seems to be the best options for the most. They “complete” MapServer with a configuration GUI and they provide a feature-rich client.

GeoMajas is fully featured, with lots of built-in functionalities that probably cover the most part of every WebGIS application needs. On the other side, it doesn't seem to be very used and its community is very small.

P.mapper and MapBender are very similar, both in technology and functionalities.

The choice of an application shall consider first of all the adherence of its functionalities to the project specifications. If none the cited applications fits the project specifications, probably, the choice should go toward libraries and custom development.

5. Proposed Reference Implementation

This chapter contains the proposal for the FreeGIS Reference Implementation. Paragraph 5.2 contains a schema illustrating the relationships between software and OGC/INSPIRE standards.

5.1. Introduction

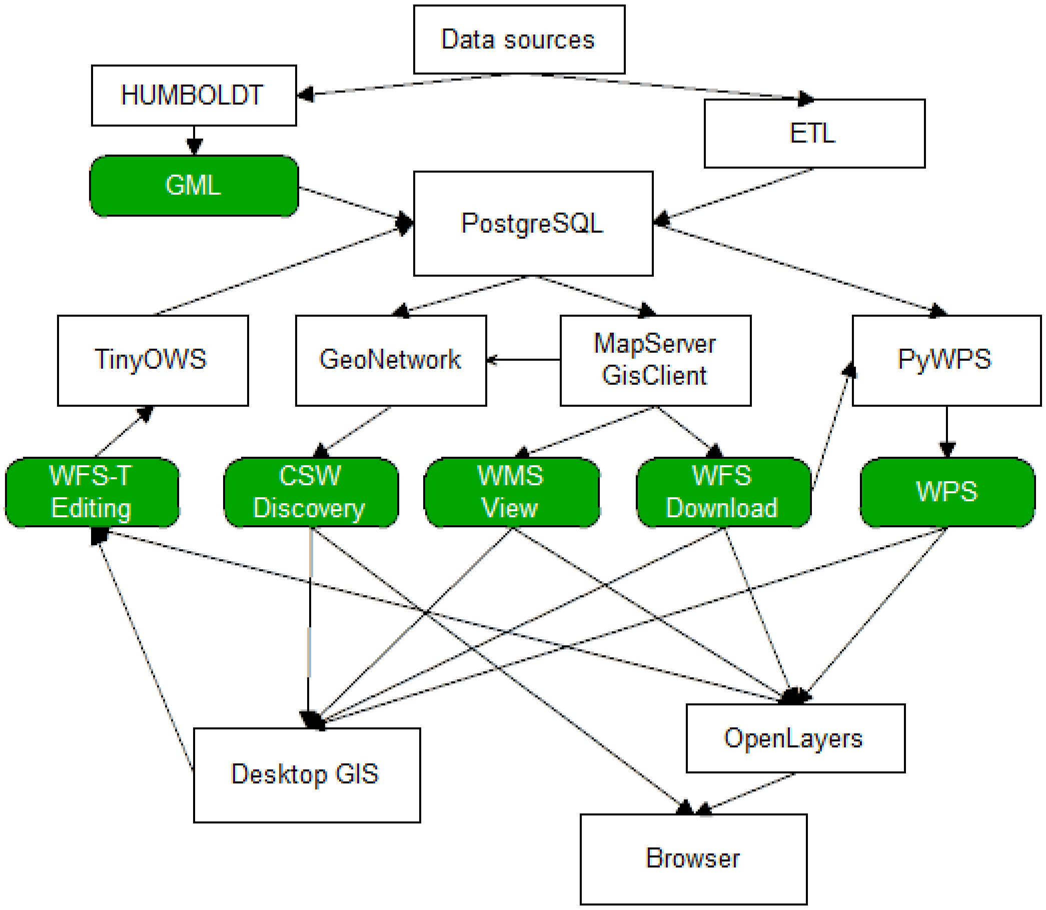
The FreeGIS Reference Implementation will satisfy the following needs:

- Publish data through standard web services, like WMS, WFS and CSW
- Publish data following the INSPIRE directive and implementing rules
- Edit data using the WFS-T standard
- Store data in a FOSS spatial DBMS
- Transform data from different data sources to standard formats and INSPIRE data specifications both with one-time transformations and automatic, on-the-fly transformations
- Offer geoprocessing functionalities on published data

5.2. Software

After evaluating the FOSS GIS software available, we chose the architecture described in the diagram below.

Green elements are OGC and/or INSPIRE standards, used for integration.



The following software will be involved in this implementation:

- The **HUMBOLDT** framework (http://community.esdi-humboldt.eu/)

  HUMBOLDT is a European project that aims to contribute to the implementation of a European Spatial Data Infrastructure, caring about the data harmonisations processes.

  The framework includes different software packages, covering almost all of the geodata transformation and harmonisation needs.
- **PostgreSQL** (http://www.postgresql.org/)

  It is the most advanced open source DBMS.
- **PostGIS** (http://postgis.refractions.net/)

  It adds support for geographic objects to the PostgreSQL DBMS, following the OpenGIS Simple Features Specification for SQL
- **MapServer** (http://mapserver.org/)

  It is the most used web mapping software.
- **GisClient** (http://www.gisclient.net/)

It provides a GUI to configure MapServer and a feature-rich webGIS application.

- **PyWPS** (http://pywps.wald.intevation.org/)

  It implements the WPS standard, providing geoprocessing functions.
- **GeoNetwork** opensource (http://geonetwork-opensource.org/)

  It is a catalog application to manage spatially referenced resources.
- **OpenLayers** (http://openlayers.org/)

  It provides a Javascript API to easily put dynamic maps in any web page.

5.3. Description

Starting from the different data sources, the user will be able to transform data using the software provided in the HUMBOLDT framework or custom ETL processes.

HUMBOLDT software can both transform data and provide transformation rules to be used in a WPS service. The output of this process is a GML file, that can be imported in a spatial enabled DBMS (like PostgreSQL) or directly in web mapping software (like MapServer or Deegree).

The user will be then able to configure a webGIS application, eventually compliant with the INSPIRE specifications, using GisClient for WMS and WFS and GeoNetwork for CSW.

The editing functionalities will be provided by TinyOWS, using the WFS-T protocol.

The geoprocessing functionalities will be provided using pyWPS. It gives the possibility to use functions from GRASS, R and, if needed, custom defined functions.

From a client perspective, the services provided will be available using a Desktop GIS standard compliant application (like gvSIG, GRASS or QGIS) or a web browser (like Firefox or Chrome).

In the browser case, the map functionalities will be provided by the OpenLayers Javascript library.

6. References

- Implementing Directive 2007/2/EC of the European Parliament and of the Council as regards interoperability of spatial data sets and services
- http://inspire.jrc.ec.europa.eu/
- INSPIRE Technical Guidance View Services v3
- INSPIRE Technical Guidance for Discovery services v3
- ISO 19115:2005(E): Geographic information – We map server interface
- ISO 19115:2003(E): Geographic information – Metadata
- Sudra Pawel, 2010 "INSPIRE-compliant web services"
- http://inspire.kademo.nl/doc/
- http://www.mapserver.org/
- http://www.geoserver.org/
- http://www.deegree.org
- http://karlinapp.ethz.ch/qgis_wms/
- http://tinyows.org/trac
- http://www.petascope.org/
- http://pywps.wald.intevation.org/
- http://geonetwork-opensource.org/