Free WebGIS Software Analysis

Authors:

Francesco D'Alesio

Peter Hopfgartner

May 2012

Free WebGIS Software Analysis

Executive summary

This report explains the structure of the Reference Implementation WebGIS application, which is being developed in the FreeGIS.net project. It illustrates different usage needs and available free and open source applications.

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.

Introduction

One of the goals of the FreeGIS project is to build a reference implementation based on Free and Open Source software able to satisfy the following needs:

1. Manage spatial data in a DBMS
2. Serve spatial data through standard and multilingual WMS, WFS and WCS.
3. Edit spatial data using both WebGI and desktop GIS applications.
4. Guarantee interoperability with common Web and Desktop GIS tools.

A rich WebGIS application is based on different components, explained in the chapters below.

There are a big variety of Free and Open Source GIS software available, and, while this is surely a big advantage, it makes harder to identify the software that better fits the different needs and that is able to interoperate with the other components.

Also, often, these software provide more than one component and they often overlaps with other.

Thanks to the wide adoption of the OGC standards, most of the available software are able to interoperate with each others, so components can mostly be assembled and exchanged leveraging standard capabilities.

Target

This report has been written for GIS experts who are interested in analysing the different open source software options available. In the FreeGIS.net project context the report is addressed tzo project team and partners to document the process followed in defining the sofdtware stack for the reference implementation.

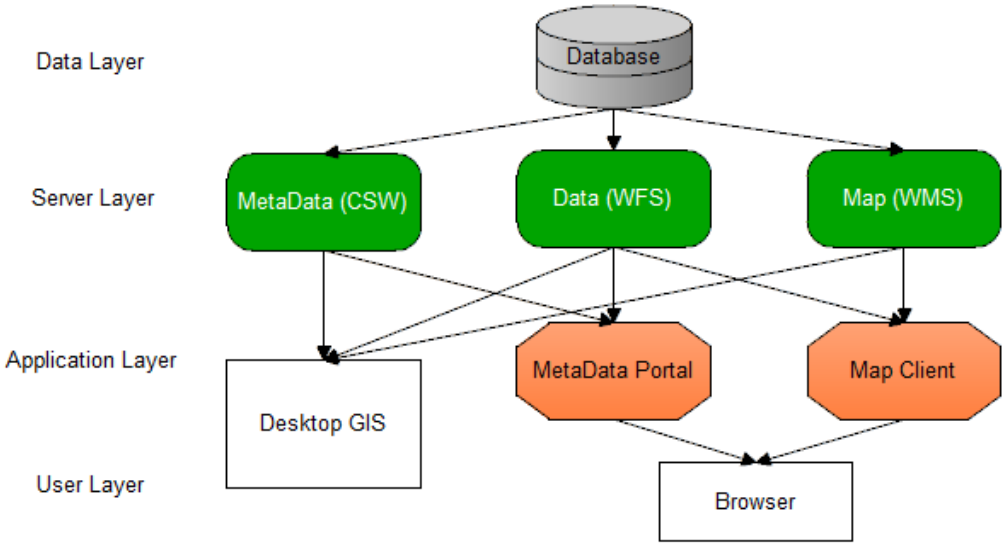
WebGIS applications structure

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.

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).

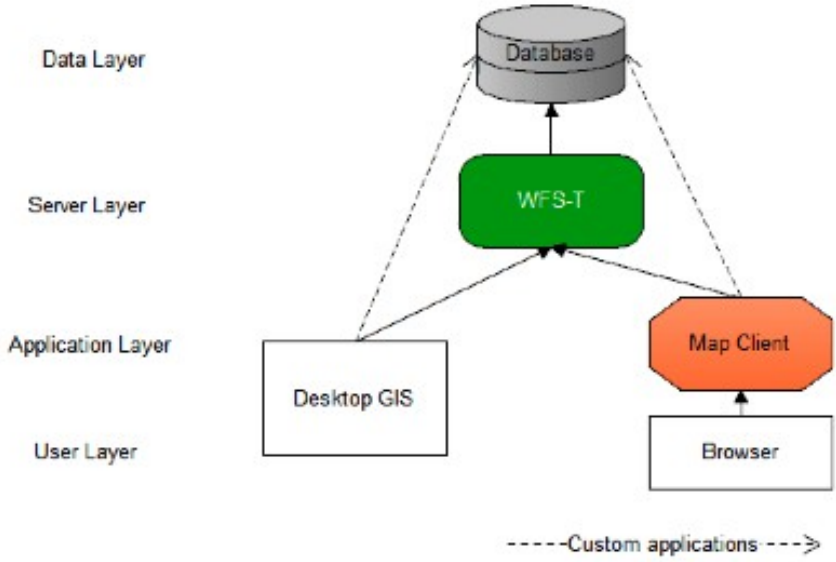
These services can be used both with a Desktop GIS application (like gvSIG or GRASS) or through web applications.

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.

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.

WebGIS software

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.

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.

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.

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

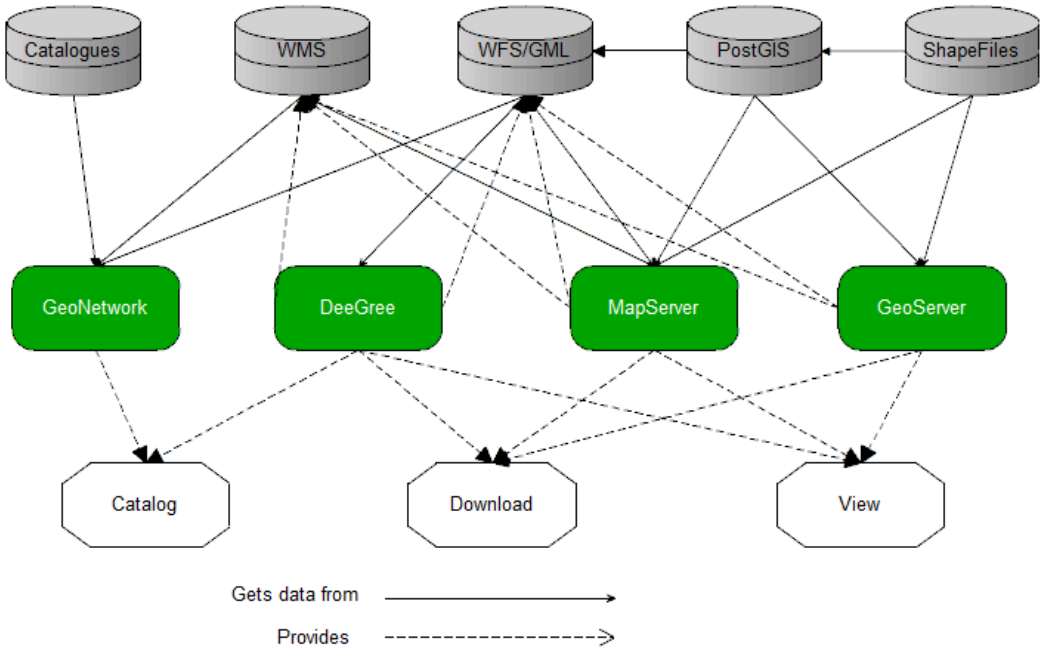
uDig is a desktop application framework for GIS. It is written in Java.

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.

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.

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.

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.

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.

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.

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.

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 GeoNetwork 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.

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.