TinyOWS, the high performance WFS-T server

Authors

- **Olivier Courtin,** Oslandia, *France*
- **Vincent Picavet,** Oslandia, *France*

KEYWORDS : WFS-T, High performance, OGC, OsGeo, INSPIRE

Modern GIS architecture make use of webservices to leverage web-based scalable applications. This lead to lightweight infrastructures, which are versatile and efficient. Moreover, OGC webservices standardisation enables those architectures to be fully interoperable. The INSPIRE directives supports OGC standards and creates a strong demands on softwares implementing those standards.

One of the need for GIS is to be able to access vector data, edit them and contribute modifications back to the data source. Of course this need to be available on top of an HTTP layer, to be able to build web-based or desktop applications on top. This use case is the one targeted by the WFS-T norm (Web Feature Service - Transactionnal). Accessing data through WFS-T helps improving the interoperability of the solution, and let us easily build applications on top of it, but it adds a layer on top of the data.

WFS-T is therefore a service for which performances are really important, as well standard compliancy, and software footprint, which should be the smallest possible. Presented in OGRS 2009 for the 0.9 release, the TinyOWS project found wider use as an official WFS-T Server for QGIS developers, and is now part of the MapServer Suite itself (since MapServer 6.2). TinyOWS [1] is OpenSource, (MIT licence), and part of the OSGeo

foundation, as it is now directly integrated under MapServer's umbrella. TinyOWS strictly implements WFS OGC standards [2] (versions 1.0.0 and 1.1.0), and passes all OGC CITE tests, including OGC CITE beta [3], which represents around 1000 unit tests. TinyOWS con be configured with a really simple and concise XML file which is edited manually, or directly with a MapServer MapFile. So you could use the very same config file to deploy both MapServer (as a WMS server for instance) and TinyOWS (as a WFS-T server). TinyOWS deeply relies on the database to perform stored data access and manipulations. As for now the PostGIS backend is the only one available, SpatiaLite and Oracle Spatial are planned for futures releases.

Current TinyOWS dev team gathers about 10 developers sending patches, enhancing features, or performing heavy tests on this application. The development infrastructure has now migrated to the Mapserver infrastructure, and the Mapserver and TinyOWS model of development have been adapted accordingly. This is a good example of how a free software project can grow and find a place in the ecosystem where it fits right, and be integrated to harness the power of complementarity.

TinyOWS is really fast, and designed to avoid CPU and memory waste. Some users who face really intensive WFS-T usages (>3000 simultaneous users) succeed to make it works on a single server architecture ! TinyOWS is therefore the perfect solution to save CO2 emissions and headaches (at least it you mind WFS-T and love webservices)...

The forthcoming TinyOWS 2.0 will bring a lot of new features, and the development will continue to TinyOWS mainly in three general directions :

→ INSPIRE compliancy

TinyOWS intends to implement the «Download Service» part of the directive [4]. The download service aims at providing a webservice to let a user download raw geospatial data. INSPIRE download service implies the implementation of at least a subset of the WFS 2.0 standards. Among the main aspects to be added, paging, stored procedure calls, language are among the features. Versionning is also a strong enhancement to what is currently provided by WFS-T services.

- More performance improvement

  There is still room for TinyOWS performance improvements, and a few ideas are already promising. One of the directions is to tighten the integration between the various components. TinyOWS could for example be plugged directly into Apache as a module, leading to more performance. Another direction would be to allow the gzipping/shipping part of TinyOWS to be run on a different server.

- Support more output formats

  Some new formats can be added to TinyOWS, on top of the currently supported ones (GML, GeoJSON). Shapefile, X3D, KML to name a few could be great to have. Customizable response XSD schema, and even other related specifications such as SOS-T (OGC Sensor Observation Service) are other interesting features that could be added too.

- Multiple database backend support

  Another area of improvement is the database backend. Only PostGIS is currently supported, but other backends could be interesting. The first step is probably to add an abstraction layer to the database backend, then add support for new backends. Good candidates include Oracle Spatial, and Spatialite. The latter would enable to have a very light WFT-T solution with embedded data. This would also mean that multi-connexion support would be added too, fixing a compatibility problem with mapserver which already supports multiple backends.

The workshop will allow each participant to successfuly :

- Install TinyOWS from sources (including libs and apps dependancies)
- Discover the first steps with TinyOWS demo installation and using QGIS as a WFS client
- Use QGIS as a WFS-T client, and check that data are really modified on database side
- Understand TinyOWS XML Configuration file in depth

- Know a step by step debug procedure when something does not work
- Use Fast-CGI and Apache Web server module, configure and compare performances
- Use MapServer as a WMS service and TinyOWS as WFS-T one, on a same dataset through a single common MapFile
- Use TinyOWS XSD schema to allow enhancing and extening GML output, for instance with CityGML output.
- Use TinyOWS GeoJSon output and OpenLayers as a WFS-T client

This workshop is therefore a complete showcase of TinyOWS software to leverage read/write webservices on GIS data. Full systems interoperability is a tough target to reach, and versatile but still highly compliant and robust software are a key factor in good system architecture implementation. Once interoperability is reached, then performance is the next hard point to get right, and is also a key factor for a successful INSPIRE / OGC web spatial data infrastructure. TinyOWS provides interoperability and performance, while keeping the configuration and complexity of the software very low.

[1] TinyOWS project homepage. TinyOWS website. Retrieved Sept 19, 2012 from http://www.tinyows.org

[2] OGC Web Feature Standard Implementation Specifications. Open Geospatial Consortium website. Retrieved Sept 19, 2012 from http://www.opengeospatial.org/standards/wfs

[3] OGC CITE Beta tests. OGC Compliance Testing Program website. Retrieved Sept 19, 2012 from http://cite.opengeospatial.org/betaTesting

[4] INSPIRE NETWORK SERVICES DRAFTING TEAM, 2009. Draft Technical Guidance for INSPIRE Download Services. Retrieved Sept 19, 2012 from http://inspire.jrc.ec.europa.eu/documents/Network_Services/INSPIRE%20Draft%20Technical%20Guidance%20Download%20(Version%202.0).pdf