

# ZOO

## THE POWERFUL WPS PLATFORM

Mr. Gérald FENOY, GeoLabs  
Dr. Nicolas BOZON, 3LIZ  
Pr. Venkatesh RAGHAVAN, OCU



# WHAT IS WPS ?

**Web Processing Service** is designed to standardize the way that GIS algorithms are made available on the Internet.

WPS specifies a mean for a client to request the execution of a spatial calculation from a service.

WPS intends to automate geoprocessing by employing geospatial semantics in a service-oriented architecture (SOA).



Open Geospatial Consortium interoperability standards since 2005



# WHY USING WPS ?

**Web Processing Service** allows to deploy and to orchestrate advanced GIS processes on the server-side.

WPS can connect to cartographic engine and spatial databases and thus drive complex spatial data infrastructures (SDI).

WPS is a generic and standardized way to use **GIS on the Cloud**.



WPS can run FOSS4G and OSGeo tools !



# WHAT IS ZOO ?

ZOO is a **WPS (Web Processing Service)** compliant server-side platform based on OGC's WPS 1.0.0

ZOO is **open source software** released under MIT/X-11 licence

ZOO is designed to create and chain web processing services easily, using **FOSS4G libs or existing code.**

ZOO is based on a C Kernel (**ZOO Kernel**) able to load dynamic libraries and to orchestrate Webservices coded in several programming languages.



# ZOO PROJECT GOALS

**Make FOSS4G libs communicate** in a standardized way using WPS

Make GIS-based Web Services deployment and chaining easier and faster

Create a ready-to-use **Webservices suite** based on stable libs

**Enable advanced Web GIS functionalities** such as:

- Conversion, reprojection and processing of data
- Advanced GIS algorithms
- External spatial-based scientific models



# HOW DOES ZOO WORK ? (1)

## ZOO Kernel handles and chains ZOO Services

A **ZOO service** is composed of:

A metadata file **.zcfg** (Title, Metadata, Inputs,Output...)

A Service Provider: « **Service Shared Object** » (SSO)  
( Dynamic library, Python modules, JAVA Class, PHP script ...)

WPS **GetCapabilities** and **DescribeProcess** requests requires only to parse .zcfg file using Flex and Bison

ZOO Kernel is able to load SSO dynamically, to extract specific functions and to execute them, answering the **WPS Execute** requests



# HOW DOES ZOO WORK ? (2)

- Execute request** parsing ( XML / KVP ) to fill internal data structure
- **Xlink:href checking** , conditional data download to provide input data value
  - **Dynamic loading** of the Service Provider Shared Objects
  - **Specific service function call**, passing internal data structure by reference
  - **ResponseDocument / Raw data output**, using the previous data structure (modified by the service itself)



# SUPPORTED LANGUAGES

**ZOO Kernel** supports several programming languages

- C / C++      Native support
- Python      Native support (Python interpreter)
- Fortran      Optionnal support (F77, F90)
- PHP      Optionnal support (PHP embedded)
- Java      Optionnal support (Java SDK)
- Javascript      Optionnal support (SpiderMonkey)





# ZOO PROJECT STATUS(1)

ZOO 1.0 released under MIT/X-11 license since April 2010:

ZOO Kernel ( C )

ZOO Services ( C and Python)

ZOO API ( Javascript )

svn checkout <http://svn.zoo-project.org/svn/trunk> zoo

ZOO 1.0 documentation and packaging in progress

Install guide and ZOO Workshop available on the wiki

ZOO 1.0 available on OSGeo-Live 4.0



# ZOO PROJECT STATUS(2)

## ZOO Services

Ogr2Ogr ( C )

GEOS/OGR ( C )

GdalTranslate (C)

GdalGrid (C)

OpenOffice Document Text Converter (Python)



<http://zoo-project.org/trac/browser/trunk/zoo-services/>

## ZOO API

Based on SpiderMonkey and ZOO's optional JavaScript support

Uses server-side JavaScript and a Proj4js adaptation

Add Logic in WPS chaining



# ZOO PROJECT EXPERIMENTS(1)

## WFS-T support implementation (Fenoy,G.)

MapServer 5.6.5  
ZOO 1.0  
WFS-T proxy



Provides a Transactional support to MapServer WFS

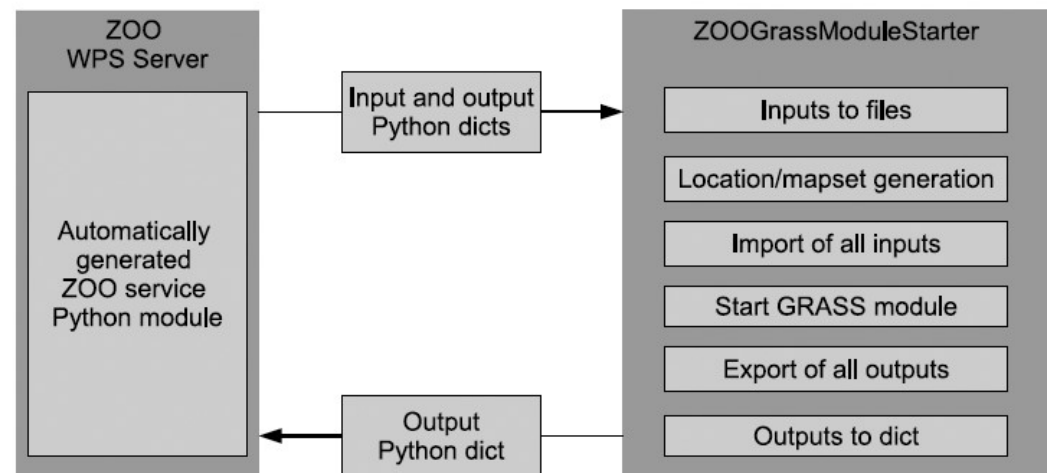
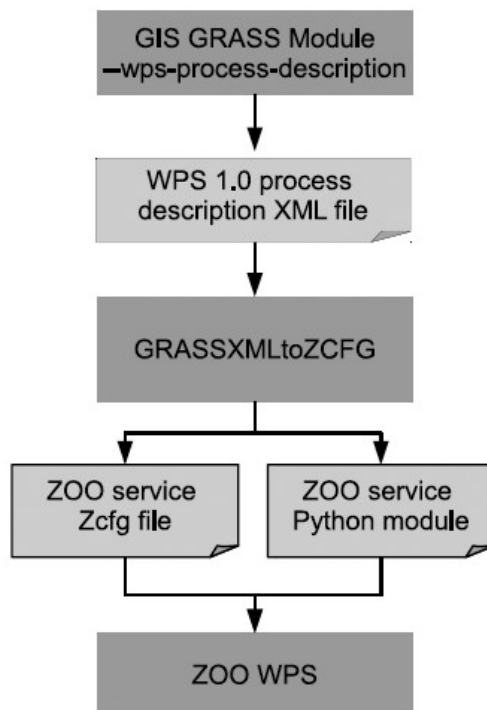
Edits the input data source geometry and attributes

Allows any data source input supported by OGR

# ZOO PROJECT EXPERIMENTS(2)

## GRASS GIS implementation (Gebbert, S.)

GRASS GIS 7  
ZOO 1.0  
ZOOGrassModuleStarter.py



# ZOO PROJECT EXPERIMENTS(3)

## OSM import (ZOOSM) (Delluchi, L.)

PythonOSMApi  
OGR python  
ZOO 1.0  
ZOOSM.py



Provides a new way to import data into OSM using WPS

Could be used for massive imports and updates in OSM

Could be extended to Export from OSM in OGR supported formats

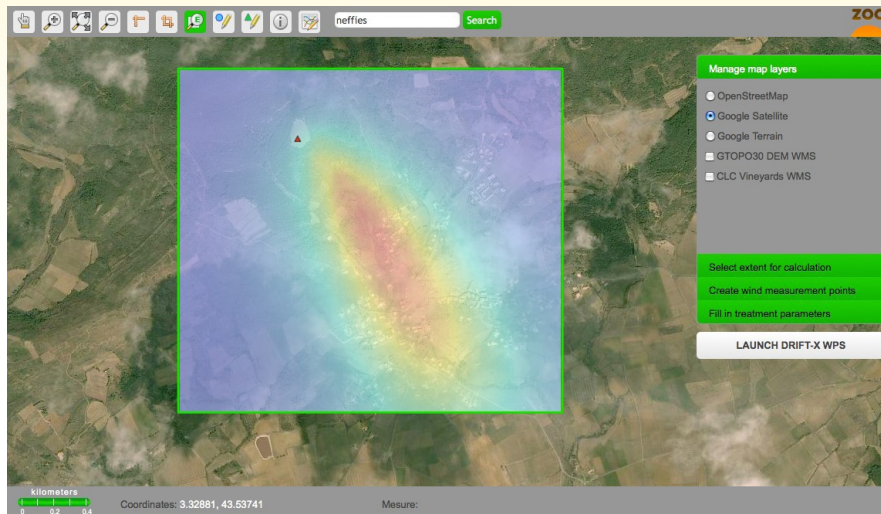
# ZOO PROJECT EXPERIMENTS(4)

## Drift-X WPS (Bozon, N. and Fenoy, G)

ZOO 1.0  
driftx.f  
GdalTranslate  
Services.py

Provides a webmapping application for Pesticide atmospheric dispersion calculation.

WPS for scientific computing



THANKS FOR YOUR TIME

[zoo-discuss@gisws.media.osaka-cu.ac.jp](mailto:zoo-discuss@gisws.media.osaka-cu.ac.jp)

