ZOO WPS the integration with GRASS GIS

Luca Delucchi

Fondazione Edmund Mach - GIS & Remote Sensing Platform

Geoinformatics FCE CTU 2011
20 May 2011, Praha (Česká republika)
What is Web Processing Service (WPS)?
What is Web Processing Service (WPS)?

- OGC standard
What is Web Processing Service (WPS)?

- **OGC standard**
- useful to create web services for invoking **geospatial processing**
WPS: introduction

What is Web Processing Service (WPS)?

- **OGC standard**
- Useful to create web services for invoking **geospatial processing**
- The last version is 1.0.0 (2007)
WPS: requests

There are three requests to work with WPS:

1. GetCapabilities: return description and metadata about the WPS server.
2. DescribeService: return description and metadata about a single service.
3. Execute: run the process and return the output of a service.
WPS: requests

There are three requests to work with WPS:

- **GetCapabilities**: return description and metadata about the WPS server.
WPS: requests

There are three requests to work with WPS

- **GetCapabilities**: return description and metadata about the WPS server

- **DescribeService**: return description and metadata about a single service
WPS: requests

There are three requests to work with WPS:

- **GetCapabilities**: return description and metadata about the WPS server
- **DescribeService**: return description and metadata about a single service
- **Execute**: run the process and return the output of a service
WPS: requests

GetCapabilities

**GetCapabilities** parameters

- `service = WPS`
- `version = 1.0.0`
- `request = GetCapabilities`
WPS: requests

GetCapabilities

GetCapabilities parameters

service = WPS

version = 1.0.0

request = GetCapabilities

http://srvcarto.fmach.it/zoo/?request=getcapabilities&version=1.0.0&service=wps

Luca Delucchi
Geoinformatics FCE CTU, 20 May 2011, Praha
WPS: requests

DescribeProcess

**DescribeProcess** parameters

- service = WPS
- version = 1.0.0
- request = DescribeProcess
- identifier = v_random
WPS: requests

**DescribeProcess**

- service = WPS
- version = 1.0.0
- request = DescribeProcess
- identifier = v_random

http://srvcarto.fmach.it/zoo/?service=wps&version=1.0.0&request=describeprocess&identifier=v_random
WPS: requests

Execute

Execute parameters

service = WPS
version = 1.0.0
request = Execute
identifier = v_random
DataInputs = n=10
ResponseDocument = output
WPS: requests

Execute

**Execute parameters**

- service = WPS
- version = 1.0.0
- request = Execute
- identifier = v_random
- DataInputs = n=10
- ResponseDocument = output

http://srvcarto.fmach.it/zoo/?service=wps&version=1.0.0&request=execute&identifier=v_random&DataInputs=n=10&ResponseDocument=output
WPS: requests

Execute

**Execute** parameters

service = WPS

version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

ResponseDocument =

output@asReference=true

http://srvcarto.fmach.it/zoo/?service=wps&version=1.0.0&request=execute&identifier=v_random&DataInputs=n=10&ResponseDocument=output@asReference=true
WPS: requests

Execute

**Execute** parameters

service = WPS

version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

ResponseDocument = output@asReference=true

http://srvcarto.fmach.it/zoo/?service=wps&version=1.0.0&request=execute&identifier=v_random&DataInputs=n=10&ResponseDocument=output@asReference=true
WPS: requests

**Execute**

**Execute parameters**

service = WPS

version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

RawDataOutput = output
WPS: requests

Execute

**Execute parameters**

service = WPS

version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

RawDataOutput = output

http://srvcarto.fmach.it/zoo/?service=wps&version=1.0.0&request=execute&identifier=v_random&DataInputs=n=10&RawDataOutput=output

Luca Delucchi Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO introduction

Free and Open Source WPS server, released under a MIT/X-11 style license

Luca Delucchi
Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO introduction

Free and Open Source WPS server, released under a MIT/X-11 style license

ZOO is made of three parts:
ZOO introduction

Free and Open Source WPS server, released under a MIT/X-11 style license

ZOO is made of three parts:

Kernel
ZOO introduction

Free and Open Source WPS server, released under a MIT/X-11 style license

ZOO is made of three parts:
- Kernel
- Services
ZOO introduction

Free and Open Source WPS server, released under a MIT/X-11 style license

ZOO is made of three parts:

Kernel Services API
ZOO Kernel

The ZOO Kernel:

- is the heart of the ZOO WPS server
- is written in C language
- manages and chains Web services
- supports several common programming languages
- works with Apache through a cgi file and a conf file
ZOO Kernel

The ZOO Kernel:
- is the heart of the ZOO WPS server
ZOO Kernel

The ZOO Kernel:
- is the heart of the ZOO WPS server
- is written in C language
The ZOO Kernel:
- is the heart of the ZOO WPS server
- is written in C language
- manage and chain Web services

Luca Delucchi
Geoinformatics FCE CTU, 20 May 2011, Praha
The ZOO Kernel:

- is the heart of the ZOO WPS server
- is written in C language
- manage and chain Web services
- support several common programming languages
The ZOO Kernel:
- is the heart of the ZOO WPS server
- is written in C language
- manage and chain Web services
- support several common programming languages
- works with Apache through a cgi file and a conf file
ZOO Services

The ZOO Services:

- GeoExt
- GeoNetwork
- GeoServer
- MapServer
- PyWPS
- GDAL
- OpenSource
- PostGIS
- GeoExt
- OGR
- GDAL
- WPS

ZOO-project
ZOO & GRASS

WPS Services

Introduction
Kernel
Services

Luca Delucchi
Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO Services

The ZOO Services:

- are the several processes that work with the Kernel
The ZOO Services:

- are the several processes that work with the Kernel
- are based on various existing libraries, like GDAL/OGR, but not only geographic (demos with OoO and QR)
The ZOO Services:

- are the several processes that work with the Kernel
- are based on various existing libraries, like GDAL/OGR, but not only geographic (demos with OoO and QR)
- are writable in C/C++, Fortran, Java, Javascript, Perl, PHP, Python
ZOO Services

The ZOO Services are composed by two parts:
The ZOO Services are composed by two parts:

- a configuration file to describe the Service, this is a *zcfg file*. 

*(Nicolo Rigacci 2008)*
ZOO Services

The ZOO Services are composed by two parts:

- a configuration file to describe the Service, this is a *zcfg* file.
- the code you want to use for your Service

"The only limit is your imagination" (Nicolo Rigacci 2008)
ZOO Services

The ZOO Services are composed by two part:
- a configuration file to describe the Service, this is a zcfg file
- the code you want use for your Service

“The only limit is your imagination” (Nicolo Rigacci 2008)
The ZOO API is:

- GeoExt
- PostGIS
- GeoServer
- MapServer
- PyWPS
- QGIS
- GDAL
- mapnik
- GeoNetwork
- Geosasource
- zoonet
The ZOO API is:

- Javascript library designed to make the WPS Process creation and chaining easier and call it like other services
ZOO API

The ZOO API is:

- Javascript library designed to make the WPS Process creation and chaining easier and call it like other services

- server-side using the Mozilla foundation JavaScript engine, SpiderMonkey
ZOO API

The ZOO API is:

- Javascript library designed to make the WPS Process creation and chaining easier and call it like other services
- server-side using the Mozilla foundation JavaScript engine, SpiderMonkey
- Proj4js adaptation for server-side reprojection
ZOO & GRASS: dependences

To work with ZOO and GRASS there are some dependences to satisfy:
ZOO & GRASS: dependences

To work with ZOO and GRASS there are some dependences to satisfy:

- ZOO (better from svn)
ZOO & GRASS: dependences

To work with ZOO and GRASS there are some dependences to satisfy:

- ZOO (better from svn)
- GRASS GIS version 7
ZOO & GRASS: dependences

To work with ZOO and GRASS there are some dependences to satisfy:

- ZOO (better from svn)
- GRASS GIS version 7
- wps-grass-bridge
ZOO & GRASS: dependences

To work with ZOO and GRASS there are some dependences to satisfy:

- ZOO (better from svn)
- GRASS GIS version 7
- wps-grass-bridge
- pyXB 1.1.2
ZOO & GRASS: installation

To install:

- Compile, install and test ZOO, GRASS and pyXB.
- Move into the wps-grass-bridge directory and:
  - Modify GlobalGrassSettings.py with your settings.
  - Copy GlobalGrassSettings.py, ZOOGrassModuleStarter.py and gms folder in the path where Apache CGI scripts run and where you put zoo loader.cgi.
- In the path where you found some modules already working, copy the processes that you want to use in the path where you copied the other files.

Now you are ready to test the processes!
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB

Luca Delucchi Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
  - modify GlobalGrassSettings.py with your setting

Now you are ready to test the processes!
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
  - modify GlobalGrassSettings.py with your setting
  - copy GlobalGrassSettings.py, ZOOGrassModuleStarter.py and gms folder in the path where Apache cgi scripts run and where you put zoo_loader.cgi

Now you are ready to test the processes!

Luca Delucchi Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
  - modify GlobalGrassSettings.py with your setting
  - copy GlobalGrassSettings.py, ZOOGrassModuleStarter.py and gms folder in the path where Apache cgi scripts run and where you put zoo_loader.cgi
  - in zoo_services folder you find some modules already working, copy the processes that you want use in the path where you copied the other files

Luca Delucchi Geoinformatics FCE CTU, 20 May 2011, Praha
ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
  - modify GlobalGrassSettings.py with your setting
  - copy GlobalGrassSettings.py, ZOOGrassModuleStarter.py and gms folder in the path where Apache cgi scripts run and where you put zoo_loader.cgi
  - in zoo_services folder you find some modules already working, copy the processes that you want use in the path where you copied the other files

Now you are ready to test the processes!
ZOO & GRASS: some suggestion

- activate rewrite.load modules of Apache
ZOO & GRASS: some suggestion

- activate rewrite.load modules of Apache
- modify /etc/apache2/sites-available/default files like

```xml
<Directory /var/www/zoo>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

create a directory (for example zoo) in the Apache folder and put inside a .htaccess file like this

```plaintext
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*.*) /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*.*) / /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*.*) /cgi-bin/zoo_loader.cgi [L,QSA]
```

create a directory for temporary files (recommend tmp) in the Apache folder, not inside the folder create before
ZOO & GRASS: some suggestion

- activate rewrite.load modules of Apache
- modify /etc/apache2/sites-available/default files like

```html
<Directory /var/www/zoo>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow, deny
    allow from all
</Directory>
```

- create a directory (for example zoo) in the Apache folder and put inside a .htaccess file like this

```text
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*)(.*)/(.*) /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*)(.*)/ /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*)(.*)/cgi-bin/zoo_loader.cgi?metapath= [L,QSA]
RewriteRule (.*)/cgi-bin/zoo_loader.cgi [L,QSA]
```

create a directory (for example tmp) in the Apache folder, not inside the folder create before
ZOO & GRASS: some suggestion

- activate rewrite.load modules of Apache
- modify /etc/apache2/sites-available/default files like

```html
<Directory /var/www/zoo>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride All
  Order allow,deny
  allow from all
</Directory>
```

- create a directory (for example zoo) in the Apache folder and put inside a .htaccess file like this

```apache
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*)(.*)/.* /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*)(.*) /cgi-bin/zoo_loader.cgi?metapath=$1 [L,QSA]
RewriteRule (.*)(.*) /cgi-bin/zoo_loader.cgi?metapath= [L,QSA]
RewriteRule (.* )/cgi-bin/zoo_loader.cgi [L,QSA]
```

- create a directory for temporary files (recommend tmp) in the Apache folder, not inside the folder create before
ZOO & GRASS: testing

To test you can try:

- copy v.report.* and use the link provided before changing the server (if you are on your PC, maybe it's localhost)
- copy other services and use the requests provided before changing the right parameters (server, identifier name and execute parameters)
- use the WPS client plugin of QGIS
ZOO & GRASS: testing

To test you can try:

- to copy `v_report.*` and use the link show before changing the server (if you are on your pc maybe it’s localhost)
ZOO & GRASS: testing

To test you can try:

- to copy v_report.* and use the link show before changing the server (if you are on your pc maybe it’s localhost)

- to copy other services and use the requests show before changing the right parameters (server, identifier name and execute parameters)
ZOO & GRASS: testing

To test you can try:

- to copy v_report.* and use the link show before changing the server (if you are on your pc maybe it’s localhost)

- to copy other services and use the requests show before changing the right parameters (server, identifier name and execute parameters)

- using the WPS client plugin of QGIS
ZOO & GRASS: link

- [http://www.zoo-project.org/site/ZooDocumentation](http://www.zoo-project.org/site/ZooDocumentation)
- [http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client](http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client)
ZOO & GRASS: link

- http://www.zoo-project.org/site/ZooDocumentation
- http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client
ZOO & GRASS: link

- [http://www.zoo-project.org/site/ZooDocumentation](http://www.zoo-project.org/site/ZooDocumentation)
- [http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client](http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client)
- [http://grass.osgeo.org/wiki/WPS](http://grass.osgeo.org/wiki/WPS)
WPS: final considerations

- WPS is very useful for the web processes...
WPS: final considerations

- WPS is very useful for the web processes...
- ...is being increasingly used...
WPS: final considerations

- WPS is very useful for the web processes...
- ...is being increasingly used...
- ...but maybe it’s not the best solution for interface library with desktop software
Thanks to

Nicolas Bozon, Gerald Fenoy, Markus Neteler, Soeren Gebbert, Venkatesh Raghavan...
Thanks to

Nicolas Bozon, Gerald Fenoy, Markus Neteler, Soeren Gebbert, Venkatesh Raghavan...

...and all GRASS and ZOO community

Luca Delucchi
Geoinformatics FCE CTU, 20 May 2011, Praha
This presentation is released under cc-by-sa license

You are free:

- to Share — to copy, distribute and transmit the work

- to Remix — to adapt the work

Under the following conditions:

- Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

- Noncommercial — You may not use this work for commercial purposes.

- Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.