

ZOO WPS the integration with GRASS GIS

Luca Delucchi

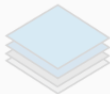
Fondazione Edmund Mach - GIS & Remote Sensing Platform

Geoinformatics FCE CTU 2011

20 May 2011, Praha (Česká republika)



WPS: introduction



What is Web Processing Service (WPS)?



PyWPS



WPS: introduction



What is Web Processing Service (WPS)?

- OGC standard



WPS: introduction

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)?

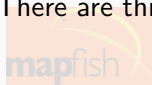
- **OGC standard**
- useful to create web services for invoking **geospatial processing**
- the last version is 1.0.0 (2007)



WPS: requests



There are three requests to work with WPS



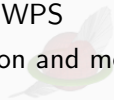
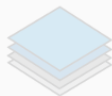
PyWPS



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



PyWPS



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>

```

-ows:Capabilities xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities.xsd"
-ows:ServiceIdentification>
  <ows:Title>WPS Server PQS FEM</ows:Title>
  <ows:Abstract>WPS Server of GIS</ows:Abstract>
  <ows:Keywords>
    <ows:Keyword>WPS</ows:Keyword>
    <ows:Keyword>GIS</ows:Keyword>
    <ows:Keyword>FEM</ows:Keyword>
    <ows:Keyword>IASMA</ows:Keyword>
    <ows:Keyword>OGR</ows:Keyword>
  </ows:Keywords>
  <ows:ServiceType>WPS</ows:ServiceType>
  <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  <ows:Fee>None</ows:Fee>
  <ows:AccessConstraints>none</ows:AccessConstraints>
-ows:ServiceIdentification>
-ows:ServiceProvider>
  <ows:ProviderName>WPSFEM</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://srvcarto.fmach.it"/>
-ows:ServiceContact>
  <ows:IndividualName>Luca Delucchi</ows:IndividualName>
  <ows:PositionName>Developer</ows:PositionName>
-ows:ContactInfo>
  +ows:Phone</ows:Phone>
  <ows:Address>
    <ows:City>San Michele all'Adige</ows:City>
    <ows:AdministrativeArea>False</ows:AdministrativeArea>
    <ows:PostalCode>38010</ows:PostalCode>
    <ows:Country>Italy</ows:Country>
    <ows:ElectronicMailAddress>luca.delucchi@iasma.it</ows:ElectronicMailAddress>
  </ows:Address>
-ows:ContactInfo>
-ows:ServiceContact>
-ows:ServiceProvider>
-ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    +ows:DCP>
    +ows:HTTP></ows:HTTP>
  </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeProcess">
    +ows:DCP>
    +ows:HTTP></ows:HTTP>
  </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Execute">
    +ows:DCP>
    +ows:HTTP></ows:HTTP>
  </ows:DCP>
  </ows:Operation>
-ows:ProcessOfferings>
-ows:Process srs:processVersion="1.0">
  <ows:Identifier>v_random</ows:Identifier>
  <ows:Title>Generates randomly 2D/3D vector points map</ows:Title>
  <ows:Abstract>
    http://grass.osgeo.org/grass70/manuals/html70_user/v_random.html
  </ows:Abstract>

```



WPS: requests

DescribeProcess

DescribeProcess parameters

service = WPS

version = 1.0.0

request = DescribeProcess

identifier = v_random



PyWPS



WPS: requests

DescribeProcess

DescribeProcess parameters

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

```
<?xml version="1.0" encoding="UTF-8" style="display:none" ?>
<DescribeProcess xmlns="http://www.opengis.net/wps/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wps.xsd" >
  <ProcessIdentifier>v_random</ProcessIdentifier>
  <ProcessTitle>v_random</ProcessTitle>
  <ProcessVersion>1.0.0</ProcessVersion>
  <ProcessInputs>
    <Input>
      <Name>input</Name>
      <Title>input</Title>
      <Description>input</Description>
      <SupportedMediaTypes>
        <SupportedMediaTypes>application/vnd.google-earth.kml+xml</SupportedMediaTypes>
      </SupportedMediaTypes>
      <DefaultValue></DefaultValue>
    </Input>
  </ProcessInputs>
  <ProcessOutputs>
    <Output>
      <Name>output</Name>
      <Title>output</Title>
      <Description>output</Description>
      <SupportedMediaTypes>
        <SupportedMediaTypes>application/vnd.google-earth.kml+xml</SupportedMediaTypes>
      </SupportedMediaTypes>
      <DefaultValue></DefaultValue>
    </Output>
  </ProcessOutputs>
  <ProcessCapabilities>
    <SupportedFormats>
      <SupportedFormat>
        <Name>application/vnd.google-earth.kml+xml</Name>
        <Title>application/vnd.google-earth.kml+xml</Title>
        <Description>application/vnd.google-earth.kml+xml</Description>
      </SupportedFormat>
    </SupportedFormats>
  </ProcessCapabilities>
  </DescribeProcess>
```



WPS: requests

Execute



Execute parameters

service = WPS

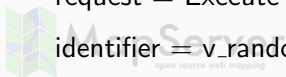
version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

ResponseDocument =
output



PyWPS



WPS: requests

Execute



Execute parameters

service = WPS

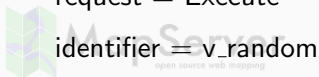
version = 1.0.0

request = Execute

identifier = v_random

DataInputs = n=10

ResponseDocument =
output@asReference=true



PyWPS



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

```

<?xml version="1.0" encoding="UTF-8" ?>
<wps:ExecuteResponse xmlns:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsExecuteResponse" service="WPS" version="1.0.0" xsi:lang="en-US" ?>
  <wps:Process wps:processVersion="1.0" ?>
    <wps:Identifier>v_random</wps:Identifier>
    <ows:Title>Generate randomly 1000 vector points map</ows:Title>
    <ows:Abstract>
      http://www.opengis.org/present/Osmansala/html/0_user/v_random.html
    </ows:Abstract>
    </wps:Process>
    <wps:Status creationTime="2011-05-27T09:46:08Z" ?>
      <wps:ProcessSucceeded>Service "v_random" run successfully.</wps:ProcessSucceeded>
    </wps:Status>
    <wps:ProcessOutput>
      <ows:Identifier>output</ows:Identifier>
      <ows:Title>Name for output vector points</ows:Title>
      <ows:Reference href="http://www.fmach.it/ogc/arcgis_random_output_110460.html" mimeType="text/html" encoding="UTF-8" schema="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd" ?>
    </ows:Output>
    </wps:ProcessOutput>
  </wps:ExecuteResponse>

```



WPS: requests

Execute

Execute parameters

service = WPS

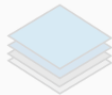
version = 1.0.0

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
```

```
<-ogr:FeatureCollection xsi:schemaLocation="http://ogr.maptools.org/output_0.xsd">  
  <-gml:boundedBy>  
    <-gml:Box>  
      <-gml:coords>  
        <gml:X>6.658833683286103</gml:X>  
        <gml:Y>2.292727959478613</gml:Y>  
      </gml:coords>  
    <-gml:coords>  
      <gml:X>88.66862795694718</gml:X>  
      <gml:Y>96.36145955713535</gml:Y>  
    </gml:coords>  
  </gml:boundedBy>  
  <-gml:featureMember>  
    <-ogr:output fid="F0">  
      <-ogr:geometryProperty>  
        <-gml:Point>  
          <gml:coordinates>-24.403484828968042,84.640226562561032,0</gml:coordinates>  
        </gml:Point>  
      </ogr:geometryProperty>  
    <-ogr:cat>1</ogr:cat>  
  </ogr:output>  
  </gml:featureMember>  
  <-gml:featureMember>  
    <-ogr:output fid="F1">  
      <-ogr:geometryProperty>  
        <-gml:Point>  
          <gml:coordinates>86.173842840908947,72.929876983598746,0</gml:coordinates>  
        </gml:Point>  
      </ogr:geometryProperty>  
    <-ogr:cat>2</ogr:cat>  
  </ogr:output>  
  </gml:featureMember>  
  <-gml:featureMember>  
    <-ogr:output fid="F2">  
      <-ogr:geometryProperty>  
        <-gml:Point>  
          <gml:coordinates>53.41782553746284,90.970139666912203,0</gml:coordinates>  
        </gml:Point>  
      </ogr:geometryProperty>  
    <-ogr:cat>3</ogr:cat>  
  </ogr:output>  
  </gml:featureMember>  
  <-gml:featureMember>  
    <-ogr:output fid="F3">  
      <-ogr:geometryProperty>  
        <-gml:Point>  
          <gml:coordinates>89.669627956947176,70.144444736719336,0</gml:coordinates>  
        </gml:Point>  
      </ogr:geometryProperty>  
    <-ogr:cat>4</ogr:cat>  
  </ogr:output>  
  </gml:featureMember>
```

ZOO introduction

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



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



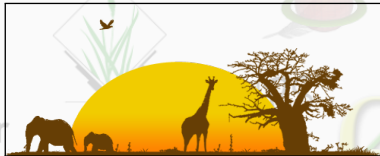
ZOO is made of three parts:

Kernel



ZOO introduction

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



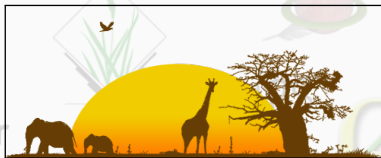
ZOO is made of three parts:

Kernel Services



ZOO introduction

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

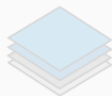


ZOO is made of three parts:

Kernel Services API



ZOO Kernel



The ZOO Kernel:



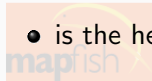
PyWPS



ZOO Kernel

The ZOO Kernel:

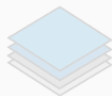
- is the heart of the ZOO WPS server



ZOO Kernel

The ZOO Kernel:

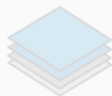
- is the heart of the ZOO WPS server
- is written in C language



ZOO Kernel

The ZOO Kernel:

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



GeoNetwork
OpenSource



GeoServer

MapServer
open source web mapping

PyWPS



mapnik



ZOO Kernel

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



ZOO Kernel

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:



PyWPS



ZOO Services

The ZOO Services:

- are the several processes that work with the Kernel



ZOO Services

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)



ZOO Services

The ZOO Services:

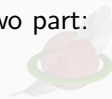
- are the several processes that work with the Kernel
- are based on various existing libraries, like GDAL/OGR, but not only geographic (demos with OoO and QR)
- are writable in C/C++, Fortran, Java, Javascript, Perl, PHP, Python



ZOO Services



The ZOO Services are composed by two part:



GeoServer



PyWPS



ZOO Services

The ZOO Services are composed by two part:

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



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



GeoNetwork
OpenSource



GeoServer

MapServer
open source web mapping

PyWPS



mapnik



ZOO Services

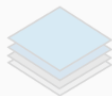
The ZOO Services are composed by two part:

- 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)



ZOO API



The ZOO API is:



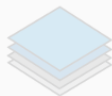
PyWPS



ZOO API

The ZOO API is:

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



GeoNetwork
OpenSource



GeoServer

MapServer
open source web mapping

PyWPS



mapnik



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

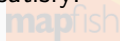
The ZOO API is:

- 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:



PyWPS



ZOO & GRASS: installation

To install:

- compile, install and test ZOO, GRASS and pyXB



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



PyWPS



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



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



ZOO & GRASS: installation

To install:

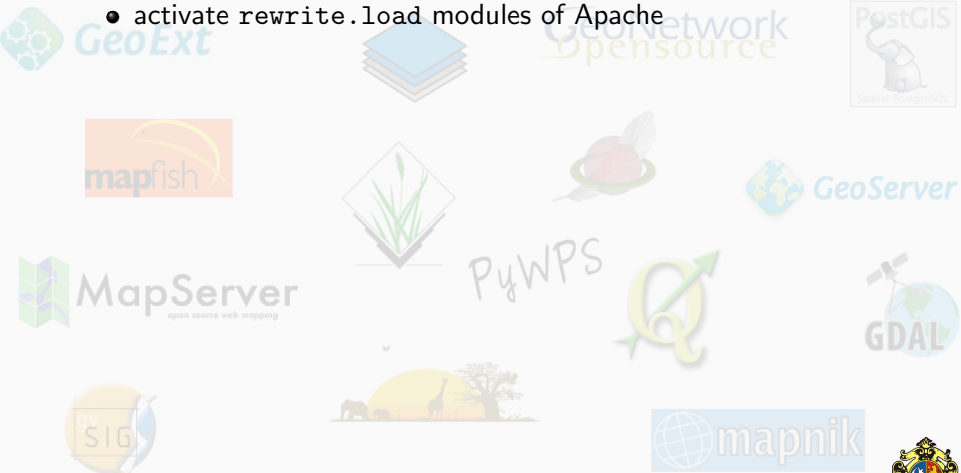
- compile, install and test ZOO, GRASS and pyXB
- move into wps-grass-bridge directory and:
 - modify GlobalGrassSettings.py with your setting
 - copy GlobalGrassSettings.py, ZOOGrassModuleStarter.py and gms folder in the path where Apache cgi scripts run and where you put zoo_loader.cgi
 - in zoo_services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

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



GeoServer



PyWPS



ZOO & GRASS: some suggestion

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

```
<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

```
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]
```



ZOO & GRASS: some suggestion

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

```
<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

```
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



WPS
ZOO-project
ZOO & GRASS
edroiti

Install
Testing
Interesting link

ZOO & GRASS: testing



To test you can try:



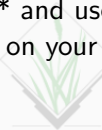
PyWPS



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



WPS
ZOO-project
ZOO & GRASS
edroiti

Install
Testing
Interesting link

ZOO & GRASS: link



- <http://www.zoo-project.org/site/ZooDocumentation>



GeoServer



PyWPS



ZOO & GRASS: link

- <http://www.zoo-project.org/site/ZooDocumentation>
- http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client



PyWPS



ZOO & GRASS: link

- <http://www.zoo-project.org/site/ZooDocumentation>
- http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client
- http://code.google.com/p/wps-grass-bridge/wiki/ZOO_WPS_Integration



ZOO & GRASS: link

- <http://www.zoo-project.org/site/ZooDocumentation>
- http://zoo-project.org/trac/wiki/ZooWebSite/QGIS_WPS_Client
- http://code.google.com/p/wps-grass-bridge/wiki/ZOO_WPS_Integration
- <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...



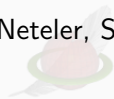
PyWPS



Thanks to



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



MapServer ... and all **GRASS** and **ZOO** community

PuWPS



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.

