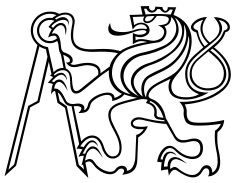
Czech technical university in Prague
Faculty of Civil engineering

Master thesis

Prague 2018 Bc. Adam Laža

Czech technical university in Prague
Faculty of Civil engineering
Study programme Geodesy and cartography
Geomatics



Master thesis

Process isolation in PyWPS framework

Izolace procesů ve frameworku PyWPS

Supervisor: Ing. Martin Landa, Ph.D.
Department of geomatics

Prague 2018

Bc. Adam Laža

Abstract

Abstract

Keywords:
Keywords

Abstrakt

Abstrakt

Klíčová slova:
Klicova slova

Declaration of authorship

Prohlaseni

In Prague .....

.....

(author sign)

Acknowledgement

Podekovani

Contents

Introduction

Mame hromadu dat, ktere je potreba zpracovat. Hodne to ulehci, kdyz to budem moct nejak standardizovat a pak pouzivat na cloudu.

https://pdfs.semanticscholar.org/bb17/7b12791d5ea58811955555be2d48226fd5ae.pdf

Uvod

Part I

Technological background

1 Web Processing Service

1.1 History

First mention of the Web Processing Service was in October 2004. Back then it was named Geoprocessing Service [1]. The specification was first implemented as a prototype in 2004 by Agriculture and Agri-Food Canada (AAFC). In its further development during a Geoprocessing Services Interoperability Experiment [2] the name was changed to "Web Processing Service" to avoid the acronym GPS, since this would have caused confusion with the conventional use of this acronym for Global Positioning System [4]. The first version of WPS was released in September 2005 [3]. The experiment demonstrated that various clients could easily access and bind to services which were set up according the WPS Implementation specification.

Currently two major versions of WPS Standard exist. The WPS version 1.0.0 is currently used mostly. If not explicitly said this thesis is dedicated to the version 1.0.0. The WPS version 2.0.0 was released in 2015 [5].

1.2 Web Processing Service

The OpenGIS® Web Processing Service (WPS) Interface Standard defines a standardized interface that facilitates the publishing of geospatil processes. Also provides rules how to standardize requests and responses for geospatial processing services.

Process means any operation on spatial data from simple ones as maps overlay or buffering to highly complex as complicated global models. Any kind of GIS funcionality can be offered to clients across network with correctly configured WPS.

Publishing means creating human-readable metadata that allow user to discover and use service as well as making available machine-readable binding information.

Data can be both vector or raster data and can be delivered across the network or be available at the server.

The interface does not specify any specific processes that can be implemented by a WPS nor any specific data inputs or outputs. instead it specifies a generic mechanisms to describe any geospatial process and data required and produced by

the process. The interface does not only provide mechanisms for calculation but also to identify required data, initiate the calculation and manage output data so clients can access it.

Web Processing Service as one of the OGC web services scpecifies three types of requests which can be requested by a client and performed by a WPS server. The implementation of these three requests is mandatory by by all servers:

- GetCapabilities
- DescribeProcess
- Execute

GetCapabilities - The request returns to client a Capabilities document that describes the abilities of the specific server implementation. It also returns the name and abstract of each of the processes that can be run on a WPS instance.

DescribeProcess - The request returns details about the processes offered by a WPS instance. Describes required inputs and produces outputs and their allowable formats.

Execute - The request allows a client to run a specified process with provided parameters and returns produced outputs.

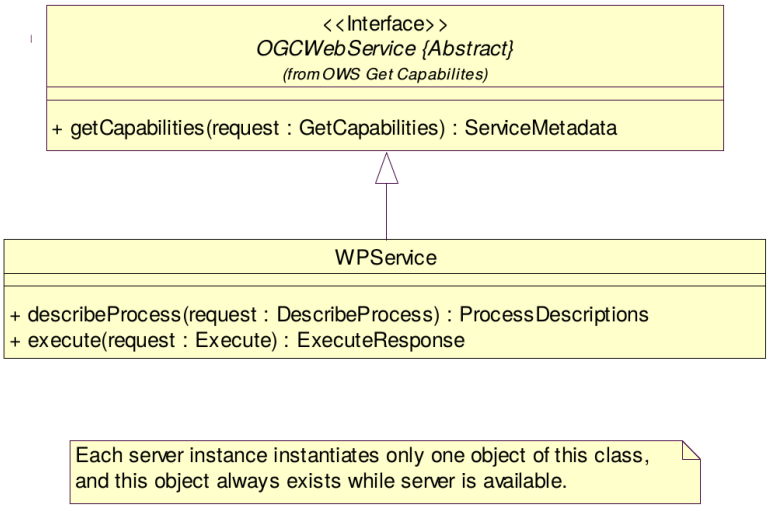


Figure 1: WPS interface UML description, source: [4]

These operations are very similar to other OGC Web Services such as WMS, WFS, and WCS. Common interface aspects are defined in the OpenGIS ® Web Services Common Implementation Specification [6]. As seen at class diagram at Fig. 1 the WPS interface class inherits the GetCapabilities operation from OGCWebService interface class. The operations Execute and DescribeProcess are specific for the WPS. The WPS operations are based on GET and POST requests.

Operation	Request encoding	
	Mandatory	Optional
GetCapabilities	KVP	XML
DescribeProcess	KVP	XML
Execute	XML	KVP

Table 1: Operations request encoding

The GetCapabilities and DescribeProcess shall use HTTP GET with KVP encoding and Execute operation shall use HTTP POST with XML encoding. Summarized in Table 1.

1.2.1 GetCapabilities

The GetCapabilities operation is mandatory. The operation allows clients to retrieve capabilities document (metadata) from a server. The response XML document contains service metadata about server and all implemented processes description.

AcceptVersion vs version, AcceptFormats vs format

GetCapabilities request

Request parameters

- *service* - Mandatory parameter, WPS is only possible value.
- *request* - Mandatory parameter, GetCapabilities is only possible value.

Name	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier text
request=GetCapabilities	Mandatory	Operation name text
AcceptVersion=1.0.0	Optional	Specification version
Sections=All	Optional	Comma-separated unordered list of sections
updateSequence=XXX	Optional	Service metadata document version
AcceptFormats=text/xml	Optional	Comma-separated prioritized sequence of response formats

Table 2: GetCapabilities operation request URL parameters, source: [6]

- *version* - Optional parameter, version number. Three non-negative integers separated by decimal point. Servers and their clients should support at least one defined version.
- *sections* - Optional parameter that contains a list of section names. Possible values are: *ServiceIdentification, ServiceProvider, OperationsMetadata, Contents, All.*
- *updateSequence* - Optional parameter for maintaining the consistency of a client cache of the contents of a service metadata document. The parameter value can be an integer, a timestamp, or any other number or string.
- *updateSequence* - Optional parameter for maintaining the consistency of a client cache of the contents of a service metadata document. The parameter value can be an integer, a timestamp, or any other number or string.
- *format* - Optional parameter that defines response format.

The GetCapabilities operation can be requested with parameters from table 2. A corresponding request URL looks like: `http://localhost:5000/wps?service=WPS&request=GetCapabilities&AcceptVersion=1.0.0&Section=ServiceIdentification,OperationsMetadata&updateSequence=XXX&AcceptFormats=text/xml`

GetCapabilities response

Normal response When GetCapabilities operation requested a client retrieve service metadata document that contains sections specified in *sections* parameter. If the parameter value is *All* or is not specified all sections retrieved.

- *ServiceIdentification* - Server metadata.
- *ServiceProvider* - Server operating organization metadata.
- *OperationsMetadata* - Metadata about operations implemented by the WPS server, including URLs to request them.
- *ProcessOfferings* - List of processes with name and brief description implemented by the WPS server.

In addition to sections each GetCapabilities response should contains:

- *version* - Specification version for GetCapabilities operation.
- *updateSequence* - Server metadata document version, value is increased whenever any change is made in complete service metadata document.

GetCapabilities exceptions In case that WPS server encounters an error a client retrieve an exception report message with one of there exception code:

- *MissingParameterValue* - GetCapabilities request does not contain a required parameter value.
- *InvalidParameterValue* - GetCapabilities request contains an invalid parameter value.
- *VersionNegotiation* - Any version from AcceptVersions parameter list does not match any version supported by the WPS server.
- *InvalidUpdateSequence* - Value of updateSequence parameter is greater than current value of service metadata updateSequence number.
- *NoApplicableCode* - Other exceptions.

1.2.2 DescribeProcess

The DescribeProcess operation is mandatory. The operation allows clients to retrieve a detailed description about one or more processes implemented by a WPS server. The detailed information describe both required inputs and produced outputs and allowed format.

DescribeProcess request

Request parameters

- *service* - Mandatory parameter, WPS is only possible value.
- *request* - Mandatory parameter, DescribeProcess is only possible value.
- *version* - Mandatory parameter, version number. Three non-negative integers separated by decimal point. Servers and their clients should support at least one defined version.
- *Identifier* - Optional parameter, list of process names separated by comma. Another possible value is *all*.

Name	Optionality	Definition and format
service=WPS	Mandatory	Service type identifier text
request=DescribeProcess	Mandatory	Operation name text
version=1.0.0	Mandatory	WPS specification version
Identifier=buffer	Optional	List of one or more process identifiers, separated by commas

Table 3: DescribeProcess operation request URL parameters, source: [6]

The DescribeProcess operation can be requested with parameters from table 3. A corresponding request URL looks like: `http://localhost:5000/wps?request=DescribeProcess&service=WPS&identifier=all&version=1.0.0`

DescribeProcess response

Normal response Normal response to DescribeProcess request contains or more process descriptions for requested process identifiers in *ProcessDescriptions* structure. Each process description contains detailed information about process in *ProcessDescription* including process inputs and outputs description. Number of inputs or outputs is not limited. Three types of input or outpus exist:

Doplnit popisy dat

- *LiteralData* -
- *ComplexData* -
- *BoundingBoxData* -

Name	Optionality	Definition and format
ProcessDescription	Mandatory	Full description of process including inputs/outputs
service=WPS	Mandatory	Service type identifier text
version=1.0.0	Mandatory	Operation specification version
lang	Mandatory	Language identifier

Table 4: Parts of ProcessDescriptions data structure, source: [4]

DescribeProcess exceptions In case that WPS server encounters an error a client retrieve an exception report message with one of there exception code:

- *MissingParameterValue* - GetCapabilities request does not contain a required parameter value.
- *InvalidParameterValue* - GetCapabilities request contains an invalid parameter value.
- *NoApplicableCode* - Other exceptions.

Name	Optionality	Definition and format
Identifier	Mandatory	Process identigier
Title	Mandatory	Process title
Abstract	Optional	Brief description
Metadata	Optional	Reference to more metadata about this process
Profile	Optional	Profile to which the WPS process complies
processVersion	Mandatory	Release version of process
WSDL	Optional	Location of a WSDL document that describes this process
DataInputs	Optional	List of the required and optional inputs
ProcessOutputs	Mandatory	List of the required and optional outputs
storeSupported	Optional	Complex data outputs can be stored by WPS server
statusSupported	Optional	Execute response can be returned quickly with status information

Table 5: Parts of ProcessDescription data structure, source: [4]

1.2.3 Execute

The Execute operation is mandatory. The operation allows clients to run a specified process implemented by a server. Inputs can be included directly in the request body or be referenced as web accesible resource. The outputs are returned in XML response document, either directly embedded within the response document or stored as resource accesible by returned URL.

Ussualy the response document is returned right after the process execution is completed. However it is possible to get response document right after sending request. In this case, returned response document contains a URL link from which the final response document can be retrieved after completed process execution. Client can request execution status update to find out the amount of processing

remaining if the execution is not completed. Shown at Fig. 2.

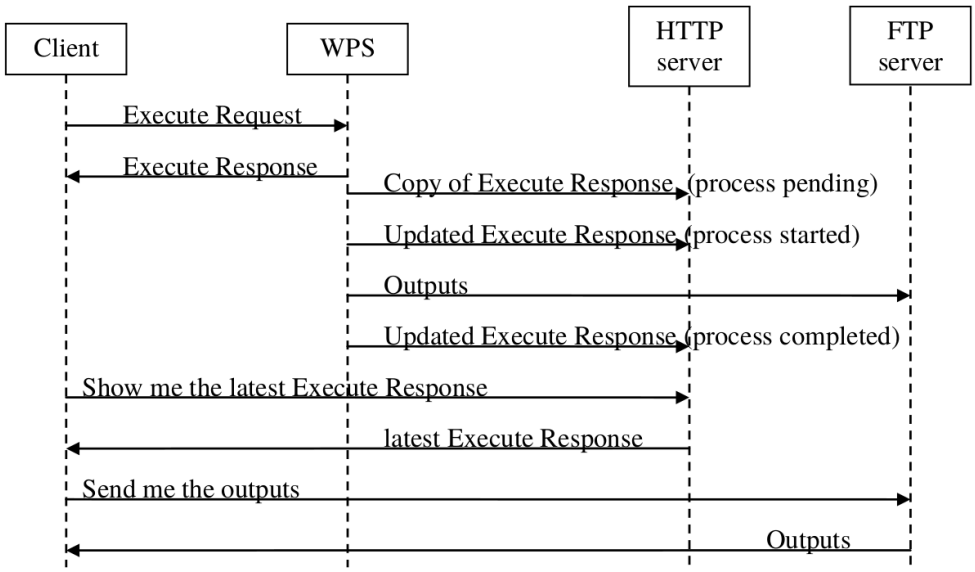


Figure 2: Activity diagram when client requests storage of results, source: [4]

Name	Optionality	Definition and format
service	Mandatory	Service type identifier text
request	Mandatory	Operation name text
version	Mandatory	WPS specification version
Identifier	Mandatory	Process identifier
DataInputs	Optional	List of inputs provided to this process execution
ResponseForm	Optional	Response type definition
language	Optional	Language identifier

Table 6: Parts of Execute operation request, source: [4]

Execute request

Execute response Ussualy the Execute operation response document is an XML document. Only exception is in case when a response form of *RawDataOutput* is requested, execution is successful and only one complex output is created, then directly the produced complex output is returned.

In ussual case response to Execute opetation is an ExecuteResponse XML document. The contents depend on ResponseForm request elements.

Name	Optionality	Definition and format
service	Mandatory	Service type identifier text
version	Mandatory	WPS specification version
language	Mandatory	Language identifier
statusLocation	Optional	Reference to location where current ExecuteResponse document is stored
serviceInstance	Mandatory	Reference to location where current ExecuteResponse document is stored
Process	Mandatory	Process description
Status	Mandatory	Execution status of the process
DataInputs	Optional	List of inputs provided to this process execution
OutputDefinitions	Optional	List of definitions of outputs desired from executing this process
ProcessOutputs	Optional	List of values of outputs from process execution

Table 7: Parts of ExecuteResponse data structure, source: [4]

1.3 WPS implementations

The OGS WPS is just interface standard that provides rules for standardizing requests and response. It also defines how clients can request the execution of defined processes and how the outputs are handled. There are several open-source projects that implement this standard across the platforms or programming languages.

- *PyWPS* - Python implementation. This thesis is dedicated to this implementation.
- *Zoo Project* - WPS implementation written in C, Python and JavaScript.
- *WPS.NET* - WPS implementation on .NET platform.

- *52°North WPS* - Java implementation.
- *deegree* - Java implementation of many OGC standards including WPS.
- *WPSint* - Java Spring implementation.

2 PyWPS

PyWPS is server-side implementation of the OGC WPS Standard. It is written in Python. It is currently supporting WPS 1.0.0.

3 Docker

Docker is a Linux container technology that allows package and ship applications and everything it needs to execute into a standard format, and run them on any infractructure.

Docker container vs. Virtual machine Both virtual machines and docker containers are two ways how to deploy multiple, isolated applications on a single platform. They both offer a way to isolate an application and its dependencies into a self-contained unit that can run anywhere. They both offer some kind of virtualization. They differ in architecture, see Fig. ?? and ??.

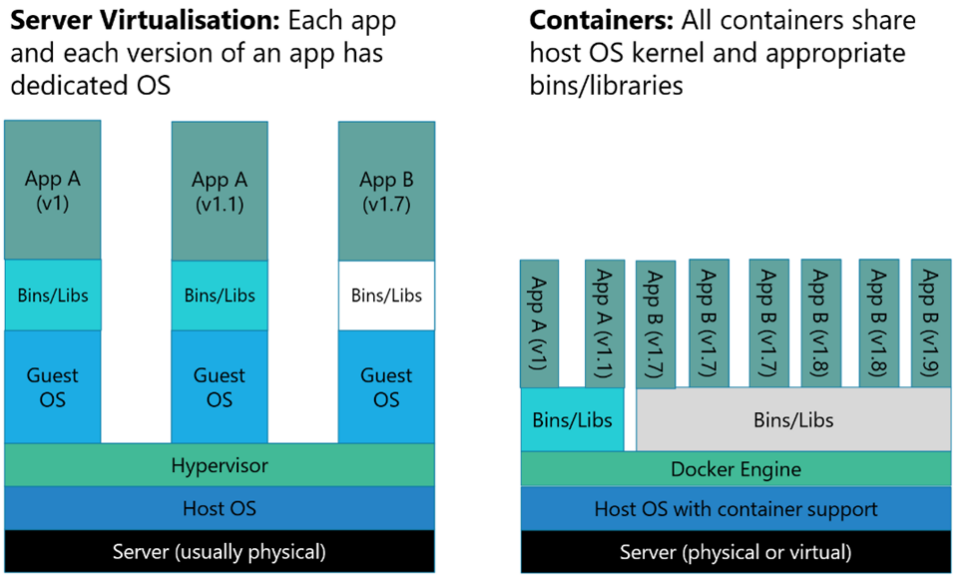


Figure 3: Container vs VM differences, source `https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/09/23`

Let's start with virtual machine Fig. ?? and its layers description from bottom up:

- *Infrastructure* - It can be a PC, developer's laptop, a physical server in data-center but as well a virtual private server in the cloud as Microsoft Azure or Amazon EC2.
- *Host OS*

Part II

Practical part

Seznam použitých zkratek

KVP	Key Value Pair
OGC	Open Geospatial Consortium
URL	Uniform Resource Locator
VM	Virtual machine
WPS	Web Processing Service
WMS	Web Map Service
WFS	Web Feature Service
WCS	Web Coverage Service
XML	eXtensible Markup Language

References

[1] Mark Reichardt *OGC Newsletter - October 2004, OGC document number 04-043* [online]. URL: <http://www.opengeospatial.org/pressroom/newsletters/200410>

[2] Sam Bacharach *OGC announces Web Processing Services Interoperability Experiment* [online]. URL: <http://www.opengeospatial.org/pressroom/pressreleases/414>

[3] Open Geospatial Consortium Inc. *OpenGIS ® Web Processing Service, OGC document number 05-007r4, ver. 0.4.0* [online]. URL: <https://portal.opengeospatial.org/files/?artifact_id=13149&version=1&format=doc>

[4] http://www.opengeospatial.org/pressroom/newsletters/200410

[5] Open Geospatial Consortium *OGC® WPS 2.0 Interface Standard Corrigendum 1, OGC document number 06-121r3* [online]. URL: <https://portal.opengeospatial.org/files/?artifact_id=13149&version=1&format=doc>

[6] Open Geospatial Consortium Inc. *OGC Web Services Common Specification, OGC document number 14-065* [online]. URL: <https://portal.opengeospatial.org/files/?artifact_id=20040>

[7] Docker *Docker documentation* [online]. URL: <https://docs.docker.com/>