Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

Volume 15 *Seoul, South Korea* Article 30

2015

Modifications To Web Processing Service Standard For Client-Side Geoprocessing

Evgeny Panidi
Department of Cartography and Geoinformatics, Saint-Petersburg State University

Eduard Kazakov
Department of Cartography and Geoinformatics, Saint-Petersburg State University

Anton Terekhov
Department of Cartography and Geoinformatics, Saint-Petersburg State University

Evgeny Kapralov
Department of Cartography and Geoinformatics, Saint-Petersburg State University

AT8-04

MODIFICATIONS TO WEB PROCESSING SERVICE STANDARD FOR CLIENT-SIDE GEOPROCESSING

Evgeny Panidi, Eduard Kazakov, Anton Terekhov and Evgeny Kapralov

Department of Cartography and Geoinformatics, Saint-Petersburg State University
33 10-th Line V.O., Saint-Petersburg 199178, Russia
Email: panidi@yandex.ru

ABSTRACT

Nowadays we see the rapid growth of solutions number for geospatial data processing in the Web (i.e. geoprocessing). One of the main trends of Web geotechnologies evolution is the transition from Web map applications to the Web GIS applications, which are supplement the maps delivery with the analytic tools providing to the end user through Web interface. In fact, the only general open standard describes implementation rules for Web geoprocessing services. This is the Open Geospatial Consortium Web Processing Service standard, which is fully server-oriented. Moreover, the vast majority of currently used solutions (both open source and proprietary) are server-oriented, i.e. assume the server resources only as the computational resource. However, some researchers underline that it is possible way to transmit the executable code to the client for client-side computations and geoprocessing. Also, some general Web architecture concepts assume the effectiveness of client-side computations, e.g. Fog Computing concept. Our practical experience also shows that in some cases it is useful to have ability of client-side geoprocessing, which is not opposite but complement technology to the server-side processing technologies. In addition, we believe that it is more useful to have the ability to run the same processing tool by choice on server or client side. We name such double-sided services as Hybrid Geoprocessing Web Services.

We study and discuss the approaches to gap filling in client-side geoprocessing general schema. For this purpose, we implemented previously the getProcess request as addition to the WPS protocol. Additionally at the previous steps of our study, we proposed a possible structure of getProcess request and draft XML file structure for its response, which describes the list of executable resources and their dependencies. Currently we working on detailed methodology of processing tools implementation and testing. We use the Python programming language as primary development tool, because of its applicability to build both server- and client-side cross-platform processing tools using single core program code. We use Python also for implementation of needed infrastructure components, such as HGWS server that supports the getProcess request/response performing, and client-side Runtime Environment that provides executable code orchestration on the client. Achieved results need to be discussed widely and carefully. However, main conclusion of our current work is that client-side geoprocessing schema in general could be relatively simple and compatible backward with current standards. The HGWS concept is applicable when implementing client-side geoprocessing Web services in small-scale projects and could be the entering point for study of distributed geoprocessing systems implementation.

1. INTRODUCTION

Only a few years ago the primary use cases of the Web, when operating with geospatial data (geodata), were the data delivery to the end user and the graphical representation in the browser or in the desktop application, such as the well-known Google Earth geobrowser (*http://www.google.com/earth/*), and desktop software of Geographic Information Systems (desktop GISs). Nowadays we may observe the rapidly growing amount of different Web-geospatial applications. Geospatial technologies implementation in the Web became the basis for the Geoweb (Scharl A. *at al.*, 2007) knowledge area formation, which integrates the geospatial aspect into data structuring and data search technologies. Also the use of Web

technologies in Geoinformation Science and Technology (GIS&T) made possible the emergence of such disciplines as Web Mapping and Web Cartography (Kraak M.-J. *at al.*, 2001; Fu P. *at al.*, 2010), which are incorporate the techniques and methods of graphical map-like data visualization in the Web. This way of common use of the Web-based tools and geodata was quite natural at the early evolution stages, taking into account that the Web infrastructure in General was formed primarily as a mean of universalized data representation. However, some new trends are formed at this cross-scientific and multidisciplinary knowledge area.

One of the most important trends is the shift from use of passive operations with geodata (i.e. storage, transmission and visualization) to active operations (i.e. data mining and pattern recognition, data processing and different area specific types of data analysis). One of the sources for this shift is the paradigm change in Web technologies, namely the implementation of the Web 2.0 concept (O'Reilly T., 2005), that implies (despite the presence of different interpretations) the user interaction with the Web environment and the transition from static Web sites to Web(-based) applications. On the other hand, the evolution of Web mapping/Web cartography included not only the implementation of Web mapping applications and interactive geodata representation, but repeated the way of the GIS&T in whole and transited from presentation to processing. In this way, the term "Web GIS" has become widely used in recent years. This term indicates the Web-based systems and applications, targeted on solving analytical problems using geodata and geodata manipulation problems in general (Fu P. *at al.*, 2010).

These evolutionary processes are confirmed with the Open Geospatial Consortium (OGC) standards appearance chronology (*http://www.opengeospatial.org/*):

- Web Map Service (WMS) – 2000 (Doyle A. (ed.), 2000).
- Web Feature Service (WFS) – 2002 (Vretanos P.A., (ed.), 2002).
- Web Coverage Service (WCS) – 2003 (Evans J. (ed.), 2003).
- Web Processing Service (WPS) – 2007 (Schut P. (ed.), 2007).
- Web Coverage Processing Service (WCPS) – 2009 (Baumann P. (ed.), 2009).

Additionally, it should be noted that the paradigm of Cloud Computing (Buyya R. *at al.*, 2011) and the use of thin clients are the most used basis for current Web-based geotechnologies. The services, which are implemented accordingly to the last two of above-mentioned standards, transmit only the result data to the user and look like dynamically generated data services for the user, despite the standards' names. This is due to that these standards are focused on the server-side data processing. The processing tools in this case are hidden deeply from the user at the server side. Thus, current standards describe the implementation rules for geoprocessing Web services, which could be fitted well only within the architecture of cloud solutions or computing clusters where the data processing is performed on the server side under centralized control. This centralized-control approach is used in other current trend areas such as development of the technologies for integration of distributed geospatial data arrays and technologies for big geospatial data.

Finally, another one current trend of automation of the Internet and Web makes the growing impact onto GIS&T. This trend related with such concepts as Web 3.0 (Calacanis J., 2007), Semantic Web (Berners-Lee T. *at al.*, 2001), Semantic Web Services (SWS) (McIlraith S.A. *at al.*, 2001), and Internet of Things (IoT) (Ashton K., 2009). These concepts together form the opportunities for implementation of machine-readable services in addition to machine-readable data. The growing impact of these concepts is confirmed by appearance

of OGC standards related to the design and use of sensor grids, which are included into the OGC Sensor Web Enablement framework (Bröring A. *at al.*, 2011).

It is quite remarkable in this context that the gap remains in the area of client resources use. The use of decentralized systems (Grid Computing systems for example) has a small reflection in geoinformation technologies, except of some case projects (Aktas M.S. *at al.*, 2005; Coene Y. *at al.*, 2007). However, the consideration of modern concept of Fog Computing (Fernando N. *at al.*, 2013; Hong K. *at al.*, 2013; Abdelshkour M., 2015) in conjunction with the concept of Semantic Web Services, allows to assume that the question of client-side computing in Web-based geodata processing services will be relevant in the near future.

Thus, taking into account these concepts, when implementing the Web-based geodata processing services, we could make next conclusions:

- We should not neglect the computational resources available on the client side, which are sufficient for solving many computational tasks and data processing tasks.
- When implementing the Web services for data processing (including geodata processing) on the client side, we need to automate the search and use of such services.
- Therefore, standardization of the processing execution rules on the client side, is needed additionally.

The use of Web services for client-side geodata processing involves the transmitting to the client of the executable files of some process or processes, which contain the source code of some processing algorithms. The idea of executable code transmitting is quite obvious and is not new in general. In relation to the geodata processing, it was mentioned by other authors also (Coene Y. *at al.*, 2007; Keens S. (ed.), 2007). However, widely-known and standardized solution of this idea is not known for us. In our project, we study the ways and possibilities of unification (and standardization in the future) of the executable code transmitting methods and means for the Web geoprocessing services (Kazakov, 2013; Kazakov *at al.*, 2015; Panidi, 2013; Panidi, 2014; Panidi *at al.*, 2015). We working on implementations of such services, which imply the ability to perform processing on both server and client side. These services we name as the Hybrid Geoprocessing Web Services (HGWS).

2. PROJECT SCOPE

The main subjective reason of the project in our case is the presence of a number of thematic projects related to the geodata processing. Some thematic specialists involved in these projects, who do not have advanced skills in the use of GIS software for data analysis. In this context, being the GIS professionals, we face with the task of design, implementation, configuring, and providing some software tools for data processing and analysis for the end user (some thematic specialist in other words).

It is obvious that the end user could explore new software tools easier if these tools are in a familiar interface. It is also obvious that the most intuitive desktop interface for many users today is the Web interface. Such background became the main prerequisite for the desire to use the Web services for geodata processing in our thematic projects.

However, many times we faced a situation in which the use of currently standardized

technologies becomes irrelevant in small and medium-size projects, due to focusing of these technologies on the server-side processing. The server-oriented feature of geospatial Web standards leads to the need of building of the server or cloud-based computing infrastructure, and additionally involves the time spending when transmitting initial data to the server for processing.

Thus, the main goal of our project is the technologies and tools development for complementation of existing Web-based approaches to geodata processing with the client-side processing techniques. Moving to this goal, we assume:

- To minimize requirements (software and hardware) for the server-side part of system by transmitting only compact executable processing application components to the user.
- To ensure backward compatibility with existing standards, especially by providing backward compatibility with the OGC WPS standard and possibility to perform processing on the client or server side by user's choice, accordingly to available resources and context of the project.
- To give the user more flexible control of geodata that stored at the client side when processing with Web services by providing a client side processing for applicable cases.

2.1 Development tools

As the base software platform, we use the Apache HTTP Server (*http://httpd.apache.org*) and PyWPS WPS server (*http://pywps.wald.intevation.org/*). The choice of HTTP server is due only to the Apache administering experience and do not imply any special requirements to HTTP server performance. The absence of these requirements concerned with the fact that currently we perform the project as a case study and a prototype solution, but not as an industrial solution. The choice of PyWPS is due to that this WPS server was the only stand-alone solution (not integrated into any geospatial software complex for servers) at the time of the project launch. PyWPS was used for WPS functionality implementation.

As the primary program code development tool, we use the Python programming language (*https://www.python.org/*). All HGWS functionality that differs from WPS functionality (Mueller M. *at al.*, 2015; Schut P. (ed.), 2007) were implemented using Python as separate HGWS software tools (server-side or client-side).

However, when implementing the project we pay special attention to the exclusion of fixed bindings to the programming tools and try to implement independent architecture that suitable for the design of abstract specification.

2.2 Web interface

For purposes of HGWSs publishing and testing, we created a prototype of geoportal interface (*http://195.70.211.131/*). The Web interface that implemented in this prototype is very simple. It and includes several descriptive Web pages (such as home page, page with a list of publications, etc.) and two main pages for user interactions. These are the personal user account page and process execution page (Fig. 1).

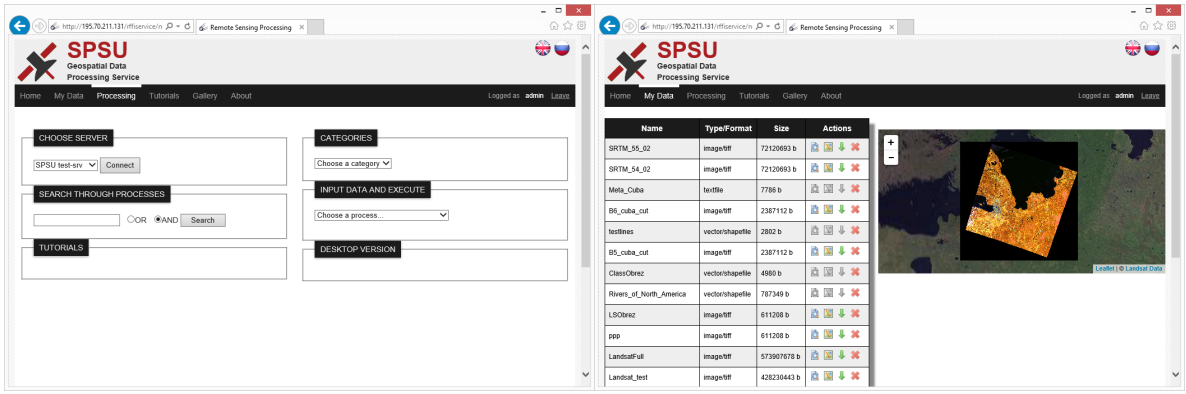

Figure 1. Process execution page (on the left) and personal user account page (on the right) at the prototype of HGWS geoportal.

Besides the possibilities of data processing on the server and downloading client-side processing tools, the resource allows to store amount of datasets on the server. This possibility eliminates the need of repeated uploading and downloading the same datasets, for example in the case of running multiple operations step by step.

Additionally, it is possible to explore the results of processing on the server using, previews directly in the browser window. This feature allows to reject incorrect processing results, without downloading on client computer. The personal user account could be used after user's registration on the Web site, which is verified by site administrator in manual mode.

Process execution page is arranged as follows. A WPS getCapabilities request is generated automatically, when the user selects some of available WPS/HGWS servers in the first dropdown list on the page (Fig. 1). After receiving of response, the response content is used for generating of list of the processes available at the selected server. This list appears in another one dropdown list on the page. Also, the text descriptions are loaded on page in the case of availability. User can do the search in these descriptions by entering any keywords in search box. After selection of a desired process, the WPS describeProcess request is generated and the page is updated with the input boxes for process inputs using response content. In the case of presenting of some geodata files in the process inputs list, the corresponding input boxes include dropdown lists of files, which are stored under user's personal account and have required format.

After filling all required input boxes of process inputs, user could execute the process. In the case of server-side execution, process output files appear in the user's personal account and could be observed online in preview mode or downloaded.

3. IMPLEMENTATION

The implemented prototype system of publishing and execution of HGWSs includes two base subsystems. Server subsystem incorporates a stack of the HTTP, WPS and HGWS servers and implements the access to software tools for geodata processing. Client subsystem incorporates a set of applications required for graphical user interface providing, downloading of the process software components and executing process on the client computer. These subsystems are presented on schematic diagram (Fig. 2).

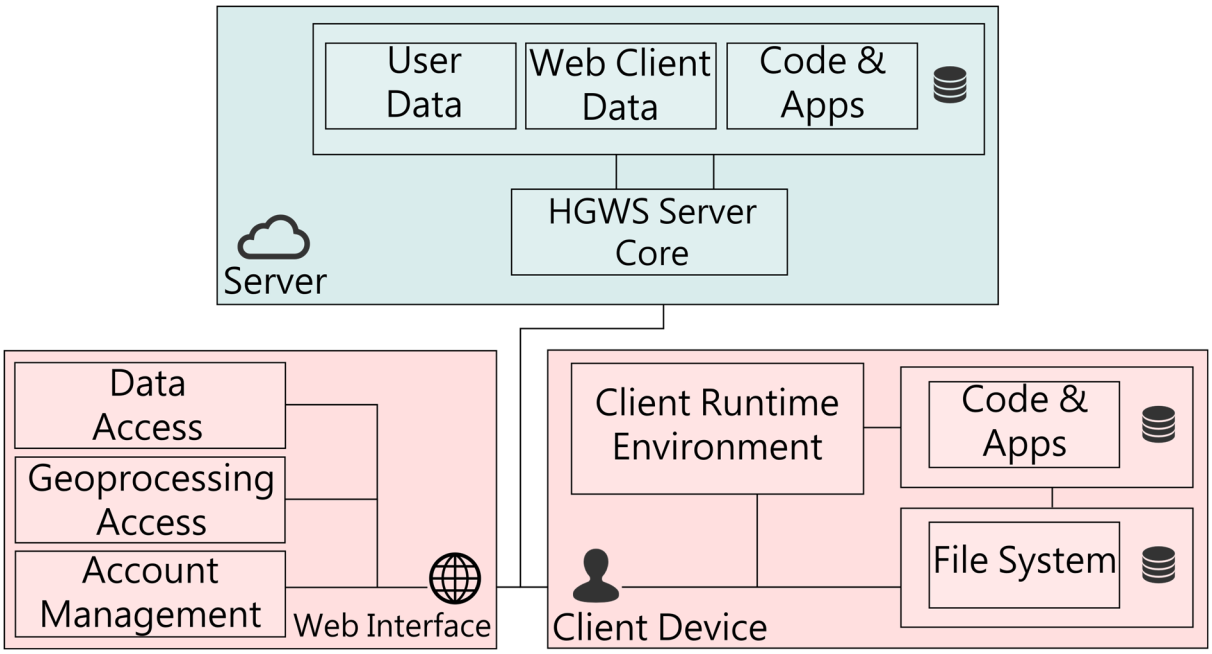

Figure 2. Server (upper block) and client (lower blocks) subsystems of implemented prototype of the HGWSs publishing and execution system.

In our implementation, we have separated WPS and HGWS servers. The last one provides only the client-side processing functionality. However, in general case (as it presented on the diagram) the HGWS server should provide both the client-side and the server-side processing functionality. Server subsystem accepts client requests and controls the flows of data files and executable files, as well as server databases, which store all of these files. The user can interact with stack of the servers in two modes. The first mode is via the Web interface, by uploading data to the server and runing server-side processing. Either the second mode is via the desktop application (client Runtime Environment or RE), by receiving on a local client device all of necessary software components of the process.

3.1 Server subsystem

The stack of servers performs the task of interaction with user via the Web interface or RE. OGC WPS standard implements requests, which are needed for server-side processing. For handling of the request(s), which is(are) needed for client-side processing, we need to define the structure of transmitted data (i.e. transmitted software components).

Here we need to introduce two new concepts concerning the types of downloadable software components. The concept of Functional Module (fModule) assumes discrete file, which serves the program code of some processing algorithms (it can be a Python script file, DLL, Java bytecode file, etc.). The Extra Application (extraApp) is the standalone program (usually compiled binary file) that implements some processing algorithms and can be executed directly for processing of some specific geodata files. In this way, we may reproduce some open source GISs, such as SAGA (*http://www.saga-gis.org*) or GRASS (*http://grass.osgeo.org*) for HGWS implementation (Fig. 3). This approach assumes the separate transmitting of the application core (extraApp) and its libraries (fModules).

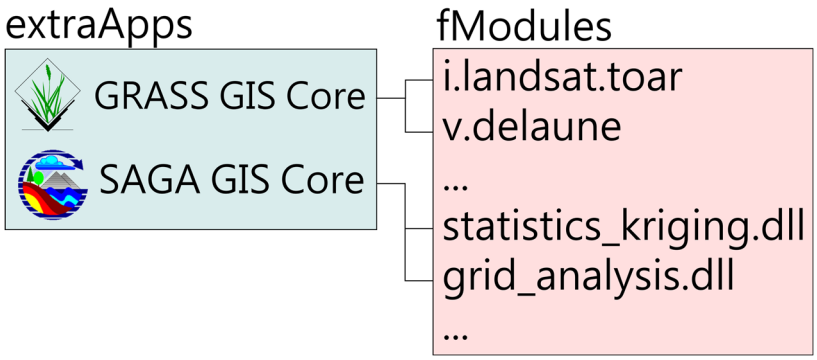

Figure 3. Separation of the cores of applications (extraApps) and their libraries (fModules).

The extraApp and fModule definitions are needed because of potentially complicated structure of process program code and possibility of dependencies in code structure.

Additionally we need to define and store the metadata, which describe properties of process as well as properties and dependences of the software components of both types. Each process could be associated with an unlimited number of fModules and extraApps. The only limitation is needed to store the software component files. This is the uniqueness of file names within the system. Other metadata parameters for software components (except name) are the file type, description, operation system compatibility record. Metadata parameters for process include the process name, abstract (description), inputs and outputs lists, desktop (client-side) version availability flag and lists of fModules and extraApps (Fig. 4).

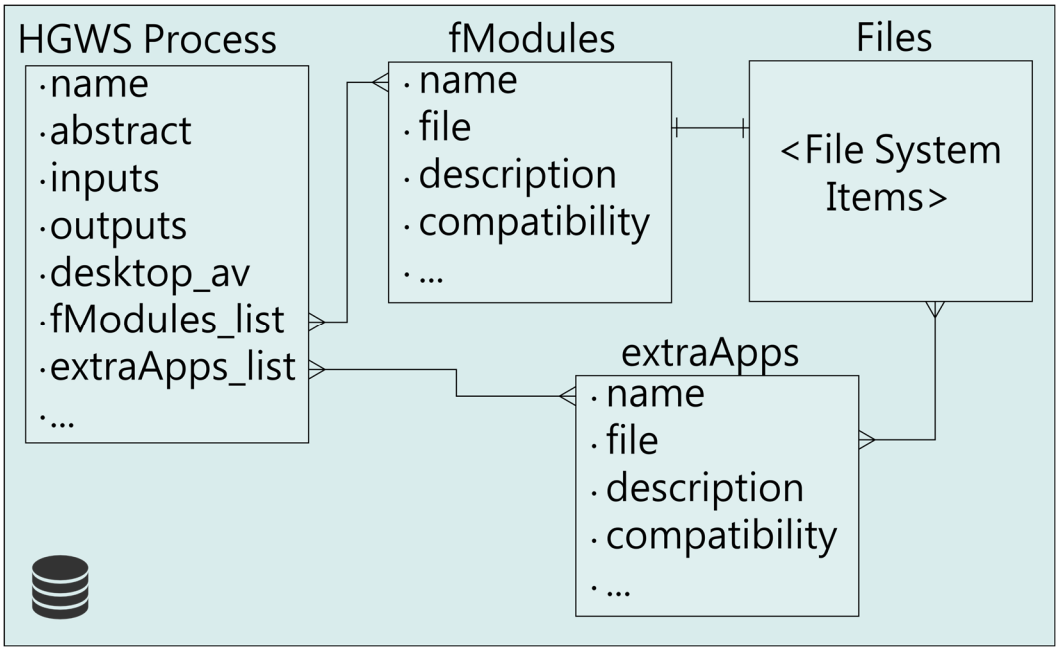

Figure 4. Structure of software components of the HGWS process.

3.1.1 *Requests*

The extension of the WPS standard with the possibilities of client-side processing and software components transmitting involves the implementation of a new request type that will be used for getting software components on the client side. We named this request as getProcess request. The parameters of this new request type are listed in Table 1:

Table 1. getProcess request parameters.

Name	Presence	Description
Request	Required	Identifies service request. Must be "getProcess".
Service	Required	Identifies service type. Must be "WPS".
Version	Required	Identifies service version.
Identifier	Mandatory	Process identifier as listed in the Capabilities document.
Platform	Required	Identifies user's desktop platform. Could be "Win32", "Win64", "Mac", "Linux Debian" etc.

The getProcess request could be executed using GET request method with above-mentioned parameters:

http://some.host/server?
Request=getProcess&
Service=WPS&
Version=x.0.0&
Platform=Win32&
Identifier=index-NDVI

This request takes its place in the chain of interactions between client and server (Fig. 5). The user chooses the server and connects to it. After that, client executes the WPS getCapabilities request to the selected server, and gets the response, which is extended with some value of the "desktop-available" attribute. After this, client could parse the list of available processes as well as the list of the processes available for client-side processing.

At the next step, the user selects some process, and the client executes the WPS describeProcess request. Then client parses the lists of the process inputs and process outputs from the describeProcess response, and generates the Web or desktop graphical user interface, depending on use of the browser or RE. Additionally, in the case of client-side processing, the entry point or the program base interface is generated that connects to the user interface and receives all process inputs from the user interface as arguments.

Next, the HGWS getProcess request executed in the case of client-side processing. Client parses the response and gets the lists of fModules and extraApps. Now client becomes able to download the needed software components. In our implementation, we store all downloaded software components into some standard directory, so previously downloaded software components could be excluded from downloading list.

All inputs that were received from user interface can be transmitted then to one of downloaded fModules, which has the name equal to the process name and provides input function with unified name.

Finally, the Execute request can be executed to start the processing.

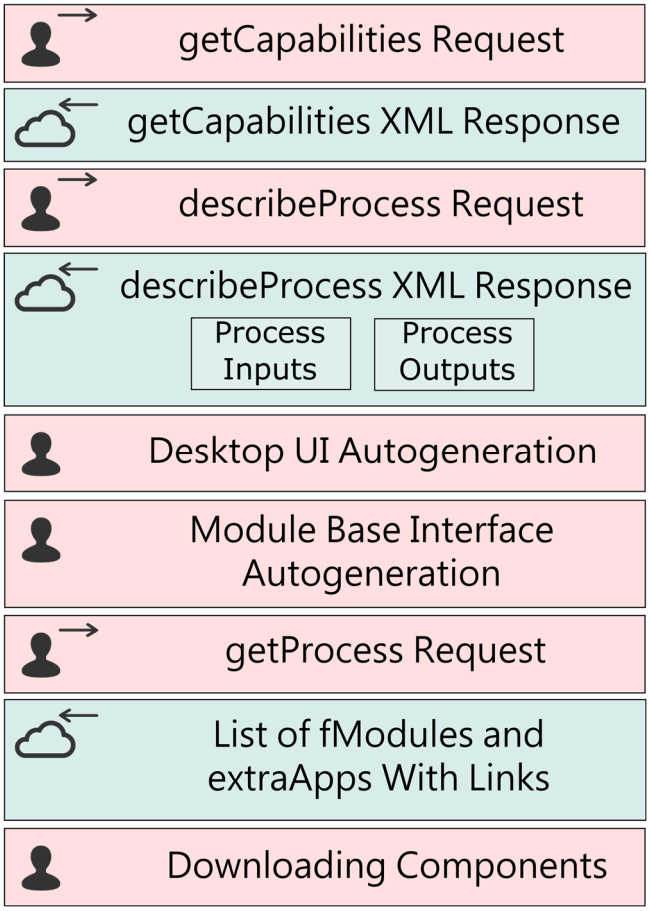

Figure 5. Chain of the requests to server in the case of HGWS processing.

3.1.2 *getProcess response structure*

The getProcess response is XML document that contains all necessary information about the software components needed for client-side processing using particular process. This document could include unlimited number of descriptions of the fModules and extraApps with metadata and download links. Here is an example fragment for such XML document:

```
<!--Module parameters-->
<complexType name="FModule">
    <complexContent>
        <name>sampleFModule</name>
        <annotation>sampleFModule annotation</annotation>
        <compability>
            <!-- Description of platform compatibility for module-->
            Win32-common
        </compability>
        <type>
            <!--Type of software component-->
            DLL
        </type>
        <downloadLink>URL</downloadLink>

        <usedExtraApps>
```

```
                    <complexContent>
                        <name>SampleExtraApp</name>
                        <annotation>sampleExtraApp annotation</annotation>
                        <compability>Win32-common</compability>
                        <type>Binary</type>
                        <downloadLink>URL</downloadLink>
                    </complexContent>
                </usedExtraApps>
            </complexContent>
        </complexType>
```

3.2 Client subsystem

The RE is needed on a client computer to provide the process execution. In the simplest case, a software client (desktop application) or a Web browser that displays a graphical user interface are able to play the role of the RE. However, the software client is more usable, due to security limitations of running executable files directly from browser interface.

In our solution, the RE is a desktop application, which displays the user interface; implements the tasks of loading, storage and removal of the process executable files; verifies are the client computer parameters applicable to run geoprocessing; and controls the starting and completing of geoprocessing. More broadly, the RE could include some external software, which are necessary to run process executable code, in particular the Python interpreter, in our case.

We assume two scenarios. The first is downloading of the settings file using our Web interface, and automatic opening of this file in the RE application with automatic initialization of the processing (i.e. connection to the server, loading the needed software components, and user interface generating). The second is working in manual mode, when the user enters the HGWS server URL and other parameters, and performs step by step all of the operations needed to select and parametrize some of the available processes.

In case of the first scenario, the settings file could be a JSON file, which incorporates all necessary values of process parameters:

```
{
  "server": {
            "host" : 195.70.211.131/hgws/hgws?
            "port": 8080
            }
  "HGWSversion": "0.1v",
  "Request": "getProcess",
  "ProcessName": "imagery-landsat8-to-radiance",
}
```

Using this file, the client RE may automatically execute needed requests, analyze responses, and alerting the user with errors or correct processing finalization.

The RE downloads software components of the process using getProcess response content and, organizes the storage of downloaded executable files and metadata.

Additionally, RE generates graphical user interface using describeProcess response content, as well as the base program interface that connects the user interface with process executable files.

The RE prototype is developed using Python PyQt (*https://riverbankcomputing.com/software/pyqt/intro*) tools to provide cross-platform interoperability, and Python PycURL library to implement HTTP requests. The main menu (Fig. 6) allows opening the JSON settings file manually, configuring the server connection settings to the server, and observing the list of already downloaded processes (Fig. 7) with the ability to run them. The user interface is generated in a simple manner. Each process input parameter is mapped to interface element that was rendered by PyQt (Fig. 8).

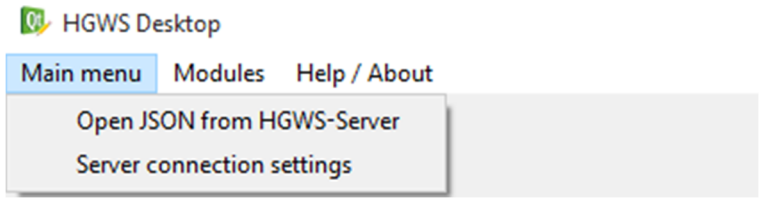

Figure 6. RE main menu.

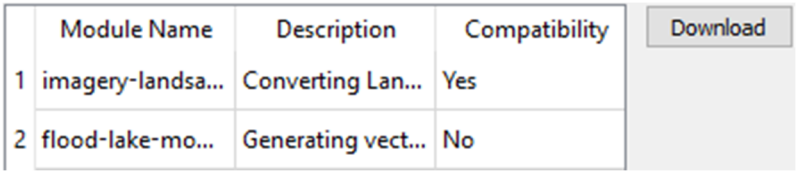

Figure 7. List of available processes.

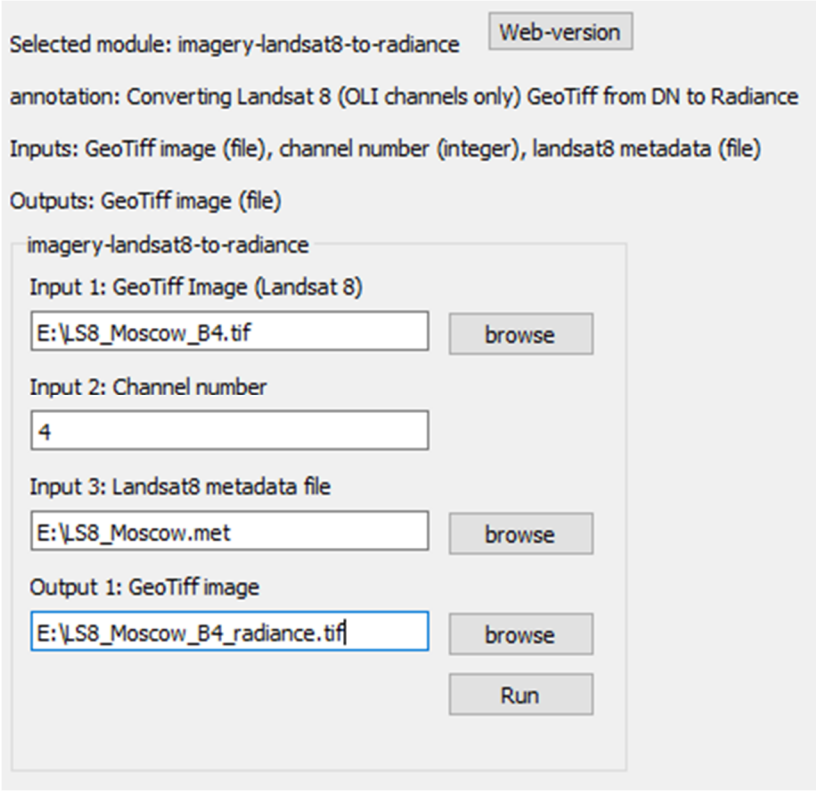

Figure 8. User interface of selected process.

4. RESULTS

Summing up, we should note that selected approach is workable in general and allows to solve the task of providing client-side computations when using Web services for geodata processing. Of course, some issues require further investigations still. The most interesting of these are the security issues of running of the executable code received from a third party server, and issues of the processes orchestration both in case of processing under user control and in case of machine-to-machine interface implementations.

However, we have formulated approach and created prototypes of software tools. These tools we can already use when performing small-scale projects for implementation of traditional advantages of the Software as a Service architecture (Software & Information Industry Association, 2001), such as universal and user-friendly front-end interface and low-cost back-end and processing program code administration.

In conclusion, we may point the basic methodology regulations, which we formulated as a result of working on the project. These regulations formalize the process of designing and implementing of HGWSs:

- Providing the backward compatibility of the HGWS processes and HGWS servers with WPS clients.
- Providing the maximal atomization of software components of the process, i.e. decomposition of the complex libraries into the amount of separated libraries.
- Providing the ability of choice of the side (client/server) for processing when executing processes.
- Providing the metadata with descriptions of the dependencies between the software components of the processes.
- Providing the metadata with descriptions of the client resources and functionality, which are needed for the process execution.

5. OUTLOOK

Our research opened a number of theoretical and practical issues, which require additional consideration. Next, we mention briefly some of them.

We have previously identified the need to ensure the safe running of executable code on the client side. Of course, if we execute program code that received from a third party server, the security issue requires a deep study. However, detailed regulation of security issues is hardly possible in abstract specifications or implementation standards for such services. At the current stage, we are of the opinion that the issue of the safe running of executable code must be considered in detail individually in the context of each specific project. Theoretical and conceptual foundations of security should be discussed as part of the theory of Web Services in general.

The question of services orchestration and formation of complex systems for geodata processing was also mentioned. It is an urgent task, but, we believe, has no universal solution, because of the complexity and nonlinearity of many data-processing procedures. Part of the problem of orchestration is to provide a consistent and unified description and structuring of the processing inputs and outputs. This issue can be solved by developing descriptive standards for the valid structure of data and metadata.

Another important area of research is related closely to the Internet of Things and Semantic Web Services concepts. This complex task associated with providing of automatic interaction between data servers, processing servers and tools servers. The, tools servers are the host servers of transmittable data processing services (HGWS servers in our terminology). The implementation of automated communication between the servers can be useful in distributed geodata (including big geodata) processing systems.

In our opinion, the conceptual gap of the resources allocation and redistribution on the server side for server-side processing requires some studies also. This gap is associated closely with WPS standard, which we used in our project as a base standard. We resolved the problem of redistribution of the limited server resources between users through the mechanism of user personal accounts. However, this mechanism is extremely inconvenient from the point of view of administration costs and from the point of view of processing automation abilities. The decision is likely to be dynamic allocation and redistribution of resources and implementation of data life time limit on the server as the data attribute.

Finally, current results of our project itself require detailed discussion between users and developers. Whereas, the fact that the research was carried out as a case study, we assume that our findings may have gaps and failures. Nevertheless, we hope that our work can be served as the basis for further research of other authors in this direction.

6. ACNOWLEGEMENTS

The study was partially supported by Russian Foundation for Basic Research (RFBR), research project No. 13-05-12079 ofi_m.

7. REFERENCES

Abdelshkour M., 2015. IoT, from Cloud to Fog Computing. Available at: http://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing

Aktas M.S., Aydin G., Fox G.C., Gadgil H., Pierce M. and Sayar A., 2005. Information Services for Grid/Web Service Oriented Architecture (SOA) Based Geospatial Applications. Community Grids Lab, Indiana University, Technical Report. Bloomington, Indiana, USA

Ashton K., 2009. That "Internet of Things" Thing. *RFID Journal*. Available at: http://www.rfidjournal.com/articles/view?4986

Baumann P. (ed.), 2009. Web Coverage Processing Service (WCPS) Language Interface Standard. OGC 08-068r2. Version: 1.0.0, 2009-03-25

Berners-Lee T., Hendler J., Lassila O., 2001. The Semantic Web. *Scientific American Magazine*. May 2001

Bröring A., Echterhoff J., Jirka S., Simonis I., Everding T., Stasch C., Liang S., and Lemmens R., 2011. New Generation Sensor Web Enablement. *Sensors*, Volume 11(3), pp. 2652-2699, doi:10.3390/s110302652

Buyya R., Broberg J., Goscinski A. (eds.), 2011. *Cloud Computing: Principles and*

Paradigms. John Wiley & Sons Inc., 637 p., DOI: 10.1002/9780470940105.fmatter

Calacanis J., 2007. Web 3.0, the "official" definition. Available at: http://calacanis.com/2007/10/03/web-3-0-the-official-definition/

Coene Y., Marchetti P.G., Smolders S., 2007. Architecture and Standards for a Distributed Digital Library of Geospatial Services. Third Italian Research Conference on Digital Library Systems (IRCDL 2007) Padova 29–30 January 2007 Proceedings pp. 52–60

Doyle A. (ed.), 2000. OpenGIS Web Map Server Interface Implementation Specification. OGC 00-028. Version 1.0.0, 2000-04-19

Evans J. (ed.), 2003. OGC Web Coverage Services (WCS). OGC 03-065r6. Version 1.0.0, 2003-08-27

Fernando N., Loke S.W., Rahayu W., 2013. Mobile cloud computing: A survey. *Future Generation Computer Systems*, Volume 29, Issue 1, pp. 84–106

Fu P., Sun J., 2010. *Web GIS: Principles and Applications*. ESRI Press. ISBN 978-1-58948-245-6

Hong K., Lillethun D., Ramachandran U., Ottenwälder B., Koldehofe B., 2013. Mobile fog: a programming model for large-scale applications on the internet of things. Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing, 16 August 2013, Hong Kong, China. pp. 15-20, doi:10.1145/2491266.2491270

Kazakov E.E., 2013. Approaches to Implementation of Web Services for Processing and Analysis of Remote Sensing Data. In Proceedings of the Earth Remote Sensing From Space: Algorithms, Technology, Data – Young Scientists Workshops (Altai State University, Barnaul, Russia, October 2-6, 2013). pp. 82-86. In Russian. Available at: http://elibrary.asu.ru/xmlui/bitstream/handle/asu/203/read.7book

Kazakov E., Terekhov A., Kapralov E., and Panidi E., 2015. WPS-based technology for client-side remote sensing data processing. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-7/W3, pp. 643-649, doi:10.5194/isprsarchives-XL-7-W3-643-2015

Keens S. (ed.), 2007. Discussions, findings, and use of WPS in OWS-4. OGC Discussion Paper. OGC 06-182r1. Version 0.9.1, 2007-05-10

Kraak M.-J., Brown A., 2001. *Web Cartography Developments and Prospects*. London and New York: Taylor & Francis

McIlraith S.A., Son T.C., and Zeng H., 2001. Semantic Web Services. *IEEE Intelligent Systems*, Volume 16, Issue 2, pp. 46-53

Mueller M., Pross B. (eds.), 2015. OGC WPS 2.0 Interface Standard. OGC Implementation Standard. OGC 14-065. Version 2.0, 2015-03-05

O'Reilly T., 2005. What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software. Available at: http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

Panidi E.A., 2013. Geoservices for Online Remote Sensing Data Processing. In Proceedings of the Earth Remote Sensing From Space: Algorithms, Technology, Data – Young Scientists Workshops (Altai State University, Barnaul, Russia, October 2-6, 2013). pp. 74-81. In Russian. Available at: http://elibrary.asu.ru/xmlui/bitstream/handle/asu/203/read.7book

Panidi E.A., 2014. Towards Client-Side Web Processing Services. Proceedings of OSGeo's European Conference on Free and Open Source Software for Geospatial (FOSS4G-Europe 2014) July 15-17 2014 Jacobs University, Bremen, Germany. 4 p. Issued online. Available at: http://europe.foss4g.org/2014/content/toward-client-side-web-processing-services.html

Panidi E., Kazakov E., Kapralov E., Terekhov A., 2015. Hybrid Geoprocessing Web Services. In SGEM2015 Conference Proceedings (15th International Multidisciplinary Scientific GeoConference SGEM, Albena, Bulgaria, June 18-24, 2015), Book 2 Vol. 1, pp. 669-679

Scharl A., Tochtermann K. (Eds.), 2007. *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*. London: Springer-Verlag.

Schut P. (ed.), 2007. OpenGIS Web Processing Service. OpenGIS Standard. OGC 05-007r7. Version 1.0.0, 2007-06-08

Software & Information Industry Association, 2001. Software As A Service: Strategic Backgrounder. Washington, D.C., USA

Vretanos P.A., (ed.), 2002. Web Feature Service Implementation Specification. OGC 02-058. Version 1.0.0, 2002-05-17