EU Framework Program for Research and Innovation (WATER-4a-2014 - H2020)

Project Nr: 641821

Applying European market leadership to river basin networks and spreading of innovation on water ICT models, tools and data.

Deliverable D5.4
Interoperability recommendations report

Version 1.0

Due date of deliverable: 31/08/2016
Actual submission date: 14/12/2016

Document control page

Title	D5.4 Interoperability recommendations report			
Creator	CS_52N			
Editor	CS_52N, SJ_52N			
Description	This deliverable is based on the results of the Interoperability Experiment and provides a list of practical recommendations to increase interoperability at different levels for tools and policy briefs.			
Publisher	WaterInnEU Consortium			
Contributors	WaterInnEU Partners			
Type	Text			
Format	MS-Word			
Language	EN-GB			
Creation date	01/12/2016			
Version number	1.0			
Version date	14/12/2016			
Last modified by	CS_52N			
Rights				
Dissemination level		CO (confidential, only for members of the consortium)		
	X	PU (public)		
		PP (restricted to other programme participants)		
		RE (restricted to a group specified by the consortium)		
	When restricted, access granted to:			
Nature	X	R (report)		
		P (prototype)		
		D (demonstrator)		
		O (other)		
Review status		Draft	*Where applicable:*	
		WP leader accepted		Accepted by the PTB
		PMB quality controlled		Accepted by the PTB as public document
	X	Coordinator accepted		
Action requested		to be revised by all WaterInnEU partners		
		for approval of the WP leader		
		for approval of the PMB		
		for approval of the Project Coordinator		
		for approval of the PTB		
Requested deadline	14/12/2016			

Revision history			
Version	Date	Modified by	Comments
0.1	01-12-2016	CS_52N, SJ_52N	Initial Version
0.2	05-12-2016	CS_52N	Edit of Sections 2, 3, 4
0.3	09-12-2016	CS_52N, SJ_52N	Revision of Sections 2 and 3 according to feedback
0.4	12-12-2016	CS_52N	Revision according to feedback from partners.
1.0	14-12-2016	CS_52N, EJ_52N	Final edits

Executive Summary

This document describes recommendations based on the results of the River Basin Interoperability Pilot (RiBaSE). The purpose of this experiment was to assess the efficacy of Open Geospatial Consortium (OGC) standards for describing status and dynamics of surface water in river basins, to demonstrate their applicability and finally to increase awareness of emerging hydrological standards as WaterML 2.0. Also, this pilot was used to identify potential gaps in OGC standards in water domain applications, applied to a flooding scenario in present work. The proposed architecture for the pilot consisted of OGC Sensor Observation Services, OGC Web Processing Services, a Publish/Subscribe component that based on the recently published OGC Publish/Subscribe standard, and of an OGC Web Feature Service for storing the flooded areas. The solution was successfully tested in three transboundary river basins: Scheldt, Maritsa and Severn. However, some issues were identified from which the recommendations presented in this report were derived. Since the pilot was focusing on technical standards for geospatial software interoperability, they do not contain recommendations related to organisational structures, human procedures, etc.. The target audience are decision makers, data managers and model implementers with a technical background. However, the recommendations presented here may serve as input for higher-level water management policies. Furthermore, some specific recommendations for the standards used are given. The OGC may consider these recommendations to improve the standards.

The pilot demonstrated that the usage of open standards eases integration of different data sources and avoids technical (syntactical) as well as semantic conflicts. However, a pre-requisite is that data and models are open and accessible. Since observing and modelling our environment is a complex task, the standards for representing the observations and models are of a complex and generic nature. We hence recommend hiding the complexity in the user interfaces, defining profiles of these standards for the hydrology domain like already done with WaterML 2.0 and the SOS Hydrology profile, and simplifying the implementation of Web clients by also providing lightweight REST bindings for the service interfaces instead of the heavy weight SOAP bindings.

Table of Contents

List of Figures

1. Introduction

This document describes recommendations for tool harmonization and policy briefs based on the results of the River Basin Interoperability Pilot (RiBaSE). A brief overview on the pilot is given in Section 2. For a comprehensive description of the pilot, see WaterInnEU - D5.3. After the overview on the pilot, general recommendations for enhancing interoperability between software tools in the water domain are described (Section 3), followed by recommendations for the standards used in the pilot (Section 44). Finally, conclusions are presented in Section 5.

2. Overview on RiBaSE Pilot

In a nutshell, the RiBaSE pilot aimed to demonstrate and evaluate how standardized data models and interfaces between software tools may ease the integration of data from several sources in a flood model and how to provide access to the flood model and its results in an interoperable manner. The flood model of the pilot calculates flooded areas by fusing river flow data with additional base date, e.g. a digital elevation model (DEM). The following questions were considered in the pilot:

- Which technical standards ease the integration of different data sources from different geographic regions with a flood model?
- How can the flood model be shared and applied in several use cases?
- Which additional client components are needed to (i) execute the model, (ii) to visualize and explore the resulting affected areas, and (iii) to subscribe for notifications, in case a point of interest is within a predicted flooded area? Do the standardized interfaces allow the usage of different out-of-the-box clients?

In order to address these questions, the participants of the pilot defined a general architecture that was tested in three different case studies (Pesquer et. al. 2016). Figure 1 shows the general architecture, which is based on standards defined by the Open Geospatial Consortium (OGC), a standardization body that defines standards for interoperability between Geographic Information Systems (GIS) and Web applications. The RiBaSE architecture consists of the following components:

- Sensor Observation Service (SOS): The SOS (Bröring et. al. 2012) is used for the provision of observations and corresponding sensor metadata. A specialization of this standard is available for the hydrological domain relying on the WaterML standard (Andres et. al. 2014). In the pilot, participants provided input data via SOSs.
- Web Processing Service (WPS): The WPS (Müller & Pross 2015) defines a common interface for running geocomputations in the Web. These geocomputations may range from simple calculations to complex environmental models. The WPS was used to deploy the flood model in the different case studies and to make it executable in the Web.
- Transactional Web Feature Service (WFS-T): The WFS (Vretanos 2016) is a component that provides a standardized interface for storing and accessing geographic features with vector geometries. The models and encodings for the geographic features are defined in the Geography Markup Language (GML). The transactional interface of the WFS allows inserting new data. In the pilot, the WFS was used to store the flooded areas generated by the model.
- Publish/Subscribe (Publish/Subscribe): The OGC Publish/Subscribe specification (Braeckel et. al. 2016) has been recently released and complements the standards above by defining an interface that allows data providers to publish their geospatial information and clients to subscribe to this information.

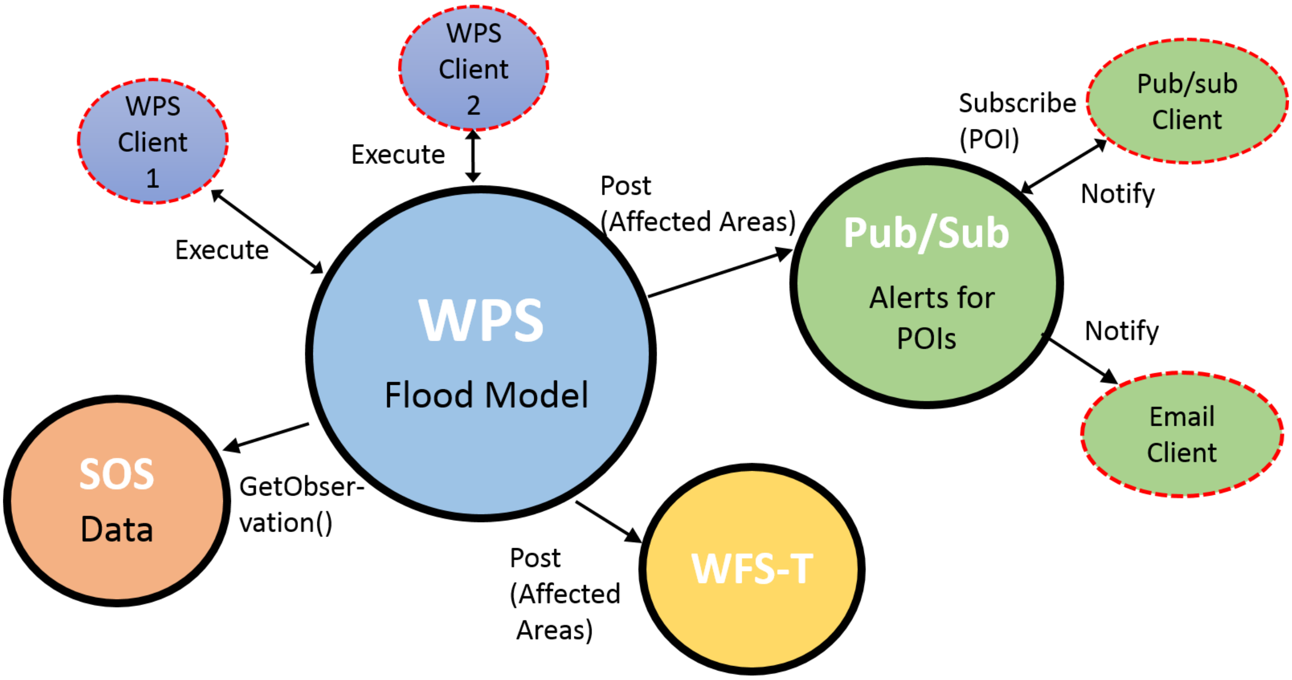

Figure 1: Overview on the architecture defined in the interoperability pilot.

The architecture has been applied in three different case studies covering the basins of the river Severn in the UK, of the river Scheldt in Belgium and the Netherlands, and of the river Maritsa in Bulgaria, as shown in Figure 2.

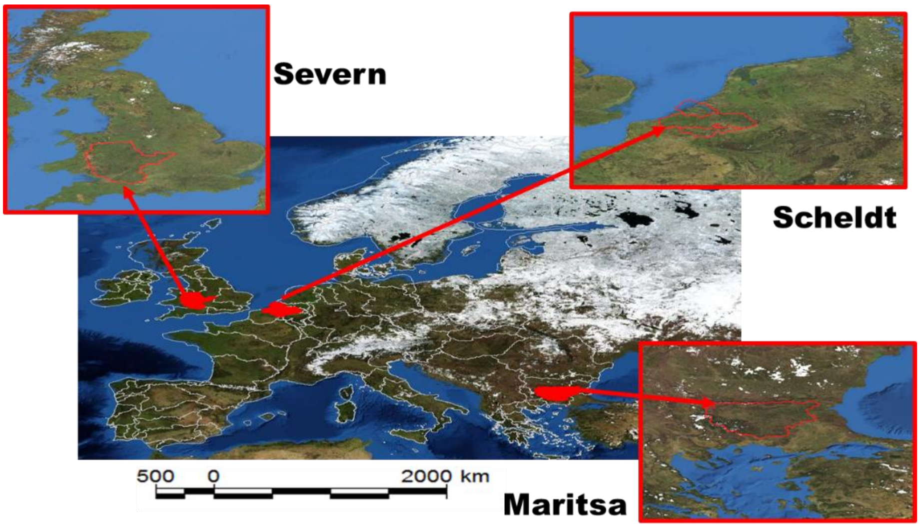

Figure 2: Map location of the pilot river basins

3. General Recommendations

Free access to data is a pre-requisite for sustainable applications in the water domain and for the generation of higher-level information products like flood predictions across borders and over different administrative levels (regional, national, continental). Furthermore, it facilitates reproducibility of model results and scientific studies. However, open access to data is still limited. For example, while measurement data for the Scheldt river basin is available to the public at the Flemish Water Management portal Waterinfo.be portal[1], no such portal exists for the Maritza river basin and getting access to the data for this area is difficult. We consider open access to base data and measurements from public authorities as a prerequisite for interoperability and applicability of models and other products in the water domain.

Recommendation 1*: Base data including geographic maps as well as sensor measurements gathered by public authorities should be publicly accessible via the Web relying on open data licenses.*

Currently, if the data is publicly available in the Web, it is provided in a variety of different data structures and formats, e.g. as Comma-Separated-Values (csv), ESRI's Shape Files (shp), Excel spreadsheets (xls) and many others. Open standards like those defined by the OGC, aim to address this issue and harmonize the technical interfaces and data formats to access the data.

Recommendation 2*: Open data should be provided through standardized interfaces and in standardized formats.*

Assuming that data providers are offering their data to the public in an open and standardized way, the question remains how potential users are able to find the data. As realized with the WaterInfo.be portal, a common entry point is needed that provides an overview on the data provided and additional information about the sensors as well as other potential data source.

Recommendation 3*: A common entry point (URL) to the data should be provided. The entry point should also provide metadata about the data provided, at least the spatial and temporal extent, the formats in which data is available and the phenomena that are represented, e.g. rainfall or flow-rate.*

This common entry point can be referenced from European or global water portals like the WaterInnEU marketplace or the EIP-Water marketplace or GEOSS Discovery and Access Broker. The metadata may be utilized by search engines or catalogues.

While the standards for data provision largely focus on a syntactic interoperability, reaching semantic interoperability would be the next step. The approach for semantic interoperability is to formalize knowledge in the form of vocabularies or ontologies and to refer to the concepts defined in these vocabularies or ontologies from the data. However, common vocabularies for hydrological phenomena are not yet available in a machine-readable form. An example of an ontology that also models observable parameters is the NASA SWEET ontology[2].

Recommendation 4: *If possible, existing vocabularies should be used to describe the semantics of the data. Since there is no common vocabulary available for the hydrological domain, we recommend developing such a vocabulary under the umbrella of a large organization such as the World Meteorological Organization (WMO) or by the hydrology domain working group at OGC.*

[1] http://www.waterinfo.be/

[2] https://sweet.jpl.nasa.gov/

Besides the provision of data, the next step is to enable tool and model interoperability. This was demonstrated in the pilot by deploying the flood model for the different river basins of the case studies. A common approach for deploying models in the Web is currently missing. However, as a general recommendation, we recommend using standardized interfaces for deploying the model in the Web and providing a description of the model with inputs, outputs, and model parameters, preferably in a machine-readable way.

Recommendation 5*: If analysis tools or models are executable in the Web, they should be provided by a standardized interface with machine-readable descriptions of model inputs, outputs and model parameters. The OGC WPS standard should serve as a basis for defining the standardized interfaces.*

Due to the complexity of physical processes in our environment and sensor instruments observing these physical processes, the standards for observations and environmental models come along with a high degree of complexity. However, this complexity also hinders a broad acceptance of the standards, since not all of the complexity is needed for specific use cases and certain users. To deal with this issue, we recommend building simple user interfaces that hide the complexity of the standards and only provide the information and functionality needed for the use case.

Recommendation 6*: User interfaces of software supporting complex technical standards, should be intuitive, kept simple and hide the complexity of underlying standards. Only those information and functionality needed should be offered to the user.*

Besides complexity, some of the standards used are intentionally kept generic in order to cover a broad variety of use cases. It is encouraged to simplify and restrict the standards for specific domains and use cases. As an example, the process definition in the basic OGC Web Processing Service standard does not restrict the functionality and the inputs and outputs of processes, which can be of any type. In order to enable a higher degree of interoperability, these inputs and outputs need to be further specialized for specific geoprocessing functionality in so called process *profiles*. A concrete example of in the hydrology domain is WaterML 2.0 (Taylor 2014), which is a profile of the generic Observations and Measurements (O&M) standard (Cox 2007).

Recommendation 7*: Profiles of base standards should be specified for the hydrological domain.*

Finally, in order to achieve wide adoption of these standards, extensions for the commonly used software tools need to be provided to avoid that the users need to use other software with other workflows and procedures. As examples, the packages *sos4r*[3] and *WaterML*[4] provide access to OGC Sensor Observation Services and WaterML data in the widely used statistical software R by maintaining the commonly used patterns of R. Similar extensions should be used for other common modelling and GIS software.

Recommendation 8*: Extensions to support open standards should be developed for commonly used software to facilitate a broad adoption of the standards and ease the usage for end users.*

[3] https://cran.r-project.org/web/packages/sos4R/index.html
[4] https://cran.r-project.org/web/packages/WaterML/index.html

4. Recommendations for the standards used

After providing general recommendations in the previous section, this section provides recommendations for the different standards that have been tested in the pilot.

4.1. Sensor Observation Service and WaterML 2.0

A common standard for fixed in-situ measurements in the water domain is WaterML 2.0 (Taylor 2014). Based on WaterML 2.0, a hydrology profile for the Sensor Observation Service (Andres et. al., 2014) has been developed that should be used, if a standardized Web Service should provide the data, e.g. in spatial data infrastructures. Since both of these standards target the domain of hydrology and are already in use by several institutions, we recommend using these standards. In addition, the SOS allows filtering on latest values and may be utilized to provide real-time access to the measurements.

Recommendation 9: *We recommend WaterML 2.0 for encoding of time series data at fixed measurement locations.*

Recommendation 10: *The hydrology profile of the Sensor Observation Service should be used for Web Services providing measurements in WaterML 2.0.*

The current OGC Sensor Observation Service specification only defines a SOAP binding[5], i.e. the SOS defines service operations and the operation requests and responses are encoded using XML. From a Web client developer perspective, it is much easier to interact with Application Programming Interfaces (APIs) that follow the REST style defined in Fielding's dissertation (Fielding 2000). Though the architectural style is not restricted to HTTP and JSON as data encoding, the vast majority of REST architectures relies on HTTP and JSON. We hence recommend defining a JSON encoding for WaterML and a REST binding for the Sensor Observation Service.

Recommendation 11: *A JSON encoding for WaterML should be defined and used for data exchange with thin Web clients, e.g. time series viewer.*

Recommendation 12: *A REST binding for the Sensor Observation Service should be specified.*

4.2. Web Processing Service

As mentioned above, the OGC Web Processing Service standard is the base specification for providing geoprocessing facilities in the Web in an interoperable manner. In the pilot, the WPS has been used to provide an interface for executing the flood model in the Web.

Since geoprocessing functionalities may range from simple operators, e.g. intersecting polygons, to complex environmental models like run-off models, the WPS standard is kept generic and does not restrict the functionality and corresponding in- and outputs. Instead, the standard explicitly mentions that process profiles should be defined that restrict the functionality and process parameters.

[5] Simple Object Access Protocol (SOAP) is a network protocol for data exchange and remote procedure calls between distributed systems and is standardized by the W3C consortium. More information can be found at https://www.w3.org/TR/soap/.

Recommendation 13: *WPS process profiles and best practices should be defined that specify how to deploy hydrological models, e.g. run-off or flood models.*

The approach for profiling specific WPS processes may result in a large number of process profiles. How can one know whether there is already an existing process profile for the model he is planning to deploy? The current way is to take a look on the OGC website and check whether some of the standard documents contain profiles defined for the problem at hand. This is cumbersome and needs to be simplified in future in order to facilitate the adoption of profiles and to avoid duplicating profiles. We hence recommend creating a registry for process profiles.

Recommendation 14: *We recommend creating a registry for process profiles of the WPS standard that allows publishing of and searching for profiles.*

As already stated above, REST interfaces relying on the basic HTTP operations simplify the complexity of implementing Web clients accessing the data or, in case of the WPS, processing resources. We hence recommend also specifying a REST binding for the WPS.

Recommendation 15: *A REST binding for the OGC Web Processing Service should be specified.*

While the OGC WPS standard defines the Web interface for describing and accessing the geoprocessing facilities, the actual coupling with other OGC services and data sources may be realized in different ways. The approach chosen in the pilot is illustrated in Figure 3.

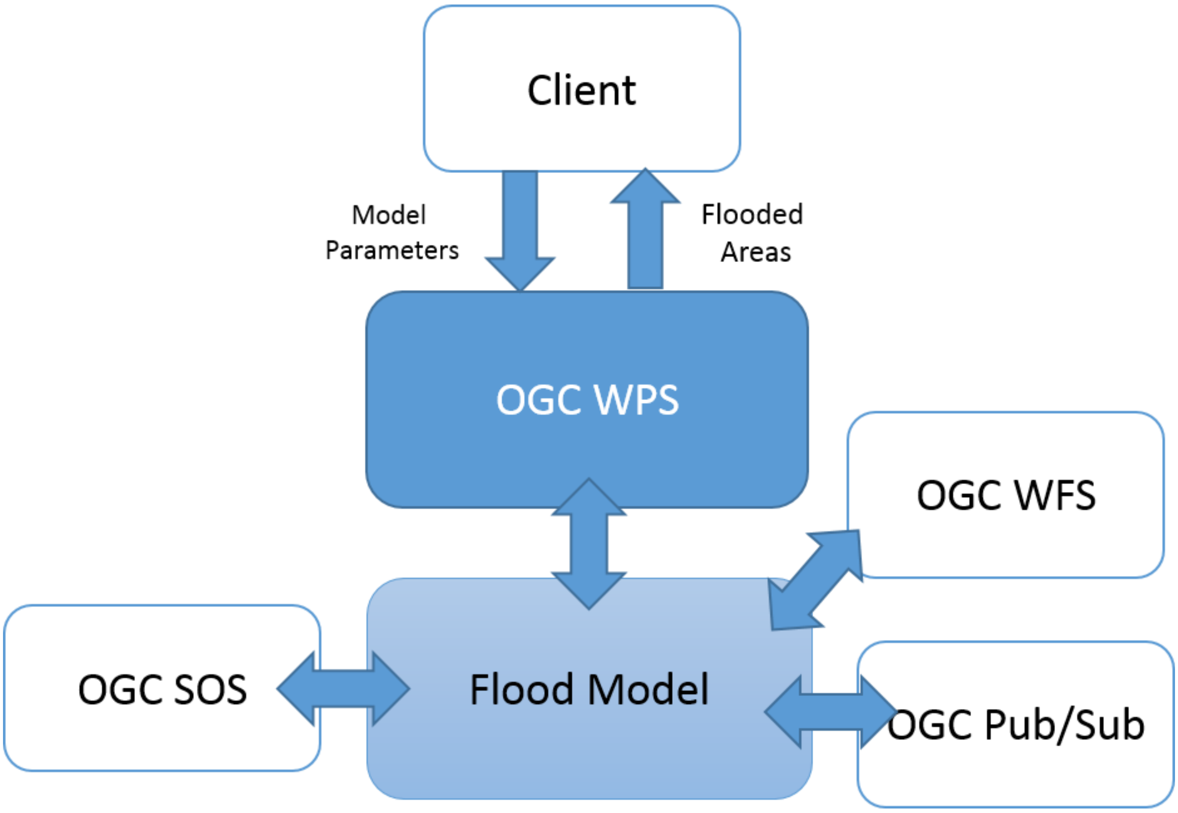

Figure 3: Tight-coupling of WPS process implementation with other OGC services

In this approach, the endpoints of the SOS providing input data and of the Publish/Subscribe and WFS where the data is pushed to are pre-defined in the model implementation and cannot be defined by WPS clients. The advantage of this approach is that less complexity is needed on the client side, as the client does not need to know about SOS, Publish/Subscribe and WFS. However, it also has drawbacks since the approach is less flexible and a new WPS needs to be set up in case another SOS should be used as data source. We consider this approach useful for models that are highly specialized for specific regions and environments and cannot easily be applied to other regions and data sources.

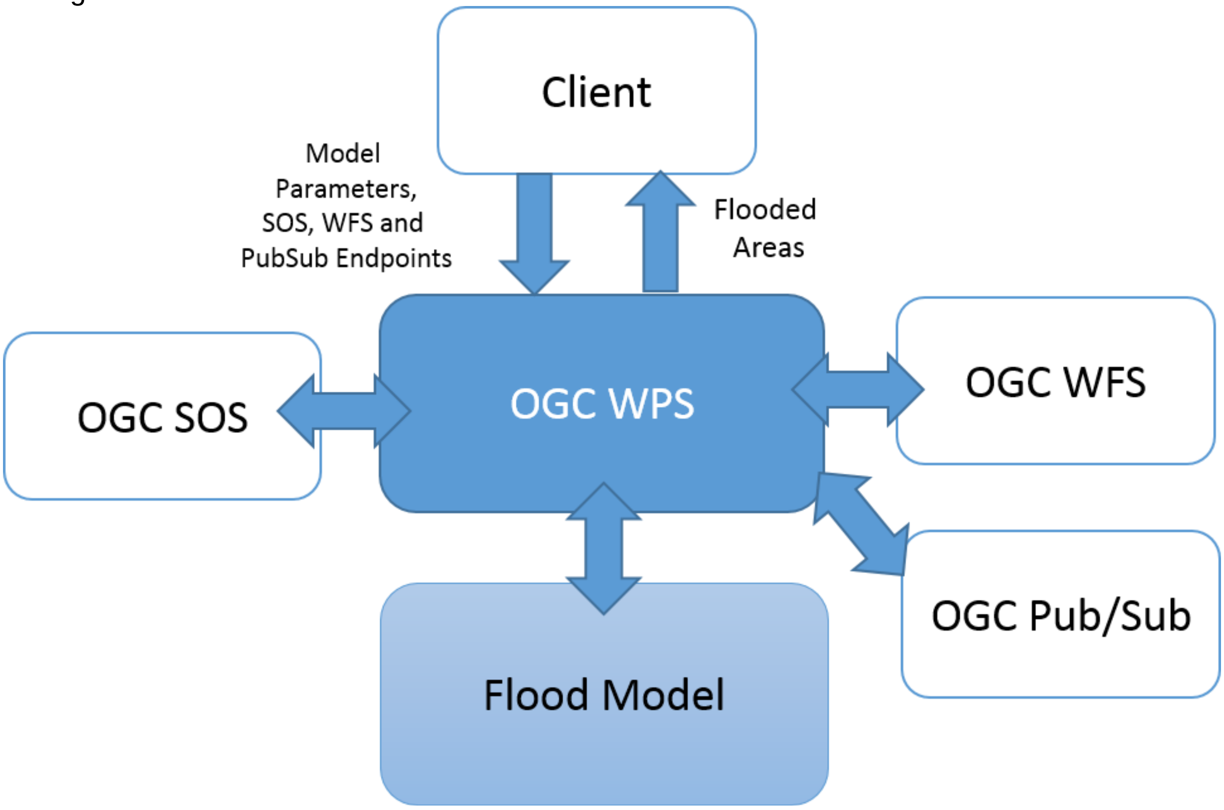

Figure 4: Loose coupling of WPS process implementation with other OGC services.

The other option would be that the endpoints of other OGC services are defined by clients and can be passed in an execution request for a specific process (Figure 4). This has the advantage that a process can be easily applied to other data sources and its results can be distributed to endpoints defined by the client. As can be seen in the two examples, there is no one-fits-all solution. However, we consider developing best practices of how to deploy models in the Web using WPS as a necessary next step in order to ease the deployment and provide guidance.

Recommendation 16: *We recommend developing best practices for coupling geoprocessing (analysis) facilities provided by WPS with other OGC services that provide input data or are used for storing the outputs, e.g. WFS or SOS.*

4.3. Publish/Subscribe

For allowing clients to subscribe for notifications, if a point of interested is affected by a predicted flood area, a RESTful Publish/Subscribe component has been implemented that is based on the recently published OGC Publish/Subscribe specification (Braeckel et. al. 2016). The current core specification defines the core functionality independently of the concrete binding technology (e.g., SOAP, REST, and KVP). Besides the Publish/Subscribe standard, two services were specified for providing publish/subscribe patterns for OGC Sensor Web standards, the Sensor Alert Service and the Sensor Event Service. Since both standards did not become official OGC implementation standards and the Publish/Subscribe component has been specified to harmonize the Publish/Subscribe interfaces for all OGC standards, the usage of the Publish/Subscribe standard was chosen in the pilot.

Extensions to the core specification are used to add specific binding technologies to this specification. Currently, the Publish/Subscribe specification only provides a binding extension for the SOAP protocol (Braeckel & Bigagli 2016). SOAP, however, is sometimes very cumbersome and difficult to use, especially in Web-based client technologies such as JavaScript. For this reason, we consider the specification of other binding extensions for the Publish/Subscribe to be urgently required.

Recommendation 17: *Binding extensions for other binding technologies besides SOAP should be specified. In particular, we recommend a RESTful Publish/Subscribe binding due to its simplicity and widespread use.*

While most of the core concepts are well standardized, a common specification for publishing messages to a specific publication seems to be missing. For the purpose of interoperability, however, it is an essential requirement to know how messages get published over the Publish/Subscribe component. Especially, if the Publish/Subscribe interface is implemented as an independent component like in this pilot. Therefore, we recommend to add the specification of a publish operation to the OGC Publish/Subscribe specification.

Recommendation 18: *A publish operation should be specified within the OGC Publish/Subscribe specification.*

One of the core differences of the OGC Publish/Subscribe specification to other types of publish/subscribe-broker is that it allows different types of message delivery. The idea is to abstract from concrete message types by which subscribers can deliver messages. The Subscription type of the Publish/Subscribe specification contains information on individual publications, i.e. an identifier of a publication as well as optional filters. In addition, each subscription has to include information about the delivery method, the delivery location, and additional delivery related parameters. However, not every delivery method has an explicit delivery location, for instance, web-socket communication. In addition, delivery location and delivery parameters are of any type.

Recommendation 19: *The deliveryLocation attribute of a Subscription should be optional.*

Recommendation 20: *Profiles and best practices should be defined for specific delivery methods, e.g. for SMTP or Web sockets, that restrict the generic delivery parameters to match the delivery method.*

Furthermore, each subscription can only subscribe with a single delivery method. Many use-cases, however, may require multiple ways of notification (e.g. additional email notifications). In this way, it is always required to have multiple subscriptions, one for each required delivery method. This increases the client-side complexity of managing subscriptions and also decreases the performance of the broker. Thus, we recommend to allow multiple delivery methods within the Subscription type.

Recommendation 21: *The Subscription type defined in the Publish/Subscribe specification should allow multiple delivery methods.*

Filters as well as content types are kept generic in the Publish/Subscribe core specification and not further restricted. In the pilot, a spatial filter (Point-in-Polygon) was defined using the Filter Encoding Specification (FES) and this filter applied to the geometries returned in the GML file of the Web Processing Service (see Section 4.3 in D5.3 for more details). Since the definition of the filter is depending on the content type, we recommend defining profiles for the Publish/Subscribe standard that restrict the content type and define how filters on the content can be defined.

Recommendation 22: *Profiles and best practices of the OGC Publish/Subscribe specification should be defined for filters on specific content types.*

5. Conclusions

The pilot demonstrated that the usage of open standards eases integration of different data sources and allows deploying different flood models in the Web using the same standardized interface. These standards thus avoid technical (syntactical) as well as semantic conflicts. However, a pre-requisite is that data and models are open and accessible. Since observing and modelling our environment is a complex task, the standards for representing the observations and models are of a complex and generic nature. We hence recommend hiding the complexity in the user interfaces, to define profiles of these standards for the hydrology domain like already done with WaterML 2.0 and the SOS Hydrology profile, and to simplify the implementation of Web clients by also providing lightweight REST bindings for the service interfaces instead of the heavy-weight SOAP bindings. Since humans are creatures of habit, we consider it crucial to provide extensions for supporting these standards for their commonly used tools in order to support broad adoption of these standards enabling interoperability.

6. Bibliography

Andres, Volker, Simon Jirka and Michael Utech, eds. (2014). OGC Sensor Observation Service 2.0 Hydrology Profile. OGC 14-004r1. Open Geospatial Consortium Inc. Accessed 05 December 2016. http://docs.opengeospatial.org/bp/14-004r1/14-004r1.html

Braeckel, Aaron., Bigagli, Lorenzo, Echterhoff, Johannes, eds. (2016). OGC Publish/Subscribe Inteface Standard 1.0 – Core. OGC 13-131r1. Accessed 05 December 2016. http://docs.opengeospatial.org/is/13-131r1/13-131r1.html

Brackel, Aaron, Bigagli, Lorenzo, eds. (2016). OGC Publish/Subscribe Interface Standard 1.0 SOAP Protocol Binding Extension. OGC 13-133r1. Accessed 05 December 2016. http://docs.opengeospatial.org/is/13-133r1./13-133r1.html

Bröring, Arne, Christoph Stasch, and Johannes Echterhoff, eds . (2012). OGC Sensor Observation Service Interface Standard. OGC 12-006. Open Geospatial Consortium Inc. Accessed 12 December 2012. https://portal.opengeospatial.org/files/?artifact_id=47599

Cox, Simon, ed. . (2007). Observation and Measurements - Part 1 - Observation Schema. OGC 07-022r1. Open Geospatial Consortium Inc. Accessed 22 March 2011. http://portal.opengeospatial.org/files/?artifact_id=22466

Fielding, Roy Thomas. (2000) Architectural styles and the design of network-based software architectures. Ph.D. thesis. University of California, Irvine.

Müller, Matthias, and Benjamin Pross, eds. (2015) OGC WPS 2.0 Interface Standard. Open Geospatial Consortium Inc. Accessed 05 December 2016. http://docs.opengeospatial.org/is/14-065/14-065.html

Pesquer L., Masó J., Stasch C., Jirka S., Arctur D. (2016) RiBaSE: A pilot for testing the web integration of water-related information and models. Geospatial Sensor Webs Conference 2016. August 29-31, 2016, Münster, Germany.

Pons X., 2000. MiraMon. Geographical Information System and Remote Sensing Software, CREAF. The Internet: http://www.miramon.cat

Reed C., Buehler K., McKee L. (2015) OGC consensus: How successful standards are made ISPRS International Journal of Geo-Information, 4 (3), pp. 1693-1706

Taylor, Peter, ed. (2014). OGC® WaterML 2.0: Part 1- Timeseries. OGC 10-126r4. Open Geospatial Consortium Inc. Accessed 05 December 2016. https://portal.opengeospatial.org/files/?artifact_id=57222

Vretanos, Panagiotis (Peter) A., ed. OGC Web Feature Service 2.0 Interface Standard – With Corrigendum. Accessed 05 December 2016. http://docs.opengeospatial.org/is/09-025r2/09-025r2.html

WaterInnEU project deliverable (2015) D5.3 Compendium of available standards. http://ddd.uab.cat/record/149981

7. Acronyms

API	Application Programming Interface
DEM	Digital Elevation Model
EU	European Union
FES	Filter Encoding Standard
GIS	Geographic Information System
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
ISO	International Organization for Standardization
INSPIRE	Infrastructure for Spatial Information in Europe
JSON	Java Script Object Notation
Publish/Subscribe	Publish/Subscribe
OGC	Open Geospatial Consortium
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
URL	Unified Resource Locator
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
XML	eXtensible Markup Language