Licensing OGC Geoprocessing Services as a foundation for commercial use in SDIs

Bastian Schäffer, Bastian Baranski
Institute for Geoinformatics
University of Münster
Münster, Germany
{schaeffer, baranski}@uni-muenster.de

Theodor Foerster
Geo-Information processing department
International Institute for Geo-Information Science and
Earth Observation, ITC
Enschede, the Netherlands
foerster@itc.nl

***Abstract*—This paper presents an approach for enabling the commercial use of OGC Web Processing Services in Spatial Data Infrastructures (SDIs). In particular, it is studied how standard Geoprocessing Services can be enhanced in order to support ad-hoc license agreements directly in-process, without any prior offline negotiated agreements being necessary between geoprocessing provider and geoprocessing user. A security enabled architecture including the description of interactions between the different entities is described. Additionally, a classification of potential licenses for Geoprocessing Services is provided. These licenses are understood as dynamic negotiation of access rights in contrast to classical role-based access control. Finally, the presented ideas are verified by a proof-of- concept implementation following a real world scenario.**

Keywords-Web Processing Service;, Licensin; Security; Software as a Service (SaaS)

I. INTRODUCTION

Geospatial Web Services organized in Spatial Data Infrastructures (SDIs) are designed to provide and share georesources (geodata and spatial models) across organizational and technical boundaries. In particular, Geospatial Web Services enable the access to remote georesources on-demand [1]. They are described on a technical level through standards for data encodings and service interfaces, such as established by the Open Geospatial Consortium (OGC). In the past, SDIs focused on data provision and data portrayal [2]. Recently, the additional integration of processing capabilities into SDIs through Geoprocessing Services has been a topic for research, as i.e. demonstrated by [3]. However, for sustainable use of SDIs as well as for commercial applications, the ad-hoc integration of processing capabilities and the accounting of on-demand use of Geoprocessing Services have to be considered.

This was the starting point to design a security-enabled architecture as a transparent layer for Geoprocessing Services. In particular, this paper will focus on state-of-the-art interface specifications of the OGC such as the Web Processing Service (WPS) [4] and define a generic security extension. On an abstract level, such an extension is technology-independent and leads to a common security-enabled architecture for OGC Web Services. But besides the technical challenge there is a legal barrier still in place, limiting especially the on-demand use of Geoprocessing Services in commercial applications. This is due to the fact that for commercial use, it is necessary to establish an agreement between georesource provider and georesource user regarding the terms and conditions of use regarding the specific geoprocess [5]. Currently, these agreements are still treated offline, typically in the form of a written license agreement and signed by all involved parties (such as it is common practice for licensing ArcGIS server). It is easily imaginable that this time-consuming way of licensing clearly contradicts the goal of seamless integration and agile interaction of Geospatial Web Services. This gap was already identified by the initiative for the Infrastructure for Spatial Information in Europe (INSPIRE), resulting in the demand for e-commerce services in the INSPIRE Directive, Article 14(4) [6].

This paper gives first an overview of general security requirements as well as security concepts in the context of Geospatial Web Services. This is followed by a twofold concept describing an abstract security layer and a classification of different licenses for Geoprocessing Services. At first, an abstract security layer for Geoprocessing Services is introduced which includes a description of how standard Geoprocessing Services can be enhanced in order to support ad-hoc license agreements directly in-process, without any prior offline negotiated agreements being necessary between geoprocess provider and geoprocess user. At second, different types of licenses are classified based on the structure of Geoprocessing capabilities such as the offering of different processes and Quality of Service parameters.

The introduced licenses are electronically established and are legally equivalent to paper-based licenses. They bridge the gap between the legal and the technical world by automatically defining aspects of licensing, such as access rights and price models. Therefore, they present a foundation for on-demand use of geoprocessing functionality as the next evolution step in SDIs.

II. Background

This section provides a review of related work in the context of web based geoprocessing and security and licensing.

A. *Web Processing Services*

The Web Processing Service (WPS) interface specification describes a standardized method to publish and execute web-based processes for any type of geoprocess. According to the WPS interface specification [4], a process is defined as any calculation operating on spatially referenced data.

In detail, the WPS interface specification describes three operations, which are all handled in a stateless manner: *GetCapabilities*, *DescribeProcess* and *Execute*. *GetCapabilities* is common to any type of OGC Web Service and returns service metadata. In case of WPS, it also returns a brief description of the processes offered by the specific service instance. To get more information about the hosted geoprocesses, the WPS provides process metadata through the *DescribeProcess* operation. This operation describes all parameters, which are required to run the process. Based on this information the client can perform the *Execute* operation upon the designated process. As every OGC Web Service, the WPS communicates through HTTP-GET and HTTP-POST based on an OGC-specific XML-message encoding. However, the WPS interface specification does not describe any aspect regarding licensing as it is designed in this article.

B. *Security for Webservices*

Web Service Security is a very wide area. Therefore only a very brief overview can be provided here. It can be distinguished into the following aspects in a Service Oriented Architecture (SOA) [7],[8]: Intrusion Detection, Integrity, Privacy and non-repudiation. In addition, services need to know who wants to access a resource (authentication) and if the entity is allowed to access the requested resource (authorization). In general, authentication describes the verification of an identity of an entity while authorization is commonly referred to as access control [9]. Licensing as described in [5] adds an ad-hoc notion to it and allows the dynamic negotiation of access rights in contrast to classical role-based access control [9] concepts. In the sense of [5] a license consists basically of a grand and an issuer. The grant includes a *Principal* as the entity to whom the right has been granted, a *Right* as the act associated to the right that has been granted and a *Resource* as the resource associated to the act above.

On a technical level, there are multiple mainstream IT standards available: The OASIS eXtensible Access Control Markup Language (XACML) [10] describes a basic security architecture, protocol and license encoding, which serves as the foundation for the presented concepts in Section III B. This can be combined with OASIS' Security Assertion Markup Language [11] as a common encoding for security tokens. On the message level, the OASIS WS-Security (WSS) [12] describes a secure message exchange that is used on a technical level to transport security tokens, encrypt and sign message. This specification is accompanied by WS-Trust [13] for managing trust and WS-Policy [14] for the encoding of preconditions. On the transport layer, protocols like TLS/SSL can be used.

III. Geoprocessing Licensing Concept

The concept for licensing Geoprocessing Services consists of a security architecture and a classification of licenses for Geoprocessing Services.

A. *Security Architecture*

The security architecture is based on a common policy-based XACML security architecture [10] and it incorporates findings resulting from research conducted in last OGC testbeds and a modified version of the OGC DRM reference model [5] for the incorporation of Geoprocessing services.

At first, a *static model* with components of the security architecture is presented. This is followed by an analysis of trust relationships between these components. Finally, the interactions between these components are presented in the *dynamic model*.

1) Static Model

The static model describes the components of the architecture. As depicted in Figure 1, two different domains exist. The client domain consumes on-demand a previously unknown Geoprocessing Service from the server domain.

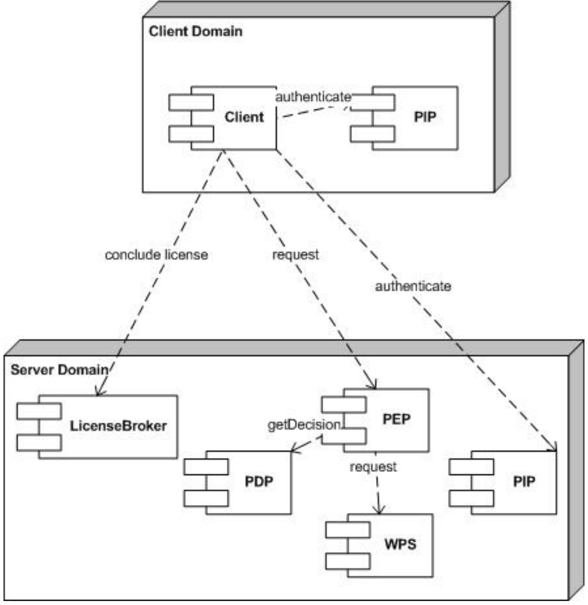


Figure 1. Overview of the architecture as an UML component diagram.

In detail, the client domain has a client component, which is responsible for invoking secured the Geoprocessing Service and gathering all required security tokens. Besides, the client domain holds a Policy

Information Point (PIP) to issue identity tokens for authentication purposes as described in principle in the XACML architecture [10]. The interface can be adopted from WS-Trust [13], which defines a Security Token Service (STS) for this task. Different identity token encodings are possible. In this paper we relied on SAML based identity tokens (see section II. B).

The server domain consists of a WPS, which performs the actual Geoprocessing tasks. This service is secured by a Policy Enforcement Service (Policy Enforcement Point, PEP) adopted from the common XACML security model [10] responsible for the enforcement of access decisions taken by the Policy Decision Point (PDP). The WPS instance can be easily configured (e.g. by means of a firewall) that it only allows access from the PEP's IP address. Thereby, any communication between the two domains has to go through the PEP security layer. Thus, the Policy Enforcement Point serves as a transparent component securing the WPS without touching the secured service. To be transparent to the client domain, it has the same interface as the secured WPS, but adds additional security functionality, such as issuing preconditions and enforcing usage rights. The decision about the usage rights is made by the Policy Decision Point (PDP) coming also from the common security model [10]. It takes the license, identity tokens and request and matches the license principal with the identity token subject and intersects the request georesource with the granted rights described in the license.

In addition, a PIP is provided in the server domain as a trusted issuer for identity tokens. The common security architecture is extended by a LicenseBroker for providing license templates, observing the negotiating process, issuing and managing licenses. An interface for such a LicenseBroker has been described in [15]. Section 3 defines specific license types for Geoprocessing Services served by such a LicenseBroker component.

2) *Dynamic Model*

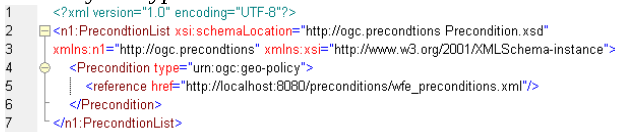
The dynamic model (Figure 2&3) shows the interactions between the different components presented in the static model. At first, the client requests the metadata of the secured WPS through the common GetCapabilities operation in order to discover the offered processes. The PEP intercepts this request and forwards it to the secured service. The returned metadata is then enriched with security preconditions (for example the need for a previously negotiated license for accessing the service).

Preconditions play a key role in security enabled web services and therefore also in Geospatial Rights Management (GeoRM) enabled OGC Web Services (OWS) as well. In general, preconditions publicly announce a potential web service requestor, which conditions (in the context of security-which security model, tokens, encryption mechanism, formats, trusted issuing services etc. are required / supported). This concept ensures interoperability by allowing services to fulfil all required preconditions prior to the service invocation.

To indicate such preconditions the Metadata response has to be used. Listing 1 shows a proposed solution by extending the GetCapabilities response. The existing *access constraints* element in a GetCapabilities response allows only plain text and no further xml elements according to schemas from OWS-Common. Therefore, we extended the metadata as shown in Listing 1.

A <*PreconditionList*> element which serves as a container for 1..n <*Precondition*> elements. Each of these elements has a mandatory *type* attribute, which contains a urn indicating the encoding of the document represented either by a <*value*> or by a <*reference*> element. In Listing 2 *urn:oasis:WS-Policy* indicates a widely accepted WS-Policy [14] encoding for the document referenced by *http:/localhost:8080/preconditions/wfe_preconditions.xml*.

Other encodings are also possible but have to be also indicated by the *type* urn.



```
<?xml version="1.0" encoding="UTF-8"?>
<n1:PrecondtionList xsi:schemaLocation="http://ogc.precondtions Precondition.xsd"
xmlns:n1="http://ogc.precondtions" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Precondition type="urn:ogc:geo-policy">
        <reference href="http://localhost:8080/preconditions/wfe_preconditions.xml"/>
    </Precondition>
</n1:PrecondtionList>
```

Listing 1. GetCapabilities precondition extension.

The client has to verify that it understands the preconditions encoding and that it can fulfill the required preconditions indicated in the received metadata. In terms of licensing Geoprocessing Services, preconditions require three elements: A license token for authorization purposes describing the granted rights, an identity token for authentication purposes and a digital signature of the requests for non-repudiation purposes. Other security mechanisms such as encryption are optional. The client has to understand the encoding of the preconditions such as the token type and the specific encoding of the different tokens. It also has to obtain and trust the identity and licensing token issuing services. For instance, in a WS-Trust encoding, the <*TokenIssuer*> element of a required security token presents the service endpoint, where the client can obtain a specific token.

As shown in Figure 2, the next step for the client is to obtain a license token and therefore request the capabilities metadata of the LicenseBroker, based on the URL described in the secured services preconditions. The LicenseBroker metadata is also extended with preconditions following the same pattern as described above. It describes where the required identity token can be obtained from (e.g. the <*TokenIssuer*> element in a WS-Trust encoding). Once the client has successfully verified the preconditions, it has to obtain an identity token from the given issuer service (e.g. STS). Figure 2 shows that the client first gets an identity token from its trusted PIP. The client details, such as public key information, is usually stored there in advance and it can authenticate itself via e.g. a predefined username and password using HTTP Basic Authentication. Since the clients PIP is usually not the same PIP as described in the preconditions of the secured service or the license broker preconditions metadata, the server domain would not trust it and reject the request

However, the PIP stated in the preconditions may accept identity tokens from the client's PIP for authentication, if both PIPs have a trust relationship. Therefore the client sends its I identity token to the servers PIP and if both services trust each other (based on i.e. WS-Trust), a identity token valid for the server domain is returned. This token can then be used for authentication in the server domain.

With this newly obtained identity token for the server's domain, the client can then negotiate and conclude the license with the LicenseBroker. Since the LicenseBroker trusts the identity token, the included public key in the identity token of the requesting client can be used to verify that the requestor is the same entity that signed and thereby issued the request to order a license. This allows the LicenseBroker to define the subject of the ID token as the principle in the license. Otherwise, it would be possible to conclude a potentially commercial license with a stolen identity token. Based on the negotiated license type (see section III.2), the resource and action section of the license are also filled.

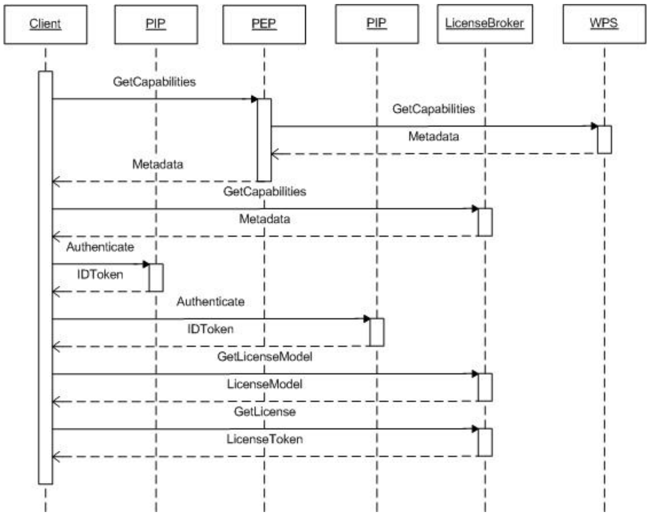


Figure 2. Dynamic Model-Preparation Phase as an UML sequence diagram

A reference to the license token is returned to the calling client which is then used together with the ID token for the actual secured service execute request (Figure 3). Again, the secured service's Policy Enforcement Point intercepts the request and verifies that the signature is valid. The public key for verifying this can be extracted from the attached identity token, which can be trusted because it is issued by the trusted issuer specified in the preconditions.

After verification of the identity of the requestor, the Policy Enforcement Point resolves the license and checks the signature as well. The license together with the identity token and the actual WPS execute request is sent to the Policy Decision Point, which checks that the subject of the request is the granted holder of the license rights (license principle).

Additionally, the Policy Decision Point checks that the requested georesource stated in the WPS execute request is equivalent to the one stated in the license (action and resource). In case all checks are successful, the Policy Enforcement Point is allowed to forward the request to the secured WPS. The results are upon termination returned back to the client.

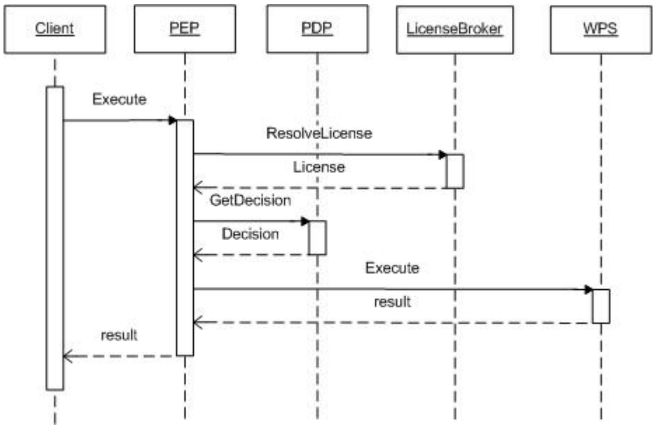


Figure 3. Dynamic Model-execution Phase as an UML sequence diagram

In case of an asynchronous request, which is supported by the WPS, only a reference encoded as a URL is returned back. This URL holds in that case a unique session key attached by the Policy Enforcement Point in order to only allow the dedicated recipient to resolve the correct georesource. Encryption is favorable in that case to ensure that no one can intercept the client server communication.

B. Geoprocessing License Classification

Geoprocessing Services have in some parts different capabilities compared to other Geospatial Web Services such as Geospatial Web Services for data delivery or discovery. This section introduces a classification of licenses especially for Geoprocessing Services. Such a classification is required to enable secure and on-demand license negotiation between georesource providers and clients for commercial applications. The proposed classification is based on a review of Geoprocessing capabilities and common QoS parameters. It extends the basic licensing concepts presented in [5] that are limited to data providing services.

In principle, web services in and in particular OGC Web Services have three levels. The service level, which is defined by the URL, the set of operations and the set of georesources provided by each operation.

Licensing can be applied to all three levels. On a service level, granting entitled subjects access to the whole service and all its operations and georesources as shown in figure 4.

One level below on the operations level, the licenses need to be acquired to access a certain operation with access to all georesources provided by this operation.

The next level is the georesource level. In case of Geoprocessing Services, the general access to offered processes could be restricted. For instance, one could obtain a license for process *buffer* but not for any other processes offered by a specific Geoprocessing Service. This could be designed in a more fine-grained fashion, so that a license is granting access to a specific georesource but with restrictions on the parameter range. Typical WPS processes allow the input of data and additional parameters for tweaking the process. For instance, a license which grants a subject general access to a buffer process, but only allows a distance parameter in the interval of [0.0, 10.0]. For a grater range, another (potentially more expensive) license needs to be acquired, because the computations are more extensive.

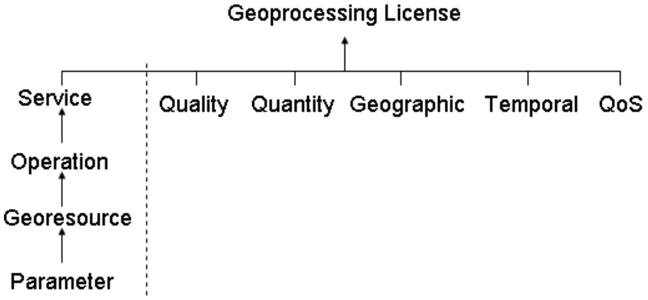


Figure 4. License Hierarchy

This classification can be combined with Quality, Quantity, Temporal and Geographic license aspects regarding the data that is delivered or processed. Additionally, general Web Service quality (Quality of Service (QoS) aspects and resource requirements for processing huge amount of data could be addressed in the license terms.

Quality aspects are the definition of a resulting data quality, such as a specific image resolution. This is specific for Geoprocessing Services, since the processing of input data can result in different quality levels.

Quantity aspects relate to the quantity input and output data, such as the number of features or pixel being processed. The computation, and therefore the amount of computational resources such as memory or CPU cycles is highly dependent on the quantity of input data. Licenses for different quantities can therefore be an important means to define adequate cost models.

Temporal aspects describe a restriction on either the settlement of the computation (e.g. after 11 p.m.) or the duration of the computation (e.g. max. 3 hours). It is imaginable that it may be less expensive to request a computation in non-busy timeslots rather than in high demand phases.

Geographic aspects are not limited for Geoprocessing services, but also apply to it. For instance, a surface simplification algorithm over mountainous terrain requires a different license then the same algorithm over non mountainous terrain. General Quality of Service (QoS) parameters for Web Services are described in [16]. These service quality parameters may include availability, accessibility, performance, scalability, capacity of Web Services and other network-related QoS requirements.

Furthermore, the overall performance of Geoprocessing Services depends on the algorithm mightiness, the amount as well as complexity of input data and the operational availability of computational resources like the number of available CPUs and disk space. In [17] the resource requirements of jobs for submission to Grid Computing infrastructures are described. The defined resource requirements also apply to processing services, such as a license which grants the availability of 3 CPUs for a time frame of 6 hours.

IV. Case Study

This section presents a real-world scenario, which illustrates the presented ideas and acts as a proof-of-concept implementation based on open-source componments published through 52°North. The scenario is constructed around the ad-hoc acquisition of Geoprocessing functionality. The Douglas-Peucker simplification algorithm [18] is used in this case to simplify complex road geometries in the north-west of Spain. As described in Section 3, a client has to explore the existing functionality and security preconditions. Since we will use OpenLayers as an OTS client, which does not come with any security capabilities, a proxy façade is used on the client side to first negotiate and conclude the license. Finally, this license is presented as an endpoint to the OpenLayers client.

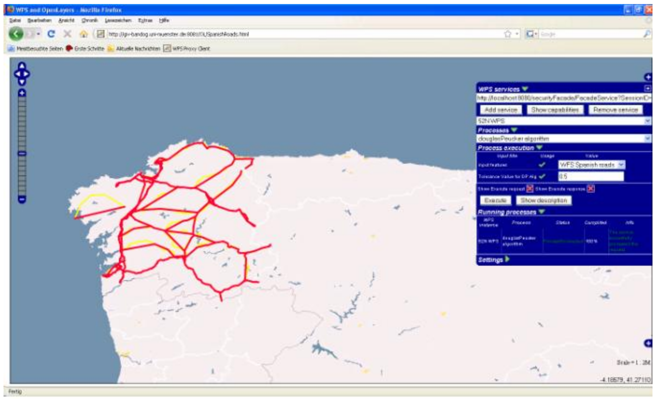
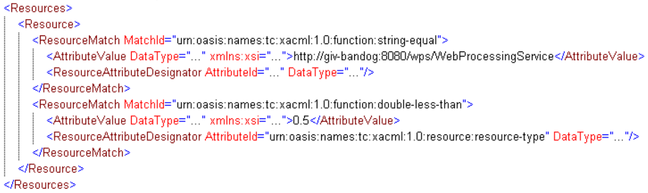


Figure 5. Open Layers client showing the result of an execute request to a secured WPS.

As presented in Section 3.3, different licenses are possible for Geoprocessing Services. In this case, a license is selected, which offers specific quality parameters for a Douglas-Peucker algorithm. A simplification ratio as described by [19] is used as a Quality parameter regarding the level of detail of the output data. The simplification ratio allows the user to choose any value between 0.0 (no simplification) and 1.0 (full simplification). Listing 2 shows a snippet of the full license describing the licensed simplification ratio encoded in XACML.

The algorithm has two inputs: a feature dataset and the simplification factor. After the license and an identity token are acquired by the proxy façade, the user can use OpenLayers to execute the process with an input layer and a specified simplification ratio following the protocol of Section III A 2).



```
<Resources>
  <Resource>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="..." xmlns:xsi="...">http://giv-bandog:8080/wps/WebProcessingService</AttributeValue>
      <ResourceAttributeDesignator AttributeId="..." DataType="..."/>
    </ResourceMatch>
    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:double-less-than">
      <AttributeValue DataType="..." xmlns:xsi="...">0.5</AttributeValue>
      <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-type" DataType="..."/>
    </ResourceMatch>
  </Resource>
</Resources>
```

Listing 2. Resource section of license

On the server side, the PDP has to verify that the request has a simplification ratio that matches the granted one in the license and that the identity of the requestor is identical to the subject in the license. Figure 5 shows the resulting geometries as a new layer (red) in OpenLayers. A simplification ratio of 0.5 was licensed, which yielded a reduction of the number of intermediate points compared to the original layer (yellow).

V. Conclusion

This paper describes an approach for enabling the commercial use of OGC Web Processing Services. An architecture including the interactions between the different entities is described. On this basis, a classification of licenses for Geoprocessing Services is given. The presented concept is successfully validated via a proof-of-concept implementation using a quality parameter based license. However, the approach is designed to be extensible in order to combine different types of licenses, it is nevertheless out of scope for this paper to discuss specific price model. When price models come into play, accounting in and billing have to be regarded that potentially could be performed by the LicenseBroker.

On a technical level, is was shown, that only a license, identity token and a signature are necessary to enable licensing for geoprocessing services. Besides, the presented architecture can be applied to existing software without touching the non-security enabled backend services.

In general, the presented twofold concept can be seen as an important step forward towards commercial applications in SDIs: Transition from the classical licensing model of GIS packages (e.g. ArcGIS desktop or server) to accountable on-demand processing services. The presented concept allows automated systems to negotiate licenses and to establish trustful Web Service interaction in an ad-hoc way. Thereby the identified gap of missing processing capabilities in SDIs will be solved also in a commercial dimension. This means that for the future, we can foresee, that no longer full GIS packages might be sold, but rather specific Geoprocessing functionality is dynamically licensed in an on-demand fashion.

In particular, the introduced concept is also valid for the emerging trend of Software as a Service (SaaS) [20] realized by Cloud Computing models [21]. In this context, GIS functions can be delivered in the SaaS way in an on demand fashion with a strong commercial perspective. This leads exactly to the definition of Cloud Computing [21]. Therefore, the presented concepts can be seen as a foundation for commercial and sustainable use of Web Processing Services in the future and potentially cloud enabled SDIs.

References

[1] J., McLaughlin and R., Groot, Geospatial data infrastructure: concepts, cases and good practice, Oxford:University Press, 2000

[2] C. Kiehle, K. Greve and C. Heier, "Standardized Geoprocessing– Taking spatial data infrastructures one step further," in Proceedings of the 9th AGILE International Conference on Geographic Information Science, 2006, pp. 273-282.

[3] Brauner, J., Foerster, T., Schaeffer, B. and Baranski, B., "Towards a Research Agenda for Geoprocessing Services", In J. Haunert, B. Kieler, & J. Milde (Eds.), 12th AGILE International Conference on Geographic Information Science, 2009, Hanover, Germany. Available: http://www.ikg.uni-hannover.de/agile/fileadmin/agile/paper/124.pdf

[4] OGC, OpenGIS Web Processing Service, 2007. Available: http://portal.opengeospatial.org/files/?artifact_id=24151

[5] OGC, GeoRM ReferenceModel, 2007. Available: http://portal.opengeospatial.org/files/?artifact_id=14085

[6] EU, Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 Establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union, L108, 2007.

[7] D. G. Firesmith, "Specifying Reusable Security Requirements" *Journal of Object Technology*, vol. 3 no. 1 (Jan-Feb 2004), pp. 61-75.

[8] M., Hafner and R., Breu, Security Engineering for SOA, Berlin, Heidelberg: Springer, 2009.

[9] D. F., Ferraiolo and R. Kuhn, "Role-Based Access Control," *Proceedings of the 15th NIST-NSA National Computer Security Conference*, Baltimore, MD, Oct 1992, pp. 13-16.

[10] OASIS, eXtensible Access Control Markup Language (XACML), 2000. Available: http://docs.oasis-open.org/xacml/2.0/XACML-2.0-OS-NORMATIVE.zip

[11] OASIS, Security Assertion Markup Language (SAML), 2005. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[12] OASIS, Web Service Security (WS-Security), 2006. Available: http://www.oasis-open.org/specs/index.php#wssv1.0

[13] OASIS, Web Service Trust (WS-Trust), 2007. Available: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html

[14] W3C, Web Service Policy (WS-Policy), 2007.

[15] OGC, License Broker Engineering Report, 2008. Available: http://portal.opengeospatial.org/files/?artifact_id=28162&version=2

[16] K. Lee, J. Jeon, W. Lee, S.-H. Jeong and S.-W. Park, "QoS for web services: Requirements and possible approaches", *Tech. rep.*, W3C, Web Services Architecture Working Group, 2003. Available: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/

[17] A. Anjomshoaa, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva., "Job Submission Description Language (JSDL) Specification", Version 1.0. Open Grid Forum, Grid Resource Allocation Agreement Protocol Working Group, 2007.

[18] D. H., Douglas and T. K., Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its

caricature", in The Canadian Cartographer, 10(2), 1973 pp. 112-122.

[19] T. Foerster, J. E. Stoter, B., Koebben and P., van Oosterom, "A Generic Approach to Simplification of Geodata for Mobil Applications", in M., Wachowicz and L., Bodum (Eds), AGILE 2007, Aalborg University, Denmark, 2007.

[20] M. Turner, D. Budgen and P. Brereton, "Turning software into a service",*IEEE Computer*, 36(10), Oct 2003.

[21] National Institute of Standards and Technology (NIST), Cloud Computing. 2009. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc