OPENGIS PROJECT DOCUMENT 05-036

TITLE:	**GeoXACML, a spatial extension to XACML**	
AUTHOR:	**Name:**	**Dr. Andreas Matheus**
	Address:	**University of the Federal Armed Forces Germany** **Werner-Heisenberg-Weg 39** **D-85579 Neubiberg, Germany**
DATE:	**June 16, 2005**	
CATEGORY:	**Discussion Paper**	

1. Background

Different initiatives of Geospatial Data Infrastructures (GDI) exist to provide interoperable access to distributed and heterogeneous geodata. One possible implementation of a GDI can be based on services, e.g. Web Services that communicate over a network like the Internet. The deployed services support interoperable access by implementing open standards – among others – of the Open Geospatial Consortium (OGC). The most famous examples of these services are the OGC' Web Map Service (WMS) and the Web Feature Service (WFS).

For commercial use of a service-based GDI, different security mechanisms must extend the basic infrastructure in order to ensure communication confidentiality and integrity as well as message authenticity and service availability.

In addition to these security requirements, access control is important in order to enforce restricted access to protected geodata. The prerequisite of access control is authentication, as it allows the proof of claimed identities to which access rights are associated. The proof of identity is guaranteed by a three-factor authentication mechanism: (1) What you have (e.g. an X.509 certificate), (2) What you know (e.g. a PIN or username/password) and (3) What you are (e.g. biometrics like a fingerprint). Depending on the requirements for proving the claimed identity, one, two or all three factors must be present.

Access Control in itself just controls, what user can access which resource (e.g. features or a map) using a particular operation. Different standards exist to establish access control for various requirements. However, no standard exists that allows the declaration and enforcement of access restrictions for the geospatial problem domain.

This document describes the profiling of the OASIS standard XACML in order to support the declaration and enforcement of access rights, based on GML feature types, features and the spatial relationship between a feature and a given (restriction) geometry. This extention – called GeoXACML – uses the XACML extension points to define the required functionality. This document also gives examples for access restrictions and how to be declared in GeoXACML.

Because GeoXACML is specific to the geospatial problem domain, a standardization with OASIS is not possible. Therefore, this document is posted to the OGC community in order to provide a possible recommendation, how to declare and enforce access rights for object-oriented geodata in an interoperable way.

2. References

[GML2]	Geography Markup Language (GML2.1.2), Open Geospatial Consortium, 2002
[Matheus-1]	Declaration and Enforcement of Access Restrictions for Distributed Geospatial Information Objects, Dissertation, Technische Universität München, 2005
[Matheus-2]	Authorization for a Service-based Geospatial Data Infrastructure, Presentation at the 52nd OGC Technical Meeting, New York City, 2005
[OASIS]	eXtensible Access Control Markup Language (XACML) Version 1.0, 18 February 2003
[OGC 99]	OpenGIS Simple Features Specification For SQL, Open Geospatial Consortium, 1999
[OGC 02]	Topic 12: OpenGIS Service Architecture, Open Geospatial Consortium, 2002
[Vivid]	Java Topology Suite Version 1.4 , Vivid Solutions, November 4, 2003

3. Proposal

This OGC document proposes one possible solution for the declaration and enforcement of access restrictions for object-oriented geodata, available through a Service-based Geo Data Infrastructure. It is the intension of the author to motivate the requirement for such an access control, give a problem statement, discuss an alternative approach and describe the solution, based on GeoXACML.

3.1 Motivation

Today, a typical Service-based Geo Data Infrastructure (GDI) does not provide a fine-grained access control mechanism. In figure 3.1, a simple example of a Service-based GDI is shown. It consists of a user (*Alice*), a client and three services:

- Institution A provides a Web Map Service for mapping the resource *Satellite Images*
- Institution B provides a Web Map Service for mapping the resource *Road* and *HWY*
- Institution B provides a Web Feature Service for accessing the resource *Building*

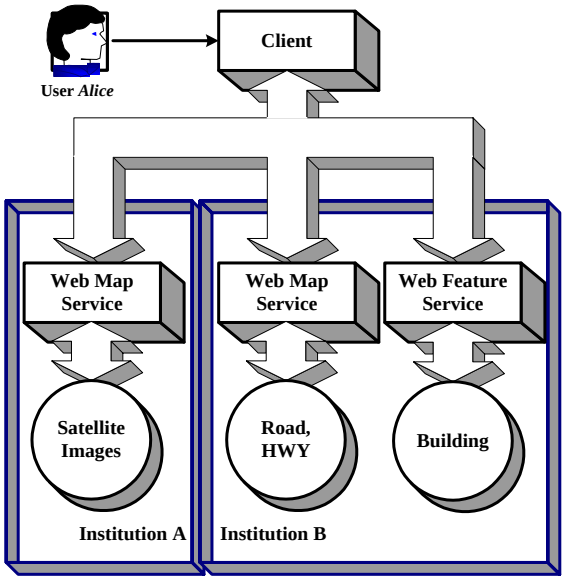

Figure 3.1: A simple Service-based GDI

Typical use-case: Let's assume that *Alice* requests a composite map, consisting of *satellite images* from *Institution A* and *roads* from *Institution B*. Because the services are interoperable, as implementing the OGC's Web Map Service interface, this is possible.

Access Control use-case: Let's assume that *Institution A* restricts the mapping of *satellite images*. Alice, can still request a composite map of the *satellite images* and the *roads* as an authorized user to the Enforcement Service WMS of institution A, if the access control system provides interoperable access. This can be achieved by the architecture, as shown in figure 3.2.

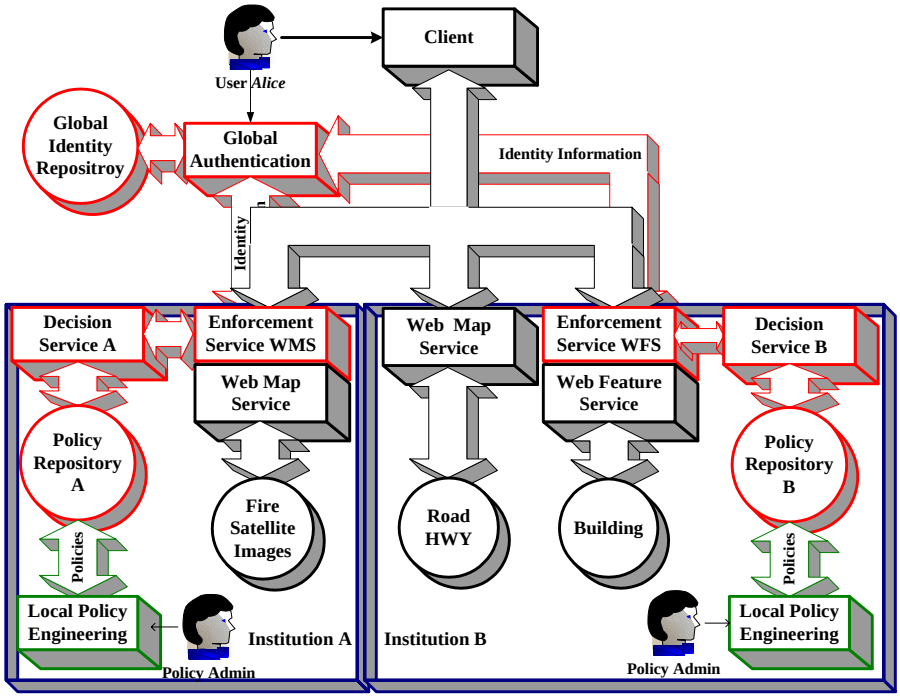

Figure 3.2: A Service-based GDI, extended by an interoperable access control

In figure 3.2 is shown a possible service-based GDI with access control. The components from the example GDI of figure 3.1 are colored in black. The components that belong to the runtime environment (enforcement and authorization of declared of permissions) of the infrastructure are colored in red and the administrative components for the declaration of permissions are colored in green.

The interoperable access to the restricted resources is guaranteed by the specific architecture of the Enforcement Services, as they façade the Web Map and Web Feature Services. Each Enforcement Service provides the same interfaces as the façade service. For the example in figure 3.2, the Enforcement Service WMS functions as a WMS and the Enforcement Service WFS functions as a WFS (see figure 3.2). The façade and the service can be seen as a cascade, where the façade processes authorization related information and obtains the result for the actual requests from the cascaded geodata service. Typically, the façade resides in the de-militarized zone and the geodata service resides in the local network.

Even though it is not the intension of this document to address authentication issues, the infrastructure above shows an authentication for single-sign-on as it is a pre-requisite for distributed access control. One possible implementation of this authentication can be implemented by using X.509 certificates and Security Assertion Markup Language[1] (SAML) compliant identity information exchange.

3.2 Architecture Discussion

The introduced architecture, as shown in figure 3.2, separates the enforcement and authorization in two different services. This allows a centralized storage of permissions in a Policy Repository. One Decision service can derive authorization decisions for multiple Enforcement Services by obtaining the declared permissions from that central repository.

[1] SAML is an open standard by OASIS.

In contrast to the proposed architecture of figure 3.2 and the declaration of GeoXACML permissions, the enforcement of access restrictions can be obtained by setting up individual services. The service internal data structuring and the data available is then set up in such a way that it is compliant to existing access restrictions.

For example, a WMS provides a topological map for Germany. Based on the access restriction that only Alice and Bob can request maps for the administrative area of Bavaria, a specific WMS can be set up that provides the topological map just for the area of Bavaria. For all other areas, the map is white. If Alice and Bob are authorized users to the service, no further authorization decision must be derived. Such an infrastructure has the advantage that no authorization decision must be derived, based on actual request parameters. Even though this architecture saves the processing of an authorization decision, it has two major drawbacks:

1. The setup of a service depends on the existing access restrictions. If the restrictions change, e.g. Alice may request maps for Bavaria and Baden-Wurtemberg, a new service must be setup, or the existing service must be modified. For many different access restrictions, this results in a large set of specific services, which must be maintained and even more important must enforce – as a hole – the intended restrictions. To maintain the permission consistent services becomes more and more error-prone, the more different restrictions exist or the more the existing permissions change.
2. The updating of the data that is provided by the services must take place in such a way that all provided data sets are in sync. This means that if at one service the topological map of Bavaria is updated, the same geodata may be provided by another service or services. Therefore, these services must be updated at the same time in order to guarantee synchronized data. This management of synchronous updates is a very complex task that must be supported in an error-free way.

The three main advantages of the proposed approach are that

1. … existing services can stay untouched and the existing updating procedures do not require modification. As shown in the architecture of figure 3.2, the principle "embedding without touching" is realized.
2. … inconsistency tests for the declared permissions can be performed, based on the central Policy Repository. This ensures an error-free enforcement of declared permissions.
3. … the modification of existing restrictions is kept separate from the data service as it requires to modify the declared permissions. In order to not influence the existing Decision Service, a mirror for the Decision Service and the Policy Repository can be used to make the changes. After performing consistency checks and other relevant tests, the mirrored Policy Repository can be made active.

The disadvantage of the approach ist that an authorization decision must be derived for each request. Therefore, the overall response time increases by the processing time to derive the authorization decision. The main factors, influencing the processing time of deriving the authorization decision are the complexity of the permissions, their structuring and the number of features in the **`<ResourceContent>`** element.

3.3 Access Control Requirements

Based on a poll at the Intergeo in 2002 about the access control requirements for a service-based GDI, the following requirements – assuming an object-oriented data model – have been addressed:

(I) It is not sufficient to restrict the access to the entire service (coarse-grained restriction), as it allows an authorized user to access all provided geodata.

(II) (**Class-based** restrictions): It must be possible to restrict access to all resource objects of the same class.

For a WMS, this requirement refers to the capability to restrict the map access for each individual layer or for a set of layers. For the feature-info request, this requirement refers to the capability to restrict the access for a feature type or a set of feature types.

For a WFS, this requirement refers to the capability to restrict the access for a feature type or a set of feature types.

(III) (**Object-based** restrictions): It must be possible to restrict access to individual objects of a class.

There is no equivalent for the WMS map access. For the feature-info request, this requirement refers to the capability to restrict access for individual features.

For a WFS, this requirement refers to the capability to restrict access for individual features (one feature or a set of features).

(IV) (**Spatial** restrictions): It must be possible to restrict access based on the resource geometry if a certain spatial relation to a given geometry exists.

For a WMS, this requirement refers to the capability that maps can be requested, if the area of interest (expressed by the parameter BBOX) is, e.g., within[2] of the restricted area.

For the WFS, this requirement refers to the capability that features can be accessed if being, e.g., within a given area or touching[3] a given line.

(V) It must be possible to declare **positive and negative permissions** in order to express restrictions like, user A can read all features of type Building, but not the building, named "The White House".

(VI) It must be possible to **combine the class-based, object-based and spatial permissions**. As an example, user A can read all features of type Building within the area of Washington D.C.[4], but not the building named "The White House".

3.4 Example Access Restrictions

This section gives example access restrictions in descriptive text. Each of these examples is declared in GeoXACML and listed in Appendix C. Please note that the examples do not cope with the full functionality of GeoXACML. In particular, no combination of class-, object-based and spatial permissions is used.

3.4.1 WMS examples

Common to all examples, describing restrictions for a WMS map access is that the structure of the resource content is described by a valid GML feature collection (WMSTemplate.xsd), according to the GML application schema as listed in Appendix A. Please note that the template is a proposed structure that can be altered if necessary.

Declaration and enforcement of access restrictions for the feature-info interface is similar to the WFS interface for the operation read.

3.4.1.1 Class-based restrictions

P1a: *Alice* can *map* layer *COASTL*

P1b: *Bob* can *map* all layers, except the layer *COASTL*

P1c: All other requests are denied

3.4.1.2 Object-based restrictions

No example is defined for the object-based restriction, because it is similar to the WFS examples, using the operation read.

3.4.1.3 *Spatial restrictions*

P2a: *Alice* can *map* layer *COASTL within* the area *3 0,6 1,6 5,1 5,0 2,3 0*, expressed in the CRS *foo*

P2b: All other requests are denied

---

[2] All possible spatial functions are listed in table 4.2.

[3] All possible spatial functions are listed in table 4.2.

[4] For the declaration of the permission, the representing geometry (gml:Polygon) is being used.

3.4.2 WFS examples

Common to all examples, describing restrictions for a WFS access is that the structure of the resource content is described by a valid GML feature collection (CityModel.xsd), according to the GML application schema as listed in Appendix B.

3.4.2.1 Class-based restrictions

P3a: *Alice* can *read* all features of type *Building*

P3b: *Bob* can *write* all features of type *Intersection* and can *read* all features of type *Building*

P3c: All other requests are denied

3.4.2.2 Object-based restrictions

P4a: *Alice* cannot *read* the feature building, as it is identified by the fid "*The White House*"

P4b: All other requests are denied

3.4.2.3 Spatial restrictions

P5a: *Alice* can *read* all features of type *Building*, if within the area *3 0,6 1,6 5,1 5,0 2,3 0*, expressed in the CRS *foo*.

P5b: All other requests are denied

4. Profiling XACML, version 1.0

Most of the above requirements are supported by XACML in a native way. However, XACML does not support the requirement (IV): the declaration and enforcement of spatial restrictions. In order to support that requirement, the XACML extension points can be used for declaring new geometry attributes and condition functions, which support the validation of a certain spatial relation. The GML2 simple geometries (see [GML2]) are being used to define geometry attributes. The functions for testing spatial relations are taken from [OGC 99, section 2.1.1.2: Methods for testing Spatial Relations between geometric objects][5]. These functions are implemented in the Java Topology Suite (see [Vivid]).

Before defining the GeoXACML extension, the basic capabilities of extending XACML is introduced.

4.1 Extending XACML

The proposed spatial extension of XACML, named GeoXACML, is based on the open standard XACML, version 1.0 of OASIS (see [OASIS]). It was developed during the dissertation project (see [Matheus]).

For a detailed description of the eXtensible Access Control Markup Language (XACML), the reader is referred to the original documents from OASIS (see [XACML]). This section shows the XACML extension points for attributes and functions, as they provide the hooks for the spatial extension.

XACML's definition of policy file structure and the constructs to be used can be found in the XML-Schema file, named cs-xacml-schema-policy-01.xsd.

4.1.1 Extending data-types for attributes

Each **<AttributeValue>** element has a **DataType** attribute that holds a valid URI, which defines the syntax and semantics of the attribute.

```
<xs:element name="AttributeValue" type="xacml:AttributeValueType"/>
<xs:complexType name="AttributeValueType" mixed="true">
  <xs:sequence>
```

[5] The spatial function *Relate* is not being used.

```
    <xs:any namespace="##any" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Listing 4.1: XACML schema definition of the **<AttributeValue>** element

For XACML, only unstructured **<AttributeValue>** elements are defined (see [XACML], section 10.2.7 on page 87). One XACML **<AttributeValue>** example, defining the String ***Alice*** is shown by the following code segment:

```
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
```

4.1.2 Extending functions

Each **<Function>** element has a FunctionID attribute that holds a valid URI, which defines the syntax and semantics of the function.

```
<xs:element name="Function" type="xacml:FunctionType"/>
<xs:complexType name="FunctionType">
  <xs:attribute name="FunctionId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

Listing 4.2: XACML schema definition of the **<Function>** element

For XACML different **<Function>** elements are defined (see [XACML], section 10.2.8 on page 87). One XACML **<Function>** example, as it is being used in the **<Condition>** element below, is the string-comparision function:

```
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

4.2 GeoXACML

This section defines GeoXACML, version 1.0, profiling OASIS' XACML, version 1.1.

4.2.1.1 Defining structured attributes

According to the extensibility, as defined in the non-normative section 8 of the XACML, version 1.1 specification, a structured **<AttributeValue>** can be defined by flattening the structured **<AttributeValue>** to a string, using the ASCII encoding. This approach – as it is non-normative – is not being used for GeoXACML. Instead, the 'regular' XML structuring is being used.

In that sense, the definition of a geographic **<AttributeValue>** is achieved by

1. assigning the appropriate value to the attribute **DataType**, and
2. ensuring the corresponding syntax, according to the GML 2.1.2 definition.

This way of defining new XACML **<AttributeValue>** elements requires an extension to the **<AttributeSelector>**, as it must realize the new attribute data-types.

Because the GML geometry data types are being used, the original syntax and semantics as defined by the OGC standard (see [GML2]) may not be changed. This is reflected by using the GML URI definitions for the XACML **<AttributeValue>** elements.

Geometry type	GeoXACML URI	GML 2.1.2 element
Point	http://www.opengis.net/gml#point	gml:Point
LineString	http://www.opengis.net/gml#linestring	gml:LineString
LienarRing	http://www.opengis.net/gml#linearring	gml:LinearRing

Polygon	http://www.opengis.net/gml#polygon	gml:Polygon
Box	http://www.opengis.net/gml#box	gml:Box

Table 4.1: GeoXACML **<AttributeValue>** elements, defining GML geometry data-types

One example of a GeoXACML **<AttributeValue>** element, representing a Polygon is shown in the following code segment[6]:

```
<AttributeValue DataType="http://www.opengis.net/gml#polygon">
  <gml:Polygon gid="P2" srsName="foo">
    <gml:outerBoundaryIs>
      <gml:LinearRing>
        <gml:coordinates>3 0,6 1,6 5,1 5,0 2,3 0</gml:coordinates>
      </gml:LinearRing>
    </gml:outerBoundaryIs>
   </gml:Polygon>
</AttributeValue>
```

Listing 4.3: GeoXACML Polygon **<AttributeValue>**

4.2.1.2 Defining spatial functions

The functions of GeoXACML, allow the validation of the existence of a specific spatial relation between two geometries. For the exact semantics of the functions, the reader is referred to [OGC 99] and [Vivid]. The URI that is used in the GeoXACML definition refers to the authors namespace[7]. GeoXACML of this version defines the functions, to be used in the **<Function>** element.

Function	**GeoXACML URI**
Disjoint	www.andreas-matheus.de/geoxacml/1.0/function#disjoint
Touches	www.andreas-matheus.de/geoxacml/1.0/function#touches
Crosses	www.andreas-matheus.de/geoxacml/1.0/function#crosses
Within	www.andreas-matheus.de/geoxacml/1.0/function#within
Overlaps	www.andreas-matheus.de/geoxacml/1.0/function#overlaps
Intersects	www.andreas-matheus.de/geoxacml/1.0/function#intersects
Equals	www.andreas-matheus.de/geoxacml/1.0/function#equals

Table 2: GeoXACML spatial functions

4.2.2 The Coordinate Reference System identification

Even though the topology of geometries in invariant for changes of the Coordinate Reference System (CRS), it must be ensured that the tested geometries are defined, using the same – identical – CRS. For the deriving of an authorization decision from a spatial permission, this implies that both geometries refer to the same CRS. Basically two options exist:

1. The CRS is inserted in the AuthorizatonDecisionRequest as an AttributeValue element. This requires that the Enforcement Service can obtain that information. If the CRS information is available as an AttributeValue (see table 4.3), a global check for the entire ResourceContent can be performed.
2. The CRS is taken from the geometry to be tested. This has the advantage that the equivalence of the geometry's CRS can be tested, immediately before the spatial relation is evaluated.

[6] The example uses the CRS foo, refering to the Cartesian Reference System.

[7] www.andreas-matheus.de

AttributeValue	GeoXACML URI	Data-type
CRS-ID	www.andreas-matheus.de/geoxacml/1.0/resource#crs-id	anyURI

Table 4.3: AttributeValue element for expressing the CRS identifier

4.2.3 The Service / Operation identification

Access restrictions are associated to the geodata, even though the data is accessible through service operations. Because the declaration of the permissions rely on the XML structure of the exchanged information, the restrictions are related to a particular service operation.

As an example, WFS 1 and WFS 2 provide access to the same geodata, but WFS 1 returns the geodata in a different structure than WFS 2. This means that – even though the permissions for both services are the same – the Xpath expressions for matching the resources are different for the WFS 1 and WFS 2. This requires that it must be possible to associate a permission to a service / operation combination in order to rely on the correct structuring of the resources, as it is the content of the **`<ResourceContent>`** element.

In order to express that relation, GeoXACML defines to additional attributes, as listed in table 4.4.

AttributeValue	GeoXACML URI	Data-type
Service-ID	www.andreas-matheus.de/geoxacml/1.0/resource#service-id	String
Operation-ID	www.andreas-matheus.de/geoxacml/1.0/resource#operation-id	String

Table 4.4: AttributeValue elements for expressing the relationship service / operation

4.2.4 Declaration of spatial permissions, using GeoXACML

For the declaration of access permissions, using GeoXACML it is important to keep in mind the deriving of an authorization decision obeys to the same semantics as it is defined by XACML. In that sense, the difference in GeoXACML is that it supports the declaration of spatial permissions, using geometry attributes and functions that resolve spatial relations between a resource geometry and a permission geometry. Each permission is represented by the **`<Rule>`** element that contains of an optional **`<Condition>`** element. This element is mandatory for expressing spatial permissions as it allows the use of geometry **`<AttributeValue>`** elements and spatial **`<Function>`** elements.

```
1 <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2  <Function FunctionId="www.andreas-matheus.de/geoxacml/1.0/function#within"/>
3    <AttributeValue DataType="http://www.opengis.net/gml#polygon">
4      <Polygon gid="P" srsName="foo">
5        <outerBoundaryIs>
6          <LinearRing>
7             <coordinates>3 0,6 1,6 5,1 5,0 2,3 0</coordinates>
8          </LinearRing>
9        </outerBoundaryIs>
10     </Polygon>
11   </AttributeValue>
12   <AttributeSelector RequestContextPath="//am:Intersection/am:location"
       DataType="http://www.opengis.net/gml#point"/>
14</Condition>
```

Listing 4.4: Example GeoXACML condition

The example of a **`<Condition>`** element from listing 4.4 expresses the Boolean condition, which evaluates to true if any of the locations of an intersection is within the area `3 0,6 1,6 5,1 5,0 2,3 0` . The following sequence of actions is computed to evaluate the result:

1. It is checked, if the RequestContent of the XACML AuthorizationRequest contains an XML encoded document that uses the namespace definition `am`
2. Each element of the RequestContent, matching the Xpath expression `//am:Intersection/am:location` , the geometry must be of type `http://www.opengis.net/gml#point`
3. For each location it must be checked, if the CRS of the location geometry matches the CRS of the permission geometry, expressed as a polygon
4. For each location geometry it must be checked if the spatial relation Within is true compared to the permission geometry
5. If no location exists, the **`<Condition>`** returns the Boolean value *False* and *True* otherwise.

4.3 Implementation issues

The use of additional **`<AttributeValue>`** elements requires specific handling by the AttributeSelector. In the example above, the AttributeSelector in line 12 must recognize and process the **`<AttributeValue>`** element of type gml:Point. This can be achieved by implementing a specific AttributeSelector.

In order to store the geometry for the spatial **`<AttributeValue>`** elements, the Java Topology Suite data-types can be used.

4.3.1 The AttributeSelector module

The handling of the geometry **`<AttributeValue>`** elements can be achieved by implementing a specific AttributeSelectorModule.

4.3.2 The AttributeFinderModule module

The handling of the spatial **`<Function>`** elements an be achieved by implementing a specific AttributeFinderModule.

5. Conclusion

The declaration and enforcement of access restrictions, using GeoXACML provides the capabilities for deploying a distributed access control system in order to protect access to geodata, as it is available through a Service-based Geo Data Instrastructure.

Since the first presentation in January 2005 at the OGC Technical Meeting in New York City, different demonstrations are available:

- First demo, as presented at the 52nd meeting: Prototype Implementation of Geospatial Authorization
- Second demo, as presented at the 53rd meeting: https://proton.do.isst.fraunhofer.de/services/

6. Appendix

6.1 A – WMSResoruceContent.xsd

This XML-Schema example defines the structure of a WMS related *ResourceContent*, as it is being used with the XACML *AuthorizationDecisionRequest*.

Please note that the layer names are specific to the WMS, where the access is to be protected. For the example below, the WMS provides access to the layers *BUILTUPA*, *COASTL* and *POLBNDL*.

```
<xs:schema targetNamespace="http://www.in.tum.de/am" xmlns="http://www.w3.org/2001/XMLSchema"
          xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:am="http://www.in.tum.de/am"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml"
          elementFormDefault="qualified">
  <xs:import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
  <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlinks.xsd"/>
  <!-- Definition of the root element for the resource content-->
  <xs:element name="WMSResurceContent" type="am:WMSFeatureCollectionType"
          substitutionGroup="gml:_FeatureCollection"/>
  <xs:complexType name="WMSFeatureCollectionType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureCollectionType"/>
    </xs:complexContent>
  </xs:complexType>
  <!-- Definition of layers, represented by features -->
  <xs:element name="BUILTUPA" type="am:WMSFeatureType" substitutionGroup="gml:_Feature"/>
  <xs:element name="COASTL" type="am:WMSFeatureType" substitutionGroup="gml:_Feature"/>
  <xs:element name="POLBNDL" type="am:WMSFeatureType" substitutionGroup="gml:_Feature"/>
  <!-- Definition of the common WMS feature type -->
  <xs:complexType name="WMSFeatureType">
    <xs:complexContent>
      <xs:restriction base="gml:AbstractFeatureType">
        <xs:sequence minOccurs="0">
          <xs:element name="PointOfInterest" type="gml:PointType"/>
          <!-- The element PointOfInterest represents the location for requesting additional
        information, using the GetFeatureInfo interface-->
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

Listing A.1: The XML-Schema definition for a WMS based *ResourceContent*

```
<WMSResurceContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.in.tum.de/am"
          xmlns:gml="http://www.opengis.net/gml" xsi:schemaLocation="http://www.in.tum.de/am
          WMSResourceContent.xsd">
  <gml:boundedBy>
    <gml:Box srsName="EPSG:4326">
      <gml:coordinates decimal=".">-97.105 24.913,-78.794 36.358</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember><COASTL/></gml:featureMember>
</WMSResurceContent>
```

Listing A.2: A ResourceContent example for a WMS GetMap request for the layer *COASTL* and the area of interest *-97.105 24.913,-78.794 36.358*, expressed in CRS *EPSG:4326*

6.2 B – CityModel.xsd

This XML-Schema example defines the structure of a WFS related *ResourceContent*, as it is being used with the XACML *AuthorizationDecisionRequest*.

Please note that the feature-type definitions are specific to the WFS, where the access is to be protected. For the example below, the WFS provides access to the feature-types *Street*, *Intersection* and *Building*.

```
<schema targetNamespace="http://www.in.tum.de/am" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
          xmlns:am="http://www.in.tum.de/am" xmlns="http://www.w3.org/2001/XMLSchema"
          elementFormDefault="qualified">
  <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlinks.xsd"/>
  <!-- -->
  <element name="CityModel" type="am:CityModelType" substitutionGroup="gml:_FeatureCollection"/>
  <element name="Street" type="am:StreetType" substitutionGroup="gml:_Feature"/>
  <element name="Intersection" type="am:IntersectionType" substitutionGroup="gml:_Feature"/>
  <element name="Building" type="am:BuildingType" substitutionGroup="gml:_Feature"/>
  <complexType name="CityModelType">
    <complexContent>
      <extension base="gml:AbstractFeatureCollectionType"/>
```

```
    </complexContent>
  </complexType>
  <complexType name="StreetType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence minOccurs="0">
          <element name="name"/>
          <element name="line" type="gml:LineStringType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="IntersectionType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence minOccurs="0">
          <element name="name"/>
          <element name="location" type="gml:PointType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="BuildingType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence minOccurs="0">
          <element name="address"/>
          <element name="shape" type="gml:PolygonType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

Listing B.1: XML-Schema example for a WFS based ResourceContent

```
<CityModel xmlns="http://www.in.tum.de/am" xmlns:am="http://www.in.tum.de/am"
          xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://http://www.in.tum.de/am
          CityModel.xsd" fid="CityModel">
  <gml:boundedBy>
    <gml:Box gid="box1" srsName="foo">
      <gml:coordinates cs="," ts=" " decimal=".">-1 2, 6 5</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <Building fid="The White House">
      <address>1600 Pennsylvania Avenue NW, Washington, DC 20500</address>
      <shape srsName="foo">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coord><gml:X>-1</gml:X><gml:Y>2</gml:Y></gml:coord>
            <gml:coord><gml:X>0</gml:X><gml:Y>2</gml:Y></gml:coord>
            <gml:coord><gml:X>0</gml:X><gml:Y>3</gml:Y></gml:coord>
            <gml:coord><gml:X>-1</gml:X><gml:Y>3</gml:Y></gml:coord>
            <gml:coord><gml:X>-1</gml:X><gml:Y>2</gml:Y></gml:coord>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </shape>
    </Building>
  </gml:featureMember>
  <gml:featureMember>
    <Building fid="The Empire State Building">
      <address>350 Fifth Ave / Ste. 3201, New York, NY 10118</address>
      <shape srsName="foo">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coord><gml:X>5</gml:X><gml:Y>4</gml:Y></gml:coord>
            <gml:coord><gml:X>6</gml:X><gml:Y>4</gml:Y></gml:coord>
            <gml:coord><gml:X>6</gml:X><gml:Y>5</gml:Y></gml:coord>
            <gml:coord><gml:X>5</gml:X><gml:Y>5</gml:Y></gml:coord>
            <gml:coord><gml:X>5</gml:X><gml:Y>4</gml:Y></gml:coord>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </shape>
    </Building>
  </gml:featureMember>
</CityModel>
```

Listing B.1: Example GML feature collection for the *CityModel* application schema

6.3 C – GeoXACML encoding of the example restrictions

The example encoding is non-normative, because different possible encoding in GeoXACML exist. The main difference between the – logically equivalent – encoding is the processing time for deriving an authorization decision.

Common to all declarations is that the association to the service and the operation is realized by the GeoXACML **<AttributeValue>** elements www.andreas-matheus.de/geoxacml/1.0/resource#service-id and http://www.andreas-matheus.de/geoxacml/1.0/resource_-_operation-id. Assuming that the WMS has the URL www.wms.com and the WFS has the URL www.wfs.com, the appropriate matching for the policy's <Target> element can be realized.

6.3.1 P1

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="P1"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">www.wms.com</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#service-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GetMap</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#operation-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="P1a" Effect="Permit">
    <Description>Alice can map layer COASTL</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:WMSResurceContent/gml:featureMember/am:COASTL)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">map</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Rule RuleId="P1b-Deny" Effect="Deny">
    <Description>Bob cannot map layer COASTL</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Bob</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
```

```
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:WMSResurceContent/gml:featureMember/am:COASTL)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">map</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Rule RuleId="P1b-Permit" Effect="Permit">
    <Description>Bob can map all layers</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Bob</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">map</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
  </Target>
</Rule>
</Policy>
```

Listing C.1: GeoXACML Policy declaration for restriction P1

6.3.2 P2

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="P2"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">www.wms.com</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#service-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GetMap</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#operation-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="P2a" Effect="Permit">
    <Description>Alice can map layer COASTL within the area 3 0,6 1,6 5,1 5,0 2,3 0, expressed in the CRS foo.
          All other requests are denied.</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
          <AttributeSelector RequestContextPath="count(//am:WMSResurceContent/gml:featureMember/am:COASTL)"
        DataType="http://www.w3.org/2001/XMLSchema#integer"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">map</AttributeValue>
          <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
        DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
    <Function FunctionId="http://www.andreas-matheus.de/geoxacml/1.0/function#within"/>
    <AttributeValue DataType="http://www.opengis.net/gml#polygon">
      <Polygon gid="P2" srsName="foo">
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>3 0,6 1,6 5,1 5,0 2,3 0</coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </AttributeValue>
    <AttributeSelector RequestContextPath="//am:WMSResurceContent/gml:boundedBy"
        DataType="http://www.opengis.net/gml#box"/>
  </Condition>
 </Rule>
 <Rule RuleId="P2b" Effect="Deny"/>
</Policy>
```

Listing C.2: GeoXACML declaration for restriction P2

6.3.3 P3

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="P3"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">www.wms.com</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#service-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GetMap</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#operation-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="P3a" Effect="Permit">
    <Description>Alice can read all features of type Building</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:CityModel/gml:featureMember/am:Building)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
```

```
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Rule RuleId="P3b" Effect="Permit">
    <Description>Bob can write all features of type Intersection</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Bob</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:CityModel/gml:featureMember/am:Intersection)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">write</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
  <Rule RuleId="P3c" Effect="Deny"/>
</Policy>
```

Listing C.3: GeoXACML declaration for restriction P3

6.3.4 P4

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="P4"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">www.wms.com</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#service-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GetMap</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#operation-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="P4a" Effect="Permit">
    <Description>Alice cannot read the feature building, as it is identified by the fid “The White House”. All
          other requests are denied</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
```

```
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:CityModel/gml:featureMember/am:Building)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">The White House</AttributeValue>
      <AttributeSelector RequestContextPath="//am:CityModel/gml:featureMember/am:Building/@fid"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Condition>
  </Rule>
  <Rule RuleId="P4b" Effect="Deny"/>
</Policy>
```

Listing C.4: GeoXACML declaration for restriction P4

6.3.5 P5

```
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="P5"
          RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">www.wms.com</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#service-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GetMap</AttributeValue>
          <ResourceAttributeDesignator AttributeId="http://www.andreas-
          matheus.de/geoxacml/1.0/resource#operation-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="P5a" Effect="Permit">
    <Description>Alice can read all features of type Building, if within the area 3 0,6 1,6 5,1 5,0 2,3 0,
          expressed in the CRS foo. All other requests are denied.</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Alice</AttributeValue>
            <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-
          subject" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
            <AttributeSelector RequestContextPath="count(//am:CityModel/gml:featureMember/am:Building)"
          DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
      <Function FunctionId="http://www.andreas-matheus.de/geoxacml/1.0/function#within"/>
```

```
      <AttributeValue DataType="http://www.opengis.net/gml#polygon">
        <Polygon gid="P2" srsName="foo">
          <outerBoundaryIs>
            <LinearRing>
              <coordinates>3 0,6 1,6 5,1 5,0 2,3 0</coordinates>
            </LinearRing>
          </outerBoundaryIs>
        </Polygon>
      </AttributeValue>
      <AttributeSelector RequestContextPath="//am:CityModel/gml:featureMember/am:Building/am:shape"
          DataType="http://www.opengis.net/gml#polygon"/>
    </Condition>
  </Rule>
  <Rule RuleId="P5b" Effect="Deny"/>
</Policy>
```

Listing C.5: GeoXACML declaration for restriction P5