

Open Geospatial Consortium

Submission Date: 2013-09-05

Approval Date: 2013-09-25

Publication Date: 2013-11-06

External identifier of this OGC® document: <http://www.opengis.net/doc/DP/GEOXACML/GML3-Extension>

Internal reference number of this OGC® document: OGC 13-101

Category: OGC® Publicly Available Discussion Paper

Editor: Andreas Matheus

OGC Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 GML 3.2.1 Encoding Extension

Copyright © 2013 Open Geospatial Consortium
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Attention

This document is written using the OGC template for Implementation Standards and contains normative language.

But it is important to notice that this OGC publication represents the opinion of the author and submitters regarding GeoXACML 3.0 GML 3.2.1 extension.

Warning

i

Document type: OGC® Discussion Paper
Document stage: Draft OGC Standard
Document language: English

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Contents

1.	Scope.....	6
2.	Conformance.....	6
2.1	Overview.....	6
2.2	Specification identifier.....	6
2.3	Conformance classes related to implementation.....	6
2.4	Backward Compatibility with GML Extension to GeoXACML 1.0.....	7
3.	References.....	8
4.	Terms and Definitions.....	8
5.	Conventions	9
5.1	Abbreviated terms	9
5.2	Document presentation of the specification.....	9
6.	GML encoded geometries within GeoXACML 3.0 (informative)	10
6.1	Introduction to GML encoded geometries	10
6.2	Introduction to GML nil geometries in the context of GeoXACML	10
6.3	Introduction to including complex data types into GeoXACML 3.0	13
6.4	Introduction to XML encoded resources in GeoXACML 3.0	14
7.	GML 3.2 Simple Features Profile Encoding Extension (normative).....	19
7.1	Introduction (informative).....	19
	Annex A: Conformance Class Abstract Test Suite (Normative)	21

i. Abstract

This standard defines the version 3.0 of a valid GML 3.2.1 geometry encoding as defined in Geography Markup Language (GML) simple features profile (with Corrigendum) to be used with the GeoXACML 3.0 Core standard.

The use of this encoding extension to GeoXACML 3.0 Core enables the direct use of GML 3.2.1 encoded geometries into a GeoXACML 3.0 Policy, an Authorization Decision Request or in an Authorization Decision's Obligation element. It thereby improves the performance of deriving access control decisions, where geometries are involved as existing GML 3.2.1 geometry encodings must not be transformed to Well Known Text (WKT) as supported by GeoXACML 3.0 Core. Furthermore, the use of this encoding extension simplifies the implementation of a Policy Enforcement Point as it must not provide the transformation functions from GML to WKT and vice versa.

This encoding extension has its normative base in Geography Markup Language (GML) simple features profile (with Corrigendum).

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, GeoXACML, XACML, GML, access control, encoding extension

iii. Preface

This document defines the version 3.0 of the GML 3.2.1 geometry encoding extension to GeoXACML 3.0 Core . It hereby supersedes the previous version GeoXACML 1.0 encoding extension for GML. It is important to notice that version 2.0 of GeoXACML does not exist and therefore no encoding extension of version 2.0!

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

University of the Bundeswehr

Oracle

Defense Information Systems Agency (DISA)

National Geospatial-Intelligence Agency (NGA)

v. Submitters

All questions regarding this submission should be directed to the editor:

Name	Affiliation
Andreas Matheus	andreas.matheus@unibw.de

1. Scope

This Standard defines Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 GML 3.2.1 Encoding Extension as an encoding extension to GeoXACML 3.0 Core which allows the processing of GML 3.2.1 encoded geometries. The supported geometry encodings for GML 3.2.1 is defined in Geography Markup Language (GML) simple features profile (with Corrigendum).

In particular, the use of this extension supports the direct use of GML 3.2.1 encoded geometries as part of the GeoXACML 3.0 Policy, an Authorization Decision Request or in an Authorization Decision's Obligation element.

2. Conformance

2.1 Overview

This Standard defines one standardization target type:

- implementation: GML 3.2.1 based geometry encoding that must be understood by a Policy Administration Point (policy creation and maintenance), a Policy Enforcement Point (processing of authorization decision requests and authorization decisions) and a geospatially enriched Policy Decision Point (GeoPDP) implementation (processing of GeoXACML policies and requests and producing an authorization decision).

NOTE: GeoXACML 3.0 is XACML 3.0 schema compliant. Therefore, a GeoXACML 3.0 Policy, or a ADR or AD instance is using the XACML 3.0 namespace. However, the GML encoded geometry instance will use the GML namespace.

2.2 Specification identifier

All requirements-classes and conformance-classes described in this document are owned by the specification identified as

<http://www.opengis.net/spec/GEOXACML/3.0/GML3-Extension>

2.3 Conformance classes related to implementation

The conformance rules are based on geometry encoding options as defined by Geography Markup Language (GML) simple features profile (with Corrigendum).

Table 2 — Conformance classes related GeoXACML implementation

Conformance class	Description	Clause
SF-0	Support the use of GML geometry encoding based on GML 3.2 Simple Features Profile compliant with SF-0	
SF-1	Support the use of GML geometry encoding based on GML 3.2 Simple Features Profile compliant with SF-1	
SF-2	Support the use of GML geometry encoding based on GML 3.2 Simple Features Profile compliant with SF-2	

2.4 Backward Compatibility with GML Extension to GeoXACML 1.0

This version of a GML Encoding Extension to GeoXACML 3.0 is normatively based on Geography Markup Language (GML) simple features profile (with Corrigendum).

Because this normative base introduces constraints on GML 3.2.1 in terms of supported geometry encodings, not all geometry encodings supported by GML 3 may be used. In that sense, the backwards compatibility of a GeoXACML 1.0 policy using GML encoded geometries is limited. The backwards compatibility is broken if the used GML 3 geometry in the GeoXACML 1.0 policy is not supported by the <Simple Features for GML> specification! For those policies, a transformation to GeoXACML 3.0 using the GML 3.2.1 encoding extension is not possible! Also, any GeoXACML 1.0 policy using GML 2 encoding extension is not backwards compatible, because – as of now - GML 2 encodings are not defined in any encoding extension.

3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

OASIS eXtensible Access Control Markup Language (XACML) Version 3.0, 22 January 2013, <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>

OASIS eXtensible Access Control Markup Language (XACML) Version 3.0 XML Schema, <http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd>

Geography Markup Language (GML) simple features profile (with Corrigendum), OGC Document 10-100r3

Geospatial eXtensible Access Control Markup Language (GeoXACML) 3.0 Core, OGC Document TBD

*OGC Naming Authority (OGC-NA) Policies & Procedures OGC Document 09-046r2
<http://www.opengis.net/doc/POL/OGC-NA/1.1>*

Policy Directives for Writing and Publishing OGC Standards: TC Decisions. OGC Document 06-135r7. <http://www.opengis.net/doc/POL/STD>

The Specification Model — A Standard for Modular specifications OGC Document 08-131r3. <http://www.opengis.net/doc/POL/SPEC>

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

4.1 GML

Geography Markup Language

4.2 WKT

Well Known Text

5. Conventions

5.1 Abbreviated terms

GeoXACML Geospatial eXtensible Access Control Markup Language

GML	Geography Markup Language
ISO	International Organization for Standardization
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
UML	Unified Modeling Language
SF	Simple Features
WKT	Well Known Text
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

5.2 Document presentation of the specification

This document presents the GML based geometry encoding based on GML 3.2.1 Simple Features Profile to be used with GeoXACML 3.0 Core specification using a representation that follows the structures defined in the OGC Policy [*The Specification Model — A Standard* for Modular specifications OGC Document 08-131r3. <http://www.opengis.net/doc/POL/SPEC>]. All normative material is organized as requirements, requirements classes, conformance tests and conformance classes. Each is identified with a URI, and the content and dependencies are described in tables whose structure matches the specification model.

6. GML encoded geometries within GeoXACML 3.0 (informative)

6.1 Introduction to GML encoded geometries

The Geography Markup Language (GML) enables a standards based approach for encoding geometry information using XML. The following example illustrates a GML 3.2 encoding of a geometry instance of type Circle.

```
<gml:CircleByCenterPoint numArc="1" xmlns:gml="http://www.opengis.net/gml/3.2">
  <gml:pos srsName="urn:ogc:def:crs:EPSG::4979"
    >29.963745015416 -90.029951432619</gml:pos>
  <gml:radius uom="m">1250</gml:radius>
</gml:CircleByCenterPoint>
```

Figure 1 — GML 3.2 geometry encoding example of CircleByCenterPoint

6.2 Introduction to GML nil geometries in the context of GeoXACML

In GML 3.2, nullable geometries are a concept that must be manifested in a GML Application Schema. For illustration purposes, let's assume we have the following GML 3.2 Application Schema defined:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.opengis.org/geoxacml/3.0/example"
  xmlns:example="http://www.opengis.org/geoxacml/3.0/example"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">

  <xs:import namespace="http://www.opengis.net/gml/3.2"
  schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>

  <xs:complexType name="LocationType">
    <xs:sequence>
      <xs:choice>
        <xs:element ref="gml:Point"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="nilReason" type="gml:NilReasonType"/>
    <xs:attribute ref="gml:id"/>
  </xs:complexType>

  <xs:complexType name="UserContextType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
        <xs:sequence>
          <xs:element name="Location" minOccurs="1" maxOccurs="1"
nillable="true" type="geoxacml:LocationType"/>
          <xs:element name="DateTime" minOccurs="1" maxOccurs="1"
nillable="false" type="xs:dateTime"/>
          <xs:element name="Name" minOccurs="1" maxOccurs="1"
nillable="false" type="xs:string"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```

        <xs:element name="Role"      minOccurs="0" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:element name='UserContext' type='geoxacml:UserContextType' />
</xs:schema>

```

Figure 2 — Example GML 3.2 Application Schema

In the Application Schema above, it is possible to nil the <Location> element which includes a GML <Point> element. In the following, we can instanciate two different user contexts, one where a location is provided and another where the location information is withheld.

```

<example:UserContext xmlns="http://www.opengis.org/geoxacml/3.0/example"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml/3.2"
xsi:schemaLocation="http://www.opengis.org/geoxacml/3.0 application-schema-
example.xsd"
gml:id="u1">
    <example:Location gml:id="l1" >
        <gml:Point gml:id="p1" srsName="EPSG:4326">
            <gml:pos>48.136944 11.575278</gml:pos>
        </gml:Point>
    </example:Location>
    <example:DateTime>2013-03-13T11:38:00Z</example:DateTime>
    <example:Name>Joe</example:Name>
</example:UserContext>

```

Figure 3 — Example 1 of a User Context based on example Application Schema

```

<example:UserContext xmlns="http://www.opengis.org/geoxacml/3.0/example"
xmlns:example="http://www.opengis.org/geoxacml/3.0/example"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml/3.2"
xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd
http://www.opengis.org/geoxacml/3.0/example application-schema-example.xsd"
gml:id="u1">
    <example:Location gml:id="l2" xsi:nil="true" nilReason="withheld"/>
    <example:DateTime>2013-03-13T11:38:00Z </example:DateTime>
    <example:Name>Joe</example:Name>
</example:UserContext>

```

Figure 4 — Example 2 of a User Context based on example Application Schema

The question now arrives how can these user context example be mapped to GeoXACML 3.0 using a set of simple <AttributeValue> elements. The mapping can take place such, that the <DateTime> element is mapped to an <AttributeValue> of data type

<http://www.w3.org/2001/XMLSchema#dateTime> and the user name provided in element <Name> can be mapped to an <AttributeValue> of data type <http://www.w3.org/2001/XMLSchema#string>. The mapping of the <Location> attribute to an <AttributeValue> of data type urn:ogc:def:dataType:geoxacml:3.0:geometry is – of course - different as illustrated below.

```
<AttributeValue DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry"
  xmlns:gml="http://www.opengis.net/gml/3.2">
  <gml:Point gml:id="p1" srsName="EPSG:4326">
    <gml:pos>48.136944 11.575278</gml:pos>
  </gml:Point>
</AttributeValue>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
  >2013-03-13T11:38:00Z</AttributeValue>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
  >Joe</AttributeValue>
```

Figure 5 — Example 1 mapped to GeoXACML using <AttributeValue>

```
<AttributeValue DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  gml:nil="true"
  gml:nilReason="withheld"/>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime">
  >2013-03-13T11:38:00Z</AttributeValue>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
  >Joe</AttributeValue>
```

Figure 6 — Example 2 mapped to GeoXACML using <AttributeValue>

In cases, where the XML document from the GML Application Schema is inserted into an Authorization Decision Request into the <Content> element, it is not possible to use the IsNil or NilReason functions, as these functions operate on parameter data type Geometry only. But the nilled elements are not of type Geometry.

However, the use of Xpath expressions can be used to identify whether a XML element is nil and thereby not encoding a geometry as a child element.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  ReturnPolicyIdList="false"
  CombinedDecision="false"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xsi:schemaLocation="http://www.opengis.net/gml/3.2
  http://schemas.opengis.net/gml/3.2.1/gml.xsd
  http://www.opengis.org/geoxacml/3.0/example application-schema-example.xsd
  urn:oasis:names:tc:xacml:3.0:core:schema:wd-17 http://docs.oasis-
```

```

open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd">
  <Attributes Category="access-subject">
    <Attribute AttributeId="Location" IncludeInResult="false">
      <AttributeValue DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry"
gml:nil="true" gml:nilReason="withheld"/>
    </Attribute>
    <Attribute AttributeId="DateTime" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#dateTime"
>2013-03-01T11:38:00Z</AttributeValue>
    </Attribute>
    <Attribute AttributeId="Name" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
>Joe</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="access-ressource">
    <Content>
      <example:UserContext
        xmlns:example="http://www.opengis.org/geoxacml/3.0/example"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:gml="http://www.opengis.net/gml/3.2"
        xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd
http://www.opengis.org/geoxacml/3.0/example application-schema-example.xsd"
        gml:id="u1">
        <example:Location gml:id="12" xsi:nil="true" nilReason="withheld"/>
        <example:DateTime>2013-03-01T11:38:00Z</example:DateTime>
        <example:Name>Joe</example:Name>
      </example:UserContext>
    </Content>
  </Attributes>
</Request>

```

Figure 7 — Example 2 mapped to GeoXACML ADR into <Content> element

For the example above, the following Xpath expressions inside a GeoXACML policy can be used to test whether the <Location> element is nilled and to obtain the nil reason:

- boolean(/example:UserContext/example:Location/@xsi:nil='true')
- string(/example:UserContext/example:Location/@nilReason)

6.3 Introduction to including complex data types into GeoXACML 3.0

The ability to include GML and hence XML encoded geometries into GeoXACML is defined in XACML 3.0 through the data type extension point and the “lax” processing of the <AttributeValue> content.

An <AttributeValue> element has an attribute named DataType, which is of type xs:anyURI. According to the extension capabilities of XACML, additional attribute values can be defined by associating a unique DataType identifier to it.

Section 8.2 of the XACML specification states that “*<xacml:AttributeValue> and <xacml-context:AttributeValue> elements MAY contain an instance of a structured XML data-type.*”.

This capability allows the definition of the `geometry` data type, as defined by GeoXACML 3.0 Core and to include a GML encoded geometry.

```

<xs:element name="AttributeValue" type="xacml:AttributeValueType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="AttributeValueType" mixed="true">
  <xs:complexContent mixed="true">
    <xs:extension base="xacml:ExpressionType">
      <xs:sequence>
        <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:extension>
  </xs:complexContent>
</xs:complexType>

```

Figure 8 — XACML schema definition of the `<AttributeValue>` element

The following example illustrates the use of the GML encoding for a CircleByCenterPoint with a XACML 3.0 `<AttributeElement>` and the GeoXACML 3.0 data type `geometry`.

```

<AttributeValue xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry">
  <gml:CircleByCenterPoint numArc="1"
    xmlns:gml="http://www.opengis.net/gml/3.2">
    <gml:pos srsName="urn:ogc:def:crs:EPSG::4979"
      >29.963745015416 -90.029951432619</gml:pos>
    <gml:radius uom="m">1250</gml:radius>
  </gml:CircleByCenterPoint>
</AttributeValue>

```

Figure 9 — GeoXACML 3.0 geometry `<AttributeValue>` example with a GML3 encoded geometry instance

6.4 Introduction to XML encoded resources in GeoXACML 3.0

The typical use of this extension provides support for any GeoXACML request context that shall include geometries using GML 3.2 encoding. For example, a OGC Web Service HTTP/POST encoded request is encoded in XML. An in a case where geometries express parameters of the request (e.g. BBOX), the encoded geometry(ies) could directly be included into the request context.

Further more important is the use case where the entire OWS request (e.g. a WFS-T TRANSACTION request) is subject for access restrictions. When using XML to encode

the request and the geometries are encoded in GML 3.2, it is possible to copy the OWS request 1:1 and insert it into the GeoXACML request context with no changes!

```

<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  ReturnPolicyIdList="false"
  CombinedDecision="false"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
  http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd">
  <Attributes Category="access-subject">
    <Attribute AttributeId="subject:location:1" IncludeInResult="false">
      <AttributeValue
        DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry">Point (29.963745015416 -
        90.029951432619)</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="access-ressource">
    <Content>
      <GetFeature
        service="WFS"
        version="2.0.0"
        xmlns="http://www.opengis.net/wfs/2.0"
        xmlns:myns="http://www.someserver.com/myns"
        xmlns:fes="http://www.opengis.net/fes/2.0"
        xmlns:gml="http://www.opengis.net/gml/3.2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.opengis.net/wfs/2.0
        http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
        http://www.opengis.net/gml/3.2
        http://schemas.opengis.net/gml/3.2.1/gml.xsd">
        <Query typeNames="myns:RoadSegments myns:RoadSegments"
          aliases="RS1 RS2">
          <fes:Filter>
            <fes:And>
              <fes:BBOX>
                <fes:ValueReference>/RS1/geometry</fes:ValueReference>
                <gml:Envelope srsName="urn:ogc:def:crs:EPSG::1234">
                  <gml:lowerCorner>10 10</gml:lowerCorner>
                  <gml:upperCorner>20 20</gml:upperCorner>
                </gml:Envelope>
              </fes:BBOX>
              <fes:BBOX>
                <fes:ValueReference>/RS2/geometry</fes:ValueReference>
                <gml:Envelope srsName="urn:ogc:def:crs:EPSG::1234">
                  <gml:lowerCorner>10 10</gml:lowerCorner>
                  <gml:upperCorner>20 20</gml:upperCorner>
                </gml:Envelope>
              </fes:BBOX>
              <fes:Crosses>
                <fes:ValueReference>/RS1/geometry</fes:ValueReference>
                <fes:ValueReference>/RS2/geometry</fes:ValueReference>
              </fes:Crosses>
            </fes:And>
          </fes:Filter>
        </Query>
      </GetFeature>
    </Content>
  </Attributes>
</Request>
```

Figure 10 — Example of a WFS POST request included into a GeoXACML 3.0 ADR

Another important use case for this extension is the ability to get access decisions for OWS responses, if encoded in XML and geometries encoded in GML 3.2. For example, a WFS response (the feature collection) is the resource for which a decision shall be derived or upon which filtering shall be applied. With the use of this extension, it is possible (again) to include the entire WFS feature collection into the GeoXACML 3.0 ADR.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  ReturnPolicyIdList="false"
  CombinedDecision="false"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
  http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd">
  <Attributes Category="access-subject">
    <Attribute AttributeId="subject:location:1" IncludeInResult="false">
      <AttributeValue
        DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry">Point (29.963745015416 -
        90.029951432619)</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="access-ressource">
    <Content>
      <wfs:FeatureCollection
        timeStamp="2008-09-07T19:00:00"
        numberReturned="2"
        numberMatched="unknown"
        xmlns="http://www.someserver.com/myns"
        xmlns:myns="http://www.someserver.com/myns"
        xmlns:wfs="http://www.opengis.net/wfs/2.0"
        xmlns:gml="http://www.opengis.net/gml/3.2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.opengis.net/wfs/2.0
        http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
        http://www.someserver.com/myns
        ./DescribeFeatureType_Example01_Response.xsd">
        <wfs:member>
          <TreesA_1M gml:id="TRESSA_1M.1013">
            <extent>
              <gml:Polygon gml:id="GID_10"
srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList srsDimension="2">65.588264 -120.000000
65.590782 -120.003571 65.590965 -120.011292 65.595215
                    -120.022491 65.592880 -120.031212 65.586121 -120.019363
65.585365 -120.030350 65.581848 -120.045082 65.584938
                    -120.059540 65.590500 -120.067284 65.595436 -120.067284
65.613441 -120.067337 65.613777 -120.067337 65.606346
                    -120.060997 65.605545 -120.045517 65.599777 -120.022675
65.601036 -120.003975 65.602081 -120.000000 65.602081
                    -120.000000 65.588264 -120.000000</gml:posList>

```

```

        </gml:LinearRing>
    </gml:exterior>
</gml:Polygon>
</extent>
<id>0000000002</id>
<treeType>Maple</treeType>
</TreesA_1M>
</wfs:member>
<wfs:member>
    <RoadL_1M gml:id="ROADL_1M.1914">
        <centerLine>
            <gml:LineString gml:id="GID_5"
srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -
59.488991 -52.198524 -59.485958 -52.169559
                    -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -
52.136417 -59.447968 -52.127190 -59.422928
                    -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -
59.371311 -52.121300</gml:posList>
            </gml:LineString>
        </centerLine>
        <designation>HYW 401</designation>
        <surfaceType>ASPHALT</surfaceType>
        <nLanes>12</nLanes>
    </RoadL_1M>
    </wfs:member>
</wfs:FeatureCollection>
</Content>
</Attributes>
</Request>

```

Figure 11 — Example of a WFS Feature Collection included into the GeoXAML 3.0 ADR

The above examples leverage examples from the WFS 2.0 specification.

```

<?xml version="1.0"?>
<GetFeature
service="WFS"
version="2.0.0"
xmlns="http://www.opengis.net/wfs/2.0"
xmlns:myns="http://www.someserver.com/myns"
xmlns:fes="http://www.opengis.net/fes/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd">
<Query typeNames="myns:RoadSegments myns:RoadSegments"
aliases="RS1 RS2">
<fes:Filter>
<fes:And>
<fes:BBBox>
<fes:ValueReference>/RS1/geometry</fes:ValueReference>
<gml:Envelope srsName="urn:ogc:def:crs:EPSG::1234">
<gml:lowerCorner>10 10</gml:lowerCorner>
<gml:upperCorner>20 20</gml:upperCorner>

```

```

</gml:Envelope>
</fes:BBOX>
<fes:BBOX>
<fes:ValueReference>/RS2/geometry</fes:ValueReference>
<gml:Envelope srsName="urn:ogc:def:crs:EPSG::1234">
<gml:lowerCorner>10 10</gml:lowerCorner>
<gml:upperCorner>20 20</gml:upperCorner>
</gml:Envelope>
</fes:BBOX>
<fes:Crosses>
<fes:ValueReference>/RS1/geometry</fes:ValueReference>
<fes:ValueReference>/RS2/geometry</fes:ValueReference>
</fes:Crosses>
</fes:And>
</fes:Filter>
</Query>
</GetFeature>

```

Figure 12 — Example WFS POST request [Example 15, OGC 09-025r1 and ISO/DIS 19142 144]

```

<?xml version="1.0" ?>
<wfs:FeatureCollection
timeStamp="2008-09-07T19:00:00"
numberReturned="2"
numberMatched="unknown"
xmlns="http://www.someserver.com/myns"
xmlns:myns="http://www.someserver.com/myns"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
http://www.someserver.com/myns
./DescribeFeatureType_Example01_Response.xsd">
<wfs:member>
<TreesA_1M gml:id="TRESSA_1M.1013">
<extent>
<gml:Polygon gml:id="GID_10" srsName="urn:ogc:def:crs:EPSG::4326">
<gml:exterior>
<gml:LinearRing>
<gml:posList srsDimension="2">65.588264 -120.000000 65.590782 -120.003571 65.590965 -
120.011292 65.595215 -120.022491 65.592880 -120.031212 65.586121 -120.019363 65.585365 -
120.030350 65.581848 -120.045082 65.584938 -120.059540 65.590500 -120.067284 65.595436 -
120.067284 65.613441 -120.067337 65.613777 -120.067337 65.606346 -120.060997 65.605545 -
120.045517 65.599777 -120.022675 65.601036 -120.003975 65.602081 -120.000000 65.602081 -
120.000000 65.588264 -120.000000</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</extent>
<id>0000000002</id>
<treeType>Maple</treeType>
</TreesA_1M>
</wfs:member>
<wfs:member>
<RoadL_1M gml:id="ROADL_1M.1914">
<centerLine>
<gml:LineString gml:id="GID_5" srsName="urn:ogc:def:crs:EPSG::4326">
<gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -59.488991 -52.198524 -59.485958 -
52.169559 -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -52.136417 -59.447968 -
52.127190 -59.422928 -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -59.371311 -
52.121300</gml:posList>
</gml:LineString>

```

```
</centerLine>
<designation>HYW 401</designation>
<surfaceType>ASPHALT</surfaceType>
<nLanes>12</nLanes>
</RoadL_1M>
</wfs:member>
</wfs:FeatureCollection>
```

Figure 13 — Example WFS response [Example 1, OGC 09-025r1 and ISO/DIS 19142 144]

7. GML 3.2 Simple Features Profile Encoding Extension (normative)

7.1 Introduction (informative)

This encoding extension to GeoXACML 3.0 Core leverages the requirement “7.3.7 Requirement: Geometry encoding using XML” and provides one possible XML encoding to be used; hence GML 3.2.

In contrast to the WKT geometry encoding as introduced in the GeoXACML 3.0, GML 3.2 does not support the equivalent encoding to `Empty`. However, as GML defines the concept of nil, it is important for this encoding extension to define two functions to operate on nil geometries: (i) `IsNil(xs: Any) : Boolean` which enables a GeoXACML policy to test for nil and (ii) `NilReason(xs: Any) : String` which supports to fetch the nil reason for further processing.

Req 1 Function IsNil

This function SHALL have the signature `IsNil(Geometry) : Boolean` and the identifier as `urn:ogc:def:function:geoxacml:3.0:geometry-is-nil`.

This function shall return a TRUE value if the XML element, which according to a GML application schema includes a GML encoded geometry, is marked a `nil="true"`.

This function shall return a FALSE value otherwise.

Req 2 Function NilReason

This function SHALL have the signature `NilReason(Geometry) : String` and the identifier as `urn:ogc:def:function:geoxacml:3.0:geometry-nil-reason`.

This function shall return the value of the attribute `nilReason` used for nilled XML elements that carry a GML encoded geometry.

Annex A: Conformance Class Abstract Test Suite (Normative)

Conformance class: SF-0

Test that the implementation supports all valid geometry encodings in GML as listed in Geography Markup Language (GML) simple features profile (with Corrigendum) for SF-0 in “Annex A Compliance testing“ according to Req 7 „Geometry encoding using XML“ defined by GeoXACML 3.0 Core.

Conformance class: SF-1

Test that the implementation supports Conformance Class SF-0.

Test that the implementation supports all valid geometry encodings in GML as listed in Geography Markup Language (GML) simple features profile (with Corrigendum) for SF-1 in “Annex A Compliance testing“.

Test that the following functions are supported by the implementation:

Function URN	Requirement
urn:ogc:def:function:geoxacml:3.0:geometry-is-nil	Req 1
urn:ogc:def:function:geoxacml:3.0:geometry-nil-reason	Req 2

Conformance class: SF-2

Test that the implementation supports Conformance Class SF-1.

Test that the implementation supports all valid geometry encodings in GML as listed in Geography Markup Language (GML) simple features profile (with Corrigendum) for SF-2 in “Annex A Compliance testing“.

Annex B: Conformance Test Files (normative)

Annex C: Bibliography**Annex D: Document revision history**

Date	Release	Author	Paragraph modified	Description
2013-03-20	0.1	Andreas Matheus	All	Document created
2013-05-03	0.2	Andreas Matheus	All	Comments received from Norman Brickman (MITRE) incorporated