OGC GEOSPATIAL EXTENSIBLE ACCESS CONTROL MARKUP LANGUAGE (GEOXACML) 3.0 JSON PROFILE V1.0

STANDARD Implementation

DRAFT

Version: 0.6
Submission Date: 2023-02-06
Approval Date: 2029-03-30
Publication Date: 2029-03-30
Editor: Andreas Matheus

Notice for Drafts: This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.
License Agreement

Use of this document is subject to the license agreement at https://www.ogc.org/license

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: http://ogc.standardstracker.org/

Copyright notice

Copyright © 2023 Open Geospatial Consortium
To obtain additional rights of use, visit https://www.ogc.org/legal

Note

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.
# CONTENTS

I. ABSTRACT .................................................................................................................. vi

II. KEYWORDS ................................................................................................................. vi

III. SECURITY CONSIDERATIONS .................................................................................. vii

IV. SUBMITTING ORGANIZATIONS ............................................................................. viii

1. SCOPE ........................................................................................................................ 2

2. CONFORMANCE .......................................................................................................... 4
   2.1. Conformance Class Data Model ........................................................................... 4
   2.2. Conformance Class Core ....................................................................................... 4

3. NORMATIVE REFERENCES ........................................................................................ 7

4. TERMS AND DEFINITIONS ....................................................................................... 9

5. CONVENTIONS ............................................................................................................. 11
   5.1. Identifiers .............................................................................................................. 11

6. INTRODUCTION TO GEOXACML 3.0 JSON PROFILE V1.0 .................................. 13

7. GEOXACML 3.0 JSON PROFILE V1.0 REQUIREMENTS ........................................ 15
   7.1. Requirement Class Data Model (abstract) .......................................................... 15
   7.2. Requirements Class GeoXACML 3.0 JSON Profile v1.0 Core ............................... 18

8. MEDIA TYPES FOR ANY DATA ENCODING(S) ......................................................... 31

ANNEX A (NORMATIVE) ABSTRACT TEST SUITE ....................................................... 33
   A.1. Conformance Class Data Model ........................................................................... 33
   A.2. Conformance Class Core ....................................................................................... 36

ANNEX B (INFORMATIVE) EXAMPLES FOR THE GEOXACML 3.0 JSON PROFILE V1.0
   ................................................................................................................................. 44
   B.1. Examples how to encode Geometry in ADR .......................................................... 44
   B.2. Example GeoXACML 3.0 policy, request and response ......................................... 46

ANNEX C (INFORMATIVE) REVISION HISTORY ........................................................... 50

BIBLIOGRAPHY .............................................................................................................. 52
<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>CONFORMANCE CLASS 1</td>
<td>4</td>
</tr>
<tr>
<td>CONFORMANCE CLASS 2</td>
<td>4</td>
</tr>
</tbody>
</table>
ABSTRACT

This OGC Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 JSON Profile v1.0 (GeoXACML 3.0 JSON Profile) defines an extension to the JSON Profile of XACML 3.0 Version 1.1 for supporting GeoXACML Authorization Decision Requests and Authorization Decision encoded in JSON. This ensures an easy uptake in environments where JSON is the preferred encoding.

For supporting Geometry as defined by GeoXACML 3.0 Core conformance class, this profile extends the Attribute DataType definition from JSON Profile of XACML 3.0 Version 1.1 with the geometry data-type urn:ogc:def:dataType:geoxacml:3.0:geometry

The GeoXACML 3.0 JSON Profile Standard supports the Attribute value to use Well-Known-Text, Well-Known-Binary (hex-encoding) or GeoJSON as an encoding alternative for the geometry data-type defined in GeoXACML 3.0.

To support the use of the GeoXACML 3.0 specific attributes srid, precision and allowTransformation, this profile extends the default JSON schema definition from JSON Profile of XACML 3.0 Version 1.1 accordingly.

KEYWORDS

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

ogcdoc, OGC document, XACML, GeoXACML, JSON, Profile
SECURITY CONSIDERATIONS

The GeoXACML 3.0 JSON Profile does not introduce specific attack vectors to those typically considered when parsing and validating JSON encoded service request / response. When the origin of the data is unknown, great care should be applied when parsing the data. The EdwardHuang post lists five common denial of service attack scenarios when parsing JSON data.

Mitigating attacks for the GeoXACML 3.0 JSON Profile are not different from those that can be found on the Internet. To mitigate parsing attacks, the use of a tight JSON schema in combination with meaningful limits for payload size, object size, nesting depth, etc. should be considered.

The use of JWS (JSON Web Signature) may be used in addition to this profile to establish trust in the origin of the request / response.
SUBMITTING ORGANIZATIONS

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

- Secure Dimensions GmbH
- Natural Resources Canada (NRCAN)
- Defense Information Systems Agency (DISA)
1

SCOPE
This Standard defines an extension to JSON Profile of XACML 3.0 Version 1.1 for supporting the encoding of a GeoXACML Authorization Decision Request and Authorization Decision in JSON.

This profile defines the encoding options for a Geometry instance as defined in GeoXACML 3.0 based on Well-Known-Text, Well-Known-Binary and GeoJSON.
CONFORMANCE
CONFORMANCE

This Standard defines two Conformance Classes.

Conformance with this Standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

In order to conform to this OGC® Standard, a software implementation shall pass all tests defined in Annex A.

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

2.1. Conformance Class Data Model

<table>
<thead>
<tr>
<th>CONFORMANCE CLASS 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTIFIER</td>
</tr>
<tr>
<td>REQUIREMENTS CLASS</td>
</tr>
<tr>
<td>TARGET TYPE</td>
</tr>
<tr>
<td>CONFORMANCE TESTS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

2.2. Conformance Class Core

<table>
<thead>
<tr>
<th>CONFORMANCE CLASS 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTIFIER</td>
</tr>
</tbody>
</table>
## CONFORMANCE CLASS 2

### REQUIREMENTS CLASS
- Requirements class 2: /req-class/core

### PREREQUISITE
- Requirements class 1: /req-class/data-model

### TARGET TYPE
- Implementation

### CONFORMANCE TESTS
- Conformance test A.7: /conf/core/media-type-impl
- Conformance test A.8: /conf/core/json-schema-impl
- Conformance test A.9: /conf/core/wkt-impl
- Conformance test A.10: /conf/core/wkb-impl
- Conformance test A.11: /conf/core/geojson-impl
- Conformance test A.12: /conf/core/allow-transformation-impl
- Conformance test A.13: /conf/core/crs-impl
- Conformance test A.14: /conf/core/axis-order-impl-1
- Conformance test A.15: /conf/core/axis-order-impl-2
- Conformance test A.16: /conf/core/error-reporting-impl
3

NORMATIVE REFERENCES
The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

**OGC Geospatial eXtensible Markup Language (GeoXACML) 3.0**, Draft OGC 22-049, OGC, 2023


4

TERMS AND DEFINITIONS
No terms and definitions are listed in this document.

All terms and definition can be found in GeoXACML 3.0.

The following JSON property names are defined for the Attribute element according to the name convention of JSON Profile of XACML 3.0 Version 1.1:

- SRID: Element property as defined in GeoXACML 3.0
- Precision: Element property as defined in GeoXACML 3.0
- AllowTransformation: Element property as defined in GeoXACML 3.0
5

CONVENTIONS
CONVENTIONS

This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

http://www.opengis.net/spec/geoxacml/3.0/json-profile/1.0

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.
INTRODUCTION TO GEOXACML 3.0 JSON PROFILE V1.0
The XACML Version 3.0 Standard defines an XML-based policy language and the structure for the encoding of Authorization Decision Request (ADR) and Authorization Decision (AD) in XML. A more lightweight and compact encoding in JSON is defined in JSON Profile of XACML 3.0 Version 1.1.

The GeoXACML 3.0 Standard supports the XML encoding of ADR and AD. This is because that feature is inherited from XACML Version 3.0. The direct use of JSON Profile of XACML 3.0 Version 1.1 is not possible, because it does not support the Geometry data-type as defined in OGC GeoXACML 3.0.

Also, declaring additional properties for the Attribute element is not possible which downgrades the geometry encodings to use the GeoXACML 3.0 default CRS urn:ogc:def:crs:OGC::CRS84.

To support the same expressiveness for encoding the ADR and AD in JSON, as it is possible for XML, this Profile defines the new media-type application/geoxacml+json which supports the GeoXACML 3.0 data-type urn:ogc:def:dataType:geoxacml:3.0:geometry and additional properties for the Attribute element.
7

GEOXACML 3.0 JSON PROFILE V1.0 REQUIREMENTS
GEOXACML 3.0 JSON PROFILE V1.0 REQUIREMENTS

This section defines the requirements to extend the JSON Profile of XACML 3.0 Version 1.1 for encoding ADR and AD in JSON. This profile is typically used with a GeoXACML 3.0 implementation as a service, hence supports the API conformance class.

7.1. Requirement Class Data Model (abstract)

The standardization target for this requirements class is Implementation Specification.

The Data Model Requirements Class defines additional properties for the <Attribute> element as defined in JSON Profile of XACML 3.0 Version 1.1.

### REQUIREMENTS CLASS 1: GEOXACML 3.0 JSON PROFILE V1.0 DATA MODEL

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/req-class/data-model</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBLIGATION</td>
<td>requirement</td>
</tr>
<tr>
<td>TARGET TYPE</td>
<td>Implementation Specification</td>
</tr>
<tr>
<td>CONFORMANCE CLASS</td>
<td>Conformance class 1: /conf/data-model</td>
</tr>
<tr>
<td>PREREQUISITES</td>
<td>GeoXACML 3.0 JSON Profile of XACML 3.0 Version 1.1</td>
</tr>
</tbody>
</table>
| NORMATIVE STATEMENTS | Requirement 1: /req/data-type  
|                    | Requirement 2: /req/axis-order  
|                    | Requirement 3: /req/geojson  
|                    | Requirement 4: /req/wkb  
|                    | Requirement 5: /req/wkt  
|                    | Requirement 6: /req/media-type  
|                    | Requirement 7: /req/srid  
|                    | Requirement 8: /req/precision  
<p>|                    | Requirement 9: /req/allow-transformation |</p>
<table>
<thead>
<tr>
<th>REQUIREMENT 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDENTIFIER</strong></td>
<td>/req/data-type</td>
</tr>
<tr>
<td><strong>INCLUDED IN</strong></td>
<td>Requirements class 1:/req-class/data-model</td>
</tr>
<tr>
<td><strong>STATEMENT</strong></td>
<td>This profile extends the DataType definition from JSON Profile of XACML 3.0 Version 1.1, §3.3.1 with the GeoXACML 3 data-type urn:ogc:def:dataType:geoxacml:3.0:geometry.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REQUIREMENT 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDENTIFIER</strong></td>
<td>/req/crs</td>
</tr>
<tr>
<td><strong>INCLUDED IN</strong></td>
<td>Requirements class 1:/req-class/data-model</td>
</tr>
<tr>
<td><strong>STATEMENT</strong></td>
<td>This profile inherits the default Coordinate Reference System (CRS) identifier (urn:ogc:def:crs:OGC::CRS84) as specified in [geoxacml3].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REQUIREMENT 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDENTIFIER</strong></td>
<td>/req/axis-order</td>
</tr>
<tr>
<td><strong>INCLUDED IN</strong></td>
<td>Requirements class 1:/req-class/data-model</td>
</tr>
<tr>
<td><strong>STATEMENT</strong></td>
<td>This profile inherits the axis order (longitude/latitude) as specified in [geoxacml3].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REQUIREMENT 4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDENTIFIER</strong></td>
<td>/req/media-type</td>
</tr>
<tr>
<td><strong>INCLUDED IN</strong></td>
<td>Requirements class 1:/req-class/data-model</td>
</tr>
<tr>
<td><strong>STATEMENT</strong></td>
<td>This profile defines the media-type application/geoxacml+json.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REQUIREMENT 5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDENTIFIER</strong></td>
<td>/req/geojson</td>
</tr>
</tbody>
</table>
REQUIREMENT 5

INCLUDED IN
Requirements class 1:/req-class/data-model

STATEMENT
This profile defines the value for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1 may use the GeoJSON Geometry object as defined in The GeoSON Format, §3.1 to represent a geometry value.

REQUIREMENT 6

IDENTIFIER /req/wkt

INCLUDED IN
Requirements class 1:/req-class/data-model

STATEMENT
This profile defines the value for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1 may use the WKT Geometry encoding as defined in OGC Simple Features to represent a geometry value.

REQUIREMENT 7

IDENTIFIER /req/wkb

INCLUDED IN
Requirements class 1:/req-class/data-model

STATEMENT
This profile defines the value for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1 may use the WKB Geometry encoding as hex-string as defined in OGC Simple Features to represent a geometry value.

REQUIREMENT 8

IDENTIFIER /req/srid

INCLUDED IN
Requirements class 1:/req-class/data-model

STATEMENT
This profile defines the additional element SRID for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1.

REQUIREMENT 9

IDENTIFIER /req/precision
**REQUIREMENT 9**

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/req-class/data-model</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDED IN</td>
<td>Requirements class 1: /req-class/data-model</td>
</tr>
<tr>
<td>STATEMENT</td>
<td>This profile defines the additional element Precision for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1.</td>
</tr>
</tbody>
</table>

**REQUIREMENT 10**

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/req/allow-transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDED IN</td>
<td>Requirements class 1: /req-class/data-model</td>
</tr>
<tr>
<td>STATEMENT</td>
<td>This profile defines the additional element AllowTransformation for the Attribute element defined in JSON Profile of XACML 3.0 Version 1.1, §3.3.1.</td>
</tr>
</tbody>
</table>

---

### 7.2. Requirements Class GeoXACML 3.0 JSON Profile v1.0 Core

The standardization target for this requirements class is Implementation.

**REQUIREMENTS CLASS 2: GEOXACML 3.0 JSON PROFILE V1.0 CORE**

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/req-class/core</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBLIGATION</td>
<td>requirement</td>
</tr>
<tr>
<td>TARGET TYPE</td>
<td>Implementation</td>
</tr>
<tr>
<td>CONFORMANCE CLASS</td>
<td>Conformance class 2: /conf/core</td>
</tr>
<tr>
<td>PREREQUISITE</td>
<td>Requirements class 1: /req-class/data-model</td>
</tr>
<tr>
<td>NORMATIVE STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>Requirement 11: /req/wkt-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 12: /req/wkb-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 13: /req/geojson-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 14: /req/crs-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 15: /req/axis-order-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 16: /req/media-type-impl</td>
<td></td>
</tr>
<tr>
<td>Requirement 17: /req/srid-impl</td>
<td></td>
</tr>
</tbody>
</table>
REQUIREMENTS CLASS 2: GEOXACML 3.0 JSON PROFILE V1.0 CORE

Requirement 18: /req/precision-impl
Requirement 19: /req/allow-transformation-impl
Requirement 20: /req/error-reporting-impl
Requirement 21: /req/json-schema-impl

REQUIREMENT 11

IDENTIFIER /req/wkt-impl
INCLUDED IN Requirements class 2: /req-class/core
STATEMENT An implementation SHALL support the Well Known Text (WKT) encoding as defined in OGC Simple Features for expressing geometry value in the Attribute value.

REQUIREMENT 12

IDENTIFIER /req/wkb-impl
INCLUDED IN Requirements class 2: /req-class/core
STATEMENT An implementation SHALL support the Well Known Binary (WKB) hexstring encoding as defined in OGC Simple Features for expressing geometry value in the Attribute value.

REQUIREMENT 13

IDENTIFIER /req/geojson-impl
INCLUDED IN Requirements class 2: /req-class/core
STATEMENT An implementation SHALL support the GeoJSON encoding as defined in The GeoSON Format for expressing geometry value in the Attribute value.

REQUIREMENT 14

IDENTIFIER /req/crs-impl
INCLUDED IN Requirements class 2: /req-class/core
REQUIREMENT 14

**STATEMENT**
An implementation SHALL use the default CRS for calculating the geometry coordinate values unless specified otherwise using the element SRID for the Attribute element defined in Clause 7.1, Requirement 8: /req/srid.

---

REQUIREMENT 15

**IDENTIFIER** /req/axis-order-impl

**INCLUDED IN**
Requirements class 2: /req-class/core

**STATEMENT**
An implementation SHALL use the default axis order for serialization of the geometry coordinate values unless specified otherwise, indicated by using the element SRID for the Attribute element defined in Clause 7.1, Requirement 8: /req/srid.

---

REQUIREMENT 16

**IDENTIFIER** /req/media-type-impl

**INCLUDED IN**
Requirements class 2: /req-class/core

**STATEMENT**
An implementation SHALL support the media-type application/geoxacml+json. The media type application/geoxacml+json SHALL be used in association with the HTTP Content-Type and Accept headers when sending Authorization Decision Request and asking to receive a GeoXACML 3.0 compliant Authorization Decision via HTTP transport.

---

REQUIREMENT 17

**IDENTIFIER** /req/srid-impl

**INCLUDED IN**
Requirements class 2: /req-class/core

**STATEMENT**
An implementation SHALL support the additional element SRID for the Attribute element defined in Clause 7.1, Requirement 8: /req/srid. An implementation SHALL set the SRID value to the integer that identifies the CRS which was used to calculate the geometry coordinate values.

---

REQUIREMENT 18

**IDENTIFIER** /req/precision-impl
REQUIREMENT 18

INCLUDED IN  Requirements class 2: /req-class/core

STATEMENT  This profile defines the additional element Precision for the Attribute element defined in Clause 7.1, Requirement 9: /req/precision.

REQUIREMENT 19

IDENTIFIER  /req/allow-transformation-impl

INCLUDED IN  Requirements class 2: /req-class/core

STATEMENT  This profile defines the additional element AllowTransformation for the Attribute element defined in Clause 7.1, Requirement 10: /req/allow-transformation.

REQUIREMENT 20

IDENTIFIER  /req/error-reporting-impl

INCLUDED IN  Requirements class 2: /req-class/core

STATEMENT  An implementation SHALL support the MissingAttributeDetail element as part of the StatusCode element when reporting an Indeterminate decision with value code urn:ogc:def:identifier:geoxacml:3.0:crs-error as defined in GeoXACML 3.0.

REQUIREMENT 21

IDENTIFIER  /req/json-schema-impl

INCLUDED IN  Requirements class 2: /req-class/core

STATEMENT  An implementation SHALL support the JSON request schema as defined below for the ADR structure when using the application/geoxacml+json media type.

```json
{
    "$schema": "http://json-schema.org/draft-06/schema",
    "$id": "Request-with-geometry.schema.json",
    "title": "JSON schema of Request object defined in GeoXACML 3.0 JSON Profile v1.0",
    "definitions": {
        "RequestReferenceType": {
```
"type": "object",
"properties": {
  "ReferenceId": {
    "type": "array",
    "items": {
      "description": "Each item is a Category/Id",
      "type": "string"
    },
    "minItems": 1
  }
},
"required": ["ReferenceId"],
"additionalProperties": false
},
"MultiRequestsType": {
  "type": "object",
  "properties": {
    "RequestReference": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/RequestReferenceType"
      },
      "minItems": 1
    }
  },
  "required": ["RequestReference"],
  "additionalProperties": false
},
"RequestType": {
  "type": "object",
  "properties": {
    "ReturnPolicyIdList": {
      "type": "boolean"
    },
    "CombinedDecision": {
      "type": "boolean"
    },
    "XPathVersion": {
      "type": "string"
    },
    "Category": {
      "type": "array",
      "items": {
        "$ref": "common-std-with-geometry.schema.json#/definitions/AttributeCategoryType"
      },
      "minItems": 1
    },
    "MultiRequests": {
      "$ref": "#/definitions/MultiRequestsType"
    }
  },
  "required": ["Category"],
  "additionalProperties": false
},
"type": "object"
"properties": {
  "Request": {
    "$ref": "#/definitions/RequestType"
  }
},
"required": [
  "Request"
],
"additionalProperties": false
}

Figure 1 — GeoXACML 3.0 JSON Profile Request schema

1

"AttributeType": {  
"type": "object",
"properties": {  
"AttributeId": {  
"type": "string",
"format": "uri-reference"
},  
"Issuer": {  
"type": "string"
},  
"IncludeInResult": {  
"type": "boolean"
},  
"DataType": {  
"type": "string",
"format": "uri-reference"
},  
"Value": {  
"$ref": "#/definitions/AttributeValue"  
},  
"SRID": {  
"type": "number"
},  
"AllowTransformation": {  
"type": "boolean"
},  
"Precision": {  
"type": "number"
}  
"required": [  
"AttributeId",
"Value"
],  
"additionalProperties": false
},
"AttributeCategoryType": {  
"type": "object",
"properties": {  
"CategoryId": {  
"type": "string",
"format": "uri-reference"
},  
"Id": {  
"type": "string"
},  
"Content": {  
"type": "string"
},  
"Attribute": {  
"type": "array",
"items": {  
"$ref": "#/definitions/AttributeType"
}  
"minItems": 0
}  
"required": [  
"CategoryId"
]}
"additionalProperties": false

"IdReferenceType": {
  "type": "object",
  "properties": {
    "Id": {
      "type": "string",
      "format": "uri-reference"
    },
    "Version": {
      "type": "string"
    }
  },
  "required": [
    "Id"
  ],
  "additionalProperties": false
}

Figure 2 — common-std-with-geometry.schema.json

{ "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://geojson.org/schema/Geometry.json",
  "title": "GeoJSON Geometry",
  "oneOf": [
    {
      "title": "GeoJSON Point",
      "type": "object",
      "required": [
        "type",
        "coordinates"
      ],
      "properties": {
        "type": {
          "type": "string",
          "enum": [
            "Point"
          ]
        },
        "coordinates": {
          "type": "array",
          "minItems": 2,
          "items": {
            "type": "number"
          }
        },
        "bbox": {
          "type": "array",
          "minItems": 4,
          "items": {
            "type": "number"
          }
        }
      }
    }
  ]
}

"title": "GeoJSON LineString",
"type": "object",
"required": [
  "type",
  "coordinates"
],
"properties": {
  "type": {
    "type": "string",
    "enum": ["LineString"]
  },
  "coordinates": {
    "type": "array",
    "minItems": 2,
    "items": {
      "type": "array",
      "minItems": 2,
      "items": {
        "type": "number"
      }
    }
  },
  "bbox": {
    "type": "array",
    "minItems": 4,
    "items": {
      "type": "number"
    }
  }
},

"title": "GeoJSON Polygon",
"type": "object",
"required": [
  "type",
  "coordinates"
],
"properties": {
  "type": {
    "type": "string",
    "enum": ["Polygon"]
  },
  "coordinates": {
    "type": "array",
    "items": {
      "type": "array",
      "minItems": 4,
      "items": {
        "type": "array",
        "minItems": 2,
        "items": {
          "type": "number"
        }
      }
    }
  }
}
"bbox": { "type": "array", "minItems": 4, "items": { "type": "number" } },
},
"title": "GeoJSON MultiPoint", "type": "object", "required": [ "type", "coordinates" ],
"properties": { "type": { "type": "string", "enum": [ "MultiPoint" ] }, "coordinates": { "type": "array", "items": { "type": "array", "minItems": 2, "items": { "type": "number" } } }, "bbox": { "type": "array", "minItems": 4, "items": { "type": "number" } } },
"title": "GeoJSON MultiLineString", "type": "object", "required": [ "type", "coordinates" ],
"properties": { "type": { "type": "string", "enum": [ "MultiLineString" ] }, "coordinates": { "type": "array", "items": { "type": "array", "minItems": 2, "items": { "type": "number" } } } },
"type": "array",
"minItems": 2,
"items": {
  "type": "number"
}
},
"bbox": {
  "type": "array",
  "minItems": 4,
  "items": {
    "type": "number"
  }
}
},
"title": "GeoJSON MultiPolygon",
"type": "object",
"required": [
  "type",
  "coordinates"
],
"properties": {
  "type": {
    "type": "string",
    "enum": [
      "MultiPolygon"
    ],
  },
  "coordinates": {
    "type": "array",
    "items": {
      "type": "array",
      "items": {
        "type": "array",
        "minItems": 4,
        "items": {
          "type": "array",
          "minItems": 2,
          "items": {
            "type": "number"
          }*
        }
      }
    }
  }
}
},
"bbox": {
  "type": "array",
  "minItems": 4,
  "items": {
    "type": "number"
  }
}
]
Figure 3 — GeoXACML 3.0 JSON Profile Geometry schema

3source https://geojson.org/schema/Geometry.json
8

MEDIA TYPES FOR ANY DATA ENCODING(S)
This Standard defines the following Media Type to be used for an Authorization Decision Request and Authorization Decision encoded according to this profile:

- application/geoxacml+json
ANNEX A (NORMATIVE) ABSTRACT TEST SUITE
# ANNEX A
## (NORMATIVE) ABSTRACT TEST SUITE

### A.1. Conformance Class Data Model

The purpose of the tests from this conformance class is to construct different ADRs that are send to an implementation compliant with the API Conformance Class.

| **CONFORMANCE TEST A.1** |
|----------------------------|-----------------------------|
| **IDENTIFIER**             | /conf/data-model/definition |
| **REQUIREMENTS**           | Requirement 1: /req/data-type  
Requirement 8: /req/srid  
Requirement 9: /req/precision  
Requirement 10: /req/allow-transformation |
| **INCLUDED IN**            | Conformance class 1: /conf/data-model |
| **INDIRECT PREREQUISITE**  | Conformance class 1: /conf/data-model |
| **INDIRECT**               | /conf/data-model |
| **TEST PURPOSE**           | To validate that the JSON schema for validating ADR contains the Attribute properties SRID, Precision, AllowTransformation. |
| **TEST-METHOD-TYPE**       | Manual Inspection |
| **TEST METHOD**            | Construct a JSON encoded ADR and validate that the Attribute element contains a valid geometry and the following elements: |
| **A**                      | SRID and verify that its value is of type Integer. |
| **B**                      | Precision and verify that its value is of type Integer. |
| **C**                      | AllowTransformation and verify that its value is of type Boolean. |
### CONFORMANCE TEST A.2

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/conf/data-model/crs-axis-order</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENTS</td>
<td>Requirement 2: /req/crs</td>
</tr>
<tr>
<td></td>
<td>Requirement 3: /req/axis-order</td>
</tr>
<tr>
<td>INCLUDED IN</td>
<td>Conformance class 1: /conf/data-model</td>
</tr>
<tr>
<td>INDIRECT PREREQUISITE</td>
<td>Conformance class 1: /conf/data-model</td>
</tr>
<tr>
<td>INDIRECT</td>
<td>/conf/data-model</td>
</tr>
<tr>
<td>TEST PURPOSE</td>
<td>To validate that a JSON encoded ADR uses the default CRS and axis-order when no SRID element is present.</td>
</tr>
<tr>
<td>TEST-METHOD-TYPE</td>
<td>Manual Inspection</td>
</tr>
<tr>
<td>TEST METHOD</td>
<td>Construct a JSON encoded ADR and have the Attribute value contain a geometry serialized in the default CRS (urn:ogc:def:crs: OGC::CRS84) and default axis order (longitude/latitude).</td>
</tr>
<tr>
<td>A</td>
<td>Verify that the coordinates of the geometry are calculated using urn:ogc:def:crs: OGC::CRS84.</td>
</tr>
<tr>
<td>B</td>
<td>Verify that the coordinate order uses longitude/latitude.</td>
</tr>
</tbody>
</table>

### CONFORMANCE TEST A.3

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/conf/data-model/media-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENT</td>
<td>Requirement 4: /req/media-type</td>
</tr>
<tr>
<td>INCLUDED IN</td>
<td>Conformance class 1: /conf/data-model</td>
</tr>
<tr>
<td>INDIRECT PREREQUISITE</td>
<td>Conformance class 1: /conf/data-model</td>
</tr>
<tr>
<td>INDIRECT</td>
<td>/conf/data-model</td>
</tr>
<tr>
<td>TEST PURPOSE</td>
<td>To validate that a HTTP POST request for sending a JSON encoded ADR uses the media type application/geoxacml+json.</td>
</tr>
<tr>
<td>TEST-METHOD-TYPE</td>
<td>Manual Inspection</td>
</tr>
</tbody>
</table>

---

**OPEN GEOSPATIAL CONSORTIUM 22-050**
## CONFORMANCE TEST A.3

**TEST METHOD**
Construct a HTTP POST request which body is a JSON encoded ADR compliant \[\_conf_data-model\_conf_data-model\] and \[\_conf_data-model\_conf_crs-axis-order\] and set the Content-Type header to value application/geoxacml+json.

**A**
Verify that the HTTP POST request uses media type application/geoxacml+json.

## CONFORMANCE TEST A.4

**IDENTIFIER**
/\conf/data-model/wkt

**REQUIREMENT**
Requirement 6: /req/wkt

**INCLUDED IN**
Conformance class 1: /conf/data-model

**INDIRECT PREREQUISITE**
Conformance class 1: /conf/data-model

**INDIRECT**
/\conf/data-model

**TEST PURPOSE**
To validate that a JSON encoded ADR contains an Attribute value that is a WKT compliant geometry.

**TEST-METHOD-TYPE**
Manual Inspection

**TEST METHOD**
Construct a JSON encoded ADR compliant \[\_conf_data-model\_conf_data-model\] and \[\_conf_data-model\_conf_crs-axis-order\] and set the Attribute value to WKT encoded geometry.

**A**
Verify that the Attribute value is valid WKT.

## CONFORMANCE TEST A.5

**IDENTIFIER**
/\conf/data-model/wkb

**REQUIREMENT**
Requirement 7: /req/wkb

**INCLUDED IN**
Conformance class 1: /conf/data-model

**INDIRECT PREREQUISITE**
Conformance class 1: /conf/data-model

**INDIRECT**
/\conf/data-model
CONFORMANCE TEST A.5

TEST PURPOSE
To validate that a JSON encoded ADR contains an Attribute value that is a WKB hexstring compliant geometry.

TEST-METHOD-TYPE
Manual Inspection

TEST METHOD
Construct a JSON encoded ADR compliant [_conf_data-model_conf_data-model] and[_conf_data-model_conf_crs-axis-order] and set the Attribute value to WKB hexstring encoded geometry.

A
Verify that the Attribute value is valid WKB hexstring.

CONFORMANCE TEST A.6

IDENTIFIER /conf/data-model/geojson

REQUIREMENT Requirement 5:/req/geojson

INCLUDED IN Conformance class 1:/conf/data-model

INDIRECT PREREQUISITE Conformance class 1:/conf/data-model

INDIRECT /conf/data-model

TEST PURPOSE To validate that a JSON encoded ADR contains an Attribute value that is a GeoJSON compliant geometry.

TEST-METHOD-TYPE Manual Inspection

TEST METHOD Construct a JSON encoded ADR compliant [_conf_data-model_conf_data-model] and[_conf_data-model_conf_crs-axis-order] and set the Attribute value to WKB hexstring encoded geometry.

A Verify that the Attribute value is valid GeoJSON geometry.

A.2. Conformance Class Core

CONFORMANCE TEST A.7

IDENTIFIER /conf/core/media-type-impl
## CONFORMANCE TEST A.7

| REQUIREMENTS   | Conformance class 2: /conf/core  
|                | Requirement 16: /req/media-type-impl |

| INCLUDED IN    | Conformance class 2: /conf/core |

| TEST PURPOSE   | To validate that the implementation accepts the media-type GeoXACML_JSON for HTTP headers Content-Type and Accept. |

| TEST-METHOD-TYPE | Postman |

| TEST METHOD      | Send the ADR constructed and verified in /conf/data-model/data-model via HTTP POST to the implementation's /decision endpoint (as defined in API Conformance Class of GeoXACML 3.0) and verify that the request was not rejected, e.g. with HTTP status code 415. |

| A               | Send the ADR with HTTP POST and Content-Type set to GeoXACML_JSON and verify that the response status code is not 415. |

| B               | Send the ADR with HTTP POST and Content-Type and Accept set to GeoXACML_JSON and verify that the response Content-Type is set to GeoXACML_JSON. |

## CONFORMANCE TEST A.8

| IDENTIFIER      | /conf/core/json-schema-impl |

| REQUIREMENTS    | Conformance class 2: /conf/core  
|                | Requirement 21: /req/json-schema-impl  
|                | Requirement 16: /req/media-type-impl  
|                | Requirement 17: /req/srid-impl  
|                | Requirement 18: /req/precision-impl  
|                | Requirement 19: /req/allow-transformation-impl |

| INCLUDED IN    | Conformance class 2: /conf/core |

| TEST PURPOSE   | To validate that the implementation processes an ADR compliant to this GeoXACML 3.0 JSON Profile v1.0 with no error. |

| TEST-METHOD-TYPE | Postman or OpenAPI |

| TEST METHOD      | Send the ADR constructed and verified in /conf/data-model/data-model via HTTP POST to the implementation's /decision endpoint (as defined in API Conformance Class of GeoXACML 3.0) and verify that the received response (the AD) does not indicate a processing error. |

| A               | Send the ADR constructed and verified in /conf/data-model/data-model with HTTP POST to the /decision endpoint and verify that the response does not contain an error. |
## CONFORMANCE TEST A.9

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/conf/core/wkt-impl</th>
</tr>
</thead>
</table>
| REQUIREMENTS   | Conformance class 2: /conf/core  
                    Requirement 11: /req/wkt-impl |
| INCLUDED IN    | Conformance class 2: /conf/core |
| TEST PURPOSE   | To validate that the implementation processes an ADR compliant to this GeoXACML 3.0 JSON Profile v1.0 with no error, when the Attribute value contains a WKT encoded geometry. |
| TEST-METHOD-TYPE | Postman or OpenAPI |
| TEST METHOD    | Send the ADR constructed and verified in /conf/data-model/wkt via HTTP POST to the implementation's /decision endpoint (as defined in API Conformance Class of GeoXACML 3.0) and verify that the received response (the AD) does not indicate a processing error.  
                    A Send the ADR constructed and verified in /conf/data-model/wkt with HTTP POST and Content-Type GeoXACML_JSON to the /decision endpoint and verify that the response does not contain an error. |

## CONFORMANCE TEST A.10

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/conf/core/wkb-impl</th>
</tr>
</thead>
</table>
| REQUIREMENTS   | Conformance class 2: /conf/core  
                    Requirement 12: /req/wkb-impl |
| INCLUDED IN    | Conformance class 2: /conf/core |
| TEST PURPOSE   | To validate that the implementation processes an ADR compliant to this GeoXACML 3.0 JSON Profile v1.0 with no error, when the Attribute value contains a WKT encoded geometry. |
| TEST-METHOD-TYPE | Postman or OpenAPI |
| TEST METHOD    | Send the ADR constructed and verified in /conf/data-model/wkb via HTTP POST to the implementation's /decision endpoint (as defined in API Conformance Class of GeoXACML 3.0) and verify that the received response (the AD) does not indicate a processing error.  
                    A Send the ADR constructed and verified in /conf/data-model/wkb with HTTP POST and Content-Type GeoXACML_JSON to the /decision endpoint and verify that the response does not contain an error. |

## CONFORMANCE TEST A.11

| IDENTIFIER     | /conf/core/geojson-impl |
### CONFORMANCE TEST A.11

**REQUIREMENTS**
- Conformance class 2: /conf/core
- Requirement 13: /req/geojson-impl

**INCLUDED IN**
- Conformance class 2: /conf/core

**TEST PURPOSE**
To validate that the implementation processes an ADR compliant to this GeoXACML 3.0 JSON Profile v1.0 with no error, when the Attribute value contains a WKT encoded geometry.

**TEST-METHOD-TYPE**
- Postman or OpenAPI

**TEST METHOD**

| A | Send the ADR constructed and verified in /conf/data-model/geojson via HTTP POST to the implementation’s /decision endpoint (as defined in API Conformance Class of GeoXACML 3.0) and verify that the received response (the AD) does not indicate a processing error. |

### CONFORMANCE TEST A.12

**IDENTIFIER**
/conf/core/allow-transformation-impl

**REQUIREMENTS**
- Conformance class 2: /conf/core
- Requirement 19: /req/allow-transformation-impl

**INCLUDED IN**
- Conformance class 2: /conf/core

**TEST PURPOSE**
To validate that the implementation honors the allowTransformation value.

**TEST-METHOD-TYPE**
- Postman or OpenAPI

**TEST METHOD**

| A | Verify that the implementation is compliant to conformance class CRS Transformation |
| B | Construct a test geometry |
| C | Construct a simple GeoXACML policy that compares two geometries (e.g. using geometry-equals): The first geometry is obtained from the ADR and the second geometry is obtained from the policy. Use the test geometry for the policy |
| D | Construct an ADR containing the test geometry and send the ADR to the implementation |
## CONFORMANCE TEST A.12

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Verify that the AD contains the desired decision and not an error</td>
</tr>
</tbody>
</table>

### CONFORMANCE TEST A.13

<table>
<thead>
<tr>
<th>Identifier</th>
<th>/conf/core/crs-impl</th>
</tr>
</thead>
</table>
| Requirements      | Conformance class 2: /conf/core  
|                   | Requirement 14: /req/crs-impl |
| Included In       | Conformance class 2: /conf/core |
| Test Purpose      | To validate that the implementation honors the default CRS CRS84. |
| Test Method Type  | Postman or OpenAPI |
| Test Method       | Use a GeoXACML policy that compares two geometries (e.g. using geometry-equals) where the policy geometry is using the default CRS and send an ADR with a geometry using the default CRS. An implementation that honors the default CRS should process the ARD with no errors.  
| A                | Construct a test geometry using the default CRS |
| B                | Construct a simple GeoXACML policy that compares two geometries (e.g. using geometry-equals): The first geometry is obtained from the ADR and the second geometry is obtained from the policy. Use the test geometry for the policy |
| C                | Construct an ADR containing the test geometry and send the ADR to the implementation |
| D                | Verify that the AD contains the desired decision and not an error |

### CONFORMANCE TEST A.14

<table>
<thead>
<tr>
<th>Identifier</th>
<th>/conf/core/axis-order-impl-1</th>
</tr>
</thead>
</table>
| Requirements      | Conformance class 2: /conf/core  
|                   | Requirement 15: /req/axis-order-impl |
| Included In       | Conformance class 2: /conf/core |
| Test Purpose      | To validate that the implementation honors the default axis-order Axis_Order. |
| Test Method Type  | Postman or OpenAPI |
| Test Method       | |
### CONFORMANCE TEST A.14

**TEST METHOD**

Use a GeoXACML policy that compares two geometries (e.g. using `geometry-equals`) where the policy geometry is using the default CRS and default axis-order and send an ADR with a geometry using the default CRS and default axis-order. An implementation that honors the default axis-order should process the ADR with no errors.

**A**

Construct a test geometry using the default CRS and axis-order

**B**

Construct a simple GeoXACML policy that compares two geometries (e.g. using `geometry-equals`): The first geometry is obtained from the ADR and the second geometry is obtained from the policy. Use the test geometry for the policy

**C**

Construct an ADR containing the test geometry and send the ADR to the implementation

**D**

Verify that the AD contains the desired decision and not an error

### CONFORMANCE TEST A.15

**IDENTIFIER**

`/conf/core/axis-order-impl-2`

**REQUIREMENTS**

Conformance class 2: `/conf/core`
Requirement 15: `/req/axis-order-impl`

**INCLUDED IN**

Conformance class 2: `/conf/core`

**TEST PURPOSE**

To validate that the implementation honors the default axis-order `Axis_Order`.

**TEST-METHOD-TYPE**

Postman or OpenAPI

**TEST METHOD**

Repeat test `/conf/core/axis-order-impl-1` where the geometry in the ADR has swapped axis.

**A**

Construct a test geometry using the default CRS and axis-order

**B**

Construct a simple GeoXACML policy that compares two geometries (e.g. using `geometry-equals`): The first geometry is obtained from the ADR and the second geometry is obtained from the policy. Use the test geometry for the policy with swapped coordinates

**C**

Construct an ADR containing the test geometry and send the ADR to the implementation

**D**

Verify that the AD contains the desired decision and not an error
## CONFORMANCE TEST A.16

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>/conf/core/error-reporting-impl</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENTS</td>
<td>Conformance class 2: /conf/core</td>
</tr>
<tr>
<td></td>
<td>Requirement 20: /req/error-reporting-impl</td>
</tr>
<tr>
<td>INCLUDED IN</td>
<td>Conformance class 2: /conf/core</td>
</tr>
<tr>
<td>TEST PURPOSE</td>
<td>To validate that the implementation provides GeoXACML geometry specific error information.</td>
</tr>
<tr>
<td>TEST-METHOD-TYPE</td>
<td>Postman or OpenAPI</td>
</tr>
<tr>
<td>TEST METHOD</td>
<td>Verify that the implementation supports the encoding of the MissingAttributeDetail as defined in GeoXACML 3.0.</td>
</tr>
</tbody>
</table>

A

Instantiate the implementation with a GeoXACML policy that returns an Indeterminate decision response for any request where the geometry value is not encoded using the default CRS. Send such a JSON encoded ADR with HTTP Content-Type and Accept headers set to application/geoxacml+json to the implementation. Evaluate the response and in particular verify that the response has status Indeterminate and that there is a MissingAttributeElement encoded in JSON.

DESCRIPTION

**NOTE** In principle, this test in JSON should result in the same level of expressiveness as the equivalent test conducted in XML.
ANNEX B (INFORMATIVE) EXAMPLES FOR THE GEOXACML 3.0 JSON PROFILE V1.0
ANNEX B
(INFORMATIVE)
EXAMPLES FOR THE GEOXACML 3.0 JSON PROFILE V1.0

The following sections illustrate the use of the GeoXACML 3.0 JSON Profile v1.0.

B.1. Examples how to encode Geometry in ADR

```json
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": "POINT(-77.035278 38.889444)"
  }
}
```

Figure B.1 — Geometry encoding in WKT with default CRS

```json
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": "010100000000000000000000400000000000001040"
  }
}
```

Figure B.2 — Geometry encoding in WKB with default CRS

```json
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": [
      "POINT(-77.035278 38.889444)",
      "Point (-122.4538755 37.8106729)"
    ]
  }
}
```

Figure B.3 — Geometry bag encoding in WKT with default CRS
Figure B.4 — Homogeneous Geometry Collection encoding in WKT with default CRS

```
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": "GEOMETRYCOLLECTION(POINT(-77.035278 38.889444), Point (-122.4538755 37.8106729))"
  }
}
```

Figure B.5 — Geometry encoding in GeoJSON with default CRS

```
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": {
      "type": "Point",
      "coordinates": [-77.035278, 38.889444]
    }
  }
}
```

Figure B.6 — Geometry bag encoding in GeoJSON with default CRS

```
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Value": [
      {
        "type": "Point",
        "coordinates": [-77.035278, 38.889444]
      },
      {
        "type": "Point",
        "coordinates": [-77.035278, 38.889444]
      }
    ]
  }
}
```

Figure B.7 — Geometry encoding in WKT with CRS EPSG:3857

```
{
  "Attribute": {
    "AttributeId": "subject-location",
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "SRID": 3857,
    "Value": "POINT(-8571600.791082066 4579425.812870098)"
  }
}
```
Figure B.8 — Geometry encoding in WKT with precision of 4 decimal places

Figure B.9 — Geometry encoding in WKT with allowTransformation=true

B.2. Example GeoXACML 3.0 policy, request and response

Figure B.10 — Request example using GeoXACML 3.0 JSON schema
"StatusCode": { 
  "Value": "urn:ogc:def:function:geoxacml:3.0:crs-error"
},  
"StatusMessage": "Geometry must be encoded using specified CRS",
"StatusDetail": { 
  "MissingAttributeDetail": { 
    "DataType": "urn:ogc:def:dataType:geoxacml:3.0:geometry",
    "Category": "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject",
    "SRID": 3857,
    "AttributeId": "subject-location"
  }
},  
"Decision": "Indeterminate"
]

Figure B.11 — Response example using GeoXACML
3.0 JSON schema including MissingAttributeDetail

NOTE The Response above can be received using the GeoXACML 3.0 policy below.

```xml
xmns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns6="http://www.w3.org/2005/Atom"
xmns:ns5="http://authzforce.github.io/core/xmns/pdp/8"
xmns:ns4="http://authzforce.github.io/pap-dao-flat-file/xmns/properties/3.6"
xmns:ns3="http://authzforce.github.io/rest-api-model/xmns/authz/5"
PolicySetId="root"
Version="1"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:3.0:policy-combining-algorithm:deny-overrides">
  <xacml3:Target />
  <xacml3:Policy PolicyId="urn:ogc:geoxacml:3.0:conformance-test:core:policy:geometry-encoding"
Version="1"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides">
    <xacml3:Description>http://www.opengis.net/spec/GEOXACML/3.0/Core/conf/function-equals/support-valid</xacml3:Description>
    <xacml3:Target />
    <xacml3:Rule RuleId="precision6" Effect="Permit">
      <xacml3:Target>
        <xacml3:AnyOf>
          <xacml3:AllOf>
            <xacml3:Match MatchId="urn:ogc:def:function:geoxacml:3.0:geometry-has-precision">
              <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">
                4</xacml3:AttributeValue>
            </xacml3:MatchDesignator>
            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            AttributeId="subject-location" DataType="urn:ogc:def:dataType:geoxacml:3.0:geometry"
            MustBePresent="true"
          </xacml3:AllOf>
        </xacml3:AnyOf>
      </xacml3:Target>
    </xacml3:Rule>
  </xacml3:Policy>
</xacml3:PolicySet>
```
Figure B.12 — GeoXACML 3.0 policy example causing a MissingAttributeDetail response when the request geometry is not in CRS84 and AllowTransformation=false
ANNEX C (INFORMATIVE) REVISION HISTORY
## ANNEX C

(INFORMATIVE)

REVISION HISTORY

Table C.1

<table>
<thead>
<tr>
<th>DATE</th>
<th>RELEASE</th>
<th>EDITOR</th>
<th>PRIMARY CLAUSES MODIFIED</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-11-08</td>
<td>0.1</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>initial version</td>
</tr>
<tr>
<td>2022-12-19</td>
<td>0.2</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>support for additional properties in Attribute element; JSON schema added</td>
</tr>
<tr>
<td>2022-12-22</td>
<td>0.3</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>simplification of requirements classes and conformance classes</td>
</tr>
<tr>
<td>2023-01-12</td>
<td>0.4</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>align requirements, requirements classes, conformance tests using the new Metanorma annotations</td>
</tr>
<tr>
<td>2023-01-13</td>
<td>0.5</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>applied OGC NA-Policy to Metanorma annotations</td>
</tr>
<tr>
<td>2023-02-06</td>
<td>0.6</td>
<td>Andreas Matheus</td>
<td>all</td>
<td>incorporated Carl Reed’s comments</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY