Authorization for a Service-based Geospatial Data Infrastructure

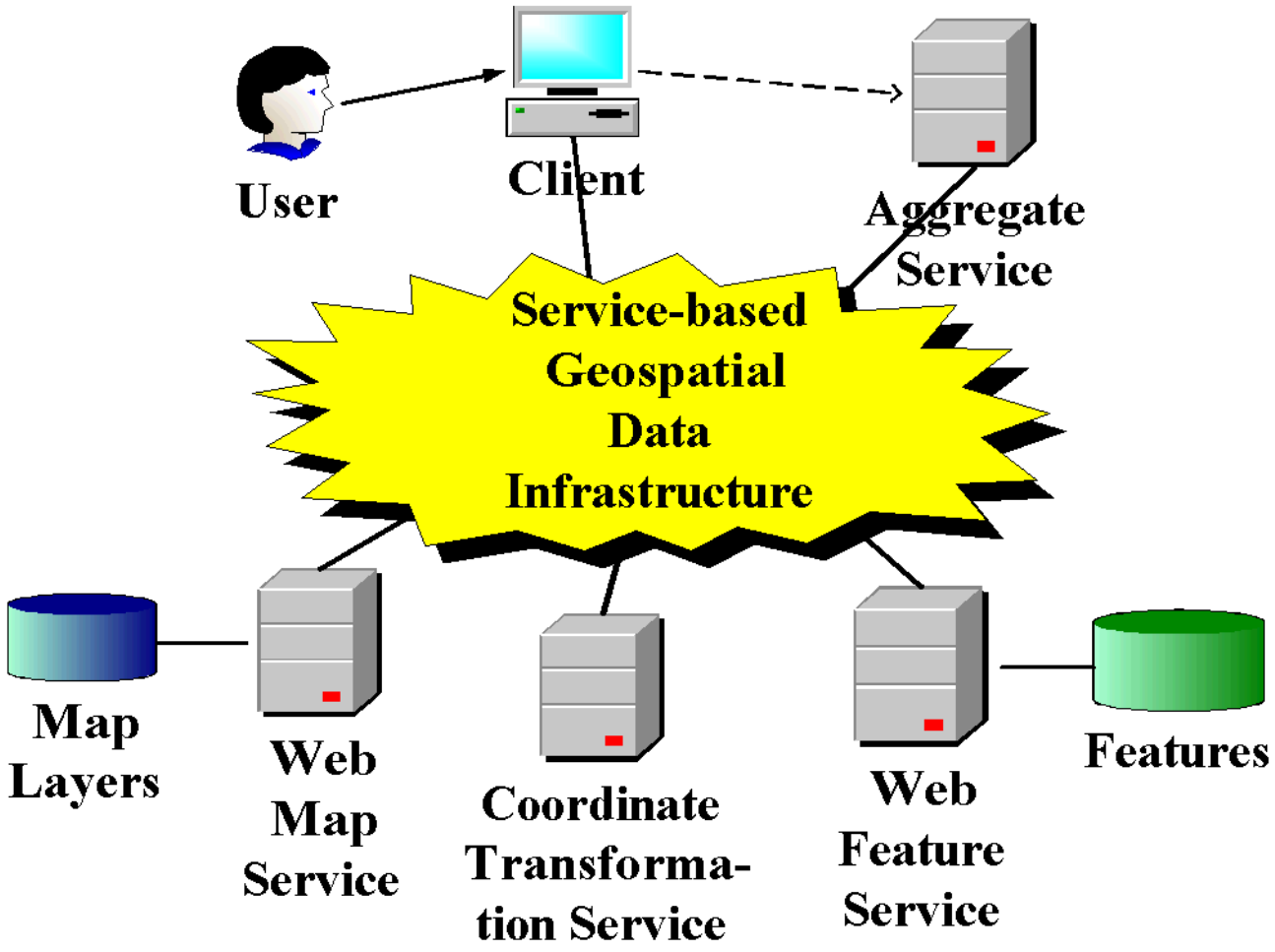
Andreas Matheus
Technische Universität München
matheus@in.tum.de

January 17, 2005

Overview for this Presentation

1. Service-based Geospatial Data Infrastructure (SGDI) and authorization
2. Access control requirements for geospatial information objects (GML feautres)
3. Declaration of access restrictions
4. Enforcement of declared access restrictions
5. Example restrictions
6. Online demonstration for restricting access to WMS
7. Spatial Authorization $\Leftrightarrow$ GeoDRM
8. Conclusion
9. Outlook

An example Service-based Geospatial Data Infrastructure

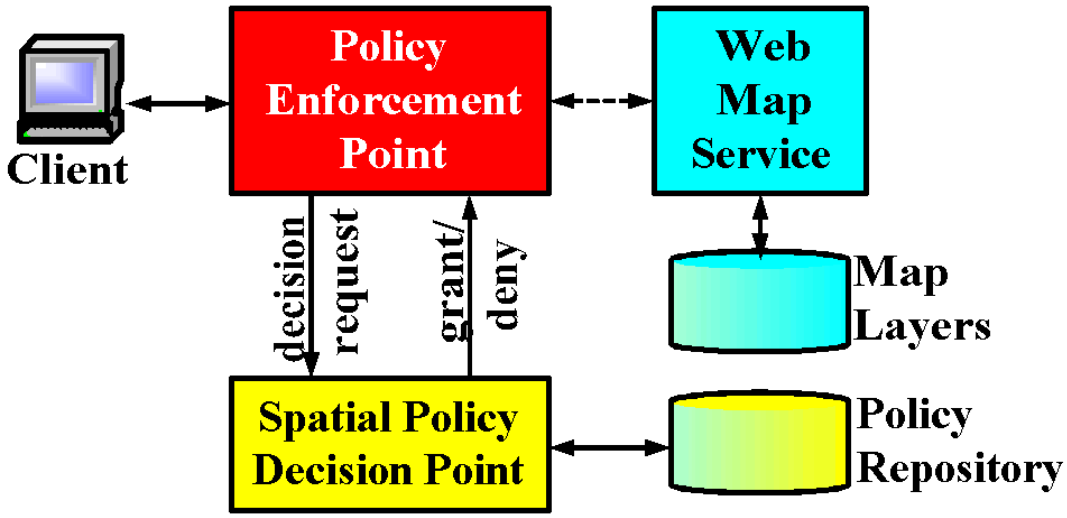


Authorization for Distributed Services

- Requires standardization
- Major companies provides OGC interface implementation
- If using OGC certified services for establishing a SGDI, authorization is required

This presentation introduces an approach how to declare and enforce authorization that can also be used for OGC services

Example extension to a Web Map Service providing authorization



- PEP mirrors the interfaces of the Web Map Service
- PEP creates authorization decision request to the spatial PDP and enforces the authorization decision
- PDP derives an authorization decision based on decision request and encoded restrictions

Authorization Requirements*

class-based restrictions: Allows to restrict access to object instances of the same class (feature type)

object-based restrictions: Allows to restrict access to individual objects, selected by non-spatial properties (individual features)

spatial restrictions: Allows to restrict access to individual objects, selected by their geometry

*Most important from poll at Intergeo 2002

Declaration of Permissions using XACML*

XACML is based on a Rule-based authorization model

XACML capabilities support encoding of restrictions of type

class-based: O.K. for XML/GML encoding

object-based: O.K. for XML/GML encoding

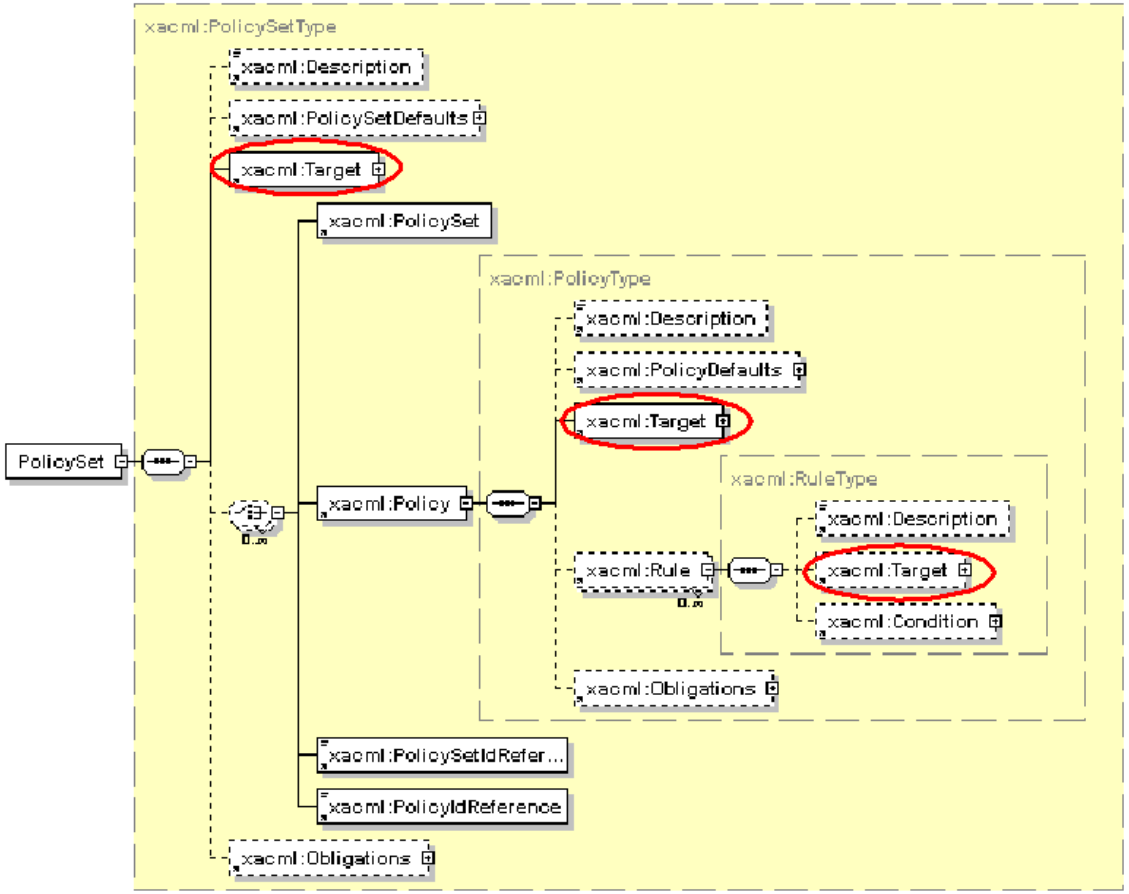
spatial: Not supported $\Rightarrow$ GeoXACML

GeoXACML

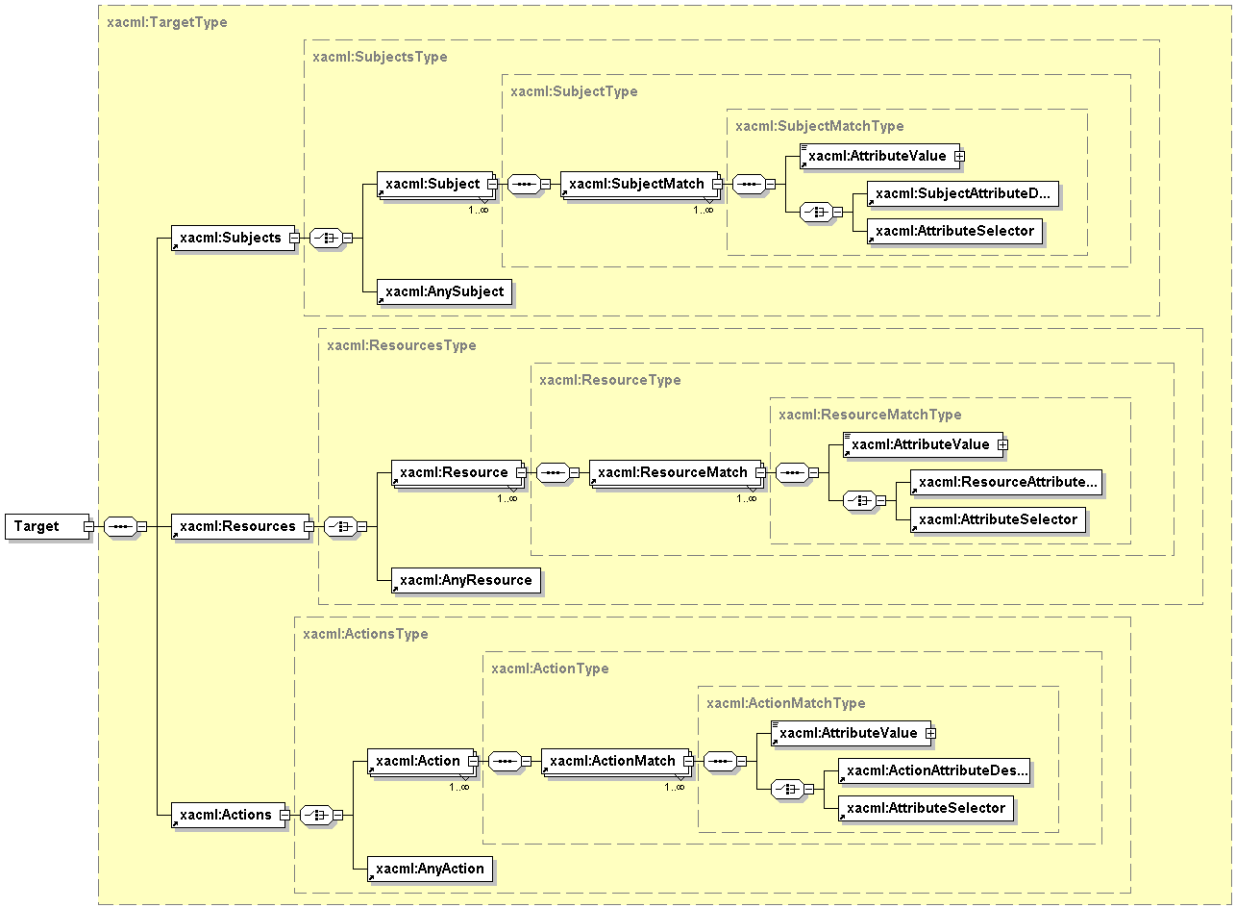
- Developed in dissertation
- Not a standard. But possibly standardized through OGC?

*eXtensible Access Control Markup Language, standard by OASIS

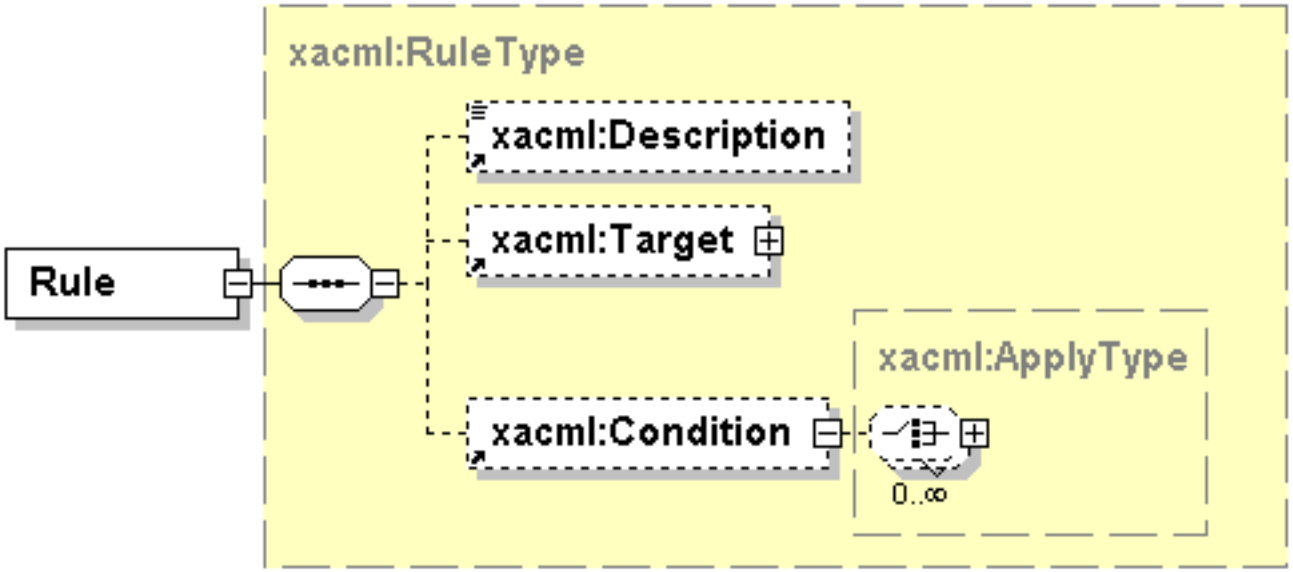
XACML Policy File Structure



XACML: The <Target> element



XACML: The <Rule> element



- Effect = {Deny, Permit}
- Output = {Deny, Permit, NotApplicable, Indeterminate}

GeoXACML* = XACML plus Geospatial specifics

Geospatial datatypes:

- URN-prefeix: `"http://www.opengis.net/gml#"`
- Datatypes: `PointAttribute, LineStringAttribute, LinearRingAttribute, BoxAttribute, PolygonAttribute`
- Example: `"http://www.opengis.net/gml#PointAttribute"`

Topological relations:

- URN-prefix:`"urn:oasis:names:tc:geoxacml:1.0:function:"`
- Methods: `disjoint, touches, crosses, within, overlaps, intersects, equals, contains`
- Example: `"urn:oasis:names:tc:geoxacml:1.0:function:within"`

*Geospatial eXtensible Access Control Markup Language

GeoXACML* = XACML plus Geospatial specifics

Combining algorithms:

- URN-prefix: `"urn:oasis:names:tc:geoxacml:1.0:rule-combining-algorithm:"`
- Name of algorithm: `or, and`
- Example: `"urn:oasis:names:tc:geoxacml:1.0:rule-combining-algorithm:and"`

*Geospatial eXtensible Access Control Markup Language

Declaration of Class-based (Feature Type) Permissions

$$Rule^C := \{SM_R, OM_R, RM_R, XM_R^C, C^C\} \rightarrow \{Deny, Permit\} \quad (1)$$

$$C^C := \begin{cases} \epsilon & \rightarrow True \\ \{\mathsf{XF}, XM_C^C, \mathcal{V}\} & \rightarrow \begin{cases} True, & \mathsf{XF}(\mathsf{XM}_\mathsf{C}^\mathsf{C}) \equiv \mathcal{V} \\ False, & else \end{cases} \end{cases} \quad (2)$$

Rule^C	Rule, encoding the class-based restriction
C^C	Condition for declaring a class-based restriction
SM_R	Subject matching expression of a Rule
OM_R	Operation matching expression of a Rule
RM_R	Resource matching expression of a Rule
$\mathsf{XM}_\mathsf{R}^\mathsf{C}$	Xpath expression of a Rule for matching resource content objects that represent a class, resp. a GML feature type
$\mathsf{XM}_\mathsf{C}^\mathsf{C}$	Xpath expression of a Condition for matching resource content objects that represent a class, resp. a GML feature type
XF	Xpath function
$\mathcal{V}$	Set of values

Declaration of Object-based (Feature) Permissions

$$Rule^O = \{SM_R, OM_R, RM_R, XM_R^C, C^O\} \rightarrow \{Deny, Permit\} \quad (3)$$

$$C^O = \{XF, XM_C^O, \mathcal{V}\} \rightarrow \begin{cases} True, & XF(XM_C^O) \equiv \mathcal{V} \\ False, & else \end{cases} \quad (4)$$

$Rule^O$	Rule, encoding an object-based restrictions
C^O	Condition, expressing a matching condition of non-spatial characteristics of objects, resp. features
SM_R	Subject matching expression of a Rule
OM_R	Operation matching expression of a Rule
RM_R	Resource matching expression of a Rule
XM_C^O	Xpath expression of a Condition for matching objects, resp. GML features according to their non-spatial characteristics
XF	Xpath function
$\mathcal{V}$	Tuple of values, used for comparison

Declaration of Object-based (Feature) Permissions

$$Rule^S := \{SM_R, OM_R, RM_R, XM_R^C, C^S\} \rightarrow \{Deny, Permit\} \quad (5)$$

$$C^S := \{XM_C^S, TCF, G_P\} = \{TCF(G_R, G_P)\} \quad (6)$$

$$TCF \in \{disjoint, touches, crosses, within, overlaps, intersects, equals, contains, \neg disjoint, \neg touches, \neg crosses, \neg within, \neg overlaps, \neg intersects, \neg equals, \neg contains\} \quad (7)$$

$$, \quad G_P := \texttt{Polygon} \quad (8)$$

$Rule^S$	Rule, encoding a spatial restriction
C^S	Condition, restricting the matching on spatial characteristics of objects, resp. features
SM_R	Subject matching expression of a Rule
OM_R	Operation matching expression of a Rule
RM_R	Resource matching expression of a Rule
XM_C	Xpath expression of a Condition
XM_C^S	Xpath expression of a Condition for matching objects, resp. GML features according to their non-spatial characteristics
TCF	Topological condition function, such as Disjoint, Touches, Crosses, Within, Overlaps, Intersects or Equals
G_R	Resource Object geometry
G_P	Permission geometry

Example Class-based Permission

- Any subject can read features if not containing features of type `am:Building`

$R^C = \{*, read, \epsilon, //*, C^C\} \rightarrow Permit$
$C^C = \{count,$ /am:CityModel/gml:featureMember/am:Building, $\{0\}\}$

Example Class-based Permission

- Any subject can read features if not containing features of type `am:Building`

$$R^C = \{*, read, \epsilon, //*, C^C\} \rightarrow Permit$$
$$C^C = \{count, \text{/am:CityModel/gml:featureMember/am:Building}, \{0\}\}$$

```
1 <Subjects>
2   <AnySubject/>
3 </Subjects>
```

Example Class-based Permission

- Any subject can read features if not containing features of type `am:Building`

$$R^C = \{*, read, \epsilon, //*, C^C\} \rightarrow Permit$$
$$C^C = \{count, \text{/am:CityModel/gml:featureMember/am:Building}, \{0\}\}$$

```
1 <Resources>
2   <Resource>
3     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
4       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
5       <AttributeSelector RequestContextPath="count(///am:CityModel/gml:featureMember/am:Building)"
6        DataType="http://www.w3.org/2001/XMLSchema#integer"/>
7     </ResourceMatch>
8   </Resource>
9 </Resources>
```

Example Class-based Permission

- Any subject can read features if not containing features of type `am:Building`

$$R^C = \{*, read, \epsilon, //*, C^C\} \rightarrow Permit$$
$$C^C = \{count, \text{/am:CityModel/gml:featureMember/am:Building}, \{0\}\}$$

```
1 <Actions>
2   <Action>
3     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
4       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
5       <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
6        DataType="http://www.w3.org/2001/XMLSchema#string"/>
7     </ActionMatch>
8   </Action>
9 </Actions>
```

Example Object-based Permissions

- Bob can read the features, which address is ”5 Street D”

$$R^O = \{Bob, read, \epsilon, \text{//*/am:Building}, C^O\} \rightarrow Permit$$
$$C^O = \{equal, \text{./am:address=”5 Street D”}, \{True\}\}$$

Example Object-based Permissions

- Bob can read the features, which address is ”5 Street D”

$$R^O = \{Bob, read, \epsilon, \text{//*/am:Building}, C^O\} \rightarrow Permit$$
$$C^O = \{equal, \text{./am:address=”5 Street D”}, \{True\}\}$$

```
1  <Subjects>
2    <Subject>
3      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
4        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Bob</AttributeValue>
5        <SubjectAttributeDesignator
6         SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
7         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
8         DataType="http://www.w3.org/2001/XMLSchema#string"/>
9      </SubjectMatch>
10   </Subject>
11 </Subjects>
```

Example Object-based Permissions

- Bob can read the features, which address is ”5 Street D”

$$R^O = \{Bob, read, \epsilon, \text{//*/am:Building}, C^O\} \rightarrow Permit$$
$$C^O = \{equal, \text{./am:address=”5 Street D”}, \{True\}\}$$

```
1 <Resources>
2   <Resource>
3     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal">
4       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">True</AttributeValue>
5       <AttributeSelector RequestContextPath="//am:Building/am:Address='5 Street D'"
6        DataType="http://www.w3.org/2001/XMLSchema#string"/>
7     </ResourceMatch>
8   </Resource>
9 </Resources>
```

Example Object-based Permissions

- Bob can read the features, which address is ”5 Street D”

$$R^O = \{Bob, read, \epsilon, \text{//*/am:Building}, C^O\} \rightarrow Permit$$
$$C^O = \{equal, \text{./am:address=”5 Street D”}, \{True\}\}$$

```
1 <Actions>
2   <Action>
3     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
4       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
5       <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
6        DataType="http://www.w3.org/2001/XMLSchema#string"/>
7     </ActionMatch>
8   </Action>
9 </Actions>
```

Example simple Spatial Permission

- Bob can read features of type `Building` if within a given permission area, encoded by the `Polygon`=$\{$foo*,0 0,10 0,10 4,0 4,0 0$\}$

$$R^S = \{Bob, read, foo, \text{//am:Building}, C^S\} \rightarrow Permit$$
$$C^S = \{\text{./am:shape}, within, \{\text{foo,0 0,10 0,10 4,0 4,0 0}\}\}$$

*This example does not use a known Coordinate Reference System

Example simple Spatial Permission

- Bob can read features of type `Building` if within a given permission area, encoded by the `Polygon`={foo,0 0,10 0,10 4,0 4,0 0}

$$R^S = \{Bob, read, foo, \text{//am:Building}, C^S\} \rightarrow Permit$$
$$C^S = \{\text{./am:shape}, within, \{\text{foo,0 0,10 0,10 4,0 4,0 0}\}\}$$

```
1  <Subjects>
2    <Subject>
3      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
4        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Bob</AttributeValue>
5        <SubjectAttributeDesignator
6          SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
7          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
8          DataType="http://www.w3.org/2001/XMLSchema#string"/>
9      </SubjectMatch>
10   </Subject>
11 </Subjects>
```

Example simple Spatial Permission

- Bob can read features of type `Building` if within a given permission area, encoded by the `Polygon`={foo,0 0,10 0,10 4,0 4,0 0}

$$R^S = \{Bob, read, foo, \text{//am:Building}, C^S\} \rightarrow Permit$$
$$C^S = \{\text{./am:shape}, within, \{\text{foo,0 0,10 0,10 4,0 4,0 0}\}\}$$

```
1  <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2    <Function FunctionId="urn:oasis:names:tc:geoxacml:1.0:function:within"/>
3      <AttributeValue DataType="http://www.opengis.net/gml#polygon">
4        <gml:Polygon gid="P2" srsName="foo" xmlns:gml="http://www.opengis.net/gml">
5          <gml:outerBoundaryIs>
6            <gml:LinearRing>
7              <gml:coordinates ts="," cs=" ">0 0,10 0,10 4,0 4,0 0</gml:coordinates>
8            </gml:LinearRing>
9          </gml:outerBoundaryIs>
10       </gml:Polygon>
11     </AttributeValue>
12   <AttributeSelector RequestContextPath="//am:CAPITALS/gml:Point"
13    DataType="http://www.opengis.net/gml#point"/>
14 </Condition>
```

Example simple Spatial Permission

- Bob can read features of type `Building` if within a given permission area, encoded by the `Polygon`={foo,0 0,10 0,10 4,0 4,0 0}

$$R^S = \{Bob, read, foo, \text{//am:Building}, C^S\} \rightarrow Permit$$
$$C^S = \{\text{./am:shape}, within, \{\text{foo,0 0,10 0,10 4,0 4,0 0}\}\}$$

```
1 <Actions>
2   <Action>
3     <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
4       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
5       <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
6        DataType="http://www.w3.org/2001/XMLSchema#string"/>
7     </ActionMatch>
8   </Action>
9 </Actions>
```

Example complex Spatial Permission

- Alice can read features of type `Building` if not within permission area G_{P1} and G_{P2}

$$
\begin{aligned}
P1 =& \{*, *, \epsilon, //*, and, R1, R2\}^{*} \\
R1 =& \{Alice, read, \epsilon, \text{//am:Building}, C1\} \rightarrow Permit \\
C1 =& \{\text{./am:location}, \neg within, G_{P1}\} \\
R2 =& \{Alice, read, \epsilon, \text{//am:Building}, C2\} \rightarrow Permit \\
C2 =& \{\text{./am:location}, \neg within, G_{P2}\}
\end{aligned}
$$

*The Policy combines multiple Rules, using the combining algorithm and

Example complex Spatial Permission

- Alice can read features of type `Building` if not within permission area G_{P1} and G_{P2}

$$
\begin{aligned}
P1 =& \{*, *, \epsilon, //*, and, R1, R2\}^{*} \\
R1 =& \{Alice, read, \epsilon, \text{//am:Building}, C1\} \rightarrow Permit \\
C1 =& \{\text{./am:location}, \neg within, G_{P1}\} \\
R2 =& \{Alice, read, \epsilon, \text{//am:Building}, C2\} \rightarrow Permit \\
C2 =& \{\text{./am:location}, \neg within, G_{P2}\}
\end{aligned}
$$

```
1  <Policy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   PolicyId="urn:diss:complex-spatial-within:policy-1"
3   RuleCombiningAlgId="urn:oasis:names:tc:geoxacml:1.0:rule-combining-algorithm:and">
4     ...
5     <Rule RuleId="diss:complex-spatial-within:rule1" Effect="Permit">
6     ...
```

*The Policy combines multiple Rules, using the combining algorithm and

Example complex Spatial Permission

- Alice can read features of type `Building` if not within permission area G_{P1} and G_{P2}

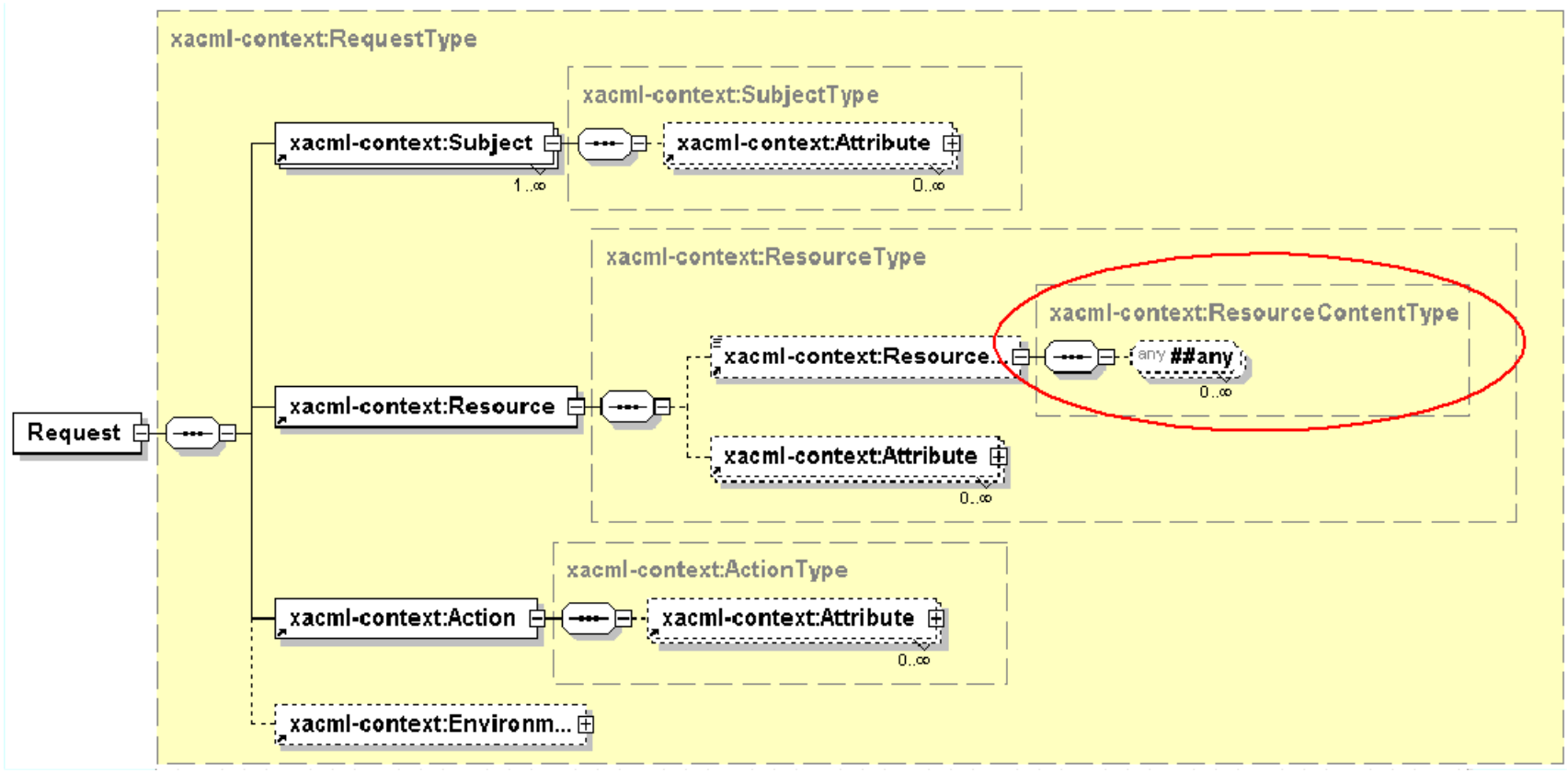
```
 6   <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
 7     <Function FunctionId="urn:oasis:names:tc:geoxacml:1.0:function:within"/>
 8     <AttributeValue DataType="http://www.opengis.net/gml#polygon">
 9       <gml:Polygon gid="P1" srsName="foo" xmlns:gml="http://www.opengis.net/gml">
10         <gml:outerBoundaryIs>
11           <gml:LinearRing>
12             <gml:coordinates>coordinates of G_P1</gml:coordinates>
13           </gml:LinearRing>
14         </gml:outerBoundaryIs>
15       </gml:Polygon>
16     </AttributeValue>
17     <AttributeSelector RequestContextPath="//am:Building/am:location"
18      DataType="http://www.opengis.net/gml#point"/>
19   </Condition>
20   </Rule>
21   <Rule RuleId="diss:complex-spatial-within:rule2" Effect="Permit">
22   ...
```

Example complex Spatial Permission

- Alice can read features of type `Building` if not within permission area G_{P1} and G_{P2}

```
22    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
23      <Function FunctionId="urn:oasis:names:tc:geoxacml:1.0:function:within"/>
24      <AttributeValue DataType="http://www.opengis.net/gml#polygon">
25        <gml:Polygon gid="P2" srsName="foo" xmlns:gml="http://www.opengis.net/gml">
26          <gml:outerBoundaryIs>
27            <gml:LinearRing>
28              <gml:coordinates>coordinates of G_P2</gml:coordinates>
29            </gml:LinearRing>
30          </gml:outerBoundaryIs>
31        </gml:Polygon>
32      </AttributeValue>
33      <AttributeSelector RequestContextPath="//am:Building/am:location"
34       DataType="http://www.opengis.net/gml#point"/>
35    </Condition>
36    </Rule>
37  </Policy>
```

XACML Schema for the Authorization Decision Request



A XML Schema Example for a Possible Resource Content

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <xs:schema targetNamespace="http://www.in.tum.de/am" xmlns="http://www.w3.org/2001/XMLSchema"
 3     xmlns:am="http://www.in.tum.de/am" xmlns:gml="http://www.opengis.net/gml" ... >
 4 <xs:import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
 5 <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlinks.xsd"/>
 6
 7 <xs:element name="CityModel" type="am:CityModelType" substitutionGroup="gml:_FeatureCollection"/>
 8 <xs:element name="Building" type="am:BuildingType" substitutionGroup="gml:_Feature"/>
 9 <xs:complexType name="CityModelType">
10   <xs:complexContent>
11     <xs:extension base="gml:AbstractFeatureCollectionType"/>
12   </xs:complexContent>
13 </xs:complexType>
14 <xs:complexType name="BuildingType">
15   <xs:complexContent>
16     <xs:extension base="gml:AbstractFeatureType">
17       <xs:sequence minOccurs="0">
18         <xs:element name="Address"/>
19         <xs:element name="Shape" type="gml:PolygonType"/>
20       </xs:sequence>
21     </xs:extension>
22   </xs:complexContent>
23 </xs:complexType>
24 </xs:schema>
```

A XML Example of a Possible Resource Content

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <CityModel xmlns="http://www.in.tum.de/am" xmlns:xlink="http://www.w3.org/1999/xlink"
 3            xmlns:gml="http://www.opengis.net/gml" xmlns:am="http://www.in.tum.de/am"
 4            xsi:schemaLocation="http://http://www.in.tum.de/am CityModel.xsd"
 5            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" fid="CityModel">
 6   <gml:boundedBy>
 7     <gml:Box gid="box1" srsName="foo">
 8       <gml:coord><gml:X>0</gml:X><gml:Y>0</gml:Y></gml:coord>
 9       <gml:coord><gml:X>4</gml:X><gml:Y>4</gml:Y></gml:coord>
10     </gml:Box>
11   </gml:boundedBy>
12   <gml:featureMember>
13     <Building fid="BuildingA">
14       <Address>Street A</Address>
15       <Shape srsName="foo">
16         <gml:outerBoundaryIs>
17           <gml:LinearRing>
18             <gml:coordinates cs="," ts=" ">-1,2 0,0 0,3 -1,3 -1,2</gml:coordinates>
19           </gml:LinearRing>
20         </gml:outerBoundaryIs>
21       </Shape>
22     </Building>
23   </gml:featureMember>
24 </CityModel>
```

XACML Authorization Decision Request

```
 1  <Request xmlns="urn:oasis:names:tc:xacml:1.0:context"
 2   xmlns:xacml-context="urn:oasis:names:tc:xacml:1.0:context" ... >
 3     <Subject>
 4       <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
 5        DataType="http://www.w3.org/2001/XMLSchema#string">
 6          <AttributeValue>Alice</AttributeValue>
 7       </Attribute>
 8     </Subject>
 9     <Action>
10       <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
11        DataType="http://www.w3.org/2001/XMLSchema#string">
12          <AttributeValue>write</AttributeValue>
13       </Attribute>
14     </Action>
15     ...
```

XACML Authorization Decision Request

```
15    <Resource>
16      <ResourceContent><![CDATA[
17        <CityModel xmlns="http://www.in.tum.de/am" xmlns:am="http://www.in.tum.de/am"
18         xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
19         xsi:schemaLocation="http://http://www.in.tum.de/am CityModel.xsd"
20         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" fid="CityModel">
21        <gml:boundedBy>
22          <gml:Box gid="box1" srsName="foo">
23            <gml:coordinates cs=" " ts="," decimal=".">0 0,4 4</gml:coordinates>
24          </gml:Box>
25        </gml:boundedBy>
26        <gml:featureMember>
27          <Building fid="HouseB">
28            <address>5 Street D</address>
29              <shape srsName="foo">
30                <gml:outerBoundaryIs><gml:LinearRing>
31                  <gml:coordinates cs="," ts=" ">5,4 6,4 6,5 5,5 5,4</gml:coordinates>
32                </gml:LinearRing></gml:outerBoundaryIs>
33              </shape>
34            </Building>
35          </gml:featureMember>
36        </CityModel>]]>
37      </ResourceContent>
38    ...
```

XACML Authorization Decision Request

```
38        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
39         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
40           <AttributeValue>http://myService.com/myOperation</AttributeValue>
41        </Attribute>
42     </Resource>
43  </Request>
```

XACML Authorization Decision Response

Permit, no processing error:

```
1 <Response>
2   <Result>
3     <Decision>Permit</Decision>
4     <Status>
5       <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
6     </Status>
7   </Result>
8 </Response>
```

Indeterminate, syntax error:

```
1 <Response>
2   <Result>
3     <Decision>Indeterminate</Decision>
4     <Status>
5       <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:syntax−error"/>
6     </Status>
7   </Result>
8 </Response>
```

Declaration and Enforcement of Access to a WMS

GetCapabilities: No restriction required

GetMap: Enforcement of Class-based and Spatial restrictions possible

- LAYERS: is interpreted as a class (feature type)
- BBOX: defines area of interest

GetFeatureInfo: Enforcement of Class-based and Spatial restrictions possible. If response is XML/GML encoded, enforcement of Object-based restrictions also possible.

- QUERY_LAYERS: interpreted as a class (feature type)
- X,Y: define -in combination with the original GetMap request parameter the location of the queried feature
- SRS: Used to calculate the feature's "real world" location for the GetFeatureInfo request

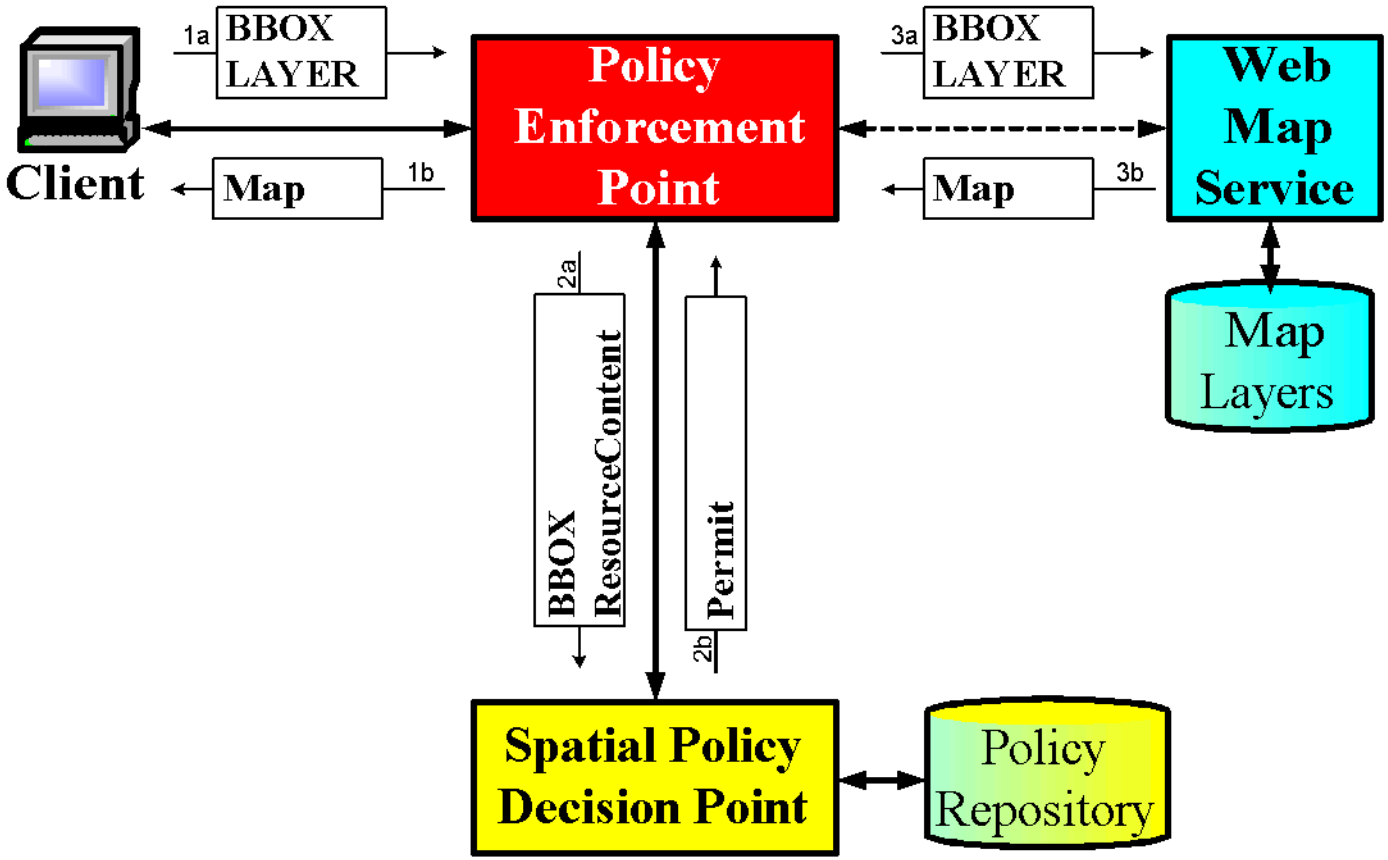
GML application Schema for a WMS ResourceContent

```
 1 <schema targetNamespace="http://www.in.tum.de/am" xmlns:gml="http://www.opengis.net/gml"
 2   xmlns:am="http://www.in.tum.de/am" xmlns="http://www.w3.org/2001/XMLSchema" ... >
 3   <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
 4   <import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlinks.xsd"/>
 5   <!-- Definition of the root element for the resource content-->
 6   <element name="WMSResurceContent" type="am:WMSFeatureCollectionType"
 7     substitutionGroup="gml:_FeatureCollection"/>
 8   <complexType name="WMSFeatureCollectionType">
 9     <complexContent><extension base="gml:AbstractFeatureCollectionType"/></complexContent>
10   </complexType>
11   <!-- Definition of layers, represented by features -->
12   <element name="Building" type="am:WMSFeatureType" substitutionGroup="gml:_Feature"/>
13   <!-- Definition of the common WMS feature type -->
14   <complexType name="WMSFeatureType">
15     <complexContent><restriction base="gml:AbstractFeatureType">
16       <sequence minOccurs="0" maxOccurs="1">
17         <!-- The element PointOfInterest represents the location for requesting additional
18         information, using the GetFeatureInfo interface-->
19           <element name="PointOfInterest" type="gml:PointType"/>
20       </sequence>
21     </restriction></complexContent>
22   </complexType>
23 </schema>
```

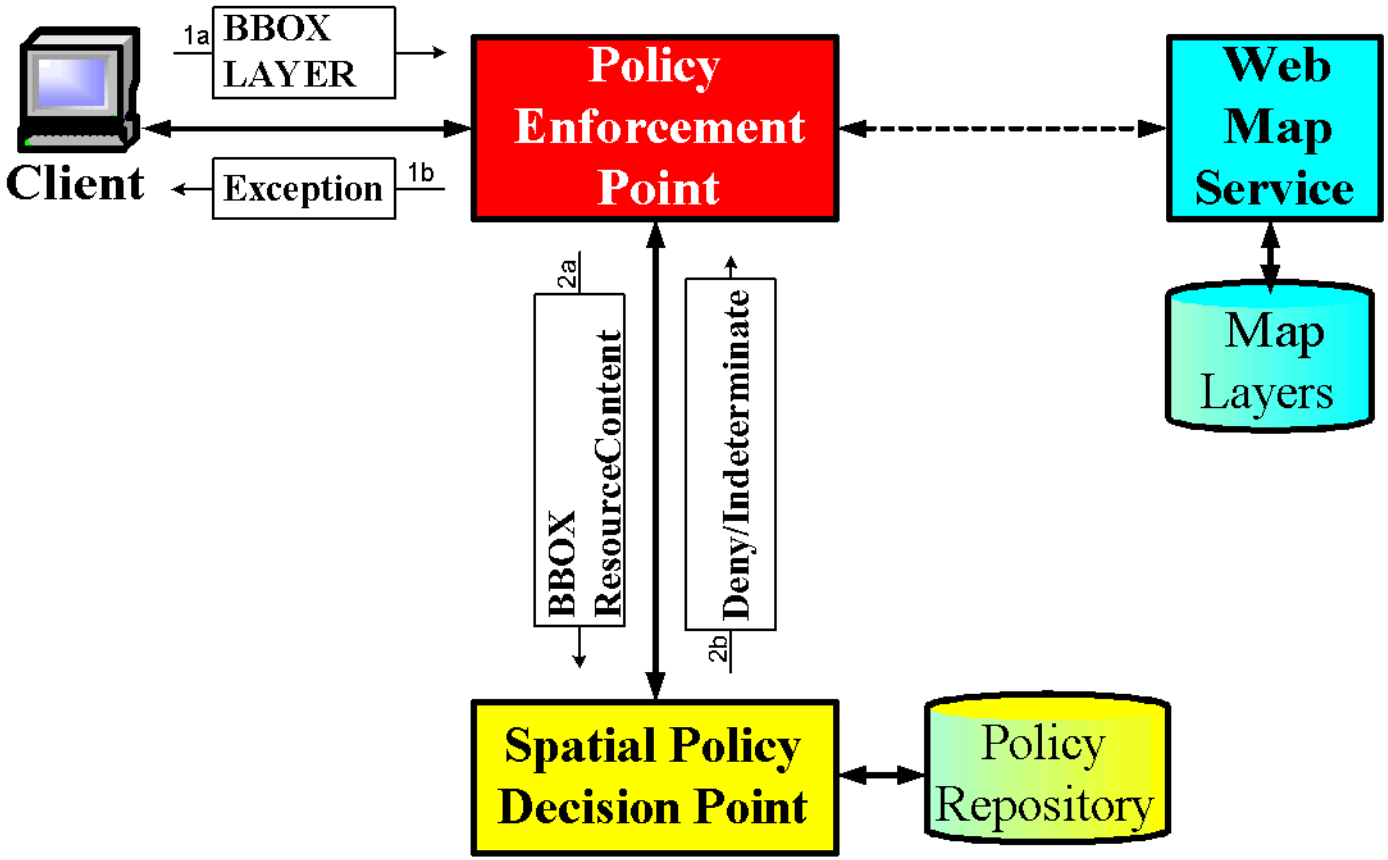
An example WMS ResourceContent

```
 1 <Request xmlns="urn:oasis:names:tc:xacml:1.0:context" ... >
 2   ...
 3   <Resource>
 4     <ResourceContent>
 5       <WMSResurceContent xmlns:xsi="http://www.w3.org/2001/XMLSchema−instance"
 6       xmlns="http://www.in.tum.de/am" xmlns:gml="http://www.opengis.net/gml"
 7       xsi:schemaLocation="http://www.in.tum.de/am WMS.xsd">
 8         <gml:boundedBy>
 9           <gml:Box srsName="EPSG:4326">
10             <gml:coordinates decimal=".">−97.105 24.913,−78.794 36.358</gml:coordinates>
11           </gml:Box>
12         </gml:boundedBy>
13         <gml:featureMember>
14           <Building/>
15         </gml:featureMember>
16       </WMSResurceContent>
17     </ResourceContent>
18     ...
19 </Request>
```

Enforcement Scenario for Web Map Service (Permitted)



Enforcement Scenario for Web Map Service (Denied)



Conclusion

- Introduced authorization supports three types of restrictions
  - Class-based restrictions
  - Object-based restrictions
  - Spatial restrictions
- Based on an extension of an existing standard (XACML from OASIS)
- Based on GML and JTS (Java Topology Suite)
- PEP is deployed as a fassade to the actual service. This does not require to change the implementation of the actual service.
- Interoperability is supported by the standard
- Geodata service can be set up without regard to possible permissions

Online Demonstration

Geospatial Intelligence Viewer

Load context from URL

http://geopep.informatik.tu-muenchen.de/WMS-PEP/World.asp

How does it fit into GeoDRM?

- Introduced access control provides the means for controlling the release of licenses for offline use of a particular content
- Online access in order to download a resource content (e.g. feature collection from WFS)
- Offline use, e.g. in order to modify the content (e.g. delete,create,update featurs in the feature collection)
- Offline use requires license (key) which can be requested from license (key) service

  Who likes to work with the content (subject)

  What is the intended operation (action)

  Which features are to be accessed from the content (resource content)
- License service issues an authorization decision request to the PDP and receives Permit/Deny, which will control the release of a license

What is not covered in this presentation?

- How to carry the authentication information from the client to the PEP. Solution can be based on SAML.
- The creation of correct and error-free policies is difficult using an XML-Editor.
  - Desired: A graphical user interface that allows the creation of permissions based on the high-level model (tupels and geometries).
  - Idea: GUI that requests maps from an existing WMS and allows the ”painting” of the permission geometry and input of the tuples from which the GUI creates the GeoXACML policy structure.
- Maintenace of existing policy repositories and their modificaiton (add/delte/change policies and rules)
- Enforcement for Web Feature Service
- Policy management if service supports requests for multiple CRSs

Outlook: Permit with Constraints allows filtering of allowed features

WMS: Requires image processing in the PEP based on authorization response from Spatial PDP. E.g. Permit + Constraints

1) Authorization decision comprises of Decision + Geometry-Collection of denied areas.
2) PEP creates ”not authorized” image from geometry collection and creates resulting map.

WFS: Requires modification of service parameters (e.g. Filter) or modification of GML service response

1) PEP creates authorization decision request based on service paramter (pre-processing approach)
2) PEP creates authorization decision request for each feature of the WFS response (post-processing approach)
3) PEP creates authorization decision request based on service parameter and response (hybrid approach)

Authentication Information

- Demonstration does not ”carry” authentication information
- SOAP communication between client and PEP
  - Enables the transport auf authentication information
  - Can be added to the SOAP-header
  - WMS or WFS request/response can be added to the SOAP body
- Authentication in a distributed environment
  - Requires an Authentication Service
  - Authentication Service provides login capabilities and provides authentication information
  - The authorization is based on this authentication information, encoded in e.g. SAML
- If PEP provides WMS interfaces, a SOAP/HTTP-POST interface for WMS is required
  - Maybe WMS 3.0 that supports SOAP/HTTP-POST interfaces?

Further Research Topics

1) Policy-editor, visualization of restrictions, Validation of a policy-repository
   - Policy-editor including WMS-client to retreave maps
   - Allow drawing of permission geometries
   - Support the input of permissions in a high-level syntax (e.g. as introduced)
   - Supports the saving of the geometries using GML
   - Support for creation of GeoXACML rules and possibly policies and policy sets
2) Permit with constraints (result features = requested features [allowed features)
   - Allow the filtering of allowd features
   - Pre-processing allows modification of request parameters. Is it always possible?
   - Post-processing allows the modification of the result. Is it always feasible?
   - Hybrid approach: Combine pre- and post-processing advantages
   - Develop for WMS and WFS

Further Research Topics

3) Permission management for aggregate services
   - Aggregate service hides used services. Upon which identity are the permissions linked? Possibly multiple identities: The access is permitted if the identities are aggregate-service and subject X
   - Can a user fulfill all tasks with a given aggregate-service? Does he have enough access rights?
   - Which access rights does it require to fullfill a particular task, using an aggregate service?

The final slide

Thank you for your attention

Questions, please...