DRM policies for Web Map Service[1]

Alban Gabillon
Université de la Polynésie française
BP 6570 Faa'a Aéroport
+689 80 38 80
alban.gabillon@upf.pf

Patrick Capolsini
Université de la Polynésie française
BP 6570 Faa'a Aéroport
+689 80 38 83
patrick.capolsini@upf.pf

ABSTRACT

Open Digital Rights Language (ODRL) is an extensible language for specifying Rights Policy in the context of Digital Rights Management (DRM) applications. The OpenGIS® Web Map Service (WMS) supports the creation and display of maps. In this paper we extend ODRL to accommodate licensing for geographic data created by WMS.

Categories and Subject Descriptors

K.6.m [**Management of computing and information System**]: Miscellaneous – *Security*

General Terms

Security

Keywords

Rights Expression Language, Digital Rights Management, Web Map Service, Geospatial Data

1. INTRODUCTION

For more than thirty years, many research studies have been dealing with access control. Current models (e.g. see [1-3]) consider dynamic access rules which can depend on contextual conditions. However, those access control techniques cannot provide us with a satisfying solution to the problem of controlling duplication and dissemination of data.

Recent research studies have started to produce new security models which are called usage control models [4]. A usage control model is more general than an access control model. It aims at controlling not only the access but also the use of data.

The main application of usage control is DRM (Digital Rights Management) and "the fundamental goal of DRM is to send data with usage policy into a remote, possibly hostile, computing environment and know that the policy will be respected" [5].

ACM SPRINGL '08 , November 4, 2008. Irvine, CA, USA (c) 2008 ACM ISBN 978-1-60558-324-2/08/11...$5.00.

From the above definition, we can see that there are two aspects to DRM. The first aspect is content management. It refers to the security mechanisms which enforce the security policy. These security mechanisms are implemented by means of cryptographic techniques. The second aspect is policy management. It refers to the Right Expression Language (REL) which is used to write the security policy. There are several existing RELs. The two most prominent ones are ODRL [6] and XrML [7]. Both of them are based on XML. XrML is the basis for the REL developed for Moving Picture Experts Group (MPEG) [8]. ODRL was adopted by The Open Mobile Alliance and used in their DRM specifications.

Historically, the primary objective of DRM was to control the distribution of consumer playable media and, in a more general way, the distribution of copyrighted digital content. However, DRM is gradually finding new areas of application. More and more, DRM technologies are used to protect trade secrets which are different from copyrighted material, and for whom the primary issue is industrial or corporate espionage or inadvertent release. Used in this context, DRM is referred to as Enterprise Digital Rights Management (E-DRM).

ORDL and XrML have been following a similar evolution. Whereas, XrML 1.0 was designed to write policies protecting songs and videos, XrML 2.0 is not specific to any medium or type of resource. ODRL which was designed from the beginning as an extensible language is being enriched with new profiles. It has created a profile that supports Creative Commons licenses and is working on a profile for geospatial data and a profile for Dublin Core Metadata Initiative (DCMI) metadata.

Regarding the geospatial domain, the OGC (Open Geospatial Consortium) Geospatial DRM Reference Model [9] highlights the fact that existing REL cannot be used for licensing of geographic information unless they are extended. The GeoDRM RM states that the rights model must accommodate licensing for geographic data which are dynamically created by using OpenGIS web services [10] or delivered as a static product (on a CD-ROM for instance). The GeoDRM RM states also that licensing of geographic information requires support to declare and enforce rights as they are based on the geometry of the digital content.

In this paper, our aim is to extend ODRL to accommodate licensing for geographic data created by OpenGIS Web Map Services (WMS) [11]. Our work can serve as a basis for a

[1] This work was conducted as part of the ANR funded project under reference ANR-SESUR-2007-FLUOR

complete ODRL WMS-profile. Writing such a profile could be one of the duty of the ODRL Geospatial Profile Working Group.

Note, that in our work we shall not consider the different business relationships for digital geographic data which are defined by the GeoDRM RM. First, we hardly see to which extent are these relationships geo-specific. Second, it is obvious that these business relationships were devised for copyrighted (geospatial) data. However, geospatial information is not necessarily copyrighted. It can be sensitive information in an E-DRM system. It can be a military secret in a network-centric warfare system…etc

The remainder of this paper is organized as follows: in Section 2 we review the specificity of geospatial data, geospatial services and WMS. In Section 3, we briefly review ODRL and recall the procedure for extending it. In Section 4, we define an ODRL WMS-profile. In Section 5, we show one example of license for WMS. In Section 6, we sketch the architecture of a DRM platform for controlling access and use of data created by a WMS. In Section 7, we conclude this paper.

2. GEOSPATIAL DATA & WMS

During the last decade, the amount and complexity of geographic data have increased exponentially. Geographic Information Systems (GIS) software was developed and this technology is now fully operational and worldwide used. Specialized commercial software like Pitney Bowes MapInfo® [*http://www.mapinfo.com/*] or ESRI ArcGIS® [*http://www.esri.com/*] are widely spread and OpenSource freeware solutions like QuantumGIS [*http://www.qgis.org/*] or GRASS [*http://grass.itc.it*] are becoming more and more relevant to achieve geographic data management, retrieval, storage, treatment and analysis.

GIS software generally deal with two main types of spatial data, namely raster and vector data. Aerial photography and satellite imagery are common raster data. Vector data can be of different types (points, lines, poly-lines, polygons,…etc) and can represent a large variety of features themes such as points of interest, roads, buildings, borders of states, elevation contours, restricted access areas and so on. Each geographic data is handled as a layer and layers can be overlaid to get the final desired representation of the dataset. All GIS layers (either raster or vector) can be stored as files or as records in a spatially enabled object-relational database.

The basic features of GIS software include : zoom in, zoom out, pan, x/y localization, choice of predefined scales, layers overlay, hierarchy and re-order of layers, layers transparency, measurements (distances and surfaces), outlining and annotating tools, … Advanced features include complex data views and queries, projections, data analyzes, geographic modeling, data sharing, statistics and so on.

While professional GIS tools were growing up, also grew the Internet and the needs for easy visualization and exchange of geographic data. Easy visualization is being achieved by the mean of easy-to-use dedicated or web-based interfaces. Nowadays, 3D GeoSpatial viewers like Google earth®, Skyline Globe® or MicroSoft Virtual Earth® are universally used by both novice and expert users. The need for geographic data exchanges led to the creation of the Open Geospatial Consortium (OGC). OGC is an international industry consortium of over 350 companies, agencies and universities participating in a process to develop, support and promote interoperable solutions to "geo-enable" the Web. The OGC publishes the OpenGIS® Specifications concerning Geographic Databases and data exchange protocols. Two of the most popular OpenGIS® implementation specifications are the Web Features Service (WFS) [12] and the Web Map Service (WMS). Hereafter, we very briefly present WFS and we describe WMS in details.

The OpenGIS® Web Feature Service allows a client to simply retrieve (WFS) or retrieve and update (WFS-Transactional) geospatial data encoded in the OpenGIS® Geography Markup Language (GML) [13]. WFS provides the opportunity to handle and manage real and accurate GeoData from different available WFS servers. The response to a WFS request is a GML file containing the feature information behind a map image.

The OpenGIS® Web Map Service supports the creation and display of registered and superimposed map-like views of information coming from multiple heterogeneous sources. Unlike WFS, WMS delivers data in a *pictorial format* such as PNG, GIF or JPEG, or occasionally as XML-expressed Scalable Vector Graphics (SVG). Three requests are defined: the first one (GetCapabilities) returns service-level metadata, the second one (GetMap) returns a map according to well-defined geographic and dimensional parameters and the optional third one (GetFeatureInfo) returns information about particular features shown on a map. Most WMS servers may be requested via an URL issued from a standard web-browser or any WMS-enabled software. As far as a WMS client requests an image format that supports transparency (GIF or PNG), it can access and accurately overlay maps from one or more available WMS servers allowing the creation of a network of distributed map servers from which customized maps can be built. Furthermore, a WMS server may act as a client for another WMS server leading to the concept of cascading map servers. A basic GetCapabilities request sent to an "University of Minesota (UMN) MapServer" looks like :

http://webgis.upf.pf/cgi-bin/mapserv?MAP=/path_to_map&SERVICE=WMS&REQUEST=GetCapabilities.

Some optional parameters can be added. The response is an XML file containing service metadata formatted according to the appropriate OpenGis® XML Schema. In addition to some general information, the service provides critical information concerning the layers and styles of the geographic information. Although layers may be hierarchically nested as desired, conceptually, each layer is a distinct entity and is individually described by a `Layer` element. Besides some basic metadata elements, any layer presents three fundamental elements, namely: `EX_GeographicBoundingBox`, `CRS` (Coordinate Reference System) and `BoundingBox`. These elements aim at defining the geographical extent of the layer (a rectangle generally defined by its lower-left and upper-right corners). The `EX_GeographicBoundingBox` is always returned by the WMS server and gives the extent of the layer using the standard spatial coordinate system known by any GPS-user as the Latitude / Longitude system. This coordinate system is called World Geodetic System 1984 (WGS84) by geographers. This so-called "un-projected" coordinate system is certainly very useful to locate a geographical zone on the globe but specialists often prefer working in "projected" coordinate systems like the Uniform Transverse Mercator (UTM). If the layer can be served in one or

more of these "projected" coordinate systems then, for each supported system, a pair of `CRS/BoundingBox` elements is provided.

A basic GetMap request sent to an "UMN MapServer" [14] server looks like: *http://webgis.upf.pf/cgi-bin/mapserv?MAP=/path_to_map&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&LAYERS=layer1,layer2&CRS=cr_identifier&BBOX=minx,miny,maxx,maxy&FORMAT=image/png&WIDTH=640&HEIGHT=480*

The first four parameters are mandatory. A few other optional parameters of minor importance exist but are not described here. The response to a valid WMS GetMap request is an image file in the specified `format` (MIME type) having the dimension `width` by `height` pixels. The output image renders the requested layers by drawing first the leftmost in the list, then the next one over it, and so on. The `crs` and `bbox` parameters (see definition here above) define the geographic extent of the desired map. Conjunction of the requested image size (`width` and `height` parameters of the WMS request) with the *in situ* geographical extent (`bbox` parameter) leads to the definition of the *zoom factor*. For instance, a 640 by 480 pixels image representing a geographic area of 4 km by 3 km has a scale factor of 0.00625 km/px. Applying a zoom factor of 2 on this image leads to keeping the same output size of 640 by 480 (same `width` and `height` values) but changing the actual `bbox` parameter to get a geographical extent of 2 km by 1.5 km leading to a scale factor of 0.003125 km/px.

3. ODRL

ODRL v1.1 is an open, XML-based, extensible language for specifying usage control policies. ODRL utilises two XML schemas. One schema defines the Expression Language elements and the relationships between these elements, the other defines the Data Dictionary elements. The Data Dictionary elements are used to instantiate the Expression Language elements when writing a rights policy.

ODRL can express offers and agreements involving the three following basic Expression Language elements :

- `asset`
- `permission`
- `party`

`asset` is used to declare objects to which the rights policy applies. `permission` is used to define the rights policy. Permissions are rules regulating access and use of objects. `party` is used to declare subjects involved in the offer and/or agreement.

The following XML document is an example of a very simple agreement. Cyrus is granted the permission to watch the DVD "Winnie the Pooh".

```
<o-ex:rights
     xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
     xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
   <o-ex:agreement>
      <o-ex:asset>
            <o-ex:context>
         <o-dd:uid>dvd:Winnie the Pooh</o-dd:uid>
            </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:play />
      </o-ex:permission>
      <o-ex:party>
         <o-ex:context>
            <o-dd:uid>Cyrus</o-dd:uid>
         </o-ex:context>
      </o-ex:party>
   </o-ex:agreement>
</o-ex:rights>
```

The above document is valid with regard to the two ODRL schemas. Elements prefixed with `o-ex` are Expression Language elements whereas elements prefixed with `o-dd` are Data Dictionary elements. `context` element is generally used to identify real life entities like assets or parties but can serve to other purposes as well. What is not shown in this very simple example is that permissions can contain constraints, requirements and conditions. Constraints define limits to permissions. Requirements are obligations which should be fulfilled before exercising permissions. Conditions specify exceptions that, if occur, expire permissions.

ODRL can easily be extended by writing Custom Dictionaries. Let us consider the previous example and let us assume that Cyrus' father wants Cyrus to learn English. DVD "Winnie the Pooh" can be watched in several languages including French, English … etc. We can write a Custom Dictionary giving Cyrus' father the possibility to declare that Cyrus is permitted to watch the DVD "Winnie the Pooh" only in English:

```
<xsd:schema targetNamespace=http://www.upf.
pf/~gabillon/dvd-DD
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:o-ex=http://odrl.net/1.1/ODRL-EX"
   xmlns:dvd="http://www.upf.pf/~gabillon/dvd-DD"
   elementFormDefault="qualified"
   attributeFormDefault="qualified">
   <xsd:import
        namespace="http://odrl.net/1.1/ODRL-EX"
        schemaLocation="http://odrl.net/1.1.ODRL-
        EX-11.xsd"/>
   <xsd:element
      name="audio-track"
      type="o-ex:constraintType"
      substitutionGroup="o-ex:constraintElement"/>
</xsd:schema>
```

Basically, in the above schema, we declare a new namespace and we extend the ODRL dictionary with a new constraint `audio-track`. We can now express that Cyrus is permitted to watch the DVD "Winnie the Pooh" only in English:

```
<o-ex:rights
xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
xmlns:dvd="http://www.upf.pf/~gabillon/dvd-DD">
...
        <o-dd:play>
           <o-ex:constraint>
              <dvd:audio-track>
                 <o-ex:context>
                    <o-dd:uid>English</o-dd:uid>
                 </o-ex:context>
              </dvd:audio-track>
           </o-ex:constraint>
        </o-dd:play>
...
</o-ex:rights>
```

The constraint restricts the permission to play the DVD to the English version. This simple example has shown us how we can easily extend ODRL to cope with different application contexts. Other examples showing the expressive power of ODRL can be found in [15].

4. WMS SECURITY

4.1 Principles

Dealing with access and use of data created by Web Map Services poses the problem of identifying the asset. As noticed in [16], "the online browsing of map information available from a Web Map Service creates an infinite number of different contents. Here context information of the content such as the layer, the resolution, the area of interest, the styles etc. can be part of a unique identification". Following this suggestion, writing ODRL licenses for contents created by WMS could be simply done with core ODRL by identifying each content with its corresponding WMS query (which is a URL). However, it would be unrealistic to enumerate in a license each and every permitted WMS query.

The main idea behind our WMS-profile is to extend ODRL with *one context element* allowing us to identify a set of contents created by a range of WMS queries. This context element is of complex type and allows us to specify ranges of values for the main WMS parameters.

Design of our WMS-profile was driven by the need to express rights policies regulating access and zoom-in operations to some coordinates and/or layers. We consider only the basic WMS i.e. the WMS `GetMap` and `GetCapabilitites` requests and we focus on WMS parameters which are the most relevant to consider for these security needs, namely the `layers` and `bbox` parameters.

In this paper, we do not define a formal model for geographic data. Our aim is only to extend ODRL in order to be able to write rights policies protecting contents created by WMS. Let us however mention that we could formally express such rights policies by using the model defined in [17].

In the remainder of this section, elements of our Custom Dictionary should not be confused with WMS query parameters. Elements of our Dictionary are prefixed with the namespace `wms`.

Fig 1. : ODRL WMS Context Model

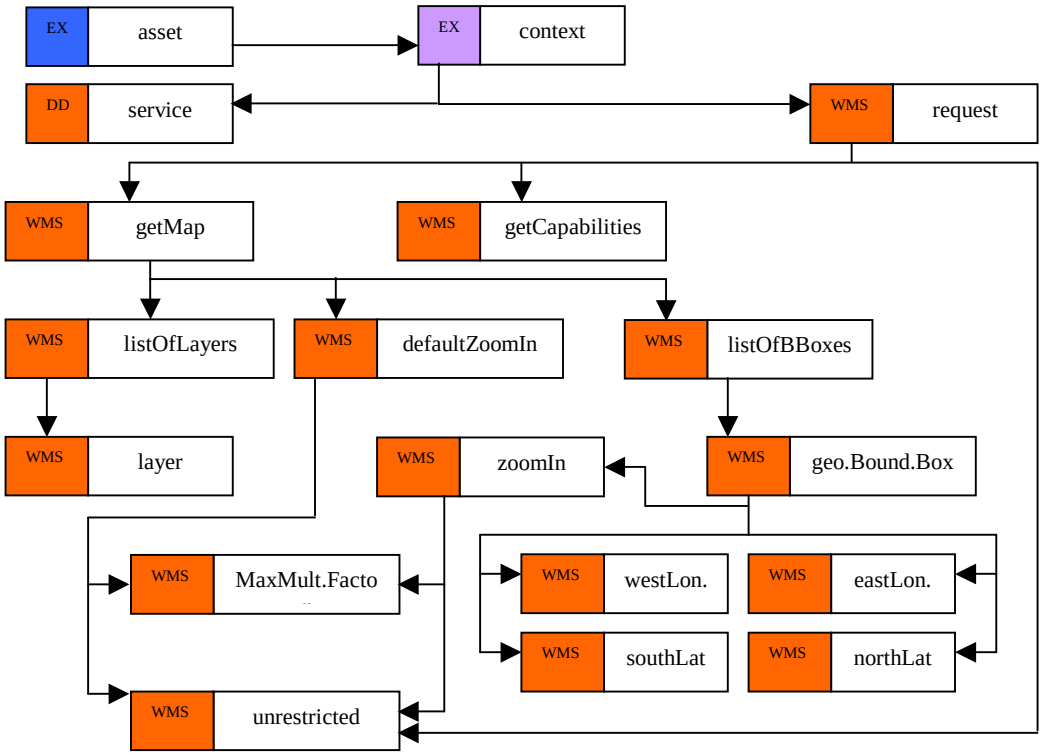


Figure 1 shows our WMS context model. For identifying a set of contents created by a range of WMS queries, we use the core ODRL Data Dictionary element `o-dd:service` and we define our Custom Dictionary context element `wms:request`. Element `o-dd:service` contains an URL to the WMS, whereas element `wms:request`, which is of a complex type, contains sub-elements for specifying ranges of permitted values for the main WMS parameters. The `wms:request` element either contains the `wms:getMap` element, the `wms:getCapabilities` element, or the `wms:unrestricted` element. At this stage, the `wms:unrestricted` element would be used to express that WMS parameters are not restricted at all. `wms:getCapabilities` element would be used to identify the asset created by the WMS `GetCapabilities` request. Element `wms:getMap` is of a complex type and contains other elements for specifying permitted ranges of values for the main WMS `GetMap` parameters, namely `layers` and `bbox` parameters. This `wms:getMap` element contains either the `wms:unrestricted` element or the triple consisting of `wms:listOfLayers`, `wms:defaultZoomIn` and `wms:listOfBBoxes` elements. Element `wms:listOfLayers` is used to specify the list of permitted

layers which can be included in the WMS `layers` parameter. Important additional details on the `wms:listOfLayers` are given in Section 4.2.

In the WMS protocol, zoom-in and zoom-out operations are performed by means of the WMS `bbox`, `height` and `width` parameters. In the remainder of this paper, we make the simplifying assumption that the `height` and `width` parameters remain constant. Consequently,

- Zoom-in and zoom-out operations can be performed by only changing the value of the `bbox` parameter.
- The user querying the WMS is always provided with a map of the same size.

Combination of the `wms:defaultZoomIn` element and `wms:listOfBBoxes` element is used to specify the permitted range of values for the WMS `bbox` parameter. Element `wms:defaultZoomIn` defines a default Maximum Zoom-in Factor (MZF), whereas `wms:listOfBBoxes` defines different areas with a specific MZF.

Let $(minx^{map}, miny^{map}, maxx^{map}, maxy^{map})$ be the value of the `EX_GeographicBoundingBox` of the map, i.e. it is the value of the `EX_GeographicBoundingBox` of the largest layer. Recall that $minx^{map}, miny^{map}, maxx^{map}, maxy^{map}$ are expressed in terms of longitudes and latitudes (see Section 2).

Let q be a WMS `GetMap` query.

Let $(minx, miny, maxx, maxy)$ be the value of the `bbox` parameter of query q. $minx, miny, maxx, maxy$ are also expressed in terms of longitudes and latitudes.

Let z be the lowest MZF of *all the areas overlapped* by the bounding box defined in the `bbox` parameter.

`bbox` parameter is valid if,

$$\max\left(\begin{array}{l}\left(maxx^{map} - minx^{map}\right)/\left(maxx - minx\right),\\ \left(maxy^{map} - miny^{map}\right)/\left(maxy - miny\right)\end{array}\right) \le z \quad (1)$$

Section 4.2 gives more information on the `wms:listOfBBoxes` element and Section 5 illustrates the utility of the different MZFs.

4.2 Custom Dictionary

In this section, we give a step by step description of our Custom Dictionary `wms.xsd`. It starts with the declaration of the `wms` namespace:

```
<xsd:schema
targetNamespace="http://www.opengeospatial.org/standards/wms"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:o-ex=http://odrl.net/1.1/ODRL-EX
xmlns:wms="http://www.opengeospatial.org/standards/wms"
 elementFormDefault="qualified"
 attributeFormDefault="qualified">
  <xsd:import
    namespace="http://odrl.net/1.1/ODRL-EX"
    schemaLocation="http://odrl.net/1.1/ODRL-EX-11.xsd" />
```

Below is the declaration of two elements and one attribute. The first element is the context element **`wms:request`**. The other is **`wms:unrestricted`** of type emptyType. Utility of the **`wms:setFunction`** attribute will be explained later.

```
  <xsd:element
      name="request"
      type="wms:requestType"
      substitutionGroup="o-ex:contextElement"/>
  <xsd:element
      name="unrestricted"
      type="wms:emptyType"/>
  <xsd:attribute
      name="setFunction"
      type="wms:setFunctionType"/>
  <xsd:complexType    name="emptyType"/>
```

Value of the previous attribute can be one of the followings:

```
<xsd:simpleType            name="setFunctionType">
    <xsd:restriction       base="xsd:string">
      <xsd:enumeration     value="include" />
      <xsd:enumeration     value="equalTo" />
      <xsd:enumeration     value="subsetOf" />
    </xsd:restriction>
</xsd:simpleType>
```

A request is either unrestricted, equal to `getCapabilities`, or equal to `getMap`.

```
  <xsd:complexType    name="requestType">
    <xsd:choice>
      <xsd:element    ref="wms:unrestricted"/>
      <xsd:element    name="getMap"
                      type="wms:getMapType"/>
      <xsd:element    name="getCapabilities"
                      type="wms:emptyType"/>
    </xsd:choice>
  </xsd:complexType>
```

A getMap request is either unrestricted or restricted according to the **`wms:listOfLayers`**, **`wms:defaultZoomIn`** and **`wms:listOfBBoxes`** elements.

```
  <xsd:complexType     name="getMapType">
    <xsd:choice>
      <xsd:element     ref="wms:unrestricted"/>
      <xsd:sequence>
        <xsd:element
            name="listOfLayers"
            type="wms:listOfLayersType"/>
        <xsd:element
            name="defaultZoomIn"
            type="wms:zoomInType"/>
        <xsd:element
            name="listOfBBoxes"
            type="wms:listOfBBoxesType"
            minOccurs="0"/>
      </xsd:sequence>
    </xsd:choice>
  </xsd:complexType>
```

Element **`wms:listOfLayers`** consists of an unbounded list of layers and a required attribute **`wms:setFunction`**. If the value of this attribute is **`subsetOf`** then it means that querying any subset of the specified list of layers is permitted. If the value is **`equalTo`** then it means that only querying the exact specified list of layers is permitted. If the value is **`include`** then it means that querying any set of layers which includes the specified list of layers is permitted. Example given in Section 5 illustrates the importance of this attribute.

```
  <xsd:complexType name="listOfLayersType">
    <xsd:sequence>
      <xsd:element    name="layer"
                      type="xsd:string"
                      minOccurs="1"
                      maxOccurs="unbounded"/>
```

```
    </xsd:sequence>
    <xsd:attribute
              ref="wms:setFunction"
              use="required"/>
</xsd:complexType>
```

A zoom-in operation is either unrestricted or restricted to a maximum multiplicative factor.

```
<xsd:complexType    name="zoomInType">
   <xsd:choice>
      <xsd:element   ref="wms:unrestricted"/>
      <xsd:element
              name="maxMultiplicativeFactor"
              type="xsd:nonNegativeInteger"/>
   </xsd:choice>
</xsd:complexType>
```

Areas where the default MZF does not apply, are defined by means of bounding boxes.

```
<xsd:complexType    name="listOfBBoxesType">
   <xsd:sequence>
       <xsd:element
          name="geographicBoundingBox"
          type="wms:geographicBoundingBoxType"
          minOccurs="1"
          maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Element **`wms:geographicBoundingBox`** defines one bounding box where the default MZF does not apply. Required sub-element **`wms:zoomIn`** specifies the MZF for this bounding box.

```
<xsd:complexType
              name="geographicBoundingBoxType">
   <xsd:sequence>
      <xsd:element
           name="westLongitude"
           type="xsd:decimal"/>
      <xsd:element
           name="southLatitude"
           type="xsd:decimal"/>
      <xsd:element
           name="eastLongitude"
           type="xsd:decimal"/>
      <xsd:element
           name="northLatitude"
           type="xsd:decimal"/>
      <xsd:element
          name="zoomIn"
          type="wms:zoomInType"/>
   </xsd:sequence>
</xsd:complexType>
```

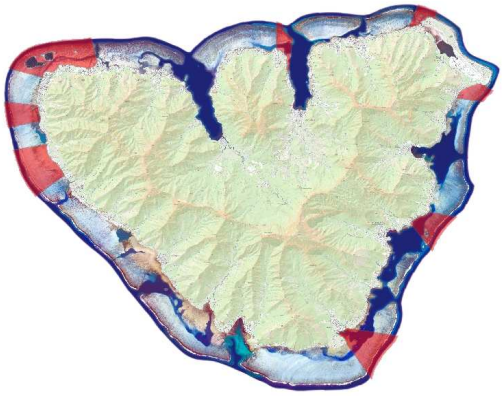
In the next section, we give one example of a rights policy for WMS.

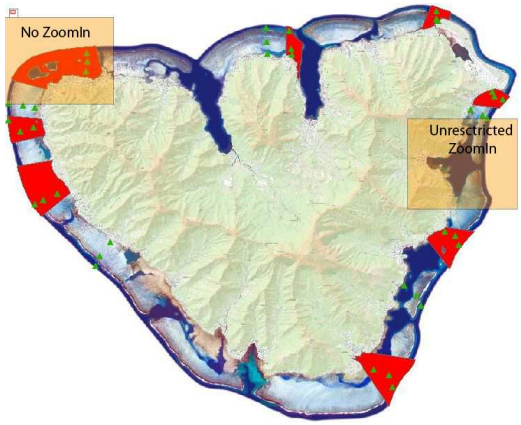
5. EXAMPLE OF A RIGHTS POLICY

ODRL model can express offers and agreements. Offers are proposals for specific rights over their assets. Agreements are when parties enter into contracts. In this section, we illustrate our WMS-profile with one example of an offer. The policy expressed in the offer protects access to a WMS delivering maps of Moorea Island (French Polynesia) consisting of one, two or three layers. The first layer `composite` is a raster image of Moorea. The second layer `amp` consists of a set of vector data showing protected marine areas in Moorea. The third layer `criobe` consists of a set of vector data showing the location of sensitive zones for sampling and monitoring reef resources. Basically, we would like to express that whenever the WMS query includes the `criobe` layer, zoom-in should be restricted to a default MZF. This is because, the exact position of zones for sampling and monitoring reef resources should not be disclosed. Moreover, in one given area, zoom-in should be completely forbidden whereas in one other given area zoom-in should be unrestricted. More precisely, the rights policy included in the offer should express the following permissions:

- It is permitted to display and print the result of a `GetCabapilities` query
- It is permitted to display and print the result of any `GetMap` query whose `layers` parameter is equal to any subset of `{composite,amp}`. For such queries, zoom-in should be everywhere unrestricted. Figure 2(a) shows Moorea Island with the two layers `{composite,amp}`. Protected Marine areas (`amp` layer) are in red.
- It is permitted to display the result of any `GetMap` query whose `layers` parameter includes the layer `criobe`. However, for such queries, the default MZF should be equal to 5. Moreover, there is one area more sensitive than the others, with a specific MZF equal to 1 (i.e. zoom-in is forbidden). On the contrary, there is one area with an unrestricted MZF (it corresponds to the harbour zone for the boats coming from Tahiti). Finally, it is permitted to print the result of not more than 10 of such queries. Figure 2(b) shows Moorea Island with the three layers `{composite,amp,criobe}` and the two bounding boxes where the MZF is different from the default MMZF. Zones for sampling and monitoring reef resources are the small green triangles.

Fig 2(a) Fig 2(b). : Moorea Island with Protected Marine Areas (red) and Zones for Sampling and Monitoring Reef Resources (green triangles)





Our policy starts with the declaration of assets. The first asset identifies the set of contents resulting from queries whose `layers` parameter is equal to any subset of `{composite,amp}`.

```
<o-ex:rights
   xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
   xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
   xmlns:wms="http://www.opengeospatial.org/stan
              dards/wms">
<o-ex:offer>
   <o-ex:asset o-ex:id="ASSET1">
        <o-ex:context>
        <o-dd:uid>WMS_ASSET1</o-dd:uid>
        <wms:request>
           <wms:getMap>
                <wms:listOfLayers
                       wms:setFunction="subsetOf"
                >

                <wms:layer>composite</wms:layer>
                   <wms:layer>amp</wms:layer>
                </wms:listOfLayers>
                <wms:defaultZoomIn>
                <wms:unrestricted/>
        </wms:defaultZoomIn>
           </wms:getMap>
        </wms:request>
        <o-dd:service>
http://webgis.upf.pf/cgi-
bin/mapserv?map=/home/webgis/cartoweb3/projects/M
oorea/server_conf/Moorea/Moorea_limited.map
        </o-dd:service>
     </o-ex:context>
   </o-ex:asset>
```

The second asset identifies the set of contents resulting from queries whose `layers` parameter includes the layer `criobe`. There are limitations on the `bbox` parameter resulting from the default MZF equal to 5 and one area with a specific MZF equal to 1. In one other area, the MZF remains unrestricted.

```
<o-ex:asset o-ex:id="ASSET2">
  <o-ex:context>
        <o-dd:uid>WMS_ASSET2</o-dd:uid>
        <wms:request>
          <wms:getMap>
            <wms:listOfLayers
               wms:setFunction="include">
               <wms:layer>criobe</wms:layer>
            </wms:listOfLayers>
            <wms:defaultZoomIn>
               <wms:maxMultiplicativeFactor>
                  5
               </wms:maxMultiplicativeFactor>
            </wms:defaultZoomIn>
            <wms:listOfBBoxes>
               <wms:geographicBoundingBox>
                 <wms:westLongitude>
                    -149.928742
                 </wms:westLongitude>
                 <wms:southLatitude>
                    -17.500643
               </wms:southLatitude>
               <wms:eastLongitude>
                    -149.890238
               </wms:eastLongitude>
               <wms:northLatitude>
                    -17.471618
               </wms:northLatitude>
               <wms:zoomIn>
                   <wms:maxMultiplicativeFactor>
                      1
                   </wms:maxMultiplicativeFactor>
               </wms:zoomIn>
               </wms:geographicBoundingBox>
               <wms:geographicBoundingBox>
                 <wms:westLongitude>
                    -149.788651
               </wms:westLongitude>
               <wms:southLatitude>
                    -17.537058
               </wms:southLatitude>
               <wms:eastLongitude>
                    -149.749546
               </wms:eastLongitude>
               <wms:northLatitude>
                    -17.507470
               <</wms:northLatitude>
               <wms:zoomIn>
                       <wms:unrestricted/>
               </wms:zoomIn>
               </wms:geographicBoundingBox>
            </wms:recenter>
          </wms:getMap>
        </wms:request>
        <o-dd:service>
http://webgis.upf.pf/cgi-
bin/mapserv?map=/home/webgis/cartoweb3/projects/M
oorea/server_conf/Moorea/Moorea_limited.map
        </o-dd:service>
  </o-ex:context>
</o-ex:asset>
```

In order to illustrate the role of the different MZFs, let us consider a WMS `GetMap` query q with a `layer` parameter including the layer `criobe`:

- If the bbox parameter of query q defines a bounding box overlapping the area where zoom-in is forbidden (MZF = 1) then z is equal to 1. Equation 1 (see Section 4.1) cannot be satisfied. The bbox parameter is invalid and query q should be rejected.
- If the bbox parameter of query q defines a bounding box overlapping both the area where zoom-in is unrestricted and the area around where zoom-in is restricted to the default MZF of 5, then z is equal to 5. bbox is valid only if equation 1 holds.
- If the bbox parameter of query q defines a bounding box included in the area where zoom-in is unrestricted then bbox is valid.

The last asset identifies the result of a GetCapabilities request.

```
<o-ex:asset o-ex:id="ASSET3">
   <o-ex:context>
     <o-dd:uid>WMS_ASSET3</o-dd:uid>
       <wms:request>
              <wms:getCapabilities/>
       </wms:request>
       <o-dd:service>
http://webgis.upf.pf/cgi-
bin/mapserv?map=/home/webgis/cartoweb3/projects/M
oorea/server_conf/Moorea/Moorea_limited.map
       </o-dd:service>
   </o-ex:context>
</o-ex:asset>
```

Finally, the *usage policy* expresses the usage rights over the assets

```
<o-ex:permission>
   <o-ex:asset o-ex:idref="ASSET1"/>
   <o-ex:asset                    o-
ex:idref="ASSET3"/>
     <o-dd:display />
     <o-dd:print />
</o-ex:permission>
<o-ex:permission>
   <o-ex:asset o-ex:idref="ASSET2"/>
     <o-dd:display />
     <o-dd:print>
       <o-ex:constraint>
         <o-dd:count>10</o-dd:count>
       </o-ex:constraint>
     </o-dd:print>
</o-ex:permission>
</o-ex:offer>
</o-ex:rights>
```

One should note that there is no intersection between the range of queries defining ASSET1 and the range of queries defining ASSET2. In fact, this should be a general rights policy design principle. In case a license includes more than one WMS assets, there should be no intersection between the various ranges of queries defining the assets. Following this principle leads to a better readability of the rights policy and prevents conflicts between rights.

The schema presented in Section 4 and the above license can be both accessed to at the following URLs:

- Schema: *http://pages.upf.pf/Alban.Gabillon/odrl/wms.xsd*
- License: *http://pages.upf.pf/Alban.Gabillon/odrl/policy.xml*

The WMS delivering maps of Moorea Island can be accessed to through the following URL:

- Moorea: *http://pages.upf.pf/Patrick.Capolsini/rech/WMS_maps.htm*

6. DRM ARCHITECTURE

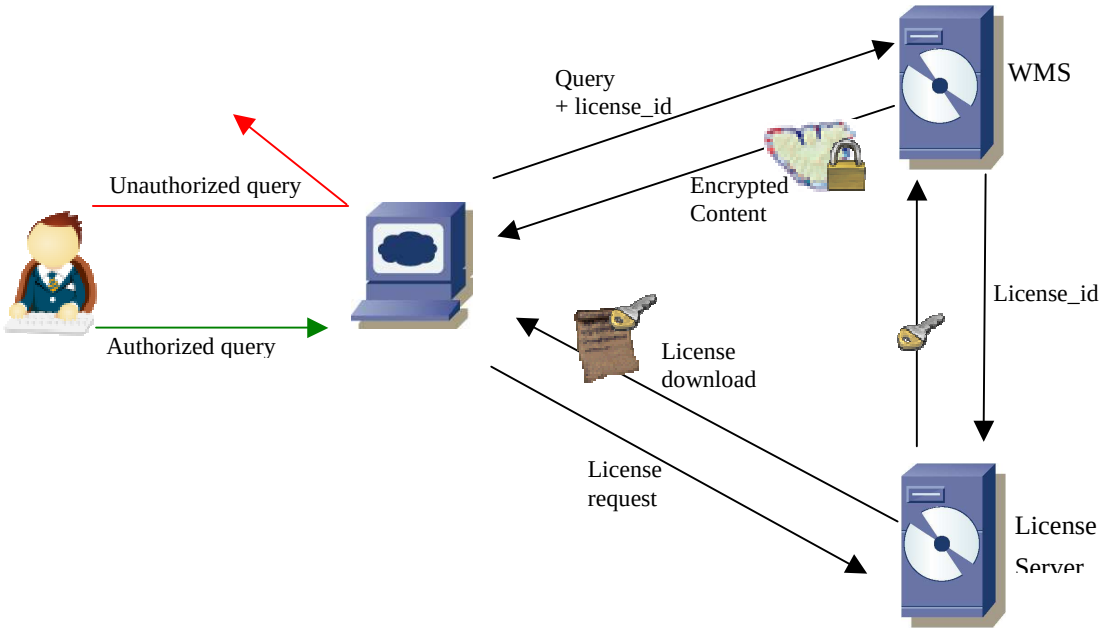
Our paper is more on how to write an ODRL WMS-profile than to deal with the security mechanisms for enforcing the rights policy. However, in this section, we sketch a security architecture for regulating access and usage of WMS contents (see Figure 3). First the client downloads a license from the license server, together with a decryption key. The licence contains the security policy defining the permitted queries. Unauthorized queries are filtered-out by the DRM-enabled client. Recall that in a DRM application, clients are trusted and perform the security controls. Authorized queries are sent to the WMS together with a reference to the licence. The WMS acquires the encryption key corresponding to the license from the license server. The WMS encrypts the content it has created as a result of the query and sends it to the client. The client decrypts the content. Differences between our WMS-DRM architecture and a traditional DRM architecture are the followings:

- In traditional DRM applications, encryption/decryption keys are associated to contents. In our architecture, since the Web Map Service dynamically creates various contents in response to various queries, we cannot associate encryption/decryption keys to contents. We rather associate keys to licenses. Whenever the DRM-enabled client sends a query to the WMS it also sends the license id. This id is used by the WMS to retrieve the corresponding encryption key from the license server. Note that encryption keys can be periodically renewed. Whenever, expiration date for the key has passed the user should acquire a new key for his/her license.
- In traditional DRM applications, queries are not filtered-out by the DRM-enabled client. If a user asks for a song he/she has not the proper license then he/she will simply not be able to play the song. In our application, since encryption keys are associated to licenses and not to contents, filtering-out unauthorized queries should be performed somewhere. It could be done at the server side but it would require the server (i) to retrieve the license either from the client or the license server and (ii) to decide on accepting or rejecting the query. Such a solution would not adequate to the DRM philosophy and would create overhead tasks at the server side. Now, the risk of our architecture is that the user manages to bypass the DRM client to submit an unauthorized query (together with a valid licence_id) to the WMS. In order to reduce such a risk, the DRM-enabled client has to communicate with the WMS via an authenticated and secure channel. In fact, as a generalisation, the three entities (Client, License Server and WMS) should always communicate through secure and authenticated channels.

Finally, let us mention that once encrypted contents (basically maps) have been decrypted, usage of these maps is controlled by

the DRM-enabled client according to the policy included in the license. At the end of section 5 there is a very simple example of such a usage policy.

Fig 3. : DRM architecture



7. CONCLUSION

The purpose of this paper was to build the foundations for an ODRL WMS-profile. Writing a complete profile, not limited to the basic WMS, could be one of the task of the ODRL Geospatial Profile Working Group. Ideas developed in this work can be reused in other RELs or in Access Control languages like GeoXACML [18]. Our approach could also be followed for writing an ODRL WFS profile.

In this paper we focused on WMS queries but not on the contents created by the WMS. As seen in Section 2, these contents can be of several types: raster images, SVG images or getCapabilities XML documents. Therefore, in a full WMS DRM-controlled environment, we would need specific ODRL profiles for these contents. Regarding raster images, we would need specific permissions like rotation, reduction, editing, segmentation etc. Regarding SVG images, we would need specific permissions not only on the image as a whole but also on some individual vector data belonging to the image. The model defined in [19] could help us to write such a profile. Regarding getCapabilities XML documents, we would need specific permissions for filtering out some XML nodes (those which refer to the layers that are forbidden to query for instance). For this latter profile, we could benefit from the large amount of work which was done in the area of access controls to XML data (see [20, 21] for instance).

We plan to extend our work by developing such profiles and by building a prototype of a WMS DRM-enabled platform based on the principles sketched in Section 6.

8. References

[1] Bertino, E., et al., A Logical Framework for Reasoning about Access Control Models, in ACM Workshop on Role Based Access Control. 2003: Chantilly, Virginia. p. 41-52.

[2] Jajodia, S., et al., Flexible Support for Multiple Access Control Policies. ACM Transactions on Database Systems, 2001. 26(2): p. 214-260.

[3] Cuppens, F. and A. Miège, Modelling Contexts in the Or-BAC Model, in 19th Annual Computer Security Applications Conference (ACSAC '03). 2003.

[4] J.Park and R.Sandhu, The UCON-ABC Usage Control Model. ACM Transactions on Information and System Security, 2004. 7(1): p. 128-174.

[5] LaMacchia, B.A., An introduction to rights Management Technologies, in GeoDRM Workshop. 2004.

[6] Iannella, R., Open Digital Rights Language (ODRL), in ODRL.net. 2002.

[7] ContentGuard, XrML 2.0 Technical Overview. 2002.

[8] Rightscom, The MPEG-21 Rights Expression Language. 2003.

[9] Volwes, G., Geospatial Digital Rights Management Reference Model (GeoDRM RM). Open Geospatial Consortium Inc., 2006. OGC(R) 06-004r3.

[10] Consortium, O.G. Open Geospatial Consortium Inc. - About Us. 2008 [cited; Available from: http://www.opengeospatial.org/about.

[11] Beaujardiere, J.d.l., OpenGIS(R) Web Map Server Implementation Specification. Open Geospatial Consortium Inc., 2006. OGC(R) 06-042.

[12] Vretanos, P.A., OpenGIS(R) Filter Encoding Implementation Specification. Open Geospatial Consortium Inc., 2005. OGC(R) 04-095.

[13] Portele, C., OpenGIS(R) Geography Markup Language (GML) Encoding Standard. Open Geospatial Consortium Inc., 2007. OGC(R) 07-036.

[14] Kropla, B., Beginning MapServer - Open Source GIS Development. 2005: Springer. 448.

[15] Guth, S., ODRL Initiative Response to LTSC DREL Requirements. 2004.

[16] Matheus, A., Authorization for digital rights Management in the geospatial domain, in 5th ACM workshop on Digital rights management. 2005: Alexandria, VA, USA. p. 55-64.

[17] Atluri, V. and S.A. Chun, A geotemporal role-based authorisation system. International Journal of Information and Computer Security, 2007. 1(1/2): p. 143-168.

[18] Matheus, A. and J. Herrmann, Geospatial eXtensible Access Control Markup Language (GeoXACML). Open Geospatial Consortium Inc., 2008. OGC(R) 07-026r2.

[19] Damiani, M.L., et al., GEO-RBAC : A spatially Aware RBAC. ACM Tansactions on Information Systems and Security, 2006. 00(00): p. 1-34.

[20] Gabillon, A. and E. Bruno, Regulating Access to XML documents, in Fifteenth Annual IFIP WG 11.3 Working Conference on Database Security. 2001: Niagara on the Lake, Ontario, Canada.

[21] Damiani, E., et al., Securing XML Documents, in 2000 International Conference on Extending Database Technology (EDBT2000). 2000: Konstanz, Germany.