Open Geospatial Consortium Inc.

Date: 2008-09-12

Reference number of this document: OGC 08-076

Version: 0.9

Category: OpenGIS® Discussion Paper

Editor: Rüdiger Gartmann, con terra GmbH

OpenGIS® OWS-5 GeoRM License Broker Discussion Paper

Warning

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

Document type: OpenGIS® Discussion Paper
Document subtype: OWS-5 Engineering report
Document stage: Discussion Paper
Document language: English

Contents

Page

i. Preface

This Discussion Paper documents the results from work done in the OGC Web Services Phase 5 (OWS-5) Test Bed. Specifically, the document describes the License Broker Service (LB-Service) that was developed and implemented within the Geoprocessing & Workflow (GPW) Thread.

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards. In particular, the word "SHALL" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

con terra GmbH

iv. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Rüdiger Gartmann	con terra GmbH

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2008-05-15	.9	RG	All	Document initialized
2008-07-01	.9	Carl Reed	Many	Get ready for posting as DP

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

vii. Future work

This document is a test bed result, specifying a License Broker Service (LB-Service). Based on this approach, this document should be matured and submitted as a candidate implementation standard. Therefore, the document needs improvements on required elements of implementation specifications such as an abstract test suite, UML models etc.

Foreword

This License Broker Service (LB-Service) Discussion Paper is an Engineering Report resulting from the work in the OWS-5 test bed. A LB-Service allows automatic license conclusion and thus fills the gap in the "GeoDRM Engineering Viewpoint and Supporting Architecture" document (OGC #06-184r2), which defined a Geo Rights Management (GeoRM) infrastructure allowing license based access control to services as a result of the OWS-4 testbed.

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 Inc. 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

Introduction

This document describes a License Broker Service (LB-Service) as specified and implemented in the OWS-5 test bed. The LB-Service provides configurable license models, which may contain configuration parameters to be defined by the licensee. The setting of these parameters affects the actual license to be created by the LB-Service.

An LB-Service client may request available license models for a given resource. This may result in a set of license models covering the requested resource. If the result set is empty, no license models covering the requested resource are available, if the result includes more than one license model, the client may choose among them.

The configuration parameters defined in each license model have to be defined by the client in order to conclude a license. Since from the legal perspective a license conclusion is a contract between a licensee and a licensor, the identities of both are included in a OrderLicense request.

In order to allow non-binding trial orders, License Broker servers may offer a TryLicense operation that evaluates an actual license configuration. The evaluation may result in a set of conditions which would apply in case of a license conclusion with the actual parameter set. This allows for instance to define a price for a certain license conclusion.

The OrderLicense request results in a license creation. A license is maintained remotely and referenceable. Thus, the OrderLicense response includes a reference to the created license instead of the license itself. This allows an easy license revocation process, which just needs to invalidate the actual license. Then this license cannot be used any more, although there might still be license references available.

This LB-Service interface fits into the OWS-4 GeoDRM architecture defined in OGC document #06-184r2 and thus completes the OWS-4/OWS-5 GeoRM suite.

OpenGIS® License Broker Service Implementation Specification ER

1 Scope

This OGC® document specifies a License Broker Service (LB-Service) being able to conclude configurable licenses for geospatial resources. This LB-Service service can be used in combination with the OWS-4 GeoDRM infrastructure described in OGC document #06-184r2 in order to conclude licenses which grant access to geospatial resources.

2 Normative references

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

ISO 19105:2000, *Geographic information — Conformance and Testing*

OGC 06-121r3, *OpenGIS® Web Services Common Specification*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

OASIS *Security Assertion Markup Language (SAML) v2.0*

W3C *XML-Signature Syntax and Processing (W3C Recommendation 12 February 2002)*

In addition to this document, this standard includes several normative XML Schema Document files as specified in Annex A.

3 Terms and definitions

For the purposes of this standard, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] and in OpenGIS® Abstract Specification Topic 18 - Geospatial Digital Rights Management Reference Model (GeoDRM RM) [06-004r4] SHALL apply. In addition, the following terms and definitions apply.

3.1 License Broker Service

Service offering means to conclude a license.

3.2 License Manager Service

Service maintaining licenses and resolving license references

3.3 License Reference

This is a reference to a license that is stored in a License Manager Service. This reference can be used as evidence for the license whenever necessary.

4 Conventions

4.1 Abbreviated terms

Most of the abbreviated terms listed in Sub-clause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

SAML Security Assertion Markup Language

4.2 UML notation

Diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 05-008].

4.3 Used parts of other documents

This document uses significant parts of document [OGC 05-008]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

Table 1 — Contents of data dictionary tables

Column title	Column contents
Names (left column)	Two names for each included parameter or association (or data structure). The first name is the UML model attribute or association role name. The second name uses the XML encoding capitalization specified in Subclause 11.6.2 of [OGC 05-008]. The name capitalization rules used are specified in Subclause 11.6.2 of [OGC 05-008]. Some names in the tables may appear to contain spaces, but no names contain spaces.
Definition (second column)	Specifies the definition of this parameter (omitting un-necessary words such as "a", "the", and "is"). If the parameter value is the identifier of something, not a description or definition, the definition of this parameter should read something like "Identifier of TBD".
Data type and value (third column) or Data type (if are no second items are included in rows of table)	Normally contains two items: The mandatory first item is often the data type used for this parameter, using data types appropriate in a UML model, in which this parameter is a named attribute of a UML class. Alternately, the first item can identify the data structure (or class) referenced by this association, and references a separate table used to specify the contents of that class (or data structure). The optional second item in the third column of each table should indicate the source of values for this parameter, the alternative values, or other value information, unless the values are quite clear from other listed information.
Multiplicity and use (right or fourth column) or Multiplicity (if are no second items are included in rows of table)	Normally contains two items: The mandatory first item specifies the multiplicity and optionality of this parameter in this data structure, either "One (mandatory)", "One or more (mandatory)", "Zero or one (optional)", or "Zero or more (optional)". The second item in the right column of each table should specify how any multiplicity other than "One (mandatory)" shall be used. If that parameter is optional, under what condition(s) shall that parameter be included or not included? If that parameter can be repeated, for what is that parameter repeated?

When the data type used for this parameter, in the third column of such a table, is an enumeration or code list, all the values specified by a specific OWS shall be listed, together with the meaning of each value. When this information is extensive, these values and meanings should be specified in a separate table that is referenced in the third column of this table row.

The data type of many parameters, in the third table column, is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

The contents of these data dictionary tables are normative, including any table footnotes.

5 License Broker Service overview

The specified License Broker Service (LB-Service) allows the conclusion of licenses for certain resources. A resource can be a service, a data file or a process. The LB-Service supports the negotiation and conclusion process by offering license templates

The LB-Service interface specifies four operations that can be requested by a client and performed by a LB-Server. Those operations are:

a) GetCapabilities (required implementation by servers) – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. This operation also supports negotiation of the standard version being used for client-server interactions.

b) GetLicenseModel – This operation allows a client to request and receive back detailed information on license option for a given resource, including configuration parameters allowing a client to define license parameters.

c) TryLicense (optional implementation by servers) – This operation allows a client to submit a license configuration in order to check the configuration's validity. This operation does not lead to conclusion of a license.

d) OrderLicense – This operation allows a client to conclude a license, submitting all information required by a license model, including required configuration parameters. The response is a license reference token which references the license at the place where it is maintained (usually a License Manager Service).

These operations have many similarities to other OGC Web Services, including the WMS, WFS, and WCS. Many of these interface aspects that are common with other OWSs are thus specified in the OpenGIS® Web Services Common Implementation Specification [OGC 05-008]. Many of these common aspects are normatively referenced herein, instead of being repeated in this standard.

Figure 1 is a simple UML diagram summarizing the LB-Service interface. This class diagram shows that the LB-Service interface class inherits the getCapabilities operation from the OGCWebService interface class, and adds the LB-Service operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the LB-Service interface is provided in Annex C (informative).

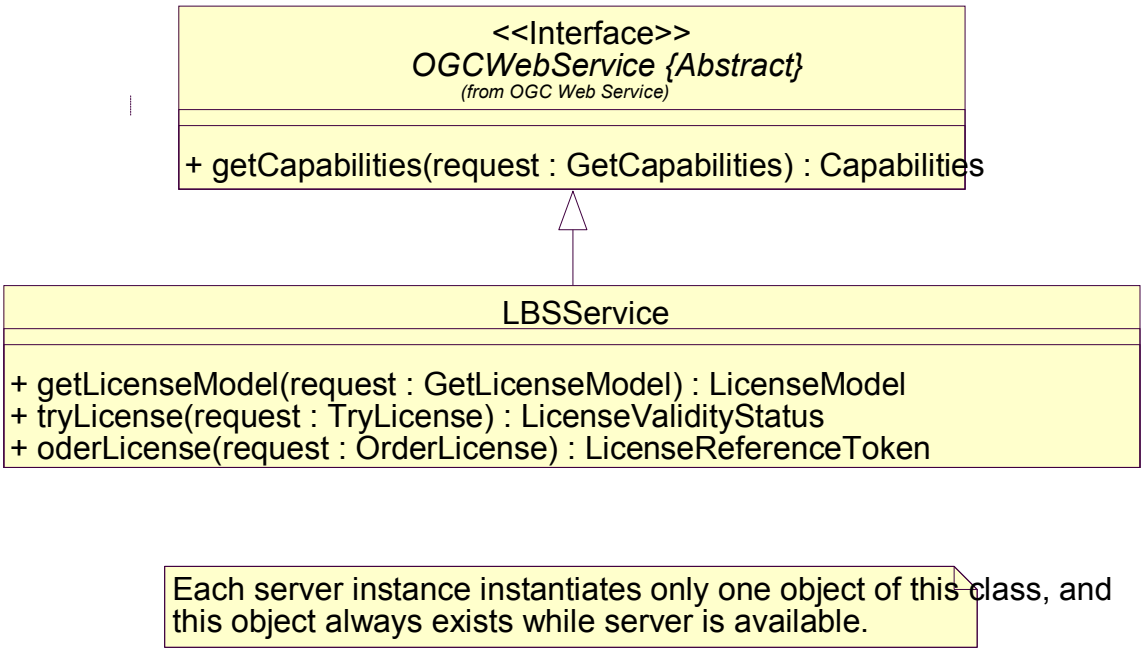

Figure 1 — LB-Service interface UML diagram

NOTE In this UML diagram, the request and response for each operation is shown as a single parameter that is a data structure containing multiple lower-level parameters, which are discussed in subsequent clauses. The UML classes modeling these data structures are included in the complete UML model in Annex C.

Each of the LB-Service operations is described in more detail in subsequent clauses.

6 Shared aspects

6.1 Introduction

This clause specifies aspects of the LB-Service behavior that are shared by several operations.

6.2 Operation request encoding

The encoding of operation requests SHALL use HTTP POST with XML, SOAP encoding as specified in Clause 11 of [OGC 05-008]. Table 2 summarizes the LB-Service operations and their encoding methods defined in this standard.

Table 2 — Operation request encoding

Operation name	Request encoding
GetCapabilities (required)	KVP or XML
GetLicenseModel (required)	XML
TryLicense (optional)	XML

OrderLicense	XML

7 GetCapabilities operation (mandatory)

7.1 Introduction

The mandatory GetCapabilities operation allows clients to retrieve service metadata from a server. The response to a GetCapabilities request **SHALL** be an XML document containing service metadata about the server, including specific information about the LB-Service. This clause specifies the XML document that a LB-Server **SHALL** return to describe its capabilities.

7.2 Operation request

The GetCapabilities operation request shall be as specified in Subclauses 7.2 and 7.3 of [OGC 05-008]. The value of the "service" parameter **SHALL** be "LBS". The allowed set of service metadata (or Capabilities) XML document section names and meanings **SHALL** be as specified in Tables 3 and 7 of [OGC 05-008], with the additions listed in Table 3 below.

Table 3 — Additional parameter name values and meanings

Name	Definition	Data type and values	Multiplicity and use
language	Defines the language in which the LBS has to respond	xs:language, language identifier as specified in IETF RFC 4646	Zero or one (optional)

The "Multiplicity and use" column in Table 1 of [OGC 05-008] specifies the optional status of each listed parameter in the GetCapabilities operation request. Table 4 specifies the implementation of those parameters by LB-Service clients and servers.

Table 4 — Implementation of parameters in GetCapabilities operation request

Names	Multiplicity	Client implementation	Server implementation
service service	One (mandatory)	Each parameter shall be implemented by all clients, using specified value	Each parameter shall be implemented by all servers, checking that each parameter is received with specified value
request request	One (mandatory)		
AcceptVersions AcceptVersions	Zero or one (optional)	Should be implemented by all software clients, using specified values	Shall be implemented by all servers, checking if parameter is received with specified value(s)
Sections Sections	Zero or one (optional) [b]	Each parameter may be implemented by each client [b] If parameter not provided, shall expect default response	Each parameter may be implemented by each server [a] If parameter not implemented or not received, shall provide
updateSequence updateSequence	Zero or one (optional) [b]		

AcceptFormats AcceptFormats	Zero or one (optional) [b]	If parameter provided, shall allow default or specified response	default response If parameter implemented and received, shall provide specified response
language	Zero or one (optional) [b]	If parameter provided, shall expect response in requested language If parameter not provided, shall expect response in default language	If parameter not implemented or not received, shall provide response in default language If parameter implemented and received, shall provide response in specified language
a A specific OWS is allowed to make mandatory server implementation of any of these three parameters. b If a specific OWS makes mandatory server implementation of any of these three parameters, that parameter can also be made mandatory in the operation request, also requiring client implementation of this parameter.			

All LB-Servers **SHAL**L implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers **SHALL** also implement SOAP transfer via HTTP POST of the GetCapabilities operation request, using XML encoding only.

7.3 GetCapabilities operation response

7.3.1 Normal response

The service metadata document **SHALL** contain the optional sections specified in Table 5. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections can be requested and **SHALL** be returned when requested.

Table 5 — Section name values and contents

Section name	Contents
ServiceIdentification	Metadata about this specific server. The schema of this section SHALL be the same as for all OWSs, as specified in Subclause 7.4.3 and owsServiceIdentification.xsd of [OGC 05-008].
ServiceProvider	Metadata about the organization operating this server. The schema of this section SHALL be the same for all OWSs, as specified in Subclause 7.4.4 and owsServiceProvider.xsd of [OGC 05-008].
OperationsMetadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The basic contents and organization of this section SHALL be the same as for all OWSs, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008].
SupportedResources	Metadata about the data served by this server. For the LB-Server, this section SHALL contain data about resources the LB-Server offers licenses for, as specified in Subclause 7.3.3 below.

In addition to these sections, each service metadata document **SHALL** include the mandatory "version" and optional updateSequence parameters specified in Table 6 in Subclause 7.4.1 of [OGC 05-008].

7.3.2 OperationsMetadata section standard contents

For the LB-Service, the OperationsMetadata section **SHALL** be the same as for all OGC Web Services, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008]. The mandatory values of various (XML) attributes SHALL be as specified in Table 6. Similarly, the optional attribute values listed in Table 7 SHALL be included or not depending on whether that operation is implemented by that server. In Table 6 and Table 7, the "Attribute name" column uses dot-separator notation to identify parts of a parent item. The "Attribute value**"** column references an operation parameter, in this case an operation name, and the meaning of including that value is listed in the right column.

Table 6 — Required values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	GetCapabilities	The GetCapabilities operation is implemented by this server.
	GetLicenseModel	The GetLicenseModel operation is implemented by this server.
	OrderLicense	The OrderLicense operation is implemented by this server.

Table 7 — Optional values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	TryLicense	The TryLicense operation is implemented by this server.

In addition to the optional values listed in Table 6, there are many optional values of the "name" attributes and "value" elements in the OperationsMetadata section, which may be included when considered useful. Most of these attributes and elements are for recording the domains of various parameters and quantities.

EXAMPLE 1 The domain of the exceptionCode parameter could record all the codes implemented for each operation by that specific server. Similarly, each of the GetCapabilities operation optional request parameters might have its domain recorded.

EXAMPLE 2 The domain of the Sections parameter in the GetCapabilities operation request could record all the sections implemented by that specific server.

All LB-Servers SHALL specify the encodings that may be sent using HTTP POST transfer of operation requests. For LB-Services, an ows:Constraint element SHALL be included, with "PostEncoding" as the value of the "name" attribute and specifying "SOAP" as the value for each allowed encoding as specified in Subclause 11.8.

If the HTTP POST connect point URL is different for different encodings of the operation requests, this ows:Constraint element SHALL be included in each Post element. If the connect point URL is the same for all encodings of all operation requests, this ows:Constraint element SHALL be included in the OperationsMetadata element.

7.3.3 SupportedResources section contents

The SupportedResources section of a service metadata document contains metadata about the resources this server offers licenses for.

Table 8 — SupportedProductIDs data structure

Names	Definition	Data type and values	Multiplicity and use
SupportedResources	List of all resource IDs which the LBS offers licenses for.	Character string, not empty	One or more (mandatory)

The "Multiplicity and use" columns in Table 6 through Table 16 in [OGC 05-008], and Table 8 of this document, specify the optionality of each listed parameter and data structure in the GetCapabilities operation response. All the "mandatory" parameters and data structures SHALL be implemented by all OWS servers, using a specified value(s).

The "updateSequence" parameter defined in Table 6 of [OGC 05-008] is optional implementation by OWS servers. As indicated in Table 4 of this document, the "updateSequence" parameter may be implemented by each server, but a specific OWS is allowed to require or prohibit server implementation of this parameter. If a specific OWS requires server implementation of this parameter, this parameter SHALL also be required in the operation response. Similarly, if a specific OWS prohibits server implementation of this parameter, this parameter SHALL also be prohibited in the operation response.

All other "optional" parameters and data structures, in the GetCapabilities operation response, should be implemented by all OWS servers using specified values, whenever and wherever each is considered useful metadata for that server.

7.3.4 Capabilities document XML encoding

A XML schema fragment for a LB-Service metadata document extends ows:CapabilitiesBaseType in owsCommon.xsd of [OGC 05-008], and is:

```
<xs:complexType name="LBSGetCapabilitiesResponseType">
   <xs:complexContent>
      <xs:extension base="ows:CapabilitiesBaseType">
         <xs:sequence>
            <xs:element name="SupportedResources"
      type="xs:string" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="language" type="xs:language"
      use="optional"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

7.3.5 Capabilities document example

In response to GetCapabilities operation request, a LB-Server might generate a document that looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<lbs:LBSGetCapabilitiesResponse version="0.5.0"
xsi:schemaLocation="http://www.conterra.de/lbs
LicenseBroker_0_5.xsd" xmlns:lbs="http://www.conterra.de/lbs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2001/XMLSchema" >
   <ows:ServiceIdentification>
      <ows:Title>OWS-5 LBS</ows:Title>
      <ows:Abstract>Demo Capabilities</ows:Abstract>
      <ows:ServiceType>LBS</ows:ServiceType>
      <ows:ServiceTypeVersion>0.5.0</ows:ServiceTypeVersion>
      <ows:Fees>none</ows:Fees>
      <ows:AccessConstraints>none</ows:AccessConstraints>
   </ows:ServiceIdentification>
   <ows:ServiceProvider>
      <ows:ProviderName>con terra</ows:ProviderName>
      <ows:ServiceContact>
         <ows:IndividualName>Rüdiger
Gartmann</ows:IndividualName>
         <ows:ContactInfo>
            <ows:Phone>
               <ows:Voice>+49 251 7474-301</ows:Voice>
            </ows:Phone>
            <ows:Address>
               <ows:DeliveryPoint>Martin-Luther-King-Weg
24</ows:DeliveryPoint>
               <ows:City>Münster</ows:City>
               <ows:PostalCode>D-48155</ows:PostalCode>
               <ows:Country>Germany</ows:Country>

   <ows:ElectronicMailAddress>r.gartmann@conterra.de</ows:Electro
nicMailAddress>
            </ows:Address>
         </ows:ContactInfo>
      </ows:ServiceContact>
   </ows:ServiceProvider>
   <ows:OperationsMetadata>
      <ows:Operation name="GetCapabilities">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="www.service.url"/>
               <ows:Post xlink:href="www.service.url">
                  <ows:Constraint name="PostEncoding">
                     <ows:AllowedValues>
                        <ows:Value>SOAP</ows:Value>
                     </ows:AllowedValues>
                  </ows:Constraint>
               </ows:Post>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
```



```
      <ows:Operation name="GetLicenseModel">
         <ows:DCP>
            <ows:HTTP>
               <ows:Post xlink:href="www.service.url">
                  <ows:Constraint name="PostEncoding">
                     <ows:AllowedValues>
                        <ows:Value>SOAP</ows:Value>
                     </ows:AllowedValues>
                  </ows:Constraint>
               </ows:Post>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="TryLicense">
         <ows:DCP>
            <ows:HTTP>
               <ows:Post xlink:href="www.service.url">
                  <ows:Constraint name="PostEncoding">
                     <ows:AllowedValues>
                        <ows:Value>SOAP</ows:Value>
                     </ows:AllowedValues>
                  </ows:Constraint>
               </ows:Post>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="OrderLicense">
         <ows:DCP>
            <ows:HTTP>
               <ows:Post xlink:href="www.service.url">
                  <ows:Constraint name="PostEncoding">
                     <ows:AllowedValues>
                        <ows:Value>SOAP</ows:Value>
                     </ows:AllowedValues>
                  </ows:Constraint>
               </ows:Post>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
   </ows:OperationsMetadata>
   <lbs:SupportedResources>Resource1</lbs:SupportedResources>
   <lbs:SupportedResources>Resource2</lbs:SupportedResources>
</lbs:LBSGetCapabilitiesResponse>
```

7.3.6 GetCapabilities Exceptions

When a LBS server encounters an error while performing a GetCapabilities operation, it SHALL return an exception report message as specified in Clause 8 of [OGC 05-008]. The allowed exception codes SHALL include those listed in Table 5 of Subclause 7.4.1 of [OGC 05-008], if the updateSequence parameter is implemented by the server.

Additionally, the following exception is allowed:

exceptionCode value	Meaning of code	"locator" value
LanguageNotSupported	Request is for a language that is not supported by this server	Codes of supported languages

8 GetLicenseModel operation (mandatory)

8.1 Introduction

The GetLicenseModel operation allows LB-Service clients to request a complete description of available license models for a given ressource. Those license models include configuration parameters and legal information associated with the actual license model.

This description is used to enable a client to display all license-relevant information and to request all specified configuration parameters from a user.

8.2 GetLicenseModel operation request

8.2.1 GetLicenseModel request parameters

The GetLicenseModel request uses attributes to define the requested service type has to be "LBS", the requested version of the License Broker Service, and optionally the preferred language.

This operation is used to retrieve license models for a given resource, identified by a "ResourceID". As a result zero or more license models for the requested resource will be returned. Thise license models include configuration parameters reqresenting license variables which have to be filled by the user. Furthermore in may include several textual attribute statements which have to be displayed to the user, since they are license parts and have to be accepted by the user before concluding a license, which is a legal act.

A request to perform the GetLicenseModel operation SHALL include the use of the data structure specified in Table 9 and Table 10.

Table 9 — Parameters in GetLicenseModel operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is "LBS"	One (mandatory)
version	Standard version for operation	Character String type, not empty Value is specified by each Implementation Standard and Schemas version	One (mandatory)
language	Language identifyer	xs:language type, RFC4646 language code. Must be a language listed in the Capabilities Languages element.	Zero or one (optional)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

Table 10 — Sections in GetLicenseModel operation request

Name	Definition	Multiplicity and use
ResourceID	ID of the resource to be licensed	One or more (mandatory)

NOTE 2 The data type of many parameters is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

8.2.2 GetLicenseModel request XML encoding (mandatory)

All License Broker Service servers SHALL implement SOAP via HTTP POST transfer of the GetLicenseModel operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a GetLicenseModel operation request encoded in XML:

```
<xs:element name="GetLicenseModel">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="ResourceID" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="service" type="xs:string"
use="required"/>
      <xs:attribute name="version" type="xs:string"
use="required"/>
      <xs:attribute name="language" type="xs:string"/>
   </xs:complexType>
</xs:element>
```


8.3 GetLicenseModel operation response

8.3.1 Normal response parameters

The normal response to a valid GetLicenseModel operation request SHALL contain zero or more licenseModel sections as listed in Table 12. Additionally it may contain a language attribute as defined in Table 11.

Table 11 — Parameters of the GetLicenseModel response

Names	Definition	Data type and values	Multiplicity and use
language	Language of the actual license model	xs:language type, RFC4646 language code. Must be a language listed in the Capabilities Languages element.	Zero or one (optional)

Table 12 — Section name values and contents of the GetLicenseModel operation response

Section Name	Contents	Data Type	Multiplicity and use
LicenseModel	Definition of license model	License model type, see section 8.3.1.1	zero or more

8.3.1.1 LicenseModelType Contents

A LicenseModel section contains an ID as parameter, indicating a license model ID which is unique for any LBS service as described in Table 13.

Table 13 — Parameters of the LicenseModel section

Names	Definition	Data type and values	Multiplicity and use
id	License model identifier	Character string type, not empty	One (mandatory)

Furthermore, a license model contains one or more Resource sections, defining the resources covered by the actual license model, zero or more LicenseAttribute elements, carrying a mandatory name and an optional title attribute for describing arbitrary license information, and a Licensor section, relating to the license issuer. These sections are listed in Table 14.

Table 14 — Section name values and contents of the LicenseModel section

Section Name	Contents	Data Type	Multiplicity and use
Resource	Definition of resource being targeted by the license model	Resource type, see section 8.3.1.2	One or more (mandatory)
LicenseAttribute	Attributes attached to the license model	Character string type, not empty	Zero or more (optional)
Licensor	Licensor identification	Licensor type, see section 8.3.1.7	One (mandatory)

8.3.1.2 ResourceType Contents

The resource section defines the resources covered by the actual license model, including configuration options. The parameters of this section are listed in Table 15:

Table 15 — Parameters of the ResourceType

Names [a]	Definition	Data type and values	Multiplicity and use
id	Resource identifier	Character string type, not empty	One (mandatory)
name	Resource name	Character string type, not empty	Zero or one (optional)
title	Resource title	Character string type, not empty	Zero or one (optional)

The id parameter uniquely identifies the actual resource, which can also carry a unique name or a title which may differ in different languages.

The Resource section contains sub-sections as listed in Table 16.

Table 16 — Section name values and contents of the ResourceType

Section Name	Contents	Data Type	Multiplicity and use
Abstract	Resource description	Character string type, not empty	Zero or one (optional)
ConfigParams	Configuration parameters	Config params type, see table	Zero or more (optional)
ResourceAttribute	Any resource-specific information	Character string type, not empty	Zero or more (optional)

The Abstract element contains a resource description which may depend on the language attribute. The ConfigParams section is further described in section 8.3.1.3. Furthermore, arbitrary resource attributes may be defined, carrying a name and an optional title attribute. The latter may carry a language dependant title whereas the name is fixed for all languages.

8.3.1.3 ConfigParamsType Contents

Table 17 — Section name values and contents of the ConfigParamsType

Section Name	Contents	Data Type	Multiplicity and use
Parameter	Configuration parameter	Parameter type, see section	Zero or more (optional)

8.3.1.4 ParameterType Contents

The parameter type has two mandatory attributes, a name and a type. The type is used to specify the expected data type of the parameter. Additionally, a title may be defined, being language dependant.

Table 18 — Parameters of the ParameterType

Names	Definition	Data type and values	Multiplicity and use
name	Parameter name	Character string type, not empty	One (mandatory)
title	Parameter title (language specific)	Character string type, not empty	Zero or one (optional)
type	Parameter type	xs:AnySimpleType	One (mandatory)

The parameter type includes one of the elements described in Table 19:

Table 19 — Section name values and contents of the ParameterType

Section Name	Contents	Data Type	Multiplicity and use
Value	Parameter value	ValueType, see section 8.3.1.5	Either one (mandatory)
Select	Multi-select values	Select Type, see section 8.3.1.6	

8.3.1.5 ValueType Contents

A ValueType is a character string which may be empty. If it is not empty it SHALL be interpreted as default value. It includes the following parameters:

Table 20 — Parameters of the ParameterType

Names	Definition	Data type and values	Multiplicity and use
title	Value title, may be language specific	Character string type, not empty	Zero or one (optional)
selected	Indicates if value is selected	xs:boolan	Zero or one (optional)

8.3.1.6 SelectType Contents

A select type contains one parameter as described in Table 24

Table 21 — Parameters of the SelectType

Names	Definition	Data type and values	Multiplicity and use
multiple	Defines if multiple values are allowed for this parameter	xs:boolean	Zero or one (optional), false if omitted

Furthermore, it contains the following element:

Table 22 — Section name values and contents of the ParameterType

Section Name	Contents	Data Type	Multiplicity and use
Value	Parameter value	ValueType, see section 8.3.1.5	One or more (mandatory)

8.3.1.7 LicensorType Contents

The licensor element is used to identify the license issuer. It may contain the following parameter:

Table 23 — Parameters of the LicensorType

Names [a]	Definition	Data type and values	Multiplicity and use
id	Licensor ID	Character string type, not empty	Zero or one (optional)

The licensor type elements are described in Table 24.

Table 24 — Section name values and contents of the LicensorType

Section Name	Contents	Data Type	Multiplicity and use
LicensorAddress	Contact information of Licensor	Address type, see table	One (mandatory)
Subject	SAML subject of the licensor	SAML SubjectType, see [1], clause 2.4.1	One (mandatory)

8.3.1.8 AddressType Contents

The address type contains contact information for the subject described.

Table 25 — Section name values and contents of the AddressType

Section Name	Contents	Data Type	Multiplicity and use
Name	Contact name	Character string type, not empty	One (mandatory)
Forename	Contact forename	Character string type, not empty	One (mandatory)
Title	Contact title	Character string type, not empty	Zero or one (optional)
CompanyName	Name of Company	Character string type, not empty	Zero or one (optional)
Street	Street address	Character string type, not empty	One (mandatory)
Streetnum	Street number	Character string type, not empty	One (mandatory)
Zip	Zip code	Character string type, not empty	One (mandatory)
City	City name	Character string type, not empty	One (mandatory)
State	State name	Character string type, not empty	Zero or one (optional)
Country	County name	Character string type, not empty	One (mandatory)
Phone	Phone number	Character string type, not empty	Zero or one (optional)
Fax	Fax number	Character string type, not empty	Zero or one (optional)
Email	Email address	Character string type, not empty	One (mandatory)

8.3.2 GetLicenseModel Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2007 sp2
(http://www.altova.com)-->
<lbs:LBSGetLicenseModelResponse language="en-us"
xsi:schemaLocation="http://www.conterra.de/lbs
LicenseBroker_0_5.xsd" xmlns:lbs="http://www.conterra.de/lbs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
   <lbs:LicenseModel ID="License1">
      <lbs:Resource id="Resource1" title="A Resource"
name="resource">
         <lbs:Abstract>This is the description of resource
1.</lbs:Abstract>
         <lbs:ConfigParams>
            <lbs:Parameter type="int" title="Number"
name="number">
               <lbs:Value>7</lbs:Value>
            </lbs:Parameter>
```



```
            </lbs:ConfigParams>
            <lbs:ProductAttribute title="Product Info" name="info">
            This is some information
            </lbs:ProductAttribute>
         </lbs:Resource>
         <lbs:LicenseAttribute title="Disclaimer"
name="disclaimer">This is a disclaimer!</lbs:LicenseAttribute>
         <lbs:Licensor ID="conterra">
            <lbs:LicensorAddress>
               <lbs:name>Rüdiger</lbs:name>
               <lbs:forename>Gartmann</lbs:forename>
               <lbs:street>Martin-Luther-King-Weg</lbs:street>
               <lbs:streetnum>24</lbs:streetnum>
               <lbs:zip>48155</lbs:zip>
               <lbs:city>Münster</lbs:city>
               <lbs:country>Germany</lbs:country>
               <lbs:phone>+49 251 7474-301</lbs:phone>
               <lbs:email>r.gartmann@conterra.de</lbs:email>
            </lbs:LicensorAddress>
            <saml:Subject>
               <saml:NameID>ruedigergartmann</saml:NameID>
            </saml:Subject>
         </lbs:Licensor>
      </lbs:LicenseModel>
</lbs:LBSGetLicenseModelResponse>
```

8.3.3 GetLicenseModel exceptions

When a LBS server encounters an error while performing a GetLicenseModel operation, it SHALL return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes SHALL include those listed in Table 10. For each listed exceptionCode, the contents of the "locator" parameter value SHALL be as specified in the right column of Table 26.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 26 — Exception codes for GetLicenseModel operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
OptionNotSupported	Request is for an option that is not supported by this server	Identifier of option not supported
UnknownResourceID	No license model for the given ProductID available	ProductID

LanguageNotSupported	Request is for a language that is not supported by this server	Codes of supported languages

9 TryLicense Operation (Optional)

9.1 Introduction

The TryLicenseModel request uses attributes to define the requested service type has to be "LBS", the requested version of the License Broker Service, and optionally the preferred language.

The TryLicense operation submits a fully configured license request to the LBS but does not, in contrast to the OrderLicense request which uses the same request encoding, in a license conclusion. Moreover, the TryLicense response informs about the feasibility of a certain license configuration and about conditions (for instance a price) which would apply to a license conclusion with the given configuration.

Thus, the TryLicense operation can be used in order to inform about the conditions of a license conclusion before actually ordering this license. Omitting the TryLicense operation and directly invoking the OrderLicense operation would result in a blind order.

9.2 TryLicense Operation Request

9.2.1 TryLicense Operation Request Parameters

A request to perform the TryLicense operation SHALL include the use of the data structure specified in Table 27 and Table 28.

Table 27 — Parameters in TryLicense operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is "LBS"	One (mandatory)
version	Standard version for operation	Character String type, not empty Value is specified by each Implementation Standard and Schemas version	One (mandatory)
language	Language identifyer	Character String type, RFC4646 language code of the human readable text. Must be a language listed in the Capabilities Languages element.	Zero or one (optional)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

Table 28 — Sections in GetLicenseModel operation request

Name	Definition	Multiplicity and use

Name	Definition	Multiplicity and use
LicenseConfiguration	Actual configuration of a license model, see section 9.2.1.1	One (mandatory)

NOTE 2 The data type of many parameters is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

All the "optional" parameters and data structures, in the TryLicense operation request, should be implemented by all LB-Service clients using specified values, for each implemented LB-Server to which that parameter or data structure applies. Similarly, all the "optional" parameters and data structures SHALL be implemented by all LB-Servers, for each implemented LB-Service to which that parameter or data structure applies.

9.2.1.1 LicenseConfigurationType Contents

This section includes a complete license configuration. The expected parameters are defined in Table 29, the sub-sections in Table 30.

Table 29 — Parameters of the LicenseConfigurationType

Names	Definition	Data type and values	Multiplicity and use
id	License model identifier	Character string type, not empty	One (mandatory)

Table 30 — Section name values and contents of the LicenseConfigurationType

Section Name	Contents	Data Type	Multiplicity and use
Resource	Resource configuration	ResourceType, see section 9.2.1.2	One or more (mandatory)
LicenseAttribute	Arbitrary attributes attached to a license	Character string type, not empty	Zero or more (optional)
Licensor	Identity and contact details of license issuer	LicensorType, see section 8.3.1.7	One (mandatory)
Licensee	Identity and contact details of licensee	LicenseeType, see section 9.2.1.7	One (mandatory)

The resource elements include the actual license configuration that has to be submitted. The LicenseAttribute and Licensor elements have to be identical to the GetLicenseModel response of the actual license model and are interpreted as a confirmation of the licensee. If they differ from the license model, this SHALL result in an error. The Licensee element has to provide the licensee's identity and contact details.

Table 31 — Parameters of the LicenseAttribute Element

Names	Definition	Data type and values	Multiplicity and use
name	Attribute name	Character string type, not empty	One (mandatory)
title	Attribute title, may be language specific	Character string type, not empty	Zero or one (optional)

9.2.1.2 ResourceType Contents

The resource section defines the actual configuration parameter settings for the resources to be licensed. The parameters of this section are listed in Table 32:

Table 32 — Parameters of the ResourceType

Names	Definition	Data type and values	Multiplicity and use
id	Resource identifier	Character string type, not empty	One (mandatory)
name	Resource name	Character string type, not empty	Zero or one (optional)
title	Resource title	Character string type, not empty	Zero or one (optional)

The mentioned parameter values have to be the same as defined in the GetLicenseModel response.

The Resource section contains sub-sections as listed in Table 33.

Table 33 — Section name values and contents of the ResourceType

Section Name	Contents	Data Type	Multiplicity and use
Abstract	Resource description	Character string type, not empty	Zero or one (optional)
ConfigParams	Configuration parameters	Config params type, see table	Zero or more (optional)
ResourceAttribute	Any resource-specific information	Character string type, not empty	Zero or more (optional)

The Abstract element and the ResourceAttribute elements have to contain the same values as submitted in the GetLicenseModel response for the corresponding license model ID. The ConfigParams section is further described in section 9.2.1.3.

9.2.1.3 ConfigParamsType Contents

This section includes all configuration parameter values defined in the license model.

Table 34 — Section name values and contents of the ConfigParamsType

Section Name	Contents	Data Type	Multiplicity and use
Parameter	Configuration parameter	Parameter type, see section	Zero or more (optional)

9.2.1.4 ParameterType Contents

The parameters listed in Table 35 have to be the same as submitted in the GetLicenseModel response.

Table 35 — Parameters of the ParameterType

Names	Definition	Data type and values	Multiplicity and use
name	Parameter name	Character string type, not empty	One (mandatory)
title	Parameter title (language specific)	Character string type, not empty	Zero or one (optional)
type	Parameter type	xs:AnySimpleType	One (mandatory)

The parameter type includes one of the elements described in Table 19:

Table 36 — Section name values and contents of the ParameterType

Section Name	Contents	Data Type	Multiplicity and use
Value	Parameter value	ValueType, see section 8.3.1.5	Either one (mandatory)
Select	Multi-select values	Select Type, see section 8.3.1.6	

9.2.1.5 ValueType Contents

A ValueType is a character string. In a TryLicense request this value SHALL not be empty. It defines the actual parameter settings for an actual license evaluation. It includes the following parameters:

Table 37 — Parameters of the ParameterType

Names	Definition	Data type and values	Multiplicity and use
title	Value title, may be language specific	Character string type, not empty	Zero or one (optional)
selected	Indicates if value is selected	xs:boolan	Zero or one (optional)

9.2.1.6 SelectType Contents

A select type contains one parameter as described in Table 24

Table 38 — Parameters of the SelectType

Names	Definition	Data type and values	Multiplicity and use
multiple	Defines if multiple values are allowed for this parameter	xs:boolean	Zero or one (optional), false if omitted

Furthermore, it contains the following element:

Table 39 — Section name values and contents of the ParameterType

Section Name	Contents	Data Type	Multiplicity and use
Value	Parameter value	ValueType, see section 8.3.1.5	One or more (mandatory)

9.2.1.7 LicenseeType Contents

The licensee element is used to identify the subject willing to conclude a license. The licensee type elements are described in Table 24.

Table 40 — Section name values and contents of the LicensorType

Section Name	Contents	Data Type	Multiplicity and use
DefaultAddress	Contact information of Licensor	Address type, see section 8.3.1.8	One (mandatory)
DeliveryAddress	Special address	Address type, see section 8.3.1.8	Zero or one (optional)
Subject	SAML subject of the licensee	SAML SubjectType, see [1], clause 2.4.1	One (mandatory)

9.3 TryLicense Operation Response

9.3.1 Normal response parameters

The normal response to a valid TryLicense operation request SHALL contain one LicenseConfirmation section. This LicenseConformation section is an extended LicenseConfigurationType (see section 9.2.1.1) that repeats the submitted license configuration of the TryLicense request and adds zero or more optional condition elements as an extension. It also may include an optional language parameter, indicating the language used within this license configuration.

Table 41 shows the parameter of the TryLicense response, the response elements are defined in Table 42.

Table 41 — Parameters of the TryLicense response

Names [a]	Definition	Data type and values	Multiplicity and use
language	Language used within this license configuration	Character string type, not empty	Zero or one (optional)

Table 42 — Section name values and contents of the TryLicense operation response

Section Name	Contents	Data Type	Multiplicity and use
LicenseConfirmation	Confirmation of validity of license configuration,	LicenseConfigurationType (see section 9.2.1.1), extended by Condition element (see section 9.3.1.1)	One (mandatory)

9.3.1.1 Condition Element

The condition element is used to define conditions that will be applied when ordering a license with the actual configuration (for instance a price).

Table 43 — Section Condition element of the TryLicense operation response

Section Name	Contents	Data Type	Multiplicity and use
Condition	Any condition which is attached to the actual license configuration	Character string type, not empty	Zero or more (optional)

Table 44 — Parameters of the Conditions Element

Name	Definition	Data type and values	Multiplicity and use
name	Name of the condition	Character string type, not empty	One (mandatory)
title	Title of the condition, may be language-specific	Character string type, not empty	Zero or one (optional)

9.4 TryLicense Exceptions

When a LB-Server encounters an error while performing a TryLicense operation, it SHALL return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes SHALL include those listed in Table 10. For

each listed exceptionCode, the contents of the "locator" parameter value SHALL be as specified in the right column of Table 45.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 45 — Exception codes for GetLicenseModel operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
OptionNotSupported	Request is for an option that is not supported by this server	Identifier of option not supported
LanguageNotSupported	Request is for a language that is not supported by this server	Codes of supported languages
InconsistentLicenseModel	License elements are not consistent to the license model	List of inconsistent parameters
InvalidSubject	The submitted subject (licensee or licensor) is invalid	Reason for invalidity

10 OrderLicense Operation (Mandatory)

10.1 Introduction

The OrderLicense operation uses the same request as the TryLicense operation with the difference that the OrderLicense operation leads to a legally binding license conclusion.

10.2 OrderLicense Operation Request

Since the OrderLicense operation request is identical to the TryLicense operation request, see section 9.2 for further information.

10.3 OrderLicense Operation Response

The OrderLicense response consists of a LicenseReferenceType, which is a restricted SAML 2.0 assertion, representing a reference to the actual license that was concluded as result of the OrderLicense request.

The parameters of this assertion are defined in SAML [1], the content of the allowed SAML elements is defined in Table 46.

Table 46 — Elements of the LicenseReferenceType

Name	Definition	Data type and values	Multiplicity and use
Issuer	Identity of the license reference issuer	saml:NameIDType, see [1], clause 2.2.2	One (mandatory)
Signature	Signature for this license reference	ds:SignatureType, see [2], clause 4.1	Zero or one (optional)
Subject	Identity of the licensee	saml:SubjectType, see [1], clause 2.4.1	Zero or one (optional)
AttributeStatement	Statements for license retrieval	saml:AttributeStatementType, see section 10.3.1	One (mandatory)

10.3.1 AttributeStatementType Contents

The AttributeStatement has to provide at least two attributes, the URL of the License Manager Service that maintains the license, and the ID of the actual license, both encoded as SAML attributes (see [1], clause 2.7.3.1). The content of those attributes supported by a LBS service are defined in Table 47.

Table 47 — Elements of the LicenseReferenceType

Name Attribute [a]	Definition	Data type and values [b]	Multiplicity and use
LicenseManagerURI	URI of the License Manager	Character string, not empty	One (mandatory)
LicenseID	ID of the created license	Character string, not empty	One (mandatory)

[a] These values are the contents of the name parameter of the SAML attribute

[b] Although the data type is defined as xs:AnyType by SAML, a LBS server SHALL submit non-empty character strings

10.3.2 OrderLicense Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2007 sp2
(http://www.altova.com)-->
<lbs:LBSOrderLicenseResponse ID="ID_1" Version="0.5.0"
IssueInstant="2008-05-15T09:30:47.0Z"
xsi:schemaLocation="http://www.conterra.de/lbs
LicenseBroker_0_5.xsd" xmlns:lbs="http://www.conterra.de/lbs"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
   <saml:Issuer>conterra</saml:Issuer>
   <saml:Subject>
<saml:NameID>customer</saml:NameID> </saml:Subject>
   <saml:AttributeStatement>
      <saml:Attribute Name="LicenseManagerURI">
```



```
      <saml:AttributeValue>www.conterra.de</saml:AttributeValue>
         </saml:Attribute>
         <saml:Attribute Name="LicenseID">
            <saml:AttributeValue>916743582</saml:AttributeValue>
         </saml:Attribute>
      </saml:AttributeStatement>
</lbs:LBSOrderLicenseResponse>
```

10.3.3 OrderLicense Exceptions

When a LB-Server encounters an error while performing an OrderLicense operation, it SHALL return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes SHALL include those listed in Table 10. For each listed exceptionCode, the contents of the "locator" parameter value SHALL be as specified in the right column of Table 45.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 48 — Exception codes for GetLicenseModel operation

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
OptionNotSupported	Request is for an option that is not supported by this server	Identifier of option not supported
LanguageNotSupported	Request is for a language that is not supported by this server	Codes of supported languages
InconsistentLicenseModel	License elements are not consistent to the license model	List of inconsistent parameters
InvalidSubject	The submitted subject (licensee or licensor) is invalid	Reason for invalidity
LicenseNotProduceable	The license production did not succeed	Reason

Annex A
(normative)

XML Schema Documents

In addition to this document, this standard includes several normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. After OGC acceptance of a Version 1.0.0 of this standard, these XML Schema Documents will also be posted online at the URL http://schemas.opengeospatial.net/TBD/1.0.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files SHALL be considered authoritative.

The LBS abilities now specified in this document use two specified XML Schema Documents included in the zip file with this document. These XML Schema Documents combine the XML schema fragments listed in various subclauses of this document, eliminating duplications. These XML Schema Documents are named:

LicenseBroker_0_5.xsd

License.xsd

These XML Schema Documents use and build on the OWS common XML Schema Documents specified [OGC 05-008], named:

ows19115subset.xsd

owsCommon.xsd

owsDataIdentification.xsd

owsExceptionReport.xsd

owsGetCapabilities.xsd

owsOperationsMetadata.xsd

owsServiceIdentification.xsd

owsServiceProvider.xsd

All these XML Schema Documents contain documentation of the meaning of each element and attribute, and this documentation SHALL be considered normative as specified in Subclause 11.6.3 of [OGC 05-008].

Bibliography

[1] OASIS, *Security Assertion Markup Language (SAML) v2.0,* http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip

[2] W3C, *XML-Signature Syntax and Processing,* W3C Recommendation 12 February 2002, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

