

Open Geospatial Consortium Inc.

Date: 2007-04-17

Reference number of this OGC® document: OGC 06-184

Version: 0.9.2

Category: OGC® Discussion Paper

Editor: Christian Elfers, Roland M. Wagner

GeoDRM Engineering Viewpoint and supporting Architecture (OWS-4 GeoDRM Interoperability Report)

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

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: Interoperability Report
Document subtype: Discussion Paper
Document stage:
Document language: English

Contents		Page
1	Scope.....	1
2	Compliance	2
3	References.....	2
4	Terms and definitions	3
4.1	License.....	3
4.2	Assertion.....	3
4.3	License Reference Token	3
4.4	Identity Token	3
4.5	General Broker/Manager (informative).....	3
4.6	Explicit contracting (informative).....	3
4.7	Implicit contracting (informative).....	3
4.8	Specialized Broker/Manager (informative).....	3
4.9	Operation Model (informative)	3
5	Conventions	4
5.1	RM-ODP Viewpoints	4
6	Use Cases	5
6.1	Use Case #1: Unrestricted Use License	5
6.2	Use Case #2: Distributor License	6
6.3	Use Case #3: End User License	7
6.4	Use Case #4: WFS-T Feature Updater	8
7	GeoDRM Architecture.....	10
7.1	Architecture model	11
7.2	GeoDRM Components (Services and Applications).....	12
7.2.1	Gatekeeper Service	14
7.2.2	Authentication Service.....	15
7.2.3	Authorization Service	15
7.2.4	License Manager Service.....	16
7.2.4.1	Interface description.....	17
7.2.4.2	Interface security.....	18
7.2.4.3	Signing 18	
7.2.4.4	Creation of a License Reference	18
7.2.4.5	Capabilities 19	
7.2.5	License Broker Service	19
7.2.5.1	Interface description.....	20
7.2.5.2	Interface security.....	21
7.2.5.3	Capabilities 21	
7.2.6	GeoDRM-enabled Client	22
7.2.6.1	Implementation choices	24
7.2.6.2	GeoDRM Client Implementations	24

8	Information model	31
8.1	Identities and Identity Tokens	31
8.1.1	Identity token encoding.....	31
8.1.1.1	Username / Password.....	32
8.1.1.2	Kerberos 32	
8.1.1.3	PKI / X509 Certificates.....	32
8.1.1.4	SAML 33	
8.2	Licenses and License Reference Token	35
8.2.1	Licenses.....	36
8.2.2	License encoding	38
8.2.3	License Reference Tokens	39
8.2.4	License Reference Token Encoding	40
8.3	Extensions to Capabilities – GeoDRM Preconditions	42
8.3.1	Type of a Precondition.....	42
8.3.2	Issuing authority.....	43
8.3.3	Subject of a precondition	43
8.3.4	Precondition Encoding.....	44
8.3.5	Example for preconditions - Combined identity and license precondition	47
9	Technical Workflows.....	48
9.1	Information and Interpretation of Preconditions	48
9.2	Authentication	50
9.3	License (issuing/negotiation, resolving).....	51
9.3.1	License issuing/negotiation.....	52
9.3.2	License resolving	54
9.4	Gatekeeper-Client Communication	55
9.5	Authorization.....	57
10	Distribution and Deployment model.....	59
10.1	GeoDRM-enabled Infrastructure.....	59
10.2	Deployment Scenarios.....	60
10.2.1	Single-Operator Scenario.....	61
10.2.2	Single Sharing: Access control only scenario.....	62
10.2.3	Single Sharing: Anonymous License Click-through scenario	63
10.2.4	Federation scenario	64
10.2.5	Distributor scenario.....	65
11	Demo Application Scenarios	66
11.1	Application Scenario “Feature Updates”	66
11.1.1	Relation to Use Cases	66
11.1.2	Application Context.....	66
11.1.3	Deployment and Configuration.....	66
11.1.3.1	Licensed Rights.....	67
11.1.4	Walk-Through.....	70
11.2	Application Scenario “Breaking the glass”	75

11.2.1	Relation to Use Cases from the RFQ.....	75
11.2.2	Application Context.....	75
11.2.3	Deployment and Configuration.....	76
11.2.3.1	Gatekeeper Preconditions	77
11.2.3.2	License Offer and Policy-Template	78
11.2.4	Walk-Through.....	81
11.2.4.1	Find and Get Capabilities from GeoDRM protected service.....	81
11.2.4.2	Authentication.....	81
11.2.4.3	License negotiation.....	82
11.2.4.4	Usage of the GeoDRM protected service	85
11.2.5	Extension opportunities	85
12	Informative: Enterprise Business Roles and Processes	86
12.1	GeoDRM Roles	86
12.2	Business Processes	88
12.3	Publish Phase.....	91
12.4	Find Phase	91
12.5	Procurement Phase: Establishment Process	91
12.6	Procurement Phase: Management Process	92
12.7	Delivery (Bind) Phase	93
12.8	Chained Business Phases.....	93
13	Future Work.....	95
14	References.....	97

i. Preface

This document was developed by the OWS-4 GeoDRM Thread Group as part of the OGC Interoperability Program OWS-4 initiative. The OWS4 initiative was started in June 2006 and finalized with a demonstration in early December. The results were presented at the OGC San Diego TC meeting in the GeoDRM WG, Security WG and in the Architecture WG in mid December.

Suggested additions, changes, and comments on this report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

ii. Submitting organizations

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

- con terra GmbH, Münster, Germany
- ESRI, Inc., Redlands, CA, USA
- Fraunhofer ISST, Dortmund, Germany
- Traverse Technologies Inc., Cambridge, MA, USA
- Universität der Bundeswehr (UniBW), München, Germany

iii. Document contributor contact points

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

Name	Organization
Jan Drewnak (JDR)	con terra
Christian Elfers (CEL)	con terra (Editor)
Roland M. Wagner (RMW)	con terra (Editor)
Cristian Opincaru	University of the German Armed Forces (UniBW)

iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
2006-12-02	0.0.1	Elfers	All	Initialized based on outline v0.9
2006-12-18	0.0.2	Wagner	7, 11.2	Initial section content.
2006-12-22	0.0.5	Elfers	All	Major changes to all chapters beside 7
2007-01-03	0.0.6	Elfers	8-10, 11, 12	Initial section content.
2007-01-16	0.0.7	Elfers	9, 10	Added workflows, identity token chapters, minor changes to other chapters
2007-01-23	0.0.8	Elfers	7, 8, 9, 11, 13	Moved Enterprise Viewpoint related content to 13 (former 7) Added contributions of ESRI Inc, ISST, Uni BW to chapters 7, 8, 9 & 11
2007-02-05	0.0.9	Wagner/ Elfers	All	Layout and final corrections
2007-02-09	1.0.0	Wagner/ Elfers		Release
2007-02-12	1.0.1	Elfers	8.3.5, comments	Changed the copyright to 2007, fixed a figure problem in chapter 8.3.5
2007-04-17	1.0.2	Elfers	Introduction	Editorial changes

v. Changes to the OGC Abstract Specification

This paper is intended to feed a discussion, therefore no change requests are scheduled currently.

A change request for the GeoDRM Reference Model (RM) (#06-004r4) is possible, because this report adds the GeoDRM Engineering Viewpoint and architectural issues. Currently the GeoDRM RM is in a voting process.

vi. Future work

The future work is described in detail in chapter 13 – Future Work.

Foreword

After the foundation of the geospatial digital rights management working group (GeoDRM WG) in summer 2004, the creation of the abstract reference model and fundamental work in the first OWS-3 GeoDRM initiative, the OWS-4 GeoDRM activity focused on the engineering aspects of an overall architecture for GeoDRM and refined the GeoDRM business phases (informative) to bridge the gap between abstract reference model and implementation.

OWS-4 GeoDRM added also a proof-of-concept for the architecture was provided with two demonstrators that are aligned with the abstract reference model and the use cases defined in the OWS-4 RFQ.

Although many issues are still subject for discussions and agreements, OWS-4 GeoDRM was a clear milestone in the development of geospatial digital rights management systems.

This document was developed during the OWS4 initiative of the OGC. It was contributed by the organizations involved in the GeoDRM thread of this initiative. The document is intended as a discussion paper. It does neither cancel nor replace other OGC documents in whole or in part. Two other documents are also planned as result of OWS-4 GeoDRM: Trusted Geo Services IPR (OGC#06-107) and the OWS Common Change Proposal (OGC#06-177). The intention of this document is to complement the other two documents.

Introduction

The Open Web Service Initiative 4, Thread GeoDRM, aimed to develop an engineering viewpoint to amend the GeoDRM Reference Model. A distribution model is needed, because the management of intellectual property rights (IPR) involves many, often legally and technically independent organizations. **Error! Reference source not found.** shows eight related roles, which are used to transfer usage rights from an IPR owner to an end-user. The rights are transferred via contracting between owner, provider and customer.



Figure 1 - Chain of roles for IPR management

Interoperable interfaces are needed to allow a chaining of different functionalities between independent organizations to enhance IPR management efficiency due to electronic support. Therefore a good understanding of roles, business processes and organizational interfaces is fundamental to derive electronic interfaces. A key element is this transposition of licenses, which are results of a contract. Figure 2 depicts the lifecycle of a license, suitable components and operating roles.

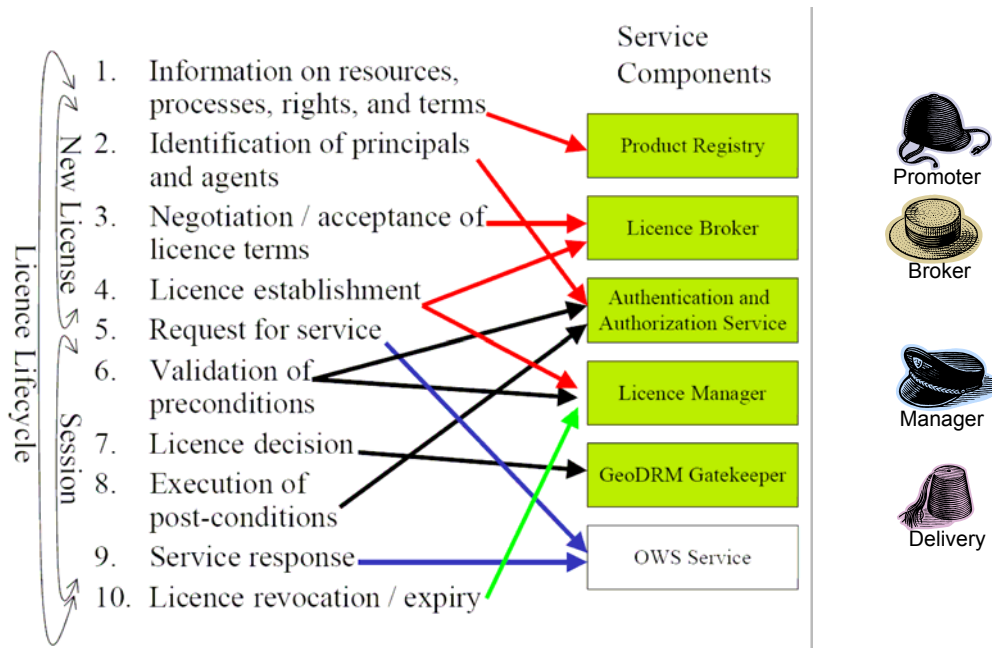


Figure 2– License lifecycle and components

Because of the on-going development of spatial data infrastructures (SDI) and already deployed SDI instances, an embedding-without-touching integration method is required to integrate the new business functions (user identification and licensing). Another reason is the wide range of different applied operation models, e.g. identification only, licensing only or identification and licensing. This requires a flexible framework (GeoDRM).

This document delivers a report about the identified processes, information models and software components. Chapter 5 introduces a set of convention, which are used in this document. Chapter 1 shows the defined use cases for this initiative. Chapter 7 gives a technical architecture and describes the components in detail. Chapter 7 defines used information models. Chapter 9 relates information models and components via workflows. Chapter 9 shows two extreme deployments to illustrate the wide range of potential constellations. Chapter 11 depicts the described models with screenshots from the developed OWS-4 GeoDRM demonstrator. Chapter 12 identifies and refines business processes and relates them to roles (informative).

Although many issues remain still subject for discussion, the concept of a GeoDRM component “Gatekeeper Service” has proven to be very stable. This component is placed in front of an OWS services and receives and checks all requests. This component offers also an enhanced capability document with GeoDRM related descriptions. It is important to note, that meaning and definition of the “OWS-4 Gatekeeper” differs from the gatekeeper as defined in the Reference Model. The following table indicates the corresponding modules of the OWS-4 and the GeoDRM Reference Model architecture:

<i>OWS-4 GeoDRM Architecture</i>	<i>GeoDRM Reference Model Architecture</i>
Gatekeeper	Security
Authorization Service*	Gatekeeper

* Note: the OWS-4 GeoDRM Authorization Service is intended to provide access control decision functionality on both: licenses and predefined rights. The Gatekeeper from the GeoDRM Reference Model Architecture is intended to provide access control decision functionality on licenses only.

Next to the gatekeeper services also the “License Manager Service” component interfaces were defined in detail. The “License Broker Service” component was only touched conceptual (although a simple License Broker Application was developed). The components “Authentication and Authorization Service” were developed and defined as separate service components. The interface of the Authorization Service is currently not a subject for standardization because of it’s internal usage. The component “Product Registry” is considered in general as a regular OGC catalog. More refined metadata definitions are needed for a full integrated workflow, which were not defined in detail in OWS-4.

OpenGIS[®] OWS-4 GeoDRM Engineering Viewpoint

1 Scope

After the introduction of the Web Mapping Service Specification in April 2000, important Geospatial Web and Spatial Data Infrastructure (SDI) components began to be developed. Today, in 2007, major parts of the SDI vision have become real. Implementation Specifications like the WMS, WFS, WCS, CS-W and GML can be used to build up global service-oriented and interoperable SDI's. Interoperable software products have been developed and deployed. The concept of a SDI based on OGC components has been proven with realized SDIs at many organizational levels worldwide.

From an economic point of view, SDI provides the communication and transport mechanism for trading/selling/distributing of spatial content. The development and implementation of business models for trading spatial data has already started. The OGC has not yet developed specifications for interoperable trading capabilities; there is a risk that inappropriate and/or proprietary solutions could limit the capacity of SDI for serving geospatial lines of business.

The OGC GeoDRM Reference Model (RM) has been developed as initial abstract specifications to address these needs. The RM defines a license model for trading geospatial content / services, lays out a number of use cases and GeoDRM workflow roles, and begins the task of specifying GeoDRM operations. The RM focuses on license structures and decisions, leaving more generic trading elements such as discovery, negotiation, authentication, and enforcement to external protocols and implementation specifications.

The GeoDRM thread in OWS-3 (which preceded development of the RM) investigated ways in which protocol bindings for OWS services (e.g. WMS, WFS) could be extended in a interoperable manner with existing technologies to accommodate a simple "click-through" trading scenario. The GeoDRM thread in OWS-4 investigated more involved licensing scenarios with the aim of

- Definition of preconditions for services Combination of identities and licenses to
- providing concepts to issue, transport and use of licenses to

This GeoDRM engineering viewpoint document describes use cases and concepts for GeoDRM, as well as references to distributed computing concepts which are not GeoDRM *sensu stricto* but are required for any GeoDRM implementation. The capabilities identified here describe the requirements to be met by the OWS computation

and information models. This document focuses on conceptual technical aspects to reflect the engineering viewpoint. Implementation details like encodings and service interface (binding) specifications are added selective for information purposes but as those belong to the technology viewpoint, they are in general out of scope for this document.

2 Compliance

This report does not have any compliances issues.

3 References

- OWS-4 GeoDRM IPR Trusted Geo Services, #06-107
- OWS-4 GeoDRM IPR: Common Change Proposal - GeoDRM enablement, #06-177
- GeoDRM Reference Model, #06-004r4
- OWS3: Access Control & Terms of Use (ToU) "Click-through" IPR Management, #05-111r2

4 Terms and definitions

4.1 License

Representation of grants that convey to principals the rights to use specified resources subject to specified conditions Source: [GeoDRM RM]

4.2 Assertion

An assertion is a statement made about a subject (similar to claim as used in [OWS4Trust])

4.3 License Reference Token

A license reference token represents a collection of assertions that express a reference to a license that a subject is in possession of.

4.4 Identity Token

An identity token represents a collection of assertions that describe a subject.

4.5 General Broker/Manager (informative)

A broker or a manager instance supporting all sharing functions. An example is a broker, supporting only (identity, price & order and license brokering).

4.6 Explicit contracting (informative)

Contracting with an distinct user interaction, e.g. clicking of a button

4.7 Implicit contracting (informative)

Contracting without an distinct user interaction, e.g. installation of a software or visit of a web page

4.8 Specialized Broker/Manager (informative)

A broker or a manager instance supporting only a subset of sharing functions. An example is a license broker, supporting only license brokering.

4.9 Operation Model (informative)

Applied model with an instantiated combination of sharing functions (user identification, licensing and pricing & ordering).

5 Conventions

The following elements are defined as conventions, because they are often used in the document and have a definition set character.

5.1 RM-ODP Viewpoints

RM-ODP defines the semantics of fundamental concepts and constructs of information management used for specification of any system (computer-based or otherwise) independently of a specific methodology, technology, or tool(set). In this manner, all stakeholders of an information management project could use the same explicitly defined system of concepts, thus providing for traceability between and maintainability of business, IT system, and technology specifications (source: Wikipedia).

The RM-ODP identifies five viewpoints for understanding a system.

- Enterprise Viewpoint
- Information Viewpoint
- Computational Viewpoint
- Engineering Viewpoint
- Technology Viewpoint

Although not to exhaust the definitions, a simple orientation about the viewpoints is given here. The enterprise viewpoint describes business processes and relationships. The information viewpoint describes information models. The computational viewpoint focuses on interface descriptions. The engineering viewpoint describes the distribution of elements. The technology viewpoint takes concrete technologies, e.g. SOAP, HTTP into account.

6 Use Cases

The following use cases are taken to ground the OWS-4 GeoDRM development. These cases give an impression about the wide range of different applied operation models. Use case #1 excises a simple click-through contracting. Use #2 shows a limited value chain with three independent players. Use case #3 aims to define a more professional transaction. Use case #4 focuses on a read/write situation.

The use cases are taken from the RFQ and are refined due to the OWS-4 GeoDRM results.

6.1 Use Case #1: Unrestricted Use License

Use Case Description: This use case describes “unrestricted” access to map layer resources based on a session license in which the user has read a statement of terms-of-use and agreed to them with a click-through gesture.	
Actors (Initiators): User of WMS	Actors (Receivers) Same as initiator
Pre-Conditions: <ul style="list-style-type: none"> - User requires WMS map layers. - User has access to WMS client. - User is able to discover WMS services with the needed layers through a CS/W catalog document 	Post-Conditions: <p>WMS map layers are viewable within the user’s WMS client software.</p>
System Components (may be combined) <ul style="list-style-type: none"> - GeoDRM-enabled CS/W: Catalog Service Web Profile - GeoDRM-enabled WMS: Web Map Service - GeoDRM-enabled Web WMS - GeoDRM-enabled Desktop WMS - (License) Broker: presents license offers and establishes licenses - (License) Manager: stores and matches licenses - GeoDRM Gatekeeper: decides whether a specific request is valid under a specific license - Authentication & Authorization: “security” implements authentication of license decision elements and authorization of consequences 	
Basic Course of Action: <ol style="list-style-type: none"> 1. Client queries a CS-W and/or WMS to determine if needed map layers are available and under what terms 	

2. User selects layers of interest
3. GeoDRM Client obtains terms of use
4. User agrees to terms presented by GeoDRM Client
5. Client returns license acknowledgement to Broker Server
6. Broker Server stores established license with session identity and returns acknowledgement token
7. WMS/GeoDRM Client issues map layer request with license acknowledgement token to WMS
8. Gatekeeper Server validates identity of user and authenticity of license information, decides that license applies to request.
9. WMS returns map layer to client
10. (Alternate unrestricted use distribution) WMS Server seen by the client is cascading both the map layers and license offer / acknowledgement from one or more other servers

6.2 Use Case #2: Distributor License

<p>Use Case Description: This use case describes “distributor” rights to WMS map layers. The provider of a cascading WMS operates under a license with an originating WMS to re-distribute on its own one or more map layers to clients under an unrestricted use license.</p>	
<p>Actors (Initiators): User of WMS and provider of cWMS</p>	<p>Actors (Receivers) Same as initiators</p>
<p>Pre-Conditions:</p> <ul style="list-style-type: none"> - User requires WMS map layers. - User has access to WMS client. - cWMS provider is able to cascade map layers from one or more originating WMS Servers 	<p>Post-Conditions:</p> <p>WMS map layers are viewable within the user’s WMS client software.</p>
<p>System Components</p> <ul style="list-style-type: none"> - CS/W: Catalog Service Web Profile - WMS: Web Map Service - cWMS: Cascading Web Map Service - License Broker: presents license offers and establishes licenses - License Manager: stores and matches licenses - License Gatekeeper: decides whether a specific request is valid under a specific license - License Enforcer: “security” implements authentication of license decision elements 	

and authorization of consequences
<p>Basic Course of Action:</p> <ol style="list-style-type: none"> 1. cWMS provider establishes a distributor license with an originating WMS for one or more map layers and receives a license acknowledgement token. 2. User queries a CS/W and/or the cWMS to determine if needed map layers are available and under what terms 3. User selects layers of interest 4. GeoDRM Client obtains terms of use 5. User agrees to terms 6. Broker Server stores established license and returns acknowledgement token 7. WMS/GeoDRM Client issues map layer request to cWMS with license acknowledgement token. 8. Gatekeeper Server validates identity of user and authenticity of license information, decides whether license applies to request. 9. cWMS issues map layer request to originating WMS with its own (distribution license) acknowledgement token 10. WMS returns map layer to cWMS 11. cWMS returns map layer(s) to client.

6.3 Use Case #3: End User License

<p>Use Case Description: This use case describes “end user” rights to WMS map layers and/or WFS feature collections for specifically identified individual users. The end user rights may be individual or may be based on an individual’s role (e.g. membership) in a licensed organization. The end user license may carry specific pre-conditions and constraints which need to be satisfied before a request can be honored.</p>	
Actors (Initiators): User of WMS	Actors (Receivers) Same as initiator
<p>Pre-Conditions:</p> <ul style="list-style-type: none"> - User requires WMS map layers. - User has access to WMS client. 	<p>Post-Conditions:</p> <p>WMS map layers are viewable within the user’s WMS client software.</p>
<p>System Components</p> <ul style="list-style-type: none"> - CS/W: Catalog Service Web Profile - WMS: Web Map Service - License Broker: presents license offers and establishes licenses 	

<ul style="list-style-type: none"> - License Manager: stores and matches licenses - License Gatekeeper: decides whether a specific request is valid under a specific license - Authentication & Authorization: “security” implements authentication of license decision elements and authorization of consequences
<p>Basic Course of Action:</p> <ol style="list-style-type: none"> 1. User queries a CS/W and/or WMS/WFS and/or WMC to determine if needed map layers or datasets are available and under what terms 2. User selects layers and/or datasets of interest 3. User logs in and is authenticated with a specific identity (e.g. username/password) 4. Server matches identity with established individual or organization license and returns acknowledgement token 5. Client issues map layer or dataset request with license acknowledgement token 6. Server validates identity of user and authenticity of license information, decides whether license applies to request and whether any pre-conditions and constraints are met (e.g. time of request, area of request, state of daily usage quotas) 7. Server returns map layer or dataset to client

6.4 Use Case #4: WFS-T Feature Updater

Use Case Identifier: GeoDRM #4	Use Case Name: WFS-T Feature Updater
Use Case Domain: OWS-4 GeoDRM Feature Update	Status: Final 04/11/06
Use Case Description: This use case describes “Updater” rights to provide specific feature update transactions to a WFS-T server.	
Actors (Initiators): Remote editor / updater of feature collection	Actors (Receivers): Analyst reviewing, managing, and utilizing feature collection
<p>Pre-Conditions:</p> <ul style="list-style-type: none"> - Feature collection is configured through a WFS-T - Updater has new/changed features to transact - Updater has a WFS-T client - Updater has an established update license - Analyst has a WFS-T client, authenticated session with WFS-T server, license to review/approve 	<p>Post-Conditions:</p> <p>New/updated features are available for query from the WFS-T.</p>

feature updates, and license to query / use feature collection	
<p>System Components</p> <ul style="list-style-type: none"> - WFS-T: GeoDRM-enabled Web Feature Service Transactional - WNS: Web Notification Service - License Manager: License Information Point - GeoDRM Gatekeeper: License Decision Point - WA2S: Authentication and Authorization Service - 	
<p>Basic Course of Action:</p> <ol style="list-style-type: none"> 1. User #1 (feature updater) prepares new/updated features for transaction 2. User logs in at client and client establishes authenticated session with WFS-T / WFS-T access enforcement endpoint (client, server, and user are authenticated by WA2S to establish chain of trust) 3. User initiates a pending feature transaction against WFS-T 4. WFS-T access enforcement point requests authorization of transaction from the WA2S 5. WA2S determines that the update request requires a license decision, WFS-T retrieves license from License Manager corresponding to requested usage and requests validation from GeoDRM Gatekeeper. 6. GeoDRM Gatekeeper validates the transaction as a licensed usage and returns effect conditions, WA2S then authorizes the transaction with a constraint (notification and audit) 7. WFS-T performs (pending) transaction, responds to user with transaction id 8. WFS-T registers a transaction notification with WNS and a licensed usage with the License Manager according to license effect conditions. 9. User #2 (analyst) is notified by WNS of a pending transaction. 10. User #2 retrieves usage record to verify licensed action and licensee. 11. User # 2 retrieves and reviews features in the pending transaction. 12. User # 2 approves and updates the status of the pending transaction. 13. New / updated features are available for use by other licensed users 	

7 GeoDRM Architecture

Geospatial Digital Rights Management (GeoDRM) is defined in the OGC GeoDRM Reference Model as the packaging, distributing, controlling and tracking of geospatial content based on rights and licensing information. More generally it can be taken to cover a broad spectrum of capabilities and underlying technologies supporting description, identification, trading, protecting monitoring and tracking of all forms of rights usages for both tangible and intangible (electronic) assets, including the management of rights-holders relationships.

For the purpose of the OWS-4 initiative, GeoDRM consists of standards, technologies, and practices which enable interoperable trading of geospatial content to be implemented on top of OWS services. Needed GeoDRM functionalities in that sense are controlling of access to OWS services, using identities and licenses as access granting “keys” and an interoperable ways to request, issue and transport them in the network of interacting services. OWS-4 GeoDRM did not focus on usage protection after the content was delivered to a client.

A proper OWS-4 GeoDRM Architecture therefore needs take access control with authentication, licensing and authorization into account. It serves as an “umbrella architecture” in which specialized topics of licensing (non commercial/terms-of-use and commercial /pricing and ordering) and security (identity handling, trust, encryption, etc.) fit seamlessly into.

The architecture was (partly) proofed by accompanying implementations and demonstrators that were in scope of the OWS-4 GeoDRM initiative.

7.1 Architecture model

The architecture is based on the XACML authorization model ([XACML]) and uses its stereotypes. The following figure provides an overview of the architecture.

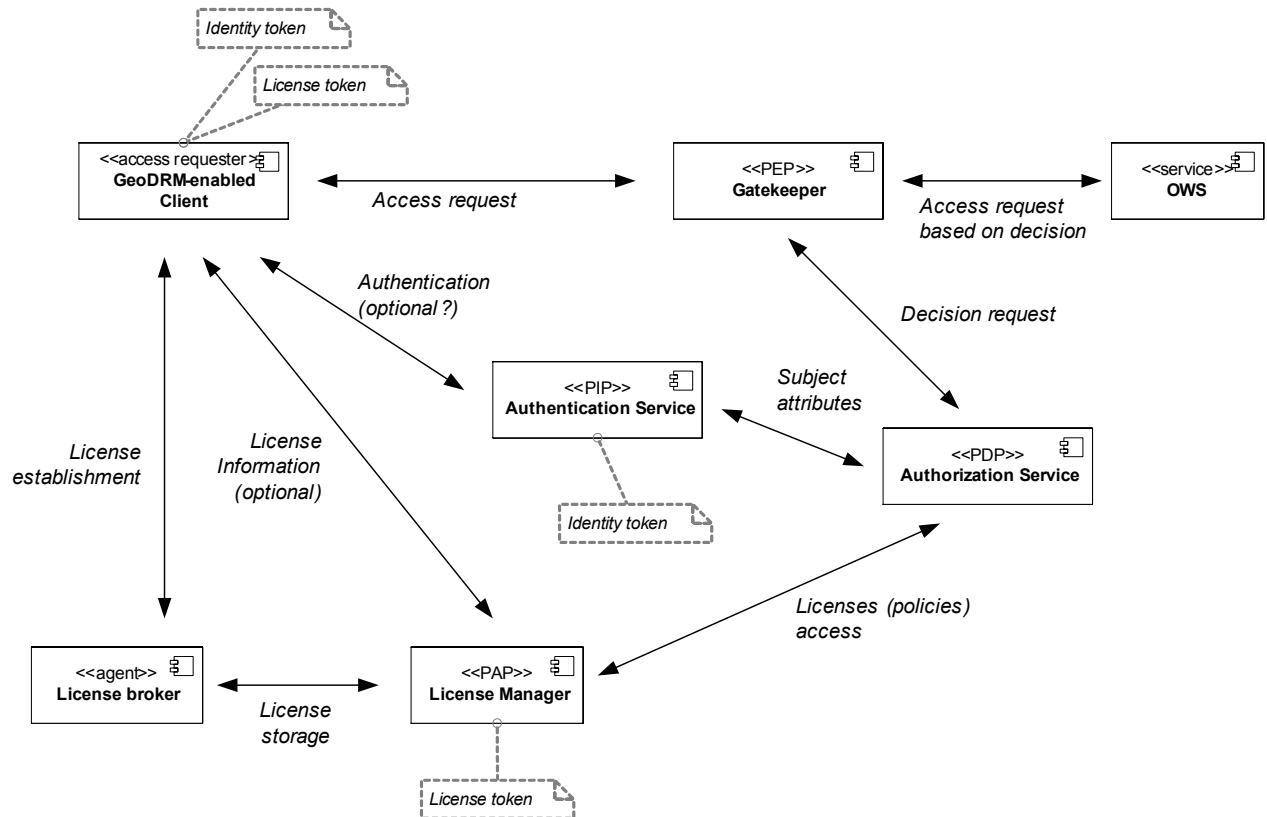


Figure 3 - GeoDRM Architecture based on XACML authorization model

According to the model the components realize the following stereotypes:

- **PEP (Policy Enforcement Point):** a component providing enforcement functionalities based on an authorization decision made by a PDP. In the OWS-4 GeoDRM architecture the Gatekeeper component owns this stereotype as the main purpose of the gatekeeper is to control access to OWS services and enforce policies.
- **PAP (Policy Administration Point):** A component that stores and maintains policies. In the OWS-4 GeoDRM architecture the License Manager Service owns this stereotype. The Manager is responsible for maintaining licenses which include policies that apply to a set of resources, actions and subjects and that are used as base input for the authorization and therefore enforcement.

- PDP (Policy Decision Point): A component that evaluates an authorisation request issued e.g. by a PEP against policies found in a PAP. In the OWS-4 GeoDRM architecture this is the task of the authorization service. By applying that model, the Authorization Service could use any types of PAP implementations as base for the access control decisions. This includes classical predefined policy stores like a database as well as a License Manager PAP or even both.
- PIP (Policy Information Point): A component providing external policy context and attributes to the PDP. In the OWS-4 GeoDRM architecture this could be the task of the authentication service, providing additional information about a particular user that is needed to perform the PDP decision request. There may be other PIP implementations used as well in order to provide various kinds of information needed to make a decision.

By using a common model and stereotypes for the architecture, even parts of it can be reused in completely different environments that deal maybe only with classical, identity based access control or that incorporate with other components that follow the same model. A few examples about derived deployment scenarios are given in chapter 10.2.

7.2 GeoDRM Components (Services and Applications)

The OWS-4 GeoDRM architecture is composed of the service and client components that were included in the RFQ for the OWS-4 GeoDRM initiative. The components focus on both general and specific aspects of a GeoDRM enabled system. The general aspects cover: Authentication and Authorization for OWS services whereas the specific aspects cover issuing and usage of licenses for OWS services.

The focus of the OWS-4 GeoDRM is limited in number and elaboration of the overall components that may be part of a GeoDRM system. Not covered are for example business components for commercial license and user identity establishment and management.

The following figure shows the components of the OWS-4 GeoDRM architecture and their interrelations as well as their interfaces and dependencies.

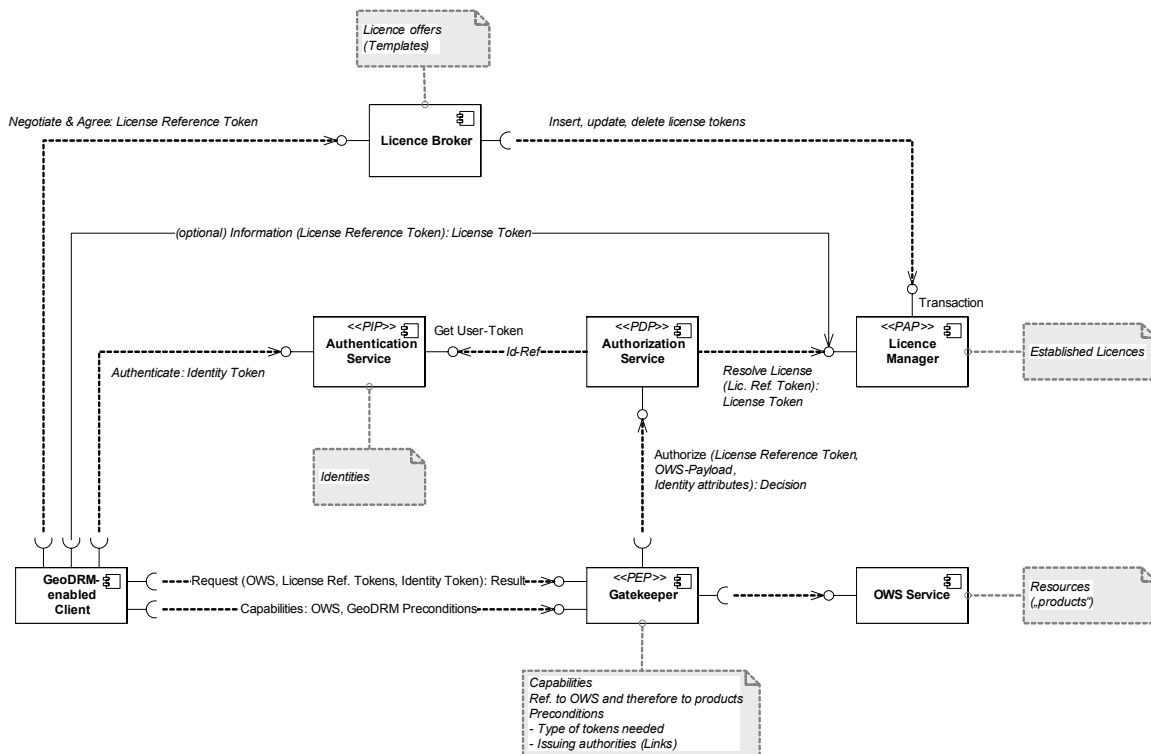


Figure 4 - Components overview and relations

In brief the components are distinguished as follows:

- The Gatekeeper controls access to OWS service (single point of control) and is responsible to provide the client with all necessary information needed to achieve access (extends OWS capabilities with GeoDRM preconditions; single point of information for the client).
- The License Broker negotiates and issues licenses with the client. It can provide different offers for those licenses; each could have a different workflow for negotiation/contracting: terms-of-use, payment, etc. Licenses are handed as “references” to the client.
- The License Manager manages licenses in a GeoDRM enabled system. Successfully negotiated licenses are stored here and can be viewed any time from the customer, the provider and the delivery. It provides those licenses for authorization, information and extension/modification purposes.
- The Authentication service does the authentication and issues identities for the client.

- The GeoDRM enabled Client is able to communicate to the OWS via the Gatekeeper (note: definition an figure assume an exisiting OWS Client with extended GeoDRM functionalities, see chapter 7.2.6 for additional information).

A more detailed description is provided in the following chapters.

7.2.1 Gatekeeper Service

The GeoDRM Gatekeeper operates as an additional functionality to existing OWS services (either independent component (proxy) or integrated into the OWS). It has covers access control (“protected service”) and intercepts all incoming messages (i.e. all messages pointed to the protected service). The SOAP message protocol with a HTTP binding was used to enclose and transport the well-known OWS POST/GET HTTP items.

The Gatekeeper has no special interface but the responsibility to process information associated with requests, which can be SOAP header information [WS-S] and the request itself, e.g. a SOAP body portion. It enforces authorization decisions taken by an attached PDP and takes care of possible obligations bound to the authorization decision.

The gatekeeper offers GeoDRM extended Capabilities, which adds additional GeoDRM specific information elements to the OWS Capabilities (see GeoDRM preconditions).

Based on the decision of the authorization service, the gatekeeper is in charge of passing or blocking a request to the OWS. Therefore, the Gatekeeper takes a central role in the GeoDRM architecture and serves the role of a policy enforcement point (PEP).

Interface
<ul style="list-style-type: none"> - GetCapabilities <ul style="list-style-type: none"> ▪ Description: GetCapabilites is defined as an unprotected starting operation ▪ Parameters: OWS specific ▪ Returns: Capabilities with precondition information , OWS Exception - Non-GetCapabilities <ul style="list-style-type: none"> ▪ Description: Each non-GetCapabilities request requires authentication and authorization ▪ Parameters: Identity token, License reference, OWS Request for protected service ▪ Returns: OWS Reponse/Exception from protected service

7.2.2 Authentication Service

The Authentication Service authenticates subjects by means of an arbitrary authentication method and issues an “authentication token” that represents asserted subject information that is stored at the Authentication Service. Spoken in SAML terms: The identity provider (Authentication Service) returns an assertion artifact (reference to assertion), that represents an authentication assertion.

The Interface should offer two main functionalities:

- Authentication of users and providing of references to identities (identity token)
- Options to access user information (identity) by providing the appropriate reference

Like the license broker, Authentication service hands out references to identities (identity tokens) for successfully authenticated users (principals). That reference could be used to communicate to a GeoDRM- (Gatekeeper-) protected OWS. For authorization purposes, the authorization service could request the needed identity information from authentication service (e.g. if needed by the license/policy decision rule set).

Interface
<ul style="list-style-type: none"> - GetSAMLResponse <ul style="list-style-type: none"> ▪ Description: Get an authentication SAML artifact ▪ Parameters: Credentials ▪ Returns: SAMLAssertion with AuthenticationStatement, Exception

7.2.3 Authorization Service

The Authorization Service decides, whether certain requests are permitted for a certain subject, where request consists of a certain action on certain resource (-type). Request and subject information are passed by the Gatekeeper component in a suitable way. To derive an authorization decision the Authorization Service requests the license from the License Manager. The license is referenced in the client request and is submitted to the Authorization Service by the Gatekeeper.

The interface includes functionalities to perform an authorization decision by taking resource, subject and action parameters as input. A preparation for the process of decision taking may include submitting the license reference and/or an identity reference as input.

Interface
<ul style="list-style-type: none"> - PrepareLicense <ul style="list-style-type: none"> ▪ Description: As the PDP does not store licenses it has to obtain corresponding rulesets. This is done before request evaluation. ▪ Parameters: LicenseReferenceToken ▪ Returns: Status message, Exception - EvaluateRequest <ul style="list-style-type: none"> ▪ Description: This is the actual decision taking ▪ Parameters: XACML Request consisting of Subject, Action and Resources ▪ Returns: XACML Response, containing decisions: NotApplicable, Deny, Permit, Indeterminate

7.2.4 License Manager Service

It is the task of the License Manager Service to manage licenses. This management includes discovery and transactional functionalities for licenses. It may be

- Permanent and persistent management (for permanent licenses) or transient management (for such licenses that have a time-based validity; e.g. session).
- The License Manager Service provides licenses to other components of the GeoDRM architecture. Especially the Authorization Service uses the License Manager Service retrieve applicable licenses for a request or a license reference for decision making.

The License Manager Service therefore serves the role of a policy administration point (PAP). It works closely together with

- License Broker Service: Negotiated transient and persistent licenses are stored at the License Manager Service.
- Authorization Service: During access control, the Authorization Service will provide the Gatekeeper with a decision. Beside the service request/response provided by the Gatekeeper, the Authorization Service will ask the License Manager Service for the appropriate licenses.

7.2.4.1 Interface description

The interface of the License Manager is logically divided into a discovery and a transaction interface.

- Transaction: methods needed to create, update and delete policies.
- Discovery: methods to find license, that match a request or a license token.
The discovery interface includes at a minimum a method to find a license token that matches to a license-token-reference. Other methods may be introduced to provide more sophisticated discovery functionalities (e.g. give all licenses that are applicable to a specific resource/action/subject etc.).

Interface technology will be SOAP/WS-S-based. Below you find an informal description of the License Manger Service's interface.

Interface
<ul style="list-style-type: none"> - GetLicense <ul style="list-style-type: none"> ▪ Description: retrieve an existing license ▪ Parameters: <ul style="list-style-type: none"> ▪ one of <ul style="list-style-type: none"> • license id • license reference (embedded into a SAML Assertion) ▪ Returns: GetLicenseResponse element, containing the complete requested license if it exists - CreateLicense <ul style="list-style-type: none"> ▪ Description: Store a license and make it accessible ▪ Parameter: a single license document ▪ Returns: CreateLicenseResponse element containing the operation result, i.e. success or failure (e.g. due to duplicate id, ...) - ReplaceLicense <ul style="list-style-type: none"> ▪ Description: replaces an existing license ▪ Parameter: a single license document that has the id of an existing license ▪ Returns: ReplaceLicenseResponse element containing the operation result, i.e. success or failure (e.g. due to not existing id, ...) - DeleteLicense <ul style="list-style-type: none"> ▪ Description: removes an existing license

- Parameter: id of the license to be deleted
- Returns: DeleteLicenseResponse element containing the operation result, i.e. success or failure (e.g. due to not existing id, ...)
- CreateLicenseReference
 - Description: creates a unique, non-permanent reference to a licence
 - Parameters: license id and expiration time
 - Returns: CreateLicenseReferenceResponse element containing the reference id (embedded into a SAML Assertion)

7.2.4.2 Interface security

Access to the License Manager Service provides the client with the power to establish “facts” that produce an immediate feedback on access control policies. The creation of a license grants permissions, license removal deprives the “owner” of the license (the subject whom is granted a permission) of the associated permission. Thus, access to the License Manager Service itself has to be restricted to trusted parties. In case of a SOAP based implementation of the License Manager Service, it is recommended to apply Web Services-Security (WS-S) to only grant access to trusted clients. Implementation can consider defining different access policies for the interface operation. The GetLicense operations may be subject to less restrictive policies, while license creation, replacement and deletion policies are more restrictive. If different license modifying clients use the license manager service to manage licenses, it is also recommended, but not necessary, only allow the clients to only manager their “own” licenses. This is a crucial requirement if different trust levels are established between the license manager service clients.

7.2.4.3 Signing

There is not only a trust relationship between license manager service and license managing clients but also between license consumer license consumer (an entity that retrieves licenses managed by the license manager service via the GetLicense or CreateLicenseReference operation) and license manager service. To enable the license consumer to verify the origin of license the license manager service digitally signs an issued license. The license consumer can then check if it trusts the issuing party and conditionally process the retrieved license.

7.2.4.4 Creation of a License Reference

In order to exchange licenses independently of the actual license encoding used by the license manager service, the service provides the CreateLicenseReference operation. A license reference is a license encoding-neutral reference (or id) to a license. The license reference differs from a license in that it has a limited temporal validity. If the license manager service issues a license reference, it has to guarantee that it can resolve the reference back to the actual license document inside the period of validity for the reference.

7.2.4.5 Capabilities

To interact with the License Manager Service the client has to have knowledge about the specific service instance. This contains

- operation access requirements: definition of access control methods that can be applied to authenticate the client when accessing access restricted operations, e.g. by means of WS-Policy
- maximal license reference validity: the maximum time a license manager service guarantees to be able to resolve license references into licenses

7.2.5 License Broker Service

Note: Specification or Development of a License Broker Service was not part of OWS-4. The following specification is theoretical and not proofed by prototyping.

A License Broker Service is the entity that is entitled to assign a license on behalf of the license issuer to a license consumer. This assignment can be of different type:

- Assignment is permanent and persistent (again, classical access control)
- Assignment requires a commercial pre-process (buying an assignment)
- Assignment requires a reconciliation between issuer and consumer (agreement to terms-of-use)

The license broker serves as a policy (license) negotiation and contracting point that is used by the GeoDRM client (License Issuing Service). References to such a License Issuing Service should be included in the capabilities (“precondition” part, see chapter 8.3) of a GeoDRM enabled service in order to enable a client to find, bind a License Broker Service and to get the appropriate license to use a service.

The main tasks of the License Broker Service are therefore:

- maintain references between products and applicable licenses (has the knowledge which licenses are available for which resources)
- offer possible license agreement
- negotiate a license agreement and
- conclude an agreement
- Induce the License Manager Service to create (store) a new license.

7.2.5.1 Interface description

The interface should cover functionalities to get available offers for licenses that are applicable to a service (its resources or products), to negotiate them if the offers contains user-selectable options for certain attributes and finally to conclude the license offer/contract. Some licenses require an identity for the license establishment.

As a final result of the interface interaction the License Broker Service will store an established license within the License Manager Service and the user will be provided with a reference to that license. This reference could be used to access the token itself (for displaying purposes) or it could be used to be communicated to a GeoDRM- (Gatekeeper-) protected OWS.

Below you find an informal description of the License Broker Service's interface.

Interface
<ul style="list-style-type: none"> - GetLicenseOffer <ul style="list-style-type: none"> ▪ Operation description: Retrieve available offers for a desired product for evaluation, negotiation and agreement: ▪ Request parameters <ul style="list-style-type: none"> ▪ product identifier ▪ optionally: identity ▪ Response content: a list of license offers for the requested product identifier with options (parameter name and potential values) for user defined input. - NegotiateLicenseOffer <ul style="list-style-type: none"> ▪ Operation description: Evaluates user input (e.g. text fields or option lists) and may reduce offer to relevant parts. The result is not legally binding und may be repeated ▪ Request parameters <ul style="list-style-type: none"> ▪ product identifier ▪ parameter name and user defined values ▪ Response content: List of preconditions for the requested product IDs with options (parameter name and potential values) with acceptable user defined input and indication of unaccepted user input. The responded offer may be reduced to relevant elements - ConcludeLicenceOffer <ul style="list-style-type: none"> ▪ Operation description: Although almost the same as NegotiateLicenceOffer, it represents a legally binding interaction, which

is not repeatable. In many cases a user identity is required. If successful, the License Broker Service stores an appropriate license at the federated License Manager Service(s).

- Request parameters
 - product identifier
 - parameter name and user defined values
 - optionally: identity
 - Response content:
 - A receipt with the concluded license (license reference) and a transaction number digitally signed by the License Broker Service
 - Reference to license itself
- A license reference is not the license token itself. The License Manager Service must be able to resolve the reference into the complete license.

7.2.5.2 Interface security

If the interface or parts have to be secured is depended on the business process for negotiating licenses. For example in strong security environments, access to the License Broker Service itself has to be restricted to trusted parties. When SOAP is used as implementation binding, it is recommended to apply Web Services-Security (WS-S) to only grant access to trusted or authenticated clients.

7.2.5.3 Capabilities

To interact with the License Broker Service the client has to have knowledge about the specific service instance. This contains

- Operation and product specific access requirements: definition of access control methods that can be applied to authenticate the client when accessing access restricted operations, e.g. by means of WS-Policy.
- List of available product and license offers that refer to those products
- If a digital identity necessary to induce the creation of personalized licenses, the License Broker Service may request the client to present a valid identity token (please refer to chapter 8.1). Such a requirement can be implemented as done in the OWS-4 Gatekeeper using precondition statements (see chapter 8.3).

7.2.6 GeoDRM-enabled Client

A GeoDRM-enabled OWS client exchanges messages with a GeoDRM-enabled OWS service. The purpose of the client (from a communications point of view) is to create request messages, send them to the service, wait for the response and then interpret the content of the response message. The communication is shown in the following picture:

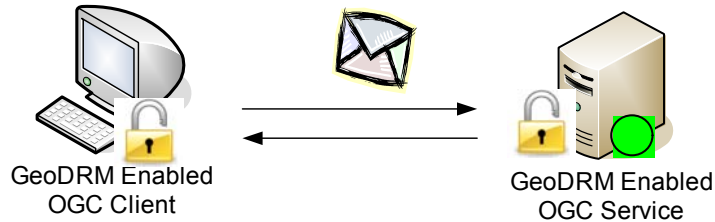


Figure 5 - GeoDRM Client – Service communication

The following picture details the content of the message:

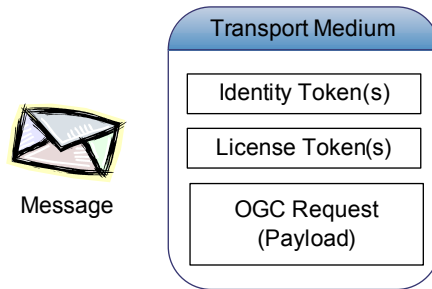


Figure 6 - Message content

As seen in this picture, a request message contains a payload in the form of an OGC request (such as GetMap / GetFeature / etc.) and several security tokens (be that either identity or license tokens). The message is transported over the network by means of a transport protocol (such as SOAP, HTTP POST/GET). Therefore, payload and the tokens are encapsulated and encoded according to the transport protocol used.

The following picture details the GeoDRM Enabled OGC Client and shows its main components. Please note, that these components can be realized as a single integrated software program or as independent software components.

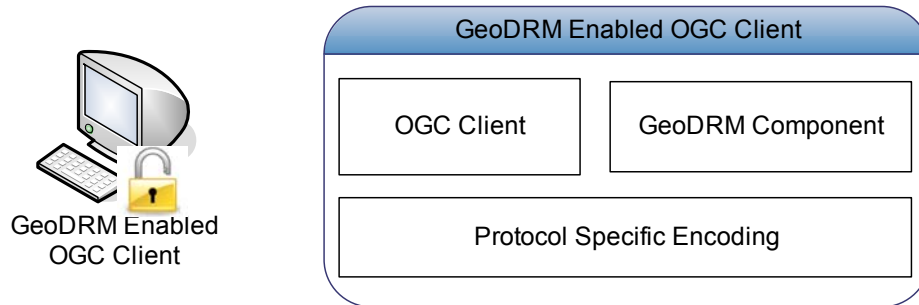


Figure 7 - GeoDRM Enabled Client composition

The role of each of the 3 components:

- OGC Client – is a WMS / WFS / etc. client. Its purpose is to implement the geospatial specific tasks & generate the payload of the message (GetMap / GetFeature / Transaction / etc.)
- GeoDRM Component – its task is to take care of the GeoDRM relevant aspects. These include the following:
 - Parsing the GeoDRM relevant metadata of the service (capabilities XML documents, WSDL, etc.) and discovering whether the service requires security tokens, what type of security tokens, how these should be encoded, etc.
 - Acquisition of Identity Tokens and / or License Tokens if necessary.
 - Management of the security tokens – this might imply persistently saving the tokens, updating the tokens when necessary, reacquisition if the tokens expire, key management if PKI is required by tokens, etc.
 - Selecting the proper tokens to be attached to a request message
- Protocol Specific Encoding – it is the task of this component to bundle the payload and the security tokens in a message and encode this message according to the transport protocol being used

7.2.6.1 Implementation choices

From a service point of view it does not matter how the client is implemented. All that a service need to know is that the client properly implements the service interface and that it generates proper request messages (the payload and security tokens are bundled and encoded correctly).

However, three approaches for a client component were implemented:

- “Integrated” – here all the components are bundled in a single software program. This has the advantage that the client can be implemented efficiently. This was one of the approaches taken by the conterra/ESRI implementation, see chapter 7.2.6.2.2 for details.
- “Client-Side Proxy” – a stand-alone software component, which covers GeoDRM functions. This allows a simple integration of currently existing software (which do not support GeoDRM). This approach was taken by the UniBW implementation, (see section 7.2.6.2.5 for details).
- “Server-Side Proxy” – this implementation follow the same approach as the “Client-Side Proxy” but offers its functionality on the server-side. This was the other approach taken by the conterra/ESRI implementation See chapter 7.2.6.2.3 for details.

7.2.6.2 GeoDRM Client Implementations

7.2.6.2.1 The GeoDRM Enabled Client from conterra / ESRI

The GeoDRM-enabled OWS client serves as point of access to a trusted and secured geospatial services infrastructure. GeoDRM-enablement could be achieved by integrating additional functionality to an existing OWS client or as a separated component acting as proxy. In both cases, the functionalities and responsibilities of this component are:

- Request and interpret GeoDRM-specific preconditions (either as exception or capabilities) and engage the appropriate workflows with authentication service and/or license broker in order to full fill all stated requirements to use the OWS service.
- Establish a connection to an authentication service in order to get a valid identity token (only needed for online authentication; offline authentication identities may be provided to the client asynchronously, e.g. an x.509 certificate via email).
- Establish a connection to a license broker to negotiate the required license and get the appropriate license reference token.
- Enhance to the communication to a GeoDRM-enabled service by adding the appropriate identity and/or license reference tokens to the service request

The GeoDRM Enabled Client was implemented as integrated GeoDRM Client by extending an existing WMS map viewer and as a server side proxy client

7.2.6.2.2 Integrated WMS Client

The following figure shows the component diagram of the GeoDRM-enabled OWS client.

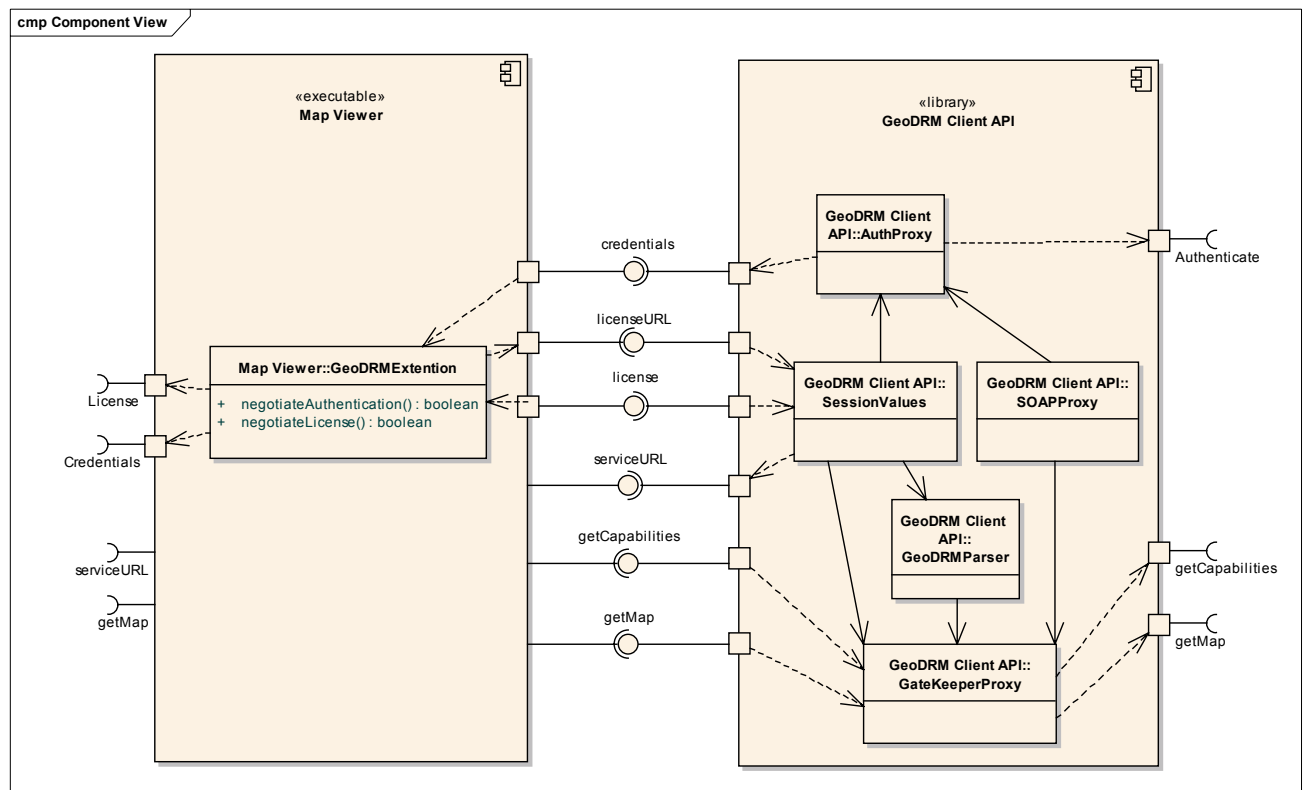


Figure 8 - GeoDRM Enabled Client component diagram

An existing map viewer client was extended to interoperate with a GeoDRM enabled service. Since the map viewer already understands some sections of GeoDRM payloads (except the security/token sections) a facade API was used as a mediator between the Map Viewer and the GeoDRM infrastructure.

The Map Viewer does not implement SOAP communication which is a requirement for the GeoDRM client to communicate with the gate keeper and authentication service. Map Viewer messages are therefore encapsulated into SOAP envelopes in the API by the SOAPProxy class. This process may also involve the API attaching to the SOAP message a security header formed from license and/or authentication tokens received. Messages

received by the API are stripped off their SOAP specific elements and the payload passed on to the Map Viewer.

Map Viewer Component

The Map Viewer Component serves as the GUI for the user. The user inputs a series of information which in the end results in a viewable map. The following was implemented in the Map Viewer application.

- Interface Definitions
 - serviceURL - GeoDRM service URL input by user
 - License - User negotiations License with license broker and inputs (copy & paste) license reference tokens received from negotiation
 - Credentials - User inputs credentials (username and password) for authentication
 - getMap - User chooses layers to view. These layers are extracted from the capabilities document and presented to the as viewing options.

GeoDRM Client API Component

The GeoDRM Client API Component is the interface between the Map Viewer and the other GeoDRM components.

- Interface Definitions
 - Credentials - Once the Map Viewer obtains the credentials from the user, these credentials are passed onto the GeoDRM API which in turn passes it to the Authentication service. The resulting authentication reference is stored in the SessionValues class.
 - licenseURL - In order for the Map Viewer GUI to present the license for negotiation, a URL for the license needs to be supplied from the API. This URL originates from the capabilities file (SessionValues class).
 - License - Once the license token is obtained by the user, the license token is passed back to the GeoDRM API and stored in the SessionValues class.
 - Service URL - URL of GeoDRM service to initiated GeoDRM services.
 - getCapabilities - The map viewer needs the capabilities document in order to display to the user which services the user the layers the user is allowed to access.
 - getMap - The Map Viewer requests a map from the API to be displayed. The API uses the URL and tokens to request the Map from the gatekeeper.

7.2.6.2.3 Server side proxy client

The GeoDRM Client API Component was reused to build a web based GeoDRM proxy client that acts similar to the WFS Client side proxy from UniBW (see chapter 7.2.6.2.5) but runs on the server side (server side proxy). It can be used to point to GeoDRM Enabled OWS services and will coordinate the acquisition of the proper identity and/or license reference tokens and add them to the service requests.

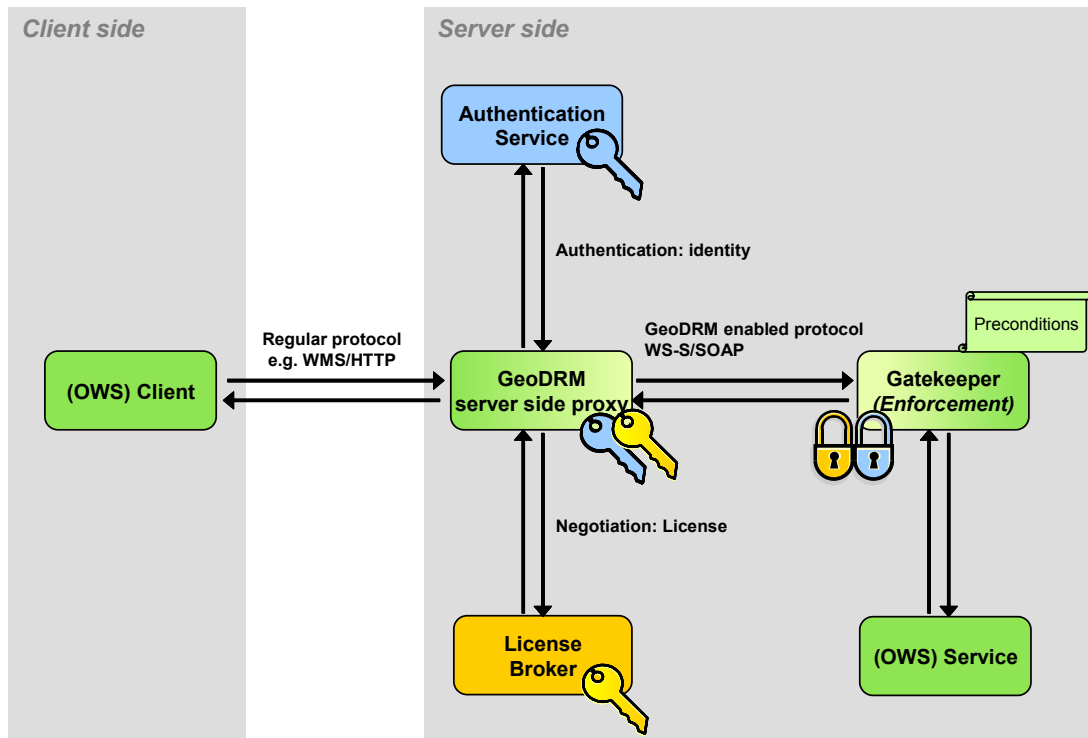


Figure 9 - Communication flow with server-side proxy

As shown in Figure 9 all GeoDRM enabling components run on the server side (independent of the distribution opportunities). The GeoDRM server side proxy is set up as a web application that offers either integrated user interfaces (for authentication purposes) or points to external user interfaces (for license negotiation with the license broker application). A user has to interact with those interfaces to get the appropriate keys that will open access to a GeoDRM protected service. “Open access” is realized via a new service end point that proxies the underlying OWS service interface and communicates to the Gatekeeper service that does the protection.

Advantage is that such a system is easy to set up and that there are no requirements for the client side. Even existing off-the-shelf OWS client and service software can be used. Disadvantages are that the “last mile” between OWS client and GeoDRM server side proxy is only weakly secured and that this approach uses a dynamic service endpoint from the client’s perspective.

7.2.6.2.4 Demonstration

The con terra/ESRI Inc GeoDRM enabled client was demonstrated during the GeoDRM WG session from the OGC TC meeting in San Diego, CA. The integrated WMS client can be found at:

<http://prime.esri.com/arcexplorer>

7.2.6.2.5 The WFS Client side proxy from UniBW

During OWS-4, UniBW implemented a client-side proxy for transactional WFS as in-kind contribution to the initiative. The client was demonstrated during the OWS-4 demonstration event in New Jersey in front of the sponsors. Furthermore the client was also demonstrated during the GeoDRM WG session from the OGC TC meeting in San Diego, CA. The client is open source software and can be found at the following URL (together with further information & demos):

<http://iisdemo.informatik.unibw-muenchen.de/ows4/>

7.2.6.2.5.1 Architecture

The client from UniBW follows the client-side proxy pattern – see figure below. It is implemented in Java and runs on the computer where the OGC Client runs. The OGC client (can be any off-the-shelf WFS client, has been successfully tested with Geomedia from Intergraph and the uDIG open source client) communicates with the GeoDRM Client-Side Proxy by means of the normal WFS protocol (using HTTP POST). The Client-Side Proxy enhances the request with the appropriate security tokens, encodes it using SOAP and then sends the message to a GeoDRM enabled service.

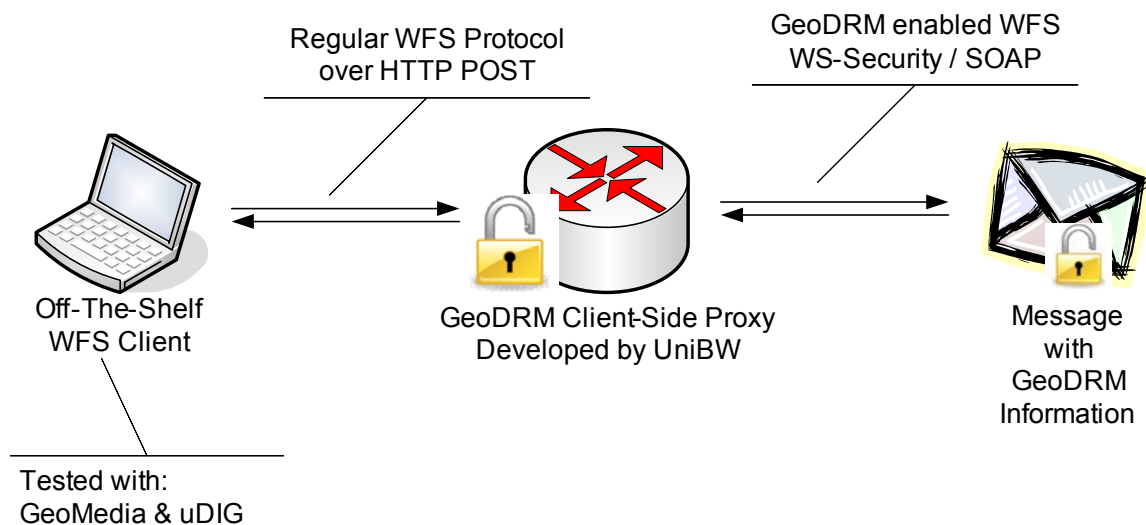


Figure 10 - Communication flow with client-side proxy

7.2.6.2.5.2 Graphical User Interface

In order to have an idea about how the client is built and its features, the following figure shows a screenshot of the client:

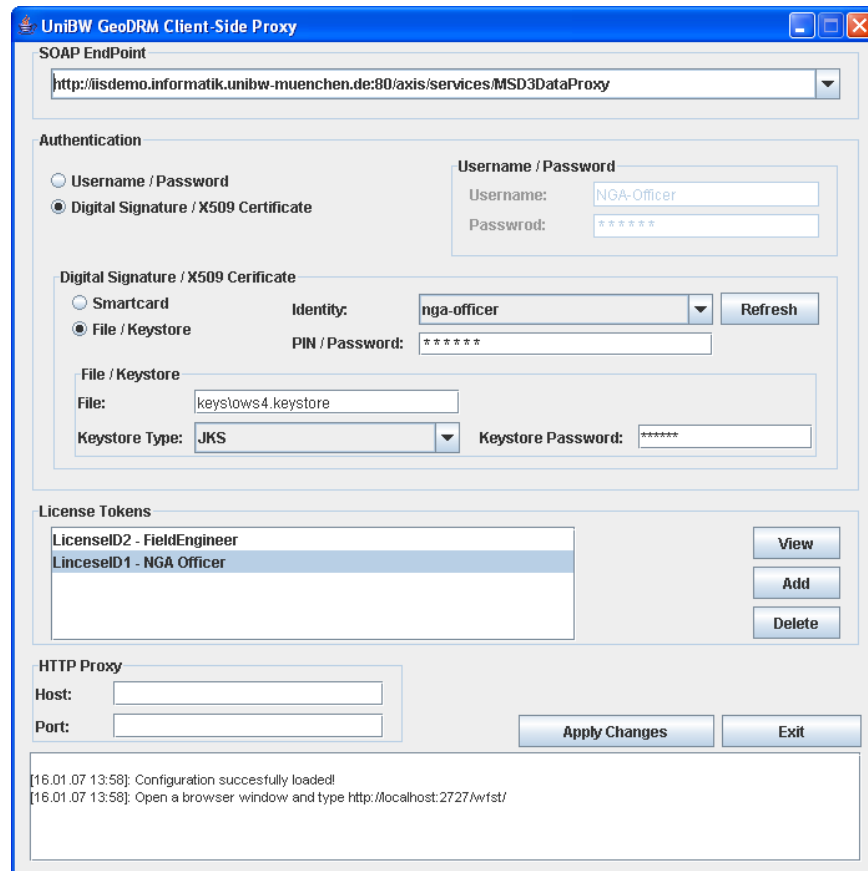


Figure 11 - User interface of the client-side proxy

The client allows the user to configure the following:

- The end-point of the remote service
- Authentication method
 - Username / Password
 - Digital Signature / X.509 certificates
 - The private key / certificates can be either stored on a smart card or in a regular file
- License token

- A simple license token management is provided (this way the user can use several license tokens)

The client was implemented in Java and is built on several open source packages: Apache AXIS, Apache WSS4J, Internet2 OpenSAML.

7.2.6.2.5.3 Demonstration

A demonstration of this client can be found here:

<http://iisdemo.informatik.unibw-muenchen.de/ows4/>

8 Information model

8.1 Identities and Identity Tokens

Identity as a concept is represented in a system with informational artifacts which contain identity information. In the context of a federated system identity is controlled by a service instance called Authentication Service.

The encoding of identity relies upon the purpose and characteristics of an implementing system, whereas the GeoDRM group focused on Anonymous, Single and Group identity. All system entities have to implement at least one identity model to allow reference from authenticated identity to contracting identity to authorized identity. The implemented identity information model consisted of a username, a password, an alias, and a group, whilst not all those attributes were used throughout all of the given use cases. Also the identity information model was not encoded to the full extent by all system entities.

An identity token refers to a single identity and is defined in a well-known format for exchange. Identity tokens may have various limitations in validity associated with them, e.g. a certain interval of time. There is no implicit need for the processing service to call back to the Authentication Service for validation if the validity of a qualified (session-associated) token can be ensured by usage of signing techniques.

Identity tokens, as far as this document is concerned, are sent together with a request to the service provider. Each request to the service provider that needs to be authenticated (each message exchanged) will contain one (or more) identity token. These tokens should be attached to the message in a manner that would not allow eavesdroppers to capture the token and use it in reply attacks.

During OWS-4 the group only focused on SOAP as the transport protocol. SOAP was chosen because a lot of the standardization effort in the field of security was invested in it during the past years. Future initiatives of the OGC should investigate how the Get and Post bindings that use HTTP as transport protocol can accommodate security tokens.

8.1.1 Identity token encoding

One way to implement security tokens is via the WS-Security specification from OASIS. This specification proposes a standard set of SOAP extensions that can be used when building secure Web services to implement message content integrity and confidentiality. This specification is flexible and is designed to be used as the basis for securing Web services within a wide variety of security models including PKI, Kerberos, and SSL. Specifically, this specification provides support for multiple security token formats, multiple trust domains, multiple signature formats, and multiple encryption technologies. The token formats and semantics for using these are defined in the associated profile documents. It is not the purpose of this document to describe in detail the syntax and semantics for identity tokens and how each token type should be attached to SOAP

messages. This information is found in WS-Security and its accompanying token profiles. In the following only the authentication method and point to the WSS token profile will be enumerated.

8.1.1.1 Username / Password

Username / Password authentication is described in the Username Token Profile v1.1 available at the following address:

<http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf>

Although this authentication method is quite trivial, the above mentioned specification does a good job by adding features that help protecting the password and avoiding reply attacks. These features include the use of hashed passwords, nonces, creation dates. Furthermore a key derivation algorithm is presented which can be used for when computing Message Authentication Codes (MACs) or as a symmetric key for encryption.

8.1.1.2 Kerberos

Kerberos is a computer network authentication protocol which allows individuals communicating in an insecure network to prove their identity to one another in a secure fashion. Kerberos is a project originating from MIT, has various implementation, one of them being used in Microsoft Windows networks. The Kerberos Token Profile v1.1 (with errata) is available at the following address:

<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-KerberosTokenProfile.pdf>

The document describes the use of Kerberos tokens with respect to WS-Security. It specifies how to encode Kerberos tickets in SOAP messages and how use these tokens for digital signatures and encryption.

This authentication method was not used during OWS4.

8.1.1.3 PKI / X509 Certificates

An X.509 certificate specifies a binding between a public key and a set of attributes that includes (at least) a subject name, issuer name, serial number and validity interval. Public Key Infrastructures based on X.509 certificates are widely deployed today for authentication (usually done by means of digital signatures) and encryption.

The use of X.509 authentication framework in the context of SOAP web services is described in X.509 Token Profile v1.1 (with errata), available at the following address:

<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-x509TokenProfile.pdf>

The document describes how X.509 certificates (or references to them) can be attached to SOAP messages. Furthermore the document describes how digital signatures and encryption blocks can reference keys attached in this fashion.

If a digital signature is verified and the key used for the digital signature is associated to an identity, a service provider can assert, using the X509 certificate attached to the message (and assuming that this certificate is valid) the identity of the message emitter.

8.1.1.4 SAML

A particular method for encoding identity tokens in federated environments is the Security Assertion Markup Language Standard [SAML]. SAML makes use of statements which assert certain characteristics of a subject (claims), e.g. a subject's authentication, name and role. SAML can be used to encode qualified identity tokens and may be combined with XML-Signature. Security Assertions Markup Language (SAML) is an XML standard from OASIS designed for exchanging security information. SAML is a very flexible specification and can be used in a multitude of scenarios. One of the problems that it tries to solve is the Single Sign On problem and in this way it can be used for authentication in scenarios. Because it addresses Single Sign On, SAML is of course ideal for federation scenarios where users come from a different security domain than the one where the service provider is. SAML denotes a various types of statements for expressing identity information, which all are referring to an included subject. The structure of an assertion object is shown below.

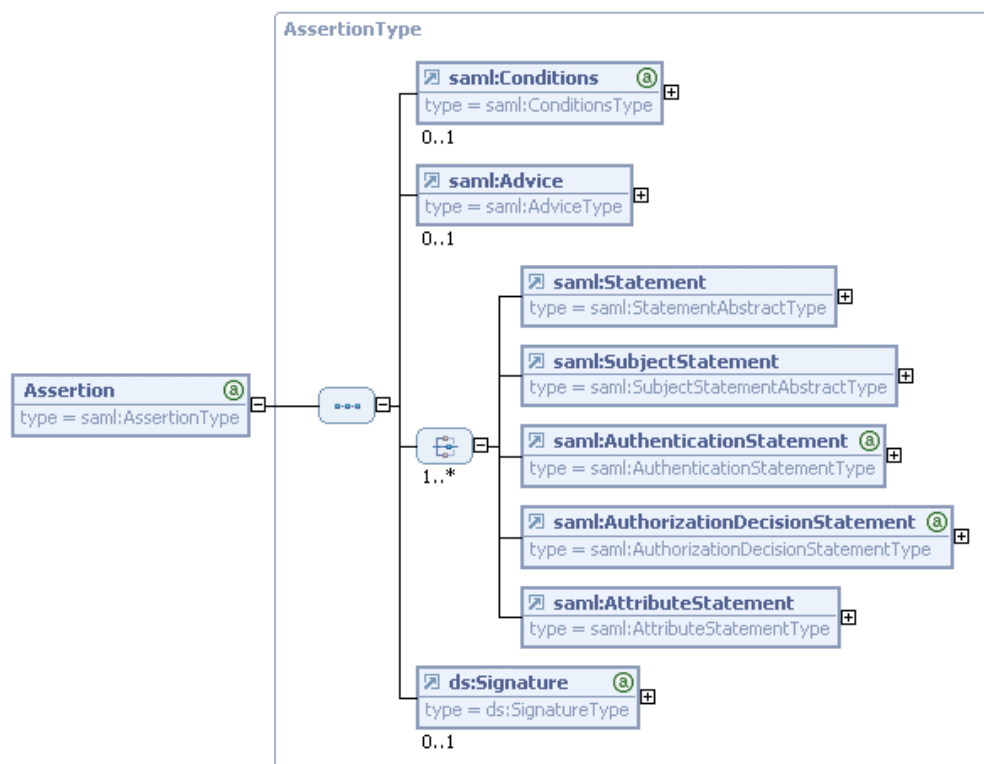


Figure 12 - SAML Assertion

An AuthenticationStatement, which is used to assert that a subject did indeed authenticate with the identity provider at a particular time using a particular method of authentication is depicted below.

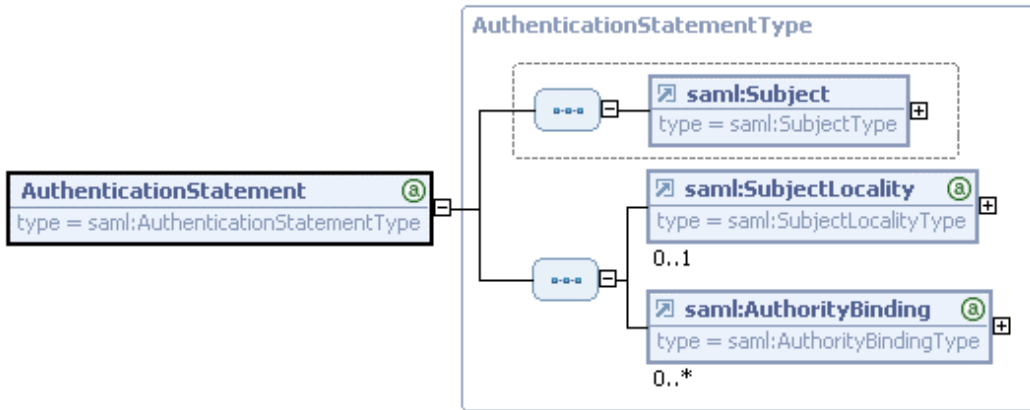


Figure 13 - SAML AuthenticationStatement

Other characteristics applying to the given subject can be expressed by SAML's AttributeStatements.

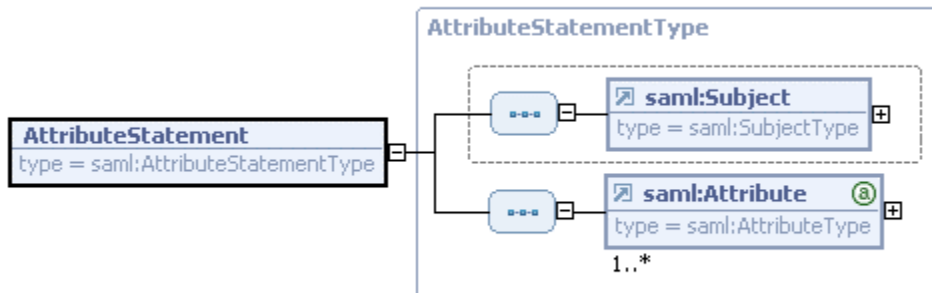


Figure 14 - SAML AttributeStatement

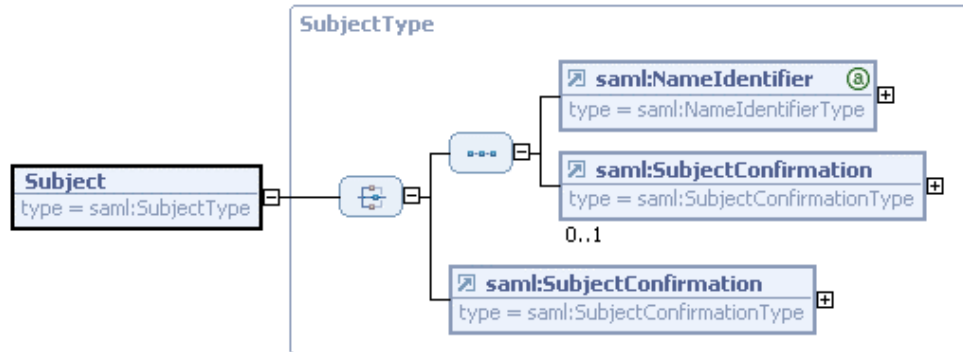


Figure 15 - SAML Subject

The integrity of identity tokens handled by various parties can be assured by XML digital signature [XMLSig], which is applied to an Assertion artifact containing one or more Statements. If the identity provider is known to the service provider, esp. PEP, PDP, the integrity of the identity information contained in an identity token can be verified.

8.2 Licenses and License Reference Token

A GeoLicense is the expression of the rights and constraints on those rights to be performed against a geospatial resource. It is the container expressing the rights to use a specified geospatial resource, for a given geographical space, over a specific period of time – subject to other conditions (from [GeoDRM RM]).

The GeoDRM group distinguishes between a license and a license reference that acts like a pointer to a particular license. A license is a document containing information about the issuer of a license and the description of a policy that can grant permission if all conditions and obligations are fulfilled. A license reference is a small pointer like document with reference to the issuer that points to the location of a license and has no more license information than that.

Usually, the License Broker Service returns a license reference to the contracting party (“the customer/user”) as a token (license reference token). Together with the service request and identity token, the user transmits the license reference to gatekeeper to signal, that the license that is represented by the reference shall be used in policy decision making. The gatekeeper forwards the reference to the Authorization Service, which resolves the reference communicating with the issuing License Manager Service and takes the resulting license into account during the decision making process.

8.2.1 Licenses

The following diagram provides an overview of the logical model of a license as used in the OWS-4 GeoDRM initiative:

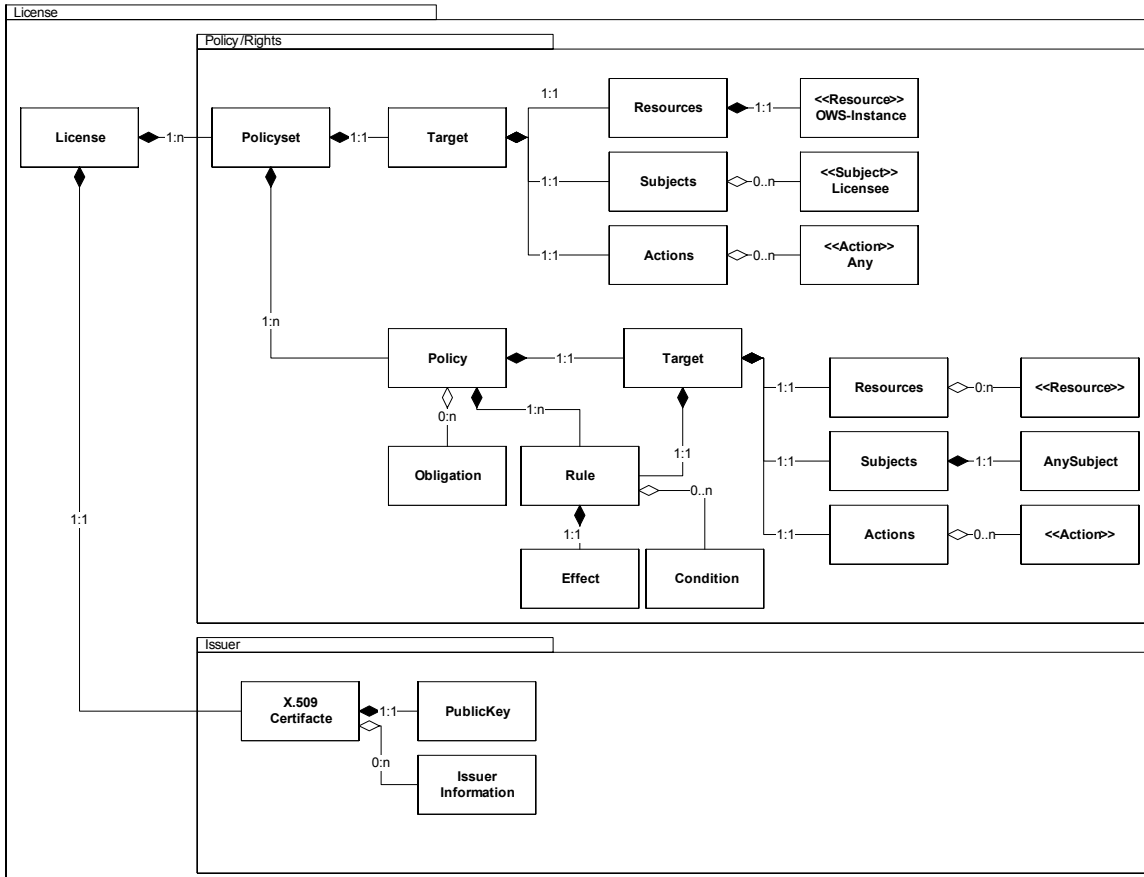


Figure 16 - License Model

Licenses consist of one or more rights assigned to a user (licensee) and a signature that represents the license issuer.

A license contains at least one policysset. This policysset is a container of policies that represent the rights the license include and therefore the licensee owns. The policies/rights used in OWS-4 are always positive (~grants). The authorization side uses a “positive, closed” decision making pattern which is expected to be the most secure one. Only positive policies are used to avoid conflicting policies which could lead to security leaks and a granting permission is only given, if the positive decision is non-ambiguous. All “not applicable” decisions are interpreted as “deny”.

Each policy may apply to a combination of resources and actions while the subject is usually defined on the root of the policysset as the subject is expected to be the licensee. The licensee could be expressed in various ways including single human entities, groups, roles or indirect entities like sessions or IP-addresses.

Resource and actions *can* refer to the requests and resources that a particular OWS service offers. E.g. a WMS may offer a resource “layer” and a WFS a resource “featuretype”. Actions may accordingly be “GetMap” or “GetFeature”. Good experience was made with including the whole service instance and type as a default resource for licenses applying the OWS services. Beside those resources and actions that could be derived from the service interface specification there are truly other actions and resources. Problem with those definitions is that they are difficult to interpret when sharing licenses among multiple decisions points. On the other hand defining a fix list of actions/resources that could be used for grants within policies may be too restrictive and inflexible.

In any case it is proposed to define and use strong typed subject, actions and resources that should include a non-ambiguous designator for the type and the instance (maybe even the datatype). For example:

Type	Instance	
OGC:WMS	http://SERVER/wms?	Resource
OGC:WMS:Request	GetMap	Action
Subject:Identifier	Persons_name	Subject

Types should be defined as Uniform Resource Names (URN).

Each policy may have conditions that have to be met in order to “get the grant”. Various types of conditions could be realized by modifying the policies that are part of the license. For example, the IP matching was formulated as equals-condition within the policy. Other conditions that could be checked during the authorization process could be time, date or other information that could be derived from the interaction context (scale, extent, etc.)

Other tests included an interesting feature of the used XACML encoding that enabled it to formulate obligations that are bound to a grant and that have to be performed by the enforcement point (Gatekeeper). Such obligations could be used to express pre- or post-processing instructions outside the gatekeeper. Like reduction of the quality of map images for guest accounts, clipping of certain areas, filtering of sensitive attributes etc.

8.2.2 License encoding

The encoded and applied licenses used in OWS-4 GeoDRM contain the following information:

- **id**: unique character sequence that identifies a license instance
- **policy set**: a set of policies describing permissions/denials attached to the license
- **signature**: a digital signature that ensures the integrity of the license and contains license issuer information (at least the name of issuer)

Licenses shall be encoded using the schema denoted in Figure 17.

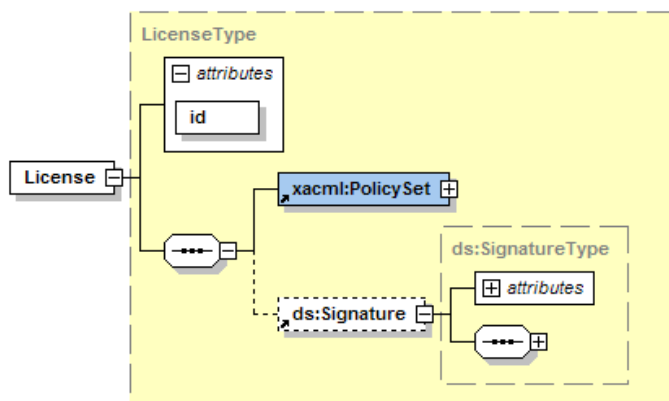


Figure 17 - License encoding schema

The License id attribute may be of any type but has to be unique in the application system. The policy set shall be encoded using the eXtensible Access Control Markup Language (XACML), version 1.1 [XACML]. The <License> element shall be signed using “SOAP Security Extensions: Digital Signature” [XMLSig].

In OWS-4 GeoDRM rights/grants are encoded as XACML, but other REL languages could be applied as well.

Problem with a GeoDRM architecture dealing with different REL encodings is the encoding dependency between:

- License Broker and Manager for license creation and modification
- License Manager and Authorization Service for license resolving
- Gatekeeper and Authorization Service for decision making

In OWS-4 the group tried to face those problems by using a common container structure for the license that contains the policies as a block and a reference token to be communicated. That way the License Broker is enabled to create a license in the REL it

supports, pack it into a common license object which is transmitted and save in the manager using the manager's interface (which is not depended on any REL context definition language). Same counts for resolving licenses.

The latter problem remains; the dependency is obviously: a decision making component like the authorization service need to support the REL implementation(s). And the gatekeeper needs to know what input is required by the decision making component (this is usually not only depended on REL but to make it worse on the decision making capabilities).

Proposal: Limit dependencies on REL encoding among the interfaces and information objects, inform about the dependency on supported REL(s) for example via the Gatekeepers capability extensions (preconditions).

The Licence Issuer is represented as X.509 Certificate. This certificate includes sufficient information about the issuer. In addition it will be used to sign the rights parts of the licence to ensure integrity and enables trust between license-provider and consumer.

Future requirements may need to use another encoding for the issuer as its information is limited (e.g. if issuer lineage information is needed).

8.2.3 License Reference Tokens

OWS-4 GeoDRM use and communicate license tokens in the form of a reference to the real "license" (therefore named "license reference token"). That way a license reference token as defined here is only a pointer, that is not valid without resolving the complete license token. This decision was made in order to make the general architecture more flexible in terms of what encoding is used and avoid REL encoding details in the interfaces with predominantly dynamic binding, license are transported as reference tokens.

A reference token will be created by the License Manager after successful negotiation between Client and License Broker. The reference will be handed out to the client who adds it (as WS-S Header) to the Gatekeeper communication. To ensure the integrity of the reference it will be signed as well. The Authorization Service will use the wsse:Reference element to dynamically resolve the complete license token via the license identifier (AssertionIDReference) from the License Manager.

License references are not directly bound to a subject or user for whom the license reference was created. Instead, the entity that uses the reference (e.g. the GeoDRM-enabled client) to signal the use of the underlying fully-featured license document, vouches that the reference belongs to the subject who it acts for. The subject is identified by additional authentication information inside the WS-S header but outside the license reference by means of a x.509 certificate, for example.

8.2.4 License Reference Token Encoding

A License reference used in OWS-4 GeoDRM was encoded as an assertion of the Security Assertions Markup Language, V1.1 [SAML]. License reference information is expressed by attribute values of an AttributeStatement.

The following figure shows integration of license reference information into a SAML Assertion:

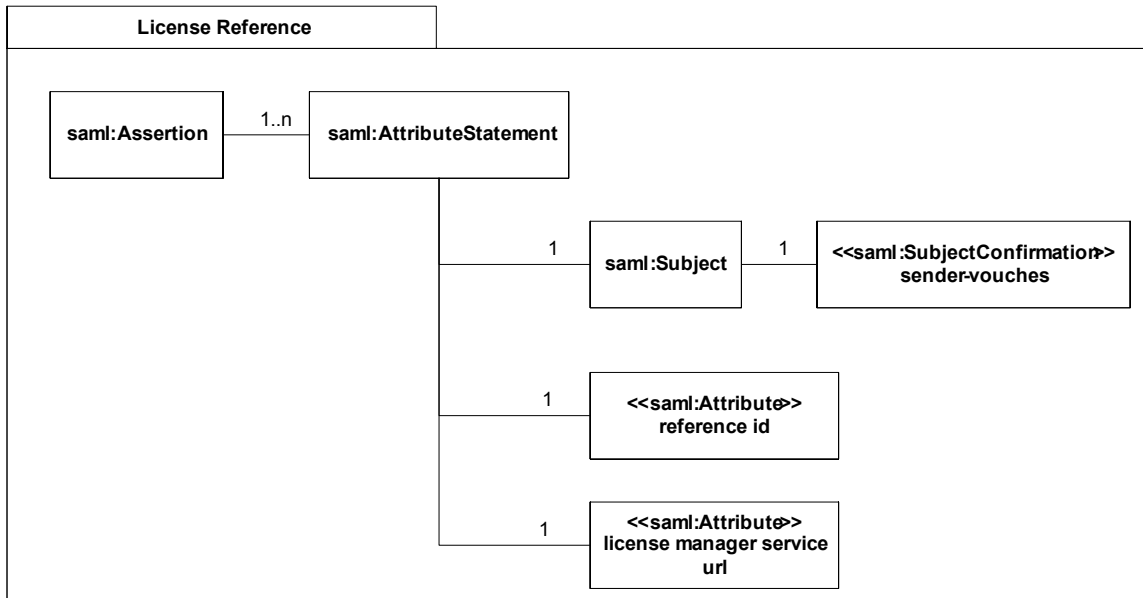


Figure 18 - License Reference model

An assertion that represents a license reference must at least contain an AttributeStatement element including:

sender vouches

This element contains the confirmation that the sender of the assertion vouches for the assignability of license reference and externally provided identity information.

The text content of the saml:SubjectConfirmation element shall be `urn:oasis:names:tc:SAML:1.0:cm:sender-vouches`.

reference id

This attribute contains a unique character sequence that identifies a license reference instance (not the license instance or id!).

The value of the “AttributeName” XML attribute of this element shall be

`urn:opengeospatial:ows4:geodrm`.

The value of the “AttributeName” XML attribute of this element shall be

`urn:opengeospatial:ows4:geodrm:licenseID`.

The text content of the “AttributeValue” child element of this saml:Attribute shall be the license reference id.

license manager service url

This attribute holds a reference to the License Manager Service that stores the license token itself.

The value of the “AttributeName” XML attribute of this element shall be

```
urn:opengeospatial:ows4:geodrm.
```

The value of the “AttributeName” XML attribute of this element shall be

```
urn:opengeospatial:ows4:geodrm:licenseManagerUrl.
```

The text content of the “AttributeValue” child element of this saml:Attribute shall be the URL to the License Manager Service.

As shown, identities and license tokens are both encoded utilizing the SAML Assertion approach. In OWS-4 both tokens are kept separately from each other. It may be very useful to see and understand license reference tokens not as separate assertion but included into in the identity token and as a part of it. That way a user’s identity could carry assertions about the licenses he owns inside. This is a potential subject to be further elaborated on.

8.3 Extensions to Capabilities – GeoDRM Preconditions

From an information model point of view, preconditions should describe the policies under which a GeoDRM enabled service is useable.

Therefore preconditions need to include at least the following information

- Type: "what" kind of tokens a client has to provide
- Issuing authority: "where" the client could get those tokens if this process is not out-of-bands
- Subject of a precondition: "wherefore" they are needed (~ product definition that a service offers)

Preconditions are additional, GeoDRM functionalities describing elements that should be integrate able to existing OWS capabilities as well as to upcoming (WSDL-based) capabilities, but also to other describing information objects like catalogue-metadata, web map context documents, exceptions, etc.

8.3.1 Type of a Precondition

Preconditions define what is needed in order to do something on a spatial resource. In other words, preconditions define the lock and what type of key is needed in order to open it. These keys are referred to as "tokens". At least two different types of tokens were identified:

- identity tokens, where the key is an identity
- license (reference) tokens, where the key is the possession of a license

Chapter 8.2 explains the difference between a license and a license reference token in detail. As license reference tokens are used for communication purpose, these are the type of tokens described in the preconditions. However, further extension may include direct transport of complete license tokens based on different encodings. In order to support those alternatives, type information for license tokens could be extended by the following additional information:

- What encodings are accepted
- What subtype of license tokens are accepted (license reference token and/or license token)

8.3.2 Issuing authority

Some types of identity or license tokens may be obtained "out-of-band" meaning asynchronously via email, postal etc. Those out-of-bands tokens are already in the possession of a client and do not need to be issued dynamically.

Especially in "federated" infrastructure scenarios those tokens are requested, negotiated and issued on-the-fly. An issuing authority can be used to get an appropriate token. This requires that a precondition does not only state what type of token it needs but also where clients (or requestors) can obtain these tokens. That way, the preconditions need issuing authority information if tokens are to be requested at runtime. The issuing authority should include:

- Type of issuing authority
- Service endpoint (URL) of authority if it is a service

8.3.3 Subject of a precondition

The subject of a precondition defines to what a precondition is related to. As OWS-4 GeoDRM deals with OWS services, it is most likely that the subject is an OWS service instance, its functional capabilities or its contents. These are expressed via their capabilities-document. Therefore, precondition expressions are related to OWS capabilities and their elements.

There are scenarios that impose that a more complex definition of a subject is required.

- There are valid scenarios where preconditions need to be applied to certain combinations of content and functionality: e.g. "you need to be authenticated in order to perform a GetFeatureInfo-Request on the layer "parcel property" where as the other layers are free to be used".
- Other scenarios impose, that preconditions apply to **derived** contents/functionalities etc. e.g. a service-cascade: "you are only allowed to cascade the layer aerial imagery in your own service, if you buy a "CommercialPayPerClick"-License". Other examples may refer a precondition to a subject that is not standardized, but domain(or software)-specific like "in order to print the map as PDF file, you need to include a reference to me as the owner of IPR"

To support those scenarios it is necessary to extend the definition of the subject of a precondition. This explicitly defined subject could refer to capability elements, can combine and extended them as needed by the scenario. As such a precondition includes not only the token-types and possible issuing authorities but the subject as well.

From a business perspective such a subject could be seen as merchandise that is offered through a service. As stated above, a service could offer several different "products",

each of them with the potential to have other preconditions that apply. Therefore, the subject of a precondition should be technically equal to the definition of what the product is.

8.3.4 Precondition Encoding

From the encoding of view, those preconditions should be express- and transportable in harmony with the general transport and encoding mechanism SOAP, WS-Security and relative (compatible) standards.

"WS-Policy" [WS-P], "WS-PolicyAttachment" [WS-PA] and "WS-Addressing" [WS-A] are compatible with WS-S and related web service technologies and standards. Therefore those were the standards used to encode preconditions.

Type Encoding

Both general types define encodings of token: for identity tokens these are inline with WS-S (e.g. X509-certifact, SAML-Assertions, Username/Password, etc.). For license tokens, OWS-4 GeoDRM supports a SAML-Assertion based encoding

From an encoding point of view, WS-Policy could be used as a general container for preconditions. WS-SecurityPolicies [WS-SP] could very well be used for describing identity tokens types. For license tokens types, it is necessary to define GeoDRM schema for the types. This extension is inline with the WS-Policy standard.

Issuing Authority Encoding

From an encoding point of view, WS-Policy does not define an authority, but WS-Federation extends the mechanisms defined in WS-PolicyAssertions ([WS-PA], Extension to WS-Policy [WS-P]) to define a "RelatedService" assertion. It defines (but is not limited to) four types of such services:

- Related Identity Provider (IP).
- Related Security Token Service (STS).
- Related Attribute Service (AS).
- Related Pseudonym Service (PS).

The first two are of interest: "Identity Provider" is (from the logical point of view) the Authentication Service and a "Security Token Service" is a License Broker Service [OWS4Trust].

Subject Encoding

When applying WS-Policy in SOAP-WS infrastructures, they are embedded into WSDL-descriptions. WS-Policy elements are "per se" applicable to the following artefacts of a WSDL-document:

- service: WS-Policy elements that apply to the whole service
- binding/ports: WS-Policy elements that apply to ports, e.g. functions and their binding
- message/type: WS-Policy elements that apply to communication and messaging (request, response)

Mapping that to existing OWS capabilities would enable to apply WS-Policy based preconditions to the whole OWS instance, to single requests or to sub resources like layers, etc. As WS-Policy Elements are placed within the capabilities xml element that they refer to (and that is assumed to be the subject of a WS-Policy), they are spreaded in the capabilities document. This approach is called an "in-line" precondition.

From an encoding point of view, "in-line" preconditions can be implemented with WS-Policy "as is", no definition of a subject is needed for this. Example

```
<Service>
  <...></...>
  <wsp:Policy>
    <[REFERS TO "Service"]/>
  </wsp:Policy>
</Service>
```

From an encoding point of view, more complex subject (product) definitions should be defined "en-block". Such can be done using the WS-PolicyAttachment extension [WS-PA]. As this allows the definition of a domain-specific subject that a WS-Policy applies to.

```
<wsp:PolicyAttachment>
  <wsp:AppliesTo>
    <[DEFINITION DOMAIN SPECIFIC SUBJECT; PRODUCT]/>
  </wsp:AppliesTo>
  <wsp:Policy>
    <[REFERS TO DOMAIN SPECIFIC SUBJECT]/>
  </wsp:Policy>
</wsp:PolicyAttachment>
```

Such an "en-block" definition is valid and interpretable "stand-alone". This subject-information could be used by the client in order to get "fully informed" and to get an idea of what action might require which token. Other advantages are that such a block could be integrated into other information models as well (such as catalogue-

metadata, WMC, exceptions, etc.). When moving OGC service specifications and bindings to SOAP, these constructs remain valid and could be reused.

From the technical point of view, both "in-line" and "en-block" may be used even together; both seem to be valid. The in-line is especially interesting for preconditions that apply to whole service, because they are pretty easy to set up and maintain. On the other hand, the "in-line" approach offers a very high degree of freedom to interpret, especially when there is more than one WS-Policy statement and when these are "mixed-type referred".

Therefore, OWS-4 GeoDRM uses the "en-block" approach. The domain-specific definition of the subject is based on WS-Addressing [WS-A] and subject for further elaboration and standardization (profiling).

8.3.5 Example for preconditions - Combined identity and license precondition

```

<?xml version="1.0" encoding="UTF-8" ?>
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <WMT_MS_Capabilities version="1.1.1">
- <Service>
  <Name>OGC:WMS</Name>
  <Title>Web Map Service GeoDRM</Title>
  <Abstract>ArcIMS 9.1.0GeoDRM Web Map Service</Abstract>
+ <KeywordList>
  <OnlineResource xlink:href="http://prime.esri.com:80/wmsconnector/com.esri.wms.Esrimap/GeoDRM?" xlink:type="simple"
    xmlns:xlink="http://www.w3.org/1999/xlink" />
+ <ContactInformation>
  <Fees>none</Fees>
- <AccessConstraints>
- <wsp:PolicyAttachment wsu:id="BreakingTheGlas" xmlns:geodrm="urn:ogc:ows4:geodrm:licensing"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:xlink="http://www.w3.org/1999/xlink">
- <wsp:AppliesTo>
- <wsa:EndpointReference>
  <wsa:Address>http://prime.esri.com/wmsconnector/com.esri.wms.Esrimap</wsa:Address>
  - <wsa:ReferenceProperties>
    <geodrm:Product id="WMS-PRIME" />
  </wsa:ReferenceProperties>
  </wsa:EndpointReference>
  </wsp:AppliesTo>
- <wsp:Policy wsu:id="IdentityPrecondition">
- <wsse:RelatedService wsse:ServiceType="wsse:ServiceIP">
  - <wsa:EndpointReference>
    <wsa:Address>http://hammer.do.isst.fhg.de/auth/services/auth</wsa:Address>
  </wsa:EndpointReference>
  </wsse:RelatedService>
- <wsse:SecurityToken wsp:Usage="wsp:Required">
  <wsse:TokenType>SAMLAssertion</wsse:TokenType>
  </wsse:SecurityToken>
  </wsp:Policy>
- <wsp:Policy wsu:id="LicensePrecondition">
- <wsse:RelatedService wsse:ServiceType="wsse:ServiceSTS">
  - <wsa:EndpointReference>
    <wsa:Address>http://212.124.44.170:9090/licensebroker</wsa:Address>
  </wsa:EndpointReference>
  </wsse:RelatedService>
- <geodrm:LicenseToken wsp:Usage="wsp:Required">
  <wsse:TokenType>SAMLAssertion</wsse:TokenType>
  </geodrm:LicenseToken>
  </wsp:Policy>
  </wsp:PolicyAttachment>
</AccessConstraints>
</Service>
+ <Capability>
  </WMT_MS_Capabilities>
</soapenv:Body>
</soapenv:Envelope>

```

9 Technical Workflows

This chapter contains the base interactions workflows that were identified in the OWS-4 GeoDRM initiative. Those are:

- Information about GeoDRM specific preconditions that apply to a GeoDRM enabled OWS service
- Authentication and retrieval of identity tokens
- License issuing and retrieval of license reference tokens as well as resolving of full licenses
- Authorization of service requests/responses against grants included in a license
- GeoDRM client interaction with the gatekeeper (GeoDRM enablement for OWS services)

There may be other workflows as well like manual license establishment, commercial licenses, payments, etc. But those are out of scope for this document.

9.1 Information and Interpretation of Preconditions

Every interaction with an OWS service requires the OWS client to request, fetch and parse the service's capabilities. When dealing with GeoDRM enabled services, the client needs to retrieve and parse GeoDRM specific preconditions as well. Those preconditions may force the client to engage follow-up workflows to full fill those preconditions.

Actors

- OWS Service: The service that is GeoDRM-enabled. It provides OWS capabilities describing the features of the service and its information resources.
- Gatekeeper: The service that protects the OWS. It acts as extension to the OWS service and adds the GeoDRM specific preconditions to the OWS capabilities.
- OWS Client: Plain OWS client without GeoDRM extensions. It uses the GeoDRM client as a client side proxy.
- GeoDRM Client: Client side proxy for the OWS client. The GeoDRM client cares about retrieving and parsing of the Gatekeeper's preconditions and engagement of appropriate follow-up workflows to meet the preconditions (like authentication or license negotiation).

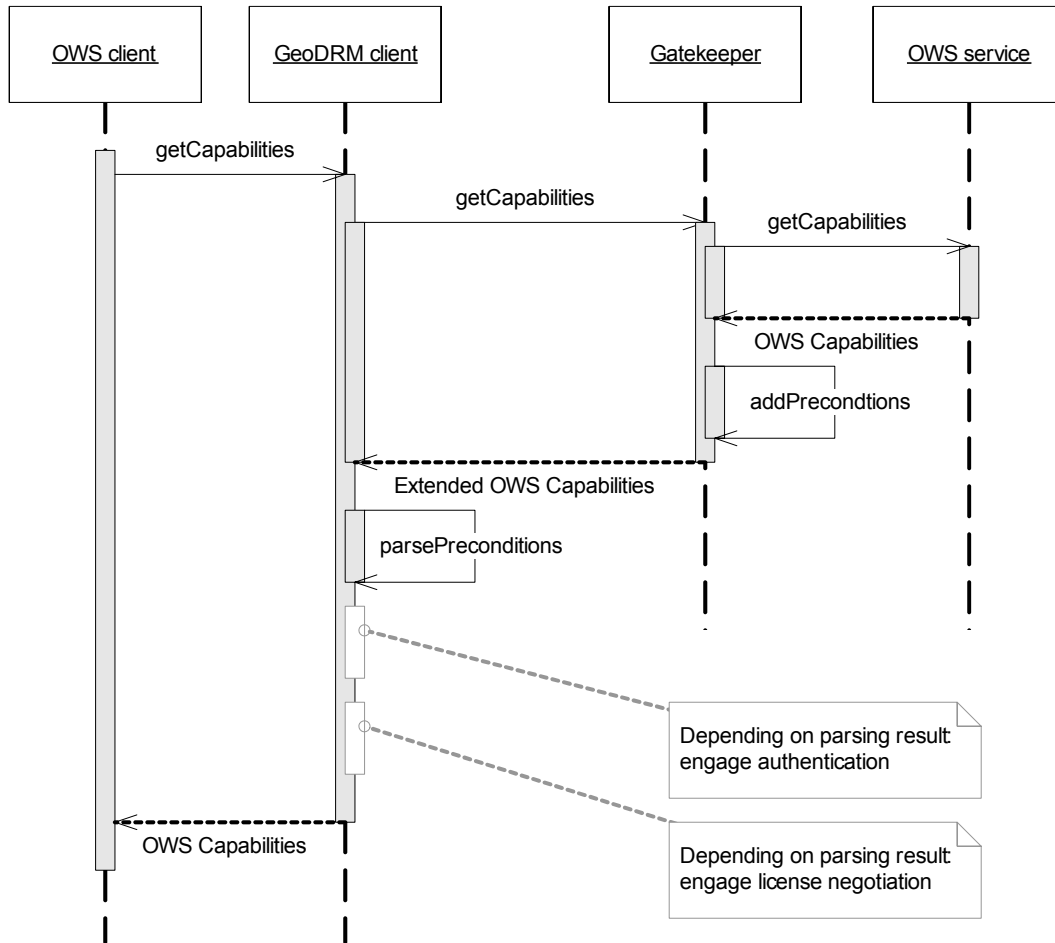


Figure 19 - Sequence: Information and Interpretation of Preconditions

The following paragraph explains the precondition parsing on the client side.

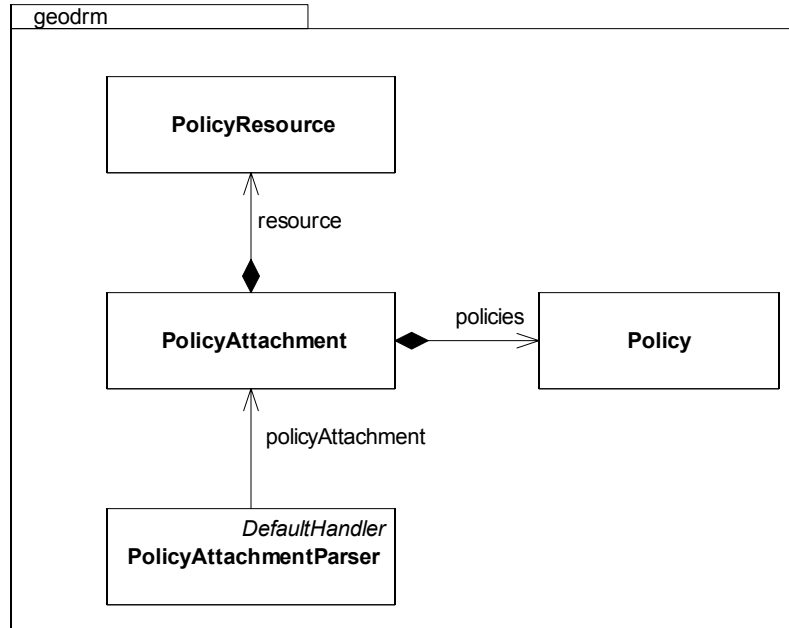


Figure 20- Objects used while parsing GeoDRM elements

The parsing is performed in a two stage process. The first process involves the GeoDRM section of the capabilities being parsed by the GeoDRM API. That results in the objects shown in the figure above. These objects, which reflect the GeoDRM element structure, are expected in the capabilities document.

The policy object has an associated URL and the GeoDRM service type. The client can determine which service can be connected for the retrieval of a valid token and which token type. The client also deduces what protocol should be used to retrieve the token (SOAP, HTTP,...). Each policy will also protect certain resources in the policy attachment. The current implementation assumes supports only a single policy in the capabilities covering.

9.2 Authentication

If the GeoDRM enabled client evaluates the preconditions, which are necessary for the usage of the protected service, the user has to obtain an account at the AuthenticationService (if the account does not exist already). The creation and verification of user accounts by an identity provider and the enablement of trust between the Authentication Service (identity provider) and the GeoDRM gatekeeper (service provider) is out-of-bands in this description.

After a user got authenticated by an Authentication Service, his GeoDRM client receives identity information for further proceeding. The received identity artifact has to be attached to each request, e.g. by using [WS-S] [SAMLToken]. If the identity information is signed by a trusted party [XMLSig] the GeoDRM Gatekeeper itself is able to decide whether the request was submitted by an authenticated subject or not. In case of receiving

an identity reference the GeoDRM Gatekeeper has to call back to an appropriate identity provider (Authentication Service) for authentication.

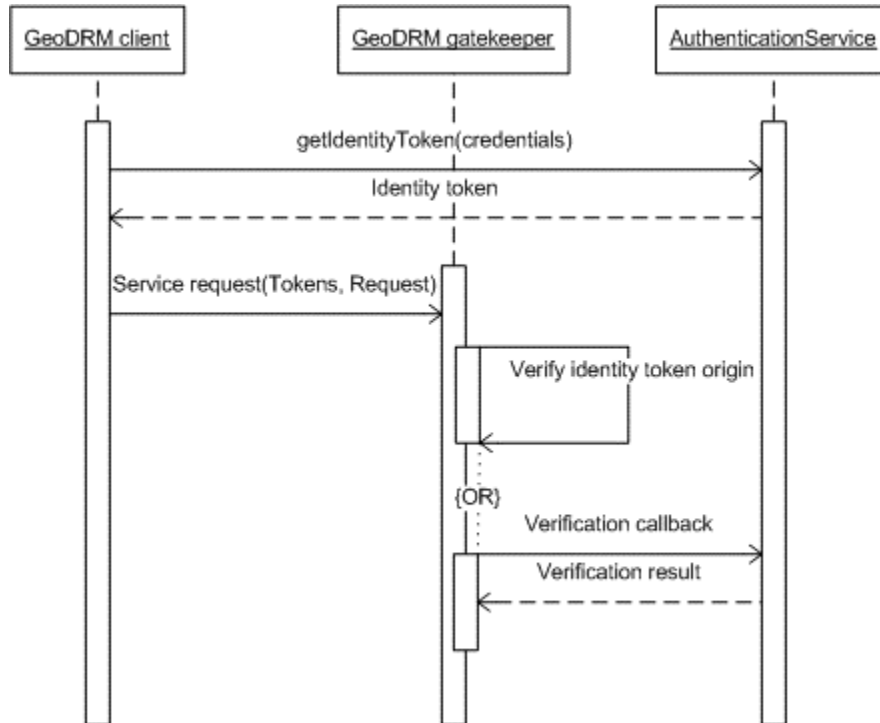


Figure 21 - Sequence - Authentication

9.3 License (issuing/negotiation, resolving)

This chapter will provide a description of the two most important workflows for licenses:

- Issuing/negation and establishment of licenses
- Resolving of licenses

Those two workflows are separated because the architectural design foresees that it will provide more flexibility in terms of what encoding is used for the license itself if the majority of the (most dynamical) interactions use standardized references that could point to licenses which are encoded in various rights expression languages.

As the License Broker component was not included in the OWS-4 GeoDRM initiative the appropriate parts of the following workflows have not been implemented. However, a simple License Broker Web Application was developed in order to proof the overall concept, but not in detail.

9.3.1 License issuing/negotiation

The GeoDRM Client may receive a precondition that expresses the need of establishing a contract with a license in order to use an OWS Service. This precondition includes information about the type and issuing authority of the needed license. The client uses this description to contact the authority (here: the License Broker) and engage the negotiation/issuing process.

The License Broker receives the identifier of the product (service usage) that the client needs a license for and engages the negotiation process by providing a license offer. An example is the acceptance of terms-of-use that applies to the product (service) or a payment process. In the case of potential negotiations, optional parameter values may be set by a customer via its client. The terms-of-use case may not have a negotiation only an “accept” or “reject” choice.

After successful negotiation, the customer client concludes the offered contract and the License Broker creates and stores a license at the License Manager. The Manager will act accordingly if the Broker is a trusted party and return a license identifier. For further communication purposes, the broker will request a license reference token from the manager that is valid for the client (customer) that concluded the offer. This license reference token is used by the client while communicating with the gatekeeper’s protected OWS service.

Actors

- License Manager: The License Manager stores a license as a result of the (negotiation) and agreement between the customer and the License Broker.
- License Broker: entity that has permission to create licenses and license references on behalf of the provider.
- GeoDRM Client: entity that contacts a License Broker in order to retrieve a license for a product identified by a product id.

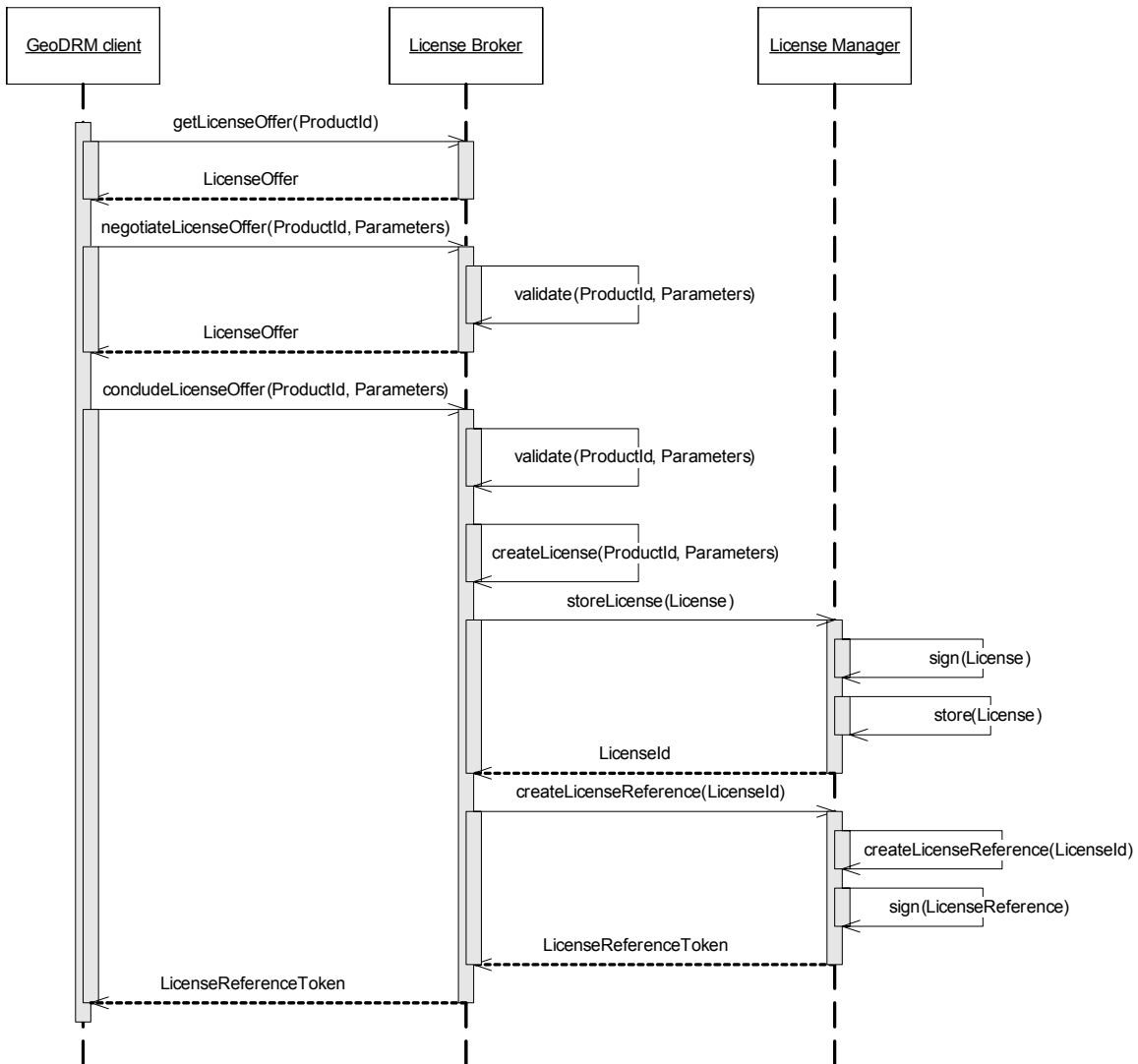


Figure 22 - Sequence: License issuing/negotiation

As soon as a license is stored at the License Manager Service it is in force and part of potential policy evaluations of an Authorization Service. Therefore special trust has to be established between License Broker and License Manager for issuing as well as between Authorization Service and License Manager for resolving of licenses.

9.3.2 License resolving

The GeoDRM Client receives after the agreement a license reference token. The token is only a pointer to the license, which is stored in the License Manager Service. If needed, a trusted party can resolve the reference pointer license and receives a full license.

Those trusted parties may be either the License Broker in order to extend/modify an existing license, an Authorization Service that needs to fetch the license in order to evaluate if a request/response is covered by the grants included in the license or by the client/user itself in order get information about established licenses. Those trust relations with the License Manager should have a long-term nature for the License Broker and Authorization Service as all three share a static business relation. However the trust relation with the client (customer) should support short-term interactions: all customers are allowed to verify and therefore retrieve their licenses. Client authentication is needed to establish those trusted relations. WS-S and common identity credentials could be used for this.

Actors

- License Manager: The License Manager stores the license and resolves it via a license reference token.
- Trusted party: entity that has permission to resolve license references

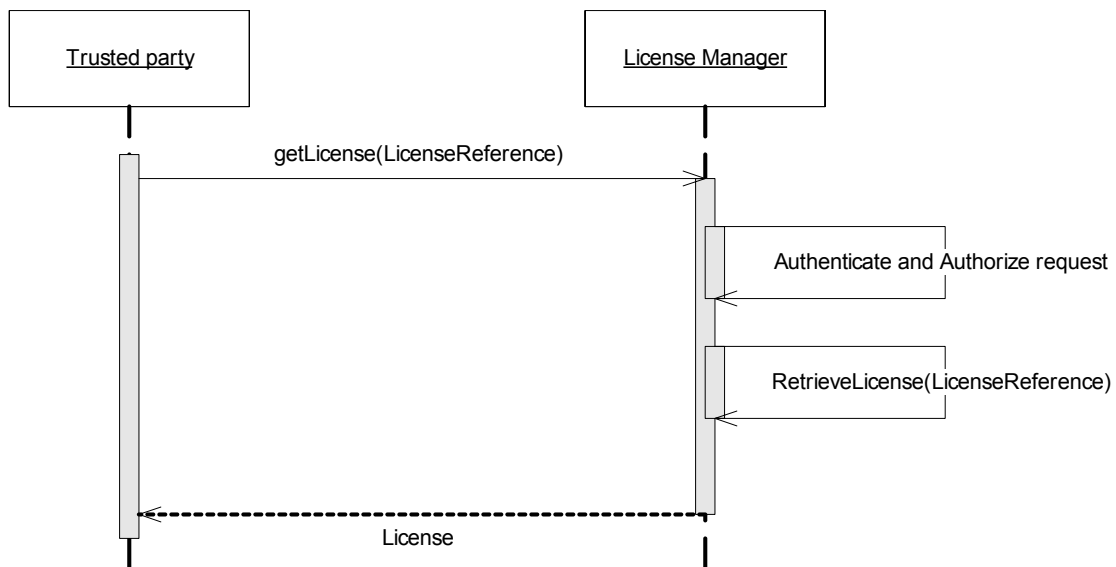


Figure 23 - Sequence: License resolving

9.4 Gatekeeper-Client Communication

A GeoDRM Client calls for the capabilities as a first interoperable operation in order to establish a connection with a gatekeeper protected service. The Gatekeeper acts like an OWS due to this nature as a façade and offers the capabilities document. Therefore the gatekeeper calls the OWS capabilities document and enhances it with GeoDRM specific information (“preconditions” see chapter 8.3). The enhanced document is parsed and found by the GeoDRM enabled client. According to the preconditions the client has to engage the appropriate workflows to either issue a license, authenticate or both. Both workflows will result in tokens that the client stores locally and add to all subsequent interactions with the Gatekeeper/OWS service. The Gatekeeper will proxy all requests after:

- Extraction and validation of the provided tokens (this may include a check, if a trust relation exists for the License Broker or Authentication Service)
- Decomposition of the SOAP header and body information into a request target in terms of “resources”, “subject” and “actions”.
- Authorization check whether the request is covered by the grants derived from the license using the request target.

Actors:

- GeoDRM enabled Client (may be integrated in OWS Client)
- Gatekeeper
- OWS Service
- (License Broker, Authentication and Authorization Service)

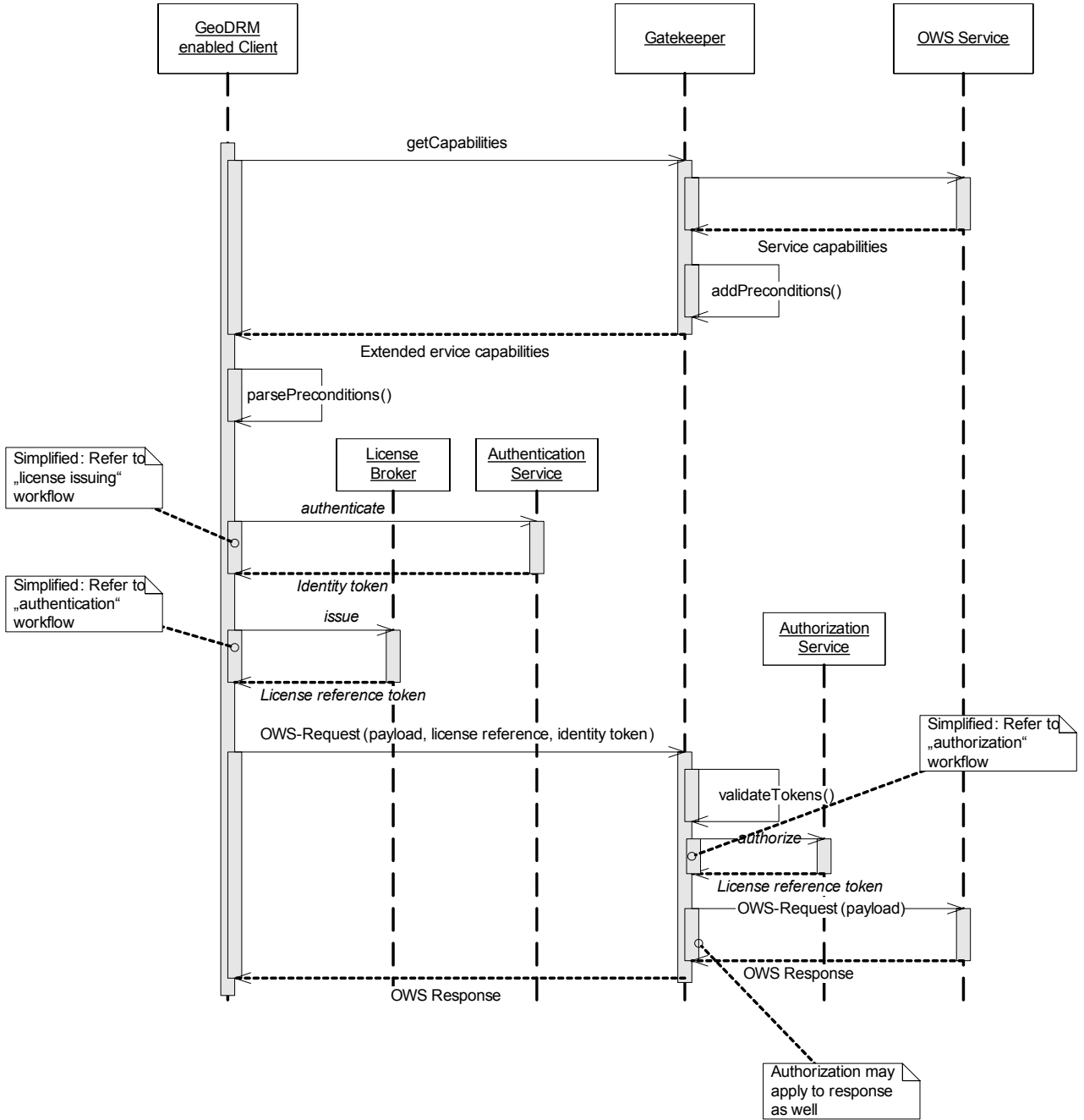


Figure 24 - Sequence: Gatekeeper-Client Communication

9.5 Authorization

The Authorization Service decides always on a given rule set, which may be included in a license. If the license was issued for a certain user or group of users, which means it is explicitly including a party, the Authorization Service also may have to check if the requesting user is the user or part of the user group referenced in the license. Therefore user identity has to be ensured, meaning either the identity token is qualified to fulfil that (e.g. signed SAML Authentication Assertion) or the Authorization Service will have to request the Authentication Service in order to obtain user identity information by passing an identity reference provided by the user in the service request. If no license is referenced in a client's request the Authorization Service could request all licenses that are relevant for the given request/subject tuple (=resource/action/subject) from the License Manager.

Actors:

- GeoDRM Gatekeeper: will contact the Authorization Service in order to evaluate if the request is covered by appropriate grants
- Authorization Service: will fetch the license from the License Manager, extracts the included grants and performs the access control decision. In some cases it may request additional user assertions from an Authentication Service.
- Authentication Service: provides additional user assertions if required
- License Manager: provides licenses to the Authorization Service

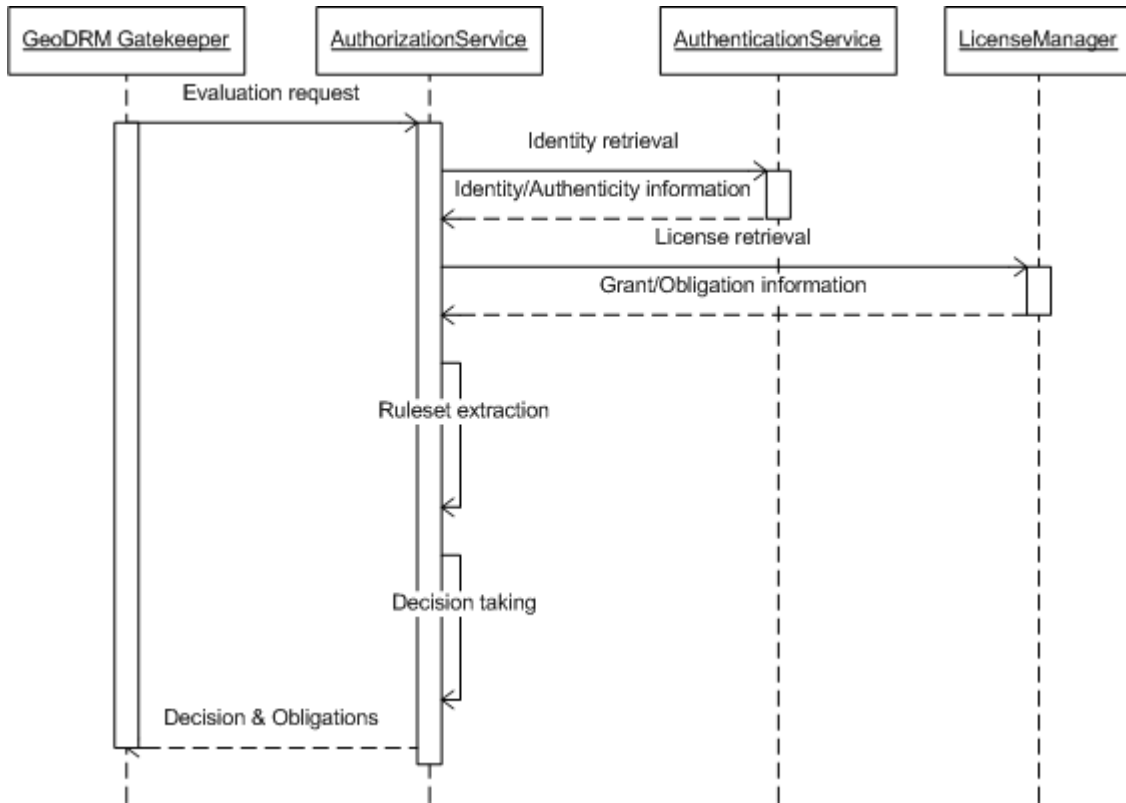


Figure 25 - Sequence: Authorization

10 Distribution and Deployment model

10.1 GeoDRM-enabled Infrastructure

As part of an (spatial data) infrastructure, the components of the OWS-4 GeoDRM architecture need to be distributable in a network. Those distributed components need to link each other either at runtime or in advance. Chapter 10.2 gives an idea about various deployment scenarios that determine the linkage of components and therefore the distribution transparencies of the system. The OWS-4 GeoDRM proposes an architecture that should be capable of supporting the short and mid-term relevant scenarios and is extensible to support future ones and additional requirements.

To achieve this, it is necessary to implement the components on the basis of a highly flexible and extensible infrastructure that supports most of the functionality as common “it-standards” without using too much “geo specific” extensions.

The former widely used HTTP GET/POST protocol is not capable to provide the required flexibility, extensibility or industry support for the new sharing functionalities introduced by GeoDRM. The group found that SOAP, WS-Security and related specifications and standards provide the necessary functionality. It is very likely that the major parts of interface and encoding could be done as “profiling” of those existing specifications instead of forging new ones. The following standards were used as a base for the GeoDRM enabled infrastructure:

- SOAP as overall communication protocol
- WS-S as transport protocol for identity and license reference tokens
- XML-Signature for signing purposes
- XACML as encoding for grants included in a license
- SAML as encoding for identity tokens, license reference tokens and as protocol for the online authentication
- WS-Policy, WS-PolicyAttachments, WS-Addressing and WS-SecurityPolicies for encoding of gatekeeper preconditions

In order to support “non-SOAP” OGC specifications and implementations, both client side proxy (GeoDRM enabled client) and the server side proxy (gatekeeper) are able to transform the transport protocol from HTTP to SOAP and vice versa. All other communication as well as the internal GeoDRM enabled client Gatekeeper communication is fully SOAP based and therefore capable to utilize all advantages that WS-Security and relative it standards provide.

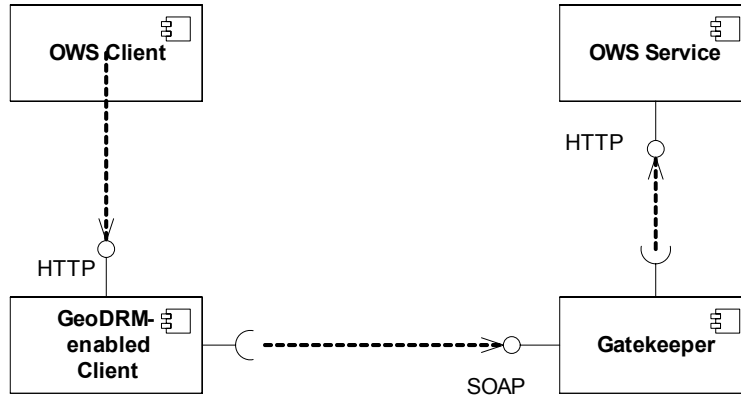


Figure 26 - Support “non-soap” OGC clients/services

Due to some disadvantages such an approach may serve only as short/mid-term solution used only until OGC specifications are upgraded to SOAP. The major disadvantage is that the service endpoint for the viewpoint of the OWS client is not the OWS service or its surrounding gatekeeper, it’s another client component. This imposes a whole set of difficulties including metadata entries for services, need to rewrite capability url’s, dynamic client-side url’s in WMC documents etc.

10.2 Deployment Scenarios

The OWS-4 GeoDRM architecture allows setting up and operating of various deployment scenarios. A concrete deployment scenario depends on the applied operation/business model and the used sharing functions. Some operation models may use only a single sharing function, e.g. identity as a precondition.

A fully distributed deployment with independent organizations supporting in each case only a single business (sharing) function, a separated establishment and management and an independent SDI agency results in a large number of organizations. The number can be augmented by introducing parallel instances, e.g. multiple Promoters or Deliveries. Each organization needs to be described with a separated linkage address together with the provided functions and responsibilities.

The aim of this paragraph shows some examples to illustrate the wide range of possible deployments and to point out a few important prerequisites that these scenarios impose. It may point out future requirements that need to be elaborated further by the GeoDRM WG, Security WG and Price and Order Processing WG.

10.2.1 Single-Operator Scenario

In the simplest deployment scenario all services run at the provider. He acts as provider, agent and owner as one legal entity to the client. The following deployment model shows the component distribution for this scenario.

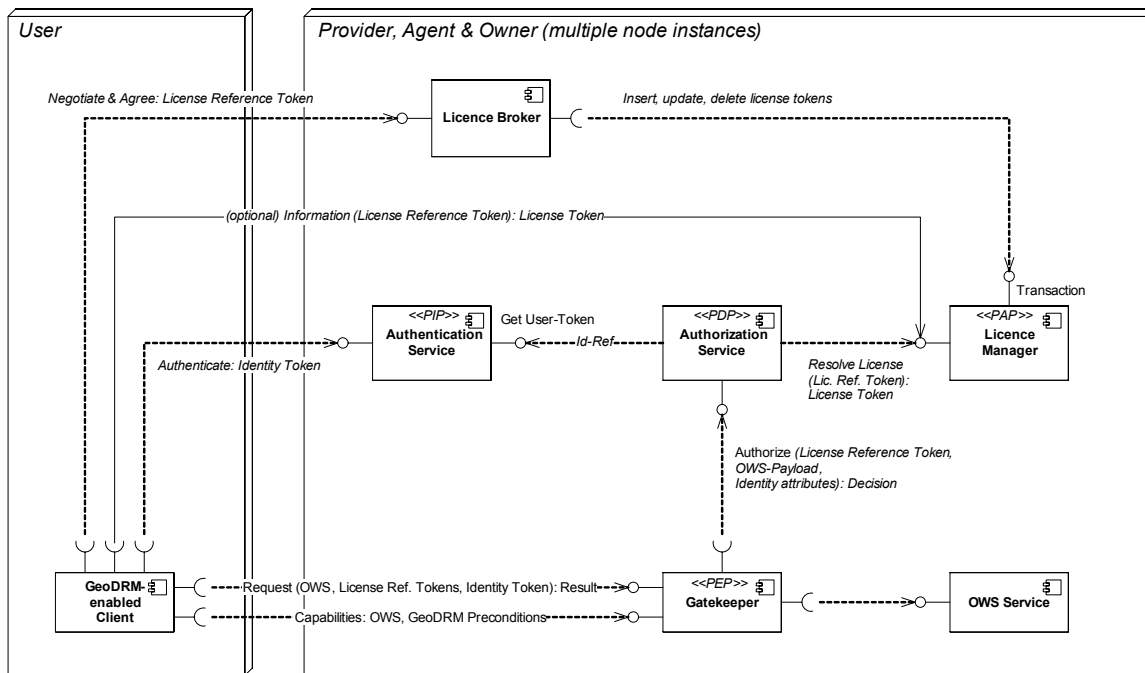


Figure 27 - Deployment: Single-Operator Scenario

The deployment model shows that the GeoDRM client component is distributed and able to communicate to multiple distributed “fully featured” GeoDRM service nodes. This demonstrates the huge importance of using simple, interoperable tokens and standards for those interface communications that involve dynamic and only at runtime known components. Those interfaces have a very priority to be standardized or profiled; these are (ordered):

- Interface of the Gatekeeper used by the GeoDRM enabled client
- Interface of the Authentication Service used by the GeoDRM enabled client (not needed if one wants to go for an out-of-the-band approach to acquire tokens and if they contain all information like certificates)
- Interface of the License Manager used by Authorization Service and License Broker
- Interface of the License Broker used by the GeoDRM enabled client

All other interfaces could be assumed as of secondary priority because they are (at least in this scenario) static and the components know each other a priori. However, even the secondary interfaces could be prioritized to support different business/operation models as show in the following scenarios.

Note: this difference of importance of interfaces is one of the main reasons why the OWS-4 GeoDRM team decided to use license reference tokens encoded as SAML and therefore fully compliant to the used WS-S standards instead of directly transporting the full featured license content as token (which would imply the need to provide various implementation specifications instead of one simple for the references).

10.2.2 Single Sharing: Access control only scenario

This deployment scenarios shows how parts of the GeoDRM components and its architecture could be used for operate authentication/authorization (“access control only”) scenarios without caring about license at all.

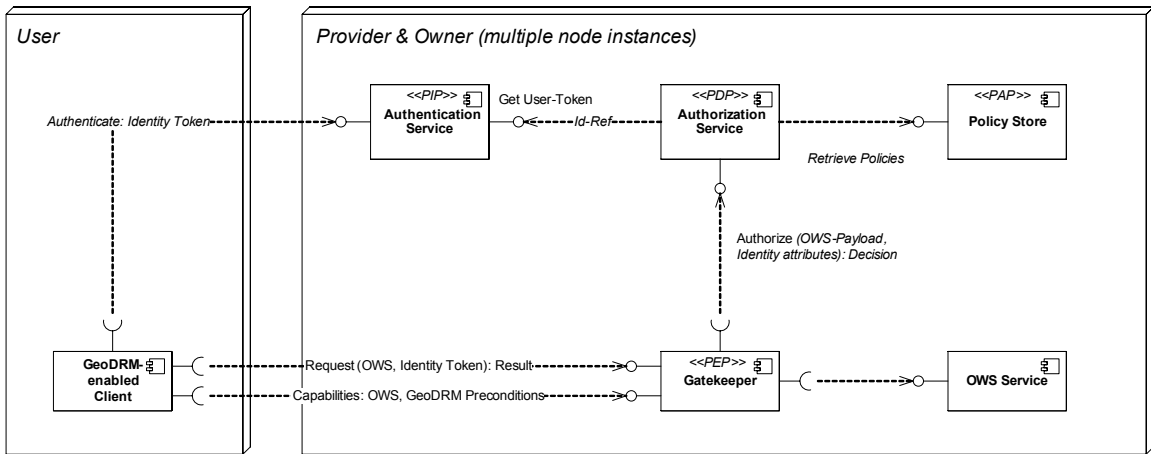


Figure 28 - Deployment: Single Sharing: Access control only scenario

In such a scenario the gatekeeper will provide authentication preconditions only. An authentication may be necessary (if authentication is done online) but it may be left out as well if the usage of offline authentication facilities like certificates is sufficient.

In difference to a scenario that includes the sharing functionality “licensing” the Authorization Service accesses policies (grants) that were predefined and stored in a simple Policy Store (like a database).

10.2.3 Single Sharing: Anonymous License Click-through scenario

This deployment is similar to the previous one. The difference is that the only support sharing functionality is “licensing”. Again, such a scenario could be build with the proposed GeoDRM architecture and its components.

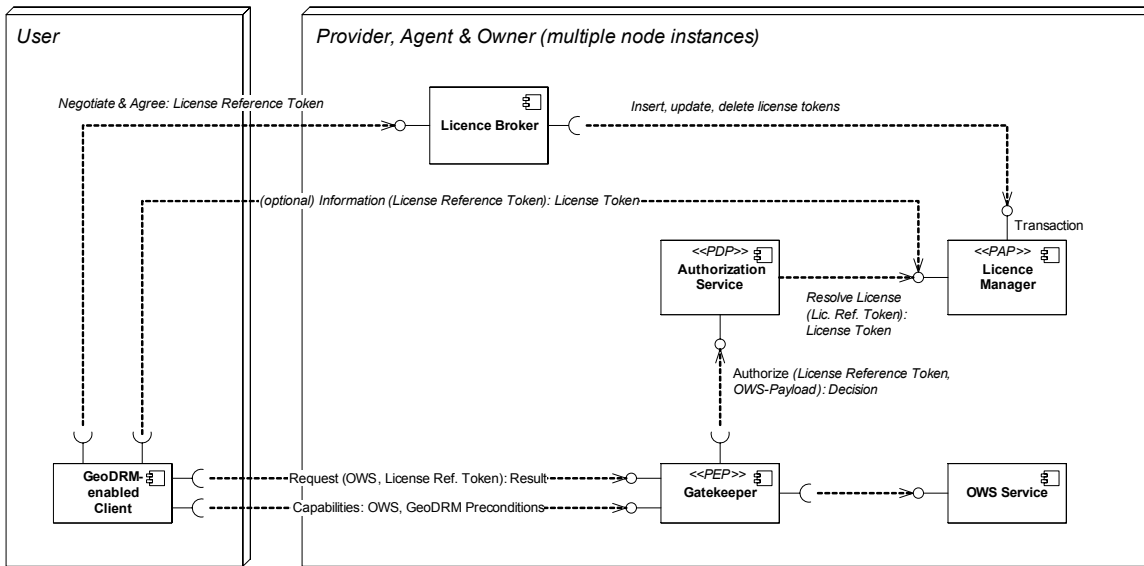


Figure 29 - Deployment: Anonymous License Click-through scenario

In such a scenario the Gatekeeper will provide license preconditions only. Those licenses could only be of type “anonymous”, e.g. simple terms-of-use acceptance for a certain session or client IP-address.

10.2.4 Federation scenario

This scenario separates business functionality between different roles or actors. It separates the Authentication Service from the rest of the system. This is a typical approach in other infrastructures to support federations in which a single authentication service provides identities for multiple service providers.

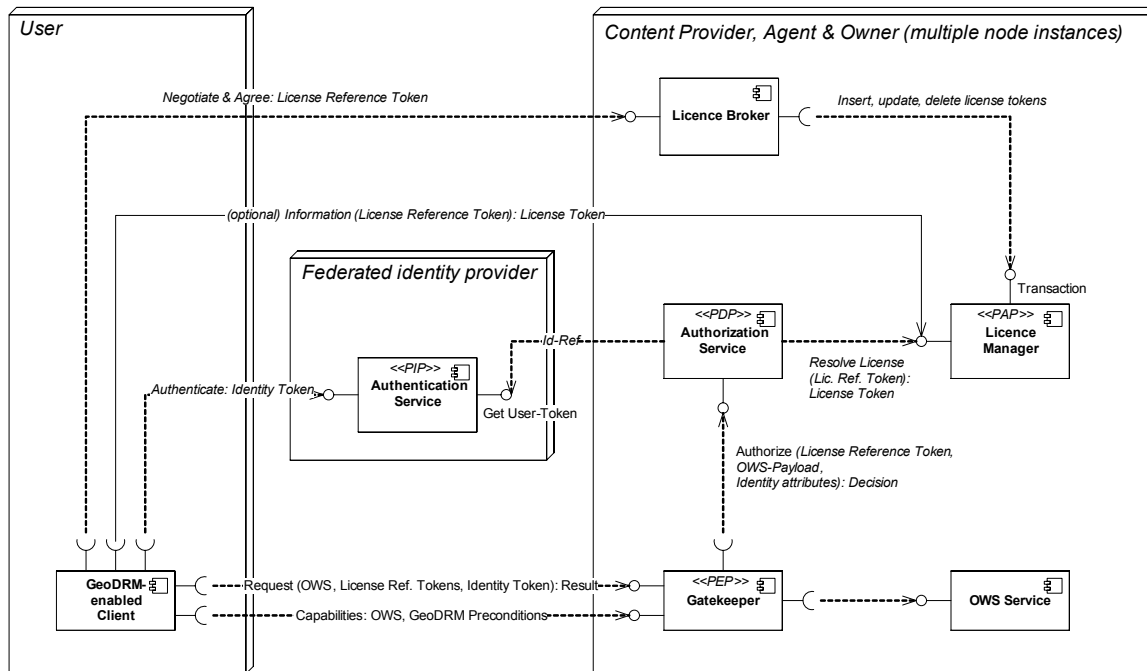


Figure 30 - Deployment: Federation scenario

Because the business functions are separated between different nodes, a trusted relation has to be established between the Content Provider and the Authentication Service. The Content Provider has to trust identities that were issued by a central authentication service and the authentication service has to trust the authorization in cases where the provider needs additional information about a user. As a result communication between Authorization and Authentication needs to be secured:

- Providers objective: Integrity of Identity (with regards to origin – trusted Authentication – and tampering during communication) → Use of signed Identity
- Users objective (maybe also providers): Privacy of Identity → Encrypt Identity

Please note, that the interface of the Authentication Service used by the Authorization Service is optional. This could be used if further information about the requesting client is needed in order to perform the authorization decision.

10.2.5 Distributor scenario

In this deployment scenario, the License Broker is separated and acts as distributor for a single provider.

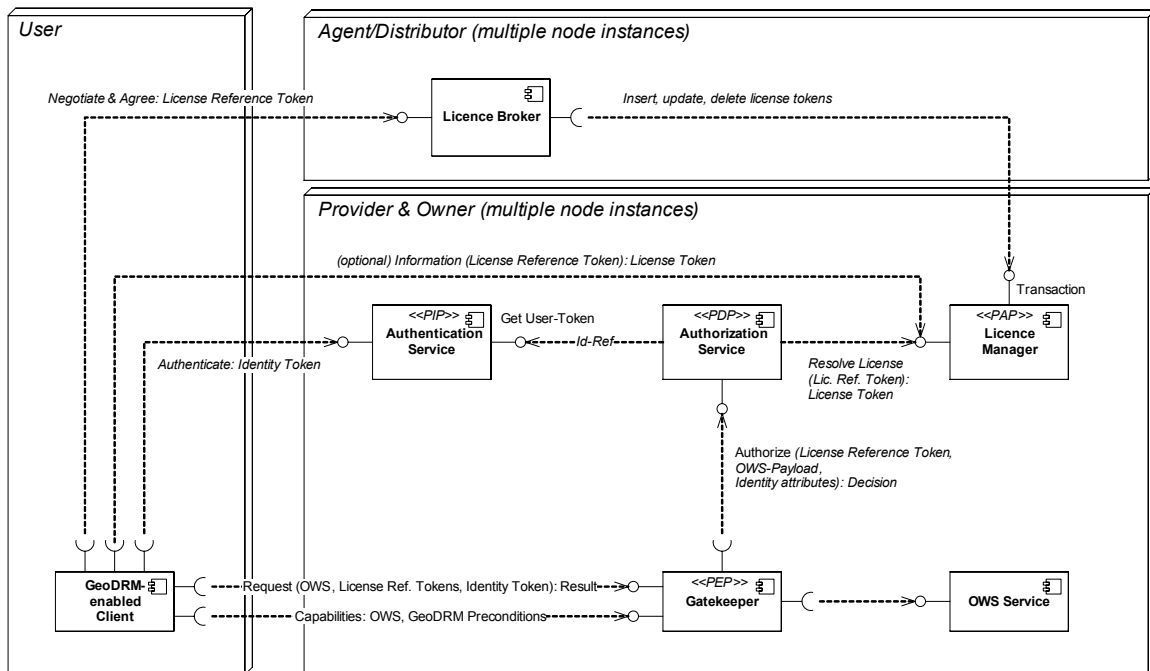


Figure 31 - Deployment: Distributor scenario

Again business functions are separated between different nodes and a trusted relation has to be established between the agent and provider/owner. Such a distribution would allow competition between different brokers as they “resell” licenses on behalf of the provider whilst the provider stays in complete control about all established licenses. This enables the provider to modify or revoke license at any time.

To enable such a scenario, additional information is needed to be exchanged:

- There is a need to exchange information about the applicable licenses (license offers), that the agent could negotiate and that the provider/owner will accept
- There is a need to define which License Manager the License Broker should use (and maybe also trust between both)

Exchange of those additional information pieces may be done static in a preconfigured way or dynamically at runtime.

11 Demo Application Scenarios

This chapter describes the demo application scenarios that were shown as part of the OWS-4 demo. The demo application scenarios were built utilizing the architecture, infrastructure, client and service components and information models described in this document. They full fill the requirements derived from the use cases listed in chapter 1.

11.1 Application Scenario “Feature Updates”

11.1.1 Relation to Use Cases

The demonstration of the UniBW implementation is based on the GeoDRM use case #4, which is illustrated in chapter 6.4 “Use Case #4: WFS-T Feature Updater”.

11.1.2 Application Context

Please refer to chapter 6.4 “Use Case #4: WFS-T Feature Updater” for a description of the application context.

11.1.3 Deployment and Configuration

The architecture for this application scenario is illustrated in the figure below.

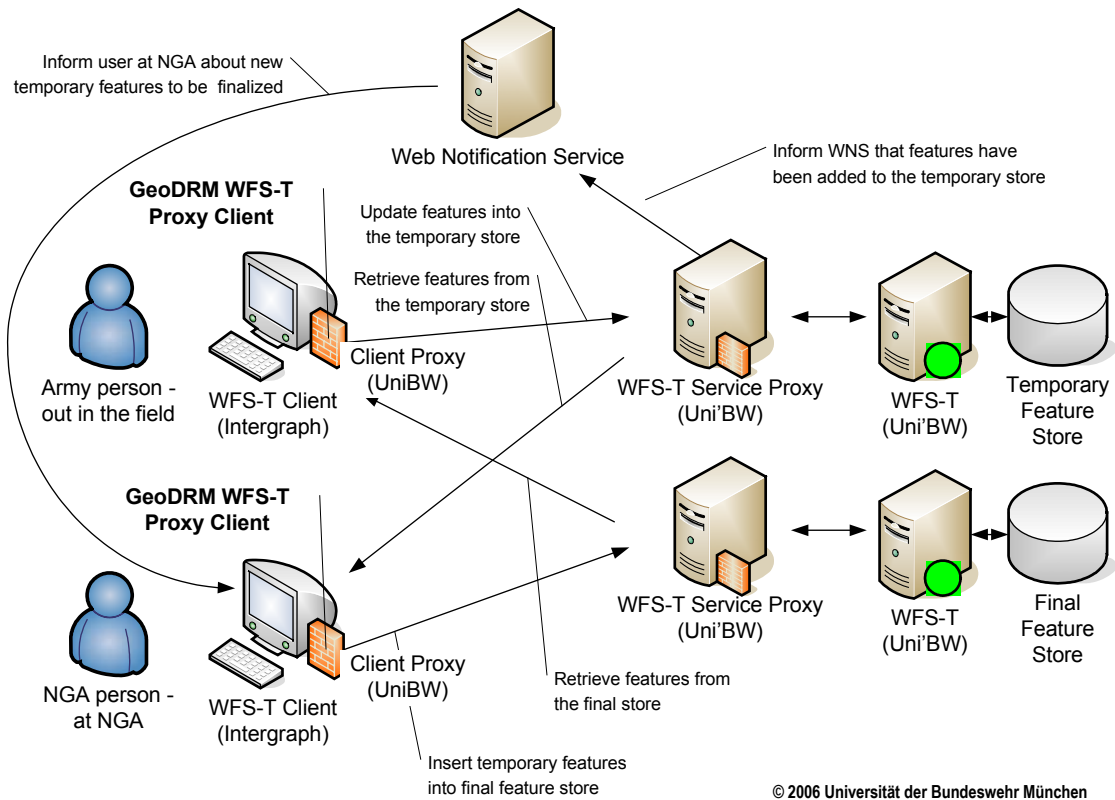


Figure 32 - Architecture for GeoDRM enabled Services and Clients, according to use case #4

In order to demonstrate the GeoDRM enablement for an “off-the-shelf” WFS-T Client and a WFS Service the proxy approach has been used. In order to enable GeoDRM functionality on the service side, a WFS-T Service Proxy was implemented. It accepts SOAP messages, containing identity and license tokens (as described in the Trusted Geoservices IPR) and functions as a GeoDRM Gatekeeper (as described in the GeoDRM RM). In a similar fashion, the GeoDRM Client Proxy accepts the regular WFS-T requests from the client and creates SOAP messages, which are send to the Service Proxy. In addition to the Client and Service Proxy, the UniBW also provided the WFS-T, hosting the MSD3 features.

During the work on site, the attendees of the UniBW (Andreas Matheus and Cristian Opincaru) integrated the Client Proxy and final testing in cooperation with the attendees from the Intergraph Corporation. Also, the final rights licensed to the NGA Officer and the Field Engineer, where adapted to fit best the given scenario. For more information about the UniBW contribution, please visit our web site at

<http://iisdemo.informatik.unibw-muenchen.de/ows4/>.

11.1.3.1 Licensed Rights

The licensed rights are as follows

AnySubject -- These permissions are granted to all subjects

This enables a simple navigation in the data without seeing the airport features

Operation / Action	Feature Type	Area
GetCapabilities	N/A	N/A
DescribeFeatureType	N/A	N/A
GetFeature / GetFeature	ows4:Road_L	N/A
GetFeature / GetFeature	ows4:River_L	N/A

NGA-Officer – Additional permissions

This enables the Analyst to see the airport features

Operation / Action	Feature Type	Area
GetFeature / GetFeature	ows4:Aerodrome_A	N/A
GetFeature / GetFeature	ows4:Helipad_P	N/A
GetFeature / GetFeature	ows4:Taxiway_A	N/A
GetFeature / GetFeature	ows4:Aircraft_Hangar_A	N/A
GetFeature / GetFeature	ows4:Runway_A	N/A
GetFeature / GetFeature	ows4:Apron_A	N/A

This enables the Analyst to create new and update/delete existing features

Operation / Action	Feature Type	Area
Transaction / Insert, Update, Delete	ows4:Helipad_P	N/A
Transaction / Insert, Update, Delete	ows4:Taxiway_A	N/A
Transaction / Insert, Update, Delete	ows4:Runway_A	N/A

Field-Engineer -- Additional permissions

This enables the Analyst to see the airport features

Operation / Action	Feature Type	Area
GetFeature / GetFeature	ows4:Aerodrome_A	N/A
GetFeature / GetFeature	ows4:Helipad_P	N/A
GetFeature / GetFeature	ows4:Taxiway_A	N/A
GetFeature / GetFeature	ows4:Aircraft_Hangar_A	N/A
GetFeature / GetFeature	ows4:Runway_A	N/A
GetFeature / GetFeature	ows4:Apron_A	N/A

This enables the Analyst to create new and update existing features

Operation / Action	Feature Type	Area
Transaction / Insert	ows4:Helipad_P	WITHIN A1 (see figure below)
Transaction / Update	ows4:Helipad_P	N/A
Transaction / Update	ows4:Taxiway_A	N/A
Transaction / Update	ows4:Runway_A	N/A

Geometry based access restrictions

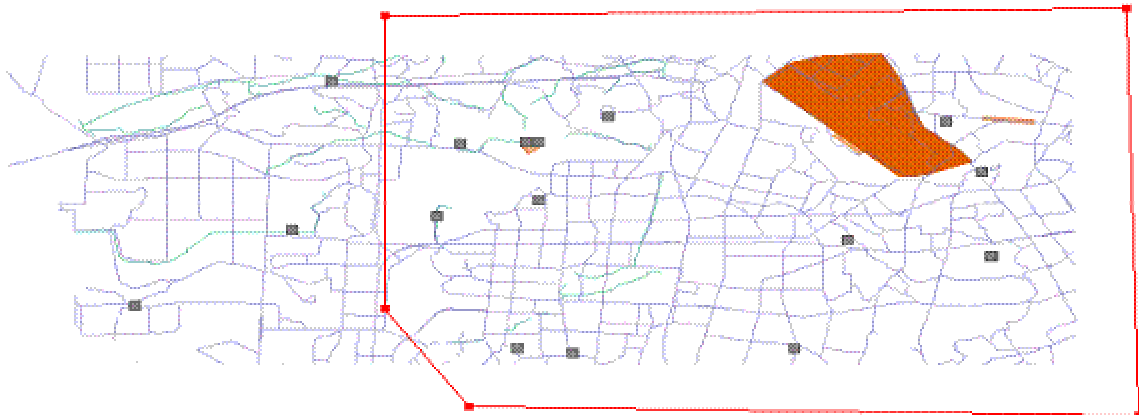


Figure 33 - Access Control Area A1 for modification of helipad feature information

Coordinates for A1

(-74.96789132745889, 39.383275615837945), (-74.96789132745889, 39.296675134185634), (-74.94733464747071, 39.268245683138154), (-74.78331858373527, 39.265621426118386), (-74.78638021692498, 39.38546249668775), (-74.96789132745889, 39.383275615837945), (-74.96789132745889, 39.383275615837945), (-74.96789132745889, 39.383275615837945)

The licensed rights for the NGA-Officer – using a GeoXACML Policy encoding – can be obtained at the following address

http://iisdemo.informatik.unibw-muenchen.de/ows4/GeoPDP/msd3itc/service?Request=GetPolicy&PolicySetId=LICENSE_ID_1

The licensed rights for the Field-Engineer – using a GeoXACML Policy encoding – can be obtained at the following address

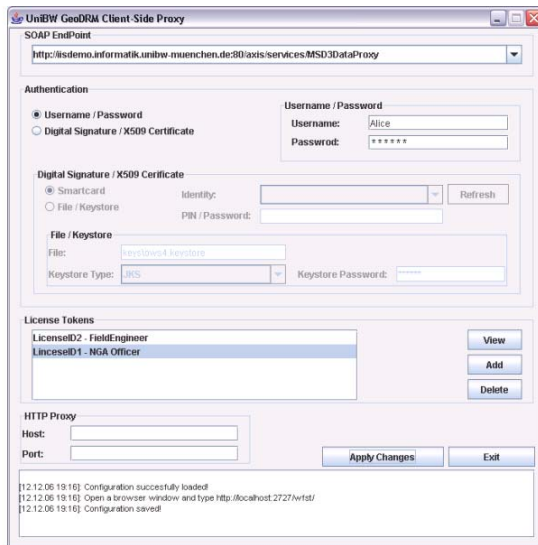
http://iisdemo.informatik.unibw-muenchen.de/ows4/GeoPDP/msd3itc/service?Request=GetPolicy&PolicySetId=LICENSE_ID_2

11.1.4 Walk-Through

The final demonstration at the *Port Authority of New York and New Jersey (PANYNJ)* was presented by Stan Tillman from Intergraph. A demo movie (with sound), which is representative for the real demo, can be found online at

<http://iisdemo.informatik.unibw-muenchen.de/ows4/movie/OWS4%20-%20GeoDRM.html>

In the following, the screenshots from that movie are illustrated and explained.



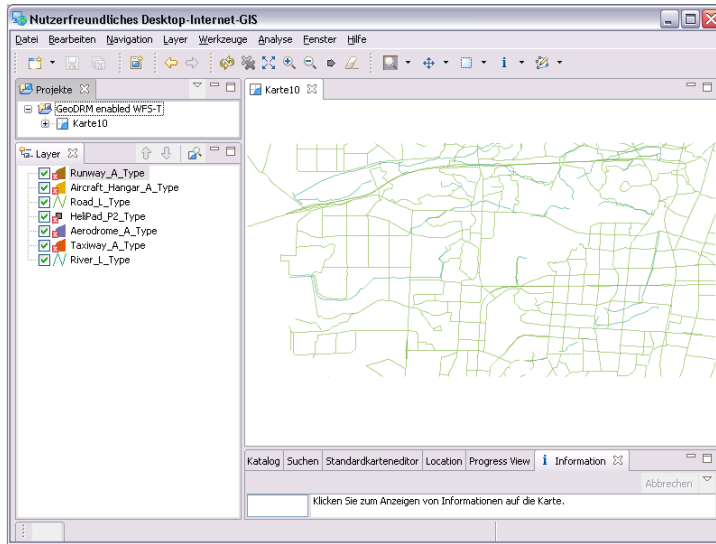
In the following the OWS-4 GeoDRM team will demonstrate the usage of the GeoDRM components using the uDIG open source client. In this scenario, two users have different permissions:

- The NGA Officer has unrestricted rights to access and modify the whole dataset
- The Field Engineer has restricted access in the area around the airport (in red), as illustrated in the figure above.

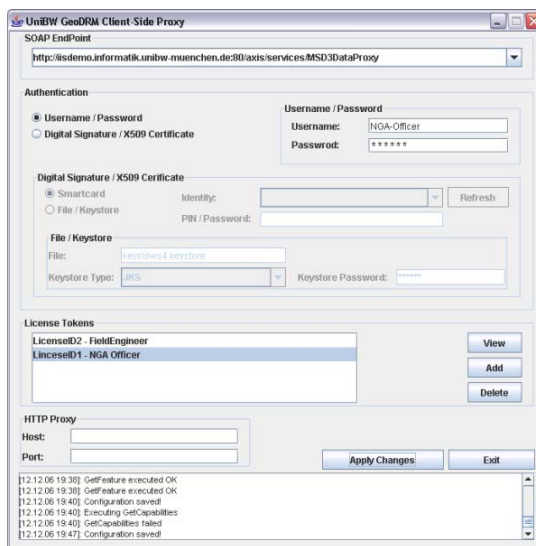


The figure shows an unsuccessful login. The username "Alice" was logged.

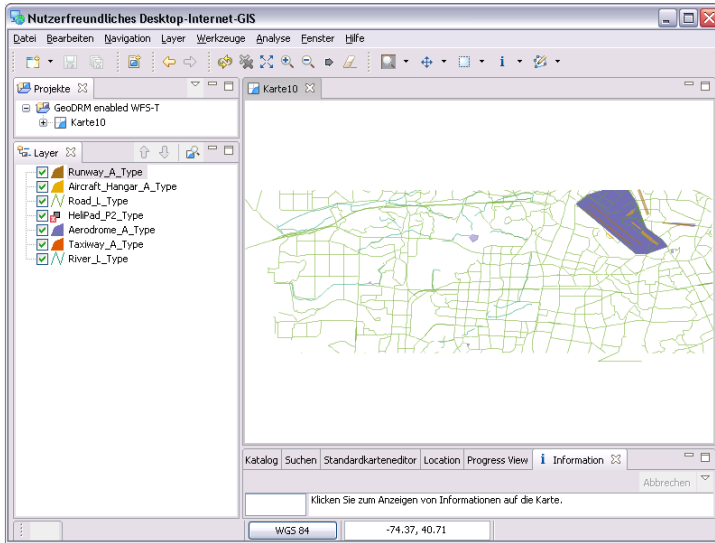
As expected, because the server does not know this user, it throws an XML exception stating that the provided credentials could not be verified.



As an unauthorized user (Any Subject) the unprotected feature types are shown.

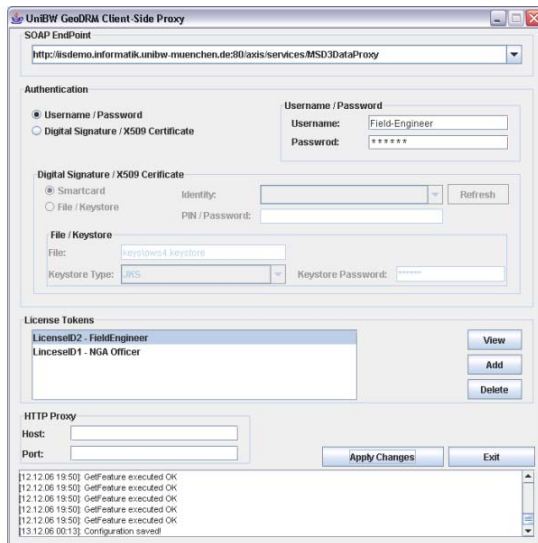


Then the user logs in as the NGA Officer. As mentioned earlier this user has unrestricted permissions over the whole data set. Therefore he is able to see the whole dataset.



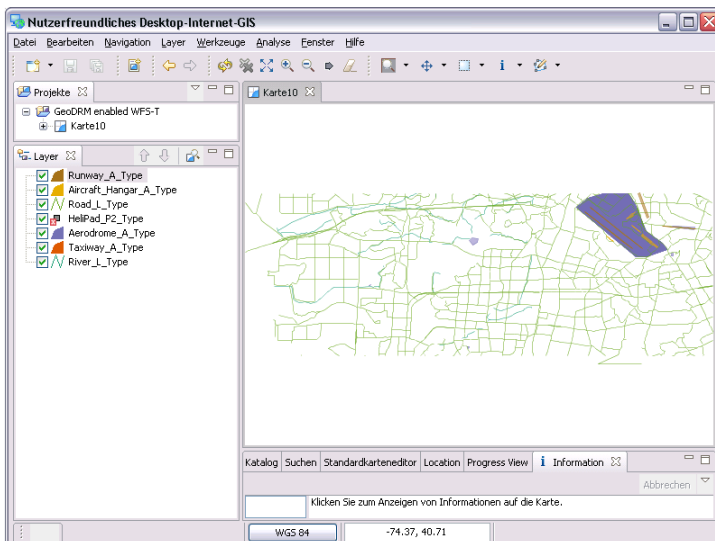
In the next case the log in uses the Field Engineer role. As previously mentioned, the Field Engineer has only restricted access. Please notice that the client is not displaying the

Helipad_P2 feature type (see the red x near the feature type name). The Field-Engineer is only allowed to see this layer in the area around the airport. A Zoom into the airport area will display the layer correctly.

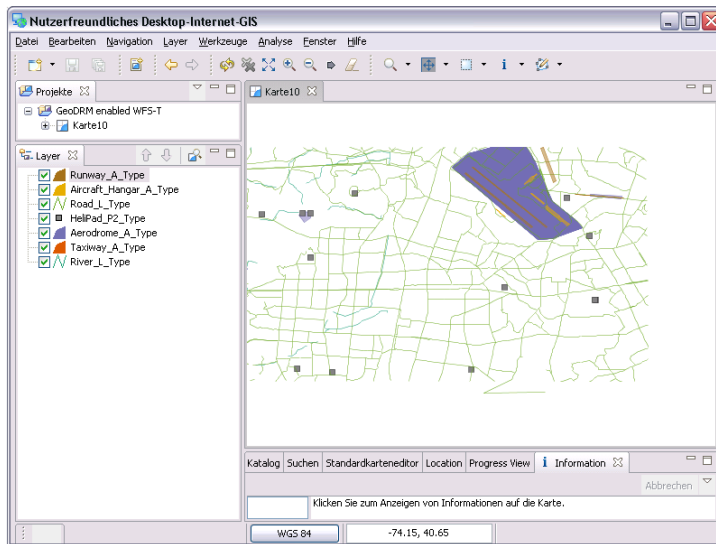


This figure demonstrates how GeoDRM can protect a Transactional WFS. After all, it is very important that the data be written / deleted only by the authorized persons.

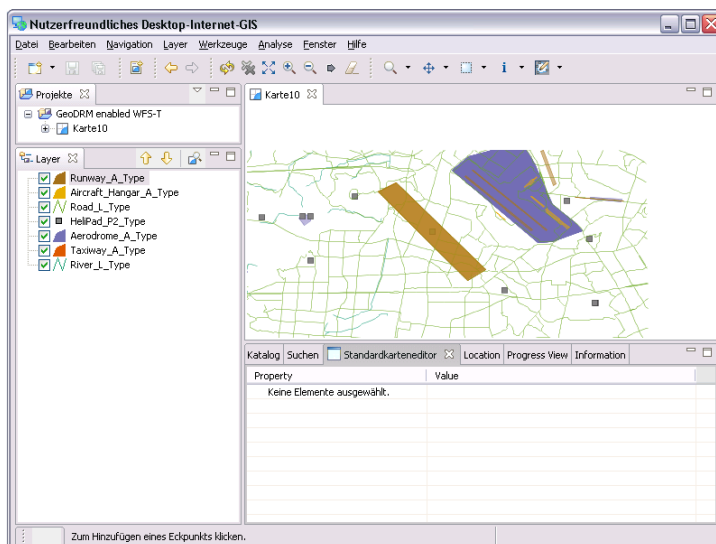
The login uses the Field-Engineer role.



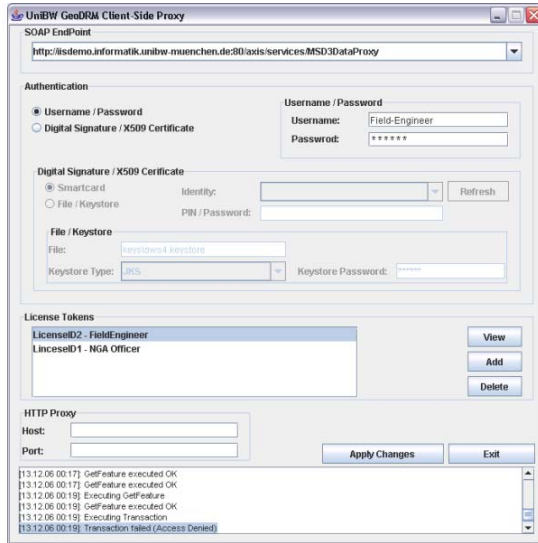
As Field-Engineer, the feature type Helipad is not shown, because the user does not have the rights.



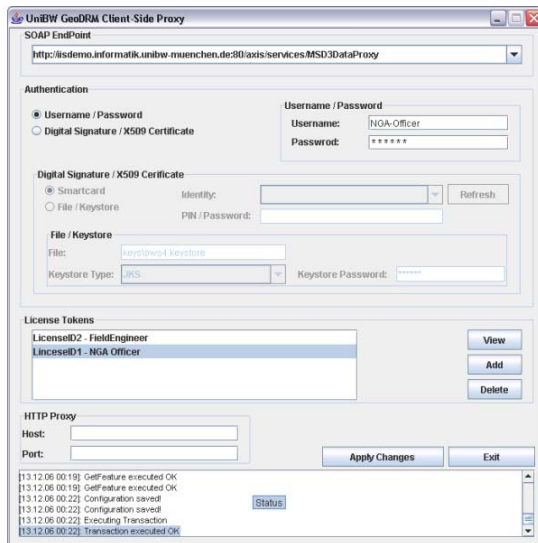
After zooming into the area around the airport, the Helipads are displayed, according to the licensed rights.



The field-engineer try's to insert a new runway. Because this user does not have the permissions to introduce new features, the transaction will fail. Notice the Access Denied message in the log of the GeoDRM Proxy.



Access Deny message for the previous request to insert a Runway.



After logging in a as NGA-Officer the insert of the created Runway succeeds.

11.2 Application Scenario “Breaking the glass”

11.2.1 Relation to Use Cases from the RFQ

The context of this application scenario is the requirements analyses derived from use cases #1 and #3. Both use case are from an engineering point of view very similar and do only differ in the license and the formulated preconditions. Both use cases require that the user is required to establishes a license in order to use a by GeoDRM means protected OWS service resource.

Use case 1 depicts a click-through scenario, where the user has to agree to a terms-of-use statement in order to receive a license, which is valid only for that session. Therefore the use case requires no explicit authentication of the user in order to establish a license. The binding of the subject to the license is done through IP address matching instead of users credentials. The Gatekeeper has to provide a single precondition to the user that expresses the need to provide a license, which could be obtained from a particular License Broker.

Use Case #3 shows an impersonate licenses with grants that apply to explicitly identified individual users. This use case requires therefore the authentication of a user and end user licenses that are bound to a particular identity. The Gatekeeper has to provide two preconditions to:

- Provide an identity token which could be obtained from a particular Authentication Service.
- Provide a license reference token which could be obtained from a particular License Broker.

Use case #3 includes additional constraints that are part of the license and that have to be met.

Use case #2 is a distributor use case where the user/client of the system is a cascading OWS service instead of a client and where the license negotiation is done in advance to the use case itself (predefined license). As the License Manager Service is able to provide persistent licenses as well as transient ones this use case does not differ from a technical perspective.

11.2.2 Application Context

The context chosen for this application is called “breaking the glas”. Background for this is a scenario in which a GeoDRM protected service is available only in emergency case. If the conditions for an emergency case are given, the service is usable without restrictions e.g. for early responders.

The scenario deals with a WMS service providing information about archaeological sites and endangered species. In case of a wild fire, early responders should be allowed to use

the service. To identify if a situation is an emergency situation (wild fire), the user has to read an appropriate terms-of-use statement. Agreement to that statement means “yes, this is an emergency situation”. In case of misuse, the user acts illegal is can be punished.

The demo implemented the strong identity binding required by use case #3, but the alternative (weak identity binding through IP address matching) has been tested as well, appropriate preconditions, license offers and policies have been created.

11.2.3 Deployment and Configuration

The following figure shows the deployment of the software components for the application demo “Breaking the glass”.

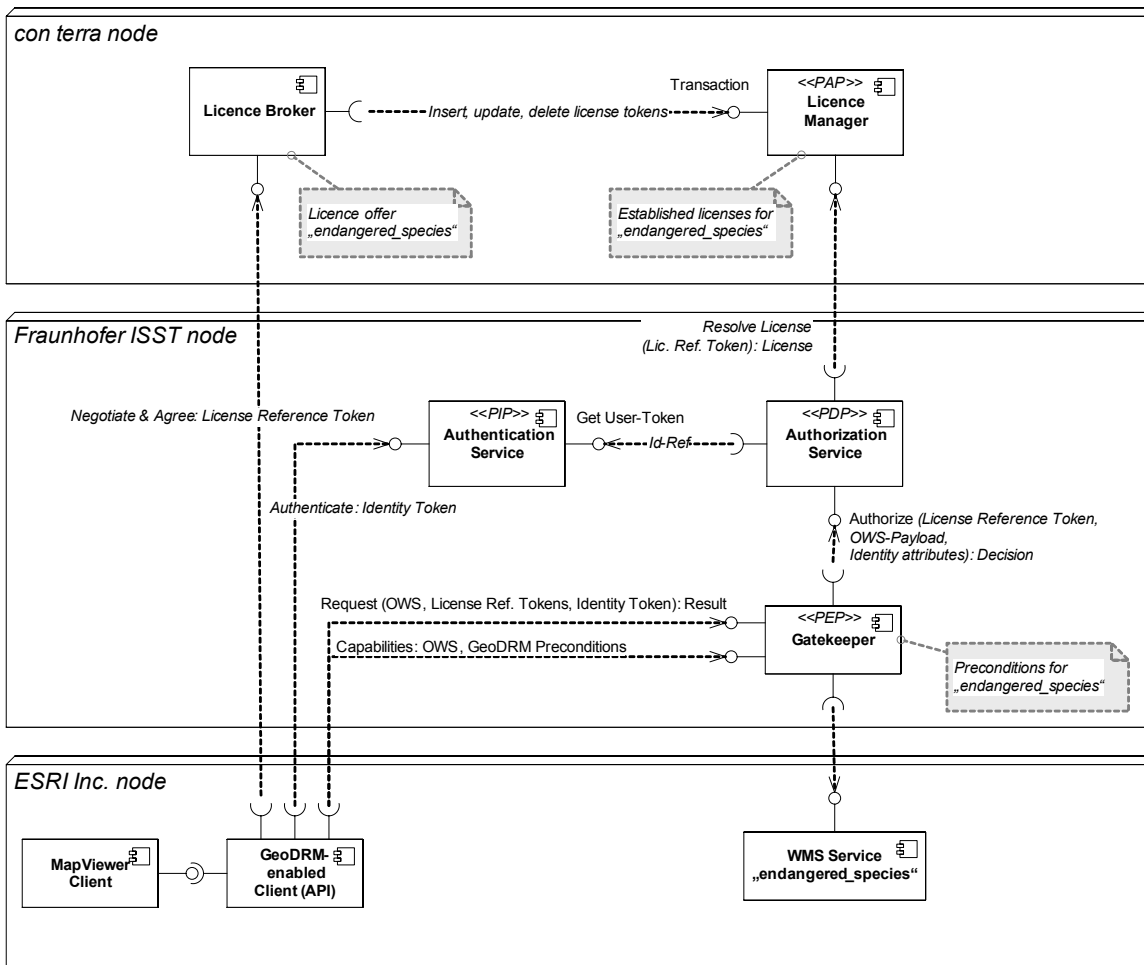


Figure 34 - Deployment: Application Scenario “Breaking the glass”

Only configuration details for the GeoDRM components are listed here. WMS service and client are out-of-scope.

11.2.3.1 Gatekeeper Preconditions

For the gatekeeper being a central component in the demo application scenario preconditions in configuration have to be met.

- Enablement of trust between Gatekeeper and Authentication Service, which can be done by importing the appropriate certificates containing public keys.
- Determination of the Authorization Service to be used by the Gatekeeper.
- Enablement of OWS request analysis by the Gatekeeper for the identification of resources and actions requested by the GeoDRM enabled client.
- Building a capabilities attachment/amendment according to the terms of service of the secured OWS for propagation of the preconditions/obligations plus modification of the capabilities information.

```
<wsp:PolicyAttachment wsu:id="BreakingTheGlas"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:geodrm="urn:ogc:ows4:geodrm:licensing">
  <wsp:AppliesTo>
    <wsa:EndpointReference>
      <wsa:Address>http://prime.esri.com/wms/EndangeredSpecies</wsa:
Address>
      <wsa:ReferenceProperties>
        <geodrm:Product id="WMS-Prime" />
      </wsa:ReferenceProperties>
    </wsa:EndpointReference>
  </wsp:AppliesTo>

  <wsp:Policy wsu:id="IdentityPrecondition">
    <wsse:RelatedService wsse:ServiceType="wsse:ServiceIP">
      <wsa:EndpointReference>
        <wsa:Address>http://auth.idservice.fhg.de/services/auth</wsa:A
ddress>
      </wsa:EndpointReference>
    </wsse:RelatedService>
    <wsse:SecurityToken wsp:Usage="wsp:Required">
      <wsse:TokenType>SAMLAssertion</wsse:TokenType>
    </wsse:SecurityToken>
  </wsp:Policy>
```

```

    <wsp:Policy wsu:id="LicensePrecondition">
      <wsse:RelatedService wsse:ServiceType="wsse:ServiceSTS">
        <wsa:EndpointReference>
          <wsa:Address>http://212.124.44.170:9090/licensebroker</wsa:Address>
        </wsa:EndpointReference>
      </wsse:RelatedService>
      <geodrm:LicenseToken wsp:Usage="wsp:Required">
        <wsse:TokenType>SAMLAssertion</wsse:TokenType>
      </geodrm:LicenseToken>
    </wsp:Policy>
  </wsp:PolicyAttachment>

```

The given attachment example contains an endpoint reference describing an instance of the GeoDRM product. Furthermore it states that there are two preconditions to be met, in which the first condition is a SecurityToken with encoding SAMLAssertion and the other one the defined LicenseToken (reference) from the GeoDRM context, encoded using SAMLAssertion. So there are two SAM Assertions, which have to be obtained from the referenced services to be presented with the requests.

11.2.3.2 License Offer and Policy-Template

For this demo application scenario, the License Broker is able to provide the client with a license offer for a product named “endangered_species”.

```

<product>
  <name>endangered_species</name>
  <policy-template>endangered_species.xml</policy-template>
  <duration-of-validity>8</duration-of-validity>
  <contracting-type>terms-of-use</contracting-type>
  <legal-text>http://creativecommons.org/licenses/by-nc-sa/2.5/legalcode</legal-text>
  <parameters>
    <parameter>subject-identifier</parameter>
  </parameters>
</product>

```

The offer defines a template for grants that will be included in the license, duration of validity and a contracting type. The OWS-4 implementation of the License Broker web application supports only the contracting type “terms-of-use”. That type defines an additional parameter which represents the terms-of-use statement or its location. This scenario uses a creative commons legal code as example. A single parameter has to be provided by the client that is his subject-identifier. This identifier will be the subject that the grant is given to the licensee. This is a simple implementation, which offers several

points for enhancements and better support for WS-* standards, but it shows the concept of an offer.

The offer refers to a policyset-template that matches to the product and represents the grants that the license could include. The implementation uses just simple templates, but again this shows a potential solution if a more generic grant-creation mechanism is implemented.

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet PolicySetId="342323-43223-12211"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:first-applicable"
xmlns="urn:oasis:names:tc:xacml:1.0:policy">
  <Description>This license allows to perform any action on
any (sub)-resource of the WMS-Service</Description>
  <Target>
    <Subjects>
      <SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string"><!--
subject-identitifier to be placed here--></AttributeValue>
        <SubjectAttributeDesignator
SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </SubjectMatch>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">http://prime.esri.com/wms/angered_species</Attrib
uteValue>
            <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">OGC:WMS</AttributeValue>
```

```

        <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
type" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ResourceMatch>
    </Resource>
</Resources>
<Actions><AnyAction/></Actions>
</Target>
<Policy PolicyId="Unrestricted_usage_policy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
combining-algorithm:first-applicable">
    <Target>
        <Subjects><AnySubject/></Subjects>
        <Resources><AnyResource/></Resources>
        <Actions><AnyAction/></Actions>
    </Target>
    <Rule Effect="Permit" RuleId="Unrestricted">
        <Target>
            <Subjects><AnySubject/></Subjects>
            <Resources><AnyResource/></Resources>
            <Actions><AnyAction/></Actions>
        </Target>
    </Rule>
</Policy>
</PolicySet>

```

The template contains the following properties:

- Resource
 - Type: OGC:WMS
 - Instance: http://prime.esri.com/wms/endangered_species
- Action
 - Any action, as the grant offers unrestricted usage of the named service
- Subject
 - Type: subject-id
 - Instance: will dynamically be added when License Broker creates a new instance of the policysset template

11.2.4 Walk-Through

11.2.4.1 Find and Get Capabilities from GeoDRM protected service

First of all the user inputs the URL of the GeoDRM-enabled service (which points to the Gatekeeper that acts like the GeoDRM enabled service) into a Map Viewer form. This step is manually but the URL could as well be the result of a catalogue query. The GeoDRM enabled client requests the capabilities from the service. The returned capabilities document determines whether authentication (see chapter 12.2.4.2) and/or license negotiation (see chapter 12.2.4.3) occurs.



Figure 35 - Define service endpoint dialog

11.2.4.2 Authentication

After parsing the appropriate precondition that states that an authentication token is required, the user is presented with an authentication screen.

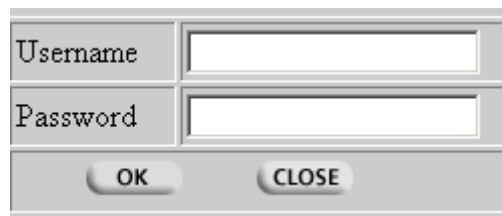


Figure 36 - Authentication dialog

The credentials obtained from the user are presented to the Authentication Service according to the interface described in chapter 8.2.2. The Authentication Service calls back to its identity database. The application scenario identity database was built in a relational database tied to the Authentication Service by JDBC. If the identity information retrieval was successful a SAML Assertion factory was used to create the signed identity token, if it was not the client is presented with an exception message, which the user will be informed about. Continuation of a successful sign on includes handing back the identity token to the authentication client for further proceeding.

11.2.4.3 License negotiation

After parsing the appropriate precondition that states that a license token is required, the GeoDRM enabled client call the issuing authority and requests a contract offer. The user receives it as described in 11.2.3.2. Due to the reason that the License Broker is implemented as web application for demo purposes only, the will be displayed on the clients web browser window. As the contracting type of the offer is “terms-of-use” the statement is automatically loaded and displayed.



Figure 37 - License agreement dialog 1/2

The user has to scroll down and accept the terms-of-use statement.

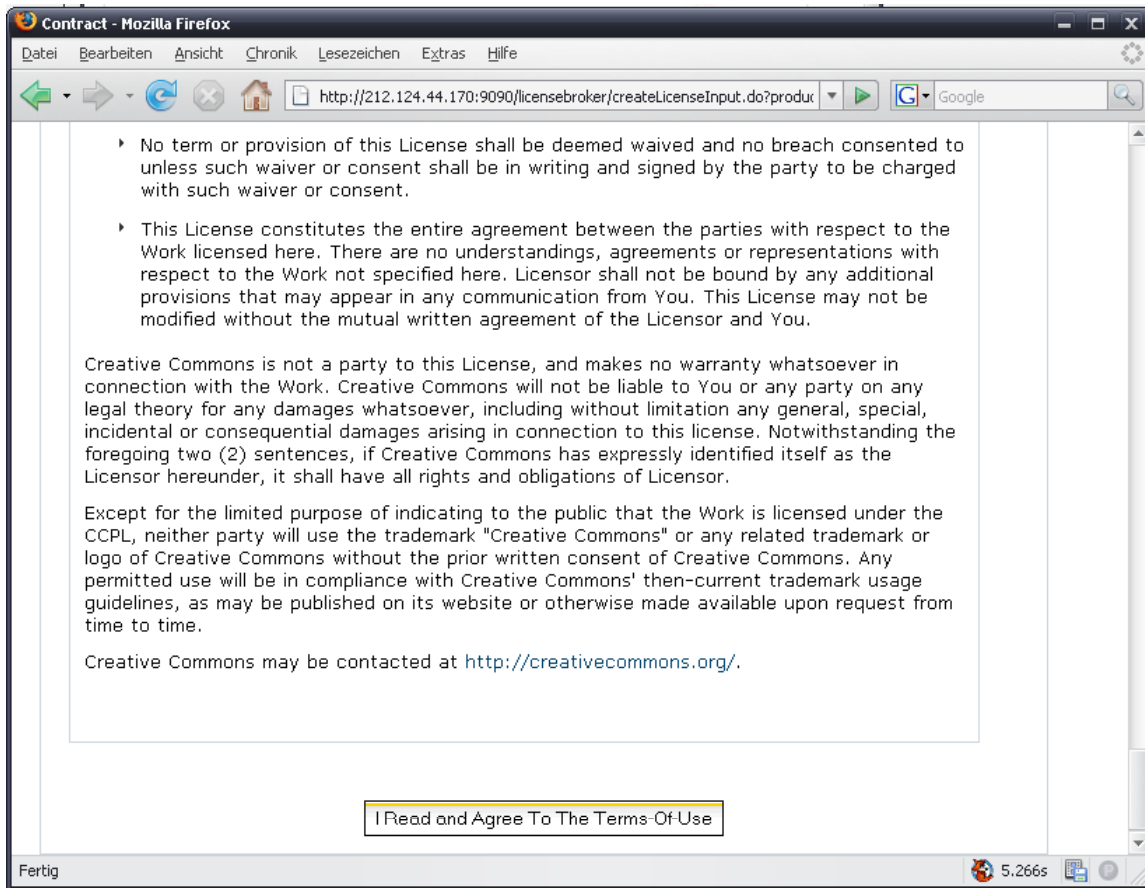


Figure 38 - License agreement dialog 22

Agreement will cause the License Broker to create and store a license as well as creating a license reference token, which is handed back to client (next display is for information purposes and can be switched off).

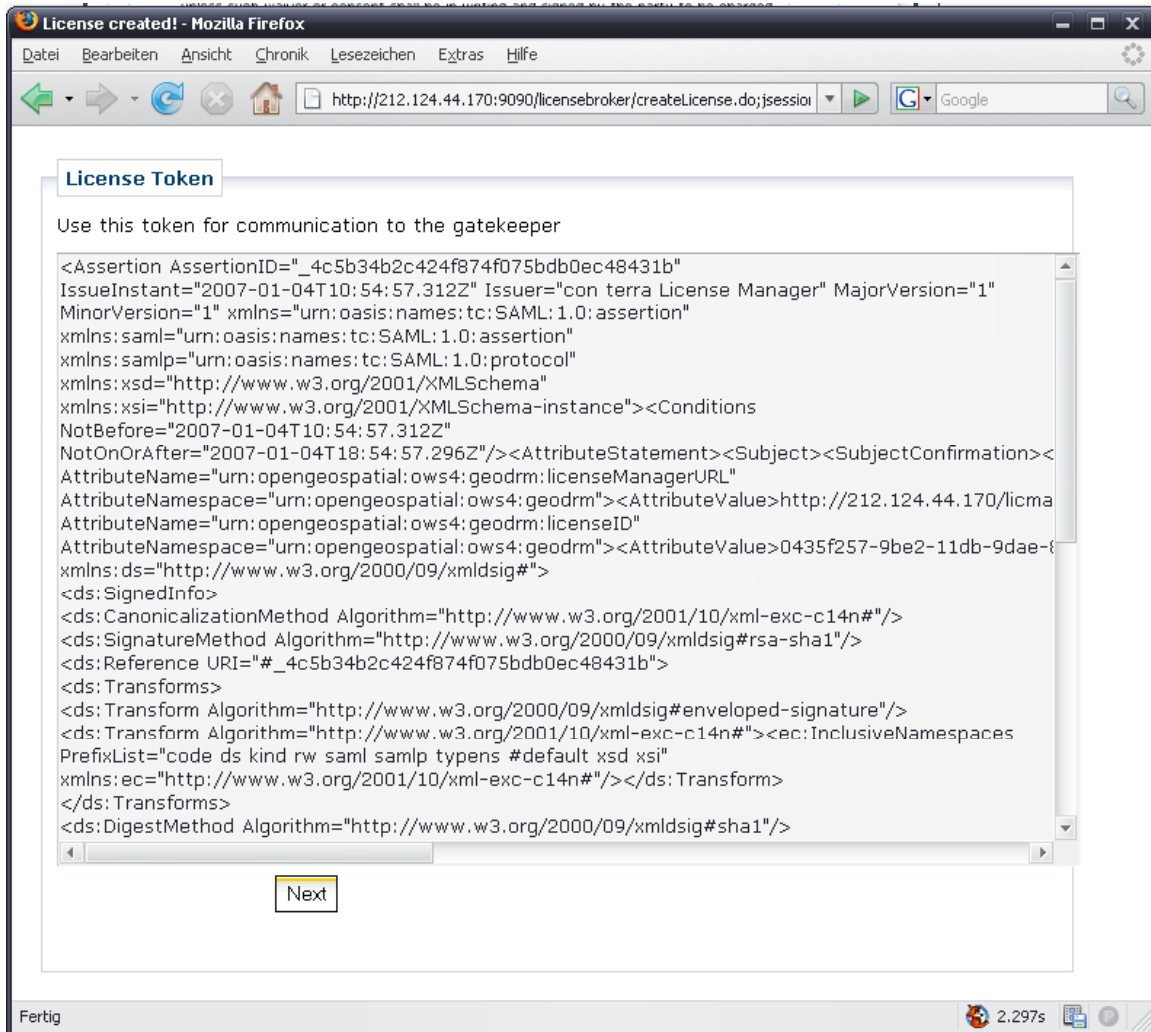


Figure 39 - License reference token hand-over dialog

A click on the “Next” button will hand over the license reference token to the GeoDRM enabled client. This precondition is now met.

11.2.4.4 Usage of the GeoDRM protected service

Once required tokens (authentication or license) are acquired, requests can be made to a GeoDRM-enabled service, it behaves completely like a “normal” WMS. In the example of a WMS service, protected layers may be requested.

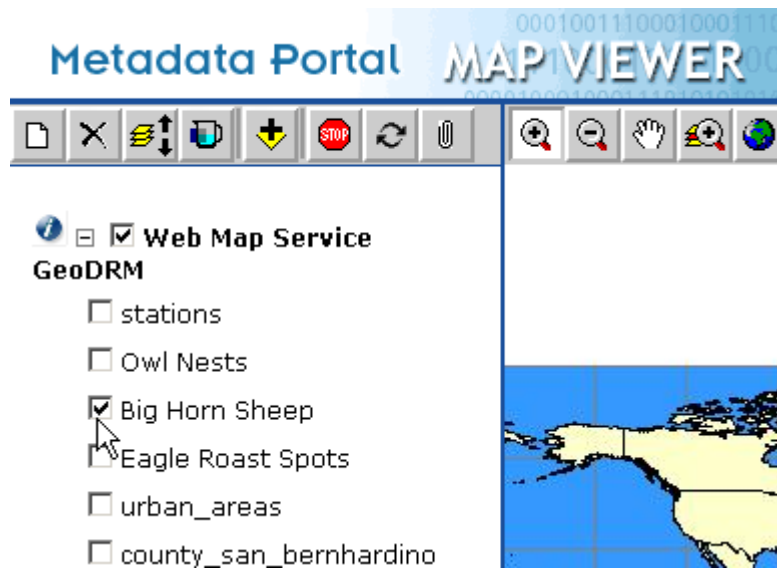


Figure 40 - Service usage

11.2.5 Extension opportunities

Through the generic architecture, every client and service application could be used in such a scenario. The demo was tested with the SPS service from the EOS-1 scenario and the appropriated client application.




12 Informative: Enterprise Business Roles and Processes



The contents of the following chapters may provide valuable feedback and input to the GeoDRM Reference Model as it refines the roles and processes from a business perspective. It is meant to be an informative addition, because the results were not discussed in detail in the OWS-4 GeoDRM team.





12.1 GeoDRM Roles

Change Request to GeoDRM Reference Model

The operation of an SDI node requires the fulfillment of different responsibilities. Therefore roles are introduced to express typical responsibilities. The geoDRM Reference Model introduced already some roles, which are extended in OWS4. Some roles have a legal relationship (e.g. customer - broker) others have only a technical (end-user – delivery) relationship.

Phase	Role Names <small>(# of instances within a SDI node)</small>	Icons	Description	See GeoDRM RM
Publish	SDI Committee (1)		A SDI Committee is responsible for a SDI domain. It operates at least a catalogue and registers SDI domain common conventions.	
	SDI Agency (0..∞)		Executive organisation with acts on behalf the SDI committee.	
	SDI Provider (0..∞)		Primary contract party for Customer. Contracts other roles (promoter, broker, manager and delivery) to set up a business network. Registers it at the SDI committee.	
	IPR Owner (0..∞)		An owner owns intellectual property.	x

Find	End-User (0..∞)		An individual person who accesses and uses a product. An example is an employee of a company who acts on behalf the company.	x
	Promoter (1..∞)		A promoter is an operator of product catalogues with product descriptions, e.g. OGC CS-W and ISO 19115/19139 metadata.	

Procure Manage / Establish	Customer (0..∞)		A customer is a legal entity and acquires rights to access and use offered products.	x
	Broker (0..∞)		An individual person who accesses and uses a product. An example is an employee of a company who acts on behalf the company.	(x)
	Manager (0..∞)		A manager maintains accounts on behalf of the provider and the customer.	(x)
Delivery /bind	Delivery (0..∞)		A deliveryman is responsible for operating data & processing services and their delivery.	(x)

12.2 Business Processes

Subject for Enhancement of the GeoDRM Reference Model.

The purpose of the process model is to refine the business phases of a GeoDRM-enabled system introduced with the geoDRM Reference Model. It is important to note, that the process model whereas still part of the enterprise viewpoint could give some general structuring guides to define the technical system within the engineering viewpoint. The proposed structures have not been implemented in general in the engineering viewpoint. This means, that the validity of the process model is still subject to be proven. This counts especially for the generalized manager and broker components.

Geospatial digital rights management (GeoDRM) aims to automate the *management* of intellectual property rights (IPR) between related institutions. The emphasis on management is necessary, because computer have no legal IPR understanding and can act only on behalf of a legal person. Key components of management are automated business processes between different roles. Rights of use can be transferred from a legal person to another by using contracts. Therefore *electronic contracting* enhances efficiently. After the processes and the needed information models are identified and justified, software components and interfaces can be derived for automated support. Manual and automated business process can be co-existing to reduce software blackouts or to address different user groups differentially in many cases.

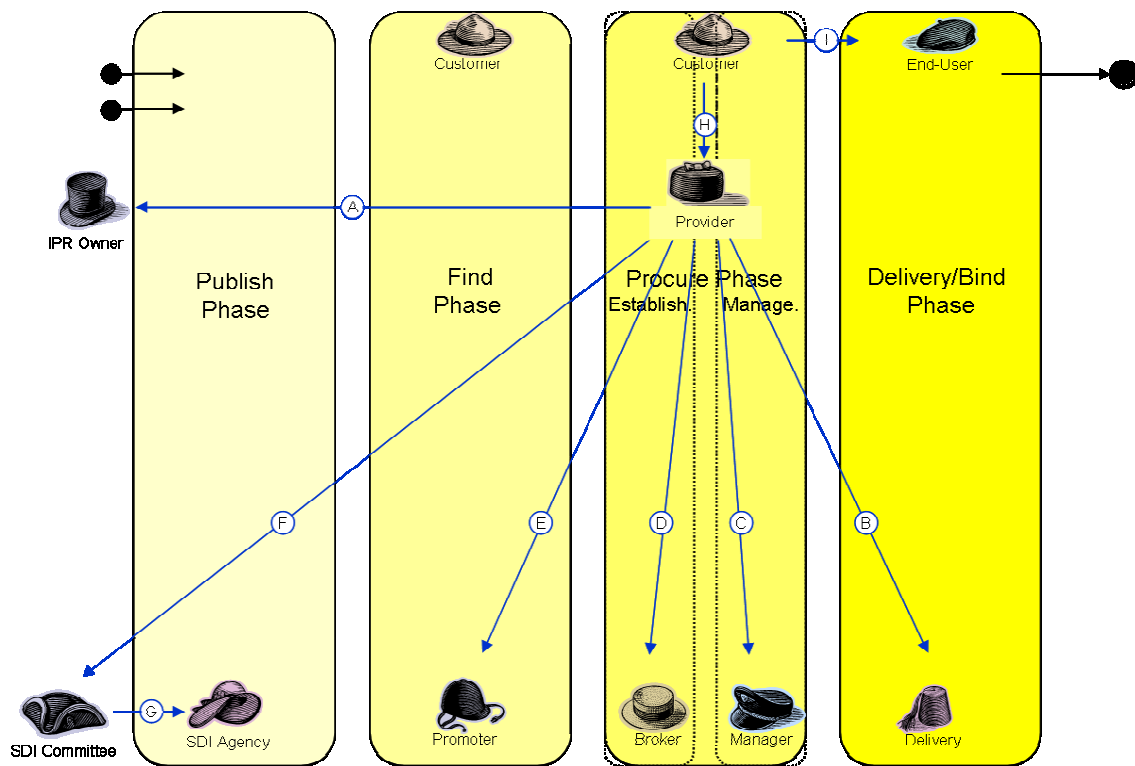


Figure 41—Legal relationships / internal agreements

This chapter uses some roles, which were introduced in 5, to describe the business process between them. Although the “publish-find-bind” process is well known due to the service-oriented architecture (SOA) concept, explicit contracting adds an additional phase “procure” into the process “publish-find-procure-bind”. The procurement phase is separated into two parts to reflect the potential organisational separation. **Error! Reference source not found.** shows the roles, their legal relationships and the fundamental business phases. If multiple roles are covered by a single legal institution, the legal relationship transforms into an internal agreement.

The roles “provider” and “customers” are key players. The provider establishes and maintains the provider site network. The provider receives the usage rights for a product with a contract from an IPR Owner (A). The relationship (B) covers the storage and operation of delivery services (e.g. WMS, WFS...). In the case of an explicit contracting for customers in the procurement phase, the relationship (C) and (D) need to be established. The relationship (E) covers the promotion of the product with product catalogues. The relationships (A...E) can be summarized as a “SDI provider business network”. If the product should be registered in a SDI node for further compatibility, the relationship to a SDI committee (F) is needed. The committee may delegate the relationship to an SDI agency (G).

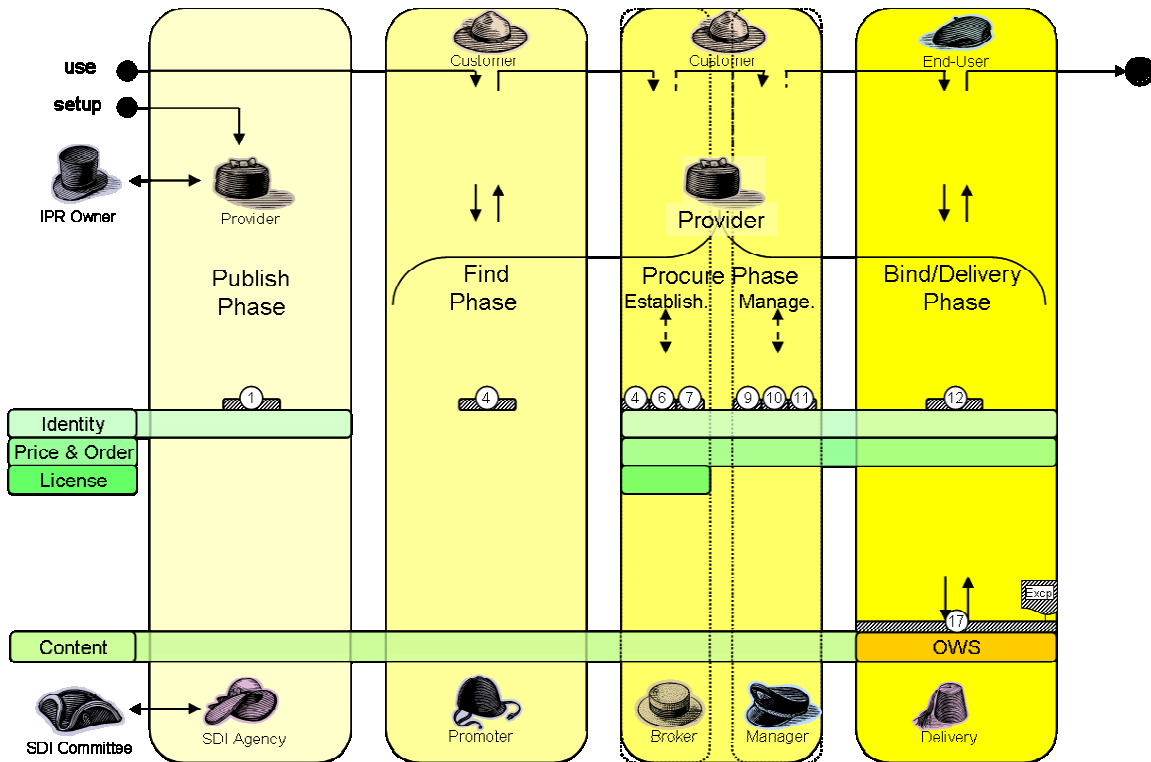


Figure 42– Example of sharing function use (Identity – Price & Order – License)

The customer may have a customer site network or even a value chain. Currently only the relationship “customer” as a legal institution and “end-user” as a natural person, e.g. employee, is modelled (I).

The business relationship between customer and provider site network are supported by the sharing functions “identity”, “price & order” and “license”. The degree of usage depends on the applied operation model. **Error! Reference source not found.** depicts an example of used sharing functions. The digital identity of a provider is required to publish a new product (1). A customer may not need to use any sharing functions to find a product (4). This is important to attract a wide audience. In this case an explicit procurement with pricing & ordering and licensing is needed to establish (4...7) a legal relationship. Because the product has usage-depending pricing & ordering, the management (9...11) and the delivery (12) needs to be supported with identity and price & order functions.

The following paragraphs below describe the processes in each business phase in detail.

12.3 Publish Phase

Although the publish phase is currently not directly addressed by interoperable services, the phase is getting more relevant, because GeoDRM enablement requires more Metadata descriptions for workflows and preconditions. INSPIRE introduced the term “upload services”. The depict example above shows the wide range of potential sharing functions usage. Therefore there is an extensive need to describe the potential variations to allow *full-informed* usage. That means that a user is able to receive all necessary preconditions and steps *prior* he decides to start the process. The opposite approach is the trial & error usage which fits for simple operation models without major legal risks (see OWS-3 GeoDRM: #05-111r2). The following descriptions were identified:

- OWS Capabilities, (B1)
- GeoDRM enhanced Gatekeeper Capabilities, (B2)
- Manager Capabilities, (C)
- Broker Capabilities, (D)
- Product Catalogue Metadata (ISO19115/39) , (E)
- Additional Registrations (e.g. ProductID, tbd), (F)

Some descriptions are already defined or proposed within OGC (B1, E) and OWS4. geoDRM (B2), others are expected (C), (D) and (F). See 8.2.3 for details. The following phases were conceptual studied and partly implemented in the OWS-4 GeoDRM initiative.

12.4 Find Phase

In the find phase a potential Customer interacts with a Promoter via a catalogue service. The goal is to find suitable products. The description of products can use metadata schemas and include legally not binding information.

The result of this phase is a product identifier for the desired product.

12.5 Procurement Phase: Establishment Process

The first part of the procurement phase is the establishment of a contract between a Customer and a Provider to cover the product usage or some sharing functions accounts. This interaction happens between the Customer and a Broker, who is acting on behalf of the Provider.

Multiple, often independent Brokers are used in parallel to achieve a better market penetration. Therefore this role is clearly separated from the Manager role although both belong to the procurement phase.

The establishment of sharing function accounts can be chained sequentially with the same operation set to allow different independent *Specialized Brokers* to establish a separated account for each sharing function. In case of a *General Broker* as a single institution, the establishment of all sharing aspects in parallel. Therefore the GeoDRM Engineering Viewpoint includes security and pricing & ordering aspects from a business process and distribution point of view.

The results of an establishment process are:

- *Establishment of a user identity account*
- *Establishment of a licensing account*
- *Establishment of a pricing & ordering account*
- *Establishment of data service usage contract*

After a new account was established legally, the broker creates a new account at the contracted manager's site.

12.6 Procurement Phase: Management Process

The role "Manager" was defined to maintain the Customer relationship on behalf of the Provider. In opposite to multiple Brokers, the number of Manager instances is often one. Also the roles "Provider" and "Manager" are covered often within a single institution to maintain a close relationship to Customers. A duplicated Manager may also be operated at the customer's site, but this idea needs more consideration.

A Manager maintains the sharing functions accounts. A Specialized Manager operates only a subset of sharing functions, e.g. "identity" only. A Generalized Manager operates all sharing functions accounts. The Manager has technical relationships to the Provider, to the Broker, to the Delivery and to the Customer. The access is controlled by the identity sharing function. Some products are contracted on usage bases. Examples are trial licenses, e.g. with 5 free trial requests. Other examples are pre-paid accounts and a cent price per request. Pending on the applied operation model the role "Delivery" might request a balance check at the Manager's site prior delivery. And onwards a count down request after the product was delivered successfully. The Customer may check and maintain this account balance any time, e.g. for a prepaid balance or changing a telephone number.

The result is an updated account or viewed account statement.

12.7 Delivery (Bind) Phase

The last phase in a classical business process is the delivery of the product. This phase is called “bind” in the SOA world. The delivery may start upon request by an end-user, or by asynchronous delivery (e.g. eMail or satellite broadcast). The role “Delivery” may check a balance at the Manager, if required by the product type.

The result is a delivered product (, an updated account) and a delivery receipt.

12.8 Chained Business Phases

Figure 43 shows the aligned business phases, with electronic client and service site components, interfaces and roles. It unifiers a classic business process “promotion, offer, agreement and delivery” and the SOA “publish-find-(procure)-bind”. The Provider starts the initialization of a new product with a “setup” process in the publish phase.

The Customer starts the usage with the find phase. Therefore he uses a catalogue client to search a catalogue service (4). The result is a product identifier. The customer interacts with a Broker Client together with a Broker Service (5, 6 and 7) to establish a new contract. If successful a new account is created at the manager (8). He may also check his accounts prior delivery (9, 10 and 11). An integrated variant of Broker, Manager and Gatekeeper Clients is often called GeoDRM Client.

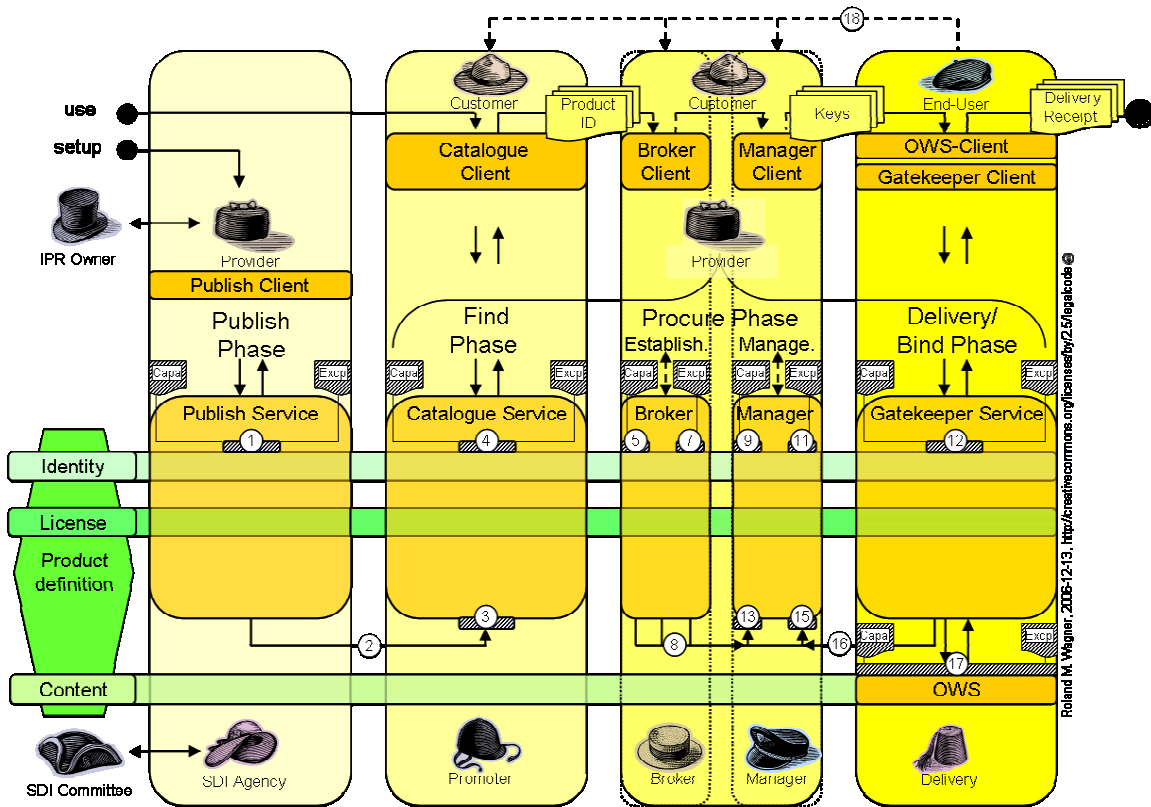


Figure 43–Business Phases and workflows

The End-user uses a specialized client (OWS-Client) to create a data service request (OWS), which will be tunneled via the Gatekeeper Client and the Gatekeeper service (12, 17), together with suitable sharing tokens (identity, pricing, licensing). These tokens allow a check at the manager’s site (16, 13, 14 and 15). The customer may update, upgrade, downgrade his subscribed products by re-entering the find or procure phase (18).

13 Future Work

Although this document contains the engineering viewpoint for a GeoDRM-enabled architecture that was proved by implementations and prototype demonstrators, the following items may be subject to further refinement and investigation:

- A more sophisticated description (Capabilities) of the distributed components (License Broker, License Manager, Authentication Service and GeoDRM Gatekeeper).
- A better understanding of what are “products” in terms of a GeoDRM enabled system and how those products are defined in a technical manner. Especially if a service offers multiple products composed of a composition of resources and actions.
- Potential issues that may arise if more than one encoding for licenses (and especially the included grants) are used.
- The “General Manager” and “General Broker” pattern is subject to be proved. For the time being it’s an idea.
- License Broker interface and functionalities needed to be elaborated. Especially its capabilities, further elaboration of license types, offers and contracting processes.
- Relation to price & order processing.

14 References

- [XACML] OASIS Open (2003): eXtensible Access Control Markup Language (XACML) Version 1.1
<http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf>
- [SAML] OASIS Open (2003): Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1
<http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf>
- [XMLSig] W3C (2000): XML-Signature
<http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/>
- [WS-S] OASIS Web Services Security, Version 1.1, February 2006, online at
<http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
- [OWS4Trust] OWS-4 IPR: Trusted Geo Services, OGC #06-107
- [GeoDRM RM] The GeoDRM Reference Model, February 2006, OGC #06-004r4
- [WS-P] Web Services Policy Framework (WS-Policy Version 1.2)
<http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf>
- [WS-PA] Web Services Policy Attachment (WS-PolicyAttachment Version 1.2)
<http://specs.xmlsoap.org/ws/2004/09/policy/ws-policyattachment.pdf>
- [WS-A] Web Services Addressing (WS-Addressing)
<http://www.w3.org/Submission/ws-addressing/>
- [WS-SP] Web Services Security Policy Language (WS-SecurityPolicy Version 1.1)
<http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>
- [WS-S] Web Services Security v1.0 (WS-Security 2004)
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
- [SAMLToken] Web Services Security
SAML Token Profile 1.0
<http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf>

[GEOXACML GeoXACML, a spatial extension to XACML, June 2005, OGC #05-036
]