deegree owsProxy v2.5

lat/lon GmbH
Aennchenstr. 19
53177 Bonn
Germany
Tel ++49 - 228 - 184 96-0
Fax ++49 - 228 - 184 96-29
info@lat-lon.de
www.lat-lon.de

Dept. of Geography
Bonn University
Meckenheimer Allee 166
53115 Bonn

Tel. ++49 228 732098

Change log

Datum	***Beschreibung***	***Author***
2006-10-23	Kleinere Layout-Anpassungen	Markus Müller
2006-11-02	Alle Beispiele mit prefixes versehen; WFSPolicy-Beispiel umd FeatureTypes ergänzt	Andreas Poth
2007-01-15	Harmonisierung mit Standard-deegree Dokumentationsstruktur	Markus Müller
2007-03-01	Präszisierung zur Benutzung eines WAS	Markus Müller
2007-03-12	Präzisierung der Condition-Parameter; Ergänzung der WFS-Policy um transaktionale Parameterdefinitionen	Andreas Poth
2007-03-16	Redaktionelle Anpassungen	Markus Müller
2008-01-31	Ergänzung neuer Konfigurationsmöglichkeiten	Andreas Poth
2008-02-04	Rechtschreibekorrekturen; Umstellung des Dokumentes auf deutsche Sprache	Markus Lupp
2008-05-19	Korrektur von Kap. 3.4	Andreas Poth
2009-10-08	Bug fix	Andreas Poth

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

deegree ist ein Java Framework, das die Bausteine für Geodateninfrastrukturen (GDI) zur Verfügung stellt. Seine ganze Architektur wurde auf Basis von Standards des Open Geospatial Consortium (OGC) und des ISO Technical Committee 211 – Geographic information / Geoinformatics (ISO/TC 211) entwickelt. . deegree umfasst außer OGC Web Services auch Clients. deegree ist Freie Software, wird durch die GNU Lesser General Public Licens (GNU LGPL) geschützt und steht über http://www.deegree.org zur Verfügung.

deegree2 ist das neue Release von deegree, das eine Anzahl von Funktionalitäten zur Verfügung stellt, zu denen deegree1 nicht in der Lage war. Diese Dokumentation beschreibt die Installation und Konfiguration von deegree owsProxy, einem Teil von deegree iGeoSecurity.

Die Spezifikationen des Open Geospatial Consortium sind die Grundlage von Interoperabilität in Geodateninfrastrukturen. Als Schnittstellenspezifikationen treffen sie keine Festlegungen über Nutzer, Berechtigungen und Sicherheitsmechanismen. Ein Web Map Service oder Catalogue Service, dessen Adresse bekannt ist, kann von jedermann genutzt werden. Insbesondere seit eine große Menge von OWS-Clients für jedermann zur Verfügung stehen, ist der Zugriff über eine offene Schnittstelle kein Expertenwissen mehr. In der Regel steht der gesamte Datenumfang eines OWS-Servers zur Verfügung. Die Auskunft über verfügbare Datenbestände ist laut Spezifikation eine wesentliche Voraussetzung für Interoperabilität und somit unbedingt gewollt.

Ein Stellvertreter (Proxy) vor dem eigentlichen OGC Web Service (OWS) verhält sich wie der entsprechende OWS selbst, bietet aber zusätzliche Filtermöglichkeiten für Ein- und Ausgabe an, beispielsweise:

- Es werden mehrere Sichten auf einen OWS eingerichtet. Unterschiedliche Nutzer haben jeweils nur Zugriff auf eine Teilmenge dieser Sichten (WMS1 für Planungsdaten, WMS2 für Topographie, WMS3 für Gewässerdaten etc.)
- In einem owsProxy können verschiedene Nutzer nur auf bestimmte Layer, FeatureTypes etc. zugreifen
- Der Zugriff kann darüber hinaus räumlich und thematisch eingeschränkt sein, also z.B. nur für ein bestimmtes Stadtviertel, ein bestimmtes Ausgabeformat, eine maximale Auflösung etc..
- Diese Einschränkungen können für die verschiedenen Operationen eines OWS (z.B. `GetCapabilities`, `GetMap`, `GetFeature`, `GetRecords`, `Transaction` etc.) einzeln gesteuert werden.
- Zugriffsrechte können sowohl global als auch nutzerspezifisch definiert werden.

Der in deegree realisierte **owsProxy** ermöglicht die Definition der genannten Filtermöglichkeiten. Zur Zeit können OGC WMS, WFS und CS-W durch den deegree **owsProxy** gesichert werden.

Außer owsProxy umfasst deegree eine Anzahl anderer Dienst und Cliens. Eine komplette Liste der deegree Komponenten ist einsehbar unter:

`http://www.lat-lon.de` → Produkte

Installationspakete bestimmer Komponenten können unter der folgenden URL gefunden werden:

`http://www.deegree.org` → Download

Die Web Services von deegree sind als Javamodule implementiert, die von einen zentralen Servlet (dem “dispatcher”) gesteuert werden. Dieses Servlet muss in einem entsprechenden Web Server / Servlet Engine veröffentlicht werden. Die meisten der verbreiteten Web Server unterstützen Servlettechnologie, womit deegree universell einsetzbar ist. Der Einsatz der Apache Tomat 5.5 Servlet Engine wird empfohlen aufgrund seiner großen Verbreitung und seines Status als Open Source-Produkt.

2 Download / Installation

2.1 Voraussetzungen

Zur Installation von deegree owsProxy werden benötigt:

- Java (JRE or JSDK) version 1.5.x
- Tomcat 5.5.x

Zur Installation dieser Komponenten sollte die entsprechende Dokumentation auf java.sun.com und tomcat.apache.org herangezogen werden.

2.2 deegree owsProxy release

Bislang steht kein Installationspaket für deegree owsProxy zur Verfügung. Aus diesem Grund müssen die entsprechenden Dateien aus dem deegree CVS bezogen werden.

Um einen owsProxy zu installieren werden die folgenden Komponenten benötigt:

- eine Konfiguration für den owsProxy a configuration for the WAS and/or the WSS (the WAS is optional)
- ein deegree2.jar (entweder vorkompiliiert oder aus den Klassen des CVS erzeugt)
- optional: eine U3R-Datenbank
- ein entsprechender OGC Web Services

2.3 Installation

Der deegree **owsProxy** ist als Java Servlet-Filter konzipiert und nicht auf den Einsatz in Kombination mit deegree OWS beschränkt. Sollte sich der zu schützende Dienst auf einem physikalisch oder logisch anderen Server befinden, kann der deegree **owsProxy** mit einem einfachen Proxy-Servlet kombiniert werden, das die eingehenden validen Anfragen an den entsprechenden Dienst weiterleitet bzw. dessen Antworten entgegen nimmt. Dies ermöglicht es u.a., dass neben Diensten, die ebenfalls als Java Servlets realisiert sind, auch in anderen Sprachen entwickelte OWS wie der UMN MapServer mittels des deegree OWSProxies geschützt werden können.

Zur Installation muss der **owsProxy** zunächst als ServletFilter in einen Web Kontext eingebunden werden. Dies geschieht über entsprechende Einträge im Deployment Descriptor des jeweiligen Kontexts. Damit die Servlet Engine benötigte Java-Klasse findet, muss das Archiv deegree2.jar im Klassenpfad der Servlet-Engine registriert werden.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
   <filter>
      <filter-name>OWSProxy</filter-name>
      <filter-class>
         org.deegree.security.owsproxy.ConfigurableOWSProxyServletFilter
      </filter-class>
      <init-param>
         <param-name>WMS:POLICY</param-name>
         <param-value>./resources/wmspolicy.xml</param-value>
      </init-param>
      <init-param>
         <param-name>CSW:POLICY</param-name>
         <param-value>./resources/cswpolicy.xml</param-value>
      </init-param>
      <init-param>
      <init-param>
         <param-name>AuthenticationSettings</param-name>
         <param-value>/WEB-INF/conf/security/authentication.xml</param-value>
      </init-param>
      <init-param>
         <param-name>PROXYURL</param-name>
         <param-value>http://myHost:8080/owsproxy/proxy</param-value>
      </init-param>
      <init-param>
         <param-name>ALTREQUESTPAGE</param-name>
         <param-value>/altrequestpage.jsp</param-value>
      </init-param>
      <init-param>
         <param-name>ALTRESPONSEPAGE</param-name>
         <param-value>/altresponsepage.jsp</param-value>
      </init-param>
   </filter>
   <filter-mapping>
      <filter-name>OWSProxy</filter-name>
      <url-pattern>/*</url-pattern>
   </filter-mapping>
   <servlet>
      <servlet-name>SimpleProxyServlet</servlet-name>
      <servlet-class>
         org.deegree.enterprise.servlet.SimpleProxyServlet
      </servlet-class>
      <init-param>
         <param-name>WMS:HOST</param-name>
         <param-value>http://localhost:8081/deegree/services</param-value>
      </init-param>
      <init-param>
         <param-name>CSW:HOST</param-name>
         <param-value>http://localhost:8081/deegree/services</param-value>
      </init-param>
   </servlet>
   <servlet-mapping>
      <servlet-name>SimpleProxyServlet</servlet-name>
      <url-pattern>/proxy</url-pattern>
   </servlet-mapping>
</web-app>
```

Im angegebenen Beispiel wird der **owsProxy *nicht*** im selben Context wie die zu schützenden deegree WMS und CSW installiert. Somit wird die Einrichtung des Proxy-Servlets erforderlich. Damit wird jede Anfrage, die den deegree WMS bzw. CSW adressiert zunächst von owsProxy entgegen genommen und auf ihre Gültigkeit überprüft. Erst wenn der Filter die Anfrage als gültig identifiziert ist, wird sie an Proxy-Servlet weitergeleitet, das seinerseits die Anfrage an den über die Parameter WMS:HOST bzw. CSW:HOST OWS weitergibt. Umgekehrt, geht die Antwort des OWS ebenfalls zunächst an den ServletFilter und wird dort validiert und ggf. modifiziert.

Sowohl dem eigentlichen owsProxy als auch dem ProxyServlet müssen bei der Registrierung im Deployment Descriptor Initialisierungsparameter übergeben werden. Um dem owsProxy bekannt zu machen, für welche OWS er als Filter agieren soll und ihm gleichzeitig mitzuteilen, wo die entsprechenden Filterregeln abgelegt sind, wird für jeden zu sichernden Dienst ein Parameter beginnend mit dem OWS Typ gefolgt von ':Policy' definiert. Der Wert dieser Parameter verweist auf eine so genannte Policy-Datei, die die anzuwendenden Regeln enthält. Eine detaillierte Beschreibung der Policies findet sich in Kapitel 3. Über den Initialisierungsparameter 'AuthenticationSettings' wird auf eine XML-Datei verwiesen, in der festgelegt wird, welche Verfahren in welcher Reihenfolge zur Authentifizierung von Nutzern verwendet werden sollen. Auch dies wird in Kapitel 3 näher beschrieben.

Wird das deegree Proxy Servlet verwendet, muss diesem bekannt gemacht werden, für welche Dienste es als Proxy agieren soll. Eine Instanz eines Proxy Servlets kann gleichzeitig als Stellvertreter für mehrere unterschiedliche OWS fungieren, aber immer nur einen Dienst eines Typs kapseln. Über die Init-Parameter wird definiert, an welche Dienste Anfragen weitergeleitet werden sollen. Die entsprechenden Parameternamen beginnen mit dem Servicetyp der gefolgt wird von ':HOST'. Der Wert der Parameter ist jeweils die Basis-URL des jeweiligen Dienstes.

3 Basiskonfiguration

Der deegree OWS realisiert ein pessimistisches Rechtekonzept. D.h. zunächst sind alle Zugriffe (Wertebereich der bekannten Parameter) blockiert. Ein Administrator muss daher jeden Zugriff und jeden erlaubten Wertebereich über so genannte Policy-Dateien explizit frei geben. Damit dies effizient möglich ist, können u.a. Wertebereiche für einzelne Parameter oder Parametergruppen als Ganzes freigegeben werden.

Ferner ist es möglich, den Zugriff auf bestimmte Methoden oder Wertebereiche einzelner Parameter an individuelle Nutzerrechte zu koppeln. Zu diesem Zweck muss der deegree owsProxy mit einer Benutzer- und Rechteverwaltung (deegree U3R) verbunden werden.

Damit der owsProxy Anfragen ihm bekannten Nutzern zuordnen kann, verfügt er über mehrere Authentifizierungsmechnismen. Welche dieser Mechnismen wie genutzt werden, wird, wie bereits beschrieben, in einer gesonderten XML-Datei festgelegt, die dem owsProxy mittels des Initialisierungsparameters 'AuthenticationSettings' übergeben wird.

3.1 Festlegung der Nutzerauthentifizierung

Der owsProxy kennt vier vordefinierte Methoden der Nutzerauthentifizierung, die letztendlich alle auf die deegree Rechteverwaltung zugreifen (siehe U3R Dokumentation):

1. Nutzername/Passwort: In der Anfrage müssen der Name und das zugehörige Passwort des Nutzer über die dafür vorgesehenen Parameter angegeben werden.
2. Web Authentication Service (WAS): Hierbei handelt es sich um einen in der GDI-NRW spezifizierten Dienst, der eine zeitlich begrenzt gültige SessionID vergibt, bzw. eine vorhandene SessionID auf ihre Gültigkeit überprüft. (siehe deegree WASS Dokumentation).
3. UserPrincipal: Hierbei wird der Nutzername aus dem HTTP-Header der eingehenden Anfrage extrahiert. I.d.R. wurde in diesem Fall bereits vom verwendeten Server ein Nutzer grundsätzlich authentifiziert, daher steht dem owsProxy lediglich der Nutzername zur Verfügung, d.h. in der deegree Rechteverwaltung dürfen den Nutzern keine Passworte zugeordnet sein (siehe U3R Dokumentation).

4. IP-Adresse: In diesem Fall wird die IP-Adresse des anfragenden Nutzer mit der deegree Rechteverwaltung abgeglichen. Dabei kann es sich sowohl um exakte Adressen (z.B. 127.0.0.1) also auch um Adressmuster (z.B. 19.1.10.*) handeln. Formal entspricht eine solche Adresse bzw. ein solches Muster einem Nutzernamen in der deegree Rechteverwaltung und muss dort auch in dieser Form (ohne Passwort) definiert sein.

Die vom owsProxy zu verwendenden Authentifizierungsmethoden werden, wie bereits dargestellt, in einer eigenen Datei definiert. Diese enthält eingebettet in das Wurzelelement die einem owsProxy bekannten Authentifizierungsmethoden.

```
<?xml version="1.0" encoding="UTF-8"?>
<Authentications>
   <Method name="user:password">
      <class>org.deegree.security.UserPasswordAuthentication</class>
   </Method>
   <Method name="WAS">
      <class>org.deegree.security.WASAuthentication</class>
      <init-param>
         <name>WAS</name>
         <value>
<![CDATA[http://localhost:8081/was/was?
REQUEST=DescribeUser&Service=WAS&version=1.0.0&SESSIONID=[SESSIONID]]]>
         </value>
      </init-param>
   </Method>
   <Method name="UserPrincipal">
      <class>org.deegree.security.UserPrincipalAuthentication</class>
   </Method>
   <Method name="IP-Address">
      <class>org.deegree.security.IPAddressAuthentication</class>
      <init-param>
         <name>pattern</name>
         <value>127.0.0.*,localhost,19.1.10.21</value>
      </init-param>
   </Method>
</Authentications>
```

Wie aus dem oben stehenden Beispiel zu ersehen ist, kann eine solche Datei mehrere Authentifizierungsmethoden definieren. In diesem Fall versucht ein owsProxy zunächst, ob ein Nutzer mit der ersten definierten Methode authentifiziert werden kann. Gelingt dies nicht, versucht er es mit der zweiten. Dies wird solange fortgesetzt, bis die Authentifizierung entweder erfolgreich war oder bis alle definierten Methoden angewendet wurden. Im letzteren Fall, d.h. wenn keine Authentifizierungsmethode erfolgreich war, wird die Anfrage eines Nutzers abgelehnt.

Für jede Authentifizierungsmethode werden der Name der verantwortlichen Java-Klasse sowie ggf. benötigte Initialisierungs-Parameter angegeben. Die verwendeten Java-Klassen müssen die abstrakte Klasse org.deegree.security.AbstractAuthentication implementieren. Dadurch ist es möglich, weitere ggf. projektspezifische Authentifizierungsmethoden zu deklarieren, ohne in den vorhandenen deegree-Code eingreifen zu müssen.

3.1.1 Nutzername/Passwort

Dies ist die einfachste der in deegree verfügbaren Authentifizierungsmethoden. Hierfür werden aus einer eingehenden Anfrage der Benutzername und das zugehörige Passwort ausgelesen und gegen die deegree Rechteverwaltung abgeglichen. Diese Authentifizierungsmethode wird deklariert durch das Setzen des Attributs Method/@name='user:password' (siehe Beispiel oben).

Im Fall von KVP kodierten Anfragen (i.d.R. HTTP-Get) werden Benutzername und Passwort über die Parameter 'USER' und 'PASSWORD' ausgelesen. Bei XML-kodierten Anfragen (i.d.R. HTTP-Post) wird erwartet, dass das Wurzelelement der Anfrage die Attribute 'user' und 'password' aufweist.

3.1.2 Web Authentication Service

Der Web Authentication Service (WAS) wurde im Rahmen der GDI-NRW spezifiziert. Die mittels Method/@name='WAS' deklarierte Authentifizierungsmethode nutzt diesen, um die in einer Anfrage enthaltene SessionID auf ihre Gültigkeit zu prüfen und den zugehörigen Nutzer zu identifizieren. Hierzu benötigt ein owsProxy ein Anfragepattern, in dem er den Platzhalter [SESSIONID] gegen die in einer Anfrage enthaltene SessionID austauscht. Dieses Pattern wird der Authentifizierungsmethode über den Initialisierungsparameter 'WAS' übergeben (Details zum WAS finden sich in der entsprechenden GDI-NRW Spezifikation sowie in der deegree WASS-Dokumentation).

Im Fall von KVP kodierten Anfragen (i.d.R. HTTP-Get) wird die SessionID über den Parameter 'SESSIONID' ausgelesen. Bei XML-kodierten Anfragen (i.d.R. HTTP-Post) wird erwartet, dass das Wurzelelement der Anfrage das Attribut 'sessionID' aufweist.

3.1.3 UserPrincipal

Vor allem innerhalb von Netzwerken, bei denen sich ein Nutzer zentral anmeldet, wird die Nutzerauthentifizierung bei Anfragen gegen einen Server innerhalb des HTTP-Header mitgesendet. Eine Anfrage bzw. die im Header enthaltene Kennung wird in diesem Fall zunächst vom Server geprüft, bevor die eigentliche Anfrage an den owsProxy weitergereicht wird. Daher steht in diesem Fall im owsProxy lediglich der Name des Nutzers zur Verfügung; da der Nutzer bereits gegen das Netzwerk bzw. den Server authentifiziert wurde, muss keine weitere Prüfung gegen die deegree Rechteverwaltung erfolgen. Dieser muss lediglich der Nutzer bekannt sein. Soll UserPrincipal als Authentifizierungsmethode verwendet werden, müssen die entsprechenden Nutzer ohne Passwort in der deegree Rechteverwaltung registriert sein!

Enthält eine eingehende Anfrage keine Nutzerkennung im Header, setzt deegree als default Wert den in der Datei org.deegree.enterprise.servlet.ServletRequestWrapper.properties definierten Nutzernamen. Dieser agiert in diesem Fall als anonymer Nutzer. ***Daraus folgt, dass die Authentifizierungsmethode 'UserPrincipal' immer dann deklariert werden muss, wenn anonymen Nutzern über die deegree Rechteverwaltung Zugriffsrechte eingeräumt werden sollen!*** (Werden Zugriffsrechte in den weiter unten beschriebenen policy-Dateien nicht nutzerspezifisch deklariert, kann die Vereinbarung der Authentifizierungsmethode 'WAS' selbstverständlich entfallen.

3.1.4 IP-Address

Die letzte der in deegree vordefinierten Authentifizierungsmethoden nutzt die IP-Adresse eines anfragenden Nutzers, um diesen gegen die deegree Rechteverwaltung zu authenfizieren. Hierzu können IP-Adressen und Adressmuster von der deegree Rechteverwaltung wie normale Nutzer (ohne Passwort) verwaltet werden.

Der owsProxy prüft im Fall der über Method/@name='IP-Adress' deklarierten Methode zunächst, ob die Adresse eines anfragenden Nutzer mit einer der über Komma getrennten Adressen/Muster übereinstimmt, die mittels des Initialisierungsparameters 'pattern' an die Authentifizierungsmethoden übergeben wurden. Ist dies der Fall, wird die Adresse bzw. das Muster, das zur anfragenden Adresse passt, im Folgenden als Nutzername verwendet. Das heisst, die Überprüfung, ob die vorhandenen Rechte zum Ausführen, der eingegangenen Anfrage ausreichend sind, erfolgt über einen Nutzer dessen Name dem des zutreffenden IP-Musters bzw. der zutreffenden IP-Adresse entspricht.

Damit ist es z.B. möglich, alle Mitarbeiter eine Behörde, einer Organisation oder einer Firma über ihre IP-Adresse zu authentifizieren und Rechte innerhalb der deegree Rechteverwaltung nur einmal für die gesamte Behörde/Firma/Organisation zu vergeben und nicht für jeden Nutzer einzeln.

3.2 Policy Dateien

Eine Policy-Datei enthält Informationen darüber, welche Zugriffe auf einen Dienst erlaubt sind, d.h. sie enthält Freigaben und keine Verbote! Ferner kann in ihr festgelegt werden, ob eine Kopplung an eine Benutzer- und Rechtedatenverwaltung erfolgen soll und wenn ja, welche Parameter davon betroffen sind. Aufgrund der unterschiedlichen Funktionen und ihrer Parameter unterscheiden sich die Policy-Dateien für die verschiedenen abzusichernden Dienste. Lediglich einige Elemente werden von allen Diensten in gleicher Form genutzt. Das folgende Beispiel zeigt den Ausschnitt einer Policy-Datei, wie er in seiner Struktur allen Diensten gemein ist:

```
<?xml version="1.0" encoding="UTF-8"?>
<OWSPolicy xmlns="http://www.deegree.org/security" service="CSW">
   <Security>
      <RegistryClass>org.deegree.security.drm.SQLRegistry</RegistryClass>
      <ReadWriteTimeout>300</ReadWriteTimeout>
      <RegistryConfig>
         <jdbc:JDBCConnection xmlns:jdbc="http://www.deegree.org/jdbc">
            <jdbc:Driver>String</jdbc:Driver>
            <jdbc:Url>String</jdbc:Url>
            <jdbc:User>String</jdbc:User>
            <jdbc:Password>String</jdbc:Password>
            <jdbc:SecurityConstraints/>
            <jdbc:Encoding>ISO-8859-1</jdbc:Encoding>
         </jdbc:JDBCConnection>
      </RegistryConfig>
      <authentificationSettings>
         <authentificationService>
            <OnlineResource
               xlink:type="simple" xlink:href="http://localhost:8081/was/was"/>
         </authentificationService>
      </authentificationSettings>
   </Security>
   <GeneralConditions>
      <Conditions>
         <Parameter name="getContentLength" userCoupled="false">
            <Value>1000</Value>
         </Parameter>
         <Parameter name="postContentLength" userCoupled="false">
            <Value>1000000</Value>
         </Parameter>
         <Parameter name="header" userCoupled="false">
            <Any/>
         </Parameter>
         <Parameter name="requestMethod" userCoupled="false">
            <Value>GET,POST</Value>
         </Parameter>
      </Conditions>
   </GeneralConditions>
   <Requests>
      [...]
   </Requests>
```

```
</OWSPolicy>
```

Innerhalb des Root-Elements im <Security>-Element erfolgt zunächst die optionale Definition des Zugangs zur Benutzer- und Rechteverwaltung und (ebenfalls optional) zu einem Authentifizierungsdienst. Der gesamte Block wird nur dann benötigt, wenn einzelne oder alle Anfrageparameter an individuelle Benutzerrechte gekoppelt werden sollen. Im Anschluss an den security-Block erfolgt die Definition allgemeiner, d.h. nicht service-spezifischer Zugriffsrechte. Diese sind im Element <GeneralConditions> abgelegt und beziehen sich auf ausgewählte HTTP-Parameter.

3.2.1 Security

Wird eine Kopplung an individuelle Benutzerrechte definiert, muss dem owsProxy zunächst mitgeteilt werden, welche Art von Registry (zur Zeit ist nur org.deegree.security.drm.SQLRegistry verfügbar) verwendet werden soll und wie der Zugang zu ihr aussieht. Eine SQLRegistry ist in Form eines Datenbankschemas realisiert, daher müssen die benötigten Zugriffsparameter auf die jeweilige Datenbank angegeben werden.

- Driver: Java Treiberklasse für den JDBC-Zugang (abhängig von der verwendeten Datenbank; im Beispiel ist eine ODBC-Datenbank konfiguriert)
- Url: Adresse und Name unter der die entsprechende Datenbank über den JDBC-Treiber verfügbar ist
- User: Benutzername zum Zugriff auf die Datenbank (optional)
- Password: (optional)

Das Element <ReadWriteTimeout> definiert, nach wieviel Millisekunden die Rechteverwaltung den Versuch eines Zugriffs auf die Registry abbricht, falls diese nicht antwortet.

Neben dem Zugriff auf die Benutzer- und Rechtedatenbank kann innerhalb des security Elements die Adresse eines Authentifizierungsdienstes definiert werden. Diese Angabe ist optional. Fehlt sie, muss ein Benutzer an jede Anfrage, die er schickt, seinen Name (USER=XYZ) und sein Passwort (PASSWORD=ABC) anfügen. Ist dagegen ein Authentifizierungsdienst angegeben, kann anstelle der sehr unsicheren Verwendung des Benutzernamens und Passworts eine SessionID (z.B. SESSIONID=2e3r45t) verwendet werden, die sich der Benutzer zuvor mittels der Anfrage GetSession beim Authentifizierungsdienst abgeholt hat. In diesem Fall überprüft der owsProxy zunächst die Gültigkeit der mitgeschickten SessionID und ermittelt anschließend die zugehörigen Benutzerdaten, die eine Validierung der Anfrage gegen die Rechtedatenbank ermöglichen. Wird weder eine SessionID noch Benutzername und Passwort an eine Anfrage angehängt, geht der owsProxy von einem anonymen Benutzer aus. Die Kennung des anonymen Nutzers kann über die Datei org.deegree.enterprise.servlet.ServletRequestWrapper.properties konfiguriert werden. Standardmässig ist die Kennung des anonymen Nutzers auf 'default' gesetzt.

Bsp.:

```
http://myhost:8080/deegree/proxy?
request=GetCapabilities&version=1.1.1&service=WMS&&...&user=latlon&p
assword=latlonspassword
```

```
http://myhost:8080/deegree/proxy?
request=GetMap&version=1.1.1&bbox=1,1,10,10&width=400&height=500  ...
&sessionID=3tr364g2
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<GetFeature outputFormat="GML2" xmlns="http://www.opengis.net/wfs"
 xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 sessionID="5z4g447">
   <Query typeName="Europe">
      <ogc:Filter>
         <ogc:BBOX>
            <ogc:PropertyName>GEOM</ogc:PropertyName>
            <gml:Box>
               <gml:coordinates>1,40 12,56</gml:coordinates>
            </gml:Box>
         </ogc:BBOX>
      </ogc:Filter>
   </Query>
</GetFeature>
```

Mit dem Web Authentification Service (WAS) definiert die GDI-NRW einen Dienst, der in der Lage ist, die Anfrage GetSession zu bearbeiten und als Antwort eine SessionID liefert, die für einen definierten Zeitraum gültig ist. Deegree erweitert diesen Dienst um die Anfrage DescribeUser, um den Besitzer einer SessionID zu ermitteln. Eine genaue Beschreibung des WAS erfolgt in einer eigenen Dokumentation.

3.2.2 GeneralConditions

In Sektion <GeneralConditions> werden HTTP-bezogene Regeln definiert, die verhindern sollen, dass Anfragen, die bereits aufgrund ihrer 'äusseren' Form als sinnlos oder bedenklich eingestuft werden können, bis zum zu sichernden Dienst weitergeleitet werden. Die Freigabe bestimmter Wertebereiche für einzelne HTTP-bezogener Parameter entspricht in ihrer Form der, wie sie zur Freigabe von dienstespezifischen Parametern verwendet wird. Daher steht die folgende Beschreibung stellvertretend für alle weiter unten beschriebenen dienstespezifischen Parameterdefinition/-freigaben.

Innerhalb des Elements <GeneralConditions> folgt das Element <Conditions>, das eine Liste von vier Parameterdefinitionen enthält, deren gültiger Wertebereich definiert werden muss. Jeder Parameter verfügt über die beiden Attribute 'name' und 'userCoupled'. Das erste Attribut gibt den Namen des Parameters an, dessen gültiger Wertebereich festgelegt werden soll. Über das Attribute 'userCoupled' kann definiert werden, ob die Überprüfung des gültigen Wertebereichs benutzerspezifisch erfolgen soll oder nicht. Wird userCoupled=“false“ gesetzt, gilt für alle Nutzer der selbe über die eingebetteten Elemente <Value> definierte Wertebereich. Ist userCoupled=“true“, wird anhand der Rechtedatenbank wird ermittelt, ob die Parameter ihrer Anfrage gültig sind.

Eingebettet in <Parameter> ist alternativ eine Liste von <Value> Elementen oder das leere Element <Any/>. Über <Value> können gültige Werte für einen Parameter festgelegt werden. Wird statt dessen <Any/> verwendet, wird jeder Wert einer Anfrage für diesen Parameter akzeptiert. Wird weder ein oder mehrere <Value> Elemente noch <Any/> definiert, wird vom owsProxy kein Wert für diesen Parameter akzeptiert. Dies kann z.B. sinnvoll sein, um anonymen Nutzer die Verwendung eines Parameters bzw. einer Funktion zu untersagen, bekannten Nutzern mit entsprechenden Rechten aber zu gestatten.

Zur Zeit können vier Parameter innerhalb von GeneralConditions definiert werden:

1. getContentLength: maximal zulässige Anzahl der Zeichen einer HTTP-Get Anfrage
2. postContentLength: maximal zulässige Anzahl der Zeichen einer HTTP-Post Anfrage

3. header: zulässige HTTP-Header Parameter (zur Zeit nicht implementiert)
4. requestMethod: kommaseparierte Liste der zulässigen HTTP-Methoden (OWS unterstützen zur Zeit standardmäßig nur GET und POST)

3.3 WMS Policies

Eine Policy für einen WMS-Filter ergänzt die im vorherigen Kapitel erläuterten allgemeinen Bedingungen/Regeln um solche, die spezifisch für einen WMS sind. Dazu gehört vor allen die Definition von anfragespezifischen Bedingungen und von Regeln zur Modifikation der Antworten eines WMS. Das folgende Beispiel zeigt eine Policy zur Absicherung eines WMS.

```
<?xml version="1.0" encoding="UTF-8"?>
<dgsec:OWSPolicy service="WMS" xmlns:dgsec="http://www.deegree.org/security"
   xmlns:xlink="http://www.w3.org/1999/xlink">
   <dgsec:Security>
<dgsec:RegistryClass>org.deegree.security.drm.SQLRegistry</dgsec:RegistryClass>
      <dgsec:ReadWriteTimeout>300</dgsec:ReadWriteTimeout>
      <dgsec:RegistryConfig>
         <dgjdbc:JDBCConnection xmlns:dgjdbc="http://www.deegree.org/jdbc">
            <dgjdbc:Driver>oracle.jdbc.OracleDriver</dgjdbc:Driver>
            <dgjdbc:Url>dgjdbc:oracle:thin:@localhost:1521:latlon</dgjdbc:Url>
            <dgjdbc:User>system</dgjdbc:User>
            <dgjdbc:Password>latlondba01</dgjdbc:Password>
            <dgjdbc:SecurityConstraints/>
            <dgjdbc:Encoding>iso-8859-1</dgjdbc:Encoding>
         </dgjdbc:JDBCConnection>
      </dgsec:RegistryConfig>
   </dgsec:Security>
   <dgsec:GeneralConditions>
      <dgsec:Conditions>
         <dgsec:Parameter name="getContentLength" userCoupled="false">
            <dgsec:Value>1000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="postContentLength" userCoupled="false">
            <dgsec:Value>10000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="header" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="requestMethod" userCoupled="false">
            <dgsec:Value>GET,POST</dgsec:Value>
         </dgsec:Parameter>
      </dgsec:Conditions>
   </dgsec:GeneralConditions>
   <dgsec:Requests>
      <dgsec:GetCapabilities>
         <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
               <dgsec:Value>GetCapabilities</dgsec:Value>
               <dgsec:Value>capabilities</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>1.1.0</dgsec:Value>
               <dgsec:Value>1.1.1</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="updatesequence" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Parameter name="layers" userCoupled="false">
               <dgsec:Value>stadtbezirke</dgsec:Value>
```

```
            <dgsec:Value>gebaeude</dgsec:Value>
            <dgsec:Value>quartiere</dgsec:Value>
         </dgsec:Parameter>
      </dgsec:PostConditions>
   </dgsec:GetCapabilities>
   <dgsec:GetMap>
      <dgsec:PreConditions>
         <dgsec:Parameter name="request" userCoupled="false">
            <dgsec:Value>GetMap</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="layers" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="version" userCoupled="false">
            <dgsec:Value>1.1.0</dgsec:Value>
            <dgsec:Value>1.1.1</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="bgcolor" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="transparency" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="exception" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="bbox" userCoupled="false">
            <dgsec:Value>3411000,5829000,3690000,6028000,EPSG:31467</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="format" userCoupled="false">
            <dgsec:Value>image/jpeg</dgsec:Value>
            <dgsec:Value>image/png</dgsec:Value>
            <dgsec:Value>image/gif</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="maxWidth" userCoupled="false">
            <dgsec:Value>1600</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="maxHeight" userCoupled="false">
            <dgsec:Value>1600</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="resolution" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
      </dgsec:PreConditions>
      <dgsec:PostConditions>
         <dgsec:Any/>
      </dgsec:PostConditions>
   </dgsec:GetMap>
   <dgsec:GetFeatureInfo>
      <dgsec:PreConditions>
         <dgsec:Parameter name="request" userCoupled="false">
            <dgsec:Value>GetFeatureInfo</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="version" userCoupled="false">
            <dgsec:Value>1.1.0</dgsec:Value>
            <dgsec:Value>1.1.1</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="featureInfoLayers" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="infoFormat" userCoupled="false">
            <dgsec:Value>application/vnd.ogc.gml</dgsec:Value>
            <dgsec:Value>text/html</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="maxFeatureCount" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="exception" userCoupled="false">
            <!-- any exception format defined here must also be valid
```

```
                    for GetMap -->
                <dgsec:Any/>
            </dgsec:Parameter>
        </dgsec:PreConditions>
        <dgsec:PostConditions>
            <dgsec:Any/>
        </dgsec:PostConditions>
    </dgsec:GetFeatureInfo>
    <dgsec:GetLegendGraphic>
        <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
                <dgsec:Value>GetLegendGraphic</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
                <dgsec:Value>1.1.1</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="layers" userCoupled="false">
                <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter name="maxWidth" userCoupled="false">
                <dgsec:Value>50</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="maxHeight" userCoupled="false">
                <dgsec:Value>50</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="format" userCoupled="false">
                <dgsec:Value>image/jpeg</dgsec:Value>
                <dgsec:Value>image/png</dgsec:Value>
                <dgsec:Value>image/gif</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="exception" userCoupled="false">
                <dgsec:Any/>
            </dgsec:Parameter>
        </dgsec:PreConditions>
        <dgsec:PostConditions>
            <dgsec:Any/>
        </dgsec:PostConditions>
    </dgsec:GetLegendGraphic>
  </dgsec:Requests>
</dgsec:OWSPolicy>
```

Im Anschluss an die Elemente <Security> und <GeneralConditions> folgt die Sektion <Request>. In ihr sind alle vom owsProxy zu bearbeitenden Anfragen aufgeführt. Fehlt eine Anfrage, obwohl der zu schützende Dienst prinzipiell in der Lage wäre sie zu bearbeiten, wird jede entsprechende Anfrage vom owsProxy als nicht gültig abgewiesen. Jeder registrierte Request verfügt über die beiden Unterelemente <PreConditions> und <PostConditions>. Innerhalb dieser Element werden die Bedingungen definiert, die darüber entscheiden, ob eine eingehende Anfrage akzeptiert oder abgewiesen wird. Die Syntax der Definitionen entspricht dabei der, wie sie für die <GeneralConditions> bereits erläutert wurde.

Die innerhalb von <PreConditions> definiert Bedingungen beziehen sich auf die eingehende Anfrage, d.h. auf die Validierung bevor der zu schützende Dienst angesprochen wird; <PostConditions> bezieht sich dagegen auf die Nachbearbeitung der vom zu schützenden Dienst gelieferten Antwort auf eine Anfrage. Am Beispiel des GetCapabilities-Requests kann dies sehr gut veranschaulicht werden.

Neben dem Servicetype (service=WMS) muss ein GetCapabilities-Request an einen OGC WMS den Parameter und REQEUST (z.B. REQEUST=GetCapabilities) enthalten. Optional ist die Verwendung von VERSION (z.B. VERSION=1.1.0) und UPDATESEQUENCE (z.B. UPDATESEQUENCE=1234). Im oben wiedergegebenen Beispiel ist in den <PreConditions> für GetCapabilities festgelegt, dass Anfragen mit dem REQEUST-Parameter 'GetCapabilities' und 'capabilities' sowie die Versionsnummern '1.1.0' und '1.1.1' akzeptiert werden. Hinsichtlich der UPDATESEQUENCE erfolgt keine Einschränkung (<Any/>).

Auf eine gültige GetCapabilities-Anfrage antwortet ein WMS mit einem Capabilities-Dokument, das u.a. eine Liste aller verfügbaren Layer enthält. Soll jedoch nur eine Untermenge diese Layer an den Anfragenden Client zurückgegeben werden, kann dies mittels der <PostConditions> festgelegt werden. Die <PostConditions> des GetCapabilities-Requests kennen den Parameter 'layers'. Über diesen kann eine Positivliste aller Layer festgelegt werden, die den owsProxy passieren, alle übrigen Layer werden aus dem Capabilities-Dokument ausgefiltert. Damit ist es z.B. möglich mehrere thematische Sichten auf einen WMS zu erzeugen oder die sichtbaren Inhalte in Abhängigkeit zum Netzsegment aus dem eine Anfrage eingeht, zu definieren. Auch kann die Liste der sichtbaren Layer an individuelle Benutzerrechte gekoppelt werden indem

```
<parameter name="layers" userCoupled="true"/>
```

definiert wird.

Sollen weder <PreConditions> noch <PostConditions> in irgendeiner Form beschränkt werden, kann anstelle der Definition der einzelnen Parameter auch das Element <Any/> verwendet werden.

Im Folgenden werden die Parameter beschrieben, für die zur Zeit Bedingungen konfiguriert werden können. Wenn nicht weiter spezifiziert ist ihre Bedeutung in den OGC WMS Spezifikationen definiert.

3.3.1 Parameter für GetCapabilities:

preConditions-Parameter:

- request: muss zur Zeit 'GetCapabilities' oder <Any/> sein
- version: zulässige Werte für die angefragte Versionsnummer
- updatesequence: zulässige Werte für die angefragte Updatesequenz

postConditions-Parameter

- layers (Liste der validen WMS-Layer): Liste der Layer, die einen owsProxy passieren dürfen

3.3.2 Parameter für GetMap

preConditions-Parameter:

- request: muss zur Zeit 'GetMap' oder <Any/> sein
- version: zulässige Werte für die angefragte Versionsnummer
- layers; Die Definition der validen Layer für einen GetMap-Request umfasst sowohl die Namen der validen Layer als auch die der für sie zulässigen Styles. Die für einen Layer gültigen Styles werden als kommaseparierte Liste getrennt durch '|' an den jeweiligen Layernamen angehängt. Der Schlüssel any legt fest, dass jeder Style für einen Layer gültig ist. Unabhängig davon, ist der default-Style grundsätzlich zulässig.

Bsp.:

```
stadtbezirke|blaugefuellt,rotumrandet

gebaeude|$any$
```

Für den Layer 'stadtbezirke' sind die Styles *blaugefuellt* und *rotumrandet* sowie der default Style zulässig, für den Layer 'gebaeude' ist jeder Style zulässig.

- bgcolor: Liste der zulässig anfragbaren Hintergrundfarben
- transparency: Liste der zulässig anfragbaren Transparenzwerte
- exception: Liste der zulässig anfragbaren exception-Formate
- bbox: die angefragte Boundingbox muss vollständig in der angegeben enthalten sein
- format: Liste der zulässig anfragbaren Formate
- maxWidth: maximal zulässig Breite der Karte in Pixel
- maxHeight: maximal zulässig Höhe der Karte in Pixel
- resolution: bestmögliche Auflösung der Karte gemessen in diagonaler Größe des durch einen Pixels abgebildeten Raumausschnitts in Metern
- sld: Liste der zulässig anfragbaren URLs über die SLDs referenziert werden dürfen
- sld_body (zur Zeit nur <Any/> möglich)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.3.3 Parameter für GetFeatureInfo

Da ein GetFeatureInfo-Request eine Kopie des vorherigen GetMap-Requests enthält, der in diesem Zusammenhang ebenfalls auf sein Gültigkeit überprüft wird, gelten gleichzeitig die für GetMap gesetzten Bedingungen.

preConditions-Parameter:

- request: muss zur Zeit 'GetFeatureInfo' oder <Any/> sein
- version: zulässige Werte für die angefragte Versionsnummer
- featureInfoLayers: Liste der Layer auf denen GetFeatureInfo-Anfragen ausgeführt werden dürfen
- infoFormat: Liste der zulässig anfragbaren Info-Formate
- maxFeatureCount: Zahl der maximal zurückgegeben FeatureInfos
- exception: Liste der zulässig anfragbaren exception-Formate

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.3.4 Parameter für GetLegendGraphic

preConditions-Parameter:

- request: muss zur Zeit 'GetLegendGraphic' oder <Any/> sein
- version: zulässige Werte für die angefragte Versionsnummer
- layers: Liste der Layer auf denen GetLegendGraphic-Anfragen ausgeführt werden dürfen
- maxWidth:maximal zulässig Breite eines Legendensymbols in Pixel
- maxHeight: maximal zulässig Höhe eines Legendensymbols in Pixel
- format: Liste der zulässig anfragbaren Formate
- exception: Liste der zulässig anfragbaren exception-Formate
- sld: Liste der zulässig anfragbaren URLs über die SLDs referenziert werden dürfen
- sld_body (zur Zeit nur <Any/> möglich)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.4 WFS Policies

Eine Policy-Datei zur Absicherung eines WFS ist hinsichtlich ihrer Konstruktion identisch zu einer WMS-Policy, lediglich die möglichen Request- und Parameterdefinition sind verschieden. Das folgende Beispiel zeigt den WFS-spezifischen Auszug aus einer entsprechenden Policy.

```
<?xml version="1.0" encoding="UTF-8"?>
<dgsec:OWSPolicy service="WFS" xmlns:dgsec="http://www.deegree.org/security"
 xmlns:xlink="http://www.w3.org/1999/xlink">
   <dgsec:Security>
<dgsec:RegistryClass>org.deegree.security.drm.SQLRegistry</dgsec:RegistryClass>
      <dgsec:ReadWriteTimeout>300</dgsec:ReadWriteTimeout>
      <dgsec:RegistryConfig>
         <dgjdbc:JDBCConnection xmlns:dgjdbc="http://www.deegree.org/jdbc">
            <dgjdbc:Driver>oracle.jdbc.OracleDriver</dgjdbc:Driver>
            <dgjdbc:Url>dgjdbc:oracle:thin:@localhost:1521:latlon</dgjdbc:Url>
            <dgjdbc:User>system</dgjdbc:User>
            <dgjdbc:Password>latlondba01</dgjdbc:Password>
            <dgjdbc:SecurityConstraints/>
            <dgjdbc:Encoding>iso-8859-1</dgjdbc:Encoding>
         </dgjdbc:JDBCConnection>
      </dgsec:RegistryConfig>
   </dgsec:Security>
   <dgsec:GeneralConditions>
      <dgsec:Conditions>
         <dgsec:Parameter name="getContentLength" userCoupled="false">
            <dgsec:Value>1000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="postContentLength" userCoupled="false">
            <dgsec:Value>10000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="header" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="requestMethod" userCoupled="false">
            <dgsec:Value>GET,POST</dgsec:Value>
         </dgsec:Parameter>
      </dgsec:Conditions>
   </dgsec:GeneralConditions>
   <dgsec:Requests>
      <dgsec:GetCapabilities>
         <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
               <dgsec:Value>GetCapabilities</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>1.1.0</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="updatesequence" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Parameter name="featureTypes" userCoupled="false">
               <dgsec:Value>{http://www.deegree.org/app}:FT1</dgsec:Value>
               <dgsec:Value>{http://www.deegree.org/app}:FT2</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PostConditions>
      </dgsec:GetCapabilities>
      <dgsec:GetFeature>
         <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
               <dgsec:Value>GetFeature</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>1.1.0</dgsec:Value>
```

```
            </dgsec:Parameter>
            <dgsec:Parameter name="featureTypes" userCoupled="false">
               <dgsec:Value>{http://www.deegree.org/app}:FT1</dgsec:Value>
               <dgsec:Value>{http://www.deegree.org/app}:FT2</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="format" userCoupled="false">
               <dgsec:Value>text/xml; subtype=gml/3.1.1</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="maxFeatures" userCoupled="false">

            <dgsec:Any/>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <deegreeSec:PostConditions>
            <deegreeSec:Parameter name="instanceFilter" userCoupled="true">
               <deegreeSec:ComplexValue>
                  <wfs:Query typeName="app:WPVS"
                     xmlns:app="http://www.deegree.org/app"
                     xmlns:wfs="http://www.opengis.net/wfs"
                     xmlns:ogc="http://www.opengis.net/ogc">
                     <ogc:Filter>
                        <ogc:Or>
                           <ogc:PropertyIsEqualTo>
                              <ogc:PropertyName>app:texturetype</ogc:PropertyName>
                              <ogc:Literal>specific</ogc:Literal>
                           </ogc:PropertyIsEqualTo>
                           <ogc:PropertyIsGreaterThanOrEqualTo>
                              <ogc:PropertyName>app:diffusecolor</ogc:PropertyName>
                              <ogc:Literal>0 0 1</ogc:Literal>
                           </ogc:PropertyIsGreaterThanOrEqualTo>
                        </ogc:Or>
                     </ogc:Filter>
                  </wfs:Query>
               </deegreeSec:ComplexValue>
               <deegreeSec:ComplexValue>
                  <wfs:Query typeName="citygml:Building"
                     xmlns:citygml="http://www.citygml.org/citygml/1/0/0"
                     xmlns:gml="http://www.opengis.net/gml"
                     xmlns:wfs="http://www.opengis.net/wfs"
                     xmlns:ogc="http://www.opengis.net/ogc">
                     <ogc:Filter>
                        <ogc:PropertyIsEqualTo>
                           <ogc:PropertyName>gml:name</ogc:PropertyName>
                           <ogc:Literal>Wirtschaftsgebäude</ogc:Literal>
                        </ogc:PropertyIsEqualTo>
                     </ogc:Filter>
                  </wfs:Query>
               </deegreeSec:ComplexValue>
            </deegreeSec:Parameter>
         </deegreeSec:PostConditions>
      </dgsec:GetFeature>
      <dgsec:DescribeFeatureType>
         <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
               <dgsec:Value>DescribeFeatureType</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>1.1.0</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="format" userCoupled="false">
               <dgsec:Value>text/xml; subtype=gml/3.1.1</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="featureTypes" userCoupled="true"/>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:DescribeFeatureType>
   </dgsec:Requests>
```

```
</dgsec:OWSPolicy>
```

3.4.1 Paremeter für GetCapabilities:

preConditions-Parameter:

- request: muss zur Zeit 'GetCapabilities' oder <Any/> sein
- version: zulässige Werte für die angefragteVersionsnummer
- updatesequence: zulässige Werte für die angefragte Updatesequence

postConditions-Parameter

- featureTypes: Liste von validen i.d.R. Qualifizierten FeatureType-Namen ({namespace}:Name)

3.4.2 Parameter für GetFeature

Zur Zeit besteht keine Möglichkeit den ggf. mit einem GetFeature-Request verbundenen Filterausdruck auf seine inhaltliche Gültigkeit zu überprüfen, da dies ausgesprochen aufwändig ist. In in einer zukünftigen Version des owsProxy wird es jedoch möglich sein, für ausgewählte FeatureTypes OGC:Filter zu definieren, die über eine UND-Verknüpfung mit der eingehenden Anfrage verknüpft werden und so die Menge der zugreifbaren Feature allgemein oder nutzerspezifisch beschränkt.

preConditions-Parameter:

- request: muss zur Zeit 'GetFeature' oder <Any/> sein
- version: zulässige Werte für die angefragteVersionsnummer
- featureTypes: Liste von validen i.d.R. Qualifizierten FeatureType-Namen ({namespace}:Name)
- format: zulässige Anfrageformate
- maxFeatures

postConditions-Parameter

- instanceFilter: OGC WFS Queries über angegeben wird, welche Featureinstanzen den owsProxy passieren dürfen (siehe Beispiel oben)
- horizontalAccuracy: maximal zulässige horizontale Genauigkeit
- verticalAccuracy: maximal zulässige vertikale GenauigkeitXSLT: Referenz auf ein XSLT-Skript, dass ggf. Komplexere Filterungen durchführt

3.4.3 Parameter für DescribeFeatureType

preConditions-Parameter:

- request: muss zur Zeit 'DescribeFeatureType' oder <Any/> sein
- version: zulässige Werte für die angefragteVersionsnummern
- featureTypes (entspricht typeName in der WFS Spezifikation)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.4.4 Parameter für WFS_Insert

PreConditions-Parameter:

- typeName: Liste von validen i.d.R. Qualifizierten FeatureType-Namen ({namespace}:Name)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.4.5 Parameter für WFS_Update

PreConditions-Parameter:

- typeName: Liste von validen i.d.R. Qualifizierten FeatureType-Namen ({namespace}:Name)

postConditions-Parameter

- instanceFilter: OGC WFS Queries über angegeben wird, welche Featureinstanzen den owsProxy passieren dürfen (siehe Beispiel oben)

3.4.6 Parameter für WFS_Delete

PreConditions-Parameter:

- typeName: Liste von validen i.d.R. Qualifizierten FeatureType-Namen ({namespace}:Name)

postConditions-Parameter

- instanceFilter: OGC WFS Queries über angegeben wird, welche Featureinstanzen den owsProxy passieren dürfen (siehe Beispiel oben)

3.4.7 Usercoupled postConditions

Werden postConditions für GetFeature, WFS_Delete und WFS_Update userCoupled definiert, müssen die über den Parameter instanceFilter definierten Queries in einen kapselnden XML-Ausdruck eingeschlossen werden.

```
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
   <ogc:And>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>horizontalAccuracy</ogc:PropertyName>
         <ogc:Literal>10</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>verticalAccuracy</ogc:PropertyName>
         <ogc:Literal>1</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>xslt</ogc:PropertyName>
         <ogc:Literal>file:/d:/temp/test.xsl</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>instanceFilter</ogc:PropertyName>
         <ogc:Literal>
             <![CDATA[<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc"
                xmlns:app="http://www.deegree.org/app">
            <ogc:And>
               <ogc:PropertyIsGreaterThan>
                  <ogc:PropertyName>app:dateOfBirth</ogc:PropertyName>
                  <ogc:Literal>1820</ogc:Literal>
               </ogc:PropertyIsGreaterThan>
               <ogc:PropertyIsEqualTo>
                   <ogc:PropertyName>
                      app:placeOfBirth/app:Place/app:country/app:Country/app:name
                    </ogc:PropertyName>
                  <ogc:Literal>Germany</ogc:Literal>
               </ogc:PropertyIsEqualTo>
            </ogc:And>
      </ogc:Filter>]]></ogc:Literal>
      </ogc:PropertyIsEqualTo>
   </ogc:And>
</ogc:Filter>
```

Wie zusehen ist das Literal der letzten PropertyIsEqualTo-Expression ein in eine CDATA-Section eingebetteter Filterausdruck. Die CDATA-Section ist hier zwingend, da ein Literal nur einen String enthalten darf; die Interpretation als Filterausdruck ist Aufgabe des owsProxies.

3.5 CSW Policies

Eine Policy-Datei zur Absicherung eines CSW ist hinsichtlich ihrer Konstruktion identisch zu den übrigen Policy-dateien, lediglich die möglichen Request- und Parameterdefinition sind verschieden. Das folgende Beispiel zeigt den CSW-spezifischen Auszug aus einer entsprechenden Policy.

```
<?xml version="1.0" encoding="UTF-8"?>
<dgsec:OWSPolicy xmlns:dgsec="http://www.deegree.org/security" service="CSW">
   <dgsec:Security>
<dgsec:RegistryClass>org.deegree.security.drm.SQLRegistry</dgsec:RegistryClass>
      <dgsec:ReadWriteTimeout>300</dgsec:ReadWriteTimeout>
      <dgsec:RegistryConfig>
         <dgjdbc:JDBCConnection xmlns:dgjdbc="http://www.deegree.org/jdbc">
            <dgjdbc:Driver>oracle.jdbc.OracleDriver</dgjdbc:Driver>
```

```
            <dgjdbc:Url>dgjdbc:oracle:thin:@localhost:1521:latlon</dgjdbc:Url>
            <dgjdbc:User>system</dgjdbc:User>
            <dgjdbc:Password>latlondba01</dgjdbc:Password>
            <dgjdbc:SecurityConstraints/>
            <dgjdbc:Encoding>iso-8859-1</dgjdbc:Encoding>
         </dgjdbc:JDBCConnection>
      </dgsec:RegistryConfig>
   </dgsec:Security>
   <dgsec:GeneralConditions>
      <dgsec:Conditions>
         <dgsec:Parameter name="getContentLength" userCoupled="false">
            <dgsec:Value>1000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="postContentLength" userCoupled="false">
            <dgsec:Value>1000000</dgsec:Value>
         </dgsec:Parameter>
         <dgsec:Parameter name="header" userCoupled="false">
            <dgsec:Any/>
         </dgsec:Parameter>
         <dgsec:Parameter name="requestMethod" userCoupled="false">
            <dgsec:Value>GET,POST</dgsec:Value>
         </dgsec:Parameter>
      </dgsec:Conditions>
   </dgsec:GeneralConditions>
   <dgsec:Requests>
      <dgsec:GetCapabilities>
         <dgsec:PreConditions>
            <dgsec:Parameter name="request" userCoupled="false">
               <dgsec:Value>GetCapabilities</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>2.0.0</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="updatesequence" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter name="sections" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:GetCapabilities>
      <dgsec:DescribeRecord>
         <dgsec:PreConditions>
            <dgsec:Parameter userCoupled="false" name="typeName">
               <dgsec:Value>csw:record</dgsec:Value>
               <dgsec:Value>csw:profile</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter userCoupled="false" name="schemaLanguage">
               <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter userCoupled="false" name="outputFormat">
               <dgsec:Value>text/xml</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:DescribeRecord>
      <dgsec:GetRecords>
         <dgsec:PreConditions>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>2.0.0</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="maxRecords" userCoupled="false">
               <dgsec:Value>1000</dgsec:Value>
            </dgsec:Parameter>
```

```
            <dgsec:Parameter name="metadataFormat" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter name="outputFormat" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter name="outputSchema" userCoupled="false">
               <dgsec:Any/>
            </dgsec:Parameter>
            <dgsec:Parameter name="resultType" userCoupled="false">
               <dgsec:Value>HITS</dgsec:Value>
               <dgsec:Value>hits</dgsec:Value>
               <dgsec:Value>RESULTS</dgsec:Value>
               <dgsec:Value>results</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="sortBy" userCoupled="false">
               <dgsec:Value>Title</dgsec:Value>
               <dgsec:Value>Date</dgsec:Value>
               <dgsec:Value>Envelope</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="typeNames" userCoupled="false">
               <dgsec:Value>csw:dataset</dgsec:Value>
               <dgsec:Value>csw:datasetcollection</dgsec:Value>
               <dgsec:Value>csw:service</dgsec:Value>
               <dgsec:Value>csw:application</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="elementSetName" userCoupled="false">
               <dgsec:Value>brief</dgsec:Value>
               <dgsec:Value>summary</dgsec:Value>
               <dgsec:Value>full</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:GetRecords>
      <dgsec:GetRecordById>
         <dgsec:PreConditions>
            <dgsec:Parameter name="version" userCoupled="false">
               <dgsec:Value>2.0.0</dgsec:Value>
            </dgsec:Parameter>
            <dgsec:Parameter name="elementSetName" userCoupled="false">
               <dgsec:Value>brief</dgsec:Value>
               <dgsec:Value>summary</dgsec:Value>
               <dgsec:Value>full</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:GetRecordById>
      <dgsec:CSW_Insert>
         <dgsec:PreConditions>
            <dgsec:Parameter userCoupled="false" name="metadataFormat">

   <dgsec:Value>{http://metadata.dgiwg.org/smXML}:MD_Metadata</dgsec:Value>

   <dgsec:Value>{http://schemas.opengis.net/iso19115full}:MD_Metadata</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:CSW_Insert>
      <dgsec:CSW_Update>
         <dgsec:PreConditions>
            <dgsec:Parameter userCoupled="false" name="metadataFormat">
      <dgsec:Value>{http://metadata.dgiwg.org/smXML}:MD_Metadata</dgsec:Value>
   <dgsec:Value>{http://schemas.opengis.net/iso19115full}:MD_Metadata</dgsec:Value>
```

```
            </dgsec:Parameter>
            <dgsec:Parameter userCoupled="false" name="typeName">
               <dgsec:Value>csw:dataset</dgsec:Value>
               <dgsec:Value>csw:datasetcollection</dgsec:Value>
               <dgsec:Value>csw:series</dgsec:Value>
               <dgsec:Value>csw:application</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:CSW_Update>
      <dgsec:CSW_Delete>
         <dgsec:PreConditions>
            <dgsec:Parameter userCoupled="false" name="typeName">
               <dgsec:Value>csw:dataset</dgsec:Value>
               <dgsec:Value>csw:datasetcollection</dgsec:Value>
               <dgsec:Value>csw:series</dgsec:Value>
               <dgsec:Value>csw:application</dgsec:Value>
            </dgsec:Parameter>
         </dgsec:PreConditions>
         <dgsec:PostConditions>
            <dgsec:Any/>
         </dgsec:PostConditions>
      </dgsec:CSW_Delete>
   </dgsec:Requests>
</dgsec:OWSPolicy>
```

Die Sicherung der Anfragen GetCapabilities, GetRecords, DescribeRecord und GetRecordById wird unterstützt. Zusätzlich ist die Absicherung transaktionaler Anfragen möglich. Da sich diese in Insert-, Update- und Delete-Operationen gliedern, werden für jede dieser Operationen separate Gültigkeitsbereiche für die einzelnen Parameter definiert.

3.5.1 Parameter für GetCapabilities

preConditions-Parameter:

- request
- version
- updatesequence

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.2 Parameter für DescribeRecord (zur Zeit nur partiell unterstützt)

preConditions-Parameter:

- version
- typeName

- schemaLanguage
- outputFormat

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.3 Parameter für GetRecords

Zur Zeit besteht keine Möglichkeit den ggf. mit einem GetRecords-Request verbundenen Filterausdruck auf seine inhaltliche Gültigkeit zu überprüfen, da dies ausgesprochen aufwändig ist. In einer zukünftigen Version des owsProxy wird es jedoch möglich sein, für ausgewählte FeatureTypes OGC:Filter zu definieren, die über eine UND-Verknüpfung mit der eingehenden Anfrage verknüpft werden und so die Menge der zugreifbaren Feature allgemein oder nutzerspezifisch beschränkt.

preConditions-Parameter:

- request: muss zur Zeit 'GetRecords' oder <Any/> sein
- version: zulässige Werte für die angefragteVersionsnummer
- maxRecords: Maximale Anzahl der zulässig abfragbaren Records
- metadataFormat: Liste der zulässig abfragbaren Metadatenformate
- outputFormat: Liste der zulässig abfragbaren Formate (z.B: text/xml)
- outputSchema:
- resultType: Liste der anfragbaren ResultTypes (HITS | RESULTS)
- sortBy: Liste der Properties (voll qualifizierte Namen) nach denen Ergebnisse sortiert werden dürfen
- typeNames: Liste der anfragbaren typeNames (z.B. csw:record)
- elementSetName: Liste der anfragbaren elementSetNames (z.B. full)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.4 Parameter für GetRecordById

preConditions-Parameter:

- version: zulässige Werte für die angefragteVersionsnummer
- elementSetName: Liste der anfragbaren elementSetNames (z.B. full)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.5 Parameter für CSW_Insert

preConditions-Parameter:

- metadataFormat

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.6 Parameter für CSW_Update

preConditions-Parameter:

- metadataFormat: Liste der zulässig abfragbaren Metadatenformate
- typeName: Liste der anfragbaren typeNames (z.B. csw:record)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

3.5.7 Parameter für CSW_Delete

preConditions-Parameter:

- typeName: Liste der anfragbaren typeNames (z.B. csw:record)

postConditions-Parameter

- zur Zeit keine (<Any/> muss gesetzt sein)

4 Anhang A (XML-Schema Dateien)

4.1 Allgemein

```
<xs:schema targetNamespace="http://www.deegree.org/jdbc"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:jdbc="http://www.deegree.org/jdbc" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:complexType name="JDBCConnectionType">
      <xs:annotation>
         <xs:documentation>
       For JDBC based Datastores only, contains the necessary connection
information.
       </xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="Driver" type="xs:string"/>
         <xs:element name="Url" type="xs:string"/>
         <xs:element name="User" type="xs:string" minOccurs="0"/>
         <xs:element name="Password" type="xs:string" minOccurs="0"/>
         <xs:element name="SecurityConstraints" minOccurs="0"/>
         <xs:element name="Encoding" minOccurs="0"/>
         <xs:element name="AliasPrefix" minOccurs="0"/>
         <xs:element name="SDEDatabase" minOccurs="0">
            <xs:annotation>
               <xs:documentation>
                        Only applies to ArcSDE connections.
                     </xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="SDEVersion" minOccurs="0">
            <xs:annotation>
               <xs:documentation>
                        Only applies to ArcSDE connections.
                     </xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="OracleWorkspace" minOccurs="0">
            <xs:annotation>
               <xs:documentation>
          Name of the workspace to switch to on connect. Only applies to Oracle
10g. Default workspace is "LIVE".
          </xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="JDBCConnection" type="jdbc:JDBCConnectionType"/>
</xs:schema>

<xs:schema targetNamespace="http://www.deegree.org/security"
xmlns:jdbc="http://www.deegree.org/jdbc"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:deegree="http://www.deegree.org/security"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <!-- import OGC commons to use some basic types-->
   <xs:import namespace="http://www.w3.org/1999/xlink"
schemaLocation="./xlinks.xsd"/>
   <xs:import namespace="http://www.deegree.org/jdbc"
schemaLocation="./jdbc_connection.xsd"/>
   <xs:element name="Policy" type="deegree:PolicyType">
      <xs:annotation>
         <xs:documentation>root element of the deegree policy document for
protecting Web map Services</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:complexType name="PolicyType">
```

```
      <xs:sequence>
         <xs:element name="Security" type="deegree:SecurityType" minOccurs="0"/>
         <xs:element name="GeneralConditions" type="deegree:GeneralConditionsType"
minOccurs="0"/>
         <xs:element name="Requests" type="deegree:RequestsType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="service" type="xs:string" use="required"/>
   </xs:complexType>
   <xs:complexType name="RequestsType">
      <xs:sequence>
         <xs:element ref="deegree:_Request" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SecurityType">
      <xs:annotation>
         <xs:documentation>cofiguration off the access to the security management
system/database</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="RegistryClass" type="xs:string"/>
         <xs:element name="ReadWriteTimeout" type="xs:int" default="500"/>
         <xs:element name="RegistryConfig" type="deegree:RegistryConfigType"/>
         <xs:element name="AuthentificationSettings"
type="deegree:AuthentificationSettingsType" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="RegistryConfigType">
      <xs:sequence>
         <xs:element ref="jdbc:JDBCConnection" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="GeneralConditionsType">
      <xs:choice minOccurs="0">
         <xs:element name="Conditions" type="deegree:ConditionType" minOccurs="0"/>
         <xs:element name="any" type="xs:string" minOccurs="0">
            <xs:annotation>
               <xs:documentation>the presents of this element indicates that no
constraints are made</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:choice>
   </xs:complexType>
   <xs:complexType name="_RequestType" abstract="true">
      <xs:annotation>
         <xs:documentation>A concrete _requestType will encapsulate and describe the
conditions for
         a request and the repsonse to it for a OWS.  Also a concrete _requestType
must be named like
         the request which conditions will be described</xs:documentation>
      </xs:annotation>
      <xs:choice minOccurs="0">
         <xs:sequence>
            <xs:element name="PreConditions" type="deegree:ConditionType"
minOccurs="0"/>
            <xs:element name="PostConditions" type="deegree:ConditionType"
minOccurs="0"/>
         </xs:sequence>
         <xs:element name="Any" type="xs:string" minOccurs="0">
            <xs:annotation>
               <xs:documentation>the presents of this element indicates that no
constraints are made</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:choice>
   </xs:complexType>
   <xs:element name="_Request" type="deegree:_RequestType" abstract="true"/>
   <xs:complexType name="AuthentificationSettingsType">
      <xs:sequence>
```

```
         <xs:element name="AuthentificationService"
type="deegree:AuthentificationServiceType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="AuthentificationServiceType">
      <xs:sequence>
         <xs:element name="OnlineResource" type="deegree:OnlineResourceType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="OnlineResourceType">
      <xs:attributeGroup ref="xlink:simpleLink"/>
   </xs:complexType>
   <xs:complexType name="GetCapabilitiesType">
      <xs:annotation>
         <xs:documentation>GetCapabilties will be used by all OWS so it defined in
the base schema</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetCapabilities" type="deegree:GetCapabilitiesType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="ConditionType">
      <xs:annotation>
         <xs:documentation>If nor parameter or any is defined no request will
fullfill a condintion. This realizes
         the concept that a security filter will block everything except it
especially allowed</xs:documentation>
      </xs:annotation>
      <xs:choice minOccurs="0">
         <xs:element ref="deegree:_Parameter" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="Any" type="xs:string" minOccurs="0">
            <xs:annotation>
               <xs:documentation>the presents of this element indicates that no
constraints are made</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:choice>
   </xs:complexType>
   <xs:element name="_Condition" type="deegree:ConditionType" abstract="true"/>
   <xs:complexType name="ParameterType">
      <xs:choice>
         <xs:sequence>
            <xs:element name="Value" type="xs:string" minOccurs="0"
maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>Unordered list of all the valid values for this
parameter or other quantity.
                     For those parameters that contain a list or sequence of
values, these values shall be for
                     individual values in the list. The allowed set of values and
the allowed server restrictions
                     on that set of values shall be specified in the
Implementation Specification for this service. </xs:documentation>
               </xs:annotation>
            </xs:element>
            <xs:element name="ComplexValue" minOccurs="0" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>
                     a condition may contains complex values like filter expressions
that can be put
                     into a ComplexValue element as part of a condition
                  </xs:documentation>
               </xs:annotation>
               <xs:complexType name="ExtensionType">
                  <xs:sequence>
                     <xs:any namespace="##any"/>
                  </xs:sequence>
```

```
            </xs:complexType>
         </xs:element>
      </xs:sequence>
      <xs:element name="Any" type="xs:string" minOccurs="0">
         <xs:annotation>
            <xs:documentation>the presents of this element indicates that no
constraints are made on a parameter</xs:documentation>
         </xs:annotation>
      </xs:element>
   </xs:choice>
   <xs:attribute name="userCoupled" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:element name="_Parameter" type="deegree:ParameterType" abstract="true"/>
</xs:schema>
```

4.2 WMS Policy

```
<xs:schema targetNamespace="http://www.deegree.org/security"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:deegree="http://www.deegree.org/security" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:include schemaLocation="deegreePolicy.xsd"/>
   <xs:element name="OWSPolicy" type="deegree:PolicyType">
      <xs:annotation>
         <xs:documentation>root element of the deegree policy document for
protecting Web map Services</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:complexType name="GetMapType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetMap" type="deegree:GetMapType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="GetFeatureInfoType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetFeatureInfo" type="deegree:GetFeatureInfoType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="GetLegendGraphicType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetLegendGraphic" type="deegree:GetLegendGraphicType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="GetScaleBarType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetScaleBar" type="deegree:GetScaleBarType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="WMSParameterType">
      <xs:complexContent>
         <xs:extension base="deegree:ParameterType">
            <xs:attribute name="name" type="deegree:WmsParamNameType"
use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="Parameter" type="deegree:WMSParameterType"
substitutionGroup="deegree:_Parameter"/>
   <xs:simpleType name="WmsParamNameType">
      <xs:restriction base="xs:string">
```

```
            <xs:enumeration value="requestMethod"/>
            <xs:enumeration value="getContentLength"/>
            <xs:enumeration value="postContentLength"/>
            <xs:enumeration value="header"/>
            <xs:enumeration value="request"/>
            <xs:enumeration value="version"/>
            <xs:enumeration value="updatesequence"/>
            <xs:enumeration value="layers"/>
            <xs:enumeration value="bbox"/>
            <xs:enumeration value="resolution"/>
            <xs:enumeration value="maxWidth"/>
            <xs:enumeration value="maxHeight"/>
            <xs:enumeration value="transparency"/>
            <xs:enumeration value="bgcolor"/>
            <xs:enumeration value="color"/>
            <xs:enumeration value="labelColor"/>
            <xs:enumeration value="bottomLabel"/>
            <xs:enumeration value="topLabel"/>
            <xs:enumeration value="maxSize"/>
            <xs:enumeration value="sld"/>
            <xs:enumeration value="sld_body"/>
            <xs:enumeration value="format"/>
            <xs:enumeration value="exception"/>
            <xs:enumeration value="featureInfoLayers"/>
            <xs:enumeration value="infoFormat"/>
            <xs:enumeration value="maxFeatureCount"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:schema>
```

4.3 WFS Policy

```
<xs:schema targetNamespace="http://www.deegree.org/security"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:deegree="http://www.deegree.org/security" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:include schemaLocation="deegreePolicy.xsd"/>
   <xs:element name="OWSPolicy" type="deegree:PolicyType">
      <xs:annotation>
         <xs:documentation>root element of the deegree policy document for
protecting Web map Services</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:complexType name="GetFeatureType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetFeature" type="deegree:GetFeatureType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="DescribeFeatureTypeType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="DescribeFeatureType" type="deegree:DescribeFeatureTypeType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="WFSParameterType">
      <xs:complexContent>
         <xs:extension base="deegree:ParameterType">
            <xs:attribute name="name" type="deegree:WFSParamNameType"
use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="InsertType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
```

```
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="WFS_Insert" type="deegree:InsertType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="UpdateType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="WFS_Update" type="deegree:UpdateType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="DeleteType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="WFS_Delete" type="deegree:DeleteType"
substitutionGroup="deegree:_Request"/>
   <xs:element name="Parameter" type="deegree:WFSParameterType"
substitutionGroup="deegree:_Parameter"/>
   <xs:simpleType name="WFSParamNameType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="requestMethod"/>
         <xs:enumeration value="getContentLength"/>
         <xs:enumeration value="postContentLength"/>
         <xs:enumeration value="header"/>
         <xs:enumeration value="request"/>
         <xs:enumeration value="version"/>
         <xs:enumeration value="updatesequence"/>
         <xs:enumeration value="featureTypes"/>
         <xs:enumeration value="bbox"/>
         <xs:enumeration value="format"/>
         <xs:enumeration value="exception"/>
         <xs:enumeration value="maxFeatures"/>
         <xs:enumeration value="typeName"/>
         <xs:enumeration value="resultType"/>
         <xs:enumeration value="traverseXLinkDepth"/>
         <xs:enumeration value="traverseXLinkExpiry"/>
         <xs:enumeration value="instanceFilter"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

4.4 CSW Policy

```
<xs:schema targetNamespace="http://www.deegree.org/security"
xmlns:deegree="http://www.deegree.org/security"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:include schemaLocation="deegreePolicy.xsd"/>
   <xs:element name="OWSPolicy" type="deegree:PolicyType">
      <xs:annotation>
         <xs:documentation>root element of the deegree policy document for
protecting Catalogue Service Web Profile</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:complexType name="GetRecordsType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="GetRecords" type="deegree:GetRecordsType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="GetRecordByIdType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
```

```
   <xs:element name="GetRecordById" type="deegree:GetRecordByIdType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="DescribeRecordType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="DescribeRecord" type="deegree:DescribeRecordType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="InsertType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="CSW_Insert" type="deegree:InsertType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="UpdateType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="CSW_Update" type="deegree:UpdateType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="DeleteType">
      <xs:complexContent>
         <xs:extension base="deegree:_RequestType"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="CSW_Delete" type="deegree:DeleteType"
substitutionGroup="deegree:_Request"/>
   <xs:complexType name="CSWParameterType">
      <xs:complexContent>
         <xs:extension base="deegree:ParameterType">
            <xs:attribute name="name" type="deegree:CSWParamNameType"
use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="Parameter" type="deegree:CSWParameterType"
substitutionGroup="deegree:_Parameter"/>
   <xs:simpleType name="CSWParamNameType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="requestMethod"/>
         <xs:enumeration value="getContentLength"/>
         <xs:enumeration value="postContentLength"/>
         <xs:enumeration value="header"/>
         <xs:enumeration value="request"/>
         <xs:enumeration value="version"/>
         <xs:enumeration value="updatesequence"/>
         <xs:enumeration value="resultType"/>
         <xs:enumeration value="maxRecords"/>
         <xs:enumeration value="namespace"/>
         <xs:enumeration value="outputFormat"/>
         <xs:enumeration value="outputSchema"/>
         <xs:enumeration value="metadataFormat"/>
         <xs:enumeration value="typeNames"/>
         <xs:enumeration value="elementSetName"/>
         <xs:enumeration value="sortBy"/>
         <xs:enumeration value="sections"/>
         <xs:enumeration value="distributedSearch"/>
         <xs:enumeration value="responseHandler"/>
         <xs:enumeration value="typeName"/>
         <xs:enumeration value="schemaLanguage"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```