Open Geospatial Consortium

Date: 2012-01-25

Reference number of this document: OGC 11-086r1

Category: OGC Public Engineering Report

Editor(s): Jan Herrmann
Andreas Matheus

OGC® OWS-8 Aviation Thread - Authoritative AIXM Data Source Engineering Report

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is 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, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OGC Public Engineering Report
Document stage:	Approved for Public Release
Document language:	English

Table of contents

Figures

Page

Tables

Page

Listings

i. Preface

This engineering report was prepared as a deliverable for the OGC Web Services, Phase 8 (OWS-8) initiative of the OGC Interoperability Program. This document presents the results of the authoritative data source work within the OWS-8 aviation thread. It describes how to provide access control for WFS-T v2.0 instances serving as authoritative AIXM data sources.

This document is a deliverable for the OGC Web Services 8 (OWS-8) testbed activity. OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver proven candidate standards or revisions to existing standards into OGC's Standards Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

The OWS-8 sponsors are organizations seeking open standards for their interoperability requirements. After analyzing their requirements, the OGC Interoperability Team recommend to the sponsors that the content of the OWS-8 initiative be organized around the following threads:

* Observation Fusion

* Geosynchronization (Gsync)

* Cross-Community Interoperability (CCI)

* Aviation

More information about the OWS-8 testbed can be found at:

http://www.opengeospatial.org/standards/requests/74

OGC Document [11-139] “OWS-8 Summary Report” provides a summary of the OWS-8 testbed and is available for download:

https://portal.opengeospatial.org/files/?artifact_id=46176

License Agreement

ii. Document Terms and Definitions

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

iii. Submission and Contribution Contact Points

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

Name	Organization
Jan Herrmann	Technische Universität München
Andreas Matheus	Universität der Bundeswehr München

iv. Revision History

Date	Release	Editor	Primary clauses modified	Description
2011/06/26	0.0.1	JH	All	draft of intended structure
2011/08/28	0.4.0	JH	All	initial writing
2011/09/01	0.5.0	AM	10	
2011/09/05	0.6.0	JH	All	bug fixes, references
2011/09/06	0.7.0	AM	All	
2011/09/06	1.0.0	JH	Appendix C	adding of the GeoXACML policy

v. Changes to the OGC Abstract Specification

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

vi. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

OWS-8 Testbed

OWS testbeds are part of OGC's Interoperability Program, a global, hands-on and collaborative prototyping program designed to rapidly develop, test and deliver Engineering Reports into OGC's Specification Program, where they are formalized for public release. In OGC's Interoperability Initiatives, international teams of technology providers work together to solve specific geoprocessing interoperability problems posed by the Initiative's sponsoring organizations. OGC Interoperability Initiatives include test beds, pilot projects, interoperability experiments and interoperability support services - all designed to encourage rapid development, testing, validation and adoption of OGC standards.

In November 2010, the OGC issued a call for sponsors for an OGC Web Services, Phase 8 (OWS-8) Testbed activity. The activity completed in September 2011. This engineering Report describes work produced within the Aviation Thread of OWS-8.

This thread builds on the Aeronautical Information Management (AIM) and Aviation threads of OWS-6 and OWS-7 respectively, and seeks to further develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Information Exchange Model (WXXM) in an OGC Web Services environment (cp. [19], 4.5).

AIXM and WXXM are developed by FAA and EUROCONTROL as global standards for the representation and exchange of aeronautical and weather information, respectively. Both models were designed as a basis for enabling the transition to a net-centric, global interoperable Air Transport System (ATS). FAA and EUROCONTROL seek to leverage the process and results of the OWS-8 Aviation Thread in their efforts to increase industry adoption of AIXM and WXXM, and to support the operational use and validation of these emerging standards. Both agencies also plan to use those standards in their System Wide Information Management (SWIM)-related components of the US NextGen and EU SESAR programs.

In OWS-8, one goal of the Aviation Thread is to address how to implement an Authoritative Data Source for AIXM data managed and served by WFS v2.0 instances. Focus of the related work items is the enforcement of fine-grained access rights referring to transactional WFS-T operations.

OWS-8 Aviation Thread - Authoritative AIXM Data Source Engineering Report

1 Overview

This engineering report describes how to provide access control for WFS-T 2.0 instances in the OWS-8 Authoritative AIXM Data Source scenario.

Section 6 provides some background information on popular conceptual rights models and introduces various types of rights that usually need to be enforced in OpenGIS Web Feature Service (WFS) based architectures. Section 6 continues with an analysis of existing rights models under the given requirements. The conclusion of this analysis is that an access control system for WFS instances needs to use a hybrid rights model that combines the ideas of expressive rule- and role-based rights models.

Section 7 captures how to integrate rule- and role-based access control systems in the overall system architecture and describes the architecture of and information flow within these systems.

Section 8 shows how the needed hybrid rights model can be implemented based on the OASIS XACML specification [5], the OGC GeoXACML specification [8][9] and related XACML profiles like the XACML v3.0 Multiple Decision Profile [23], the XACML v3.0 Hierarchical Resource profile [22], the XACML v3.0 RBAC Profile [21] and the XACML v3.0 OWS profile [25]. For each required right type examples are given that highlight the techniques how to use the languages defined in the mentioned specifications in the WFS use case.

Section 9 describes the evaluation of the access control solution presented in section 7 and 8 in the Authoritative AIXM Data Source use case. In this context we introduce some sample business rules provided by the Federal Aviation Authority (FAA) and their XACML compliant implementation.

Section 10 briefly describes the implementation of the access control system components and of the demo client.

Section 11 concludes this report by providing a short summary and a list of important work items that need to be addressed next.

2 Bibliography

[1] E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Mitre technical report 2547, May 1973.

[2] E. Bell and L. J. LaPadula. Secure computer systems: Unified exposition and multics interpretation. Mitre technical report 2997, May 1975.

[3] K. J. Biba. Integrity considerations for secure computer systems. Mitre technical report 3153, April 1975.

[4] Core and hierarchical role based access control (RBAC) profile of XACML v2.0. RBAC profile. OASIS Standard. 01 February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf

[5] eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Committee Specification 02. August 2011.

[6] eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard. 01 February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[7] M. Carrie Garceau. Sample Business Rules for SAA Scheduling Scenario - Version 0.4. FAA, Air Traffic Organization, Mission Support Services Aeronautical Information Management (AJV-2), July 2011.

[8] Geospatial eXtensible Access Control Markup Language (GeoXACML), OGC Implementation Standard. 20 February 2008. http://www.opengeospatial.org/standards/geoxacml.

[9] Geospatial eXtensible Access Control Markup Language (GeoXACML) v3.0, OGC Implementation Standard Draft. August 2011.

[10] J. Herrmann. Access Control in Service-oriented Architectures applied to spatial data infrastructures. PhD thesis, Technische Universität München, Germany, September 2011 (expected).

[11] J. Herrmann and A. Matheus. OWS-6 GeoXACML engineering report. OGC public engineering report, Open Geospatial Consortium (OGC), July 2009.

[12] Hierarchical resource profile of XACML v2.0, OASIS Standard. 01 February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-hier-profile-spec-os.pdf.

[13] International Committee for Information Technology Standards. Information technology - role based access control (rbac). Ansi/incits standard, InterNational Committee for Information Technology Standards (INCITS), 2004.

[14] ISO10181-3 ISO/IEC 10181-3:1996 Information technology – Open Systems Interconnection -- Security frameworks for open systems: Access control framework.

[15] Multiple resource profile of XACML v2.0, OASIS Standard, 01 February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-mult-profile-spec-os.pdf.

[16] P. Needham. Oracle label security with oracle database 11g release 2. ORACLE White Paper, 2009.

[17] OGC, Open Geospatial Consortium Inc.: OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture, Version: 1.2.0, Date: 2006-10-05, http://portal.opengeospatial.org/files/?artifact_id=18241

[18] Cristian Opincaru. Service Oriented Security Architecture applied to Spatial Data Infrastructures., Munich 2008. http://deposit.ddb.de/cgi-bin/dokserv?idn =988029642&dok_var=d1&dok_ext=pdf&filename=988029642.pdf.

[19] Request for Quotation (RFQ) And Call for Participation (CFP) OGC Web Services Initiative - Phase 8 (OWS-8) - Annex B OWS-8 Architecture, OGC. November 2010. http://portal.opengeospatial.org/files/?artifact_id=41688.

[20] RFC3198 IETF RFC 3198: Terminology for Policy-Based Management, November 2001. http://www.ietf.org/rfc/rfc3198.txt

[21] XACML v3.0 Core and Hierarchical Role Based Access Control (RBAC) Profile v1.0. OASIS Committee Specification 02. August 2011.

[22] XACML v3.0 Hierarchical Resource Profile v1.0. OASIS Committee Specification 02. August 2011.

[23] XACML v3.0 Multiple Decision Profile v1.0. OASIS Committee Specification 02. August 2011.

[24] XACML v2.0 OGC Web Service Profile v0.8. OGC Draft Specification. August 2011.

[25] XACML v3.0 OGC Web Service Profile v0.8. OGC Draft Specification. August 2011.

[26] XACML v3.0 OGC Web Service Profile v0.8 - Extension WFS 2.0. OGC Draft Specification. August 2011.

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. Please note that some terms and definitions are taken from the XACML specification [6] and the XACML v3.0 Multiple Decision Profile v1.0 [23] and are included here for easy reading.

Access control - Controlling access in accordance with a policy

Action - An operation on a resource

(XACML) Attribute - Characteristic of an entity that may be referenced in a predicate or target. A specific instance of an attribute, determined by the attribute name and type, the identity of the attribute holder and (optionally) the identity of the issuing authority

(XACML) Authorization decision - The result of evaluating applicable policy, returned by the PDP to the PEP. A function that evaluates to “Permit”, “Deny”, “Indeterminate” or “NotApplicable", and (optionally) a set of obligations

(XACML) Authorization Decision Request (ADR) - The request by a PEP or Context Handler to a PDP to render an authorization decision

Bag – An unordered collection of values, in which there may be duplicate values

Condition - An expression of predicates. A function that evaluates to "True", "False" or “Indeterminate”

Context Handler - The system entity that converts decision requests in the native request format to the XACML canonical form and converts authorization decisions in the XACML canonical form to the native response format

(XACML) evaluation context - The canonical representation of a decision request and an authorization decision

Effect - The intended consequence of a satisfied rule (either "Permit" or "Deny")

Environment - The set of attributes that are relevant to an authorization decision and are independent of a particular subject, resource or action

Global Authorization Decision Request (global A.D.R.) – an access control decision request referring to one or multiple resources

Global Authorization Decision Response – an aggregation of individual access control decision responses

Individual Authorization Decision Request (individual A.D.R.) – a decision request referring to exactly one resource node

Individual Authorization Decision Response – a decision response referring to exactly one resource node

Obligation - An operation specified in a rule, policy or policySet element that should be performed by the Obligation Handler in conjunction with the enforcement of an authorization decision

Policy - A set of rules, an identifier for the rule-combining algorithm and (optionally) a set of obligations. May be a component of a policy set

Policy Administration Point (PAP) - The system entity that creates a policy or policy set

Policy-combining algorithm - The procedure for combining the decision and obligations from multiple policies

Policy Decision Point (PDP) - The system entity that evaluates applicable policy and renders an authorization decision. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in [20]. This term corresponds to "Access Decision Function" (ADF) in [14].

Policy Enforcement Point (PEP) - The system entity that performs access control, by making decision requests and enforcing authorization decisions. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in [20]. This term corresponds to "Access Enforcement Function" (AEF) in [14].

Policy information point (PIP) - The system entity that acts as a source of attribute values

Policy set - A set of policies, other policy sets, a policy-combining algorithm and (optionally) a set of obligations. May be a component of another policy set

Predicate - A statement about attributes whose truth can be evaluated

Resource - Data, service or system component

Rule - A target, an effect, a condition and obligations. A component of a policy

Rule-combining algorithm - The procedure for combining decisions from multiple rules

Subject - An actor whose attributes may be referenced by a predicate

Target - The set of decision requests that a rule, policy or policy set is intended to evaluate.

4 Conventions

4.1 Abbreviated Terms

AD	Authorization decision
ADR	Authorization decision request
AIXM	Aeronautical Information Exchange Model
GeoPDP	PDP implementing GeoXACML
GeoXACML	Geospatial eXtensible Access Control Markup Language
GML	Geography Markup Language
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OWS	OGC Web Service
OWS-6/7/8	OGC Web Services Initiative, Phase 6/7/8
PAP	Policy Administration Point
PDP	Policy Decision Point implementing XACML
PEP	Policy Enforcement Point
SDI	Spatial Data Infrastructure
SOA	Service Oriented Architecture
URL	Uniform Resource Locator
URN	Uniform Resource Names
WFS(-T)	Web Feature Service (-Transactional)
XACML	eXtensible Access Control Markup Language

XML	eXtensible Markup Language

5 Introduction

WFS-T 2.0 instances serving AIXM information shall be official, recognized data sources that only publish reliable and accurate data. To meet this requirement appropriate access control systems need to be in place, that ensure that the update of existing AIXM features (by adding a time slice to the feature) and the insertion of new features meet various business rules.

Previous OWS initiatives focused on the authorized retrieval of AIXM information via WFS instances. The Aviation Thread of the OWS-8 initiative focuses on the secure update and insert of new AIXM 5.1 information into the underlying databases of WFS-T 2.0 instances.

In the following sections we identify a suitable rights model for an access control system protecting WFS-T 2.0 based AIXM data sources. We discuss how to define the required authorizations and how to implement and configure the components of the access control system enforcing these rights.

6 The Access Rights Model for the Authoritative AIXM Data Source

During the design and development phase of an access control system one has to agree on an appropriate conceptual and logical access rights model. The chosen models need to be sufficiently expressive to describe the types of access rights that need to be enforced in the given application domain.

Section 6.1 introduces popular rights models and summarizes their main characteristics. Section 6.2 lists types of access rights that frequently need to be enforced when protecting Web Feature Services and other OGC Web Services. Section 6.3 evaluates the presented rights models with respect to the required types of authorizations. Section 6.3.6 summarizes the results of this chapter.

6.1 Conceptual Access Rights Models

This section introduces the most popular conceptual rights models. In order to give a structured overview the rights model taxonomy shown in Figure 1 is used.

Figure 1: Taxonomy of rights models

6.1.1 SAR-based Rights Models

The abstract term **subject** is used to refer to entities like users, agents, services, processes etc. The characteristic property of subject entities is that they initialize interactions with the **resources** of the system. The resources of the system are e.g. members of classes like computer, service, file or feature and usually need to be protected. Each of these resource classes defines a number of operations and by calling these operations subjects can perform certain **actions** on the resources of the system.

The central characteristic of Subject-Action-Resource-based rights models (short: SAR-based rights models) is that rights are modeled by a ternary relation as shown in Figure 2. Hence from a conceptual perspective a right in a SAR-based model is a (subject-id_i, action-id_j, resource-id_k) tupel and describes an allowed (or a denied - in case of an open world assumption) action of a specific subject on a specific (abstract) resource.

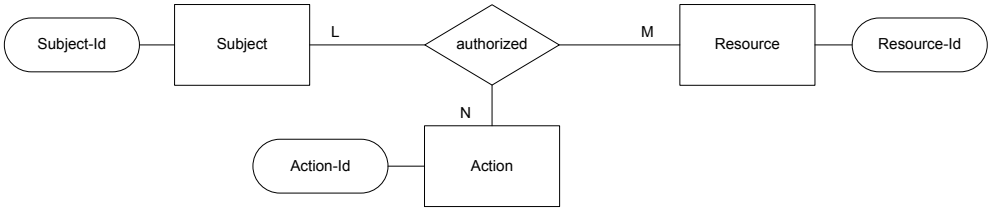

Figure 2: Conceptual design of a SAR-based rights model

During the evaluation of SAR-based rights the access control system has to determine the resources that are affected by the intended interaction. After the corresponding resource-id values have been identified, the access control system has to check if the interacting subject (represented by its subject-id) is allowed to perform the intended action (represented by an action-id value) on these resources. This checking is realized by searching for matching entries in the set of defined (subject-id, action-id, resource-id) tupels.

In case the administrators used abstractions of the existing subjects, actions and resources for their right definitions (e.g. subject-id = ”adult” or resource-id = “building-within-germany”), the access control systems additionally needs to verify if the subject, the action and the involved resources are members of one of the abstractions used in the right definitions.

Every conceptual SAR-based model can usually be mapped to different logical models. Popular logical SAR-based rights models are access control tables, access control lists and capability lists.

6.1.2 View-based Rights Models

One way how to extend SAR-based rights models is adding a new resource class called view (cp. Figure 3). A view is defined by a select query that specifies in its projection and selection clause a certain subset of the data stored in a database. By using a view-id as a

resource-id in a right tupel, one defines a right that refers to the set of resources as specified by the corresponding select query. The advantage of binding rights to views or "select queries" respectively is that one implicitly defines in one right tupel a set of rights that refers to all the resources in the view.

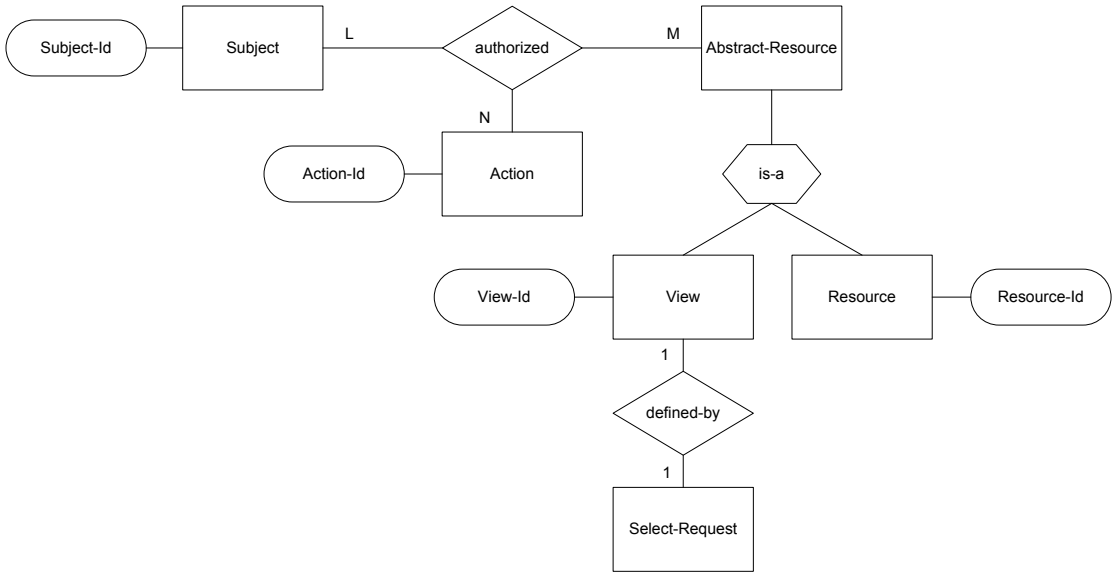

Figure 3: Conceptual design of a view-based rights model

6.1.3 Tagging-based Rights Models

Another class of conceptual rights models is the so-called tagging-based rights model class. Prominent representatives of this rights model class are e.g. the Bell-La-Padula Model [1][2], the Biba Model [3] and the Oracle Label-Security Model [16].

The central idea of models in this class is to label the resources with security-tags defining their level of sensitivity. Tag values could e.g. be 1 for "top secret", 2 for "secret", 3 for "confidential", 4 for "restricted" and 5 for "unclassified". Some tagging models additionally support the attachment of security-tags to schema elements. Adding a tag to a schema element implies that all instances of the schema element "inherit" the corresponding security-tag.

Next to the classification of the resources, subjects are also associated with security-tags. These tags represent the clearance levels of the subjects. In order to calculate authorization decisions, a tagging-based access control system has to compare the security-tag of the interacting subject with the tags of the affected resources by following a specific, system wide strategy. If e.g. the so-called "no-read-up" strategy is used, one only gets read access if the value of the security-tag of the resource is less than the value of the security-tag of the subject.

This short introduction of the central idea behind tagging-based rights models reveals that a right in such a model is defined in three steps: $assign_{subject\text{-}tag}$, $assign_{resource\text{-}tag}$ and $authorized_{strategy}$ (cp. Figure 4). Depending on the requirements of the use case these three right definition steps can be performed by different groups of administrators on different organizational levels.

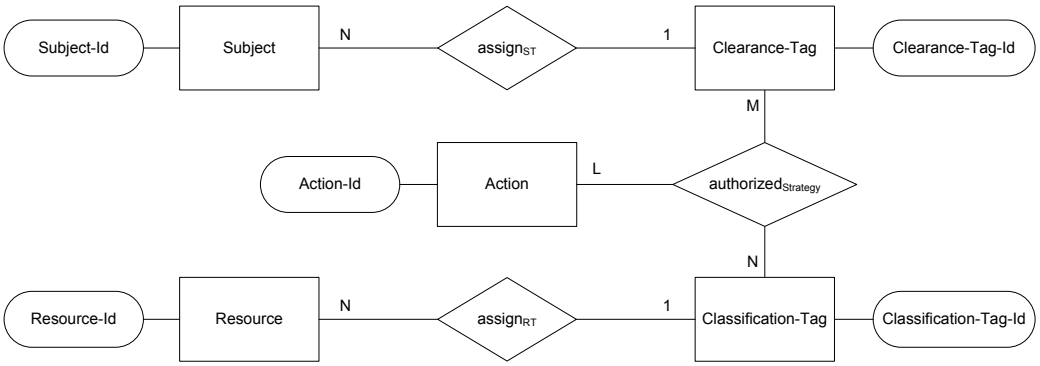

Figure 4: Conceptual design of a tagging-based rights model

6.1.4 Rule-based Rights Models

In a rule[1]-based rights model authorizations are defined through (Condition, Effects) tupels that are called access control rules. The condition part defines the applicability of a rule. If a rule's condition expression evaluates to 'true' in a given context, its effects are incorporated when calculating the authorization decision.

The access control process in an access control system using a rule-base rights model can be summarized as follows: Based on interaction attempt the access control system generates an authorization decision request (ADR) that defines the current evaluation context of the access control system. While evaluating the ADR the access control system has to determine, which of the access control rules are applicable in the given evaluation context. After identifying the applicable rules, the effects of these rules are combined and an authorization decision response is calculated.

This rough overview of the access control process in rule-based access control systems shows that every rule-based rights model also requires a corresponding evaluation context model. The more information is included in the evaluation context model, the more powerful access rules can be defined. Evaluation context models can be designed very application specific or very generic. Figure 5 shows a very generic evaluation context model.

[1] Some literature uses the synonym term "attribute-based rights model". We do not use this point-missing term as "attribute" is a synonym for "information" and it is meaningless to highlight that rights (as they always do) refer to some information entities.

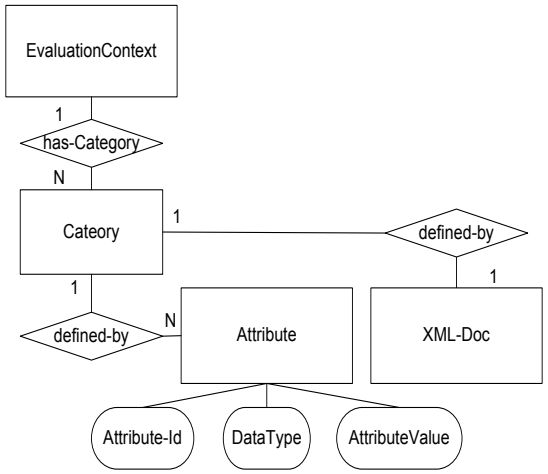

Figure 5: Conceptual design of a generic evaluation context model

Next to the design of a conceptual and logical evaluation context model a suitable rule-based rights model needs to be developed. Figure 6 shows an example of a conceptual rule-based rights model. The visualized model supports the definition of access control rules that must have an effect of permit or deny (called the sign of a rule). Additionally each rule can optionally have functional effects. These functional effects can e.g. cause the rewrite of an intercepted message (cp. 8.3.2) or imply the augmentation of the evaluation context by external data needed to calculate an authorization decision (cp. 8.3.3).

A condition expression of an access control rule is composed of literal-, pointer- and function-expressions. Pointers are used to refer to information items in the authorization decision requests. Rule-Container entities are “buckets“ that can hold any number of rule and rule-Container entities. Every rule-Container also has an assigned condition expression that defines the applicability of the container. Rule-Container entities can e.g. be used to structure the policy in order to enhance the performance of the access control process and for various other reasons.

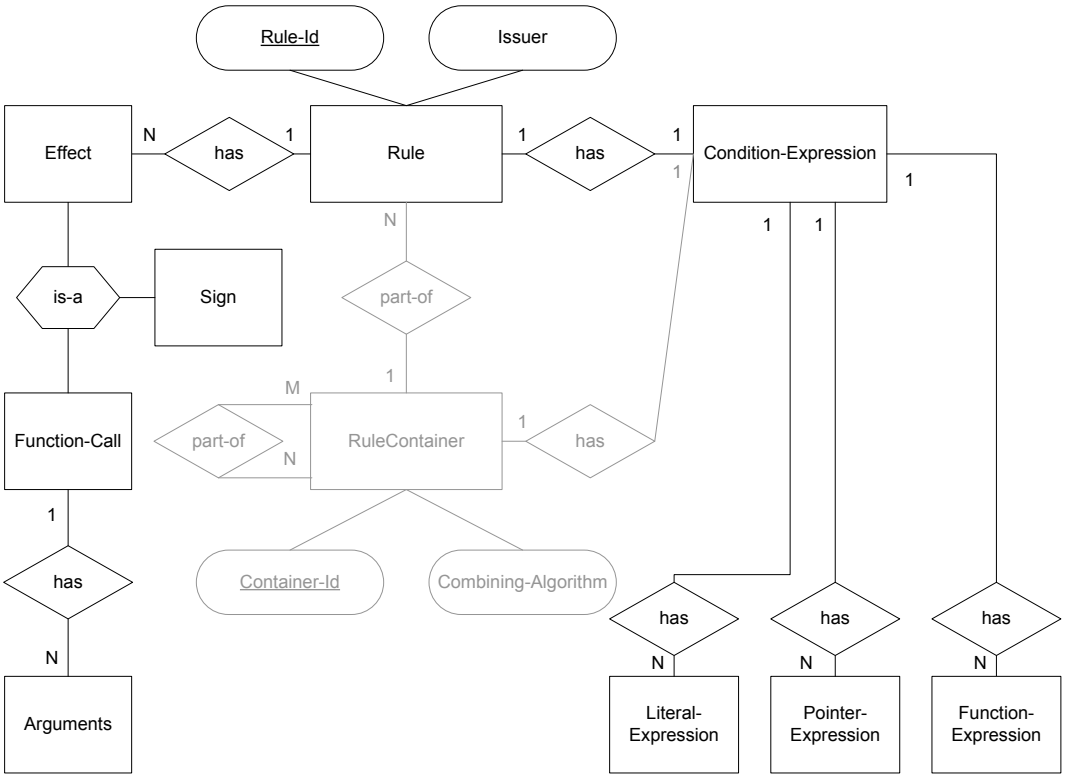

Figure 6: Conceptual design of access control rules and rule-Containers

After the development of a suitable conceptual evaluation context model and a corresponding conceptual rule model, one has to map this model to a suitable logical representation. A very expressive and popular logical evaluation context and rule model is e.g. defined in the eXtensible Access Control Markup Language (XACML) OASIS specification [6].

6.1.5 Role-based Rights Models

Role-based rights models introduce an indirection when defining authorizations. In a role-based rights model (see e.g. Figure 7) the privileges (i.e. the subject independent parts of the access rights) are not directly assigned to individual subjects. Instead privileges are assigned to roles that e.g. represent certain functional duties within an organization. Further roles are assigned to subjects that can activate a subset of their roles depending on the tasks they need to fulfill.

The role entity type and the thereby introduced indirection simplifies the administration of rights as it is more stable to bind privileges to a role representing a certain job position, compared to assigning privileges to an advancing subject directly. Additionally the concept of role hierarchy, through which the inheritance of privileges is described, helps simplifying the administration of large access control policies.

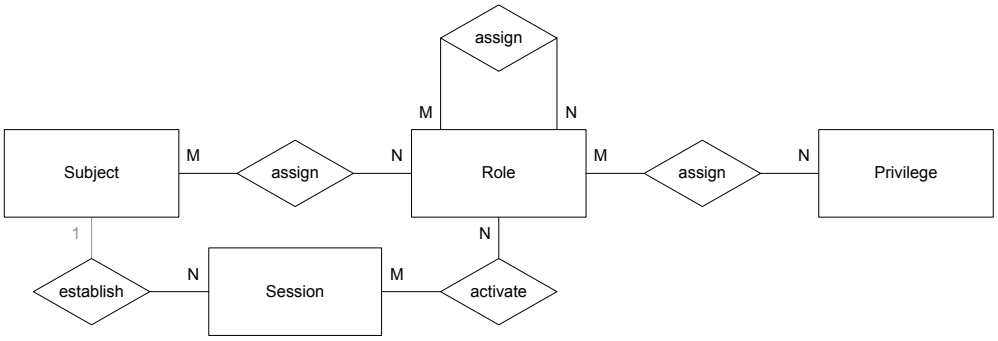

Figure 7: Conceptual design of the $RBAC_1$ model [13]

6.2 Required types of authorizations

The protection of WFS instances and their underlying data bases requires the enforcement of rights with different characteristics. The following subsections introduce various types of authorization semantics that frequently need to be enforced when securing WFS instances. The presented right specific requirements were indentified in various expert interviews, working group sessions, consulting projects and in the course of an in-depth secondary literature research.

6.2.1 Rights referring to individual resources

There is the need to define access rights that control the possible interactions with the individual resources of spatial data infrastructures. These resources belong to different resource classes that are shown in Figure 8.

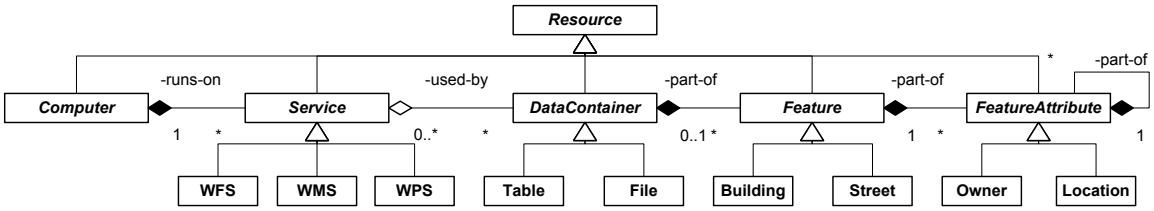

Figure 8: Classification of resources in spatial data infrastructures

The support of rights that refer to resources of different classes is very important as it simplifies the administration of rights and at the same time supports an easy implementation of the least privilege principle.

Only supporting rights that can refer to coarse-grained resources like computers or services would imply the risk that access to more fine-grained resources (e.g. certain building features) is unnecessarily restrictive or too permissive.

On the other side the binding of rights to naturally existing, coarse-grained abstractions of information entities is very helpful, as it allows expressing a huge set of rights in one single right. A right that e.g. declares that a user is allowed to have read access on building "xyz", stands for a set of rights that permit the user to read all attributes that are associated with that building.

6.2.2 Rights referring to service, feature and attribute classes

Next to the definition of rights that refer to resource instances of certain classes, it is required to support the definition of rights that refer to individual resource classes. A class-based right represents an authorization that refers to all existing and future instances of this class. Rights that refer to resource classes are e.g. "Alice is denied to use services of type WFS" and "Bob is permitted to have read access on features of class Street". Class-based rights simplify the administration of the access control policy and allow to directly express frequently intended authorization semantics.

6.2.3 Rights referring to resources with certain properties

The SDI use case usually implies that access to millions of resources need to be controlled. Due to scalability problems the assignment of rights per resource is therefore not a suitable approach. Further a translation of the authorization semantics that are expressed in terms of conditions on features into rights per resource implies additional problems. Translating a right that e.g. states "permit if the buildings price is less than one million" into rights per building, by using the building-ids of buildings with a current price less than one million is in most cases not suitable. The states of the resources usually change frequently which implies that the set of rights, now referring to individual resources, would have to be updated constantly - which obviously causes an unacceptable administrative overhead.

To address these administrative problems it is necessary to support the definition of rights that natively refer to resources with certain properties. Right definitions must therefore contain condition expressions that express certain constraints on the properties of resources they are intended to refer to. Rights that refer to resources with certain properties are e.g. "Alice is allowed to read data of building features if each of the buildings costs less than one million US$" and "Bob is denied access to building data if he is not the owner of the building".

One special requirement in the geospatial problem domain is that spatial conditions over geometric properties of features need to be expressible. It is e.g. frequently required to express rights like: "if buildings are within a certain area than permit access to their data". Table 1 lists various spatial functions that are needed to define spatial rights.

Topological Functions	Constr. Geometric Functions	Miscellaneous Functions
Equals	Buffer	Distance
Disjoint	Boundary	IsWithinDistance
Touches	Union	Length
Crosses	Intersection	Area

Topological Functions	Constr. Geometric Functions	Miscellaneous Functions
Within	Difference	
Contains	SymDifference	
Overlaps	Centroid	
Intersects	ConvexHull	

Table 1: Functions for the definition of spatial rights

6.2.4 Rights refering to subjects with certain properties

As it is the case for resources, it is required to bind rights to subjects with certain properties. One could e.g. need to define rights that refer to subjects that have a specific citizenship, are over the age of 21 or are currently within the US. The last example points out that defining condition expressions over the subject attributes also requires the support of spatial functions as listed in Table 1.

6.2.5 Rights refering to actions with certain properties

Controlling access to services implies that rights must be enforced that refer to any operation that can be called by the subjects (e.g. insert, read, update and delete operations). If a services groups operations into classes (e.g. the transaction class of a WFS), it is helpful to support rights referring to these action classes.

6.2.6 Rights refering to environment states with certain properties

The state of the environment of an access control system can e.g. be described by attributes like “current-time”, “access-history” or by complex application specific state-documents that e.g. describe the current natural disaster state, system load, etc.

In various scenarios it is required to define rights that are dependant on the state of the environment. In the OWS-6 project it was e.g. required to define rights that allow access to building feature data in a disaster area if the firemen are within a certain distance of a disaster location. This example case not only shows the need for rights that refer to environment states with certain properties, but also highlights that spatial functions are needed to express condition expressions referring to spatial environment state variables (cp. Table 1).

6.2.7 Rights refering to arguments of service requests with certain properties

A subject usually has to pass various arguments when calling an operation of a Web Service. The invocation of the WFS update method e.g. requires that a projection and selection clause is specified that define the part of the features’ data that needs to be updated. Further new feature attributes or a whole new feature has to be passed as an argument in the update request and will replace the specified subset of the features’ data.

Diverse security requirements, commercial interests and the enforcement of integrity constraints require the support of rights that refer to the arguments of service requests. It can e.g. be necessary to ensure that a subject working for the land survey office of region A can only insert building data to a WFS feature store if the new building features are within area A.

6.2.8 Support of positive and negative rights

The definition of an access control policy can be simplified if the used rights model supports the definition of positive (i.e. access permitting) and negative (access denying) rights. By combining positive and negative rights it is easy to implement exceptional rules. One could e.g. define a positive right that allows subjects over 21 to use a certain service and additionally one could specify a negative right that denies Alice (who is over 21) to use this service.

Without the support of rights with different signs and the introduction of intended runtime conflicts, existing rights might need to be changed in order to implement exceptional rules. While this is a valid and recommended approach it might not always be the first choice, as it can imply complex administrative tasks. Further this approach requires that the existing rights can be freely updated. This is often not the case as certain parts of the policy can be included by reference or have been defined by administrators of other administrative domains and are therefore marked read only.

6.3 Evaluation of rights models in the OWS use case

In the following subsections we analyze which of the rights models introduced in section 6.1 allow the definition of all types of rights required in the OWS use case (cp. 6.2).

6.3.1 Suiteability of SAR-based models

SAR-based models support the definition of rights that refer to individual resources of different classes. However they do not support the definition of rights that refer to subjects, actions and resources with specific properties. Rights are tuples that refer to the ids of subjects, actions and resources and there is no mean to define conditions on the attributes of these entities. Another disadvantage is the fact that SAR-based models do not support the definition of rights that refer to environment states and to arguments of service requests with certain properties.

6.3.2 Suiteability of view-based models

Opposed to SAR-based models, View-based models support the definition of rights that refer to fine-grained resources with certain properties. However they still suffer from the other conceptual weaknesses of SAR-based models. It is not possible to express rights that refer to subjects, actions, environment states and request arguments with certain properties. Another disadvantage is that the realization of the view concept is often only possible within a data base management system as it would otherwise imply very high view maintenance costs. This conflicts with the principle that security should be achieved through security services that are loosely coupled with the application (cp. 7.1).

6.3.3 Suiteability of tagging-based models

In order to use a tagging-based model it must be possible to tag the resources with security markup. This assumption is problematic in the "access control for OGC Web Services" use case. If the service that needs to be protected belongs to another security domain, one cannot add security tags to the resources served by this service. Further resources that are inserted or generated in real time (e.g. by an OGC Sensor Observation Service) do not have assigned security tags yet. Further tagging-based models do not support the definition of rights that refer to request arguments and environment states with certain properties. Another disadvantage is the fact, that all resources have to be tagged individually which does not scale very well and causes an unacceptable administrative overhead when e.g. protecting WFS instances serving millions of features.

Note that an automated tagging mechanism based on rules (i.e. the translation of right conditions to resource-ids) is in general not an applicable solution as the automatically generated tags would have to be recalculated after every update of the resource, subject or environment state attributes.

6.3.4 Suiteability of rule-based models

The expressiveness of rule-based rights models is dependant on the used evaluation context model. We start by analyzing how evaluation context models have to be designed in order to be useful in the OGC Web Services use case. Afterwards we focus on the types of conditions and effects that must supported in order to be able to define the needed types of rules.

6.3.4.1 Evaluation models in the OWS use case

In the OWS use case the intercepted request or response and data referring to the subject, the service and the environment state can be used to initialize an evaluation context. To support the required types of rights (cp. 6.2) it is necessary to include information of all these entity types in the evaluation context model. Hence suitable evaluation context models must be specializations of the abstract model shown in Figure 9. Models of this class are called Subject-Service-Message-Environment evaluation context models (short: SSME evaluation context models).

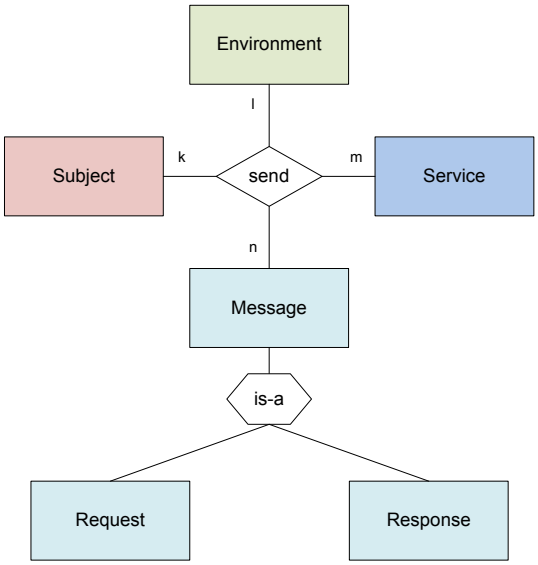

Figure 9: The abstract SSME evaluation context model

SSME evaluation context models can be instantiated based on an intercepted OWS request or response. In most cases evaluation contexts have to be generated based on the intercepted OWS requests. In cases where the request or response based approach could be used, it is always favorable to use the first, as this avoids unnecessary network load and processing steps in the services. Note that the initialization of response based evaluation contexts is avoidable in most cases and is only needed to implement very special types of authorization semantics.

6.3.4.2 Expressivness of access control rules refering to SSME model instances

In the following it will be analyzed if the required types of rights can be expressed by access control rules that refer SSME model conformant evaluation contexts.

Support of positive and negative rights

Access control rules can have a permit or deny effect and hence support the definition of positive and negative rights. Exceptional rules can easily be implemented by defining access control rules with different effects and by additionally specifying an appropriate conflict resolution algorithm for the corresponding rule-Container.

Support of spatial rights

If spatial functions can be used in condition expressions of rules, one can define the required spatial rights.

Support of rights referring to subjects, computers, services, actions and environment states with certain properties

If the pointers of rule condition expressions can select data in the subject, service, action and environment categories of evaluation contexts, one can define rights referring to subjects, computers, services, actions and environment states with certain properties.

Support of rights referring to request arguments with certain properties

Access control rules referring to SSME evaluation contexts can express rights that refer to the arguments of service requests, as this information is directly included in the authorization decision requests under the &message; category.

Support of rights referring to feature- and attribute-classes

If the OWS requests have projection clauses, one can define condition expressions that point to the information contained in these clauses. Hence access control rules can refer to the feature and attribute classes from which a subject e.g. wants to retrieve, delete or update data.

Support of rights referring to features and attributes with certain properties

The definition of rights that refer to features and attributes with certain properties is not always achievable by defining condition expressions and permit/deny effects.

Requiring certain characteristics of the selection clauses of intercepted requests is not appropriate as the subjects need to have as much flexibility as possible when selecting the intended features and attribute sets.

Defining condition expressions that refer to features and attributes contained in the service responses is also not a suitable solution in many use cases. This approach is e.g. not applicable in case of insert, delete and update interactions. Even in case of read request it is has severe limitations, as the responses might not contain the data the rule conditions refer to. If there is e.g. a rule that denies read actions on building data within area A that is served by a WFS, this rule would not be evaluable based on responses, where the subjects did not select the buildings' location attributes.

To solve this expressiveness problem one needs to have the capability to define two additional types of rule effects:

Rewrite Effects
By defining rule effects that cause a rewrite of the selection and projection clauses of intercepted requests, one defines rights referring to features and attributes with certain properties. By e.g. adding a predicate like within(Building.Geometry, $Polygon_{area\_A}$) per conjunction to the original selection predicate of an intercepted WFS GetFeature request that refers to the building feature class, one restricts the intended interaction scope of the subject to buildings that are within area A.

Next to the extension of the selection predicates by static or dynamically calculated predicates one can also delete or update certain parts of an intercepted request and

thereby enforce various types of authorizations. E.g. the deletion of attribute names in the projection clauses of read requests has the effect that the responses will not contain the corresponding attributes (assuming they are optional according to the schema of this feature type). Another example is e.g. the update of the FeatureVersion attribute of GetFeature requests or the update of the feature that is to be inserted by a transaction/insert request. The first rewrite will control which feature versions can be selected and the second will ensure some integrity constraints.

PIP-control Effects
There might be situations where authorization decisions can only be computed based on information that was not available in the originally created evaluation context. Assume e.g. that a subject-id is know, but its age is not. In order to support the definition of a rule that controls access dependant on the subjects' age one needs a mechanism to add the required information to the evaluation context. Defining rules with special functional effects (called PIP-control Effects) that tell some entity from where to get the additionally needed information and how to insert this data in the evaluation context is a suitable concept how to enhance the expressiveness of the rule based model in this direction.

PIP-control Effects cannot only be associated to subjects but also to action, resource and environment.

6.3.5 Suiteability of role-based models

The use of role-based rights models introduces various advantages like e.g. the enhancement of the stability of rights, the capability to selectively activate rights and the support of rights inheritance relationships. In the OWS use case it is especially important to benefit from these advantages and hence the rights models of access control systems for these architectures should also incorporate the concepts of role-based rights models.

6.3.6 Conclusion

The analyses of the previous subsections have shown that neither SAR-, view- nor tagging-based rights models are suitable to describe the rights needed in the OWS use case. Instead a rights model that combines the ideas of rule- and role-based models is the appropriate model under the given requirements. It is important, that the resulting hybrid model not only supports the definition of rules with permit or deny effects but also the declaration of functional effects through which one can achieve the rewriting of intercepted OWS message and the extension of evaluation contexts. To support the required expressiveness one further needs to develop an evaluation context model that is an appropriate specialisation of the SSME model.

Based on the XACML and GeoXACML specification [6][9] and the XACML RBAC Profile [21], the XACML Multiple Decision Profile [23] and the XACML Hierarchical

Resource Profile [22] one can define such an expressive hybrid rights model and a corresponding evaluation context model.

The core of this set of relevant specifications is the XACML specification which defines two mutually dependent XML languages. One language is used to define access control rules and rule-Container elements and the other is used to define authorization decision requests and responses.

Although XACML offers a huge set of functions and data types that can be used to define policy elements with complex condition expressions, it does not support the definition of spatial rights. This conflicts with the requirement for these types of rights in the OWS use case.

To solve this limitation a spatial extension of XACML, called GeoXACML [8], was defined. GeoXACML has been standardized by the OGC and adds the capability to express spatial authorization semantics by supporting attributes of a geometric data types and the spatial functions listed in Table 1 in condition parts of XACML <Rule>, <Policy> and <PolicySet> elements.

The requirements defined in the XACML v3.0 RBAC Profile specify how to use the language constructs provided by the XACML and GeoXACML specification (short: (Geo)-XACML specification) in order to implement the $RBAC_0$ and $RBAC_1$ model of the NIST (see [13]). The use this profile on top of the (Geo)XACML specification establishes the required coupling of a rule and role-based rights model.

In the OWS use case the messages exchanged between the subjects and the services can be XML encoded and can therefore represent a set of hierarchically structured information entities. This circumstance implies that the XACML v3.0 Multiple Decision Profile and the XACML v3.0 Hierarchical Resource Profile have to be used. Next two others these two profiles define how to express complex XACML encoded rights referring to information entities in XML documents.

The mentioned OASIS and OGC specifications are designed to be usable in many different use cases. For a specific application domain, like e.g. the OWS use case, one can define additional guidelines that enhance interoperability within and between distributed and collaborative (Geo)XACML based access control system components. These guidelines further facilitate the development and configuration of appropriate access control solutions. Such a set of guidelines has been defined in the XACML v3.0 OGC Web Services Profile [25] and its service specific extensions (e.g. [26]). The guidelines of the XACML OGC Web Services Profile describe how an XACML Context Handler shall generate XACML authorization decision requests from OWS messages exchanged in OGC Web Service based architectures and from other available information. They further describe how the Context Handler shall process XACML obligations defining an access control specific rewriting of OWS messages and an extension of XACML evaluation contexts.

7 Architecture and Information Flow

At the beginning of this section a suitable security architecture design pattern is presented (cp. 7.1). Afterwards the general architecture of rule-based access control systems (cp. 7.2) and the information flow within these systems (cp. 7.3) is introduced.

7.1 General Security Architecture

Separating security aspects as much as possible from the implementation of an OGC Web Service allows securing existing OWS instances without security related code changes. This separation of concerns further enables leveraging available IT-security concepts and implementations.

When externalizing security functionalities it is advantageous to provide the security capabilities through separate security services (e.g. authentication, authorization and audit services). Security services can be flexibly combined and can be used in different configurations for several geo-processing services (details see [18]). Each of these security services can itself be composed of further services.

Advantages of a modular security architecture approach are e.g.:

- Splitting the security solution into separated functional components reduces the associated development and maintenance complexity.
- The solution is fully scalable and easy to upgrade. New security services can be easily inserted and existing services can be upgraded without affecting the others.

Because of the mentioned advantages we use a service oriented security architecture. The focus of this report is the access control service only. A question that needs to be addressed is where to initialize the access control process in the overall architecture. Figure 10 shows components (see ACS[2] boxes) in which the access control process could be initialized.

No assumptions on the client side software configuration

In OWS based architectures one cannot assume that subjects interact with OWS instances though client programs with specific built-in security functionalities. It can e.g. be the case that subjects interact with services through ordinary web browsers. The consequence of this situation is that the access control process cannot be initialized and enforced in components labeled 1 and 2.

[2] ACS - abbr. for Access Control System.

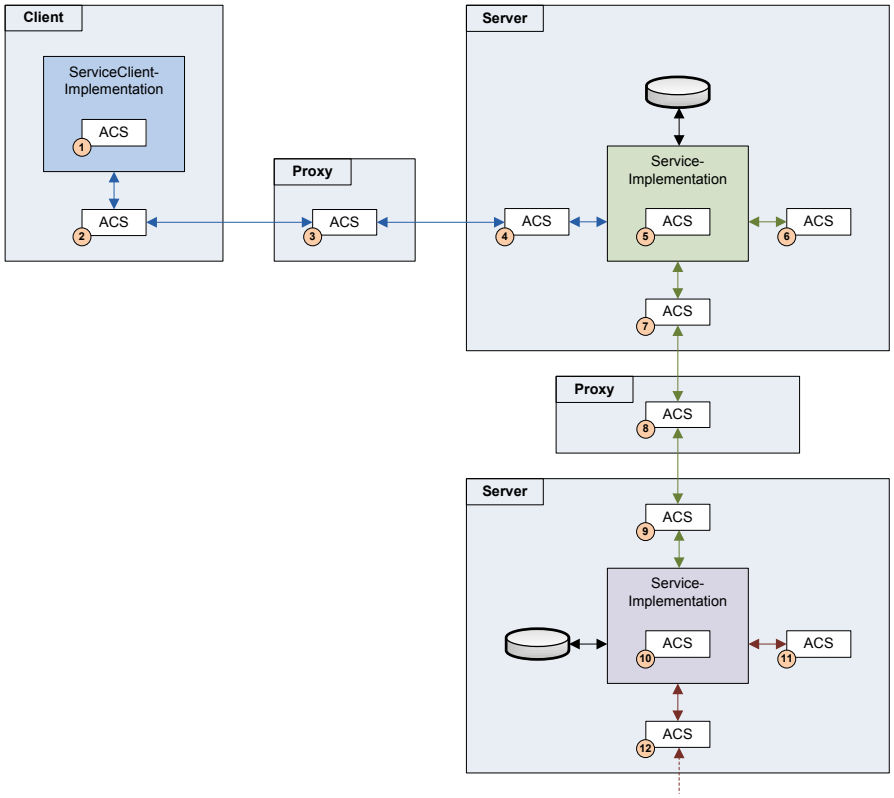

Figure 10: Candidate components for the initialization of the access control process

Access rights cannot be controlled "behind" services

Enforcing access rights in the components 6 to 12 implies that the access control process operates on the sub-requests and/or the corresponding responses. This is problematic in cases where the required authorization semantics can only be enforced based on the messages exchanged between the interacting subject and the service (e.g. GetCapabilities or wps:execute requests). Next to this problem the post-service access control approach is not realizable if the components 8 to 12 belong to other, independent administrative domains.

Independency of enforceable rights of the used service implementations

OWS implementations used in SOAs are usually proprietary, from different vendors and have no or variably powerful built-in access control capabilities. It cannot be assumed that all service implementations provide sufficiently expressive access control functionalities. Hence the access control process cannot be realized in the components 5 and 10.

The requirements listed above clearly reduce the number of suitable components and imply that the access control process can only be enforced in the components 3 or 4. This

implies that an appropriate rights model must support the definition of rights that (next to others) refer to the intercepted messages. Whether component 3 (i.e. a dedicated proxy server) or component 4 (i.e. a server-side proxy component) is more suitable is dependent on the characteristics and requirements of the given use case. Component 4 could e.g. be in favor as it allows local calls of the access control system and thus implies certain performance advantages. In contrast, the initialization of the access control process in component 3 can result in scalability and availability advantages.

7.2 Architecture of XACML based Access Control Systems

Figure 11 shows the architecture of the proposed rule- and role-based access control system and give a rough impression of the internal information flow. The access control system serves as a proxy component that intercepts messages exchanged between subjects and WFS instances.

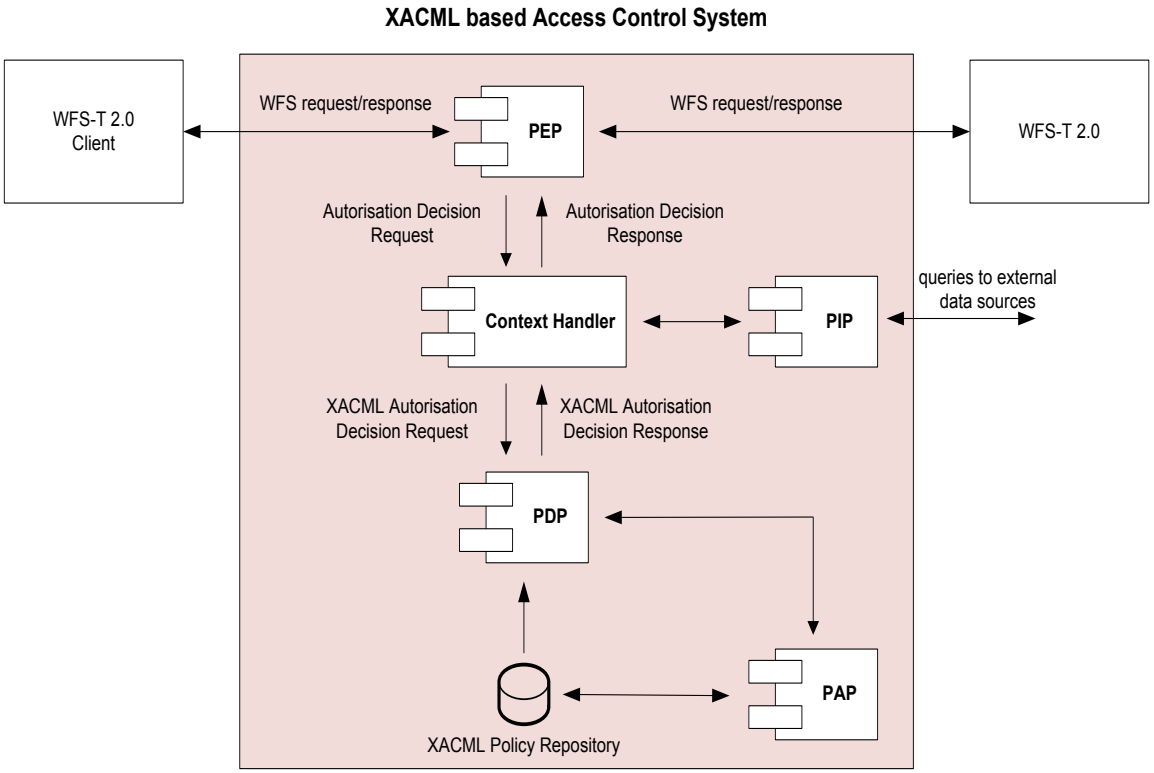

Figure 11: Architecture of an XACML based Access Control System

The **Policy Enforcement Point (PEP)** is the entry point in the access control process. Based on an intercepted WFS request or response, the PEP generates an authorization decision request in an implementation specific (or already XACML compliant) format and sends it to the **Context Handler**.

When receiving an authorization decision request from a PEP the Context Handler generates an XACML authorization decision request (XACML ADR) based on the

information already included in the received request and - if required - based on additional information that can be queried from external information sources through the **Policy Information Point (PIP**).

The Context Handler forwards the generated XACML ADR to the **Policy Decision Point (PDP)**. The PDP evaluates the incoming ADR by searching for applicable rules defined in the currently loaded XACML policy. The effects of all rules that evaluate to 'true' or 'indeterminate' under the given ADR are combined and an XACML authorization decision response is returned to the Context Handler.

The Context Handler interprets the result returned by the PDP and acts correspondingly (Details see 7.3). In the end the Context Handler translates the XACML encoded authorization decision response back into the application specific authorization decision request/response language (if this is not XACML) and will then forward the response to the PEP. The PEP will in turn act according to the result of the access control process.

The **Policy Administration Point (PAP)** is the component that allows policy administrators to retrieve, insert, update, delete, test and analyse XACML encoded access rights. Additionally the PAP can be used by the PDPs to query relevant parts of XACML policies.

It is important to highlight, that the presented architecture of rule-based access control systems is very flexible. Each of the introduced components can be replicated and distributed as required. In addition, certain components can be aggregated into one component. . For example, one can implement a PEP that consolidates the PEP and Context Handler functionality.

7.3 Information Flow Classes in XACML based Access Control Systems

Depending on the characteristics of an XACML ADR and the state of the XACML policy, a specific XACML authorization decision response is returned by the PDP to the Context Handler. Each XACML authorization decision response can be assigned to exactly one of the response classes listed below:

a) permit response without rewrite-obligations

b) deny response without rewrite-obligations

c) permit response with rewrite-obligations

d) deny response with rewrite-obligations

e) indeterminate response with missing-attribute information and/or PIP-control-obligations

The information flow in an XACML based access control system is dependant on the type of XACML authorization decision response retuned by the PDP. The following subsections illustrate the different information flows resulting from the different types of XACML authorization decision responses.

Note that the examples in the subsections below describe access control processes that are triggered after intercepting WFS requests. It is straight forward to adjust these examples to scenarios where it is unavoidable to perform the access control process on intercepted WFS responses. The characteristics of the different information flow variations remain unaffected whether you perform request- or response-based access control. Further it should be noted that the PEP components shown in the diagrams incorporate the Context Handler functionality. This simplifying assumption keeps the diagrams simple and helps focusing the relevant aspects.

7.3.1 Permit XACML authorization decision response without rewrite-obligations

Figure 12 shows the consequences on the information flow in cases where the Context Handler receives a consolidated **permit** XACML authorization decision response without any rewrite-obligations. According to the granting effect of the XACML authorization decision response, the PEP forwards the intercepted, authorized WFS request to the WFS.

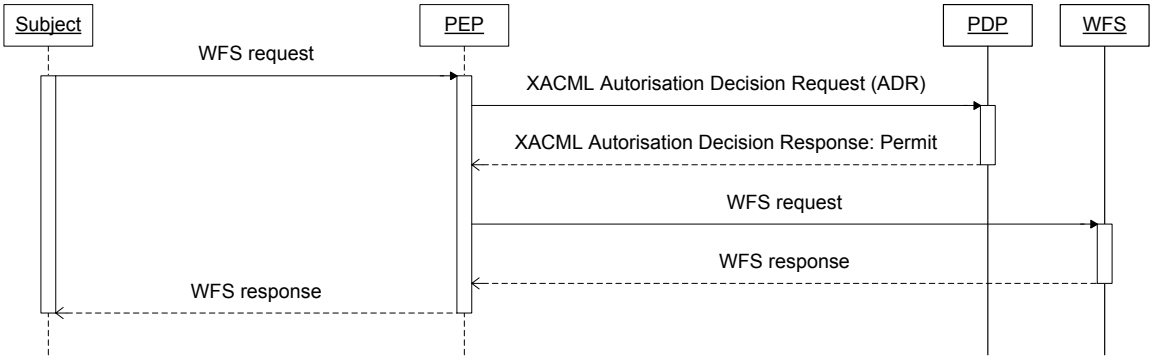

Figure 12: Information flow in case of a permit XACML authorization decision response without rewrite-obligations

7.3.2 Deny XACML authorization decision response without rewrite-obligations

Figure 13 illustrates the consequences on the information flow in cases where the Context Handler receives a **deny** XACML authorization decision response without any rewrite-obligations. Based on such an authorization decision response the PEP has to create and submit an OWS exception report that will inform the requesting subject about the denial of his intended interaction with the service.

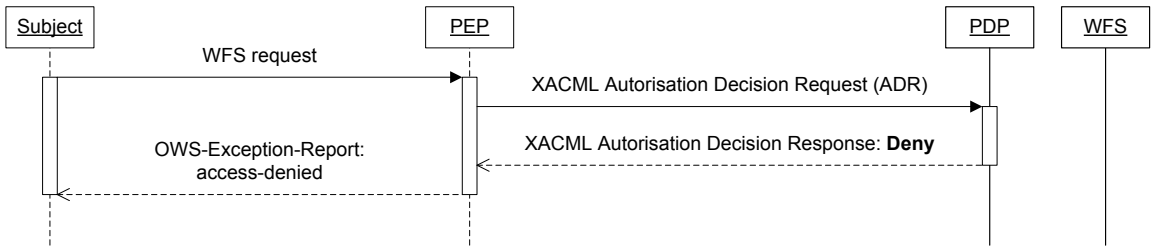

Figure 13: Information flow in case of a deny XACML authorization decision response without rewrite-obligations

7.3.3 Permit XACML authorization decision response with rewrite-obligations

Figure 14 shows the consequences on the information flow in cases where the Context Handler receives a **permit** XACML authorization decision response with rewrite-obligations. Opposed to the former two cases the rewrite-obligations contained in the XACML authorization decision response force the Context Handler (included in the PEP in the example) to call an Obligation Handler. The Obligation Handler processes the transformation instructions defined in the rewrite-obligations. The transformation instructions imply a rewrite of the representation of the intercepted WFS message in the ADR. After returning the modified XACML ADR to the PEP, the ADR specific representation of the intercepted and now modified WFS message is mapped back to its original encoding (either XML or KVP) and than forwarded to the WFS. It is important to point out that the permit decision causes the PEP to forward the rewritten WFS request to the service without informing the subject about the rewrite. This approach is therefore called the opaque rewriting approach

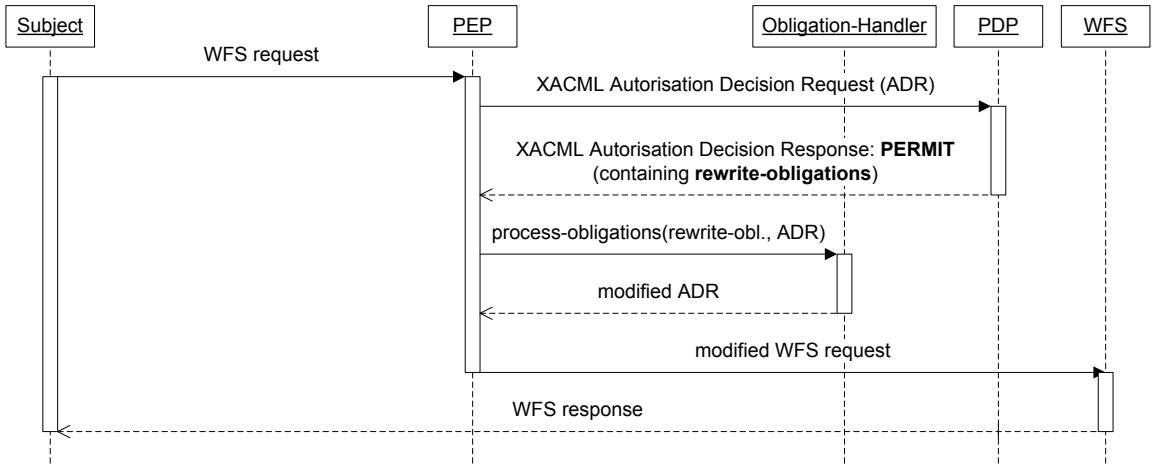

Figure 14: Information flow in case of a permit XACML authorization decision response with rewrite-obligations

7.3.4 Deny XACML authorization decision response with rewrite-obligations

Figure 15 illustrates the information flow in cases where the Context Handler receives a **deny** XACML authorization decision response with rewrite-obligations. Compared to the opaque rewriting approach introduced in the last subsection, the rewritten request is not directly forwarded to the service. Instead the subject is informed about the denial of its intended interaction through an OWS exception report. This report contains, next to the deny information, the rewritten version of the originally submitted WFS request. The subject can now choose to use the rewritten request instead of its former request, to cancel the intended interaction or can decide to define a new request from scratch. The advantage of using the rewritten request is that it represents the intersection of its once intended, but not fully permitted request and the set of authorized interactions of this subject. Hence it represents a proposal, automatically generated by the access control system that is as close as possible to the original intension and still compliant with the access control policy in place

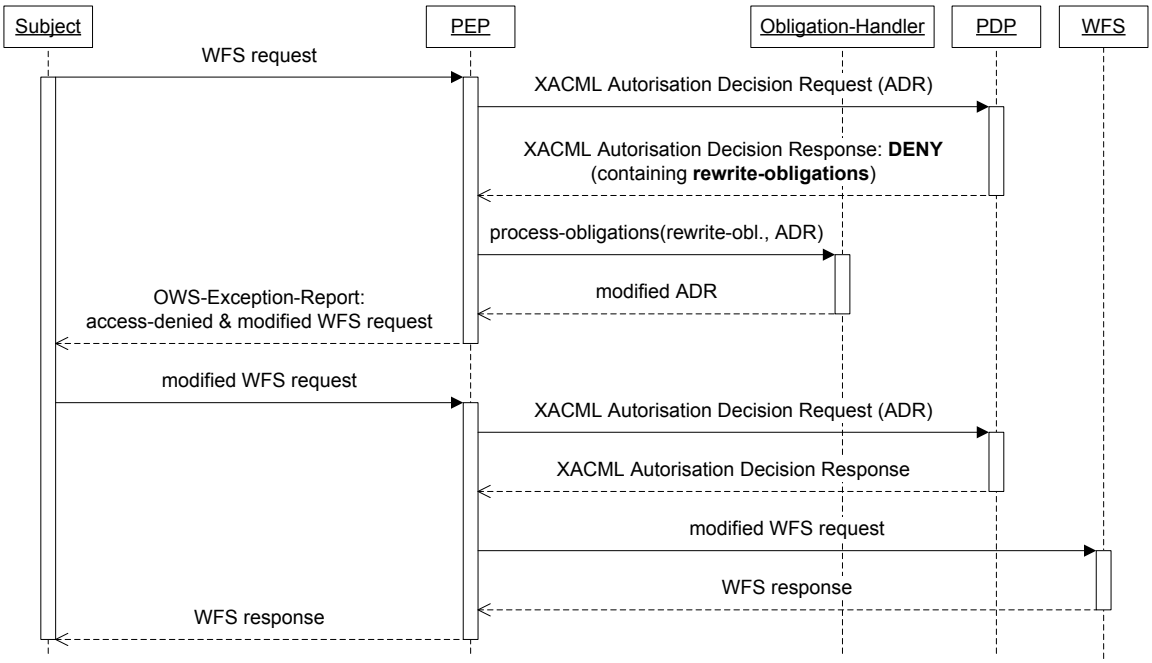

Figure 15: Information flow in case of a deny XACML authorization decision response with rewrite-obligations

Figure 15 visualizes the situation where the subject uses the proposed rewritten request and submits it to the service. For the second access control phase one can e.g. define a special subtree in the policy that checks whether the intercepted request was digitally signed by the access control system (i.e. was generated by the access control system in a previous access control process) and whether the time stamp associated with the signature is within a certain range. If this is the case the access control system knows without further evaluation, that the intercepted request is authorised.

The insertion of entries into the repository containing the digital signatures and the assignment of their validity period can be either realised by adding appropriate PIP-control-obligations to a deny rule element containing rewrite-obligations or implicitly through an appropriately implemented Obligation Handler.

The augmentation of the ADRs by this information can happen by default, through the Context Handler or can be controlled by the policy through the XACML missing-attribute mechanism or through PIP-control-obligations.

7.3.5 Indeterminate response with missing-attribute information and/or PIP-control-obligations

Figure 16 shows the consequences on the information flow in cases where the Context Handler receives an **indeterminate** response with missing-attribute information and/or PIP-control-obligations. A Context Handler receiving an XACML authorization decision response with a <Decision> element contents of "Indeterminate" with a status code of urn:oasis:names:tc:xacml:1.0:status:missing-attribute, will try to retrieve the missing information. There are two options how this can be done.

In case of missing XACML attributes the Context Handler implementation can use built-in methods to resolve the values of the missing XACML attributes. The <StatusDetail> element lists the names and data-types of any attributes that are needed by the PDP to refine its decision. To support this approach in a specific application domain one has to clearly define the expected behaviour of the Context Handler, in case one of the used XACML attributes turns out to be missing in the ADR.

A more generic solution that is also available if information under <Content> elements is missing (i.e. in case of indeterminate responses that result from <AttributeSelector> elements that cannot be evaluated), can be realized by using PIP-control-obligations. These obligations contain instructions that tell the PIP from where to retrieve more data, how the corresponding PIP query should look like and how the resulting response should be included in the original XACML ADR.

After extending the original ADR the Context Handler can resubmit the extended ADR. Now the PDP has all the information that caused the missing-attribute indeterminate response in the first run and can finally calculate the requested authorization decision.

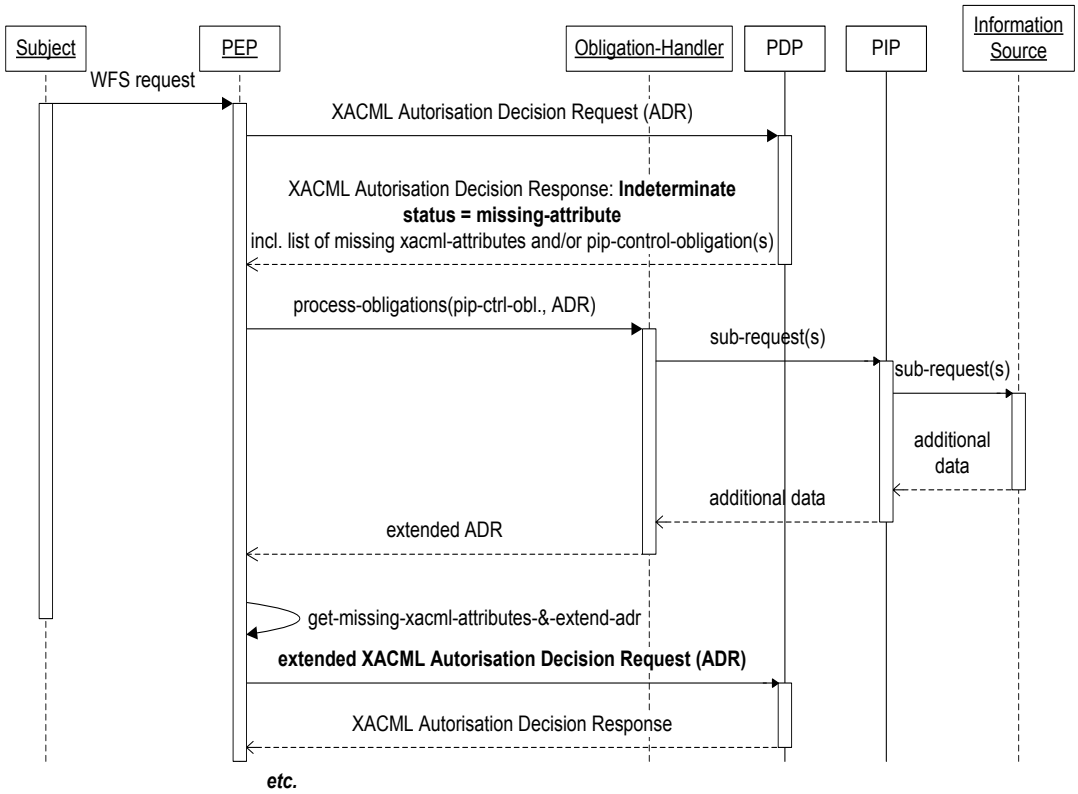

Figure 16: Information flow in case of an indeterminate response with missing-attribute information and/or PIP-control-obligations

8 Techniques to implement the required types of rights in (Geo)XACML

This section explains how to generate adequate XACML ADRs based on intercepted WFS messages and how to implement the required types of the rights. For each type of rights we present an XACML code fragment that demonstrates how to express authorization semantics of that kind.

All examples given in this section are not AIXM specific and intend to explain the concepts only. The application of these concepts to protect WFS instances that process AIXM data will be shown in the upcoming section 9.

Note that the interested reader is recommended to have detailed knowledge on the language constructs provided by the XACML v3.0 specification, the GeoXACML specification and the related profiles (cp. 6.3.6).

8.1 XACML based implementation of the SSME evaluation context model

Section 6.3.4.1 has shown that evaluation context models must be specializations of the abstract SSME evaluation context model shown in Figure 9, to support the types of rights required in the OWS use case. The sample XACML ADR presented under Listing 1 demonstrates how a SSME model conformant XACML evaluation context can look like. The visualized ADR describes the following situation: A user with activated &citizen; role, german citizenship and a current location within germany wants to interact with a specific WFS running on a server with certain hardware and software characteristics.

```
<Request ...>
  <Attributes Category="&access-subject;">
    <Attribute AttributeId="&role;" IncludeInResult="false">
      <AttributeValue DataType="&string;">&citizen;</AttributeValue>
    </Attribute>
    <Attribute AttributeId="&citizenship;" IncludeInResult="false">
      <AttributeValue DataType="&string;">german</AttributeValue>
    </Attribute>
    <Attribute AttributeId="&current-location;" IncludeInResult="false">
      <AttributeValue DataType="&geometry;"><gml:Point ...>
        ...<!-- a place in Munich --></gml:Point></AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="&recipient-subject;">
    <Attribute AttributeId="&ip-adress;" IncludeInResult="false">
      <AttributeValue DataType="&string;">123.123.123.123</AttributeValue>
    </Attribute>
    <Attribute AttributeId="&os-recipient-host;" IncludeInResult="false">
      <AttributeValue DataType="&string;">windows</AttributeValue>
    </Attribute>
    <Attribute AttributeId="&mem-recipient-host;" IncludeInResult="false">
      <AttributeValue DataType="&integer;">1.000.000.000</AttributeValue>
    </Attribute>
```

```
    <Attribute AttributeId="&service-url;" IncludeInResult="false">
      <AttributeValue
DataType="&string;">http://domainA.com/wfs</AttributeValue>
    </Attribute>
    <Attribute AttributeId="&service-type;" IncludeInResult="false">
      <AttributeValue DataType="&string;">&WFS-1.1;</AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes Category="&message;">
    ...<!-- a <Content> element or <Attribute> element set based representation
of the intercepted OWS message -->
  </Attributes>
  <Attributes Category="&environment;">
    <Content>
      <EnvironmentState>
        <disasters>
          <disaster>
            <type>fire</type>
            <startTime>2011-02-01T09:23</startTime>
            <endTime>2011-02-02 T11:16</endTime>
            <spatialExtend>
              <gml:Polygon xmlns:gml="http://www.opengis.net/gml/3.2">
                ...<!-- area affected by disaster -->
              </gml:Polygon>
            </spatialExtend>
          </disaster>
          ...<!-- further ongoing or previous disaster events-->
        </disasters>
      </EnvironmentState>
    </Content>
    <Attribute AttributeId="&current-dateTime;" IncludeInResult="false">
      <AttributeValue DataType="&dateTime;">2011-02-04T12:28</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

Listing 1: SSME model conformant XACML v3.0 ADR

The intercepted OWS message can be included in the ADR in an XML encoded form below the <Content> element and /or through a set of <Attribute> elements (details see [25]). To support the definition of very expressive authorization semantics it is necessary to use the <Content> based representation of OWS messages in XACML ADRs (details see [10][11])

8.2 XACML based implementation of rights referring to machines, services, subjects and environment states with certain properties

This section shows how to define XACML encoded policy elements that refer to machines, services, subjects and environment states with certain properties. To keep the code samples as short as possible, only the interesting parts of the XACML policy elements are demonstrated. All condition expressions presented in the following subsections evaluate to ‘true’, given the evaluation context shown in Listing 1.

8.2.1 Rights refering to certain machines

Assuming that some machine specific attributes are included under the &recipient-subject; category (i.e. the implementation of the service entity type of the SSME model) in the evaluation contexts, one can define rights that refer to machines with specific properties.

Listing 2 shows an XACML v3.0 encoded condition expression that describes the test whether the IP-address of the machine the subjects wants to interact with equals “123.123.123.123”.

```
<Match MatchId="&string-equal;">
  <AttributeValue DataType="&string;">123.123.123.123</AttributeValue>
  <AttributeDesignator Category="&recipient-subject;" AttributeId="&ip-
address;" DataType="&string;" MustBePresent="true"/>
</Match>
```

Listing 2: Condition expression that refers to machines with a specific IP-address

Listing 3 presents another XACML v3.0 condition expression that demonstrates how to check if the machines, the subjects want to interact with, have less than one gigabyte main memory and are running under a Windows operating system.

```
<AllOf>
  <Match MatchId="&string-equal;">
    <AttributeValue DataType="&string;">windows</AttributeValue>
    <AttributeDesignator Category="&recipient-subject;" AttributeId="&os-
recipient-host;" DataType="&string;" MustBePresent="false"/>
  </Match>
  <Match MatchId="&integer-greater-than;">
    <AttributeValue DataType="&integer;">1.000.000.000 </AttributeValue>
    <AttributeDesignator Category="&recipient-subject;" AttributeId="&mem-
recipient-host;" DataType="&integer;" MustBePresent="false"/>
  </Match>
</AllOf>
```

Listing 3: Condition expression that refers to machines with a specific hardware and software configuration

8.2.2 Rights refering to certain services

Context Handlers that conform to the guidelines defined in the core requirement class &xop;/RC/1.1 of the XACML v3.0 OWS profile include the &service-type; and &service-url; XACML <Attribute> elements under the &recipient-subject; category.

Listing 4 contains the definition of a condition expression that tests if the subject intends to interact with a specific WFS 1.1 service instance.

```
<Match MatchId="&string-equal;">
```



```
  <AttributeValue DataType="&string;">http://domainA.com/wfs</AttributeValue>
  <AttributeDesignator Category="&recipient-subject;" AttributeId="&service-
url;" DataType="&string;" MustBePresent="false"/>
</Match>
```

Listing 4: Condition expression that refers to a specific service instance

In contrast the condition expression defined in Listing 5 evaluates if the subject wants to communicate with any instance of the WFS 1.1 service class.

```
<Match MatchId="&string-equal;">
  <AttributeValue DataType="&string;">&WFS-1.1;</AttributeValue>
  <AttributeDesignator Category="&recipient-subject;" AttributeId="&service-
type;" DataType="&string;" MustBePresent="false"/>
</Match>
```

Listing 5: Condition expression that refers to a specific service class

8.2.3 Rights refering to certain subjects

An XACML v3.0 OWS profile conformant Context Handler includes subject specific information below the &access-subject; category in form of XACML <Attribute> elements and/or in form of a <Content> element. According to the XACML v3.0 RBAC profile subject attributes like the activated roles are e.g. included below the &access-subject; category are represented by a set of <Attribute> elements. We assume that the Context Handler addsvnext to the &role; <Attribute> elements a &citizenship; and a &current-location; XACML <Attribute> element below this category when generating ADRs.

The code fragment shown in Listing 6 demonstrates how to check if the interacting subject has activated a role named &citizen;

```
<Match MatchId="&string-equal;">
  <AttributeValue DataType="&string;">&citizen;</AttributeValue>
  <AttributeDesignator Category="&access-subject;" AttributeId="&role;"
DataType="&string;" MustBePresent="false"/>
</Match>
```

Listing 6: Condition expression that refers to subjects with a specific activated role

Listing 7 further demonstrates how to formulate condition expressions that test if the currently active subject is german and is currently located within Germany.

```
<AllOf>
  <Match MatchId="&string-equal;">
    <AttributeValue DataType="&string;">german</AttributeValue>
    <AttributeDesignator Category="&access-subject;"
AttributeId="&citizenship;" DataType="&string;" MustBePresent="false"/>
  </Match>
  <Match MatchId="&contains;">
    <AttributeValue DataType="&geometry;">
      <gml:Polygon ...><!-- area of Germany -->...</gml:Polygon>
    </AttributeValue>
    <AttributeDesignator Category="&access-subject;" AttributeId="&current-
location;" DataType="&geometry;" MustBePresent="false"/>
  </Match>
```



```
</AllOf>
```

Listing 7: Condition expression that refers to subjects with a specific location and citizenship

8.2.4 Rights refering to certain environment states

To demonstrate the implementation of environment state dependant rights, we assume that the <Content> element below the &environment; category of the generated XACML v3.0 ADRs describe the current disaster situation. Further we assume that there is an additional XACML <Attribute> element below this category that represents the current date and time. Listing 8 shows a sample definition of such a &environment; ADR category.

```
<Attributes Category="&environment;">
  <Content>
    <EnvironmentState>
      <disasters>
        <disaster>
          <type>fire</type>
          <startTime>2011-02-01T09:23</startTime>
          <endTime>2011-02-02 T11:16</endTime>
          <spatialExtend>
            <gml:Polygon ...>
              <!-- area affected by disaster -->...
            </gml:Polygon>
          </spatialExtend>
        </disaster>
        ...<!-- further ongoing or previous disaster events-->
      </disasters>
    </EnvironmentState>
  </Content>
  <Attribute AttributeId="&current-dateTime;" IncludeInResult="false">
    <AttributeValue DataType="&dateTime;">2011-02-04T12:28</AttributeValue>
  </Attribute>
</Attributes>
```

Listing 8: &environment; category describing a specific disaster situation and the current date and time

The condition expression shown in Listing 9 demonstrates how to define a right that refers to disasters events that started at most 5 days earlier. Hence this example not only presents how to express environment state specific rights but also highlights how to define time dependant rights. Note that the condition expression visualized below can only be defined as child of XACML <Condition> elements because of the required nesting of XACML functions.

```
<Condition>
  <Apply FunctionId="&any-of;">
    <Function FunctionId="&dateTime-greater-than;"/>
    <Apply FunctionId="&dateTime-subtract-dayTimeDuration;">
```

```
      <AttributeDesignator Category="&environment;" AttributeId="&current-
dateTime;" DataType="&dateTime;" MustBePresent="false"/>
      <AttributeValue DataType="&dayTimeDuration;">P5D</AttributeValue>
    </Apply>
    <AttributeSelector Category="&environment;"
Path="/EnvironmentState/disasters/disaster/startTime/text()"
DataType="&dateTime;" MustBePresent="false"/>
  </Apply>
</Condition>
```

Listing 9: Condition expression that refers to certain environment states

8.3 XACML based implementation of rights referring to WFS messages

The aim of this section is to demonstrate how to define rights referring to the different types of WFS messages. It is assumed that the transformation of the intercepted WFS messages to their ADR specific representation happens as mandated by the conformance classes &xop;/RC/1.2, &xop;/RC/1.3(&WFS: 2.0;), &xop;/RC/1.4(&WFS:2.0;), &xop;/RC/1.9(&WFS:2.0;) and &xop;/RC/1.11(&WFS:2.0;) defined the XACML v3.0 OWS profile and its WFS specific extension document.

8.3.1 Rights referring to /Transaction/Insert requests

Based on a closed world assumption the XACML v3.0 rule shown in Listing 10 demonstrates how to implement a right that verifies that all insert-able features are members of the Building feature class and have a location attribute that represents a geometry within Germany. The lines 3-8 guarantee that the rule is only applied if the &content-selector; XACML attributes of the derived individual ADRs refers to the children nodes of /wfs:Transaction/wfs:Insert nodes. This implies that the global ADRs should at least refer to /wfs:Transaction/wfs:Insert nodes and their direct descendents.

```
<Rule RuleId="12345" Effect="Permit">
  <Target><AnyOf><AllOf>
  <Match MatchId="&xpath-node-equal;">
    <AttributeValue DataType="&xpath;"
XPathCategory="&message;"...>/wfs:Transaction/wfs:Insert/*</AttributeValue>
    <AttributeDesignator AttributeId="&content-selector;" DataType="&xpath;"
Category="&message;" MustBePresent="false"/>
  </Match>
  <Match MatchId="&string-equal;">
    <AttributeValue DataType="&string;">Building</AttributeValue>
    <AttributeSelector Category="&message;" Path="name(.)"
ContextSelectorId="&content-selector;" DataType="&string;"
MustBePresent="false" />
  </Match>
  <Match MatchId="&integer-equal;">
    <AttributeValue DataType="&string;">1</AttributeValue>
    <AttributeSelector Category="&message;" Path="count(./Location)"
ContextSelectorId="&content-selector;" DataType="&string;"
MustBePresent="false" />
  </Match>
  <Match MatchId="&contains;">
```



```
    <AttributeValue DataType="&geometry;">
      <gml:Polygon ...><!-- area of Germany -->...</gml:Polygon>
    </AttributeValue>
    <AttributeSelector Category="&message;" Path="./Location/Polygon"
ContextSelectorId="&content-selector;" DataType="&geometry;"
MustBePresent="false" />
  </Match>
</AllOf></AnyOf></Target></Rule>
```

Listing 10: Rule that verifies various properties of insert-able features

8.3.2 Rights referring to /GetFeature requests

Listing 11 describes an XACML rule that refers to GetFeature requests. Through the defined projection clause specific condition expression the right refers to a specific feature type and an optional attribute class. Note that the correct evaluation of this right requires the normalization of the WFS projection clauses (as defined in the WFS specific extension of the XACML v3.0 OWS profile [26]) before inserting the WFS requests in the ADRs.

```
<Rule RuleId="abcdefg" Effect="Deny">
  <Target><AnyOf><AllOf>
    <Match MatchId="&xpath-node-equal;">
      <AttributeValue DataType="&xpath;"
XPathCategory="&message;">/wfs:GetFeature/wfs:Query</AttributeValue>
      <AttributeDesignator AttributeId="&content-selector;" DataType="&xpath;"
Category="&message;" MustBePresent="false"/>
    </Match>
    <Match MatchId="&string-equal;">
      <AttributeValue DataType="&string;">Building</AttributeValue>
      <AttributeSelector Category="&message;" Path="./@typeName"
ContextSelectorId="&content-selector;" DataType="&string;"
MustBePresent="false" />
    </Match>
    <Match MatchId="&string-equal;">
      <AttributeValue DataType="&string;">Price</AttributeValue>
      <AttributeSelector Category="&message;" Path="PropertyName/text()"
ContextSelectorId="&content-selector;" DataType="&string;"
MustBePresent="false" />
    </Match>
  </AllOf></AnyOf></Target>
</Rule>
```

Listing 11: Restricting read access of building's price attributes

Listing 12 demonstrates how to define rules that can cause the rewrite of the selection clause of intercepted WFS GetFeature requests. The <Target> element of the rule ensures that rule only applies to individual ADRs that describe that a subject named Alice has sent an GetFeature request that has <Query> elements included selecting data from the

Building feature class. If these conditions hold for an incoming individual ADR, the rewrite obligation of this positive rule will be discharged. This causes that the selection predicat “within(location, PolygonGermany)” is added appropriately under a <Filter> element below the <Query> element under consideration. The XSLT stylesheet defined in the rewrite obligation causes the required transformation. Note that the stylesheet has to deal with three different cases:

A) there was no <Filter> element below the currently processed Query element,

B) there was a <Filter> element with no <And> child

C) there was a Filter/AND node.

Next to the XSLT stylesheet there are two further <AttributeAssignmentExpression> elements below the rewrite-obligation. One tells the Context Handler which ADR representation form of the intercepted message shall be transformed back into its original encoding and the other is used to pass parameters to the stylesheet (here the node the individual ADR refers to - cp. the &content-selector; element of the individual ADR). Details on the use of the <AttributeAssignmentExpression> elements in rewrite-obligations can be found in [25], section 6.12.

```
<Rule Effect="Permit"...>
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="&string-equal;">
          <AttributeValue DataType="&string;">Alice</AttributeValue>
          <AttributeDesignator Category="&access-subject;"
AttributeId="&subject-id;" DataType="&string;" MustBePresent="true"/>
        </Match>
        <Match MatchId="&xpath-node-equal;">
          <AttributeValue DataType="&xpath;" Category="&message;">
/wfs:GetFeature/wfs:Query[@typeName="Building"]
          </AttributeValue>
          <AttributeDesignator AttributeId="&content-selector;"
DataType="&xpath;" Category="&message;" MustBePresent="true"/>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <ObligationExpressions>
    <ObligationExpression ObligationId="rewrite-examp."
FulfillOn="Permit">
      <!-- the rewrite expressions in form of an XSLT Stylesheet -->
```



```
      <AttributeAssignmentExpression AttributeId="&xslt-rewrite-
stylesheet">
        <AttributeValue DataType="&xslt;">
          <xsl:stylesheet ... version="2.0">
            <!-- hook for argument passing mechanism -->
            <xsl:param name="&init-select-node;:&xslt-
argument;:content-selector" select="<!--dynamically-assigned-by-
context-handler -->"/>
            <!-- static predicate that shall be added to selection
predicate of the intercepted GetFeature request -->
            <xsl:param name="predicate-to-add">
              <ogc:Within>
                <ogc:PropertyName>location</ogc:PropertyName>
                  <gml:Polygon srsName="osgb:BNG">
                    <gml:outerBoundaryIs><gml:LinearRing>
                      <gml:coordinates> 528000.000,178856.330 ...
                      </gml:coordinates>
                    </gml:LinearRing></gml:outerBoundaryIs>
                  </gml:Polygon>
                </ogc:Within>
              </xsl:param>
            <xsl:template match="node()|@*">
              <xsl:choose>
                <xsl:when test="self::node() = $&init-select-
node;:&xslt-argument;:content-selector">
                  <xsl:call-template name="modify-query"/>
                </xsl:when>
                <xsl:otherwise>
                  <xsl:copy>
                    <xsl:apply-templates select="node()|@*"/>
                  </xsl:copy>
                </xsl:otherwise>
              </xsl:choose>
            </xsl:template>
            <xsl:template name="modify-query">
            <xsl:copy>
              <xsl:apply-templates select="@*"/>
              <xsl:apply-templates select="wfs:PropertyName"/>
              <ogc:Filter>
                <xsl:if test="not(ogc:Filter)">
                  <xsl:copy-of select="$predicate-to-add"/>
                </xsl:if>
                <xsl:if test="ogc:Filter">
```



```
                  <ogc:And>
                    <xsl:if test="not(ogc:Filter/ogc:And)">
                      <xsl:copy-of select="$predicate-to-add"/>
                      <xsl:apply-templates select="ogc:Filter/node()"/>
                    </xsl:if>
                    <xsl:if test="ogc:Filter/ogc:And">
                      <xsl:apply-templates
select="ogc:Filter/ogc:And/node()"/>
                      <xsl:copy-of select="$predicate-to-add"/>
                    </xsl:if>
                  </ogc:And>
                </xsl:if>
              </ogc:Filter>
              </xsl:copy>
            </xsl:template>
          </xsl:stylesheet>
        </AttributeValue>
      </AttributeAssignmentExpression>
      <!-- argument that shall be passed to the xslt stylesheet -->
      <AttributeAssignmentExpression AttributeId="&xslt-argument;:arg1"
Category="&xop;:category:obligation">
        <AttributeDesignator AttributeId="&content-selector;"
DataType="&xpath;" Category="&message;" MustBePresent="true"/>
      </AttributeAssignmentExpression>
      <AttributeAssignmentExpression AttributeId="&adr-representation-
to-map;" Category="&xop;:category:obligation">
        <AttributeValue
DataType="&string;">&xop;:category:message:content
</AttributeValue>
      </AttributeAssignmentExpression>
    </ObligationExpression>
  </ObligationExpressions>
</Rule>
```

Listing 12: XSLT based definition of an XACML v3.0 rewrite rule

Figure 17 shows the effect of applying the rule and rewrite obligation respectively shown in the listing above on a sample global XACML ADR.

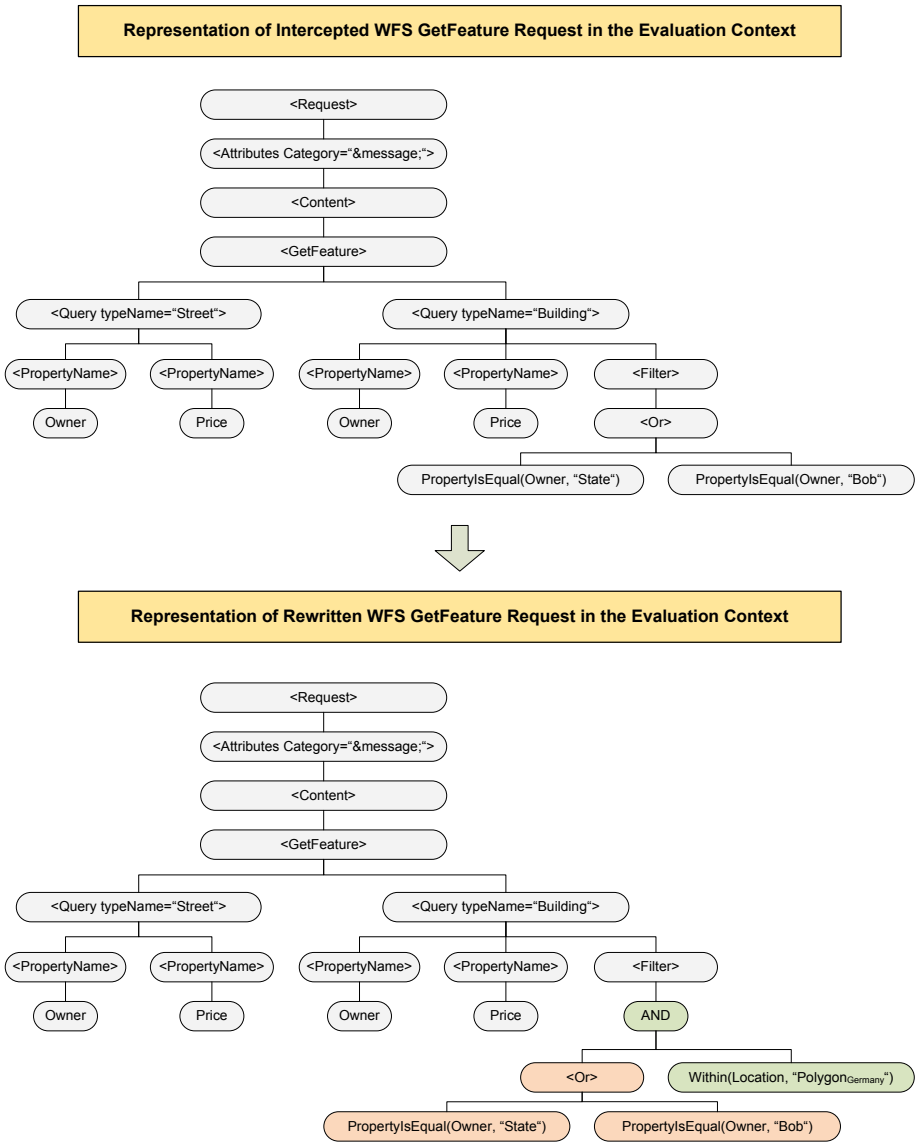

Figure 17: Rewrite effects of the sample rewrite rule defined in Listing 12

Listing 13 demonstrates how to define response based rights (if that cannot be avoided).

```
<Rule RuleId="abcdefg" Effect="Deny">
  <Target><AnyOf><AllOf>
    <Match MatchId="&xpath-node-equal;">
      <AttributeValue DataType="&xpath;" XPathCategory="&message;">
        /wfs:FeatureCollection/FeatureMember/Building/Price
      </AttributeValue>
      <AttributeDesignator AttributeId="&content-selector;" DataType="&xpath;"
Category="&message;" MustBePresent="false"/>
    </Match>
```



```
    <Match MatchId="&integer-less-than;">
      <AttributeValue DataType="&integer;">1,000,000</AttributeValue>
      <AttributeSelector Category="&message;" Path="./text()"
ContextSelectorId="&content-selector;" DataType="&integer;"
MustBePresent="false"/>
    </Match>
  </AllOf></AnyOf></Target>
</Rule>
```

Listing 13: Response based rule that refers to buildings' price properties with a value greater than one million

8.3.3 Rights referring to /Transaction/Delete requests

The rule defined in Listing 14 shows how to control the PIP through a PIP-control obligation. The demo scenario is as follows: A user wants to delete some building feature instances but he is only allowed to delete building features that are within Germany. Further the user shall get an authorization decision that tells him if it's intended delete request is permitted or not. In the example it shall be not acceptable to calculate a rewritten request that would imply that only the intersection of to-be-deleted and allowed-to-be-deleted feature instances will actually be deleted. The subject must always know if the intended features will be deleted or not, before any delete action is committed. This requirement rules out the rewriting approach. To implement the required authorization semantics one needs to define a rule with a PIP-control obligation (cp. Listing 14).

```
<Rule RuleId="abcdefg" Effect="Permit" >
  <Target><AnyOf><AllOf>
    <Match MatchId="&xpath-node-equal;">
      <AttributeValue DataType="&xpath;"
XPathCategory="&message;">/wfs:Transaction/wfs:Delete</AttributeValue>
      <AttributeDesignator AttributeId="&content-selector;" DataType="&xpath;"
Category="&message;" MustBePresent="false"/>
    </Match>
    <Match MatchId="&string-equal;">
      <AttributeValue DataType="&string;">Building</AttributeValue>
      <AttributeSelector Category="&message;" Path="./@typeName"
ContextSelectorId="&content-selector;" DataType="&string;"
MustBePresent="false" />
    </Match>
  </AllOf></AnyOf></Target>
  <Condition>
    <Apply FunctionId="&all-of;">
      <Function FunctionId="&contains;"/>
      <AttributeValue DataType="&geometry;">
        <gml:Polygon>...<!-- area of Germany --></gml:Polygon>
      </AttributeValue>
      <AttributeSelector Category="&response-to-subrequest;"
Path="/FeatureCollection/FeatureMemeber/Building/Location/Polygon"
DataType="&geometry;" MustBePresent="true" IndeterminantHandler="&pip-control-
obligation;"/>
    </Apply>
  </Condition>
  <ObligationExpressions>
    <ObligationExpression ObligationId="&pip-control-obligation;">
```



```
      <AttributeAssignmentExpression AttributeId="&xslt-to-generate-pip-
request;">
        <AttributeValue DataType="&xslt;">
          <xslt:transform xmlns:xslt="http://www.w3.org/1999/XSL/Transform">
            ...<!-- a xslt style sheet that transforms the intercepted
/transaction/delete Element into a /GetFeature/Query Element that selects the
location attributes of the features to be deleted  -->
          </xslt:transform>
        </AttributeValue>
      </AttributeAssignmentExpression>
      <AttributeAssignmentExpression AttributeId="&target-category;">
        <AttributeValue DataType="&string;">&response-to-
subrequest;</AttributeValue>
      </AttributeAssignmentExpression>
      <AttributeAssignmentExpression AttributeId="&target-type;">
        <AttributeValue DataType="&string;">&content-element; </AttributeValue>
      </AttributeAssignmentExpression>
      <AttributeAssignmentExpression AttributeId="&service-url;">
        <AttributeDesignator Category="&recipient-subject;"
AttributeId="&service-url;" DataType="&string;" MustBePresent="true"/>
      </AttributeAssignmentExpression>
    </ObligationExpression>
  </ObligationExpressions>
</Rule>
```

Listing 14: Controlling the PIP through PIP-control obligations

The rule shown above gets evaluated in case the subject submits a /Transaction/Delete query that refers to the building feature class. In case the category &response-to-subrequest; is not present - which is always the case in the first evaluation run of this rule - the PDP generates an indeterminate authorization decision response that indicates that there was some data missing below the <Content> element of the &response-to-subrequest; category. Thanks to the included PIP-control-obligation, the Context Handler can provide the PIP with a dynamically generated sub-request and a destination address. The PIP will forward the sub-request to the corresponding service endpoint and return the response to the sub-request to the Context Handler. In the example this response contains the location of all buildings that shall be deleted. Afterwards the Context Handler augments the original ADR by adding the &response-to-subrequest; category with a <Content> element that now contains the response to the sub-request. The extended ADR is than send to the PDP for a new evaluation run and, thanks to the added category, the rule can now successfully be evaluated to permit or deny. More detailed information on the definition and evaluation of PIP-control obligations can be found in [25].

9 Evaluation

In this section we introduce a sample set of business rules involved in the process of managing Special Activity Airspace Schedule information taken from [7] (cp. 9.1). Additionally another set of AIXM specific business rules is presented in section 9.2. The (Geo)XACML policy reflecting the introduced business rules can be found under: http://grid01.informatik.unibw-muenchen.de/OWS-8.geoxacml and in Appendix C. Also online available are two simple client pages (http://grid01.informatik.unibw-muenchen.de/snowflake.php and http://grid01.informatik.unibw-muenchen.de/comsoft.php), that allow to test the described and deployed policy when interacting with the Snowflake or Comsoft WFS-T v2.0 respectively.

Note that the enforcement of the policy on test requests happens in the OWS-8 aviation architecture. Hence submitted and authorized WFS-T requests like e.g. insert schedule requests imply the generation of SAA scheduling events.

9.1 FAA sample business rules for the SAA scheduling scenario

The following description of the business rules in the SAA scheduling scenario. The content of the following subsections has been copied from the corresponding sections in [7].

Notes:

Information contained in the presented scenarios comes from the National Special Activity Airspace Program (NSAAP) Concept of Operations[3], Final Requirements Document[4], and FAA Order 7400.8[5]. Other information is based on the understanding of the requirements for the management of Special Activity Airspace in the NAS of the author of [12], which served as the primary source for the following sections. Necessary background information on AIXM can be found here http://www.aixm.aero/public/standard_page/download.html.

All business rules mentioned in this section should not be considered as definitive or validated in any manner. The information provided here is solely for the purpose of the

[3] Federal Aviation Administration, *Operational Concept for Special Activity Airspace (SAA),* Version 1.0, September 2, 2010.

[4] Federal Aviation Administration, *Functional Requirements Document for National Special Activity Airspace Project (NSAAP),* Version 0.93, April 4, 2011.

[5] Federal Aviation Administration, *FAA Order JO 7400.8S Special Use Airspace,* February 16, 2010.

development of test scenarios for the OWS-8 Authoritative Data Source Study and the encoding of AIXM Business Rules in (Geo)XACML.

9.1.1 Subject-Role assignment

Table 2 shows the assignment of subjects to roles and their facilities.

Name	Role	Facility
April	Military Operations Specialist	FAA, JACKSONVILLE ARTCC
Bill	Air Traffic Controller	FAA, JACKSONVILLE ARTCC
Carmen	Military Operations Specialist	FAA, NEW YORK ARTCC
Doug	Air Traffic Controller	FAA, NEW YORK ARTCC
Edward	SAA Scheduler	USAF, AIR ARMAMENT CENTER, EGLIN AFB
Eric	SAA Scheduler	US ARMY, FORT DIX
Frank	SAA Scheduler	US NAVY, FLEET AREA CONTROL AND SURVEILLANCE FACILITY JACKSONVILLE
Gary	General Internet User	N/A

Table 2: User-Role assignment

9.1.2 Role-Permission assignment

Role: Military Operations Specialists (MOS)

- Can query all SAA data
- Can insert schedule requests and pending, disapproved, and approved SAA schedules for airspaces for which their facility is the Controlling Agency
  - ***BR001:*** *Military Operations Specialists can insert SAA schedules with AIXM element "**reservationPhase**" of **pending**, **disapproved**, and **approved**.*
  - ***BR002:*** *Military Operations Specialists can insert SAA schedule if their facility is the same as the Controlling Agency Unit name of the airspace they are attempting to schedule.(The controlling agency is the unit associated with the airspace that has associated ATC Service type as "ACS". Appendix A provides the gml:identifiers and names of the controlling and using agencies for each airspace).*

Role: Air Traffic Controller (ATC)

- Can query all SAA data
- Cannot insert any data into the DB

Role: SAA Scheduler (SAAS)

- Can query all SAA data
- Can insert SAA schedule requests and pending schedules for airspaces for which their facility is the Using Agency.
    - ***BR003:*** *SAA scheduler can insert SAA schedules with AIXM element "**reservationPhase**" of **pending**. (They cannot insert "**reservationPhase**" of **approved** or **disapproved**.)*
    - ***BR004:*** *SAA scheduler can insert **SAA schedule** if their **facility** is the same as the **Using Agency** Unit name of the airspace they are attempting to schedule. .(The using agency is the unit associated with the airspace that has associated ATC Service type as "OTHER". Appendix B provides the gml:identifiers and names of the controlling and using agencies for each airspace).*

Role: General Internet User (GIU)

- Can query a limited set of SAA data (the individual elements that can be queried are provided in Appendix B)
- Cannot insert any data into the DB

9.1.3 Further sample business rules

The status of the airspace reservation request is critical to determination of permissions in Scenarios 9.1.3.1 - 9.1.3.3. The status of the airspace reservation request is defined by the AIXM element *reservationPhase* in the SAA AIXM extension. Scenario 9.1.3.1 will address permissions for a *reservationPhase* of **PENDING**. Scenario 9.1.3.2 and 9.1.3.3 will address permissions for a *reservationPhase* of **APPROVED**.

9.1.3.1 Pending SAA Schedule Insert Authorization Identification based on query of additional data source

In this scenario, the users attempt to insert a pending Special Activity Airspace schedule for EGLIN C MOA, FL into the database. Based on the permissions of the user, the system determines if the action is authorized and responds accordingly. The event-trace diagram provided below (cp. Figure 18) details the steps involved in this event and the process is also further described in the text below.

The user submits the pending SAA Schedule insert request and the system must determine if the user is authorized to perform this action. The policy for SAA schedule inserts identifies that

1. The incoming message must be a **pending** SAA schedule.
2. The user's role must have permissions to perform insert function for a pending SAA schedule.
   - ***BR005: The user must belong to a user role that has permission to insert a pending SAA Schedule into the database****. (The roles with this permission are SAA Scheduler and Military Operations Specialist- see section 9.1.2).*
3. For a **pending** SAA schedule the **users must be assigned to a facility that is either the controlling agency or the using agency for the special activity airspace they are attempting to schedule**.
   - ***BR006:*** *To insert a pending SAA Schedule into the system, the users facility must be either the controlling agency or the scheduling agency for the SAA the user is attempting to insert a SAA Schedule request.*

The first rule above is simply evaluated based on the elements contained in the AIXM message. The second rule is also simply determined based on the role of the user. The third rule, however, requires additional processing through PIP-control-obligations that determine the Controlling Agency and Using Agency for the SAA the user is attempting schedule through the WFS-T Insert operation. This information is stored in the definition of the SAA can be queried using the WFS GetFeature operation. Here, we use the Snowflake SAA sample data provided for the SAA pilot study. The query requests the Controlling Agency and Using Agency for EGLIN C MOA, FL, which will be returned as FAA, JACKSONVILLE ARTCC (Controlling Agency) and USAF, AIR ARMAMENT CENTER, EGLIN AFB (Using Agency), respectively. To be authorized to insert a schedule request for the airspace the facility of the user must either be FAA, JACKSONVILLE ARTCC or USAF, AIR ARMAMENT CENTER, EGLIN AFB.

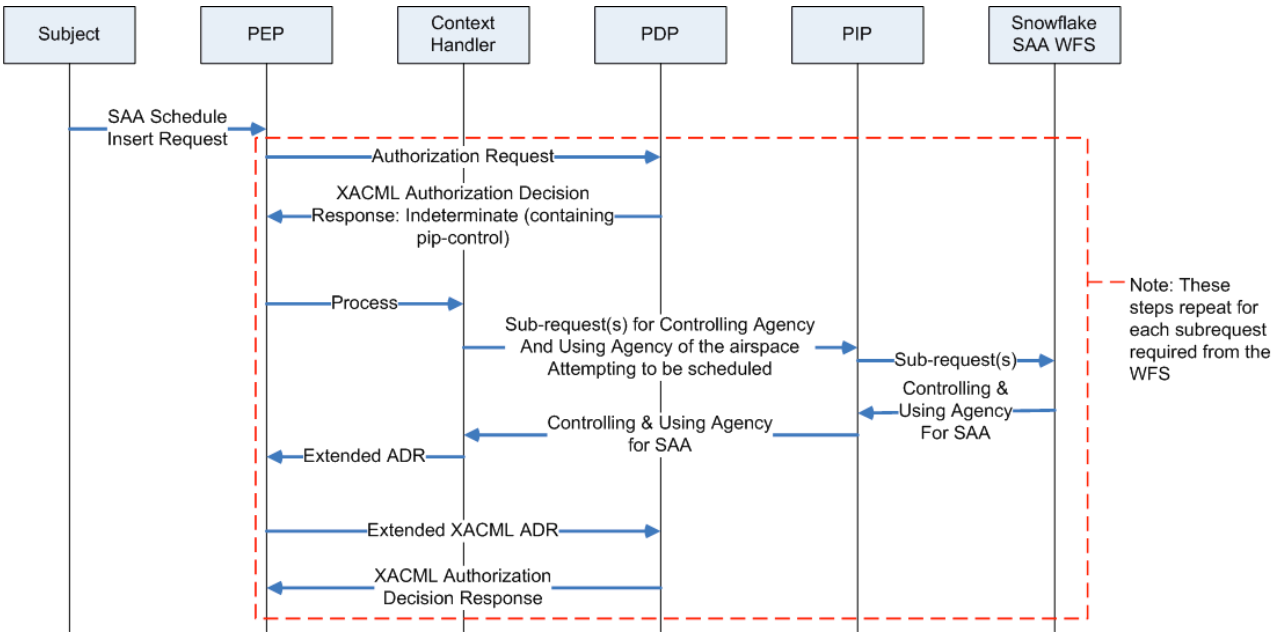

Figure 18: Enforcement of business rule BR006[6]

Based on the roles, rules, and scenario provided above, the following results would be expected for the individual users when they attempt to insert the pending SAA schedule for EGLIN C MOA, FL.

Name	Role	Facility	Insert Result	Rule(s) Broken
April	MOS	FAA, JACKSONVILLE ARTCC	Allow	none
Bill	ATC	FAA, JACKSONVILLE ARTCC	Deny	BR005
Carmen	MOS	FAA, NEW YORK ARTCC	Deny	BR006 (BR002)
Doug	ATC	FAA, NEW YORK ARTCC	Deny	BR005, BR006
Edward	SAAS	USAF, AIR ARMAMENT CENTER, EGLIN AFB	Allow	none
Eric	SAAS	FORT DIX	Deny	BR006 (BR004)
Frank	SAAS	FLEET AREA CONTROL AND SURVEILLANCE FACILITY JACKSONVILLE	Deny	BR006 (BR004)

[6] In the following sequence diagram the Obligation Handler is assumed to be implemented in the Context Handler component.

Gary	GIU	N/A	Deny	BR005, BR006

Table 3: Expected results when inserting the pending SAA schedule for EGLIN C MOA, FL

9.1.3.2 Approved SAA Schedule Request Insert Authorization Identification based on query of additional data source

In this scenario, the users attempt to schedule an SAA by insert of an approved Special Activity Airspace schedule for EGLIN C MOA, FL into the database. Based on the permissions of the user, the system determines if the action is authorized and responds accordingly. The event-trace diagram provided below (cp. Figure 19) details the steps involved in this event and the process is also further described in the text below.

The user submits the **approved** SAA Schedule insert request and the system must determine if the user is authorized to perform this action. The policy for SAA schedule inserts identifies that

1. The incoming message must be an **approved** SAA schedule type.
2. The user's role must have permissions to perform insert function for an approved SAA schedule.
   - ***BR007: The user must belong to a user role that has permission to insert an approved SAA Schedule into the database****. (The role with this permission is a Military Operations Specialist- see section 9.1.2).*
3. For an approved SAA schedule the users must be assigned to the facility that is the controlling agency for the special activity airspace they are attempting to schedule.
   - ***BR008:*** *To insert an approved SAA Schedule into the system, the* ***users facility must be the controlling agency for the SAA for which they are attempting to insert an approved SAA Schedule****.*

The first rule above is simply evaluated based on the elements contained in the AIXM message. The second rule is also simply determined based on the role of the user. The third rule, however, requires additional processing through the PIP-Control to determine the Controlling Agency for the SAA the user is attempting schedule through the WFS Insert operation. This information is stored in the definition of the SAA can be queried using the WFS GetFeature operation. Here, we use the Snowflake SAA sample data provided for the SAA pilot study. The query requests the Controlling Agency for EGLIN C MOA, FL, which will be returned as FAA, JACKSONVILLE ARTCC. To be authorized to insert an approved schedule for the airspace the facility of the user must be FAA, JACKSONVILLE ARTCC.

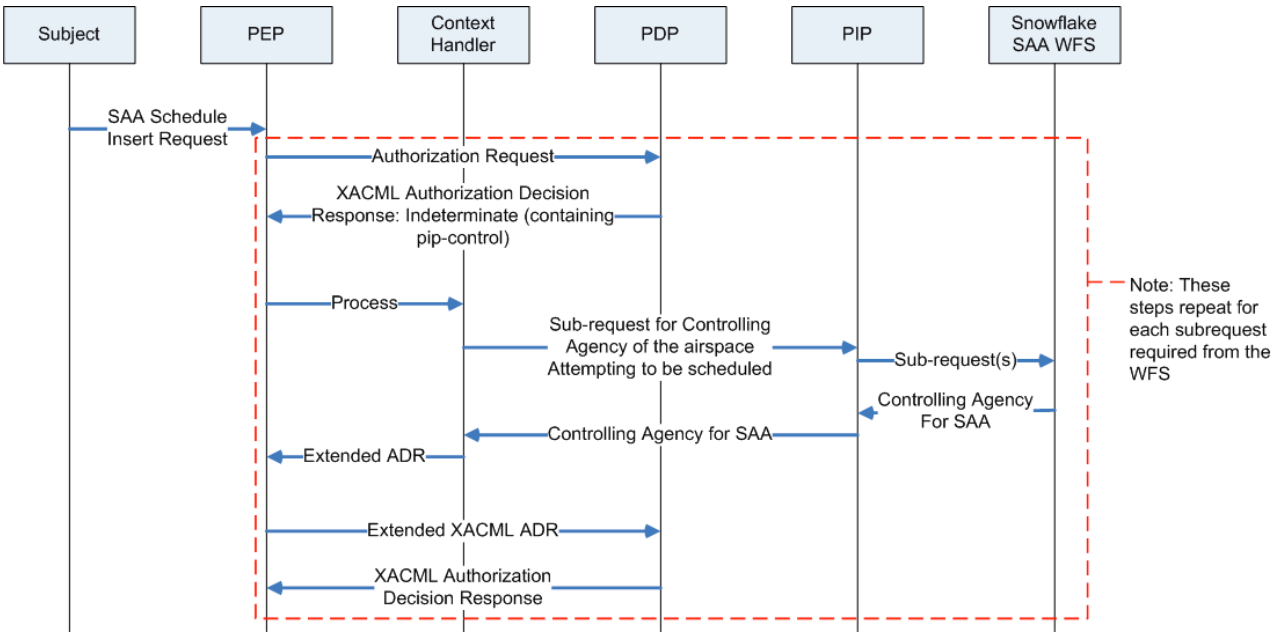

Figure 19: Enforcement of business rule BR008

Based on the roles, rules, and scenario provided above, the following results would be expected for the individual users when they attempt to insert the SAA schedule for EGLIN C MOA, FL.

Name	Role	Facility	Insert Result	Rule(s) Broken
April	MOS	FAA, JACKSONVILLE ARTCC	Allow	none
Bill	ATC	FAA, JACKSONVILLE ARTCC	Deny	BR007
Carmen	MOS	FAA, NEW YORK ARTCC	Deny	BR008 (BR002)
Doug	ATC	FAA, NEW YORK ARTCC	Deny	BR007, BR008
Edward	SAAS	USAF, AIR ARMAMENT CENTER, EGLIN AFB	Deny	BR007, BR008
Eric	SAAS	FORT DIX	Deny	BR007, BR008 (BR004)
Frank	SAAS	FLEET AREA CONTROL AND SURVEILLANCE FACILITY JACKSONVILLE	Deny	BR007, BR008 (BR004)
Gary	GIU	N/A	Deny	BR007, BR008

Table 4: Expected results when inserting the pending SAA schedule for EGLIN C MOA, FL

9.1.3.3 Approved SAA Schedule Request Insert Authorization Identification based on geography determination

This scenario is identical to the scenario described under section 9.1.3.2 except that instead of using an the name of the controlling agency and users' facility for determination of the facilities allowed to insert an approved SAA schedule, the determination is made using GeoXACML to determine if the airspace attempting to be scheduled is within the geographical boundaries of the users' facility.

Note: This may not be a valid rule that could be used for SAA scheduling, as there are a few SAAs that may actually cross the geographical boundary of two Controlling authorities. In these situations the airspace is assigned to one of the facilities whose area of control it touches. This seems to occur primarily with restricted areas. However, this scenario is likely valid for the Military Operations Area chosen as an example airspace. The main reason for choosing this scenario—even though the scenario itself may not be fully valid—is to test the capability of GeoXACML and the ability to determine authorization based on geographic features for AIXM. Valid use of this standard would require further research to identify the rules that could be enforced using geometry as a rule reference.

In this scenario, the users attempt to schedule a Special Activity Airspace by insert of an **approved** SAA schedule request for EGLIN C MOA, FL into the database. Based on the permissions of the user, the system determines if the action is authorized and responds accordingly. The process is further described in the text below.

The user submits the approved SAA Schedule insert request and the system must determine if the user is authorized to perform this action. The policy for SAA schedule inserts identifies that

1. The incoming message must be a schedule request for EGLIN C MOA, FL.
2. The user's role must have permissions to perform insert function for an **approved** SAA schedule.
   - ***BR007:*** *The user must belong to a user role that has permission to insert an approved SAA Schedule into the database. (The role with this permission is a* ***Military Operations Specialist****- see section 9.1.2).*
3. The polygon that defines the airspace the user is attempting to schedule must be **Within** the polygon that defines the area of authority of the users' facility.

- ***BR009:*** *To insert a new SAA Schedule into the system, the geometry of the airspace attempting to be scheduled must be within* ***the allowed scheduling geometry of the user's facility****.*

The first rule above is simply evaluated based on the elements contained in the AIXM message. The second rule is also simply determined based on the role of the user. The third rule, however, requires additional processing to

1. determine the area of responsibility of the users facility
   - *Note: the area of responsibility of the users facilities in this scenario are defined in section 9.1.1. If these are stored in an external database an additional step may be required to query for these.*
2. determine if the geometry that defines the airspace schedule request is within the area of responsibility of the user's facility.
   - *Note: the geometry of the airspace is captured in the definition of the airspace attempting to be scheduled. Therefore, determination of this geometry will also require query of the external SAA database for the Boundaries of the SAA.*

For EGLIN C MOA, FL the boundary of the airspace should be returned as a geometry defined by the following GML geometry (cp. Listing 15).

```
<aixm:Surface srsName="urn:ogc:def:crs:OGC:1.3:CRS84" gml:id="Surface1_3220">
 <gml:patches>
  <gml:PolygonPatch>
   <gml:exterior>
    <gml:LinearRing>
     <gml:posList srsDimension="2" count="6">-86.416 30.967 -86.216 30.967 -
86.175 30.729 -86.333 30.725 -86.416 30.883 -86.416 30.967</gml:posList>
    </gml:LinearRing>
   </gml:exterior>
  </gml:PolygonPatch>
 </gml:patches>
</aixm:Surface>
```

Listing 15: GML encoded boundary of the EGLIN C MOA, FL airspace

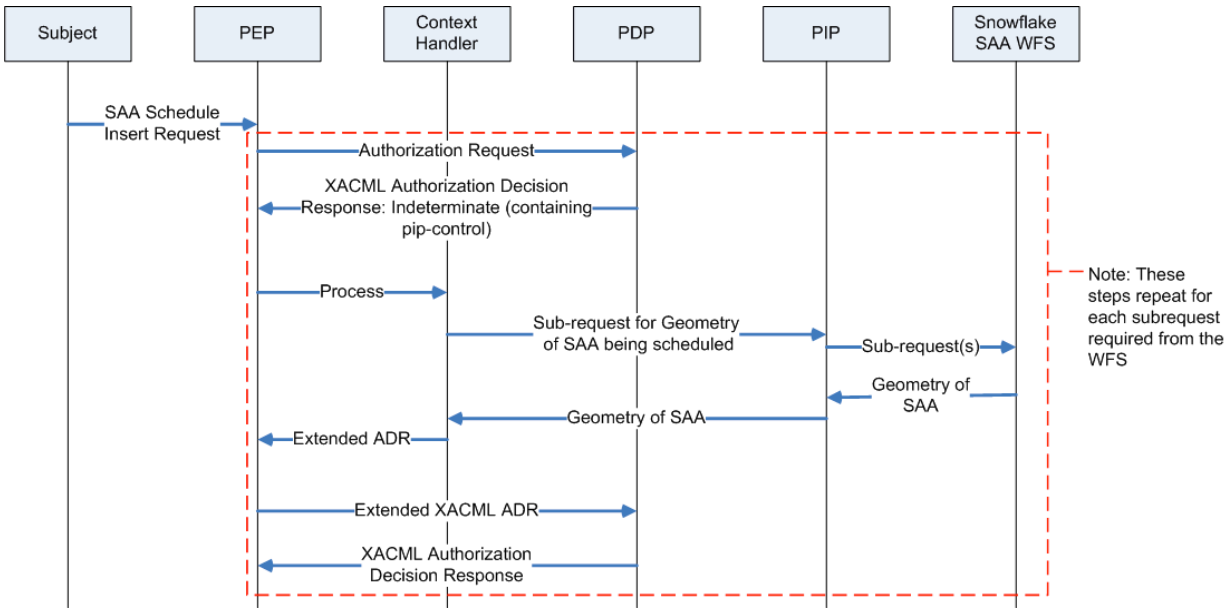

Figure 20: Enforcement of business rule BR009

Based on the roles, rules, and scenario provided above, the following results would be expected for the individual users when they attempt to insert the SAA schedule for EGLIN C MOA, FL.

Name	Role	Facility	Insert Result	Rule(s) Broken
April	MOS	FAA, JACKSONVILLE ARTCC	Allow	none
Bill	ATC	FAA, JACKSONVILLE ARTCC	Deny	BR007
Carmen	MOS	FAA, NEW YORK ARTCC	Deny	BR009
Doug	ATC	FAA, NEW YORK ARTCC	Deny	BR007, BR009
Edward	SAAS	USAF, AIR ARMAMENT CENTER, EGLIN AFB	Deny	BR007, BR009
Eric	SAAS	FORT DIX	Deny	BR007, BR009
Frank	SAAS	FLEET AREA CONTROL AND SURVEILLANCE FACILITY JACKSONVILLE	Deny	BR007, BR009

Gary	GIU	N/A	Deny	BR007, BR009

Table 5: Expected results when inserting the SAA schedule for EGLIN C MOA, FL

9.1.3.4 SAA schedule data query scenario

In this scenario, the users attempt to query for SAA schedules for EGLIN C MOA, FL using the WFS GetFeature operation. Based on the permissions of the user, the system determines if the action is authorized and responds accordingly.

The user submits the SAA Schedule query request and the system must determine if the user is authorized to perform this action. The policy for SAA schedule query identifies the individual permission of the users based on their role. (see section 9.1.1 and appendix B)

Based on the roles, rules, and scenario provided above, the following results would be expected for the individual users when they attempt to insert the SAA schedule for EGLIN C MOA, FL.

Name	Role	Facility	Query Result
April	MOS	FAA, JACKSONVILLE ARTCC	Allow
Bill	ATC	FAA, JACKSONVILLE ARTCC	Allow
Carmen	MOS	FAA, NEW YORK ARTCC	Allow
Doug	ATC	FAA, NEW YORK ARTCC	Allow
Edward	SAAS	USAF, AIR ARMAMENT CENTER, EGLIN AFB	Allow
Eric	SAAS	FORT DIX	Allow
Frank	SAAS	FLEET AREA CONTROL AND SURVEILLANCE FACILITY JACKSONVILLE	Allow
Gary	GIU	N/A	Allow, but with the elements the user does not have access to filtered out (see appendix B).

9.2 Sample business rules for the Comsoft WFS-T

The Comsoft WFS-T (expecting SOAP encoded messages) and the underlying Estonia data set is another context in which the developed concepts where evaluated. The list below describes the business rules that need to be enforced for the users Dave, Claris, Jane, Joe, Bob and Alice in this scenario.

Dave

- Dave can do commissioning for feature type aixm:RadarSystem.
- Dave cannot do de-commissioning for feature type aixm:RadarSystem.

Claris

- Claris can do de-commissioning for feature type aixm:RadarSystem.
- Claris cannot do commissioning for feature type aixm:RadarSystem.

Jane

- Jane can obtain features of type Runway within 50km radius of Tartu, Estonia.

Joe

- Joe can obtain features of type aixm:Airspace with interpretation BASELINE and PERMDELTA
- Joe cannot obtain corrections of that featrue type
- Joe cannot obtain features of any other type than aixm:Airspace

Bob

- Bob can execute the GetFeature operation with no limitation.
- Bob cannot execute the WFS-T operations CREATE, DELETE, and UPDATE. This limitation ensures that Bob can only use the subset of the WFS-T 2.0 capabilities as supported by the AIXM.
- Bob can execute the WFS-T INSERT operation
- Bob can obtain features of type Runway NOT limited to within 50km radius of Tartu, Estonia

Alice

- Alice has not permission to use the Comsoft WFS.

9.3 (Geo)XACML based implementation of the OWS-8 Exqmple Business Rules

The business rules described in the last two sections have been formally described in an (Geo)XACML policy. Due to the size of the policy, it is not included here or in the appendix of this document. However the policy is available online (see http://grid01.informatik.unibw-muenchen.de/OWS-8.geoxacml). The concepts used to define the required sample rights have already been introduced in section 8.2.

Note that opposed to section 8 were all (Geo)XACML examples were XACML v3.0 conformant, the implementation of the sample FAA and Comsoft/Estonia business rules are conformant to the XACML v2.0 specification. XACML v2.0 was used because the standardization status of the XACML v3.0 specification at the time of writing was still committee specification. It was hence decided to use the latest officially adopted XACML standard, which currently is version 2.0. Note that it is fairly straight forward to map the XACML v2.0 policy into a semantically equivalent XACML v3.0 policy. In doing so one benefits from the various features incorporated in the new 3.0 version of the XACML related specifications.

For the Business Rules provided by the FAA, it turned out that based on the fictitious use case (see 9.1.3.3) the URI-based association between arispaces, controlling and using agencies and user facilities could be (partially) replaced by topological conditions, expressed in the GeoXACML policy. This is possible based on the geometry of the airspace and the geometries representing the authorized areas for the user facility. The following snippet from the OWS-8 policy shows this capability:

```
<Condition>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
    <!-- User facility has authorization area that includes airspace -->
    <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-within">
      <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-one-and-only">
        <AttributeSelector RequestContextPath="…/AirTrafficControlServiceTimeSlice/ACS
         /Airspace/horizontalProjection/ElevatedSurface"
         DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry"/>
      </Apply>
      <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-one-and-only">
        <SubjectAttributeDesignator AttributeId="USER_FACILITY_AUTH_AREA"
         DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry"/>
      </Apply>
    </Apply>
  </Apply>
</Condition>
```

Listing 16: GeoXACML Condition verifying that airspace geometry is topological Within the user facility authorized area

10 Implementation of the Access Control System Components

The Access Control System consists of three major components:

1) The PEP

2) The Context Handler (including the PIP)

3) The GeoPDP

The PEP is implemented as an Apache 2 Web Server configured as a Reverse Proxy. As such, it intercepts HTTP requests for a given URI (e.g. /service/WFS) and forwards the request to the appropriate Apache 2 Module.

The Context Handler is implemented as an Apache 2 Module which is loaded at Apache startup and executed if the Apache intercepts a WFS request on a given URI. In correspondence with the XACML information flow, the Context Handler creates the XACML Authorization Decision Request which is sent to the GeoPDP.

The GeoPDP is a Web Service that returns XACML Authorization Decision(s) upon an XACML Authorization Decision Request.

10.1.1 PEP

The Policy Enforcement Point basically is an Apache 2 Web Server configured to run as a Reverse Proxy. The following configuration snippet illustrates this:

```
<Location /service/WFS/Comsoft>
        Order            deny, allow
        Allow            from all
        // Reverse Proxy Settings
        ProxyPass        http://.../cadas-aimdb/wfs
</location>
```

Listing 17: Reverse Proxy configuration for the Comsoft WFS-T

In order to forward intercepted requests to the Context Handler, implemented as an Apache 2 Module, the module must be loaded and activated for the given URI. Loading can be achieved by the LoadModule directive:

```
LoadModule authz_wfs_module .../apache/modules/mod_authz_wfs.so
```

Listing 18: Loading the WFS-T Context Handler

Activation for a particular WFS URI can be achieved by the following lines

```
<Location /service/WFS/Comsoft>
        ...
        // Context Handler activation
```

```
        OWSType          WFS
        GeoPDP           on
        GeoPDPURL        http://.../GeoPDP/service/OWS-8
</Location>
```

Listing 19: Configuration for activating the Context Handler

10.1.2 Context Handler

The Context Handler implements the duties as described in the XACML specification. Therefore, the main task is to interpret the intercepted WFS request and create an XACML conformant Authorization Decision Request.

For a GET request, it analysis the HTTP query string and creates an XACML AttributeValue representation. For a POST request, it inserts the POSTed request into the XACML ADR under the <ResourceContent> element. How this is done in detail is described in the XACML v2.0 OWS Profile specification within the following Requirement Classes:

Requirements Class(es)
{&xop;/RC/1.2, &xop;/RC/1.3(&WFS: 2.0;), &xop;/RC/1.4(&WFS:2.0;), &xop;/RC/1.9(&WFS:2.0;), &xop;/RC/1.11(&WFS:2.0;) }

Listing 20: Requirements Classes used by the Context Handler to construct the XACML ADR

For any Authorization Decision that indicates “Missing Attributes” the Context Handler instructs the PIP to fetch them. For OWS-8, the Context Handler and the PIP are an instance for AIXM and as such understand to resolve “Missing Attributes” for “aixm:controllingAgency” and “aixm:usingAgency”.

The missing attribute “aixm:controllingAgency” in the context of scheduling airspaces will trigger a series of WFS requests to obtain the instances of aixm:Unit that represent the controlling agency for the airspace to be scheduled. In a similar fashion, the aixm:Unit representing the using agency of the airspace to be scheduled is fetched from the WFS.

The logic, how to actually fetch the AIXM features from the protected WFS is implemented in the PIP.

10.1.3 PIP

The Policy Information Point (PIP) for OWS-8 implements the logic how to fetch the AIXM features representing the controlling and using agency for a scheduled airspace. Based on the gml:identifier for the airspace to be scheduled, the PIP first fetches the aixm:AirTrafficControlService instance that is responsible for the airspace. Because the forward references from the Airspace instance to the AirTrafficControlService are not part of the standard AIXM model (only available as an optional extension), the PIP requests the AirTrafficControlService which holds as (backward) references the airspace.

Next, the PIP fetches the Unit representing the controlling and using agency. Finally, the PIP fetches the BASELINE of the airspace to be scheduled. This is required as the airspace scheduling request must not include he geometry of the airspace, but for the geographic access rights, that geometry must be present.

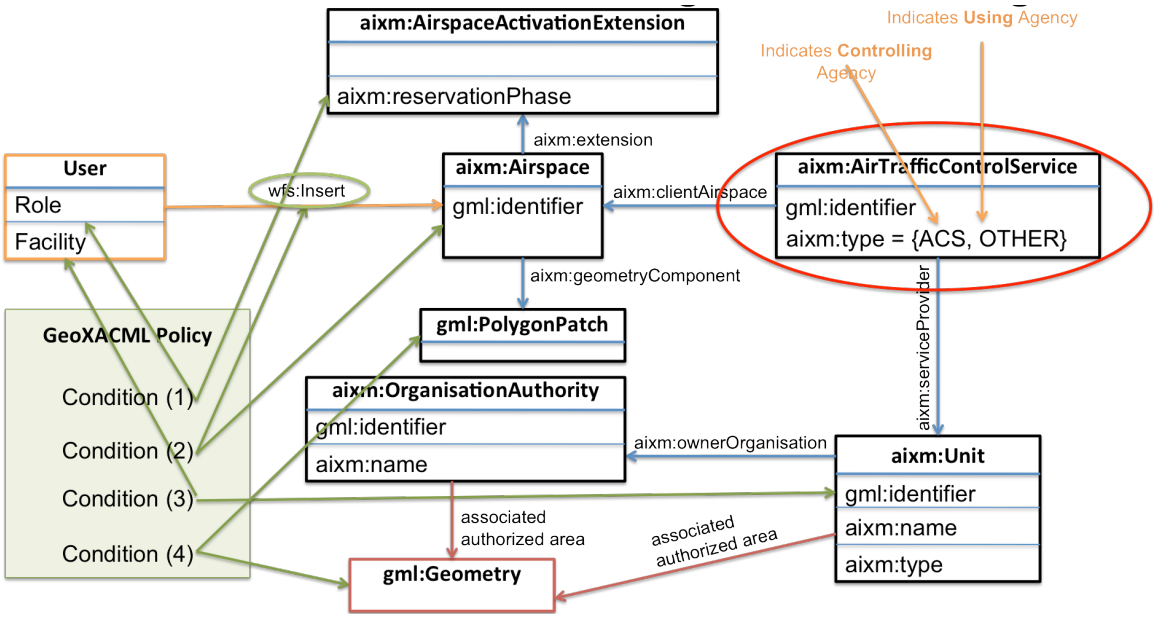

Figure 21: Information Linking for SAA Scheduling

The figure above shows all AIXM features that are involved in deriving an authoritation decision for SAA Scheduling. All AIXM features – fetched in sequential requests to the WFS - are composed in an XML document that is returned to the Context Handler so that an extended ADR can be issued to the GeoPDP. The sequence of requests is illustrated in figures 18, 19 and 20.

10.1.4 PDP

The PDP involved in OWS-8 is a GeoPDP implementing GeoXACML v1.0 BASIC, including extensions A+B. It is therefore possible to use GML 2 and GML 3 geometry encodings and topological test functions, as described by the specification.

10.1.5 Obligation and Error Hander

The Obligation Handler is part of the Context Handler that is capable of modifying a WFS and WFS-T request and response based on Obligations returned by the Authorization Decision.

The ability to modify an intercepted WFS-T request implements the Opaque Security Option where the use is not aware of the fact that something was changed.

In cases where the Authorization Decision is DENY and no obligations are present, the Error Handler returns the appropriate WFS Exception.

Please note that at present, the WFS specification does not define any security specific error codes. Therefore, the used error code 403 and the text "Not Authorized" is not compliant. Perhaps a change request to the OWS Common specification is a way forward to add security specific error codes (cp. 11.2.2)?

10.1.6 Demo-Client

Two demo clients have been implemented that show various access restrictions for the Authoritative Data Store.

The Comsoft demo page illustrates access constraints for fetching AIXM features with different interpretations: BASELINE, TEMPDELTA and PERMDELTA. Further more, it is illustrated how to distinguish between a Commissioning and a Decommissioning and implement access constraints regarding the separation of duty: One user can do a Commissioning but the same use cannot do Decommissioning and vice versa.

The Snowflake demo page illustrates the use cases created by the FAA concerning the scheduling of airspaces. In particular, the duties of controlling and using agencies for PENDING and APPROVED scheduling requests are taken into account here.

More information on the demo client and the evaluation of the security infrastructure can be found in section 9 or this ER.

Please note that both demo pages create AIXM features in the WFS using the Transaction/Insert operation and that therefore all permitted actions can take quite long.

11 Summary and Outlook

11.1 Summary

This engineering report outlines how to provide access control for WFS-T 2.0 instances used to query and manage AIXM data. We proved interoperability as the same software components are used to protect a WFS-T implementation from Comsoft and from Snowflake. The former WFS-T hosts Estonian data and requires to use a SOAP binding and the latter hosts FAA data and requires XML POST binding.

We started with an introduction of the most popular conceptual rights models. Afterwards we summarized the requirements towards access control systems for these infrastructures and finally analyzed the presented conceptual rights models. We were able to conclude that a suitable access control systems for OWS based SDIs must use a hybrid rights model that combines the concepts of rule- and role-based rights models.

One possible implementation of the needed rights model is defined in the latest XACML and GeoXACML specification and the XACML related profiles. Based on these specifications one can implement powerful and standardised access control systems that not only protect WFS-T 2.0 instances but also any Geo Web Services and spatial data found in SDIs.

We demonstrated how to generate XACML ADRs based on intercepted WFS messages and showed how to implement the required types of rights. After the demonstration how to use XACML in the OWS use case we address the implementation of the sponsors business rules in the Aviation/AIXM scenario. The developed and deployed access control system and the formal definition of the rights can be tested online under http://grid01.informatik.unibw-muenchen.de/comsoft.php and under http://grid01.informatik.unibw-muenchen.de/snowflake.php.

11.2 Future Work items

11.2.1 Standardisation of the XACML v3.0 OWS Profile

There are many things that need to be taken into account when using (Geo)XACML to protect OWS based architectures. Thanks to the developed XACML OWS profile there is a set of guidelines how to use GeoXACML to protect OGC Web Services. These rules provide enhanced interoperability in GeoXACML based access control systems for OWS and support an easier applicability and implementation of XACML or GeoXACML based access control systems in OWS environments.

One of the next steps of the GeoXACML SWG has to be the continuation of the standardization process of the XACML v3.0 and v2.0 OWS profile and its service specific extension documents.

11.2.2 Returning Access Control Process Information to the User and Binding Security Related Information to the Request

Another issue that needs to be addressed is how security related information, like a simple access denied message or the notification that the OWS response was filtered because of insufficient permissions is returned in a standardized way to the requestor. The conceptual problem is how to bind security responses form different security services (e.g. the access control system) to the actual OWS response. Should the information be included inside the OWS response itself? Should there be a SecurityServiceReport next to the actual OWS response? If so what would be its content and how to bind the two information entities together? Further research and standardisation effort is needed in this direction.

The problem how to bind access control process results or security information in general to the actual OWS response is closely related to the problem of how to bind security information to an OWS **request**. A general solution for the problem of binding security process results to an OWS response should ideally also be applicable to bind security information to an OWS request.

11.2.3 Interplay of the Access Control System with the validation service

Within the OWS-8 project there was another work package that addressed the issue of how to define and enforce the required validation rules for incoming and outgoing AIXM WFS messages. Although these work items were addressed by other OWS project members we briefly want to document in this engineering report the relation and implications of our work to the related work package.

In general the test whether the exchanged messages are valid according to a specific schema can happen at three different locations:

Client side validation

In order to perform the validation on the client side one needs a schema validation engine build into or callable from the client application.

Server side validation

Alternatively or in parallel one can check the validity of the exchanged OWS messages at the server machine. In this case the OWS implementation needs to test the conformance to the associated schema document(s).

Note that in cases where the schema documents are written in different languages (e.g. in XML Schema, RelaxNG and Schematron) one could e.g. use a wrapping XML schema that integrates the other schema documents through the <xsd:appinfo> element mechanism (cp. e.g. http://www.xfront.com/ExtendingSchemas.html#Options).

Validation within the access control system

Validation rules have some similarities with access control rules. Like access control rules, validation rules define conditions that must hold for the exchanged XML messages. One central difference however is the fact that validation rules are usually defined independent of the interacting subject and must therefore hold for everybody. Further validation rules do not provide the same concepts as access control rules. It is e.g. not possible to define sophisticated effects, combine effects etc.

Albeit these differences validation rules can be evaluated within the access control system. To support the evaluation of validation rules during the access control process one only needs an additional XACML function:

```
validate(schema-definition, AttributeSelector/Designator(Pointer-to-ADR-
Content-or-Attribute-element)
```

By using this function in an XACML rule one can e.g. test if the incoming request (or a certain part of it) is valid against a specific XML schema document, Schematron rule etc. In an XACML implementation that supports this validate function one could directly use the access control components to initialise the schema validation process. How the validation of the schema documents is realised is hidden and can be realized through a built-in off-the-shelf schema processor engine or one could wrap the validation engine by a WPS interface and call the remote validation service through the standardised wps:excecute method.

Listing 21 shows how such a to-be-standardised validation function could be used within an XACML <Rule>, <Policy> and <PolicySet> element.

```
<Match MatchId="&validate-schema;">
  <AttributeValue DataType="&xsd;">
    <!--xsd that defines the restricted version of the insertable building
features -->
    <xs:schema>....<xs:schema
  </AttributeValue>
  <AttributeSelector Category="message" Path="/Transaction/Insert/Building"
DataType="&xml;" MustBePresent="false"/>
</Match>
```

Listing 21: Demonstration how to check schema validity with XACML policy elements

11.2.4 PAP Web Service

Specifications like XACML, GeoXACML and the XACML v3.0 OGC Web Service profile make it a lot easier to implement powerful access control systems that protect the Geo Web Services and spatial data in spatial data infrastructures (SDIs). The underlying hybrid right model of these systems, that combines rule-, rewrite- and role-based models, guarantees that expressive fine grained access rights can be defined and enforced.

The new challenge that arises when using these access control systems is to provide suitable administration systems that support the sound administration of the emerging complex policies.

One important concept that is needed in order to address the administration problem is a suitable administration model (see e.g. [10], Section 5.2 and 5.3). Further one needs to define an appropriate PAP Web Service interface (see [10], 5.3.3.2). Another related research question is what kind of analysis functions can be provided by such an XACML based PAP Web Service. An important open question in this direction is which of the existing spatial and non-spatial logic calculi must be used in order to build a powerful spatial reasoning engine for GeoXACML policies.

We plan to concentrate future research in these interesting areas and hope that the "administration of (Geo)XACML policies" topic will be pushed forward in future OGC test beds and OWS initiatives.

Appendix A Airspace Legal Definition

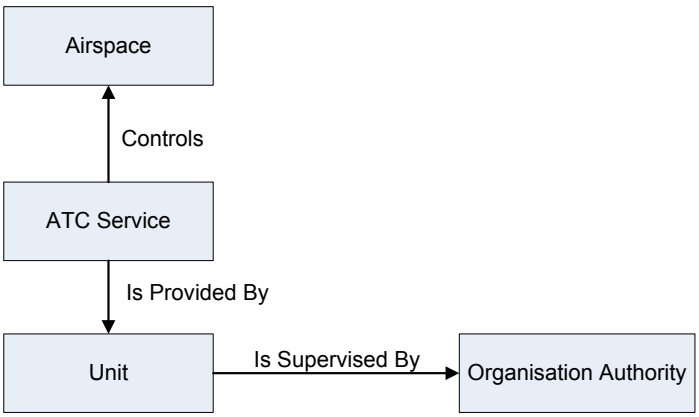

Figure 22: Relationship of relevant AIXM features used in determining Controlling Agency and Using Agency for an airspace.

Airspace #1 Eglin C MOA

Legal Definition:

EGLIN C MOA, FL : Altitudes 1,000 feet AGL to but not including FL 180; occasional use to 200 feet AGL by NOTAM.

Using agency Using agency. U.S. Air Force, Commander, Air Armament Center, Eglin AFB, FL.

Boundaries Beginning at lat. 30°58'01"N., long. 86°25'00"W.; to lat. 30°58'01"N., long. 86°13'00"W.; to lat. 30°43'46"N., long. 86°10'30"W.; to lat. 30°43'31"N., long. 86°20'00"W.; to lat. 30°53'01"N., long. 86°25'00"W.; to the point of beginning.

Controlling agency FAA, Jacksonville ARTCC

Times of use Intermittent, 0600-2100 Monday-Friday; other times by NOTAM.

Controlling agency. FEDERAL AVIATION ADMINISTRATION, Jacksonville ARTCC

Using agency. U.S. Air Force, Commander, Air Armament Center, Eglin AFB, FL.

AIXM Representation:

Airspace:

AIXM Object or Feature	AIXM Element	value
Airspace (Feature)		
	name	EGLIN C MOA, FL
	GML identifier	50f046a4-de5f-48c4-b5d3-d36e8fe29428

Controlling Agency:

AIXM Object or Feature	AIXM Element	value
Air Traffic Control Service (Feature)		
	type	**ACS**
	GML identifier	79bbff5e-14d2-45ee-b659-391319480b9e
Unit (Feature)		
	name	FAA, JACKSONVILLE ARTCC
	Type	ARTCC
	GML identifier	82A95872-9182-2362-E044-00212803DA06
Organization Authority (Feature)		

	Name	FEDERAL AVIATION ADMINISTRATION
	GML Identifier	82A95872-8EC5-2362-E044-00212803DA06

Using Agency:

AIXM Object or Feature	AIXM Element	value
Air Traffic Control Service (Feature)		
	type	**OTHER**
	GML identifier	9d3312b7-d938-4ea1-9191-28465240c2d8
Unit (Feature)		
	name	USAF, AIR ARMAMENT CENTER, EGLIN AFB
	type	MILOPS
	GML identifier	7ec1ce21-6ff2-497d-863c-156e060165e1
Organization Authority (Feature)		
	Name	UNITED STATES AIR FORCE
	GML Identifier	b742c5fd-ab69-4044-a07d-e6ca0caf6886

Airspace #2 R-5001B Fort Dix, NJ

Legal Definition:

R-5001B FORT DIX, NJ : Time of designation Continuous, sunrise Friday to sunset Sunday, other times by NOTAM, 48 hours in advance. Designated altitudes From 4,000 feet MSL to and including 8,000 feet MSL. **Using agency Using agency. Commanding General, Fort Dix, NJ**. Boundaries Beginning at lat. 40?02'45"N., long. 74?26'59"W.; to lat. 40?00'00"N., long. 74?26'19"W.; to lat. 39?59'00"N., long. 74?25'07"W.; to lat. 39?58'00"N., long. 74?24'59"W.; to lat. 39?57'30"N., long. 74?25'16"W.; to lat. 39?57'23"N., long. 74?25'49"W.; to lat. 39?58'45"N., long. 74?27'59"W.; to lat. 39?58'45"N., long. 74?31'24"W.; to lat. 40?01'53"N., long. 74?33'29"W.; to lat. 40?02'45"N., long. 74?32'29"W.; to the point of beginning. **Controlling agency FAA, New York ARTCC.**

AIXM Representation:

Airspace:

AIXM Object or Feature	AIXM Element	value
Airspace (Feature)		
	name	R-5001B FORT DIX, NJ
	GML identifier	d93082ee-67df-44a3-9114-2b3d660116e2

Controlling Agency:

AIXM Object or Feature	AIXM Element	value
Air Traffic Control Service (Feature)		
	type	**ACS**
	GML identifier	e307e809-df13-4e8e-9216-79bbde97fa81
Unit (Feature)		

	name	FAA, NEW YORK ARTCC
	Type	ARTCC
	GML identifier	82A95872-918C-2362-E044-00212803DA06
Organization Authority (Feature)		
	Name	FEDERAL AVIATION ADMINISTRATION
	GML Identifier	82A95872-8EC5-2362-E044-00212803DA06

Using Agency:

AIXM Object or Feature	AIXM Element	value
Air Traffic Control Service (Feature)		
	type	**OTHER**
	GML identifier	c443c5cc-26f3-4ac8-96ea-fc97d15efcaf
Unit (Feature)		
	name	FORT DIX
	type	MILOPS
	GML identifier	c5aecf30-c270-48c2-9a9d-54a44a8adb21
Organization Authority (Feature)		
	Name	UNITED STATES ARMY
	GML Identifier	db8067e6-eae4-45b2-94e2-

		b2e8442d5dd6

Airspace #3 Mayport High MOA

Legal Definition:

MAYPORT HIGH MOA, FL : Altitudes 3,000 feet MSL to but not including FL 180. **Using agency Using agency. U.S. Navy, Fleet Area Control and Surveillance Facility, NAS Jacksonville, FL.** Boundaries Beginning at lat. 30?32'01"N., long. 81?21'14"W.; to lat. 30?29'50"N., long. 81?20'58"W.; thence clockwise via an 8.5-statute-mile radius arc of NAS Mayport (centered at lat. 30?23'31"N., long. 81?25'25"W.; to lat. 30?19'24"N., long. 81?18'20"W.; to lat. 30?21'21"N., long. 81?25'39"W.; to lat. 30?22'11"N., long. 81?26'29"W.; to lat. 30?27'01"N., long. 81?26'14"W.; to the point of beginning. **Controlling agency FAA, Jacksonville ARTCC**. Times of use Intermittent by NOTAM, 1800-2200, not to exceed 8 one-hour block times per month.

AIXM Representation:

Airspace:

AIXM Object or Feature	AIXM Element	value
Airspace (Feature)		
	name	MAYPORT HIGH MOA, FL
	GML identifier	a9998e16-6a03-4cbf-9c1b-e425d19c832c

Controlling Agency:

AIXM Object or Feature	AIXM Element	value

Air Traffic Control Service (Feature)		
	type	**ACS**
	GML identifier	f9a9f273-bfc2-46ea-8800-f4e95a44b270
Unit (Feature)		
	name	FAA, JACKSONVILLE ARTCC
	Type	ARTCC
	GML identifier	82A95872-9182-2362-E044-00212803DA06
Organization Authority (Feature)		
	Name	FEDERAL AVIATION ADMINISTRATION
	GML Identifier	82A95872-8EC5-2362-E044-00212803DA06

Using Agency:

AIXM Object or Feature	AIXM Element	value
Air Traffic Control Service (Feature)		
	type	**OTHER**
	GML identifier	51462d77-4d8a-45fa-ac57-b6b9dd5f48e8
Unit (Feature)		
	name	FLEET AREA CONTROL AND SURVEILLANCE FACILITY

		JACKSONVILLE
	type	MILOPS
	GML identifier	1cccc350-cd5c-4955-a8db-9631c9f58f8f
Organization Authority (Feature)		
	Name	UNITED STATES NAVY
	GML Identifier	e90d5789-146f-4bb4-bb2c-798337379772

Appendix B Permissions to query SAA Definition and Schedule Elements

Appendix B.1 SAA Legal Definition Elements

AIXM Object or Feature	*AIXM Element*	*Permitted Query Elements for Air Traffic Controller and Military Operations Specialist*	*Permitted Query Elements for General Internet User*
General SAA Information			
Airspace (Feature)		Yes	Yes
	name	Yes	Yes
	designator	Yes	Yes
	type	Yes	Yes
	administrativeArea	Yes	Yes
Boundaries			
AirspaceGeometryComponent		Yes	Yes
	operation	Yes	Yes
	operationSequence	Yes	Yes
	theAirspaceVolume	Yes	Yes
AirspaceVolume		Yes	Yes
	horizontalProjection	Yes	Yes
Curve		Yes	Yes
GeoBorder (Feature)		Yes	Yes
	name	Yes	Yes
	type	Yes	Yes
	border	Yes	Yes
Surface		Yes	Yes
AirspaceVolume		Yes	Yes
	upperLimit	Yes	Yes
	upperLimitReference	Yes	Yes
	maximumLimit	Yes	Yes
	maximumLimitReference	Yes	Yes
	lowerLimit	Yes	Yes
	lowerLimitReference	Yes	Yes
	minimumLimit	Yes	Yes
	minimumLimitReference	Yes	Yes
	upperLimitInclusive	Yes	Yes
	maximumLimitInclusive	Yes	Yes
	lowerLimitInclusive	Yes	Yes
	minimumLimitInclusive	Yes	Yes
Airspace (Feature)		Yes	Yes
	timeAhead	Yes	Yes
AirspaceActivation		Yes	Yes
	status	Yes	Yes
	issueNotam	Yes	Yes
	timeInterval	Yes	Yes
TimeInAdvance		Yes	Yes
	type	Yes	Yes
	timeInAdvance	Yes	Yes
Timesheet		Yes	Yes
	timeReference	Yes	Yes

AIXM Object or Feature	*AIXM Element*	*Permitted Query Elements for Air Traffic Controller and Military Operations Specialist*	*Permitted Query Elements for General Internet User*
	startDate	Yes	Yes
	endDate	Yes	Yes
	day	Yes	Yes
	dayTil	Yes	Yes
	startTime	Yes	Yes
	startEvent	Yes	Yes
	startTimeRelativeEvent	Yes	Yes
	startEventInterpretation	Yes	Yes
	endTime	Yes	Yes
	endEvent	Yes	Yes
	endTimeRelativeEvent	Yes	Yes
	endEventInterpretation	Yes	Yes
	daylightSavingAdjust	Yes	Yes
	excluded	Yes	Yes
AirTrafficControlService (Feature)		Yes	Yes
	type	Yes	Yes
	clientAirspace	Yes	Yes
	serviceProvider	Yes	Yes
Unit (Feature)		Yes	Yes
	name	Yes	Yes
	type	Yes	Yes
	designator	Yes	Yes
	military	Yes	Yes
	ownerOrganisation	Yes	Yes
OrganisationAuthority (Feature)		Yes	Yes
	name	Yes	Yes
	designator	Yes	Yes
AirTrafficControlService (Feature)		Yes	Yes
	type	Yes	Yes
	clientAirspace	Yes	Yes
	serviceProvider	Yes	Yes
Unit (Feature)		Yes	Yes
	name	Yes	Yes
	type	Yes	Yes
	designator	Yes	Yes
	military	Yes	Yes

AIXM Object or Feature	*AIXM Element*	*Permitted Query Elements for Air Traffic Controller and Military Operations Specialist*	*Permitted Query Elements for General Internet User*
	ownerOrganisation	Yes	Yes
OrganisationAuthority (Feature)		Yes	Yes
	name	Yes	Yes
	designator	Yes	Yes

Table 6: SAA Legal Definition Elements

Appendix B.2 SAA Schedule Request Elements

AIXM Object or Feature	*AIXM Element*	*Permitted Query Elements for Air Traffic Controller and Military Operations Specialist*	*Permitted Query Elements for General Internet User*
AircraftDetail		Yes	No
	callSign	Yes	No
AircraftGroup		Yes	No
	aircraftType	Yes	No
	flyingUnit	Yes	No
	indicatedAirSpeed	Yes	No
	aircraft	Yes	No
Airspace (Feature)			
	name	Yes	Yes
	designator	Yes	Yes
	type	Yes	Yes
	activation	Yes	Yes
AirspaceActivation			
	activity	Yes	Yes
	status	Yes	Yes
	levels	Yes	Yes
	creationDate	Yes	Yes
	approvalDate	Yes	Yes
	lastModifiedDate	Yes	No
	issueNotam	Yes	Yes
	creator	Yes	No
	approver	Yes	No
	lastModifier	Yes	No
	reservationID	Yes	No
	reservationPhase	Yes	No
	reservationUserAction	Yes	No
	liveFire	Yes	No
	lightsOut	Yes	No
	sorties	Yes	No
	separationStandard	Yes	No
	aircraftInvolved	Yes	No
AirspaceLayer			
	upperLimit	Yes	Yes
	upperLimitReference	Yes	Yes
	lowerLimit	Yes	Yes
	lowerLimitReference	Yes	Yes

AIXM Object or Feature	*AIXM Element*	*Permitted Query Elements for Air Traffic Controller and Military Operations Specialist*	*Permitted Query Elements for General Internet User*
	altitudeInterpretation	Yes	Yes
	upperLimitInclusive	Yes	Yes
	lowerLimitInclusive	Yes	Yes
ContactInformation			
	name	Yes	No
	title	Yes	No
	networkNode	Yes	No
	phoneFax	Yes	No
OnlineContact			
	eMail	Yes	No
TelephoneContact			
	voice	Yes	No
	facsimile	Yes	No

Table 7: SAA Schedule Request Elements

Appendix C GeoXACML encoded policy for the OWS-8 scenario

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  PolicySetId="OWS-8-Authoritative-Data-Store-Level-1"
  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os access_control-xacml-2.0-policy-
schema-os.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Description>This policy set applies to all services provided at the domain
    http://grid01.informatik.unibw-muenchen.de The contained policy sets and policies are a
    framework for plugin of runtime specific policies for enforcing access constraints as defined by
    the FAA as level 1 to 5 and 6. </Description>
  <Target/>
  <Policy PolicyId="Level_1"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description>This is the Level 1 Policy acting for service /service/WFS/OWS-8 as a framework for
      plugin of Rules that apply to users, group of users or rules.</Description>
    <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
    </PolicyDefaults>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
              >/service/WFS/Comsoft</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:uri"
              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
              >/service/WFS/Snowflake</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:uri"
              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule RuleId="Alice" Effect="Deny">
      <Description>Alice cannot access the service</Description>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Alice</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>
          </Subject>
        </Subjects>
      </Target>
    </Rule>
    <Rule RuleId="Andreas" Effect="Permit">
      <Description>Andreas can do all operations :-)</Description>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

```
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Andreas</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>
          </Subject>
        </Subjects>
      </Target>
    </Rule>
    <Rule RuleId="Bob" Effect="Permit">
      <Description>Bob can access the service using the non-transactional requests</Description>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Bob</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>
          </Subject>
        </Subjects>
        <Resources>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >GetCapabilities</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetCapabilities'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >DescribeFeatureType</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='DescribeFeatureType'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >GetFeature</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
```

```
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetFeature'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >GetPropertyValue</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetPropertyValue'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Transaction</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                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"
                >Insert</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:operation"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='Insert'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      </Target>
    </Rule>
    <Rule RuleId="Joe" Effect="Permit">
      <Description>Joe can request BASELINE and PERMDELTA information for feature type
        aixm:Airspace.</Description>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Joe</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>
```

```
          </Subject>
        </Subjects>
        <Resources>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >GetFeature</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetFeature'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      </Target>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#string"
                RequestContextPath="//*[local-name()='Query' and
@typeNames='aixm:Airspace']//*[local-name()='PropertyIsEqualTo']//*[local-name()='ValueReference'
and text()='aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:interpretation']/..//*[local-
name()='Literal']/text()"
                />
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >BASELINE</AttributeValue>
          </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#string"
                RequestContextPath="//*[local-name()='Query' and
@typeNames='aixm:Airspace']//*[local-name()='PropertyIsEqualTo']//*[local-name()='ValueReference'
and text()='aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:interpretation']/..//*[local-
name()='Literal']/text()"
                />
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >PERMDELTA</AttributeValue>
          </Apply>
        </Apply>
      </Condition>
    </Rule>
    <Rule RuleId="Joe" Effect="Deny">
      <Description>Joe cannot request corrections for the feature type aixm:Airspace.</Description>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >Joe</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>
          </Subject>
        </Subjects>
        <Resources>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

```
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
                >GetFeature</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
              <AttributeSelector
                RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetFeature'])"
                DataType="http://www.w3.org/2001/XMLSchema#integer"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      </Target>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#string"
                RequestContextPath="//*[local-name()='Query' and
@typeNames='aixm:Airspace']//*[local-name()='PropertyIsEqualTo']//*[local-name()='ValueReference'
and text()='aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:interpretation']/..//*[local-
name()='Literal']/text()"
                />
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >PERMDELTA</AttributeValue>
          </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-greater-than">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#integer"
                RequestContextPath="count(//*[local-name()='Query' and
@typeNames='aixm:Airspace']//*[local-name()='PropertyIsEqualTo']//*[local-name()='ValueReference'
and text()='aixm:timeSlice/aixm:AirspaceTimeSlice/aixm:correctionNumber'])"
                />
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
          </Apply>
        </Apply>
      </Condition>
    </Rule>
  </Policy>
  <Policy PolicyId="Jane"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description>Jane can request features of type aixm:RunwayCentrelinePoint within 50km of Tartu,
      Estonia (26.714828 58.370884).</Description>
    <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
    </PolicyDefaults>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Jane</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>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
```

```
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
              >/service/WFS/Comsoft</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:uri"
              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >GetFeature</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeSelector
              RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='GetFeature'])"
              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule RuleId="aixm:ElevatedPoint" Effect="Permit">
      <Target/>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-greater-than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
            <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#integer"
              RequestContextPath="count(//*[local-name()='Query' and
@typeNames='aixm:RunwayCentrelinePoint'])"
              />
          </Apply>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">0</AttributeValue>
        </Apply>
      </Condition>
    </Rule>
    <Obligations>
      <Obligation ObligationId="urn:ogc:filter:FILTER" FulfillOn="Permit">
        <AttributeAssignment AttributeId="Tartu_50km_radius"
          DataType="http://www.w3.org/2001/XMLSchema#string">&lt;fes:Filter

xmlns:fes=&quot;http://www.opengis.net/fes/2.0&quot;&gt;&lt;fes:BBOX&gt;&lt;fes:ValueReference&gt;ai
xm:ElevatedPoint&lt;/fes:ValueReference&gt;&lt;gml:Envelope
          xmlns:gml=&quot;http://www.opengis.net/gml/3.2&quot;
          srsName=&quot;urn:ogc:def:crs:OGC:1.3:CRS84&quot;&gt;&lt;gml:lowerCorner&gt;26.247193
          58.144504&lt;/gml:lowerCorner&gt;&lt;gml:upperCorner&gt;27.16196

58.66467&lt;/gml:upperCorner&gt;&lt;/gml:Envelope&gt;&lt;/fes:BBOX&gt;&lt;/fes:Filter&gt;</Attribute
Assignment>
      </Obligation>
    </Obligations>
  </Policy>
  <Policy PolicyId="Dave"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description>Dave can do commissioning for feature type aixm:RadarSystem.</Description>
    <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
    </PolicyDefaults>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Dave</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>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
              >/service/WFS/Comsoft</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:uri"
              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >Transaction</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
              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"
              >Insert</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:operation"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeSelector
              RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='Insert'])"
              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule RuleId="aixm:RadarSystem:Decommissioning" Effect="Permit">
      <Target/>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-and-only">
              <AttributeSelector
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'validTime']//*[local-name() = 'beginPosition']"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-and-only">
              <AttributeSelector
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'featureLifetime']//*[local-name() = 'beginPosition']"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
            </Apply>
          </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#boolean"
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'featureLifetime']//*[local-name() = 'endPosition'] = ''"/>
            </Apply>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">true</AttributeValue>
          </Apply>
```

```
        </Apply>
      </Condition>
    </Rule>
  </Policy>
  <Policy PolicyId="Claris"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description>Claris can do decommissioning for feature type aixm:RadarSystem.</Description>
    <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
    </PolicyDefaults>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Claris</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>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI"
              >/service/WFS/Comsoft</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:uri"
              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"
              >Transaction</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:request"
              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"
              >Insert</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:ogc:def:profile:xacml:2.0:operation"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeSelector
              RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='Insert'])"
              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule RuleId="aixm:RadarSystem:Commissioning" Effect="Permit">
      <Target/>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:dateTime-one-and-only">
              <AttributeSelector
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'validTime']//*[local-name() = 'beginPosition']"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
```

```
              <AttributeSelector
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'featureLifetime']//*[local-name() = 'endPosition']"
                DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
            </Apply>
          </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-one-and-only">
              <AttributeSelector DataType="http://www.w3.org/2001/XMLSchema#boolean"
                RequestContextPath="//*[local-name() = 'RadarSystem']//*[local-name() =
'featureLifetime']//*[local-name() = 'endPosition'] = ''"/>
            </Apply>
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#boolean">false</AttributeValue>
          </Apply>
        </Apply>
      </Condition>
    </Rule>
  </Policy>
  <PolicySet PolicySetId="SAA_Scheduling" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:first-applicable">
    <Description>This PolicySet declares the FAA example business rules concerning SAA Scheduling as
described in http://portal.opengeospatial.org/files/?artifact_id=44831</Description>
    <PolicySetDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
    </PolicySetDefaults>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeSelector
              RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='Insert'])"
              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
            <AttributeSelector
              RequestContextPath="count(xacml-context:Request/xacml-context:Resource/xacml-
context:ResourceContent//*[local-name()='AirspaceActivationExtension']//*[local-
name()='reservationUserAction'])"
              DataType="http://www.w3.org/2001/XMLSchema#integer"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Policy PolicyId="MOS" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
      <Description>This Policy declares the Role Military Operations Specialist
permissions</Description>
      <PolicyDefaults>
        <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
      </PolicyDefaults>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">MOS</AttributeValue>
              <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </SubjectMatch>
          </Subject>
        </Subjects>
        <Resources>
```

```
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"></AttributeValue>
              <ResourceAttributeDesignator AttributeId="ControllingAgency"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </ResourceMatch>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"></AttributeValue>
              <ResourceAttributeDesignator AttributeId="UsingAgency"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      </Target>
      <Rule RuleId="CREATE_PENDING" Effect="Permit">
        <Target>
          <Resources>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PENDING</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">CREATE</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
          </Resources>
        </Target>
        <Condition>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
              <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_UUID"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
              <!-- controlling agency-->
              <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='ACS']/..//*[local-
name()='Unit']/*[local-name()='identifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
              <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_UUID"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
              <!-- using agency-->
              <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='OTHER']/..//*[local-
name()='Unit']/*[local-name()='identifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
          </Apply>
        </Condition>
```

```
      </Rule>
      <Rule RuleId="APPROVED" Effect="Permit">
        <Target>
          <Resources>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">APPROVED</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
          </Resources>
        </Target>
        <Condition>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
            <!-- User belongs to the controlling agency of the airspace -->
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
              <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_UUID"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
              <!-- controlling agency-->
              <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='ACS']/..//*[local-
name()='Unit']/*[local-name()='identifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
            <!-- User facility has authorization area that includes airspace -->
            <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-within">
              <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-one-and-only">
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='ACS']/..//*[local-
name()='Airspace']//*[local-name()='horizontalProjection']/*[local-name()='ElevatedSurface']"
DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry"/>
              </Apply>
              <Apply FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_AUTH_AREA"
DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry"/>
              </Apply>
            </Apply>
          </Apply>
        </Condition>
      </Rule>
    </Policy>
    <Policy PolicyId="ATC" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
      <Description>This Policy declares the Role Air Traffic Controller permissions</Description>
      <PolicyDefaults>
        <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
      </PolicyDefaults>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">ATC</AttributeValue>
              <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </SubjectMatch>
          </Subject>
```

```
        </Subjects>
      </Target>
      <Rule RuleId="allways_deny" Effect="Deny"/>
    </Policy>
    <Policy PolicyId="SAAS" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
      <Description>This Policy declares the Role SAA Scheduler permissions</Description>
      <PolicyDefaults>
        <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
      </PolicyDefaults>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">SAAS</AttributeValue>
              <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </SubjectMatch>
          </Subject>
        </Subjects>
        <Resources>
          <Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"></AttributeValue>
              <ResourceAttributeDesignator AttributeId="ControllingAgency"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </ResourceMatch>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"></AttributeValue>
              <ResourceAttributeDesignator AttributeId="UsingAgency"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      </Target>
      <Rule RuleId="CREATE_PENDING" Effect="Permit">
        <Target>
          <Resources>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">CREATE</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PENDING</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
          </Resources>
        </Target>
        <Condition>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
              <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_UUID"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
```

```
              <!-- controlling agency-->
              <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='ACS']/..//*[local-
name()='Unit']/*[local-name()='identifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
              <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                <SubjectAttributeDesignator AttributeId="USER_FACILITY_UUID"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
              </Apply>
              <!-- using agency-->
              <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirTrafficControlServiceTimeSlice']/*[local-name()='type' and text()='OTHER']/..//*[local-
name()='Unit']/*[local-name()='identifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
          </Apply>
        </Condition>
      </Rule>
      <Rule RuleId="APPROVED" Effect="Deny">
        <Target>
          <Resources>
            <Resource>
              <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">APPROVED</AttributeValue>
                <AttributeSelector RequestContextPath="xacml-context:Request/xacml-
context:Resource/xacml-context:ResourceContent//*[local-
name()='AirspaceActivationExtension']//*[local-name()='reservationUserAction']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
              </ResourceMatch>
            </Resource>
          </Resources>
        </Target>
      </Rule>
    </Policy>
    <Policy PolicyId="GIU" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
      <Description>This Policy declares the Role General Internet User permissions</Description>
      <PolicyDefaults>
        <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
      </PolicyDefaults>
      <Target>
        <Subjects>
          <Subject>
            <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
              <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">GIU</AttributeValue>
              <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </SubjectMatch>
          </Subject>
        </Subjects>
      </Target>
      <Rule RuleId="allways_deny" Effect="Deny"/>
    </Policy>
  </PolicySet>
</PolicySet>
```