Details on how to use the XACML obligation mechanism to achieve a rewriting of intercepted Web Service messages: (this text snippet is part of the latest draft of the XACML v3.0 OGC Web Service profile"

Note: To simplify the formulations in this section, it is assumed that the XACML Obligation Handler is implemented within the XACML Context Handler. This is however not a mandatory prerequisite and all defined guidelines remain valid even if the Obligation Handler is implemented as a separate software component.

Requirement: http://www.opengis.net/doc/IS/X3OP/1.0./R/1.11.2

A conformant XACML Context Handler implementation shall be able to process <xacml:Obligation> elements that have an ObligationId XML attribute equal &rewrite-obligation; as described below. These obligations are called rewrite obligations in the following and shall always at least include exactly one <xacml:AttributeAsssignment> element with an AttributeId XML attribute equal &xslt-rewrite-stylesheet;. The value of its <xacml:AttributeValue> element shall be a valid XSLT v2.0 style sheet as defined in Fehler! Verweisquelle konnte nicht gefunden werden. and its DataType XML attribute value shall equal &xslt;. Consequently the only child elements of <AttributeValue> elements of &xslt-rewrite-stylesheet; <xacml:AttributeAsssignment> elements shall be <xsl:transform> or <xsl:stylesheet> elements respectively. The Category XML attribute of &xslt-rewrite-stylesheet; <xacml:AttributeAsssignment> elements shall equal &obligation;.

The Obligation Handler within an XACML Context Handler shall be able to execute XSLT style sheets, defined by the values of the &xslt-rewrite-stylesheet; <xacml:AttributeAsssignment> elements. The input document for the first XSLT transformation step (details see below) shall be the global or multiple ADR that was generated or received by the XACML Context Handler. The input document for the first transformation step may also be an individual XACML ADR but only if this individual XACML ADR was not derived automatically from a multiple XACML ADR.

To support the passing of parameters to XSLT style sheets, the <xslt:param> element mechanism and the instantiation of suitable xslt-argument <xacml:AttributeAssigment> elements in rewrite obligations is used.

For each to be passed xslt-argument <xacml:AttributeAssignmentExpression> element one needs to define a <xsl:param> element within the XSLT stylesheet. The "AttributeId" XML attribute value of an xslt-argument <xacml:AttributeAssignmentExpression> element shall be identical with the value of the "name" XML attribute value of the corresponding <xsl:param> element. Valid AttributeId and name XML attribute values respectively shall either start with the prefix &init-select-node-argument; or &init-text-node-argument;.

The PDP generates xslt-argument <AttributeAssignment> elements based on the applicable <xacml:AttributeAssignmentExpression> elements. All xslt-argument <AttributeAssignment> elements shall belong to the &obligation; category. A conformant XACML Context Handler shall use the values of the xslt-argument <AttributeAssignment> elements in an XACML authorization decision to initialize the corresponding <xsl:param> elements. The prefix of the AttributeAssignment/@AttributeId values determines how the XACML Context Handler shall initialize the corresponding <xsl:param> element.

In case the AttributeAssignment/@AttributeId value starts with the string &init-select-node-argument;, the XACML Context Handler shall assign the value of the AttributeAssignment/AttributeValue/text() node to the corresponding xsl:param/@select node.

In case the AttributeAssignment/@AttributeId value starts with the string &init-text-node-argument;, the XACML Context Handler shall assign the value of the AttributeAssignment/AttributeValue/text() node to the corresponding xsl:param/text() node.

To access the arguments passed through the described mechanism one can use the $<name-of-xsl-param-element> construct as defined in the XSLT specification (cp. Fehler! Verweisquelle konnte nicht gefunden werden., section 9.2).

An XACML authorization decision response may contain multiple &rewrite-obligation; <xacml:Obligation> elements, each containing a different &xslt-rewrite-stylesheet; <xacml:Attribute> definition and optionally &obligation-priority; <xacml:Attribute> element definitions of type integer and category &obligation;. A conformant XACML Context Handler shall process these XACML authorization decision responses as follows.

The transformations defined in the &rewrite-obligation; <xacml:Obligation> elements shall be sorted in descending order, according to the value of &obligation-priority; <xacml:Attribute> element. The absence of an &obligation-priority; <xacml:Attribute> element definition in a rewrite <xacml:Obligation> element implies the lowest priority value “0” for the corresponding XSLT stylesheet. The Context Handler shall process the list of XSLT style sheets in descending order. The input document for the first transformation shall be the original global, multiple or individual ADR y. The output of the XSLT processor after applying the first XSLT style sheet is an ADR y’. The y’ document will be the input document for the next transformation step and will be transformed to y’’ etc.

After all applicable XSLT style sheets have been processed, the XACML Context Handler shall extract the OWS specific information from the modified XACML ADR. Therefore the Context Handler first needs to determine whether the OWS message information was included under the <xacml:Content> element of the &message; category or through a set of <xacml:Attribute> elements under the &message; category. This can be inferred based on the values of the &xml-req-mapping;, &kvp-req-mapping; and &xml-resp-mapping; <xacml:Attribute> elements included in the modified XACML ADR (cp. &x3op;/R/1.1.4 in section Fehler! Verweisquelle konnte nicht gefunden werden.). In cases where the Context Handler represents intercepted OWS messages below <xacml:Content> elements as well as through sets of <xacml:Attribute> elements the rewrite obligation shall contain an &representation-to-map; <xacml:AttributeAssignment> element. The value of this element (either &content; or &attribute-set;) uniquely instructs the Context Handler which ADR specific representation of the OWS message shall be used when generating the rewritten version of the OWS message in its originally intercepted encoding.

After identifying the OWS message specific elements in the modified XACML ADR, the Context Handler shall use this information to generate the rewritten OWS message in the encoding as originally intercepted by the PEP. The target encoding for this transformation is deduce-able based on the value of the &original-message-encoding; <xacml:Attribute> element, that is also included in the XACML ADR. The guidelines needed to perform this transformation are defined by the imported requirement classes &x3ope;/ERC/*extension-id*/700 (cp. 6.12.3).

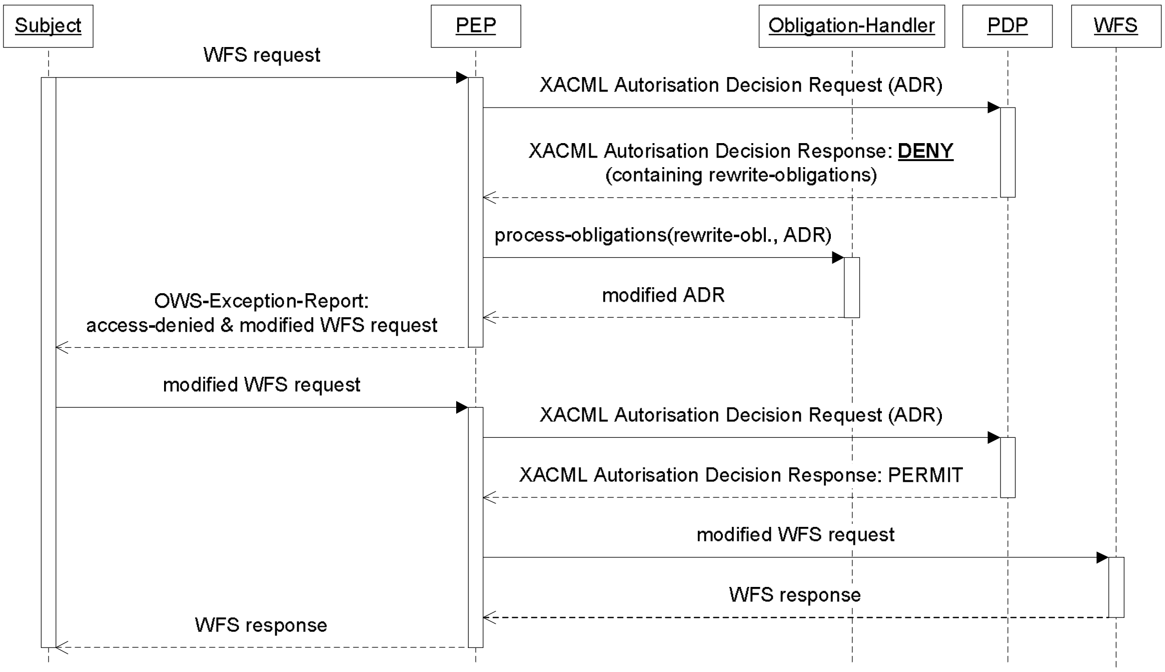
The modified OWS message shall than be sent to the PEP, accompanied with the aggregated permit or deny authorization decision. In case of a deny authorization decision the PEP shall return an “access denied” error message to the interacting access-subject that also contains the rewritten OWS message (transparent rewriting approach – cp. appendix C.1). In case of a permit authorization decision request, the modified OWS message shall be directly forwarded to the intended destination without notifying the recipient-subject about the modification of the originally intercepted message (opaque rewriting approach – cp. appendix C.2).

Note: This requirement class guarantees that the Context Handler understands and discharges all received rewrite obligations correctly The requirements specified in this clause therefore implicitly define how policy writers shall define rewrite obligations.

Note: Rewrite obligations refer to the OWS message representation in the XACML ADR. This greatly simplifies policy administration as the conditions and rewrite expressions defined by the administrators apply to the same message model and encoding which is furthermore independent of the actual encoding of the originally intercepted OWS message.

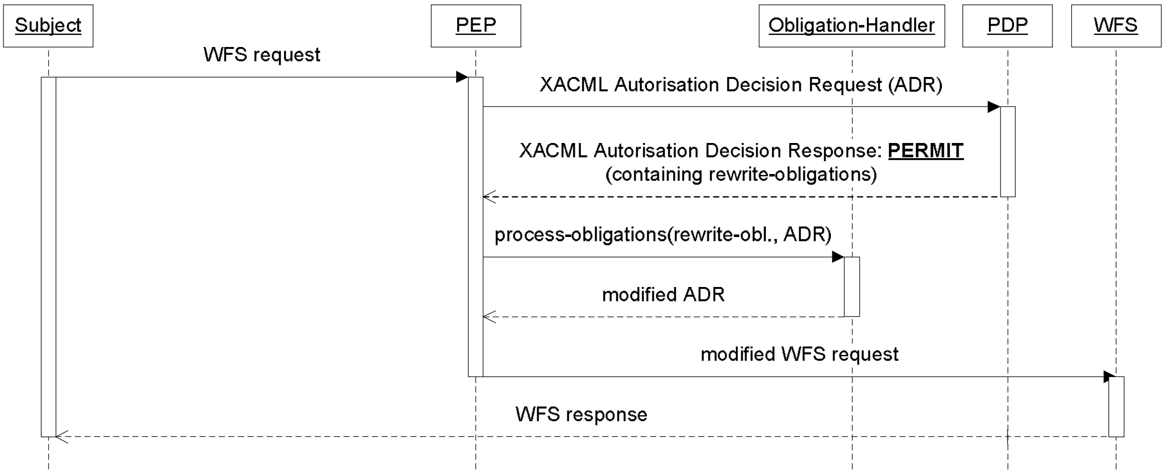
Annex C Sequence Diagrams

C.1 Rewrite Obligations - Transparent Approach



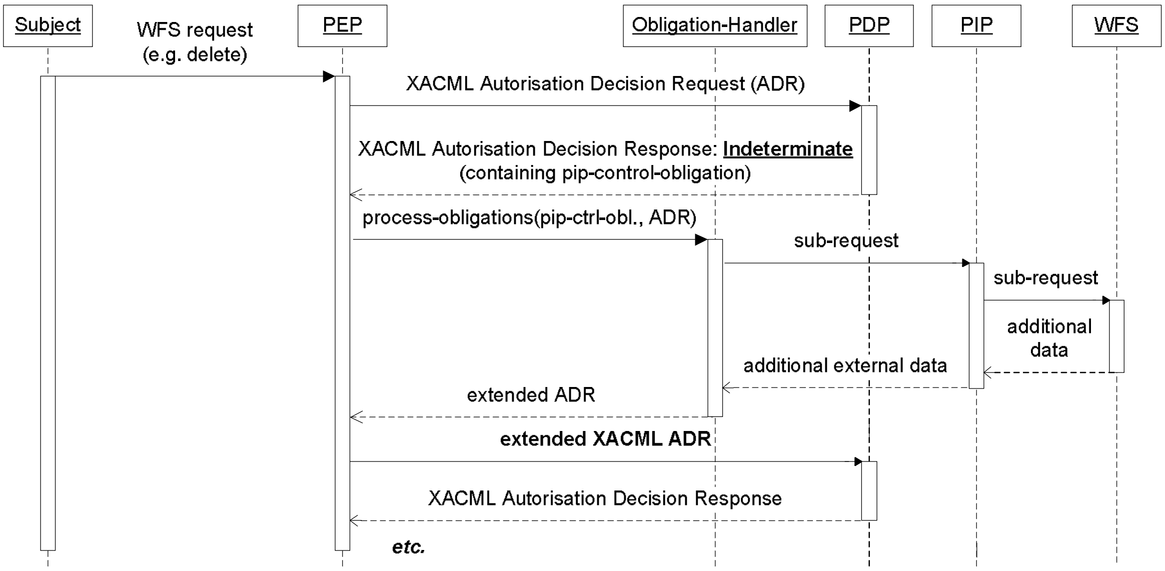
Subject
PEP
Obligation-Handler
PDP
WFS
WFS request
XACML Autorisation Decision Request (ADR)
XACML Autorisation Decision Response: DENY
(containing rewrite-obligations)
process-obligations(rewrite-obl., ADR)
modified ADR
OWS-Exception-Report:
access-denied & modified WFS request
modified WFS request
XACML Autorisation Decision Request (ADR)
XACML Autorisation Decision Response: PERMIT
modified WFS request
WFS response
WFS response

C.2 Rewrite Obligations - Opaque Approach



Subject
PEP
Obligation-Handler
PDP
WFS
WFS request
XACML Autorisation Decision Request (ADR)
XACML Autorisation Decision Response: PERMIT
(containing rewrite-obligations)
process-obligations(rewrite-obl., ADR)
modified ADR
modified WFS request
WFS response

C.3 Extension Obligations


Subject
PEP
Obligation-Handler
PDP
PIP
WFS
WFS request
(e.g. delete)
XACML Autorisation Decision Request (ADR)
XACML Autorisation Decision Response: Indeterminate
(containing pip-control-obligation)
process-obligations(pip-ctrl-obl., ADR)
sub-request
sub-request
additional
data
additional external data
extended ADR
extended XACML ADR
XACML Autorisation Decision Response
etc.

Annex D Example rewrite Rule