



---

# XACML 2.0 RSA 2008 Interop Scenarios Version 0.12

**Working Draft**

**15 April 2008**

**Specification URIs:**

**This Version:**

<http://docs.oasis-open.org/xacml/xacml-2.0-core-RSA2008-interop-draft-12-02.doc>

**Previous Version:**

<http://docs.oasis-open.org/xacml/xacml-2.0-core-RSA2008-interop-draft-12-01.doc>

**Latest Version:**

<http://docs.oasis-open.org/xacml/xacml-2.0-core-RSA2008-interop-draft-12-02.doc>

**Latest Approved Version:**

none

**Technical Committee:**

OASIS XACML TC

**Chair(s):**

Hal Lockhart  
Bill Parducci

**Editor(s):**

Rich Levinson, Oracle Corporation  
Erik Rissanen, Axiomatics  
David Staggs, Department of Veterans Affairs (SAIC)  
Denis Pilipchuk, BEA Systems, Inc.  
Duane DeCouteau, Department of Veterans Affairs (Edmond Scientific Company)  
Dilli Dorai, Sun Microsystems  
Mike Davis, Department of Veterans Affairs

**Related work:**

This specification is related to:

- Extensible Access Control Markup Language (XACML) Version 2.0

**Declared XML Namespace(s):**

[list namespaces here]  
[list namespaces here]

**Abstract:**

This document specifies scenarios that may be used to demonstrate interoperability of multiple PDP, PEP, and PIP modules that were implemented based on the XACML 2.0 Core Specification.

**Status:**

This document was last revised or approved by the OASIS XACML TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/xacml>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/xacml/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/xacml>.

---

## Notices

Copyright © OASIS® 1993–2008. All Rights Reserved. OASIS trademark, IPR and other policies apply.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

---

# Table of Contents

1	Introduction.....	5
1.1	Overview of XACML 2.0 .....	5
1.1.1	Policy Evaluation .....	5
1.1.2	Decision request and response.....	5
1.1.3	Policy Update and Retrieval .....	6
1.2	Terminology .....	6
1.3	Normative References .....	7
1.4	Non-Normative References .....	7
1.5	Interoperability Use Cases for XACML 2.0.....	8
1.5.1	Healthcare Fine Grain Authorization Use Cases .....	12
1.5.2	Use Case: Coarse Grain Authorization Decision Request/Response .....	13
1.5.3	Use Case: Policy Exchange .....	14
2	Use Cases: Healthcare: Fine Grained Authorization .....	15
2.1	Introduction to the Healthcare Application.....	15
2.1.1	Flexibility provided by XACML.....	15
2.1.2	Use of Virtual Roles.....	15
2.1.3	Scenario structure .....	17
2.1.4	Use Case Policy Pseudo-code.....	20
2.2	Detailed Description of Fine Grained Authorization Use Cases.....	23
2.2.1	Details: HL7 Role/Permission.....	24
2.2.2	Details: HL7 Patient Consent Directive .....	37
2.2.3	Details: Attribute based rules .....	44
2.2.4	Details: Emergency access Use Case .....	48
2.2.5	Data filtering Use Case .....	54
A.	Acknowledgements .....	58
B.	Non-Normative Text .....	59
C.	Revision History.....	60

---

# 1 Introduction

The purpose of this document is to present a set of use case scenarios that can be used demonstrate interoperability between products from multiple vendors that contain components that comply with the XACML 2.0 Specification [XACML20].

In this introduction, first an overview of XACML 2.0 will be presented, then a brief description of the use cases will show how interoperability of XACML 2.0 components can be demonstrated within the use cases.

Note: A **hyperlink index of all the messages and policies** used in this document to enable easy navigation within the document may be found here: [[xacml-msg-policy-index](#)]. At the beginning of each message and policy is a link back to the index so that it is easy to go to a message, then if you want to go to another one just go back to the index and pick it.

## 1.1 Overview of XACML 2.0

It is assumed that the reader is familiar with the XACML 2.0 Specification [XACML20], and that the following brief contextual summary will be sufficient to relate the subject matter of this document to the conceptual framework of the XACML 2.0 specification.

The following sections describe what XACML 2.0 policies are and how they are evaluated, how decision requests are submitted for evaluation and results returned, and how policies are made available for evaluation.

In order to distinguish this Interop document from the first XACML Interop document [Interop 01], which will now be referred to as “XACML Interop 01 (Burton 2007)”, this interop will be generically referred to as “**XACML Interop 02 (RSA 2008)**”.

### 1.1.1 Policy Evaluation

The XACML 2.0 Specification defines an XML-oriented **policy** language, which is intended to be used at a Policy Decision Point (**PDP**) to represent the set of **policies** that the **PDP** will use to evaluate **decision requests** received from a Policy Enforcement Point (**PEP**).

**Policies** contain expressions that define dynamic access relationship **conditions** between **subjects** and **resources** based on **attributes** associated with the subject(s) making an access request, **attributes** associated with the **resource(s)** to which access is being requested, **attributes** associated with the **action** intended to be applied to the **resource**, and **attributes** of the operational **environment** (such as time of day).

The **PDP** determines the set of **policies** that are **applicable** to the **request**, evaluates the **applicable policies** by collecting **attribute** information from the **request** and using it where appropriate in the **policy expressions** and returns a **decision**, which may be one of: **permit**, **deny**, **indeterminate**, or **not applicable**.

### 1.1.2 Decision request and response

In addition to the policy language described in the previous section, XACML 2.0 also specifies XML-oriented request and response structures, referred to as contexts, which are used to submit decision requests and to return decision results.

The general functional model is that a PEP will submit a request message to a PDP, which will process the request message, and then return a response message to the PEP. One possible method for packaging up messages for PEP/PDP exchange is described in the SAML 2.0 profile of XACML 2.0 [SAML-XACML20].

45 The request context has many similarities to the main policy language, particularly because the request  
46 must contain the attributes that required by the applicable policies to produce decisions. In fact, one of the  
47 main challenges of interoperability testing is to ensure that the correct set of subject, action, resource,  
48 and environment attributes are collected in the request context, which will be sufficient to enable  
49 evaluation of the applicable policies.

50 The response context contains the decision results, which includes status and details of what steps might  
51 need to be taken to in cases where decisions could not be reached because all the required attributes  
52 were not included in the request. In addition, obligations may be included in the response context that  
53 directs the PEP as to follow-up operations that must be executed.

### 54 **1.1.3 Policy Update and Retrieval**

55 The XACML 2.0 Core Specification [XACML20] does not explicitly address how policies are made  
56 available to the PDP or controlled once they are available to the PDP. However, a XACML 2.0 entity,  
57 referred to as a Policy Administration Point (PAP) is functionally defined as “a system entity that creates a  
58 policy or policy set”. Additional references are contained within the XACML 2.0 Core Specification that  
59 explain the responsibilities of the PAP regarding such topics as composition of policy sets and  
60 maintaining unique identifiers for policies.

61 Two possible mechanism for policy administration between a PAP and PDP are described in the SAML  
62 2.0 profile for XACML 2.0 [SAML-XACML20]. One mechanism is a SAML-based request-response  
63 protocol where the PDP queries the PAP for policies. The other is a simple SAML Assertion-based  
64 storage format, which a PAP may use for placing policies in a generic repository, which may be accessed  
65 directly by the PDP.

## 66 **1.2 Terminology**

67 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD  
68 NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described  
69 in **Error! Reference source not found.**

## 70 1.3 Normative References

- 71 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
72 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 73 [XACML20] T. Moses, *XACML 2.0 Core: eXtensible Access Control Markup Language*  
74 *(XACML) Version 2.0*, [http://docs.oasis-open.org/xacml/2.0/access\\_control-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf)  
75 [xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf), OASIS Standard, 1 February 2005.
- 76 [SAML-XACML20] A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML 2.0 Errata*,  
77 [http://www.oasis-open.org/committees/download.php/15447/xacml-2.0-saml-](http://www.oasis-open.org/committees/download.php/15447/xacml-2.0-saml-errata-wd.zip)  
78 [errata-wd.zip](http://www.oasis-open.org/committees/download.php/15447/xacml-2.0-saml-errata-wd.zip), Working Draft 01, 17 November 2005.
- 79
- 80 [SX20-ASSN-SCH] [access\\_control-xacml-2.0-saml-assertion-schema-os.xsd](http://www.oasis-open.org/committees/download.php/11474/access_control-xacml-2.0-saml-assertion-schema-os.xsd), [http://www.oasis-](http://www.oasis-open.org/committees/download.php/11474/access_control-xacml-2.0-saml-assertion-schema-os.xsd)  
81 [open.org/committees/download.php/11474/access\\_control-xacml-2.0-saml-](http://www.oasis-open.org/committees/download.php/11474/access_control-xacml-2.0-saml-assertion-schema-os.xsd)  
82 [assertion-schema-os.xsd](http://www.oasis-open.org/committees/download.php/11474/access_control-xacml-2.0-saml-assertion-schema-os.xsd)
- 83 [SX20-PROT-SCH] [access\\_control-xacml-2.0-saml-protocol-schema-os.xsd](http://www.oasis-open.org/committees/download.php/11475/access_control-xacml-2.0-saml-protocol-schema-os.xsd), [http://www.oasis-](http://www.oasis-open.org/committees/download.php/11475/access_control-xacml-2.0-saml-protocol-schema-os.xsd)  
84 [open.org/committees/download.php/11475/access\\_control-xacml-2.0-saml-](http://www.oasis-open.org/committees/download.php/11475/access_control-xacml-2.0-saml-protocol-schema-os.xsd)  
85 [protocol-schema-os.xsd](http://www.oasis-open.org/committees/download.php/11475/access_control-xacml-2.0-saml-protocol-schema-os.xsd)
- 86 [HL7-PERM] HL7 Security Technical Committee, HL7 Version 3 Standard: Role-based Access  
87 Control Healthcare Permission Catalog, (Available through  
88 <http://www.hl7.org/library/standards.cfm>), Release 1, Designation: ANSI/HL7 V3  
89 RBAC, R1-2008, Approval Date 2/20/2008.
- 90 [HL7-CONSENT] HL7 Consent Related Vocabulary confidentialityCodes Recommendation,  
91 <http://lists.oasis-open.org/archives/xacml-demo-tech/200712/doc00003.doc>, from  
92 project submission: [http://lists.oasis-open.org/archives/xacml-demo-](http://lists.oasis-open.org/archives/xacml-demo-tech/200712/msg00015.html)  
93 [tech/200712/msg00015.html](http://lists.oasis-open.org/archives/xacml-demo-tech/200712/msg00015.html)

## 94 1.4 Non-Normative References

- 95 [superceded-by-errata] A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML 2.0*, [http://docs.oasis-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf)  
96 [open.org/xacml/2.0/access\\_control-xacml-2.0-saml-profile-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf), OASIS  
97 Standard, 1 February 2005 (original spec, superceded by errata spec).
- 98 [SAML-XACML20V2] A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML Version 2*,  
99 [http://www.oasis-open.org/committees/download.php/24681/xacml-profile-](http://www.oasis-open.org/committees/download.php/24681/xacml-profile-saml2.0-v2-spec-wd-5-en.pdf)  
100 [saml2.0-v2-spec-wd-5-en.pdf](http://www.oasis-open.org/committees/download.php/24681/xacml-profile-saml2.0-v2-spec-wd-5-en.pdf), Working Draft 05, 19 July 2007 (current working  
101 draft covers all versions of XACML).
- 102
- 103 [XACML-RBAC] A. Anderson, *Core and hierarchical role based access control (RBAC)*  
104 *profile of XACML v2.0*, [http://docs.oasis-open.org/xacml/2.0/access\\_control-](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf)  
105 [xacml-2.0-rbac-profile1-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf), OASIS Standard, 1 February 2005.
- 106 [INTEROP-01] R. Levinson, D. Pilipchuk, *XACML 2.0 Interop Scenarios*, [http://www.oasis-](http://www.oasis-open.org/committees/download.php/24475/xacml-2.0-core-interop-draft-12-04.doc)  
107 [open.org/committees/download.php/24475/xacml-2.0-core-interop-draft-12-](http://www.oasis-open.org/committees/download.php/24475/xacml-2.0-core-interop-draft-12-04.doc)  
108 [04.doc](http://www.oasis-open.org/committees/download.php/24475/xacml-2.0-core-interop-draft-12-04.doc), Working Draft, 22 June 2007.
- 109 [HL7-RoleEng] HL7 Security Technical Committee, *HL7 Role Based Access Control (RBAC)*  
110 *Role Engineering Process*, <http://www.hl7.org/library/> (search rbac),  
111 Version 1.3, March 1, 2008.
- 112 [HITSP] Healthcare Information Technology Standards Panel (HITSP) at [www.hitsp.org](http://www.hitsp.org).
- 113
- 114

## 115 **1.5 Interoperability Use Cases for XACML 2.0**

116 This section is a brief introduction to the interoperability use cases that are specified in the remainder of  
117 this document.

118 Unlike the XACML Interop 01, in June 2007, which concentrated on core functionality, such as basic  
119 Authorization Decision Request/Response and Policy Exchange, XACML Interop 02 will focus on a  
120 specific application environment (health care) and concentrate on the development of fine grained  
121 authorization use cases in a Role Based Access Control environment.

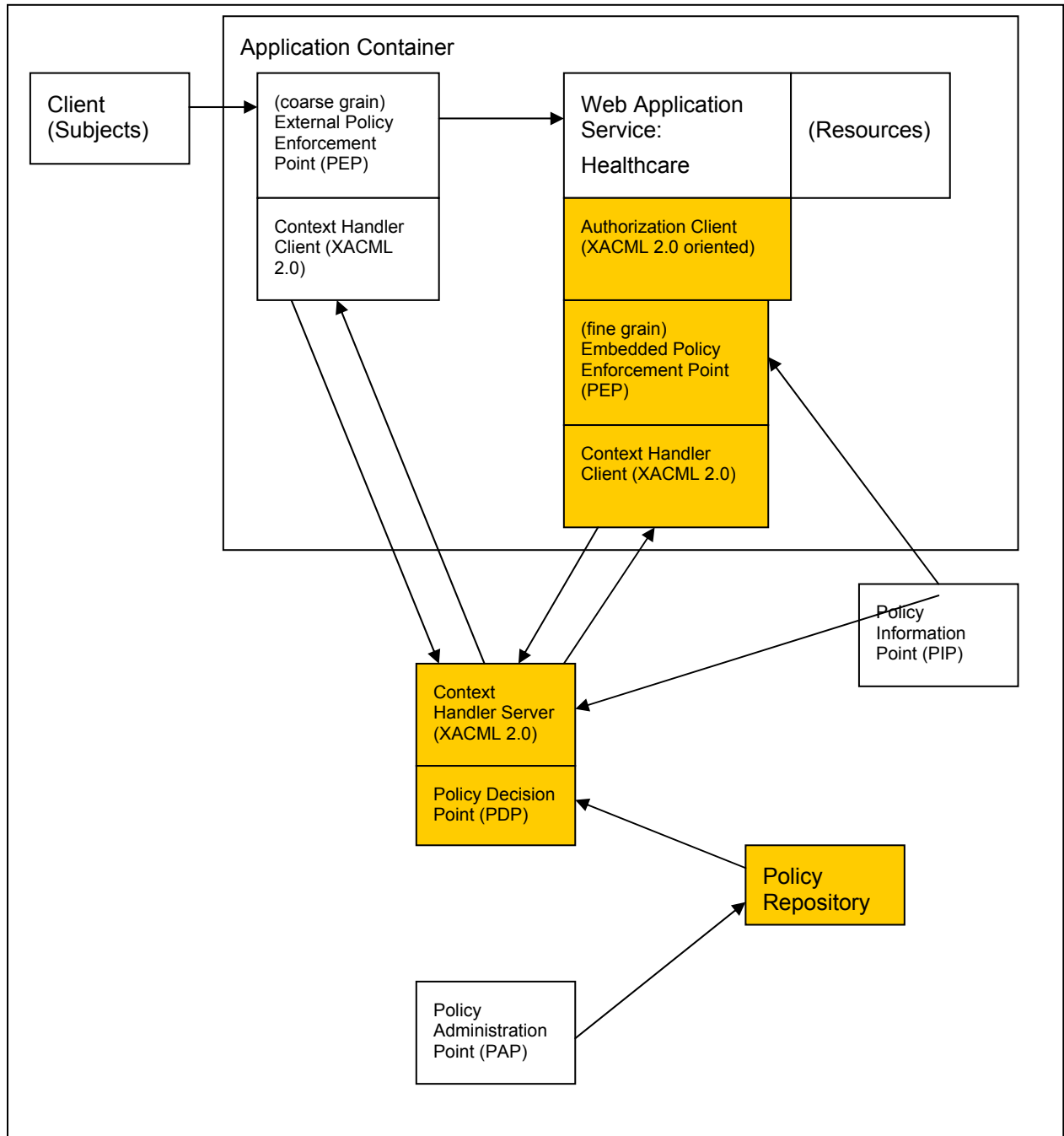
122 High level descriptions of the Interop 01 use cases are carried over from Interop 01 in order to provide  
123 additional context for understanding the scope of the Interop 02 use cases.

124 Note: A **hyperlink index of all the messages and policies** used in this document to  
125 enable easy navigation within the document may be found here: [[xacml-msg-policy-](#)  
126 [index](#)]. At the beginning of each message and policy is a link back to the index so that it  
127 is easy to go to a message, then if you want to go to another one just go back to the  
128 index and pick it.

129 The following diagram shows overall use case environment:

130





132

133

134

135

Figure 1

136 In the figure above, it is assumed that the interoperable vendor-specific product components (shaded)  
137 that will be demonstrated at the Interop event include:

- 138 • the Authorization Client/PIP-client,
- 139 • the PEP/context-handler-client,

- the PDP/context-handler-service/PIP-client,

141

142 All other modules shown on the diagram are assumed to be part of the common environment

143 A brief description of both the vendor-specific and generic components on the diagram follows:

- **Client:** For this Interop, the Client will be a standard web browser, with screens displayed from a Web Application Service.

- **Application Container:** A typical application server platform which hosts web applications and provides common services for those web applications such as authentication and coarse-grain authorization, general APIs for a variety of services including providing contexts for the applications such as authenticated user contexts.

- **Policy Enforcement Point (PEP) External (coarse-grain):** this module will only be incidentally included in the Interop 2 Test Environment. It is generally responsible for authorizing a user's access to an application and establishing a user context from which the application may obtain high level user identity attributes, such as corporate role (e.g. employee).

- **Context Handler Client (XACML 2.0):** this is a general purpose XACML 2.0 component that is responsible for assembling XACML 2.0 Authorization Decision Requests based on parameters supplied by any type of PEP and implementing the communication protocol to send the request to a PDP and for returning Authorization Decision Responses from the PDP to the PEP. The Context Handler Client is generally considered a built-in part of the PEP, but it has proven useful to identify it explicitly for its message handling capabilities, which involve normative XACML 2.0 message formats as distinct from the non-normative higher level PEP enforcement capabilities. Note: this same logical functional module is available for use by both the external and embedded PEP modules and is shown separately associated with each respectively in the diagram.

- **Context Handler Server (XACML 2.0):** this is a general purpose XACML 2.0 component that is responsible for handling XACML Authorization Decision Requests and setting up a context for delivering the Request to the PDP. It also may aid the PDP by calling out to PIPs for additional authorization attribute data needed for Policy evaluation. Finally, it handles the Response context and when the PDP returns, it packages the Response context to a XACML Decision Response to be returned by the communication protocol to the PEP.

- **Policy Information Point (PIP):** an enterprise-specific repository of attribute data that is made available to support authorization decisions. In general, access is enterprise-specific and the Context Handler will need to be outfitted with custom modules to access one or more PIPs.

- **Policy Decision Point (PDP) (XACML 2.0):** this is the main XACML 2.0 Policy Evaluation module that implements the normative XACML 2.0 Policy structures described in [XACML 20].

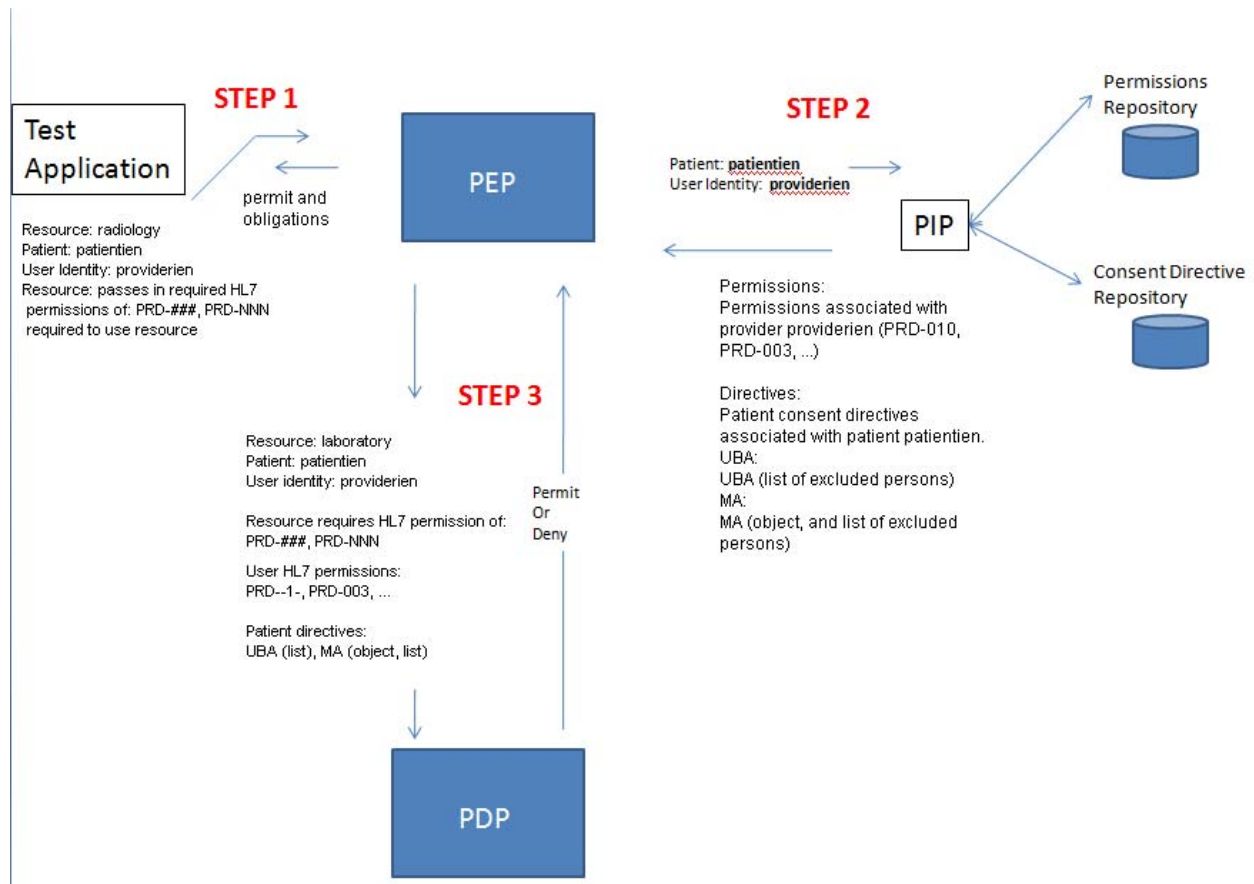
- **Policy Repository:** this is a generic vendor-specific mechanism for storing common XACML 2.0 Policies. In general, XACML 2.0 Policies are not stored in the XML format represented in the [XACML 20] specification. (However, it is expected that the policies from the repository can be exported and imported in XML format as needed to demonstrate interoperability and consistency of Policy representation.)

- **Policy Administration Point (PAP):** this is a generic vendor-specific module for managing XACML 2.0 Policies that may be stored in and retrieved from the Policy Repository. For the purposes of Interop 02, the PAP is just assumed to exist as needed and does not play an active role in the use cases.

- **Web Application Service:** this is the main healthcare web application that provides operations and access to the Resources of the healthcare environment. The main purpose it plays in the Interop 02 scenarios is to demonstrate how fine-grained authorization requests may be externalized from application logic in a standard manner. In the Interop 02 environment, all authorization requests and responses are handled by a single application-specific module. It generally will do exactly what it is directed based on the results of the authorization request, such as informing the caller if they have been denied access, obtaining commitments from the caller for access to sensitive information, filtering data based on user access rights, etc.

- 191 • **Resources:** these are the main health care resources, such as patient records, medical  
192 documents such as lab tests, reports, images, etc. The resources are generally tagged with  
193 attributes indicating access requirements that may come into play in authorization decisions.
- 194 • **Policy Enforcement Point (PEP) Embedded (fine-grain):** this module is typically a vendor-  
195 specific API kit, which can be embedded as part of an application process. One of its main  
196 features is to provide a standard API (in Interop 02, a custom API was used to specifically meet  
197 the Interop requirements) to the Authorization Client to facilitate passing XACML Authorization  
198 Request attributes and returning XACML Response Decisions and Obligations. As shown in the  
199 diagram this module uses application context as a basis for obtaining attributes required for  
200 authorization from a PIP. Typical attributes collected at this level include HL7 Provider  
201 Permissions, HL7 Resource Permission, and HL7 Patient Privacy Constraints.
- 202 • **Authorization Client:** this module provides a standard API (in Interop 02, a custom API was  
203 used to specifically meet the Interop requirements) to enterprise applications for submitting  
204 authorization requests and returning authorization responses and obligations. Typically, only key  
205 application identifiers of actors in context are passed over this interface, such as in a healthcare  
206 application, a Provider ID, a Patient ID, and Resource ID and operation.

207  
208 A more detailed view of the interaction between the components described above is shown in Figure 2.  
209



210  
211  
212 Figure 2. Example interaction between components  
213

214 In step 1, user access to a resource (e.g. patient record or application functionality) within the healthcare  
215 application triggers a call to the PEP. The application identifies the resource being accessed and passes  
216 the patient identifier (patientEIN), the application user identity (providerEIN) and the permissions required

217 to access the resource. In this model the application subject matter experts determine the set of  
218 permissions that the user must possess to access the resource.

219  
220 In step 2, the PEP gathers information that will be needed to evaluate the request. The PEP requests two  
221 types of information from the PIP: enterprise permissions and patient consent directives. The  
222 permissions held by the application user are retrieved by the PIP from a permission repository based on  
223 the providerEIN or enterprise role. Permissions held by the application user are enumerated as a list of  
224 permission codes [HL7-Perm]. The operation of these permissions can be constrained by the patient  
225 through consent directives [HL7-Consent]. Consent directives are retrieved by the PIP from a consent  
226 directive repository based on the patientEIN. Consent directives are vary as to type of directive. A  
227 complete bar to access a patient's medical record can be specified using a UBA directive based on the  
228 providerEIN or enterprise role. A directive masking data object within a patient's medical record can be  
229 specified using a MA directive based on the object and providerEIN or enterprise role.

230  
231 In step 3, the PEP (and related context manager) assembles the acquired information into the decision  
232 request. The request contains the resource being accessed, which can be a complete medical record or  
233 data object within a patient's medical record (e.g. a radiology report). The request also includes  
234 permissions that that are required for access to the resource. The application user's permissions are  
235 passed in the request as well as any consent directives previously made by the subject of the record  
236 demarcated by type (i.e. UBA, MA, etc.).

237  
238 The PDP evaluates the appropriate policies and returns an access decision to the PEP which is passed  
239 to the application along with any obligations.

240

## 241 **1.5.1 Healthcare Fine Grain Authorization Use Cases**

242 The RSA 2008 XACML 2.0 Interop will consist of a Healthcare Application that will demonstrate the use of  
243 XACML 2.0 to handle fine grain authorization use cases. . The use cases are drawn from the work of the  
244 Healthcare Information Technology Standards Panel [HITSP] effort in support of the American Health  
245 Information Community (AHIC) use cases.

246 The use cases selected are drawn from three common categories of access control found in the  
247 healthcare environment: enterprise permissions, patient directives, and business rules. Enterprise  
248 permissions are rights to access certain enterprise information or functionality. Patient directives are  
249 specific restrictions by the subject of the information on access or treatment of the information. Business  
250 rules impose actions on system components that must be honored prior to granting access.

251 The use cases in this section all operate within the infrastructure shown in Figure 1. In particular, all use  
252 cases are driven by web browser client that accesses the Healthcare Web Application Service hosted in a  
253 generic application server container, which provides front-end authentication and coarse grained  
254 authorization services to enable a user context to be presented to the Healthcare Application.

255 The Healthcare application will process user requests and when authorization is needed for a specific  
256 action within the Healthcare application, the application logic will collect user attributes, resource  
257 attributes, and any other context needed to request authorization of the current user for the specific action  
258 on the specific resource. The application-level authorization request will be submitted to a common  
259 Authorization Client that is designed to be a common point for submission of requests from anywhere  
260 within the Healthcare application.

261 The Embedded PEP in figure 1 will take requests from the Authorization Client and submit them to the  
262 ContextHandler Client, which is a standard XACML 2.0 Request/Response message handler.

### 263 **1.5.1.1 Use Case: Fine Grain HL7 Role/Permission based access control**

264 This Fine Grain HL7 Role/Permission (RBAC) use case will demonstrate the use of the XACML 2.0 RBAC  
265 Profile for defining PolicySets that can be used to govern access to resources and it will demonstrate the  
266 use of HL7 Identifiers [HL7-Perm] for identifying Roles and Permissions in both the Policy and application  
267 contexts.

268

### 269 **1.5.1.2 Use Case: Fine Grain HL7 Patient Consent Directives access control**

270 This Fine Grain HL7 Patient Consent Directive will demonstrate that a patient is able to control access to  
271 part or all of their record.

272 Note:In principle, rules can be established that define the bare minimum of visibility to  
273 healthcare records, which presumably would include some access, for example, by the  
274 attending physician who entered the records, who presumably is protected from having to  
275 ever disclose such information by the rules of doctor/patient confidentiality.

276 The Patient Directives will be collected by the application and passed to the Authorization Client as part of  
277 the Authorization Decision Request. To accomplish this, patients use consent directives to constrain  
278 functions that are expressed by healthcare enterprise permissions. These constraints use a specific  
279 vocabulary to ensure semantic interoperability [[HL7-Consent](#)].

280

### 281 **1.5.1.3 Use Case: Fine Grain Signed Progress Note Attributes Rule based control**

282 The Fine Grain Signed Progress Note use case will demonstrate the ability enforce a business rule. In  
283 this use case, a document that has been requested by a user has metadata attributes indicating who the  
284 document author is and whether the author has digitally signed the document to indicate that it is ready  
285 for broader distribution. The reason for enforcing this business rule is that unfinished progress note can  
286 contain unconfirmed information that should not be used as the basis of action by other clinicians. The  
287 metadata attributes of the progress note will be passed by the application to the Authorization Client and  
288 be included in the Authorization Request. The PolicySet will determine whether in addition to normal  
289 access requirements have been met, that this additional condition also has been met in order to render  
290 the correct authorization decision.

291

### 292 **1.5.1.4 Use Case: Fine Grain Emergency Override Obligations**

293 Obligations that may be triggered in an emergency override include increased logging of activities. In this  
294 case, an emergency is declared and access control policies are overridden to prevent loss of life or  
295 severe injury to the patient. During the emergency, increased logging will be required at the PEP to  
296 ensure exceptions to standard access control policies was appropriate.

### 297 **1.5.1.5 Use Case: Fine Grain Data Filtering Obligations**

298 TBD

## 299 **1.5.2 Use Case: Coarse Grain Authorization Decision Request/Response**

300 The generic Interop 01 Authorization Decision Request/Response use case is based on a Client  
301 application requesting services from a service application that has access to resources necessary for  
302 servicing the requests. In general the client will request the resources in an application enterprise domain-  
303 specific manner, which is, in general, totally independent and outside of the security infrastructure  
304 governed by XACML 2.0.

305 The way XACML 2.0 is introduced to the client-service application environment is shown in Figure 1. An  
306 external (coarse-grain) PEP is inserted to the data stream between the client and service. The PEP, itself,  
307 may be considered to be a domain-specific entity, such as a web server or a servlet engine, however, the  
308 domain-specific PEP will have an extension capability, to which a XACML 2.0 context handler is attached.  
309 In general, the context handler can be either local to the PEP or PDP, but since we are interested in PEP-  
310 PDP "interoperability", only the PEP-local case will be considered.

311 The XACML 2.0 request context and response context are represented in Figure 1 by the PEP->PDP and  
312 PDP->PEP arrows, respectively.

313 Interop 02 does not include any coarse grain external use cases that were not covered in Interop 01.

314 **1.5.3 Use Case: Policy Exchange**

315 The Interop 01 Policy Exchange use case is based on a PAP entity creating policies and placing them in  
316 a repository. The PDP retrieves the policies from the repository and uses them in the process of  
317 evaluating the Authorization Decision requests.

318 Interop 02 does not currently include any explicit policy exchange use cases that were not demonstrated  
319 in Interop 01.

320

---

## 321 2 Use Cases: Healthcare: Fine Grained Authorization

### 322 2.1 Introduction to the Healthcare Application

323 Note: A **hyperlink index of all the messages and policies** used in this document to  
324 enable easy navigation within the document may be found here: [[xacml-msg-policy-](#)  
325 [index](#)]. At the beginning of each message and policy is a link back to the index so that it  
326 is easy to go to a message, then if you want to go to another one just go back to the  
327 index and pick it.

#### 328 2.1.1 Flexibility provided by XACML

329 The Interop demonstrates how it is possible to use XACML to separate access control logic from the  
330 business logic provided by an application. The application does not make access control decisions itself,  
331 rather it exports its resource model to the policy writers. The access control policy is then defined with  
332 XACML based on the vocabulary that the application provides.

333 The Interop demonstrates how an HL7-based access control vocabulary model can be implemented  
334 using XACML.

335 However, a completely different access control model would be possible. For instance if the same  
336 medical application is deployed in a different regulatory environment, the access control model can be  
337 changed in the XACML policies, without modifications to the application itself. This benefits the  
338 application vendor as the same application can be used by a wider audience, and the customers who get  
339 access to a wider selection of applications and better flexibility.

340 It will also be possible to change the access control model as requirements change in the future. This will  
341 save time and money as the problems with “legacy” applications in the future will be smaller if those  
342 applications are XACML-based. As long as the application itself still meets the requirements, only the  
343 policies need to be changed to meet the new access control requirements.

344

#### 345 2.1.2 Use of Virtual Roles

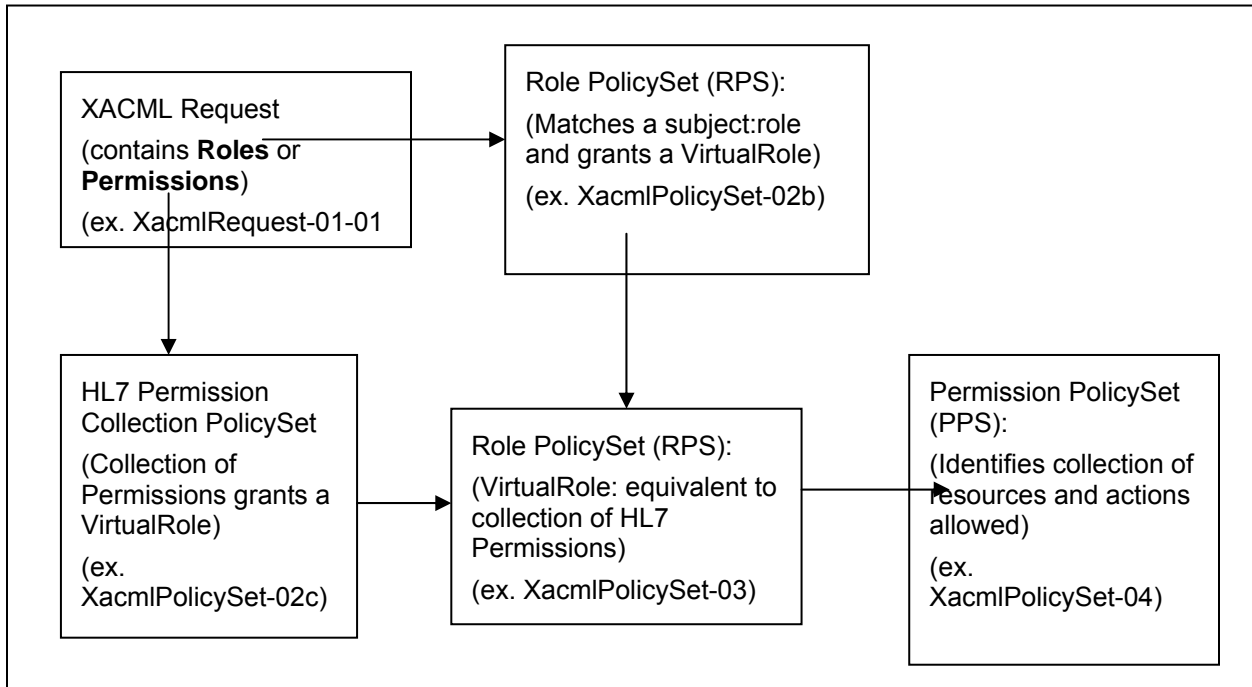
346 Analysis of the VA Healthcare requirements determined that a particular variation of Role Based Access  
347 Control (RBAC) was required. It has been determined by the HL7 Security Technical Committee that for  
348 inter-organization purposes that collections of specific commonly understood HL7 Permission Identifiers  
349 will be used instead of Role Identifiers as a basis of access control decisions:

350 “Roles are not currently part of the HL7 permission catalog definition. At this time, roles  
351 are considered to be locally defined by organizations that build them using HL7 standard  
352 permissions. Roles that are inter-organizational in scope may be added to this process  
353 at a future date.” [[HL7-RoleEng](#)]

354 Therefore, these scenarios will be based on the assumption that the requesting user has obtained a  
355 collection of HL7 Permissions that are available from the User Context in the Application Container. The  
356 XACML Requests, therefore, will be constructed such that these Permissions are included as a set of  
357 Subject Attributes, each of which contains an individual HL7 Permission.

358 In general, an organization may have local roles defined for any number of purposes including those that  
359 comply with the HL7 requirements. Therefore the PolicySets being used for this Interop application are  
360 designed to accommodate both the HL7 Permission-based concept of Roles and the traditional concept  
361 of Roles.

362 The scenarios are based on the concept that access to the resources requires a collection of HL7  
363 Permissions. The PolicySets are designed such that specific collections of HL7 Permissions map to a  
364 “VirtualRole”. Resources are protected based on the VirtualRoles that a user has obtained based on the  
365 HL7 Permissions that are included in the XACML Request.



367

368 The diagram above shows the basic policy structure used to implement virtual roles. It is an extension of  
 369 the XACML RBAC Profile model, which is based on Role PolicySets (RPS) and Permission PolicySets  
 370 (PPS). It extends the usual model where a web application has the name of a user role and sends it to  
 371 the PDP for authorization, to also support a scenario where instead of a role, the application has a  
 372 collection of Permissions that it can send to the PDP.

373 Since a role may actually be considered a collection of permissions there are basically two manners in  
 374 which the same concept may be implemented: i.e. the concept of a particular role can be represented by  
 375 either a container-supplied role name, or a collection of permissions that the container is passing through  
 376 from some other source. Therefore, two completely different sets of information can be sent in a Request  
 377 which represents the same actual collection of permissions.

378 Therefore, a virtual role is defined to represent the actual collection of permissions (i.e. in the XACML  
 379 RBAC model this translates to the list of Permission PolicySets (PPSs) that the RPS points to. In the  
 380 diagram above the box with ex XacmlPolicySet-03 is the one that is used for this virtual role, whereas the  
 381 ex. XacmlPolicySet-04 is used as a PPS that the RPS points to. This example RPS plus PPS is the  
 382 standard XACML RBAC Profile model.)

383 However, since we have 2 completely different request types that can map to this same virtual role, we do  
 384 not have the Target defined in the virtual role, but instead we define two independent Targets in two  
 385 separate PolicySets, one which checks the Request for a container defined role (ex. XacmlPolicySet-02b)  
 386 and one that checks the Request for a specific collection of permissions, which are effectively just another  
 387 way to represent the same role.

388 In practice, one might expect that there be some role management infrastructure such that the role  
 389 names used in the containers and the collection of permissions that were contained in that role could be  
 390 exported in some manner and imported to the PolicySet definitions so that Policy Administrators would  
 391 not need to worry about maintaining consistency of these definitions at the Policy level, but maintain the  
 392 consistency in an external role management system. Such role management is beyond the scope of this  
 393 Interop and therefore the PolicySets will be manually maintained in this document.

394



### 395 **2.1.3 Scenario structure**

396 The scenarios that are in the following sections have the following general characteristics:

- 397 • Each “scenario” consists of a request and response.
- 398 • The scenarios are grouped in such a way that a group of scenarios forms a meaningful sequence  
399 of requests and responses in terms of a real world sequence of operations a user might perform  
400 in order to complete a specific task, such as accessing their account and then performing a  
401 transaction within their account.
- 402 • The scenarios are quasi-stateful in the sense that the sequence within a group has a specific  
403 order, and the response to the request of scenario ‘n’ within the sequence, generally may be used  
404 as the starting point for the request for scenario ‘n+1’ within the same scenario sequence.
- 405 • The first scenario of a group is preceded by simply entering the URL of the scenario group which  
406 will return a form that contains the initial default values of the first request. The user may fill in the  
407 available fields in the form to replace the default values.

408 Associated with each scenario are 3 sets of data:

- 409 1. Client request and response data
- 410 2. XACML request and response xml messages
- 411 3. XACML xml policy structures that are to be applied to the XACML Request and used to  
412 prepare the XACML Response.

413

414 All the data elements used in scenarios are identified by names in the associated XACML Vocabulary.

415 These data sets, the elements they contain, and the XACML Vocabulary they use are described in detail  
416 in the following sections.

417

#### 418 **2.1.3.1 Client request and response data**

419 There is a table for the request and for the response for each scenario. Each table has 3 columns:

- 420 • Variable Name: this is the name that the variable has when it comes in the request from the client  
421 that is initiating the scenario. If the client is a browser doing a form POST, then the variable name  
422 is the “name” part of an HTML form element name/value pair. If the client is a SOAP client, then  
423 the variable name is the leaf tag name (with no namespace prefix) of an xpath expression that  
424 might be used to obtain the value.
- 425 • Value: this is the value part of the variable. In an HTML form element it is the value of the  
426 name/value pair. In a SOAP request it is the content part of the variable tag. For each scenario,  
427 this value is the value that will be in the request that is submitted to the PEP, as opposed to the  
428 initial or default value the variable might have had before the request was actually submitted.
- 429 • urn: this is the XACML Vocabulary name of this variable. It is used to map the variables from the  
430 client request to a specific attribute in the XACML Request, and to map the attributes in the  
431 XACML Response to extensions to the client request that will be delivered to the Web  
432 Application. In addition, it identifies the additional variables that the Web Application will add to  
433 the response that is returned to the client.

434

435 Note: the client request table contains placeholders for all variables that get delivered to the Web  
436 Application, and these values may be original default values, updated values that were updated by the  
437 user preliminary to submitting the request, resource variables that should not be displayed initially to the  
438 user and have prefix “Resource”, and obligation values to be updated by the PEP to pass back XACML  
439 response data for later display such as the Decision and the Status and have prefix “Obligation”.

440 Note: in general, a XACML Request can contain Subject, Action, Resource, and Environment attributes.  
441 In these scenarios, all these attributes are being provided within the client request. Any resource  
442 attributes that the client needs will be provided from the Web Application in the previous response, which

443 sets the framework for the next request. Typically, in real world situations one might expect Subject  
444 attributes to be provided by an identity management module at authentication time or by a PIP from the  
445 ContextHandler, one might expect Resource attributes to be obtained from the application itself which  
446 might play a role in executing the xacml request for authorization, or by a PIP from the ContextHandler.  
447 The net result is that eventually the attributes must be provided to successfully get authorized, and since  
448 this interop is focussed on the vendors interfacing across the xacml req/rsp interface, we are simply  
449 providing a mix of attributes from the Subject and Resource using the above mechanisms to avoid  
450 complications with potentially proprietary attribute accessing schemes outside the scope of the Interop.

### 451 2.1.3.2 XACML Request and Response

452 This section describes the xacml Request and Response messages for the scenario. All Request and  
453 Response attributes are identified by a urn from the XACML Interop Vocabulary, which enables seamless  
454 mapping of data values between the client layer and the policy layer.

455 It is recommended that the SAML 2.0 profile of XACML v2.0 [[SAML-XACML20](#)] be used for PEP-PDP  
456 communications. (Note: make sure to use [[SX20-ASSN-SCH](#)] and [[SX20-PROT-SCH](#)] schema files and  
457 specification in 17-Nov-05 Errata version.)

458 ISSUE: TBD: It is assumed there will be no signing or encryption of messages in the XACML Request  
459 and Response protocols.

460 Following are the expected SOAP-wrapped request and response messages. Further analysis needs to  
461 be done here to confirm these formats and determine if they can be used by the participating vendors.

462 **Sample SOAP SAML XACML Request wrapper:** [[xacml-msg-policy-index](#)]

```
463 <?xml version="1.0" encoding="UTF-8"?>  
464 <soapenv:Envelope  
465   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
466   xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
467   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
468   <soapenv:Body>  
469     <xacml-samlp:XACMLAuthzDecisionQuery  
470       xmlns:xacml-samlp="urn:oasis:xacml:2.0:saml:protocol:schema:os"  
471       ID="_e064bd912f83c1544fea110307000acf"  
472       IssueInstant="2007-05-21T22:00:36Z"  
473       Version="2.0">  
474       <xacml-context:Request  
475         xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">  
476         <!-- See [XACML-Request-01] for sample content of this element -->  
477         </xacml-context:Request>  
478       </xacml-samlp:XACMLAuthzDecisionQuery>  
479     </soapenv:Body>  
480   </soapenv:Envelope>
```

481 The request message above contains 3 protocol levels:

- 482 1. soapenv: is the SOAP layer. A SOAP Envelope contains a SOAP Body.
- 483 2. xacml-samlp: is the SAML protocol layer, which is enabled by the XACML extension to the SAML  
484 protocol, which is described in [[SAML-XACML-20](#)] specification and in the [[SX20-PROT-SCH](#)]  
485 schema. Note that the usual samlp: is not declared here because xacml-samlp: extends samlp:  
486 and will transparently include the samlp: base declarations.
- 487 3. xacml-context: is the XACML request/response layer which is described in [[XACML-CORE](#)].

490 **Sample SOAP SAML XACML response wrapper:** [[xacml-msg-policy-index](#)]

```
491 <soapenv:Envelope  
492   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
493   xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
494   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
495   <soapenv:Body>  
496     <samlp:Response  
497       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"  
498       ID="A12345602"  
499       Version="2.0"  
500       IssueInstant="2007-05-09T00:00:01Z">  
501     <samlp:Status>
```

```

502     <samlp:StatusCode
503       Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
504   </samlp:Status>
505   <saml:Assertion
506     xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
507     Version="2.0"
508     ID="A12345603"
509     IssueInstant="2007-05-09T00:00:01Z">
510     <saml:Issuer>xacml.interop.com</saml:Issuer>
511     <saml:Statement
512       xmlns:xacml-saml="urn:oasis:xacml:2.0:saml:assertion:schema:os"
513       xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
514       <xacml-context:Response
515         xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
516         <!-- See [XACML-Response-01] for sample content of this element -->
517       </xacml-context:Response>
518     </saml:Statement>
519   </saml:Assertion>
520 </samlp:Response>
521 </soapenv:Body>
522 </soapenv:Envelope>

```

523 The response message above contains 3 protocol levels:

- 524 1. soapenv: is the SOAP layer. A SOAP Envelope contains a SOAP Body.
- 525 2. samlp: is the SAML Protocol layer, which is explicitly declared this time because in the response
- 526 case the xacml extension is lower in the samlp: protocol. In particular, samlp: requires a
- 527 saml:Assertion, which in turn includes a saml:Statement. It is within the saml:Statement that the
- 528 xacml extension occurs and is referred to as xacml-saml: because it extends the
- 529 saml:Assertion/saml:Statement with the XACMLAuthzDecisionStatementType. The details are
- 530 described in the [SAML-XACML-20] specification and the [SX20-ASSN-SCH] schema.
- 531 3. xacml-context: is the XACML request/response layer which is described in [XACML-CORE].

532

### 533 2.1.3.3 XACML Policy

534 This section describes the policy and how it is applied to the xacml Request and Response. Policy data

535 references attributes by urn, which enables seamless mapping between the policies and the XACML

536 Requests and Responses.

537 The policies given are intended to represent one possible implementation of the policies that represent

538 the rules. It is expected that each vendor will implement their own policies using the rules that are given

539 and the resulting policies should be the functional equivalent of the samples given with each scenario.

540 TBD: It is expected that there will be some testing required before these sample policies are considered

541 to be totally correct, and this document will be updated with the final working version of the policies.

542

#### 543 **2.1.4 Use Case Policy Pseudo-code**

544 It was found to be very helpful in the process of requirements analysis to identify the logic of the  
545 processing that the policies are expected to do, as well as identifying the specific fine-grained attributes  
546 that are involved in the policy decisions. Various means were attempted to simplify the XACML Policy  
547 representations and it turned out that simple if-else pseudo-code (p-code) was the most efficient means  
548 of explaining the policy processing for both the business-oriented contributors and the technical  
549 contributors to communicate the requirements.

550 Additional examples of the use of p-code, with an emphasis on policy and rule combining structures, can  
551 be found in Appendix C of [[XACML-2.0-CORE](#)].

552 The following block of p-code represents the basic logic of all 5 use cases covered in this document:

```
553 if ( ! (request.subject.locality == request.environment.locality) )
554   if ( ! ("hl7.pea-001" ==
555     any-of(request.subject.hl7.permission)) )
556     Result = Deny
557   else
558     Result = Permit
559     response.add(Obligation(emergency.override, ffon-permit))
560   end
561 end
562 if ( ! Result == Deny )
563   if (request.hl7.conf-code == "UBA")
564     if ( ! (request.subject.subject-id ==
565       any-of(request.resource.hl7.dissented-subject-id) ) )
566       Result = Permit
567     else
568       Result = Deny
569       response.add(Obligation(privacy.constraint, ffon-deny))
570     end
571   end
572 end
573 if ( ! (Result == Deny) )
574   if (request.hl7.conf-code == "MA")
575     if (request.subject.subject-id ==
576       any-of(request.resource.hl7.object.1.dissented-subject-id) )
577       Result = Permit
578       response.add(Obligation(privacy.constraint.object.1, ffon-permit))
579     end
580     ...
581     if (request.subject.subject-id ==
582       request.resource.hl7.object.n.dissented-subject-id)
583       Result = Permit
584       response.add(Obligation(privacy.constraint.object.n, ffon-permit))
585     end
586   end
587 end
588 if ( ! (Result == Deny))
589   if (request.resource.type == "resource.hl7.progress-note")
590     if (request.resource.progress-note.signed == false)
591       if ( ! (request.subject.subject-id ==
592         anyof(request.resource.progress-note.author-subject-id) ) )
593         Result = Deny
594       end
595     end
596   end
597 end
598 if ( ! (Result == Deny))
599   if (request.subject.role == role.hl7.physician)
600     check-vrole-permissions()
601   end
602   check-vrole-permissions() //unscreened permission comparison
603   if ( ! (Result == Permit) ) // screened permission comparison
604     if ( (hl7.prd-003 == subset-of(subject.hl7.permission[n-values]) &&
605       (hl7.prd-005 == subset-of(subject.hl7.permission[n-values]) &&
606       (hl7.prd-006 == subset-of(subject.hl7.permission[n-values]) &&
607       (hl7.prd-009 == subset-of(subject.hl7.permission[n-values]) &&
608       (hl7.prd-010 == subset-of(subject.hl7.permission[n-values]) &&
609       (hl7.prd-012 == subset-of(subject.hl7.permission[n-values]) &&
610       (hl7.prd-017 == subset-of(subject.hl7.permission[n-values]) )
611       check-vrole-permissions()
612     end
613   end
614   // need to add here a deny if no permit found
615 end
616
617 check-vrole-permissions()
618   if (request.resource.type == "hl7-medical-record")
619     if ( request.resource.hl7.permission[m-values] ==
620       subset-of(subject.hl7.permission[n-values] )
621       Result = Permit
622     end
623   end
```

624           return

625       There are 5 logic blocks in the main p-code module above, each headed by an “if ( ! ( Result == Deny ) ”  
626       clause. This clause represents the deny-overrides algorithm of the top-level policy set. Within each clause  
627       is a Policy or PolicySet that is a direct child of the top-level PolicySet.

628       The functionality encapsulated in each logic block is as follows:

- 629           • 1<sup>st</sup> logic block: emergency access logic
- 630           • 2<sup>nd</sup> logic block: patient consent UBA logic
- 631           • 3<sup>rd</sup> logic block: patient consent MA (data filtering) logic
- 632           • 4<sup>th</sup> logic block: business rules logic block
- 633           • 5<sup>th</sup> logic block: HL7 role/permissions logic block

634       The realization of the logic of these modules within xacml is described in the 5 use case sections below:  
635       sections 2.2.1 -> 2.2.5.

636

## 637 2.2 Detailed Description of Fine Grained Authorization Use Cases

638 Index to sample messages and policies:

- 639 1. [Pseudo code represents the basic logic of all 5 use cases](#)
- 640 2. [Sample SOAP SAML XACML Request wrapper](#)
- 641 3. [Sample SOAP SAML XACML response wrapper](#)
- 642 4. [Use Case 1: HL7 Role/Permissions Data Elements](#)
- 643 5. [XacmlRequest-01-01](#)
- 644 6. [XacmlPolicySet-01-top-level](#)
- 645 7. [XacmlPolicySet-02b-N](#)
- 646 8. [XacmlPolicySet-02c-N-PermCollections](#)
- 647 9. [XacmlPolicySet-03-N-RPS-med-rec-vrole](#)
- 648 10. [XacmlPolicySet-04-N:PPS:PRD-004](#)
- 649 11. [Use Case 2: HL7 Patient Consent Directive Data Elements](#)
- 650 12. [XacmlRequest-02-01](#)
- 651 13. [XacmlRequest-02-02](#)
- 652 14. [XacmlPolicySet-02a-CDA](#)
- 653 15. [Use Case 3: HL7 Attribute Based Rules Data Elements](#)
- 654 16. [XacmlRequest-03-01](#)
- 655 17. [XacmlPolicySet-02d-prog-note](#)
- 656 18. [Use Case 4: Emergency Access Data Elements](#)
- 657 19. [XacmlRequest-04-01](#)
- 658 20. [XacmlRequest-04-02](#)
- 659 21. [XacmlPolicySet-02f-emergency](#)
- 660 22. [Use Case 5: Data Filtering Data Elements](#)
- 661 23. [XacmlRequest-05-01](#)
- 662 24. [XacmlPolicySet-02e-MA](#)

663 The above list of links is intended to be used to anchor quasi-random navigation within the document.  
664 Most of the destination on the above list include a backpointer link that comes back to this list. So, one  
665 can start in the above list, click a link, look around, and then click the backpointer to come back here.  
666 (may not seem that great, but it can be a lot easier than scrolling up and down looking for things)

667 **Cautionary Note:** It has been found since the interop that at least XacmlPolicySet-01-  
668 top-level should be defined as “**ordered-deny-overrides**” and **not** simply  
669 “**deny-overrides**”. The result appears to be that some implementations, while giving a  
670 correct response, may return Obligations different than what is expected based on the  
671 ordered processing as specified in the p-code. Further consideration of this situation will  
672 be analyzed, and resources permitting, this document will be updated with a full  
673 explanation and corrections wherever required.

674 **2.2.1 Details: HL7 Role/Permission**

675 **2.2.1.1 Scenarios for HL7 Role/Permission**

676 **2.2.1.1.1 DEMO HL7 Permission Access Control (related permissions granted)**

677  
678 Initial State / Pre-condition:

- 679 • Dr. Alice has all related permissions to read a medical record.
- 680 • Dr. Alice attempts to view the medical records for Anthony Gurrola.

681 Result:

- 682 • Dr. Alice is able to access the medical record including Anthony Gurrola's sensitive data.

683  
684 Dr. Alice would have the following HL7 Permissions in Table 1 available in order to access a medical  
685 record

Permissions Granted	HL7 Permission Code	HL7 Permission Title
√	PRD-006	Patient Identification and Lookup
√	PRD-017	Review Progress Notes
√	PRD-012	Review Past Visits
√	PRD-003	Review Medical History
√	PRD-005	Review Vital signs/Patient Measurements
√	PRD-009	Review Current Directory of Provider Information
√	PRD-010	Review Patient Medications

**Table 1 – RBAC Permissions identified to complete use case**

686  
687  
688 **2.2.1.1.2 DEMO HL7 Permission Access Control (no permissions granted)**

689  
690 Initial State / Pre-condition:

- 691 • Dr. Alice has all related permissions to read a medical record.
- 692 • Using screen supplied by the application, the security administrator removes HL7 Permissions  
693 that were initially assigned to Dr. Alice.
- 694 • Dr. Alice attempts to view the medical records for Anthony Gurrola.

695  
696 RESULT:

- 697 • Dr. Alice is unable to access the medical record including Anthony Gurrola's sensitive data.

698



699

Permissions Granted	HL7 Permission Code	HL7 Permission Title
	PRD-006	Patient Identification and Lookup
	PRD-017	Review Progress Notes
	PRD-012	Review Past Visits
	PRD-003	Review Medical History
	PRD-005	Review Vital signs/Patient Measurements
	PRD-009	Review Current Directory of Provider Information
	PRD-010	Review Patient Medications
<b>Table 2 – RBAC Permissions identified to complete use case</b>		

700

701

702 **2.2.1.2 Detailed Data: HL7 Role/Permissions**

703 **2.2.1.2.1 Detailed Data Elements**

704 The following list describes the critical data elements that are passed in the requests that are needed for  
 705 the policies to work. Note that the purpose of the table is to identify the variable identifiers and values that  
 706 will be used in different scenarios.

707 **Use Case 1: HL7 Role/Permissions Data Elements:** [[xacml-msg-policy-index](#)]

Variable AttributeId Value(s)	Full Variable AttributeId URN Full Value URN(s)
subject:subject-id Dr. Alice	urn:oasis:names:tc:xacml:1.0:subject:subject-id Dr. Alice
subject:locality Facility A	urn:oasis:names:tc:xacml:1.0:subject:locality Facility A
subject:hl7:permission hl7:prd-003 hl7:prd-005 hl7:prd-006 hl7:prd-009 hl7:prd-010 hl7:prd-012 hl7:prd-017	urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission urn:va:xacml:2.0:interop:rsa8:hl7:prd-003 urn:va:xacml:2.0:interop:rsa8:hl7:prd-005 urn:va:xacml:2.0:interop:rsa8:hl7:prd-006 urn:va:xacml:2.0:interop:rsa8:hl7:prd-009 urn:va:xacml:2.0:interop:rsa8:hl7:prd-010 urn:va:xacml:2.0:interop:rsa8:hl7:prd-012 urn:va:xacml:2.0:interop:rsa8:hl7:prd-017
subject:role hl7:physician Note: <b>optional</b> this is alternative to permission set above	urn:oasis:names:tc:xacml:2.0:subject:role urn:va:xacml:2.0:interop:rsa8:role:hl7:physician
resource:resource-id Anthony Gurrola	urn:oasis:names:tc:xacml:1.0:resource:resource-id Anthony Gurrola
resource:hl7:type resource:hl7:medical-record	urn:va:xacml:2.0:interop:rsa8:resource:hl7:type urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record
resource:hl7:permission hl7:prd-003 hl7:prd-005 hl7:prd-006 hl7:prd-009 hl7:prd-010 hl7:prd-012 hl7:prd-017	urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission urn:va:xacml:2.0:interop:rsa8:hl7:prd-003 urn:va:xacml:2.0:interop:rsa8:hl7:prd-005 urn:va:xacml:2.0:interop:rsa8:hl7:prd-006 urn:va:xacml:2.0:interop:rsa8:hl7:prd-009 urn:va:xacml:2.0:interop:rsa8:hl7:prd-010 urn:va:xacml:2.0:interop:rsa8:hl7:prd-012 urn:va:xacml:2.0:interop:rsa8:hl7:prd-017
resource:hl7:confidentiality-code xxx-DummyConfCode Note: <b>optional</b> this is to test unknown conf-code ignored	urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code xxx-DummyConfCode
resource:hl7:dissented-subject-id Dr. Alice	urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id Dr. Alice

environment:locality Facility A	urn:va:xacml:2.0:interop:rsa8:environment:locality Facility A

708

## 709 2.2.1.2.2 Detailed Request, PolicySets, Response

710 The following request contains all the permissions necessary to obtain a Permit access decision. If any or  
711 all permissions are removed or changed, then the access decision is expected to be Deny.

### 712 XacmlRequest-01-01: [xacml-msg-policy-index]

```
713 <?xml version="1.0" encoding="UTF-8"?>
714 <Request
715   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
716   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
717   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
718     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
719
720   <!-- ***** -->
721   <!-- Test case 1-01: Should be Perm: Dr A has all reqd perms -->
722   <!-- ***** -->
723
724   <!-- Sample request. In this case a physician is trying to access -->
725   <!-- The medical record of a patient. The record has been marked -->
726   <!-- with both the CDA and N confidentiality codes and -->
727   <!-- there is a registered consent for the record. -->
728   <Subject>
729     <Attribute
730       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
731       DataType="http://www.w3.org/2001/XMLSchema#string">
732       <AttributeValue>Dr. Alice</AttributeValue>
733     </Attribute>
734     <Attribute
735       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
736       DataType="http://www.w3.org/2001/XMLSchema#string" >
737       <AttributeValue>Facility A</AttributeValue>
738     </Attribute>
739     <Attribute
740       AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
741       DataType="http://www.w3.org/2001/XMLSchema#string">
742       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
743       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
744       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
745       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
746       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
747       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
748       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
749     </Attribute>
750   </Subject>
751   <Resource>
752     <Attribute
753       AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
754       DataType="http://www.w3.org/2001/XMLSchema#string">
755       <AttributeValue>Anthony Gurrola</AttributeValue>
756     </Attribute>
757     <Attribute
758       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"
759       DataType="http://www.w3.org/2001/XMLSchema#string">
760       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
761       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
762       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
763       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
764       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
765       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
766       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
767     </Attribute>
768     <Attribute
769       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
770       DataType="http://www.w3.org/2001/XMLSchema#string">
771       <AttributeValue>xxx-DummyConfCode</AttributeValue>
772     </Attribute>
773     <Attribute
774       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
775       DataType="http://www.w3.org/2001/XMLSchema#string">
776       <AttributeValue>Dr. Alice</AttributeValue>
777     </Attribute>
```

```

778 <Attribute
779   AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
780   DataType="http://www.w3.org/2001/XMLSchema#string">
781   <AttributeValue
782     >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
783   </AttributeValue>
784 </Attribute>
785 </Resource>
786 <Action/>
787 <Environment>
788   <Attribute
789     AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
790     DataType="http://www.w3.org/2001/XMLSchema#string" >
791     <AttributeValue>Facility A</AttributeValue>
792   </Attribute>
793 </Environment>
</Request>

```

794 **EndOfXacmlRequest-01-01**

795

796 The following PolicySet contains the Roles and Permissions for evaluating these requests.

797

798 **XacmlPolicySet-01-top-level: [xacml-msg-policy-index]**

```

799 <?xml version="1.0" encoding="UTF-8"?>
800 <PolicySet
801   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
802   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
803   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
804     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
805   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel"
806   PolicyCombiningAlgId=
807     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
808   <Description>
809     Top level policy set which combines the CDA and N confidentiality codes.
810   </Description>
811   <Target/>
812   <PolicySet
813     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:emergency"
814     PolicyCombiningAlgId=
815       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
816     <Target/>
817     <PolicySetIdReference
818       >urn:va:xacml:2.0:interop:rsa8:policysetid:emergency</PolicySetIdReference>
819   </PolicySet>
820   <PolicySet
821     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:CDA"
822     PolicyCombiningAlgId=
823       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
824     <Target>
825       <Resources>
826         <Resource>
827           <ResourceMatch
828             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
829             <AttributeValue
830               DataType="http://www.w3.org/2001/XMLSchema#string"
831               >UBA</AttributeValue>
832             <ResourceAttributeDesignator
833               AttributeId=
834                 "urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
835               DataType="http://www.w3.org/2001/XMLSchema#string"/>
836             </ResourceMatch>
837           </Resource>
838         </Resources>
839       </Target>
840     <PolicySetIdReference
841       >urn:va:xacml:2.0:interop:rsa8:policysetid:CDA</PolicySetIdReference>
842   </PolicySet>
843   <PolicySet
844     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:MA"
845     PolicyCombiningAlgId=
846       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">

```

```

847 <Target>
848   <Resources>
849     <Resource>
850       <ResourceMatch
851         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
852         <AttributeValue
853           DataType="http://www.w3.org/2001/XMLSchema#string"
854           >MA</AttributeValue>
855         <ResourceAttributeDesignator
856           AttributeId=
857             "urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
858           DataType="http://www.w3.org/2001/XMLSchema#string"/>
859         </ResourceMatch>
860       </Resource>
861     </Resources>
862   </Target>
863 <PolicySetIdReference
864   >urn:va:xacml:2.0:interop:rsa8:policysetid:MA</PolicySetIdReference>
865 <Policy
866   PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:MA:default-to-permit"
867   RuleCombiningAlgId=
868     "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
869 <Target/>
870 <Rule
871   RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA"
872   Effect="Permit">
873 <Description>
874   If a Deny was obtained for object above then set Permit by default.
875 </Description>
876 </Rule>
877 </Policy>
878 </PolicySet>
879 <PolicySet
880   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:bus-rule"
881   PolicyCombiningAlgId=
882     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
883 <Target>
884 <Resources>
885 <Resource>
886 <ResourceMatch
887   MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
888 <AttributeValue
889   DataType="http://www.w3.org/2001/XMLSchema#string"
890   >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
891 <ResourceAttributeDesignator
892   AttributeId=
893     "urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
894   DataType="http://www.w3.org/2001/XMLSchema#string"/>
895 </ResourceMatch>
896 </Resource>
897 </Resources>
898 </Target>
899 <PolicySetIdReference
900   >urn:va:xacml:2.0:interop:rsa8:policysetid:progress-note</PolicySetIdReference>
901 </PolicySet>
902 <PolicySet
903   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:N"
904   PolicyCombiningAlgId=
905     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
906 <Target/>
907 <PolicySetIdReference
908   >urn:va:xacml:2.0:interop:rsa8:policysetid:N</PolicySetIdReference>
909 <PolicySetIdReference
910   >urn:va:xacml:2.0:interop:rsa8:policysetid:N:PermCollections</PolicySetIdReference>
911 </PolicySet>
912 </PolicySet>

```

### 913 EndOfXacmlPolicySet-01-top-level

914 The policyset above, policyset-id:toplevel, is where a Request enters for processing. One can think of this  
915 narrative as having the Request in hand and going along through the policy structures and testing the  
916 Request to determine subsequent steps to take.

917 The purpose of the toplevel policysset is to perform a sequence of operations, which will govern what kind  
918 of Response will be sent in answer to this Request. For example, the first check is whether privacy  
919 constraints must be applied, which only is relevant if the Request is successful, but it is also done first  
920 because there are some conditions where the Request will be denied regardless of subsequent  
921 processing. Rather than try to explain too much up front, the narrative that follows will simply explain what  
922 is happening at each point along the way and after one reading, one can go back and look at the whole  
923 thing with more context.

924 Enter policyssetid:toplevel with a Request (ex XacmlRequest-01-01 or XacmlRequest-02-01). The policy  
925 algorithm is deny-overrides, so if we get denied along the way we are done. The Target is empty, so all  
926 Requests are governed by this policy. (\*\*please see [cautionary note] regarding the likelihood that this  
927 policy really should use the combining algorithm: ordered-deny-overrides)

928 Enter policyssetid:toplevel:CDA which also has policy algorithm deny-overrides. For this policysset to apply,  
929 the Request must satisfy the requirements of the Target, which first checks if the Request contains an  
930 attribute with a resource:confidentiality-code equal to "UBA". If so, then you need to go to  
931 policyssetid:CDA, which is below in the next use case section. If that policysset returns Deny then we are  
932 done. If it returns Permit, we continue. If we did not have a "UBA" attribute the Target was NotApplicable,  
933 so we also continue.

934 Enter policyssetid:toplevel:N. (Note: because the policyssetid:toplevel is deny-overrides, even if we come  
935 out of policyssetid:CDA with a Permit, we are not done and must continue with any subsequent policies  
936 encountered.) Policysset toplevel:N has a policy algorithm of permit-overrides, so if we get a permit along  
937 the way, we exit this policysset with a Permit. (Note: the fact that we may be entering this policysset with a  
938 Permit does not apply to this policysset because the only "Permits" that apply are those granted within the  
939 current policysset.)

940

#### 941 XacmlPolicySet-02b-N: [xacml-msg-policy-index]

```
942 <?xml version="1.0" encoding="UTF-8"?>
943 <PolicySet
944   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
945   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
946   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
947     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
948   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N"
949   PolicyCombiningAlgId=
950     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
951   <Description>
952     Policy set for evaluating the subject:role attributes.
953     This implements an RBAC policy. This policy set matches
954     subject roles and refers to permission policy sets.
955   </Description>
956   <Target/>
957   <PolicySet
958     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:physician"
959     PolicyCombiningAlgId=
960       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
961     <Target>
962       <Subjects>
963         <Subject>
964           <SubjectMatch
965             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
966             <AttributeValue
967               DataType="http://www.w3.org/2001/XMLSchema#string"
968               >urn:va:xacml:2.0:interop:rsa8:role:hl7:physician</AttributeValue>
969             <SubjectAttributeDesignator
970               AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
971               DataType="http://www.w3.org/2001/XMLSchema#string"/>
972             </SubjectMatch>
973           </Subject>
974         </Subjects>
975       </Target>
976       <PolicySetIdReference
977         >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
978     </PolicySet>
979   </PolicySet>
```

980 EndOfXacmlPolicySet-02b-N

981

982 XacmlPolicySet-02c-N-PermCollections: [xacml-msg-policy-index]

```
983 <?xml version="1.0" encoding="UTF-8"?>
984 <PolicySet
985   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
986   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
987   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
988     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
989   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:PermCollections"
990   PolicyCombiningAlgId=
991     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
992   <Description>
993     Policy set for evaluating the subject:hl7:permission attributes.
994     This implements an RBAC policy. This policy set matches
995     subject roles and refers to permission policy sets.
996   </Description>
997   <Target/>
998   <PolicySet
999     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set"
1000     PolicyCombiningAlgId=
1001       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
1002   <Target/>
1003   <PolicySet
1004     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set-0"
1005     PolicyCombiningAlgId=
1006       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1007   <Description>
1008     This PolicySet bypasses the role checking by permission and allows the
1009     automatic subject and resource matching to take place without screening out
1010     the role permissions as in the nest policy.
1011   </Description>
1012   <Target/>
1013   <PolicySetIdReference
1014     >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
1015   </PolicySet>
1016   <PolicySet
1017     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set-1"
1018     PolicyCombiningAlgId=
1019       "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1020   <Description>
1021     This PolicySet is intended to map collections of permissions to
1022     the virtual roles. This logic is not active in current demo, but thought
1023     that it might be useful if client can send in request as either named role
1024     or collection of permissions. It is left in benign state, by having inserted
1025     the previous PolicySet, which performs the same processing without screening
1026     the Targets.
1027     This PolicySet is left for reference and analysis, as it embodies the concept
1028     of virtual roles.
1029   </Description>
1030   <Target>
1031     <Subjects>
1032       <Subject>
1033         <SubjectMatch
1034           MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1035           <AttributeValue
1036             DataType="http://www.w3.org/2001/XMLSchema#string"
1037             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
1038           <SubjectAttributeDesignator
1039             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1040             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1041           </SubjectMatch>
1042         <SubjectMatch
1043           MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1044           <AttributeValue
1045             DataType="http://www.w3.org/2001/XMLSchema#string"
1046             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
1047           <SubjectAttributeDesignator
1048             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1049             DataType="http://www.w3.org/2001/XMLSchema#string"/>
```



```

1050     </SubjectMatch>
1051     <SubjectMatch
1052         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1053         <AttributeValue
1054             DataType="http://www.w3.org/2001/XMLSchema#string"
1055             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
1056         <SubjectAttributeDesignator
1057             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1058             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1059         </SubjectMatch>
1060     <SubjectMatch
1061         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1062         <AttributeValue
1063             DataType="http://www.w3.org/2001/XMLSchema#string"
1064             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
1065         <SubjectAttributeDesignator
1066             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1067             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1068         </SubjectMatch>
1069     <SubjectMatch
1070         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1071         <AttributeValue
1072             DataType="http://www.w3.org/2001/XMLSchema#string"
1073             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
1074         <SubjectAttributeDesignator
1075             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1076             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1077         </SubjectMatch>
1078     <SubjectMatch
1079         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1080         <AttributeValue
1081             DataType="http://www.w3.org/2001/XMLSchema#string"
1082             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
1083         <SubjectAttributeDesignator
1084             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1085             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1086         </SubjectMatch>
1087     <SubjectMatch
1088         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1089         <AttributeValue
1090             DataType="http://www.w3.org/2001/XMLSchema#string"
1091             >urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
1092         <SubjectAttributeDesignator
1093             AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1094             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1095         </SubjectMatch>
1096     </Subject>
1097 </Subjects>
1098 </Target>
1099 <PolicySetIdReference
1100 >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
1101 </PolicySet>
1102 </PolicySet>
1103 </PolicySet>

```

1104 **EndOfXacmlPolicySet-02c-N-PermCollections**

1105

1106 **XacmlPolicySet-03-N-RPS-med-rec-vrole: [xacml-msg-policy-index]**

```

1107 <?xml version="1.0" encoding="UTF-8"?>
1108 <PolicySet
1109     xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
1110     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1111     xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
1112         http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
1113     PolicySetId=
1114         "urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole"
1115     PolicyCombiningAlgId=
1116         "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1117     <Description>
1118         Policy set that points to the Permission PolicySet for medical record
1119         resources and actions.

```

```

1120     </Description>
1121     <Target/>
1122     <PolicySetIdReference
1123       >urn:va:xacml:2.0:interop:rsa8:policysetid:N:PPS:PRD-004</PolicySetIdReference>
1124   </PolicySet>

```

1125 **EndOfXacmlPolicySet-03-N-RPSmed-rec-vrole**

1126

1127 **XacmlPolicySet-04-N:PPS:PRD-004: [xacml-msg-policy-index]**

```

1128 <?xml version="1.0" encoding="UTF-8"?>
1129 <PolicySet
1130   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
1131   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1132   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
1133     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
1134   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:PPS:PRD-004"
1135   PolicyCombiningAlgId=
1136     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1137   <Description>
1138     Policy set for the PRD-004 permission. This permission allows
1139     access to all medical records.
1140   </Description>
1141   <Target/>
1142   <Policy
1143     PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:N:PPS:PRD-004:1"
1144     RuleCombiningAlgId=
1145       "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
1146     <Target>
1147       <Resources>
1148         <Resource>
1149           <ResourceMatch
1150             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1151             <AttributeValue
1152               DataType="http://www.w3.org/2001/XMLSchema#string"
1153               >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
1154             <ResourceAttributeDesignator
1155               AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1156               DataType="http://www.w3.org/2001/XMLSchema#string"/>
1157             </ResourceMatch>
1158           </Resource>
1159           <Resource>
1160             <ResourceMatch
1161               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1162               <AttributeValue
1163                 DataType="http://www.w3.org/2001/XMLSchema#string"
1164                 >urn:va:xacml:2.0:interop:rsa8:resource:hl7:demographics</AttributeValue>
1165               <ResourceAttributeDesignator
1166                 AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1167                 DataType="http://www.w3.org/2001/XMLSchema#string"/>
1168               </ResourceMatch>
1169             </Resource>
1170           <Resource>
1171             <ResourceMatch
1172               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1173               <AttributeValue
1174                 DataType="http://www.w3.org/2001/XMLSchema#string"
1175                 >urn:va:xacml:2.0:interop:rsa8:resource:hl7:chart</AttributeValue>
1176               <ResourceAttributeDesignator
1177                 AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1178                 DataType="http://www.w3.org/2001/XMLSchema#string"/>
1179               </ResourceMatch>
1180             </Resource>
1181           <Resource>
1182             <ResourceMatch
1183               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1184               <AttributeValue
1185                 DataType="http://www.w3.org/2001/XMLSchema#string"
1186                 >urn:va:xacml:2.0:interop:rsa8:resource:hl7:problemlist</AttributeValue>
1187               <ResourceAttributeDesignator
1188                 AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1189                 DataType="http://www.w3.org/2001/XMLSchema#string"/>

```

```

1190     </ResourceMatch>
1191 </Resource>
1192 <Resource>
1193   <ResourceMatch
1194     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1195     <AttributeValue
1196       DataType="http://www.w3.org/2001/XMLSchema#string"
1197       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:procedures</AttributeValue>
1198     <ResourceAttributeDesignator
1199       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1200       DataType="http://www.w3.org/2001/XMLSchema#string" />
1201   </ResourceMatch>
1202 </Resource>
1203 <Resource>
1204   <ResourceMatch
1205     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1206     <AttributeValue
1207       DataType="http://www.w3.org/2001/XMLSchema#string"
1208       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:laboratory</AttributeValue>
1209     <ResourceAttributeDesignator
1210       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1211       DataType="http://www.w3.org/2001/XMLSchema#string" />
1212   </ResourceMatch>
1213 </Resource>
1214 <Resource>
1215   <ResourceMatch
1216     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1217     <AttributeValue
1218       DataType="http://www.w3.org/2001/XMLSchema#string"
1219       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology</AttributeValue>
1220     <ResourceAttributeDesignator
1221       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1222       DataType="http://www.w3.org/2001/XMLSchema#string" />
1223   </ResourceMatch>
1224 </Resource>
1225 <Resource>
1226   <ResourceMatch
1227     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1228     <AttributeValue
1229       DataType="http://www.w3.org/2001/XMLSchema#string"
1230       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medications</AttributeValue>
1231     <ResourceAttributeDesignator
1232       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1233       DataType="http://www.w3.org/2001/XMLSchema#string" />
1234   </ResourceMatch>
1235 </Resource>
1236 <Resource>
1237   <ResourceMatch
1238     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1239     <AttributeValue
1240       DataType="http://www.w3.org/2001/XMLSchema#string"
1241       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:vitals</AttributeValue>
1242     <ResourceAttributeDesignator
1243       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1244       DataType="http://www.w3.org/2001/XMLSchema#string" />
1245   </ResourceMatch>
1246 </Resource>
1247 <Resource>
1248   <ResourceMatch
1249     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1250     <AttributeValue
1251       DataType="http://www.w3.org/2001/XMLSchema#string"
1252       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
1253     <ResourceAttributeDesignator
1254       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1255       DataType="http://www.w3.org/2001/XMLSchema#string" />
1256   </ResourceMatch>
1257 </Resource>
1258 <Resource>
1259   <ResourceMatch
1260     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
1261     <AttributeValue
1262       DataType="http://www.w3.org/2001/XMLSchema#string"

```

```

1263         >urn:va:xacml:2.0:interop:rsa8:resource:hl7:patientsearch</AttributeValue>
1264     <ResourceAttributeDesignator
1265         AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1266         DataType="http://www.w3.org/2001/XMLSchema#string"/>
1267     </ResourceMatch>
1268 </Resource>
1269 </Resources>
1270 </Target>
1271 <Rule
1272     RuleId="urn:va:xacml:2.0:interop:rsa8:policy:N:PPS:PRD-004:1:rule:1"
1273     Effect="Permit">
1274     <Condition>
1275
1276         <!-- Returns true iff the first argument is a subset of the second argument -->
1277         <!-- i.e. the permissions required by the resource must be a -->
1278         <!-- subset of the permissions supplied by the subject -->
1279
1280         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-subset">
1281
1282             <!-- 1st argument: returns the values of all Attributes with -->
1283             <!-- DataType="http://www.w3.org/2001/XMLSchema#string" and -->
1284             <!-- AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission" -->
1285             <ResourceAttributeDesignator
1286                 DataType="http://www.w3.org/2001/XMLSchema#string"
1287                 AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"/>
1288
1289             <!-- 2nd argument: returns the values of all Attributes with -->
1290             <!-- DataType="http://www.w3.org/2001/XMLSchema#string" and -->
1291             <!-- AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission" -->
1292             <SubjectAttributeDesignator
1293                 DataType="http://www.w3.org/2001/XMLSchema#string"
1294                 AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"/>
1295
1296         </Apply>
1297     </Condition>
1298 </Rule>
1299 <Rule
1300     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:N:PPS:PRD-004:1:rule:2"
1301     Effect="Deny">
1302     <Description>
1303         If a Permit was not obtained above then set Deny by default.
1304     </Description>
1305 </Rule>
1306 </Policy>
1307 </PolicySet>

```

1308 **EndOf XacmlPolicySet-04-N:PPS:PRD-004**

1309

1310 **2.2.2 Details: HL7 Patient Consent Directive**

1311 **2.2.2.1 Scenarios for HL7 Patient Consent Directive**

1312 **2.2.2.1.1 DEMO HL7 Consent Directive Access Control (no consent restrictions)**

1313

1314 Initial State / Pre-condition:

- 1315
- Dr. Bob has all related permissions to read a medical record (c.f. Table 1)
  - Patient has not specified any constraints that would limit Dr. Bob's access to their record.
- 1316
- 1317

1318 Scenario:

- 1319
- Dr. Bob attempts to view the medical records for Anthony Gurrola.
- 1320

1321 RESULT:

- 1322
- Dr. Bob is able to access the medical record including Anthony Gurrola's sensitive data.
- 1323

State	Name	HL7 Code	Definition
OFF	Masked access (MA)	MA	Access to a record is restricted to users or roles specified by the subject of the record. Users who are not authorized to access the record will not be notified that the record is masked.
OFF	User based access (UBA)	UBA	Access to a record is restricted to identified users

**Table 3 – Confidentiality codes identified to complete use case**

1324

1325 **2.2.2.1.2 DEMO HL7 Consent Directive Access Control (Consent Restriction applied)**

1326

1327

1328 Initial State / Pre-condition:

- 1329
- Dr. Bob has all related permissions to read a medical record.
- 1330

1331 Scenario:

- 1332
- With Patient set to Anthony Gurrola and using the 'Set Patient Elections' screen,
  - security administrator creates a patient directive preventing Dr. Bob from viewing Gurrola's medical record.
  - Note, the security administrator (instead of the patient) is creating the directive to simplify the demonstration.
  - Dr. Bob attempts to view the medical records for Anthony Gurrola.
- 1333
- 1334
- 1335
- 1336
- 1337
- 1338

1339 RESULT:

1340  
1341

- Dr. Bob is unable to access the medical record

State	Name (Code)	HL7 Code	Definition
OFF	Masked access (MA)	MA	Access to a record is restricted to users or roles specified by the subject of the record. Users who are not authorized to access the record will not be notified that the record is masked.
ON	User based access (UBA)	UBA	Access to a record is restricted to identified users

**Table 4 – Confidentiality codes identified to complete use case**

1342  
1343  
1344

1345 **2.2.2.2 Detailed Data: HL7 Patient Consent Directive**

1346 **2.2.2.2.1 Detailed Data Elements**

1347 Note: Only new elements for this use case are added here. Other elements in the requests should be  
 1348 same as in previous Detailed Data Elements sections, especially the first: section 2.2.2.1.1

1349 **Use Case 2: HL7 Patient Consent Directive Data Elements:** [\[xacml-msg-policy-index\]](#)

Variable AttributeId Value(s)	Full Variable AttributeId URN Full Value URN(s)
subject:role hl7:physician (Note: this is alternative to permission set used in prev use case)	urn:oasis:names:tc:xacml:2.0:subject:role urn:va:xacml:2.0:interop:rsa8:role:hl7:physician
subject:subject-id Dr. Alice	urn:oasis:names:tc:xacml:1.0:subject:subject-id Dr. Alice
resource:hl7:confidentiality-code UBA	urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code UBA
resource:hl7:dissented-subject-id Dr. Alice	urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id Dr. Alice

1350

1351

## 1352 2.2.2.2 Detailed Request, PolicySets, Response

### 1353 XacmlRequest-02-01: [xacml-msg-policy-index]

```
1354 <?xml version="1.0" encoding="UTF-8"?>
1355 <Request
1356   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
1357   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1358   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
1359     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
1360
1361   <!-- ***** -->
1362   <!-- Test case 2-01: Should be Deny + Obl: has role but needs perms -->
1363   <!-- ***** -->
1364
1365   <!-- Sample request. In this case a physician is trying to access -->
1366   <!-- The medical record of a patient. The record has been marked -->
1367   <!-- with both the CDA and U confidentiality codes and -->
1368   <!-- there is a registered consent for the record. -->
1369   <Subject>
1370     <Attribute
1371       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1372       DataType="http://www.w3.org/2001/XMLSchema#string">
1373       <AttributeValue>Dr. Alice</AttributeValue>
1374     </Attribute>
1375     <Attribute
1376       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
1377       DataType="http://www.w3.org/2001/XMLSchema#string" >
1378       <AttributeValue>Facility A</AttributeValue>
1379     </Attribute>
1380     <Attribute
1381       AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
1382       DataType="http://www.w3.org/2001/XMLSchema#string">
1383       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:role:hl7:physician</AttributeValue>
1384     </Attribute>
1385   </Subject>
1386   <Resource>
1387     <Attribute
1388       AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
1389       DataType="http://www.w3.org/2001/XMLSchema#string">
1390       <AttributeValue>
1391         >Anthony Gurrola</AttributeValue>
1392     </Attribute>
1393     <Attribute
1394       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
1395       DataType="http://www.w3.org/2001/XMLSchema#string">
1396       <AttributeValue>UBA</AttributeValue>
1397     </Attribute>
1398     <Attribute
1399       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
1400       DataType="http://www.w3.org/2001/XMLSchema#string">
1401       <AttributeValue>Dr. Alice</AttributeValue>
1402     </Attribute>
1403     <Attribute
1404       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1405       DataType="http://www.w3.org/2001/XMLSchema#string">
1406       <AttributeValue>
1407         >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
1408     </Attribute>
1409   </Resource>
1410   <Action/>
1411   <Environment>
1412     <Attribute
1413       AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
1414       DataType="http://www.w3.org/2001/XMLSchema#string" >
1415       <AttributeValue>Facility A</AttributeValue>
1416     </Attribute>
1417   </Environment>
1418 </Request>
```

### 1419 EndOfXacmlRequest-02-01



1420 The above request is for role-based access, the details of which in the policy vs the comparison to the  
1421 collection of required resource permissions is TBD.

1422 The following request contains the subject collection of permissions to compare with the resource  
1423 collection of required permissions as in XacmlRequest-01-01. The difference here is that a real  
1424 confidentiality-code UBA is provided which should result in this user being denied access since the  
1425 subject-id will match the dissented-subject-id.

1426 **XacmlRequest-02-02: [xacml-msg-policy-index]**

```
1427 <?xml version="1.0" encoding="UTF-8"?>
1428 <Request
1429   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
1430   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1431   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
1432     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
1433
1434   <!-- ***** -->
1435   <!-- Test case 2-02: Should be Deny + Obl: Dr A is on dissented list -->
1436   <!-- ***** -->
1437
1438   <!-- Sample request. In this case a physician is trying to access -->
1439   <!-- The medical record of a patient. The record has been marked -->
1440   <!-- with both the CDA and N confidentiality codes and -->
1441   <!-- there is a registered consent for the record. -->
1442   <Subject>
1443     <Attribute
1444       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1445       DataType="http://www.w3.org/2001/XMLSchema#string">
1446       <AttributeValue>Dr. Alice</AttributeValue>
1447     </Attribute>
1448     <Attribute
1449       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
1450       DataType="http://www.w3.org/2001/XMLSchema#string" >
1451       <AttributeValue>Facility A</AttributeValue>
1452     </Attribute>
1453     <Attribute
1454       AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:h17:permission"
1455       DataType="http://www.w3.org/2001/XMLSchema#string">
1456       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-003</AttributeValue>
1457       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-005</AttributeValue>
1458       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-006</AttributeValue>
1459       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-009</AttributeValue>
1460       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-010</AttributeValue>
1461       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-012</AttributeValue>
1462       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-017</AttributeValue>
1463     </Attribute>
1464   </Subject>
1465   <Resource>
1466     <Attribute
1467       AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
1468       DataType="http://www.w3.org/2001/XMLSchema#string">
1469       <AttributeValue>Anthony Gurrola</AttributeValue>
1470     </Attribute>
1471     <Attribute
1472       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:h17:permission"
1473       DataType="http://www.w3.org/2001/XMLSchema#string">
1474       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-003</AttributeValue>
1475       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-005</AttributeValue>
1476       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-006</AttributeValue>
1477       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-009</AttributeValue>
1478       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-010</AttributeValue>
1479       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-012</AttributeValue>
1480       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:h17:prd-017</AttributeValue>
1481     </Attribute>
1482     <Attribute
1483       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:h17:confidentiality-code"
1484       DataType="http://www.w3.org/2001/XMLSchema#string">
1485       <AttributeValue>UBA</AttributeValue>
1486     </Attribute>
1487     <Attribute
1488       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:h17:dissented-subject-id"
```

```

1489     DataType="http://www.w3.org/2001/XMLSchema#string">
1490       <AttributeValue>Dr. Alice</AttributeValue>
1491     </Attribute>
1492   <Attribute
1493     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1494     DataType="http://www.w3.org/2001/XMLSchema#string">
1495     <AttributeValue
1496       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
1497     </AttributeValue>
1498   </Attribute>
1499 </Resource>
1500 <Action/>
1501 <Environment>
1502   <Attribute
1503     AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
1504     DataType="http://www.w3.org/2001/XMLSchema#string" >
1505     <AttributeValue>Facility A</AttributeValue>
1506   </Attribute>
1507 </Environment>
</Request>

```

1508 EndOfXacmlRequest-02-02

1509

1510 XacmlPolicySet-02a-CDA: [xacml-msg-policy-index]

```

1511 <?xml version="1.0" encoding="UTF-8"?>
1512 <PolicySet
1513   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
1514   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1515   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
1516     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
1517   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:CDA"
1518   PolicyCombiningAlgId=
1519     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1520 <Description>
1521   Policy set for the UBA confidentiality code.
1522 </Description>
1523 <Target/>
1524 <Policy
1525   PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:CDA"
1526   RuleCombiningAlgId=
1527     "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
1528 <Target/>
1529 <Rule
1530   RuleId="urn:va:xacml:2.0:interop:rsa8:rule:CDA:1"
1531   Effect="Permit">
1532 <Description>
1533   If the access subject is NOT one of those users which consent has
1534   been removed, then permit.
1535 </Description>
1536 <Target/>
1537 <Condition>
1538 <!-- True if hl7:dissented-subject-id NOT EQUAL TO subject:subject-id -->
1539 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
1540 <!-- True if hl7:dissented-subject-id EQUAL TO subject:subject-id -->
1541 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
1542 <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
1543 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
1544 <SubjectAttributeDesignator
1545   AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1546   DataType="http://www.w3.org/2001/XMLSchema#string"/>
1547 </Apply>
1548 <ResourceAttributeDesignator
1549   AttributeId=
1550     "urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
1551   DataType="http://www.w3.org/2001/XMLSchema#string"/>
1552 </Apply>
1553 </Apply>
1554 </Condition>
1555 </Rule>
1556 <Rule
1557   RuleId="urn:va:xacml:2.0:interop:rsa8:rule:CDA:2"
1558   Effect="Deny">

```

```
1559     <Description>
1560         If a Permit was not obtained above then set Deny by default.
1561     </Description>
1562 </Rule>
1563 <Obligations>
1564     <!-- These obligations provide specific instructions to PEP in the response -->
1565     <!-- This obligation instructs the PEP the request was denied based on -->
1566     <!-- privacy constraints set by the patient. -->
1567     <Obligation
1568         ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:privacy:constraint"
1569         FulfillOn="Deny" />
1570 </Obligations>
1571 </Policy>
1572 </PolicySet>
```

### 1573 **EndOfXacmlPolicySet-02a-CDA**

1574 The policy above, policy-id:CDA, the first rule uses function:string-equal to test whether the user's  
1575 subject-id is equal to the subject-id of the resource attribute hl7:dissented-subject-id. If they are equal,  
1576 then, because of the function:not that is outside of the function:string-equal, the first Rule's Condition will  
1577 evaluate to False and the Rule's effect will be ignored and processing will continue with the 2<sup>nd</sup> Rule.

1578 However, if the strings are not equal, then the first rule's Condition will evaluate to True and the Rule's  
1579 Effect, Permit, will be applied, and the user will escape the Policy with a Permit because the policy is  
1580 permit-overrides.

1581 If the strings are equal, then the 2<sup>nd</sup> Rule will be evaluated, which always gives a Deny, which is the  
1582 intended effect of the hl7:dissented-subject-id attribute, which is to Deny access to named users.

1583 Finally, the Obligations section is checked to see if any Obligations need to be returned. Since there is an  
1584 Obligation with FulfillOn="Deny", this Obligation needs to be returned with the Response to the PEP,  
1585 which will cause the PEP to apply privacy constraints, which, in this case notifies the PEP that the user  
1586 has been denied because of the privacy constraint and the PEP can take any action that can be  
1587 configured based on that information.

1588

1589 **2.2.3 Details: Attribute based rules**

1590 **2.2.3.1 Scenarios: Attribute based rules**

1591 **2.2.3.1.1 Security Business Rule (Access Denied)**

1592 Business Rule:

- 1593
- The author of a progress note is able to read, change, and delete a progress note that they have initiated.
- 1594
- All other physicians are unable to view the progress note until it is digitally signed by the author.
- 1595
- 1596

1597 Initial State / Pre-condition:

- 1598
- Both Dr. Alice and Dr. Bob have all related permissions to read a medical record.
- 1599
- There are no constraints on the record imposed by a patient directive.
- 1600

1601 Scenario:

- 1602
- Dr. Alice begins a progress note for Anthony Gurrola, but has not digitally signed the progress note,
- 1603
- Dr. Bob attempts to access Anthony Gurrola's record.
- 1604
- 1605

1606 RESULT:

- 1607
- Dr. Bob is unable to view the unsigned progress note written by Dr. Alice
- 1608
- 1609

1610 **2.2.3.1.2 Security Business Rule (Access Granted)**

1611 Business Rule:

- 1612
- The author of a progress note is able to read, change, and delete a progress note that they have initiated.
- 1613
- All other physicians are unable to view the progress note until it is digitally signed by the author.
- 1614
- 1615

1616 Initial State / Pre-condition:

- 1617
- Both Dr. Alice and Dr. Bob have all related permissions to read a medical record.
- 1618
- There are no constraints on the record imposed by a patient directive.
- 1619

1620 Scenario:

- 1621
- Dr. Alice completes and digitally signs a progress note for Anthony Gurrola.
- 1622
- Dr. Bob attempts to access Anthony Gurrola's record.
- 1623

1624 RESULT:

- 1625
- Dr. Bob is able to view the signed progress note written by Dr. Alice

1626 **2.2.3.2 Detailed Data: Attribute Based Rules**

1627 **2.2.3.2.1 Detailed Data Elements**

1628 Note: Only new elements for this use case are added here. Other elements in the requests should be  
 1629 same as in previous Detailed Data Elements sections, especially the first: section 2.2.2.1.1

1630 **Use Case 3: HL7 Attribute Based Rules Data Elements** [[xacml-msg-policy-index](#)]

Variable AttributeId Value(s)	Full Variable AttributeId URN Full Value URN(s)
subject:subject-id Dr. Alice	urn:oasis:names:tc:xacml:1.0:subject:subject-id Dr. Alice
resource:hl7:progress-note:signed	urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:signed False
resource:hl7:progress-note: author-subject-id Dr. Bob	urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:author-subject-id Dr. Bob
resource:hl7:type resource:hl7:progress-note	urn:va:xacml:2.0:interop:rsa8:resource:hl7:type urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note

1631

1632 **2.2.3.2.2 Detailed Request, Policy Sets, Response**

1633 **XacmlRequest-03-01:** [[xacml-msg-policy-index](#)]

```

1634 <?xml version="1.0" encoding="UTF-8"?>
1635 <Request
1636   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
1637   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1638   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
1639   http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
1640
1641   <!-- ***** -->
1642   <!-- Test case 3-01: Should be Deny + Obl: signed = F, DrA not author -->
1643   <!-- ***** -->
1644
1645   <!-- Sample request. In this case a physician is trying to access -->
1646   <!-- The medical record of a patient. The record has been marked -->
1647   <!-- with both the CDA and N confidentiality codes and -->
1648   <!-- there is a registered consent for the record. -->
1649   <Subject>
1650     <Attribute
1651       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1652       DataType="http://www.w3.org/2001/XMLSchema#string">
1653       <AttributeValue>Dr. Alice</AttributeValue>
1654     </Attribute>
1655     <Attribute
1656       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
1657       DataType="http://www.w3.org/2001/XMLSchema#string" >
1658       <AttributeValue>Facility A</AttributeValue>
1659     </Attribute>
1660     <Attribute
1661       AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1662       DataType="http://www.w3.org/2001/XMLSchema#string">
1663       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
1664       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
1665       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
1666       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
1667       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
1668       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
1669       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
1670     </Attribute>
  
```

```

1671 </Subject>
1672 <Resource>
1673   <Attribute
1674     AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
1675     DataType="http://www.w3.org/2001/XMLSchema#string">
1676     <AttributeValue>Anthony Gurrola</AttributeValue>
1677   </Attribute>
1678   <Attribute
1679     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"
1680     DataType="http://www.w3.org/2001/XMLSchema#string">
1681     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
1682     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
1683     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
1684     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
1685     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
1686     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
1687     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
1688   </Attribute>
1689   <Attribute
1690     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
1691     DataType="http://www.w3.org/2001/XMLSchema#string">
1692     <AttributeValue>xxx-DummyConfCode</AttributeValue>
1693   </Attribute>
1694   <Attribute
1695     AttributeId=
1696     "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:signed"
1697     DataType="http://www.w3.org/2001/XMLSchema#string">
1698     <AttributeValue>False</AttributeValue>
1699   </Attribute>
1700   <Attribute
1701     AttributeId=
1702     "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:author-subject-id"
1703     DataType="http://www.w3.org/2001/XMLSchema#string">
1704     <AttributeValue>Dr. Bob</AttributeValue>
1705   </Attribute>
1706   <Attribute
1707     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1708     DataType="http://www.w3.org/2001/XMLSchema#string">
1709     <AttributeValue
1710       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
1711   </Attribute>
1712 </Resource>
1713 <Action/>
1714 <Environment>
1715   <Attribute
1716     AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
1717     DataType="http://www.w3.org/2001/XMLSchema#string" >
1718     <AttributeValue>Facility A</AttributeValue>
1719   </Attribute>
1720 </Environment>
1721 </Request>

```

## 1722 EndOfXacmlRequest-03-01

1723 The above request should fail because the progress note signed attribute is "False" AND the requestor  
 1724 subject-id, "Dr. Alice" is not the author, who is "Dr. Bob". If either signed is set to "True" or the requestor  
 1725 subject-id is changed to "Dr. Bob" then the request should succeed.

1726

## 1727 XacmlPolicySet-02d-prog-note: [\[xacml-msg-policy-index\]](#)

```

1728 <?xml version="1.0" encoding="UTF-8"?>
1729 <PolicySet
1730   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
1731   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1732   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
1733     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
1734   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:progress-note"
1735   PolicyCombiningAlgId=
1736     "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
1737   <Description>
1738     Policy set for the business rule for unsigned progress notes.
1739   </Description>

```

```

1740 <Target/>
1741 <Policy
1742   PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:progress-note"
1743   RuleCombiningAlgId=
1744     "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
1745   <Target/>
1746   <Rule
1747     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:sig"
1748     Effect="Permit">
1749     <Description>
1750       If the progress-note is signed allow any user to see it. If not signed
1751       then only author may see it.
1752     </Description>
1753     <Target/>
1754     <Condition>
1755       <!-- True if resource:hl7:progress-note:signed EQUAL TO True -->
1756       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
1757         <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
1758         <AttributeValue
1759           DataType="http://www.w3.org/2001/XMLSchema#string"
1760           >True</AttributeValue>
1761         <ResourceAttributeDesignator
1762           AttributeId=
1763             "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:signed"
1764           DataType="http://www.w3.org/2001/XMLSchema#string"/>
1765         </Apply>
1766       </Condition>
1767     </Rule>
1768   <Rule
1769     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:author"
1770     Effect="Permit">
1771     <Description>
1772       If a Permit was not obtained then subject must be author.
1773     </Description>
1774     <Target/>
1775     <Condition>
1776       <!-- True if hl7:dissented-subject-id EQUAL TO subject:subject-id -->
1777       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
1778         <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
1779         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
1780           <SubjectAttributeDesignator
1781             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1782             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1783           </Apply>
1784           <ResourceAttributeDesignator
1785             AttributeId=
1786               "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:author-subject-id"
1787             DataType="http://www.w3.org/2001/XMLSchema#string"/>
1788           </Apply>
1789         </Condition>
1790       </Rule>
1791     <Rule
1792       RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:deny-sig"
1793       Effect="Deny">
1794       <Description>
1795         If a Permit was not obtained above then set Deny by default.
1796       </Description>
1797     </Rule>
1798   <Obligations>
1799     <!-- These obligations provide specific instructions to PEP in the response -->
1800     <!-- This obligation informs the PEP access denied unsigned non-author -->
1801     <Obligation
1802       ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:deny:unsigned:non-author"
1803       FulfillOn="Deny"/>
1804   </Obligations>
1805 </Policy>
1806 </PolicySet>

```

1807 **EndOfXacmlPolicySet-02d-prog-note**

1808

1809 **2.2.4 Details: Emergency access Use Case**

1810 **2.2.4.1 Scenarios for Emergency access Use Case**

1811 **2.2.4.1.1 Scenario: Emergency Access (no emergency declared)**

1812 Initial state:

- 1813 • Dr. Charlie from Facility B does not have permission to access electronic patient records from  
1814 Facility A.
- 1815 • Dr. Charlie has permissions to declare an emergency which would subsequently allow him  
1816 access records at Facility A during the emergency.

1817

1818 Scenario:

- 1819 • Patient from Facility A presents to the family clinic in Facility B with severe chest pain.
- 1820 • Dr. Charlie attempts to access the medical history of the patient from Facility A.

1821

1822 RESULT:

- 1823 • Dr. Charlie is unable to access the medical history of the patient from Facility A.

1824

1825 **2.2.4.1.2 Scenario: Emergency Access (emergency declared)**

1826 Initial state:

- 1827 • Dr. Charlie from Facility B does not have permission to access electronic patient records from  
1828 Facility A.
- 1829 • Dr. Charlie has permissions to declare an emergency which would subsequently allow him  
1830 access records at Facility A during the emergency.

1831

1832 Scenario:

- 1833 • Patient from Facility A presents to the family clinic in Facility B with severe chest pain.
- 1834 • Dr. Charlie declares an emergency.

1835

1836 RESULT 1:

- 1837 • Dr. Charlie successfully declares an emergency.

1838

1839

Emergency Access Granted	HL7 Permission Code	HL7 Permission Title
√	PEA-001	Declare Emergency Access

**Table 05 – RBAC Permissions identified to complete Use Case**

1840

1841 RESULT 2:

- 1842 • Dr. Charlie is able to view the patient's history from facility A despite the lack of required  
1843 permissions.



- 1844 • This scenario has been described with the phrase “patient safety trumps enterprise security.”
- 1845 • Comprehensive logging of Dr. Charlie’s activities (above and beyond standard logging) is
- 1846 triggered due to the declaration of an emergency.

1847

Facility B - Permissions Granted	HL7 permission code	HL7 Permission Title
	PRD-006	Patient Identification and Lookup
	PRD-017	Review (Read) Progress Notes
	PRD-012	Review (Read) Past Visits
	PRD-003	Review (Read) Medical History
	PRD-005	Review (Read) Vital signs/Patient Measurements
	PRD-009	Review (Read) Current Directory of Provider Information
	PRD-011	Review Patient Allergies
	PRD-010	Review (Read) Patient Medications
<b>Table 6 – RBAC Permissions granted to Dr. Charlie at Facility B (Prior to Emergency Declaration)</b>		

1848

1849

1850 **2.2.4.2 Detailed Data: Emergency Access**

1851 **2.2.4.2.1 Detailed Data Elements: Emergency Access**

1852 Note: Only new elements for this use case are added here. Other elements in the requests should be  
 1853 same as in previous Detailed Data Elements sections, especially the first: section 2.2.2.1.1

1854 **Use Case 4: Emergency Access Data Elements [xacml-msg-policy-index]**

Variable AttributeId Value(s)	Full Variable AttributeId URN Full Value URN(s)
subject:locality Facility B	urn:oasis:names:tc:xacml:1.0:subject:locality Facility B
environment:locality Facility A	urn:va:xacml:2.0:interop:rsa8:environment:locality Facility A
subject:hl7:permission hl7:pea-001	urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission urn:va:xacml:2.0:interop:rsa8:hl7:pea-001
resource:hl7:permission hl7:pea-001	urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission urn:va:xacml:2.0:interop:rsa8:hl7:pea-001

1855

1856

1857 **2.2.4.2.2 Detailed Request, Policy Sets, Response: Emergency Access**

1858

1859 **XacmlRequest-04-01: [xacml-msg-policy-index]**

```
1860 <?xml version="1.0" encoding="UTF-8"?>
1861 <Request
1862   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
1863   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1864   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
1865     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
1866
1867   <!-- ***** -->
1868   <!-- Test case 4-01: Should be Deny: Dr C not from Facility A -->
1869   <!-- ***** -->
1870
1871   <!-- Sample request. In this case a physician is trying to access -->
1872   <!-- The medical record of a patient. Because the physician is from -->
1873   <!-- a different facility (locality) the request should be rejected -->
1874   <Subject>
1875     <Attribute
1876       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
1877       DataType="http://www.w3.org/2001/XMLSchema#string">
1878       <AttributeValue>Dr. Charlie</AttributeValue>
1879     </Attribute>
1880     <Attribute
1881       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
1882       DataType="http://www.w3.org/2001/XMLSchema#string" >
1883       <AttributeValue>Facility B</AttributeValue>
1884     </Attribute>
1885     <Attribute
1886       AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
1887       DataType="http://www.w3.org/2001/XMLSchema#string">
1888     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
1889     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
1890     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
1891     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
1892     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
1893     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
1894     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
1895     </Attribute>
1896   </Subject>
1897   <Resource>
1898     <Attribute
1899       AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
1900       DataType="http://www.w3.org/2001/XMLSchema#string">
1901       <AttributeValue>Anthony Gurrola</AttributeValue>
1902     </Attribute>
1903     <Attribute
1904       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"
1905       DataType="http://www.w3.org/2001/XMLSchema#string">
1906       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
1907     </Attribute>
1908     <Attribute
1909       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
1910       DataType="http://www.w3.org/2001/XMLSchema#string">
1911       <AttributeValue>xxx-DummyConfCode</AttributeValue>
1912     </Attribute>
1913     <Attribute
1914       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
1915       DataType="http://www.w3.org/2001/XMLSchema#string">
1916       <AttributeValue>Dr. Alice</AttributeValue>
1917     </Attribute>
1918     <Attribute
1919       AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
1920       DataType="http://www.w3.org/2001/XMLSchema#string">
1921     <AttributeValue
1922       >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
1923     </Attribute>
1924   </Resource>
1925   <Action/>
1926 </Environment>
```

1927  
1928  
1929  
1930  
1931  
1932  
1933

```
<Attribute
  AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
  DataType="http://www.w3.org/2001/XMLSchema#string" >
  <AttributeValue>Facility A</AttributeValue>
</Attribute>
</Environment>
</Request>
```

1934 **EndOfXacmlRequest-04-01**

1935 The request above is the initial request from Dr. Charlie from facility B attempting to access the patient  
1936 record at facility A, which is denied because Dr. Charlie does not have access permissions for Facility A.

1937

1938 **XacmlRequest-04-02: [xacml-msg-policy-index]**

1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995

```
<?xml version="1.0" encoding="UTF-8"?>
<Request
  xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
    http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
  <!-- ***** -->
  <!-- Test case 4-02: Should be Perm + Obl: Dr A has emergency perm -->
  <!-- ***** -->

  <!-- Sample request. In this case a physician is trying to access -->
  <!-- The medical record of a patient. The record has been marked -->
  <!-- with both the CDA and N confidentiality codes and -->
  <!-- there is a registered consent for the record. -->
  <Subject>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Dr. Charlie</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
      DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility B</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:pea-001</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Anthony Gurrola</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:pea-001</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>xxx-DummyConfCode</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Dr. Alice</AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue
```

```

1996         >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
1997     </Attribute>
1998 </Resource>
1999 <Action/>
2000 <Environment>
2001     <Attribute
2002         AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
2003         DataType="http://www.w3.org/2001/XMLSchema#string" >
2004         <AttributeValue>Facility A</AttributeValue>
2005     </Attribute>
2006 </Environment>
2007 </Request>

```

2008 **EndOfXacmlRequest-04-02**

2009

2010 **XacmlPolicySet-02f-emergency:** [[xacml-msg-policy-index](#)]

```

2011 <?xml version="1.0" encoding="UTF-8"?>
2012 <PolicySet
2013     xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
2014     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2015     xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
2016         http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
2017     PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:emergency"
2018     PolicyCombiningAlgId=
2019         "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
2020 <Description>
2021     Policy set to allow emergency access for non-facility subjects.
2022     Returns Deny if user not from supported facility AND does not have emergency perm
2023     Returns Permit if not from supported facility AND not denied access
2024     Returns NotApplicable if plain old user from supported facility
2025 </Description>
2026 <Target/>
2027 <Policy
2028     PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:emergency"
2029     RuleCombiningAlgId=
2030         "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
2031 <Target/>
2032 <Rule
2033     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:emergency:deny"
2034     Effect="Deny">
2035 <Description>
2036     If the subject is not from a supported facility AND
2037     if the subject does not have emergency permission THEN Deny access.
2038 </Description>
2039 <Target/>
2040 <Condition>
2041 <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality -->
2042 <!-- AND if hl7:pea-001 NOT EQUAL TO ANYOF subject:hl7:permission -->
2043 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
2044 <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality -->
2045 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
2046 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2047 <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
2048 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
2049 <SubjectAttributeDesignator
2050     AttributeId=
2051         "urn:oasis:names:tc:xacml:1.0:subject:locality"
2052     DataType="http://www.w3.org/2001/XMLSchema#string"/>
2053 </Apply>
2054 <EnvironmentAttributeDesignator
2055     AttributeId=
2056         "urn:va:xacml:2.0:interop:rsa8:environment:locality"
2057     DataType="http://www.w3.org/2001/XMLSchema#string"/>
2058 </Apply>
2059 </Apply>
2060 <!-- True if hl7:pea-001 NOT EQUAL TO ANYOF subject:hl7:permission -->
2061 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
2062 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2063 <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
2064 <AttributeValue
2065     DataType="http://www.w3.org/2001/XMLSchema#string"

```

```

2066         >urn:va:xacml:2.0:interop:rsa8:hl7:pea-001</AttributeValue>
2067     <SubjectAttributeDesignator
2068         AttributeId=
2069         "urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
2070         DataType="http://www.w3.org/2001/XMLSchema#string"/>
2071     </Apply>
2072 </Apply>
2073 </Apply>
2074 </Condition>
2075 </Rule>
2076 <Rule
2077     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:emergency:permit"
2078     Effect="Permit">
2079     <Description>
2080         If a Deny was not obtained above AND subject not part of a supported
2081         facility then subject must have emergency permission.
2082     </Description>
2083     <Target/>
2084     <Condition>
2085         <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality -->
2086         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
2087             <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2088                 <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
2089                 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
2090                     <SubjectAttributeDesignator
2091                         AttributeId=
2092                         "urn:oasis:names:tc:xacml:1.0:subject:locality"
2093                         DataType="http://www.w3.org/2001/XMLSchema#string"/>
2094                     </Apply>
2095                     <EnvironmentAttributeDesignator
2096                         AttributeId=
2097                         "urn:va:xacml:2.0:interop:rsa8:environment:locality"
2098                         DataType="http://www.w3.org/2001/XMLSchema#string"/>
2099                     </Apply>
2100                 </Apply>
2101             </Condition>
2102         </Rule>
2103     <Obligations>
2104         <!-- These obligations provide specific instructions to PEP in the response -->
2105         <!-- This obligation informs the PEP user granted emergency access -->
2106         <Obligation
2107             ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:emergency:permit"
2108             FulfillOn="Permit"/>
2109     </Obligations>
2110 </Policy>
2111 </PolicySet>

```

2112 **EndOfXacmlPolicySet-02f-emergency**

2113

2114

2115 **2.2.5 Data filtering Use Case**

2116 **2.2.5.1 Scenarios for Data Filtering Use Case**

2117 **2.2.5.1.1 Scenario: Data Filtering (all permissions granted)**

2118

2119 Initial State / Pre-condition:

- 2120 • Dr. Bob has all related permissions to read a medical record.

2121

2122 Scenario:

- 2123 • With Patient set to Anthony Gurrola and using the ‘Set Patient Elections’ screen on the demo application,
- 2124
- 2125 • a security administrator creates a patient directive (e.g. using a written request by Mr. Gurrola)
- 2126 preventing Dr. Bob from viewing his radiology record.

2127

2128 RESULT:

- 2129 • Dr. Bob is able to access the patient Anthony Gurrola’s medical record except the portion
- 2130 containing his radiology record.

2131

State	Name (Code)	Definition
ON	Masked access (MA)	Access to a record is restricted to users or roles specified by the subject of the record. Users who are not authorized to access the record will not be notified that the record is masked.
OFF	User based access (UBA)	Access to a record is restricted to identified users

**Table 7 – Confidentiality codes identified to complete Use Case**

2132

2133

2134 *Note to developers:* The patient election in the data filtering use case will constrain the permission  
 2135 granted to a physician to view data of a specific patient. This is shown by specifically naming “Dr. Bob”  
 2136 as not allowed to view specific information, e.g. radiology in the demonstration application. We then  
 2137 demonstrate how the radiology panel of the chart is not shown in the application (data filtering).

2138

2139 **2.2.5.2 Detailed Data: Data Filtering**

2140 **2.2.5.2.1 Detailed Data Elements**

2141 Note: Only new elements for this use case are added here. Other elements in the requests should be  
 2142 same as in previous Detailed Data Elements sections, especially the first: section 2.2.2.1.1

2143 **Use Case 5: Data Filtering Data Elements** [[xacml-msg-policy-index](#)]

Variable AttributeId Value(s)	Full Variable AttributeId URN Full Value URN(s)
subject:subject-id Dr. Alice	urn:oasis:names:tc:xacml:1.0:subject:subject-id Dr. Alice
resource:hl7:confidentiality-code MA	urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code MA
resource:hl7:radiology: dissented-subject-id Dr. Alice	urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology:dissented-subject-id Dr. Alice

2144

2145 **2.2.5.2.2 Detailed Request, Policy Sets, Response**

2146

2147 **XacmlRequest-05-01:** [[xacml-msg-policy-index](#)]

```

2148 <?xml version="1.0" encoding="UTF-8"?>
2149 <Request
2150   xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
2151   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2152   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
2153     http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
2154
2155   <!-- ***** -->
2156   <!-- Test case 5-01: Should be Perm + Obl: Dr A is on dissented list -->
2157   <!-- ***** -->
2158
2159   <!-- Sample request. In this case a physician is trying to access -->
2160   <!-- The medical record of a patient. The record has been marked -->
2161   <!-- with both the CDA and N confidentiality codes and -->
2162   <!-- there is a registered consent for the record. -->
2163   <Subject>
2164     <Attribute
2165       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
2166       DataType="http://www.w3.org/2001/XMLSchema#string">
2167       <AttributeValue>Dr. Alice</AttributeValue>
2168     </Attribute>
2169     <Attribute
2170       AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
2171       DataType="http://www.w3.org/2001/XMLSchema#string" >
2172       <AttributeValue>Facility A</AttributeValue>
2173     </Attribute>
2174     <Attribute
2175       AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
2176       DataType="http://www.w3.org/2001/XMLSchema#string">
2177       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
2178       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
2179       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
2180       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
2181       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
2182       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
2183       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>

```

```

2184     </Attribute>
2185 </Subject>
2186 <Resource>
2187   <Attribute
2188     AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
2189     DataType="http://www.w3.org/2001/XMLSchema#string">
2190     <AttributeValue>Anthony Gurrola</AttributeValue>
2191   </Attribute>
2192   <Attribute
2193     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"
2194     DataType="http://www.w3.org/2001/XMLSchema#string">
2195     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
2196     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
2197     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
2198     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
2199     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
2200     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
2201     <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
2202   </Attribute>
2203   <Attribute
2204     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
2205     DataType="http://www.w3.org/2001/XMLSchema#string">
2206     <AttributeValue>MA</AttributeValue>
2207   </Attribute>
2208   <Attribute
2209     AttributeId=
2210     "urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology:dissented-subject-id"
2211     DataType="http://www.w3.org/2001/XMLSchema#string">
2212     <AttributeValue>Dr. Alice</AttributeValue>
2213   </Attribute>
2214   <Attribute
2215     AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
2216     DataType="http://www.w3.org/2001/XMLSchema#string">
2217     <AttributeValue
2218     >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
2219   </Attribute>
2220 </Resource>
2221 <Action/>
2222 <Environment>
2223   <Attribute
2224     AttributeId="urn:va:xacml:2.0:interop:rsa8:environment:locality"
2225     DataType="http://www.w3.org/2001/XMLSchema#string">
2226     <AttributeValue>Facility A</AttributeValue>
2227   </Attribute>
2228 </Environment>
2229 </Request>

```

## 2230 EndOfXacmlRequest-05-01

2231 In the above sample request, Dr. Alice is the subject and also appears on the patient's do not allow  
2232 access to named users list. Therefore, Dr. Alice should have an Obligation returned with an otherwise  
2233 permitted access, which says do not show radiology information.

2234

## 2235 XacmlPolicySet-02e-MA: [[xacml-msg-policy-index](#)]

```

2236 <?xml version="1.0" encoding="UTF-8"?>
2237 <PolicySet
2238   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
2239   xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance
2240   http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
2241   PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:MA"
2242   PolicyCombiningAlgId=
2243   "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
2244   <Description>
2245     Policy set for the MA confidentiality code.
2246   </Description>
2247   <Target/>
2248   <Policy
2249     PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:MA"
2250     RuleCombiningAlgId=
2251     "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
2252

```



```

2253 <Target/>
2254 <Rule
2255   RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA:1"
2256   Effect="Deny">
2257   <Description>
2258     If the access subject is NOT one of those users which consent has
2259     been removed, then deny.
2260     Note: there is reverse logic here because the Obligation that denies
2261     access to the user for this object must be issued when the user has
2262     obtained a Permit. So, the caller of this policy must know to reverse
2263     sense as well.
2264   </Description>
2265   <Target/>
2266   <Condition>
2267     <!-- True if hl7:radiology:dissented-subject-id NOTEQUALTO subject:subject-id -->
2268     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
2269       <!-- True if hl7:radiology:dissented-subject-id EQUALTO subject:subject-id -->
2270       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
2271         <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
2272         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
2273           <SubjectAttributeDesignator
2274             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
2275             DataType="http://www.w3.org/2001/XMLSchema#string"/>
2276         </Apply>
2277         <ResourceAttributeDesignator
2278           AttributeId=
2279             "urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology:dissented-subject-id"
2280           DataType="http://www.w3.org/2001/XMLSchema#string"/>
2281         </Apply>
2282       </Apply>
2283     </Condition>
2284   </Rule>
2285   <Rule
2286     RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA:2"
2287     Effect="Permit">
2288     <Description>
2289       If a Deny was not obtained above then set Permit by default.
2290     </Description>
2291   </Rule>
2292   <Obligations>
2293     <!-- These obligations provide specific instructions to PEP in the response -->
2294     <!-- This obligation instructs the PEP to apply privacy constraints to -->
2295     <!-- user's responsibility for the data. -->
2296     <Obligation
2297       ObligationId=
2298         "urn:va:xacml:2.0:interop:rsa8:obligation:ma:privacy:constraint:radiology"
2299       FulfillOn="Permit"/>
2300   </Obligations>
2301 </Policy>
2302 </PolicySet>

```

2303 **EndOfXacmlPolicySet-02e-MA**

2304

2305

---

## A. Acknowledgements

2306 The following individuals have participated in the creation of this specification by virtue of their  
2307 contributions to the planning, preparation, and execution of the XACML 2.0 Interop event at the RSA  
2308 Conference 2008 and are gratefully acknowledged:

2309 **Sponsors:**

2310 Jane Harnad, OASIS

2311 Dee Schur, OASIS

2312 **Participants:**

2313 Erik Rissanen, Axiomatics

2314 Babak Sadighi, Axiomatics

2315 Wayne Delisser, BEA Systems, Inc.

2316 Cynthia Ding, BEA Systems, Inc.

2317 Hal Lockhart, BEA Systems, Inc. (Lead/Moderator)

2318 Denis Pilipchuk, BEA Systems, Inc.

2319 Anil Tappetla, Cisco Systems, Inc.

2320 Ed Coyne, Department of Veterans Affairs

2321 Mike Davis, Department of Veterans Affairs

2322 Duane DeCouteau, Department of Veterans Affairs (Edmond Scientific Company)

2323 Suzanne Gonzales-Webb, Department of Veterans Affairs (SAIC)

2324 David Staggs, Department of Veterans Affairs (SAIC)

2325 Kathleen Connor, Fox Systems

2326 John Moehrke, GE Healthcare

2327 Craig Forster, IBM

2328 Maryann Hondo, IBM

2329 Vernon Murdoch, IBM

2330 Anthony Nadalin, IBM

2331 Saket Kaushik, Oracle Corporation

2332 Rich Levinson, Oracle Corporation

2333 Prateek Mishra, Oracle Corporation

2334 Hari Sastry, Oracle Corporation

2335 Anil Saldhana, Red Hat

2336 Dilli Dorai, Sun Microsystems

2337 Luis Bernardo, Symlabs

2338 Sampo Kellomaki, Symlabs

2339

2340 [Participant Name, Affiliation | Individual Member]

2341

---

## **B. Non-Normative Text**

2343

---

## C. Revision History

2344

[optional; should not be included in OASIS Standards]

2345

<b>Revision</b>	<b>Date</b>	<b>Editor</b>	<b>Changes Made</b>
[Rev number]	[Rev Date]	[Modified By]	[Summary of Changes]

2346

2347