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

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].

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

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

1.1.3 Policy Update and Retrieval

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

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

1.2 Terminology

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

1.3 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[XACML20] T. Moses, *XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0*, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf , OASIS Standard, 1 February 2005.

[SAML-XACML20] A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML 2.0* ***Errata***, http://www.oasis-open.org/committees/download.php/15447/xacml-2.0-saml-errata-wd.zip, Working Draft 01, 17 November 2005.

[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

[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

[HL7-PERM] HL7 Security Technical Committee, HL7 Version 3 Standard: Role-based Access Control Healthcare Permission Catalog, (Available through http://www.hl7.org/library/standards.cfm), Release 1, Designation: ANSI/HL7 V3 RBAC, R1-2008, Approval Date 2/20/2008.

[HL7-CONSENT] HL7 Consent Related Vocabulary confidentialityCodes Recommendation, http://lists.oasis-open.org/archives/xacml-demo-tech/200712/doc00003.doc, from project submission: http://lists.oasis-open.org/archives/xacml-demo-tech/200712/msg00015.html

1.4 Non-Normative References

[superceded-by-errata**]** A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML 2.0*, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf, OASIS Standard, 1 February 2005 (original spec, superceded by errata spec).

[SAML-XACML20V2] A. Anderson, H. Lockhart, *SAML 2.0 profile of XACML Version 2*, 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 draft covers all versions of XACML).

[XACML-RBAC] A. Anderson, ***Core and hierarchical role based access control (RBAC) profile of XACML v2.0***, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf, OASIS Standard, 1 February 2005.

[INTEROP-01] R. Levinson, D. Pilipchuk, ***XACML 2.0 Interop Scenarios***, http://www.oasis-open.org/committees/download.php/24475/xacml-2.0-core-interop-draft-12-04.doc, Working Draft, 22 June 2007.

[HL7-RoleEng] HL7 Security Technical Committee, ***HL7 Role Based Access Control (RBAC) Role Engineering Process***, http://www.hl7.org/library/ (search rbac), Version 1.3, March 1, 2008.

[HITSP] Healthcare Information Technology Standards Panel (HITSP) at www.hitsp.org.

1.5 Interoperability Use Cases for XACML 2.0

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

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

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

The following diagram shows overall use case environment:

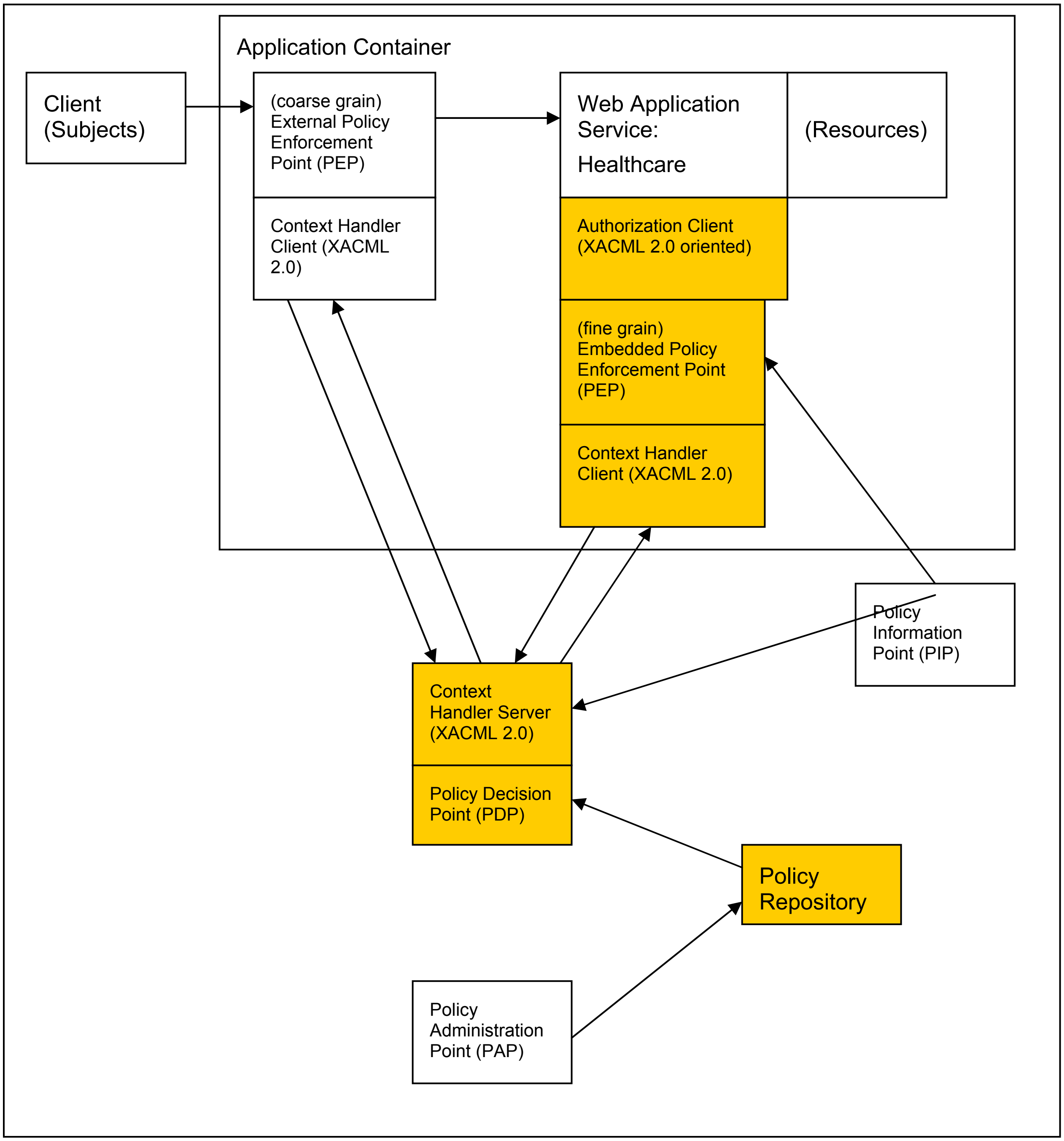

Application Container
Client (Subjects)
(coarse grain) External Policy Enforcement Point (PEP)
Context Handler Client (XACML 2.0)
Web Application Service: Healthcare
(Resources)
Authorization Client (XACML 2.0 oriented)
(fine grain) Embedded Policy Enforcement Point (PEP)
Context Handler Client (XACML 2.0)
Policy Information Point (PIP)
Context Handler Server (XACML 2.0)
Policy Decision Point (PDP)
Policy Repository
Policy Administration Point (PAP)

Figure 1

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

- the Authorization Client/PIP-client,
- the PEP/context-handler-client,

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

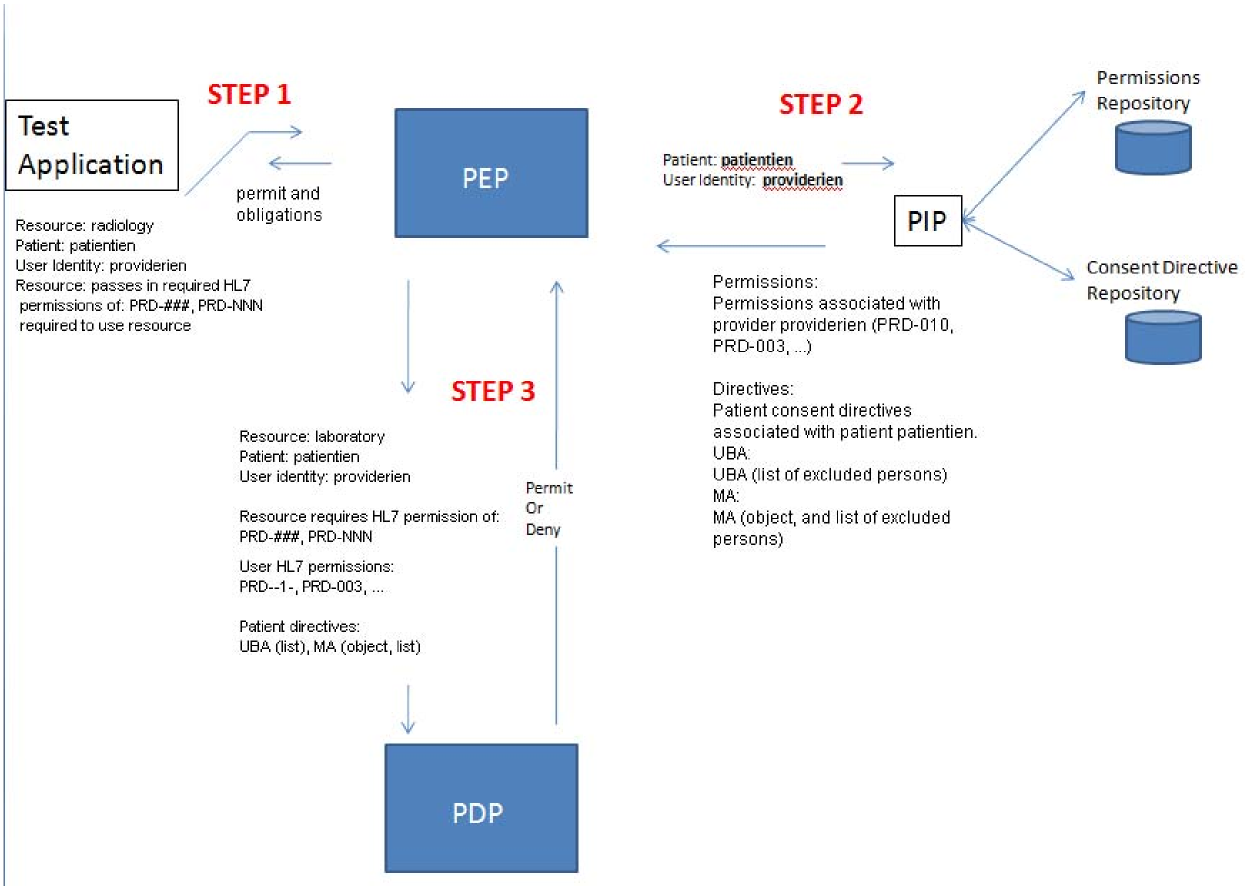
All other modules shown on the diagram are assumed to be part of the common environment

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.

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

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



Test Application
STEP 1
PEP
STEP 2
Permissions Repository
Patient: patientien
User Identity: providerien
PIP
permit and obligations
Resource: radiology
Patient: patientien
User Identity: providerien
Resource: passes in required HL7 permissions of: PRD-###, PRD-NNN required to use resource
Consent Directive Repository
Permissions:
Permissions associated with provider providerien (PRD-010, PRD-003, ...)
Directives:
Patient consent directives associated with patient patientien.
UBA:
UBA (list of excluded persons)
MA:
MA (object, and list of excluded persons)
STEP 3
Resource: laboratory
Patient: patientien
User identity: providerien
Resource requires HL7 permission of: PRD-###, PRD-NNN
User HL7 permissions: PRD--1-, PRD-003, ...
Patient directives: UBA (list), MA (object, list)
Permit Or Deny
PDP

Figure 2. Example interaction between components

In step 1, user access to a resource (e.g. patient record or application functionality) within the healthcare application triggers a call to the PEP. The application identifies the resource being accessed and passes the patient identifier (patientEIN), the application user identity (providerEIN) and the permissions required to access the resource. In this model the application subject matter experts determine the set of permissions that the user must possess to access the resource.

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

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

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

1.5.1 Healthcare Fine Grain Authorization Use Cases

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

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

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

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

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

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

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

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

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

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

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

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

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

1.5.1.4 Use Case: Fine Grain Emergency Override Obligations

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

1.5.1.5 Use Case: Fine Grain Data Filtering Obligations

TBD

1.5.2 Use Case: Coarse Grain Authorization Decision Request/Response

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

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

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

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

1.5.3 Use Case: Policy Exchange

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

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

2 Use Cases: Healthcare: Fine Grained Authorization

2.1 Introduction to the Healthcare Application

> 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.

2.1.1 Flexibility provided by XACML

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

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

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

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

2.1.2 Use of Virtual Roles

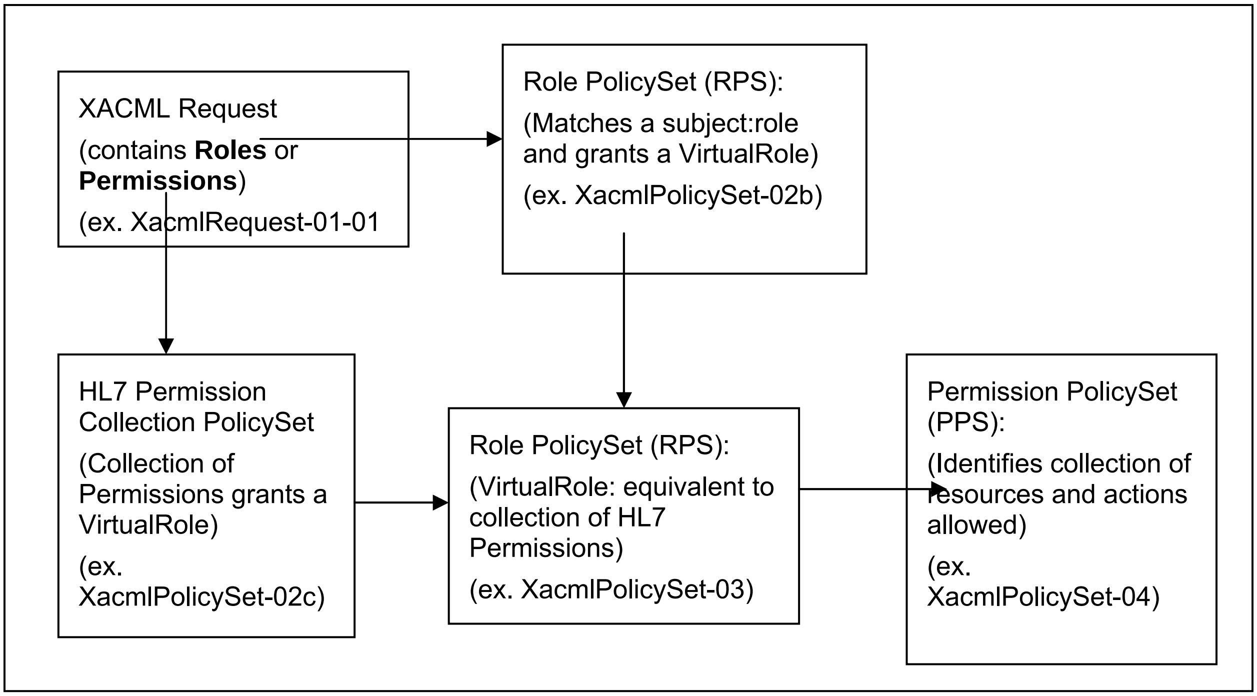
Analysis of the VA Healthcare requirements determined that a particular variation of Role Based Access Control (RBAC) was required. It has been determined by the HL7 Security Technical Committee that for inter-organization purposes that collections of specific commonly understood HL7 Permission Identifiers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

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

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

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

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


XACML Request
(contains **Roles** or **Permissions**)
(ex. XacmlRequest-01-01
Role PolicySet (RPS):
(Matches a subject:role and grants a VirtualRole)
(ex. XacmlPolicySet-02b)
HL7 Permission Collection PolicySet
(Collection of Permissions grants a VirtualRole)
(ex. XacmlPolicySet-02c)
Role PolicySet (RPS):
(VirtualRole: equivalent to collection of HL7 Permissions)
(ex. XacmlPolicySet-03)
Permission PolicySet (PPS):
(Identifies collection of resources and actions allowed)
(ex. XacmlPolicySet-04)

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

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

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

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

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

2.1.3 Scenario structure

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

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

Associated with each scenario are 3 sets of data:

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

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

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

2.1.3.1 Client request and response data

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

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

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

Note: in general, a XACML Request can contain Subject, Action, Resource, and Environment attributes. In these scenarios, all these attributes are being provided within the client request. Any resource attributes that the client needs will be provided from the Web Application in the previous response, which sets the framework for the next request. Typically, in real world situations one might expect Subject attributes to be provided by an identity management module at authentication time or by a PIP from the ContextHandler, one might expect Resource attributes to be obtained from the application itself which might play a role in executing the xacml request for authorization, or by a PIP from the ContextHandler. The net result is that eventually the attributes must be provided to successfully get authorized, and since this interop is focussed on the vendors interfacing across the xacml req/rsp interface, we are simply providing a mix of attributes from the Subject and Resource using the above mechanisms to avoid complications with potentially proprietary attribute accessing schemes outside the scope of the Interop.

2.1.3.2 XACML Request and Response

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

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

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

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

Sample SOAP SAML XACML Request wrapper: [xacml-msg-policy-index]

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
    xmlns:soapenv ="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <xacml-samlp:XACMLAuthzDecisionQuery
        xmlns:xacml-samlp="urn:oasis:xacml:2.0:saml:protocol:schema:os"
        ID="_e064bd912f83c1544fea110307000acf"
        IssueInstant="2007-05-21T22:00:36Z"
        Version="2.0">
      <xacml-context:Request
          xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
        <!-- See [XACML-Request-01] for sample content of this element -->
      </xacml-context:Request>
    </xacml-samlp:XACMLAuthzDecisionQuery>
  </soapenv:Body>
</soapenv:Envelope>
```

The request message above contains 3 protocol levels:

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

Sample SOAP SAML XACML response wrapper: [xacml-msg-policy-index]

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

```
        <samlp:StatusCode
          Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
      </samlp:Status>
      <saml:Assertion
          xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
          Version="2.0"
          ID="A12345603"
          IssueInstant="2007-05-09T00:00:01Z">
        <saml:Issuer>xacml.interop.com</saml:Issuer>
        <saml:Statement
            xmlns:xacml-saml="urn:oasis:xacml:2.0:saml:assertion:schema:os"
            xsi:type="xacml-saml:XACMLAuthzDecisionStatementType">
          <xacml-context:Response
              xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os">
            <!-- See [XACML-Response-01] for sample content of this element -->
          </xacml-context:Response>
        </saml:Statement>
      </saml:Assertion>
    </samlp:Response>
  </soapenv:Body>
</soapenv:Envelope>
```

The response message above contains 3 protocol levels:

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

2.1.3.3 XACML Policy

This section describes the policy and how it is applied to the xacml Request and Response. Policy data references attributes by urn, which enables seamless mapping between the policies and the XACML Requests and Responses.

The policies given are intended to represent one possible implementation of the policies that represent the rules. It is expected that each vendor will implement their own policies using the rules that are given and the resulting policies should be the functional equivalent of the samples given with each scenario.

TBD: It is expected that there will be some testing required before these sample policies are considered to be totally correct, and this document will be updated with the final working version of the policies.

2.1.4 Use Case Policy Pseudo-code

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

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

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

```
if ( ! (request.subject.locality == request.environment.locality) )
  if ( ! ("hl7.pea-001" ==
          any-of(request.subject.hl7.permission)) )
    Result = Deny
  else
    Result = Permit
    response.add(Obligation(emergency.override, ffon-permit))
  end
end
if ( ! Result == Deny )
  if (request.hl7.conf-code == "UBA")
    if ( ! (request.subject.subject-id ==
            any-of(request.resource.hl7.dissented-subject-id) ) )
      Result = Permit
    else
      Result = Deny
      response.add(Obligation(privacy.constraint, ffon-deny)
    end
  end
end
if ( ! (Result == Deny )
  if (request.hl7.conf-code == "MA")
    if (request.subject.subject-id ==
        any-of(request.resource.hl7.object.1.dissented-subject-id) )
      Result = Permit
      response.add(Obligation(privacy.constraint.object.1, ffon-permit)
    end
      ...
    if (request.subject.subject-id ==
        request.resource.hl7.object.n.dissented-subject-id)
      Result = Permit
      response.add(Obligation(privacy.constraint.object.n, ffon-permit)
    end
  end
end
if ( ! (Result == Deny))
  if (request.resource.type == "resource.hl7.progress-note)
    if (request.resource.progress-note.signed == false)
      if ( ! (request.subject.subject-id ==
              anyof(request.resource.progress-note.author-subject-id) ) )
        Result = Deny
      end
    end
  end
end
if ( ! (Result == Deny))
  if (request.subject.role == role.hl7.physician)
    check-vrole-permissions()
  end
  check-vrole-permissions() // unscreened permission comparison
  if ( ! (Result == Permit) ) // screened permission comparison
    if ( (hl7.prd-003 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-005 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-006 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-009 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-010 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-012 == subset-of(subject.hl7.permission[n-values]) &&
         (hl7.prd-017 == subset-of(subject.hl7.permission[n-values]) )
      check-vrole-permissions()
    end
  end
  // need to add here a deny if no permit found
end

check-vrole-permissions()
  if (request.resource.type == "hl7-medical-record")
    if ( request.resource.hl7.permission[m-values] ==
         subset-of(subject.hl7.permission[n-values] )
       Result = Permit
    end
  end
```

```
    return
```

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

The functionality encapsulated in each logic block is as follows:

- 1st logic block: emergency access logic
- 2nd logic block: patient consent UBA logic
- 3rd logic block: patient consent MA (data filtering) logic
- 4th logic block: business rules logic block
- 5th logic block: HL7 role/permissions logic block

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

2.2 Detailed Description of Fine Grained Authorization Use Cases

Index to sample messages and policies:

1. **Pseudo code represents the basic logic of all 5 use cases**
2. **Sample SOAP SAML XACML Request wrapper**
3. **Sample SOAP SAML XACML response wrapper**
4. **Use Case 1: HL7 Role/Permissions Data Elements**
5. **XacmlRequest-01-01**
6. **XacmlPolicySet-01-top-level**
7. **XacmlPolicySet-02b-N**
8. **XacmlPolicySet-02c-N-PermCollections**
9. **XacmlPolicySet-03-N-RPS-med-rec-vrole**
10. **XacmlPolicySet-04-N:PPS:PRD-004**
11. **Use Case 2: HL7 Patient Consent Directive Data Elements**
12. **XacmlRequest-02-01**
13. **XacmlRequest-02-02**
14. **XacmlPolicySet-02a-CDA**
15. **Use Case 3: HL7 Attribute Based Rules Data Elements**
16. **XacmlRequest-03-01**
17. **XacmlPolicySet-02d-prog-note**
18. **Use Case 4: Emergency Access Data Elements**
19. **XacmlRequest-04-01**
20. **XacmlRequest-04-02**
21. **XacmlPolicySet-02f-emergency**
22. **Use Case 5: Data Filtering Data Elements**
23. **XacmlRequest-05-01**
24. **XacmlPolicySet-02e-MA**

The above list of links is intended to be used to anchor quasi-random navigation within the document. Most of the destination on the above list include a backpointer link that comes back to this list. So, one can start in the above list, click a link, look around, and then click the backpointer to come back here.

(may not seem that great, but it can be a lot easier than scrolling up and down looking for things)

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

2.2.1 Details: HL7 Role/Permission

2.2.1.1 Scenarios for HL7 Role/Permission

2.2.1.1.1 DEMO HL7 Permission Access Control (related permissions granted)

Initial State / Pre-condition:

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

Result:

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

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

2.2.1.1.2 DEMO HL7 Permission Access Control (no permissions granted)

Initial State / Pre-condition:

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

RESULT:

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

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 2 – RBAC Permissions identified to complete use case

2.2.1.2 Detailed Data: HL7 Role/Permissions

2.2.1.2.1 Detailed Data Elements

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

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: **optional** 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: **optional** 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

2.2.1.2.2 Detailed Request, PolicySets, Response

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

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

```
<?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 1-01: Should be Perm: Dr A has all reqd perms          -->
  <!-- **************************************************************** -->

  <!-- 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. Alice</AttributeValue>
    </Attribute>
    <Attribute
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
        DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility A</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:prd-003</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:prd-003</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-01-01

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Top level policy set which combines the CDA and N confidentiality codes.
  </Description>
  <Target/>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:emergency"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Target/>
    <PolicySetIdReference
      >urn:va:xacml:2.0:interop:rsa8:policysetid:emergency</PolicySetIdReference>
  </PolicySet>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:CDA"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >UBA</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId=
                  "urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <PolicySetIdReference
      >urn:va:xacml:2.0:interop:rsa8:policysetid:CDA</PolicySetIdReference>
  </PolicySet>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:MA"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >MA</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId=
                 "urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <PolicySetIdReference
      >urn:va:xacml:2.0:interop:rsa8:policysetid:MA</PolicySetIdReference>
    <Policy
        PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:MA:default-to-permit"
        RuleCombiningAlgId=
          "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
      <Target/>
      <Rule
          RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA"
          Effect="Permit">
        <Description>
          If a Deny was obtained for object above then set Permit by default.
        </Description>
      </Rule>
    </Policy>
  </PolicySet>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:bus-rule"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId=
                  "urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <PolicySetIdReference
      >urn:va:xacml:2.0:interop:rsa8:policysetid:progress-note</PolicySetIdReference>
  </PolicySet>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:toplevel:N"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
    <Target/>
    <PolicySetIdReference
      >urn:va:xacml:2.0:interop:rsa8:policysetid:N</PolicySetIdReference>
    <PolicySetIdReference
     >urn:va:xacml:2.0:interop:rsa8:policysetid:N:PermCollections</PolicySetIdReference>
  </PolicySet>
</PolicySet>
```

EndOfXacmlPolicySet-01-top-level

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

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

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

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

Enter policysetid:toplevel:N. (Note: because the policysetid:toplevel is deny-overrides, even if we come out of policysetid:CDA with a Permit, we are not done and must continue with any subsequent policies encountered.) Policyset toplevel:N has a policy algorithm of permit-overrides, so if we get a permit along the way, we exit this policyset with a Permit. (Note: the fact that we may be entering this policyset with a Permit does not apply to this policyset because the only “Permits” that apply are those granted within the current policyset.)

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for evaluating the subject:role attributes.
      This implements an RBAC policy. This policy set matches
      subject roles and refers to permission policy sets.
  </Description>
  <Target/>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:physician"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:role:hl7:physician</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
    </Target>
    <PolicySetIdReference
   >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
  </PolicySet>
</PolicySet>
```

EndOfXacmlPolicySet-02b-N

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:PermCollections"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for evaluating the subject:hl7:permission attributes.
      This implements an RBAC policy. This policy set matches
      subject roles and refers to permission policy sets.
  </Description>
  <Target/>
  <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
   <Target/>
   <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set-0"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Description>
      This PolicySet bypasses the role checking by permission and allows the
      automatic subject and resource matching to take place without screening out
      the role permissions as in the nest policy.
    </Description>
    <Target/>
    <PolicySetIdReference
   >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
   </PolicySet>
   <PolicySet
      PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:med-rec-perm-set-1"
      PolicyCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
    <Description>
      This PolicySet is intended to map collections of permissions to
      the virtual roles. This logic is not active in current demo, but thought
      that it might be useful if client can send in request as either named role
      or collection of permissions. It is left in benign state, by having inserted
      the previous PolicySet, which performs the same processing without screening
      the Targets.
      This PolicySet is left for reference and analysis, as it embodies the concept
      of virtual roles.
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-003</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
    </Target>
    <PolicySetIdReference
   >urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole</PolicySetIdReference>
   </PolicySet>
  </PolicySet>
</PolicySet>
```

EndOfXacmlPolicySet-02c-N-PermCollections

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId=
      "urn:va:xacml:2.0:interop:rsa8:policysetid:N:RPS:med-rec-vrole"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set that points to the Permission PolicySet for medical record
      resources and actions.
  </Description>
  <Target/>
  <PolicySetIdReference
    >urn:va:xacml:2.0:interop:rsa8:policysetid:N:PPS:PRD-004</PolicySetIdReference>
</PolicySet>
```

EndOfXacmlPolicySet-03-N-RPSmed-rec-vrole

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:N:PPS:PRD-004"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for the PRD-004 permission. This permission allows
      access to all medical records.
  </Description>
  <Target/>
  <Policy
      PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:N:PPS:PRD-004:1"
      RuleCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:demographics</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:chart</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:problemlist</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:procedures</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:laboratory</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medications</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:vitals</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
             >urn:va:xacml:2.0:interop:rsa8:resource:hl7:patientsearch</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:policy:N:PPS:PRD-004:1:rule:1"
        Effect="Permit">
      <Condition>

        <!-- Returns true iff the first argument is a subset of the second argument -->
        <!-- i.e. the permissions required by the resource must be a                -->
        <!--      subset of the permissions supplied by the subject                 -->

        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-subset">

          <!-- 1st argument: returns the values of all Attributes with              -->
          <!-- DataType="http://www.w3.org/2001/XMLSchema#string" and               -->
          <!-- AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"  -->
          <ResourceAttributeDesignator
              DataType="http://www.w3.org/2001/XMLSchema#string"
              AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:permission"/>

          <!-- 2nd argument: returns the values of all Attributes with              -->
          <!-- DataType="http://www.w3.org/2001/XMLSchema#string" and               -->
          <!-- AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"   -->
          <SubjectAttributeDesignator
              DataType="http://www.w3.org/2001/XMLSchema#string"
              AttributeId="urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"/>

        </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:N:PPS:PRD-004:1:rule:2"
        Effect="Deny">
      <Description>
          If a Permit was not obtained above then set Deny by default.
      </Description>
    </Rule>
  </Policy>
</PolicySet>
```

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

2.2.2 Details: HL7 Patient Consent Directive

2.2.2.1 Scenarios for HL7 Patient Consent Directive

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

Initial State / Pre-condition:

- 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.

Scenario:

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

RESULT:

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

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

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

Initial State / Pre-condition:

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

Scenario:

- 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.

RESULT:

- 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

2.2.2.2 Detailed Data: HL7 Patient Consent Directive

2.2.2.2.1 Detailed Data Elements

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

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

2.2.2.2.2 Detailed Request, PolicySets, Response

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

```
<?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 2-01: Should be Deny + Obl: has role but needs perms   -->
  <!-- **************************************************************** -->

  <!-- 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 U 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. Alice</AttributeValue>
     </Attribute>
    <Attribute
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
        DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility A</AttributeValue>
    </Attribute>
     <Attribute
         AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
         DataType="http://www.w3.org/2001/XMLSchema#string">
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:role:hl7:physician</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:confidentiality-code"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>UBA</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
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-02-01

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

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

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

```
<?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 2-02: Should be Deny + Obl: Dr A is on dissented list  -->
  <!-- **************************************************************** -->

  <!-- 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. Alice</AttributeValue>
     </Attribute>
    <Attribute
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
        DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility A</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
     </Attribute>
    <Attribute
        AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
         DataType="http://www.w3.org/2001/XMLSchema#string">
       <AttributeValue>UBA</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
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-02-02

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

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:CDA"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for the UBA confidentiality code.
  </Description>
  <Target/>
  <Policy
      PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:CDA"
      RuleCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
    <Target/>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:CDA:1"
        Effect="Permit">
      <Description>
        If the access subject is NOT one of those users which consent has
        been removed, then permit.
      </Description>
      <Target/>
      <Condition>
       <!-- True if hl7:dissented-subject-id NOT EQUAL TO subject:subject-id -->
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
        <!-- True if hl7:dissented-subject-id EQUAL TO subject:subject-id  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <SubjectAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Apply>
          <ResourceAttributeDesignator
              AttributeId=
                "urn:va:xacml:2.0:interop:rsa8:resource:hl7:dissented-subject-id"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
       </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:CDA:2"
        Effect="Deny">
      <Description>
           If a Permit was not obtained above then set Deny by default.
      </Description>
    </Rule>
    <Obligations>
      <!-- These obligations provide specific instructions to PEP in the response -->
      <!-- This obligation instructs the PEP the request was denied based on      -->
      <!-- privacy constraints set by the patient.                                -->
      <Obligation
          ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:privacy:constraint"
          FulfillOn="Deny"/>
    </Obligations>
  </Policy>
</PolicySet>
```

EndOfXacmlPolicySet-02a-CDA

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

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

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

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

2.2.3 Details: Attribute based rules

2.2.3.1 Scenarios: Attribute based rules

2.2.3.1.1 Security Business Rule (Access Denied)

Business Rule:

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

Initial State / Pre-condition:

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

Scenario:

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

RESULT:

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

2.2.3.1.2 Security Business Rule (Access Granted)

Business Rule:

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

Initial State / Pre-condition:

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

Scenario:

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

RESULT:

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

2.2.3.2 Detailed Data: Attribute Based Rules

2.2.3.2.1 Detailed Data Elements

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

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

2.2.3.2.2 Detailed Request, Policy Sets, Response

XacmlRequest-03-01: [xacml-msg-policy-index]

```
<?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 3-01: Should be Deny + Obl: signed = F, DrA not author -->
  <!-- **************************************************************** -->

  <!-- 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. Alice</AttributeValue>
    </Attribute>
    <Attribute
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
        DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility A</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:progress-note:signed"
        DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>False</AttributeValue>
    </Attribute>
    <Attribute
        AttributeId=
        "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:author-subject-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>Dr. Bob</AttributeValue>
    </Attribute>
    <Attribute
        AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:type"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-03-01

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

XacmlPolicySet-02d-prog-note: [xacml-msg-policy-index]

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:progress-note"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for the business rule for unsigned progress notes.
  </Description>
  <Target/>
  <Policy
      PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:progress-note"
      RuleCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
    <Target/>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:sig"
        Effect="Permit">
      <Description>
        If the progress-note is signed allow any user to see it. If not signed
        then only author may see it.
      </Description>
      <Target/>
      <Condition>
        <!-- True if resource:hl7:progress-note:signed EQUAL TO True  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string"
                >True</AttributeValue>
          <ResourceAttributeDesignator
              AttributeId=
                "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:signed"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:author"
        Effect="Permit">
      <Description>
          If a Permit was not obtained then subject must be author.
      </Description>
      <Target/>
      <Condition>
        <!-- True if hl7:dissented-subject-id EQUAL TO subject:subject-id  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <SubjectAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Apply>
          <ResourceAttributeDesignator
           AttributeId=
            "urn:va:xacml:2.0:interop:rsa8:resource:hl7:progress-note:author-subject-id"
           DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:progress-note:deny-sig"
        Effect="Deny">
      <Description>
          If a Permit was not obtained above then set Deny by default.
      </Description>
    </Rule>
    <Obligations>
      <!-- These obligations provide specific instructions to PEP in the response -->
      <!-- This obligation informs the PEP access denied unsigned non-author      -->
      <Obligation
        ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:deny:unsigned:non-author"
        FulfillOn="Deny"/>
    </Obligations>
  </Policy>
</PolicySet>
```

EndOfXacmlPolicySet-02d-prog-note

2.2.4 Details: Emergency access Use Case

2.2.4.1 Scenarios for Emergency access Use Case

2.2.4.1.1 Scenario: Emergency Access (no emergency declared)

Initial state:

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

Scenario:

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

RESULT:

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

2.2.4.1.2 Scenario: Emergency Access (emergency declared)

Initial state:

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

Scenario:

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

RESULT 1:

- Dr. Charlie successfully declares an emergency.

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

Table 05 – RBAC Permissions identified to complete Use Case

RESULT 2:

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

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

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

Table 6 – RBAC Permissions granted to Dr. Charlie at Facility B (Prior to Emergency Declaration)

2.2.4.2 Detailed Data: Emergency Access

2.2.4.2.1 Detailed Data Elements: Emergency Access

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

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

2.2.4.2.2 Detailed Request, Policy Sets, Response: Emergency Access

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

```
<?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 4-01: Should be Deny: Dr C not from Facility A         -->
  <!-- **************************************************************** -->

  <!-- Sample request. In this case a physician is trying to access   -->
  <!-- The medical record of a patient. Because the physcian is from  -->
  <!-- a different facility (locality) the request should be rejected -->
  <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:prd-003</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
      <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:prd-006</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
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
```

```
    <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>
```

EndOfXacmlRequest-04-01

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

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

```
<?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
```

```
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-04-02

XacmlPolicySet-02f-emergency: [xacml-msg-policy-index]

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:emergency"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set to allow emergency access for non-facility subjects.
      Returns Deny if user not from supported facility AND does not have emergency perm
      Returns Permit if not from supported facility AND not denied access
      Returns NotApplicable if plain old user from supported facility
  </Description>
  <Target/>
  <Policy
      PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:emergency"
      RuleCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Target/>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:emergency:deny"
        Effect="Deny">
      <Description>
        If the subject is not from a supported facility AND
.       if the subject does not have emergency permission THEN Deny access.
      </Description>
      <Target/>
      <Condition>
       <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality  -->
       <!--  AND if hl7:pea-001 NOT EQUAL TO ANYOF subject:hl7:permission     -->
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
           <SubjectAttributeDesignator
               AttributeId=
                 "urn:oasis:names:tc:xacml:1.0:subject:locality"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Apply>
          <EnvironmentAttributeDesignator
              AttributeId=
                "urn:va:xacml:2.0:interop:rsa8:environment:locality"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </Apply>
        </Apply>
        <!-- True if hl7:pea-001 NOT EQUAL TO ANYOF subject:hl7:permission  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string"
              >urn:va:xacml:2.0:interop:rsa8:hl7:pea-001</AttributeValue>
            <SubjectAttributeDesignator
                AttributeId=
                  "urn:va:xacml:2.0:interop:rsa8:subject:hl7:permission"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </Apply>
        </Apply>
       </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:emergency:permit"
        Effect="Permit">
      <Description>
          If a Deny was not obtained above AND subject not part of a supported
           facility then subject must have emergency permission.
      </Description>
      <Target/>
      <Condition>
        <!-- True if subject:locality NOT EQUAL TO ANYOF environment:locality  -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
           <SubjectAttributeDesignator
               AttributeId=
                 "urn:oasis:names:tc:xacml:1.0:subject:locality"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Apply>
          <EnvironmentAttributeDesignator
              AttributeId=
                "urn:va:xacml:2.0:interop:rsa8:environment:locality"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </Apply>
        </Apply>
      </Condition>
    </Rule>
    <Obligations>
      <!-- These obligations provide specific instructions to PEP in the response -->
      <!-- This obligation informs the PEP user granted emergency access      -->
      <Obligation
        ObligationId="urn:va:xacml:2.0:interop:rsa8:obligation:emergency:permit"
        FulfillOn="Permit"/>
    </Obligations>
  </Policy>
</PolicySet>
```

EndOfXacmlPolicySet-02f-emergency

2.2.5 Data filtering Use Case

2.2.5.1 Scenarios for Data Filtering Use Case

2.2.5.1.1 Scenario: Data Filtering (all permissions granted)

Initial State / Pre-condition:

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

Scenario:

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

RESULT:

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

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

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

2.2.5.2 Detailed Data: Data Filtering

2.2.5.2.1 Detailed Data Elements

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

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

2.2.5.2.2 Detailed Request, Policy Sets, Response

XacmlRequest-05-01: [xacml-msg-policy-index]

```
<?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 5-01: Should be Perm + Obl: Dr A is on dissented list  -->
  <!-- ************************************************************** -->

  <!-- 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. Alice</AttributeValue>
     </Attribute>
    <Attribute
        AttributeId="urn:oasis:names:tc:xacml:1.0:subject:locality"
        DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>Facility A</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</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:prd-003</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-005</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-006</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-009</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-010</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-012</AttributeValue>
       <AttributeValue>urn:va:xacml:2.0:interop:rsa8:hl7:prd-017</AttributeValue>
     </Attribute>
    <Attribute
        AttributeId="urn:va:xacml:2.0:interop:rsa8:resource:hl7:confidentiality-code"
         DataType="http://www.w3.org/2001/XMLSchema#string">
       <AttributeValue>MA</AttributeValue>
    </Attribute>
    <Attribute
        AttributeId=
          "urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology: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
        >urn:va:xacml:2.0:interop:rsa8:resource:hl7:medical-record</AttributeValue>
    </Attribute>
  </Resource>
  <Action/>
  <Environment>
    <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>
```

EndOfXacmlRequest-05-01

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

XacmlPolicySet-02e-MA: [xacml-msg-policy-index]

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
    PolicySetId="urn:va:xacml:2.0:interop:rsa8:policysetid:MA"
    PolicyCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides">
  <Description>
      Policy set for the MA confidentiality code.
  </Description>
  <Target/>
  <Policy
      PolicyId="urn:va:xacml:2.0:interop:rsa8:policyid:MA"
      RuleCombiningAlgId=
        "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Target/>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA:1"
        Effect="Deny">
      <Description>
        If the access subject is NOT one of those users which consent has
        been removed, then deny.
        Note: there is reverse logic here because the Obligation that denies
        access to the user for this object must be issued when the user has
        obtained a Permit. So, the caller of this policy must know to reverse
        sense as well.
      </Description>
      <Target/>
      <Condition>
       <!-- True if hl7:radiology:dissented-subject-id NOTEQUALTO subject:subject-id -->
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:not">
        <!-- True if hl7:radiology:dissented-subject-id EQUALTO subject:subject-id   -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
          <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <SubjectAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </Apply>
          <ResourceAttributeDesignator
            AttributeId=
             "urn:va:xacml:2.0:interop:rsa8:resource:hl7:radiology:dissented-subject-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
       </Apply>
      </Condition>
    </Rule>
    <Rule
        RuleId="urn:va:xacml:2.0:interop:rsa8:rule:MA:2"
        Effect="Permit">
      <Description>
          If a Deny was not obtained above then set Permit by default.
      </Description>
    </Rule>
    <Obligations>
      <!-- These obligations provide specific instructions to PEP in the response -->
      <!-- This obligation instructs the PEP to apply privacy constraints to      -->
      <!--  user's responsibility for the data.                                   -->
      <Obligation
          ObligationId=
           "urn:va:xacml:2.0:interop:rsa8:obligation:ma:privacy:constraint:radiology"
          FulfillOn="Permit"/>
    </Obligations>
  </Policy>
</PolicySet>
```

EndOfXacmlPolicySet-02e-MA

A. Acknowledgements

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

Sponsors:

Jane Harnad, OASIS
Dee Schur, OASIS

Participants:

Erik Rissanen, Axiomatics
Babak Sadighi, Axiomatics
Wayne Delisser, BEA Systems, Inc.
Cynthia Ding, BEA Systems, Inc.
Hal Lockhart, BEA Systems, Inc. (Lead/Moderator)
Denis Pilipchuk, BEA Systems, Inc.
Anil Tappetla, Cisco Systems, Inc.
Ed Coyne, Department of Veterans Affairs
Mike Davis, Department of Veterans Affairs
Duane DeCouteau, Department of Veterans Affairs (Edmond Scientific Company)
Suzanne Gonzales-Webb, Department of Veterans Affairs (SAIC)
David Staggs, Department of Veterans Affairs (SAIC)
Kathleen Connor, Fox Systems
John Moehrke, GE Healthcare
Craig Forster, IBM
Maryann Hondo, IBM
Vernon Murdoch, IBM
Anthony Nadalin, IBM
Saket Kaushik, Oracle Corporation
Rich Levinson, Oracle Corporation
Prateek Mishra, Oracle Corporation
Hari Sastry, Oracle Corporation
Anil Saldhana, Red Hat
Dilli Dorai, Sun Microsystems
Luis Bernardo, Symlabs
Sampo Kellomaki, Symlabs

[Participant Name, Affiliation | Individual Member]

B. Non-Normative Text

C. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
[Rev number]	[Rev Date]	[Modified By]	[Summary of Changes]