**Appendix A: A Very Simple XACML Policy for our Proof-of-Concept Implementation**

<?xml version="1.0" encoding="UTF-8"?>

<Policy

 xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"

 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance

 xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17

 http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd"

 PolicyId="urn:oasis:names:tc:xacml:3.0:example:SimplePolicy1"

 Version="1.0"

 RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides">

 <Description>

 Overall XACML Description for demo company "Demo Corp".

 </Description>

 <Rule

 RuleId= "urn:oasis:names:tc:xacml:3.0:example:SimpleRule1"

 Effect="Permit">

 <Description>

 Check if the user's purpose is 'PUBLICHEALTH'; if so, permit access to the ID of patients.

 </Description>

 <Target>

 <Subjects>

 <Subject>

 <SubjectMatch

 MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">

 <AttributeValue

 DataType=http://www.w3.org/2001/XMLSchema#string>

 PUBLICHEALTH</AttributeValue>

 <SubjectAttributeDesignator

 MustBePresent="false"

 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

 AttributeId="urn:oasis:names:tc:xspa:1.0:subject:purposeofuse"

 DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>

 </SubjectMatch>

 </Subject>

 </Subjects>

 <Resources>

 <Resource>

 <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">

 <AttributeValue

 DataType="http://www.w3.org/2001/XMLSchema#anyURI">

 http://www.InternalSite.com/Database/Patients.html

 </AttributeValue>

 <ResourceAttributeDesignator

 DataType=http://www.w3.org/2001/XMLSchema#anyURI

 AttributeId="urn:oasis:names:tc:xspa:1.0:subject:npi"/>

 </ResourceMatch>

 </Resource>

 </Resources>

 </Target>

 </Rule>

</Policy>

**Appendix B: The Simple P3P Policy that was derived from the XACML Policy in Appendix A**

<?xml version="1.0"?>

<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1" xmlns:p3p11="http://www.w3.org/2006/01/P3Pv11"

 xmlns:pe="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">

 <EXPIRY max-age="86400"/>

 <POLICY name="pe:temporary" discuri="http://www.example.com/hi.html">

 <ENTITY>

 <DATA-GROUP>

 <DATA ref="#business.name">Demo Corp</DATA>

 <DATA ref="#business.contact-info.online.email">admin@democorp.org</DATA>

 <DATA ref="#business.contact-info.online.uri">http://www.democorp.org</DATA>

 <DATA ref="#business.contact-info.postal.organization">Organization Name</DATA>

 <DATA ref="#business.contact-info.postal.city">Ottawa</DATA>

 <DATA ref="#business.contact-info.postal.country">Canada</DATA>

 </DATA-GROUP>

 </ENTITY>

 <ACCESS>

 <none/>

 </ACCESS>

 <DISPUTES-GROUP>

 <DISPUTES resolution-type="independent" service="http://www.PrivacySeal.example.org"

 short-description="PrivacySeal.example.org">

 <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>

 <REMEDIES>

 <correct/>

 </REMEDIES>

 </DISPUTES>

 <DISPUTES resolution-type="service" service="http://www.sample.example.com/privacy.html"

 short-description="Sample Example Customer Service">

 <LONG-DESCRIPTION>

 If you have any questions or concerns about our privacy policy please email privacy@sample.example.com or call 1-877-EXAMPLE.

 </LONG-DESCRIPTION>

 </DISPUTES>

 </DISPUTES-GROUP>

 <STATEMENT>

 <PURPOSE>

 <extension>

 <ppurpose> <health/> </ppurpose>

 </extension>

 <current/>

 <develop/>

 <admin/>

 </PURPOSE>

 <RECIPIENT>

 <ours/>

 <public/>

 </RECIPIENT>

 <RETENTION>

 <government/>

 </RETENTION>

 <CONSEQUENCE>

 Check if the users' purpose is 'PUBLICHEALTH'; if so, permit access to the ID of patients.

 </CONSEQUENCE>

 <DATA-GROUP>

 <DATA>

 <CATEGORIES> <government/> </CATEGORIES>

 </DATA>

 </DATA-GROUP>

 </STATEMENT>

 </POLICY>

</POLICIES>

**Appendix C: An Example XSLT document that derives a P3P Policy from an XACML Policy**

Note: this example is relatively extensive, but is still a proof-of-concept XSLT document; it does not handle all possible XACML policies, rules, or allowable formats. It recognizes and processes the following elements:

* PolicySet and its attributes
* Policy and its attributes
* Expiry
* Access
* Statement
* Recipient and attributes (not including “description”)
* Purpose and attributes
* Retention
* Data-group and data
* Non-Identifiable

The three elements below are read from other sources:

* Entity data
* The URL of descuri
* The URL of opturi

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p3p="http://www.w3.org/2002/01/P3Pv1" xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy" xmlns:n1="records" exclude-result-prefixes="xacml n1" <xsl:output method="xml" encoding="UTF-8" indent="yes"/>

 <!--global parameters,for discuri, opturi, and entity files-->

 <xsl:param name="discuri-file">http://www.xmlspy.com/privacy.html</xsl:param>

 <xsl:param name="opturi-file">http:www.xmlspy.com/privacy.html</xsl:param>

 <xsl:param name="entity-file">entityfile.xml</xsl:param>

 <xsl:param name="recipient-descrip-file">recipient.txt</xsl:param>

 <!--other keys and global variables-->

 <xsl:variable name="entity-id">entity</xsl:variable>

 <xsl:variable name="owner">owner</xsl:variable>

 <xsl:variable name="nonowner">nonowner</xsl:variable>

 <!-- for the records.xml file-->

 <xsl:key name="record" match="n1:Record" use="@id"/>

 <!--keys: rule grouping and global variables-->

 <xsl:key name="rule-group" match="xacml:Rule" use="@Effect"/>

 <!--keys: -->

 <xsl:variable name="string-equal-key">urn:oasis:names:tc:xacml:1.0:function:string-equal</xsl:variable>

 <xsl:variable name="string-one-and-only-key">urn:oasis:names:tc:xacml:1.0:function:string-one-and-only</xsl:variable>

 <xsl:variable name="or-key">urn:oasis:names:tc:xacml:1.0:function:or</xsl:variable>

 <xsl:variable name="and-key">urn:oasis:names:tc:xacml:1.0:function:and</xsl:variable>

 <xsl:variable name="recipient-subject-key">urn:oasis:names:tc:xacml:1.0:subject-category:recipient-subject</xsl:variable>

 <xsl:variable name="current-date-key">urn:oasis:names:tc:xacml:1.0:environment:current-dateTime</xsl:variable>

 <xsl:variable name="date-comparision-key">urn:oasis:names:tc:xacml:1.0:function:date-less-or-equal</xsl:variable>

 <xsl:variable name="action-purpose-key">urn:oasis:names:tc:xacml:1.0:action:action-purpose</xsl:variable>

 <xsl:variable name="action-purpose-specific-key">urn:oasis:names:tc:xacml:1.0:action:action-purpose-specific</xsl:variable>

 <xsl:variable name="subject-category-key">urn:oasis:names:tc:xacml:1.0:subject-category</xsl:variable>

 <xsl:variable name="xpath-key">urn:oasis:names:tc:xacml:1.0:resource:xpath</xsl:variable>

 <xsl:variable name="rescource-id-key">urn:oasis:names:tc:xacml:1.0:resource:resource-id</xsl:variable>

 <!--xacml:PolicySet=>p3p: POLICIES-->

 <xsl:template match="/xacml:PolicySet">

 <p3p:POLICIES>

 <xsl:attribute name="xsi:schemaLocation">http://www.w3.org/2002/01/P3Pv1 http://www.w3.org/2002/01/P3Pv1.xsd</xsl:attribute>

 <xsl:apply-templates select="xacml:Policy"/>

 </p3p:POLICIES>

 </xsl:template>

 <!--xacml:policy=>p3p:POLICY-->

 <xsl:template match="xacml:Policy">

 <p3p:POLICY>

 <xsl:attribute name="name"><xsl:value-of select="@PolicyId"/></xsl:attribute>

 <xsl:attribute name="discuri"><xsl:value-of select="$discuri-file"/></xsl:attribute>

 <xsl:attribute name="opturi"><xsl:value-of select="$opturi-file"/></xsl:attribute>

 <!--data file name-->

 <xsl:variable name="rescource-node" select="xacml:Target//xacml:ResourceMatch[xacml:ResourceAttributeDesignator/@AttributeId=$rescource-id-key]"/>

 <xsl:variable name="data-file" select="normalize-space($rescource-node/xacml:AttributeValue/text())"/>

 <xsl:call-template name="entity"/>

 <xsl:call-template name="access">

 <xsl:with-param name="file-uri" select="$data-file"/>

 </xsl:call-template>

 <xsl:call-template name="statements">

 <xsl:with-param name="file-uri" select="$data-file"/>

 </xsl:call-template>

 </p3p:POLICY>

 </xsl:template>

 <!--entity information from the "entityfile.xml"-->

 <xsl:template name="entity">

 <p3p:ENTITY>

 <xsl:variable name="entity-info" select="document($entity-file)"/>

 <xsl:for-each select="$entity-info">

 <!--change to document records.xml nodes-->

 <p3p:DATA-GROUP>

 <xsl:for-each select="key('record',$entity-id)/n1:Data-group/\*">

 <p3p:DATA ref="{@ref}">

 <xsl:value-of select="."/>

 </p3p:DATA>

 </xsl:for-each>

 </p3p:DATA-GROUP>

 </xsl:for-each>

 </p3p:ENTITY>

 </xsl:template>

 <!--access information-->

 <xsl:template name="access">

 <!--for owner rules, regardless of whether it is write or read -->

 <xsl:param name="file-uri"/>

 <xsl:variable name="group" select="//xacml:Rule"/>

 <xsl:for-each select="$group">

 <xsl:variable name="current-node" select="current()"/>

 <xsl:variable name="category" select="xacml:Condition//xacml:SubjectAttributeDesignator[@AttributeId=$subject-category-key]"/>

 <xsl:for-each select="$category">

 <xsl:if test=" ../@FunctionId=$string-one-and-only-key and ancestor::\*[2]/xacml:AttributeValue/text()=$owner">

 <xsl:apply-templates select="$current-node" mode="access">

 <xsl:with-param name="data-file-uri" select="$file-uri"/>

 </xsl:apply-templates>

 </xsl:if>

 </xsl:for-each>

 </xsl:for-each>

 </xsl:template>

 <xsl:template match="xacml:Condition" mode="access">

 <xsl:param name="data-file-uri"/>

 <!--construct element-->

 <xsl:variable name="permit-group" select="//xacml:Rule[@Effect='Permit']"/>

 <xsl:variable name="$deny-group" select="//xacml:Rule[@Effect='Deny']"/>

 <xsl:for-each select="$permit-group">

 <xsl:for-each select="../[@FunctionId=$string-equal-key]">

 <xsl:if test=" ../@FunctionId=$string-one-and-only-key">

 <xsl:variable name="attribute-value" select="xacml:AttributeValue"/>

 <xsl:variable name="selector" select=".//xacml:AttributeSelector"/>

 <xsl:if test="$selector and contains($selector/@RequestContextPath,'contact')"></xsl:if>

 </xsl:if>

 </xsl:for-each>

 </xsl:for-each>

 <xsl:for-each select="$deny-group">

 <xsl:for-each select="../[@FunctionId=$string-equal-key]">

 <xsl:if test=" ../@FunctionId=$string-one-and-only-key">

 <xsl:variable name="attribute-value" select="xacml:AttributeValue"/>

 <xsl:variable name="selector" select=".//xacml:AttributeSelector"/>

 <xsl:if test="$selector and contains($selector/@RequestContextPath,'nonident')"></xsl:if>

 </xsl:if>

 </xsl:for-each>

 </xsl:for-each>

 <xsl:if test="">

 <p3p:ACCESS>

 <xsl:value-of select="ident-contact"/>

 </p3p:ACCESS>

 </xsl:if>

 <xsl:if test="">

 <p3p:ACCESS>

 <xsl:value-of select="other-ident"/>

 </p3p:ACCESS>

 </xsl:if>

 <xsl:if test="">

 <p3p:ACCESS>

 <xsl:value-of select="contact-and-other"/>

 </p3p:ACCESS>

 </xsl:if>

 …

*Please note that due to space constraints, only a portion of the XSLT document is shown in this paper. The full XSLT document is available at* [*http://www.site.uottawa.ca/~cadams/papers/XSLTDoc(ExtendedAPEX-AppendixC).pdf*](http://www.site.uottawa.ca/~cadams/papers/XSLTDoc%28ExtendedAPEX-AppendixC%29.pdf)*.*