STANDARDS RELATED DOCUMENT

ADatP-4774.1

CONFIDENTIALITY METADATA LABEL SYNTAX (CMLS) – IMPLEMENTATION GUIDANCE

Edition A Version 1

NOVEMBER 2021

NORTH ATLANTIC TREATY ORGANIZATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

10 November 2021

1. The enclosed Standard-related Document ADatP-4774.1, Edition A, Version 1 CONFIDENTIALITY METADATA LABEL SYNTAX (CMLS) – IMPLEMENTATION GUIDANCE, which has been approved in conjunction with ADatP-4774 by the nations in the Consultation, Command and Control Board (C3B), is promulgated herewith.

2. ADatP-4774.1 Edition A, Version 1 is effective upon receipt.

3. This NATO standardization document is issued by NATO. In case of reproduction, NATO is to be acknowledged. NATO does not charge any fee for its standardization documents at any stage, which are not intended to be sold. They can be retrieved from the NATO Standardization Document Database ((https://nso.nato.int/nso/) or through your national standardization authorities.

4. This publication shall be handled in accordance with C-M(2002)60.

Dimitrios SIGOULAKIS
Major General, GRC (A)
Director, NATO Standardization Office

INTENTIONALLY BLANK

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1. BACKGROUND

The Primary Directive on Information Management (PDIM) (Reference [11]) prescribes the application of metadata and markings in accordance with NATO policies and directives to facilitate sharing and control of NATO information.

The PDIM defines metadata as structured information that describes, explains, locates, and otherwise makes it easier to retrieve and use an information resource. The structure consists of ‘elements’, each of which will contain ‘values’. The values relate to the resource itself, there may be controls over what the actual values can be.

Metadata is a key enabler for the effective and efficient management of information. Modern automated information systems require information resources to be labelled with metadata.

1.2. OBJECTIVE

The NATO Core Metadata Specification (Reference [7]) defines a set of core metadata elements to support information management in the Alliance.

This document recognizes the existence of communities of interest’s specific metadata standards and aims at steering their evolution in the mid to long term and at providing a single mediation standard in the short term to achieve sharing of information among different communities of interest.

1.3. SCOPE

NATO Core Metadata Specification (NCMS) (Reference [7]) applies to all NATO information and to any information resource handled or processed by NATO’s communications and information systems. NCMS describes information resource and supports its consistent and appropriate handling.

All NATO civil and military bodies are mandated to use NCMS.

Allies and Partners must also use NCMS when handling NATO information.

1.4. NATO METADATA REGULATORY STANDARDS

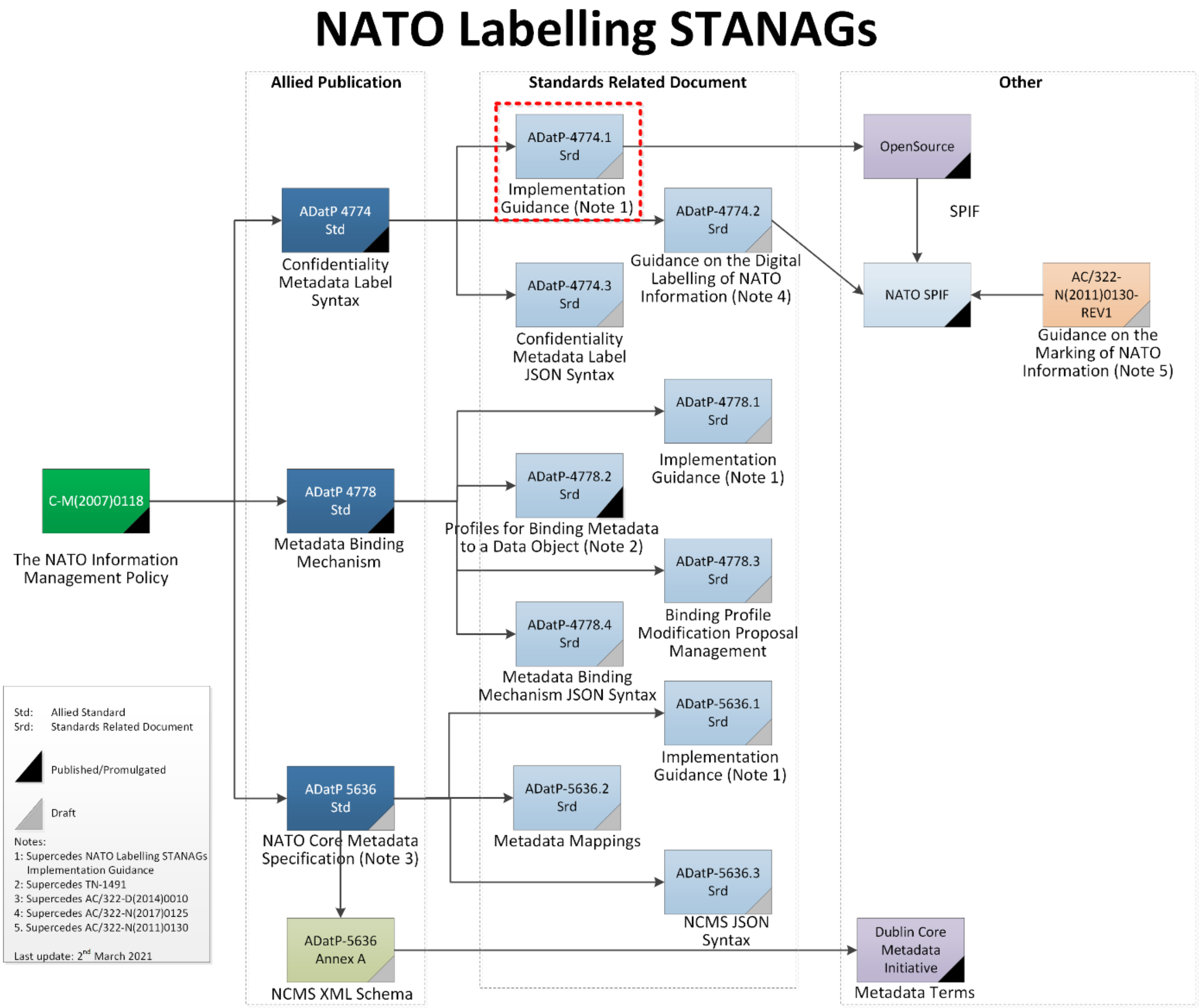
NATO has the following metadata standards:

- **ADatP-5636** NATO Core Metadata Specification defines the core set of metadata elements that must be used to support interoperable information exchange

- **ADatP-4774** Confidentiality Metadata Label Syntax provides support for the Security Layer metadata elements
- **ADatP-4778** Metadata Binding Mechanism describes how to consistently bind metadata (of any sort) to a finite data object

A number of separate, informative, Standard-related Documents (SRDs) are complementing these three metadata standards by providing implementation and other guidance, see Figure 1.

Figure 1: NATO Labelling STANAGs


NATO Labelling STANAGs
Allied Publication
Standards Related Document
Other
ADatP-4774.1 Srd
Implementation Guidance (Note 1)
OpenSource
SPIF
ADatP 4774 Std
Confidentiality Metadata Label Syntax
ADatP-4774.2 Srd
Guidance on the Digital Labelling of NATO Information (Note 4)
ADatP-4774.3 Srd
Confidentiality Metadata Label JSON Syntax
NATO SPIF
AC/322-N(2011)0130-REV1
Guidance on the Marking of NATO Information (Note 5)
ADatP-4778.1 Srd
Implementation Guidance (Note 1)
C-M(2007)0118
The NATO Information Management Policy
ADatP 4778 Std
Metadata Binding Mechanism
ADatP-4778.2 Srd
Profiles for Binding Metadata to a Data Object (Note 2)
ADatP-4778.3 Srd
Binding Profile Modification Proposal Management
ADatP-4778.4 Srd
Metadata Binding Mechanism JSON Syntax
ADatP-5636.1 Srd
Implementation Guidance (Note 1)
ADatP 5636 Std
NATO Core Metadata Specification (Note 3)
ADatP-5636.2 Srd
Metadata Mappings
ADatP-5636.3 Srd
NCMS JSON Syntax
ADatP-5636 Annex A
NCMS XML Schema
Dublin Core Metadata Initiative
Metadata Terms
Std: Allied Standard
Srd: Standards Related Document
Published/Promulgated
Draft
Notes:
1: Supercedes NATO Labelling STANAGs Implementation Guidance
2: Supercedes TN-1491
3: Supercedes AC/322-D(2014)0010
4: Supercedes AC/322-N(2017)0125
5. Supercedes AC/322-N(2011)0130
Last update: 2nd March 2021

This SRD is the Implementation Guidance for the Confidentiality Metadata Label Syntax (highlighted in a red, dashed box in Figure 1).

CHAPTER 2 OVERVIEW

2.1. INTRODUCTION

This Implementation Guidance describes a number of components that may be used to support the implementation of metadata confidentiality label within a given service/solution.

This Implementation Guidance is optional and implementers of ADatP-4774 are free to follow the guidance that they feel is most appropriate to their particular requirements.

The Implementation Guidance will be periodically updated with new guidance on how to support the implementation of the Confidentiality Metadata Label syntax based upon implementation experience

2.2. DATA CENTRIC SECURITY

The vision of Data Centric Security (DCS) is to deliver shareable Alliance information, protected at source, controlled for life.

DCS allows for variation in how protection requirements are determined, in what way the enforcement of the protection policy is executed, and in the choice of the underlying access control model. The variation is driven by evolution in the following directions:

1. The level of detail for describing information with metadata;
    a. *Sensitivity metadata.*
    b. *Common core metadata.*
2. The granularity of access control;
    a. *Clearance based.*
    b. *Attribute based.*
3. The level of object protection;
    a. *Deny or Grant Access Control.*
    b. *Cryptographic Access Control.*

The DCS Vision and Strategy (Reference [12]) defines three Maturity Levels (listed below) that have been determined by the variation in the different directions of evolution:

i. Basic Labelling – the majority of new data objects are labelled
    a. *Labelling and binding compliant with STANAG 4774 and STANAG 4778*
    b. Guard capability to mediate release based on confidentiality labels
    c. Use and management of metadata with the NATO Enterprise
ii. Enhanced Labelling – the majority of shared data objects are labelled and domain boundary release controlled

a. Integration with the NATO Enterprise Identify and Access Management
b. Granular labelling of all shareable data objects, including legacy data
c. Rich metadata compliant with STANAG 5636
d. Alliance-wide attribute-based access control
e. Agile response to changing security environment
f. Metadata labels applied to non-finite data streams e.g. voice and video, with appropriate guard technology

iii. Cryptographic protection – data objects controlled post-release

a. Cryptographic protection for data objects in transit and at rest
b. Controlled sharing of released data objects (federated digital rights management)
c. Converged cloud platforms for multi-level data separation
d. Increasing automation of information sharing and redaction.

Each of the Maturity Levels builds upon the foundations of the previous Maturity Level, and so all three Maturity Levels are dependent upon STANAG 4774 and 4778, and Maturity Levels 2, and 3, are dependent on STANAG 5636[1].

Implementation of STANAGs 4774, 4778 and 5636 across the NATO Alliance facilitates evolution of DCS in the direction of increasing the level of detail for describing information with metadata. Evolution of DCS in the directions of granularity of access control and level of object protection will require the implementation of additional standards and specifications.

This STANAG 4774.1 Implementation Guidance SRD thus provides guidance on the confidentiality labelling of data objects for all of the DCS Maturity Levels.

2.3. SECURITY POLICY INFORMATION FILE

A Security Policy Information File (SPIF) contains the values domains for the ConfidentialityInformation element of a Confidentiality Metadata Label. It contains the values for the PolicyIdentifier, Classification[2] and Category elements and the relationships between them (for example, a given Category value may not be valid when a given Classification is selected).

The SPIF can be used to support the:

- generation of confidentiality metadata labels by presenting the user with the valid options for each element;
- validation of confidentiality metadata labels against the identified policy;
- generation of human-readable markings for presentation to a use; and
- generation of equivalent confidentiality metadata labels in a different policy.

[1] At the time of writing this SRD, ADatP-5636 has been approved by C3B, and has been delivered to the NSO for ratification.

[2] 'Classified information' defined by C-M(2002)49-REV1 and 'non-classified information' by CM(2002)60.

A system/application can use the SPIF directly, or indirectly by using derived artefacts e.g. stylesheets.

Details of the SPIF specification and the artefacts that can be derived from the SPIF are contained in Chapter 3.

Note that SRD ADatP-4774.2, “Guidance on the Digital Labelling of NATO Information” (see Figure 1), provides details of the value domains for the NATO policy identifier SPIF.

2.4. CONFIDENTIALITY LABEL

ADatP-4774 defines a number of different confidentiality labels, including:

- originatorConfidentialityLabel
- alternativeConfidentialityLabel
- successorConfidentialityLabel

Chapter 4 provides implementation guidance for the use of these confidentiality labels and defines a set of rules that should be applied when determining how to handle information labelled based on these confidentiality labels. In addition, it also defines rules for determining the dominant label for a data object that is granularly labelled.

2.5. CONFIDENTIALITY CLEARANCE

ADatP-4774 defines a XML syntax to describe a clearance that is aligned the confidentiality labels. The clearance is used by the access control framework to make an access control decision based upon the confidentiality labels bound to a Data Object.

The clearance is not associated with a Data Object but rather with the initiator who is requesting access to the data object. The access control framework must therefore be able to locate the initiator’s clearance from a standard location.

Chapter 5 provides implementation guidance for the location of a confidentiality clearance within an LDAP directory entry and a SAML Assertion token.

2.6. ACCESS CONTROL FRAMEWORK

The confidentiality label can be used within an access control framework to make decisions on the release of a data object.

Chapter 6 describes the how use of the confidentiality label and confidentiality clearance within a XACML version 3.0, and the use of data-flow model based Label Based Access Control model

2.7. LABEL CATALOGS

A confidentiality label catalog contains a list of complete agreed confidentiality metadata labels. A user or application can select one of the confidentiality metadata labels and then include it directly in a metadata binding. In this way, the user/application does not need to know about the specific syntax of the metadata confidentiality label.

As the confidentiality label catalog contains a complete confidentiality metadata label, it contains all of the mandatory elements of a metadata confidentiality label and therefore the user/system does not need to provide any additional information.

In addition, the confidentiality label catalog can be validated (e.g. against the corresponding SPIF) and updated independently of the application.

This approach works well when a small, known, set of metadata confidentiality labels are used within a system.

A basic confidentiality label catalog can be–derived from a SPIF using a simple transformation.

Implementation Guidance on the definition and use of a confidentiality label catalog based on the standard Genericode Code List syntax is provided in Chapter 7.

CHAPTER 3 SECURITY POLICY INFORMATION FILE

3.1. INTRODUCTION

The SPIF is a governing policy to systems, supporting the application of access control using confidentiality labels with the following key features:

a) Confidentiality label and confidentiality clearance validity: The SPIF defines which classification[3] levels are valid and their associated hierarchy, and defines whether categories are permissive, restrictive or informative. The validation of a confidentiality label, or confidentiality clearance, must determine that:
   i. The governing policy is known and the confidentiality label, or confidentiality clearance, is syntactically correct; and,
   ii. The combination of classification and categories is valid as defined by the SPIF.

b) Render confidentiality label as human-readable marking. Confidentiality labels are used in the digitalized environment and are designed to be machine-readable. As such, it is imperative that users are informed of the sensitivity of the information based on the confidentiality label being used. The SPIF defines how confidentiality labels are displayed (in a variety of languages) in order to ensure a consistent representation as a confidentiality marking. A SPIF will also support where and how the confidentiality marking is required to be presented. For example, a governing policy may require the confidentiality marking to be positioned at the top and bottom of a document and be displayed with a different colour depending upon the classification.

c) Access Control Rules: To support access control based on confidentiality labels and confidentiality clearances. The SPIF defines a hierarchy for classifications and the rules for handling categories depending on whether they are restrictive, permissive or informative. In addition, with a governing policy, SPIF defines rules when an access control decision is required when information is not labelled or when an identity that is requesting access to labelled information does not have a clearance.

An XML schema for representing a SPIF is defined at www.xmlspif.org and reproduced for convenience in ANNEX A.

The attributes and elements of the XML SPIF are summarised in the following sections.

An example SPIF is provided in ANNEX B to highlight the use of the attributes and elements described.

[3] As presented at ADatP 4774.2, Chapter 2, 3.2 Classification

The SPIF can be used to capture the value domains defined in ADatP-4774.2 “Guidance on the Digital Labelling of NATO Information” (Reference [3]) in a machine processable form.

3.2. ATTRIBUTES

The top-level SPIF element contains the following attributes, as shown in Table 1.

Table 1:Top-level SPIF Attributes

Attribute	**Description**	**REQUIRED / OPTIONAL**
schemaVersion	Version of the XML SPIF.	REQUIRED
version	Differentiates between versions of the Policy represented by the SPIF.	OPTIONAL
creationDate	Date when the SPIF was generated or updated.	REQUIRED
originatorDN	Unique information (distinguished name encoded as defined in RFC 4514) identifying the creator of the SPIF.	REQUIRED
keyIdentifier	Identifies the public key that was used to sign the SPIF.	REQUIRED
privilegeId	An object identifier (OID) that identifies the syntax used by the clearance category (must match the value indicated by the rbacId). The value MUST be 1.3.26.0.4774.5.24.1.	REQUIRED
rbacId	An object identifier (OID) that identifies the syntax of the category that is used in conjunction with the SPIF. The value MUST be 1.3.26.0.4774.5.24.1.	REQUIRED
docRefURI	A Unique Resource Identifier (URI) that references the Governing Policy that the SPIF is representing.	OPTIONAL
userRefURI	A Unique Resource Identifier (URI) that references governing policy documentation that further describes the use of the values defined in the SPIF.	OPTIONAL
validity	The validity of the SPIF (i.e. the SPIF may be used under certain contexts, such as a specific exercise). The attributes in this attributeGroup are notBefore and notAfter and are both of type xs:dateTime.	OPTIONAL

3.3. ELEMENTS

The XML SPIF consists of a sequence of the following elements:

1. defaultSecurityPolicyId
2. securityPolicyId
3. securityClassification
4. securityCategoryTagSet
5. privacyMarks
6. equivalentPolicies
7. extensions

Note that the XML SPIF specification is a Non-NATO Standard (NNSTD), and hence, in places, uses terminology that is not consistent with NATO usage when used to capture the NATO policy. Specifically:

The XML SPIF, in line with other standards such as S/MIME ESS (Reference [14]) and X.400 MHS (Reference [15]) treats “UNCLASSIFIED” as a “security classification” and uses this term for all classifications, whilst the “Management of Non-classified NATO Information” (Reference [16]) explicitly states that NATO UNCLASSIFIED is not a “security classification”.

As a NNSTD, the terminology cannot be changed, thus in the NATO context, the following terms used in this sectionshould be interpreted as follow:

1. The terms “defaultSecurityPolicyId” and “securityPolicyId” should be interpreted as “defaultPolicyId” and “policyId”.
2. the term “securityClassification” should be interpreted as “classification”.

3.3.1. defaultSecurityPolicyId

The values represented with this element cater for interoperability with COIs, nations or organisations that do not support the NATO Labelling STANAGs. This element is represented by an object identifier and a name and is an OPTIONAL element.

The default policy identifier shall be used as a basis for access control decisions when there is no confidentiality label bound to the information, and this value may be used when one or more of the following is true:

- the originating identity has no confidentiality clearance;
- the receiving identity has no confidentiality clearance;
- the originating and receiving COIs or organisations have no agreed mapping between policies.

The organisation can determine if it supports this concept by setting the default policy. The SPIF can then be used to map the default policy to one classification within another policy. For example, if incoming unlabelled data, or a label that was not able to be mapped, entered the SPIFs domain, then this setting can assert that the data is handled as unclassified in NATO ICT systems.

3.3.2. securityPolicyId

A unique object identifier (OID) and a name representing the owner of the policy. This element is REQUIRED.

3.3.3. securityClassification

The securityClassification element defines the classifications within the policy, and also provides equivalency mappings between classifications in different policies.

The securityClassification element is REQUIRED and has the following set of attributes as specified in Table 2.

Table 2: securityClassification Attributes

Attribute	Description	REQUIRED / OPTIONAL
name	Classification name to be used in the display, e.g., RESTRICTED;	REQUIRED
color	Foreground and background colours associated with the classification;	OPTIONAL
lacv	LabelAndCertValue represents the value assigned (encoded) to this classification in the confidentiality label.	REQUIRED
hierarchy	A unique value representing the position of the classification within the scheme of ordering classifications. For example, this can be used for determining the dominant classification across multiple confidentiality labels.	REQUIRED

The securityClassification also includes a sequence of the following OPTIONAL elements:

- equivalentClassification: this is an unbounded sequence of equivalent classifications from other policies. This element includes the name of the other policy for referential validity checks, the lacv value, and any additional required categories associated with this equivalent mapping. It also includes an applied attribute that determines if the equivalent mapping occurs when asserting the label (encrypt), receiving the label (decrypt) or both (both);

- markingData: the confidentiality marking information attached with the information (see Section Markings);

- markingQualifier: Specifies a suffix or prefix for qualifying the markingData (see Section Markings);

- requiredCategory: selecting a classification to go into a confidentiality label may require the inclusion of certain categories. This element allows for stipulating which category group must be associated with the classification. This element additionally supports an operation where the amount of categories can be specified for inclusion. This operation supports:

- only one (onlyOne),
- one or more (oneOrMore), and
- all (all).

3.3.4. securityCategoryTagSet

An OPTIONAL element that consists of a sequence of equivalentSecurityCategoryTagSet, requiredCategory, a unique object identifier (OID) id, a name and securityCategoryTagSet.

The securityCategoryTagSet consist of a sequence of one or more category tags (securityCategoryTag) that indicates the type of category tag by its tagType, enumType and tag7Encoding attributes as follows:

- permissive:
    - tagType attribute value of enumerated and enumType attribute value of permissive.; or
    - tagType attribute value of permissive
- restrictive:
    - tagType attribute value of enumerated and enumType attribute value of restrictive; or,
    - tagType attribute value of restrictive
- informative:
    - tagType attribute value of tagType7 and tag7Encoding attribute value of bitSetAttributes

The securityCategoryTag consists of zero or more categories (tagCategory), and includes information on whether one or more categories can be selected (singleSelection: true|false), constraints on the number of categories that can be selected (maxSelection or minSelection) and marking information (markingQualifier - see section 2.4).

A category consists of equivalent categories, marking data, required classifications, excluded classifications, required categories, excluded categories, time validity and whether it is obsolete or not. Table 3 describes the elements and attributes for a category (tagCategory).

Table 3: tagCategory Attributes

Attribute / Element	**Description**	**REQUIRED / OPTIONAL**
equivalentSecCategoryTag	Refer to Section Equivalencies	OPTIONAL
markingData	Refer to Section Markings	OPTIONAL
markingQualifier	Refer to Section Markings	OPTIONAL

excludedClass	A classification (defined in a SPIF) that is prohibited from being used in conjunction with the category in a confidentiality label.	OPTIONAL
requiredCategory	A category (defined in a SPIF) that is required to be included in the confidentiality label when this category is used.	OPTIONAL
excludedCategory	A category (defined in a SPIF) that is prohibited from being used in conjunction with the category in a confidentiality label.	OPTIONAL
name	The category name.	REQUIRED
lacv	LabelAndCertValue represents the value assigned (encoded) to this category in a binary confidentiality label.	REQUIRED
userInput	The format of the category that can be entered by a user when creating a confidentiality label. The choice of this attribute value is: string; integer; or date (the format of which is defined by the dateFormat attribute).	OPTIONAL
requiredClass	A classification (defined in a SPIF) that is required to be used with the category in the same confidentiality label.	OPTIONAL
obsolete	Used to distinguish whether the category is not used within the context of the policy. Provides backwards compatibility when validating confidentiality labels that may contain categories that are no longer enforced by the policy. Default value is false.	OPTIONAL
dateFormat	The ISO 8601 date format to be used to support the userInput attribute, e.g. YYYY-MM-DD	OPTIONAL
validity	A group of attributes that determine the period in which the category is valid. The attributes are notBefore and notAfter and are both of type xs:dateTime.	OPTIONAL

3.3.5. privacyMarks

Privacy Marks are not part of any access control decision. However, they can be carried in the confidentiality label. The XML SPIF provides the ability to stipulate marking codes and marking phrases along with privacy marks.

3.3.6. equivalentPolicies

The Equivalent Policies element allows for the object identifier and the name to be asserted for policies that this policy has equivalent mappings with. If the SPIF has no equivalent policies mapped then the use of this element is NOT REQUIRED. If a policy mapping exists between this policy and one or more policies then the use of this element is REQUIRED. See section 2.4.

3.3.7. extensions

The extensions element provides elements in support of organisation-specific extensions. For instance, the STANAG 4778 Binding Profile for SPIFs make use of the extension element to store the BindingInformation which can be used to associated metadata with the SPIF.

3.4. Consistency

XML, XSD and XPATH provide mechanisms to verify transitive checks for the validity of cross references within the SPIF. The following transitive checks are undertaken when completing an XML SPIF:

i. all reference policies have a unique object identifier;
ii. all reference policies have a unique name;
iii. all equivalent classifications reference a known equivalent policy;
iv. all equivalent categories reference a known equivalent policy;
v. all security classifications have a unique lacv value;
vi. all security classifications have a unique name;
vii. all security classifications have a unique hierarchy value;
viii. all security classifications have a unique colour;
ix. all categories reference a known required classification;
x. all categories reference a known excluded classification;
xi. all tag sets have a unique name;
xii. all categories reference a known required category tag set;
xiii. all categories reference a known excluded category tag set;
xiv. all classifications reference a known required category tag set;
xv. simpleType checks to ensure validity of inputted data, e.g., OID, string, datetime, integer.

3.5. MARKINGS

One of the roles of the XML SPIF is to consistently display confidentiality labels across the organisation. The XML SPIF provides for this with the usage of Marking Codes, Marking Phrases and Colours.

The element for identifying the confidentiality marking information associated with information is the markingData element. The markingData element is an optional child element of the SPIF, tagCategory, securityClassification, securityClassifications, privacyMark and privacyMarks. Table 4 lists the supported attributes and elements of the markingData element.

Table 4: markingData Elements and Attributes

Element / Attribute	**Description**	**REQUIRED / OPTIONAL**
phrase	The confidentiality marking phrase associated with the element that the attribute is a member of.	REQUIRED
code	A sequence of one or more marking codes which identifies the location of the marking phrase. See Table 6 for the list of code attribute values.	REQUIRED

The element for qualifying the markingData associated with information is the markingQualifier element. The markingQualifier element is an optional child element of the SPIF, tagCategory, securityClassification, securityClassifications, privacyMark and privacyMarks. Table 5 lists the supported attributes and elements of the markingQualifier element.

Table 5: markingQualifier Elements and Attributes

Element / Attribute	**Description**	**REQUIRED / OPTIONAL**
qualifier	A sequence of one or more qualifier elements. The qualifier element contains two REQUIRED attributes: 1) markingQualifier - the qualifier phrase i.e. “REL TO” or “,” and, 2) qualifierCode – where the markingQualifier phrase is located. The allowed attribute values are: i. prefix; ii. suffix; or iii. separator	REQUIRED
markingCode	An attribute that identifies where the markingData phrase attribute value is to be physically applied. See Table 6 for the list of markingCode attribute values.	REQUIRED

Marking Codes are used to identify the location of the applicable confidentiality marking as required by an associated classification or category. Marking phrases define the phrase to be displayed or printed at the required location directed by the associated classification or category. Table 6 lists the supported Marking Codes for displaying classifications and categories in a confidentiality label.

Table 6: Marking Codes

Marking Code	Description
pageTop	Display on top of the page or viewing area e.g. a page header.
pageBottom	Display on bottom of the page or viewing area e.g. a page footer.
pageTopBottom	Display on top and bottom of the page or viewing area e.g. a page header and a page footer.
documentStart	Display at start of the document e.g. cover page.
documentEnd	Display at end of the document e.g. end page.
noNameDisplay	Disable display of classification or category; only display the marking phase.
noMarkingDisplay	Do not display marking phrase on output; display marking phrase only during user input.
supressClassName	Suppress display of classification name and only display the category.
firstLineofText	Apply to the first line of a text bodypart e.g. the body text of an email message.
lastLineofText	Apply to the last line of a text bodypart e.g. the body text of an email message.
subject	Apply to the subject of an email message
xHeader	Apply to the header of an email message. The actual header name is held within the prefix qualifier.
portionMarking	Apply to a specific portion of a document.
inputTitle	Display in the title or label of a GUI element. The actual header name is held within the prefix qualifier.
waterMark	Display as the watermark behind the main text of a document.
replacePolicy	Replace the policy marking phrase.

The XML SPIF also allows for markings applied across an entire set of categories. An example of this would be a Releasable To set of categories where each country is represented as a single category. Marking Qualifiers allow for the display of marking phrases at the location specified by the Marking Qualifiers qualifierCode along with the list of categories. The qualifierCode supports this concept as prefix, suffix or separator.

In order to further facilitate the consistent display of confidentiality labels, the XML SPIF provides a colouring scheme that can be applied to the classifications. The colouring scheme supports W3C Color Naming scheme and W3C Color Hex Scheme. The colours can be set to be displayed as a mix of foreground and background colours.

The XML SPIF supports confidentiality markings being displayed in multiple languages. This is supported by adding the xml:lang attribute as an attribute of the qualifier or markingData elements.

3.6. EQUIVALENCIES

The XML SPIF (as mentioned previously) supports mapping classifications and categories from one policy to one or more policies. The XML SPIF provides for this with the use of equivalentPolicies, equivalentClassification and equivalentSecCategoryTag.

The equivalentPolicies element contains one or more equivalentPolicy element.

Table 7 specifies the cardinality of the attributes of the equivalentPolicy element.

Table 7: equivalentPolicy Attributes

Attribute	**Description**	**REQUIRED / OPTIONAL**
name	The name of the equivalent policy.	REQUIRED
id	The equivalent policy identifier.	REQUIRED
userRefURI	A URI to user documentation on how the equivalent policy should be used.	OPTIONAL
docRefURI	A URI that describes the equivalent policy and its associated classification and category values.	OPTIONAL
requiredCategory	One or more categories that must be included in the equivalent confidentiality label.	OPTIONAL

For any classifications that have an equivalency with the equivalent policy indicated by the existence of an equivalent policy (see equivalentPolicies) then that classification shall contain an equivalentClassification element.

Table 8 specifies the cardinality of the attributes of the equivalentClassification element.

Table 8: equivalentClassification Attributes

Attribute	**Description**	**REQUIRED / OPTIONAL**
policyRef	The target policy for the equivalent confidentiality label. The attribute value must exist as an equivalentPolicy name attribute value.	REQUIRED
lacv	The lacv of the equivalent classification of the equivalent policy.	REQUIRED
applied	When equivalency should be applied. The values are: • encrypt – At origination	REQUIRED

	• decrypt – At reception (or by an intermediary) • both – At both origination and reception	
requiredCategory	One or more categories that must be included in the equivalent confidentiality label values.	OPTIONAL

For any categories that have an equivalency with the equivalent policy indicated by the existence of an equivalent policy (see equivalentPolicies) then that category shall contain an equivalentSecCategoryTag element.

Table 9 specifies the cardinality of the attributes of the equivalentSecCategoryTag element.

Table 9: equivalentSecCategoryTag Attributes

Attribute	**Description**	**REQUIRED / OPTIONAL**
policyRef	The target policy for the equivalent confidentiality label. The attribute value must exist as an equivalentPolicy name attribute value.	REQUIRED
lacv	The lacv of the equivalent category (tagCategory) of the equivalent policy.	REQUIRED
applied	When equivalency should be applied. The values are: • encrypt – At origination • decrypt – At reception (or by an intermediary) • both – At origination and reception	REQUIRED
tagSetId	The id of the equivalent category tag set (securityCategoryTagSet).	REQUIRED
tagType	Identifies the type of the tag (permissive, restrictive or informative) – See tagType above for values required.	REQUIRED
enumType	Identifies whether the tagType attribute value enumerated is restrictive or permissive.	OPTIONAL (if tagType is enumerated then it is REQUIRED).

3.7. MANAGEMENT

The policy (represented as an XML SPIF) is the central component for enforcing access control based on confidentiality labels and confidentiality clearances. The XML SPIF should be accessible within the organization’s domain(s) for those applications and services that are required to make informed decisions on labelling information. In addition, modifications to the XML SPIF must be promulgated in real-time. The XML SPIF can be made available within an enterprise-wide directory or an enterprise-wide registry or repository.

It is recommended that the XML SPIF is digitally signed in order to protect the integrity of the policy whilst providing authenticity over the creator or modifier of the policy, and non-repudiation.

The XML SPIF reflects the entire set of requirements specified by the organization policy. For certain use cases it may be beneficial to provide a subset of the policy requirements in order to reflect those requirements that need to be enforced for those use cases. As such, a subset of the XML SPIF (or partial XML SPIF) can be derived for each of those use cases. An example of such a use case would be a bilaterally agreed set of confidentiality labels that can be exchanged between information partners within a federation. Only those classifications and categories that are bilaterally agreed to be shared need be present in the partial XML SPIF.

3.8. SCHEMA

An XML schema for representing a SPIF is published at:
http://www.xmlspif.org/schema/xmlspif.xsd.

The SPIF XML Schema is also held within the NATO Metadata Registry and Repository (NMRR) at:
https://nmrr.ncia.nato.int/rest/doc/NATO/Information%20Assurance/Security%20Policy/spif-21.xsd (account required).

The XML SPIF schema (Version 2.1) is also provided in ANNEX A.

An example policy encoded as a SPIF is provided in ANNEX B.

3.9. NATO EXTENSIONS

3.9.1. Introduction

The NATO SPIF (XML Schema) captures the majority of the information associated with the NATO Security Policy[4]. However, there is some information that NATO currently maintains that cannot be represented in the machine processable SPIF.

Rather that define separate XML artefacts to capture this information, a number of extensions have been defined that can be used within the SPIF XML Schema so that only a single XML artefact is required to be managed and kept consistent.

Two extensions are currently defined:

1. Memberships – containing details of the memberships of NATO approved activities (see section 3.9.2).

[4] The NATO Security Policy is the governing policy for the labelling application within the NATO. C-M(2002)49-REV1, Security Within The North Atlantic Treaty Organization, 20 November 2020

2. BindingInformation - containing metadata associated with the SPIF (see section 3.9.3).

3.9.2. Membership

NATO maintains the details of the memberships of NATO approved activities, which for example, may be referenced in the Releasability (Releasable To) marking. This membership information may be useful when making an access control decision and it is appropriate to keep this information with the other policy information.

NATO have defined a simple extension to the SPIF to capture this information in a machine processable form.

Figure 2 shows an abridged example of a SPIF with the details of the member nations of the Euro-Atlantic-Partnership Council (EAPC).

Figure 2: Example SPIF Extension showing the Members of the EAPC.

```
<spif:SPIF>
…
  <spif:extensions>
    <nspif:memberships xmlns:nspif="urn:nato:stanag:4774:spif:extensions:1:0">
      <nspif:membership spif:name="EAPC" spif:lacv="2001">
        <nspif:member spif:name="ARM" spif:lacv="051" />
        <nspif:member spif:name="AUT" spif:lacv="040" />
        <nspif:member spif:name="AZE" spif:lacv="031" />
        <nspif:member spif:name="BLR" spif:lacv="112" />
        <nspif:member spif:name="BIH" spif:lacv="070" />
        <nspif:member spif:name="FIN" spif:lacv="246" />
        <nspif:member spif:name="GEO" spif:lacv="268" />
        <nspif:member spif:name="IRL" spif:lacv="372" />
        <nspif:member spif:name="KAZ" spif:lacv="398" />
        <nspif:member spif:name="KGZ" spif:lacv="417" />
        <nspif:member spif:name="MLT" spif:lacv="470" />
        <nspif:member spif:name="MDA" spif:lacv="498" />
        <nspif:member spif:name="RUS" spif:lacv="643" />
        <nspif:member spif:name="SRB" spif:lacv="688" />
        <nspif:member spif:name="SWE" spif:lacv="752" />
        <nspif:member spif:name="CHE" spif:lacv="756" />
        <nspif:member spif:name="TJK" spif:lacv="762" />
        <nspif:member spif:name="NOM" spif:lacv="807" />
        <nspif:member spif:name="TKM" spif:lacv="795" />
        <nspif:member spif:name="UKR" spif:lacv="804" />
        <nspif:member spif:name="UZB" spif:lacv="860" />
      </nspif:membership>
    </nspif:memberships>
  </spif:extensions>
</spif:SPIF>
```

The XML schema for the memberships extension is published in the NMRR.

3.9.3. BindingInformation

It is necessary to associate a set NATO Core Metadata with all data objects, including SPIFs, within NATO. STANAG 4778 “Metadata Binding Mechanism” defines a mechanism, and associated XML schema, to generically associate metadata with a data object.

A STANAG 4778 BindingInformation XML element can be used to extend the SPIF to include metadata by including it within the spif:extensions element. This is defined in the ADatP-4778.2 “Profiles for Binding Metadata to Data Objects” Standard-related Document, section 12.11.

Figure 3 shows an abridged example of a SPIF containing the mandatory NCMS metadata.

Figure 3: Example SPIF Extension for NCMS metadata

```
<spif:SPIF>
…
  <spif:extensions>
    <s4778:BindingInformation
      xmlns:s4778="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:s5636="urn:nato:stanag:5636:A:1:elements"
      xmlns:dcterms=http://purl.org/dc/terms/
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <s4778:MetadataBindingContainer>
        <s4778:MetadataBinding>
          <s4778:Metadata>
            <s4774:originatorConfidentialityLabel>
              <s4774:ConfidentialityInformation>
                <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
                <s4774:Classification>UNCLASSIFIED</s4774:Classification>
              </s4774:ConfidentialityInformation>
              <s4774:OriginatorID IDType="uniformResourceIdentifier">
              </s4774:OriginatorID>
              <s4774:CreationDateTime>2021-02-08T16:11:43Z
              </s4774:CreationDateTime>
            </s4774:originatorConfidentialityLabel>
          </s4778:Metadata>
          <s4778:Metadata>
            <s5636:Title>ACME SECURITY POLICY INFORMATION FILE</s5636:Title>
          </s4778:Metadata>
          <s4778:Metadata>
            <s5636:Creator type="organization">NATO C&amp;I Agency
            </s5636:Creator>
          </s4778:Metadata>
          <s4778:Metadata>
            <s5636:Publisher type="organization">NATO C&amp;I Agency
            </s5636:Publisher>
          </s4778:Metadata>
          <s4778:Metadata>
            <s5636:Identifier>urn:oid:1.2.3.4</s5636:Identifier>
          </s4778:Metadata>
          <s4778:Metadata>
            <s5636:DateCreated>2021-03-16T12:36:54+01:00</s5636:DateCreated>
          </s4778:Metadata>
          <s4778:DataReference URI="" />
        </s4778:MetadataBinding>
      </s4778:MetadataBindingContainer>
    </s4778:BindingInformation>
  </spif:extensions>
</spif:SPIF>
```

3.10. CONFIDENTIALITY LABEL VALUE DOMAINS

3.10.1 Introduction

The XML SPIF contains the value domains that are to be used within a confidentiality label with the corresponding policy. The SPIF is general purpose and contains value domains for different confidentiality label formats.

For example, for the value domain of the classification element of a confidentiality label

- some confidentiality label formats may use the *name* attribute (a string) of a SPIF *securityClassification* element, whereas
- other confidentiality label (e.g. binary) formats may use *lacv* attribute (an integer) of a SPIF *securityClassification* element.

The following sections specify the value domains obtained from a SPIF for the specific elements/fields of:

1. an ADatP-4774 confidentiality metadata label.
2. a ESS label (Reference [14]).

3.10.2 ADatP-4774 Confidentiality Metadata Label Value Domains

Table 10 shows the value domains contained within a SPIF for each of the elements/attribute contained within the ConfidentialityInformation element of an ADatP-4774 Confidentiality Metadata Label.

The following columns are used in the table:

1) Element – the name of the confidentiality element
2) Attribute – the name of the attribute of confidentiality. If the attribute is '-', then the SPIF value domain applies to the content of the element.
3) XPath Expression – an XPath expression that identifies the values within the SPIF that constitute the value domain of the element/attribute. Note the XPath expression may use elements from the confidentiality label (e.g. PolicyIdentifier, TagName), shown in italic, to provide the correct value domain.

Table 10 - SPIF Derived Value Domains for XML Confidentiality Metadata Labels

Confidentiality Information		**SPIF Value Domain**
Element	**Attribute**	**XPath Expression**[5]
PolicyIdentifier	-	//securityPolicyId/@name
	URI	//securityPolicyId/@id
Classification	-	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //securityClassification/@name
	URI	N/A
PrivacyMark	-	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //privacyMark/@name
Category	TagName	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //securityCategoryTagSet/@name
	URI	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //securityCategoryTagSet[@name=*TagName*]/@id
	Type	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //securityCategoryTagSet[@name=*TagName*] /securityCategoryTag@tagType
GenericValue	-	/SPIF[securityPolicyId/@name=*PolicyIdentifier*] //securityCategoryTagSet[@name=*TagName*] /securityCategoryTag[@tagType=*Type*] /tagCategory/@name

The value domain of the ADatP-4774 PolicyIdentifier element is the name attribute of the SPIF securityPolicyId element. As a SPIF only contains a single securityPolicyId, the name effectively identifies SPIF from which all the other value domains are drawn.

The value domain of the ADatP-4774 PolicyIdentifier URI attribute is the id attribute of the SPIF securityPolicyId element. The SPIF id attribute must be prefixed with “urn:oid:” to generate an appropriate ADatP-4774 URI attribute.

The value domain of the ADatP-4774 Classification element is the name attribute of the SPIF securityClassification elements.

The ADatP-4774 Classification URI attribute is not used so no value domain is drawn from the SPIF.

The value domain of the ADatP-4774 PrivacyMark element is the name attribute of the SPIF privacyMark elements.

The value domain of the ADatP-4774 Category TagName attribute is the *name* attribute of the SPIF securityCategoryTagSet elements.

The value domain of the ADatP-4774 Category URI attribute is the *id* attribute of the SPIF securityCategoryTagSet elements with the selected TagName. This is a single value. The SPIF id attribute must be prefixed with “urn:oid:” to generate an appropriate ADatP-4774 URI attribute.

[5] Namespace prefixes are removed from the XPath expression for readability.

The value domain of the ADatP-4774 Category Type attribute is the tagType attribute of the SPIF securityCategoryTag elements that are in the securityCategoryTagSet element with the selected TagName. Note that SPIF tagType must be transformed to uppercase to provide a valid value for a confidentiality metadata label Category Type.

The value domain of the GenericValue element is the name attribute of the SPIF tagCategory elements that are in the SPIF securityCategoryTagSet with the selected TagName and the SPIF securityCategoryTag with the selected Type.

3.10.2 ESS Label Value Domains

Table 11 shows the value domains contained within a SPIF for each of the fields contained within the ESSSecurityLabel field (Reference [14]) and where the security-categories field is defined in Annex C of ACP145 (Reference [17]).

The following columns are used in the table:

1) Field – the name of the ESSSecurity Label field
2) Sub-field – the name of the sub-field of the field. If the field is '-', then the SPIF value domain applies to the content of the field.
3) XPath Expression – an XPath expression that identifies the values within the SPIF that constitute the value domain of the element/attribute. Note the XPath expression may use elements from the confidentiality label (e.g. security-policy-identifier, tagName), shown in italic, to provide the correct value domain.

Table 11 - SPIF Derived Value Domains for an ESS Label with ACP145 categories

ESSSecurityLabel		**SPIF Value Domain**
Field	**Sub-field**	**XPath Expression**[6]
security-policy-identifier[7]	-	//securityPolicyId /@id
security-classification[8]	-	/SPIF[securityPolicyId/"@id=*security-policy-identifier*] //securityClassification /@lacv
privacy-mark	-	/SPIF[securityPolicyId/@id=*security-policy-identifier*] //PrivacyMark /@name
restrictive Bitmap	tagName	/SPIF[securityPolicyId/"@id=*security-policy-identifier*] //securityCategoryTagSet [securityCategoryTag[@tagType="restrictive"]] /@id
	attributeFlags	/SPIF[securityPolicyId=*PolicyIdentifier*] //securityCategoryTagSet[@id=*TagName* /securityCategoryTag

[6] Namespace prefixes are removed from the XPath expression for readability.

[7] Note that "security-policy-identifier" is the standard term used by both Reference [14] and Reference [17]. In the NATO context, this should be interpreted as the "policy-identifier".

[8] Note that "security-classification" is the standard term used by both Reference [14] and Reference [17]. In the NATO context, this should be interpreted as the "classification".

ESSSecurityLabel		SPIF Value Domain
Field	**Sub-field**	**XPath Expression[6]**
		[@tagType=”restrictive”] /tagCategory /@lacv
enumerated Permissive Attributes	tagName	/SPIF[securityPolicyId/”@id=*security-policy-identifier*] //securityCategoryTagSet [securityCategoryTag [@tagType=”enumerated” and @enumType=”permissive”]] /@id
	attributeList	/SPIF[securityPolicyId=*PolicyIdentifier*] //securityCategoryTagSet[@id=*tagName*] /securityCategoryTag [@tagType=”enumerated” and @enumType=”permissive”] /tagCategory /@lacv
enumerated Restrictive Attributes	tagName	/SPIF[securityPolicyId/”@id=*security-policy-identifier*] //securityCategoryTagSet [securityCategoryTag [@tagType=”enumerated” and @enumType=”restrictive”]] /@id
	attributeList	/SPIF[securityPolicyId=*PolicyIdentifier*] //securityCategoryTagSet[@id=*TagName*] /securityCategoryTag [@tagType=”enumerated” and @enumType=”restrictive”] /tagCategory /@lacv
permissive Bitmap	tagName	/SPIF[securityPolicyId/”@id=*security-policy-identifier*] //securityCategoryTagSet [securityCategoryTag[@tagType=”permissive”]] /@id
	attributeFlags	/SPIF[securityPolicyId=*PolicyIdentifier*] //securityCategoryTagSet [@id=*TagName*] /securityCategoryTag [@tagType=”permissive”] /tagCategory /@lacv
informative Attributes	tagName	/SPIF[securityPolicyId/”@id=*security-policy-identifier*] //securityCategoryTagSet [securityCategoryTag[@tagType=”tagType7”]] /@id
	bitSetAttributes	/SPIF[securityPolicyId/”@id=*security-policy-identifier*] //securityCategoryTagSet[@id=*tagName*] /securityCategoryTag [@tagType=”tagType7” and @type7Encoding=”bitSetAttributes”] /tagCategory /@lacv

ESSSecurityLabel		SPIF Value Domain
Field	**Sub-field**	**XPath Expression[6]**
	securityAttributes	/SPIF[securityPolicyId/"@id=*security-policy-identifier*] //securityCategoryTagSet[@id=tagName] /securityCategoryTag [@tagType="tagType7" and @type7Encoding="securityAttributes"] /tagCategory /@lacv

The value domain of the ESSSecurityLabel security-policy-identifier field is the id attribute of the SPIF securityPolicyId element. As a SPIF only contains a single securityPolicyId, the id effectively identifies SPIF from which all the other value domains are drawn.

The value domain of the ESSSecurityLabel security-classificaiton field is the lacv attribute of the SPIF securityClassification elements.

The value domain of the ESSSecurityLabel PrivacyMark element is the name attribute of the SPIF privacyMark elements.

The value domain of the ADatP-4774 Category TagName attribute is the *name* attribute of the SPIF securityCategoryTagSet elements.

The value domain of the ACP145 restrictiveBitmap tagName sub-field is the *id* attribute of the SPIF securityCategoryTagSet elements that contain a securityCategoryTag with a tagType attribute with the value "restrictive".

The value domain of the ACP145 restrictiveBitmap attributeFlags sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute wit the value "restrictive", and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*. Note that the lacv values represent bit positions within the attributeFlags.

The value domain of the ACP145 enumeratedPermissiveAttributes tagName sub-field is the *id* attribute of the SPIF securityCategoryTagSet elements that contain a securityCategoryTag with a tagType attribute with the value "enumerated" and a enumType attribute with the value "permissive".

The value domain of the ACP145 enumeratedPermissiveAttributes attributeList sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute wit the value "enumerated" and and eneumType attribute of "permissive", and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*.

The value domain of the ACP145 enumeratedRestrictiveAttributes tagName sub-field is the *id* attribute of the SPIF securityCategoryTagSet elements that contain a securityCategoryTag with a tagType attribute with the value “enumerated” and an enumType attribute with the value “restrictive”.

The value domain of the ACP145 enumeratedRestrictiveAttributes attributeList sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute wit the value “enumerated” and and eneumType attribute of “restrictive”, and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*.

The value domain of the ACP145 permissiveBitmap tagName sub-field is the *id* attribute of the SPIF securityCategoryTagSet element that contain a securityCategoryTag with a tagType attribute with the value “permissive”.

The value domain of the ACP145 permissiveBitmap attributeFlags sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute wit the value “permissive”, and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*. Note that the lacv values represent bit positions within the attributeFlags.

The value domain of the ACP145 informativeAttributes tagName sub-field is the *id* attribute of the SPIF securityCategoryTagSet elements that contain a securityCategoryTag with a tagType attribute with the value “tagType7”.

The value domain of the ACP145 informativeAttributes bitSetAttributes sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute with the value “tagType7” and a type7Encoding with the value “bitSetAttributes”, and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*. Note that the lacv values represent bit positions within the bitSetAttributes.

The value domain of the ACP145 informativeAttributes securityAttributes sub-field is the *lacv* attribute of the SPIF tagCategory elements that are children of a securityCategory that has a tagType attribute with the value “tagType7” and a type7Encoding with the value “securityAttributes”, and which in turn is a child of a securityCategoryTagSet that has an id attribute with the selected *tagName*.

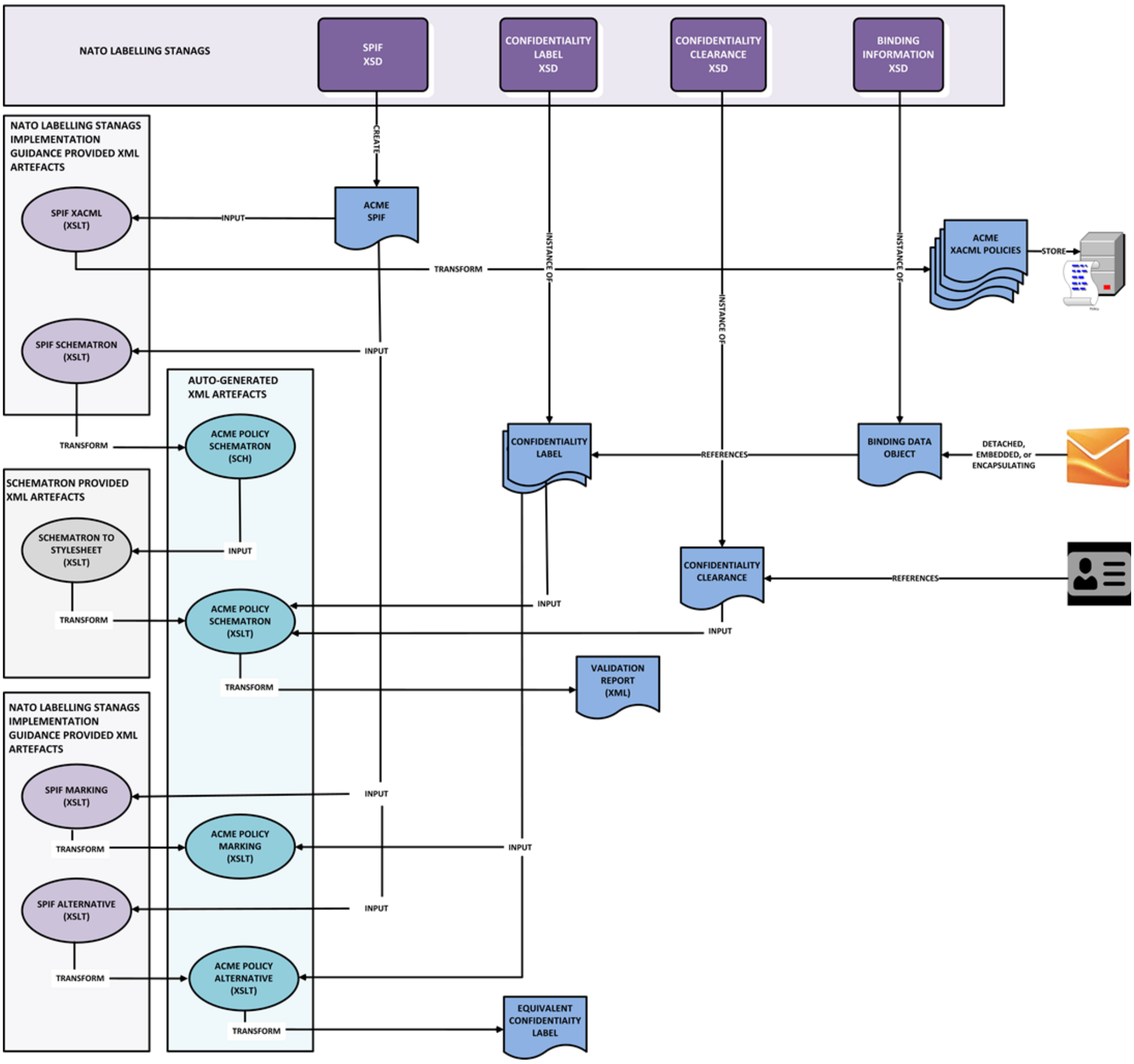
3.11. DERIVED XML ARTEFACTS

3.11.1. Introduction

The key component of the NATO Labelling STANAGs is the XML SPIF. Using widely available open standard technologies, such XML Stylesheet Language Transformations (XSLT), a number of supporting XML artefacts can be derived from a SPIF to support the consistent application of the policy.

Figure 4 below illustrates the components of STANAG 4774 and STANAG 4778 providing an overview of how the consistent application of the security policy can be supported based on a set of XML artefacts derived from the SPIF.

Figure 4: NATO Labelling STANAGs XML Artefacts


NATO LABELLING STANAGS
SPIF XSD
CONFIDENTIALITY LABEL XSD
CONFIDENTIALITY CLEARANCE XSD
BINDING INFORMATION XSD
NATO LABELLING STANAGS IMPLEMENTATION GUIDANCE PROVIDED XML ARTEFACTS
CREATE
ACME SPIF
SPIF XACML (XSLT)
INPUT
TRANSFORM
INSTANCE OF
ACME XACML POLICIES
STORE
SPIF SCHEMATRON (XSLT)
INPUT
INSTANCE OF
AUTO-GENERATED XML ARTEFACTS
TRANSFORM
ACME POLICY SCHEMATRON (SCH)
CONFIDENTIALITY LABEL
REFERENCES
BINDING DATA OBJECT
DETACHED, EMBEDDED, or ENCAPSULATING
SCHEMATRON PROVIDED XML ARTEFACTS
SCHEMATRON TO STYLESHEET (XSLT)
INPUT
CONFIDENTIALITY CLEARANCE
REFERENCES
TRANSFORM
ACME POLICY SCHEMATRON (XSLT)
INPUT
INPUT
TRANSFORM
VALIDATION REPORT (XML)
NATO LABELLING STANAGS IMPLEMENTATION GUIDANCE PROVIDED XML ARTEFACTS
SPIF MARKING (XSLT)
INPUT
TRANSFORM
ACME POLICY MARKING (XSLT)
INPUT
SPIF ALTERNATIVE (XSLT)
INPUT
TRANSFORM
ACME POLICY ALTERNATIVE (XSLT)
TRANSFORM
EQUIVALENT CONFIDENTIAITY LABEL

The NATO Labelling STANAGs implementation guidance provided XML artefacts illustrated in Figure 4, and they are described in the following sections.

3.11.2. Schematron

Schematron is a rules-based validation language that can be used to make assertions about the presence or absence of patterns within XML documents.

Schematron assertions can be defined for a specific policy to ensure that all of the values present within a confidentiality label or confidentiality clearance lie within the correct value domain.

For example, an assertion can be made that if the PolicyIdentifier element in a confidentiality label has the value “ACME” from the example policy (see ANNEX B), then the Classification element must have one of the values, “PUBLIC”, “CONFIDENTIAL” or “INTERNAL” (see Figure 5).

Figure 5: ACME Classification Assertion

```
<iso:rule context="s4774:Classification">
  <iso:let name="classification"
  value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
 <iso:assert test="$classification = 'public' or
   $classification = 'confidential' or
   $classification = 'internal'">
   Invalid security classification <iso:value-of select="." />,
   expected 'PUBLIC',  'CONFIDENTIAL' or 'INTERNAL'.
  </iso:assert>
</iso:rule>
```

A generic XML stylesheet can be used to transform a SPIF into set of corresponding Schematron rules, which can be used to ensure the semantic validity of an ADatP-4774 confidentiality label in the given policy. This stylesheet is available in the NMRR at: https://nmrr.ncia.nato.int/rest/doc/NATO/Information%20Assurance/Security%20Policy/spif2conflabel-schematron.xsl.

Running the stylesheet against the ACME example policy (See ANNEX B) results in a complete set of Schematron rules, as shown in Figure 6.

Figure 6: ACME Schematron Rules

```
<?xml version="1.0" encoding="utf-8"?>
<!--

      *** DO NOT EDIT ***
      Automatically generated by spif2conflabel-schematron.xsl.

    -->
<iso:schema                     xmlns:iso="http://purl.oclc.org/dsdl/schematron"
xmlns="http://purl.oclc.org/dsdl/schematron"
xmlns:spif="http://www.xmlspif.org/spif">
  <iso:title>
        Schematron for the ACME Policy
      </iso:title>
  <iso:ns           uri="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
prefix="s4774" />
  <iso:pattern name="PolicyIdentifier">
```

```
    <iso:rule context="s4774:PolicyIdentifier">
      <iso:let name="policy"
value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
      <iso:assert test="$policy = 'acme'">
        Invalid policy identifier <iso:value-of select="." />, expected "ACME".
      </iso:assert>
    </iso:rule>
  </iso:pattern>
  <iso:pattern name="Classification">
    <iso:rule context="s4774:Classification[text()='CONFIDENTIAL']">
      <iso:assert test="../s4774:Category[@TagName='Releasable
To']/s4774:GenericValue[text()='MOCK'] or ../s4774:Category[@TagName='Releasable
To']/s4774:GenericValue[text()='PHONY']">None of the required categories for
classification CONFIDENTIAL are not present.</iso:assert>
    </iso:rule>
    <iso:rule context="s4774:Classification">
      <iso:let name="classification"
value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
      <iso:assert test="$classification = 'public' or $classification =
'confidential' or $classification = 'internal'">
        Invalid classification <iso:value-of select="." />, expected 'PUBLIC',
'CONFIDENTIAL' or 'INTERNAL'.
      </iso:assert>
    </iso:rule>
  </iso:pattern>
  <iso:pattern name="Categories">
    <iso:rule context="s4774:Category">
      <iso:let name="tagName"
value="translate(@TagName,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwx
yz')" />
      <iso:assert test="$tagName = 'releasable to' or $tagName = 'administrative'
or $tagName = 'sensitive'">Invalid category name <iso:value-of select="@TagName"
/>, expected 'Releasable To',  'Administrative' or 'Sensitive'</iso:assert>
    </iso:rule>
  </iso:pattern>
  <iso:pattern name="CategoryTags">
    <iso:rule context="s4774:Category[@TagName='Releasable To']">
      <iso:let name="type"
value="translate(@Type,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'
)" />
      <iso:assert test="$type='permissive'">
        Invalid tag type  <iso:value-of select="@Type" /> for category Releasable
To, expected "permissive".
      </iso:assert>
    </iso:rule>
    <iso:rule context="s4774:Category[@TagName='Releasable
To']/s4774:GenericValue">
      <iso:let name="value"
value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
      <iso:assert test="$value = 'mock' or $value = 'phony'">
        Invalid value <iso:value-of select="." /> for category Releasable
To</iso:assert>
```

```
    </iso:rule>
    <iso:rule context="s4774:Category[@TagName='Administrative']">
      <iso:let                                                       name="type"
value="translate(@Type,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'
)" />
      <iso:assert test="$type='informative'">
        Invalid  tag  type   <iso:value-of  select="@Type"  />  for  category
Administrative, expected "informative".
      </iso:assert>
    </iso:rule>
    <iso:rule
context="s4774:Category[@TagName='Administrative']/s4774:GenericValue">
      <iso:let                                                      name="value"
value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
      <iso:assert test="$value = 'staff' or $value = 'finance' or $value = 'sales'
or $value = 'engineering'">
        Invalid    value    <iso:value-of    select="."    />    for    category
Administrative</iso:assert>
    </iso:rule>
    <iso:rule context="s4774:Category[@TagName='Sensitive']">
      <iso:let                                                       name="type"
value="translate(@Type,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz'
)" />
      <iso:assert test="$type='restrictive'">
        Invalid tag type  <iso:value-of select="@Type" /> for category Sensitive,
expected "restrictive".
      </iso:assert>
    </iso:rule>
    <iso:rule context="s4774:Category[@TagName='Sensitive']/s4774:GenericValue">
      <iso:let                                                      name="value"
value="translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"
/>
      <iso:assert test="$value = 'red' or $value = 'blue'">
        Invalid value <iso:value-of select="." /> for category Sensitive
      </iso:assert>
    </iso:rule>
    <iso:rule context="s4774:ConfidentialityInformation">
      <iso:assert test="&#xA;         not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName = 'Releasable To']/s4774:GenericValue = 'MOCK'))&#xA;
">Releasable To 'MOCK' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;        not((s4774:Classification = 'INTERNAL') and
(s4774:Category[@TagName = 'Releasable To']/s4774:GenericValue = 'MOCK'))&#xA;
">Releasable To 'MOCK' is excluded at classification 'INTERNAL'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName = 'Releasable To']/s4774:GenericValue = 'PHONY'))&#xA;
">Releasable To 'PHONY' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;        not((s4774:Classification = 'INTERNAL') and
(s4774:Category[@TagName = 'Releasable To']/s4774:GenericValue = 'PHONY'))&#xA;
">Releasable To 'PHONY' is excluded at classification 'INTERNAL'
      </iso:assert>
    </iso:rule>
    <iso:rule context="s4774:ConfidentialityInformation">
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName = 'Administrative']/s4774:GenericValue = 'STAFF'))&#xA;
">Administrative 'STAFF' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'CONFIDENTIAL')
and    (s4774:Category[@TagName    =    'Administrative']/s4774:GenericValue    =
'STAFF'))&#xA;          ">Administrative  'STAFF'  is  excluded  at  classification
'CONFIDENTIAL'
      </iso:assert>
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName      =      'Administrative']/s4774:GenericValue      =
'FINANCE'))&#xA;       ">Administrative 'FINANCE' is excluded at classification
'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'CONFIDENTIAL')
and    (s4774:Category[@TagName    =    'Administrative']/s4774:GenericValue    =
'FINANCE'))&#xA;       ">Administrative 'FINANCE' is excluded at classification
'CONFIDENTIAL'
      </iso:assert>
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName = 'Administrative']/s4774:GenericValue = 'SALES'))&#xA;
">Administrative 'SALES' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'CONFIDENTIAL')
and    (s4774:Category[@TagName    =    'Administrative']/s4774:GenericValue    =
'SALES'))&#xA;          ">Administrative  'SALES'  is  excluded  at  classification
'CONFIDENTIAL'
      </iso:assert>
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName      =      'Administrative']/s4774:GenericValue      =
'ENGINEERING'))&#xA;          ">Administrative  'ENGINEERING'  is  excluded  at
classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'CONFIDENTIAL')
and    (s4774:Category[@TagName    =    'Administrative']/s4774:GenericValue    =
'ENGINEERING'))&#xA;          ">Administrative  'ENGINEERING'  is  excluded  at
classification 'CONFIDENTIAL'
    </iso:assert>
    </iso:rule>
    <iso:rule context="s4774:ConfidentialityInformation">
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName   =   'Sensitive']/s4774:GenericValue   =   'RED'))&#xA;
">Sensitive 'RED' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;         not((s4774:Classification = 'CONFIDENTIAL')
and  (s4774:Category[@TagName  =  'Sensitive']/s4774:GenericValue  =  'RED'))&#xA;
">Sensitive 'RED' is excluded at classification 'CONFIDENTIAL'
      </iso:assert>
      <iso:assert test="&#xA;          not((s4774:Classification = 'PUBLIC') and
(s4774:Category[@TagName   =   'Sensitive']/s4774:GenericValue   =   'BLUE'))&#xA;
">Sensitive 'BLUE' is excluded at classification 'PUBLIC'
      </iso:assert>
      <iso:assert test="&#xA;        not((s4774:Classification = 'CONFIDENTIAL')
and  (s4774:Category[@TagName  =  'Sensitive']/s4774:GenericValue  =  'BLUE'))&#xA;
">Sensitive 'BLUE' is excluded at classification 'CONFIDENTIAL'
      </iso:assert>
    </iso:rule>
  </iso:pattern>
</iso:schema>
```

Other stylesheets can be developed can be developed in a similar manner to validate other confidentiality label syntaxes (e.g. a COI-specific syntax), or the confidentiality clearance.

Schematron rules can be interpreted directly by a Schematron engine, however, they can also be transformed into an XML stylesheet which allows the rules to be verified by a Stylesheet engine (which is more generally available than a Schematron engine).

A generic XML stylesheet to transform a set of Schematron rules into a XML stylesheet is provided at:

https://github.com/Schematron/stf/blob/master/iso-schematron-xslt1/iso_svrl_for_xslt1.xsl

The result of processing the ACME Schematron rules with this XML Stylesheet would result in another XML Stylesheet that can in turn be used to transform an XML document containing confidentiality labels in the ACME policy into a report in the Schematron Validation Report Language (SVRL). For example, processing the confidentiality shown in Figure 7.

Figure 7: Valid Confidentiality Label

```
<?xml version="1.0" encoding="utf-8"?>
<s4774:originatorConfidentialityLabel
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
 <s4774:ConfidentialityInformation>
  <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
  <s4774:Classification>INTERNAL</s4774:Classification>
  <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
   <s4774:GenericValue>RED</s4774:GenericValue>
  </s4774:Category>
 </s4774:ConfidentialityInformation>
 <s4774:OriginatorID IDType="rfc822Name">alan.ross@reach.nato.int
 </s4774:OriginatorID>
 <s4774:CreationDataTime>2017-03-14T09:00:00</s4774:CreationDataTime>
</s4774:originatorConfidentialityLabel>
```

Figure 7 would result in the SVRL document as shown in Figure 8.

Figure 8: SVRL - Valid Confidentiality Label

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
```

```
<svrl:schematron-output
  title="Schematron for the ACME Security Governing Policy"
  schemaVersion=""
  xmlns:svrl="http://purl.oclc.org/dsdl/svrl"
  xmlns:schold="http://www.ascc.net/xml/schematron"
  xmlns:iso="http://purl.oclc.org/dsdl/schematron"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
  <!--
              -->
  <svrl:ns-prefix-in-attribute-values prefix="s4774"
   uri="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="s4774:PolicyIdentifier" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="s4774:Classification" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="s4774:Category" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="s4774:ConfidentialityInformation" />
  <svrl:fired-rule context="s4774:Category[@TagName='Sensitive']" />
  <svrl:fired-rule
   context="s4774:Category[@TagName='Sensitive']/s4774:GenericValue" />
</svrl:schematron-output>
```

The report shows that rules has fired for the Confidentiality Information, PolicyIdentifier, Classification, Category, and the “Sensitive” Category and it values. As the report does not contained a *svrl:failed-assert* element, the confidentiality label has passed verification of all the rules.

However, processing the confidentiality shown in Figure 9 and that would result in the SVRL document as shown in Figure 10.

Figure 9: Semantically Invalid Confidentiality Label

```
<?xml version="1.0" encoding="utf-8"?>
<s4774:originatorConfidentialityLabel
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
 <s4774:ConfidentialityInformation>
  <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
  <s4774:Classification>CONFIDENTIAL</s4774:Classification>
  <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
   <s4774:GenericValue>RED</s4774:GenericValue>
  </s4774:Category>
 </s4774:ConfidentialityInformation>
 <s4774:OriginatorID IDType="rfc822Name">alan.ross@reach.nato.int
 </s4774:OriginatorID>
 <s4774:CreationDataTime>2017-03-14T09:00:00</s4774:CreationDataTime>
</s4774:originatorConfidentialityLabel>
```

Figure 10: SVRL document - Semantically Invalid Confidentiality Label

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<svrl:schematron-output title="Schematron for the ACME Governing Policy"
 schemaVersion=""
 xmlns:svrl="http://purl.oclc.org/dsdl/svrl"
 xmlns:schold="http://www.ascc.net/xml/schematron"
 xmlns:iso="http://purl.oclc.org/dsdl/schematron"
 xmlns:xhtml="http://www.w3.org/1999/xhtml"
 xmlns:xs=http://www.w3.org/2001/XMLSchema
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
  <!--
-->
  <svrl:ns-prefix-in-attribute-values prefix="s4774"
   uri="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="slab:PolicyIdentifier" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="slab:Classification[text()='CONFIDENTIAL']" />
  <svrl:failed-assert test="
    ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='MOCK']
    or
   ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='PHONY']"
   location="/*[local-name()='originatorConfidentialityLabel']/*[local-
name()='ConfidentialityInformation']/*[local-name()='Classification']">
    <svrl:text>None of the required categories for classification CONFIDENTIAL
are not present.</svrl:text>
  </svrl:failed-assert>
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="slab:Category" />
  <svrl:active-pattern document="" />
  <svrl:fired-rule context="slab:ConfidentialityInformation" />
  <svrl:fired-rule context="slab:Category[@TagName='Sensitive']" />
  <svrl:fired-rule
   context="slab:Category[@TagName='Sensitive']/slab:GenericValue" />
</svrl:schematron-output>
```

This report shows that same rules have fired as before, (for the Confidentiality Information, PolicyIdentifier, Classification, Category, and the “Sensitive” Category and its values) but an additional rule has fired as the Classification value is “CONFIDENTIAL”.

This additional rule, derived from the ACME SPIF (see fragment below Figure 11) requires that an “ACME CONFIDENTIAL” confidentiality label must also have “Releasable To” category of “MOCK” or “PHONY”.

Figure 11: SPIF securityClassification element with oneOrMore requiredCategories

```
    …
    <spif:securityClassification name="CONFIDENTIAL" lacv="2" hierarchy="2">
      <spif:equivalentClassification applied="both" policyRef="MOCK" lacv="3"/>
      <spif:markingData xml:lang="fr" phrase="CONFIDENTIEL">
        <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:requiredCategory operation="oneOrMore">
        <!-- MOCK -->
        <spif:categoryGroup
          tagSetRef="Releasable To" tagType="enumerated"
          enumType="permissive" lacv="100"/>
        <!-- PHONY -->
        <spif:categoryGroup
          tagSetRef="Releasable To" tagType="enumerated"
          enumType="permissive" lacv="200"/>
      </spif:requiredCategory>
    </spif:securityClassification>
    …
```

Figure 12: Schematron rule for securityClassification with oneOrMore requiredCategories

```
  …
  <iso:pattern name="Classification">
    <iso:rule context="slab:Classification[text()='CONFIDENTIAL']">
      <iso:assert test="../slab:Category[@TagName='Releasable
To']/slab:GenericValue[text()='MOCK'] or ../slab:Category[@TagName='Releasable
To']/slab:GenericValue[text()='PHONY']">None of the required categories for
classification CONFIDENTIAL are not present.</iso:assert>
    </iso:rule>
    …
  <iso:pattern>
  …
```

In turn, a corresponding XML stylesheet template, derived from the Schematron rule using an XML stylesheet, is shown in Figure 13.

Figure 13: XML Stylehseet template for securityClassification with oneOrMore requiredCategories

```
<!--RULE -->
<xsl:template match="slab:Classification[text()='CONFIDENTIAL']"
  priority="1001" mode="M3" xmlns:svrl="http://purl.oclc.org/dsdl/svrl">
  <svrl:fired-rule context="slab:Classification[text()='CONFIDENTIAL']"/>

   <!--ASSERT -->
  <xsl:choose>
    <xsl:when test="
    ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='MOCK']
    or
   ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='PHONY']"
    />
    <xsl:otherwise>
      <svrl:failed-assert test="
    ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='MOCK']
    or
   ../slab:Category[@TagName='Releasable To']/slab:GenericValue[text()='PHONY']"
      >
        <xsl:attribute name="location">
          <xsl:apply-templates select="." mode="schematron-select-full-path" />
        </xsl:attribute>
        <svrl:text>
None of the required categories for classification CONFIDENTIAL are not present.
        </svrl:text>
      </svrl:failed-assert>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:apply-templates select="*" mode="M3" />
</xsl:template>
```

As the example in Figure 13 contains neither of the required “Releasable To” values, the verification of the rule failed and as a result a *svrl:failed-assert* element is included within the report containing details of the failure.

Note that the Schematron rules do not verify that the confidentiality label is syntactically correct. This may be performed by validating the confidentiality label against the ADatP-4774 XML Schema.

3.11.3. Stylesheets

Stylesheets can be generated from the SPIF that can then be used to perform other transformations. This includes:

1. Marking Generation – generate the correct marking from a confidentiality label

3.11.3.1. Marking Generation

The machine-processable confidentiality label is not designed to be read directly by a user. A confidentiality label needs to be rendered as a human-readable confidentiality marking (e.g. that is display in a document footer).

The confidentiality marking is required to be in accordance with the organization policy. As well as the value domains for the confidentiality label, the SPIF contains the details of how to render those values within a confidentiality marking. This includes details such as:

- Value to display – a code may be used in the confidentiality label that must rendered in a form the user can understand e.g. BEL is displayed as “Belgium”
- Language variants e.g. BEL is displayed as “Belgique”.
- Prefixes and suffices e.g. “Releasable To”
- Separators between multiple category values e.g. “,”.

An XML stylesheet can be used to transform a SPIF in to a stylesheet, which can, in turn, be used to render a confidentiality label as a confidentiality marking.

This XML stylesheet is available in the NMRR at:

https://nmrr.ncia.nato.int/rest/doc/NATO/Information%20Assurance/Security%20Policy/spif2marking.xsl

The stylesheet has parameters that allow the selection of:

1. the language (“lang”) of the marking that the resulting stylesheet will generate (note the SPIF must contain details for this language); and
2. the marking (“markingCode”) that will be generated (different markings may be displayed at the document start, compared to a document footer).

Applying this stylesheet to the example ACME policy results in the following XML stylesheet which will generate English language confidentiality marking for confidentiality labels with the ACME policy identifier.

Figure 14: Confidentiality labels with the ACME policy identifier

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
  xmlns:spif="http://www.xmlspif.org/spif">
  <!--
        *** DO NOT EDIT ***
        Automatically generated by spif2marking.xsl at 2017-03-14T09:00:00Z-->
  <xsl:output method="text" indent="yes" encoding="UTF-8" />
  <xsl:template
match="slab:originatorConfidentialityLabel|slab:alternativeConfidentialityLabel"
>
    <xsl:apply-templates select="slab:ConfidentialityInformation" />
  </xsl:template>

  <xsl:template match="slab:ConfidentialityInformation">
    <xsl:variable name="policy">
      <xsl:value-of select="translate(slab:PolicyIdentifier,
       'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
    </xsl:variable>
    <xsl:if test="$policy = 'acme'">
      <xsl:variable name="context">
```

```
        <xsl:value-of
select="translate(slab:Category[@TagName='Context']/slab:GenericValue[not(text()
= 'Releasable')], 'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
      </xsl:variable>
      <xsl:variable name="classification">
        <xsl:value-of select="translate(slab:Classification,
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
      </xsl:variable>
      <xsl:apply-templates select="slab:PolicyIdentifier">
        <xsl:with-param name="context" select="$context" />
        <xsl:with-param name="classification" select="$classification" />
      </xsl:apply-templates>
      <xsl:text xml:space="preserve"> </xsl:text>
      <xsl:apply-templates select="slab:Classification">
        <xsl:with-param name="classification" select="$classification" />
      </xsl:apply-templates>
      <xsl:text xml:space="preserve"> </xsl:text>
      <xsl:apply-templates select="slab:Category[@TagName='Releasable To']"
mode="ReleasableTo">
        <xsl:with-param name="context" select="$context" />
      </xsl:apply-templates>
      <xsl:apply-templates select="slab:Category[@TagName='Administrative']"
mode="Administrative">
        <xsl:with-param name="context" select="$context" />
      </xsl:apply-templates>
      <xsl:apply-templates select="slab:Category[@TagName='Sensitive']"
mode="Sensitive">
        <xsl:with-param name="context" select="$context" />
      </xsl:apply-templates>
    </xsl:if>
  </xsl:template>

  <xsl:template match="slab:PolicyIdentifier">
    <xsl:param name="context" />
    <xsl:param name="classification" />
    <xsl:value-of select="translate(., 'abcdefghijklmnopqrstuvwxyz',
'ABCDEFGHIJKLMNOPQRSTUVWXYZ')" />
  </xsl:template>

  <xsl:template match="slab:Classification">
    <xsl:param name="classification" />
    <xsl:choose>
      <xsl:when test="$classification='public'">PUBLIC</xsl:when>
      <xsl:when test="$classification='confidential'">CONFIDENTIAL</xsl:when>
      <xsl:when test="$classification='internal'">INTERNAL</xsl:when>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="slab:Category" mode="ReleasableTo">
    <xsl:param name="context" />
    <xsl:apply-templates select="slab:GenericValue[translate(text(),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz') != $context]"
mode="ReleasableTo">
      <xsl:sort select="text()" />
    </xsl:apply-templates>
    <xsl:text xml:space="preserve"> </xsl:text>
  </xsl:template>

  <xsl:template match="slab:Category" mode="Administrative">
    <xsl:param name="context" />
    <xsl:apply-templates select="slab:GenericValue[translate(text(),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz') != $context]"
mode="Administrative">
      <xsl:sort select="text()" />
    </xsl:apply-templates>
    <xsl:text xml:space="preserve"> </xsl:text>
  </xsl:template>
  <xsl:template match="slab:Category" mode="Sensitive">
    <xsl:param name="context" />
    <xsl:apply-templates select="slab:GenericValue[translate(text(),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz') != $context]"
mode="Sensitive">
      <xsl:sort select="text()" />
    </xsl:apply-templates>
    <xsl:text xml:space="preserve"> </xsl:text>
 </xsl:template>

  <xsl:template match="slab:GenericValue" mode="ReleasableTo">
    <xsl:variable name="value">
      <xsl:value-of                                         select="translate(.,
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$value='mock'">MOCK</xsl:when>
      <xsl:when test="$value='phony'">PHONY</xsl:when>
    </xsl:choose>
  </xsl:template>
  <xsl:template match="slab:GenericValue" mode="Administrative">
    <xsl:variable name="value">
      <xsl:value-of                                         select="translate(.,
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$value='staff'">STAFF</xsl:when>
      <xsl:when test="$value='finance'">FINANCE</xsl:when>
      <xsl:when test="$value='sales'">SALES</xsl:when>
      <xsl:when test="$value='engineering'">ENGINEERING</xsl:when>
    </xsl:choose>
  </xsl:template>

  <xsl:template match="slab:GenericValue" mode="Sensitive">
    <xsl:variable name="value">
      <xsl:value-of                                         select="translate(.,
'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')" />
    </xsl:variable>
    <xsl:choose>
      <xsl:when test="$value='red'">RED</xsl:when>
      <xsl:when test="$value='blue'">BLUE</xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

3.11.4. XACML Policies

The XML SPIF defines rules to check confidentiality labels against confidentiality clearances as part of an access control decision (ACDF). As the syntax of the confidentiality labels and confidentiality clearances are defined in XML the use of XPATH within XACML Version 3.0 naturally fits as an access control framework that can be leveraged.

The generic XACML provided stylesheet that can be used to generate a XACML policy set based on any XML SPIF is provided below:

Figure 15: XACML Policies

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl"
    xmlns:spif="http://www.xmlspif.org/spif"
                xmlns:ext="http://exslt.org/common"
 >
    <xsl:output method="xml" indent="yes"/>
    <xsl:variable                                           name="permissiveAlg"
select="'urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of'"
/>
    <xsl:variable                                          name="restrictiveAlg"
select="'urn:oasis:names:tc:xacml:1.0:function:string-subset'" />

     <xsl:template match="/">
      <xsl:variable name="policy">
       <xsl:value-of select="//spif:securityPolicyId/@name"/>
      </xsl:variable>
      <xsl:for-each select="//spif:securityCategoryTag">
       <xsl:variable name="tagName" select="translate(@name, ' ','')"/>
       <xsl:variable name="algorithm">
        <xsl:choose>
         <xsl:when           test="contains(@tagType,'permissive')           or
contains(@enumType,'permissive')">
        <xsl:value-of select="$permissiveAlg"/>
         </xsl:when>
         <xsl:otherwise>
          <xsl:value-of select="$restrictiveAlg"/>
         </xsl:otherwise>
        </xsl:choose>
       </xsl:variable>
       <xsl:if     test="not(@tagType     ='tagType7')     and     not(@tagType
='notApplicable')">
       <xsl:result-document    method="xml"    href="{$policy}-Policy-{$tagName}-
XACML.xml">
        <Policy           xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyId="{$policy}-Security-Policy-{$tagName}"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-
unless-permit" Version="1.0">
          <PolicyDefaults>
            <XPathVersion>http://www.w3.org/TR/1999/REC-xpath-
19991116</XPathVersion>
```

```
          </PolicyDefaults>
          <Target>
            <AnyOf>
              <AllOf>
                <!-- Do we have a Policy? NOT APPLICABLE if not -->
                <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string"><xsl:value-of
select="$policy"/></AttributeValue>
                  <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'
and local-name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='PolicyIdentifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                </Match>
                <!-- Do we have a our Category? NOT APPLICABLE if not -->
                <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string"><xsl:value-of
select="@name"/></AttributeValue>
                  <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'
and local-name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Category']/@TagName" DataType="http://www.w3.org/2001/XMLSchema#string"
/>
                </Match>
              </AllOf>
            </AnyOf>
          </Target>
          <Rule RuleId="rule-{$tagName}-policy" Effect="Permit">
            <Description>Rule to match <xsl:value-of select="@name"/>
policy</Description>
            <Condition>
              <!-- Assert Label Category Values has atleast one member in
Clearance Releasable To Values -->
              <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                  <!-- Assert Label Classification Value is subset of Clearance
ClassificationList\Classification Values -->
                  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                    <!-- Assert Label PolicyIdentifier Value equals Clearance
PolicyIdentifier Value -->
                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                      <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
```

```
                        <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'
and local-name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='PolicyIdentifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                      </Apply>
                      <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                        <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and
local-name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='PolicyIdentifier']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                      </Apply>
                    </Apply>
                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-subset">
                      <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'
and local-name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Classification']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                      <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and
local-name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and local-
name()='ClassificationList']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Classification']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                    </Apply>
                  </Apply>
                <Apply FunctionId="{$algorithm}">
                  <AttributeSelector MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'
and local-name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Category'][@TagName='{@name}']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='GenericValue']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
```

```
                  <AttributeSelector                       MustBePresent="false"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
Path="/*[namespace-uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and
local-name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Category'][@TagName='{@name}']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='GenericValue']/text()"
DataType="http://www.w3.org/2001/XMLSchema#string" />
                </Apply>
              </Apply>
            </Condition>
          </Rule>
        </Policy>
       </xsl:result-document>
        </xsl:if>
      </xsl:for-each>
     </xsl:template>
</xsl:stylesheet>
```

For each category tag set in the XML SPIF a separate XACML policy is created of the form:

<policy>-Security-Policy-<tagName>-XACML.xml.

All XACML policies generated are known as the XACML policy set.

The subject for each XACML policy is the confidentiality clearance and the target for each policy is the confidentiality label.

The following XACML design principles have been enabled:

i. Each policy ensures that the PolicyIdentifier value domain of the label is the SPIF policy identifier; and,
ii. Each policy determines that the label contains a: Permissive; or, Restrictive category (relative to the policy).

Each policy contains a single rule for each policy with an Effect value of Permit (in other words the Condition needs to evaluate to true).

The rule asserts the following:

i. The label policy identifier domain values exists in the clearance policy identifier domain value
ii. The label classification domain value exists in the clearance classification list;
iii. For Permissive categories that at least one of the category domain values exists in the clearance
iv. For Restrictive categories that all the category domain values exists in the clearance.

If these conditions are true the XACML decision is PERMIT.

If one or all of these are not met for whatever reason then the combining rule policy of deny-unless-permit ensures that the XACML decision is DENY.

The XACML policy set produced by running the generic XACML provided stylesheet against the ACME SPIF (see ANNEX B) consist of two XACML policies:
ACME Releasable To,

1. ACME Releasable To XACML Policy
2. ACME Sensitive XACML Policy

Figure 16: ACME Releasable To XACML Policy

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        xmlns:spif="http://www.xmlspif.org/spif"
        xmlns:ext="http://exslt.org/common"
        PolicyId="ACME--Policy-ReleasableTo"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:deny-unless-permit"
        Version="1.0">
   <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</XPathVersion>
   </PolicyDefaults>
   <Target>
      <AnyOf>
         <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ACME</AttributeValue>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Match>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Releasable
To</AttributeValue>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
```

```
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Category']/@TagName"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Match>
         </AllOf>
      </AnyOf>
   </Target>
   <Rule RuleId="rule-ReleasableTo-policy" Effect="Permit">
      <Description>Rule to match Releasable To policy</Description>
      <Condition>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
               <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                     <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                        Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                     <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                        Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0'       and       local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
               </Apply>
               <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
subset">
                  <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                     Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
```

```
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Classification']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                     Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and local-
name()='ClassificationList']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Classification']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-
least-one-member-of">
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Category'][@TagName='Releasable To']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='GenericValue']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0' and local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='Category'][@TagName='Releasable To']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0' and local-
name()='GenericValue']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Apply>
         </Apply>
      </Condition>
   </Rule>
</Policy>
```

Figure 17: ACME Sensitive XACML Policy (ACME-Policy-Sensitive-XACML.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        xmlns:spif="http://www.xmlspif.org/spif"
        xmlns:ext="http://exslt.org/common"
        PolicyId="ACME-Governing-Policy-Sensitive"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:deny-unless-permit"
        Version="1.0">
   <PolicyDefaults>
      <XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</XPathVersion>
   </PolicyDefaults>
   <Target>
      <AnyOf>
         <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ACME</AttributeValue>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Match>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Sensitive</AttributeValue>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Category']/@TagName"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Match>
         </AllOf>
      </AnyOf>
   </Target>
   <Rule RuleId="rule-Sensitive-policy" Effect="Permit">
      <Description>Rule to match Sensitive policy</Description>
      <Condition>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
```

```
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
               <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                     <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                        Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
                  <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                     <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                        Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0'       and       local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='PolicyIdentifier']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
               </Apply>
               <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
subset">
                  <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                     Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Classification']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                     Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0'       and       local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0'       and       local-
name()='ClassificationList']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Classification']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
            </Apply>
            <Apply     FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
subset">
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='originatorConfidentialityLabel']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='ConfidentialityInformation']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Category'][@TagName='Sensitive']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='GenericValue']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
               <AttributeSelector MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                  Path="/*[namespace-
uri()='urn:nato:stanag:4774:confidentialityclearance:1:0'       and       local-
name()='ConfidentialityClearance']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='Category'][@TagName='Sensitive']/*[namespace-
uri()='urn:nato:stanag:4774:confidentialitymetadatalabel:1:0'     and     local-
name()='GenericValue']/text()"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </Apply>
         </Apply>
      </Condition>
   </Rule>
</Policy>
```

These XACML policies can be used within the Access Control Framework described in Chapter 6.

INTENTIONALLY BLANK

CHAPTER 4 CONFIDENTIALITY LABEL

4.1. INTRODUCTION

The confidentiality label provides a set of metadata indicating the sensitivity of a piece of information. The confidentiality label is a structured representation and in conjunction with a policy contains controlled values that specifies how that information is to be handled by the systems, applications, services and users being governed by the policy.

The confidentiality label (as specified in ADatP-4774) comprises of confidentiality metadata and supplementary metadata, such as the originator and the time the confidentiality label was created.

The confidentiality label is extensible to support COI-specific or organizational-specific requirements.

An example confidentiality label based on the ACME policy is shown in Figure 18.

Figure 18: ACME Confidentiality Label Example

```
<?xml version="1.0" encoding="utf-8"?>
<s4774:originatorConfidentialityLabel
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
 ReviewDateTime="2022-03-14T09:00:00">
 <s4774:ConfidentialityInformation>
  <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
  <s4774:Classification>INTERNAL</s4774:Classification>
  <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
   <s4774:GenericValue>RED</s4774:GenericValue>
  </s4774:Category>
 </s4774:ConfidentialityInformation>
 <s4774:OriginatorID IDType="rfc822Name">
    alan.ross@reach.nato.int</s4774:OriginatorID>
 <s4774:CreationDataTime>2017-03-14T09:00:00</s4774:CreationDataTime>
</s4774:originatorConfidentialityLabel>
```

4.2. ORIGINATOR AND ALTERNATIVE CONFIDENTIALITY LABEL

The example confidentiality label above is an originator confidentiality label. An originator confidentiality label indicates the confidentiality metadata (hence sensitivity) in the information domain where the information (and confidentiality label) was created and under the control of the originator’s policy.

When information is to be shared amongst partners that may be governed by a different policy the information is required to be protected in a manner commensurate with the sensitivity applied to that information in the originating information domain.

Typically, bilateral agreements between partners are instantiated that support equivalent mappings of subsets of the governing security policies in order for appropriate handling of the information within a recipient information domain.

ADatP-4774 provides (in addition to the originator confidentiality label) an alternative confidentiality label. The alternative confidentiality label provides the equivalent confidentiality metadata as the originator confidentiality label represented under the recipient governing policy.

The example given below illustrates how an originator confidentiality label governed by the ACME Policy is mapped to the alternative confidentiality label governed by the MOCK policy. (The example also illustrates the binding of both confidentiality labels to the information (in this example a detached document) based on STANAG 4778).

Figure 19: Originator confidentiality label governed by the ACME Policy

```
<?xml version="1.0" encoding="utf-8"?>
<s4778:BindingInformation
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd=http://www.w3.org/2001/XMLSchema
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
 xmlns:s4778="urn:nato:stanag:4778:bindinginformation:1:0"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <s4778:MetadataBindingContainer>
  <s4778:MetadataBinding>
   <s4778:Metadata>
    <s4774:originatorConfidentialityLabel ReviewDateTime="2022-03-07T12:30:00Z">
     <s4774:ConfidentialityInformation>
      <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
      <s4774:Classification>CONFIDENTIAL</s4774:Classification>
      <s4774:Category TagName="Releasable To" Type="PERMISSIVE">
       <s4774:GenericValue>MOCK</s4774:GenericValue>
      </s4774:Category>
     </s4774:ConfidentialityInformation>
     <s4774:OriginatorID IDType="rfc822Name">
      alan.ross@reach.nato.int</s4774:OriginatorID>
     <s4774:CreationDateTime>2017-03-07T12:30:00Z</s4774:CreationDateTime>
    </s4774:originatorConfidentialityLabel>
    <s4774:alternativeConfidentialityLabel>
     <s4774:ConfidentialityInformation>
      <s4774:PolicyIdentifier>MOCK</s4774:PolicyIdentifier>
      <s4774:Classification>CONFIDENTIAL</s4774:Classification>
      <s4774:Category TagName="Releasable To" Type="PERMISSIVE">
       <s4774:GenericValue>MOCK</s4774:GenericValue>
      </s4774:Category>
     </s4774:ConfidentialityInformation>
     <s4774:OriginatorID IDType="uniformResourceIdentifier">
      https://nmbs001.ncia.nato.int/metadatabinding.svc</s4774:OriginatorID>
     <s4774:CreationDateTime>2017-03-15T09:30:00Z</s4774:CreationDateTime>
    </s4774:alternativeConfidentialityLabel>
   </s4778:Metadata>
   <s4778:DataReference URI="http://sever.ncia.nato.int/docs/example.doc"/>
  </s4778:MetadataBinding>
 </s4778:MetadataBindingContainer>
</s4778:BindingInformation>
```

It is recommended that the following rules are applied when determining how to handle information labelled based on the NATO Labelling STANAGs.

[Rule - 1] When equivalency mapping is applied between different governing policies, the originator confidentiality label must never be overwritten.

[Rule - 2] Equivalency mapping between different governing policies must result in the equivalent confidentiality label being recorded as the alternative confidentiality label. How and where [Rule - 1] and [Rule - 2] occurs is down to the specific implementations.

[Rule - 3] The application or service processing the binding must first determine if the policy identifier of the originator confidentiality label (contained in the binding) is governed by the same policy governing the application or service. If the governing policy is the same then use the originator confidentiality label for handling the information (or rendering the confidentiality marking to the user).

[Rule - 4] If the governing policy of the application or service is different to the policy identifier (policy owner) of the originator confidentiality label the application or service must iterate through the alternative confidentiality labels until it finds the alternative confidentiality label that is governed by the same policy as the application or service. The application or service shall use the alternative confidentiality label governed under the same policy for handling the information (or rendering the confidentiality marking to the user).

[Rule - 5] When information leaves a partner domain (that contains an alternative confidentiality label governed by that partner domain) the alternative confidentiality label may be removed from the binding.

4.3. SUCCESSOR CONFIDENTIALITY LABEL

The NATO Labelling STANAGs support policies where labelled information is required to be downgraded based on time or mission completion, for example.

The successor confidentiality label supports downgrading of confidentiality labels based on the inclusion of a successor confidentiality label when originally creating and labelling the information or when the ReviewDateTime of the originator confidentiality label has elapsed.

The successor confidentiality label is included within the originator confidentiality label.

The following rules should be applied when using the successor confidentiality label.

[Rule - 1] An application or service processing an originator confidentiality is required to determine the ReviewDateTime (if available). If the

ReviewDateTime has elapsed the application or service is required to use the successor confidentiality label for handling the information (or rendering the confidentiality marking to the user).

[Rule - 2] If the ReviewDateTime is not present in the originator confidentiality label the application or service is required to determine if the SuccessionDateTime attribute of the successor confidentiality label has elapsed. If the SuccessionDateTime has elapsed the application or service is required to use the successor confidentiality label for handling the information (or rendering the confidentiality marking to the user).

An example of a successor confidentiality label is given below:

Figure 20: Successor Confidentiality Label Example

```
<?xml version="1.0" encoding="utf-8"?>
<s4774:originatorConfidentialityLabel
 xmlns:4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
 <s4774:ConfidentialityInformation>
  <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
  <s4774:Classification>INTERNAL</s4774:Classification>
  <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
   <s4774:GenericValue>RED</s4774:GenericValue>
  </s4774:Category>
 </s4774:ConfidentialityInformation>
 <s4774:OriginatorID IDType="rfc822Name">
  alan.ross@reach.nato.int</s4774:OriginatorID>
 <s4774:SuccessionHandling>
  <s4774:SuccessionDateTime>2022-03-14T09:00:00</s4774:SuccessionDateTime>
  <s4774:successorConfidentialityLabel>
   <s4774:ConfidentialityInformation>
    <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
    <s4774:Classification>PUBLIC</s4774:Classification>
   </s4774:ConfidentialityInformation>
  </s4774:successorConfidentialityLabel>
 </s4774:SuccessionHandling>
 <s4774:CreationDataTime>2017-03-14T09:00:00</s4774:CreationDataTime>
</s4774:originatorConfidentialityLabel>
```

4.4. DOMINANT CONFIDENTIALITY LABEL

Use cases exist where multiple confidentiality labels may be bound to information (and subsets thereof). In these use cases there may be a requirement to determine the dominant confidentiality label or the dominant confidentiality marking to display to users (for them to be informed of the overall classification of the information). This section describes the generic rules (based on the XML SPIF specification) that are required to be enforced in order to determine the dominant confidentiality label (for which a dominant confidentiality marking can be rendered).

These rules are to be used as a basis for determine the dominant confidentiality label.

Additional rules may be applied based on the organization policy that is to be enforced.

[Rule - 1] All confidentiality labels must have the same policy identifier (the governing organization policy that is being enforced). Any confidentiality label containing a foreign policy must be mapped to the equivalent governing policy and stored as an alternative confidentiality label.

[Rule - 2] In the case that a confidentiality label cannot be mapped to an equivalent governing policy confidentiality label (for whatever reason) a policy decision is required. This may be that human intervention is required or the default system high confidentiality label is applied.

[Rule - 3] The dominant confidentiality label classification value must be determined based on the classification hierarchy value as specified in the XML SPIF.

For each category that is contained in a confidentiality label the following rules apply:

[Rule - 4] For all confidentiality labels that contain a permissive category (of the same type) the dominant confidentiality label must contain a permissive category (of that type) with the intersection of the category values.

[Rule - 5] For all confidentiality labels that contain a permissive category (of the same type) and the intersection of the category values is empty the dominant confidentiality label must not contain that permissive category.

[Rule - 6] If a confidentiality label (within a set of confidentiality labels) does not contain a permissive category (of the same type) that one or more confidentiality labels contain the dominant confidentiality label must not contain that permissive category.

[Rule - 7] For each confidentiality label that exists with one or more restrictive categories (of the same type) the dominant confidentiality label must contain a restrictive category (of that type) with the union of the category values.

[Rule - 8] For each confidentiality label that exists with one or more informative categories (of the same type) the dominant confidentiality label may contain an informative category (of that type) with the union of the category values.

[Rule - 9] Once a dominant confidentiality label has been evaluated it must be validated against the XML SPIF to determine if it is a valid confidentiality label. If the dominant confidentiality label is not valid, a policy decision is required. This may be that human intervention is required or the default system high confidentiality label is applied.

Table 12 provides a number of examples (based on the ACME XML SPIF and using confidentiality markings for brevity) illustrate the rules defined above.

Table 12: Dominant Label Examples

Confidentiality Label			**Determined by**
Source 1	**Source 2**	**Dominant**	
ACME PUBLIC	MOCK CONFIDENTIAL	ACME CONFIDENTIAL	[Rule - 1] and [Rule - 9]
ACME PUBLIC	MOCK SECRET	Policy decision required	[Rule - 2] and [Rule - 9]
ACME PUBLIC	ACME INTERNAL	ACME INTERNAL	[Rule - 3] and [Rule - 9]
ACME CONFIDENTIAL REL TO MOCK, PHONY	ACME CONFIDENTIAL REL TO MOCK	ACME CONFIDENTIAL REL TO MOCK	[Rule - 4] and [Rule - 9]
ACME CONFIDENTIAL REL TO MOCK, PHONY	ACME INTERNAL	ACME INTERNAL	[Rule - 3] [Rule - 6] and [Rule - 9]
ACME INTERNAL RED	ACME INTERNAL BLUE	ACME INTERNAL RED, BLUE	[Rule - 7] and [Rule - 9]
ACME INTERNAL STAFF	ACME INTERNAL	ACME INTERNAL STAFF	[Rule - 7] and [Rule - 9]
ACME CONFIDENTIAL REL TO MOCK	ACME CONFIDENTIAL REL TO PHONY	Policy decision required	[Rule - 5] and [Rule - 9][9]

[9] The resulting dominant confidentiality label based on [Rule - 5] would result in ACME CONFIDENTIAL. The ACME XML SPIF requires that a CONFIDENTIAL classification must contain the required category Releasable To (hence the Policy decision required result).

CHAPTER 5 CONFIDENTIALITY CLEARANCE

5.1. INTRODUCTION

The confidentiality clearance is an Information Assurance (IA) attribute that is bound to an entity (by being included as an attribute associated with an entity in a directory store, as an X.509 PKIX attribute certificate, or as an attribute included as an identity claim in a security token, such as a SAML assertion token). A confidentiality clearance Access Control Information (ACI) follows a similar structure to the confidentiality label which allows for unambiguous access control decisions to be made based on the confidentiality label associated with the target that is being accessed and the policy associated with the domain where the target resides.

ADatP-4774 defines an XML schema for representing the confidentiality clearance.

The PolicyIdentifier element is used to identify the security policy to which the confidentiality clearance relates. The PolicyIdentifier indicates the semantics of the ClassificationList and Category elements.

An example confidentiality clearance based on the ACME XML SPIF is given below:

Figure 21: ACME XML SPIF Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<sclr:ConfidentialityClearance
 xmlns:sclr="urn:nato:stanag:4774:confidentialityclearance:1:0"
 xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
 <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
 <sclr:ClassificationList>
  <s4774:Classification>PUBLIC</s4774:Classification>
  <s4774:Classification>CONFIDENTIAL</s4774:Classification>
  <s4774:Classification>INTERNAL</s4774:Classification>
 </sclr:ClassificationList>
 <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
  <s4774:GenericValue>RED</s4774:GenericValue>
 </s4774:Category>
 <s4774:Category TagName="Releasable To" Type="PERMISSIVE">
  <s4774:GenericValue>MOCK</s4774:GenericValue>
 </s4774:Category>
</sclr:ConfidentialityClearance>
```

Confidentiality clearances are not required to include any informative categories as these security categories are not enforced in an access control decision.

5.2. IDENTITY ATTRIBUTE

To support access control within an organization or within a federation of organisations attributes associated with identities must be agreed and the dialect must be standardized. The following section proposes two mechanisms for specifying the confidentiality clearance as an identity attribute.

5.2.1. LDAP Attribute

An LDAP attribute that can hold multiple confidentiality clearances (for different policies) has been proposed as shown below:

Figure 22: LDAP Attribute

```
confidentialityClearance
  attributetype (2.16.840.1.101.2.2.1.198
  NAME ‘confidentialityClearance'
  DESC ‘Multi-valued attribute containing one or more clearances as specified in
ADatP-4774'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
)
```

This attribute is an OPTIONAL attribute of the AUXILIARY objectClass aCPAccessControlEntity. Thus, adding the aCPAccessControlEntity auxiliary object class to a LDAP directory entry, allows any entry (e.g. iNetOrgPerson, device, country) to have an associated confidentiality clearance.

A proposal to include the confidentiality clearance attribute as an ACP 133 attribute has been submitted to the CCEB Directory Services Working Group.

5.2.2. SAML Assertion Security Token

Security Assertion Markup Language (SAML) is an open-standards security tokens format that carries identity information. The SAML security token contains assertions that are used to evaluate authorisation decisions based on the claims (or attributes) included within it. The confidentiality clearance is recommended to be included in a SAML security token as follows:

Table 13: SAML Assertion Security Token

Name	URI	Type	Comments
Clearance	urn:nato:stanag:4774: confidentialityclearance:1:0	xs:string	Some implementations may require XML string to be BASE64URL(UTF8(XMLString)) encoded.

5.2.3. JSON WEB Token

JSON Web Token (JWT) is an open-standard security token format (Reference [13]) that carries identity, authentication and contextual information. JWT is widely used for different types of tokens, such as ID Tokens, Access Tokens and Proof of Possession (PoP) Tokens. The JWT contains assertions that are used to evaluate authorisation decisions based on the claims (or attributes) included within it.

A confidentiality clearance is recommended to be included in a JWT as follows:

Table 14: JSON Web Token

JSON Attribute Name	**URI**	**JSON Attribute Value**
aciClr	urn:nato:stanag:4774: confidentialityclearance:1:0	BASE64URL(UTF8(XMLString))

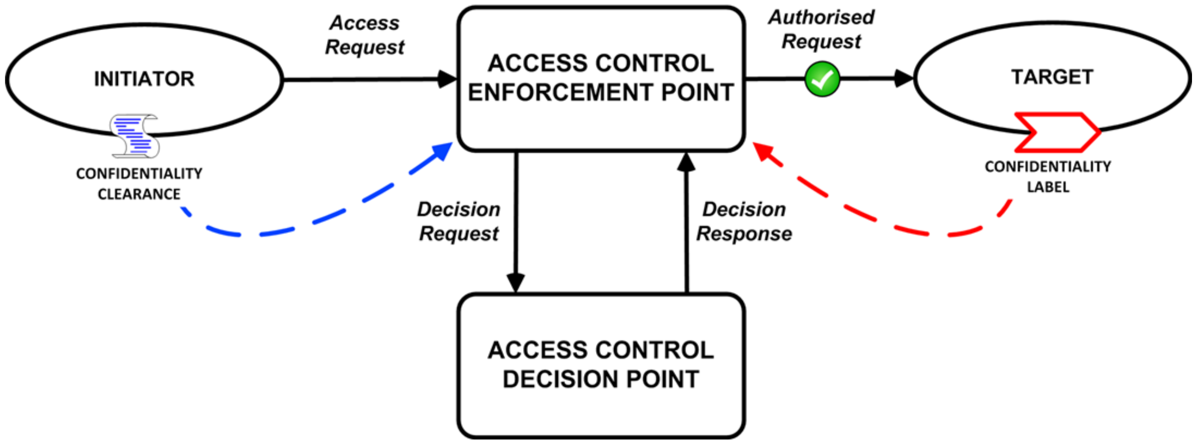
INTENTIONALLY BLANK

CHAPTER 6 ACCESS CONTROL FRAMEWORK

6.1. INTRODUCTION

Confidentiality Label-based Access Control uses initiator and target bound Access Control Information (ACI). The confidentiality label ACI is bound to the target, for example a resource. The confidentiality clearance ACI is bound to the initiator (identity, for example a user or a cross domain boundary interface). The confidentiality label ACI and the confidentiality label ACI are used in making access control decisions, which thereby allows data to be securely accessed or transferred between entities. This type of access control supports environments where there are large number of initiators and large number of targets. Figure 23 depicts the Access Control Concept illustrating the role of the Access Control Enforcement Point and the Access Control Decision Point.

Figure 23: Confidentiality Label Based Access Control



The Confidentiality Label Based Access Control model can be implemented as a component of the Attribute Based Access Control model supported by the eXtensible Access Control Markup Language (XACML).

6.2. XACML MODEL

The Confidentiality Label Based Access Control model is based on the XACML version 3.0 data-flow model (Reference (u)).

The main components of XACML version 3.0 data-flow model are:

1. *Policy Decision Point (PDP)*: the system entity that evaluates the applicable policy or policy set (see Section XACML Policies) and renders an authorization decision. The PDP is the Access Control Decision Point illustrated in Figure 24.

2. *Policy Enforcement Point (PEP)*: the system entity that performs access control, by making decision requests and enforcing authorization decisions based on the confidentiality clearance and confidentiality label. The PEP retrieves the confidentiality label from the information in the request from the initiator. Additionally the PEP retrieves the confidentiality label from the security token in the request or from the Policy Information Point (PIP, see below). The PEP is the Access Control Enforcement Point illustrated in Figure 23;
3. *Policy Administration Point (PAP)*: the system entity that creates a policy or policy set. The PAP manages the XML SPIF(s) for the organization and translates the XML SPIF(s) to XACML policies (see Section XACML Policies);
4. *Policy Information Point (PIP)*: the system entity that acts as a source of attribute values. The PIP may contain the confidentiality clearance attribute for entities within the organization domain.

6.3. ADatP-4774 XACML REQUEST

As mentioned above, the PEP invokes the PDP to make the access control request including the confidentiality label as the target ACI (or resource) and the confidentiality clearance as the initiator ACI (or subject).

The XACML policy uses:

a) *AttributeSelector* elements with attribute category “urn:oasis:names:tc:xacml:3.0:attribute-category:resource” to identify the confidentiality label and XPATH expressions to be evaluated against the confidentiality label content sent over in the request; and,

b) *AttributeSelector* elements with attribute category “urn:oasis:names:tc:xacml:1.0:subject-category:access-subject” to identify the confidentiality clearance and XPATH expressions to be evaluated against the confidentiality clearance content sent over in the request.

As such, the PEP does not need to know the structure of the clearance or the label it just has to send the XML of each over in the Access Control request to the PDP.

An example access control request between the PDP and the PEP based on the ACME security policy is illustrated below:

Figure 24: Access control request between the PDP and the PEP based on the ACME security policy - example

```
<acdf:Request CombinedDecision="false" ReturnPolicyIdList="false"
  xmlns:acdf="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
 <acdf:Attributes
   Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
  <acdf:Content>
   <sclr:ConfidentialityClearance
    xmlns:sclr="urn:nato:stanag:4774:confidentialityclearance:1:0"
    xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
    <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
    <sclr:ClassificationList>
     <s4774:Classification>PUBLIC</s4774:Classification>
     <s4774:Classification>CONFIDENTIAL</s4774:Classification>
     <s4774:Classification>INTERNAL</s4774:Classification>
    </sclr:ClassificationList>
    <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
     <s4774:GenericValue>RED</s4774:GenericValue>
    </s4774:Category>
    <s4774:Category TagName="Releasable To" Type="PERMISSIVE">
     <s4774:GenericValue>MOCK</s4774:GenericValue>
    </s4774:Category>
   </sclr:ConfidentialityClearance>
  </acdf:Content>
 </acdf:Attributes>
 <acdf:Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-
category:resource">
  <acdf:Content>
   <s4774:originatorConfidentialityLabel
    xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
    ReviewDateTime="2022-03-14T09:00:00">
    <s4774:ConfidentialityInformation>
     <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
     <s4774:Classification>INTERNAL</s4774:Classification>
     <s4774:Category TagName="Sensitive" Type="RESTRICTIVE">
      <s4774:GenericValue>RED</s4774:GenericValue>
     </s4774:Category>
    </s4774:ConfidentialityInformation>
    <s4774:OriginatorID IDType="rfc822Name">
     alan.ross@reach.nato.int</s4774:OriginatorID>
    <s4774:CreationDataTime>2017-03-14T09:00:00</s4774:CreationDataTime>
   </s4774:originatorConfidentialityLabel>
  </acdf:Content>
 </acdf:Attributes>
</acdf:Request>
```

INTENTIONALLY BLANK

CHAPTER 7 LABEL CATALOGS

7.1. INTRODUCTION

A confidentiality label is a complex metadata element, which contains many sub-elements. Whilst some applications may allow the user to enter the appropriate values for these sub-elements (for example, Classification, PolicyIdentifier), or generate them automatically on behalf of the user (for example, CreationDateTime, OriginatorID), some applications may wish to allow the user to select a complete, valid confidentiality label from a drop-down list.

A catalog, or code list, of confidentiality labels can maintained and published and used a run-time by an application.

This Implementation Guidance recommends the use of the Genericode Code List specification (Reference [1]). Genericode is also adopted by the NCMS Implementation Guidance (see Chapter 5, Reference [7]).

7.2. GENERICODE LABEL CATALOG

A Genericode codelist can be defined that provides a set of values that can be used the originatorConfidentialityLabel metadata term. For example, Figure 25 shows a Genericode code list of originatorConfidentialityLabels (in column “confidentialityLabel”) which are indexed by the values in the “marking” column, which in this case contains the marking that corresponds to the confidentiality label.

Figure 25: Genericode Label Catalog

```
<gc:CodeList
  xmlns:s4774="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
  xmlns:gc="http://docs.oasis-open.org/codelist/ns/genericode/1.0/"
  xmlns:xhtml="https://www.w3.org/1999/xhtml/">
  <Annotation>
    <Description>
      <xhtml:p xmlns:xhtml="http://www.w3.org/1999/xhtml">Autogenerated from the
'keyword' worksheet in the Excel workbook '20170706_NU_NCMS-
CodeLists.xlsx'.</xhtml:p>
    </Description>
  </Annotation>
  <Identification>
    <ShortName>keyword</ShortName>
    <LongName>Keywords</LongName>
    <Version>1</Version>
    <CanonicalUri>urn:nato:ikm:ncms:1:0:codelist:confidentiality-
labels:acme</CanonicalUri>
    <CanonicalVersionUri>urn:nato:ikm:ncms:1:0:codelist:confidentiality-
labels:acme:1</CanonicalVersionUri>
  </Identification>
  <ColumnSet>
```

```
    <Column Id="marking" Use="required">
      <ShortName>Marking</ShortName>
      <Data Type="xsd:normalizedString" />
    </Column>
    <Column Id="confidentialityLabel" Use="required">
      <ShortName>Confidentiality Label</ShortName>
      <Data Type="originatorConfidentialityLabel"
DataTypeLibrary="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"/>
    </Column>
    <Key Id="codeKey">
      <ShortName>CodeKey</ShortName>
      <ColumnRef Ref="marking"/>
    </Key>
  </ColumnSet>
  <SimpleCodeList>
    <Row>
      <Value ColumnRef="marking">
        <SimpleValue>ACME UNCLASSIFIED</SimpleValue>
      </Value>
      <Value ColumnRef="confidentialityLabel">
        <ComplexValue>
          <s4774:originatorConfidentialityLabel>
            <s4774:ConfidentialityInformation>
              <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
              <s4774:Classification>UNCLASSIFIED</s4774:Classification>
            </s4774:ConfidentialityInformation>
            <s4774:OriginatorID
IDType="uniformResourceIdentifier">https://nmrr.ncia.nato.int/rest/doc/NATO%20In
terim/IA/Security%20Policy/spif2gc.xsl</s4774:OriginatorID>
            <s4774:CreationDateTime>2017-02-
08T16:11:43Z</s4774:CreationDateTime>
          </s4774:originatorConfidentialityLabel>
        </ComplexValue>
      </Value>
    </Row>
    <Row>
      <Value ColumnRef="marking">
        <SimpleValue>ACME RESTRICTED</SimpleValue>
      </Value>
      <Value ColumnRef="confidentialityLabel">
        <ComplexValue>
          <s4774:originatorConfidentialityLabel>
            <s4774:ConfidentialityInformation>
              <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
              <s4774:Classification>RESTRICTED</s4774:Classification>
            </s4774:ConfidentialityInformation>
            <s4774:OriginatorID
IDType="uniformResourceIdentifier">https://nmrr.ncia.nato.int/rest/doc/NATO%20In
terim/IA/Security%20Policy/spif2gc.xsl</s4774:OriginatorID>
            <s4774:CreationDateTime>2017-02-
08T16:11:43Z</s4774:CreationDateTime>
          </s4774:originatorConfidentialityLabel>
        </ComplexValue>
      </Value>
    </Row>
    <Row>
      <Value ColumnRef="marking">
        <SimpleValue>default</SimpleValue>
      </Value>
      <Value ColumnRef="confidentialityLabel">
        <ComplexValue>
          <s4774:originatorConfidentialityLabel>
            <s4774:ConfidentialityInformation>
              <s4774:PolicyIdentifier>ACME</s4774:PolicyIdentifier>
              <s4774:Classification>VERY SECRET</s4774:Classification>
            </s4774:ConfidentialityInformation>
            <s4774:OriginatorID
IDType="uniformResourceIdentifier">https://nmrr.ncia.nato.int/rest/doc/NATO%20In
terim/IA/Security%20Policy/spif2gc.xsl</s4774:OriginatorID>
            <s4774:CreationDateTime>2017-02-
08T16:11:43Z</s4774:CreationDateTime>
          </s4774:originatorConfidentialityLabel>
        </ComplexValue>
      </Value>
    </Row>
  </SimpleCodeList>
</gc:CodeList>
```

There are three confidentiality labels defined, identified by the markings “ACME UNCLASSIFIED”, “ACME RESTRICTED” and “default”.

A client could use this Genericode code list to present the user with a marking and then select the corresponding originatorConfidentialityLabel to include in the metadata binding.

Note that in this case the Genericode code list does not necessarily contain all of the valid values, but rather a subset of commonly used values within the COI, which allows for easier selection.

Note also, as there is no association between the value and the codelist from which it was drawn, the value cannot be validated against the code list. However, the value can be independently validated by other means if required (for example, by validating the originatorConfidentialityLabel against the corresponding SPIF).

INTENTIONALLY BLANK

CHAPTER 8 REFERENCE MATERIALS

8.1. REFERENCES

[1] OASIS Code List Representation (Genericode) Version 1.0, http://docs.oasis-open.org/codelist/cs-genericode-1.0/doc/oasis-code-list-representation-genericode.pdf, 28 December 2007

[2] ADatP-4774 “Confidentiality Metadata Label Syntax”, Edition A, Version 1, December 2017

[3] ADatP-4774.2 “Guidance on the Digital Labelling of NATO Information”, Standard-related Document (SRD), Edition A, Version 1, March 2021

[4] ADatP-4778 “Metadata Binding Mechanism”, Edition A, Version 1, October 2018

[5] ADatP-4778.1, “Implementation Guidance”, Standard-related Document (SRD), Edition A, Version 1 (draft)

[6] ADatP-4778.2, “Profiles for Binding Metadata to a Data Object”, Standard-related Document (SRD), Edition A, Version 1, November 2020

[7] ADatP-5636, “NATO Core Metadata Specification (NCMS)” A, Version 1 (draft)

[8] ADatP-5636.1, “Implementation Guidance”, Standard-related Document (SRD), Edition A, Version 1 (draft)

[9] ISO/IEC-19757-3 “Information Technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation: Schematron”, January 2016

[10] W3C “XML Stylesheet Language Transformation (XSLT), Version 2.0”, https://www.w3.org/TR/xslt/all/, April 2009

[11] C-M(2008)0113, “The Primary Directive on Information Management (PDIM)”, 18 December 2008

[12] IMSM-1049-2019 (INV) “Data Centric Security Vision and Strategy for the Alliance Federation, including the NATO Enterprise” 17th May 2019

[13] RFC 7519 “JSON Web Token (JWT)” May 2015

[14] IETF RFC 2634 “Enhanced Security Services for S/MIME”, https://datatracker.ietf.org/doc/html/rfc2634, June 1999

[15] ITU-T Recommendation X.411 “Information Technology – Message Handling Sustems (MHS) – Message Transfer System: Abstract Service Definition and Procedures” June 1999

[16] C-M(2002)60 “Management of Non-classified NATO Information”

[17] ACP145(A) “Interim Implementation Guide for ACP 123/STANAG 4406 Messaging Services Between Nations”, CCEB, September 2008

INTENTIONALLY BLANK

ANNEX A SPIF XML SCHEMA

A.1. INTRODUCTION

An XML schema for representing a SPIF is published at:
http://www.xmlspif.org/schema/xmlspif.xsd.

The SPIF XML Schema is also held within the NATO Metadata Registry and Repository (NMRR) at:

https://nmrr.ncia.nato.int/rest/doc/NATO/Information%20Assurance/Security%20Policy/spif-21.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns="http://www.xmlspif.org/spif"
  xmlns:spif="http://www.xmlspif.org/spif"
  targetNamespace="http://www.xmlspif.org/spif"
  elementFormDefault="unqualified" attributeFormDefault="unqualified"
  xml:lang="en">

  <xs:annotation>
    <xs:documentation>
      <div>
        <h1>About the SPIF namespace</h1>

        <div class="bodytext">
          <p>
            This schema document describes the Security Policy Information File
(SPIF) namespace, in a form
            suitable for import by other schema documents.
          </p>
          <p>A SPIF describes a confidentiality labelling policy including:</p>
          <ul>
            <li>Policy - its name and other unique identifers</li>
            <li>Classifications - the valid classifications within the policy and
their associated values for use within a confidentiality label and to support the
access control decision function</li>
            <li>Categories - the valid categories with the policy  and their
associated values for use within a confidentiality label and to support the access
control decision function</li>
            <li>Relationships - the relationships (e.g. required, excluded)
between categories and classification and other categories</li>
            <li>Equivalency - the equivalent values of classifications and
categories in another policy</li>
            <li>Marking - instructions how to generate a marking from
classification and category values</li>
```

```
            <li>Input - directions for how the user may enter free-form category
values</li>
          </ul>
          <p>A SPIF can be used to promote the consistent use of confidentiality
labels and marking and may be used, for example, to</p>
          <ul>
            <li>generate a semantically valid confidentiality label</li>
            <li>generate a semantically valid clearance</li>
            <li>verify the validity of a confidentiality label</li>
            <li>generate an corresponding marking</li>
            <li>generate a equivalent  confidentiality label in an alternate
policy</li>
          </ul>
          <p>
            See <a href="http://www.xmlspif.org/">http://www.xmlspif.org</a> for
further information.
          </p>
        </div>
      </div>
    </xs:documentation>
  </xs:annotation>

  <xs:import                    namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="xml.xsd"/>

  <!-- Version -->
  <xs:simpleType name="version">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The version of the schema:</p>
          <ul>
            <li>1.0 - the original schema derived from SDN.801</li>
            <li>
              2.0 - updated to support for:
              <ul>
                <li>Validity period for the whole SPIF and individual category
values</li>
                <li>MarkingData and MarkingQualifier for privacyMark</li>
                <li>MarkingData and MarkingQualifier for privacyMarks</li>
                <li>MarkingData         and         MarkingQualifier         for
securityClassifications</li>
                <li>Constrain the number of privacy mark values that can be
selected</li>
                <li>MarkingQualifier with tagCategory</li>
                <li>Better contraints for the number of allowed tags from a
tagset</li>
                <li>DateFormat for Date category values</li>
                <li>MarkingData and MarkingQualifier for an ObjectIDData</li>
                <li>MarkingData and MarkingQualifier for a SPIF</li>
                <li>Required categories for an equivalentPolicy</li>
                <li>Required categories for an equivalentClassification</li>
                <li>Equivalency between tag sets and allow required categories
for a equivalentTagSet</li>
              </ul>
            </li>
            <li>
              2.1 - small update for :
              <ul>
                <li>Additional <a href="#type_markingCode">markingCode</a> for
policy annotation.</li>
                <li>Additional schema constraints</li>
              </ul>
            </li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="1.0" />
      <xs:enumeration value="2.0" />
      <xs:enumeration value="2.1" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Object Identifier (OID) -->
  <xs:simpleType name="oid">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>An Object IDentifier as a string, for example 1.3.26.1.</p>
          <p>
            For further information see X.680 or <a href="www.oid-
info.com">www.oid-info.com</a>
          </p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <!-- v2.0: Bug fix to pattern -->
      <xs:pattern value="[0-2](\.[0-9]+)+" />
      <!-- End of v2.0 addition -->
    </xs:restriction>
  </xs:simpleType>
  <!-- Label and Certificate Value (Integer) -->
  <xs:simpleType name="lacvInt">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The Label and Certificate Value as an integer member type.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer" />
  </xs:simpleType>
  <!-- Label and Certificate Value (String) -->
  <xs:simpleType name="lacvString">
    <xs:annotation>
      <xs:documentation>
        <xs:documentation>
          <div class="bodytext">
```

```
            <p>The Label and Certificate Value as an string member type.</p>
            <p>Typically used for category values rather than classifications.</p>
          </div>
        </xs:documentation>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <!-- Label and Certificate Value -->
  <xs:simpleType name="lacv">
    <xs:annotation>
      <xs:documentation>
        <xs:documentation>
          <div class="bodytext">
            <p>The Label and Certificate Value type, which is the union of the
lacvInt and lacvString types.</p>
            <p>This value is encoded within the classification and categories in
a confidentiality label or a security clearance (which may be held within a
certificate).</p>
          </div>
        </xs:documentation>
      </xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="lacvInt lacvString" />
  </xs:simpleType>
  <!-- v2.0: new types -->
  <xs:simpleType name="selectionInt">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The selection integer member type, which allows the specification of
the maximum number of selections to be made.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer" />
  </xs:simpleType>
  <xs:simpleType name="selectionString">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The selection string member type, which have specific values: </p>
          <ul>
            <li>unbounded - any number of selections can be made.</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="unbounded" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="selection">
    <xs:annotation>
      <xs:documentation>
```

```
        <div class="bodytext">
          <p>The selection type, which is the union of the selectionInt and
selectionString types, allows the specification of the maximum number of category
values that can be made.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="selectionInt selectionString" />
  </xs:simpleType>
  <xs:simpleType name="equivalencyAction">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The equivalencyAction type indicates the action to be performed on a
category value when mapping a confidentiality label to an equivalent policy.</p>
          <p>The values are: </p>
          <ul>
            <li>discard - it is acceptable that the original category value has
no mapping. The tagSetId and lacv will not be used.</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="discard"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- End of v2.0 additions -->
  <!-- Operation -->
  <xs:simpleType name="operation">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>
            The operation type indicates how many of the categories within an <a
href="#type_optionalCategoryGroup">optionalCategoryGroup</a> are required.
          </p>
          <p>The values are: </p>
          <ul>
            <li>onlyOne - only one of the values identified within the
optionalCategoryGroup are required.</li>
            <li>onlyOne - one or more of the values identified within the
optionalCategoryGroup are required.</li>
            <li>all - all of the values identified within the
optionalCategoryGroup are required.</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="onlyOne" />
      <xs:enumeration value="oneOrMore" />
      <xs:enumeration value="all" />
    </xs:restriction>
  </xs:simpleType>
  <!-- User Input -->
  <xs:simpleType name="userInput">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>
            The format of a <a href="#tag_tagCategory">tagCategory</a> value that
can be entered by the user.
          </p>
          <p>The values are: </p>
          <ul>
            <li>string - an aribtrary string</li>
            <li>integer - an unsigned integer</li>
            <li>date - a date in the format defined by the dateFormat attribute.
</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="string" />
      <xs:enumeration value="integer" />
      <xs:enumeration value="date" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Classification Hierarchy -->
  <xs:simpleType name="hierarchy">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>
            The hierarchy type represents the hierarchical value of a
classification, as opposed to the value that will be placed into a confidentiality
label or certificate (the lacv).
          </p>
          <p>
            The hierachy value is used to determine the dominance of
classification values, for example, when making an access control decision.
          </p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer" />
  </xs:simpleType>
  <!-- Classification  Name -->
  <xs:simpleType name="className">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Classification name - the name of a classification (limited to a
maximum of 256 characters).</p>
          <p>The classification name is the default marking phrase for the
classification.</p>
          <p>The classificaiton name is also used to identify any classifications
that are excluded by a tagCategory.</p>
```

```
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="excludedClass" type="className" />
  <!-- Policy Name -->
  <xs:simpleType name="policyName">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Policy name - the name of a policy (limited to a maximum of 256
characters).</p>
          <p>The policy name is also used to identify the policy for equivalent
policies, classifications and categoryTags.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Marking Phrase -->
  <xs:simpleType name="markingPhrase">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Marking phrase - a string (limited to a maximum of 256 characters)
that will be used in generation a marking from a confidentiality label.</p>
          <p>Multiple marking phrases may be concatenated to generate the final
marking, and different marking phrases may be used in different locations.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Tag Set Name -->
  <xs:simpleType name="tagSetName">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Tag Set Name - the name (limited to a maximum of 256 characters) of
a set of tags (or categories).</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Generalised Time -->
  <xs:simpleType name="genTime">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Generalised Time - a string (limited to a maximum of 256 characters)
that represents a time.</p>
          <p>It may take one of three forms:</p>
          <ul>
            <li>Local time - `YYYYMMDDHH[MM[SS[.fff]]]', where the optional fff
is accurate to three decimal places.</li>
            <li>Universal time (UTC time, or Zulu time) -
`YYYYMMDDHH[MM[SS[.fff]]]Z'.</li>
            <li>Offset from Universal time. `YYYYMMDDHH[MM[SS[.fff]]]+-HHMM'</li>
          </ul>
          <p>Note that these formats are not currently enforced within the
type.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <!-- Marking Code -->
  <xs:simpleType name="markingCode">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Marking Code - the location to display a marking phrase.</p>
          <p>The values are:</p>
          <ul>
            <li>pageTop - Display on top of the page or viewing area e.g.
header.</li>
            <li>pageBottom - Display on bottom of the page or viewing area e.g.
footer.</li>
            <li>pageTopBottom - Display on top and bottom of the page or viewing
area e.g. header and footer.</li>
            <li>documentStart - Display at the start of document e.g. cover
page</li>
            <li>documentEnd - Display at the end of document e.g. end page</li>
            <li>noNameDisplay - Do not display of the classification or category
name; only display the marking phrase.</li>
            <li>noMarkingDisplay - Do not display marking phrase on output;
display marking phrase only during operator input</li>
            <li>suppressClassName - Do not display of the classification name,
but display category.</li>
            <li>firstLineOfText - Display on the first line of the body text e.g.
the body text of an email message.</li>
            <li>lastLineOfText - Display on the last line of the body text e.g.
the body text of an email message.</li>
            <li>subject - The subject of an email message.</li>
            <li>xHeader - The header of an email message. The actual header name
is held within the prefix qualifier.</li>
            <li>portionMarking - Display on a portion of a document</li>
            <li>inputTitle - Display a title, or label, on a GUI element. The
title will be held within the prefix qualifier.</li>
```

```
            <li>waterMark - Display as a watermark behind the main text of a
document.</li>
            <li>replacePolicy - Replace the policy marking phrase.</li>
        </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="pageTop" />
      <xs:enumeration value="pageBottom" />
      <xs:enumeration value="pageTopBottom" />
      <xs:enumeration value="documentStart" />
      <xs:enumeration value="documentEnd" />
      <xs:enumeration value="noNameDisplay" />
      <xs:enumeration value="noMarkingDisplay" />
      <xs:enumeration value="suppressClassName" />
      <!-- v2.0: additional marking codes -->
      <xs:enumeration value="firstLineOfText" />
      <xs:enumeration value="lastLineOfText" />
      <xs:enumeration value="subject" />
      <xs:enumeration value="xHeader" />
      <xs:enumeration value="portionMarking" />
      <xs:enumeration value="inputTitle" />
      <xs:enumeration value="waterMark" />
      <!-- End of v2.0 additions -->
      <!-- v2.1: additional marking code -->
      <xs:enumeration value="replacePolicy" />
      <!-- End of v2.1 additions -->
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="code" type="markingCode" />
  <!-- Tag Type -->
  <xs:simpleType name="tagType">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Tag Type - the type of tag category.</p>
          <p>The values are:</p>
          <ul>
            <li>notApplicable - not applicable</li>
            <li>restrictive - bit set of tag categories where all of the selected
tag categories are required in the clearance.</li>
            <li>enumerated - integer set of tag categories, with tag further
refined by the enumType.</li>
            <li>permissive - bit set of tag categories where at least one of the
selected tag categories are required in the clearance</li>
            <li>tagType7 - (or informative) tag categories that are not used for
access control.</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="notApplicable" />
      <xs:enumeration value="restrictive" />
      <xs:enumeration value="enumerated" />
      <xs:enumeration value="permissive" />
      <xs:enumeration value="tagType7" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Enumerated Type -->
  <xs:simpleType name="enumType">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Enum Type - the type of an enumerated tag category.</p>
          <p>The values are:</p>
          <ul>
            <li>restrictive - tag categories where all of the selected tag
categories are required in the clearance.</li>
            <li>permissive - tag categories where at least one of the selected
tag categories are required in the clearance</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="restrictive" />
      <xs:enumeration value="permissive" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Tag Type 7 Encoding -->
  <xs:simpleType name="tag7Encoding">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Tag7 Encoding - the type of tagType7 (informative) tagType.</p>
          <p>The same value must be used for all tagType7 tag categories within a
catgeory tag set.</p>
          <p>The values are:</p>
          <ul>
            <li>bitSetAttributes - bit set values </li>
            <li>securityAttributes - integer set value c.f. enumerated permissive
or restrictive</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="bitSetAttributes" />
      <xs:enumeration value="securityAttributes" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Qualifier Code -->
  <xs:simpleType name="qualifierCode">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Qualifier Code - indicates how a markingQualifier is to be
applied</p>
```

```
          <p>The values are:</p>
          <ul>
            <li>prefix - as a prefix to the values</li>
            <li>suffix - as a suffix to the values</li>
            <li>separator - as a separator between the values</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="prefix" />
      <xs:enumeration value="suffix" />
      <xs:enumeration value="separator" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Applied -->
  <xs:simpleType name="applied">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>Use this value to indicate that the equivalency may be applied when
considering the clearance of the recipient.</p>
          <p>The values are:</p>
          <ul>
            <li>encrypt - by the originator (e.g. before sending an email to the
recipient)</li>
            <li>decrypt - by the recipient (e.g. before opening an email)</li>
            <li>both - by both the originator (e.g. before sending an email to
the recipient) and recipient (e.g. before opening an email)</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="encrypt"/>
      <xs:enumeration value="decrypt"/>
      <xs:enumeration value="both"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- Colour (W3C) -->
  <xs:simpleType name="colorW3C">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The color W3C member type, which allows the specification of a color
using a standard W3C color name.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="aqua" />
      <xs:enumeration value="black" />
      <xs:enumeration value="blue" />
      <xs:enumeration value="fuschia" />
      <xs:enumeration value="gray" />
      <xs:enumeration value="green" />
      <xs:enumeration value="lime" />
      <xs:enumeration value="maroon" />
      <xs:enumeration value="navy" />
      <xs:enumeration value="olive" />
      <xs:enumeration value="purple" />
      <xs:enumeration value="red" />
      <xs:enumeration value="silver" />
      <xs:enumeration value="teal" />
      <xs:enumeration value="white" />
      <xs:enumeration value="yellow" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Colour (RGB) -->
  <xs:simpleType name="colorRGB">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The color RGB member type, which allows the specification of a color
using Red Green Blue (RGB) values.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:pattern value="#[0-9a-fA-F]{6}" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Colour -->
  <xs:simpleType name="color">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The color type, which is the union of the colorW3C and colorRGB
types, allows the specification of a color.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:union memberTypes="colorW3C colorRGB" />
  </xs:simpleType>
  <!-- v2.0: validity period for elements of the SPIF -->
  <xs:attributeGroup name="validity">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>A group of attributes that determine the period in which the
associated elementy is valid</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:attribute name="notBefore" type="xs:dateTime" />
    <xs:attribute name="notAfter" type="xs:dateTime" />
  </xs:attributeGroup>
  <!-- End of v2.0 addition -->
  <!-- Optional Category Data -->
  <xs:complexType name="optionalCategoryData">
```

```
    <xs:annotation>
      <xs:documentation>Categories associated with specific classification or
category.</xs:documentation>
    </xs:annotation>
    <xs:attribute name="tagSetRef" type="tagSetName" use="required" />
    <xs:attribute name="tagType" type="tagType" use="required" />
    <xs:attribute name="enumType" type="enumType" use="optional" />
    <xs:attribute name="lacv" type="lacv" />
    <!-- It would be useful to have a "name" attribute as an alternative to the
lacv -->
    <xs:attribute name="all" type="xs:boolean" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="categoryGroup" type="optionalCategoryData" />
  <xs:element name="excludedCategory" type="optionalCategoryData" />
  <!-- Optional Category Group -->
  <xs:complexType name="optionalCategoryGroup">
    <xs:sequence>
      <xs:element ref="categoryGroup" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="operation" type="operation" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="requiredCategory" type="optionalCategoryGroup" />
  <!-- Update Info -->
  <xs:complexType name="updateInfo" />
  <xs:element name="updateInfo" type="updateInfo" />
  <!-- Equivalent Classification -->
  <xs:complexType name="equivalentClassification">
    <!-- v2.0: requiredCategories for this classification -->
    <xs:sequence>
      <xs:element ref="requiredCategory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <!-- End of v2.0 addition-->
    <xs:attribute name="policyRef" type="policyName" use="required" />
    <xs:attribute name="lacv" type="lacvInt" use="required" />
    <xs:attribute name="applied" type="applied" use="required" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="equivalentClassification" type="equivalentClassification" />
  <!-- Equivalent Policy -->
  <xs:complexType name="equivalentPolicy">
    <xs:annotation>
      <xs:documentation>Consolidates all equivalent policies in the
SPIF</xs:documentation>
    </xs:annotation>
    <!-- v2.0: requiredCategories when mapping into an equivalent policy. E.g. to
include a REL TO relating to original policy -->
    <xs:sequence>
      <xs:element ref="requiredCategory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <!-- End of v2.0 addition -->
    <xs:attribute name="name" type="policyName" use="required" />
    <xs:attribute name="id" type="oid" use="required" />
    <xs:attribute name="userRefURI" type="xs:anyURI" />
    <xs:attribute name="docRefURI" type="xs:anyURI" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="equivalentPolicy" type="equivalentPolicy" />
  <!-- Equivalent Policies -->
  <xs:complexType name="equivalentPolicies">
    <xs:sequence>
      <xs:element ref="equivalentPolicy" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="equivalentPolicies" type="equivalentPolicies" />
  <!-- Privacy Mark -->
  <xs:complexType name="privacyMark">
    <xs:annotation>
      <xs:documentation>A privacy mark that may be used in the
label.</xs:documentation>
    </xs:annotation>
    <!-- v2.0: markingData and markingQualifier for privacyMark -->
    <xs:sequence>
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <!-- End of v2.0 addition -->
    <xs:attribute name="name" type="xs:string" />
    <xs:attribute name="obsolete" type="xs:boolean" default="false" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="privacyMark" type="privacyMark" />
  <!-- Privacy Marks -->
  <xs:complexType name="privacyMarks">
    <xs:sequence>
      <xs:element ref="privacyMark" maxOccurs="unbounded" />
      <!-- v2.0: marking Data and markingQualifier for privacyMarks-->
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
      <!-- End of v2.0 addition -->
    </xs:sequence>
    <!-- v2.0: constrain the number of privacy marks -->
    <xs:attribute name="maxSelection" type="selection" default="unbounded" />
    <xs:attribute name="minSelection" type="selection" default="unbounded" />
    <!-- End of v2.0 additions -->
  </xs:complexType>
  <xs:element name="privacyMarks" type="privacyMarks" />
  <!-- Marking Data -->
  <xs:complexType name="markingData">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The markingData identifies the marking information attached to the
data object</p>
            <p>It consists of:</p>
          <ul>
            <li>phrase - the marking phrase</li>
            <li>code - a sequence of marking codes which identifies where the
marking phrase is physically applied.</li>
```

```
          </ul>
          <p>If the markingPhrase is absent, then the markingCode applies to the
SecurityClassification classificationName, TagCategories secCategoryName or SPIF
securityPolicyId name, depending on which component includes the markingData.</p>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="code" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="phrase" type="markingPhrase" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="markingData" type="markingData" />
  <!-- Classification -->
  <xs:complexType name="securityClassification">
    <xs:annotation>
      <xs:documentation> Classification</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element          ref="equivalentClassification"         minOccurs="0"
maxOccurs="unbounded" />
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="requiredCategory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="name" type="className" use="required" />
    <xs:attribute name="color" type="color" />
    <xs:attribute name="lacv" type="lacvInt" use="required" />
    <xs:attribute name="hierarchy" type="hierarchy" use="required" />
    <xs:attribute name="obsolete" type="xs:boolean" default="false" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="securityClassification" type="securityClassification" />
  <!-- Classifications -->
  <xs:complexType name="securityClassifications">
    <xs:sequence>
      <xs:element ref="securityClassification" maxOccurs="unbounded" />
      <!-- v2.0: markingData and markingQualifiers for securityClassifications -
->
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
      <!-- End of v2.0 addition -->
    </xs:sequence>
    <!-- v2.0: name for the classifications and allow any other attribute to be
associated with them -->
    <xs:attribute name="name" type="xs:string"/>
    <xs:anyAttribute/>
    <!-- End of v2.0 addition -->
  </xs:complexType>
  <xs:element name="securityClassifications" type="securityClassifications" />
  <!-- Equivalent Category Tag -->
  <xs:complexType name="equivalentSecCategoryTag">
    <xs:attribute name="policyRef" type="policyName" use="required" />
    <xs:attribute name="tagSetId" type="oid" use="required" />
    <xs:attribute name="tagType" type="tagType" use="required" />
    <xs:attribute name="enumType" type="enumType" />
    <xs:attribute name="lacv" type="lacv" use="required" />
    <xs:attribute name="applied" type="applied" use="required" />
    <!-- v2.0: action to be performed during equivalency -->
    <xs:attribute name="action" type="equivalencyAction" />
    <!-- End of v2.0 addition -->
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="equivalentSecCategoryTag" type="equivalentSecCategoryTag" />
  <!-- Tag Category -->
  <xs:complexType name="tagCategory">
    <xs:sequence>
      <xs:element          ref="equivalentSecCategoryTag"          minOccurs="0"
maxOccurs="unbounded" />
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
      <!-- v2.0: associate markingQualifiers with tagCategories -->
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
      <!-- End of v2.0 addition -->
      <xs:element ref="excludedClass" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="requiredCategory" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="excludedCategory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required" />
    <xs:attribute name="lacv" type="lacv" use="required" />
    <xs:attribute name="userInput" type="userInput" />
    <xs:attribute name="requiredClass" type="className" />
    <xs:attribute name="obsolete" type="xs:boolean" default="false" />
    <!-- v2.0: format of dates to be entered -->
    <xs:attribute name="dateFormat" type="xs:string">
      <xs:annotation>
        <xs:documentation>Format as defined in ISO 8601</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="validity" />
    <!-- End of v2.0 addition -->
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="tagCategory" type="tagCategory" />
  <!-- Qualifier -->
  <xs:complexType name="qualifier">
    <xs:attribute name="markingQualifier" type="markingPhrase" use="required" />
    <xs:attribute name="qualifierCode" type="qualifierCode" use="required" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="qualifier" type="qualifier" />
  <!-- Marking Qualifier -->
  <xs:complexType name="markingQualifier">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The markingQualifier qualifies the markingData associated with a data
object (e.g. it specifies a suffix or a prefix). </p>
          <p>It consists of:</p>
          <ul>
```

```
            <li>qualifier - a qualifier (e.g. a suffix, prefix or separator)</li>
            <li>markingCode - a code which identifies where the phrase is to be
physically applied.</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="qualifier" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="markingCode" type="markingCode" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="markingQualifier" type="markingQualifier" />
  <!-- Category Tag -->
  <xs:complexType name="securityCategoryTag">
    <xs:sequence>
      <xs:element ref="tagCategory" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" />
    <xs:attribute name="tagType" type="tagType" use="required" />
    <xs:attribute name="enumType" type="enumType" />
    <xs:attribute name="tag7Encoding" type="tag7Encoding" />
    <xs:attribute name="singleSelection" type="xs:boolean" default="false" />
    <!-- v2.0: Constrain the number of selections (beyond one or any) -->
    <xs:attribute name="maxSelection" type="selection" default="unbounded" />
    <xs:attribute name="minSelection" type="selection" default="unbounded" />
    <!-- End of v2.0 addition -->
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="securityCategoryTag" type="securityCategoryTag" />
  <!--  Category Tag Set -->
  <xs:complexType name="securityCategoryTagSet">
    <xs:sequence>
      <xs:element ref="securityCategoryTag" maxOccurs="unbounded" />
      <!-- v2.0: mark equivalency between tag sets (e.g. REL TO, EYES ONLY) and
allows categories to be added to equivalent label based on tag set presence (e.g.
add REL TO NATO if a REL TO tagset is present) -->
      <xs:element      ref="equivalentSecurityCategoryTagSet"      minOccurs="0"
maxOccurs="unbounded" />
      <!-- End of v2.0 addition -->
    </xs:sequence>
    <xs:attribute name="name" type="tagSetName" use="required" />
    <xs:attribute name="id" type="oid" use="required" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="securityCategoryTagSet" type="securityCategoryTagSet" />
  <!-- v2.0: to allow requiredCategories for an equivalent policy when a tag set
is present (and possibly allow algorithmic mapping of tag sets ) -->
  <!-- Equivalent Category Tag Set -->
  <xs:complexType name="equivalentSecurityCategoryTagSet">
    <xs:sequence>
      <xs:element ref="requiredCategory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="policyRef" type="policyName" use="required" />
    <xs:attribute name="name" type="tagSetName" />
    <xs:attribute name="id" type="oid" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element                            name="equivalentSecurityCategoryTagSet"
type="equivalentSecurityCategoryTagSet" />
  <!-- End of v2.0 addition -->
  <!-- Category Tag Sets -->
  <xs:complexType name="securityCategoryTagSets">
    <xs:sequence>
      <xs:element ref="securityCategoryTagSet" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="securityCategoryTagSets" type="securityCategoryTagSets" />
  <!-- Object ID Data -->
  <xs:complexType name="objectIdData">
    <!-- v2.0: associate markingData and markingQualifiers with a policy -->
    <xs:sequence>
      <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <!-- End of v2.0 addition -->
    <xs:attribute name="name" type="policyName" use="required" />
    <xs:attribute name="id" type="oid" use="required" />
    <xs:anyAttribute />
  </xs:complexType>
  <xs:element name="defaultSecurityPolicyId" type="objectIdData" />
  <xs:element name="securityPolicyId" type="objectIdData" />
  <!-- Extensions -->
  <xs:complexType name="extensions">
    <xs:annotation>
      <xs:documentation>A set of vendor-specific extensions</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="extensions" type="extensions" />
  <!-- Security Policy Information File -->
  <xs:element name="SPIF">
    <xs:annotation>
      <xs:documentation>
        <div class="bodytext">
          <p>The complete Security Policy Information.</p>
          <p>It contains:</p>
          <ul>
            <li>defaultSecurityPolicyId - identifies the security policy which
will apply if data is received without a confidentiality label</li>
            <li>securityPolicyId - identifies the security policy to which the
SPIF applies</li>
            <li>updateInfo - not currently used</li>
            <li>securityClassifications - the set of classifications defined
within the policy, together with their equivalency mappings</li>
```

```
            <li>securityCategoryTagSets - the set of category tags defined within
the policy, together with their equivalency mappings</li>
            <li>privacyMarks - the set of privacy marks defined within the policy,
together with their equivalency mappings</li>
            <li>equivalentPolicies - consolidated list of all equivalent policies
used within the SPIF</li>
            <li>markingData</li>
            <li>markingQualifer</li>
            <li>extensions - provides a mechanism to include additional
capabilities as future requirements are identified.</li>
            <li>schemaVersion - the version of the schema being used</li>
            <li>version - the version of the SPIF. Changes to the SPIF will
generally update the version.</li>
            <li>creationDate - the date the SPIF was created/updated</li>
            <li>originatorDN - the distinguished name (DN) of creator of the SPIF,
using an LDAP encoding as defined in RFC 4514.</li>
            <li>keyIdentifier identifies the key used to sign the SPIF.</li>
            <li>privilegeId - identifies the syntax that is included in the
clearance attribute category of relying certificates</li>
            <li>rbacId - identifies the syntax of the  category that is used in
conjunction with the SPIF</li>
            <li>userRefURI - a reference to a document that provides further
information on the use of the values defined within the SPIF. </li>
            <li>docRefURI - a reference to a document that provides information
on the values defined within the SPIF.</li>
            <li>validity - the validaty of the SPIF (e.g. it may only be used for
a specific exercise)</li>
          </ul>
        </div>
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="defaultSecurityPolicyId" minOccurs="0" />
        <xs:element ref="securityPolicyId" />
        <xs:element ref="updateInfo" minOccurs="0" />
        <xs:element ref="securityClassifications" />
        <xs:element ref="securityCategoryTagSets" minOccurs="0" />
        <xs:element ref="privacyMarks" minOccurs="0" />
        <xs:element ref="equivalentPolicies" minOccurs="0" />
        <!-- v2.0: markingData and markingQualifier with a SPIF -->
        <xs:element ref="markingData" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="markingQualifier" minOccurs="0" maxOccurs="unbounded" />
        <!-- End of v2.0 addition -->
        <xs:element ref="extensions" minOccurs="0" />
      </xs:sequence>
      <xs:attribute name="schemaVersion" type="version" use="required" />
      <xs:attribute name="version" type="xs:integer" default="1" />
      <xs:attribute name="creationDate" type="genTime" use="required" />
      <xs:attribute name="originatorDN" type="xs:string" use="required" />
      <xs:attribute name="keyIdentifier" type="xs:string" use="required" />
      <xs:attribute name="privilegeId" type="oid" use="required" />
      <xs:attribute name="rbacId" type="oid" use="required" />
      <xs:attribute name="userRefURI" type="xs:anyURI" />
      <xs:attribute name="docRefURI" type="xs:anyURI" />
      <!-- v2.0: validity of SPIF -->
      <xs:attributeGroup ref="validity" />
      <!-- End of v2.0 addition -->
      <xs:anyAttribute />
    </xs:complexType>
    <!-- Constraints -->
    <xs:unique name="uqPolicyId">
      <xs:selector     xpath="spif:equivalentPolicies/spif:equivalentPolicy     |
spif:defaultSecurityPolicyId" />
      <xs:field xpath="@id" />
    </xs:unique>
    <xs:key name="stPolicy">
      <xs:selector     xpath="spif:equivalentPolicies/spif:equivalentPolicy     |
spif:defaultSecurityPolicyId" />
      <xs:field xpath="@name" />
    </xs:key>
    <xs:keyref name="refPolicy1" refer="stPolicy">
      <xs:selector
xpath="spif:securityClassifications/spif:securityClassification/spif:equivalentC
lassification" />
      <xs:field xpath="@policyRef" />
    </xs:keyref>
    <xs:keyref name="refPolicy2" refer="stPolicy">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet/spif:securityCat
egoryTag/spif:tagCategory/spif:equivalentSecCategoryTag" />
      <xs:field xpath="@policyRef" />
    </xs:keyref>
    <xs:key name="stClass">
      <xs:selector
xpath="spif:securityClassifications/spif:securityClassification" />
      <xs:field xpath="@name" />
    </xs:key>
    <xs:unique name="uqClass">
      <xs:selector
xpath="spif:securityClassifications/spif:securityClassification" />
      <xs:field xpath="@lacv" />
    </xs:unique>
    <xs:unique name="uqClassColor">
      <xs:selector
xpath="spif:securityClassifications/spif:securityClassification" />
      <xs:field xpath="@color" />
    </xs:unique>
    <!-- v2.1: Uniqueness of securityTagSet ids -->
    <xs:unique name="uqSecurityCategoryTagSetId">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet" />
      <xs:field xpath="@id" />
    </xs:unique>
    <!-- End of v2.1 addition -->
    <xs:keyref name="refClass1" refer="stClass">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet/spif:securityCat
egoryTag/spif:tagCategory" />
      <xs:field xpath="@requiredClass" />
    </xs:keyref>
    <xs:keyref name="refClass2" refer="stClass">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet/spif:securityCat
egoryTag/spif:tagCategory/spif:excludedClass" />
      <xs:field xpath="." />
    </xs:keyref>
    <xs:key name="stTagSet">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet" />
      <xs:field xpath="@name" />
    </xs:key>
    <xs:keyref name="refTagSet1" refer="stTagSet">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet/spif:securityCat
egoryTag/tagCategory/spif:requiredCategory/spif:categoryGroup" />
      <xs:field xpath="@tagSetRef" />
    </xs:keyref>
    <xs:keyref name="refTagSet2" refer="stTagSet">
      <xs:selector
xpath="spif:securityCategoryTagSets/spif:securityCategoryTagSet/spif:securityCat
egoryTag/spif:tagCategory/spif:excludedCategory" />
      <xs:field xpath="@tagSetRef" />
    </xs:keyref>
    <xs:keyref name="refTagSet3" refer="stTagSet">
      <xs:selector
xpath="spif:securityClassifications/spif:securityClassification/spif:requiredCat
egory/spif:categoryGroup" />
      <xs:field xpath="@tagSetRef" />
    </xs:keyref>
  </xs:element>
</xs:schema>
```

INTENTIONALLY BLANK

ANNEX B EXAMPLE ACME SPIF

B.1. INTRODUCTION

The following SPIF, for the ACME security policy, uses the XML SPIF schema to encapsulate a fictitious policy illustrating the elements and attributes described in Figure 26.

The ACME policy has:

- Policy: ACME
- classifications: “PUBLIC”, “CONFIDENTIAL” and “INTERNAL”
- “Releasable To” permissive category with values: “MOCK” and “PHONY”.
- “Administrative” informative category with values: “STAFF”, “FINANCE”, “SALES” and “ENGINEERING”
- “Sensitive” restrictive category with values: “RED” and “BLUE”

- The ACME policy also includes equivalencies to the “MOCK” policy where appropriate.

This example policy, and the associated SPIF, are used in a number of the examples within this document.

Figure 26 ACME Policy Example

```
<?xml version="1.0" encoding="UTF-8"?>
<spif:SPIF
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:spif="http://www.xmlspif.org/spif"
  xsi:schemaLocation="http://www.xmlspif.org/spif spif.xsd"
  schemaVersion="2.1" version="1" creationDate="20170314090000Z"
  originatorDN="CN=Alan Ross,O=SMHS Ltd,C=GB"
  keyIdentifier="6AA4BA9F66BFCD44" privilegeId="1.3.26.0.4774.5.24.1"
  rbacId="1.3.26.0.4774.5.24.1">
  <spif:securityPolicyId name="ACME" id="1.3.6.1.4.1.31778.110.1"/>
  <spif:securityClassifications>
    <spif:securityClassification name="PUBLIC" lacv="1" hierarchy="1">
      <spif:markingData xml:lang="fr" phrase="PUBLIC">
        <spif:code>documentStart</spif:code>
      </spif:markingData>
    </spif:securityClassification>
    <spif:securityClassification name="CONFIDENTIAL" lacv="2" hierarchy="2">
      <spif:equivalentClassification applied="both" policyRef="MOCK" lacv="3"/>
      <spif:markingData xml:lang="fr" phrase="CONFIDENTIEL">
        <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:requiredCategory operation="oneOrMore">
        <!-- MOCK -->
        <spif:categoryGroup
          tagSetRef="Releasable To" tagType="enumerated"
          enumType="permissive" lacv="100"/>
        <!-- PHONY -->
        <spif:categoryGroup
          tagSetRef="Releasable To" tagType="enumerated"
          enumType="permissive" lacv="200"/>
      </spif:requiredCategory>
    </spif:securityClassification>
   <spif:securityClassification name="INTERNAL" lacv="3" hierarchy="3">
    <spif:markingData xml:lang="fr" phrase="INTERNE">
     <spif:code>documentStart</spif:code>
    </spif:markingData>
   </spif:securityClassification>
  </spif:securityClassifications>
  <spif:securityCategoryTagSets>
    <spif:securityCategoryTagSet name="Releasable To"
      id="1.3.6.1.4.1.31778.111.1">
      <spif:securityCategoryTag
        name="Releasable To" tagType="enumerated"
        enumType="permissive" singleSelection="false">
        <spif:tagCategory name="MOCK" lacv="100">
         <equivalentSecCategoryTag policyRef="MOCK"
           tagSetId="1.3.6.1.4.1.31778.121.1" tagType="enumerated"
           enumType="permissive" lacv="1000" applied="both" />
         <spif:markingData phrase="MOCK">
            <spif:code>documentStart</spif:code>
          </spif:markingData>
          <spif:markingData xml:lang="fr" phrase="MOQUER">
            <spif:code>documentStart</spif:code>
          </spif:markingData>
         <spif:excludedClass>PUBLIC</spif:excludedClass>
         <spif:excludedClass>INTERNAL</spif:excludedClass>
        </spif:tagCategory>
        <spif:tagCategory name="PHONY" lacv="200">
         <equivalentSecCategoryTag policyRef="MOCK"
           tagSetId="1.3.6.1.4.1.31778.121.1" tagType="enumerated"
           enumType="permissive" lacv="2000" applied="both" />
          <spif:markingData phrase="PHONY">
            <spif:code>documentStart</spif:code>
          </spif:markingData>
          <spif:markingData xml:lang="fr" phrase="FAUX">
            <spif:code>documentStart</spif:code>
          </spif:markingData>
         <spif:excludedClass>PUBLIC</spif:excludedClass>
         <spif:excludedClass>INTERNAL</spif:excludedClass>
        </spif:tagCategory>
        <spif:markingQualifier markingCode="pageTop">
          <spif:qualifier markingQualifier="REL TO " qualifierCode="prefix"/>
          <spif:qualifier xml:lang="fr" markingQualifier="REL TO "
            qualifierCode="prefix"/>
          <spif:qualifier markingQualifier="," qualifierCode="separator"/>
        </spif:markingQualifier>
      </spif:securityCategoryTag>
    </spif:securityCategoryTagSet>
   <spif:securityCategoryTagSet name="Administrative"
     id="1.3.6.1.4.1.31778.111.2">
    <spif:securityCategoryTag name="Administrative" tagType="tagType7"
      tag7Encoding="bitSetAttributes" singleSelection="false">
     <spif:tagCategory name="STAFF" lacv="1">
```

```
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:tagCategory name="FINANCE" lacv="2">
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:tagCategory name="SALES" lacv="3">
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:tagCategory name="ENGINEERING" lacv="4">
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:markingQualifier markingCode="pageTop">
      <spif:qualifier markingQualifier=" " qualifierCode="separator"/>
     </spif:markingQualifier>
    </spif:securityCategoryTag>
   </spif:securityCategoryTagSet>
   <spif:securityCategoryTagSet name="Sensitive" id="1.3.6.1.4.1.31778.111.3">
    <spif:securityCategoryTag name="Sensitive" tagType="restrictive"
     singleSelection="false">
     <spif:tagCategory name="RED" lacv="1">
      <spif:markingData phrase="RED">
       <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:markingData xml:lang="fr" phrase="ROUGE">
       <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:tagCategory name="BLUE"  lacv="2">
      <spif:markingData phrase="BLUE">
       <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:markingData xml:lang="fr" phrase="BLEU">
       <spif:code>documentStart</spif:code>
      </spif:markingData>
      <spif:excludedClass>PUBLIC</spif:excludedClass>
      <spif:excludedClass>CONFIDENTIAL</spif:excludedClass>
     </spif:tagCategory>
     <spif:markingQualifier markingCode="pageTop">
      <spif:qualifier markingQualifier=" " qualifierCode="separator"/>
     </spif:markingQualifier>
    </spif:securityCategoryTag>
   </spif:securityCategoryTagSet>
  </spif:securityCategoryTagSets>
  <spif:equivalentPolicies>
    <spif:equivalentPolicy name="MOCK" id="1.3.6.1.4.1.31778.120.1"
     docRefURI="MOCK Security Policy.spif"/>
  </spif:equivalentPolicies>
</spif:SPIF>
```

ADatP-4774.1(A)(1)