Profile for Binding Metadata to a Data Object

A.1 Applicability

The Metadata Binding Profiles shall be used in the implementation of NATO Common Funded Systems.

NATO systems and services must be able to apply, review and verify Binding Information. These functions may be performed locally or remotely i.e. by a local client API or through a service. Metadata assigned to a data object, especially those elements consisting of confidentiality (or sensitivity) metadata labels, must be properly bound to the data object to facilitate improved information management, consistent application of security policies and enhanced collaboration within the NATO enterprise and between NATO and NATO partners.

The Metadata Binding Mechanism and the complementary Binding Profiles use only recognized international and industry standards. The standards used are consistent with the use already declared by other services.

The Metadata Binding Mechanism and the Binding Profiles employ modular techniques and are extensible to provide agility in adapting to new use cases or scenarios. In other words, the Metadata Binding Mechanism is designed to support the binding of any metadata (and formats) to any type of finite data object.

This profile supports improved interoperability by providing a standard mechanism to bind metadata to data objects. It provides both a standard syntax to represent the cryptographic and non-cryptographic bindings of arbitrary metadata (e.g. core, COI and national metadata) together with a specification of how the binding should be carried with the data object. This profile allows the NATO Enterprise, COIs and Nations to exchange metadata about a data object in a standard, extensible, manner and thus support the correct handling of the data object in accordance with its associatred metadata.

A.1.1 Relationship to the NATO C3 Taxonomy

Existing applications do not support labelling of information in accordance to STANAG 4774 and the Binding Profiles (described in this document). A labelling service, the NATO Metadata Binding Service (NMBS, [NCI Agency TR/2012/SPW007959/02, 2012]), offers a solution to this problem, by providing a capability within the network to support the verification and generation of STANAG 4774 confidentiality labels (and other metadata) bound to data objects. The NMBS offers the following interfaces to consumers: SOAP-based; REST-based; C# API; and Java API. The NMBS architecture allows for deploying the NMBS as a remote service or as a plugin for a client application, such as JChat. For legacy systems and applications that label information in a Community of Interest (COI) specific approach, the NMBS provides a configurable mechanism for supporting conversion between COI-specific labels to STANAG 4774 and equivalencies between different security policies.

The NMBS, shown in Figure 1, provides a flexible and extensible service. The NMBS supports service clients by providing metadata enumerations (of which 'confidentiality label catalogues' are an example), binding metadata to information objects (in order to produce binding data) and validating binding data. When binding metadata to data objects the NMBS will handle the metadata in the form of a label, e.g. a STANAG 4774 Confidentiality Label; the NMBS can therefore also be regarded as a 'labelling service'.

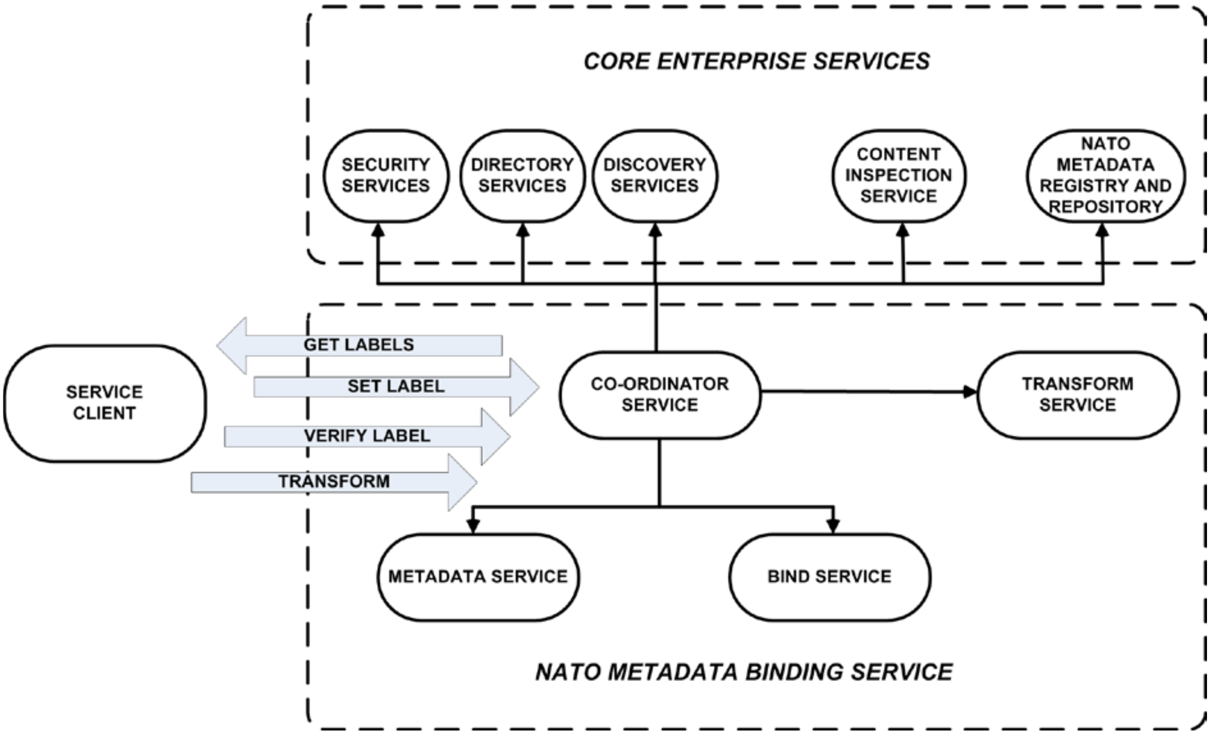


Figure 1: NMBS overall architecture

The NMBS makes use of existing Core Enterprise Services such as Security Services (including PKI services) and the Enterprise Directory Service as shown in Figure 1.

The NMBS is comprised of the following services, as depicted in Figure 1:

- Coordinator Service
  - Orchestrates the services required to support the requested metadata selection or binding operation;
  - A Service Client will find the Coordinator Service in the Service registry when discovering the NMBS; and,
  - Discovers the CES Services required to support the requested binding service.
- Metadata Service
  - Generates Metadata Enumerations from Metadata Policy;
  - Verifies Metadata against Metadata Policy; and,
  - Assigns Metadata to the Data Object.
- Transform Service
  - Supports Metadata Interoperability through equivalency mapping;

  - Notified when supporting artefacts (such as metadata policies) are created, updated or deleted in the NATO Metadata Registry and Repository (NMRR).
- Bind Service
  - Binds Metadata to the Data Object; and,
  - Verifies binding data.

A.1.1.1 Core Enterprise Services

Due to the generic nature of the NMBS, there are multiple options for its location within the taxonomy. The sections below discuss the following options:

1) NMBS as a CIS Security service within Infrastructure Services;

Figure 2 contains an excerpt of the Transformational C3 Classification Taxonomy [TIDE C3 Classification Taxonomy v2.0] which shows where the NMBS could fit as a CIS Security service within the SOA Platform Services (solid black circle), for the case that the type of metadata bound is CIS security metadata, such as STANAG 4774 confidentiality labels.

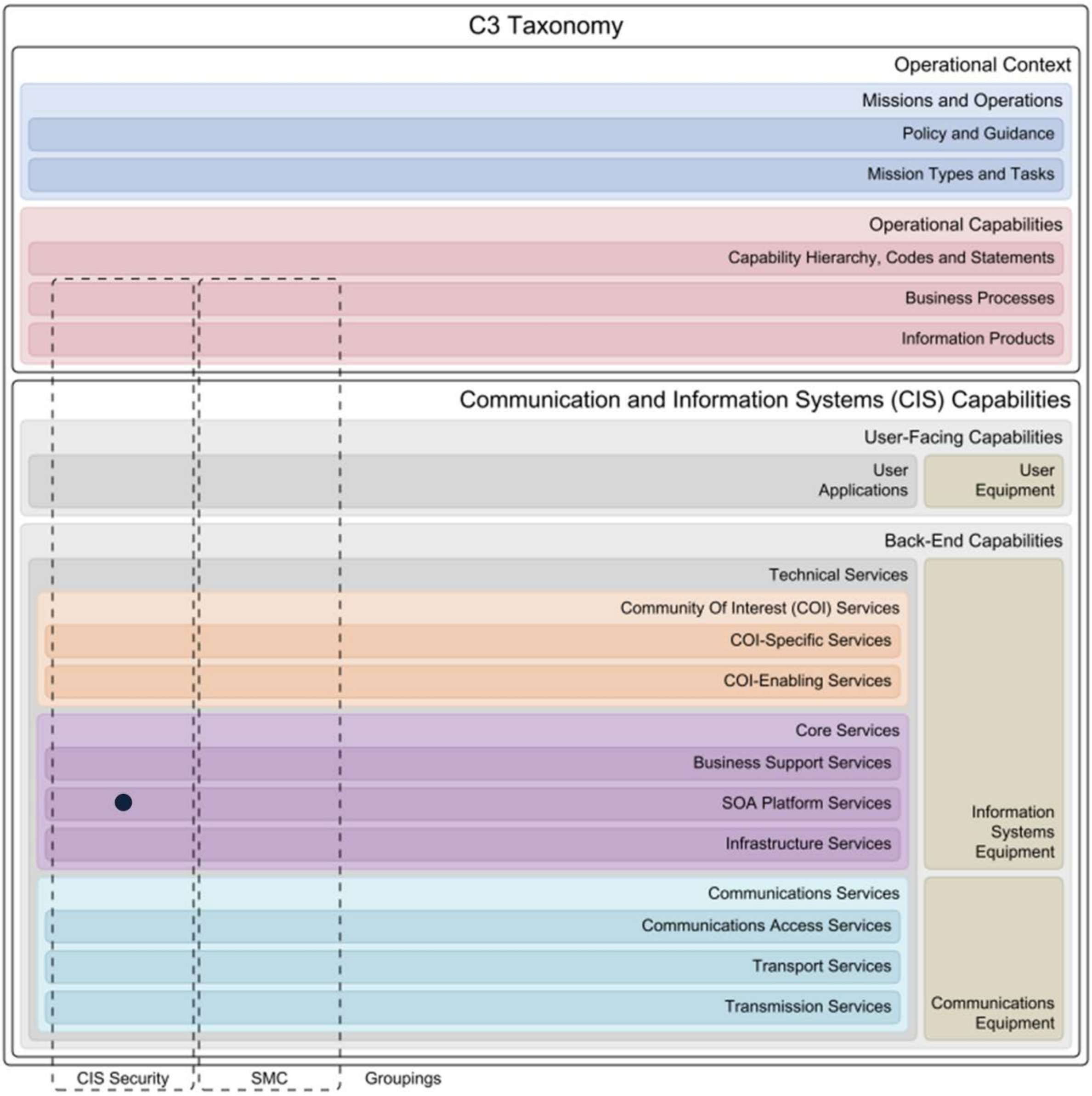


Figure 2 NMBS as Infrastructure Services -> CIS Security Service

2) NMBS as a Distributed service within Core Services.

The Coordinator Service, the Bind Service, the Metadata Service and The Transform Service may themselves be independently identified within the C3 Taxonomy. The Coordinator Service can be viewed as a SOA Platform Service and the Bind Service can be viewed as CIS Security Service given the involvement of cryptographic operations in the generation and verification of strong bindings (see Figure 3 below). The Metadata Service can reside in different locations as they may provide support for COI-specific, COI-enabling and Core Enterprise Service specific types of metadata (see Figure 3). In case the Metadata Service solely provides support for CIS security metadata, it could be identified as a CIS Security Service (see Figure 3 below).

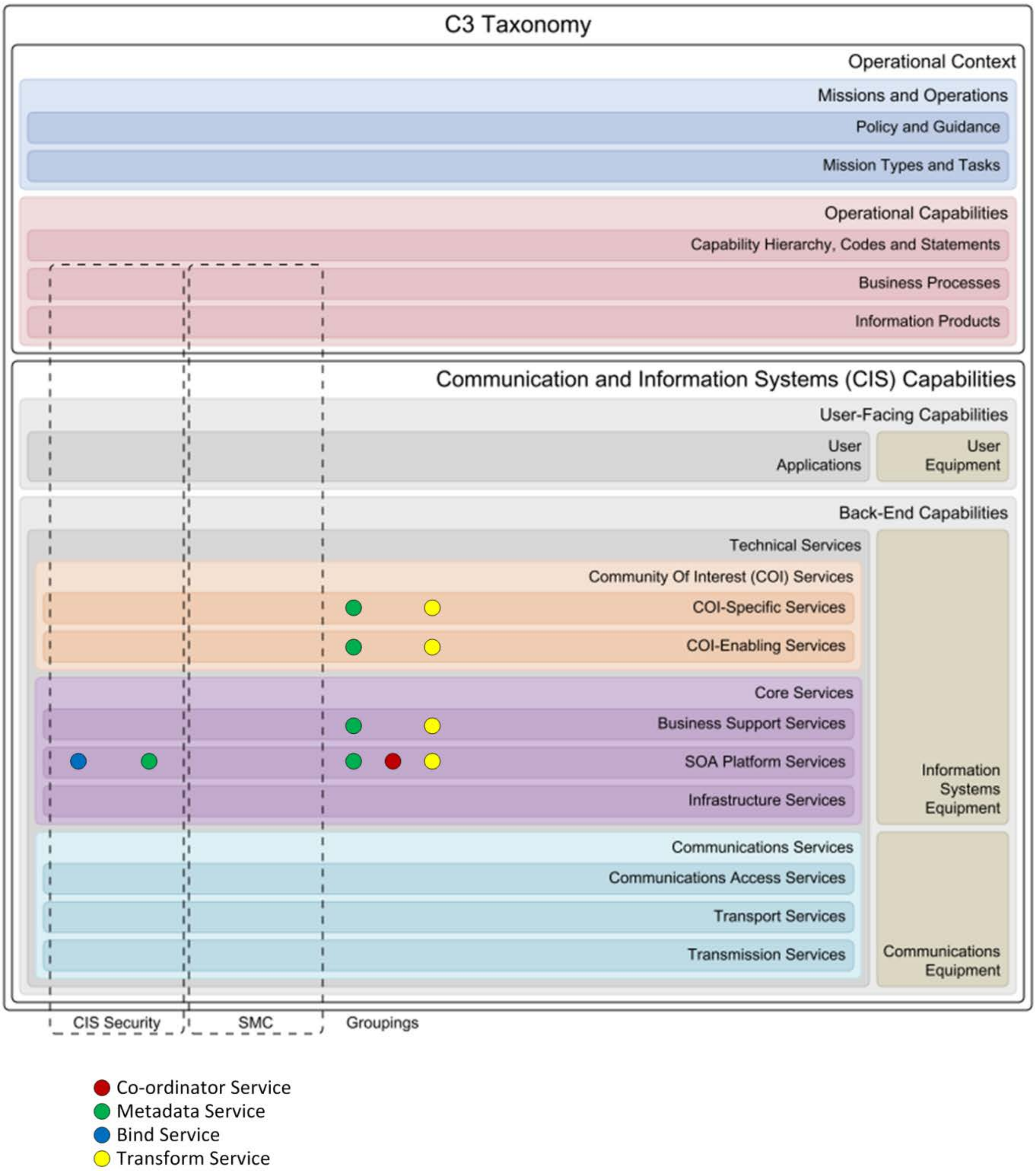


Figure 3 NMBS as a Distributed Service

A.2 Binding Concepts

The binding concepts, approaches, information, management and applications are presented and described in existing specifications for the binding mechanism (Reference [64]) and (Reference [62]). These concepts are used throughout the description of the binding profiles.

A.2.1 Verification and Conformance

A.2.1.1 Approach to Validating Service Interoperability Points

The Table below illustrates the Binding Profiles that were validated at CWIX 2016 along with the test partners involved with the validation.

Binding Profile	Partners	Comments
Embedded Binding in XML Object	USA-TIES CWP@CWIX 2016 USA-USMC CIG@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016	Success Resulted in OCDR: ODCR_Report_00010
Cryptographic Artefact	NATO-Cross-Domain Binding@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016	Success
SMTP	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016 (Infodas Gateway generated STANAG 4778 SMTP Binding)	Success
Office	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016 TUR-VAG-XML-GATEWAY@CWIX 2016 FRA-JACENT@CWIX 2016 NATO-SOA Platform@CWIX 2016	Success
XMPP	NATO-Cross-Domain Binding@CWIX 2016	Success

Binding Profile	Partners	Comments
	NATO-NCIA-JCHAT@CWIX 2016 NATO-SOA Platform@CWIX 2016	Resulted in OCDR: ODCR_Report_000103

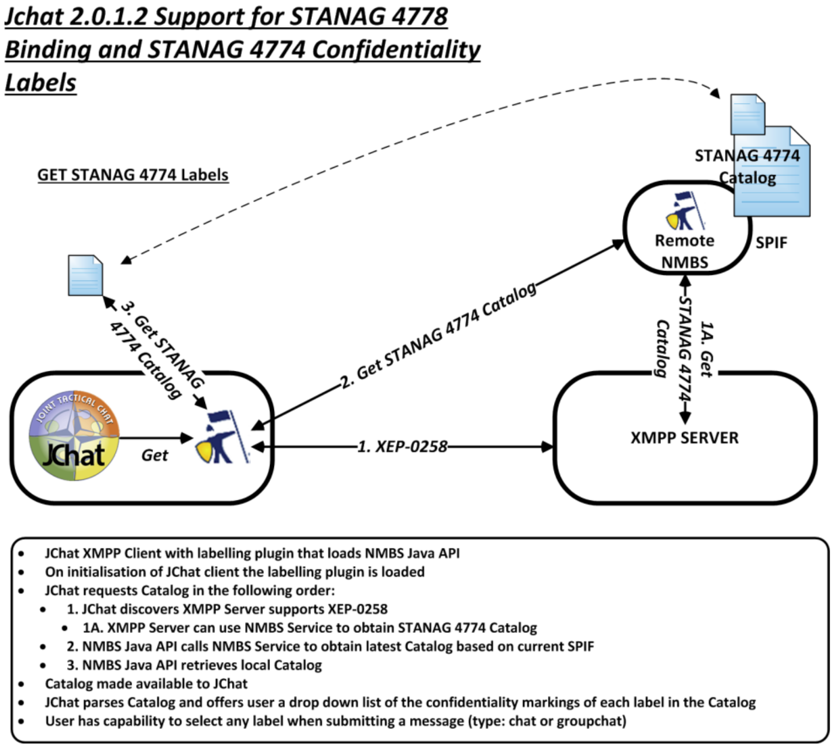
Annex E further specifies the individual test cases that were ran at CWIX 2016 for validating the Binding Profiles specified in this document.

A.2.1.2 Experimentation and Demonstration

A.2.1.2.1 NMBS Support for End Point Labeller (EPLs)

The EPLs are used within user applications, such as JChat and Microsoft Office to apply STANAG 4774 Confidentiality and Alternative Labels to information products. The valid set of Labels are retrieved from the NMBS either as a remote service or an API client plugin (as shown in **Error! Reference source not found.** below), and then applied to the information product using the Binding profiles defined. The EPLs provide a rich set of capability to support: Display of Originator and/or Alternative Confidentiality Labels; Cryptographic Binding; and, Granular labelling for subsets of the information product such as Word paragraphs or PowerPoint slides.

Error! Reference source not found. below illustrates the architecture implemented by JChat to support binding of STANAG 4774 confidentiality labels to XMPP messages.



Jchat 2.0.1.2 Support for STANAG 4778 Binding and STANAG 4774 Confidentiality Labels

- JChat XMPP Client with labelling plugin that loads NMBS Java API
- On initialisation of JChat client the labelling plugin is loaded
- JChat requests Catalog in the following order:
  - 1. JChat discovers XMPP Server supports XEP-0258
    - 1A. XMPP Server can use NMBS Service to obtain STANAG 4774 Catalog
  - 2. NMBS Java API calls NMBS Service to obtain latest Catalog based on current SPIF
  - 3. NMBS Java API retrieves local Catalog
- Catalog made available to JChat
- JChat parses Catalog and offers user a drop down list of the confidentiality markings of each label in the Catalog
- User has capability to select any label when submitting a message (type: chat or groupchat)

Figure 4: NMBS to support EPLs, such as JChat

A.2.1.2.2 NMBS Support for Cross Domain (or cross COI) Information Exchange

When information is produced within a Community of Interest (COI), national domain or organizational domain (such as NATO) that information is initially shared within that domain. As such, labelling solutions are often developed for that specific domain. Users, applications, services and systems within that domain all support and understand the specific labelling method. When information is to be shared between different COIs or domains then the domain-specific labelling method may not be understood or supported outside of that domain. To facilitate information sharing a common interpretation of the labels and binding (i.e. 'where can the labels be located in the information', 'what does the label mean' and 'to which data does the label pertain') is required.

Friendly Force Tracking (FFT) is the capability to provide positional information from the lowest possible level, aimed to enhance situational awareness, combat effectiveness and to avoid fratricide by providing the location of friendly forces. This information is introduced to the tactical and operational pictures via a coordinated flow of information by means of available communication circuits. Tracked objects are typically ground vehicles or dismounted soldiers of national, NATO or coalition owned forces. At CWIX 2016 FFT established a second security domain to simulate the exchange of track information through a guard. Figure 5 below illustrates the architecture that was implemented.

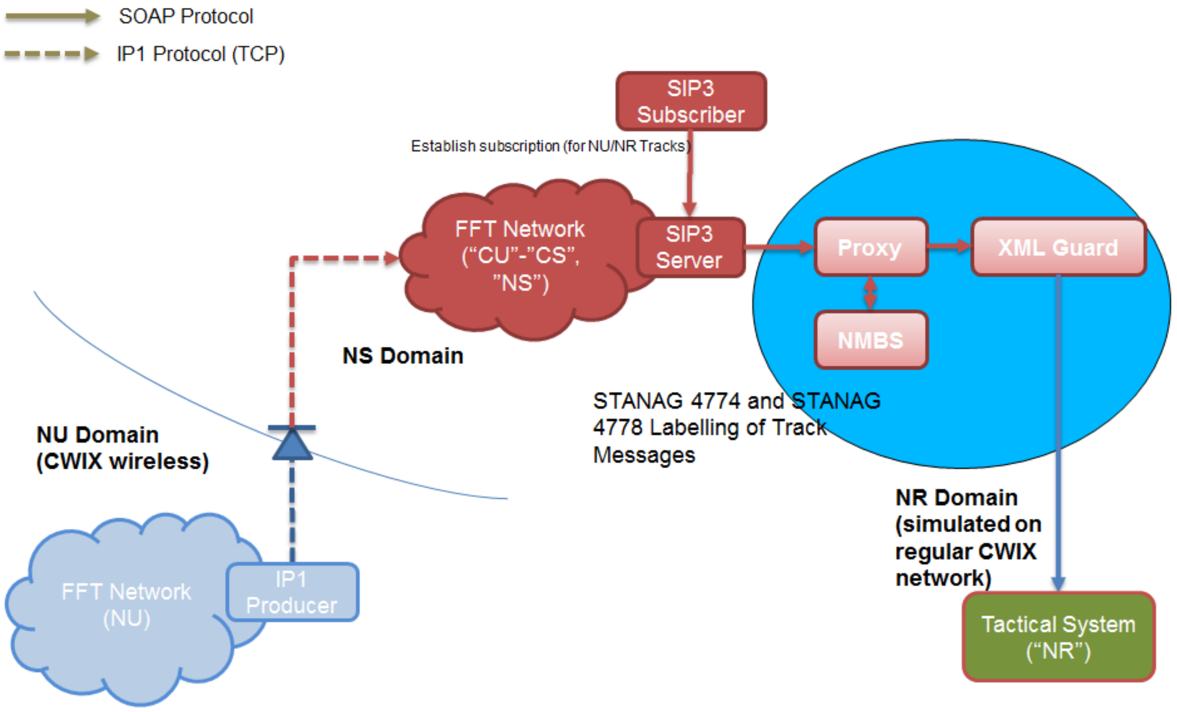

SOAP Protocol
IP1 Protocol (TCP)
SIP3 Subscriber
Establish subscription (for NU/NR Tracks)
FFT Network ("CU"-"CS", "NS")
SIP3 Server
Proxy
XML Guard
NMBS
NS Domain
STANAG 4774 and STANAG 4778 Labelling of Track Messages
NU Domain (CWIX wireless)
NR Domain (simulated on regular CWIX network)
FFT Network (NU)
IP1 Producer
Tactical System ("NR")

Figure 5 NMBS to support COI-Specific and Cross Domain Information Exchange Labelling

The NMBS Proxy acts as a transparent service for both provider and consumer that is responsible for invoking the NMBS and forwarding compliant tracks to the XML Guard. The NMBS validates the COI-specific APP11-D MTF labels, and transforms the COI-specific confidentiality labels to STANAG 4774 confidentiality labels. Those confidentiality labels are then bound to the tracks based on the SOAP binding profile and the cryptographic binding profile described in this document. The XML Guard is responsible for making sure that each STANAG 4774 confidentiality label is evaluated against the release policy for the interconnection and that the cryptographic binding is digitally verified.

In this architecture the XML Guard is only required to support the STANAG 4774 Confidentiality Label syntax and the Binding Profiles. It is the responsibility of the NMBS to perform the mappings between COI-specific labels and STANAG 4774 confidentiality labels.

A.2.2 Configuration Management and Governance

The development of the confidentiality labels and binding profiles was based on the use of XML being recognized as the 'lingua franca' for information exchange. XML offers a varied and readily available array of XML tools. The NMBS was designed on this principle and as such extensively uses XML-based standards for supporting its operations. For example, XML stylesheets (XSLTs) are used for mapping between different metadata types and formats and validating metadata. The NMBS is able to support dynamic updates to policies and associated artefacts represented in XML, such as Security Policy Information Files (SPIFs) representing organizational security labelling policies, with its loosely-coupled integration with the NATO Metadata Registry and Repository (NMRR).

Annex B : Metadata Binding Mechanism

B.1 Introduction

This annex defines the syntax and semantics of a Metadata Binding Mechanism to support the conduct of a mission through efficient and effective information management, enabling decision-making by the sharing of information within and between NATO elements, the nations and their respective Communities of Interest.

This appendix incorporates support for the binding of sensitivity metadata, encoded in the format and syntax of the Confidentiality Metadata Label (Reference [62]and reference [63]) to data objects.

This appendix is derived from "NATO Profile for the 'Binding of Metadata to Data Objects', Version 1.1", (Reference [11]), which is a profile of the general binding specification outlined in (Reference [10]).

B.2 Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].

- Words in italics indicate terms derived from Section B.3.

- Courier font indicates syntax derived from various W3C XML Signature (Reference [15]) and XPATH 1.0 (Reference [42]) standards referenced in this Appendix.

B.3 Metadata Binding Syntax

An XML schema is defined which contains the elements and attributes of the Metadata Binding.

This section specifies the different elements and attributes of the Metadata Binding.

B.3.1 Namespaces and Constraints

The table below summarizes the XML namespaces and corresponding prefixes used throughout for the Metadata Binding.

Table 1: XML Namespaces for Metadata Binding

Attribute	Namespace
mb	urn:nato:stanag:4778:bindinginformation:1:0
ds	http://www.w3.org/2000/09/xmldsig#
xmime	http://www.w3.org/2005/05/xmlmime
xs	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

B.3.2 The MetadataBindingContainer Element

The *MetadataBindingContainer* is the top-level element of the Metadata Binding and SHALL be present.

The *MetadataBindingContainer* element contains one or more *MetadataBinding* element(s).

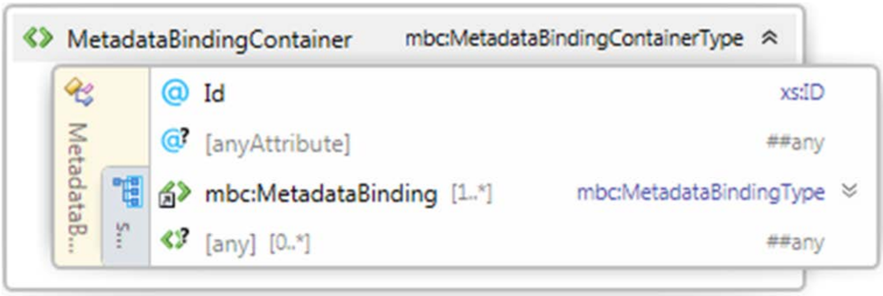


Figure 6: The MetadataBindingContainer Element

The table below specifies the use of attributes of the *MetadataBindingContainer* element with example values.

Table 2: Attributes of MetadataBindingContainer

Attribute	Mandatory/Optional/Prohibited	Notes	Example values
Id	Optional	Unique identifier of this element instance	"mb-bindingId-006"

The *MetadataBindingContainer* can be extended with additional elements and attributes to support Community of Interest or National binding requirements.

These additional elements and attributes are for local use and MAY be ignored by other systems.

B.3.3 The MetadataBinding Element

The MetadataBinding element:

- SHALL contain at least one or more metadata components (a choice of *Metadata* or *MetadataReference* elements);
- SHALL contain at least one data object component (a choice of *Data* or *DataReference* elements).

All of the metadata (*Metadata* or *MetadataReference* elements) is bound to all of the data objects (*Data* or *DataReference* elements) within the *MetadataBinding* element.

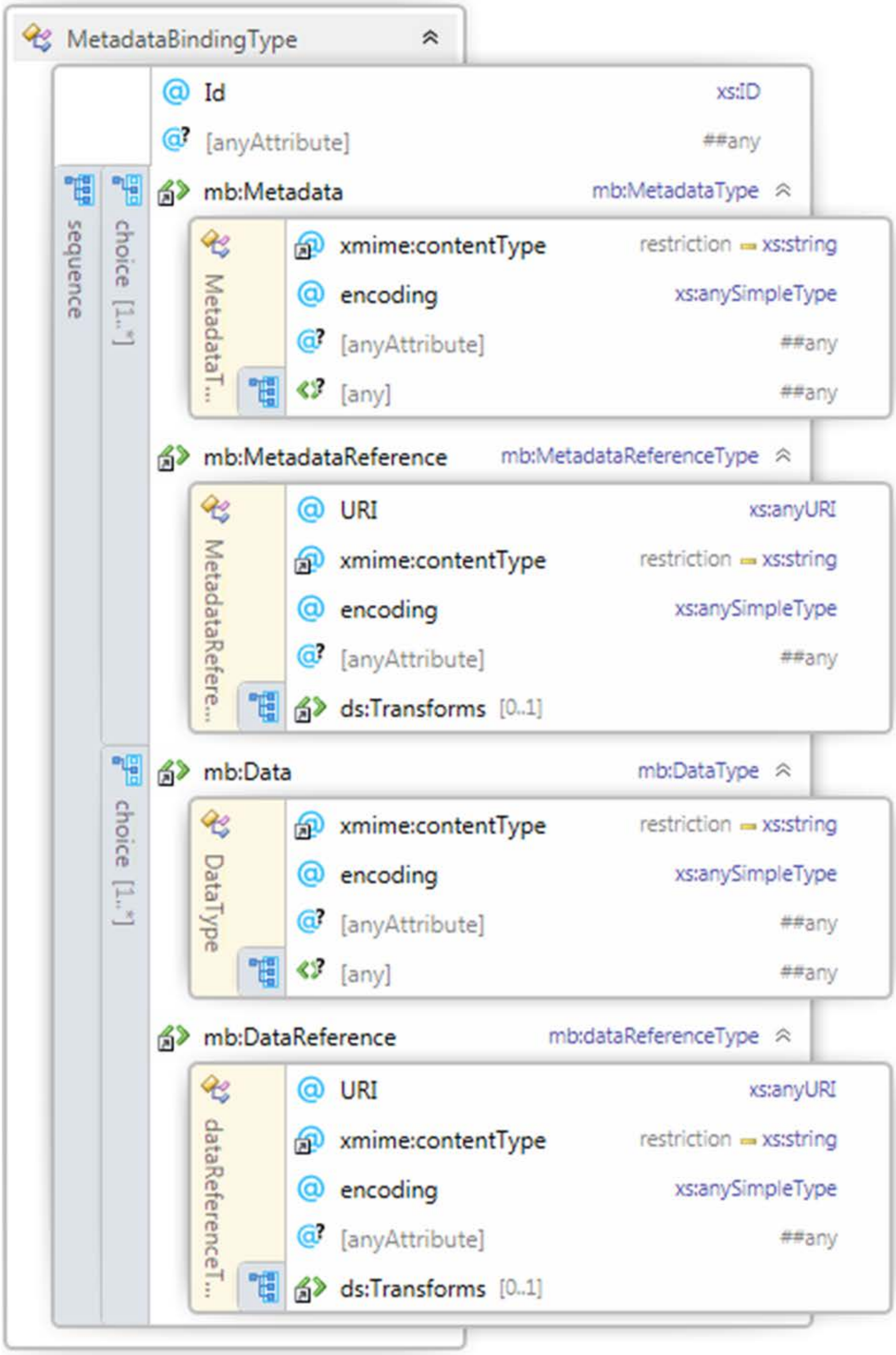


Figure 7: The MetadataBinding Element

The table below specifies the use of the child elements of the *MetadataBinding* element.

Table 3: Child Elements of the MetadataBinding Element

Component	Mandatory/ Optional/ Prohibited	Element Choices	Notes
Metadata	Mandatory	Metadata	Element used to embed metadata within the *MetadataBinding* element. There may be one or more *Metadata* elements.
		MetadataReference	Element used for referencing detached metadata. There may be one or more *MetadataReference* elements.
Data	Mandatory	Data	Element used to embed a data object that is bound to the metadata (embedded or detached) within the *MetadataBinding* element. There may be one or more *Data* elements.
		DataReference	Element used to reference detached data that is bound to the metadata (embedded or detached) within the *MetadataBinding* element. There may be one or more *DataReference* elements.

The table below specifies the use of attributes of the *MetadataBindingContainer* element with example values.

Table 4: Attributes of MetadataBindingContainer

Attribute	Mandatory/Optional/Prohibited	Notes	Example values
Id	Optional	Unique identifier of this element instance	"mb-metadatabindingId-007"

The *MetadataBinding* can be extended with additional elements and attributes to support Community of Interest or National binding requirements.

These additional elements and attributes are for local use and MAY be ignored by other systems.

B.3.4 The Metadata Element

In the case that metadata is embedded within the Metadata Binding; one or more *Metadata* elements SHALL be present in a *MetadataBinding* element.

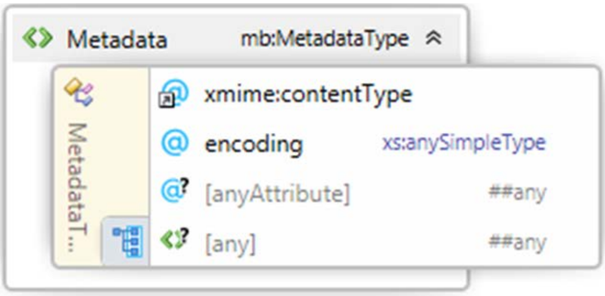


Figure 8: The Metadata Element

Table 5 below describes the use of the attributes of the *Metadata* element.

Table 5: Attributes of the Metadata Element

Attribute	Mandatory/ Optional/ Prohibited	Notes	Example values	Default Value
xmime:contentType	Optional	Attribute imported from (Reference [16]). This attribute is used to signify the content type (also known as media type or MIME type). Attribute values from (Reference [20])	"image/jpeg", "text/xml; charset=utf-16"	"text/xml; charset=utf-8"
encoding	Optional	Imported from (Reference [17]). This attribute is only used to signify how the non-XML data object is encoded.	"base64Binary" or "hexBinary"	N/A

If the *MetaData* element does not contain a *contentType* attribute value the Default Value will be assumed.

The *contentType* attribute MAY be extended to support types that are not IANA registered MIME types. Support for non-registered MIME types are for local use and MAY be ignored by other systems.

For non-XML metadata, it may be necessary for the metadata to undergo an encoding transformation in order to be embedded in XML. In the case where metadata is encoded and is contained in the *MetaData* element the *encoding* attribute SHALL be present with a value indicating the type of encoding.

The *MetaData* element can be extended with additional elements and attributes to support Community of Interest or National binding requirements. These additional elements and attributes are for local use and MAY be ignored by other systems.

B.3.5 The MetadataReference Element

In the case that metadata is detached from the Metadata Binding; one or more *MetadataReference* elements SHALL be present in a *MetadataBinding* element.

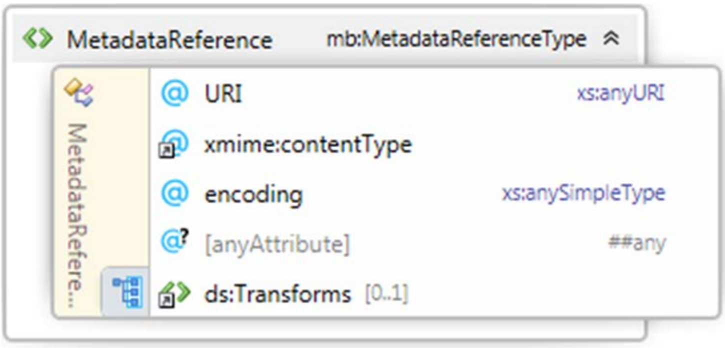


Figure 9: The MetadataReference Element

The table below describes the use of the attributes of the *MetadataReference* element.

Table 6: Attributes of the MetadataReference Element

Attribute/ Element	Mandatory/ Optional/ Prohibited	Notes	Example values
URI	Mandatory	A URI reference conformant with (Reference [21]) that indicates the location of the metadata.	http://smhs.co.uk/policy/smhs/red http://www.someserver.net/example.json#/emailAddr
xmime:contentType	Optional	Attribute imported from (Reference [16]). This attribute is used to signify the content type (also known as media type or MIME type) of the metadata. Attribute values from (Reference [20])	"image/jpeg", "text/xml; charset=utf-16"
encoding	Optional	Attribute imported from (Reference [19]). This attribute is used to signify	"binary"

Attribute/ Element	Mandatory/ Optional/ Prohibited	Notes	Example values
		how non-XML metadata is encoded.	
ds:Transforms	Optional	Element imported from (Reference [15]). Supports the inclusion of an XPath expression for specifying the referenced subset (portion) of an XML data object. There may be zero or one `ds:Transforms` elements.	"<Transforms xmlns="http://www.w3.org/2000/09/xmldsig#"> <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116"> <XPath>ancestor-or-self::*[local-name()='DocumentRoot' and namespace-uri()='http://example.com/</XPath> </Transform> </Transforms>"

In the case where the *contentType* attribute is present it SHOULD contain a value from the IANA registered list at (Reference [20]).

The *contentType* attribute MAY be extended to support types that are not IANA registered MIME types. Support for non-registered MIME types are for local use and MAY be ignored by other systems.

In the case where the *encoding* attribute is present it SHALL contain a value from the IANA registered list at (Reference [19]).

If the *MetadataReference* element does not contain a *contentType* attribute value the default value "*text/xml; charset=utf-8*" will be assumed.

The URI generic syntax is specified in (Reference [21]). The URI attribute is mandatory in order to locate the data object and subsets of the data object.

If the metadata is XML and only a subset (portions) of the XML is to be referenced as metadata, XPath expressions can be used to achieve this. XPath expressions are permitted in the `Transforms` element as specified in (Reference [15]). The *Transforms* reference type of the *MetadataReference* element is based on the `Transforms` element as specified in (Reference [15]).

One and only one *Transform* element (child element of the *Transforms* element) SHALL be present. The input for the *Transform* element is the dereferenced URI attribute value of the *MetadataReference* element.

The *Transform* element SHALL have an *Algorithm* attribute with the value http://www.w3.org/TR/1999/REC-xpath-19991116.

The *Transform* element SHALL contain one or more child *XPath* elements.

The XPath expression contained within a child *XPath* element of the *Transform* element SHALL be compliant with XPATH 1.0 (Reference [42]).

The evaluation of the XPath expression contained within a child *XPath* element of the *Transform* element SHALL be compliant with XML Signature XPath Filtering (XMLDSIG, Reference [15]).

XML (de)serialization may result in a namespace prefix to be redefined within the XML document. As a result XPATH transformations may become invalid. In order to avoid invalid XPATH transformations as a result of redefined namespace prefixes it is RECOMMENDED that the `local-name` and `namespace-uri` functions (as described in Reference [42]) are used.

The *MetadataReference* can be extended with additional elements and attributes to support Community of Interest or National binding requirements. These additional attributes are for local use and MAY be ignored by other systems.

B.3.6 The *Data* Element

In the case that data is embedded within the Metadata Binding; one or more *Data* elements SHALL be present in a *MetadataBinding* element.

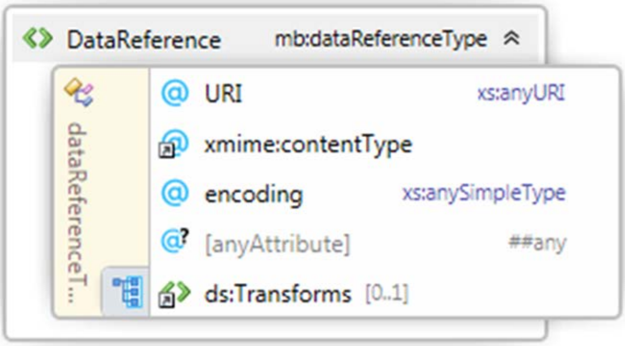


Figure 10: The Data Element

The table below describes the use of the attributes of the *Data* element, together with example values.

Table 7: Attributes of the Data Element

Attribute	Mandatory/ Optional/ Prohibited	Notes	Example values	Default Value
xmime:contentType	Optional	Attribute imported from (Reference [16]). This attribute is used to signify the content type (also known as media type or MIME type). Attribute values from (Reference [20])	“image/jpeg”, “text/xml; charset=utf-16”	“text/xml; charset=utf-8”
encoding	Optional	Imported from (Reference [17]). This attribute is only used to signify how the non-XML data object is encoded.	“base64Binary” or “hexBinary”	N/A

If the *Data* element does not contain a *contentType* attribute value the Default Value will be assumed.

The *contentType* attribute MAY be extended to support types that are not IANA registered MIME types. Support for non-registered MIME types are for local use and MAY be ignored by other systems.

For non-XML data objects, it may be necessary for the data object to undergo an encoding transformation in order to be embedded in XML. In the case where a data object is encoded and is contained in the child element of the *Data* element the *encoding* attribute SHALL be present with a value indicating the type of encoding.

The *Data* element can be extended with additional elements and attributes to support Community of Interest or National binding requirements. These additional attributes are for local use and MAY be ignored by other systems.

B.3.7 The DataReference Element

In the case that data is detached from the Metadata Binding; one or more *DataReference* elements SHALL be present in a *MetadataBinding* element.

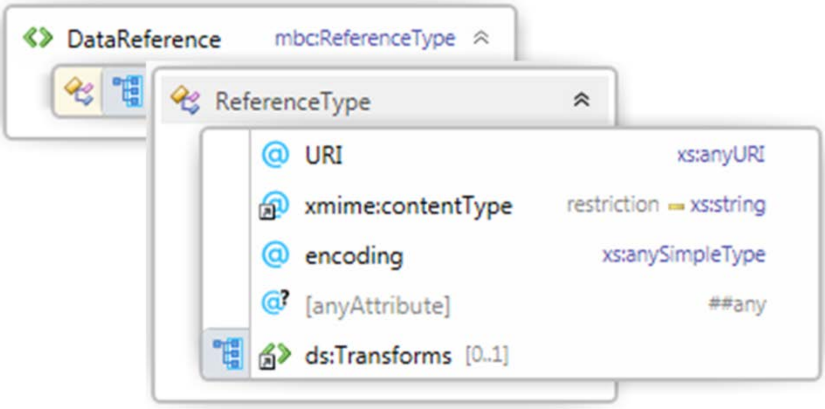


Figure 11: The DataReference Element

Table 8 describes the use of the attributes of the *DataReference* element.

Table 8: Attributes of the DataReference Element

Attribute/Element	Mandatory/ Optional/ Prohibited	Notes	Example values
xmime:contentType	Optional	Attribute imported from (Reference [16]). This attribute is used to signify the content type (also known as media type or MIME type). Attribute values from (Reference [20])	“image/jpeg”, “text/xml; charset=utf-16”
Encoding	Optional	Attribute imported from (Reference [19]). This attribute is used to signify how non-XML data object is encoded.	“8bit”, “binary”
URI	Mandatory	A URI reference conformant with (Reference [21]) that indicates the location of the data object.	“http://www.someserver.net/example.doc”
ds:Transforms	Optional	Element imported from (Reference [15]). Supports the inclusion of an XPath expression for specifying the referenced subset (portion) of an XML data object.	“<Transforms xmlns="http://www.w3.org/2000/09/xmldsig#" > <Transform Algorithm="http://ww

Attribute/Element	Mandatory/ Optional/ Prohibited	Notes	Example values
		There may be zero or one `ds:Transforms` elements.	w.w3.org/TR/1999/REC-xpath-19991116"> <XPath>ancestor-or-self::*[local-name()='DocumentRoot' and namespace-uri()='http://example.com/</XPath> </Transform> </Transforms>"

In the case where the *contentType* attribute is present it SHOULD contain a value from the IANA registered list at (Reference [20]).

The *contentType* attribute MAY be extended to support types that are not IANA registered MIME types. Support for non-registered MIME types are for local use and MAY be ignored by other systems.

In the case where the *encoding* attribute is present it SHALL contain a value from the IANA registered list at (Reference [19]).

If the *DataReference* element does not contain a *contentType* attribute value the default value "*text/xml; charset=utf-8*" will be assumed.

The URI generic syntax is specified in (Reference [21]). The URI attribute is mandatory in order to locate the data object and subsets of the data object.

If the data object is XML and subsets of the data object need to be referenced, the use of the *Transforms* element as a child element of the *DataReference* element is permitted.

The use of the *Transforms* element for referencing subsets of a data object is the same as specified for the *MetadataReference* element.

The *DataReference* can be extended with additional elements and attributes to support Community of Interest or National binding requirements. These additional elements are for local use and MAY be ignored by other systems.

B.3.8 Schema

The schema for the Metadata Binding is shown in Appendix B-1: Metadata Binding Syntax.

B.3.8.1Binding Information Syntax

The Binding Information is represented as an XML structure that includes the Metadata Binding (the *MetadataBindingContainer* element), or the compound of the Metadata Binding and the Cryptographic Artefact. The Binding Information is stored as a Binding Data Object (BDO)[1].

This section specifies the syntax of the Binding Information.

This section is derived from the Binding of Metadata to Data Objects outlined in (Reference [11]).

Where discrepancies arise between this section and (Reference [11]), this section is definitive.

This section specifies the different elements for the Binding Information element.

B.3.8.1.1 The BindingInformation Element

When creating a BDO the standard name *BindingInformation* SHALL be used as the parent element that holds the Metadata Binding and the Cryptographic Artefact (if present).

The *BindingInformation* element SHALL be qualified with the 'urn:nato:stanag:4778:bindinginformation:1:0' namespace.

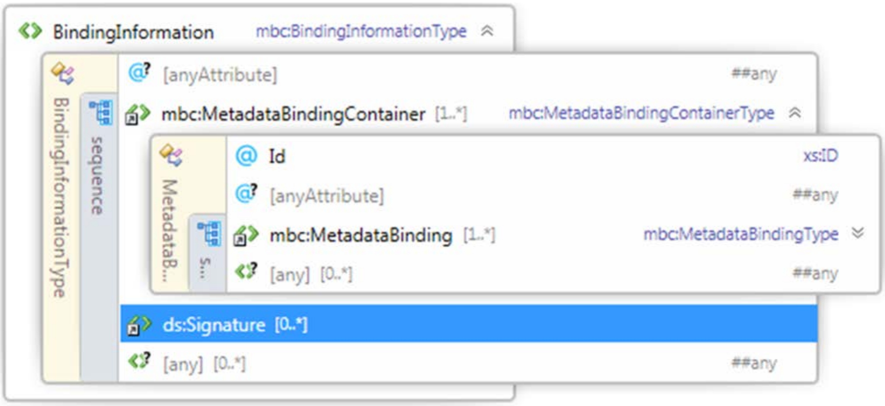


Figure 12: The BindingInformation Element

Table 9 describes the use of the elements of the *BindingInformation* element.

Table 9: Elements of the BindingInformation Element

Element	Mandatory/Optional/ Prohibited	Notes
mbc:MetadataBindingContainer	Mandatory	This element represents the Metadata Binding. There may be one or more *MetadataBindingContainer* elements in a *BindingInformation* element.

[1] The Binding Data Object (BDO) may be stored separately (as a data object stored locally or as a data object stored in a repository) or may be embedded within a data object.

Element	Mandatory/Optional/ Prohibited	Notes
		There may be one or more *mbc:MetadataBindingContainer* elements
ds:Signature	Optional	This element represents the digital signature of the Metadata Binding. There may be zero or more *Signature* elements in a *BindingInformation* element (Reference 15). There may be zero or more `ds:Signature` elements

The *BindingInformation* can be extended with additional elements and attributes to support Community of Interest or National binding requirements.

These additional elements and attributes are for local use and MAY be ignored by other systems.

B.3.8.2Binding Information Management

To ensure that information (data of a finite size) is handled effectively, efficiently and securely the BDO SHALL be accessible to information management services (Reference [2]) that are applying information management policies. The information management services SHALL be able to uniquely associate the BDO with a data object, i.e. for a given data object information management services SHALL be able to locate the associated BDO.

In order to facilitate simplified processing for information management services it is RECOMMENDED that metadata is embedded within a BDO and not referenced. Hence, the metadata is contained in the *Metadata* element of the Metadata Binding.

A BDO can be categorised as follows as defined in (Reference [64]:

- Encapsulating – whereby the data object is embedded within the BDO;
- Embedded – whereby the BDO is embedded in a XML data object and the data object is identified via a reference ; and
- Detached – whereby the data object is external to the BDO and is identified via a reference.

B.3.8.2.1 Encapsulating Binding Information

An Encapsulating BDO is an XML document that SHALL have the *BindingInformation* element as the root element of that XML document (as specified in Section B.3.8.1).

It is RECOMMENDED that an Encapsulating BDO only contain *Data* element(s) within a Metadata Binding.

An Encapsulating BDO SHALL be capable of supporting any type of data object in a *Data* element within a Metadata Binding.

B.3.8.2.2 Embedded Binding Information

If a data object is XML and the schema definition for that XML document supports extensibility then the BDO can be embedded as a sibling element of that data object (or other sibling elements of the parent element). In this case, the root element of the BDO SHALL be the *BindingInformation* element (as specified in Section B.3.8.1).

A non-XML data object may be capable of supporting an Embedded BDO as a sibling node of that data object (or other sibling nodes of the parent node). In this case, the root element of the BDO SHALL be the *BindingInformation* element (used to hold the Metadata Binding and the Cryptographic Artefact (if present) as children). In this case it MAY be required to encode the Embedded BDO in a format supported by the media type of the parent data object.

It is RECOMMENDED that an Embedded BDO only contain *DataReference* element(s) within a Metadata Binding.

It is RECOMMENDED that the *DataReference* element(s) contained within a Metadata Binding are not references external to the data object that the BDO is a child element of.

B.3.8.2.3 Detached Binding Information

A Detached BDO is external to the data object(s) that is (are) being referenced. In this case, the root element of the BDO SHALL be the *BindingInformation* element ((as specified in Section B.3.8.1).

A Detached BDO can be a standalone XML document with external references to physically separated data objects.

A Detached BDO can be a child node of a parent node, whereby the BDO contains references to other child nodes of that parent node. In this case the BDO is external to the child nodes (of the parent node) that it references.

It is RECOMMENDED that a Detached BDO only contain *DataReference* element(s) within a Metadata Binding.

It is RECOMMENDED that the *DataReference* element(s) contained within a Metadata Binding are external to the BDO.

B.3.8.2.4 Same-Document References

In the case of Embedded and Detached BDOs *DataReference* elements can be used to refer to Same-Document references. In other words, the reference that is to be dereferenced based on the *DataReference URI* attribute value is contained within the same data object as the BDO.

In all cases dereferencing a null *DataReference URI* attribute value (*URI=""*) MUST return the root node of the document (data object).

In all cases dereferencing a Base Uri (refer to Reference [21] Section 5.1.3) *DataReference URI* attribute value (for example; *URI="http://www.example.com/example.json"*) MUST return the root node of the document (data object).

A component of the URI (see Reference [21]) is the fragment identifier that is used to build a URI reference. A URI reference is a powerful concept that allows indirect identification of a secondary resource (within the Same-Document) by reference to a primary resource. The fragment identifier is indicated by the presence of a number sign ("#") character and terminated by the end of the URI (for example; *URI="#foo"*;or, *URI="http://www.example.com/example.json#foo"*).

The significance of the fragment identifier is a function of the content type (also known as media type or MIME type). In other words, unless the content type is known the syntax and the semantics for interpreting the fragment identifier are unknown. Content types are registered on the internet and the registered list of content types is maintained at the Internet Assigned Numbers Authority IANA (Reference [20]). Content types that are registered with IANA also specify how applications must interpret fragment identifiers. As such, the interpretation of the fragment identifier is dependent upon the content type of the data object.

In all cases dereferencing a *DataReference URI* attribute value that contains a fragment URI, the characters and interpretation of those characters after the number sign ('#') character MUST conform to the syntax and semantics of fragment identifiers specified by the content type identified by the *xmime:contentType* attribute value.

Appendix B-1: Metadata Binding Syntax

This appendix provides the complete syntax for the Metadata Binding Mechanism in XML format.

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

<!--
*********************************************************************

                   NATO UNCLASSIFIED

XML Schema for capturing the Metadata Binding specification for
binding metadata to data objects.

           I
          / \
        -< + >-
          \ /
           I                NCI AGENCY
   ##   #    ####    #      P.O. box 174
   ##   #   #    #   #      2501 CD The Hague
   # #  #   #        #
   #  # #   #    #   #      Core Enterprise Services
   #   ##    ####    #
     A  G  E  N  C  Y

*********************************************************************
-->

<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-Instance"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
  targetNamespace="urn:nato:stanag:4778:bindinginformation:1:0"
  version="1.4"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:appinfo>
      <UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0</UniqueIdentifier>
      <Name>Metadata Binding Schema</Name>
      <Definition>Schema for binding metadata to data objects</Definition>
      <VersionIndicator>1.4</VersionIndicator>
      <UsageGuidance>Used within NATO to bind metadata to data objects, including the NATO
Core Metadata.</UsageGuidance>
      <RestrictionType/>
      <RestrictionValue/>
      <ConfidentialityLabel ReviewDateTime="2021-04-22T09:00:00Z">
        <ConfidentialityInformation>
          <PolicyIdentifier>NATO</PolicyIdentifier>
          <Classification>UNCLASSIFIED</Classification>
          <Category Type="PERMISSIVE" TagName="Context">
            <GenericValue>NATO-7</GenericValue>
          </Category>
        </ConfidentialityInformation>
        <CreationDateTime>2016-04-22T09:00:00Z</CreationDateTime>
      </ConfidentialityLabel>
    </xs:appinfo>
    <xs:documentation>
      The schema can be used with the confidentiality label schema to bind confidentiality
label metadata (such as those defined in the NATO Core Metadata Specification NCMS)) to
data objects.
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/2000/09/xmldsig#"/>
```

```
  <xs:import namespace="http://www.w3.org/2005/05/xmlmime"
schemaLocation="http://www.w3.org/2005/05/xmlmime"/>

  <xs:element name="BindingInformation" type="mb:BindingInformationType"/>
  <xs:complexType name="BindingInformationType" id="bindingInformationType">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:bindingInformationTy
pe</UniqueIdentifier>
        <Name>Binding Information Type</Name>
        <Definition></Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance>Used to bind arbitary metadata to data objects</UsageGuidance>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ds:Signature" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="mb:MetadataBindingContainer" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

  <xs:element name="MetadataBindingContainer" type="mb:MetadataBindingContainerType"/>
  <xs:complexType name="MetadataBindingContainerType" id="metadataBindingContainerType">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:metadataBindingConta
inerType</UniqueIdentifier>
        <Name>Metadata Binding Container Type</Name>
        <Definition>A sequence of Metadata Bindings</Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="mb:MetadataBinding" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:ID"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

  <xs:element name="MetadataBinding" type="mb:MetadataBindingType"/>
  <xs:complexType name="MetadataBindingType" id="metadataBindingType">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:metadataBindingType<
/UniqueIdentifier>
        <Name>Metadata Binding Type</Name>
        <Definition>A binding between metadata and data objects.</Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
```

```
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="mb:Metadata"/>
        <xs:element ref="mb:MetadataReference"/>
      </xs:choice>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="mb:Data"/>
        <xs:element ref="mb:DataReference"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:ID"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

  <xs:element name="Metadata" type="mb:MetadataType"/>
  <xs:complexType name="MetadataType" id="metadataType" mixed="true">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:metadataType</Unique
Identifier>
        <Name>Metadata Type</Name>
        <Definition>In-line metadata.</Definition>
        <VersionIndicator>1.3</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax"/>
    </xs:sequence>
   <xs:attribute ref="xmime:contentType"/>
   <xs:attribute name="encoding"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

  <xs:element name="MetadataReference" type="mb:MetadataReferenceType"/>
  <xs:complexType name="MetadataReferenceType" id="metadataReferenceType">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:metadataReferenceTyp
e</UniqueIdentifier>
        <Name>Metadata Reference Type</Name>
        <Definition>A reference to a piece of metadata.</Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="mb:ReferenceType">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="Data" type="mb:DataType"/>
  <xs:complexType name="DataType" id="dataType" mixed="true">
       <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:dataType</UniqueIden
tifier>
        <Name>Data Type</Name>
        <Definition>In-line data object.</Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax"/>
    </xs:sequence>
    <xs:attribute ref="xmime:contentType"/>
    <xs:attribute name="encoding"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

  <xs:element name="DataReference" type="mb:dataReferenceType"/>
 <xs:complexType name="dataReferenceType" id="dataReferenceType">
  <xs:annotation>
   <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:dataReferenceType</U
niqueIdentifier>
    <Name>Data Reference Type</Name>
    <Definition>A reference to a data object.</Definition>
    <VersionIndicator>1.1</VersionIndicator>
    <UsageGuidance></UsageGuidance>
    <RestrictionType></RestrictionType>
    <RestrictionValue></RestrictionValue>
   </xs:appinfo>
   <xs:documentation>

   </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
   <xs:extension base="mb:ReferenceType">
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>

  <xs:complexType name="ReferenceType" id="referenceType">
    <xs:annotation>
      <xs:appinfo>

<UniqueIdentifier>urn:nato:stanag:4778:bindinginformation:1:0:appinfo:referenceType</Uniqu
eIdentifier>
        <Name>Reference Type</Name>
        <Definition>A reference to a data object, or part of a data object.</Definition>
        <VersionIndicator>1.2</VersionIndicator>
        <UsageGuidance></UsageGuidance>
        <RestrictionType></RestrictionType>
        <RestrictionValue></RestrictionValue>
      </xs:appinfo>
      <xs:documentation>

      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ds:Transforms" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="URI" type="xs:anyURI" use="required"/>
    <xs:attribute ref="xmime:contentType"/>
    <xs:attribute name="encoding"/>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>

</xs:schema>
```

Appendix B-2: Example Bindings

This section contains fictitious examples that illustrate the different categorisations of Binding Information as Binding Data Objects (BDOs). Also included are examples of BDOs that illustrate the flexibility of the Metadata Binding Mechanism to support one-to-one, one-to-many, many-to-one and many-to many relationships between metadata and data objects. All examples given in this Appendix use Confidentiality Metadata Labels (Reference [9]) as example metadata.

Encapsulating BDO

The following example depicts the encapsulation of an XML data object within the *Data* element of a BDO.

```
    <mb:BindingInformation
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
          <mb:Metadata>
           <slab:originatorConfidentialityLabel
            xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
             <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
             <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
             2015-09-30T12:30:00Z
            </slab:CreationDateTime>
           </slab:originatorConfidentialityLabel>
          </mb:Metadata>
          <mb:Data>
            <Document xmlns="http://example.com/doc">
              <Title>BDO Examples</Title>
              <Author>alan.ross@reach.nato.int</Author>
              <Abstract>
               Example XML File to support illustration of different types of BDO
              </Abstract>
              <Introduction>....</Introduction>
              <Chapter Id="chapter-1">
                <Paragraph Id="para-1-1" />
                <Paragraph Id="para-1-2" />
              </Chapter>
              <Chapter Id="chapter-2">
                <Paragraph Id="para-2-1" />
                <Paragraph Id="para-2-2" />
              </Chapter>
            </Document>
          </mb:Data>
        </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
    </mb:BindingInformation>
```

This example shows a non-XML data object (image) embedded within the *Data* element of a BDO.

```
    <mb:BindingInformation
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
     xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
          <mb:Metadata>
           <slab:originatorConfidentialityLabel
            xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
             <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
             <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
             2015-09-30T12:30:00Z
            </slab:CreationDateTime>
           </slab:originatorConfidentialityLabel>
          </mb:Metadata>
         <mb:Data
          encoding="base64Binary"
          xmime:contentType="text/html">
          VGV4dHVhbCBpbGx1c3RyYXRpb24gb2YgZW5jYXBzdWxhdGluZyBCRE8=
         </mb:Data>
        </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
    </mb:BindingInformation>
```

Embedded BDO

This example illustrates a BDO that is embedded within an XML document. As the data object is an XML object the *DataReference URI* attribute value is interpreted as a Same-Document reference (see Section B.3.8.2.4). As such, for this example the null *URI* attribute value ‘""’ is interpreted as the whole data object (root element) is bound to the metadata.

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
</Document>
```

This example illustrates a BDO that is embedded within an XML document. As the data object is an XML object the *DataReference URI* attribute value is interpreted as a Same-Document reference (see Section B.3.8.2.4). As such, for this example the null *URI* attribute value ‘""’ is interpreted as the

whole data object (root element) is bound to the metadata. This example also shows how an XPath expression can be used to realise the same interpretation that the whole data object is bound to the metadata.[2]

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="">
     <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
       <ds:XPath>
        ancestor-or-self::*[local-name()='Document' and namespace-
uri()='http://example.com/doc']
       </ds:XPath>
      </ds:Transform>
     </ds:Transforms>
    </mb:DataReference>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
</Document>
```

Detached BDO

The following example shows a BDO that is referencing a physically separate data object to the BDO. By using a *DataReference URI* attribute value that is dereferenced using the HTTP (Reference [44]) uri scheme to locate the data object, the root node of the physically separate data object is bound to the metadata (as specified in Same-Document References Section B.3.8.2.4).

```
<mb:BindingInformation
```

[2] Note: the use of the XPath expression in this example is only for illustration. XPath expressions are used for binding a subset(s) of a data object(s) to the metadata. The main purpose for illustrating the use of XPath expressions was the use of the local-name and *uri-namespace* functions to avoid redefinitions of namespace prefixes.

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <DataReference
     URI="http://www.example.com/images/image.png"
     xmime:contentType="image/png" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

This example shows a BDO that is referencing an external data object to the BDO. Both the data object and the BDO are siblings of another data object. By using a *DataReference URI* attribute value that is dereferenced using a shortname XPointer to locate the data object, the external data object (in this example: `<Paragraph Id="para-2-2" />`) is bound to the metadata. This example also illustrates the use of the Same-Document reference as specified in Section B.3.8.2.4. In this case the *contentType* attribute of the *DataReference* element is not set and the default value of *"text/xml; charset=utf-8"* is assumed. As such, the syntax and semantics for interpreting the fragment identifier (value of the *DataReference URI* attribute) is conformant with XPointer (Reference [43]) as specified in Reference [18].

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="#para-2-1" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
</Document>
```

One-to-One Metadata Binding

All previous examples illustrate a one-to-one relationship of binding a single data object to a single metadata label.

One-to-Many Metadata Binding

This example depicts a detached BDO that binds multiple physically separated data objects to a single metadata label.

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <DataReference
     URI="http://www.example.com/images/image1.png"
     xmime:contentType="image/png" />
    <DataReference
     URI="http://www.example.com/images/image2.png"
     xmime:contentType="image/png" />
    <DataReference
     URI="http://www.example.com/images/image3.png"
     xmime:contentType="image/png" />
    <DataReference
     URI="http://www.example.com/images/image4.png"
     xmime:contentType="image/png" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
</Document>
```

Many-to-One Metadata Binding

The following example illustrates an embedded BDO that binds two metadata labels to a single data object.

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
     <slab:alternateConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:alternateConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
</Document>
```

Many-to-Many Metadata Binding

This example shows an embedded BDO that contains multiple *MetadataBinding* elements. The first *MetadataBinding* element binds the root element of the document to the two metadata labels. The second *MetadataBinding* element binds a subset of the data object (in this example: `<Introduction/>`) to two metadata labels.

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
 <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
     <slab:alternateConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:alternateConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="" />
   </mb:MetadataBinding>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>RESTRICTED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
     <slab:alternateConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>RESTRICTED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:alternateConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="">
     <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
       <ds:XPath>
        ancestor-or-self::*[local-name()='Introduction' and namespace-
uri()='http://example.com/doc']
       </ds:XPath>
      </ds:Transform>
     </ds:Transforms>
    </mb:DataReference>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

```
    </Document>
```

The following example shows a BDO that is referencing a physically separate data object to the BDO. By using a *DataReference URI* attribute value that is dereferenced using the HTTP (Reference [44]) uri scheme to locate the data object, the root node of the physically separate data object is bound to the metadata (as specified in Same-Document References Section B.3.8.2.4). As such, in this example, the root node of the data object is the root value of a JavaScript Object Notation (JSON; Reference [45]) document. As the *contentType* attribute value of the *DataReference* element is "*application/json*" the syntax and semantics for interpreting the fragment identifier are conformant with the JSON Pointer (Reference [46]) to bind the "*foo*" JSON data object to the metadata.

```
    <mb:BindingInformation
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
     <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
       <mb:Metadata>
        <slab:originatorConfidentialityLabel
         xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
         <slab:ConfidentialityInformation>
          <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
          <slab:Classification>UNCLASSIFIED</slab:Classification>
         </slab:ConfidentialityInformation>
         <slab:CreationDateTime>
          2015-09-30T12:30:00Z
         </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
       </mb:Metadata>
       <DataReference
        URI="http://www.example.com/example.json"
        xmime:contentType="application/json" />
      </mb:MetadataBinding>
      <mb:MetadataBinding>
       <mb:Metadata>
        <slab:originatorConfidentialityLabel
         xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
         <slab:ConfidentialityInformation>
          <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
          <slab:Classification>RESTRICTED</slab:Classification>
         </slab:ConfidentialityInformation>
         <slab:CreationDateTime>
          2015-09-30T12:30:00Z
         </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
       </mb:Metadata>
       <DataReference
        URI="http://www.example.com/example.json#foo"
        xmime:contentType="application/json" />
      </mb:MetadataBinding>
     </mb:MetadataBindingContainer>
    </mb:BindingInformation>
```

Annex C : Metadata Binding Profiles

This annex describes how the Binding Information is applied to specific data formats and protocols. It provides the normative specification for profiling the handling of the BDO dependent upon the type of data object or protocol. They are supported by these profiles in the following annexes:

- Web Services (SOAP-based and REST-based web services).
- Informal messaging (SMTP/MIME internet email);
- Collaboration (Text-based instant messaging);
- Document management (including Office Tools); and
- Arbitrary Files.

Appendix C-1: Simple Object Access Protocol (SOAP) Profile

C.1 SOAP Introduction

It is recognised that service providers and service consumers implementing web services based on SOAP operate under different frameworks and application contexts. As such, this profile includes support for both SOAP 1.1 (Reference [13]) and SOAP 1.2 (Reference [14]). Where there is a requirement to bind metadata to a SOAP data object exchanged between a service consumer and a service provider, that data object must adhere to this profile.

C.1.1 Namespace Constraints

The table below summarises the XML namespaces and corresponding prefixes used throughout for the binding of metadata to SOAP data objects and portions thereof.

Table 10: XML Namespaces and Prefixes

Prefix	Namespace
mb	urn:nato:stanag:4778:bindinginformation:1:0
	urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver
soap	http://schemas.xmlsoap.org/soap/envelope/ or http://www.w3.org/2003/05/soap-envelope
soap11	http://schemas.xmlsoap.org/soap/envelope/
soap12	http://www.w3.org/2003/05/soap-envelope
wsa	http://www.w3.org/2005/08/addressing
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsse11	http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd
xs	http://www.w3.org/2001/XMLSchema
xsi	http://www/w3.org/2001/XMLSchema-instance

C.1.2 Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Section B.3.4 and Section B.3.8.
- `Courier font` indicates syntax derived from various W3C XML Signature (Reference [15]) and SOAP (References [13], [14]) standards referenced in this Annex.

C.1.3 SOAP Message Structure

The SOAP message structure is specified in (References [13], [14]). Dependent upon system information exchange requirements it may be necessary that the whole SOAP message is bound to the metadata or subsets of the SOAP message are bound to the metadata. As such, Binding Information SHALL be represented either as: an Embedded BDO; or, a Detached BDO.

The BDO is contained in a `Security` header that must include the *BindingInformation* element only (as a child element of the `Security` element).

If the SOAP message is SOAP 1.1 the `Security @actor` attribute must be included with a value of *urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver*.

If the SOAP message is SOAP 1.2 the `Security @role` attribute must be included with a value of *urn:nato:stanag:4778:bindinginformation:1:0:role:bindingInformationReceiver*.

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

An example of a BDO embedded in a SOAP 1.1 message that illustrates the binding of the SOAP message to metadata is provided in Figure 13. Also illustrated is the use of the `actor` attribute to support multiple Security elements. This example uses Confidentiality Metadata Labels (Reference [9]) as example metadata.

```
<soap11:Envelope xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
 <soap11:Header>
  <wsse:Security
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
   soap11:actor="http://www.nato.int/2015/06/nl/mb/role/bindingInformationReceiver">
   <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <mb:MetadataBindingContainer>
     <mb:MetadataBinding>
      <mb:Metadata>
       <slab:originatorConfidentialityLabel
        xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
        <slab:ConfidentialityInformation>
         <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
         <slab:Classification>UNCLASSIFIED</slab:Classification>
        </slab:ConfidentialityInformation>
        <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
       </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference URI="" />
     </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
   </mb:BindingInformation>
  </wsse:Security>
 </soap11:Header>
 <soap11:Body>
  <Track xmlns="http://example.com/trackInformation">
   ....
  </Track>
 </soap11:Body>
</soap11:Envelope>
```

Figure 13: Example Embedded BDO for SOAP

An example of a detached BDO contained in a SOAP 1.1 message that illustrates the binding of an external data object in the SOAP body to metadata is provided in Figure 14. This example uses Confidentiality Metadata Labels (Reference [9]) as example metadata.

Figure 14 illustrates the use of XPointer and XPath to reference the data object. Also illustrated is the use of the `actor` attribute to support multiple Security elements.

```
<soap11:Envelope xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
 <soap11:Header>
  <wsse:Security
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
   soap11:actor="http://www.nato.int/2015/06/nl/mb/role/bindingInformationReceiver">
   <mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <mb:MetadataBindingContainer>
     <mb:MetadataBinding>
      <mb:Metadata>
       <slab:originatorConfidentialityLabel
        xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
        <slab:ConfidentialityInformation>
         <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
         <slab:Classification>UNCLASSIFIED</slab:Classification>
        </slab:ConfidentialityInformation>
        <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
       </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:DataReference URI="">
       <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
         <ds:XPath>
          ancestor-or-self::*[local-name()='Track' and namespace-
uri()='http://example.com/trackInformation']
         </ds:XPath>
        </ds:Transform>
       </ds:Transforms>
      </mb:DataReference>
     </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
   </mb:BindingInformation>
  </wsse:Security>
 </soap11:Header>
 <soap11:Body>
  <Track xmlns="http://example.com/trackInformation">
   ....
  </Track>
 </soap11:Body>
</soap11:Envelope>
```

Figure 14: Example Detached BDO for SOAP

Appendix C-2: Representational State Transfer (REST) Profile

C.2 RESTful Architecture

REST is an architectural style defined as a set of constraints on a distributed hypermedia system and implemented by a set of standard protocols that adhere to these constraints. The REST architectural style can be employed for implementing web services which are known as RESTful web services. RESTful web services rely upon the Hypertext Transport Protocol (HTTP) (Reference [44]) as the standard interface between service providers and service consumers utilizing the HTTP verbs GET, PUT, POST, DELETE, etc. in their specified manner. Resources that are exposed through RESTful web services are identified by URIs and are represented to service consumers in any (mutually agreed) media type format. In other words, a URI identifies a resource, rather than a representation, and when a service consumer asks a service provider for a resource, the service provider will respond with the best possible representation for that resource, given the service consumer's preferences. In an environment where data objects must have bound metadata, the resource identified in the URI will already contain a BDO (detached, encapsulating or embedded). As such, there is no requirement for metadata binding that is specific for REST.

C.2.1 Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Section B.3.4 and Section B.3.8.
- `Courier font` indicates syntax derived from SIO[3]-Label (Reference [23]) and HTTP (Reference [44]) referenced in this Annex.

C.2.2 HTTP Request/Response for RESTful Web Services

However, in the cases where there is a requirement for BDOs to be located in the HTTP protocol layer it is RECOMMENDED the SIO-Label (Reference [23]) as a HTTP Entity message header for HTTP Entity requests and responses.

The BDO is an embedded BDO that MUST contain at least one *MetadataBinding* that contains a null *DataReference URI* attribute value (refer to Same-Document References Section B.3.8.2.4) that semantically indicates a binding relationship to the HTTP Entity message request or response.

The *DataReference xmime:contentType* attribute MUST be present with a value of *`message/http`*.

The BDO MUST be included in the SIO-Label header `label` parameter.

The SIO-Label `label` parameter value MUST be the `base64` encoding of the BDO.

[3] SIO stands for Security Information Object, as defined in X.841

The SIO-Label `label` parameter MUST not support Reference [61] for parameter value continuation.

The SIO-Label `type` parameter MUST be present with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

Figure 15 illustrates an HTTP POST request with the SIO-Label HTTP header field with the value as specified in this Annex. Figure 16: Base64 Decoded Embedded BDO illustrating the binding of the HTTP POST illustrates the base64 decoded value of the `label` value parameter. This example uses Confidentiality Metadata Labels (Reference [9]) as example metadata.

```
POST /token HTTP/1.1
Host: server.example.com
SIO-Label: type="urn:nato:stanag:4778:bindinginformation:1:0" label="<base64 encoded
BDO>"
Content-Type: text/xml

<Document>
….
</Document>
```

Figure 15: An example HTTP POST Request which includes an embedded BDO

```
<mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <mb:DataReference URI="" xmime:contentType="message/http"/>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 16: Base64 Decoded Embedded BDO illustrating the binding of the HTTP POST

Appendix C-3: Simple Mail Transfer Protocol (SMTP) Profile

C.3 SMTP Introduction

This profile specifies the mechanism for binding metadata to Internet Email (Reference [22]) including MIME entities. A MIME entity can be a sub-part, sub-parts of a message or the message with all its sub-parts. A MIME entity that is the message includes only the MIME message headers and MIME body, and does not include the Internet Email headers. For the purposes of this Appendix a message is an Internet Email conformant with Reference [22] that can optionally include MIME entities.

This profile does not support the capability for referencing subsets of Internet Email headers.

C.3.1 Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Section B.3.4 and Section B.3.8.
- `Courier font` indicates syntax derived from SIO[4]-Label (Reference [23]), Message-ID ((Reference [24])) and Content-ID (Reference [24]) URI schemes and MIME Entities (Reference [47]) referenced in this Appendix.

C.3.2 Internet Email Structure

The BDO is an embedded BDO that MUST contain at least one *MetadataBinding* that contains a *DataReference URI* attribute value conformant with the Message-ID Uniform Resource Locator, `mid`, scheme according to (Reference [24]). By conforming to Reference [24] to syntactically and semantically interpret the *DataReference URI* attribute allows for the metadata to be bound to the entire message.

The *DataReference xmime:contentType* attribute value is NOT REQUIRED when the *URI* attribute value is the `mid` URI scheme.

The *DataReference xmime:contentType* attribute is present the value SHALL be *`message/rfc822`* when the *URI* attribute value is the `mid` URI scheme.

This profile requires that the SIO-Label header field as specified in (Reference [23]) is used to embed the BDO within the Informal Email.

The BDO MUST be included in the SIO-Label header `label` parameter.

The SIO-Label `label` parameter value MUST be the `base64` encoding of the BDO.

The SIO-Label `type` parameter MUST be present with the value *urn:nato:stanag:4778:bindinginformation:1:0*.

[4] SIO stands for Security Information Object, as defined in X.841

It must be noted that the `label` parameter SHALL conform with Reference [61] (as specified in Reference [23]) specifically in relation to parameter value continuation.

Depending upon the line length limit (recommended to be 78 characters or less and not more than 998 characters – see Reference [22]) the `label` parameter SHALL be split into multiple `label` parameters, as illustrated below.

label*0="PFNlY0xhYmVsIHhtbG5zPSJodHRwOi8vZXhhbX";

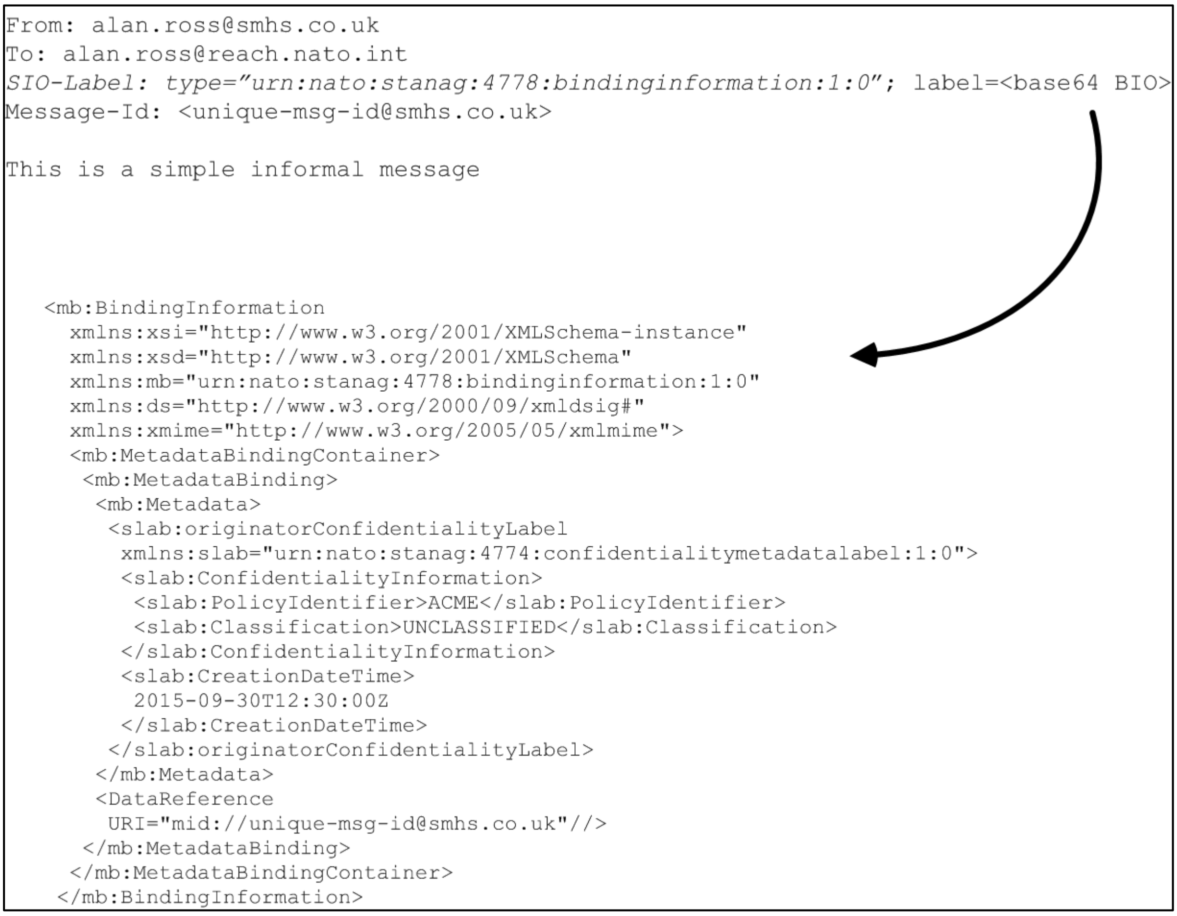
label*1="BsZS5jb20vc2VjLWxhYmVsLzAiPjxQb2xpY3lJ";

label*2="ZGVudGlmaWVyIFVSST0idXJuOm9pZDoxLjEiLz";

label*3="48Q2xhc3NpZmljYXRpb24+MzwvQ2xhc3NpZmlj";

label*4="YXRpb24+PC9TZWNMYWJlbD4=";

An example of an Embedded BDO contained in the SIO-Label header field of an Informal Email that illustrates the binding of Confidentiality Metadata Labels (Reference [9]) as example metadata to the message is provided in Figure 17.



```
From: alan.ross@smhs.co.uk
To: alan.ross@reach.nato.int
SIO-Label: type="urn:nato:stanag:4778:bindinginformation:1:0"; label=<base64 BIO>
Message-Id: <unique-msg-id@smhs.co.uk>

This is a simple informal message




   <mb:BindingInformation
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
     xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
     <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
       <mb:Metadata>
        <slab:originatorConfidentialityLabel
         xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
         <slab:ConfidentialityInformation>
          <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
          <slab:Classification>UNCLASSIFIED</slab:Classification>
         </slab:ConfidentialityInformation>
         <slab:CreationDateTime>
          2015-09-30T12:30:00Z
         </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
       </mb:Metadata>
       <DataReference
        URI="mid://unique-msg-id@smhs.co.uk"//>
      </mb:MetadataBinding>
     </mb:MetadataBindingContainer>
    </mb:BindingInformation>
```

Figure 17: Example of Binding Confidentiality Metadata Label to Email

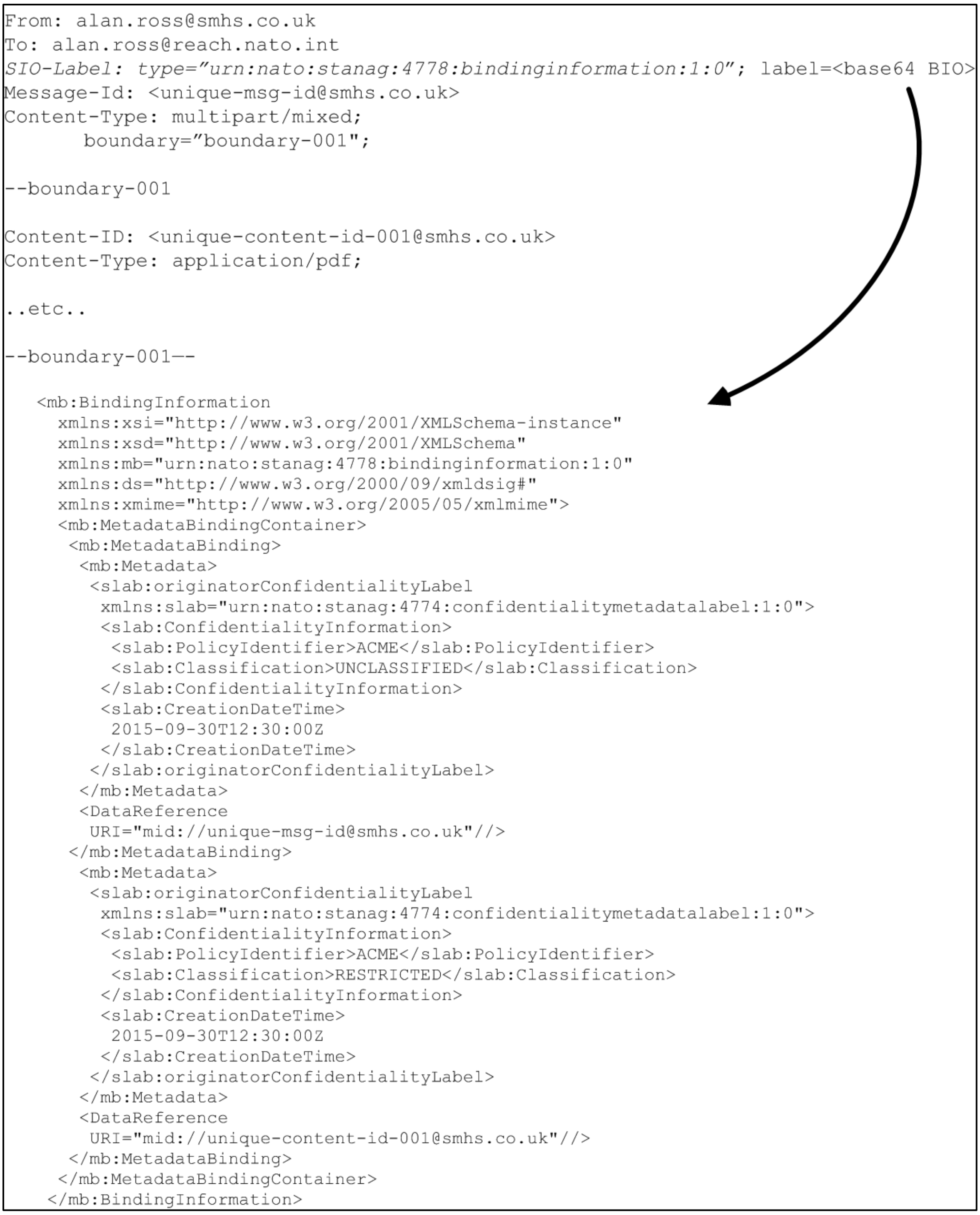
In the case where metadata is to be bound to individual MIME bodyparts, the *URI* attribute of the *DataReference* element MUST use the Content-ID Uniform Resource Locator, `cid`, scheme according to (Reference [24]).

The *DataReference xmime:contentType* attribute MUST NOT be present.

The MIME `Content-Type` header field that indicates the internet media type of the MIME bodypart MUST be used to infer the content type of the data object referenced by the *DataReference URI* attribute value.

The example provided in Figure 18 illustrates an Embedded BDO contained in the SIO-Label header field of an informal email where Confidentiality Metadata Labels (Reference [9]) as example metadata are bound to:

1) a message; and,
2) a MIME bodypart included in the message.



```
From: alan.ross@smhs.co.uk
To: alan.ross@reach.nato.int
SIO-Label: type=”urn:nato:stanag:4778:bindinginformation:1:0”; label=<base64 BIO>
Message-Id: <unique-msg-id@smhs.co.uk>
Content-Type: multipart/mixed;
       boundary=”boundary-001";

--boundary-001

Content-ID: <unique-content-id-001@smhs.co.uk>
Content-Type: application/pdf;

..etc..

--boundary-001—-

   <mb:BindingInformation
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
     xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
     <mb:MetadataBindingContainer>
      <mb:MetadataBinding>
       <mb:Metadata>
        <slab:originatorConfidentialityLabel
         xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
         <slab:ConfidentialityInformation>
          <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
          <slab:Classification>UNCLASSIFIED</slab:Classification>
         </slab:ConfidentialityInformation>
         <slab:CreationDateTime>
          2015-09-30T12:30:00Z
         </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
       </mb:Metadata>
       <DataReference
        URI="mid://unique-msg-id@smhs.co.uk"//>
      </mb:MetadataBinding>
       <mb:Metadata>
        <slab:originatorConfidentialityLabel
         xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
         <slab:ConfidentialityInformation>
          <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
          <slab:Classification>RESTRICTED</slab:Classification>
         </slab:ConfidentialityInformation>
         <slab:CreationDateTime>
          2015-09-30T12:30:00Z
         </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
       </mb:Metadata>
       <DataReference
        URI="mid://unique-content-id-001@smhs.co.uk"//>
      </mb:MetadataBinding>
     </mb:MetadataBindingContainer>
    </mb:BindingInformation>
```

Figure 18: Example Binding of Confidentiality Metadata Labels to Email Message and Attachments

Appendix C-4: Extensible Message and Presence Protocol (XMPP) Profile

C.4 XMPP Introduction

Confidentiality metadata labels can be supported in XMPP stanzas as indicated by XEP-0258 (Reference [28]) whereby a mechanism for carrying Enhanced Security Services (ESS) Security labels (Reference [8]) is standardized. This profile extends the XEP-0258 (Reference [28]) specification to support carrying an Embedded or Detached BDO for `Message` stanzas. This profile supports the XMPP use cases for one-to-one instant messaging and multi-user chat.

Future profiles for XMPP will specify support for carrying BDOs in `IQ` stanzas specifically to support Publish Subscribe mechanisms such as those defined in XEP-0060 Publish-Subscribe (Reference [29]).

C.4.1 Notational Conventions

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Section B.3.4 and Section B.3.8.
- `Courier font` indicates syntax derived from XMPP (References [48], [49] and [31]) and XEP-0258 (Reference [28]) referenced in this Appendix.

C.4.2 Message Stanza Structure

The `Message` stanza structure is specified in (Reference [49]). Dependent upon system information exchange requirements it may be necessary that the `Message` stanza is bound to the metadata or subsets of the `Message` stanza are bound to the metadata. As such, Binding Information SHALL be represented either as: an Embedded BDO; or, a Detached BDO.

Figure 19 illustrates the high-level structure of a `Message` stanza that contains an Embedded BDO contained within a XEP-0258 `securitylabel` element.

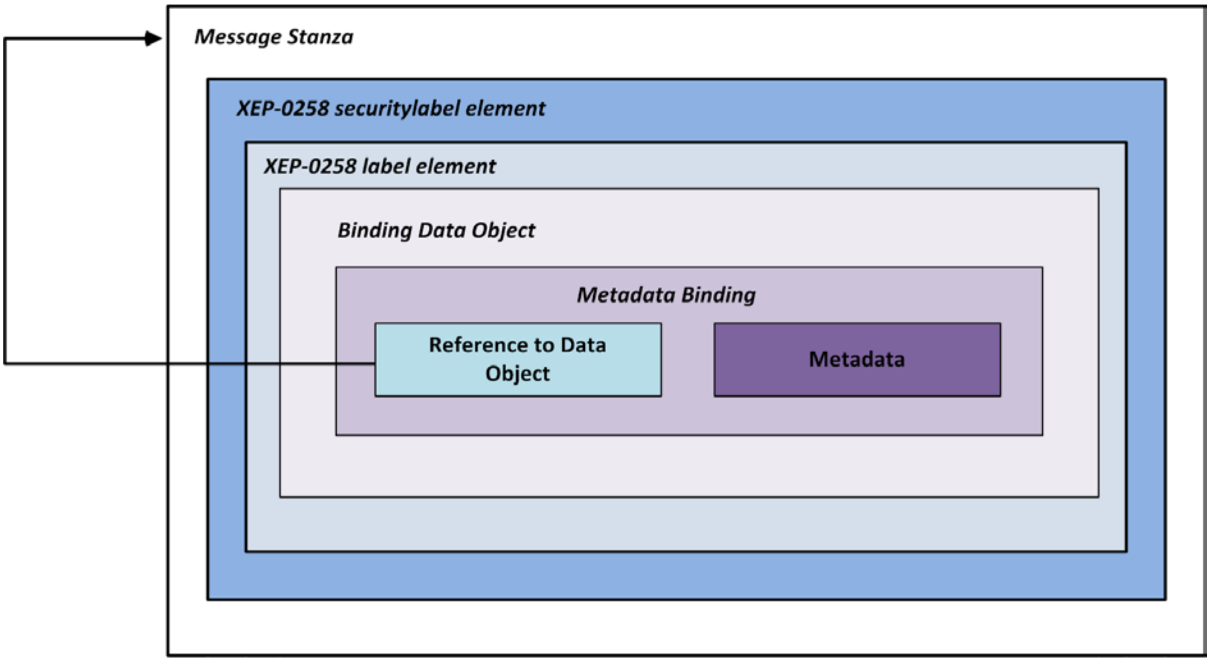
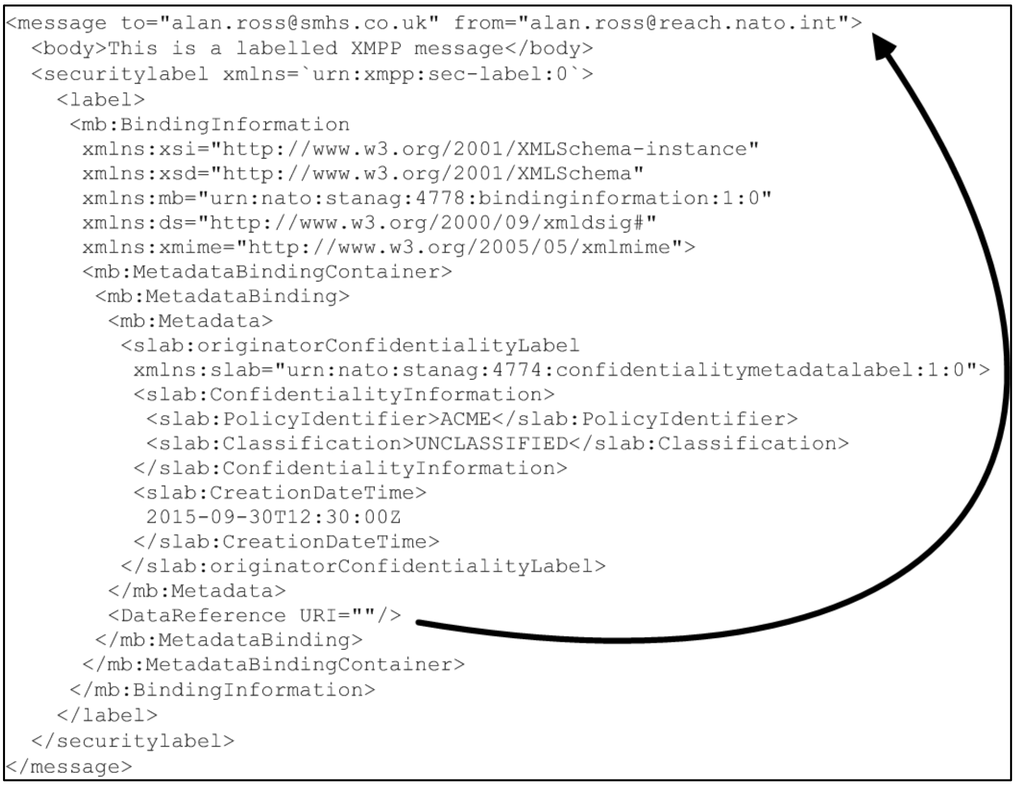


Figure 19: Structure of Message Stanza Containing Embedded BDO

The BDO is contained in a `label` child element of a XEP-0258 `securitylabel` element that must include the *BindingInformation* element only (as a child element of the `label` element).

It is RECOMMENDED that metadata is contained within the *Metadata* child element of the *MetadataBinding* element; not referenced with the use of the *MetadataReference* element.

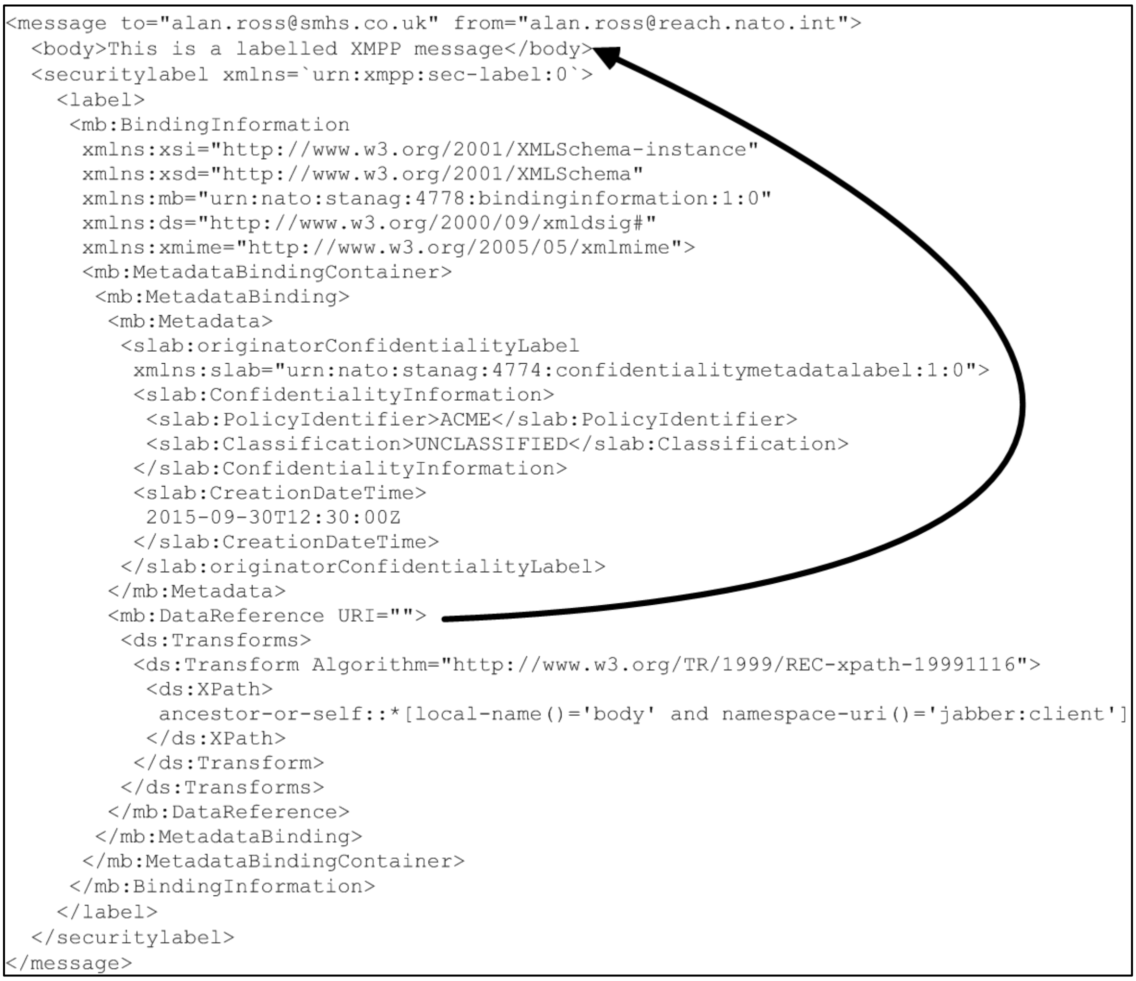
An example of a BDO embedded in a `Message` stanza that illustrates the binding of the entire Message stanza to metadata is provided in Figure 20. This example uses Confidentiality Metadata Labels (Reference [9]) as example metadata.



```
<message to="alan.ross@smhs.co.uk" from="alan.ross@reach.nato.int">
  <body>This is a labelled XMPP message</body>
  <securitylabel xmlns=`urn:xmpp:sec-label:0`>
    <label>
     <mb:BindingInformation
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      <mb:MetadataBindingContainer>
       <mb:MetadataBinding>
        <mb:Metadata>
         <slab:originatorConfidentialityLabel
          xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
           <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
           <slab:Classification>UNCLASSIFIED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
           2015-09-30T12:30:00Z
          </slab:CreationDateTime>
         </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <DataReference URI=""/>
       </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
     </mb:BindingInformation>
    </label>
  </securitylabel>
</message>
```

Figure 20: Example Embedded Binding Data Object for Message Stanza (XMPP)

An example of a detached BDO contained in a `Message` stanza that illustrates the binding of the `body` element (child of the `Message` stanza) to metadata is provided in Figure 20. This example illustrates the use of XPaths for referencing the `body` element. This example uses Confidentiality Metadata Labels (Reference [9]) as example metadata.



```
<message to="alan.ross@smhs.co.uk" from="alan.ross@reach.nato.int">
  <body>This is a labelled XMPP message</body>
  <securitylabel xmlns=`urn:xmpp:sec-label:0`>
    <label>
     <mb:BindingInformation
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      <mb:MetadataBindingContainer>
       <mb:MetadataBinding>
        <mb:Metadata>
         <slab:originatorConfidentialityLabel
          xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
           <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
           <slab:Classification>UNCLASSIFIED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
           2015-09-30T12:30:00Z
          </slab:CreationDateTime>
         </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI="">
         <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
           <ds:XPath>
            ancestor-or-self::*[local-name()='body' and namespace-uri()='jabber:client']
           </ds:XPath>
          </ds:Transform>
         </ds:Transforms>
        </mb:DataReference>
       </mb:MetadataBinding>
      </mb:MetadataBindingContainer>
     </mb:BindingInformation>
    </label>
  </securitylabel>
</message>
```

Figure 21 Example Detached Binding Data Object Contained in Message Stanza (XMPP)

Appendix C-5: Office Open XML Formats Profile

C.5 OOXML Introduction

The Office Open XML Formats (OOXML) are defined ISO/IEC 29500 (Reference [33]) and offer standards for representing office documents, including spreadsheets, presentations and word processing documents.

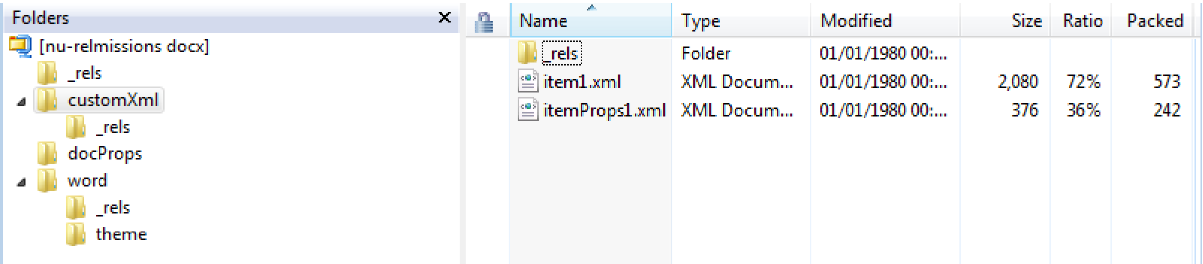
OOXML adopts a structured format which consists of a number of XML-based files packaged into an archive file according to the Open Packaging Conventions (OPC), which is defined in Part 2 of ISO/IEC 29500 (Reference [33]).

OOXML allows for custom XML files to be included within the package without impacting the underlying application. This provides a mechanism for a metadata to be bound to the OOXML document and maintained within the package.

This profile for the OOXML describes how metadata should be maintained.

C.5.1 Structure

The structure of an OOXML package consists of a number of folders which contain different components of the document.



Folders

- [nu-relmissions docx]
  - _rels
  - customXml
    - _rels
  - docProps
  - word
    - _rels
    - theme

Name	Type	Modified	Size	Ratio	Packed
_rels	Folder	01/01/1980 00:...			
item1.xml	XML Docum...	01/01/1980 00:...	2,080	72%	573
itemProps1.xml	XML Docum...	01/01/1980 00:...	376	36%	242

Figure 22: General Structure of an OPC Package

The structure, as shown in Figure 22, generally consists of:

- An application specific folder, for example “word”, “ppt” or “xl”.
- A “customXml” folder in which arbitrary XML files can be stored.
- A “docProps” folder in which core and custom document properties are held.
- Multiple “_rels” folder which contains details of the parts within a folder.

This structure is then packaged into an archive file with an application specific extension (for example, .docx).

The document that is displayed to a user is generally split over a number of different XML files contained with the package. This does not present a problem when applying granular metadata to different parts of the document.

However care must be taken when the intention is to bind metadata to the complete document (refer to Microsoft Office File Types section below for normative text related to binding metadata to a whole document). For example, the XML file /word/document.xml within a Microsoft Word OPC package does not contain the headers or footers of the document (these are contained in the separate files /word/header1.xml and /word/footer1.xml.)

C.5.2 Custom XML

In order to support metadata binding within an OPC package, a single CustomXML file SHALL be maintained within the OPC package with the Metadata Binding Container namespace, “urn:nato:stanag:4778:bindinginformation:1:0”.

DataReference elements SHALL be used to reference the files within the OPC package.

Data elements SHALL NOT be used.

When referring to files, or portions of files, within the OPC package, absolute URIs from the package root SHALL be used with the *DataReference* element. For example,

<DataReference URI=”/word/document.xml”/>

Microsoft Office File Types

Microsoft Office, since Microsoft Office 2007, has used the OOXML standard for a number of its document types, as shown in Table 11.

Table 11: Microsoft Use of OOXML Standard

Application	Extension	Application Folder in Package	Whole Document Package Files
Microsoft Word	.docx	word	/word/document.xml
			/word/header<N>.xml
			/word/footer<N>.xml
			/word/media/*
			/word/footnotes.xml
			/word/endnotes.xml
			/docProps/app.xml
			/docProps/core.xml
			/docProps/custom.xml
Microsoft Excel	.xslx	xl	/xl/workbook.xml
			/xl/worksheets/sheet<N>,xml
			/docProps/app.xml
			/docProps/core.xml
			/docProps/custom.xml
Microsoft PowerPoint	.pptx	ppt	/ppt/presentation.xml
			/ppt/slideLayout.xml
			/ppt/slideMaster/slideMaster<N>.xml
			/ppt/slides/slide<N>.xml
			/ppt/presProps.xml
			/ppt/viewProps.xml
			/docProps/app.xml
			/docProps/core.xml
			/docProps/custom.xml

When binding metadata to a complete document (as opposed to a specific part), all of the files in the “Whole Document Package Files” (see Table 11) SHALL be referenced in the binding.

Figure 23 shows the contents of a CustomXML file, stored in customXml/item1.xml for a Microsoft Word document.

```
<mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <DataReference
     URI="word/document.xml"/>
    <DataReference
     URI="word/header1.xml"/>
    <DataReference
     URI="word/footer1.xml"/>
    <DataReference
     URI="word/media/image.jpeg"
     xmime:contentType="image/jpeg" />
    <DataReference
     URI="word/footnotes.xml"/>
    <DataReference
     URI="word/endnotes.xml"/>
    <DataReference
     URI="docProps/app.xml"/>
    <DataReference
     URI="docProps/core.xml"/>
    <DataReference
     URI="docProps/custom.xml"/>
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 23: Example CustomXML file

Appendix C-6: Generic Open Packaging Convention (OPC) Profile

C.6 OPC Introduction

There are many other files types that do not use the OOXML format, and do not currently have a profile for supporting the *BindingInformation* with their own file format.

It is therefore useful to define a generic mechanism that can be used with an arbitrary file.

This profile defines a packaging mechanism, based upon the Open Packaging Container defined in OOXML, to associate a file and the *BindingInformation* into a single, archive file.

C.6.1 File Package

One of the common ways to package a number of files together is to use the archive file format. An archive file may contain a number of different files and an associated folder structure.

This profile adopts the Open Packaging Conventions (OPC) as defined as Part 2 of the Office Open XML specification (Reference [33]).

By adopting OPC this profile provides a structured and consistent mechanism for associating *BindingInformation* with a data object within an archive file.

This profile uses the same customXml files and relationships within the archive file as those defined in the OOXML profile, as shown in Figure 24.

Specifically:

- A top-level relationship within the package is defined which identifies the file with which the *BindingInformation* will be associated.
- The file is held in a folder called "files"
- The *BindingInformation* is held within a file called "customXml".
- The *DataReference* URI attribute is specified as the full path to the file.
- A relationship is defined between the file and the *BindingInformation*.

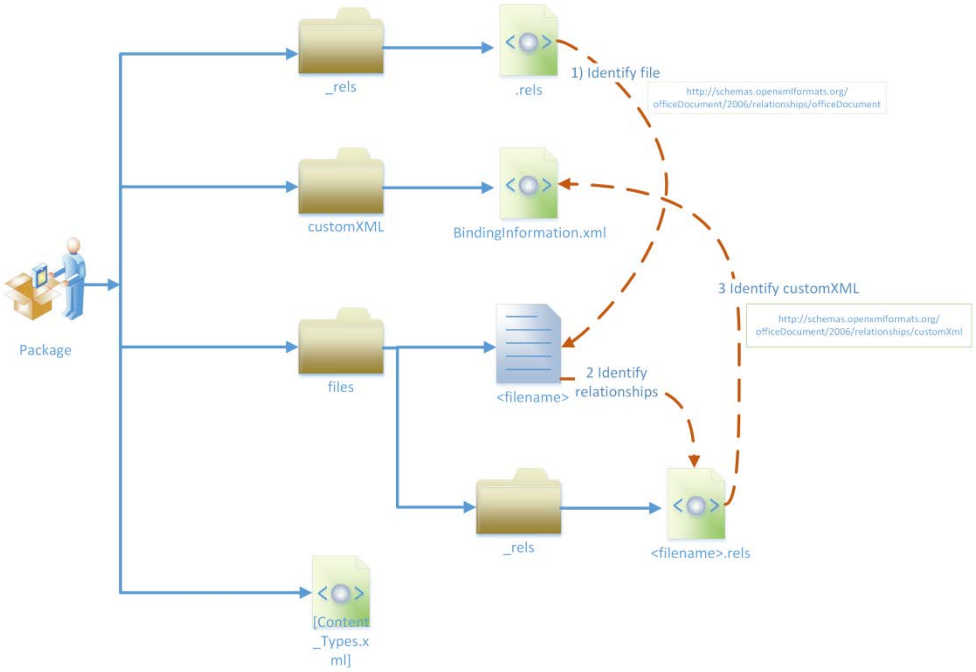


Figure 24: OPC Structure for packaging BindingInformation with an arbitrary file

This approach allows multiple files, of different types, to be held within the same package and be bound to distinct metadata. Figure 25 shows an example customXML file for a package containing the file “image1.jpeg”.

```
<mb:BindingInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
  <mb:MetadataBindingContainer>
   <mb:MetadataBinding>
    <mb:Metadata>
     <slab:originatorConfidentialityLabel
      xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
      <slab:ConfidentialityInformation>
       <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
       <slab:Classification>UNCLASSIFIED</slab:Classification>
      </slab:ConfidentialityInformation>
      <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
     </slab:originatorConfidentialityLabel>
    </mb:Metadata>
    <DataReference
     URI="files/image1.jpeg"
     xmime:contentType="image/jpeg" />
   </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
 </mb:BindingInformation>
```

Figure 25: Example Packaged CustomXML file

Appendix C-7: Sidecar Files Profile

C.7 Sidecar Files Introduction

If a file cannot be packaged (for example, if it is a file on a file share which needs to be accessed using the original applications), a simple naming convention to relate the BDO with the data object is proposed.

Sidecar files allow the association of metadata with a data object for which there is no profile.

This approach is well known and understood for associating data (typically metadata) with other data of a different format.

C.7.1 File Package

A simple naming convention is defined that allows the Binding Data Object to be maintained in a separate, but identifiable, file to the data object file, as shown in Figure 26.

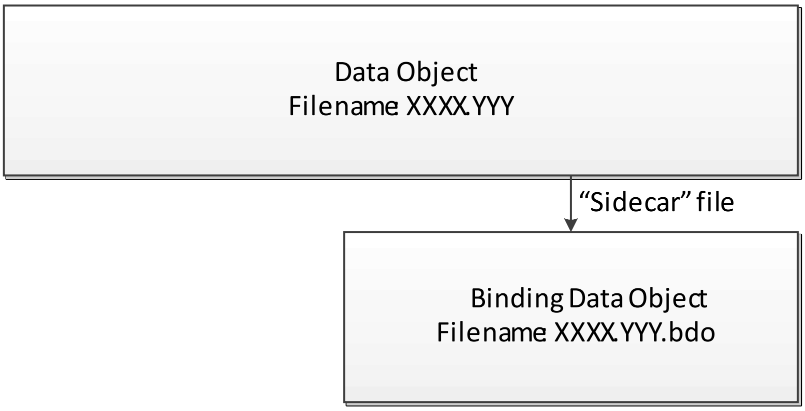


Figure 26: BDO as a Sidecar File

The name of the Binding Data Object file SHALL be the same as the data object file, with a further ".bdo" suffix.

Values used in *DataReference* URI with the BDO SHALL use relative paths and assume that the data object resides at the same location as the BDO.

For example, distinct metadata may be associated with an image file, "image1.jpeg", by creating a *BindingInformation* element and storing it as "image1.jpeg.bdo" in the same folder as the original file.

Figure 27 shows an example sidecar file for "image1.jpeg".

```
     <mb:BindingInformation
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
       <mb:MetadataBindingContainer>
        <mb:MetadataBinding>
         <mb:Metadata>
          <slab:originatorConfidentialityLabel
     xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
     <slab:ConfidentialityInformation>
      <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
      <slab:Classification>UNCLASSIFIED</slab:Classification>
     </slab:ConfidentialityInformation>
     <slab:CreationDateTime>
      2015-09-30T12:30:00Z
     </slab:CreationDateTime>
    </slab:originatorConfidentialityLabel>
   </mb:Metadata>
   <DataReference URI="./image1.jpeg" xmime:contentType="image/jpeg" />
  </mb:MetadataBinding>
 </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Figure 27: Example Sidecar file

Annex D : Cryptographic Artefact Profiles

A metadata binding provides additional information specifying which metadata belongs to which data object(s) and provides a verifiable reference between metadata and data. A non-cryptographic binding provides a reference between the metadata and the data. This reference can be structurally verified to be correct. However, no assumptions besides this can be made. In contrast, cryptographic bindings are used to provide a certain level of integrity protection, and authenticity and non-repudiation of the entity that generated the metadata binding.

A cryptographic binding (that includes cryptographic artefacts) uses cryptographic techniques and mechanisms like cryptographic digests, message authentication codes or digital signatures in order to protect the binding. Such cryptographic techniques and mechanisms are subject to the level of assurance required for protecting the integrity of the binding and for establishing confidence for the authenticity of the entity creating the binding. The level of assurance required for protecting the integrity of the binding and for establishing confidence for the authenticity of the entity creating the binding is a matter for organizational, national or federation security policies. As such, Appendix 3 does not specify mandatory to implement cryptographic techniques or mechanisms for generating a cryptographic artefact. However, the intention of Appendix 3 is to profile the use of cryptographic protocols, which can be used to implement support for different cryptographic techniques and mechanisms, for generating cryptographic artefacts to be stored in a cryptographic binding.

The first version of Appendix 3 will profile the XML Signature (XMLDSIG, Reference [15]) cryptographic protocol for generating a cryptographic artefact using digital signatures and key-hashed message authentication code (HMAC, Reference [51]) as the cryptographic techniques and mechanisms.

Table 12 below lists the supported cryptographic protocols and cryptographic mechanisms that are profiled for generating cryptographic artefacts.

Table 12: Supported Cryptographic Protocols and Mechanisms Profiles

Cryptographic Protocol	Cryptographic Mechanism	Reference
XML Signature (Reference [15])	Digital Signature	Appendix 3 Annex D.1 Appendix 3 Annex D.2
	Keyed-Hash Message Authentication Code	Appendix 3 Annex D.1 Appendix 3 Annex D.3

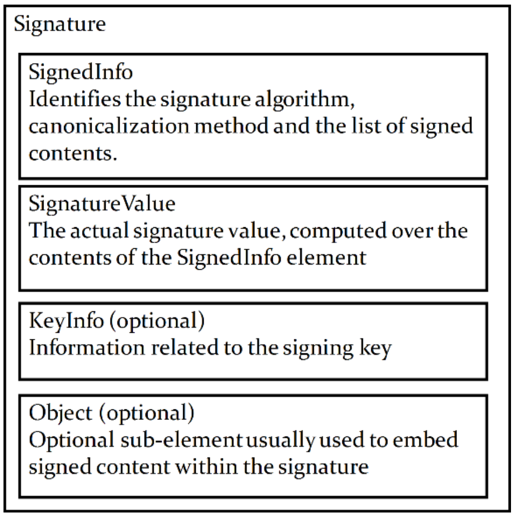
Further revisions to this appendix may be required to profile other cryptographic protocols such as Secure/Multipurpose Internet Mail Extensions (SMIME, Reference [52]) or JSON Web Signature (JWS, Reference [53]), for example, or to update supported cryptographic algorithms by either introducing new algorithms or deprecating existing algorithms.

Appendix D-1: XML Signature Cryptographic Artefact Profile

XML Signature (XMLDSIG, Reference [15]) offers powerful and flexible mechanisms that can support a wide variety of cryptographic requirements. XMLDSIG provides integrity, authentication and non-repudiation services for data (including metadata) of any type. XMLDSIG is applied to arbitrary data whereby a data object is digested with the resulting value stored in an element which is then digested and cryptographically signed. XMLDSIG indicates the location of the data object either by reference (in the case of an enveloped or detached signature) or by value (in the case of an enveloping signature whereby the signature contains the data object that is to be signed).

In order to highlight the differences and avoid duplication of text from XMLDSIG, a delta specification approach has been taken. Annex C will refer to the relevant sections of XMLDSIG and will identify any necessary clarifications and/or amendments to these sections. This approach provides traceability and puts the delta text in context. It is required that Annex C is read together with XMLDSIG.

Figure 28 illustrates the structure of an XML Signature element including the primary sibling elements: `SignedInfo`; `SignatureValue`; `KeyInfo`; and, `Object`.



Signature

SignedInfo
Identifies the signature algorithm, canonicalization method and the list of signed contents.

SignatureValue
The actual signature value, computed over the contents of the SignedInfo element

KeyInfo (optional)
Information related to the signing key

Object (optional)
Optional sub-element usually used to embed signed content within the signature

Figure 28: XML Signature Structure

This Annex will use the same structure as illustrated in Figure 28 to profile those requirements that are generic for XML Signature based cryptographic artefacts and to further refine those requirements for cryptographic artefacts generated with the use of digital signatures or keyed-hash message authentication codes). In particular this Annex will be divided into the following sub sections:

- General requirements for XMLDSIG including `SignedInfo`, `SignatureValue` and `Object` elements (refer to D.1);
- Specific requirements for XMLDSIG `SignedInfo` and `KeyInfo` elements related to digital signatures (refer to Annex D.2); and,
- Specific requirements for XMLDSIG `SignedInfo` and `KeyInfo` elements related to keyed-hashed message authentication codes (refer to Annex D.3).

Example Binding Data Objects containing cryptographic artefacts conformant with this profile are illustrated in Annex D.

The notational conventions used for this Annex are as follows:

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in *italics* indicate terms derived from Section B.3.4 and Section B.3.8.
- `Courier font` indicates syntax derived from various W3C XML Signature (Reference [15]) standard referenced in this Annex.
- *`Courier font`* indicates syntax derived from Web Services Security (WSS) (Reference [41]) standard Section 10 referenced in this Annex.

D.1 General XMLDSIG Requirements

Unless otherwise stated, all statements that apply to XMLDSIG also apply to this profile.

An entity that creates XML Signatures conformant with this profile (known as Originator) is REQUIRED to perform the processing rules for Core Generation as specified in XMLDSIG Section 3.1.

An entity that interprets and processes XML Signatures conformant with this profile (known as Recipient) is REQUIRED to perform the processing rules for Core Validation as specified in XMLDSIG Section 3.2.

D.1.1 Signature Types

Three types of signatures exist in XMLDSIG: enveloping signatures whereby the signature envelopes the data object to be signed; enveloped signatures whereby the signature is embedded within the data object; and, detached signatures whereby the signature and the data object reside independently.

Enveloping, Enveloped and Detached signature types are supported in this profile.

D.1.2 Same-Document URI-References

This section refers to XMLDSIG Section 4.3.3.1, 4.3.3.2 and 4.3.3.3.

The significance of the URI fragment identifier for dereferencing subsets of data objects is a function of the type (media type) of the data object. Identification for the media type of a data object is supported in the general binding mechanism with the use of the *xmime:contentType* attribute. The *xmime:contentType* attribute for non-XML is a required attribute of the *DataReference* and *MetadataReference* elements.

In the case where the *xmime:contentType* attribute is present in the *DataReference* or *MetadataReference* element, the *xmime:contentType* attribute value specifies a non-XML data object type and the URI attribute value of the *DataReference* or *MetadataReference* element is deemed to be a '`same-document`' reference (as specified in XMLDSIG Section 4.3.3.2) the following requirements are to be followed:

- Originator MUST create a `Manifest` element for each *DataReference* or *MetadataReference* elements contained in the *bindingInformation* that includes a `Reference` element (as specified in Manifest section of D.1);
- The `Manifest` element that the Originator creates MUST be stored as a child element of an `Object` element;
- Recipient SHOULD perform the following additional Core Validation processing rules:
  - For each `Reference` in the `Manifest`:
    - Obtain the data object to be digested located by the `URI` attribute in the `Reference` element (According to the semantics specified for the URI fragment identifier defined by the media type);
    - Digest the resulting data object using the `DigestMethod` (as specified in the Reference section in of D.1).
    - Compare the generated digest value against `DigestValue` in the `Manifest Reference`; if there is any mismatch, validation fails.

D.1.2.1 XML Security Uniform Resource Identifiers (URIs)

XML security algorithm identifiers have been defined in a number of different specifications such as XML Signature, XML Encryption and RFCs. XML Security Algorithm Cross-Reference (Reference [56]) provides a non-normative list of identifiers that have been defined by XML Signature (References [15] and [57]), XML Encryption (References [58] and [59]) and Additional XML Security Uniform Resource Identifiers (URIs, Reference [54]).

This Appendix profiles the use of those algorithm identifiers listed in Reference [56] specifying whether support for that algorithm is mandatory, optional or prohibited for signature generation.

Mandatory and optional algorithms on signature generation MUST be supported on signature validation.

Prohibited algorithms on signature generation MAY be supported on signature validation.

D.1.2.2 Core Signature Syntax

This section refers to XMLDSIG Section 4.

D.1.2.2.1 Signature

This section refers to XMLDSIG Section 4.1.

In the case where a cryptographic binding is required the *bindingInformation* element (specified in appendix 1) MUST contain at least one *Signature* element.

D.1.2.2.2 SignatureValue

This section refers to XMLDSIG Section 4.2.

D.1.2.2.3 SignedInfo

This section refers to XMLDSIG Section 4.3.

D.1.2.2.4 CanonicalizationMethod

This section refers to XMLDSIG Section 4.3.1.

The `CanonicalizationMethod Algorithm` attribute MUST be one of the following:

- http://www.w3.org/TR/2001/REC-xml-c14n-20010315
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
- http://www.w3.org/2006/12/xml-c14n11
- http://www.w3.org/2006/12/xml-c14n11#WithComments
- http://www.w3.org/2001/10/xml-exc-c14n#
- http://www.w3.org/2001/10/xml-exc-c14n#WithComments
- http://www.w3.org/2010/10/xml-c14n2.

D.1.2.2.5 SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute is REQUIRED.

The value of the `SignatureMethod Algorithm` is further specified depending on the cryptographic technique and mechanism being used (refer to Appendix D.2 for Digital Signatures or Appendix D.3 for HMAC).

D.1.2.2.6 Reference

This section refers to XMLDSIG Section 4.3.3.

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element there MUST be a `Reference` element

In the use case identified in Same-Document URI-References there MUST be a `Reference` element that identifies the `Manifest` element.

For each *MetadataBinding* element included in the *bindingInformation* element there MUST be a `Reference` element that identifies each *MetadataBinding* element.

D.1.2.2.6.1 URI

This section refers to XMLDSIG Section 4.3.3.1.

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element that contains a *URI* attribute with a value there MUST be a `Reference` element with the same `URI` attribute value, except in the case identified in Same-Document URI-References.

In the case identified in Same-Document URI-References there MUST be a `URI` attribute present with the value referencing the *Manifest* element.

For each *MetadataBinding* element included in the *bindingInformation* element there MUST be a `Reference URI` attribute with a shortname XPointer (Reference [43]) as the attribute value that identifies each *MetadataBinding* element.

D.1.2.2.6.2 Transforms

This section refers to XMLDSIG Section 4.3.3.4.

For Embedded BDOs an Enveloped Binding Data Object transform MUST first be applied to remove the *BindingInformation* element from the digest calculation of the `Reference` element containing the *BindingInformation* element. The Enveloped Binding Data Object transform element MUST have `Transform Algorithm` attribute value of *http://www.w3.org/TR/1999/REC-xpath-19991116* and MUST contain the following XPath element:

```
<XPath>
   not(ancestor-or-self::*[local-name() = ‘BindingInformation’ and
   namespace-uri() = ‘urn:nato:stanag:4778:bindinginformation:1:0’
</XPath>
```

For each *DataReference* or *MetadataReference* element included in a *bindingInformation* element that contains a *Transforms* element the first (or next in the case of Embedded BDOs) `Transform` element of the `Reference Transforms` element MUST be the *Transform* element from the *DataReference* or *MetadataReference* element.

For each *MetadataBinding* element included in the *bindingInformation* element there MAY be a `Transform` element (child of the `Transforms` element) that includes an XPath (Reference [42]) expression to identify *MetadataBinding* element.

For each *MetadataBinding*, *DataReference*, and *MetadataReference* that is identified by an XPath expression the `Transform` element MUST have an `Algorithm` attribute with the value '*http://www.w3.org/TR/1999/REC-xpath-19991116*'.

Other `Transform` elements MAY be present.

For other `Transform` elements the `Transform Algorithm` attribute MUST have one of the following values:

- http://www.w3.org/2000/09/xmldsig#base64
- http://www.w3.org/TR/1999/REC-xpath-19991116
- http://www.w3.org/2002/06/xmldsig-filter2
- http://www.w3.org/2000/09/xmldsig#enveloped-signature
- http://www.w3.org/TR/1999/REC-xslt-19991116
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315
- http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
- http://www.w3.org/2006/12/xml-c14n11
- http://www.w3.org/2006/12/xml-c14n11#WithComments
- http://www.w3.org/2001/10/xml-exc-c14n#
- http://www.w3.org/2001/10/xml-exc-c14n#WithComments
- http://www.w3.org/2010/10/xml-c14n2.

D.1.2.2.6.3 DigestMethod

This section refers to XMLDSIG Section 4.3.3.5.

The `DigestMethod Algorithm` attribute MUST conform to the specifications detailed in the table below.

Table 13: DigestMethod Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/ Prohibited
http://www.w3.org/2001/04/xmldsig-more#md5	Prohibited
http://www.w3.org/2000/09/xmldsig#sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#sha224	Optional
http://www.w3.org/2001/04/xmlenc#sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#sha384	Optional
http://www.w3.org/2001/04/xmlenc#sha512	Optional
http://www.w3.org/2001/04/xmlenc#ripemd160	Optional

D.1.2.2.6.4 DigestValue

This section refers to XMLDSIG Section 4.3.3.6.

D.1.2.2.7 KeyInfo

This section refers to XMLDSIG Section 4.4.

The KeyInfo element is REQUIRED.

Refer to the relevant section, dependent upon the cryptographic technique and mechanism being used (refer to Appendix D.2 for Digital Signatures or Appendix D.3 for HMAC), for further profiling of the KeyInfo element.

D.1.2.2.8 Object

This section refers to XMLDSIG Section 4.5.

The Object element is REQUIRED.

D.1.2.3 Additional Signature Syntax

This section refers to XMLDSIG Section 5.

D.1.2.3.1 Manifest

This section refers to XMLDSIG Section 5.1.

The Manifest element is REQUIRED only to support the use case for Same-Document URI-References.

The Originator MUST obtain the data object to be digested by dereferencing the *URI* attribute value in the *MetadataReference* or *DataReference* element in accordance to the semantics specified for the URI fragment identifier defined by the media type (identified in the *MetadataReference contentType* or *DataReference contentType* attribute value).

The Originator MUST perform the processing rules for Reference Generation as specified in XMLDSIG Section 3.1.1 with the following constraint:

The Reference element URI attribute value MUST be the same value as the *DataReference* (or *MetadataReference*) *URI* attribute value.

In other cases the use of the Manifest element is NOT REQUIRED.

In the case where the use of the Manifest element is required it is RECOMMENDED that the originator create a Reference element, including the identification of the Manifest element, any transform elements, the digest algorithm and the DigestValue in order to be included in the signature

D.1.2.3.2 SignatureProperties

This section refers to XMLDSIG Section 5.2.

D.1.2.3.3 TimeStamp

This section refers to Web Services Security (WSS) (Reference [41]) Section 10.

The *TimeStamp* element MUST be present indicating the time that the cryptographic binding was created as a value of the *Created* element.

The *ValueType* attribute of the *Created* element MUST be *xsd:dateTime*.

The `Expires` element (child element of the `TimeStamp` element) is NOT REQUIRED.

The inclusion of an indication when the cryptographic binding was created supports the following two use cases:

1) Detection of replay attacks; and,
2) A valid cryptographic binding at time of signing, however, the key material used for creating the signature may have expired, been revoked or other.

It is RECOMMENDED that the originator create a `Reference` element, including the identification of the `TimeStamp` element in order to be included in the signature.

D.2 Digital Signature Cryptographic Artefact

Implementations that use digital signatures as the cryptographic mechanism for producing cryptographic artefacts are REQUIRED to be conformant with Appendix D.1 and this section.

D.2.1 SignedInfo

This section refers to XMLDSIG Section 4.3.

D.2.1.1 SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute MUST conform to the specifications detailed in the table below.

Table 14: SignatureMethod (PKI) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmldsig#dsa-sha1	Prohibited
http://www.w3.org/2009/xmldsig11#dsa-sha256	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-md5	Prohibited
http://www.w3.org/2000/09/xmldsig#rsa-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#rsa-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#rsa-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-sha512	Optional
http://www.w3.org/2001/04/xmldsig-more#rsa-ripemd160	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512	Optional

D.2.1.2 KeyInfo

This section refers to XMLDSIG Section 4.4.

The `KeyInfo` element is REQUIRED.

D.2.1.2.1 KeyName

This section refers to XMLDSIG Section 4.4.1.

The `KeyName` element SHALL NOT be present.

D.2.1.2.2 KeyValue

This section refers to XMLDSIG Section 4.4.2.

The `KeyValue` MAY be present.

D.2.1.2.3 RetrievalMethod

This section refers to XMLDSIG Section 4.4.3.

The `RetrievalMethod` SHALL NOT be present.

D.2.1.2.4 X509Data

This section refers to XMLDSIG Section 4.4.4.

The `X509Data` element is REQUIRED.

In strategic systems with high throughput, certificates MUST be included.

X.509 version 3 certificates MUST be supported.

The certificate profile specified in Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (Reference [60]) MUST be supported.

The Originator SHOULD include at least one chain of certificates up to, but not including, a Certificate Authority (CA) that it believes that the Recipient may trust as authoritative.

Each certificate MUST be included in an `X509Certificate` element.

The Recipient SHOULD be able to handle an arbitrarily large number of certificates and chains.

In those cases where certificates may not be transmitted one of the `X509IssuerSerial`, `X509SKI` and `X509SubjectName` elements MUST be present.

The `X509CRL` element is NOT REQUIRED.

The CRL SHOULD be looked up based on the CRL Distribution Point (CDP) contained in the certificate.

The CRL profile specified in Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (Reference [60]) MUST be supported.

D.2.1.2.5 PGPData

This section refers to XMLDSIG Section 4.4.5.

The `PGPData` element SHALL NOT be present.

D.2.1.2.6 SPKIData

This section refers to XMLDSIG Section 4.4.6.

The `SPKIData` element SHALL NOT be present.

D.2.1.2.7 MgmtData

This section refers to XMLDSIG Section 4.4.7.

The `MgmtData` element SHALL NOT be present.

D.3 Keyed-Hash Message Authentication Code Cryptographic Artefact

Implementations that use keyed-hash message authentication codes (Reference [51]) as the cryptographic mechanism for producing cryptographic artefacts are REQUIRED to be conformant with Appendix D.1 and this section.

D.3.1 SignedInfo

This section refers to XMLDSIG Section 4.3.

D.3.1.1 SignatureMethod

This section refers to XMLDSIG Section 4.3.2.

The `SignatureMethod Algorithm` attribute MUST conform to the specifications detailed in the table below.

Table 15: SignatureMethod (HMAC) Algorithm Identifiers

Algorithm Identifier	Mandatory/Optional/Prohibited
http://www.w3.org/2000/09/xmldsig#hmac-sha1	Prohibited
http://www.w3.org/2001/04/xmldsig-more#hmac-sha224	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-sha256	Mandatory
http://www.w3.org/2001/04/xmldsig-more#hmac-sha384	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-sha512	Optional
http://www.w3.org/2001/04/xmldsig-more#hmac-ripemd160	Optional

In the case whereby the `HMACOutputLength` is used for HMAC algorithms the errata to XMLDSIG (Reference [55]) MUST be followed.

D.3.1.2 KeyInfo

This section refers to XMLDSIG Section 4.4.

The `KeyInfo` element is REQUIRED.

D.3.1.2.1 KeyName

This section refers to XMLDSIG Section 4.4.1.

The `KeyName` element MAY be present.

D.3.1.2.2 KeyValue

This section refers to XMLDSIG Section 4.4.2.

The `KeyValue` SHALL NOT be present.

D.3.1.2.3 RetrievalMethod

This section refers to XMLDSIG Section 4.4.3.

The `RetrievalMethod` SHALL NOT be present.

D.3.1.2.4 X509Data

This section refers to XMLDSIG Section 4.4.4.

The `X509Data` SHALL NOT be present.

D.3.1.2.5 PGPData

This section refers to XMLDSIG Section 4.4.5.

The `PGPData` element SHALL NOT be present.

D.3.1.2.6 SPKIData

This section refers to XMLDSIG Section 4.4.6.

The `SPKIData` element SHALL NOT be present.

D.3.1.2.7 MgmtData

This section refers to XMLDSIG Section 4.4.7.

The `MgmtData` element SHALL NOT be present.

D.4 Examples

This section contains fictitious examples that illustrate cryptographic Binding Data Objects (BDOs) that contain cryptographic artefacts conformant with Annex D. All examples given in this Annex use Confidentiality Metadata Labels (Reference [9]) as example metadata.

Encapsulating Cryptographic BDO Containing an Enveloped Signature with a Keyed-Hash Message Authentication Code Cryptographic Artefact

```
<mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <Signature Id="id-a99fac99-513d-4b08-8158-ef862e4d9f80"
xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"
/>
      <Reference URI="#id-66bb29e1-9696-4ea0-be3c-f7d0096a0d81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>9JBAVs2gUWUzFh8uUl1ubXW13VgQxli3NM+CF0vQG14=</DigestValue>
      </Reference>
      <Reference URI="#id-d55d0123-babc-467f-b309-62e95291a9e4">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>8G8AHBPiAJ+W6PUOq+W/Vua+iO7Zj6GzooPRmkqtqnY=</DigestValue>
      </Reference>
      <Reference URI="#id-b3eaf318-700f-4740-b43e-2def8d98db81">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>Kx02/WnFE/2MN7lEuWemAiDetsJZ+8lJt4nvg4GyRNc=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>g3nzbBiu7msmVHfCjmVqqSiimlASoBSM/hxqFN7YxH0=</SignatureValue>
    <KeyInfo Id="id-b3eaf318-700f-4740-b43e-2def8d98db81">
      <KeyName>HMAC_SECRET_KEY</KeyName>
    </KeyInfo>
    <Object Id="id-17250b2d-f0f5-4457-9e21-23db31e3460d">
      <SignatureProperties Id="id-d55d0123-babc-467f-b309-62e95291a9e4">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T15:58:44Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
      </SignatureProperties>
    </Object>
  </Signature>
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding Id="#id-66bb29e1-9696-4ea0-be3c-f7d0096a0d81">
      <mb:Metadata>
        <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
            <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
            <slab:Classification>UNCLASSIFIED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:Data>
        <Document xmlns="">
          <Title>BDO Examples</Title>
          <Author>alan.ross@reach.nato.int</Author>
          <Abstract>
           Example XML File to support illustration of different types of BDO and
cryptographic artefacts
          </Abstract>
          <Introduction>....</Introduction>
          <Chapter Id="chapter-1">
            <Paragraph Id="para-1-1" />
            <Paragraph Id="para-1-2" />
          </Chapter>
          <Chapter Id="chapter-2">
            <Paragraph Id="para-2-1" />
            <Paragraph Id="para-2-2" />
          </Chapter>
        </Document>
      </mb:Data>
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Encapsulating Cryptographic BDO Containing an Enveloped Signature with a Digital Signature Cryptographic Artefact

```
<mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
  <Signature Id="id-fb00da79-4b32-4fcc-a302-4dbf789212e3"
xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
      <Reference URI="#id-20c07ca8-6960-4a36-bdd1-e3cb299f82c3">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>fAXcjRa4zlLyB+lchyBK/9JzlsoZSbxNCmr/27nA9aI=</DigestValue>
      </Reference>
      <Reference URI="#id-82744679-a547-40aa-a683-cf97619054fe">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>j5AgAamc6cv54VDzl0kDlQ4wYZLLAU3761eFOUWvtX0=</DigestValue>
      </Reference>
      <Reference URI="#id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <DigestValue>hWUoi0gFxnFsGnHJO/V2eNg/silda814PSP2/WlsqtU=</DigestValue>
      </Reference>
    </SignedInfo>

<SignatureValue>gItAuwdEykw5xDht50TOei1xfT0q7KLaUXm4w/2rnpTjoxiODTI3Wr8D4fmx/404bVrX23S
tY6HHT/dxDPcgODa+K9YL/pl3y8RvIrfWGhiZReY5AUj1EF3mxI22ari/ao0shKe18aPJ0J2RmGH3t30qrHfvUX
cIcREIOT1S6GajpNCOJPYoa9yb400MOx0oRHXkFegnQ5eXeSBIh2u4DhwL0I4GSeuYA9FVt8qyvla9EnTTS6fG2
+gLjd6YEQzfIBvVtrY5b9WnhqqiHy5tyepZgVtMSEXrukWrNELpvwC467KR+MincgUA9RlsAEvCBaR4oQKTUOxB
Q5tD+N/FzQ==</SignatureValue>
    <KeyInfo Id="id-9920a48c-c3a1-45d0-a81c-1ce04d1d8de6">
      <X509Data>
              <X509Certificate>MIIDM……wIBAgIJAI29/+A/MN7RPAx5eOKQg==</X509Certificate>
      </X509Data>
    </KeyInfo>
    <Object Id="id-63fc02c0-10b6-49fd-9759-7bfb1d52ecf7">
      <SignatureProperties Id="id-82744679-a547-40aa-a683-cf97619054fe">
        <SignatureProperty>
          <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
            <wsu:Created>2015-11-13T16:01:38Z</wsu:Created>
          </wsu:TimeStamp>
        </SignatureProperty>
      </SignatureProperties>
    </Object>
  </Signature>
  <mb:MetadataBindingContainer>
    <mb:MetadataBinding Id="#id-20c07ca8-6960-4a36-bdd1-e3cb299f82c3">
      <mb:Metadata>
        <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
          <slab:ConfidentialityInformation>
            <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
            <slab:Classification>UNCLASSIFIED</slab:Classification>
          </slab:ConfidentialityInformation>
          <slab:CreationDateTime>
         2015-09-30T12:30:00Z
        </slab:CreationDateTime>
        </slab:originatorConfidentialityLabel>
      </mb:Metadata>
      <mb:Data>
        <Document xmlns="">
          <Title>BDO Examples</Title>
          <Author>alan.ross@reach.nato.int</Author>
          <Abstract>
           Example XML File to support illustration of different types of BDO and
cryptographic artefacts
          </Abstract>
          <Introduction>....</Introduction>
          <Chapter Id="chapter-1">
            <Paragraph Id="para-1-1" />
            <Paragraph Id="para-1-2" />
          </Chapter>
          <Chapter Id="chapter-2">
            <Paragraph Id="para-2-1" />
            <Paragraph Id="para-2-2" />
          </Chapter>
        </Document>
      </mb:Data>
    </mb:MetadataBinding>
  </mb:MetadataBindingContainer>
</mb:BindingInformation>
```

Embedded Cryptographic BDO Containing an Enveloped Signature with a Keyed-Hash Message Authentication Code Cryptographic Artefact

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO and cryptographic
artefacts
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
  <mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
    <Signature Id="id-134ce280-1682-4963-b868-6621b480ce26"
xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"
/>
        <Reference URI="">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <XPath>not(ancestor-or-self::*[local-name() = 'BindingInformation' and
namespace-uri() = 'http://www.nato.int/2014/06/nl/mb'])</XPath>
```

```
            </Transform>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>RYxJZ8BN/MR2D0BDxiCxGSDaQvGFKQ86udb0Ov5A2s4=</DigestValue>
        </Reference>
        <Reference URI="#id-c9e4f1c8-ad34-4d4d-9909-827570de41a2">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>WKOWdda84YLuSqbaZsS8LQ6kqF6HR0dfC+iz/e+KPf0=</DigestValue>
        </Reference>
        <Reference URI="#id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>UbMTebL9lKFARnG1qWOpQ1DiuCFPzs6W1hse9gPOxUk=</DigestValue>
        </Reference>
        <Reference URI="#id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>z7+6QZiSTqYMHCiy9o3uxGfA8q5ScEeHlHZs3w9+8S4=</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>dk7Ds4Atik6yF/wKZjOIDVGGyv1rigTDLj6gRsqCTHY=</SignatureValue>
      <KeyInfo Id="id-001c1a07-74c4-4815-aecd-dd1bcba8bc9c">
        <KeyName>HMAC_SECRET_KEY</KeyName>
      </KeyInfo>
      <Object Id="id-4dcc6c48-6ed0-4cf0-b386-b85f7ee0c826">
        <SignatureProperties Id="id-1bd95780-277f-44b3-99fd-b2b69505ae5a">
          <SignatureProperty>
            <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
              <wsu:Created>2015-11-13T16:07:37Z</wsu:Created>
            </wsu:TimeStamp>
          </SignatureProperty>
        </SignatureProperties>
      </Object>
    </Signature>
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding Id="#id-c9e4f1c8-ad34-4d4d-9909-827570de41a2">
        <mb:Metadata>
          <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
              <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
          <slab:alternateConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
              <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
          </slab:alternateConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI="" />
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>
</Document>
```

Embedded Cryptographic BDO Containing a Detached Signature with a Digital Signature Cryptographic Artefact

```
<Document xmlns="http://example.com/doc">
  <Title>BDO Examples</Title>
  <Author>alan.ross@reach.nato.int</Author>
  <Abstract>
  Example XML File to support illustration of different types of BDO and cryptographic
artefacts
  </Abstract>
  <Introduction>....</Introduction>
  <Chapter Id="chapter-1">
    <Paragraph Id="para-1-1" />
    <Paragraph Id="para-1-2" />
  </Chapter>
  <Chapter Id="chapter-2">
    <Paragraph Id="para-2-1" />
    <Paragraph Id="para-2-2" />
  </Chapter>
  <mb:BindingInformation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:mb="urn:nato:stanag:4778:bindinginformation:1:0">
    <Signature Id="id-3a7079e1-adeb-47b0-a4df-86a5f2962f57"
xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
/>
        <Reference URI="#para-2-2">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>0JsT5SNKuCYoe91tl8n590Hcy/UivrId3Zf6kJy7pdg=</DigestValue>
        </Reference>
        <Reference URI="#id-b073db91-a8b3-4905-809d-82e92b0d0ecc">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>a3yUgG8j0eIPI6ZSw7aw4JPHO1SBglS0+Fb7lwVmMeo=</DigestValue>
        </Reference>
        <Reference URI="#id-7d5d0648-59c1-48a9-a3bc-a07a24f0a67b">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>zMHgHTwG+OtqPY8+T4cwYGby2UoSv71QJ2eU0peB5ds=</DigestValue>
        </Reference>
        <Reference URI="#id-45f67abd-5803-4933-acb8-5061adde54f4">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <DigestValue>g8jESHIgXr4bGZFwOzh2O4r8Vv0y6jfH7qKgQTGV9ww=</DigestValue>
        </Reference>
      </SignedInfo>

<SignatureValue>C1yPwzpU/ngO42sXo2HHZTtbXNTe2FAXf2RivMy5u6z/xoNlmi/mHm5ejZPFWkoGaUmWDad
REcc5lI6XBYXeks2YVyMh05uDRCQLPYNkIAx3BpUFH7y9JUklj4WvlDBeZ2GwNhp463QMvn8pF35cXw1f86Vc0M
3CtAm5MNbnS6BqqswdygCF/HivjHcQSnYGRhI4vegelwfYyhFRHQQ1OE3ytUDR8VLKZfgyK3M6mcQjvlHtL2qjR
xMHrkQQtt8oBQk6iAWxYgbqeIzqw3cIYL5jb/ML2UOycGgwUIqGFx95EouKuOMZSN8e2dnaVaHp26XlzpdJkyTk
Vr5/T7v3hA==</SignatureValue>
      <KeyInfo Id="id-45f67abd-5803-4933-acb8-5061adde54f4">
        <X509Data>
              <X509Certificate> MIIDM……wIBAgIJAI29/+A/MN7RPAx5eOKQg==</X509Certificate>
        </X509Data>
      </KeyInfo>
      <Object Id="id-221fefa8-fd81-4f98-8784-ac4a08e4eece">
        <SignatureProperties Id="id-7d5d0648-59c1-48a9-a3bc-a07a24f0a67b">
          <SignatureProperty>
            <wsu:TimeStamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
              <wsu:Created>2015-11-13T16:04:59Z</wsu:Created>
            </wsu:TimeStamp>
          </SignatureProperty>
        </SignatureProperties>
      </Object>
    </Signature>
    <mb:MetadataBindingContainer>
      <mb:MetadataBinding Id="#id-b073db91-a8b3-4905-809d-82e92b0d0ecc">
        <mb:Metadata>
          <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
              <slab:Classification>UNCLASSIFIED</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI="" />
      </mb:MetadataBinding>
      <mb:MetadataBinding>
        <mb:Metadata>
          <slab:originatorConfidentialityLabel
xmlns:slab="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0">
            <slab:ConfidentialityInformation>
              <slab:PolicyIdentifier>ACME</slab:PolicyIdentifier>
              <slab:Classification>CONFIDENTIAL</slab:Classification>
            </slab:ConfidentialityInformation>
            <slab:CreationDateTime>
       2015-09-30T12:30:00Z
      </slab:CreationDateTime>
          </slab:originatorConfidentialityLabel>
        </mb:Metadata>
        <mb:DataReference URI="#para-2-1" />
      </mb:MetadataBinding>
    </mb:MetadataBindingContainer>
  </mb:BindingInformation>
</Document>
```

Annex E : CWIX 2016 Validation

The table below provides the results from validation exercise at CWIX 2016 where each of the profiles presented in Appendix C and Appendix D were tested with national implementations. The hyperlinks provided point directly to the test results stored on the CWIX 2016 portal.

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
Test Case 01783@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016			FIN-LION-FMN 2@CWIX 2016
Test Case 02654@CWIX 2016	USA-TIES CWP@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016 NATO-SGA@CWIX 2016			
Test Case 03311@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016	USA-TIES CWP@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 NATO-SGA@CWIX 2016			
Test Case 03314@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016			
Test Case 03319@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016 USA-TIES CWP@CWIX 2016 USA-USMC CIG@CWIX 2016	USA-TIES CWP@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016 USA-USMC CIG@CWIX 2016			
Test Case 03325@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016 FIN-LION-FMN 2@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016 FIN-LION-FMN 2@CWIX 2016			

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
	CAN-DCA LE@CWIX 2016 USA-USMC CIG@CWIX 2016	CAN-DCA LE@CWIX 2016 USA-USMC CIG@CWIX 2016			
Test Case 03336@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016		
Test Case 03344@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016			
Test Case 03352@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016			
Test Case 03752@CWIX 2016	NATO-FFTS Verification@CWIX 2016	POL-AFCCS TOPAZ@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016		
Test Case 03753@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016		
Test Case 03754@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016		
Test Case 04053@CWIX 2016	USA-GCCS-J@CWIX 2016	POL-AFCCS TOPAZ@CWIX 2016			
Test Case 04142@CWIX 2016	NATO-FFTS Verification@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CW	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016	SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	IX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04203@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016	NATO-SOA Platform@CWIX 2016			
Test Case 04262@CWIX 2016	PRT-NFFI GW@CWIX 2016	NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016			
Test Case 04263@CWIX 2016	PRT-NFFI GW@CWIX 2016	NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016	
Test Case 04281@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016		

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
Test Case 04282@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-FFTS Verification@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016		
Test Case 04284@CWIX 2016	POL-HMS HEKTOR@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 NATO-FFTS Verification@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04306@CWIX 2016	POL-HMS JASMINE@CWIX 2016	NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		1.0@CWIX 2016 PRT-NFFI GW@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016		Binding@CWIX 2016	
Test Case 04307@CWIX 2016	POL-HMS JASMINE@CWIX 2016	NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 PRT-NFFI GW@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04330@CWIX 2016	ITA-EI BFT@CWIX 2016	POL-HMS HEKTOR@CWIX 2016 ROU-FMN 1.0@CWIX 2016 CZE-FFT	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		Techtest@CWIX 2016 POL-HMS JASMINE@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 PRT-NFFI GW@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016	SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04331@CWIX 2016	NLD-ELIAS FFT@CWIX 2016	NATO-FFTS Verification@CWIX 2016 PRT-NFFI GW@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		HEKTOR@CWIX 2016			
Test Case 04333@CWIX 2016	NLD-ELIAS FFT@CWIX 2016	NATO-FFTS Verification@CWIX 2016 PRT-NFFI GW@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04341@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016		Binding@CWIX 2016	
Test Case 04342@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 NATO-FFTS Verification@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04347@CWIX 2016	ITA-NAVY HELIOS@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016	2016, USA-FTAMS@CWIX 2016	OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04352@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016	NATO-SOA Platform@CWIX 2016 CZE-SOA@CWIX 2016			
Test Case 04418@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-Cross Domain IEG@CWIX 2016		
Test Case 04435@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-SOA Platform@CWIX 2016		
Test Case 04437@CWIX 2016	NATO-SOA Platform@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016			
Test Case 04438@CWIX 2016	NATO-SOA Platform@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016			
Test Case 04443@CWIX 2016	CZE-FFT Techtest@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CW	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 NATO-FFTS Verification@CWIX 2016	2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	IX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04460@CWIX 2016	POL-HMS HEKTOR@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
Test Case 04469@CWIX 2016	CZE-FFT Techtest@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 NATO-FFTS Verification@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04471@CWIX 2016	CZE-FFT Techtest@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 NATO-FFTS	DEU-RuDi OpenCOP@CWIX 2016, ITA-EI BFT@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
		Verification@CWIX 2016			
Test Case 04482@CWIX 2016	ITA-EI BFT@CWIX 2016	PRT-NFFI GW@CWIX 2016 POL-HMS HEKTOR@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 ROU-FMN 1.0@CWIX 2016 CZE-FFT Techtest@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 POL-HMS JASMINE@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04527@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	CAN-DCA LE@CWIX 2016			
Test Case 04593@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	NATO-SOA Platform@CWIX 2016			
Test Case 04595@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	NATO-SOA Platform@CWIX 2016			
Test Case 04596@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	CZE-SOA@CWIX 2016			

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
Test Case 04604@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-NCIA-JCHAT@CWIX 2016	NATO-SOA Platform@CWIX 2016		
Test Case 04629@CWIX 2016	POL-AFCCS TOPAZ@CWIX 2016	PRT-NFFI GW@CWIX 2016 NLD-ELIAS FFT@CWIX 2016 FIN-SEALION@CWIX 2016 DEU-RuDi OpenCOP@CWIX 2016 POL-AFCCS TOPAZ@CWIX 2016 ITA-EI BFT@CWIX 2016 ROU-FMN 1.0@CWIX 2016 POL-HMS JASMINE@CWIX 2016 CZE-FFT Techtest@CWIX 2016 POL-HMS HEKTOR@CWIX 2016	DEU-RuDi OpenCOP@CWIX 2016, ITA-NAVY CID SERVER@CWIX 2016, USA-FTAMS@CWIX 2016	USA-GCCS-J@CWIX 2016 NATO-FFTS Verification@CWIX 2016 NATO-NCIA-NIRIS Current OPS@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016	
Test Case 04632@CWIX 2016	NATO-Cross-Domain Binding@CWIX 2016	NATO-SOA Platform@CWIX 2016			
Test Case 04679@CWIX 2016	NATO-NCIA-JTS-4@CWIX 2016	NATO-NCIA-JTS-4@CWIX 2016	NATO-Cross Domain IEG@CWIX 2016		
Test Case 04680@CWIX 2016	NATO-NCIA-ICC@CWIX 2016	NATO-NCIA-ICC@CWIX 2016	NATO-Cross Domain IEG@CWIX 2016		

52 Test Cases	Provider	Consumer	Mediator	Distributor	Observer
Test Case 04703@CWIX 2016	FRA-JACENT@CWIX 2016	NATO-SOA Platform@CWIX 2016			
Test Case 04704@CWIX 2016	NATO-SOA Platform@CWIX 2016	FRA-JACENT@CWIX 2016			
Test Case 04723@CWIX 2016	CAN-DCA LE@CWIX 2016 NATO-Cross-Domain Binding@CWIX 2016		CAN-DCA LE@CWIX 2016, NATO-Cross-Domain Binding@CWIX 2016		
Test Case 04730@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	DEU-MISSION-PKI@CWIX 2016			
Test Case 04731@CWIX 2016	TUR-VAG-XML-GATEWAY@CWIX 2016	GBR-DMarGSX-GeoMetOc@CWIX 2016			

Annex F : References

[1] Guidance on the Marking of NATO Information, June 2011, AC/322-N(2011)0130.

[2] The Primary Directive on Information Management, 18 December 2008, C-M(2008)0113 (INV).

[3] C-M(2002)49, "Security Within the North Atlantic Treaty Organisation", 17 June 2002.

[4] C-M(2002)60, "The Management of NATO Non-Classified Information", 11 July 2002

[5] Directive on the Security of Information, 17 January 2012, AC/35-D/2002-REV4.

[6] AC/322-D(2004)0021 (INV), "INFOSEC Technical and Implementation Guidance for Electronic Labelling of NATO Information", March 2004

[7] AC/322-D(2014)0010, "NATO Core Metadata Specification (NCMS)", January 2015

[8] IETF RFC 2634, "Enhanced Security Services for S/MIME", at http://tools.ietf.org/html/rfc2634, June 1999.

[9] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium

[10] NATO RTO RTG-031/IST-068, "XML in Cross-Domain Security Solutions", draft, March 2010

[11] NCIA TN-1455-REV1, "NATO Profile for the Binding of Metadata to Data Objects", Version 1.1, December 2012

[12] NCIA TN-1456-REV1, "NATO Profile for the XML Confidentiality Label Syntax, Version 1.1, January 2013

[13] W3C SOAP Version 1.1, 2000, "Simple Object Access Protocol (SOAP 1.1", at http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, W3C Recommendation, W3C, 8 May 2000.

[14] W3C SOAP Version 1.2, 2007, "SOAP Version 1.2", at http://www.w3.org/TR/soap12-part1/, W3C Recommendation, W3C, 27 April 2007.

[15] W3C XMLDSIG-CORE, 2008, "XML- Signature Syntax and Processing (Second Edition)", at http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/, W3C Recommendation, W3C, 10 June 2008

[16] W3C XMLMIME, 2005, "Describing Media Content of Binary Data in XML", at http://www.w3.org/TR/xml-media-types/, W3C

[17] W3C XMLMIME, 2004, "XML Schema Part 2: Datatypes Second Edition", at http://www.w3.org/TR/xmlschema-2/, W3C Recommendation, W3C, 28 October 2004.

[18] IETF RFC 7303, “XML Media Types”, at http://tools.ietf.org/html/rfc7303, July 2014

[19] IANA Transfer Encodings, 2009, “Transfer Encodings”, at http://www.iana.org/assignments/transfer-encodings/transfer-encodings.xhtml, 17 August 2009.

[20] IANA MIME Media Types, “MIME Media Types”, at http://www.iana.org/assignments/media-types

[21] IETF RFC 3986, “Uniform Resource Identifier (URI): Generic Syntax”, at http://tools.ietf.org/html/rfc3986, January 2005.

[22] IETF RFC 5322, “Internet Message Format”, at http://tools.ietf.org/html/rfc5322, October 2008.

[23] IETF RFC 7444, “Security Labels in Internet Email”, K. Zeilenga and A. Melnikov, February 2015.

[24] IETF RFC 2392, “Content-ID and Message-ID Uniform Resource Locators”, at http://tools.ietf.org/html/rfc2392, August 1998.

[25] IETF RFC 6522, “The Multipart/Report Media Type for the Reporting of Mail System Administrative Messages”, at http://tools.ietf.org/html/rfc6522, January 2012

[26] IETF RFC 3464, “An Extensible Message Format for Delivery Status Notifications”, at http://tools.ietf.org/html/rfc3464, January 2003.

[27] IETF RFC 3798, “Message Disposition Notification”, at http://tools.ietf.org/html/rfc3798, May 2004

[28] XEP-0258, “Security Labels in XMPP”, version 1.1, at http://www.xmpp.org/extensions/xep-0258.html, April 2013

[29] XEP-0060, “Publish-Subscribe”, version 1.3, at http://www.xmpp.org/extensions/xep-0060.html, July 2010

[30] XEP-0030, “Service Discovery”, version 2.4, at http://www.xmpp.org/extensions/xep-0030.html, June 2008

[31] IETF RFC 6122, “Extensible Messaging and Presence Protocol (XMPP): Address Format”, at http://tools.ietf.org/html/rfc6122, March 2011

[32] XEP-0030, “Service Discovery”, version 2.4, at http://www.xmpp.org/extensions/xep-0030.html, June 2008

[33] ISO/IEC 29500-2 “Office Open XML File Formats - Part 2: Open Packaging Conventions”, at http://standards.iso.org/ittf/PubliclyAvailableStandards/c061796_ISO_IEC_29500-2_2012.zip, August 2012

[34] C-M(2007)0118, NATO Information Management Policy, 11 December 2007

[35] C-M(2002)0049, NATO Security Policy

[36] C-M(2009)0145, NATO Interoperability Policy, 3 December 2009

[37] C-M(2015)0003, NATO Federated Mission Networking Implementation Plan, 29 January 2015

[38] AC/322-D(2005)0053-REV2, NATO Information Management Policy and NATO Network Enabled Capability (NNEC) Strategies for Data and Technical Services, 14 Sept 2009

[39] MCM-0106-2014, NATO Federated Mission Networking Capability

[40] AC/322-D(2004)0024-REV2, 2008]: North Atlantic Council Document Annex 1 To AC/322-D(2004)0024-REV2, "NATO Public Key Infrastructure (NPKI) Certificate Policy", NATO C3 Board, Brussels, Belgium, February 2008

[41] Web Services Security (WS-Security), SOAP Message Security 1 (WS-Security 2004), OASIS Standard Specification, 1 February 2006

[42] W3C XPath 1.0, 1999, "XML Path Language (XPath) – Version 1.0", at http://www.w3.org/TR/xpath/, W3C Recommendation, W3C, 16 November 1999

[43] W3C XPointer, 2002, "XML Pointer Language (XPointer)", at http://www.w3.org/TR/xptr/, W3C Working Draft, W3C, 16 August 2002

[44] IETF RFC 7230, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", at http://tools.ietf.org/html/rfc7230, June 2014.

[45] IETF RFC 7159, "The JavaScript Object Notation (JSON) Data Interchange Format", at http://tools.ietf.org/html/rfc7159, March 2014.

[46] IETF RFC 6901, "JavaScript Object Notation (JSON) Pointer", at http://tools.ietf.org/html/rfc6901, April 2013.

[47] IETF RFC 2045, "Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies", at http://tools.ietf.org/html/rfc2045, November 1996

[48] IETF RFC 6120, "Extensible Messaging and Presence Protocol (XMPP): Core", at http://tools.ietf.org/html/rfc6120, March 2011

[49] IETF RFC 6121, "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", at http://tools.ietf.org/html/rfc6121, March 2011

[50] XEP-0314, "Security Labels in PubSub", version 0.1, at http://www.xmpp.org/extensions/xep-0314.html, July 2012

[51] IETF RFC 2104, "HMAC: Keyed-Hashing for Message Authentication", at http://tools.ietf.org/html/rfc2104, February 1997

[52] IETF RFC 5751, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", at http://tools.ietf.org/html/rfc5751, January 2010

[53] IETF RFC 7515, "JSON Web Signature (JWS)", at http://tools.ietf.org/html/rfc7515, May 2015

[54] IETF RFC 6931, "Additional XML Security Uniform Resource Identifiers (URIs)", at http://tools.ietf.org/html/rfc6931, April 2013

[55] W3C XMLDSIG-2nd-Ed Errata, 2014, "Errata for XML Signature 2nd Edition", at http://www.w3.org/2008/06/xmldsigcore-errata.html, W3C Recommendation, W3C, 01 October 2014

[56] W3C XMLSEC, 2013, "XML Security Algorithm Cross-Reference", at http://www.w3.org/TR/xmlsec-algorithms, W3C Working Group Note, W3C, 11 April 2013.

[57] W3C XMLDSIG-CORE1, 2013, "XML Signature Syntax and Processing Version 1.1", at http://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/, W3C Recommendation, W3C, 11 April 2013

[58] W3C XMLENC-CORE, 2002, "XML Encryption Syntax and Processing", at http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, W3C Recommendation, W3C, 10 December 2002.

[59] W3C XMLENC-CORE1, 2013, "XML Encryption Syntax and Processing Version 1.1", at http://www.w3.org/TR/2013/REC-xmlenc-core1-20130411/, W3C Recommendation, W3C, 11 April 2013.

[60] IETF RFC 5280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", at http://tools.ietf.org/html/rfc5280, May 2008

[61] IETF RFC 2231, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", at http://tools.ietf.org/html/rfc2231, November 1997.

[62] AC/322(CP/1)WP(2014)0002 and AC/322(CP/4)WP(2014)0001, "Technical Standard for Confidentiality Labelling of NATO Information", July 2014

[63] STANAG 4774: Confidentiality Metadata Label Syntax, submitted for ratification, May 2016

[64] STANAG 4778: Metadata Binding Mechanism, draft, May 2016