Extensible Message and Presence Protocol (XMPP) Binding Profile

1. Introduction

The term labelling is the process of determining the appropriate metadata for a given data object, creating the metadata label and binding the metadata label to the data object. A binding is a relationship between a data object and a metadata label. A binding is realized by applying a binding mechanism. If a metadata label must be bound to a data object, both the metadata label and the data object are input to the binding mechanism. The output of the binding mechanism is the binding of a data object and metadata label (see **Figure 1**) which says that the data object and the metadata label belong together. The binding can be recorded as a structured data object, known as a Binding Data Object (BDO).

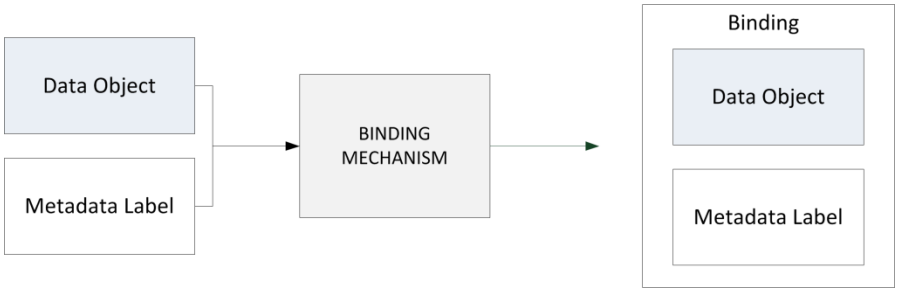

Figure 1 Creation of a binding

STANAG 4778 (Reference [3]) standardizes the binding of a data object and metadata label by specifying a common binding mechanism and a syntax for representing the BDO. However, to support information management and information sharing requirements it is necessary to further profile the application of STANAG 4778 to facilitate locating a BDO and embedding a BDO in data objects.

2. XMPP Introduction

Confidentiality metadata labels can be supported in XMPP stanzas as indicated by XEP-0258 (Reference [4]) whereby a mechanism for carrying Enhanced Security Services (ESS) Security labels (Reference [1]) is standardized. This profile extends the XEP-0258 (Reference [4]) 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 [5]).

3. Identification

The profile for XMPP is uniquely identified by the Canonical Identifier shown in Table 1.

Table 1: Profile Identifiers

Type	Identifier
Canonical Identifier	urn:nato:stanag:4778:profile:xmpp
Version Identifier	urn:nato:stanag:4778:profile:xmpp:1:0

It is recognized that this profile may evolve during its review cycle. For example, a review might identify:

- changes to the base XMPP standard
- improvements to the existing profiles based upon operational feedback

Therefore this version of the profile is uniquely identified by the Version Identifier shown in Table 1.

Subsequent versions of this profile will maintain the same Canonical Identifier, but define a new Version Identifier.

4. Standards

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

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

[3] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium

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

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

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

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

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

5. 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 Reference [3].
- `Courier font` indicates syntax derived from XMPP (References [7], [8] and [6]) and XEP-0258 (Reference [4]).

6. Message Stanza Structure

The `Message` stanza structure is specified in (Reference [8]). 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 2 illustrates the high-level structure of a `Message` stanza that contains an Embedded BDO contained within a XEP-0258 `securitylabel` element.

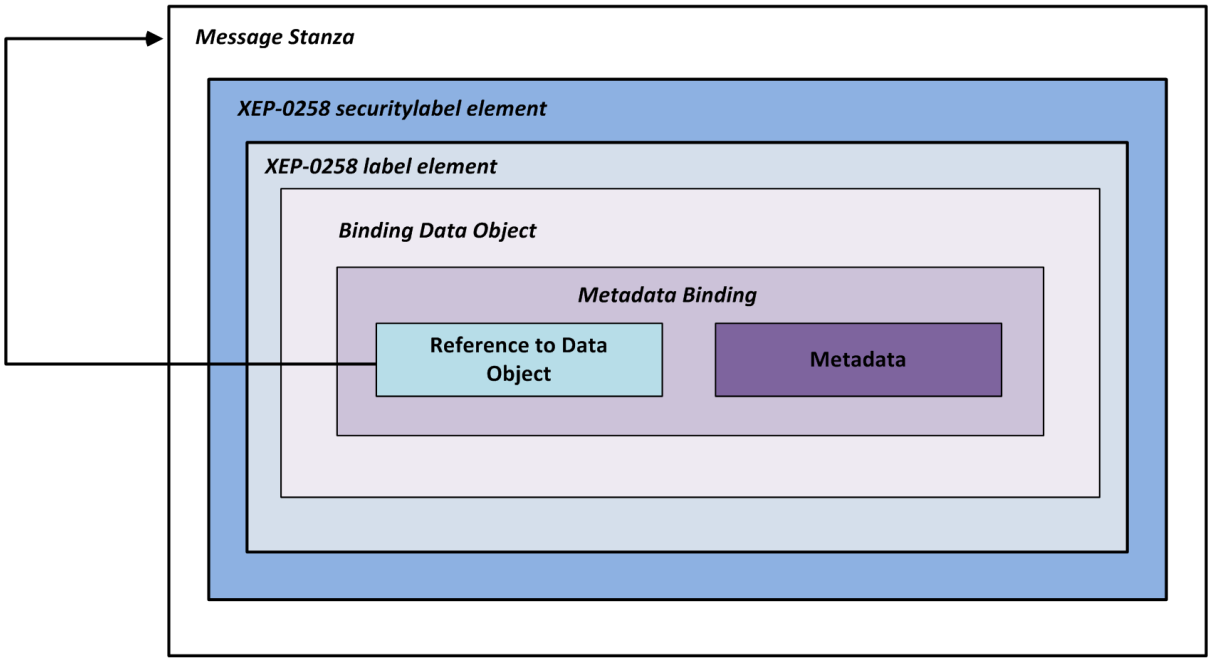

Figure 2: Structure of Message Stanza Containing Embedded BDO

The BDO SHALL be 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 3**. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

Figure 3: 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 4**. This example illustrates the use of XPaths for referencing the `body` element. This example uses Confidentiality Metadata Labels (Reference [2]) as example metadata.

Note in **Figure 4** the *namespace-uri* attribute value is set as `jabber:client`. In XMPP, `stanzas` may belong to different XMPP `content namespaces` i.e. `jabber:client` and `jabber:server` depending on whether the XMPP `stream` is negotiated between an XMPP `client` and XMPP `server` or an XMPP `server` and a peer XMPP `server`, respectively. The only difference between the two is that the `to` and `from` attributes are optional on `stanzas` qualified by the `jabber:client` namespace and required on `stanzas` qualified by the `jabber:server` namespace. To accommodate the re-scoping of XMPP content namespaces as described above the following rules apply:

1) If the XMPP Binding Profile is supported only between XMPP `peer servers` the *namespace-uri* attribute value SHALL be `jabber:server`; otherwise,
2) The default *namespace-uri* attribute value SHALL be `jabber-client`.

Figure 4 Example Detached Binding Data Object Contained in Message Stanza (XMPP)