# Simple Object Access Protocol (SOAP) 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).

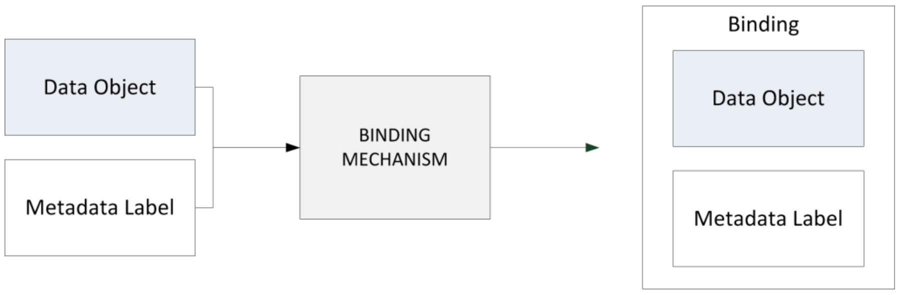

**Figure 1 Creation of a binding**

STANAG 4778 (Reference [2]) 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. 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 [3]) and SOAP 1.2 (Reference [4]). To support information sharing between partners it may be necessary to locate a Binding Data Object (BDO) in the SOAP protocol layer. Metadata may be bound to the whole data object (SOAP message) or may be bound to subsets of the SOAP message (data object(s) in the SOAP body). Where there is a requirement to bind metadata to a SOAP message or data object (s) within the SOAP body that is exchanged between a service consumer and a service provider, the SOAP Binding Profile specified must be adhered to.

## 3. Identification

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

**Table 1: Profile Identifiers**

| Type | Identifier |
|---|---|
| Canonical Identifier | urn:nato:stanag:4778:profile:soap |
| Version Identifier | urn:nato:stanag:4778:profile:soap:1:0 |

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

- changes to the base SOAP 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] STANAG 4774, Confidentiality Metadata Label Syntax, Brussels, Belgium
[2] STANAG 4778, Metadata Binding Mechanism, Brussels, Belgium
[3] 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.
[4] W3C SOAP Version 1.2, 2007, "SOAP Version 1.2", at http://www.w3.org/TR/soap12-part1/, W3C Recommendation, W3C, 27 April 2007.
[5] 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

## 5. 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 2: 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 |

## 6. 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 [2].
- `Courier font` indicates syntax derived from various W3C XML Signature (Reference [5]) and SOAP (References [3], [4]) standards.

## 7. SOAP Message Structure

The SOAP message structure is specified in (References [3], [4]**Error! Reference source not found.**). 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 SHALL 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 SHALL 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 SHALL 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 2. Also illustrated is the use of the `actor` attribute to support multiple Security elements. This example uses Confidentiality Metadata Labels (Reference [1]) 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 2: 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 3**. This example uses Confidentiality Metadata Labels (Reference [1]) as example metadata.

**Figure 3** 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 3: Example Detached BDO for SOAP**