Specification

Acronym:	PEPPOL
Grant Agreement number:	224974
Title:	Pan-European Public Procurement Online

PEPPOL Transport Infrastructure BusDox Common Definitions

Version: 1.01

Authors:

Gert Sylvest (NITA/Avanade)
Jens Jakob Andersen (NITA)
Klaus Vilstrup Pedersen (DIFI)
Mikkel Hippe Brun (NITA)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Version	Date	Author	Organisation	Description
1.0	20100215	Jens Jakob Andersen	NITA	First version (pending EC approval)
1.01	20101001	Klaus Vilstrup Pedersen	DIFI	EC Approved

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright

This deliverable is released under the terms of the **Creative Commons Licence** accessed through the following link: http://creativecommons.org/licenses/by/3.0/.

In short, it is free to
Share — to copy, distribute and transmit the work
Remix — to adapt the work
Under the following conditions
Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Contributors

Organisations

DIFI (Direktoratet for forvaltning og IKT)[1], Norway, www.difi.no
NITA (IT- og Telestyrelsen)[2], Denmark, www.itst.dk
BRZ (Bundesrechenzentrum), Austria

Persons

Gert Sylvest, NITA/Avanade (editor)
Jens Jakob Andersen, NITA
Klaus Vilstrup Pedersen, DIFI
Mike Edwards, NITA/IBM
Mikkel Hippe Brun, NITA
Paul Fremantle, NITA/WSO2
Philip Helger, BRZ
Thomas Gundel, NITA/IT Crew

[1] English: Agency for Public Management and eGovernment

[2] English: National IT- and Telecom Agency

Table of Content

1 Introduction

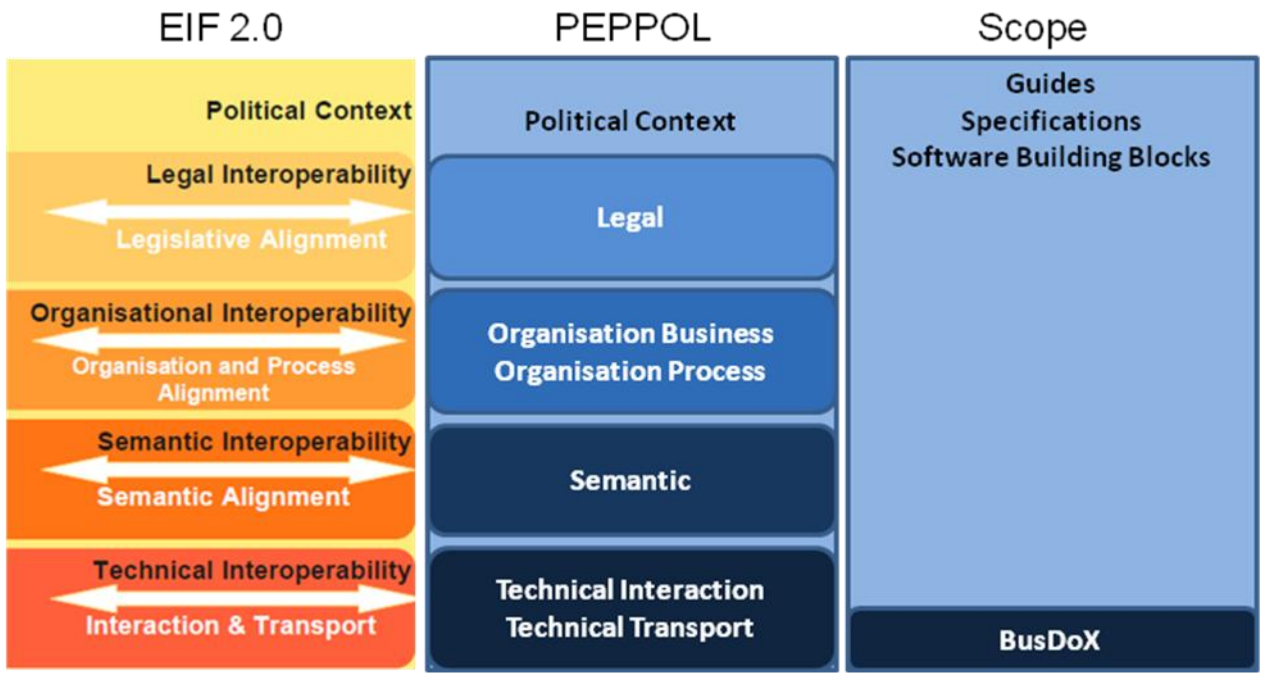
1.1 Objective

This document contains the definitions and terms that are common between the Business Document Exchange Network (BUSDOX) service metadata and transport specifications. These are:

- The START and LIME transport specifications
- The SML (Service Metadata Locator) and SMP (Service Metadata Publishing) specifications
- A scheme for process identifiers. This scheme is identified by the string "cenbii_procid_pia".

1.2 Scope

This specification relates to the Technical Transport Layer i.e. BusDox specifications. The BusDox specifications can be used in many interoperability settings. In the PEPPOL context, it provides transport for procurement documents as specified in the PEPPOL Profiles.



1.3 Goals and non-goals

The goal of this document is to describe the following:

- Define the extensible scheme format of Participant Identifiers
- Define the extensible scheme format of Document Type Identifiers
- Define the scheme format of Profile/Process Identifiers
- Define terms that are common among specifications, such as 'Access Point'
- Describe the notational conventions that apply to all Transport Infrastructure Specifications
- Define the common Schemas for headers in the START (Secure Trusted Asynchronous Reliable Transport) and LIME (Lightweight Message Exchange Profile) transport profiles

- Describe the use of WS-Addressing headers common to all SOAP based BUSDOX transport profiles
- Describe the relation to WS-I basic profile 1.1 requirements

1.4 Terminology

Terminology use applies to all BUSDOX Transport Infrastructure Specifications.

Indication of Requirement Levels

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 RFC 2119.

Common terms

*Business Document Exchange Network (**BUSDOX**) Infrastructure Sphere:* the set of peer Access Points and Service Metadata Publishing and Locator services that:

1. Meet the governance requirements of any BUSDOX infrastructure instance
2. Are listed as endpoints by BUSDOX Service Metadata Publishers
3. Are accessible under service level agreements for APs defined in any instance of a BUSDOX infrastructure

*BUSDOX Access Point/**AP**:* a peer in a BUSDOX Infrastructure instance.

SAML Token/Assertion: The terms SAML token and SAML assertion will be used interchangeably.

*Source Access Point/**SrcAP**:* An Access Point sending a message to another Access Point

*Destination Access Point/**DestAP**:* An Access Point receiving a message from a SrcAP.

*Secure Trusted Asynchronous Reliable Transport/**START**:* The Secure Trusted Asynchronous Reliable Transport, see the START transport specification.

*Lightweight Message Exchange/**LIME**:* The Lightweight Message Exchange Transport, see the LIME transport specification.

Lightweight Client/**LC**: Client application that communicates with a LIME compliant Access Point (LIME-AP)
*Lightweight Profile Access Point/**LIME-AP**:* A BUSDOX Access Point that exposes the Lightweight Profile Interface towards client applications. An LIME-AP may be an existing VAN or a new service offered by governments or private companies.
*Message Channel/**MC**:* An LIME-AP offers a Message Channel (**MC**) interface to the LC. There are (at least) two MCs available to the LC.
*Inbound/Outbound Message Channel/**InMC**/**OutMC**:* An Inbound Message Channel stores messages destined for the LC, and the Outbound Message Channel is used by the LC as a relay for messages destined for other companies.
*Endpoint Reference/**EPR**:* Each Message Channel of a LIME-AP is uniquely identified by a WS-Addressing Endpoint Reference (EPR)
*Channel Identifier/**ChannelID**:* A channel of an LIME-AP.

*Service Metadata Locator service/**SML**:* A service which provides a client with the capability of discovering the Service Metadata Publisher endpoint associated with a particular participant identifier. A client uses this service in order to find where information is held about services for a particular participant business.
*Service Metadata Publisher/**SMP**:* A service metadata publisher offers a service on the network where information about services of specific participant businesses can be found and retrieved. It is necessary for a client application to retrieve the metadata about the services for a target participant business before the client can use those services to send messages to the participant business.
*Service Metadata Consumer/**SMC:*** A Service Metadata Consumer is any entity consuming Service Metadata provided by a Service Metadata Publisher. This is typically the sender of the document, or the sender side Access Point.
Participant Identifier: A participant business level identifier such as GLN (a GS1 Global Location Number) or DUNS (Dun & Bradstreet) number that is used to identify a trading partner. In the context of BUSDOX, participant identifiers are used to discover services associated with trading partners in Service Metadata.
*Secure Token Service/**STS**:* a service that offers security tokens to network clients.

*Transport Layer Security/**TLS**:* Transport Layer Security is the standard most used to secure HTTP. It is the successor to Secure Sockets Layer (SSL).

*Service Metadata Publisher Certificate (**SMP Certificate**):* A certificate used by a specific SMP to create all signatures of the signed resources.

Notational conventions

Notational conventions have been adopted from **Feil! Fant ikke referansekilden.** and apply to all BUSDOX Transport Infrastructure Specifications.
Pseudo-schemas are provided for each component, before the description of the component. They use BNF-style conventions for attributes and elements: "?" denotes optionality (i.e. zero or one occurrences), "*" denotes zero or more occurrences, "+" one or more occurrences, "[" and "]" are used to form groups, and "|" represents choice. Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema. Elements with simple content are conventionally assigned a value which corresponds to the type of their content, as defined in the normative schema. Pseudo schemas do not include extension points for brevity.

```
<!-- sample pseudo-schema -->
<defined_element
      required_attribute_of_type_string="xs:string"
      optional_attribute_of_type_int="xs:int"? >
  <required_element />
  <optional_element />?
  <one_or_more_of_these_elements />+
  [ <choice_1 /> | <choice_2 /> ]*
</defined_element>
```

Normative references

This section only covers references to documents referred to from this document.

[WSA-1.0] "Web Services Addressing 1.0 - Core" (http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/) and "Web Services Addressing 1.0 - SOAP Binding", http://www.w3.org/TR/ws-addr-soap/
[SOAP-1.1] "SOAP Version 1.1", http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[RFC-2119] "Key words for use in RFCs to Indicate Requirement Levels", http://www.ietf.org/rfc/rfc2119.txt
[BP-1.1] "Basic Profile Version 1.1", http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Non-normative references
[WSDL-2.0] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", http://www.w3.org/TR/wsdl20/
[RFC4122] “A Universally Unique Identifier (UUID) URN Namespace”, http://www.ietf.org/rfc/rfc4122.txt

1.5 Namespaces

The following table lists XML namespaces that are used in this document. The choice of any namespace prefix is arbitrary and not semantically significant.

Namespace Prefix	Namespace
wsa	http://www.w3.org/2005/08/addressing
ids	http://busdox.org/transport/identifiers/1.0/
xs	http://www.w3.org/2001/XMLSchema

2 Identifiers

This section defines what participant business-, document- and process-identifiers are, and how they are represented in XML elements and URLs.

2.1 Notational conventions

For describing the textual format of identifiers, the following conventions are used:

Everything within the curly brackets {} can be substituted by specific values.
Everything with square brackets [] represents optional content, whether literals or not.
Everything outside the curly brackets must be treated as literals.

For example, for an identifier with the value "0010:5798000000001", the format definition

/{identifier}/service[/endpointName]

Can be instantiated to either of the strings

/0010:5798000000001/service

And

/0010:5798000000001/service/endpointName

2.2 On the use of percent encoding in URLs

When any types of BUSDOX identifiers are used in URLs, each section between slashes MUST be percent encoded individually, i.e. section by section, according to **Feil! Fant ikke referansekilden.**. For example, this implies that for an URL in the form of "/{identifier scheme}::{id}/services/{docType}", the slash literals MUST NOT be URL encoded.

2.3 On Scheme Identifiers

Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of the BUSDOX specifications. BUSDOX defines a few generally usable identifiers, but any instance of the BUSDOX infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Any Scheme Identifier defined outside of this specification MUST take the form <domain>-<identifierArea>-<identifier type>, such as for example "busdox-actorid-upis".

- Domain: BUSDOX
- Identifier area: Actor Identifiers ('actorid')
- Identifier type: "Universal Participant Identifier Scheme" (upis)

Scheme identifiers are used in the SML spec for creating DNS aliases for SMPs, and may therefore only contain the following characters:
[a-z], [A-Z], [0-9], [-]
A scheme identifier SHOULD be as short as possible, and MUST NOT exceed 25 characters.

2.4 Participant Identifier

Participant identifiers logically consist of a *scheme identifier* and the *participant identifier* itself. Participant identifiers are associated with groups of services, or Service Metadata.

The *scheme identifier* indicates the specification of the participant identifier format, i.e. its representation and meaning. Currently there is only one scheme defined, the *Universal Participant Identifier Scheme*, and is identified by the string literal "busdox-actorid-upis".

Universal Participant Identifier Scheme
This specification only describes the 'Universal Participant Identifier scheme', although other domain-specific schemes may be defined.
The scheme indicates the format follow identifier format:
{type identifier}:{participant identifier}.
The type identifier is 4-digit number indicating the type of participant identifier, such as GLN, DUNS, CVR or another scheme. This scheme is dynamically extendable and will be decided outside of this specification. Note that the {type identifier} part of the participant identifier *is not* equivalent to the scheme of the identifier - the {type identifier} indicates the *type of participant identifier*, whereas the scheme identifier indicates the format and semantics of an identifier string.
For the moment, only the type identifier "0010" has been defined as indicating a GLN number, such as for example "0010:5798000000001".

XML format for Participant Identifiers
The <ParticipantIdentifier> element is used to represent participant identifiers and scheme information.
Pseudo-scheme for ParticipantIdentifier:
```
 <smp:ParticipantIdentifier scheme="xs:string">
   xs:string
</smp:ParticipantIdentifier>
```

Where the 'scheme' attribute indicates the scheme of the participant identifier.

Field	Description
ParticipantIdentifier	A participant identifier which may be associated with a group of services.
ParticipantIdentifier/ @scheme	The scheme of the participant identifier. This scheme indicates the *format* of the participant identifier (i.e. the textual format) – not its semantic type (e.g. DUNS or GLN). This scheme type MUST be in the form of a URI. The URI that indicates the "Universal Participant Identifier" format is "busdox-actorid-upis" When processing a participant identifier in XML format, it MUST be treated as case insensitive.

Using participant identifiers in URLs
The following format is used:
{identifier scheme}::{id}
Where 'identifier scheme' is the format of the identifier, and 'id' is the participant identifier itself, following the format indicated by the 'identifier type' part. The {id} could for example be the format defined as 'Universal Participant Identifier' scheme.
In a URL, the string represented by "{identifier scheme}::{id}" MUST be percent encoded following **Feil! Fant ikke referansekilden.**, and the guidelines given above.

Non-normative example that uses the Universal Participant Identifier Format, assuming the participant identifier “0010:5798000000001”:
busdox-actorid-upis::0010:5798000000001
In percent encoded form:
busdox-actorid-upis%3a%3a0010%3a5798000000001
When processing a participant identifier in an URL, it MUST be treated as case insensitive. Note that any surrounding slashes which belong to the URL rather than the various identifiers (which may take the forms of URLs) are *not* percent encoded.

2.5 DocumentIdentifier

Documents are represented by an identifier (identifying the document type) and a scheme type which represents the scheme or format of the identifier itself. It is outside the scope of this document to list identifier schemes that may be valid in the BUSDOX context.
This specification defines a single identifier scheme, the ‘QName/Subtype Identifier Scheme’, which is identified by the following URI:
busdox-docid-qns
This scheme is based on a concatenation of the document namespace, root element, and optional (and document-dependent) subtype:
{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}].
Where ‘[ ]’ denotes an optional part of the identifier, and everything outside ‘{ }’ are string literals.
For example, in the case of a NES-UBL order, this document can then be identified by

- **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2
- **Document element local name:** Order
- **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same namespace + document element name)

The document type identifier will then be:
urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0

XML Representation of Document Identifiers
The <DocumentIdentifier> element is used to represent document identifiers and scheme information.
Pseudo-scheme for DocumentIdentifier:
```
  <DocumentIdentifier scheme=”xs:string”>xs:string</DocumentIdentifier>
```

Where the ‘scheme’ attribute indicates the scheme of the document identifier.

Field	Description
DocumentIdentifier	A document identifier representing a specific document type. In the case of a NES-UBL order document, this would be “urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0”.
ParticipantIdentifier/ @scheme	The scheme of the document identifier. This scheme identifier MUST be in the form of a URI. This document defines the ‘QName/Subtype Identifier Scheme’, which is identified by the following URI: *busdox-docid-qns* When processing a document identifier in XML format, it MUST be treated

	as case insensitive.

URL representation of Document Identifiers

When representing document identifiers in URLs, the document identifier itself will be prefixed with the scheme identifier. For example, the 'QName/Subtype Identifier Scheme' is indicated by this identifier:
"busdox-docid-qns".
The format of this is:
{identifier scheme}::{id}
In the case that the 'QName/Subtype Identifier Scheme' is used, the complete format is:
{identifier scheme}::{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}].
As a non-normative example, in the case of a NES-UBL order, this document can then be identified by

- **Identifier scheme:** busdox-docid-qns
- **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2
- **Document element local name:** Order
- **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same namespace + document element name)

The document type identifier will then be:
busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0
Rules for parsing this identifier:

- The text up until the first "::" is the identifier scheme identifier
- The text before the next to last "::" and last "::" is the root namespace
- The text between the last occurrence of '::' and last occurrence of '##' OR end of the string is the document element local name.
- The text following the first '##' after the document element local name (if any) is the subtype identifier.

Note that although namespaces and element names are case sensitive, the document identifier MUST be treated as case insensitive.
This string must be percent encoded (see **Feil! Fant ikke referansekilden.**, sect. 2.1) if used in an URL. In that case, the above identifier will then read as:
busdox-docid-qns%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aOrder-2%3a%3aOrder%23%23UBL-2.0

Note the limitation that XML document types with the following characteristics MUST NOT be referenced using Service Metadata Publishing:

- Documents with only local names (i.e. without namespaces)
- Documents that need to be identified with a subtype identifier, and where the subtype part of the identifier does not correspond to a specific, mandatory attribute value or element value in the document that is based on XML Schema simple content.

2.6 Process Identifiers

A process identifier represents a process that a specific document type can participate in. Process identifiers consist of the process identifier itself, and a scheme or identifier format type. As for the other schemed identifier types, additional process identifier schemes may be defined outside of the BUSDOX specifications.

XML Representation of Process Identifiers

Pseudo-schema for the ProcessIdentifier XML element:

```
<ProcessIdentifier scheme=”xs:string”>xs:string</ProcessIdentifier>
```

Description of the individual fields (elements and attributes).

Field	**Description**
ProcessIdentifier	The identifier of the process. A process is identified by a string that is defined outside of this specification. For example, the CEN BII may choose to indicate a UBL-based 'simple procurement' process (or 'profile' in UBL terminology) with the identifier "BII07", or UBL-based basic invoice exchange profile with the identifier "BII04". This document just defines one process identifier, which represents documents that are *not* sent under any specific process: busdox:noprocess The process identifier MUST be treated as case insensitive. This identifier MUST be used with the process scheme "busdox-procid-transport"
ProcessIdentifier/@scheme	Indicates the format of the process identifier. The format of this identifier may be different from domain to domain – for example, in a domain where BPEL definitions of processes exist, a technical identifier derived from the BPEL definitions may be used, whereas a taxonomy may be created in another domain. Processes (or profiles) defined by the CEN BII workshop could for example choose to use the identifier "cenbii-procid-ubl" to indicate the format of process identifiers. This document just defines one process scheme identifier, which represents transport-specific process identifiers: busdox-procid-transport Currently the only valid process identifier under this scheme is the identifier "busdox:noprocess", which indicates that a message is not sent under any named process.

2.7 BUSDOX defined identifiers

The identifiers defined in this section are used both with the Service Metadata specifications and the transport specifications. Every BUSDOX message has associated metadata included, in the form of headers, so that Access Points can route messages without relying on knowledge of the message payload.
For an XML Schema for these elements, see the section "**Feil! Fant ikke referansekilden.**".

Recipient Participant Identifier
This element represents the participant identifier of the ultimate recipient. This is used for Service Metadata Lookup and message forwarding. Pseudo schema for this element is:
```
<ids:RecipientIdentifier scheme=”xs:string”>xs:string</ids:RecipientIdentifier>
```

This element contains both the Recipient identifier and identifier scheme. It follows the rules for XML representation of participant identifiers laid out in section **Feil! Fant ikke referansekilden.**.

Sender Participant Identifier
This element represents the participant identifier of the original sender. This is used for Service Metadata lookup and message forwarding. Pseudo schema for this element is:
```
<ids:SenderIdentifier scheme=”xs:string”>xs:string<ids:SenderIdentifier>
```

This element contains both the Sender identifier and identifier scheme. This element contains both the Recipient identifier and identifier type. It follows the rules for XML representation of participant identifiers laid out in section **Feil! Fant ikke referansekilden.**.

Document Type Identifier
This element represents the type of document enclosed in the message. This is used for Service Metadata lookup/routing. Pseudo schema for this element is:
```
<ids:DocumentIdentifier scheme=”xs:string”>xs:string</ids:DocumentIdentifier>
```

This element contains the Document identifier and identifier scheme. It follows the rules for XML representation of document identifiers laid out in section **Feil! Fant ikke referansekilden.**.

Process Type Identifier
This element represents the type of process that a document may participate in. This is used for Service Metadata lookup. Pseudo schema for this element is:
```
<ids:ProcessIdentifier scheme=”xs:string”>xs:string</ids:ProcessIdentifier>
```
This element follows the rules for XML representation of process identifiers laid out in section 2.6.

Message Identifier
Because BUSDOX Messages may pass between several parties (for example in the “four-corner” model, from LIME client1 to AP1 to AP2 to LIME client2), it is desirable to have a constant message identifier that uniquely identifies the message across multiple hops, for tracing purposes. This message identifier is contained in the ‘MessageIdentifier’ element:

    <ids:MessageIdentifier>xs:string</ids:MessageIdentifier>

Channel Identifier
Channel identifiers are used with the START and LIME profiles in order to discern between multiple channels of communication within each AP. It has the following form:

    <ids:ChannelIdentifier>xs:string</ids:ChannelIdentifier>

2.8 Basic Profile

The conformance Requirements in the WS-I Basic Profile 1.1 specification MUST be followed.

2.9 SOAP 1.1

APs MUST use SOAP 1.1 for all message exchange.
Messages MUST use the document/literal style.

2.10 Use of HTTP

Messages MUST use the “text/xml” media type, unless MTOM is being used. If MTOM is in place, the HTTP media type MUST be “multipart/related”, and the content-type of the part containing the SOAP message MUST be “text/xml”.
The SOAPAction HTTP Header MUST be used, and MUST correspond to the WS-Addressing ‘Action’ attribute.

2.11 WS-Addressing 1.0

This profile only supports the use of WS-Addressing 1.0.

The <wsa:To> Header
All outgoing message interactions MUST include a wsa:To header. Responses MAY omit this header.

The <wsa:MessageIdentifier> Header
All message interactions MUST include a wsa:MessageIdentifier header.
Any participant that assigns a value to a <wsa:MessageIdentifier> header block MUST ensure that there is negligible probability that that participant or any other participant will accidentally assign the same identifier to any other message. Parties SHOULD use the UUID standard (see [RFC4122])for generating message Ids.
Note that [WSA-1.0] requires that <wsa:MessageIdentifier> values be absolute IRIs.

The <wsa:RelatesTo> Header
Responses MUST have exactly one <wsa:RelatesTo> header present, and if the RelationshipType attribute is available, it MUST be set to the value “http://www.w3.org/2005/08/addressing/reply”.

EndpointReferences
EndpointReferences MUST NOT have <wsa:Metadata> elements.

3 Undeliverable Messages

There may be situations where an Access Point is unable to deliver the message sent by either the LIME client or another START Access Point. These include but are not limited to:

- The Recipient ID is not listed in the SML
- The particular document type or process is not listed in the SMP
- The recipient AP is not contactable over a long period of time.
- The recipient AP is contactable but the message cannot be delivered due to faults or WSRM Sequence problems.
- The certificate of the receiving AP has been revoked.

- A queue at the Access Point is full.

In these cases, the AP at which the error is identified SHOULD create a business message targeted at the sender identifier which conveys that the message is undeliverable. An example of this message is:

```
       <ids:MessageUndeliverable>
            <ids:MessageIdentifier>
                    uuid:45989-2429-132412313
            </ids:MessageIdentifier>
            <ids:ReasonCode>METADATA_ERROR</ids:ReasonCode>
            <ids:Details>Some further details here</ids:Details>
     </ids:MessageUndeliverable>
```

When sending this message, the AP MUST use the following process:

1. Look up the sender identifier together with the proscribed document identifier and process identifier (see below) in the Service Metadata Locator and Service Metadata Publisher according to the normal BUSDOX approach.

2. If there is no entry for the sender identifier or the MessageUndeliverable document, then no message will be sent and the sender will not be notified of the MessageUndeliverable status. In this case, the AP SHOULD log this locally for the benefit of system administrators.

3. If the AP has an identifier in the BUSDOX network, then the AP MUST use this identifier for sending the MessageUndeliverable message. However, it is not a requirement that APs have BUSDOX participant identifiers. In the case that the AP does not have a participant identifier, then the message should have the following fixed value for the ids:SenderIdentifier header:

```
<smp:ParticipantIdentifier scheme=”busdox-actorid-transport”>
    busdox:sender
</smp:ParticipantIdentifier>
```

This message is sent as a business message to the appropriate endpoint that has been resolved from the SML and SMP. This MAY be a LIME inbox for the original sender, or it MAY be a special destination managed by the sender's AP.
The XML is defined as follows.

/ids:MessageUndeliverable

The holding element is required. It has no attributes.

/ids:MessageUndeliverable/ids:MessageIdentifier

This required element contains the message identifier of the undelivered message. It is exactly the same as the corresponding header of the message that was not delivered.

/ids:MessageUndeliverable/ids:ReasonCode

This required element contains one of the following values of type xs:string:

1. METADATA_ERROR: This string indicates that there was an addressing error with the message and no matching metadata can be found: for example the participant identifier is not available in the SML, or the document or process type is not listed in the SMP, or not known by the receiving Access Point.

2. TRANSPORT_ERROR: This string indicates that the message cannot be delivered because of a failure to transport the document to the next hop in the chain. For example, the receiving AP's endpoint is not responding within a significant timeout, or there are faults in trying to connect, or the WSRM sequence is failing to complete.

3. RECIPIENT_ERROR: There is a problem with the recipient. For example: The recipient has a full inbox; the recipient is uncontactable; the recipient's contract with the AP has expired or they have been suspended by the governance body.

4. SECURITY_ERROR: This string indicates that there was an authentication error or security fault returned by the receiving AP, or that the receiver's certificate has been revoked.

5. OTHER_ERROR: This string indicates that there was an error other than the above.

/ids:MessageUndeliverable/ids:Details

This required element of type xs:string contains human readable descriptions of the details of the problem. There is no restriction on its use. It MAY be empty.

DocumentIdentifier

The DocumentIdentifier of the metadata of a MessageUndeliverable message MUST be set to

```
<ids:DocumentIdentifier scheme="busdox-docid-qns">
      http://busdox.org/transport/lime/1.0/::MessageUndeliverable
</ids:DocumentIdentifier>
```

ProcessIdentifier

The ProcessIdentifier of the metadata of a MessageUndeliverable message MUST be set to

```
<ids:ProcessIdentifier scheme="busdox-actorid-transport">
       busdox:noprocess
</ids:ProcessIdentifier>
```

4 Appendix

4.1 XML Schema for message identifiers

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Identifiers"
targetNamespace="http://busdox.org/transport/identifiers/1.0/"
           elementFormDefault="qualified"
           xmlns="http://busdox.org/transport/identifiers/1.0/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:annotation>
    <xs:documentation>
      Common identifiers for WSDLs and Schemas
    </xs:documentation>
  </xs:annotation>

  <xs:element name="ParticipantIdentifier"
type="ParticipantIdentifierType"/>
  <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
  <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>

  <xs:element name="RecipientIdentifier"
type="ParticipantIdentifierType"/>
  <xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
  <xs:element name="MessageIdentifier" type="MessageIdentifierType"/>
  <xs:element name="ChannelIdentifier" type="ChannelIdentifierType"/>

  <xs:complexType name="ParticipantIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="DocumentIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ProcessIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="MessageIdentifierType">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ChannelIdentifierType">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

</xs:schema>
```