PEPPOL
PAN-EUROPEAN PUBLIC PROCUREMENT ONLINE

Specification

OpenPEPPOL AISBL

Transport Infrastructure Coordinating Community

ICT - Models

PEPPOL Transport Infrastructure AS2 Profile

Authors:
Edmund Gray, IT Sligo, Ireland
Martin Forsberg, DIGG (former ESV), Sweden

Version: 1.2
Status: In use

Revision History

Version	Date	Author	Organisation	Description
draft	2013-06-13	Edmund Gray Martin Forsberg	IT Sligo ESV	First version
Final	2013-12-09	Team		Final version with input from public review.
1.01	2018-02-09	Philip Helger	BRZ	Made TLS 1.2 mandatory (line 302) Layout cleansing
1.2	2019-01-28	Jerry Dimitriou	OpenPEPPOL	Updated CMS Specification to RFC 5652, making SHA-256 mandatory and SHA-1 optional. Examples aligned according to the new CMS Spec Updated S/MIME Specification to v3.2 (RFC 5751) Ports are required in the range 443 or 44300 – 44399 to align with AS4 specs.

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Contributors

Martin Forsberg, ESV

Edmund Gray, Institute of Technology Sligo

Markus Gudmundsson, Unimaze Software

Jostein Frømyr, Difi/Edisys Consulting

Klaus Vilstrup Pedersen, DIFI

Steinar Overbeck Cook

Oriol Bausà, Invinet

Sven Rasmussen, DIGST

Stefano Monti, EPOCA/IntercentER

Padraig Harte, Institute of Technology Sligo

Philip Helger, BRZ

Jerry Dimitriou, OpenPEPPOL Operating Office

Hans Berg, Tickstar

Risto Collanus, Visma

Bård Langöy, Pagero

Table of contents

1 Introduction

This specification is designed to facilitate becoming a compliant Access Point under the governance of the OpenPEPPOL Association. The OpenPEPPOL Association is comprised of public and private members of the PEPPOL community (see http://peppol.eu) and has taken over responsibilities for PEPPOL specifications, building blocks and services. Throughout this document the word PEPPOL refers to both the community and the association involving these responsibilities and reflects the requirements of the PEPPOL Transport Infrastructure Coordinating Community (TICC).

1.1 Objective

This document describes a specification to be used to exchange business messages between Access Points (AP) as part of the PEPPOL infrastructure. It uses the AS2 specification as specified in RFC4130 HTTP Applicability Statement 2 (AS2). AS2 was chosen because of its popularity among existing EDI Service Providers and the fact that it has already undergone extensive interoperability testing. This specification therefore focusses on leveraging these existing AS2 systems to become part of the PEPPOL network of Access Points. This specification will show how these systems can be enhanced by using the PEPPOL Service Metadata Lookup (SML), based on the appropriate BUSDOX specification, to dynamically exchange various security parameters including Public keys, Endpoint URLs etc. and therefore automate the inclusion of new or modified APs.

The PEPPOL AS2 Specification uses security settings which are equivalent to the Secure Trusted Asynchronous Reliable Transport (START) security settings, the original PEPPOL document exchange protocol. AS2 uses an S/MIME-based profile which provides security using Digital Certificates in much the same way as START. Therefore, the same Certificates can be used for both protocols. It also uses URLs to identify the Endpoint addresses therefore the Service Metadata obtained from existing SMPs can be reused for AS2 Endpoints.

AS2 provides a Transport infrastructure for exchanging structured business data securely using the HTTP transfer protocol. This exchange is normally XML but can also exchange other Electronic Data Interchange (EDI) formats such as the UN Electronic Data Interchange for Administration, Commerce, and Transport (UN/EDIFACT) format. The data is packaged using standard MIME structures. Authentication and data confidentiality are obtained by using Cryptographic Message Syntax (CMS) with S/MIME security body parts. Authenticated acknowledgements make use of multipart/signed Message Disposition Notification (MDN) responses to the original HTTP message. This provides a non-repudiation of receipt[1] for the exchange of an electronic business message and therefore assures the sender of the message transport status.

The PEPPOL AS2 Transport Specification defines a secure, reliable profile using a set of well-known standards and specifications for PEPPOL Access Point's data exchange:

- BUSDOX Metadata Lookup and publishing specifications and services
- PEPPOL Business Message Envelope of UN/CEFACT Standard Business Document Header
- RFC 4130 HTTP Applicability Statement 2 – AS2
- RFC 2616 Hyper Text Transfer Protocol
- RFC 1767 EDI Content Type
- RFC 3023 XML Media Types
- RFC 1847 Security Multiparts for MIME
- RFC 3462 Multipart/Report
- RFC 2045 to 2049 MIME RFCs
- RFC 3798 Message Disposition Notification
- RFC 5751 S/MIME v3.2 Specification
- RFC 5652 Cryptographic Message Syntax (CMS)
- RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2

[1] The term non-repudiation of receipt (NRR) is often used in combination with receipts. NRR refers to a legal event that occurs only when the original sender of an interchange has verified the signed receipt coming back from recipient of the message, and has verified that the returned MIC value inside the MDN matches the previously recorded value for the original message

- RFC 8446 The Transport Layer Security (TLS) Protocol Version 1.3

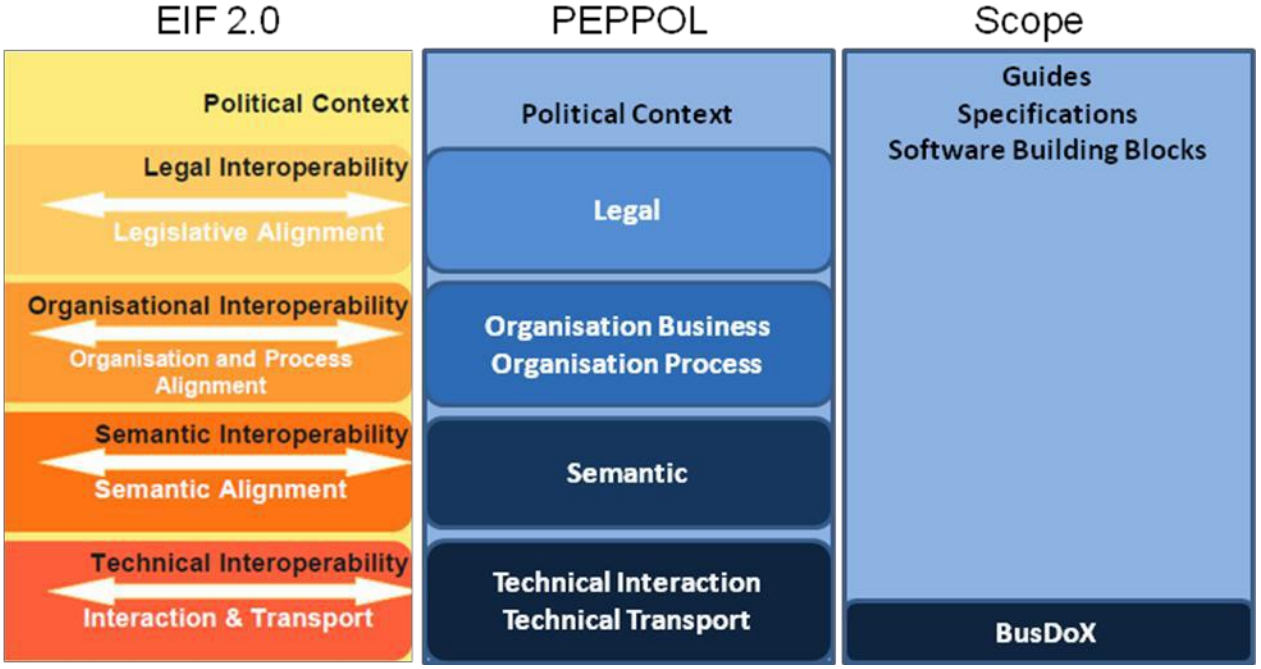
PEPPOL Access Points communicate in a peer-to-peer model across the internet to form the PEPPOL infrastructure. Each Access Point derives the endpoint addresses of other PEPPOL Access Points through the PEPPOL Service Metadata Publishing/Lookup (SMP/SML) Infrastructure.

In order to instantiate a working network, certain profile information is expected. The complete PEPPOL infrastructure includes governance models, certificate rules, identifier formats, and other profiling information published elsewhere. This specification therefore excludes such profiling information but refers to them when appropriate.

This specification profile describes the usage of these standards to support the requirements of PEPPOL. In particular the usage of these standards is restricted to certain patterns to enable interoperability to be achieved.

1.2 Scope

This specification relates to the Technical Transport Layer i.e. PEPPOL specifications. The PEPPOL specifications can be used in many interoperability settings, it provides transport for e-procurement messages for both pre and post award scenarios as specified in the PEPPOL Profiles.



1.3 Goals and non-goals

The goal of this profile is to support a high level of assurance and proof-of-delivery across the PEPPOL Infrastructure. The profile is designed to:

- Facilitate implementers to leverage existing systems and therefore gain access to PEPPOL, without the need to make significant changes to existing systems.
- Clearly state the transport level requirements in a single document.
- Identify the additional steps required to update an existing AS2 system so it complies with the requirements and can therefore participate as a PEPPOL compliant Access Point (AP).
- Define a simple, interoperable, reliable and safe communications pattern that APs can use to communicate.
- Define the message exchange formats and patterns clearly.
- Ensure that messages are reliably delivered between APs, including providing the prerequisites for logging and proof-of-delivery for messages at the transport level
- Ensure confidentiality during the exchange by using transport-level encryption using Transport Level

Security (TLS).

- Ensure integrity and authenticity of received messages. This is maintained by using the Cryptographic Message Syntax (CMS) specified in RFC 5652, which is used to digitally sign, digest, authenticate and encrypt the electronic message.
- Establish a common format for representing authentication and authorisation events using PEPPOL provided Digital Certificates.
- Recipients can assume that senders are trusted by the trust chain of the PEPPOL issued certificates and the Governance documents already signed by members.

The Profile does NOT address:

- The verification of certificates, format of participant identifiers, and other details required to create a full instantiation of PEPPOL.
- Communication with PEPPOL Service Metadata services.

1.4 Terminology

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.

1.4.1 Normative references

Bradner S., “Key words for use in RFCs to Indicate Requirement Levels”, RFC 2119, March 1997

Moberg D., “MIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP, Applicability Statement 2 (AS2)”, RFC 4130, July 2005.

Hansen T., “Message Disposition Notification”, RFC 3798, May 2004.

Ramsdell, B., Turner, S., “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification”, RFC 5751, January 2010.

Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 3462, January 2003.

Ramsdell, B., Turner, S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling", RFC 5750, January 2010.

Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

2 Overview

The PEPPOL AS2 specification provides a secure reliable approach for messages exchange from one PEPPOL Access Point (AP) to another. The key factor here is utilizing the SMP lookup in an efficient way so that existing APs can use the retrieved metadata to automatically facilitate the exchange. A pre-requisite for using this profile in the PEPPOL Infrastructure is that the business message is wrapped in a message envelope. It should be noted that AS2 is payload agnostic, but the use of AS2 in the PEPPOL infrastructure context requires the use of business message envelope. The envelope provides a standard way to encapsulate routing information irrespective of the type of business message used. It therefore obviates the need for APs to read the contents of the business message. The envelope carries, in its header, several of the service metadata elements that are necessary for the receiving AP to ensure that the message is sent to the correct channel and service. The details of this envelope are described in a separate document entitled “PEPPOL Business Message Envelope (SBDH)”.

Figure 1 Illustration of a Business Document Envelope

The Business Message Envelope contains a unique identifier (InstanceIdentifier) used to identify a specific instance of a Message Envelope. This identifier is something completely different form the AS2 Message-ID described in this specification. The Message-ID is unique for every AS2 transmission. Hence, if a message is resent, the InstanceIdentifier may be the same but the AS2 Message-ID must be different.

2.1 A typical workflow

A typical workflow between SrcAP (source Access Point) to DestAP (destination Access Point) might be:

- An electronic message is issued by Company C1 and handed over to SrcAP for transportation to the DestAP and finally delivered to ultimate receiver Company C2. The method used to communicate between C1 and SrcAP and correspondingly DestAP and C2 is outside the scope of this document but the SrcAP MUST assure the authenticity of Company C1 and integrity of the message (4-corner model).
- The message handed over by C1 to the SrcAP, includes an envelope with required information such as:
  - Recipient Identifier and identifier type
  - Sender Identifier and identifier type
  - Document identifier
  - Process identifier
- The SrcAP uses the recipient identifier and specific document and process information to look-up the necessary service metadata from the SML/SMP system. The SrcAP may decide to cache the service metadata depending on transaction volumes or other factors, which should be no longer than 24 hrs.
- SrcAP validates that the metadata was signed by a PEPPOL certificate.
- SrcAP gets PEPPOL issued Private Key X509 certificate for signing from its own certificate stores.
- SrcAP MUST ensure that the message envelope carries the correct headers containing identifiers for recipient and sender, process type and document identifier.

- SrcAP signs the message using the PEPPOL AP Certificate Private Key.
- SrcAP uses HTTPS to send message securely to DestAP using the URL as retrieved from the SMP and in accordance with AS2 specification RFC 4130.
- DestAP responds (synchronously) with a signed proof-of-delivery message to SrcAP using the Message Delivery Notification (MDN) specification as specified in the AS2 specification RFC 4130.
- Finally SrcAP archives the MDN as a signed proof-of-delivery of the message. The expectation is that most Access Points will act as both SrcAP and DestAP, however this is not required by the specifications.

3 Specification Profile Details

The following requirements apply to the PEPPOL AS2 Profile. The functionality used in this profile of AS2 is included in the AS2 Version 1.0.

3.1 Use of HTTP

AS2 is based on the transmission using HTTP protocol. It consists of a set of headers and a payload. HTTP header names in the specification are always to be treated case insensitive. HTTP header values are to be treated case sensitive.

3.2 Use of Digital Certificates

In this specification the use of PKI ensures security of transmission by using PEPPOL supplied certificates for signing and the use of a signed MDN provides a non-repudiatable transaction. The sender does this by verifying the signed MDN with the receiving partner's public key, and by verifying that the returned MIC (Message Integrity Check) value in the MDN is the same as the MIC for the original message.

3.3 Message Exchange

This profile uses HTTPS for secure transport and S/MIME for content, including a digital signature, to send any electronic business message from one Access Point to another. The transmission should be idempotent so that the SrcAP can resend to DestAP without fear of duplicate error responses.

The SrcAP SHOULD implement a resend strategy but it may be configured dependent on the business context.

3.4 Prerequisites for communication

Before an Access Point can deliver a message to another Access Point, the SrcAP MUST have the following information, which it MAY find in the BUSDOX Service Metadata Publishing document:

- The Endpoint/EndpointReference as Address (URL) for the DestAP.
- The Endpoint/Certificate (see the use of AS2-To header value below)

The transmission MUST include an enveloped message with the following Service Metadata defined in the separate document specification PEPPOL Business Message Envelope (SBDH):

Necessary value	**Location in SBDH**
RecipientIdentifier	/StandardBusinessDocument/StandardBusinessDocumentHeader/Sender/Identifier
SenderIdentifier	/StandardBusinessDocument/StandardBusinessDocumentHeader/Receiver/Identifier
DocumentIdentifier	/StandardBusinessDocument/StandardBusinessDocumentHeader/BusinessScope/Scope[Type='**DOCUMENTID**']/InstanceIdentifier
ProcessIdentifier	/StandardBusinessDocument/StandardBusinessDocumentHeader/BusinessScope/Scope[Type='**PROCESSID**']/InstanceIdentifier

3.5 Delivery of PEPPOL messages

The SrcAP will consider the message to be delivered when it receives an MDN signifying that the message has been successfully processed and no error is received. Each message has a unique Id (Message-Id field in the AS2 header), and the SrcAP should verify which messages have yet to be receipted by comparing with the Original-Message-Id in the MDN. The MDN MUST be sent synchronously. The Message-Id MUST be globally unique.

3.5.1 Use of AS2-From and AS2-To headers

The AS2-From and AS2-To headers are used as mandatory to identify SrcAP and DestAP. The DestAP MUST accept messages from SrcAP provided it is correctly authenticated by the use of validating that the payload is signed with valid PEPPOL Certificates.

The DestAP MUST NOT require pre-configuration (or bi-lateral agreements) for new SrcAP and should be able to dynamically determine a new or changed SrcAP.

The values of AS2-From MUST correspond to the CN-value (Common Name) of the AP Certificate used in the transmission. The AS2-To value MUST correspond to the CN-value of the DestAP Certificate. The CN-value can be retrieved from the Endpoint/Certificate in the service metadata (from the SMP). The CN is issued by OpenPEPPOL and therefore these identifiers cannot be set until a PEPPOL signing Digital Certificate is available.

The value of an AS2-To header in an MDN MUST match the value of the AS2-From header value in the corresponding request message. Likewise, the value for the AS2-From header in an MDN MUST match the value of the AS2-To header in the corresponding AS2 request message.

3.5.2 Non-normative AS2 Headers example

```
content-disposition = attachment; filename="smime.p7m"
as2-from = APP_1000000002
connection = close, TE
ediint-features = multiple-attachments, CEM
date = Fri, 29 Nov 2013 15:12:00 CET
as2-to = APP_1000000003
disposition-notification-to = http://domain.com/cipa-as2-access-point-
wrapper/AS2Receiver
message-id = <mendelson_opensource_AS2-1385734320013-0@APP_1000000002_mend>
subject = AS2 message
from = as2@company.com
as2-version = 1.2
disposition-notification-options = signed-receipt-protocol=optional, pkcs7-
signature; signed-receipt-micalg=optional, sha-256, sha256, sha-1, sha1
content-type = multipart/signed; protocol="application/pkcs7-signature";
micalg=sha1; boundary="----=_Part_1_1908557897.1385734320094"
host = as2server.DestAP.com
mime-version = 1.0
recipient-address = http://domain.com/cipa-as2-access-point-
wrapper/AS2Receiver
```

3.5.3 Non-normative AS2 Headers MDN example

```
as2-from = APP_1000000003
connection = close
ediint-features = multiple-attachments, CEM
date = Fri, 29 Nov 2013 15:12:05 CET
server = mendelson opensource AS2 1.1 build 41 - www.mendelson-e-c.com
as2-to = APP_1000000002
content-length = 3035
message-id = <mendelson_opensource_AS2-1385734320013-0@APP_1000000002_mend>
as2-version = 1.2
content-type = multipart/signed; protocol="application/pkcs7-signature";
micalg=sha-256;  boundary="----=_Part_61_13593581.1385637260652"
mime-version = 1.0
```

3.5.4 Faults/Errors returned

Typically, all AS2 errors from DestAP are returned using the MDN and the error reported in the "disposition-field". The DestAP has several integrity checks all of which may return errors. If the disposition-field states "MDN-sent-automatically; processed" then the transmission was successful. When it is not successful, the "disposition-field" MUST include a disposition-modifier indicating the error or failure (see list below). Other errors would be considered normal socket or HTTP errors and are outside the scope of this document. A failure indicates that the DestAP cannot understand the MDN requirements of the SrcAP. A warning indicates that the message was accepted for further processing although there were errors. The AS2 standard specification contains a number of faults. The list below enumerates some examples of failures/errors/warnings that SHOULD be used. This specification also adds faults that may occur in the PEPPOL infrastructure.

Failures	Possible cause
Failure: unsupported format	if the DestAP determines that a signed receipt cannot be returned because it does not support the requested protocol format.
Failure: unsupported MIC-algorithms	The SrcAP requested a MIC-Algorithm which the DestAP does not support
Failure: sender-equals-receiver	The AS2-To name is identical to the AS2-From name.
Errors	**Possible cause**
Error: decryption-failed	the DestAP could not decrypt the message contents.
Error: authentication-failed	the DestAP could not authenticate the sender. Sender in this profile is the sending AP.
Error: integrity-check-failed	the DestAP could not verify content integrity.
Error: participant-not-accepted (*)	The DestAPcould cannot identify the participant as described in the received service metadata. This may occur if the SrcAP is using stale cached service metadata that has been updated.
Error: document-type-id-not-accepted (*)	The DestAP does not accept documents of this type. The document identifier, as described in the envelope, does not correspond to the DestAPs and records. This may occur if the SrcAP is using stale cached service metadata that has been updated.
Error: process-id-not-accepted (*)	The DestAP does not accept documents of this type. The process identifier, as described in the envelope, does not correspond to the DestAPs and records. This may occur if the SrcAP is using stale cached service metadata that has been updated.
Error: unexpected-processing-error	a catch-all for any additional processing errors.
Warnings	**Possible causes**
Warning: duplicate-document	An identical message already exists at the DestAP.

(*) Error types recommended by OpenPEPPOL (not part of the AS2 RFC)

3.6 Security

PEPPOL supplied certificates MUST be used for message signing and the returned MDN MUST also be signed using SHA-256 algorithm, according to RFC 5652[2]. The MDN validation process ensures a non-repudiable transaction. The sender does this by verifying the signed MDN with the receiving partner's public key, and by verifying that the returned MIC (Message Integrity Check) value in the MDN is the same as the MIC for the original message. Messages MUST be encrypted during transport. This is achieved using a transport protocol (HTTPS) which obviates the need for message level encryption. An encryption at message envelope level MUST NOT be applied.

3.6.1 Message Authentication and Integrity

Authentication and integrity of messages is established by means of digital signatures applied to the S/MIME message. The authentication algorithm performs the following (according to RFC 4130):

- The message integrity check (MIC or Message Digest), is decrypted using the sender's public key.
- A MIC on the signed contents (the MIME header and encoded EDI object, as per RFC 1767) in the message received is calculated using the same one-way hash function that the sender used.
- The MIC extracted from the message that was sent and the MIC calculated using the same one-way hash function that the sending trading partner used are compared for equality.

[2] SHA-1 algorithm MUST only be supported for backwards compatibility during the migration process.

3.6.2 Responses

The signed MDN, when received by the sender of the EDI Interchange, can be used by the sender as follows (according to RFC 4130):

- As an acknowledgement that the EDI Interchange sent was delivered and acknowledged by the receiving trading partner. The receiver does this by returning the original-message-id of the sent message in the MDN portion of the signed receipt.
- As an acknowledgement that the integrity of the EDI Interchange was verified by the receiving trading partner. The receiver does this by returning the calculated MIC of the received EC Interchange (and 1767 MIME headers) in the "Received-content-MIC" field of the signed MDN.
- As an acknowledgement that the receiving trading partner has authenticated the sender of the EDI Interchange.
- As a non-repudiation of receipt when the signed MDN is successfully verified by the sender with the receiving trading partner's public key and the returned MIC value inside the MDN is the same as the digest of the original message.

3.6.3 Validation

The receiver of either request or response messages MUST validate the message signature (PEPPOL issued X.509 certificates) including issuer signature, test of validity period and Certificate trust chain through PEPPOL provided root and intermediate certificates. The sender SHOULD NOT provide a certificate chain as part of the certificate information in a transmission. Depending on local policy, the receiver SHOULD check revocation status of any certificates used to sign and encrypt the message.

The SrcAP SHOULD validate that the Subject Unique Identifier of the certificate used to sign the response messages matches the Subject Unique Identifier of the certificate published in the Service Metadata Publisher (SMP).

When validating a signed response message, the SrcAP SHOULD check that the certificate in the response matches the metadata received from the Service Metadata Publisher. This is done by comparing the subject common name in the certificate to the value stated in the metadata. This check ensures that only the legitimate Access Point stated in the service metadata will be able to produce correct responses. If the MIC provided in the MDN response does not equal the MIC computed by the sender, this must be handled out-of-band.

3.6.4 Use of HTTPS

Messages MUST be transmitted using HTTPS POST using trusted SSL certificates - which prevents a “man-in-the-middle” attack - as follows:

- The DestAP MUST implement HTTPS with certificate chains to certificate authorities which would be considered to be trusted by the PEPPOL community.
- It SHOULD be a 2048 bit Certificate or better.
- The certificate MUST correctly identify the DestAP URL e.g. no self-signed certificates.
- The certificate MUST NOT be expired or revoked.
- The DestAP MUST use a simple TLS handshake.
- It MUST use TLS v1.2 (as described in RFC 5246) or TLS v1.3 (as defined in RFC 8446).
- The DestAP URL MUST only refer to HTTPS.
- The DestAP URL SHOULD use the default port 443. This assures firewall rules are often setup in advance. In case this is not possible, then the DestAP MUST use a port from the range 44300 to 44399 inclusive.
- The DestAP MAY use wildcard certificates to facilitate multiple URLs under the same trusted domain.

3.6.5 Reliable exchange behaviour

The Request-URI[3] identifies a process for unpacking and handling the message data and for generating a reply for the client that contains a signed message disposition acknowledgement (MDN). The MDN is returned in the

[3] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

HTTP response message body. This request/reply transactional interchange provides secure, reliable, and authenticated exchange using HTTP as a transfer protocol.

The following requirements ensure that the reliable messaging framework effectively delivers messages from SrcAP to DestAP, or leaves the Access Points with a clear status of the transmitted messages.

- The SrcAP MUST assume unacknowledged messages are not delivered or accepted and SHOULD resend within a reasonable time span.
- The SrcAP MUST assume that only messages which have been receipted without error or failure have been successfully delivered.
- If the SrcAP is sending a transmission, then the DestAP closes the connection after 5 to 15 seconds to allow the channel to be reused and/or ensure SrcAP has received the signed acknowledgement response.
- The SrcAP SHOULD keep a persistent log of these signed acknowledgements for a reasonable length of time.

4 Appendix A

4.1 Example Failures/Errors

(Source RFC 4130) The following set of examples represents allowable constructions of the Disposition field that combine the historic constructions above with optional RFC 3798 error, warning, and failure fields. AS2 implementations MAY produce these constructions. However, AS2 servers are not required to recognize or process optional error, warning, or failure fields at this time. Note that the use of the multiple error fields in the second example below provides for the indication of multiple error conditions.

Message handled successfully:

```
Disposition: automatic-action/MDN-sent-automatically; processed
```

Message with 2 errors:

```
Disposition: automatic-action/MDN-sent-automatically;
  processed/error: decryption-failed
Error: The signature did not decrypt into a valid PKCS#1 Type-2 block.
Error: The length of the decrypted key does not equal the octet
   length of the modulus.
```

Message handled with a warning:

```
Disposition: automatic-action/MDN-sent-automatically;
  processed/warning: duplicate-document
Warning: An identical message already exists at the
  destination server.
```

Message handled with a failure:

```
Disposition: automatic-action/MDN-sent-automatically;
  failed/failure: sender-equals-receiver
Failure: The AS2-To name is identical to the AS2-From name.
```

4.2 Sample instance document

Source: PEPPOL Business Message Envelope (SBDH)

```
<?xml version="1.0" encoding="UTF-8"?>
<StandardBusinessDocument xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader">
    <StandardBusinessDocumentHeader>
        <HeaderVersion>1.0</HeaderVersion>
        <Sender>
        <Identifier Authority="iso6523-actorid-upis">0088:7315458756324</Identifier>
        </Sender>
        <Receiver>
        <Identifier Authority="iso6523-actorid-upis">0088:4562458856624</Identifier>
        </Receiver>
        <DocumentIdentification>
        <Standard>urn:oasis:names:specification:ubl:schema:xsd:Invoice-2</Standard>
            <TypeVersion>2.1</TypeVersion>
            <InstanceIdentifier>123123</InstanceIdentifier>
            <Type>Invoice</Type>
            <CreationDateAndTime>2013-02-19T05:10:10</CreationDateAndTime>
        </DocumentIdentification>
        <BusinessScope>
            <Scope>
                <Type>DOCUMENTID</Type>
                <InstanceIdentifier>
urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##
urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:pepp
```

```
ol4a:ver2.0::2.1</InstanceIdentifier>
          </Scope>
          <Scope>
              <Type>PROCESSID</Type>
<InstanceIdentifier>urn:www.cenbii.eu:profile:bii04:ver1.0</InstanceIdentifier>
          </Scope>
       </BusinessScope>
   </StandardBusinessDocumentHeader>
   <Invoice
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-
2" xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2">
       <cbc:UBLVersionID>2.0</cbc:UBLVersionID>
       <cbc:CustomizationID
schemeID="PEPPOL">urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:#urn:www.pepp
ol.eu:bis:peppol4a:ver1.0</cbc:CustomizationID>
       <cbc:ProfileID>urn:www.cenbii.eu:profile:bii04:ver1.0</cbc:ProfileID>
       <cbc:ID>008660-AB</cbc:ID>
       <cbc:IssueDate>2011-05-10</cbc:IssueDate>
       <cbc:InvoiceTypeCode>380</cbc:InvoiceTypeCode>
<!-- reduced instance file -->
   </Invoice>
</StandardBusinessDocument>
```