Specification

OpenPEPPOL AISBL

Transport Infrastructure
Coordinating Community

ICT - Models

PEPPOL Transport Infrastructure AS4 Profile

Authors:
Sander Fieten, Chasquis Consulting
Philip Helger, Bundesrechenzentrum

Version: 1.0

Status: In use

Revision history

Version	Date	Author	Organisation	Description
0.1a	20170815	Sander Fieten	Chasquis Consulting	Initial draft version
0.1b	20170901	Sander Fieten	Chasquis Consulting	Updated draft. Initial text inserted in section 3
0.1c	20170908	Sander Fieten Philip Helger	Chasquis Consulting Bundesrechenzentrum	Updated draft based on review. Added text to section 3.
0.1d	20170925	Sander Fieten	Chasquis Consulting	Completed section 3. Updated introduction chapter Updated references
0.1e	20170929	Sander Fieten	Chasquis Consulting	Added line numbers
0.9	20171204	Sander Fieten Philip Helger	Chasquis Consulting Bundesrechenzentrum	Update based on review comments. Main changes: Support for AS4 Compression Feature required, but use optional. Use of AS4 message level security only, no transport level security Clarified usage of Service and Action
1.0	20171208	Sander Fieten	Chasquis Consulting	TICC CMB approved version for publication

Contributors

Sander Fieten, Chasquis Consulting, sander@chasquis-consulting.com
Philip Helger, Bundesrechenzentrum, philip.helger@brz.gv.at

Table of contents

1 Introduction

This specification is designed to facilitate becoming a compliant AS4 Access Point in the PEPPOL eDelivery Network governed by the OpenPEPPOL Association. The goal is to create an easy to use “connect once, connect to all” network to facilitate cross-border trade. The OpenPEPPOL Association is comprised of public and private members of the PEPPOL community (see http://peppol.eu) and is responsible for PEPPOL BIS (Business Interoperability Specifications), building blocks and services. Throughout this document the word PEPPOL refers to both the community and the association involved in these responsibilities.

In September 2016 PEPPOL signed a *Letter of Understanding* with the European Commission in which they agreed on a process how and conditions for migration of the message exchange protocol of the PEPPOL eDelivery Network from the current AS2 protocol to the AS4 protocol, which was chosen by the Commission as standard in their eDelivery architecture to foster the development of the Digital Single Market. The e-SENS large scale project successfully tested the use of the AS4 message exchange protocol in several business domains, including eProcurement. Several profiles of specifications, including AS4, SMP and BDXL[1] were created in the e-SENS project providing guidelines on implementation. PEPPOL agreed with the European Commission to use these e-SENS profiles as the basis for their next generation specifications for the PEPPOL eDelivery Network. Therefore, these profiles need to be adapted as PEPPOL specifications. This document is the PEPPOL specification for the message exchange between Access Points using the AS4 protocol.

1.1 Objective

This document describes a specification to be used to exchange business messages between Access Points (Access Point) part of the PEPPOL eDelivery Network. It uses the AS4 specification as specified by OASIS and the profile created thereof by the e-SENS project. This specification will show how these systems can be enhanced by using the PEPPOL Service Metadata Lookup (SML) and Provider (SMP), based on the appropriate BUSDOX specifications, to dynamically exchange various message transmission parameters such as Certificates to use for message level security and Endpoint URLs and therefore automate the inclusion of new or modified Access Points.

AS4 provides a transport infrastructure for exchanging any business data securely using the HTTP transfer protocol. In the PEPPOL eDelivery network this exchange currently consists of one Standard Business Document XML as specified in the **[TIA-AP-PROV]**, the AS4 protocol however allows to exchange any other format including multi-part business documents.
This specification therefore does not prescribe or restrict the use of any specific business document format. The PEPPOL Business Interoperability Specifications (BIS) specify which business documents are used within the different PEPPOL domains and they should also specify any messaging protocol specific bindings.

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, providing transport for e-procurement messages for both pre and post award scenarios as specified in the PEPPOL BIS.

[1] The BDXL OASIS standard is an enhancement of the PEPPOL SML specification to locate the service meta-data provider of a participant.

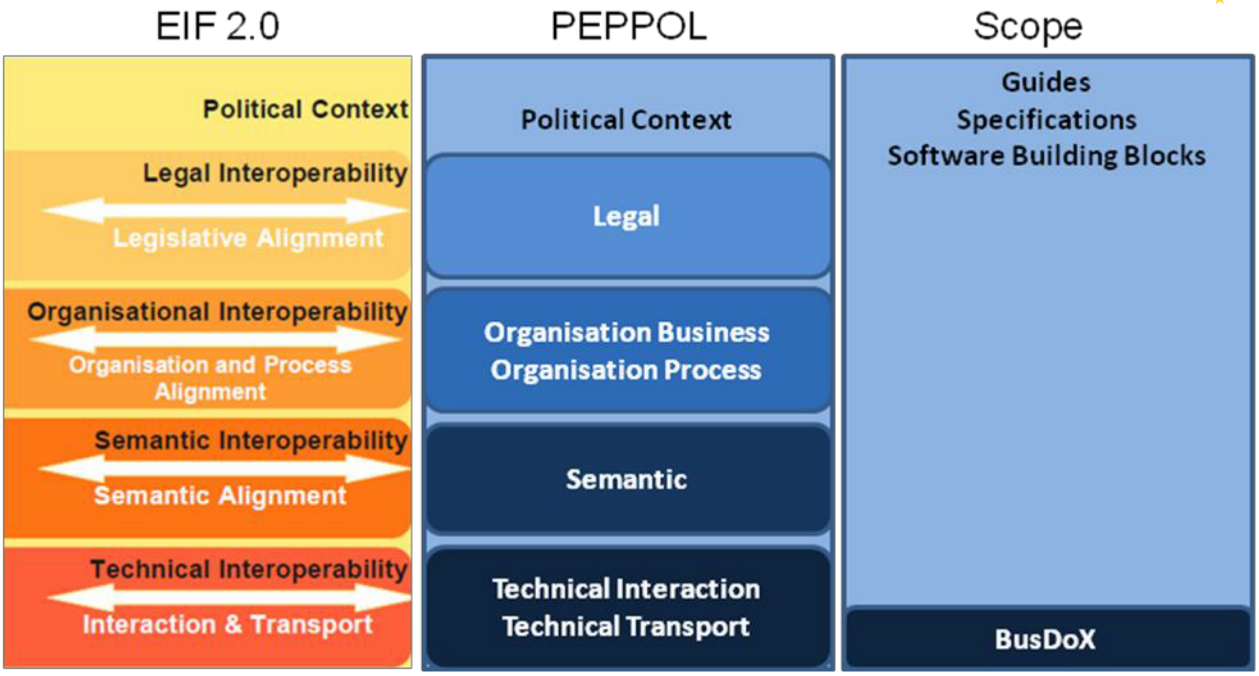

Figure 1: The EIF 2.0 model and the relation to the PEPPOL specifications

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.
- Clearly state the transport level requirements in a single document.
- Identify the additional steps required to update an existing AS4 system so it complies with the requirements and can therefore participate as a PEPPOL compliant Access Point.
- Define a simple, interoperable, reliable and safe communications pattern that Access Points can use to communicate.
- Define the message exchange formats and patterns clearly.
- Ensure that messages are reliably delivered between Access Points, including providing the prerequisites for logging and proof-of-delivery for messages at the transport level
- Ensure confidentiality during the exchange by using message level encryption using AS4 encryption.
- Ensure integrity and authenticity of received messages. This is maintained by using the ebMS security features, which are used to digitally sign, digest and authenticate 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.
- Support all pre- and post-award message exchanges.

The Profile does NOT address:

- The verification of certificates, format of participant identifiers, and other details required to create a full instantiation of PEPPOL.
- The format of business documents, e.g. use of SBDH, ASiC, etc.
- The communication protocol with PEPPOL Service Metadata Provider services.
- Retrieval/exchange of metadata required for the business document exchange

1.3 Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.4 Normative references

[RFC2119] *Key words for use in RFCs to Indicate Requirement Levels*, IETF RFC, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC7230] *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*, IETF RFC, June 2014, https://tools.ietf.org/html/rfc7230

[AS4-Profile] *AS4 Profile of ebMS 3.0 Version 1.0*. 23 January 2013, OASIS Standard, http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/os/AS4-profile-v1.0-os.html

[PEPPOL-SMP] *PEPPOL Transport Infrastructure Service Metadata Publishing (SMP)*, Version 1.1.0, 15 August 2012, https://github.com/OpenPEPPOL/peppol-eia/raw/master/1-ICT_Architecture/1-ICT-Transport_Infrastructure/13-ICT-Models/ICT-Transport-SMP_Service_Specification-110.pdf

[PEPPOL-ID-POL] *PEPPOL Transport Infrastructure Policy for use of Identifiers*, Version 3.0, 3 February 2014, https://github.com/OpenPEPPOL/documentation/raw/master/TransportInfrastructure/PEPPOL_Policy%20for%20use%20of%20identifiers-300.pdf

[ebMS3CORE] *OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features*, 1 October 2007, OASIS Standard, http://docs.oasis-open.org/ebxml- msg/ebms/v3.0/core/ebms_core-3.0-spec.pdf

[WSS111-X509] *Web Services Security X.509 Certificate Token Profile Version 1.1.1*, 18 May 2012, OASIS Standard, http://docs.oasis-open.org/wss-m/wss/v1.1.1/os/wss-x509TokenProfile-v1.1.1-os.pdf

[XML-DSIG1] *XML Signature Syntax and Processing Version 1.1*, 11 April 2013, W3C Recommendation, http://www.w3.org/TR/xmldsig-core1/

[XML-ENC1] *XML Encryption Syntax and Processing Version 1.1*, 11 April 2013, W3C Recommendation, http://www.w3.org/TR/xmlenc-core1/

1.5 Non-normative references

[SML] *PEPPOL Transport Infrastructure Service Metadata Locator (SML)*, Version 1.01, 10 October 2010, https://github.com/OpenPEPPOL/peppol-eia/raw/master/1-ICT_Architecture/1-ICT-Transport_Infrastructure/13-ICT-Models/ICT-Transport-SML_Service_Specification-101.pdf

[PEPPOL-AS2] *PEPPOL Transport Infrastructure AS2 Profile*, Version 1.00, 9 December 2013, https://github.com/OpenPEPPOL/peppol-eia/raw/master/1-ICT_Architecture/1-ICT-Transport_Infrastructure/13-ICT-Models/ICT-Transport-AS2_Service_Specification-100.pdf

[BUSDOX-CDF] *PEPPOL Transport Infrastructure BusDox Common Definitions, Version 1.01, 1 October 2010, https://github.com/OpenPEPPOL/peppol-eia/blob/master/1-ICT_Architecture/1-ICT-Transport_Infrastructure/13-ICT-Models/ICT-Transport-BusDox_Definitions-101.pdf*

[XML-DSIG] *XML Signature Syntax and Processing*, 10 June 2008, W3C Recommendation, http://www.w3.org/TR/xmldsig-core/

[XML-ENC] *XML Encryption Syntax and Processing,* 10 December 2002, W3C Recommendation, http://www.w3.org/TR/xmlenc-core/

[eSENS-SAT] *e-SENS eDelivery Solution Architecture Template (SAT - eDelivery)*, version 1.8, 30 December 2016, http://wiki.ds.unipi.gr/display/ESENS/SAT+-+eDelivery

[eSENS-AS4] *e-SENS AS4 Profile (PR - AS4)*, version 1.11, 30 December 2016, http://wiki.ds.unipi.gr/display/ESENS/PR+-+AS4+-+1.11

[TIA-AP-PROV] *PEPPOL Transport Infrastructure Agreements - PEPPOL AP Provider Agreement*, Version 3.0, 18 June 2012, included in https://github.com/OpenPEPPOL/documentation/raw/master/TransportInfrastructure/TIA-PA-AP-SMP-web-watermarked.zip

[BDXL] *Business Document Metadata Service Location Version 1.0*, OASIS Standard, 01 August 2017, http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.html

[OASIS-SMP] *Service Metadata Publishing (SMP) Version 1.0,* OASIS Standard, 01 August 2017, http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/bdx-smp-v1.0.html

2 Overview and Context (Non-Normative)

2.1 The PEPPOL architecture

To fulfill the “connect once, connect to all” principal the PEPPOL eDelivery Network uses a four-corner architecture where participants use an Access Point of their choice to connect to the network and which then takes care of the message exchange with the participant’s business partner (through the Access Point chosen by the partner). An Access Point provider may offer additional services to their customers beside the required eDelivery service.

Within the PEPPOL eDelivery network *service location* and *capability lookup* building blocks are used to dynamically setup the communication between Access Points. Based on the metadata of the document to send the sending Access Point will determine the destination Access Point service and setup the messaging configuration based on the data retrieved from the capability lookup (SMP lookup).

As the service capability metadata includes the message exchange protocol(s) supported it allows for use of different message exchange protocols in different document exchanges. Note that this document only specifies how an Access Point must integrate the AS4 message exchange protocol into its service offering but that the **[TIA-AP-PROV]** may also require support for other message exchange protocols (like at the time of writing of this document AS2).

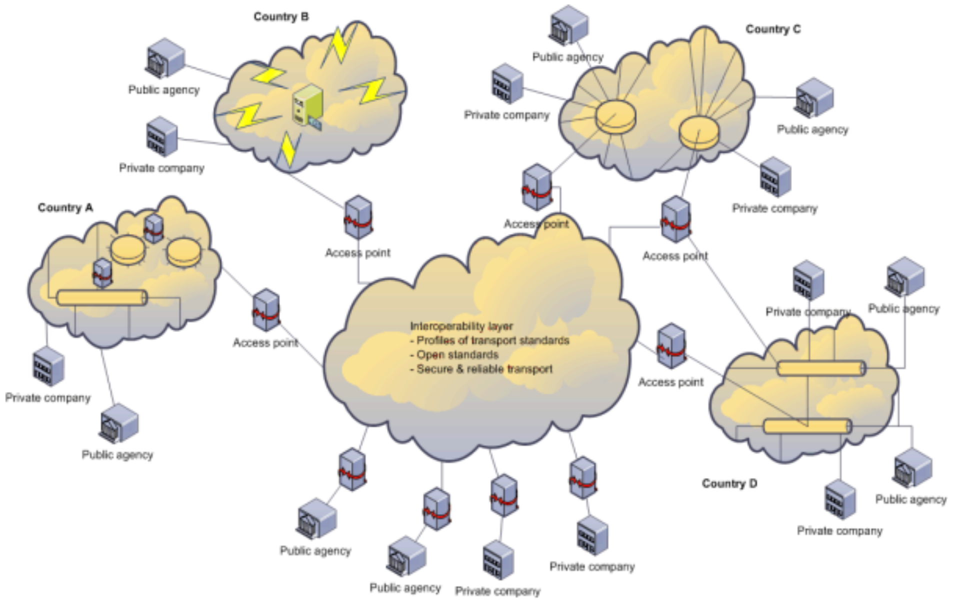

Figure 2: Overview of the PEPPOL eDelivery Network

The architecture of the PEPPOL eDelivery Network was also used as input to the e-SENS large scale project. This resulted in the **[eSENS-SAT]** using the four-corner model as well, however leaving different options for the messaging configuration between Access Points. For the dynamic configuration of the message exchange between Access Points there is also a difference between the **[eSENS-SAT]** and the PEPPOL specifications as in e-SENS the (newer) **[BDXL]** and **[OASIS-SMP]** specifications are used for *service location* and *capability lookup* whereas in PEPPOL **[SML]** and **[PEPPOL-SMP]** are used. Since the OASIS specifications are based on the work done earlier in PEPPOL the specifications are very similar but not completely backwards compatible. Therefore, the e-SENS profiles of specifications cannot directly be reused in PEPPOL.

2.2 The ebMS / AS4 messaging model

As **[AS4-Profile]** is a profile of ebMS version 3, it uses the messaging model described in section 2 of **[ebMS3CORE]**. This abstract model, shown in figure 3, defines how business documents are exchanged between two business partners and what is in scope of the ebMS specifications.

In it there is a clear separation of concern between the components responsible for processing the business data, the business applications, and the components responsible for the execution of the actual message exchanges, called *Message Service Handlers*, or MSH for short. This strict separation between business and messaging functionality allows to make the MSH available as standard off-the-shelf software making it easier to add the AS4 message exchange protocol to a solution. This is the same concept as the Access Points already used in the PEPPOL eDelivery Network for many years.

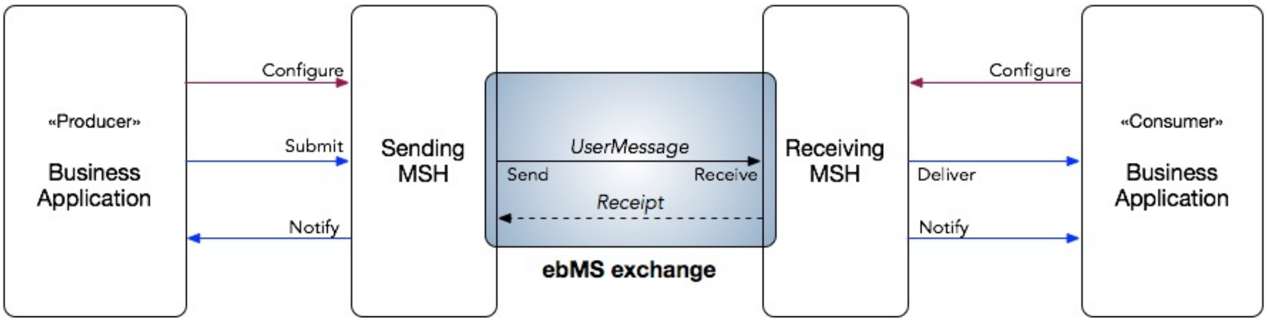

Figure 3: The abstract messaging model of an ebMS message exchange.

There are five abstract operations defined in the model of which only the *Send* and *Receive* operation are in scope of the ebMS specifications. The other three, *Submit*, *Deliver* and *Notify*, which apply to the integration of the MSH with the business application, are out of scope for the ebMS specification and are therefore implementation specific. An operation not explicitly defined in the model but required to allow useful deployment of an MSH, is *Configure* which allows one to setup the MSH for the message exchange with the partner. It will therefore be in these abstractly defined operations where implementations will differ and which will be important when integrating an MSH into a complete solution.

For the configuration of the message exchanges between two MSHs ebMS version 3 also uses an abstract concept, called P-Modes, short for *processing modes*. A P-Mode, described in section 4 of **[ebMS3CORE]**, is a set of parameters that each specify a specific detail of a message exchange, for example the identifiers of the sender and receiver of a message and the algorithm used for signing a message. When parties are going to set up a message exchange they need to agree on the P-Mode(s) to use.

To facilitate P-Mode creation and to ensure interoperability between parties, *profiles* can be created to predefine a set of values for certain P-Mode parameters. The OASIS technical committee responsible for the ebMS Specification have already created such a profile with **[AS4-Profile]**. It however still has a lot of options one can choose from when setting up the message exchange. Therefore, it is common that within a domain further profiling takes place to specify in detail how the message exchanges should be executed.

In the e-SENS project a more detailed profile of **[AS4-Profile]** was developed (see **[eSENS-AS4]**) that specifies the packaging of business data in the messages and how to secure the message exchange based on the requirements gathered across business domains part/target of the Digital Single Market initiative.

This document is the PEPPOL AS4 profile and specifies how Access Points in the PEPPOL eDelivery Network must configure their P-Modes. It builds on **[eSENS-AS4]** and tailors it to the specific requirements of the PEPPOL eDelivery Network.

2.3 A typical workflow

As described above the **[AS4-Profile]** only specifies how the communication between two MSHs should work based on a given P-Mode but does not prescribe how that P-Mode should be created. The PEPPOL AS4 profile (this document) defines how Access Points in the PEPPOL eDelivery Network should setup their P-Modes to exchange business documents when using AS4 as message exchange protocol. By specifying how the Access Points must create their P-Modes the interoperability in the PEPPOL eDelivery Network is ensured.

For the specification of the PEPPOL AS4 profile the Access Point is considered as one, accepting and delivering business documents from/to the connected participants and exchanging them between Access Points using AS4. How the MSH and the component(s) - containing the functionality to receive and deliver business document from and to the participants - are integrated, is out of scope for the PEPPOL AS4 profile and left up to the Access Point provider/implementer.

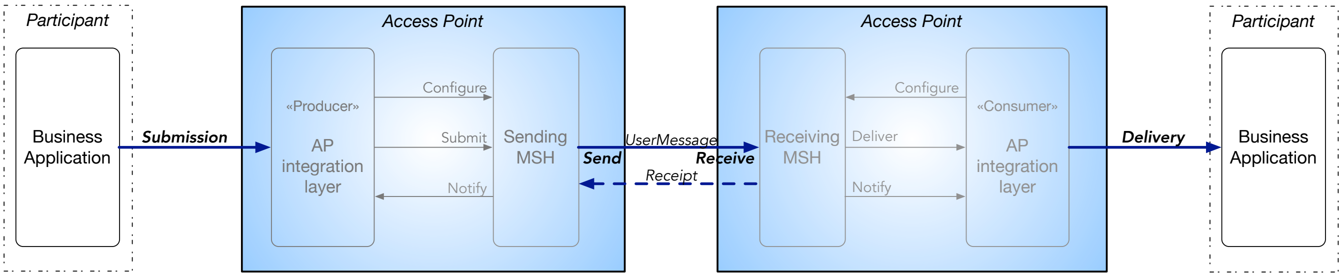

Figure 4: The scope of the PEPPOL AS4 profile is the Access Points as a whole without looking at their internal structure.

Figure 5 shows an example workflow in the PEPPOL eDelivery Network to setup and execute the exchange of business documents between participants. It starts with the participant able to receive a business document to register this new capability at their selected Access Point provider (1). The provider will then register this new document in the SMP (2) so that other Access Points are able to setup their messaging configuration to send these documents to the recipient's Access Point. Furthermore, the Access Point needs to ensure a P-Mode is configured so it can receive the new business document on behalf of the participant (3). When these steps have been performed the participant is successfully connected to the network and able to receive the business document from any other participant.

On the sending side the process starts with the participant submitting the business document it wants to send to its Access Point (4)[2]. After receiving the business document from the participant the sender's Access Point will use the DNS to determine which SMP server provides the metadata needed to setup the message exchange with the Access Point of the recipient (5) and (6). The next step is to query the SMP server for the metadata applicable to the business document to exchange (7) and use the SMP result (8) to create the correct P-Mode (configure the Access Point) for sending the document to the recipient's Access Point (9). At this point the message exchange can be executed between the two Access Point (10, 12). As a last step in the message exchange the sending Access Point informs the sender of the message about the result of the message

[2] Note that a service provider may also offer as an additional service to create/transform the business document in which case the participant provides only the relevant data to construct the business document.

exchange. How the sender is informed is out of the scope of this profile. It can for example be done by sending a notification to the participant (13).

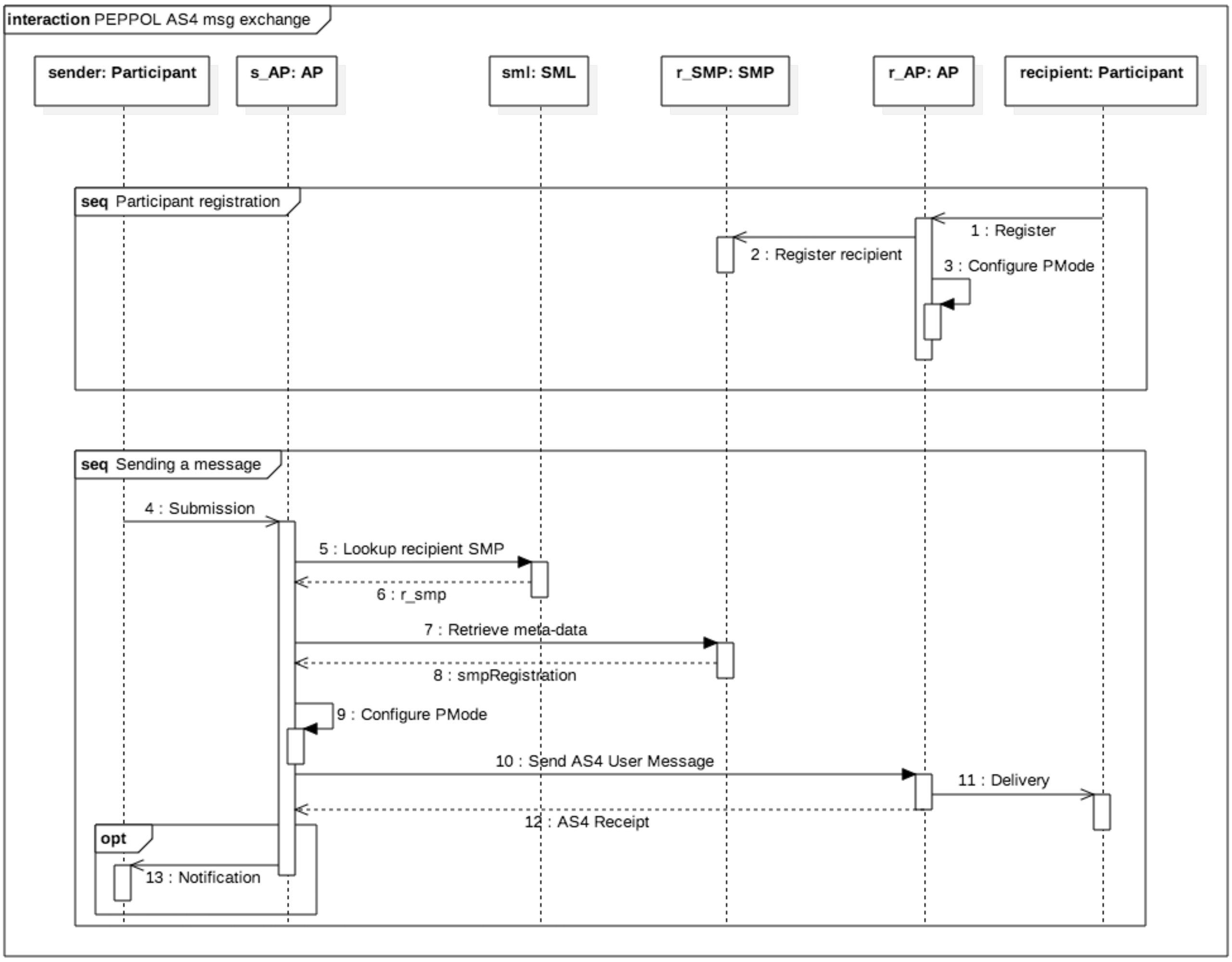

Figure 5: Sequence diagram of a typical workflow for a message exchange in the PEPPOL eDelivery Network.

Note that in the diagram presented here the business document is delivered to the recipient (11) before the AS4 Receipt is sent back to the sender's Access Point (12) but that this is not required by neither **[AS4-Profile]** nor this profile. As in the four-corner model the Access Point provider is however assumed to be acting on behalf of its connected participants and the business document is still considered to be delivered to the participant when it is successfully received by the Access Point. See also section 3.5 of this profile for more information on the semantics of the AS4 Receipt.

3 Specification Profile Details

3.1 Baseline

The PEPPOL AS4 profile is based on **[eSENS-AS4]** which was developed and tested in a 4-corner model by the e-SENS project. This means it will use the same profile settings where applicable and define specific settings when required for use in the PEPPOL eDelivery Network.

Therefore, the normative baseline of this profile is the *AS4 ebHandler Conformance Clause* as specified in section 6.1 of **[AS4-Profile]**. This conformance clause includes required support for both the Push and Pull message exchange patterns, but as only the One-Way Push message exchange pattern is used in the PEPPOL eDelivery Network, these requirements are relaxed and Access Points are NOT REQUIRED to support the One-Way Pull pattern.

3.2 Message packaging

As defined in section 5 of **[ebMS3CORE]** the payloads of an ebMS User Message may be contained in either the SOAP Body or separate MIME attachments[3]. Since this profile however uses the AS4 Compression Feature (see below) which applies only to payloads packaged in attachments the Access Point MUST include all payloads as MIME attachments.

> NOTE: When sending large messages an Access Point MAY use the http chunked transfer encoding to enable more streamlined processing. As specified in section 4.1 of **[RFC7230]** Access Points MUST support this encoding when receiving messages.

The "Content-Disposition" MIME header as described in section 5.1.9 of **[AS4-Profile]** SHALL NOT be used to exchange the filename of an attached payload. If the exchanged business document consists of multiple parts that need to be identifiable to enable cross referencing between parts a *Part Property* with name *PartId* MUST be used (see also issue 52 registered with the OASIS ebMS TC). The actual identifier must be provided by the application that composes the multi-part business document.

> NOTE: This does not imply that an Access Point cannot include this header in the MIME message, but only that it should not be used to identify the payload and a receiving Access Point MAY ignore the header.

The AS4 Compression Feature as specified in section 3.1 of **[AS4-Profile]** MUST be supported and it is RECOMMENDED to be used, i.e. **PMode[1]. PayloadService.CompressionType** SHOULD be set to *application/gzip*. As described in **[AS4-Profile]** it is not required to compress payloads that are already in a compressed format. This means that an Access Point MUST NOT reject a received message that contains uncompressed payloads even if **PMode[1].PayloadService.CompressionType** has value *application/gzip*. Because the payloads are already compressed either natively or using the AS4 Compression Feature the http compression encoding (see **[RFC7230]** chapter 4.2) on the transport layer SHALL NOT be used.

[3] The option to use to SOAP Body for including the payload only applies to XML payloads. The specification does not provide any statements on including non-XML payloads in the SOAP Body.

3.3 ebMS User Message metadata

The message partition channel feature as defined in **[ebMS3CORE]** is not needed for the message exchanges between the Access Points in the PEPPOL eDelivery Network. Therefore the default MPC is used, i.e. **PMode[1].BusinessInfo.MPC** MUST be set to:

http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC

> NOTE: Because the default MPC is used the `eb3:UserMessage/@mpc` attribute MAY be omitted in the ebMS message header.

Because the AS4 User Message is only exchanged between the Access Points these should be identified as sender and receiver. As in **[PEPPOL-AS2]** the *Subject CommonName* (CNAME) of the PEPPOL Access Point Certificate issued to the Access Point is used as identifier.
The *Sending* Access Point MUST set both the **PMode.Initiator.Party** and **PMode.Responder.Party** parameters and MUST use the certificate registered in the SMP for the AS4 endpoint of the *Receiving* Access Point (`//smp:Endpoint[@transportProfile="peppol-transport-as4-v1_0"]/smp:Certificate`)[4] to retrieve the correct identifier for **PMode.Responder.Party**.
As the *Receiving* Access Point does not know beforehand from which other Access Point it will received messages it MUST only set the **PMode.Responder.Party** parameter[5] to the *Subject* CNAME of its PEPPOL Access Point Certificate. Both the *Sending* and *Receiving* Access Point MUST use *urn:fdc:peppol.eu:2017:identifiers:ap* as value for the **PartyId.type** parameter.

The `//smp:Endpoint[@transportProfile="peppol-transport-as4-v1_0"]/wsa:EndpointReference/wsa:Address` element from the SMP registration retrieved for the business document to send MUST be used [by the *Sending* Access Point] as value for **PMode[1].Protocol.Address**.

As the message exchange between two Access Points in the PEPPOL eDelivery Network is based on the **[TIA-AP-PROV]** the **PMode.Agreement** parameter which is used to indicate the business agreement that governs the message exchange MUST have value *urn:fdc:peppol.eu:2017:agreements:tia:ap_provider* without type attribute. The reference to the agreement is included in the `eb3:AgreementRef` element of the ebMS messaging header. This element also includes an optional attribute `pmode` which can be used to include the **PMode.ID**. This attribute MUST NOT be used as Access Points may use just one generic P-Mode for receiving messages (see below).

When sending the business document the Access Point MUST set **PMode[1].BusinessInfo.Service** to the PEPPOL process identifier as specified in the PEPPOL BIS. The **PMode[1].BusinessInfo.Service.type** MUST be set to the fixed value *urn:fdc:peppol.eu:2017:identifiers:proc-id*. The Service value MUST be formatted as follows (similar to the generic URL formatting defined in **[BUSDOX-CDF]**): «scheme identifier»*::*«process identifier value». The values for scheme and process identifier SHALL NOT use URL percent encoding.
PMode[1].BusinessInfo.Action MUST be set to business document's encoded document type identifier as defined in the PEPPOL BIS. The document type identifiers MUST be formatted as specified in **[PEPPOL-ID-POL]**.

[4] The receiving Access Point provider is responsible for the registration of the required meta-data, see section 3.7

[5] In the P-Mode at least the Initiator or Responder needs to be defined but it isn't necessary to define both. This way more "generic" P-Modes can be created that can accept/send to multiple partners.

Note that these meta-data are also used for querying the SMP and therefore the values of these P-Mode parameters match the values of the SMP registration. How the Access Points gets these meta-data, i.e. whether they are provided by the participant or derived from the submitted business document (e.g. from the SBDH) is out of scope of this specification and left to the Access Point provider.

Receiving Access Points MUST ensure that they have configured one or more P-Modes so they can receive messages for all combinations of document type and process (including scheme) identifiers referenced by AS4 endpoints (i.e. `transportProfile` attribute has value *peppol-transport-as4-v1_0*) that they have registered in the SMP. Note that an Access Point MAY use a “generic” P-Mode to receive the registered business documents. Such a generic P-Mode only defines the parameters related to the Access Point itself but no business document specific ones.

Using the `eb:ConversationId` and `eb:MessageProperties` elements in the ebMS message header additional meta-data about the exchanged business document can be included in the AS4 message. This can simplify processing of the business document as the contents don’t need to be read and parsed. A PEPPOL BIS should specify how these elements are to be used in specific transactions.

As the `eb:ConversationId` element is required it must always have a value. If no value is included in the submission of the business document to the Access Point, the Access Point MUST set the value of `eb:ConversationId` to “1” as specified in section 4.3 of **[ebMS3CORE]**.

> NOTE: Since these information elements are part of the ebMS header which is not encrypted using the ebMS message level encryption they should not be used to exchange confidential information.

3.4 Error handling

When an Access Point detects an error in a received message the resulting ebMS Error must be send back synchronously as a response, i.e. **PMode[1].ErrorHandling.Report.AsResponse** MUST be set to *true*. As described in the ebMS specifications a SOAP Fault may be added to the message when reporting an ebMS Error back to the sender. To reduce interoperability risks however Access Points SHOULD NOT add the SOAP Fault when reporting errors. It is RECOMMENDED that errors generated for received message are reported to an Access Point operator so the problem may be resolved out of band[6].

When receiving a business document the Access Point will need to check whether it services the addressed participant to be able to deliver the message. When a MSH allows to execute custom validations of the content of a User Message during the ebMS message processing, it is RECOMMENDED that the Access Point includes the check on the addressee and generates and sends back an ebMS Error in case the addressed participant is not serviced by the Access Point. The `errorCode` attribute of the generated Error MUST be set to *EBMS:0004* (Other error) and its `severity` attribute MUST be set to *failure*. Furthermore the `errorDetail` attribute MUST have value *PEPPOL:NOT_SERVICED* to indicate that the addressed participant is not serviced by the Access Point.

[6] This could be implemented by having the MSH component of the Access Point notify the integration layer by setting the **PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** to *true.* Support for this parameter is however not required by the ebHandler Conformance Clause and therefore the profile only requires logging of the error.

Receiving Access Points MUST either handle the error internally or notify the participant that submitted the message in error about a received ebMS Error, i.e. **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer** SHOULD be set to *true*.

When reporting an Error back to the sender of the erroneous message it MUST be signed when a P-Mode can be assigned to the received message. Access Points MUST however accept both unsigned as well signed Errors as for some errors it may not be possible to relate them to a P-Mode (which defines the signing certificate to use). When the Error message is signed an Access Point MAY validate the signature, but SHOULD report errors only locally, i.e. not respond with an Error message.

3.5 Reliability

For the participants in the PEPPOL eDelivery Network it is important to get assurance about the successful delivery of the business documents they send to their trading partner. When using AS4 as the message exchange protocol, the *Reception Awareness* feature of **[AS4-Profile]** is used to ensure message delivery and provide acknowledgement of reception. This feature uses the Receipt signal message to ensure that a Sending MSH is aware of whether or not a sent User Message is successfully received by the Receiving MSH.

It must be noted that by default the Receipt does not necessarily mean that the message is successfully delivered to the participant, as the MSH is not required to ensure delivery to the business application before sending the Receipt, see also section 3.4 of **[AS4-Profile]**. In the context of the PEPPOL eDelivery Network this distinction however is not relevant as the Access Point acts on behalf of the participants it services and therefore successful receipt of the message by the Access Point also indicates that business document is or will be delivered to the addressed participant. Due to misconfiguration of the SMP registration however, an Access Point may in exceptional cases receive a message for a participant that it doesn’t service. If this is not detected during the ebMS message processing (as described in the previous section) the Access Point Provider MUST ensure that this error is handled out of band.

To ensure that Receipts are sent Access Points MUST use the AS4 Reception Awareness feature, therefore **PMode[1].ReceptionAwareness** MUST be set to *true*, and **PMode[1].Security.SendReceipt.ReplyPattern** MUST have value *Response*. Use of this feature implies that Receipts must be sent for received messages, i.e. **PMode[1].Security.SendReceipt** MUST be *true*. It is RECOMMENDED to send the Receipt signal after the *Deliver* operation has been successfully completed.

Some business transactions require that a business level acknowledgement contains the exact timestamp when the acknowledged message was received. As the Access Point act on behalf of the participant this is the time that the Access Point has successfully received the message, which in case of AS4 is the moment that the Receipt is created. Therefore Access Point MUST be able to provide the meta-data of a created Receipt to the participant’s business application.

Furthermore the retry function of the AS4 Reception Awareness feature SHOULD be used to increase reliability of the message exchanges, i.e. **PMode[1].ReceptionAwareness.Retry** MUST be set to *true*. Which retry parameters should be used depends on the business document that is exchanged and are therefore not profiled.

When no Receipt is received within the configured time window, the sending Access Point MUST inform the participant that submitted the business document about the delivery failure. The time to wait for a Receipt and the way to inform the participant are out of scope of this profile and left to the specific configuration of the

Access Point. To enable the business application to take action when message delivery to its destination is not confirmed, the *MissingReceipt* error MUST be reported to the business application, i.e. **PMode[1].ErrorHandling.Report. MissingReceiptNotifyProducer** MUST be set to *true*.

The duplicate detection and elimination function of the Reception Awareness feature (see section 3.2 of **[AS4-Profile]**) MUST be used by the Access Point: **PMode[1]. ReceptionAwareness.DuplicateDetection** MUST be set to *true* and detected duplicates SHOULD NOT be delivered to the participant. It is RECOMMENDED that Access Points check the complete message log for duplicates, but they MUST check at least the last seven days.

3.6 Security

As this profile's scope is limited to the AS4 message exchange between the Access Points the security requirements specified only apply to the communication between the Access Points. It is the responsibility of the Access Point provider and its connected participants to ensure that the information is also sufficiently secured during the communication between Access Point and participant.

It also implies that additional security measures may need to be undertaken on the business level, i.e. on the business documents exchanged between the business applications of the participants, depending on the requirements of the business domain. An example is the encryption of tenders in the pre-award domain. Such business level security is outside the scope of this profile and should be specified in the respective PEPPOL BIS.

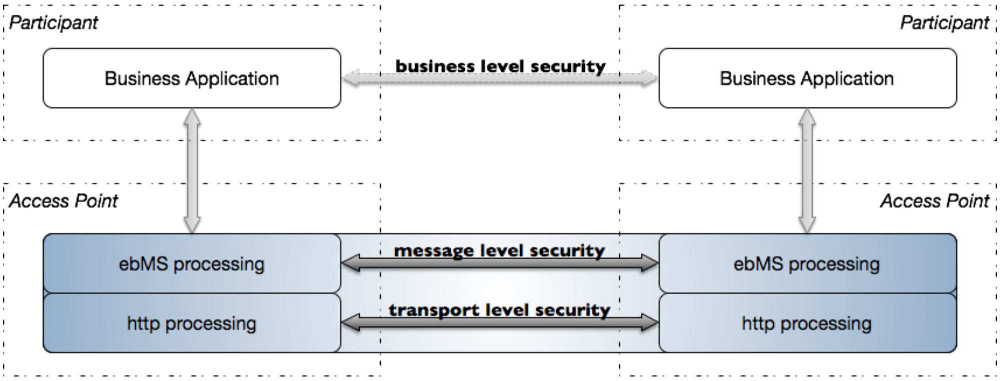

Figure 6 The scope of this profile is limited to the message level and transport level security.

3.6.1 Transport level security

As shown in figure 6 both transport and message level security are used to secure the message exchange between the Access Points. Since the message level security already provides the security features also provided by the transport level security it is not necessary to also apply transport level security between the Access Points. Therefore Access Points SHALL NOT use transport level security (TLS). Note that transport security must currently be applied when using the AS2 message exchange protocol as **[PEPPOL-AS2]** does not provide all required security features (i.e. encryption) on the message level.

3.6.2 Message level security

The ebMS security features are used to protect the confidentiality and integrity of the exchanged information and to ensure non-repudiation of receipt. This means that AS4 User Messages MUST be both signed and encrypted to protect the integrity and confidentiality of the business documents. As specified in section 5.1.8 of **[AS4-Profile]** the Access Point MUST acknowledge received User Messages using a signed non-repudiation Receipt which contains the digest of the payloads of the original message.

Both **[ebMS3CORE]** and **[AS4-Profile]** reference the WS-Security version 1.1 specifications. The cryptographic algorithms included in this version (through reference to **[XML-DSIG]** and **[XML-ENC]**) however are not up to date anymore as weaknesses have been discovered and their use is discouraged. Therefore implementations used within the PEPPOL eDelivery Network MUST support the newer algorithms as specified in **[XML-DSIG1]** and **[XML-ENC1]**, more specifically SHA-256 must be supported for signing messages and AES128-GCM for encryption. The table below provides a complete specification of the algorithms to use for signing and encryption:

P-Mode parameter	Profiled value
PMode[1].Security.Signature.HashFunction	Fixed value: *http://www.w3.org/2001/04/xmlenc#sha256*
PMode[1].Security.Signature.Algorithm	Fixed value: *http://www.w3.org/2001/04/xmldsig-more#rsa-sha256*
PMode[1].Security.Encryption. KeyTransportAlgorithm	Fixed value: *http://www.w3.org/2001/04/xmlenc#rsa-oaep*
PMode[1].Security.Encryption. KeyTransportAlgorithmParameters[7]	The following values MUST be used for the key transport parameters: • Mask generation function: *http://www.w3.org/2009/xmlenc11#mgf1sha256* • Digest generation function: *http://www.w3.org/2001/04/xmlenc#sha256*
PMode[1].Security.Encryption.Algorithm	Fixed value: *http://www.w3.org/2009/xmlenc11#aes128-gcm*

The certificate issued by the PEPPOL PKI to an Access Point provider MUST be used for signing the ebMS message. It MUST be included as a *Binary Security Token* as specified in **[WSS111-X509]**, i.e. **PMode[1].Security.Signature.X509TokenReferenceType**[8] MUST have value *BinarySecurityToken*. This allows the receiving Access Point to validate the certificate using the PEPPOL PKI CA certificates without the need to know the certificate of the sending Access Point beforehand.

When sending the User Message the Access Point MUST use the certificate as included in the SMP registration retrieved based on the submitted business document, i.e. **PMode[1].Security.Encryption.Certificate** MUST be set to `//smp:Endpoint[@transportProfile="peppol-transport-as4-v1_0"]/smp:Certificate`. Since the receiving Access Point already knows its certificate it doesn't need to be included in the message and profiling of the reference method is not required. This implies that Access Point MUST support all three methods as specified in **[WSS111-X509]** when receiving messages.

3.7 SMP transport profile identifier

The previous sections described how the AS4 message exchange must be setup and executed between two Access Points in the PEPPOL eDelivery Network. As described in section 2.3 and shown in figure 5 the receiving Access Point must register the meta-data on the document types it can receive in the SMP to enable the sending Access Point to setup the P-Modes required to execute the message exchange. **[PEPPOL-SMP]** specifies which meta-data the receiving Access Point must register in the SMP. To indicate that the Access Point is able to

[7] As described in issue 45 registered with OASIS ebMS TC the parameters **PMode[1].Security. Encryption. KeyTransportAlgorithm** and **PMode[1].Security.Encryption.KeyTransportAlgorithmParameters** are not defined in **[ebMS3CORE]** but are needed for a complete configuration of the MSH.

[8] As noted in issue 69 as registered in the OASIS ebMS TC's issue tracker this P-Mode parameter is not defined in **[ebMS3CORE]** but is needed for a complete configuration of a MSH.

receive the registered business document using this profile of the AS4 message protocol it MUST add an `smp:Endpoint` element with a *`transportProfile`* attribute having value *peppol-transport-as4-v1_0* to the SMP registration of the business document.

Appendix A P-Mode parameter overview

This appendix provides an overview of all P-Mode parameters for which the PEPPOL AS4 profile prescribes what values to use. The parameters are grouped in the same way as in section 2.1.3 of **[AS4-Profile]** that specifies which P-Mode parameters must be supported by an MSH conforming to the *ebHandler Conformance Clause* (the baseline for this PEPPOL AS4 profile).

Also shown in the tables below is whether the parameter is also profiled in **[eSENS-AS4]** and whether the same or a different value is used, indicated by a ✓ respectively ⨯. Note that this overview is provided for information only and that the normative statements in the section 3 take precedence over the values presented here.

A.1 General P-Mode parameters

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode.ID	Not used	
PMode.Agreement	Fixed value: *urn:fdc:peppol.eu:2017:agreements:tia:ap_provider*	
PMode.MEP	Fixed value: *http://www.oasis-open.org/committees/ebxml-msg/oneWay*	
PMode.MEPbinding	Fixed value: *http://www.oasis-open.org/committees/ebxml-msg/push*	
PMode.[Initiator | Responder].Party	One **PartyId** with value the *Subject CNAME* of the PEPPOL Access Point Certificate issued to the Access Point, e.g. APP_1000000100 Fixed value for **PartyId.type**: *urn:fdc:peppol.eu:2017:identifiers:ap*	
PMode.[Initiator | Responder].Role	Fixed value: *urn:fdc:peppol.eu:2017:roles:ap:as4*	
PMode.[Initiator | Responder].Authorization	Not used	✓

NOTE: The receiving Access Point should only set the **PMode.Responder** parameters while the sending Access Point should set both the **PMode.Initiator** and **PMode.Responder** parameters using the certificate retrieved from the SMP.

A.2 PMode[1].Protocol

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].Protocol.Address	URL of the receiving Access Point as registered in the SMP: `//smp:Endpoint[@transportProfile="peppol-transport-as4-v1_0"]/wsa:EndpointReference/wsa:Address`	✗
PMode[1].Protocol.SOAPVersion	Fixed value: *1.2*	✓

NOTE: The XPath expression given here is slightly different from the one specified in **[eSENS-AS4]** because the PEPPOL eDelivery Network uses **[PEPPOL-SMP]** instead of **[OASIS-SMP]** which was used in e-SENS.

A.3 PMode[1].BusinessInfo

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].BusinessInfo.MPC	Fixed value: *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC*	✓
PMode[1].BusinessInfo.Service	The PEPPOL Process identifier of the business document formatted as follows: «scheme id»::«process id value» Example: *cenbii-procid-ubl::urn:www.cenbii.eu:profile:bii01:ver2.0*	✓
PMode[1].BusinessInfo.Service.type	Fixed value: *urn:fdc:peppol.eu:2017:identifiers:proc-id*	✓
PMode[1].BusinessInfo.Action	The encoded PEPPOL Document type identifier of the business document, as registered in the SMP: `//DocumentIdentifier` Example: *busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1*	✓
PMode[1].BusinessInfo.Properties[]	The AP MUST be able to set these properties as specified in the PEPPOL BIS that applies to the business document to send.	✗

NOTE 1: The PEPPOL identifiers used for these P-Mode parameters must be formatted as specified in **[PEPPOL-ID-POL]**.

NOTE 2: **[ebMS3CORE]** does not require setting values for these P-Mode parameters, so one P-Mode could be used to handle exchanges of different business document. Access Points therefore are not required to set the P-Mode parameters in this group explicitly for each business document exchange but must guarantee their P-Mode configuration is setup in such a way that it ensures that the business documents can be exchanged.

NOTE 3: The MPC does not need to be explicitly specified in the P-Modes as this value is assumed to be the default one if no value is given in either P-Mode or message.

NOTE 4: The **PMode[1].BusinessInfo.Properties[]** parameter defines the *Message Properties* that are included in the ebMS header. As explained in appendix D of **[ebMS3CORE]** the value for these properties can also be provided when the business document to send is submitted to the Access Point and therefore don't need to be defined directly in the P-Mode.

A.4 PMode[1].ErrorHandling

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].ErrorHandling.Report.AsResponse	Fixed value: *true*	✓
PMode[1].ErrorHandling.Report.ReceiverErrorsTo	Not used	✓
PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer	Fixed value: *true*	
PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer	Not used	✗
PMode[1].ErrorHandling.Report.MissingReceiptNotifyProducer	Fixed value: *true*	✓

NOTE 1: Although most parameters in this group relate to the integration between the MSH and the Access Point's integration component which in this profile are considered as a whole and strictly speaking therefore would not apply, they are included as these settings will be required to implement error reporting to the participant.

NOTE 2: As noted in issue 59 registered with the OASIS ebMS TC the **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer** parameter does not need to be supported by an AS4 MSH as it is only applicable when reliable messaging as defined in section 8 of **[ebMS3CORE]** is used. This however is not supported in AS4 and therefore support for this parameter is not required. In**[eSENS-AS4]** the parameter however is profiled setting the value to *true*. As the value of **PMode[1].ErrorHandling.Report. MissingReceiptNotifyProducer** is also prescribed to be *true* the Producer is already informed when no Receipt is received which should also be considered as non-delivery.

A.5 P-Mode[1].PayloadService

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].PayloadService.CompressionType	Recommended value *"application/gzip"*	✓

NOTE: If a payload is already compressed the Access Point is not required to use AS4 compression.

461 ## A.6 PMode[1].ReceptionAwareness

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].ReceptionAwareness	Fixed value: *true*	✓
PMode[1].ReceptionAwareness.DuplicateDetection	Fixed value: *true*	✓
PMode[1].ReceptionAwareness.DetectDuplicates.Parameters	Duplicates MUST be eliminated.	✓
PMode[1].ReceptionAwareness.Retry	Fixed value: *true*	✓
PMode[1].ReceptionAwareness.Retry.Parameters	Not profiled	

462 ## A.7 PMode[1].Security

P-Mode parameter	Profile requirements	Defined in e-SENS profile?
PMode[1].Security.WSSVersion	Fixed value: *1.1.1*	
PMode[1].Security.X509.Sign	At least the `eb:Messaging`, `SOAP:Body` elements and all SOAP attachments MUST be signed.	✓
PMode[1].Security.Signature.Certificate	The PEPPOL Access Point certificate of the sending Access Point MUST be used.	
PMode[1].Security.Signature.X509TokenReferenceType	The *Binary Security Token reference* MUST be used and reference a binary security token of type *X509v3* (i.e. include only the Access Point certificate).	
PMode[1].Security.Signature.HashFunction	Fixed value: *http://www.w3.org/2001/04/xmlenc#sha256*	✓
PMode[1].Security.Signature.Algorithm	Fixed value: *http://www.w3.org/2001/04/xmldsig-more#rsa-sha256*	✓
PMode[1].Security.X509.Encryption.Encrypt	Only the SOAP attachments MUST be encrypted.	✓
PMode[1].Security.Encryption.Certificate	`//smp:Endpoint[@transportProfile="peppol-transport-as4-v1_0"]/smp:Certificate`	✗
PMode[1].Security.Encryption.X509TokenReferenceType	Not profiled	
PMode[1].Security.Encryption.KeyTransportAlgorithm	Fixed value: *http://www.w3.org/2001/04/xmlenc#rsa-oaep*	✗

PMode[1].Security.Encryption. KeyTransportAlgorithmParameters	Fixed values for: Mask generation function: *http://www.w3.org/2009/xmlenc11 #mgf1sha256* Digest generation function: *http://www.w3.org/2001/04/xmlenc#sha256*	✗
PMode[1].Security.Encryption.Algorithm	Fixed value: *http://www.w3.org/2009/xmlenc11#aes128-gcm*	✓
PMode[1].Security.UsernameToken	Not used	✓
PMode[1].Security.PModeAuthorize	Not used	✓
PMode[1].Security.SendReceipt	Fixed value: *true*	✓
PMode[1].Security.SendReceipt.ReplyPattern	Fixed value: *response*	✓
PMode[1].Security.SendReceipt.ReplyTo	Not used	✓
PMode[1].Security.SendReceipt.NonRepudiation	Fixed value: *true*	✓

NOTE 1: The difference between this profile and **[eSENS-AS4]** in the profiled values for the **PMode[1].Security. Encryption.Certificate** parameter results from the PEPPOL eDelivery Network using **[PEPPOL-SMP]** while **[eSENS-AS4]** is based on the newer **[OASIS-SMP]**.

NOTE 2: Although support for the profiled algorithms for encryption key transport is defined as optional in **[XML-ENC1]**, all major platforms support them and interoperability tests have shown no issues in use.

NOTE 3: Beside the newer algorithms for the encryption key transport as required by this profile, **[eSENS-AS4]** also allows the older algorithm (although the newer ones are recommended).