INTERNATIONAL STANDARD

ISO 15000-2

First edition
2021-02

Electronic business eXtensible Markup Language (ebXML) —

Part 2:
Applicability Statement (AS) profile of ebXML messaging service

Reference number
ISO 15000-2:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by the OASIS ebXML Messaging Services Technical Committee (as “OASIS AS4 Profile of ebMS 3.0 Version 1.0”) and drafted in accordance with its editorial rules. It was assigned to Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration* and adopted under the "fast-track procedure".

A list of all parts in the ISO 15000 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Historically, the platform for mission-critical business-to-business (B2B) transactions has steadily moved from proprietary value-added networks (VANs) to Internet-based protocols free from the data transfer fees imposed by the VAN operators. This trend has been accelerated by lower costs and product ownership, a maturing of technology, internationalization, widespread interoperability, and marketplace momentum. The exchange of electronic data interchange (EDI) business documents over the Internet has substantially increased along with a growing presence of extensible markup language (XML) and other document types such as binary and text files.

The Internet messaging services standards that have emerged provide a variety of options for end users to consider when deciding which standard to adopt. These include pre-Internet protocols, the EDIINT series of IETF RFC 3355 AS1, IETF RFC 4130 AS2 and IETF RFC 4823 AS3, simple XML over hypertext transport protocol (HTTP), government specific frameworks, OASIS ebXML messaging (ebMS) 2.0, and web services variants. As Internet messaging services standards have matured, new standards are emerging that leverage prior B2B messaging services knowledge for applicability to web services messaging.

The emergence of the OASIS ebMS 3.0 Standard, now ISO 15000-1:2021, represents a leap forward in Web Services B2B messaging services by meeting the challenge of composing many web services standards into a single comprehensive specification for defining the secure and reliable exchange of documents using web services. ISO 15000-1:2021 composes the fundamental web services standards W3C SOAP 1.1, W3C SOAP 1.2, W3C SOAP with Attachments, OASIS WS-Security 1.0 and 1.1, W3C WS-Addressing, and the OASIS reliable messaging standards WS-Reliability 1.1 and WS-ReliableMessaging - currently at version 1.2, together with guidance for the packaging of messages and receipts along with definitions of messaging choreographies for orchestrating document exchanges.

Like AS2, ISO 15000-1:2021 brings together many existing standards that govern the packaging, security, and transport of electronic data under the umbrella of a single specification document. While ISO 15000-1:2021 represents a leap forward in reducing the complexity of web services B2B messaging, the specification still contains numerous options and comprehensive alternatives for addressing a variety of scenarios for exchanging data over a web services platform.

In order to fully take advantage of the AS2 success story, this profile of ISO 15000-1:2021 has been developed. Using ISO 15000-1:2021 as a base, a subset of functionality has been defined along with implementation guidelines adopted based on the “just-enough” design principles and AS2 functional requirements to trim down ISO 15000-1:2021 into a more simplified and AS2-like specification for web services B2B messaging. The main benefits of AS4 compared to AS2 are:

- compatibility with web services standards;
- message pulling capability;
- a built-in receipt mechanism.

AS4 also provides a minimal client conformance profile that supports data exchanges that have lower-end requirements and do not require (the equivalent of) some of the more advanced capabilities of AS2 and ISO 15000-1:2021, such as support for multiple payloads, message receipts and signing or encryption of messages and receipts.

Profiling ISO 15000-1:2021 means:

- defining a subset of ISO 15000-1:2021 options to be supported by the AS4 handler;
- deciding which types of message exchanges shall be supported, and how these exchanges should be conducted (level of security, binding to HTTP, etc.);
- deciding of AS4-specific message contents and practices (how to make use of the ebMS message header fields, in an AS4 context);
- deciding of some operational best practices, for the end-user.

The overall goal of a profile for a standard is to ensure interoperability by:

- establishing particular usage and practices of the standard within a community of users;
- defining the subset of features in this document that needs to be supported by an implementation.

Two kinds of profiles are usually considered when profiling an existing standard:

1. **Conformance profiles**. These define the different ways a product can conform to a standard, based on specific ways to implement this document. A conformance profile is usually associated with a specific conformance statement. Conformance profiles are of prime interest for product managers and developers: they define a precise subset of features to be supported.
2. **Usage profiles** (also called deployment profiles). These define how a standard should be used by a community of users, in order to ensure best compatibility with business practices and interoperability. Usage profiles are of prime interest for IT end-users: they define how to configure the use of a standard (and related product) as well as how to bind this document to business applications. A usage profile usually points at required or compatible conformance profile(s).

AS4 is defined as a combination of:

- three primary AS4 conformance profiles (see Clause 4) that define three subsets of ISO 15000-1:2021 features, at least one of which is to be supported by an AS4 implementation;
- a set of additional features (see Clause 5);
- an optional complementary conformance profile (see Clause 6) that specifies how to use AS4 endpoints with ISO 15000-1:2021 intermediaries. This is based on a simplified subset of the multi-hop messaging feature defined in the ebMS 3.0 Part 2, Advanced Features specification;
- an AS4 usage profile (see Clause 7) that defines how to use an AS4-compliant implementation in order to achieve similar functions as specified in AS2.

The three primary AS4 conformance profiles (CP) are the following:

(1) The **AS4 ebHandler CP**. This conformance profile supports both sending and receiving roles, and for each role both message pushing and message pulling;

(2) The **AS4 light client CP**. This conformance profile supports both sending and receiving roles, but only message pushing for sending and message pulling for receiving. In other words, it does not support incoming HTTP requests, and may have no fixed IP address.

(3) The **AS4 minimal client CP**. Like the light client CP, this conformance profile does not support the push transport channel binding for the receiving role and therefore does not require HTTP server capabilities. As its name indicates, this CP omits all but a minimal set of features.

Compatible existing conformance profiles for ISO 15000-1:2021 are the following:

- Gateway RM V3 or Gateway RX V3: a message service handler (MSH) implementing any of these profiles will also be conforming to the AS4 ebHandler CP (the reverse is not true).

Full compliance to AS4 actually requires and/or authorizes a message handler to implement a few additional features beyond these conformance profiles, as described in clause 8. These additional features are described in Clause 5.

Electronic business eXtensible Markup Language (ebXML) —

Part 2: Applicability Statement (AS) profile of ebXML messaging service

1 Scope

This document describes the AS4 Profile, which provides a subset of the functionality of ISO 15000-1:2021, along with implementation guidelines based on the "just-enough" design principles and electronic data interchange functional requirements to trim down ISO 15000-1:2021 into a more simplified specification for web services business-to-business messaging.

It specifies:

- three conformance profiles of ISO 15000-1:2021 (see Clause 4);
- a number of AS4 additional features (see Clause 5);
- complementary requirements for the AS4 multi-hop profile (see Clause 6);
- AS4 usage profile of ISO 15000-1:2021 (see Clause 7);
- definitions of conformance (see Clause 8).

Annex A provides some sample messages to support implementation.

Annex B provides a sample XSLT stylesheet to generate an AS4 receipt.

This document is applicable to all types of organizations (e.g., commercial enterprises, government agencies, not-for-profit organizations) that exchange documents or data electronically using messaging.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 15000-1:2021. *Electronic business eXtensible Markup Language (ebXML) — Part 1: Messaging Service 3.0 Core Specification*.

INTERNET ENGINEERING TASK FORCE (IETF). RFC 1952. *GZIP file format specification version 4.3*. IETF RFC. May 1996. http://tools.ietf.org/html/rfc1952

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2045. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.* IETF RFC. November 1996. http://www.ietf.org/rfc/rfc2045.txt

INTERNET ENGINEERING TASK FORCE (IETF). RFC 2616. *Hypertext Transfer Protocol — HTTP/1.1.* IETF RFC. June 1999. Available from http://www.ietf.org/rfc/rfc2616.txt

OASIS. *OASIS ebXML Business Signals Schema*, 21 December 2006. OASIS Standard. http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0

OASIS. *OASIS ebXML Messaging Services Version 3.0: Part 2, Advanced Features.* Committee Specification 01, 19 May 2011. OASIS committee specification. Available at http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/part2/201004/ebms-v3-part2.odt

OASIS. *Web Services Security: SOAP Message Security 1.1.* OASIS Standard incorporating Approved Errata. 1 November 2006. Available from http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf

OASIS. *Web Services Security UsernameToken Profile 1.1.* OASIS Standard. 1 February 2006. Available from http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-UsernameTokenProfile.pdf.

OASIS. *Web Services Security X.509 Certificate Token Profile 1.1*. OASIS Standard incorporating Approved Errata. 1 November 2006. Available from http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-x509TokenProfile.pdf

WEB SERVICES INTEROPERABILITY ORGANIZATION. *WS-I Attachments Profile Version 1.0*, WS-I Final Material. 20 April 2004. Available from http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html

WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Profile Version 2.0*, WS-I Final Material. 9 November 2010. Available from http://ws-i.org/Profiles/BasicProfile-2.0-2010-11-09.html

WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Security Profile Version 1.1*, WS-I Final Material. 24 January 2010. Available from http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html

WORLD WIDE WEB CONSORTIUM (W3C). *SOAP Version 1.2 Part 1: Messaging Framework.* W3C Recommendation. 27 April 2007. Available from http://www.w3.org/TR/soap12-part1/

WORLD WIDE WEB CONSORTIUM (W3C). *SOAP Messages with Attachments*, W3C Note. 11 December 2000. Available from http://www.w3.org/TR/SOAP-attachments

WORLD WIDE WEB CONSORTIUM (W3C). *Web Services Addressing 1.0 – Core.* W3C Recommendation. 9 May 2006. Available from http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/

WORLD WIDE WEB CONSORTIUM (W3C). *Extensible Markup Language (XML) 1.0.* W3C Recommendation 26 November 2008. Available from http://www.w3.org/TR/REC-xml/

WORLD WIDE WEB CONSORTIUM (W3C). *XML Signature Syntax and Processing (Second Edition).* W3C Recommendation. 10 June 2008. Available from http://www.w3.org/TR/xmldsig-core/

WORLD WIDE WEB CONSORTIUM (W3C). *XML Encryption Syntax and Processing*. 10 December, 2002. Available from http://www.w3.org/TR/xmlenc-core/

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 15000-1:2021 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

4 AS4 conformance profiles for ISO 15000-1:2021

4.1 General

AS4 is more than a conformance profile, in the sense given in the *OASIS ebXML Messaging Services, Version 3.0: Conformance Profiles* OASIS committee specification. It is a combination of a conformance profile and a usage profile, as explained in the Introduction. Consequently, only this clause is conforming to the format recommended in the *OASIS ebXML Messaging Services, Version 3.0: Conformance Profiles* OASIS committee specification for describing conformance profiles. The usage profile part (clause 7) is following a format based on tables similar to those found in the OASIS *Deployment Profile Template for OASIS ebXML Message Service 2.0 Standard*.

4.2 The AS4 ebHandler conformance profile

4.2.1 General

The AS4 ebHandler conformance profile addresses common functional requirements of e-Business/e-Government gateways. It is identified by the URI:

http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cprofiles/200809/as4ebhandler

NOTE: this URI is only an identifier, not a document address.

4.2.2 Feature set

The AS4 CP is defined in Table 1, using the table template and terminology provided in Annex G ("Conformance") of ISO 15000-1:2021

Conformance profile: **AS4 ebHandler**	**Profile summary**: <"Sending+Receiving" / "AS4 ebHandler" / Level 1 / HTTP 1.1 + SOAP 1.2 + WSS 1.1 >
Functional aspects	**Profile feature set**
ebMS MEP	The following ebMS simple message exchange patterns (MEPs) shall be supported both as Initiating and Responding partner: ● One-way / push ● One-way / pull This does not prevent an implementation to also support asynchronous two-way MEPs. Regardless of which MEP is used, the sending of an `eb:Receipt` message shall be supported: ● For the one-way / push, both "response" and "callback" reply patterns shall be supported. ● For the one-way / pull, the "callback" pattern is the only viable option, and the user message sender shall be ready to accept an `eb:Receipt` either piggybacked on (or bundled with) an `eb:PullRequest`, or piggybacked on another user Message, or sent separately. In all MEPs, the user message receiver shall be able to send an `eb:Receipt` as a separate message (i.e. not piggybacked on an `eb:PullRequest` message or on another user message). An MSH conforming to this profile is therefore not required to bundle an

	`eb:Receipt` with any other ebMS header or message body. The `ebbpsig:NonRepudiationInformation` element as defined in the *OASIS ebXML Business Signals Schema*, 21 December 2006 OASIS Standard shall be used as content for the `eb:Receipt` message, i.e. when conforming to this profile a receiving MSH shall be able to create an `eb:Receipt` with such a content, and a sending MSH shall be able to process it.
Reliability	Reception awareness, defined as the ability for a sending ebHandler to notify its application (message producer) of lack of reception of an `eb:Receipt` related to a sent message, shall be supported. This implies support for: ● correlating `eb:Receipt` elements with previously sent user messages, based on the ebMS message identifier; ● detection of a missing `eb:Receipt` for a sent message; ● ability to report an error to the message producer in case no `eb:Receipt` has been received for a sent message. The semantics for sending back an `eb:Receipt` message is as follows: a well-formed ebMS user message has been received and the MSH is taking responsibility for its processing (additional application-level delivery semantics, and payload validation semantics are not relevant). Support for a WS reliable messaging specification is optional.
Security	The following security features shall be supported: ● Support for the OASIS *Web Services Security: SOAP Message Security 1.1.* OASIS Standard. ● Support for username / password token, digital signatures and encryption, as specified in OASIS *Web Services Security UsernameToken Profile 1.1,* OASIS *Web Services Security X.509 Certificate Token Profile 1.1,* and the W3C Recommendations *XML Signature Syntax and Processing* and W3C *XML Encryption Syntax and Processing.* ● Support for content-only transforms. ● Support for security of attachments. ● Support for message authorization at P-Mode level (see ISO a:—, 10.11) Authorization of the Pull signal, for a particular MPC, shall be supported at minimum. ● Transport-level secure protocols such as SSL or TLS. Two authorization options shall be supported by an MSH in the receiving role, and at least one of them in the sending role: ● **Authorization Option 1**: Use of the WSS security header targeted to the "ebms" actor, as specified in clause 10.12 of ISO 15000-1, with the `wsse:UsernameToken` profile. This header may either come in addition to the regular wsse security header (XML signature for authentication), or may be the sole wsse header, if a transport-level secure protocol such as SSL or TLS is used. ● **Authorization Option 2**: Use of a regular wsse security header, using XML signature[a] and X509 for authentication, and no additional wsse security header targeted to "ebms". In that case, the MSH shall be able to use the credential present in this security header for Pull authorization, i.e. to associate these with a specific MPC. The use of transport-level secure protocols such as SSL or TLS is recommended.

Error generation and reporting	The following error processing capabilities shall be supported: ● Capability of the receiving MSH to report errors from message processing, either as ebMS error messages or as SOAP faults to the sending MSH. The following modes of reporting to a sending MSH are supported: ○ Sending error on the back channel of the underlying protocol (**ErrorHandling.Report.AsResponse**="true"). ○ Capability to report to a third-party address (**ErrorHandling.Report.ReceiverErrorsTo**=<other address>). ● Capability of sending MSH to report generated errors as notifications to the message producer (support for **Report.ProcessErrorNotifyProducer**="true")(e.g. delivery failure). ● Generated errors: All specified errors in ISO 15000-1 shall be generated when applicable, except for `EBMS:0010`: On a receiving MSH, there is no requirement to generate error `EBMS:0010` for discrepancies between message header and the **P-Mode.reliability** and **P-Mode.security** features. A receiving MSH shall generate such errors for other discrepancies.
Message partition channels	Message partition channels (MPC) shall be supported in addition to the default channel, so that selective pulling by a partner MSH is possible. This means AS4 handlers shall be able to use the `@mpc` attribute and to process it as expected.
Message packaging	The following features shall be supported both on sending and receiving sides: ● Support for attachments following IETF RFC 2045 and the W3C SOAP-with-attachments recommendation. ● Support for message properties. ● Support for processing messages that contain both a signal message unit (`eb:SignalMessage`) and a user message unit (eb:UserMessage) – this may happen when a same ebMS message carries message units for different MEP instances. Per WS-I Basic Profile 2.0, at most one payload may be inserted as direct child element of the SOAP `Body`.
Interoperability Parameters	The following interoperability parameters values shall be supported for this conformance profile: ● **Transport:** HTTP 1.1, conform IETF RFC 2616. ● **SOAP version:** 1.2, conform W3C *SOAP Version 1.2 Part 1: Messaging Framework* W3C Recommendation. ● **Reliability Specification:** none. ● **Security Specification:** WSS 1.1 conform OASIS *Web Services Security: SOAP Message Security 1.1.*
[a] XML signature allows arbitrary XSLT transformations when constructing the plaintext over which a signature or reference is created. Conforming applications that allow use of XSLT transformations when verifying either signatures or references are encouraged to maintain lists of "safe" transformations for a given partner, service, action and role combination. Static analysis of XSLT expressions with a human user audit is encouraged for trusting a given expression as "safe".	

Table 1: AS4 ebHandler feature set

4.2.3 WS-I conformance profiles

The Web-Services Interoperability consortium has defined guidelines for interoperability of SOAP messaging implementations. In order to ensure maximal interoperability across different SOAP stacks, e.g. multipurpose Internet mail extensions (MIME) and HTTP implementations, implementations shall comply with the following WS-I profiles whenever related features are used:

- WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Security Profile Version 1.1* Basic Security Profile (BSP) 1.1.
- WEB SERVICES INTEROPERABILITY ORGANIZATION. *WS-I Attachments Profile Version 1.0* Attachment Profile (AP) 1.0 with regard to the use of MIME and SOAP with Attachments.

NOTE 1:Compliance with AP1.0 would normally require compliance with BP1.1, which in turn requires the absence of a SOAP envelope in the HTTP response of a One-Way MEP (R2714). However, recent BP versions such as BP1.2 and BP2.0 override this requirement. Consequently, the AS4 ebHandler conformance profile does not require conformance to these deprecated requirements inherited from BP1.1 (R2714, R1143) regarding the use of HTTP.

NOTE 2:WS-I compliance is here understood as requiring that the features exhibited by an AS4 ebHandler shall comply with these WS-I profiles. For example, since only SOAP 1.2 is required by the AS4 ebHandler, the requirements from BSP 1.1 that depend on SOAP 1.1 would not apply. Similarly, none of the requirements for DESCRIPTION (WSDL) or REGDATA (UDDI) apply here, as these are not used.

Implementations shall also conform to the following WS-I profile:

- WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Profile Version 2.0*, Basic Profile 2.0 (BP2.0).

4.2.4 Processing mode parameters

4.2.4.1 General

This subclause contains a summary of P-Mode parameters relevant to AS4 features for this conformance profile. An AS4 handler shall support and understand those that are mentioned as "required". For each parameter, one the following situations applies:

- Full support is required: an implementation shall support the possible options for this parameter.
- Partial support is required: support for a subset of values is required.
- No support is required: an implementation is not required to support the features controlled by this parameter, and therefore is not required to understand this parameter.

An AS4 handler is expected to support the P-Mode set of section 4.2.4.2, both as a sender (of the user message) and as a receiver.

4.2.4.2 General P-Mode parameters

- **PMode.ID**: support required.
- **PMode.Agreement:** support required.

- **PMode.MEP:** support required for: http://www.oasis-open.org/committees/ebxml-msg/one-way
- **PMode.MEPbinding:** support required for: http://www.oasis-open.org/committees/ebxml-msg/push and http://www.oasis-open.org/committees/ebxml-msg/pull.
- **PMode.Initiator.Party:** support required.
- **PMode.Initiator.Role:** support required.
- **PMode.Initiator.Authorization.username** and **PMode.Initiator.Authorization.password:** support required for: `wsse:UsernameToken`.
- **PMode.Responder.Party:** support required.
- **PMode.Responder.Role:** support required.
- **PMode.Responder.Authorization.username** and **PMode.Responder.Authorization.password:** support required for: `wsse:UsernameToken`.

4.2.4.3 PMode[1].Protocol

- **PMode[1].Protocol.Address:** support required for "http" protocol (IETF RFC 2616).
- **PMode[1].Protocol.SOAPVersion:** support required for the WORLD WIDE WEB CONSORTIUM (W3C). *SOAP Version 1.2 Part 1: Messaging Framework.*

4.2.4.4 PMode[1].BusinessInfo

- **PMode[1].BusinessInfo.Service:** support required.
- **PMode[1].BusinessInfo.Action:** support required.
- **PMode[1].BusinessInfo.Properties[]:** support required.
- **(PMode[1].BusinessInfo.PayloadProfile[]:** support not required**)**
- **(PMode[1].BusinessInfo.PayloadProfile.maxSize:** support not required**)**

4.2.4.5 PMode[1].ErrorHandling

- **(PMode[1].ErrorHandling.Report.SenderErrorsTo:** support not required**)**
- **PMode[1].ErrorHandling.Report.ReceiverErrorsTo:** support required (for address of the MSH sending the message in error or for third-party).
- **PMode[1].ErrorHandling.Report.AsResponse:** support required (true/false).
- **(PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** support not required**)**
- **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**: support required (true/false)

- **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer:** support required (true/false)

4.2.4.6 PMode[1].Reliability

Support not required.

4.2.4.7 PMode[1].Security

- **PMode[1].Security.WSSVersion:** support required for: 1.1.
- **PMode[1].Security.X509.Sign:** support required.
- **PMode[1].Security.X509.Signature.Certificate:** support required.
- **PMode[1].Security.X509.Signature.HashFunction:** support required.
- **PMode[1].Security.X509.Signature.Algorithm:** support required.
- **PMode[1].Security. X509.Encryption.Encrypt:** support required.
- **PMode[1].Security.X509.Encryption.Certificate:** support required.
- **PMode[1].Security.X509.Encryption.Algorithm:** support required.
- **(PMode[1].Security.X509.Encryption.MinimumStrength:** support not required**)**
- **PMode[1].Security.UsernameToken.username:** support required.
- **PMode[1].Security.UsernameToken.password:** support required.
- **PMode[1].Security.UsernameToken.Digest:** support required (true/false)
- **(PMode[1].Security.UsernameToken.Nonce:** support not required**)**
- **PMode[1].Security.UsernameToken.Created:** support required.
- **PMode[1].Security.PModeAuthorize:** support required (true/false)
- **PMode[1].Security.SendReceipt:** support required (true/false)
- **Pmode[1].Security.SendReceipt.ReplyPattern:** support required (both “response” and “callback”))

4.3 The AS4 light client conformance profile

4.3.1 General

The AS4 light client conformance profile addresses common functional requirements of e-Business/e-Government light gateways. It is identified by the URI:

http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cprofiles/200809/as4lightclient

NOTE: this URI is only an identifier, not a document address.

As indicated by its name, this profile applies only to one side of an MEP (acting as a “client” to the other party). It is not required and often not even possible for two MSHs conforming to this profile to engage in a point-to-point exchange. Indeed, at least one MSH shall be ready to receive an incoming HTTP request in any MEP as defined in ebMS, but this profile does not require this capability. As a result, when an MSH is conforming exclusively to this profile, it can only engage into point-to-point exchanges with MSHs that conform to “more” than this profile – e.g. MSHs that conform to the ebHandler profile– in order to be able to receive requests. Two light clients can also exchange messages using store-and-forward ebMS3 intermediaries, as described in Clause 6.

4.3.2 Feature set

The AS4 light client feature set is described in Table 2.

Conformance profile: **AS4 light client**	**Profile summary**: <“Sending+Receiving” / “AS4 light client” / Level 1 / HTTP 1.1 + SOAP 1.2>
Functional aspects	**Profile feature set**
ebMS MEP	The following message exchange patterns (MEPs) shall be supported as Initiating partner: ● One-way / push ● One-way / pull This does not prevent an implementation to also support two-way MEPs. The following requirement details apply for each MEP: ● For the one-way / push, the “response” reply pattern shall be supported on the **PMode[1].Security.SendReceipt.ReplyPattern** parameter by the initiating client MSH. ● For the one-way / pull, the “callback” pattern is the only viable option, and the receiving MSH (initiating light client) shall be able to send an `eb:Receipt` separately from the `eb:PullRequest`. It may additionally be able to send an `eb:Receipt` piggybacked on an `eb:PullRequest`. In all MEPs, the user Message receiver shall be able to send an `eb:Receipt` as a separate message (i.e. not piggybacked on an `eb:PullRequest` message or on another user message). An MSH conforming to this profile is therefore not required to bundle an `eb:Receipt` with any other ebMS header or message body. However, when receiving an `eb:Receipt`, an MSH conforming to this profile shall be able to process an `eb:Receipt` bundled with another ebMS message header or body. The `ebbpsig:NonRepudiationInformation` element as defined in the *OASIS ebXML Business Signals Schema*, December 2006 OASIS Standard, shall be used as content for the `eb:Receipt` message, i.e. when conforming to this profile a receiving MSH shall be able to create an `eb:Receipt` with such a content, and a sending MSH shall be able to process it.

Reliability	Reception awareness, defined as the ability for a sending light client to notify its application (message producer) of lack of reception of an `eb:Receipt` related to a sent message, shall be supported. This implies support for: - Correlating `eb:Receipt` elements with previously sent user messages, based on the ebMS message identifier. - Detection of a missing `eb:Receipt` for a sent message. - Ability to report an error to the message producer in case no `eb:Receipt` has been received for a sent message. The semantics for sending back an `eb:Receipt` message is as follows: a well-formed ebMS user message has been received and the MSH is taking responsibility for its processing, (additional application-level delivery semantics, and payload validation semantics are not relevant). Support for a WS reliable messaging specification is optional.
Security	Both authorization options for message pulling (authorizing an `eb:PullRequest` for a particular MPC) described in the ebHandler conformance profile shall be supported: 1. Support for username / password token: minimal support for `wss:UsernameToken` profile in the Pull signal - for authorizing a particular MPC. Support for adding a WSS security header targeted to the “ebms” actor, as specified in ISO 15000-1:2021, 10.11, with the `wsse:UsernameToken` profile. The use of transport-level secure protocol such as SSL or TLS is recommended. 2. Support for a regular wsse security header (XML signature for authentication, use of X509), and no additional wsse security header targeted to “ebms”. Implementations shall conform to the OASIS *Web Services Security: SOAP Message Security 1.1.* OASIS Standard, and the W3C Recommendations *XML Signature Syntax and Processing* and W3C *XML Encryption Syntax and Processing* The use of transport-level secure protocols such as SSL or TLS is recommended.
Error generation and reporting	Error notification to the local message producer shall be supported (e.g. reported failure to deliver pushed messages). The reporting of message processing errors for pulled messages to the remote party shall be supported via error messages (errors may be bundled with another pushed message or a pull request signal message.).
Message Partition Channels	Sending on the default message partition channel is sufficient (support for additional message partitions is not required).
Message packaging	Support shall be provided for attachments – i.e. an XML message payload may use the SOAP body or a MIME part. MIME packaging shall conform to IETF RFC 2045 and the W3C SOAP-with-attachments recommendation. Support shall be provided for message properties. Per WS-I Basic Profile 2.0, at most one payload may be inserted as direct child element of the SOAP `Body`.
Interoperability Parameters	The following interoperability parameters values shall be supported for this conformance profile: - **Transport:** HTTP 1.1, conform IETF RFC 2616. - **SOAP version:** 1.2, conform W3C *SOAP Version 1.2 Part 1: Messaging Framework* W3C Recommendation. - **Reliability Specification:** none. - **Security Specification:** WSS 1 conform OASIS *Web Services Security: SOAP Message Security 1.1.*

Table 2: AS4 light client feature set

4.3.3 WS-I conformance requirements

This conformance profile will require compliance with the following WS-I profile:

1. WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Profile Version 2.0* (BP2.0).

The features exhibited by an AS4 light client ebMS conformance profile shall comply with this WS-I profile.

4.3.4 Processing mode parameters

4.3.4.1 General

This subclause contains a summary of P-Mode parameters relevant to AS4 features for this conformance profile. An AS4 light client shall support and understand those that are mentioned as "required". For each parameter, one of the following situations applies:

- Full support is required: An implementation is supposed to support the possible options for this parameter.
- Partial support is required: Support for a subset of values is required.
- No support is required: An implementation is not required to support the features controlled by this parameter, and therefore not required to understand this parameter.

An AS4 light client is expected to support the P-Mode set of clause 4.3.4.2 both as a sender (of the user message, in case of a one-way / push) and as a receiver (in case of a one-way / pull).

4.3.4.2 General P-Mode parameters

- **PMode.ID**: support required.
- **PMode.Agreement:** support required.
- **PMode.MEP:** support required for: http://www.oasis-open.org/committees/ebxml-msg/one-way
- **PMode.MEPbinding:** support required for: http://www.oasis-open.org/committees/ebxml-msg/push and http://www.oasis-open.org/committees/ebxml-msg/pull.
- **PMode.Initiator.Party:** support required.
- **PMode.Initiator.Role:** support required.
- **PMode.Initiator.Authorization.username** and **PMode.Initiator.Authorization.password:** support required for: `wsse:UsernameToken`. (as initiator of the one-way / pull)
- **PMode.Responder.Party:** support required.
- **PMode.Responder.Role:** support required.

- **PMode.Responder.Authorization.username** and **PMode.Responder.Authorization.password:** support not required.

4.3.4.3 PMode[1].Protocol

- **PMode[1].Protocol.Address:** support required for “http” protocol (IETF RFC 2616).
- **PMode[1].Protocol.SOAPVersion:** support required for the WORLD WIDE WEB CONSORTIUM (W3C). *SOAP Version 1.2 Part 1: Messaging Framework*.

4.3.4.4 PMode[1].BusinessInfo

- **PMode[1].BusinessInfo.Service:** support required**.**
- **PMode[1].BusinessInfo.Action:** support required**.**
- **PMode[1].BusinessInfo.Properties[]:** support required.
- **(PMode[1].BusinessInfo.PayloadProfile[]:** support not required**)**
- **(PMode[1].BusinessInfo.PayloadProfile.maxSize:** support not required**)**

4.3.4.5 PMode[1].ErrorHandling

- **(PMode[1].ErrorHandling.Report.SenderErrorsTo:** support not required**)**
- **PMode[1].ErrorHandling.Report.AsResponse:** support required (true/false) as initiator of the one-way / push, as well as for the `eb:PullRequest` signal (PMode[1][s]).
- **(PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** support not required**)**
- **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**: support required (true/false)
- **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer:** support required (true/false)

4.3.4.6 Pmode[1].Reliability

Support not required.

4.3.4.7 PMode[1].Security

- **PMode[1].Security.WSSVersion:** support required for: 1.1
- **PMode[1].Security.X509.Sign:** support required.
- **PMode[1].Security.X509.Signature.Certificate:** support required.
- **PMode[1].Security.X509.Signature.HashFunction:** support required.
- **PMode[1].Security.X509.Signature.Algorithm:** support required.

- **PMode[1].Security. X509.Encryption.Encrypt:** support not required.
- **PMode[1].Security.X509.Encryption.Certificate:** support not required.
- **PMode[1].Security.X509.Encryption.Algorithm:** support not required.
- **(PMode[1].Security.X509.Encryption.MinimumStrength:** support not required**)**
- **PMode[1].Security.UsernameToken.username:** support required.
- **PMode[1].Security.UsernameToken.password:** support required.
- **PMode[1].Security.UsernameToken.Digest:** support required (true/false)
- **(PMode[1].Security.UsernameToken.Nonce:** support not required**)**
- **PMode[1].Security.UsernameToken.Created:** support required.
- **PMode[1].Security.PModeAuthorize:** support required (true/false)
- **PMode[1].Security.SendReceipt:** support required (true/false)
- **Pmode[1].Security.SendReceipt.ReplyPattern:** support required for “response” if PMode.MEPbinding is "push", and for "callback" if PMode.MEPbinding is "pull".

4.4 The AS4 minimal client conformance profile

4.4.1 General

The AS4 minimal client addresses low-end functional data exchange requirements. It also supports business processes that do not require signing of messages and of message receipts. It is identified by the URI:

http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cprofiles/200809/as4minimalclient

NOTE: this URI is only an identifier, not a document address.

As indicated by its name, this profile applies only to one side of an MEP (acting as a “client” to the other party). It is not required and often not even possible for two MSHs conforming to this profile to engage in a point-to-point exchange. Indeed, at least one MSH shall be ready to receive an incoming (HTTP) request in any MEP as defined in ebMS, but this profile does not require this capability. As a result, when an MSH is conforming exclusively to this profile, it can only engage into point-to-point exchanges with MSHs that conform to “more” than this profile –e.g. MSHs that conform to the ebHandler profile– in order to be able to receive requests.

4.4.2 Feature set

The feature set of the minimal client conformance profile is described in Table 3.

Conformance profile: **AS4 minimal client**	**Profile summary**: <“Sending” / “AS4 minimal client” / Level 1 / HTTP 1.1 + SOAP 1.2>

Functional aspects	**Profile feature set**
ebMS MEP	The following message exchange patterns (MEPs) shall be supported as Initiating partner: ● One-way / Push ● One-way / Pull The requirement to support Pull is relaxed in the AS4 minimal sender conformance statement. No support for receipts is required: the **PMode[1].Security.SendReceipt** parameter does not need to be supported for value "true".
Reliability	Support for a WS reliable messaging specification is not required. Support for Reception Awareness is not required.
Security	Implementations shall conform to the OASIS *Web Services Security: SOAP Message Security 1.1.* OASIS Standard. The first authorization option for message pulling (authorizing an `eb:PullRequest` for a particular MPC) described in the ebHandler conformance profile should be supported: 1. Support for adding a WSS security header targeted to the "ebms" actor, as specified in clause 10.11 of ISO 15000-1, with the `wsse:UsernameToken` profile, OASIS. *Web Services Security UsernameToken Profile 1.1.* This requirement is relaxed in the AS4 minimal sender (Clause 8.5) Support for the OASIS WSS Web Services Security X.509 Certificate Token Profile, *1.1* is not required. The use of transport-level secure protocols such as SSL or TLS is recommended.
Error generation and reporting	Error notification to the local message producer shall be supported (e.g. reported failure to deliver pushed messages).
Message partition channels	Sending on the default message partition channel is sufficient (support for additional message partitions is not required.)
Message packaging	Support for attachments is not required – i.e. an XML message payload will always use the SOAP body. Per WS-I Basic Profile 2.0, at most one payload may be inserted as direct child element of the SOAP `Body`. Support for message properties is not required.
Interoperability parameters	The following interoperability parameters values shall be supported for this conformance profile: ● **Transport:** HTTP 1.1, conform IETF RFC 2616. ● **SOAP version:** 1.2, conform W3C *SOAP Version 1.2 Part 1: Messaging Framework* W3C Recommendation. ● **Reliability Specification:** none. ● **Security Specification:** none.

Table 3: AS4 minimal client feature set

4.4.3 WS-I conformance requirements

This conformance profile will require compliance with the following WS-I profile:

- WEB SERVICES INTEROPERABILITY ORGANIZATION. *Basic Profile Version 2.0* (BP2.0).

The features exhibited by an AS4 minimal client ebMS conformance profile shall comply with this WS-I profile.

4.4.4 Processing mode parameters

4.4.4.1 General

This subclause contains a summary of P-Mode parameters relevant to AS4 features for this conformance profile. An AS4 minimal client shall support and understand those that are mentioned as "required". For each parameter, one of the following situations applies:

- Full support is required: An implementation is supposed to support the possible options for this parameter.
- Partial support is required: Support for a subset of values is required.
- No support is required: An implementation is not required to support the features controlled by this parameter, and therefore not required to understand this parameter.

An AS4 minimal client is expected to support the P-Mode set specified in 4.4.4.2. as a sender of the user message.

4.4.4.2 General P-Mode parameters

- **PMode.ID**: support required.
- **PMode.Agreement:** support required.
- **PMode.MEP:** support required for: http://www.oasis-open.org/committees/ebxml-msg/one-way
- **PMode.MEPbinding:** support required for: http://www.oasis-open.org/committees/ebxml-msg/push.
- **PMode.Initiator.Party:** support required**.**
- **PMode.Initiator.Role:** support required**.**
- **PMode.Initiator.Authorization.username** and **PMode.Initiator.Authorization.password:** support not required.
- **PMode.Responder.Party:** support required**.**
- **PMode.Responder.Role:** support required**.**
- **PMode.Responder.Authorization.username** and **PMode.Responder.Authorization.password:** support not required.

4.4.4.3 PMode[1].Protocol

- **PMode[1].Protocol.Address:** support required for “http” protocol (IETF RFC 2616).
- **PMode[1].Protocol.SOAPVersion:** support required for the WORLD WIDE WEB CONSORTIUM (W3C). *SOAP Version 1.2 Part 1: Messaging Framework*.

4.4.4.4 PMode[1].BusinessInfo

- **PMode[1].BusinessInfo.Service:** support required**.**
- **PMode[1].BusinessInfo.Action:** support required**.**
- **PMode[1].BusinessInfo.Properties[]:** support not required.
- **(PMode[1].BusinessInfo.PayloadProfile[]:** support not required**)**
- **(PMode[1].BusinessInfo.PayloadProfile.maxSize:** support not required**)**

4.4.4.5 PMode[1].ErrorHandling

- **(PMode[1].ErrorHandling.Report.SenderErrorsTo:** support not required**)**
- **PMode[1].ErrorHandling.Report.AsResponse:** support required (true/false) as initiator of the one-way / push.
- **(PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** support not required**)**
- **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**: support required (true/false)
- **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer:** support not required

4.4.4.6 Pmode[1].Reliability

Support not required.

4.4.4.7 Pmode[1].Security

Support not required.

- **PMode[1].Security.SendReceipt:** support not required.

4.5 Conformance profiles compatibility

The AS4 profile is compatible with the following ISO 15000-1 conformance profiles, defined in the *OASIS ebXML Messaging Services, Version 3.0: Conformance Profiles.* OASIS committee specification, 24 April 2010:

1. Gateway RM V2/3
2. Gateway RM V3
3. Gateway RX V2/3
4. Gateway RX V3

AS4 may be deployed on any MSH that conforms to one of these four conformance profiles.

AS4 may also be deployed on an MSH that supports B2B messaging protocols other than ebMS, such as IETF RFC 4130 AS2. Such an MSH could be used by organizations that use AS2 for some business partners, or for some types of documents, and AS4 for others.

5 AS4 additional features

5.1 General

This clause defines features that were not specified in the ISO 15000-1 and therefore out of scope for the ISO 15000-1 conformance profiles (ebHandler CP and light client CP). These features should be considered as additional capabilities that are either required by or made optional to AS4 implementations as indicated in the conformance statements in clause 8.

Tables 4 and 5 can be used for adding user-defined profiling requirements to be adopted within a business community. Whenever the feature, or its profiling, is mandatory, the right-side column (“profile requirement”) will specify it.

5.2 Compression

The AS4 compression feature provides configurable (de)compression of application payloads. AS4 messages containing compressed application payloads are built in conformance with the W3C SOAP with Attachments (SwA) specification[1]. Each compressed payload is carried in a separate MIME body part, following IETF RFC 2045. Compression of the SOAP envelope and/or of a payload contained within the SOAP Body of an ebMS message is not supported by the feature described here. However, if compression of the SOAP envelope is required then the content-coding feature of IETF RFC 2616, HTTP/1.1, may be used.

To compress the payload(s) of a message payload, the IETF RFC 1952 GZIP compression algorithm shall be used. Compression shall be applied before payloads are attached to the SOAP message.

The eb:PartInfo element in the message header that relates to the compressed message part, shall have an eb:Property element with @name ="CompressionType":

```
<eb:Property name="CompressionType">application/gzip</eb:Property>
```

The content type of the compressed attachment shall be "application/gzip".

These are indicators to the receiving MSH that the sending MSH has compressed a payload part. The receiving AS4 MSH shall decompress any payload part(s) compressed by the sending MSH before delivering the message.

When compression, signature and encryption are required, any attached payload(s) shall be compressed prior to being signed and/or encrypted.

[1] Although a SOAP 1.2 version of SwA has not been formally submitted to W3C, it appears that most SOAP products have anticipated that usage, and after investigation, it appears that they have done so in a consistent, interoperable way. this document is acknowledging these *de facto* upgrades of SwA, (see ISO 15000-1:2021, Annex C).

Packaging requirements:

- An `eb:PartInfo/eb:PartProperties/eb:Property/@name="MimeType"` value shall be used to identify the MIME type of the payload before compression was applied.

- For XML payloads, an `eb:PartInfo/eb:PartProperties/eb:Property/@name="CharacterSet"` value should be used to identify the character set of the payload before compression was applied. The values of this property shall conform to the values defined in clause 4.3.3 of the WORLD WIDE WEB CONSORTIUM (W3C). *Extensible Markup Language (XML) 1.0.* W3C Recommendation 26 November 2008.

Example:

```
<eb:PartInfo href="cid:attachment1234@example.com" >
  <eb:PartProperties>
        <eb:Property name="MimeType">application/xml</eb:Property>
        <eb:Property name="CharacterSet">utf-8</eb:Property>

       <eb:Property name="CompressionType">application/gzip</eb:Property>
  </eb:PartProperties>

<eb:PartInfo>
```

An additional P-Mode parameter is defined, which shall be supported as part of the compression feature:

- **PMode[1].PayloadService.CompressionType:** (either absent, empty or equal to “application/gzip”)

Value=“application/gzip”: the AS4 sending MSH should compress the attached payload(s) over this MEP segment. GZIP compression of payloads in data formats that provide native, built-in compression typically often does not result in good compression ratios and is therefore not required.

Absent or empty (default): no compression is used over this MEP segment.

In case of error during decompression, the following error shall be used: `Code = EBMS:0303, Short Description = DecompressionFailure, Severity = Failure, Category = Communication`.

5.3 Reception awareness features and duplicate detection

These capabilities described in Table 4 make use of the `eb:Receipt` as the sole type of acknowledgment. Duplicate detection only relies on the `eb:MessageInfo/eb:MessageId`.

Features	Profile requirements
Reception awareness error handling (required support)	Ability for the MSH expecting an `eb:Receipt` to generate an error in case no eb:Receipt has been received for a sent message. It is recommended that this error be a new error: `Code = EBMS:0301, Short Description = MissingReceipt, Severity = Failure, Category = Communication`. Ability for the MSH expecting an `eb:Receipt` to report a `MissingReceipt` error to the message producer.
Message retry (optional support)	Ability for a user message sender that has not received an expected `eb:Receipt` to resend the user message. If doing so, the `eb:MessageInfo/eb:MessageId` element of the resent message and of the original user message shall be same. When resending a message for which non-repudiation of receipt is required, the sender shall ensure that the hash values for the digests to be included in the `eb:Receipt` (i.e. the content of `ebbpsig:MessagePartNRInformation` elements), do not vary from the original message to the retry(ies), so that non-repudiation of receipt can be asserted based on the original message and the receipt of any of its retries.
Duplicate detection (required support)	Ability for the MSH receiving a user message to detect and/or eliminate duplicates based on `eb:MessageInfo/eb:MessageId`. If duplicates are just detected (not eliminated) then at the very least the receiving MSH shall notify its application (message consumer) of the duplicates. For examples, these could be logged. Related quantitative parameters (time window for the detection, or maximum message log size) are left to the implementation.

Table 4: Reception awareness features and duplicate detection

The following additional P-Mode parameters are defined as part of the reception awareness feature:

- **PMode[1].ReceptionAwareness:** (true / false)
  NOTE: when set to true, the **PMode[1].Security.SendReceipt** shall also be set to true.

- **PMode[1].ReceptionAwareness.Retry:** (true / false)

- **PMode[1].ReceptionAwareness.Retry.Parameters:**. (contains a composite string specifying: (a) maximum number of retries or some timeout, (b) frequency of retries or some retry rule). The string contains a sequence of parameters of the form: name=value, separated by either comas or ';'. Example: "`maxretries=10,period=3000`", in case the retry period is 3 000 ms.

- **PMode[1].ReceptionAwareness.DuplicateDetection:** (true / false)

- **PMode[1].ReceptionAwareness.DetectDuplicates.Parameters:** (contains an implementation specific composite string. As an example this string may specify either (a) maximum size of message log over which duplicate detection is supported, (b) maximum time window over which duplicate detection is supported). The string contains a sequence of parameters of the form: name=value, separated by either comas or ';'. Example: "`maxsize=10Mb,checkwindow=7D`", in case the duplicate check window is guaranteed of 7 days minimum.

5.4 Alternative pull authorization

In addition to the two authorization options described in the AS4 conformance profile (Clause 4), an implementation may optionally decide to support a third authorization technique, based on transient security (SSL or TLS).

SSL/TLS can provide certificate-based client authentication. Once the identity of the pulling client is established, the Security module may pass this identity to the ebms module, which can then associate it with the right authorization entry, e.g. the set of MPCs this client is allowed to pull from.

This third authorization option, compatible with AS4 although not specified in ISO 15000-1:2021, relies on the ability of the ebMS module to obtain the client credentials. This capability represents an (optional) new feature. When using this option for authorizing pulling, there is no need to insert any WS-Security header in the Pull request at all.

5.5 Semantics of receipt in AS4

The notion of receipt in ISO 15000-1 is not associated with any particular semantics, such as delivery assurance. However, when combined with security (signing), it is intended to support non-repudiation of receipt (NRR).

In AS4, the `eb:Receipt` message serves both as a business receipt (its content is profiled in Clause 4), and as a reception indicator, being a key element of the reception awareness feature. No particular delivery semantics can be assumed however: the sending of an `eb:Receipt` only means the following, from a message processing viewpoint:

- The related ebMS user message has been received and is well-formed.

- The message has been successfully processed by the receiving MSH (i.e. not just "received"). Successful processing of a message means that none of the MSH operations needed over this message has generated an error.

- Because the latest steps of a message processing in the receiving MSH (leading to actual "delivery" to the message consumer) may vary greatly in their implementation from one implementation to the other, it is left to implementers to clarify to users at what exact step of the MSH processing flow the `eb:Receipt` is sent.

The meaning of not getting an expected `eb:Receipt`, for the sender of a related user message, is one of the following:

1. The user message was lost and never received by the receiving MSH.

2. The user message was received, but the `eb:Receipt` was never generated, e.g. due to a faulty configuration (P-Mode).

3. The user message was received, the `eb:Receipt` was sent back but was lost on the way.

See Clause 7.2.8 for AS4 usage rules about receipts.

NOTE: The use of the phrase 'business receipt' in AS4 is to distinguish the nature of the AS4/ebMS3 receipt as being sufficient for non-repudiation of receipt (NRR). In this sense it is very similar to the message disposition notification (MDN, IETF RFC 3798) response that is used by AS2 as a business receipt for non-repudiation. This receipt in AS4/ebMS3 contains the same information as the MDN, and thus distinguishes itself from the web services reliable messaging (sequence) acknowledgment.

5.6 Sub-channels for message pulling

Optionally, the sub-channel feature defined in clause 2 of the OASIS ebMS V3 Part 2 Advanced Features specification, for intermediaries in a multi-hop context, may be supported by an AS4 MSH. On the sending side of an AS4 exchange, this feature will apply to a sending AS4 MSH in the same way it applies to the edge intermediary in ebMS V3 Part 2.

In short, this feature allows for a producer application to submit messages intended for many receiving parties (i.e. different Client AS4 MSHs) over the same MPC, possibly covered by a single P-Mode. This MPC is configured for message pulling and will be authorized for different pulling endpoints (AS4 clients). This MPC is associated with a set of sub-channels to which different authorization credentials apply. Each client will be authorized to pull on its own sub-channel. Sub-channels are identified by an MPC identifier extension as illustrated below:

If the MPC identifier is an URI of the form:

```
http://sender.example.com/mpc123
```

A sub-channel of this MPC may have an identifier of the form:

```
http://sender.example.com/mpc123/subc42
```

The `@mpc` attribute value in the message is not altered so the message is still considered as sent over this MPC (mpc123). The sub-channel identifier is only apparent in the pull request messages generated by the receiver MSH.

The following additional P-Mode parameter is defined and shall be used when sub-channels are used:

- **Pmode[1].BusinessInfo.subMPCext**:: this parameter specifies the subchannel extension to be used. For example if **PMode[1].BusinessInfo.MPC** = "http://sender.example.com/mpc123" and **subMPCext** = "subc42" then the subchannel to pull from is: "http://sender.example.com/mpc123/subc42".

On the receiving MSH side, support for this feature means the ability to understand this P-Mode parameter in order to issue `eb:PullRequest` signals with the proper subchannel MPC value, while being able to process received pulled messages that contain the MPC value corresponding to the core channel.

5.7 Additional features errors

The error codes provided in Table 5 are extending the set of ISO 15000-1 error codes to support the AS4 additional features. They are to be generated and/or processed by an AS4 MSH depending on which feature is supported (i.e. depending on the conformance profile):

Error code	*Short description*	*Recommended severity*	*Category value*	*Description or semantics*
EBMS:0301	MissingReceipt	failure	Communication	A receipt has not been received for a message that was previously sent by the MSH generating this error.
EBMS:0302	InvalidReceipt	failure	Communication	A receipt has been received for a message that was previously sent by the MSH generating this error, but the content does not match the message content (e.g. some part has not been acknowledged, or the digest associated does not match the signature digest, for NRR).
EBMS:0303	Decompression-Failure	failure	Communication	An error occurred during the decompression.

Table 5: AS4 additional features errors

6 Complementary requirements for the AS4 multi-hop profile

6.1 General

The OASIS ebMS 3.0 Part 2, Advanced Features specification defines several advanced messaging features. One of these is a multi-hop feature that provides functionality to exchange ebMS messages through clouds of intermediaries, or *I-Clouds*. These intermediaries serve various purposes, including message routing and store-and-forward (or store-and-collect) connections. Intermediaries allow messages to flow through a *multi-hop* path and serve to interconnect (private or public) networks and clouds. To exchange AS4 messages through clouds of intermediaries, AS4 endpoints shall implement an optional profile of the OASIS ebMS 3.0 Part 2, Advanced Features specifications specified in this document in order to converse via ebMS intermediaries. This profile is complementary to the primary profiles defined in Clause 4. This complementary profile:

- simplifies the fine-grained endpoint configuration options of the OASIS ebMS 3.0 Part 2, Advanced Features specification to a single processing mode parameter (clause 6.4);
- extends the capability of AS4 endpoints to exchange messages in a peer-to-peer fashion to exchanges across intermediaries (clause 6.5).

Clause 6.2 provides the rationale and context for using AS4 and intermediaries. Clause 6.3 defines some general constraints and assumptions. Clause 6.4 presents the single additional processing mode parameter required for multi-hop. Clause 6.5 provides a minimal interoperability subset for AS4 endpoints in an *I-Cloud*.

6.2 Rationale and context

A key motivation for AS4 is to provide a simplified profile of ISO 15000-1 that allows small and medium-size enterprises (SMEs) to exchange messages using web services. Two situations can be distinguished:

- Situations where one partner in an exchange is an SME and the other is a larger organization. AS4 allows SME trading partners of a large organization to operate “client-only” endpoints and pull messages from a B2B gateway server operated by the large organization. That B2B gateway operates as a server and is addressable and available for pulling. These exchanges can be said to be *asymmetric*.
- Situations where all partners are SMEs, organized in collaborative SME B2B networks. In these situations there is no single larger partner that the other partners are organized around. These exchanges can be said to be *symmetric*.

When two endpoints exchange messages directly, they cannot both be client-only endpoints. Intermediaries can serve SME networks that use client-only endpoints by offering store-and-collect capabilities, just like Internet service providers (ISPs) offer mailbox services for email, value-added network (VAN) services offer document exchange services, and cloud-based file storage services offer secure temporary storage and exchange of large files.

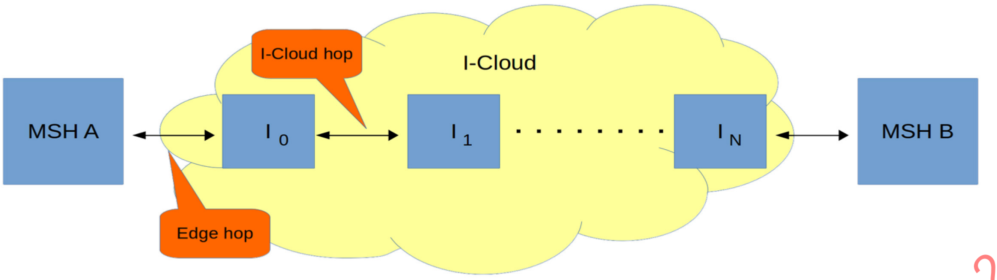

Figure 1: AS4 multi-hop messaging

As shown in Figure 1, in an AS4 multi-hop topology messages can be sent any time to MSH A or MSH B as long as the I-Cloud is able to forward messages to AS4 edge intermediaries I_0 and I_N, from which they can be pulled at a convenient time.

6.3 General constraints

This profile defines the following general constraints:

- Whether or not two AS4 endpoints exchange user messages in a peer-to-peer fashion or across an I-Cloud is determined by a single processing mode parameter.
- Sender and receiver MSH can diverge in some “init” and “resp” parameters (terminology from clause 2.7.2 of the OASIS ebMS 3.0 Part 2 Advanced Features specification), as some parameters in an exchange relate to the edge intermediaries, not to the ultimate destination MSH.
- Whether or not an AS4 endpoint returns related response signals (receipts, errors) in a peer-to-peer fashion or across an I-Cloud is not based on configuration, but is determined by how the associated user message was delivered:
  - Receipts and errors for user messages received directly are sent back directly.
  - Receipts and errors for user messages received through an I-Cloud are sent back through the I-Cloud.
- Edge intermediaries connect to AS4 endpoints as servers: they do not pull messages from endpoints.
- Pull signals from AS4 endpoints target AS4 edge intermediaries and are not forwarded across an I-Cloud.

- An AS4 edge intermediary that is capable of delivering a particular user message to an AS4 endpoint should be configured to provide initial reverse routing of any related signals (receipts, errors).

- There is no requirement to support WS-ReliableMessaging sequence lifecycle messages.

6.4 Processing mode parameter

In this profile, AS4 processors either operate in peer-to-peer exchange mode or exchange messages across intermediaries based on the value of a single processing mode parameter, defined in clause 6.4.2 of the OASIS ebMS 3.0 Part 2, Advanced Features Specification: **Pmode[1].Protocol.AddActorOrRoleAttribute**.

- If this value is set to *true* for a P-Mode, the ebMS header in AS4 user messages shall have a SOAP 1.2 `role` attribute and its value shall be set to the fixed value "http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/part2/200811/nextmsh".

- For AS4, the default value of this parameter is *false*, meaning that the SOAP 1.2 `role` attribute is not present. In SOAP 1.2, this is equivalent to the attribute being present with the value "http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver".

6.5 AS4 endpoint requirements

The ebMS 3.0 multi-hop feature defined in the ebMS 3.0 Part 2 Advanced Features specification specifies requirements on endpoints to be able to exchange messages in an I-Cloud. This subclause further constrains these requirements and provides a minimal interoperability subset for AS4 endpoints. AS4 endpoints supporting multi-hop messaging shall implement this minimal interoperability subset of the ebMS 3.0 Part 2 Advanced Features specification. The structure of this subclause follows the structure of clause 2.6 of the OASIS ebMS 3.0 Part 2, Advanced Features specification, which considers initiating messages and responding messages and uses the Web Services Addressing specification. Implementations shall therefore conform to the W3C Web Services Addressing specification.

Three types of initiating messages are distinguished:

- User messages: No special processing is required of an AS4 processor, other than being able to insert the `role` attribute with the appropriate value, subject to the selected processing mode, as specified in Clause 6.4.

- ebMS signal messages: This AS4 profile constrains this further as follows:

    - No `ebint:RoutingInput` reference parameter and no `role` attribute are added to `eb:PullRequest` messages.

    - AS4 endpoints shall not send initiating error messages.

- Non-ebMS messages: this situation is not relevant in the case of AS4 as it does not require support for web services protocols like the OASIS *Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.2*, OASIS Standard. For this reason there is no need to support initiating non-ebMS messages.

Clause 2.6 of the OASIS ebMS 3.0 Part 2, Advanced Features specification distinguishes the following types of responding messages:

- Non-ebMS Messages: this situation is not relevant in the case of AS4, because AS4 does not require support for web services protocols that return signal messages, such as reliable messaging acknowledgments.
- ebMS response user messages. This is handled in the same way as ebMS request user messages.
- ebMS signal messages. These messages are making use of WS-Addressing headers as defined in *Web Services Addressing 1.0 – Core* W3C Recommendation, under certain conditions.

This profile restricts or relaxes further the use of and/or support for these "wsa" headers for use with responding ebMS signal messages. Implementations shall conform to the W3C *SOAP Version 1.2 Part 1: Messaging Framework* W3C Recommendation.

- AS4 endpoints are not required to support `wsa:ReplyTo` header or `wsa:FaultTo` when generating responses.
- If the user message that the signal relates to does not contain a `role` attribute with a value of http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/part2/200811/nextmsh, processing of signals is as specified in ISO 15000-1 and in the other chapters of this document.
- If the user message that the signal relates to does contain a `role` attribute with a value of http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/part2/200811/nextmsh, a response signal shall contain:
    - a `wsa:To` header element with value http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/part2/200811/icloud;
    - a `wsa:Action` header element with value http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay.receipt or http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay.error;
    - a WS-Addressing reference parameter with content as specified in the subclause "Inferred RoutingInput for the reverse path" of subclause 2.6.2 of the OASIS ebMS 3.0 Part 2, Advanced Features specification conform W3C *SOAP Version 1.2 Part 1: Messaging Framework.* W3C Recommendation.

In the latter case, the value of the MPC attribute shall be set based on the value of the MPC attribute in the user message. If that value is not set, the default value http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC is assumed (as defined in ISO 15000-1:2021, 6.4.1):

- The MPC value for an AS4 receipt signal is formed by concatenating the string "`.receipt`" to the (default) MPC value of the received message.
- The MPC value for an AS4 error signal is formed by concatenating the string "`.error`" to the (default) MPC value of the message in error.

7 AS4 usage profile of ISO 15000-1

7.1 General

While the previous sections were describing messaging handler requirements for AS4 compliance (i.e. mostly intended for product developers), this clause is about configuration and usage options.

This clause is split in two major subclauses:

- **AS4 usage rules**: this subclause provides the rules for using messaging features in an AS4-compliant way.
- **AS4 usage agreements**: this subclause provides notes to the users on the main options left open by the AS4 profiles, which have to be agreed on in order to interoperate.

Both subclauses are about features that are under responsibility of the user when using an AS4-compliant product.

7.2 AS4 usage rules

7.2.1 Core components / modules to be used

Table 6 summarizes which functional modules in the ISO 15000-1 specification are required to be implemented by the AS4 profile, and whether or not these modules are actually profiled for AS4.

ISO 15000-1 Component name and reference	Profiling status
Messaging model (clause 5)	Usage: **Required** Profiled: **Yes**
Message pulling and partitioning (clause 6)	Usage: **Required** Profiled: **No** The profiling of QoS associated with pulling is defined in another module. The MPC and pulling feature itself are not profiled.
Processing modes (clause 7)	Usage: **Required** Profiled: **Yes**
Message packaging (clause 8)	Usage: **Required** Profiled: **Yes** The default business process defines acceptable defaults for role, service and action. Bundling options for message headers (piggybacking) are restricted.

ISO 15000-1 Component name and reference	**Profiling status**
Error handling (clause 9)	Usage: **Required** Profiled: **Yes** Some new error codes regarding reception awareness are added.
Security module (clause 10)	Usage: **Required** Profiled: **Yes** Guidance is provided regarding which part(s) of the message may be encrypted and included in the signature. Further guidance is provided on how to secure the `eb:PullRequest` signal and the preventing of replay attacks.
Reliable messaging module (clause 11)	Usage: **Not Required** Profiled: **No** This profile does not require the use of the reliable messaging module using either WS-ReliableMessaging or WS-Reliability. It relies instead on `eb:Receipts` for supporting a light reliability feature called reception awareness.

Table 6: AS4 core components / modules to be used

7.2.2 Bundling rules

Table 7 specifies bundling rules for AS4.

Scope of the profile feature	Defines bundling (or “piggybacking”) rules of ebMS MEPs, including receipts.
Specification feature	Message packaging
Specification reference	ISO 15000-1:2021, 8.2.5.
Profiling rule (a)	This profile supports the One-Way/Push MEP. Both synchronous and asynchronous transport channels for the response (`eb:Receipt`) are allowed by this profile.
Profiling rule (b)	This profile supports the One-Way/Pull MEP. When sending an `eb:Receipt` for this MEP, a receiving MSH conforming to this profile may bundle the `eb:Receipt` with any other ebMS message header (including an `eb:PullRequest` signal) or message body.

Table 7: AS4 bundling rules

7.2.3 Security element

Table 8 specifies the use of WSS features.

Specification feature	Use of WSS features
Specification reference	ISO 15000-1:2021, 10.2
Profiling rule (a)	When using digital signatures or encryption, an AS4 MSH implementation shall use the OASIS. *Web Services Security X.509 Certificate Token Profile 1.1*. OASIS Standard incorporating Approved Errata. 1 November 2006.
Alignment	• OASIS. *Web Services Security: SOAP Message Security 1.1.* OASIS Standard incorporating Approved Errata. 1 November 2006. • OASIS. *Web Services Security X.509 Certificate Token Profile 1.1*. OASIS Standard incorporating Approved Errata. 1 November 2006.

Table 8: AS4 security element

7.2.4 Signing messages

Table 9 specifies the signing of AS4 messages.

Specification feature	Digital Signatures for SOAP message headers and body
Specification reference	ISO 15000-1:2021, 10.3
Profiling rule (a)	AS4 MSH implementations shall use detached signatures as defined by the W3C *XML-Signature Syntax and Processing (Second Edition).* W3C Recommendation. 10 June 2008 when signing AS4 user or signal messages. Enveloped Signatures as defined by that specification are not supported by or authorized in this profile.
Profiling rule (b)	AS4 MSH implementations shall include the entire `eb:Messaging` SOAP header block and the (possibly empty) SOAP Body in the signature. The `eb:Messaging` header should be referenced using the “id” attribute.

Table 9: AS4 signing messages

7.2.5 Signing SOAP with attachments messages

Table 10 specifies signing SOAP with attachments messages.

Specification feature	Signing attachments
Specification reference	ISO 15000-1:2021, 10.4 W3C XML-Signature Syntax and Processing (Second Edition). W3C Recommendation.
Profiling rule (a)	AS4 MSH implementations shall use the Attachment-Content-Only transform when building application payloads using the W3C SOAP with Attachments specification. The Attachment-Complete transform is not supported by this profile.
Profiling rule (b)	AS4 MSH implementations shall include the entire `eb:Messaging` header block and all MIME body parts of included payloads in the signature.

Table 10: AS4 signing SOAP with attachments messages

7.2.6 Encrypting messages

Table 11 specifies message encryption for AS4.

Specification feature	Encrypting messages
Specification reference	ISO 15000-1:2021, 10.5 W3C XML Encryption Syntax and Processing. W3C Recommendation.
Profiling rule (a)	If an AS4 user message is to be encrypted, AS4 MSH implementations shall encrypt all payload parts. However, AS4 MSH implementations shall not encrypt the `eb:Messaging` header. If confidentiality of data in the `eb:Messaging` header is required, implementations should use transport level security.
Profiling rule (b)	If an AS4 user message is to be encrypted and the user-specified payload data is to be packaged in the SOAP `Body,` AS4 MSH implementations shall encrypt the SOAP `Body`.

Table 11: AS4 encrypting messages

7.2.7 Encrypting SOAP with attachments messages

Table 12 specifies encryption of SOAP with attachments messages.

Specification feature	Encryption of message attachments.
Specification reference	ISO 15000-1:2021, 10.6
Profiling rule (a)	If an AS4 user message is to be encrypted and the user-specified payload data is to be packaged in conformance with the the W3C SOAP-with-Attachments specification, AS4 MSH implementations shall encrypt the MIME body parts of included payloads.

Table 12: AS4 encrypting SOAP with attachments messages

7.2.8 Generating receipts

Table 13 specifies generation of receipts in AS4.

Specification feature	`eb:Receipt` signal messages
Specification reference	ISO 15000-1:2021, 10.13.2 (persistent signed receipt) ISO 15000-1:2021, 8.2.4.4, `eb:Messaging/eb:SignalMessage/eb:Receipt`
Profiling rule (a): receipts for reception awareness	When an `eb:Receipt` is to be used solely for reception awareness, the sender of the `eb:Receipt` shall contain a copy of the `eb:UserMessage` structure of the received AS4 message. The `eb:RefToMessageId` in the `eb:MessageInfo` group in the `eb:SignalMessage` contains the message identifier of the received message.

Profiling rule (b): receipts for non-repudiation of receipt (NRR)	When an `eb:Receipt` is to be used for non-repudiation of receipt, the content of the `eb:Receipt` element shall be a valid `ebbpsig:NonRepudiationInformation` element. When an `eb:Receipt` is to be used for non-repudiation of receipt (NRR), the sender of the receipt: • shall use `ds:Reference` elements containing digests of the original message parts for which NRR is required. Message parts shall not be identified using `ebbpsig:MessagePartIdentifier` elements. • shall sign the AS4 receipt signal message. When signed receipts are requested in AS4 that make use of default conventions, the sending message handler (i.e. the MSH sending messages for which signed receipts are expected) shall identify message parts (referenced in `eb:PartInfo` elements in the received user message) and shall sign the SOAP body and all attachments using the http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Content-Signature-Transform. The receiving message handler (i.e. the MSH generating the receipt signal) can reuse the `ds:Reference` elements from the `ds:SignedInfo` reference list in the received message. The sending message handler shall not encrypt any signed content before signing (Clause 10.7 in ISO 15000-1). If using compression in an attachment, the sending message handler shall sign the data after compression (see 4.2). Variations from default conventions can be agreed to bilaterally, but conforming implementations are only required to provide receipts using the default conventions described in this subclause.
Profiling rule (c)	An AS4 message that has been digitally signed shall be acknowledged with a message containing an `eb:Receipt` signal that itself is digitally signed. The `eb:Receipt` shall contain the information necessary to provide non-repudiation of receipt of the original message, as described in profiling rule (b). The digest(s) to be inserted in the `ebbpsig:MessagePartNRInformation` element(s) or the receipt, related to the original message parts for which a receipt is required, is contained in the signature information of the original message (`ds:SignedInfo` element), as only those parts that have been signed are subject to NRR. This means a receiving message handler may not have to compute digests outside its security module.

Table 13: AS4 generating receipts

Annex B contains a stylesheet that can be used to generate AS4 receipts.

7.2.9 MIME header and filename information

Table 14 specifies optional filename preservation in AS4.

Specification feature	Optional presence of a “filename” value in “Content-disposition” header on MIME body parts.
Specification reference	MIME specification, IETF RFC 2015.
Profiling rule (a)	The “Content-disposition” header on MIME body parts, when used, shall carry file name information. Implementations shall support the setting (when sending) and reading (when receiving) of “Content-disposition” header,
Profiling rule (b)	When end users wish to supply file names and have that information confidential, they should use TLS/SSL based encryption.

Table 14: AS4 MIME header and filename information

7.3 AS4 usage agreements

7.3.1 General

This subclause defines the operational aspect of the profile configuration aspects that users agree on, mode of operation, etc to interoperate.

All the user agreement options related to a specific type of message exchange instance (e.g. related to a specific type of business transaction) are controlled by the processing mode (P-Mode) parameters defined in ISO 15000-1. This subclause only lists the parameters that are particularly relevant to AS4.

7.3.2 AS4 usage agreement parameters

7.3.3 Controlling content and sending of receipts

Table 15 specifies content and sending of AS4 receipts.

Scope of the Profile feature	Choice among options in sending receipts.
Specification feature	`eb:Receipt` signal messages
Specification reference	ISO 15000-1:2021, 5.2

Usage profiling (a)	shall `eb:Receipt` signals be used for non-repudiation of receipt (NRR), or just act as reception awareness feature? For non-repudiation, the `eb:Receipt` element shall contain a well-formed `ebbpsig:NonRepudiationInformation` element. This is indicated by the new P-Mode parameter: - **PMode[1].Security.SendReceipt.NonRepudiation :** value = 'true' (to be used for non-repudiation of receipt), value = 'false' (to be used simply for reception awareness).
Usage profiling (b)	Receipts for One-Way/Push MEP: Both synchronous and asynchronous transport channels for the response `eb:Receipt` are allowed by this profile. (Values "Response" and "Callback") This option is controlled by the P-Mode parameter: - **PMode[1].Security.SendReceipt.ReplyPattern:** value = 'Response' (sending receipts on the HTTP response or back-channel). - **PMode[1].Security.SendReceipt.ReplyPattern:** value = 'Callback' (sending receipts using a separate connection.)
Usage profiling (c)	Receipts for the One-Way/Pull MEP: - **Pmode[1].Security.SendReceipt.ReplyPattern:** value = 'Callback' (sending receipts using a separate connection, and not bundled with an `eb:PullRequest`.)

Table 15: AS4 controlling content and sending of receipts

7.3.4 Error handling options

Table 16 specifies error handling options for AS4.

Specification feature	Error handling options
Specification reference	ISO 15000-1:2021, Clause 9
Usage profiling (a): Receiver-side error	All receiver-side error reporting options are left for users to agree on, including the choice to not report at all: - **PMode[1].ErrorHandling.Report.ReceiverErrorsTo:** recommendation is to report such receiver-side errors to the sender. Otherwise: report URI that is different from sender URI? - **PMode[1].ErrorHandling.Report.AsResponse:** recommendation for one-way messages (except when pulling is in use) is value="true": report errors on the back-channel of erroneous messages. Errors for pulled messages can only be reported on a separate connection. - **PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer:** (true / false) for controlling escalating the error to the application layer.

Usage profiling (b): Reception awareness errors	What is the behavior of a sender that failed to receive a receipt (even after message retries)? • No error reporting (in case no reception awareness required). • Error reporting from the sender MSH to its message producer (application-level notification). Error type: `EBMS:0301:MissingReceipt` (see Clause 5.7, additional features errors) P-Mode parameter: • **PMode[1].ErrorHandling.Report.MissingReceiptNotifyProducer**: (new) true if (b), false if (a). • **PMode[1].ErrorHandling.Report.SenderErrorsTo**: (in case an error should be sent about such failures – e.g. to a third party if not to the original receiver of the non-acknowledged user message.)
Usage profiling (c): Error about receipts	How are errors about receipt messages reported? P-Mode parameters: • **PMode[1].ErrorHandling.Report.SenderErrorsTo:** reporting URI that is different from receiver URI? • **PMode[1].ErrorHandling.Report.AsResponse:** (true / false) In case of receipts already sent over the HTTP back-channel, can only be "false" meaning such errors will be sent over separate connection. • **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer:** (true / false) for controlling escalating the error to the application layer.

Table 16: AS4 error handling options

7.3.5 Securing the pull request

Table 17 specifies securing the AS4 Pull request.

Specification feature	Pulling authorization options
Specification reference	ISO 15000-1:2021, clause 10.12.2. AS4 conformance profile authorization options (clauses 4.2.2, 4.3.2 and 4.4.2)

Usage profiling (a)	An AS4 sending MSH may authenticate a receiving MSH that sends an `eb:PullRequest` in two ways: 1. (Option 1 in 4.2.2, 4.3.2 and 4.4.2) Use of the WSS security header targeted to the “ebms” actor, as specified in clause 10.11 of ISO 15000-1, with the `wsse:UsernameToken` profile as specified in the OASIS. *Web Services Security UsernameToken Profile 1.1* OASIS Standard. 2. (Option 2 in 4.2.2 and 4.3.2) By using the OASIS *Web Services Security X.509 Certificate Token Profile 1.1*. OASIS Standard incorporating Approved Errata coupled with the message partition channel that a pull signal is accessing for pulling messages. P-Mode parameters: • **PMode.Initiator.Authorization:** shall be set to true (the initiator of a Pull request shall be authorized). • **PMode.Initiator.Authorization.username:** (for option 1) • **PMode.Initiator.Authorization.password:** (for option 1) • **PMode[1].Security.PModeAuthorize:** shall be set to true in the PMode leg describing the transfer of a pulled message. • **PMode[1].Security.X509.sign**: (for option 2) • **PMode[1].Security.X509.SignatureCertificate**: (for option 2) In option (2), the P-Mode parameters about X509 are controlling both the authentication of `eb:PullRequest` signals and authentication of other user messages.
Usage profiling (b)	`eb:PullRequest` signals: are they sent using the HTTPS transport protocol with optional client-side authentication? P-Mode parameter: • **PMode[1].Protocol.Address**: The URL scheme will indicate whether HTTPS is used or not.

Table 17: AS4 securing the PullRequest

7.3.6 Reception awareness parameters

Table 18 specifies AS4 message retry and duplication detection options.

Specification feature	Message retry and duplicate detection options
Specification reference	AS4 profile (this document), AS4 additional features (clause 5)
Usage profiling (a): Sender options	In case reception awareness is used: what is the behavior of a sender that did not receive a receipt? (a) No message retry. (b) Resend the message. retry parameters: to agree on: (1) retry count, (2) retry frequency P-Mode parameters (additional to those defined in ISO 15000-1:2021): • **PMode[1].ReceptionAwareness:** (true / false) • **PMode[1].ReceptionAwareness.Retry:** (true / false) • **PMode[1].ReceptionAwareness.Retry.Parameters:** (contains a composite string specifying: (a) maximum number of retries or some timeout, (b) frequency of retries or some retry rule.
Usage profiling (b): Receiver options	Is duplicate detection enabled? (a) No. Duplicates are not detected. (b) The receiver detects and eliminates duplicates based on `eb:MessageInfo/eb:MessageId`. P-Mode parameters (additional to those defined in ISO 15000-1:2021): • **PMode[1].ReceptionAwareness.DuplicateDetection:** (true / false) • **PMode[1].ReceptionAwareness.DuplicateDetection.Parameters**

Table 18: AS4 reception awareness parameters

7.3.7 Default values of some P-Mode parameters

Table 19 specifies default values of some P-Mode parameters for AS4.

Specification feature	Default values and authorized values for main P-Mode parameters.
Specification reference	ISO 15000-1:2021, D.3
Usage profiling (a)	**PMode.MEP** parameter will be constrained to the following value: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay
Usage profiling (b)	**PMode.MEPbinding** parameter will be constrained to the following values: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/pull

Usage profiling (c)	**PMode.Initiator.Role** parameter will have the following default value: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator
Usage profiling (d)	**PMode.Responder.Role** parameter will have the following default value: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder
Usage profiling (e)	**PMode[1].BusinessInfo.Service** parameter will have the following default value: http://docs.oasis-open.org/ebxml-msg/as4/200902/service *This default is a P-Mode content default: absence of the P-Mode itself will cause the default value defined in the ISO 15000-1 (clause 7.4) to apply. This value is usually enforced by the MSH implementation itself.*
Usage profiling (f)	**PMode[1].BusinessInfo.Action** parameter will have the following default value: http://docs.oasis-open.org/ebxml-msg/as4/200902/action *This default is a P-Mode content default: absence of the P-Mode itself will cause the default value defined in the ISO 15000-1:2021, 7.4 to apply. This value is usually enforced by the MSH implementation itself*
Usage profiling (g)	**PMode[1].Reliability** parameters are not supported by this profile

Table 19: AS4 P-Mode parameter default values

7.3.8 HTTP confidentiality and security

Table 20 specifies HTTP confidentiality and security options for AS4.

Specification feature	HTTP security management and options This table is intended as a guide for users, to specify their own agreements on HTTP confidentiality and security.
Specification reference	ISO 15000-1:2021, Clause 10 and D.3.6.
Usage profiling (a)	Is HTTP transport-layer encryption required? What protocol version(s)?
Usage profiling (b)	What encryption algorithm(s) and minimum key lengths are required?
Usage profiling (c)	What certification authorities (CAs) are acceptable for server certificate authentication?
Usage profiling (d)	Are direct-trust (self-signed) server certificates allowed?

Usage profiling (e)	Is client-side certificate-based authentication allowed or required?
Usage profiling (f)	What client certificate authorities are acceptable?
Usage profiling (g)	What certificate verification policies and procedures shall be followed?

Table 20: AS4 use of HTTP confidentiality and security

7.3.9 Deployment and processing requirements for CPAs

Table 21 specifies collaboration protocol agreement (CPA) access for AS4.

Usage profile feature	CPA access
Usage profiling (a)	Is a specific registry for storing CPAs required? If so, provide details.
Usage profiling (b)	Is there a set of predefined CPA templates that can be used to create given parties' CPAs?
Usage profiling (c)	Is there a particular format for file names of CPAs, in case that file name is different from CPAId value?

Table 21: AS4 CPA access

7.3.10 Message payload and flow profile

Table 22 specifies message payload a flow profile for AS4.

Usage profile feature	Message quantitative aspects
Usage profiling (a)	What are typical and maximum message payload sizes that shall be handled? (maximum, average)
Usage profiling (b)	What are typical communication bandwidth and processing capabilities of an MSH for these Services?
Usage profiling (c)	Expected volume of message flow (throughput): maximum (peak), average?
Usage profiling (d)	How many payload containers shall be present?
Usage profiling (e)	What is the structure and content of each container? [List MIME Content-Types and other process-specific requirements.] Are there restrictions on the MIME types allowed for attachments?
Usage profiling (f)	How is each container distinguished from the others? [By a fixed ordering of containers, a fixed manifest ordering, or specific Content-ID values.]. Any expected relative order of attachments of various types?
Usage profiling (g)	Is there an agreement that message part filenames shall be present in MIME

	Content-Disposition parameter?

Table 22: AS4 payload and flow

7.3.11 Additional deployment or operational requirements

Table 23 specifies additional operational or deployment conditions.

Usage profile feature	Operational or deployment conditions
Usage profiling (a)	Operational or deployment aspects that are object to further requirements or recommendations.

Table 23: AS4 deployment and operations other requirements