ENTSOG AS4 Agreements and Agreement Updates

Version Rev_1 – 2017-01-09

Table of contents

1 Introduction

The current version Rev_3 of the ENTSOG AS4 profile [AS4TSO] introduces the concept of agreements and agreement updates to the ENTSOG AS4 profile. The agreement concept is an established ebXML concept for (versions of) messaging technical configurations that is applied to AS4 [AS4]. In ENTSOG AS4, it is used for certificate updates. Agreement Update, a generalisation of peer-to-peer online certificate exchange, is standardised at OASIS in the separate ebCore Agreement Update specification [AU]. Together, agreements and agreement updates enable a flexible mechanism for exchanging certificates and for managing and updating security configurations for signing and encryption of ENTSOG AS4 messages.

Software vendors should note that as AS4 products for use with the ENTSOG profile will be required to support these new features from 1 July 2017, to make it possible to implement these features in production. This document explains some of the consequences of the use of these concepts with AS4 and the impact and requirements on AS4 implementations and their deployments, in order to allow solution providers to determine if their products need any enhancements or adjustments to support this features and to make any adjustments in time for use by their users.

The audience for this document are providers of solutions supporting the AS4 Usage Profile for TSO [AS4TSO] and implementers of those solutions within the gas industry. This document is informative only. It is only intended to help implementers, but does not replace or overrule the functionality specified in the ENTSOG AS4 profile and the underlying AS4 and ebCore Agreement Update specifications. This document complements the “Setting Up and AS4 System” document [AS4SETUP].

2 Agreements

2.1 Agreements and P-Modes

In ebMS3, **AgreementRef** is one of the fields in the **Messaging** header. The content of the **AgreementRef** header is configured using the **PMode.Agreement** processing mode parameter ([EBMS3], section 5.2.2.7). In ebMS3, **AgreementRef** is optional. However, in the ENTSOG AS4 profile, its use is mandatory and the P-Mode parameter must therefore be set.

Depending on how the P-Modes (processing modes) are configured, in ebMS3 there may be, for a given combination of the parameters **Pmode.Initiator.Party** (controlling the **From** header value), **Pmode.Responder.Party** (the **To** header), **Pmode[1].BusinessInfo.Service** (the **Service** header) and **Pmode[1].BusinessInfo.Action** (the **Action** header), multiple P-Modes with distinct values for **Pmode.Agreement** (which controls the **AgreementRef** header). This functionality is used in the ENTSOG AS4 profile and must be supported by AS4 products used to exchange ENTSOG AS4 messages.

In this document we will discuss implementation aspects of agreements and P-Mode selection for incoming and outgoing messages. For these cases we will look at configurations and updates to configurations.

2.2 Certificates and P-Modes

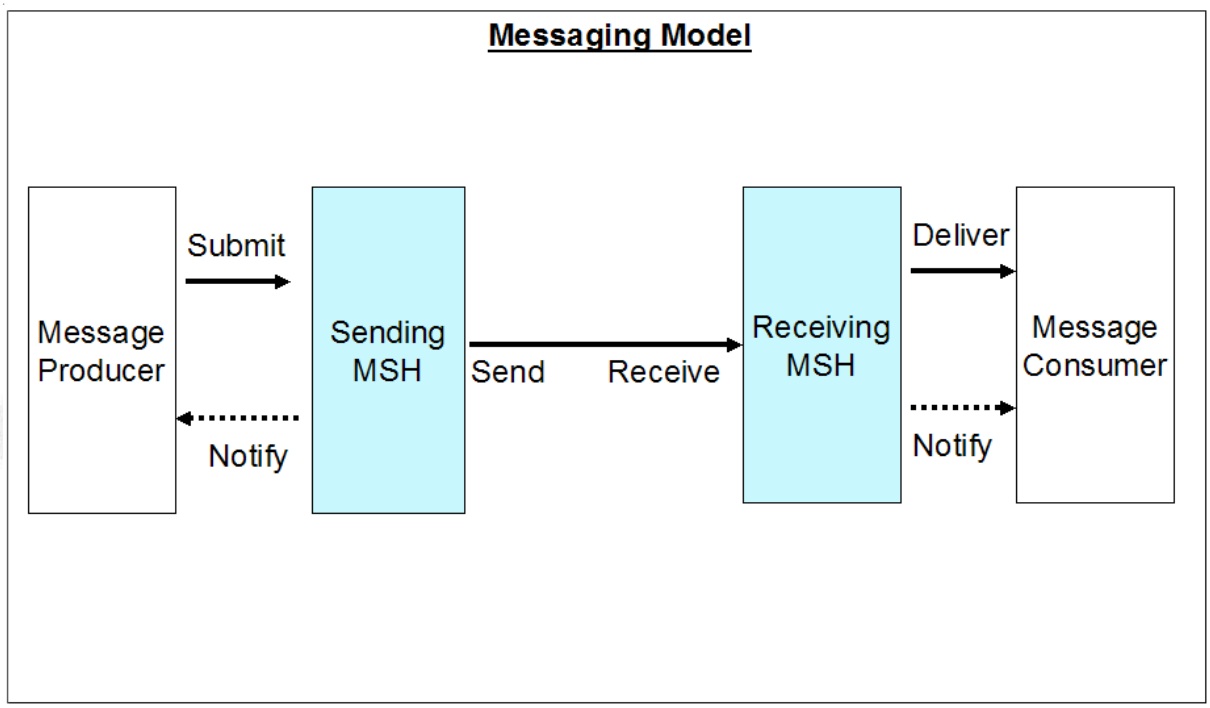
P-Modes and P-Modes for ENTSOG AS4 in particular specify the certificates used to secure the AS4 message. The **Pmode[1].Security.X509.Signature.Certificate** specifies the signing certificate and **Pmode[1].Security.X509.Encryption.Certificate** the encryption certificate.

Different P-Modes for the same communication partner may therefore use different certificates. In the ENTSOG AS4 profile, certificates are associated with agreements. It is not sufficient for a product to allow certificates to be specified for communication partners. Instead, certificates are specified for P-Modes and linked not only to the communication partner but also to the **Pmode.Agreement** parameter, reflecting one configuration with that partner.

P-Modes that specify use of certificates have a validity interval that is dependent on the validity interval of those certificates. The ENTSOG AS4 profile is designed to support certificate rollover by having overlapping validity intervals for certificates and for the P-Modes that use them. AS4 products for use with ENTSOG AS4 must support this feature.

2.3 Incoming Messages

The expected handling of agreements in incoming messages has an impact on the way an AS4 MSH implements the **Receive** operation (see Figure 1 in section 2.1.1, Messaging Model, of ebMS3 Core [EBMS3]).

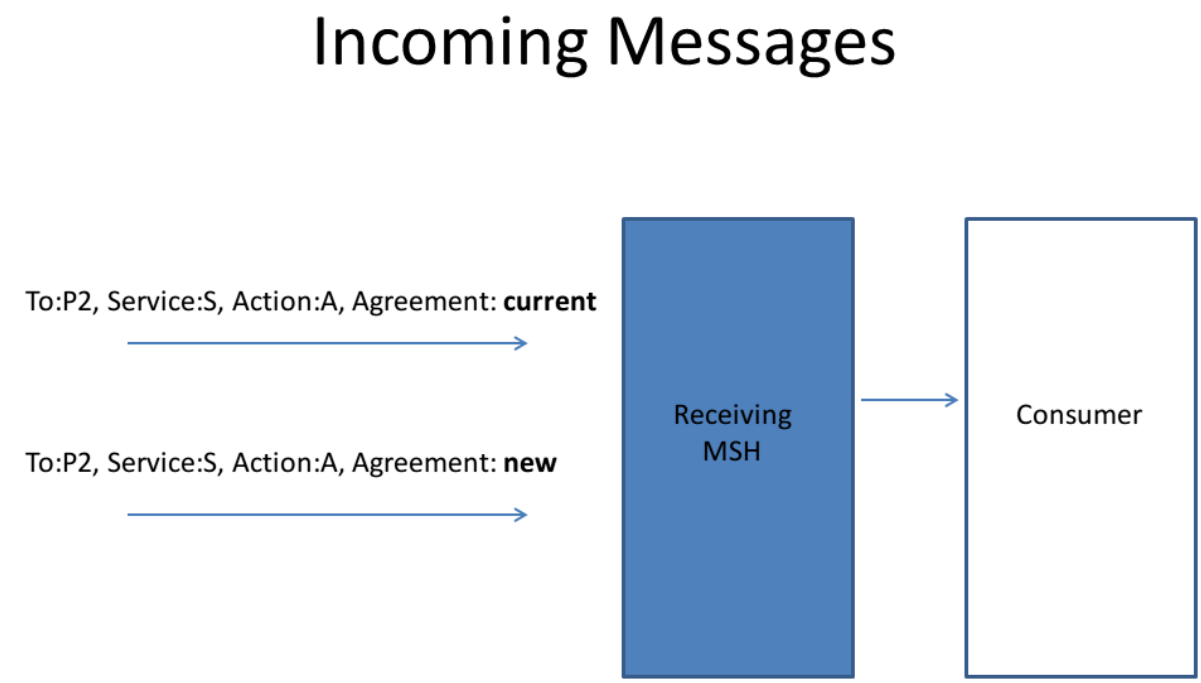


The ENTSOG AS4 profile requires AS4 products to be able to set the **AgreementRef** header value and requires products to include the **AgreementRef** header value when looking up the P-Mode that controls the processing of an incoming message ([AS4TSO], section 2.2.3.1). This is a constraint on implementations, as the ebMS3 and AS4 specifications are underspecified as to how P-Mode are determined and this behaviour is therefore implementation-specific (see https://issues.oasis-open.org/browse/EBXMLMSG-48). Not all currently available AS4 products support these requirements. Vendors looking to provide solutions to the gas industry for the ENTSOG AS4 profile are strongly encouraged to verify that their products support these requirements.

The ENTSOG AS4 Usage Profile ([AS4TSO], section 2.3.2) also specifies that, for a specific value for **Pmode.Agreement**, it must be possible to unambiguously determine the signing and encryption certificates of the parties involved. For two messages with the same values for all ebMS3 headers except for **AgreementRef**, different P-Modes are expected to have been configured and are to be inferred from incoming messages. It also means that if one or both parties change(s) one or more certificates, a new P-Mode is expected to be created and agreed by the parties with a distinct agreement identifier. Taken together, these constraints imply that, for any incoming AS4 message, a compliant AS4 MSH is expected to be able to infer unambiguously the certificates that are expected to be used.

The following diagram illustrates P-Mode inference for the ENTSOG AS4 profile. It assumes that P-Modes for a current agreement and a new agreement are deployed at the same time and that there are two P-Modes deployed that have the same values P2, S, A for To PartyId, Service and Action, respectively, but differ in the values for Agreement (**current** or **new**). The P-Mode configuration to use with the incoming messages can be found by the receiving MSH by using the **AgreementRef** header as one of the keys to find the P-Mode that has this value

for the **Pmode.Agreement** parameter.



If an incoming message uses (a) different certificate(s) than is specified for the Agreement (and therefore the P-Mode), the message must be rejected using an **EBMS:0003** error, because it is inconsistent with the processing mode of the MSH.

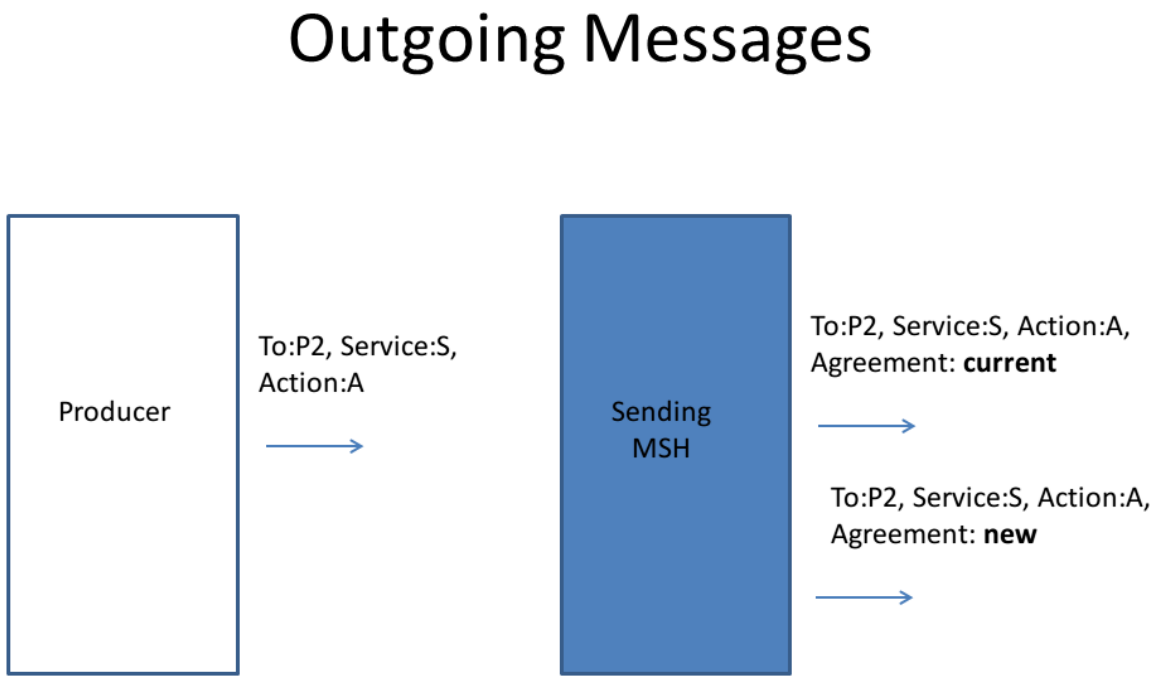
When deploying a new agreement, for incoming messages there is no need to immediately deactivate all P-Modes related to the current agreement. To the contrary, it can be useful, for incoming messages, to deploy the new P-Mode set in parallel to the current set. Any messages still using the current agreement (including any delayed retries of previously unsuccessful transmission) can then still be processed successfully. This is a deployment decision, not a product requirement, but it is an option that the product must enable.

2.4 Outgoing Messages

The expected handling of agreements in outgoing messages relates to the way an AS4 MSH implements the **Submit** operation (see Figure 1 in section 2.1.1, Messaging Model, of ebMS3 Core [EBMS3]). The **Submit** operation in ebMS3 is an abstract operation, the implementation of which is left to implementations. This is intentional, as ebMS3 is about B2B messaging whereas the backend interfaces are specific to enterprises, middleware, frameworks programming languages etc. all of which are out of scope for ebMS3 as a standard. Likewise, ENTSOG AS4 also does not provide a specification for this operation.

Agreements in ENTSOG AS4 are aspects of the configuration of the ebMS3 MSH and do not relate to any business functionality or semantics and are therefore not relevant to the message **Producer** or **Consumer**. This means that the **Agreement** is not a parameter that business applications or middleware should have to specify when, in the role of a messaging **Producer**, they **Submit** metadata and a payload to be sent to a communication partner to the MSH. Products must support this and should only require the **Producer** to provide enough metadata to allow the MSH to infer the P-Mode to be used for sending. This metadata is left to implementations, but is likely to include at the minimum the destination **PartyID** and party type, **Service** and **Action**.

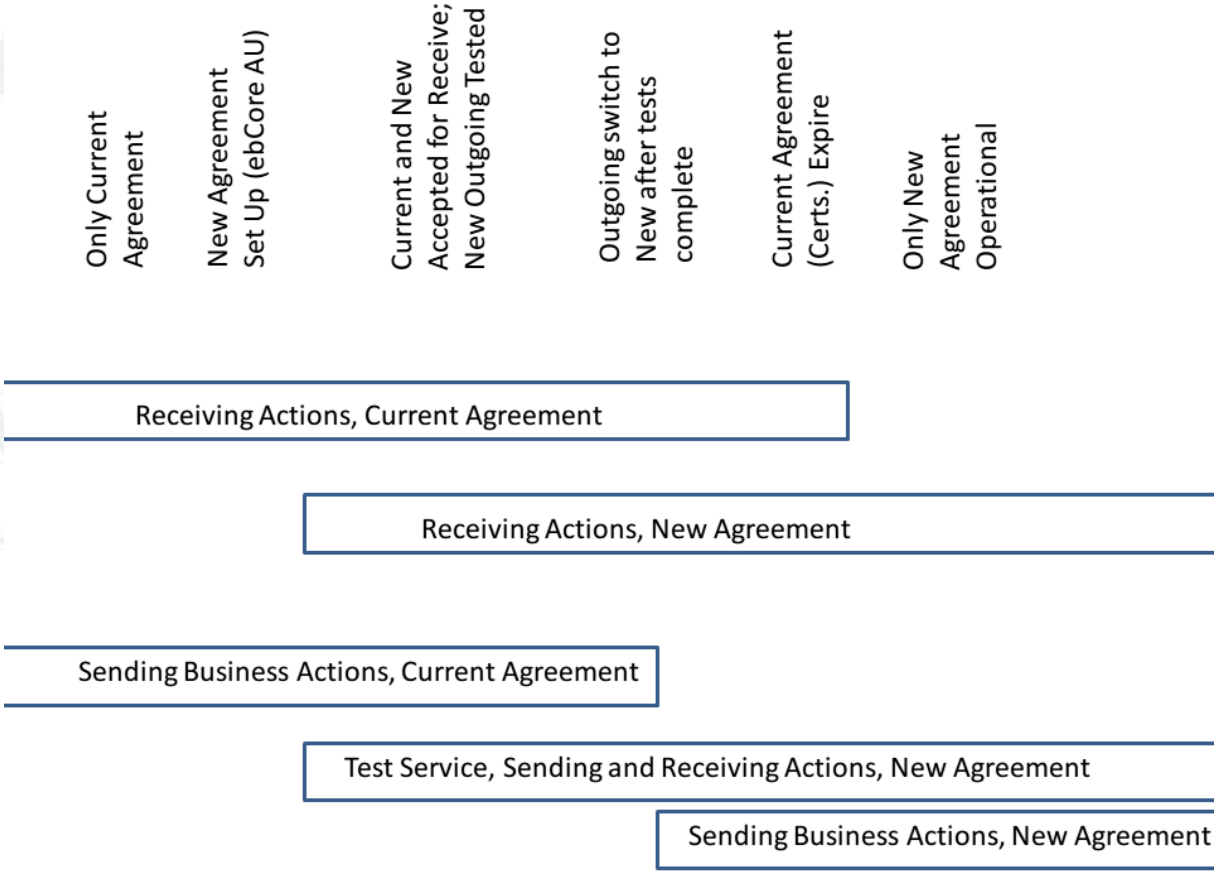
In principle, in situations where there are multiple deployed P-Modes with different **Agreement** identifiers for different sets of certificates, the metadata provided with the **Submit** call is not sufficiently unambiguous to allow the sending MSH to determine which P-Mode to use. In these situations, the logic to select one of multiple candidate P-Modes is left to implementations.



If the MSH is configured as described in section 3.4 of ENTSOG's "How to Set Up an AS4 System" [AS4SETUP], there actually need not be more than one applicable agreement. That document recommends that, before using the agreement for business messages, it should be used with the ebMS3 test service, as profiled in the ENTSOG AS4 profile. The test message validates the correct configuration of the AS4 MSH and the environment in which it

is deployed, including properly configured certificates and properly configured firewalls. If there is a new agreement, for outgoing messages, initially only the outgoing "test" message needs to be deployed. Business messages (with business values for **Service** and **Action**) at this point can still use the old agreement. Once use of the test service has successfully validated the new agreement, the other P-Modes for the new agreement, covering outgoing business messages, can be deployed and the P-Modes for outgoing messages for the previous agreement deactivated. This is a deployment decision, not a product requirement.

The timeline in the following diagram illustrates this.



Initially, only one agreement is deployed, the current agreement. Its P-Modes are used for both incoming and outgoing messages.

Then, a new agreement is formed, using ebCore Agreement Update. The AU protocol guarantees an agreed activation date for the agreement, so that parties can prepare the deployment of the relevant P-Modes.

Parties must be able to receive messages using the new agreement P-Modes from the agreed activation date and time.

In principle, they can also use P-Modes for outgoing messages from that in time. However, the diagram illustrates that, to be sure there are no issues with the new certificates, it is

recommendable to start with the test service. Once the test is successful, the new P-Modes for outgoing business messages are safe to be used.

3 Agreement Updates

3.1 ebCore Agreement Update

The v2r0 version of the ENTSOG AS4 profile mandates AS4 implementations to provide an API to manage AS4 configuration, but does not mandate a specific API. The new v3r0 version more specifically requires the API to provide all functionality to implement the OASIS ebCore Agreement Update specification [AU]. The latest version of the ENTSOG AS4 profile mandates support for Agreement Update from 01.07.2017, and uses the protocol to update certificates.

Sections 4.1 and 4.4 of the Agreement Update specification provide information on the use of ebCore Agreement Update with ebMS3 and AS4. In essence, support for agreement update for certificate updates requires a product to be able to configure a new set of P-Modes using a set of currently deployed P-Modes as input, copying all parameter-value settings except for those parameters that reference a certificate that is replaced. How the P-Mode configuration is represented in the product (as database tables, XML files etc.) is left to the product.

Agreement Update in ENTSOG AS4 uses AS4 itself to communicate the configuration updates. For the MSH, these messages are just payloads targeting a specific pre-identified service. The XML structures defined in Agreement Update can be processed within or outside of the AS4 product, as long as any updates to the MSH configuration can be done using the product's configuration API.

4 References

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

[AS4SETUP] Setting up an AS4 System. ENTSOG INT0697-161115_Rev_2 http://www.entsog.eu/public/uploads/files/publications/INT%20Network%20Code/2015/INT0697_150625_Setting%20up%20an%20AS4%20System%20v1r0.pdf

[AU] OASIS ebCore Agreement Update Specification Version 1.0. OASIS Committee Specification. http://docs.oasis-open.org/ebcore/ebcore-au/v1.0/

[AS4MT] ENTSOG AS4 Mapping Table. http://www.entsog.eu/publications/as4#ENTSOG-AS4-MAPPING-TABLE

[AS4TSO] ENTSOG AS4 Usage Profile for TSOs. ENTSOG INT 0488-161511_Rev_3 http://www.entsog.eu/publications/as4#AS4-USAGE-PROFILE

[EBMS3] OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features. OASIS Standard. 1 October 2007. http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/

193 # 5 Revision History

Revision	Date	Editor	Changes Made
v0r1	2016-11-07	PvdE	First Draft for discussion
V0r2	2016-12-05	PvdE	Feedback from November ITC KG meeting; diagrams from the presentation included.
Rev_0	2016-12-13	ITC KG	For Approval by INT WG
Rev_1	2017-01-09	ITC KG	For Approval by INT WG

194