FINAL

EFET COMMUNICATION STANDARD

REL1

CREATED BY EFET

Revision History

Version	**Date**	**Changes**	**Author of changes**
0.1a	Sep 2009	Initial version for discussion within the EFET IT Architecture WG.	EFET IT Architecture Work Group
0.2a	Dec 2009	Updated following feedback from the EFET IT Architecture WG.	EFET IT Architecture Work Group
0.3a	Feb 2010	Updated following feedback from the EFET IT Architecture WG.	EFET IT Architecture Work Group
0.3b	Feb 2010	Updated following internal review.	EFET IT Architecture Work Group
0.3c	Apr 2010	Updated following feedback from the EFET IT Architecture WG, issued to software providers.	EFET IT Architecture Work Group
0.4a	June 2010	Updated to incorporate feedback from implementers and changes to support Hub processing	EFET IT Architecture Work Group
1.0	September 2010	Final version	EFET IT Architecture Work Group

Copyright notice

This document and the information contained herein is provided on an "as is" basis.

EFET DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Content

List of Figures

1 Executive Summary

1.1 The Need for EFET Standards

Problem Definition

Communication is an essential key to the successful integration of business processes. Successful communication requires that the communicating parties **speak the same language.** This fact is as important in electronic communication as it is in face to face communication.

As volumes increase in energy trading, business transactions are occurring more rapidly, and trading volumes are growing, traditional means of communication like phone and fax are necessarily being replaced as a core communication medium by automated electronic communication.

Increasingly energy trading companies are looking towards the integration of internal and external business processes, with the eventual aim of straight-through processing. This is to enhance process efficiency, as well as to reduce operational risk, both of which reduce overall transaction costs.

The energy trading industry does not have in use widely accepted electronic communication standards. Like the financial industry there are some standards for specific parts of the industry, but the fragmentation is arguably even higher in the energy trading industry. Currently each service provider (exchanges, broker platforms, clearing houses, matching services, etc.) and each software vendor use their own proprietary "standard", requiring implementation of a different interface and cumbersome translation for each of these "standards". This results in a costly and risky "spaghetti" network of interfaces.

To solve the business process integration problem, common electronic communication standards (a common language) must be established and adopted across the energy industry. The messages and processes that need standardisation in the energy trading industry include Trade Confirmations, Scheduling and Logistics, Clearing and Settlement, and Quotes.

By standardising the exchange of this information and the corresponding processes both internally and externally, companies can reduce costs and streamline business processes. Standardisation must be driven by the industry itself, and coordinated and governed by an accepted industry wide neutral body.

The Solution: EFET Standards

The European Federation of Energy Traders (EFET) is an independent industry wide body that can coordinate the creation and maintenance of industry standards. Under EFET governance, project work groups comprising members from the industry have been convened that are specifically responsible for defining the **EFET Communication Standard**.

EFET's approach is to define a dedicated standard for each business process and a single common communication standard to provide a common foundation for electronic data exchange required by each business processes. This structure is visualised in Figure 1 Structure of the EFET Standards.

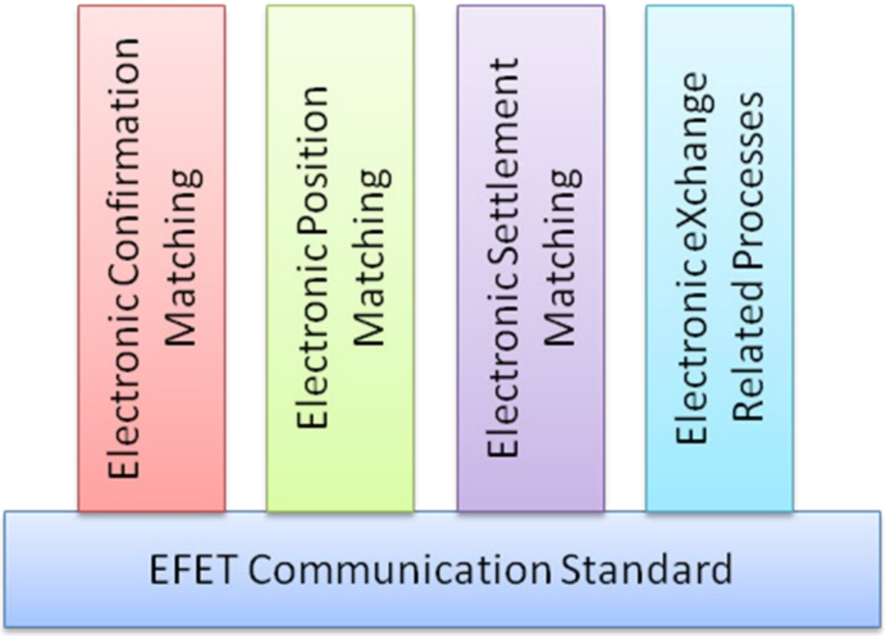
Electronic Confirmation Matching
Electronic Position Matching
Electronic Settlement Matching
Electronic eXchange Related Processes
EFET Communication Standard

Figure 1 Structure of the EFET Standards

There are currently four business process standards defining the structure of electronic messages and how these electronic messages are exchanged. The EFET Communication Standard defines the technical communication protocol for electronic messages exchanged in the energy trading environment, and therefore can be considered a general standard.

1.2 Conclusions

Communication is an essential key to the successful integration of business processes. To solve the business process integration problem, common electronic communication standards (a common language) must be established and adopted across the energy industry.

Benefits of the EFET Communication Standard include the reduction of technical barriers to communication within the European energy market between all market players including traders, brokers, transmission system operators and financial service providers such as clearing houses and clearing banks. A common, widely adopted open communication standard also supports competition through removal of barriers to market entry for new service providers as well as facilitating both peer-to-peer communication and central service provision.

It is expected that further standardisation work will be done to facilitate the electronic exchange of data to further increase efficiency in the European Energy Industry. This is out of the scope of this document.

2 Overview

2.1 Roles and Responsibilities in Standardisation

The EFET Board oversee all the activities undertaken or sponsored by EFET. Responsibility for coordination of Back Office activities has been delegated to the Back Office (BO) Group. Responsibility for coordination of IT activities has been delegated to the IT Architecture (ITA) Work Group. Other project work groups exist to focus on and develop standards for specific processes, e.g. confirmation matching. Each business standard is sponsored by the EFET Board, controlled by the BO Group and comprises specialist personnel from business and IT functions. The communication standard is the responsibility of the ITA Work Group.

2.2 Version Control

EFET standards documentation comprises a single document with chapters and sections.

1) The single document shall be a release item under control of the relevant EFET work group on behalf of the EFET Board with major versioning e.g. 1.0, 2.0, 3.0.
2) Each chapter shall be a configuration item within the single document controlled by either the IT Architecture Work Group and audited via the Revision History between major releases leading to intermediate versioning e.g. 1.1, 1.2, 1.3. (Also release version with change bars)

Note that draft versions are signified by using a letter: e.g. 1.1a.

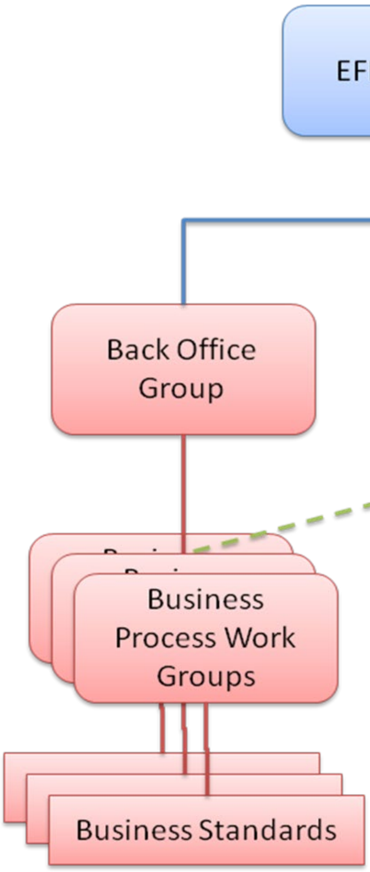

Figure 2 Organisation of EFET Work Groups and Responsibilities for Standards Documentation

3 EFET Communication Protocol and Interfaces

This chapter presents the communication protocol used to transport all EFET standard business process messages. The communication protocol can be extended to other messages as require.

3.1 Business Requirements

The communication standard defined here seeks to support the business requirement for flexibility within the standard. As well as allowing for diversity in B2B communication between different organisations, this requirement also relates to broader application of the standard to support other processes, message sets and potentially communication with (and within) other segments of the energy market. It is accepted that such flexibility in the standard means more complexity in implementation of EFET business processes, however this consideration is outweighed by the requirement for the standard to provide a solid foundation for broad based B2B interoperation, as well as to support diversity in B2B communication for EFET business processes and beyond.

3.2 EFET Communication Model

The EFET communication model is designed to be simple to implement, but flexible enough to enable new business processes to be added as required. The model can be summarised as a **message originator** sends a message containing a payload to a **message recipient**.

The standard does not mandate or assume anything about

- processes which create or consume the payload
- the format of the payload
- the security of the payload

This is defined at a process specific level, see sub-section Payload in section 3.5.

By taking a payload agnostic approach, new processes can be easily added, including support for non-EFET processes (e.g. EASEEgas). The model also enables the use of a service partner to route messages on behalf of end user organisations.

Each business process enables users to define where messages will originate from and be sent to. Note that it is not mandated that this is the same end point for all processes. The implementation of process end points is out of scope for this document.

3.3 Security and Reliability Requirements

Most Internet protocols only support channel encryption/authentication (as in the case of HTTPS), not end-to-end document encryption/authentication from application to application. To secure the path from application to application via the internet, DMZs and firewalls of the communicating organisations, the EFET Communication Protocol must meet the following security requirements:

- **Authentication** (verification of identity)
- **Confidentiality** (document encryption)
- **Data Integrity** (document signing)
- **Non-repudiation** (proof of document receipt)

Authentication ensures that message originators are whom they purport to be.

Confidentiality ensures that messages can be read only by authorised entities.

Data integrity ensures that messages are unchanged from their source and have not been accidentally or maliciously altered.

Non-repudiation ensures that strong and substantial evidence is available that a message has been sent or received.

Furthermore in order for these security measures to take full effect the EFET Communication Protocol must implement reliable messaging over unreliable underlying transport mechanisms, i.e. it must guarantee that

- an originator knows if a message has been delivered or not,
- a message is received at most once by a recipient.

3.4 Existing Standards Used

All external reference documents are listed in Appendix A.

The EFET Communication Standard is based on the ebXML Message Service Protocol defined in the ebXML Message Service Specification Version 2.0 (Appendix A.[1]).

ebXML Messaging is in use as an existing standard and is already widely accepted by different industries and provides the required security related functions.

3.5 EFET Communication Protocol Profile

Motivation

The ebXML Message Service Specification defines a generic protocol for exchanging electronic business documents that is designed to work for many different scenarios and use cases. This is achieved by introducing a number of options for business applications implementing this specification.

Hence each electronic business process based on ebXML Messaging must make a number of choices about the exact use of the ebXML Message Service Protocol in order to achieve compatibility and interoperability between different business applications implementing this process.

The EFET Communication Standard defines and fixes a certain subset of these choices to be shared by all electronic business process specifications based on it. The definition of the remaining choices is passed on to each individual electronic business process specification.

Definition of Terms

An "EFET Communication Protocol Profile" specifies a choice for every optional item or degree of freedom given in the ebXML Message Service Specification Version 2.0.

Every EFET Communication Protocol Profile is split into two parts, one process-specific part called an "EFET Communication Business Process Profile" and one fixed and shared part called "EFET ebXML Messaging Standard Profile".

For one particular electronic business process the associated EFET Communication Business Process Profile is named "EFET Communication *[Business Process Name]* Profile" where the bracket expression is replaced with the name of this electronic business process.

Each electronic business process specification based on the EFET Communication Standard must contain exactly one such profile.

The EFET Communication Standard defines the EFET ebXML Messaging Standard Profile and the list of items to be considered in each EFET Communication Business Process Profile.

Specification

The EFET Communication Protocol Profile is structured alongside the two central architectural elements of the ebXML Message Service Specification, the ebXML Message Service Handler (see Appendix A.[1], 1.2.4) and the ebXML Message Structure (see Appendix A.[1], 2.1).

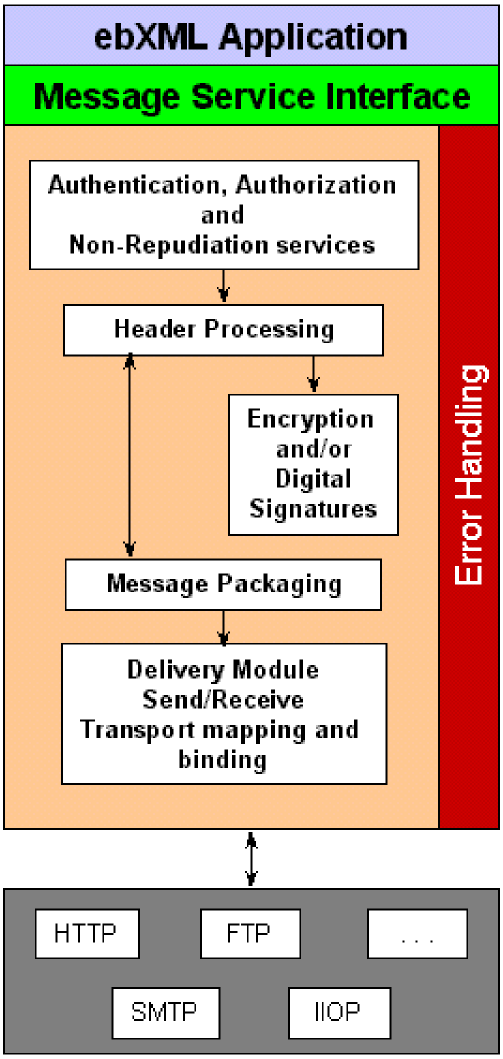
ebXML Application
Message Service Interface
Authentication, Authorization and Non-Repudiation services
Header Processing
Encryption and/or Digital Signatures
Message Packaging
Delivery Module Send/Receive Transport mapping and binding
Error Handling
HTTP
FTP
. . .
SMTP
IIOP

Figure 3 ebXML Message Service

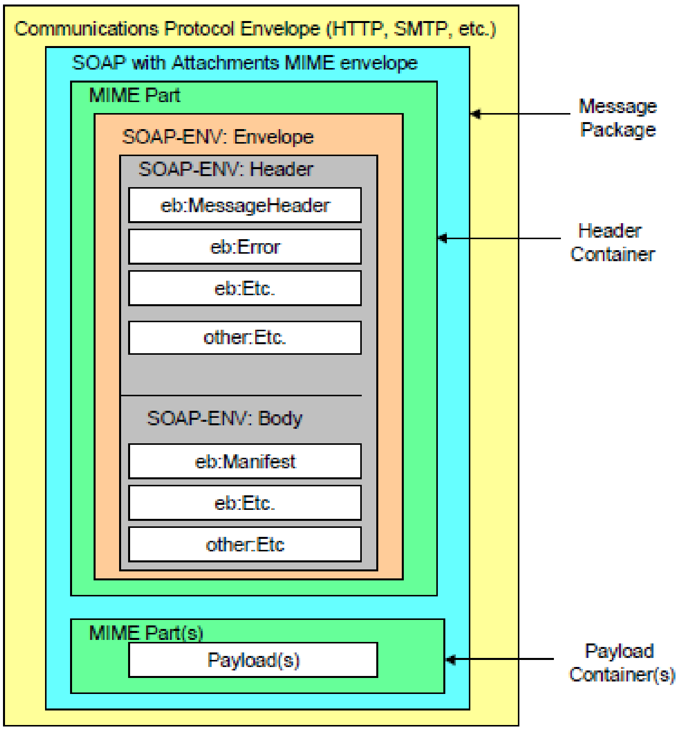

Figure 4 ebXML Message Structure

ebXML Services

MessageService

This section refers to Appendix A.[1], 1.2.4 and is part of the EFET ebXML Messaging Standard Profile.

Every business application conforming to the EFET Communication Standard and implementing an electronic business process specification based on this standard must utilise a compliant ebXML Message Service implementation configured (or customised) with all settings contained in the EFET Communication Protocol Profile associated to this electronic business process.

Ping Service

This section refers to Appendix A.[1], 8 and is part of the EFET ebXML Messaging Standard Profile.

The Message Service Handler Ping Service must be available.

Communication Partner Agreements

The ebXML Message Service Specification requires an agreement to exist between communication partners (see Appendix A.[1], 1.2.2, lines 340 – 360) and it uses a Collaboration Protocol Agreement (CPA, see Appendix A.[2], 8) to describe it.

The EFET ebXML Messaging Standard Profile mandates constraints on these agreements and defines them in terms of CPA elements, attributes, and their values.

This standard does not mandate the actual use of CPAs but every compliant EFET Communication Standard implementation must behave as specified by the ebXML Message Service Specification with regards to CPA settings.

This standard does not mandate a particular process to put such a communication partner agreement into existence between two partners.

Message Security

nonrepudiationOfOrigin Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.2 and is part of the EFET ebXML Messaging Standard Profile.

The attribute Boolean value is "true".

nonrepudiationOfReceipt Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.3 and is part of the EFET ebXML Messaging Standard Profile.

The attribute Boolean value is "true".

confidentiality Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.5 and is part of the EFET ebXML Messaging Standard Profile.

The attribute Boolean value is "false".

Note : This setting refers to the ebXML Message which is not encrypted in a persistent manner. However, this standard defines the message payload to be encrypted in a persistent manner, see section "Transport-Specific Items".

Protocol Element

This section refers to Appendix A.[2], 8.6 and 7.6.5.1 and is part of the EFET ebXML Messaging Standard Profile.

The element string value is "http://www.w3.org/2000/09/xmldsig#";

HashFunction Element

This section refers to Appendix A.[2], 8.6 and 7.6.5.2 and is part of the EFET ebXML Messaging Standard Profile.

The element string value is "http://www.w3.org/2000/09/xmldsig#sha1";

SIGNATUREALGORITHM ELEMENT

This section refers to Appendix A.[2], 8.6 and 7.6.5.3 and is part of the EFET ebXML Messaging Standard Profile.

The element string value is "http://www.w3.org/2000/09/xmldsig#rsa-sha1";

Message Exchange

DELIVERYSEMANTICS ATTRIBUTE

This section refers to Appendix A.[2], 8.6 and 7.6.4.1 and is part of the EFET ebXML Messaging Standard Profile.

The attribute enumeration value is "OnceAndOnlyOnce".

IDEMPOTENCY ATTRIBUTE

This section refers to Appendix A.[2], 8.6 and 7.6.4.2 and is part of the EFET ebXML Messaging Standard Profile.

The attribute Boolean value is "true".

MESSAGEORDERSEMANTICS ATTRIBUTE

This section refers to Appendix A.[2], 8.6 and 7.6.4.3 and is part of the EFET ebXML Messaging Standard Profile.

The attribute enumeration value is "NotGuaranteed".

Note : The reason for this setting is that a guaranteed message order conflicts with a mandatory synchronous reply for certain transport protocols (see Appendix A.[1], 9.1).

Transport Protocols

The EFET ebXML Messaging Standard Profile defines three transport protocol configurations:

- HTTPS
- HTTP
- SMTP

The following section lists all transport specific configuration items and subsequent sections define the configuration settings for each transport protocol, the minimal compliancy requirements and circumstances under which each might be used.

A business application implementing a business process specification compliant with this standard (i.e. a bespoke build or 3rd party software product) must be (at least minimally) compliant with the settings defined in the following sections. An installation (i.e. a site specific configuration of an implementation) whilst constrained by the scope of the implementation, will only be required to comply with the EFET ebXML Messaging Standard Profile for those transport protocols that it is configured to support. Any transport protocol that an installation supports in a compliant way can be used to communicate bilaterally with other organisations.

In case the EFET ebXML Messaging Standard Profile and the relevant EFET Communication Business Process Profile do not mandate settings (e.g. the number of retries, the retry interval, and the persist duration) the actual values can be agreed bilaterally between installations using compliant implementations of the standards that support such flexibility.

Note : The 'Default Values and Minimal Compliancy Requirements' defined in the following sections are not intended to restrict the transport-specific settings with respect to the business benefit they provide through support for flexibility and/or the selection of settings on a bilateral basis. Implementations which meet all the requirements of the settings given below shall be considered fully compliant; whereas implementations that meet the minimal compliancy requirements shall be considered compliant for those aspects of the standards that they implement in a compliant way. In this way variations in compliant implementations are both permitted and transparently declared.

Transport-Specific Items

syncReplyMode Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.1 and is part of the EFET ebXML Messaging Standard Profile.

secureTransport Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.4 and is part of the EFET ebXML Messaging Standard Profile.

authenticated Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.6 and is part of the EFET ebXML Messaging Standard Profile.

authorized Attribute

This section refers to Appendix A.[2], 8.6 and 7.5.11.7 and is part of the EFET ebXML Messaging Standard Profile.

SendingProtocol Element

This section refers to Appendix A.[2], 8.6 and 7.5.13.1 and is part of the EFET ebXML Messaging Standard Profile.

This element has a **version** attribute.

ReceivingProtocol Element

This section refers to Appendix A.[2], 8.6 and 7.5.13.2 and is part of the EFET ebXML Messaging Standard Profile.

This element has a **version** attribute.

Protocol Element

This section refers to Appendix A.[2], 8.6 and 7.5.16.1 and is part of the EFET ebXML Messaging Standard Profile.

This element has a **version** attribute.

Retries Element

This section refers to Appendix A.[2], 8.6 and 7.6.4.4 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **Retries** element
  - *integer value*

The EFET ebXML Messaging Standard Profile defines only a recommended default value.

RetryInterval Element

This section refers to Appendix A.[2], 8.6 and 7.6.4.4 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **RetryInterval** element
  - *integer value*

The EFET ebXML Messaging Standard Profile defines only a recommended default value.

PersistDuration Element

This section refers to Appendix A.[2], 8.6 and 7.6.4.5 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **PersistDuration** element
  - *integer value*

HTTPS

All implementations compliant with this standard must as a minimum compliancy requirement be capable of supporting this protocol although it is not a requirement of the standard that all compliant installations must use this protocol. Installations wishing to support this protocol MUST be fully compliant with the settings laid out in the table below.

Transfer encodings are not required but allowed.

Transport-specific Item	Element/Attribute Value	Default Values
syncReplyMode Attribute	“true”	
secureTransport Attribute	“true”	
authenticated Attribute	“false”	
authorized Attribute	“false”	
SendingProtocol Element/ ReceivingProtocol Element	“HTTP” (version “1.1”)	
Protocol Element	SSL (version “3.0”)	
Retries Element		3
RetryInterval Element		1 minute

HTTP

All implementations compliant with this standard may support this protocol. Installations wishing to support this protocol must be fully compliant with the settings laid out in the table below.

Transfer encodings are not required but allowed.

Transport-specific Item	Element/Attribute Value	Default Values
syncReplyMode Attribute	“true”	
secureTransport Attribute	“false”	
authenticated Attribute	“false”	
authorized Attribute	“false”	
SendingProtocol Element/ ReceivingProtocol Element	“HTTP” (version “1.1”)	
Protocol Element		
Retries Element		3
RetryInterval Element		1 minute

SMTP

All implementations compliant with this standard may support this protocol. Installations wishing to support this protocol must be fully compliant with the settings laid out in the table below.

The transfer encoding must be BASE64 to avoid any interoperability issues.

The use of S/MIME must be agreed bilaterally at the installation level. If it is implemented by an installation then that installation shall be required to be compliant.

Transport-specific Item	Element/Attribute Value	Default Values

Transport-specific Item	Element/Attribute Value	Default Values
syncReplyMode Attribute	“false”	
secureTransport Attribute	“false”	
authenticated Attribute	“false”	
authorized Attribute	“false”	
SendingProtocol Element/ ReceivingProtocol Element	“SMTP” (version “RFC821”)	
Protocol Element		
Retries Element		3
RetryInterval Element		10 minutes

SOAP

EBXML SOAP EXTENSION ELEMENTS SCHEMA

This section refers to Appendix A.[1] and is part of the EFET Communication Business Process Profile.

The ebXML SOAP Extension Elements Schema allows the use of foreign namespace elements or attributes (see **ANY** and **ANYATTRIBUTE** elements) as extensions to the elements and attributes defined in this schema.

While business process specifications are discouraged to make use of this feature, they must specify every process-specific extension element or attribute and define their exact semantics.

SOAP Envelope

XML PROLOG

This section refers to Appendix A.[1], 2.2 and is part of the EFET ebXML Messaging Standard Profile.

The SOAP Message’s XML Prolog must be present and it must contain a XML declaration.

The XML declaration must contain an encoding declaration.

The SOAP Message XML document encoding must be UTF-8.

SOAP Header

FROM AND TO ELEMENTS

This section refers to Appendix A.[1], 3.1.1 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **FROM** element and **TO** element
  - **PARTYID** element
    - *occurrence and ordering*
    - *string value*
    - **TYPE** attribute
      - *presence*
      - *string value*
  - **ROLE** element
    - *presence*
    - *string value*

CPAId Element

This section refers to Appendix A.[1], 3.1.2 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **CPAId** element
  - *string value*

ConversationId Element

This section refers to Appendix A.[1], 3.1.3 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **ConversationId** element
  - *string value*

Furthermore the relevant business process specification must define the exact usage of ConversationId, i.e. under which circumstances a communication party initiates a new conversation, how a new ConversationId is generated by this party, and which subsequent messages belong to this conversation and hence carry the same ConversationId.

Service Element

This section refers to Appendix A.[1], 3.1.4 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **Service** element
  - *string value*
  - **type** attribute
    - *presence*
    - *string value*

Furthermore the relevant business process specification must define the available services.

Action Element

This section refers to Appendix A.[1], 3.1.5 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **Action** element
  - *string value*

Furthermore the relevant business process specification must define the actions for each available service.

MessageId Element

This section refers to Appendix A.[1], 3.1.6.1 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **MessageId** element
  - *string value*

Furthermore the relevant business process specification must define how to generate a globally unique Id for each message.

RefToMessageId Element

This section refers to Appendix A.[1], 3.1.6.3 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **RefToMessageId** element
  - *presence (only in non-error case)*
  - *string value (i.e. which message to refer to)*

TimeToLive Element

This section refers to Appendix A.[1], 3.1.6.4 and 6.4.5 and is part of the EFET ebXML Messaging Standard Profile.

The **TimeToLive** element must be present since this is a requirement for reliable messaging (see section "Message Exchange").

DuplicateElimination Element

This section refers to Appendix A.[1], 3.1.7 and is part of the EFET ebXML Messaging Standard Profile.

The **DuplicateElimination** element must be present since this is a requirement for reliable messaging (see section "Message Exchange").

Signature Element

This section refers to Appendix A.[1], 4.1.1 and is part of the EFET ebXML Messaging Standard Profile.

A **Signature** element must be present since this is a requirement for secure messaging (see section "Message Security").

All mandatory and recommended signature generation steps in Appendix A.[1], 4.1.3 must be followed.

SyncReply Element

This section refers to Appendix A.[1], 4.3.1 and is part of the EFET ebXML Messaging Standard Profile.

The presence of this element depends on the transport protocol used, see section "Transport-Specific Items".

AckRequested Element

This section refers to Appendix A.[1], 6.3.1 and is part of the EFET ebXML Messaging Standard Profile.

An **AckRequested** element must be present whenever the ebXML Message Service Specification allows for it (see in particular Appendix A.[1], 6.3.1.4).

The relevant business process specification will (at least) implicitly define if this element is targeted at the "NextMSH" or "ToPartyMSH" or if two **AckRequested** elements are present, one for each target (see Appendix A.[1], 6.3.1., 6.3.1.1). This also determines the **actor** attribute value.

The **signed** attribute Boolean value is "true" (see section "Message Security").

Acknowledgment Element

This section refers to Appendix A.[1], 6.3.2 and is part of the EFET ebXML Messaging Standard Profile.

An Acknowledgment element must not be present in a message containing a payload.

An Acknowledgement message is defined as a message compliant with this specification except that

- it has an Acknowledgment element,
- it has an empty SOAP Body element (i.e. it does not contain a payload, see Appendix A.[1], 2.1.4),
- it complies with Appendix A.[1], 6.3.1.4,
- it complies with Appendix A.[1], 6.3.2.7.

MessageOrder Element

This section refers to Appendix A.[1], 9.1 and is part of the EFET ebXML Messaging Standard Profile.

The **MessageOrder** element must not be present, see section "Message Exchange".

SOAP Body

Manifest Element

This section refers to Appendix A.[1], 3.2 and is part of the EFET ebXML Messaging Standard Profile.

The **Manifest** element must be present.

Payload data must not be present in the SOAP Body.

Reference Element

This section refers to Appendix A.[1], 3.2.1 and is part of an EFET Communication Business Process Profile. The following must be specified.

- **Reference** element

- **XLINK:HREF** attribute
    - *URI value*
- **XLINK:ROLE** attribute
    - *presence*
    - *URI value*
- **SCHEMA** element
    - *presence and occurrence*
    - **LOCATION** attribute
        - *URI value*
    - **VERSION** attribute
        - *presence*
        - *string value*

Payload

This section refers to Appendix A.[1], 2.1.4 and is part of an EFET Communication Business Process Profile.

The EFET ebXML Messaging Standard Profile defines only a recommended default payload scheme.

DEFAULT PAYLOAD SCHEME

The payload consists of exactly two payload containers, the first containing an encrypted business document and the second containing the business document signature. The business document is signed by the **business document creator** and encrypted by the message originator for the message recipient. Every EFET Communication Business Process Profile which makes use of this default payload scheme must specify how to determine the business document creator. This definition must in particular enable any business document consumer to verify the business document signature in order to ensure document integrity and non-repudiation.

The business document format is not restricted. In particular it is not mandatory to use XML documents as business documents.

The security algorithms and encodings used are given in the following table.

Security Item	Default Payload Scheme Setting
Document Signing	Business document signature algorithm is “SHA1 with RSA” (see Appendix A.[4]). Content encoding as “CMS signed data” (see Appendix A.[5]).
Document Compression	Compression methodology is gzip compliant (see Appendix A.[6]).
Document Encryption	Document encryption algorithm is “Three Key Triple-DES in Cipher-Block-Chaining mode” (see Appendix A.[3]). Asymmetric Key encryption algorithm is “RSA” (see Appendix A.[4]). Content encoding as "CMS enveloped data" (see Appendix A.[5]).

The algorithm for constructing the two payload containers for an outgoing message is as follows.

1. Sign the business document with the business document creator’s private key.
2. Create a CMS signed data entity including the business document signature, excluding the business document, and optionally including relevant certificates. This is the business document signature.
   Adding certificates is not mandatory since

   a. these are already included within the XML Signature in the SOAP envelope (as required by the XML Signature standard),
   b. attached certificate chains have no value since they may be faked by the sending party. Instead it is expected that the receiving party obtains the required certificates via a trusted path from the issuing certificate authority.

3. Compress the business document.
4. Generate a random content encryption key.
5. Encrypt the compressed business document with the content encryption key.
6. Encrypt the content encryption key with the message recipient's public key.
7. Create a CMS enveloped data entity including the encrypted business document and the encrypted content encryption key and include it as the first payload container in the ebXML message. Include the CMS signed data entity as the second payload container in the ebXML message. The **content-transfer-encoding** should not be restricted as all MIME capable applications must be able to handle any **content-transfer-encoding**. The content-type reflects how the data is to be processed. For both payload containers the correct content-type is "application/pkcs7-mime".

The algorithm for retrieving the original business document from an ebXML message is as follows.

1. Decrypt the encrypted content encryption key using the message recipient's private key.
2. Decrypt the compressed business document using the content encryption key.
3. Decompress the compressed business document.
4. Validate the business document signature using the business document creator's public key.
5. Where the "**REFERENCE** Element" specifies a schema for the business document payload container, then the business document must be checked against this schema and recognised as valid.

3.6 Example for an HTTP EFET ebXML Message

This section is not normative.

```
POST /pontonxp/SoapListener HTTP/1.1
Content-Length: 9538
SOAPAction: "ebXML"
Message-Id: MID-1267031382598@ponton.xp
Content-Type: multipart/related; boundary="----=_Part_2_16772381.1267035685908"; start="<EbXml-
   Envelope-1267035685877>"; type="text/xml"
User-Agent: PontonXP/3.1.14
Host: xptest.ponton-consulting.de

------=_Part_2_16772381.1267035685908
Content-Type: text/xml; charset=utf-8
Content-Transfer-Encoding: binary
Content-Id: <EbXml-Envelope-1267035685877>


<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap env="http://schemas.xmlsoap.org/soap/envelope/">
   <soap-env:Header>
      <eb:MessageHeader xmlns:eb="http://www.oasis-open.org/committees/ebxml-
            msg/schema/msg-header-2_0.xsd" soap-env:mustUnderstand="1" eb:version="2.0">
         <eb:From>
            <eb:PartyId eb:type="EIC">11XRWETRADING--0</eb:PartyId>
         </eb:From>
         <eb:To>
            <eb:PartyId eb:type="EIC">11XELECTRABEL--Z</eb:PartyId>
```

```
        </eb:To>
        <eb:CPAId>http://www.efet.org/cpa/dummy.xml</eb:CPAId>
        <eb:ConversationId> MID-1079985404061@RWE.com</eb:ConversationId>
        <eb:Service eb:type="EFET">ECM-3.3</eb:Service>
        <eb:Action>TradeConfirmation</eb:Action>
        <eb:MessageData>
            <eb:MessageId>MID-1079985404061@RWE.com</eb:MessageId>
            <eb:Timestamp>2004-03-22T19:56:44</eb:Timestamp>
            <eb:TimeToLive>2004-03-22T20:00:24</eb:TimeToLive>
        </eb:MessageData>
        <eb:DuplicateElimination />
    </eb:MessageHeader>
    <eb:SyncReply xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
            header-2_0.xsd" soap-env:mustUnderstand="1" eb:version="2.0" soap-
            env:actor="http://schemas.xmlsoap.org/soap/actor/next" />
    <eb:AckRequested xmlns:eb="http://www.oasis-open.org/committees/ebxml-
            msg/schema/msg-header-2_0.xsd" soap-env:mustUnderstand="1" eb:version="2.0"
            eb:signed="true" />
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
    <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <ds:XPath>
    not (ancestor-or-self::node() [@soap-env:actor="urn:oasis:names:tc:ebxml-
    msg:actor:nextMSH"] | ancestor-or-self::node() [@soap-
    env:actor="http://schemas.xmlsoap.org/soap/actor/next"])
    </ds:XPath>
    </ds:Transform>
    <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>R0gTzbPGWmTeu5YvUkoMEgbeRFI=</ds:DigestValue>
    </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
    bXHsA25+FNPfFvmEgS2iD10V+sQgNvqLVetVJ4XUeuOdntPG8Y57h/BAZ9exgU2vyuR8ZK2rYdyW
    eFvoJFj0zscYifdoBNtR0Y9jeHazRpS/NOwBbrTJQtdrzhxSJfUBCd6gmkbWS33gYhoGBbMX/XV4
    xGOyQAcgWuHmqZsTTAE=
    </ds:SignatureValue>
    <ds:KeyInfo>
    <ds:X509Data>
    <ds:X509Certificate>
    MIIDUDCCAjigAwIBAgIGAScAUysFMA0GCSqGSIb3DQEBBQUAMIGdMQswCQYDVQQGEwJERTEQMA4G
    A1UEBxMHSGFtYnVyZzEfMB0GA1UEChMWUG9udG9uIENvbnN1bHRpbmcgR21iSDEYMBYGA1UECxMP
    TmV0d29yayBTZXJ2aWNlMRcwFQYDVQQDEw5Qb250b24gUm9vdCBDQTEoMCYGCSqGSIb3DQEJARYZ
    aW5mb0Bwb250b24tY29uc3VsdGluZy5kZTAeFw0xMDAyMjMxNDEzMjJaFw0xMzAyMjQxNDEzMjJa
    MIG3MQswCQYDVQQGDAJERTEQMA4GA1UECAwHSGFtYnVyZzEQMA4GA1UEBwwHSGFtYnVyZzEfMB0G
    A1UECgwWUG9udG9uIENvbnN1bHRpbmcgR21iSDEZMBcGA1UECwwQRUZFVG5ldCBIZWxwZGVzazEY
    MBYGA1UEAwwPdHJhZGVyX3NlcnZlcjAxMS4wLAYJKoZIhvcNAQkBFh9yZXR0c2NobGFnQHBvbnRv
    bi1jb25zdWx0aW5nLmRlMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCj75SKL92KH6U7tyNo
    zCPKx4WcrZLv87bIn1FdlalBXxrutSiZcSD+1dDd/CATyHRrCXLuTorfLGDaayTZaxsrzvRsa752
    k43J7TOpB+ZBwg5uzpeEUjsBVWyfAs6Ir2MwohOiCH0AgzElch9fCjSCv6fzKJJR0BYtQsh4U91J
    cwIDAQABMA0GCSqGSIb3DQEBBQUAA4IBAQCGsS2dq/mtsD2qwd4Hf1SYo1kni7Fut/aGKJXHd/EN
    +kQZCdsGRueqvvs9Yjfwz+lF3VSDxWn/zFIUWc5DSQoXFLjDrvqd3te0A/CiowLDYwU17INWd5vt
    1nqpEixmjmg6PQ7U3R4GaA1wkbLx/FnZhanV13d66Z6x94vlm9L96N/UXwnJH1t1Cl3HGlGWxPax
    +RrOZysDiu4KS5/F3MAAPkBJefR/aTLT09z/Sjq7/+Ff2hEFY1SGyNiYzH7C5Afsi4lySoWBnscZ
    Fxk74GvZsTduZA3jHhWuswRkrB/uuz1Sz8smoTtQfjVmuJ+Kw/Y+X4G17GqUQaXNsDVcvleU
    </ds:X509Certificate>
    </ds:X509Data>
    <ds:X509Certificate>
    MIIEKTCCAxGgAwIBAgIGAQBLkwElMA0GCSqGSIb3DQEBBAUAMIGdMQswCQYDVQQGEwJERTEQMA4G
    A1UEBxMHSGFtYnVyZzEfMB0GA1UEChMWUG9udG9uIENvbnN1bHRpbmcgR21iSDEYMBYGA1UECxMP
    TmV0d29yayBTZXJ2aWNlMRcwFQYDVQQDEw5Qb250b24gUm9vdCBDQTEoMCYGCSqGSIb3DQEJARYZ
    aW5mb0Bwb250b24tY29uc3VsdGluZy5kZTAgFw0wNDExMTgxMjA1NTNaGA8zMDA0MTExODEyMDU1
    M1owgZ0xCzAJBgNVBAYTAkRFMRAwDgYDVQQHEwdIYW1idXJnMR8wHQYDVQQKExZQb250b24gQ29u
    c3VsdGluZyBHbWJIMRgwFgYDVQQLEw9OZXR3b3JrIFNlcnZpY2UxFzAVBgNVBAMTDlBvbnRvbiBS
    b290IENBMSgwJgYJKoZIhvcNAQkBFhlpbmZvQHBvbnRvbi1jb25zdWx0aW5nLmRlMIIBIjANBgkq
    hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxxcbfTHhaHBYigpkATJxRcx3RDy9itZaF+C8PiQfh7Ij
    JnwBK2R3xwwcOqRgkCzM8vSkXAg8QzRsZzGAI1xS/NaWzJz0hruC/txXqWkb8cU7p+TmMf2mEqwm
    e8ocdZ/9a4hvmP8qVVGlD/fBG5UPXM1Kzgnbl4AfM7YjnEy9VpQmg1TvVYWwE3A8Xf1goVlaeOmD
    FJpA8UR1tr7GLnm9DMg+r7tkywS2HvfqmpBdHKbeFKR+1iay109s1UC45cdps573aquvziVt69RG
    hW0FcJRtFf6fT1oRvDqO59FZATP1MIQbSX8lahCidpa80SL039Xxuy9C+pKGTdXKFMTw/wIDAQAB
    o2swaTARBglghkgBhvhCAQEEBAMCAAcwDwYDVR0TBAgwBgEB/wIBADAkBgNVHREEHTAbgRlpbmZv
    QHBvbnRvbi1jb25zdWx0aW5nLmRlMB0GA1UdDgQWBBTENu6VUG1Q9ITP0vzdeHA/w0xxRzANBgkq
    hkiG9w0BAQQFAAOCAQEAg4C1g6EE8ZCqRzHVDixfe3wOr9pfI0reLH19BfZXf6UNsocASbBd7LVJ
    24FcJK4hN+Bxo65npYVO+RgL4F7JE4zK3cOkb/j8sNQ/XpwXzdr36xdIO3y0sEX/N7B9DoTOqw+/
    I4KFiZIrVSi1ZCvXV4JO9jItJFzDmSpx7CSCaJvFO2ZGSr73PpCSpLLSUNkA/urAKo2oSHCU0Bee
    DrP2MoV4hIzp47yRK7v9TJQOfxEuKN3nHQ0Tj0eP+19Hki7jYgyhx+NJD/FHySEDtaWjtZzVBSVD
    5aMCNOFGJsakU2aN0cFjx4VOixeM0YVQjiwSS8Z8+/bjzNmRo8F6gIJXJw==
    </ds:X509Certificate>
    </ds:X509Data>
    </ds:KeyInfo>
    </ds:Signature>
  </soap-env:Header>
  <soap-env:Body>
    <eb:Manifest xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
        header-2_0.xsd" eb:version="2.0">
      <eb:Reference xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="cid: payload-
          1267035685877"
          xlink:role="http://www.efet.org/ebXMLAttachment/Roles/BusinessDocument">
        <eb:Schema eb:location="http://www.efet.org/Schemas/eCM/V3R3/EFET-CNF-V3R3.xsd"
            eb:version="3.31" />
        <eb:Description xml:lang="en">EFET Payload</eb:Description>
      </eb:Reference>
      <eb:Reference xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="cid: payload.pkcs7.sig"
          xlink:role="http://www.efet.org/ebXMLAttachment/Roles/BusinessDocumentSignature">
        <eb:Description xml:lang="en">Signature</eb:Description>
```

```
            </eb:Reference>
        </eb:Manifest>
    </soap-env:Body>
</soap-env:Envelope>
------=_Part_2_16772381.1267035685908
Content-Type: application/pkcs7-mime
Content-Transfer-Encoding: binary
Content-Id: payload-1267035685877
Content-Disposition: attachment;filename="payload.encrypted.data"

XXXXXXXXX binary data here

------=_Part_2_16772381.1267035685908
Content-Type: application/pkcs7-mime
Content-Transfer-Encoding: binary
Content-Id: payload.pkcs7.sig
Content-Disposition: attachment;filename="payload.pkcs7.sig"

XXXXXXXXX binary data here
```

3.7 EFET Communication Business Process Profile Template

This section is not normative.

Business process specifications might want to include the following table to define an EFET Communication Business Process Profile.

Configuration Item	Element/Attribute Value(s) Further Definitions	Remarks
Retries Element		
RetryInterval Element		
PersistDuration Element		
ebXML SOAP Extension Elements or Attributes		
From and **To** Elements		
CPAId Element		
ConversationId Element		
Service Element		
Action Element		
MessageId Element		
RefToMessageId Element		
Reference Element		
Payload Scheme		

3.8 Out-Of-Scope Topics

This section is not normative.

There are a number of topics in conjunction with the EFET Communication Standard which are deemed not in scope of this standard and not in scope of any single business process specification based on this standard.

It is assumed that communication partners participating in a common collection of business processes will form interest or user groups and that such groups have the right scope for agreeing standards for the following topics.

Agreement Management

A mechanism for negotiating, creating, modifying, exchanging, and storing bilateral "Communication Partner Agreements" needs to be agreed and implemented.

As a default solution the EFETnet Central Registry Service for Partner Agreements may be used.

Public Key Infrastructure and Certificate Management

The security features specified in this standard depend on the availability of a Public Key Infrastructure (PKI) and in particular a Certificate Authority (CA) issuing communication partner certificates. It is recommended that certificates of any available CAs should be accepted and used to verify signatures and to access public keys of recipients for document encryption.

As a default solution, certificates may be issued and maintained by the EFETnet helpdesk.

Furthermore certificates must be distributed between communication partners, possibly as part of communication partner agreements.

The effectiveness of cryptographic security measures depends (among other things) on key lengths. It is recommended to agree minimal or required key lengths for the different types of keys in use.

Appendix A. Normative References

[1] ebXML Message Service Specification Version 2.0,
OASIS ebXML Messaging Services Technical Committee, 1 April 2002
http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf

[2] Collaboration-Protocol Profile and Agreement Specification Version 1.0
ebXML Trading-Partners Team, 10 May 2001
http://www.ebxml.org/specs/ebCCP.pdf

[3] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
National Institute for Standards and Technology, 19 May 2008 (Revised)
http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf

[4] PKCS #1: RSA Cryptography Specifications Version 2.0
The Internet Engineering Task Force RFC 2437, October 1998
http://www.ietf.org/rfc/rfc2437.txt

[5] Cryptographic Message Syntax (CMS)
The Internet Engineering Task Force RFC 3852, July 2004
http://www.ietf.org/rfc/rfc3852.txt

[6] gzip (GNU zip) home site
http://www.gzip.org/