PARTICIPANT BUILD PACK 3

FRC B2B SYSTEM ARCHITECTURE

PREPARED BY: MARKET DEVELOPMENT

DOCUMENT REF: 305131

VERSION: 3.2

DATE : 17 MAY 2015

FINAL

Version History

VERSION	DATE	AUTHOR(S)	CHANGES AND COMMENTS
0.1	7/2/2002	Neil Belford	• Distribution Draft
1.0	8/2/2002	Neil Belford	• First Release
1.1	22/2/2002	Neil Belford	• Participant Build Pack 3 Release
1.2	4/4/2002	Neil Belford	• Changes include additional new information about timing requirements in the message and transaction cycles.
1.3	30/4/2002	Neil Belford	• Change to ebXML Message Acknowledgement. Other minor changes to text.
1.4	23/02/2004	Darren Field	• Change Request 59. Updated referenced document definitions to generic version references, to enable document updates without requiring a change to this document. Removed references to SMTP
			• Minor changes to introduction and approval section to accommodate post FRC operation
1.5	8/11/2004	Danny McGowan	• Minor change to the Related Documents section 1.3 to reference the aseXML Standards and Guidelines version in the GIP
			• Change Request 55. New paragraph included in section 3.1.5. This change introduces a consistent way of handling duplicate transaction. Changes to the aseXML guidelines are also necessary.
1.6	18/07/2005	Danny McGowan	• Minor changes to approval section to reflect organisational changes
1.7	01/06/2009	Danny McGowan	• Minor changes to approval section to reflect organisational changes
3.0	01/07/2010	Stefanie Monaco	• Ensure document complies with AEMO standard. • Replace references to MSOR with relevant NGR & RMP references. • Update terminology to correspond with current usage and definitions. • Deleted acronyms and placed into consolidated PBP2 Glossary. • Deleted Appendix A as information is no longer relevant. • Update images • Add References to Predecessors
3.1	1/1/2014	Danny McGowan	• IN004/12 – Redundant Provision and minor GIP and Spec Pack changes
3.2	17/05/2015	Allan Ng	• IN023/14 - FRC HUB Upgrade Project.

			• Replaced reference to Hansen to MarketNet • Remove reference to redundant “Provision of FRC Hub Tele-Housing and Services” document

Executive Summary

This document is the FRC B2B System Architecture document for delivery of the IT Systems to support the operation of the Victorian Gas Retail Market.

This document forms part of the Gas Interface Protocol and has been prepared in accordance with Part 15B of the National Gas Rules [and the Approved Process under rule 135EC of the National Gas Rules].

References to Predecessors

To reflect the governance changes implemented on 1 July 2009, this document has been amended to remove references to the Victorian Energy Networks Corporation (VENCorp) and replace such references with Australian Energy Market Operator (AEMO). Where any content inadvertently refers to VENCorp it should be read as referring to AEMO.

In relation to the aseXML schema, it should be noted that participant ID “VENCORP” remains as the participant ID for AEMO as the gas market operator in Victoria and Queensland.

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the FRC B2B System. It is intended to convey the significant architectural decisions that define the FRC B2B System. Using this document, participants will be able to make technology choices and design gateway interfaces to the FRC Hub.

1.2 Audience

The document has been written for business and IT personnel within industry participants in the gas industry, as well as AEMO business and IT personnel. It is expected that the audience will have a familiarity with the overall business endeavour of the Gas Retail Market in Victoria and with the artefacts listed in the Related Documents section of this document.

1.3 Related Documents

REF	ARTEFACT NAME	VERSION	RESPONSIBLE PARTY OR AUTHORS
1	Full Retail Gas Contestability B2B Infrastructure Report	1.0	Housley Communications
2	ebXML Message Service Specification	1.0	UN/CEFACT and OASIS
3	Participant Build Pack 1	Current version as published in the GIP	AEMO
4	Participant Build Pack 2	Current version as published in the GIP	AEMO
5	aseXML Standards and Guidelines	Current version as published in the GIP	ASWG
6	FRC B2B System Specifications	Current version as published in the GIP	AEMO

1.4 Outline

Chapter 2. *Architectural Overview* - Gives the overview and a definition of the FRC B2B System architecture.

Chapter 3. *Application Layer* – This section describes functional and operational aspects of the aseXML Transaction Application. Topics include the obligations of the application, schema and schema validation, and interoperability. The section also describes the participant Communications Infrastructure, Public Key Infrastructure and the Message Service Interface, which will mediate between the Message Service Handler and these other applications / infrastructure-elements.

Chapter 4. *Message Layer* – The Message Service Handler (MSH) is the centrepiece of the messaging system in the FRC B2B System. The MSH is the implementation of the ebXML Message Service Specification. Packaging, routing, and delivery are dealt with in detail. The handler services, being Message Status Request, and MSH Ping are also described.

Chapter 5. *Transport Layer* – The Message Transport Interface section deals with the interface between the Message Service Handler and transport protocols to be supported. Network infrastructure is dealt with; topics here are the system topology, gateways, the hub, and expected network traffic.

Chapter 6. *Security* – Security issues are descriptively treated, an emphasis here is to give participants an understandable treatment of the issues involved as well as the expected implementation detail, and participant requirements. Details, which would themselves compromise the security approach, have been excluded. The three sections are Key Management, Encryption, and Digital Signature.

2. Architectural Overview

2.1 Design Outline

The recommendations of the Housley report for the Victorian Gas Industry set the majority of the parameters that frame the design of the FRC B2B System (FBS). This system is comprised of four types of components

- The FRC Hub
- The FBS Certificate Authority
- The FBS Test and Certification Gateway
- Multiple Participant FBS Gateways

The system relationships are shown in Figure 1.

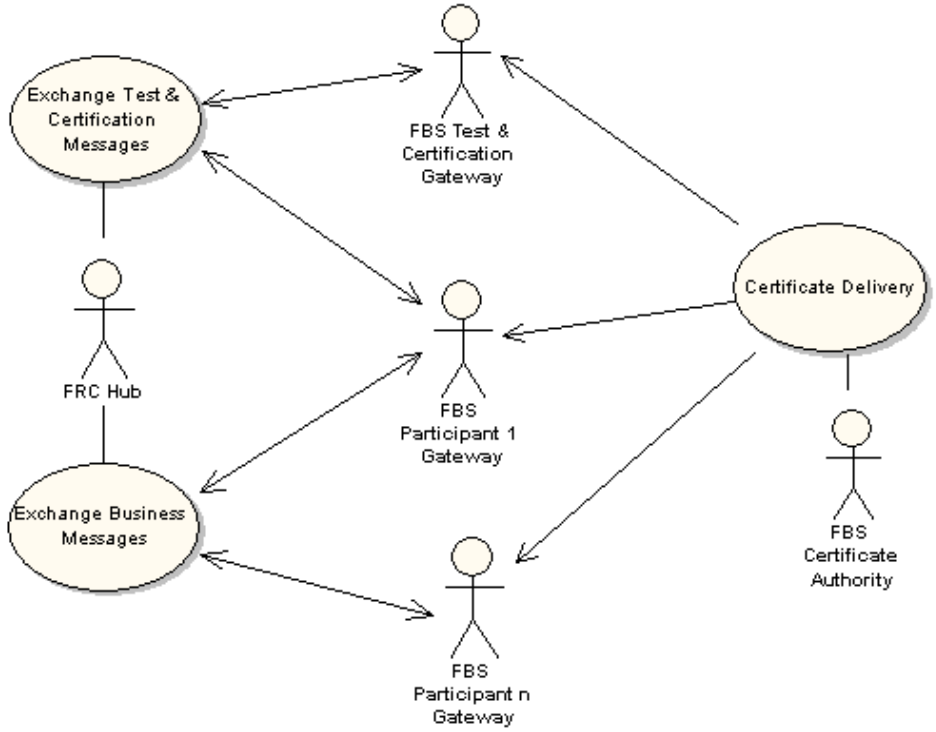

Figure 2.1. FRC B2B System

AEMO is responsible for designing and developing the FRC B2B System, which comprises of the design and delivery of the FRC hub, the FBS Certificate Authority, the FBS Test and Certification Gateway and the provision of FBS Gateway protocols, in the form of the build packs, to the participants.

Structural details that need to be noted are:

2.1.1 Messaging

Message handling is to be implemented using ebXML envelopes. FBS Gateways will deploy a complying ebXML Message Service Handler (MSH)

2.1.2 Topology

The FRC B2B System specifies a hub-and-spoke architecture. The participants are responsible for developing and maintaining their own FBS Gateways.

2.1.3 Transport Protocols

The transport protocol to be supported is HTTP.

2.1.4 Encryption

Transport layer encryption using X509v3 certificates with SSL will provide data security. The FBS Certificate Authority will issue X509v3 certificates for encryption/decryption.

2.1.5 Authentication and Non-Repudiation

Authentication and Non-Repudiation of Receipt will be established by complying with ebXML signed reliable delivery with signed acknowledgement. The signature will apply to the ebXML payload, using X509v3 certificates. The FBS Certificate Authority will issue certificates for signing and verifying.

2.1.6 Networks

The FRC B2B System will be configured to accept messages from the Internet, and from MarketNet. Participants may only use these two networks.

2.2 Component Layers

The FRC B2B System, as it applies to FBS Gateways, is described in terms of implementation in component layers. These layers are logical components; implementation details are the responsibility of the participants.

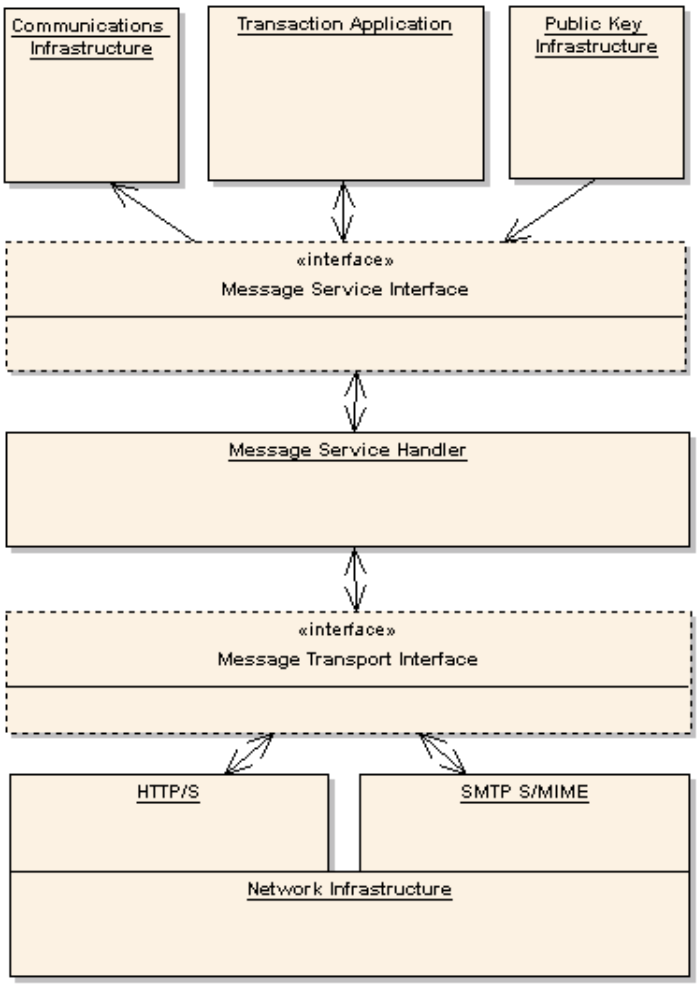

Figure 2.2. Logical Component View

2.3 Communication methodology

A most significant feature of the architecture is the functional separation of transactions and messages.

- Transactions are conducted on the basis of a set of industry rules encapsulated in the aseXML schema and expressed in aseXML documents. They will be managed by an aseXML transaction application.
- Messages will be managed by a Message Service Handler and will either

Contain aseXML documents (either transactions or transaction acknowledgements) as their payload, or

Be message acknowledgements of a message with such a payload.

Messages will conform to the ebXML standard, and will use the features provided within ebXML to provide security and non-repudiation.

2.3.1 Signals and Responses

2.3.1.1 Message Acknowledgements

In the FBS an ebXML message receiver will send a signed ebXML Message Acknowledgement as a signal to indicate that all the following conditions have been met:

1. A signed ebXML message with a payload has been received from a known source.
2. That the payload signature is valid.
3. That the payload is a well formed and schema valid aseXML document.
4. Message Acknowledgments must be received within time constraints. These constraints are defined in Section 2.3 of the *FBS System Specifications* document.
5. The time by which the acknowledgement should be received is referred to as **t1**.

2.3.1.2 Transaction Acknowledgements

In the FBS an aseXML application will send a Transaction Acknowledgement as a signal to indicate that the following about the aseXML document:

1. The transaction was part of a valid XML document.
2. The transaction has passed business rule validity checks and has been accepted for business processing.
3. The receiver has taken full responsibility for the transaction even though further work to provide a Transaction Response may be ongoing.
4. The time by which High Priority Services should provide a Transaction Acknowledgement is governed by strict constraints. These constraints are discussed here and are defined in Section 2.3 of the *FBS System Specifications* document.
5. The time by which the acknowledgement should be received is referred to as **t2**.

2.3.1.3 Transaction Responses

The conditions that govern the generation of a Transaction Response are defined in the Retail Market Procedures.

- The time by which the Transaction Response should be received is referred to as **t3**.

2.3.2 Message Exchange Scenarios

These scenarios show an abstraction of the ebXML reliable messaging process. This process is discussed in detail in section 4.5 of this document, and the discussion there describes the role of the hub in this process. The scenarios here deal at a high level with interchanges between participants.

The figure below depicts a successful message exchange scenario. A message may carry a transaction or a transaction acknowledgment.

Figure 2.3.2.1. Successful Message Exchange Sequence Diagram

Message delivery may fail in several ways. One scenario occurs for example, if the message has a payload that does not satisfy the signature that has been applied to it. On this ground the recipient of the message rejects it and issues a message in error message.

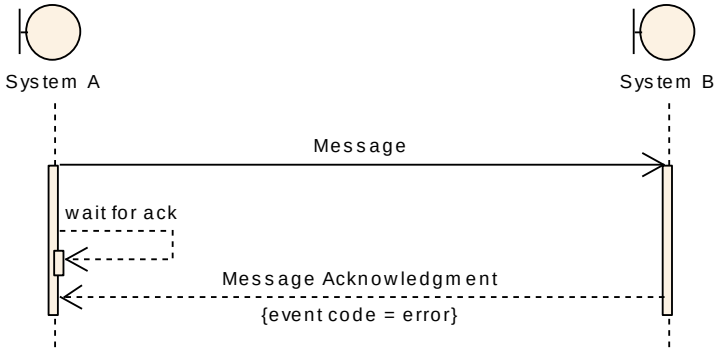

Figure 2.3.2.2. Failed Message Exchange Sequence Diagram

Another delivery failure scenario may be caused by the fact that a message, or its acknowledgement has been lost. A message acknowledgment hasn't arrived within predetermined time interval. The message originator (System A) will timeout and will either retry sending the message again or inform the System A about a communications failure. The detail as to how the FBS deals with these circumstances is described in section 4.5.

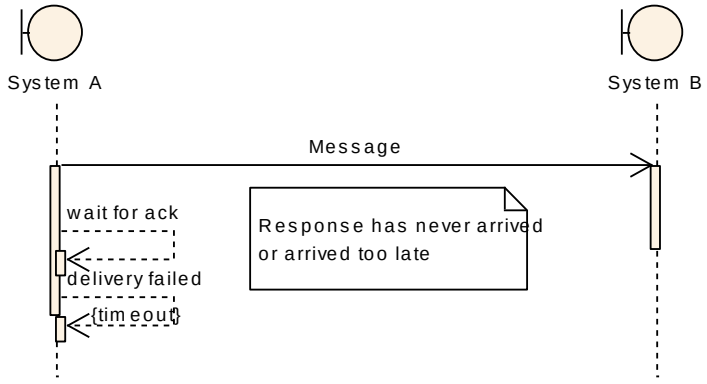

Figure 2.3.2.3. Lost Message Acknowledgment Sequence Diagram

2.3.3 Transaction Exchange Scenario

The transaction exchange process is an application level process. Transactions are described in an aseXML document; this document is carried as the payload of an ebXML message.

The figure below shows an example of a successful transaction exchange scenario. A transaction, for example a Customer Transfer Request, is sent to AEMO. AEMO acknowledges receipt of the transaction. Following some internal processing, AEMO issues a transaction to that Participant and this transaction is also duly acknowledged.

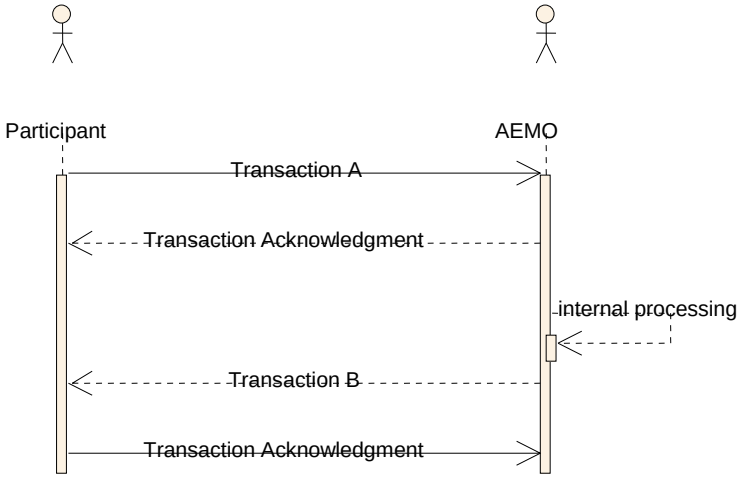

Figure 2.3.3.1 Transaction Exchange Sequence Diagram

2.4 Timing Considerations for Transaction Priorities

The FBS has a variety of high-level performance requirements with respect to time. The first decomposition of these requires inspection of the Transaction Cycle latencies, and each step in the Transaction Cycle decomposes further into a complete Message Cycle. This chapter works from the bottom up, dealing with the Message Cycle then the Transaction Cycle.

Transaction priorities and their related time constraints are an aseXML application level consideration. These timing considerations are then superimposed on the ebXML reliable messaging system. The reliable messaging system has own set of timing parameters; however it is important to note that these are independent of aseXML Transaction performance requirements.

2.4.1 Message Cycle Latencies

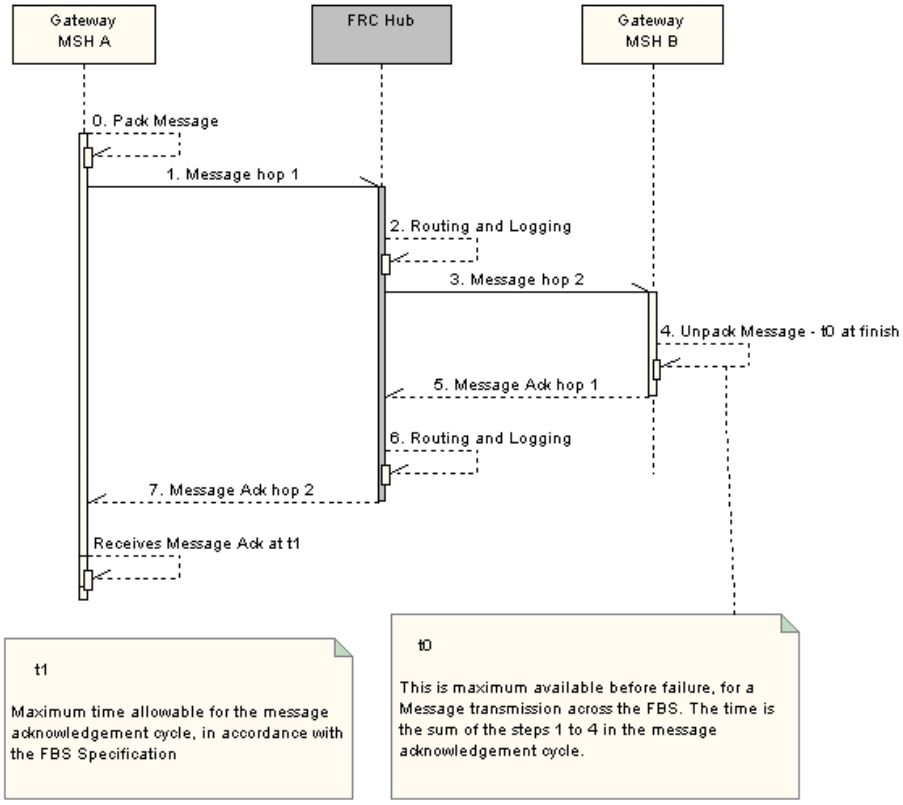

Figure 2.4.1.1 Timing steps in the Message cycle

Expected time latencies that apply to steps 1 through 7 in figure 2.4.1.1 are detailed in the following table

		TRANSMISSION LATENCIES				NOMINAL FAILURE POINTS	
	Step	99% < t	0.9% < t	0.1% > t		medium and low priority	high priority
	0						
	1	10	180	360		360	10
	2	2	30	40		40	10
	3	10	180	360		360	10
	4	6	180	360		**360**	**30**
	5	5	10	20		20	10
	6	2	10	20		20	10
	7	5	10	20		20	10

		TRANSMISSION LATENCIES				NOMINAL FAILURE POINTS	
Single cycle		**40**	**600**	**1180**		1180	90
Attempts						3	1
Cycle total						**3540**	**90**
Time t0						**3480**	**60**
Time t1						**3600**	**120**

- All times in this table are in seconds.
- The step column refers to the steps in figure 2.4.1.1
- The Transmission latencies columns refer to a % of transactions against a time interval 't'. In the cases of the 99% and 0.9% columns, the time refers to the maximum time 't' by which the respective steps are expected to be complete, while in the case of the 0.1% column the expectation is that the time taken will be greater than time 't'.
- The nominal failure points are broken into two categories – those for medium and low priority messages, and those for high priority messages. It is expected that high priority messages will be around 1 KB in size, whereas low and medium priority messages may be up to three orders of magnitude larger in size.
- The Single Cycle row refers to the maximum time expectation for a single attempt within the messaging cycle. The ebXML reliable messaging protocol provides for repeat attempts.
- The Attempts row refers to the maximum number of times a message can be sent. The number of retries is the number of attempts minus 1.
- The Cycle Total is the total duration allowing for all attempts.
- At the end of step 4 in the message cycle, at most t0 has elapsed before the message contents have been handed over to the application
- Time t1 is the maximum time allowable for the message and acknowledgement cycle to be complete.

2.4.2 Transaction and Response Cycle Latencies

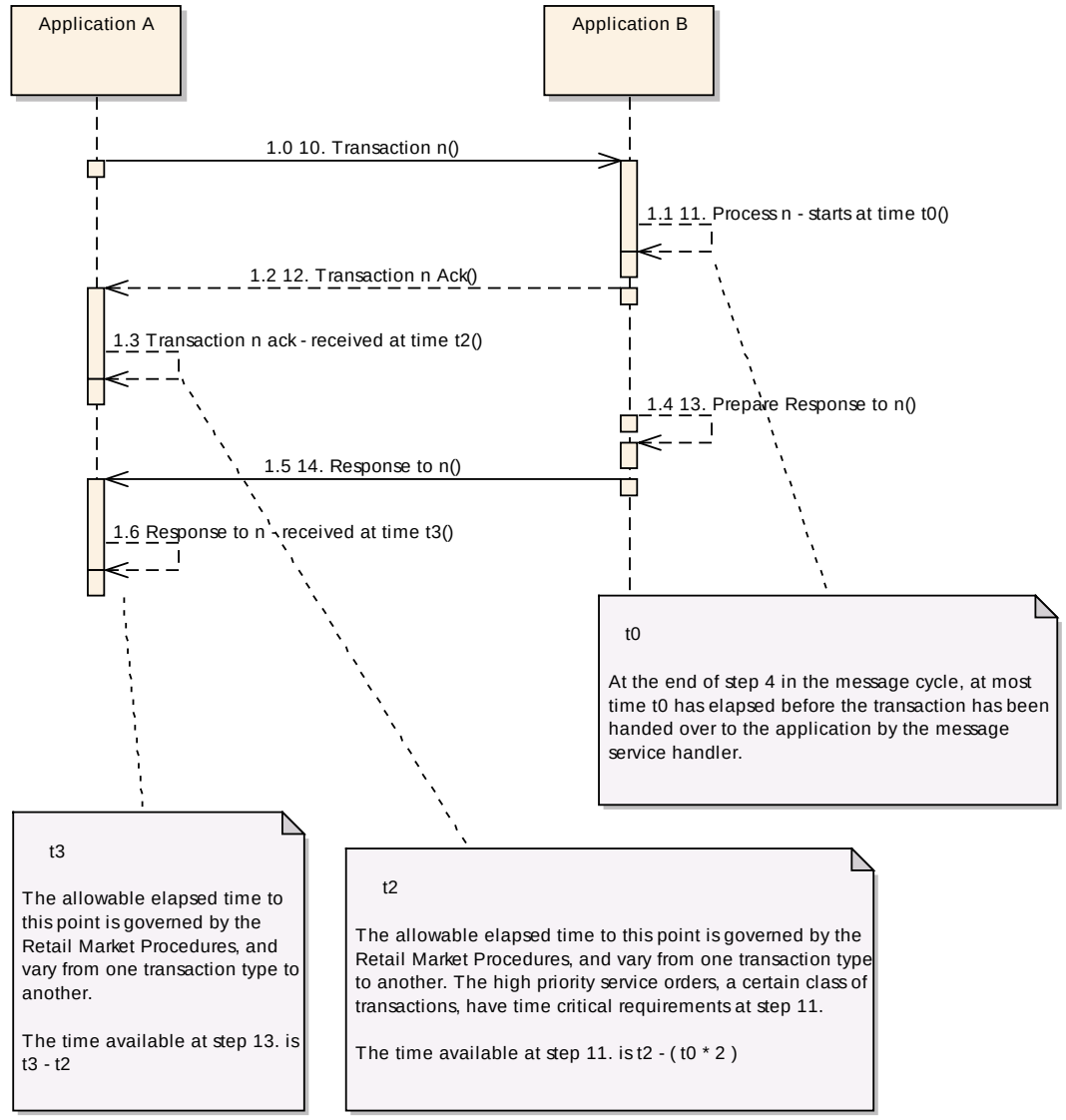

Figure 2.4.2.1 Timing steps in the Transaction and Response Cycles

- The maximum time for completion of each of steps 10, 12, and 14 = time t0.
- Time available at step 11 = t2 – (t0 * 2)
- Time available at step 13 = t3 – t2

2.4.3 Time Definitions:

1. Allowable time drift on Message Service Handlers is ± 15 seconds; hence maximum time error will be 30 seconds.
2. t0 is the maximum time duration before transaction processing can begin.

   For high priority messages there are no retries counted, and message acknowledgement times are not included

   t0 = sum(steps 1 to 4)

   For other medium and low priority messages t0 = t1
3. t1 is the maximum time duration before message acknowledgment receipt

   The t1 times are defined in Section 2.3 of the *FRC B2B System Specifications*
4. t2 is the maximum time duration before transaction acknowledgement receipt

   The high priority t2 time is defined in Section 2.3 of the *FRC B2B System Specifications*

   All other t2 times are governed by the Retail Market Procedures and will be policed by audit.
5. t3 is the maximum time duration before a transaction response is received.

   The t3 times are governed by the Retail Market Procedures and will be policed by audit.

3. Application Layer

3.1 Transaction Application

Transactions are handled in the FBS Gateway at an application level. Transactions are expressed in aseXML. The aseXML application is not responsible for transport, routing and packaging. The applications will interact with back end systems at participant sites, and hence will need to run on disparate hardware and software platforms. Participants are responsible for the development and deployment of their own aseXML application. There will be an ongoing need for maintenance and compliance with aseXML as it evolves.

3.1.1 aseXML Document Format

An aseXML document must contain a Header that identifies the document, its originator and the destination. The document may carry either transactions or acknowledgments.

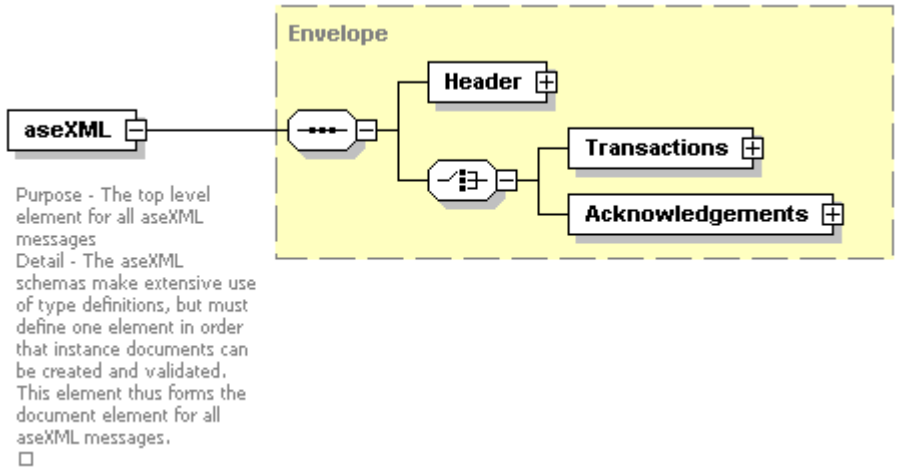

Figure 3.1.1. Top Level Format of all aseXML documents

3.1.2 aseXML Header Format

All the guidelines pertaining to the aseXML headers established in the *aseXML Standards and Guidelines* document will be adhered to. This information contained in the header is needed both by the aseXML Transaction Application and by the Message Service Handler header processor for addressing the ebXML message. The aseXML Header structure is shown in figure 3.1.2.

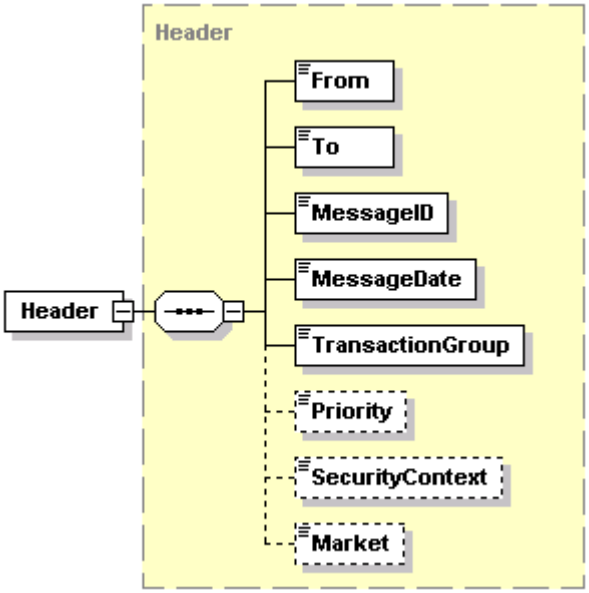

Figure 3.1.2. aseXML Header Element Format

3.1.3 aseXML Transaction Format

One or more transactions can be encapsulated within a single Message.

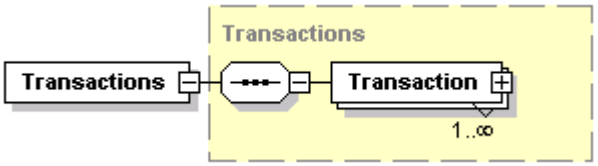

Figure 3.1.3.1. aseXML Transactions Element Format

Here an individual transaction can be present according to the types in the following figure.

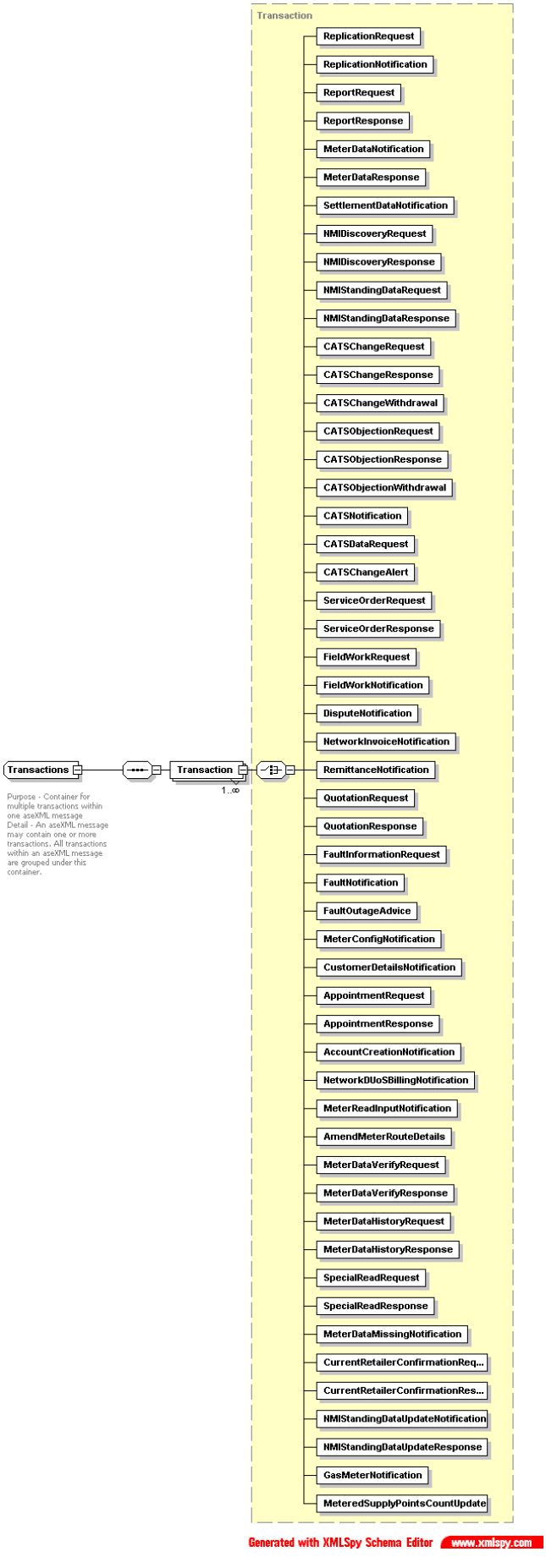
Transaction
ReplicationRequest
ReplicationNotification
ReportRequest
ReportResponse
MeterDataNotification
MeterDataResponse
SettlementDataNotification
NMIDiscoveryRequest
NMIDiscoveryResponse
NMIStandingDataRequest
NMIStandingDataResponse
CATSChangeRequest
CATSChangeResponse
CATSChangeWithdrawal
CATSObjectionRequest
CATSObjectionResponse
CATSObjectionWithdrawal
CATSNotification
CATSDataRequest
CATSChangeAlert
ServiceOrderRequest
ServiceOrderResponse
FieldWorkRequest
FieldWorkNotification
DisputeNotification
NetworkInvoiceNotification
RemittanceNotification
QuotationRequest
QuotationResponse
FaultInformationRequest
FaultNotification
FaultOutageAdvice
MeterConfigNotification
CustomerDetailsNotification
AppointmentRequest
AppointmentResponse
AccountCreationNotification
NetworkDUoSBillingNotification
MeterReadInputNotification
AmendMeterRouteDetails
MeterDataVerifyRequest
MeterDataVerifyResponse
MeterDataHistoryRequest
MeterDataHistoryResponse
SpecialReadRequest
SpecialReadResponse
MeterDataMissingNotification
CurrentRetailerConfirmationReq...
CurrentRetailerConfirmationRes...
NMIStandingDataUpdateNotification
NMIStandingDataUpdateResponse
GasMeterNotification
MeteredSupplyPointsCountUpdate
Transactions
Transaction
1..∞
Purpose - Container for multiple transactions within one aseXML message
Detail - An aseXML message may contain one or more transactions. All transactions within an aseXML message are grouped under this container.
Generated with XMLSpy Schema Editor
www.xmlspy.com

Figure 3.1.3.2. aseXML Transaction Element Types

(on previous page)

3.1.4 aseXML Acknowledgment Format

An individual message may carry any number of transaction acknowledgments. The part of the aseXML protocol that will not be used is the aseXML Message Acknowledgement. There will be no aseXML Message Acknowledgement in the FBS.

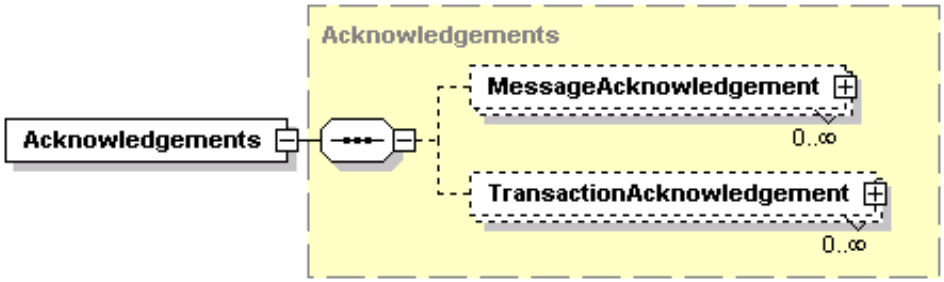

Figure 3.1.4.1. aseXML Acknowledgments Element Format

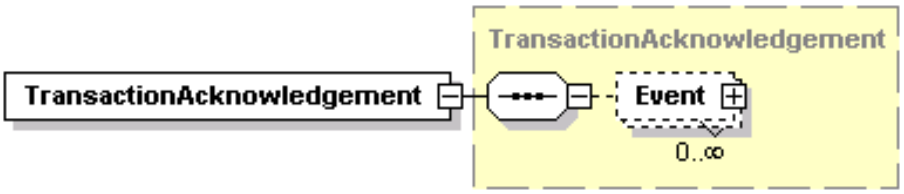

Figure 3.1.4.2. Transaction Acknowledgment Format

Event codes are those provided for by aseXML and are described in *Participant Build Pack 2*.

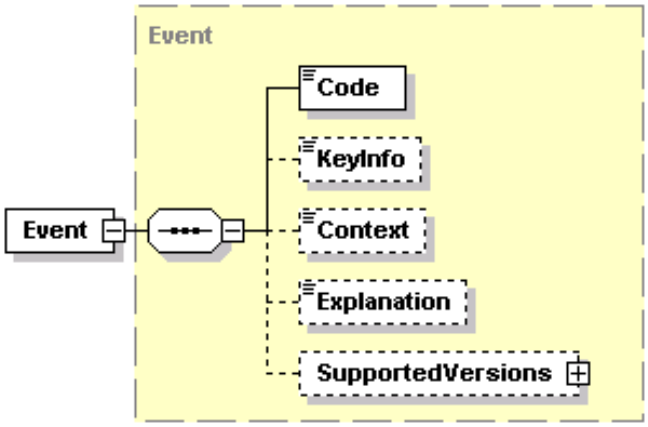

Figure 3.1.4.3. aseXML Event Element Format

Supported Versions element is made of at least one valid supported version element.

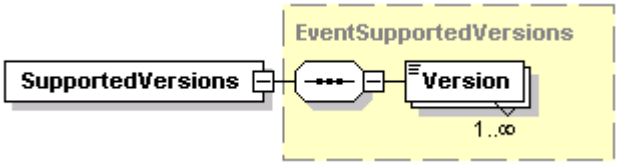

Figure 3.1.4.4. aseXML Supported Versions Element Format

3.1.5 Transaction Obligations

1. A participant receiving a transaction is obliged to respond with an aseXML transaction acknowledgement. This standard is established by the *aseXML Standards and Guidelines* document.
2. A participant receiving a suspected duplicate transaction, based on Transaction ID, is obliged to respond in accordance with the rules for handling duplicate transactions in the aseXML Standards and Guidelines document.
3. Transaction ordering remains a Transaction Application problem. Ordering is not managed by ebXML Message Service Handlers.
4. Multiple transactions may be grouped into a single aseXML document but when this is done the transactions all must be in the same transaction group.
5. The business rules implied by the aseXML schema are the subject of a joint industry working-group, the aseXML Standards Working Group (ASWG). Conformance to those rules is expected of the Transaction Application.

3.1.6 Schema Development

The aseXML schema is under the control of the ASWG. This group or its successor will mandate the schema that describes the creation of valid aseXML documents.

3.1.7 Version control

The current version of the aseXML schema is one of the artefacts listed in the Gas Interface Protocol.

AEMO may only amend the Gas Interface Protocol after AEMO has undertaken a consultation process prescribed in part 15B of the National Gas Rules (NGR). Any amendments will come into effect on the date of their publication on AEMO's website or such later date as is specified by AEMO on its website in relation to those amendments.

3.1.8 Interoperability

The FBS administration will provide a reference participant site – the FBS Certification Gateway, against which participants can certify that their aseXML documents conform to the schema and are valid and interoperable. This site will be addressable using its participant Id, which is stated in Section 4.3.1 of the *FRC B2B System Specifications* document. This Id will conform with, and be used in the same fashion, as other participant Id's.

3.2 External Connectivity

In addition to the interface with the aseXML Transaction Application, participant systems will need components that provide communications infrastructure, and public key infrastructure. The level of sophistication of this infrastructure is the responsibility of the individual participants.

3.2.1 Communications Infrastructure

Communications infrastructure is required in the participant gateway for the delivery of errors and alerts. This may be as simple as error logs, or automated email alerts; or it may be part of a much larger enterprise messaging-infrastructure.

3.2.2 Public Key Infrastructure

Public Key Infrastructure requirements for the FBS are modest and could be managed by a system operator copying files from one location to another; however it is likely that participant enterprises have deployed, or plan to deploy, enterprise wide PKI solutions.

The FBS Certificate Authority will interoperate with the participant PKI solution, which in turn should interoperate with the Message Service Handler.

3.3 Message Service Interface

The message service interface is the mechanism by which the Message Service Handler interacts with other components in the participant system. These interfaces may be in the form of API's or GUI's.

The major roles it will need to play are:

3.3.1 Document management

Mediate the transmission of aseXML documents to and from the Message Service Handler to the Transaction Application including the formatting of ebXML headers from aseXML documents. The definitive mappings for this process are defined in Section 4 of the *FRC B2B System Specifications* document.

3.3.2 Communications Infrastructure Interface

Participants may find it useful to have their Message Service Interface interact directly with existing communications applications or infrastructure. The need for this will be driven by errors and alerts raised both by their own gateway deployment, and by error and alerts reported to their Message Service Handler by the hub MSH, or by other participant MSH's.

Standard Service and Action names will be defined. Service names and Action mappings are described in Section 4.1 of the *FRC B2B System Specifications* document.

3.3.3 Public Key Infrastructure Interface

The participant public key infrastructure will need to be accessible and interoperable with the Message Service Handler.

4. Message Layer

4.1 Message Service Handler

The aspects of ebXML that have led to us adopting the Housley recommendation for it as the FRC B2B Transport Routing and Packaging (TRP) solution is that it delivers a system that will put the Message Service Handling within appropriate messaging infrastructure. Key elements of this infrastructure are:

- Message level ping service
- Message status request
- Reliable messaging implementation
- XMLDSIG signing for authentication and non-repudiation of receipt.

This end-to-end functionality could not easily be achieved using aseXML alone. To do this would mean the management boundaries for TRP between the gateway and application would become substantially blurred. Considerable custom TRP software would need to be written by participants for the Transaction Application, and substantial additional work would be needed from the aseXML Standards Working Group (ASWG). Furthermore the ongoing maintenance of the TRP standard and the infrastructure that supports it would be a perpetual R&D problem for the ASWG.

There are commercially available tools that implement the ebXML messaging service. AEMO strongly recommends that participant organizations make use of such tools.

4.2 ebXML Message Service Specification

The ebXML Message Service Specification is one of a series of ebXML specifications produced by OASIS UN/CEFACT. The complete set of specifications is available at http://www.ebxml.org. The set of specifications enables a modular, complete electronic business framework.

The *ebXML Message Service Specification v 1.0* is the only piece of the ebXML framework we will be deploying in this version of the Hub. In adopting ebXML it means the FBS will be able to move to establishing a full ebXML Web Services framework when this becomes appropriate, without participants needing to replace existing infrastructure.

The ebXML Message Service Specification focuses on defining a communications-protocol neutral method for exchanging the electronic business messages. It defines specific enveloping constructs that support reliable and secure delivery of business information.

The specification defines a flexible enveloping technique that permits ebXML-compliant messages to contain payloads of any format type. This implies maximum flexibility both for participants who wish to use their gateway for other purposes, and for the extensibility of the FBS.

Participant Message Service Handlers will be required to conform to *ebXML Message Service Specification v 1.0*, according to the configuration descriptions herein and the specifications in the *FRC B2B System Specifications* document.

4.3 Packaging

An ebXML Message is a communication protocol independent MIME/Multipart message envelope, structured in compliance with the SOAP Messages with Attachments [SOAPATTACH] specification, referred to as a Message Package.

There are two logical MIME parts within the Message Package:

- A MIME part, referred to as the Header Container, containing one SOAP 1.1 compliant message. This XML document is referred to here as a *SOAP Message.*
- Zero or more MIME parts, referred to as Payload Containers, containing application level payloads.

The SOAP Message is an XML document that consists of the SOAP Envelope element. This is the root element of the XML document representing the SOAP Message. The SOAP Envelope element consists of the following:

- One SOAP Header element. This is a generic mechanism for adding features to a SOAP Message, including ebXML specific header elements.
- One SOAP Body element. This is a container for message service handler control data and information related to the payload parts of the message.

Carrying ebXML headers in *SOAP Messages* does not mean that ebXML overrides the existing semantics of SOAP, but rather that the semantics of ebXML over SOAP maps directly onto SOAP semantics.

These package structure details are for informative purposes about the relationship between the ebXML and SOAP, as this relationship has only recently stabilised. Specific implementation details here refer directly to ebXML configuration in conjunction with the *ebXML Message Service Specification v 1.0*. The aspects pertaining to the SOAP wrapper described here will largely be invisible to participants deploying tool-based implementations of the Message Service Handler

The general structure and composition of an ebXML Message is described in the following figure.

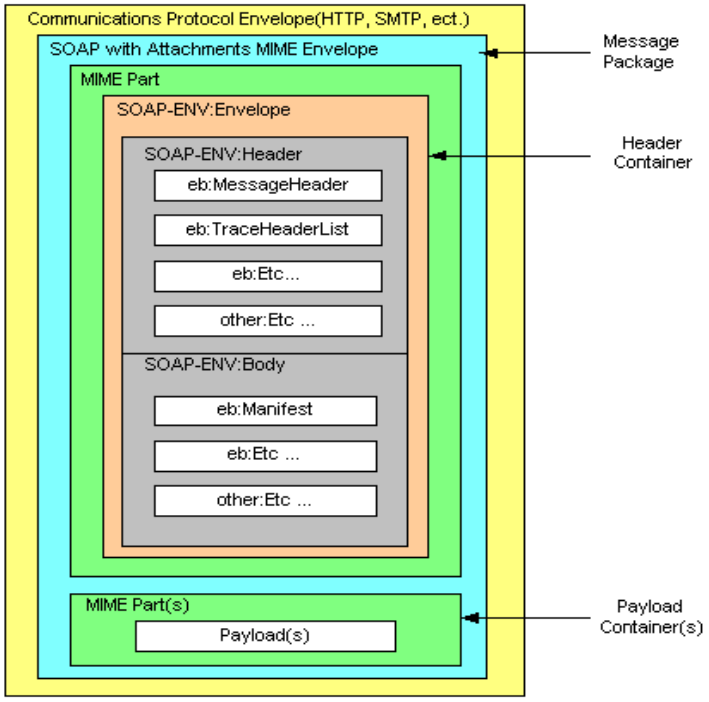

Figure 4.3. ebXML Message Structure and Composition.

The specific parts of the ebXML packaging elements and attributes that require tailored configuration for the FBS are as follows:

4.3.1 Payload

In the FBS, the only payload cargo carried will be a single aseXML document. The aseXML document may contain one or more transactions or transaction-acknowledgements.

The contents of each Payload Container must be identified by the ebXML Message Manifest element within the SOAP Body. The mechanism for doing this is described in the *ebXML Message Service Specification v 1.0*

The ebXML Message Service Specification makes no provision, nor limits in any way, the structure or content of application payloads. Payloads can be simple plain-text objects or complex nested multipart objects. The specification of the structure and composition of payload objects is the prerogative of the organization that defines the business process or information exchange that uses the ebXML Message Service.

4.3.2 MessageHeader element

The MessageHeader element is required in all ebXML Messages. It must be present as a child element of the SOAP Header element.

The MessageHeader element is a composite element comprised of the following ten subordinate elements:

4.3.2.1 From and To elements

The From element identifies the party that originated the message. The To element identifies the party that is the intended recipient of the message.

The From and the To elements each contain one or more PartyId child elements.

A uniform addressing scheme for the PartyId element in the From and To elements is specified in Sections 4.3.2 and 4.3.3 of the *FRC B2B System Specifications* document. This defines how participants address messages. This scheme is based on the Participant ID naming system agreed to by the VGRRC.

4.3.2.2 CPAId

The CPAId element is a string that identifies the parameters governing the exchange of messages between the parties.

A mechanism for deriving a unique entry for this field is described in Section 4.3.4 of the *FRC B2B System Specifications* document.

4.3.2.3 ConversationId

The ConversationId element is a string identifying the set of related messages that make up a conversation between two Parties. This element is mandatory but currently has no role in the FBS. The mechanism for handling this field is described in Section 4.3.6 of the *FRC B2B System Specifications* document.

4.3.2.4 Service

The Service element identifies the service that acts on the message. Service element will be used to describe different services to the Message Service Interface. Service names will be mapped to priority. Service names and mappings are described in Section 4.1.1 of the FRC B2B System Specifications document.

4.3.2.5 Action

The Action element identifies a process within a Service that processes the Message. Action names and mappings are described in Section 4.1.2 of the *FRC B2B System Specifications* document.

4.3.2.6 MessageData

The MessageData element provides a means of uniquely identifying an ebXML Message. It contains the following four subordinate elements, which should be configured as follows:

- MessageId – The element MessageId is a unique identifier for the message, conforming to [RFC2392]. The "local part" of the identifier as defined in [RFC2392] is

implementation dependent and a uniform scheme for deriving this element is described in Section 4.3.5 of the *FRC B2B System Specifications* document.

- Timestamp – Configured as per Section 2.4 of the *FRC B2B System Specifications* document.
- RefToMessageId - Configured as per the *ebXML Messaging Services Specification V1.0*.
- TimeToLive - Configured as per Section 4.2.5 of the *FRC B2B System Specifications* document.

4.3.2.7 QualityOfServiceInfo

The QualityOfServiceInfo element identifies the quality of service with which the message is delivered. This element has three attributes:

- deliverySemantics
- messageOrderSemantics
- deliveryReceiptRequested

The deliverySemantics attribute indicates whether or not a message is sent reliably and needs to be set to OnceAndOnceOnly.

MessageOrderSemantics and deliveryReceiptRequested attributes should be their respective default values.

4.3.2.8 SequenceNumber

The ITDF considered a uniform scheme for populating this element for use in conjunction with the optional ConversationId element to facilitate the delivery of sequential messages. It was decided not to use this element.

4.3.2.9 Description

Configured as per the ebXML Messaging Services Specification V1.0.

4.4 Routing

4.4.1 TraceHeaderList element

A TraceHeaderList element consists of one or more TraceHeader elements.

4.4.2 TraceHeader element

While the From and To elements contain participant addresses, routing between two participants via the hub is achieved by configuring the TraceHeader to address the hub.

The TraceHeader element contains information about a single transmission of a message between two instances of a MSH. If a message traverses multiple hops by passing through one or more intermediate MSH nodes as it travels between the From Party MSH and the To Party MSH, then each transmission over each successive “hop” results in the addition of a new TraceHeader element by the Sending MSH.

In the FBS participants will send messages with one intermediate hop, that being the hub. Methodology for doing this is described in the *ebXML Messaging Service Specification V1.0*. The hub address to be used in this field is specified in Section 4.4.2 of the *FRC B2B System Specifications* document.

4.4.3 Via element

The Via element is an ebXML extension to the SOAP Header that is used to convey information to the next ebXML Message Service Handler (MSH) due to receive the message.

This MSH may be a MSH operated by an intermediary, or it may be the To party. In particular, the Via element is used to hold data that can vary from one hop to another.

Treatments of certain of its attributes are significant for participants in the FBS. Those attributes are described here.

4.4.3.1 syncReply attribute

This attribute should not be present; this is semantically equivalent to its presence with a value of "false".

4.4.3.2 reliableMessagingMethod attribute

This attribute should not be present; this is semantically equivalent to its presence with a value of "ebXML".

4.5 Delivery

A single aseXML document may hold one or more aseXML transactions or one or more aseXML transaction acknowledgements, but not a mix of transactions and transaction acknowledgements. Given there may be more than one transaction per document, ebXML reliable messaging alone is not sufficient to ensure application receipt of all transactions. Every aseXML transaction requires a transaction acknowledgement. From within the Transaction-Application there will be a Transaction Acknowledgement for each transaction verifying that the transaction is available to the application.

For the purposes of the ebXML Message Service Handler (MSH), aseXML documents containing transactions or transaction acknowledgements are undifferentiated as message payload documents.

Every message in the FBS will receive a message acknowledgement using the ebXML reliable messaging protocol. The physical process that transpires during a successful message delivery is as per figure 4.

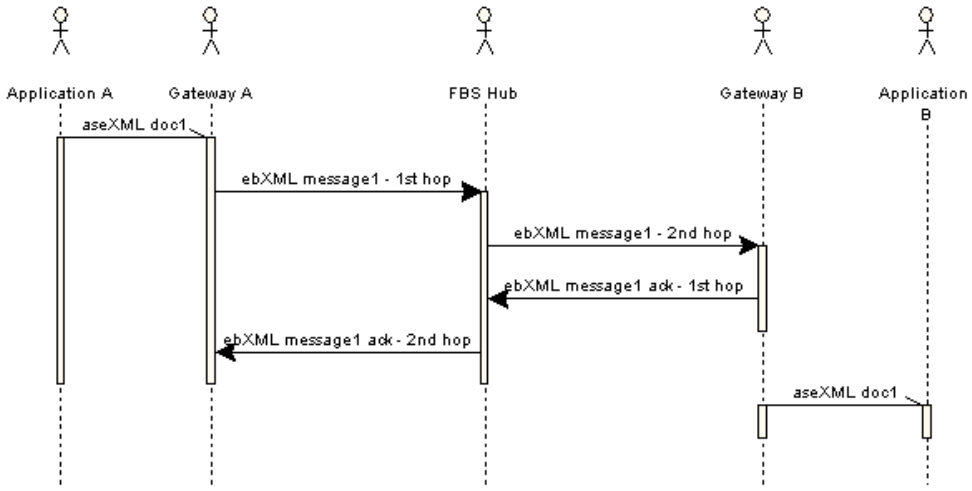

Figure 4.5. ebXML Message Delivery Sequence

The FBS hub performs routing, protocol management, logging, and audit support. It does not participate as a FROM or TO party in the reliable messaging process. It does not act as a store-and-forward interim step in the message acknowledgement cycle. The sender of a message will expect to receive an acknowledgement message from the intended recipient routed via the FBS Hub, but it will not expect an acknowledgement from the hub itself. Under this regime the system provides authenticated and non-repudiable end-to-end delivery of aseXML documents between participants.

Given the error message functionality that forms part of the Reliable Messaging specification, and the message status services, the system will deliver highly dependable, highly automated, self-documenting message delivery.

4.5.1 Reliable Messaging

Delivery of messages within the FBS will conform to the *ebXML Message Service Specification v 1.0*. The details can be found in Chapter 9 of that document.

Reliable messaging is a protocol that provides a mechanism whereby two MSH can reliably exchange messages that are sent using 'reliable message' semantics ensuring that the *To Party* receives the message once and once only.

It is the case that messages routed through the hub may be subject to specific regulatory timing requirements. As we are using the Internet and stateless protocols (HTTP), it is imperative that the sending MSH is responsible for escalation in the event that a timing requirement is not met. Facilities to support this are part of the ebXML protocol, in the form of the TimeToLive, and persistDuration elements. Further information regarding this can be found in the chapter entitled Message Service Interface. Notwithstanding the above, as part of the Reliable Messaging specification, it is incumbent on any MSH to send a delivery failure notification where appropriate. See the Failed Message Delivery section in this chapter.

4.5.2 Persistent Storage and System Failure

A MSH that participates as a gateway in the FBS must keep messages in persistent storage. After a system interruption or failure the MSH must ensure that messages that are in persistent storage are processed in the same way as if the system failure or interruption had not occurred.

These requirements are described in detail in the *ebXML Message Service Specification v 1.0*.

4.5.3 Reliable messaging parameters

These parameters are to be used by participants in the FBS according to the following guidelines.

4.5.3.1 deliverySemantic

All messages in the FBS will have the deliverySemantic value set to OnceAndOnlyOnce. With MSH's that conform to the Reliable Messaging Protocol, the deliverySemantic of OnceAndOnlyOnce will mean that the Transaction Application or other process at the To Party will receive the message once and only once

Participants will be required to support and deploy the OnceAndOnlyOnce semantic.

4.5.3.2 MshTimeAccuracy

The mshTimeAccuracy parameter indicates the minimum accuracy a Receiving MSH keeps the clocks it uses when checking, for example, TimeToLive. Its value is in the format "mm:ss" which indicates the accuracy in minutes and seconds. This value is specified in Section 4.2.1 of the *FRC B2B System Specifications* document.

4.5.3.3 TimeToLive

The TimeToLive value indicates the time by which a message should be delivered to and processed by the To Party. It must conform to an XML Schema timeInstant.

In this context, the TimeToLive has expired if the time of the internal clock of the Receiving MSH is greater than the value of TimeToLive for the message.

Maximum or absolute TimeToLive values are specified in Section 4.2.5 of the *FRC B2B System Specifications* document.

4.5.3.4 ackRequested

The ackRequested value is used by the Sending MSH to request that the Receiving MSH returns an acknowledgment message with an Acknowledgment element.

All messages in the FBS will have the value of ackRequested set to Signed, which indicates that a signed Acknowledgement is requested. Upon the completion of this signed message and signed acknowledgement pair, non-repudiation of the message contents will be in effect.

4.5.3.5 retries

The retries value is an integer value that specifies the maximum number of times a Sending MSH should attempt to redeliver an unacknowledged message using the same Communications Protocol. This value is specified in Section 4.2.2 of the *FRC B2B System Specifications* document.

4.5.3.6 retryInterval

The retryInterval value is a time value, expressed as duration in accordance with the [XMLSchema] timeDuration data type. This value specifies the minimum time the Sending MSH MUST wait between retries, if an Acknowledgment Message is not received. This value is in Section 4.2.3 of the *FRC B2B System Specifications* document.

4.5.3.7 persistDuration

The persistDuration value is the minimum length of time, expressed as a [XMLSchema] timeDuration that data from a reliably sent *Messag*e, is kept in Persistent Storage by a Receiving MSH. If the persistDuration has passed since the message was first sent, a Sending MSH should not resend a message with the same MessageId. If a message cannot be sent successfully before persistDuration has passed, then the Sending MSH should escalate the failure to an appropriate level by deploying a service from the Message Service Interface.

Maximum or absolute PersistDuration values are specified in Section 4.2.4 of the *FRC B2B System Specifications* document.

4.5.4 ebXML reliable messaging protocol

The ebXML Reliable Messaging Protocol described in *ebXML Message Service Specification v 1.0* must be adhered to by Message Service Handlers, in conjunction with the parameter settings described in the previous section. The five parts of the protocol thoroughly described in that document are:

1. Sending message behaviour
2. Receiving message behaviour
3. Generating an acknowledgement message
4. Resending lost messages and duplicate filtering
5. Duplicate message handling

Given the protocols are adhered to, the receipt of the Acknowledgment Message indicates that the message being acknowledged has been successfully received and either processed or persisted by the Receiving MSH. An Acknowledgment Message must contain a MessageData element with a RefToMessageId that contains the same value as the MessageId element in the message being acknowledged.

The ebXML Acknowledgement Message makes the aseXML Message Acknowledgement redundant. Participants will not send aseXML Message Acknowledgements to the FBS.

4.5.4.1 Generating an acknowledgement message

There is an important clarification to section 9.3.3 of the ebXML Message Service Specification v 1.0. In the FBS implementation of ebXML Message Service the Acknowledgement element will be sent asynchronously and the value of the message header elements must be set as per the ebXML specification with the following clarification.

- The From element must be populated with the To element extracted from the message received and this is the only PartyId element from the message received to be included in this From element.
- The To element must be populated with the From element extracted from the message received and this is the only PartyId element from the message received to be included in this To element.

4.5.5 Failed message delivery

If a message cannot be delivered, the MSH or process must send a delivery failure notification to the *From Party*. A description of this and all error handling services are described in the *ebXML Message Service Specification v 1.0.*

4.6 Message Service Handler Services

The FBS Message Service Handlers will support two services that are designed to help provide smooth operation of the FBS:

- Message Status Request
- Message Service Handler Ping

4.6.1 Message Status Request Service

This service is to be implemented by all participants, according to the ebXML Messaging Service Specification V1.0, noting that the methodology for participants is Reliable Messaging.

4.6.2 Message Service Handler Ping

This service is to be implemented by all participants, according to the ebXML Messaging Service Specification V1.0.

5. Transport Layer

5.1 Message Transport Interface

The Message Transport Interface is concerned with the communication protocol bindings and technical details for carrying *ebXML Message Service* messages for the following communication protocols:

- Hypertext Transfer Protocol [HTTP]
- Simple Mail Transfer Protocol [SMTP]

HTTP is the only communication protocol that is supported in the FBS.

5.1.1 HTTP/S

Hypertext Transfer Protocol Version 1.1 [HTTP] (http://www.ietf.org/rfc2616.txt) is the minimum level of protocol that MUST be used. All communication will be done by asynchronous HTTP post.

A non standard port will be used for HTTP/S. The port to use is specified in Section 4.4.2 of the *FRC B2B System Specifications* document.

There are required specifications concerning the implementation of ebXML over HTTP in appendix B of the ebXML Messaging Service Specification V1.0.

5.2 Network Infrastructure

5.2.1 Topology

The topology of the B2B solution is spoke and hub, as per the Housley report recommendations. In this topology each participant is represented as a member node at the end of a spoke. Member nodes communicate with each other via a message exchange hub. This is a two-hop process - sending node to hub – hub to receiving node.

5.2.2 Gateway

5.2.2.1 Message Service Handler

The gateway Message Service Handler is required to perform conforming ebXML transactions with other participants. This implies appropriately addressed, digitally signed ebXML messages and acknowledgements. Beyond complying with the requirements of the FBS, the actual gateway technology is not mandated.

It is anticipated that participants will deploy an appropriately configured commercially available ebXML gateway solution rather than develop their own.

5.2.2.2 Certification

The MSH gateways test and certify themselves against reference sites provided by the FBS administration. These sites will supply services

- To provide for verification of correct aseXML schema use by participant application software.
- To provide verification of correct ebXML gateway configuration

The certification site will not provide a test suite for verification of business rules deployed in participant application software.

5.2.3 Hub

5.2.3.1 Access Methods

The hub will be configured to accept messages from participants connected to the Internet and MarketNet.

These existing network options in conjunction with the proposed protocols provide appropriate levels of security and reliability for B2B transactions, including faults.

6. Security

An Internet based message service, by its very nature, presents certain security risks. An ebXML Message Service may be at risk by means of:

- Unauthorized access Data integrity and/or confidentiality attacks (e.g. through man-in-the-middle attacks)
- Denial-of-Service and
- IP spoofing

Each security risk is described in detail in ebXML Technical Architecture Risk Assessment – http://www.ebxml.org/specs/secRISK.pdf. Beyond the requirements herein, participants should make themselves familiar with these risks and institute countermeasures balanced against an assessment of the inherent risks and the value of the asset(s) that might be placed at risk.

The system will require some Public Key Infrastructure. The system will employ transport layer encryption via SSL or S/MIME to protect the data in transit, and as part of the ebXML reliable messaging solution, will provide signed messages and signed acknowledgements, which will provide authentication and non-repudiation.

6.1 Key management

Key management is a major issue that needs to be addressed with respect to the capabilities of the Message Service Handler. In particular, the MSH will be called upon to apply digital signatures; the appropriate private keys must be available to the MSH. Private keys must be managed very carefully and deliberately. Thus, some configuration will be necessary to establish the key management mechanisms to be used in the particular FBS Gateway. Gateway Public Key Infrastructure (PKI) will be the responsibility of participants as described earlier in Chapter 3 of this document.

The FBS Certificate Authority will issue and distribute certificates, but local administration of keys will be the responsibility of an enterprise level PKI solution by each of the participants.

For additional information, refer to the Participant User Guide.

6.2 Encryption

The system will employ transport layer encryption via SSL to protect the data in transit. The keys used will be 128 bit X509v3 and will be provided by the FBS administration.

In the FBS there will not be:

- ebXML payload encryption within the aseXML document
- element level encryption within the aseXML document

6.3 Digital Signature

To create a digital signature for a message, the data to be signed is transformed by an algorithm that takes as input the private key of the sender. Because a transformation determined by the sender's private key can only be undone if the reverse transform takes as a parameter the sender's public key, a recipient of the transformed data can be confident of the origin of the data (the identity of the sender). If the data can be verified using the sender's public key, then it must have been signed using the corresponding private key (to which only the sender should have access).

For signature verification to be meaningful, the verifier must have confidence that the public key does actually belong to the sender. A certificate, issued by the Certificate Authority, is an assertion of the validity of the binding between the certificate's subject and their public key such that other users can be confident that the public key corresponds to the subject.

In the FBS, the ebXML payload (i.e. the aseXML document) will be signed. The ability to do this needs be a feature of the ebXML MSH product that is chosen by participants.

The algorithms and key configurations used by this process are described in detail in Chapter 11 of the *ebXML Message Service Specification v 1.0*.