1 Communication Protocol Bindings

1.1 Introduction [snip]

1.2 HTTP [snip]

1.3 SMTP

The Simple Mail Transfer Protocol [RFC821] and its companion documents [RFC822] and [ESMTP] makeup the suite of specifications commonly referred to as Internet Electronic Mail. These specifications have been augmented over the years by other specifications, which define additional functionality "layered on top" of these baseline specifications, these include:

- Multipurpose Internet Mail Extensions (MIME) [RFC2045], [RFC2046], [RFC2387]
- SMTP Service Extension for Authentication [RFC2554]
- SMTP Service Extension for Secure SMTP over TLS [RFC2487]

Typically, Internet Electronic Mail Implementations consist of two "agent" types:

- Message Transfer Agent (MTA)
  - Programs that send and receive mail messages with other MTA's on behalf of MUA's. Microsoft Exchange Server is an example of a MTA
- Mail User Agent (MUA)
  - Electronic Mail programs are used to construct electronic mail messages and communicate with an MTA to send/retrieve mail messages. Microsoft Outlook is an example of a MUA.

MTA's often serve as "mail hubs" and can typically service hundreds or more MUA's.

MUA's are responsible for constructing Electronic Mail messages in accordance with the Internet Electronic Mail Specifications identified above. This section describes the "binding" of an ebXML compliant message for transport via e-mail from the perspective of a MUA. No attempt is made to define the binding of an ebXML message exchange over SMTP from the standpoint of a MTA.

1.3.1 Minimum level of supported protocols

- Simple Mail Transfer Protocol [RFC821] and [RFC822]
- MIME [RFC2045] and [RFC2046]
- Multipart/Related MIME [RFC2387]

1.3.2 Sending ebXML messages over SMTP

- Prior to sending messages over SMTP an ebXML message MUST be formatted according to ebXML Message Service Specification section **xx.yy**. Additionally the messages must also conform to the syntax, format and encoding rules specified by MIME [RFC2045], [RFC2046] and [RFC2387].

- Many types of data that a party might desire to transport via email are represented as 8bit characters or binary data. Such data cannot be transmitted over SMTP [RFC821], which restricts mail messages to 7bit US-ASCII data with lines no longer than 1000 characters including any trailing CRLF line separator. If a sending Message Service Handler knows that a receiving MTA, or ANY intermediary MTA's, are restricted to handling 7-bit data then any ebXML header or payload data that uses 8 bit (or binary) representation must be "transformed" according to the encoding rules specified in section 6 of [RFC2045]. In cases where a Message Service Handler knows that a receiving MTA and ALL intermediary MTA's are capable of handling 8-bit data then no transformation is needed on any part of the ebXML message.

- The rules for forming an ebXML message for transport via SMTP are as follows:

    1. **If using [RFC821] restricted transport paths**, apply transfer encoding to all 8-bit data that will be transported in a ebXML header or payload body part, according to the encoding rules defined in section 6 of [RFC2045]. The Content-Transfer-Encoding MIME header MUST be included in the MIME envelope portion of any body part that has been transformed (encoded).

    2. The Content-Type: Multipart/Related MIME header with the associated parameters, from the ebXML Message Envelope MUST appear as an email MIME header.

    3. All other MIME headers that constitute the ebXML Message Envelope MUST also become part of the email MIME header.

    4. The mandatory SOAPAction MIME header field must also be included in the e-mail MIME header and must have the value of ebXML:

        SOAPAction: **ebXML**

        Where Service and Action are values of the corresponding elements from the ebXML MessageHeader.

    5. The "MIME-Version: 1.0" header must appear as an email MIME header.

    6. The e-mail header "To:" MUST contain the [RFC822] compliant e-mail address of the ebXML message service handler.

    7. The e-mail header "From:" MUST contain the [RFC822] compliant e-mail address of the senders ebXML message service handler.

    8. Construct a "Date:" e-mail header in accordance with [RFC822]

    9. Other headers MAY occur within the e-mail message header in accordance with [RFC822] and [RFC2045], however ebXML Message Service Handlers MAY choose to ignore them.

The example below shows a MINIMAL example of an e-mail message containing an ebXML message:

```
From: ebXMLhandler@imacompany.com
To: ebXMLhandler@imacompany2.com
Date: Thu, 08 Feb 2001 19:32:11 CST
MIME-Version: 1.0
SOAPAction: ebXML
Content-type: multipart/related; boundary="BoundarY"; type="text/xml";
      start=" <ebxhmheader111@imacompany.com>"

--BoundarY
Content-ID: <ebxhmheader111@imacompany.com>
Content-Type: text/xml
```

```
<SOAP-ENV:Envelope  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'>
<SOAP-ENV:Header xmlns:ebh1='http://www.ebxml.org/namespaces/ebxmlheader1'>
    <MessageHeader mustUnderstand="1">
        <From>
            <PartyId>urn:duns:123456789</PartyId>
        </From>
        <To>
            <PartyId>urn:duns:912345678</PartyId>
        </To>
        <CPAId>20001209-133003-28572</CPAId>
        <ConversationId>20001209-133003-28572</ConversationId>
        <Service>OrderProcessing</Service>
        <Action>NewORder</Action>
        <MessageData>
            <MessageId>imacompany.com.20001209-133003-28572</MessageId>
            <TimeStamp>20010215111212Z</TimeStamp>
        </MessageData>
        <ReliableMessagingInfo>NEED THIS FROM LATEST
SPEC</ReliableMessagingInfo>
    </MessageHeader>

</SOAP-ENV:Header>
<SOAP-ENV:Body  xmlns:ebh2='http://www.ebxml.org/namespaces/ebxmlheader2'>
    <Manifest mustUnderstand="1">
        <Reference xlink:href="urn:cid:ebxmlpayload111@imacompany.com"
             xlink:role="XLinkRole"
           xlink:type="simple"
             xlink:label="XLinkLabel">
             <Description xml:lang="en-us">Purchase Order 1</Description>
        </Reference>
    </Manifest>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--BoundarY
Content-ID: <ebxmlpayload111@imacompany.com>
Content-Type: text/xml

<purchase_order>
    <po_number>1</po_number>
    <part_number>123</part_number>
    <price currency="USD">500.00</price>
</purchase_order>

--BoundarY--
```

1.3.3 Response Messages

All ebXML response messages, including errors and acknowledgements, are delivered asynchronously between ebXML Message Service Handlers (MSH's). Each response message MUST be constructed in accordance with the rules specified in the section titled "Sending ebXML messages over SMTP" elsewhere in this document.

ebXML Message Service Handlers MUST be capable of receiving a delivery failure notification message sent by an MTA. An MSH that receives a delivery failure notification message SHOULD examine the message to determine which ebXML message, sent by the MSH, resulted in a message delivery failure. The MSH SHOULD attempt to identify the application responsible for sending the offending message that caused the failure. The MSH SHOULD attempt to notify the application that a message delivery failure has occurred. If the MSH is unable to determine the source of the offending message the MSH administrator should be notified.

MSH's which cannot identify a received message as a valid ebXML message or a message delivery failure SHOULD retain the unidentified message in a "dead letter" folder.

A MSH SHOULD place an entry in an audit log indicating the disposition of each received message.

1.3.4 Access Control

Implementers MAY protect their ebXML message service handlers from unauthorized access through the use of an access control mechanism. The SMTP access authentication process described in "SMTP Service Extension for Authentication" [RFC2554] defines the ebXML recommended access control mechanism to protect a SMTP based ebXML Message Service Handler from unauthorized access.

1.3.5 Confidentiality and Communication Protocol Level Security

An ebXML Message Service handler MAY use transport layer encryption to protect the confidentiality of ebXML messages. The IETF "SMTP Service Extension for Secure SMTP over TLS" specification [RFC2487] provides the specific technical details and list of allowable options, which may be used.

1.3.6 References

[RFC2554] Myers, J. "SMTP Service Extension for Authentication", RFC 2554, March 1999.

[RFC2487] Hoffman, P., "SMTP Service Extension for Secure SMTP over TLS", RFC 2487, January 1999

[ESMTP] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extensions", RFC 1869, November 1995.

[SASL] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[RFC821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

[RFC2045] Freed, N., Borenstein, N., "Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996

[RFC2046] Freed, N., Borenstein, N., "Multipurpose Internet Mail Extensions(MIME) Part Two: Media Types", RFC 2046, November 1996

[RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, August 1998