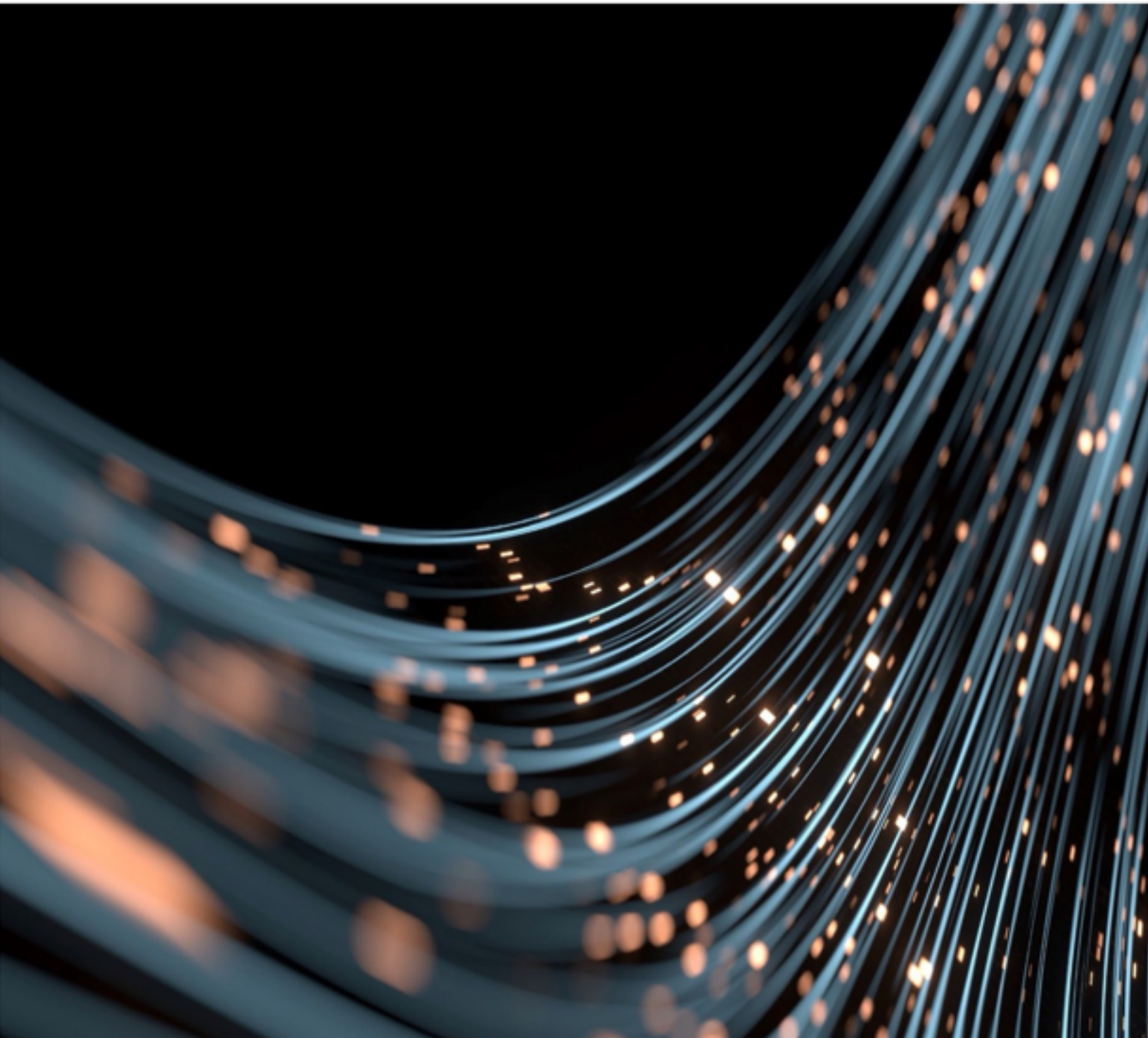
DIGITAL BUSINESS NETWORKS ALLIANCE™

XHE Envelope Profile

Version 1.0

July 2023



1 Table of Contents

2 Version History

Revision date	Version	Change description	Editor
07/20/2023	1.0	Initial version	BPC Market Pilot Technical Committee

3 Introduction

3.1 Scope

This specification is a profile of the Exchange Header Envelope (XHE) Version 1.0 standard developed jointly by the UN/CEFACT and OASIS, as published here:

https://docs.oasis-open.org/bdxr/xhe/v1.0/xhe-v1.0-oasis.html

The XHE is a digital envelope with standardized header information for relaying business documents between Access Points (Corners 2 and 3), as well as optional end-to-end security and integrity between business users (Corners 1 and 4). All business documents exchanged between Access Points in the DBNAlliance network MUST be enclosed in an XHE envelope. All XHE envelope instances send over the DBNAlliance network MUST conform to this specification.

3.2 Conformance

The keywords 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in this specification are to be interpreted as described in RFC2119 and RFC 8174 when, and only when, they appear in all capitals, as shown here.

3.3 Terms and Definitions

For the purpose of this specification, all terms shall have the definitions defined in the document Terms and Definitions of the DBNAlliance version 1.0.

3.4 Disclaimers and Copyright

Views expressed here are not necessarily those of, and should not be attributed to, any particular DBNAlliance participant or organization. They are not intended to provide business or legal advice, nor are they intended to promote or advocate a specific action, payment strategy, or product. Readers should consult with their own business and legal advisors.

This specification is the work product of the DBNAlliance, and readers are free to republish this specification in whole or in part without further permission, as long as the work is attributed to the DBNAlliance, and in no way suggests the DBNAlliance sponsors, endorses or recommends any organization or its services or products. Other product names and company names referenced within this document may be either trademarks or service marks of their respective owners.

3.5 Mandatory Use

As a network policy, all business documents and other content exchanged over the DBNAlliance network MUST be sent within an XHE envelope.

This document is the sole normative XHE Envelope Profile specification in the DBNAlliance exchange framework. All business documents exchanged between Access Points in the DBNAlliance network MUST be enclosed in an XHE envelope and all XHE envelope instances send over the DBNAlliance network MUST conform to this specification.

All contents of this document, except for the provided examples, is normative, unless otherwise specified. All examples provided are non-normative, unless otherwise specified.

Any supporting artefacts developed in relation to this document are non-normative, unless explicitly referenced from this document as being normative. In case of discrepancies between this document and any supporting or related artefact, this document contains the normative information.

4 General Use

4.1 Referencing This profile

The following identifier MUST be used when referencing this XHE envelope profile:

```
dbnaalliance-envelope-1.0
```

And when referencing the data model using the QName/Subtype Identifier scheme:

```
http://docs.oasis-open.org/bdxr/ns/XHE/1/ExchangeHeaderEnvelope::XHE##dbnalliance-envelope-1.0
```

All implementations and uses of XHE envelopes in the DBNAlliance network using the above identifiers MUST be conformant with this specification.

4.2 Envelope Technology

The technology used for business document envelopes in the DBNAlliance network is the Exchange Header Envelope (XHE) Version 1.0 OASIS Standard and XML syntax expression using the OASIS semantic identifiers as published here:

https://docs.oasis-open.org/bdxr/xhe/v1.0/os/xhe-v1.0-os-oasis.html

All XHE envelopes sent through the DBNAlliance network MUST also conform to all conformance clauses in the above specification.

4.3 Encoding

XHE envelopes in the DBNAlliance network MUST be UTF-8 encoded.

5 Payloads

5.1 General

An envelope MUST contain at least one business document.

The payload with `ID` = 1 in an envelope MUST be the principal business document of the transaction.

The envelope MAY contain additional payloads related to the principal business document as specified in section 5.3.

A `Payload` element of an envelope MUST contain either a `PayloadContent` child element or a `PayloadExternalReference` child element, i.e., it MUST NOT have both and it MUST NOT have neither. The semantic requirements of a business document MAY disallow the use of externally referenced resources.

The `Payload` element with `ID` = 1 in the envelope, i.e., the principal business document, MUST contain a `PayloadContent` element. In other words, the principal business document of the transaction MUST be conveyed within the envelope and MUST NOT be an externally referenced resource.

5.2 Batch Sending

This XHE profile does not support batch sending. An XHE envelope MUST contain exactly one principal business document transaction, i.e., the envelope MUST NOT be used for batch sending multiple business document transactions within a single envelope. Any business requirement including batch sending must be specified in a different XHE profile.

5.3 Attaching Associated Documents and Artefacts

An XHE envelope MAY contain additional payloads related to the principal business document. For example, an envelope containing a UBL Invoice may also contain a PDF rendering or other visual presentation of the invoice to assist the recipient, and it may contain supporting documentation related to the content of the invoice.

Unless explicitly specified in the business document profile or specification, the validation of additional payloads MUST NOT be required, and a receiver MUST NOT reject a document based on the processing of additional payloads. The sender of the envelope MUST NOT assume that the recipient has the required capabilities to process additional payload(s).

To avoid ambiguity, the first occurring `Payload` element within the envelope MUST contain the principal business document of the transaction.

6 Data Model

An implementer of this profile MUST implement all elements specified in the table below. Implementers MAY implement additional XHE elements, however they MUST NOT require that a recipient be capable of understanding them.

Element or attribute	Cardinality	Definition and use
XHE	1..1	Root element of the envelope.
└ XHEVersionID	1..1	The version of the XHE specification in use. This value MUST be set to: `1.0`
└ CustomizationID	1..1	A reference to this data model, formatted according to the QName/Subtype Identifier scheme. This value MUST be set to: `http://docs.oasis-open.org/bdxr/ns/XHE/1/ExchangeHeaderEnvelope::XHE##dbnalliance-envelope-1.0`
└ CustomizationID /@schemeID	1..1	Identifies the Customization ID as of type QName/Subtype Identifier scheme. This value MUST be set to: `bdx-docid-qns`
└ ProfileID	1..1	A reference to this specification. This value MUST be set to: `dbnalliance-envelope-1.0` as defined in section 4.1.
└ Header	1..1	
└└ ID	1..1	The identifier of the envelope instance. The combination of the `Header/ID` and `Header/FromParty` MUST be unique, i.e., there MUST NOT exist two envelope instances with the same `Header/ID` from the same sender.
└└ CreationDateTime	1..1	The date and time the envelope instance was created.
└└ FromParty	1..1	The party sending the envelope.
└└└ PartyIdentification	1..1	
└└└└ ID	1..1	The party identifier of the sender. MUST be in the format specified in the DBNAlliance Policy for Using Identifiers.
└└└└ ID/@schemeID	1..1	Identifies the scheme used for the party identification as specified in the DBNAlliance Policy for Using Identifiers.
└└ ToParty	1..n	The intended final recipient(s) of the envelope.
└└└ PartyIdentification	1..1	
└└└└ ID	1..1	The party identifier of the intended final recipient. MUST be in the format specified in the DBNAlliance Policy for Using Identifiers.
└└└└ ID/@schemeID	1..1	Identifies the scheme used for the party identification as specified in the DBNAlliance Policy for Using Identifiers.
└ Payloads	1..1	
└└ Payload	1..n	An envelope payload.
└└└ ID	1..1	The identifier of the payload within the envelope. The `Payload/ID` MUST be set to the ordinal position of the payload, i.e., payload IDs MUST be numbered sequentially starting with the number 1.
└└└ Description	0..1	An OPTIONAL human readable description of the payload.
└└└ ContentTypeCode	1..1	The MIME Type of the payload content. For XML payload content the `ContentTypeCode` MUST be set to: `application/xml`

Element or attribute	Cardinality	Definition and use
		For all other payload content types, the `ContentTypeCode` MUST be set to an IANA registered MIME Type.
LLL ContentTypeCode /@listID	0..1	An OPTIONAL attribute specifying that the `ContentTypeCode` value is a MIME Type. When set, this attribute MUST be set to: `MIME`
LLL CustomizationID	0..1	If defined in the business document profile or specification of the payload, this MUST be set to the Customization ID as specified therein. Otherwise, MUST NOT be used.
LLL CustomizationID /@schemeID	0..1	The identifier of the scheme used for the `CustomizationID` if one is defined.
LLL ProfileID	0..1	If defined in the business document profile or specification of the payload, this MUST be set to the Profile ID as specified therein. Otherwise, MUST NOT be used.
LLL ProfileID /@schemeID	0..1	The identifier of the scheme used for the `ProfileID` if one is defined.

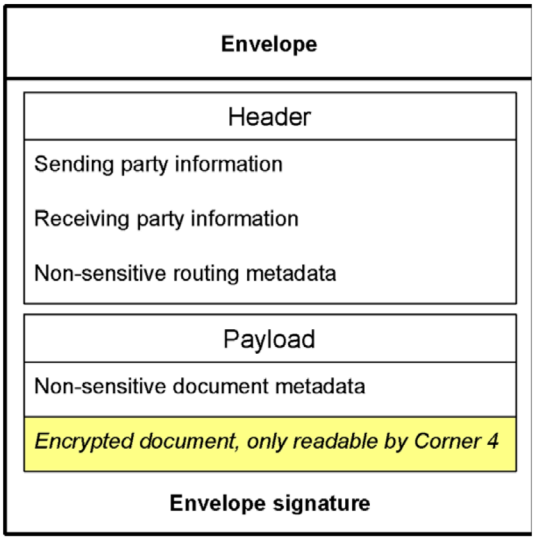
LLL InstanceEncryptionIndicator	1..1	When the payload content is encrypted, this value MUST be set to:
		`true`
		
		Otherwise, when the payload content is not encoded, this value MUST be set to:
		`false`

LLL InstanceEncryptionMethod	0..1	The method or algorithm used for encrypting payload content. When encryption is used, payloads MUST be encrypted using one of the supported encryption methods and algorithms as specified in section 7.3 and the value of this element MUST be set to the corresponding identifier.
LLL InstanceHashValue	0..0	The `InstanceHashValue` MUST NOT be included in the envelope.
LLL PayloadContent	0..1	Contains the payload content when conveyed within the envelope (see also section 5.1). When the payload content is XML then 1) it MUST have exactly one apex element in the XML element tree, 2) the XML MUST be UTF-8 encoded, and 3) inclusion of the XML in the envelope MUST NOT cause the envelope to fail schema validation. When the payload contains entirely textual information, the content MUST be encoded according to XML text encoding rules, such as the escaping of special markup characters. When the payload contains binary information, the payload content MUST be Base64 encoded.
LLL PayloadExternalReference	0..1	Contains a reference to the payload when the payload is located at an external location and not conveyed within the envelope (see also section 5.1).
LLLL ID	1..1	The absolute URL of the external payload.

7 Encryption

7.1 General

The XHE profile specification used in the DBNAlliance network supports encrypting one or more payloads of an envelope. The separation of header and payload information in the envelope facilitates Access Points to process and route envelopes without knowledge of their content. Encrypting one or more payloads of an envelope allows Senders and Receivers (Corners 1 and 4) to maintain the confidentiality of sensitive information while still allowing Access Point services to route the message, as illustrated here:



An entity registered to receive documents through the DBNAlliance network (Corner 4) MAY choose to support receiving envelopes with encrypted payloads. Likewise, an entity sending documents through the DBNAlliance network (Corner 1) MAY choose to support payload encryption when sending envelopes.

A Corner 1 MUST NOT send encrypted envelope payloads unless the Corner 4 has made a public certificate available for encryption purposes as specified in section 7.2.2.

A Corner 4 entity MUST NOT obligate its senders to encrypt envelope payloads unless such requirement is explicitly stated in the business process profile or specification.

7.2 Digital Certificates

7.2.1 Allowed Certificates

The certificates used for encrypting payloads MUST be valid for use in the DBNAlliance network in accordance with the DBNAlliance Certificate Policy and MUST be issued to the final recipient (Corner 4).

7.2.2 Use of SMP

Envelope payload encryption in the DBNAlliance is accomplished through a combination of symmetric and asymmetric encryption, using a randomly generated symmetric key: Corner 1 encrypts the envelope payload(s) using a randomly generated symmetric key. The symmetric key is encrypted using Corner 4's public key and passed to Corner 4 within the envelope, Corner 4 unencrypts the symmetric key using their own private key, and then unencrypts the payload(s) using the unencrypted symmetric key. To facilitate this, Corner 4 needs to make its public key available to Corner 1. This is done by adding the public key to the SMP `ServiceMetadata` record of the business document being transacted.

To receive envelopes with encrypted payloads, Corner 4 MUST publish a public key certificate to all endpoint elements of the SMP `ServiceMetadata` record through which they accept payload encryption as specified in section 7.2.3. The public key certificate MUST be valid in accordance with section 7.2.1, and MUST set its `TypeCode` value to:

```
dbnalliance-envelope-1.0#encryption
```

By making their public key certificate available for a given endpoint of a given business document type, Corner 4 signals that senders MAY encrypt payloads when sending exactly this business document type through exactly this endpoint.

Corner 1 MUST use the public key certificate published by Corner 4 as specified above when encrypting payloads and MUST verify that the certificate is valid as specified in section 7.2.1. Corner 1 MUST NOT send encrypted payloads to Corner 4 unless a valid certificate with a `TypeCode` as specified above is published in Corner 4's SMP record for the given endpoint and business document type.

7.2.3 SMP Data Model for Publishing the Certificate

The following table only shows the additional data to be added to Corner 4's SMP `ServiceMetadata` record when publishing a public key certificate for payload encryption. For the SMP data model in general, please refer to the DBNAlliance SMP Profile specification.

Element or attribute	Cardinality	Definition and use
ServiceMetadata		
└ *ProcessMetadata*		
└└ *Endpoint*		
└└└ **Certificate**		The SMP record MUST contain exactly one `Certificate` element in addition to any other `Certificate` elements required for the given endpoint, such as to any certificate(s) necessary for the transport protocol.
└└└└ **TypeCode**	1..1	The Certificate MUST contain a `TypeCode` element with its value set to exactly: `dbnalliance-envelope-1.0#encryption`
└└└└ **Description**	0..1	An OPTIONAL description of the certificate.

Element or attribute	Cardinality	Definition and use
└└└└ **ContentBinaryObject**	1..1	The complete base64 portion (i.e., not including the PEM header or footer) of the PEM formatted X.509 public key certificate.
└└└└ **ContentBinaryObject/@mimeCode**	1..1	An attribute specifying the MIME code of the data contained in the `ContentBinaryObject`. This value MUST be set to exactly: `application/base64`

7.3 Supported Algorithms

7.3.1 Symmetric Key Generation and Encryption

Before encrypting an envelope payload, Corner 1 MUST generate a unique and random cryptographic symmetric key of the specified algorithm and size and MUST encrypt the key using RSA-OAEP with the public key certificate published by Corner 4. The encrypted key MUST be included in the `EncryptedKey` element of the XML Encryption structure. The Algorithm attribute of the `EncryptionMethod` of the `EncryptedKey` element MUST be set to:

```
http://www.w3.org/2009/xmlenc11#rsa-oaep
```

The use of the `KeyName` element is OPTIONAL.

7.3.2 Payload Encryption

Payload encryption MUST be done using AES-256 with GCM mode in conformance with the XML Encryption Syntax and Processing Version 1.1 W3C Recommendation published here: https://www.w3.org/TR/xmlenc-core1/#sec-AES-GCM.

The payload MUST be encrypted using the symmetric key generated as specified in section 7.3.1 above.

The value of the `InstanceEncryptionMethod` element of the XHE `Payload` container containing the encrypted payload MUST be set to:

```
http://www.w3.org/2009/xmlenc11#aes256-gcm
```

7.4 Use of XML Encryption

The symmetric key and the encrypted payload MUST be packaged within an XML `EncryptedData` Encryption structure in conformance with the XML Encryption Syntax and Processing Version 1.1 W3C Recommendation as published here: https://www.w3.org/TR/xmlenc-core1.

The XML Encryption structure MUST use the following namespace:

```
http://www.w3.org/2001/04/xmlenc#
```

All XML Encryption occurrences MUST be schema valid.

7.5 Validation of Encrypted payloads

Payload encryption facilitates confidentiality between senders (Corner 1) and receivers (Corner 4) of business documents by concealing their content from intermediaries, including from Access Points (Corners 2 and 3). Consequently, the logical tasks of ensuring business document validity, such as business rule, schema, and other validations, are necessarily associated with the premises of Corners 1 and 4 rather with their respective Access Point service providers.

Notwithstanding, the application of payload encryption SHALL NOT revoke any responsibilities or obligations to send valid business documents through the DBNAlliance network, such as those defined in the network policies. Network Access Points and their end-users need to address how to ensure business document validity when sending as encrypted as the payload of an envelope.

8 Envelope signing

XHE envelopes SHOULD NOT be digitally signed. The application of a digital signature SHALL NOT give any legal and/or technical significance or meaning to the envelope.

Appendix A: Example Envelope (non- normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<XHE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.oasis-open.org/bdxr/ns/XHE/1/ExchangeHeaderEnvelope"
  xmlns:xha="http://docs.oasis-open.org/bdxr/ns/XHE/1/AggregateComponents"
  xmlns:ext="http://docs.oasis-open.org/bdxr/ns/XHE/1/ExtensionComponents"
  xmlns:xhb="http://docs.oasis-open.org/bdxr/ns/XHE/1/BasicComponents">
  <xhb:XHEVersionID>1.0</xhb:XHEVersionID>
  <xhb:CustomizationID schemeID="bdx-docid-qns"
    >http://docs.oasis-
open.org/bdxr/ns/XHE/1/ExchangeHeaderEnvelope::XHE##dbnalliance-envelope-
1.0</xhb:CustomizationID>
  <xhb:ProfileID>dbnalliance-envelope-1.0</xhb:ProfileID>
  <xha:Header>
    <xhb:ID>100001</xhb:ID>
    <xhb:CreationDateTime>2021-08-19T10:14:00Z</xhb:CreationDateTime>
    <xha:FromParty>
      <xha:PartyIdentification>
        <xhb:ID schemeID="GLN">0123456789012</xhb:ID>
      </xha:PartyIdentification>
    </xha:FromParty>
    <xha:ToParty>
      <xha:PartyIdentification>
        <xhb:ID schemeID="DUNS">123456789</xhb:ID>
      </xha:PartyIdentification>
    </xha:ToParty>
  </xha:Header>
  <xha:Payloads>
    <xha:Payload>
      <xhb:ID>1</xhb:ID>
      <xhb:Description>This is an invoice</xhb:Description>
      <xhb:ContentTypeCode listID="MIME">application/xml</xhb:ContentTypeCode>
      <xhb:CustomizationID schemeID="bdx-docid-qns"
        >urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##dbnalliance-
1.0-data-Core</xhb:CustomizationID>
      <xhb:InstanceEncryptionIndicator>false</xhb:InstanceEncryptionIndicator>
      <xha:PayloadContent>
        <Invoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="urn:oasis:names:specification:ubl:schema:xsd :Invoice-2"

xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"

xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2">
          <!-- UBL invoice, removed for brevity -->
        </Invoice>
      </xha:PayloadContent>
    </xha:Payload>
    <xha:Payload>
      <xhb:ID>2</xhb:ID>
      <xhb:Description>This is an associated PDF file sent encrypted as an attachment
to the
        invoice</xhb:Description>
      <xhb:ContentTypeCode listID="MIME">application/pdf</xhb:ContentTypeCode>
      <xhb:InstanceEncryptionIndicator>true</xhb:InstanceEncryptionIndicator>
      <xhb:InstanceEncryptionMethod>http://www.w3.org/2009/xmlenc11#aes256-
gcm</xhb:InstanceEncryptionMethod>
      <xha:PayloadContent>
        <!-- XML Encryption structure, removed for brevity -->
      </xha:PayloadContent>
    </xha:Payload>
    <xha:Payload>
```

```
      <xhb:ID>3</xhb:ID>
      <xhb:Description>This is a reference to a payload available at an external
        location</xhb:Description>
      <xhb:ContentTypeCode listID="MIME">image/svg+xml</xhb:ContentTypeCode>
      <xhb:InstanceEncryptionIndicator>false</xhb:InstanceEncryptionIndicator>
      <xha:PayloadExternalReference>
        <xhb:ID>https://businesspaymentscoalition.org/wp-content/uploads/bpc-logo-
svg.svg</xhb:ID>
      </xha:PayloadExternalReference>
    </xha:Payload>
  </xha:Payloads>
</XHE>
```