Security Markup Language (SAML) 2.0 Identifier and Protocol Profiles for Backend Attribute Exchange (BAE) v2.0

Final Version 1.0.0

January 23, 2012

Acknowledgments

The authors of this document, the Identity, Credential and Access Management (ICAM) Architecture Working Group (AWG), would like to acknowledge the work done by DHS Science & Technology Directorate and DOD DMDC West.

Table of Contents

1 Introduction

This is a SAML V2.0 profile of Identifiers and Protocol that specifies how a subject who has been issued a Credential is represented as a SAML Subject, how an assertion regarding such a subject is produced and consumed by two entities known as BAE Attribute Providers (BAE-AP).

1.1 Terminology

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]

> …they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

```
Listings of XML schemas appear like this.
```

```
Example code listings appear like this.
```

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	*XML Namespace*	*Comments*
`saml:`	`urn:oasis:names:tc:SAML:2.0:assertion`	This is the SAML V2.0 assertion namespace [SAMLCore]. This is the default namespace used throughout this document.
`samlp:`	`urn:oasis:names:tc:SAML:2.0:protocol`	This is the SAML V2.0 protocol namespace [SAMLCore].
`md:`	`urn:oasis:names:tc:SAML:2.0:metadata`	This is the SAML V2.0 metadata namespace [SAMLMeta].
`query:`	`urn:oasis:names:tc:SAML:metadata:ext:query`	This is the SAML metadata query extension namespace [SAMLMeta-Ext].

Prefix	XML Namespace	Comments
ds:	http://www.w3.org/2000/09/xmldsig#	This is the W3C XML Signature namespace, defined in the XML-Signature Syntax and Processing specification [XMLSig] and schema [XMLSig-XSD].
xenc:	http://www.w3.org/2001/04/xmlenc#	This is the W3C XML Encryption namespace, defined in the XML Encryption Syntax and Processing specification [XMLEnc] and schema [XMLEnc-XSD].
xs:	http://www.w3.org/2001/XMLSchema	This is the XML Schema namespace [Schema1].
xsi:	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

This specification uses the following typographical conventions in text: `<UnqualifiedElement>`, `<ns:QualifiedElement>`, `Attribute`, **Datatype**, `OtherKeyword`.

1.2 Normative References

[FIPS 201] *FIPS 201-1, Personal Identity Verification (PIV) of Federal Employees and Contractors*, NIST, March 2006 See http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201- 1-chng1.pdf

[FIPS 140-2] *Security Requirements for Cryptographic Modules* May 2001. See http://csrc.nist.gov/publications/fips/fips140- 2/fips1402.pdf

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. See http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January 1999. See http://www.ietf.org/rfc/rfc2246.txt

[RFC2253] M. Wahl et al. *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*. IETF RFC 2253, December 1997. See http://www.ietf.org/rfc/rfc2253.txt

[RFC3280] R. Housley et al. *Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL)* Profile. IETF RFC 3280, April 2002. See http://www.ietf.org/rfc/rfc3280.txt

[SAMLBind] S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. See http://docs.oasis- open.org/security/saml/v2.0/saml- bindings-2.0-os.pdf

[SAMLCore] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See http://docs.oasis- open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAMLMeta] S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See http://docs.oasis- open.org/security/saml/v2.0/saml- metadata-2.0-os.pdf

[SAMLMeta-Ext] T. Scavo and S. Cantor. *Metadata Extension for SAML V2.0 and V1.x Query Requesters*. OASIS Standard, November 2007. Document ID sstc-saml-metadata- ext-query-OS. See http://docs.oasis- open.org/security/saml/Post2.0/sstc-saml-metadata-ext- query-os.pdf

[SAMLProf] S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures.* World Wide Web Consortium Recommendation, May 2001. See http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[SSL3] A. Freier et al. *The SSL Protocol Version 3.0*, IETF Internet-Draft, November 1996. See http://wp.netscape.com/eng/ssl3/draft302.txt

[X509Query-XSD] *Schema for SAML V2.0 Deployment Profiles for X.509 Subjects*. OASIS, December 2006. Document ID sstc- saml-metadata-x509-query.xsd. See http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

[XMLEnc] D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web Consortium Recommendation, December 2002. See http://www.w3.org/TR/2002/REC- xmlenc-core-20021210/

[XMLEnc-XSD] *XML Encryption Schema*. World Wide Web Consortium Recommendation, December 2002. See http://www.w3.org/TR/2002/REC-xmlenc-core- 20021210/xenc-schema.xsd

[XMLSig] D. Eastlake et al. *XML- Signature Syntax and Processing*. World Wide Web Consortium Recommendation, February 2002. See http://www.w3.org/TR/2002/REC- xmldsig-core-20020212/

[XMLSig-XSD] *Schema for XML Signatures*. World Wide Web Consortium Recommendation, February 2002. See

http://www.w3.org/TR/2002/REC- xmldsig- core-20020212/xmldsig-core-schema.xsd

1.3 Non-Normative References

[BAESpecv1] *Backend Attribute Exchange Architecture and Interface Specification Version 1.0*, GSA, May 15, 2008, See http://www.smart.gov/awg/documents/BackendArchitectureInterfaceSpec.pdf

[NFIPIV] *Personal Identity Verification Interoperability for Non-Federal Issuers*, Federal CIO Council, March 2009

[PACS] *PACS IMPLEMENTATION GUIDANCE, Version. 2.3*, GSIAB, December 20, 2005 See http://www.smartcard.gov/iab/documents/PACS.pdf

[NIST800-95] *NIST 800-95, Guide to Secure Web Services,* NIST, August 2007 See http://csrc.nist.gov/publications/nistpubs/800-95/SP800- 95.pdf

[SAMLSecure] F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See http://docs.oasis-open.org/security/saml/v2.0/saml- sec- consider- 2.0-os.pdf

[SAMLGloss] J. Hodges et al. *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See http://docs.oasis- open.org/security/saml/v2.0/saml- glossary-2.0-os.pdf

[SAMLMIOP] S. Cantor et al. *SAML V2.0 Metadata Interoperability Profile Version 1.0.* OASIS Committee Draft, March 2009. See http://docs.oasis- open.org/security/saml/Post2.0/sstc-metadata-iop.html

2 SAML 2.0 Profiles of Locally Unique Identifiers (LUID) for BAE

In order to query an attribute service to retrieve the information about a Subject, it is necessary to utilize an identifier that is unique across the domain in which the Subject exists. The BAE specification uses the term **Locally Unique Identifier (LUID)** to define this identifier.

The BAE architecture has the ability to support multiple LUID formats. The *SAML 2.0 Profiles for LUIDs* in this document include:

- Federal Agency Smart Credential Number (FASC-N) from a PIV Authentication Certificate
- UUID from a PIV-I Authentication Certificate
- X.509 Subject Distinguished Name from a X.509 Certificate

It is expected that if credentials with LUID types other than what is profiled in this document are used in a BAE implementation, the Federation Operator governing that Community of Interest will define the profiles necessary for that credential type.

In SAML 2.0, **Subject Name Identifiers** (`<saml:NameID>`) are used to represent LUIDs.

2.1 Federal Agency Smart Credential Number (FASC-N) as LUID

The *SAML 2.0 Profile for FASC-N as a LUID* describes how a Subject who has been issued a PIV Card with a PIV authentication certificate is represented as a SAML V2.0 Subject. The term *PIV Authentication Certificate* as used in this specification refers to an X.509 end user certificate [RFC3280] that is resident on a PIV Card.

The FASC-N structure limits its uniqueness, and as such its use, to the U.S. Government.

2.1.1 Required Information

Identification:

```
urn:idmanagement.gov:icam:bae:v2:
SAML:2.0:nameid-format:fasc-n
```

Extends:

N/A

2.1.2 Profile Description

This deployment profile specifies a SAML V2.0 `<saml:Subject>` element that represents a principal who has been issued a PIV Authentication certificate. The principal is identified using the Federal Agency Smart Credential - Number (FASC-N). The Requester may obtain the FASC-N either by the direct authentication of the principal using a PIV Card or by the Requester having access to a trusted source of

information, such as a directory service, that contains the FASC-N information of the principal.

2.1.3 <saml:Subject> Usage

- There MUST be exactly one <saml:Subject> per <samlp:AttributeQuery>.
- The <saml:Subject> element MUST contain exactly one of <saml:NameID> or <saml:EncryptedID>.
- The BAE Requester MAY choose to encrypt the subject identifier as a means of applying confidentiality to the name identifier. In such a case, the <saml:Subject> element MUST contain a <saml:EncryptedID> element carrying the encrypted value of the <saml:NameID> element (using XML Encryption as specified in [XMLEnc]).
- The <saml:Subject> element MAY contain a <saml:SubjectConfirmation> element that includes the certificate of the Subject referenced by the <saml:NameID> element. In such a case, the following requirements MUST be satisfied:
  - The value of the <saml:SubjectConfirmation> Method attribute MUST be urn:oasis:names:tc:SAML:2.0:cm:sender-vouches
  - The <saml:SubjectConfirmationData> element MUST include a <ds:KeyInfo> element with one <ds:X509Certificate> element as its child.
  - The <ds:X509Certificate> element MUST contain the certificate of the Subject referenced by the <saml:NameID> element
- The <saml:Subject> element MAY contain additional <saml:SubjectConfirmation> elements that are out of scope for this profile.

2.1.4 <saml:NameID> Usage

If the <saml:Subject> element contains a <saml:NameID> element, the following requirements MUST be satisfied:

- The <saml:NameID> element MUST have a Format attribute whose value is urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:fasc-n
- As specified in [SAMLCore], the NameQualifier attribute of the <saml:NameID> element SHOULD be omitted.
- The value of the <saml:NameID> element MUST be the character representation of the FASC-N.
  1. The FASC-N character representation MUST be 32 characters in length and will not include character representations of the start sentinel, end sentinel, field separators and the LRC.
  2. The character representation MUST be in the order as shown in Fig 5 of the [PACS], excluding start and end sentinels, field separators and the LRC.

3. Missing values MUST be filled with zeros if the value is unknown or not set.

AC (4)	SC (4)	CN (6)	CS (1)	ICI (1)	PI (10)	OC (1)	OI (4)	POA (1)
7000	1234	000000	1	1	9000000001	1	7000	5

- AC (Agency Code)
- SC (System Code)
- CN (Credential Number)
- CS (Credential Series)
- ICI (Individual Credential Issue)
- PI (Person Identifier)
- OC (Organizational Category)
- OI (Organization Identifier)
- POA (Person/Organization Association Category)

2.1.5 <saml:EncryptedID> Usage

If the <saml:Subject> element contains a <saml:EncryptedID> element, the content of the enclosed <xenc:EncryptedData> element MUST be an encrypted <saml:NameID> element that satisfies the requirements of the previous section.

2.1.6 <saml:NameID> Example

```
<saml:Subject>
   <saml:NameID
     Format=" urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:fasc-n">
     700012340000021100000000000000000
   </saml:NameID>
</saml:Subject>
```

2.1.7 <saml:NameID> with Optional Certificate Example

```
<saml:Subject>
   <saml:NameID
     Format=" urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:fasc-n">
     700012340000021100000000000000000
   </saml:NameID>
   <SubjectConfirmation
      Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
      <saml:SubjectConfirmationData>
         <ds:KeyInfo>
           <ds:X509Data>
             <ds:X509Certificate>
                   MIICiDCCAXACCQDE+....
             </ds:X509Certificate>
           </ds:X509Data>
         </ds:KeyInfo>
      <saml:SubjectConfirmationData>
   </SubjectConfirmation>
</saml:Subject>
```

2.1.8 <saml:EncryptedID> Example

```
<saml:Subject>
   <saml:EncryptedID
     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
     <xenc:EncryptedData
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        ...
     </xenc:EncryptedData>
     <xenc:EncryptedKey
        Recipient="urn:idmanagement.gov:icam:bae:v2:7000:0000">
        ...
    </xenc:EncryptedKey>
   </saml:EncryptedID>
</saml:Subject>
```

2.2 PIV-I UUID as LUID

The *SAML 2.0 Profile for UUID as a LUID* describes how a Subject who has been issued a PIV-I Card with a Card Authentication Certificate is represented as a SAML V2.0 Subject. The term *Card Authentication Certificate* as used in this specification refers to an X.509 end user certificate [RFC3280] that is resident on a PIV-I Card.

2.2.1 Required Information

Identification:

`urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:uuid`

Extends:

N/A

2.2.2 Profile Description

This deployment profile specifies a SAML V2.0 <`saml:Subject`> element that represents a principal who has been issued a PIV-I Card Authentication certificate. The principal is identified using the [RFC4122] formatted version of the UUID per [NIST800-73]. The Requester may obtain the UUID either by the direct authentication of the principal using a PIV-I Card or by the Requester having access to a trusted source of information, such as a directory service, that contains the UUID information of the principal.

2.2.3 <`saml:Subject`> Usage

- There MUST be exactly one <`saml:Subject`> per <`samlp:AttributeQuery`>.
- The <`saml:Subject`> element MUST contain exactly one of <`saml:NameID`> or <`saml:EncryptedID`>.
- The BAE Requester MAY choose to encrypt the subject identifier as a means of applying confidentiality to the name identifier. In such a case, the <`saml:Subject`> element MUST contain a <`saml:EncryptedID`> element carrying the encrypted value of the <`saml:NameID`> element (using XML Encryption as specified in [XMLEnc]).
- The <`saml:Subject`> element MAY contain a <`saml:SubjectConfirmation`> element that includes the certificate of the Subject referenced by the <`saml:NameID`> element. In such a case, the following requirements MUST be satisfied:
    - The value of the <`saml:SubjectConfirmation`> Method attribute MUST be `urn:oasis:names:tc:SAML:2.0:cm:sender-vouches`
    - The <`saml:SubjectConfirmationData`> element MUST include a <`ds:KeyInfo`> element with one <`ds:X509Certificate`> element as its child.

    - The <ds:X509Certificate> element MUST contain the certificate of the Subject referenced by the <saml:NameID> element
- The <saml:Subject> element MAY contain additional <saml:SubjectConfirmation> elements that are out of scope for this profile.

2.2.4 <saml:NameID> Usage

If the <saml:Subject> element contains a <saml:NameID> element, the following requirements MUST be satisfied:

- The <saml:NameID> element MUST have a Format attribute whose value is urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:uuid
- As specified in [SAMLCore], the NameQualifier attribute of the <saml:NameID> element SHOULD be omitted.
- The value of the <saml:NameID> element MUST be the URN representation of the UUID found in the Card Authentication Certificate.

2.2.5 <saml:EncryptedID> Usage

If the <saml:Subject> element contains a <saml:EncryptedID> element, the content of the enclosed <xenc:EncryptedData> element MUST be an encrypted <saml:NameID> element that satisfies the requirements of the previous section.

2.2.6 <saml:NameID> Example

```
<saml:Subject>
   <saml:NameID
     Format="urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:uuid">
     urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
   </saml:NameID>
</saml:Subject>
```

2.2.7 <saml:NameID> with Optional Certificate Example

```
<saml:Subject>
   <saml:NameID
     Format="urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:uuid">
     urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
   </saml:NameID>
   <SubjectConfirmation
      Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
      <saml:SubjectConfirmationData>
         <ds:KeyInfo>
           <ds:X509Data>
             <ds:X509Certificate>
                   MIICiDCCAXACCQDE+....
             </ds:X509Certificate>
           </ds:X509Data>
         </ds:KeyInfo>
      <saml:SubjectConfirmationData>
   </SubjectConfirmation>
</saml:Subject>
```

2.2.8 <saml:EncryptedID> Example

```
<saml:Subject>
   <saml:EncryptedID
     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
     <xenc:EncryptedData
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        ...
     </xenc:EncryptedData>
     <xenc:EncryptedKey
Recipient="urn:idmanagement.gov:icam:bae:v2:052488204eb0b251cccba1d9e9672486d5
f9b045:ACME-CORP">
        ...
    </xenc:EncryptedKey>
   </saml:EncryptedID>
</saml:Subject>
```

2.3 X.509 Subject DN as LUID

The *SAML 2.0 Profile of X.509 Subject DN as LUID* describes how a service provider can represent the Subject Distinguished Name (Subject DN) field of the Subject X.509 identity certificate as a SAML 2.0 Subject.

The term *X.509 identity certificate* as used in this specification refers to an X.509 end entity certificate [RFC3280] or a certificate based on an X.509 end entity certificate (such as an X.509 proxy certificate [RFC3820]).

2.3.1 Required Information

Identification:

`urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`

Extends:

N/A

2.3.2 Profile Description

This deployment profile specifies a SAML V2.0 `<saml:Subject>` element that represents a principal who has been issued an X.509 identity certificate. The principal is identified using the Subject Distinguished Name (Subject DN). The Requester may obtain the Subject DN either by the direct authentication of the principal using an X.509 Certificate or by the Requester having access to a trusted source of information, such as a directory service, that contains the Subject DN information of the principal.

2.3.3 `<saml:Subject>` Usage

- There MUST be exactly one `<saml:Subject>` per `<samlp:AttributeQuery>`.
- The `<saml:Subject>` element MUST contain exactly one of `<saml:NameID>` or `<saml:EncryptedID>`.
- The BAE Requester MAY choose to encrypt the subject identifier as a means of applying confidentiality to the name identifier. In such a case, the `<saml:Subject>` element MUST contain a `<saml:EncryptedID>` element carrying the encrypted value of the `<saml:NameID>` element (using XML Encryption as specified in [XMLEnc]).
- The `<saml:Subject>` element MAY contain a `<saml:SubjectConfirmation>` element that includes the certificate of the Subject referenced by the `<saml:NameID>` element. In such a case, the following requirements MUST be satisfied:
  - The value of the `<saml:SubjectConfirmation>` Method attribute MUST be `urn:oasis:names:tc:SAML:2.0:cm:sender-vouches`
  - The `<saml:SubjectConfirmationData>` element MUST include a `<ds:KeyInfo>` element with one `<ds:X509Certificate>` element as its child.

    - The <ds:X509Certificate> element MUST contain the certificate of the Subject referenced by the <saml:NameID> element
- The <saml:Subject> element MAY contain additional <saml:SubjectConfirmation> elements that are out of scope for this profile.

2.3.4 <saml:NameID> Usage

If the <saml:Subject> element contains a <saml:NameID> element, the following requirements MUST be satisfied:

- The <saml:NameID> element MUST have a Format attribute whose value is urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName, as defined in section 8.3.3 of [SAMLCore]
- As specified in [SAMLCore], the NameQualifier attribute of the <saml:NameID> element SHOULD be omitted.
- The value of the <saml:NameID> element MUST have a value that is the Subject DN from the principal's X.509 identity certificate
- Subject DN from the principal's X.509 identity certificate must be as per RFC 2253. Case should not be modified.

2.3.5 <saml:EncryptedID> Usage

If the <saml:Subject> element contains a <saml:EncryptedID> element, the content of the enclosed <xenc:EncryptedData> element MUST be an encrypted <saml:NameID> element that satisfies the requirements of the previous section.

2.3.6 <saml:NameID> Example

```
<saml:Subject>
   <saml:NameID
     Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
     DN=First.Last,OU=MyBizUnit,O=MyOrg,C=US
   </saml:NameID>
</saml:Subject>
```

2.3.7 <saml:NameID> with Optional Certificate Example

```
<saml:Subject>
   <saml:NameID
     Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
     DN=First.Last,OU=MyBizUnit,O=MyOrg,C=US
   </saml:NameID>
   <SubjectConfirmation
      Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
      <saml:SubjectConfirmationData>
         <ds:KeyInfo>
           <ds:X509Data>
             <ds:X509Certificate>
                   MIICiDCCAXACCQDE+....
             </ds:X509Certificate>
           </ds:X509Data>
         </ds:KeyInfo>
      <saml:SubjectConfirmationData>
   </SubjectConfirmation>
</saml:Subject>
```

2.3.8 <saml:EncryptedID> Example

```
<saml:Subject>
   <saml:EncryptedID
     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
     <xenc:EncryptedData
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        ...
     </xenc:EncryptedData>
     <xenc:EncryptedKey
        Recipient="urn:idmanagement.gov:icam:bae:v2:7000:0000">
        ...
    </xenc:EncryptedKey>
   </saml:EncryptedID>
</saml:Subject>
```

3 SAML 2.0 Profile of Locale Identifier (LI) for BAE

The BAE Architecture supports both Direct and Brokered Attribute Exchange Models. In order to retrieve the attributes of subjects who are in remote domains, it is critical that sufficient information be made available to the Requesting BAE Broker to enable it to route the query to a BAE Broker that is authoritative for the attributes of the Subject. The BAE specification uses the term **Locale Identifier (LI)** to define the routing information that is embedded within the unique identifier assigned to a BAE Requester and/or Responder.

To enable this in an interoperable and scalable manner, it is necessary to establish a Federation Operator that is responsible for (among other items):

- Assigning a unique identifier `(entityID)` for each BAE Requester and Responder
- Managing a CA that is responsible for issuing the trust certificate, unique to each Requester and Responder, that is used for Digital Signature and Message Encryption
- Managing and distributing the metadata for the BAE environment

It is expected that there may be many Federation Operators with associated BAE Brokers that may need to interoperate, and that providing the flexibility to manage the (LI) within a Federation to the associated Federation Operator provides flexibility, scalability and manageability of the BAE eco-system.

3.1 Locale Identifier Assignment

- The Locale Identifier format and assignment MUST be centrally managed for a Federation Environment by the Federation Operator in order to assure uniqueness and prevent namespace collisions
- The Trust Certificate generation as well as the Requester/Responder unique identifier `(entityID)` assignments SHOULD be under the authority of the Federation Operator
- The CN of the Trust Certificate that is generated for a BAE Requester or BAE Responder, to be used for Digital Signature and Message Encryption, MUST be the unique identifier `(entityID)` of the BAE Broker.
- The format of the unique identifier `(entityID)` assigned to each Requester and Responder in a BAE environment MUST be the following:

  `urn:idmanagement.gov:icam:bae:v2:[LI]`

  where [LI] = Locale Identifier

- The unique identifier MUST be used as the lookup key `(entityID)` element that uniquely identifies BAE Requestors and Responders in the BAE Metadata

3.1.1 LI Assignment when using PIV Cards with FASC-N as the LUID

For interoperable Federal Government Usage across Agency and Department Boundaries, the BAE Federation Operator MUST use the following conventions to define the [LI]:

- [LI] MUST be a combination of AC and OI and has the format

  ```
  [AC]:[OI]
  ```

- In cases where the OI is not present in the PIV credentials that are issued by an Agency or Department, the value should be 0000
- In cases where both AC and OI present, the decision as to which (or both) are used for routing is left up to the discretion of the BAE Responder

  AC: Agency Code

  NIST SP800-87 Agency Code for Federal Agencies

  OI: Organizational ID

  NIST SP800-87 Agency Code for Federal Agencies

  e.g.

  ```
  urn:idmanagement.gov:icam:bae:v2:7000:0000 - for DHS
  urn:idmanagement.gov:icam:bae:v2:2100:1700 – for DOD
  urn:idmanagement.gov:icam:bae:v2:4700:4700 - for GSA
  ```

3.1.2 LI Assignment when using PIV-I Cards with UUID as the LUID

For interoperable Federal Government Usage between Federal and Non-Federal Issuers, the Federal BAE Federation Operator MUST use the following conventions to define the [LI]:

- The Federal BAE Federation Operator MUST assign the [LI] to a Non-Federal Issuer wishing to interoperate with the Federal BAE Enviornment
- [LI] MUST be a combination of AKI and ORG and has the format:

  ```
  [AKI]:[ORG]
  ```

  AKI: Authority Key Identifier
  SHA1 hash of the public key of the issuing CA and is available on the PIV-I Certificate as specified in the FBCA Cerficiate Policy for PIV-I

ORG: Organization Name (for Affiliated Users)

ORG = Affiliated Organization Name

Available as part of the Subject DN on the PIV-I Certificate, given that the FBCA policy for PIV-I requires that, for Affiliated Organizations, the Subject DN must have the format `CN=Subscriber's Full Name, OU=Affiliated Organization Name, {Base DN}`

e.g.

`urn:idmanagement.gov:icam:bae:v2:052488204eb0b251cccba1d9e9672486d5f9b045:ACME-CORP`

ORG: Organization Name (for Non-Affiliated Users)

ORG = Entity CA's Name

Use the Entity CA's Name, available from the second OU, as part of the the Subject DN which has the format `CN=Subscriber's Full Name, ou=Unaffiliated, ou=Entity CA's Name, {Base DN}`

e.g.

`urn:idmanagement.gov:icam:bae:v2:052488204eb0b251cccba1d9e9672486d5f9b045:ENTITY-CA-NAME`

3.1.3 LI Assignment when using Subject DN from X.509 Certificates as LUID

It is expected that the Federation Operator for communites that utilize the X.509 Subject DN as the LUID will define the value of the [LI] to assure uniqueness within the enviornment. The mechanisms used to assure such uniqueness are outside the scope of this profile.

If there is an expectation that a Community of Interest that has defined their own [LI] convention will need to interoperate at some point in time with a Federal Government BAE Federation environment, it is RECOMMENDED that the Community of Interest utilize the same conventions for [LI] as defined by the "LI Assignment when using PIV-I Cards with UUID as LUID" i.e. Utilize the combination of Authority Key Identifier (AKI) and Affiliated Organization Name (ORG) to define the [LI]

3.2 Locale Identifier Usage

- `<saml:Issuer>` MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format `urn:idmanagement.gov:icam:bae:v2:[LI]` where [LI] is the Locale Identifier of a BAE Requester or BAE Responder

- The `<samlp:Destination>` MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format `urn:idmanagement.gov:icam:bae:v2:[LI]` where [LI] is the Locale Identifier of a BAE Requester or BAE Responder
- The `<entityID>` attribute of the `<EntityDescriptor>` element in the BAE Metadata that represents a BAE Requester and/or Responder MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format `urn:idmanagement.gov:icam:bae:v2:[LI]` where [LI] is the Locale Identifier of a BAE Requester or BAE Responder

3.2.1 LI Example

```
<samlp:AttributeQuery
   Destination="urn:idmanagement.gov:icam:bae:v2:7000:0000"
   ID="_550fc1750"
   IssueInstant="2010-11-04T05:06:54.893-07:00"
   Version="2.0">
   <saml:Issuer>urn:idmanagement.gov:icam:bae:v2:2100:1700</saml:Issuer>
   <saml:Subject>
       <saml:NameID
          Format="urn:idmanagement.gov:icam:bae:v2:SAML:2.0:
                  nameid-format:fasc-n">
          70001234000002110000000000000000
       </saml:NameID>
   </saml:Subject>
</samlp:AttributeQuery>
```

4 SAML 2.0 Protocol Profile for BAE

In the BAE environment, when making an attribute request, the BAE Broker is acting in a "**BAE Requester**" role. When returning an attribute response, the BAE Broker is acting in a "**BAE Responder**" role. The BAE Architecture supports both a **Direct Exchange** model, in which organizations have stood up their own BAE Brokers to exchange information with external entities, as well as a **Brokered Exchange** model, in which one organization which for a variety of reasons, is utilizing the BAE Broker of another organization to share its attributes. In the direct exchange model, the **Ultimate Requester** and the **Ultimate Responder** are the same as each organization's respective BAE Brokers. In the brokered exchange model, the organization leveraging the services or another organization could be the Ultimate Requester or Ultimate Responder depending on its role.

The *SAML 2.0 Protocol Profile for Backend Attribute Exchange* specifies a profile of SAML 2.0 for the exchange of attributes between a BAE Requester and a BAE Responder. This profile relies on the *SAML 2.0 Profiles for LUID* specified in this document.

As such, a BAE Responder is a typical SAML attribute authority [SAMLGloss] and a BAE Requester is equivalent to a SAML attribute requester.

4.1 Required Information

Identification:

`urn:idmanagement.gov:icam:bae:v2:SAML:2.0:profiles:query:attribute:nameid-cleartext`

`urn:idmanagement.gov:icam:bae:v2:SAML:2.0:profiles:query:attribute:nameid-encrypted`

Extends:

Assertion Query/Request Profile [SAMLProf]

4.2 Profile Description

This deployment profile describes the use of the SAML V2.0 Assertion Query and Request Protocol [SAMLCore] in conjunction with the SAML V2.0 SOAP Binding [SAMLBind] to retrieve the attributes of a principal who has been identified via the SAML LUID Profile(s) specified in the prior sections. The attribute exchange MUST conform to the Assertion Query/Request Profile given in section 6 of [SAMLProf] unless otherwise specified below.

A BAE Requester sends a SAML V2.0 `<samlp:AttributeQuery>` message directly to a BAE Responder. This message contains a name identifier that identifies a principal. The BAE Requester MUST have previously determined unique identifier of the principal. The details of this step are out of scope for this deployment profile.

If the BAE Responder receiving the request can:

- Recognize the name identifier; and
- Fulfill the request subject to any applicable policies;

The BAE Responder responds with a successful encrypted `<samlp:Response>` containing the relevant attributes for the identified principal.

The name identifier MAY be encrypted. The BAE Broker MUST advertise, using its metadata, if it will require and/or support name identifier encryption using one or more of the Profile ID's given below:

Profile ID supporting cleartext Subject Name Identifier (LUID) in Request:

```
urn:idmanagement.gov:icam:bae:v2:
SAML:2.0:profiles:query:attribute:nameid-cleartext
```

Profile ID supporting encrypted Name Identifier (LUID) in Request:

```
urn:idmanagement.gov:icam:bae:v2:
SAML:2.0:profiles:query:attribute:nameid-encrypted
```

4.3 <samlp:AttributeQuery> Issued by BAE Requester

To initiate the profile, the BAE Requester uses a synchronous binding such as the SAML SOAP Binding [SAMLBind] to send a SAML V2.0 `<samlp:AttributeQuery>` message to an Attribute Service endpoint at a BAE Responder.

4.3.1 <samlp:AttributeQuery> Usage

The `<samlp:AttributeQuery>` element MUST conform to the following rules:

- As required by the Assertion Query/Request Profile [SAMLProf], the `<samlp:AttributeQuery>` element MUST contain a `<saml:Issuer>` element.
  - The `<saml:Issuer>` element MUST be the unique identifier of the Ultimate Requester issued by the BAE Federation Operator.
  - `<saml:Issuer>` MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format `urn:idmanagement.gov:icam:bae:v2:[LI]` where [LI] is the Locale Identifier of the Ultimate Requester
- A `<samlp:AttributeQuery>` MAY include the `<samlp:Consent>` attribute to communicate to the BAE-AP whether a Subject gave consent for attribute release, and under what conditions.
  - Possible values for the Consent Identifiers can be found in Section 8.4 of [SAMLCore]
  - The Federation Operator MAY define additional values for Consent Identifiers.

- A <samlp:AttributeQuery> MUST include the <samlp:Destination> attribute
  - The <samlp:Destination> element MUST be the unique identifier of the Ultimate Responder issued by the BAE Federation Operator
  - <samlp:Destination> MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format urn:idmanagement.gov:icam:bae:v2:[LI] where [LI] is the Locale Identifier of the Ultimate Responder
- The <saml:Subject> element MUST conform to one of the *SAML 2.0 LUID Profile(s)* defined in this document
- The <samlp:AttributeQuery> element MAY include one or more <saml:Attribute> elements
- The <samlp:AttributeQuery> element MUST contain a <ds:Signature> element carrying the signature of the BAE Requester

4.3.2 <saml:Attribute> Usage

- A <saml:Attribute> element MAY have a NameFormat attribute.
  - If present, NameFormat MUST be set to one of the following values:
    urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified
    urn:oasis:names:tc:SAML:2.0:attrname-format:uri
    urn:oasis:names:tc:SAML:2.0:attrname-format:basic
- Each <saml:Attribute> MAY contain the FriendlyName attribute.
- A <saml:Attribute> MAY contain multiple <saml:AttributeValue>s.
  - If multiple <saml:AttributeValue>s exist, a BAE Responder MUST choose one of the values presented in the request.

4.3.3 Use of SSL/TLS

All requests MUST be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain transport level confidentiality and message integrity. In addition, the requester MAY use SSL/TLS server authentication, if that is a BAE Federation Operator requirement i.e. All BAE Brokers MUST use the same transport level security implementation within a specific Community of Interest.

4.3.4 Use of Encryption

The SAML V2.0 Assertions and Protocols specification [SAMLCore] defines the <saml:EncryptedID> element as a means of applying confidentiality to the name identifier.

The Ultimate Requester MAY use the <saml:EncryptedID> to carry the subject identifier of the principal in the <samlp:AttributeQuery> element.

If encryption is used, exactly one of the following procedures MUST be followed:

- The Ultimate Requester generates a new symmetric key to encrypt the principal's name identifier. After performing the encryption, the Ultimate Requester places the resulting ciphertext in the <xenc:EncryptedData> element. The symmetric key MUST be encrypted with the Ultimate Responder's public key and the resulting ciphertext placed in the <xenc:EncryptedKey> element.
- The Ultimate Requester uses a previously established symmetric key to encrypt the principal's name identifier. After performing the encryption, the Ultimate Requester places the resulting ciphertext in the <xenc:EncryptedData> element. In this case, the <saml:EncryptedID> element MUST NOT contain an <xenc:EncryptedKey> element.

A symmetric key transmitted in an <xenc:EncryptedKey> element MUST NOT be later reused by the Requester as a previously established symmetric key.

An encryption algorithm satisfying FIPS 140-2 Security Requirements [FIPS 140-2] SHALL be used for all encryption operations.

4.3.5 Use of Digital Signature

The SAML V2.0 Assertions and Protocols specification [SAMLCore] describes how to use the <ds:Signature> element (defined in [XMLSig]) as a means of providing integrity and authenticity for a message.

A BAE Requester MUST sign the <samlp:AttributeQuery> element to allow the BAE Responder to authenticate the origin and verify the integrity of the request.

- In the Direct Attribute Exchange Model, the Ultimate Requester and the BAE Broker acting in the Requester role are the same, as such the BAE Broker signs the <samlp:AttributeQuery>
- In the Brokered Attribute Exchange Model, the Ultimate Requester signs the <samlp:AttributeQuery>

A BAE Broker in the Requester role MUST authenticate itself to the BAE Broker in the Responder role by signing the request using a WS-Security Digital Signature that covers both the <soap:Body> and the <wsu:Timestamp> element in the SOAP header. It MUST also use the signing certificate that has been issued to it by the Federation Operator that manages the BAE environment. (e.g. BAE CA in the Federal Government)

The following additional and/or clarifying rules apply to digital signatures:

- The SignatureMethod MUST be compliant to FIPS 186
- The DigestMethod MUST be compliant to FIPS 180-2
- The CanonicalizationMethod MUST be Exclusive Canonicalization
- The Enveloped Signature method MUST be used to sign the <samlp:AttributeQuery>

- The SOAP security header MUST contain a timestamp (i.e. <wsu:Timestamp>), as defined in the WS-Security specifications with both a <wsu:Created> element and optionally a <wsu:Expires> element.
- The Detached Signature method MUST be used and must cover both the <soap:Body> and the <wsu:Timestamp> elements.

4.4 <samlp:Response> Issued by BAE Responder

The BAE Responder MUST process the request as outlined in [SAMLCore]. After processing the message or upon encountering an error, the BAE Responder MUST return a <samlp:Response> message containing an appropriate status code to the BAE Requester to complete the SAML protocol exchange.

If the BAE Responder is successful in locating one or more attributes for this principal, they will be included in the response. If the BAE Responder is not able to map the <saml:Subject> element to a local principal, it MUST return an error.

If no <saml:Attribute> elements are included in the query (i.e. an empty query), the BAE Responder returns all attributes for this principal, subject to policy. If the BAE Responder is unable to resolve attributes for this principal (for any reason), it MUST return an error.

4.4.1 <samlp:Response> Usage

If the request is successful, the <samlp:Response> element MUST conform to the following rules:

<samlp:Response> MUST include the <samlp:Destination> attribute

- The <samlp:Destination> element MUST be the unique identifier of the Ultimate Requester issued by the Federation Operator and this SHOULD be the same as the <saml:Issuer> element in the <samlp:AttributeQuery>.
- <samlp:Destination> MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format urn:idmanagement.gov:icam:bae:v2:[LI] where [LI] is the Locale Identifier of the Ultimate Requester

It MUST contain at least one <saml:EncryptedAssertion> element (but no <saml:Assertion> elements).

The encrypted content of each <saml:EncryptedAssertion> element is a <saml:Assertion> element that MUST satisfy the following conditions:

- The <saml:Assertion> element MUST contain a <ds:signature> element carrying the signature of the Ultimate Responder.
- The <saml:Subject> element (which strongly matches the subject of the query [SAMLCore]) SHOULD NOT contain a <saml:SubjectConfirmation> element.

- The <saml:Assertion> element MUST contain a <saml:Conditions> element with NotBefore and NotOnOrAfter attributes.
- The <saml:Assertion> element MUST contain a <saml:Issuer>
  1. The <saml:Issuer> element MUST be the unique identifier of the Ultimate Responder issued by the Federation Operator.
  2. <saml:Issuer> MUST be a Uniform Resource Identifier reference within the Federation Operator domain of the format urn:idmanagement.gov:icam:bae:v2:[LI] where [LI] is the Locale Identifier of the Ultimate Responder.
- The <saml:Assertion> element SHOULD contain a <saml:Audience> element whose value is identical to the value of the <saml:Issuer> element in the request.
- Other conditions (including other <saml:Audience> elements) MAY be included as required by the BAE Requester or at the discretion of the BAE Responder.
- The <saml:Assertion> element MUST contain zero or one <saml:AttributeStatement>s
  1. Each <saml:AttributeStatement> MUST contain one or more <saml:Attribute>s, which MAY contain any number of <saml:AttributeValue>s
  2. The <saml:AttributeStatement> MUST use <saml:Attribute> and MUST NOT use <saml:EncryptedAttribute>
  3. The use of URI-formatted attribute names from well known registries is RECOMMENDED
  4. The BAE compliant Attribute Service MUST NOT send attributes that are not requested by the Relying Party
  5. Relying Parties SHOULD NOT accept <saml:Assertion>s containing attributes that have not been negotiated out of band or via metadata

If the BAE Responder wishes to return an error, it MUST NOT include any encrypted assertions in the <samlp:Response> message.

Possible error responses include the following:

- The BAE Responder MAY return one of the status codes urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile or urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue as suggested in section 3.3.2.3 of [SAMLCore].
- If the BAE Responder does not recognize the <saml:NameID> element or otherwise is unable to map the <saml:NameID> element to a local principal

name, it MAY return the following status code:
`urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal`

4.4.2 Use of SSL/TLS

All responses MUST be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain transport level confidentiality and message integrity. In addition, the responder MAY use SSL/TLS server authentication, if that is a BAE Federation Operator requirement i.e. All BAE Brokers MUST use the same transport level security implementation within a specific Community of Interest.

4.4.3 Use of Encryption

The SAML V2.0 Assertions and Protocols specification [SAMLCore] defines the `<saml:EncryptedAssertion>` element as a means of applying confidentiality to the contents of an assertion.

The Ultimate Responder MUST use the `<saml:EncryptedAssertion>` element to carry the returned attribute values for the Principal.

To encrypt the `<saml:Assertion>` element, exactly one of the following procedures MUST be followed:

- The Ultimate Responder generates a new symmetric key to encrypt the `<saml:Assertion>` element. After performing the encryption, the Ultimate Responder places the resulting ciphertext in the `<xenc:EncryptedData>` element. The symmetric key MUST be encrypted with the Ultimate Requester's public key and the resulting ciphertext placed in the `<xenc:EncryptedKey>` element.
- The Ultimate Responder uses a symmetric key previously established with the Ultimate Requester to encrypt the `<saml:Assertion>` element. After encrypting the `<saml:Assertion>` element using this key, the Ultimate Responder places the resulting ciphertext in the `<xenc:EncryptedData>` element. In this case, however, the `<saml:EncryptedAssertion>` element MUST NOT contain an `<xenc:EncryptedKey>` element.
- If the Ultimate Requester did not include a symmetric key in the `<samlp:AttributeQuery>` for decryption of the `<saml:EncryptedID>`, the Ultimate Responder uses a previously established symmetric key to encrypt the `<saml:Assertion>`. If the Ultimate Responder reuses a key in this manner, the `<saml:EncryptedAssertion>` element MUST NOT contain an `<xenc:EncryptedKey>`element.

An encryption algorithm satisfying FIPS 140-2 Security Requirements [FIPS 140- 2] SHALL be used for all encryption operations.

The unique identifier of the Ultimate Requester as provided in the `<saml:Issuer>` element of the `<samlp:AttributeQuery>` or the information found in the digital certificate used for signing the `<samlp:AttributeQuery>` SHOULD be used to

select the public key that the Ultimate Responder uses to encrypt the response. The mechanism by which this lookup is accomplished is out of scope for this deployment profile.

4.4.4 Use of Digital Signatures

The SAML V2.0 Assertions and Protocols specification [SAMLCore] describes how to use the `<ds:Signature>` element (defined in [XMLSig]) as a means of providing integrity and authenticity for a message.

- The recipient MUST authenticate the sender by verifying the signature upon receipt of the message.
- Signature verification MUST use the public key in the sender's BAE certificate.
- The recipient MUST verify the revocation status of the sender BAE certificate used to sign the message. The recipient SHOULD use one of the following methods for revocation verification:
    1. CDP Extension – the signature certificate will include a Certificate Revocation List (CRL) Distribution Point (CDP) extension point.
    2. OCSP – The OCSP URI is available via the AuthorityInformationAccess extension.
    3. CRL – the CRL location (in the directory or web site) can be statically configured into the software, and CRL downloaded periodically.

The `<saml:Assertion>` element in the response MUST be signed before the encryption operation takes place to ensure integrity.

- In the Direct Attribute Exchange Model, the Ultimate Responder and the BAE Broker acting in the Responder role are the same; as such the BAE Broker signs the `<saml:Assertion>`
- In the Brokered Attribute Exchange Model, the Ultimate Responder signs the `<saml:Assertion>`

A BAE Broker in the Responder role MUST authenticate itself to the BAE Broker in the Requester role by signing the response using a WS-Security Digital Signature that covers both the `<soap:Body>` and the `<wsu:Timestamp>` element in the SOAP header. It MUST also use the signing certificate that has been issued to it by the Federation Operator that manages the BAE environment. (e.g. BAE CA in the Federal Government)

The following additional and/or clarifying rules apply to digital signatures:

- The SignatureMethod MUST be compliant to FIPS 186
- The DigestMethod MUST be compliant to FIPS 180-2
- The CanonicalizationMethod MUST be Exclusive Canonicalization
- The Enveloped Signature method MUST be used to sign the `<samlp:AttributeQuery>`

- The SOAP security header MUST contain a timestamp (i.e. <wsu:Timestamp>), as defined in the WS-Security specifications with both a <wsu:Created> element and optionally a <wsu:Expires> element.
- The Detached Signature method MUST be used and must cover both the <soap:Body> and the <wsu:Timestamp> elements.

4.4.5 <samlp:AttributeQuery> Example

```
<samlp:AttributeQuery
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="aaf23196-1773-2113-474a-fe114412ab72"
  Destination="urn:idmanagement.gov:icam:bae:v2:1:7000:0000"
  Version="2.0"
  IssueInstant="2006-07-17T22:26:40Z">
  <saml:Issuer>urn:idmanagement.gov:icam:bae:v2:1:2100:1700</saml:Issuer>
  <saml:Subject>
    <saml:NameID
      Format="urn:idmanagement.gov:icam:bae:v2:SAML:2.0:nameid-format:fasc-n">
          700012340000021100000000000000000
    </saml:NameID>
  </saml:Subject>

  <saml:Attribute Name="nc:PersonGivenName"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
  </saml:Attribute>

  <saml:Attribute Name="nc:PersonMiddleName"
       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
  </saml:Attribute>

  <saml:Attribute Name="nc:PersonSurName"
       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
  </saml:Attribute>

</samlp:AttributeQuery>
```

4.4.6 <samlp:Response> Example

```
<samlp:Response
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
  ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac"
  Destination="urn:idmanagement.gov:icam:bae:v2:1:2100:1700"
  Version="2.0"
  IssueInstant="2006-07-17T22:26:41Z">
  <saml:Issuer>urn:idmanagement.gov:icam:bae:v2:1:7000:0000</saml:Issuer>
  <samlp:Status>
    <samlp:StatusCode
        Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <saml:Assertion
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       ID="a144e8f3-adad-594a-9649-924517abe933"
       Version="2.0"
       IssueInstant="2006-07-17T22:26:41Z">
  <saml:Issuer>urn:idmanagement.gov:icam:bae:v2:1:7000:0000</saml:Issuer>
    <saml:Subject>
      <saml:NameID
         Format="urn:idmanagement.gov:icam:bae:v2:
SAML:2.0:nameid-format:fasc-n">
            700012340000021100000000000000000
      </saml:NameID>
</saml:Subject>
<saml:Conditions
      NotBefore="2006-07-17T22:21:41Z"
      NotOnOrAfter="2006-07-17T22:51:41Z">
      <saml:AudienceRestriction>
      <saml:Audience>
      urn:idmanagement.gov:icam:bae:v2:1:2100:1700
      </saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
 <saml:AttributeStatement>

<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
     Name="nc:PersonGivenName">
     <saml:AttributeValue>James</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
      Name="nc: PersonMiddleName">
      <saml:AttributeValue>Tiberius</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
      Name="nc: PersonSurName">
      <saml:AttributeValue>Kirk</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>
```

5 Security and Privacy Considerations

The motivation for this deployment profile is to specify a secure means of obtaining SAML attributes by a relying party in possession of a valid subject identifier.

5.1 Background

The SAML Security and Privacy specification [SAMLSecure] provides general background material relevant to all SAML bindings and profiles. Section 6.1 of [SAMLSecure], in particular, considers the security requirements of the SAML SOAP binding, and is therefore pertinent to this deployment profile. In addition, section 3.1.2 of the SAML Bindings specification [SAMLBind] provides further security guidelines regarding SAML bindings.

5.2 General Security Requirements

In this deployment profile, the system entity that performs the authentication, or has access to a trusted source containing Subject information, must have a secure means of obtaining the subject identifier of the subject. This system entity must be securely linked to the BAE Requester that subsequently initiates this deployment profile by issuing a SAML V2.0 `<samlp:AttributeQuery>` for that subject to the appropriate asserting party. The mechanism by which these system entities are linked is out of scope for this deployment profile.

Local policy settings at the BAE Responder will determine whether or not the asserting party is permitted to return attributes for the requested subject.

As noted in [NIST800-95] Section 3.5.3.6

> "In particular, it is important to recognize that once a SAML assertion has been issued, it is not possible to control its dissemination. An entity that receives a SAML assertion may pass it on to other, potentially malicious entities as part of the system [..] Because of this, it is possible that a malicious entity may attempt to use SAML assertions in replay attacks (in particular, authentication assertions and authorization decision assertions are likely to be replayed). There are a number of techniques that can mitigate this threat, including:
>
> - Encrypting the assertion will prevent a third party from viewing it, although a malicious entity may attempt to resend the encrypted assertion.
> - Signing the entire message rather than the assertion itself, using WS-Security in a SOAP response or SSL/TLS in a HTTP response. This way, an attacker must resend the whole message to be successful.
> - Enforcing validity periods and ensuring that the IssueInstant of the assertion is reasonable. This will minimize the amount of time during which an attacker may successfully execute a replay attack."

5.3 Metadata Security

As noted in [SAMLMIOP]:

> “Metadata becomes a critical tool for the revocation of compromised sites and keys, and all of the standard practices in the use of tools like CRLs become relevant to the consumption of metadata. The specification has the mechanisms to address these issues, but they have to be used. Specifically, metadata obtained via an insecure transport should be both signed, and should expire; so that consumers are forced to refresh it often enough to limit the damage from compromised information. Either the validUntil or cacheDuration attribute may be appropriate to mitigate this threat, depending on the exchange mechanism.
>
> In addition, distributing signed metadata without expiration over an untrusted channel (e.g., posting it on a public web site) creates an exposure. An attacker can corrupt the channel and substitute an old metadata file containing a compromised key and proceed to use that key together with other attacks to impersonate a site. Repeatedly expiring (using a validUntil attribute) and reissuing the metadata limits the window of exposure, just as a CRL does. Note that the cacheDuration attribute does not prevent this attack"

5.4 BAE Responder Policy

BAE Requesters may explicitly enumerate the required attributes in queries or may issue so-called “empty queries” that essentially request all available attributes. Regardless of the attribute requirements called out in the query (or in metadata, if used for this purpose), it is the BAE Responder that determines the actual attributes returned to the BAE Requester. Thus a responsible BAE Responder will initiate and enforce policy that strictly limits the attributes released to BAE Requesters.

5.5 Caching of Attributes

A BAE Requester will most likely provide a capability to cache user attributes returned in assertions. If so, cache expiration settings should be configurable by administrators.

5.6 User Privacy

The identity of the principal for which the assertion was issued SHOULD NOT be human readable (that is, stored in clear text) in log files, cache files or the cache repository (as applicable).

6 Implementation Guidance (Informative)

The following non-normative guidelines are provided for the convenience of implementers.

6.1 Credential to Locale Identifier Mapping Guidelines

In order for a relying party to compose an attribute query to retrieve the attributes of a Subject from a BAE Broker, it needs the following pieces of information:

- A Locally Unique Identifier (LUID) of the Subject retrieved from a credential or a Trusted Source that is conformant to the *SAML 2.0 Profiles for LUIDs* described in this document
- A Locale Identifier (LI) for the BAE Broker to which the query should be routed to that is conformant to the *SAML 2.0 Profiles for LIs* described in this document

The following guidance specifies how, for credential types that are presented to a relying party AND are relevant to Federal Government use cases, the LI can be derived from information found in the credential.

It is expected that if credentials with LUID types other than what is profiled in this document are used in a BAE implementation, the Federation Operator governing the environment will define the rules for credential to LI mapping specific to that credential type.

6.1.1 PIV Card Information to LI Mapping

HSPD12 requires the use of a common, interoperable credential across the federal government called the PIV Card. The unique identifier present in a PIV Card is the FASC-N, which is the LUID that is supported by the BAE specification for Federal use.

When managed and distributed within a closed system (such as the U.S. Federal Government), the uniqueness of the FASC-N as a subject identifier is ensured. Within the Federal Government, it is noted that in the FASC-N the Agency Code (AC), System Code (SC), Credential Number (CN), Credential Series (CS) and Individual Credential Issue (ICI) are defined exactly as in the SEIWG-012 credential number.

It is acknowledged that in some cases, the Person Identifier (PI), Organizational Category (OC), Organizational Identifier (OI) and Person/Organization Association Category (POA) are not defined which limits the utility of the FASC-N to simply identifying the credential and not a person (as is needed to support a Logical Access Control scenario).

In cases where the PI, OC, OI and POA are not defined as part of the FASC-N, the AC may be used by a relying party to identify the credential issuer and to route the request to the issuer's BAE Responder, which can then use the information in the FASC-N to map the Subject to a principal in its security domain. How this mapping is accomplished is out of scope for this deployment profile.

As noted in the *SAML 2.0 Profile for LI* section of this document, the `entityID` that describes an organization MUST have the following explicit naming convention:

```
urn:idmanagement.gov:icam:bae:v2:[LI]
```

Further, for interoperable Federal Government Usage across Agency and Department Boundaries, the BAE Federation Operator will use the following conventions to define the [LI]:

- [LI] MUST be a combination of AC and OI and has the format

  ```
  [AC]:[OI]
  ```

- In cases where the OI is not present in the PIV credentials that are issued by an Agency or Department, the value should be 0000
- In cases where both AC and OI present, the decision as to which (or both) are used for routing is left up to the discretion of the BAE Responder

  AC: Agency Code

  NIST SP800-87 Agency Code for Federal Agencies

  OI: Organizational ID

  NIST SP800-87 Agency Code for Federal Agencies

  e.g.

  ```
  urn:idmanagement.gov:icam:bae:v2:7000:0000 - for DHS
  urn:idmanagement.gov:icam:bae:v2:2100:1700 – for DOD
  urn:idmanagement.gov:icam:bae:v2:4700:4700 - for GSA
  ```

The above provides the ability to map the AC and OI elements obtained by the Relying Party from the FASC-N in the PIV Authentication Certificate to the unique identifier (`entityID`) assigned to the Subject's BAE Broker. Using this `entityID` (which contains the LI) as the lookup key in the BAE Metadata, the relying party is able to route the request to the appropriate BAE broker.

The above also provides a binding between the Unique Identifier for a BAE (the `entityID`) and the public key of the signing/encryption certificate generated for the BAE and distributed via the BAE metadata (Given that the CN of the certificate contains the `entityID` as its value). This allows a Responder the option to parse the certificate used to sign the attribute query for the `entityID` of the attribute requester (Ultimate Requester), which in turn can be used to:

- Verify with a high degree of assurance, by comparing the `entityID` parsed from the certificate to the `entityID` found in the `<saml:Issuer>` element in the request, the identity of the Ultimate Requester.

- Dynamically look up, in the BAE Metadata, the public key of the Ultimate Requester that MUST be used to encrypt the response.

6.1.2 PIV-I Card Information to LI Mapping

Non-Federal Issuers (NFIs) of identity cards have expressed a desire to produce identity cards that can technically interoperate with Federal government Personal Identity Verification (PIV) systems and can be trusted by Federal government Relying Parties. In response to this, the Federal government's Federal CIO Council released the Personal Identity Verification Interoperability for Non-Federal Issuers guidance in May 2009, which provides information for entities wishing to implement an identity card that is technically interoperable with a Federally-issued PIV card and can be trusted by Federal relying parties. Subsequently, a revised X.509 Certificate Policy for the Federal Bridge Certification Authority (FBCA) has been published that comprehensively addresses PIV-I.

As noted in the *SAML 2.0 Profile for LI* section of this document, the `entityID` that describes an organization MUST have the following explicit naming convention:

```
urn:idmanagement.gov:icam:bae:v2:[LI]
```

Further, for interoperable Federal Government Usage between Federal and Non-Federal Issuers, the Federal BAE Federation Operator will use the following conventions to define the [LI]:

- The Federal BAE Federation Operator MUST assign the [LI] to a Non-Federal Issuer wishing to interoperate with the Federal BAE Enviornment
- [LI] MUST be a combination of AKI and ORG and has the format:

  ```
  [AKI]:[ORG]
  ```

  AKI: Authority Key Identifier
  SHA1 hash of the public key of the issuing CA and is available on the PIV-I Certificate as specified in the FBCA Cerficiate Policy for PIV-I

  ORG: Organization Name (for Affiliated Users)

  ORG = Affiliated Organization Name

  Available as part of the Subject DN on the PIV-I Certificate, given that the FBCA policy for PIV-I requires that, for Affiliated Organizations, the Subject DN must have the format `CN=Subscriber's Full Name, OU=Affiliated Organization Name, {Base DN}`

  e.g.

  ```
  urn:idmanagement.gov:icam:bae:v2:052488204eb0b251cccba1d9e9672486d5f9b045:ACME-CORP
  ```

ORG: Organization Name (for Non-Affiliated Users)

ORG = Entity CA's Name

Use the Entity CA's Name, available from the second OU, as part of the the Subject DN which has the format `CN=Subscriber's Full Name, ou=Unaffiliated, ou=Entity CA's Name, {Base DN}`

e.g.

`urn:idmanagement.gov:icam:bae:v2:052488204eb0b251cccba1d9e9672486d5f9b045:ENTITY-CA-NAME`

The above provides the ability to map the combination of Authority Key Identifier and Organization Name found on the Certificate, and which provides uniqueness within a Federation, to the unique identifier (`entityID`) assigned to the Subject's BAE Broker by the Federal BAE Federation Operator. Using this `entityID` (which contains the LI) as the lookup key in the BAE Metadata, the relying party is able to route the request to the appropriate BAE broker.

The above also provides a binding between the Unique Identifier for a BAE (the `entityID`) and the public key of the signing/encryption certificate generated for the BAE and distributed via the BAE metadata (Given that the CN of the certificate contains the `entityID` as its value). This allows a Responder the option to parse the certificate used to sign the attribute query for the `entityID` of the attribute requester (Ultimate Requester), which in turn can be used to:

- Verify with a high degree of assurance, by comparing the `entityID` parsed from the certificate to the `entityID` found in the <`saml:Issuer`> element in the request, the identity of the Ultimate Requester.

Dynamically look up, in the BAE Metadata, the public key of the Ultimate Requester that MUST be used to encrypt the response.

6.1.3 X.509 Certificate Information to LI Mapping

There are Communities of Interest within the Government as well as Federation environments outside the Government that currently currently utilize X.509 Certificates or would prefer to use them internally within their organization.

As noted in the *SAML 2.0 Profile for LI* section of this document, the `entityID` that describes an organization MUST have the following explicit naming convention:

`urn:idmanagement.gov:icam:bae:v2:[LI]`

It is expected that the Federation Operator for these communities of interest will define the value of the [LI] to assure uniqueness within the enviornment. The mechanisms used to assure such uniqueness are outside the scope of this profile.

If there is an expectation that a Community of Interest that has defined their own [LI] convention will need to interoperate at some point in time with a Federal Government BAE Federation environment, it is RECOMMENDED that the Community of Interest utilize the same conventions for [LI] as defined by the “PIV-I Card Information to LI Mapping” i.e. Utilize the combination of Authority Key Identifier (AKI) and Organization Name (ORG) to define the [LI]

The above provides the ability to map the LI to the unique identifier (`entityID`) assigned to the Subject’s BAE Broker by the BAE Federation Operator. Using this `entityID` (which contains the LI) as the lookup key in the BAE Metadata, the relying party is able to route the request to the appropriate BAE broker.

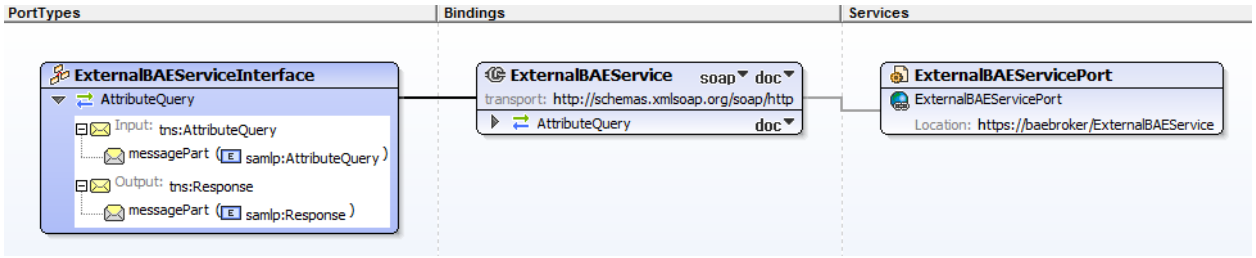
The above also provides a binding between the Unique Identifier for a BAE (the `entityID`) and the public key of the signing/encryption certificate generated for the BAE and distributed via the BAE metadata (Given that the CN of the certificate contains the `entityID` as its value). This allows a Responder the option to parse the certificate used to sign the attribute query for the `entityID` of the attribute requester (Ultimate Requester), which in turn can be used to:

- Verify with a high degree of assurance, by comparing the `entityID` parsed from the certificate to the `entityID` found in the `<saml:Issuer>` element in the request, the identity of the Ultimate Requester.

Dynamically look up, in the BAE Metadata, the public key of the Ultimate Requester that MUST be used to encrypt the response.

6.2 Web Service WSDL for SAML 2.0 Profile of BAE

The implementation of the [BAESpecv1] is a SOAP based web service. The “Single PIV Cardholder BAE Model” as defined in the [BAESpecv1], and profiled in this document, is implemented by the “AttributeQuery” operation in the following WSDL, which is provided as a convenience to implementers.




```
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:idmanagement.gov:icam:bae:v2:externalbaeservice"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:ns="urn:idmanagement.gov:icam:bae:v2:externalbaeservice"
    name="ExternalBAEService" targetNamespace="urn:idmanagement.gov:icam:bae:v2:externalbaeservice">
    <types>
        <xsd:schema>
            <xsd:import schemaLocation="saml-schema-protocol-2.0.xsd" namespace="urn:oasis:names:tc:SAML:2.0:protocol"/>
        </xsd:schema>
    </types>
    <message name="AttributeQuery">
        <part name="messagePart" element="samlp:AttributeQuery"/>
    </message>
    <message name="Response">
        <part name="messagePart" element="samlp:Response"/>
    </message>
    <portType name="ExternalBAEServiceInterface">
        <operation name="AttributeQuery">
            <input message="tns:AttributeQuery"/>
            <output message="tns:Response"/>
        </operation>
    </portType>
    <binding name="ExternalBAEService" type="tns:ExternalBAEServiceInterface">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="AttributeQuery">
            <soap:operation soapAction="AttributeQuery" style="document"/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
    <service name="ExternalBAEServicePort">
        <port name="ExternalBAEServicePort" binding="tns:ExternalBAEService">
            <soap:address location="https://FICAM_BAE_v2/ExternalBAEService"/>
        </port>
    </service>
</definitions>
```

```
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:idmanagement.gov:icam:bae:v2:externalbaeservice"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:ns="urn:us:gov:bae:v2:service:external"
name="ExternalBAEService"
targetNamespace="urn:idmanagement.gov:icam:bae:v2:externalbaeservice">
<types>
    <xsd:schema>
      <xsd:import schemaLocation="saml-schema-protocol-2.0.xsd"
         namespace="urn:oasis:names:tc:SAML:2.0:protocol"/>
      </xsd:schema>
</types>
<message name="AttributeQuery">
  <part name="messagePart" element="samlp:AttributeQuery"/>
</message>
<message name="Response">
  <part name="messagePart" element="samlp:Response"/>
</message>
<portType name="ExternalBAEServiceInterface">
  <operation name="AttributeQuery">
    <input message="tns:AttributeQuery"/>
    <output message="tns:Response"/>
  </operation>
</portType>
<binding name="ExternalBAEService"
  type="tns:ExternalBAEServiceInterface">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="AttributeQuery">
      <soap:operation soapAction="AttributeQuery" style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
</binding>
<service name="ExternalBAEServicePort">
  <port name="ExternalBAEServicePort"
    binding="tns:ExternalBAEService">
    <soap:address location="https://baebroker/ExternalBAEService"/>
  </port>
</service>
</definitions>
```

6.3 Attribute Exchange Pattern Implementation for BAE

As noted in the BAE Overview Document, the Federal ICAM Backend Attribute Exchange Implements the following design patterns:

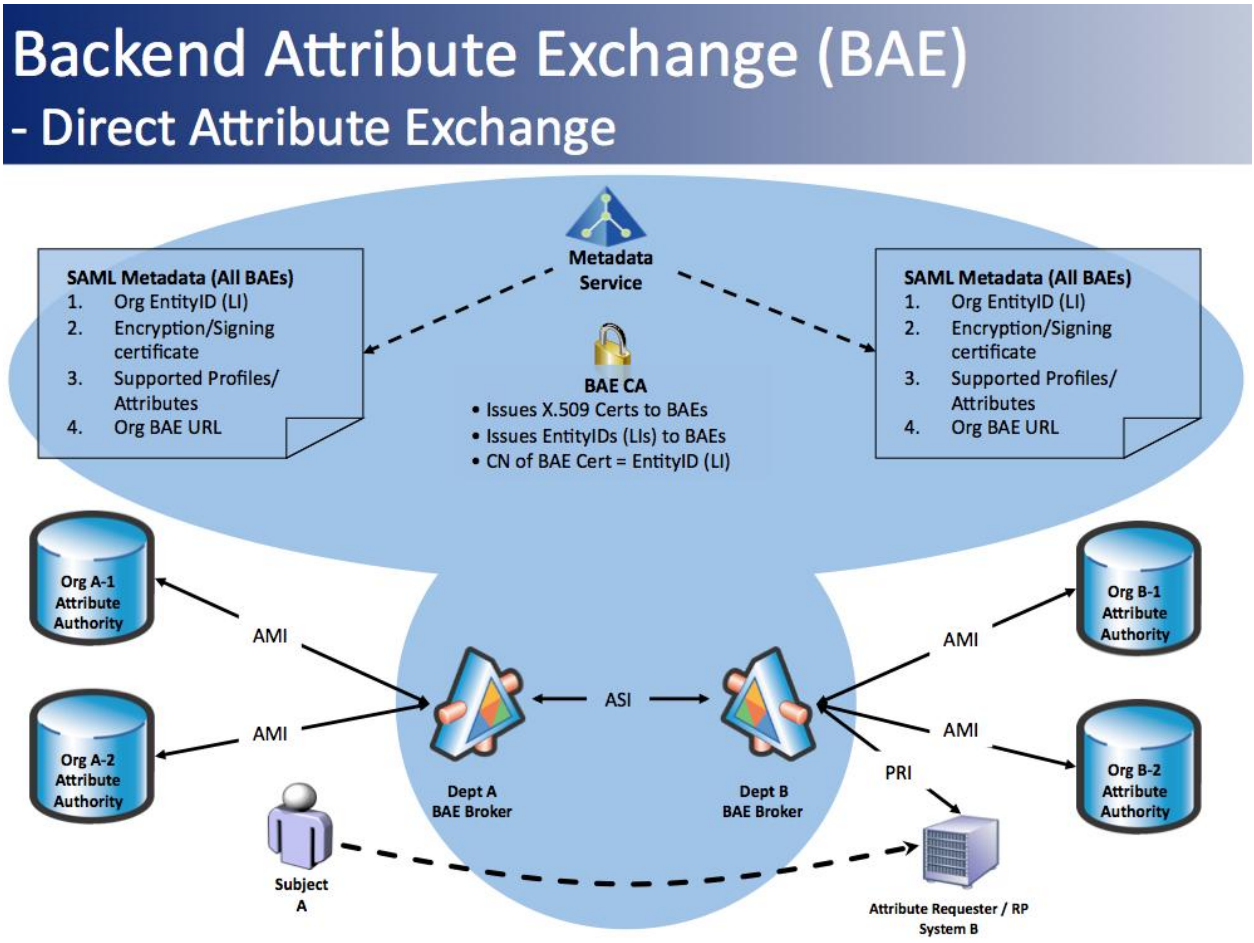
- The **BAE Direct Attribute Exchange Model** is an implementation of the "Organizational Query Design Pattern"
- The **BAE Brokered Attribute Exchange Mode**l is an implementation of the "Single Point of Query Design Pattern"

In addition, the following also holds true regarding the BAE:

- The "External BAE Service" in the BAE Architecture is an implementation of the Attribute Service Interface (ASI) described in the patterns. The technical profiles regarding the implementation of this interface are described earlier in this document.
- The "Internal BAE Service" in the BAE Architecture is an implementation of the Protected Resource Interface (PRI). This interface implementation is left up to the discretion of agency implementations.

The implementation flow given below is informative in nature and is not meant to be the only way to implement a BAE Direct and Brokered exchange models.

6.3.1 Implementation – BAE Direct Attribute Exchange (Informative)



Preconditions

- Locale Identifier (LI) format = urn:bae:OC:OI

  OC: Organizational Category

  1. Federal Government Agency
  2. State Government Agency
  3. Commercial Enterprise
  4. Foreign Government

  OI: Organizational ID

  NIST SP800-87 Agency Code for Federal Agencies

  e.g:

  urn:bae:1:7000 - for DHS

urn:bae:1:9700 - for DOD

- Department A LI = “urn:bae:1:a” (Responder = Ultimate Responder)
  Department B LI = “urn:bae:1:b” (Requester = Ultimate Requester)
- Credential is a PIV Card
- Subject Name Identifier (LUID) is a FASC-N

The above provides a mapping between the OC and OI elements obtained by the Relying Party from the FASC-N and the entityID (LI) assigned to the BAE Broker who is the authoritative source of attributes for the Subject which is needed to properly route the request to the appropriate BAE Broker.

The above also provides a binding between the LI for a BAE (the entityID) and the public key of the signing/encryption certificate generated for the BAE and distributed via the BAE metadata. This allows a Relying Party the option to parse the certificate used to sign the attribute query for the EntityID (LI) of the attribute requester (Ultimate Requester), which in turn can be used to:

- Verify with a high degree of assurance, by comparing the entityID (LI) parsed from the certificate to the entityID (LI) found in the <saml:Issuer> element in the request, the identity of the Ultimate Requester.
- Dynamically look up, in the BAE Metadata, the public key of the Ultimate Requester that MUST be used to encrypt the response.

Task

Dept B (urn:bae:1:b) System requesting attributes of Subject in Dept A (urn:bae:1:a)

Actors

Dept A EntityID (LI) - urn:bae:1:a
Dept A Signing/Encryption Cert - <x.509.cert.A>
Dept A BAE Broker Endpoint - http://A/url.a

Dept B EntityID (LI) - urn:bae:1:b
Dept B Signing/Encryption Cert - <x.509.cert.B>
Dept C BAE Broker Endpoint - http://B/url.b

Request Flow

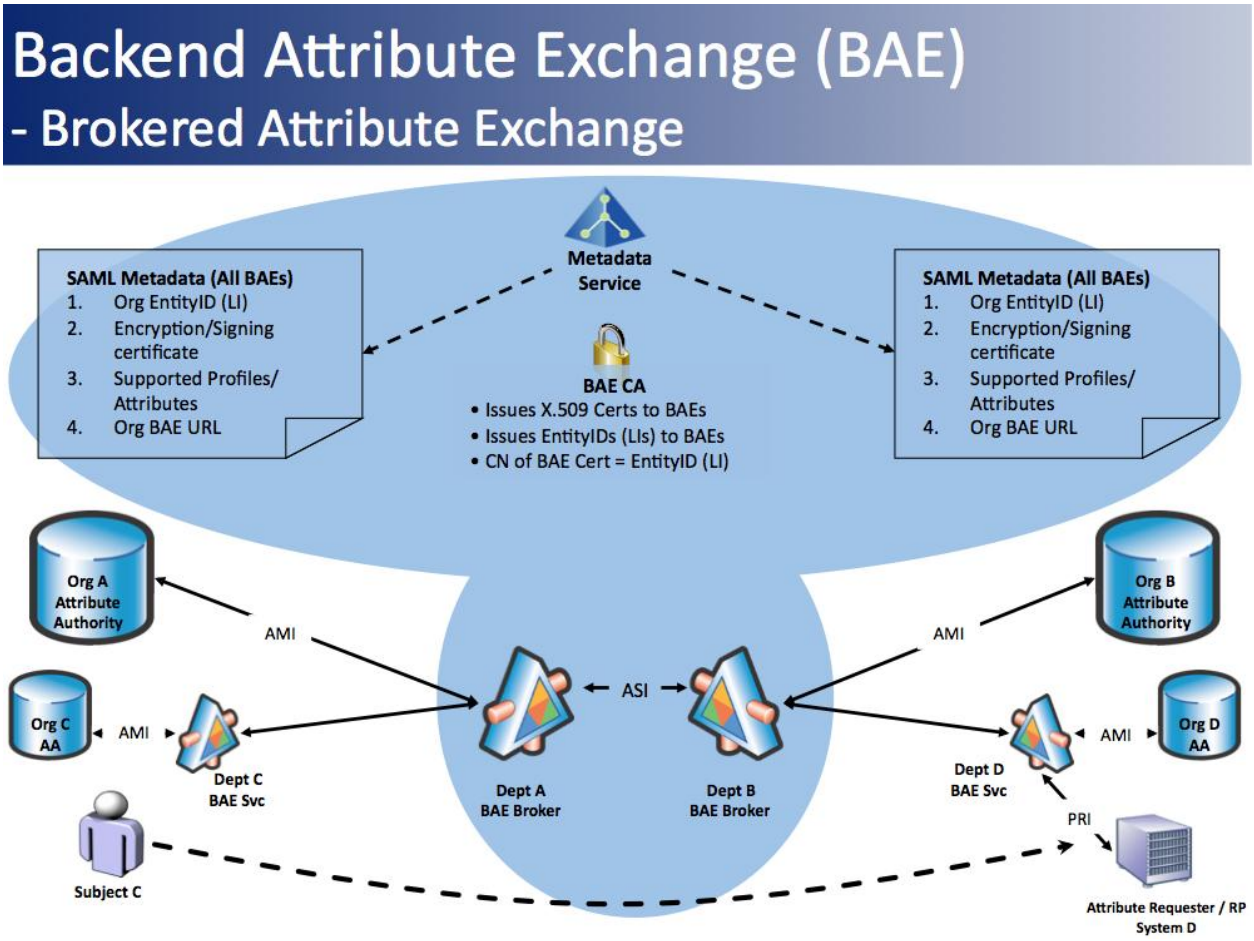
1. Attribute Requester B obtains the FASC-N of the Subject A i.e. the LUID (from a credential or trusted repository) and passes the FASC-N over a trusted channel to the BAE Broker B
2. The BAE Broker B Internal Service maps the OC and OI in the FASC-N to the LI (“urn:bae:1:a”) of the Subject A’s BAE Broker A.
3. BAE Broker B formulates <samlp:AttributeQuery>
   Destination = “urn:bae:1:a”
   Issuer = “urn:bae:1:b”

4. BAE Broker B encrypts Subject/NameID in <samlp:AttributeQuery> using <x.509.cert.A> which it finds in the BAE metadata using urn:bae:1:a as the lookup key
5. BAE Broker B applies <x.509.cert.B> Digital Signature to <samlp:AttributeQuery>
6. BAE Broker B routes request to http://B/url.a based on <Destination> in request
7. BAE Broker A validates Digital Signature on SOAP Request as being that of "urn:bae:1:b"
8. BAE Broker A checks <Destination>, sees it is for itself, strips "urn:bae:1:b" signature on SOAP Request
9. BAE Broker A un-encrypts Subject NameID (LUID) in the Request using its private key

Response Flow

1. BAE Broker A Queries its AA and Retrieves attributes of Requested User using the LUID
2. BAE Broker A Creates an Attribute Response
   Issuer = "urn:bae:1:a"
   Destination = "urn:bae:1:b"
   <Destination> value is taken from Request <saml:Issuer> or request Digital Signature
3. BAE Broker A encrypts Response using public key in Request <saml:Issuer> i.e. <x.509.cert.B>
4. BAE Broker A digitally signs <samlp:Response > using its <x.509.cert.A>
5. BAE Broker A signs SOAP Response using <x.509.cert.A>
6. BAE Broker A sends response to http://A/url.b
7. BAE Broker B validates Digital Signature on SOAP Response as being that of "urn:bae:1:a"
8. BAE Broker B checks <Destination>, sees it is itself, strips "urn:bae:1:a" signature on SOAP Response
9. BAE Broker B un-encrypts <saml:EncryptedAssertion> in the Response using its private key
10. BAE Broker B forwards response over secure channel to Attribute Requester B

6.3.2 Implementation – BAE Brokered Attribute Exchange (Informative)



Preconditions

- Locale Identifier (LI) format = urn:bae:OC:OI

  OC: Organizational Category

  1. Federal Government Agency
  2. State Government Agency
  3. Commercial Enterprise
  4. Foreign Government

  OI: Organizational ID

  NIST SP800-87 Agency Code for Federal Agencies

  e.g:

  urn:bae:1:7000 - for DHS

urn:bae:1:9700 - for DOD

- Department A LI = “urn:bae:1:a” (Responder)
  Department B LI = “urn:bae:1:b” (Requester)
  Department C LI = “urn:bae:1:c” (Ultimate Responder)
  Department D LI = “urn:bae:1:d” (Ultimate Requester)
- Credential is a PIV Card
- Subject Name Identifier (LUID) is a FASC-N

The above provides a mapping between the OC and OI elements obtained by the Relying Party from the FASC-N and the entityID (LI) assigned to the BAE Broker who is the authoritative source of attributes for the Subject which is needed to properly route the request to the appropriate BAE Broker.

The above also provides a binding between the LI for a BAE (the entityID) and the public key of the signing/encryption certificate generated for the BAE and distributed via the BAE metadata. This allows a Relying Party the option to parse the certificate used to sign the attribute query for the entityID (LI) of the attribute requester (Ultimate Requester), which in turn can be used to:

- Verify with a high degree of assurance, by comparing the entityID (LI) parsed from the certificate to the entityID (LI) found in the <saml:Issuer> element in the request, the identity of the Ultimate Requester.
- Dynamically look up, in the BAE Metadata, the public key of the Ultimate Requester that MUST be used to encrypt the response.

Task

- Dept D (urn:bae:1:d) requesting attributes of user in Dept C (urn:bae:1:c) i.e. Department D is the Ultimate Requester
  - Dept D is sharing Dept B's BAE Broker (urn:bae:1:b) i.e. URL of BAE broker is same for B&D in SAML Metadata (separate LI/EntityID & Certs)
  - Dept B BAE Broker should NOT have any data visibility into Dept D request/response messages
  - Dept C is sharing Dept A’s BAE Broker (urn:bae:1:a) i.e. URL of BAE broker is same for A&C in SAML Metadata (separate LI/EntityID & Certs) i.e. Department C is the Ultimate Responder
  - Dept A BAE Broker should NOT have any data visibility into Dept C request/response messages

Actors

Dept A EntityID (LI) - urn:bae:1:a
Dept A Signing/Encryption Cert - <x.509.cert.A>
Dept A BAE Broker Endpoint - http://A/url.a

Dept C EntityID (LI) - urn:bae:1:c
Dept C Signing/Encryption Cert - <x.509.cert.C>
Dept C BAE Broker Endpoint - http://A/url.a

Dept B EntityID (LI) - urn:bae:1:b
Dept B Signing/Encryption Cert - <x.509.cert.B>
Dept B BAE Broker Endpoint - http://B/url.b

Dept D EntityID (LI) - urn:bae:1:d
Dept D Signing/Encryption Cert - <x.509.cert.D>
Dept D BAE Broker Endpoint - http://B/url.b

Request Flow

1. Attribute Requester D obtains the FASC-N (LUID) of the Subject C and passes the FASC-N over a trusted channel to the Dept D BAE Svc.
2. The Dept D BAE Svc maps the OC and OI in the FASC-N to the LI/entityID ("urn:bae:1:c") of the Subject C's BAE Broker.
3. Dept D BAE Svc formulates <samlp:AttributeQuery>
   Destination = "urn:bae:1:c"
   Issuer = "urn:bae:1:d"
4. Dept D BAE Svc encrypts Subject/NameID in <samlp:AttributeQuery> using <x.509.cert.C>
5. Dept D BAE Svc applies Digital Signature <x.509.cert.C> to <samlp:AttributeQuery>
6. Dept D BAE Svc sends Request to its BAE Broker B over trusted channel
7. BAE Broker B applies Digital Signature <x.509.cert.B> to SOAP Request
8. BAE Broker B routes Request to http://B/url.a based on <Destination> in Request i.e. URL of BAE Broker for A & C are the same in the SAML Metadata, although both A & C have their own unique EntityID's (LI) and Signing/Encryption Certs Issued to them by the BAE CA.
9. BAE Broker A validates Digital Signature on SOAP Request as being that of "urn:bae:1:b"
10. BAE Broker A has no visibility into <saml:EncryptedID> since it does not have <x.509.cert.C> private key
11. BAE Broker A checks <Destination>, sees it is not itself, and routes to Dept C BAE Svc "urn:bae:1:c "
12. As an option BAE Broker A strips incoming digital signature from BAE Broker B and applies its own Digital Signature <x.509.cert.A> to routed Message
13. Dept C BAE Svc may validate "urn:bae:1:a" digital signature

14. Dept C BAE Svc un-encrypts Subject/NameID (LUID) in the Request using its private key

Response Flow

1. Dept C BAE Svc Queries its AA and Retrieves attributes of Requested User
2. Dept C BAE Svc Creates an Attribute Response
   Issuer = “urn:bae:1:c”
   Destination = “urn:bae:1:d”
   <Destination> in Response is from Request <saml:Issuer>
3. Dept C BAE Svc encrypts Response using public key of Request <saml:Issuer> i.e. <x.509.cert.D>
4. Dept C BAE Svc digitally signs <samlp:Response> using <x.509.cert.C>
5. Dept C BAE Svc sends response to its BAE Broker A
6. BAE Broker A validates “urn:bae:1:c” digital signature applied to <samlp:Response>
7. BAE broker has no visibility into Response since it is encrypted and does not have access to <x.509.cert.D> private key
8. BAE Broker A signs SOAP Response using <x.509.cert.A>
9. BAE Broker A routes response to http://A/url.b using <Destination> in Response i.e. URL of BAE Broker for D & B are the same in the SAML Metadata, although both D & B have their own unique EntityID's (LI) and Signing/Encryption Certs Issued to them by the BAE CA.
10. BAE Broker B validates Digital Signature on SOAP Response as being that of “urn:bae:1:a”
11. BAE Broker B has no visibility into <saml:EncryptedAssertion> since it does not have <x.509.cert.D> private key
12. BAE Broker B checks <Destination> on Response and routes to Dept D BAE Svc “urn:bae:1:d”
13. As an option, BAE Broker B may strip “urn:bae:1:a” signature on SOAP Response and apply its own digital signature <x.509.cert.B> to routed message
14. Dept D BAE Svc may validate “urn:bae:1:b” digital signature
15. Dept D BAE Svc un-encrypts <saml:EncryptedAssertion> in the Response using its private key
16. Dept D BAE Svc forwards response over secure channel to Attribute Requester D