



Biometric Identity Assurance Services (BIAS) SOAP Profile, Version 1.0

Working Draft 07

27 November 2008

Specification URIs:

This Version:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.pdf>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.doc>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0-wd-07.html>

Previous Version:

N/A

Latest Version:

<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.pdf>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.doc>
<http://docs.oasis-open.org/bias/soap-profile/v1.0/biasprofile-1.0.html>

Technical Committee:

OASIS Biometric Identity Assurance Services (BIAS) Integration TC

Chair(s):

Cathy Tilton, Daon

Editor(s):

TBD

Related Work:

This specification is related to:

- ANSI INCITS 442-2008, Biometric Identity Assurance Services (BIAS)

Declared XML Namespace(s):

<http://docs.oasis-open.org/bias/ns/bias-1.0/>

Abstract:

This document specifies a SOAP profile that implements the BIAS abstract operations specified in INCITS 442 as SOAP messages.

Status:

This document was last revised or approved by the OASIS BIAS TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/bias/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/bias/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/bias/>.

Notices

Copyright © OASIS® 2008. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", "BIAS" are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction.....	7
1.1	Purpose/Scope	7
1.2	Overview.....	7
1.3	Background.....	7
1.4	Relationship to Other Standards.....	7
1.5	Terminology	8
1.6	References	8
1.6.1	Normative References.....	8
1.6.2	Non-Normative References.....	9
2	Design Concepts and Architecture (non-normative).....	10
2.1	Philosophy	10
2.2	Context	10
2.3	Architecture.....	10
3	Data model/data dictionary.....	12
3.1	Documentation Conventions.....	12
3.2	Common Elements	12
3.2.1	ApplicationIdentifier	12
3.2.2	ApplicationUserIdentifier.....	13
3.2.3	BDB-info	13
3.2.4	BIASBiometricDataType.....	14
3.2.5	BIASFaultCode.....	14
3.2.6	BIASFaultDetail	15
3.2.7	BIASIdentity.....	15
3.2.8	BIASIDType.....	15
3.2.9	BiographicDataItem Type	16
3.2.10	BiographicDataSetType.....	16
3.2.11	BiographicDataType	16
3.2.12	BiometricDataElementType.....	17
3.2.13	BiometricDataListType	17
3.2.14	BIR-info	17
3.2.15	CandidateListType.....	18
3.2.16	CandidateType	18
3.2.17	CapabilityListType	18
3.2.18	CapabilityName	18
3.2.19	CapabilityType.....	19
3.2.20	CBEFF_BIR_ListType	20
3.2.21	CBEFF_BIR_Type.....	20
3.2.22	CBEFF_XML_BIR_Type	20
3.2.23	Classification	21
3.2.24	ClassificationAlgorithmType	21
3.2.25	ClassificationData.....	21
3.2.26	EncounterListType.....	21
3.2.27	FusionInformationListType	21

3.2.28	FusionInformationType	22
3.2.29	GenericRequestParameters	22
3.2.30	IdentityModelType	22
3.2.31	InformationType.....	23
3.2.32	ListFilterType	23
3.2.33	MatchType.....	23
3.2.34	ProcessingOptionsType	23
3.2.35	ProductID	23
3.2.36	QualityData.....	24
3.2.37	ResponseStatus	24
3.2.38	ReturnCode	24
3.2.39	SB-info.....	24
3.2.40	Score	24
3.2.41	TokenType	25
3.2.42	VendorIdentifier	25
3.2.43	Version	25
3.2.44	VersionType	25
3.3	BIAS Operations.....	25
3.3.1	Primitive Operations	25
3.3.2	Aggregate Operations	61
4	Message structure and rules	72
4.1	Purpose and constraints	72
4.2	Message requirements	73
4.3	Identifying operations.....	74
4.3.1	Operation name element	74
4.3.2	WS-Addressing Action.....	75
4.4	Security.....	76
4.4.1	Use of SSL 3.0 or TLS 1.0.....	76
4.4.2	Data Origin Authentication.....	76
4.4.3	Message Integrity	76
4.4.4	Message Confidentiality	76
4.4.5	Security Considerations.....	76
4.4.6	Security of Stored Data	77
4.4.7	Key Management	77
4.5	Use with other WS* standards	77
4.6	Tailoring.....	77
5	Error handling.....	78
6	Conformance.....	79
A.	XML Schema.....	80
B.	Use Cases (non-normative).....	136
B.1	Verification Use Case.....	136
B.2	Asynchronous Verification.....	137
B.3	Primitive Verification.....	138
B.4	Identification Use Case	139
B.5	Biometric Enrollment	140

B.6 Primitive Enrollment	141
C. Samples (non-normative)	142
C.1 Create Subject Request/Response Example	142
C.2 Set Biographic Data Request/Response Example	144
C.3 Set Biometric Data Request/Response Example	145
D. Acknowledgements	147
E. Revision History	148
F. Open Issues (temporary annex)	149

1 Introduction

1.1 Purpose/Scope

This Organization for the Advancement of Structured Information Standards (OASIS) Biometric Identity Assurance Services (BIAS) profile specifies how to use the eXtensible Markup Language (XML) defined in ANSI INCITS 442-2008 – Biometric Identity Assurance Services [INCITS-BIAS] to invoke Simple Object Access Protocol (SOAP) -based services that implement BIAS operations. These SOAP-based services enable an application to invoke biometric identity assurance operations remotely in a Services Oriented Architecture (SOA) infrastructure.

Not included in the scope of BIAS is the incorporation of biometric authentication as an integral component of an authentication or security protocol.

1.2 Overview

In addition to this introduction, this standard includes the following:

- Chapter 2 contains conformance requirements (TBD).
- Chapter 3 presents the design concepts and architecture for invoking SOAP-based services that implement BIAS operations.
- Chapter 4 presents the namespaces necessary to implement this profile, INCITS BIAS operations and data elements, and identifies relationships to external data definitions.
- Chapter 5 presents the BIAS message structure, as well as rules and considerations for its application.
- Chapter 6 presents information on error handling.
- Appendices include the OASIS BIAS XML schema, sample Web Service Definition Language (WSDL) and WSDL templates, use cases, acknowledgements, and the revision history of this profile.

1.3 Background

In late 2005/early 2006, a gap was identified in the existing biometric standards portfolio with respect to biometric services. The Biometric Identity Assurance Services standard proposal was for a collaborative effort between government and private industry to provide a services-based framework for delivering identity assurance capabilities, allowing for platform and application independence. This standard proposal required the attention of two major technical disciplines: biometrics and service architectures. The expertise of both disciplines was required to ensure the standard was technically sound, market relevant, and achieved widespread adoption. The International Committee for Information Technology Standards (INCITS) M1 provided the standards leadership relevant to biometrics, defining the “taxonomy” of biometric operations and data elements. OASIS provided the standards leadership relevant to service architectures with an initial focus on web services, defining the schema and protocol.

The driving requirements of the BIAS standard proposal were to provide the ability to remotely invoke biometric operations across an SOA infrastructure; to provide business level operations without constraining the application/business logic that implements those operations; to be as generic as possible – technology, framework, & application domain independent; and to provide basic capabilities that can be used to construct higher level, aggregate/composite operations.

1.4 Relationship to Other Standards

This OASIS BIAS profile comprises a companion standard to ANSI INCITS 442-2008 – Biometric Identity Assurance Services, which defines the BIAS requirements and taxonomy, specifying the identity

assurance operations and the associated data elements. This OASIS BIAS profile specifies the design concepts and architecture, data model and data dictionary, message structure and rules, and error handling necessary to invoke SOAP-based services that implement BIAS operations.

Together, the BIAS standard and the BIAS profile provide an open framework for deploying and remotely invoking biometric-based identity assurance capabilities that can be readily accessed across an SOA infrastructure.

This relationship allows the leveraging of the biometrics and web services expertise of the two standards development organizations. Existing standards are available in both domains and many of these standards will provide the foundation and underlying capabilities upon which the biometric services depend.

1.5 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

The following additional terms and definitions are used:

Note: The terms and definitions specified in INCITS (InterNational Committee for Information Technology Standards) (Project 1823-D) also apply to this Standard.

BIAS

Biometric Identity Assurance Services

BIR

Biometric Information Record

ESB

Enterprise Service Bus

SOA

Service-Oriented Architecture

CBEFF

Common Biometric Exchange Formats Framework - data elements and BIR formats specified in ISO/IEC 19785-1

BIAS implementation

software entity that is capable of creating, processing, sending, and receiving BIAS messages

BIAS endpoint

runtime entity, identified by an endpoint IRI, capable of sending and receiving BIAS messages, and containing a running BIAS implementation

BIAS message

message that can be sent from a BIAS endpoint to another BIAS endpoint through a BIAS link channel

request BIAS message

BIAS message conveying a request for an action to be performed by the receiving BIAS endpoint

response BIAS message

BIAS message conveying a response to a prior request BIAS message

1.6 References

1.6.1 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

- 90
- 91 **[INCITS-BIAS]** ANSI INCITS 442-2008, Biometric Identity Assurance Services (BIAS), May 2008
- 92 <http://www.incits.org>
- 93
- 94 **[SOAP11]** Simple Object Access Protocol (SOAP) 1.1, 8 May 2000
- 95 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 96 **[WS-Addr]** Web Services Addressing (WS-Addressing), 10 Aug 2004
- 97 <http://www.w3.org/Submission/ws-addressing/>
- 98 **[WS-I-Basic]** Basic Profile Version 1.1, 10 April 2006
- 99 <http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html>
- 100 **[WS-I-Bind]** Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile
- 101 Version 1.0, 24 August 2004
- 102 <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html>
- 103 **[WSDL11]** Web Services Description Language (WSDL) 1.1, 15 March 2001
- 104 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- 105 **[XML 10]** Extensible Markup Language (XML) 1.0, 16 August 2006
- 106 <http://www.w3.org/TR/2006/REC-xml-20060816/>
- 107 **[XOP]** XML-binary Optimized Packaging, W3C Recommendation, 25 January 2005
- 108 <http://www.w3.org/TR/2005/REC-xop10-20050125/>

1.6.2 Non-Normative References

- 110 **[RFC2246]** T. Dierks & C. Allen, *The TLS Protocol, Version 1.0*, January 1999
- 111 <http://www.ietf.org/rfc/rfc2246.txt>
- 112 **[RFC2617]** J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*,
- 113 June 1999
- 114 <http://www.ietf.org/rfc/rfc2617.txt>
- 115 **[RFC3280]** R. Housley, et al, Internet X.509 Public Key Infrastructure Certificate and
- 116 Certificate Revocation List (CRL) Profile, April 2002
- 117 <http://www.ietf.org/rfc/rfc3280.txt>
- 118 **[SAML]** Security Assertion Markup Language (SAML), Oasis Standard, March 2005
- 119 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 120 **[SSL3]** SSL 3.0 Specification
- 121 <http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM>
- 122 **[X509]** X.509: Information technology - Open Systems Interconnection - The Directory:
- 123 Public-key and attribute certificate frameworks, ITU-T, August 2005

2 Design Concepts and Architecture (non-normative)

2.1 Philosophy

Rather than define a totally new and unique messaging protocol for biometric services, this specification instead defines a method for using existing biometric and Web services standards to exchange biometric data and perform biometric operations.

2.2 Context

Today, biometric systems are being developed which collect, process, store and match biometric data for a variety of purposes. In many cases, data and/or capabilities need to be shared between systems or systems serve a number of different client stakeholders. As architectures move towards services-based frameworks, access to these biometric databases and services is via a Web services front-end. However, lack of standardization in this area has led implementers to develop customized services for each system/application.

BIAS is intended to provide a common, yet flexible, Web services interface that can be used within both closed and open SOA systems. Figure 1, below, depicts the context in which the BIAS messages will be implemented.

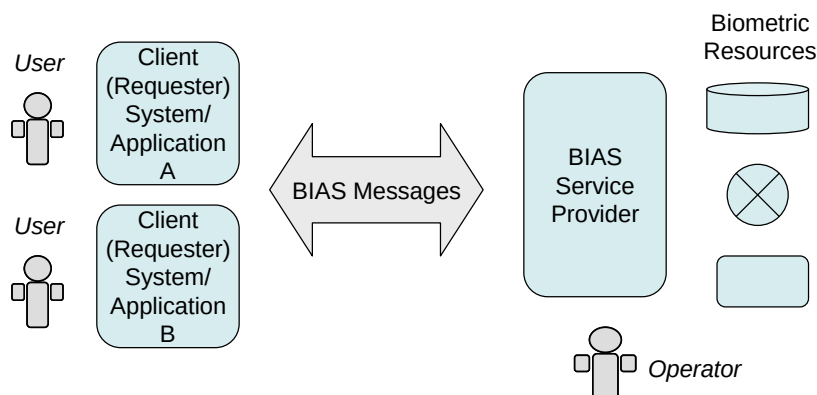


Figure 1. BIAS Context

The clients (requesters) may use standard discovery mechanisms (i.e., UDDI directories) to discover the BIAS service provider (implementation) or, particularly in closed systems, the URI and WSDL for the service provider may be known a priori by the client BIAS application developer.

2.3 Architecture

BIAS Web services are intended to be used within systems employing a services framework, such as a services-oriented architecture (SOA) (although implementations are not limited to this environment). As such, it is recognized that the clients may interact directly with the BIAS service provider or layers may exist between the client and the service provider, for example as an ESB or other application layer.

The BIAS Architecture as shown in Figure 2, in which:

- A Client request to the BIAS Web services may be triggered by a human interaction OR any proxy system such as an ESB.
- Client sends and receives SOAP messages that conform to the BIAS schemas

- The calls to the BIAS Implementation use OASIS Service Interfaces and Bindings (via WSDL)
- The BIAS implementation maps the service call to the appropriate internal API or set of APIs and returns data according to the service interface.

Note that services are represented as circles.

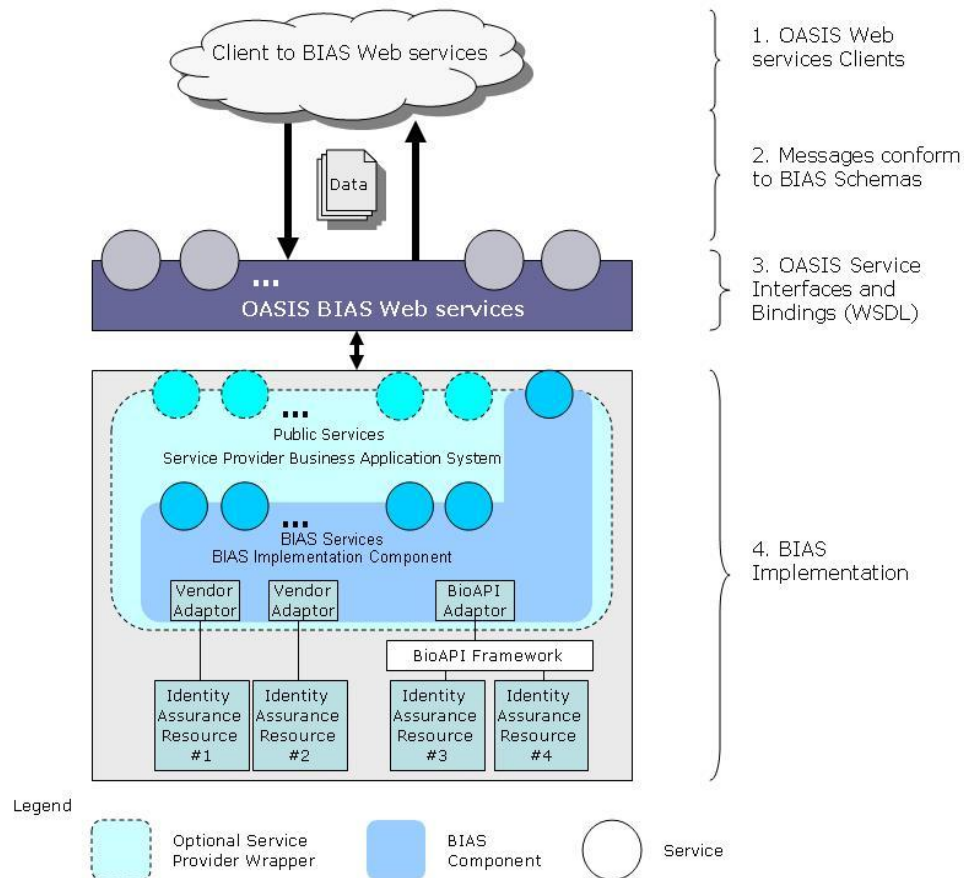


Figure 2. BIAS Architecture

At the heart of the BIAS messaging protocol are the concepts of BIAS messages, endpoints, and links.

BIAS implementation

A BIAS implementation is a software entity that is capable of creating, processing, sending, and receiving BIAS messages. This standard does not define requirements for the BIAS implementation other than defining the messages and protocols used by the endpoints.

BIAS messages

A BIAS message is a one that can be sent from a BIAS endpoint to another BIAS endpoint over a TCP/IP link.

BIAS endpoints

A BIAS endpoint is a runtime entity, uniquely identified and accessed by an endpoint IRI, capable of sending and receiving BIAS messages.

Note that when not publicly and directly exposed, the endpoints for purposes of this specification are the BIAS service provider exposing BIAS services and the component that directly interacts with that service provider, e.g., the business application or ESB, rather than the ultimate end client requester.

3 Data model/data dictionary

This section describes the BIAS operations and the data elements contained therein. Common data elements are defined for use in one or more operations. These include common data types or return codes. The operations are listed alphabetically, with each operation containing a request and a response message.

BIAS operations and data elements are defined in ANSI INCITS 442-2008. The elements, complex types and simple types described for the BIAS messages belong to the following namespace: <http://docs.oasis-open.org/bias/ns/bias-1.0/>. See Appendix B for the XML schema.

3.1 Documentation Conventions

Each operation has a section describing the request and response messages and the input and output parameters. Additionally, each common element has a section describing its content. The input and output of each message and the comment elements are detailed in a table as described in the figure below. Each field that forms part of the input/output to the message is detailed in the table.

Header Name	Description	Values	Value Meaning
Field	The name of the field.		
Type	The XML schema type of the field.		
#	The cardinality of the field	1	One occurrence
		0..1	Zero or one occurrence
		0..*	Zero or more occurrences
		1..*	One or more occurrences
?	Defines if the field must be present.	Y	Yes – is always required
		N	No – is not always required, an optional field.
		C	Conditional – requirement is dependent on system or message conditions.
Meaning	Gives a short description of the field's use		

Figure 3. BIAS Message Input/Output Dictionary Table Headings

Fields Hierarchy Explained:

To denote the field hierarchy the symbol $\uparrow$ is used to denote the child-of relationship.

3.2 Common Elements

3.2.1 ApplicationIdentifier

Type:	string
Description:	Identifies an application.

Min Length:	1
Max Length:	255

200 3.2.2 ApplicationUserIdentifier

Type:	string
Description:	Identifies an application user or instance.
Min Length:	1
Max Length:	255

201 3.2.3 BDB-info

Field	Type	#	?	Meaning
BDB-info				For a description of each data element, see the referenced CBEFF standards in the CBEFF XML BIR Type definition.
↑ challenge-response	string	0..1	N	
↑ index	string	0..1	N	
↑ format-owner	integer	1	Y	
↑ format-type	integer	1	Y	
↑ security-Encryption-options	string	1	Y	
↑ creation-date	string	0..1	N	
↑ validity-period	string	0..1	N	
↑ type	string	0..1	N	
↑ subtype	string	0..1	N	
↑ data-type	string	0..1	N	
↑ product-owner	integer	1	Y	
↑ product-type	integer	1	Y	
↑ purpose	string	0..1	N	
↑ quality		0..1	N	
↑ value	string	1	Y	
↑ algorithm-owner	string	0..1	N	
↑ algorithm-type	string	0..1	N	
↑ device-owner	integer	0..1	N	
↑ device-type	integer	0..1	N	

202 3.2.4 BIASBiometricDataType

Field	Type	#	?	Meaning
BIASBiometricDataType			Y	Wraps the various BIAS biometric types. The operations that use this type specify which elements are required.
↗ BIRList	CBEFF_BIR_ListType	0..1	N	A list of CBEFF-BIR elements.
↗ BIR	CBEFF_BIR_Type	0..1	N	Contains biometric information in either a non-XML or an XML representation.
↗ InputBIR	CBEFF_BIR_Type	0..1	N	Maps to specific INCITS BIAS elements as required by that specification.
↗ ReferenceBIR	CBEFF_BIR_Type	0..1	N	Maps to specific INCITS BIAS elements as required by that specification.
↗ BiometricDataList	BiometricDataListType	0..1	N	A list of biometric data elements.

203 3.2.5 BIASFaultCode

Type:	string
Description:	Error code referenced in a SOAP fault.

204 BIASFaultCode Enumeration Values

Value	Description
UNKNOWN_ERROR	The service failed for an unknown reason.
UNSUPPORTED_CAPABILITY	A requested capability is not supported by the service implementation.
INVALID_INPUT	The data in a service input parameter is invalid.
BIR_QUALITY_ERROR	Biometric sample quality is too poor for the service to succeed.
INVALID_BIR	The input BIR is empty or in an invalid or unrecognized format.
BIR_SIGNATURE_FAILURE	The service could not validate the signature, if used, on the input BIR.
BIR_DECRYPTION_FAILURE	The service could not decrypt an encrypted input BIR.
INVALID_ENCOUNTER_ID	The input encounter ID is empty or in an invalid format.
INVALID_SUBJECT_ID	The input subject ID is empty or in an invalid format.
UNKNOWN_SUBJECT	The subject referenced by the input subject ID does not exist.
UNKNOWN_GALLERY	The gallery referenced by the input gallery ID does not exist.
UNKNOWN_ENCOUNTER	The encounter referenced by the input encounter ID does not exist.

205 **3.2.6 BIASFaultDetail**

Field	Type	#	?	Meaning
BIASFaultDetail			Y	Defines the error information associated with a SOAP fault.
↑ BIASFaultType	BIASFaultCode	1	Y	References an error code.
↑ BIASFaultMessage	string	1	Y	Provides an explanation of the fault.
↑ BIASFaultDescription	string	0..1	N	Provides detailed information about a BIAS fault, such as trace details.

206 **3.2.7 BIASIdentity**

Field	Type	#	?	Meaning
BIASIdentity			Y	Defines a single element for encapsulating the data associated with an Identity. Includes the Identity's reference identifiers, biographic data, and biometric data. The operations that use this type specify which elements are required.
↑ SubjectID	BIASIDType	0..1	C	A system unique identifier for a subject. Required as input to many operations.
↑ IdentityClaim	BIASIDType	0..1	N	An identifier by which a subject is known to a particular gallery or population group.
↑ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
↑ EncounterList	EncounterListType	0..1	N	A list of encounters associated with a subject.
↑ BiographicData	BiographicDataType	0..1	N	An Identity's biographic data.
↑ BiographicDataElements	BiographicDataType	0..1	N	An Identity's biographic data elements that are stored in the implementing system.
↑ BiometricData	BIASBiometricDataType	0..1	N	An Identity's biometric data.

207 **3.2.8 BIASIDType**

Type:	string
Description:	A BIAS Identifier.

208 3.2.9 BiographicDataItemType

Field	Type	#	?	Meaning
BiographicDataItemType			Y	Defines a single biographic data element.
↑ Name	string	1	Y	The name of the biographic data item.
↑ Type	string	1	Y	The data type for the biographic data item.
↑ Value	string	0..1	N	The value assigned to the biographic data item.

209 3.2.10 BiographicDataSetType

Field	Type	#	?	Meaning
BiographicDataSetType			Y	Defines a set of biographic data that is formatted according to the specified format.
↑ <i>unspecified</i>	any	0..*	N	Biographic data formatted according to a specific format.

210 BiographicDataSetType Attributes

Name	Type	Use	Meaning
name	string	required	The name of the biographic data format. Use these names for common formats: EFTS, EBTS, NIEM, xNAL, HR-XML.
version	string	optional	The version of the biographic data format. Use these versions for common formats: (EFTS) 7.1, (EBTS) 1.2, (NIEM) 1.0 or 2.0, (xNAL) 2.0 or 3.0, (HR-XML) 2.5.
source	string	required	Reference to a URI describing the biographic data format. Use these sources for common formats: (EFTS) www.fbi.gov, (EBTS) www.biometrics.dod.mil, (NIEM) www.niem.gov, (xNAL) www.oasis-open.org, (HR-XML) www.hr-xml.org.
type	string	required	The biographic data format type. Use these types for common formats: (EFTS and EBTS) ASCII, (NIEM, xNAL, and HR-XML) XML.

211 3.2.11 BiographicDataType

Field	Type	#	?	Meaning
BiographicDataType			Y	Defines a set of biographic data elements, utilizing either the BiographicDataItemType to represent a list of elements or the BiographicDataSetType to represent a complete, formatted set of biographic information. One of the following elements must be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.

Field	Type	#	?	Meaning
↑ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

212 3.2.12 BiometricDataElementType

Field	Type	#	?	Meaning
BiometricDataElementType			Y	Provides descriptive information about biometric data, such as the biometric type, subtype, and format, contained in the BDB of the CBEFF-BIR.
↑ BiometricType	hexBinary	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricTypeCount	positiveInteger	0..1	N	The number of biometric records having the biometric type recorded in the biometric type field.
↑ BiometricSubType	hexBinary	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.
↑ BDBFormatOwner	string	1	Y	Identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data.
↑ BDBFormatType	string	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.

213 3.2.13 BiometricDataListType

Field	Type	#	?	Meaning
BiometricDataListType			Y	A list of biometric data elements.
↑ BiometricDataElement	BiometricDataElementType	0..*	N	Data structure containing information about a biometric record.

214 3.2.14 BIR-info

Field	Type	#	?	Meaning
BIR-info		0..1	Y	For a description or definition of each data element, see the referenced CBEFF standards in the CBEFF XML BIR Type schema.
↑ creator	string	0..1	N	
↑ index	string	0..1	N	
↑ payload	anyType	0..1	N	
↑ integrity-options	string	0..1	N	

Field	Type	#	?	Meaning
↑ creation-date	string	0..1	N	
↑ validity-period	string	0..1	N	
↑ patron-format-owner	string	0..1	N	
↑ patron-format-type	string	0..1	N	

215 3.2.15 CandidateListType

Field	Type	#	?	Meaning
CandidateListType			Y	Defines a set of candidates, utilizing the CandidateType to represent each element in the set.
↑ Candidate	CandidateType	0..*	N	A single candidate.

216 3.2.16 CandidateType

Field	Type	#	?	Meaning
CandidateType			Y	Defines a single candidate as a possible match in response to a biometric identification request.
↑ Score	string	0..1	N	The match score.
↑ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↑ BIRList	CBEFF_BIR_ListType	1	Y	Biometric data associated with the candidate match.

217 CandidateType Attributes

Name	Type	Use	Meaning
Rank	integer	required	The rank of the candidate in relation to other candidates for the same biometric identification operation.

218 3.2.17 CapabilityListType

Field	Type	#	?	Meaning
CapabilityListType			Y	Defines a set of capabilities.
↑ Capability	CapabilityType	0..*	N	A single capability.

219 3.2.18 CapabilityName

Type:	string
Description:	A list of capability items.

220 CapabilityName Enumeration Values

Value	Description
AggregateInputDataOptional	A data element accepted as optional input by the implementing system for the aggregate services.
AggregateInputDataRequired	A data element required as input by the implementing system for the aggregate services.
AggregateProcessingOption	A processing option supported by the implementing system for the aggregate services.
AggregateReturnData	A data element returned by the implementing system for the aggregate services.
AggregateServiceDescription	Describes the processing logic of an aggregate service supported by the implementing system.
BiographicDataSet	Identifies a biographic data set supported by the implementing system.
CBEFFPatronFormat	A patron format supported by the implementing system.
ClassificationAlgorithmType	A classification algorithm type supported by the implementing system.
ConformanceClass	Identifies the conformance class of the BIAS implementation.
Gallery	A gallery or population group supported by the implementing system.
IdentityModel	Identifies whether the implementing system is person-centric or encounter-centric based.
QualityAlgorithm	A quality algorithm vendor and algorithm vendor product ID supported by the implementing system.
SupportedBiometric	A biometric type supported by the implementing system.
TransformOperation	A transform operation type supported by the implementing system.

221 3.2.19 CapabilityType

Field	Type	#	?	Meaning
CapabilityType			Y	Defines a single capability supported by an implementing system.
↑ CapabilityName	CapabilityName	1	Y	The name of the capability.
↑ CapabilityID	string	0..1	N	An identifier assigned to the capability by the implementing system.
↑ CapabilityDescription	string	0..1	N	A description of the capability.
↑ CapabilityValue	string	0..1	N	A value assigned to the capability.
↑ CapabilitySupportingValue	string	0..1	N	A secondary value supporting the capability.

222 **3.2.20 CBEFF_BIR_ListType**

Field	Type	#	?	Meaning
CBEFF_BIR_ListType			Y	A list of CBEFF-BIR elements.
↑ _{BIR}	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

223 **3.2.21 CBEFF_BIR_Type**

Field	Type	#	?	Meaning
CBEFF_BIR_Type			Y	Represents biometric information, with either a non-XML or XML representation. One of the following elements must be present.
↑ _{unspecified}	any	0..1	C	Must be present if XML_BIR is not used.
↑ _{XML_BIR}	CBEFF_XML_BIR_Type	0..1	C	Must be present if the <i>unspecified</i> element is not used.

224 **3.2.22 CBEFF_XML_BIR_Type**

Field	Type	#	?	Meaning
CBEFF_XML_BIR_Type			Y	An XML representation for biometric information. This schema represents CBEFF data elements from both the US (INCITS 398-2005 Revision 1) and International (ISO/IEC 19785-1:2006) versions of the CBEFF standard. The data elements in the two standards are fairly similar. However, a data element may have a different meaning and/or a different set of valid values based on which standard is used. The cbeff-version data element specifies whether the US or International version is being used. An additional data element, other, has been added to this schema to capture non-standard information and to allow for flexibility to any future changes in the existing CBEFF standards. For a description or definition of each data element, see the referenced CBEFF standards.
↑ _{patron-format-identifier}	string	0..1	N	
↑ _{version}	Version	0..1	N	
↑ _{cbeff-version}	Version	0..1	N	
↑ _{bir-info}	BIR-info	0..1	N	
↑ _{bdb-info}	BDB-info	0..1	N	

Field	Type	#	?	Meaning
↑ sb-info	SB-info	0..1	N	
↑ other		0..*	N	Used to capture non-standard information and to allow for flexibility to any future changes in the existing CBEFF standards.
↑ unspecified	any	0..*	N	
↑ BIR	base64Binary	1	Y	

225 3.2.23 Classification

Type:	string
Description:	The result of a classification.

226 3.2.24 ClassificationAlgorithmType

Type:	string
Description:	Type of classification algorithm that was used to perform the classification.

227 3.2.25 ClassificationData

Field	Type	#	?	Meaning
ClassificationData			Y	Contains information on classification results and the algorithm used to determine the classification.
↑ Classification	Classification	1	Y	The result of the classification.
↑ ClassificationAlgorithmType	ClassificationAlgorithmType	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

228 3.2.26 EncounterListType

Field	Type	#	?	Meaning
EncounterListType			Y	Defines a set of encounters.
↑ EncounterID	BIASIDType	0..*	N	The identifier of an encounter.

229 3.2.27 FusionInformationListType

Field	Type	#	?	Meaning
FusionInformationListType			Y	Contains at a minimum two sets of fusion input elements, as input to the PerformFusion operation.

Field	Type	#	?	Meaning
↑ FusionElement	FusionInformationType	2..*	Y	A set of fusion information.

230 3.2.28 FusionInformationType

Field	Type	#	?	Meaning
FusionInformationType			Y	Represents the information necessary to perform a fusion operation.
↑ BiometricType	hexBinary	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricSubType	string	0..1	N	More specifically defines the type of biometric data stored in the biometric record.
↑ AlgorithmOwner	string	1	Y	The owner or vendor of the algorithm used to determine the score or decision.
↑ AlgorithmType	string	1	Y	The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision.
↑ Score	unsignedLong	0..1	C	The similarity score assigned by the matching algorithm. Either a Score or a Decision element must be used.
↑ Decision	string	0..1	C	The match decision assigned by the matching algorithm. Either a Score or a Decision element must be used.

231 3.2.29 GenericRequestParameters

Field	Type	#	?	Meaning
GenericRequestParameters			Y	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested.

232 3.2.30 IdentityModelType

Type:	string
Description:	Defines the requested or supported identity model.

233 IdentityModelType Enumeration Values

Value	Description
encounter	Designation for the encounter-centric model.
person	Designation for the person-centric model.

234 3.2.31 InformationType

Field	Type	#	?	Meaning
InformationType			Y	Allows for an unlimited number of data element types, and it does not specify nor require any particular data element.
↑ unspecified	any	0..*	N	

235 3.2.32 ListFilterType

Field	Type	#	?	Meaning
ListFilterType			Y	Provides a method to filter the amount of information returned in a search of biometric data.
↑ BiometricTypeFilter	hexBinary	1..*	Y	Limits the returned information to a specific type of biometric, as defined by CBEFF.
↑ IncludeBiometricSubType	boolean	1	Y	A Boolean flag indicating if biometric subtype information should be returned.

236 3.2.33 MatchType

Type:	boolean
Description:	The result of a fusion method.

237 3.2.34 ProcessingOptionsType

Field	Type	#	?	Meaning
ProcessingOptionsType			Y	BIAS aggregate operations support the ability to include various processing options which direct and possibly control the business logic for that operation. The ProcessingOptionsType provides a method to represent those options. Processing options should be defined by the implementing system.
↑ Option	string	0..*	N	An option supported by the implementing system.

238 3.2.35 ProductID

Type:	string
Description:	The vendor's ID for a particular product.

239 **3.2.36 QualityData**

Field	Type	#	?	Meaning
QualityData			Y	Contains information about a biometric sample's quality and the algorithm used to compute the quality.
↑ QualityScore	Score	0..1	N	The quality of a biometric sample.
↑ AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
↑ AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.
↑ AlgorithmVersion	VersionType	0..1	N	The version of the algorithm used to determine the quality.

240 **3.2.37 ResponseStatus**

Field	Type	#	?	Meaning
ResponseStatus			Y	
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

241 **3.2.38 ReturnCode**

Type:	unsignedLong
Description:	Return value specifying success or other condition.

242 **ReturnCode Enumeration Values**

Value	Description
0	Success

243 **3.2.39 SB-info**

Field	Type	#	?	Meaning
SB-info			Y	For a description or definition of each data element, see the referenced CBEFF standards in the CBEFF XML BIR Type schema.
↑ format-owner	integer	1	Y	
↑ format-type	integer	1	Y	

244 **3.2.40 Score**

Type:	integer
Description:	Match result or quality score.

3.2.41 TokenType

Field	Type	#	?	Meaning
TokenType			Y	Defines a token that is returned for asynchronous processing.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

3.2.42 VendorIdentifier

Type:	string
Description:	Identifies a vendor.

3.2.43 Version

Field	Type	#	?	Meaning
Version			Y	For a description or definition of each data element, see the referenced CBEFF standards in the CBEFF XML BIR Type schema.
↑ major	integer	1	Y	
↑ minor	integer	1	Y	

3.2.44 VersionType

Type:	string
Description:	The version of a component.

3.3 BIAS Operations

3.3.1 Primitive Operations

3.3.1.1 AddSubjectToGallery

[AddSubjectToGalleryResponse](#)

AddSubjectToGallery([AddSubjectToGalleryRequest](#) addSubjectToGalleryRequest)

The AddSubjectToGallery operation registers a subject to a given gallery or population group. As an optional parameter, the value of the claim to identity by which the subject is known to the gallery may be specified. This claim to identity must be unique across the gallery. If no claim to identity is specified, the subject ID (assigned with the [CreateSubject](#) operation) will be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometrics that will be added to the gallery must be specified.

260 **Input**

Field	Type	#	?	Meaning
AddSubjectToGalleryRequest			Y	Register a subject to a given gallery or population group.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "AddSubjectToGallery".
↳ GalleryID	BIASIDType	1	Y	The identifier of the gallery or population group to which the subject will be added.
↳ Identity	BIASIdentity	1	Y	The identity to add to the gallery.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ IdentityClaim	BIASIDType	0..1	N	An identifier by which a subject is known to a particular gallery or population group.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

261 **Output**

Field	Type	#	?	Meaning
AddSubjectToGalleryResponse			Y	The response to an AddSubjectToGallery operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.

3.3.1.2 CheckQuality

[CheckQualityResponse](#)

CheckQuality([CheckQualityRequest](#) checkQualityRequest)

The CheckQuality operation returns a quality score for a given biometric. The biometric input is provided in a CBEFF basic structure or CBEFF record, which in this specification is called a CBEFF-BIR. The algorithm vendor and algorithm vendor product ID may be optionally provided in order to request a particular algorithm's use in calculating the biometric quality. If an algorithm vendor is provided then the algorithm vendor product ID is required. If no algorithm vendor is provided, the implementing system will provide the algorithm vendor and algorithm vendor product ID that were used to calculate the biometric quality as output parameters.

Input

Field	Type	#	?	Meaning
CheckQualityRequest			Y	Calculate a quality score for a given biometric.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "CheckQuality".
↳ BiometricData	BIASBiometricDataType	1	Y	Data structure containing a single biometric sample for which a quality score is to be determined.
↳ BIR	CBEFF_BIR_Type	1	Y	The biometric sample.
↳ Quality	QualityData	0..1	N	Specifies a particular algorithm vendor and vender product ID.

Field	Type	#	?	Meaning
AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.

273 Output

Field	Type	#	?	Meaning
CheckQualityResponse			Y	The response to a CheckQuality operation.
ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
Message	string	0..1	N	A short message corresponding to the return code.
QualityInfo	QualityData	1	Y	Contains the quality information for the submitted biometric sample.
QualityScore	Score	0..1	N	The quality of a biometric sample.
AlgorithmVendor	VendorIdentifier	1	Y	The vendor of the quality algorithm used to determine the quality score.
AlgorithmVendorProductID	ProductID	1	Y	The vendor's ID for the algorithm used to determine the quality.
AlgorithmVersion	VersionType	1	Y	The version of the algorithm used to determine the quality.

274 3.3.1.3 ClassifyBiometricData

275 [ClassifyBiometricDataResponse](#)

276 ClassifyBiometricData([ClassifyBiometricDataRequest](#) classifyBiometricDataRequest)

277 The ClassifyBiometricData operation attempts to classify a biometric sample. For example, a fingerprint
278 biometric sample may be classified as a whorl, loop, or arch (or other classification classes and sub-
279 classes).

280 To obtain the types of classification algorithms and classes, see the [QueryCapabilities](#) operation.

281 Input

Field	Type	#	?	Meaning
ClassifyBiometricDataRequest			Y	Classifies a biometric sample.

Field	Type	#	?	Meaning
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ClassifyBiometricData".
↑ BiometricData	BIASBiometricDataType	1	Y	Data structure containing a single biometric sample for which the classification is to be determined.
↑ BIR	CBEFF BIR Type	1	Y	The biometric sample.

282

Output

Field	Type	#	?	Meaning
ClassifyBiometricDataResponse			Y	The response to a ClassifyBiometricData operation, containing the classification of a biometric sample.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ClassificationData	ClassificationData	1	Y	Information on the results and type of classification performed.
↑ Classification	Classification	1	Y	The result of the classification.
↑ ClassificationAlgorithmType	ClassificationAlgorithmType	1	Y	Identifies the type of classification algorithm that was used to perform the classification.

3.3.1.4 CreateSubject

[CreateSubjectResponse](#)

CreateSubject([CreateSubjectRequest](#) createSubjectRequest)

The CreateSubject operation creates a new subject record and associates a subject ID to that record. As an optional parameter, the subject ID may be specified by the caller. If no subject ID is specified, the CreateSubject operation will generate one.

Input

Field	Type	#	?	Meaning
CreateSubjectRequest			Y	
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "CreateSubject".
↑ Identity	BIASIdentity	0..1	N	
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

Output

Field	Type	#	?	Meaning
CreateSubjectResponse			Y	The response to a CreateSubject operation, containing the subject ID of the new subject record.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	BIASIdentity	1	Y	
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

3.3.1.5 DeleteBiographicData

[DeleteBiographicDataResponse](#)

DeleteBiographicData([DeleteBiographicDataRequest](#) deleteBiographicDataRequest)

294 The DeleteBiographicData operation erases all of the biographic data associated with a given subject
 295 record. In the encounter-centric model the operation erases all of the biographic data associated with a
 296 given encounter, and therefore the encounter ID must be specified.

297 When deleting data, BIAS implementations may completely erase the information in order to prevent the
 298 ability to reconstruct a record in whole or in part, or they may track and record the deleted information for
 299 auditing and/or quality control purposes.

300 Input

Field	Type	#	?	Meaning
DeleteBiographicDataRequest			Y	Erase all of the biographic data associated with a given subject record or, in the encounter-centric model, with a given encounter.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteBiographicData"
↳ Identity	BIASIdentity	1	Y	
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

301 Output

Field	Type	#	?	Meaning
DeleteBiographicDataResponse			Y	The response to a DeleteBiographicData operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↶ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↶ Message	string	0..1	N	A short message corresponding to the return code.

3.3.1.6 DeleteBiometricData

[DeleteBiometricDataResponse](#)

DeleteBiographicData([DeleteBiometricDataRequest](#) deleteBiographicDataRequest)

The DeleteBiometricData operation erases all of the biometric data associated with a given subject record. In the encounter-centric model the operation erases all of the biometric data associated with a given encounter, and therefore the encounter ID must be specified.

When deleting data, BIAS implementations may completely erase the information in order to prevent the ability to reconstruct a record in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

Input

Field	Type	#	?	Meaning
DeleteBiometricDataRequest			Y	Erase all of the biometric data associated with a given subject record or, in the encounter-centric model, with a given encounter.
↶ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↶ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↶ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↶ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteBiometricData".
↶ Identity	BIASIdentity	1	Y	
↶ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

Field	Type	#	?	Meaning
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

312 Output

Field	Type	#	?	Meaning
DeleteBiometricDataResponse			Y	The response to a DeleteBiometricData operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.

313 3.3.1.7 DeleteSubject

314 [DeleteSubjectResponse](#)

315 DeleteSubject([DeleteSubjectRequest](#) deleteSubjectRequest)

316 The DeleteSubject operation deletes an existing subject record and, in an encounter-centric model, any
 317 associated encounter information from the system. This operation also removes the subject from any
 318 registered galleries.

319 When deleting a subject, BIAS implementations may completely erase the subject information in order to
 320 prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the
 321 deleted information for auditing and/or quality control purposes.

322 Input

Field	Type	#	?	Meaning
DeleteSubjectRequest			Y	Delete an existing subject record and, in an encounter-centric model, any associated encounter information.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	BIASIdentity	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.

Field	Type	#	?	Meaning
⤴ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteSubject".
⤴ Identity	BIASIdentity	1	Y	The identity of the subject to delete.
⤴ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.

323 Output

Field	Type	#	?	Meaning
DeleteSubjectResponse			Y	The response to a DeleteSubject operation.
⤴ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
⤴ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
⤴ Message	string	0..1	N	A short message corresponding to the return code.

324 3.3.1.8 DeleteSubjectFromGallery

325 [DeleteSubjectFromGalleryResponse](#)

326 DeleteSubjectFromGallery([DeleteSubjectFromGalleryRequest](#) deleteSubjectFromGalleryRequest)

327 The DeleteSubjectFromGallery operation removes the registration of a subject from a gallery or
 328 population group. The subject is identified by either the subject ID or the claim to identity that was
 329 specified in the [AddSubjectToGallery](#) operation.

330 Input

Field	Type	#	?	Meaning
DeleteSubjectFromGalleryRequest			Y	Remove the registration of a subject from a gallery or population group.
⤴ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
⤴ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
⤴ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.

Field	Type	#	?	Meaning
↗ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "DeleteSubjectFromGallery".
↗ GalleryID	BIASIDType	1	Y	The identifier of the gallery or population group from which the subject will be deleted.
↗ Identity	BIASIdentity	1	Y	The identity to remove from the gallery.
↗ SubjectID	BIASIDType	0..1	C	A system unique identifier for a subject. Required if an Identity Claim is not provided.
↗ IdentityClaim	BIASIDType	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Subject ID is not provided.

331 Output

Field	Type	#	?	Meaning
DeleteSubjectFromGalleryResponse			Y	The response to a DeleteSubjectFromGallery operation.
↗ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↗ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↗ Message	string	0..1	N	A short message corresponding to the return code.

332 3.3.1.9 GetIdentifySubjectResults

333 [GetIdentifySubjectResultsResponse](#)

334 GetIdentifySubjectResults([GetIdentifySubjectResultsRequest](#) getIdentifySubjectResultsRequest)

335 The GetIdentifySubjectResults operation retrieves the identification results for the specified token. This
336 operation is used in conjunction with the [IdentifySubject](#) operation. If the IdentifySubject operation is
337 implemented as an asynchronous service, the implementing system returns a token and the
338 GetIdentifySubjectResults operation is used to poll for the results of the original IdentifySubject request.

339 **Input**

Field	Type	#	?	Meaning
GetIdentifySubjectResultsRequest			Y	Retrieve the identification results for a specified token, which was returned by the IdentifySubject operation.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetIdentifySubjectResults".
↳ Token	TokenType	1	Y	A value used to retrieve the results of an IdentifySubject request.
↳ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↳ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

340 **Output**

Field	Type	#	?	Meaning
GetIdentifySubjectResultsResponse			Y	The response to a GetIdentifySubjectResults operation, which includes a candidate list.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↗ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↗ Message	string	0..1	N	A short message corresponding to the return code.
↗ CandidateList	CandidateListType	1	Y	A rank-ordered list of candidates that have a likelihood of matching the input biometric sample.
↗ Candidate	CandidateType	0..*	N	A single candidate.
↗ Score	string	0..1	N	The match score.
↗ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↗ BIRList	CBEFF_BIR_ListType	1	Y	Biometric data associated with the candidate match.
↗ BIR	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

3.3.1.10 IdentifySubject

[IdentifySubjectResponse](#)

IdentifySubject([IdentifySubjectRequest](#) identifySubjectRequest)

The IdentifySubject operation performs an identification search against a given gallery for a given biometric, returning a rank-ordered candidate list of a given maximum size.

If the IdentifySubject operation is implemented as a synchronous service, the implementing system immediately processes the request and returns the results in the candidate list. If the IdentifySubject operation is implemented as an asynchronous service, the implementing system returns a token, which is an indication that the request is being handled asynchronously. In this case, the [GetIdentifySubjectResults](#) operation is used to poll for the results of the IdentifySubject request.

Input

Field	Type	#	?	Meaning
IdentifySubjectRequest			Y	Perform an identification search against a given gallery for a given biometric.
↗ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↗ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.

Field	Type	#	?	Meaning
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "IdentifySubject".
↑ GalleryID	BIASIDType	1	Y	The identifier of the gallery or population group which will be searched.
↑ Identity	BIASIdentity	1	Y	Contains the BIR, a data structure containing the biometric sample for the search.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ BIR	CBEFF_BIR_Type	1	Y	Contains biometric information in either a non-XML or an XML representation.
↑ MaxListSize	integer	1	Y	The maximum size of the candidate list that should be returned.

352

Output

Field	Type	#	?	Meaning
IdentifySubjectResponse			Y	The response to an IdentifySubject operation, returning a rank-ordered candidate list.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ CandidateList	CandidateListType	0..1	C	A rank-ordered list of candidates that have a likelihood of matching the input biometric sample. Returned with successful synchronous request processing.

Field	Type	#	?	Meaning
↑ Candidate	CandidateType	0..*	N	A single candidate.
↑ Score	string	0..1	N	The match score.
↑ BiographicData	BiographicDataType	0..1	N	Biographic data associated with the candidate match.
↑ BIRList	CBEFF BIR ListType	1	Y	Biometric data associated with the candidate match.
↑ BIR	CBEFF BIR Type	0..*	N	CBEFF structure containing information about a biometric sample.
↑ Token	TokenType	0..1	C	A token used to retrieve the results of the IdentifySubject operation. Returned with asynchronous request processing.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

3.3.1.11 ListBiographicData

[ListBiographicDataResponse](#)

ListBiographicData([ListBiographicDataRequest](#) listBiographicDataRequest)

The ListBiographicData operation lists the biographic data elements stored for a subject using the Biographic Data Elements output parameter. Note that no actual biographic data is returned by this operation (see the [RetrieveBiographicInformation](#) operation to obtain the biographic data). In the encounter-centric model, an encounter ID may be specified to indicate that only the biographic data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the operation returns the list of encounter IDs which contain biographic data using the Encounter List output parameter, and the Biographic Data Elements output parameter is empty.

Input

Field	Type	#	?	Meaning
ListBiographicDataRequest			Y	Lists the biographic data elements stored for a subject.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.

Field	Type	#	?	Meaning
↗ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↗ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↗ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ListBiographicData".
↗ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↗ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↗ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.

364 Output

Field	Type	#	?	Meaning
ListBiographicDataResponse			Y	The response to a ListBiographicData request, containing a list of biographic data elements stored for a subject. In the encounter-centric model, the biographic data elements for a specific encounter are returned. If an encounter ID is not specified and encounter data exists for the subject, the list of encounter IDs which contain biographic data is returned.
↗ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↗ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.

Field	Type	#	?	Meaning
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ Identity	BIASIdentity	1	Y	Contains a list of biographic data elements associated with a subject or encounter; non-empty if the service was successful, biographic data exists, and either (a) the person-centric model is being used or (b) the encounter-centric model is being used and an encounter identifier was specified.
↳ BiographicDataElements	BiographicDataType	0..1	C	An Identity's biographic data elements that are stored in the implementing system.
↳ BiographicDataItem	BiographicDataItemType	0..*	N	A single biographic data element.
↳ Name	string	1	Y	The name of the biographic data item.
↳ Type	string	1	Y	The data type for the biographic data item.
↳ EncounterList	EncounterListType	0..1	C	A list of encounter ID's associated with a subject and which contain biographic data; non-empty if the service was successful, biographic data exists, the encounter-centric model is being used, and an encounter identifier was not specified.
↳ EncounterID	BIASIDType	0..*	N	The identifier of an encounter.

3.3.1.12 ListBiometricData

[ListBiometricDataResponse](#)

ListBiometricData([ListBiometricDataRequest](#) listBiometricDataRequest)

368 The ListBiometricData operation lists the biometric data elements stored for a subject using the Biometric
 369 Data List output parameter. Note that no actual biometric data is returned by this operation (see the
 370 [RetrieveBiometricInformation](#) operation to obtain the biometric data). In the encounter-centric model, an
 371 encounter ID may be specified to indicate that only the biometric data elements stored for that encounter
 372 should be returned. If an encounter ID is not specified and encounter data exists for the subject, the
 373 operation returns the list of encounter IDs which contain biometric data using the Encounter List output
 374 parameter, and the Biometric Data List output parameter is empty.

375 An optional parameter may be used to indicate a filter on the list of returned data. Such a filter may
 376 indicate that only biometric types should be listed (e.g., face, finger, iris, etc.) or that only biometric
 377 subtypes for a particular biometric type should be listed (e.g., all fingerprints: left slap, right index, etc.). If
 378 a filter is not specified, all biometric type and biometric subtype information are listed (e.g., left index
 379 finger, right iris, face frontal, etc.).

380 Input

Field	Type	#	?	Meaning
ListBiometricDataRequest			Y	Lists the biometric data elements stored for a subject.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "ListBiometricData".
↳ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
↳ ListFilterType	ListFilterType	0..1	N	Indicates what biometric information should be returned.

Field	Type	#	?	Meaning
↗ BiometricTypeFilter	hexBinary	1..*	Y	Limits the returned information to a specific type of biometric, as defined by CBEFF.
↗ IncludeBiometricSubType	boolean	1	Y	A Boolean flag indicating if biometric subtype information should be returned.

381 Output

Field	Type	#	?	Meaning
ListBiometricDataResponse			Y	The response to a ListBiometricData operation, containing a list of biometric data elements stored for a subject. In the encounter-centric model, the biometric data elements for a specific encounter are returned. If an encounter ID is not specified and encounter data exists for the subject, the list of encounter IDs which contain biometric data is returned.
↗ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↗ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↗ Message	string	0..1	N	A short message corresponding to the return code.

Field	Type	#	?	Meaning
↑ Identity	BIASIdentity	0..1	N	Includes a list of biometric data elements associated with a subject or encounter or a list of encounter ID's associated with a subject and which contain biometric data.
↑ BiometricData	BIASBiometricDataType	0..1	C	An Identity's biometric data.
↑ BiometricDataList	BiometricDataListType	0..1	N	A list of biometric data elements.
↑ BiometricDataElement	BiometricDataElementType	1..*	Y	Data structure containing information about a biometric record.
↑ BiometricType	hexBinary	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↑ BiometricTypeCount	positiveInteger	0..1	N	The number of biometric records having the biometric type recorded in the biometric type field.
↑ BiometricSubType	hexBinary	0..1	N	More specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF.

Field	Type	#	?	Meaning
↳ BDBFormatOwner	string	1	Y	Identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data.
↳ BDBFormatType	string	1	Y	Identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field.
↳ EncounterList	EncounterList Type	0..1	C	A list of encounter ID's associated with a subject and which contain biometric data; non-empty if the service was successful, biometric data exists, the encounter-centric model is being used, and an encounter identifier was not specified.
↳ EncounterID	BIASIDType	1..*	Y	The identifier of an encounter.

3.3.1.13 PerformFusion

[PerformFusionResponse](#)

PerformFusion([PerformFusionRequest](#) performFusionRequest)

The PerformFusion operation accepts either match score or match decision information and creates a fused match result. The [FusionInformationListType](#), through the [FusionInformationType](#), provides specific elements for match score input and match decision input. The fusion method and processes are left to the implementing system.

Field	Type	#	?	Meaning
PerformFusionRequest			Y	Accepts either match score or match decision information and creates a fused match result.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "PerformFusion".
↳ FusionInput	FusionInformationListType	1	Y	Score or decision input information to the fusion method.
↳ FusionElement	FusionInformationType	2..*	Y	A set of fusion information.
↳ BiometricType	hexBinary	1	Y	The type of biological or behavioral data stored in the biometric record, as defined by CBEFF.
↳ BiometricSubType	string	0..1	N	More specifically defines the type of biometric data stored in the biometric record.
↳ AlgorithmOwner	string	1	Y	The owner or vendor of the algorithm used to determine the score or decision.

Field	Type	#	?	Meaning
AlgorithmType	string	1	Y	The Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision.
Score	unsignedLong	0..1	C	The similarity score assigned by the matching algorithm. Either a Score or a Decision element must be used.
Decision	string	0..1	C	The match decision assigned by the matching algorithm. Either a Score or a Decision element must be used.

390 Output

Field	Type	#	?	Meaning
PerformFusionResponse			Y	The response to the PerformFusion operation.
ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
Message	string	0..1	N	A short message corresponding to the return code.
Match	MatchType	1	1	Indicates the result of the fusion method.

391 3.3.1.14 QueryCapabilities

392 [QueryCapabilitiesResponse](#)

393 QueryCapabilities([QueryCapabilitiesRequest](#) queryCapabilitiesRequest)

394 The QueryCapabilities operation returns a list of the capabilities, options, galleries, etc. that are supported
395 by the BIAS implementation. Refer to Annex A in the INCITS BIAS standard for conformance
396 requirements regarding which capability names an implementation must use in the QueryCapabilities
397 operation.

398 **Input**

Field	Type	#	?	Meaning
QueryCapabilitiesRequest			Y	Returns a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "QueryCapabilities".

399 **Output**

Field	Type	#	?	Meaning
QueryCapabilitiesResponse			Y	The response to a QueryCapabilities operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ CapabilityList	CapabilityListType	1	Y	A list of capabilities supported by the BIAS implementation.
↳ Capability	CapabilityType	0..*	N	A single capability.
↳ CapabilityName	CapabilityName	1	Y	The name of the capability.

Field	Type	#	?	Meaning
↳ CapabilityID	string	0..1	N	An identifier assigned to the capability by the implementing system.
↳ CapabilityDescription	string	0..1	N	A description of the capability.
↳ CapabilityValue	string	0..1	N	A value assigned to the capability.
↳ CapabilitySupportingValue	string	0..1	N	A secondary value supporting the capability.

3.3.1.15 RetrieveBiographicInformation

[RetrieveBiographicInformationResponse](#)

RetrieveBiographicInformation([RetrieveBiographicInformationRequest](#) retrieveBiographicInfoRequest)

The RetrieveBiographicInformation operation retrieves the biographic data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the operation will return the biographic data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the operation returns the biographic information associated with the most recent encounter.

Input

Field	Type	#	?	Meaning
RetrieveBiographicInformationRequest			Y	Retrieves the biographic data associated with a subject ID.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveBiographicInformation".

Field	Type	#	?	Meaning
↑ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↑ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.

409 Output

Field	Type	#	?	Meaning
RetrieveBiographicInformationResponse			Y	The response to a RetrieveBiographicInformation operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	BIASIdentity	1	Y	Includes the set of biographic data associated with a subject.
↑ BiographicData	BiographicDataType	1	Y	An Identity's biographic data. One of the following elements must be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.
↑ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

3.3.1.16 RetrieveBiometricInformation

[RetrieveBiometricInformationResponse](#)

RetrieveBiometricInformation([RetrieveBiometricInformationRequest](#) retrieveBiometricInformationRequest)

The RetrieveBiometricInformation operation retrieves the biometric data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the operation will return the biometric data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the operation returns the biometric information associated with the most recent encounter. The operation provides an optional input parameter to specify that only biometric data of a certain type should be retrieved.

Input

Field	Type	#	?	Meaning
RetrieveBiometricInformationRequest			Y	Retrieves the biometric data associated with a subject ID.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveBiometricInformation".
↑ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter.
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↑ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
↑ BiometricType	string	0..1	N	The type of biological or behavioral data to retrieve.

420 Output

Field	Type	#	?	Meaning
RetrieveBiometricInformationResponse			Y	The response to a RetrieveBiometricInformation operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	BIASIdentity	1	Y	Includes the biometric data associated with a subject.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ BIRList	CBEFF_BIR_ListType	1	Y	A list of CBEFF-BIR elements.
↑ BIR	CBEFF_BIR_Type	0..*	N	CBEFF structure containing information about a biometric sample.

421 3.3.1.17 SetBiographicData

422 [SetBiographicDataResponse](#)

423 SetBiographicData([SetBiographicDataRequest](#) setBiographicDataRequest)

424 The SetBiographicData operation associates biographic data to a given subject record. The Identity
 425 Model parameter in the request indicates whether the biographic information should replace any existing
 426 biographic information (person-centric model) or if a new encounter should be created and associated
 427 with the subject (encounter-centric model). For encounter-centric models, the encounter ID may be
 428 specified by the caller in order to link biographic and biometric information (assuming biometric
 429 information was previously associated using the [SetBiometricData](#) operation). If the encounter ID is
 430 omitted for the encounter-centric model, the operation returns a system-assigned encounter ID.

431 Input

Field	Type	#	?	Meaning
SetBiographicDataRequest			Y	Associates biographic data to a given subject record.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.

Field	Type	#	?	Meaning
↗ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↗ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↗ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "SetBiographicData".
↗ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to store.
↗ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↗ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
↗ BiographicData	BiographicDataType	1	Y	An Identity's biographic data. One of the following elements must be present.
↗ LastName	string	0..1	C	The last name of a subject.
↗ FirstName	string	0..1	C	The first name of a subject.
↗ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.
↗ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.
↗ IdentityModel	IdentityModelType	1	Y	Indicates a person-centric or encounter-centric model.

432 Output

Field	Type	#	?	Meaning
SetBiographicDataResponse			Y	The response to a SetBiographicData operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	BIASIdentity	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↑ EncounterID	BIASIDType	1	Y	The identifier of an encounter associated with the subject.

433 3.3.1.18 SetBiometricData

434 [SetBiometricDataResponse](#)

435 SetBiometricData([SetBiometricDataRequest](#) setBiometricDataRequest)

436 The SetBiometricData operation associates biometric data to a given subject record. The Identity Model
 437 parameter in the request indicates whether the biometric information should replace any existing
 438 biometric information (person-centric model) or if a new encounter should be created and associated with
 439 the subject (encounter-centric model). For encounter-centric models, the encounter ID may be specified
 440 by the caller in order to link biographic and biometric information (assuming biographic information was
 441 previously associated using the [SetBiographicData](#) operation). If the encounter ID is omitted for the
 442 encounter-centric model, the operation returns a system-assigned encounter ID.

443 Input

Field	Type	#	?	Meaning
SetBiometricDataRequest			Y	Associates biometric data to a given subject record.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "SetBiometricData".

Field	Type	#	?	Meaning
↑ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to store.
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↑ EncounterID	BIASIDType	0..1	N	The identifier of an encounter associated with the subject.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ BIRList	CBEFF BIR ListType	1	Y	A list of CBEFF-BIR elements.
↑ BIR	CBEFF BIR Type	1..*	Y	CBEFF structure containing information about a biometric sample.
↑ IdentityModel	IdentityModelType	1	Y	Indicates a person-centric or encounter-centric model.

444 Output

Field	Type	#	?	Meaning
SetBiometricDataResponse			Y	The response to a SetBiometricData operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Identity	BIASIdentity	0..1	C	In an encounter-centric model, identifies the encounter ID assigned to a new encounter.
↑ EncounterID	BIASIDType	1	Y	The identifier of an encounter associated with the subject.

445 3.3.1.19 TransformBiometricData

446 [TransformBiometricDataResponse](#)

447 TransformBiometricData([TransformBiometricDataRequest](#) transformBiometricDataRequest)

448 The TransformBiometricData operation transforms or processes a given biometric in one format into a
449 new target format.

450 **Input**

Field	Type	#	?	Meaning
TransformBiometricDataRequest			Y	Transforms or processes a given biometric in one format into a new target format.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "TransformBiometricData".
↳ InputBIR	CBEFF_BIR_Type	1	Y	Data structure containing the biometric information to be transformed.
↳ TransformOperation	unsignedLong	1	Y	Value indicating the type of transformation to perform.

451 **Output**

Field	Type	#	?	Meaning
TransformBiometricDataResponse			Y	The response to a TransformBiometricData operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ OutputBIR	CBEFF_BIR_Type	0..1	N	Data structure containing the new, transformed biometric information.

3.3.1.20 UpdateBiographicData

[UpdateBiographicDataResponse](#)

UpdateBiographicData([UpdateBiographicDataRequest](#) updateBiographicDataRequest)

The UpdateBiographicData operation updates the biographic data for an existing subject record. The operation replaces any existing biographic data with the new biographic data. In the encounter-centric model, the encounter ID must be specified.

Input

Field	Type	#	?	Meaning
UpdateBiographicDataRequest			Y	Updates the biographic data for a given subject record.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "UpdateBiographic Data".
↳ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biographic data to update.
↳ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.

Field	Type	#	?	Meaning
↑ BiographicData	BiographicDataType	1	Y	An Identity's biographic data. One of the following elements must be present.
↑ LastName	string	0..1	C	The last name of a subject.
↑ FirstName	string	0..1	C	The first name of a subject.
↑ BiographicDataItem	BiographicDataItemType	0..*	C	A single biographic data element.
↑ BiographicDataSet	BiographicDataSetType	0..1	C	A set of biographic data information.

459 Output

Field	Type	#	?	Meaning
UpdateBiographicDataResponse			Y	The response to an UpdateBiographicData operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.

460 3.3.1.21 UpdateBiometricData

461 [UpdateBiometricDataResponse](#)

462 UpdateBiometricData([UpdateBiometricDataRequest](#) updateBiometricDataRequest)

463 The UpdateBiometricData operation updates the biometric data for an existing subject record. The
 464 operation includes an optional parameter indicating if the new biometric sample should be merged with
 465 the existing biometric sample. If this parameter is set to "False" or is not used in the request, the
 466 operation replaces the existing biometric sample with the new biometric sample. In the encounter-centric
 467 model, the encounter ID must be specified.

468 Input

Field	Type	#	?	Meaning
UpdateBiometricDataRequest			Y	Updates a single biometric sample for a given subject record.

Field	Type	#	?	Meaning
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "UpdateBiometricData".
↑ Identity	BIASIdentity	1	Y	Identifies the subject or, in the encounter-centric model, a subject and an encounter, and includes the biometric data to update.
↑ SubjectID	BIASIDType	1	Y	A system unique identifier for a subject.
↑ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required for encounter-centric models.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ BIR	CBEFF_BIR_Type	1	Y	Contains biometric information in either a non-XML or an XML representation.
↑ Merge	boolean	0..1	N	Value indicating if the input biometric sample should be merged with any existing biometric information.

469 Output

Field	Type	#	?	Meaning
UpdateBiometricDataResponse			Y	The response to an UpdateBiometricData operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
⤴ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
⤴ Message	string	0..1	N	A short message corresponding to the return code.

3.3.1.22 VerifySubject

[VerifySubjectResponse](#)

VerifySubject([VerifySubjectRequest](#) verifySubjectRequest)

The VerifySubject operation performs a 1:1 verification match between a given biometric and either a claim to identity in a given gallery or another given biometric. As such either the Identity Claim or Reference BIR input parameters are required.

Input

Field	Type	#	?	Meaning
VerifySubjectRequest			Y	Performs a 1:1 verification match between a given biometric and either a claim to identity in a given gallery or another given biometric.
⤴ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
⤴ Application	BIASIdentity	0..1	N	Identifies the requesting application.
⤴ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
⤴ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "VerifySubject".
⤴ GalleryID	BIASIDType	0..1	C	The identifier of the gallery or population group of which the subject must be a member. Required if an Identity Claim is provided.
⤴ Identity	BIASIdentity	1	Y	Includes the identifying information and/or input and reference biometric samples.

Field	Type	#	?	Meaning
↑ IdentityClaim	BIASIDType	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Reference BIR is not provided.
↑ BiometricData	BIASBiometricDataType	1	Y	An Identity's biometric data.
↑ InputBIR	CBEFF BIR Type	1	Y	Maps to specific INCITS BIAS elements as required by that specification.
↑ ReferenceBIR	CBEFF BIR Type	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.

477 Output

Field	Type	#	?	Meaning
VerifySubjectResponse			Y	The response to a VerifySubject operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
↑ Score	Score	0..1	N	The score if the biometric information matched.

478 3.3.2 Aggregate Operations

479 3.3.2.1 Enroll

480 [EnrollResponse](#)

481 Enroll([EnrollRequest](#) enrollRequest)

482 The Enroll operation adds a new subject or, in an encounter-centric model, a new encounter to the
483 system. This may be accomplished in a number of different ways according to system requirements
484 and/or resources. If the Enroll operation is implemented as a synchronous service, the implementing

485 system immediately processes the request and returns the results in the Return Data parameter. If the
 486 Enroll operation is implemented as an asynchronous service, the implementing system returns a token in
 487 the Return Data parameter, which is an indication that the request is being handled asynchronously. In
 488 this case, the [GetEnrollResults](#) operation is used to poll for the results of the Enroll request.

489 Input

Field	Type	#	?	Meaning
EnrollRequest			Y	Adds a new subject or, in an encounter-centric model, a new encounter to the system.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Enroll".
↳ ProcessingOptions	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.
↳ Option	string	0..*	N	An option supported by the implementing system.
↳ InputData	InformationType	1	Y	Contains the input data for the operation, as required by the implementing system.

490 Output

Field	Type	#	?	Meaning
EnrollResponse			Y	The response to an Enroll operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ ReturnData	InformationType	0..1	N	Contains the output data for the response.

3.3.2.2 GetEnrollResults

[GetEnrollResultsResponse](#)

GetEnrollResults([GetEnrollResultsRequest](#) getEnrollResultsRequest)

The GetEnrollResults operation retrieves the enrollment results for the specified token. This operation is used in conjunction with the [Enroll](#) operation. If the Enroll operation is implemented as an asynchronous service, the implementing system returns a token and the GetEnrollResults operation is used to poll for the results of the original Enroll request.

Input

Field	Type	#	?	Meaning
GetEnrollResultsRequest			Y	Retrieves the enrollment results for the specified token.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetEnrollResults".
↑ Token	TokenType	1	Y	A value used to retrieve the results of the Enroll request.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

Output

Field	Type	#	?	Meaning
GetEnrollResultsResponse			Y	The response to a GetEnrollResults operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.

Field	Type	#	?	Meaning
↰ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↰ Message	string	0..1	N	A short message corresponding to the return code.
↰ ReturnData	InformationType	0..1	N	Contains the output data for the response.

3.3.2.3 GetIdentifyResults

[GetIdentifyResultsResponse](#)

GetIdentifyResults([GetIdentifyResultsRequest](#) getIdentifyResultsRequest)

The GetIdentifyResults operation retrieves the identification results for the specified token. This operation is used in conjunction with the [Identify](#) operation. If the Identify operation is implemented as an asynchronous service, the implementing system returns a token and the GetIdentifyResults operation is used to poll for the results of the original Identify request.

Input

Field	Type	#	?	Meaning
GetIdentifyResultsRequest			Y	Retrieves the identification results for the specified token
↰ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↰ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↰ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↰ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetIdentifyResults".
↰ Token	TokenType	1	Y	A value used to retrieve the results of the Identify request.
↰ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↰ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed

Field	Type	#	?	Meaning
				to be available.

508 Output

Field	Type	#	?	Meaning
GetIdentifyResultsResponse			Y	The response to a GetIdentifyResults operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	InformationType	0..1	N	Contains the output data for the response.

509 3.3.2.4 GetVerifyResults

510 [GetVerifyResultsResponse](#)

511 GetVerifyResult([GetVerifyResultsRequest](#) getVerifyResultsRequest)

512 The GetVerifyResults operation retrieves the verification results for the specified token. This operation is
513 used in conjunction with the [Verify](#) operation. If the Verify operation is implemented as an asynchronous
514 service, the implementing system returns a token and the GetVerifyResults operation is used to poll for
515 the results of the original Verify request.

516 Input

Field	Type	#	?	Meaning
GetVerifyResultsRequest			Y	Retrieves the verification results for the specified token
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "GetVerifyResults".

Field	Type	#	?	Meaning
↑ Token	TokenType	1	Y	A value used to retrieve the results of the Verify request.
↑ TokenValue	string	1	Y	A value returned by the implementing system that is used to retrieve the results to an operation at a later time.
↑ Expiration	date	1	Y	A date and time at which point the token expires and the operation results are no longer guaranteed to be available.

517 Output

Field	Type	#	?	Meaning
GetVerifyResultsResponse			Y	The response to a GetVerifyResults operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	InformationType	0..1	N	Contains the output data for the response.
↑ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
↑ Score	Score	0..1	N	The score if the biometric information matched.

518 3.3.2.5 Identify

519 [IdentifyResponse](#)

520 Identify([IdentifyRequest](#) identifyRequest)

521 The Identify operation performs an identification function according to system requirements and/or
522 resources. If the Identify operation is implemented as a synchronous service, the implementing system
523 immediately processes the request and returns the results in the Return Data parameter. If the Identify
524 operation is implemented as an asynchronous service, the implementing system returns a token in the
525 Return Data parameter, which is an indication that the request is being handled asynchronously. In this
526 case, the [GetIdentifyResults](#) operation is used to poll for the results of the Identify request.

527 Input

Field	Type	#	?	Meaning
IdentifyRequest			Y	Performs an identification function.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Identify".
↳ ProcessingOptions	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.
↳ Option	string	0..*	N	An option supported by the implementing system.
↳ InputData	InformationType	1	Y	Contains the input data for the aggregate services.

528 Output

Field	Type	#	?	Meaning
IdentifyResponse			Y	The response to an Identify operation.
↳ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↳ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↳ Message	string	0..1	N	A short message corresponding to the return code.
↳ ReturnData	InformationType	0..1	N	Contains the output data for the response.

529 3.3.2.6 RetrieveInformation

530 [RetrieveInformationResponse](#)

531 RetrieveInformation([RetrieveInformationRequest](#) retrieveInformationRequest)

532 The RetrieveInformation operation retrieves requested information about a subject, or in an encounter-
533 centric model about an encounter. In a person-centric model, this operation may be used to retrieve both

534 biographic and biometric information for a subject record. In an encounter-centric model, this operation
 535 may be used to retrieve biographic and/or biometric information for either a single encounter or all
 536 encounters. Either a subject ID or encounter ID must be specified.

537 **Input**

Field	Type	#	?	Meaning
RetrieveInformationRequest			Y	Retrieves requested information about a subject or encounter.
↳ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↳ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↳ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↳ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "RetrieveInformation".
↳ ProcessingOptions	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed, and may identify what type(s) of information should be returned.
↳ Option	string	0..*	N	An option supported by the implementing system.
↳ Identity	BIASIdentity	1	Y	Includes the identifier of the subject or encounter.
↳ SubjectID	BIASIDType	0..1	C	A system unique identifier for a subject. Required if an Encounter ID is not provided.
↳ EncounterID	BIASIDType	0..1	C	The identifier of an encounter associated with the subject. Required if a Subject ID is not provided.

538 Output

Field	Type	#	?	Meaning
RetrieveInformationResponse			Y	Response to a RetrieveInformation operation.
↑ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↑ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↑ Message	string	0..1	N	A short message corresponding to the return code.
↑ ReturnData	InformationType	0..1	N	Contains the output data for the response.

539 3.3.2.7 Verify

540 [VerifyResponse](#)

541 Verify([VerifyRequest](#) verifyRequest)

542 The Verify operation performs a 1:1 verification function according to system requirements and/or
543 resources. Either the Identity Claim or Reference BIR input parameters are required. If the Verify
544 operation is implemented as a synchronous service, the implementing system immediately processes the
545 request and returns the results in the Return Data parameter. If the Verify operation is implemented as an
546 asynchronous service, the implementing system returns a token in the Return Data parameter, which is
547 an indication that the request is being handled asynchronously. In this case, the [GetVerifyResults](#)
548 operation is used to poll for the results of the Verify request.

549 Input

Field	Type	#	?	Meaning
VerifyRequest			Y	Performs a 1:1 verification function.
↑ GenericRequestParameters	GenericRequestParameters	0..1	N	Common request parameters that can be used to identify the requester.
↑ Application	ApplicationIdentifier	0..1	N	Identifies the requesting application.
↑ ApplicationUser	ApplicationUserIdentifier	0..1	N	Identifies the user or instance of the requesting application.
↑ BIASOperationName	string	0..1	N	Identifies the BIAS operation that is being requested: "Verify".
↑ ProcessingOptions	ProcessingOptionsType	1	Y	Options that guide how the aggregate service request is processed.

Field	Type	#	?	Meaning
↗ Option	string	0..*	N	An option supported by the implementing system.
↗ InputData	InformationType	1	Y	Contains the input data for the aggregate services.
↗ Identity	BIASIdentity	1	Y	Includes either the Identity Claim or Reference BIR.
↗ IdentityClaim	BIASIDType	0..1	C	An identifier by which a subject is known to a particular gallery or population group. Required if a Reference BIR is not provided.
↗ BiometricData	BIASBiometricDataType	0..1	N	An Identity's biometric data.
↗ ReferenceBIR	CBEFF BIR Type	0..1	C	Maps to specific INCITS BIAS elements as required by that specification. Required if an Identity Claim is not provided.
↗ GalleryID	BIASIDType	0..1	C	The identifier of the gallery or population group of which the subject must be a member. Required if an Identity Claim is provided.

550 Output

Field	Type	#	?	Meaning
VerifyResponse			Y	The response to a Verify operation.
↗ ResponseStatus	ResponseStatus	1	Y	Returned status for the operation.
↗ Return	ReturnCode	1	Y	The return code indicates the return status of the operation.
↗ Message	string	0..1	N	A short message corresponding to the return code.
↗ ReturnData	InformationType	0..1	N	Contains the output data for the response.

Field	Type	#	?	Meaning
↑ Match	boolean	0..1	N	Indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR.
↑ Score	Score	0..1	N	The score if the biometric information matched.

551

4 Message structure and rules

BIAS operations and data elements are defined in XML in the INCITS 422 BIAS standard. This OASIS standard further specifies the full XML schema (see Appendix B) and specifies how this XML is packaged and exchanged as SOAP messages.

SOAP 1.1 messages consist of three elements: an envelope, header data, and a message body. BIAS request-response protocol elements MUST be enclosed within the SOAP message body. The general structure of the BIAS SOAP message is shown in Figure 4, below. The data model for BIAS is addressed in Chapter 4.

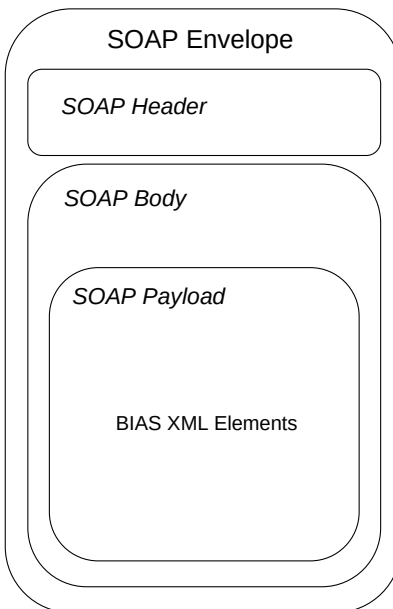


Figure 4. BIAS SOAP Structure

Biometric data, regardless of native format, is carried as a binary structure. As such, options exist on how this data is carried within the SOAP structure. It can be carried as embedded Base-64 objects or [XOP] can be used – this standard allows for either method.

4.1 Purpose and constraints

This document defines a SOAP profile describing how the XML elements defined in INCITS 442 are to be used as the payload of a SOAP message and the rules for structuring and exchanging such a messages. Philosophical tenets include:

- SOAP messages will carry BIAS XML payloads.
- SOAP messages will follow WS-I and will deviate only when absolutely necessary.
- Message structures and interchanges will be kept as simple as possible – “nice to have” features will be addressed in future revisions.
- XML schemas will be produced based on INCITS 442.
- BIAS will support a broad range of application domains.
- BIAS will allow for a variety of biometric and biographic data formats to be used
- Only the SOAP messaging will be defined – no message protocols or client/server agents will be defined.
- Basic usage/formatting rules (beyond WS-I) will be defined.

- Existing biometric and Web services standards will be leveraged wherever possible.
- Sample WSDL and use cases will be provided as an aid in implementation.
- Use of basic SOAP will allow all other compatible WS* standards (and discovery mechanisms) to be used in conjunction with BIAS messaging.
- BIAS will support both secure (i.e., using existing security mechanisms such as WS-Security, SAML, etc.) and non-secure implementations.
- Generic biometric operations will be defined – use of biometrics within a Web services authentication protocol is not addressed.
- OASIS namespace rules will be followed, though some external schemas may also be referenced.

4.2 Message requirements

BIAS SOAP messages MUST conform to [WS-I-Basic] and [WS-I-Bind]. A single BIAS SOAP message MUST contain only one BIAS service request (or single BIAS service response). Binary components of BIAS messages are already Base-64 encoded and therefore do not need to be conveyed as SOAP attachments.

The system model used for BIAS conversations over SOAP is a simple request-response model. BIAS comprises both synchronous and asynchronous operations, with the majority being of the former type. Asynchronous operations are implemented through message pairs. That is, there are separate messages to request the operation and to request the results of the operation. These have been defined for those operations that are likely to take significant time to complete. For example, an identify operation may be implemented as either a synchronous or asynchronous service as follows:

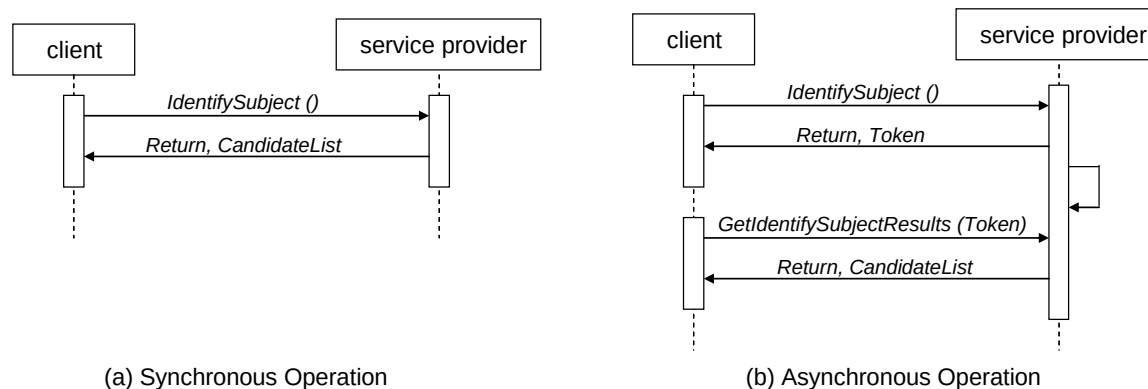


Figure 5. Example of Synchronous and Asynchronous BIAS Operations

The basic process for using SOAP for BIAS operations is:

- A system entity acting as a BIAS requester transmits a BIAS request element within the body of a SOAP message to a system entity acting as a BIAS responder. The BIAS requester MUST NOT include more than one BIAS request per SOAP message or include any additional XML elements in the SOAP body.
- The BIAS responder MUST return either a BIAS response element within the body of another SOAP message or generate a SOAP fault. The BIAS responder MUST NOT include more than one BIAS response per SOAP message or include any additional XML elements in the SOAP body. If a BIAS responder cannot, for some reason, process a BIAS request, it MUST generate a SOAP fault. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] Section 4.1.)
- On receiving a BIAS response in a SOAP message, the BIAS requester MUST NOT send a fault code or other error messages to the BIAS responder. Since the format for the message

617 interchange is a simple request-response pattern, adding additional items such as error
618 conditions would needlessly complicate the protocol.

619 SOAP 1.1 also defines an optional data encoding system. This system is not used within the BIAS SOAP
620 binding. This means that BIAS messages can be transported using SOAP without re-encoding from the
621 "standard" BIAS schema to one based on the SOAP encoding.

622 NOTE: [SOAP11] references an early draft of the XML Schema specification including an
623 obsolete namespace. BIAS requesters SHOULD generate SOAP documents referencing only the
624 final XML schema namespace. BIAS responders MUST be able to process both the XML schema
625 namespace used in [SOAP11] as well as the final XML schema namespace.

626 4.3 Identifying operations

627 Receivers of BIAS SOAP messages require a method of easily identifying the operation being requested
628 (or response being provided). This should be possible without the receiver needing to infer it from the
629 sum of the elements provided within the body of the SOAP message. The BIAS SOAP profile allows for
630 two methods of identifying BIAS operations:

- 631 • Explicit named element in body of the SOAP message
- 632 • Use of WS-Addressing Action element

633 4.3.1 Operation name element

634 The BIAS message sender (requester) will include within the body of the BIAS SOAP message an XML
635 element <BIASOperationName>. The receiver (service provider) can search for this tag within a received
636 BIAS SOAP message to determine what operation is being requested. There is no requirement related to
637 the ordering of this element within the message, though it is recommended that it be included early in the
638 message to aid in human readability. This same approach would be used in response messages.

639 An example of this method for the CreateSubject operation is shown below:

```
640
641 POST /bias HTTP/1.1
642 Host: www.acme.com
643 Content-Type: application/soap+xml; charset="utf-8"
644 Content-Length: nnnn
645 SOAPAction: ""
646 <?xml version="1.0"?>
647 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
648   <soap:Body>
649     <tns:CreateSubjectRequest
650       xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
651       <tns:GenericRequestParameters>
652         <tns:Application>BIAS Application</tns:Application>
653         <tns:ApplicationUser>BIAS User</tns:ApplicationUser>
654         <tns:BIASOperationName>CreateSubject</tns:BIASOperationName>
655       >
656       </tns:GenericRequestParameters>
657       <tns:Identity>
658         <tns:SubjectID>123456789</tns:SubjectID>
659       </tns:Identity>
660     </tns:CreateSubjectRequest>
```

661 </soap:Body>

662 </soap:Envelope>

663 4.3.2 WS-Addressing Action

664 WS-Addressing [WS-Addr] provides a mechanism for including action information inside any SOAP
665 message. The information is in the SOAP Header. The WS-Addressing 'Action' element is used to
666 indicate the intent of the message. The value is a URI identifying that intent; however, there are no
667 restrictions on the format or specificity of the URI nor a requirement that it can be resolved. Adoption of
668 this option also requires that the WS-Addressing 'To', 'ReplyTo', and 'MessageID' elements are supplied,
669 as they are mandatory elements in a request-reply message pattern as used within BIAS. Response
670 messages would also need to use WS-Addressing, requiring the 'To' (matching the 'ReplyTo' element in
671 the request), 'RelatesTo' (matching the 'MessageID' element in the request), and 'RelationshipType'
672 (default value to "wsa:Reply") elements.

673 Use of WS-Addressing is optional in this profile as is this method of using the 'Action' field for this
674 purpose. However, when BIAS is used within an environment using WS-Addressing, it is recommended
675 that this approach for use of the 'Action' field to carry the BIAS operation name is employed, either alone
676 or in combination with the BIASOperationName approach described in 5.3.1.

677 An example for a message request for the CreateSubject operation would look like the following:

678

```
679     POST /bias HTTP/1.1
680     Host: www.acme.com
681     Content-Type: application/soap+xml; charset="utf-8"
682     Content-Length: nnnn
683     SOAPAction: ""
684
685     <?xml version="1.0"?>
686     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
687                   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
688         <soap:Header>
689             <wsa:MessageID>some-ID</wsa:MessageID>
690             <wsa:ReplyTo>
691                 <wsa:Address>response-URI</wsa:Address>
692             </wsa:ReplyTo>
693             <wsa:To>destination-URI</wsa:To>
694             <wsa:Action>CreateSubject</wsa:Action>
695         </soap:Header>
696         <soap:Body>
697             <tns:CreateSubjectRequest
698
699             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/" />
700         </soap:Body>
701     </soap:Envelope>
702
```

4.4 Security

The end-points that exchange SOAP messages (or handle the contents of the BIAS operations) are expected to be protected and trusted such that message-level security mechanisms are not required. The use of SSL (HTTPS) or VPN technology that provides end-point to end-point security should be sufficient. Other mechanisms such as Signed XML or WSS could be implemented. Unless stated otherwise, the following security statements apply to all BIAS bindings.

4.4.1 Use of SSL 3.0 or TLS 1.0

Unless otherwise specified, in any BIAS binding's use of SSL 3.0 [SSL3] or TLS1.0 [RFC2246], servers MUST authenticate clients using a X.509 v3 certificate [X509]. The client MUST establish server identity based on contents of the certificate (typically through examination of the certificate's subject DN field, subjectAltName attribute, etc.). Use of transport level security in the form of SSL or TLS is optional but highly recommended. Use of these mechanisms alone may not be sufficient for end-to-end integrity and confidentiality, however (see 5.4.3 and 5.4.4 below).

4.4.2 Data Origin Authentication

Authentication of both the BIAS requester and the BIAS responder associated with a message is OPTIONAL and depends on the environment of use: Authentication mechanisms available at the SOAP message exchange layer or from the underlying substrate protocol (for example, in many bindings the SSL/TLS or HTTP protocol) MAY be utilized to provide data origin authentication. Transport authentication will not meet end-to-end origin authentication requirements in bindings where the BIAS SOAP message passes through an intermediary – in this case, message authentication is recommended. Note that SAML [SAML] may be used as the mechanism for parties to authenticate to one another.

4.4.3 Message Integrity

Message integrity of both BIAS requests and BIAS responses is OPTIONAL and depends on the environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP message exchange layer MAY be used to ensure message integrity. Transport integrity will not meet end-to-end integrity requirements in bindings where the BIAS SOAP message passes through an intermediary – in this case, message integrity is recommended.

4.4.4 Message Confidentiality

Message confidentiality of both BIAS requests and BIAS responses is OPTIONAL and depends on the environment of use. The security layer in the underlying substrate protocol or a mechanism at the SOAP message exchange layer MAY be used to ensure message confidentiality. Transport confidentiality will not meet end-to-end confidentiality requirements in bindings where the BIAS SOAP message passes through an intermediary.

4.4.5 Security Considerations

Before deployment, each combination of authentication, message integrity, and confidentiality mechanisms SHOULD be analyzed for vulnerability in the context of the specific protocol exchange and the deployment environment. IETF RFC 2617 [RFC2617] describes possible attacks in the HTTP environment when basic or message digest authentication schemes are used. Special care should be given to the impact of possible caching on security.

745 **4.4.6 Security of Stored Data**

746 This specification does not address security considerations for stored data. It is the purview of the BIAS
747 service provider to implement security mechanisms and protect data at rest as per their own security
748 policies.

749 **4.4.7 Key Management**

750 This specification does not address key management considerations with respect to implementation of
751 cryptographic security mechanisms (e.g., for authenticity, integrity, or confidentiality).

752 **4.5 Use with other WS* standards**

753 The intent of specifying SOAP bindings for BIAS messages is to enable the full range of existing Web
754 services standards to be able to be applied. Some may be normative while others can be optionally
755 applied (i.e., WS-Security). Still others may require additional profiling to be used in an interoperable
756 manner (e.g., WS-Notification); this is left to a future revision. However, the intent is to avoid specifying
757 anything in the first, base version that would preclude the use of such standards in the future.

758 **4.6 Tailoring**

759 This standard provides for a common method of implementing biometric Web services; however, it does
760 not guaranteed interoperability in a specific application. In some cases further tailoring or profiling of this
761 standard may be required in order to further constrain the implementation options available.

5 Error handling

There are two levels of errors that can be returned in an error response: system and service errors.

- System-level errors occur when the implementing system cannot service a request. They could result due to an internal logic error or because the implementing system does not support a particular request.
- Service-level errors occur when there is a problem transmitting or representing the service request. They could result due to an invalid service request or because of a communications error.

The INCITS BIAS standard defines the error condition codes for system-level errors.

Every Web service (operation) described in the BIAS WSDL may result in a fault message that will be returned in the response by the service provider in the event of an error. The fault message contains a FaultCode element as defined by the SOAP 1.1 specification (see section 4.2.5). The fault message shall contain a Detail element in a common format, as described by the BIASFault element (see section 4.2.6).

The schema provided in Annex B defines “BIASFaultCode” and “BIASFaultDetail” types as well as “BIASFault”, “BIASFaultType”, “BIASFaultMessage” and “BIASFaultDescription” elements.

6 Conformance

Annex A specifies the conformance requirements for systems/components claiming conformance to this standard.

{Ed note: This section is included pending conclusion of an open action item on this subject. See also <http://docs.oasis-open.org/templates/TCHandbook/ConformanceGuidelines.html> related to this clause.}

A. XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://docs.oasis-open.org/bias/ns/bias-1.0/"
  targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
      targetNamespace="http://docs.oasis-open.org/bias/ns/bias-1.0/">
      <xsd:complexType name="AggregateRequestTemplate" abstract="true">
        <xsd:annotation>
          <xsd:documentation>Base template for BIAS aggregate service requests.</xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
          <xsd:extension base="RequestTemplate">
            <xsd:sequence>
              <xsd:element name="ProcessingOptions" type="ProcessingOptionsType" minOccurs="1" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Options that guide how the aggregate service request is
processed.</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="InputData" type="InformationType" minOccurs="1" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Contains the input data for the aggregate service request.</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
            </xsd:sequence>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="AggregateResponseTemplate" abstract="true">
        <xsd:annotation>
          <xsd:documentation>Base template for BIAS aggregate service responses.</xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
          <xsd:extension base="tns:ResponseTemplate">
            <xsd:sequence>
              <xsd:element name="ReturnData" type="tns:InformationType" minOccurs="0" maxOccurs="1">
                <xsd:annotation>
                  <xsd:documentation>Contains the output data for the aggregate service response.</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
            </xsd:sequence>
          </xsd:extension>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:simpleType name="ApplicationIdentifier">
        <xsd:annotation>
```



```

833     <xsd:documentation>Identifies an application.</xsd:documentation>
834 </xsd:annotation>
835 <xsd:restriction base="xsd:string">
836     <xsd:minLength value="1" />
837     <xsd:maxLength value="255" />
838 </xsd:restriction>
839 </xsd:simpleType>
840 <xsd:simpleType name="ApplicationUserIdentifier">
841     <xsd:annotation>
842         <xsd:documentation>Identifies an application user or instance.</xsd:documentation>
843     </xsd:annotation>
844     <xsd:restriction base="xsd:string">
845         <xsd:minLength value="1" />
846         <xsd:maxLength value="255" />
847     </xsd:restriction>
848 </xsd:simpleType>
849 <xsd:complexType name="BDB-info">
850     <xsd:annotation>
851         <xsd:documentation>
852             For a description or definition of each data element, see the
853             referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
854         </xsd:documentation>
855     </xsd:annotation>
856     <xsd:sequence>
857         <xsd:element name="challenge-response" type="xsd:string" minOccurs="0" maxOccurs="1" />
858         <xsd:element name="index" type="xsd:string" minOccurs="0" maxOccurs="1" />
859         <xsd:element name="format-owner" type="xsd:integer" minOccurs="1" maxOccurs="1" />
860         <xsd:element name="format-type" type="xsd:integer" minOccurs="1" maxOccurs="1" />
861         <xsd:element name="security-Encryption-options" type="xsd:string" minOccurs="1" maxOccurs="1" />
862         <xsd:element name="creation-date" type="xsd:string" minOccurs="0" maxOccurs="1" />
863         <xsd:element name="validity-period" type="xsd:string" minOccurs="0" maxOccurs="1" />
864         <xsd:element name="type" type="xsd:string" minOccurs="0" maxOccurs="1" />
865         <xsd:element name="subtype" type="xsd:string" minOccurs="0" maxOccurs="1" />
866         <xsd:element name="data-type" type="xsd:string" minOccurs="0" maxOccurs="1" />
867         <xsd:element name="product-owner" type="xsd:integer" minOccurs="1" maxOccurs="1" />
868         <xsd:element name="product-type" type="xsd:integer" minOccurs="1" maxOccurs="1" />
869         <xsd:element name="purpose" type="xsd:string" minOccurs="0" maxOccurs="1" />
870         <xsd:element name="quality" minOccurs="0" maxOccurs="1">
871             <xsd:complexType>
872                 <xsd:sequence>
873                     <xsd:element name="value" type="xsd:string" minOccurs="1" maxOccurs="1" />
874                     <xsd:element name="algorithm-owner" type="xsd:string" minOccurs="0" maxOccurs="1" />
875                     <xsd:element name="algorithm-type" type="xsd:string" minOccurs="0" maxOccurs="1" />
876                 </xsd:sequence>
877             </xsd:complexType>
878         </xsd:element>
879         <xsd:element name="device-owner" type="xsd:integer" minOccurs="0" maxOccurs="1" />
880         <xsd:element name="device-type" type="xsd:integer" minOccurs="0" maxOccurs="1" />
881     </xsd:sequence>
882 </xsd:complexType>
883 <xsd:complexType name="BIASBiometricDataType">
884     <xsd:annotation>
885         <xsd:documentation>Wraps the various BIAS biometric types.</xsd:documentation>

```

```

886     </xsd:annotation>
887     <xsd:sequence>
888         <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="0" maxOccurs="1">
889             <xsd:annotation>
890                 <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
891             </xsd:annotation>
892         </xsd:element>
893         <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
894             <xsd:annotation>
895                 <xsd:documentation>Contains biometric information in either a non-XML and an XML
896 representation.</xsd:documentation>
897             </xsd:annotation>
898         </xsd:element>
899         <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
900             <xsd:annotation>
901                 <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
902 specification.</xsd:documentation>
903             </xsd:annotation>
904         </xsd:element>
905         <xsd:element name="ReferenceBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
906             <xsd:annotation>
907                 <xsd:documentation>Maps to specific INCITS BIAS elements as required by that
908 specification.</xsd:documentation>
909             </xsd:annotation>
910         </xsd:element>
911         <xsd:element name="BiometricDataList" type="tns:BiometricDataListType" minOccurs="0" maxOccurs="1">
912             <xsd:annotation>
913                 <xsd:documentation>A list of biometric data elements.</xsd:documentation>
914             </xsd:annotation>
915         </xsd:element>
916     </xsd:sequence>
917 </xsd:complexType>
918 <xsd:simpleType name="BIASFaultCode">
919     <xsd:annotation>
920         <xsd:documentation></xsd:documentation>
921     </xsd:annotation>
922     <xsd:restriction base="xsd:unsignedLong">
923         <xsd:enumeration value="UNKNOWN_ERROR">
924             <xsd:annotation>
925                 <xsd:documentation>The service failed for an unknown reason.</xsd:documentation>
926             </xsd:annotation>
927         </xsd:enumeration>
928         <xsd:enumeration value="UNSUPPORTED_CAPABILITY">
929             <xsd:annotation>
930                 <xsd:documentation>A requested capability is not supported by the service
931 implementation.</xsd:documentation>
932             </xsd:annotation>
933         </xsd:enumeration>
934         <xsd:enumeration value="INVALID_INPUT">
935             <xsd:annotation>
936                 <xsd:documentation>The data in a service input parameter is invalid.</xsd:documentation>
937             </xsd:annotation>
938         </xsd:enumeration>

```

```

939     <xsd:enumeration value="BIR_QUALITY_ERROR">
940     <xsd:annotation>
941         <xsd:documentation> Biometric sample quality is too poor for the service to succeed.</xsd:documentation>
942     </xsd:annotation>
943 </xsd:enumeration>
944 <xsd:enumeration value="INVALID_BIR">
945     <xsd:annotation>
946         <xsd:documentation> The input BIR is empty or in an invalid or unrecognized format.</xsd:documentation>
947     </xsd:annotation>
948 </xsd:enumeration>
949 <xsd:enumeration value="BIR_SIGNATURE_FAILURE">
950     <xsd:annotation>
951         <xsd:documentation> The service could not validate the signature, if used, on the input
952 BIR.</xsd:documentation>
953     </xsd:annotation>
954 </xsd:enumeration>
955 <xsd:enumeration value="BIR_DECRYPTION_FAILURE">
956     <xsd:annotation>
957         <xsd:documentation> The service could not decrypt an encrypted input BIR.</xsd:documentation>
958     </xsd:annotation>
959 </xsd:enumeration>
960 <xsd:enumeration value="INVALID_ENCOUNTER_ID">
961     <xsd:annotation>
962         <xsd:documentation> The input encounter ID is empty or in an invalid format.</xsd:documentation>
963     </xsd:annotation>
964 </xsd:enumeration>
965 <xsd:enumeration value="INVALID_SUBJECT_ID">
966     <xsd:annotation>
967         <xsd:documentation> The input subject ID is empty or in an invalid format.</xsd:documentation>
968     </xsd:annotation>
969 </xsd:enumeration>
970 <xsd:enumeration value="UNKNOWN_SUBJECT">
971     <xsd:annotation>
972         <xsd:documentation> The subject referenced by the input subject ID does not exist.</xsd:documentation>
973     </xsd:annotation>
974 </xsd:enumeration>
975 <xsd:enumeration value="UNKNOWN_GALLERY">
976     <xsd:annotation>
977         <xsd:documentation> The gallery referenced by the input gallery ID does not exist.</xsd:documentation>
978     </xsd:annotation>
979 </xsd:enumeration>
980 <xsd:enumeration value="UNKNOWN_ENCOUNTER">
981     <xsd:annotation>
982         <xsd:documentation> The encounter referenced by the input encounter ID does not
983 exist.</xsd:documentation>
984     </xsd:annotation>
985 </xsd:enumeration>
986 </xsd:restriction>
987 </xsd:simpleType>
988 <xsd:complexType name="BIASFaultDetail">
989     <xsd:annotation>
990         <xsd:documentation> Defines the error information associated with a SOAP fault.</xsd:documentation>
991     </xsd:annotation>

```

```

992     <xsd:sequence>
993     <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultType" type="tns:BIASFaultCode">
994         <xsd:annotation>
995             <xsd:documentation>References an error code.</xsd:documentation>
996         </xsd:annotation>
997     </xsd:element>
998     <xsd:element minOccurs="1" maxOccurs="1" name="BIASFaultMessage" type="xsd:string">
999         <xsd:annotation>
1000             <xsd:documentation>Provides an explanation of the fault.</xsd:documentation>
1001         </xsd:annotation>
1002     </xsd:element>
1003     <xsd:element minOccurs="0" maxOccurs="1" name="BIASFaultDescription" type="xsd:string">
1004         <xsd:annotation>
1005             <xsd:documentation>Provides detailed information about a BIAS fault, such as trace
1006 details.</xsd:documentation>
1007         </xsd:annotation>
1008     </xsd:element>
1009 </xsd:sequence>
1010 </xsd:complexType>
1011 <xsd:complexType name="BIASIdentity">
1012     <xsd:annotation>
1013         <xsd:documentation>
1014             Defines a single element for encapsulating the data associated
1015             with an Identity. Includes the Identity's reference identifiers,
1016             biographic data, and biometric data.
1017         </xsd:documentation>
1018     </xsd:annotation>
1019     <xsd:sequence>
1020         <xsd:element name="SubjectID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1021             <xsd:annotation>
1022                 <xsd:documentation>A system unique identifier for a subject.</xsd:documentation>
1023             </xsd:annotation>
1024         </xsd:element>
1025         <xsd:element name="IdentityClaim" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1026             <xsd:annotation>
1027                 <xsd:documentation>An identifier by which a subject is known to a particular gallery or population
1028 group.</xsd:documentation>
1029             </xsd:annotation>
1030         </xsd:element>
1031         <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
1032             <xsd:annotation>
1033                 <xsd:documentation>The identifier of an encounter associated with the subject, required for encounter-
1034 centric models.</xsd:documentation>
1035             </xsd:annotation>
1036         </xsd:element>
1037         <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
1038             <xsd:annotation>
1039                 <xsd:documentation>A list of encounters associated with a subject.</xsd:documentation>
1040             </xsd:annotation>
1041         </xsd:element>
1042         <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1043             <xsd:annotation>
1044                 <xsd:documentation>An Identity's biographic data.</xsd:documentation>

```

```

1045     </xsd:annotation>
1046 </xsd:element>
1047 <xsd:element name="BiographicDataElements" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1048   <xsd:annotation>
1049     <xsd:documentation>An Identity's biographic data elements that are stored in the implementing
1050 system.</xsd:documentation>
1051   </xsd:annotation>
1052 </xsd:element>
1053 <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="0" maxOccurs="1">
1054   <xsd:annotation>
1055     <xsd:documentation>An Identity's biometric data.</xsd:documentation>
1056   </xsd:annotation>
1057 </xsd:element>
1058 </xsd:sequence>
1059 </xsd:complexType>
1060 <xsd:simpleType name="BIASIDType">
1061   <xsd:annotation>
1062     <xsd:documentation>A BIAS identifier</xsd:documentation>
1063   </xsd:annotation>
1064   <xsd:restriction base="xsd:string" />
1065 </xsd:simpleType>
1066 <xsd:complexType name="BiographicDataItemType">
1067   <xsd:annotation>
1068     <xsd:documentation>Defines a single biographic data element.</xsd:documentation>
1069   </xsd:annotation>
1070   <xsd:sequence maxOccurs="unbounded">
1071     <xsd:element name="Name" type="xsd:string" minOccurs="1" maxOccurs="1">
1072       <xsd:annotation>
1073         <xsd:documentation>The name of the biographic data item.</xsd:documentation>
1074       </xsd:annotation>
1075     </xsd:element>
1076     <xsd:element name="Type" type="xsd:string" minOccurs="1" maxOccurs="1">
1077       <xsd:annotation>
1078         <xsd:documentation>The data type for the biographic data item.</xsd:documentation>
1079       </xsd:annotation>
1080     </xsd:element>
1081     <xsd:element name="Value" type="xsd:string" minOccurs="0" maxOccurs="1">
1082       <xsd:annotation>
1083         <xsd:documentation>The value assigned to the biographic data item.</xsd:documentation>
1084       </xsd:annotation>
1085     </xsd:element>
1086   </xsd:sequence>
1087 </xsd:complexType>
1088 <xsd:complexType name="BiographicDataSetType">
1089   <xsd:annotation>
1090     <xsd:documentation>Defines a set of biographic data that is formatted according to the specified
1091 format.</xsd:documentation>
1092   </xsd:annotation>
1093   <xsd:sequence>
1094     <xsd:any namespace="##any">
1095       <xsd:annotation>
1096         <xsd:documentation>Biographic data formatted according to a specific format.</xsd:documentation>
1097       </xsd:annotation>

```

```

1098     </xsd:any>
1099 </xsd:sequence>
1100 <xsd:attribute name="name" type="xsd:string" use="required">
1101     <xsd:annotation>
1102         <xsd:documentation>The name of the biographic data format. Use these names for common formats: EFTS,
1103 EBTS, NIEM, xNAL, HR-XML.</xsd:documentation>
1104     </xsd:annotation>
1105 </xsd:attribute>
1106 <xsd:attribute name="version" type="xsd:string" use="optional">
1107     <xsd:annotation>
1108         <xsd:documentation>The version of the biographic data format. Use these versions for common formats: (EFTS)
1109 7.1, (EBTS) 1.2, (NIEM) 1.0 or 2.0, (xNAL) 2.0 or 3.0, (HR-XML) 2.5.</xsd:documentation>
1110     </xsd:annotation>
1111 </xsd:attribute>
1112 <xsd:attribute name="source" type="xsd:string" use="required">
1113     <xsd:annotation>
1114         <xsd:documentation>Reference to a URI describing the biographic data format. Use these sources for common
1115 formats: (EFTS) www.fbi.gov, (EBTS) www.biometrics.dod.mil, (NIEM) www.niem.gov, (xNAL) www.oasis-open.org,
1116 (HR-XML) www.hr-xml.org.</xsd:documentation>
1117     </xsd:annotation>
1118 </xsd:attribute>
1119 <xsd:attribute name="type" type="xsd:string" use="required">
1120     <xsd:annotation>
1121         <xsd:documentation>The biographic data format type. Use these types for common formats: (EFTS and EBTS)
1122 ASCII, (NIEM, xNAL, and HR-XML) XML.</xsd:documentation>
1123     </xsd:annotation>
1124 </xsd:attribute>
1125 </xsd:complexType>
1126 <xsd:complexType name="BiographicDataType">
1127     <xsd:annotation>
1128         <xsd:documentation>
1129             Defines a set of biographic data elements, utilizing either the
1130             BiographicDataItemType to represent a list of elements or the
1131             BiographicDataSetType to represent a complete, formatted set of
1132             biographic information.
1133         </xsd:documentation>
1134     </xsd:annotation>
1135     <xsd:choice maxOccurs="unbounded">
1136         <xsd:element name="LastName" type="xsd:string" minOccurs="0" maxOccurs="1">
1137             <xsd:annotation>
1138                 <xsd:documentation>The last name of a subject.</xsd:documentation>
1139             </xsd:annotation>
1140         </xsd:element>
1141         <xsd:element name="FirstName" type="xsd:string" minOccurs="0" maxOccurs="1">
1142             <xsd:annotation>
1143                 <xsd:documentation>The first name of a subject.</xsd:documentation>
1144             </xsd:annotation>
1145         </xsd:element>
1146         <xsd:element name="BiographicDataItem" type="tns:BiographicDataItemType" maxOccurs="unbounded">
1147             <xsd:annotation>
1148                 <xsd:documentation>A single biographic data element.</xsd:documentation>
1149             </xsd:annotation>
1150         </xsd:element>

```

```

1151     <xsd:element name="BiographicDataSet" type="tns:BiographicDataSetType" maxOccurs="1">
1152         <xsd:annotation>
1153             <xsd:documentation>A set of biographic data information.</xsd:documentation>
1154         </xsd:annotation>
1155     </xsd:element>
1156 </xsd:choice>
1157 </xsd:complexType>
1158 <xsd:complexType name="BiometricDataElementType">
1159     <xsd:annotation>
1160         <xsd:documentation>
1161             Provides descriptive information about biometric data, such as
1162             the biometric type, subtype, and format, contained in the BDB of
1163             the CBEFF-BIR.
1164         </xsd:documentation>
1165     </xsd:annotation>
1166     <xsd:sequence>
1167         <xsd:element name="BiometricType" type="xsd:hexBinary" minOccurs="1" maxOccurs="1">
1168             <xsd:annotation>
1169                 <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1170 CBEFF.</xsd:documentation>
1171             </xsd:annotation>
1172         </xsd:element>
1173         <xsd:element name="BiometricTypeCount" type="xsd:positiveInteger" minOccurs="0" maxOccurs="1">
1174             <xsd:annotation>
1175                 <xsd:documentation>The number of biometric records having the biometric type recorded in the biometric
1176 type field.</xsd:documentation>
1177             </xsd:annotation>
1178         </xsd:element>
1179         <xsd:element name="BiometricSubType" type="xsd:hexBinary" minOccurs="0" maxOccurs="1">
1180             <xsd:annotation>
1181                 <xsd:documentation>More specifically defines the type of biometric data stored in the biometric record, as
1182 defined by CBEFF.</xsd:documentation>
1183             </xsd:annotation>
1184         </xsd:element>
1185         <xsd:element name="BDBFormatOwner" type="xsd:string" minOccurs="1" maxOccurs="1">
1186             <xsd:annotation>
1187                 <xsd:documentation>Identifies the standards body, working group, industry consortium, or other CBEFF
1188 biometric organization that has defined the format for the biometric data.</xsd:documentation>
1189             </xsd:annotation>
1190         </xsd:element>
1191         <xsd:element name="BDBFormatType" type="xsd:string" minOccurs="1" maxOccurs="1">
1192             <xsd:annotation>
1193                 <xsd:documentation>Identifies the specific biometric data format specified by the CBEFF biometric
1194 organization recorded in the BDB Format Owner field.</xsd:documentation>
1195             </xsd:annotation>
1196         </xsd:element>
1197     </xsd:sequence>
1198 </xsd:complexType>
1199 <xsd:complexType name="BiometricDataListType">
1200     <xsd:annotation>
1201         <xsd:documentation>A list of biometric data elements.</xsd:documentation>
1202     </xsd:annotation>
1203     <xsd:sequence>

```

```

1204     <xsd:element name="BiometricDataElement" type="tns:BiometricDataElementType" minOccurs="0"
1205 maxOccurs="unbounded">
1206     <xsd:annotation>
1207     <xsd:documentation>Data structure containing information about a biometric record.</xsd:documentation>
1208     </xsd:annotation>
1209     </xsd:element>
1210 </xsd:sequence>
1211 </xsd:complexType>
1212 <xsd:complexType name="BIR-info">
1213 <xsd:annotation>
1214 <xsd:documentation>
1215     For a description or definition of each data element, see the
1216     referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
1217 </xsd:documentation>
1218 </xsd:annotation>
1219 <xsd:sequence>
1220 <xsd:element name="creator" type="xsd:string" minOccurs="0" maxOccurs="1" />
1221 <xsd:element name="index" type="xsd:string" minOccurs="0" maxOccurs="1" />
1222 <xsd:element name="payload" type="xsd:anyType" minOccurs="0" maxOccurs="1" />
1223 <xsd:element name="integrity-options" type="xsd:string" minOccurs="0" maxOccurs="1" />
1224 <xsd:element name="creation-date" type="xsd:string" minOccurs="0" maxOccurs="1" />
1225 <xsd:element name="validity-period" type="xsd:string" minOccurs="0" maxOccurs="1" />
1226 <xsd:element name="patron-format-owner" type="xsd:string" minOccurs="0" maxOccurs="1" />
1227 <xsd:element name="patron-format-type" type="xsd:string" minOccurs="0" maxOccurs="1" />
1228 </xsd:sequence>
1229 </xsd:complexType>
1230 <xsd:complexType name="CandidateListType">
1231 <xsd:annotation>
1232 <xsd:documentation>
1233     Defines a set of candidates, utilizing the Candidate Type to
1234     represent each element in the set.
1235 </xsd:documentation>
1236 </xsd:annotation>
1237 <xsd:sequence>
1238 <xsd:element name="Candidate" type="tns:CandidateType" minOccurs="0" maxOccurs="unbounded">
1239 <xsd:annotation>
1240 <xsd:documentation>A single candidate.</xsd:documentation>
1241 </xsd:annotation>
1242 </xsd:element>
1243 </xsd:sequence>
1244 </xsd:complexType>
1245 <xsd:complexType name="CandidateType">
1246 <xsd:annotation>
1247 <xsd:documentation>
1248     Defines a single candidate as a possible match in response to a
1249     biometric identification request.
1250 </xsd:documentation>
1251 </xsd:annotation>
1252 <xsd:sequence>
1253 <xsd:element name="Score" type="xsd:string" minOccurs="0" maxOccurs="1">
1254 <xsd:annotation>
1255 <xsd:documentation>The match score.</xsd:documentation>
1256 </xsd:annotation>

```



```

1257     </xsd:element>
1258     <xsd:element name="BiographicData" type="tns:BiographicDataType" minOccurs="0" maxOccurs="1">
1259       <xsd:annotation>
1260         <xsd:documentation>Biographic data associated with the candidate match.</xsd:documentation>
1261       </xsd:annotation>
1262     </xsd:element>
1263     <xsd:element name="BIRList" type="tns:CBEFF_BIR_ListType" minOccurs="1" maxOccurs="1">
1264       <xsd:annotation>
1265         <xsd:documentation>Biometric data associated with the candidate match.</xsd:documentation>
1266       </xsd:annotation>
1267     </xsd:element>
1268   </xsd:sequence>
1269   <xsd:attribute name="rank" type="xsd:integer" use="required">
1270     <xsd:annotation>
1271       <xsd:documentation>The rank of the candidate in relation to other candidates for the same biometric
1272 identification operation.</xsd:documentation>
1273     </xsd:annotation>
1274   </xsd:attribute>
1275 </xsd:complexType>
1276 <xsd:complexType name="CapabilityListType">
1277   <xsd:annotation>
1278     <xsd:documentation>Defines a set of capabilities.</xsd:documentation>
1279   </xsd:annotation>
1280   <xsd:sequence>
1281     <xsd:element name="Capability" type="tns:CapabilityType" minOccurs="0" maxOccurs="unbounded">
1282       <xsd:annotation>
1283         <xsd:documentation>A single capability.</xsd:documentation>
1284       </xsd:annotation>
1285     </xsd:element>
1286   </xsd:sequence>
1287 </xsd:complexType>
1288 <xsd:simpleType name="CapabilityName">
1289   <xsd:annotation>
1290     <xsd:documentation>A list of capability items.</xsd:documentation>
1291   </xsd:annotation>
1292   <xsd:restriction base="xsd:string">
1293     <xsd:enumeration value="AggregateInputDataOptional">
1294       <xsd:annotation>
1295         <xsd:documentation>A data element accepted as optional input by the implementing system for the
1296 aggregate services.</xsd:documentation>
1297       </xsd:annotation>
1298     </xsd:enumeration>
1299     <xsd:enumeration value="AggregateInputDataRequired">
1300       <xsd:annotation>
1301         <xsd:documentation>A data element required as input by the implementing system for the aggregate
1302 services.</xsd:documentation>
1303       </xsd:annotation>
1304     </xsd:enumeration>
1305     <xsd:enumeration value="AggregateProcessingOption">
1306       <xsd:annotation>
1307         <xsd:documentation>A processing option supported by the implementing system for the aggregate
1308 services.</xsd:documentation>
1309       </xsd:annotation>

```

```

1310     </xsd:enumeration>
1311     <xsd:enumeration value="AggregateReturnData">
1312         <xsd:annotation>
1313             <xsd:documentation>A data element returned by the implementing system for the aggregate
1314 services.</xsd:documentation>
1315         </xsd:annotation>
1316     </xsd:enumeration>
1317     <xsd:enumeration value="AggregateServiceDescription">
1318         <xsd:annotation>
1319             <xsd:documentation>Describes the processing logic of an aggregate service supported by the implementing
1320 system.</xsd:documentation>
1321         </xsd:annotation>
1322     </xsd:enumeration>
1323     <xsd:enumeration value="BiographicDataSet">
1324         <xsd:annotation>
1325             <xsd:documentation>Identifies a biographic data set supported by the implementing
1326 system.</xsd:documentation>
1327         </xsd:annotation>
1328     </xsd:enumeration>
1329     <xsd:enumeration value="CBEFFPatronFormat">
1330         <xsd:annotation>
1331             <xsd:documentation>A patron format supported by the implementing system.</xsd:documentation>
1332         </xsd:annotation>
1333     </xsd:enumeration>
1334     <xsd:enumeration value="ClassificationAlgorithmType">
1335         <xsd:annotation>
1336             <xsd:documentation>A classification algorithm type supported by the implementing
1337 system.</xsd:documentation>
1338         </xsd:annotation>
1339     </xsd:enumeration>
1340     <xsd:enumeration value="ConformanceClass">
1341         <xsd:annotation>
1342             <xsd:documentation>Identifies the conformance class of the BIAS implementation.</xsd:documentation>
1343         </xsd:annotation>
1344     </xsd:enumeration>
1345     <xsd:enumeration value="Gallery">
1346         <xsd:annotation>
1347             <xsd:documentation>A gallery or population group supported by the implementing
1348 system.</xsd:documentation>
1349         </xsd:annotation>
1350     </xsd:enumeration>
1351     <xsd:enumeration value="IdentityModel">
1352         <xsd:annotation>
1353             <xsd:documentation>Identifies whether the implementing system is person-centric or encounter-centric
1354 based.</xsd:documentation>
1355         </xsd:annotation>
1356     </xsd:enumeration>
1357     <xsd:enumeration value="QualityAlgorithm">
1358         <xsd:annotation>
1359             <xsd:documentation>A quality algorithm vendor and algorithm vendor product ID supported by the
1360 implementing system.</xsd:documentation>
1361         </xsd:annotation>
1362     </xsd:enumeration>

```

```

1363     <xsd:enumeration value="SupportedBiometric">
1364     <xsd:annotation>
1365     <xsd:documentation>A biometric type supported by the implementing system.</xsd:documentation>
1366     </xsd:annotation>
1367     </xsd:enumeration>
1368     <xsd:enumeration value="TransformOperation">
1369     <xsd:annotation>
1370     <xsd:documentation>A transform operation type supported by the implementing
1371 system.</xsd:documentation>
1372     </xsd:annotation>
1373     </xsd:enumeration>
1374 </xsd:restriction>
1375 </xsd:simpleType>
1376 <xsd:complexType name="CapabilityType">
1377 <xsd:annotation>
1378 <xsd:documentation>Defines a single capability supported by an implementing system.</xsd:documentation>
1379 </xsd:annotation>
1380 <xsd:sequence>
1381 <xsd:element name="CapabilityName" type="tns:CapabilityName" minOccurs="1" maxOccurs="1">
1382 <xsd:annotation>
1383 <xsd:documentation>The name of the capability.</xsd:documentation>
1384 </xsd:annotation>
1385 </xsd:element>
1386 <xsd:element name="CapabilityID" type="xsd:string" minOccurs="0" maxOccurs="1">
1387 <xsd:annotation>
1388 <xsd:documentation>An identifier assigned to the capability by the implementing
1389 system.</xsd:documentation>
1390 </xsd:annotation>
1391 </xsd:element>
1392 <xsd:element name="CapabilityDescription" type="xsd:string" minOccurs="0" maxOccurs="1">
1393 <xsd:annotation>
1394 <xsd:documentation>A description of the capability.</xsd:documentation>
1395 </xsd:annotation>
1396 </xsd:element>
1397 <xsd:element name="CapabilityValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1398 <xsd:annotation>
1399 <xsd:documentation>A value assigned to the capability.</xsd:documentation>
1400 </xsd:annotation>
1401 </xsd:element>
1402 <xsd:element name="CapabilitySupportingValue" type="xsd:string" minOccurs="0" maxOccurs="1">
1403 <xsd:annotation>
1404 <xsd:documentation>A secondary value supporting the capability.</xsd:documentation>
1405 </xsd:annotation>
1406 </xsd:element>
1407 </xsd:sequence>
1408 </xsd:complexType>
1409 <xsd:complexType name="CBEFF_BIR_ListType">
1410 <xsd:annotation>
1411 <xsd:documentation>A list of CBEFF-BIR elements.</xsd:documentation>
1412 </xsd:annotation>
1413 <xsd:sequence>
1414 <xsd:element name="BIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="unbounded">
1415 <xsd:annotation>

```

```

1416     <xsd:documentation>CBEFF structure containing information about a biometric sample.</xsd:documentation>
1417   </xsd:annotation>
1418 </xsd:element>
1419 </xsd:sequence>
1420 </xsd:complexType>
1421 <xsd:complexType name="CBEFF_BIR_Type">
1422   <xsd:annotation>
1423     <xsd:documentation>Represents biometric information, with either a non-XML or XML
1424 representation.</xsd:documentation>
1425   </xsd:annotation>
1426   <xsd:choice minOccurs="1" maxOccurs="1">
1427     <xsd:any namespace="##any" />
1428     <xsd:element name="XML_BIR" type="tns:CBEFF_XML_BIR_Type" />
1429   </xsd:choice>
1430 </xsd:complexType>
1431 <xsd:complexType name="CBEFF_XML_BIR_Type">
1432   <xsd:annotation>
1433     <xsd:documentation>
1434       An XML representation for biometric information. This schema
1435       represents CBEFF data elements from both the US
1436       (INCITS 398-2005 Revision 1) and International
1437       (ISO/IEC 19785-1:2006) versions of the CBEFF standard. The data
1438       elements in the two standards are fairly similar. However, a data
1439       element may have a different meaning and/or a different set of
1440       valid values based on which standard is used. The cbeff-version
1441       data element specifies whether the US or International version is
1442       being used. An additional data element, other, has been added to
1443       this schema to capture non-standard information and to allow for
1444       flexibility to any future changes in the existing CBEFF standards.
1445       For a description or definition of each data element, see the
1446       referenced CBEFF standards.
1447     </xsd:documentation>
1448   </xsd:annotation>
1449   <xsd:sequence>
1450     <xsd:element name="patron-format-identifier" type="xsd:string" minOccurs="0" maxOccurs="1" />
1451     <xsd:element name="version" type="tns:Version" minOccurs="0" maxOccurs="1" />
1452     <xsd:element name="cbeff-version" type="tns:Version" minOccurs="0" maxOccurs="1" />
1453     <xsd:element name="bir-info" type="tns:BIR-info" minOccurs="0" maxOccurs="1" />
1454     <xsd:element name="bdb-info" type="tns:BDB-info" minOccurs="0" maxOccurs="1" />
1455     <xsd:element name="sb-info" type="tns:SB-info" minOccurs="0" maxOccurs="1" />
1456     <xsd:element name="other" minOccurs="0" maxOccurs="unbounded">
1457       <xsd:complexType>
1458         <xsd:sequence>
1459           <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
1460         </xsd:sequence>
1461       </xsd:complexType>
1462     </xsd:element>
1463     <xsd:element name="BIR" type="xsd:base64Binary" minOccurs="1" maxOccurs="1" />
1464   </xsd:sequence>
1465 </xsd:complexType>
1466 <xsd:simpleType name="Classification">
1467   <xsd:annotation>
1468     <xsd:documentation>The result of a classification.</xsd:documentation>

```

```

1469     </xsd:annotation>
1470     <xsd:restriction base="xsd:string" />
1471 </xsd:simpleType>
1472 <xsd:simpleType name="ClassificationAlgorithmType">
1473     <xsd:annotation>
1474         <xsd:documentation>Type of classification algorithm that was used to perform the
1475 classification.</xsd:documentation>
1476     </xsd:annotation>
1477     <xsd:restriction base="xsd:string" />
1478 </xsd:simpleType>
1479 <xsd:complexType name="ClassificationData">
1480     <xsd:annotation>
1481         <xsd:documentation>Contains information on classification results and the algorithm used to determine the
1482 classification.</xsd:documentation>
1483     </xsd:annotation>
1484     <xsd:sequence>
1485         <xsd:element name="Classification" type="tns:Classification" minOccurs="1" maxOccurs="1">
1486             <xsd:annotation>
1487                 <xsd:documentation>The result of the classification.</xsd:documentation>
1488             </xsd:annotation>
1489         </xsd:element>
1490         <xsd:element name="ClassificationAlgorithmType" type="tns:ClassificationAlgorithmType" minOccurs="1"
1491 maxOccurs="1">
1492             <xsd:annotation>
1493                 <xsd:documentation>Identifies the type of classification algorithm that was used to perform the
1494 classification.</xsd:documentation>
1495             </xsd:annotation>
1496         </xsd:element>
1497     </xsd:sequence>
1498 </xsd:complexType>
1499 <xsd:complexType name="EncounterListType">
1500     <xsd:annotation>
1501         <xsd:documentation>Defines a set of encounters.</xsd:documentation>
1502     </xsd:annotation>
1503     <xsd:sequence>
1504         <xsd:element name="EncounterID" type="tns:BIASIDType" minOccurs="0" maxOccurs="unbounded">
1505             <xsd:annotation>
1506                 <xsd:documentation>The identifier of an encounter.</xsd:documentation>
1507             </xsd:annotation>
1508         </xsd:element>
1509     </xsd:sequence>
1510 </xsd:complexType>
1511 <xsd:complexType name="FusionInformationListType">
1512     <xsd:annotation>
1513         <xsd:documentation>
1514             Contains at a minimum two sets of fusion input
1515             elements, as input to the PerformFusion request.
1516         </xsd:documentation>
1517     </xsd:annotation>
1518     <xsd:sequence>
1519         <xsd:element name="FusionElement" type="tns:FusionInformationType" minOccurs="2"
1520 maxOccurs="unbounded">
1521             <xsd:annotation>

```

```

1522     <xsd:documentation>A set of fusion information.</xsd:documentation>
1523   </xsd:annotation>
1524 </xsd:element>
1525 </xsd:sequence>
1526 </xsd:complexType>
1527 <xsd:complexType name="FusionInformationType">
1528   <xsd:annotation>
1529     <xsd:documentation>Represents the information necessary to perform a fusion
1530 operation.</xsd:documentation>
1531   </xsd:annotation>
1532   <xsd:sequence>
1533     <xsd:element name="BiometricType" type="xsd:hexBinary" minOccurs="1" maxOccurs="1">
1534       <xsd:annotation>
1535         <xsd:documentation>The type of biological or behavioral data stored in the biometric record, as defined by
1536 CBEFF.</xsd:documentation>
1537       </xsd:annotation>
1538     </xsd:element>
1539     <xsd:element name="BiometricSubType" type="xsd:string" minOccurs="0" maxOccurs="1">
1540       <xsd:annotation>
1541         <xsd:documentation>More specifically defines the type of biometric data stored in the biometric
1542 record.</xsd:documentation>
1543       </xsd:annotation>
1544     </xsd:element>
1545     <xsd:element name="AlgorithmOwner" type="xsd:string" minOccurs="1" maxOccurs="1">
1546       <xsd:annotation>
1547         <xsd:documentation>The owner or vendor of the algorithm used to determine the score or
1548 decision.</xsd:documentation>
1549       </xsd:annotation>
1550     </xsd:element>
1551     <xsd:element name="AlgorithmType" type="xsd:string" minOccurs="1" maxOccurs="1">
1552       <xsd:annotation>
1553         <xsd:documentation>The Algorithm Owner's identifier for the specific algorithm product and version used to
1554 determine the score or decision.</xsd:documentation>
1555       </xsd:annotation>
1556     </xsd:element>
1557     <xsd:choice minOccurs="1" maxOccurs="1">
1558       <xsd:element name="Score" type="xsd:unsignedLong">
1559         <xsd:annotation>
1560           <xsd:documentation>The similarity score assigned by the matching algorithm.</xsd:documentation>
1561         </xsd:annotation>
1562       </xsd:element>
1563       <xsd:element name="Decision" type="xsd:string">
1564         <xsd:annotation>
1565           <xsd:documentation>The match decision assigned by the matching algorithm.</xsd:documentation>
1566         </xsd:annotation>
1567       </xsd:element>
1568     </xsd:choice>
1569   </xsd:sequence>
1570 </xsd:complexType>
1571 <xsd:complexType name="GenericRequestParameters">
1572   <xsd:annotation>
1573     <xsd:documentation>Common request paramters that can be used to identify the
1574 requester.</xsd:documentation>

```

```

1575     </xsd:annotation>
1576   <xsd:sequence>
1577     <xsd:element name="Application" type="tns:ApplicationIdentifier" minOccurs="0" maxOccurs="1">
1578       <xsd:annotation>
1579         <xsd:documentation>Identifies the requesting application.</xsd:documentation>
1580       </xsd:annotation>
1581     </xsd:element>
1582     <xsd:element name="ApplicationUser" type="tns:ApplicationUserIdentifier" minOccurs="0" maxOccurs="1">
1583       <xsd:annotation>
1584         <xsd:documentation>Identifiers the user or instance of the requesting application.</xsd:documentation>
1585       </xsd:annotation>
1586     </xsd:element>
1587     <xsd:element name="BIASOperationName" type="xsd:string" minOccurs="0" maxOccurs="1">
1588       <xsd:annotation>
1589         <xsd:documentation>Identifiers the BIAS operation name that is being requested.</xsd:documentation>
1590       </xsd:annotation>
1591     </xsd:element>
1592   </xsd:sequence>
1593 </xsd:complexType>
1594 <xsd:simpleType name="IdentityModelType">
1595   <xsd:annotation>
1596     <xsd:documentation>Defines the requested or supported identity model.</xsd:documentation>
1597   </xsd:annotation>
1598   <xsd:restriction base="xsd:string">
1599     <xsd:enumeration value="encounter" />
1600     <xsd:enumeration value="person" />
1601   </xsd:restriction>
1602 </xsd:simpleType>
1603 <xsd:complexType name="InformationType">
1604   <xsd:annotation>
1605     <xsd:documentation>
1606       Allows for an unlimited number of data element types, and it does
1607       not specify nor require any particular data element.
1608     </xsd:documentation>
1609   </xsd:annotation>
1610   <xsd:sequence>
1611     <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
1612   </xsd:sequence>
1613 </xsd:complexType>
1614 <xsd:complexType name="ListFilterType">
1615   <xsd:annotation>
1616     <xsd:documentation>
1617       Provides a method to filter the amount of information returned in
1618       a search of biometric data.
1619     </xsd:documentation>
1620   </xsd:annotation>
1621   <xsd:sequence>
1622     <xsd:element name="BiometricTypeFilter" type="xsd:hexBinary" minOccurs="1" maxOccurs="unbounded">
1623       <xsd:annotation>
1624         <xsd:documentation>Limits the returned information to a specific type of biometric, as defined by
1625         CBEFF.</xsd:documentation>
1626       </xsd:annotation>
1627     </xsd:element>

```

```

1628     <xsd:element name="IncludeBiometricSubtype" type="xsd:boolean" minOccurs="1" maxOccurs="1">
1629     <xsd:annotation>
1630     <xsd:documentation>A Boolean flag indicating if biometric subtype information should be
1631 returned.</xsd:documentation>
1632     </xsd:annotation>
1633     </xsd:element>
1634 </xsd:sequence>
1635 </xsd:complexType>
1636 <xsd:simpleType name="MatchType">
1637 <xsd:annotation>
1638 <xsd:documentation>The result of a fusion method.</xsd:documentation>
1639 </xsd:annotation>
1640 <xsd:restriction base="xsd:boolean" />
1641 </xsd:simpleType>
1642 <xsd:complexType name="ProcessingOptionsType">
1643 <xsd:annotation>
1644 <xsd:documentation>
1645 BIAS aggregate services support the ability to include various
1646 processing options which direct and possibly control the business
1647 logic for that service. The ProcessingOptionsType provides a
1648 method to represent those options. Processing options should be
1649 defined by the implementing system.
1650 </xsd:documentation>
1651 </xsd:annotation>
1652 <xsd:sequence>
1653 <xsd:element name="Option" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
1654 <xsd:annotation>
1655 <xsd:documentation>An option supported by the implementing system.</xsd:documentation>
1656 </xsd:annotation>
1657 </xsd:element>
1658 </xsd:sequence>
1659 </xsd:complexType>
1660 <xsd:simpleType name="ProductID">
1661 <xsd:annotation>
1662 <xsd:documentation>The vendor's ID for a particular product.</xsd:documentation>
1663 </xsd:annotation>
1664 <xsd:restriction base="xsd:string" />
1665 </xsd:simpleType>
1666 <xsd:complexType name="QualityData">
1667 <xsd:annotation>
1668 <xsd:documentation>Contains information about a biometric sample's quality and the algorithm used to
1669 compute the quality.</xsd:documentation>
1670 </xsd:annotation>
1671 <xsd:sequence>
1672 <xsd:element name="QualityScore" type="tns:Score" minOccurs="0" maxOccurs="1">
1673 <xsd:annotation>
1674 <xsd:documentation>The quality of a biometric sample.</xsd:documentation>
1675 </xsd:annotation>
1676 </xsd:element>
1677 <xsd:element name="AlgorithmVendor" type="tns:VendorIdentifier" minOccurs="1" maxOccurs="1">
1678 <xsd:annotation>
1679 <xsd:documentation>The vendor of the quality algorithm used to determine the quality
1680 score.</xsd:documentation>

```



```

1681     </xsd:annotation>
1682 </xsd:element>
1683 <xsd:element name="AlgorithmVendorProductID" type="tns:ProductID" minOccurs="1" maxOccurs="1">
1684     <xsd:annotation>
1685         <xsd:documentation>The vendor's ID for the algorithm used to determine the quality.</xsd:documentation>
1686     </xsd:annotation>
1687 </xsd:element>
1688 <xsd:element name="AlgorithmVersion" type="tns:VersionType" minOccurs="0" maxOccurs="1">
1689     <xsd:annotation>
1690         <xsd:documentation>The version of the algorithm used to determine the quality.</xsd:documentation>
1691     </xsd:annotation>
1692 </xsd:element>
1693 </xsd:sequence>
1694 </xsd:complexType>
1695 <xsd:complexType name="RequestTemplate" abstract="true">
1696     <xsd:annotation>
1697         <xsd:documentation>Base template for BIAS primitive service requests.</xsd:documentation>
1698     </xsd:annotation>
1699     <xsd:sequence>
1700         <xsd:element name="GenericRequestParameters" type="tns:GenericRequestParameters" minOccurs="0"
1701 maxOccurs="1" />
1702     </xsd:sequence>
1703 </xsd:complexType>
1704 <xsd:complexType name="ResponseStatus">
1705     <xsd:sequence>
1706         <xsd:element name="Return" type="tns:ReturnCode" minOccurs="1" maxOccurs="1">
1707             <xsd:annotation>
1708                 <xsd:documentation>The return code indicates the return status of the operation.</xsd:documentation>
1709             </xsd:annotation>
1710         </xsd:element>
1711         <xsd:element name="Message" type="xsd:string" minOccurs="0" maxOccurs="1">
1712             <xsd:annotation>
1713                 <xsd:documentation>A short message corresponding to the return code.</xsd:documentation>
1714             </xsd:annotation>
1715         </xsd:element>
1716     </xsd:sequence>
1717 </xsd:complexType>
1718 <xsd:complexType name="ResponseTemplate" abstract="true">
1719     <xsd:annotation>
1720         <xsd:documentation>Base template for BIAS responses.</xsd:documentation>
1721     </xsd:annotation>
1722     <xsd:sequence>
1723         <xsd:element name="ResponseStatus" type="tns:ResponseStatus" minOccurs="1" maxOccurs="1">
1724             <xsd:annotation>
1725                 <xsd:documentation>Returned status for the operation.</xsd:documentation>
1726             </xsd:annotation>
1727         </xsd:element>
1728     </xsd:sequence>
1729 </xsd:complexType>
1730 <xsd:simpleType name="ReturnCode">
1731     <xsd:annotation>
1732         <xsd:documentation>BIAS Operation Return Codes</xsd:documentation>
1733     </xsd:annotation>

```

```

1734     <xsd:restriction base="xsd:unsignedLong">
1735         <xsd:enumeration value="0">
1736             <xsd:annotation>
1737                 <xsd:documentation>Success</xsd:documentation>
1738             </xsd:annotation>
1739         </xsd:enumeration>
1740     </xsd:restriction>
1741 </xsd:simpleType>
1742 <xsd:complexType name="SB-info">
1743     <xsd:annotation>
1744         <xsd:documentation>
1745             For a description or definition of each data element, see the
1746             referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
1747         </xsd:documentation>
1748     </xsd:annotation>
1749     <xsd:sequence>
1750         <xsd:element name="format-owner" type="xsd:integer" minOccurs="1" maxOccurs="1" />
1751         <xsd:element name="format-type" type="xsd:integer" minOccurs="1" maxOccurs="1" />
1752     </xsd:sequence>
1753 </xsd:complexType>
1754 <xsd:simpleType name="Score">
1755     <xsd:annotation>
1756         <xsd:documentation>Match result or quality score.</xsd:documentation>
1757     </xsd:annotation>
1758     <xsd:restriction base="xsd:integer" />
1759 </xsd:simpleType>
1760 <xsd:complexType name="TokenType">
1761     <xsd:annotation>
1762         <xsd:documentation>Defines a token that is returned for asynchronous processing.</xsd:documentation>
1763     </xsd:annotation>
1764     <xsd:sequence>
1765         <xsd:element name="TokenValue" type="xsd:string" minOccurs="1" maxOccurs="1">
1766             <xsd:annotation>
1767                 <xsd:documentation>A value returned by the implementing system that is used to retrieve the results to a
1768                 service at a later time.</xsd:documentation>
1769             </xsd:annotation>
1770         </xsd:element>
1771         <xsd:element name="Expiration" type="xsd:date" minOccurs="1" maxOccurs="1">
1772             <xsd:annotation>
1773                 <xsd:documentation>A date and time at which point the token expires and the service results are no longer
1774                 guaranteed to be available.</xsd:documentation>
1775             </xsd:annotation>
1776         </xsd:element>
1777     </xsd:sequence>
1778 </xsd:complexType>
1779 <xsd:simpleType name="VendorIdentifier">
1780     <xsd:annotation>
1781         <xsd:documentation>Identifies a vendor.</xsd:documentation>
1782     </xsd:annotation>
1783     <xsd:restriction base="xsd:string" />
1784 </xsd:simpleType>
1785 <xsd:complexType name="Version">
1786     <xsd:annotation>

```

```

1787     <xsd:documentation>
1788         For a description or definition of each data element, see the
1789         referenced CBEFF standards in the CBEFF_XML_BIR_Type schema.
1790     </xsd:documentation>
1791 </xsd:annotation>
1792 <xsd:sequence>
1793     <xsd:element name="major" type="xsd:integer" minOccurs="1" maxOccurs="1" />
1794     <xsd:element name="minor" type="xsd:integer" minOccurs="1" maxOccurs="1" />
1795 </xsd:sequence>
1796 </xsd:complexType>
1797 <xsd:simpleType name="VersionType">
1798     <xsd:annotation>
1799         <xsd:documentation>The version of a component.</xsd:documentation>
1800     </xsd:annotation>
1801     <xsd:restriction base="xsd:string" />
1802 </xsd:simpleType>
1803 <xsd:element name="AddSubjectToGalleryRequest">
1804     <xsd:complexType>
1805         <xsd:annotation>
1806             <xsd:documentation>
1807                 Register a subject to a given gallery or population group.
1808             </xsd:documentation>
1809         </xsd:annotation>
1810         <xsd:complexContent>
1811             <xsd:extension base="tns:RequestTemplate">
1812                 <xsd:sequence>
1813                     <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
1814                         <xsd:annotation>
1815                             <xsd:documentation>The identifier of the gallery or population group to which the subject will be
1816 added.</xsd:documentation>
1817                         </xsd:annotation>
1818                     </xsd:element>
1819                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
1820                         <xsd:annotation>
1821                             <xsd:documentation>The identity to add to the gallery.</xsd:documentation>
1822                         </xsd:annotation>
1823                     </xsd:element>
1824                 </xsd:sequence>
1825             </xsd:extension>
1826         </xsd:complexContent>
1827     </xsd:complexType>
1828 </xsd:element>
1829 <xsd:element name="AddSubjectToGalleryResponse">
1830     <xsd:complexType>
1831         <xsd:annotation>
1832             <xsd:documentation>
1833                 The response to an AddSubjectToGallery request.
1834             </xsd:documentation>
1835         </xsd:annotation>
1836         <xsd:complexContent>
1837             <xsd:extension base="tns:ResponseTemplate" />
1838         </xsd:complexContent>
1839     </xsd:complexType>

```

```

1840 </xsd:element>
1841 <xsd:element name="CheckQualityRequest">
1842   <xsd:complexType>
1843     <xsd:annotation>
1844       <xsd:documentation>
1845         Calculate a quality score for a given biometric.
1846       </xsd:documentation>
1847     </xsd:annotation>
1848     <xsd:complexContent>
1849       <xsd:extension base="tns:RequestTemplate">
1850         <xsd:sequence>
1851           <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
1852             <xsd:annotation>
1853               <xsd:documentation>Data structure containing a single biometric sample for which a quality score is to be
1854 determined.</xsd:documentation>
1855             </xsd:annotation>
1856           </xsd:element>
1857           <xsd:element name="Quality" type="tns:QualityData" minOccurs="0" maxOccurs="1">
1858             <xsd:annotation>
1859               <xsd:documentation>Specifies a particular algorithm vendor and vender product ID.</xsd:documentation>
1860             </xsd:annotation>
1861           </xsd:element>
1862         </xsd:sequence>
1863       </xsd:extension>
1864     </xsd:complexContent>
1865   </xsd:complexType>
1866 </xsd:element>
1867 <xsd:element name="CheckQualityResponse">
1868   <xsd:complexType>
1869     <xsd:annotation>
1870       <xsd:documentation>
1871         The response to a CheckQuality request.
1872       </xsd:documentation>
1873     </xsd:annotation>
1874     <xsd:complexContent>
1875       <xsd:extension base="tns:ResponseTemplate">
1876         <xsd:sequence>
1877           <xsd:element name="QualityInfo" type="tns:QualityData" minOccurs="1" maxOccurs="1">
1878             <xsd:annotation>
1879               <xsd:documentation>Contains the quality information for the submitted biometric
1880 sample.</xsd:documentation>
1881             </xsd:annotation>
1882           </xsd:element>
1883         </xsd:sequence>
1884       </xsd:extension>
1885     </xsd:complexContent>
1886   </xsd:complexType>
1887 </xsd:element>
1888 <xsd:element name="ClassifyBiometricDataRequest">
1889   <xsd:complexType>
1890     <xsd:annotation>
1891       <xsd:documentation>
1892         Classifies a biometric sample.

```

```

1893     </xsd:documentation>
1894 </xsd:annotation>
1895 <xsd:complexContent>
1896   <xsd:extension base="tns:RequestTemplate">
1897     <xsd:sequence>
1898       <xsd:element name="BiometricData" type="tns:BIASBiometricDataType" minOccurs="1" maxOccurs="1">
1899         <xsd:annotation>
1900           <xsd:documentation>Data structure containing a single biometric sample for which the classification is to
1901 be determined.</xsd:documentation>
1902         </xsd:annotation>
1903       </xsd:element>
1904     </xsd:sequence>
1905   </xsd:extension>
1906 </xsd:complexContent>
1907 </xsd:complexType>
1908 </xsd:element>
1909 <xsd:element name="ClassifyBiometricDataResponse">
1910   <xsd:complexType>
1911     <xsd:annotation>
1912       <xsd:documentation>
1913         The response to a ClassifyBiometricData request, containing
1914         the classification of a biometric sample.
1915       </xsd:documentation>
1916     </xsd:annotation>
1917     <xsd:complexContent>
1918       <xsd:extension base="tns:ResponseTemplate">
1919         <xsd:sequence>
1920           <xsd:element name="ClassificationData" type="tns:ClassificationData" minOccurs="1" maxOccurs="1">
1921             <xsd:annotation>
1922               <xsd:documentation>Information on the results and type of classification
1923 performed.</xsd:documentation>
1924             </xsd:annotation>
1925           </xsd:element>
1926         </xsd:sequence>
1927       </xsd:extension>
1928     </xsd:complexContent>
1929   </xsd:complexType>
1930 </xsd:element>
1931 <xsd:element name="CreateSubjectRequest">
1932   <xsd:complexType>
1933     <xsd:annotation>
1934       <xsd:documentation>
1935         Create a new subject record.
1936       </xsd:documentation>
1937     </xsd:annotation>
1938     <xsd:complexContent>
1939       <xsd:extension base="tns:RequestTemplate">
1940         <xsd:sequence>
1941           <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
1942             <xsd:annotation>
1943               <xsd:documentation>Subject ID of the identity to create.</xsd:documentation>
1944             </xsd:annotation>
1945           </xsd:element>

```

```

1946         </xsd:sequence>
1947     </xsd:extension>
1948 </xsd:complexContent>
1949 </xsd:complexType>
1950 </xsd:element>
1951 <xsd:element name="CreateSubjectResponse">
1952     <xsd:complexType>
1953         <xsd:annotation>
1954             <xsd:documentation>
1955                 The response to a CreateSubject request, containing the subject
1956                 ID of the new subject record.
1957             </xsd:documentation>
1958         </xsd:annotation>
1959         <xsd:complexContent>
1960             <xsd:extension base="tns:ResponseTemplate">
1961                 <xsd:sequence>
1962                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
1963                         <xsd:annotation>
1964                             <xsd:documentation>Contains the subject ID of the new subject record.</xsd:documentation>
1965                         </xsd:annotation>
1966                     </xsd:element>
1967                 </xsd:sequence>
1968             </xsd:extension>
1969         </xsd:complexContent>
1970     </xsd:complexType>
1971 </xsd:element>
1972 <xsd:element name="DeleteBiographicDataRequest">
1973     <xsd:complexType>
1974         <xsd:annotation>
1975             <xsd:documentation>
1976                 Erase all of the biographic data associated with a given
1977                 subject record or, in the encounter-centric model, with a
1978                 given encounter.
1979             </xsd:documentation>
1980         </xsd:annotation>
1981         <xsd:complexContent>
1982             <xsd:extension base="tns:RequestTemplate">
1983                 <xsd:sequence>
1984                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
1985                         <xsd:annotation>
1986                             <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
1987                         </xsd:annotation>
1988                     </xsd:element>
1989                 </xsd:sequence>
1990             </xsd:extension>
1991         </xsd:complexContent>
1992     </xsd:complexType>
1993 </xsd:element>
1994 <xsd:element name="DeleteBiographicDataResponse">
1995     <xsd:complexType>
1996         <xsd:annotation>
1997             <xsd:documentation>
1998                 The response to a DeleteBiographicData request.

```

```

1999     </xsd:documentation>
2000     </xsd:annotation>
2001     <xsd:complexContent>
2002         <xsd:extension base="tns:ResponseTemplate" />
2003     </xsd:complexContent>
2004 </xsd:complexType>
2005 </xsd:element>
2006 <xsd:element name="DeleteBiometricDataRequest">
2007     <xsd:complexType>
2008         <xsd:annotation>
2009             <xsd:documentation>
2010                 Erase all of the biometric data associated with a given
2011                 subject record or, in the encounter-centric model, with a
2012                 given encounter.
2013             </xsd:documentation>
2014         </xsd:annotation>
2015         <xsd:complexContent>
2016             <xsd:extension base="tns:RequestTemplate">
2017                 <xsd:sequence>
2018                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2019                         <xsd:annotation>
2020                             <xsd:documentation>Contains either the subject ID or encounter ID reference.</xsd:documentation>
2021                         </xsd:annotation>
2022                     </xsd:element>
2023                 </xsd:sequence>
2024             </xsd:extension>
2025         </xsd:complexContent>
2026     </xsd:complexType>
2027 </xsd:element>
2028 <xsd:element name="DeleteBiometricDataResponse">
2029     <xsd:complexType>
2030         <xsd:annotation>
2031             <xsd:documentation>
2032                 The response to a DeleteBiometricData request.
2033             </xsd:documentation>
2034         </xsd:annotation>
2035         <xsd:complexContent>
2036             <xsd:extension base="tns:ResponseTemplate" />
2037         </xsd:complexContent>
2038     </xsd:complexType>
2039 </xsd:element>
2040 <xsd:element name="DeleteSubjectRequest">
2041     <xsd:complexType>
2042         <xsd:annotation>
2043             <xsd:documentation>
2044                 Delete an existing subject record and, in an encounter-centric
2045                 model, any associated encounter information.
2046             </xsd:documentation>
2047         </xsd:annotation>
2048         <xsd:complexContent>
2049             <xsd:extension base="tns:RequestTemplate">
2050                 <xsd:sequence>
2051                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">

```

```

2052         <xsd:annotation>
2053             <xsd:documentation>Subject ID of the identity to delete.</xsd:documentation>
2054         </xsd:annotation>
2055     </xsd:element>
2056 </xsd:sequence>
2057 </xsd:extension>
2058 </xsd:complexContent>
2059 </xsd:complexType>
2060 </xsd:element>
2061 <xsd:element name="DeleteSubjectResponse">
2062     <xsd:complexType>
2063         <xsd:annotation>
2064             <xsd:documentation>
2065                 The response to a DeleteSubject request.
2066             </xsd:documentation>
2067         </xsd:annotation>
2068         <xsd:complexContent>
2069             <xsd:extension base="tns:ResponseTemplate" />
2070         </xsd:complexContent>
2071     </xsd:complexType>
2072 </xsd:element>
2073 <xsd:element name="DeleteSubjectFromGalleryRequest">
2074     <xsd:complexType>
2075         <xsd:annotation>
2076             <xsd:documentation>
2077                 Remove the registration of a subject from a gallery or
2078                 population group.
2079             </xsd:documentation>
2080         </xsd:annotation>
2081         <xsd:complexContent>
2082             <xsd:extension base="tns:RequestTemplate">
2083                 <xsd:sequence>
2084                     <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2085                         <xsd:annotation>
2086                             <xsd:documentation>The identifier of the gallery or population group from which the subject will be
2087 deleted.</xsd:documentation>
2088                         </xsd:annotation>
2089                     </xsd:element>
2090                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2091                         <xsd:annotation>
2092                             <xsd:documentation>The identity to remove from the gallery.</xsd:documentation>
2093                         </xsd:annotation>
2094                     </xsd:element>
2095                 </xsd:sequence>
2096             </xsd:extension>
2097         </xsd:complexContent>
2098     </xsd:complexType>
2099 </xsd:element>
2100 <xsd:element name="DeleteSubjectFromGalleryResponse">
2101     <xsd:complexType>
2102         <xsd:annotation>
2103             <xsd:documentation>
2104                 The response to a DeleteSubjectFromGallery request.

```



```

2105     </xsd:documentation>
2106 </xsd:annotation>
2107 <xsd:complexContent>
2108   <xsd:extension base="tns:ResponseTemplate" />
2109 </xsd:complexContent>
2110 </xsd:complexType>
2111 </xsd:element>
2112 <xsd:element name="GetIdentifySubjectResultsRequest">
2113   <xsd:complexType>
2114     <xsd:annotation>
2115       <xsd:documentation>
2116         Retrieve the identification results for a specified token,
2117         which was returned by the Identify Subject service.
2118       </xsd:documentation>
2119     </xsd:annotation>
2120     <xsd:complexContent>
2121       <xsd:extension base="tns:RequestTemplate">
2122         <xsd:sequence>
2123           <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2124             <xsd:annotation>
2125               <xsd:documentation>A value used to retrieve the results of an IdentifySubject
2126 request.</xsd:documentation>
2127             </xsd:annotation>
2128           </xsd:element>
2129         </xsd:sequence>
2130       </xsd:extension>
2131     </xsd:complexContent>
2132   </xsd:complexType>
2133 </xsd:element>
2134 <xsd:element name="GetIdentifySubjectResultsResponse">
2135   <xsd:complexType>
2136     <xsd:annotation>
2137       <xsd:documentation>
2138         The response to a GetIdentifySubjectResults request, which includes a candidate list.
2139       </xsd:documentation>
2140     </xsd:annotation>
2141     <xsd:complexContent>
2142       <xsd:extension base="tns:ResponseTemplate">
2143         <xsd:sequence>
2144           <xsd:element name="CandidateList" type="tns:CandidateListType" minOccurs="1" maxOccurs="1">
2145             <xsd:annotation>
2146               <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input
2147 biometric sample.</xsd:documentation>
2148             </xsd:annotation>
2149           </xsd:element>
2150         </xsd:sequence>
2151       </xsd:extension>
2152     </xsd:complexContent>
2153   </xsd:complexType>
2154 </xsd:element>
2155 <xsd:element name="IdentifySubjectRequest">
2156   <xsd:complexType>
2157     <xsd:annotation>

```

```

2158     <xsd:documentation>
2159         Perform an identification search against a given gallery for
2160         a given biometric.
2161     </xsd:documentation>
2162 </xsd:annotation>
2163 <xsd:complexContent>
2164     <xsd:extension base="tns:RequestTemplate">
2165         <xsd:sequence>
2166             <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="1" maxOccurs="1">
2167                 <xsd:annotation>
2168                     <xsd:documentation>The identifier of the gallery or population group which will be
2169 searched.</xsd:documentation>
2170                 </xsd:annotation>
2171             </xsd:element>
2172             <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2173                 <xsd:annotation>
2174                     <xsd:documentation>Contains the BIR, a data structure containing the biometric sample for the
2175 search.</xsd:documentation>
2176                 </xsd:annotation>
2177             </xsd:element>
2178             <xsd:element name="MaxListSize" type="xsd:integer" minOccurs="1" maxOccurs="1">
2179                 <xsd:annotation>
2180                     <xsd:documentation>The maximum size of the candidate list that should be
2181 returned.</xsd:documentation>
2182                 </xsd:annotation>
2183             </xsd:element>
2184         </xsd:sequence>
2185     </xsd:extension>
2186 </xsd:complexContent>
2187 </xsd:complexType>
2188 </xsd:element>
2189 <xsd:element name="IdentifySubjectResponse">
2190     <xsd:complexType>
2191         <xsd:annotation>
2192             <xsd:documentation>
2193                 The response to an IdentifySubject request, returning a
2194                 rank-ordered candidate list.
2195             </xsd:documentation>
2196         </xsd:annotation>
2197         <xsd:complexContent>
2198             <xsd:extension base="tns:ResponseTemplate">
2199                 <xsd:choice minOccurs="1" maxOccurs="1">
2200                     <xsd:element name="CandidateList" type="tns:CandidateListType">
2201                         <xsd:annotation>
2202                             <xsd:documentation>A rank-ordered list of candidates that have a likelihood of matching the input
2203 biometric sample; returned with successful synchronous request processing.</xsd:documentation>
2204                         </xsd:annotation>
2205                     </xsd:element>
2206                     <xsd:element name="Token" type="tns:TokenType">
2207                         <xsd:annotation>
2208                             <xsd:documentation>A token used to retrieve the results of the IdentifySubject request; returned with
2209 asynchronous request processing.</xsd:documentation>
2210                         </xsd:annotation>

```

```

2211         </xsd:element>
2212     </xsd:choice>
2213 </xsd:extension>
2214 </xsd:complexContent>
2215 </xsd:complexType>
2216 </xsd:element>
2217 <xsd:element name="ListBiographicDataRequest">
2218     <xsd:complexType>
2219         <xsd:annotation>
2220             <xsd:documentation>
2221                 Lists the biographic data elements stored for a subject.
2222             </xsd:documentation>
2223         </xsd:annotation>
2224     <xsd:complexContent>
2225         <xsd:extension base="tns:RequestTemplate">
2226             <xsd:sequence>
2227                 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2228                     <xsd:annotation>
2229                         <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2230 encounter.</xsd:documentation>
2231                     </xsd:annotation>
2232                 </xsd:element>
2233             </xsd:sequence>
2234         </xsd:extension>
2235     </xsd:complexContent>
2236 </xsd:complexType>
2237 </xsd:element>
2238 <xsd:element name="ListBiographicDataResponse">
2239     <xsd:complexType>
2240         <xsd:annotation>
2241             <xsd:documentation>
2242                 The response to a ListBiographicData request, containing a list
2243                 of biographic data elements stored for a subject. In the
2244                 encounter-centric model, the biographic data elements for a
2245                 specific encounter are returned. If an encounter ID is not
2246                 specified and encounter data exists for the subject, the list
2247                 of encounter IDs which contain biographic data is returned.
2248             </xsd:documentation>
2249         </xsd:annotation>
2250     <xsd:complexContent>
2251         <xsd:extension base="tns:ResponseTemplate">
2252             <xsd:sequence>
2253                 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2254                     <xsd:annotation>
2255                         <xsd:documentation>
2256                             Contains a list of biographic data elements associated with a
2257                             subject or encounter; non-empty if the service was
2258                             successful, biographic data exists, and either (a) the
2259                             person-centric model is being used or (b) the
2260                             encounter-centric model is being used and an encounter
2261                             identifier was specified.
2262                         </xsd:documentation>
2263                     </xsd:annotation>

```

```

2264     </xsd:element>
2265     <xsd:element name="EncounterList" type="tns:EncounterListType" minOccurs="0" maxOccurs="1">
2266         <xsd:annotation>
2267             <xsd:documentation>
2268                 A list of encounter ID's associated with a subject and
2269                 which contain biographic data; non-empty if the service
2270                 was successful, biographic data exists, the
2271                 encounter-centric model is being used, and an encounter
2272                 identifier was not specified.
2273             </xsd:documentation>
2274         </xsd:annotation>
2275     </xsd:element>
2276 </xsd:sequence>
2277 </xsd:extension>
2278 </xsd:complexContent>
2279 </xsd:complexType>
2280 </xsd:element>
2281 <xsd:element name="ListBiometricDataRequest">
2282     <xsd:complexType>
2283         <xsd:annotation>
2284             <xsd:documentation>
2285                 Lists the biometric data elements stored for a subject. Note
2286                 that no actual biometric data is returned by this service (see
2287                 the RetrieveBiometricInformation service to obtain the biometric
2288                 data).
2289             </xsd:documentation>
2290         </xsd:annotation>
2291     <xsd:complexContent>
2292         <xsd:extension base="tns:RequestTemplate">
2293             <xsd:sequence>
2294                 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2295                     <xsd:annotation>
2296                         <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2297 encounter.</xsd:documentation>
2298                     </xsd:annotation>
2299                 </xsd:element>
2300                 <xsd:element name="ListFilter" type="tns:ListFilterType" minOccurs="0" maxOccurs="1">
2301                     <xsd:annotation>
2302                         <xsd:documentation>Indicates what biometric information should be returned.</xsd:documentation>
2303                     </xsd:annotation>
2304                 </xsd:element>
2305             </xsd:sequence>
2306         </xsd:extension>
2307     </xsd:complexContent>
2308 </xsd:complexType>
2309 </xsd:element>
2310 <xsd:element name="ListBiometricDataResponse">
2311     <xsd:complexType>
2312         <xsd:annotation>
2313             <xsd:documentation>
2314                 The response to a ListBiometricData request, containing a list
2315                 of biometric data elements stored for a subject. In the
2316                 encounter-centric model, the biometric data elements for a

```

2317 specific encounter are returned. If an encounter ID is not
 2318 specified and encounter data exists for the subject, the list
 2319 of encounter IDs which contain biometric data is returned.
 2320 </xsd:documentation>
 2321 </xsd:annotation>
 2322 <xsd:complexContent>
 2323 <xsd:extension base="tns:ResponseTemplate">
 2324 <xsd:sequence>
 2325 <xsd:element name="Identity" type="tns:BIASIdentity">
 2326 <xsd:annotation>
 2327 <xsd:documentation>
 2328 Includes a list of biometric data elements associated
 2329 with a subject or encounter or a list of encounter ID's
 2330 associated with a subject and which contain biometric
 2331 data.
 2332 </xsd:documentation>
 2333 </xsd:annotation>
 2334 </xsd:element>
 2335 </xsd:sequence>
 2336 </xsd:extension>
 2337 </xsd:complexContent>
 2338 </xsd:complexType>
 2339 </xsd:element>
 2340 <xsd:element name="PerformFusionRequest">
 2341 <xsd:complexType>
 2342 <xsd:annotation>
 2343 <xsd:documentation>
 2344 Accepts either match score or match decision information and creates a fused match result.
 2345 </xsd:documentation>
 2346 </xsd:annotation>
 2347 <xsd:complexContent>
 2348 <xsd:extension base="tns:RequestTemplate">
 2349 <xsd:sequence>
 2350 <xsd:element name="FusionInput" type="tns:FusionInformationListType" minOccurs="1" maxOccurs="1">
 2351 <xsd:annotation>
 2352 <xsd:documentation>Score or decision input information to the fusion method.</xsd:documentation>
 2353 </xsd:annotation>
 2354 </xsd:element>
 2355 </xsd:sequence>
 2356 </xsd:extension>
 2357 </xsd:complexContent>
 2358 </xsd:complexType>
 2359 </xsd:element>
 2360 <xsd:element name="PerformFusionResponse">
 2361 <xsd:complexType>
 2362 <xsd:annotation>
 2363 <xsd:documentation>
 2364 The response to the PerformFusion request.
 2365 </xsd:documentation>
 2366 </xsd:annotation>
 2367 <xsd:complexContent>
 2368 <xsd:extension base="tns:ResponseTemplate">
 2369 <xsd:sequence>

```

2370     <xsd:element name="Match" type="tns:MatchType" minOccurs="1" maxOccurs="1">
2371     <xsd:annotation>
2372     <xsd:documentation>Indicates the result of the fusion method</xsd:documentation>
2373     </xsd:annotation>
2374     </xsd:element>
2375 </xsd:sequence>
2376 </xsd:extension>
2377 </xsd:complexContent>
2378 </xsd:complexType>
2379 </xsd:element>
2380 <xsd:element name="QueryCapabilitiesRequest">
2381 <xsd:complexType>
2382 <xsd:annotation>
2383 <xsd:documentation>
2384     Returns a list of the capabilities, options, galleries, etc.
2385     that are supported by the BIAS implementation.
2386 </xsd:documentation>
2387 </xsd:annotation>
2388 <xsd:complexContent>
2389     <xsd:extension base="tns:RequestTemplate" />
2390 </xsd:complexContent>
2391 </xsd:complexType>
2392 </xsd:element>
2393 <xsd:element name="QueryCapabilitiesResponse">
2394 <xsd:complexType>
2395 <xsd:annotation>
2396 <xsd:documentation>
2397     The response to a QueryCapabilities request.
2398 </xsd:documentation>
2399 </xsd:annotation>
2400 <xsd:complexContent>
2401     <xsd:extension base="tns:ResponseTemplate">
2402     <xsd:sequence>
2403     <xsd:element name="CapabilityList" type="tns:CapabilityListType" minOccurs="1" maxOccurs="1">
2404     <xsd:annotation>
2405     <xsd:documentation>A list of capabilities supported by the BIAS implementation.</xsd:documentation>
2406     </xsd:annotation>
2407     </xsd:element>
2408     </xsd:sequence>
2409     </xsd:extension>
2410     </xsd:complexContent>
2411     </xsd:complexType>
2412 </xsd:element>
2413 <xsd:element name="RetrieveBiographicInformationRequest">
2414 <xsd:complexType>
2415 <xsd:annotation>
2416 <xsd:documentation>
2417     Retrieves the biographic data associated with a subject ID.
2418 </xsd:documentation>
2419 </xsd:annotation>
2420 <xsd:complexContent>
2421     <xsd:extension base="tns:RequestTemplate">
2422     <xsd:sequence>

```

```

2423     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2424     <xsd:annotation>
2425     <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2426 encounter.</xsd:documentation>
2427     </xsd:annotation>
2428     </xsd:element>
2429 </xsd:sequence>
2430 </xsd:extension>
2431 </xsd:complexContent>
2432 </xsd:complexType>
2433 </xsd:element>
2434 <xsd:element name="RetrieveBiographicInformationResponse">
2435 <xsd:complexType>
2436 <xsd:annotation>
2437 <xsd:documentation>
2438 The response to a RetrieveBiographicInformation request,
2439 containing the biographic data associated with a subject ID. In
2440 the encounter-centric model, the biographic data associated with
2441 a specified encounter is returned. If the encounter ID is not
2442 specified in the encounter-centric model, the biographic
2443 information associated with the most recent encounter is returned.
2444 </xsd:documentation>
2445 </xsd:annotation>
2446 <xsd:complexContent>
2447 <xsd:extension base="tns:ResponseTemplate">
2448 <xsd:sequence>
2449 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2450 <xsd:annotation>
2451 <xsd:documentation>Includes the set of biographic data associated with a subject.</xsd:documentation>
2452 </xsd:annotation>
2453 </xsd:element>
2454 </xsd:sequence>
2455 </xsd:extension>
2456 </xsd:complexContent>
2457 </xsd:complexType>
2458 </xsd:element>
2459 <xsd:element name="RetrieveBiometricInformationRequest">
2460 <xsd:complexType>
2461 <xsd:annotation>
2462 <xsd:documentation>
2463 Retrieves the biometric data associated with a subject ID.
2464 </xsd:documentation>
2465 </xsd:annotation>
2466 <xsd:complexContent>
2467 <xsd:extension base="tns:RequestTemplate">
2468 <xsd:sequence>
2469 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2470 <xsd:annotation>
2471 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an
2472 encounter.</xsd:documentation>
2473 </xsd:annotation>
2474 </xsd:element>
2475 <xsd:element name="BiometricType" type="xsd:string" minOccurs="0" maxOccurs="1">

```

```

2476         <xsd:annotation>
2477         <xsd:documentation>The type of biological or behavioral data to retrieve.</xsd:documentation>
2478         </xsd:annotation>
2479     </xsd:element>
2480 </xsd:sequence>
2481 </xsd:extension>
2482 </xsd:complexContent>
2483 </xsd:complexType>
2484 </xsd:element>
2485 <xsd:element name="RetrieveBiometricInformationResponse">
2486 <xsd:complexType>
2487 <xsd:annotation>
2488 <xsd:documentation>
2489     The response to a RetrieveBiometricInformation request,
2490     containing the biometric data associated with a subject ID. In
2491     the encounter-centric model, the biometric data associated with
2492     a specified encounter is returned. If the encounter ID is not
2493     specified in the encounter-centric model, the biometric
2494     information associated with the most recent encounter is returned.
2495 </xsd:documentation>
2496 </xsd:annotation>
2497 <xsd:complexContent>
2498 <xsd:extension base="tns:ResponseTemplate">
2499 <xsd:sequence>
2500 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2501 <xsd:annotation>
2502 <xsd:documentation>Includes the biometric data associated with a subject.</xsd:documentation>
2503 </xsd:annotation>
2504 </xsd:element>
2505 </xsd:sequence>
2506 </xsd:extension>
2507 </xsd:complexContent>
2508 </xsd:complexType>
2509 </xsd:element>
2510 <xsd:element name="SetBiographicDataRequest">
2511 <xsd:complexType>
2512 <xsd:annotation>
2513 <xsd:documentation>
2514     Associates biographic data to a given subject record.
2515 </xsd:documentation>
2516 </xsd:annotation>
2517 <xsd:complexContent>
2518 <xsd:extension base="tns:RequestTemplate">
2519 <xsd:sequence>
2520 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2521 <xsd:annotation>
2522 <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2523 and includes the biographic data to store.</xsd:documentation>
2524 </xsd:annotation>
2525 </xsd:element>
2526 <xsd:element name="IdentityModel" type="tns:IdentityModelType" minOccurs="1" maxOccurs="1">
2527 <xsd:annotation>
2528 <xsd:documentation>Indicates a person-centric or encounter-centric model.</xsd:documentation>

```



```

2529         </xsd:annotation>
2530     </xsd:element>
2531 </xsd:sequence>
2532 </xsd:extension>
2533 </xsd:complexContent>
2534 </xsd:complexType>
2535 </xsd:element>
2536 <xsd:element name="SetBiographicDataResponse">
2537     <xsd:complexType>
2538         <xsd:annotation>
2539             <xsd:documentation>
2540                 The response to a SetBiographicData request.
2541             </xsd:documentation>
2542         </xsd:annotation>
2543         <xsd:complexContent>
2544             <xsd:extension base="tns:ResponseTemplate">
2545                 <xsd:sequence>
2546                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2547                         <xsd:annotation>
2548                             <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
2549 encounter.</xsd:documentation>
2550                         </xsd:annotation>
2551                     </xsd:element>
2552                 </xsd:sequence>
2553             </xsd:extension>
2554         </xsd:complexContent>
2555     </xsd:complexType>
2556 </xsd:element>
2557 <xsd:element name="SetBiometricDataRequest">
2558     <xsd:complexType>
2559         <xsd:annotation>
2560             <xsd:documentation>
2561                 Associates biometric data to a given subject record.
2562             </xsd:documentation>
2563         </xsd:annotation>
2564         <xsd:complexContent>
2565             <xsd:extension base="tns:RequestTemplate">
2566                 <xsd:sequence>
2567                     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2568                         <xsd:annotation>
2569                             <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2570 and includes the biometric data to store.</xsd:documentation>
2571                         </xsd:annotation>
2572                     </xsd:element>
2573                     <xsd:element name="IdentityModel" type="tns:IdentityModelType" minOccurs="1" maxOccurs="1">
2574                         <xsd:annotation>
2575                             <xsd:documentation>Indicates a person-centric or encounter-centric model.</xsd:documentation>
2576                         </xsd:annotation>
2577                     </xsd:element>
2578                 </xsd:sequence>
2579             </xsd:extension>
2580         </xsd:complexContent>
2581     </xsd:complexType>

```

```

2582     </xsd:element>
2583     <xsd:element name="SetBiometricDataResponse">
2584       <xsd:complexType>
2585         <xsd:annotation>
2586           <xsd:documentation>
2587             The response to a SetBiometricData request.
2588           </xsd:documentation>
2589         </xsd:annotation>
2590         <xsd:complexContent>
2591           <xsd:extension base="tns:ResponseTemplate">
2592             <xsd:sequence>
2593               <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="0" maxOccurs="1">
2594                 <xsd:annotation>
2595                   <xsd:documentation>In an encounter-centric model, identifies the encounter ID assigned to a new
2596 encounter.</xsd:documentation>
2597                 </xsd:annotation>
2598               </xsd:element>
2599             </xsd:sequence>
2600           </xsd:extension>
2601         </xsd:complexContent>
2602       </xsd:complexType>
2603     </xsd:element>
2604     <xsd:element name="TransformBiometricDataRequest">
2605       <xsd:complexType>
2606         <xsd:annotation>
2607           <xsd:documentation>
2608             Transforms or processes a given biometric in one format into a new target format.
2609           </xsd:documentation>
2610         </xsd:annotation>
2611         <xsd:complexContent>
2612           <xsd:extension base="tns:RequestTemplate">
2613             <xsd:sequence>
2614               <xsd:element name="InputBIR" type="tns:CBEFF_BIR_Type" minOccurs="1" maxOccurs="1">
2615                 <xsd:annotation>
2616                   <xsd:documentation>Data structure containing the biometric information to be
2617 transformed.</xsd:documentation>
2618                 </xsd:annotation>
2619               </xsd:element>
2620               <xsd:element name="TransformOperation" type="xsd:unsignedLong" minOccurs="1" maxOccurs="1">
2621                 <xsd:annotation>
2622                   <xsd:documentation>Value indicating the type of transformation to perform.</xsd:documentation>
2623                 </xsd:annotation>
2624               </xsd:element>
2625             </xsd:sequence>
2626           </xsd:extension>
2627         </xsd:complexContent>
2628       </xsd:complexType>
2629     </xsd:element>
2630     <xsd:element name="TransformBiometricDataResponse">
2631       <xsd:complexType>
2632         <xsd:annotation>
2633           <xsd:documentation>
2634             The response to a TransformBiometricData request.

```

```

2635     </xsd:documentation>
2636 </xsd:annotation>
2637 <xsd:complexContent>
2638   <xsd:extension base="tns:ResponseTemplate">
2639     <xsd:sequence>
2640       <xsd:element name="OutputBIR" type="tns:CBEFF_BIR_Type" minOccurs="0" maxOccurs="1">
2641         <xsd:annotation>
2642           <xsd:documentation>Data structure containing the new, transformed biometric
2643 information.</xsd:documentation>
2644         </xsd:annotation>
2645       </xsd:element>
2646     </xsd:sequence>
2647   </xsd:extension>
2648 </xsd:complexContent>
2649 </xsd:complexType>
2650 </xsd:element>
2651 <xsd:element name="UpdateBiographicDataRequest">
2652   <xsd:complexType>
2653     <xsd:annotation>
2654       <xsd:documentation>
2655         Updates the biographic data for a given subject record.
2656       </xsd:documentation>
2657     </xsd:annotation>
2658     <xsd:complexContent>
2659       <xsd:extension base="tns:RequestTemplate">
2660         <xsd:sequence>
2661           <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2662             <xsd:annotation>
2663               <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2664 and includes the biographic data to update.</xsd:documentation>
2665             </xsd:annotation>
2666           </xsd:element>
2667         </xsd:sequence>
2668       </xsd:extension>
2669     </xsd:complexContent>
2670   </xsd:complexType>
2671 </xsd:element>
2672 <xsd:element name="UpdateBiographicDataResponse">
2673   <xsd:complexType>
2674     <xsd:annotation>
2675       <xsd:documentation>
2676         The response to an UpdateBiographicData request.
2677       </xsd:documentation>
2678     </xsd:annotation>
2679     <xsd:complexContent>
2680       <xsd:extension base="tns:ResponseTemplate" />
2681     </xsd:complexContent>
2682   </xsd:complexType>
2683 </xsd:element>
2684 <xsd:element name="UpdateBiometricDataRequest">
2685   <xsd:complexType>
2686     <xsd:annotation>
2687       <xsd:documentation>

```

```

2688         Updates a single biometric sample for a given subject record.
2689         </xsd:documentation>
2690     </xsd:annotation>
2691     <xsd:complexContent>
2692         <xsd:extension base="tns:RequestTemplate">
2693             <xsd:sequence>
2694                 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2695                     <xsd:annotation>
2696                         <xsd:documentation>Identifies the subject or, in the encounter-centric model, a subject and an encounter,
2697 and includes the biometric data to update.</xsd:documentation>
2698                     </xsd:annotation>
2699                 </xsd:element>
2700                 <xsd:element name="Merge" type="xsd:boolean" minOccurs="0" maxOccurs="1">
2701                     <xsd:annotation>
2702                         <xsd:documentation>Value indicating if the input biometric sample should be merged with any existing
2703 biometric information.</xsd:documentation>
2704                     </xsd:annotation>
2705                 </xsd:element>
2706             </xsd:sequence>
2707         </xsd:extension>
2708     </xsd:complexContent>
2709 </xsd:complexType>
2710 </xsd:element>
2711 <xsd:element name="UpdateBiometricDataResponse">
2712     <xsd:complexType>
2713         <xsd:annotation>
2714             <xsd:documentation>
2715                 The response to an UpdateBiometricData request.
2716             </xsd:documentation>
2717         </xsd:annotation>
2718         <xsd:complexContent>
2719             <xsd:extension base="tns:ResponseTemplate" />
2720         </xsd:complexContent>
2721     </xsd:complexType>
2722 </xsd:element>
2723 <xsd:element name="VerifySubjectRequest">
2724     <xsd:complexType>
2725         <xsd:annotation>
2726             <xsd:documentation>
2727                 Performs a 1:1 verification match between a given biometric and
2728                 either a claim to identity in a given gallery or another given
2729                 biometric.
2730             </xsd:documentation>
2731         </xsd:annotation>
2732         <xsd:complexContent>
2733             <xsd:extension base="tns:RequestTemplate">
2734                 <xsd:sequence>
2735                     <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
2736                         <xsd:annotation>
2737                             <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
2738 member.</xsd:documentation>
2739                         </xsd:annotation>
2740                     </xsd:element>

```

```

2741     <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2742     <xsd:annotation>
2743     <xsd:documentation>Includes the identifying information and/or input and reference biometric
2744 samples.</xsd:documentation>
2745     </xsd:annotation>
2746     </xsd:element>
2747 </xsd:sequence>
2748 </xsd:extension>
2749 </xsd:complexContent>
2750 </xsd:complexType>
2751 </xsd:element>
2752 <xsd:element name="VerifySubjectResponse">
2753 <xsd:complexType>
2754 <xsd:annotation>
2755 <xsd:documentation>
2756 The response to a VerifySubject request.
2757 </xsd:documentation>
2758 </xsd:annotation>
2759 <xsd:complexContent>
2760 <xsd:extension base="tns:ResponseTemplate">
2761 <xsd:sequence>
2762 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
2763 <xsd:annotation>
2764 <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
2765 the Identity Claim or the Reference BIR.</xsd:documentation>
2766 </xsd:annotation>
2767 </xsd:element>
2768 <xsd:element name="Score" type="tns:Score" minOccurs="0" maxOccurs="1">
2769 <xsd:annotation>
2770 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
2771 </xsd:annotation>
2772 </xsd:element>
2773 </xsd:sequence>
2774 </xsd:extension>
2775 </xsd:complexContent>
2776 </xsd:complexType>
2777 </xsd:element>
2778 <xsd:element name="EnrollRequest">
2779 <xsd:complexType>
2780 <xsd:annotation>
2781 <xsd:documentation>
2782 The Enroll aggregate service adds a new subject or, in an
2783 encounter-centric model, a new encounter to the system. This may
2784 be accomplished in a number of different ways according to
2785 system requirements and/or resources. If the Enroll aggregate
2786 service is implemented as a synchronous service, the
2787 implementing system immediately processes the request and
2788 returns the results in the ReturnData parameter. If the Enroll
2789 aggregate service is implemented as an asynchronous service, the
2790 implementing system returns a token in the ReturnData
2791 parameter, which is an indication that the request is being
2792 handled asynchronously. In this case, the GetEnrollResults
2793 service is used to poll for the results of the Enroll request.

```

```

2794     </xsd:documentation>
2795 </xsd:annotation>
2796 <xsd:complexContent>
2797   <xsd:extension base="tns:AggregateRequestTemplate" />
2798 </xsd:complexContent>
2799 </xsd:complexType>
2800 </xsd:element>
2801 <xsd:element name="EnrollResponse">
2802   <xsd:complexType>
2803     <xsd:annotation>
2804       <xsd:documentation>The response to an Enroll request.</xsd:documentation>
2805     </xsd:annotation>
2806     <xsd:complexContent>
2807       <xsd:extension base="tns:AggregateResponseTemplate" />
2808     </xsd:complexContent>
2809   </xsd:complexType>
2810 </xsd:element>
2811 <xsd:element name="GetEnrollResultsRequest">
2812   <xsd:complexType>
2813     <xsd:annotation>
2814       <xsd:documentation>
2815         The GetEnrollResults aggregate service retrieves the enrollment
2816         results for the specified token. This service is used in
2817         conjunction with the Enroll aggregate service. If the Enroll
2818         aggregate service is implemented as an asynchronous service, the
2819         implementing system returns a token, and the GetEnrollResults
2820         service is used to poll for the results of the original Enroll
2821         request.
2822       </xsd:documentation>
2823     </xsd:annotation>
2824     <xsd:complexContent>
2825       <xsd:extension base="tns:RequestTemplate">
2826         <xsd:sequence>
2827           <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2828             <xsd:annotation>
2829               <xsd:documentation>A value used to retrieve the results of the Enroll request.</xsd:documentation>
2830             </xsd:annotation>
2831           </xsd:element>
2832         </xsd:sequence>
2833       </xsd:extension>
2834     </xsd:complexContent>
2835   </xsd:complexType>
2836 </xsd:element>
2837 <xsd:element name="GetEnrollResultsResponse">
2838   <xsd:complexType>
2839     <xsd:annotation>
2840       <xsd:documentation>The response to a GetEnrollResults request.</xsd:documentation>
2841     </xsd:annotation>
2842     <xsd:complexContent>
2843       <xsd:extension base="tns:AggregateResponseTemplate" />
2844     </xsd:complexContent>
2845   </xsd:complexType>
2846 </xsd:element>

```

```

2847 <xsd:element name="GetIdentifyResultsRequest">
2848   <xsd:complexType>
2849     <xsd:annotation>
2850       <xsd:documentation>
2851         The GetIdentifyResults aggregate service retrieves the
2852         identification results for the specified token. This service is
2853         used in conjunction with the Identify aggregate service. If the
2854         Identify aggregate service is implemented as an asynchronous
2855         service, the implementing system returns a token, and the
2856         GetIdentifyResults service is used to poll for the results of
2857         the original Identify request.
2858       </xsd:documentation>
2859     </xsd:annotation>
2860     <xsd:complexContent>
2861       <xsd:extension base="tns:RequestTemplate">
2862         <xsd:sequence>
2863           <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">
2864             <xsd:annotation>
2865               <xsd:documentation>A value used to retrieve the results of the Identify request.</xsd:documentation>
2866             </xsd:annotation>
2867           </xsd:element>
2868         </xsd:sequence>
2869       </xsd:extension>
2870     </xsd:complexContent>
2871   </xsd:complexType>
2872 </xsd:element>
2873 <xsd:element name="GetIdentifyResultsResponse">
2874   <xsd:complexType>
2875     <xsd:annotation>
2876       <xsd:documentation>The response to a GetIdentifyResults request.</xsd:documentation>
2877     </xsd:annotation>
2878     <xsd:complexContent>
2879       <xsd:extension base="tns:AggregateResponseTemplate" />
2880     </xsd:complexContent>
2881   </xsd:complexType>
2882 </xsd:element>
2883 <xsd:element name="GetVerifyResultsRequest">
2884   <xsd:complexType>
2885     <xsd:annotation>
2886       <xsd:documentation>
2887         The GetVerifyResults aggregate service retrieves the verification
2888         results for the specified token. This service is used in
2889         conjunction with the Verify aggregate service. If the Verify
2890         aggregate service is implemented as an asynchronous service, the
2891         implementing system returns a token, and the GetVerifyResults
2892         service is used to poll for the results of the original Verify
2893         request.
2894       </xsd:documentation>
2895     </xsd:annotation>
2896     <xsd:complexContent>
2897       <xsd:extension base="tns:RequestTemplate">
2898         <xsd:sequence>
2899           <xsd:element name="Token" type="tns:TokenType" minOccurs="1" maxOccurs="1">

```

```

2900         <xsd:annotation>
2901         <xsd:documentation>A value used to retrieve the results of the Verify request.</xsd:documentation>
2902         </xsd:annotation>
2903     </xsd:element>
2904 </xsd:sequence>
2905 </xsd:extension>
2906 </xsd:complexContent>
2907 </xsd:complexType>
2908 </xsd:element>
2909 <xsd:element name="GetVerifyResultsResponse">
2910 <xsd:complexType>
2911 <xsd:annotation>
2912 <xsd:documentation>The response to a GetVerifyResults request.</xsd:documentation>
2913 </xsd:annotation>
2914 <xsd:complexContent>
2915 <xsd:extension base="tns:AggregateResponseTemplate">
2916 <xsd:sequence>
2917 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
2918 <xsd:annotation>
2919 <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
2920 the Identity Claim or the Reference BIR.</xsd:documentation>
2921 </xsd:annotation>
2922 </xsd:element>
2923 <xsd:element name="Score" type="Score" minOccurs="0" maxOccurs="1">
2924 <xsd:annotation>
2925 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
2926 </xsd:annotation>
2927 </xsd:element>
2928 </xsd:sequence>
2929 </xsd:extension>
2930 </xsd:complexContent>
2931 </xsd:complexType>
2932 </xsd:element>
2933 <xsd:element name="IdentifyRequest">
2934 <xsd:complexType>
2935 <xsd:annotation>
2936 <xsd:documentation>
2937 The Identify aggregate service performs an identification
2938 function according to system requirements and/or resources. If
2939 the Identify aggregate service is implemented as a synchronous
2940 service, the implementing system immediately processes the
2941 request and returns the results in the ReturnData parameter. If
2942 the Identify aggregate service is implemented as an asynchronous
2943 service, the implementing system returns a token in the
2944 ReturnData parameter, which is an indication that the request is
2945 being handled asynchronously. In this case, the
2946 GetIdentifyResults service is used to poll for the results of
2947 the Identify request.
2948 </xsd:documentation>
2949 </xsd:annotation>
2950 <xsd:complexContent>
2951 <xsd:extension base="tns:AggregateRequestTemplate" />
2952 </xsd:complexContent>

```



```

2953     </xsd:complexType>
2954 </xsd:element>
2955 <xsd:element name="IdentifyResponse">
2956   <xsd:complexType>
2957     <xsd:annotation>
2958       <xsd:documentation>The response to an Identify request.</xsd:documentation>
2959     </xsd:annotation>
2960     <xsd:complexContent>
2961       <xsd:extension base="tns:AggregateResponseTemplate" />
2962     </xsd:complexContent>
2963   </xsd:complexType>
2964 </xsd:element>
2965 <xsd:element name="RetrieveInformationRequest">
2966   <xsd:complexType>
2967     <xsd:annotation>
2968       <xsd:documentation>
2969         The RetrieveInformation aggregate service retrieves requested
2970         information about a subject, or in an encounter-centric model
2971         about an encounter. In a person-centric model, this aggregate
2972         service may be used to retrieve both biographic and biometric
2973         information for a subject record. In an encounter-centric model,
2974         this aggregate service may be used to retrieve biographic and/or
2975         biometric information for either a single encounter or all
2976         encounters. Either a SubjectID or EncounterID must be specified
2977         in the Identify parameter.
2978       </xsd:documentation>
2979     </xsd:annotation>
2980     <xsd:complexContent>
2981       <xsd:extension base="tns:RequestTemplate">
2982         <xsd:sequence>
2983           <xsd:element name="ProcessingOptions" type="tns:ProcessingOptionsType" minOccurs="1" maxOccurs="1">
2984             <xsd:annotation>
2985               <xsd:documentation>Options that guide how the service request is processed, and may identify what
2986 type(s) of information should be returned.</xsd:documentation>
2987             </xsd:annotation>
2988           </xsd:element>
2989           <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
2990             <xsd:annotation>
2991               <xsd:documentation>Includes the identifier of the subject or encounter.</xsd:documentation>
2992             </xsd:annotation>
2993           </xsd:element>
2994         </xsd:sequence>
2995       </xsd:extension>
2996     </xsd:complexContent>
2997   </xsd:complexType>
2998 </xsd:element>
2999 <xsd:element name="RetrieveInformationResponse">
3000   <xsd:complexType>
3001     <xsd:annotation>
3002       <xsd:documentation>The response to a RetrieveInformation request.</xsd:documentation>
3003     </xsd:annotation>
3004     <xsd:complexContent>
3005       <xsd:extension base="tns:AggregateResponseTemplate" />

```

```

3006     </xsd:complexContent>
3007 </xsd:complexType>
3008 </xsd:element>
3009 <xsd:element name="VerifyRequest">
3010     <xsd:complexType>
3011         <xsd:annotation>
3012             <xsd:documentation>
3013                 The Verify aggregate service performs a 1:1 verification
3014                 function according to system requirements and/or resources.
3015                 Either the IdentityClaim or ReferenceBIR input data elements in
3016                 the Identity parameter are required. If the Verify aggregate
3017                 service is implemented as a synchronous service, the
3018                 implementing system immediately processes the request and returns
3019                 the results in the ReturnData parameter. If the Verify aggregate
3020                 service is implemented as an asynchronous service, the
3021                 implementing system returns a token in the ReturnData parameter,
3022                 which is an indication that the request is being handled
3023                 asynchronously. In this case, the GetVerifyResults service is
3024                 used to poll for the results of the Verify request.
3025             </xsd:documentation>
3026         </xsd:annotation>
3027     <xsd:complexContent>
3028         <xsd:extension base="tns:AggregateRequestTemplate">
3029             <xsd:sequence>
3030                 <xsd:element name="Identity" type="tns:BIASIdentity" minOccurs="1" maxOccurs="1">
3031                     <xsd:annotation>
3032                         <xsd:documentation>Includes either the IdentityClaim or ReferenceBIR.</xsd:documentation>
3033                     </xsd:annotation>
3034                 </xsd:element>
3035                 <xsd:element name="GalleryID" type="tns:BIASIDType" minOccurs="0" maxOccurs="1">
3036                     <xsd:annotation>
3037                         <xsd:documentation>The identifier of the gallery or population group of which the subject must be a
3038                         member.</xsd:documentation>
3039                     </xsd:annotation>
3040                 </xsd:element>
3041             </xsd:sequence>
3042         </xsd:extension>
3043     </xsd:complexContent>
3044 </xsd:complexType>
3045 </xsd:element>
3046 <xsd:element name="VerifyResponse">
3047     <xsd:complexType>
3048         <xsd:annotation>
3049             <xsd:documentation>The response to a Verify request.</xsd:documentation>
3050         </xsd:annotation>
3051     <xsd:complexContent>
3052         <xsd:extension base="tns:AggregateResponseTemplate">
3053             <xsd:sequence>
3054                 <xsd:element name="Match" type="xsd:boolean" minOccurs="0" maxOccurs="1">
3055                     <xsd:annotation>
3056                         <xsd:documentation>Indicates if the Input BIR matched either the biometric information associated with
3057                         the Identity Claim or the Reference BIR.</xsd:documentation>
3058                     </xsd:annotation>

```

```

3059         </xsd:element>
3060         <xsd:element name="Score" type="Score" minOccurs="0" maxOccurs="1">
3061             <xsd:annotation>
3062                 <xsd:documentation>The score if the biometric information matched.</xsd:documentation>
3063             </xsd:annotation>
3064         </xsd:element>
3065     </xsd:sequence>
3066 </xsd:extension>
3067 </xsd:complexContent>
3068 </xsd:complexType>
3069 </xsd:element>
3070 <xsd:element name="BIASFault" type="tns:BIASFaultDetail" />
3071 </xsd:schema>
3072 </wsdl:types>
3073 <wsdl:message name="AddSubjectToGalleryRequestMessage">
3074     <wsdl:part name="AddSubjectToGalleryRequest" element="AddSubjectToGalleryRequest" />
3075 </wsdl:message>
3076 <wsdl:message name="AddSubjectToGalleryResponseMessage">
3077     <wsdl:part name="AddSubjectToGalleryResponse" element="AddSubjectToGalleryResponse" />
3078 </wsdl:message>
3079 <wsdl:message name="CheckQualityRequestMessage">
3080     <wsdl:part name="CheckQualityRequest" element="CheckQualityRequest" />
3081 </wsdl:message>
3082 <wsdl:message name="CheckQualityResponseMessage">
3083     <wsdl:part name="CheckQualityResponse" element="CheckQualityResponse" />
3084 </wsdl:message>
3085 <wsdl:message name="ClassifyBiometricDataRequestMessage">
3086     <wsdl:part name="ClassifyBiometricDataRequest" element="ClassifyBiometricDataRequest" />
3087 </wsdl:message>
3088 <wsdl:message name="ClassifyBiometricDataResponseMessage">
3089     <wsdl:part name="ClassifyBiometricDataResponse" element="ClassifyBiometricDataResponse" />
3090 </wsdl:message>
3091 <wsdl:message name="CreateSubjectRequestMessage">
3092     <wsdl:part name="CreateSubjectRequest" element="CreateSubjectRequest" />
3093 </wsdl:message>
3094 <wsdl:message name="CreateSubjectResponseMessage">
3095     <wsdl:part name="CreateSubjectResponse" element="CreateSubjectResponse" />
3096 </wsdl:message>
3097 <wsdl:message name="DeleteBiographicDataRequestMessage">
3098     <wsdl:part name="DeleteBiographicDataRequest" element="DeleteBiographicDataRequest" />
3099 </wsdl:message>
3100 <wsdl:message name="DeleteBiographicDataResponseMessage">
3101     <wsdl:part name="DeleteBiographicDataResponse" element="DeleteBiographicDataResponse" />
3102 </wsdl:message>
3103 <wsdl:message name="DeleteBiometricDataRequestMessage">
3104     <wsdl:part name="DeleteBiometricDataRequest" element="DeleteBiometricDataRequest" />
3105 </wsdl:message>
3106 <wsdl:message name="DeleteBiometricDataResponseMessage">
3107     <wsdl:part name="DeleteBiometricDataResponse" element="DeleteBiometricDataResponse" />
3108 </wsdl:message>
3109 <wsdl:message name="DeleteSubjectRequestMessage">
3110     <wsdl:part name="DeleteSubjectRequest" element="DeleteSubjectRequest" />
3111 </wsdl:message>

```

```

3112 <wsdl:message name="DeleteSubjectResponseMessage">
3113   <wsdl:part name="DeleteSubjectResponse" element="DeleteSubjectResponse" />
3114 </wsdl:message>
3115 <wsdl:message name="DeleteSubjectFromGalleryRequestMessage">
3116   <wsdl:part name="DeleteSubjectFromGalleryRequest" element="DeleteSubjectFromGalleryRequest" />
3117 </wsdl:message>
3118 <wsdl:message name="DeleteSubjectFromGalleryResponseMessage">
3119   <wsdl:part name="DeleteSubjectFromGalleryResponse" element="DeleteSubjectFromGalleryResponse" />
3120 </wsdl:message>
3121 <wsdl:message name="GetIdentifySubjectResultsRequestMessage">
3122   <wsdl:part name="GetIdentifySubjectResultsRequest" element="GetIdentifySubjectResultsRequest" />
3123 </wsdl:message>
3124 <wsdl:message name="GetIdentifySubjectResultsResponseMessage">
3125   <wsdl:part name="GetIdentifySubjectResultsResponse" element="GetIdentifySubjectResultsResponse" />
3126 </wsdl:message>
3127 <wsdl:message name="IdentifySubjectRequestMessage">
3128   <wsdl:part name="IdentifySubjectRequest" element="IdentifySubjectRequest" />
3129 </wsdl:message>
3130 <wsdl:message name="IdentifySubjectResponseMessage">
3131   <wsdl:part name="IdentifySubjectResponse" element="IdentifySubjectResponse" />
3132 </wsdl:message>
3133 <wsdl:message name="ListBiographicDataRequestMessage">
3134   <wsdl:part name="ListBiographicDataRequest" element="ListBiographicDataRequest" />
3135 </wsdl:message>
3136 <wsdl:message name="ListBiographicDataResponseMessage">
3137   <wsdl:part name="ListBiographicDataResponse" element="ListBiographicDataResponse" />
3138 </wsdl:message>
3139 <wsdl:message name="ListBiometricDataRequestMessage">
3140   <wsdl:part name="ListBiometricDataRequest" element="ListBiometricDataRequest" />
3141 </wsdl:message>
3142 <wsdl:message name="ListBiometricDataResponseMessage">
3143   <wsdl:part name="ListBiometricDataResponse" element="ListBiometricDataResponse" />
3144 </wsdl:message>
3145 <wsdl:message name="PerformFusionRequestMessage">
3146   <wsdl:part name="PerformFusionRequest" element="PerformFusionRequest" />
3147 </wsdl:message>
3148 <wsdl:message name="PerformFusionResponseMessage">
3149   <wsdl:part name="PerformFusionResponse" element="PerformFusionResponse" />
3150 </wsdl:message>
3151 <wsdl:message name="QueryCapabilitiesRequestMessage">
3152   <wsdl:part name="QueryCapabilitiesRequest" element="QueryCapabilitiesRequest" />
3153 </wsdl:message>
3154 <wsdl:message name="QueryCapabilitiesResponseMessage">
3155   <wsdl:part name="QueryCapabilitiesResponse" element="QueryCapabilitiesResponse" />
3156 </wsdl:message>
3157 <wsdl:message name="RetrieveBiographicInformationRequestMessage">
3158   <wsdl:part name="RetrieveBiographicInformationRequest" element="RetrieveBiographicInformationRequest" />
3159 </wsdl:message>
3160 <wsdl:message name="RetrieveBiographicInformationResponseMessage">
3161   <wsdl:part name="RetrieveBiographicInformationResponse" element="RetrieveBiographicInformationResponse" />
3162 </wsdl:message>
3163 <wsdl:message name="RetrieveBiometricInformationRequestMessage">
3164   <wsdl:part name="RetrieveBiometricInformationRequest" element="RetrieveBiometricInformationRequest" />

```

```

3165 </wsdl:message>
3166 <wsdl:message name="RetrieveBiometricInformationResponseMessage">
3167   <wsdl:part name="RetrieveBiometricInformationResponse" element="RetrieveBiometricInformationResponse" />
3168 </wsdl:message>
3169 <wsdl:message name="SetBiographicDataRequestMessage">
3170   <wsdl:part name="SetBiographicDataRequest" element="SetBiographicDataRequest" />
3171 </wsdl:message>
3172 <wsdl:message name="SetBiographicDataResponseMessage">
3173   <wsdl:part name="SetBiographicDataResponse" element="SetBiographicDataResponse" />
3174 </wsdl:message>
3175 <wsdl:message name="SetBiometricDataRequestMessage">
3176   <wsdl:part name="SetBiometricDataRequest" element="SetBiometricDataRequest" />
3177 </wsdl:message>
3178 <wsdl:message name="SetBiometricDataResponseMessage">
3179   <wsdl:part name="SetBiometricDataResponse" element="SetBiometricDataResponse" />
3180 </wsdl:message>
3181 <wsdl:message name="TransformBiometricDataRequestMessage">
3182   <wsdl:part name="TransformBiometricDataRequest" element="TransformBiometricDataRequest" />
3183 </wsdl:message>
3184 <wsdl:message name="TransformBiometricDataResponseMessage">
3185   <wsdl:part name="TransformBiometricDataResponse" element="TransformBiometricDataResponse" />
3186 </wsdl:message>
3187 <wsdl:message name="UpdateBiographicDataRequestMessage">
3188   <wsdl:part name="UpdateBiographicDataRequest" element="UpdateBiographicDataRequest" />
3189 </wsdl:message>
3190 <wsdl:message name="UpdateBiographicDataResponseMessage">
3191   <wsdl:part name="UpdateBiographicDataResponse" element="UpdateBiographicDataResponse" />
3192 </wsdl:message>
3193 <wsdl:message name="UpdateBiometricDataRequestMessage">
3194   <wsdl:part name="UpdateBiometricDataRequest" element="UpdateBiometricDataRequest" />
3195 </wsdl:message>
3196 <wsdl:message name="UpdateBiometricDataResponseMessage">
3197   <wsdl:part name="UpdateBiometricDataResponse" element="UpdateBiometricDataResponse" />
3198 </wsdl:message>
3199 <wsdl:message name="VerifySubjectRequestMessage">
3200   <wsdl:part name="VerifySubjectRequest" element="VerifySubjectRequest" />
3201 </wsdl:message>
3202 <wsdl:message name="VerifySubjectResponseMessage">
3203   <wsdl:part name="VerifySubjectResponse" element="VerifySubjectResponse" />
3204 </wsdl:message>
3205 <wsdl:message name="EnrollRequestMessage">
3206   <wsdl:part name="EnrollRequest" element="EnrollRequest" />
3207 </wsdl:message>
3208 <wsdl:message name="EnrollResponseMessage">
3209   <wsdl:part name="EnrollResponse" element="EnrollResponse" />
3210 </wsdl:message>
3211 <wsdl:message name="GetEnrollResultsRequestMessage">
3212   <wsdl:part name="GetEnrollResultsRequest" element="GetEnrollResultsRequest" />
3213 </wsdl:message>
3214 <wsdl:message name="GetEnrollResultsResponseMessage">
3215   <wsdl:part name="GetEnrollResultsResponse" element="GetEnrollResultsResponse" />
3216 </wsdl:message>
3217 <wsdl:message name="GetIdentifyResultsRequestMessage">

```

```

3218     <wsdl:part name="GetIdentifyResultsRequest" element="GetIdentifyResultsRequest" />
3219 </wsdl:message>
3220 <wsdl:message name="GetIdentifyResultsResponseMessage">
3221     <wsdl:part name="GetIdentifyResultsResponse" element="GetIdentifyResultsResponse" />
3222 </wsdl:message>
3223 <wsdl:message name="GetVerifyResultsRequestMessage">
3224     <wsdl:part name="GetVerifyResultsRequest" element="GetVerifyResultsRequest" />
3225 </wsdl:message>
3226 <wsdl:message name="GetVerifyResultsResponseMessage">
3227     <wsdl:part name="GetVerifyResultsResponse" element="GetVerifyResultsResponse" />
3228 </wsdl:message>
3229 <wsdl:message name="IdentifyRequestMessage">
3230     <wsdl:part name="IdentifyRequest" element="IdentifyRequest" />
3231 </wsdl:message>
3232 <wsdl:message name="IdentifyResponseMessage">
3233     <wsdl:part name="IdentifyResponse" element="IdentifyResponse" />
3234 </wsdl:message>
3235 <wsdl:message name="RetrieveInformationRequestMessage">
3236     <wsdl:part name="RetrieveInformationRequest" element="RetrieveInformationRequest" />
3237 </wsdl:message>
3238 <wsdl:message name="RetrieveInformationResponseMessage">
3239     <wsdl:part name="RetrieveInformationResponse" element="RetrieveInformationResponse" />
3240 </wsdl:message>
3241 <wsdl:message name="VerifyRequestMessage">
3242     <wsdl:part name="VerifyRequest" element="VerifyRequest" />
3243 </wsdl:message>
3244 <wsdl:message name="VerifyResponseMessage">
3245     <wsdl:part name="VerifyResponse" element="VerifyResponse" />
3246 </wsdl:message>
3247 <wsdl:portType name="BIAS_v1">
3248     <wsdl:operation name="AddSubjectToGallery">
3249         <wsdl:input message="tns:AddSubjectToGalleryRequestMessage" name="AddSubjectToGalleryRequest" />
3250         <wsdl:output message="tns:AddSubjectToGalleryResponseMessage" name="AddSubjectToGalleryResponse" />
3251     </wsdl:operation>
3252     <wsdl:operation name="CheckQuality">
3253         <wsdl:input message="tns:CheckQualityRequestMessage" name="CheckQualityRequest" />
3254         <wsdl:output message="tns:CheckQualityResponseMessage" name="CheckQualityResponse" />
3255     </wsdl:operation>
3256     <wsdl:operation name="ClassifyBiometricData">
3257         <wsdl:input message="tns:ClassifyBiometricDataRequestMessage" name="ClassifyBiometricDataRequest" />
3258         <wsdl:output message="tns:ClassifyBiometricDataResponseMessage" name="ClassifyBiometricDataResponse" />
3259     </wsdl:operation>
3260     <wsdl:operation name="CreateSubject">
3261         <wsdl:input message="tns:CreateSubjectRequestMessage" name="CreateSubjectRequest" />
3262         <wsdl:output message="tns:CreateSubjectResponseMessage" name="CreateSubjectResponse" />
3263     </wsdl:operation>
3264     <wsdl:operation name="DeleteBiographicData">
3265         <wsdl:input message="tns:DeleteBiographicDataRequestMessage" name="DeleteBiographicDataRequest" />
3266         <wsdl:output message="tns:DeleteBiographicDataResponseMessage" name="DeleteBiographicDataResponse" />
3267     </wsdl:operation>
3268     <wsdl:operation name="DeleteBiometricData">
3269         <wsdl:input message="tns:DeleteBiometricDataRequestMessage" name="DeleteBiometricDataRequest" />
3270         <wsdl:output message="tns:DeleteBiometricDataResponseMessage" name="DeleteBiometricDataResponse" />

```

```

3271     </wsdl:operation>
3272     <wsdl:operation name="DeleteSubject">
3273         <wsdl:input message="tns:DeleteSubjectRequestMessage" name="DeleteSubjectRequest" />
3274         <wsdl:output message="tns:DeleteSubjectResponseMessage" name="DeleteSubjectResponse" />
3275     </wsdl:operation>
3276     <wsdl:operation name="DeleteSubjectFromGallery">
3277         <wsdl:input message="tns:DeleteSubjectFromGalleryRequestMessage" name="DeleteSubjectFromGalleryRequest"
3278 />
3279         <wsdl:output message="tns:DeleteSubjectFromGalleryResponseMessage"
3280 name="DeleteSubjectFromGalleryResponse" />
3281     </wsdl:operation>
3282     <wsdl:operation name="GetIdentifySubjectResults">
3283         <wsdl:input message="tns:GetIdentifySubjectResultsRequestMessage" name="GetIdentifySubjectResultsRequest"
3284 />
3285         <wsdl:output message="tns:GetIdentifySubjectResultsResponseMessage"
3286 name="GetIdentifySubjectResultsResponse" />
3287     </wsdl:operation>
3288     <wsdl:operation name="IdentifySubject">
3289         <wsdl:input message="tns:IdentifySubjectRequestMessage" name="IdentifySubjectRequest" />
3290         <wsdl:output message="tns:IdentifySubjectResponseMessage" name="IdentifySubjectResponse" />
3291     </wsdl:operation>
3292     <wsdl:operation name="ListBiographicData">
3293         <wsdl:input message="tns:ListBiographicDataRequestMessage" name="ListBiographicDataRequest" />
3294         <wsdl:output message="tns:ListBiographicDataResponseMessage" name="ListBiographicDataResponse" />
3295     </wsdl:operation>
3296     <wsdl:operation name="ListBiometricData">
3297         <wsdl:input message="tns:ListBiometricDataRequestMessage" name="ListBiometricDataRequest" />
3298         <wsdl:output message="tns:ListBiometricDataResponseMessage" name="ListBiometricDataResponse" />
3299     </wsdl:operation>
3300     <wsdl:operation name="PerformFusion">
3301         <wsdl:input message="tns:PerformFusionRequestMessage" name="PerformFusionRequest" />
3302         <wsdl:output message="tns:PerformFusionResponseMessage" name="PerformFusionResponse" />
3303     </wsdl:operation>
3304     <wsdl:operation name="QueryCapabilities">
3305         <wsdl:input message="tns:QueryCapabilitiesRequestMessage" name="QueryCapabilitiesRequest" />
3306         <wsdl:output message="tns:QueryCapabilitiesResponseMessage" name="QueryCapabilitiesResponse" />
3307     </wsdl:operation>
3308     <wsdl:operation name="RetrieveBiographicInformation">
3309         <wsdl:input message="tns:RetrieveBiographicInformationRequestMessage"
3310 name="RetrieveBiographicInformationRequest" />
3311         <wsdl:output message="tns:RetrieveBiographicInformationResponseMessage"
3312 name="RetrieveBiographicInformationResponse" />
3313     </wsdl:operation>
3314     <wsdl:operation name="RetrieveBiometricInformation">
3315         <wsdl:input message="tns:RetrieveBiometricInformationRequestMessage"
3316 name="RetrieveBiometricInformationRequest" />
3317         <wsdl:output message="tns:RetrieveBiometricInformationResponseMessage"
3318 name="RetrieveBiometricInformationResponse" />
3319     </wsdl:operation>
3320     <wsdl:operation name="SetBiographicData">
3321         <wsdl:input message="tns:SetBiographicDataRequestMessage" name="SetBiographicDataRequest" />
3322         <wsdl:output message="tns:SetBiographicDataResponseMessage" name="SetBiographicDataResponse" />
3323     </wsdl:operation>

```

```

3324 <wsdl:operation name="SetBiometricData">
3325   <wsdl:input message="tns:SetBiometricDataRequestMessage" name="SetBiometricDataRequest" />
3326   <wsdl:output message="tns:SetBiometricDataResponseMessage" name="SetBiometricDataResponse" />
3327 </wsdl:operation>
3328 <wsdl:operation name="TransformBiometricData">
3329   <wsdl:input message="tns:TransformBiometricDataRequestMessage" name="TransformBiometricDataRequest" />
3330   <wsdl:output message="tns:TransformBiometricDataResponseMessage"
3331 name="TransformBiometricDataResponse" />
3332 </wsdl:operation>
3333 <wsdl:operation name="UpdateBiographicData">
3334   <wsdl:input message="tns:UpdateBiographicDataRequestMessage" name="UpdateBiographicDataRequest" />
3335   <wsdl:output message="tns:UpdateBiographicDataResponseMessage" name="UpdateBiographicDataResponse" />
3336 </wsdl:operation>
3337 <wsdl:operation name="UpdateBiometricData">
3338   <wsdl:input message="tns:UpdateBiometricDataRequestMessage" name="UpdateBiometricDataRequest" />
3339   <wsdl:output message="tns:UpdateBiometricDataResponseMessage" name="UpdateBiometricDataResponse" />
3340 </wsdl:operation>
3341 <wsdl:operation name="VerifySubject">
3342   <wsdl:input message="tns:VerifySubjectRequestMessage" name="VerifySubjectRequest" />
3343   <wsdl:output message="tns:VerifySubjectResponseMessage" name="VerifySubjectResponse" />
3344 </wsdl:operation>
3345 <wsdl:operation name="Enroll">
3346   <wsdl:input message="tns:EnrollRequestMessage" name="EnrollRequest" />
3347   <wsdl:output message="tns:EnrollResponseMessage" name="EnrollResponse" />
3348 </wsdl:operation>
3349 <wsdl:operation name="GetEnrollResults">
3350   <wsdl:input message="tns:GetEnrollResultsRequestMessage" name="GetEnrollResultsRequest" />
3351   <wsdl:output message="tns:GetEnrollResultsResponseMessage" name="GetEnrollResultsResponse" />
3352 </wsdl:operation>
3353 <wsdl:operation name="GetIdentifyResults">
3354   <wsdl:input message="tns:GetIdentifyResultsRequestMessage" name="GetIdentifyResultsRequest" />
3355   <wsdl:output message="tns:GetIdentifyResultsResponseMessage" name="GetIdentifyResultsResponse" />
3356 </wsdl:operation>
3357 <wsdl:operation name="GetVerifyResults">
3358   <wsdl:input message="tns:GetVerifyResultsRequestMessage" name="GetVerifyResultsRequest" />
3359   <wsdl:output message="tns:GetVerifyResultsResponseMessage" name="GetVerifyResultsResponse" />
3360 </wsdl:operation>
3361 <wsdl:operation name="Identify">
3362   <wsdl:input message="tns:IdentifyRequestMessage" name="IdentifyRequest" />
3363   <wsdl:output message="tns:IdentifyResponseMessage" name="IdentifyResponse" />
3364 </wsdl:operation>
3365 <wsdl:operation name="RetrieveInformation">
3366   <wsdl:input message="tns:RetrieveInformationRequestMessage" name="RetrieveInformationRequest" />
3367   <wsdl:output message="tns:RetrieveInformationResponseMessage" name="RetrieveInformationResponse" />
3368 </wsdl:operation>
3369 <wsdl:operation name="Verify">
3370   <wsdl:input message="tns:VerifyRequestMessage" name="VerifyRequest" />
3371   <wsdl:output message="tns:VerifyResponseMessage" name="VerifyResponse" />
3372 </wsdl:operation>
3373 </wsdl:portType>
3374 <wsdl:binding type="tns:BIAS_v1" name="BIAS_v1HttpBinding">
3375   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
3376   <wsdl:operation name="AddSubjectToGallery">

```



```

3377     <soap:operation soapAction="AddSubjectToGallery" />
3378     <wsdl:input name="AddSubjectToGalleryRequest">
3379         <soap:body use="literal" />
3380     </wsdl:input>
3381     <wsdl:output name="AddSubjectToGalleryResponse">
3382         <soap:body use="literal" />
3383     </wsdl:output>
3384     <wsdl:fault name="BIASFault">
3385         <soap:fault name="BIASFault" use="literal" />
3386     </wsdl:fault>
3387 </wsdl:operation>
3388 <wsdl:operation name="CheckQuality">
3389     <soap:operation soapAction="CheckQuality" />
3390     <wsdl:input name="CheckQualityRequest">
3391         <soap:body use="literal" />
3392     </wsdl:input>
3393     <wsdl:output name="CheckQualityResponse">
3394         <soap:body use="literal" />
3395     </wsdl:output>
3396     <wsdl:fault name="BIASFault">
3397         <soap:fault name="BIASFault" use="literal" />
3398     </wsdl:fault>
3399 </wsdl:operation>
3400 <wsdl:operation name="ClassifyBiometricData">
3401     <soap:operation soapAction="ClassifyBiometricData" />
3402     <wsdl:input name="ClassifyBiometricDataRequest">
3403         <soap:body use="literal" />
3404     </wsdl:input>
3405     <wsdl:output name="ClassifyBiometricDataResponse">
3406         <soap:body use="literal" />
3407     </wsdl:output>
3408     <wsdl:fault name="BIASFault">
3409         <soap:fault name="BIASFault" use="literal" />
3410     </wsdl:fault>
3411 </wsdl:operation>
3412 <wsdl:operation name="CreateSubject">
3413     <soap:operation soapAction="CreateSubject" />
3414     <wsdl:input name="CreateSubjectRequest">
3415         <soap:body use="literal" />
3416     </wsdl:input>
3417     <wsdl:output name="CreateSubjectResponse">
3418         <soap:body use="literal" />
3419     </wsdl:output>
3420     <wsdl:fault name="BIASFault">
3421         <soap:fault name="BIASFault" use="literal" />
3422     </wsdl:fault>
3423 </wsdl:operation>
3424 <wsdl:operation name="DeleteBiographicData">
3425     <soap:operation soapAction="DeleteBiographicData" />
3426     <wsdl:input name="DeleteBiographicDataRequest">
3427         <soap:body use="literal" />
3428     </wsdl:input>
3429     <wsdl:output name="DeleteBiographicDataResponse">

```

```

3430     <soap:body use="literal" />
3431 </wsdl:output>
3432 <wsdl:fault name="BIASFault">
3433     <soap:fault name="BIASFault" use="literal" />
3434 </wsdl:fault>
3435 </wsdl:operation>
3436 <wsdl:operation name="DeleteBiometricData">
3437     <soap:operation soapAction="DeleteBiometricData" />
3438     <wsdl:input name="DeleteBiometricDataRequest">
3439         <soap:body use="literal" />
3440     </wsdl:input>
3441     <wsdl:output name="DeleteBiometricDataResponse">
3442         <soap:body use="literal" />
3443     </wsdl:output>
3444     <wsdl:fault name="BIASFault">
3445         <soap:fault name="BIASFault" use="literal" />
3446     </wsdl:fault>
3447 </wsdl:operation>
3448 <wsdl:operation name="DeleteSubject">
3449     <soap:operation soapAction="DeleteSubject" />
3450     <wsdl:input name="DeleteSubjectRequest">
3451         <soap:body use="literal" />
3452     </wsdl:input>
3453     <wsdl:output name="DeleteSubjectResponse">
3454         <soap:body use="literal" />
3455     </wsdl:output>
3456     <wsdl:fault name="BIASFault">
3457         <soap:fault name="BIASFault" use="literal" />
3458     </wsdl:fault>
3459 </wsdl:operation>
3460 <wsdl:operation name="DeleteSubjectFromGallery">
3461     <soap:operation soapAction="DeleteSubjectFromGallery" />
3462     <wsdl:input name="DeleteSubjectFromGalleryRequest">
3463         <soap:body use="literal" />
3464     </wsdl:input>
3465     <wsdl:output name="DeleteSubjectFromGalleryResponse">
3466         <soap:body use="literal" />
3467     </wsdl:output>
3468     <wsdl:fault name="BIASFault">
3469         <soap:fault name="BIASFault" use="literal" />
3470     </wsdl:fault>
3471 </wsdl:operation>
3472 <wsdl:operation name="GetIdentifySubjectResults">
3473     <soap:operation soapAction="GetIdentifySubjectResults" />
3474     <wsdl:input name="GetIdentifySubjectResultsRequest">
3475         <soap:body use="literal" />
3476     </wsdl:input>
3477     <wsdl:output name="GetIdentifySubjectResultsResponse">
3478         <soap:body use="literal" />
3479     </wsdl:output>
3480     <wsdl:fault name="BIASFault">
3481         <soap:fault name="BIASFault" use="literal" />
3482     </wsdl:fault>

```

```

3483 </wsdl:operation>
3484 <wsdl:operation name="IdentifySubject">
3485   <soap:operation soapAction="IdentifySubject" />
3486   <wsdl:input name="IdentifySubjectRequest">
3487     <soap:body use="literal" />
3488   </wsdl:input>
3489   <wsdl:output name="IdentifySubjectResponse">
3490     <soap:body use="literal" />
3491   </wsdl:output>
3492   <wsdl:fault name="BIASFault">
3493     <soap:fault name="BIASFault" use="literal" />
3494   </wsdl:fault>
3495 </wsdl:operation>
3496 <wsdl:operation name="ListBiographicData">
3497   <soap:operation soapAction="ListBiographicData" />
3498   <wsdl:input name="ListBiographicDataRequest">
3499     <soap:body use="literal" />
3500   </wsdl:input>
3501   <wsdl:output name="ListBiographicDataResponse">
3502     <soap:body use="literal" />
3503   </wsdl:output>
3504   <wsdl:fault name="BIASFault">
3505     <soap:fault name="BIASFault" use="literal" />
3506   </wsdl:fault>
3507 </wsdl:operation>
3508 <wsdl:operation name="ListBiometricData">
3509   <soap:operation soapAction="ListBiometricData" />
3510   <wsdl:input name="ListBiometricDataRequest">
3511     <soap:body use="literal" />
3512   </wsdl:input>
3513   <wsdl:output name="ListBiometricDataResponse">
3514     <soap:body use="literal" />
3515   </wsdl:output>
3516   <wsdl:fault name="BIASFault">
3517     <soap:fault name="BIASFault" use="literal" />
3518   </wsdl:fault>
3519 </wsdl:operation>
3520 <wsdl:operation name="PerformFusion">
3521   <soap:operation soapAction="PerformFusion" />
3522   <wsdl:input name="PerformFusionRequest">
3523     <soap:body use="literal" />
3524   </wsdl:input>
3525   <wsdl:output name="PerformFusionResponse">
3526     <soap:body use="literal" />
3527   </wsdl:output>
3528   <wsdl:fault name="BIASFault">
3529     <soap:fault name="BIASFault" use="literal" />
3530   </wsdl:fault>
3531 </wsdl:operation>
3532 <wsdl:operation name="QueryCapabilities">
3533   <soap:operation soapAction="QueryCapabilities" />
3534   <wsdl:input name="QueryCapabilitiesRequest">
3535     <soap:body use="literal" />

```

```

3536     </wsdl:input>
3537     <wsdl:output name="QueryCapabilitiesResponse">
3538         <soap:body use="literal" />
3539     </wsdl:output>
3540     <wsdl:fault name="BIASFault">
3541         <soap:fault name="BIASFault" use="literal" />
3542     </wsdl:fault>
3543 </wsdl:operation>
3544 <wsdl:operation name="RetrieveBiographicInformation">
3545     <soap:operation soapAction="RetrieveBiographicInformation" />
3546     <wsdl:input name="RetrieveBiographicInformationRequest">
3547         <soap:body use="literal" />
3548     </wsdl:input>
3549     <wsdl:output name="RetrieveBiographicInformationResponse">
3550         <soap:body use="literal" />
3551     </wsdl:output>
3552     <wsdl:fault name="BIASFault">
3553         <soap:fault name="BIASFault" use="literal" />
3554     </wsdl:fault>
3555 </wsdl:operation>
3556 <wsdl:operation name="RetrieveBiometricInformation">
3557     <soap:operation soapAction="RetrieveBiometricInformation" />
3558     <wsdl:input name="RetrieveBiometricInformationRequest">
3559         <soap:body use="literal" />
3560     </wsdl:input>
3561     <wsdl:output name="RetrieveBiometricInformationResponse">
3562         <soap:body use="literal" />
3563     </wsdl:output>
3564     <wsdl:fault name="BIASFault">
3565         <soap:fault name="BIASFault" use="literal" />
3566     </wsdl:fault>
3567 </wsdl:operation>
3568 <wsdl:operation name="SetBiographicData">
3569     <soap:operation soapAction="SetBiographicData" />
3570     <wsdl:input name="SetBiographicDataRequest">
3571         <soap:body use="literal" />
3572     </wsdl:input>
3573     <wsdl:output name="SetBiographicDataResponse">
3574         <soap:body use="literal" />
3575     </wsdl:output>
3576     <wsdl:fault name="BIASFault">
3577         <soap:fault name="BIASFault" use="literal" />
3578     </wsdl:fault>
3579 </wsdl:operation>
3580 <wsdl:operation name="SetBiometricData">
3581     <soap:operation soapAction="SetBiometricData" />
3582     <wsdl:input name="SetBiometricDataRequest">
3583         <soap:body use="literal" />
3584     </wsdl:input>
3585     <wsdl:output name="SetBiometricDataResponse">
3586         <soap:body use="literal" />
3587     </wsdl:output>
3588     <wsdl:fault name="BIASFault">

```

```

3589     <soap:fault name="BIASFault" use="literal" />
3590   </wsdl:fault>
3591 </wsdl:operation>
3592 <wsdl:operation name="TransformBiometricData">
3593   <soap:operation soapAction="TransformBiometricData" />
3594   <wsdl:input name="TransformBiometricDataRequest">
3595     <soap:body use="literal" />
3596   </wsdl:input>
3597   <wsdl:output name="TransformBiometricDataResponse">
3598     <soap:body use="literal" />
3599   </wsdl:output>
3600   <wsdl:fault name="BIASFault">
3601     <soap:fault name="BIASFault" use="literal" />
3602   </wsdl:fault>
3603 </wsdl:operation>
3604 <wsdl:operation name="UpdateBiographicData">
3605   <soap:operation soapAction="UpdateBiographicData" />
3606   <wsdl:input name="UpdateBiographicDataRequest">
3607     <soap:body use="literal" />
3608   </wsdl:input>
3609   <wsdl:output name="UpdateBiographicDataResponse">
3610     <soap:body use="literal" />
3611   </wsdl:output>
3612   <wsdl:fault name="BIASFault">
3613     <soap:fault name="BIASFault" use="literal" />
3614   </wsdl:fault>
3615 </wsdl:operation>
3616 <wsdl:operation name="UpdateBiometricData">
3617   <soap:operation soapAction="UpdateBiometricData" />
3618   <wsdl:input name="UpdateBiometricDataRequest">
3619     <soap:body use="literal" />
3620   </wsdl:input>
3621   <wsdl:output name="UpdateBiometricDataResponse">
3622     <soap:body use="literal" />
3623   </wsdl:output>
3624   <wsdl:fault name="BIASFault">
3625     <soap:fault name="BIASFault" use="literal" />
3626   </wsdl:fault>
3627 </wsdl:operation>
3628 <wsdl:operation name="VerifySubject">
3629   <soap:operation soapAction="VerifySubject" />
3630   <wsdl:input name="VerifySubjectRequest">
3631     <soap:body use="literal" />
3632   </wsdl:input>
3633   <wsdl:output name="VerifySubjectResponse">
3634     <soap:body use="literal" />
3635   </wsdl:output>
3636   <wsdl:fault name="BIASFault">
3637     <soap:fault name="BIASFault" use="literal" />
3638   </wsdl:fault>
3639 </wsdl:operation>
3640 <wsdl:operation name="Enroll">
3641   <soap:operation soapAction="Enroll" />

```

```

3642     <wsdl:input name="EnrollRequest">
3643         <soap:body use="literal" />
3644     </wsdl:input>
3645     <wsdl:output name="EnrollResponse">
3646         <soap:body use="literal" />
3647     </wsdl:output>
3648     <wsdl:fault name="BIASFault">
3649         <soap:fault name="BIASFault" use="literal" />
3650     </wsdl:fault>
3651 </wsdl:operation>
3652 <wsdl:operation name="GetEnrollResults">
3653     <soap:operation soapAction="GetEnrollResults" />
3654     <wsdl:input name="GetEnrollResultsRequest">
3655         <soap:body use="literal" />
3656     </wsdl:input>
3657     <wsdl:output name="GetEnrollResultsResponse">
3658         <soap:body use="literal" />
3659     </wsdl:output>
3660     <wsdl:fault name="BIASFault">
3661         <soap:fault name="BIASFault" use="literal" />
3662     </wsdl:fault>
3663 </wsdl:operation>
3664 <wsdl:operation name="GetIdentifyResults">
3665     <soap:operation soapAction="GetIdentifyResults" />
3666     <wsdl:input name="GetIdentifyResultsRequest">
3667         <soap:body use="literal" />
3668     </wsdl:input>
3669     <wsdl:output name="GetIdentifyResultsResponse">
3670         <soap:body use="literal" />
3671     </wsdl:output>
3672     <wsdl:fault name="BIASFault">
3673         <soap:fault name="BIASFault" use="literal" />
3674     </wsdl:fault>
3675 </wsdl:operation>
3676 <wsdl:operation name="GetVerifyResults">
3677     <soap:operation soapAction="GetVerifyResults" />
3678     <wsdl:input name="GetVerifyResultsRequest">
3679         <soap:body use="literal" />
3680     </wsdl:input>
3681     <wsdl:output name="GetVerifyResultsResponse">
3682         <soap:body use="literal" />
3683     </wsdl:output>
3684     <wsdl:fault name="BIASFault">
3685         <soap:fault name="BIASFault" use="literal" />
3686     </wsdl:fault>
3687 </wsdl:operation>
3688 <wsdl:operation name="Identify">
3689     <soap:operation soapAction="Identify" />
3690     <wsdl:input name="IdentifyRequest">
3691         <soap:body use="literal" />
3692     </wsdl:input>
3693     <wsdl:output name="IdentifyResponse">
3694         <soap:body use="literal" />

```

```

3695     </wsdl:output>
3696     <wsdl:fault name="BIASFault">
3697       <soap:fault name="BIASFault" use="literal" />
3698     </wsdl:fault>
3699   </wsdl:operation>
3700   <wsdl:operation name="RetrieveInformation">
3701     <soap:operation soapAction="RetrieveInformation" />
3702     <wsdl:input name="RetrieveInformationRequest">
3703       <soap:body use="literal" />
3704     </wsdl:input>
3705     <wsdl:output name="RetrieveInformationResponse">
3706       <soap:body use="literal" />
3707     </wsdl:output>
3708     <wsdl:fault name="BIASFault">
3709       <soap:fault name="BIASFault" use="literal" />
3710     </wsdl:fault>
3711   </wsdl:operation>
3712   <wsdl:operation name="Verify">
3713     <soap:operation soapAction="Verify" />
3714     <wsdl:input name="VerifyRequest">
3715       <soap:body use="literal" />
3716     </wsdl:input>
3717     <wsdl:output name="VerifyResponse">
3718       <soap:body use="literal" />
3719     </wsdl:output>
3720     <wsdl:fault name="BIASFault">
3721       <soap:fault name="BIASFault" use="literal" />
3722     </wsdl:fault>
3723   </wsdl:operation>
3724 </wsdl:binding>
3725 <wsdl:service name="BIAS_v1">
3726   <wsdl:port name="BIAS_v1HttpBindingPort" binding="tns:BIAS_v1HttpBinding">
3727     <soap:address location="http://your_URL_to_your_BIAS_implementation" />
3728   </wsdl:port>
3729 </wsdl:service>
3730 </wsdl:definitions>
3731

```

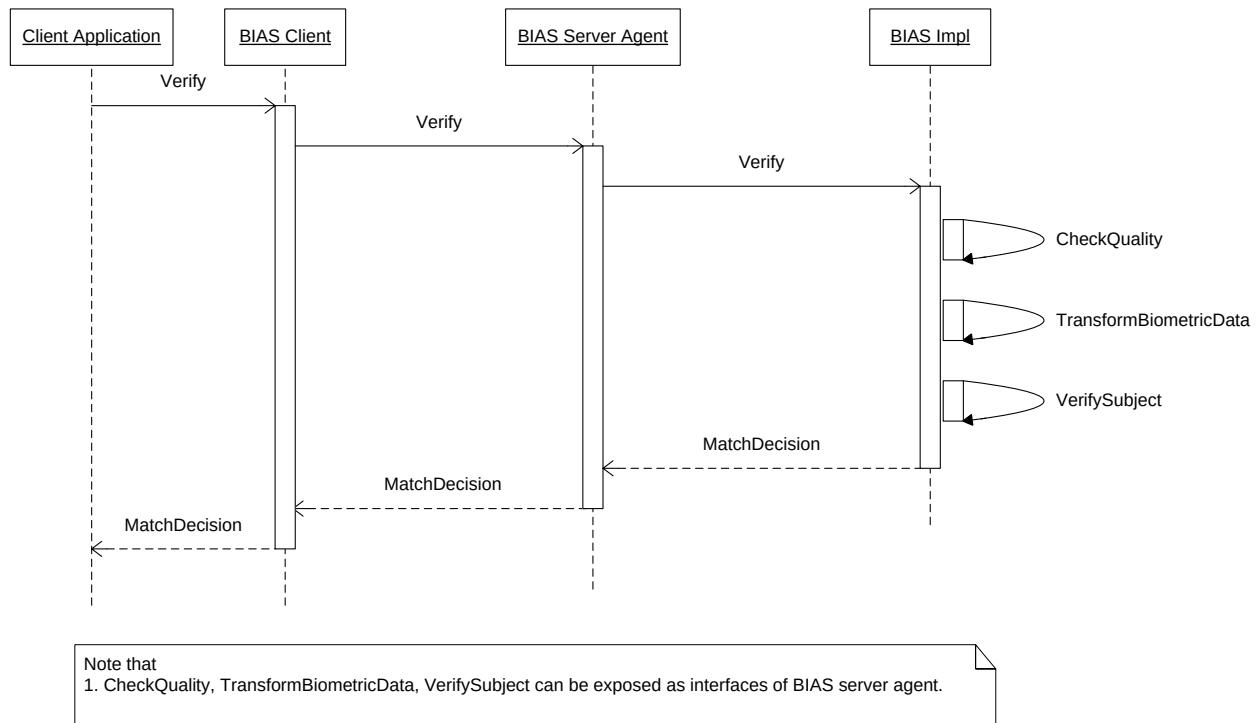
B. Use Cases (non-normative)

The intent of this annex is to provide operational sequence diagrams / flow charts that show how the higher level usage scenarios within [INCITS-BIAS] could be implemented using the BIAS SOAP profile. The following use cases are given:

- Verification (synchronous/aggregate)
- Verification (asynchronous/aggregate)
- Verification (primitive)
- Identification (primitive)
- Enrollment (aggregate)
- Enrollment (primitive)

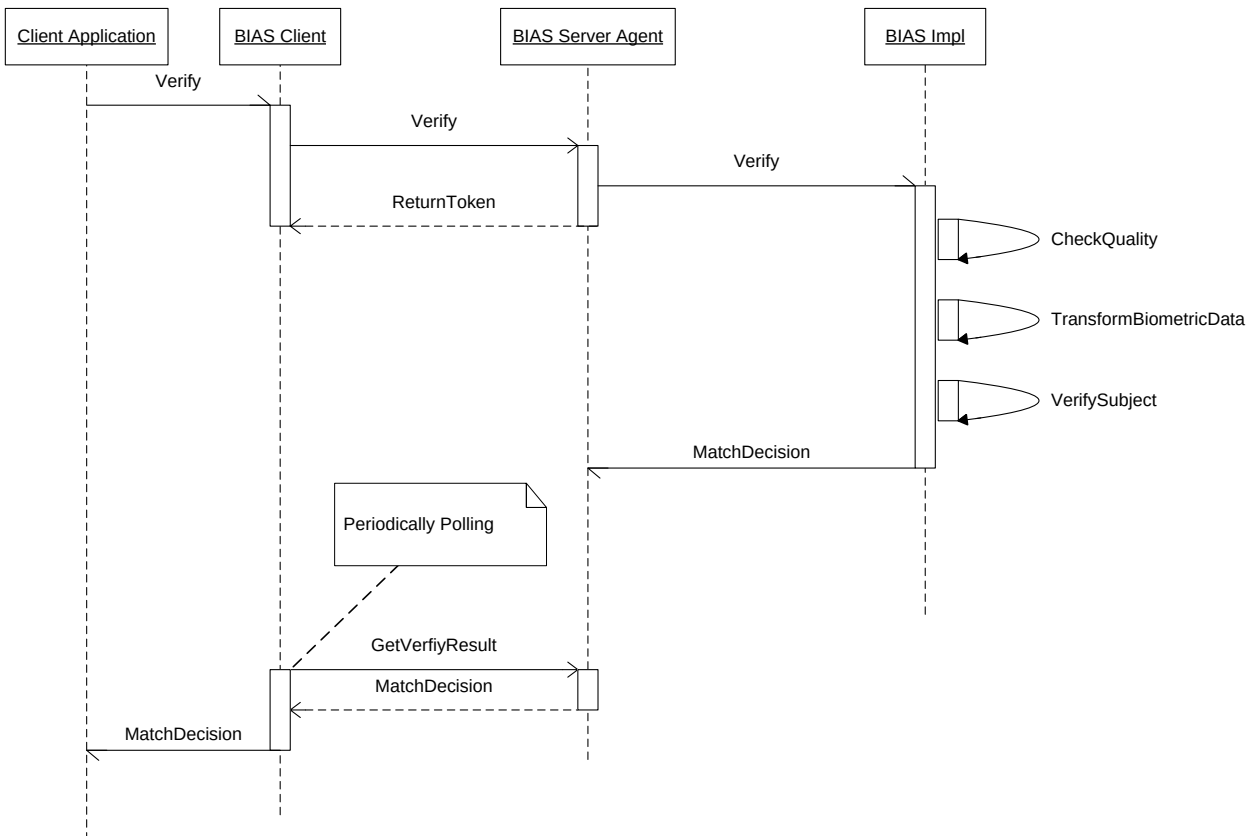
B.1 Verification Use Case

This use case uses the aggregate Verify operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.



B.2 Asynchronous Verification

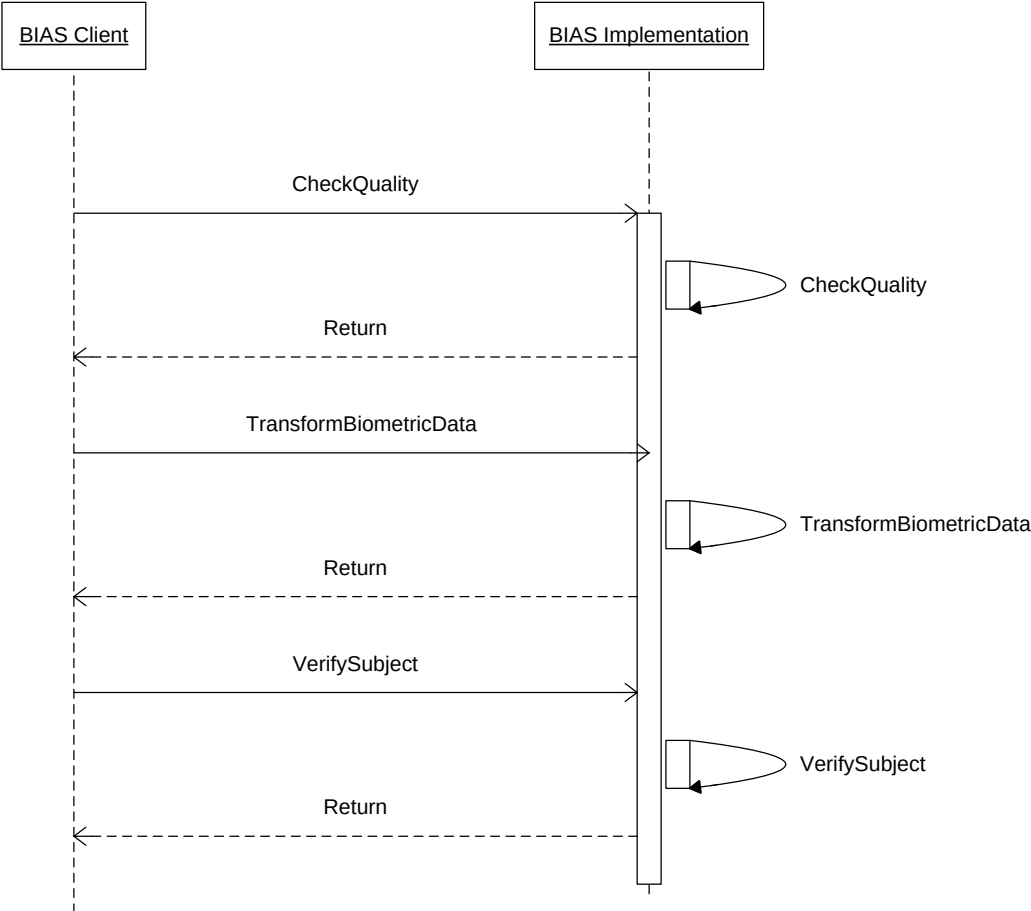
In this use case, the requester issues two requests – the BIAS Verify request to initiate the operation followed by a BIAS GetVerifyResult request to retrieve the results of that operation.



Note that
1. CheckQuality, TransformBiometricData, VerifySubject can be exposed as interfaces of BIAS server agent.

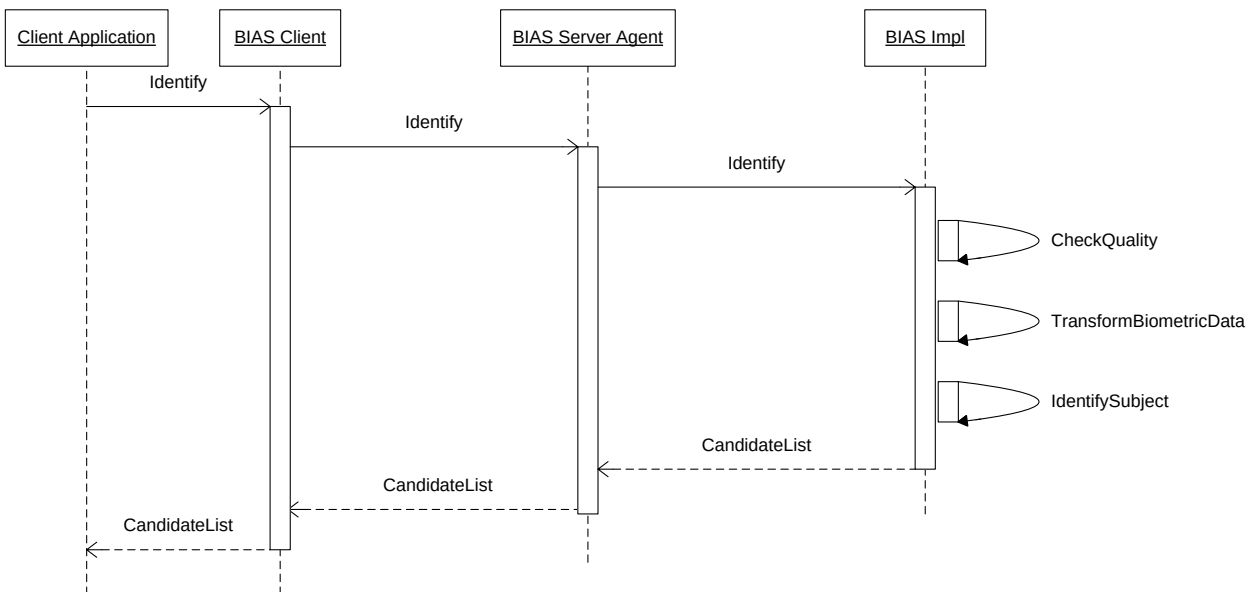
B.3 Primitive Verification

In this use case, the verification operation is performed as a series of requests using the BIAS primitive operations. In this case, the client rather than the service provider controls the workflow of the higher level operation.



B.4 Identification Use Case

This use case uses the aggregate Identify operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.



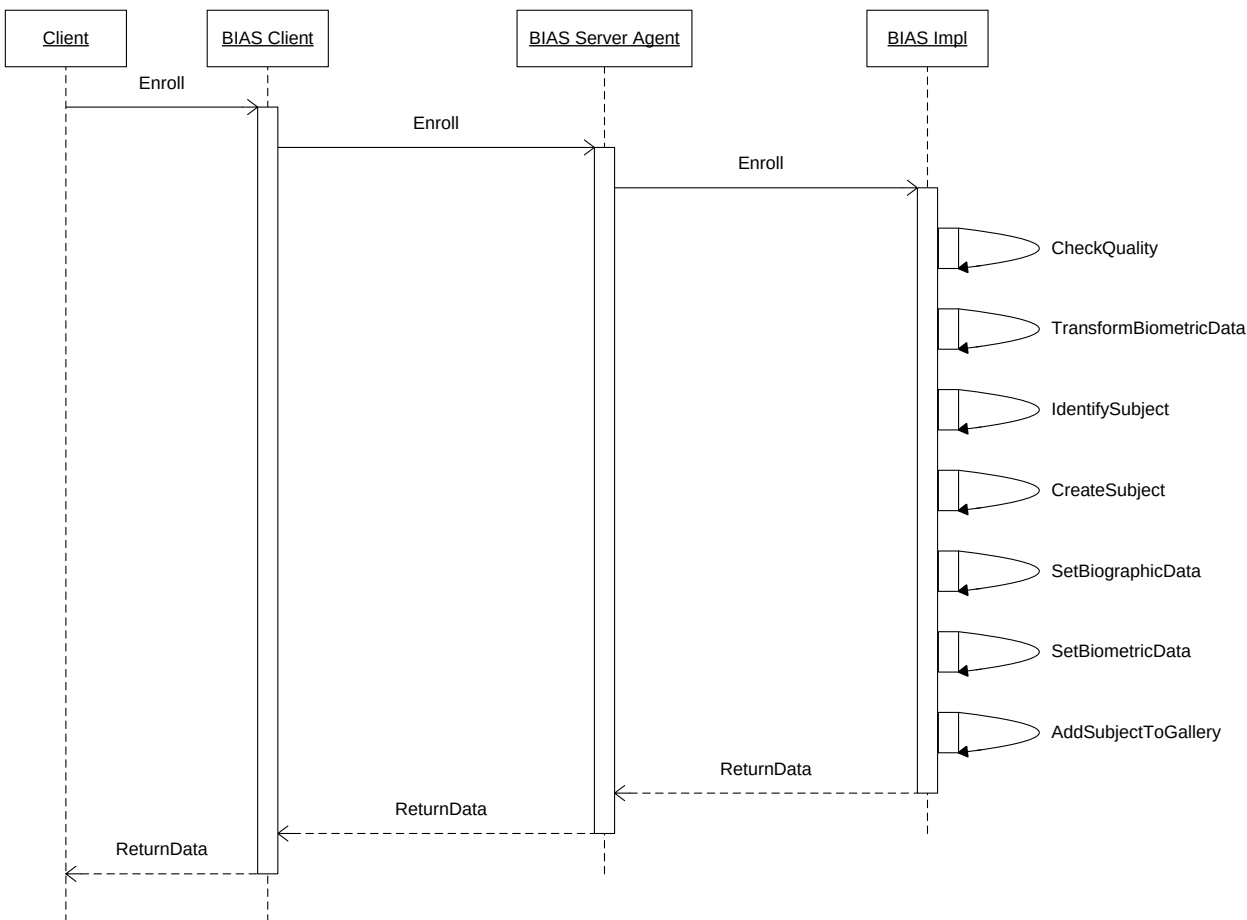
Note that

1. CheckQuality, TransformBiometricData, VerifySubject can be exposed as interfaces of BIAS server agent.

B.5 Biometric Enrollment

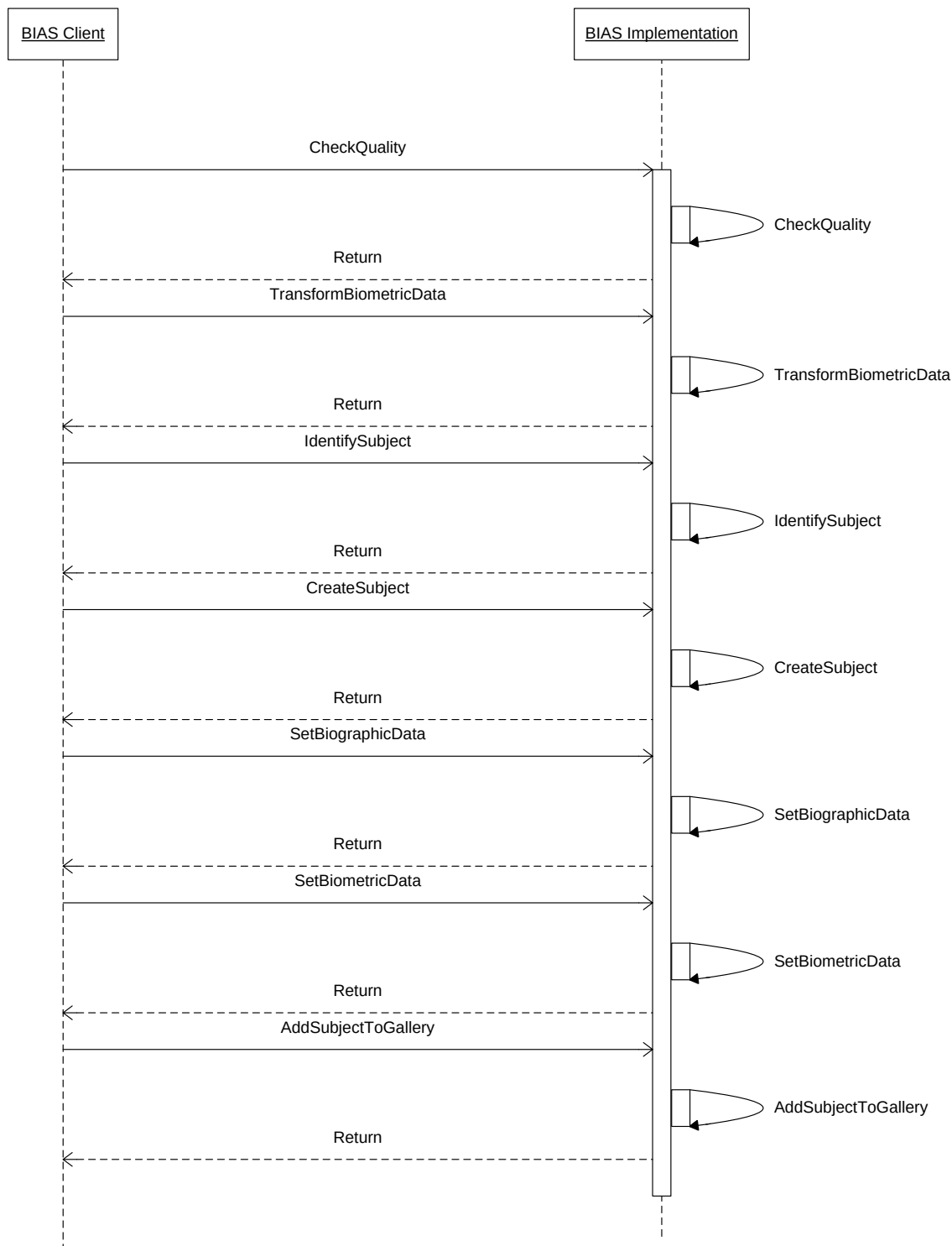
This use case uses the aggregate Enroll operation in which a single request results in some set of operations (in this case, a series of primitive BIAS operations) being performed by the BIAS service provider.

Here, if the result of the IdentifySubject is no matches found, then the subject is added to the gallery. If a match had been found then other logic may have been applied (e.g., return candidate list, add encounter for existing subject, etc.).



B.6 Primitive Enrollment

In this use case, the enrollment operation is performed as a series of requests using the BIAS primitive operations. In this case, the client rather than the service provider controls the workflow of the higher level operation.



C. Samples (non-normative)

C.1 Create Subject Request/Response Example

INCITS BIAS Specification

```
<interface name="CreateSubject">
  <parameter name="SubjectID" type="xs:string" direction="inout"
use="optional" />
  <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

OASIS BIAS Examples

Simple Create Subject Request:

```
POST /bias HTTP/1.1
Host: www.acme.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "CreateSubject"

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <tns:CreateSubjectRequest
                                xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/"
  />
  </soap:Body>
</soap:Envelope>
```

Create Subject Request with SubjectID Parameter:

```
POST /bias HTTP/1.1
Host: www.acme.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "CreateSubject"

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <tns:CreateSubjectRequest
xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/"
    <tns:Identity>
      <tns:SubjectID>123456789</tns:SubjectID>
    </tns:Identity>
    </tns:CreateSubjectRequest>
  </soap:Body>
```

3827 </soap:Envelope>

3828 **Create Subject Request with Optional OASIS BIAS Content:**

3829 POST /bias HTTP/1.1

3830 Host: www.acme.com

3831 Content-Type: application/soap+xml; charset="utf-8"

3832 Content-Length: nnnn

3833 SOAPAction: "CreateSubject"

3834 <?xml version="1.0"?>

3835 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

3836 <soap:Body>

3837 <tns:CreateSubjectRequest

3838 xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">

3839 <tns:GenericRequestParameters>

3840 <tns:Application>BIAS Application</tns:Application>

3841 <tns:ApplicationUser>BIAS User</tns:ApplicationUser>

3842 </tns:GenericRequestParameters>

3843 <tns:Identity>

3844 <tns:SubjectID>123456789</tns:SubjectID>

3845 </tns:Identity>

3846 </tns:CreateSubjectRequest>

3847 </soap:Body>

3848 </soap:Envelope>

3849 **Simple Create Subject Response:**

3850 HTTP/1.1 200 OK

3851 Content-Type: application/soap+xml; charset="utf-8"

3852 Content-Length: nnnn

3853 <?xml version="1.0"?>

3854 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

3855 <soap:Body>

3856 <tns:CreateSubjectResponse

3857 xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">

3858 <tns:ResponseStatus>

3859 <tns:Return>0</tns:Return>

3860 </tns:ResponseStatus>

3861 <tns:Identity>

3862 <tns:SubjectID>123456789</tns:SubjectID>

3863 </tns:Identity>

3864 </tns:CreateSubjectResponse>

3865 </soap:Body>

3866 </soap:Envelope>

3867 **Create Subject Response with Optional OASIS BIAS Content:**

```

3868 HTTP/1.1 200 OK
3869 Content-Type: application/soap+xml; charset="utf-8"
3870 Content-Length: nnnn
3871 <?xml version="1.0"?>
3872 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3873     <soap:Body>
3874         <tns:CreateSubjectResponse
3875 xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3876             <tns:ResponseStatus>
3877                 <tns:Return>0</tns:Return>
3878                 <tns:Message>Subject ID 123456789 successfully
3879 created.</tns:Message>
3880             </tns:ResponseStatus>
3881             <tns:Identity>
3882                 <tns:SubjectID>123456789</tns:SubjectID>
3883             </tns:Identity>
3884         </tns:CreateSubjectResponse>
3885     </soap:Body>
3886 </soap:Envelope>
3887

```

C.2 Set Biographic Data Request/Response Example

INCITS BIAS Specification

```

3890 <interface name="SetBiographicData">
3891     <parameter name="SubjectID" type="xs:string" direction="in" />
3892     <parameter name="IdentityModel" type="IdentityModelType" direction="in"
3893 />
3894     <parameter name="EncounterID" type="xs:string" direction="inout"
3895 use="optional" />
3896     <parameter name="BiographicData" type="BiographicDataType" direction="in"
3897 />
3898     <parameter name="Return" type="xs:unsignedLong" direction="out" />
3899 </interface>

```

OASIS BIAS Examples

Set Biographic Data Request:

```

3902 POST /bias HTTP/1.1
3903 Host: www.acme.com
3904 Content-Type: application/soap+xml; charset="utf-8"
3905 Content-Length: nnnn
3906 SOAPAction: "SetBiographicData"
3907 <?xml version="1.0"?>
3908 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3909     <soap:Body>

```



```

3910         <tns:SetBiographicDataRequest
3911             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3912             <tns:Identity>
3913                 <tns:SubjectID>123456789</tns:SubjectID>
3914                 <tns:BiographicData>
3915                     <tns:BiographicDataItem>
3916                         <tns:Name>Last</tns:Name>
3917                         <tns:Type>string</tns:Type>
3918                         <tns:Value>Doe</tns:Value>
3919                     </tns:BiographicDataItem>
3920                 </tns:BiographicData>
3921             </tns:Identity>
3922             <tns:IdentityModel>person</tns:IdentityModel>
3923         </tns:SetBiographicDataRequest>
3924     </soap:Body>
3925 </soap:Envelope>

```

3926 **Set Biographic Data Response:**

```

3927 HTTP/1.1 200 OK
3928 Content-Type: application/soap+xml; charset="utf-8"
3929 Content-Length: nnnn
3930 <?xml version="1.0"?>
3931 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3932     <soap:Body>
3933         <tns:SetBiographicDataResponse
3934             xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3935             <tns:ResponseStatus>
3936                 <tns:Return>0</tns:Return>
3937             </tns:ResponseStatus>
3938         </tns:SetBiographicDataResponse>
3939     </soap:Body>
3940 </soap:Envelope>
3941

```

3942 **C.3 Set Biometric Data Request/Response Example**

3943 *INCITS BIAS Specification*

```

3944 <interface name="SetBiometricData">
3945     <parameter name="SubjectID" type="xs:string" direction="in" />
3946     <parameter name="IdentityModel" type="IdentityModelType" direction="in"
3947 />
3948     <parameter name="EncounterID" type="xs:string" direction="inout"
3949 use="optional" />
3950     <parameter name="BIRList" type="CBEFF_BIR_ListType" direction="in" />

```

```

3951         <parameter name="Return" type="xs:unsignedLong" direction="out" />
3952     </interface>

3953     OASIS BIAS Examples

3954     Set Biometric Data Request:

3955     POST /bias HTTP/1.1
3956     Host: www.acme.com
3957     Content-Type: application/soap+xml; charset="utf-8"
3958     Content-Length: nnnn
3959     SOAPAction: "SetBiometricData"

3960     <?xml version="1.0"?>
3961     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3962         <soap:Body>
3963             <tns:SetBiometricDataRequest
3964                 xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3965                 <tns:Identity>
3966                     <tns:SubjectID>123456789</tns:SubjectID>
3967                     <tns:BiometricData>
3968                         <tns:BIRList>
3969                             <tns:BIR>biometric data</tns:BIR>
3970                         </tns:BIRList>
3971                     </tns:BiometricData>
3972                 </tns:Identity>
3973                 <tns:IdentityModel>person</tns:IdentityModel>
3974             </tns:SetBiometricDataRequest>
3975         </soap:Body>
3976     </soap:Envelope>

3977     Set Biometric Data Response:

3978     HTTP/1.1 200 OK
3979     Content-Type: application/soap+xml; charset="utf-8"
3980     Content-Length: nnnn

3981     <?xml version="1.0"?>
3982     <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3983         <soap:Body>
3984             <tns:SetBiometricDataResponse
3985                 xmlns:tns="http://docs.oasis-open.org/bias/bias-1.0/">
3986                 <tns:ResponseStatus>
3987                     <tns:Return>0</tns:Return>
3988                 </tns:ResponseStatus>
3989             </tns:SetBiometricDataResponse>
3990         </soap:Body>
3991     </soap:Envelope>

```

D. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Name	Affiliation
Mr. Young Bang	Booz Allen Hamilton
Mr. Murty Gurajada *	Raining Data Corporation
Mr. Dale Hapeman	US Department of Defense
Dr. Charles Li	Raytheon
Mr. John Mayer-Splain	US Department of Homeland Security
Dr. Ross Michaels	NIST
Mr. Ash Parikh *	Raining Data Corporation
Mr. Matthew Swayze	Daon
Mr. Guy Swope	Raytheon
Mrs. Catherine Tilton	Daon
Mr. Alessandro Triglia	OSS Nokalva
Mr. Brad Wing *	US Department of Homeland Security
Mr. Michael Wittman	Raytheon
Mr. Gregory Zektser	Booz Allen Hamilton

* Though no longer members of the BIAS TC, these individuals contributed in the early stages of the development of this standard.

In addition, the inputs from the INCITS technical committee M1 are also gratefully appreciated.

4001

E. Revision History

4002

Revision	Date	Editor	Changes Made
0.01	2008-05-23	TBD	Initial draft
0.02	2008-07-23	TBD	Inserted data dictionary Added normative references Updated sec 3 & 5 + Annex B
0.03	2008-08-19	TBD	WSDL updated
0.04	2008-09-11	TBD	Updated references Added security requirements Corrected Fig. 3
0.05	2008-09-29	TBD	SSL/TLS requirement clarified Reordered material in 5.3 & App C/D Updated references 2 new use cases added (App C) Updated examples in App D
0.06	2008-11-17	TBD	Added BIAS operation name methods (new 5.3 + 4.2.27 & App B)
0.06a	2008-11-20	TBD	Updated references
0.07	2008-11-27	TBD	Revised fault structures and error handling

F. Open Issues (temporary annex)

This annex is a temporary holding area for issues yet to be fully resolved.

- ◆ Document conformance requirements (Clause 2 & Annex A).
- ◆ Determine if our approach on fault and error handling is consistent with best practices.
- ◆ Determine if additional information is needed related to how to host, publish, and discover BIAS services.
- ◆ Determine if any additional requirements are needed with respect to the use of Signed XML or WSS (more specific than contained in 1st para of 5.4)?
- ◆ Consider looking at the SAML Security Considerations document for appropriate content/references.
- ◆ Determine if the operation name needs to be included in the Response as well as the Request message.
- ◆ Investigate how SOAP handles correlation of responses to a requests. Is there anything specific we need to incorporate to facilitate this? What are best practices in this regard?
- ◆ Add sample code in Annex D for each use case in Annex C.
- ◆ Do we need to include a non-normative reference to BioAPI (ISO/IEC 19784) as it is shown in Figure 2?
- ◆ Do we need to add entries for all acronyms found in the document, including common ones (e.g., URI, UDDI, SSL, TLS, HTTP, HTTPS, VPN, WSS, XML)?