INTERNATIONAL STANDARD

ISO/IEC 30108-1

First edition
2015-11-01

Corrected version
2016-04-15

Information technology — Biometric Identity Assurance Services —

Part 1: BIAS services

Technologies de l'information — Service d'assurance de l'identité biométrique (BIAS) —

Partie 1: Services BIAS

Reference number
ISO/IEC 30108-1:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

ISO/IEC 30108 consists of the following parts, under the general title *Information technology — Biometric Identity Assurance Services*:

— *Part 1: BIAS Services*

This corrected version of ISO/IEC 30108-1:2015 incorporates the following corrections plus other minor editorial modifications.

Table A.1: The following rows were missing and have been added to the end of the table:

Get Verify Results		C*			C*		C*
Identify		X					X
Retrieve Data		X			X		
Update		X			X		X
Verify		X			X		X
Capability Information Items							
AggregateInputDataOptional		X			X		X
AggregateInputDataRequired		X			X		X
AggregateProcessingOption		X			X		X
AggregateReturnData		X			X		X
AggregateServiceDescription		X			X		X
BiographicDataSet	X	X	X	X	X		
CBEFFPatronFormat	X	X	X	X	X	X	X

ClassificationAlgorithmType	X						
ConformanceClass	X	X	X	X	X	X	X
Gallery	X	X	X				
IdentityModel	X	X	X	X	X		
ComparisonAlgorithm	X	X	X	X	X	X	X
ComparisonScore	X	X	X	X	X	X	X
QualityAlgorithm	X						
SupportedBiometric	X	X	X	X	X	X	X
TransformOperation	X						

Introduction

This part of ISO/IEC 30108 defines the architecture, operations, data elements, and basic requirements for biometric identity assurance services – a framework for the implementation of generic, biometric-based identity services within a services-oriented environment. An identity in the context of BIAS comprises a subject, biographic data, and biometric data. Other parts are intended to define specific BIAS implementations (or bindings) within specific environments, for example, SOAP web services.

BIAS services are generic in nature, being modality neutral, and not targeted at any particular business application. These services include those related to identity data management, transformation, and biometric comparison. Services are invoked by a BIAS requester and implemented by a BIAS service provider (responder). It does not prescribe the architecture or business logic of either the requester or service provider.

Two categories of identity services are defined – primitive and aggregate. Primitive services are more atomic and well-defined, whereas the aggregate services tend to be higher level and enable more flexibility on the part of the BIAS service provider.

Two identity models are also defined – person-centric and encounter-based. Person-centric systems maintain a single up-to-date record (set of data) for a given subject, whereas an encounter-based system retains data related to each interaction the subject has with the system.

This part of ISO/IEC 30108 represents a version of BIAS subsequent to that previously standardized by INCITS and OASIS, therefore, it is denoted as version 2.0.

Information technology — Biometric Identity Assurance Services —

Part 1: BIAS services

1 Scope

This part of ISO/IEC 30108 defines biometric services used for identity assurance that are invoked over a services-based framework. It provides a generic set of biometric and identity-related functions and associated data definitions to allow remote access to biometric services.

The binding of these services to specific frameworks is not included in this part of ISO/IEC 30108, but will be the subject of subsequent parts.

Although focused on biometrics, this part of ISO/IEC 30108 will necessarily include support for other related identity assurance mechanisms such as biographic and document capabilities. BIAS is intended to be compatible with and used in conjunction with other biometric standards as described in Clause 3.

Specification of biometric functionality is limited to remote (backend) services. Services between a client-side application and biometric capture devices are not within the scope of this part of ISO/IEC 30108.

Integration of biometric services as part of an authentication service or protocol is not within the scope of this part of ISO/IEC 30108.

2 Conformance

Annex A specifies the conformance requirements for systems/components claiming conformance to this part of ISO/IEC 30108.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19785-1, *Information technology — Common Biometric Exchange Formats Framework — Part 1: Data element specification*

ISO/IEC 19785-3, *Information technology — Common Biometric Exchange Formats Framework — Part 3: Patron format specifications*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
biometric sample
analogue or digital representation of biometric characteristics prior to biometric feature extraction

Note 1 to entry: As an example, a record containing the image of a finger is a biometric sample.

4.2
claim to identity
biometric claim
assertion that an individual is or is not the bodily source of a specified or unspecified biometric reference in an identity assurance system

4.3
encounter
event in which the BIAS requester interacts with a *subject* (4.11) resulting in data being collected during or about the encounter

Note 1 to entry: The event may involve collection of biographic, biometric, document, and/or contextual data during an enrolment or recognition interaction.

4.4
encounter-centric system
system that supports encounter processing maintaining a one-to-many relationship between *subjects* (4.11) and *encounters* (4.3) and which does not necessarily contain a single, unique set of information for each subject

4.5
gallery
group of *subjects* (4.11) related by a common purpose, designation, or status

EXAMPLE A watch list or a set of subjects entitled to a certain benefit.

4.6
identification
biometric identification
process of searching against a biometric enrolment database to find and return the biometric reference identifier(s) attributable to a single individual

4.7
identity assurance
process of establishing, determining, and/or confirming a subject identity

4.8
merge
combination of biometric data during the process of updating an enrolment record

Note 1 to entry: The "merge" operation is implementation specific, however, it may include either adding a new sample to a multi-sample record or performing some level of biometric fusion, for example, sample or feature level fusion.

4.9
merge
combination of two or more subject records into a single subject record

4.10
person-centric model
identity model in which a single master record is maintained on a *subject* (4.11) which is updated over time when additional, newer, or better biographic, biometric, and/or document information becomes available and which does not maintain separate historical data records for each system encounter with the subject

4.11
subject
person who is known to an identity assurance system

Note 1 to entry: The person may also be a biometric capture subject or biometric data subject, but this is not the case in all situations.

4.12
verification
biometric verification
process of confirming a *biometric claim* (4.2) through biometric comparison

Note 1 to entry: Verification is usually performed through a one-to-one comparison in which a *biometric sample* (4.1) from one individual (probe) is compared to a biometric reference(s) from one individual to produce a comparison decision (match/no-match) and optionally, a comparison score.

5 Symbols and abbreviated terms

AFIS	Automated Fingerprint Identification System
BIAS	Biometric Identity Assurance Services
BIR	Biometric Information Record
CBEFF	Common Biometric Exchange Formats Framework
ESB	Enterprise Service Bus
ID	Identity/Identification/Identifier
OASIS	Organization for the Advancement of Structured Information Standards
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol

6 System context

This clause provides an overview of Service-Oriented Architectures, the BIAS architecture, and BIAS implementation considerations.

6.1 Service-oriented architectures

Service-Oriented Architectures are software architectures in which reusable services are deployed onto application servers and then consumed by clients in different applications or business processes. They are intended to decouple the implementation of a software service from the interface that calls that service. This allows clients of a service to rely on a consistent interface regardless of the implementation technology of the service [JDJ] (see Annex C).

Biometric services are one of the types of services that can be provided over such a remote interface in a distributed information system across a collection of networks. This can occur in a 2-tier, 3-tier, or N-tier environment. A diagram of a simple N-tier architecture is shown in Figure 1.

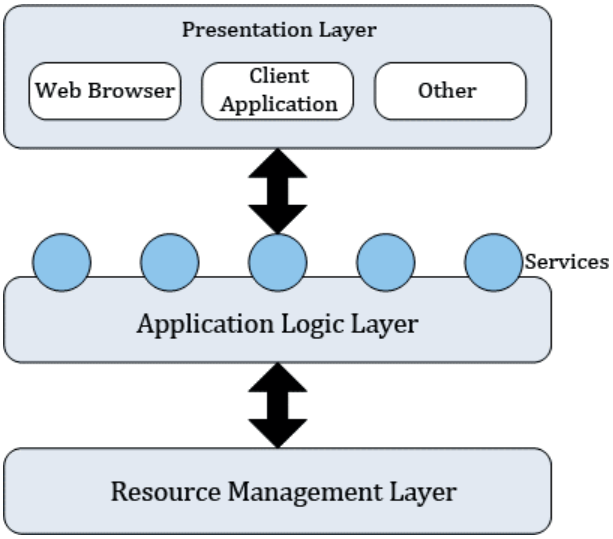


Figure 1 — Simple N-Tier architecture

In this diagram, BIAS services are defined between the application logic layer and the resource management layer.

Examples of biometric resources that are of interest may include one or more of the following:

— A fingerprint verification server;

— A 1:N iris search engine;

— A facial biometric watch list;

— A criminal or civil automated fingerprint identification system (AFIS);

— A name-based biographic identity database;

— An archive of biometric identifiers;

— A population of subjects.

It is desired that a generic set of services be defined that allows clients to remotely access and manage these capabilities. To the extent possible, domain specific implementations are to be avoided.

NOTE This part of ISO/IEC 30108 is intended to support a wide variety of application domains which may include government (e.g. background checking, border management, and criminal justice), enterprise (e.g. logical access control), and commercial biometric identity management implementations (e.g. employee databases).

Services are well defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services. Services can be easily assembled to form a collection of autonomous and loosely-coupled business processes.

It is not the intention that specific business logic be instantiated within the service definitions – this logic is more appropriate within the application logic layer – either in the higher level system initiating the series of requests, or within the middleware [e.g. an enterprise service bus (ESB), workflow manager, or biometric middleware] as appropriate. To do so would of necessity make the interface less generic, modular, and flexible and require that the interface be updated each time the logic changed, defeating one of the primary purposes of the services architecture.

The services to be defined are not targeted at a particular SOA implementation or framework. Instead, they are defined in such a manner as to be able to be utilised within any such architecture. This is accomplished by separately defining (in another standard) the bindings to that

architecture/implementation. For example, Web services bindings are defined in the OASIS BIAS Messaging Protocol.

6.2 BIAS architecture

The BIAS architecture consists of the following components:

— BIAS services (interface definition);

— BIAS data (schema definition);

— BIAS bindings (defined outside this standard).

The BIAS services expose a common set of operations to external requesters of these operations. These requesters may be an external system, a Web application, or an intermediary. The BIAS services themselves are platform and language independent. The BIAS services may be implemented with differing technologies on multiple platforms. For example, OASIS is defining Web services bindings for the BIAS services.

Figure 2 depicts the BIAS services within an application environment. BIAS services provide basic biometric functionality as modular and independent operations which can be assembled in many different ways to perform and/or support a variety of business processes. BIAS services can be publicly exposed directly and/or utilised indirectly in support of a service-provider's own public services.

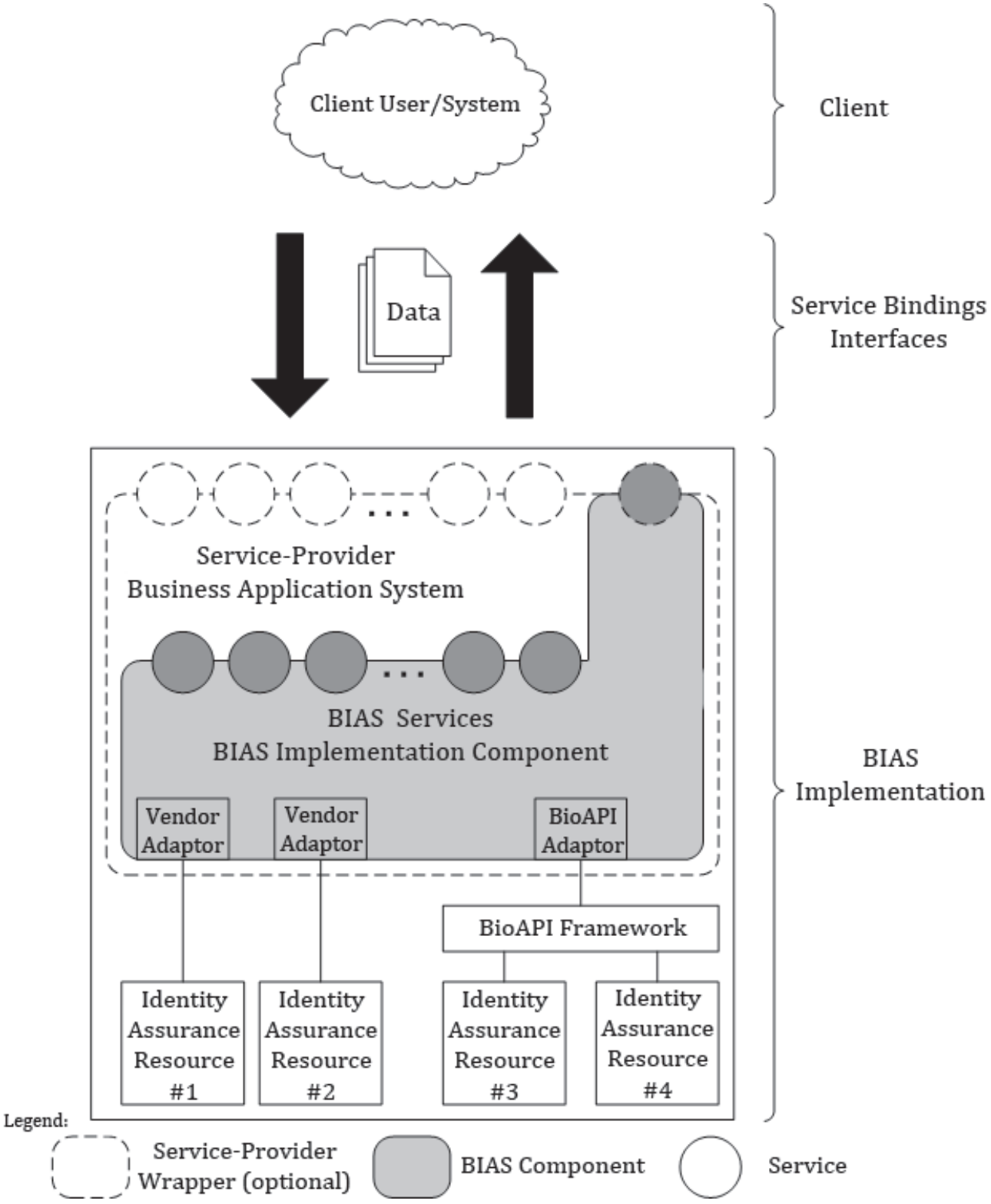


Figure 2 — BIAS application environment

6.3 Identity models

Some identity systems are person-centric and others are encounter-centric. That is, some base transactions on a unique identifier associated with an individual human being while others track

"biometric encounters" which may or may not be linked through such an identifier. Figure 3 provides context to further explain these concepts.

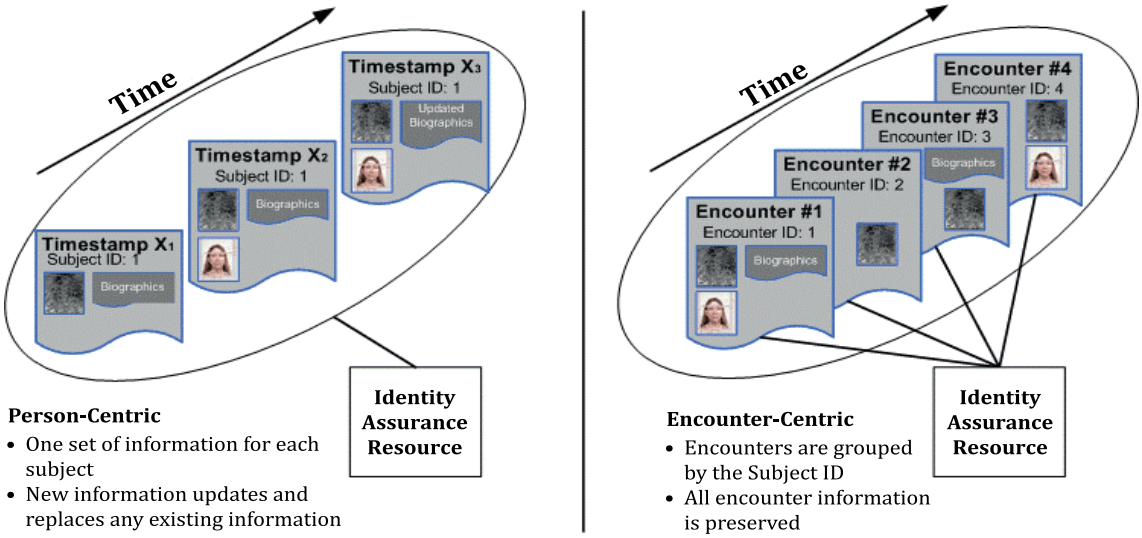


Figure 3 — Person-centric and encounter-centric views

In a person-centric model, as new data is received for a given subject, it is either added (if it does not already exist) or replaces previous data (if it already exists). For example, referring to Figure 3, if the initial enrolment contains biographic data and a set of fingerprints these are stored. If subsequently, a photo is received, it is added to the person-centric record. If later new biographic data is received (e.g. new address), it replaces the originally stored data. In this way, the "master" subject record is continuously updated to contain the most accurate, current information with (in general, but not exclusively) no need to retain historical data. This model is used, for example, in access control type systems.

An encounter-centric model, in comparison, retains all data received for every interaction with the subject. Initially, the subject record is created and populated with enrolment data (for example, fingerprint, facial photo, and biographic data as shown in Figure 3). Subsequently, if new fingerprint data is captured for that subject (during an interaction event, or encounter), a new encounter is created containing all data obtained during that encounter. The system now has two encounters for that subject. Later, in a third encounter, fingerprint and biographic data is captured and stored, in addition to the data previously stored in encounters one and two. Encounter IDs, unique to a subject, are assigned to each encounter. This model is used, for example, in case management type systems.

EXAMPLE In a border management system, a person is enroled in the system the first time they enter the country. A subject is created and biometric and biographic data are set, following any required identification operation(s). This represents the first encounter. Subsequently, when the person crosses the border in the future, their passport number is used as a claim of identity, a verification of that identity is performed, as well as any identification operations. Biometric and biographic data collected during this interaction is retained as another separately identified encounter (see Annex D for additional use cases).

Encounters can be classified as either enrolment or recognition encounters. Figure 4 depicts an example in which both biographic and biometric data are submitted during different types of encounters. Diagrams for additional scenarios are found in Annex D.

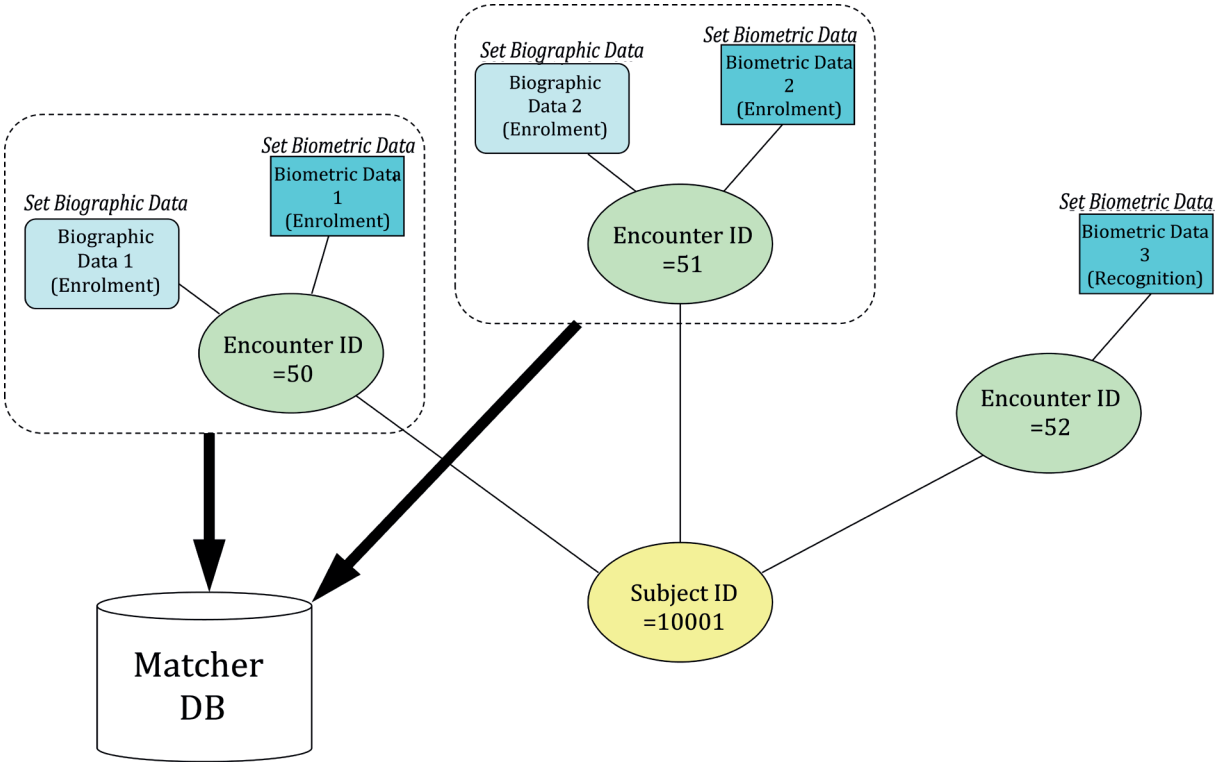


Figure 4 — Example of encounter types

NOTE Encounter type may be set using the ***Create Encounter*** operation. In addition, the Purpose field within the CBEFF header may be set to indicate this (see 8.2 for more information on biometric data).

6.4 Identity databases

Biometric systems frequently include both a master database as well as a comparison engine database. The former is used to store all identity information (including encounter information). The latter, however, usually contains a single set of biometric reference data, even within an encounter-based system and minimal, if any, biographic data. Systems use various schemes to determine which data to include within the biometric engines and when that data is updated. Figure 5 provides an example of how these databases may exist within a BIAS implementation. This is not always the case; however, as in some systems the master and comparison engine databases may be one and the same.

Within this part of ISO/IEC 30108, all services (with the exception of comparison services, such as ***IdentifySubject***) operate upon the master database unless otherwise stated, with any associated updates to the comparison engine databases (when separate) left to the policies of the BIAS service provider.

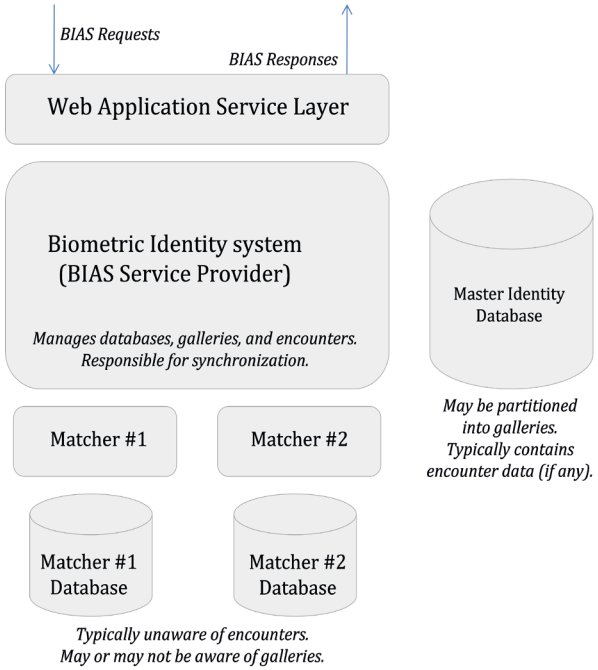


Figure 5 — Example master/engine database implementation

EXAMPLE In the previous border management example, fingerprint, face, and iris biometric data is collected on the first encounter. All information is stored in the master database. Additionally, selected fingerprint and iris data are processed into templates and stored within the appropriate comparison engine database. During the second encounter, only the data required for verification and identification is collected, and this data is stored in the master database. Templates are generated and used to perform the comparison operations. Based on service provider policy, following successful verification, it is found that the left index and ring fingerprints are of better quality than those collected during the first encounter and the engine database is updated with these fingerprints (replacing the original templates).

6.5 BIAS implementation considerations (informative)

Biometric services and the applications which use them, particularly in an identity assurance context, have unique characteristics which are summarised below:

— Some services can be performed very quickly while others (such as a 1:N identification within a large population) can take considerable time (on the order of hours) to complete. Therefore, the interface should support both synchronous and asynchronous operations;

— Biometric operations may be singular or multi-biometric.

Biometric data is in nearly all cases considered personal information and thus privacy protection is always a consideration.

— Before a biometric, biographic and/or document data transaction occurs between two different entities, the terms and conditions of the use of the data should be negotiated and made transparent. The following questions may be addressed:

  — Who will be the recipient of the data to be shared?

  — For what purpose(s) can the recipient use this data?

  — Who/what authorises this data to be shared with the recipient for this purpose?

  — How long may the recipient retain the data?

  — How must data be destroyed at the end of a retention period?

— May the recipient share this data with other entities? If so, with whom? For what purpose(s)? How long may the third party retain the data?

— A recipient would later have to understand what they are accepting and the terms and conditions of the agreement.

NOTE Implementers may consider implementation of a Biometric Policy (BP) and/or Biometric Practice Statement (BPS0 as defined in ISO 19092. A BP is a set of rules that specify the applicability of biometric information for use by a class of applications or community that share common security and privacy requirements. A BP may be used to control access to biometric information by an entity acting in some role or for some purpose (e.g. information sharing). A BPS describes the security and privacy practices that an organization follows during the biometric information life cycle, including business, legal, regulatory, and technical considerations for information sharing and exchange.

Security is important for any kind of SOA. As in the case of privacy protection, before a biometric, biographic and/or document data transaction occurs between two different entities, the security characteristics provided by the BIAS implementation should be known.

For the purposes of data integrity and quality assurance, a capability for creation of a chain of custody should be created to track events, changes, and transfers of data. For example, the deletion of data should be tracked and logged.

NOTE Organization management should require that formal mechanisms be used to document and report biometric system events, including system malfunctions and other security incidents, and additions, modifications, deletions, and transfers of data. Incident management results and metrics gathered from biometric systems event journals should be used in a periodic review process that causes security controls to be re-evaluated over time. An on-going, systematic approach to reviews should inform the continuous improvement of the biometric information management system over time.

Methods for service level monitoring and compliance with agreed-upon service level agreements may be critical to requesters and implementers. Terms and conditions for these capabilities may need to be negotiated.

7 Biometric Identity Assurance Services

This clause defines two categories of BIAS services, primitive and aggregate. Primitive services are lower-level operations that are used to request a specific capability. Aggregate services operate at a higher-level, performing a sequence of primitive operations in a single request. (An example of such a sequence would be a negative search where a 1:N identification which results in no matches being found is immediately followed by the addition of the biometric sample into that search population.) Implementers are not restricted to one or the other but may provide (and requesters use) a mixture of both primitive and aggregate types.

Aggregate services do not have to utilize primitive services.

7.1 BIAS interface XML schema

A goal for this BIAS Standard is to be as language and protocol independent as possible. To that end, the services are specified using a simple XML schema as defined below.

```
<?xml version = "1.0" encoding="utf-8" ?>
<xs:schema id="BIAS_Interface" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:complexType name="InterfaceType">
       <xs:sequence>
          <xs:element name="parameter" type="ParameterType"
             minOccurs="0" maxOccurs="unbounded" />
       </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required" />
   </xs:complexType>
   <xs:complexType name="ParameterType">
      <xs:attribute name="name" type="xs:string" use="required" />
      <xs:attribute name="type" type="xs:string" use="required" />
      <xs:attribute name="direction" type="DirectionType" use="required" />
```



```
      <xs:attribute name="use" type="UseType" use="optional"
            default="required" />
   </xs:complexType>
   <xs:simpleType name="DirectionType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="in" />
         <xs:enumeration value="out" />
         <xs:enumeration value="inout" />
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="UseType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="required" />
         <xs:enumeration value="optional" />
         <xs:enumeration value="conditional" />
      </xs:restriction>
   </xs:simpleType>
   <xs:element name="interface" type="InterfaceType"></xs:element>
</xs:schema>
```

Each service is identified by an *<interface>* tag and must include a *name* attribute. Service parameters are identified by a *<parameter>* tag and must include a *name*, *type*, and *direction* attribute. The *direction* attribute specifies whether the parameter is an input parameter (*in*), an output parameter (*out*), or an input/output parameter (*inout*). Parameters may also include a *use* attribute to indicate if the parameter is required, optional, or conditional. If the parameter is conditional, the service description must identify the conditions.

7.2 Primitive services

BIAS specifies the following set of primitive services.

7.2.1 Add Subject To Gallery

```
<interface name="AddSubjectToGallery">
   <parameter name="GalleryID" type="xs:string" direction="in" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="IdentityClaim"
         type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.1.1 Description

The ***Add Subject To Gallery*** service shall register 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 shall be unique across the gallery. If no claim to identity is specified, the subject ID (assigned with the ***Create Subject*** service) shall 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 shall be specified.

NOTE In the BIAS model, the creation and management of galleries are the responsibility of the service provider implementation. Services are not exposed to the requester for this purpose.

7.2.1.2 Parameters

Gallery ID (input) – the identifier of the gallery or population group to which the subject will be added

Subject ID (input) – the identifier of the subject

Identity Claim (input, optional) – the identifier by which the subject is known to the gallery

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Return (output) – return value indicating success or specifying a particular error condition

7.2.1.3 Return codes

The ***Add Subject To Gallery*** service may return any of the return codes defined in Clause 9.

7.2.2 Check Quality

```
<interface name="CheckQuality">
   <parameter name="BIR"
         type="CBEFF_BIR_Type" direction="in" use="conditional" />
   <parameter name="SubjectID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="AlgorithmVendor"
         type="xs:string" direction="inout" use="optional" />
   <parameter name="AlgorithmVendorProductID"
         type="xs:string" direction="inout" use="conditional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="QualityScore"
         type="iso-iec19785-3-7:Quality" direction="out" />
   <parameter name="AlgorithmVersion" type="xs:string" direction="out" />
</interface>
```

7.2.2.1 Description

The ***Check Quality*** service shall return a quality score for a given biometric or a specified subject. Either a biometric sample or a subject ID shall be provided. The biometric input is provided in a CBEFF basic structure or CBEFF record, which in this part of ISO/IEC 30108 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 shall provide the algorithm vendor and algorithm vendor product ID that were used to calculate the biometric quality as output parameters.

NOTE Algorithm Vendors are registered with the ISO Biometric Registration Authority and assigned unique identifiers as defined in ISO/IEC 19785-2. Algorithm Product IDs are assigned by the registered algorithm vendor.

7.2.2.2 Parameters

BIR (input, conditional) – data structure containing a single biometric sample for which a quality score is to be determined; required if no Subject ID is provided

Subject ID (input, conditional) – the identifier of the subject; required if no BIR is provided

Algorithm Vendor (input, optional / output) – the identifier of the vendor of the quality algorithm used to determine the quality

Algorithm Vendor Product ID (input, conditional / output) – the vendor assigned ID for the algorithm used to determine the quality; required as input if algorithm vendor is provided

Return (output) – return value indicating success or specifying a particular error condition

Quality Score (output) – the quality of the biometric, as defined by the Quality type in the XML Patron Format specified in ISO/IEC 19785-3

Algorithm Version (output) – the version of the algorithm used to determine the quality

7.2.2.3 Return codes

The ***Check Quality*** service may return any of the return codes defined in Clause 9.

7.2.3 Classify Biometric Data

```
<interface name="ClassifyBiometricData">
   <parameter name="BIR" type="CBEFF_BIR_Type" direction="in" />
   <parameter name="ClassificationAlgorithmType"
         type="xs:string" direction="inout" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Classification" type="xs:string" direction="out" />
</interface>
```

7.2.3.1 Description

The ***Classify Biometric Data*** service shall attempt to classify a biometric sample. For example, a fingerprint biometric sample may be classified as a whorl, loop, or arch (or other classification classes and sub-classes). The types of classification algorithms and classes are not specified here, rather they are left for the implementing system to define. If no classification algorithm is input, then the BIAS service provider will make the selection.

7.2.3.2 Parameters

BIR (input) – data structure containing a single biometric sample for which the classification is to be determined

Classification Algorithm Type (input, optional / output) – identifies the type of classification algorithm to be used (input) and that was used (output) to perform the classification (e.g. for fingerprints, Henry classification)

Return (output) – return value indicating success or specifying a particular error condition

Classification (output) – the result of the classification

7.2.3.3 Return codes

The ***Classify Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.4 Create Encounter

```
<interface name="CreateEncounter">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterType" type="EncounterCategoryType" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="inout" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.4.1 Description

The ***Create Encounter*** service shall, for the specified subject, create a new encounter record and associate an encounter ID to that record. If not provided by the requester, the ***Create Encounter*** service shall generate an encounter ID that uniquely identifies the encounter within the subject record in the system.

NOTE Typically the BIAS service provider will assign the encounter ID. In the event that the requester assigns the encounter ID, it shall be used unless it duplicates an existing encounter ID in which case an error shall be returned.

The ***Create Encounter*** service is performed prior to a ***Set Biographic Data***, ***Set Biometric Data***, or ***Set Document Data*** operation.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The BIAS service provider will create the encounter and will set the encounter type to “recognition”.

7.2.4.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter Type (input) – the category of encounter

Encounter ID (input, optional / output) – the identifier of the encounter

Return (output) – return value indicating success or specifying a particular error condition

7.2.4.3 Return Codes

The ***Create Encounter*** service may return any of the return codes defined in Clause 9.

7.2.5 Create Subject

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

7.2.5.1 Description

The ***Create Subject*** service shall create a new subject record and associate a subject ID to that record. The ***Create Subject*** service shall generate a subject ID that uniquely identifies the subject in the system.

NOTE When cross-system uniqueness is required, UUIDs should be used for Subject IDs.

7.2.5.2 Parameters

Return (output) – return value indicating success or specifying a particular error condition

Subject ID (output) – the identifier of the subject

7.2.5.3 Return codes

The ***Create Subject*** service may return any of the return codes defined in Clause 9.

7.2.6 Delete Biographic Data

```
<interface name="DeleteBiographicData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterID"
            type="xs:string" direction="in" use="conditional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.6.1 Description

The ***Delete Biographic Data*** service shall erase all of the biographic data associated with a given subject record. In the encounter-centric model the service shall erase all of the biographic data associated with a given encounter, and therefore the encounter ID shall be specified. If no encounter ID is specified, or it is null, biographic data will be removed from all encounters. If a gallery is specified, biographic data will be deleted from that gallery only. 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.

7.2.6.2 Parameters

Subject ID (input) – the identifier of the subject

Gallery ID (input, optional) – the identifier of the gallery or population group from which the biographic information will be deleted

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Return (output) – return value indicating success or specifying a particular error condition

7.2.6.3 Return codes

The ***Delete Biographic Data*** service may return any of the return codes defined in Clause 9.

7.2.7 Delete Biometric Data

```
<interface name="DeleteBiometricData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="BiometricType"
         type="iso-iec19785-3-7:Multiple-types" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.7.1 Description

The ***Delete Biometric Data*** service shall remove biometric data from a given subject record. In the encounter-centric model, the encounter ID shall be specified. If no encounter ID is specified, or it is null, biometric data will be removed from all encounters. If a gallery is specified, biometric data will be deleted from that gallery only. If a biometric type(s) is specified, then only biometric data of that type shall be deleted. 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.

7.2.7.2 Parameters

Subject ID (input) – the identifier of the subject

Gallery ID (input, optional) – the identifier of the gallery or population group from which the biometric information will be deleted

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Biometric Type (input, optional) – the type of biological or behavioural data to delete, as defined by the Multiple-types type in the XML Patron Format specified in ISO/IEC 19785-3

Return (output) – return value indicating success or specifying a particular error condition

7.2.7.3 Return Codes

The ***Delete Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.8 Delete Document Data

```
<interface name="DeleteDocumentData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="DocumentCategory"
         type="xs:string" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.8.1 Description

The ***Delete Document Data*** service shall erase all of the document data of the specified category(ies) associated with a given subject record. In the encounter-centric model the service shall erase all of the document data associated with a given encounter, and therefore the encounter ID shall be specified. If no encounter ID is specified, or it is null, document data will be removed from all encounters. If

no categories are specified, then all categories (for the specified encounters) shall be deleted. 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.

7.2.8.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Document Category (input, optional) – the category(ies) of the identity documents to be retrieved

Return (output) – return value indicating success or specifying a particular error condition

7.2.8.3 Return Codes

The ***Delete Document Data*** service may return any of the return codes defined in Clause 9.

7.2.9 Delete Encounter

```
<interface name="DeleteEncounter">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID" type="xs:string" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.9.1 Description

The ***Delete Encounter*** service shall delete an existing encounter record from the system. When deleting an encounter, BIAS implementations may completely erase the encounter information in order to prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.9.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input) – the identifier of the encounter

Return (output) – return value indicating success or specifying a particular error condition

7.2.9.3 Return Codes

The ***Delete Encounter*** service may return any of the return codes defined in Clause 9.

7.2.10 Delete Subject

```
<interface name="DeleteSubject">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.10.1 Description

The ***Delete Subject*** service shall delete an existing subject record and, in an encounter-centric model, any associated encounter information from the system. This service shall also remove the subject from any registered galleries. When deleting a subject, BIAS implementations may completely erase the subject information in order to prevent the ability to reconstruct a record or records in whole or in part, or they may track and record the deleted information for auditing and/or quality control purposes.

7.2.10.2 Parameters

Subject ID (input) – the identifier of the subject

Return (output) – return value indicating success or specifying a particular error condition

7.2.10.3 Return Codes

The ***Delete Subject*** service may return any of the return codes defined in Clause 9.

7.2.11 Delete Subject From Gallery

```
<interface name="DeleteSubjectFromGallery">
   <parameter name="GalleryID" type="xs:string" direction="in" />
   <parameter name="SubjectID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="IdentityClaim"
         type="xs:string" direction="in" use="conditional" />
<parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.11.1 Description

The ***Delete Subject From Gallery*** service shall remove the registration of a subject from a gallery or population group. The subject shall be identified by either the subject ID or the claim to identity that was specified in the ***Add Subject To Gallery*** service.

7.2.11.2 Parameters

Gallery ID (input) – the identifier of the gallery or population group from which the subject will be deleted

Subject ID (input, conditional) – the identifier of the subject; required if an identity claim is not provided

Identity Claim (input, conditional) – the identifier by which the subject is known to the gallery; required if a subject ID is not provided

Return (output) – return value indicating success or specifying a particular error condition

7.2.11.3 Return Codes

The ***Delete Subject From Gallery*** service may return any of the return codes defined in Clause 9.

7.2.12 Get Identify Subject Results

```
<interface name="GetIdentifySubjectResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="CandidateList" type="CandidateListType" direction="out" />
</interface>
```

7.2.12.1 Description

The ***Get Identify Subject Results*** service shall retrieve the identification results for the specified token. This service is used in conjunction with the ***Identify Subject*** service. If the ***Identify Subject*** service is implemented as an asynchronous service, the implementing system returns a token, and the ***Get Identify Subject Results*** service is used to poll for the results of the original ***Identify Subject*** request.

7.2.12.2 Parameters

Token (input) – a value used to retrieve the results of the ***Identify Subject*** request

Return (output) – return value indicating success or specifying a particular error condition

Candidate List (output) – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample

7.2.12.3 Return Codes

The ***Get Identify Subject Results*** service may return any of the return codes defined in Clause 9.

7.2.13 Identify Subject

```
<interface name="IdentifySubject">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="Gallery "
         type="CandidateListType" direction="in" use="optional" />
   <parameter name="BIR" type="CBEFF_BIR_Type" direction="in" />
   <parameter name="MaxListSize" type="xs:positiveInteger" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="CandidateList"
         type="CandidateListType" direction="out" use="conditional" />
   <parameter name="Token"
         type="TokenType" direction="out" use="conditional" />
</interface>
```

7.2.13.1 Description

The ***Identify Subject*** service shall perform an identification search against a given gallery for a given biometric type, returning a rank-ordered candidate list of a given maximum size.

If the ***Identify Subject*** service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the candidate list. If the ***Identify Subject*** service is implemented as an asynchronous service, the implementing system shall return a token, which is an indication that the request is being handled asynchronously. In this case, the ***Get Identify Subject Results*** service shall be used to poll for the results of the ***Identify Subject*** request.

The candidate list shall include only those candidates whose comparison score exceeds a system-defined threshold, thus indicating a likely match with the subject. In the event that there are more candidates than the requested Max List Size, the system shall determine which candidate is included in the last position in the candidate list.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The BIAS service provider will create the encounter and will set the encounter type to “recognition”.

7.2.13.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group which will be searched; this parameter may also be used to identify an external system where the identification request should be forwarded, if this capability is supported by the implementing system. This parameter shall not be used in conjunction with *Gallery*

Gallery (input, optional) – a list of BIRs that shall be used instead of a stored gallery. This parameter shall not be used in conjunction with *Gallery ID*

BIR (input) – data structure containing the biometric sample for the search

NOTE: When multiple samples are included as input (e.g. in a multimodal operation), a complex BIR is used.

Max List Size (input) – the maximum size of the candidate list that should be returned

Return (output) – return value indicating success or specifying a particular error condition

Candidate List (output, conditional) – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample; returned with successful, synchronous request processing

Token (output) – a token used to retrieve the results of the ***Identify Subject*** request; returned with asynchronous request processing. If set to zero, operation is processed synchronously and candidate list is returned. If set to a non-zero value, operation is processed asynchronously and ***Get Identify Results*** must be used to retrieve the results.

7.2.13.3 Return Codes

The ***Identify Subject*** service may return any of the return codes defined in Clause 9.

NOTE Multiple scores/candidates is already built in as a score comes with a CandidateType which is a member of Candidate List.

7.2.14 List Biographic Data

```
<interface name="ListBiographicData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
         type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="BiographicDataElements"
         type="BiographicDataType" direction="out" use="conditional" />
   <parameter name="EncounterList"
         type="EncounterListType" direction="out" use="conditional" />
</interface>
```

7.2.14.1 Description

The ***List Biographic Data*** service shall list 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 service (see the ***Retrieve Biographic Data*** service 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 service shall return the list of encounter IDs which contain biographic data using the Encounter List output parameter, and the Biographic Data Elements output parameter shall be empty.

7.2.14.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

Return (output) – return value indicating success or specifying a particular error condition

Biographic Data Elements (output, conditional) – 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

Encounter List (output, conditional) – 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

7.2.14.3 Return Codes

The ***List Biographic Data*** service may return any of the return codes defined in Clause 9.

7.2.15 List Biometric Data

```
<interface name="ListBiometricData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
         type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="ListFilter"
         type="ListFilterType" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="BiometricDataElements"
         type="BiometricDataListType" direction="out" use="conditional" />
   <parameter name="EncounterList"
         type="EncounterListType" direction="out" use="conditional" />
</interface>
```

7.2.15.1 Description

The ***List Biometric Data*** service shall list the biometric data elements stored for a subject using the Biometric Data Elements output parameter. Note that no actual biometric data is returned by this service (see the ***Retrieve Biometric Data*** service to obtain the biometric data). In the encounter-centric model, an encounter ID may be specified to indicate that only the biometric data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain biometric data using the Encounter List output parameter, and the Biometric Data Elements output parameter shall be empty.

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

7.2.15.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

List Filter (input, optional) – indicates what biometric information should be returned

Return (output) – return value indicating success or specifying a particular error condition

Biometric Data Elements (output, conditional) - a list of biometric data elements associated with a subject or encounter; non-empty if the service was successful, biometric 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

Encounter List (output, conditional) – 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

7.2.15.3 Return Codes

The ***List Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.16 List Document Data

```
<interface name="ListDocumentData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
         type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="DocumentDataCategories"
         type="string" direction="out" use="conditional" />
   <parameter name="EncounterList"
         type="EncounterListType" direction="out" use="conditional" />
</interface>
```

7.2.16.1 Description

The ***List Document Data*** service shall list the document categories stored for a subject using the Document Data Elements output parameter. Note that no other document data is returned by this service (see the ***Retrieve Document Data*** service to obtain document data by category). In the encounter-centric model, an encounter ID may be specified to indicate that only the document data elements stored for that encounter should be returned. If an encounter ID is not specified and encounter data exists for the subject, the service shall return the list of encounter IDs which contain document data using the Encounter List output parameter, and the Document Data Elements output parameter shall be empty.

7.2.16.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

Return (output) – return value indicating success or specifying a particular error condition

Document Data Categories (output, conditional) – a list of document categories associated with a subject or encounter; non-empty if the service was successful, document 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

Encounter List (output, conditional) – a list of encounter ID's associated with a subject and which contain document data; non-empty if the service was successful, document data exists, the encounter-centric model is being used, and an encounter identifier was not specified

7.2.16.3 Return Codes

The ***List Document Data*** service may return any of the return codes defined in Clause 9.

7.2.17 Perform Fusion

```
<interface name="PerformFusion">
   <parameter name="FusionInput"
         type="FusionIdentityListType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Match" type="xs:boolean" direction="out" />
</interface>
```

7.2.17.1 Description

The ***Perform Fusion*** service shall accept either comparison score or comparison decision information and create a fused comparison result. The FusionInformationListType, through the FusionInformationType, provides specific elements for comparison score input and comparison decision input for a single identity, while the FusionIdentityListType provides the ability to submit

multiple identities to the Perform Fusion service (see 8.6). The fusion method and processes are left to the implementing system.

7.2.17.2 Parameters

Fusion Input (input) – score or decision input information to the fusion method for each identity

Return (output) – return value indicating success or specifying a particular error condition

Match (output) – indicates the result of the fusion method

7.2.17.3 Return Codes

The ***Perform Fusion*** service may return any of the return codes defined in Clause 9.

7.2.18 Query Capabilities

```
<interface name="QueryCapabilities">
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="CapabilityList" type="CapabilityListType" direction="out" / >
</interface>
```

7.2.18.1 Description

The ***Query Capabilities*** service shall return a list of the capabilities, options, galleries, etc. that are supported by the BIAS implementation. Table 1 provides a list of capabilities. Refer to Annex A for conformance requirements regarding which capability names an implementation must use in the ***Query Capabilities*** service. If the implementing system does not support a capability item, the Capability Value can be set to null in the response.

Proprietary and additional information may be returned by returning capabilities that are not part of the listed capabilities in Table 1. When returning capabilities that are not listed in Table 1, the Capability Description shall describe the capability.

For each capability described in Table 1, the Capability Name shall be set to that shown in Column 1.

Table 1 — List of capability items

Capability name	Capability description
AggregateInputDataOptional	A capability information item shall be provided for each data element accepted as optional input by the implementing system for the aggregate services. The Capability Value shall be equal to the name of the data element accepted by the aggregate services. The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma: — "Delete" — "Enrol" — "Identify" — "Verify" — "All"
AggregateInputDataRequired	A capability information item shall be provided for each data element required as input by the implementing system for the aggregate services. The Capability Value shall be equal to the name of the data element required by the aggregate services. The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma: — "Delete" — "Enrol" — "Identify" — "Verify" — "All"
AggregateProcessingOption	A capability information item shall be provided for each processing option supported by the implementing system for the aggregate services. The Capability Value shall be equal to the option identifier, or "key" field, for the Processing Option parameter in the aggregate services. The Capability Supporting Value shall be equal to the option value, or "value" field, for the Processing Option parameter in the aggregate services, if applicable. The Capability Additional Info shall indicate which aggregate services support the processing option, using one or more of the following values, each separated by a comma: — "Delete" — "Enrol" — "Identify" — "Verify" — "Retrieve" — "All"

Table 1 *(continued)*

Capability name	Capability description
AggregateReturnData	A capability information item shall be provided for each data element returned by the implementing system for the aggregate services. The Capability Value shall be equal to the name of the data element returned by the aggregate services. The Capability Supporting Value shall indicate which aggregate services support the data element, using one or more of the following values, each separated by a comma: — "Delete" — "Enrol" — "Identify" — "Verify" — "Retrieve" — "All"
AggregateServiceDescription	A capability information item shall be provided for each aggregate service supported by the implementing system, describing the processing logic of the aggregate system. The Capability Value shall contain a description of the processing logic of the aggregate services. The Capability Supporting Value shall indicate which aggregate service is described by this capability information item, using one of the following values: — "Delete" — "Enrol" — "Identify" — "Verify" — "Retrieve"
BiographicDataSet	A capability information item shall be provided to identify the biographic data sets supported by the implementing system. The Capability Value shall contain the name of the supported biographic data format (e.g. "EFTS" or "NIEM"). The Capability Supporting Value shall contain the version of the supported biographic data format. The Capability Additional Info shall contain the supported biographic data format type (e.g. ASCII or XML).
CBEFFPatronFormat	A capability information item shall be provided for each patron format supported by the implementing system. The Capability Value shall contain the format owner. The Capability Supporting Value shall contain the format type.
ClassificationAlgorithmType	A capability information item shall be provided for each classification algorithm type supported by the implementing system. The Capability Value shall be set to the name of the supported classification algorithm type.

Table 1 *(continued)*

Capability name	Capability description
ConformanceClass	A capability information item shall be provided to identify the conformance class of the BIAS implementation (see Annex A for more information). The Capability Value shall be set to one of the following: — "1" (for Class 1 conformance) — "2" (for Class 2 conformance) — "3" (for Class 3 conformance) — "4" (for Class 4 conformance) — "5" (for Class 5 conformance)
Gallery	A capability information item shall be provided for each gallery or population group supported by the implementing system. The Capability Value shall be equal to the value for the Gallery ID parameter in the ***Add Subject to Gallery***, ***Delete Biographic Data, Delete Biometric Data, Delete Subject From Gallery***, ***Identify Subject***, ***Retrieve Biographic Data, Retrieve Biometric Data, Retrieve Document Data, Set Biographic Data, Set Biometric Data, Set Document Data,*** and ***Verify Subject*** services.
IdentityModel	A capability information item shall be provided to identify whether the implementing system is person-centric or encounter-centric based. The Capability Value shall be set to one of the following: — "person" — "encounter"
ComparisonAlgorithm	A capability information item shall be provided for each comparison algorithm vendor and algorithm vendor product ID supported by the implementing system. The Capability Value shall contain the algorithm vendor. The Capability Supporting Value shall contain the algorithm vendor product ID. The Capability Additional Info shall be set to the biometric type, as defined by the XML Patron Format in ISO/IEC 19785-3, that corresponds to the comparison algorithm. The Capability Description shall contain the software version of the comparison algorithm.
ComparisonScore	A capability information item shall be provided to identify the use of comparison scores returned by the implementing system. The Capability Value shall be set to the end-of-score-range that signifies a match. The Capability Supporting Value shall be set to the end-of-score-range that signifies a no-match. The Capability Additional Info shall be set to the biometric type, as defined by the XML Patron Format in ISO/IEC 19785-3, that corresponds to the comparison score range.

Table 1 *(continued)*

Capability name	Capability description
QualityAlgorithm	A capability information item shall be provided for each quality algorithm vendor and algorithm vendor product ID supported by the implementing system. The Capability Value shall contain the algorithm vendor. The Capability Supporting Value shall contain the algorithm vendor product ID. The Capability Additional Info shall be set to the biometric type, as defined by the XML Patron Format in ISO/IEC 19785-3, that corresponds to the quality algorithm. The Capability Description shall contain the software version of the quality algorithm.
SupportedBiometric	A capability information item shall be provided for each biometric type supported by the implementing system. The Capability Value shall be set to the biometric type, as defined by the XML Patron Format in ISO/IEC 19785-3 (for example, the biometric type for finger is represented as “finger”). The Capability Supporting Value shall indicate if the implementing system supports comparison for the biometric type, using one of the following values: — “1” (identification) — “2” (verification) — “3” (identification and verification) — “4” (no comparison supported)
TransformOperation	A capability information item shall be provided for each transform operation type and transform control combination supported by the implementing system. The Capability Value shall be equal to the value for the Transform Operation parameter in the ***Transform Biometric Data*** service. The Capability Supporting Value shall specify the value of the Transform Control parameter in the ***Transform Biometric Data*** service. The value returned may be either a single value or a range of values. If a range of values is returned, the Capability Description shall specify additional information for the value of the Transform Control parameter. If the Transform Operation does not support a Transform Control, the Capability Supporting value shall be set to “NotApplicable”.

7.2.18.2 Parameters

Return (output) – return value indicating success or specifying a particular error condition

Capability List (output) – a list of capabilities supported by the BIAS implementation

7.2.18.3 Return Codes

The ***Query Capabilities*** service may return any of the return codes defined in Clause 9.

7.2.19 Retrieve Biographic Data

```
<interface name="RetrieveBiographicData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
```

```
        type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
        type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="BiographicDataList"
        type="BiographicDataListType" direction="out" />
</interface>
```

7.2.19.1 Description

The ***Retrieve Biographic Data*** service shall retrieve the list of biographic data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the service shall return the set of biographic data associated with that encounter (list contains a single set). If the encounter ID is not specified in the encounter-centric model, the service shall return the list of biographic information associated with the most recent encounter. If no gallery ID is specified, a list of biographic data from all galleries shall be returned.

7.2.19.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group from which the biographic information will be retrieved

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

Return (output) – return value indicating success or specifying a particular error condition

Biographic Data List (output) – a list of biographic data associated with the subject or encounter

7.2.19.3 Return Codes

The ***Retrieve Biographic Data*** service may return any of the return codes defined in Clause 9.

7.2.20 Retrieve Biometric Data

```
<interface name="RetrieveBiometricData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
        type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
        type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="BiometricType"
        type="iso-iec19785-3-7:Multiple-types" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="BIRList" type=" CBEFF_BIR_ListType " direction="out" />
</interface>
```

7.2.20.1 Description

The ***Retrieve Biometric Data*** service shall retrieve the biometric data associated with a subject ID. In the encounter-centric model, the encounter ID may be specified and the service shall return the biometric data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the service shall return the biometric information associated with the most recent encounter. If no gallery ID is specified, biometric data from all galleries shall be returned. The service provides an optional input parameter to specify that only biometric data of a certain type should be retrieved.

7.2.20.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group from which the biometric information will be retrieved

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

Biometric Type (input, optional) – the type of biological or behavioural data to retrieve, as defined by the Multiple-types type in the XML Patron Format specified in ISO/IEC 19785-3

Return (output) – return value indicating success or specifying a particular error condition

BIR List (output) – data structure containing the retrieved biometric samples

7.2.20.3 Return Codes

The ***Retrieve Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.21 Retrieve Document Data

```
<interface name="RetrieveDocumentData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="optional" />
   <parameter name="EncounterType"
         type="EncounterCategoryType" direction="in" use="optional" />
   <parameter name="DocumentCategory"
         type="xs:string" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="DocumentDataList"
         type="DocumentDataListType" direction="out" />
</interface>
```

7.2.21.1 Description

The ***Retrieve Document Data*** service shall retrieve the list of document data associated with a subject ID for the category(ies) specified. In the encounter-centric model, the encounter ID may be specified and the service shall return the list of document data associated with that encounter. If the encounter ID is not specified in the encounter-centric model, the service shall return the list of document information associated with the most recent encounter for which document data exists. If no gallery ID is specified, document data from all galleries shall be returned. If no document category is specified, all documents associated with the subject (and encounter ID, if present) shall be returned.

7.2.21.2 Parameters

Gallery ID (input) – the identifier of the gallery or population group from which the biographic information will be retrieved

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional) – the identifier of the encounter

Encounter Type (input, optional) – the category of encounter

Document Category (input, optional) – the category(ies) of the identity documents to be retrieved

Return (output) – return value indicating success or specifying a particular error condition

Document Data List (output) – a list of document data associated with the subject or encounter

7.2.21.3 Return Codes

The ***Retrieve Document Data*** service may return any of the return codes defined in Clause 9.

7.2.22 Set Biographic Data

```
<interface name="SetBiographicData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="BiographicData" type="BiographicDataType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.22.1 Description

The ***Set Biographic Data*** service shall associate biographic data to a given subject record. Depending on the identity model in use, the biographic information should replace any existing biographic information (person-centric model) or it shall be added within a new encounter (encounter-centric model). If biometric or document data was also collected during the same encounter, the same encounter ID shall be specified by the caller in order to link the biographic data with the associated biometric and/or document information (using the ***Set Biometric Data*** and ***Set Document Data*** services). If the encounter ID is omitted for the encounter-centric model, an error shall be returned.

NOTE Biographic data is typically set within the master database; however, this is implementation specific.

For encounter-based systems, the ***Create Encounter*** service shall be called prior to ***Set Biographic Data***. The Encounter ID assigned as a result (and returned by the ***Create Encounter*** service) shall be used as input to this service.

7.2.22.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group to which the biographic will be added

Subject ID (input) – the identifier of the subject

Encounter ID (input, conditional) – the identifier of the encounter

Biographic Data (input) – a list of biographic data to associate with the subject or encounter

Return (output) – return value indicating success or specifying a particular error condition

7.2.22.3 Return Codes

The ***Set Biographic Data*** service may return any of the return codes defined in Clause 9.

7.2.23 Set Biometric Data

```
<interface name="SetBiometricData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="BIRList" type="CBEFF_BIR_ListType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.23.1 Description

The ***Set Biometric Data*** service shall associate biometric data to a given subject record. Depending on the identity model in use, the biometric information should replace any existing biometric information (person-centric model) or it shall be added within a new encounter (encounter-centric model). If

biographic or document data was also collected during the same encounter, the same encounter ID shall be specified by the caller in order to link the biometric data with the associated biographic and/or document information (using the ***Set Biographic Data*** and ***Set Document Data*** services). If the encounter ID is omitted for the encounter-centric model, an error shall be returned.

NOTE Biometric data is typically set within the master database; update of the comparison engine database (if separate) is per service provider policy.

For encounter-based systems, the ***Create Encounter*** service shall be called prior to ***Set Biometric Data***. The Encounter ID assigned as a result (and returned by the ***Create Encounter*** service) shall be used as input to this service.

7.2.23.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group to which the biometric will be added

Subject ID (input) – the identifier of the subject

Encounter ID (input, optional / output) – the identifier of the encounter

BIR List (input) – data structure containing the new biometric sample(s)

Return (output) – return value indicating success or specifying a particular error condition

7.2.23.3 Return Codes

The ***Set Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.24 Set Document Data

```
<interface name="SetDocumentData">
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="DocumentData" type="DocumentDataListType" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="inout" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.24.1 Description

The ***Set Document Data*** service shall associate identity document data to a given subject record. Depending on the identity model in use, the document information should replace any existing document information for the same document category (person-centric model) or it shall be added within a new encounter (encounter-centric model). If biographic or biometric data was also collected during the same encounter, the same encounter ID shall be specified by the caller in order to link the document data with the associated biographic and/or biometric information (using the ***Set Biographic Data*** and ***Set Biometric Data*** services). If the encounter ID is omitted for the encounter-centric model, an error shall be returned.

NOTE Document data is typically set within the master database only; however, this is implementation specific.

For encounter-based systems, the ***Create Encounter*** service shall be called prior to ***Set Document Data***. The Encounter ID assigned as a result (and returned by the ***Create Encounter*** service) shall be used as input to this service.

7.2.24.2 Parameters

Gallery ID (input, optional) – the identifier of the gallery or population group to which the biographic will be added

Subject ID (input) – the identifier of the subject

Document Data (input) – a list of document data to associate with the subject or encounter

Encounter ID (input, optional / output) – the identifier of the encounter

Return (output) – return value indicating success or specifying a particular error condition

7.2.24.3 Return Codes

The ***Set Document Data*** service may return any of the return codes defined in Clause 9.

7.2.25 Transform Biometric Data

```
<interface name="TransformBiometricData">
   <parameter name="InputBIR" type="CBEFF_BIR_Type" direction="in" />
   <parameter name="TransformOperation" type="xs:unsignedLong" direction="in" />
   <parameter name="TransformControl"
         type="xs:string" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="OutputBIR" type="CBEFF_BIR_Type" direction="out" />
</interface>
```

7.2.25.1 Description

The ***Transform Biometric Data*** service shall transform or process a given biometric in one format into a new target format. Examples of transformations include:

— Feature Extraction

— Centring or cropping biometric images

— Standard biometric data format conversion

The service includes an optional parameter to allow the requestor to specify certain controls for a particular transform operation (e.g. compression ratio, target record size, or target resolution). See the ***Query Capabilities*** service for a description of the use of the Transform Operation and Transform Control parameters.

7.2.25.2 Parameters

Input BIR (input) – data structure containing the biometric information to be transformed

Transform Operation (input) – value indicating the type of transformation to perform

Transform Control (input, optional) – specifies controls for the requested transform operation

Return (output) – return value indicating success or specifying a particular error condition

Output BIR (output) – data structure containing the new, transformed biometric information

7.2.25.3 Return Codes

The ***Transform Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.26 Update Biographic Data

```
<interface name="UpdateBiographicData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="BiographicData" type="BiographicDataType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.26.1 Description

The ***Update Biographic Data*** service shall update the biographic data for an existing subject record. The service shall replace any existing biographic data with the new biographic data. In the encounter-centric model, the encounter ID shall be specified.

7.2.26.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Biographic Data (input) – list of updated biographic data elements

Return (output) – return value indicating success or specifying a particular error condition

7.2.26.3 Return Codes

The ***Update Biographic Data*** service may return any of the return codes defined in Clause 9.

7.2.27 Update Biometric Data

```
<interface name="UpdateBiometricData">
   <parameter name="SubjectID" type="xs:string" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="Merge" type="xs:boolean" direction="in" use="optional" />
   <parameter name="BIR" type="CBEFF_BIR_Type" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.27.1 Description

The ***Update Biometric Data*** service shall update a single biometric sample for an existing subject record. The service includes an optional parameter indicating if the new biometric sample should be merged with the existing biometric sample. If this parameter is set to “False” or is not used in the service request, the service shall replace the existing biometric sample with the new biometric sample. In the encounter-centric model, the encounter ID shall be specified.

NOTE The term “merge” is implementation specific; however, it may include either adding the sample to a multi-sample record or performing some level of biometric fusion (e.g. sample or feature level fusion).

7.2.27.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Merge (input, optional) – value indicating if the input biometric sample should be merged with any existing biometric information

BIR (input) – data structure containing the new biometric sample

Return (output) – return value indicating success or specifying a particular error condition

7.2.27.3 Return Codes

The ***Update Biometric Data*** service may return any of the return codes defined in Clause 9.

7.2.28 Update Document Data

```
<interface name="UpdateDocumentData">
<parameter name="SubjectID" type="xs:string" direction="in" />
<parameter name="EncounterID"
type="xs:string" direction="in" use="conditional" />
<parameter name="DocumentData" type="DocumentDataListType" direction="in" />
<parameter name="Return" type="xs:unsignedLong" direction="out" />
</interface>
```

7.2.28.1 Description

The ***Update Document Data*** service shall update the document data for an existing subject record. The service shall replace any existing document data of the same category with the new document data. In the encounter-centric model, the encounter ID shall be specified.

7.2.28.2 Parameters

Subject ID (input) – the identifier of the subject

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Document Data (input) – list of updated document data

Return (output) – return value indicating success or specifying a particular error condition

7.2.28.3 Return codes

The ***Update Document Data*** service may return any of the return codes defined in Clause 9.

7.2.29 Verify subject

```
<interface name="VerifySubject">
   <parameter name="InputBIR" type="CBEFF_BIR_Type" direction="in" />
   <parameter name="ReferenceBIR"
         type="CBEFF_BIR_Type" direction="in" use="conditional" />
   <parameter name="IdentityClaim"
         type="xs:string" direction="in" use = "conditional" />
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Match" type="xs:boolean" direction="out" />
   <parameter name="Score" type="xs:float" direction="out" use="optional" />
</interface>
```

7.2.29.1 Description

The ***Verify Subject*** service shall perform a 1:1 verification comparison between a given biometric and either a claim to identity in a given gallery or another provided biometric. As such, either the Identity Claim or Reference BIR input parameters are required.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The BIAS service provider will create the encounter and will set the encounter type to “recognition”.

7.2.29.2 Parameters

Input BIR (input) – data structure containing the biometric sample for the search

NOTE When multiple samples are included as input (e.g. in a multimodal operation), a complex BIR is used.

Reference BIR (input, conditional) – data structure containing the biometric sample that will be compared to the Input BIR, required if no Identity Claim is provided

Identity Claim (input, conditional) – the identifier by which the subject is known to the gallery, required if no Reference BIR is provided

NOTE An identity claim may be the Subject ID or some other unique piece of biographic data used by the gallery as a key (e.g. account number or username).

Gallery ID (input, optional) – the identifier of the gallery or population group of which the subject must be a member

Return (output) – return value indicating success or specifying a particular error condition

Match (output) – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

Score (output, optional) – the comparison score, if the biometric information matched

7.2.29.3 Return Codes

The ***Verify Subject*** service may return any of the return codes defined in Clause 9.

7.3 Aggregate Services

BIAS offers the following set of aggregate services. The intent of BIAS is to standardize the service request; system requirements and organizational business rules will determine how the service is implemented. While the description for an aggregate service may provide examples of how each one may be implemented using the primitive services defined in 7.2, service providers are not required to utilise any of the primitive services when implementing the aggregate services.

7.3.1 Delete

```
<interface name="Delete">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
   <parameter name="SubjectID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="InputData"
         type="InformationType" direction="in" use="optional"/>
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Token" type="TokenType" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.1.1 Description

The ***Delete*** aggregate service shall delete an existing subject or, in an encounter-centric model, an existing encounter from the system. This may be accomplished in a number of different ways according to system requirements and/or resources. For example, this aggregate service may initiate one or more ***Delete Subject from Gallery, Delete Biographic Data, Delete Biometric Data*** and ***Delete Subject*** primitive services to delete subject information from the system.

If the ***Delete*** aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the ***Delete*** aggregate service is implemented as an asynchronous service, the implementing system shall return a token in the Return Data parameter, which is an indication that the request is being handled asynchronously. In this case, the ***Get Deletion Results*** service shall be used to poll for the results of the ***Delete*** request.

7.3.1.2 Parameters

Processing Options (input) – options that guide how the service request is processed

Subject ID (input, conditional) – the identifier assigned to the subject

Encounter ID (input, conditional) – the identifier of the encounter, required for encounter-centric models

Input Data (input, optional) – contains an input data record, which may include biometric data

Return (output) – return value indicating success or specifying a particular error condition

Token (output, conditional) – a token used to retrieve the results of the ***Delete*** request; returned with asynchronous request processing. If set to zero, operation is processed synchronously. If set to a non-zero value, operation is processed asynchronously and ***Get Deletion Results*** must be used to retrieve the results.

Return Data (output, optional) – contains a return data record

7.3.1.3 Return Codes

The ***Delete*** aggregate service may return any of the return codes defined in Clause 9.

7.3.2 Enrol

```
<interface name="Enrol">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
   <parameter name="InputData" type="InformationType" direction="in" />
   <parameter name="EncounterID"
         type="xs:string" direction="inout" use="conditional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="SubjectID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="Token" type="TokenType" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.2.1 Description

The ***Enrol*** aggregate service shall add a new subject or, in an encounter-centric model, a new encounter to the system. This may be accomplished in a number of different ways according to system requirements and/or resources. For example, this aggregate service may initiate one or more ***Identify Subject*** primitive service requests to determine if the given subject is already known to the system. If the subject is not previously known to the system, any or all of the ***Create Subject***, ***Set Biographic Data***, ***Set Biometric Data***, and ***Add Subject to Gallery*** primitive services may be utilised to add subject information to the system. If the subject is previously known to the system, the service may (1) do nothing; (2) initiate an ***Update Biographic Data*** and/or ***Update Biometric Data*** primitive service request in a person-centric model; or (3) initiate a ***Set Biographic Data*** and/or ***Set Biometric Data*** primitive service request in an encounter-centric model.

For encounter-centric models, the encounter ID may optionally be specified by the caller. If the encounter ID is omitted for the encounter-centric model, the service shall return a system-assigned encounter ID.

If the ***Enrol*** aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the ***Enrol*** aggregate service is implemented as an asynchronous service, the implementing system shall return a non-zero token in the Token parameter, which is an indication that the request is being handled asynchronously. In this case, the ***Get Enrol Results*** service shall be used to poll for the results of the ***Enrol*** request.

7.3.2.2 Parameters

Processing Options (input) – options that guide how the service request is processed

Input Data (input) – contains a subject enrolment record

Encounter ID (input, optional / output, conditional) – the identifier of the encounter; required for encounter-centric models

Return (output) – return value indicating success or specifying a particular error condition

Subject ID (output, conditional) – the identifier assigned to the subject

Token (output, conditional) – a token used to retrieve the results of the ***Enrol*** request; returned with asynchronous request processing. If set to zero, operation is processed synchronously. If set to a non-zero value, operation is processed asynchronously and ***Get Enrol Results*** must be used to retrieve the results.

Return Data (output, optional) – contains a return data record

7.3.2.3 Return codes

The ***Enrol*** aggregate service may return any of the return codes defined in Clause 9.

7.3.3 Get Deletion Results

```
<interface name="GetDeletionResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.3.1 Descriptions

The ***Get Deletion Results*** aggregate service shall retrieve the deletion results for the specified token. This service is used in conjunction with the ***Delete*** aggregate service. If the ***Delete*** aggregate service is implemented as an asynchronous service, the implementing system returns a token, and the ***Get Deletion Results*** service is used to poll for the results of the original ***Delete*** request.

7.3.3.2 Parameters

Token (input) – a value used to retrieve the results of the ***Delete*** request

Return (output) – return value indicating success or specifying a particular error condition

Return Data (output, optional) – contains a return data record

7.3.3.3 Return Codes

The ***Get Deletion Results*** aggregate service may return any of the return codes defined in Clause 9.

7.3.4 Get Enrol Results

```
<interface name="GetEnrolResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="EncounterID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="SubjectID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.4.1 Descriptions

The ***Get Enrol Results*** aggregate service shall retrieve the enrolment results for the specified token. This service is used in conjunction with the ***Enrol*** aggregate service. If the ***Enrol*** aggregate service is implemented as an asynchronous service, the implementing system returns a token, and the ***Get Enrol Results*** service is used to poll for the results of the original ***Enrol*** request.

7.3.4.2 Parameters

Token (input) – a value used to retrieve the results of the ***Enrol*** request

Return (output) – return value indicating success or specifying a particular error condition

Encounter ID (output, conditional) – the identifier of the encounter, if assigned

Subject ID (output, conditional) – the identifier assigned to the subject

Return Data (output, optional) – contains a return data record

7.3.4.3 Return Codes

The ***Get Enrol Results*** aggregate service may return any of the return codes defined in Clause 9.

7.3.5 Get Identify Results

```
<interface name="GetIdentifyResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="CandidateList"
         type="CandidateListType" direction="out" use="conditional" />
   <parameter name="EncounterID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.5.1 Description

The ***Get Identify Results*** aggregate service shall retrieve the identification results for the specified token. This service is used in conjunction with the ***Identify*** aggregate service. If the ***Identify*** aggregate service is implemented as an asynchronous service, the implementing system returns a non-zero token, and the ***Get Identify Results*** service is used to poll for the results of the original ***Identify*** request.

7.3.5.2 Parameters

Token (input) – a value used to retrieve the results of the ***Identify*** request

Return (output) – return value indicating success or specifying a particular error condition

Candidate List (output, conditional) – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample

Encounter ID (output, conditional) – the identifier of the encounter, if assigned

Return Data (output, optional) – contains a return data record

7.3.5.3 Return codes

The ***Get Identify Results*** aggregate service may return any of the return codes defined in Clause 9.

7.3.6 Get Update Results

```
<interface name="GetUpdateResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.6.1 Descriptions

The ***Get Update Results*** aggregate service shall retrieve the update results for the specified token. This service is used in conjunction with the ***Update*** aggregate service. If the ***Update*** aggregate service is implemented as an asynchronous service, the implementing system returns a token, and the ***Get Update Results*** service is used to poll for the results of the original ***Update*** request.

7.3.6.2 Parameters

Token (input) – a value used to retrieve the results of the ***Update*** request

Return (output) – return value indicating success or specifying a particular error condition

Return Data (output, optional) – contains a return data record

7.3.6.3 Return Codes

The ***Get Update Results*** aggregate service may return any of the return codes defined in Clause 9.

7.3.7 Get Verify Results

```
<interface name="GetVerifyResults">
   <parameter name="Token" type="TokenType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Match" type="xs:boolean" direction="out" />
   <parameter name="Score" type="xs:float" direction="out" use="optional" />
   <parameter name="EncounterID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.7.1 Description

The ***Get Verify Results*** aggregate service shall retrieve the verification results for the specified token. This service is used in conjunction with the ***Verify*** aggregate service. If the ***Verify*** aggregate service is implemented as an asynchronous service, the implementing system returns a non-zero token, and the ***Get Verify Results*** service is used to poll for the results of the original ***Verify*** request.

7.3.7.2 Parameters

Token (input) – a value used to retrieve the results of the ***Verify*** request

Return (output) – return value indicating success or specifying a particular error condition

Match (output) – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

Score (output, optional) – the comparison score, if the biometric information matched

Encounter ID (output, conditional) – the identifier of the encounter, if assigned

Return Data (output, optional) – contains a return data record

7.3.7.3 Return Codes

The ***Get Verify Results*** aggregate service may return any of the return codes defined in Clause 9.

7.3.8 Identify

```
<interface name="Identify">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
```

```
    <parameter name="InputData" type="InformationType" direction="in" />
    <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
    <parameter name="MaxListSize" type="xs: positiveInteger" direction="in" />
    <parameter name="Return" type="xs:unsignedLong" direction="out" />
    <parameter name="CandidateList"
          type="CandidateListType" direction="out" use="conditional" />
    <parameter name="EncounterID"
          type="xs:string" direction="out" use="conditional" />
    <parameter name="Token" type="TokenType" direction="out" use="conditional" />
    <parameter name="ReturnData"
          type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.8.1 Description

The ***Identify*** aggregate service shall perform an identification function according to system requirements and/or resources. For example, a system may have multiple galleries of subjects, and may utilise any or all of these galleries, via calls to the ***Identify Subject*** primitive service, to perform a system-level identification function. The system may perform additional actions based on input data and/or results of the ***Identify Subject*** primitive service requests. For example, in an encounter-centric model, this aggregate service may search three separate galleries of subjects, and if a match is found it may then utilise the ***Set Biographic Data*** and/or ***Set Biometric Data*** primitive services to create a new encounter for the subject. If this occurs, the service shall return a system-assigned encounter ID.

If the ***Identify*** aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the ***Identify*** aggregate service is implemented as an asynchronous service, the implementing system shall return a non-zero token in the Token parameter, which is an indication that the request is being handled asynchronously. In this case, the ***Get Identify Results*** service shall be used to poll for the results of the ***Identify*** request.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The BIAS service provider will create the encounter and will set the encounter type to “recognition”.

7.3.8.2 Parameters

Processing Options (input) – options that guide how the service request is processed

Input Data (input) – contains an input data record, which at a minimum must include biometric data

Gallery ID (input, optional) – the identifier of the gallery or population group which will be searched; this parameter may also be used to identify an external system where the identification request should be forwarded, if this capability is supported by the implementing system

Max List Size (input) – the maximum size of the candidate list that should be returned

Return (output) – return value indicating success or specifying a particular error condition

Candidate List (output, conditional) – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample; returned with successful, synchronous request processing

Encounter ID (output, conditional) – the identifier of the encounter, if assigned

Token (output, conditional) – a token used to retrieve the results of the ***Identify*** request; returned with asynchronous request processing. If set to zero, operation is processed synchronously and candidate list is returned. If set to a non-zero value, operation is processed asynchronously and ***Get Identify Results*** must be used to retrieve the results.

Return Data (output, optional) – contains a return data record

7.3.8.3 Return Codes

The ***Identify*** aggregate service may return any of the return codes defined in Clause 9.

7.3.9 Retrieve Data

```
<interface name="RetrieveData">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
   <parameter name="SubjectID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="ReturnData" type="InformationType" direction="out" />
</interface>
```

7.3.9.1 Description

The ***Retrieve Data*** aggregate service shall retrieve requested information about a subject, or in an encounter-centric model about an encounter. In a person-centric model, this aggregate service may be used to retrieve both biographic and biometric information for a subject record. In an encounter-centric model, this aggregate service may be used to retrieve biographic and/or biometric information for either a single encounter or all encounters. Either a subject ID or encounter ID must be specified.

7.3.9.2 Parameters

Processing Options (input) – options that guide how the service request is processed, and may identify what type(s) of information should be returned

Subject ID (input, conditional) – the identifier of the subject; required if no encounter ID is provided

Encounter ID (input, conditional) – the identifier of the encounter; required if no subject ID is provided

Return (output) – return value indicating success or specifying a particular error condition

Return Data (output) – contains a return data record

7.3.9.3 Return Codes

The ***Retrieve Data*** aggregate service may return any of the return codes defined in Clause 9.

7.3.10 Update

```
<interface name="Update">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
   <parameter name="SubjectID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="EncounterID"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="InputData" type="InformationType" direction="in" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Token" type="TokenType" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.10.1 Description

The ***Update*** aggregate service shall update specified information about a subject, or in an encounter-centric model about an encounter. In a person-centric model, this aggregate service may be used to update both biographic and biometric information for a subject record. In an encounter-centric model, this aggregate service may be used to update biographic and/or biometric information for either a single encounter or all encounters. Either a subject ID or encounter ID must be specified.

7.3.10.2 Parameters

Processing Options (input) – options that guide how the service request is processed, and may identify what type(s) of information should be returned

Subject ID (input, conditional) – the identifier of the subject; required if no encounter ID is provided

Encounter ID (input, conditional) – the identifier of the encounter; required if no subject ID is provided

Input Data (input) – contains subject information to update

Return (output) – return value indicating success or specifying a particular error condition

Token (output, conditional) – a token used to retrieve the results of the Update request; returned with asynchronous request processing. If set to zero, operation is processed synchronously. If set to a non-zero value, operation is processed asynchronously and Get Update Results must be used to retrieve the results.

Return Data (output, optional) – contains a return data record

7.3.10.3 Return Codes

The ***Update*** aggregate service may return any of the return codes defined in Clause 9.

7.3.11 Verify

```
<interface name="Verify">
   <parameter name="ProcessingOptions"
         type="ProcessingOptionsType" direction="in" />
   <parameter name="InputData" type="InformationType" direction="in" />
   <parameter name="ReferenceBIR"
         type="CBEFF_BIR_Type" direction="in" use="conditional" />
   <parameter name="IdentityClaim"
         type="xs:string" direction="in" use="conditional" />
   <parameter name="GalleryID" type="xs:string" direction="in" use="optional" />
   <parameter name="Return" type="xs:unsignedLong" direction="out" />
   <parameter name="Match" type="xs:boolean" direction="out" />
   <parameter name="Score" type="xs:float" direction="out" use="optional" />
   <parameter name="EncounterID"
         type="xs:string" direction="out" use="conditional" />
   <parameter name="Token" type="TokenType" direction="out" use="conditional" />
   <parameter name="ReturnData"
         type="InformationType" direction="out" use="optional" />
</interface>
```

7.3.11.1 Description

The ***Verify*** aggregate service shall perform a 1:1 verification function between a given biometric and either a claim of identity in a given gallery or another provided biometric and according to system requirements and/or resources. Either the Identity Claim or Reference BIR input parameters are required. The system may perform additional actions based on input data and/or results of verification. For example, in an encounter-centric model, this aggregate service may initiate a request to the ***Verify Subject*** primitive service, and if a match is found it may then utilise the ***Set Biographic Data*** and/or ***Set Biometric Data*** primitive services to create a new encounter for the subject. If this occurs, the service shall return a system-assigned encounter ID.

For encounter-centric models, the encounter ID may optionally be specified by the caller. If the encounter ID is omitted for the encounter-centric model, the service shall return a system-assigned encounter ID.

NOTE When in encounter mode, for comparison operations, it is not necessary to explicitly create an encounter. The BIAS service provider will create the encounter and will set the encounter type to “recognition”.

If the ***Verify*** aggregate service is implemented as a synchronous service, the implementing system shall immediately process the request and return the results in the Return Data parameter. If the ***Verify***

aggregate service is implemented as an asynchronous service, the implementing system shall return a non-zero token in the Token parameter, which is an indication that the request is being handled asynchronously. In this case, the ***Get Verify Results*** service shall be used to poll for the results of the ***Verify*** request.

7.3.11.2 Parameters

Processing Options (input) – options that guide how the service request is processed, and may identify what type(s) of information should be returned

Input Data (input) – contains an input data record, which at a minimum must include biometric data

Reference BIR (input, conditional) – data structure containing the biometric sample that will be compared to the Input BIR, required if no Identity Claim is provided

Identity Claim (input, conditional) – the identifier by which the subject is known to the gallery, required if no Reference BIR is provided

Gallery ID (input, optional) – the identifier of the gallery or population group of which the subject must be a member

Return (output) – return value indicating success or specifying a particular error condition

Match (output) – indicates if the Input BIR matched either the biometric information associated with the Identity Claim or the Reference BIR

Score (output, optional) – the comparison score, if the biometric information matched

Encounter ID (output, conditional) – the identifier of the encounter, if assigned

Token (output, conditional) – a token used to retrieve the results of the ***Verify*** request; returned with asynchronous request processing. If set to zero, operation is processed synchronously. If set to a non-zero value, operation is processed asynchronously and ***Get Verify Results*** must be used to retrieve the results.

Return Data (output, optional) – contains a return data record

7.3.11.3 Return Codes

The ***Verify*** aggregate service may return any of the return codes defined in Clause 9.

8 Data elements and data types

A goal of BIAS is to be flexible to the amount and types of biographic and biometric information available to and used by a system. The parameters “Biographic Data” and “Biometric Data” are meant to be general in this sense in order to allow this flexibility. This clause includes information on how this flexibility can be specified and supported by implementing systems.

8.1 Biographic data

BIAS defines three data types to provide flexibility for the amount and types of biographic data supported by implementing systems. The Biographic Data Item Type shall represent a single biographic data item, and the Biographic Data Set Type shall represent a set of biographic information in a specified format. The Biographic Data Type is a common type that shall represent either a set or list of biographic data.

8.1.1 Biographic Data Type

```
<xs:complexType name="BiographicDataType">
   <xs:choice maxOccurs="unbounded">
      <xs:element name="LastName" type="xs:string" minOccurs="0" />
```

```
      <xs:element name="FirstName" type="xs:string" minOccurs="0" />
      <xs:element name="BiographicDataItem" type="BiographicDataItemType"
            maxOccurs="unbounded" />
       <xs:element name="BiographicDataSet" type="BiographicDataSetType" />
   </xs:choice>
</xs:complexType>
```

8.1.1.1 Description

The Biographic Data Type defines a set of biographic data elements, utilizing either the Biographic Data Item Type to represent a list of elements or the Biographic Data Set Type to represent a complete, formatted set of biographic information. This type also includes optional fixed fields for representing first and last names, two common biographic data elements.

8.1.1.2 Definitions

Last Name (optional) – the last name of a subject

First Name (optional) – the first name of a subject

Biographic Data Item – a single biographic data element

Biographic Data Set – a set of biographic data information

8.1.2 Biographic Data Item Type

```
<xs:complexType name="BiographicDataItemType">
   <xs:sequence>
      <xs:element name="Name" type="xs:string" />
      <xs:element name="Type" type="xs:string" />
      <xs:element name="Value" type="xs:string" minOccurs="0" />
   </xs:sequence>
</xs:complexType>
```

8.1.2.1 Description

The Biographic Data Item Type defines a single biographic data element. The biographic data item *name* and *type* are required elements, while the *value* is optional.

8.1.2.2 Definitions

Name – the name of the biographic data item (i.e. “PersonName”)

Type – the data type for the biographic data item (i.e. “xs:string”)

Value (optional) – the value assigned to the biographic data item (i.e. “John Doe”)

8.1.3 Biographic Data Set Type

```
<xs:complexType name="BiographicDataSetType">
   <xs:sequence>
      <xs:any namespace="##any"/>
   </xs:sequence>
   <xs:attribute name="name" type="xs:string" use="required" />
   <xs:attribute name="version" type="xs:string" use="optional" />
   <xs:attribute name="source" type="xs:string" use="required" />
   <xs:attribute name="type" type="xs:string" use="required" />
</xs:complexType>
```

8.1.3.1 Description

The Biographic Data Set Type defines a set of biographic data that is formatted according to the specified format. This data type allows biographic information in an Electronic Fingerprint Transmission Specification (EFTS) or pre-defined XML format, for example, to be used with BIAS messages.

8.1.3.2 Definitions

name – the name of the biographic data format (e.g. “EFTS”)

version (optional) – the version of the biographic data format (e.g. “7.1” or “1.0”)

source – reference to a URI describing the biographic data format

type – the biographic data format type (e.g. “XML”)

8.1.3.3 Common Biographic Data Format References

In order to provide a consistent representation of some common biographic data formats, this specification will establish common values for the attributes used in the Biographic Data Set Type. The common biographic data format shall have the *name, source, and type* attributes in common. The *version* attribute, if used, shall reflect the source organization’s assigned version. These shall be registered with the ISO Biometric Registration Authority. Samples of common biographic data formats are shown in Table B.1 in Annex B.

8.1.4 Biographic Data List Type

```
<xs:complexType name="BiographicDataListType">
   <xs:sequence>
      <xs:element name="BiographicData" type="BiographicDataType"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.1.4.1 Description

The Biographic Data List Type shall provide a list of biographic data.

8.1.4.2 Definitions

Biographic Data – data structure containing information about a biographic record

8.2 Biometric Data

This defines how to represent biometric data in the BIAS services.

8.2.1 CBEFF BIR Type

```
<xs:complexType name="CBEFF_BIR_Type">
   <xs:sequence>
      <xs:element name="BIR_Information" minOccurs="0">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="bir-info" type="iso-iec19785-3-7:BIR-info"
                     minOccurs="0" />
               <xs:element name="bdb-info" type="iso-iec19785-3-7:BDB-info"
                     minOccurs="0" />
               <xs:element name="sb-info" type="iso-iec19785-3-7:SB-info"
                     minOccurs="0" />
            </xs:sequence>
         </xs:complexType>
      </xs:element>
      <xs:element name="BIR">
         <xs:complexType>
            <xs:choice minOccurs="1">
               <xs:element name="BinaryBIR" type="xs:base64Binary" />
               <xs:element name="URI_BIR" type="xs:anyURI" />
               <xs:element name="XML_BIR" type="iso-iec19785-3-7:BIR" />
            </xs:choice>
         </xs:complexType>
      </xs:element>
```

```
   </xs:sequence>
   <xs:attribute name="format-owner" type="iso-iec19785-3-7:Registered-int"
         use="required" />
   <xs:attribute name="format-type" type="iso-iec19785-3-7:Registered-int"
         use="required" />
</xs:complexType>
```

8.2.1.1 Description

Biometric information shall be packaged as a biometric information record (BIR) in a CBEFF structure, called a CBEFF-BIR in this part of ISO/IEC 30108, with the biometric data embedded in the biometric data block, as defined by ISO/IEC 19785-1. This part of ISO/IEC 30108 does not require any specific CBEFF patron format, and recognises that BIAS implementations may support only one patron format. Applications and implementations may also choose to support multiple patron formats.

The CBEFF BIR Type schema shall be used to represent biometric information. This schema provides for either a non-XML CBEFF-BIR representation, a reference to a URI containing a CBEFF-BIR, or an XML CBEFF-BIR representation. Non-XML CBEFF-BIR representations shall be base-64 encoded, converting non-ASCII text and binary image data into ASCII form. XML CBEFF-BIR representations shall adhere to the XML Patron Format as defined in ISO/IEC 19785-3, Amd1(2010). The XML Patron Format is referenced with the namespace "http://standards.iso.org/iso-iec/19785/-3".

BIR metadata information may be optionally included to make it easier to understand what is in a CBEFF-BIR without having to parse the actual BIR content. This is especially helpful when using the non-XML CBEFF-BIR representation or a reference to a URI containing a CBEFF-BIR.

NOTE BIR information elements are intended to reflect information in the BIR (if a simple BIR) or information present in the topmost level of a BIR (if a complex BIR).

8.2.1.2 Definitions

bir-info – contains information about the CBEFF-BIR

bdb-info – contains information about the BDB in a simple CBEFF-BIR

sb-info – contains information about the security block, if used, in a simple CBEFF-BIR

Binary BIR – a non-XML CBEFF-BIR

URI BIR – a URI reference to a CBEFF-BIR

XML BIR – an XML CBEFF-BIR, using the XML Patron Format as defined in ISO/IEC 19785-3

format-owner – identifies the Patron format owner

format-type – identifies the Patron format type

8.2.2 CBEFF BIR List Type

```
<xs:complexType name="CBEFF_BIR_ListType">
   <xs:sequence>
      <xs:element name="BIR" type="CBEFF_BIR_Type"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

NOTE As an alternative to a BIR List, a complex BIR can be used which contains multiple BIRs/modalities.

8.2.2.1 Description

The CBEFF BIR List Type shall provide a list of CBEFF-BIR elements.

8.2.2.2 Definitions

BIR – CBEFF structure containing biometric data and associated metadata

8.2.3 Biometric Data Element Type

```
<xs:complexType name="BiometricDataElementType">
   <xs:sequence>
      <xs:element name="BiometricType" type="iso-iec19785-3-7:Multiple-types" />
      <xs:element name="BiometricTypeCount" type="xs:positiveInteger"
            minOccurs="0" />
      <xs:element name="BiometricSubType" type="iso-iec19785-3-7:Subtype"
            minOccurs="0" />
      <xs:element name="BDBFormatOwner" type="iso-iec19785-3-7:Registered-int" />
      <xs:element name="BDBFormatType" type="iso-iec19785-3-7:Registered-int" />
   </xs:sequence>
</xs:complexType>
```

8.2.3.1 Description

The Biometric Data Element Type shall provide descriptive information about biometric data, such as the biometric type, subtype, and format, contained in the BDB of the CBEFF-BIR.

8.2.3.2 Definitions

Biometric Type – the type of biological or behavioral data stored in the biometric record, as defined by CBEFF

Biometric Type Count (optional) – the number of biometric records having the biometric type recorded in the biometric type field

Biometric Subtype (optional) – more specifically defines the type of biometric data stored in the biometric record, as defined by CBEFF

BDB Format Owner – identifies the standards body, working group, industry consortium, or other CBEFF biometric organization that has defined the format for the biometric data

BDB Format Type – identifies the specific biometric data format specified by the CBEFF biometric organization recorded in the BDB Format Owner field

8.2.4 Biometric Data List Type

```
<xs:complexType name="BiometricDataListType">
   <xs:sequence>
      <xs:element name="BiometricDataElement" type="BiometricDataElementType"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.2.4.1 Description

The Biometric Data List Type shall provide a list of biometric data elements.

8.2.4.2 Definitions

Biometric Data Element – data structure containing information about a biometric record

8.3 Document Data

BIAS includes metadata about one or more identity documents as well as (optionally) images of scanned identity documents.

8.3.1 Document Data Type

```
<xs:complexType name="DocumentDataType">
   <xs:sequence maxOccurs="unbounded">
      <xs:element name="DocumentCategory" type="xs:string" />
      <xs:element name="DocumentIDNumber" type="xs:string" minOccurs="0" />
      <xs:element name="DocumentIssuanceCountryCode"
            type="iso_3166:CountryAlpha2CodeType" minOccurs="0" />
      <xs:element name="DocumentIssuingOrganization" type="xs:string"
            minOccurs="0" />
      <xs:element name="DocumentIssuanceDate" type="xs:dateTime"
            minOccurs="0" />
      <xs:element name="DocumentExpirationDate" type="xs:dateTime"
            minOccurs="0" />
      <xs:element name="DocumentLastName" type="xs:string" minOccurs="0" />
      <xs:element name="DocumentFirstName" type="xs:string" minOccurs="0" / >
      <xs:element name="DocumentMiddleName" type="xs:string" minOccurs="0" />
      <xs:element name="DocumentValidity" type="xs:boolean" minOccurs="0" / >
      <xs:element name="DocumentValidityText" type="xs:string" minOccurs="0" />
      <xs:element name="DocumentImage" type="xs:base64Binary" minOccurs="0" />
   </xs:sequence>
</xs:complexType>
```

8.3.1.1 Description

The Document Data Type defines a set of document data elements providing information about the presented identity document. This type also includes an optional field for an image of the document.

8.3.1.2 Definitions

Document Category – the type of identity document presented (e.g. passport)

Document ID Number (optional) – the number associated with the identity document (e.g. passport number)

Document Issuance Country Code (optional) – the ISO 2-character code for the country which issued the document or from within which it was issued

Document Issuing Organization (optional) – the entity which issued the identity document

Document Issuance Date (optional) – the date upon which the identity document was issued

Document Expiration Date (optional) – the date upon which the identity document is no longer valid (expires)

Document Last Name (optional) – the family name of the person to whom the identity document was issued, as contained within the document itself

Document First Name (optional) – the first given name of the person to whom the identity document was issued, as contained within the document itself

Document Middle Name (optional) – the second given name of the person to whom the identity document was issued, as contained within the document itself

Document Validity (optional) – the assessed validity of the identity document (e.g. as the result of local or online validity checks)

Document Validity Text (optional) – details or remarks associated with the assessed validity (e.g. description of validity issue)

Document Image – a scanned image of the subject document (e.g. passport picture page)

8.3.2 Document Data List Type

```
<xs:complexType name="DocumentDataListType">
   <xs:sequence>
      <xs:element name="Document" type="DocumentDataType"
            minOccurs="0" maxOccurs="unbounded" />
```

```
   </xs:sequence>
</xs:complexType>
```

8.3.2.1 Description

The Document Data List Type shall provide a list of documents.

8.3.2.2 Definitions

Document – data structure containing information about a document and optionally an image of that document

8.4 Candidate Lists

Candidate lists are returned in the response to a biometric identification request. BIAS defines two data types to represent candidate lists. The Candidate Type shall represent a single candidate, and the Candidate List Type shall represent a set or list of candidates.

8.4.1 Candidate Type

```
<xs:complexType name="CandidateType">
   <xs:sequence>
      <xs:element name="ScoreList" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Score" type="ScoreType" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    <xs:element name="Subject ID" type="xs:string" />
    <xs:element name="BiographicData" type="BiographicDataType"
          minOccurs="0" />
 </xs:sequence>
 <xs:attribute name="rank" type="xs:int" use="required" />
</xs:complexType>

<xs:complexType name="ScoreType">
   <xs:sequence>
      <xs:element name="Value" type="xs:float" />
      <xs:element name="BiometricType" type="iso-iec19785-3-7:Multiple-types"minOccurs="0"
/>
      <xs:element name="BiometricSubType" type="iso-iec19785-3-7:Subtype" minOccurs="0" />
   </xs:sequence>
</xs:complexType>
```

8.4.1.1 Description

The Candidate Type defines a single candidate as a possible match in response to a biometric identification request. The *Subject ID* is a required element, while the *score* and *biographic data* are optional.

8.4.1.2 Definitions

Rank – the rank of the candidate in relation to other candidates for the same biometric identification operation

Subject ID – the identifier of the subject

ScoreList (optional) – a list of comparison score(s) and optionally the type and subtype of the relating biometric

Biographic Data (optional) – biographic data associated with the matching candidate

8.4.2 Candidate List Type

```
<xs:complexType name="CandidateListType">
   <xs:sequence>
      <xs:element name="Candidate" type="CandidateType"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.4.2.1 Description

The Candidate List Type defines a set of candidates, utilizing the Candidate Type to represent each element in the set.

8.4.2.2 Definitions

Candidate – a single candidate

8.5 Capabilities

Implementing systems will have various capabilities to support BIAS services. These capabilities include information on supported biometric comparison capabilities, supported galleries, supported processing options for the aggregate services, supported biographic formats, etc. BIAS defines two data types to represent these capabilities. The Capability Type shall represent a single capability, and the Capability List Type shall represent a set of capabilities.

8.5.1 Capability Type

```
<xs:complexType name="CapabilityType">
   <xs:sequence>
      <xs:element name="CapabilityName" type="xs:string" />
      <xs:element name="CapabilityID" type="xs:string" minOccurs="0" />
      <xs:element name="CapabilityDescription" type="xs:string" minOccurs="0" />
      <xs:element name="CapabilityValue" type="xs:string" minOccurs="0" />
      <xs:element name="CapabilitySupportingValue" type="xs:string"
            minOccurs="0" />
      <xs:element name="CapabilityAdditionalInfo" type="xs:string"
            minOccurs="0" />
   </xs:sequence>
</xs:complexType>
```

8.5.1.1 Description

The Capability Type defines a single capability supported by an implementing system. Each supported capability shall be identified by a *Capability Name*. Some supported capabilities will have an associated *Capability Value* (e.g. supported galleries will have a Gallery ID value), while others will not (e.g. biometric comparison support for a specific biometric type).

8.5.1.2 Definitions

Capability Name – the name of the capability, as defined by the implementing system

Capability ID (optional) – an identifier assigned to the capability by the implementing system

Capability Description (optional) – a description of the capability

Capability Value (optional) – a value assigned to the capability

Capability Supporting Value (optional) – a secondary value supporting the capability

Capability Additional Info (optional) – contains additional information for the supported capability

8.5.2 Capability List Type

```
<xs:complexType name="CapabilityListType">
   <xs:sequence>
```

```
      <xs:element name="Capability" type="CapabilityType"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.5.2.1 Description

The Capability List Type defines a set of capabilities, utilizing the Capability Type to represent each element in the set.

8.5.2.2 Definitions

Capability – a single capability

8.6 Fusion Information

Fusion information is sent as input to fusion services. BIAS defines two data types to represent fusion information. The Fusion Information Type shall represent a single set of fusion information, and the Fusion Information List Type shall represent a set or list of fusion information elements.

8.6.1 Fusion Information Type

```
<xs:complexType name="FusionInformationType">
   <xs:sequence>
      <xs:element name="BiometricType" type="iso-iec19785-3-7:Multiple-types" />
      <xs:element name="BiometricSubType" type="iso-iec19785-3-7:Subtype"
            minOccurs="0" />
       <xs:element name="AlgorithmOwner" type="xs:string" />
      <xs:element name="AlgorithmType" type="xs:string" />
      <xs:choice minOccurs="1">
         <xs:element name="Score" type="xs:float" />
         <xs:element name="Decision" type="xs:string" />
      </xs:choice>
   </xs:sequence>
</xs:complexType>
```

8.6.1.1 Description

The Fusion Information Type represents the information necessary to perform a fusion operation. It shall include the biometric type and subtype, if applicable, for which the biometric comparison was performed, the comparison algorithm identifying information, and either a score (for score-level fusion) or a decision (for decision-level fusion).

8.6.1.2 Definitions

Biometric Type – the type of biological or behavioral data stored in the biometric record, as defined by CBEFF

Biometric Subtype (optional) – more specifically defines the type of biometric data stored in the biometric record

Algorithm Owner – the owner or vendor of the algorithm used to determine the score or decision

Algorithm Type – the Algorithm Owner's identifier for the specific algorithm product and version used to determine the score or decision

NOTE Algorithm Owners are registered with the ISO Biometric Registration Authority and assigned unique identifiers as defined in ISO/IEC 19785-2. Algorithm Types are assigned by the registered algorithm owner.

Score – the similarity score assigned by the comparison algorithm

Decision – the match decision assigned by the comparison algorithm

8.6.2 Fusion Information List Type


```
<xs:complexType name="FusionInformationListType">
   <xs:sequence>
      <xs:element name="FusionElement" type="FusionInformationType"
            minOccurs="2" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.6.2.1 Description

The Fusion Information List Type shall contain at a minimum two sets of fusion input elements, utilizing the Fusion Information Type to represent a single set of fusion information.

8.6.2.2 Definitions

Fusion Element – a set of fusion information

8.6.3 Fusion Identity List Type

```
<xs:complexType name="FusionIdentityListType">
   <xs:sequence>
      <xs:element name="FusionIdentity" type="FusionInformationListType"
            maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.6.3.1 Description

The Fusion Identity List Type shall contain fusion input elements for one or more identities, utilizing the Fusion Information List Type to represent a single set of fusion information for each identity.

8.6.3.2 Definitions

Fusion Identity – a set of fusion information for a single identity

8.7 Other Data Types

This describes the remaining data types defined by this part of ISO/IEC 30108.

8.7.1 Encounter Category Type

```
<xs:simpleType name="EncounterCategoryType">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Enrolment"/>
      <xs:enumeration value="Recognition"/>
      <xs:enumeration value="Unspecified"/>
      <xs:pattern value="([a-zA-Z0-9])+"/>
   </xs:restriction>
</xs:simpleType>
```

8.7.1.1 Description

The Encounter Category Type identifies the type of encounter (interaction) during which the identity (biographic, biometric, and/or document) data was collected from the subject as determined by the requester.

8.7.1.2 Definition

As given above.

8.7.2 Encounter List Type

```
<xs:complexType name="EncounterListType">
   <xs:sequence>
      <xs:element name="EncounterID" type="xs:string"
```

```
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.7.2.1 Description

The Encounter List Type defines a set of encounters.

8.7.2.2 Definition

Encounter ID – the identifier of an encounter (unique to a subject)

8.7.3 Information Type

```
<xs:complexType name="InformationType">
   <xs:sequence>
      <xs:any namespace="##any" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.7.3.1 Description

The Information Type provides a method to represent any type of data element. It could represent a combination of the Biometric Data Type and Biographic Data Type defined in Clause 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this part of ISO/IEC 30108 or by the implementing system.

8.7.3.2 Definitions

The Information Type allows for an unlimited number of data element types, and it does not specify nor require any particular data element.

8.7.4 List Filter Type

```
<xs:complexType name="ListFilterType">
   <xs:sequence>
      <xs:element name="BiometricTypeFilter"
            type="iso-iec19785-3-7:Multiple-types" maxOccurs="unbounded" />
      <xs:element name="IncludeBiometricSubtype" type="xs:boolean" />
   </xs:sequence>
</xs:complexType>
```

8.7.4.1 Description

The List Filter Type provides a method to filter the amount of information returned in a search of biometric data.

8.7.4.2 Definitions

Biometric Type Filter – limits the returned information to a specific type of biometric, as defined by CBEFF

Include Biometric Subtype – a Boolean flag indicating if biometric subtype information should be returned

8.7.5 Option Type

```
<xs:complexType name="OptionType">
   <xs:sequence>
      <xs:element name="Key" type="xs:string" />
      <xs:element name="Value" type="xs:string" minOccurs="0" />
   </xs:sequence>
</xs:complexType>
```

8.7.5.1 Description

BIAS aggregate services support the ability to include various processing options which direct and possibly control the business logic for that service. Together with the Processing Options Type, the Option Type provides a method to represent those options. Processing options should be defined by the implementing system.

8.7.5.2 Definitions

Key – the identifier of an option supported by the implementing system

Value – the value for an option supported by the implementing system

8.7.6 Processing Options Type

```
<xs:complexType name="ProcessingOptionsType">
   <xs:sequence>
      <xs:element name="Option" type="OptionType"
            minOccurs="0" maxOccurs="unbounded" />
   </xs:sequence>
</xs:complexType>
```

8.7.6.1 Description

BIAS aggregate services support the ability to include various processing options which direct and possibly control the business logic for that service. The Process Options Type provides a method to represent those options. Processing options should be defined by the implementing system.

8.7.6.2 Definitions

Option – an option supported by the implementing system

8.7.7 Token Type

```
<xs:complexType name="TokenType">
   <xs:sequence>
      <xs:element name="TokenValue" type="xs:string" />
      <xs:element name="Expiration" type="xs:date" />
   </xs:sequence>
</xs:complexType>
```

8.7.7.1 Description

Some of the BIAS services may be handled asynchronously, such as the ***Identify Subject*** service. Upon receiving a request for the service, the implementing system may either process the request synchronously and return the results directly or it may process the request asynchronously and return a non-zero token that can be used to retrieve the results at some later time. If a token is returned, the client/requester will be responsible for polling for the results, using the token as the input parameter. The Token Type defines the token that is returned for asynchronous processing.

The encoding of the Expiration element shall be the concatenation of the following components (in order):

a) the “year” component, consisting of the year encoded in four digits;

b) the “month” component, consisting of the month encoded in two digits;

c) the “day” component, consisting of the day encoded in two digits;

d) the letter “T”;

e) the “hour” component, consisting of the hour encoded in two digits;

f) the “minute” component, consisting of the minute encoded in two digits;

g) the “second” component, consisting of the second encoded in two digits;

h) the letter “Z”.

8.7.7.2 Definitions

Token Value – a value returned by the implementing system that is used to retrieve the results to a service request at a later time

Expiration – a date and time at which point the token expires and the service results are no longer guaranteed to be available (format: *YYYYMMDD*T*HHMMSS*Z)

9 Error handling and notification

As with any messaging interface, there is a need to define effective measures for handling and propagating error conditions. All BIAS services have a Return parameter for conveying return values and error condition codes.

9.1 Successful service calls

BIAS defines a return value for successful service calls.

```
SUCCESS = 0
```

9.2 Error condition codes

BIAS defines two levels of errors that are represented by the error condition codes: System and Service errors. In addition, BIAS also allows for implementations to define their own error condition codes to represent error conditions that are not already defined by this part of ISO/IEC 30108.

System-level errors are created in the core part of BIAS and 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 are created in the service-dependent part in BIAS and 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. This part of ISO/IEC 30108 will define the error condition codes for system-level errors, since the service-level errors depend on the service mechanisms in which the BIAS is implemented such as the Web services. The service-level errors are specified in other parts of ISO/IEC 30108.

BIAS defines the following set of system-level error codes:

`UNKNOWN_ERROR=1`
The service failed for an unknown reason.

`UNSUPPORTED_CAPABILITY=2`
A requested capability is not supported by the service implementation.

`INVALID_INPUT=3`
The data in a service input parameter is invalid.

`BIR_QUALITY_ERROR=4`
Biometric sample quality is too poor for the service to succeed.

`INVALID_BIR=5`
The input BIR is empty or in an invalid or unrecognised format.

`BIR_SIGNATURE_FAILURE=6`
The service could not validate the signature, if used, on the input BIR.

`BIR_DECRYPTION_FAILURE=7`
The service could not decrypt an encrypted input BIR.

`INVALID_ENCOUNTER_ID=8`

The input encounter ID is empty or in an invalid format.

INVALID_SUBJECT_ID=9
The input subject ID is empty or in an invalid format.

UNKNOWN_SUBJECT=10
The subject referenced by the input subject ID does not exist.

UNKNOWN_GALLERY=11
The gallery referenced by the input gallery ID does not exist.

UNKNOWN_ENCOUNTER=12
The encounter referenced by the input encounter ID does not exist.

UNKNOWN_BIOGRAPHIC_FORMAT=13
The biographic data format is not known or not supported.

UNKNOWN_IDENTITY_CLAIM=14
The identity referenced by the input identity claim does not exist.

INVALID_IDENTITY_CLAIM=15
The identity claim requested is already in use.

NONEXISTENT_DATA=16
The data requested for deletion does not exist.

UNKNOWN_DOCUMENT_CATEGORY=17
The data requested for deletion does not exist.

INVALID_TOKEN=18
The data requested for deletion does not exist.

TOKEN_EXPIRED=19
The data requested for deletion does not exist.

DUPLICATE_ENCOUNTER_ID=20
The input encounter ID for a new encounter already exists for that subject.

IDENTIFICATION_RESULT_NOT_YET_AVAILABLE=21
The result of an asynchronous identification process is not yet available.

UNKNOWN_FORMAT=22
An unknown format was detected.

INVALID_LICENSE=23
An invalid license was found.

SERVICE_NOT_IMPLEMENTED=24
The requested service/function is not implemented.

INVALID_ENCOUNTER_TYPE=25
The input encounter type is not a recognised category.

INVALID_PROCESSING_OPTION=26
The supplied processing options are invalid.

CANNOT_STORE_DATA=27
Data cannot be stored due to an internal reason.

CANNOT_PROCESS_DATA=28
Data cannot be processed.

CANNOT_CREATE_TEMPLATE=29
A biometric template could not be created from the raw input data (either for purpose enrol or verify/identify).

CANNOT_DELETE_DATA=30

Data cannot be deleted.

CANNOT_RETRIEVE_DATA=31
Data cannot be retrieved (read).

CANNOT_INITIALIZE_INTERNAL_MODULES=32
Internal modules cannot be initialized and/or loaded.

CANNOT_VERIFY_DATA=33
Cannot perform a 1:1 verification of the supplied and /or stored data.

CANNOT_IDENTIFY_DATA=34
Cannot perform a 1:N identification of the supplied and/or stored data.

INVALID_CONFIGURATION=35
The (central) system was configured improperly.

CANNOT_CHECK_QUALITY=36
The quality check cannot be performed due to an internal reason.

INTERNAL_DATABASE_ERROR=37
An internal error during the database connection occurred.

CANNOT_UPDATE_DATA=38
Data cannot be updated.

BIOMETRIC_TYPE_NOT_SUPPORTED=39
The biometric modality/type is not supported by the system.

10 Security

Security is important for any kind of distributed computing environment, and even more so when distributed computing occurs over open networks. Web services, for example, generally operate as a public internet Web service, an intranet Web service, or a combination of both. BIAS implementers will ultimately choose the level of security and the security capabilities provided by their system(s). BIAS bindings, which are defined outside of this part of ISO/IEC 30108, need to include security capabilities and provide BIAS implementers with these options.

BIAS bindings shall provide options for message confidentiality and integrity. BIAS bindings shall provide options for message transport confidentiality and integrity. BIAS bindings shall provide options to support access controls in order to authenticate users of a service. BIAS bindings also shall provide options for mutual authentication between clients (requesters) and servers (service providers).

Annex A
(normative)

Conformance requirements

A.1 General

Conformance to this part of ISO/IEC 30108 falls into the following classes:

Class 1: Full Primitive Services Implementation

Class 2: Full Aggregate Services Implementation

Class 3: Limited Primitive Services Implementation

Class 4: Minimum Primitive Services Implementation

Class 5: Minimum Aggregate Services Implementation

Class 6: Matcher Primitive Services Implementation

Class 7: Matcher Aggregate Services Implementation

Conformance requirements for these classes are defined in A.2.

A.2 Class conformance requirements

To claim conformance to this part of ISO/IEC 30108, implementations shall provide the mandatory services and capability item information for their conformance class, as defined below, in accordance with the service definitions in Clause 7.

Implementations shall accept all valid input parameters and return valid outputs, as defined in Clause 7 and Clause 8. Implementations shall perform error handling as defined in Clause 9. Service bindings shall implement the security requirements as defined in Clause 10.

All classes shall implement the ***Query Capabilities*** service.

Table A.1 is a summary of conformance requirements by class. Details are provided following the table.

Table A.1 — BIAS conformance classes

Service/Capability	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Primitive Services							
Add Subject To Gallery	X		X				
Check Quality	X						
Classify Biometric Data	X						
Create Encounter	C**		C**	C**			
Create Subject	X		X	X		X	
Delete Biographic Data	X		X	X			
Delete Biometric Data	X		X	X		X	
Delete Document Data	X		X				
Delete Encounter	C**		C**	C**			

Table A.1 *(continued)*

Service/Capability	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Delete Subject	X		X	X			
Delete Subject From Gallery	X		X				
Get Identify Subject Results	C*		C*			C*	
Identify Subject	X		X			X	
List Biographic Data	X		X	X			
List Biometric Data	X		X	X		X	
List Document Data	X		X				
Perform Fusion	X						
Query Capabilities	X	X	X	X	X	X	X
Retrieve Biographic Data	X		X	X			
Retrieve Biometric Data	X		X	X		X	
Retrieve Document Data	X		X				
Set Biographic Data	X		X	X			
Set Biometric Data	X		X	X		X	
Set Document Data	X		X				
Transform Biometric Data	X						
Update Biographic Data	X		X	X			
Update Biometric Data	X		X	X		X	
Update Document Data	X		X				
Verify Subject	X		X	X		X	
Aggregate Services							
Delete		X					X
Enrol		X			X		X
Get Delete Results		C*					C*
Get Enrol Results		C*			C*		C*
Get Identify Results		C*					C*
Get Update Results		C*			C*		C*
Get Verify Results		C*			C*		C*
Identify		X					X
Retrieve Data		X			X		
Update		X			X		X
Verify		X			X		X
Capability Information Items							
AggregateInputDataOptional		X			X		X
AggregateInputDataRequired		X			X		X
AggregateProcessingOption		X			X		X
AggregateReturnData		X			X		X
AggregateServiceDescription		X			X		X
BiographicDataSet	X	X	X	X	X		
CBEFFPatronFormat	X	X	X	X	X	X	X
ClassificationAlgorithmType	X						
ConformanceClass	X	X	X	X	X	X	X
Gallery	X	X	X				

Table A.1 *(continued)*

Service/Capability	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
IdentityModel	X	X	X	X	X		
ComparisonAlgorithm	X	X	X	X	X	X	X
ComparisonScore	X	X	X	X	X	X	X
QualityAlgorithm	X						
SupportedBiometric	X	X	X	X	X	X	X
TransformOperation	X						
* Conditional - required if associated service is implemented as asynchronously.							
** Conditional - required if encounter-centric model is implemented.							

A.2.1 Class 1: Full Primitive Services Implementation

A Full Primitive Services Implementation shall provide the following BIAS services:

— Add Subject To Gallery

— Check Quality

— Classify Biometric Data

— Create Encounter (conditional – required if encounter-centric model is implemented)

— Create Subject

— Delete Biographic Data

— Delete Biometric Data

— Delete Document Data

— Delete Encounter (conditional – required if encounter-centric model is implemented)

— Delete Subject

— Delete Subject From Gallery

— Get Identify Subject Results(conditional – required if associated service is implemented as asynchronously)

— Identify Subject

— List Biographic Data

— List Biometric Data

— List Document Data

— Perform Fusion

— Query Capabilities

— Retrieve Biographic Data

— Retrieve Biometric Data

— Retrieve Document Data

— Set Biographic Data

— Set Biometric Data

— Set Document Data

— Transform Biometric Data

— Update Biographic Data

— Update Biometric Data

— Update Document Data

— Verify Subject

A Full Primitive Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

— BiographicDataSet

— CBEFFPatronFormat

— ClassificationAlgorithmType

— ConformanceClass

— Gallery

— IdentityModel

— ComparisonAlgorithm

— ComparisonScore

— QualityAlgorithm

— SupportedBiometric

— TransformOperation

A.2.2 Class 2: Full Aggregate Services Implementation

A Full Aggregate Services Implementation shall provide the following BIAS services:

— Delete

— Enrol

— Get Delete Results (conditional – required if associated service is implemented as asynchronously)

— Get Enrol Results (conditional – required if associated service is implemented as asynchronously)

— Get Identify Results (conditional – required if associated service is implemented as asynchronously)

— Get Update Results (conditional – required if associated service is implemented as asynchronously)

— Get Verify Results (conditional – required if associated service is implemented as asynchronously)

— Identify

— Retrieve Data

— Update

— Verify

— Query Capabilities

A Full Aggregate Services Implementation shall provide the following capability information items in response to the ***Query Capabilities*** service:

— AggregateInputDataOptional

— AggregateInputDataRequired

— AggregateProcessingOption

— AggregateReturnData

— AggregateServiceDescription

— BiographicDataSet

— CBEFFPatronFormat

— ConformanceClass

— Gallery

— IdentityModel

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

A.2.3 Class 3: Limited Primitive Services Implementation

A Limited Primitive Services Implementation provides many of the primitive services, except for ***Check Quality***, ***Classify Biometric Data***, ***Perform Fusion***, ***Transform Biometric Data,*** and all document services. A Limited Primitive Services Implementation shall provide the following BIAS services:

— Add Subject To Gallery

— Create Encounter (conditional – required if encounter-centric model is implemented)

— Create Subject

— Delete Biographic Data

— Delete Biometric Data

— Delete Document Data

— Delete Encounter (conditional – required if encounter-centric model is implemented)

— Delete Subject

— Delete Subject From Gallery

— Get Identify Subject Results (conditional – required if associated service is implemented as asynchronously)

— Identify Subject

— List Biographic Data

— List Biometric Data

— List Document Data

— Query Capabilities

— Retrieve Biographic Data

— Retrieve Biometric Data

— Retrieve Document Data

— Set Biographic Data

— Set Biometric Data

— Set Document Data

— Update Biographic Data

— Update Biometric Data

— Update Document Data

— Verify Subject

A Limited Primitive Services Implementation shall provide the following capability information items in response to the ***Query Capabilities*** service:

— BiographicDataSet

— CBEFFPatronFormat

— ConformanceClass

— Gallery

— IdentityModel

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

A.2.4 Class 4: Minimum Primitive Services Implementation

A Minimum Primitive Services Implementation does not provide biometric identification services. A Minimum Primitive Services Implementation shall provide the following BIAS services:

— Create Encounter (conditional – required if encounter-centric model is implemented)

— Create Subject

— Delete Biographic Data

— Delete Biometric Data

— Delete Encounter (conditional – required if encounter-centric model is implemented)

— Delete Subject

— List Biographic Data

— List Biometric Data

— Query Capabilities

— Retrieve Biographic Data

— Retrieve Biometric Data

— Set Biographic Data

— Set Biometric Data

— Update Biographic Data

— Update Biometric Data

— Verify Subject

The optional Gallery ID parameter shall not be used in the above services for Class 4 implementations, as the service for adding a subject to a gallery is not required.

A Minimum Primitive Services Implementation shall provide the following capability information items in response to the ***Query Capabilities*** service:

— BiographicDataSet

— CBEFFPatronFormat

— ConformanceClass

— IdentityModel

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

A.2.5 Class 5: Minimum Aggregate Services Implementation

A Minimum Aggregate Services Implementation does not provide biometric identification services. A Minimum Aggregate Services Implementation shall provide the following BIAS services:

— Enrol

— Get Enrol Results (conditional – required if associated service is implemented as asynchronously)

— Get Update Results (conditional – required if associated service is implemented as asynchronously)

— Get Verify Results (conditional – required if associated service is implemented as asynchronously)

— Retrieve Data

— Update

— Verify

— Query Capabilities

A Minimum Aggregate Services Implementation shall provide the following capability information items in response to the ***Query Capabilities*** service:

— AggregateInputDataOptional

— AggregateInputDataRequired

— AggregateProcessingOption

— AggregateReturnData

— AggregateServiceDescription

— BiographicDataSet

— CBEFFPatronFormat

— ConformanceClass

— IdentityModel

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

A.2.6 Class 6: Matcher Primitive Services Implementation

A Matcher Primitive Services Implementation only provides biometric identification services. A Matcher Primitive Services Implementation shall provide the following BIAS services:

— Create Subject

— Delete Biometric Data

— Get Identify Subject Results (conditional – required if associated service is implementation as asynchronously)

— Identify Subject

— Query Capabilities

— Retrieve Biometric Data

— Set Biometric Data

— Update Biometric Data

— Verify Subject

A Matcher Primitive Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

— CBEFFPatronFormat

— ConformanceClass

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

A.2.7 Class 7: Matcher Aggregate Services Implementation

A Matcher Aggregate Services Implementation only provides biometric identification services. A Matcher Aggregate Services Implementation shall provide the following BIAS services:

— Query Capabilities

— Delete

— Enrol

— Get Delete Results (conditional – required if associated service is implementation as asynchronously)

— Get Enrol Results (conditional – required if associated service is implementation as asynchronously)

— Get Identify Results (conditional – required if associated service is implementation as asynchronously)

— Get Update Results (conditional – required if associated service is implemented as asynchronously)

— Get Verify Results (conditional – required if associated service is implementation as asynchronously)

— Identify

— Update

— Verify

A Matcher Aggregate Services Implementation shall provide the following capability information items in response to the Query Capabilities service:

— AggregateInputDataOptional

— AggregateInputDataRequired

— AggregateProcessingOption

— AggregateReturnData

— AggregateServiceDescription

— CBEFFPatronFormat

— ConformanceClass

— ComparisonAlgorithm

— ComparisonScore

— SupportedBiometric

Annex B
(informative)

Sample biographic data format references

B.1 General

Table B.1 shows the sample biographic data formats. The items in this table may be deleted or updated or new items may be added in the future. The biographic data formats are defined by a country or a third party organization.

Table B.1 — Sample biographic data formats

Biographic data format	*name*		*source*	*type*
FBI EFTS Type-2 (versions 7.x or earlier)	FBI-EFTS		http://www.fbibiospecs.org/	ASCII
FBI EBTS Type-2 (version 8.x or later)	FBI-EBTS		http://www.fbibiospecs.org/	ASCII
DOD EBTS Type-2	DOD-EBTS		http://www.biometrics.dod.mil/	ASCII
Interpol Implementation of ANSI/NIST-ITL	INT-I		http://www.interpol.int/	ASCII
NIEM	NIEM		http://www.niem.gov/	XML
CIQ xNAL	xNAL		http://www.oasis-open.org/	XML
HR-XML	HR-XML		http://www.hr-xml.org/	XML

NOTE 1 If FBI, DOD, or Interpol create an XML version of their specification, the *type* attribute may be set to XML.

NOTE 2 Biographic data format names are registered with the ISO biometric registration authority (www.IBIA.org/cbeff). See the registry website for current complete list.

Annex C
(informative)

Example usage scenarios

C.1 General

The following usage scenarios illustrate the use of BIAS services in various application domains. The following usage scenarios are provided:

Border Management – Benefits

Border Management – Entry

Border Management – Entry with Watch List Hit

Online Banking

Transportation Worker Identification

C.2 Usage scenario: Border management — Benefits

An individual within the United States is applying for immigration benefits for which an in-person interview is required. There is neither a watch list hit nor a derogatory Federal Bureau of Investigation (FBI) Integrated Automated Fingerprint Identification System (IAFIS) record for the Individual.

C.2.1 Basic flow of events

a) Individual applies for immigration status modification.

b) Staff member reviews the individual's status change application and enters the application data.

c) System does not locate a record for the individual based on the information entered (Verify Subject).

d) Individual is sent for fingerprinting.

e) Staff member scans the machine-readable documents.

f) Staff member records and enters the individual's 10 fingerprint images (Create Subject, Set Biographic Data, Set Biometric Data).

g) System retrieves and displays data to staff member (Retrieve Biographic Data).

h) Staff member notifies the individual of interview request.

i) Individual appears for interview.

j) Adjudicator scans the machine-readable documentation.

k) Adjudicator captures the individual's fewer-than-10 verification fingerprint images.

l) System confirms the match of the captured verification prints against the 10-print on file (Verify Subject).

m) System retrieves and displays to the Adjudicator the Individual's application data (Retrieve Biographic Data).

n) Adjudicator makes and records the status decision.

o) System links the decision and current encounter data to the individual's record (Update Biographic Data).

C.2.2 Pre-conditions

— Machine-readable documentation available

— No biometric watch list hit

— No derogatory IAFIS records located

— Individual appearing for interview at a later date

C.2.3 Post-conditions

Individual enrolled into the identity assurance system

Individual's data displayed to the Adjudicator

Status decision made and recorded

C.3 Usage scenario: Border management — Entry

An individual is attempting to enter the United States. There is neither a watch list hit nor a derogatory FBI IAFIS record for the individual.

C.3.1 Basic flow of events

a) Individual arrives at the border.

b) Officer scans the individual's machine-readable documentation.

c) System locates 2-print record in biometric database based on the information entered (Verify Subject).

d) The system retrieves and displays to the Officer the individual's primary view data (Retrieve Biographic Data, Retrieve Biographic Data).

d) Officer captures the individual's 10 fingerprint images.

f) Officer makes and records the entry decision.

g) System links the decision and current encounter data to the individual's record (Set Biographic Data, Set Biometric Data).

C.3.2 Pre-conditions

— Machine-readable documentation available

— 2-print record on file

— No biometric watch list hit

— No derogatory IAFIS records located

— No secondary inspection necessary

C.3.3 Post-conditions

— Individual's data displayed to the officer

— Existing 2-print updated by the newly-captured 10-print

— Entry decision made and recorded

— Current encounter record linked to the individual

C.4 Usage scenario: Border management — Entry with watch list hit

An individual is attempting to enter the United States. There is a watch list hit, but there are no derogatory FBI IAFIS records for the individual.

C.4.1 Basic flow of events

a) Individual arrives at the border.

b) Officer scans the individual's machine-readable documentation.

c) System locates a 2-print watch list record for the individual based on the information entered (Verify).

d) Officer sends the individual to secondary inspection.

e) The system retrieves and displays to the Secondary Officer the individual's secondary view data, including watch list hit (Retrieve Data).

f) Secondary Officer captures the individual's 10 fingerprint images.

g) Secondary Officer makes and records the entry decision.

h) System links the decision and current encounter data to the individual's record (Set Biographic Data, Set Biometric Data).

C.4.2 Pre-conditions

— Machine-readable documentation available

— Biometric watch list hit

— No derogatory IAFIS records located

C.4.3 Post-conditions

— Individual's data displayed to the Officer

— Biometric watch list hit confirmed

— Entry decision made and recorded

— Current encounter record linked to the individual

C.5 Usage scenario: Online banking

An individual has a bank account at XYZ Bank. He would like to access his account information and perform transactions related to his account. The account holder uses his home PC with a biometric device (e.g. an iris camera) installed. In lieu of a password, the bank has configured their online banking web application to use biometric verification.

C.5.1 Basic flow of events

The following describe enrolling an individual's biometric and accessing a bank account.

C.5.1.1 Enrolment

The bank has issued the individual a one-time password to allow him to enrol his biometric into the system. The individual accesses the online banking site and selects 'biometric enrolment'. He enters his account number and one-time password to access this function. Once verified, the enrolment application is initiated. The individual follows the steps to capture his biometric data and to perform a local 1:1 comparison against that data to ensure it will be "matchable". Once suitable data is acquired, it is submitted to the bank as an enrolment [Set Biometric Data]. At this point, the individual's biometric data has been associated with his identity (account).

NOTE Enrolment could also be performed in person at the bank, but a similar scenario would apply, less the one-time password.

C.5.1.2 Account access

Once an individual is biometrically enrolled, he would like to perform an online transaction. He accesses the online banking site and enters his account number. At this point, the individual is challenged to present his biometric (e.g. capture his iris). The individual interacts with the device to capture the biometric data. This data is then transmitted to the bank for verification [Verify Subject]. If the verification is successful, the bank will provide access to the transaction screens for the individual's account.

C.5.2 Pre-conditions

— Individual has already been enrolled and setup with an account

— If the individual has not used his biometric device online before (i.e. with his browser), an active-X control (or Java applet) may have to be downloaded before he can use it with the online banking application. This may be transparent to him (or partially or not at all, depending on his security settings). It is possible that it may not be required at all, depending on what software came with the device and was loaded when it was installed.

C.5.3 Post-conditions

— Individual's biometrics are associated with his account.

C.6 Usage scenario: Transportation worker identification

This use case describes the use of biometrics for the purpose of transportation worker identification. In particular, the use of biometric services in this process are highlighted.

C.6.1 Detailed description

In accordance with the Transportation Security Administration (TSA), the following steps are involved in the issuance of a biometric identity credential:

a) A registered employer (or local facility) initiates a request for a transportation worker identification card (TWIC) to be generated.

b) TWIC applicant completes pre-enrolment and in-person enrolment.

c) Enrolment record request sent to IDMS.

d) 1:N check of Reference Biometric performed.

e) Name-based threat assessment initiated and go, no-go results are returned to IDMS.

f) Authorization to produce TWIC and requisite data sent to Card Production Facility.

g) The user's card is personalised and encoded.

h) Card is securely shipped to the designated Enrolment Center.

i) TWIC Applicant returns to Enrolment Center, validates his identity using the reference biometric, and the card is electronically unlocked and issued.

j) IDMS is notified of TWIC issuance and activation.

k) TWIC holder requests access privileges at transportation facility.

l) Local facility notifies IDMS that access privileges have been granted.

m) Threat/intelligence information is received, and the generated watch list is compared to IDMS.

n) TWIC Hotlist is broadcast to all facilities, as well as a specific notification to any site where privileges have been granted.

o) Revocation and Disposition.

The above, however, does not address the subsequent use of the identity credential for access control purposes. This includes the use for both physical and logical access (i.e. access to secure areas/buildings and to computer/network resources respectively).

C.6.2 Basic flow of events

The following concentrate on those portions of the workflow involving biographic or biometric identity data and operations. Major suboperations (simplified) are defined as:

— Pre-enrolment (optional)

— Enrolment

— Enrolment processing

— Credential issuance

— Privilege granting

— Access control

C.6.2.1 Pre-enrolment

Each transportation worker to be issued a TWIC card may optionally pre-enrol. This involves accessing a web site and entering biographic data. This data is stored for the applicant. [Create Subject, Set Biographic Data] At this time, the applicant may also schedule an enrolment.

C.6.2.2 Enrolment

At the enrolment session, the operator (trusted agent) works with the applicant to:

— Enter (if not pre-enrolled) or verify and edit (if pre-enrolled) biographic data [Create Subject, Set Biographic Data] or [Retrieve Biographic Data, Update Biographic Data]

— Scan/Validate source documents

— Capture ten-prints (fingerprint images) [Set Biometric Data]

— Capture facial photograph [Set Biometric Data]

C.6.2.3 Enrolment processing

Once all of the biographic and biometric data has been collected and sent to the IDMS, enrolment processing is initiated. This consists of:

— 1:N duplicate check against TWIC fingerprint database [Identify Subject, Add Subject to Gallery]

— Watch list (and/or other threat screening) check [Identify Subject (if performed locally)]

— Criminal History Records Check (external interface to FBI IAFIS)

— Name-based checks (external)

C.6.2.4 Credential insurance

If all enrolment processing completes with no adverse information, resulting in an "approval" decision, then the TWIC credential may be issued as follows:

— Card production package is generated [Retrieve Biographic Data, Retrieve Biographic Data]

— Card data is received by the production facility, which produces the card and send it to the enrolment center or origin

— The worker is notified that his/her card is ready for pickup and returns to the enrolment center

— The worker's fingerprint(s) is captured and verified against the IDMS [Verify Subject]

— The fingerprint may also be verified against the fingerprint stored on the card.

— The card is activated.

C.6.2.5 Privilege granting

At each transportation facility to which the worker requires access, privileges must be granted. This involves the following:

The worker presents his/her TWIC card to the facility agent who validates the card (this may involve checking of expiration date, hotlist/CRL, and/or other security features of the card)

— The worker's fingerprint is compared against that stored on the card (local operation)

— Optionally, the worker may be enrolled in a local (operational biometric) [Create Subject, Set Biometric Data]

— The worker is added to the local logical or physical access control system (account created)

C.6.2.6 Access control

The worker presents his/her card to the reader

Biometric is read off the card (if off-card comparison is used)

The worker's biometric is captured and either

Compared against that read off the card, or

Compared against the corresponding record in the (central) access control database [Verify Subject]

C.6.3 Special requirements

Card management functions, though necessary, are not addressed in the use case.

— Revocation is based on a cryptographic CRL and a card hotlist and does not involve biometric operations.

C.6.4 Pre-conditions

— Sponsorship is a requirement prior to enrolment, but is not addressed in the use case.

C.6.5 Post-conditions

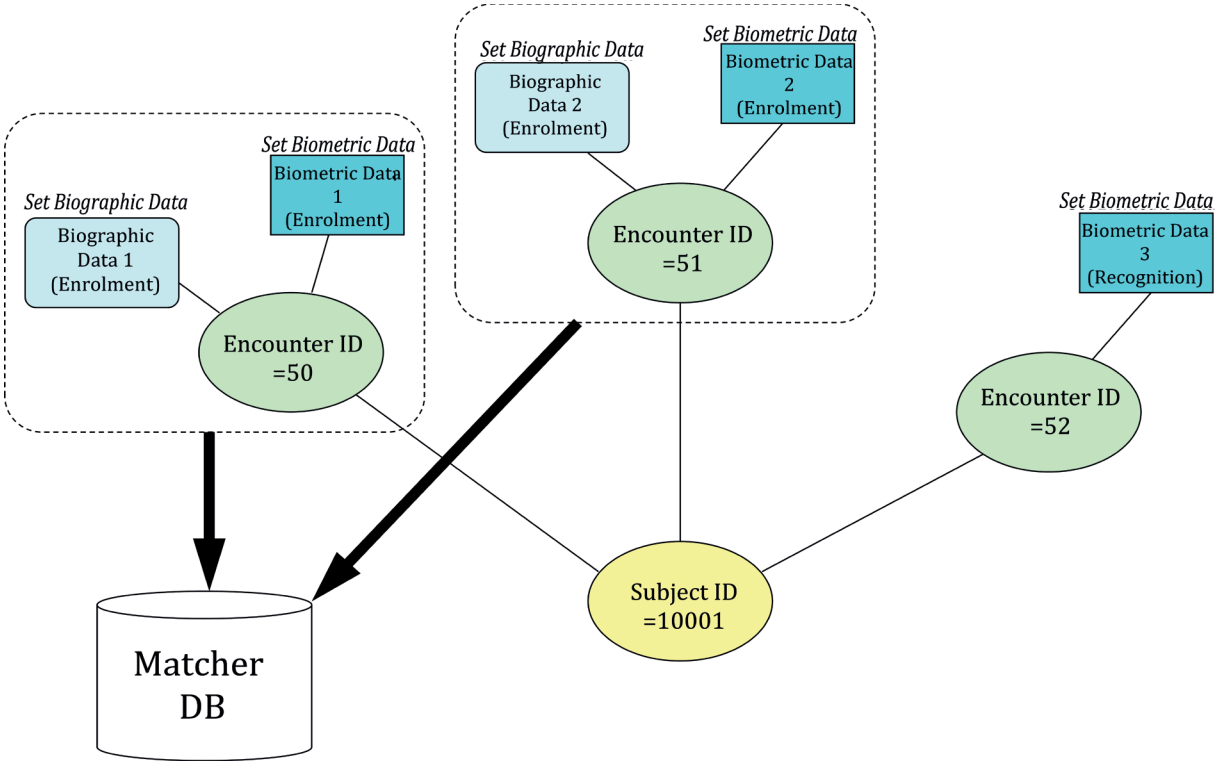
— Periodically, the biometrics of a worker may be queried against one or more threat watch lists; however, this is not depicted in the use case.

Annex D
(informative)

Example encounter scenarios

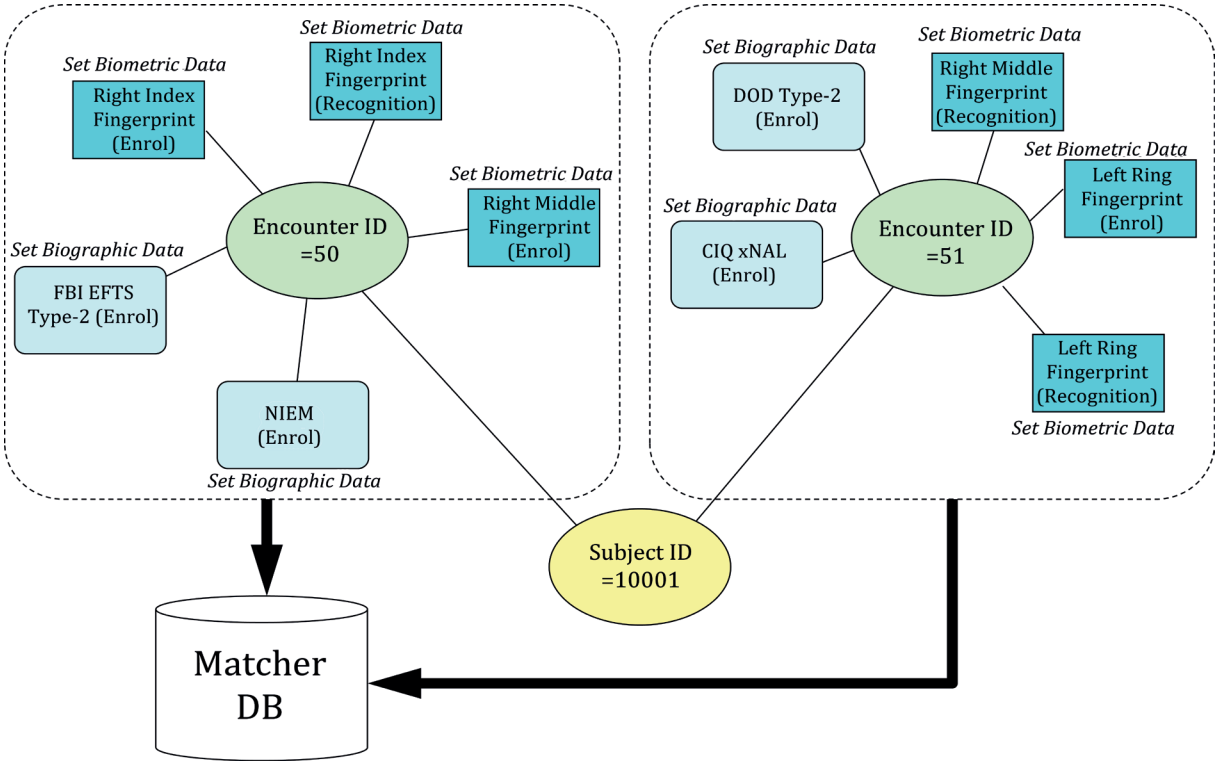
D.1 Typical scenario

The following diagram depicts a scenario in which biometric and biographic data are collected from a subject during two different enrolment encounters and where biometric data is collected from the same subject during a third recognition encounter.



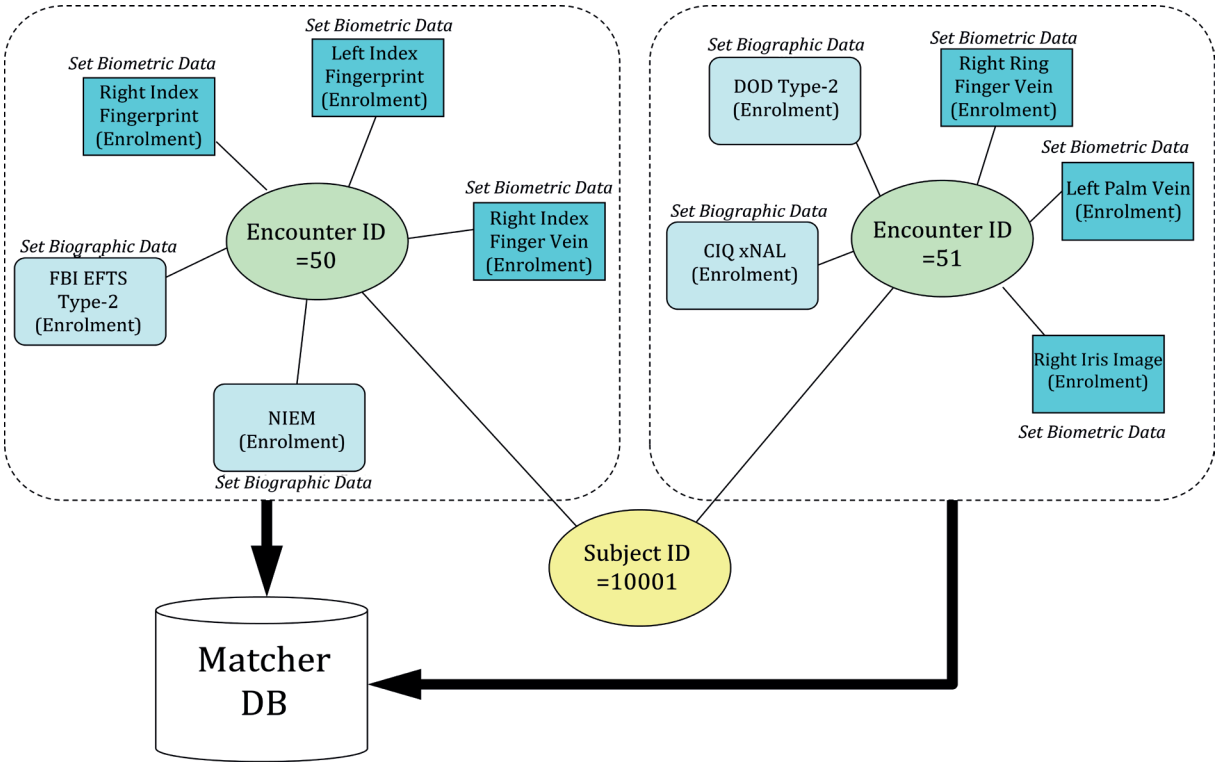
D.2 Possible scenario

The following diagram depicts a two-encounter scenario in which each contain a mixture of enrolment and recognition data.



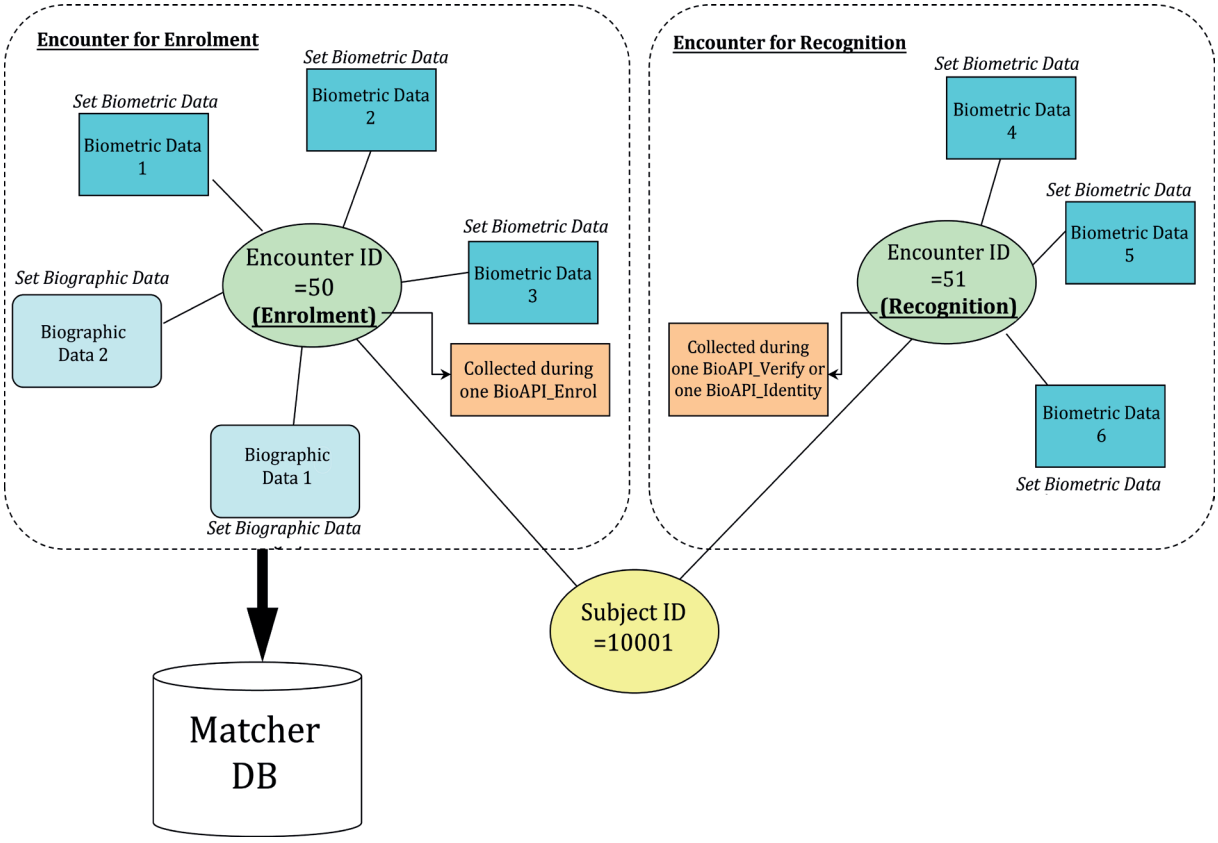
D.3 Multi-biometric scenario

The following diagram depicts a scenario in which two enrolment encounters occur. The first encounter collects biographic and two biometric modalities. The second encounter collects biographic data in different formats, along with biometrics of different subtypes/modalities.



D.4 Separate enrolment/recognition scenario

The following diagram depicts a single enrolment encounter followed by a single (perhaps the first of many) recognition encounter.



Bibliography

[1] ISO/IEC 19784-1:2006, *Information technology — Biometric application programming interface — Part 1: BioAPI specification*

[2] ISO/IEC 19785-2:2006, *Information technology — Common Biometric Exchange Formats Framework — Part 2: Procedures for the operation of the Biometric Registration Authority*

[3] ISO/IEC 24708, *Information technology — Biometrics — BioAPI Interworking Protocol*

[4] JDJ *Service-Oriented Architecture: Beyond Web Services,* JDJ, http://java.sys-con.com/read/44368_p.htm

[5] *OASIS Biometric Identity Assurance Services (BIAS) SOAP Profile, v1.0, May 2012*

ICS 35.040
Price based on 78 pages