

**InterNational Committee for Information Technology Standards (INICTS)  
INCITS Secretariat, Information Technology Industry Council (ITI)  
1250 Eye St. NW, Room 200, Washington, DC 20005  
Telephone 202-737-8888; Fax 202-638-4922  
email: incits@itic.org**

**Document:** M1/06-0432  
**Date:** May 18, 2006  
**Reply to:** Matt Swayze  
**Phone:** 703-984-4004 / 240-418-6195  
**Email:** [matthew.swayze@daon.com](mailto:matthew.swayze@daon.com)

**BIAS Draft Revision 1**

## INCITS Project 1823-D

InterNational Committee for Information Technology Standards (INCITS)  
INCITS Secretariat, Information Technology Industry Council (ITI)  
1250 Eye St. NW, Suite 200, Washington, DC 20005  
Telephone 202-737-8888; Fax 202-638-4922  
email: [incits@itic.org](mailto:incits@itic.org)

**Title:** BIAS  
**Source:** Project Editor  
**Date:** May 18, 2006  
**Revision:** 1

Revision	Date	M1 Document #	Comments
0	February 7, 2006	M1/06-0127	Base document
1	May 18, 2006	M1/06-0432	First draft

**Project Editor:**  
Matt Swayze  
Daon, Inc.  
[matthew.swayze@daon.com](mailto:matthew.swayze@daon.com)  
703-984-4004 / 240-418-6195

**Contents**

**Foreword** ..... **iii**

**Introduction**..... **iv**

**1 Scope** ..... **1**

**2 Conformance**..... **1**

**3 Normative References** ..... **1**

**4 Terms and Definitions** ..... **2**

4.1 Encounter..... 2

4.2 Encounter-Centric ..... 2

4.3 Gallery..... 2

4.4 Identification..... 2

4.5 Identity Assurance ..... 2

4.6 Person-Centric ..... 2

4.7 Subject..... 2

4.8 Verification ..... 3

**5 Symbols and Abbreviated Terms** ..... **3**

**6 System Context**..... **3**

6.1 Service Oriented Architectures ..... 3

6.2 BIAS Architecture..... 5

6.3 BIAS Requirements..... 6

**7 Biometric Identity Assurance Services**..... **7**

7.1 BIAS Interface XML Schema ..... 7

7.2 Primitive Services ..... 8

7.2.1 Add Subject To Gallery ..... 8

7.2.2 Check Background..... 9

7.2.3 Check Quality..... 9

7.2.4 Classify Biometric Data ..... 10

7.2.5 Create Subject ..... 10

7.2.6 Delete Biographic Data ..... 11

7.2.7 Delete Biometric Data ..... 11

7.2.8 Delete Subject..... 12

7.2.9 Delete Subject from Gallery ..... 12

7.2.10 Identify Subject ..... 13

7.2.11 List Biographic Data ..... 14

7.2.12 List Biometric Data ..... 14

7.2.13 Perform Fusion ..... 16

7.2.14 Retrieve Biographic Information ..... 16

7.2.15 Retrieve Biometric Information ..... 16

7.2.16 Set Biographic Data ..... 17

7.2.17 Set Biometric Data ..... 18

7.2.18 Transform Biometric Data..... 19

7.2.19 Update Biographic Data ..... 19

7.2.20 Update Biometric Data ..... 20

7.2.21 Verify Subject ..... 21

7.3 Aggregate Services ..... 22

7.3.1 Enroll ..... 22

7.3.2 Identify ..... 23

7.3.3 Retrieve Information ..... 24

7.3.4 Verify ..... 25

**8 Data Elements and Data Types ..... 26**

8.1 Biographic Data ..... 27

8.1.1 Biographic Data Item Type ..... 27

8.1.2 Biographic Data Type ..... 27

8.2 Biometric Data ..... 28

8.3 Candidate List ..... 28

8.3.1 Candidate Type ..... 28

8.3.2 Candidate List Type ..... 29

8.4 Encounter List ..... 29

8.4.1 Encounter List Type ..... 29

8.5 Identity Model ..... 29

8.5.1 Identity Model Type ..... 29

8.6 Processing Options ..... 30

8.6.1 Processing Options Type ..... 30

**9 Error Handling and Notification ..... 30**

**10 Security ..... 30**

**Annex A: Conformance Requirements..... 32**

**Annex B: Bibliography..... 33**

**Annex C: Example Usage Scenarios ..... 34**

**Annex D: Issues to be Resolved ..... 35**

**Foreword**

INCITS (The InterNational Committee for Information Technology Standards) is the ANSI recognized Standards Development Organization for information technology within the United States of America. Members of INCITS are drawn from Government, Corporations, Academia and other organizations with a material interest in the work of INCITS and its Technical Committees. INCITS does not restrict membership and attracts participants in its technical work from 13 different countries, and operates under the rules of the American National Standards Institute.

In the field of Biometrics, INCITS has established the Technical Committee M1. Standards developed by this Technical Committee have reached consensus throughout the development process and have been thoroughly reviewed through several Public Review processes. In addition, the INCITS Executive Board and the ANSI Board of Standards Review have approved this American National Standard for Publication as an INCITS Standard.

## Introduction

Biometric technologies are being used today in a wide variety of applications and environments. At the same time, enterprises – both commercial and government – have been moving towards services-based architectures as the framework for their enterprise infrastructures. As biometrics become a larger part of the greater identity assurance capability, the need to access these services remotely across those services-oriented frameworks will become necessary.

A current gap exists in standards related to the use of biometric technology in a services oriented architecture (SOA). The Biometric Identity Assurance Services (BIAS) standard is intended to fill that gap by defining a framework for deploying and invoking biometrics-based identity assurance capabilities that can be readily accessed using services (e.g. web services).

Development of this standard necessarily requires expertise in two distinct technology domains – biometrics and service architectures. The two standards organizations that are the leaders in these areas are INCITS and the Organization for the Advancement of Structured Information Standards (OASIS) respectively. The work has been partitioned between the two organizations such that INCITS develops an INCITS standard for biometric services and OASIS develops an OASIS standard for the web services integration. These two standards will be separate but interrelated.

The BIAS standard will help ensure biometric-based solutions are robust and maintainable, while providing a mechanism for accessing an organization's biometric services. BIAS should significantly increase the functional opportunities for implementing identity related functions in a services-oriented framework, allowing for platform and application independence. Presently-developed SOA methods for exchanging information, transactions and security data should provide useful methods, constraints, and patterns for the broader and more robust use of BIAS data. This standard is intended to have the following characteristics:

- Focused on biometrics (though not exclusively)
- Biometric device, type, and vendor independent
- Leverage existing standards where appropriate
- Multi-platform, open
- Primarily focused on remote invocations (services), i.e. not dealing with local devices

## 1 Scope

BIAS defines biometric services used for identity assurance that are invoked over a services-based framework. It is intended to provide 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 project, but will be the subject of separate standards. The first such standard (for a Web services framework) is planned to be developed by OASIS by the BIAS Integration Technical Committee.

Although focused on biometrics, it will necessarily include support for other related identity assurance mechanisms such as biographic and token capabilities. BIAS is intended to be compatible with and used in conjunction with other biometric standards as described in clause 3.

Specification of single-platform biometric functionality (e.g., client-side capture, etc.) is not within the scope of this standard.

Integration of biometric services as part of an authentication service or protocol is not within the scope of this standard; however, it is possible that some of the basic biometric services defined herein may be used by such an implementation in the future.

## 2 Conformance

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

## 3 Normative References

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

- OASIS BIAS Messaging Protocol
- ISO/IEC 19784-1, Information Technology – Biometric Application Programming Interface – Part 1: BioAPI Specification
- ISO/IEC 19785-1, Information Technology – Common Biometric Exchange Formats Framework – Part 1: Data Element Specification

- ISO/IEC 19785-2, Information Technology – Common Biometric Exchange Formats Framework – Part 2: Procedures for the Operation of the Biometric Registration Authority

## **4 Terms and Definitions**

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

### **4.1 Encounter**

An interaction with a subject. Each encounter may contain unique information collected during the encounter and/or describing the encounter.

### **4.2 Encounter-Centric**

A system that supports encounter processing, maintaining a one-to-many relationship between subjects and encounters, and which does not necessarily contain a single, unique set of information for each subject.

### **4.3 Gallery**

A group of individuals, related by a common purpose, designation, or status. For example: a watch list, or a set of individuals entitled to a certain benefit.

### **4.4 Identification**

A biometric system function that performs a one-to-many search, in which a biometric sample(s) from one individual is compared against the biometric references of many individuals to return the identifiers of those with a specified degree of similarity.

### **4.5 Identity Assurance**

The process of establishing, determining, and/or confirming a subject identity.

### **4.6 Person-Centric**

A system that maintains a single, unique view of a subject, and which does not support encounter processing.

### **4.7 Subject**

A person.

## 4.8 Verification

A biometric system function that performs a one-to-one comparison, in which a biometric sample(s) from one individual is compared to biometric reference(s) from one individual to produce a comparison score

## 5 Symbols and Abbreviated Terms

AFIS	Automated Fingerprint Identification System
BIAS	Biometric Identity Assurance Services
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

## 6 System Context

### 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].

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 6-1, below [Alonso].

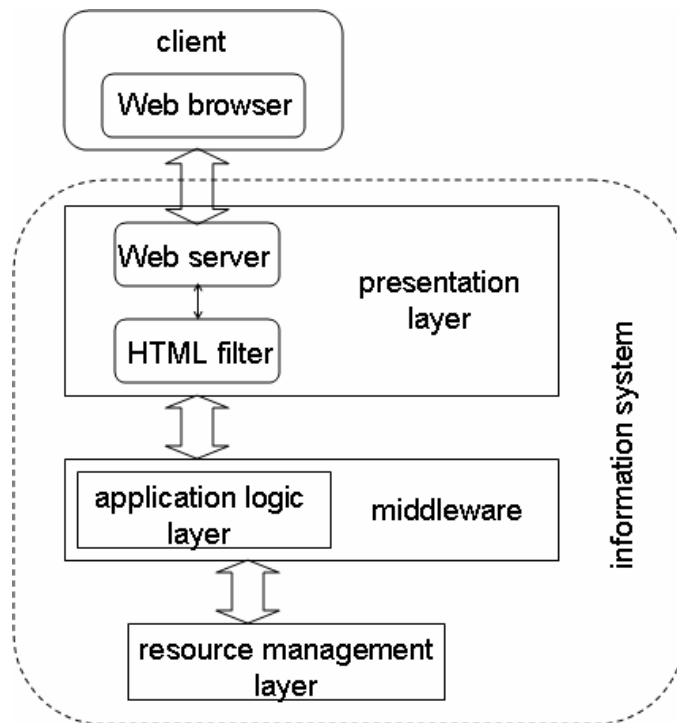


Figure 6-1. Simple N-tier Architecture

In this simple 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 matching server
- A 1:N iris search/match 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 standard 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 and that can be easily assembled to form a collection of autonomous and loosely-coupled business processes. [Papazoglou]

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 utilized 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 interface exposes 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.

Figure 6-2 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 or utilized indirectly in support of a service-provider's own public services.

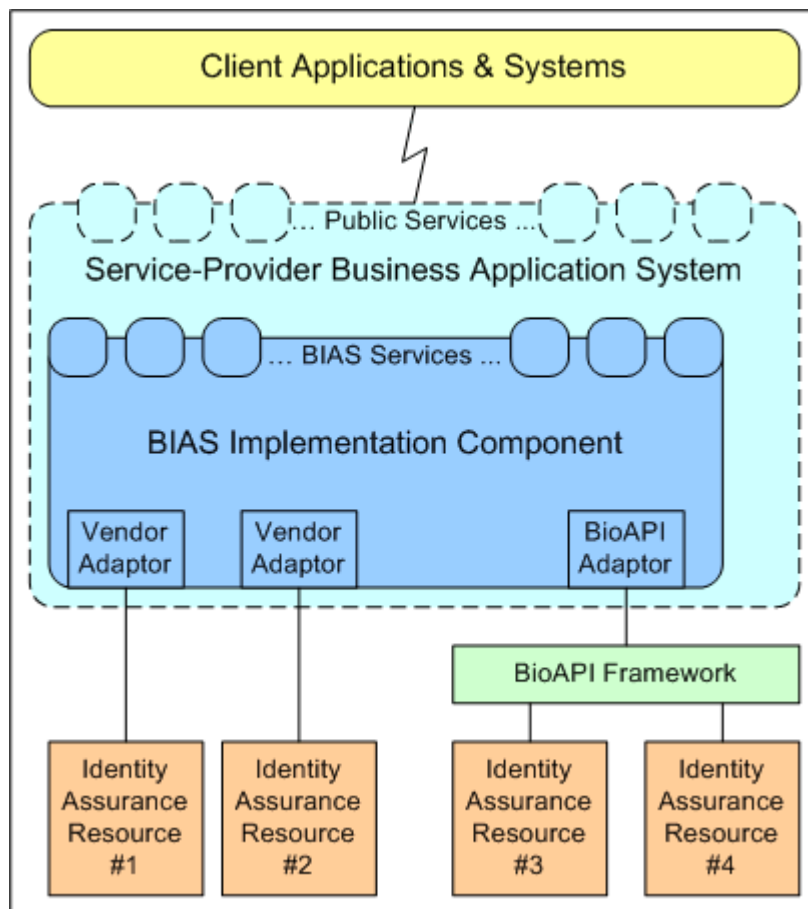


Figure 6-2. BIAS Application Environment

### 6.3 BIAS Requirements

Biometric services, and the applications which use them, particularly in an identity assurance context, imply some unique requirements which are summarized 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 must support both synchronous and asynchronous operations.
- Upon update of a record within a biometric/identity resource, notification of either the owner/originator of that record or of a 3rd party may be required. Therefore, the ability to setup and execute such notifications (initiated from the service side) is needed.
- Some primitive services lend themselves to natural groupings and sequencing – this may justify creation of some ‘aggregate services’ which perform a series of primitive operations based on a single request. (An example of such a grouping would be a negative search in which a 1:N identification which results in a ‘no

match' is immediately followed by the addition of the sample biometric record into that search population.)

- Biometric operations may be singular or multi-biometric.
  - Some 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.
  - Biometric data is in nearly all cases considered personal information and thus privacy protection is always a consideration.
- Before a biometric and/or biographic 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 be shared?
    - For what purpose(s) can the recipient use this data?
    - Who/what authorizes this data to be shared with the recipient for this purpose?
    - How long may the recipient retain the data? (Also state requirements for how data must 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.
- 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.

## 7 Biometric Identity Assurance 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:complexType>
</xs:schema>
```

```

</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 an `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 offers 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 registers a subject to a given gallery or population group. As an optional parameter, the value of the claim to identity by which the subject is known to the gallery may be specified. This claim to identity must be unique. If no claim to identity is specified, the subject ID (assigned with the Create Subject service) will be used as the claim to identity. Additionally, in the encounter-centric model, the encounter ID associated with the subject's biometrics that will be added to the gallery must be specified.

### 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

## 7.2.2 Check Background

TBD – BIAS recognizes the need to provide a capability to search other systems. The format and even the name of this service are still being discussed.

## 7.2.3 Check Quality

```
<interface name="CheckQuality">
  <parameter name="BIR" type="BioAPIBIR" direction="in" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="QualityScore" type="xs:int" direction="out" />
  <parameter name="AlgorithmVendor"
    type="xs:string" direction="out" />
  <parameter name="Algorithm" type="xs:string" direction="out" />
  <parameter name="AlgorithmVersion"
    type="xs:string" direction="out" />
</interface>
```

### 7.2.3.1 Description

The Check Quality service returns a quality score for a given biometric.

### 7.2.3.2 Parameters

*BIR (input)* – data structure containing a single biometric sample for which a quality score is to be determined

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

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

*Quality Score (output)* – the quality of the biometric

*Algorithm Vendor (output)* – the vendor of the quality algorithm used to determine the quality

*Algorithm (output)* – the algorithm used to determine the quality

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

NOTE: It may be possible to create a single "Algorithm ID", consisting of an 'Algorithm Owner' and 'Algorithm Type' which can be assigned in the same manner as a Format ID or Product ID. (For discussion.)

## 7.2.4 Classify Biometric Data

TBD

## 7.2.5 Create Subject

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

### 7.2.5.1 Description

The Create Subject service creates a new subject record and associates a subject ID to that record. The subject ID may be specified by the caller as an option or generated by the service.

### 7.2.5.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

## 7.2.6 Delete Biographic Data

```
<interface name="DeleteBiographicData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <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 erases all of the biographic data associated with a given subject record. In the encounter-centric model the service erases all of the biographic data associated with a given encounter, and therefore the encounter ID must be specified. Deleting data requires that the information is erased completely, preventing the ability to reconstruct a record in whole or in part.

### 7.2.6.2 Parameters

*Subject ID (input)* – the identifier of the subject

*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

## 7.2.7 Delete Biometric Data

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

### 7.2.7.1 Description

The Delete Biometric Data service removes biometric data from a given subject record. In the encounter-centric model, the encounter ID must be specified. Deleting data requires that the information is erased completely, preventing the ability to reconstruct a record in whole or in part.

### 7.2.7.2 Parameters

*Subject ID (input)* – the identifier of the subject

*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

### 7.2.8 Delete Subject

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

#### 7.2.8.1 Description

The Delete Subject service deletes an existing subject record and, in an encounter-centric model, any associated encounter information from the system. This service will also remove the subject from any registered galleries. Deleting a subject requires that the subject information is erased completely, preventing the ability to reconstruct a record or records in whole or in part.

#### 7.2.8.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

### 7.2.9 Delete Subject from Gallery

```
<interface name="DeleteSubjectFromGallery">
  <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="Return"
    type="xs:unsignedLong" direction="out" />
</interface>
```

#### 7.2.9.1 Description

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

### 7.2.9.2 Parameters

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

*Subject ID (input)* – the identifier of the subject

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

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

### 7.2.10 Identify Subject

```
<interface name="IdentifySubject">
  <parameter name="GalleryID" type="xs:string" direction="in" />
  <parameter name="BIR" type="BioAPIBIR" direction="in" />
  <parameter name="MaxListSize" type="xs:int" direction="in" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="CandidateList" type="CandidateListType"
    direction="out" />
</interface>
```

#### 7.2.10.1 Description

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

#### 7.2.10.2 Parameters

*Gallery ID (input)* – the identifier of the gallery or population group which will be searched

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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

*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)* – a rank-ordered list of candidates that have a likelihood of matching the input biometric sample

NOTE: Additional search parameters and controls, and how they are specified, need to be considered. (For discussion.)

## 7.2.11 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="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>
```

### 7.2.11.1 Description

The List Biographic Data service lists the biographic data elements stored for a subject. 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 will return the list of encounter IDs which contain biographic data.

### 7.2.11.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

*Encounter List (output, conditional)* – a list of encounter ID's associated with a subject and which contain biographic data

## 7.2.12 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="ListFilter"
        type="TBD" direction="in" use="optional" />
    <parameter name="Return"
        type="xs:unsignedLong" direction="out" />
    <parameter name="BiometricData" type="TBD"
        direction="out" use="conditional" />
    <parameter name="EncounterList" type="EncounterListType"
        direction="out" use="conditional" />
</interface>

```

### 7.2.12.1 Description

The List Biometric Data service lists the biometric data elements stored for a subject. 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 will return the list of encounter IDs which contain biometric data.

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...), that all biometric information should be listed (e.g. left index finger, right iris, face frontal, etc...), or that only biometric information for a particular biometric type should be listed (e.g. all fingerprints: left slap, right index, etc...).

### 7.2.12.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

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

NOTE: The data type for this parameter will be further specified pending any discussion on use of the BioAPI Patron Format to represent biometric data.

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

*Biometric Data (output, conditional)* - a list of biometric data associated with a subject or encounter

NOTE: The data type for this parameter will be further specified pending any discussion on use of the BioAPI Patron Format to represent biometric data.

*Encounter List (output, conditional)* – a list of encounter ID's associated with a subject and which contain biometric data

### 7.2.13 Perform Fusion

TBD

### 7.2.14 Retrieve Biographic Information

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

#### 7.2.14.1 Description

The Retrieve Biographic Information service retrieves the biographic data associated with a subject ID. In the encounter-centric model, either the encounter ID may be specified or the service will return the information associated with the most recent encounter.

#### 7.2.14.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

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

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

### 7.2.15 Retrieve Biometric Information

```
<interface name="RetrieveBiometricInformation">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="in" use="optional" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="BIR" type="BioAPIBIR" direction="out" />
</interface>
```

#### 7.2.15.1 Description

The Retrieve Biometric Information service retrieves the biometric data associated with a subject ID. In the encounter-centric model, either the encounter ID may be

specified or the service will return the information associated with the most recent encounter.

### 7.2.15.2 Parameters

*Subject ID (input)* – the identifier of the subject

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

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

*BIR (output)* – data structure containing the retrieved biometric sample

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

## 7.2.16 Set Biographic Data

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

### 7.2.16.1 Description

The Set Biographic Data service associates biographic data for a given subject record. An input flag indicates whether the biographic information should replace any existing biographic information (person-centric model) or if a new encounter should be created and associated with the subject (encounter-centric model). For encounter-centric models, the encounter ID may be specified by the caller in order to link biographic and biometric information (assuming biometric information was previously associated using the Set Biometric Data service). If the encounter ID is omitted for the encounter-centric model, the service will return a system-assigned encounter ID.

### 7.2.16.2 Parameters

*Subject ID (input)* – the identifier of the subject

*Identity Model (input)* – indicates a person-centric or encounter-centric model

*Encounter ID (input/output, optional)* – 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

## 7.2.17 Set Biometric Data

```
<interface name="SetBiometricData">
  <parameter name="SubjectID" type="xs:string" direction="in" />
  <parameter name="IdentityModel" type="IdentityModelType"
    Direction="in" />
  <parameter name="EncounterID"
    type="xs:string" direction="inout" use="optional" />
  <parameter name="BIR" type="BioAPIBIR" direction="in" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
</interface>
```

### 7.2.17.1 Description

The Set Biometric Data service associates biometric data for a given subject record. The identity model parameter indicates whether the biometric information should replace any existing biometric information (person-centric model) or if a new encounter should be created and associated with the subject (encounter-centric model). For encounter-centric models, the encounter ID may be specified by the caller in order to link biographic and biometric information (assuming biographic information was previously associated using the Set Biographic Data service). If the encounter ID is omitted for the encounter-centric model, the service will return a system-assigned encounter ID.

Parameters:

*Subject ID (input)* – the identifier of the subject

*Identity Model (input)* – indicates a person-centric or encounter-centric model

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

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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

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

## 7.2.18 Transform Biometric Data

```
<interface name="TransformBiometricData">
  <parameter name="InputBIR" type="BioAPIBIR" direction="in" />
  <parameter name="TransformOperation"
    type="xs:unsignedLong" direction="in" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="OutputBIR" type="BioAPIBIR" direction="out" />
</interface>
```

### 7.2.18.1 Description

The Transform Biometric Data service transforms or processes a given biometric in one format into a new target format. Examples of transformations include:

- Feature Extraction
- Centering or cropping biometric images
- Standard biometric data format conversion
- Etc...

### 7.2.18.2 Parameters

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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

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

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

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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

## 7.2.19 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" />
</interface>
```

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

### 7.2.19.1 Description

The Update Biographic Data service updates the biographic data for an existing subject record. In the encounter-centric model, the encounter ID must be specified.

### 7.2.19.2 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

## 7.2.20 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="BioAPIBIR" direction="in" />
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
</interface>
```

### 7.2.20.1 Description

The Update Biometric Data service updates the biometric data for an existing subject record. In the encounter-centric model, the encounter ID must be specified. In the person-centric model, an input flag indicates if the input biometric data should either replace or be merged with the existing biometric data.

### 7.2.20.2 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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

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

### 7.2.21 Verify Subject

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

#### 7.2.21.1 Description

The Verify Subject service performs a 1:1 verification match between a given biometric and either a claim to identity in a given gallery or another given biometric.

#### 7.2.21.2 Parameters

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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

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

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

*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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

*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)* – the score if the biometric information matched

NOTE: Search parameters and controls, and how they are specified, need to be considered. (For discussion.)

### 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.

#### 7.3.1 Enroll

```
<interface name="Enroll">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="TBD" direction="in">
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="TBD" direction="out">
</interface>
```

##### 7.3.1.1 Description

The Enroll aggregate service adds 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 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 services may be utilized 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 service request in a person-centric model; or (3) initiate a Set Biographic Data and/or Set Biometric Data service request in an encounter-centric model.

### 7.3.1.2 Parameters

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

*Input Data (input)* – contains a subject enrollment record

NOTE: The format for how the Input Data is specified needs to be considered. The Input Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

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

*Return Data (output)* – contains a return data record

NOTE: The format for how the Return Data is specified needs to be considered. The Return Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

## 7.3.2 Identify

```
<interface name="Identify">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="TBD" direction="in">
  <parameter name="Return"
    type="xs:unsignedLong" direction="out" />
  <parameter name="ReturnData" type="TBD" direction="out">
</interface>
```

### 7.3.2.1 Description

The Identify aggregate service performs an identification function according to system requirements and/or resources. For example, a system may have multiple galleries of subjects, and may utilize any or all of these galleries, via calls to the Identify Subject service, to perform a system-level identification function. The system may perform additional actions based on input flags and/or results of the Identify Subject 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 utilize the Set Biographic Data and/or Set Biometric Data services to create a new encounter for the subject.

### 7.3.2.2 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

NOTE: The format for how the Input Data is specified needs to be considered. The Input Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

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

*Return Data (output)* – contains a return data record

NOTE: The format for how the Return Data is specified needs to be considered. The Return Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

NOTE: Additional search parameters and controls, and how they are specified, need to be considered. (For discussion.)

### 7.3.3 Retrieve Information

```
<interface name="RetrieveInformation">
  <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="TBD" direction="out">
</interface>
```

#### 7.3.3.1 Description

The Retrieve Information aggregate service retrieves 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.

#### 7.3.3.2 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

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

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

*Return Data (output)* – contains a return data record

NOTE: The format for how the Return Data is specified needs to be considered. The Return Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

### 7.3.4 Verify

```
<interface name="Verify">
  <parameter name="ProcessingOptions"
    type="ProcessingOptionsType" direction="in">
  <parameter name="InputData" type="TBD" direction="in">
  <parameter name="IdentityClaim"
    type="xs:string" direction="in" use="conditional" />
  <parameter name="ReferenceBIR"
    type="BioAPIBIR" 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:int" direction="out" />
  <parameter name="ReturnData" type="TBD" direction="out">
</interface>
```

#### 7.3.4.1 Description

The Verify aggregate service performs a 1:1 verification function according to system requirements and/or resources. The system may perform additional actions based on input flags and/or results of verification. For example, in an encounter-centric model, this aggregate service may initiate a request to the Verify Subject service, and if a match is found it may then utilize the Set Biographic Data and/or Set Biometric Data services to create a new encounter for the subject.

#### 7.3.4.2 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

NOTE: The format for how the Input Data is specified needs to be considered. The Input Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

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

*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

NOTE: The Biometric Data parameter is proposed to be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

*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)* – the score if the biometric information matched

*Return Data (output)* – contains a return data record

NOTE: The format for how the Return Data is specified needs to be considered. The Return Data could be a combination of the Biometric Data Type and Biographic Data Type defined in Section 8, or it could represent a completely different data exchange model (an EFTS record, for example), either defined/referenced in this standard or by the implementing system. (For discussion.)

NOTE: Search parameters and controls, and how they are specified, need to be considered. (For discussion.)

## 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 section includes information on how this flexibility can be specified and supported by implementing systems.

NOTE: A generic set of data elements needs to be discussed, agreed to, and defined. The services in Clause 7 imply the following data elements/groups: Biographic Data, Biometric Data, Candidate List, Encounter ID, Gallery ID, Identity Claim, Match Decision, Match Score, Reference Biometric, Quality Algorithm, Quality Algorithm Vendor, Quality Algorithm Version, Quality Score, Sample Biometric, Subject ID.

## 8.1 Biographic Data

BIAS defines two data types to provide flexibility for the amount and types of biographic data supported by implementing systems. The Biographic Data Item Type represents a single biographic data item, and the Biographic Data Type represents a set or list of biographic data.

### 8.1.1 Biographic Data Item Type

```
<xs:complexType name="BiographicDataItemType">
  <xs:sequence>
    <xs:element name="name" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="type" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="value" type="xs:string"
      minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

#### 8.1.1.1 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.

#### 8.1.1.2 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")

### 8.1.2 Biographic Data Type

```
<xs:complexType name="BiographicDataType">
  <xs:sequence>
    <xs:element name="biographicDataItem"
      type="BiographicDataItemType"
      minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

#### 8.1.2.1 Description

The Biographic Data Type defines a set of biographic data elements, utilizing the Biographic Data Item Type to represent each element in the set.

### 8.1.2.2 Definitions

*biographic data item* – a single biographic data element

## 8.2 Biometric Data

BIAS proposes that biometric information be based on the BioAPI Patron Format as defined in ISO/IEC 19784-1:2006. Applications and implementations can translate to/from the BioAPI BIR as necessary. (For discussion before the "BioAPIBIR" type is further specified.)

## 8.3 Candidate List

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

### 8.3.1 Candidate Type

```
<xs:complexType name="CandidateType">
  <xs:sequence>
    <xs:element name="score" type="xs:string"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="biographicData" type="BiographicDataType"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="BIR" type="BioAPIBIR"
      minOccurs="1" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="rank" type="xs:int" use="required" />
</xs:complexType>
```

#### 8.3.1.1 Description

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

#### 8.3.1.2 Definitions

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

*score (optional)* – the match score

*biographic data (optional)* – biographic data associated with the candidate match

*BIR* – biometric data associated with the candidate match

### 8.3.2 Candidate List Type

```
<xs:complexType name="CandidateListType">
  <xs:sequence>
    <xs:element name="candidate" type="CandidateType"
      minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

#### 8.3.2.1 Description

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

#### 8.3.2.2 Definitions

*candidate* – a single candidate

## 8.4 Encounter List

### 8.4.1 Encounter List Type

```
<xs:complexType name="EncounterListType">
  <xs:sequence>
    <xs:element name="encounterID" type="xs:string"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

#### 8.4.1.1 Description

The Encounter List Type defines a set of encounters.

#### 8.4.1.2 Definition

*Encounter ID* – the identifier of an encounter

## 8.5 Identity Model

BIAS supports both an encounter-centric and a person-centric identity model.

### 8.5.1 Identity Model Type

```
<xs:simpleType name="IdentityModelType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="encounter" />
    <xs:enumeration value="person" />
  </xs:restriction>
</xs:simpleType>
```

## 8.6 Processing Options

BIAS aggregate services support the ability to include various processing options which direct and possibly control the business logic for that service. Processing options should be defined by the implementing system.

### 8.6.1 Processing Options Type

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

## 9 Error Handling and Notification

Based upon the nature of a web service environment there is a need to define effective measures for error handling and system notifications.

The Session Initiation Protocol (SIP) is a very applicable solution for handling this need for many reasons. At the root of its applicability is the advantage that SIP is a simple protocol that is easily integrated with other common established protocols, mainly the Internetworking Protocol (IP). As it relates to this web service environment, SIP also integrates rather easily with HTTP and other web-based protocols.

*"In large part because SIP borrows heavily from HTTP and other internet standards from the IETF, lots of web-like technologies can be used to build SIP applications. SIP development looks and feels a lot like web development, and there are a lot of web developers out there."*<sup>1</sup>

The robust notification functionality of SIP stems from the ability to produce messages in a wide range of multimedia forms. These methods are already very much common place in industry and government such as email and text messaging. The recent emergence of mobile technologies can provide tremendous avenues in which SIP can be utilized to distribute notifications in addressing the needs of varying systems.

## 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

---

<sup>1</sup> <http://www.sipcenter.com/sip.nsf/html/IMS+IP+Multimedia+Subsystem>

generally operate as a public internet web service, an intranet web service, or a combination of both. The following security concerns need to be addressed in an SOA:

- Message confidentiality and integrity.
- Transport confidentiality and integrity.
- Preventing exceptions from revealing information about the service architecture.
- Protecting the service from spurious messages.

Currently, SSL (Secure Socket Layer) is the most common security scheme used by web services, along with several XML based security initiatives also being pursued. The OASIS Web Services Security (WSS) Technical Committee is working on web services security for higher level security services. The scope of this Technical Committee includes support of security mechanisms that use XML digital signatures and XML encryption to provide SOAP message confidentiality and integrity, and carrying security information for multiple parties<sup>2</sup>. The OASIS Web Services Secure Exchange (WS-SX) Technical Committee is defining extensions to the OASIS WSS to enable trusted SOAP message exchanges involving multiple message exchanges. The OASIS WSS specification describes a base mechanism for securing SOAP messages but does not deal with trust brokering, multi-message exchanges, and policies describing how to secure message exchanges with a web service<sup>3</sup>.

The OASIS Security Services Technical Committee (SSTC) is involved in defining, enhancing, and maintaining a standard XML-based framework for creating and exchanging authentication and authorization information. Currently Security Assertions Markup Language (SAML) is at Version 2.0, which is used by architectures that require interoperable security solutions<sup>4</sup>.

---

<sup>2</sup> More information can be found at <http://www.oasis-open.org/committees/wss/charter.php>

<sup>3</sup> More information can be found at <http://www.oasis-open.org/committees/ws-sx/charter.php>

<sup>4</sup> More information can be found at <http://www.oasis-open.org/committees/security/charter.php>

## Annex A Conformance Requirements

TBD

## Annex B Bibliography

- ISO/IEC FDIS 19784.1 BioAPI Specification – Part 1
- ISO/IEC 2ndWD 19784.1 BioAPI Specification – Part 2, Biometric Archive Function Provider Interface
- ISO/IEC WD 24708, BioAPI Interworking Protocol (BIP)
- <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnpag2/html/wssp.asp>
- <http://www.sipcenter.com/sip.nsf/html/IMS+IP+Multimedia+Subsystem>
- [JDJ] Service-Oriented Architecture: Beyond Web Services, JDJ, [http://java.sys-con.com/read/44368\\_p.htm](http://java.sys-con.com/read/44368_p.htm)
- [Alonso] *Web Services – Concepts, Architectures, and Applications*, Alonso, Casati, Kuno and Machiranj, Springer, 2004

## Annex C Example Usage Scenarios

TBD

## Annex D: Issues to be Resolved (Temporary Annex)

The following list of issues have been identified and still need to be discussed and resolved:

- 1) Definition of the underlying data model
- 2) The ability to monitor services that need to comply with established service level agreements
- 3) Identification and Verification search parameters (e.g. thresholds)
- 4) Asynchronous and synchronous operations
- 5) System notification processing
- 6) The “Check Background” service and integrating with other identity assurance systems
- 7) Determination and/or specification of the “Processing Options” parameter for several of the aggregate services; should they be defined in the standard or by the system implementation?
- 8) Define and detail the remaining primitive services (marked as TBD).
- 9) Provide a context diagram and explanation around concepts of Subjects, Encounters, and Galleries.
- 10) Mechanism for discovering capabilities, supported options, gallery information, etc...