UIDAI

Unique Identification Authority of India
Planning Commission,
Yojana Bhavan,
Sansad Marg,
New Delhi 110001

AADHAAR AUTOMATED BIOMETRIC IDENTIFICATION SUBSYSTEM INTERFACE

AADHAR ABIS Interface – RC – 2

Table of Contents

1. Introduction

The Unique Identification Authority of India (UIDAI) has been created, with the mandate of providing a unique identity to all Indian residents. The UIDAI has published an approach[1] to pursuing this mandate, and plans to use biometrics to eliminate duplicates and ensure uniqueness at enrolment. The AADHAR system is being designed to eventually service the entire population of India, and will involve the biometric identification of 1.2 billion residents. Since the estimated database size (1.2 Billion residents) is an order of magnitude larger than the current largest biometric database (115 Million), the biometric subsystem will have to be constantly monitored for accuracy, scalability and performance. To de-risk the entire program, the system will operate multiple concurrent solutions with the ability to introduce and test newer solutions.

Each Automated Biometric Identification Subsystem (ABIS) must implement an interface that is compliant with this specification. That de-couples the biometric subsystem from the main application logic, and enables a management layer that can orchestrate across the multiple solution providers, continuously measure accuracy, performance and enable better decision making.

Initially, the same interface will be used for enrolment and authentication. At a later time, authentication may be moved out to a separate sub-system for better scaling, and replication. More generally, as the UID database grows in size, the requirements of the biometric system may evolve, and implementers of this interface will have to keep up with them.

1.1 Objective of this document

This is the Release Candidate 1 for this interface. It is expected that the AADHAR server will launch with this version of the interface, and that any biometric solution that is part of the launch will conform to this specification.

The version number for this API is 1.0.

1.2 Interface Overview

The AADHAR servers communicate with the ABIS over message queues. Each request is an XML string, and sent over the message queue. The ABIS is expected to send exactly one response for each request. The messaging subsystem is compatible with the AMQP protocol. There are 3 dedicated message queue pairs (request / response) per ABIS:

1. Administrative
   All administrative requests will be delivered on this message queue and the system will be expected to respond immediately to these requests. Given the

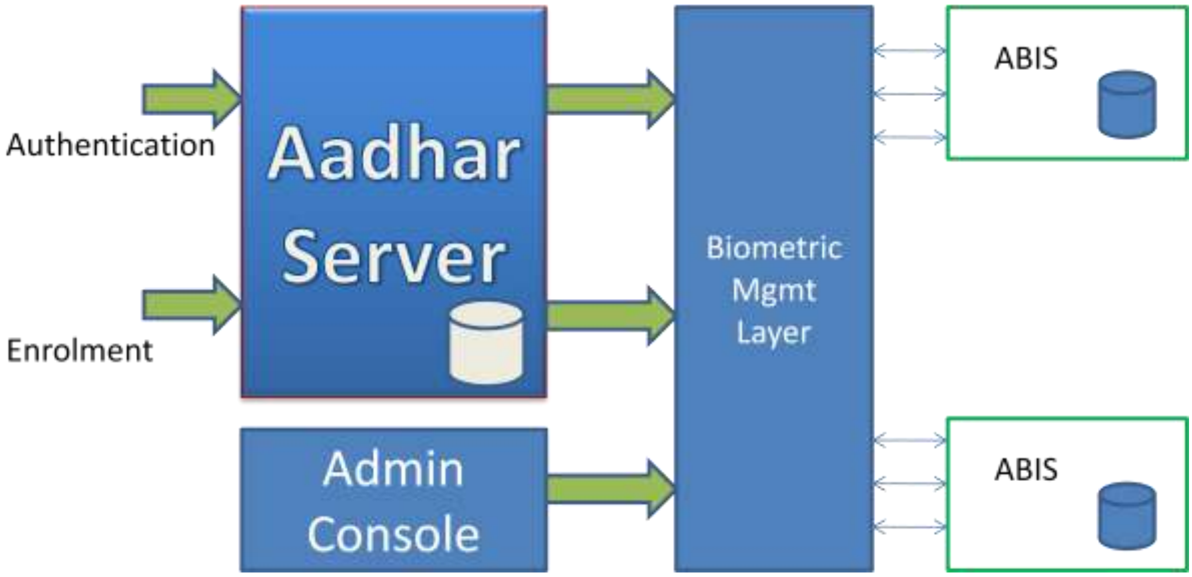
[1] The document may be found at http://www.uidai.gov.in/documents/Creating_a_unique_identity_for_every_resident_in_India.pdf

powerful requests that can be sent on this queue, additional security requirements will be placed on this queue to ensure authorized use.

2. Enrolment
   Most data requests are sent on this queue, including operations to insert, delete, and de-duplicate the residents based on their biometric data.
3. Authentication
   Authentication requests from external clients are sent on this queue. Since authentication is an online process, the ABIS is expected to provide a rapid response to these requests.

SLAs will be specified for each of these queues separately.

Large data objects are not to be sent across these queues – instead a URL is provided in the interface to allow the ABIS to access the data.



1.3 References

UIDAI Publications

Draft approach to providing a unique identity to all Indian residents

  http://www.uidai.gov.in/documents/Creating_a_unique_identity_for_every_resident_in_India.pdf

Demographic Data Standards and Verification Procedure committee Report

  http://www.uidai.gov.in/documents/UID_DDSVP_Committee_Report_v1.0.pdf

Biometric committee report:

  http://www.uidai.gov.in/documents/Biometrics_Standards_Committee%20report.pdf

Standards

The following standards documents are referenced in this document.

  JTC 1/SC 37 Agreed Harmonized Core Biometric Terms and Definitions
  ISO/IEC 19785-3

ISO/IEC 19794-2
ISO/IEC 19794-4
ISO/IEC 19794-6
ISO/IEC 19794-5
ISO/IEC 19795-1

Protocols

AMQP: Version 0.8

1.4 Concurrence, Parallelism, and Transactional Properties

The ABIS must eventually respond to each request. At any given time, many requests may be pending with an ABIS. The solution must provide consistent results, i.e the results from a sequence of operations should be identical with the same sequence of operations carried out in a non-concurrent manner.

For example:
INSERT followed by IDENTIFY

> When an **Insert** request is followed by an **Identify** request, the ABIS must ensure that the data carried by the **Identify** request is compared with the data being added by the **Insert** request.

IDENTIFY followed by INSERT

> When an **Identify** request is followed by an **Insert** request, the ABIS must ensure that the data carried by the **Identify** request is NOT compared with the data being added by the **Insert** request.

The reference database must always be in a consistent state, even in the presence of failures. In the event of a request failing, the ABIS must guarantee that the solution state has not been affected by that request.

2. Glossary

ABIS

Automated Biometric Identification Subsystem.

Candidate List

Each identification operation returns a candidate list, containing zero or more Reference IDs, corresponding to the records that match the provided biometric data.

Gallery

A subset of the population in the reference database used as a target to search for a specific entry. The ABIS interface specifies a dynamic subset for each request. Optional parameter to **Identify,** and **Verify** requests.

galleryUrl

Optional parameter to **Identify** requests. Dereferencing this URL returns a list of referenceIds separated by carriage returns.

Master Database

The database stored within the UID system, which contains all the information related to residents, applications, etc. Specified here to explicitly distinguish it from the Reference Database, this is a part of the ABIS.

maxResults

The maximum number of candidates returned from the identification functions in the candidate list.

referenceId

The reference record is the set of biometric and demographic information originating from a single enrolment. This is stored in the master database, and a view is maintained in the reference database in each ABIS. The reference id is a common identifier which links all of these records – which makes it a surrogate identifier for an enrolment application. Zero or more enrolment applications may be eventually associated with a single UID.

The referenceId is a 128 bit number specified like a UID - as a 36 character ASCII string. For e.g

01234567-89AB-CDEF-0123-456789ABCDEF

referenceUrl

This URL is provided to the ABIS, and used to reference the biometric data associated with an application. The interface provides read-only access to a CBEFF packaged with multiple views of a person's biometric data.
The ABIS must support the following protocols within this URL:

http

The biometric data for enrolment includes multiple types of images, including

- Unsegmented fingerprint images (4-4-2), or
- Unsegmented fingerprint images (4-1-4-1), or
- Unsegmented fingerprint images (2-2-2-2-2), or
- Individual fingerprint images (segmented)
- IRIS images (Left, and Right),
- Face photograph.

The biometric data for authentication could include fingerprint data from 0 or more fingers – either as images, or ISO fingerprint minutiae records, IRIS images, and face images. Fingerprint and IRIS images could be segmented or not.

The following standards are expected to be used for this data:

- CBEFF
  ISO/IEC 19785-3
- Fingerprints:
  ISO/IEC 19794-2
  ISO/IEC 19794-4
- IRIS:
  ISO/IEC 19794-6
- Face:
  ISO/IEC 19794-5

Reference Database

The reference database within the ABIS against which all incoming requests are matched. This database is built up through Insert / Delete requests to the ABIS, and each entry indexed by the reference Id. It is expected that all active ABISs have identical reference databases. The UID server needs to ensure this, even in the presence of failures.

requestId

A requestId is associated with each request sent to the ABIS. This is returned with the response. The ABIS must not use the requestId outside the context of the request.

The requestId is a UUID specified as a 36 character ASCII string. For ex.

01234567-89AB-CDEF-0123-456789ABCDEF

UUID is documented in RFC 4122

requestReference

In addition to the requested, an optional text attribute requestReference may be sent with each ABIS request. If present in the request, this must be returned with the response. The ABIS must not use this attribute for any purpose other than to return it with the response.

The requestReference is a string limited to at most 50 characters in length.

scaledScore

The scaled score returned by the ABIS for identification operations is the achieved false positive identification rate, (or achieved false match rate) based on all available biometric information. It is an integer, and must be calculated as follows:

round (-10 * log$_{10}$ (achieved false positive identification rate))

With this representation

Achieved FPIR or Achieved FMR	scaledScore
1 in 1,000	30
1 in 10,000	40
1 in 100,000	50

targetFMR

Function of Target False Match Rate. This is the rate at which non-mated verification requests are expected to return match decision.

It is represented as integer, and must be calculated as follows:

round (-10 * log$_{10}$ **(<target false match rate>*))***

With this representation

Target False Match Rate	targetFMR
1 in 1,000	30
1 in 10,000	40
1 in 100,000	50

targetFPIR

Function of Target False Positive Identification Rate (FPIR) specified in identification requests. This is the error rate at which open-set identification requests are expected to return non-empty candidate list.

It is represented as integer, and must be calculated as follows:

round (-10 * log$_{10}$ **(<target false positive identification rate>*))***

With this representation

Target False Positive Identification Rate	targetFPIR
1 in 1,000	30
1 in 10,000	40
1 in 100,000	50

3. ABIS Interface Use Cases

The AADHAR servers manage incoming external enrolment and authentication requests along with other internal management requests. These requests are handled through an internal management layer, which isolates the business processes from the actual management of the biometric subsystem. This section documents various possible workflows within the biometric sub-system to provide some context to the usage of the ABIS API. However, it must be noted that these are only for illustration, and that the AADHAR servers could adopt different strategies to meet the evolving business objectives.

3.1 Possible Strategies

The AADHAR server could adopt certain strategies with regards to the management of the biometric data. Other strategies could be adopted with regards to dealing with multiple ABIS. These strategies could change over time based on the constant monitoring and feedback received from the system, as well as with the growth of the reference database.

3.1.1 1 Entry per enrolled resident (per UID issued)

In the event that AADHAR decides to allow only one entry in the ABIS per UID issued, it could use this interface in the following ways:

1. Enrolment
    a. Each enrolment request leads to a new referenceId, and a new entry in the ABIS. (Insert)
    b. On duplicate detection (Identify), the entry is deleted from the ABIS
2. Authentication requests
    a. The incoming data is compared with the single entry in the ABIS through an Identify / Verify request with a Gallery of size 1.
3. Re-enrolment requests
    a. A new entry is inserted, and the old one deleted.

Data Retention

Since only the most recent biometric is maintained in the system, there is no purge of data based on the data retention policies.

Missing biometrics

Since only the most recent biometric data set is maintained in the system, it is possible that the resident may have presented themselves with fewer biometric traits at the time of reenrolment (for ex. They may report with a bandaged finger, or there may be a failure to capture event). This will reduce the resident's ability to authenticate with the missing trait.

3.1.2 1 Entry per Application

In the event that AADHAR decides to maintain multiple entries in the ABIS per UID issued, it could use this interface in the following ways:

- Enrolment
    a. Each enrolment request leads to a new referenceId, and a new entry in the ABIS.
    b. On duplicate detection, the entry is not removed, but is associated with the UID.
- Authentication requests
    a. The incoming data is compared with the multiple entries in the ABIS through an Identify / Verify request with a Gallery that points to the multiple entries for the UID.
- Re-enrolment requests
    a. A new entry is inserted, and the referenceId associated with the UID.

Data Retention

Depending on the data retention policy, older records may have to be purged some time after newer biometric records have been inserted into the ABIS.

Missing biometrics

Since all biometric data set is maintained in the system, the resident will continue to be authenticated against all biometrics, possibly mitigating the impact of a temporary bandaged finger or a failure to capture.

3.1.3 Identify sent to only 1 ABIS at a time

The AADHAR servers could chose to send Inserts to all solutions, and Identify requests to only one ABIS at a time. This would minimize the workload. The ABIS to which the identify request is sent could be determined randomly, or based on the ABIS's demonstrated ability to better deal with certain populations (Children, Elderly, Poor quality biometrics, High quality biometrics).

3.1.4 Identify sent to all ABIS concurrently

The AADHAR servers could chose to send Inserts to all solutions, and Identify requests to all ABIS concurrently. While this could be expensive, it would allow the ABIS systems to be better calibrated for accuracy, and performance. In particular, it could be instructive to analyze the results when the solutions differ in terms of quality scores, or even the ability to detect duplicates. The results of this analysis could be used to decide on future partitioning of Identify requests.

3.1.5 Probes

The AADHAR servers are expected to send probe requests to the various ABIS systems as a part of continuous tuning, monitoring, and testing. Probes labelled with ground truth should be sent to all ABIS for tuning and calibrating the scaledScores. The probes without labels should be sent to all ABIS, and the responses compared for accuracy.

Probes must test for matching data, as well as non-matching data. A probing strategy will be created, which will include targeted gathering of data for use as probes. Certain probes may only use a single biometric, while others may use all the available data.

3.2 Biometric Policies

The UID authority may direct the ABIS suppliers to implement certain policies, which could change from time to time. While these do not have a direct impact on the API specification, they certainly impact the implementation. Examples of these policies include:

- Not using (or using) the 'region' parameter available with the demographic data.
- Ignoring the fingerprint sequencing for comparison when there is missing biometrics.
- Ignoring the sequencing for IRIS images

3.3 Security Guidelines

The ABIS holds biometric data of residents, and will be expected to conform to the UID system's security policies. These policies cover

- Access Control
- Data Security
- Privacy
- Possible Encryption
- Logging, etc.

3.4 Possible Workflows

These workflows are indicative, and provided for a better understanding of the use case of this API. Other uses may be made of the API, and the vendors should not assume only these workflows.

3.4.1 Enrolment

Here is a possible workflow for standard enrolment.

```
For each enrolment request, UID system will
    Generate a referenceId based on the enrolment id.
    Save captured biometric and demographic data – indexed by referenceId.
    Send an Insert request to all ABIS implementations.
    Send an Identify request to some (1 or more) ABIS specifying the requested FPIR.
    <WAIT>
        On completion of all requests, determine if the enrolment data matches existing enrolled residents.

    On low confidence match
        Initiate manual investigation.
        <WAIT>
        Determine if there is a match (Get higher confidence).

    On non-match (empty candidate list)
        Successful enrolment is completed, and an UID is issued.
        Associate Reference ID and corresponding enrolment ID with the UID
    On high confidence match
        Reject the enrolment
        Send a Delete command to all biometric solutions
```

3.4.2 Authentication

Here is a potential biometric based authentication flow.

```
For each authentication request
    Get the all referenceIds based on the UID
    Send a VERIFY request to one or more biometric solutions, specifying the requested FMR, with the referenceIds inline in the Gallery.  The biometric data is also supplied inline.
    <WAIT>
        On completion of all requests, determine the best possible match.

    On  match
        Return Success
    Else
        Return Failure
```

3.4.3 Adding a new Biometric Solution

Here is a potential mechanism to add a new biometric solution.

```
Send a CLEAR request to the new ABIS.
For each UID issued
    For each referenceId associated with the UID
        Get the raw biometric data from storage.
        Send an INSERT request to the ABIS
```

While this process is still running, this ABIS will continue to receive INSERT and DELETE requests from the enrolment process.

Once this process is complete (and the ABIS has caught up with the backlog), it will start receiving **Verify** and **Identify** requests as well.

3.4.4 Error Handling

Each request to the ABIS must receive a single response with a maximum time delay. If the ABIS does not respond within this time, the UID server may send the same request to the same (resend) or to another ABIS.

If any of the requests to the ABIS fail, the UID server must handle the failure cases, including the case where some solutions may succeed while others fail. The options available to the server include:

1. Retrying the request
2. Sending the request to another ABIS
3. Deleting the entire transaction, and rejecting the user's request

4. API Methods and System Interface

This section documents the system interface, and the API calls.

Requests and responses are XML packets delivered to and from the ABIS over respective message queues. Each request results in one and only one response from the ABIS. Each request, contains a requestId, the version of the API, and a timestamp. The timestamp is an integer, representing the number of milliseconds since Jan 1, 1970 0:00 UTC. Since the API is asynchronous, the requestId is used to connect requests with the appropriate response. The biometric solution must not use requestId outside the scope of a request. Each response carries the requestId from the original request, and the timestamp at which the request was completed. A request may contain an optional attribute requestReference. The biometric solution must not use the requestReference in any way, except for returning it with the response.

4.1 Message Queues

The following table shows the list of requests that can be sent on each of the 3 queues.

Management	**Enrolment**	**Authentication**
Shutdown	Insert	Verify
Clear	Identify	Ping
Ping	Verify	
GetPendingRequests	Delete	
GetReferenceCount	Ping	
Configure		

4.2 Request Parameters

The following parameters are common to the requests, and responses. The types are documented here to avoid duplication, and potential discrepancies.

Parameter Types

requestId	UUID – 128 bit integer represented as a 36 char string
referenceId	128 bit integer represented as a 36 char string
referenceUrl	String. Protocol constrained to http .
Return value	Integer
failureReason	Integer
scaledScore	Integer
targetFMR	Integer
targetFPIR	Integer
maxResults	Integer

4.3 Return Codes

The following return codes are standard for all the responses within the ABIS interface.

0	not used
1	Success
2	Failed

4.4 Failure Reasons

The following failure reasons are standard for all the responses within the ABIS interface. Failure reason is only required when the request fails.

0	Success – not used
1	Internal Error – Unknown
2	Aborted
3	Invalid Request –XML Error
4	Invalid Request –Parameter value invalid
5	Invalid Request – referenceId already in use
6	Invalid Request – referenceId not in use
7	Unexpected Error – Unable to access Biometric Data
8	Unable to perform request

4.5 Solution Discovery

The UID server and biometric solutions discover each other through configuration files, which contain the queue names that they must subscribe to, and communicate with.
The exact configuration details are TBD.
All 3 communication queue pairs per ABIS are available in the configuration files.

4.6 API Methods

The following methods must be implemented by the ABIS provider.

4.6.1 Insert

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Insert referenceId=”” referenceUrl=””>
  <Attribute name=”” value=””/>
  </Insert>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
</Response>
```

Behavior

The ABIS must extract the biometric, data from the referenceUrl, process the data, and store it locally within the ABIS. Demographic data corresponding to the person is passed as name value pairs on this interface. The referenceId provided by the UID server is used to refer to each entry in the database uniquely.

The referenceId must not be active prior to this operation.

Processing the biometric data may involve segmenting images, and extracting features / templates / minutae from these segmented images.

De-duplication will not be performed in this operation.

The referenceUrl provides access to a CBEFF packaged biometric dataset, described later in this document.

The following demographic attributes may be sent to the ABIS:

Year of Capture
Year of Birth
Sex
Region Code

Use of these attributes is subject to policies specified by the UID. However, there is no support for static partitioning of the residents based on these attributes.

This method is used to add a resident’s data to the ABIS during enrolment. It is also used to initialize the reference database while adding a new provider.

4.6.2 Identify

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Identify referenceId=”” referenceUrl=””
            maxResults=”” targetFPIR=”.”>
    <Gallery url=””> <!-- Optional -->
      <referenceId id=””/>
      [...]
    </Gallery>
  </Identify>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
  <CandidateList count=”” more=”true|false”>
    <Candidate referenceId=”” scaledScore=””/>
    [...]
  </CandidateList>
</Response>
```

Expected behaviour:

The Identify request is delivered on the Enrolment queue. An Identify request provides a 1:n comparision, comparing a matched set of biometrics with a collection.

The matched set could be specified by one of the following attributes:

1. referenceId
   Here the biometrics to be compared are already in the reference database in the ABIS from a previous Insert request.

2. referenceUrl
   The biometrics packet is obtained by dereferencing this URL.

Exactly one of referenceId or referenceUrl must be provided.

This data is to be matched against biometric data already in the ABIS database. This collection of data is specified by:

1. Default
   If no subset is specified, the data is matched against all entries in the reference database in the ABIS.
2. Gallery
   A small collection of referenceIds are specified inline in the request. There is no support for static, pre-defined Galleries in the UID system.
3. galleryUrl
   A collection of referenceIds are obtained by dereferencing the **url** attribute.

If the Gallery is specified, the Gallery element must specify either the url, or a list of referenceIds, but not both. At least 1 referenceId must be provided. All referenceIds provided must be valid and present in the reference database.

Candidates that match the input condition with an achieved FPIR better than the targetFPIR are returned. The maximum number of results returned is specified in the request (maxResults). If more results may be available, the ABIS should indicate that in the response. The ‘best match’ may or may not be a part of these results. The scaled score, which represents the achieved FPIR is returned for each candidate.

The Identify request must match with all Insert requests that are ahead of it in the queue excluding itself, and not with any requests that are after it in the queue.

Possible Uses:

a. Identify with a referenceId, and no Gallery specifications is expected to be the primary method of ensure uniqueness of residents while issuing a UID.
b. Identify with a referenceUrl, and a small Gallery is expected to be used to authenticate individuals as part of an offline process. For instance enrolment client operators, whose biometrics are sent along with enrolment data may be authenticated in this manner. This is also expected to be used for other use cases such as retrieving resident UIDs, where a smaller list can be created on the basis of demographic data.
c. Identify with a referenceUrl (with or without a Gallery) is expected to be used for investigative workflows, probes, and benchmarking.

4.6.3 Verify

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Verify referenceUrl=”” maxResults=”” targetFMR=””>
    <Biometrics> <!-- Optional -->
      <Biometric type=”FMR|FIR|IIR”>
        Encoded (base64) biometric
      </Biometric>
      [...]
    </Biometrics>
    <Gallery>
      <referenceId id=””/>
      [...]
    </Gallery>
  </Verify>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
  <CandidateList count=”” more=””>
    <Candidate referenceId=”” scaledScore=””/>
    [...]
  </CandidateList>
</Response>
```

Expected behaviour:
The Verify request is delivered on the Authentication queue. This is similar to the Identify request, and provides for a 1:few comparison, comparing a matched set of biometrics with a collection.
The matched set could be specified by exactly one of the following attributes:

1. Inline biometrics
   The biometric data captured at the authentication device is delivered inline to the ABIS. Multiple biometric elements may be included in this request, all of them presumed to be of a single person. Each element contains only a single view..
   a. **type=FMR**: The biometric data is of type Fingerprint Minutae Record. This data is in ISO minutae format with no proprietary extensions allowed.
   b. **type=FIR**: The biometric data is of type Fingerprint Image Record. The data is a fingerprint image packaged in ISO 19794-4 format, which could contain a be compressed (or uncompressed) image, of type PNG, WSQ, or Jpeg200.
   c. **type=IIR**: The biometric data is of type Iris Image Record. The data is an Iris image packaged in ISO 19794-6 format, which could contain a compressed (or uncompressed) image, which could be of type PNG, or Jpeg200.
2. referenceUrl
   The biometrics packet is obtained by dereferencing this URL.

Biometric data from one subject MUST be provided, either inline, or through referenceUrl, but not both.

This data is to be matched against biometric data already in the ABIS database. This collection of data is specified by the Gallery, which specifies a small collection of referenceIds inline in the request. The Gallery must contain at least one referenceId, and all referenceIds provided inline in the XML request must be valid and present in the reference database.

Candidates that match the input condition with an achieved FMR better than the targetFMR are returned. The maximum number of results returned is specified in the request (maxResults). If more results may be available, the ABIS should indicate that in the response. The 'best match' may or may not be a part of these results. The scaled score, which represents the achieved FMR is returned for each candidate.

Possible Uses:

> Depending on the strategy, the Gallery could be of size 1 or a small number. These entries are assumed to be from the same individual, and the ABIS is used to verify the identity of this person.
> Since this is used as a part of an online process, stringent SLAs can be expected for this request.

4.6.4 Delete

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Delete referenceId=””/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
</Response>
```

Expected behaviour:
Remove the specific entry from the reference database.

Expected usage:
This could be used to remove duplicates found as a result of Identify. Other use cases depend on the overall strategy of the UID servers (1 entry per UID or 1 per application).

4.6.5 Ping

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Null/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
</Response>
```

Always returns success.
This request may be sent on any of the 3 queues, and is used to test connectivity / liveness of the system.

4.6.6 GetPendingJobs

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <GetPendingJobs/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
  <Jobs count=””/>
</Response>
```

Always returns success.
This request is sent on the management queue, and is used to report the number of pending requests within the ABIS – i.e. those requests which have been received by the ABIS, but a response for which has not yet been sent. This data is used for monitoring, and analytics.

4.6.7 Shutdown

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Shutdown/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
  <Jobs count=””/>
</Response>
```

When this request is received, the ABIS must stop accepting requests from the enrolment and authentication queues. All requests that have already been received must be responded to – either by successful completion, or by responding to them with errors. The number of requests that were pending, and had to be aborted is returned along with the response.

The database is expected to stay consistent – i.e. any failed requests must guarantee that the system has not been modified.

4.6.8 GetReferenceCount

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <GetReferenceCount/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” failureReason=”” count=””/>
</Response>
```

When this request is received, the ABIS must respond with the size of the Reference Database.
This could be used as a simple check while confirming that all ABIS systems are in sync.

4.6.9 Configure

The ABIS is configured through files at this time. However, in the future an API method may be provided to allow more dynamic configuration.

4.6.10 Clear

```
<Request requestId=”” version=”1.0” timeStamp=””>
  <Clear/>
</Request>
```

```
<Response requestId=”” timeStamp=””>
  <Return value=”” [failureReason=””]/>
</Response>
```

When this request is received on the management queue, the ABIS must stop all current processing, respond to the pending requests, and delete all entries that were inserted previously. When that is complete, and the response sent – the ABIS may continue processing requests from the message queues.

This method is expected to be used as a part of the process to initialize (re-initialize) an ABIS.

4.7 Data Format

All biometric and demographic data is supplied to the ABIS through a URL. Dereferencing this URL provides a CBEFF packet that contains

- Unsegmented fingerprint images (4-4-2), or
- Unsegmented fingerprint images (4-1-4-1), or
- Unsegmented fingerprint images (2-2-2-2-2), or
- Segmented fingerprint images
- IRIS images (Left, and Right),
- Face photograph.

Multiple views may be available for each biometric trait. For Insert requests, all segmented / unsegmented images are constrained to specify the positions that correspond to each image[2].

At this time, for Insert requests, all images are constrained to be either uncompressed or lossless PNG images. Verify requests deal with fingerprints in ISO minutae format with no proprietary extensions, or images that are constrained to be the following formats:

Fingerprint	Uncompressed, PNG, WSQ or JPEG 2000
Iris	Uncompressed, PNG, JPEG2000
Face	Uncompressed, PNG, JPEG, or JPEG2000

[2] Unknown positions are not allowed for enrolment. However, they may be allowed for authentication.

The CBEFF packet is based on ISO/IEC 19785-3, with Patron ISO/IEC JTC 1 SC 37-Biometrics, Patron Identifier 257, Patron Format Identifier 7 – XML patron format.

Some key aspects of the XML Patron format

- This patron format is based on W3C XML 1.0. It supports all the mandatory and optional data elements specified in ISO/IEC 19785-1. It can support either a simple BIR or a complex BIR structure where each intermediate node or leaf of the structure is itself a BIR (called a "child BIR").
- An instance of a BIR or child BIR contains either a BDB or one or more BIR children, but never contains both.
- If any <bir> element in a hierarchy of <bir> elements specifies an abstract value for a given data element, that abstract value can be overridden by a different abstract value in any of its descendant <bir> elements
- The <bdb-info> element (if present) carries information about either the BDB of the BIR (if the <bir> element has a child <bdb> element) or about the BDBs of the descendant BIRs that have a child <bdb> element (if the <bir> element has one or more child <bir> elements).

4.7.1 Illustrative CBEFF Format

Applying this format for the purposes of this API, the xml structure would be as follows.

```
<?xml version='1.0' encoding="utf-8"?>
<bir xmlns="urn:oid:1.1.19785.0.257.1.7.0">
  <version major="1" minor="0"/>
  <cbeff-version major="2" minor="0"/>
  <bir-info  integrity="false" />
  <bir>
    <!-- left iris -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="9" type="iris"
               subtype="left" creation-date="20100308T113005Z" />
    <bdb>
      SUlSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- right iris -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="9" type="iris"
               subtype="right" creation-date="20100308T113025Z" />
    <bdb>
      SUlSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- first attempt of the right slap -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="7" type="finger"
```

```
               subtype="right pointer finger middle-finger ring-
finger little-finger"
               creation-date="20100308T113115Z" />
    <bdb>
      RklSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- second attempt of the right slap -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="7" type="finger"
               subtype="right pointer-finger middle-finger ring-
finger little-finger"
               creation-date="20100308T113145Z" />
    <bdb>
      RklSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- two thumbs -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="7" type="finger"
               subtype="left right thumb"
               creation-date="20100308T113235Z" />
    <bdb>
      RklSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- left slap -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="7" type="finger"
               subtype="left pointer-finger middle-finger ring-
finger little-finger"
               creation-date="20100308T113355Z" />
    <bdb>
      RklSADEwMQA...=
    </bdb>
  </bir>
  <bir>
    <!-- face -->
    <bir-info  integrity="false" />
    <bdb-info  format-owner="257" format-type="8" type="face"
               creation-date="20100308T113145Z" />
    <bdb>
      RkFDADEwMQA...=
    </bdb>
  </bir>
</bir>
```