Republic of the Philippines
PHILIPPINE STATISTICS AUTHORITY

Reference No. 2019SBAC-05-017

16 May 2019

SUBJECT: **Supplemental Bid Bulletin No. 5**

Dear Prospective Bidder:

This serves as the official transmittal of Bid Bulletin No. 5 for the Supply, Delivery and Managed Services of Registration Kits for Philippine Identification System (PhilSys).

This Supplemental/Bid Bulletin No. 5 shall form part of the Bid Bulletin No. 4 as attachment Annex B. Any provision in the Bidding Documents inconsistent herewith is hereby amended, modified and superseded accordingly.

The bulletin will be posted on the websites of PhilGEPS and PSA.

Truly yours,

CANDIDO J. ASTROLOGO JR.
Assistant National Statistician, Standards Service
Chairperson, Special Bids and Awards Committee (SBAC)

EU/RCA/NGP/JPC

Annex B: Biometric Functions API

Biometric Functions API

MOSIP uses biometrics in the registration and the authentication processes. As part of this it requires specialized processing of the biometrics data for different types of biometrics. This document defines the interface for the Java Library providing the functional support for processing biometrics.

Function	Quality Check
Use Cases	1) When a biometric image is received by MOSIP in the registration client using a forced capture, this method is used to check the quality of the image. 2) Server side validation of quality of biometric images uses this method 3) When external biometric images are received to be put on record this method is used to determine the quality of the received biometric image
Signature	Input Parameters Biometric Image in "Biometric Image Record" format. This could be FIR, IIR etc. Control Flags is an optional list of name value pairs that can be used to configure the behavior of the library. Output Parameters QualityScore object with Quality score and Analytics. The quality score is on a scale of 100 - Higher is better. Errors / Exceptions Unsupported biometric type Unsupported image format Processing error
Behavior	Fingerprint • The biometric image record is a Fingerprint Image Record. The FIR structure is explained in a later section • The image is a jpeg2000 format lossless image • The quality score will be using NFIQ2 for 500 dpi images and NFIQ for other densities • The analytics data returned can have information on finger index, liveness, etc. IRIS • The biometric image record is a Iris Image Record. The IIR structure is explained in a later section • The image is a jpeg2000 format lossless image • The quality score will be on a scale of 100 and will factor focus, blur, eyelid position etc.

	• The analytics data returned can have information on eye index, eyelid position, iris obscuration, gaze angle etc. Face • The biometric image record is a Face Image Record. The FaceIR structure is explained in a later section • The image is a jpeg2000 format lossless image • The quality score will be on a scale of 100 and will factor ICAO standards • The analytics data returned can have information on tilt, missing landmarks, lighting etc.
Notes	For new biometrics that come up the quality check could be different. The quality score could be a single or composite score in the future.

Function	Matcher
Use Cases	1) Used for matching a biometric received in an auth transaction with the biometrics on record 2) Used for authentication of operators in offline mode 3) Used for prevention of erroneous submission of operator biometrics in place of registrant's biometric on registration client
Signature	Input Parameters Sample Input Image Record (1) - This is a Biometric Image Record with metadata and image data. This is the freshly received input which needs to be matched. Match List of Image Records (n) - This is the set of biometrics on record that the input images needs to be matched against. The smaller this list the better the performance. Also there will be outer limits to the size of this list based on the library used. Control Flags is an optional list of name value pairs that can be used to configure the behavior of the library. Output Parameters List of MatchScore object with Match score and Analytics. The match score is on a scale of 100 - Higher is better. Errors / Exceptions Unsupported biometric type Unsupported image format Mismatch in biometric types (sample to record) Mixed biometric types (mix of types in the on records list) Processing error
Behavior	Fingerprint • The biometric record will have a jpeg2000 format lossless image or

	minutiae in an ISO template (FMR). • The on record biometrics will be jpeg200 format lossless images or biometric extracts tagged to a specific extractor. • Best matches are provided for an image to image match. The sample and on records data are both images. The matcher uses its own extraction algorithm on the images to be compared. • In case the sample is not an image but minutiae (FMR) then the match accuracy and efficacy might be lower as the extraction templates and algorithms might not be the same • In case the on record extract is a format that the matcher does not understand the match will not happen. Also if the comparison is between two extracts (Minutiae) the False Rejection and False Acceptance cases might be high • A typical 1 : n match is used when comparing one finger received to a set of fingers (upto 10) on record for that person or a limited set of people (upto 10 per person). • The match score will return a score specific to the library as well as a confidence score on a scale of 100. IRIS • Input and on record images are jpeg2000 format lossless images • The matcher uses its own algorithm to perform extraction, segmentation and identifying patterns on the images being compared. • Typical comparisons are 1 : 2 for a person and 1 : n for multiple people. (deduplication scenario) • The match score will return a score specific to the library as well as a confidence score on a scale of 100. Face • Input and on record images are jpeg2000 format lossless images • The matcher uses its own algorithm to perform extraction, segmentation and identifying landmarks on the images being compared. • Typical comparisons are 1 : 1 for one person and 1 : n for multiple people (deduplication scenario) • The match score will return a score specific to the library as well as a confidence score on a scale of 100.
Notes	As new biometrics come up the matching score will be as per the sample set and standards defined for that biometric.

Function	Composite Matcher
Use Cases	1) Used for matching multiple modes of biometrics of an individual
Signature	Input Parameters

	Sample List of Input Image Records (1 set) - This is a Biometric Image Record list with metadata and image data belonging to an individual. This is the freshly received input which needs to be matched. Match List of Image Records (1 set) - This is the set of biometrics on record that the input images needs to be matched against belonging to an individual. Control Flags is an optional list of name value pairs that can be used to configure the behavior of the library. Output Parameters A MatchScore object with Match score and Analytics. The match score is on a scale of 100 - Higher is better. Errors / Exceptions Unsupported biometric type Unsupported image format Mismatch in biometric types (sample to record) Processing error
Behavior	Multiple Fingerprints • The biometric record will have a jpeg2000 format lossless image or minutiae in an ISO template (FMR). • There will be multiple fingers in the input based on which the composite match has to be done. • A 1 : 1 composite match is done when comparing one set of fingers received to a set of fingers on record for that person. • The match score will return a score specific to the library as well as a confidence score on a scale of 100. Multiple Biometric Types • Input and on record images are jpeg2000 format lossless images • The data consists of mixed biometric type like Iris + Fingerprint etc. The data on record should have the biometric elements being passed in for a match to happen. • The match is a 1 : 1 composite match. • The match score will return a score specific to the library as well as a confidence score on a scale of 100.
Notes	As new biometrics come up the matching score will be as per the sample set and standards defined for that biometric.

Function	Extractor
Use Cases	1) Used to extract salient features and patterns of a biometric to use in fast comparison 2) In case of fingerprints this is called Minutiae and a standard representation of minutiae is an ISO template for FMR

Annex B: Biometric Functions API

Signature	Input Parameters Biometric Image in "Biometric Image Record" format. This could be FIR, IIR, FaceIR etc. Control Flags is an optional list of name value pairs that can be used to configure the behavior of the library. Output Parameters Biometric Extract Record in the form of FMR or a proprietary structure. The extract also contains a quality score. Errors / Exceptions Unsupported biometric type Unsupported image format Processing error
Behavior	Fingerprint ● The extractor will extract either an ISO template FMR or extract a proprietary representation that can give better results. The extract record will be marked with the format and any additional metadata needed. IRIS ● Currently there are no known IRIS extraction standards. Any template extracted can be consumed only by a corresponding matcher. Face ● Face analysis yields landmarks on face and these are typically stored in proprietary formats. Any template extracted can be used by a corresponding matcher.
Notes	For non fingerprint biometrics characteristics and patterns and landmarks might be identified and stored in a custom format. This format will be proprietary to each extractor and can be only consumed by its corresponding matcher. The extract will also contain meta information about the extractor and the version of the algorithm it uses and any other assumption it has made in the process of extraction that can be useful during matching.

Function	Segmenter
Use Cases	1) Used to split images into individual biometric segments when received from external sources
Signature	Input Parameters Biometric Image in "Biometric Image Record" format. This could be FIR, IIR, FaceIR etc. Control Flags is an optional list of name value pairs that can be used to configure the behavior of the library.

Annex B: Biometric Functions API

	Output Parameters List of Biometric Image Record that contains the individual biometrics Errors / Exceptions Unsupported biometric type Unsupported image format Processing error
Behavior	Fingerprint • Input will contain unsegmented image such as left slap or right slap or two thumbs • Output will be biometrics of each finger present in the input image IRIS • Input will contain unsegmented image such as both eyes • Output will be biometrics of each eye present in the input image Face • Not applicable at present
Notes	The segmenter will identify the individual fingers present.

Image Record Formats

Image Records will have three parts

- Header Information
- Image Block
- Security Block

The header information will comply to the standard specified here:
https://www.iso.org/standard/60458.html

The image block itself will comply to formats specified here
https://github.com/mosip/mosip/wiki/MOSIP-Biometric-Data-Specifications#data-standards-for-registration

The Security Block will be as per CBEFF specification. It is optional.

The cbeff format reference is here -
http://docs.oasis-open.org/bioserv/BIAS/v2.0/csprd01/schemas/cbeff_ed2.xsd

Score Object

The quality score object will have two sections. One is the score section and the other is the analytics section.

Score Parameters

- Graded score on a scale of 100. This is an unsigned float that represents a % and can have values between 0 and 100 including decimals like 82.45.
- Internal Score as defined on the scale of the implementing library. This is an unsigned long

Analytics

- List of name value pairs that can be used to convey additional information. The values filled are specific to the implementing library. This could contain information about the aspects where quality is failing for e.g. ICAO compliance for tilt or lighting. In case of matches it could contain information like the NIST score, the algorithm used for matching and more.

Annex B: Biometric Functions API

Appendix A - Java Specifications

```
class Score
{
   float scaledScore;  //0  -  100, used for internal classification and
efficacy analysis
   long internalScore;  // used against threshold specified in config file
   KeyValuePair[] AnalyticsInfo;  // detailed breakdown and other
 information
}

class CompositeScore
{
   float scaledScore;  //0  -  100, used for internal classification and
efficacy analysis
   long internalScore;  // used against threshold specified in config file
   Score[] individualScores;  // List of score for individual matches.
 Array size matches the input sample array size.
   KeyValuePair[] AnalyticsInfo;   // detailed breakdown and other
information
}

class BiometricRecord
{
   VersionType version;
   VersionType cbeffVersion;
   Object[] other;
   BIRInfo birInfo;
   BDBInfo bdbInfo;
   SBInfo sbInfo;
   String bdb;  // Base64 Binary
   String sb;  // Base64 Binary
 }

class BIRInfo
{
   String creator;
   String index;  // UUID with pattern
[a-fA-F0-9]{8}\-([a-fA-F0-9]{4}\-){3}[a-fA-F0-9]{12}
   String payload;
   Boolean integrity;
   DateTime creationDate;
   DateTime notValidBefore;
   DateTime notValidAfter;
 }

class BDBInfo
{
   String challengeResponse;   // Base64 Binary
   String index;  // UUID with pattern
[a-fA-F0-9]{8}\-([a-fA-F0-9]{4}\-){3}[a-fA-F0-9]{12}
   RegistryInfo format;
   Boolean encryption;
   DateTime creationDate;
   DateTime notValidBefore;
   DateTime notValidAfter;
   BiometricType[] type;
   BiometricSubType subtype;
   ProcessedLevelType level;
   RegistryInfo product;
   RegistryInfo captureDevice;
   RegistryInfo featureExtractionAlgorithm;
   RegistryInfo comparisonAlgorithm;
   RegistryInfo compressionAlgorithm;
   PurposeType purpose;
   QualityType quality;
}

class RegistryInfo
{
   String organization;
   String type;
}
```

Annex B: Biometric Functions API

```
class SBInfo
{
   RegistryInfo format;
}

class VersionType
{
   int major;
   int minor;
}

class QualityType
{
   RegistryInfo algorithm;
   Score score;
   String qualityCalculationFailed;
}

enum BiometricType
{
   Scent,
   DNA,
   Ear,
   Face,
   Finger,
   Foot,
   HandGeometry,
   Vein,
   Iris,
   Retina,
   Voice,
   Gait,
   Keystroke,
   LipMovement,
   SignatureSign,
   Palm,
   BackOfHand,
   Wrist
}

enum BiometricSubType
{
   Left,
   Right,
   Thumb,
   IndexFinger,
   MiddleFinger,
   RingFinger,
   LittleFinger
}

enum ProcessedLevelType
{
   Raw,
   Intermediate,
   Processed
}

enum PurposeType
{
   Verify,
   Identify,
   Enroll,
   EnrollVerify,
   EnrollIdentify,
   Audit
}


interface IBioApi
{
   Score[] match(BiometricImageRecord sample, BiometricRecord[] gallery,
KeyValuePair[] flags);
```

Annex B: Biometric Functions API

```
   CompositeScore compositeMatch(BiometricRecord[] sampleList,
BiometricRecord[] recordList, KeyValuePair[] flags);
   QualityScore  checkQuality(BiometricRecord  sample,  KeyValuePair[]
flags);
   BiometricRecord  extractTemplate(BiometricRecord  sample,  KeyValuePair[]
flags);
   BiometricRecord[]  segment(BiometricRecord  sample,  KeyValuePair[]
flags);
}
```