Index of Content

1 Introduction..............................................................................................................................7
2 Software Architecture...........................................................................................................8
3 Application Profiles for German Identity Documents.........................................................10
   3.1 Application German Identity Card..................................................................................10
   3.2 Application German Electronic Passport........................................................................17
   3.3 Application German Electronic Residence Permit.........................................................24
   3.4 Verification ePassport and Identity Card using facial biometrics........................................31
4 Application Profiles for Biometric Visa...............................................................................40
   4.1 Application for Biometric Visa......................................................................................40
   4.2 Basic Identity Check Biometric Visa...............................................................................49
   4.3 Extended Identity Check Biometric Visa..........................................................................56
5 Mapping between Application Profiles / Function Modules.............................................63
6 List of Abbreviations............................................................................................................65
7 Bibliography........................................................................................................................68
8 Annex.......................................................................................................................................70
   8.1 Working Example: Enrolment.........................................................................................70
   8.2 Further specifications for SFPI and BSFPs.......................................................................73
List of Figures

Figure 2-1: Software Architecture for (biometric) document issuing authorities.................................8
Figure 2-2: Software Architecture for biometric quality assurance..........................................................9
Figure 2-3: QA module provider interface for face biometrics....................................................................9
Figure 2-4: QA module provider interface for fingerprint biometrics..........................................................9
Figure 3-1: Application German Identity Card - Photographer..................................................................10
Figure 3-2: Application German Identity Card – Live enrolment station..................................................11
Figure 3-3: Software Architecture for the Application German Identity Card.............................................13
Figure 3-4: Application German Identity Card..........................................................................................15
Figure 3-5: Application German Electronic Passport - Photographer.......................................................17
Figure 3-6: Application German Electronic Passport – Live enrolment station.........................................18
Figure 3-7: Software Architecture for the Application German Electronic Passport...............................20
Figure 3-8: Application German Electronic Passport................................................................................22
Figure 3-9: Application German Electronic Residence Permit - Photographer..........................................24
Figure 3-10: Application German Electronic Residence Permit – Live enrolment station........................25
Figure 3-11: Software Architecture for the Application German Electronic Residence Permit.................27
Figure 3-12: Application German Electronic Residence Permit.................................................................29
Figure 3-13: Verification ePassport and Identity Card - Facial Image.........................................................32
Figure 3-14: Software Architecture for the Verification of ePassports and electronic identity cards..........32
Figure 3-15: Verification ePassport and Identity Card using facial biometrics...........................................38
Figure 4-1: Instances of the Biometric Visa Application process.................................................................40
Figure 4-2: Application Biometric Visa - Live enrolment station.................................................................41
Figure 4-3: Application Biometric Visa - Photographer...............................................................................42
Figure 4-4: Software Architecture for the Application of Biometric Visa..................................................43
Figure 4-5: Application for a biometric visa...............................................................................................47
List of Tables

Table 1: Overview of BioSPI_Capture - Application German Identity Card........................................14
Table 2: Mapping for Application German Identity Card.................................................................16
Table 3: Overview of BioSPI_Capture - Application German Electronic Passport..........................21
Table 4: Mapping for Application German Electronic Passport......................................................23
Table 5: Overview of BioSPI_Capture - Application German Electronic Residence Permit...............28
Table 6: Mapping for Application German Electronic Residence Permit........................................28
Table 7: Overview of BioSPI_Capture - Verification ePassport and identity card............................34
Table 8: Overview of BioSPI_CreateTemplate - Verification ePassport and identity card................35
Table 9: Overview of BioSPI_Process - Verification ePassport and identity card............................35
Table 10: Overview of BioSPI_VerifyMatch - Verification ePassport and identity card.....................36
Table 11: Mapping for Verification ePassport and Identity Card.....................................................39
Table 12: Overview of BioSPI_Capture - Application for Biometric Visa.......................................44
Table 13: Overview of BioSPI_ControlUnit - Application for Biometric Visa................................45
Table 14: Mapping for Application for Biometric Visa.................................................................48
Table 15: Overview of BioSPI_Capture – Basic Identity Check Biometric Visa..............................52
Table 16: Overview of BioSPI_ControlUnit - Basic Identity Check Biometric Visa........................53
Table 17: Mapping for Basic Identity Check Biometric Visa..........................................................55
Table 18: Overview of BioSPI_Capture – Extended Identity Check Biometric Visa.........................60
Table 19: Overview of BioSPI_ControlUnit - Extended Identity Check Biometric Visa......................60
List of Figures

Table 20: Mapping for Extended Identity Check Biometric Visa ................................................................. 62
Table 21: Mapping between Application Profiles and Function Modules: German Identity Document .......................................................................................................................... 63
Table 22: Mapping between Application Profiles and Function Modules: Biometric Visa ...................... 64
Table 23: BioAPI SFPI ControlCode specification ...................................................................................... 73
Table 24: BioAPI SFPI InputData specification ......................................................................................... 73
1 Introduction

This document specifies the Software Architecture and Application Profiles (APs) within the scope of TR Biometrics. Against the background of conformance testing the Software Architecture and the required interfaces are introduced and described. Afterwards Application Profiles are explained and relevant Function Modules are assigned that have been introduced in part 1 of TR Biometrics (TR-03121-1).

An overview how this document can be used and how it correlates to the other parts of this guideline is given in TR-03121-1.

This document contains the mapping between Application Profiles and Function Modules.
2 Software Architecture

The Software Architecture pursues the uniform strategy to integrate biometric processes in different Enrolment-, Verification- and Identification scenarios within the scope of German public sector applications. The Software Architecture is based on open standards, in particular BioAPI 2.0 [ISO_19784-1], where available. Applications (e.g. German Identity Card Face Application) are using the BioAPI 2.0 Framework to access the particular functionality for the specific Application Profile, which is implemented in a BioAPI 2.0 Biometric Service Provider (BSP). Figure 2-1 gives an overview of the different enclosed layers, where the type of the Applications and BSPs should be seen as an example.

![Software Architecture Diagram]

In the following sections all relevant Application Profiles (APs) for electronic Identity Documents are introduced and described. An Application Profile supports for every biometric feature one or more separate Biometric Service Providers (BSP) that can be addressed through a standardised BioAPI_* interface according to [ISO_19784-1] by the application. A working example how a BSP is called by the application is given in the annex of this document. While the BSP typically implements a complex workflow, it uses a hardware component with a rather simple interface (e.g. acquisition of a single image). To address the sensor hardware, basically two approaches are possible.

One possibility is that a BSP can manage a sensor directly, which means that all functionality for initialisation, loading, processing and termination of the device is included in the BSP in a vendor specific way, like integrating the sensor vendors Software Development Kit (SDK).

Another possibility is the realisation of two disjoint components – a BSP on the application side and a Biometric Sensor Function Provider (BSFP) [ISO_19784-4] on the device side. Both components can be offered by different producers since the interfaces are standardised through BioSFPI [ISO_19784-4]. Nevertheless, further specifications for BSFPs and the SFPI are made within this technical guideline to address certain issues in terms of biometric sensors. Annex 8.2 specifies some additional parameters for fingerprint sensors and further communication requirements when using BSFPs.
Regardless of which alternative is chosen, additional information is set for the input and output interfaces. A BSP has to support the BioSPI_* interfaces according to [ISO_19784-1]. Additionally these interfaces are at the same time important for conformance testing. Besides the BioAPI interfaces that are used for the implementation of the requirements regarding the public sector applications the BSPs have to provide interfaces or methods that are used when a BSP is applied for conformance testing (e.g. loading of a predefined image instead of acquiring one directly). Interfaces for conformance testing are described in part 3 of [TR_03122].

In the case of quality assurance of biometric data no open standards or interfaces are available. Thus, a flexible provider-based architecture is specified for quality assurance. As shown in figure 2-2, a common QA module interface is used which chooses a QA module provider for quality assurance. The actual quality check of biometric data is implemented within the QA module provider. This architecture ensures high flexibility by using different quality assurance algorithms although the QA module interface is not standardised.

![Diagram](image)

It is up to the implementer of the quality assurance of how to specify this interface. One important requirement is to have an interface that supports various QA module providers which accept binary biometric data as input and return the quality assurance results in XML format according to the valid Function Module QA of the according profile (see figures 2-3 and 2-4).

![Figure 2-3: QA module provider interface for face biometrics](image)

![Figure 2-4: QA module provider interface for fingerprint biometrics](image)
3 Application Profiles for German Identity Documents

3.1 Application German Identity Card

The following Application Profile describes the application for an electronic German Identity Card.

Introduction

The requirements for the application of an electronic German Identity Card are determined by the PAuswG (Gesetz über Personalausweise und den elektronischen Identitätsnachweis), which mandates biometric characteristics to be included in the chip of the German Identity Card. By legal requirements, the inclusion of a facial image is mandatory, the inclusion of fingerprints is optional. Therefore, all requirements on the applicant for fingerprint capture shall be considered optional in the application process.

Process overview

For the application of a national identity card a photograph as well as two fingerprints of the applicant have to be captured electronically. This is done through the later described Function Modules Acquisition Hardware, Acquisition Software and Biometric Image Processing. There are two acceptable possibilities to provide a standard compliant facial image.

In the regular process, the photograph is taken by a photographer. It has to be assured that the picture is compliant with the requirements of biometric images [ISO_FACE]. This sequence is presented in figure 3-1. After a visual inspection of the image with the help of a photo guideline (“Fotomustertafel”) and if needed a photo template (“Lichtbildschablone“) the photo is scanned and afterwards processed by the quality assurance module (software).

Furthermore a signature is captured, too, but this is not part of the description within TR Biometrics.
Alternatively, the image can be taken by a live enrolment station as demonstrated in figure 3-2. The live enrolment station may support a quality assurance module (software) that helps the applicant to take a picture which complies to the requirements. In this case, coding of quality information (according to the requirements of the quality assurance function module) can optionally be provided.

In any case, using a quality assurance module (software) in the process is basically mandatory.

In the case of a non-compliant image, the official has to make a final decision regarding exceptions (based on legal rules) for further processing.

A fingerprint sensor is used to capture two fingerprints of the applicant. Normally this will be the left and the right index finger. In some cases it is not possible to get a compliant image for these fingers. Alternatively, the thumbs, middle fingers or ring fingers can be used. While the image is taken (by Acquisition Hardware, Acquisition Software and Biometric Image Processing), the quality of the fingerprint is assessed. If fingerprints are captured within a live enrolment station it has to be assured by organisational instruments that an adequate identification check is performed for further proceeding.

The biometric data can be reduced in size by lossy compression, but multiple lossy compressions are not allowed. For facial images software-based quality assurance is made on the processed and compressed image data. For fingerprints the compression is performed after the QA process.

The facial image and the fingerprints as well as the additional quality information, which is connected to the biometric data, are coded and then passed to the calling application.

**Target audience**

The Application Profile “Application German Identity Card” is relevant for the following instances:

- Identity card authorities (Personalausweisbehörden)
- Photographers
- Identity card producers
- Suppliers of hardware and software components
3 Application Profiles for German Identity Documents

- Vendors of live enrolment stations
Software Architecture Overview

The application is using the biometric functionality through a BioAPI 2.0 Framework. The described functionality shall be provided by two BioAPI 2.0 compliant Capture Biometric Service Providers (BSPs) that encapsulate the complete process and hardware management facilities as described in figure 3-3. The first BSP implements the process to capture the facial image and the second BSP implements the process to capture the fingerprints.

![Software Architecture for the Application German Identity Card](image)

The application may not need any knowledge about the processes described in this profile, it may simply accept the results of the Function Module Coding. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

- the BioAPI 2.0-compliant Framework,
- the capture BSP providing the complete biometric functionality and
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware.

Besides the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), both BSPs must provide among other the function BioSPI_Capture to deliver the result data. It shall be called by the application through the BioAPI 2.0 Framework with the following function parameters:
### Table 1: Overview of BioSPI_Capture - Application German Identity Card

#### Relevant standards and conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:
▸ ISO/IEC 19794-5 „Information technology – Biometric data interchange formats – Part 5: Face Image Data” [ISO_FACE]

Information for Function Modules

Note: In the workflow for face software based quality assurance will be run after the Compression, see module P-PH-GID for details.

An overview of the modules and interfaces between the modules is given in figure 3-4. All Function Modules necessary for the Application Profile “Application German Identity Card” are presented in table 2.
## Application Profiles for German Identity Documents

| Application Profile → Application German Identity Card |
|-----------------|------------------|
| **Function Module**² | **Facial Image** | **Fingerprints** |
| ↓ | | |
| **Process** | P-PH-APP | P-FP-GID |
| **Acquisition Hardware** | AH-PH-FBS/AH-PH-DC/AH-PH-VID³ | AH-FP-FTR |
| **Acquisition Software** | AS-PH-FBS/AS-PH-DC | AS-FP-SF/AS-FP-MF |
| **Biometric Image Processing** | BIP-PH-FBS/BIP-PH-DC | BIP-FP-APP |
| **Quality Assurance** | QA-PH-SB, QA-PH-PG, QA-PH-PT | QA-FP-APPD |
| **Compression** | COM-PH-JP2 | COM-FP-WSQR |
| **Coding** | COD-PH-GID | COD-FP-GID |
| **Operation** | O-PH-APP | O-FP-GID |
| **User Interface** | UI-PH-APP | UI-FP-APP |
| **Reference Storage** | REF-PH-GID | REF-FP-GID |
| **Biometric Comparison** | | |
| **Logging** | LOG-PH-GID | LOG-FP-GID |
| **Evaluation** | EVA-PH-GID | EVA-FP-GID |

Table 2: Mapping for Application German Identity Card

---

² Entries separated by a slash represent alternative Function Modules.

³ For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.
3.2 Application German Electronic Passport

The following Application Profile describes the application for a German Electronic Passport (ePassport).

Introduction

The requirements for the application of an German electronic passport are determined by the PaßG (Paßgesetz), which mandates biometric characteristics to be included in the chip of the German Electronic Passport. These requirements are based on the EU regulation 2252/2004 [EC_2252/2004].

By legal requirements, the inclusion of a facial image and of two fingerprints is mandatory.

Process overview

For the application of an ePassport a photograph as well as two fingerprints of the applicant have to be captured electronically. This is done through the later described Function Modules Acquisition Hardware, Acquisition Software and Biometric Image Processing. There are two acceptable possibilities to provide a standard compliant facial image.

In the regular process, the photograph is taken by a photographer. It has to be assured that the picture is compliant with the requirements of biometric images [ISO_FACE]. This sequence is presented in figure 3-5. After a visual inspection of the image with the help of a photo guideline ("Fotomustertafel") and if needed a photo template ("Lichtbildschablone") the photo is scanned and afterwards processed by the quality assurance module (software).

4 Furthermore a signature is captured, too, but this is not part of the description within TR Biometrics.
Alternatively, the image can be taken by a live enrolment station as demonstrated in figure 3-6. The live enrolment station may support a quality assurance module (software) that helps the applicant to take a picture which complies to the requirements. In this case, coding of quality information (according to the requirements of the quality assurance function module) can optionally be provided.

In any case, using a quality assurance module (software) in the process is basically mandatory.

In the case of a non-compliant image, the official has to make a final decision regarding exceptions (based on legal rules) for further processing.

A fingerprint sensor is used to capture two fingerprints of the applicant. Normally this will be the left and the right index finger. In some cases it is not possible to get a compliant image for these fingers. Alternatively, the thumbs, middle fingers or ring fingers can be used. While the image is taken (by Acquisition Hardware, Acquisition Software and Biometric Image Processing), the quality of the fingerprint is assessed. If fingerprints are captured within a live enrolment station it has to be assured by organisational instruments that an adequate identification check is performed for further proceeding.

The biometric data can be reduced in size by lossy compression, but multiple lossy compressions are not allowed. For facial images software-based quality assurance is made on the processed and compressed image data. For fingerprints the compression is performed after the QA process.

The facial image and the fingerprints as well as the additional quality information, which is connected to the biometric data, are coded and then passed to the calling application.

**Target audience**

The Application Profile “Application German Electronic Passport” is relevant for the following instances:

- Passport authorities (Passbehörden)
- Photographers
- Passport producers
▸ Suppliers for hardware and software components
▸ Vendors of live enrolment stations
Software Architecture Overview

The application is using the biometric functionality through a BioAPI 2.0 Framework. The described functionality shall be provided by two BioAPI 2.0 compliant Capture Biometric Service Providers (BSPs) that encapsulate the complete process and hardware management facilities as described in figure 3-7. The first BSP implements the process to capture the facial image and the second BSP implements the process to capture the fingerprints.

![Software Architecture for the Application German Electronic Passport](image)

The application may not need any knowledge about the processes described in this profile, it may simply accept the results of the Function Module Coding. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

- the BioAPI 2.0-compliant Framework,
- the capture BSP providing the complete biometric functionality and
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware.

Beside the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), both BSPs must provide among other the function BioSPI_Capture to deliver the result data. It shall be called by the application through the BioAPI 2.0 Framework with the following function input parameters:
## BioSPI_Capture

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
<td>Input</td>
<td>BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
<td>Input</td>
<td>BioAPI_NO_SUBTYPEAVAILABLE</td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>const BioAPI_BIR_BIOMETRIC_DATA_FORMAT (OutputFormat)</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Handle to the result data, encoded as a Biometric Information Record (BIR). It can be obtained as defined by the BioAPI 2.0 standard (with the function BioAPI_GetBIRFromHandle)</td>
</tr>
<tr>
<td>int32_t Timeout</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AuditData</td>
<td>Output</td>
<td>This optional parameter is not covered by this guideline, it is left to the implementation of the BSP to deliver audit data.</td>
</tr>
</tbody>
</table>

*Table 3: Overview of BioSPI_Capture - Application German Electronic Passport*

### Relevant standards and conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:
3 Application Profiles for German Identity Documents

- ICAO Document 9303 Machine Readable Travel Documents, Part 3 - Size 1 and Size 2
- ISO/IEC 19784-1 „Information technology – Biometric application programming interface –
- ISO/IEC 19794-4 „Information technology – Biometric data interchange formats – Part 4:
  Finger Image Data” [ISO_FINGER]
- ISO/IEC 19794-5 „Information technology – Biometric data interchange formats – Part 5:
  Face Image Data” [ISO_FACE]

Information for Function Modules

![Diagram of Function Modules](image)

*Figure 3-8: Application German Electronic Passport*

Note: In the workflow for face software based quality assurance will be run after the Compression,
see module P-PH-GID for details.

An overview of the modules and interfaces between the modules is given in figure 3-8. All Function
Modules necessary for the Application Profile “Application German Electronic Passport” are
presented in table 4.
<table>
<thead>
<tr>
<th>Application Profile →</th>
<th>Application German Electronic Passport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Module⁵</td>
<td></td>
</tr>
<tr>
<td>↓</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>P-PH-APP</td>
</tr>
<tr>
<td></td>
<td>P-FP-GID</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-FBS/AH-PH-DC/AH-PH-VID⁶</td>
</tr>
<tr>
<td></td>
<td>AH-FP-FTR</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-PH-FBS/AS-PH-DC</td>
</tr>
<tr>
<td></td>
<td>AS-FP-SF/AS-FP-MF</td>
</tr>
<tr>
<td>Biometric Image</td>
<td>BIP-PH-FBS/BIP-PH-DC</td>
</tr>
<tr>
<td>Processing</td>
<td>BIP-FP-APP</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-SB, QA-PH-PG, QA-PH-PT</td>
</tr>
<tr>
<td></td>
<td>QA-FP-APPD</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-PH-JP2</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-GID</td>
</tr>
<tr>
<td>COD-FP-GID</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-APP</td>
</tr>
<tr>
<td></td>
<td>O-FP-GID</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-APP</td>
</tr>
<tr>
<td></td>
<td>UI-FP-APP</td>
</tr>
<tr>
<td>Reference Storage</td>
<td>REF-PH-GID</td>
</tr>
<tr>
<td></td>
<td>REF-FP-GID</td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-GID</td>
</tr>
<tr>
<td></td>
<td>LOG-FP-GID</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-GID</td>
</tr>
<tr>
<td></td>
<td>EVA-FP-GID</td>
</tr>
</tbody>
</table>

Table 4: Mapping for Application German Electronic Passport

---

⁵ Entries separated by a slash represent alternative Function Modules.

⁶ For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.
3.3 Application German Electronic Residence Permit

The following Application Profile describes the application for a German Electronic Residence Permit.

Introduction

The requirements for the application of a German Electronic Residence Permit are determined by the AufenthG (Aufenthaltsgesetz), which mandates biometric characteristics to be included in the chip of the German Electronic Residence Permit. These requirements are based on the EU regulations 1030/2002 and 380/2002.

By legal requirements, the inclusion of a facial image and of two fingerprints is mandatory.

Process overview

For the application of an Electronic Residence Permit a photograph as well as two fingerprints of the applicant have to be captured electronically. This is done through the later described Function Modules Acquisition Hardware, Acquisition Software and Biometric Image Processing. There are two acceptable possibilities to provide a standard compliant facial image.

In the regular process, the photograph is taken by a photographer. It has to be assured that the picture is compliant with the requirements of biometric images [ISO_FACE]. This sequence is presented in figure 3-9. After a visual inspection of the image with the help of a photo guideline (“Fotomustertafel”) and if needed a photo template (“Lichtbildschablone”) the photo is scanned and afterwards processed by the quality assurance module (software).

Furthermore a signature is captured, too, but this is not part of the description within TR Biometrics.
Alternatively, the image can be taken by a live enrolment station as demonstrated in figure 3-10. The live enrolment station may support a quality assurance module (software) that helps the applicant to take a picture which complies to the requirements. In this case, coding of quality information (according to the requirements of the quality assurance function module) can optionally be provided.

In any case, using a quality assurance module (software) in the process is basically mandatory.

In the case of a non-compliant image, the official has to make a final decision regarding exceptions (based on legal rules) for further processing.

![Diagram showing the process flow](image)

Figure 3-10: Application German Electronic Residence Permit – Live enrolment station

A fingerprint sensor is used to capture two fingerprints of the applicant. Normally this will be the left and the right index finger. In some cases it is not possible to get a compliant image for these fingers. Alternatively, the thumbs, middle fingers or ring fingers can be used. While the image is taken (by Acquisition Hardware, Acquisition Software and Biometric Image Processing), the quality of the fingerprint is assessed. If fingerprints are captured within a live enrolment station it has to be assured by organisational instruments that an adequate identification check is performed for further proceeding.

The biometric data can be reduced in size by lossy compression, but multiple lossy compressions are not allowed. For facial images software-based quality assurance is made on the processed and compressed image data. For fingerprints the compression is performed after the QA process.

The facial image and the fingerprints as well as the additional quality information, which is connected to the biometric data, are coded and then passed to the calling application.

**Target audience**

The Application Profile “Application German Electronic Residence Permit” is relevant for the following instances:

- Foreign affairs authorities (Ausländerbehörden)
- Photographers
- Electronic Residence permit producers
Application Profiles for German Identity Documents

- Suppliers for hardware and software components
- Vendors of live enrolment stations
Software Architecture Overview

The application is using the biometric functionality through a BioAPI 2.0 Framework. The described functionality shall be provided by two BioAPI 2.0 compliant Capture Biometric Service Providers (BSPs) that encapsulate the complete process and hardware management facilities as described in figure 3-11. The first BSP implements the process to capture the facial image and the second BSP implements the process to capture the fingerprints.

The application may not need any knowledge about the processes described in this profile, it may simply accept the results of the Function Module Coding. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

- the BioAPI 2.0-compliant Framework,
- the capture BSP providing the complete biometric functionality and
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware.

Beside the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), both BSPs must provide among other the function BioSPI_Capture to deliver the result data. It shall be called by the application through the BioAPI 2.0 Framework with the following function input parameters:
### BioSPI_Capture

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
<td>Input</td>
<td>BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
<td>Input</td>
<td>BioAPI_NO_SUBTYPE_AVAILABLE</td>
</tr>
<tr>
<td>const BioAPI_BIR_BIOMETRIC_DATA_FORMAT</td>
<td>Input (OutputFormat)</td>
<td>This parameter must denote format owner and format type of the encoding as described by the Function Module Coding. These values are registered and published by the Federal Office for Information Security (BSI). While the format owner is the same for both BSPs, the format type is different (one type for the result of the facial coding module and another type for the fingerprint coding module).</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *CapturedBIR</td>
<td>Output</td>
<td>Handle to the result data, encoded as a Biometric Information Record (BIR). It can be obtained as defined by the BioAPI 2.0 standard (with the function BioAPI_GetBIRFromHandle)</td>
</tr>
<tr>
<td>int32_t Timeout</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AuditData</td>
<td>Output</td>
<td>This optional parameter is not covered by this guideline, it is left to the implementation of the BSP to deliver audit data.</td>
</tr>
</tbody>
</table>

*Table 5: Overview of BioSPI_Capture - Application German Electronic Residence Permit*

### Relevant standards and conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:
Information for Function Modules

Note: In the workflow for face software based quality assurance will be run after the Compression, see module P-PH-GID for details.

An overview of the modules and interfaces between the modules is given in figure 3-12. All Function Modules necessary for the Application Profile “Application German Electronic Residence Permit” are presented in table 6.
### Table 6: Mapping for Application German Electronic Residence Permit

<table>
<thead>
<tr>
<th>Application Profile →</th>
<th>Application German Electronic Residence Permit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Module&lt;sup&gt;8&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>↓</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>P-PH-APP</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-FBS/AH-PH-DC/AH-PH-VID&lt;sup&gt;9&lt;/sup&gt;</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-PH-FBS/AS-PH-DC</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-PH-FBS/BIP-PH-DC</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-SB, QA-PH-PG, QA-PH-PT</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-PH-JP2</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-GID</td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-APP</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-APP</td>
</tr>
<tr>
<td>Reference Storage</td>
<td>REF-PH-GID</td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-GID</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-GID</td>
</tr>
</tbody>
</table>

---

<sup>8</sup> Entries separated by a slash represent alternative Function Modules.

<sup>9</sup> For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.
3.4 Verification ePassport and Identity Card using facial biometrics

The following Application Profile describes the identity verification in the context of electronic passports and identity cards.

Introduction

European legislation - European Regulation No. 2252/2004 - as well as national legal requirements (e.g. German PAuswG, Gesetz über Personalausweise und den elektronischen Identitätsnachweis or German Passgesetz, PaßG) mandate biometric characteristics to be included in electronic passports or in electronic identity cards, respectively. By legal requirements, the inclusion of a facial image is mandatory for both kinds of documents, the inclusion of fingerprints is optional for the German identity card. Therefore, verification of identity documents will primarily depend on examination of facial features whereas verification of identity based on fingerprints can be performed if available (this will be part of a separate Application Profile “Verification ePassport and Identity Card using fingerprint biometrics”10).

Process Overview

For the electronic verification of ePassports and national identity cards the identity document has to be equipped with biometric data which has been acquired and stored based on defined processes (e.g. based on the AP Application German Identity Card). Verification is typically part of border control which is performed at airports, seaports or land borders.

For biometric verification, authenticated biometric reference data (i.e. the face image of data group 2 according to the ICAO LDS Structure [ICAO_7_08]) is loaded from the electronic identity document. Furthermore, a live image is captured from the document holder by using a digital camera (i.e. acquisition hardware). The live image needs to be of sufficient quality for the comparison (e.g. by methods of prequalification). Finally, the identity of the document holder is checked by comparing both face images (compare figure 3-13).

Furthermore, for evaluation purposes it is necessary to conduct quality assurance and cross-comparison measures to get information of the biometric performance and quality of the border control process. Such an evaluation workflow is specified within the Software Architecture of this profile. Nevertheless, this part is independent of the actual border control process due to the architecture specification which ensures that evaluation and verification workflows are completely separated. The evaluation workflow is considered to be run in the background but in parallel to the actual border control process. Acquired and used biometric data for this evaluation are safely deleted right after results of the evaluation process are available.

10 This Application Profile will be added in a further version of the Technical Guideline.
3 Application Profiles for German Identity Documents

Target Audience

The Application Profile “Verification ePassport and Identity Card using facial biometrics” is relevant for the following instances:

▸ (Border) police authorities
▸ Suppliers of hardware and software components

Software Architecture Overview

The application is using the biometric functionality through a BioAPI 2.0 Framework and a QA module interface not standardised any further in this guideline. The described functionality for the verification process shall be provided by two BioAPI 2.0 compliant Biometric Service Providers (BSPs) - Capture BSP and Verification Engine - that encapsulate the complete capture process, hardware management and verification facilities as described in figure 3-14. The first BSP implements the capturing of live face data and the second BSP implements the process to verify facial images. For biometric quality assurance the provider-based QA module interface, as described in chapter 2, is available.

To combine the actual verification process with additional evaluation processing (Quality Assurance, Cross-comparison) the application needs to have knowledge about the processes described in this profile which are specified in the Function Module Process. The application
requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework and the QA module interface.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

▸ the BioAPI 2.0-compliant Framework,
▸ the BSPs (here: Capture BSP and Verification Engine) providing the complete biometric functionality of the verification workflow,
▸ optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware,
▸ the QA module interface and
▸ the QA module provider for facial features.

Besides the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), the Capture BSP must provide among other the biometric function BioSPI_Capture to deliver the result data. It shall be called by the application through the BioAPI 2.0 Framework with the following function parameters:
<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Type</strong></th>
<th><strong>Value/Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
<td>Input</td>
<td>BioAPI_PURPOSE_VERIFY</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
<td>Input</td>
<td>BioAPI_NO_SUBTYPE_AVAILABLE</td>
</tr>
<tr>
<td>const BioAPI_BIR_ BIOMETRIC_DATA_FORMAT</td>
<td>Input (OutputFormat)</td>
<td>This parameter must denote format owner and format type of the encoding (such as BMP, JPEG, JPEG2000, ...)</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *CapturedBIR</td>
<td>Output</td>
<td>Handle to the result data, encoded as a Biometric Information Record (BIR). It can be obtained as defined by the BioAPI 2.0 standard (with the function BioAPI_GetBIRFromHandle)</td>
</tr>
<tr>
<td>int32_t Timeout</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AuditData</td>
<td>Output</td>
<td>This optional parameter is not covered by this guideline, it is left to the implementation of the BSP to deliver audit data.</td>
</tr>
</tbody>
</table>

*Table 7: Overview of BioSPI_Capture - Verification ePassport and identity card*

The Verification Engine must provide among other the biometric functions BioSPI_CreateTemplate, BioSPI_Process, and BioSPI_VerifyMatch to deliver the result data. It shall also be called by the application through the BioAPI 2.0 Framework with the following function parameters:
### BioSPI_CreateTemplate

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>const BioAPI_INPUT_BIR *CapturedBIR;</td>
<td>Input</td>
<td>The previously captured data.</td>
</tr>
<tr>
<td>const BioAPI_INPUT_BIR *ReferenceTemplate</td>
<td>Input</td>
<td>Unused</td>
</tr>
<tr>
<td>const BioAPI_BIR_BIOMETRIC_DATA_FORMAT</td>
<td>Input</td>
<td>This parameter must denote format owner and format type of the encoding. The format is defined by the according vendor of the comparison software.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *NewTemplate</td>
<td>Output</td>
<td>The new template</td>
</tr>
<tr>
<td>BioAPI_DATA*Payload</td>
<td>Input</td>
<td>Unused</td>
</tr>
<tr>
<td>BioAPI_UUID *TemplateUUID</td>
<td>Output</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
</tbody>
</table>

Table 8: Overview of BioSPI_CreateTemplate - Verification ePassport and identity card

### BioSPI_Process

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_Handle BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>Const BioAPI_INPUT_BIR *CapturedBIR</td>
<td>Input</td>
<td>The previously captured data.</td>
</tr>
<tr>
<td>const BioAPI_BIR_BIOMETRIC_DATA_FORMAT</td>
<td>Input</td>
<td>This parameter must denote format owner and format type of the encoding. The format is defined by the according vendor of the comparison software.</td>
</tr>
<tr>
<td>BioAPI_BIRHANDLE *ProcessedBIR</td>
<td>Output</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
</tbody>
</table>

Table 9: Overview of BioSPI_Process - Verification ePassport and identity card
### 3 Application Profiles for German Identity Documents

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Type</strong></th>
<th><strong>Value/Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_Handle BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_FMR MaxFMRRRequested</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard; compare FM CMP-PH-VID</td>
</tr>
<tr>
<td>Const BioAPI_INPUT_BIR *ProcessedBIR</td>
<td>Input</td>
<td>The BIR to be verified</td>
</tr>
<tr>
<td>Const BioAPI_INPUT_BIR *ReferenceTemplate</td>
<td>Input</td>
<td>The reference to be verified against</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AdaptedBIR</td>
<td>Output</td>
<td>Unused</td>
</tr>
<tr>
<td>BioAPI_BOOL *Result</td>
<td>Output</td>
<td>Whether the verification was successful</td>
</tr>
<tr>
<td>BioAPI_FMR *FMRAchieved</td>
<td>Output</td>
<td>The achieved FMR</td>
</tr>
<tr>
<td>BioAPI_DATA *Payload</td>
<td>Output</td>
<td>Unused</td>
</tr>
</tbody>
</table>

Table 10: Overview of BioSPI_VerifyMatch - Verification ePassport and identity card

The QA module interface has to be specified according to the definition in chapter 2. At least JPEG and JPEG2000 formats have to be supported input formats of the implemented QA provider. Support of further formats for quality checks of the live captured face images might be needed depending on the implemented workflow.

### Relevant Standards and Conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:

- ISO/IEC 19794-5 „Information technology – Biometric data interchange formats – Part 5: Face Image Data” [ISO_FACE]
Information for Function Modules

An overview of the modules and interfaces between the modules is given in figure 3-15. All Function Modules necessary for the Application Profile “Verification ePassport and Identity Card using facial biometrics” are presented in table 11.

Figure 3-15: Verification ePassport and Identity Card using facial biometrics
### Table 11: Mapping for Verification ePassport and Identity Card

<table>
<thead>
<tr>
<th>Application Profile</th>
<th>Verification ePassport and Identity Card</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function Module</strong></td>
<td>Face</td>
</tr>
<tr>
<td>Process</td>
<td>P-PH-VID</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-VID</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-PH-VID</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-PH-VID</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-VID, (QA-PH-SB(^{11}))</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-PH-VID</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-VID</td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-VID</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-VID</td>
</tr>
<tr>
<td>Reference Storage</td>
<td></td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td>CMP-PH-VID</td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-VID</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-VID</td>
</tr>
</tbody>
</table>

\(^{11}\) This quality assurance is optional and it is only intended to be used for evaluation purposes.
4 Application Profiles for Biometric Visa

4.1 Application for Biometric Visa

For the Schengen Member States the following Application Profile describes the application for a Biometric Visa document.

Introduction

Harmonisation of EU visa regulation and the introduction of a central Visa Information System (VIS) by the Schengen Member States are connected to preparations for the issuance and handling of Biometric Visa Documents. The requirements for the application for a Biometric Visa are determined by [EC_2252/2004]. EU regulations mandate biometric characteristics to be transmitted to the central VIS database. By legal requirements, the inclusion of both a facial image as well as the full set of fingerprints is mandatory. The technical guideline in hand does not replace the EU specifications in any way, instead it amends the relevant biometric requirements to ensure high quality and interoperability.

Figure 4-1 illustrates the involvement of different instances in the visa application process. In the application office both the facial image and the fingerprints are captured electronically and submitted to the National Central Authority (NCA), that forwards the application to the central VIS respectively the Biometric Matching System (BMS). As an optional instance, the Biometric Evaluation Authority (BEA) has the duty to collect the logging data and generate statistics and reports.

Note that specifications of the VIS are out of the scope of this document.
Process overview

For the application of a Biometric Visa a photograph as well as all available fingerprints of the applicant have to be captured electronically. This is done through the later described Function Modules, especially:

- Acquisition Hardware,
- Acquisition Software and
- Biometric Image Processing.

In general, one should aim for a high quality of the captured facial images. There are two acceptable possibilities to provide a standard compliant facial image:

An image can be taken by a live enrolment station (including a digital camera) as demonstrated in figure 4-2 satisfying requirements of biometric images [ISO_FACE]. The live enrolment station may support a quality assurance module (software) that helps the applicant to take a picture which complies to the requirements. In this case, coding of quality information (according to the requirements of the quality assurance function module) can optionally be provided.

Although a photograph taken by a photographer can be used for the application, it is recommended for quality reasons to pass the afore described process using a live enrolment station. In case of a photo taken by a photographer it has to be assured that the picture is compliant with the requirements of biometric images [ISO_FACE]. This sequence is presented in figure 4-3. After a visual inspection of the image with the help of a photo guideline and, if needed, a photo template the photo is scanned and afterwards processed by the quality assurance module (software).

In any case, using a quality assurance module (software) in the process is recommended.
In the case of a non-compliant image, the official has to make a final decision regarding exceptions (based on legal rules) for further processing.

A fingerprint sensor is used to capture all available fingerprints of the applicant. Regulations require ten fingerprints to be submitted. Fingerprints that are unavailable (e.g. due to injury) have to be marked accordingly. All present fingers have to be captured. While the images are taken (by Acquisition Hardware, Acquisition Software, and Biometric Image Processing), the quality of the fingerprints is assessed. If the fingerprint scanner is not under direct supervision of the officer at the application counter - e.g. it is included in a live enrolment station - it has to be ensured that the fingerprints are allocated to the according applicant and that the applicant does not try to spoof fingerprints.

The biometric data can be reduced in size by lossy compression, but multiple lossy compressions are not allowed. For facial images software-based quality assurance is performed on the processed and compressed image data. For fingerprints, the compression is performed after the QA process.

The facial image and the fingerprints are coded according to the Function Module Coding. Afterwards, the encoded data is passed to the calling application.

It is highly recommended to perform statistical evaluations on a national level by the Biometric Evaluation Authority (BEA) based on logged quality data. Apart from these evaluations of the enrolment procedures, this also allows for further quality assurance on a national level. Due to the restricted decision information provided by the VIS, it seems adequate to perform own quality measurements using at least the VIS/BMS QA. Other mechanisms may be included where appropriate.

**Target audience**

The Application Profile “Application Biometric Visa” is relevant for the following instances:

- Consular authorities
- Border police authorities (in exceptional cases of application requests)
Software Architecture Overview

The application is using the biometric functionality through a BioAPI 2.0 Framework. The described functionality shall be provided by two BioAPI 2.0 compliant Capture Biometric Service Providers (BSPs) that encapsulate the complete process and hardware management facilities as described in figure 4-4. The first BSP implements the process to capture the facial image and the second BSP implements the process to capture the fingerprints.

The application may not need any knowledge about the processes described in this profile, it may simply accept the results of the Function Module Coding. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

The complete setting considered in this guideline consists of:

- the BioAPI 2.0-compliant Framework,
- the capture BSP providing the complete biometric functionality and
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware.

Besides the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), both BSPs must provide among other the function BioSPI_Capture to deliver the result.
data. If it is necessary to set additional data for coding the function BioSPI_ControlUnit has to be provided and implemented, too. Both functions shall be called by the application through the BioAPI 2.0 Framework with the following function parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
<td>Input</td>
<td>BioAPI_PURPOSE_ENROLL</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
<td>Input</td>
<td>BioAPI_NO_SUBTYPE_AVAILABLE</td>
</tr>
<tr>
<td>const BioAPI_BIR_ BIOMETRIC_ DATA_FORMAT</td>
<td>Input (OutputFormat)</td>
<td>This parameter must denote format owner and format type of the encoding as described by the Function Module Coding. These values are registered and published by the Federal Office for Information Security (BSI). While the format owner is the same for both BSPs, the format type is different (one type for the result of the facial coding module and another type for the fingerprint coding module).</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *CapturedBIR</td>
<td>Output</td>
<td>Handle to the result data, encoded as a Biometric Information Record (BIR). It can be obtained as defined by the BioAPI 2.0 standard (with the function BioAPI_GetBIRFromHandle)</td>
</tr>
<tr>
<td>int32_t Timeout</td>
<td>Input</td>
<td>Use -1 to let the BSP decide itself how to handle timeouts.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AuditData</td>
<td>Output</td>
<td>This optional parameter is not covered by this guideline, it is left to the implementation of the BSP to deliver audit data.</td>
</tr>
</tbody>
</table>

*Table 12: Overview of BioSPI_Capture - Application for Biometric Visa*
### BioSPI_ControlUnit

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_UNIT_ID UnitID</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard. Represents the ID of the BioAPI Unit.</td>
</tr>
<tr>
<td>uint32_t ControlCode</td>
<td>Input</td>
<td>This parameter must denote the control code as described by the Function Module Coding. This value is defined and published by the Federal Office for Information Security (BSI).</td>
</tr>
<tr>
<td>const BioAPI_DATA *InputData</td>
<td>Input</td>
<td>Pointer to the input data structure related to the given ControlCode. This parameter must denote the input data as described by the Function Module Coding.</td>
</tr>
<tr>
<td>BioAPI_DATA *OutputData</td>
<td>Output</td>
<td>Regular usage as defined by the BioAPI 2.0 standard. Output data is not returned by the ControlUnit call in terms of using it for application of biometric visa.</td>
</tr>
</tbody>
</table>

*Table 13: Overview of BioSPI_ControlUnit - Application for Biometric Visa*

In order to ensure proper functioning of the BSP it is necessary to call the BioSPI_ControlUnit function before the BioSPI_Capture call. If ControlCode and InputData are not defined within the according Function Module Coding the BioSPI_ControlUnit call can be omitted.

**Relevant standards and conditions**

In addition to the legal requirements (see above), further basic directives and standards are applicable:

- ISO/IEC 19794-5 „Information technology – Biometric data interchange formats – Part 5: Face Image Data” [ISO_FACE]
- Regulation (EC) No. 767|2008 of the European Parliament and of the Council of 9 July 2008 concerning the Visa Information System (VIS) and the exchange of data between Member States on short-stay visas (VIS Regulation) [EC_767_2008]
Information for Function Modules

Note: In the workflow for face software based quality assurance will be run after the Compression, see the module P-PH-APP for details.

An overview of the modules and interfaces between the modules is given in figure 4-5. All Function Modules necessary for the Application Profile “Application for Biometric Visa” are presented in table 14.
### Application Profiles for Biometric Visa

<table>
<thead>
<tr>
<th>Application Profile →</th>
<th>Biometric Visa</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function Module</strong></td>
<td>Facial Image</td>
</tr>
<tr>
<td>Process</td>
<td>P-PH-APP</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-FBS/AH-PH-DC/AH-PH-VID&lt;sup&gt;13&lt;/sup&gt;</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-PH-FBS/AS-PH-DC</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-PH-FBS/BIP-PH-DC</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-SB, QA-PH-PG, QA-PH-PT</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-PH-JPG/COM-PH-JP2</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-VAPP</td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-APP</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-APP</td>
</tr>
<tr>
<td>Reference Storage</td>
<td>REF-PH-VAPP</td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-VAPP</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-VAPP</td>
</tr>
</tbody>
</table>

**Table 14: Mapping for Application for Biometric Visa**

<sup>12</sup> Entries separated by a slash represent alternative Function Modules.

<sup>13</sup> For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.
4.2 Basic Identity Check Biometric Visa

The following Application Profile describes a basic identity check i.e. verification in the context of a biometric visa document. This scenario is often related to as first position at border control checks and it is optimised for fast completion, high throughput and ease of use. For a quality-oriented verification process also suitable for 1-to-n identification purposes, please refer to the Application Profile Extended Identity Check Biometric Visa which is related to as second position (compare section 4.3).

Introduction

Harmonisation of EU visa regulation and the introduction of a central Visa Information System (VIS) for the Schengen area mandate member states to make preparations in order to issue and handle Biometric Visa Documents. Applicable regulations for identity verification in the context of biometric visa are determined by [VIS-08].

Process Overview

Note: The following process encloses the acquisition of the biometric feature, the transmission of the data and the later response of the VIS. The actual comparison is out of scope of this technical guideline.

The verification of a Biometric Visa Document consists of both a visual check of the centrally electronically stored and printed facial image as well as an electronic fingerprint comparison against the central VIS database. Fingerprint comparison utilises the following Function Modules

- FM Acquisition Hardware,
- FM Acquisition Software,
- FM Biometric Image Processing,
- FM Compression,
- FM Coding,
- FM Biometric Comparison and
- FM Operation

as described later in this document. Figure 4-6 illustrates the parties and information flow involved in the verification process. In a first step, the visa sticker number is read from the visa document. A request is coded and transmitted to the VIS via the National Central Authority (NCA). This entity returns information on whether biometric reference data is available, as well as the stored facial image. If biometric reference data is available for the requested visa, fingerprint verification is started and live-captured fingerprint data is sent to the Biometric Matching System (BMS) of the VIS for comparison (coding of all captured fingerprints is done according to VIS-ANSI/NIST specification [VIS-ANSI_NIST]). The BMS informs the requesting entity of the comparison result or errors that have been encountered.
Optionally, it is possible to directly send the verification request (including the live-captured fingerprint data) to the BMS of the VIS. In that case, a verification result is only returned if reference fingerprints are stored accordingly. If no reference fingerprints are found the system will return an appropriate error message.

For the acquisition of a live fingerprint image, both single-finger and multi-finger capture devices may be used. Details concerning the selection of the finger to be checked can be found in the corresponding Function Module. In case of failure of the basic identity check, the border control officer decides if the traveller is expected to undergo an extended identity check procedure as described in section 4.3 or not.

It is required to perform statistical evaluations on a national level by the Biometric Evaluation Authority (BEA) based on logged data. For this reason, additional quality assurance of the live image data is conducted. Within the Software Architecture an evaluation workflow is specified to get information about the quality of live data. This data is then used for performing statistical evaluations on a national level.

**Target Audience**

The Application Profile “Basic Identity Check Biometric Visa” is relevant for the following instances:

- Border guard and other police authorities
- Suppliers of hardware and software components
- Operator of the Biometric Evaluation Authority (BEA)

**Software Architecture Overview**

The application is using the biometric functionality through a BioAPI 2.0 Framework and a non-standardised QA module interface for evaluation purposes. Capturing live image data shall be
provided by a BioAPI 2.0 compliant Capture Biometric Service Provider (BSP) that encapsulates the complete process and hardware management facilities. The BSP implements the process to capture the fingerprint(s).

The actual verification is conducted by the VIS BMS (Biometric Matching System of the Visa Information System). As the National Central Authority is the connector between the border control posts and the VIS BMS it defines the interfaces needed for conducting verifications against the BMS. The application itself only needs to care about capturing live fingerprint data, sending the data via the NCA to the VIS BMS, and retrieving and interpreting the result of the BMS.

For biometric quality assurance no standardised framework or interface is available. Thus, the provider-based architecture, as specified in chapter 2, is used for quality assurance.

To combine the actual authentication process with additional evaluation processing (Quality Assurance) the application needs to have knowledge about the processes described in this profile which are specified in the Function Module Process. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework and the QA module interface.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

- the BioAPI 2.0-compliant Framework,
- the BSP (here: Capture BSP) providing the complete biometric functionality of the capture process,
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware,
- the QA module interface and
- the QA module provider for fingerprints.

Besides the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), the Capture BSP must provide among other the biometric function BioSPI_Capture to
deliver the result data. If it is necessary to set additional data for coding the function BioSPI_ControlUnit has to be provided and implemented, too. Both functions shall be called by the application through the BioAPI 2.0 Framework with the following function parameters:

<table>
<thead>
<tr>
<th>BioSPI_Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
</tr>
<tr>
<td>const BioAPI_BIR_ BIOMETRIC_ DATA_FORMAT</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *CapturedBIR</td>
</tr>
<tr>
<td>int32_t Timeout</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *AuditData</td>
</tr>
</tbody>
</table>

*Table 15: Overview of BioSPI_Capture – Basic Identity Check Biometric Visa*
### BioSPI_ControlUnit

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BioAPI_UNIT_ID UnitID</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard. Represents the ID of the BioAPI Unit.</td>
</tr>
<tr>
<td>uint32_t ControlCode</td>
<td>Input</td>
<td>This parameter must denote the control code as described by the Function Module Coding. This value is defined and published by the Federal Office for Information Security (BSI).</td>
</tr>
<tr>
<td>const BioAPI_DATA *InputData</td>
<td>Input</td>
<td>Pointer to the input data structure related to the given ControlCode. This parameter must denote the input data as described by the Function Module Coding.</td>
</tr>
<tr>
<td>BioAPI_DATA *OutputData</td>
<td>Output</td>
<td>Regular usage as defined by the BioAPI 2.0 standard. Output data is not returned by the ControlUnit call in terms of using it for application of biometric visa.</td>
</tr>
</tbody>
</table>

*Table 16: Overview of BioSPI_ControlUnit - Basic Identity Check Biometric Visa*

In order to ensure proper functioning of the BSP it is necessary to call the BioSPI_ControlUnit function before the BioSPI_Capture call. If ControlCode and InputData are not defined within the according Function Module Coding the BioSPI_ControlUnit call can be omitted.

The QA module interface has to be specified according to the definition in chapter 2. At least the ANSI/NIST-ITL format for fingerprints has to be supported as input format of the implemented QA provider. Support of further formats for quality checks might be needed depending on the implemented workflow.

### Relevant Standards and Conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:

4 Application Profiles for Biometric Visa

- Regulation (EC) No. 767|2008 of the European Parliament and of the Council of 9 July 2008 concerning the Visa Information System (VIS) and the exchange of data between Member States on short-stay visas (VIS Regulation) [EC_767_2008]

Information for Function Modules

An overview of the modules and interfaces between the modules is given in figure 4-8. All Function Modules necessary for the Application Profile “Basic Identity Check Biometric Visa” are presented in table 17.
<table>
<thead>
<tr>
<th>Application Profile</th>
<th>Function Module</th>
<th>Basic Identity Check Visa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process</td>
<td>P-FP-VBIC</td>
</tr>
<tr>
<td></td>
<td>Acquisition Hardware</td>
<td>AH-FP-FTR</td>
</tr>
<tr>
<td></td>
<td>Acquisition Software</td>
<td>AS-FP-MF/AS-FP-SF</td>
</tr>
<tr>
<td></td>
<td>Biometric Image Processing</td>
<td>BIP-FP-VIS</td>
</tr>
<tr>
<td></td>
<td>Quality Assurance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Compression</td>
<td>COM-FP-WSQ</td>
</tr>
<tr>
<td></td>
<td>Coding</td>
<td>COD-FP-VIS</td>
</tr>
<tr>
<td></td>
<td>Operation</td>
<td>O-FP-ACQ</td>
</tr>
<tr>
<td></td>
<td>User Interface</td>
<td>UI-FP-VBIC</td>
</tr>
<tr>
<td></td>
<td>Reference Storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Biometric Comparison</td>
<td>CMP-FP-VIC</td>
</tr>
<tr>
<td></td>
<td>Logging</td>
<td>LOG-FP-VIC</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>EVA-FP-VIC</td>
</tr>
</tbody>
</table>

_Entries separated by a slash represent alternative Function Modules._

*Table 17: Mapping for Basic Identity Check Biometric Visa*
4.3 Extended Identity Check Biometric Visa

The following Application Profile describes an extended identity check of a biometric visa document. This scenario is often related to as second position at border control checks and it is optimised for quality-based verification as well as identification purposes. For a verification targeted towards fast completion and ease of use, please refer to the Application Profile Basic Identity Check Biometric Visa (see section 4.2).

Introduction

Harmonisation of EU visa regulation and the introduction of a central Visa Information System (VIS) by the Schengen states mandate member states to make preparations in order to issue and handle Biometric Visa Documents. Applicable regulations for identity verification in the context of biometric visa are determined by [VIS-08].

Process Overview

Note: The following process encloses the acquisition of the biometric feature, the transmission of the data and the later response of the VIS. The actual comparison is out of scope of this technical guideline.

The extended identity check of a Biometric Visa Document consists of both a visual check of the centrally electronically stored and printed facial image as well as an electronic fingerprint comparison against the central VIS database. Contrary to the basic identity check process, verification and/or identification is conducted within this process. Fingerprint comparison utilises the following Function Modules

▸ FM Acquisition Hardware,
▸ FM Acquisition Software,
▸ FM Biometric Image Processing,
▸ FM Quality Assurance,
▸ FM Compression,
▸ FM Coding,
▸ FM Operation and
▸ FM Biometric Comparison

as described later in this document. Figure 4-9 illustrates the parties and information flow involved in the extended identity check process. If information about the according visa is not available yet, typically, the visa sticker number is read from the visa document in a first step (if a biometric verification is conducted). A request is coded and transmitted to the VIS via the National Central Authority (NCA). This entity returns information on whether biometric reference data is available,
as well as the stored facial image. If biometric reference data for the requested visa is available, the identity check is started and live-captured fingerprint data is acquired.

Optionally, it is possible to directly send the verification request (including the live-captured fingerprint data) to the BMS of the VIS. In that case, a verification result is only returned if reference fingerprints are stored accordingly. If no reference fingerprints are found the system will return an appropriate error message.

In case of using biometric identifications to determine the identity of the visa holder, all fingerprints are captured directly without needing the visa sticker number or the information if reference fingerprint data is stored. The result of the biometric identification will give feedback about this fact and further information of if the visa holder could be identified or not.

For the acquisition of live fingerprint images, multi-finger capture devices are required. In general, all ten fingers have to be captured (depending on possible handicaps of the document holder). After capturing, local quality assurance is conducted in order to obtain best possible identity check results. If necessary, capturing is repeated for certain fingers due to low quality of fingerprint images. All captured and quality-checked fingerprint data is then encoded according to the VIS-ANSI/NIST specification [VIS-ANSI_NIST] and sent to the VIS BMS for comparison. It is up to the border control post to decide if verification or identification or both identity checks are conducted. The BMS informs the requesting entity of the comparison result or errors that have been encountered.

In case of failure of the extended identity check, the traveller is expected to undergo a further, manual border control procedure by the border control office.

It is required to perform statistical evaluations on a national level by the Biometric Evaluation Authority (BEA) based on logged data. For this reason, additional quality assurance of the live image data is conducted. Within the Software Architecture an evaluation workflow is specified to get information about the quality of live data. This data is then used for performing statistical evaluations on a national level.
**Target Audience**

The Application Profile “Extended Identity Check Biometric Visa” is relevant for the following instances:

- Border guard and other police authorities
- Suppliers of hardware and software components
- Operator of the Biometric Evaluation Authority (BEA)

**Software Architecture Overview**

The application is using the biometric functionality through a BioAPI 2.0 Framework and a non-standardised QA module interface for evaluation purposes. The process of capturing live image data shall be encapsulated in a BioAPI 2.0 compliant Capture Biometric Service Provider (BSP) which implements the capturing and hardware management facilities. Furthermore, the Capture BSP cares about local quality assurance for using the best possible fingerprint image data for identity check.

The actual identity check is conducted by the VIS BMS (Biometric Matching System of the Visa Information System). As the National Central Authority (NCA) is the connector between the border control posts and the VIS BMS it defines the interfaces needed for conducting identity checks against the BMS. The application itself only needs to care about capturing live fingerprint data, sending the data via the NCA to the VIS BMS, and retrieving and interpreting the result of the BMS.

For further biometric quality assurance no standardised framework or interface is available. Thus, the provider-based architecture, as specified in chapter 2, is used for quality assurance.
To combine the actual authentication process with additional evaluation processing (Quality Assurance) the application needs to have knowledge about the processes described in this profile which are specified in the according Function Module Process. The application requests the respective data through the BioAPI_* interfaces (according to [ISO_19784-1]) from the BioAPI 2.0 Framework and the QA module interface.

As described in chapter 2, the hardware functionality may be accessed directly by the BSP or as an option by using a sensor through the Biometric Sensor Function Provider Interface (BioSFPI).

Therefore the complete setting considered in this guideline consists of

- the BioAPI 2.0-compliant Framework,
- the BSP (here: Capture BSP) providing the complete biometric functionality (and local quality assurance) of the capture process,
- optionally Biometric Sensor Function Providers (BSFP) for distributed access to the sensor hardware,
- the QA module interface and
- the QA module provider for fingerprints.

Besides the necessary BioAPI functions to load a BSP and create an attach session (compare BioAPI_*), the Capture BSP must provide among other the biometric function BioSPI_Capture to deliver the result data. If it is necessary to set additional data for coding the function BioSPI_ControlUnit has to be provided and implemented, too. Both functions shall be called by the application through the BioAPI 2.0 Framework with the following function parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
<td>Input</td>
<td>Regular usage as defined by the BioAPI 2.0 standard.</td>
</tr>
<tr>
<td>BIOAPI_BIR_PURPOSE</td>
<td>Input</td>
<td>BioAPI_PURPOSE_VERIFY or BioAPI_PURPOSE_IDENTIFY</td>
</tr>
<tr>
<td>BioAPI_BIR_SUBTYPE</td>
<td>Input</td>
<td>BioAPI_NO_SUBTYPE_AVAILABLE</td>
</tr>
<tr>
<td>const BioAPI_BIR_BIOMETRIC_DATA_FORMAT</td>
<td>Input (OutputFormat)</td>
<td>This parameter must denote format owner and format type of the encoding as described by the Function Module Coding. These values are registered and published by the Federal Office for Information Security (BSI).</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE *CapturedBIR</td>
<td>Output</td>
<td>Handle to the result data, encoded as a Biometric Information Record (BIR). It can be obtained as defined by the BioAPI 2.0 standard (with the</td>
</tr>
</tbody>
</table>
function BioAPI_GetBIRFromHandle)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int32_t Timeout</td>
<td>Input</td>
<td>Use -1 to let the BSP decide itself how to handle timeouts.</td>
</tr>
<tr>
<td>BioAPI_BIR_HANDLE</td>
<td>Output</td>
<td>This optional parameter is not covered by this guideline, it is left to the implementation of the BSP to deliver audit data.</td>
</tr>
</tbody>
</table>

Table 18: Overview of BioSPI_Capture – Extended Identity Check Biometric Visa

<table>
<thead>
<tr>
<th>BioSPI_ControlUnit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>BioAPI_HANDLE BSPHandle</td>
</tr>
<tr>
<td>BioAPI_UNIT_ID UnitID</td>
</tr>
<tr>
<td>uint32_t ControlCode</td>
</tr>
<tr>
<td>const BioAPI_DATA *InputData</td>
</tr>
<tr>
<td>BioAPI_DATA *OutputData</td>
</tr>
</tbody>
</table>

Table 19: Overview of BioSPI_ControlUnit - Extended Identity Check Biometric Visa

In order to ensure proper functioning of the BSP it is necessary to call the BioSPI_ControlUnit function before the BioSPI_Capture call. If ControlCode and InputData are not defined within the according Function Module Coding the BioSPI_ControlUnit call can be omitted.

The QA module interface has to be specified according to the definition in chapter 2. At least the ANSI/NIST-ITL format for fingerprints has to be supported as input format of the implemented QA provider. Support of further formats for quality checks might be needed depending on the implemented workflow.
Standards and Conditions

In addition to the legal requirements (see above), further basic directives and standards are applicable:


▸ Regulation (EC) No. 767|2008 of the European Parliament and of the Council of 9 July 2008 concerning the Visa Information System (VIS) and the exchange of data between Member States on short-stay visas (VIS Regulation) [EC_767_2008]


Information for Function Modules

An overview of the modules and interfaces between the modules is given in figure 4-11. All Function Modules necessary for the Application Profile “Extended Identity Check Biometric Visa” are presented in table 20.
<table>
<thead>
<tr>
<th>Function Module&lt;br&gt;15</th>
<th>Extended Identity Check Biometric Visa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>P-FP-VEIC</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-FP-FTR</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-FP-MF</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-FP-VIS</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-FP-CRM</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-FP-WSQ</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-FP-VIS</td>
</tr>
<tr>
<td>Operation</td>
<td>O-FP-ACQ</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-FP-VEIC</td>
</tr>
<tr>
<td>Reference Storage</td>
<td></td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td>CMP-FP-VIC</td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-FP-VIC</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-FP-VIC</td>
</tr>
</tbody>
</table>

*Table 20: Mapping for Extended Identity Check Biometric Visa*

---

15 Entries separated by a slash represent alternative Function Modules.
5 Mapping between Application Profiles / Function Modules

Tables 21 and 22 show which part of the Function Module is valid for the respective Application Profile. One field of the table can contain several entries. Furthermore, the table can contain empty fields.

<table>
<thead>
<tr>
<th>Application Profile → Function Module(^{16})</th>
<th>German Identity Documents</th>
<th>Veriﬁcation ePassport and Identity Card</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Application German Identity Card, Application German Electronic Passport, Application German Electronic Residence Permit</td>
<td></td>
</tr>
<tr>
<td>Face</td>
<td>Finger</td>
<td>Face</td>
</tr>
<tr>
<td>P-PH-APP</td>
<td>P-FP-GID</td>
<td>P-PH-VID</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-FBS/ AH-PH-DC/ AH-PH-VID(^{17})</td>
<td>AH-FP-FTR</td>
</tr>
<tr>
<td>Acquisition Software</td>
<td>AS-PH-FBS/ AS-PH-DC</td>
<td>AS-FP-SF/ AS-FP-MF</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-PH-FBS/ BIP-PH-DC</td>
<td>BIP-FP-APP</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-SB, QA-PH-PG, QA-PH-PT</td>
<td>QA-FP-APPD, QA-PH-VID, (QA-PH-SB(^{18}))</td>
</tr>
<tr>
<td>Compression</td>
<td>COM-PH-JP2</td>
<td>COM-FP-WSQR</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-GID</td>
<td>COD-FP-GID</td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-APP</td>
<td>O-FP-GID</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-APP</td>
<td>UI-FP-APP</td>
</tr>
<tr>
<td>Reference Storage</td>
<td>REF-PH-GID</td>
<td>REF-FP-GID</td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td></td>
<td>CMP-PH-VID</td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-GID</td>
<td>LOG-FP-GID</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-GID</td>
<td>EVA-FP-GID</td>
</tr>
</tbody>
</table>

Table 21: Mapping between Application Profiles and Function Modules: German Identity Document

\(^{16}\) Entries separated by a slash represent alternative Function Modules.

\(^{17}\) For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.

\(^{18}\) This quality assurance is optional and it is only intended to be used for evaluation purposes.
## Mapping between Application Profiles / Function Modules

<table>
<thead>
<tr>
<th>Application Profile → Function Module</th>
<th>Biometric Visa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Application Biometric Visa</td>
</tr>
<tr>
<td></td>
<td>Face</td>
</tr>
<tr>
<td>Process</td>
<td>P-PH-APP</td>
</tr>
<tr>
<td>Acquisition Hardware</td>
<td>AH-PH-FBS/ AH-PH-DC/ AH-PH-VID&lt;sup&gt;20&lt;/sup&gt;</td>
</tr>
<tr>
<td>Biometric Image Processing</td>
<td>BIP-PH-FBS/ BIP-PH-DC</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>QA-PH-SB, QA-PH-PG, QA-PH-PT</td>
</tr>
<tr>
<td>Coding</td>
<td>COD-PH-VAPP</td>
</tr>
<tr>
<td>Operation</td>
<td>O-PH-APP</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI-PH-APP</td>
</tr>
<tr>
<td>Reference Storage</td>
<td>REF-PH-VAPP</td>
</tr>
<tr>
<td>Biometric Comparison</td>
<td></td>
</tr>
<tr>
<td>Logging</td>
<td>LOG-PH-VAPP</td>
</tr>
<tr>
<td>Evaluation</td>
<td>EVA-PH-VAPP</td>
</tr>
</tbody>
</table>

*Table 22: Mapping between Application Profiles and Function Modules: Biometric Visa*

---

19 Entries for different biometric characteristics are separated by a comma whereas alternatives are separated by a slash.

20 For Live Enrolment Stations AH-PH-VID applies, whereas AH-PH-FBS or AH-PH-DC applies for photographers.
6 List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQ</td>
<td>Acquisition</td>
</tr>
<tr>
<td>AH</td>
<td>Acquisition Hardware</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>APP</td>
<td>Application</td>
</tr>
<tr>
<td>AP</td>
<td>Application Profiles</td>
</tr>
<tr>
<td>AS</td>
<td>Acquisition Software</td>
</tr>
<tr>
<td>BEA</td>
<td>Biometric Evaluation Authority</td>
</tr>
<tr>
<td>BIP</td>
<td>Biometric Image Processing</td>
</tr>
<tr>
<td>BMS</td>
<td>Biometric Matching System</td>
</tr>
<tr>
<td>BioAPI</td>
<td>Biometric Application Programming Interface</td>
</tr>
<tr>
<td>BioSFPI</td>
<td>Biometric Sensor Function Provider Interface</td>
</tr>
<tr>
<td>BioSPI</td>
<td>BioAPI Service Provider Interface</td>
</tr>
<tr>
<td>BSI</td>
<td>Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security)</td>
</tr>
<tr>
<td>BFP</td>
<td>Biometric Function Provider</td>
</tr>
<tr>
<td>BSFP</td>
<td>Biometric Sensor Function Provider</td>
</tr>
<tr>
<td>BSP</td>
<td>Biometric Service Provider</td>
</tr>
<tr>
<td>CMP</td>
<td>Biometric Comparison</td>
</tr>
<tr>
<td>COD</td>
<td>Coding</td>
</tr>
<tr>
<td>COM</td>
<td>Compression</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>CRM</td>
<td>Cross-matching</td>
</tr>
<tr>
<td>DC</td>
<td>Digital camera</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>EVA</td>
<td>Evaluation</td>
</tr>
<tr>
<td>FBS</td>
<td>Flat bed scanner</td>
</tr>
<tr>
<td>FP</td>
<td>Fingerprint</td>
</tr>
<tr>
<td>FTR</td>
<td>Frustrated total reflection</td>
</tr>
<tr>
<td>GID</td>
<td>German Identity Document</td>
</tr>
<tr>
<td>ICAO</td>
<td>International Civil Aviation Organization</td>
</tr>
<tr>
<td>ID</td>
<td>Identity</td>
</tr>
<tr>
<td>JPG</td>
<td>JPEG (ISO/IEC 10918-1)</td>
</tr>
<tr>
<td>JP2</td>
<td>JPEG 2000 (ISO/IEC 15444)</td>
</tr>
<tr>
<td>LOG</td>
<td>Logging</td>
</tr>
<tr>
<td>MF</td>
<td>Multi finger</td>
</tr>
<tr>
<td>NCA</td>
<td>National Central Authority</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>O</td>
<td>Operation</td>
</tr>
<tr>
<td>P</td>
<td>Process</td>
</tr>
<tr>
<td>PG</td>
<td>Photo guideline</td>
</tr>
<tr>
<td>PH</td>
<td>Photo</td>
</tr>
<tr>
<td>PT</td>
<td>Photo template</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>REF</td>
<td>Reference Storage</td>
</tr>
<tr>
<td>SB</td>
<td>Software based</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SF</td>
<td>Single finger</td>
</tr>
<tr>
<td>TR</td>
<td>Technische Richtlinie (Technical Guideline)</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>VAPP</td>
<td>Visa Application</td>
</tr>
<tr>
<td>VBIC</td>
<td>Basic Identity Check Biometric Visa</td>
</tr>
<tr>
<td>VEIC</td>
<td>Extended Identity Check Biometric Visa</td>
</tr>
<tr>
<td>VIC</td>
<td>Visa Identity Check</td>
</tr>
<tr>
<td>VID</td>
<td>Verification Identity Document</td>
</tr>
<tr>
<td>VIS</td>
<td>Visa Information System</td>
</tr>
<tr>
<td>WSQ</td>
<td>Wavelet Scalar Quantisation</td>
</tr>
<tr>
<td>WSQR</td>
<td>Wavelet Scalar Quantisation for reference storage</td>
</tr>
</tbody>
</table>
7 Bibliography


<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
8 Annex

8.1 Working Example: Enrolment

In the following a standard procedure for Enrolment is presented for the application level within the scope of the Software Architecture. A detailed description of the interfaces can be found in [ISO_19784-1].

- Initialisation of the framework: BioAPI_Init()
- Enumeration of all available BSPs: BioAPI_EnumBSPs()
- Loading of a BSP: BioAPI_BSPLoad(explUUID, ...)
  (registration & event handling)
- Setup of a BSP session: BioAPI_BSPAttach(&explHandle, ...)
- Start of the Acquisition: BioAPI_Capture(explHandle, ...)
- Request for the result data: BioAPI_GetBIRFromHandle(explHandle, ...)
- Processing of the result data ...
- End of the Session: BioAPI_BSPDetach(explHandle, …)
- Unloading of a BSP: BioAPI_BSPUnload(explUUID, …)
- Termination of the framework: BioAPI_Terminate()

// BioAPI_Demo_Application.cpp :
// Defines the entry point for the console application.

#include "stdafx.h"
#include "BioAPI.h"

#define BioAPI_MAJOR (2)
#define BioAPI_MINOR (0)

int _tmain(int argc, _TCHAR* argv[])
{

    BioAPI_RETURN result;

    // initialise BioAPI framework
    BioAPI_VERSION version;
    version = (BioAPI_VERSION)((BioAPI_MAJOR << 4) | BioAPI_MINOR);
    result = BioAPI_Init( version );

    if ( result != BioAPI_OK ) {
        printf( "Error: BioAPI_Init failed!\n" );

        return -1;
    }
}
// load the BSP with UUID ffffffff-ffff-ffff-ffff-ffffffffffff

// Note: this is just exemplary, a real UUID would look different
// UUIDs are defined by RFC 4122

BioAPI_UUID uuid = { 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
                   0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
                   0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff };

result = BioAPI_BSPLoad( &uuid, NULL, NULL );

if ( result != BioAPI_OK ) {
    printf( "Error: BioAPI_BSPLoad failed!\n" );
    return -1;
}

// attach to BSP

BioAPI_HANDLE handle;

result = BioAPI_BSPAttach( &uuid, version, NULL, 0, &handle );

if ( result != BioAPI_OK ) {
    printf( "Error: BioAPI_BSPAttach failed!\n" );
    return -1;
}

// do something with the BIR

// capture

BioAPI_BIR_BIOMETRIC_DATA_FORMAT dataformat;

// Note: these values are just exemplary. Data should be requested
// in a format that has been registered by the IBIA
// (see http://www.ibia.org/cbeff/)

dataformat.FormatOwner = 0xffff;
dataformat.FormatType = 0xffff;

BioAPI_BIR_HANDLE capturedBIRHandle;

result = BioAPI_Capture(handle, BioAPI_PURPOSE_ENROLL,
                         BioAPI_NO_SUBTYPE_AVAILABLE, &dataformat,
                         &capturedBIRHandle, -1, NULL );

if ( result != BioAPI_OK ) {
    printf( "Error: BioAPI_Capture failed!\n" );
    return -1;
}

// get BIR from handle

BioAPI_BIR capturedBIR;
capturedBIR.BiometricData.Data = NULL;
capturedBIR.SecurityBlock.Data = NULL;

result = BioAPI_GetBIRFromHandle( handle, capturedBIRHandle,
                                   &capturedBIR );
if ( result != BioAPI_OK ) {
    printf( "Error: BioAPI_GetBIRFromHandle failed!\n" );
    return -1;
}

// save Biometric Data Block to file
FILE* file = fopen( "C:\\captureResult.bin", "wb" );
if ( !fwrite( capturedBIR.BiometricData.Data,
             capturedBIR.BiometricData.Length, 1, file ) ) {
    printf( "Error: fwrite failed!\n" );
    return -1;
}
fclose( file );

// free allocated data
if ( capturedBIR.BiometricData.Data )
    BioAPI_Free( capturedBIR.BiometricData.Data );
if ( capturedBIR.SecurityBlock.Data )
    BioAPI_Free( capturedBIR.SecurityBlock.Data );

// detach from BSP
BioAPI_BSPDetach( handle );

// unload the BSP
BioAPI_BSPUnload( &uuid, NULL, NULL );

// terminate BioAPI framework
BioAPI_Terminate();
return 0;
8.2 Further specifications for SFPI and BSFPs

Specification for additional SFPI communication

Certain biometric sensors require information about the chosen subtype of the biometric modality to be acquired. As an example, especially in terms of auto-capture functionality, multi-fingerprint sensors need to know when to trigger auto-capture. This functionality is mainly based on the information of how many fingers should be placed on the sensor surface. Through the available Sensor Function Provider Interface (SFPI) standardised in [ISO_19784-4] such information exchange is not provided by given functions. Hence, a BioSFPI_ControlUnit call is required to inform the selected BSFP which subtypes were chosen during the BSP function calls. Table 23 defines the ControlCode and table 24 the according XML-InputData structure for this BioSFPI_ControlUnit call.

<table>
<thead>
<tr>
<th>Organisation Name</th>
<th>ControlCode</th>
<th>InputData Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federal Office for Information Security (BSI)</td>
<td>0x603A0BC7</td>
<td>bioapi-bsfp</td>
</tr>
</tbody>
</table>

*Table 23: BioAPI SFPI ControlCode specification*

**XSD representation**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Status</th>
<th>Number</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bioapi-bsfp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtype</td>
<td>string</td>
<td>O</td>
<td>0..*</td>
<td>Specifies the subtype mask of the biometric modality. Enumeration with following values: “left”, “right”, “thumb” “pointerfinger”, “middlefinger”, “ringfinger”, “littlefinger”, “multiple”.</td>
</tr>
</tbody>
</table>

*Table 24: BioAPI SFPI InputData specification*

The XSD representation (bioapi_bsfp.xsd) of this InputData structure is as follows whereas the type type.subtype.mask is defined in bio_common.xsd in Annex A of TR-03121-3.
XML example coding

The following XML example coding (bioapi.bsfp.xml) describes a right slap consisting of all four fingers used as subtype for fingerprint acquisition.

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <bio_common:Subtype>right</bio_common:Subtype>
  <bio_common:Subtype>pointerfinger</bio_common:Subtype>
</bio_common:bioapi-bsfp>
```
Finger handicaps (amputated or currently not available) can be considered by only setting the appropriate subtype values of each available finger. As an example, a handicap of the right ring finger when trying to capture the right slap of a person expects following subtype values to be set: “right”, “pointerfinger”, “middlefinger”, “littlefinger”. Considering the acquisition of iris images, as an example, only the value “left” should be set for capturing the left iris.

**Specification of additional BSFP parameters**

If biometric sensors are encapsulated within a Biometric Sensor Function Provider (BSFP) according to the biometric imaging sensor units specification in Annex A of [ISO_19784-4], further information about the biometric sensor may be needed in the Biometric Service Provider (BSP) or even in the application using the BSP. Such information of the BSFP may be needed by the BSP to determine which available and installed BSFPs might be appropriate for the required application scenario. Thus, further information of biometric devices might be stored in the *AdditionalParameters* element of the *BioSFPI_BSFPImagePropertySchema* stored in the BioAPI component registry during installation of the BSFP.

In the following, additional parameters and requirements for fingerprint sensors encapsulated within BSFPs are specified.

**Requirements for fingerprint sensors**

Fingerprint sensors encapsulated within BSFPs are required to support hardware- or software-based auto-capture functionality. Furthermore, it is required at least to support the function *BioSFPI_GetPackets* for fingerprint data exchange between the calling BSP and the selected BSFP. By sending “last packet” during the call of this function, triggered auto-capture of the BSFP can be easily signalled to the BSP (see [ISO_19784-4]).

The function *BioSFPI_GetPackets* will only transmit the final captured image. For displaying live stream images in the BSP GUI callbacks shall be used. Via calling the function *BioSFPI_SetGUICallbacks* GUI streaming and GUI state callbacks shall be enabled. GUI streaming callbacks shall be used for transmitting the live stream data between the BSFP and the BSP. GUI state callbacks shall be used for transmitting further information. Additionally, the BSP has to respond to received GUI state callback messages. The response BioAPI_CAPTURE_SAMPLE shall be sent if the BSP manually triggers capturing. BioAPI_CONTINUE means that the capture process shall continue.

Certain fingerprint sensors return information about location and positioning errors. This information shall be provided to the BSP by using the *Message* parameter of the GUI state callback. Following error codes might then be transmitted via the *Message* parameter to signal location errors:
▸ BioAPIERR_LOCATION_TOO_LEFT: Signals that the finger was located too left on the device.
▸ BioAPIERR_LOCATION_TOO_RIGHT: Signals that the finger was located too right on the device.
▸ BioAPIERR_LOCATION_TOO_FORWARD: Signals that the finger was located too forward on the device.
▸ BioAPIERR_LOCATION_TOO_BACKWARD: Signals that the finger was located too backward on the device.

This information might then be used in the BSP to display guiding information (e.g. in form of appropriate arrows) to the user in the live stream.

For storing additional information in the AdditionalParameters element of the BioSFPI_BSFPImagePropertySchema of the BFP schema of the installed BSFP, the structure TR03121_BSFP_IMAGE_PROPERTY_SCHEMA_ADDITIONAL_PARAMETERS_FINGERPRINT is specified (see definition of additional parameters structure for fingerprint sensors below). It contains following information:

▸ AdditionalParametersID: This UUID describes the type and content of additional parameters being used within this AdditionalParameters element. For fingerprint sensors in the scope of this technical guideline it is defined to be set to the value ec7d9afb-45a0-4490-8e26-5d8bc3e71671.

▸ MaximumNumberOfSupportedFingers: This element describes the maximum number of supported fingers being captured at once by the fingerprint sensor. For single-finger sensors this value is set to 1 while for four-finger sensors this value is set to 4, typically.

▸ SensorType: This element describes the sensor technology (optical with FTR, optical without FTR, capacitive, thermic, ultrasonic, radio-frequency, pressure sensitive, ...).

▸ SensorArchitecture: This element describes the sensor architecture (swipe, area sensors, ...).

▸ LifeFingerDetectionSupported: This element describes if the sensor is capable of detecting life fingers.

▸ AcquisitionMethod: This element describes which acquisition method the sensor is capable of (flat fingerprints, rolled fingerprints, ...).

▸ SensorDPI: This element describes the maximum scanning resolution in dots per inch (dpi).

▸ SensorAreaWidth: This element describes the width of the sensor area surface (in millimeters).

▸ SensorAreaHeight: This element describes the height of the sensor area surface (in millimeters).

▸ AutoCaptureSupported: This element describes if the sensor supports either hardware- or software-based auto-capture functionality.

▸ SensorCertification: This element contains information about certifications of the fingerprint sensor (e.g. FBI Appendix F, PIV, BSI certified, ...).

Definition of additional parameters structure for fingerprint sensors
/*****************************************************************************
*  BioSFPI_BSFPImagePropertySchema_AdditionalParameters.h
*  **************************************************************************/

#ifndef BIOSFPI_BSFPIMAGEPROPERTYSCHEMA_ADDITIONALPARAMETERS_H
#define BIOSFPI_BSFPIMAGEPROPERTYSCHEMA_ADDITIONALPARAMETERS_H

#include "BioAPI.h"

// TR03121_FINGERPRINT_SENSORTYPE
// This type describes the fingerprint sensor type
typedef uint8_t TR03121_FINGERPRINT_SENSORTYPE;

#define TR03121_FINGERPRINT_SENSORTYPE_UNKNOWN (0x00)
#define TR03121_FINGERPRINT_SENSORTYPE_OPTICAL_FTR (0x01)
#define TR03121_FINGERPRINT_SENSORTYPE_OPTICAL_NON_FTR (0x02)
#define TR03121_FINGERPRINT_SENSORTYPE_CAPACITIVE (0x03)
#define TR03121_FINGERPRINT_SENSORTYPE_THERMIC (0x04)
#define TR03121_FINGERPRINT_SENSORTYPE_ULTRASONIC (0x05)
#define TR03121_FINGERPRINT_SENSORTYPE_RF (0x06)
#define TR03121_FINGERPRINT_SENSORTYPE_PRESSURE_SENSITIVE (0x07)
#define TR03121_FINGERPRINT_SENSORTYPE_OTHER (0xFF)

// TR03121_FINGERPRINT_SENSORARCHITECTURE
// This type describes the fingerprint sensor architecture
typedef uint8_t TR03121_FINGERPRINT_SENSORARCHITECTURE;
#define TR03121_FINGERPRINT_SENSORARCHITECTURE_UNKNOWN (0x00)
#define TR03121_FINGERPRINT_SENSORARCHITECTURE_AREA (0x01)
#define TR03121_FINGERPRINT_SENSORARCHITECTURE_SWIPE (0x02)
#define TR03121_FINGERPRINT_SENSORARCHITECTURE_OTHER (0xFF)

//--
//-- TR03121_FINGERPRINT_ACQUISITION_METHOD
//-- This type describes the fingerprint acquisition method
typedef uint8_t TR03121_FINGERPRINT_ACQUISITION_METHOD;

#define TR03121_FINGERPRINT_ACQUISITION_METHOD_UNKNOWN (0x00)
#define TR03121_FINGERPRINT_ACQUISITION_METHOD_FLAT (0x01)
#define TR03121_FINGERPRINT_ACQUISITION_METHOD_ROLLED (0x02)
#define TR03121_FINGERPRINT_ACQUISITION_METHOD_OTHER (0xFF)

//--
//-- TR03121_FINGERPRINT_SENSOR_CERTIFICATION
//-- This bitmask describes the certification of the fingerprint sensor
typedef uint8_t TR03121_FINGERPRINT_SENSOR_CERTIFICATION;

#define TR03121_FINGERPRINT_SENSOR_CERTIFICATION_NO_CERTIFICATION (0x00)
#define TR03121_FINGERPRINT_SENSOR_CERTIFICATION_FBI_APPENDIX_F (0x01)
#define TR03121_FINGERPRINT_SENSOR_CERTIFICATION_PIV (0x02)
#define TR03121_FINGERPRINT_SENSOR_CERTIFICATION_BSI (0x04)
#define TR03121_FINGERPRINT_SENSOR_CERTIFICATION_OTHER (0x80)

//--
//-- TR03121_BSFP_IMAGEPROPERTY_SCHEMA_ADDITIONALPARAMETERS_FINGERPRINT
//--
// Defines the format of the data encoded in the AdditionalParameters element of
// the BSFPImagePropertySchema as defined in ISO/IEC 19784-4 (fingerprint)

typedef struct tr03121_bsfp_image_property_schema_additional_parameters_fingerprint
{
    BioAPI_UUID AdditionalParametersID; // UUID of the additional parameters
    // defined in TR-03121

    uint8_t MaximumNumberOfSupportedFingers; // Maximum numbers of fingers being
    // captured by the sensor at once

    TR03121_FINGERPRINT_SENSOR_TYPE SensorType; // Fingerprint sensor type

    TR03121_FINGERPRINT_SENSOR_ARCHITECTURE SensorArchitecture; // Fingerprint sensor architecture

    BioAPI_BOOL LifeFingerDetectionSupported; // Support of life finger detection

    TR03121_FINGERPRINT_ACQUISITION_METHOD AcquisitionMethod; // Supported acquisition method of
    // sensor

    uint32_t SensorDPI; // Resolution of sensor (DPI)

    uint32_t SensorAreaWidth; // Sensor area width in millimeters

    uint32_t SensorAreaHeight; // Sensor area height in millimeters

    BioAPI_BOOL AutoCaptureSupported; // Support of auto capture functionality

    TR03121_FINGERPRINT_SENSOR_CERTIFICATION SensorCertification; // Availability of sensor certification
} TR03121_BSFP_IMAGE_PROPERTY_SCHEMA_ADDITIONAL_PARAMETERS_FINGERPRINT;

#endif

Serialisation of additional parameters structure

Serialisation is necessary for writing the above mentioned data structure into the
AdditionalParameters element of the BioSFPI_BSFPImagePropertySchema. Furthermore,
deserialisation is necessary for the other way around. Below, appropriate functions are defined for
this purpose. Following Annex D.2 of the BioAPI specification[ISO_19784-1], functions needed
for serialisation and deserialisation of Biometric Information Records (BIRs) are used, accordingly.
void SerializeImagePropertySchemaAdditionalParametersFingerprint(
    TR03121_BSFP_IMAGE_PROPERTY_SCHEMA_ADDITIONAL_PARAMETERS_FINGERPRINT addParams,
    BioSFPI_BSFP_IMAGE_PROPERTY_SCHEMA & imagePropertySchema)
{
    // get size of serialized data and allocate memory
    uint32_t size = sizeof(TR03121_BSFP_IMAGE_PROPERTY_SCHEMA_ADDITIONAL_PARAMETERS_FINGERPRINT);
    imagePropertySchema.AdditionalParameters = malloc(size);
    imagePropertySchema.AdditionalParametersSize = size;
    uint8_t* buffer = (uint8_t*)imagePropertySchema.AdditionalParameters;
    // now serialize data into buffer
    buffer = SerializeByteArrayToBuffer(buffer, addParams.AdditionalParametersID,
        sizeof(BioAPI_UUID));
    buffer = SerializeByteToBuffer(buffer, addParams.MaximumNumberOfSupportedFingers);
    buffer = SerializeByteToBuffer(buffer, addParams.SensorType);
    buffer = SerializeByteToBuffer(buffer, addParams.SensorArchitecture);
    buffer = SerializeByteToBuffer(buffer, addParams.LifeFingerDetectionSupported);
    buffer = SerializeByteToBuffer(buffer, addParams.AcquisitionMethod);
    buffer = SerializeIntToBuffer(buffer, addParams.SensorDPI);
    buffer = SerializeIntToBuffer(buffer, addParams.SensorAreaWidth);
    buffer = SerializeIntToBuffer(buffer, addParams.SensorAreaHeight);
    buffer = SerializeByteToBuffer(buffer, addParams.AutoCaptureSupported);
    buffer = SerializeByteToBuffer(buffer, addParams.SensorCertification);
}
/**
 * Name: DeserializeImagePropertySchemaAdditionalParametersFingerprint
 * Purpose: Deserializes additional parameters for fingerprint sensors as defined
 * in TR 03121-2
 * Parameters:
 *   addParams [OUT] - Deserialized struct of additional parameters
 *   imagePropertySchema [IN] - Data to be serialized
 * Return Value : BioAPI_BOOL indicating success of deserialization
 */

BioAPI_BOOL DeserializeImagePropertySchemaAdditionalParametersFingerprint(
    TR03121_BSFP_IMAGEPROPERTY_SCHEMA_ADDITIONALPARAMETERS_FINGERPRINT* addParams,
    const BioSFPI_BSFP_IMAGEPROPERTY_SCHEMA & imagePropertySchema)
{
    uint8_t* buffer = (uint8_t*)imagePropertySchema.AdditionalParameters;
    buffer = ExtractByteArrayFromBuffer(buffer, addParams->AdditionalParametersID,
        sizeof(BioAPI_UUID));

    BioAPI_UUID fpUUID = { 0xec, 0x7d, 0x9a, 0xfb, 0x45, 0xa0, 0x44, 0x90, 0x26,
        0x5d, 0x8b, 0xc3, 0xe7, 0x16, 0x71 };

    if (memcmp(&fpUUID, &addParams->AdditionalParametersID, sizeof(BioAPI_UUID)) == 0) {
        // only deserialize if UUID is properly set
        buffer = ExtractByteFromBuffer(buffer, &addParams->MaximumNumberOfSupportedFingers);
        buffer = ExtractByteFromBuffer(buffer, &addParams->SensorType);
        buffer = ExtractByteFromBuffer(buffer, &addParams->SensorArchitecture);
        buffer = ExtractByteFromBuffer(buffer, &addParams->LifeFingerDetectionSupported);
        buffer = ExtractByteFromBuffer(buffer, &addParams->AcquisitionMethod);
        buffer = ExtractIntFromBuffer(buffer, &addParams->SensorDPI);
        buffer = ExtractIntFromBuffer(buffer, &addParams->SensorAreaWidth);
        buffer = ExtractIntFromBuffer(buffer, &addParams->SensorAreaHeight);
        buffer = ExtractByteFromBuffer(buffer, &addParams->AutoCaptureSupported);
    }
}
8 Annex

```c
buffer = ExtractByteFromBuffer(buffer, &addParams->SensorCertification);
}
else
    return BioAPI_FALSE;

return BioAPI_TRUE;
}

/**
 * Name:    FreeSerializedImagePropertySchemaAdditionalParametersFingerprint
 * Purpose: Frees allocated memory during serialization
 * Parameters:
 *    * imagePropertySchema [IN] - Struct containing serialized data
 *    * Return Value : NONE
 */

void FreeSerializedImagePropertySchemaAdditionalParametersFingerprint(
    BioSFPI_BSFP_IMAGE_PROPERTY_SCHEMA & imagePropertySchema)
{
    if (imagePropertySchema.AdditionalParameters)
    {
        free(imagePropertySchema.AdditionalParameters);
        imagePropertySchema.AdditionalParameters = NULL;
        imagePropertySchema.AdditionalParametersSize = 0;
    }
}

Specification of additional SFPI functions

Due to a missing function for unloading the loaded BSFP in the SFPI specification of the standard [ISO_19784-4] it might be impossible for some biometric devices to clean up properly all allocated resources. Without such a function in the interface the only possibility for unloading would be to clean up when the BSFP module (e.g. a Windows DLL) is unloaded. Nevertheless, some devices may have problems with unloading their sensors at this stage. Hence, an unload function related to
definitions of the BioAPI 2.0 standard [ISO_19784-1] is defined within the scope of this technical guideline.

```c
BioAPI_RETURN BioSFPI_BSFPUndload

    (const BioAPI_UUID *BSFPUuid,
     BioSFPI_EventHandler BioSFPINotifyCallback);
```

**Description**

This function disables events and de-registers the BSP event-notification function. The BSFP shall perform clean up operations, reversing the initialisation performed in BioSFPI_BSFPLoad.

**Parameters**

- **BSFPUuid (input)** – The UUID of the invoked BSFP.
- **BioSFPINotifyCallback (input)** – A function pointer for the event handler that manages events of type BioAPI_EVENT.

**Return Value**

A BioAPI_RETURN value indicating success or specifying a particular error condition. The value BioAPI_OK indicates success. All other values represent an error condition.

**Errors**

BioAPIERR_INVALID_UUID