1 Introduction

A4Vision offers a customized hardware/software solution for integrators and end-users seeking reliable door access control. VISION ACCESS™ Software Development Kit (VA SDK) is a professional software kit for designing, developing and implementing access control systems (ACSs) based on A4Vision's 3D Face Recognition technology: This section is a brief introduction to the VA SDK and provides you with two subjects:

- Overview
- Quick Start.

1.1 Overview

As organizations search for more secure authentication methods for user access, e-commerce, and other security applications, biometrics is gaining increasing attention. Biometrics is an open-ended set of technologies used to recognize a person based upon physical or behavioral characteristics. Some of the most challenging solutions are based on facial biometrics. Of them, three-dimensional (3D) face recognition is the most promising one.

The VA SDK is intended to facilitate customizing and leveraging the power of 3D facial identification solutions based on A4Vision's 3D technology (www.a4vision.com).

The VA SDK enables the addition of facial biometrics and secure connection into back-end applications, deployment of the overall system and its integration into existing third-party applications. The VA SDK manages Face Reader (FR) stations, performs user enrollment, auditing, system analysis and monitoring of stand-alone solutions locally and centrally across network-configured access control. The FR hardware uses an industry standard interface protocol to communicate with external card readers and door controllers.

Examples of integration and development applications include time and attendance, monitoring and surveillance, site security, and existing physical access technology, including custom embedded hardware, firmware, and software implementations.

1.2 Quick Start

⇒ Plan your development environment.

⇒ Equip PC with ES (Enrollment Station) hardware.

⇒ Setup FR (Face Reader) hardware

⇒ Install VA SDK.

2 VA SDK Package at a Glance

This section contains a short description of the VA SDK package used for design, development, and implementation of access control systems based on A4Vision's 3D Face Recognition technology:

The section contains information on the following subjects:

- Overview
- BioAPI Connectivity
- Access Control System Network Connectivity (VA Connection)
- Licensing Infrastructure
- Reference Design Sources.

2.1 Overview

Vision Access™ SDK is a tool with the help of which you can use A4Vision's patented 3D Face recognition technology. The following provides an overview of the VA SDK components and a simple schematic of the systems integrating 3D Face biometrics.

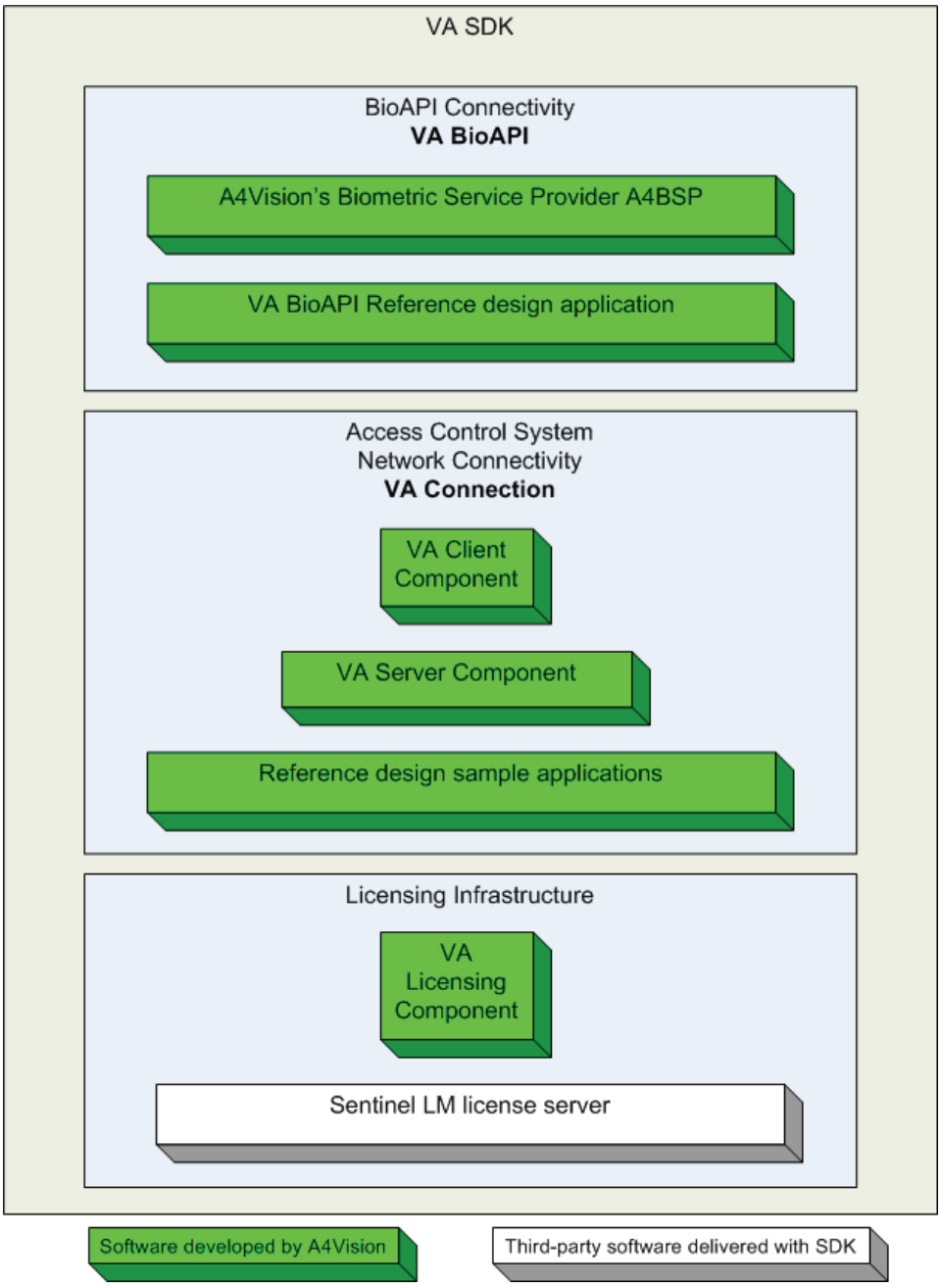

Fig. 2.1 VA SDK components.

Fig. 2.1 shows the VA SDK software components that provide two kind of connectivity to A4vision's core software modules—one through the standard BioAPI and another via the A4Vision's VA Connection components.

2.2 BioAPI Connectivity

The above-mentioned BioAPI connectivity is implemented with A4BSP—A4Vision's Biometric Service Provider module. The A4BSP is built over A4vision's core modules.

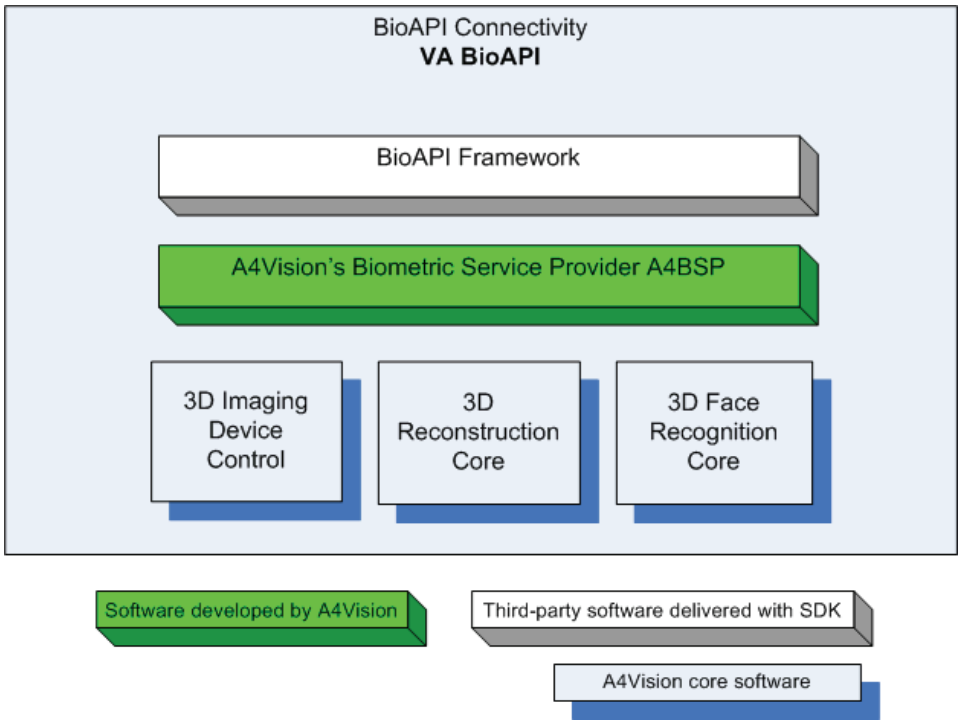

Fig. 2.2 A Biometric Service Provider.

According to the BioAPI 1.1 standard requirements (see "Glossary: Biometric Terms"), the A4BSP implements the corresponding standard basic functions:

- Enroll
- Verify
- Identify
- Capture
- Create Template
- Process
- Verify Match
- Identify Match

- BioAPI_SetGUICallbacks

Note

The BioAPI_SetGUICallbacks function allows the application to establish callbacks so that the application may control the "look-and-feel" of the biometric user interface. For details, refer to "BioAPI Specification Version 1.1."

The A4BSP controls ES (Enrollment Station) 3D Imaging devices (see Fig. 2.2). Using the A4BSP standard functions, you can:

⇒ Integrate the 3D Face recognition technology with the help of the industry standard BioAPI framework.

⇒ Develop a custom application for enrollment.

⇒ Develop client-server biometric applications.

A4Vision provides no direct interface to the system's core modules.

2.3 ACS Network Connectivity

The A4Vision solution of network connectivity—VA Connection—is intended to facilitate building an access control system (ACS) that networks standalone face recognition devices—Face Readers (FRs).

Note

The VA Face Reader (FR) is a shorthand of the Face Reader Optical Unit (FRO)—a 3D surface scanner device providing a real-time capture of 3D facial biometric data for matching the data against information stored in the corresponding local database or on the card of a person exposing his/her face to the FR. The FRO is not a standalone device, it works in conjunction and with support of the FR Controller(FRC)—a dedicated computer supporting the FR operation as part of a Vision Access Control System solution based on A4Vision technology.

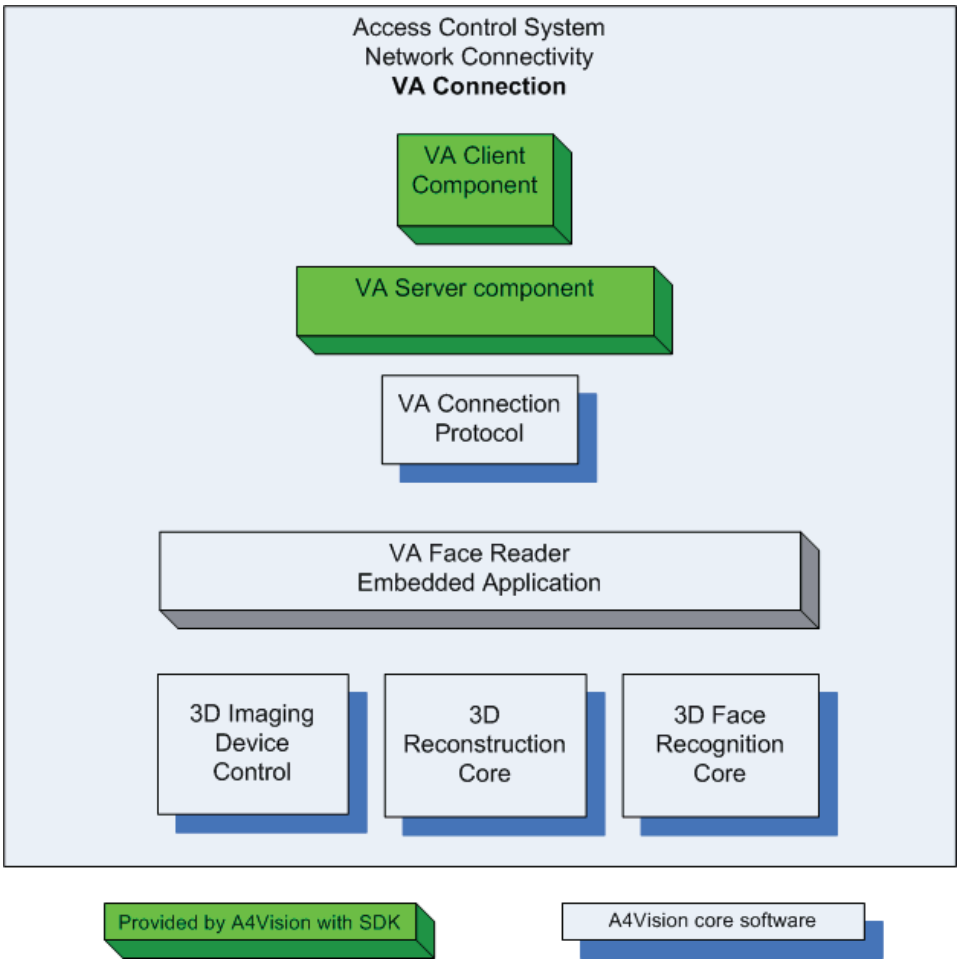

Fig. 2.3 FR network connectivity components (VA Connection).

At the heart of the VA Connection, as is the case with the VA SDK on the whole, you can find the COM (Component Object Model) technology, which is the VA Connection's foundation and its driving force. With the help of DCOM (extension of COM) components you can do the following:

⇒ Build custom ACS management software using C#, VB.NET, or C++.

⇒ Deploy face reader-based access control systems (ACSs) at end-user premises.

⇒ Enhance legacy access control systems with face readers (FRs).

⇒ Perform the technology evaluation with operational and scenario-driven testing, using several face readers within network.

A4Vision SDK is based on two components—VA Client (VACC) and VA Server (VASC).

VACC, a COM object, provides interface to the corresponding remote VASC, which is a DCOM object.

VA Server Component (VASC) serves to

⇒ Implement the server part of the VA Connection protocol to maintain local DBs (databases) on remote FRs, downloading persons visit statistics, and performing diagnostics.

⇒ Communicate with VALC (VA Licensing Component) to control licensing of VA SDK.

⇒ Communicate with VACC.

For detailed information on VA Client and Server Components, refer to sections 8.2 "Client Component" and 8.3 "Server Component", respectively.

To manage Face Readers (FRs) over a TCP/IP network, a proprietary application–level protocol, named VA Connection, is used. VASC uses the protocol to expose the FR management functionality as COM interface methods.

The FR support is implemented with the VA Face Reader Embedded application (FREA) running on the FR Controller computer (see section 12 "FR Embedded Application"). The FR Controller is delivered to a customer with the preinstalled the FREA. The application supports two modes of operations: standalone and host-driven.

Standalone mode operations:

- Verification with a person's ID stored on a card
- Verification with a person's biometric template (feature vector) and ID stored on a card
- Identification against the local list of biometric templates
- Visit result notification to Door Controller and the VASC host
- Visit statistics gathering.

Host-driven mode operations:

- Enrollment
- Verification
- Local biometric templates storage management
- Self-diagnostics
- Visit statistic loading
- Visit statistics clearing
- FR configuration update
- Embedded software update
- Reboot
- Getting logs.

For information on FREA configuration and maintenance, refer to section "FR Embedded Application." See also section 13.2 "FR Emulator" for information about how to use the FR Emulator that facilitates an application development.

2.4 Licensing Infrastructure

To use the VA SDK, one needs a license issued by A4Vision Inc. The license key is a small binary file used to configure the Sentinel LM license server. License permissions are checked by VA SDK modules at runtime. For more information, refer to section 15 "VA SDK Licensing."

VA Licensing Component (VALC) is a DCOM object that implements:

- Exposure of the interface to licensing limits checking
- Enforcement of the licensing scheme
- Management of requests for licenses from VASC, A4BSP, and VAEA.
- Communication with the Sentinel LM server.

Note

VASC (VA Server Component)—a DCOM object providing interface to FRs (face readers) through the network via the A4Vision proprietary protocol VA Connection.

A4BSP (A4Vision Biometric Service Provider)—a dynamic link library module supporting operations of A4Vision's 3D face recognition devices.

Sentinel LM—part of the SafeNet family of solutions providing electronic license management (see "Glossary General Terms").

2.5 Reference Design Sources

All the reference design sources provided with the current version of Vision Access™ SDK are built with Microsoft® Visual Studio® 7.1. Following is a brief description of the major features of the sources in use.

VA BioAPI sample application

- Fully operational code for managing the A4BSP module.
- A simple GUI and samples of operations Enroll, Verify, VerifyMatch, Identify and IdentifyMatch performed on a client PC equipped with a proper ES (Enrollment Station) device.
- Storage of enrolled templates into files.

VA Connection sample applications

- Fully operational examples of how to access and manage DCOM components (VASC) deployed on remote stations.
- Examples and a simple GUI implementing a full-fledged set of operations for managing FRs (Face Readers) bound into a network.

BirLoader sample application

- Source code samples and a simple GUI demonstrating how to upload FRs with biometrics data resulting by enrollment with the BioAPI sample application in use.

Three applications listed above complement each other. They are intended to demonstrate the functionality and interoperability of the VA SDK components, as well as the performance of biometric devices. The applications can be used to setup an access control pilot solution.

3 BIOAPI & VA Connection Data Exchange

Beginning with the release of VA SDK 2.0, the VA BioAPI components, as well as the VA Connection components, use the same biometric data format, which is a custom form of the BioAPI BIR.

The VA Connection part of the VA SDK gets all biometric data as continuous memory block with BIR_HEADER structure. Here is the memory layout:

Note

According to BioAPI Specification Version 1.1—the BioAPI framework, BIR is a container for biometric data. A BIR may contain raw sample data, partially processed (intermediate) data, or completely processed data. It may be used to enroll a user (thus being stored persistently), or may be used to verify or identify a user (thus being used transiently). The opaque biometric data is of variable length, and may be followed by a signature. The signature itself may not be a fixed length, depending on which signature standard is employed. The signature is calculated on the combined Header and BiometricData.

The BioAPI framework returns biometric data as a structure of the BioAPI_BIR type. These data should be realigned before passing to the VA Connection components. The realigning mechanism is very simple: all opaque biometric data should be placed after the BioAPI_BIR_HEADER structure.

The code sample shown below performs actual converting.

3.1 VA BioAPI to VA Connection data type conversion

For converting data of the VA BioAPI type to that of the VA Connection type, do as follows:

1. Calculate a biometric data size.
2. Allocate memory as appropriate.
3. Place a copy of the BioAPI BIR_HEADER structure at the beginning of the allocated memory.
4. Place a copy of the BIR.BiometricData structure after the BioAPI_BIR_HEADER structure.

The source code below is an example of VA BioAPI to VA Connection data type conversion.

```
// Source code in C++
void* BioAPI2Connection(BioAPI_BIR &bir) {
// Calculate the biometric data size.
```

```
 const unsigned biometricDataSize = bir.Header.Length;

// Allocate some memory.
unsigned char* bioData = new unsigned char [biometricDataSize];
unsigned char* bioOpaqueData = bioData + sizeof (BioAPI_BIR_HEADER);

 // Copy the header and biometric data.
memcpy(bioData, &bir, sizeof(BioAPI_BIR_HEADER));
memcpy(bioOpaqueData, bir.BiometricData,
        biometricDataSize - sizeof (BioAPI_BIR_HEADER));
return bioData;
}
// End of source code
```

3.2 VA Connection to VA BioAPI data type conversion

For converting data of the VA Connection type to that of the VA BioAPI type, do as follows:

1. Create a BioAPI_BIR structure.
2. Copy the BioAPI_BIR_HEADER structure from a VA Connection data to the newly created structure.
3. Calculate the size of the opaque biometric data.
4. Allocate some memory for the opaque biometric data, and then set up the BioAPI_BIR.BiometricData field
5. Copy the opaque biometric data to the BioAPI_BIR.BiometricData field.
6. Set the Signature field to NULL.

The source code below is an example of the VA Connection to VA BioAPI data type conversion.

```
// Source code in C++
BioAPI_BIR* Connection2BioAPI(void* bioData) {
// Create a BioAPI_BIR structure.
BioAPI_BIR* bir = new BioAPI_BIR;

// Copy BioAPI_BIR_HEADER from VA Connection data to the newly created structure.
memcpy(&bir->Header, bioData, sizeof (BioAPI_BIR_HEADER));
// Some part of the code is deleted as irrelevant for sample writing demonstration purposes.
// Copy the opaque biometric data to the BioAPI_BIR.BiometricData structure.
memcpy(bir->BiometricData, opaqueData, biometricDataSize);
// Set up the signature.
 bir->Signature = NULL;
 return bir;
}
// End of source cod
```