COMMON API FOR UIDAI CERTIFIED BIOMETRIC DEVICES

OBJECTIVE

Discussion on Common API methods

After successful implementation of the Common API for Android and Windows (.NET and Java) platforms, it is intended to be extended to other platforms.

Development to be taken up for platforms like **Linux, MacOS, Windows Mobile and iOS**.

Development of wrapper API’s ,native libraries and drivers by vendors for these platforms.

COMMON API

The Common API for biometric devices was envisioned to provide :

- Plug n play based support for UIDAI certified biometric devices.
- Easy integration of devices across multiple vendors with applications
- Easy Application development
- Facilitate Biometric Authentication based applications
- Universal API for both Fingerprint and Iris scanners

BRIEF OVERVIEW

The Common API has a **BiometricDeviceHandler** class which encapsulates all the biometric device related implementation.

The class is an abstract view of the vendors implementation of the Common API.

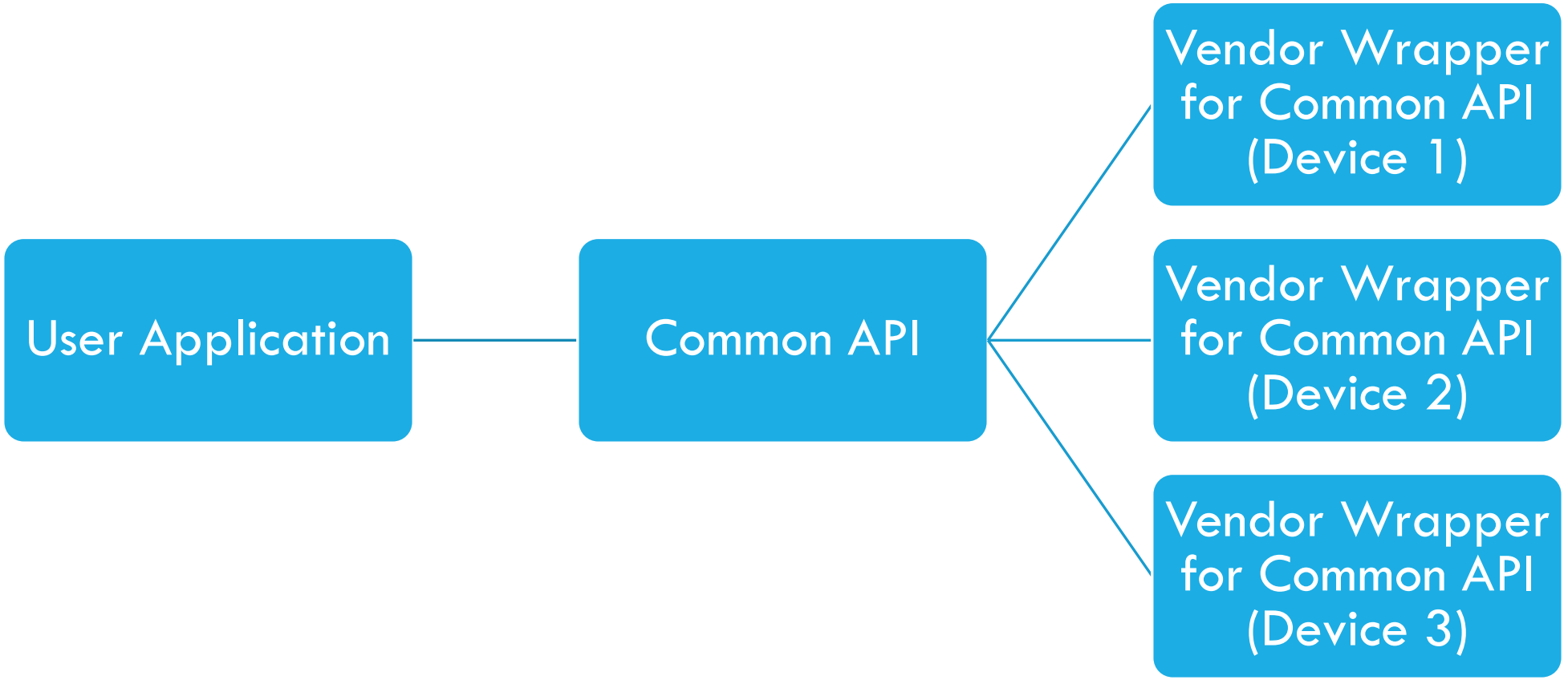
Class implementation is exactly the same for all supported biometric devices.

The biometric device class is able to sense the attached device if it is among the supported devices.

The API currently supports Android and Windows (Java and .NET)

Next phase of development focuses on developing the API for **Linux, MacOS, Windows Mobile and iOS**

COMMON API ARCHITECTURE AT A GLANCE



COMMON API CLASS AND METHODS REFERENCE

1. **BiometricDeviceHandler handler = new BiometricDeviceHandler(this, BiometricDeviceId);**

This class contains all the methods in common api required by the user application. The class is instantiated with the biometric device id and the current object.

After the handler is instantiated all methods in the common api are available for use in the application.

CONNECTED DEVICE DISCOVERY MECHANISM

Android

- In case of Android the device id fetched from the **usb broadcast** receiver event. The **pid** and vid are matched from the list of available devices in common **api**. Id the device id matches in the list, it is instantiated.

Windows

- In case of windows, **polling** method is used to fetch the connected device information. All devices in the common **api** list are initialized and the device which return 0 (success) is treated as the connected device.

METHODS AVAILABLE IN COMMON API

2. int ret = handler.InitDevice(BiometricDeviceId);

- This method initializes the device whose device id is passed as parameter.
- Return value = 0 (zero) means success
- Non zero return value means failure

3. int ret = handler.UnInitDevice();

- This method uninitializes the connected device
- Return value = 0 (zero) means success
- Non zero return value means failure

METHODS AVAILABLE IN COMMON API (CONTINUED)

4. String ret = handler.GetDeviceSerialNumber();

- This method returns the serial number of the connected device. The serial number returned should match with the serial number at the back of the device

5. Int ret = handler.GetDeviceType();

- Return the device type of the attached device as integer value
- Return value = 0 if device is a fingerpint device
- Return value = 1 if device is an iris device

METHODS AVAILABLE IN COMMON API (CONTINUED)

6. **Int ret = handler.GetAttachedDeviceVendorID();**
   - Returns the vendor id of the attached device

7. **String ret = handler.GetDeviceMake();**
   - Returns the make of the attached device

8. **String ret = handler.GetDeviceModel();**
   - Returns the model of the attached device

METHODS AVAILABLE IN COMMON API (CONTINUED)

9. **handler.SetCaptureTimeout(CaptureTimeout);**
   - Sets the timeout duration for biometric capture (in milliseconds)

10. **handler.SetThresholdQuality(ThresholdQuality);**
    - Sets the threshold value of the biometric quality in percentage
    - Value must be between 1 and 100

11. **handler.SetFingerprintBiometricDataType(BiometricDataType);**
    - Sets the biometric data type
    - Possible values are FMR, FIR and IIR

METHODS AVAILABLE IN COMMON API (CONTINUED)

12. **void handler_WrapperCallHandler(byte[ ] rawData, int height, int width, int status, string errorMessage, bool complete, byte[ ] isoData, int quality, int finalNFIQ)**

- Retrieves the biometric data during and after successful capture
- During capture the **status** value should be 0 (zero)
- After successful capture the **complete** value should be **true** and **status** = 0
- After capture is successful the **isoData** is used as biometric data for AuthXML by the user application

DEVICES INTEGRATED INTO COMMON API (ANDROID)

Sl. No.	Device Name	Device ID		Sl. No.	Device Name	Device ID
1.	Mantra MFS100	31		9.	Morpho MSO 35	38
2.	Startek FM220	33312		10.	Biomatiques EPI-1000	1003
3.	Bioenable HFDU08	1616		11.	Precision UareU 4500	11
4.	Secugen PRO20	5		12.	Precision Eikon Touch	2016
5.	Irishield MK 2120U	1002				
6.	Morpho MSO 1300	71				
7.	Morpho MSO 1350	82				
8.	Morpho MSO 30	36				

DEVICES INTEGRATED INTO COMMON API (.NET)

Sl. No.	Device Name	Device ID		Sl. No.	Device Name	Device ID
1.	Mantra MFS100	31		9.	Morpho MSO 30	36
2.	Startek FM220	33312		10.	Morpho MSO 35	38
3.	Bioenable HFDU08	1616		11.	Anaxee FS88	88
4.	Precision CSD200	2140		12.	Biomatiques EPI-1000	1003
5.	Secugen PRO20	5		13.	Precision FM220	2141
6.	Irishield MK 2120U	1002		14.	Integra FS88	8888
7.	Morpho MSO 1300	71		15.	Precision UareU 4500	11
8.	Morpho MSO 1350	82		16.	Precision Eikon Touch	2016

DEVICES INTEGRATED INTO COMMON API (JAVA)

Sl. No.	Device Name	Device ID		Sl. No.	Device Name	Device ID
1.	Mantra MFS100	31		9.	Morpho MSO 35	38
2.	Startek FM220	33312		10.	Anaxee FS88	88
3.	Bioenable HFDU08	1616		11.	Biomatiques EPI-1000	1003
4.	Precision CSD200	2140		12.	Precision FM220	2141
5.	Secugen PRO20	5		13.	Precision UareU 4500	11
6.	Morpho MSO 1300	71		14.	Precision Eikon Touch	2016
7.	Morpho MSO 1350	82				
8.	Morpho MSO 30	36				

APPLICATIONS INCORPORATING COMMON API

- Jeevan Pramaan
- BAS (Biometric Attendance System – Govt. of India)
- AEBAS (Aadhaar Enabled Biometric Attendance System – Govt. of Jharkhand.
- Sarathi
- Vahan

THANK YOU