BSA Sandbox Bootcamp

ITU Regional DFS Security Clinic Seoul

Blockchain Secure Authentication (BSA) SDK Implementation for Native Apps

Mobile SDK Walkthrough

Passwordless Blockchain Technology

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

FNSvalue

Table of Content

An overview content of the slide

01 BSA Technology Overview

02 SDK Overview

03 SDK UI Component & Functionality

04 SDK Implementation for aOS

05 SDK Implementation for iOS

06 Testing SDK Integration & Troubleshooting

07 Best Practices

08 Demo App/Use Case

09 Resources

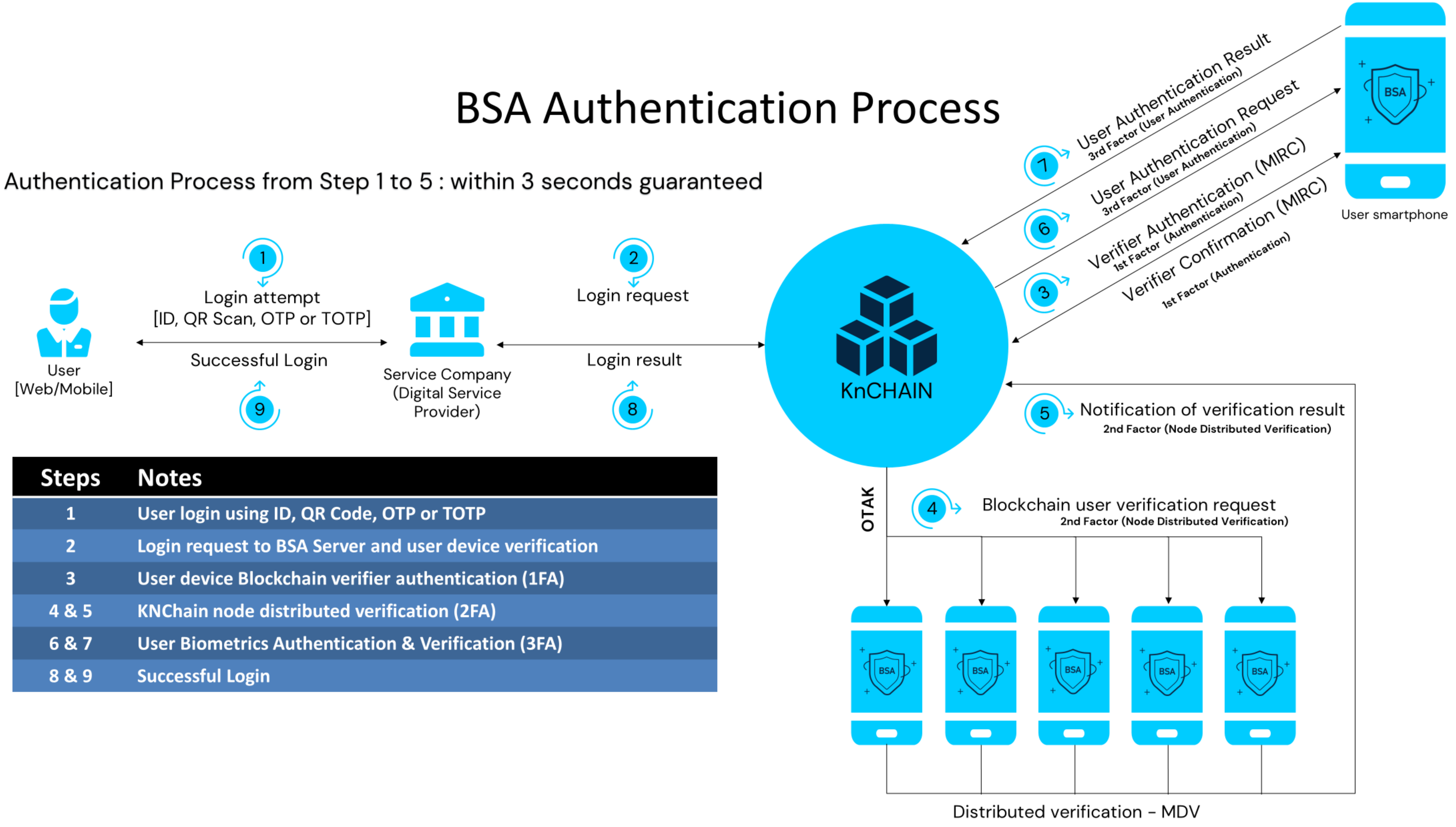
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

BSA Authentication Process

Authentication Process from Step 1 to 5 : within 3 seconds guaranteed

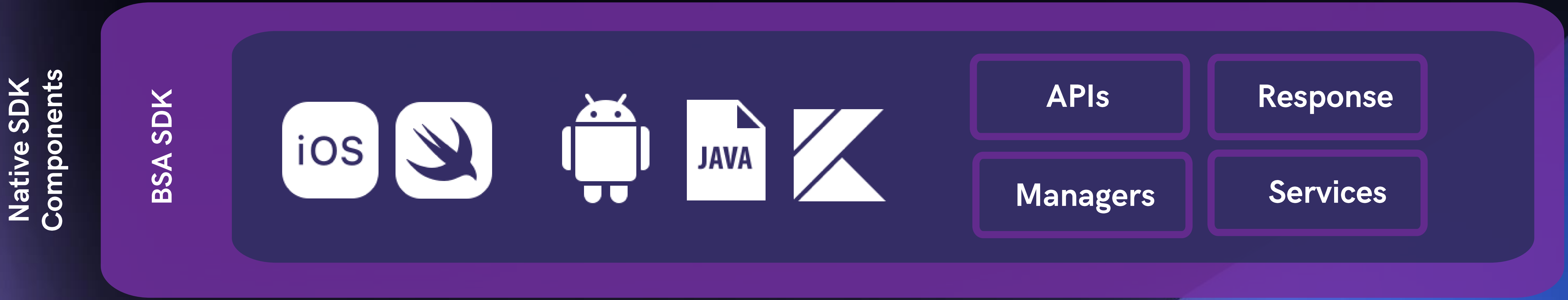
Steps	Notes
1	User login using ID, QR Code, OTP or TOTP
2	Login request to BSA Server and user device verification
3	User device Blockchain verifier authentication (1FA)
4 & 5	KNChain node distributed verification (2FA)
6 & 7	User Biometrics Authentication & Verification (3FA)
8 & 9	Successful Login


1
Login attempt
[ID, QR Scan, OTP or TOTP]
Successful Login
9
User
[Web/Mobile]
Service Company
(Digital Service
Provider)
2
Login request
Login result
8
KnCHAIN
7
User Authentication Result
3rd Factor (User Authentication)
6
User Authentication Request
3rd Factor (User Authentication)
3
Verifier Authentication (MIRC)
1st Factor (Authentication)
Verifier Confirmation (MIRC)
1st Factor (Authentication)
BSA
User smartphone
5
Notification of verification result
2nd Factor (Node Distributed Verification)
OTAK
4
Blockchain user verification request
2nd Factor (Node Distributed Verification)
BSA
BSA
BSA
BSA
BSA
Distributed verification – MDV

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

BSA SDK is a software development kit, equips developers with the necessary tools to integrate BSA authentication into their native applications

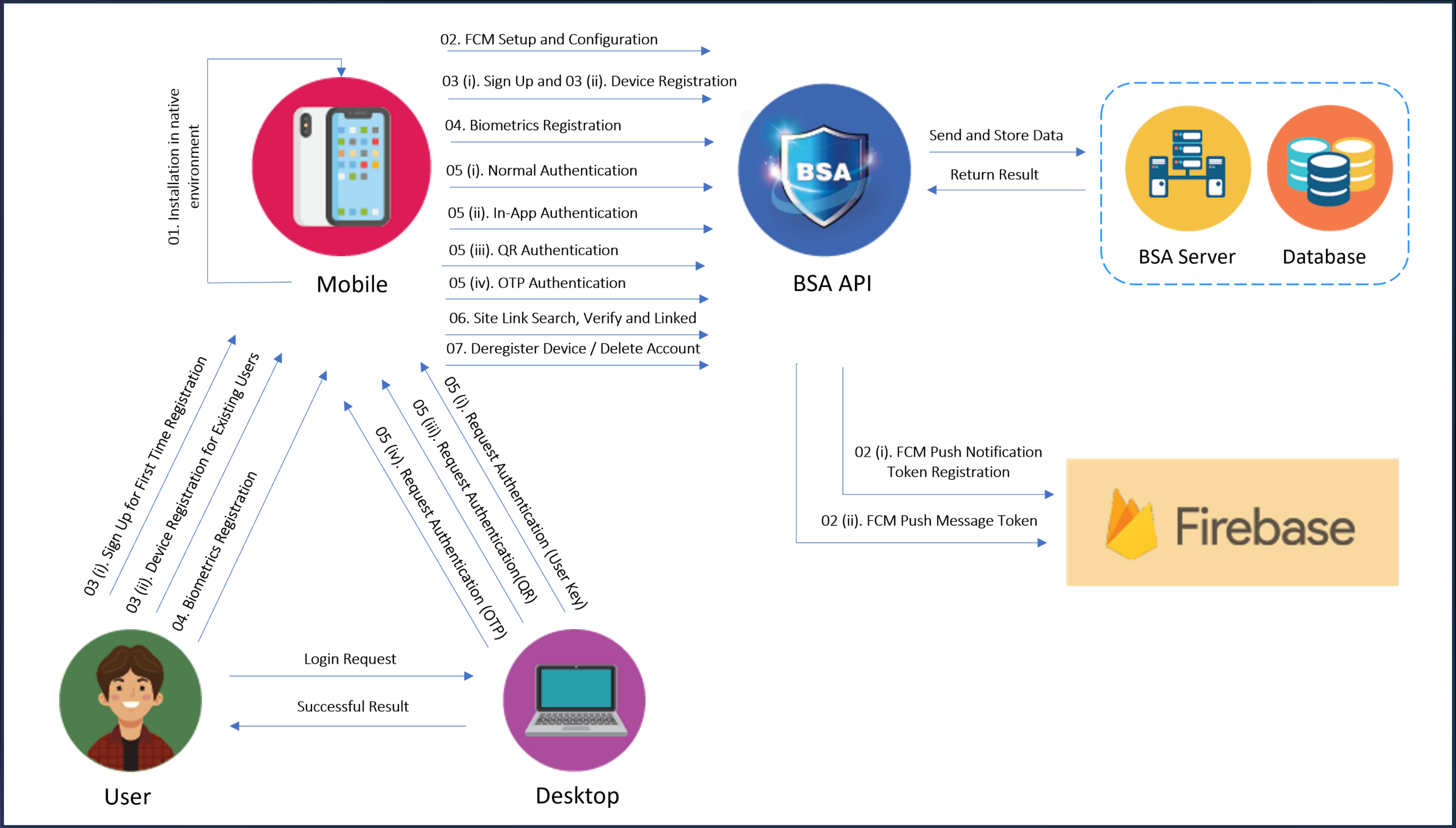


The goals is to implement Blockchain Secure Authentication Technology as user identifier and verification using Passwordless Blockchain, hence prevention security issues

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

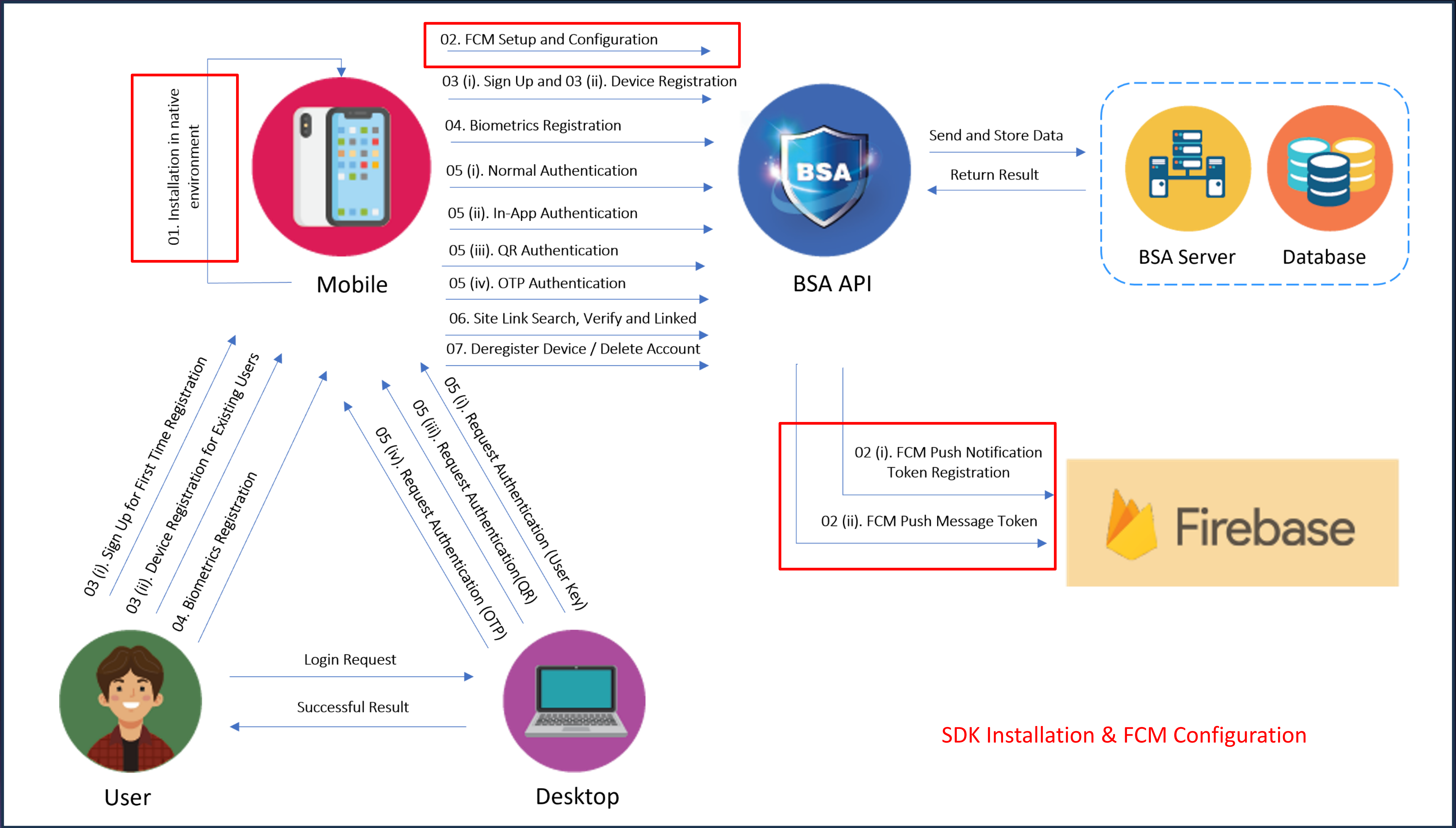
03. SDK UI Component & Functionality


02. FCM Setup and Configuration
03 (i). Sign Up and 03 (ii). Device Registration
04. Biometrics Registration
05 (i). Normal Authentication
05 (ii). In-App Authentication
05 (iii). QR Authentication
05 (iv). OTP Authentication
06. Site Link Search, Verify and Linked
07. Deregister Device / Delete Account
01. Installation in native environment
Mobile
BSA
BSA API
Send and Store Data
Return Result
BSA Server
Database
02 (i). FCM Push Notification Token Registration
02 (ii). FCM Push Message Token
Firebase
03 (i). Sign Up for First Time Registration
03 (ii). Device Registration for Existing Users
04. Biometrics Registration
05 (i). Request Authentication (User Key)
05 (iii). Request Authentication(QR)
05 (iv). Request Authentication (OTP)
Login Request
Successful Result
User
Desktop

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

03. SDK UI Component & Functionality

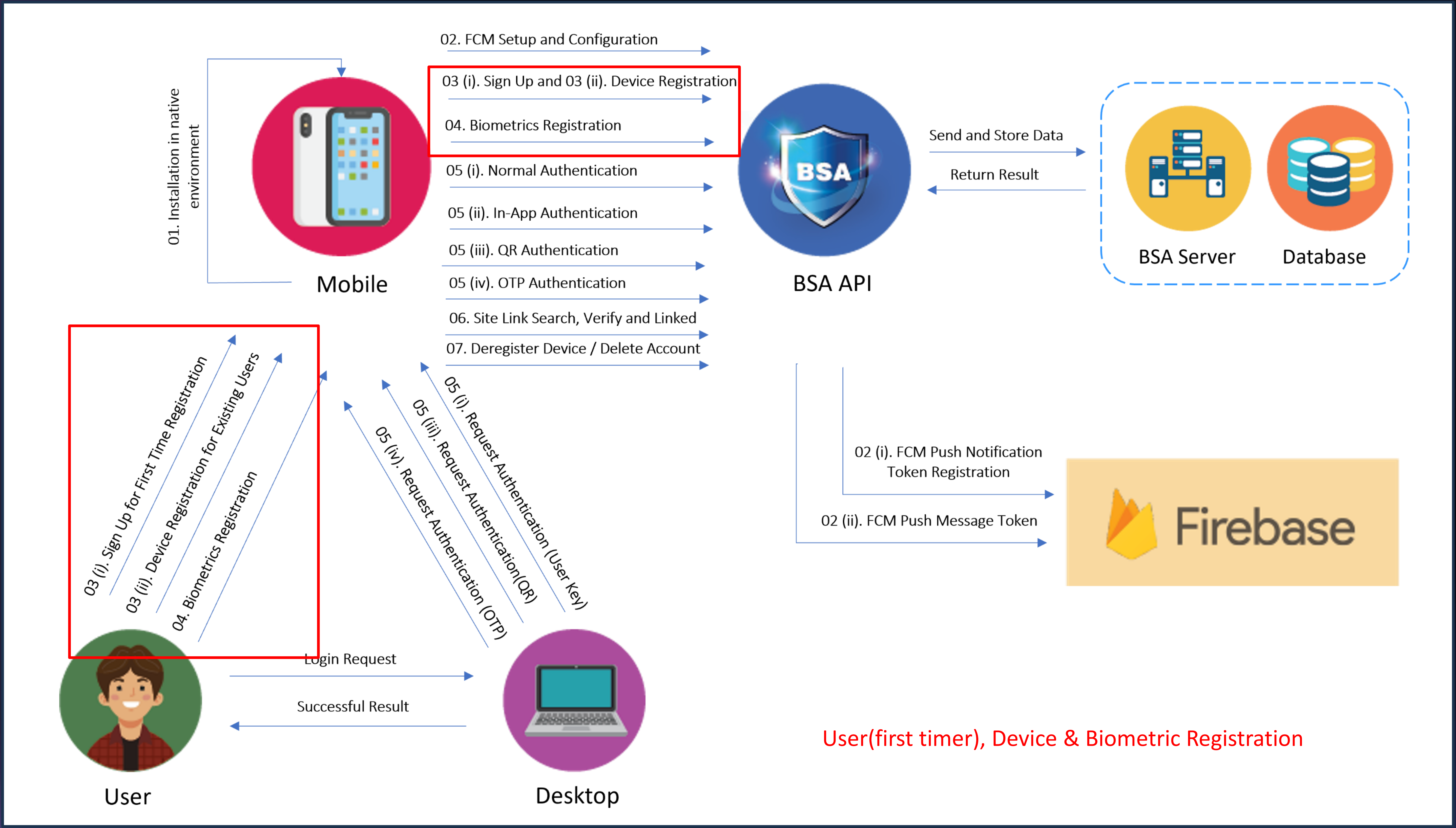

01. Installation in native environment
02. FCM Setup and Configuration
03 (i). Sign Up and 03 (ii). Device Registration
04. Biometrics Registration
05 (i). Normal Authentication
05 (ii). In-App Authentication
05 (iii). QR Authentication
05 (iv). OTP Authentication
06. Site Link Search, Verify and Linked
07. Deregister Device / Delete Account
Mobile
BSA
BSA API
Send and Store Data
Return Result
BSA Server
Database
02 (i). FCM Push Notification Token Registration
02 (ii). FCM Push Message Token
Firebase
03 (i). Sign Up for First Time Registration
03 (ii). Device Registration for Existing Users
04. Biometrics Registration
05 (i). Request Authentication (User Key)
05 (iii). Request Authentication(QR)
05 (iv). Request Authentication (OTP)
Login Request
Successful Result
User
Desktop

SDK Installation & FCM Configuration

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

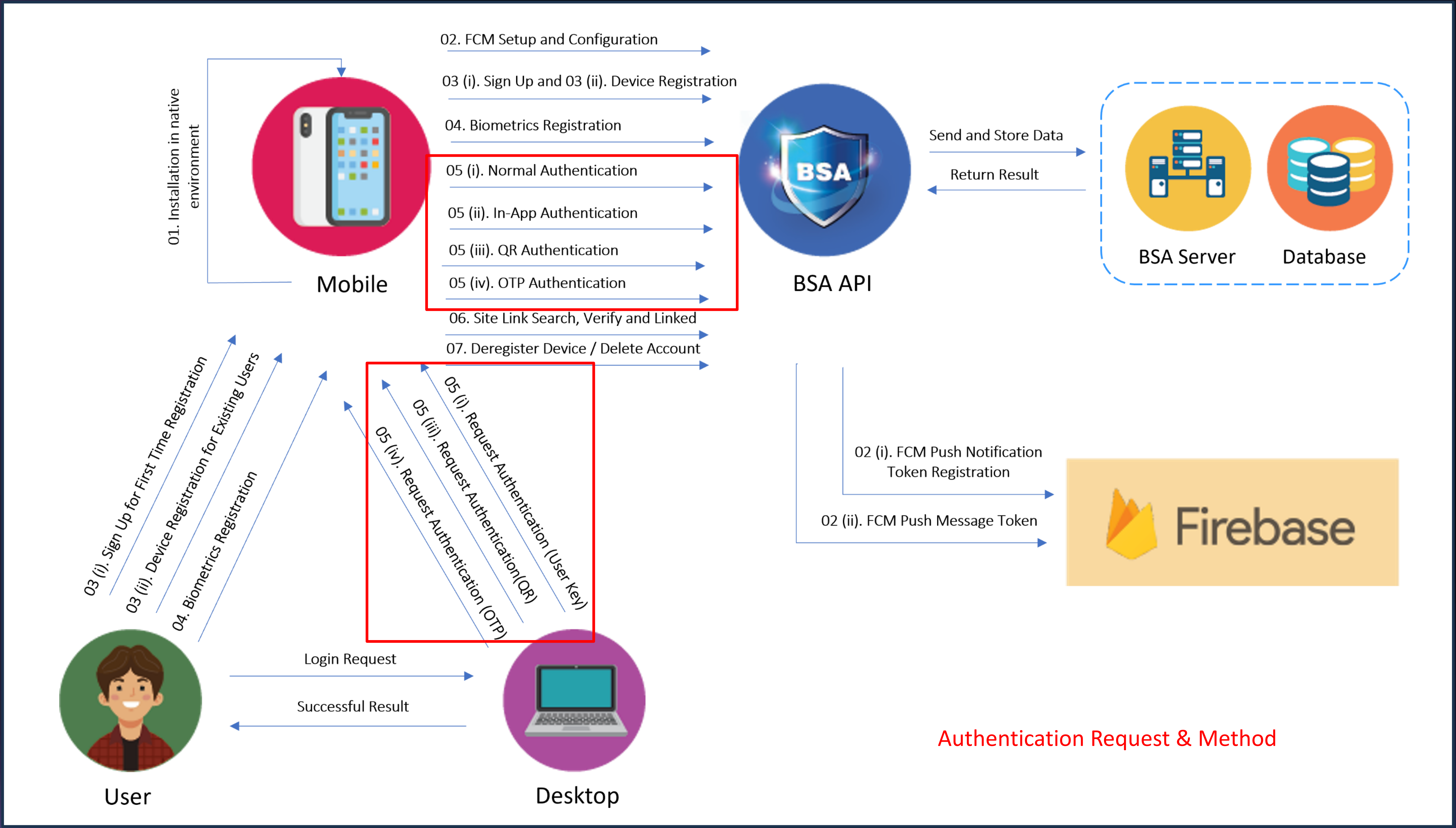
03. SDK UI Component & Functionality


01. Installation in native environment
Mobile
02. FCM Setup and Configuration
03 (i). Sign Up and 03 (ii). Device Registration
04. Biometrics Registration
05 (i). Normal Authentication
05 (ii). In-App Authentication
05 (iii). QR Authentication
05 (iv). OTP Authentication
06. Site Link Search, Verify and Linked
07. Deregister Device / Delete Account
BSA
BSA API
Send and Store Data
Return Result
BSA Server
Database
03 (i). Sign Up for First Time Registration
03 (ii). Device Registration for Existing Users
04. Biometrics Registration
05 (i). Request Authentication (User Key)
05 (iii). Request Authentication(QR)
05 (iv). Request Authentication (OTP)
02 (i). FCM Push Notification Token Registration
02 (ii). FCM Push Message Token
Firebase
Login Request
Successful Result
User
Desktop
User(first timer), Device & Biometric Registration

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

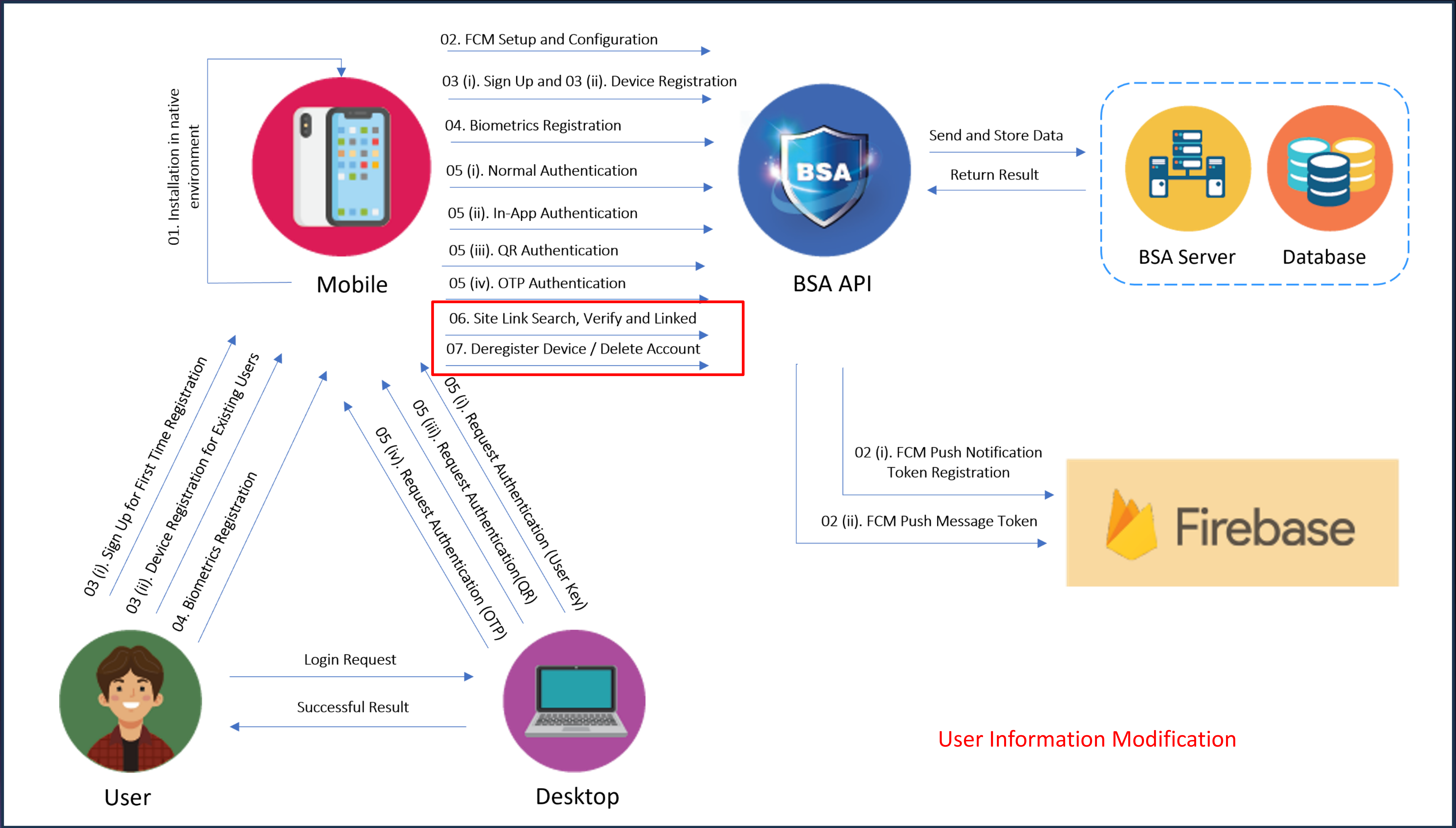
03. SDK UI Component & Functionality


02. FCM Setup and Configuration
03 (i). Sign Up and 03 (ii). Device Registration
04. Biometrics Registration
05 (i). Normal Authentication
05 (ii). In-App Authentication
05 (iii). QR Authentication
05 (iv). OTP Authentication
06. Site Link Search, Verify and Linked
07. Deregister Device / Delete Account
01. Installation in native environment
Mobile
BSA
BSA API
Send and Store Data
Return Result
BSA Server
Database
02 (i). FCM Push Notification Token Registration
02 (ii). FCM Push Message Token
Firebase
03 (i). Sign Up for First Time Registration
03 (ii). Device Registration for Existing Users
04. Biometrics Registration
05 (i). Request Authentication (User Key)
05 (iii). Request Authentication(QR)
05 (iv). Request Authentication (OTP)
Login Request
Successful Result
User
Desktop
Authentication Request & Method

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

03. SDK UI Component & Functionality


01. Installation in native environment
Mobile
02. FCM Setup and Configuration
03 (i). Sign Up and 03 (ii). Device Registration
04. Biometrics Registration
05 (i). Normal Authentication
05 (ii). In-App Authentication
05 (iii). QR Authentication
05 (iv). OTP Authentication
06. Site Link Search, Verify and Linked
07. Deregister Device / Delete Account
BSA
BSA API
Send and Store Data
Return Result
BSA Server
Database
02 (i). FCM Push Notification Token Registration
02 (ii). FCM Push Message Token
Firebase
03 (i). Sign Up for First Time Registration
03 (ii). Device Registration for Existing Users
04. Biometrics Registration
05 (i). Request Authentication (User Key)
05 (iii). Request Authentication(QR)
05 (iv). Request Authentication (OTP)
Login Request
Successful Result
User
Desktop

User Information Modification

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

03. SDK UI Component & Functionality

SECTION		UI COMPONENT	SDK FUNCTIONALITY
User Registration		Input Field for user ID, Name, Email and Phone Number	1. Duplicate Check for these input 2. registerUser() to register the user information
Authentication Method		1. Button QR Scanner, TOTP Request and OTP Request for authentication method 2. Button “Cancel” for cancel authentication process	Each type of Authentication method called
User Information Modification	Change User Authentication Type	Button with 3 different user authentication type (Normal Authentication, Biometric/Face ID Authentication and PIN/Password Authentication)	1. resetBiometricChange() to reset previous User Authentication Type Infformation 2. registerBiometric() or authDeviceCredential() to re-register the update User Authentication Type
	De-Register Device	Button “De-Register Device” or “Sign Out”	1. unRegisterDevice() to de-register device 2. reRegisterUserDevice() to re-register the device again
	Delete Account	Button “Delete Account”	1. deleteUser() 2. Once this function being called, all the user information in the database will be deleted permanently

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Prerequisites:

	Minimum	Target
Gradle	Version 6.0.0	Version 8.1.2
Android Version	Android 6.0 Marshmallow (API Level 23)	Android 13 Marshmallow (API Level 33)

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PRO HIBITED.

Installation :

Step 1

Declare BSA SDK repository in the Gradle project level

Step 2

Add dependencies required on Gradle app level

Step 3

Add permission configuration on the Manifest

Step 4

Add JAVA 8 configuration at Gradle App Level

Step 5

Initialization the SDK by adding the Client Key and API Server URL

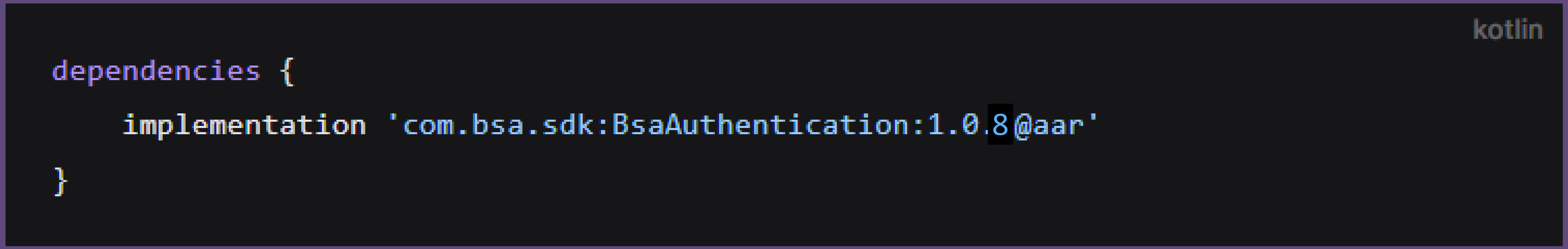
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation :

Step 1

Declare BSA SDK repository in the Gradle project level :



```kotlin
dependencies {
    implementation 'com.bsa.sdk:BsaAuthentication:1.0.8@aar'
}
```

For latest version update, kindly refer here

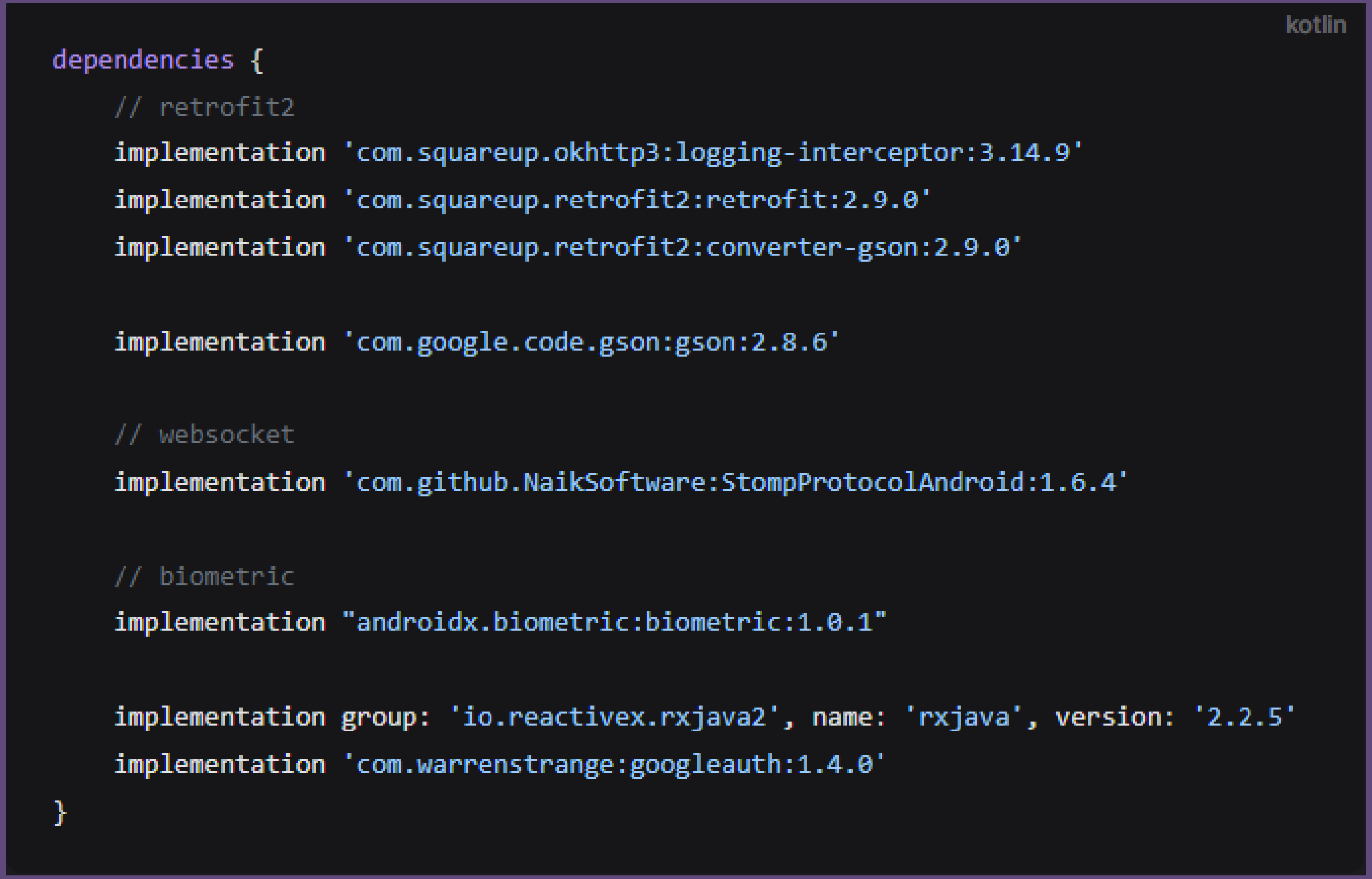
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation :

Step 2

Add dependencies required on Gradle app level :



kotlin

```kotlin
dependencies {
    // retrofit2
    implementation 'com.squareup.okhttp3:logging-interceptor:3.14.9'
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

    implementation 'com.google.code.gson:gson:2.8.6'

    // websocket
    implementation 'com.github.NaikSoftware:StompProtocolAndroid:1.6.4'

    // biometric
    implementation "androidx.biometric:biometric:1.0.1"

    implementation group: 'io.reactivex.rxjava2', name: 'rxjava', version: '2.2.5'
    implementation 'com.warrenstrange:googleauth:1.4.0'
}
```

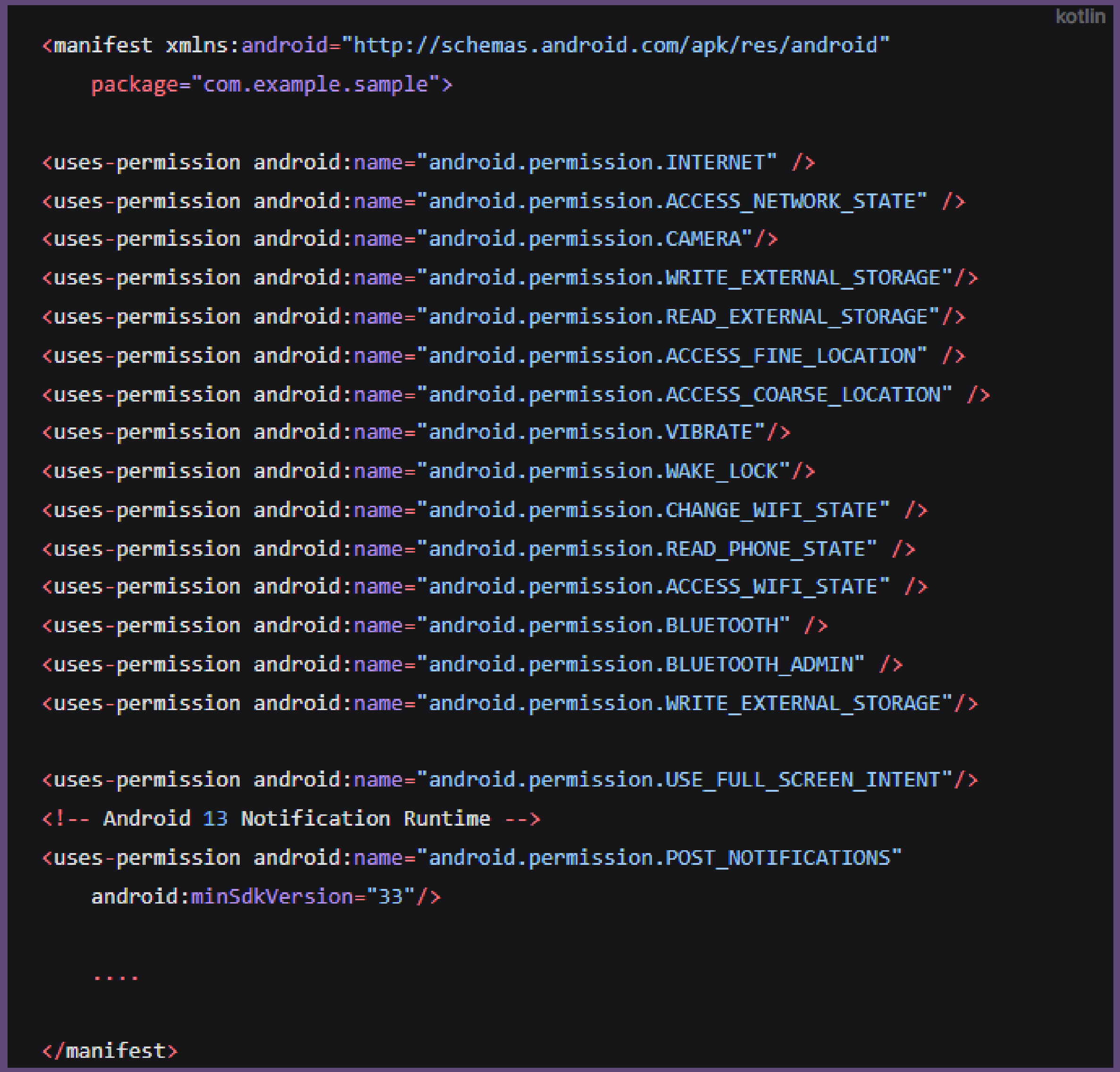
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation :

Step 3

Add permission configuration on the Manifest :



kotlin

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sample">

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<uses-permission android:name="android.permission.USE_FULL_SCREEN_INTENT"/>
<!-- Android 13 Notification Runtime -->
<uses-permission android:name="android.permission.POST_NOTIFICATIONS"
    android:minSdkVersion="33"/>


    ....


</manifest>
```

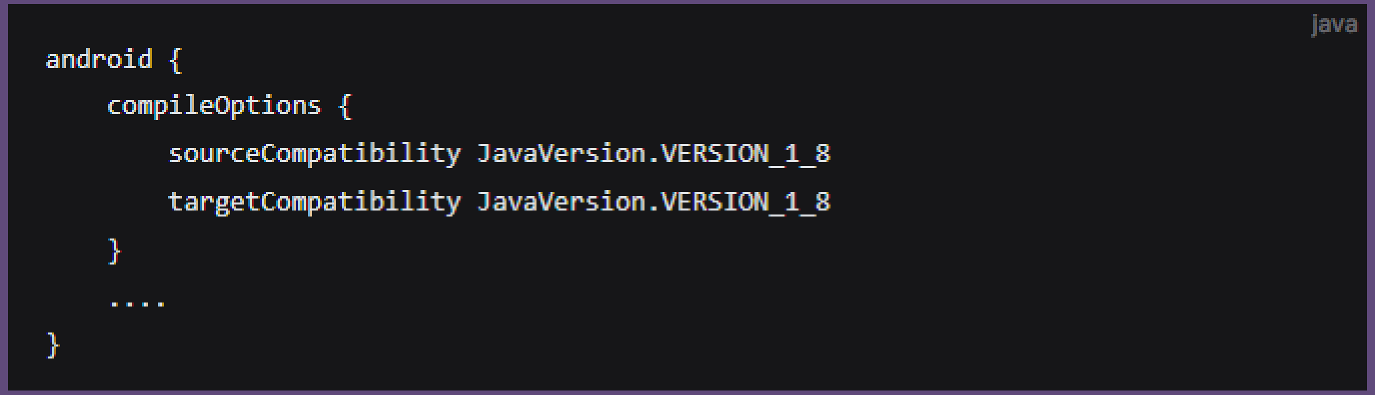
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation :

Step 4

Add JAVA 8 configuration at Gradle App Level :



```
android {
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    ....
}
```

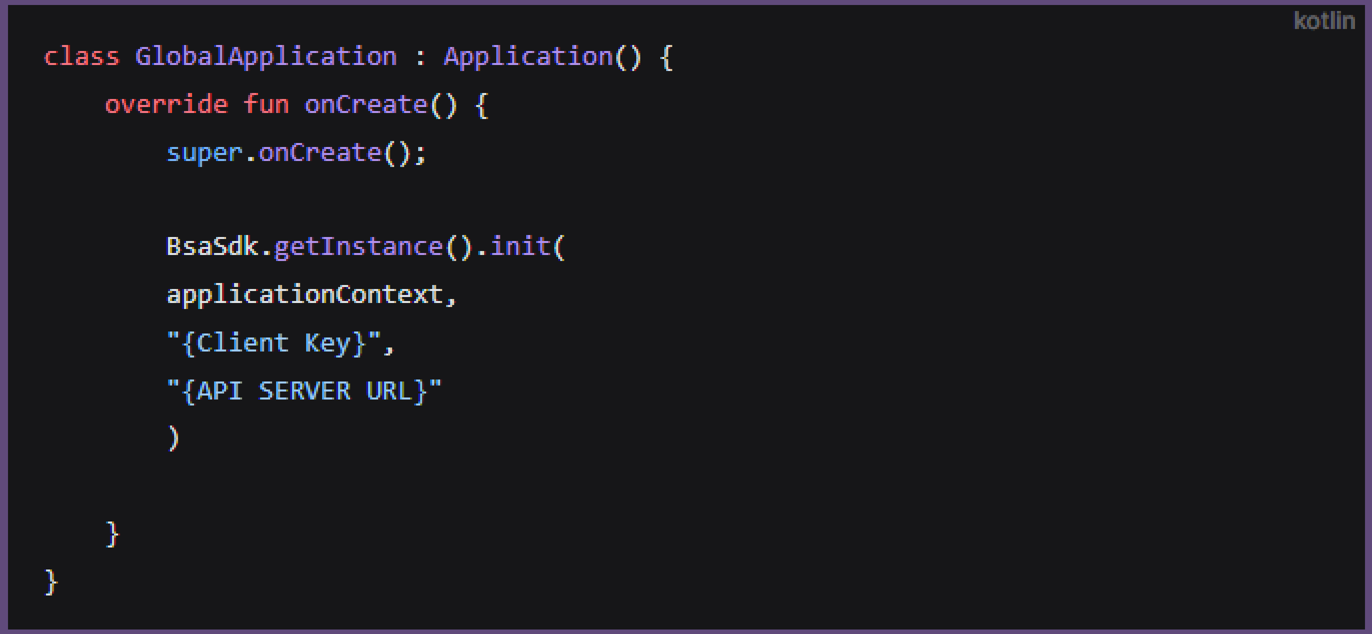
COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation :

Step 5

Initialization the SDK by adding the Client Key and API Server URL :

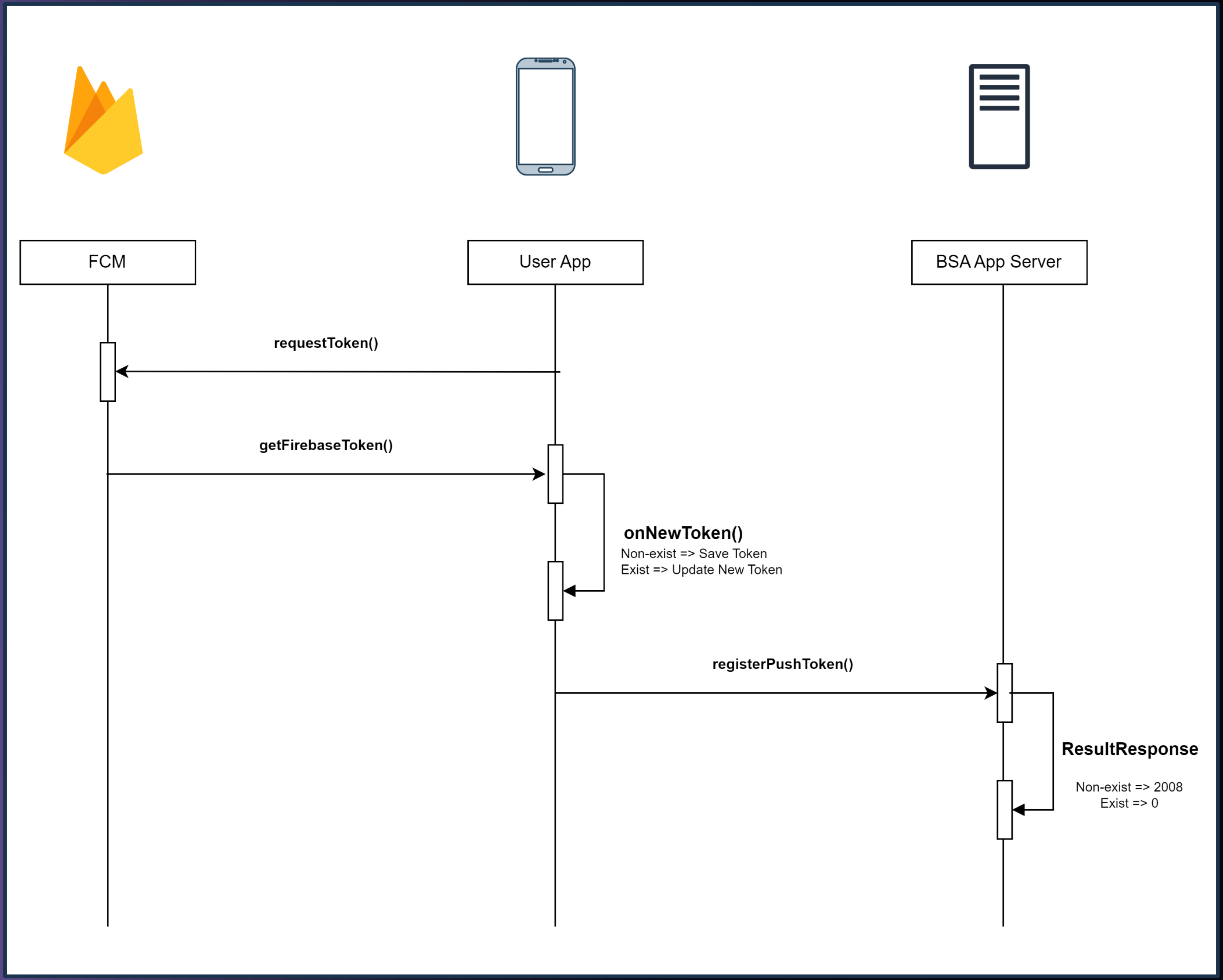


```kotlin
class GlobalApplication : Application() {
    override fun onCreate() {
        super.onCreate();

        BsaSdk.getInstance().init(
        applicationContext,
        "{Client Key}",
        "{API SERVER URL}"
        )

    }
}
```

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

FCM Setup & Configuration :



1. Create a Google Cloud Project:

   Requester should start by creating a Google Cloud project to configure Firebase services, including Firebase Cloud Messaging.

2. Generate google-services.json:

   After creating the project, requester need to set up Firebase within their application by generating a google-services.json file. This file contains crucial configuration details specific to the Firebase project.

3. Share google-services.json:

   Requester must then share the google-services.json file with the ITU team responsible for managing the BSA App Server.

4. Configure FCM on BSA App Server:

   The ITU team will utilize the provided google-services.json file to configure Firebase Cloud Messaging on the BSA App Server. This involves setting up the necessary credentials to facilitate the sending of notifications to the requester's application.

5. Testing Notification Delivery:

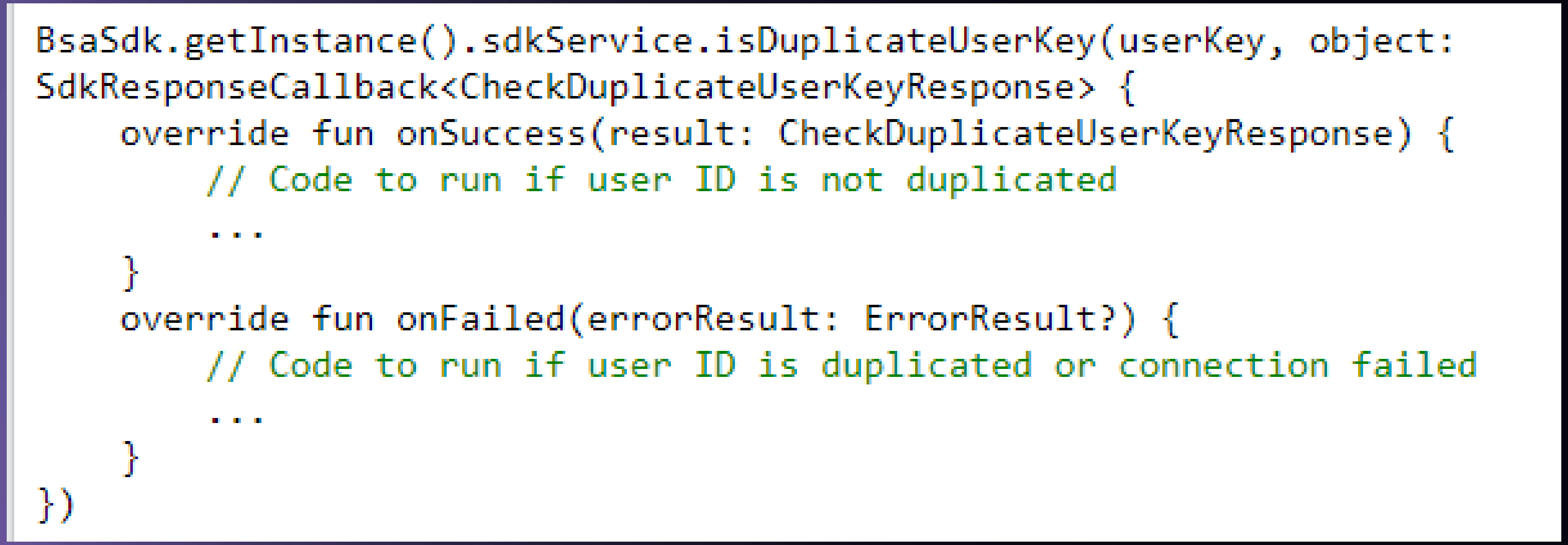
   Once the configuration is complete, requester can proceed to test whether they receive notifications from the BSA App Server. This testing phase includes sending a test notification from the server to the requester's application and verifying its successful reception.

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

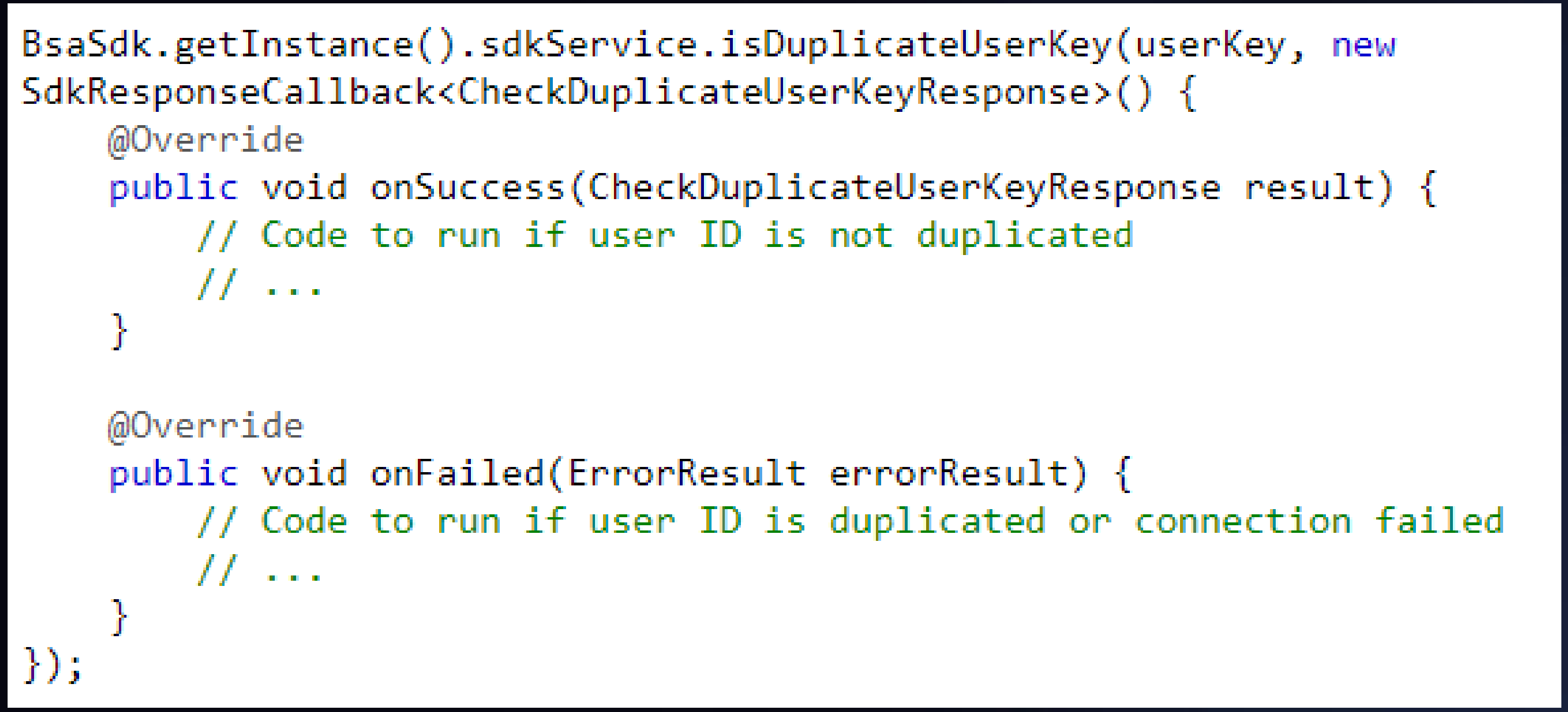
SDK Integration :

Kotlin



```kotlin
BsaSdk.getInstance().sdkService.isDuplicateUserKey(userKey, object:
SdkResponseCallback<CheckDuplicateUserKeyResponse> {
    override fun onSuccess(result: CheckDuplicateUserKeyResponse) {
        // Code to run if user ID is not duplicated
        ...
    }
    override fun onFailed(errorResult: ErrorResult?) {
        // Code to run if user ID is duplicated or connection failed
        ...
    }
})
```

Java



```java
BsaSdk.getInstance().sdkService.isDuplicateUserKey(userKey, new
SdkResponseCallback<CheckDuplicateUserKeyResponse>() {
    @Override
    public void onSuccess(CheckDuplicateUserKeyResponse result) {
        // Code to run if user ID is not duplicated
        // ...
    }

    @Override
    public void onFailed(ErrorResult errorResult) {
        // Code to run if user ID is duplicated or connection failed
        // ...
    }
});
```

BsaSdk's isDuplicateUserKey() to check if there is a duplicate user ID.

Param : userKey

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Prerequisites:

	Version
xCode	14.0 and above
iOS	14.0 and above
CocoaPods	1.14.3 and above

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

Installation with Cocoapods :

Step 1

If you don’t have Podfile yet, you can create it using command “pod init ”

Step 2

Open Podfile using command “open Podfile”

Step 3

Open Target file using command “ target `projectbsasdk (example target name)’ do “

Step 4

Add BSA SDK (with correct version and source) and Alamofire as dependency inside the target file

Step 5

Add necessary build settings using command “post_install do”

Step 6

Go to terminal, and do “pod install”

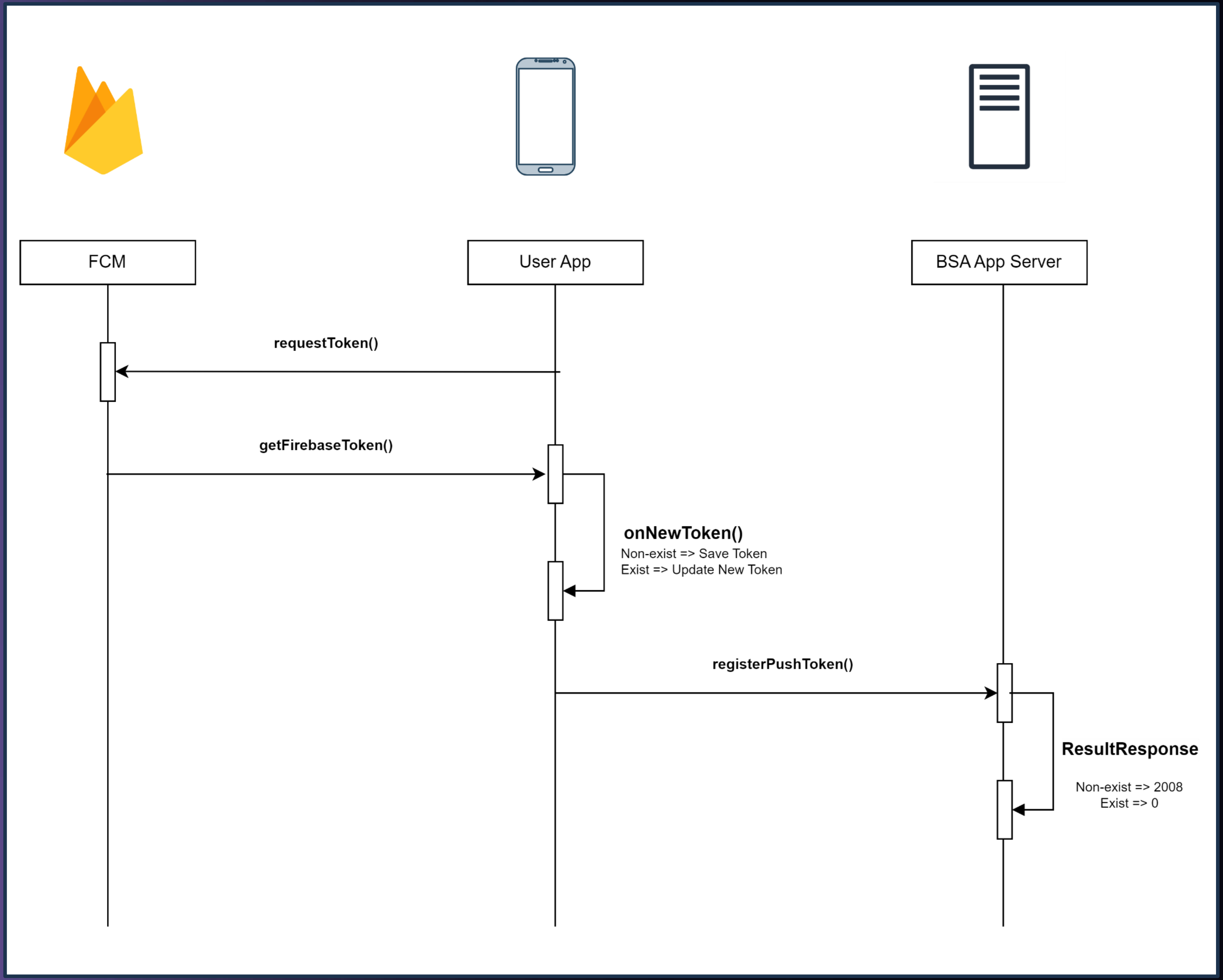
Step 7

Initialize BSA SDK in AppDelegate.swift with serverUrl, websocketUrl, securityKey and removeToken as their parameters

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

FCM Setup & Configuration :



1. Create a Google Cloud Project:

   Requester should start by creating a Google Cloud project to configure Firebase services, including Firebase Cloud Messaging.

2. Generate google-services.json:

   After creating the project, requester need to set up Firebase within their application by generating a google-services.json file. This file contains crucial configuration details specific to the Firebase project.

3. Share google-services.json:

   Requester must then share the google-services.json file with the ITU team responsible for managing the BSA App Server.

4. Configure FCM on BSA App Server:

   The ITU team will utilize the provided google-services.json file to configure Firebase Cloud Messaging on the BSA App Server. This involves setting up the necessary credentials to facilitate the sending of notifications to the requester's application.

5. Testing Notification Delivery:

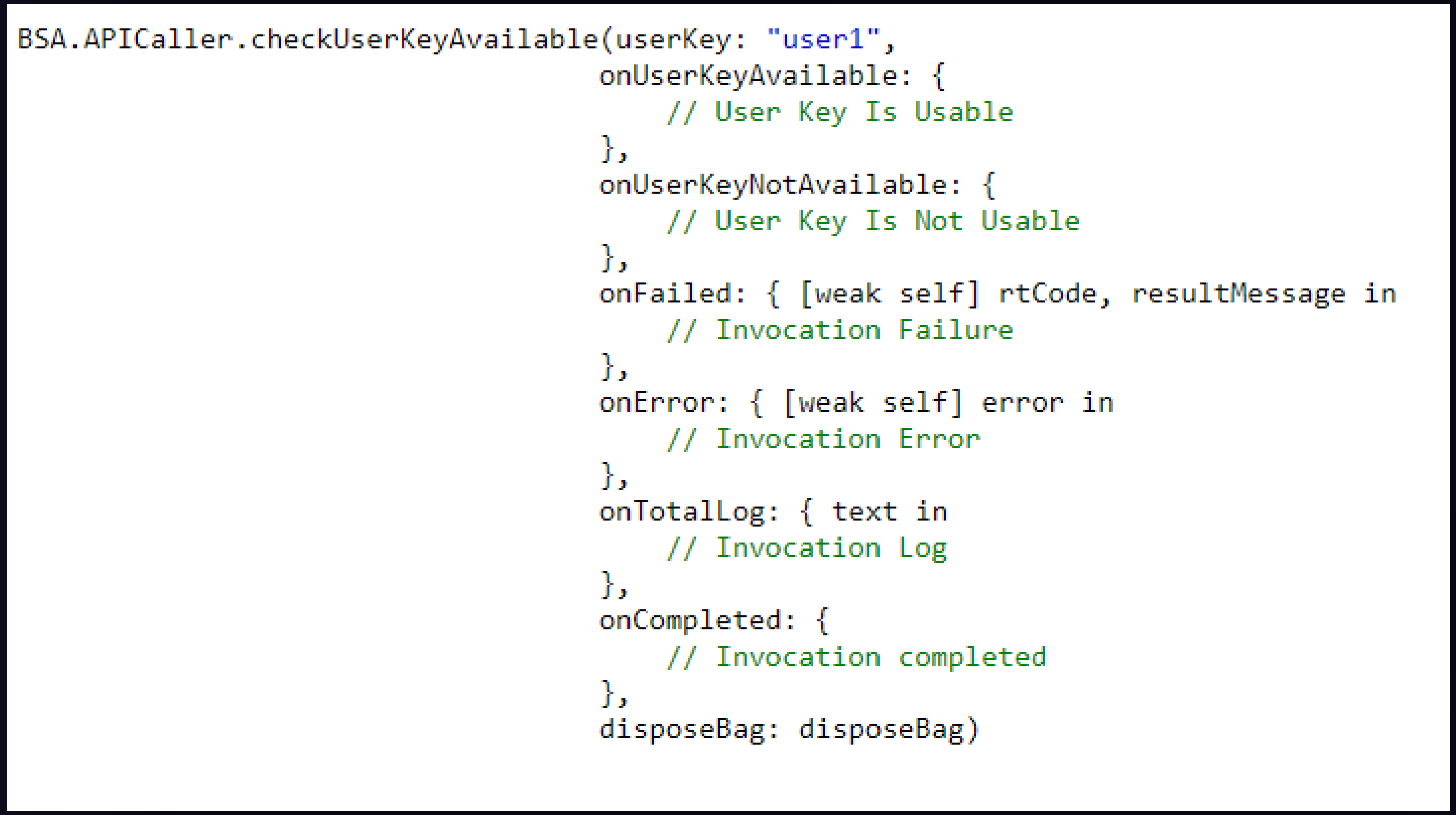
   Once the configuration is complete, requester can proceed to test whether they receive notifications from the BSA App Server. This testing phase includes sending a test notification from the server to the requester's application and verifying its successful reception.

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

SDK Integration :

Swift



```swift
BSA.APICaller.checkUserKeyAvailable(userKey: "user1",
                                   onUserKeyAvailable: {
                                       // User Key Is Usable
                                   },
                                   onUserKeyNotAvailable: {
                                       // User Key Is Not Usable
                                   },
                                   onFailed: { [weak self] rtCode, resultMessage in
                                       // Invocation Failure
                                   },
                                   onError: { [weak self] error in
                                       // Invocation Error
                                   },
                                   onTotalLog: { text in
                                       // Invocation Log
                                   },
                                   onCompleted: {
                                       // Invocation completed
                                   },
                                   disposeBag: disposeBag)
```

BsaSdk’s checkUserKeyAvailable() to check if there is a duplicate user ID.

Param : userKey

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

1. Debugging

You may use debugging feature (timber, print, log etc) to print the error code or parameter pass to see if it is success or not.

2. Crashes or unexpected behavior in the app

Whenever you interact with SDK function, and the apps is crashes, this maybe due to incorrect SDK initialization steps, incorrect permission configurations or incorrect API Parameters Called. Please double-check it. Including FCM.

3. Integration Errors and Incompatibilities

The SDK integration results in conflicts with existing libraries used in the app. Please check whether there are any known compatibility issues between BSA SDK and other current libraries used. Please used the specific version dependencies.

** You may also check the Error Code provided

COPYRIGHT © 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.

1. Follow the SDK documentation guideline meticulously and comprehensively
2. Evaluate SDK suitability and compatibility with your project
3. Implement Error Handling for user interaction and Debug Logging for developer
4. Understanding Error Code in BSA SDK and how to troubleshoot
5. Test thoroughly, have a contingencies plan and always do a back up of your development

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PRO HIBITED.

BSA SDK Flexibility Use Case

It is principle.

Banking Sector

FNSPay Demo

Implementation on each transaction process

HR Sector

BSA HR Demo

Implementation on attendance checking in and out, applicant request leave and approval with e-KYC

Lawyer Sector

Digital Signature System (DSS) Demo

Implementation on creating digital signature, user verification, and signing the documents

Demo

BSA Admin Portal

An admin portal that have special access on viewing and tracking BSA's user and allowir permission of each user level

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD

SDK Documentation

Click here [aOS]   Click here [iOS]

FCM Setup

Click here

FNSPay Demo App

Click here

Sandbox Web Portal

Click here

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PRO HIBITED.

Integration is not merely the merging of code; it's the harmonious convergence of vision, creativity, and execution.

May your integrations inspire, your solutions empower, and your journey continue to unfold.

Thank You

COPYRIGHT Ⓒ 2024. FNSM (M) SDN BHD. ALL RIGHTS RESERVED

COPYING, REPRINTING, DISTRIBUTING, OR USING THIS CONTENT IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF FNS (M) SDN BHD IS PROHIBITED.