

Infineon Android LPA

A Local Profile Assistant Example Implementation

Application Note

OPTIGA™ Connect Consumer OC1120

About this document

Scope and purpose

This document shall enable software and solution developers to understand and use the source code provided within the “Infineon Android LPA” project.

Intended audience

Software developers, solution developers and integrators who want to use or integrate the OPTIGA™ Connect Consumer products from Infineon.

Table of contents

About this document.....	1
Table of contents.....	2
List of figures	3
List of tables	4
1 Introduction	5
1.1 Scope of the Project	5
2 Getting Started	7
3 Using the Infineon Android LPA Application	8
3.1 Displaying installed profiles.....	8
3.2 Downloading a new profile	9
3.3 Profile Details	10
3.5 eUICC Details	11
3.6 Preferences.....	12
3.7 Using an external Identiv USB Reader	13
4 Software project description	14
4.1 Development environment.....	14
4.2 Core, messages and util module dependencies	14
4.3 App module dependencies	15
4.3.1 Identiv USB reader library.....	15
4.3.2 Further Dependencies.....	15
4.4 Project Structure	16
4.4.1 Package com.infineon.esim.lpa	16
4.4.3 Package com.infineon.esim.lpa.core	18
5 Issues and Limitations	19
5.1 Out of Scope	19
Glossary of Acronyms.....	20
References.....	21
Revision history.....	22

List of figures**List of figures**

Figure 1	Scope of the project marked with red rectangle.	6
Figure 2	Profile list screen.	8
Figure 3	Screenshots of the profile download process.....	9
Figure 4	Profile details screen.....	10
Figure 5	eUICC details screen.....	11
Figure 6	Preferences screen.	12
Figure 7	Infineon Android LPA on Google Pixel 4 XL with attached Identiv SCR 3500 and OPTIGA™ Connect Consumer OC1120 sample in ID1 package.....	13



List of tables

List of tables

Table 1	Development environment description	14
Table 2	Dependencies for module core.....	14
Table 3	Dependencies for module messages.....	14
Table 4	Common dependencies for modules core, messages and util	14
Table 5	Identive USB Reader Library Details.....	15
Table 6	General Code Structure Description.....	16
Table 7	Package com.infineon.esim.lpa description.....	17
Table 8	Package com.infineon.esim.lpa description.....	18
Table 9	Acronyms	20

Introduction

1 Introduction

This project aims to offer an example implementation of a Local Profile Assistant (LPA) as an Android application. The implementation shall show how easy it is to integrate the Infineon OPTIGA™ Connect Consumer products into your Android device.

With the Infineon Android LPA you will be able to

- List the installed profiles on the eSIM/eUICC
- Enable/Disable/Switch profiles
- Show profile details
- Delete profiles
- Download new profiles via QR code from Live or Test SM-DP+ profile servers
- Show eSIM/eUICC details

using an OPTIGA™ Connect Consumer OC1120 engineering sample

- in the SIM slot of the phone
- in an Identivate USB Reader connected to the phone via OTG adapter

1.1 Scope of the Project

The GSMA SGP.22 [1] distinguishes between two variants of LPAs. The LPAe inside the eUICC itself and the LPA_d in the device (host). This project gives an implementation for an LPA_d that is hosted on an Android device.

The LPA_d is further divided in three sub modules:

- Local Profile Download (LPD_d)
- Local User Interface (LUI_d)
- Local Discovery Service (LDS_d)

The scope of this project is restricted to the LPD_d and LUI_d, since the LDS_d is not necessary for downloading a new profile via a normal QR code.

This implementation therefore supports the following interfaces specified in [1]:

- ES9+: Interface between LPA_d (LPD_d) and SM-DP+ server
- ES10b, ES10c: Interface between LPA_d (LPD_d) and eUICC
- ES8+: implicitly supported
- ESeu: User interface between LPA_d (LUI_d) and End User

The scope of the project is also shown with the red rectangle in Figure 1.

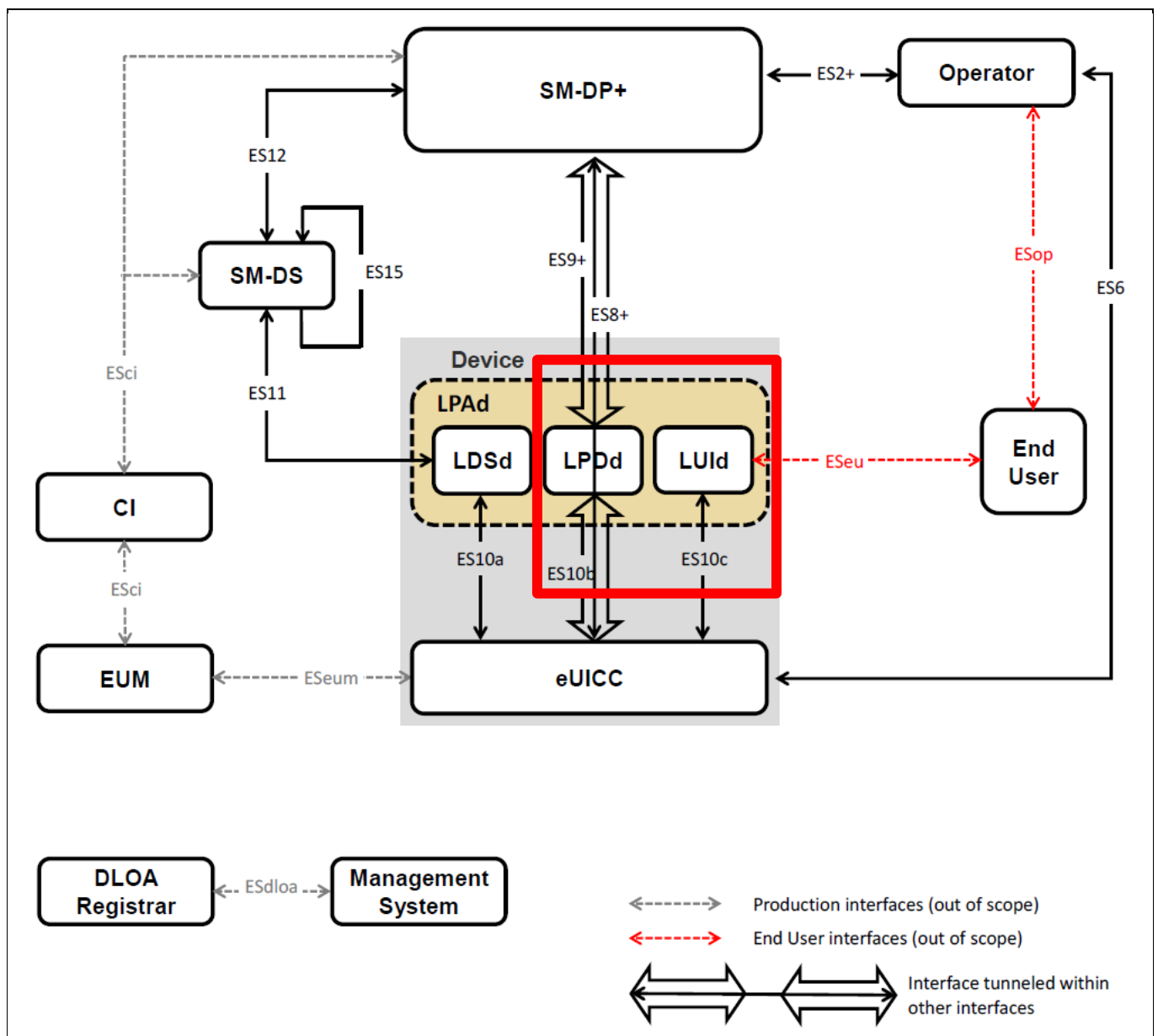


Figure 1 Scope of the project marked with red rectangle.

2 Getting Started

To get started with the software project, you'll need:

- Hardware:
 - OPTIGA™ Connect Consumer OC1120 engineering sample
 - Android phone with Android 8 or higher (e.g. Google Pixel 4 XL)
 - Optionally: Identiv USB reader (e.g. SCR3500) and OTG adapter
- Software:
 - Infineon Android LPA source code
 - Android Studio (see next steps)

Please perform the following steps to start and build the software project.

1. Download and install Android Studio
 - a. Download Android Studio from: <https://developer.android.com/studio>
 - b. Install Android Studio
 - c. Open Android Studio
2. Open Infineon Android LPA project
 - a. Unzip file app.infineonlpa.vX.X.X.zip
 - b. File -> Open and select the unzipped folder from previous step
 - c. Wait until Project and Gradle synchronization is finished.
3. Build the project
 - a. Build -> Make project

Now the Infineon Android LPA can be easily installed to your target device via Android Studio.

3 Using the Infineon Android LPA Application

3.1 Displaying installed profiles

The main screen of the application shows a profile list with the active (enabled) and available (installed but disabled) profiles. Please see the following screenshot.

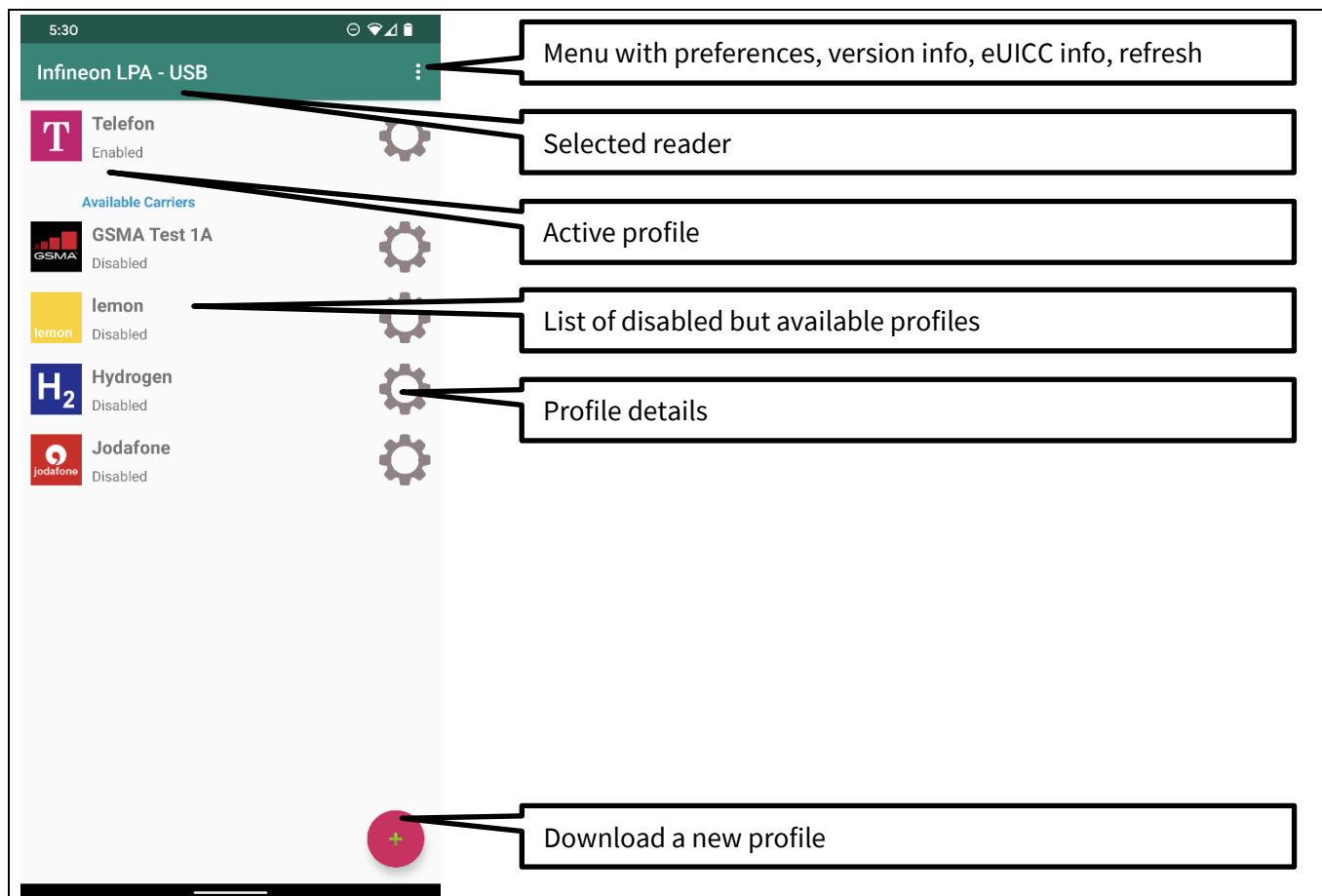


Figure 2 Profile list screen.

The following interactions are possible from the profile list:

1. Open the menu on the top right
 - a. Go to app preferences
 - b. Show eUICC info
 - c. Show app version info
 - d. Show open source licenses
 - e. Refresh profile list
2. Show details of a profile by pressing the gear symbol next to the profile
3. Enable a profile by pressing the icon or the name of a profile.
4. Download a new profile by pressing the + on the bottom right.

3.2 Downloading a new profile

To download a new profile, a QR code with the activation code is needed.

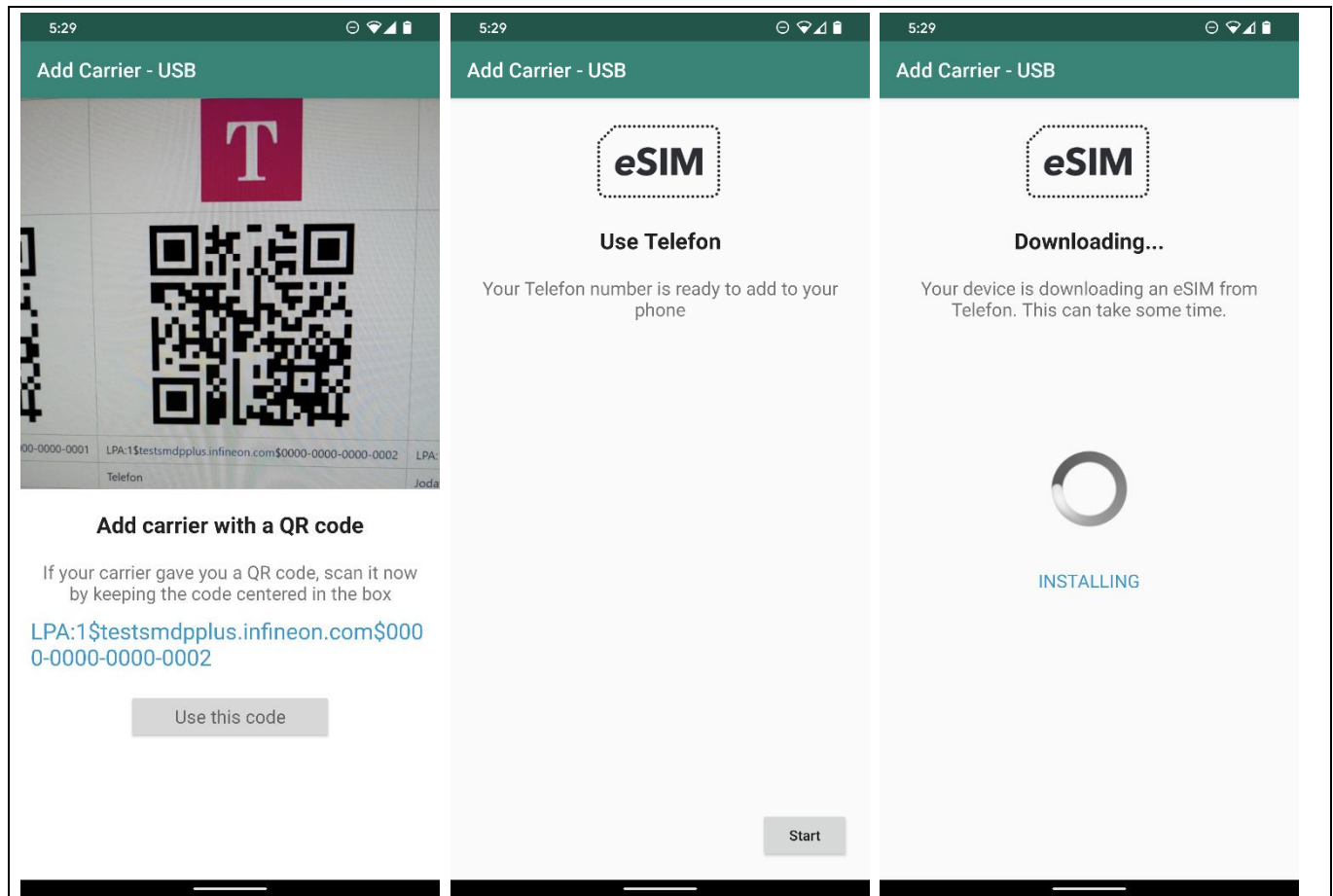


Figure 3 Screenshots of the profile download process

3.3 Profile Details

The following screenshot shows the profile details screen. Here you can see the profile nickname, the provide name, the ICCID and the enablement status. With the button enable/disable, you can enable/disable the profile. With the button delete, you can delete the profile. The latter option might only be available if the profile is already disabled (see the preference menu for details).

The nickname of the profile can be modified with a click on the pencil icon and is directly stored to the eUICC.tr

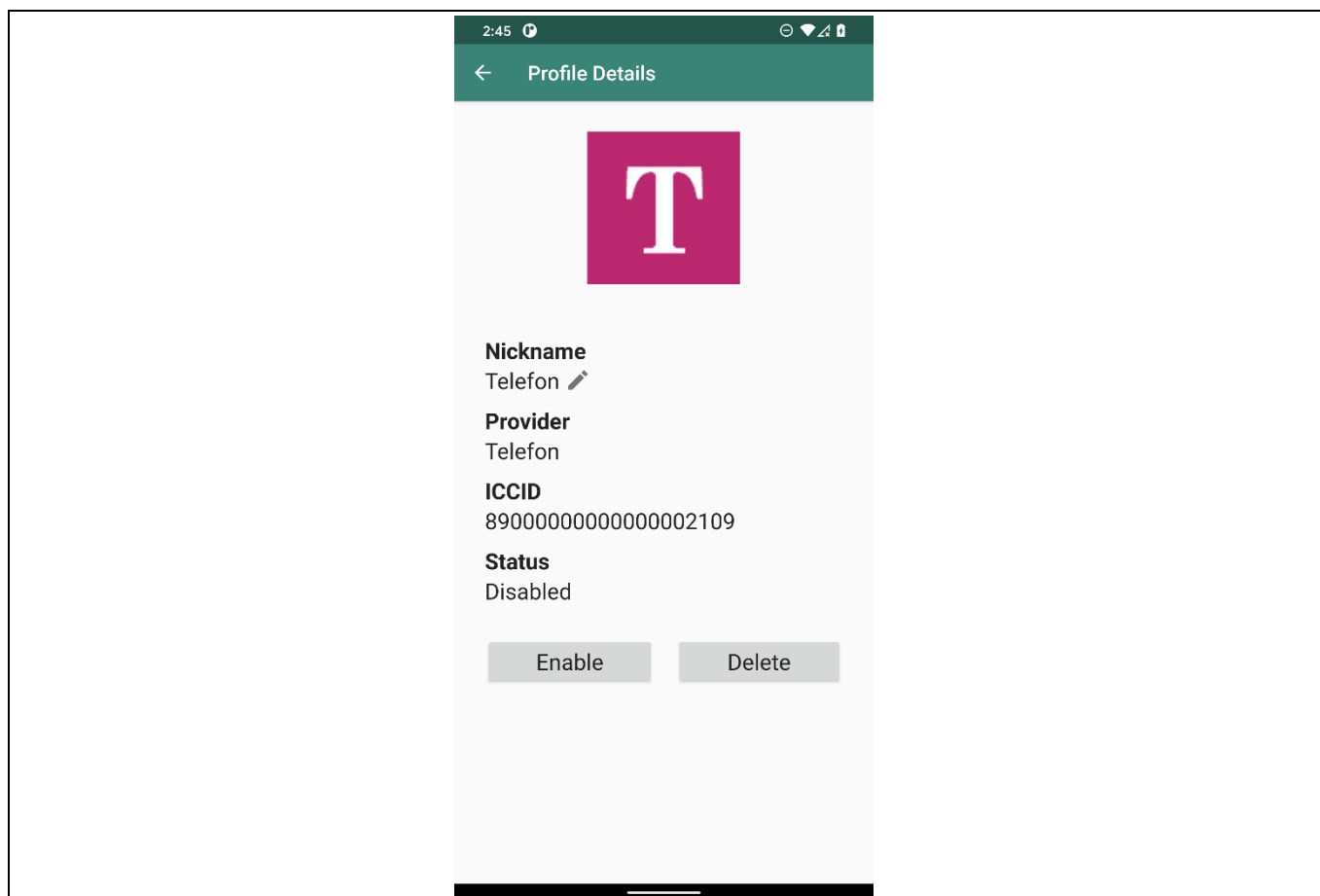


Figure 4 Profile details screen.

3.5 eUICC Details

The following screenshot shows the eUICC details screen.

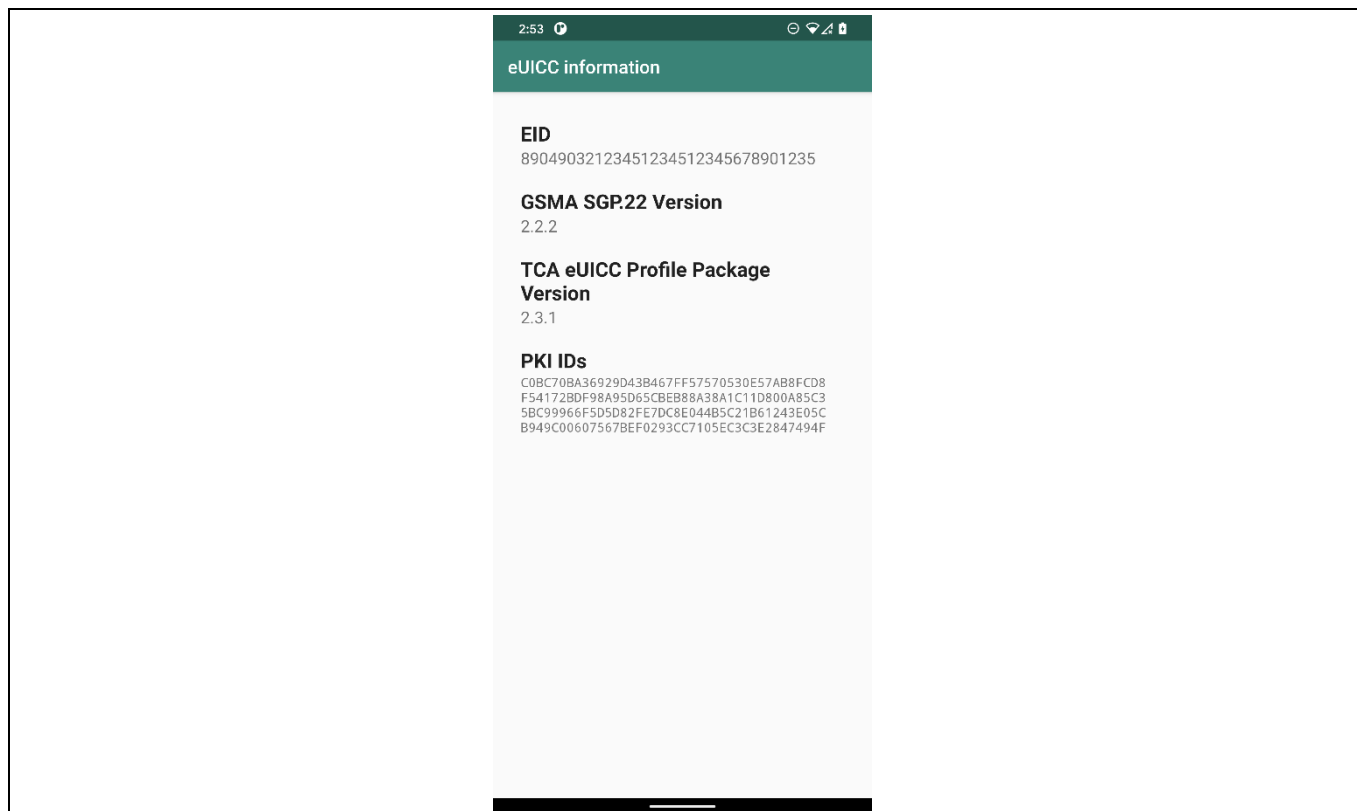


Figure 5 eUICC details screen.

3.6 Preferences

The following screenshot shows the preference screen that can be reached via the main screen by pressing the ellipsis symbol on the top right and selecting “preferences” from the menu.

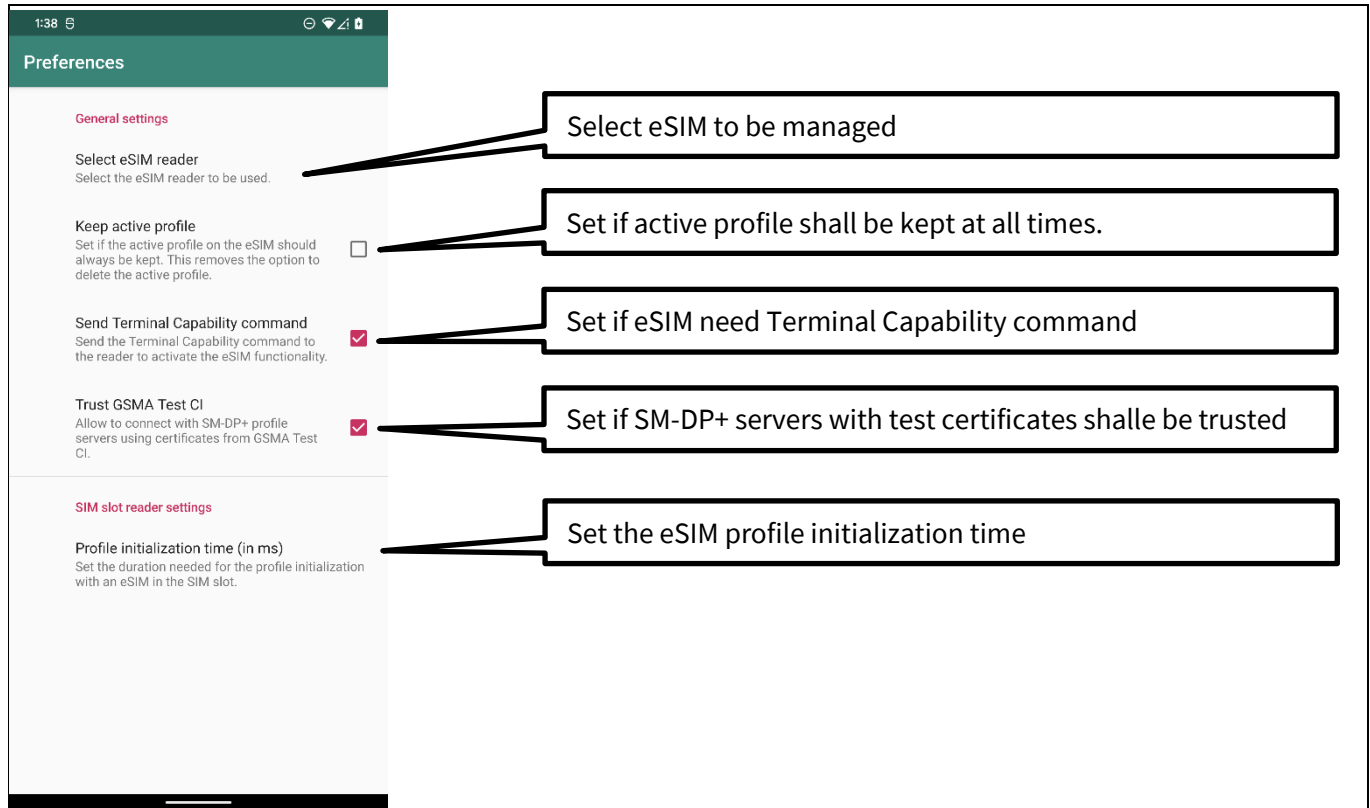


Figure 6 Preferences screen.

3.7 Using an external Identiv USB Reader

To use an external Identiv USB reader an OTG adapter has to be used. See the following image for a possible setup.

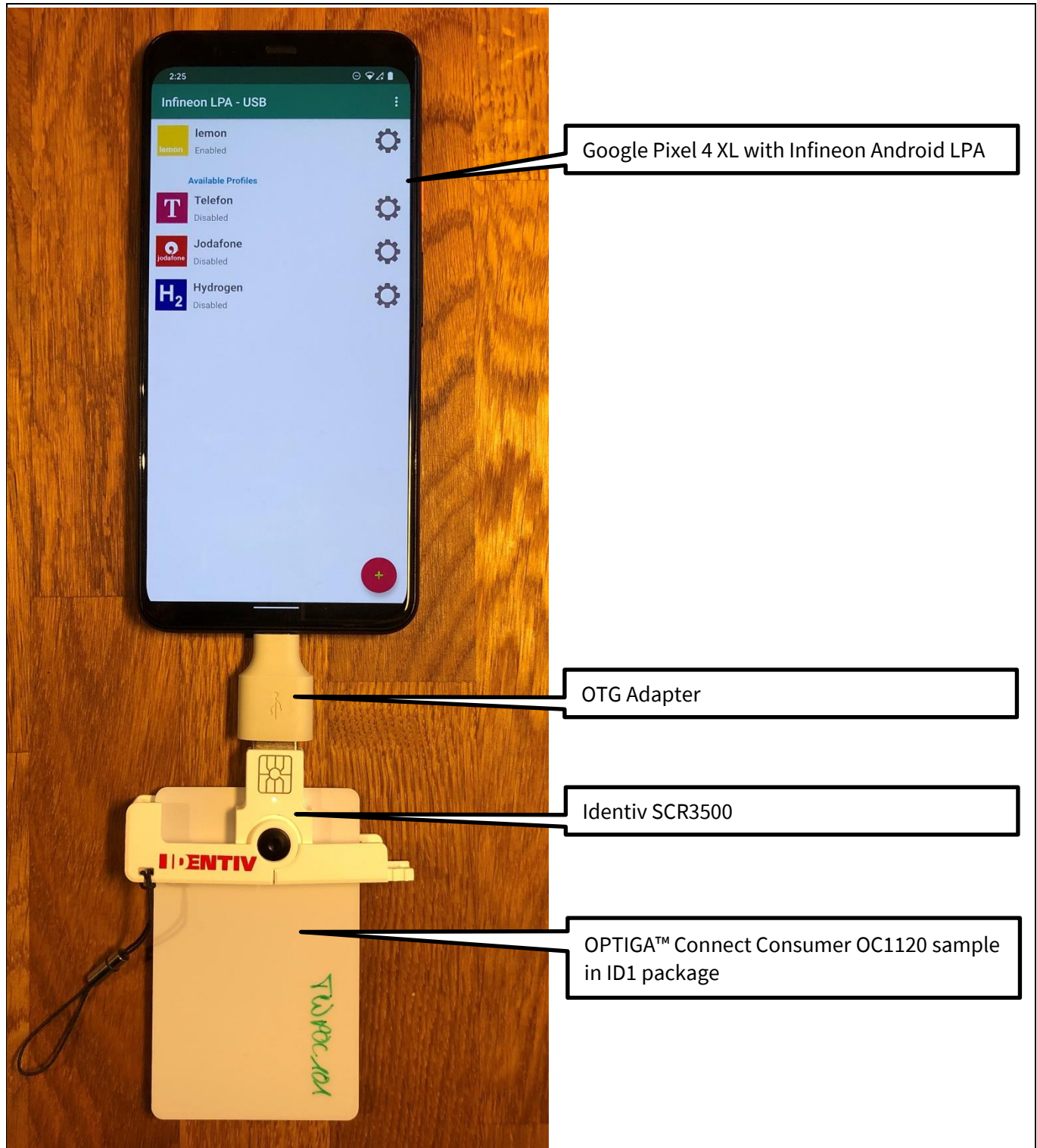


Figure 7 Infineon Android LPA on Google Pixel 4 XL with attached Identiv SCR 3500 and OPTIGA™ Connect Consumer OC1120 sample in ID1 package.

4 Software project description

4.1 Development environment

The following setup has been used to create the software project:

Table 1 Development environment description

Type	Description
IDE	Android Studio Arctic Fox 2020.3.1
Build Tool	Gradle Build Tool 7.0.2
Minimum Android SDK Version	28 (Android 8)
Target Android SDK Version	31 (Android 11)
Source Compatability	Java 8

Note: The [Android Secure Element OMAPI](#) that enables the use of an eSIM was added in Android SDK 28 (Android 8) and is a substantial requirement for this project. This is why the minimum Android SDK version is 28.

4.2 Core, messages and util module dependencies

The software dependencies are listed in the following tables.

Table 2 Dependencies for module core

Library Name	Version	Remark
messages	-	Messages module for ASN1 schema
util	-	Util module
com.beanit:jasn1	1.11.3	ASN1 codec
com.google.code.gson:gson	2.8.9	JSON codec

Table 3 Dependencies for module messages

Library Name	Version	Remark
util	-	Util module
com.beanit:jasn1	1.11.3	ASN1 codec

Common dependencies for all three modules

Table 4 Common dependencies for modules core, messages and util

Library Name	Version	Remark
androidx.appcompat:appcompat:1.4.1	1.4.1	Support for annotations
junit:junit:4.13.2	4.13.2	jUnit for unit tests

4.3 App module dependencies

4.3.1 Identiv USB reader library

As a fallback to using the internal eSIM we introduced the use of an external USB reader from Identiv such as the Identiv SCR3500.

Table 5 Identive USB Reader Library Details

Library Name	Version	Remark
com.identive.libs:androidSCard	1.2	Support for Identiv USB readers as fallback for use of internal eSIM/SE.

This library is automatically downloaded during the build process via a Gradle script in app/build.gradle.

If you want to download the library manually, follow these steps:

1. Download Identiv Android CCID Library from <https://support.identiv.com/developer-tools-for-smart-card-readers/>
2. Unzip the zip file
3. Copy the androidSCardV1.2.jar file to folder /app/libs/

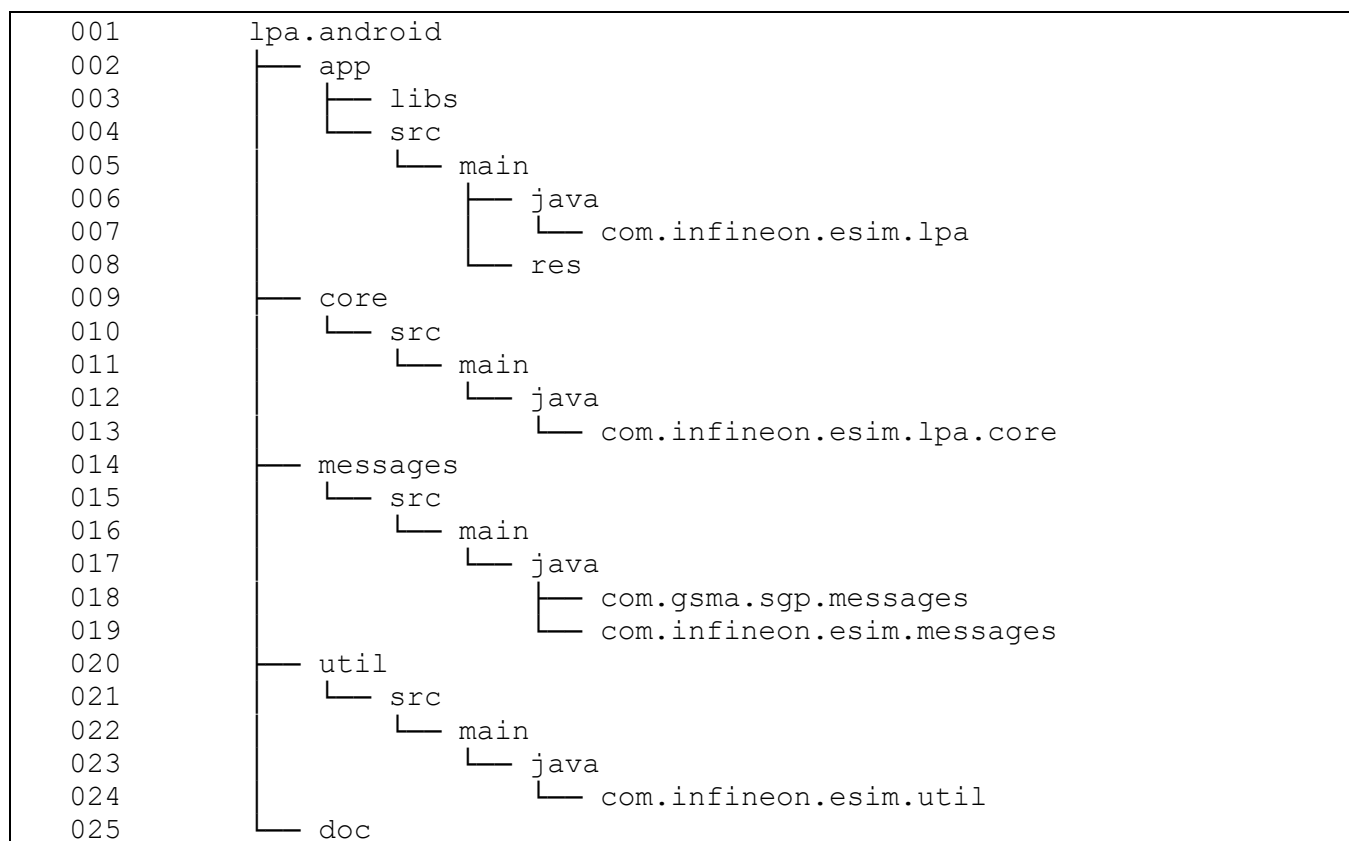
4.3.2 Further Dependencies

Please review the build.gradle files in the project.

4.4 Project Structure

The project follows the generic code structured of an Android application.

Code Listing 1 Project Structure



The main parts of the code are described in the following table.

Table 6 General Code Structure Description

Package Name	Description
app/libs	Third party libs that are not available via Gradle
app/src/main/java	Java source code of the Android application
app/src/main/res	Resources of the Android application, e.g. layouts, values, icons etc.
core/src/main/java	Basic LPA core functionality.
messages/src/main/java	GSMA SGP.22 ASN1 schema as Java classes
util/src/main/java	Utility classes
doc	Documentation, including Application Note.

4.4.1 Package com.infineon.esim.lpa

The `com.infineon.esim.lpa` package comprises the source code of the Android application and integrates the basic LPA core functionality of the `com.infineon.esim.lpa.core` LPA core library.

The general code structure of the `com.infineon.esim.lpa` package looks as follows.

Code Listing 2 Project Structure for Package com.infineon.esim.lpa

001	com.infineon.esim.lpa
002	├── data
003	├── reader
004	│ ├── identive
005	│ └── se
006	├── task
007	├── ui
008	│ ├── dialog
009	│ ├── downloadProfile
010	│ ├── euiccDetails
011	│ ├── objects
012	│ ├── preference
013	│ ├── profileDetails
014	│ ├── profileList
015	│ └── scanBarcode
016	├── util
017	│ ├── android
018	│ └── threading

In the following table the sub-packages are further explained with their functionality.

Table 7 Package com.infineon.esim.lpa description

Package Name	Description
reader	Reader implementations for Secure Element and Identiv USB reader
task	Background tasks needed for APDU communication etc.
data	Data model of the Android application
ui	User Interface implementation of the Android application
ui.dialog	Custom confirmation dialog
ui.downloadProfile	Activity for the profile download
ui.euiccDetails	Activity for eUICC details presentation
ui.objects	Objects used to UI management
ui.preference	Activity for preferences
ui.profileDetails	Activity for profile details presentation
ui.profileList	Main activity for display of the profile list
ui.scanBarcode	Activity to scan a QR code for the profile download
util	Utility classes
util.android	Utility classes for Android-specific topics
util.threading	Utility classes for multi-threading

4.4.3 Package `com.infineon.esim.lpa.core`

The `com.infineon.esim.lpa.core` contains the source code for the LPA functionality. It supports the following additional features:

- All GSMA SGP.22 specified ES9+ functions
 - `InitiateAuthentication`
 - `AuthenticateClient`
 - `GetBoundProfilePackage`
 - `HandleNotification`
 - `CancelSession`
- All GSMA SGP.22 specified ES10 functions
- Confirmation code handling
- Support for BF76 tag

The general code structure of the `com.infineon.esim.lpa.core` package looks as follows.

Code Listing 3 Project Structure for Package `com.infineon.esim.lpa`

019	<code>com.infineon.esim.lpa.core</code>
020	└─ <code>es9plus</code>
021	└─ <code>es10</code>
022	└─ <code>objects</code>
023	└─ <code>worker</code>

In the following table the sub-packages are further explained with their functionality.

Table 8 Package `com.infineon.esim.lpa` description

Package Name	Description
es9plus	ES9+ interface according to [1]
es10	ES10 interface according to [1]
objects	Objects for LPA management and module API
worker	Workers that perform LPA functionality

5 Issues and Limitations

5.1 Out of Scope

Currently out of scope of the project are the following topics:

- Using an SM-DS Discovery Service for profile download.
- Support for modem functionality that goes beyond the OMAPI functionality. E.g. AT commands.

Glossary of Acronyms

Table 9 **Acronyms**

APDU	Application Protocol Data Unit
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
AT	Stands for ATtention command which is used to control modems
BLE	Bluetooth Low Energy
BPP	Bound Profile Package
BSD	Berkeley Source Distribution
cURL	Client URL is a tool for transferring data using various network protocols.
ECDSA	Elliptic Curve Digital Signature Algorithm
eSIM	Embedded Subscriber Identity Module
ES2+	Interface used by the Operator to order profiles and perform other administrative functions.
ES9+	Interface to provide a secure transport between the SM-DP+ and the LPA for the delivery of the Bound Profile Package.
ES10	Interface between the LPA and the eSIM for profile management and transfer of a Bound Profile Package.
eUICC	Embedded Universal Integrated Circuit Card
GP	GlobalPlatform
GSM	Global System for Mobiles
GSMA	GSM Association
ID-1	A standard card size of 85.60 by 53.98 mm.
IoT	Internet of Things
JSON	JavaScript Object Notation
LPA	Local Profile Assistant
LTE	Long Term Evolution
MIT	Massachusetts Institute of Technology
MIT/X	MIT License for the X Window System
OS	Operating System
PCSC	Personal Computer / Smart Card
PPP	Point to Point Protocol
RSP	Remote SIM Provisioning
SIM	Subscriber Identity Module
SM-DP+	Subscription Manager – Data Preparation. A profile server for MNO profiles over interface ES9+.
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface
UICC	Universal Integrated Circuit Card
USB	Universal Serial Bus

References

- [1] GSM Association SGP.22 RSP Technical Specification, Version 2.4.0
- [2] [Android Developers: android.se.omapi OMPI documentation](#)

Revision history

Document version	Date of release	Description of changes
1.0.0	15 January 2021	Initial version.
2.0.0	05 August 2021	Update for release of version 2.0.0 of the application.
2.0.2	08 October 2021	Minor fixes and alignment with software version number.
3.0.0	07 February 2022	Major modularization and reduction of dependencies of the project.
4.0.0	24 June 2022	Release for distribution.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-06-24

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email:

security.chipcard.ics@infineon.com

Document reference

AppNote Number

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.