OPTIGA™ Authenticate NBT

Software integration guide

About this document

Scope and purpose

The scope of this document is to outline the process of integrating the provided host libraries for the OPTIGA™ Authenticate NBT into custom applications on various target platforms. The guide also introduces multiple example applications and provides guidance on how to evaluate them with the designated development hardware, as well as how to use them as a basis for the development of custom applications.

The purpose of this document is to provide a thorough overview of the provided off-chip software for the OPTIGA™ Authenticate NBT. It shall act as an entry point to inform readers about the different components and direct them to detailed support material based on their interests.

Intended audience

This document is primarily intended for embedded and mobile application developers. Additionally, it serves as a high-level overview for solution providers, system integrators, application developers, and product marketers.

Table of contents

About this document . . . 1
Table of contents . . . 2
List of figures . . . 3
1 Introduction . . . 4
1.1 Documentation landscape . . . 4
1.2 NFC I2C bridge tags . . . 4
2 Software overview . . . 6
2.1 General information . . . 6
2.2 Host libraries . . . 6
2.3 Example applications . . . 7
3 Software evaluation and integration . . . 9
3.1 Example application evaluation . . . 9
3.2 C host library integration . . . 9
3.3 Java host library integration . . . 11
3.4 Swift host library integration . . . 12
References . . . 13
Glossary . . . 14
Revision history . . . 15
Disclaimer . . . 16

List of figures

Figure 1 OPTIGA™ Authenticate NBT documentation landscape 4
Figure 2 System software overview 6
Figure 3 Host libraries to support the implementation of embedded applications and mobile phone apps 7
Figure 4 Overview of the example applications and their included components 8
Figure 5 OPTIGA™ Authenticate NBT Development Kit, including a PSoC™ host MCU board 9
Figure 6 Platform-independent and platform-dependent parts of an MCU implementation 10
Figure 7 Java host library architecture in an Android application 11
Figure 8 Swift host library architecture in an iOS application 12

1 Introduction

This chapter provides an explanation of how this guide fits into the OPTIGA™ Authenticate NBT documentation landscape and a short product overview.

Chapter 2 presents an outline of the software components utilized with the device and introduces the provided host libraries and example applications.

Chapter 3 provides a high-level guidance for evaluating the OPTIGA™ Authenticate NBT with the offered example applications and for integrating the host libraries into custom applications.

***Note**: For a collection of all available support material for the product, refer to its product page [1].*

1.1 Documentation landscape

Figure 1 depicts the OPTIGA™ Authenticate NBT documentation landscape and provides an overview about relevant documentation for the integration of the device into custom systems.

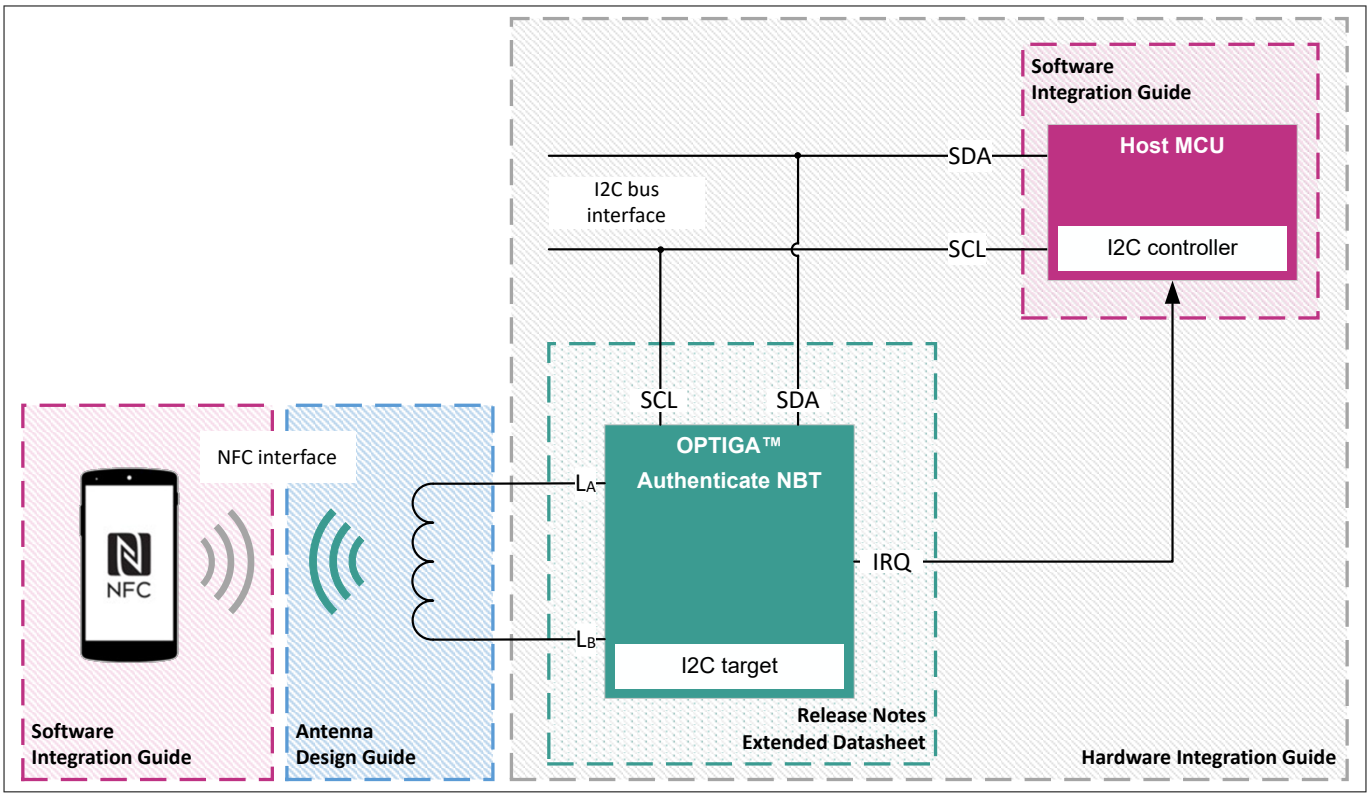


Figure 1 OPTIGA™ Authenticate NBT documentation landscape

This document, the Software Integration Guide, assists customers with the software integration of the device into custom applications on various platforms. This guide includes an overview of the software integration methodology in general and introduces the provided host libraries and example applications. While this document focuses on the high-level aspects, it also presents references to more detailed documentation for the host software integration of the OPTIGA™ Authenticate NBT.

1.2 NFC I2C bridge tags

NFC Bridge Tags are dual-interface tags that enable contactless features for IoT devices via an I2C controller interface, allowing for a touch-and-go experience with a mobile phone. On one side, the NFC Bridge Tags include a contactless passive NFC interface and on the other side, a contact-based I2C target interface that connects to the MCU of the IoT device.

1 Introduction

The OPTIGA™ Authenticate NBT harnesses the Integrity Guard 32 security architecture to provide an option for the end-user with symmetric and asymmetric cryptographic operations, as well as password-based data protection schemes. As a result, the device is ideal for security demanding applications.

This product includes device authentication, pass-through and asynchronous data transfer modes, which can be used for variety of applications such as:

- Keyless access and activation of shared mobility vehicles
- Controlled access to personal electronic devices such as HDD
- Theft prevention for electronic goods by authenticated activation

This tag can also be used in healthcare and industrial applications. The OPTIGA™ Authenticate NBT, in combination with healthcare sensors, enables access to information through an NFC-enabled mobile phone or reader. Furthermore, the device is an ideal product for industrial applications such as headless configuration and parametrization of devices, assembly line programming and fault diagnostics.

2 Software overview

This chapter provides an overview of the software components in systems utilizing the OPTIGA™ Authenticate NBT. Furthermore, the provided host software is introduced, which streamlines the process of integrating the device into various applications and systems.

2.1 General information

As illustrated in Figure 2, a typical system utilizing the OPTIGA™ Authenticate NBT also comprises an NFC reader and a host microcontroller. From the perspective of the OPTIGA™ Authenticate NBT, it is important to differentiate between on-chip software and off-chip software, as indicated.

- **On-chip software:** Describes the application(s) operating on the OPTIGA™ Authenticate NBT, encompassing its intended functionality. Each device is programmed with this software during its manufacturing process, resulting in its unalterable state thereafter
- **Off-chip software:** Describes the software applications of the overall system, not running on the OPTIGA™ Authenticate NBT (for example, embedded applications or mobile phone apps)

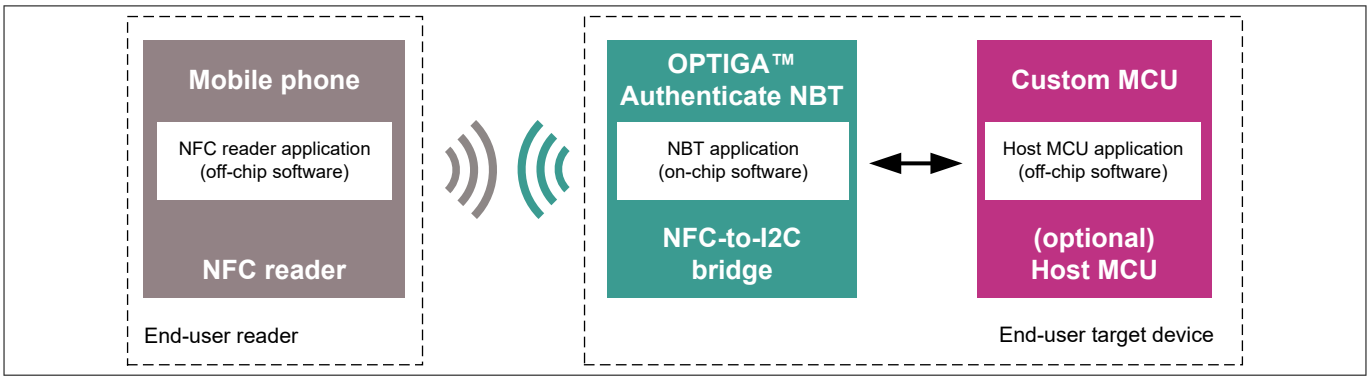


Figure 2 System software overview

Since the on-chip software is pre-flashed on the device and is not intended for modification, this guide focuses on the off-chip software components, which interact with the OPTIGA™ Authenticate NBT:

- The software on the NFC reader and/or a host microcontroller

The host microcontroller (for example, a PSoC™) is shown as *optional* in Figure 2 since a subset of the device's use cases (for example, brand protection) can be operated using the OPTIGA™ Authenticate NBT without a direct connection to a host microcontroller. In such cases, the device is powered through the NFC field and provides its functionality without the support of a microcontroller.

For off-chip software, Infineon provides multiple host libraries that can be used by NFC reader applications as well as embedded host MCU applications. Additionally, multiple example applications illustrate various use cases of the OPTIGA™ Authenticate NBT. The host libraries and example applications are described in more detail in the following sections.

2.2 Host libraries

To support the software integration of the OPTIGA™ Authenticate NBT into custom applications on various target platforms, Infineon provides several host libraries. These libraries can be included in NFC reader and host MCU applications to provide an easy-to-use communication interface to the device (either via NFC or via I2C).

The following host libraries are provided for the OPTIGA™ Authenticate NBT (see Figure 3):

- **C host library:** This library is intended to be used in C-based applications in systems that interact with the OPTIGA™ Authenticate NBT via the I2C interface. The library mainly targets embedded applications running on microcontrollers (for example in IoT devices)

- **Java host library:** This library is intended to be used in Java-based applications that interact with the device via the NFC interface. The main target platform is Android for mobile phone apps
- **Swift host library:** This library is intended to be used in Swift-based applications that interact with the device via the NFC interface. The main target platform is iOS for mobile phone apps

The host libraries are mainly intended to serve the following purposes:

- To evaluate the capabilities of the OPTIGA™ Authenticate NBT on arbitrary platforms
- The development and deployment of custom software applications on arbitrary platforms. The host libraries provide a high-level abstraction of the OPTIGA™ Authenticate NBT's functionality to simplify the development process of custom applications

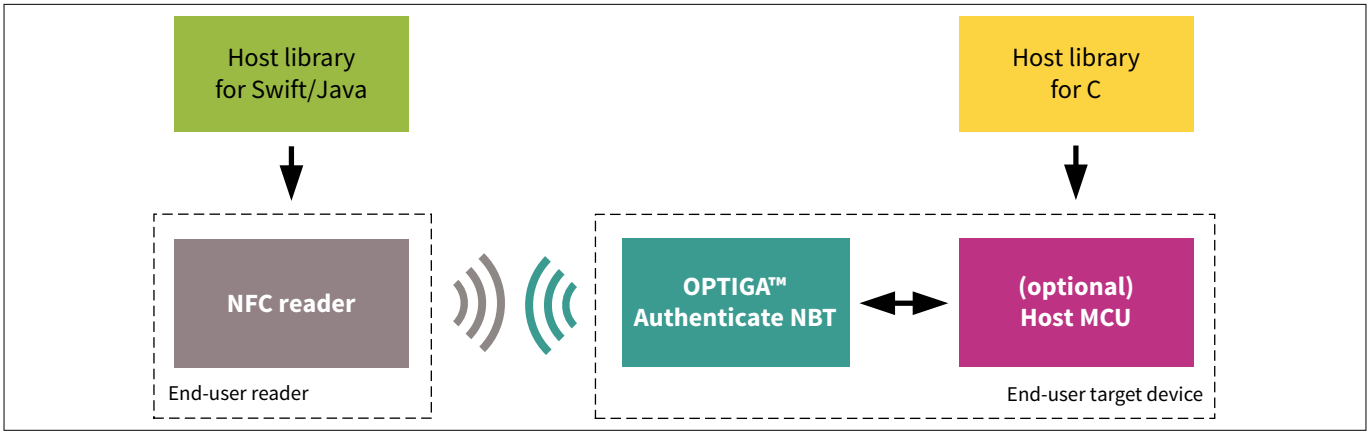


Figure 3 **Host libraries to support the implementation of embedded applications and mobile phone apps**

The three libraries (C/Java/Swift) are available as separate repositories on GitHub [2]. In addition to the source code, each repository contains detailed documentation on how to integrate the library into applications on custom platforms.

Additionally, the host library for C is also integrated in Infineon's ModusToolbox™ as middleware for specific microcontrollers, such as PSoC™ 6.

***Note**: The host libraries are provided "as is" without any warranties and liabilities.*

2.3 Example applications

Infineon provides multiple example applications that demonstrate the capabilities of the OPTIGA™ Authenticate NBT in different use cases. These implementations utilize the host libraries to demonstrate example applications on mobile phones (Android and iOS) as well as on the reference host microcontroller (PSoC™). They serve as examples for integrating and utilizing the libraries in practical applications while also providing developers with working examples to evaluate the device.

Example applications are available for the following use cases of the OPTIGA™ Authenticate NBT:

- **Brand protection with offline authentication:**
  - Demonstrates the use of the device's offline brand protection functionality
  - Includes mobile phone apps only (Android/iOS), as a host MCU is not required for brand protection
- **Brand protection with online authentication:**
  - Demonstrates the use of the device's online brand protection functionality
  - Includes the implementation of a simple python webserver with Flask only, as neither a dedicated mobile phone app nor a host MCU is needed for online brand protection. The webserver verifies the brand-protection information in the NDEF file's URL which is automatically opened after approaching the tag with an Android- or iOS-based mobile phone

- **Host parameterization via asynchronous data transfer:**
  - Demonstrates the use of the device's asynchronous data transfer mode
  - Includes mobile phone apps (Android/iOS), and an embedded (PSoC™) application
- **Host parameterization via pass-through:**
  - Demonstrates the use of the device's pass-through mode
  - Includes mobile phone apps (Android/iOS), and an embedded (PSoC™) application
- **Static connection handover:**
  - Demonstrates the use of the device's static connection handover functionality
  - Includes an embedded microcontroller application only, as the NFC reader part is automatically implemented by the Android/iOS[1)] base system without the need for a custom application
- **Personalization via NFC:**
  - Demonstrates how to personalize the device for each of the provided use case implementations via NFC
  - Includes mobile phone apps only (Android/iOS)

Figure 4 shows an overview of the provided use case implementations and their associated mobile phone, embedded and webserver applications. The example applications are provided via GitHub [2]. For more information on how to use, implement and port these applications, refer to the detailed documentation in the respective repositories.

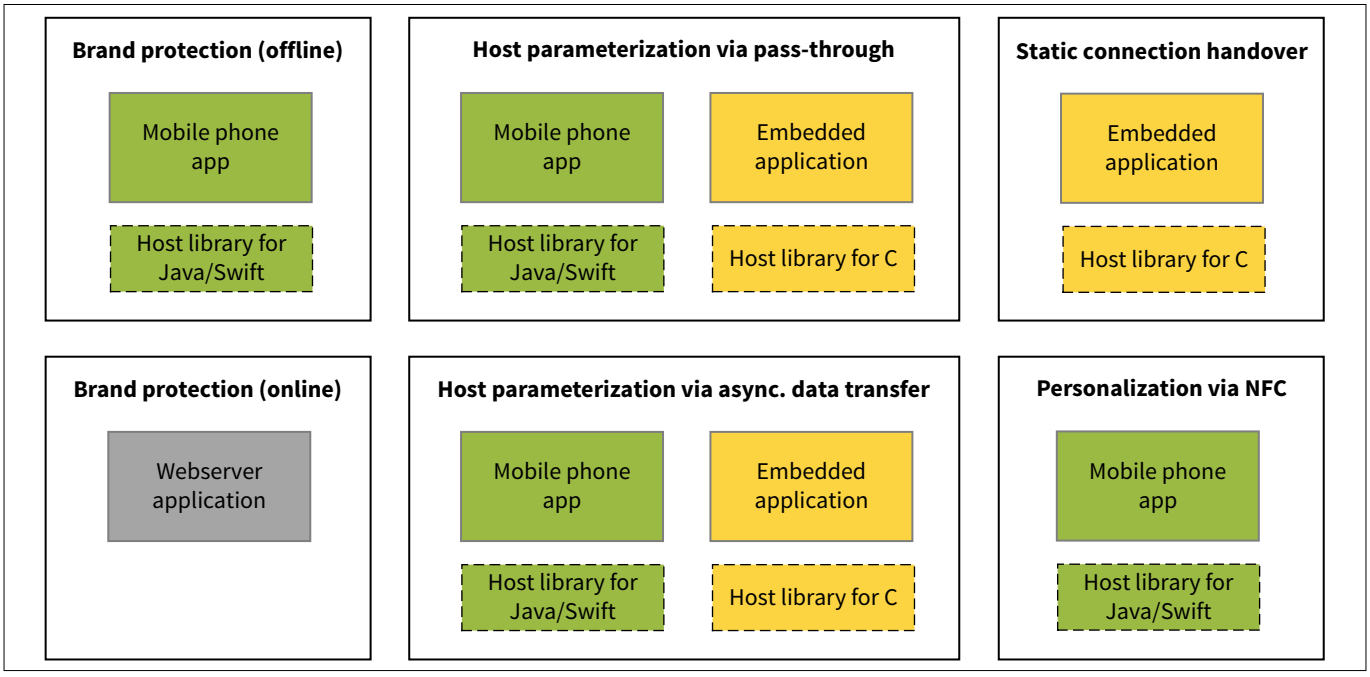


Figure 4 Overview of the example applications and their included components

The embedded MCU applications are additionally available as example applications via Infineon's ModusToolbox™. These applications can be selected when creating a new application based on the OPTIGA™ Authenticate NBT Development Kit's host microcontroller board.

***Note**: The example applications are intended for evaluation purposes without any warranties and liabilities.*

1 As of 2024, iOS devices do not yet support this functionality.

3 Software evaluation and integration

This chapter provides brief guidance for evaluating the OPTIGA™ Authenticate NBT example applications and for integrating the provided host libraries into custom projects. More detailed instructions can be found in each of the repositories' user guides on GitHub [2].

3.1 Example application evaluation

The example applications are available as source code in the form of Android Studio, ModusToolbox™, Python/ Flask and Xcode projects. All of them are hosted as individual Git repositories on GitHub. The example applications are intended to be evaluated with the OPTIGA™ Authenticate NBT Development Kit, which consists of a PSoC™ host MCU board and the OPTIGA™ Authenticate NBT Development Shield (see Figure 5). Alternatively, the shield may also be used standalone to evaluate applications which do not require an MCU (for example, brand protection). Refer to the development kit/shield product pages [9][10] for more information.

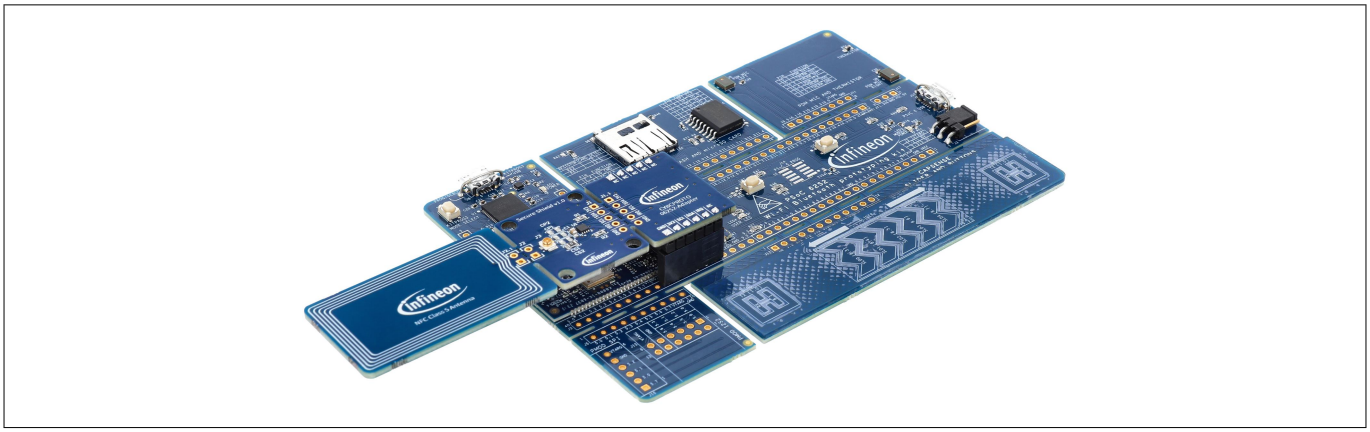


Figure 5 OPTIGA™ Authenticate NBT Development Kit, including a PSoC™ host MCU board

The individual projects can be downloaded from GitHub, and imported into the respective development IDE to build and deploy the applications. ModusToolbox™ further simplifies this step as it provides the embedded applications as examples when creating new projects based on the development kit's host microcontroller board.

Detailed instructions on how to download, build, and run the applications are available in each application's GitHub repository, the respective README.md file, and the associated user guide.

The example implementations show how to integrate and use the host libraries on specific platforms (PSoC™ 6, Android or iOS). The applications are implemented in a way that allows easy porting of the entire application to other platforms. For more information on how to port the example implementations to other platforms, refer to the associated documentation in the application's GitHub repository.

3.2 C host library integration

This section describes the process of integrating the C host library into custom embedded applications. The steps on how to port the libraries to a desired target platform and to include them into the build chain are described.

The host library provides high-level interfaces to the OPTIGA™ Authenticate NBT. While the library is designed to be platform-independent, it relies on a number of subsidiary modules, some of which require platform-dependent functionality. These modules (for example, I2C transmit/receive) are separated from the host library through specified interfaces. The platform-dependent interface implementations must be available on the desired target platform to utilize the host library's functionality (for example, the I2C functions on a PSoC™ microcontroller).

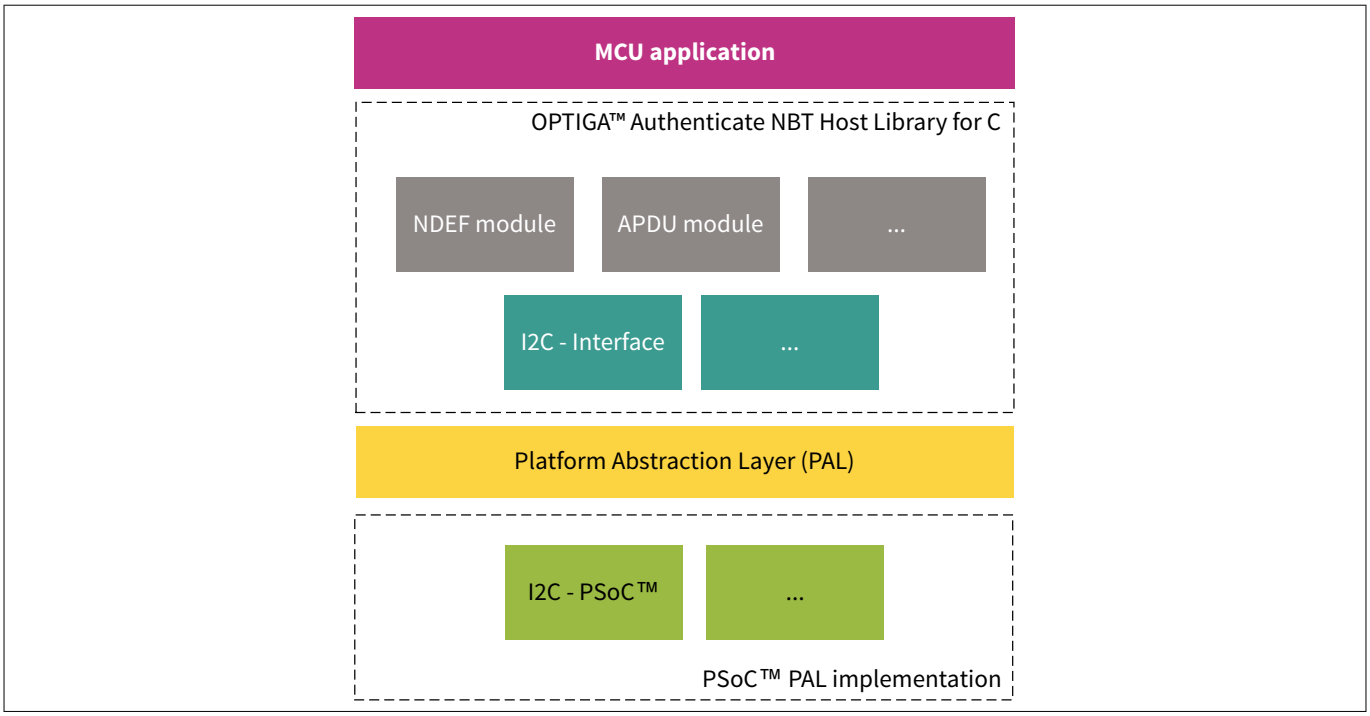


Figure 6 Platform-independent and platform-dependent parts of an MCU implementation

As illustrated in Figure 6, the high-level modules themselves are platform-independent. However, there may be dependencies that require access to platform-specific features (for example, I2C communication). To enable quick and easy porting of the library, a common platform abstraction layer (PAL) is used, which defines the interface for low-level functionality. For this reason, to port the library to an arbitrary MCU platform, a PAL is defined and must be implemented for that particular platform.

Integrating into custom applications

In most cases, a design-in process will include the development of a custom application, which uses one or more of the OPTIGA™ Authenticate NBT's core capabilities. The utilized modules of the host library will be selected in accordance with the target application's requirements. The product's use case guides [5][6][7][8] can support to identify the desired core capabilities and identify the necessary modules of the host library.

In order to develop custom applications, developers can utilize the host library and follow the steps below:

- Download the C host library from the GitHub
- Include the library in an existing or a new MCU project
- Identify the library's PAL interface and implement the required platform-dependent functions
- Include the desired modules in the MCU application and interact with the device using the host library's APIs

The host library is composed to support building via CMake scripts and thus, can be directly included in any custom CMake-based application. If the desired toolchain does not provide CMake support, the library can be included into custom projects as follows:

- Copy all used library folders (and their dependencies) to a common folder within the MCU project
- Add the folders to the compiler (may be skipped if the build system performs auto-discovery)

For more information on this procedure, refer to the respective documentation on GitHub [2].

Host library usage

This section provides an overview of the typical usage of the OPTIGA™ Authenticate NBT Host Library for C on various MCU platforms. Information can also be gathered by examining the provided example applications.

To communicate with the device and to use its functionality, it is required to configure a set of independent components and assemble them together to form a complete library stack.

The steps required to initialize these components are as follows:

- Initialize the low-level drivers (mostly the I2C driver)
- Initialize the protocol stack with the concrete PAL driver implementation (for example, I2C driver for PSoC™ using GP T=1” Protocol)
- Activate a communication channel to the OPTIGA™ Authenticate NBT using the protocol stack (protocol parameter negotiation)
- Initialize the device abstraction with the communication protocol stack

Once the OPTIGA™ Authenticate NBT abstraction is successfully initialized, use different command sets (for example, personalization, operational) to implement the required use case. For more information about the different command sets' APIs, refer to the documentation provided with the library package on GitHub [2].

Integration via ModusToolbox™

The C host library is also deployed as a middleware library via the ModusToolbox™ ecosystem. This simplifies the process of integrating the host library into ModusToolbox™ projects. In this case, the library can be easily added to an existing project via the IDE's library manager.

The ModusToolbox™ version of the C host library extends the generic C host library by implementing a PAL layer for various PSoC™ microcontrollers and by several ModusToolbox™-specific amendments (for example dedicated documentation). For more information on how to use the host library for ModusToolbox™, refer to the library's attached documentation, directly available within the IDE.

3.3 Java host library integration

This section provides a high-level overview on how to integrate the Java host library into Java-based applications. As described in previous sections, Infineon provides a dedicated host library to support the simplified interaction with the OPTIGA™ Authenticate NBT on Java-based platforms. The main focus of this library is the integration into Android applications that use the mobile phone's NFC interface to interact with the device.

Figure 7 illustrates the basic architecture of the host library for Java. The main purpose of the library is to provide a high-level abstraction of the OPTIGA™ Authenticate NBT's functionality via platform-independent Java libraries.

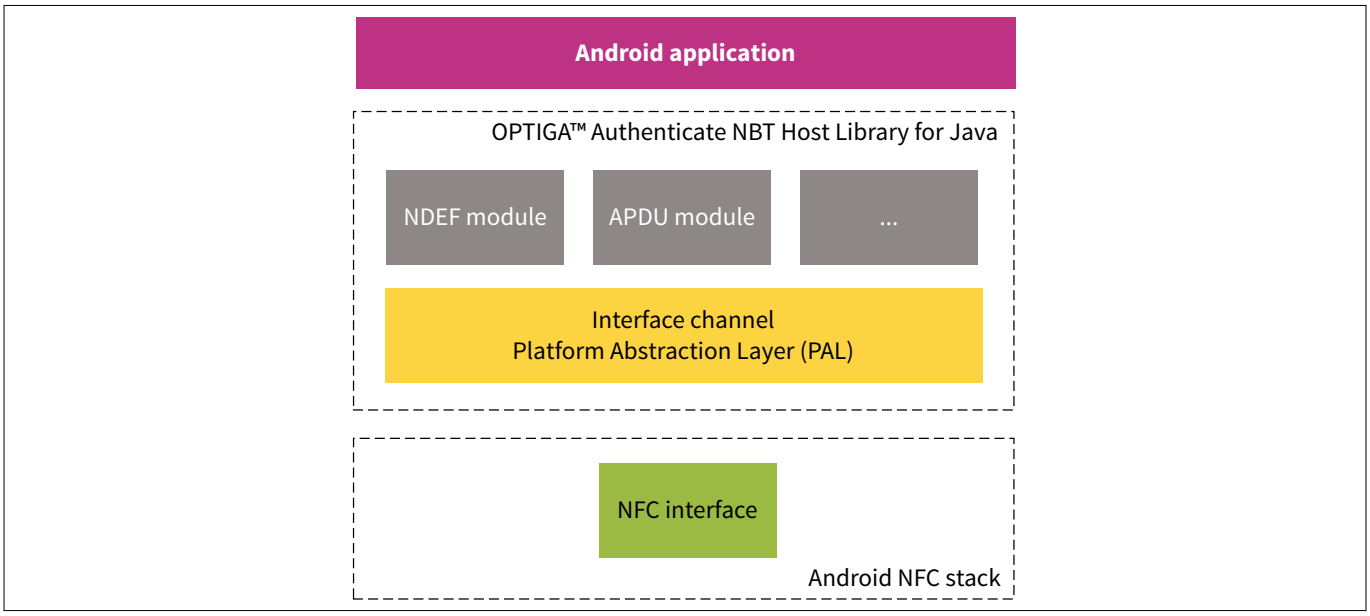


Figure 7 Java host library architecture in an Android application

The interface channel (included in the `com.infineon.hsw.channel` module) acts as the PAL for the APDU interface and allows easy integration across multiple platforms. In the case of Android, the interface channel is used to

provide access to the NFC interface abstraction and its low-level functionality. For more information on how to use the Android-specific NFC functionality, refer to [11].

Application developers can include the provided libraries in their Java projects to use the OPTIGA™ Authenticate NBT's command abstraction API. The individual components of the Java host library are provided as source code and can easily be included in any Java project. Depending on the platform and compiler, there may be different ways to include the Java host library.

For more information on the Java host library's functionality and detailed guidance on how to integrate it in Android applications, refer to the library's documentation on GitHub [2].

3.4 Swift host library integration

This section gives a high-level overview on how to integrate the Swift host library into iOS applications. The library is intended to be used in iOS applications for mobile, NFC-enabled devices (for example iPhones). The Swift host library enables the simplified development of applications which interact with the OPTIGA™ Authenticate NBT.

Figure 8 illustrates the basic architecture of the host library for Swift. The main purpose of the library is to provide a high-level abstraction of the OPTIGA™ Authenticate NBT's functionality via platform-independent Swift libraries.

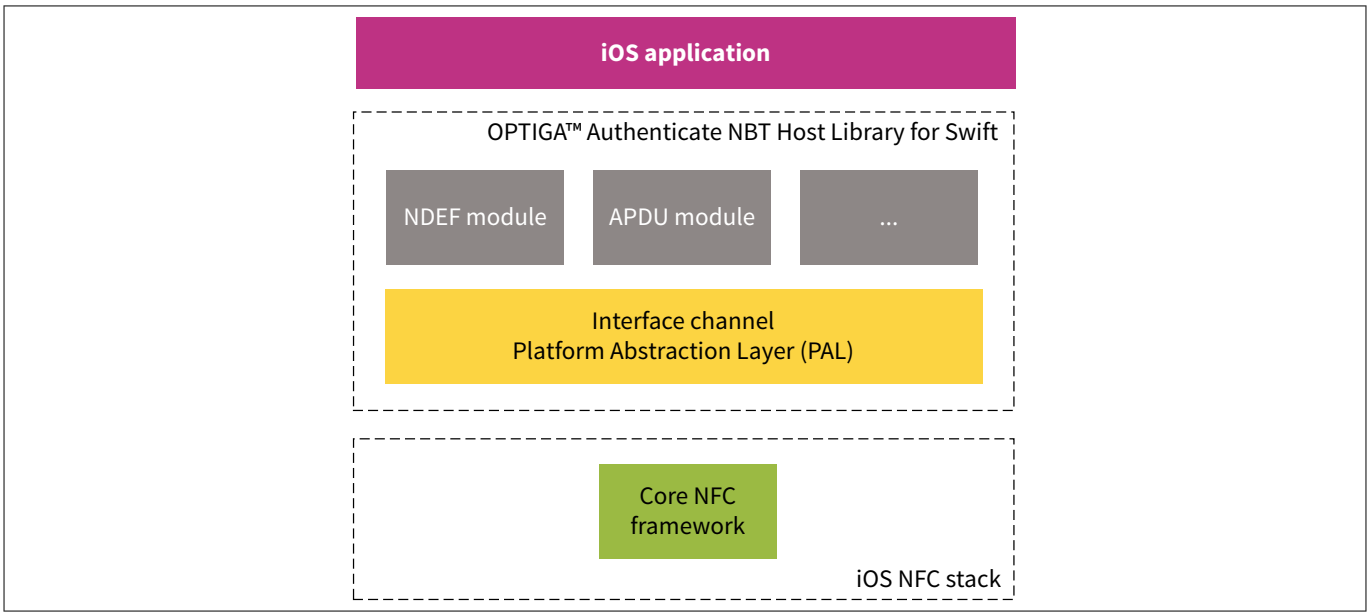


Figure 8 Swift host library architecture in an iOS application

The interface channel (included in the `InfineonChannel` package) acts as the PAL for the APDU interface and allows easy integration across multiple platforms. In the case of iOS, the interface channel is used to provide access to the NFC interface abstraction and its low-level functionality. For more information on how to use the iOS-specific NFC functionality, refer to [12].

iOS application developers can include the provided libraries in their Swift-based Xcode projects to use the OPTIGA™ Authenticate NBT's command abstraction API. The package dependency can be added as local dependency after downloading, or by defining the URL of the host library package's Git repository. Once the package is integrated, its modules can be imported and its functionality used in the project.

Depending on the Xcode version, there may be different ways to include and import Swift package files. For the latest official instructions on adding package dependencies to an application using Xcode, refer to [13]. For more information on the Swift host library's functionality and detailed guidance on how to integrate it into applications, refer to the library's documentation on GitHub [2].

References

Infineon

[1] Infineon Technologies AG: *OPTIGA™ Authenticate NBT, product website* - https://www.infineon.com/OPTIGA-Authenticate-NBT

[2] Infineon Technologies AG: *OPTIGA™ Authenticate NBT, GitHub overview repository* - github.com/Infineon/optiga-nbt

[3] Infineon Technologies AG: *OPTIGA™ Authenticate NBT, Release Notes (latest revision)*

[4] Infineon Technologies AG: *OPTIGA™ Authenticate NBT, Extended Datasheet (latest revision)*

[5] Infineon Technologies AG: *Host parameterization via asynchronous data transfer (ADT), Use Case Guide (latest revision)*

[6] Infineon Technologies AG: *Host parameterization via pass-through (PT), Use Case Guide (latest revision)*

[7] Infineon Technologies AG: *Static connection handover, Use Case Guide (latest revision)*

[8] Infineon Technologies AG: *Brand protection, Use Case Guide (latest revision)*

[9] Infineon Technologies AG: *OPTIGA™ Authenticate NBT Development Kit* - https://www.infineon.com/OPTIGA-Authenticate-NBT-Dev-Kit

[10] Infineon Technologies AG: *OPTIGA™ Authenticate NBT Development Shield* - https://www.infineon.com/OPTIGA-Authenticate-NBT-Dev-Shield

Others

[11] Google: *Android-specific NFC functionality* - developer.android.com/develop/connectivity/nfc; accessed: 2024-04-26

[12] Apple: *iOS-specific NFC functionality* - developer.apple.com/design/human-interface-guidelines/nfc; accessed: 2024-04-26

[13] Apple: *Xcode* - developer.apple.com/documentation/xcode/adding-package-dependencies-to-your-app; accessed: 2024-04-26

Glossary

APDU

application protocol data unit (APDU)
The communication unit between a smart card reader and a smart card.

API

application programming interface (API)

GP

GlobalPlatform (GP)

I2C

inter-integrated circuit (I2C)

IDE

integrated development environment (IDE)

A software application that combines multiple tools used for software development into a single environment.

iOS

iPhone operating system (iOS)

A mobile operating system created and developed by Apple Inc. exclusively for its hardware.

IoT

Internet of Things (IoT)

MCU

microcontroller unit (MCU)
One or more processor cores along with memory and programmable input/output peripherals.

NDEF

NFC data exchange format (NDEF)

A standardized data format specification by the NFC Forum to describe how a set of actions are to be encoded onto a NFC tag or to be exchanged between two active NFC devices.

NFC

near field communication (NFC)

PAL

platform abstraction layer (PAL)

PSoC™ microcontroller

A range of general-purpose MCUs built on an ultra-low-power architecture ideal for battery-operated, low-power applications including embedded IoT applications.

Revision history

Reference	Description
Revision 1.1, 2024-04-26	
All	Editorial changes
Revision 1.0, 2024-03-28	
All	Initial release

Trademarks

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

Edition 2024-04-26

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2024 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email:
CSSCustomerService@infineon.com

Document reference
IFX-akg1695104490379

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.

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.