

SDK/J Authentication Package v1.0 User's Guide

SDK/J Authentication Package Version:1.0



Copyright © 2011 Ricoh Co., Ltd.

Terms of Use and Trademarks

1. The contents of this book may be changed without notice in the future.
2. The copying, reproducing, changing, quoting, reprinting, or distributing a part/all of this book are prohibited.
3. We make no warranty, express or implied, regarding this document and the sample codes described in this document. We will not be held responsible for any of our customer's losses, damages resulting from lost profits, or claims from any third party on using this document and the sample codes described in this document.

4. Trademarks

PostScript® and Acrobat® are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Java, JVM (CVM) and CDC are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Eclipse is a trademark of eclipse.org in the United States, other countries, or both.

OSGi(TM) is a trademark, registered trademark, or service mark of The Open Services Gateway Initiative in the US and other countries.

Apache is a registered trademark of The Apache Software Foundation in the United States, other countries, or both.

Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights in those marks.

Contents

1. Introduction	2
2. Target Readers.....	3
3. Software Requirements.....	4
4. Architecture.....	5
5. Features	7
5.1. Smart Card Access.....	7
5.2. Device Access	7
5.3. Network Authentication.....	8
5.4. Panel Service	8
5.5. Authentication Service.....	8
6. Authentication using a card	9
6.1. Card Reader Driver Support.....	10
6.2. Other Support	10
6.3. Card Support	10
7. Card + User Authentication Support	11
7.1. Enhanced Authentication Support.....	12
7.2. Enhanced + Customized Authentication Support.....	14
8. Supported card readers	16
8.1. PC/SC Compliant Card Readers.....	16
8.2. Non PC/SC Card Readers.....	17
Change History	18

1. Introduction

This document gives an introduction to the SDK/J Authentication Package v1.0 that provides a framework that simplifies the process of authentication of cards and external devices.

The framework can be used to access cards and external devices, obtain necessary data from them, perform external server authentication, and use user authentication function of MFP/LP from SDK/J.

This document includes important information on using the SDK/J Authentication Package v1.0.

Please read through this document before using the package, the framework, in your application development.

2. Target Readers

This document is intended for the application developers who wish to use the SDK/J Authentication Package in their application development, and therefore assumes that they are familiar with Java programming language and the SDK/J.

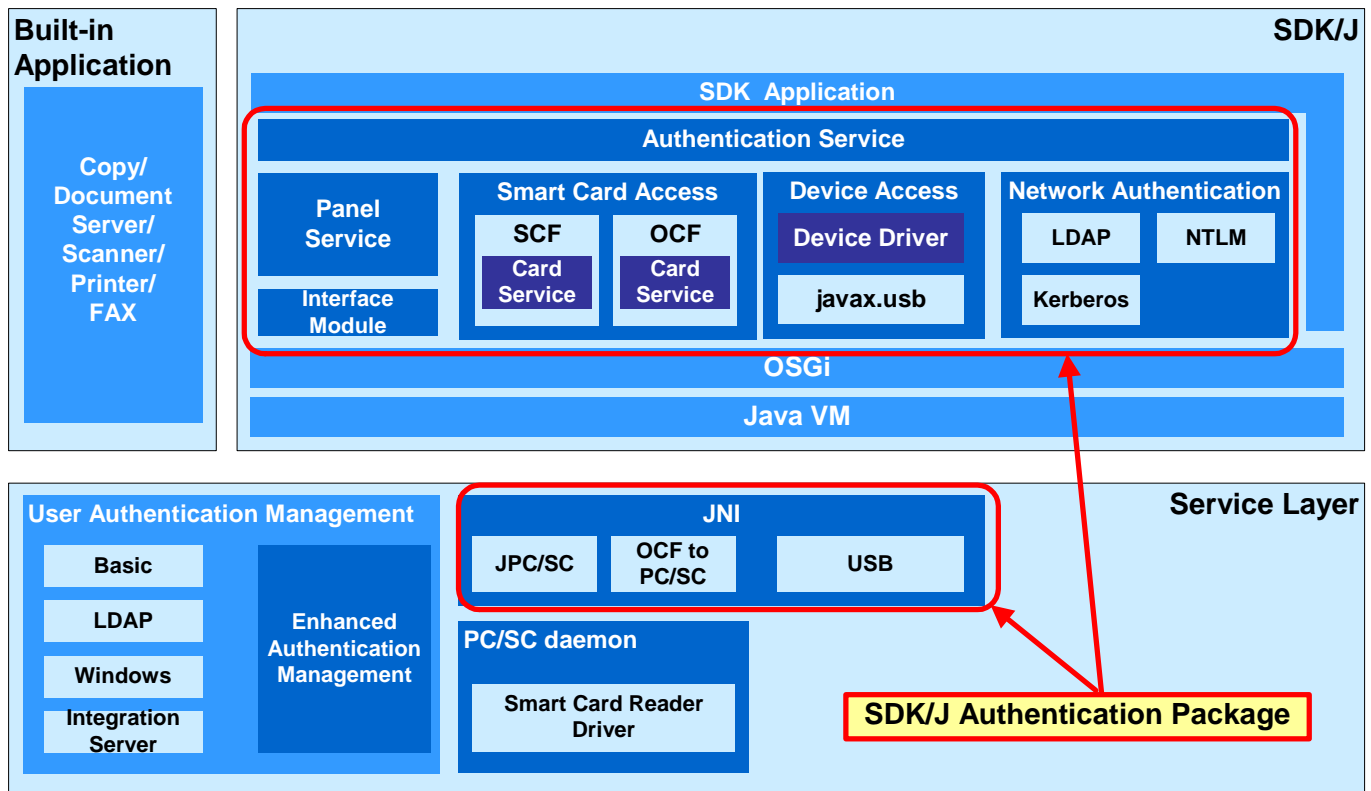
3. Software Requirements

Use of the SDK/J Authentication Package v1.0 requires an operation environment where:

- SDK/J is installed and operating properly.
- The settings for SDK/J Authentication Package are configured properly.

About the settings, please see “SDK/J Authentication Package Settings Guide”.

4. Architecture



Service Layer

JNI

JNI (Java Native Interface) allows Java codes to interact with native codes.

- "JPC/SC" provides JNI for accessing the PC/SC daemon from the SDK/J. The Smart Card Framework (SCF) uses JPC/SC.
- "OCF to PC/SC" provides JNI for accessing the PC/SC daemon from the SDK/J. The OpenCard Framework (OCF) uses OCF to PC/SC.
- "USB" provides JNI for accessing USB ports from SDK/J. SDK/J Authentication Package supports javax.usb package which is compliant with JSR80.

PC/SC daemon

"PC/SC daemon" provides CardReaderDrivers to access smart cards.

User Authentication Management

"User Authentication Management" is a module which manages the user authentication function of an MFP/LP. This module provides Basic authentication, LDAP authentication, Windows authentication and Integration Server authentication as built-in user authentication methods.

Enhanced Authentication Management

"Enhanced Authentication Management" is a module which provides I/F to use the login/logout feature of User Authentication Management from PanelService of the SDK/J Authentication Package.

Built-in Application

“Built-in Application” represents the features (applications) of an MFP/LP such as Copy, Scan, FAX, Printer, etc.

SDK/J

JavaVM, OSGi

“JavaVM” and “OSGi” together represent the standard SDK/J Platform supported by the SDK/J.

In other words, the standard SDK/J Platform is comprised of “JavaVM” and “OSGi”.

Authentication Package

“Authentication Package” represents the authentication libraries offered by the SDK/J.

This package is provided as an optional package.

Smart Card Access

“Smart Card Access” represents the library that is used to access smart cards.

This library supports PC/SC compliant USB smart card readers.

Device Access

“Device Access” represents the library that is used to access USB devices.

In order to access USB devices, Device Drivers must be implemented by using javax.usb.

Network Authentication

“Network Authentication” provides authentication methods such as LDAP, Kerberos and NTLM.

PanelService

“PanelService” represents the library that is used to use the login/logout feature of User Authentication Management from SDK/J Applications using I/F provided by Enhanced Authentication Management.

InterfaceModule

“InterfaceModule” represents an I/F between PanelService and Enhanced Authentication Management.

Authentication Service

“Authentication Service” represents an abstract service that is comprised of any combination of Card Access, Network Authentication, and Panel Service, and provides an authentication feature to SDK applications.

5. Features

5.1. Smart Card Access

The Smart Card Access offers access methods for smart cards.

SCF (SmartCard Framework)

The SCF is a Java API for accessing smart cards. The SCF is implemented using JPC/SC.

The smart cards that are compliant with ISO 7816, ISO 14443 are supported.

For details, see the SmartCard Framework Developer's Guide.

OCF (OpenCard Framework)

The OCF is a framework offered by the OpenCard Consortium.

The smart cards that are compliant with ISO 7816, ISO 14443 are supported.

For details, see the OpenCard Framework Developer's Guide.

5.2. Device Access

The Device Access offers access methods for USB devices. Implement the driver for the USB device you wish to use by using javax.usb.

As sample codes for javax.usb, Authentication Package provides sample drivers for some USB card readers.

You need to choose either "Smart Card Access" or "Device Access" depending on the type of the USB device you wish to use. For details, see chapter 8.

Note: The operation will not be guaranteed when more than one card reader device or the other devices are connected at the same time.

USB Device Type	Access Method	Memo
PC/SC compliant smart card readers	SCF or OCF	Can not use both SCF and OCF at the same time.
Other	Implement the driver using javax.usb.	

5.3. Network Authentication

The SDK/J offers the LDAP and the Kerberos that are equivalent to those packages offered by SunMicrosystems as its optional packages.

Note: - Use of the LDAP requires the JAAS and the JNDI.

- Use of the Kerberos requires the JCE.

LDAP and Kerberos libraries are deployed in the SDK/J.

Details on these packages are available at SunMicrosystems' web site.

The NTLM is available as an open source product and can be downloaded from the web site of JCIFS (<http://jcifs.samba.org>).

5.4. Panel Service

“PanelService” will unlock/lock the operation panel when User Authentication Management and Enhanced Authentication Management are set on an MFP/LP.

For details, see the Panel Service User's Guide and the Panel Service Developer's Guide.

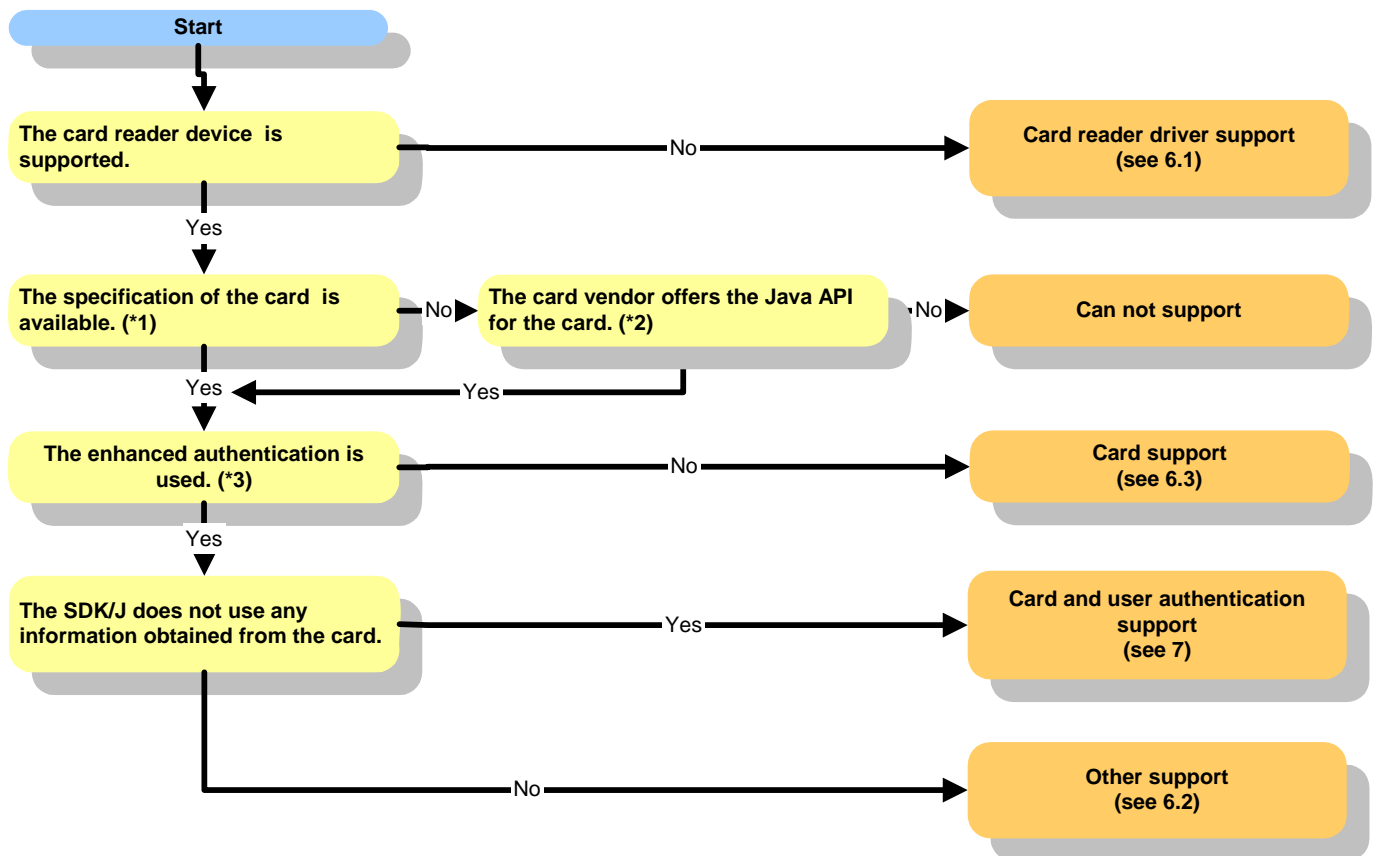
5.5. Authentication Service

The Authentication Service is, as it is mentioned earlier, an abstract service that is comprised of any combination of SmartCard Access, Network Authentication, and Panel Service.

The purpose of this abstract service is to provide an authentication feature to an SDK application, hiding the underlying components. Therefore, if there is no need to hide the underlying components, you don't have to implement this service.

6. Authentication using a card

You can find out the necessary development step(s) by using the following chart.



(*1) To obtain information of a card, the specification of the card and the format of the data are necessary.

(*2) Use the SDK/J when the Java API that is necessary to access the card is offered by the vender.

(*3) The enhanced authentication function of an MFP/LP allows for authentication using a card on the MFP/LP.

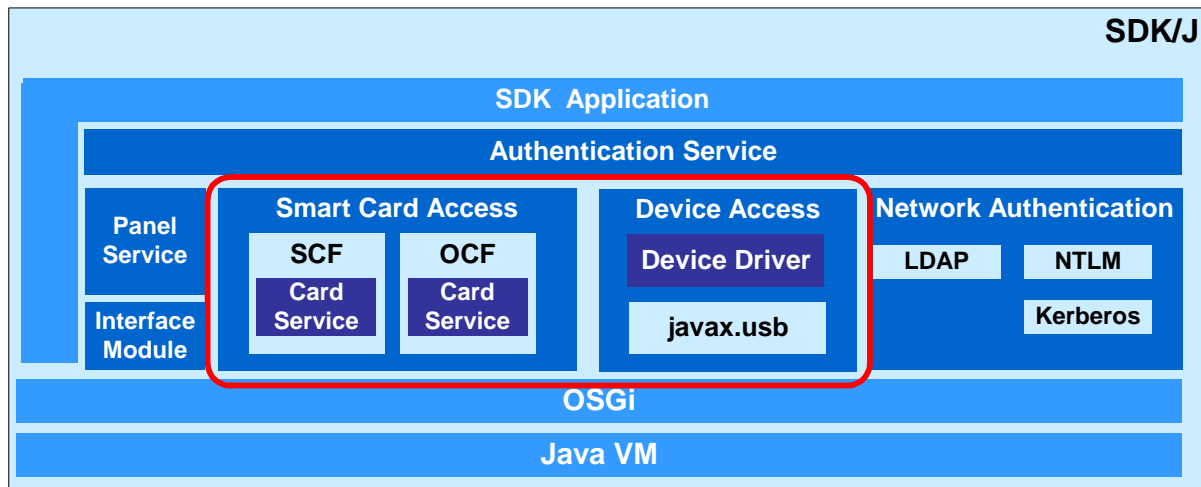
6.1. Card Reader Driver Support

See chapter 8.

6.2. Other Support

For details, contact RiDP.

6.3. Card Support



In this case, specify one access method from SCF, OCF, and javax.usb and then implement the feature to access your target card.

The SCF and the OCF offer sample CardServices for JavaCard cards and Cryptoflex cards.

The javax.usb offers sample for some USB card readers.

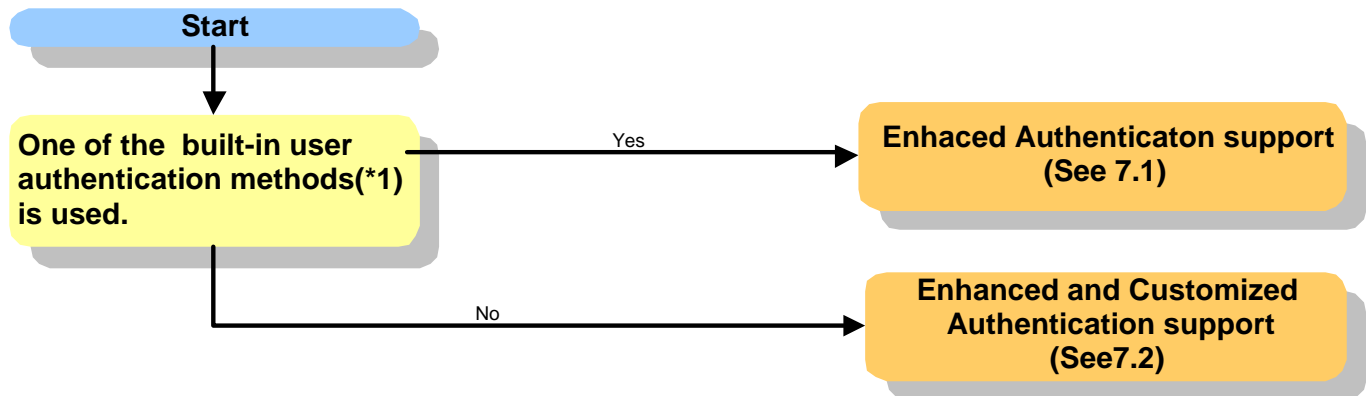
7. Card + User Authentication Support

The process you need to implement differs depending on the authentication method you use.

See the following chart.

If you use one of the built-in authentication methods, proceed to step 7.1.

If you use other authentication method than the built-in authentication methods, proceed to step 7.2.



(*1) The built-in user authentication methods available on MFPs/LPs:

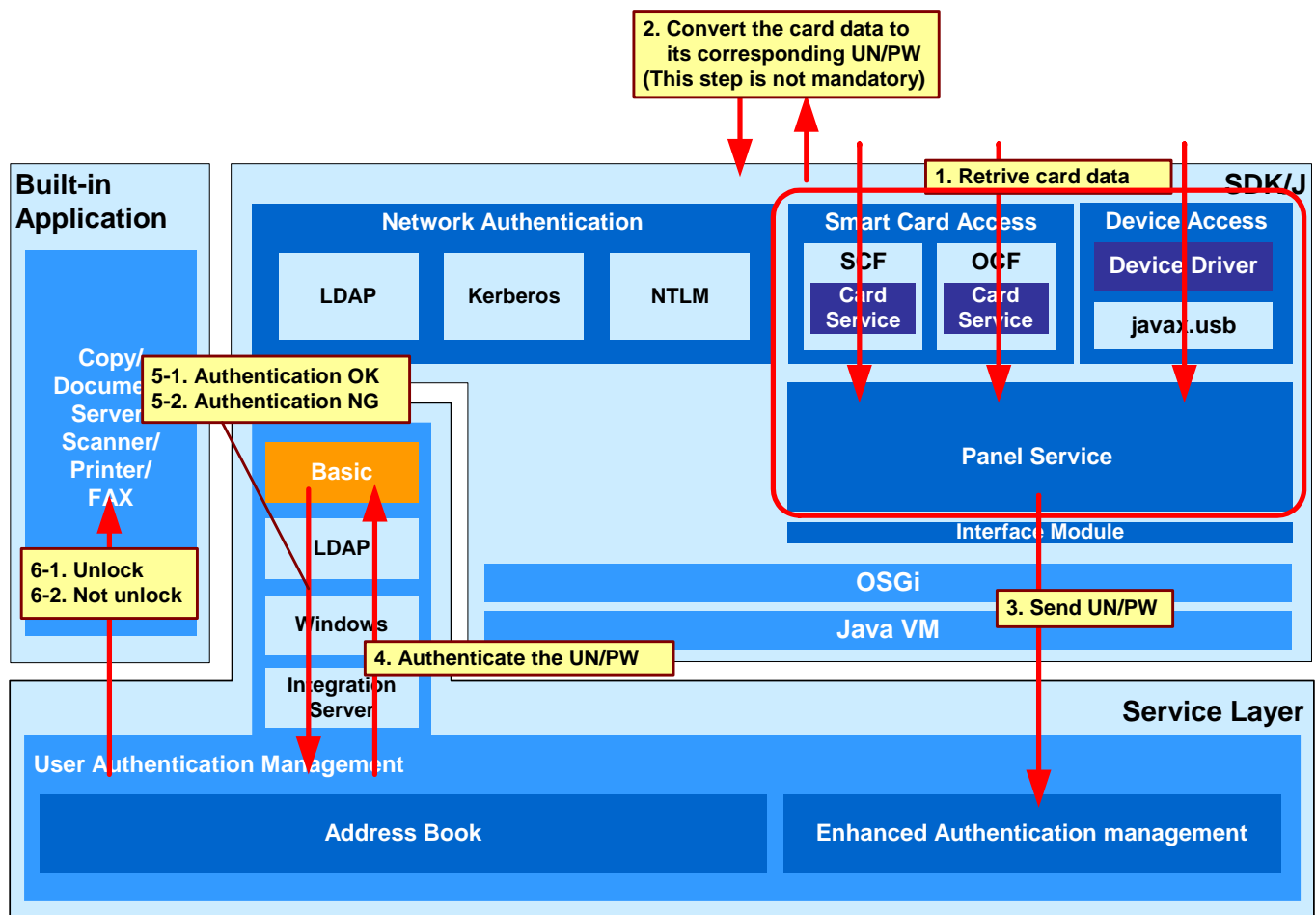
- Basic authentication
- LDAP authentication
- Windows authentication
- Integration server authentication

For details, see the user's guide for each device.

If you wish to use other user authentication method than these built-in user authentication methods, implement it by using the SDK/J Authentication Customization feature, which is a framework that allows you to implement user-specific user authentication methods in SDK/J.

Please see SDK/J developer's guide for details..

7.1. Enhanced Authentication Support



- In this case, you need to implement the process for obtaining the necessary data from the card and providing the obtained information to the User Authentication management.
- Red in the picture above indicates the essential modules that must be implemented.
- The Panel Service must be available on the device.

1. Retrieve card data

The necessary data is obtained from the card by using either "Smart Card Access" or "Device Access".

For details, see the "Smart Card Framework Developer's Guide", "OpenCard Framework Developer's Guide", and "javax.usb Developer's Guide".

2. Convert the card data to its corresponding UN/PW

If the obtained data is not used as is as the login user name and login password, it is converted into the corresponding UN/PW through an appropriate method. For instance, it is possible to obtain from the LDAP server the user name by using the card serial number.

3. Send UN/PW

The login user name and login password are sent to the User Authentication Management by using the Panel Service.

For details, see the “Panel Service Developer’s Guide”.

4. Authenticate the UN/PW

With the login user name and login password that have been passed from the Panel Service, user authentication is performed by the built-in user authentication module.

5. Authentication OK/NG

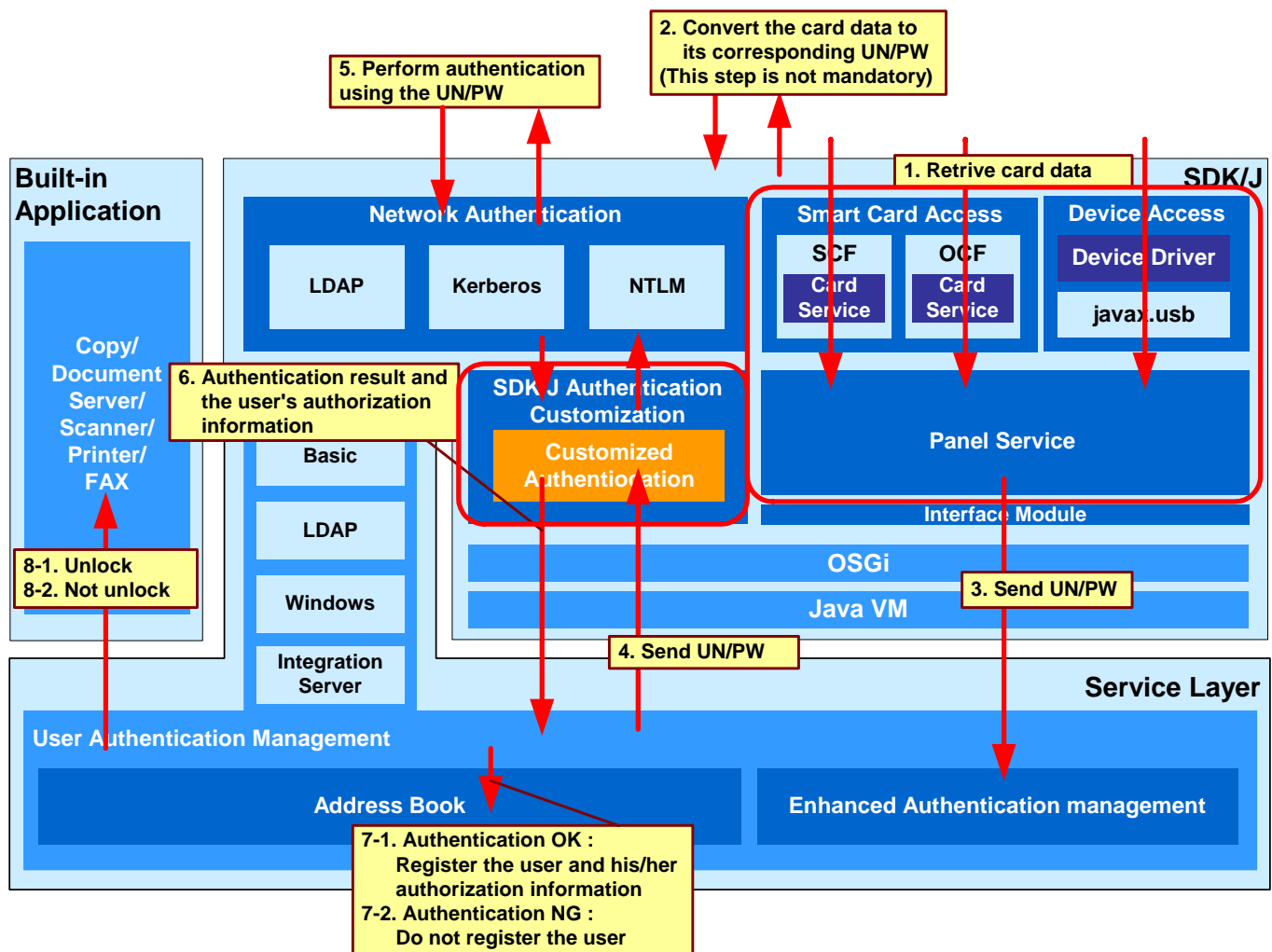
The user authentication result is determined by the built-in user authentication module.

6. Unlock/not unlock

If the user is authenticated, that is, the result of user authentication is “OK”, the operation panel of the device is unlocked, and the features on the device become available to the user based on the user’s authorization information registered with the address book of the target device.

If the user is not authenticated, that is, the result of user authentication is “NG”, the operation panel of the device remains locked.

7.2. Enhanced + Customized Authentication Support



- In this case, you need to implement the process for obtaining the necessary data from the card, providing the obtained information to the User Authentication management, and performing user authentication by the Extended Authentication application, which is implemented by using SDK/J Authentication Customization feature.
- Red in the picture above indicates the essential modules that must be implemented.
- The Panel Service and the SDK/J Customized Authentication must be available on the device.

1 - 3.

Same as "7.1 Enhanced Authentication Support."

4. Send UN/PW

The login user name and login password are sent to the Customized Authentication application, which is implemented by using SDK Authentication Customization feature.

5. Perform authentication using the UN/PW

With the login user name and login password that have been passed from the User Authentication Management, the Customized Authentication application performs user authentication. Libraries for network authentication, etc. may be used.

If the user is authenticated, the authorization information of the user is set.

6. Authentication result and the user's authorization information

The user authentication result and the authorization information of the user are returned to the User Authentication Management.

7. Address book update

If the user is authenticated, that is, the result of user authentication is "OK", the user is registered with the address book with the necessary information including authorization information, etc., or if the user is already in the address book, the record is updated with the new information.

If the user is not authenticated, that is, the result of user authentication is "NG", the user is not registered with the address book. No update to the address book is performed.

8. Unlock/not unlock

If the user is authenticated, the operation panel of the device is unlocked, and the features on the device become available to the user based on the user's authorization information registered with the address book of the target device.

If the user is not authenticated, the operation panel of the device remains locked.

8. Supported card readers

8.1. PC/SC Compliant Card Readers

PC/SC daemon uses following libraries provided by MUSCLE project as card reader drivers:

- pcsc-lite 1.2.9 beta 10 *
- ccid 1.1.0 *

The operation of the card reader has been confirmed with the following card reader devices:

Contact Type

Compliant	Supported Card Readers		For Information
	Manufacturer	Model	
ISO7816	Gemalto	GemPC Twin	http://www.gemalto.com/
		Reflex USB v3	
		GemPC Key (product id 0x3438)	
	Omnikey	CardMan 3121	http://www.omnikey.com/
	SCM	SCR 331	http://www.scmmicro.com/
	Microsystems	SDI 010 (contact slot only)	
	C3PO	LTC31 (product id 0x0006)	http://www.c3po.es/

If the card reader device you wish to use is not in the list above, contact RiDP.

* For details of pcsc-lite 1.2.9 beta10 and ccid 1.1.0, see the web site below :

<http://pcsc-lite.alioth.debian.org/>

The source code of ccid 1.1.0 is available from the web site below :

<http://anonscm.debian.org/viewvc/pcsc-lite/tags/ccid/rel-1.1.0/>

8.2. Non PC/SC Card Readers

You can develop a card reader driver using javax.usb. For details, see “javax.usb Developer's Guide”.

AuthenticationPackage provides sample drivers for the following devices :

Type	Compliant	Card Readers		More Information
		Manufacturer	Model	
Contact-less	125kHz Proximity	RFIDeas, Inc.	pcProx USB	http://www.rfideas.com/
			>RDR-6081AKU	
	>RDR-6281AKU			
	>RDR-6381AKU			
			>RDR-6N81AKU	
	13.56MHz	RFIDeas, Inc.	Air ID Enroll USB (Mifare)	http://www.rfideas.com/
			>RDR-7581AKU	
			RFID1356i Enroll USB (iClass, Mifare)	
			>RDR-7081AKU	
		Interflex	IF 72 (Legic)	http://www.interflex.de/
Contact	Magstripe	Tysso	TMSR-33	http://www.tysso.com/

Change History

Ver.1.0_R103	8. Supported card readers 8.1. PC/SC Compliant Card Readers Updated the source code of ccid 1.1.0 web sites.
Ver.1.0_R102	Based on the document included in SDK/J kit R1.02 Updated the card reader manufacturer web sites in “8. Supported card readers” section.