


# NFC Secure Element Stepping Stones

Version 1.0

Published by  **simalliance** now Trusted Connectivity Alliance

July 2013

# Table of Contents

|   |    |
|---|----|
| 1. Introduction .....   | 7  |
| 1.1 Acknowledgements .....  | 7  |
| 1.2 Reference documentation .....   | 7  |
| 1.3 Abbreviations .....   | 12 |
| 2. Device and service deployment .....  | 13 |
| 2.1 'Issuer-centric' model and 'consumer-centric' model .....   | 13 |
| 2.2 Actors and roles .....  | 14 |
| 2.3 Architecture .....  | 16 |
| 2.3.1 'Issuer-centric', vertical model .....  | 16 |
| 2.3.2 'Consumer-centric', open model .....  | 17 |
| 2.3.3 'Renting', intermediate model .....   | 18 |
| 2.4 Use-cases .....   | 19 |
| 2.4.1 NFC banking service in UICC: an example of 'issuer-centric' model .....                         | 19 |
| 2.4.2 NFC banking service in smart micro-SD: an example of 'issuer-centric' model .....               | 20 |
| 2.4.3 Transportation service and loyalty in Embedded SE: an example of 'consumer-centric' model ..... | 21 |
| 2.5 Device deployment .....   | 22 |
| 2.6 Service Deployment .....  | 24 |
| 2.6.1 Vertical model - 'issuer-centric' .....   | 24 |
| 2.6.2 Open model - 'consumer-centric' .....   | 25 |
| 2.7 Unique SE identifier .....  | 26 |
| 2.8 Credentials description .....   | 26 |
| 3. Secure Element general architecture .....  | 28 |
| 3.1 Different SE form factors .....   | 28 |
| 3.1.1 UICC .....  | 28 |
| 3.1.2 Embedded Secure Element (eSE) .....   | 30 |
| 3.1.3 Smart microSD .....   | 32 |
| 3.1.4 NFC smart microSD .....   | 33 |
| 3.1.5 NFC standalone smart microSD .....  | 33 |
| 3.2 Architecture for Embedded SE, smart microSD & UICC .....  | 35 |
| 3.3 Specifications .....  | 36 |
| 4. Abstracts on the possible SE communication buses .....   | 39 |
| 4.1 SWP .....   | 40 |
| 4.2 The SPI bus .....   | 41 |
| 4.3 The I2C bus .....   | 42 |

|       |  |    |
|-------|--|----|
| 4.3.1 | <i>SPI versus I2C</i> .....  | 43 |
| 4.4   | The NFC-WI bus (also called ECMA 373 and S2C).....                           | 43 |
| 4.5   | The Inter-chip USB protocol .....  | 45 |
| 4.6   | The DCLB protocol.....   | 46 |
| 4.7   | The SD bus .....   | 46 |
| 5.    | Remote management .....  | 49 |
| 5.1   | Evolution of OTA protocols.....  | 49 |
| 5.2   | Secure Channel protocols overview .....                                      | 50 |
| 5.2.1 | <i>Secure Channel Protocol 02</i> .....                                      | 50 |
| 5.2.2 | <i>Secure Channel Protocol 80</i> .....                                      | 50 |
| 5.2.3 | <i>Secure Channel Protocol 81</i> .....                                      | 50 |
| 5.3   | The Secure Channel Protocol 02.....  | 50 |
| 5.3.1 | <i>SCP02 APDUs</i> .....   | 50 |
| 5.3.2 | <i>SCP02 Keys</i> .....  | 51 |
| 5.4   | Stacking the protocols: SCP02 over SCP80/SCP81 .....                         | 52 |
| 5.5   | Third party application management .....                                     | 53 |
| 5.6   | Personalized SDs.....  | 54 |
| 5.7   | Two Card Content models.....   | 55 |
| 5.8   | The Admin Agent .....  | 55 |
| 6.    | Service development .....  | 57 |
| 6.2   | <i>Access Control</i> .....  | 58 |
| 6.3   | <i>The Mobile Wallet</i> .....   | 61 |
| 6.4   | <i>Role of the wallet in the SE</i> .....                                    | 62 |
| 6.4.1 | <i>The CRS Application</i> .....   | 62 |
| 6.5   | <i>Role of the wallet in the handset</i> .....                               | 62 |
| 6.6   | <i>General architecture of the Wallet</i> .....                              | 63 |
| 6.6.1 | <i>General architecture of the Core Wallet and the Extended Wallet</i> ..... | 63 |
| 6.7   | <i>Wallet logic to choose SP Application</i> .....                           | 64 |
| 6.8   | <i>Interaction with Secure Elements</i> .....                                | 65 |
| 6.9   | <i>User Interaction</i> .....  | 65 |
| 7.    | Security Certifications .....  | 66 |
| 7.1   | Introduction .....   | 66 |
| 7.2   | Payment System Type Approval Process.....                                    | 66 |
| 7.2.1 | <i>Parties involved in a Type Approval Process</i> .....                     | 67 |
| 7.2.2 | <i>Roles and Responsibilities</i> .....                                      | 67 |
| 7.2.3 | <i>Type Approval Process in general</i> .....                                | 67 |
| 7.3   | EMVCo Card Type Approval .....   | 68 |
| 7.3.1 | <i>Introduction</i> .....  | 68 |
| 7.3.2 | <i>EMVCo SE components</i> .....   | 68 |
| 7.3.3 | <i>Components of a EMVCo Secure-Element</i> .....                            | 69 |
| 7.3.4 | <i>Certification Validity and Prolongation</i> .....                         | 70 |

7.3.5 Impact of Product Changes ..... 70

7.3.6 Security Monitoring ..... 70

7.3.7 Public Certification Information ..... 70

7.4 Common Criteria Evaluation..... 70

7.4.1 Introduction ..... 70

7.4.2 CC Assurance Levels ..... 72

7.4.3 Protection Profile..... 72

7.4.4 Security features ..... 73

7.4.5 Cryptographic key management ..... 73

7.4.6 Cryptographic operations ..... 73

7.4.7 Hardware related evaluation ..... 74

7.4.8 Hardware attack scenarios and countermeasures ..... 74

7.4.9 Validity of CC Certificates ..... 75

7.4.10 Links to Certificates and PPs ..... 75

# Figure index

|  |    |
|--|----|
| FIGURE 1: ARCHITECTURE – DEPLOYMENT – ‘ISSUER-CENTRIC’ .....   | 16 |
| FIGURE 2: ARCHITECTURE – DEPLOYMENT – ‘CONSUMER-CENTRIC’ .....   | 17 |
| FIGURE 3: ARCHITECTURE – DEPLOYMENT – ‘RENTING’ MODEL .....  | 18 |
| FIGURE 4: NFC BANKING SERVICE IN UICC: AN EXAMPLE OF THE ‘ISSUER-CENTRIC’ MODEL .....                      | 20 |
| FIGURE 5: NFC BANKING SERVICES DELIVERED VIA SMART MICRO-SD: AN EXAMPLE OF AN ‘ISSUER-CENTRIC’ MODEL. .... | 21 |
| <b>FIGURE 6: DEVICE – DEPLOYMENT</b> .....   | 22 |
| FIGURE 7: SERVICE – DEPLOYMENT – ‘ISSUER-CENTRIC’ .....  | 24 |
| <b>FIGURE 8: SERVICE – DEPLOYMENT – ‘CONSUMER-CENTRIC’</b> .....   | 25 |
| <b>FIGURE 9: UICC SE ARCHITECTURE IN A SMART PHONE</b> .....   | 29 |
| <b>FIGURE 10 EMBEDDED SE ARCHITECTURE IN A SMART PHONE</b> .....   | 31 |
| <b>FIGURE 11: SMART MICROSD SE ARCHITECTURE IN SMART PHONES</b> .....                                      | 33 |
| <b>FIGURE 12: STANDALONE NFC SMART MICROSD</b> .....   | 34 |
| <b>FIGURE 13 : LAYERED STACKING OF SE/CLF PROTOCOLS</b> .....  | 40 |
| <b>FIGURE 14: NFC SE IN A HANDSET</b> .....  | 40 |
| <b>FIGURE 15: SPI BUS: SINGLE MASTER, MULTIPLE SLAVES</b> .....  | 41 |
| <b>FIGURE 16: I2C - A TWO-WIRE INTERFACE</b> .....   | 42 |
| <b>FIGURE 17: I2C COMMUNICATION</b> .....  | 42 |
| <b>FIGURE 18: THE NFC-WI INTERFACE</b> .....   | 43 |
| <b>FIGURE 19: MODIFIED MILLER CODING SCHEME (WITH A 100% MODULATION INDEX)</b> .....                       | 44 |
| <b>FIGURE 20: MANCHESTER CODING SCHEME (EXAMPLE: OR COMBINED WITH FcLOCK/16)</b> .....                     | 45 |
| FIGURE 21: THE DCLB INTERFACE .....  | 46 |
| FIGURE 22: SWP SUPPORT IN THE MICROSD .....  | 47 |
| <b>FIGURE 23: ALTERNATE CONFIGURATION FOR SWP SUPPORT IN MICROSD</b> .....                                 | 48 |
| FIGURE 24: SCP PROTOCOLS .....   | 52 |
| FIGURE 25: ARCHITECTURE OF ACCESS CONTROL .....  | 60 |
| FIGURE 26: EXAMPLE OF A WALLET ARCHITECTURE (GSMA PROPOSITION) .....                                       | 64 |
| FIGURE 27: PROCESS FOR CERTIFICATIONS .....  | 68 |

# Table index

|   |    |
|---|----|
| TABLE 1: ACTORS AND ROLES IN MOBILE NFC ECOSYSTEM ..... | 15 |
| TABLE 2: CREDENTIALS DESCRIPTION .....                  | 27 |
| TABLE 3: BRIEF FEATURES FOR DIFFERENT TYPES OF S .....  | 35 |
| TABLE 4: NFC SE ARCHITECTURE.....                       | 36 |
| TABLE 5: SPECIFICATIONS RELATED TO UICC .....           | 37 |
| TABLE 6: SPECIFICATIONS RELATED TO ESE.....             | 37 |
| TABLE 7: SPECIFICATIONS RELATED TO SMART MICROSD .....  | 38 |
| TABLE 8: CAT SE TRANSPORT PROTOCOLS .....               | 55 |

## Document History

| Version | Date       | Editor                | Remarks        |
|---------|------------|-----------------------|----------------|
| 1.0     | 29/07/2013 | Interop Working Group | Public release |

Copyright © 2013 SIMalliance Ltd.

The information contained in this document may be used, disclosed and reproduced without the prior written authorization of SIMalliance. Readers are advised that SIMalliance reserves the right to amend and update this document without prior notice. Updated versions will be published on the SIMalliance website at <http://www.simalliance.org/en/resources/recommendations/>

# 1. Introduction

Mobile near field communication (mobile NFC) is a short-range wireless communication technology which facilitates the exchange of data between mobile and other NFC enabled devices. This powerful new enabler defines important commercial scenarios for identification, payments and secure communication, particularly when security plays a fundamental role in establishing user trust.

The development of new NFC services brings a variety of new players into the NFC arena. Providers of payment, identification and other services are spawning opportunities to create new value on the secure element (SE). New players are creating opportunities and new partnerships, which together are driving an increasing need to establish interoperability across the mobile NFC ecosystem. In addition, new types of SE are now available for mobile devices, such as the embedded SE and the smart microSD, which have introduced yet more interoperability challenges.

In this document, the SIMalliance Interoperability Working Group aims to support SE interoperability, in order to enable service providers to smoothly migrate their applications between different SEs. Application providers should develop their services taking into account the full spectrum of SE architectures and this document highlights key differences of relevance to these organisations.

The document is part of the Stepping Stones family of documents, and is specifically intended as a compendium of previous versions of NFC Stepping Stones. Previously, NFC Stepping Stones focused on NFC services based on the UICC. This document builds on this scope and includes other forms of SEs, including Smart microSD and embedded Secure Element. The document is not a replacement for the UICC NFC Stepping Stones; it is complementary. Indeed this document makes reference to the UICC NFE Stepping Stones document when the technology relative to the UICC is discussed.

This technical document targets Mobile Operators, Service Providers, Handset and SE providers, together with all the participants in the NFC ecosystem who require access to SE architecture and technology.

## 1.1 Acknowledgements

There are many contributors to this document, the work of all of whom is greatly appreciated. Nonetheless, special thanks goes to those active in the SIMalliance Interoperability Working Group, including:

Eric Laffont (Comprion), Anne-Marie Praden (Gemalto), Michael Schnellinger, Nils Nitsch (G&D), Amedeo Veneroso (Incarnat STMicroelectronics), Gianluca Markos (Movenda) and Yolanda Sanz (VALID).

## 1.2 Reference documentation

Note: In this list, when the version is not explicitly mentioned the previous version is the relevant one.

| Entity | Reference                 | Title   |
|--------|---------------------------|---|
| ISO    | [1] <b>ISO/IEC 7816-3</b> | "Information technology – Identification cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols" |

|      |                      |  |
|------|----------------------|--|
|      | [2] ISO/IEC 7816-4   | "Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange"                                      |
|      | [3] ISO/IEC 13239    | "Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures"              |
|      | [4] ISO/IEC 14443-2  | "Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface"              |
|      | [5] ISO/IEC 14443-3  | "Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anti-collision"                       |
|      | [6] ISO/IEC 14443-4  | "Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol"                                      |
|      | [7] ISO/IEC 15693    | "Identification cards – Contactless integrated circuit(s) cards – Vicinity cards"  |
|      | [8] ISO/IEC 18092    | "Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)" |
|      | [9] ISO/IEC 10373    | "Identification cards -Test methods - Proximity cards- Optical memory cards- Vicinity cards"   |
| ETSI | [10] ETSI TS 102 127 | "Smart Cards; Transport protocol for CAT applications; Stage 2"  |
|      | [11] ETSI TS 102 221 | "Smart Cards; UICC-Terminal interface; Physical and logical characteristics"   |
|      | [12] ETSI TS 102 124 | "Smart Cards; Transport Protocol for UICC based Applications; Stage 1"   |
|      | [13] ETSI TS 102 223 | "Smart Cards; Card Application Toolkit (CAT)"  |
|      | [14] ETSI TS 102 224 | "Smart Cards; Security mechanisms for UICC based Applications – Functional requirements".  |
|      | [15] ETSI TS 102 225 | "Smart Cards; Secured packet structure for UICC based applications" (OTA)  |
|      | [16] ETSI TS 102 226 | "Smart Cards; Remote APDU structure for UICC based applications" (OTA)   |
|      | [17] ETSI TS 102 241 | "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card™"  |



|      |                          |   |
|------|--------------------------|---|
| [18] | <b>ETSI TS 102 312</b>   | Near Field Communication Interface and Protocol-2 (NFCIP-2) "   |
| [19] | <b>ETSI TS 102 483</b>   | "Smart Cards; UICC-Terminal interface; Internet Protocol connectivity between the UICC and terminal"                      |
| [20] | <b>ETSI TS 102 588</b>   | "Smart Cards; Application invocation Application Programming Interface (API) by a UICC webserver for Java Card™ platform" |
| [21] | <b>ETSI TS 102 600</b>   | "Smart Cards; UICC-Terminal interface; Characteristics of the USB interface"  |
| [22] | <b>ETSI TS 102 613</b>   | "Smart Cards; UICC – Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics"         |
| [23] | <b>ETSI TS 102 622</b>   | "Smart Cards; UICC – Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)"                              |
| [24] | <b>ETSI TS 102 694-1</b> | "Smart Cards; Test specification for the Single Wire Protocol (SWP) interface; Part 1: Terminal features"                 |
| [25] | <b>ETSI TS 102 694-2</b> | "Smart Cards; Test specification for the Single Wire Protocol (SWP) interface; Part 2: UICC features"                     |
| [26] | <b>ETSI TS 102 695-1</b> | "Smart Cards; Test specification for the Host Controller Interface (HCI) Part 1: Terminal features "                      |
| [27] | <b>ETSI TS 102 695-2</b> | "Smart Cards; Test specification for the Host Controller Interface (HCI) Part 2: UICC features "                          |
| [28] | <b>ETSI TS 102 695-3</b> | "Smart Cards; Test specification for the Host Controller Interface (HCI) Part 3: Host Controller features "               |
| [29] | <b>ETSI TS 102 431</b>   | "Smart Cards; Test specification for the Transport Protocol of CAT Applications (CAT_TP) validation"                      |
| [30] | <b>ETSI TS 102 835</b>   | "Smart Cards; Test Specification for SCWS Application Invocation API for Java Card™; Test Environment and Annexes"        |
| [31] | <b>ETSI TS 102 384</b>   | "Smart Cards; UICC-Terminal interface; Card Application Toolkit (CAT) conformance specification"                          |
| [32] | <b>ETSI TS 102 230</b>   | "Smart Cards; UICC-Terminal interface; Physical, electrical and logical test specification"                               |
| [33] | <b>ETSI TS 102 922</b>   | "Smart Cards; Test specification for the ETSI aspects of the IC USB interface; Part 1: Terminal features"                 |

|  |  |  |
|--|--|--|
|  | [34] ETSI TS 102 705   | "Smart Cards; Contactless API for Java Card(TM) for the UICC platform"   |
|  | [35] ETSI TS 103 115   | "Test specification for UICC API for Java Card™ for Contactless Applications; Test Environment and Annexes"                    |
| ECMA International                             | [36] ECMA-373  | Near Field Communication Wired Interface (NFC–WI)  |
| Java Service Requests (Java Community Process) | [37] JSR 000177  | "Security and Trust Services API for J2ME(TM) " (SATSA)  |
|  | [38] JSR 000257  | "Contactless Communication API"  |
| Java Card Specs                                | [39] Java Card 3.0.1   |  |
|  | [40] Java Card 2.2.2   |  |
| GlobalPlatform                                 | [41] GlobalPlatform 2.2.1, Core specification                    | (Including "Errata and precision list" Version 0.2.)   |
|  | [42] GlobalPlatform 2.2, Amendment A                             | "Confidential Card Content Management"   |
|  | [43] GlobalPlatform 2.2, Amendment B                             | "Remote Application Management over HTTP"  |
|  | [44] GlobalPlatform 2.2, Amendment C                             | "Contactless Services"   |
|  | [45] GlobalPlatform Secure Element Configuration                 | This document describes a specific implementation of the GlobalPlatform Card Specification for Secure Elements.                |
|  | [46] GlobalPlatform Secure Element Remote Application Management | GlobalPlatform Device Technology Secure Element Remote Application Management  |
|  | [47] GlobalPlatform Secure Element Access Control                | "GlobalPlatform Device Technology Secure Element Access Control"   |
|  | [48] GlobalPlatform UICC configuration                           | "UICC Configuration"   |
|  | [49] GlobalPlatform UICC Configuration Contactless Extension     | This document defines an extension of the GlobalPlatform UICC Configuration for UICCs equipped with contactless functionality. |
|  | [50] GlobalPlatform UICC Compliance Test Suite                   | "UICC Compliance test suite"   |
|  | [51] GlobalPlatform UICC Contactless Extension Test Suite        | "UICC Contactless Extension Test Suite"  |
|  | [52] GlobalPlatform Card Specification v2.1.1                    | (March 2003)   |
| 3GPP   | [53] 3GPP TS 31.115  | "Core Network & Terminals: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications"    |

|                            |   |  |
|----------------------------|---|--|
|                            | [54] <b>3GPP TS 31.116</b>                            | "Core Network & Terminals: Remote APDU Structure for (U)SIM Toolkit applications"  |
|                            | [55] <b>3GPP TS 23.040</b>                            | "Core Network and Terminals; Technical realization of the Short Message Service (SMS)"   |
|                            | [56] <b>3GPP TS 24.090</b>                            | "Group Core Network and Terminals; Unstructured Supplementary Service Data (USSD)"   |
| <b>OMA</b>                 | [57] <b>OMA TS Smartcard Web Server v1.1.1</b>        | Open Mobile Alliance: Smartcard-Web-Server   |
| <b>SIMalliance</b>         | [58] <b>SteppingStones_R7_v1.0.0</b>                  | Gives an overview based upon the ETSI Release 7 framework that references GlobalPlatform 2.1 of the Mobile Near Field Communication (Mobile NFC) technology. |
|                            | [59] <b>SteppingStones_SCWS_v.1.0.0</b>               | Analyzes and collects all information related to SCWS services and their remote management.  |
|                            | [60] <b>Open Mobile API Specification V2.03</b>       | The API specified in this document enables mobile applications to have access to different Secure Elements in a Mobile such as SIMs or embedded SEs.         |
|                            | [61] <b>Mobile NFC Stepping Stones V1.0.0</b>         | The present document is based upon the latest ETSI and GlobalPlatform specifications, main enablers for this kind of technology.                             |
| <b>SDA</b>                 | [62] <b>SD Specification Part 1</b>                   | SD Specifications; Part1; Physical Layer Specification   |
|                            | [63] <b>SD I/O</b>                                    | SD Specifications; Part E1; SDIO Specification   |
|                            | [64] <b>ASSD</b>                                      | SD Specifications Part1 A1 Advanced Security SD Extension Specification, Version 3.00  |
| <b>IETF</b>                | [65] <b>RFC 2616</b>                                  | Hypertext Transfer Protocol -- HTTP/1.1  |
|                            | [66] <b>RFC 4279</b>                                  | Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)   |
|                            | [67] <b>RFC 2246</b>                                  | The TLS protocol   |
| <b>EMVco</b>               | [68] <b>EMVCo Contactless Mobile Payment</b>          | EMVCo Contactless Mobile Payment - Type Approval Administrative Process, Version 1.0, January 2012   |
|                            | [69] <b>EMV Security Guidelines</b>                   | EMV Security Guidelines, Version 4.0 Release, December 2010  |
|                            | [70] <b>EMV Application Activation User Interface</b> | EMVCo Contactless Mobile Payment - Application Activation User Interface, Version 1.0, December 2010   |
| <b>Smart card Alliance</b> | [71] <b>Security_of_Proximity_Mobile_Payments</b>     |  |
|                            | [72] <b>NFC_and_Transit_WP_20120201</b>               |  |
| <b>GSMA</b>                | [73] <b>Mobile-Wallet-White-Paper</b>                 |  |

### 1.3 Abbreviations

|         |   |
|---------|---|
| AID     | Application Identifier                      |
| APDU    | Application Protocol Data Unit              |
| API     | Application Programming Interface           |
| APSD    | Application Provider Security Domain        |
| BIP     | Bearer Independent Protocol                 |
| CAT     | Card Application Toolkit                    |
| CAT-TP  | Card Application Toolkit Transport Protocol |
| CCM     | Card Content Management                     |
| CLF     | Contactless Front End                       |
| CRC     | Cyclic Redundancy Check                     |
| CRS     | Contactless Registry Service                |
| DAP     | Data Authentication Pattern                 |
| eSE     | Embedded Secure Element                     |
| GP      | GlobalPlatform                              |
| HCI     | Host Controller Interface                   |
| HTTP    | HyperText Transfer Protocol                 |
| IP      | Internet Protocol                           |
| ISD     | Issuer Security Domain                      |
| JavaME  | Java 2 Mobile Edition                       |
| JSR     | Java Specification Request                  |
| microSD | Micro Secure Digital Card                   |
| MNO     | Mobile Network Operator                     |
| NFC     | Near Field Communication                    |
| OTA     | Over The Air                                |
| PPSE    | Proximity Payment System Environment        |
| PSK-TLS | PreShared Key Transport Layer Security      |
| RAM     | Remote Application Management               |
| RF      | Radio Frequency                             |
| RFM     | Remote File Management                      |
| SCWS    | Smart Card Web Server                       |
| SD      | Security Domain                             |
| SE      | Secure Element                              |
| SE ID   | Secure Element Identifier                   |
| SIM     | Subscriber Identity Module                  |
| SP      | Service Provider                            |
| SWP     | Single Wire Protocol                        |
| TCP     | Transmission Control Protocol               |
| TLS     | Transport Layer Security                    |
| TSM     | Trusted Service Manager                     |
| UICC    | Universal Integrated Circuit Card           |
| USSD    | Unstructured Supplementary Service Data     |

## 2. Device and service deployment

In the past, the mobile device and its SE which, historically, has been the UICC, were focused mainly on telecommunication services, such as sending and receiving voice calls and SMS messages. The service provider (the mobile network operator (MNO)) has always been the owner of the SE and could even provide the mobile device with its own personalization. The MNO has always been the only administrator of the SE and the sole provider of the applications installed on it. The deployment of devices and applications was, therefore, controlled uniquely by this service provider.

Today, however, following the arrival of smart devices with NFC capabilities, there is a proliferation of downloadable applications, each of which requires a high level of interoperability and targets all different of devices and SEs. Several new models for deployment of devices and services are now emerging. The administration of such devices and their associated SEs is now complex and is heavily dependent on the business model and type of SE used.

### 2.1 'Issuer-centric' model and 'consumer-centric' model

In the 'issuer-centric' model, the service provider (issuer), typically an MNO, bank or transport company, controls the level of end user device security as well as the type of applications the device will support. This service provider pre-defines the list of applications that could be available on the SE. This is a 'vertical' model.

In the 'consumer-centric' model, the consumer selects applications among thousands of others, in one, or several, application portfolios. This is an 'open' model. In this model, the user chooses and purchases the SE, or the user's device carries an embedded SE (eSE).

This model increases the need for security especially when the application chosen is a security-sensitive service, such as payment transaction. The service provider and the consumer shall be confident on the required level of security on the device, and shall be confident that the device is configurable in such a way that the application and secure data are not exposed. Both the service provider and the consumer shall also be confident that other applications are unable to influence or impede the required level of security of the system.

Between these two previous extreme models, other intermediate models also exist. These include the 'renting' model where the issuer rents both space on, and support resources of, the SE to host other applications coming from other actors, as banking companies or transport companies. In such models some business agreements will bind the issuer and the hosted application provider together.

The device and service deployment may change drastically depending on the model used.

In the 'issuer-centric' model, the SE is provided by the issuer (e.g. MNO, bank or transport company) and controlled by this issuer during the whole life of the SE. The issuer domain and associated credentials, together with other security domains for some applications pre-installed on the SE at the issuance or installed in post issuance, are maintain and controlled by this issuer.

In the 'consumer-centric' model, the SE could be provided by an issuer (e.g. service provider) and 'opened' to other applications chosen by the consumer. Alternatively, the SE could be bootstrapped with issuer security domain controlled by a trusted service manager (TSM). This could be called a 'grey' SE.

## 2.2 Actors and roles

The following roles are identified in the mobile NFC ecosystem:

|   |   |
|---|---|
| <b>Chip manufacturer</b>                      | This entity is the semiconductor manufacturer of the electronic chip which contains an embedded microprocessor that will be used as the SE. This chip provided by the chip manufacturer shall integrate some dedicated hardware features in order to be compliant with the security requirements of an SE (e.g tamper resistance, cryptographic functions, non-volatile memory, etc).   |
| <b>SE manufacturer</b>                        | This entity will add on to the chip provided by the chip manufacturer the specific operating system and security functions required to transform it into an SE (a platform able to host secure applications). This entity is also in charge of the personalization of the SE on request of the issuer to create the appropriate primary security domain, called Issuer Security Domain  |
| <b>Handset manufacturer</b>                   | This entity manufactures the handset device that will host the SE.<br>When the SE is an eSE, the handset manufacturer shall solder the SE on the printed circuit board.<br>When the SE is a UICC or a smart microSD, the handset manufacturer shall provide a physical interface compliant with the removable SEs.  |
| <b>Secure Element issuer (vertical model)</b> | This entity is the provider of the SE and, in the case of the eSE, the provider of the associated handset. This entity is the owner of the Issuer Security Domain and will exchange with the SE manufacturer appropriate credentials for this primary security domain.<br>This role holds the ultimate responsibility for the SE. The SE Issuer has the responsibility to develop the card product profile, to choose the platform and application technologies to design card layout. It provides information about a SE and the controlling authority of the SE.<br>The SE issuer usually holds a particular security domain in the SE: the Issuer Security Domain (ISD). Card Content Management (CCM) operations that can be performed on this ISD are associated to the Security Domain Manager role that a SE provider may also play. |
| <b>Trusted Token Provider (open model)</b>    | This entity is the supplier of the SE in a consumer centric model which serves as the entity behind security assurance. This role is similar to the SE issuer in the vertical model. It ensures that the security level of the SE supplied is compliant to the security level claimed for this SE. The trusted token provider (TTP) is the guarantor of this security level. The SE manufacturer may or may not serve as the TTP.<br>The TTP holds a particular security domain in the SE similar to the Issuer Security Domain (ISD), and creates and provides to each domain administrator (see role described hereafter) that needs to communicate with the SE, the appropriate security domain.   |

|                              |  |
|------------------------------|--|
| <b>Domain administrator</b>  | <p>This entity is a trusted entity in charge of the management of the SE and handset for a collection of applications bound to it (Administration Domain). It will be in charge of the uploading of secure applications in the handset and SE. The SE could receive applications coming from different domain administrators bound to different application portfolios. At SE level, it is responsible for a set of security domains in the SE. Depending on the privileges associated to its security domains, the domain administrator may have the capability to directly load, install, extradite or personalize applications on behalf of an application provider.</p> <p>When it does not have sufficient privileges, or it does not have an OTA capability, it may request the help of another domain administrator to perform individual card content management operations or OTA dialog. The domain administrator role is typically the role of security domain manager, as defined in GP specifications (GlobalPlatform System - Messaging Specification for Management of Mobile-NFC Services - Version 1.0) .</p> |
| <b>Application provider</b>  | <p>This entity is a provider of applications that could be part of an application portfolio.</p> <p>This role holds the responsibility for the global (SE and Device) mobile-NFC service management towards end users, but may delegate the service management operational tasks to the Domain Administrator</p>   |
| <b>End user</b>              | A consumer that owns the device and the associated SE. This user consumes or run applications installed on the device and SE.  |
| <b>Controlling authority</b> | The optional entity managing exchange with an optional third party entity when is required by SE issuer and service provider. It is a trusted third party for both the SE issuer and for the service provider and can be used to personalize security domains in a confidential way.   |

**Table: Actors and Roles in mobile NFC ecosystem**

An ‘actor’ is a physical entity (e.g. a bank, an MNO or a handset manufacturer) that will assume one or more roles (i.e. responsibilities) depending on its business context.

Actors for the SE issuer role:

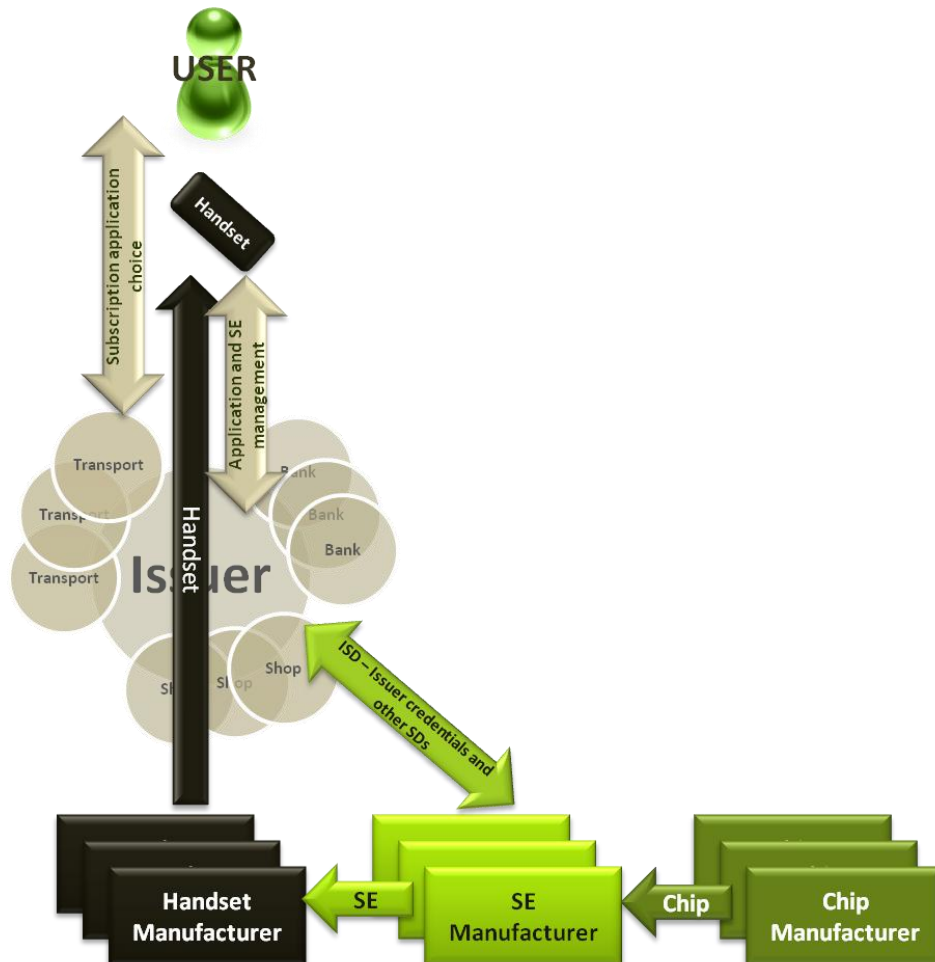
- In the case of a UICC, the SE issuer is the MNO.
- In the case of Smart microSD and in a vertical model, the SE issuer may be a service provider, a bank or a transport company.
- In the case of Smart microSD as a “grey” SE provided in shops, the Trusted Token Provider role could be adopted by the SE manufacturer itself.
- In the case of eSE, the issuer is typically the over-the-top player (OTT) (e.g. Google, Apple...)

The trusted service manager (TSM): The TSM is an optional actor in the system. This is a third party that implements one or more service management roles. TSM acts as an interface between a service provider and a MNO in a UICC-based SE, or between a service provider and handset manufacturer in an eSE, and between a service provider and another service provider in the case of Smart microSD. The TSM permits the simplification of the business deployment, using as an intermediary entity between several SE issuers and domain administrators. The business agreements are then simplified using this trusted third party.



## 2.3 Architecture

### 2.3.1 'Issuer-centric', vertical model



**Figure 1: Architecture – deployment – 'Issuer-centric'**

In the figure above, the SE manufacturer and even in some cases the handset manufacturer (eSE, or handset customized for a MNO) know the issuer who is also the service provider at the issuance of the product.

The SE manufacturer is then able to personalize the SE for the issuer with security domains dedicated to different applications, which can be pre-installed during personalization phase.

This model is the one currently deployed for UICC in mobile network communication. In these cases, the MNO is the issuer.

This model could be extended for some service providers, such as a bank or transport company, for example, by providing a SE as a smart microSD which could be connected to a mobile device.



### 2.3.2 'Consumer-centric', open model

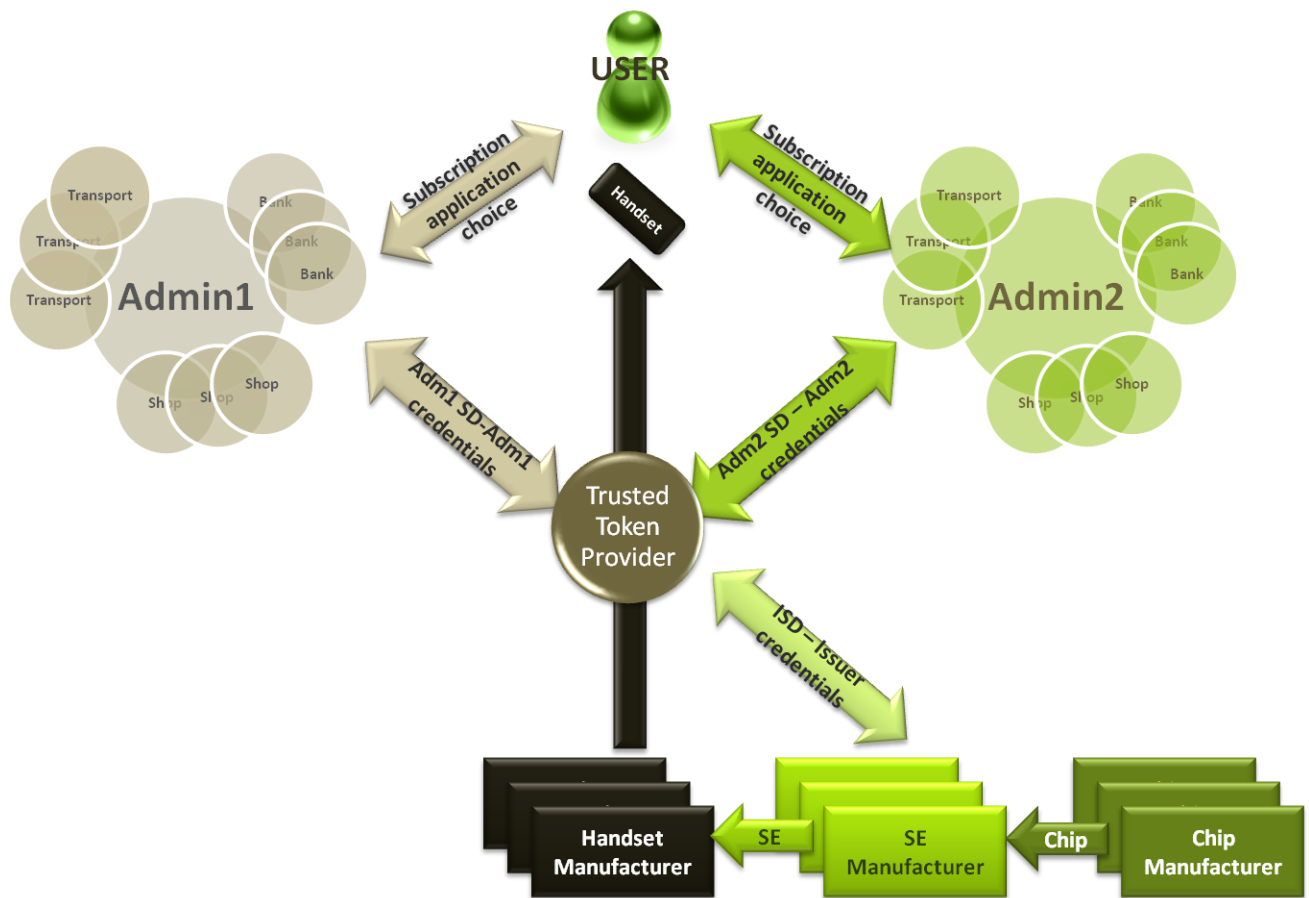


Figure 2: Architecture – deployment – 'Consumer-centric'

In the above scheme, the chip manufacturer, the SE manufacturer and handset manufacturer manufacture generic products without anticipating how they will ultimately be deployed. An 'administration domain' in the following document stands for a group of entities bound together with an agreement, each using the same infrastructure to operate their services (admin). An administration domain can be attached to an area (e.g. a country), a handset OS deployment, or a group of interests (several participating banks and/or transport companies, for example). Nevertheless, it could also be a single actor with its own infrastructure. This 'administration domain' could be seen as an applications portfolio with its administration server.

The infrastructure of the 'administration domain' provides the secure environment to install and run applications. The secure environment involves the SE and a remote server. The secure link between these entities is based on cryptographic materials shared between them.

This scheme is based on a secret in the SE, integrated at the issuance phase by the SE manufacturer, and controlled by the trusted token provider (ISD credentials). This secret is used to build the whole chain of trust that will be required to authorize the download the applications from different service providers.

In the above scheme the trusted token provider could include the chip manufacturer/SE manufacturer/handset manufacturer and represents one entity.

### 2.3.3 'Renting', intermediate model

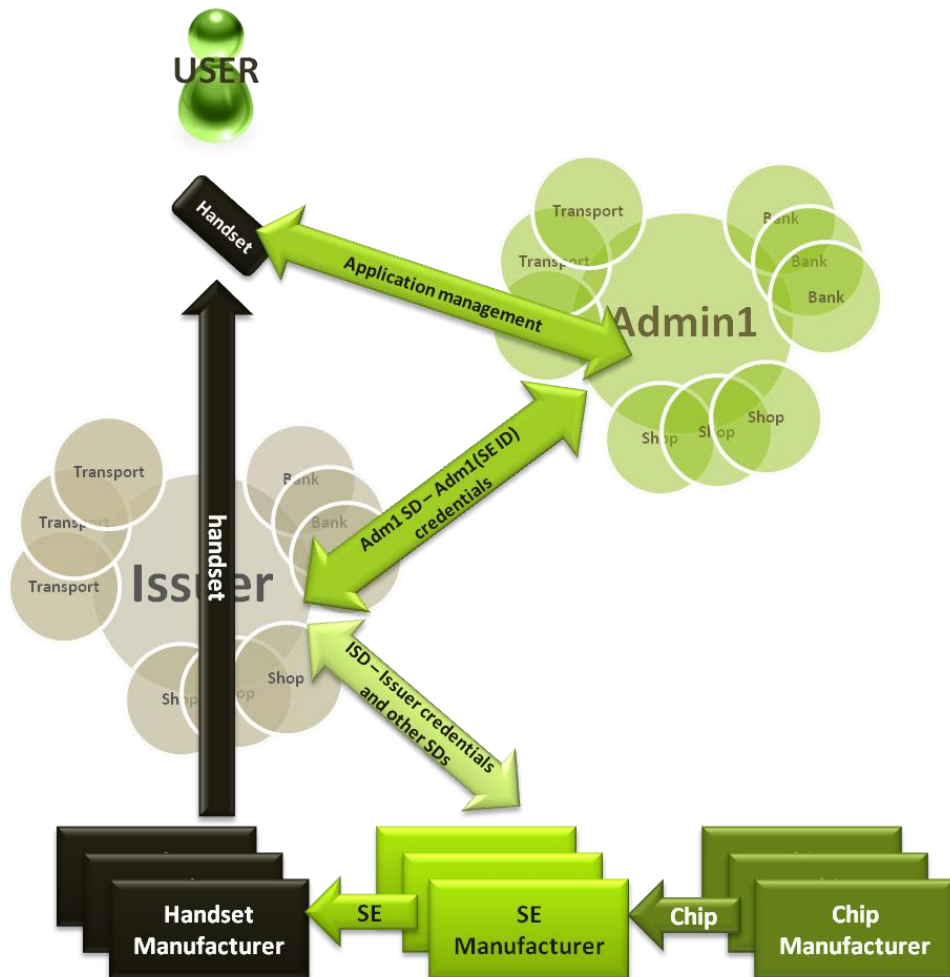


Figure 3: Architecture – deployment – 'Renting' model

In this model, the issuer rents some part and resources of the SE to host other applications coming from other actors, such as banks or transport companies. In such models some business agreements bind the issuer and the hosted application provider.

Depending on the privileges, the SE issuer's practice of renting to the other actors as banking or transport companies the architecture of the renting model is similar to the 'issuer-centric' model or 'consumer centric' model.

The bank may request that the SE issuer downloads and installs its application in the SE and handset. The SE issuer may give an ad-hoc authorization for this and provide the associated commands, or delegate the card management to the company creating the security domain for this application provider. In the first case the architecture is similar to the 'issuer-centric' model; in the other case the architecture is similar to 'consumer-centric' model except that the end user is not the initiator of this process.

The following sections describe the process for the deployment of the device with the personalization of this bootstrap environment and the deployment of applications with the establishment of the secure link between the remote admin server and the SE.

## 2.4 Use-cases

### 2.4.1 NFC banking service in UICC: an example of 'issuer-centric' model

Alice has a subscription attached to her favorite MNO 'myOp'. She has acquired with this subscription an NFC-enabled smart phone within the myOp catalog.

Alice has a bank account with 'myBank'. myBank and myOp have a commercial agreement, so Alice is offered a service that enables her to control her bank account with her smart phone and execute payments via NFC.

At the request of myBank, myOp creates a security domain for the banking application with appropriate credentials for this secure application and installs the banking application on the UICC. As a result, Alice can now execute payments via her smart phone's NFC functionality and use the device to manage her banking.

In this scenario the customer can directly perform payment transactions using the UICC card and their NFC capable handset. To achieve this goal, the UICC card must host one or more applications from one or more banking service suppliers, like MasterCard, Visa or others.

These applications must be provided and deployed in a secure way, even in third party networks. The GlobalPlatform 2.2, Amendment A defines a mechanism for confidential and secure loading, installation and personalization of these applications. According to this, application distribution can be done by the MNO, for example an Over-The-Air platform operator.

This scenario uses the 'issuer-centric' model as the MNO performs all the card management tasks.

When the MNO rents part of the UICC and delegates the card management and application management to the bank, then the model used is the 'renting' model.

The figure here after shows this ecosystem with the optional TSM.

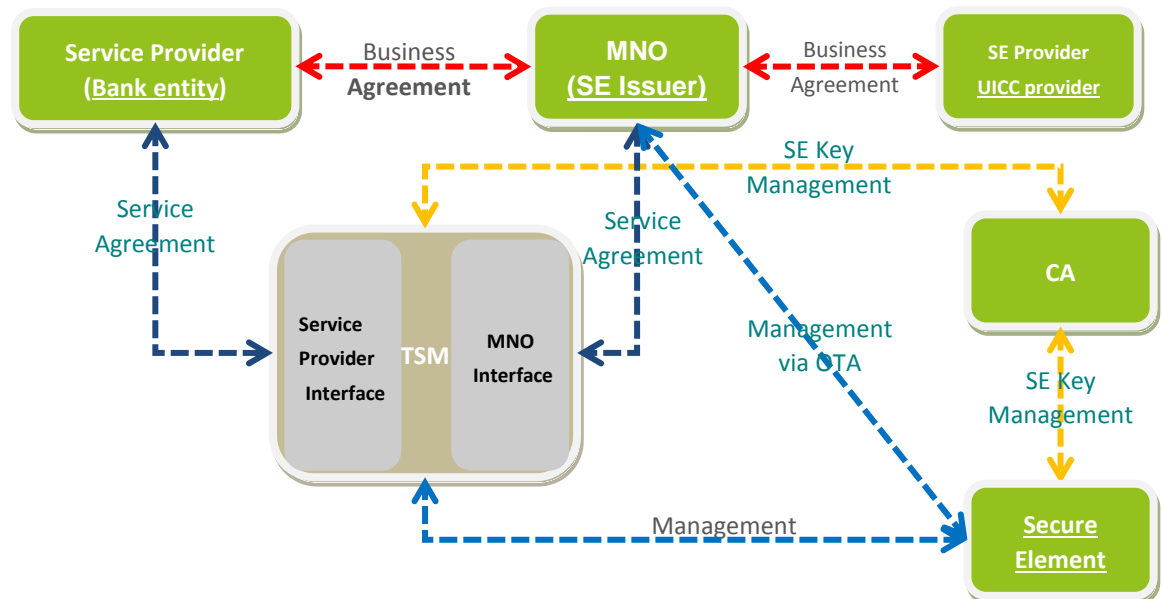


Figure 4: NFC banking service in UICC: an example of the ‘issuer-centric’ model

#### 2.4.2 NFC banking service in smart micro-SD: an example of ‘issuer-centric’ model.

Alice has a bank account in myBank. myBank offers Alice a service that enables her to control her bank account with her smart phone and execute payments via NFC. myBank delivers this service via a smart micro-SD containing the appropriate credentials and applications. Alice subscribes to this feature and receives a smart micro-SD that she can install in her NFC-enabled smart phone.

The application in the smart micro-SD enables Alice to pay for her purchases and manage her myBank account with her NFC-enabled smart phone. myBank regularly updates the application within the smart micro-SD together with the corresponding part in the handset using its card management capability.

In this scenario, the customer can directly perform payment transactions using the smart micro-SD card and his NFC capable handset. To achieve this goal the Smart micro-SD card must host one or more applications, from one or more banking service suppliers, like MasterCard or Visa.

These applications must be provided and deployed in a secure way, even in third party networks. The GlobalPlatform 2.2, Amendment A defines a mechanism for confidential and secure loading, installation and personalization of these applications. According to this, application distribution can be done by the service provider, for example using an Over-The-Air platform.

The banking service provider must deploy the applications for the micro-SD card, and must ship the micro-SD card to the final customer.

The micro-SD card shall be certificated by the payment scheme to be allowed to support loading of mobile payment applications.

In order to participate, the customer must have one or more accounts at the bank, and a smart phone with a micro-SD card slot. If the bank opts not to share space on the micro-SD with others service providers, then the user may have several smart micro-SDs from different banks. This scenario uses the 'issuer-centric' model.

The 'renting model' may also be used, if the bank opts to open its micro-SD to third parties, such as other banks or transportation companies for example, enabling them to access and utilize the SE for their own applications.

The figure here after shows this ecosystem with the optional TSM.

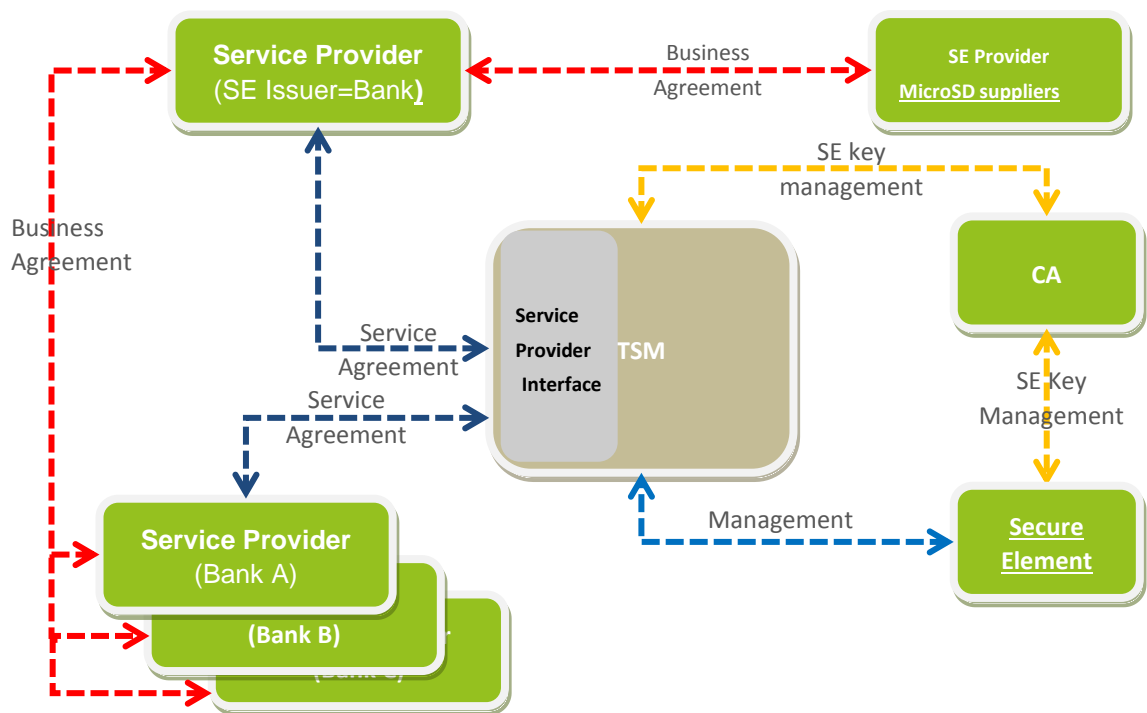


Figure 5: NFC banking services delivered via smart micro-SD: an example of an 'issuer-centric' model.

#### 2.4.3 Transportation service and loyalty in Embedded SE: an example of 'consumer-centric' model.

Alice has purchased a smart phone containing an embedded SE. For this device, an Over-The-Top player called "myOTP" offers a collection of applications for download via its website.

Alice commutes to work every day using public transportation. She needs a secure, easy and fast way to pay for her daily tickets, fees and authorisations.

Alice connects to the internet and downloads the appropriate transportation service application (handset and eSE part) corresponding to her smart phone from the myOTP website. She also subscribes to the transportation provider's NFC service.

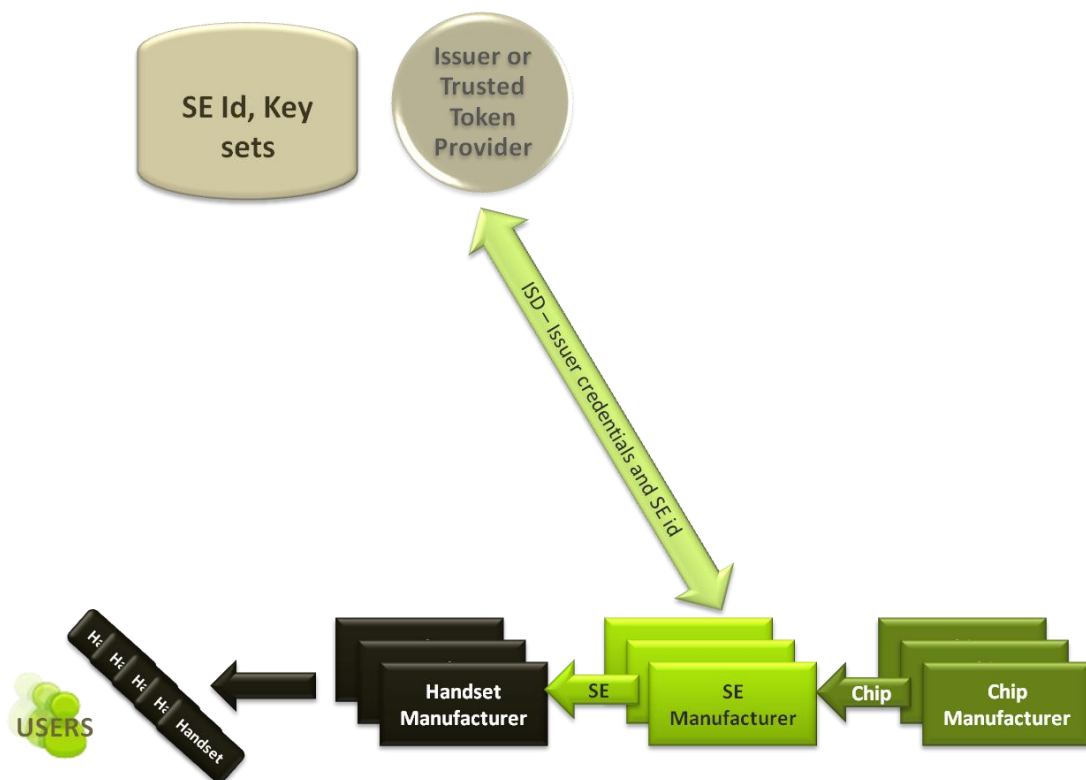
Alice is now able to use the transportation service with her smart phone. She is able to use her subscription for the daily travel from home to work. She is also able to book via the website advance tickets for the forthcoming weekend that will enable her to travel to

her family home and to meet friends. The tickets are stored in the eSE in her smart phone, and may be activated when needed.

Alice regularly shops for groceries in her local supermarket. This high-distribution company offers a loyalty scheme for smart phone users. Alice subscribes to this service online in order to gain some discounts on products she regularly consumes. She downloads the NFC loyalty application to her smart phone and the eSE. Now, each time she goes to the supermarket, she taps her smart phone onto the NFC reader available at the point of sale and is rewarded with loyalty coupons that she can redeem against future purchases at the supermarket.

For this use case the 'customer-centric' model is used, and the customer chooses the applications among those available in the appropriate administration domain. The trusted token provider and domain administrator exchanges are not displayed to the user, who therefore is given the impression that the exchange is made only between themselves and the merchant.

## 2.5 Device deployment



**Figure 6: Device – deployment**

The chip manufacturer provides the chips to the SE manufacturer including a hardware platform specifically designed to provide the highest levels of security.

The SE manufacturer installs its own specific secure operating system on the secure chip.

During this pre-issuance phase the SE manufacturer personalizes each chip, identifying it by its SE ID with a root key in a secure environment. This is known as the personalization stage. After the personalization process has been completed, the SE is market ready.

The root key is used to create a root security domain called the 'Issuer Security Domain'. This is done in collaboration with the SE provider (which can also be called the 'Trusted Token Provider').

At this stage, there is an exchange between the Trusted Token Provider and the SE manufacturer, in order to share the list of SE IDs, together with their corresponding key sets.

The Trusted Token Provider retrieves the necessary information from the SE, allowing it to connect to the SE and establish the parameters for its life cycle management (e.g. SE characteristics, OS version memory size). This enables further administration of the SE either by the issuer itself (in case of 'vertical model' or 'issuer-centric' model) or by an external administration server (in the case of an open model or consumer centric model).

The SE is then soldered either onto the device (handset) in the case of an eSE, or onto a smart microSD, or onto the UICC.

In the case of the 'vertical model', the issuer may create at the personalization stage other security domains for pre-issuance applications that are pre-installed on the SE. The corresponding credentials are stored securely and mapped onto the SE ID.

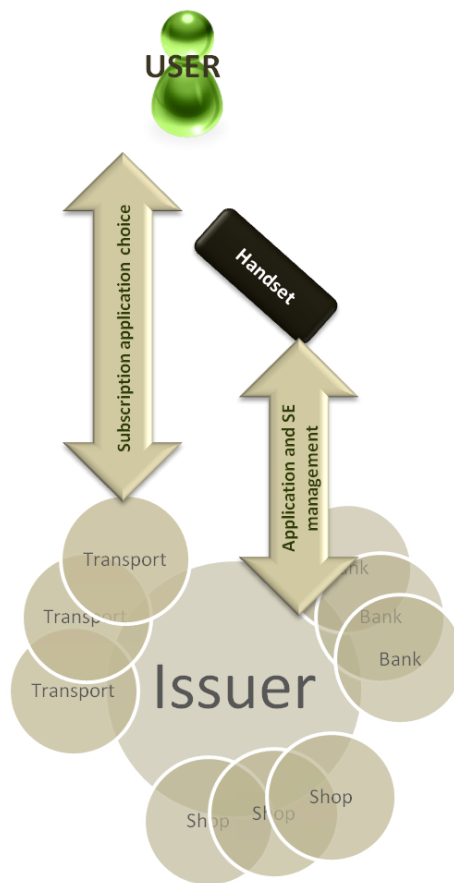
In case of smart microSD, there could be two use cases: 'vertical model' and 'open model'.

In the case of the 'vertical model', the issuer is the service provider (such as the bank or transport company) and retains control over the SE, creating different security domains for its own suite of applications.

In the case of an 'open model', the smart microSD is personalized with the issuer security domain controlled by the trusted token provider. The SE ID and the address of the Trusted Token Provider control the SE. In this document, SIMalliance refers to this type of SE as a 'grey SE'. These grey SEs may be further deployed through a high distribution market.

## 2.6 Service Deployment

### 2.6.1 Vertical model - 'issuer-centric'



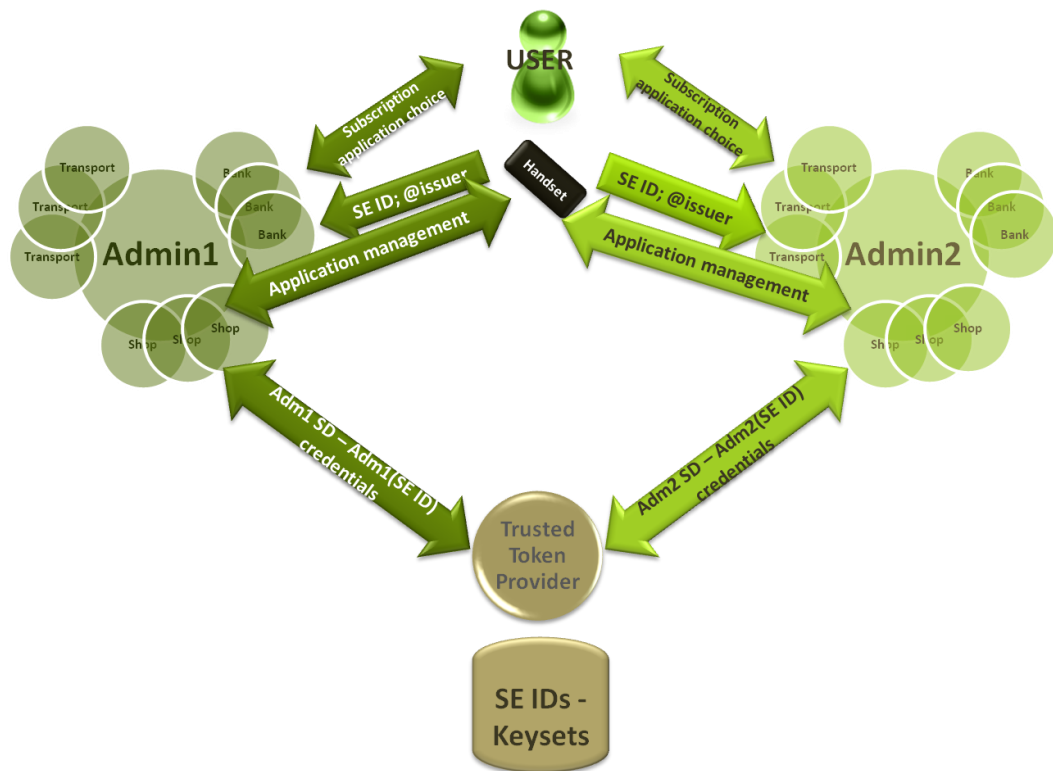
**Figure 7: Service – deployment – 'issuer-centric'**

In this model, the consumer chooses a service among the set of services proposed by the issuer and contacts the issuer in order to obtain the download of necessary software and activation functions for the service. The issuer, using its card management system, downloads and activates the service. The issuer may also perform some download, upgrade or activation of services independently. In this case, these actions are invisible to the consumer. Here, the issuer and the provider of application (such as a bank, or transport company) collaborate to form a commercial agreement.

This model of service is currently used by the MNO in UICC service deployment.



### 2.6.2 Open model - 'consumer-centric'



**Figure 8: Service – deployment – 'consumer-centric'**

When a consumer wants to use an NFC payment solution for the first time, they access a remote administration server (an application portfolio, for instance) in order to download the corresponding application.

Secured NFC applications are made up of two software components, each relative to either the handset or the eSE.

To be able to establish a secure channel and install the eSE software component of the application, the remote server requires cryptographic materials that it does not yet have.

When discovering a new SE, the administration server retrieves the unique SE identifier (See section 2.7 - 'Unique SE identifier', below) and the address of the issuer server. Other information as the type of SE may be retrieved as well in the recognition data as defined by GlobalPlatform in its configuration specifications (UICC configuration and Secure Element configuration).

Through a secure channel established between the administration server and the trusted token provider server, the administration server requests the cryptographic materials corresponding to the unique SE identifier. Only then is it able to perform the application download and configuration.

Through the secure channel, the trusted token provider server may also provide to the administration server the GlobalPlatform compliant APDU commands that need to be sent to the SE in order to create the corresponding security domain.

The NFC application part in the handset communicates with the administration server over HTTP. The application design in the handset distinguishes the admin agent in charge of communication with the SE and the actual business part in charge of the graphical user interface (GUI) and transaction management.

The protocol between the handset and administration server is relatively simple and can be performed with HTTP commands, since the only requirement in the scope of key retrieval is to be able to carry the SE unique identifier and the address of the trusted token provider server. Nevertheless, for application download and secure domain management, the interface between the handset and administration server should be compliant with the following GlobalPlatform specifications:

- Remote Application Management over HTTP; card specification v2.2, Amendment B.
- Messaging Specification for management of Mobile-NFC services.

## 2.7 Unique SE identifier

The unique SE identifier consists in a concatenation of the standardized Issuer Identification Number (IIN) and the Card Image Number/Card Identification Number (CIN). The globally unique SE ID is defined in GlobalPlatform specifications (GP Card Specification v2.2.1 [52] and GP SE config [45]).

## 2.8 Credentials description

|                               | 'Issuer-centric' Model   | 'Renting' model   |   | 'Consumer-centric' model  |
|-------------------------------|--|---|---|---|
|                               |  | 'Issuer-centric' like   | 'Consumer-centric' like   |   |
| <b>SE Issuer</b>              | SE issuer credentials define the Issuer Security Domain. Used for card (SE) management | SE issuer credentials define the Issuer Security Domain. Used for SE management to download application provider applications | SE issuer credentials define the Issuer Security Domain. Card management is left to domain admin server either in delegated mode or authorized mode |   |
| <b>Trusted Token Provider</b> |  |   |   | Credentials that defines the Issuer Security Domain 'like' (with more or less privileges than the ISD for 'issuer-centric' model) |

|                             |  |  |  |  |
|-----------------------------|--|--|--|--|
| <b>Domain Admin server</b>  | No admin server  | In this case the admin server has no capability for card management which is delegated to the SE issuer  | Card management either under the control of SE issuer (delegated mode) or totally autonomous (authorized mode) with associated memory quota (rented space) | Credentials that define the security domain of the domain administrator. Used for card management (in delegated mode or authorized mode) |
| <b>Application provider</b> | Applicative keys, proprietary keys managed by the application itself and generated in the Issuer Security Domain using the appropriate GP commands | Applicative keys, proprietary keys managed by the application itself and generated in the Issuer Security Domain using the appropriate GP commands | Applicative keys established with the SE using the security domain of the domain admin   | Applicative keys established with the SE using the security domain of the domain admin   |

Table 1: Credentials description

### 3. Secure Element general architecture

This chapter describes the different SEs that currently exist (UICC, smart microSD, embedded Secure Element), and focuses on the principle differences between them. For each type of SE, it describes a main architecture, which that involves several different levels. Basic information related to the new communication buses and protocols involved between the SE and the CLF are shown. Several interfaces between the SE and the mobile processor are described, together with a descriptive overview of the important specifications that relate to each type of SE.

An SE is a tamper resistant component which is used in a device to provide the security, confidentiality, and multiple application environments required to support various business models.

An SE may exist in a variety of form factors, including UICC, embedded SE and smart microSD. The SE should provide separate memory for each application without interactions between them. The SE resides in extremely secure chips and fall into two categories:

**Removable:** Smart Card (especially the UICC), smart microSD card.

**Non-removable:** Embedded Secure Element (eSE).

#### Main features:

**Portability:** if the handset is changed, the applications should be available on the user's new device. This should ensure continuity of service when user changes their handset. This is a challenge for the eSE.

**Security:** based on the secure functionality contained in the SE, such as Over-The-Air disablement, remotely shutting down the SE's capacity to perform any kind of operation.

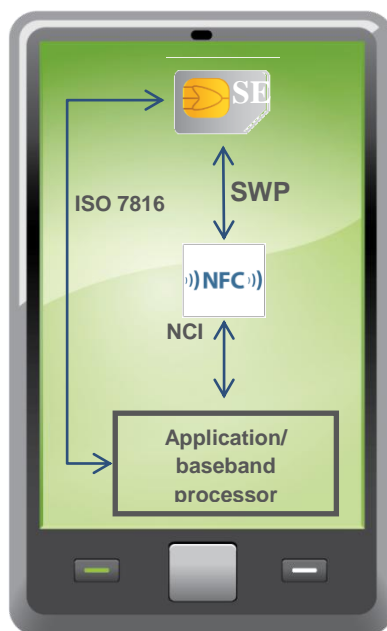
**Multi-application:** capable of hosting several applications, delivered by multiple service providers (banks, transport companies, etc.). Each service provider has access to its own domain in the SE. This ensures strict separation and isolation of the applications.

**Remote management:** enables the applications and data loading in the SE to be modified remotely.

#### 3.1 Different SE form factors

##### 3.1.1 UICC

The UICC is the physical smart card which contains the application authenticating the user in the mobile network. It contains several applications such as SIM, USIM, and others, and offers typically a Java Card-based operating system.



**Figure 9: UICC SE architecture in a smart phone**

Traditionally the UICC has been distributed by the MNO. As a result, the MNO has defined SE business models to date.

If a new application provider seeks to install its application on the SE, it should open a dialogue with the MNO to explore options for sharing access. Under these circumstances, the card operating system could have separate security domains associated with different application providers.

### Advantages

- UICC is able to use the SIM toolkit mechanism to communicate with the handset and receive information from the phone's keypad. Therefore, a user can communicate with the application in the SE without the requirement to add new software onto the phone itself.
- The standardized interface (SWP + HCI) between UICC and CLF enables the SE to support faster deployment for multiple models of handset. The UICC is based on well-established specifications from ETSI Smart Card Platform and the GlobalPlatform standards which help to eliminate problems relating to interoperability.
- The subscriber's identity and applications residing in the UICC are transferred when the user upgrades their device and remain available in UICC. The handset software components for some NFC service applications may need to be reinstalled.
- Security standards required by application providers (e.g. banks) are well established and represented in the UICC.

- The UICC can be segmented into a number of security independent domains to host several applications from a range of different issuers.
- A direct OTA channel is available between the remote admin server and UICC using the SMS, CAT-TP or RAM/RFM over HTTPs protocols.
- OTA provisioning is possible so that new applications can be downloaded remotely which can access the SE when the handset is lost or stolen, in order to disable the SE through existing (OTA) mechanisms.
- The UICC is always available and reachable through the ISO interface.
- The UICC can use the CAT (Card Application Toolkit) as defined in ETSI TS 102 223 [13] to interact with the device or with the applications in the device. For example an application in the UICC can directly display a message to the end user.

### Disadvantages

- Requires a phone with NFC and SWP capabilities.
- Requires a business agreement between the card issuer (MNO) and the other services providers (see for example the processes defined by AFSCM or GSMA).
- The service provider shall take into account, for the deployment and certification of the application, the life cycle of UICC and MNO process when issuing UICCs (e.g. new UICC released for Christmas period). As an example the UICC could remain active in the mobile for a period of 10 years, unlike current plastic plastic payment cards.

### 3.1.2 Embedded Secure Element (eSE)

This is a separate chipset in the handset. The SE is embedded in the mobile at the time of manufacturing. The eSE is connected to CLF via standard interfaces like SWP or via other interfaces like DCLB or NFC-WI. The eSE is distributed by the handset manufacturers.

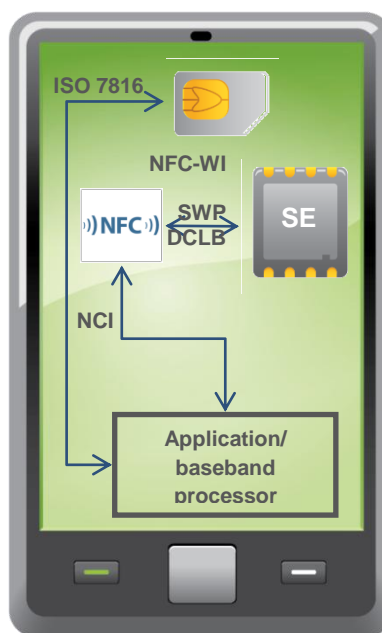


Figure 10 Embedded SE architecture in a smart phone

#### Advantages

- Can be available on mobile devices that do not handle UICC and at a lower cost than having to provide a smart microSD card.
- Does not require the same level of standardization as the removable type since the embedded SE cannot be removed from the handset which provides less interoperability problems on the Handset/eSE interface.

#### Disadvantages

- There is no portability option when the user changes their handset. Instead, the eSE requires applications to be removed from the legacy handset and re-deployed on the new handset.
- With each new device, the SE has to be certified, not only in order to re-test the applications, but also the SE itself.
- Applications need to be adapted to the connectivity protocol used between the SE and the handset.
- The certification and validation processes can be complex when proprietary interfaces are used as the SE is soldered on the device.
- Given that the SE is soldered onto the handset; the lifecycle of the SE is dependent on the lifecycle of the handset. There is no option to change the SE independently of the handset.
- A 'wipe' mechanism is required and must be utilised by the user when the device is lost, sold or stolen in order to delete all applications and data on the eSE.
- The management of a UICC SWP in a SWP enabled device integrating an eSE is a complex task which is inhibited by interoperability issues.

### 3.1.3 Smart microSD

Here, the SE is stored on a smart microSD card. The 'smart microSD' name has been defined by the SD association.

The added value of the smart microSD form factor is its potential to be utilised independently of MNOs and handset manufacturers. There are two types of smart microSD:

- Standalone NFC smart microSD, which has its own NFC controller and antenna.
- NFC smart microSD, which is connected to the NFC controller (CLF) (and then the antenna) of the device through the SWP link.

The SD Association is standardizing only the NFC smart microSD. The SD Association also defines the interface between the device and the smart microSD when sending APDUs. This interface is defined in the ASSD specifications [64].

#### Advantages

- Application providers can ship smart microSDs to their customers without involving MNOs or handset manufacturers. The applications residing on smart microSD can be transferred from the old phone to the new phone without having to re-deploy them.
- Smart microSD enables the application provider to be the owner of the SE.
- Service providers are able to print branding on the surface of the card.

#### Disadvantages

- Most handsets only carry a single smart microSD card slot. The end user most commonly uses the microSD slot to extend the memory of their device, enabling them to store personal digital valuables, such as photos. The practice of swapping from one microSD to another in order to enable NFC services could prove to be prohibitively inconvenient for the end user, especially as the microSD slot is not easily accessible in most handsets. An alternative could be for the service provider to include extra memory in the smart microSD, enabling the user to continue to store personal data. This would, however, increase the cost of the smart microSD.
- Since the SE issuer is an application provider, they have the choice to open their card to other application providers. An alternative here is for multiple smart microSDs to co-exist, supporting different applications from a range of application providers.
- No standard currently exists for an application in the microSD card to interact directly with the end user.
- Interoperability problems exist relative to the handset's access to the smart microSD, due to the fact that the ASSD specification is not yet widely deployed. Proprietary solutions using the memory storage commands defined in the SD protocol can be implemented by some smart microSD to bypass this issue.



### 3.1.4 NFC smart microSD

This section describes the case of NFC smart microSD.

The Smart microSD is distributed by the service provider. Therefore, the service provider can distribute NFC applications independently of MNOs. In this case the SE management also is handled by service provider.

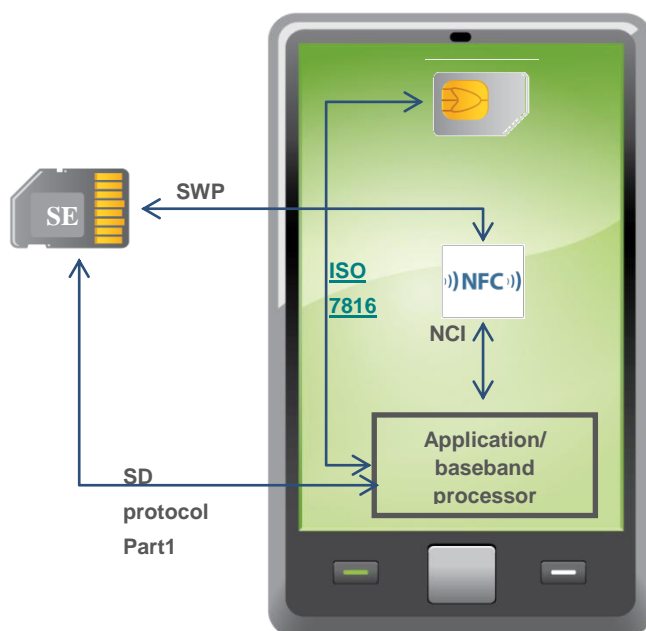


Figure 11: Smart microSD SE Architecture in smart phones

#### Advantages

- Efficiency of contactless interface: reusing the CLF and antenna of the device that has been tested and calibrated in various contexts means there are less interoperability issues linked to the integration with the device.

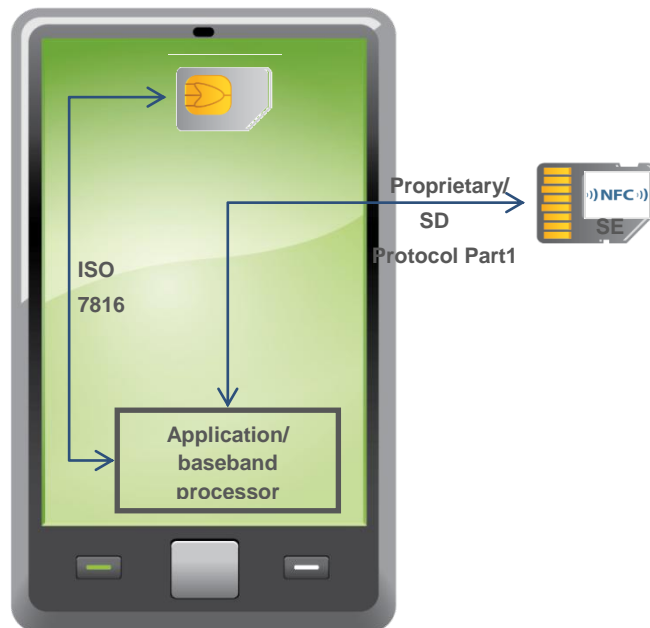
#### Disadvantages

- Requires a handset with NFC capability and that has implemented the SD interface and SWP link with CLF.

### 3.1.5 NFC standalone smart microSD

Standalone NFC smart microSD provides NFC services to devices that are not NFC enabled and have neither an NFC controller nor an

antenna.



**Figure 12: Standalone NFC smart microSD**

#### Advantages

- Standalone NFC smart microSD provides both the NFC chip and SE; it can be used in handsets without any NFC capabilities at all.

#### Disadvantages

- Depending on the physical implementation of the device, the antenna in the standalone NFC smart microSD may be shaded by some device components (e.g. battery, metallic back cover), which can interfere with the contactless signal and inhibit the communication.
- NFC operations may conflict and behave unpredictably when the standalone NFC smart microSD is inserted into a NFC-enabled device having a CLF and its own antenna. This is because potential technical barriers related to adequate testing and compatibility with the handset remain in this environment.

Table 2, below, shows the summarized features for the different types of Secure Element.

| Features<br>Types of SE                      | UICC | Smart<br>MicroSD | Smart Micro<br>SD+SWP | Smart MicroSD<br>+NFC | eSE |
|--|------|------------------|-----------------------|-----------------------|-----|
| Removable + Portability                      | ✓    | ✓                | ✓                     | ✓                     |     |
| Standardized Interface (HCI+SWP)             | ✓    | ✓                |                       |                       | ✓   |
| Can be deployed in a handset without SD slot | ✓    |                  |                       |                       | ✓   |
| MNO Control                                  | ✓    |                  |                       |                       |     |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| Handset Manufacture Control                           |   |   |   |   | ✓ |
| Service Provider Control (Banks...)                   |   | ✓ | ✓ | ✓ |   |
| Share the SE between different Application Providers. | ✓ | ✓ | ✓ | ✓ | ✓ |
| Not requires extra interface into the Handset         | ✓ |   |   | ✓ | ✓ |
| Can be shipped in a handset without NFC capability    |   |   |   | ✓ |   |
| Can stamp the brand on the top of the SE              |   | ✓ | ✓ | ✓ |   |

Table 2: Brief features for different types of SE

### 3.2 Architecture for Embedded SE, smart microSD & UICC

In the table below, there are the several architectures for each of the SE form factors that currently exist. The aim of this table is to show which of the main structures are needed in order to develop a SE. Basic information from low protocols of the chip to high level applications are shown. In the following pages, more information regarding these protocols is provided.

| SEs Levels                             | SIM                           | Smart MicroSD               | eSE                         |
|--|-------------------------------|-----------------------------|-----------------------------|
| Wallet                                 | Handset application + CRS App |                             |                             |
| Multiple Access Application            | GP Amendment C                |                             |                             |
| Multiple Application Service Provider  | GP UICC Configuration         | GP SE Configuration         | GP SE Configuration         |
| OTA                                    | ETSI TS 102 225               | SE OTA (RAM over HTTPS)     | SE OTA (RAM over HTTPS)     |
|  | RAM_OTA (TS 102 226)          |                             |                             |
|  | SCP81 (RAM over HTTPS)        |                             |                             |
|  | SCP80 over SMS                |                             |                             |
|  | CAT_TP                        |                             |                             |
| SE Access control API & Access Control | JSR 177                       | JSR 177                     | JSR 177                     |
|  | JSR 257                       | JSR 257                     | JSR 257                     |
|  | GP SE Access Control          | GP SE Access Control        | GP SE Access Control        |
|  | Simalliance Open Mobile API   | Simalliance Open Mobile API | Simalliance Open Mobile API |
|  | Proprietary SE access API     | Proprietary SE access API   | Proprietary SE access API   |

|  |                          |                         |                    |
|--|--------------------------|-------------------------|--------------------|
| User Interface                             | Smart Card Web Server    | App*                    | App*               |
|  | SIM toolkit services App |                         |                    |
| Security Channel Protocol Service Provider | SCP02                    | SCP02                   | SCP02              |
|  | SCP03                    | SCP03                   | SCP03              |
| Operating System                           | JAVA CARD                |                         |                    |
| CLF to SE                                  | HCI                      | SWP                     | SWP + HCI          |
|  | SWP                      | HCI                     | ECMA-373 (NFC-WI)  |
|  |                          |                         | DCLB (Proprietary) |
| Chip                                       | ETSI TS 102 221          | ASSD                    | None               |
|  | ISO 7816                 | SD Specification Part 1 |                    |

Table 3: NFC SE Architecture

\*Lack of standard regarding the user interface for smart microSD and eSE

### 3.3 Specifications

This chapter describes an overview of important specifications related to each SE form factor.

The SE and handset shall be coherent and compliant to ETSI SCP Release 9 Specifications. The features introduced in Rel-9 and GP 2.2 offer significant enhancements in the security of remote service management, making the deployment of secure banking applications easier and more reliable. Transportation services can be enhanced using the features, such as 'reader mode' specified in Release 9.

For more details please consult the previous SIMalliance NFC Stepping Stones 2011 document, available here: <http://www.simalliance.org/en/resources/recommendations/>

| Levels \ SEs                                     | UICC  | TEST                                  |
|--|---|---------------------------------------|
| Multiple Access Application Management           | GP Amendment C                              | UICC Contactless Extension Test Suite |
| Multiple Application Service Provider Management | GP UICC Configuration                       | UICC Compliance Test Suite            |
|  | GP UICC Configuration Contactless Extension | UICC Contactless Extension Test Suite |
| Remote   | TS 102 225                                  | None                                  |

|  |                             |                   |
|--|-----------------------------|-------------------|
| Management                             | TS 102 226                  |                   |
|  | GP Amendment B              | None              |
|  | ETSI TS 102 124 (CAT_TP)    | ETSI TS 102 431   |
| SE Access control API & Access Control | SE Access Control           | None              |
|  | Simalliance Open Mobile API | None              |
| User Interface                         | ETSI TS 102 588 (SCWS)      | ETSI TS 102 835   |
|  | ETSI TS 102 223 (STK)       | ETSI TS 102 384   |
| Operating System                       | JAVA CARD 3.0               | Java Card API/TCK |
| CLF to SE                              | ETSI TS 102 613 (SWP)       | ETSI TS 102 694-2 |
|  | ETSI TS 102 622 (HCI)       | ETSI TS 102 695-2 |
|  | ETSI TS 102 705 (HCI API)   | ETSI TS 103 115   |
| SE/Handset Interface                   | ETSI TS 102 221             | ETSI TS 102 230   |
|  | ISO 7816                    | ISO/IEC 10373     |

Table 4: Specifications related to UICC

| SEs<br>Levels                                    | eSE                         | TEST                                  |
|--|-----------------------------|---------------------------------------|
| Management Multiple Access Application           | GP Amendment C              | UICC Contactless Extension Test Suite |
| Management Multiple Application Service Provider | GP SE Configuration         | none                                  |
| Remote Management                                | GP Amendment B              | none                                  |
|  | GP SE OTA                   | none                                  |
| User Interface                                   | None                        | None                                  |
| Operating System                                 | JAVA CARD 3.0               | Java Card API/TCK                     |
| CLF to SE  | ETSI TS 102 613 SWP         | ETSI TS 102 694-2                     |
|  | ETSI TS 102 622 (HCI)       | ETSI TS 102 695-2                     |
|  | ETSI TS 102 705 (HCI API)   | ETSI TS 103 115                       |
|  | ECMA-373 (NFC-WI)           | None                                  |
|  | DCLB (Infineon Proprietary) | None                                  |
| SE/Handset                                       | None                        | None                                  |

Table 5: Specifications related to eSE

| SEs<br>Levels   | Smart MicroSD           | TEST                                     |
|---|-------------------------|--|
| Management<br>Multiple Access<br>Application              | GPAmdendment C          | Contactless Test Suite<br>(under review) |
| Management<br>Multiple<br>Application<br>Service Provider | GP SE Configuration     | None                                     |
| Remote<br>Management                                      | GP SE OTA               | None                                     |
| User Interface  | None                    | None                                     |
| Operating System  | JAVA CARD 3.0           | Java Card API                            |
| CLF to SE   | ETSI TS 102 613 SWP     | TS 102 694-2                             |
| SE/Handset<br>Interface                                   | ISO 7816                | ISO/IEC 10373                            |
|   | ASSD                    | None                                     |
|   | SD Specification Part 1 | None                                     |

Table 6: Specifications related to Smart microSD

## 4. Abstracts on the possible SE communication buses

The following section describes basic information related to the new communication buses and protocols involved relative to the SE and the CLF. It also describes the interfaces between the SE and the mobile processor.

The protocols used between the SE and the CLF are:

- For UICC: SWP/HCI
- For smart microSD: SWP/HCI
- For eSE: SWP/HCI, I2C, SPI, NFC-WI, DCLB

As SWP can be used in all three SE form factors, its selection is recommended for optimal interoperability.

The upper layer HCI is used on top of SWP, but potentially could also be stacked on top of any other physical protocol.

When HCI is used, it allows standard interoperable layers to be leveraged for application development.

The protocol used between the SE and the mobile processor (application/baseband processor) may be different depending on the SE concerned:

- For UICC: ISO7816
- For microSD: SD protocol
- For eSE: a common solution here is for the CLF to channel communications between the SE and the mobile processor. The eSE is already required to support the adequate protocol for the connection with the CLF.

Whatever the physical connection and the adopted protocol, applications over the mobile processor have the opportunity of interacting with the SE applications by using APDUs.

Based on the traditional OSI model representation, below is an illustration of the layered stacking of protocols involved in contactless terminals to access the SE:

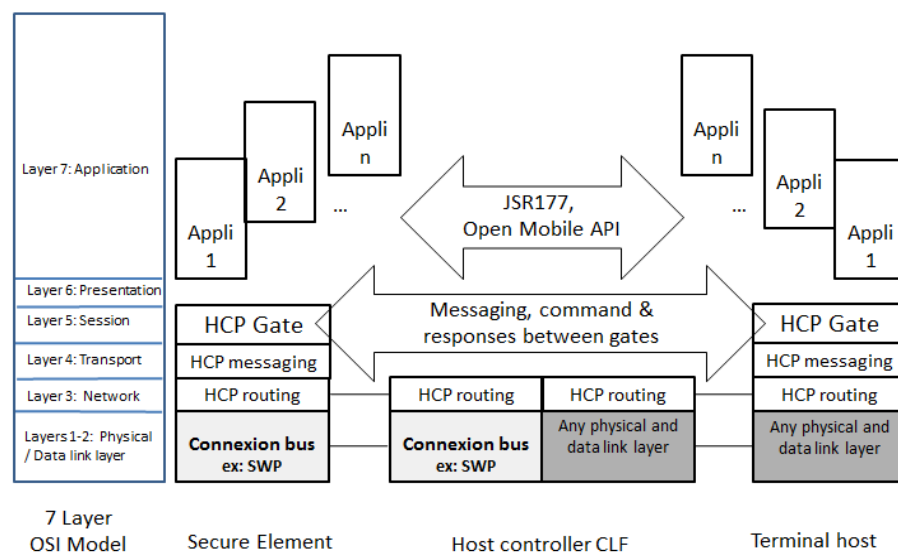


Figure 13 : Layered stacking of SE/CLF protocols

Below is an illustration of the different interactions between the key elements in a contactless handset.

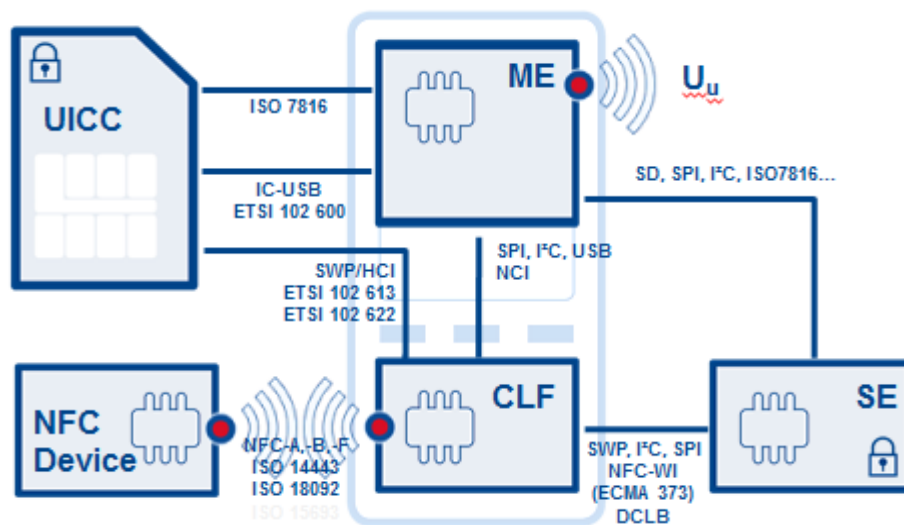


Figure 14: NFC SE in a handset

#### 4.1 SWP

SWP is the most adopted interface bus between a CLF and a SE and is a mandatory protocol to be supported in case of UICC and microSD.

As SWP can be used in all three possible SE form factors its choice is recommended for optimal interoperability.

Apart from specific pin allocation, which depends on the type of SE utilised, together with its form factor, SWP protocol is detailed in the NFC Stepping Stones 2011



document, see ref [61]. The document is also available for download here: <http://www.simalliance.org/en/resources/recommendations/>

## 4.2 The SPI bus

In case of the eSE, SPI (Serial Peripheral Interface) may be used between the CLF and the SE.

SPI bus is not used in case of microSD and UICC.

SPI is a general-purpose synchronous serial interface bus established by MOTOROLA in the mid 1980s and has been supported by various chip manufacturers ever since. SPI was designed to allow a microcontroller to communicate with peripheral devices such as EEPROMs.

The SPI bus operates in full duplex and relies on a master-slave relationship. The master initiates the data frames.

The SPI bus allows a multi-slave implementation as shown below with three slaves:

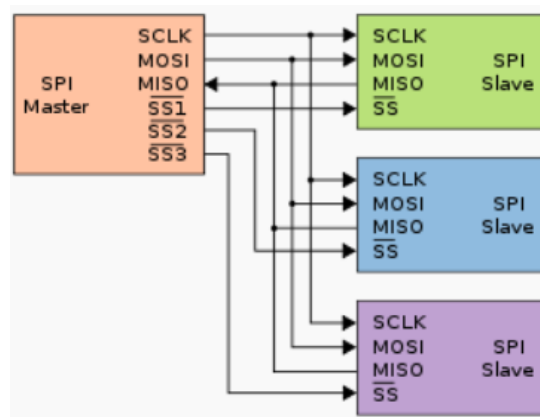


Figure 15: SPI bus: Single master, multiple slaves

The SPI bus consists of four signals:

- Clock (SCLK1)
- Master output, slave input (generated by Master (MOSI))
- Master input, slave output (generated by Slaver (MISO))
- Slave select (SS)

The above figure shows how these signals are wired in a multiple-slave configuration.

The master generates the clock and selects the slave with which it seeks to communicate.

**Communication protocol:** proprietary

**Data rates:** tens of Mbps

**Debugging tools:** dedicated SPI protocol analyzers or oscilloscope with integrated SPI analyzer.

### 4.3 The I2C bus

Like SPI, I2C (Inter-Integrated Circuit) may be used between the CLF and the SE. But I2C is not used in the cases of microSD or UICC.

I2C is a serial synchronous bus developed by Philips, also in the 1980s, with the same purpose as SPI: to interconnect processors. It was extensively used in television sets.

It is also called two-wire interface, because it only requires a data signal (SDA) and a clock signal (SCL), as shown below:

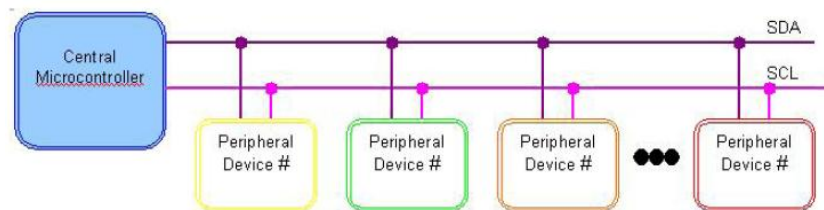


Figure 16: I2C - a two-wire interface

The master controller generates the clock (SCL) and sends the data (SDA), except for the acknowledgment, which is sent by the slave to indicate that it has received the data.

If there is no acknowledgement received, communication can be stopped or reset.

Several slaves can be plugged on the same I2C bus. So, each slave must have its own address. The address is encoded in eight bits, made of a fixed part (constructor-dependent), a configurable part (hardware-configurable) and a last read/write bit which defines the communication sense (0 for writing, 1 for reading).

Communication begins with a start bit, followed by the address (coded in eight bits), the acknowledgement, a data byte, a new acknowledgement bit and is finally terminated by a stop bit, as shown below:

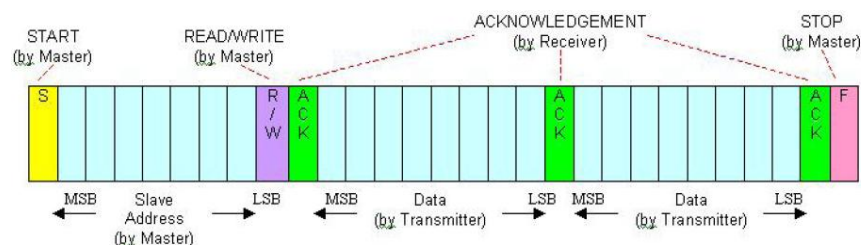


Figure 17: I2C communication

**Data rates:** Up to 400 Kbps

**Debugging tools:** Existing tools from the hardware testing industry.

#### 4.3.1 SPI versus I2C

These two buses have often been compared. The following table contains a number of known pros and cons for each:

|                      | SPI  | I2C   |
|----------------------|--|---|
| <b>Application</b>   | Better suited for data streams between processors                    | Occasional data transfers. Generally used for slave configuration                         |
| <b>Data rates</b>    | >10 Mb/s   | < 400 kb/s  |
| <b>Complexity</b>    | 3 bus lines<br>More wires more complex wiring<br>More pins on a chip | Simple, only 2 wires<br>Complexity does not scale up with number of devices               |
| <b>Addressing</b>    | Hardware (chip selection)  | Built-in addressing scheme  |
| <b>Communication</b> | No acknowledgment mechanism,<br>Only for short distances             | Better data integrity with collision detection, acknowledgment mechanism, spike rejection |
| <b>Specification</b> | No official specification  | Existing official specifications  |
| <b>Licensing</b>     | free   | free  |

Table 8: Pros and cons of SPI and I2C buses

#### 4.4 The NFC-WI bus (also called ECMA 373 and S2C)

The NFC-WI bus is defined by the ECMA-373 and can be used between the CLF and the SE in the case of eSE only.

Following the standardisation of NFC systems, this standard specifies a two-wire interface between two components called 'transceiver' and 'front-end'. Systems that implement the NFC-WI interface can thus be augmented with, for example, a wireless Front-end for NFCIP-1, as illustrated in Figure 19, below:

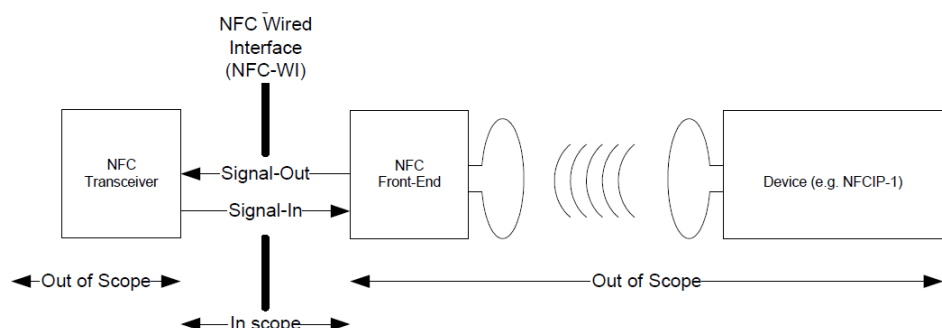


Figure 18: the NFC-WI interface

NFC-WI protocol is limited to ISO14443 type A in card emulation mode. ISO14443 type B and the reader mode are not supported by NFC-WI. Clock rate is 13.56 MHz +/- 7 kHz.

Signal In / Out is coded in Manchester and Modified Miller bit coding schemes and combine with the clock differently, depending on the bit rate, as shown below:

#### Data rates:

- Data rate 126 kb/s (F Clock/128)
  - Signal Out is coded in Modified Miller coding AND-combined with F Clock
  - Signal In is coded in Manchester coding OR combined with F Clock/16
- Data rate 212 kb/s (F Clock/64)
  - Signal Out is coded in Manchester coding XOR-combined with F Clock
  - Signal In is coded in Manchester coding
- Data rate 424 kb/s (F Clock/32)
  - Same coding scheme as F Clock/64 (like above)

#### Waveforms

##### ■ Modulation index $m = 100\%$

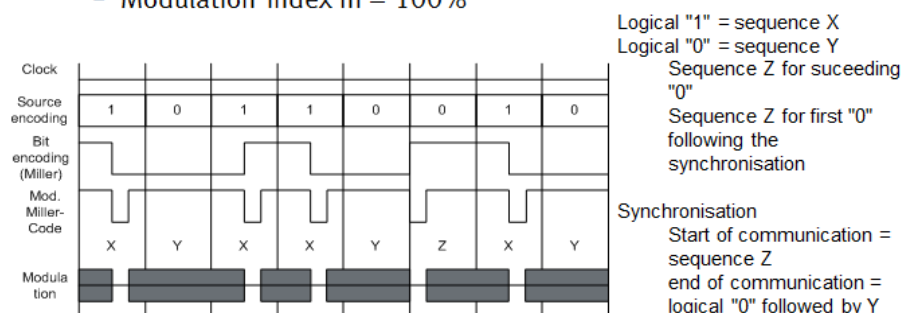
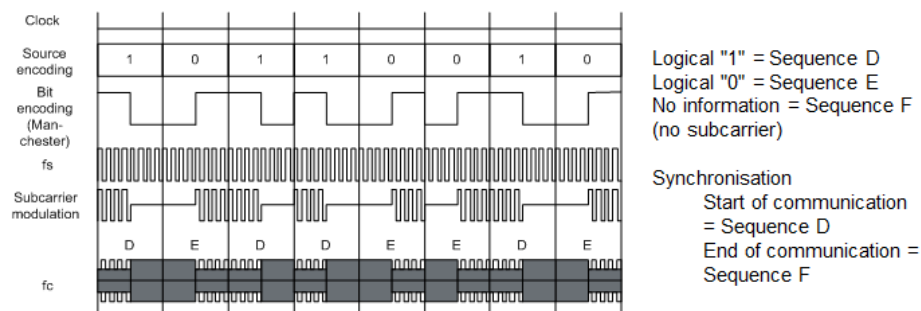


Figure 19: Modified Miller coding scheme (with a 100% modulation index)

### ■ On Off Keying (OOK) amplitude modulation



**Figure 20: Manchester coding scheme (Example: OR combined with Fclock/16)**

**Debugging tools:** Dedicated tools from the NFC / smartcard test industries or from the Oscilloscope industry.

## 4.5 The Inter-chip USB protocol

The Inter-chip USB is defined in the ETSI TS 102 600 specification and can be used between the handset's processor and the SE (or the UICC). This protocol is not available in case of smart microSD.

Inter-chip USB is the official high-speed interface between a handset and a UICC standardized by the ETSI. At this stage, UICCs using this interface yet to be widely deployed.

Inter-chip USB (introduced in March 2006) is a supplement to the USB 2.0 specification from which it is derived.

It is a low power variant of the standard USB interface dedicated to communication between components on embedded systems, i.e. on the same printed circuit, with a restriction on the maximum inter-chip distance, which is up to ten centimetres.

Inter-chip USB consists of two wires:

- IC\_DM: Inter-Chip USB D- data line
- IC\_DP: Inter-Chip USB D+ data line

Additionally, power [IC\_VDD] and ground [GND] signals are also required.

Three classes are defined within the inter-chip USB:

- ICCD: Integrated Circuit Card Devices which enable historical UICC interface to be emulated. ICCDs also transport ISO 7816-4 APDU exchanges at a higher speed.
- EEM: Ethernet Emulation Model which supports TCP/IP or UDP/IP and enables IP packets to be transported, as defined in ETSI TS 102 483.
- Mass Storage: For mass storage device emulation.

**Data rates:** up to 12 Mbps half duplex

**Debugging tools:** Dedicated spy tools or simulators from the smartcard test industry or from the Oscilloscope industry

#### 4.6 The DCLB protocol

The Digital Contactless Bridge (DCLB) was defined by Infineon to interconnect a CLF and an SE, and is licence-free. This protocol relates to the eSE but does not apply to UICC and microSD.

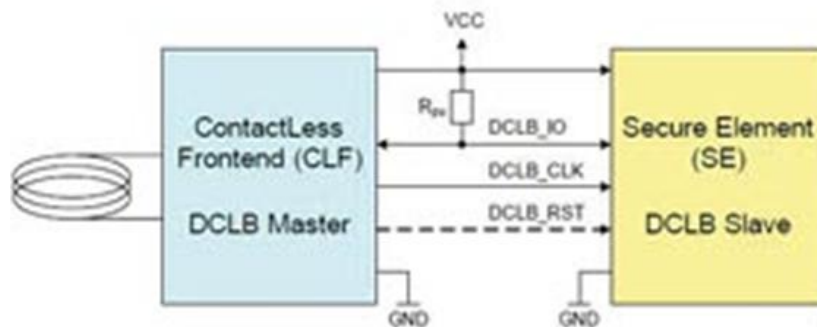


Figure 21: The DCLB interface

The DCLB bus consists of three signals, like shown above:

- DCLB\_Clock
- DCLB\_RST
- DCLB\_IO

**Data rates:** up to 848 kb/s

**Debugging tools:** Dedicated spy tools from the smartcard test industry

#### 4.7 The SD bus

The SD bus can be used to interconnect the handset's processor to the SE in the microSD case only.

The SD interface is defined in the SD specifications issued by the SD Card Association (first release in 2000).

SD Memory Card is a memory card that is specifically designed to meet the security, capacity, performance and environmental requirements inherent in newly emerging audio and video consumer electronic devices.

The Secure Digital format includes five card families available in three different form factors. The five families are the original, Standard-Capacity (SDSC), the High-Capacity (SDHC), the eXtended-Capacity (SDXC), the SDIO, which combines input/output functions with data storage, and the recently defined smartSD (short form of smart microSD card) which is a microSD card supporting ASSD (Advanced

Security SD Extension). This last family is the family used as SEs and is the SD family card that is the focus of this document.

This SmartSD family can be available in several capacities: Standard-Capacity (SDSC), High-Capacity (SDHC) or eXtended-Capacity (SDXC).

SD specification defines three form factors: original, 'mini', and 'micro'.

When referring to the SE used in a mobile environment (SmartSD card), the 'micro' size is the only form factor used.

The SD Association has defined a logo for the SmartSD card which is the combination of the microSD logo and the 'smart' pictograph.



#### 4.7.1 Access to the Secure Element

Recently, the SD Association has adopted a new physical interface to support the SWP interface.

The SDA has adopted the following solution which is compliant with new ultra high speed interface defined by SDA (UHS-II) (see above).

A new PIN is defined supporting the SWP signal (SWIO). Vdd2 pin (1.8V) is shared with UHS-II interface for NFC operation.

Specific electrical characteristics are defined.

HCI was added in order to address the smart microSD card and get extra information (HostID for microSD and specific register).

**Physical SWP pin out adopted by the SDA:**

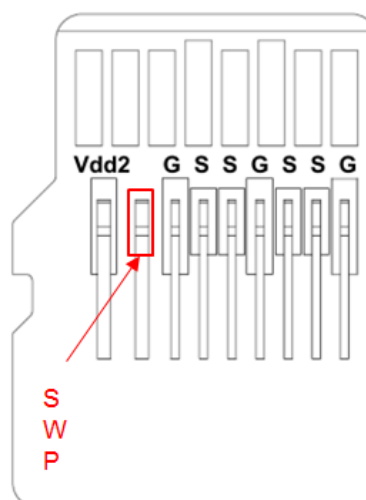


Figure 22: SWP support in the microSD

Alternate configuration

In order to support some specific legacy implementations, SDA has decided to support an alternate configuration for the SWIO PIN. In this configuration:

- The SD3.X pin definition for contactless support (Direct connection to an antenna) is reused
- Connectors are already available
- Time to market is shorter

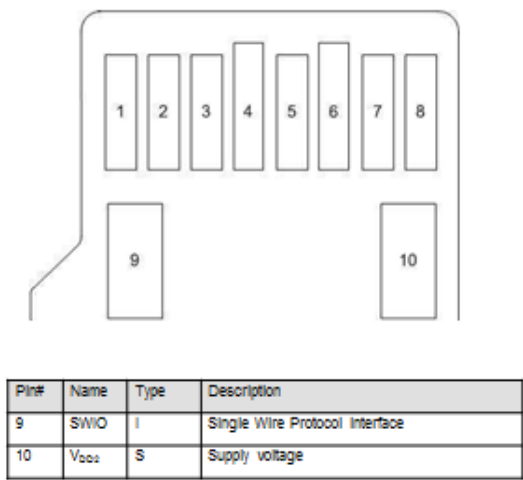


Figure 23: Alternate configuration for SWP support in microSD

**Interoperability note:**  
*This configuration is not compliant with the UHS-II standard. It is not future-proofed and should be avoided for new deployments.*



## 5. Remote management

OTA management in UICC services has always been a key element in the development of the UICC ecosystem. Since the very beginning secure remote applet management and file management as well as remote interaction with applications stored on the card has been part of the core set of UICC specifications.

With the embedded SE, the need for remote file and application management increases: having a soldered device that does not allow for easy replacement defines new management scenarios, including the management of contactless features and the utilization of the SE's capability to use fast bearers. From Release 6, ETSI and GlobalPlatform specifications have evolved to meet these new scenarios.

### 5.1 Evolution of OTA protocols

With Release 6, the BIP mechanism and CAT-TP protocol have been introduced as defined in ETSI TS 102 223 [13] and in ETSI TS 102 127 (see [10]), allowing fast bearers to be used to perform RFM and RAM. In Release 7 support for applications using the CAT-TP has also been introduced.

With the ETSI TS 102 226 Release 6 [16], the main changes introduced were the Remote Commands in Expanded Data Format supported by Remote Management Applications, adding more flexibility to the script execution engines.

In Release 7, these commands go beyond card content and file system management APDUs by also including proactive commands. Another feature introduced by Release 7 is Data Download via USSD (Unstructured Supplementary Service Data), specified in 3GPP 31.115 [53] which specifies the use of USSD application mode, enabling the transparent transport of data between an application residing in the network and a UICC based application.

As a result of this development, Release 7 allows the adoption of fast applet downloading and files management via OTA, by using the BIP protocol and the CAT-TP. The Release 7 OTA protocols have been described in Stepping Stones Release 7 (ref. [58], §15, §18).

In Release 9, OTA protocols have continued their evolution to take into account new requirements from the NFC ecosystem.

The ETSI TS 102 226 Release 9 [16] introduces the RFM and RAM over HTTPs protocol, defined in GlobalPlatform 2.2 Amd B [43]. Here, the RAM over HTTPs is the mechanism for an application provider to perform Remote Application Management (RAM) of its application according to TS 102 226 i.e. to load, install and personalize using the HTTP protocol (see [65] RFC 2616) and PSK-TLS security Over-The-Air. Remote File Management via HTTPs is specified by the above mentioned ETSI TS, by adding a RFM scenario to the RAM scenario described in [43].

Another important update is the reference to the Amendment C of the GlobalPlatform 2.2 [44]. This specification introduces the management of parameters for contactless applications. It defines mechanisms, parameters, and interfaces to be set-up and maintains the configuration of applications, controls their access to system resources like communication interfaces and memory, and focuses on parameters and mechanisms required for applications in card emulation mode.

Leveraging the RAM over HTTPs protocol defined for UICC in GlobalPlatform 2.2 Amendment B [43], GlobalPlatform defined an extension available for other types of Secure Elements (eSD and Smart microSD). This extension, also called GP SE OTA, is defined in GlobalPlatform Device, Secure Element Remote Application Management [10]. Thanks to this specification,

service providers can manage their applications using the same mechanisms and protocols in an interoperable manner regardless of the type of SE they utilise.

## 5.2 Secure Channel protocols overview

During the evolution of GlobalPlatform specifications, several secure channel protocols have been defined in order to provide applications with cryptographic services.

### 5.2.1 Secure Channel Protocol 02

SCP02 has been introduced mainly in banking environments to enable the secure management of payment applications, as described in § 5.3.

### 5.2.2 Secure Channel Protocol 80

SCP80 is the name by which GlobalPlatform identifies the OTA secure protocol defined in ETSI TS 102 225 [15] and previously defined by 3GPP 03.48 Rel-5. The SCP 80 and the ETSI TS 102 225 [15] are explained in NFC Stepping Stones 2011 [61].

### 5.2.3 Secure Channel Protocol 81

SCP81 represents the TLS secure protocol defined in GP 2.2 Amd. B [43] adopted by RAM over HTTPs (defined in GP 2.2 Amd.B as well) and by RFM over HTTPs (defined by ETSI TS 102 226 Rel-9 [16]). The SCP 81 is explained in NFC Stepping Stones 2011 [61].

## 5.3 The Secure Channel Protocol 02

The SCP02 protocol allows securing a direct connection over the ISO interface between an external entity and an on-card security domain. It replaces the deprecated SCP01 that showed specific security flaws.

Actually, the SCP02 represents a family of protocols: it allows several options that modify basic behaviors; those options are identified by an option byte usually defined as *i* (implementation option).

The implementation option changes several features of the protocol. Two implementation options are mainly used in telecom environments:

- the *i=0x15* foresees that the SE provides the external entity with a generated *card random* that is used to compute the cryptographic material used in subsequent activities.
- the *i=0x55* foresees that the external entity may compute the cryptographic material without retrieving the generated card random.

The second option is mostly adopted in telecom environments due to the fact that an SCP02 script may be pre-computed and does not depend on a specific card random. This allows secure scripts to be prepared by third parties and forwarded to the SE.

### 5.3.1 SCP02 APDUs

To setup an SCP02 session, two APDUs are exchanged between the external entity and the on-card security domain:

- **INITIALIZE UPDATE:** the APDU indicates which Key Version Number is used and provides the SE with a host challenge. The SE answers with data that is used to create the cryptographic material including, in case of *i=55*, the Card Challenge.

- **EXTERNAL AUTHENTICATE:** the APDU provides a host cryptogram computed according to the implementation option. The APDU indicates the required security level to be applied in subsequent messages.

Once the EXTERNAL AUTHENTICATE host cryptogram is verified by the SE, the secure channel is set up.

Several security levels are then possible depending on the EXTERNAL AUTHENTICATE content:

- No secure message: this security level leaves the content of the APDUs unchanged. It is used just to ensure authentication obtained by means of the successful execution of the EXTERNAL AUTHENTICATE.
- C-MAC: this security level extends the content of any APDU by a cryptographic MAC computed on the APDU itself with the key material generated at the EXTERNAL AUTHENTICATE time. This security level ensures authentication and integrity but not confidentiality.
- DECRYPTION and C-MAC: this security level adds to the previous one the encryption of the APDU content (not the APDU header), ensuring confidentiality on top of authentication and integrity.

When the APDUs are modified to add a cryptographic MAC (C-MAC) or by the encryption of the APDU content (DECRYPTION), the class byte is modified accordingly:

- bit b4=1, bit b3=0 indicates that the Secure messaging is included but C-MAC is not included
- bit b4=1, bit b3 =1 indicates that the Secure messaging is included together with C-MAC.

The R-MAC feature is not available for the considered implementation options and hence is out of the scope of this document.

### 5.3.2 SCP02 Keys

A complete key set indicated in a security domain (SD) is made by three different keys:

- The Secure Channel Encryption Key (S-ENC), used for ensuring authentication and confidentiality of the SCP session by enciphering the APDU payload in a session with DECRYPTION and C-MAC.
- The Secure Channel Message Authentication Code Key (C-MAC), used for ensuring authentication and integrity of the SCP session by computing the MAC of the APDU payload in a session with C-MAC.
- Data Encryption Key (DEK), used for enciphering new keys delivered to the UICC via the PUT KEY command.

The above keys are not used directly in a SCP02 session but, when the session is initialized, three session keys are derived from the above keys by enciphering a byte string made by:

- A constant (2 bytes, equal to '0182' in case of S-ENC, to '0101' in case of C-MAC, to '0181' in case of DEK)
- A counter (2 bytes)

- 12 bytes set to '00'

A key set is complete when all the three keys are present.

Several key sets may be present for each security domain. The INITIALIZE UPDATE APDU indicates, in the Key Version Number, which Key Set shall be used.

**Interoperability Note:**

*It is not specified how many SCP02 key set shall be available for each SD. SIMalliance members agree that neither is it possible to specify at product configuration time the number of key sets available for newly installed security domains if the card framework does not dynamically manage the required keys upon request.*

An evolution of the SCP02 is SCP03 that replaces the DES protocol by using the AES protocol.

## 5.4 Stacking the protocols: SCP02 over SCP80/SCP81

A new feature in GlobalPlatform 2.2 is the capability of using two levels of security protocols in the same session.

Usually the two stacks are:

- An SCP02 enveloped in an SCP80 session
- An SCP02 enveloped in an SCP81 session

This is obtained by:

- 1) Applying the SCP02 security to the script to be sent to the SE
- 2) Applying the SCP80 or SCP81 security to the script obtained in point 1)

The stacking of protocols has been introduced for three main reasons:

- It allows a double level of authorization of the OTA manager (e.g. the operator) and of the service manager (e.g. the bank). Without both authorizations, the actions are not allowed
- Existing personalization systems of many service providers are already based on the SCP02, but do not account for OTA personalization; by enveloping such messages via OTA it is possible to reuse existing personalization systems
- The stacking of the two protocols allows the introduction of the Admin Agent as specified below.

Each SCP protocol in a stack is usually managed by a different security domain.



**Figure 24: SCP protocols**

**Interoperability note:**

*SCP02 has several variants in order to cater for different banking environments; the variant “i=55” is considered by SIMalliance members to be the most suitable for telecom environments and hence the most interoperable.*

**Interoperability note:**

*The above scheme is specified for Remote Applet Management application. The extension to Remote File Management application is not standard and hence is not considered to be as interoperable.*

## 5.5 Third party application management

The SCP02 over SCP80/SCP81 is a mechanism that allows separation between the service provider and the card issuer. Usually the SCP02 over SCP80/SCP81 scheme works for personalization of the services and for direct interaction with the application.

Other mechanisms are more dedicated to application management, in particular to package load and application installation.

Depending on the requirements, there are several schemes relating to the download of third party applications:

- *The card issuer performs the download but it is required that the service provider verifies the download*

In this scheme, usually a DAP (Data Authentication Pattern) verification scheme is adopted. The Service Provider Security Domain is configured by a DAP privilege and a DAP key. Every application download that requires being associated to the Service Provider Security Domain requires a cryptographic hash (DAP) of the operation to be generated by the service provider. If the DAP is not present, the operation fails.

In this scheme usually the Service Provider Security Domain has no direct interaction with an OTA server and hence it is not required to have the SCP80/SCP81 keys.

This scheme allows the card issuer to be the only one able to perform card content operation via OTA, but the service provider authenticates all the card content operations directed to the security domain.

- *The service provider performs the download but the card issuer controls the operation performed (Delegated Management)*

In this scheme, a token verification approach is usually adopted. The Service Provider Security Domain prepares the operations that perform the loading, installation and deletion operations and requires the card issuer to generate a digital signature which enables the operation to be downloaded by the service provider. The operation also generates a cryptographic receipt.

This scenario is configured by assigning the delegated management privilege to the Application Provider Security Domain (APSD) and the token verification / receipt generation privileges to the Card Issuer Security Domain.

- *Dual management (Authorized Management)*

In this scheme, the card issuer assigns specific resources to the 3<sup>rd</sup> parties (quotas) of non-volatile and volatile memory. The 3<sup>rd</sup> party can execute card content operations autonomously between the quota boundaries.

The various schemes may coexist on the same SE: a TSM could have a security domain with the authorized management and the token / receipt privileges, with a APSD with DAP, etc.

In addition to the above schemes – that are used with the main scope of authenticating the parties performing the operations - to ensure confidentiality of the CAP file that is downloaded, a new mechanism has been introduced in GlobalPlatform 2.2 Amendment A that allows the download of CAP file enciphered with a specific dedicated key.

## 5.6 Personalized SDs

Security domains offer security mechanisms to applications and protocols on the SE.

A security domain may offer different security mechanisms depending on:

- The installation parameters (e.g. in the tag 0x81 of the applet specific parameters it is specified if SCP02 is supported, if SCP80 is supported, etc.)
- The SD privileges (e.g. Token Verification, DAP management, etc.)

In order to be able to use a security domain, the security domain itself must be in the personalized state, i.e. it must have at least one complete key / keyset for each of the supported protocols.

If security domain is not in the personalized state, or if the relevant SCP protocol is not supported by the SD, the management of the security of the incoming data remains the responsibility of the parent security domain.

An issue of security domain personalization is the confidentiality of the security domain keys, in particular for security domains that are installed after SE issuance. To solve this issue a new role as been defined by GlobalPlatform in GlobalPlatform 2.2 Amendment A [42]: the Controlling Authority (CA). The Controlling Authority provides mechanisms to secure the key creation for Application Provider's Security Domain (APSD) in a confidential manner (the keys shall not be known by the entity performing the OTA card content management) and to securely and confidentially perform the personalization of this newly created security domain. The controlling authority is a party trusted by both the SE issuer and the service providers. The controlling authority representative on the SE is the *Controlling Authority Security Domain* (CASD). The CASD is unique on the SE, it shall be installed and fully personalized with the CASD keys and data before the card is provided to an issuer (e.g. MNO). Several scenarios are defined in GlobalPlatform 2.2 Amendment A [42] including Push Model scenarios, where keys are pushed in a confidential manner to the APSD and a Pull Model scenario, where the keys are generated on-card and send back to the Application Provider in a confidential manner. Depending on the scenario, the keys of the CASD will be used to decipher the incoming keys of the application provider or to cipher the keys generated on-card to send them to the application provider.

## 5.7 Two Card Content models

Depending on the capabilities offered to the SE there are two different models applicable to perform card content management:

- **CAT SE:** if the SE/Handset interface provides Card Application Toolkit capabilities according to ETSI TS 102 223 [13], the SE can take advantage of the CAT interface to access toolkit and BIP powered protocols, including CAT-TP and SCP80 over SMS.

This is typical true for the UICC and not true for the eSE.

- **CATless SE:** Otherwise, if the SE/Handset interface does not provide CAT capabilities, the network interface is usually performed by an Admin Agent that resides on the Mobile Phone and interfaces with the SE. This approach is also valid for CAT SE (i.e. the UICC).

●

| Remote management Protocol       | Availability  |
|----------------------------------|---|
| RAM/RFM over SMS (SCP80)         | Only CAT SE (UICC)  |
| RAM/RFM over CAT-TP (SCP80)      | Only CAT SE (UICC)  |
| RAM/RFM over HTTPs (SCP81)       | CAT SE (UICC)<br>CATless SE (eSE) with a Companion application  |
| Second level of security (SCP02) | CAT SE (UICC) with/without a Admin Agent on the Handset<br>CATless SE (eSE) with a Admin Agent on the Handset |

**Table 7: CAT SE transport protocols**

For CAT SE transport protocols, please refer to NFC Stepping Stones 2011 [61].

A general scheme that is applicable to both CAT SEs (UICC) and CATless SEs (microSD and eSEs) is the stacking of SCP02 over SCP81. Even though management of the SCP81 protocols is operated by different entities (directly by the UICC in one scenario and by the Admin Agent on the Handset in the other), the interaction with the OTA remote server would be the same across both models, allowing the management of both categories of SE in the same way.

## 5.8 The Admin Agent

In order to connect CATless SE (such as the eSE) to the network to be able to interact with a remote platform, the concept of Admin Agent (or Admin Proxy) is introduced.

The Admin Agent is an application deployed on the handset that interacts with a remote platform managing the network protocol and the interaction with the SE.

A typical scenario is the RAM/RFM over HTTPs:

The RAM over HTTPs encapsulates scripts of RAM in a HTTPs request. In this approach, the Admin Agent manages the external HTTPs request / response and forwards the internal RAM script to the SE.



If the Admin Agent application is implemented as a third party application (e.g. an Android app or an iPhone app), then it has to access the handset / SE interface in order to forward the APDUs to the SE. This is typically performed by using the Open Mobile API by SIMalliance [60].

**Interoperability Note:**

*Deployment and personalization of the companion application is not defined and hence can introduce interoperability issues.*

*In particular as the companion application requires a specific shared key or public key certificate to secure the HTTPs connection, it is not specified how this specific key is configured in the application.*



## 6. Service development

With the introduction of additional SE architectures, the existing paradigms of user interaction and application deployment have been enriched, and more suitable for eSE and smart microSD form factors.

There are three main paradigms to provide user interaction with applications residing on the UICC:

- The Toolkit interaction, based on ETSI TS 102 223 and already described in Stepping Stones Release 7 [58]
- The Smart Card Web Server (SCWS), based on OMA SCWS and already described in SCWS Stepping Stones [59]
- An Handset Application (Admin Agent), already explained in § 5.8 for the RFM / RAM and better detailed in the following.

The Toolkit and SCWS technologies require that the SE interacts with the handset by using a Toolkit interface (SCWS is based on BIP commands that are Toolkit commands). The toolkit interaction is specified only for the UICC, reducing the options for user interaction in eSE and microSD to the Admin Agent only. As a consequence, in order to define a service that is deployable also on eSE and smart microSD, the Admin Agent approach has to be followed.

However, Toolkit and SCWS present specific advantages with respect to the Admin Agent approach: Toolkit is supported by almost any handset and SCWS allows the application to fully manage the user interaction.

As a consequence, for services that could be deployed on the UICC, it is suggested to define the application also with Toolkit and SCWS interfaces to get advantage of all the technologies.

### 6.1 Handset application components for SE services

NFC services are deployed mainly on the SE, especially the items related to security (keys, passwords, etc.). It is common to extend the capabilities of the SE service to support other applications that reside on the handset, so that the service is made by two parts: the SE based application that provides the security, the algorithm and the service logic, and the handset application that provides the user interface and, in some cases, the connectivity.

Several specifications exist to allow the interaction between the handset application and the SE application. SIMalliance has developed one of the most successful of these, called the Open Mobile API (see [60]).

The evolution of this scheme has led to security concerns related to the extent to which handset applications are able to interact with the UICC. This problem has been solved by introducing the access control mechanisms, which are discussed below.

The SE based application is usually a Java Card application. This application can access the Java Card mechanisms to provide authentication and cryptographic services to third parties. Java Card and related APIs have already been described in [58]. Additionally, GlobalPlatform has also specified in its configuration the minimum list of APIs available on SEs.

#### **Java Card APIs:**

- KeyBuilder, with support for DES, AES and RSA

- MessageDigest, with support for SHA, SHA256
- Checksum, with support for CRC16 and CRC32
- Random Data
- Signature
- Cipher

**GlobalPlatform APIs:**

- Application and Personalization interfaces
- HTTPAdministration interface (if the SE supports SCP81)
- GlobalService mechanism
- The Authority interface if the Controlling Authority mechanism is supported.

This set represents a minimum of required support that must be provided by the SE. For additional details, please refer to [45].

## 6.2 Access Control

To prevent unauthorized handset applications from accessing the SE and to prevent the unauthorized use of some resources in the SE, a SE access control is needed. Typically denial of services attacks (e.g. PIN blocking, or selection of non multi-selectable applets) can be avoided using the access control mechanism.

GlobalPlatform has defined a Secure Element Access Control [46] usable with any kind of SE (eSE, SE, Smart microSD and UICC). The Access Control rules are stored in the SE and are enforced inside the device by an Access Control Enforcer. This security mechanism is used in addition to existing protection mechanisms (such as permissions or security OS policy limiting access to sensitive APIs).

The Secure Element Access Control defined by GlobalPlatform supports the management of multiple applications from different application providers. Each application provider sets access rules that are stored in the SE and used by an enforcer in the device. The enforcer applies the rules and restricts the access to the SE accordingly. This enforcement mechanism is transparent for client applications in the device.

The architecture of the GP Secure Element Access Control [46] supports different deployment models:

- Access Rules can be managed by the issuer only (applying for example for 'issuer-centric' model). In this case, the issuer supplies the Application Access Rules on behalf of application provider in the ARA-M. A business agreement between issuer and application provider is then required.
- Access Rules can be managed by the issuer but also by application providers themselves (applying, for example, the 'consumer-centric' model). In this case the issuer provides to the application provider all the necessary data for the installation of the ARA-C in the Application Provider Security Domain. A business agreement between issuer and application provider is then required. Access rules are then supplied by the Application Provider in the ARA-C.
- On UICC, Access Rules can be defined in a PKCS#15 file system structure to ensure compatibility with former implementations of an access control mechanism. In this case, the Access Rules are managed by the UICC issuer (i.e. the MNO). A business

agreement between issuer and application provider is then required for the management of Access Rules by the issuer on behalf of the application provider.

The Access Control solution defined by GlobalPlatform is composed of several architectural components:

In the handset:

- The Access Control Enforcer is embedded in the implementation of the SE Access API of the device operating system. This Access Control Enforcer retrieves the access control rules from the Access Rule Application Master (ARA-M) in the SE, enforces these rules and restricts access to the SE according to these rules. More precisely, when a device application invokes the SE access API to open a connection with an application inside the SE, identified by its AID, the Access Control Enforcer check the Access Control Rules to grant or not the access by the device application. If the access is granted, the connection between the device application and the SE application is opened and the device application is allowed to interact with the SE application, i.e. to send APDU commands, provided those APDUs commands are allowed by the Access Rules. The Access Rules are retrieved by the Access Control Enforcer using the GlobalPlatform commands GET DATA(). Two modes are defined for the Access Control Enforcer to retrieve the Access Rules:
  - Retrieval of all the Access Rules and use a cache mechanism to avoid repeatedly retrieving access rules from the SE. In this case the Access Control Enforcer shall check whether a new version of the rules is available (i.e. refresh tag updated) prior to applying cached rules
  - Retrieval of a specific access rule for a defined SE application and a defined device.

In the SEt:

- The **Access Rule Application Master (ARA-M)** that provides the interface to the device Access Control Enforcer to retrieve the Access Rules stored inside the SE. The ARA-M is in charge of consolidating all the access rules defined: the ones stored in the ARA-M itself, the ones stored in the different ARA-Cs if any and optionally the ones stored in the ARF. The ARA-M is an application under the Issuer Security Domain. The ARA-M can also contain some Access Rules. The ARA-M provides an OTA interface for the issuer to update these Access Rules, The Access Rules can be updated using GlobalPlatform command STORE DATA().
- The **Access Rule Application Client (ARA-C)** is an application that can be instantiated under the security domain (SD) of an application provider. This ARA-C contains the Access Control Rules defined by the application provider and so, associated to the SE applications managed by this application provider. Access Rules may be downloaded or updated by the application provider through an OTA connection using GlobalPlatform command STORE DATA(). There may be several ARA-Cs in the SE, one for each Application Provider SD. This ARA-C is an optional entity if the SE Access Control is implemented.
- The **Access Rule Files (ARF)** is a PKCS#15 file system structure that can be defined in a UICC to define Access Control Rules. This structure is defined to manage backward compatibility with former device Access Control Enforcer that did not implement the ARA-M interface (GET DATA() command) but a file system based interface as defined initially by GSMA. The Access Rules stored in the ARF may be updated by the issuer through OTA connection using ETSI Remote File Management (RFM) commands. For a UICC, the following behavior is defined:
  - If the ARA-M is not present, the device Access Control Enforcer shall retrieve the Access Rules from the ARF.

- If the ARA-M is present, the Access Control Enforcer shall not retrieve the access rules from the ARF. In this case, the ARA-M might take into account the Access Rules stored in the ARF when consolidating all the Access Rules.

The figure below, extracted from the GP Access Control specification [46] shows the interaction between those components.

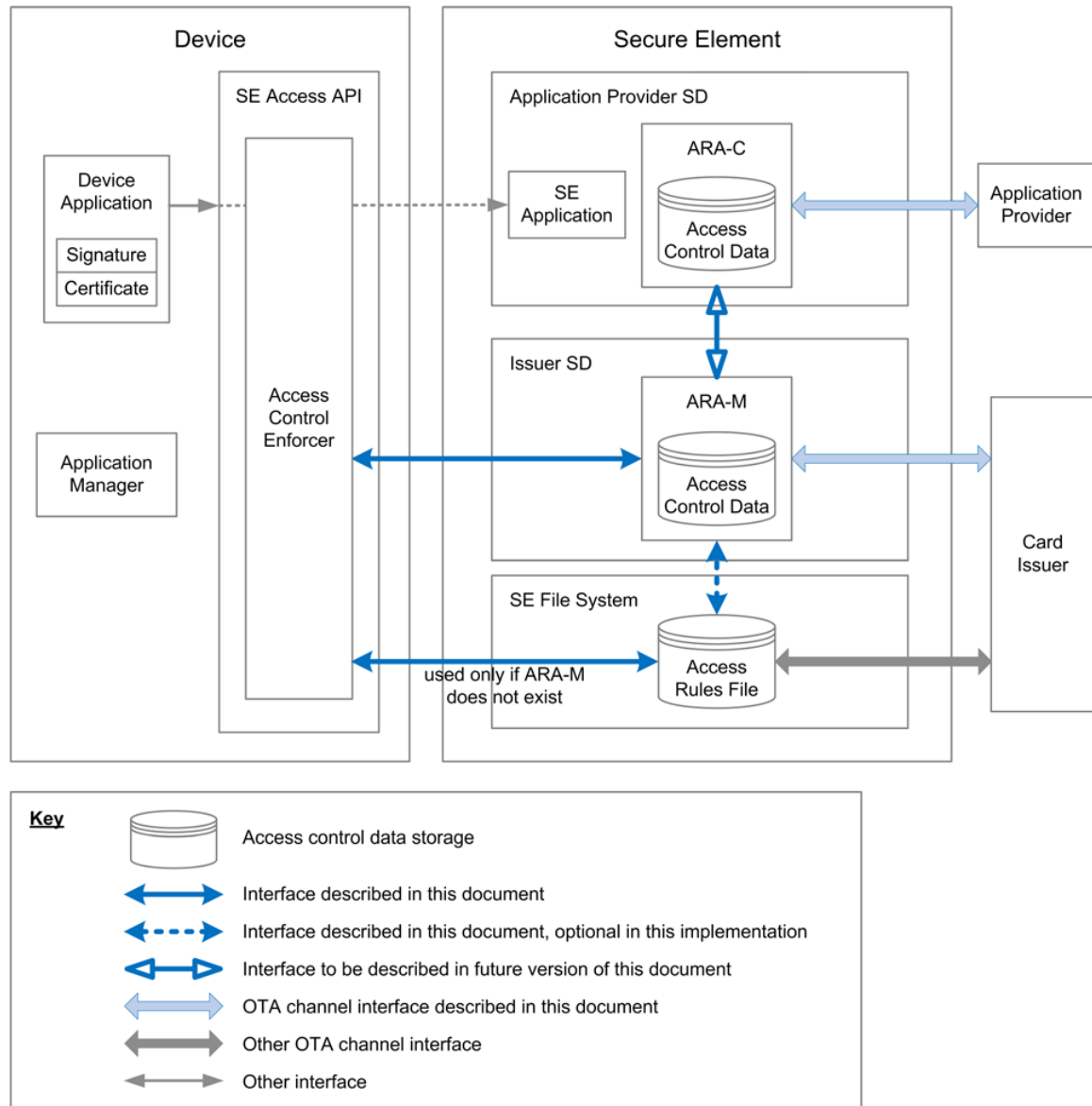


Figure 25: Architecture of Access Control

When a device application invokes the SE access API to open a connection with an application inside the SE, identified by its AID, the Access Control Enforcer check the Access Control Rules to grant or not the access by the device application. If the access is granted, the connection between the device application and the SE application is opened and the device application is allowed to interact with the SE application.

According to GP Secure Element Access Control specification [46], an Access Rule specifies that for a given SE application (or all SE applications on a given SE), all or selected device applications have access rights to:

- All APDUs, no APDUs, or selected APDUs
- NFC transaction events or no NFC transaction events

The SE application is uniquely identified by its AID whereas the device application is uniquely identified by the hash value of the certificate of its application provider.

As a device application is identified through the hash of the certificate of its application provider, the application provider can define a single Access Rule that applies for several of its device applications provided that device applications are signed with the same certificates.

Note: The GlobalPlatform specification clarifies the expected behavior when the device application is signed with a certificate within a certificate chain.

The device Access Control Enforcer shall also implement the following behaviors:

- Access to an application in a UICC shall always be denied except if explicitly granted in an Access Rule in the ARA-M/ARA-C or the ARF.
- Access to all applications in a SE other than the UICC shall be granted when the ARA-M is not installed. Once the ARA-M is installed, the access to an application in this SE shall always be denied except if explicitly granted in an Access Rule in the ARA-M/ARA-C.

In addition, GlobalPlatform specification clarifies:

- How Access Rules are combined if several Access Rules apply to the same target SE application.
- How to resolve potential conflicts between Access Rules.

### 6.3 The Mobile Wallet

The Mobile Wallet is a software application that is loaded in a mobile phone for the purpose of managing payment applications from the mobile phone and the SE. A mobile wallet application can also be used to hold and control a number of other applications (for example, loyalty), in much the same way as a physical wallet holds a collection of physical cards.

The user interface for a wallet application typically runs in the non-secure memory of a mobile device and facilitates user interaction with the payment application or applications running within the SE (supported features may include PIN entry, transaction history review and OTA functionality).

The user interface applications (UIs), also often referred to as wallets, are designed to provide the end-user, or cardholder, access to the payment applications and in turn, control over certain features relating to making payments using the payment applications that are installed on the SE within a mobile device (regardless of architecture – e.g., SWP UICC or eSE).

In general, the mobile wallet is a component that can be deployed and designed for several mobile operating systems and with requirements from third parties; payment organizations, as an example, define minimum requirements for managing payment applications.

Even though general requirements are defined, architectures and interfaces of the wallet are not fully developed leading to a variety of implementation options. In the absence of a standard wallet architecture, the GSMA architecture defined in [73] will be considered as an illustrative reference.

## 6.4 Role of the wallet in the SE

The wallet has to control a variety of different SP services, including their storage on separate SD, and their access to the SE in order to be sure that the transaction is accomplished in a secure environment.

Contactless and user-interface parameters are set in the Card Registry and cannot be changed by Java Card application by means of APIs.[60]

### Update of the Application Life Cycle (ContactlessServices Amendment C) .[60]

As explained in [61], a new life cycle state has been introduced in the application life cycle to manage the specificities of a contactless application: the Availability state. When an Applet is ACTIVATED, it is available on the contactless interface. It implies that its associated protocol parameters are reflected on the HCI RF gates of the CLF. When the Applet is installed, its state is inherited from the default state defined by its security domain. But, the state can be updated later by the Applet itself (with the associated Contactless Self Activation privilege) or by an Application having the Contactless Activation Privilege (i.e. the CRS application). If an Application has the Contactless Self Activation privilege, the CRS Application will not be involved in the activation of the Application, although the CRS Application will be notified of this activation. For all others cases, the CRS Application will process activation requests.

### 6.4.1 The CRS Application

The CRS Application is the Application which manages the Contactless Registry Service (only one per Secure Element).

The use of the CRS Application could induce interactions between the CRS and Applications in a different manner depending of the interfaces they exposed (see [61], § 3.1.3.2).

The CRS application may provide a “user-friendly” way (e.g. by Toolkit means) for the user to benefit of the services that could be provided by its Contactless Applications. When the user wants to activate a specific Contactless Application, the CRS Application will process the request and warn the user of the result:

- If there are no conflicts with ACTIVATED Contactless Applications, the user will be able to immediately use it.
- If conflicts are detected with existent ACTIVATED Contactless Applications, the CRS Application will provide to the user a way to deactivate all conflicting Contactless Applications. Then, the newly ACTIVATED Application will be available on the Contactless interface.

## 6.5 Role of the wallet in the handset

The mobile wallet is intended to facilitate the user experience, and allow the MNO or SP to differentiate by providing targeted and convenient access to the NFC Services within the mobile device and Secure Elements. The wallet application, for example, can typically list all SP services loaded into the mobile device or SE and displays their current status. Additionally, this application may also allow the user to manage the NFC settings of their mobile device.

## 6.6 General architecture of the Wallet

The mobile wallet is an application that manages the portfolio of mobile NFC services on the handset. It may also manage other services offered by mobile operators and their partners. The mobile wallet should always include some core features to support interoperability. The mobile wallet will enable the user to prioritize one NFC service over another, for example selecting an active payment card. In general, a mobile wallet application is likely to be the responsibility of the mobile operator, but there is considerable potential for other actors to establish influence over the technology as services providers.

### The service provider UI application

This application enables a user to manage a specific NFC service through a dedicated user interface (UI) and can also be launched by the mobile wallet. For the GSMA wallet white paper [70] this application, which runs on the handset, is referred to as the “UI app”. It may also contain other features that are not related to NFC. In general, the UI app is the responsibility of the service provider. In some cases, all the functionality needed by a given service provider may be included in the mobile wallet, in which case a separate UI app is not needed.

### The service provider applet

This application, which sits inside a secure domain reserved for the service provider on the UICC, securely manages NFC transactions.

### 6.6.1 General architecture of the Core Wallet and the Extended Wallet

Here is a short description of the wallet architecture as proposed by GSMA in their white paper “The Mobile Wallet” [73].

#### Core Wallet:

To provide both end-users and service providers with a consistent experience, mobile operators should ensure their mobile wallets adhere to the common set of basic principles and high level business requirements defined in this white paper – the core wallet functionality. The mobile wallet should also support interoperable standards defined by other standards bodies, where applicable. The core wallet should enable users to discover, install, update, run and uninstall mobile NFC services. Some of these activities may be implemented on a remote computer server, accessed via a mobile network, for which the wallet serves as a user interface.

#### The Extended Wallet

Beyond the core wallet functionality proposed by [73], the wallet issuer decides what other features and services are included in their wallet. By implementing optional features, referred to as the “extended wallet”, the service providers can offer mobile NFC services from within the wallet, rather than through their own UI app. These extended functions can be implemented in a generic way for specific services, such as payment, public transport ticketing, couponing, etc.

To provide such generic services, the extended wallet needs specific information from the service provider. The wallet dynamically interprets this information to offer the relevant NFC services and interacts directly with the corresponding UICC applet. This approach reduces the cost of testing NFC apps on the multitude of devices. The generic services supported by the extended wallet could include:

- Payment card functions, such as making a contactless payment or reviewing the balance on a specific account.
- Coupon management: receiving, displaying, managing and redeeming coupons.



- Receiving and redeeming NFC tickets for transport and events etc.
- Enabling access to a building.
- Providing service parameters specific to the kind of service (e.g. payment, loyalty, couponing, ticketing, building access). For example, the account number of the associated bank, expiry dates etc. for a payment service.

The following figure is part of [73] and describes how to extend a core wallet to support more functionality:

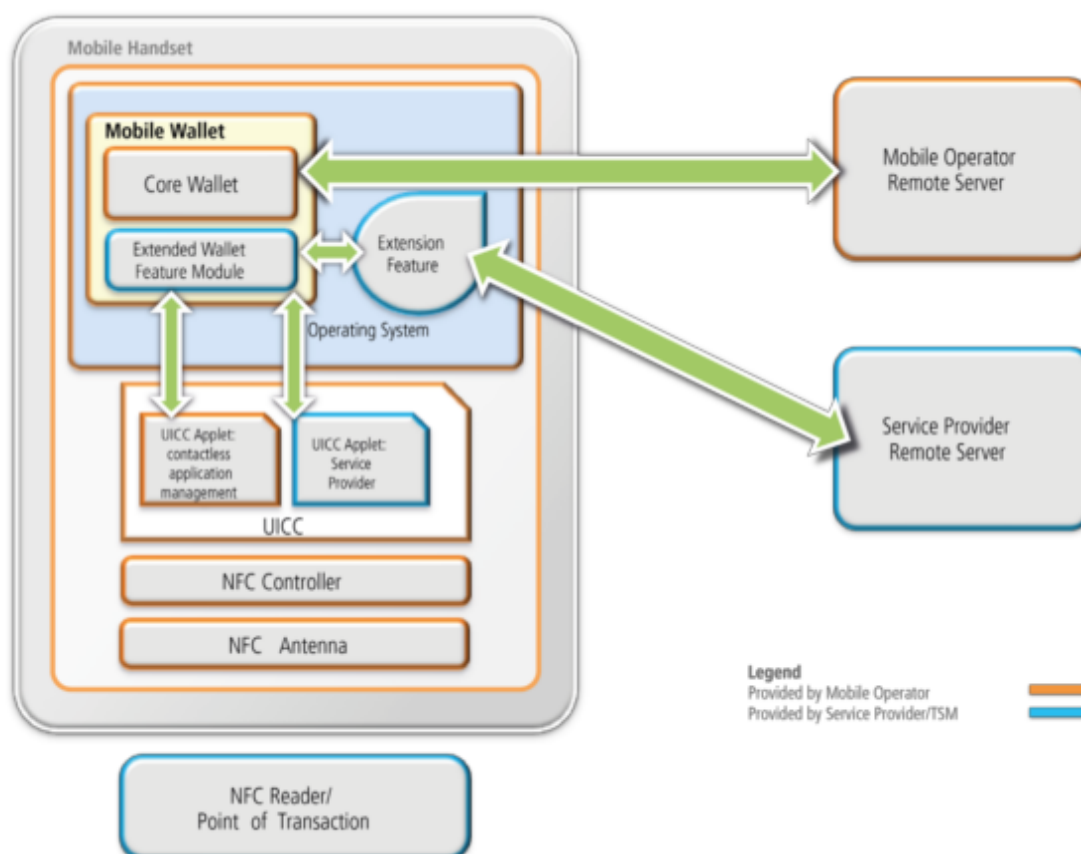


Figure 26: Example of a Wallet Architecture (GSMA proposition)

## 6.7 Wallet logic to choose SP Application

As referenced in section 7.1 (Role of the wallet in the SE) of this document, a new life cycle state has been introduced in the application life cycle to manage the specificities of a contactless application: the Availability state. When an SP\_Appllet is ACTIVATED, it is available on the contactless interface.

The logic to choose the activation of a SP\_Appllet by the NFC wallet is usually user driven but there are examples of other mechanisms. There's an even divide between mobile wallets that employ NFC and those that use GPS location-based systems for payment.



## 6.8 Interaction with Secure Elements

The interaction between the wallet and SE is provided by the CRS applet, as explained in the above paragraphs, to allow the activation, deactivation and list the executable service provider applications.

Operations to install, remove and personalize the interaction are secured using the associated security domain.

Contactless and user-interface parameters are set in the Card Registry and cannot be changed by Java Card application by means of APIs; however, by using RAM protocol it is possible to manage those parameters via OTA.

But as the wallet does not have the keys of the security domain, it is only able to forward the scripts already encrypted. It covers the role of the Companion Application (see §5.8).

Wallet and applications interaction with the SE has always to be allowed by the Access Control configuration of the Secure Element (see § 6.2),

## 6.9 User Interaction

As mentioned the wallet provides the list of active applications and of present applications from the SE with their activated/deactivated state.

### **Multiple Payment Application Management:**

The following requirements apply to UI applications that include functionality that enables the end-user to manage multiple payment applications within the SE.

### **Default Function for Multiple Payment Applications:**

All wallets that include functionality which enables end-users to manage multiple payment applications on the SE must include a function to set a default payment application, which (when activated) will be the payment application that automatically responds when the device is presented to a contactless payment reader to make a payment.

### **One-Time Override of Default Function:**

Wallets that include functionality that enable end-users to manage multiple payment applications on the SE should include a function to override the default payment application for a single payment transaction or a limited time period.

If this functionality is implemented the payment application that was previously set as the default, must revert to being the default when the override conditions have been fulfilled (i.e., once the transaction has completed, when the override time has elapsed or if the end-user has cancelled the override).

### **Payment button:**

Where this feature, or an equivalent proprietary feature, has been implemented, some services providers recommend the use of a 'pay' button as high up the menu structure of the User Interface application as possible. The number of clicks the user has to make in order to pay should be kept to a minimum.

## 7. Security Certifications

### 7.1 Introduction

In order to ensure functionality and security of used elements in the SE environment it may become necessary to offer certified products. A certified product is a product evaluated by a third party which checks the product against the requirements of a customer. However the certification usually has to be supported in terms of time and effort by the product vendor.

Depending on the targeted use case of products different types of certification and different components in the eco system may require certification. Usually only the components holding secrets (e.g. cryptographic keys) or dealing with secrets (cryptographic algorithms) require certification. It may be necessary to certify a whole environment consisting of multiple products, e.g. terminal (hardware + OS), applications residing on the terminal, SE (hardware + OS) and applications residing on the SE.

With respect to the roles defined in the different models in chapter one, it may become challenging to perform an overall certification of all components. It is likely that a terminal and the SE (with its OS) is released and certified but the applications on the terminal and on the SE will be loaded onto the components at a later stage. Therefore, it might not be feasible in terms of timelines and effort to do a full certification of all components again (including the applications) every time a new service (with new applications) is launched.

In such a case an applicable model could be for the applications to be certified as standalone products and require a certified hardware and OS (terminal and SE) to support their execution.

- For eSEs this might be an additional challenge as it is not clear how to handle an outdated certified embedded SE residing on a terminal but having applications on board which are still certified.

### 7.2 Payment System Type Approval Process

Payment schemes (or payment associations) are complex payment infrastructures organised by companies, such as EMVCo, MasterCard and Visa.

In order to assure that all the components (e.g. payment application on the SE, PoS-Terminal, background servers, etc.) of such an infrastructure interoperate reliably, these schemes are mandating a security and functional evaluation of the components from their manufacturer. Depending on the payment scheme in question, the successful evaluation of a product is either called "Type Approval" or 'Certification'.

The Certificate is usually made public and has a certain validity period, which may be renewed on certain conditions.

For volume delivery the Payment Schemes only allow Type Approved Products to be used in their infrastructure in order to keep a high level of security assurance and also functional interoperability.

Exceptions could be made for trials with a small amount of users, where a dedicated 'waiver' is needed from the Payment Scheme (the discussion of which is beyond the scope of this document).

As already stated, a certification can be split into functional and security related components. GlobalPlatform defines only a functional certification.

### 7.2.1 Parties involved in a Type Approval Process

The following instances have to cooperate for a Type Approval Process:

- Vendor - manufacturer of the product to be approved
- Payment scheme - infrastructure provider, such as EMVCo, MasterCard, VISA
- Test laboratory - executes tests on the physical components

### 7.2.2 Roles and Responsibilities

#### Vendor:

- Registers a specific product for certification to the payment scheme
- Supplies evaluation certificates of underlying components as prerequisites
- Provide various forms, product design/implementation information (possibly source code), test samples
- Authorizes laboratory to submit test results to the payment scheme

#### Payment scheme:

- Organisation of the Type Approval Process (from registration to Letter of Approval)
- Accreditation of laboratories
- Owner of specifications
- Definition of test cases
- Review the test-results
- Publishing results

#### Test laboratory

- Tests might be split across more than one laboratory (e.g. for security evaluation and functional testing)
- Check correctness and consistency of all forms before starting the test
- Commit a time line for tests to Payment Schemes and product provider
- Perform Security and/or functional tests according to payment scheme's test plan
- Ask for clarification during tests if necessary
- Provide test results to the involved parties

### 7.2.3 Type Approval Process in general

The following table shows the involved parties and their high-level interaction during the Type Approval Process:

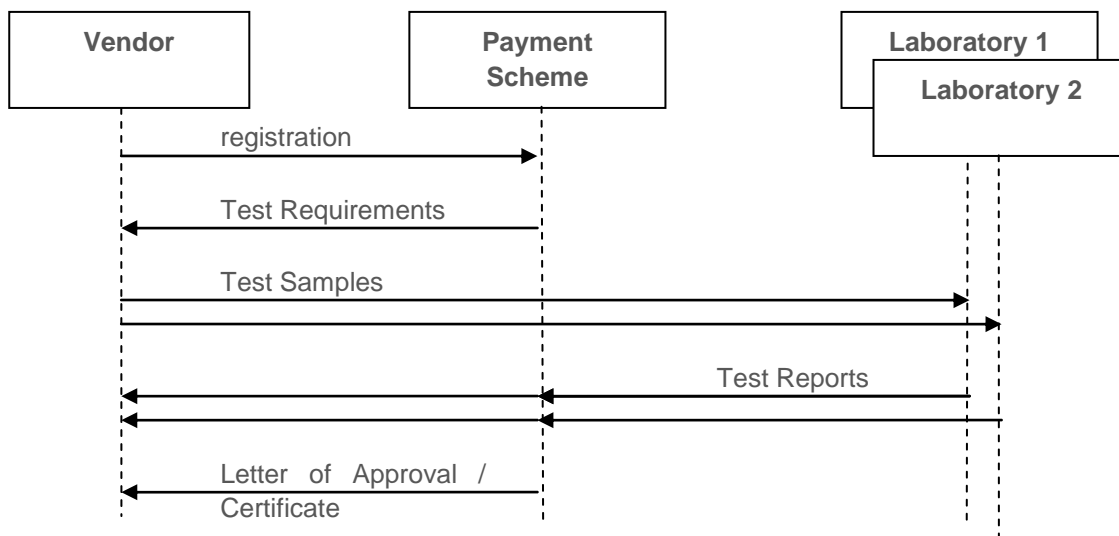


Figure 27: Process for certifications

## 7.3 EMVCo Card Type Approval

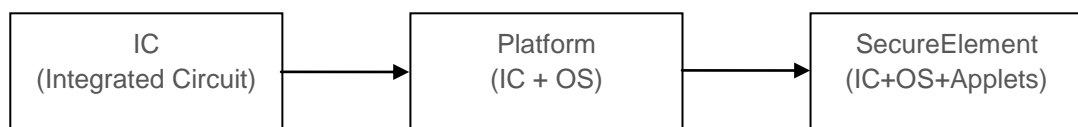
### 7.3.1 Introduction

EMVCo has defined a separate Type Approval Process for the contactless ecosystem, called CMP (Contactless Mobile Payment).

It defines different types of SEs, especially embeddedSE, microSD, and UICC.

### 7.3.2 EMVCo SE components

The base components of a SE are classified into three base-products, since each may be sourced from a variety of different industry players:



Evaluation is necessary for each of those base-products, having individual test requirements:

| Final Product      | EMVCo Evaluation             | EMVCo Certificate  |
|--------------------|------------------------------|--|
| Integrated Circuit | IC Security Evaluation       | IC Compliance Certificate (including Chip Certification Number, CCN) |
| Platform           | Platform Security Evaluation | Platform Compliance Certificate (including Platform                  |

|                               |  | Certification Number, PCN)                      |
|-------------------------------|--|---|
| ICC (integrated circuit card) | Level 2 functional eval.<br>CCD components functional eval.<br>non-CCD components function. eval.<br>ICC Security Evaluation | EMVCo ICC Compliance Cert.<br>(= Type Approval) |

**Note 1:** Steps 3 may also be performed by other Payment Schemes (VISA/Mastercard) instead of EMVCo.

**Note 2:** Here 'IC' is used in terms of an underlying hardware component.

As described in [70], the PPSE application (see below) is not subject to the security evaluation as it does not contain any sensitive data or functionality.

The result of a successful EMVCo testing for any single product will be:

- **EMVCo Compliance Certificate**  
This is the case when all security tests have passed.
- **EMVCo Restricted Compliance Certificate**  
This is the case, when open issues are remaining and the EMVCo certification secretary wants to highlight this state with the 'restrictive' indicator.

The platform product provider needs to have a valid IC Compliance Certificate before starting the EMVCo testing. For ICC Security Evaluation the ICC provider needs to have both, a valid IC Compliance Certificate and Platform Compliance Certificate, in order to start the tests.  
CCN (ChipCertNo.)/PCN (PlatformCertNo.)

### 7.3.3 Components of a EMVCo Secure-Element

A GlobalPlatform compliant SE must contain following components for the Type Approval:

- PPSE and/or SECM (SE Contactless Management, e.g. CRS application) according to EMVCo specifications
- Integrated Circuit that has received an EMVCo Compliance Certificate
- Multi-application framework, that has received a GlobalPlatform Letter of Qualification
- Other applications, e.g. Payment System-specific CMP applications that are not covered by EMVCo Type Approval (e.g. MChip Applet, VMPA Applet)

The Contactless Mobile Payment (CMP) application provides the payment service.  
The Proximity Payment System Environment (PPSE) application is notified each time a CMP application has been activated or deactivated and updates the PPSE response to be returned.

Payment applications of other payment schemes (e.g. VISA, Mastercard) are not part of EMVCo Type Approval and have to be approved by the corresponding payment scheme. EMVCo criteria for certification may be reused by other payment schemes but do not necessarily require EMVCo certification.

#### 7.3.4 Certification Validity and Prolongation

The following certification validity and revocation conditions apply at the current time of writing this document:

- Beginning of Type Approval / Certification: product is listed 3 years at EMVCo
- Extension possible: requires annual security review
- Product will be unlisted at EMVCo:
  - if Certificate is withdrawn
  - or replaced by other product
- When six years listed: automatically removed from list

#### 7.3.5 Impact of Product Changes

Product changes of EMVCo Certified products will require formal documents as follows:

- Security Impact Analysis
- Possibly also 'Delta' Analysis required

When this additional evaluation has passed, a new EMVCo Compliance Certificate is granted.

#### 7.3.6 Security Monitoring

EMVCo is monitoring the threat and security development of the SE market.

If any possible security risk is seen EMVCo will

- Inform product provider of any risk
- Possibly withdraw (restricted) Compliance Certificates

#### 7.3.7 Public Certification Information

The EMVCo certified products are published on the EMVCo website ([www.emvco.com](http://www.emvco.com)).

Note that not all certificates are listed here as each vendor of a certified product may choose whether or not to publicly list it.

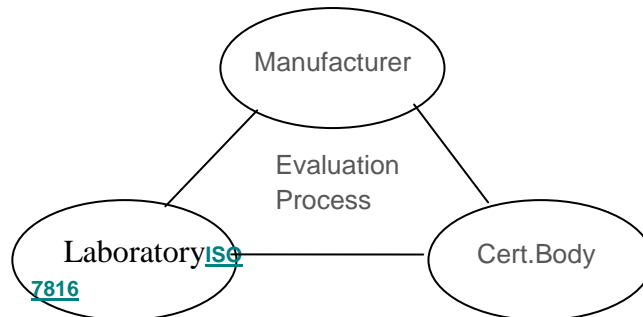
### 7.4 Common Criteria Evaluation

#### 7.4.1 Introduction

Common Criteria (CC, [www.commoncriteriaportal.org](http://www.commoncriteriaportal.org)) describes a standardized method to evaluate security IT systems and products. The main purpose is to examine the Target of Evaluation (TOE) in

respect to the security functional requirements defined in the security target and to conduct a vulnerability analysis.

The main parties involved in the evaluation process are:



- **The manufacturer:** provides the TOE and documentation, such as the security target (ST) and additional evidence.
- **The laboratory** (e.g. t-systems in Germany): is the IT security evaluation facility (ITSEF) which performs the evaluation and reports to sponsor and certifier.
- **The certification body** (e.g. BSI in Germany): supervises the evaluation of the laboratory and publishes a CC certificate and the ST on its website.

A product (TOE) has to be subject to multiple evaluations, including:

- Correctness of security functions
- Level of attack potential
- Used environment
- Used configurations

Multiple inputs (evaluation evidence) can also be used to evaluate a certain product:

- Design specifications
- Development environment
- Test results
- Code review
- Security Architecture (Vulnerability analysis)
- TOE Design (Software and Hardware)
- Development site security audits
- SPA/DPA/DFI analysis reports

### 7.4.2 CC Assurance Levels

Common Criteria has a number of different assurance classes that can be chosen from the CC specification (Part 3). To make it easier for the customer and developer CC has introduced several Evaluation Assurance Levels (EAL) which reflect upon others certain test levels with different predefined security levels, ranging from simple tests (EAL1) up to very detailed tests (EAL7). For SEs, usually EAL4+ is used as an EAL that contains all required tests of EAL4 but adds (“+”) some dedicated assurance classes from higher EALs that can be applied SEs. For example, AVA.VAN.5 contains a very rigorous vulnerability analysis. From EAL 4 (and higher) the implementation source code of the TOE must be presented to the laboratory.

The following describes the test aspects of the different EALs:

- EAL 1 Functionally tested
- EAL 2 Structurally tested
- EAL 3 Methodically tested and checked
- EAL 4 Methodically designed, tested and reviewed
- EAL 5 Semi-formally designed and tested
- EAL 6 Semi-formally verified, designed and tested
- EAL 7 Formally verified designed and tested

### 7.4.3 Protection Profile

A Protection Profile (PP) defines an implementation independent set of IT security requirements for a category of products which are intended to meet common consumer needs for IT security. The development and certification of a PP, or the reference to an existent PP, gives consumers the possibility to express their IT security needs without referring to a special product.

A Protection Profile covers following items:

- List of threats
- List of functional requirements
- List of assurance activities
- Justification that these activities address the threat

The Protection Profiles are public and can be downloaded e.g. from [www.commoncriteriaportal.org](http://www.commoncriteriaportal.org). Furthermore most Protection Profiles are evaluated itself for compliancy according to Common-Criteria, this is done by the certification body (e.g. BSI in Germany).

In case of a SE CC-Evaluation, a combination of different Protection Profiles can be used as a basis for the development of the Security-Target specification that reflects the specific security requirements of the real product (TOE).

The following public PPs ([www.commoncriteriaportal.org](http://www.commoncriteriaportal.org)) might be considered for the Security Target of a SE in the telco area:

- Java Card™ System Protection Profile Open (+closed) Configuration, Version 2.6
- (U)SIM Java Card Platform Protection Profile / Basic and SCWS Configuration



Guidelines for the creation of a Security Target are available from GlobalPlatform:

- Smart Card Security Target Guidelines v1.0  
(<http://www.globalplatform.org/specificationscard.asp>)

Depending on the evaluation model used, also dependencies between different components can be targeted by the evaluation, e.g. a chip module and the production and/or personalization environment of the chip module. Therefore the transition of security requirements may be necessary. For example if component A depends on component B then component A must at least have the same requirements as component B or higher. Otherwise also the Component Bs security can be compromised.

#### 7.4.4 Security features

When it comes to evaluation of security features such as cryptographic operations two main security functional requirement (SFR) classes of evaluation (CC Part2) are to be taken into account:

- Cryptographic key management
- Cryptographic operation

#### 7.4.5 Cryptographic key management

This part of the evaluation focusses on the lifecycle of keys used in the TOE, from their generation to their destruction. The CC described four activities in the life cycle of keys:

- Key Generation
  - Generation during production
  - Input from other entities
  - Internal (disclosed) generation (e.g. inside-hardware PKI key generation)
- Key distribution
- Key access (key usage)
- Key destruction

#### 7.4.6 Cryptographic operations

This part of the evaluation takes focus on the used algorithms together with specified key attributes (etc. key length). Typical usage:

- Encryption/decryption (data and keys)
- Signature generation/verification
- Cryptographic checksum generation/verification

The signature generation/verification and cryptographic checksum generation/verification are essential to ensuring that the TOE includes replay detection mechanisms in order to bypass such attacks.

When using external hardware for cryptographic operations (e.g. crypto co-processor) the same evaluation rules apply as for the TOE. Additionally confidentiality must be maintained when transporting data from the TOE to such external hardware or vice versa (e.g. encrypted transport, integrity check).

External hardware for cryptographic operations are also be prone to attacks described in the chapter below.

#### 7.4.7 Hardware related evaluation

In order to successfully execute software evaluation, the underlying hardware itself must also fulfill the requirements of the software. Therefore two main constraints must be applied:

- Access security (i.e .unauthorized access to (read/write) data, manipulation of processed data)
- Data integrity (persistent data shall not be manipulated, resp. manipulation shall be detected and may be counter-measured)

To fulfill these requirements, violations must be successfully detected and counter-measured by appropriate mechanism in order to:

- Guard against the revelation of secrets
- Maintain data integrity

An evaluation may also focus on separating different modules of the hardware, evaluating each of separately in order to identify possible areas of weakness. An example here could be if the cache of a CPU is not directly built into the semi-conductor and is also not protected by the hardware that surrounds and protects the CPU.

Additionally, the transport mechanism of data inside hardware can be evaluated as well, since it can contain also contain secret data (e.g. using crypto co-processor).

#### 7.4.8 Hardware attack scenarios and countermeasures

This section describes security threats that are considered in the Common Criteria supporting documentation (<http://www.commoncriteriaportal.org/files/supdocs/CCDB-2009-03-001.pdf>):

#### DPA/SPA

Differential Power Analysis (DPA) and Simple Power Analysis (SPA) are designed to extract secret data from semiconductors by analysing power consumption during operations.

Countermeasures include:

- Balancing power consumption to be independent of used secrets (by software or hardware)
- Updating secret data in certain intervals
- Adding noise into the power consumption
- Applying random execution of commands
- Keeping algorithms secret

## DFA

Differential Fault Analysis generates hardware faults during the execution of an algorithm (e.g. high temperature, over clocking etc.) and calculates secret data by comparing correct and incorrect output data.

Countermeasures include:

- Multiple calculations
- Checksum of result (better timings e.g. for RSA than doing the calculation multiple times)

## Timing attacks

This is the measurement of algorithmic execution times, since they are data dependent.

Countermeasures include random execution by time through careful software design. As the hardware platforms present different timing of executions, usually countermeasures have to be tailored on the platform itself.

## Lightning attacks

This is the manipulation of data or code segments by sending focused energy to the hardware (alpha-rays or ultraviolet rays), in order evoke a skip in a security check (skip if-clause etc.)

Countermeasures include:

- Hardware sensors to detect such attacks and stop operation on detection
- Available for almost all semi-conductors used in SE environment

### 7.4.9 Validity of CC Certificates

Currently CC certificates may have different validities, depending on the issuing national certification body.

Certificates with lifetime validity do exist. Here, the vendor must inform the certification body if a security issue was found. This will result in the certificate being revoked.

### 7.4.10 Links to Certificates and PPs

Please visit the following link to review products that are certified by Common-Criteria:

<http://www.commoncriteriaportal.org/products/>

The following link shows publicly available CC Protection-Profiles:

<http://www.commoncriteriaportal.org/pps/>