



CARDLET DEVELOPMENT GUIDELINES

RELEASE 2.4

Date 24/06/2014

Reference afscm-nfc-cardlet-development-guidelines-v2.4-20140624

	Name	Company
Authors	Pierre Le Pallec, Sophie Diallo	Bouygues Telecom
	Thierry Simon	EI Telecom
	Ahmad Saif, Olivier Briot, Patrick Picard, Laurent Fourreau, Sylvie Camus	Orange
	Michael Bensimon, Jérôme Devisme, Marilyne Eznack, Saïd Benlaadam	SFR
Editor	Grégoire Fraisse, Laurent Kisielewski	AFSCM

Document management

Version	Date	Chapters	Modification
1.0	30/06/2009	All	First release
1.1	20/10/2010	2.3.1.1 3.3	Addition of the note Concerning new parameters due to use of amendment C GP22 New (U)SAT Event List
2.0 DRAFT	19/07/2011	1.4 References Appendix 1	Addition of external references that must be taken into account as a development requirement along with the present document. Addition of a cross-check table concerning the EAL4+ PPUSIM requirements applicable to basic applications.
2.0	15/09/2011	1.4 References	Updated Gemalto reference documentation
2.1	21/12/2011	All	- Rules are now numbered - Information and free usage have been removed - Added / modified rules in compliance with UICC high level requirements V2.0 (details about new/modified/kept rules can be found in section 11) - References to platform-specific requirements have been removed
2.2	24/02/2012	All	Modified - rule 3: precised "for applications in different SD" - rule 63: added value 0x90 - rules 18 & 19: added "for basic application" to some forbidden API - rules 12, 28, 50 and 51, 55 removed Checked and precised all rules to ensure a clear understanding of what is mandatory (MUST/SHALL) and what is recommended (SHOULD) CRS section moved to section 2.5, CREL section added.
2.3	08/10/2012	All	Propositions: take into account some Cards manufacturers requirements due to the Verification Authority constraints
2.4	24/06/2014	All	- Clarifications, updates in §1 (e.g. introduction, references, definitions and document bookmarks) and corrections to improve the document (with no impact on the technical content) - Throughout the document: homogenization of terms

Version	Date	Chapters	Modification
			“shall”, “should” and “may” • §2.1: rule 0 added regarding documents to be provided; rules 1, 2 and 4 moved to §2.2 (SoA); rule 3 moved to §3.1 (JavaCard API) • §2.2: rule 7 divided into 2 rules (7a and 7b), rule 8 generalized to all Cardlets • §2.2.1-2.2.2: package installation and lock/unlock: add of a note regarding setting of self-activation privilege • §2.2.4: part on command sequences completed • §2.3: rules 9, 10 and 11 aligned with GlobalPlatform • §2.4.1: rules 13 and 14 changed as mandatory and clarified (integration of former rule 50); rule 66 added on self-activation privilege; rules 67-68-69 added on MSL, priority levels and access domain. • §2.4.2: rule 15 completed, rule 16 removed • §3.1: rule 18 partially modified, with ban of MultiSelectable methods, authorization of OwnerPIN methods and integration of former rule 51 • §3.2: rule 19 partially modified, clarified on CVM part (integration of former rule 53) and on exceptions for use of getGPCLRegistryEntry method • §3.3: rule 20 more restrictive ; rules 21 and 22 completed with add of new limitations on commands or events • §4.1: rule 24 completed • §4.2: rule 29 removed (redundant with rule 36) • §4.3 : rule 31 clarified • §6.1: rule 33 removed and integrated into rule 34 • §6.2: rule 36 clarified (integration of former rule 29), rule 40 removed (redundant with rules 20 and 34), rule 41 removed and integrated into rule 35. • §7: rules 44 reworded, rule 45 completed • §9: rule 50 removed (redundant with rules 13 and 14); rule 51 removed (redundant with rule 18) • §10: rule 53 removed (redundant with rule 19); rule 56 removed (deprecated); rule 57 changed as a recommendation; rule 70 added regarding SetIncomingAndReceive method processing; rule 59 removed (redundant with rule 22); rule 61 removed (as per GSMA), rule 65 removed • §11: synthesis updated with a new categorization of rules

TABLE DES MATIERES

1	INTRODUCTION	5
1.1	Context	5
1.2	Copyright license	5
1.3	Purpose and content of this document.....	7
1.4	References.....	8
1.5	Abbreviations	10
1.6	Definitions	10
2	MANAGEMENT RULES	12
2.1	Documentation.....	12
2.2	State of the art	12
2.2.1	Application Package Installation.....	13
2.2.2	Application lock / unlock	13
2.2.3	Application Instance / Package Removal.....	13
2.2.4	Command sequences.....	14
2.3	Development environment	14
2.4	Management life cycle	15
2.4.1	Cardlet installation.....	15
2.4.2	Cardlet Deletion.....	17
2.5	Contactless Registry Service (CRS)	17
2.6	Contactless Registry Event Listener (CREL)	18
3	USAGE OF APIs AND SIM TOOLKIT COMMANDS AND EVENTS	19
3.1	JAVACARD.....	19
3.2	GlobalPlatform	20
3.3	UICC.....	21
3.4	SIM.....	22
3.5	NFC API	22
4	MEMORY MANAGEMENT	23
4.1	Basic recommendations	23
4.2	Memory Protection	23
4.3	Memory allocation/de-allocation (RAM, Flash, EEPROM)	23
5	HANDSET RESOURCES MANAGEMENT	24
6	SYSTEM MANAGEMENT	25
6.1	File System Access	25
6.2	File Event and Management	26
7	EXCEPTION AND ERROR MANAGEMENT	28
8	ROBUSTNESS AND INTEROPERABILITY	28
8.1	General recommendations.....	28
9	INTERACTIONS.....	29
9.1	Handset (STK, USB, ...)	29
10	DEVELOPMENTS RULES	30
11	SYNTHESIS OF RULES	32

1 INTRODUCTION

1.1 Context

The Mobile Network Operators Bouygues Telecom, Orange and SFR have founded the AFSCM (Association Française du Sans Contact Mobile) to foster the development of mobile contactless services.

The AFSCM constituency includes all companies involved in the development of a sustainable market for mobile contactless services such as Service Providers, handset makers, smart card manufacturers, Mobile Network Operators, MVNOs.

The AFSCM members include mobile telephony operators, contactless service providers, manufacturers and technical service providers.

The AFSCM objective is to support the inception of new contactless services for mobile phone users. In particular, the AFSCM endeavours:

- to support service providers in the definition and deployment of contactless solutions suited for mobile subscribers to any available mobile network;
- to specify technical guidelines for the development of mobile contactless services to ensure seamless installation and consistent user experience;
- to promote the benefits of the mobile phone platform for contactless service providers, technology providers and end users.

To define a mutually beneficial mobile contactless ecosystem, the AFSCM proposes a shared mobile contactless target mark and a shared brand that distinguishes those contactless services that are available to mobile phone users.

Together, the AFSCM members contribute to the definition of innovative industry standards and best practices.

The stated objective of the AFSCM is to facilitate the development of mobile contactless services.

To this end, the founding members recognize the significance of the following success factors:

- virtuous ecosystem in which all parties involved can develop a sustainable market position,
- efficient customer support,
- smooth customer experience,
- state-of-the art application life cycle management,
- service portability in the event of a mobile equipment swap.

1.2 Copyright license

The following document is a personal, non-exclusive copyright license between you – the Licensee - and the AFSCM founding members (*), regarding the AFSCM specifications, that must be included in any copy of said specifications.

The licensee is authorized to copy, present or communicate the AFSCM specifications on any media without having to pay any license fee under the condition that the following copyright notice be included in any copy or any excerpt of the specifications

« Copyright © AFSCM 2008-2014 (Association Française du Sans Contact Mobile). All rights reserved. Terms and conditions to copy, display and communicate these specifications are available at <http://www.afscm.org/> »

The licensee is NOT authorized to create or disclose modifications or abstracts of the specifications or work derived from the specifications.

The specifications include detailed functional specifications, technical specifications, NFC handset and NFC UICC implementation guidelines, application development guidelines (Mobile Application and Cardlet) and SP-MNO interconnection guidelines.

Separate patent licenses and additional materials will be proposed to those wanting to implement solutions compliant with the AFSCM specifications, under licensing conditions to be defined in separate agreements. Information for procuring such patent licenses and additional documents will be made available upon request by e-mail to: contact@afscm.org.

THE SPECIFICATIONS ARE SUPPLIED "AS IS" AND NEITHER THE AFSCM NOR ITS MEMBERS ARE COMMITTED TO ANY WARRANTY, EXPLICIT OR IMPLICIT, REGARDING THE SPECIFICATIONS, IN PARTICULAR NO WARRANTY IS GIVEN OF QUALITY, SUITABILITY TO ANY USE WHATSOEVER, ABSENCE OF TITLE OR RIGHTS TO THE CONTENT OF THE SPECIFICATIONS, INSURANCE THAT THE USE OF THE SPECIFICATIONS WILL NOT INFRINGE INTELLECTUAL PROPERTY RIGHTS OF A THIRD PARTY SUCH AS PATENTS, TRADE MARKS, COPYRIGHTS. NEITHER THE AFSCM NOR ITS MEMBERS SHALL BE HELD LIABLE FOR ANY DAMAGE INCURRED IN CONNECTION TO THE USE, REPRESENTATION OR COMMUNICATION OF THE SPECIFICATIONS.

Neither the AFSCM name nor its trade marks shall be used under any circumstances, such as to communicate or advertise the specifications without the prior written agreement of the AFSCM.

No rights other than the rights described above are granted under this license and the rights granted under this license cannot be construed as conferring, implicitly or explicitly, any rights (through a licensing agreement or any other means) concerning AFSCM or AFSCM members inventions, know-how or intellectual property, or any of their assets resulting from, based on or required in the specifications.

This copyright license is subject to French law and shall be governed by and interpreted according to French laws. The exclusive place of jurisdiction shall be the Paris Court of Appeal, regardless of the number of claims or defendants.

By downloading the AFSCM specifications, you indicate your acceptance of these terms and conditions.

() AFSCM founding members are Bouygues Telecom, Orange France / France Telecom, SFR.*

1.3 Purpose and content of this document

The purpose of this document is to provide to Service Providers some rules, common to the Mobile Network Operators, in order to properly develop the Cardlet(s) of their mobile NFC service.

In order to guarantee a management of minimal risks, **this document formulates a set of rules split in three different categories:**

- rules related to the state-of-the-art in Applications Java Card programming,
- functional/configuration rules, helpful for best operational working order,
- security rules.

There are two types of applications: **sensitive and basic** ones:

- The sensitive applications handle sensitive assets which must be protected.
- In all other cases, the applications will be basic.

In the present document, and regarding **sensitive applications** specifically: as they must be certified on certified platforms, only functional and configuration rules shall be applied. Indeed, the sensitive applications developers will apply security and state-of-the-art rules implicitly. They already will have to follow the platform security guidelines of the Card Manufacturer to successfully pass the evaluation process done by the CESTI Lab. However, when results of certification of a sensitive application cannot be provided at the time of the AFSCM evaluation, a **waiver** can be delivered by the AFSCM Validation Committee, **provided that security rules of the present document are applied**. The conditions of this waiver will be determined by each MNO.

As for **basic applications**, they shall follow the complete set of rules defined in the present document, before being submitted to AFSCM validation process.

The present document is divided in the following sections:

- **Chapter 2** Management rules

This chapter gives recommendations regarding:

- The structure of the application and the usage by the application of the data, the resources or the sensitive functions used by the application,
- The development environment,
- The management of the application on the UICC (installation and deletion), including the description of the GlobalPlatform commands to be used.

- **Chapter 3** Usage of APIs and SIM Toolkit Commands and Events

This chapter lists the APIs and the SIM Toolkit commands and events that are either "Limited" or "Forbidden". When an API, command or event is listed as "Limited", it means that it is sensitive and its usage by the application shall be described in the Functional Specification and in the Acknowledgment to Security Requirements.

- **Chapter 4** Memory management

This chapter gives recommendations regarding memory management (deletion, allocation...) as it is a limited resource on the UICC.

- **Chapter 5** Handset Resources management

This chapter presents in a table the restrictions regarding the usage of the mobile handset resources (for instance when the end user validation is necessary).

- **Chapter 6** System Management

This chapter describes the actions on files (access types, creation). By default, access to files (reading, writing) and creation of files (excepting application files) are forbidden; files authorized in restricted access are listed.

This section also introduces the concept of Contactless Registry Service proposed by GlobalPlatform.

- **Chapter 7** Exception and error management
This chapter gives recommendations regarding the exception and error management.
- **Chapter 8** Robustness and interoperability
This chapter gives general recommendations in order to ensure robustness and interoperability of the application, including in case of RF field disruption (tearing).
- **Chapter 9** Interactions
This chapter gives recommendations regarding the interactions of the Cardlet with the mobile and the external resources.
- **Chapter 10** Developments rules
This last chapter presents, or reminds, recommendations regarding the development of a Cardlet.

Throughout the document, the key points are highlighted with the following icons:

-  **Mandatory or recommended good practice**
-  **Forbidden**
-  **Warning**

In the rules defined in this document:

- the term “**shall**” denotes mandatory requirements;
- the term “**should**” denotes recommendations;
- the term “**may**” denotes optional.

Note: similar document is available regarding the Mobile Application [R3].

1.4 References

All standard reference(s) can be found in the following list:

AFSCM

- [R1] Interface Specification Between Telecom Operators and NFC Service Providers (as found under “Interface Specifications” on AFSCM website)
- [R2] NFC Applications Validation Process, Derogation form and Delivery of the Publisher and Acknowledgment to Security Requirements (as found under “NFC Application Validation Process” on AFSCM website)
- [R3] Mobile Application Development Guidelines for the different supported OS (as found under “NFC Application Development Guidelines” on AFSCM website)
- [R4] Guidelines for Provisioning and Interconnection of Service Provider and MNO Information System (as found under “Provisioning and Interconnection Guidelines” on AFSCM website)
- [R5] NFC Mobile Handset High Level Requirements (as found under “NFC Mobile Handset and UICC Requirements” on AFSCM website)

- [R6] NFC UICC High Level Requirements (as found under “NFC Mobile Handset and UICC Requirements” on AFSCM website)

SUN Java Card specifications

- [R7] Java Card 2.2.1 API Specification
- [R8] Java Card 2.2.1 Runtime Environment Specification
- [R9] Java Card 2.2.1 VM Architecture Specification
- [R10] Code Conventions for the Java Programming Language, Sun Microsystems, 1999. Available on <http://www.oracle.com/technetwork/java/codeconv-138413.html>.
- [R11] Java Card Platform, v3.0.1 Classical Edition

Development guidelines and good practices references

- [R12] Writing Robust Java Code, by Scott W. Ambler, 2000. Available on <http://www.ambysoft.com/javaCodingStandards.pdf>.
- [R13] The Elements of Java Style, by Vermeulen, Ambler, Bumgardner, Metz, Misfeldt, Shur, & Thompson, Cambridge University Press, 2000.

GlobalPlatform specifications

- [R14] GlobalPlatform Card Specification 2.2.1

JSR Specifications

- [R15] JCP – JSR-000118: Mobile Information device Profile -Trusted MIDlet suites using X509 PKI
- [R16] JCP – JSR-000177: Security and Trust Services API for J2ME
- [R17] JCP – JSR-000257: Contactless Communication API

FFT

- [R18] (U)SIM Java Card Platform Protection Profile, ref PU-2009-RT-79, v2.0.2 (ANSSI certified)

SCP specifications

To adhere with current technology and associated specifications, we recommend that NFC application developers comply with ETSI Java Card API specifications as described in Release 6. The exception is ETSI TS 102.705 (HCI API), where developers should comply with ETSI Java Card API specifications Release 9 and Javacard 3.0. Adherence to these specifications will ensure interoperability with SIM cards of different MNOs and will also ensure backward compatibility. If a NFC application has to rely on a version of ETSI Java Card API which is higher than Release 6, then this should be verified with concerned MNOs.

- [R19] ETSI TS 102 223: Smart Cards; Card Application Toolkit (CAT)
- [R20] ETSI TS 102 225: Smart Cards; Secured packet structure for UICC based applications
- [R21] ETSI TS 102 226: Smart Cards; Remote APDU structure for UICC based applications
- [R22] ETSI TS 102 241: Smart Cards; UICC Application Programming Interface (UICC API) for Java Card™
- [R23] ETSI TS 102 705 (Rel9): Smart Cards: UICC Application Programming Interface for Java Card™ for Contactless Applications

3GPP specifications

- [R24] 3GPP TS 31.102 (ou ETSI TS 131 102) Characteristics of the USIM Application

This document takes into account the requirements of UICC manufacturers guidelines for basic applications development.

1.5 Abbreviations

AFSCM	Association Française du Sans Contact Mobile
APDU	Application Protocol Data Unit
API	Application Programming Interface
CAT	Card Application Toolkit
CREL	Contactless Registry Event Listener
CRS	Contactless Registry Service
EAL4+	Evaluation Assurance Level, refer to Common Criteria standards
GP	GlobalPlatform
JSR	Java Specification Request
MNO	Mobile Network Operator
NFC	Near Field Communication
OTA	Over The Air
PPUSIM	Protection Profile of USIM, defined by the FFT
RAM	Remote Access Management
RID	Registered application provider IDentifier
RMI	Remote Management Interface
SD	Security Domain
SE	Secure Element
SIM	Subscriber Identity Module
SP	Service Provider
TOE	Target Of Evaluation
UICC	Universal Integrated Circuit Card (aka SIM card)
USAT	Universal SIM Application Toolkit
USIM	Universal SIM

1.6 Definitions

Acknowledgment to Security Requirements	(also known as « Déclaration Spécifique de Sécurité ») Declaration document the Service Provider has to provide. It lists the used resources, the system accesses, used libraries, etc. of the NFC application. It is fully described in [R2].
Basic application	A Java Card application that: • does not have a negative impact on the certification status of the product, • should have been verified to meet a set of basic security rules, • and therefore may reside on a card along with one or more Sensitive Applications.

	A basic application shall be validated by a lab to verify the harmlessness of this application regarding the others (basic or sensitive) already existing on the platform.
Cardlet	JavaCard application loaded onto the UICC ((also known as UICC application or applet)
Functional Specification	Document describing all the functionalities of a service. It details which resources are used and how the application accesses to these resources (API, JSR, methods...). A specification is provided by the Service Provider for each application.
Mobile Application	Application loaded onto the mobile handset, providing a GUI (also known as MMI or midlet)
Sensitive application	A Java Card application that requires formal Security Certification by an Evaluation Scheme, such as Common Criteria or EMVCo. A sensitive application shall be certified on a certified platform.
Service or NFC Service	Combination of a UICC application and a MMI (also known as Mobile NFC service)
Service Provider	Entity which provides a NFC service.
UICC	Universal Integrated Circuit Card (usually known as SIM card). In the context of the NFC deployment, the UICC is certified at an EAL4+ level. This implies a number of specific development constraints linked with the UICC platform.
Validation Entity	Entity in charge of validating that the security level in the applications meets the security requirements (typically, a laboratory).

2 MANAGEMENT RULES

2.1 Documentation

✓ **Rule 0: Functional Specification and Acknowledgment to Security Requirements**

As presented in the NFC Application Validation Process [R2], the Service Provider shall provide to the Validation Entity:

- The Functional Specification of its application. This specification identifies the resources used by the application and describes the security measures organized to protect the resources of the Application.
- An “Acknowledgment to Security Requirements” that exhaustively details the use of the data, the resources or the sensitive functions by the application.

Chapter 3 of this document lists the APIs and the SIM Toolkit commands and events that are "Limited" or "Forbidden". When an API, command or event is listed as "Limited", it means that it is sensitive and its usage by the application shall be described in the Functional Specification and in the Acknowledgment to Security Requirements.

The document [R2] presents the purpose of the Acknowledgment to Security Requirements.

2.2 State of the art

This document presents mandatory or recommended complementary rules to be taken into account for developing Java applets; however applications developers should also comply with state of the art development practices.

✓ **Rule 1: Interface management and Data management** shall be clearly separated.

Separate the handling of commands and the handling of content. The essential separation is between the interface and the actual processing of application data. This is particularly important when several interfaces are available, for instance an APDU interface and a SIM Toolkit interface.

✓ **Rule 2: The interactions between different interfaces** shall be clearly defined.

When several interfaces are used, their respective roles need to be clearly defined, and this needs to be reflected in the application code.

✗ **Rule 4: Package design:** A basic application shall never be in the same package as any certified applet.

The laboratory will at least check that the basic application AID RID (Registered application provider Identifier) is not in the list of AID RID of API and certified applets. This list is to be retrieved from MNOs.

✓ **Rule 5: Entry points:** All the different entry points shall be clearly defined at, for instance: APDU level, Toolkit level, OTA script level, Personalization level, Shared methods level.

Moreover, a basic toolkit Applet shall verify that each user entry is in the range of accepted values.

- ⚠ Rule 6: Denial of Service:** developers shall avoid practices leading to any denial of service as, for instance:
- banning any Instantiation other than install() or applet's constructor,
 - limitation of use of data heap resources,
 - limitation of use of telecom file system resources ,
 - banning infinite loop,
 - limiting multiple writing in non-volatile memory to avoid memory stress.
- ❌ Rule 7a: Dead code** shall be deleted / removed from the code.
- ❌ Rule 7b: Debug information** shall be deleted / removed from the code.
- ❌ Rule 8: Coding NOP:** An application bytecode shall not include any NOP (No Operation) instruction.

2.2.1 Application Package Installation

The Load and Installation functions shall be based on the Remote Application Manager (RAM) mechanisms as defined in [R21]. The RAM mechanisms are based on the referenced GlobalPlatform Specifications in [R14].

The following GP commands are necessary to load an application package:

- INSTALL [for load]
- LOAD
- INSTALL [for install]

These commands shall be implemented as defined in the AFSCM guide [R6].

Note: During installation stage, if instantiation and activation are performed using 2 different commands (i.e. INSTALL [for install] then INSTALL [for make selectable]), the GP Contactless Self-Activation privilege has to be set.

*This setting is **mandatory** (refer to Rule 66 in part 2.4.1), due to the implementation of some SIM cards among those actually distributed – this statement applies for example to SIM cards implementing the version 1.0 of GP Amendment C.*

2.2.2 Application lock / unlock

For multiple reasons, notably in case of mobile recovery after a loss or theft, Service Provider, after having locked the application initially, may want the end user to be granted to use his NFC Service anew. The Service Provider will thus initiate an unlock process.

In order to be active and selectable on the contactless interface after such a lock / unlock operation, contactless Cardlets shall implement the self-activation privilege, as requested in Rule 66 (2.4.1).

2.2.3 Application Instance / Package Removal

The goal is to remove an application package from a UICC. This process is designed to perform the deletion of the instance of the application package and the executable load file of the application package.

The GlobalPlatform command DELETE, as defined in Specifications [R14], is used to remove an application package.

2.2.4 Command sequences

An example of command sequences is reminded below:

- **Loading**
 - *INSTALL [for load]*
 - *LOAD*
- **Instantiation**
 - *INSTALL [for install]*
- **Extradition**, if the package was not already load in the target SD
 - *INSTALL [for extradition]*
- **Activation**
 - *INSTALL [for make selectable]*
- **Personalization**
 - *INSTALL [for personalization]*
 - *STORE DATA*
- **Deletion**
 - *DELETE*

Some commands may also be grouped, as shown in the example below:

- **Loading**
 - *INSTALL [for load]*
 - *LOAD*
- **Instantiation and activation**
 - *INSTALL [for install and make selectable]*
- **Extradition**, if the package was not already load in the target SD
 - *INSTALL [for extradition]*
- **Personalization**
 - *INSTALL [for personalization]*
 - *STORE DATA*
- **Deletion**
 - *DELETE*

All kinds of sequences and command orders which shall be supported are described in the Interface Specifications [R1]: unitary or grouped commands, personalisation before or after the activation, etc.

2.3 Development environment

The development environment shall be physically and logically separated from the production environment.

Access to the source code shall be protected and reserved to duly authorized persons.

A Software Configuration Manager (SCM) that allows drawing all the modifications made on the application should be available.



Rule 9: Versioning

Basic Applications will be validated for the given version numbers; changes to the shared libraries may introduce incompatibility and security issues. So, when the Basic Application provides a shared library, the versioning policy shall be:

- at least the minor version changes if there are changes to the implementation of the exported methods;
- if the signature of a method changes, the major version shall be incremented.



Rule 10: ByteCode verification

A Basic Application shall be strictly in accordance with the Java Card specifications and shall **successfully pass** bytecode verification, using tools from Oracle or from the platform issuer of the target platform. The tools used for bytecode verifications shall be the latest version available.



Rule 11: Shared resources

Basic Applications will be validated for the given version numbers; using older versions of the shared libraries may introduce incompatibility and security issues. So, if the Basic Application uses shared libraries:

- the major version of the shared library found on the target platform shall be equal to the major version number of the library that has been used for verification of the Basic Application ;
- the minor version of the shared library found on the target platform shall be equal to or higher than the minor version number of the library that has been used for verification of the Basic Application.

2.4 Management life cycle

The life cycle of the applications shall be correctly managed: the execution of an action should be possible only in the state of the application to which it relates. The state of an application shall be protected as a sensitive data and thus some measures to protect its integrity shall be taken.

2.4.1 Cardlet installation



Rule 12: Checking of Cardlet installation parameters

The following tests methods shall be used so as to verify the « Cardlet installation parameters »:

For a pre-installed applet, check the AID instance presence using the “Get status” command. The usage of proprietary commands is forbidden for this feature.

For other applets, check the Status code returned from the install for install command.



Rule 13: Applet instance registration shall occur as late as possible in the installation phase.

For applets without SIM Toolkit registration: registration should be the last operation of the installation phase.

For SIM Toolkit applets: toolkit event registration shall occur after the applet registration, but no other operation shall occur, and in particular no object allocation. The only API used after registration shall be related to the management of the toolkit registry.



Rule 14: STK menu entries initialization

STK / USAT menu entries shall be initialized at the end of install() method. This is an extension of

the previous guideline for events implicitly registered, for instance with method `ToolkitRegistry.initMenuEntry`.

✓ **Rule 66: GP Contactless Self-Activation privilege:** Cardlets shall be set to support the GP Contactless Self-Activation privilege.

This implementation makes sense if the NFC Service customer experience expects, at the end of the installation process or after a lock / unlock procedure:

- either a first Mobile Application – Cardlet interaction, including the application selection on ISO interface, prior to any card emulation transaction;
- or an APDU command sent OTA, in order to request the Cardlet to activate its contactless state by itself.

It is recommended that the SP retrieves the contactless state of the Cardlet at first, prior to setting the self-activation privilege:

1. `GPCLSystem.getGPCLRegistryEntry(null).getCLState()`
2. `GPCLSystem.getGPCLRegistryEntry(null).setCLState(ctlessState)`

Note: in accordance with Rule 19, use of all other parameter values in method “`getGPCLRegistryEntry`” is forbidden.

✓ **Rule 67: Minimum Security Level (MSL)**

The Minimum Security Level (MSL) is used to specify the minimum level of security to be applied to Secured Packets sent to any Receiving Application (refer to [R20]).

Only the following values of the SPI byte, given in the table here under, are allowed:

b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	1	0	0	1	0
0	0	0	1	0	1	1	0
0	0	0	1	1	0	1	0
0	0	0	1	1	1	1	0

Hexa
0x12
0x16
0x1A
0x1E

1 0 : Cryptographic Checksum

0 : No cyphering

1 : Cyphering

10: Process if and only if counter value is higher than the value in the RE (note 3)

11: Process if and only if counter value is one higher than the value in the RE (note 4)

Reserved (set to zero and ignored by RE)

SPI byte format

The counter value is compared with the counter value of the last received Command Packet. This is tolerant to failures on the transport level (i.e. losses of Command Packets). A possible scenario is a global update.

This provides strict control in addition to security indicated above.

✓ **Rule 68: Priority Level**

The Priority Level shall be set to 0xFF (the lowest priority level).

All use of another Priority Level parameter value shall be duly justified.

✓ **Rule 69: Access Domain**

The Access Domain is a parameter used to define the access rights granted to a Cardlet, allowing it to perform operations on UICC files.

The Access Domain parameter shall never be set to 0x00 (full access). The value shall be set to 0xFF (no access).

All use of another Access Domain parameter value shall be duly justified.

2.4.2 Cardlet Deletion

✓ **Rule 15: Implementation of the uninstall method** in applets conforming to [R11] is mandatory.

[R11] defines the AppletEvent interface, which includes an uninstall method that is invoked when the deletion of an applet instance is requested; this method shall be implemented by all applications and used to set all static object references to 'null'.

✓ **Rule 17: Cardlet deletion phase:** All created files shall be deleted in the application deletion phase.

The USIM API provides the ability to create files, but also to delete them. During the application deletion phase (method AppletEvent.uninstall), an application that has created some files shall delete them all, leaving the SIM card in its original state.

2.5 Contactless Registry Service (CRS)

The Contactless Registry Service defined in [R14] Amendment C provides a mechanism for a User to get access to a list of CRS registered Applications on the card and to activate or deactivate these Applications. It also provides a mechanism to authenticate the end user for accessing the list of CRS registered Applications and to authenticate the end user prior to transition their Life Cycle state to ACTIVATED or DEACTIVATED and to notify on-card Utility (or other) Applications of the Life Cycle transition.

The CRS may be used by any Application on the card to register its parameters such as the information to be displayed to the end user to identify the Application, the utility Application(s) to be notified when the Application Life Cycle state is transitioned, the family of Applications the Application belongs to, etc.

The CRS services may be accessed by any on-card Application to transition the Life Cycle state of the registered Applications and to notify the Utility Applications of this transition. The CRS services shall only be accessible to the Application itself or specific privileged on-card Applications.

Note: For each CRS registered contactless Application, the CRS shall:

- *create an entry in its table indexed by the AID of the contactless Application;*
- *store the parameters provided by the contactless Application in this entry;*
- *provide CRS services to Applications.*

2.6 Contactless Registry Event Listener (CREL)

As defined in [R14] Amendment C, the Contactless Registry Event Listener (CREL) Application is an on-card Application implementing specific business rules for a specific set of Contactless Applications. The CREL Application is referenced by one or several Applications during their installation process. The CREL Application shall be notified by the OPEN after any change to the Registry entry of any referencing Application. The CREL Application shall implement and expose the CREL Application interface.

3 USAGE OF APIs AND SIM TOOLKIT COMMANDS AND EVENTS

This chapter lists the APIs and the SIM Toolkit commands and event that are:

- ⚠ "Limited" which means that this method, command or event is sensitive and its usage by the application shall be described in the Functional Specification and in the Acknowledgment to Security Requirements,
- 🚫 "Forbidden" which means that this method shall not be used.

3.1 JAVACARD

- 🚫 **Rule 3: The shareable interface** is forbidden for applications in different SD.
- ✓ **Rule 18: Restrictions on use of JavaCard API:** The Cardlet shall obey the following restrictions on the use of the Javacard API:

Packages	Class/interfaces	Method	Rules
java.rmi (*)	ALL	ALL	🚫 Forbidden
javacard.framework	APDU	ALL	Cf. process(APDU)
	Applet	Process()	⚠ Limited
		getShareableInterfaceObject	⚠ Limited under Rule 3 conditions
	JCSysstem	lookupAID()	⚠ Limited
		abortTransaction()	⚠ Limited
		getPreviousContextAID()	⚠ Limited
		getAppletShareableInterfaceObject()	⚠ Limited under Rule 3 conditions
	MultiSelectable		🚫 Forbidden
	Shareable	ALL	⚠ Limited under Rule 3 conditions
javacard.framework.service			🚫 Forbidden

(*) Note: the usage of the Java Card RMI mechanism is prohibited, because it lacks security-related features (e.g. authentication and secure channels).

3.2 GlobalPlatform

- ✓ **Rule 19: Restrictions on use of GP API:** The Cardlet shall obey the following restrictions on the use of the GlobalPlatform API:

Packages	Class/interfaces	Method	Rules
org.globalplatform.GPSystem	GPRegistryEntry	getRegistryEntry	🚫 Forbidden
		lockCard()	🚫 Forbidden
		setATRHistBytes	🚫 Forbidden
		setCardContentState	🚫 Forbidden
		terminateCard()	🚫 Forbidden
		deregisterService	🚫 Forbidden
	CVM (*)	- verify(), - getTriesRemaining(), - is*()	✓ No restriction
		ALL other methods	🚫 Forbidden for basic applications
org.globalplatform.contactless	CRELApplication	notifyCLEvent(...)	🚫 Forbidden
	CRSApplication	processCLRequest(...)	🚫 Forbidden
	GPCLSystem	getCardCLInfo(...)	🚫 Forbidden
		getGPCLRegistryEntry(...)	🚫 Forbidden except with (null) parameter, for compliance with Rule 66.
		getNextGPCLRegistryEntry(...)	🚫 Forbidden
		setCommunicationInterface(...)	🚫 Forbidden
		setVolatilePriority(...)	🚫 Forbidden

(*) Note regarding PIN usage and CVM coding: to prevent misuse of the GlobalPlatform CVM, a basic application shall restrict access to GlobalPlatform CVM functions. It shall not offer the ability to unblock the CVM, reset the CVM try counter, or change the CVM value.

3.3 UICC

- ✓ **Rule 20: Restrictions on use of UICC API:** The Cardlet shall obey the following restrictions on the use of the UICC API:

Packages	Class/interfaces	Method	Rules
uicc.access	FileView	ALL	⚠ Limited
	UICCSysTem	ALL	⚠ Limited
uicc.access.bertlvfile	ALL	ALL	⚠ Limited
uicc.access.fileadministration	ALL	ALL	🚫 Forbidden
uicc.toolkit	ALL	ALL	See the tables below for Commands and Events
uicc.hci.framework	HCIService	requestCallb ackNotificati on (*)	🚫 Forbidden

(*) This method is deprecated in [R23], from release 9.3.

- ✓ **Rule 21: Restrictions on use of SIM Toolkit commands:** The Cardlet shall obey the following restrictions on the use of the SIM Toolkit commands:

Proactive Commands	Usage
PRO_CMD_LAUNCH_BROWSER	⚠ Limited
PRO_CMD_PERFORM_CARD_APDU	⚠ Limited
PRO_CMD_POWER_OFF_CARD	⚠ Limited
PRO_CMD_PROVIDE_LOCAL_INFORMATION	⚠ Limited
PRO_CMD_RUN_AT_COMMAND	⚠ Limited
PRO_CMD_SEND_DATA	⚠ Limited
PRO_CMD_SEND_DTMF	⚠ Limited
PRO_CMD_SEND_SHORT_MESSAGE	⚠ Limited
PRO_CMD_SET_UP_CALL	⚠ Limited
PRO_CMD_CLOSE_CHANNEL	⚠ Limited
PRO_CMD_GET_CHANNEL_STATUS	⚠ Limited
PRO_CMD_OPEN_CHANNEL	⚠ Limited
PRO_CMD_RECEIVE_DATA	⚠ Limited
PRO_CMD_SEND_DATA	⚠ Limited
PRO_CMD_REFRESH	🚫 Forbidden

Note: All others proactive commands are allowed. The list of the SIM Toolkit commands that the SP wants to use for the Cardlet should be provided to the MNO at the beginning of the project.

- ✓ **Rule 22: Restrictions on use of SIM Toolkit events:** The Cardlet shall obey the following restrictions on the use of the SIM Toolkit events:

Event	Usage
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_NAA	🚫 Forbidden
EVENT_CALL_CONTROL_BY_NAA	🚫 Forbidden
EVENT_EVENT_DOWNLOAD_MT_CALL	🚫 Forbidden
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	🚫 Forbidden
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	🚫 Forbidden
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	🚫 Forbidden
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION	⚠️ Limited
EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION	⚠️ Limited
EVENT_EVENT_DOWNLOAD_BROWSING_STATUS	⚠️ Limited
EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	⚠️ Limited
EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	⚠️ Limited

Note: All others proactive commands are allowed. The list of the SIM Toolkit event that the SP wants to use for the Cardlet should be provided to the MNO at the beginning of the project.

3.4 SIM

- 🚫 **Rule 23: SIM API (TS 43.019)** is obsolete and it is forbidden to use it.

3.5 NFC API

The standardised NFC APIs are defined in [R23].

4 MEMORY MANAGEMENT

4.1 Basic recommendations

 **Rule 24: Applet object references in static fields:** Applet object references should not be stored in static fields because when an application is deleted, it is possible that memory is not properly released. If this rule is not respected, the uninstall method shall be implemented as described in Rule 15.

 **Rule 25: Object deletion feature:** Applications should not rely exclusively on the object deletion feature because:

- it may be time-consuming,
- there is no guarantee to free instantly the memory,
- and the result is not guaranteed on some cards.

Yet, it is foreseen that some applications will need to manage dynamically some information (such as transport ticket, e-coupon ...).

In case object deletion is necessary, it is recommended that the application requests to the operating system for object deletion at the end of its process.

 **Rule 26: Use of the scratch buffer** is only authorized for local computations and is forbidden for exchanges between applets.
The scratch buffer accessible through `UICCPlatform.getTheVolatileByteArray` could be used to share data with applications belonging to different Java Card contexts; usage of this buffer shall be restricted to local (not shared) computations.

 **Rule 27: Arrays allocation:** Arrays shall be allocated with a determined size.
To prevent unexpected memory failures, applications shall only allocate arrays of a fixed length, i.e., the array size shall not be dynamically calculated. Thus, the usage of constant values is mandatory.

4.2 Memory Protection

 **Rule 28: FileView objects allocation:** FileView objects shall be allocated during application installation.

 **Rule 30: Recursive code** is forbidden.
Recursion can lead to stack overflow, especially in Java Card, and it shall be avoided altogether.

4.3 Memory allocation/de-allocation (RAM, Flash, EEPROM)

 **Rule 31: Objects allocation:** All objects shall be allocated in the installation phase, or allocated with a singleton if allocated outside the installation.
No allocation should take place after the registration of an applet, i.e., in the normal use phase, during the processing of SIM Toolkit events or APDU's. The issue here is that such allocations could be repeated many times, and rapidly lead to saturation of the application's space, making the application unusable.
In case the application needs to allocate objects after applet registration, either for functional or performance reasons, it is mandatory to protect the allocation with a singleton to avoid allocation repetition.

5 HANDSET RESOURCES MANAGEMENT

This table presents the restrictions regarding the usage of the mobile handset resources. Mobile resources could be used by Card Application Toolkit (CAT), especially by proactive commands.

✓ **Rule 32: Handset resources management:** the Cardlet shall obey the following restrictions:

RESOURCES	RULES	CONDITIONS
Data		
Customer personnel information access on the handset (PIM) : directory, agenda, notes, messages, sounds, pictures ...	⚠ Limited access	Under user control
Capture interfaces access: sound record, pictures, localization...	⚠ Limited access	Under user control
Communications		
Call interception (voice, visio, data ...etc.)	⚠ Limited access	Under user control
Set Up Call	⚠ Limited access	Under user control Fixed destination (Constant address not calculated – Tags information) Variable Addresses are forbidden.
Message Services		
Messages interception (SMS, MMS, e-mail ...etc.)	⚠ Limited access	Under user control
Messages sending (SMS, MMS, e-mail ...etc.)	⚠ Limited access	Under user control Fixed destination (Constant address not calculated – Tags information) Variable Addresses are forbidden.
Local Connectivity		
Local connection establishment (serial, IR, Bluetooth, WIFI, NFC...)	⚠ Limited access	Under user control
Distant Connectivity		
Distant connection establishment (http, https...)	⚠ Limited access	Under user control Variable Addresses are forbidden
Application Management		
Management or triggering of other applications	⚠ Limited access	
Others		
Interrogation of capacities or configuration of the terminal	⚠ Limited access	Large Interrogation Forbidden : Interrogation clarifies resources used and strictly necessary for the application

6 SYSTEM MANAGEMENT

6.1 File System Access

- ✓ **Rule 34: File access:** Files authorized in restricted accesses are specified below (subject to the MNO agreement). Access to all other files (reading, writing) and creation of files (except the application files) are forbidden.

ID1	ID2	ID3	Nom	Limited access	Conditions
3F00			MF	-	
2FE2			ICCID	Read	
7F10			TELECOM	-	
	6F3A		ADN	Read/Write	Under user control
	6F3C		SMS	Read	Under user control
	6F44		LND	Read	Under user control
	5F3A		PUBLIC PHONE BOOK	-	Under user control
		4F30	PBR	Read	Under user control
		4F3A	ADN	Read/Write	Under user control
		4F09	PBC	Read/Write	Under user control
		4F40	ANR1	Read/Write	Under user control
		4F60	SNE1	Read/Write	Under user control
		4F61	SNE2	Read/Write	Under user control
		4F50	EMAIL	Read/Write	Under user control
		4F10	UID	Read/Write	Under user control
		4F22	PSC	Read/Write	Under user control
		4F23	CC	Read/Write	Under user control
		4F24	PUID	Read/Write	Under user control
7F20			GSM	-	
	6F07		IMSI	Read	
	6F14		CPHS : ONS	Read	
	6F46		SPN	Read	
7FFF (*)			USIM		
	6F07		IMSI	Read	
	6F3C		SMS	Read	Under user control
	6F46		SPN	Read	
	6F80		ICI	Read	Under user control

ID1	ID2	ID3	Nom	Limited access	Conditions
	6F81		OCI	Read	Under user control
	6F82		ICT	Read	Under user control
	6F83		OCT	Read	Under user control
	6FCE		MMSN	Read	Under user control
	6FD0		MMSICP	Read	Under user control
	6FD3		NIA	Read	Under user control

(*) see [R24]

6.2 File Event and Management

- ✓ **Rule 35: Registration to files event:** The use of ToolkitRegistry.registerFileEvent method shall be limited to specific application files (under its ADF). Applications shall not listen to event on files controlled by the MNO. This guideline extends the file access restriction guidelines to the usage of file events with the toolkit registry.
- ✓ **Rule 36: Files creation:** all needed files shall be created during installation phase.
The files created by the application shall all be clearly identified with determined identifiers. In particular, files shall not be created in a loop, or by considering installation parameters. To prevent memory failures and the saturation of the memory space (making the USIM unusable), no creation shall take place after the registration of an applet.
- ✓ **Rule 37: ADF:** Files shall only be created in ADF.
The master file (MF) is restricted to standardized files (GSM) and any others telecom operators files. The Service Provider files shall be created under an ADF dedicated for NFC applications.
- ✓ **Rule 38: Restrictions to file system access:** ADFs and files shall be accessed with determined identifiers.
Applications are not intended to access any file in the file system. Then, to determine precisely which files are accessed, applications shall only access ADFs and files with determined identifiers.
- ✓ **Rule 39: AID Values:** Constants shall be used for AID values (except application instance AID).
- ✓ **Rule 42: Phone book files integrity:** The application should verify the content written in phone book files.
The phone book structure is based on several additional elementary files, to fully manage contacts instead of simple name/dialing number couples.
 - EFADN: Abbreviated Dialing Number
 - EFANR: Additional Number
 - EFAAS: Additional number Alpha String
 - EFEXT1: Extension 1
 - EFCCP1: Capability Configuration Parameters 1
 - EFEMAIL: Email Address

- EFSNE: Second Name Entry
- EFGRP: Grouping File
- EFGAS: Grouping information Alpha String
- etc.

The framework does not provide any specific command for the management of these files and the data manipulation inside a file record is difficult. In addition, the information in these files should only consist of standard characters, without any control characters. Applications should verify the content written in phone book files to prevent erroneous or malformed data to be written.



Rule 43: Resizing of files after their creation is forbidden.

The size of files shall not be modified after their creation. Applications shall not use the `FileAdminView.resize` method; instead the application shall gracefully inform the end user that no more space is available.

7 EXCEPTION AND ERROR MANAGEMENT

- ✓ **Rule 44: Catch of exceptions:** All exceptions thrown during the execution from any defined entry point shall be caught by the application. Only ISOException shall be returned by the process() method.
- ✗ **Rule 45: Ban on exceptions:** A basic application shall not include any code that leads to the exceptions listed below, and these exceptions shall not be caught:
- SecurityException,
 - ArrayStoreException,
 - ClassCastException,
 - IndexOutOfBoundsException,
 - ArrayIndexOutOfBoundsException,
 - NegativeArraySizeException
 - NullPointerException

8 ROBUSTNESS AND INTEROPERABILITY

8.1 General recommendations

In order to assure a minimal robustness, unusual usage of the service shall be tested.

Functional interoperability shall be verified by testing all targeted smart card platforms and a significant number of handsets.

- ✗ **Rule 46:** Development and use of platform specific APIs or libraries outside the scope of the platform certification TOE (Target Of Evaluation) is forbidden.

9 INTERACTIONS

9.1 Handset (STK, USB, ...)

- ❌ **Rule 47: ISOException (REPLY_BUSY)** shall not be used as a conclusion to event processing. Exceptions thrown by toolkit applications are all caught by the CAT Runtime Environment, except instances of ISOException, with reason REPLY_BUSY, which are propagated to the terminal. Because of interoperability issues, this mechanism shall not be used by applications.
- ❌ **Rule 48: Hidden channels** are forbidden. All communication mechanisms used for application communication shall be standard.
- ❌ **Rule 49: Ban of objects with share of system interfaces:** Use of objects that implement system interfaces shared by another application is forbidden.

10 DEVELOPMENTS RULES

-  **Rule 52: GP Privileges:** a basic application shall not have the following GlobalPlatform privileges:
 - SD and associated privileges
 - Card lock
 - Card terminate
 - Card reset
 - CVM Management
 - Global Delete
 - Global lock
 - Global registry
 - Final application

-  **Rule 54: Telecom Applets:** either in 2G or 3G domains, a basic application shall never be assigned an access domain (for UICC file system or for any ADF, in normal access or administrative access) that gives more right than needed (“least privilege” principle).

-  **Rule 55: Switch statement and case:** A switch statement shall always include a default case. No case shall be ignored. If there is no specific handling to do, it is possible to declare a default clause at the end of the switch statement, and to simply include a comment indicating that there is nothing to do.

-  **Rule 57: Compliance to ISO 7816-4 specification:** An application should only accept commands and return status words that are valid according to ISO7816-4 specification, and in particular:
 - CLA should not be 0xFF.
 - INS should always be an even value, and its most significant nibble should not be 0x6 or 0x9.
 - The most significant byte of status word should be in ranges 0x6X and 0x9X.

-  **Rule 58: Low-order CLA bits** should be masked.

Since the Java Card 2.2 specification, different applets can be selected at the same time on different logical channels. As defined in ISO7816-4 specification, the two least significant bits of the CLA byte are used to indicate the channel number. To prevent any command recognition issue, applications need to ignore the channel bits, for instance masking it with byte 0xFC.

-  **Rule 70: use of SetIncomingAndReceive method**

The SetIncomingAndReceive method shall be called only for case 3 or case 4 commands.

-  **Rule 60:** Registration to proprietary events is forbidden.

This is a basic portability guideline: applications shall not manage proprietary events.

-  **Rule 62: Protection of access to handlers:** The application should protect all accesses to handlers by an exception handler. Because handlers are not always available (e.g., re-entrance case) and a ToolkitException with reason HANDLER_NOT_AVAILABLE may be thrown, implementations should protect all accesses to handlers in a try-catch block.

-
- ✓ **Rule 63: Declarations of classes, fields and methods:** The application should properly declare the attributes of classes, fields, and methods.
All classes, fields, and methods should be declared as privately as possible, and final whenever possible.

 - ✓ **Rule 64: Naming and declaration of constants:** All constants should be named and declared as static final fields.
Naming constants is important for readability and maintenance, and declaring them properly is important for size optimization and performance.

11 SYNTHESIS OF RULES

The following table classifies the rules according to their associated category: state-of-the-art, functional / configuration, security (as defined in part 1.3).

Rule ID and description	Mandatory	Recommended	Has changed? (*)	State-of-the-art	Functional / Config	Security
Rule 0: Functional Specification and Acknowledgment to Security Requirements	X		New		X	X
Rule 1: Interface management and Data management	X			X		
Rule 2: The interactions between different interfaces	X			X		
Rule 3: The shareable interface	X					X
Rule 4: Package design	X					X
Rule 5: Entry points	X					X
Rule 6: Denial of Service	X			X		X
Rule 7a: Dead code	X		Upd.	X		X
Rule 7b: Debug information	X		New	X		X
Rule 8: Coding NOP	X		Upd.	X		X
Rule 9: Versioning	X		Upd.	X		X
Rule 10: ByteCode verification	X		Upd.			X
Rule 11: Shared resources	X		Upd.	X		X
Rule 12: Checking of Cardlet installation parameters	X				X	
Rule 13: Applet instance registration	X		Upd.		X	
Rule 14: STK menu entries initialization	X				X	
Rule 15: Implementation of the uninstall method	X		Upd.			X
<i>Rule 16: Before the Cardlet deletion stage</i>			Suppr.			
Rule 17: Cardlet deletion phase	X					X
Rule 18: Restrictions on use of JavaCard API	X		Upd.			X
Rule 19: Restrictions on use of GP API	X		Upd.			X
Rule 20: Restrictions on use of UICC API	X		Upd.			X
Rule 21: Restrictions on use of SIM Toolkit commands	X		Upd.			X

Rule ID and description	Mandatory	Recommended	Has changed? (*)	State-of-the-art	Functional / Config	Security
Rule 22: Restrictions on use of SIM Toolkit events	X		Upd.			X
Rule 23: SIM API (TS 43.019)	X			X		X
Rule 24: Applet object references in static fields		X	Upd.	X		
Rule 25: Object deletion feature		X		X		
Rule 26: Use of the scratch buffer	X					X
Rule 27: Arrays allocation	X			X		
Rule 28: FileView objects allocation	X			X		
<i>Rule 29: Files creation</i>			Suppr.			
Rule 30: Recursive code	X			X		X
Rule 31: Objects allocation	X		Upd.	X		
Rule 32: Handset resources management	X					X
<i>Rule 33: Ban on file access</i>			Suppr.			
Rule 34: File access	X		Upd.			X
Rule 35: Registration to files event	X		Upd.			X
Rule 36: Files creation	X		Upd.	X		X
Rule 37: ADF	X			X		X
Rule 38: Restrictions to file system access	X					X
Rule 39: AID Values	X			X		
<i>Rule 40: Access to files controlled by the MNO</i>			Suppr.			
<i>Rule 41: Registration to files event</i>			Suppr.			
Rule 42: Phone book files integrity		X				X
Rule 43: Resizing of files	X					X
Rule 44: Catch of exceptions	X		Upd.			X
Rule 45: Ban on exceptions	X		Upd.			X
Rule 46: Development and use of platform specific APIs or libraries	X					X
Rule 47: ISOException (REPLY_BUSY)	X					X
Rule 48: Hidden channels	X					X
Rule 49: Ban of objects with share of system interfaces	X					X
<i>Rule 50: Events registration and STK menu initialization</i>			Suppr.			

Rule ID and description	Mandatory	Recommended	Has changed? (*)	State-of-the-art	Functional / Config	Security
<i>Rule 51: Use of Java Card RMI</i>			Suppr.			
Rule 52: GP Privileges	X					X
<i>Rule 53: Coding CVM: PIN usage</i>			Suppr.			
Rule 54: Telecom Applets	X					X
Rule 55: Switch statement and case	X			X		
<i>Rule 56: Use of the int type</i>			Suppr.			
Rule 57: Compliance to ISO 7816-4 specification		X	Upd.	X	X	
Rule 58: Low-order CLA bits		X		X	X	
<i>Rule 59: Registration to call and SMS control events</i>			Suppr.			
Rule 60: Registration to proprietary events	X					X
<i>Rule 61: In ViewHandler and EditHandler</i>			Suppr.			
Rule 62: Protection of access to handlers		X				X
Rule 63: Declarations of classes, fields and methods		X		X		
Rule 64: Naming and declaration of constants		X		X		
<i>Rule 65: Use of an identifier in function calls</i>			Suppr.			
Rule 66: GP Contactless Self-Activation privilege	X		New		X	
Rule 67: Minimum Security Level (MSL)	X		New		X	X
Rule 68: Priority Level	X		New		X	X
Rule 69: Access Domain	X		New		X	X
Rule 70: use of SetIncomingAndReceive method	X		New	X	X	

(*) In this column, the rules are marked:

- **New** if this rule is **New**, compared with the previous version of the specification
- **Upd.** if this rule has been **Updated** from the previous version of the specification
- **Suppr.** if this rule was in the previous version of the specification and has been **Suppressed**
- If a rule has been kept as-is from the previous version of the specification, it is not marked

CROSS-CHECK TABLE RELATED TO THE PPUSIM V2 RULES FOR BASIC APPLICATIONS.

This appendix specifically includes a cross-check table to ensure that the requirements specified in the “(U)SIM Java Card Platform Protection Profile, ref PU-2009-RT-79 [R18]” are covered in the present development guideline.

Overall, these requirements were already accounted for into the present development guideline but for coverage reasons, the below requirements are specifically copied from [R18] to makes certain that all is taken into account.

Hence, the below table comes from the section “3.4.1.1 Basic and Secure Applications Policies” of [R18] and focus on basic applications, i.e. non-sensitive applications other than EAL4+ banking applications:

- OSP.BASIC-APPS-VALIDATION requirements table:

Item	See section
Basic applications shall be associated to a digital signature which will be checked by a VA during the loading into the TOE.	Covered in [R2], “NFC Applications Validation Process”
In addition to the rules stated by the Java Card specification, the validation process shall enforce that basic applications:	
- cannot be libraries,	Covered in chap. 9.1 Handset (STK, USB, ...)
- shall not use RMI,	Covered in chap. Erreur ! Source du envoi introuvable. Erreur ! Source du renvoi introuvable.
- shall not use proprietary libraries which are not certified (except system libraries),	Covered chap 8.1 General recommendations
- access control to certified proprietary libraries is controlled by the secure application which has defined the library,	Covered in chap 2.1 Documentation, 3.1 JAVACARD, 9.1 Handset (STK, USB, ...).
- shall be associated to an identifier and this identifier has to be used in parameter of the function calls.	Covered in chap 10 Developments rules.
Application note:	
- GSM file system and API's STK application descriptors are other ways to share object between applications.	Covered in chap. 9.1 Handset (STK, USB, ...)
- Identifier usage allows easy tracking of applications calls. This is useful if a new attack path is discovered to identify the pieces of code that could be vulnerable.	Covered in chap 10 Developments rules