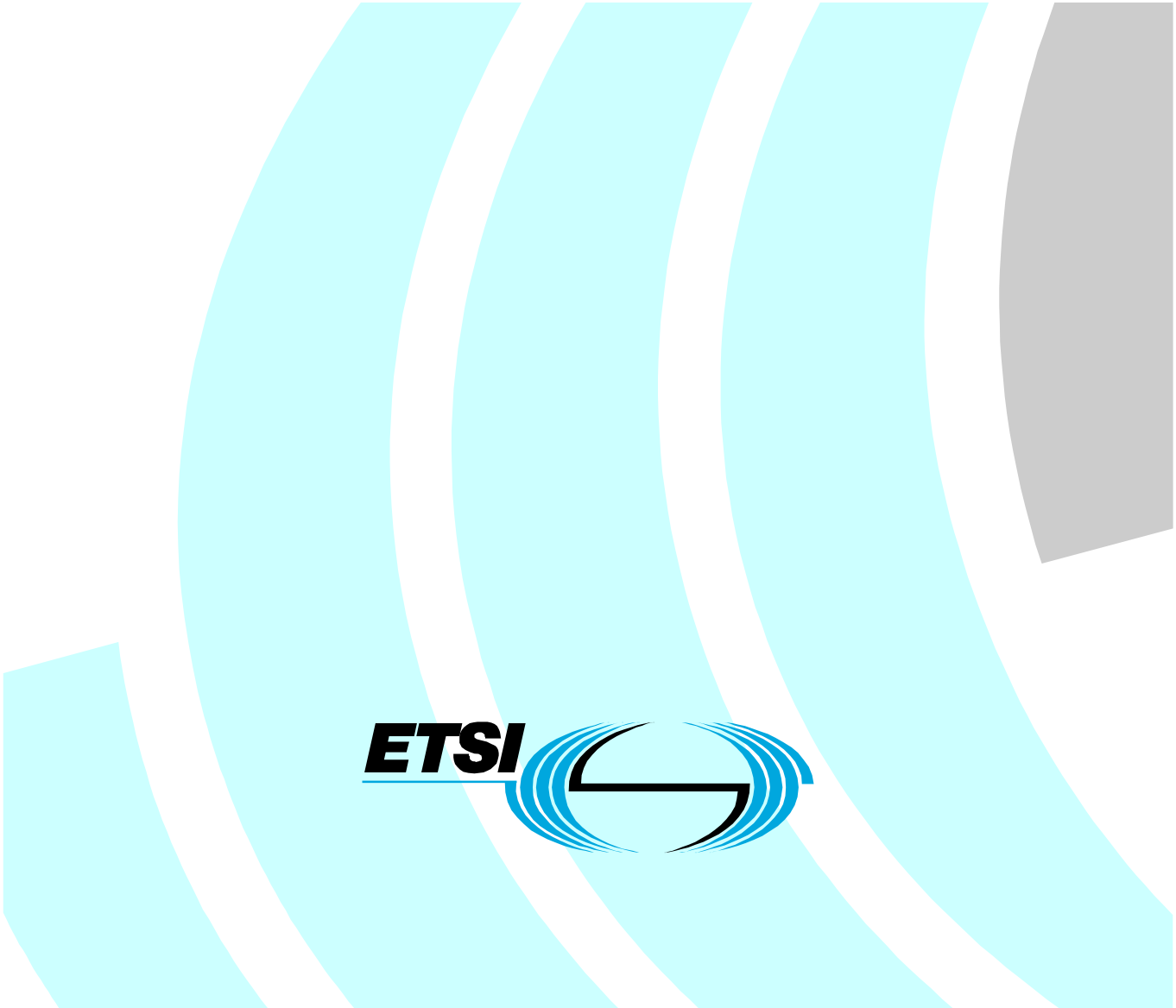
ETSI TR 102 481 V7.0.0 (2007-02)

Technical Report

Smart Cards;
Requirements for UICC - external peripherals data exchange
(Release 7)



Reference
DTR/SCP-R00003

Keywords
Smart Card

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP decides to modify the contents of the present document, it will be re-released by EP SPC with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x: the first digit:

1 presented to TC SCP for information;

2 presented to TC SCP for approval;

3 or greater indicates TC SCP approved document under change control.

y: the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z: the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document analyses the possibility to define a general mechanism for data exchange between the UICC and different kind of Terminal's external peripherals.

2 References

For the purposes of this Technical Report (TR), the following references apply:

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

[1] ISO/IEC 7816-3: "Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols".

[2] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

[3] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".

[4] ETSI TS 102 412: "Smart cards; Smart Card Platform Requirements Stage 1".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following term and definition applies:

external peripherals: device connected or integrated within the handset, such as MMC/SD slot, fingerprint sensors, GPS device

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

APDU	Application Protocol Data Unit
API	Application Programming Interface
CAT	Card Application Toolkit
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
MMC	MultiMediaCard
MMS	Multimedia Messaging Service
NUT	New UICC-Terminal interface
SD	Secure Digital

4 Use Cases

4.1 Memory Card

The UICC shall be able to seamlessly exchange data with Memory Card as a companion device inserted to mobile terminal, for example to access pictures and multimedia files directly from the UICC.

Through an appropriate set of commands the UICC could have access to the Memory Card in order to store there some pictures linked to the card phonebook, MMS, etc.

4.2 GPS

Most of 3G terminals in the field can be able to access a GPS device or will have an integrated GPS. Allowing the UICC to retrieve localization information from this device in a generic way through the Mobile Terminal can provide great flexibility for operators in designing services.

Within this scenario the UICC shall be able for example to access a standard GPS device connected to the handset reporting positioning information to a secure server. This can be used by a sales force, for example, and authorized personnel only can be able to access the geographical positions of those subscribers. Similarly, a granted server can ask the UICC for the position of the customer and the UICC could request positioning information to the GPS peripheral sending it back encrypted to the server.

Summarizing, the UICC shall have the possibility to:

- ask for GPS localization data (i.e. GPS position, GPS velocity, etc. or GPS code-phase, doppler, etc.) and other GPS parameters;
- provide data to the GPS device (i.e. GPS Assistance data).

4.3 Biometric Sensors

Services requiring mobile payments or anyway a strong security can be accessed after a biometric authentication done with a Mobile Terminal equipped with a biometric sensor (e.g. fingerprint sensor, camera, etc.). In this case an application can ask the user to authenticate itself with a biometric characteristics before accessing the payment service. When the user try to access these kind of services the UICC can exchange data with the biometric sensor through an appropriate set of commands retrieving the information required for the authentication, in case of a matching algorithm running on the UICC, or the authentication response in case of a matching algorithm running on the Mobile Terminal. This use case requires establishing a secure channel between the UICC biometric application and the biometric sensor on the terminal but also an high speed interface if the matching algorithm runs on the card.

4.4 General proprietary device interface

There are many new usecases emerging that are enhanced by allowing the UICC to use a proprietary protocol interface to a device that is connected to the terminal.

Examples are:

- Printer connection.
- Telemetry device control.
- Vending machine control.
- Car system control.
- Remote control systems.

For these usecases a set of generic commands are needed that allows the UICC to use these proprietary protocols. The UICC will need to be able to discover these devices from the terminal and establish which protocol is to be used.

It is expected that the UICC will initiate communication with this device, however some protocols may require that the connected device is able to trigger the UICC communication.

5 Preliminary Requirements Considerations

5.1 General

These peripherals management capabilities can be used by applications to store and retrieve data from memory cards, to activate a camera, to require biometric authentication using the fingerprint sensor or to exchange data with a GPS device in order to provide enhanced services.

It shall be possible to define a mechanism to reach these peripherals with both ISO/IEC 7816-3 [1] interface and NUT interface. The mechanism shall exploit the interface capabilites and for this reason it could be different in the two cases (ISO/NUT interfaces).

A list of the available peripherals (e.g. GPS device, memory slots, etc.) shall be sent by the Mobile Terminal to the UICC at power on. Moreover if a peripheral is added subsequently, the Mobile Terminal shall inform the UICC (i.e. if the user inserts a memory card in the mobile terminal, it shall communicate it to the card). The UICC shall then have the capability to ask for information and status of a particular peripheral and also to ask the Mobile Terminal to set up a connection with a peripheral chosen and to remove it.

The physical access between the UICC and the peripherals has to be done through the Mobile Terminal exploiting the standard interfaces for UICC and the peripherals themselves so that only a logical access can be established between them.

5.2 Commands Functional Description

Hereafter there is a functional overview of the commands that could be exchanged between the UICC and the Terminal in order to perform actions on a terminal peripheral. The set of commands has to be implemented in the proper way on different application protocols (i.e. APDU, CAT or HTTP) depending on the UICC-terminal interface.

GET PERIPHERAL STATUS: a command is required in order to know if the peripheral is present in the terminal, which type of peripheral is inserted in the terminal and if it is available.

OPEN PERIPHERAL: a command is required in order to open a connection with the peripheral.

PERFORM PERIPHERAL COMMAND: a command (or a set of commands) is required in order to Read/Write data from/to the peripherals and, when necessary, to set parameters on the peripherals.

CLOSE PERIPHERAL: a command is required in order to close the connection with the peripheral.

6 Conclusion

In order to define a mechanism for the UICC-External Peripherals data exchange a unique set of requirements can be specified identifying the commands functional behaviour with a CR to TS 102 412 [4].

The mechanism implementation should then be studied and specified taking into account the different application protocol depending on the adopted UICC-Terminal Interface and requiring CRs to the appropriate Technical Specifications.

Annex A: Examples

A.1 Virtual Peripheral Example

The Virtual Peripheral concept could be taken as an example in order to minimize the impact on Terminal.

In order to standardize a minimal set of commands, allowing the communication with external peripherals, the virtual peripheral concept has been adopted by many operating systems. Virtual peripherals are peripheral devices simulated by the operating system. An abstraction layer allows the definition of a common interface to control peripheral functionality just using a few APIs.

A virtual peripheral implements the software driver, which receives standard APIs and translates them in specific peripheral commands.

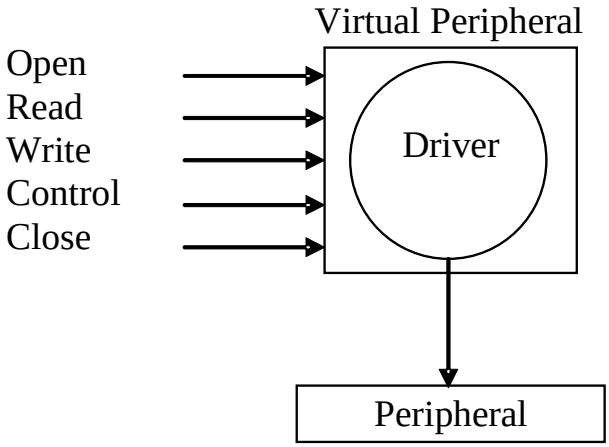


Figure 1: Virtual peripheral

Applications can get information from the peripheral or put some control value to configure/activate functionalities through Control API that can be addressed using a set of unique numbers (one per functionality) and that are specified according to the peripheral functionalities.

Once specified the APIs set, the interface between applications and virtual peripherals is fixed. To support new peripherals, developers just need to implement the new associated driver according to the peripheral functionality and the Control parameters (unique numbers and additional arguments) definition.

Virtual Peripherals concept could then be used to simplify the definition of the UICCcommands set dealing with the Open, Read, Write, Control and Close API for each terminal peripheral.

Let's consider for example the memory card use case. Without knowing which kind of memory card is inserted in the terminal an application in the UICC could be able to exchange data trough a simple set of commands such as:

- GET PERIPHERAL STATUS (peripheral=memory card).
- OPEN PERIPHERAL (peripheral=memory card).
- PERFORM PERIPHERAL COMMAND (Control).
- PERFORM PERIPHERAL COMMAND (Read; data=Data to Read).
- PERFORM PERIPHERAL COMMAND (Write; data=Data to Write).
- CLOSE PERIPHERAL (peripheral=memory card).

The terminal could interpret the meaning of each API (Open, Read, Write, Control and Close) for the memory card peripheral being able, through a specific driver, to translate this request to the proper protocol for MMC, SD card, Compact Flash or Memory Stick.

Another example could be done with a biometric sensor. In this case the advantage of the Virtual Peripherals concept is greater than the previous one due to the chance to avoid the need to know all the specific interfaces with the different sensors.

The command sequence could be the same, but in this case Read command could be the acquisition of the biometric template and the Write command could be used to pass the template stored in the UICC to the biometric sensor.
The UICC application could use the Control API in order to know which kind of biometric facility it can exploit.

Change History

Change history		
0.0.1	2006-07	SCP REQ#10
0.3.0	2006-07	SCP REQ#10
1.0.0	2006-07	SCP #26
1.1.0	2006-09	SCP #27: The reference to TS 102 412 has been added in the References Section
1.2.0	2007-01	SCP#28: Very minor editorial correction: date of the document and "smart quotes" replaced in added reference. Hanging paragraph repaired in section 5. ToC updated accordingly.
1.3.0	2007-01	SCP REQ#12: Updated Use case in section 4.4 and some minor editorials.
2.0.0	2007-01	SCP REQ#12: minor editorial changes
7.0.0	2007-01	SCP#29 approved

History

Document History		
V7.0.0	February 2007	Publication