ETSI TS 102 267 V15.0.0 (2019-02)

Smart Cards;
Connection Oriented Service API for the Java Card™ platform
(Release 15)

Reference

RTS/SCP-T006vf00

Keywords

API, protocol, smart card, testing, transport

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://ipr.etsi.org/).

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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) 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 modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 0 early working draft;
  - 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.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines an Application Programming Interface for the Java Card™ to use transport protocols (e.g. CAT_TP as defined in ETSI TS 102 127 [1]) for CAT applications.

This stage 2 document describes the interface functionalities, the interface working mechanisms and its information flow.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

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

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 127: "Smart Cards; Transport protocol for CAT applications; Stage 2 (Release 6)".

[2] ORACLE Java Card™ Specification: "Java Card™ 3.0.1 Virtual Machine Specification".

NOTE: Available at http://docs.oracle.com/javame/javacard/javacard.html.

[3] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".

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

[5] GlobalPlatform: "GlobalPlatform Card, Remote Application Management over HTTP, Card Specification v2.2- Amendment B", Version 1.1.3.

NOTE: Available at http://www.globalplatform.org/.

[6] ETSI TS 102 483: "Smart cards; UICC-Terminal interface; Internet Protocol connectivity between UICC and terminal".

[7] ETSI TS 102 225: "Smart Cards; Secured packet structure for UICC based applications".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

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

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

BIPLink: interface to access the physical layer by means of the Bearer Independent Protocol according to ETSI TS 102 223 [4]

transport layer: instance within the card framework which implements the transport protocol, e.g. CAT_TP

3.2 Symbols

Void.

3.3 Abbreviations

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

ACK	ACKnowledge
AID	Application IDentifier
API	Application Programming Interface
BIP	Bearer Independent Protocol

NOTE: See ETSI TS 102 223 [4].

CAT	Card Application Toolkit

NOTE: See ETSI TS 102 223 [4].

CAT_TP	Card Application Toolkit Transport Protocol

NOTE: See ETSI TS 102 127[1].

HTTP	HyperText Transfer Protocol
IP	Internet Protocol
JCRE	Java Card™ Runtime Environment
PDU	Packet Data Unit
SCP81	Secure Channel Protocol '81'

NOTE: See "GlobalPlatform Card Specification, Amendment B" Version 1.1.3 [5].

SCWS	Smart Card Web Server

SDU	Service Data Unit
SMS	Short Message Service
TCP	Transmission Control Protocol
TLS	Transport Layer Security

4 Overall description

4.0 General

The present document describes an API that provides applications a set of Connection Oriented Services. This API provides either direct access to the transport protocols supported by the terminal (by using the BIP) or access to transport protocol layers provided by the UICC, e.g. the CAT_TP protocol layer in the UICC or SCP81.

4.1 Connection API concept

The API is based on the concept of objects that encapsulate the features of connection service (e.g. opening a service, sending and receiving data, etc.).

The present document provides three variants of connection services to the application. These variants are based on the *Connection* interface which is an abstraction of the BIP channel which is used to exchange data.

One type of connection service is accessible through the *BIPLink* interface. It provides direct access to the transport protocol stack in the terminal. Another type of connection is accessible via the *UICCTransportLink* interface. It provides an interface to a transport protocol layer deployed on the UICC, e.g. CAT_TP. The transport protocol layer on the UICC uses BIP to transport its PDU to the terminal. The last type of connection is accessible through the *SCP81Connection* interface. It provides an interface to a transport protocol layer based on SCP81 as defined in [5].

Implementations of *BIPLink* interface and *UICCTransportLink* interface are based on the *Observer* design pattern. I.e. the application is notified about state changes (e.g. data received, channel closed, etc.) by means of events sent to the *Observer* interface.

The support of each type of connection is optional. If a type of connection is not supported, the related method in the *ConnectionServer* class shall throw a *ConnectionException* with reason code *TRANSPORT_PROTOCOL_NOT_SUPPORTED*.

5 API architecture

5.0 General

For a reference documentation of all classes and interfaces of the Connection oriented Services API refer to annex A.

5.1 API usage

5.1.1 Establishing a Connection

To perform operations on a specific link it is required to establish a *Connection;* a *Connection* can be considered as a pure point to point connection on which a link can be established.

Depending on the nature of the link, the *Connection* may be shared among different links or can be used exclusively by one link. In case of sharing, all the *Connection* shall have the same parameters. Even if the links allow to share the same *Connection*, it is possible, at *Connection* instantiation time, to specify that a *Connection* cannot be shared among several links.

The *Connection* may require some specific Toolkit resources to be available to the application, e.g. in case of BIP connections, at least one BIP channel shall be available to the application when the *Connection* is opened.

5.1.2 Opening a BIPLink

A *BIPLink* requires the usage of an underlying *Connection* to transport data according to the BIP protocol. The used *Connection* shall be a BIP connection.

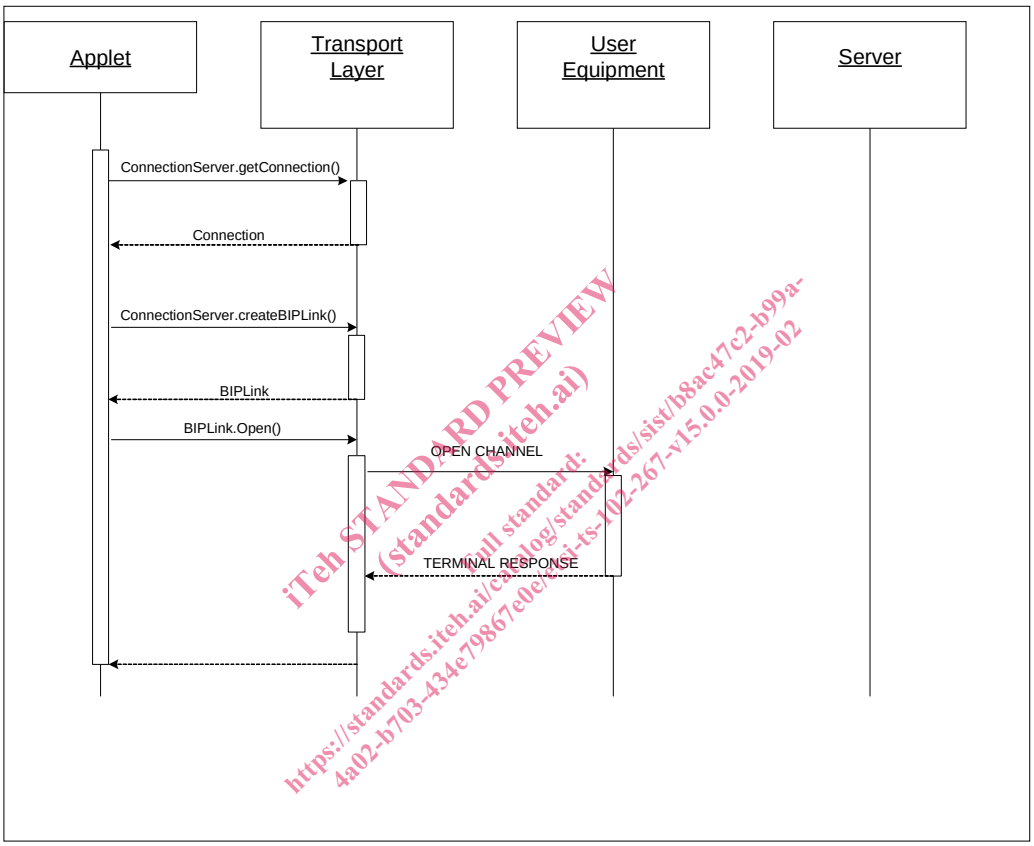

Figure 1: Opening a BIPLink

5.1.3 Creating an UICC Transport link

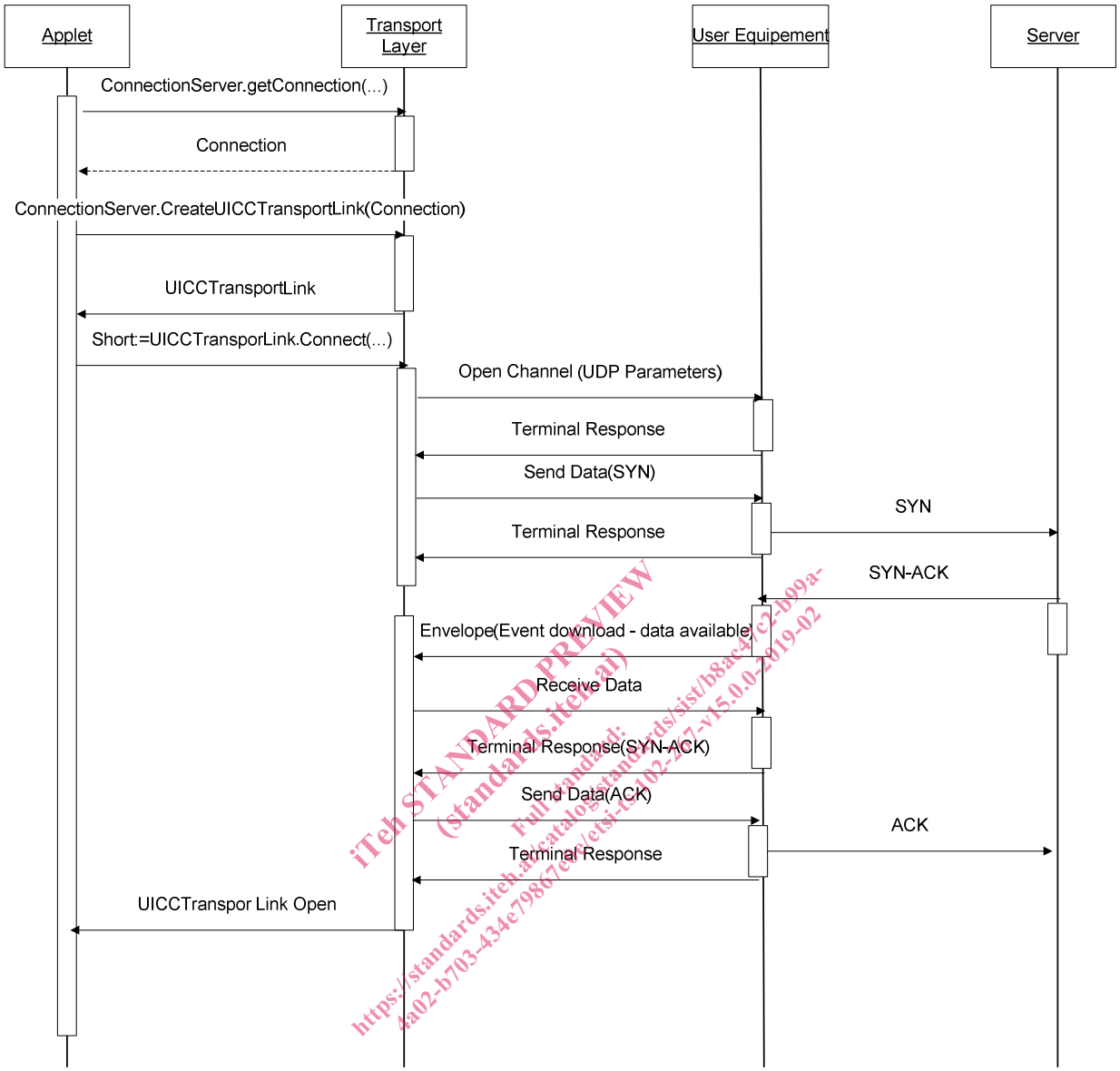

Figure 2: Creating a reliable link

The API is considered as a blocking API which means that no other applet is able to use the *uicc.toolkit.ProactiveHandler* while the call to the method *UICCTransportLink.connect()* is ongoing, i.e. the three way handshake is performed.

5.1.4 Sending data over UICC Transport Link

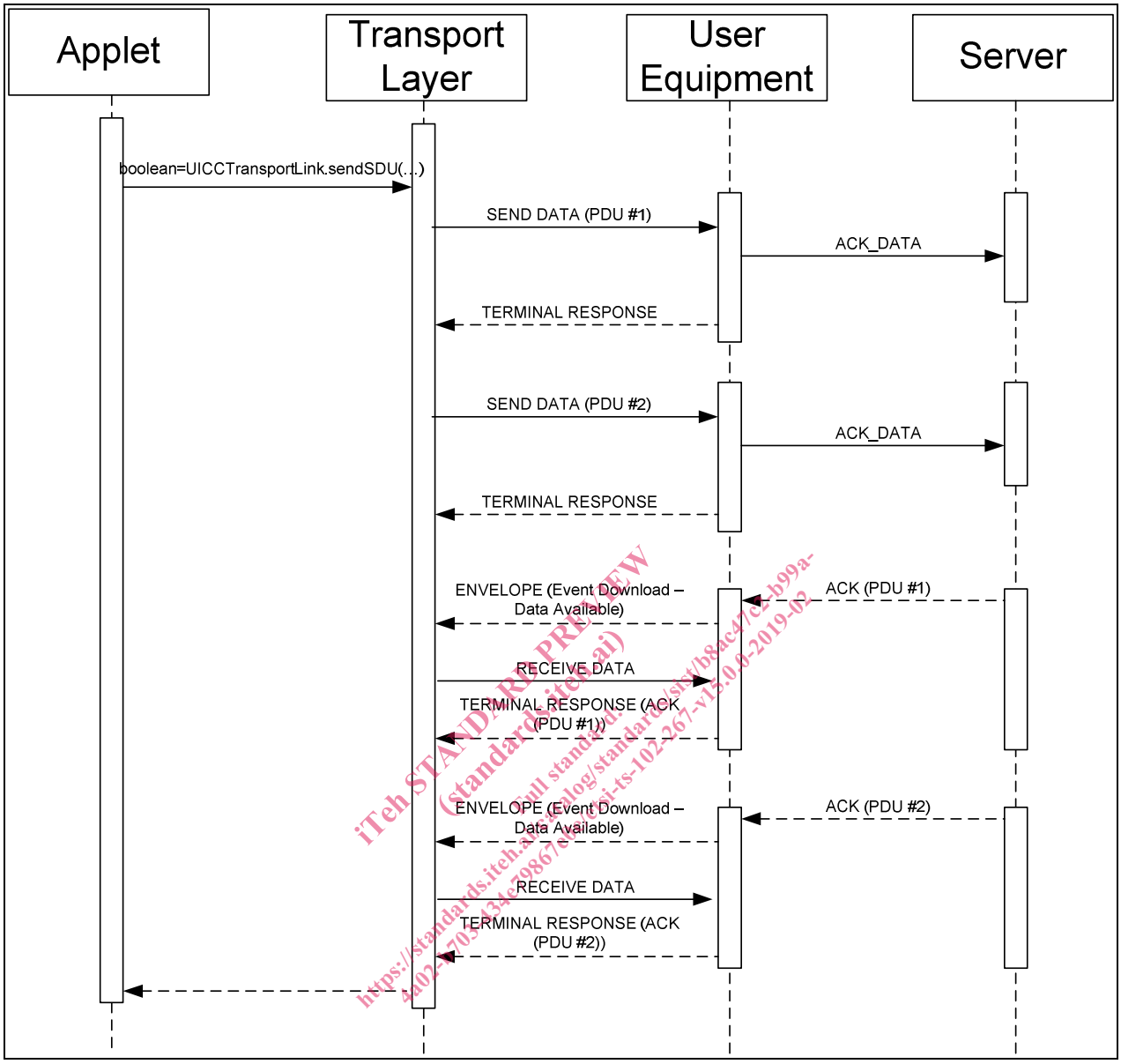

Figure 3: Sending data over Reliable Link

No other applet is able to use the *uicc.toolkit.ProactiveHandler* while the call to the method *UICCTransportLink.sendSDU*(..) is ongoing, i.e. that each PDU has been acknowledged.

After UICCTransportLink.sendSDU method returns, the uicc.toolkit.ProactiveHandler shall be available to the calling applet if it was available before the method invocation; however, handler content may be affected by the system during method execution.