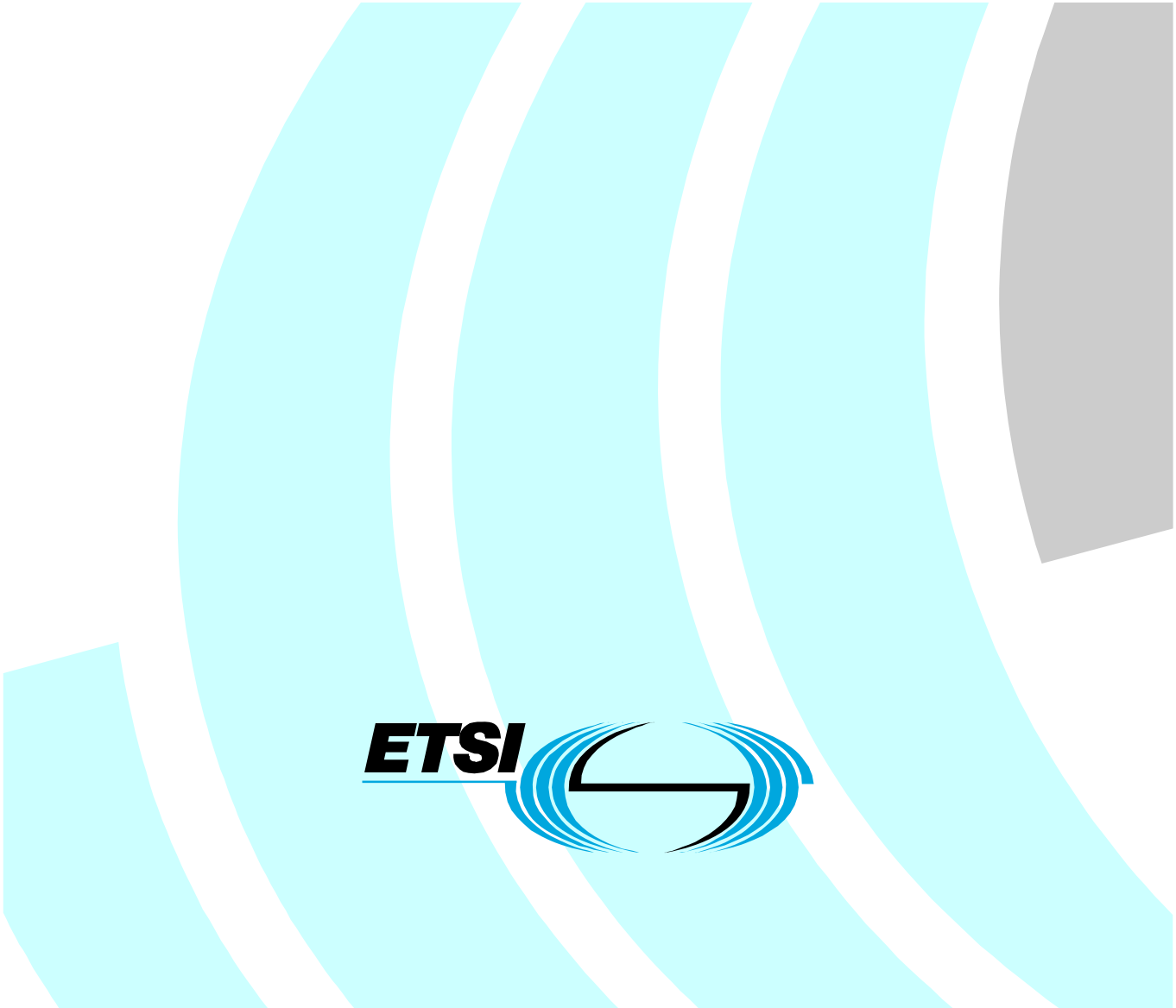
ETSI TS 102 588 V7.0.0 (2007-07)

Technical Specification

Smart Cards; Application invocation Application Programming Interface (API) by a UICC webserver for Java Card™ platform; (Release 7)



Reference
DTS/SCP-T006

Keywords
Smart Card, API

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

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

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
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

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 Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

1 Scope

The present document defines an API that allows a UICC based SCWS defined by OMA to forward Http requests to an Applet and to recive the response from the Applet. It also defines an API for the Applet to register and unregister to the SCWS.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a Technical Committee SCP document, a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

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

[1] "Hypertext Transfer Protocol -- HTTP/1.1".

NOTE: available at http://www.ietf.org/rfc/rfc2616.txt.

[2] ETSI TS 102 241: "UICC API for Java Card™".

[3] OMA "Smartcard -Web -Server Enable Architecture", OMA-AD-Smartcard-Web-Server-V1-0-20070209-C.

[4] OMA "Smartcard-Web-Server", OMA-TS-Smartcard-Web-Server-V1-0-20070209-C.

[5] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Application Programming Interface".

[6] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Runtime Environment (JCRE) Specification".

[7] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Virtual Machine Specification".

NOTE: SUN Java Card Specifications can be downloaded at http://java.sun.com/products/javacard .

[8] "Smart Cards; ETSI numbering system for telecommunication application providers".

3 Abbreviations

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

AID	Application IDentifier
API	Application Program Interface
CAT	Card Application Toolkit
FFS	For Further Study
JCRE	Java Card™ Run-time Environment
Http	HyperText Transfer Protocol
HTTP	HyperText Transfer Protocol
SCWS	Smart Card based Web Server according to OMA specifications [3] and [4]

4 Description

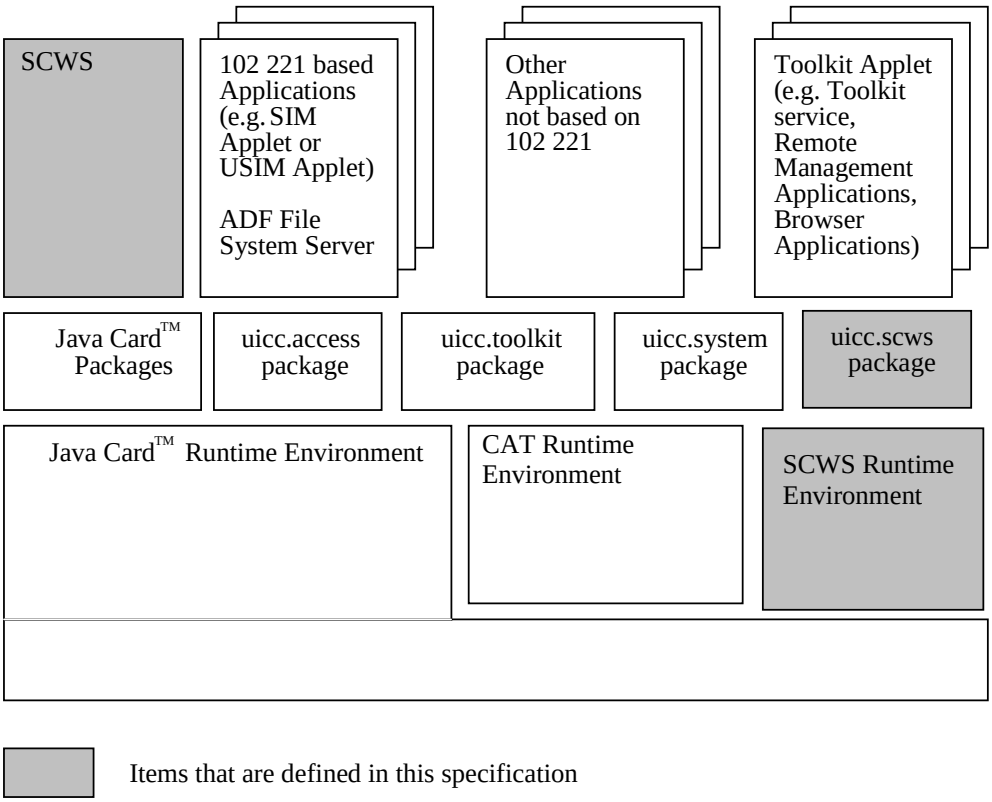
4.1 Architecture

The present document describes an API and a SCWS Runtime Environment that enables Java Card™ platform based applets, defined in [5], [6], [7], to register to and unregister from an SCWS implemented in the UICC, defined by OMA in [3] and [4].

The API enables a registered Applet to receive an incoming Http request that is forwarded by the SCWS. The API provides the necessary methods to allow registered Applets to respond with a correctly formatted Http response to the SCWS. The API provides means to the Applet to access the Http header data and the content of the Http request, to send specific Http status values, and to set the content of the Http response.

The Http request and response are defined in the Hypertext Transfer Protocol - HTTP/1.1 [1].

This API allows application programmers to extend the functionality of the SCWS defined by OMA in [3] and [4].



Smartcard Webserver (SCWS): handles Http request as defined by OMA in [3] and [4] and provides a mechanism to the Applet for the registration.

SCWS Runtime Environment: Extensions to the Java Card™ platform described in [5], [6], [7] and the CAT Runtime Environment described in TS 102 241 [2] to facilitate the communications between Applets and the SCWS.

Applet: these derive from *javacard.framework.Applet* and provide the entry points: *process, select, deselect, install* as defined in the "Java Card™ 2.2.2 Runtime Environment Specification" [6].

Registry of the SCWS: is provided as a JCRE entry point object defined in [6], and provides an interface to the Applet to pass a name to the SCWS for registration and deregistration. The registry is part of the SCWS Runtime Environment

SCWS API: consists of the package *uicc.scws*, provides the methods to register and deregister, to receive Http requests and to provide the content of the Http response.

4.2 Registration and deregistration

The registration of Applets to the SCWS enables the server to invoke a specific applet when it has received an Http request. Applet Instances can register with a name to the SCWS.

The mapping of the Http request to the name of the applet is described by OMA in [3] and [4] by the use of administrative commands {[FFS] other non-Http based mechanism.}. It is not possible to register several Applets under the same name to the SCWS. It is possible for an Applet to register several times with different names to the SCWS.

The Applet can also deregister from the SCWS. When the Applet deregisters from the SCWS the mapping information is deleted from the registry.

If an Applet is deleted then the registration information in the SCWS Registry is deleted by the SCWS Runtime Environment.

If the Applet is in a non selectable state, its registration to the SCWS is still valid.

4.3 Invocation

The SCWS invokes the Applets according to the mapping information when the complete Http request has been received by the SCWS.

Only an Applet that is in selectable state can be invoked by the SCWS.

If Applet execution ends without any invocation of the flush() method and without throwing an exception the SCWS shall finalize the response and send it.

Exceptions thrown by the invoked Applet shall not be propagated to the terminal, and the SCWS shall send an error status code according to HTTP 1.1 [1].

4.4 Transfer of response data

There are two transfer modes defined for the SCWS API: "fixed buffer size mode" and "chunked mode".

The API offers a method to switch between transfer modes. This method must be called before calling *finalizeHeader()* and before the first call of *appendContent()*.

The default transfer mode is "fixed buffer size".

The header attributes ("Content-Length: xxx" and "Transfer-Encoding: chunked") will be set according to the active transfer mode by the SCWS runtime environment. The Application is not supposed to set these attributes.

The SCWS runtime environment is not required to enforce this policy. The behaviour of the SCWS runtime environment is undefined if the application manipulates the header attributes for content length and transfer encoding.

In "fixed buffer size mode" an exception will be thrown by *appendContent()* if the buffer size would be exceeded.

In "fixed buffer size mode" no data are sent out before the application has called the *flush()* method, subsequent calls are permitted but have no effect.

In "chunked mode" a call of *flush()* sends all data in the response buffer. If there are no data in the response buffer no data will be sent.

If a call of *appendContent()* exceeds the buffer size in "chunked mode" the data in the response buffer will be sent implicitly.

Annex A (normative): Application invocation API by a UICC Webserver for the Java Card™ platform

The source files for the (102588_Annex_A_Java.zip and 102588_Annex_A_HTML.zip) are contained in ts_102588p0.zip, which accompanies the present document.

Annex B (normative): Application invocation API by a UICC Webserver for the Java Card™ platform

The export files for the uicc.scws package (102588_Annex_B_Export_Files.zip) are contained in ts_102588p0.zip, which accompanies the present document.

NOTE: See the "Java Card™ 2.2.2 Virtual Machine Specification" [7].

Annex C (normative): Application invocation API by a UICC Webserver for the Java Card™ platform package version management

Table C.1 describes the relationship between each TS 102 588 specification version and its packages AID and Major, Minor versions defined in the export files.

Table C.1

TS 102 588	uicc.scws package	
	AID	Major, Minor
	A0 00 00 00 09 00 05 FF FF FF FF 89 14 00 00 00	1.0

The package AID coding is defined in TS 101 220 [8]. The uicc.scws package AID is not modified by changes to Major or Minor Version.

The Major Version shall be incremented if a change to the specification introduces byte code incompatibility with the previous version.

History

Document history		
V7.0.0	July 2007	Publication