ETSI TS 103 999-1 V15.0.0 (2021-09)

Smart Secure Platform (SSP);
Part 1: Test Specification, general characteristics
(Release 15)

Reference
DTS/SCP-0000TSSPvf00-1

Keywords
SSP, testing

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

List of figures

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

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.

Introduction

The present document defines tests for the SSP implementations defined in ETSI TS 103 666-1 [1] independently of the respective manufacturer.

1 Scope

The present document specifies the test descriptions, test environment and conformance requirements for services running in the Smart Secure Platform and in any terminal hosting a Smart Secure Platform application.

The present document specifies the test descriptions for:

- SSP characteristics
- Physical interfaces
- SSP common layer
- Secure SCL
- Communication layers above SCL

of the SSP.

Tests for the usage or an SSP different to what is defined in ETSI TS 103 666-1 [1] are out of scope of the present document.

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.

- 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.

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

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 103 666-1: "Smart Secure Platform (SSP); Part 1: General characteristics".

[2] ETSI TS 102 230-1: "Smart Cards; UICC-Terminal interface; Physical, electrical and logical test specification; Part 1: Terminal features".

[3] ETSI TS 102 695-1: "Smart Cards; Test specification for the Host Controller Interface (HCI); Part 1: Terminal features".

[4] ETSI TS 102 695-2: "Smart Cards; Test specification for the Host Controller Interface (HCI); Part 2: UICC features".

[5] ETSI TS 102 622: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)".

[6] ETSI TS 102 230-2: "Smart Cards; UICC-Terminal interface; Physical, electrical and logical test specification; Part 2: UICC features".

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

[8] ISO/IEC 7816-4: "Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange".

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

[10] GlobalPlatform™: "Technology Virtual Primary Platform " Version 1.0.1.

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

[12] ETSI TS 102 671: "Smart Cards; Machine to Machine UICC; Physical and logical characteristics".

[13] ISO/IEC 7816-3: "Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols".

[14] ETSI TS 102 241: "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card™".

[15] ORACLE: "Application Programming Interface, Java Card™ Platform, Classic Edition 3.0.5".

[16] ORACLE: "Runtime Environment Specification, Java Card™ Platform, Classic Edition 3.0.5".

[17] ORACLE: "Virtual Machine Specification, Java Card™ Platform, Classic Edition 3.0.5".

NOTE: ORACLE Java Card™ Specifications can be downloaded at https://docs.oracle.com/javacard/3.0.5/index.html.

[18] IETF RFC 3629: "UTF-8, a transformation format of ISO 10646".

[19] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

[20] IETF RFC 8141: "Uniform Resource Names (URNs)".

[21] ETSI TS 102 705: "Smart Cards; UICC Application Programming Interface for Java Card™ for Contactless Applications".

[22] GlobalPlatform™: "Card Specification" Version 2.3.1.

NOTE: Available at https://globalplatform.org/specs-library/card-specification-v2-3-1/.

[23] GlobalPlatform™: "UICC Configuration" Version 2.0.

[24] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".

[25] ETSI TS 103 713: "Smart Secure Platform (SSP); SPI interface".

[26] IETF RFC 793: "Transmission Control Protocol".

[27] IETF RFC 792: "Internet Control Message Protocol".

[28] IETF RFC 6895: "Domain Name System (DNS) IANA Considerations".

[29] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[30] ANSI X9.63: "Public Key Cryptography for the Financial Services Industry Key Agreement and Key Transport Using Elliptic Curve Cryptography".

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.

[i.1] SCP SSP tooling.

NOTE: Available at https://forge.etsi.org/rep/scp/ssp-x509v3-generator/.

[i.2] ETSI TS 103 813: "Smart Secure Platform (SSP); Test Specification, SPI interface".

[i.3] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Physical and data link layer characteristics".

[i.4] IETF RFC 8615: "Well-Known Uniform Resource Identifiers (URIs)".

[i.5] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[i.6] IETF RFC 2818: "HTTP Over TLS".

[i.7] IETF RFC 7252: "The Constrained Application Protocol (CoAP)".

[i.8] ETSI TS 103 465: "Smart Secure Platform (SSP); Requirements Specification".

[i.9] ISO/IEC 9646-7:1995: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 103 465 [i.8] and ETSI TS 103 666-1 [1] apply.

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI TS 103 465 [i.8] and ETSI TS 103 666-1 [1] apply.

3.3 Abbreviations

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

AA	Accessor Authentication
AAA	Accessor Authentication Application
AAS	Accessor Authentication Service
AAUTH	Accessor AUTHentication
ACL	Access Control List
AID	Application IDentifier
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
ATR	Answer To Reset
CA	Certificate Authority
C-APDU	Command - APDU
CAT	Card Application Toolkit
CB	Chaining Bit
CI	Certificate Issuer
CLA	CLAss
CLF	ContactLess Frontend
CLK	CLocK
CLT	ContactLess Tunnelling
CPU	Central Processing Unit

CRON	Command Run ON
CSS	Cascading Style Sheets
DER	Distinguished Encoding Rule
DF	Dedicated File
DNS	Domain Name System
ECC	Elliptic Curve Cryptography
ECDHE	Elliptic Curve Diffie Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signal Algorithm
EEPROM	Electrically Erasable Programmable Read Only Memory
EF	Elementary File
FCI	File Control Information
FCP	File Control Parameters
FFS	For Further Study
FMD	File Management Data
FQDN	Fully Qualified Domain Name
FS	File System
FSCA	File System Control Application
FSCS	File System Control Service
FSDA	File System Data Application
FSDS	File System Data Service
GCM	Galois/Counter Mode
HCI	Host Controller Interface
HCP	Host Controller Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
I2C	Inter-Integrated Circuit
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
INS	INStruction
IP	Internet Protocol
ISO	International Organization for Standardization
KDF	Key Derivation Function
LAN	Local Area Network
MBM	Mobile Broadband Modem
MTU	Maximum Transfer Unit
NAA	Network Access Application
NFC	Near Field Communication
NID	Namespace Identifier
NOK	Not OK
NSS	Namespace Specific String
NVM	Non-Volatile Memory
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OOS	Out Of Scope
P1	Parameter 1
P2	Parameter 2
PIN	Personal Identification Number
PK	Public Key
PL	Padding Length
PPS	Protocol and Parameter Selection
RAM	Random Access Memory
RE	Runtime Environment
REE	Rich operating system Execution Environment
RFC	Request For Comments
RFU	Reserved for Future Use
RNG	Random Number Generator
RO	Read-Only
ROM	Read-Only Memory
RQ	ReQuirement
RSET	ReSET

RST	ReSeT
SCL	SSP Common Layer
SCP	Smart Card Platform
SHDLC	Simplified High Level Data Link Control
SI	SharedInfo
SoC	System on Chip
SPB	Secondary Platform Bundle
SPI	Serial Peripheral Interface
SSP	Smart Secure Platform
SSPFS	Smart Secure Platform File System
SSPUI	SSP User Interface
SUT	System Under Test
SWP	Single Wire Protocol
TCP	Transmission Control Protocol
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TLV	Tag Length Value
TRE	Tamper Resistant Element
UDP	User Datagram Protocol
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
USB	Universal Serial Bus
UTF	Universal character set Transformation Format
UUID	Universally Unique IDentifier
VNP	VPP Network Protocol
VPP	Virtual Primary Platform
XOR	eXclusive OR
WAN	Wide Area Network

3.4 Formats

3.4.1 Format of the table of optional features: Table 4.1

The columns in the optional features table, Table 4.1, have the following meaning:

Column	Meaning
Item	Item number, incrementing with each item added to the table
Service	Description of the service that might be supported by the implementation
Status	The status of the service is described following notations defined in ISO/IEC 9646-7 [i.9]: O optional - the service may be supported or not (default value)
Release	Number of the version the feature was introduced in
Support	The column is blank in the proforma and shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [i.9], are used for the support column in Table 4.1: Y or y supported by the implementation N or n not supported by the implementation N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status)
Mnemonic	The "Mnemonic" column contains mnemonic identifiers for each service

3.4.2 Format of the table of optional features: Table 4.2

The columns in the optional features table, Table 4.2, have the following meaning:

Column	Meaning
Item	Item number, incrementing with each item added to the table
Service Option/Optional Feature	Description of the service option, or optional feature that might be supported by the implementation
Status	The status of the service option / optional feature is described following notations defined in ISO/IEC 9646-7 [i.9]: O optional - the feature may be supported or not (default value) O.i qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table
Release	Number of the version the feature was introduced in
Support	The column is blank in the proforma and shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [i.9] are used for the support column in Table 4.1: Y or y supported by the implementation N or n not supported by the implementation N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status)
Mnemonic	The "Mnemonic" column contains mnemonic identifiers for each service option / optional feature

3.4.3 Format of the applicability Tables 4.3 and 4.4

The columns in the applicability tables, Table 4.3 and Table 4.4, have the following meaning:

Column	Meaning
Test Identification	A reference to the test identification(s), or range of test identifications detailed in the present document and required to validate the implementation of the corresponding item in the "Description" column
Description	A short non-exhaustive description of the test purpose is given here
Release	Number of the version the tested feature was introduced in
Rel-<x>	For a given Release, the corresponding "Rel-<x>" column lists the tests required for the SPI to be declared compliant to this Release Each entry shows the status following notations defined in ISO/IEC 9646-7 [i.9]: M mandatory - the capability is required to be supported O optional - the capability may be supported or not N/A not applicable - in the given context, it is impossible to use the capability X prohibited (excluded) - there is a requirement not to use this capability in the given context Oi qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table Ci conditional - the requirement on the capability ("M", "O", "X" or "N/A") depends on the support of other optional or conditional items. "i" is an integer identifying a unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE ...) ELSE ..." shall be used to avoid ambiguities
Support	Is blank in the proforma and is to be completed by the manufacturer in respect of each particular requirement to indicate the choices, which have been made in the implementation

3.4.4 Format of the conformance requirements tables

The columns in the requirement tables in clause 5 have the following meaning:

Column	Meaning
Req.ID	This column shows the ordinal term assigned to a requirement identified in the referenced specification. The following syntax has been used to define the unique R(equirement) terms: R<n><XX><YY>_<ZZZ> n: Identification letter for the referenced specification: Q: ETSI TS 103 666-1 [1] X: ETSI TS 102 221 [7] XX: Main clause of the core specification in which the conformance requirement is listed. YY: Sub-clause of the main clause in the core specification in which the conformance requirement is listed ZZZ: Continuously increasing number starting with '001'
Clause	The "Clause" column helps to identify the location of a requirement by listing the clause hierarchy down to the sub-clause the requirement is located in
Release	An optional column that is used if the listed requirement is valid for a specific release or a specific range of releases only, up to a specific release, or from a specific release onwards
Description	In this column the requirement text is shown. Where the text can either be a copy of the original requirement as found ETSI TS 103 666-1 [1] or ETSI TS 102 221 [7], or a text analogous to the requirement text (e.g.: if the requirement text is descriptive and can be shortened or truncated)

3.4.5 Numbers and Strings

The conventions used for decimal numbers, binary numbers and strings.

Table 3.1: Convention of Numbering and Strings

Convention	Description
nnnnn	A decimal number, e.g. PIN value or phone number
'b'	A single digit binary number
'bbbbbbbb'	An 8-bit binary number
'hh'	A single octet hexadecimal number
'hh hh…hh'	A multi-octet hexadecimal number or string
"SSSS"	A character string
NOTE: If an 'X' is present in a binary or hexadecimal number, then the digit might have any allowed value. This 'X' value does not need to be interpreted within the particular coding shown.	

3.4.6 Format of test description clauses

In general clauses with test descriptions use the following basic format:

X.Y. Group of test descriptions for a particular topic

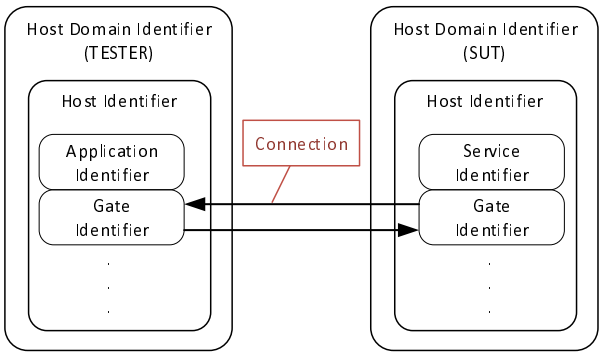
X.Y.1 Configurations

This header is used in every clause that includes configuration descriptions. It may be followed by a sentence explaining that there are no specific configurations required for this particular topic or:

X.Y.1.1 C<aaa>_<nnn> <optional>

Where each sub-header of a required configuration is built from a leading 'C' followed by <aaa>, a minimum three-digit abbreviation for the configuration description group, an underscore and <nnn>, a minimum three-digit number to identify the configuration. This sub-header may include explanatory text following the identification.

Whenever a configuration exists it is presented in a table of the following format.

Configuration ID	C<aaa>_<nnn>
Configuration description	Example: 

A Configuration description shows a drawing representing the entities involved and the connections available between instances. It does not include explanatory text.

X.Y.2 Procedures

This header is used in every clause that includes procedure descriptions. It may be followed by a sentence explaining that there are no specific procedures required for this particular topic or:

X.Y.2.1 P<aaa>_<nnn> <optional>

Where each sub-header of a required procedure is built from a leading 'P' followed by <aaa>, a minimum three-digit abbreviation for the procedure description group, an underscore and <nnn>, a minimum three-digit number to identify the procedure. This sub-header may include explanatory text following the identification.

Whenever a procedure exists it is presented in a table of the following format.

Procedure ID	P<aaa>_<nnn>
Procedure objectives	Description of the procedure objectives.
Configuration reference	C<aaa>_<nnn> See note 1.
Initial conditions	
Text and/or list of procedure IDs identifying the initial conditions that need to be fulfilled before the procedure sequence defined in this table can be executed. See note 2.	
Procedure sequence	
Step	**Description**
1	Description of procedure step #1
…	…
n	Description of procedure step #n
NOTE 1: Reference to the appropriate configuration. NOTE 2: Procedure IDs can be referenced if the integration of existing procedure sequences can avoid required procedure steps duplication to achieve the initial conditions. Referenced procedures are intended to be executed in given order.	

Procedures are sequences that are executed to prepare specific initial conditions for a test. As such they do not include verifications of any requirements.

X.Y.3 Test descriptions

This header is used for every clause that includes test descriptions. It may be followed by:

X.Y.3.1 <aaa>_<nnn> <optional>

Where each sub-header of a test description is built from <aaa>, a minimum three-digit abbreviation for the test description group, an underscore and <nnn>, a minimum three-digit number to identify the test description. This sub-header may include explanatory text following the identification.

Whenever a test description exists it is presented in a table of the following format.

Test ID	<aaa>_<nnn>	
Test objectives	Description of the test objectives. See note 1.	
Configuration reference	C<aaa>_<nnn> See note 2.	
Initial conditions		
Text and/or list of procedure IDs identifying the initial conditions that need to be fulfilled before the test sequence defined in this table can be executed. See note 3.		
Test sequence		
Step	**Description**	**Req.ID**
1	Description of test step #1	
…	…	RQ<XX><YY>_<ZZZ>
n	Description of test step #n	
NOTE 1: The descriptions reflect the objectives of the requirements verified. NOTE 2: Reference to the appropriate configuration. NOTE 3: If possible the initial conditions for the test sequence are defined by existing procedures. Referenced procedures are intended to be executed in given order.		

Requirement IDs listed in the Req.ID are references to the requirements listed in clause 5.x of the present document. A requirement listed in the test sequence is handled as verified if the response related to the listed requirement has the expected contents. Req.IDs are always assigned to a response step.

If there are no test descriptions defined for a group of tests, but related requirements are available, an appropriate sub-clause informs about the status of the requirements. E.g.:

X.Y.3.Z Requirements not testable, implicitly verified or verified elsewhere

The header of this sub-clause is adjusted depending on which condition applies for the identified requirements.

Example text for requirements referenced from a different standardization body:

The following requirements identified in <XYZ> are not tested in accordance with the present document, as they are referencing requirements from a different standardization body (<NAME>): <XX><YY>_<ZZZ>, …

Example text for requirements implicitly tested:

The following requirements identified in <XYZ> are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified: <XX><YY>_<ZZZ>, …

Example text for requirements not tested:

The following requirements identified in <XYZ> are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible: <XX><YY>_<ZZZ>, …

The clause with explanatory text for the untested or implicitly tested requirements is the last sub-clause in the Test description clause. Nevertheless, it may be provided as the first sub-clause if no executable test sequences are defined.

The hierarchy given in this example structure is not fixed. If building sub-groups is useful this has been done on the appropriate level of the test description hierarchy. Furthermore, sub-groups for all the three main clauses (Configurations, Procedures, Test descriptions) have not been generated if adding a sub-group is not useful in any of these clauses.

E.g.: common Configurations on hierarchy level 3, common Procedures on hierarchy level 3 but subgroups for the test descriptions with a new group header on level 4 and the test descriptions on level 5.

3.4.7 Dynamic content validation in ASN1 structure

In certain test cases, dynamic content returned by the DUT (e.g. value within ASN.1 structure, signature, integer,…) is processed according to the following Textx grammar:

operations ::= '<' operation (logical_operator operation) '>'*

operation ::= operation_Identifier ' (' variable_identifier (',' parameter) ')'*

operation_identifier ::= 'STORE'|'REPLACE'|'COMPARE'|'ISFIELDNOTEXIST'

logical_operator ::= 'AND'|'OR'|'XOR'

*variable_identifier ::=([A-Z]|[a-z])+[0-9]**

where:

- Operation_identifier: is the identifier identifying the operation to perform on a dynamic content of aFieldName as:
  - STORE: store the dynamic content of an aFieldName into a test tool variable identified by a variable identifier
  - REPLACE: retrieve a variable identified by Variable_identifier and replace the content of aFieldName by the content of the variable
  - COMPARE: compare the content of aFieldName with the content of a variable and return True or False to the test tool. This operator requires one or more additional parameters. The parameters may be combined for ORing them. The parameters are as follow:
    - GT: the content of the aFieldName shall be strictly greater than the content of a variable
    - LS: the content of the aFieldName shall be strictly less than the content of a variable
    - EQ: the content of the aFieldName shall be equal to the content of a variable
    - DIF: the content of the aFieldName shall be different from the content of a variable
  - ISFIELDNOTEXIST: return true, if aFieldName field does not exist
- Variable_identifier: variable identifier managed by the test tool. The variable identifier shall consist only of a set of alphanumeric characters.

The operations are inserted within a comment associated to a field as follow:

aFieldName … /* operations */

For example:

aResponse SSPCapability ::= {

aSspRelease '0000'H, /*<COMPARE(aSSPRELEASE,GT,EQ)>*/

where

aSSPRELEASE VersionType::= '0F00'H /* <STORE(aSSPRELEASE)> */

4 Tests environment architecture

4.1 Overview

Figure 4.1 illustrates the overview of the architecture for the SSP test environment.

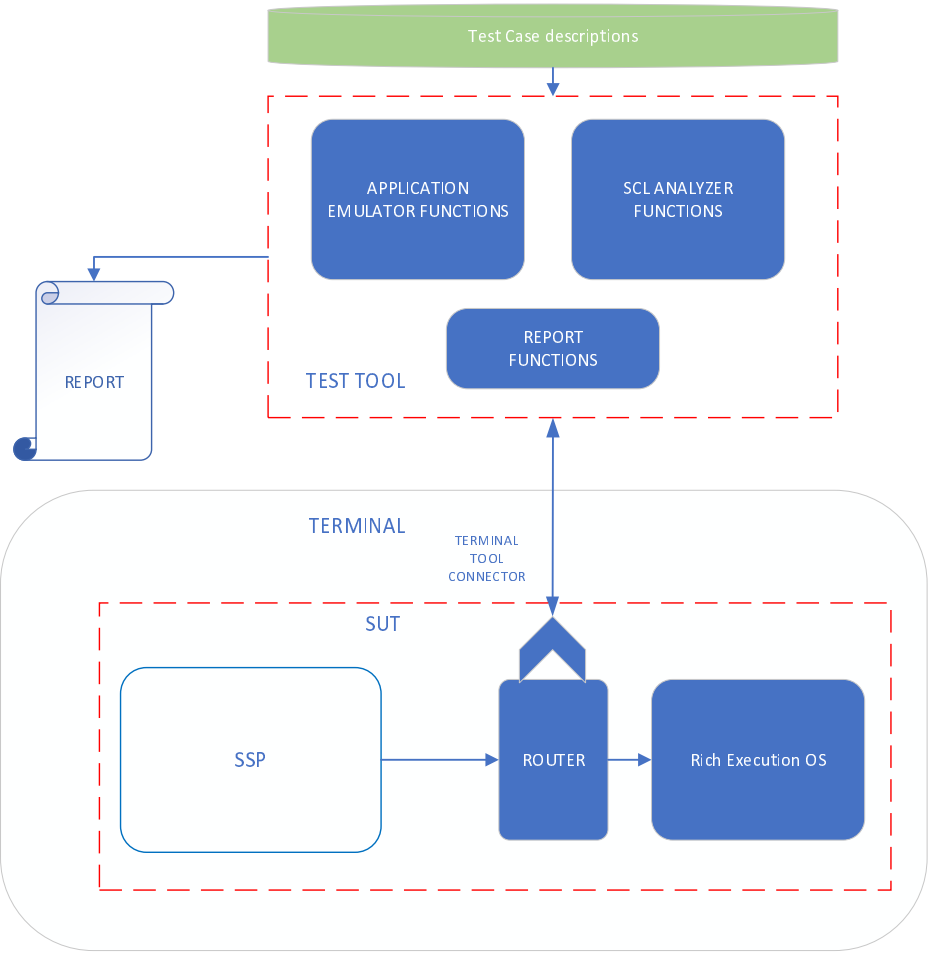


Figure 4.1: SSP test environment overview

The terminal shall contain:

- An SSP as defined in ETSI TS 103 666-1 [1].
- A router as defined in ETSI TS 103 666-1 [1].
- A Rich Execution Environment as defined in ETSI TS 103 666-1 [1].
- A terminal tool connector only available on a terminal prepared for test purposes. This terminal tool connector is mainly used to inject or extract SCL packets to/from the router. This terminal tool connector is provided by the terminal maker according to the requirements expressed in clause 4.2.1.

The test tool shall contain the following functions:

- An SCL analyser to analyse the SCL packets and to compare them with the test case expectations (which are based on test requirements).
- **For a test tool testing the services running in the SSP:** an application emulator for emulating an application running in the terminal. The tester may run multiple application emulators.

- **For a test tool testing the services running in the terminal:** an application emulator for emulating an application running in the SSP. The tester may run multiple application emulators.
- Report generator creating a report containing the verdicts based on test case outputs.

NOTE: Separate test tool implementations for terminal and SSP testing are permitted.

There are two perspectives of tests possible from Figure 4.1:

- The tests of a service running in the SSP. These tests require an emulator running a terminal application to stimulate the SSP.
- The tests of a service running in the terminal. These tests require an emulator running an SSP application to stimulate the service running in the terminal.

4.2 Test Tool Data exchange

4.2.1 Introduction

Figure 4.2 illustrates the data exchange between test tool and terminal test tool connector.

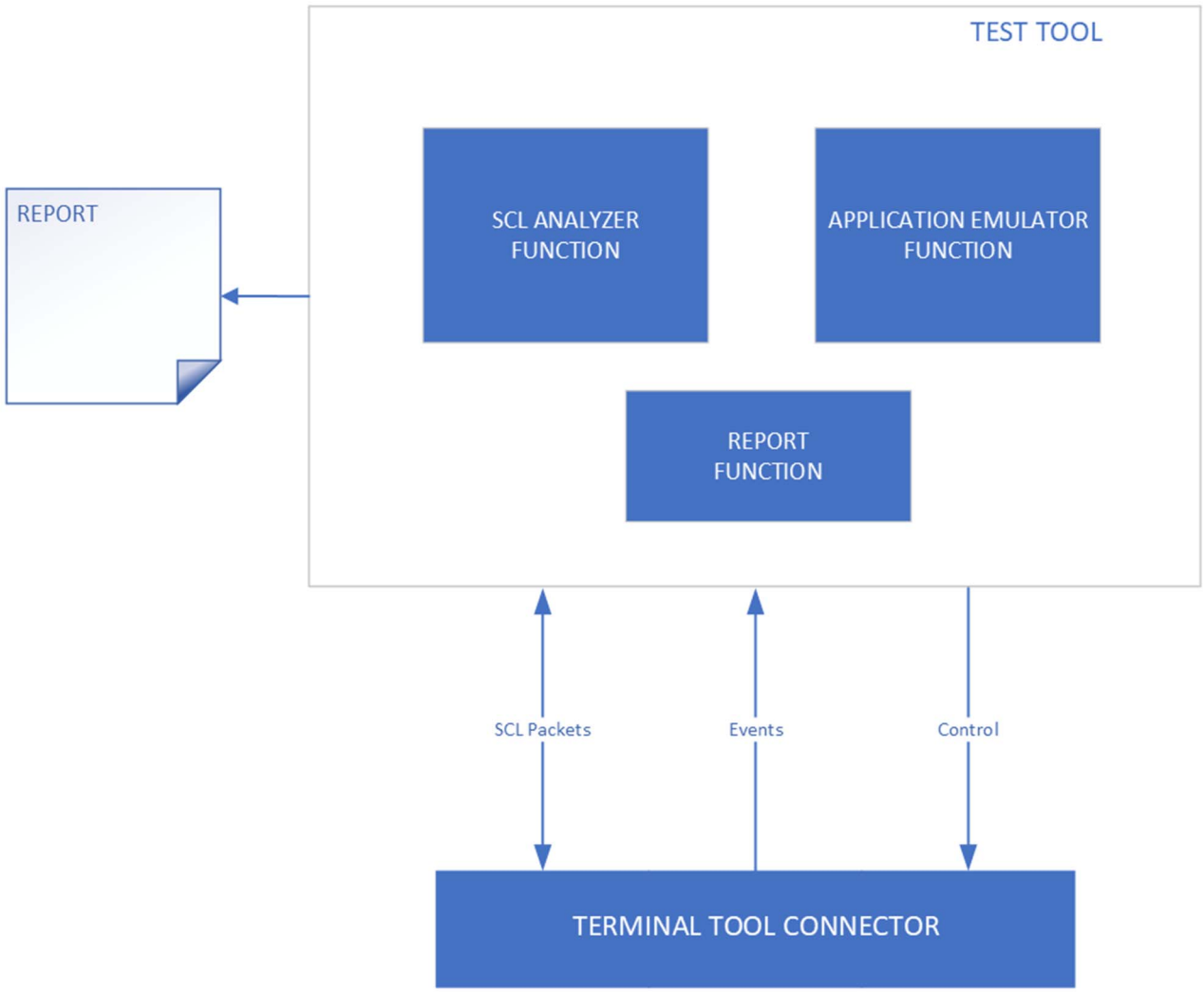


Figure 4.2: Data exchange between test tool and terminal test tool connector

The communication between the SUT and its environment is essentially based on the SCL network conveying the SCL packets.

4.2.2 Test tool requirements

The test tool shall be able to:

- extract the semantic from the SCL packets and compare it with the expected results extracted from a test case;
- emulate SSP/terminal applications to stimulate the SUT;
- collect the events from the SUT in order to get the state of the SCL host in the SUT;
- control the terminal tool connector according to directives from the tests;
- generate a report containing the verdicts based on test case expectations.

4.2.3 Terminal Test Tool connector requirements

The terminal test tool connector plugged into the router shall support the following requirements:

- It shall copy all SCL packets routed by the router into it, excluding the SCL packets from the SCL analyser.
- It shall timestamp the copy of the SCL packet.
- It shall be possible to disable the identification of the host issuing an SCL packet in order to impersonate it.
- It shall be possible to impersonate a host domain by a directive to the router.
- It shall be possible to collect events related to the SPB management (e.g. termination, exceptions, etc.).

4.3 Test of a service in the SSP

Figure 4.3 illustrates the perspective of the tests of a service running in the SSP from an application running on the terminal.

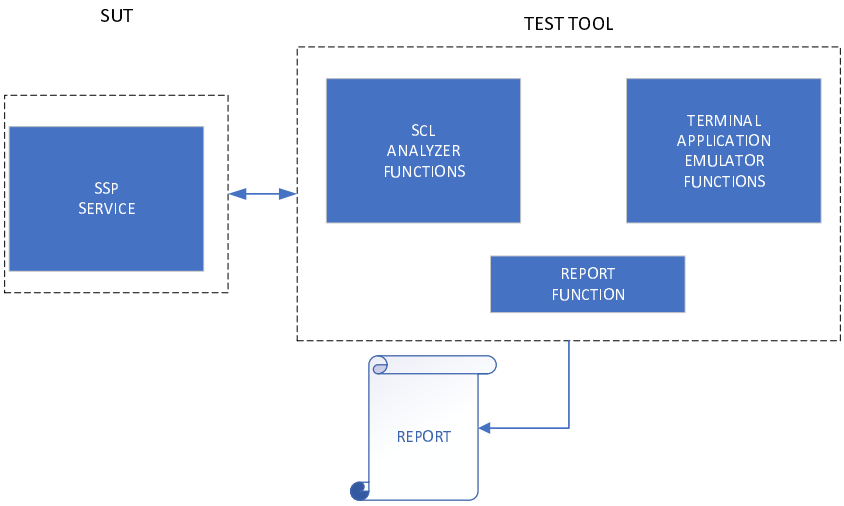


Figure 4.3: Tests of a service in the SSP

4.4 Test of a service in the terminal

Figure 4.4 illustrates the perspective of the tests of a service running in the terminal from an application running in the SSP.

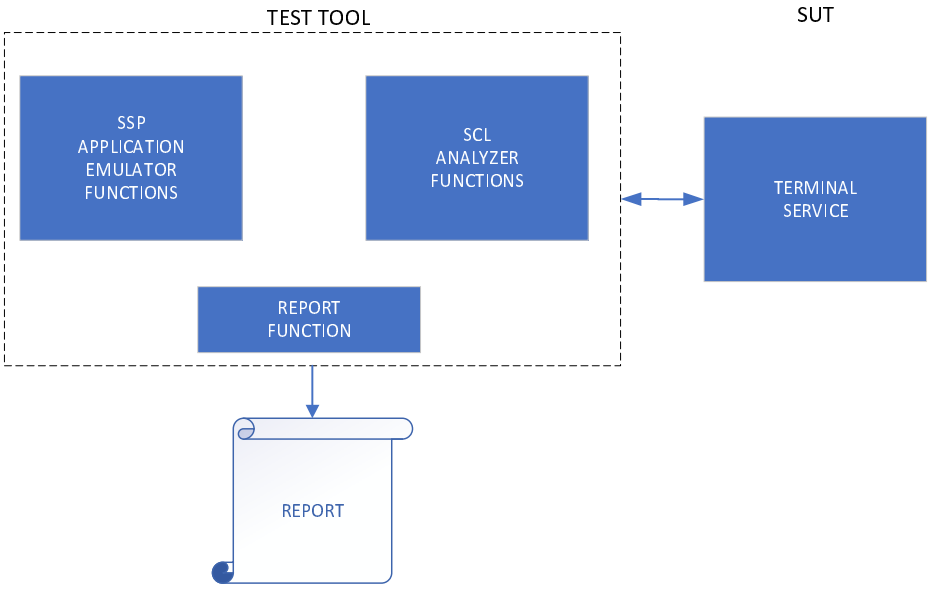


Figure 4.4: Tests of a service in the terminal

4.5 Table of services

The product vendor shall declare which services are supported by their implementation. The services are listed in Table 4.1.

See clause 3.4.1 for the format of the table.

Table 4.1: Table of optional services

Item	Service	Status	Release	Support	Mnemonic
1	UICC File System Service		Rel-15	O	O_UICC_FS
2	Card Application Toolkit Service		Rel-15	O	O_CAT
3	Accessor Authentication Service		Rel-15	O	O_AAUTH
4	SSP File System Service		Rel-15	O	O_SSP_FS
5	TCP Service		Rel-15	O	O_TCP
6	UDP Service		Rel-15	O	O_UDP
7	CRON Service		Rel-15	O	O_CRON
8	SCL HCI Service		Rel-15	O	O_SCL_HCI
9	HCP contactless		Rel-15	O	O_SCL_HCP
10	APDU Service		Rel-15	O	O_APDU

4.6 Table of service options and other optional features

The product vendor shall declare which service options and other optional features are supported by their implementation. The service options and optional features are listed in Table 4.2.

See clause 3.4.2 for the format of the table.

Table 4.2: Table of service options and optional features

Item	Service Option/Optional Feature	Status	Release	Support	Mnemonic
1	The Identity service gate returns GATE_URN_LIST (tag 81)		Rel-15	O	O_GATE_URN_LIST
2	Web-based user interface		Rel-15	O	O_SSPUI
3	Proactive polling is indicated as required		Rel-15	O	O_PROACTIVE_POLLING

4.7 Applicability table

The applicability tables in this clause are formatted as described in clause 3.4.3.

Table 4.3: Applicability table for SSP

Test Identification	Description	Release	Rel-15
FSS_0011 - FSS_0094	SSP File System	Rel-15	C004
INI_001	Capability Exchange of SSPCapabilities	Rel-15	M
SCL_031	Data-flow control in multiple hosts environment	Rel-15	M
SCL_032	loopback Data-flow control	Rel-15	M
SCL_033	Identity Service Gate parameter GATE_URN_LIST	Rel-15	C009
SCL_034	Link Service Gate additional registry entry	Rel-15	M
SCL_035	Credit based data flow control on administration gate	Rel-15	M
SSPUI_001	SSPCapabilities of SSPUI	Rel-15	C010
SSP_REF	Identified tests for SSP commands from ETSI TS 102 230-2 [6]	Rel-15	C001
LCH_REF	Identified tests for Logical channels from ETSI TS 102 230-2 [6]	Rel-15	C001
UFS_REF	Identified tests for UICC file system commands from ETSI TS 102 230-2 [6]	Rel-15	C001
ADD_REF	Identified tests for Additional commands from ETSI TS 102 230-2 [6]	Rel-15	C015
APDU_REF1	Identified tests for APDU transfer from ETSI TS 102 695-2 [4]	Rel-15	C016
APDU_REF2	Identified tests for APDU transfer from ETSI TS 102 695-2 [4]	Rel-15	C016

Table 4.4: Applicability table for Terminal

Test Identification	Description	Release	Rel-15
INI_002	Capability Exchange of TerminalCapabilities	Rel-15	M
TCP_311-TCP_312	Passive TCP Open TCP connection	Rel-15	C005
TCP_313-TCP_317	Passive TCP Open TCP connection LAN-WAN	Rel-15	C011
TCP_318-TCP_319	Passive TCP Open TCP connection	Rel-15	C005
TCP_3110	Passive TCP Open TCP connection IPV6	Rel-15	C011
TCP_321-TCP_322	Active TCP Open TCP connection	Rel-15	C005
TCP_323-TCP_324	Active TCP Open TCP connection	Rel-15	C011
TCP_331-TCP_333	Closing TCP connection	Rel-15	C005
TCP_341-TCP_342	Status TCP connection	Rel-15	C005
TCP_351	TCP data exchange	Rel-15	C005
TCP_361	Accept TCP connection	Rel-15	C005
TCP_371-TCP_3710	TCP events	Rel-15	C005
AAS_311-AAS-314	Root accessor authentication	Rel-15	C003
AAS_321-AAS-3210	Creation of an accessor PINCODE based	Rel-15	C013
AAS_331-AAS-335	Creation of an accessor password based	Rel-15	C013
AAS_341-AAS-348	Creation of an accessor pattern based	Rel-15	C013
AAS_351-AAS-353	Accessor capability	Rel-15	C005
AAS_361-AAS-369	Accessor update	Rel-15	C005
AAS_371	Accessor delete	Rel-15	C005
AAS_381	Anonymous accessor authentication	Rel-15	C005
AAS_391-AAS-393	Accessor group creation	Rel-15	C005
AAS_3101-AAS-3103	Accessor update	Rel-15	C005
SCL_031-SCL032	SCL test descriptions	Rel-15	C008
SCL_033- SCL_034	SCL URN registry	Rel-15	C014
SCL035	SCL data flow control	Rel-15	C008
SSL_031-SCL_034	Secure SCL	Rel-15	C003
UDP_031-UDP_033	UDP request socket	Rel-15	C006
UDP_041	UDP closing socket	Rel-15	C006
UDP_051-UDP_053	UDP socket datagram out	Rel-15	C006
UDP_061	UDP socket datagram in	Rel-15	C006
UDP_031-UDP_033	UDP request socket	Rel-15	C006
UDP_062	UDP socket events	Rel-15	C006
HCP_311-HCP_313	HCP contactless	Rel-15	C012

Table A.3: Execution clauses for applicability tables Table 4.3 and Table 4.4

C001	IF O_UICC_FS THEN M ELSE NA
C002	IF O_UICC_FS AND O_CAT THEN M ELSE NA
C003	IF O_AAS THEN M ELSE NA
C004	IF O_SSP_FS THEN M ELSE NA
C005	IF O_TCP THEN M ELSE NA
C006	IF O_UDP THEN M ELSE NA
C007	IF O_CRON THEN M ELSE NA
C008	IF O_SCL_HCI THEN M ELSE NA
C009	IF O_GATE_URN_LIST THEN M ELSE NA
C010	IF O_SSPUI THEN M ELSE NA
C011	IF O_TCP THEN O ELSE NA
C012	IF O_HCP THEN M ELSE NA
C013	IF O_AAS THEN O ELSE NA
C014	IF O_SCL_HCI THEN O ELSE NA
C015	IF O_UICC_FS AND O_CAT AND O_PROACTIVE_POLLING THEM M ELSE NA
C016	IF O_UICC_FS AND O_APDU THEN M ELSE N/A

5 Conformance requirements

5.0 Introduction

All references given in the conformance requirement descriptions are related to text, figures or tables provided in ETSI TS 103 666-1 [1].

5.1 SSP architecture

5.1.1 Overview

Reference: ETSI TS 103 666-1 [1], clause 5.1

RQ number	Clause	Description
RQ0501_001	5.1	The SSP is a secure element platform intended for use in a number of use cases which may have very different requirements. For that reason, the SSP is designed to be a modular platform offering a core set of features as well as a number of options that need to be selected at the time of implementation based on the intended use case.
RQ0501_002	5.1	SSP classes are defined in order to address these different use cases and in order to limit the possible configurations. An SSP class defines a configuration of the SSP platform.

5.1.2 SSP software architecture

Reference: ETSI TS 103 666-1 [1], clause 5.2

RQ number	Clause	Description
RQ0502_001	5.2	SSP Applications are programs running in the SSP.

5.1.3 SSP hardware architecture

Reference: ETSI TS 103 666-1 [1], clause 5.3

5.1.4 Protocol stacks

Reference: ETSI TS 103 666-1 [1], clause 5.4

RQ number	Clause	Description
RQ0504_001	5.4	The physical interface(s) between the SSP and the device might be selected from a range of options.
RQ0504_002	5.4	The SSP may have multiple physical interfaces.
RQ0504_003	5.4	The data link layer used over the physical interface might also be selected from a range of options.
RQ0504_004	5.4	The SSP should provide means for controlling (e.g. activating, deactivating) the data link and physical layers.
RQ0504_005	5.4	If indicated by the SSP class, the SSP shall support the SSP Common Layer (SCL) implementation comprised of optional network, transport and session layers, as described in clause 8.
RQ0504_006	5.4	If SSP Common Layer (SCL) is not supported, the SSP may support the UICC architecture as defined in ETSI TS 102 221 [7] and ETSI TS 102 622 [5].
RQ0504_007	5.4	An SSP implemented according to one of the existing form factors in ETSI TS 102 221 [7] and in ETSI TS 102 671 [12] shall support the ISO/IEC 7816-3 [13] interface and the transport of APDUs.
RQ0504_008	5.4	In addition, a mandatory core set of security features is provided, together with a number of optional security features which can be selected depending on the application.

5.1.5 Execution framework

Reference: ETSI TS 103 666-1 [1], clause 5.5

RQ number	Clause	Description
RQ0505_001	5.5	The optional or mandatory support of specific execution frameworks is defined for each specific SSP class.
RQ0505_002	5.5	The SSP may support an execution framework as defined for the UICC according to ETSI TS 102 241 [14] based on the Java Card™ Platform [15], [16] and [17].

5.2 SSP characteristics

5.2.1 Form factors

Reference: ETSI TS 103 666-1 [1], clause 6.1

RQ number	Clause	Description
RQ0601_001	6.1	The overall definition of the SSP is independent of the form factor, unless specified differently for a particular SSP class.

5.2.2 Power

Reference: ETSI TS 103 666-1 [1], clause 6.2

RQ number	Clause	Description
	6.2.1	Power mode
RQ0602_001	6.2.1	The following power modes are defined: • OPERATIONAL: when the SSP performs an internal process or processes incoming data from any of its interfaces. This mode also includes the transmission of data from and to the terminal. • SUSPENDED: the SSP does not consume any power, with the ability to resume the logical state at a later time (as described in clause 6.9 of ETSI TS 103 666-1 [1]). • IDLE: the SSP is in idle mode at any other time.
RQ0602_002	6.2.1	The power mode transition time is the maximum duration it takes the SSP to transition from one specific power mode, once SSP decided to, to another specific power mode.
	6.2.2	Power sources.
	6.2.2.1	Types of power sources.
RQ0602_003	6.2.2.1	The following power source types are defined for an SSP: • Interface: power to the SSP is provided by a communication interface according to its definition (e.g. ISO/IEC 7816-3 [13], USB). • Independent: power source which is not dependent on the power provided by any communication interface (e.g. dedicated power line).
RQ0602_004	6.2.2.1	The combined power sources shall provide sufficient power to operate the SSP in accordance with its power mode.
	6.2.2.2	Power source of type Interface.
RQ0602_005	6.2.2.2	Power provided by a communication interface is managed by the interface itself.
	6.2.2.3	Power source of type Independent.
RQ0602_006	6.2.2.3	The following voltage classes for a power source of type Independent are defined as follows, unless specified differently for an SSP class: • Class A: operational voltage class range is defined in Table 5.1 in ETSI TS 102 221 [7]. • Class B: operational voltage class range is defined in Table 5.5 in ETSI TS 102 221 [7]. • Class C: operational voltage class range is defined in Table 5.9 in ETSI TS 102 221 [7]. • Class P: operational voltage class range is proprietary and not defined in the present document (ETSI TS 103 666-1 [1]).
RQ0602_007	6.2.2.3	Supply voltage switching is outside the scope for power sources of type Independent.
RQ0602_008	6.2.2.3	Communication interfaces shall operate in relation to the voltage provided by the power source unless specified differently by the communication interface (e.g. ETSI TS 102 613 [i.3] operates at a fixed voltage level regardless of the supply voltage).
RQ0602_009	6.2.2.3	For reliable operation, the power source should meet the following characteristics: • When the power source is activated, the supply voltage should rise monotonically until reaching the operational voltage range. • The terminal should activate any communication interfaces only after the supply voltage has reached a stable level within the operational voltage range. • When the power source is deactivated, the supply voltage should fall monotonically until reaching 0 V ± 0,4 V referenced to ground.
RQ0602_010	6.2.2.3	Before activating the power source again, the supply voltage should remain at 0 V ± 0,4 V referenced to ground for at least 10 ms.
	6.2.3	Power consumption
RQ0602_011	6.2.3	The maximum power consumption is defined as the maximum amount of power used by the SSP when operating in OPERATIONAL power mode.
RQ0602_012	6.2.3	The overall power provided by the terminal to the SSP shall meet the power consumption of all active interfaces of the SSP and the internal power consumption of the SSP.
RQ0602_013	6.2.3	The maximum power consumption may be negotiated during the capability exchange procedure, as defined in clause 6.4.2 of ETSI TS 103 666-1 [1].

5.2.3 Clock

Reference: ETSI TS 103 666-1 [1], clause 6.3

RQ number	Clause	Description
RQ0603_001	6.3	The SSP shall have its own clock for the processing of all the commands, for the execution of its applications and for the access to its volatile and non-volatile memory, unless specified otherwise by the SSP class.
RQ0603_002	6.3	If a physical interface provides a clock (for example, the CLK like in the ISO/IEC 7816-3 [13] interface), this is independent from the internal clock of the SSP and shall not be used for internal processing, but only for the exchange of data over that interface.
RQ0603_003	6.3	The SSP shall make sure that its clock frequency does not cause power consumption in excess to what is negotiated with the terminal.
RQ0603_004	6.3	The SSP shall provide SSP applications with an interface to a time keeping mechanism, which measures elapsed time. The value obtained over this interface shall be based on the clock defined in this clause. Furthermore, this value shall be monotonic and increasing.

5.2.4 SSP initialization

Reference: ETSI TS 103 666-1 [1], clause 6.4

RQ number	Clause	Description
	6.4.1	SSP interface session
RQ0604_001	6.4.1	The SSP interface session begins when the physical interface and the data link layer are initialized, and the SSP is in a state where it can receive data from an end-point in the terminal or send data to an end-point in the terminal.
	6.4.2	Capability exchange
	6.4.2.1	Overall description
	6.4.2.2	SSP not supporting SCL
	6.4.2.3	SSP supporting SCL
RQ0604_002	6.4.2.3	If the UICC APDU gate described in clause 10.2.8.2 of ETSI TS 103 666-1 [1] is supported, then the capability exchange procedure shall be performed with the EXCHANGE CAPABILITIES command described in clause 10.2.3.2 of ETSI TS 103 666-1 [1].
RQ0604_003	6.4.2.3	In all other cases, the procedure should be performed when a new SCL host is registered on the SCL network controller host.
RQ0604_004	6.4.2.3	The procedure is performed by reading the parameter CAPABILITY_EXCHANGE as defined in clause 8.4.5.1.3 of ETSI TS 103 666-1 [1].
RQ0604_005	6.4.2.3	The capability exchange procedure is completed after the SCL host outside the SSP has read the CAPABILITY_EXCHANGE entry in the identity gate registry of the SCL host in the SSP and vice-versa.
	6.4.2.4	Capabilities of the terminal
RQ0604_006	6.4.2.4	Terminal release: it indicates the release of the present document that is implemented by the terminal. The major version shall have a value that is greater or equal to '0F' (which corresponds to Release 15, as the first release of the SSP).
RQ0604_007	6.4.2.4	Terminal vendor name: it indicates the terminal vendor's name encoded in UTF-8 format, as described in IETF RFC 3629 [18].
RQ0604_008	6.4.2.4	Interface power supply: it indicates the maximum current that the terminal can provide over the physical interface where the Capability Exchange procedure is performed. The value depends on the specific physical interface that is used. If the physical interface where the capability exchange procedure is performed does not provide power, value '0' is used. For the ISO/IEC 7816-3 [13] interface defined in clause 7.3 of ETSI TS 103 666-1 [1], the value indicates the maximum current in mA.
RQ0604_009	6.4.2.4	External power supply: it indicates the maximum current provided by the terminal using the external power supply. The value indicates the current in mA. The terminal shall use the same value on all the interfaces where the Capability Exchange procedure is performed. Value '0' is used when the external power supply is not present.
RQ0604_010	6.4.2.4	Toolkit terminal profile: it indicates the terminal profile used for the Card Application Toolkit. It is coded as defined in ETSI TS 102 223 [9], clause 5.2. If the TLV is absent, it means that the terminal does not support the Card Application Toolkit.
	6.4.2.5	Capabilities of the SSP

RQ number	Clause	Description
RQ0604_011	6.4.2.5	SSP release: it indicates the release of the present document (of ETSI TS 103 666-1 [1]) that is implemented by the SSP. The major version shall have a value that is greater or equal to '0F', which corresponds to Release 15 of ETSI TS 103 666-1 [1], as the first release of the SSP.
RQ0604_012	6.4.2.5	SSP vendor name: it indicates the SSP vendor's name encoded in UTF-8 format, as described in IETF RFC 3629 [18].
RQ0604_013	6.4.2.5	SSP class: it indicates the class of the SSP, as defined in clause 11 of ETSI TS 103 666-1 [1]
RQ0604_014	6.4.2.5	SSP class specific capabilities: it contains the SSP capabilities specific for the SSP class. The format is defined in the specification for that SSP class.
RQ0604_015	6.4.2.5	SSP UICC capabilities: it indicates the capabilities of the SSP to support features defined in the UICC platform: • Number of logical channels: it indicates the total number of logical channels, including the default channel, that is supported by the SSP. This value is specific for the interface where the command is exchanged and is applicable only when APDUs are used. It shall have a value between '01' and '14'. • Proactive polling requirement: it indicates if the terminal is required to perform the proactive polling, as described in clause 10.2.6.3 of ETSI TS 103 666-1 [1]. This value is specific for the interface where the command is exchanged and is applicable only when APDUs are used. If the value is FALSE, then the proactive polling is not required. In all other cases, this field shall have the value TRUE. • Support of the UICC file system: it indicates if the SSP supports the UICC file system, as described in clause 6.6.1 of ETSI TS 103 666-1 [1]. It shall have the value FALSE if the UICC file system is not supported, TRUE otherwise. • Support of Card Application Toolkit: it indicates if the SSP supports the Card Application Toolkit. It shall have the value FALSE if the Card Application Toolkit is not supported, TRUE otherwise. • Card Application Toolkit capabilities: it indicates the Card Application Toolkit procedures initiated by the terminal that the SSP supports. This field shall be present if the SSP indicates support of the Card Application Toolkit. It is coded as the value in the CAT service list data object defined in ETSI TS 102 223 [9], clause 8.102.

5.2.5 Storage

Reference: ETSI TS 103 666-1 [1], clause 6.5

RQ number	Clause	Description
RQ0605_001	6.5	Whether the NVM is within the SSP or external to the SSP is SSP class dependant. Consequently, the technical specification of each SSP class shall indicate if the NVM is allowed to be internal and/or external.
RQ0605_002	6.5	When the NVM is within the SSP, it shall be isolated and not be accessible outside the SSP.

5.2.6 Data Management

Reference: ETSI TS 103 666-1 [1], clause 6.6

RQ number	Clause	Description
	6.6.1	UICC file system
RQ0606_001	6.6.1	The SSP may support the UICC file system as specified in ETSI TS 102 221 [7], clause 8.1, clause 8.2 and clause 8.3, and the associated security features described in ETSI TS 102 221 [7], clause 9. The technical specification of each SSP class shall indicate if it is mandatory, optional or forbidden.
	6.6.2	SSP file system
	6.6.2.1	Overview
	6.6.2.2	SSP file system structure
	6.6.2.2.1	Layout
	6.6.2.2.2	Node types
RQ0606_002	6.6.2.2.2	SSP directory is a particular node that contains the list of references to other nodes and a reference to the parent directory.

RQ number	Clause	Description
RQ0606_003	6.6.2.2.2	SSP root directory is a particular node that contains the list of references to other nodes.
RQ0606_004	6.6.2.2.2	SSP file is a sequence of data bytes.
RQ0606_005	6.6.2.2.2	SSP link contains a link to an SSP file.
RQ0606_006	6.6.2.2.2	SSP link shall not link to an SSP directory (including SSP root directory), or to another SSP link.
	6.6.2.2.3	Node descriptor
RQ0606_007	6.6.2.2.3	The SSP file system shall allocate a node descriptor per node. The node descriptor shall be represented in ASN.1 syntax containing the following parameters: • aDirectory: it indicates that the type of the node is a SSP directory. • aFile: it indicates that the type of the node is a SSP file and its size in bytes. • aLink: it indicates that the type of the node is a SSP link, the size and the identity of the linked SSP file. • aMetaData: if present, it contains a collection of proprietary metadata with limited size. The content of the metadata of SSP links is the metadata of the linked file. • aACL: if present, it contains a collection of access control. If absent, the node inherits the access control list from its parent node.
RQ0606_008	6.6.2.2.3	The SSP file system shall support a tree of nodes with a minimum height of 5.
RQ0606_009	6.6.2.2.3	Each SSP directory shall support a minimum of 256 nodes.
	6.6.2.2.4	Node identity
RQ0606_010	6.6.2.2.4	All SSP files and SSP directories are referenced by a string, called node name.
RQ0606_011	6.6.2.2.4	The node name of SSP directories and SSP files shall use graphic characters, with a maximum length of 16 bytes after encoding in UTF-8 format, as described in IETF RFC 3629 [18].
RQ0606_012	6.6.2.2.4	The location of an SSP directory or of an SSP file in the hierarchical tree is described by a path.
RQ0606_013	6.6.2.2.4	A path in the hierarchical tree shall be described by a pathname. The pathname shall be a sequence of one or more node names of SSP directories concatenated by the node name separator, starting from the SSP root directory.
RQ0606_014	6.6.2.2.4	SSP file system shall support only the absolute pathname, starting from the root of the hierarchical tree.
RQ0606_015	6.6.2.2.4	The node reference shall be a string composed by a pathname followed by the node name separator and the node name. For example: "SSPFS:directory1:directory3" identifies the SSP directory 3.
RQ0606_016	6.6.2.2.4	All SSP files and SSP directories also have a short node name, which is the UUID version 5 calculated using the domain name system namespace, as defined in IETF RFC 4122 [19] from a URN, as defined in IETF RFC 8141 [20], composed concatenating "urn:etsi.org" (NID), the colon character (U+003A) and the Node reference (NSS).
RQ0606_017	6.6.2.2.4	The short node name may be used to access the node.
	6.6.2.2.5	File handling
RQ0606_018	6.6.2.2.5	An SSP file can be accessed (i.e. operated) by opening a session. The file session is referenced by a unique identifier called session ID that is provided as a response to the file session open command (FS-OP-FILE-OPEN-Service-Command).
RQ0606_019	6.6.2.2.5	The SSP file system shall support minimum two simultaneous file sessions.
RQ0606_020	6.6.2.2.5	Several file sessions may apply on the same SSP file.
RQ0606_021	6.6.2.2.5	A file session can be opened on an SSP file if the access conditions of the SSP file are satisfied.
	6.6.2.2.6	Administrative operations
RQ0606_022	6.6.2.2.6	The SSP file system supports retrieving the capabilities of the SSP file system (i.e. FS-ADMIN-GET-CAPABILITIES-Service-Command).
RQ0606_023	6.6.2.2.6	The SSP file system supports creating and deleting a node (i.e. FS-ADMIN-CREATE-NODE-Service-Command, FS-ADMIN-DELETE-NODE-Service-Command).
RQ0606_024	6.6.2.2.6	The SSP file system supports updating the attributes of a node (i.e. FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command).
	6.6.2.2.7	SSP file system access rights
RQ0606_025	6.6.2.2.7	eFSAccessRight-RequiresSecurePipe: this right indicates that, in addition to the permissions required to access the resource, the accessor shall use a secure pipe, as defined in clause 9 of ETSI TS 103 666-1 [1].
RQ0606_026	6.6.2.2.7	eFSAccessRight-ReadContent: in case of SSP file and SSP link, this right allows access to read the content. In case of SSP directory, this right allows access to the list of the contained nodes (if the command is allowed).
RQ0606_027	6.6.2.2.7	eFSAccessRight-GetInfo: this right allows access to retrieve information of a node.
RQ0606_028	6.6.2.2.7	eFSAccessRight-Write: in case of SSP file and SSP link, this right allows access to write the content. In case of SSP directory, this right allows creation of a node within the SSP directory.

RQ number	Clause	Description
RQ0606_029	6.6.2.2.7	eFSAccessRight-UpdateMetadata: this right allows the update of the metadata of the node.
RQ0606_030	6.6.2.2.7	eFSAccessRight-UpdateACL: this right allows the update of the access control list of the node.
RQ0606_031	6.6.2.2.7	eFSAccessRight-Delete: this right allows the deletion of the node.
RQ0606_032	6.6.2.2.7	eFSAccessRight-DeleteChild: this right allows the deletion of any node contained in the SSP directory, regardless of the value of eFSAccessRight-Delete of each contained node.
RQ0606_033	6.6.2.2.7	When SSP links are used for operations that access the content of nodes (i.e. FS-OP-FILE-OPEN-Service-Command) or for operations that access the metadata of nodes (i.e. FS-OP-NODE-GET-INFO-Service-Command and FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command), the SSP shall verify the access control list of both the SSP link and the linked SSP file.
	6.6.2.3	Primitives of the SSP file system
	6.6.2.3.1	FS-ADMIN-GET-CAPABILITIES-Service-Command
RQ0606_034	6.6.2.3.1	With the command FS-ADMIN-GET-CAPABILITIES-Service-Command, an SSP file system application requests the SSP file system service to retrieve the capabilities of the SSP file system. It has no parameters.
RQ0606_035	6.6.2.3.1	When the FS-ADMIN-GET-CAPABILITIES-Service-Command is successful, then the SSP file system service shall include eFS-OK in the response and the following parameters: • aVersion: major and minor release version supported by the file system control service gate; • aSimultaneousFileSessions: maximum number of simultaneous file sessions supported; • aSimultaneousFileSessionsPerFile: maximum number of simultaneous file sessions supported on the same file. This value shall be less or equal than aSimultaneousFileSessions; • aTotalCapacity: total capacity of the SSP file system in bytes; • aFreeCapacity: remaining free capacity in the SSP file system in bytes; • aMaxMetaDataSizePerNode: maximum metadata size allowed per node in bytes.
	6.6.2.3.2	FS-ADMIN-CREATE-NODE-Service-Command
RQ0606_036	6.6.2.3.2	With the command FS-ADMIN-CREATE-NODE-Service-Command, an SSP file system application may create an SSP file, an SSP directory or an SSP link within a hierarchical tree of SSP directories. It has the following parameters: • aNodeDescriptor: contains the node descriptor to create a node; • aNodeDirectoryIdentity: is the SSP Directory into which the new node shall be placed.
RQ0606_037	6.6.2.3.2	The accessor creating a node in a SSP directory shall have the eFSAccessRight-Write access rights on that SSP directory.
RQ0606_038	6.6.2.3.2	The SSP file system service shall ignore the short name included in aNodeDescriptor and compute it.
RQ0606_039	6.6.2.3.2	If the node descriptor indicates an SSP link, the SSP file system service shall ignore the file size and the metadata included in aNodeDescriptor, as the file size and the metadata are provided by the linked SSP file.
RQ0606_040	6.6.2.3.2	When the FS-ADMIN-CREATE-NODE-Service-Command is successful, then the SSP file system service shall include eFS-OK in the response.
	6.6.2.3.3	FS-ADMIN-DELETE-NODE-Service-Command
RQ0606_041	6.6.2.3.3	With the command FS-ADMIN-DELETE-NODE-Service-Command, an SSP file system application requests the SSP file system service to delete a node. It has the following parameter: • aNodeIdentity: identity of the node to be deleted.
RQ0606_042	6.6.2.3.3	An accessor is authorized to delete an SSP node if it has the eFSAccessRight-Delete right on the node to be deleted, or if it has the eFSAccessRight-DeleteChild right on the SSP directory containing the node.
RQ0606_043	6.6.2.3.3	The SSP file system shall reject the deletion of a node with the error eFS-NODE-BUSY if a session is ongoing on the node.
RQ0606_044	6.6.2.3.3	The deletion of an SSP directory implies the deletion of all the nodes contained in the SSP directory.
RQ0606_045	6.6.2.3.3	The deletion of a SSP link shall not impact the SSP file that is linked.
RQ0606_046	6.6.2.3.3	After the deletion of a node, all SSP links pointing to that node shall also be deleted by the SSP file system service, irrespective of the delete right to each SSP link.
RQ0606_047	6.6.2.3.3	After an SSP file is erased, it shall not be possible to restore its content.
RQ0606_048	6.6.2.3.3	When the FS-ADMIN-DELETE-NODE-Service-Command is successful, then the SSP file system service shall include eFS-OK in the response.

RQ number	Clause	Description
	6.6.2.3.4	FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command
RQ0606_049	6.6.2.3.4	With the command FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command, an SSP file system application requests the SSP file system service to update the access control and the metadata of a node. It has the following parameters: • aNodeIdentity: identity of the node to update; • aMetaData: the new meta data of the node; • aACL: the new access control list of the node.
RQ0606_050	6.6.2.3.4	The accessor updating the metadata of a node shall have the eFSAccessRight-UpdateMetadata right on that node.
RQ0606_051	6.6.2.3.4	If the update is performed on an SSP link, the accessor shall also have the eFSAccessRight-UpdateMetadata right on the linked node.
RQ0606_052	6.6.2.3.4	The accessor updating the access control list of a node shall have the eFSAccessRight-UpdateACL right on that node.
RQ0606_053	6.6.2.3.4	When the FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command is successful, then the SSP file system service shall include eFS-OK in the response.
	6.6.2.3.5	FS-OP-FILE-OPEN-Service-Command
RQ0606_054	6.6.2.3.5	With the command FS-OP-FILE-OPEN-Service-Command, an SSP file system application requests the SSP file system service to open a file session on a specified SSP file. It has the following parameters: • aNodeIdentity: identity of the node; • aAccessMode: the type of access to the SSP file; • aGateURI: the dynamic URI of the gate to open the pipe session for the SSP file system data gate linked to the opened SSP file for transferring the read or write data. This parameter shall be used only when the data is exchanged over a dedicated data pipe session.
RQ0606_055	6.6.2.3.5	The accessor opening a session on a SSP file or SSP link shall have the eFSAccessRight-ReadContent and/or the eFSAccessRight-Write right on that node depending on the access mode.
RQ0606_056	6.6.2.3.5	If the command is performed on an SSP link, the accessor shall also have the same right(s) on the linked node.
RQ0606_057	6.6.2.3.5	Opening a session on a file sets its current offset pointer to 0.
RQ0606_058	6.6.2.3.5	When the FS-OP-FILE-OPEN-Service-Command is successful, then SSP file system service shall include eFS-OK with following parameters in the response: • aSessionID: this is the session identifier to reference the SSP file for operation.
	6.6.2.3.6	FS-OP-FILE-CLOSE-Service-Command
RQ0606_059	6.6.2.3.6	With the command FS-OP-FILE-CLOSE-Service-Command, an SSP file system application requests the SSP file system service to close a specified file session opened by FS-OP-FILE-OPEN-Service-Command command. It has the following parameters: • aSessionID : this is the session identifier to the open SSP file.
RQ0606_060	6.6.2.3.6	If the SSP file system application sends a FS-OP-FILE-CLOSE-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall perform one of the following operations: • Terminate the ongoing command and close the ongoing session. • Reject the FS-OP-FILE-CLOSE-Service-Command command with the error eFS-NODE-BUSY.
RQ0606_061	6.6.2.3.6	When FS-OP-FILE-CLOSE-Service-Command is successful then SSP file system service shall include eFS-OK in the response.
RQ0606_062	6.6.2.3.6	If there is a pipe session associated with the aSessionID, the SSP file system application closes this pipe session.
	6.6.2.3.7	FS-OP-NODE-GET-INFO-Service-Command
RQ0606_063	6.6.2.3.7	With the command FS-OP-NODE-GET-INFO-Service-Command, an SSP file system applications requests the SSP file system service to read the information about an SSP file or an SSP directory. It has the following parameters: • aNodeIdentity: identity of the node; • aRequestType: indicates the type of the request.
RQ0606_064	6.6.2.3.7	The accessor retrieving the NodeDescriptor structure shall have the eFSAccessRight-GetInfo right on that node.
RQ0606_065	6.6.2.3.7	If the command is performed on an SSP link, the accessor shall also have the eFSAccessRight-GetInfo right on the linked node.
RQ0606_066	6.6.2.3.7	The accessor retrieving a NodeDescriptor structure list of child's node of an SSP directory (i.e. when aContain is set) shall have the eFSAccessRight-ReadContent right on that SSP directory.
RQ0606_067	6.6.2.3.7	When FS-OP-NODE-GET-INFO-Service-Command is successful, then the SSP file system service shall include eFS-OK with following optional parameter in the response: • aNodeDescriptorList: it contains the list of node descriptors requested by the SSP file system application. This list is limited to 255 node descriptors.

RQ number	Clause	Description
	6.6.2.3.8	FS-OP-FILE-READ-Service-Command
RQ0606_068	6.6.2.3.8	With the command FS-OP-FILE-READ-Service-Command an SSP file system application requests the SSP file system service to read the content of a SSP file that was previously opened with the command FS-OP-FILE-OPEN-Service-Command. It has the following parameters: • aSessionID: this is the session Identifier to reference the SSP file for operation; • aOffset: start position in the SSP file from offset 0. If omitted, read from the current offset of the SSP file; • aNumberOfBytes: number of byte to read. If set to 0, the whole SSP file shall be read out.
RQ0606_069	6.6.2.3.8	If the SSP file system application sends a FS-OP-FILE-READ-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-NODE-BUSY.
RQ0606_070	6.6.2.3.8	When FS-OP-FILE-READ-Service-Command is successful, then SSP file system service shall include eFS-OK with the following optional parameters in the response: • aData: data bytes read from the SSP file. This parameter is used only if the SSP file system application did not pass the gate URI when it opened the file session.
RQ0606_071	6.6.2.3.8	If the read data is received by the SSP file system application on a separate SCL pipe, then the FS-OP-FILE-READ-Service-Response is sent back to the SSP file system application on the same pipe as the FS-OP-FILE-READ-Service-Command after the last data byte has been received on the separate data channel.
	6.6.2.3.9	FS-OP-FILE-WRITE-Service-Command
RQ0606_072	6.6.2.3.9	With the command FS-OP-FILE-WRITE-Service-Command, an SSP file system application requests the SSP file system service to write data into an SSP file that was previously opened with the command FS-OP-FILE-OPEN-Service-Command. It has the following parameters: • aSessionID: this is the session Identifier to reference the SSP file for operation; • aOffset: start position in the SSP file from offset 0. If omitted, write from the current offset of the SSP file; • aNumberOfBytes: number of byte to write. The data shall be sent over a pipe session opened to a file system application data gate; • aData: the data buffer to write into the SSP file from the provided offset. It is recommended to use this option only for short data.
RQ0606_073	6.6.2.3.9	If the SSP file system application sends a FS-OP-FILE-WRITE-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-NODE-BUSY.
RQ0606_074	6.6.2.3.9	When FS-OP-FILE-WRITE-Service-Command is successful, then SSP file system service shall include eFS-OK in the response.
RQ0606_075	6.6.2.3.9	If the write data is sent by the SSP file system application on a separate channel, then the FS-OP-FILE-WRITE-Service-Response is sent back to the SSP file system application on the same pipe as the FS-OP-FILE-WRITE-Service-Command after the last data byte has been received on the separate data channel.
	6.6.2.3.10	FS-OP-FILE-GET-POSITION-Service-Command
RQ0606_076	6.6.2.3.10	With the command FS-OP-FILE-GET-POSITION-Service-Command, an SSP file system application requests to SSP file system service to retrieve the current offset position in an SSP file that was previously opened with the command FS-OP-FILE-OPEN-Service-Command. It has the following parameters: • aSessionID: this is the session Identifier to reference the SSP file for operation.
RQ0606_077	6.6.2.3.10	If the SSP file system application sends a FS-OP-FILE-GET-POSITION-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-NODE-BUSY.
RQ0606_078	6.6.2.3.10	When the FS-OP-FILE-GET-POSITION-Service-Command is successful then the SSP file system service shall include eFS-OK in the response with the following parameter: • aCurrentOffset: current offset of the SSP file.
	6.6.2.4	Response code
	6.6.2.4.1	Overview

RQ number	Clause	Description
RQ0606_079	6.6.2.4.1	The SSP file system service provides the following response codes to SSP file system primitives: • eFS-OK: Command completed successfully; • eFS-E-CMD-PAR-UNKNOWN: Format of the command parameters is wrong; • eFS-E-NOK: Command was rejected and/or not completed; • eFS-NODE-BUSY: The file system is already processing an operation on the file; • eFS-NODE-NOT-FOUND: Node not found; • eFS-OPERATION-ILLEGAL: Illegal operation (e.g. opening a file with a directory identity instead a file identity); • eFS-NOT-ENOUGH-SPACE: The operation exceeds the size limit of a file or the size limit of the metadata; • eFS-BAD-SESSSION-ID: The session identifier related to a file does not exist; • eFS-ACL-RULES-VIOLATIONS: The operation of the administration violates the ACL rules associated to a node; • eFS-MAX-FILE-SESSION-REACHED: The maximum number of file sessions has been reached. The possible response code returned for each primitives is shown in Table 6.3 of ETSI TS 103 666-1 [1].
	6.6.2.4.2	Response code to SSP file system primitives

5.2.7 SSP identification

Reference: ETSI TS 103 666-1 [1], clause 6.7

RQ number	Clause	Description
RQ0607_001	6.7	The SSP identification mechanism for the SSP is dependent on the SSP class and is specified for each class.

5.2.8 Runtime environment

Reference: ETSI TS 103 666-1 [1], clause 6.8

RQ number	Clause	Description
	6.8.1	CAT Runtime environment
RQ0608_001	6.8.1	If SSP supports the CAT Runtime Environment as specified in ETSI TS 102 241 [14] based on the Java Card™ Platform [15], [16] and [17], then clause 6.8.1 of ETSI TS 103 666-1 [1] shall apply.
RQ0608_002	6.8.1	If SSP supports CAT-RE, Card application toolkit specific fields in the capability exchange procedure indicate the support and the capabilities for the card application toolkit in the SSP.
RQ0608_003	6.8.1	If terminal supports CAT-RE, Card application toolkit specific fields in the capability exchange procedure indicate the support and the capabilities for the card application toolkit in the terminal.
RQ0608_004	6.8.1	If SSP implements SCL and supports UICC APDU service gate as described in ETSI TS 103 666-1 [1], clause 10.2.8.2, then CAT-RE shall send and receive APDUs as defined in ETSI TS 102 221 [7], via the UICC APDU service gate defined in ETSI TS 103 666-1 [1], clause 10.2.8.2.2.
RQ0608_005	6.8.1	If SSP implements SCL and supports UICC APDU service gate as described in ETSI TS 103 666-1 [1], clause 10.2.8.2, then CAT-RE shall Issue an EVT_TOOLKIT_REQUEST as defined in ETSI TS 103 666-1 [1], clause 10.2.8.2.3.3, if a proactive command has to be sent to the terminal.
RQ0608_006	6.8.1	If SSP implements SCL and supports UICC APDU service gate as described in ETSI TS 103 666-1 [1], clause 10.2.8.2, then CAT-RE shall map the SSP command EXCHANGE CAPABILITIES as defined in ETSI TS 103 666-1 [1], clause 10.2.3.2 to the events EVENT_PROFILE_DOWNLOAD and EVENT_FIRST_COMMAND_AFTER_ATR as defined in ETSI TS 102 241 [14].
RQ0608_007	6.8.1	If SSP implements SCL and supports CAT gate as described in ETSI TS 103 666-1 [1], clause 10.8, then CAT-RE shall Send and receive CAT commands and responses, via the CAT application gate.

RQ number	Clause	Description
RQ0608_008	6.8.1	If SSP implements SCL and supports CAT gate as described in ETSI TS 103 666-1 [1], clause 10.8, then CAT-RE shall Trigger the applets based on events received by the CAT application gate, replacing the APDU based triggering mechanism.
RQ0608_009	6.8.1	If SSP implements SCL and supports CAT gate as described in ETSI TS 103 666-1 [1], clause 10.8, then CAT-RE shall Map the capability exchange procedure to the events EVENT_PROFILE_DOWNLOAD and EVENT_FIRST_COMMAND_AFTER_ATR.
RQ0608_010	6.8.1	If SSP implement UICC file system, the events EVENT_EXTERNAL_FILE_UPDATE and EVENT_REMOTE_FILE_UPDATE shall be raised according to ETSI TS 102 241 [14] on update operations on the UICC file system.
RQ0608_011	6.8.1	If SSP is supporting the Contactless Framework as defined in ETSI TS 102 705 [21] based on the Java Card™ Platform [15] and if SSP implements SCL, the Contactless Framework shall register an HCI gate defined in ETSI TS 103 666-1 [1], clause 10.7.2.

5.2.9 SSP suspension

Reference: ETSI TS 103 666-1 [1], clause 6.9

RQ number	Clause	Description
RQ0609_001	6.9	The usage of the suspension mechanism by terminal is allowed only if the SSP has a single active physical interface.
RQ0609_002	6.9	When the SSP is suspended, the terminal deactivates the physical interface to the SSP, following the sequence specified for that physical interface
RQ0609_003	6.9	The suspension procedure can be used only when it is indicated as supported by the SSP in the capability exchange procedure.
RQ0609_004	6.9	If terminal suspends SSP, then terminal shall maintain the logical status as before the suspension and it shall resume the SSP for any event for which it had previously registered.
RQ0609_005	6.9	To resume the SSP, the terminal shall first perform the initialization of the SSP as described in ETSI TS 103 666-1 [1], clause 6.4, including the capability exchange procedure.
RQ0609_006	6.9	To resume the SSP, the electrical parameters shall remain unchanged during and after the resume operation.
RQ0609_007	6.9	If indicated as supported by the SSP in the capability exchange procedure, suspension is supported using APDU as defined in ETSI TS 103 666-1 [1], clause 10.2.7.
RQ0609_008	6.9	In case SCL is used, SSP suspension shall be rejected when there are more than 1 pipe (only pipe available is for transporting APDUs as defined in ETSI TS 103 666-1 [1], clause 10.2.8) to the SSP.

5.2.10 SSP applications

Reference: ETSI TS 103 666-1 [1], clause 6.10

RQ number	Clause	Description
	6.10.1	Overview
RQ0610_002	6.10.1	The SSP shall allow one or more SSP Applications to exchange data with other entities outside the SSP.
RQ0610_003	6.10.1	If there are no restrictions of the execution environment and/or of the application protocol, One SSP Application shall not block another SSP Application from exchanging data with the terminal on a different SSP interface session
RQ0610_004	6.10.1	If there are no restrictions of the execution environment and/or of the application protocol, One SSP Application shall not block another SSP Application from exchanging data with the terminal on the same SSP interface session, if supported by the protocol stack of the interface
	6.10.2	Ownership and security considerations
RQ0610_005	6.10.2	If the SSP implements the CAT Runtime Environment according to ETSI TS 102 241 [14], the rules and mechanisms for the management of Applications on the UICC shall apply, which are based on the GlobalPlatform Card Specification [22], its Amendments and the GlobalPlatform UICC Configuration [23] as described in ETSI TS 102 226 [24].
	6.10.3	Lifecycle management

RQ number	Clause	Description
RQ0610_006	6.10.3	If SSP Applications is running in the CAT Runtime Environment according to ETSI TS 102 241 [14], then the rules and mechanisms for the management of the lifecycle of Security Domains and Applications according to GlobalPlatform Card Specification [22] and ETSI TS 102 226 [24] shall apply.
	6.10.4	Identification and discovery

5.2.11 SSP security

Reference: ETSI TS 103 666-1 [1], clause 6.11

RQ number	Clause	Description
	6.11.1	SSP security architecture
RQ0611_001	6.11.1	The SSP is intended to provide a programmable, secure execution environment for applications.
RQ0611_002	6.11.1	Any entity external to the SSP shall not be able to directly access any hardware or software component within the SSP.
	6.11.2	Mandatory requirements
	6.11.2.1	Overview
	6.11.2.2	Security of SSP code
RQ0611_003	6.11.2.2	SSP shall provide confidentiality, integrity, and replay protection (i.e. ability to prevent outdated code from running on the same SSP and ability to prevent code of an SSP from running on another SSP) for any code executable inside the SSP.
RQ0611_004	6.11.2.2	Any SSP code shall be authenticated by the SSP entity that loads it.
	6.11.2.3	Privacy of data
	6.11.2.3.1	Secure storage
RQ0611_005	6.11.2.3.1	The SSP code and data shall be exclusively processed within the SSP.
RQ0611_006	6.11.2.3.1	The SSP code shall not be exposed outside the SSP in clear text.
RQ0611_007	6.11.2.3.1	The SSP data shall only be exposed outside the SSP under the control of the SSP.
RQ0611_008	6.11.2.3.1	If SSP code and data need to be stored outside the SSP, they shall be encrypted and integrity protected.
RQ0611_009	6.11.2.3.1	All the credentials used to encrypt the code and data shall only be stored and used within the SSP. The SSP shall depend only on its own cryptographic means.
RQ0611_010	6.11.2.3.1	The SSP shall implement mechanisms to prevent that an older version of the non-volatile storage can be re-used after it was superseded by a new SSP transaction.
	6.11.2.4	SSP transactions
RQ0611_011	6.11.2.4	An SSP transaction starts when the SSP receives a command to process and terminates when the SSP sends the response for that command. The transaction may be started by a command from the terminal, from the network or from an application running in the SSP itself.
RQ0611_012	6.11.2.4	If the status of the non-volatile memory needs to be modified after the execution of an SSP transaction, the SSP shall perform the update of the non-volatile memory before providing the response of the transaction to the client that initiated it. This includes the fact that it shall not be possible to restore the previous state of the non-volatile memory. If the NVM modification has not been successful for any reason, the previous content of NVM shall be restored.
	6.11.2.5	Attack resistance
RQ0611_013	6.11.2.5	The SSP shall be resistant to various attacks including but not limited to: • Side channel attacks such as simple power-analysis, differential power-analysis and timing analysis. Fault injection via voltage and clock frequency alterations, exposure to extreme light or temperatures. • Physical probing or tampering. • Injection via well-crafted input messages into the SSP. • Analysis through usage of test circuitry. The levels of resistance and attack prevention schemes are left to the specific SSP class.
	6.11.3	Optional requirements
	6.11.3.1	Overview
	6.11.3.2	Random number generator
RQ0611_014	6.11.3.2	An SSP may have its own Random Number Generator (RNG). The characteristics of the RNG depend on the SSP class and are defined in the corresponding specification.
	6.11.3.3	Remote provisioning

RQ number	Clause	Description
RQ0611_015	6.11.3.3	The SSP may include an optional secure mechanism in order to allow remote provisioning of its software components, including applications, part of or all the operating system. The mechanisms for the remote provisioning depend on the SSP class and are defined in the corresponding specification
	6.11.3.4	Remote auditing
RQ0611_016	6.11.3.4	Remote auditing is defined as the assessment of the integrity of the SSP hardware platform and optionally of some of the software components of the SSP by an entity outside the terminal. The assessment shall ensure with a coverage higher than 80 % that the SSP hardware platform and the optional software components have not changed since the reference SSP used for the certification.
RQ0611_017	6.11.3.4	Remote auditing process is optional. If supported: • The SSP class shall define an interface to the remote audit function of the SSP accessible from SSP Applications. • The SSP class may define an interface to the remote audit function of the SSP accessible from terminal. • The results of the remote audit function operations from the terminal interface shall be different than the ones collected from the SSP applications interface when using the same input parameters of the remote audit function.
	6.11.4	Security certification
	6.11.4.1	Overview
RQ0611_018	6.11.4.1	A certification process may be defined for each SSP class. These certification processes shall help the secure application provider to assess the level of trust it can give to the SSP and thus assess if its secure applications can be hosted by this particular SSP.

5.2.12 User interface

Reference: ETSI TS 103 666-1 [1], clause 6.12

RQ number	Clause	Description
	6.12.1	Web-based user interface
	6.12.1.1	Overview
RQ0612_01	6.12.1.1	If the SSP supports the web-based user interface, it shall: • indicate the URL to be used for the entry page in the capability exchange, as defined in clause 6.4.2.5 of ETSI TS 103 666-1 [1]; • support the SCL protocol, as defined in clause 8 of ETSI TS 103 666-1 [1]; • open a TCP server socket with local access only using the TCP control gate, as defined in clause 10.4 of ETSI TS 103 666-1 [1], using the same local port as indicated in the URL.
RQ0612_02	6.12.1.1	The web server in the SSP is accessed by the terminal using the URL retrieved during the exchange capability procedure, and using the TCP gates of the SCL protocol
	6.12.1.2	Port values
RQ0612_03	6.12.1.2	The SSP user interface should use the TCP port number 3516 for HTTP and the port 4116 for HTTP over TLS. Both ports are already reserved by IANA. Port 3516 is reserved as "smartcard Port" and port 4116 as "smartcard-TLS".
	6.12.1.3	Presentation of SSP user interface
RQ0612_04	6.12.1.3	The icon and corresponding text to indicate the availability of the SSP user interface to the user may be retrieved using the following URLs defined in the well-known URI format as defined in IETF RFC 8615 [i.4]: • Icon: SSP user interface URL as defined in clause 6.4.2.5 of ETSI TS 103 666-1 [1] followed by "/.well-known/icon.png". • Text: SSP user interface URL as defined in clause 6.4.2.5 of ETSI TS 103 666-1 [1] followed by "/.well-known/text". The text shall be encoded in UTF-8 as defined in IETF RFC 3629 [18].

5.2.13 Accessor authentication service

Reference: ETSI TS 103 666-1 [1], clause 6.13

RQ number	Clause	Description
	6.13.1.	Overview
RQ0613_001	6.13.1	Prior to access to a resource, an accessor shall authenticate itself based on some credentials with the accessor authentication service.
	6.13.2.	Access Control
	6.13.2.1	Overview
RQ0613_002	6.13.2.1	If an access control includes a grantor then the grantor shall expose an accessor identity
RQ0613_003	6.13.2.1	an operation shall be included in the list of operations allowed by the rights in the access control otherwise the operation is denied
RQ0613_004	6.13.2.1	When the grantor is present in the access control, the access to the resource is permitted to the accessor if both the accessor and the grantor are authenticated otherwise the access to the resource is denied
RQ0613_005	6.13.2.1	The rights provided in the access control are only for the accessor and are independent of any rights of the grantor.
	6.13.2.2	Description
RQ0613_006	6.13.2.2	The access control shall contain the accessor identity and the accessor rights
RQ0613_007	6.13.2.2	The access control may contain the grantor identity
	6.13.2.3	Accessor rights to a resource
RQ0613_008	6.13.2.3	The right on a resource shall apply only if the accessor has been successfully authenticated using the accessor authentication service.
	6.13.3.	Access Control List
RQ0613_009	6.13.3	An Access Control List (ACL) is a list of access controls which shall be formed using ASN1 notation as defined in clause 6.13.2 of ETSI TS 103 666-1 [1]
	6.13.4.	Accessor
	6.13.4.1	Overview
RQ0613_010	6.13.4.1	An accessor shall be either an AccessorGroup or an AccessorUser
RQ0613_011	6.13.4.1	An AccessorGroup shall contain an accessor identity (AccessorIdentity), the members of group as a SET OF accessor identity (aMembersOfGroup) and an access control list (AccessControlList)
RQ0613_012	6.13.4.1	An AccessorUser shall contain an accessor identity (AccessorIdentity) and an access control list (AccessControlList).
RQ0613_013	6.13.4.1	An AccessorUser may contain a list of accessor conditions (AccessorConditions)
RQ0613_014	6.13.4.1	All members of an AccessorGroup shall be of type AccessorUser
RQ0613_015	6.13.4.1	The authentication of one member of the group does not imply the authentication of the other members of the group
RQ0613_016	6.13.4.1	The authentication of an accessor shall verify the accessor conditions against the credentials
RQ0613_017	6.13.4.1	The operations on an accessor (e.g. delete or update the accessor) shall be allowed according to its ACL
	6.13.4.2	Anonymous accessor
RQ0613_018	6.13.4.2	The anonymous accessor shall be authenticated by the accessor authentication service
RQ0613_019	6.13.4.2	The authentication of an anonymous accessor shall not require any credentials
	6.13.4.3	Accessor identity
RQ0613_020	6.13.4.3	An accessor shall be identified by a UUID version 5.
RQ0613_021	6.13.4.3	the URN for computing the accessor UUID shall have "urn:etsi.org" as NID
RQ0613_022	6.13.4.3	the beginning of the NSS in the URN used for computing the accessor UUID shall be "SSP:ASN.1.
RQ0613_023	6.13.4.3	The accessor authentication service shall prevent that two accessors have the same accessor identity
	6.13.4.4	Accessor conditions
RQ0613_024	6.13.4.4	An accessor may have zero or more conditions described in AccessorConditions
RQ0613_025	6.13.4.4	The accessor conditions may contain an AccessorConditionsBiometry condition
RQ0613_026	6.13.4.4	The accessor conditions may contain an AccessorConditionsPIN condition
RQ0613_027	6.13.4.4	The accessor conditions may contain an AccessorConditionsToken condition
RQ0613_028	6.13.4.4	The accessor conditions may contain an AccessConditionHostDomain condition
RQ0613_029	6.13.4.4	The AccessorConditionsPIN shall only contain one of the following types: ePinNumeric, ePinPassword, ePinPattern.
RQ0613_030	6.13.4.4	The authentication using a credential inconsistent with the AccessorConditionsPIN shall fail.
RQ0613_031	6.13.4.4	The ePinNumeric indicates that the user shall present a numeric PIN
RQ0613_032	6.13.4.4	The ePinPassword indicates that the user shall present a password
RQ0613_033	6.13.4.4	The ePinPattern indicates that the user shall present a graphical pattern

RQ number	Clause	Description
RQ0613_034	6.13.4.4	The AccessorConditionsToken shall only propose an eTokenCertificate.
RQ0613_035	6.13.4.4	The AccessConditionHostDomain shall only propose a boolean
	6.13.4.5	Access rights
RQ0613_036	6.13.4.5	eAASAccessRight-RequiresSecurePipe access right in the ACL of the Accessor mandates the use of secure pipe
RQ0613_037	6.13.4.5	eAASAccessRight-Create access right in the ACL of the Accessor allows the creation of a new accessor
RQ0613_038	6.13.4.5	eAASAccessRight-Delete access right in the ACL of the Accessor allows the deletion of an existing accessor
RQ0613_039	6.13.4.5	eAASAccessRight-Update access right in the ACL of the Accessor allows the update of the ACL of an accessor
RQ0613_040	6.13.4.5	eAASAccessRight-UpdateACL access right in the ACL of the Accessor allows the update of the members of an accessor group
RQ0613_041	6.13.4.5	eAASAccessRight-UpdateGroup access right in the ACL of the Accessor allows the update of the conditions and credentials of the accessor
RQ0613_042	6.13.4.5	eAASAccessRight-UpdateCredentialPolicy access right in the ACL of the Accessor allows the update of the credential policy
RQ0613_043	6.13.4.5	eAASAccessRight-UpdateCredentialStatus access right in the ACL of the Accessor allows the update of status of credentials
	6.13.4.6	Operations on an accessor
	6.13.4.6.1	Creation
RQ0613_044	6.13.4.6.1	An accessor may be created by any other accessor that has the eAASAccessRight-Create right
	6.13.4.6.2	Deletion
RQ0613_045	6.13.4.6.2	An accessor may be deleted by another accessor that has the eAASAccessRight-Delete right
RQ0613_046	6.13.4.6.2	Upon deletion of an accessor, the SSP shall remove all entries in the access control lists of its resources pointing to the accessor.
	6.13.4.6.3	Update of the access control list
RQ0613_047	6.13.4.6.3	The access control list of an accessor may be updated by any accessor that has the eAASAccessRight-UpdateACL right
	6.13.4.6.4	Update of the conditions and credentials
RQ0613_048	6.13.4.6.4	The accessor conditions and the corresponding credentials may be updated by any accessor that has the eAASAccessRight-Update
RQ0613_049	6.13.4.6.4	An accessor shall not be able to modify its own accessor conditions or credentials, if it is not explicitly listed in its own access control list.
	6.13.4.6.5	Update of the group list
RQ0613_050	6.13.4.6.5	The list of members of an accessor group may be modified by any accessor that has the eAASAccessRight-UpdateGroup
	6.13.4.6.6	Update of the credential status and policy
RQ0613_051	6.13.4.6.6	The credential status of an accessor may be updated by any accessor that has the eAASAccessRight-UpdateCredentialStatus.
RQ0613_052	6.13.4.6.6	The credentials policy of an accessor may be updated by any accessor that has the eAASAccessRight-UpdateCredentialPolicy.
	6.13.4.7	Accessor credentials
RQ0613_053	6.13.4.7	The object AccessorCredentials is a collection of credentials and shall be represented with ASN.1 description as described at 6.13.4.7 in ETSI TS 103 666-1 [1]
RQ0613_054	6.13.4.7	aPinNumericCredential shall be a numeric string PIN
RQ0613_055	6.13.4.7	aPinNumericCredential shall not contain spaces
RQ0613_056	6.13.4.7	aPinPasswordCredential shall be a string containing a case-sensitive password.
RQ0613_057	6.13.4.7	aPinPatternCredential shall be a sequence of points
RQ0613_058	6.13.4.7	aPinPatternCredential: pattern drawn implementation shall support a width and the height of at least 3 points and at most 10 points.
RQ0613_059	6.13.4.7	aPinPatternCredential: the length of the pattern shall be between 4 and 255 points.
RQ0613_060	6.13.4.7	aPinPatternCredential: if allowed by the credential policy, the same point may appear more than once in the pattern
RQ0613_061	6.13.4.7	aCertificates shall be a set of X.509 certificates of the accessor
RQ0613_062	6.13.4.7	aHostDomainCredential shall be a list of SCL host domains.
	6.13.4.8	Accessor credential policy
RQ0613_063	6.13.4.8	AccessorCredentialsPolicy shall comply with ASN.1 description at 6.13.4.8 in ETSI TS 103 666-1 [1]
RQ0613_064	6.13.4.8	aPinNumericPolicy: aIsDisableForbiden shall indicate if PIN can be disabled
RQ0613_065	6.13.4.8	aPinNumericPolicy: aMinSize shall indicate the minimum size for PIN
RQ0613_066	6.13.4.8	aPinNumericPolicy:aMaxSize: shall indicate the maximum size for PIN, if not present, maximum size is limited to 255.

RQ number	Clause	Description
RQ0613_067	6.13.4.8	aPinNumericPolicy: aMaxAttempts shall indicate the maximum number of attempts allowed for the PIN. The value 0 indicates that an infinite number of attempts is allowed.
RQ0613_068	6.13.4.8	aPinPasswordPolicy: aMinSize shall indicate the minimum size for password.
RQ0613_069	6.13.4.8	aPinPasswordPolicy: aMaxSize shall indicate the maximum size for password, if not present, maximum size is limited to 255.
RQ0613_070	6.13.4.8	aPinPasswordPolicy: aRequiresLowerCaseLetter shall indicate if the password shall contain at least one lower case letter.
RQ0613_071	6.13.4.8	aPinPasswordPolicy: aRequiresUpperCaseLetter shall indicate if the password shall contain at least one upper case letter
RQ0613_072	6.13.4.8	aPinPasswordPolicy: aRequiresNumber shall indicate if the password shall contain at least one numeric digit (i.e. between '0' and '9').
RQ0613_073	6.13.4.8	aPinPasswordPolicy: aRequiresSymbol shall indicate if the password shall contain at least one symbol that is not a letter or a number
RQ0613_074	6.13.4.8	aPinPasswordPolicy: aMaxAttempts shall indicate the maximum number of attempts allowed for the password. The value 0 indicates that an infinite number of attempts is allowed.
RQ0613_075	6.13.4.8	aPinPatternPolicy: aMinSize shall indicate the minimum number of points in the pattern.
RQ0613_076	6.13.4.8	aPinPatternPolicy: aMaxSize shall indicate the maximum number of points in the pattern, if not present, maximum size is limited to 255.
RQ0613_077	6.13.4.8	aPinPatternPolicy: aEntryPanelMinSize shall indicate the minimum size of the width and the height of the pattern. The entry panel of the pattern may be a rectangular, as far as both sides have a size that is at least equal to aEntryPanelMinSize.
RQ0613_078	6.13.4.8	aPinPatternPolicy: aSamePointMultipleTimes shall indicate if the same point can appear multiple times in the pattern.
RQ0613_079	6.13.4.8	aPinPatternPolicy: aMaxAttempts shall indicate the maximum number of attempts allowed for the pattern. The value 0 indicates that an infinite number of attempts is allowed.
RQ0613_080	6.13.4.8	The credential of type host domain is not intended to be changed by the accessor and therefore has no defined policy.
RQ0613_081	6.13.4.8	The token-based credentials has no policy.
	6.13.4.9	Accessor credential status
RQ0613_082	6.13.4.9	AccessorCredentialsStatus shall comply with ASN.1 structure define at 6.13.4.9 in ETSI TS 103 666-1 [1]
RQ0613_083	6.13.4.9	aIsDisabled shall indicate if the related credential is disabled (authentication not needed)
RQ0613_084	6.13.4.9	aRemainingAttempts shall indicate the number of attempts remaining. 0 indicates that the credential is no more useable, no presence indicates that no maximum number of retry is defined.
RQ0613_085	6.13.4.9	The credential of type host domain has no status.
RQ0613_086	6.13.4.9	The token-based credential has no status.
	6.13.5.	Primitives of the access control
	6.13.5.1	AAS-OP-GET-CAPABILITIES-Service-Command
RQ0613_087	6.13.5.1	The accessor authentication service shall support the command AAS-OP-GET-CAPABILITIES-Service-Command as defined by ASN.1 in clause 6.13.5.1 of ETSI TS 103 666-1 [1]
RQ0613_088	6.13.5.1	Accessor authentication service shall support the AAS-OP-GET-CAPABILITIES-Service-Command command with parameter eGlobalAuthenticationService
RQ0613_089	6.13.5.1	Accessor authentication service shall support the AAS-OP-GET-CAPABILITIES-Service-Command command with parameter eAccessorStatus
RQ0613_090	6.13.5.1	When the AAS-OP-GET-CAPABILITIES-Service-Command request is successful then accessor authentication service gate shall include eAAS-OK in the response as defined by ASN.1 in clause 6.13.5.1 of ETSI TS 103 666-1 [1]
RQ0613_091	6.13.5.1	In response, aGlobalAuthenticationService: aAASVersion shall indicate major and minor release version supported by the accessor authentication service
RQ0613_092	6.13.5.1	In response, aGlobalAuthenticationService: aAccessorList shall indicate the list of all the accessors available in the SSP host
RQ0613_093	6.13.5.1	In response, aGlobalAuthenticationService: aACL shall indicate the access control list of the accessor authentication service.
RQ0613_094	6.13.5.1	In response, aAccessorStatus: aIsAuthenticated shall indicate if the accessor is authenticated in this accessor authentication service
RQ0613_095	6.13.5.1	In response, aAccessorStatus: aAccessorConditions shall indicate the accessor conditions to be authenticated
RQ0613_096	6.13.5.1	In response, aAccessorStatus: aAccessorCredentialsStatus shall indicate the status of the credentials in this accessor authentication service
RQ0613_097	6.13.5.1	In response, aAccessorStatus: aAccessorCredentialsPolicy shall indicate policies for the credentials in this accessor authentication service.
	6.13.5.2	AAS-ADMIN-CREATE-ACCESSOR-Service-Command

RQ number	Clause	Description
RQ0613_098	6.13.5.2	The accessor authentication service shall support the command AAS-ADMIN-CREATE-ACCESSOR-Service-Command as defined by ASN.1 in clause 6.13.5.2 of ETSI TS 103 666-1 [1]
RQ0613_099	6.13.5.2	Accessor authentication service shall allow an accessor to create another accessor and store its initial credentials if the accessor authentication service grants the eAASAccessRight-Create right to this accessor.
RQ0613_100	6.13.5.2	Accessor authentication service shall support the AAS-ADMIN-CREATE-ACCESSOR-Service-Command command parameter aAccessor which indicates the definition of the accessor to be created
RQ0613_101	6.13.5.2	Accessor authentication service shall support the AAS-ADMIN-CREATE-ACCESSOR-Service-Command command parameter aAccessorConditions which indicates the initial conditions of the accessor to be created
RQ0613_102	6.13.5.2	Accessor authentication service shall support the AAS-ADMIN-CREATE-ACCESSOR-Service-Command command parameter aCredential which indicates the initial credentials of the accessor to be created
RQ0613_103	6.13.5.2	if aCredential is present and the credentials are not conformant with the policies described in that, then the error eAAS- POLICY-RULES-VIOLATIONS shall be returned
RQ0613_104	6.13.5.2	Accessor authentication service shall support the AAS-ADMIN-CREATE-ACCESSOR-Service-Command command parameter aCredentialsPolicy which indicates the policy for the credentials of the accessor to be created
RQ0613_105	6.13.5.2	Accessor authentication service shall support the AAS-ADMIN-CREATE-ACCESSOR-Service-Command command parameter aCredentialsStatus which indicates initial status of the credentials of the accessor to be created
RQ0613_106	6.13.5.2	When the AAS-ADMIN-CREATE-ACCESSOR-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.2 of ETSI TS 103 666-1 [1]
	6.13.5.3	AAS-ADMIN-UPDATE-ACCESSOR-Service-Command
RQ0613_107	6.13.5.3	Accessor authentication service shall support the command AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command as defined by ASN.1 in clause 6.13.5.3 of ETSI TS 103 666-1 [1]
RQ0613_108	6.13.5.3	Accessor authentication service shall allow an accessor to update the credentials stored within a private storage of another accessor or of itself
RQ0613_109	6.13.5.3	Accessor authentication service shall allow the accessor to update the conditions and credentials if it has the eAASAccessRight-Update right
RQ0613_110	6.13.5.3	If credential policies are present in the command or previously in the accessor and the credentials are not conformant with the policies, then the error eAAS- POLICY-RULES-VIOLATIONS shall be returned
RQ0613_111	6.13.5.3	Accessor authentication service shall allow the accessor to update the access control list if it has the eAASAccessRight-UpdateACL right
RQ0613_112	6.13.5.3	Accessor authentication service shall allow the accessor to update the members of the group if it has the eAASAccessRight-UpdateGroup right
RQ0613_113	6.13.5.3	Accessor authentication service shall allow the accessor to update the credential policies if it has the eAASAccessRight-UpdateCredentialPolicy right
RQ0613_114	6.13.5.3	Accessor authentication service shall allow the accessor to update the credential status if it has eAASAccessRight-UpdateCredentialStatus right
RQ0613_115	6.13.5.3	The command shall be rejected with eAAS-ACL-RULES-VIOLATIONS if it contains any element for which the accessor does not have the rights to update
RQ0613_116	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aAccessor-Identity which indicates the accessor identity of the accessor to be updated
RQ0613_117	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aMembersOfGroup which indicates the updated list of the accessors in a group
RQ0613_118	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aACL which indicates the updated access control list for the accessor
RQ0613_119	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aSetAccessorConditions which indicates the access conditions that need to be added
RQ0613_120	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aRemoveAccessorConditions which indicates the access conditions that need to be removed. The removal of an access condition does not imply the deletion of the corresponding credentials or the change of the status.
RQ0613_121	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aSetCredential which indicates the new values of credentials to be updated

RQ number	Clause	Description
RQ0613_122	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aRemoveCredential which indicates the list of credentials that need to be removed from the SSP. The status of all credentials included in this list shall be disabled
RQ0613_123	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aCredentialsPolicy which indicates the updated credential policy. The values of credential policies that are not included in the command shall not be modified
RQ0613_124	6.13.5.3	Accessor authentication service shall support AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command command parameter aCredentialsStatus which indicates the updated credential status. The status values of credentials that are not included in the command shall not be modified
RQ0613_125	6.13.5.3	When the AAS-AAS-ADMIN-UPDATE-ACCESSOR-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.3 of ETSI TS 103 666-1 [1]
	6.13.5.4	AAS-ADMIN-DELETE-ACCESSOR-Service-Command
RQ0613_126	6.13.5.4	Accessor authentication service shall support the command AAS-ADMIN-DELETE-ACCESSOR-Service-Command as defined by ASN.1 in clause 6.13.5.4 of ETSI TS 103 666-1 [1]
RQ0613_127	6.13.5.4	Accessor authentication service shall allow an accessor to delete another accessor if it grants the eAASAccessRight-Delete to this latter accessor
RQ0613_128	6.13.5.4	Accessor authentication service shall support AAS-ADMIN-DELETE-ACCESSOR-Service-Command command parameter aAccessorIdentity which indicates the identity of the deleted accessor
RQ0613_129	6.13.5.4	When the AAS-ADMIN-DELETE-ACCESSOR-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.4 of ETSI TS 103 666-1 [1]
	6.13.5.5	AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command
RQ0613_130	6.13.5.5	Accessor authentication service shall support the command AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command as defined by ASN.1 in clause 6.13.5.5 of ETSI TS 103 666-1 [1]
RQ0613_131	6.13.5.5	Accessor authentication service shall support AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command parameter aPinNumericCredential which indicates numeric PIN credential
RQ0613_132	6.13.5.5	Accessor authentication service shall support AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command parameter aPinPasswordCredential which indicates password credential
RQ0613_133	6.13.5.5	Accessor authentication service shall support AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command parameter aPinPatternCredential which indicates pattern credential
RQ0613_134	6.13.5.5	Accessor authentication service shall support AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command parameter aAccessorTokenCertificationPath which indicates the certification path which end entity is the token generated by the accessor authentication application as defined in clause C.2 of ETSI TS 103 666-1 [1]
RQ0613_135	6.13.5.5	Accessor authentication service shall support AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command parameter aHostDomainCredential which indicates the accessor is authenticated if the command is issued by a host inside an host domain which has its UUID listed in credentials of type host domain
RQ0613_136	6.13.5.5	When the AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.5 of ETSI TS 103 666-1 [1]
RQ0613_137	6.13.5.5	In response, aCredentialsStatus shall indicate the status of the credentials after the execution of the request
RQ0613_138	6.13.5.5	In response, aServiceTokenCertificationPath shall indicate the certification path which end entity is the token generated by the accessor authentication service as defined in clause C.2 of ETSI TS 103 666-1 [1]
	6.13.5.6	AAS-OP-ACCESS-SERVICE-Service-Command
RQ0613_139	6.13.5.6	Accessor authentication service shall support the command AAS-OP-ACCESS-SERVICE-Service-Command as defined by ASN.1 in clause 6.13.5.6 in ETSI TS 103 666-1 [1]
RQ0613_140	6.13.5.6	The error code eAAS-E-NOK is returned if the usage of secure pipe is not requested by the accessor in the command, but it is required by the service
RQ0613_141	6.13.5.6	This command shall be executed only after successful authentication of the accessor, or the SSP shall reject it with the value eAAS-NOT-AUTHENTICATED

RQ number	Clause	Description
RQ0613_142	6.13.5.6	Accessor authentication service shall support AAS-OP-ACCESS-SERVICE-Service-Command command parameter aServiceIdentifier which indicates the gate identifier of a service in the SSP host
RQ0613_143	6.13.5.6	Accessor authentication service shall support AAS-OP-ACCESS-SERVICE-Service-Command command parameter aUseSecurePipe which indicates if a secure pipe is required to access the service
RQ0613_144	6.13.5.6	When AAS-OP-ACCESS-SERVICE-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.6 of ETSI TS 103 666-1 [1] where aGateIdentifier shall indicate identifier of the service gate dynamically allocated by the accessor authentication service
	6.13.5.7	AAS-OP-GET-CHALLENGE-Service-Command
RQ0613_145	6.13.5.7	Accessor authentication service shall support the command AAS-OP-GET-CHALLENGE-Service-Command with no parameters
RQ0613_146	6.13.5.7	When the AAS-OP-GET-CHALLENGE-Service-Command request is successful, then accessor authentication service gate shall include eAAS-OK in the response as described in ASN.1 in clause 6.13.5.7 of ETSI TS 103 666-1 [1] where aChallenge indicates challenge used for performing the mutual authentication between the accessor authentication service and the accessor authentication application. The challenge should be a random number of at least 128 bits. The way the challenge is generated is implementation dependant.
RQ0613_147	6.13.5.7	Accessor authentication service shall support response code eAAS-OK when the AAS-OP-GET-CHALLENGE-Service-Command command is completed successfully
	6.13.6.	Response code
	6.13.6.1	Overview
RQ0613_148	6.13.6.1	Accessor authentication service shall support response code eAAS-E-CMD-PAR-UNKNOWN when unknown parameters are used for an operation
RQ0613_149	6.13.6.1	Accessor authentication service shall support response code eAAS-E-NOK when the operation failed
RQ0613_150	6.13.6.1	Accessor authentication service shall support response code eAAS-ACL-RULES-VIOLATION when the operation violates the ACL conditions associated with an accessor
RQ0613_151	6.13.6.1	Accessor authentication service shall support response code eAAS-NOT-AUTHENTICATED when the accessor is not authenticated
RQ0613_152	6.13.6.1	Accessor authentication service shall support response code eAAS-POLICY-RULES-VIOLATION when the operation violates the credential policy
	6.13.6.2	Response code to access control primitives
RQ0613_153	6.13.6.2	Response code shall be returned in accordance with Table 6.6 of ETSI TS 103 666-1 [1]

5.3 Physical interfaces

5.3.1 Overview

Reference: ETSI TS 103 666-1 [1], clause 7.1

RQ number	Clause	Description
RQ0701_01	7.1	When the SSP contains two or more interfaces, each of them is completely independent, both electrically and logically. This implies that signalling on a contact assigned to one interface shall not affect the state of other contacts assigned to another interface. Similarly, an operation performed on one interface shall not alter the logical state of any other interface.

5.3.2 Reset

Reference: ETSI TS 103 666-1 [1], clause 7.2

RQ number	Clause	Description
RQ0702_01	7.2	Each physical interface shall support at least one of the following reset types: • Reset with dedicated line: this reset requires the presence of a dedicated line in the physical interface that indicates the reset (e.g. the RST line on the ISO/IEC 7816-3 [13] physical interface). • Logical reset: this reset is performed sending a command over the physical interface to indicate the reset to the SSP (e.g. RSET frame in SHDLC or USB Reset). This command may be sent at the data link layer or any layer above. • Hard reset: this reset is performed removing the power, if present, provided by the physical interface to the SSP (e.g. cold reset for the ISO/IEC 7816-3 [13] physical interface).
RQ0702_02	7.2	If the power provided by one physical interface is the only source of power of the SSP, a hard reset of that physical interface causes the reset of the entire SSP. In all other cases, a reset performed on any interface shall not interfere with the operations on the other interfaces, or with the operational state of the SSP itself.

5.3.3 ISO/IEC 7816 interface

Reference: ETSI TS 103 666-1 [1], clause 7.3

RQ number	Clause	Description
	7.3.1	Electrical specifications
	7.3.1.1	Electrical specifications of the interface
RQ0703_01	7.3.1.1	For the electrical specifications of the interface the provisions of ETSI TS 102 221 [7], clause 5 shall apply with the following exceptions: • The SSP may support a clock up to 20 MHz for the ISO/IEC 7816-3 [13] physical interface. • The SSP shall use an internal clock for the processing, when this is mandated by the SSP class. The SSP may use an internal clock in all other cases.
	7.3.1.2	Contacts
RQ0703_02	7.3.1.2	For the contacts the provisions of ETSI TS 102 221 [7], clause 4.5 shall apply with the following exception: • References to the usage of contacts C4 and C8 for the Inter-Chip USB interface.
	7.3.2	Initial communication establishment procedures
	7.3.2.1	SSP interface activation and deactivation
RQ0703_03	7.3.2.1	For the SSP interface activation and deactivation the provisions of ETSI TS 102 221 [7], clause 6.1 shall apply with the exceptions to the usage of contacts C4 and C8 for the Inter-Chip USB interface.
	7.3.2.2	Supply voltage switching
RQ0703_04	7.3.2.2	For the Supply voltage switching the provisions of ETSI TS 102 221 [7], clauses 6.2.0, 6.2.1 and 6.2.2 shall apply.
RQ0703_05	7.3.2.2	The maximum power consumption of the SSP after ATR shall be restricted to the values indicated in ETSI TS 102 221 [7], Table 6.4, until a different value is negotiated using the SSP capability exchange procedure, described in clause 6.4.2 of ETSI TS 102 221 [7].
	7.3.2.3	Answer To Reset content
RQ0703_06	7.3.2.3	The ATR shall be the first string of bytes sent from the SSP to the terminal after a reset has been performed. The ATR is defined in ISO/IEC 7816-3 [13].
RQ0703_07	7.3.2.3	The historical bytes indicate to the external world how to use the SSP. The information carried by the historical bytes shall follow ISO/IEC 7816-4 [8].
RQ0703_08	7.3.2.3	For the ATR the provisions of ETSI TS 102 221 [7], clauses 6.3.2 and 6.3.3 shall apply.
RQ0703_09	7.3.2.3	The ATR contains also some properties that are not related to the ISO/IEC 7816-3 [13]13 physical interface and that are negotiated between the SSP and the terminal during the capability exchange procedure described in clause 6.4.2 of ETSI TS 103 666-1 [1]. In this case, the terminal shall ignore those properties and use only the value negotiated in the capability exchange procedure.
	7.3.2.4	PPS procedure.
RQ0703_10	7.3.2.4	For the PPS procedure the provisions of ETSI TS 102 221 [7], clause 6.4 shall apply.
	7.3.2.5	Reset procedure.

RQ number	Clause	Description
RQ0703_11	7.3.2.5	For the Reset procedure the provisions of ETSI TS 102 221 [7], clause 6.5 shall apply: • The warm reset is a reset with dedicated line, as described in clause 7.2 of ETSI TS 103 666-1 [1]. • The cold reset is a hard reset, as described in clause 7.2 of ETSI TS 103 666-1 [1].
	7.3.2.6	Clock stop mode.
RQ0703_12	7.3.2.6	For the Clock stop mode the provisions of ETSI TS 102 221 [7], clause 6.6 shall apply.
	7.3.2.7	Bit/character duration and sampling time.
RQ0703_13	7.3.2.7	For the Bit/character duration and sampling time the provisions of ETSI TS 102 221 [7], clause 6.7 shall apply.
	7.3.2.8	Error handling.
RQ0703_14	7.3.2.8	For the Error handling the provisions of ETSI TS 102 221 [7], clause 6.8 shall apply.
	7.3.3	Data link protocols.
	7.3.3.1	Overview.
RQ0703_15	7.3.3.1	For the Data link protocols the provisions of ETSI TS 102 221 [7], clause 7.0 shall apply with the exceptions listed below. Only the protocol T=1 is mandatory for the terminal. The SSP shall support the protocol T=1.
	7.3.3.2	Character frame.
RQ0703_16	7.3.3.2	For the Character frame the provisions of ETSI TS 102 221 [7], clause 7.2.1 shall apply.
	7.3.3.3	Protocol T=1.
RQ0703_17	7.3.3.3	For Protocol T=1 the provisions of ETSI TS 102 221 [7], clause 7.2.3 shall apply.

5.3.4 SPI interface

Reference: ETSI TS 103 666-1 [1], clause 7.4

RQ number	Clause	Description
RQ0704_01	7.4	For the SPI interface the provisions of ETSI TS 103 713 [25] shall apply.

5.4 SSP Common Layer (SCL)

5.4.1 Introduction

Reference: ETSI TS 103 666-1 [1], clause 8.1

RQ number	Clause	Description
RQ0801_001	8.1	The SSP may support the SSP Common Layer (SCL) implementation comprised of optional network, transport and session layers
RQ0801_002	8.1	SCL shall be implemented using VNP, as specified in the GlobalPlatform VPP - Network Protocol [10] with the relevant sections and exceptions as described in ETSI TS 103 666-1 [1], clause 8.

5.4.2 SCL network

Reference: ETSI TS 103 666-1 [1], clause 8.2

RQ number	Clause	Description
RQ0802_001	8.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 3 shall apply
RQ0802_001a	8.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 3 shall apply, with the exception listed below: • One of the end points of any dynamic pipe shall be in the SSP host domain.
RQ0802_002	8.2	Table 8.1 of ETSI TS 103 666-1 [1] defines the URN for the additional gates defined in the current document, other than the ones referenced from GlobalPlatform VPP - Network Protocol [10]. All UUIDs are calculated using the version 5 of the UUID as specified in IETF RFC 4122 [19], using the domain name system namespace.
RQ0802_003	8.2	The data acknowledgement mechanism (EVT_ADM_RECEIVED) and the credit-based data flow control (EVT_ADM_CREDIT) described in ETSI TS 103 666-1 [1], clause 8.5.3 shall not apply unless otherwise specified in the gate description.

5.4.3 Protocol layers

Reference: ETSI TS 103 666-1 [1], clause 8.3

RQ number	Clause	Description
	8.3.1	Overview
RQ0803_001	8.3.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clauses 4.1 and 4.2 shall apply, with the exception listed below: • the MTU shall be 20 bytes or greater.
RQ0803_002	8.3.1	For proper operation, the protocol stack underlying the SCL shall provide a means for managing the underlying flow control.
RQ0803_003	8.3.1	There shall be an optional means for controlling (e.g. activating, deactivating) the underlying protocols and for getting the notifications from an underlying protocol (e.g. activation/deactivation of the interface by the terminal).
	8.3.2	Network layer
RQ0803_004	8.3.2	For the network layer, the provisions of GlobalPlatform VPP - Network Protocol [10], clause 4.3 shall apply.
	8.3.3	Transport layer
RQ0803_005	8.3.3	For the transport layer, the provisions of GlobalPlatform VPP - Network Protocol [10], clause 4.4 shall apply.
	8.3.4	Session layer
RQ0803_006	8.3.4	For the session layer, the provisions of GlobalPlatform VPP - Network Protocol [10], clause 4.5 shall apply.

5.4.4 SCL core services

Reference: ETSI TS 103 666-1 [1], clause 8.4

RQ number	Clause	Description
	8.4.1	Overview
RQ0804_001	8.4.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.1 shall apply for SCL core services.
	8.4.2	Common core features
RQ0804_002	8.4.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.2 shall apply for SCL common core services.
	8.4.3	Link gate
	8.4.3.1	Link service gate
	8.4.3.1.1	General description
RQ0804_003	8.4.3.1.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.3 shall apply for link gate.
RQ0804_004	8.4.3.1.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.3 shall apply for link gate with additional registry entries and SSP_MTU as defined in ETSI TS 103 666-1 [1], clause 8.4.3.1.
	8.4.3.1.2	Additional registry entries
RQ0804_005	8.4.3.1.2	Additional entries in the registry of the link service gate are defined in ETSI TS 103 666-1 [1], Table 8.2.
	8.4.3.1.3	SSP_MTU
RQ0804_006	8.4.3.1.3	SSP_MTU contains the value in bytes of the MTU of the link layer between the SCL router and the SSP. The entry shall have a value equal to or greater than 20.
RQ0804_007	8.4.3.1.3	An SCL host shall be able to send an SCL packet to the SSP without fragmentation, if the size of the SCL packet is less or equal to the value provided in this registry.
RQ0804_008	8.4.3.1.3	The SCL router shall be able to forward to the SSP any SCL packet with a size equal or smaller than the value provided in this registry, without any further fragmentation.
	8.4.3.2	Link application gate
RQ0804_009	8.4.3.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.4 shall apply for the link application gate.
	8.4.4	Administration gate
	8.4.4.1	Administration service gate
RQ0804_010	8.4.4.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.5 shall apply for the administration service gate. The credit-based data flow control mechanism and the data acknowledgement mechanism are not used in the administration service gate for its own usage (e.g. the reception of an event EVT_ADM_BIND does not trigger the emission of EVT_ADM_RECEIVED nor EVT_ADM_CREDIT).
	8.4.4.2	Administration application gate

RQ number	Clause	Description
RQ0804_011	8.4.4.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.6 shall apply for the administration application gate. The credit-based data flow control mechanism and the data acknowledgement mechanism are not used in the administration service gate for its own usage (e.g. the reception of an event EVT_ADM_BIND does not trigger the emission of EVT_ADM_RECEIVED nor EVT_ADM_CREDIT).
	8.4.5	Identity gate
	8.4.5.1	Identity service gate
	8.4.5.1.1	General description
RQ0804_012	8.4.5.1.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.7 shall apply for identity service gate.
RQ0804_013	8.4.5.1.1	The identity service gate should not list gates that are created dynamically as a result of an operation on a service of the SSP.
RQ0804_014	8.4.5.1.1	The credit-based data flow control mechanism and the data acknowledgement mechanism shall not be used in the identity service gate.
	8.4.5.1.2	Additional registry entries
RQ0804_015	8.4.5.1.2	Additional registry entries in the identity service gate are defined in ETSI TS 103 666-1 [1], Table 8.3.
	8.4.5.1.3	CAPABILITY_EXCHANGE
RQ0804_016	8.4.5.1.3	The capabilities of the host are coded with ASN.1 syntax as defined in ETSI TS 103 666-1 [1]: • clause 6.4.2.4, for SCL hosts outside the SSP host domain;
RQ0804_017	8.4.5.1.3	The capabilities of the host are coded with ASN.1 syntax as defined in ETSI TS 103 666-1 [1]: • clause 6.4.2.5, for SCL hosts inside the SSP host domain.
	8.4.5.1.4	GATE_URN_LIST
RQ0804_018	8.4.5.1.4	The GATE_URN_LIST provides an ASN.1 object containing an array of URNs according to IETF RFC 8141 [20] used to compute gate identifiers and the UUID resulting from the computation.
RQ0804_019	8.4.5.1.4	The Identity Application Gate may use this entry for service discovery.
RQ0804_020	8.4.5.1.4	The GATE_URN_LIST may have less, but shall not have more URNs than UUIDs listed in the GATE_LIST entry. All URNs provided in the GATE_URN_LIST shall be present in the GATE_LIST.
	8.4.5.2	Identity application gate
RQ0804_021	8.4.5.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.8 shall apply for identity application gate.
RQ0804_022	8.4.5.2	In addition to the provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.8, the additional entries in the gate registry defined in ETSI TS 103 666-1 [1], clause 8.4.5.1.2 shall apply for identity application gate.
RQ0804_023	8.4.5.2	The credit-based data flow control mechanism and the data acknowledgement mechanism shall not be used in the identity application gate.
	8.4.6	Loopback gate
	8.4.6.1	Loopback service gate
RQ0804_024	8.4.6.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.9 shall apply for the loopback service gate.
	8.4.6.2	Loopback application gate
RQ0804_025	8.4.6.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 5.10 shall apply for the loopback application gate.

5.4.5 SCL procedures

Reference: ETSI TS 103 666-1 [1], clause 8.5

RQ number	Clause	Description
	8.5.1	Host registration
RQ0805_001	8.5.1	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.1.2 shall apply for the SCL host registration.
	8.5.2	Host deregistration
RQ0805_002	8.5.2	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.1.4 shall apply for the SCL host deregistration.
	8.5.3	Pipe management

RQ number	Clause	Description
RQ0805_003	8.5.3	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.2 shall apply for the pipe management. Additionally, if a host receives a binding request and is not able to process the binding procedure for one or more service gates provided in the request, the host should reject the pipe session opening by answering with an EVT_ADM_BIND with a gate binding parameter using the pipe identifier '7F' for the gates on which no pipe session has been opened
RQ0805_004	8.5.3	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.2 shall apply for the pipe management. Additionally, a host shall not request a pipe binding for a service gate if this service gate has already a pipe session for this host.
	8.5.4	Registry access
RQ0805_005	8.5.4	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.3 shall apply for registry access.
	8.5.5	Hosts and gates discovery
RQ0805_006	8.5.5	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.4 shall apply for hosts and gates discovery.
	8.5.6	Loopback testing
RQ0805_007	8.5.6	The provisions of GlobalPlatform VPP - Network Protocol [10], clause 6.5 shall apply for loopback testing.

5.5 Secure SCL

5.5.1 Protocol Stack

Reference: ETSI TS 103 666-1 [1], clause 9.1.

5.5.2 Secure datagram

Reference: ETSI TS 103 666-1 [1], clause 9.2

RQ number	Clause	Description
RQ0902_001	9.2	The value of the DIVERSIFIER defined in the clause C.4.2 of ETSI TS 103 666-1 [1] shall be the logical XOR of the aChallenge value as defined in the clause 6.13.5.7 of ETSI TS 103 666-1 [1] and the service identifier (aServiceIdentifier) as defined in clause 6.13.5.6 of ETSI TS 103 666-1 [1].
RQ0902_002	9.2	PL shall correspond to the number of padding bytes appended to the message fragment (see padding below). PL is coded on bits 1 to 7.
RQ0902_003	9.2	CB shall be set to 1 for the last or only fragment of a message.
RQ0902_004	9.2	The length of the secure message fragment shall be a multiple of 16 bytes.
RQ0902_005	9.2	Secure SCL message: contains the cryptogram of the structure consisting of message fragment, padding, CB and PL. The cryptogram is generated by using a stream cipher algorithm identified by StreamCipherIdentifier value (see clause C.2.2 of ETSI TS 103 666-1 [1]).
RQ0902_006	9.2	ICHECK: Integrity check of the secure SCL message using the stream cipher algorithm identified by StreamCipherIdentifier value.
RQ0902_007	9.2	If the stream cipher algorithm is the GCM then each gate supporting the secure SCL shall manage two GCM monotonic counters which shall be incremented after sending and receiving a secure SCL message.
RQ0902_008	9.2	The GCM counter shall be set to 0 after the successful authentication of the accessor.
RQ0902_009	9.2	The secure SCL message results from the encryption by using the security function as defined in clause C.3.5 of ETSI TS 103 666-1 [1].
RQ0902_010	9.2	The key KS^2, used by this security function, shall be deduced from the Accessor Authentication Service Protocol as defined in clause 10.9 of ETSI TS 103 666-1 [1].

5.5.3 Security protocol

Reference: ETSI TS 103 666-1 [1], clause 9.3

RQ number	Clause	Description
	9.3.1	Overview
	9.3.2	Shared secret initialization
RQ0903_001	9.3.2	The Accessor Authentication Service (AAS) generates a challenge (aChallenge) and sends it to the Accessor Authentication Application (AAA) - see Figure 9.3 in ETSI TS 103 666-1 [1].
RQ0903_002	9.3.2	The accessor authentication application shall generate an ephemeral key pair (ePK.AAA.ECKA, eSK.AAA.ECKA)
RQ0903_003	9.3.2	The accessor authentication application shall generate and send the Certification_Path $_{ATK\_AAA}$ which end entity is ATK.AAA.ECKA authentication token as defined in clause C.2.2 of ETSI TS 103 666-1 [1], signed by SK.AAA.ECDSA private key
RQ0903_004	9.3.2	The accessor authentication application shall send the Certification_Path $_{ATK\_AAA}$ to the accessor authentication service.
RQ0903_005	9.3.2	The Accessor Authentication Service shall validate Certification_Path $_{ATK\_AAA}$ by using PK.CI AAA.ECDSA public key
RQ0903_006	9.3.2	The Accessor Authentication Service shall generate an ephemeral key pair (ePK.AAS.ECKA, eSK.AAS.ECKA)
RQ0903_007	9.3.2	The Accessor Authentication Service shall generate the Certification_Path $_{ATK\_AAS}$ which end entity is ATK.AAS.ECKA authentication token as defined in clause C.2.2 of ETSI TS 103 666-1 [1], signed by SK.AAS.ECDSA private key
RQ0903_008	9.3.2	The Accessor Authentication Service shall compute the shared secret ShS by using ECKA_DH (anonymous Diffie-Hellman ECC key agreement) with the ephemeral key pair eSK.AAS.ECKA and ePK.AAA.ECKA as defined in clause C.4 of ETSI TS 103 666-1 [1].
RQ0903_009	9.3.2	The Accessor Authentication Service shall send the Certification_Path $_{ATK\_AAS}$ to the accessor authentication application.
RQ0903_010	9.3.2	The accessor authentication application shall validate the Certification_Path $_{ATK\_AAS}$ by using PK.CI AAS .ECDSA public key
RQ0903_011	9.3.2	The accessor authentication application shall compute the shared secret ShS by using ECKA_DH (anonymous Diffie-Hellman ECC key agreement) with the ephemeral key pair eSK.AAA.ECKA and ePK.AAS.ECKA.
RQ0903_012	9.3.2	The accessor authentication service shall verify that the Certification_Path $_{ATK\_AAA}$ contains aChallenge challenge
RQ0903_013	9.3.2	The accessor authentication application shall verify that the Certification_Path $_{ATK\_AAS}$ contains aChallenge challenge
RQ0903_014	9.3.2	ShS shared secret is the seed which shall be used for deriving the keys for the secure SCL communication.
	9.3.3	Secure SCL shared keys generation
RQ0903_015	9.3.3	From the shared secret ShS obtained from the procedure described in clause 9.3.2 of ETSI TS 103 666-1 [1], any service or application may initiate the generation of the key data by using the KDF function defined in clause C.4 of ETSI TS 103 666-1 [1] and a DIVERSIFIER128BIT that is equal to aGateIdentifier defined in clause 6.13.5.6 of ETSI TS 103 666-1 [1].
RQ0903_016	9.3.3	Generation of key data is performed for each AAS-OP-ACCESS-SERVICE-Service-Command as defined in clause 6.13.5.7 of ETSI TS 103 666-1 [1] and when the secure SCL is required

5.5.4 Accessor authentication service procedure

Reference: ETSI TS 103 666-1 [1], clause 9.4

RQ number	Clause	Description
	9.4.1	Initialization
RQ0904_001	9.4.1	The accessor authentication application gate requests the initialization of the security protocol through AAS-OP-GET-CHALLENGE-Service-Command
RQ0904_002	9.4.1	The accessor authentication service gate returns the aChallenge as AAS-OP-GET-CHALLENGE-Service-Command response.
RQ0904_003	9.4.1	The accessor authentication application gate requests an authentication based on the authentication token with its Certification_Path $_{ATK\_AAA}$ as parameter of AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command
RQ0904_004	9.4.1	The accessor authentication service gate returns its Certification_Path $_{ATK\_AAS}$ as parameter of AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command response.

5.6 Communication layers above SCL

5.6.1 Overview

Reference: ETSI TS 103 666-1 [1], clause 10.1

5.6.2 APDU protocol

Reference: ETSI TS 103 666-1 [1], clause 10.2

RQ number	Clause	Description
	10.2.1	Introduction
	10.2.2	Command-response pairs
	10.2.2.1	General definition
RQ1002_001	10.2.2.1	The provisions of ISO/IEC 7816-4 [8], clause 5.1 shall apply, with the exceptions listed in the ETSI TS 103 666-1 [1], clause 10.2.2
	10.2.2.2	CLA Byte
RQ1002_002	10.2.2.2	The provisions of ETSI TS 102 221 [7], clause 10.1.1 shall apply.
	10.2.2.3	INS Byte
RQ1002_003	10.2.2.3	SSP shall support following commands on the basic logical channel as defined in ETSI TS 103 666-1 [1], Table 10.1: • SELECT • MANAGE CHANNEL • EXCHANGE CAPABILITIES
RQ1002_004	10.2.2.3	The values '6X' and '9X' are invalid as instructions.
	10.2.2.4	Coding of SW1 and SW2
RQ1002_005	10.2.2.4	If no application is selected and the SSP has support for the UICC file system, the provisions of ETSI TS 102 221 [7], clause 10.2.2 shall apply.
RQ1002_006	10.2.2.4	The value '61XX' is reserved as a special value when APDUs are transported over ISO/IEC 7816-4 [8] interface and shall not be used for other purposes.
	10.2.3	SSP commands
	10.2.3.1	Overview
	10.2.3.2	EXCHANGE CAPABILITIES
	10.2.3.2.1	Description
RQ1002_007	10.2.3.2.1	EXCHANGE CAPABILITIES command shall be executed immediately after the SSP Interface Session is started. The command might be executed again if some of the capabilities change.
RQ1002_008	10.2.3.2.1	The SSP and the terminal shall use the values exchanged during the last execution of EXCHANGE CAPABILITIES command.
RQ1002_009	10.2.3.2.1	The values of the EXCHANGE CAPABILITIES command shall take precedence over any equivalent value exchanged over the physical interface (for example, using the ATR) or over the transport interface.
	10.2.3.2.2	Command parameters
	10.2.3.2.3	Command data
RQ1002_010	10.2.3.2.3	Command data of "EXCHANGE CAPABILITIES" contains sequence of TLVs, coded as per ETSI TS 103 666-1 [1], clause 6.4.2.4.
	10.2.3.2.4	Command response
RQ1002_011	10.2.3.2.4	Response of command "EXCHANGE CAPABILITIES" contains sequence of TLVs, coded as per ETSI TS 103 666-1 [1], clause 6.4.2.5.
	10.2.3.3	SELECT
RQ1002_012	10.2.3.3	The provisions of ISO/IEC 7816-4 [8] and of ETSI TS 102 221 [7] for the SELECT command with P1 = '04' ("Select by DF name") shall apply. The coding of P2 is described in ETSI TS 103 666-1 [1], Table 10.3.
	10.2.4	Logical channels
	10.2.4.1	Overview
RQ1002_013	10.2.4.1	The SSP indicates the maximum number of supported logical channels during the capability exchange with the terminal.
RQ1002_014	10.2.4.1	A logical channel is opened by using a MANAGE CHANNEL command, in which the card assigns a channel number and returns it in the response.
RQ1002_015	10.2.4.1	The logical channel shall remain open until it is explicitly closed by a MANAGE CHANNEL command, or if the connection between the terminal and the SSP entity handling the APDUs is deactivated.
	10.2.4.2	Manage Channel
RQ1002_016	10.2.4.2	The provisions of ETSI TS 102 221 [7], clause 11.1.17 shall apply.

RQ number	Clause	Description
RQ1002_017	10.2.4.2	Support for "MANAGE CHANNEL" command is mandatory if the SSP indicates support for logical channels during the capability exchange procedure.
	10.2.5	UICC file system commands
	10.2.5.1	Overview
	10.2.5.2	Methods for selecting a file
RQ1002_018	10.2.5.2	The provisions of ETSI TS 102 221 [7], clause 8.4 shall apply.
	10.2.5.3	Reservation of file IDs
RQ1002_019	10.2.5.3	The provisions of ETSI TS 102 221 [7], clause 8.6 shall apply.
	10.2.5.4	Security features
RQ1002_020	10.2.5.4	The provisions of ETSI TS 102 221 [7], clause 9 shall apply.
	10.2.5.5	Additional commands
RQ1002_021	10.2.5.5	In addition to the commands described in clause 10.2.3, these additional commands, as defined in ETSI TS 103 666-1 [1], Table 10.4, shall be supported by the SSP on the default logical channel.
	10.2.6	Card Application Toolkit
	10.2.6.1	Overview
RQ1002_022	10.2.6.1	When the SSP indicates support for Card Application Toolkit according to ETSI TS 102 223 [9], the provisions of ETSI TS 102 221 [7], clause 7.4.2 shall apply.
RQ1002_023	10.2.6.1	When the physical interface used to transport APDUs allows the SSP to remotely wake up the terminal in case of proactive command, the SSP shall use that mechanism to inform the terminal of a pending proactive command. In this case, the terminal shall use the FETCH command APDU (see ETSI TS 102 221 [7]) to get the pending proactive command.
RQ1002_024	10.2.6.1	When the physical interface used to transport APDUs does not allow the SSP to remotely wake up the terminal in case of proactive command, the SSP can reply '91XX' in place of '9000' to indicate that a proactive command is pending.
RQ1002_025	10.2.6.1	In all cases, the terminal shall send the FETCH and TERMINAL RESPONSE commands on the basic logical channel 0, even if the command to which the card replied with '91XX' was sent on a logical channel different from the basic logical channel.
	10.2.6.2	Terminal profile
RQ1002_026	10.2.6.2	The Card Application Toolkit terminal profile allows the SSP to determine what the terminal is capable of, and the SSP shall then limit its instruction range accordingly.
RQ1002_027	10.2.6.2	If the terminal supports the Card Application Toolkit, the terminal profile shall be included in the capability exchange procedure.
RQ1002_028	10.2.6.2	The content of the terminal profile is defined in ETSI TS 102 223 [9], clause 5.2.
	10.2.6.3	Proactive polling
RQ1002_029	10.2.6.3	When the proactive polling is indicated as required in the capability exchange procedure, the terminal shall perform proactive polling as defined in ETSI TS 102 221 [7].
RQ1002_030	10.2.6.3	Proactive polling as defined in ETSI TS 102 221 [7] is optional: when the physical interface used to transport APDUs allows the SSP to remotely wake up the terminal;
RQ1002_031	10.2.6.3	Proactive polling as defined in ETSI TS 102 221 [7] is optional: when the Card Application Toolkit is not supported by the SSP.
	10.2.6.4	Additional commands
RQ1002_032	10.2.6.4	In addition to the commands described in ETSI TS 103 666-1 [1], clause 10.2.3, these additional commands shall be supported by the SSP on the default logical channel as defined in Table 10.5 of ETSI TS 103 666-1 [1].
	10.2.7	SSP suspension
RQ1002_033	10.2.7	If the SSP suspension is supported, these additional commands, as defined in Table 10.6 of ETSI TS 103 666-1 [1], shall be supported by the SSP on the default logical channel.
	10.2.8	APDU transfer over SCL
	10.2.8.1	Overview
	10.2.8.2	UICC APDU gate
	10.2.8.2.1	UICC APDU overview
RQ1002_034	10.2.8.2.1	If APDUs are carried over SCL, an SSP host shall contain no more than one UICC APDU service gate.
RQ1002_035	10.2.8.2.1	Each SCL host outside the SSP host domain shall not create more than one pipe session to the UICC APDU service gate for each SSP host.
RQ1002_036	10.2.8.2.1	The communication between the UICC APDU service gate and the UICC APDU application gate shall use the presentation layer defined in ETSI TS 102 622 [5], clause 5.2 and the specific part of the transport layer with the Go-and-Wait data flow control defined in ETSI TS 103 666-1 [1], clause 10.2.8.2.4.

RQ number	Clause	Description
RQ1002_037	10.2.8.2.1	The UICC APDU gates shall reuse the mechanisms described in ETSI TS 102 622 [5], clause 12.
	10.2.8.2.2	UICC APDU service gate
RQ1002_038	10.2.8.2.2	The UICC APDU service gate URN shall support the syntax as defined in ETSI TS 103 666-1 [1], clause 8.2, with the values specified in Table 8.1.
RQ1002_039	10.2.8.2.2	The UICC APDU service gate shall support the commands, events and registry as described in ETSI TS 102 622 [5], clause 12.2.
	10.2.8.2.3	UICC APDU application gate
	10.2.8.2.3.1	Commands
RQ1002_040	10.2.8.2.3.1	The UICC APDU application gate shall support the commands described in ETSI TS 102 622 [5], clause 12.3.1.
	10.2.8.2.3.2	Events
RQ1002_041	10.2.8.2.3.2	The UICC APDU application gate shall support the commands described in ETSI TS 102 622 [5], clause 12.3.2, with the addition of the events defined in ETSI TS 103 666-1 [1], Table 10.7.
	10.2.8.2.3.3	EVT_TOOLKIT_REQUEST
RQ1002_042	10.2.8.2.3.3	"EVT_TOOLKIT_REQUEST" event shall be sent by the UICC APDU service gate in idle state to indicate to the UICC APDU application gate that a proactive command is pending.
RQ1002_043	10.2.8.2.3.3	After receiving this event, the UICC APDU application gate shall send an APDU containing the STATUS command to allow the SSP to start a proactive session.
	10.2.8.2.3.4	State diagram for the UICC APDU gate
RQ1002_044	10.2.8.2.3.4	Other events or commands received on the UICC APDU gate shall not change its state, with the exception of EVT_ADM_UNBIND which can be received in any state.

5.6.3 File system protocol

Reference: ETSI TS 103 666-1 [1], clause 10.3

RQ number	Clause	Description
	10.3.1	Overview
RQ1003_001	10.3.1	The SSP file system, as defined in clause 6.6.2 of ETSI TS 103 666-1 [1], may be accessed by entities outside the SSP using the SSP file system service over the SCL protocol.
RQ1003_002	10.3.1	The SSP file system service resides in an SSP host and shall contain a single file system control service gate.
RQ1003_003	10.3.1	The SSP file system application resides in an SCL host outside the SSP host domain and it shall contain a single file system control application gate.
RQ1003_004	10.3.1	SSP file system application may contain multiple file system data application gates.
RQ1003_005	10.3.1	SSP file system service may contain multiple file system data service gates.
RQ1003_006	10.3.1	Small amount of data may then be exchanged via the file system control pipe.
RQ1003_007	10.3.1	The file system control service gate URN supports the syntax defined in clause 8.2 of ETSI TS 103 666-1 [1], with the values specified in Table 8.1 of ETSI TS 103 666-1 [1]
	10.3.2	Presentation layer
RQ1003_008	10.3.2	The file system control service gate and the file system control application gate implement an ASN.1 presentation layer using the definitions in clause 6.6.2. of ETSI TS 103 666-1 [1]
	10.3.3	File system control service gate
	10.3.3.1	Overview
RQ1003_009		An SSP file system application may access to the SSP file system service interfacing the SSP File System via a pipe session between a file system control application gate and a file system control service gate.
RQ1003_010	10.3.3.1	The file system control service gate is in charge of: • conveying the administrative (ADMIN) and operational (OP) commands; • triggering of a pipe session as defined in clause 8 of ETSI TS 103 666-1 [1] for connecting a data stream between the file system data service gate and a file system data application gate within the SSP file system service and the SSP file system application.
	10.3.3.2	Commands

RQ number	Clause	Description
RQ1003_011	10.3.3.2	The file system control service gate supports the following commands: FS-ADMIN-GET-CAPABILITIES-Service-Command, FS-ADMIN-CREATE-NODE-Service-Command, FS-ADMIN-DELETE-NODE-Service-Command, FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command, FS-OP-FILE-OPEN-Service-Command, FS-OP-FILE-CLOSE-Service-Command, FS-OP-NODE-GET-INFO-Service-Command, FS-OP-FILE-READ-Service-Command, FS-OP-FILE-WRITE-Service-Command, FS-OP-FILE-GET-POSITION-Service-Command
	10.3.3.3	Responses
RQ1003_012	10.3.3.3	The file system control service gate supports the following responses: FS-ADMIN-GET-CAPABILITIES-Service-Response, FS-ADMIN-CREATE-NODE-Service-Response, FS-ADMIN-DELETE-NODE-Service-Response, FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Response, FS-OP-FILE-OPEN-Service-Response, FS-OP-FILE-CLOSE-Service-Response, FS-OP-NODE-GET-INFO-Service-Response, FS-OP-FILE-READ-Service-Response, FS-OP-FILE-WRITE-Service-Response, FS-OP-FILE-GET-POSITION-Service-Response. All the possible response codes are described in clause 6.6.4.2 of ETSI TS 103 666-1 [1].
	10.3.3.4	Events
	10.3.4	File system control application gate
	10.3.4.1	Overview
RQ1003_013	10.3.4.1	The file system control application gate provides access to services for administration and operation on the SSP file system using the SSP file system control service gate between an SSP file system application and an SSP file system service.
RQ1003_014	10.3.4.1	All file sessions and file system data pipe sessions shall be closed upon closure of the pipe session between the file system control application gate and the file system control service gate.
	10.3.4.2	Commands
	10.3.4.3	Responses
	10.3.4.4	Events
	10.3.5	File system data service gate
	10.3.5.1	Overview
RQ1003_015	10.3.5.1	The file system data service gate provides access to a file stream between an SSP file system application and an SSP file system service.
RQ1003_016	10.3.5.1	The SSP file system service shall open a pipe session between a file system data service gate and a file system data application gate when a request to open an SSP file with FS-OP-FILE-OPEN-Service-Command command containing the aGateURI value is successful.
RQ1003_017	10.3.5.1	The file system data service gate shall implement the credit-based data flow control and data acknowledgement as defined in clause 8.5.3 of ETSI TS 103 666-1 [1]
RQ1003_018	10.3.5.1	The pipes between the file system data service gate and the file system data application gate allow conveying a data stream. The SCL packets shall have the CB bit (chaining bit) always set to 0.
	10.3.5.2	Commands
	10.3.5.3	Responses
	10.3.5.4	Events
	10.3.6	File system data application gate
	10.3.6.1	Overview
RQ1003_019	10.3.6.1	The file system data application gate provides access to transfer a file stream using the SSP file system data service gate between an SSP file system application and an SSP file system service.
RQ1003_020	10.3.6.1	The file system data application gate shall implement the credit-based data flow control and data acknowledgement as defined in clause 8.5.3. of ETSI TS 103 666-1 [1]

5.6.4 Transmission Control Protocol support

Reference: ETSI TS 103 666-1 [1], clause 10.4

RQ number	Clause	Description
	10.4	Transmission Control Protocol support
	10.4.1	Overview
RQ1004_001	10.4.1	The TCP data service gate implements the protocol stack to communicate over the SCL network to a TCP data application gate for bridging the TCP adapter and TCP consumer.
RQ1004_002	10.4.1	The TCP adapter is in charge of the creation of the TCP connection;
RQ1004_003	10.4.1	The TCP adapter is in charge of the resolution of DNS address, when this is required;
RQ1004_004	10.4.1	The TCP adapter is in charge of the triggering of a pipe session as defined in clause 8 for connecting the TCP adapter and the TCP consumer in the SSP.
RQ1004_005	10.4.1	The TCP adapter shall contain only one TCP control service gate.
RQ1004_006	10.4.1	If an SSP host supports the TCP protocol, it shall contain only one TCP consumer, which includes a single TCP control application gate.
RQ1004_007	10.4.1	An SCL host may contain one or more TCP data service gates.
RQ1004_008	10.4.1	Each TCP data application gate shall be exclusive for a single SSP Application.
RQ1004_009	10.4.1	The TCP control service gate URN shall support the syntax as defined in ETSI TS 103 666-1 [1], clause 8.2 with the values specified in ETSI TS 103 666-1 [1], Table 8.1.
	10.4.2	Management of TCP connections
	10.4.2.1	TCP connection request
	10.4.2.1.1	TCP active connection request (client mode)
RQ1004_010	10.4.2.1.1	Upon request of the TCP consumer to establish an active connection, the TCP adapter shall process a connection establishment to the remote endpoint indicated in the request.
RQ1004_011	10.4.2.1.1	If an FQDN is provided in the request, the TCP adapter shall perform a DNS resolution if supported.
RQ1004_012	10.4.2.1.1	If the DNS resolution is not supported by the TCP adapter, or the establishment of the connection failed, the TCP adapter shall indicate to the TCP consumer that the connection request failed with the appropriate error indicator.
RQ1004_013	10.4.2.1.1	After the TCP connection is successfully established, the TCP adapter shall open a TCP data pipe to the TCP data application gate identifier indicated by the TCP consumer. This shall be interpreted by the TCP consumer that the TCP connection was successfully established.
	10.4.2.1.2	TCP passive connection request (server mode)
RQ1004_014	10.4.2.1.2	Upon request of the TCP consumer to establish a connection as passive, the TCP adapter shall bind and listen to the port provided in the request
RQ1004_015	10.4.2.1.2	In case of failure, the TCP adapter shall indicate to the TCP consumer that the connection failed with the appropriate error indicator.
RQ1004_016	10.4.2.1.2	The TCP adapter shall support multiple passive requests by the TCP consumer with different TCP data application gate identifiers to the same TCP port in order to allow multiple incoming TCP connections on the same port.
RQ1004_017	10.4.2.1.2	When a connection is successfully established to the listening TCP port, the TCP adapter shall request the TCP consumer to accept the incoming connection before completing the TCP handshake.
RQ1004_018	10.4.2.1.2	If the TCP consumer accepts the incoming connection, the TCP adapter shall open a TCP data pipe to the TCP data application gate identifier indicated by the TCP consumer in the request.
RQ1004_019	10.4.2.1.2	If the TCP consumer rejects the incoming connection, the TCP adapter shall close the incoming TCP connection.
RQ1004_020	10.4.2.1.2	The TCP adapter shall accept additional incoming TCP connections to the same port until all TCP data application gate identifiers corresponding to this port are used.
	10.4.2.2	TCP connection established
RQ1004_021	10.4.2.2	All data received by the TCP adapter in its TCP endpoint shall be transferred via the related TCP data pipe to the TCP consumer.
RQ1004_022	10.4.2.2	All data sent by the TCP consumer on the TCP data pipe shall be sent to its corresponding TCP endpoint.
	10.4.2.3	TCP end of connection
RQ1004_023	10.4.2.3	If a TCP session is closed by the remote TCP endpoint, the TCP adapter shall close the related TCP data pipe session: in this case, the connection identifier is immediately released and the TCP consumer does not need to send the request to close the TCP connection.

RQ number	Clause	Description
RQ1004_024	10.4.2.3	If the TCP data application gate ends the TCP data pipe session, the TCP adapter shall terminate the connection to the remote TCP endpoint.
RQ1004_025	10.4.2.3	If the TCP control pipe session is closed, all TCP connections and TCP data pipe sessions shall be closed. All passive connection requests are terminated.
	10.4.3	Presentation layer
RQ1004_026	10.4.3	The TCP control and application gates shall implement an ASN.1 presentation layer using the definitions, described in ETSI TS 103 666-1 [1], clause 10.4.3.
	10.4.4	TCP control service gate
	10.4.4.1	Overview
	10.4.4.2	Commands
	10.4.4.2.1	List of commands
RQ1004_027	10.4.4.2.1	"TCP control service gate" shall support the commands described in ETSI TS 103 666-1 [1], clause 10.4.4.2.1
	10.4.4.2.2	TCP-REQUEST-CONNECTION-Service-Command
RQ1004_028	10.4.4.2.2	If the FQDN value is used to establish the TCP connection between a TCP consumer and TCP adapter with the command TCP-REQUEST-CONNECTION-Service-Command, the TCP adapter is responsible to perform the DNS resolution, if the feature is supported.
RQ1004_029	10.4.4.2.2	In case of eActive connection mode, the TCP adapter shall initiate a TCP connection, as described in IETF RFC 793 [26], section 3.4.
RQ1004_030	10.4.4.2.2	In case of ePassiveLocal or ePassiveAny connection mode, the TCP adapter shall open a listening port to accept incoming TCP connections.
RQ1004_031	10.4.4.2.2	"TCP-REQUEST-CONNECTION-Service-Command" command shall use the parameters described in ETSI TS 103 666-1 [1], clause 10.4.4.2.2.
RQ1004_032	10.4.4.2.2	Parameter "aDestinationAddress" shall be ignored by TCP adapter for connections in passive mode i.e. aConnectionMode with value ePassiveLocal or ePassiveAny.
RQ1004_033	10.4.4.2.2	Parameter "aDestinationAddress" is mandatory for connection in active mode i.e. aConnectionMode with value eActive.
RQ1004_034	10.4.4.2.2	For connections in passive mode, i.e. aConnectionMode with value ePassiveLocal or ePassiveAny, "aPortNumber" defines the port number that the TCP adapter shall listen on.
RQ1004_035	10.4.4.2.2	"aGateID" is the UUID of the TCP data gate that will be associated to the opened TCP connection.
RQ1004_036	10.4.4.2.2	"aTimeout" is duration of time before the terminal stops the attempt to connect to a remote sever.
RQ1004_037	10.4.4.2.2	"aNetworkParameters" indicates the network parameters using which the TCP connection shall be established.
RQ1004_038	10.4.4.2.2	"aBearerType" indicates the bearer type on which the TCP connection shall be established.
RQ1004_039	10.4.4.2.2	"aNetworkAccessName" provides information to the terminal necessary to identify the gateway entity which provides interworking with an external packet data network. If the parameter is not present, the terminal may use the default network access name in the terminal configuration or the default subscription value. It is defined in clause 8.70 of ETSI TS 102 223 [9].
RQ1004_040	10.4.4.2.2	If the terminal equipment supports a remote access login feature, "aUserLogin and aUserPassword" gives necessary information for authentication as described in ETSI TS 102 223 [9], clauses 6.6.27.2 and 6.6.27.4. The format and content of the data (data coding scheme and text string) is described in clause 8.15 of ETSI TS 102 223 [9].
RQ1004_041	10.4.4.2.2	For all the connection modes, the TCP adapter shall send the response, as described in ETSI TS 103 666-1 [1], clause 10.4.4.2.2, immediately after starting the procedure to establish the TCP connection.
RQ1004_042	10.4.4.2.2	When the connection request is successful then TCP adapter shall include eTCP-OK in the response.
RQ1004_043	10.4.4.2.2	"aConnectionID" identifier shall be unique across all open TCP sessions.
	10.4.4.2.3	TCP-CLOSE-CONNECTION-Service-Command
RQ1004_044	10.4.4.2.3	"TCP-CLOSE-CONNECTION-Service-Command" command shall use the parameters as described in ETSI TS 103 666-1 [1], clause 10.4.4.2.3
RQ1004_045	10.4.4.2.3	When successful, on command "TCP-CLOSE-CONNECTION-Service-Command", the TCP adapter shall include eTCP-OK in the response, as described in ETSI TS 103 666-1 [1], clause 10.4.4.2.3
	10.4.4.2.4	TCP-GET-STATUS-CONNECTION-Service-Command
RQ1004_046	10.4.4.2.4	"TCP-GET-STATUS-CONNECTION-Service-Command" command shall use the parameters as described in ETSI 103 666-1 1, clause 10.4.4.2.3

RQ number	Clause	Description
RQ1004_047	10.4.4.2.4	When command "TCP-GET-STATUS-CONNECTION-Service-Command " is successful the TCP adapter shall include eTCP-OK in the response, as described in ETSI TS 103 666-1 [1], clause 10.4.4.2.4
	10.4.4.3	Responses
RQ1004_048	10.4.4.3	The "TCP Control Service" gate shall support the responses defined in ETSI TS 103 666-1 [1], clause 10.4.4.3
RQ1004_049	10.4.4.3	The "TCP Control Service" gate shall support the error codes defined in ETSI TS 103 666-1 [1], Table 10.9
	10.4.5.1	Overview
	10.4.5.2	Commands
	10.4.5.2.1	List of Commands
RQ1004_050	10.4.5.2.1	"TCP CONTROL APPLICATION GATE" supports the commands, as described in ETSI TS 103 666-1 [1], clause 10.4.5.2.1
	10.4.5.2.2	TCP-ACCEPT-CONNECTION-Application-Command
RQ1004_051	10.4.5.2.2	"TCP-ACCEPT-CONNECTION-Application-Command" has the parameters, as described in ETSI TS 103 666-1 [1], clause 10.4.5.2.2
RQ1004_052	10.4.5.2.2	The TCP consumer shall include eTCP-OK in the "TCP-ACCEPT-CONNECTION-Application-Response" if the TCP client connection is accepted.
RQ1004_053	10.4.5.2.2	If the TCP client connection is not accepted, the TCP consumer shall include eTCP-E-NOK in the "TCP-ACCEPT-CONNECTION-Application-Response" if the connection is rejected due to internal reasons. The TCP adapter shall terminate the incoming TCP connection and move back to eLISTEN state. As described in ETSI TS 103 666-1 [1], clause 10.4.5.2.2
	10.4.5.3	Responses
RQ1004_054	10.4.5.3	"TCP control application gate" shall support the response, as described in ETSI TS 103 666-1 [1], clause 10.4.5.3
RQ1004_055	10.4.5.3	"TCP control application gate" shall support the error codes defined in ETSI TS 103 666-1 [1], Table 10.10
	10.4.5.4	Events
	10.4.5.4.1	List of Events
RQ1004_056	10.4.5.4.1	TCP consumer supports events, as described in ETSI TS 103 666-1 [1], clause 10.4.5.4.1
	10.4.5.4.2	EVT-TCP-ERROR-Application-Event
RQ1004_057	10.4.5.4.2	With the event EVT-TCP-ERROR-Application-Event, the TCP adapter notifies the TCP consumer that an error occurred, as described in ETSI TS 103 666-1 [1], clause 10.4.5.4.2
RQ1004_058	10.4.5.4.2	"eUNREACHABLE" parameter means that the destination IP address is unreachable as described in ICMP messages defined in IETF RFC 792 [27], In this case, the aErrorInfo parameter shall be completed with the code value defined in clause "Destination Unreachable Message" of IETF RFC 792 [27].
RQ1004_059	10.4.5.4.2	"eREDIRECTION" parameter means that a redirection occurs in the route to convey the message as described in clause "Redirect Message" of IETF RFC 792 [27], In this case, the aErrorInfo parameter shall be completed with the code value defined in clause "Redirect Message" of IETF RFC 792 [27].
RQ1004_060	10.4.5.4.2	"eIP-HEADER-WRONG" parameter means that the message format is wrong as described in clause "Parameter Problem Message" of IETF RFC 792 [27], In this case, the aErrorInfo parameter shall be completed with the code value defined in clause "Parameter Problem Message" of IETF RFC 792 [27].
RQ1004_061	10.4.5.4.2	"eTERMINAL-BUSY" parameter means that terminal is currently unable to process the command as described in clause 8.12.2 of ETSI TS 102 223 [9], In this case, the aErrorInfo parameter shall be completed with the value defined in clause 8.12.2 of ETSI TS 102 223 [9].
RQ1004_062	10.4.5.4.2	"eNETWORK-BUSY" parameter means that the network is currently unable to process the command as described in clause 8.12.3 of ETSI TS 102 223 [9], In this case, the aErrorInfo parameter shall be completed with the value defined in clause 8.12.3 of ETSI TS 102 223 [9].
RQ1004_063	10.4.5.4.2	"eCALL-CONTROL-INTERACTION-ERROR" parameter means that the connection required to establish the TCP communication was blocked by the terminal due to the call control by NAA, as described in clause 7.3 of ETSI TS 102 223 [9], In this case, the aErrorInfo parameter shall be completed with the value defined in clause 8.12.8 of ETSI TS 102 223 [9].
RQ1004_064	10.4.5.4.2	"eDNS-RESOLUTION-ERROR" parameter means that the destination FQDN could not be resolved by the DNS server, In this case, the aErrorInfo parameter shall be completed with the code value defined in IETF RFC 6895 [28], clause 2.3.

RQ number	Clause	Description
RQ1004_065	10.4.5.4.2	"eLINK-DROPPED" parameter means that the Bearer Link of the TCP connection has dropped (due to network failure or user cancellation) as described in clause 7.5.11 of ETSI TS 102 223 [9].
	10.4.6.1	Overview
RQ1004_066	10.4.6.1	The pipe session is opened when a requested TCP connection is successfully established.
RQ1004_067	10.4.6.1	The TCP session shall be closed as soon as the pipe session is closed.
RQ1004_068	10.4.6.1	The TCP data service gate shall implement the credit-based data flow control mechanism and data acknowledgement defined in ETSI TS 103 666-1 [1], clause 8.5.3.
RQ1004_069	10.4.6.1	The SCL packets shall have the CB bit (Chaining bit) always set to 0.
	10.4.6.2	Commands
	10.4.6.3	Responses
	10.4.6.4	Events
	10.4.7	TCP data application gate
	10.4.7.1	Overview
RQ1004_070	10.4.7.1	The TCP data application gate shall implement the credit-based data flow control mechanism and data acknowledgement defined in ETSI TS 103 666-1 [1], clause 8.5.3.
	10.4.7.2	Commands
	10.4.7.3	Responses
	10.4.7.4	Events
	10.4.8	Application protocols
	10.4.8.1	HTTP(S) protocol
RQ1004_071	10.4.8.1	The SSP may support HTTP as defined in IETF RFC 7230 [i.5] or HTTPS as defined in IETF RFC 2818 [i.6] using the mechanism described in clause 10.4 of ETSI TS 103 666-1 [1].
	10.4.8.2	TLS protocol
RQ1004_072	10.4.8.2	The SSP may support the TLS protocol using the mechanism described in clause 10.4 of ETSI TS 103 666-1 [1]. If supported, TLS shall be compliant with IETF RFC 8446 [29].

5.6.5 User Datagram Protocol support

Reference: ETSI TS 103 666-1 [1], clause 10.5

RQ number	Clause	Description
	10.5	User Datagram Protocol support
	10.5.1	Overview
RQ1005_001	10.5.1	The UDP adapter is in charge of creating the UDP sockets;
RQ1005_002	10.5.1	The UDP adapter is in charge of the resolution of DNS address, when this is required;
RQ1005_003	10.5.1	The UDP adapter is in charge of transferring incoming UDP packets to the appropriate UDP application gate;
RQ1005_004	10.5.1	The UDP adapter is in charge of sending outgoing UDP packets.
RQ1005_005	10.5.1	The UDP adapter shall contain only one UDP service gate.
RQ1005_006	10.5.1	If an SSP host supports the UDP protocol, it shall contain only one UDP consumer, which includes a single UDP application gate.
RQ1005_007	10.5.1	The UDP service gate URN supports the syntax as defined in ETSI TS 103 666-1 [1], clause 8.2 with the values specified in ETSI TS 103 666-1 [1], Table 8.1.
RQ1005_008	10.5.1	If the pipe session between the UDP service gate and the UDP application gate is closed, all UDP sockets shall be terminated.
	10.5.2	Presentation layer
RQ1005_009	10.5.2	The UDP service and application gates implements an ASN.1 presentation layer using the definitions of the TCP service and application gates, as described in ETSI TS 103 666-1 [1], clause 10.4.3, with the additional definitions, described in ETSI TS 103 666-1 [1], clause 10.5.2
	10.5.3	UDP service gate
	10.5.3.1	Overview
	10.5.3.2	Commands
	10.5.3.2.1	List of Commands
RQ1005_010	10.5.3.2.1	"UDP service gate" shall support the commands, described in ETSI TS 103 666-1 [1], clause 10.5.3.2.1

RQ number	Clause	Description
	10.5.3.2.2	UDP-REQUEST-SOCKET-Command
RQ1005_011	10.5.3.2.2	If the port number within the command UDP-REQUEST-SOCKET-Command is not defined, the UDP adapter assigns an available port.
RQ1005_012	10.5.3.2.2	"UDP-REQUEST-SOCKET-Command" command has parameters, as defined in ETSI TS 103 666-1 [1], clause 10.5.3.2.2
RQ1005_013	10.5.3.2.2	"aPortNumber" defines the UDP port number on the terminal, If the parameter is missing, the port will be automatically allocated by the UDP adapter;
RQ1005_014	10.5.3.2.2	"aNetworkParameters" contains the parameters for the network connection required for the UDP socket to be created, The coding is the same as the NetworkParameters defined in ETSI TS 103 666-1 [1], clause 10.4.4.2.2
RQ1005_015	10.5.3.2.2	if "aLocalOnly" has value TRUE, then the UDP socket can only accept UDP datagrams from entities in the terminal. If "aLocalOnly" has value FALSE or it is not present, then the UDP socket can accept UDP datagrams from any remote entity.
RQ1005_016	10.5.3.2.2	When the requested socket is created successfully, then UDP service gate shall respond with eUDP-OK with the parameters, as described in ETSI TS 103 666-1 [1], clause 10.5.3.2.2
RQ1005_017	10.5.3.2.2	"aSocketID" identifier shall be unique across all UDP sockets.
	10.5.3.2.3	UDP-CLOSE-SOCKET-Command
RQ1005_018	10.5.3.2.3	"UDP-CLOSE-SOCKET-Command" command has parameters, as defined in ETSI TS 103 666-1 [1], clause 10.5.3.2.3
RQ1005_019	10.5.3.2.3	When successful the UDP application gate shall respond with eUDP-OK with parameters, as defined in ETSI TS 103 666-1 [1], clause 10.5.3.2.3
	10.5.3.3	Responses
RQ1005_020	10.5.3.3	"UDP Service Gate" shall support the responses, as defined in ETSI TS 103 666-1 [1], clause 10.5.3.3
RQ1005_021	10.5.3.3	"UDP Service Gate" shall support the error codes, as defined in ETSI TS 103 666-1 [1], Table 10.11
	10.5.3.4	Events
	10.5.3.4.1	List of events
RQ1005_022	10.5.3.4.1	The UDP service gate supports the events, as described in ETSI TS 103 666-1 [1], clause 10.5.3.4.1
	10.5.3.4.2	EVT-UDP-DATAGRAM-OUT-Service-Event
RQ1005_023	10.5.3.4.2	The UDP consumer may request to send the UDP datagram by passing the IP address or the FQDN value of the server, using the coding, as described in ETSI TS 103 666-1 [1], clause 10.5.3.4.2
RQ1005_024	10.5.3.4.2	If the FQDN value is used for the UDP datagram within the event EVT-UDP-DATAGRAM-OUT-Service-Event, the UDP adapter is responsible to perform the DNS resolution, if the feature is supported.
	10.5.4	UDP application gate
	10.5.4.1	Overview
	10.5.4.2	Commands
	10.5.4.3	Responses
	10.5.4.4	Events
	10.5.4.4.1	List of events
RQ1005_025	10.5.4.4.1	The UDP application gate supports the events, as described in ETSI TS 103 666-1 [1], clause 10.5.4.4.1
	10.5.4.4.2	EVT-UDP-DATAGRAM-IN-Application-Event
RQ1005_026	10.5.4.4.2	With the event EVT-UDP-DATAGRAM-IN-Application-Event, the UDP adapter via the UDP service gate conveys to the UDP consumer via the UDP application gate a datagram received on an open UDP socket, using coding as defined in ETSI TS 103 666-1 [1], clause 10.5.4.4.2
	10.5.4.4.3	EVT-UDP-ERROR-Application-Event
RQ1005_027	10.5.4.4.3	With the event EVT-UDP-ERROR-Application-Event, the UDP adapter via the UDP service gate notifies the UDP consumer via the UDP application gate that an error occurred, using coding as defined in ETSI TS 103 666-1 [1], clause 10.5.4.4.3
	10.5.5	Application protocols
	10.5.5.1	CoAP over UDP Protocol
RQ1005_00X	10.5.5.1	The SSP may support CoAP over UDP as defined in IETF RFC 7252 [**i.7**] using the mechanism described in the clauses above.

5.6.6 CRON service support

Reference: ETSI TS 103 666-1 [1], clause 10.6

RQ number	Clause	Description
	10.6.1	Overview
RQ1006_001	10.6.1	If the CRON service is supported by an SCL host outside the SSP host domain, then it shall contain only one CRON service gate.
RQ1006_002	10.6.1	An SCL host residing in the SSP host domain may contain one CRON application gate and shall not have any CRON service gate.
RQ1006_003	10.6.1	The CRON service gate URN supports the syntax as defined in clause 8.2 of ETSI TS 103 666-1 [1] with the values specified in Table 8.1.
	10.6.2	Presentation layer
RQ1006_004	10.6.2	The CRON service gate and the CRON application gates implements an ASN.1 presentation layer using definitions in clause 10.6.2 of ETSI TS 103 666-1 [1]
	10.6.3	CRON service gate
	10.6.3.1	Overview
RQ1006_005	10.6.3.1	The time information used by the CRON service may not be reliable or accurate. SSP applications shall not rely on the time and date provided by the CRON service if they need an accurate source of time.
	10.6.3.2	Commands
	10.6.3.2.1	List of commands
	10.6.3.2.2	CRON-REQUEST-TIMER-Command
RQ1006_006	10.6.3.2.2	With the command CRON-REQUEST-TIMER-Command, an SSP Application within the SSP host requests the CRON service to create a timer, in order to be notified when it expires.
RQ1006_007	10.6.3.2.2	CRON-REQUEST-TIMER-Command shall contain either aDateTimeAbsolute or aTimeRelative as time for the initial notification
RQ1006_008	10.6.3.2.2	CRON-REQUEST-TIMER-Command may contain aPeriod as the interval for periodic notification
RQ1006_009	10.6.3.2.2	If the SSP Application requests the timer at an absolute time and the CRON Service does not support it, then the CRON service shall reject the CRON-REQUEST-TIMER-Command, responding back eCRON-E-NO-ABSOLUTE-TIME.
RQ1006_010	10.6.3.2.2	If the SSP Application requests the timer at an absolute time in the past, then the CRON service shall reject the CRONREQUEST-TIMER-Command, responding back eCRON-E-NOK.
RQ1006_011	10.6.3.2.2	When the CRON request is successful then CRON service gate shall respond with eCRON-OK and shall contain the parameter CRON-ID
RQ1006_012	10.6.3.2.2	When the CRON request is successful then CRON service gate shall respond with eCRON-OK and may contain the parameter aPersistantOverPowerCycle
	10.6.3.2.3	CRON-READ-DATE-TIME-Command
RQ1006_013	10.6.3.2.3	With the command CRON-READ-DATE-TIME-Command a CRON application gate may request to retrieve the UTCTime of the request
RQ1006_014	10.6.3.2.3	When successful the CRON Service shall respond with eCRON-OK with CRON-READ-DATE-TIME-Response which shall contain CRON-READ-DATE-TIME-Response-Parameter
RQ1006_015	10.6.3.2.3	CRON-READ-DATE-TIME-Response-Parameter shall contain aDateTime GeneralizedTime
	10.6.3.2.4	CRON-KILL-TIMER-Command
RQ1006_016	10.6.3.2.4	With the command CRON-KILL-TIMER-Command a CRON application gate may requests to kill a timer previously registered in the CRON service.
RQ1006_017	10.6.3.2.4	CRON-KILL-TIMER-Command shall contain aCRON-ID: identifier of the timer to kill.
RQ1006_018	10.6.3.2.4	When the CRON-KILL-TIMER-Command is successful the CRON application gate shall respond with eCRON-OK
	10.6.2.3.5	CRON-KILL-ALL-TIMERS-Command
RQ1006_019	10.6.3.2.5	With the command CRON-KILL-ALL-TIMERS-Command a CRON application gate may request to kill all timers registered by a SSP host.
RQ1006_020	10.6.3.2.5	When the CRON-KILL-ALL-TIMERS-Command is successful the CRON application gate shall respond with eCRON-OK with no parameters.
	10.6.3.3	Responses
RQ1006_021	10.6.3.3	The error codes in the responses of the CRON service shall be only one of those listed in Table 10.13 of ETSI TS 103 666-1 [1].
	10.6.3.4	Events
	10.6.4	CRON application gate
	10.6.4.1	Commands
	10.6.4.2	Responses
	10.6.4.3	Events

RQ number	Clause	Description
	10.6.4.3.1	List of events
RQ1006_022	10.6.4.3.1	The CRON application gate shall support the CRON-ELAPSED-TIMER-Event event.
	10.6.4.3.2	CRON-ELAPSED-TIMER-Event
RQ1006_023	10.6.4.3.2	CRON-ELAPSED-TIMER-Event event shall contain aCRON-Id as the identifier of the timer that has elapsed.

5.6.7 Contactless related applications support

Reference: ETSI TS 103 666-1 [1], clause 10.7

RQ number	Clause	Description
	10.7.2	HCP tunnelling over SCL
	10.7.2.1	Overview
RQ1007_001	10.7.2.1	A SCL pipe session allows the tunnelling of HCP packets as defined in ETSI TS 102 622 [5] to/from HCI host controller as defined in ETSI TS 102 622 [5].
RQ1007_002	10.7.2.1	The SSP host shall at most support a single pipe session to HCI service gate.
RQ1007_003	10.7.2.1	The presentation layer of the message conveyed over SCL is the HCP as defined in ETSI TS 102 622 [5], clause 5.1
RQ1007_004	10.7.2.1	The session of the HCI protocol uses the session initialization defined in ETSI TS 102 622 [5], clause 8.4 with the assumption that the outcome of the identity check mechanism of the HCI lower layers is always successful.
	10.7.2.2	SCL HCI service gate
RQ1007_005	10.7.2.2	The SCL HCI service gate provides access to a SCL HCI fragmentation and reassembly service that manages the transfer of HCP packets as defined in ETSI TS 102 622 [5] from/to a CLF compliant with the ETSI TS 102 622 [5].
RQ1007_006	10.7.2.2	The SCL HCI Service shall embed the HCP packet from the HCI Host Controller in SCL message fragments of SCL packet to the SCL HCI application gate towards the SCL HCI application in the SSP host.
	10.7.2.3	SCL HCI application gate
RQ1007_007	10.7.2.3	The SCL HCI application gate provides access to a SCL HCI application that emulates an HCI host as defined in ETSI TS 102 622 [5].
RQ1007_008	10.7.2.3	The SCL HCI application shall reassembly the HCP packet as defined in ETSI TS 102 622 [5] from the message fragments of SCL packets for the HCI host in the SSP host.

5.6.8 Card Application Toolkit (CAT) over SCL

Reference: ETSI TS 103 666-1 [1], clause 10.8

RQ number	Clause	Description
	10.8.1	Overview
RQ1008_001	10.8.1	If the CAT application gate is supported, the CAPABILITY_EXCHANGE entry in the identity gate registry of the SCL host in the SSP shall indicate support for Card Application Toolkit.
RQ1008_002	10.8.1	The SSP host shall contain no more than one CAT application gate.
RQ1008_003	10.8.1	Each SSP host shall create no more than one pipe session to CAT service gates.
RQ1008_004	10.8.1	If there are multiple hosts supporting the CAT service gate, the SSP host shall use the host in the MBM host domain, if present.
RQ1008_005	10.8.1	The communication between the CAT service gate and the CAT application gate shall use the presentation layer defined in ETSI TS 102 622 [5], clause 5.2.
RQ1008_006	10.8.1	When the SSP host has a pipe session to a CAT service gate, the SSP shall behave as described in ETSI TS 103 666-1 [1], clause 6.8.1.
	10.8.2	Structure of Card Application Toolkit (CAT) communications
RQ1008_007	10.8.2	CAT commands and responses are sent over the SCL pipe as BER-TLV data objects.
RQ1008_008	10.8.2	The tag of a BER-TLV is a constant value, length one byte, indicating it is a CAT command.
RQ1008_009	10.8.2	The length shall be coded onto 1, 2 or 3 bytes according to ETSI TS 101 220 [11], clause 7.1.2. ETSI TS 103 666-1 [1], Table 10.15 details this coding.
RQ1008_010	10.8.2	Any values for byte 1, byte 2 or byte 3 that are not shown in ETSI TS 103 666-1 [1], Table 10.15 shall be treated as an error and the whole message shall be rejected.
RQ1008_011	10.8.2	It is mandatory for COMPREHENSION-TLV data objects to be provided in the order given in the description of each command.

RQ number	Clause	Description
RQ1008_012	10.8.2	New COMPREHENSION-TLV data objects can be added to the end of a command.
	10.8.3	CAT application gate
	10.8.3.1	Overview
RQ1008_013	10.8.3.1	The events defined in ETSI TS 103 666-1 [1], clause 10.8.3 shall be sent to CAT Application gate.
	10.8.3.2	Commands
	10.8.3.3	Responses
	10.8.3.4	Events
	10.8.3.4.1	Supported events
RQ1008_014	10.8.3.4.1	The CAT application gate supports the events listed in ETSI TS 103 666-1 [1], Table 10.16.
	10.8.3.4.2	EVT_ENVELOPE_CMD
RQ1008_015	10.8.3.4.2	EVT_ENVELOPE_CMD event shall be used by the host outside the SSP in order to send an envelope command (as defined in ETSI TS 102 223 [9]) to the CAT application gate.
RQ1008_016	10.8.3.4.2	The contents of the parameter of event "EVT_ENVELOPE_CMD" are as defined in ETSI TS 102 223 [9], clause 7, with the exception of length parameters which is described in ETSI TS 103 666-1 [1], clause 10.8.2.
	10.8.3.4.3	EVT_TERMINAL_RESPONSE
RQ1008_017	10.8.3.4.3	EVT_TERMINAL_RESPONSE event shall be used by the host outside the SSP in order to send a terminal response (as defined in ETSI TS 102 223 [9]) to the CAT application gate.
RQ1008_018	10.8.3.4.3	The contents of the parameter of event "EVT_TERMINAL_RESPONSE" are as defined in ETSI TS 102 223 [9], clause 6.8, with the exception of length parameters which is described in ETSI TS 103 666-1 [1], clause 10.8.2.
	10.8.3.5	Registry
	10.8.4	CAT service gate
	10.8.4.1	Overview
	10.8.4.2	Commands
	10.8.4.3	Responses
	10.8.4.4	Events
	10.8.4.4.1	Supported events
RQ1008_019	10.8.4.4.1	The events defined in ETSI TS 103 666-1 [1], clause 10.8.4 shall be sent to CAT Service gate.
RQ1008_020	10.8.4.4.1	The CAT service gate URN supports the syntax as defined in ETSI TS 103 666-1 [1], clause 8.2, with the values specified in ETSI TS 103 666-1 [1], Table 8.1.
RQ1008_021	10.8.4.4.1	The CAT service gate supports the events listed in ETSI TS 103 666-1 [1], Table 10.19.
	10.8.4.4.2	EVT_PROACTIVE_CMD
RQ1008_022	10.8.4.4.2	"EVT_PROACTIVE_CMD" event shall be used by the SSP host in order to send a proactive command (as defined in ETSI TS 102 223 [9]) to the CAT service gate.
RQ1008_023	10.8.4.4.2	The contents of the parameter of "EVT_PROACTIVE_CMD" shall be as defined in ETSI TS 102 223 [9], clause 6.6, with the exception of length parameters which is described in ETSI TS 103 666-1 [1], clause 10.8.2.
	10.8.4.4.3	EVT_ENVELOPE_RSP
RQ1008_024	10.8.4.4.3	EVT_ENVELOPE_RSP event shall be used by the SSP host in order to send an envelope response to the CAT service gate.
RQ1008_025	10.8.4.4.3	The contents of "EVT_ENVELOPE_RSP" shall contain an optional response payload followed by SW1/SW2 status words as defined in ETSI TS 102 223 [9], clause 7. The length parameters for the optional response payload shall be as described in ETSI TS 103 666-1 [1], clause 10.8.2.
	10.8.4.5	Registry
	10.8.5	State diagram for the CAT application gate
RQ1008_026	10.8.5	The states of the CAT application gate are: • TK_ST_INIT: state of the gate when an open pipe exists to the gate but Capability Exchange indicates no terminal support for Card Application Toolkit. • TK_ST_IDLE: state of the gate when no Toolkit commands are being processed. • TK_ST_PCMD: state of the gate when one or more proactive commands are sent out and the terminal response is not yet received for all. • TK_ST_ENV (transient): state of the gate when handling an envelope command.
RQ1008_027	10.8.5	The CAT application gate shall only send proactive commands when it is in TK_ST_IDLE state.

5.6.9 Access Control Protocol

Reference: ETSI TS 103 666-1 [1], clause 10.9

RQ number	Clause	Description
	10.9.1	Introduction
RQ01091_01	10.9.1	The SSP shall have a dedicated accessor authentication service gate for each accessor.
RQ01091_02	10.9.1	The accessor authentication service gate is automatically created at the creation of the accessor.
RQ01091_03	10.9.1	The identifier of the gate has the same value of the accessor identity, as defined in ETSI TS 103 666-1 [1], clause 6.13.4.3.
RQ01091_04	10.9.1	Hosts outside the SSP host domain may contain one or more accessor authentication application gates.
RQ01091_05	10.9.1	The authentication of an accessor using a given pipe session shall not imply the authentication of the same accessor for a different host.
RQ01091_06	10.9.1	The closure of the pipe session to an accessor authentication service gate where the accessor was successfully authenticated shall result in the fact that the corresponding accessor is no longer authenticated for the host, without any impact on the authentication status of the same accessor for other hosts.
RQ01091_07	10.9.1	The pipe sessions created using the accessor authentication service shall remain open.
RQ01091_08	10.9.1	After an accessor is successfully authenticated using a pipe session, it may be used to grant permissions to other accessors authenticated from any host, using the mechanism described in ETSI TS 103 666-1 [1] clause 6.13.2.
RQ01091_09	10.9.1	All pipe sessions to an accessor authentication service gate shall be closed by the accessor authentication service when the corresponding accessor is deleted.
RQ01091_10	10.9.1	The accessor authentication service shall remove the accessor authentication service gate after closing the pipe sessions.
RQ01091_11	10.9.1	If the credentials and/or conditions of an accessor are modified, the authentication status of the accessor is not affected.
	10.9.2	Presentation layer
RQ01092_01	10.9.2	The accessor authentication control service gate and the accessor authentication control application gate implement an ASN.1 presentation layer using the definitions in ETSI TS 103 666-1 [1] clause 6.13.5.
	10.9.3	Accessor authentication service gate
	10.9.3.1	Overview
RQ01093_01	10.9.3.1	An accessor authentication application may access to the accessor authentication service via a pipe session between an accessor authentication application gate and an accessor authentication service gate.
	10.9.3.2	Commands
	10.9.3.3	Responses
	10.9.3.4	Events
	10.9.4	Accessor authentication application gate
	10.9.4.1	Overview
	10.9.4.2	Commands
	10.9.4.3	Responses
	10.9.4.4	Events

5.7 Requirements not testable, implicitly verified or verified elsewhere

5.7.1 Requirements implicitly tested

The following requirements identified in <XYZ> are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0501_001, RQ0501_002

RQ0502_001

RQ0504_001, RQ0504_002, RQ0504_003, RQ0504_004, RQ0504_005, RQ0504_006, RQ0504_007, RQ0504_008

RQ0505_001, RQ0505_002

6 Test Descriptions: SSP Characteristics

6.1 Form Factors

6.1.1 Requirements not testable, implicitly verified or verified elsewhere

6.1.1.1 Requirements not tested

The following requirements identified in clause 5.2.1 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible: RQ0601_001.

6.2 Power

6.2.1 Requirements not testable, implicitly verified or verified elsewhere

6.2.1.1 Requirements not tested

The following requirements identified in clause 5.2.2 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ0602_001, RQ0602_002, RQ0602_003, RQ0602_004, RQ0602_005, RQ0602_007, RQ0602_008, RQ0602_011.

6.2.1.2 Requirements verified elsewhere

The following requirements identified in clause 5.2.2 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0602_006, RQ0602_009, RQ0602_010, RQ0602_012, RQ0602_013.

6.3 Clock

6.3.1 Requirements not tested

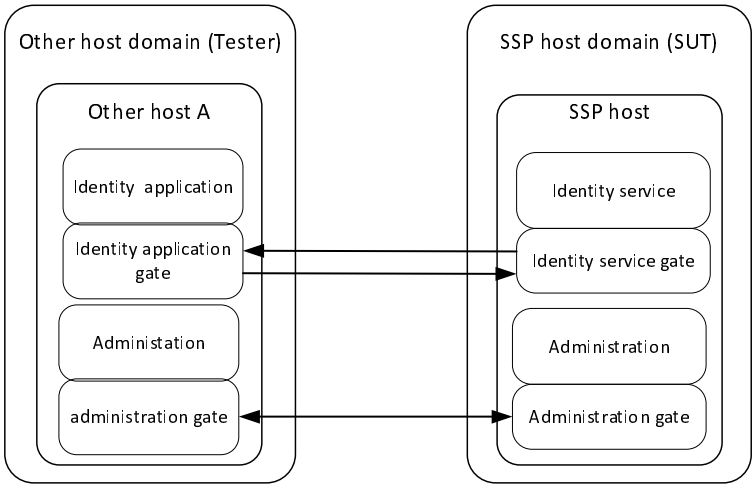
The following requirements identified in clause 5.2.3 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ0603_001, RQ0603_002, RQ0603_003, RQ0603_004.

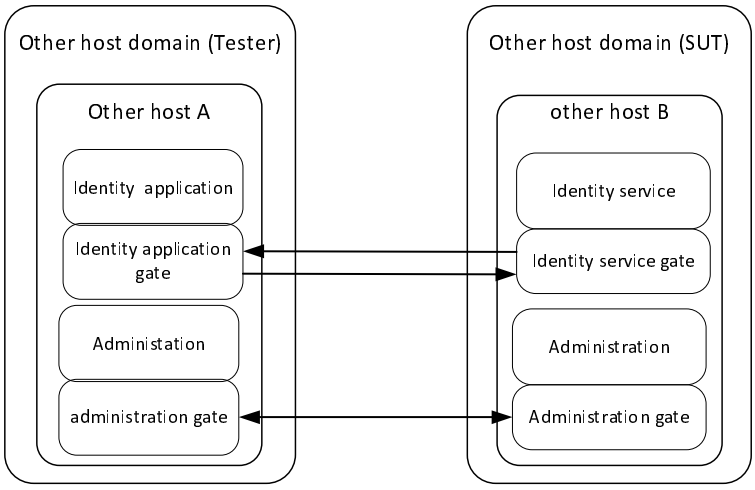
6.4 SSP Initialization

6.4.1 Configurations

6.4.1.1 CINI_001

Configuration identification	CINI_001
Configuration description	

6.4.1.2 CINI_002

Configuration identification	CINI_002
Configuration description	

6.4.1.3 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPINIconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) initialization (1)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   SSPClass ,
   SSPCapability,
   TerminalCapability,
VersionType
FROM SSPDefinitions;
-- ASN1STOP
```

6.4.2 Procedures

6.4.2.1 PINI_001 - Open a pipe session with the Identity gate of the SSP host

Procedure identification	PINI_001
Procedure objectives	The other host shall be able to open a pipe session to the identity gate of the SSP host.
Configuration reference	CINI_001
Initial conditions	
The SSP host is registered to the SCL network controller host.	
Procedure sequence	
Step	**Description**
1	Administration gate in the other host sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate in the SSP host sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).

6.4.2.2 PINI_002 - Open a pipe session with the Identity gate of the Terminal host

Procedure identification	PINI_002
Procedure objectives	The SSP host shall be able to open a pipe session to the identity gate of the Terminal host.
Configuration reference	CINI_002
Initial conditions	
The Terminal host is registered to the SCL network controller host.	
Procedure sequence	
Step	**Description**
1	Administration gate in the host A sends EVT_ADM_BIND to Administration gate in the host B (SUT) with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate in the host B (SUT)sends EVT_ADM_BIND to Administration gate in the host A (tester) with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).

6.4.3 Test descriptions

6.4.3.1 INI_001 - Capability Exchange of SSPCapabilities

Test identification	INI_001
Test objectives	To test that the capability exchange procedure is performed when another host supporting SCL is registered on the SCL network controller host.
Configuration reference	CINI_001
Initial conditions	
The procedure PINI_001 is successfully executed.	

```
-- ASN1START
aTrue BOOLEAN ::= TRUE    /*<STORE(aTrue)>*/
aFalse BOOLEAN ::= FALSE    /*<STORE(aFalse)>*/

aEMPTY_1 UTF8String ::= "" /*<STORE(aEMPTY_1)>*/
aEMPTY_2 OCTET STRING ::= ''H /*<STORE(aEMPTY_2)>*/
aSSPRELEASE VersionType::= '0F00'H /* <STORE(aSSPRELEASE)> *//* it indicates the release of the
present document that is implemented by the SSP*/
aSSPCLASS_1 SSPClass ::= eSSPClass-Integrated /* <STORE(aSSPCLASS_1)> */
aSSPCLASS_2 SSPClass ::= eSSPClass-Embedded-Type1 /* <STORE(aSSPCLASS_2)> */
aSSPCLASS_3 SSPClass ::= eSSPClass-Embedded-Type2 /* <STORE(aSSPCLASS_3)> */
aSSPCLASS_4 SSPClass ::= eSSPClass-Removable /* <STORE(aSSPCLASS_4)> */

aNBLOGICALCHANNELS_MIN INTEGER ::= 1 /* <STORE(aNBLOGICALCHANNELS_MIN)> *//* it indicates the
minimum nb of logical channels, including the default channel, that can be supported by an SSP*/
aNBLOGICALCHANNELS_MAX INTEGER ::= 14 /* <STORE(aNBLOGICALCHANNELS_MAX)> *//* it indicates the
maximum nb of logical channels, including the default channel, that can be supported by an SSP*/
-- ASN1STOP
```

Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate sends ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	

2	The Identity service gate sends aResponse to the Identity application gate.
	`-- ASN1START`
	`aResponse SSPCapability ::= {`
	`  aSspRelease '0000'H,`
	`/*<COMPARE(aSSPRELEASE,GT,EQ)>*/`
	`  aSspVendorName "0",`
	`/*<ISFIELDNOTEXIST()> OR <COMPARE(aEMPTY_1,DIF)>*/`
	`  aSspClass eSSPClass-Integrated /*<COMPARE(aSSPCLASS_1,EQ)> OR`
	`<COMPARE(aSSPCLASS_2,EQ)> OR <COMPARE(aSSPCLASS_3,EQ) OR`
	`<COMPARE(aSSPCLASS_4,EQ)>*/,`
	`  aClassSpecificCapabilities OCTET STRING : '00'H /*<ISFIELDNOTEXIST()> OR`
	`<COMPARE(aEMPTY_2,DIF)>*/,`
	`  aSspUicc {`
	`    aNumberOfLogicalChannels 1,`
	`/*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aNBLOGICALCHANNELS_MIN,EQ,GT)> AND`
	`<COMPARE(aNBLOGICALCHANNELS_MAX,EQ,LS)>`
	`*/`
	`    aProactivePollingRequirement FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse,EQ)> */`
	`    aSupportOfUiccFileSystem FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse,EQ)> */`
	`    aSupportOfCardApplicationToolkit FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse, EQ)> */`
	`    aCardApplicationToolkitCapabilities '00'H`
	`/*<ISFIELDNOTEXIST> OR <COMPARE(aEMPTY_2,DIF)>*/`
	`  },`
	`  aSspUserInterface {`
	`    aUrl '00'H`
	`/*<ISFIELDNOTEXIST OR <COMPARE(aEMPTY_1,DIF)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0604_003
		RQ0604_004
		RQ0604_005
		RQ0604_011
		RQ0604_012
		RQ0604_013
		RQ0604_014
		RQ0604_015
		RQ0606_001

6.4.3.2 INI_002 - Capability Exchange of TerminalCapabilities

Test identification	INI_002
Test objectives	To test that the capability exchange procedure is performed when a Terminal host supporting SCL is registered on the SCL network controller host.
Configuration reference	CINI_002

Initial conditions

The procedure PINI_002 is successfully executed.

```
-- ASN1START

aEMPTY_1 UTF8String ::= "" /*<STORE(aEMPTY_1)>*/
aEMPTY_2 OCTET STRING ::= ''H /*<STORE(aEMPTY_2)>*/
aTERMINALRELEASE VersionType::= '0F00'H /* <STORE(aTERMINALRELEASE)> *//* it indicates the release
of the present document that is implemented by the Terminal*/
aINTERFACEPOWERSUPPLY INTEGER ::= 0 /*<STORE(aINTERFACEPOWERSUPPLY> *//* it indicates the maximum
current that the terminal can provide over the physical interface where the Capability Exchange
procedure is performed*/
aEXTERNALPOWERSUPPLY INTEGER ::= 0 /*<STORE(aEXTERNALPOWERSUPPLY> *//* it indicates the maximum
current provided by the terminal using the external power supply*/
-- ASN1STOP
```

Test sequence

Step	Description	Requirements
1	Identity application gate sends ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	

2	The Identity service gate sends aResponse to the Identity application gate.
	`-- ASN1START`
	`aResponse TerminalCapability ::= {`
	`  aTerminalRelease '0000'H,`
	`/*<COMPARE(aTERMINALRELEASE,GT,EQ)>*/`
	`  aTerminalVendorName "0",`
	`/*<ISFIELDNOTEXIST()> OR <COMPARE(aEMPTY_1,DIF)>*/`
	`  aInterfacePowerSupply 0,`
	`/*<COMPARE(aINTERFACEPOWERSUPPLY,EQ,GT)>*/`
	`  aExternalPowerSupply 0,`
	`/*<COMPARE(aEXTERNALPOWERSUPPLY,EQ,GT)>*/`
	`  aToolkitTerminalProfile '00'H`
	`/*<ISFIELDNOTEXIST> OR <COMPARE(aEMPTY_2,DIF)>*/`
	`}`
	`-- ASN1STOP`
		RQ0604_003
		RQ0604_004
		RQ0604_005
		RQ0604_006
		RQ0604_007
		RQ0604_008
		RQ0604_009
		RQ0604_010

6.4.3.3 End of test descriptions - INITIALIZATION ASN.1 descriptions

6.4.3.3.1 Annex - End of ASN.1 structure

```
-- ASN1START
END
-- ASN1STOP
```

6.4.3.4 Implicitly tested requirements

The following requirements identified in clause 5.2.4 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0604_001.

6.5 Storage

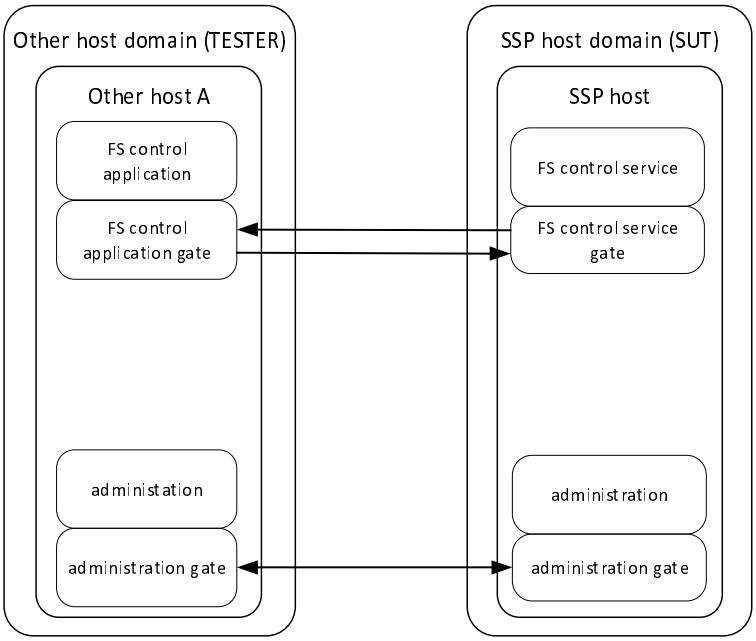
6.5.1 Requirements not tested

The following requirements identified in clause 5.2.5 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible: RQ0605_001, RQ0605_002.

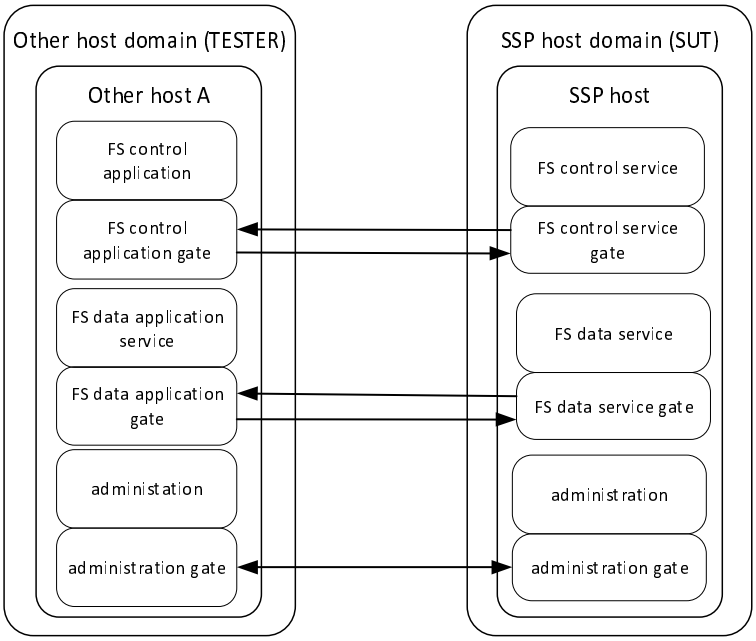
6.6 SSP File System

6.6.1 Configurations

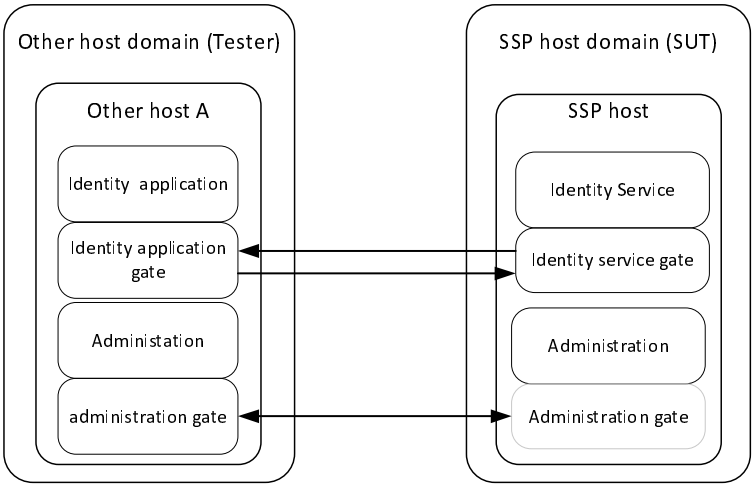
6.6.1.1 CFSS_001

Configuration identification	CFSS_001
Configuration description	

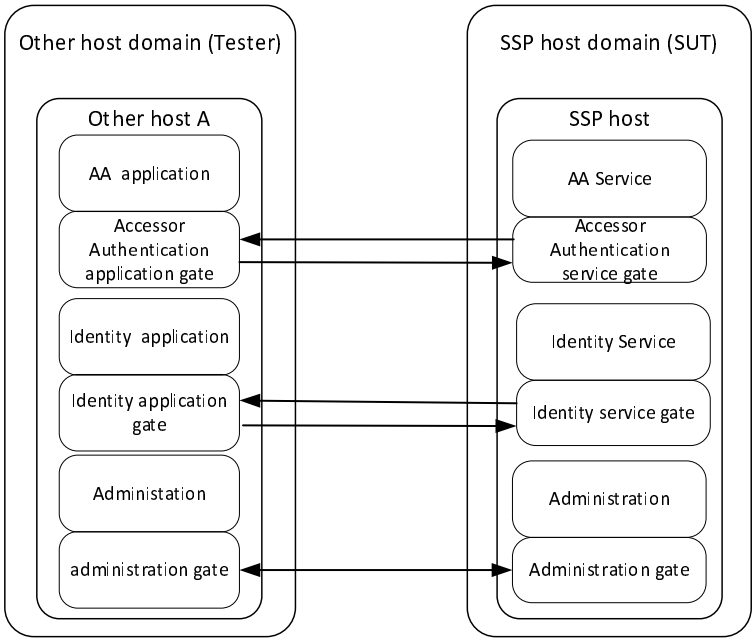
6.6.1.2 CFSS_002

Configuration identification	CFSS_002
Configuration description	

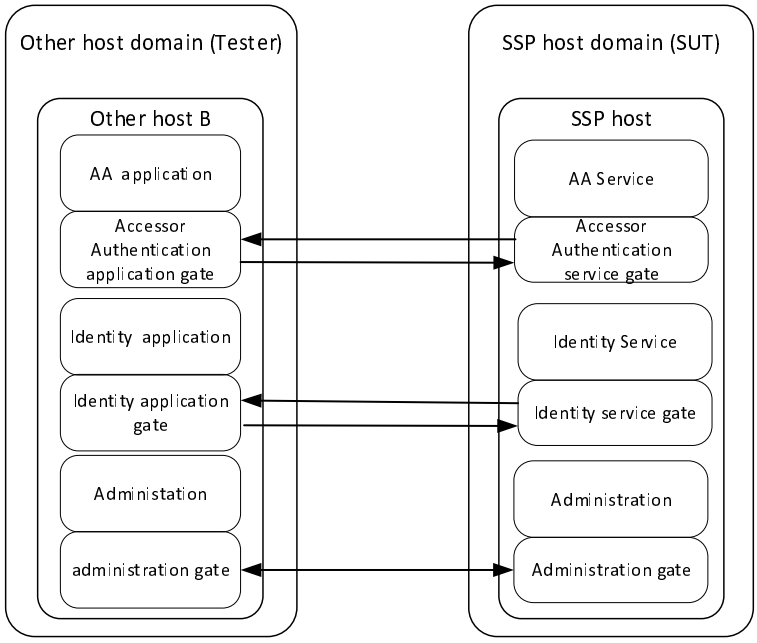
6.6.1.3 CFSS_003

Configuration identification	CFSS_003
Configuration description	

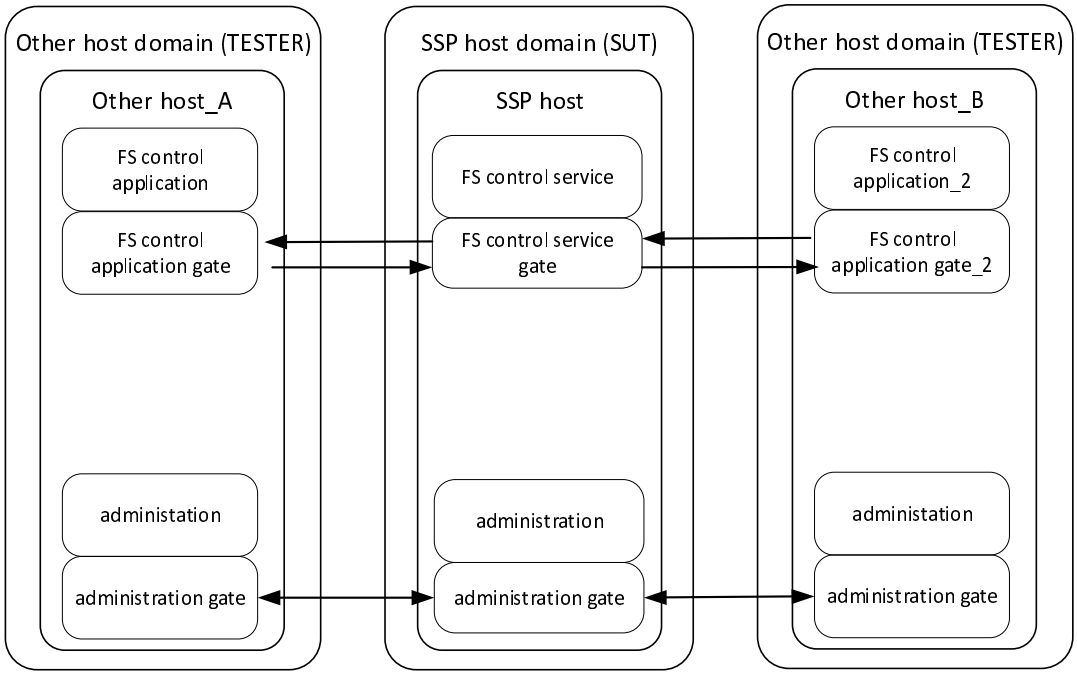
6.6.1.4 CFSS_004

Configuration identification	CFSS_004
Configuration description	

6.6.1.5 CFSS_005

Configuration identification	CFSS_005
Configuration description	The same configuration as CFSS_004 but with another Other host: Other Host_B. 

6.6.1.6 CFSS_006

Configuration identification	CFSS_006
Configuration description	

6.6.1.7 ASN.1 Configuration

The following configuration is used for the procedures and the test descriptions.

```
-- ASN1START
SSPFSconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) fs (2)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN

EXPORTS ALL;

/* Imports */
IMPORTS
   NodeName,  -- RFC5280 Certificate X.509v3
   FileSize,
   AccessMode,
   UUID,
   SessionID,
   AccessorRights,
   AccessorConditionsPIN,
   AccessorConditions,
   AAS-CONTROL-SERVICE-GATE-Commands,
   AAS-CONTROL-SERVICE-GATE-Responses,
   FS-CONTROL-SERVICE-GATE-Commands,
   FS-CONTROL-SERVICE-GATE-Responses,
   Certificate,
   AuthenticationToken,
   Version
   FROM SSPDefinitions
ECDSA-Sig-Value,
id-ecPublicKey
   FROM PKIX1Algorithms88;
```

```
eFS-Name-SSPFS NodeName ::="SSPFS"
eFS-Name-file1 NodeName ::="file1"
eFS-Name-file2 NodeName ::="file2"
eFS-Name-file3 NodeName ::="file3"
eFS-Name-file4 NodeName ::="file4"
eFS-Name-file5 NodeName ::="file5"
eFS-Name-file6 NodeName ::="file6"
eFS-Name-file7 NodeName ::="file7"
eFS-Name-file8 NodeName ::="file8"
eFS-Name-filelongfilename NodeName ::="filelongfilename"
eFS-Name-link1 NodeName ::="link1"
eFS-Name-directory1 NodeName ::="directory1"
eFS-Name-directory2 NodeName ::="directory2"
eFS-Name-directory3 NodeName ::="directory3"
eFS-Name-directory4 NodeName ::="directory4"

eFS-ID-SSPFS         UUID::='B8B7F613E7F45C9CA96EBC4BCA1B5A5C'H
eFS-ID-directory1    UUID::='805B48D9A392523BA44C1DBEB35FC2B6'H
eFS-ID-directory2    UUID::='2B80EFE42F1C534395578EAA2ECC9DD8'H
eFS-ID-directory3    UUID::='EF51886AB542579A8E52FD1A67B52C8A'H
eFS-ID-directory4    UUID::='E74184B62A9B588EB78739A7A5C2DE3B'H
eFS-ID-file1         UUID::='D44BD2F74D0B597BB70F2C66F2BE5F9B'H
eFS-ID-file2         UUID::='F979107AD6BF5743B552869717C35433'H
eFS-ID-file3         UUID::='64EAFD2989875036B664DE81BA17DCF5'H
eFS-ID-file4         UUID::='4831D4AE7B70566E939F05AC9F65C1AF'H
eFS-ID-file5         UUID::='3C5DD13F2CB050A8BCA1BE25040E6E3E'H
eFS-ID-file6         UUID::='3B968F5DFADC5CCB96B52542036EC8B9'H
eFS-ID-file7         UUID::='8055D64E010D55C4AC91250D3C1A998B'H
eFS-ID-file8         UUID::='3B851B68EFB058FBB70F1D7ED59A98F6'H
eFS-ID-filelongfilename UUID::='78D7BAFF1407582C88AF14939B139B8F'H
eFS-ID-link1         UUID::='C51F3C1E96F35ABFB24993408C998A25'H
eFS-ID-FSCS          UUID::='366BD642D7DE584ABD3BA3DCE29FC075'H -- ETSI FS control service
identifier
eFS-ID-file-fake        UUID::='00000000000000000000000000000000'H

--                   eRight-Bit1, eFSAccessRight-RequiresSecurePipe,
--                   eRight-Bit2, eFSAccessRight-ReadContent,
--                   eRight-Bit3, eFSAccessRight-GetInfo,
--                   eRight-Bit4, eFSAccessRight-Write,
--                   eRight-Bit5, eFSAccessRight-UpdateMetadata,
--                   eRight-Bit6, eFSAccessRight-UpdateACL,
--                   eRight-Bit7, eFSAccessRight-Delete,
--                   eRight-Bit8  eFSAccessRight-DeleteChild
eFS-ACL-SSPFS        AccessorRights ::= {
                     eRight-Bit2, --eFSAccessRight-ReadContent,
                     eRight-Bit3, --eFSAccessRight-GetInfo,
                     eRight-Bit4, --eFSAccessRight-Write,
                     eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                     eRight-Bit6, --eFSAccessRight-UpdateACL,
                     eRight-Bit7, --eFSAccessRight-Delete,
                     eRight-Bit8  --eFSAccessRight-DeleteChild
                     }

eFS-ACL-directory1         AccessorRights ::= {
                     eRight-Bit2, --eFSAccessRight-ReadContent,
                     eRight-Bit3, --eFSAccessRight-GetInfo,
                     eRight-Bit4, --eFSAccessRight-Write,
                     eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                     eRight-Bit6, --eFSAccessRight-UpdateACL,
                     eRight-Bit7 --eFSAccessRight-Delete,

                     }

eFS-ACL-directory2         AccessorRights ::= {
                     eRight-Bit2, --eFSAccessRight-ReadContent,
                     eRight-Bit3, --eFSAccessRight-GetInfo,
                     eRight-Bit4, --eFSAccessRight-Write,
                     eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                     eRight-Bit6, --eFSAccessRight-UpdateACL,
                     eRight-Bit7, --eFSAccessRight-Delete,
                     eRight-Bit8  --eFSAccessRight-DeleteChild
                     }
eFS-ACL-directory3         AccessorRights ::= {
                     eRight-Bit2, --eFSAccessRight-ReadContent,
                     eRight-Bit3, --eFSAccessRight-GetInfo,
                     eRight-Bit4, --eFSAccessRight-Write,
                     eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                     eRight-Bit6, --eFSAccessRight-UpdateACL,
```

```
                    eRight-Bit7, --eFSAccessRight-Delete,
                    eRight-Bit8  --eFSAccessRight-DeleteChild
                    }
eFS-ACL-directory4        AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7, --eFSAccessRight-Delete,
                    eRight-Bit8  --eFSAccessRight-DeleteChild
                    }
eFS-ACL-file1          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file2          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file3          AccessorRights ::= {

                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file4          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file5          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file6          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5 --eFSAccessRight-UpdateMetadata
                    }
eFS-ACL-file7          AccessorRights ::= {
                    eRight-Bit1, --eFSAccessRight-RequiresSecurePipe,
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-file8          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }
eFS-ACL-link1          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
```

```
                    eRight-Bit6, --eFSAccessRight-UpdateACL,
                    eRight-Bit7 --eFSAccessRight-Delete
                    }

eFS-ACL-filelongfilename  AccessorRights::={
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit5 --eFSAccessRight-UpdateMetadata,
                    }

eFS-ACL-file_upd          AccessorRights ::= {
                    eRight-Bit2, --eFSAccessRight-ReadContent,
                    eRight-Bit3, --eFSAccessRight-GetInfo,
                    eRight-Bit4, --eFSAccessRight-Write,
                    eRight-Bit5, --eFSAccessRight-UpdateMetadata,
                    eRight-Bit6 --eFSAccessRight-UpdateACL
                    }

-- urn:etsi.org:asn.1:accessor:fsa:1
eFS-ACC-FSA1        UUID::='3377F1EB69985D70BCA7D8E390DF084F'H
-- urn:etsi.org:asn.1:accessor:fsa:2
eFS-ACC-FSA2        UUID::='A2BEB42E8863555EB0DA1957001A06C2'H
eFS-ACC-ROOT        UUID::='DD61116FF0DD57F48A4F52EE70276F24'H
eAS-ID-ASS-GateID_1    UUID::='AAAAAAAABBBBCCCCDDDDEEEEEEEEEEE'H
eAS-ID-ASS-GateID_2    UUID::='AAAAAAAABBBBCCCCDDDDEEEEEEEEEEA'H
eAS-ID-ASS-GateID_3    UUID::='AAAAAAAABBBBCCCCDDDDEEEEEEEEEEB'H
eAS-Challenge       UUID::='BA64E9EE888952F4891DA79401758FF4'H


aSessionID_1 SessionID ::= 01
aSessionID_2 SessionID ::= 02
aSizeFile1 FileSize ::= 5
aSizeFile2 FileSize ::= 5
aSizeFile3 FileSize ::= 5
aSizeFile4 FileSize ::= 5

aSizeFile5 FileSize ::= 256
aSizeFile6 FileSize ::= 5
aSizeFile7 FileSize ::= 5
aSizeFile8 FileSize ::= 5
aSizeFileLF FileSize ::= 5

eRequestTypeDEF BIT STRING ::= '000'B
eOID OBJECT IDENTIFIER ::= { 0 0 }

--eAASAccessRight-RequiresSecurePipe AccessorRights ::= {eRight-Bit1 }
--eAASAccessRight-Create AccessorRights ::= { eRight-Bit2 }
--eAASAccessRight-Delete AccessorRights ::= { eRight-Bit3 }
--eAASAccessRight-Update AccessorRights ::= { eRight-Bit4 }
--eAASAccessRight-UpdateACL AccessorRights ::= { eRight-Bit5 }
--eAASAccessRight-UpdateGroup AccessorRights ::= { eRight-Bit6 }
--eAASAccessRight-UpdateCredentialPolicy AccessorRights ::= { eRight-Bit7 }
--eAASAccessRight-UpdateCredentialStatus AccessorRights ::= { eRight-Bit8 }

-- The root accessor has all accessor rights

eFS-ACL-ROOT           AccessorRights ::= {
--eAASAccessRight-RequiresSecurePipe--   eRight-Bit1,
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--      eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--      eRight-Bit5,
--eAASAccessRight-UpdateGroup--    eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }

eFS-ACL-FSA1           AccessorRights ::= {
--eAASAccessRight-RequiresSecurePipe--   eRight-Bit1,
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--      eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--      eRight-Bit5,
--eAASAccessRight-UpdateGroup--    eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }
```

```
eFS-ACL-FSA2           AccessorRights ::= {
--eAASAccessRight-RequiresSecurePipe--   eRight-Bit1,
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--      eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--      eRight-Bit5,
--eAASAccessRight-UpdateGroup--    eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }


-- ASN1STOP
```

The Authentication token and certificates are given as examples. Tools are available in the ETSI forge repository in [SCP x509v3] to generate the needed certificates for creating the certification path.

```
-- ASN1START
eAS-ATK-01 AuthenticationToken::={
      tbsToken {
        version v1,
        subjectPublicKeyInfo {
          algorithm {
            algorithm { 0 0 }
          },
          subjectPublicKey '0'B
        },
        aATK-Content {
          aChallenge '00000000000000000000000000000000'H,
          aKey-Size e128,
          aStreamCipherIdentifier aAES-CGM-StreamCipherIdentifier
        }
      },
      signatureAlgorithm {
        algorithm { 0 0 }
      },
      signature {
        r 0,
        s 0
      }
    }
eAS-CERT-01 Certificate ::= {
   tbsCertificate {
     version v3,
     serialNumber 1,
     signature {
      algorithm { 0 0 },
      parameters OCTET STRING : '00'H
     },
     issuer rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     validity {
      notBefore utcTime : "000101000000Z",
      notAfter utcTime : "000101000000Z"
     },
     subject rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     subjectPublicKeyInfo {
      algorithm {
        algorithm  id-ecPublicKey
      },
```

```
      subjectPublicKey '0'B
     },
     issuerUniqueID '0'B,
     subjectUniqueID '0'B,
     extensions {
      {
        extnID { 0 0 },
        critical FALSE,
        extnValue '00'H
      }
     }
   },
   signatureAlgorithm {
     algorithm { 0 0 },
     parameters OCTET STRING : '00'H
   },
   signature '0'B
  }

-- ASN1STOP
```

6.6.2 Procedures

6.6.2.1 PFSS_001 - Open a pipe session with the identity gate

Procedure identification	PFSS_001
Procedure objectives	The other host shall be able to open a pipe session to the identity gate of the SSP host. From the GATE_LIST registry, the UUID of the root accessor shall be listed. If the procedure is successful then a pipe session is open between the identity application in the other host and the identity service in the SSP host.
Configuration reference	CFSS_003
Initial conditions	
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. The root accessor is available in SSP prepared for procedures purpose. The Tester acting as an accessor shall be able to be authenticated by using an authentication token authenticated by a certification path.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The service identifier 'DD61116F-F0DD-57F4-8A4F-52EE70276F24' shall be present. The procedure is successful if the previous requirement is satisfied.
5	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed. This step is required to clean up the context of the tests but it is not essential for the procedure objective.

6.6.2.2 PFSS_002 - Open a pipe session with the Accessor Authentication service

Procedure identification	PFSS_002
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CFSS_004
Initial conditions	
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. This UUID is also the identity of the Root accessor. This root accessor is dedicated for the tester and assigned to the test providers using the ETSI SSP tests. The procedure PFSS_001 shall be successfully executed.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{ROOT}$: The UUID gate identifier of the root Accessor Authentication service gate (DD61116F-F0DD-57F4-8A4F-52EE70276F24).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{ROOT}$: The UUID gate identifier of the root Accessor Authentication application gate (DD61116F-F0DD-57F4-8A4F-52EE70276F24). $GATE_{ROOT}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the procedure is successful.

6.6.2.3 PFSS_003 - Authentication of the root accessor

Procedure identification	PFSS_003
Procedure objectives	The root accessor shall be able to be authenticated with the Accessor Authentication service by using: The aAAS-OP-GET-CHALLENGE-Service-Command command. The aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command. The authentication mean is based on the authentication tokens.
Configuration reference	CFSS_004

Initial conditions
The procedure PFSS_002 shall be successfully executed.

Procedure sequence	
Step	**Description**

1	AAA gate sends an AAS-CONTROL-SERVICE-GATE-Commands command to AAS gate with:
	`-- ASN1START`
	`aPFSS-003-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CHALLENGE-Service-Command : {}`
	`-- ASN1STOP`

2	AAS gate sends AAS-CONTROL-SERVICE-GATE-Responses response to AAA gate with:
	`-- ASN1START`
	`aPFSS-003-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CHALLENGE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aChallenge eAS-Challenge,`
	`  aCertificates {eAS-CERT-01}}}`
	`-- ASN1STOP`
	aCertificate is a set of certificates. aChallenge is a random number (128 bit) generated by the AAS. The value expressed in the procedure is given as example.

3	AAA gate sends AAS-CONTROL-SERVICE-GATE-Commands command to AAS gate with:
	`-- ASN1START`
	`aPFSS-003-command-02  AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {`
	`  aToken eAS-ATK-01, aTokenCertificationPath {eAS-CERT-01}}}`
	`-- ASN1STOP`
	The authentication token shall contain the challenge as recovered at the step 2.
	The authentication token shall be verified by using the certification path.

3	AAS gate sends AAS-CONTROL-SERVICE-GATE-Responses response to AAA gate with:
	`-- ASN1START`
	`aPFSS-003-response-02 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {  aToken eAS-ATK-01, aTokenCertificationPath  {eAS-CERT-01} }`
	`}`
	`-- ASN1STOP`
	The authentication token shall contain the challenge as recovered at the step 2.
	The authentication token shall be verified by using the certification path.
	The procedure is successful if the same challenge is in all authentication tokens and all of them have been verified by their certification path.

6.6.2.4 PFSS_004 - Access to the Authentication Service from the root accessor

Procedure identification	PFSS_004
Procedure objectives	The authenticated root accessor shall be able to access the Accessor Authentication service by using: The aAAS-OP-ACCESS-SERVICE-Service-Command command. If the procedure is successful then the accessor can open a secure pipe session to the Accessor Authentication service.
Configuration reference	CFSS_004

Initial conditions

The procedure PFSS_003 shall be successfully executed.

Procedure sequence

Step	Description

1	AAA gate sends an AAS-OP-ACCESS-SERVICE-Service-Command command to AAS gate with:
	`-- ASN1START`
	`aPFSS-004-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {`
	`  aServiceIdentifier 'DD61116FF0DD57F48A4F52EE70276F24 'H,`
	`  aUseSecurePipe TRUE`
	`}`
	`-- ASN1STOP`

2	AAS gate sends an AAS-OP-ACCESS-SERVICE-Service-Response response to AAA gate with:
	`-- ASN1START`
	`aPFSS-004-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aGateIdentifier eAS-ID-ASS-GateID_1 /* <STORE(eAS-ID-ASS-GateID_1)> */`
	`  }`
	`}`
	`-- ASN1STOP`
	The AAS returns the gate identifier on which the authenticated root accessor can access the accessor authentication service by using a secure pipe.
	The procedure is successful if the AAS returns eAAS-OK.

6.6.2.5 PFSS_005 - Open a pipe session with the Accessor Authentication service

Procedure identification	PFSS_005
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CFSS_004

Initial conditions
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. This UUID is also the identity of the Root accessor. This root accessor is dedicated for the tester and assigned to the test providers using the ETSI SSP tests. The accessor has obtained the gate identifier on the accessor authentication service for the root accessor by using a secure pipe session. The procedure PFSS_004 shall be successfully executed.

Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{CD}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{ROOTBIS}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_004 (eAS-ID-ASS-GateID_1).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{DC}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{ROOTBIS}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_004 (eAS-ID-ASS-GateID_1). $GATE_{ROOTBIS}$ shall be present in one of the binding parameters (see VNP [10]). If present then the procedure is successful. A secure pipe session is opened between the AAA acting for the root accessor and AAS as the authentication service.

6.6.2.6 PFSS_006 - Creation of FS accessors

6.6.2.6.1 PFSS_061 - Creation of an accessor FS Accessor 1

Procedure identification	PFSS_061
Procedure objectives	The Accessor Authentication application shall be able to create an FSA1 accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. File System Accessor 1 (FSA1): Accessor identity: eFS-ACC-FSA1 The FSA1 accessor authentication mean shall be based on the pincode.
Configuration reference	CFSS_004

Initial conditions

The procedure PFSS_005 shall be successfully executed.

Procedure sequence

Step	Description

1	AAA gate sends AAS-CONTROL-SERVICE-GATE-Commands to AAS gate with:
	`-- ASN1START`
	`aPFSS-0061-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorConditions {`
	`  aAccessConditionsPIN ePinNumeric`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-ROOT,`
	`  aAccessorRights eFS-ACL-ROOT`
	`  },`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-FSA1`
	`  }`
	`  }`
	`  },`
	`  aCredential {  aPinNumericCredential "1234"  },`
	`  aCredentialsPolicy {`
	`  aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,  aMinSize 4,  aMaxSize 255,  aMaxAttempts 3`
	`}`
	`  },`
	`  aCredentialsStatus {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`
	
	The root accessor has all rights on the procedure accessor.
	The procedure accessor shall be authenticated by using the pin code.

2	AAS gate sends aAAS-ADMIN-CREATE-ACCESSOR-Service-Response to AAA gate with:
	`-- ASN1START`
	`aPFSS-0061-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK`
	`}`
	`-- ASN1STOP`
	
	The procedure is successful if the aAAS-Service-Response is eAAS-OK.

6.6.2.6.2 PFSS_0062 - Open a pipe session with the Accessor Authentication service for the FSA1 accessor

Procedure identification	PFSS_0062
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CFSS_004
Initial conditions	
The procedure PFSS_0061 shall be successfully executed.	
Procedure sequence	
Step	**Description** **Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{TEST}$: The UUID gate identifier of the FSA1 accessor AA service gate (3377f1eb-6998-5d70-bca7-d8e390df084f).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{TEST}$: The UUID gate identifier of the FSA1 accessor AA application gate (3377f1eb-6998-5d70-bca7-d8e390df084f). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the procedure is successful.

6.6.2.6.3 PFSS_0063 - Authentication of the accessor

Procedure identification	PFSS_0063
Procedure objectives	The Accessor Authentication application shall be able to authenticate FSA1 accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.
Configuration reference	CFSS_004
Initial conditions	
The procedure PFSS_0062 shall be successfully executed.	
Procedure sequence	
Step	**Description** **Requirements**

1	AAA gate sends AAS-CONTROL-SERVICE-GATE-Commands to AAS gate with:
	`-- ASN1START`
	`aPFSS-0063-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1234"`
	`}`
	`-- ASN1STOP`

2	AAS gate sends AAS-CONTROL-SERVICE-GATE-Responses to AAA gate with:
	`-- ASN1START`
	`aPFSS-0063-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	
	The procedure is successful if the aAAS-Service-Response is eAAS-OK.

6.6.2.6.4 PFSS_0064 - Creation of an accessor FS Accessor 2

Procedure identification	PFSS_0061
Procedure objectives	The Accessor Authentication application shall be able to create an FSA2 accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. File System Accessor 2 (FSA2): • Accessor identity: eFS-ACC-FSA2 The FSA2 accessor authentication mean shall be based on the pincode.
Configuration reference	CFSS_005

Initial conditions

The procedure PFSS_005 shall be successfully executed. PFSS_005 and all of the referenced procedures shall be executed on Other Host_B.

Procedure sequence

Step	Description

1	AAA gate sends AAS-CONTROL-SERVICE-GATE-Commands to AAS gate with:
	`-- ASN1START`
	`aPFSS-0064-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eFS-ACC-FSA2,`
	`  aAccessorConditions {`
	`  aAccessConditionsPIN ePinNumeric`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-ROOT,`
	`  aAccessorRights eFS-ACL-ROOT`
	`  },`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA2,`
	`  aAccessorRights eFS-ACL-FSA2`
	`  }`
	`  }`
	`  },`
	`  aCredential {  aPinNumericCredential "1234"  },`
	`  aCredentialsPolicy {`
	`  aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,  aMinSize 4,  aMaxSize 255,  aMaxAttempts 3`
	`}`
	`  },`
	`  aCredentialsStatus {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`
	
	The root accessor has all rights on the procedure accessor.
	The procedure accessor shall be authenticated by using the pin code.

2	AAS gate sends aAAS-ADMIN-CREATE-ACCESSOR-Service-Response to AAA gate with:
	`-- ASN1START`
	`aPFSS-0064-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK`
	`}`
	`-- ASN1STOP`
	
	The procedure is successful if the aAAS-Service-Response is eAAS-OK.

6.6.2.6.5 PFSS_0065 - Open a pipe session with the Accessor Authentication service for the FSA2 accessor

Procedure identification	PFSS_0065
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CFSS_005
Initial conditions	
The procedure PFSS_0064 shall be successfully executed.	
Procedure sequence	
Step	**Description** **Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{TEST}$: The UUID gate identifier of the FSA2 accessor AA service gate (a2beb42e-8863-555e-b0da-1957001a06c2).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{TEST}$: The UUID gate identifier of the FSA2 accessor AA application gate (a2beb42e-8863-555e-b0da-1957001a06c2). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP [10]). If present then the procedure is successful.

6.6.2.6.6 PFSS_0066 - Authentication of the accessor

Procedure identification	PFSS_0066
Procedure objectives	The Accessor Authentication application shall be able to authenticate FSA2 accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.
Configuration reference	CFSS_005
Initial conditions	
The procedure PFSS_0065 shall be successfully executed.	
Procedure sequence	
Step	**Description** **Requirements**

1	AAA gate sends AAS-CONTROL-SERVICE-GATE-Commands to AAS gate with:
	`-- ASN1START`
	`aPFSS-0066-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1234"`
	`}`
	`-- ASN1STOP`

2	AAS gate sends AAS-CONTROL-SERVICE-GATE-Responses to AAA gate with:
	`-- ASN1START`
	`aPFSS-0066-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	
	The procedure is successful if the aAAS-Service-Response is eAAS-OK.

6.6.2.7 PFSS_007 - Open a secure pipe session to FS control service

6.6.2.7.1 PFSS_0071 - Access to FS control service for FSA1 with secure pipe

Procedure identification	PFSS_0071
Procedure objectives	The Accessor Authentication application on the behalf of FSA1 accessor shall be able to access the FS control service from the Accessor Authentication service using an aAAS-OP-ACCESS-SERVICE-Service-Command. The FS control service identifier is 366BD642-D7DE-584A-BD3B-A3DCE29FC075.
Configuration reference	CFSS_004
Initial conditions	
The procedure PFSS_0063 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	AAA gate sends an AAS-OP-ACCESS-SERVICE-Service-Command to AAS gate with:
	`-- ASN1START`
	`aPFSS-0071-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {`
	`  aServiceIdentifier eFS-ID-FSCS,`
	`  aUseSecurePipe TRUE`
	`}`
	`-- ASN1STOP`

2	AAS gate sends an AAS-OP-ACCESS-SERVICE-Service-Response to AAA gate with:
	`-- ASN1START`
	`aPFSS-0071-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aGateIdentifier eAS-ID-ASS-GateID_2 /* <STORE(eAS-ID-ASS-GateID_2)> */`
	`  }`
	`}`
	`-- ASN1STOP`
	
	The AAS returns the gate identifier on which the authenticated FSA1 accessor can access the File System service by using a secure pipe.
	The procedure is successful if the AAS returns eAAS-OK.

6.6.2.7.2 PFSS_0072 - Open a secure pipe session with the FS control service for the FSA1 accessor

Procedure identification	PFSS_0072
Procedure objectives	The other host shall be able to open a pipe session to the FS control service gate of the SSP host on the behalf of the FSA1 accessor. If the procedure is successful then a pipe session is open between the FS control application in the other host and the FS control service in the SSP host.
Configuration reference	CFSS_004
Initial conditions	
The procedure PFSS_0071 shall be successfully executed.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the File System service gate. • $GATE_{TEST}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_0071 (eAS-ID-ASS-GateID_2).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the file system application gate. • $GATE_{TEST}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_0071 (eAS-ID-ASS-GateID_2). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP [10]. If present then the procedure is successful. A secure pipe session is opened between the FSA1 accessor and the File System service gate.

6.6.2.7.3 PFSS_0073 - Access to FS control service for FSA2 with secure pipe

Procedure identification	PFSS_0073
Procedure objectives	The Accessor Authentication application on the behalf of FSA2 accessor shall be able to access the FS control service from the Accessor Authentication service using an aAAS-OP-ACCESS-SERVICE-Service-Command. The FS control service identifier is 366BD642-D7DE-584A-BD3B-A3DCE29FC075.
Configuration reference	CFSS_005

Initial conditions
The procedure PFSS_0066 shall be successfully executed.

Procedure sequence	
Step	**Description**

1	AAA gate sends an AAS-OP-ACCESS-SERVICE-Service-Command to AAS gate with:
	`-- ASN1START`
	`aPFSS-0073-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {`
	`  aServiceIdentifier eFS-ID-FSCS,`
	`  aUseSecurePipe TRUE`
	`}`
	`-- ASN1STOP`

2	AAS gate sends an AAS-OP-ACCESS-SERVICE-Service-Response to AAA gate with:
	`-- ASN1START`
	`aPFSS-0073-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aGateIdentifier eAS-ID-ASS-GateID_3 /* <STORE(eAS-ID-ASS-GateID_3)> */`
	`  }`
	`}`
	`-- ASN1STOP`
	
	The AAS returns the gate identifier on which the authenticated FSA2 accessor can access the File System service by using a secure pipe.
	The procedure is successful if the AAS returns eAAS-OK.

6.6.2.7.4 PFSS_0074 - Open a secure pipe session with the FS control service for the FSA2 accessor

Procedure identification	PFSS_0074
Procedure objectives	The other host shall be able to open a pipe session to the FS control service gate of the SSP host on the behalf of the FSA2 accessor. If the procedure is successful then a pipe session is open between the FS control application in the other host and the FS control service in the SSP host.
Configuration reference	CFSS_005

Initial conditions
The procedure PFSS_0073 shall be successfully executed.

Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the File System service gate. • $GATE_{TEST}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_0073 (eAS-ID-ASS-GateID_3).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the file system application gate. • $GATE_{TEST}$: The dynamically assigned UUID gate identifier returned by AAS in PFSS_0073 (eAS-ID-ASS-GateID_3). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP [10]). If present then the procedure is successful. A secure pipe session is opened between the FSA2 accessor and the File System service gate.

6.6.2.8 PFSS_008 - Create directories

6.6.2.8.1 PFSS_0081 - Create directory 1

Procedure identification	PFSS_0081
Procedure objectives	The File System Application shall be able to create a directory in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command. The objective is the creation of the directory 1
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0072 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateDirectory1 to FSCS gate with:
	`-- ASN1START`
	`aCreateDirectory1 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-directory1,`
	`  aShortName eFS-ID-directory1,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1, --FSA1`
	`  aAccessorRights eFS-ACL-directory1`
	`  }`
	`  }`
	`  },`
	`aNodeDirectoryIdentity aShortName eFS-ID-SSPFS`
	`}`
	`-- ASN1STOP`

2	FSCS gate sends aCreateDirectory2Response to FSCA gate with:
	`-- ASN1START`
	`aCreateDirectory1Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.8.2 PFSS_0082 - Create directory 2

Procedure identification	PFSS_0082
Procedure objectives	The File System Application shall be able to create a directory in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command. The objective is the creation of the directory 2
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0072 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateDirectory2 to FSCS gate with:
	`-- ASN1START`
	`aCreateDirectory2 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-directory2,`
	`  aShortName eFS-ID-directory2,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1, --FSA1`
	`  aAccessorRights eFS-ACL-directory2`
	`  }`
	`  }`
	`  },`
	`aNodeDirectoryIdentity aShortName eFS-ID-SSPFS`
	`}`
	`-- ASN1STOP`

2	FSCS gate sends aCreateDirectory2Response to FSCA gate with:
	`-- ASN1START`
	`aCreateDirectory2Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.8.3 PFSS_0083 - Create directory 3

Procedure identification	PFSS_0083
Procedure objectives	The File System Application shall be able to create a directory in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command. The objective is the creation of the directory 3
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0072 shall be successfully executed. The procedure PFSS_0081 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateDirectory3 to FSCS gate with:
	`-- ASN1START`
	`aCreateDirectory3 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-directory3,`
	`  aShortName eFS-ID-directory3,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1, --FSA1`
	`  aAccessorRights eFS-ACL-directory3`
	`  }`
	`  }`
	`  },`
	`aNodeDirectoryIdentity aShortName eFS-ID-directory1`
	`}`
	`-- ASN1STOP`

2	FSCS gate sends aResponse to FSCA gate with:
	`-- ASN1START`
	`aResponse FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.8.4 PFSS_0084 - Create directory 4

Procedure identification	PFSS_0084
Procedure objectives	The File System Application shall be able to create a directory in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command. The objective is the creation of the directory 4
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0072 shall be successfully executed. The procedure PFSS_0083 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateDirectory4 to FSCS gate with:
	`-- ASN1START`
	`aCreateDirectory4 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-directory4,`
	`  aShortName eFS-ID-directory4,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1, --FSA1`
	`  aAccessorRights eFS-ACL-directory4`
	`  }`
	`  }`
	`  },`
	`aNodeDirectoryIdentity aShortName  eFS-ID-directory3`
	`}`
	`-- ASN1STOP`

2	FSCS gate sends aCreateDirectory4Response to FSCA gate with:
	`-- ASN1START`
	`aCreateDirectory4Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9 PFSS_009 - Create files

6.6.2.9.1 PFSS_0091 - Create file 1

Procedure identification	PFSS_0091
Procedure objectives	The File System Application shall be able to create a file 1 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0083 shall be successfully executed.

Procedure sequence

Step	**Description**

1	FSCA gate sends aCreateFile1 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile1 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file1,`
	`  aShortName eFS-ID-file1,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile1`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file1`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-SSPFS`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile1Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile1Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.2 PFSS_0092 - Create file 2

Procedure identification	PFSS_0092
Procedure objectives	The File System Application shall be able to create a file 2 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0081 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile2 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile2 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file2,`
	`  aShortName eFS-ID-file2,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile2`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file2`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory1`
	`  }`
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile2Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile2Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.3 PFSS_0093 - Create file 3

Procedure identification	PFSS_0093
Procedure objectives	The File System Application shall be able to create a file 3 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0084 shall be successfully executed.

Procedure sequence

Step	Description

1	FSCA gate sends aCreateFile3 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile3 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file3,`
	`  aShortName eFS-ID-file3,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile3`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file3`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory2`
	`  }`
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile6Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile6Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.4 PFSS_0094 - Create file 4

Procedure identification	PFSS_0094
Procedure objectives	The File System Application shall be able to create a file 4 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0084 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile4 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile4 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file4,`
	`  aShortName eFS-ID-file4,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile4`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file4`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory2`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile4Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile4Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.5 PFSS_0095 - Create file 5

Procedure identification	PFSS_0095
Procedure objectives	The File System Application shall be able to create a file 5 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0083 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile5 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile5 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file5,`
	`  aShortName eFS-ID-file5,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile5`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file5`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory3`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile1Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile5Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.6 PFSS_0096 - Create file 6

Procedure identification	PFSS_0096
Procedure objectives	The File System Application shall be able to create a file 6 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0084 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile6 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile6 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file6,`
	`  aShortName eFS-ID-file6,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile6`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file6`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory4`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile6Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile6Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.7 PFSS_0097 - Create link 1

Procedure identification	PFSS_0097
Procedure objectives	The File System Application shall be able to create a link 1 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0082 shall be successfully executed. The procedure PFSS_0091 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateLink1 to FSCS gate with:
	`-- ASN1START`
	`aCreateLink1 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-link1,`
	`  aShortName eFS-ID-link1,`
	`  aNode aLink : {`
	`  aLinkedFileIdentity aShortName eFS-ID-file1,`
	`  aLinkedFileSize 10`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2`
	`  }`
	`}`
	`-- ASN1STOP`

2	FSCS gate sends aCreateLink1Response to FSCA gate with:
	`-- ASN1START`
	`aCreateLink1Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.8 PFSS_0098 - Create file 7

Procedure identification	PFSS_0098
Procedure objectives	The File System Application shall be able to create a file 7 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0083 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile7 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile7 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file7,`
	`  aShortName eFS-ID-file7,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile7`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file7`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-SSPFS`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile7Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile7Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.2.9.9 PFSS_0099 - Create file 8

Procedure identification	PFSS_0099
Procedure objectives	The File System Application shall be able to create a file 8 in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0084 shall be successfully executed.	
Procedure sequence	
Step	**Description**

1	FSCA gate sends aCreateFile8 to FSCS gate with:
	`-- ASN1START`
	`aCreateFile8 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file8,`
	`  aShortName eFS-ID-file8,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile8`
	`  },`
	`  aMetaData {`
	`  {`
	`  aTypeDatum eOID,`
	`  aData OCTET STRING : '00'H`
	`  }`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file8`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory4`
	`  }`
	
	`-- ASN1STOP`

2	FSCS gate sends aCreateFile8Response to FSCA gate with:
	`-- ASN1START`
	`aCreateFile2Response FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`

6.6.3 Test descriptions

6.6.3.1 Create node

6.6.3.1.1 FSS_0011 - Create directory and file

Test identification	FSS_0011
Test objectives	The File System Application shall be able to create a directory and a file node in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command. The SSP file system service shall ignore the short name included in aNodeDescriptor and compute it.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0072 is successfully executed. directory 1 is not created file2 is not created	

Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0011-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0011-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-directory1,`
	`  aShortName eFS-ID-directory1,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-directory1`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aNodeReference : {`
	`  eFS-Name-SSPFS`
	`  }`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0011-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0011-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_017
		RQ0606_023
		RQ0606_036
		RQ0606_037
		RQ0606_040

3	FSCA gate sends aFSS-0011-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0011-command-02  FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-file2,`
	`  aShortName eFS-ID-file-fake,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile2`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file2`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aShortName eFS-ID-directory2`
	`  }`
	
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0011-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0011-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_036
		RQ0606_037
		RQ0606_040

5	FSCA gate sends aFSS-0011-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0011-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`  },`
	`  aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0011-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0011-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`    aNodeDescriptorList {`
	`      {`
	`        aNodeName eFS-Name-file2,`
	`        aShortName eFS-ID-file2,`
	`        aNode aFile : {`
	`          aFileSize aSizeFile2`
	`        },`
	`        aACL {`
	`          {`
	`            aAccessorIdentity eFS-ACC-FSA1,`
	`            aAccessorRights eFS-ACL-file2`
	`          }`
	`        }`
	`      }`
	`    }`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_007
		RQ0606_038

6.6.3.1.2 FSS_0012 - Create link

Test identification	FSS_0012
Test objectives	The File System Application shall be able to create a link node in the SSP file system using FS-ADMIN-CREATE-NODE-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0082 is successfully executed. The procedure PFSS_0091 is successfully executed. link1 is not created.	
Test sequence	

Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0012-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0012-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`    aNodeName eFS-Name-link1,`
	`    aShortName eFS-ID-link1,`
	`    aNode aLink : {`
	`      aLinkedFileIdentity aShortName eFS-ID-file1,`
	`      aLinkedFileSize 8`
	`    }`
	`  },`
	`  aNodeDirectoryIdentity aNodeReference : {`
	`    eFS-Name-SSPFS,`
	`    eFS-Name-directory2`
	`  }`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0012-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0012-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_017
		RQ0606_023
		RQ0606_036
		RQ0606_037
		RQ0606_040

3	FSCA gate sends aFSS-0012-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0012-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-link1`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0012-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0012-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_056

5	FSCA gate sends aFSS-0012-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0012-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 0`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0012-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0012-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '0102030405'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_005
		RQ0606_039

6.6.3.2 Read file

6.6.3.2.1 FSS_0021 - Read file through Control Pipe

Test identification	FSS_0021	
Test objectives	The File System Application shall be able to open a file from the SSP file system using FS-OP-FILE-OPEN-Service-Command, to read a file using FS-OP-FILE-READ-Service-Command and close the file using aFS-OP-FILE-CLOSE-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0091 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0021-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0021-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0021-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0021-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0021-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0021-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0021-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0021-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '0102030405'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_004
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006

5	FSCA gate sends aFSS-0021-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0021-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0021-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0021-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.2.2 FSS_0022 - Read file through Data Pipe

Test identification	FSS_0022	
Test objectives	The File System Application shall be able to read a file from the SSP file system through File System Data Pipe using FS-OP-FILE-READ-Service-Command. If there is a pipe session associated with the aSessionID, the SSP file system application closes this pipe session when FS-OP-FILE-CLOSE-Service-Command is successful	
Configuration reference	CFFS_002	
Initial conditions		
The procedure PFSS_0095 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0022-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0022-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file5,`
	`  aAccessMode  eReadAccessMode,`
	`  aGateID '863391838CF658C28142D53611D52F12'H`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0022-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0022-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
	Pipe session is opened between the FSDS gate and the FSDA gate by using the 86339183-8cf6-58c2-8142-d53611d52f12 gate identifier.
		RQ0606_017
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058
		RQ1003_015
		RQ1003_016

3	FSCA gate sends aFSS-0022-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0022-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

4	FSDS gate sends a stream with the content of file5 to FSDA gate	
5	Administration gate send an acknowledgement about receiving the content of file5 to administration gate in SCL host in the SSP host domain.	

6	FSCS gate sends aFSS-0022-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0022-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_068
		RQ0606_071

7	FSCA gate sends aFSS-0022-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0022-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0022-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0022-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061
		RQ0606_075

9	FSCA gate sends aFSS-0022-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0022-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

10	FSCS gate sends aFSS-0022-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0022-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-BAD-SESSSION-ID`
	`}`
	`-- ASN1STOP`
		RQ0606_062

6.6.3.2.3 FSS_0023 - Read file with long name from file tree hierarchy

Test identification	FSS_0023
Test objectives	The File System shall support a tree of nodes with a minimum height of 5. A node file'filelongfilename' is created under 'SSPFS:directory1:directory3:directory4'. UUID: 57c73c00-5fea-5db2-b93e-92dd5691d270 from urn:SSPFS:directory1:directory3:directory4:filelongfilename The node file is write then read
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0084 is successfully executed.

Test sequence

Step	Description	Requirements

1	FSCA gate sends aFSS-0023-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0023-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-CREATE-NODE-Service-Command : {`
	`  aNodeDescriptor {`
	`  aNodeName eFS-Name-filelongfilename,`
	`  aShortName eFS-ID-filelongfilename,`
	`  aNode aFile : {`
	`  aFileSize aSizeFileLF`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-filelongfilename`
	`  }`
	`  }`
	`  },`
	`  aNodeDirectoryIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-directory3,`
	`  eFS-Name-directory4`
	`  }`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0023-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0023-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-CREATE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`	

3	FSCA gate sends aFSS-0023-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0023-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName : eFS-ID-filelongfilename,`
	`  aAccessMode  eWriteAccessMode}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0023-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0023-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_011
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

5	FSCA gate sends aFSS-0023-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0023-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '6666666666'H`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0023-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0023-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`	

7	FSCA gate sends aFSS-0023-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0023-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1,`
	`  aOffset 2`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0023-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0023-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1,`
	`  aData '666666'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_008
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006

9	FSCA gate sends aFSS-0023-command-05 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0023-command-05 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

10	FSCS gate sends aFSS-0023-response-05 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0023-response-05 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.2.4 FSS_0024 - Read file through a Secured Control Pipe

Test identification	FSS_0024	
Test objectives	The File System Application shall be able to read a file from the SSP file system using FS-OP-FILE-OPEN-Service-Command, even if the file requires secure pipe.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0098 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0024-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0024-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file7,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0024-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0024-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_025
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0024-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0024-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0024-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0024-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '0102030405'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006

5	FSCA gate sends aFSS-0024-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0024-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0024-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0024-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.2.5 FSS_0025 - Error when reading file without ReadContent access right

Test identification	FSS_0025	
Test objectives	The File System Application shall not be able to open a file for reading if no ReadContent access right is granted to the accessor.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0093 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0025-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0025-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-file3`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0025-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0025-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-ACL-RULES-VIOLATIONS`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_079

6.6.3.2.6 FSS_0026 - Error when trying to read a file while a previous command is ongoing in the same file session

Test identification	FSS_0026	
Test objectives	If the SSP file system application sends a FS-OP-FILE-READ-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-NODE-BUSY.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0026-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0026-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-file6`
	`  },`
	`  aAccessMode {eWriteAccessMode, eReadAccessMode}`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0026-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0026-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_028
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0026-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0026-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '6666666666'H`
	`}`
	`-- ASN1STOP`	

4	Immediately after aFSS-0026-command-02 FSCA gate sends aFSS-0026-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0026-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1,`
	`  aOffset 0,`
	`  aNumberOfBytes 6`
	`}`
	`-- ASN1STOP`	

5	FSCS gate sends aFSS-026-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0026-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-BUSY`
	`}`
	`-- ASN1STOP`
		RQ0606_069

6.6.3.3 Write file

6.6.3.3.1 FSS_0031 - Write file

Test identification	FSS_0031	
Test objectives	The File System Application shall be able to write a file in the SSP file system using FS-OP-FILE-WRITE-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0031-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0031-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`   aNodeIdentity aNodeReference : {`
	`    eFS-Name-SSPFS,`
	`    eFS-Name-file6`
	`  },`
	`  aAccessMode {eWriteAccessMode, eReadAccessMode}`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0031-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0031-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_028
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0031-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0031-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '6666666666'H`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0031-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0031-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_072
		RQ0606_074

5	FSCA gate sends aFSS-0031-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0031-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID  aSessionID_1,`
	`  aOffset 0,`
	`  aNumberOfBytes 6`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0031-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0031-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID  aSessionID_1,`
	`  aData '6666666666'H`
	`  }`
	`}`
	`-- ASN1STOP`	

7	FSCA gate sends aFSS-0031-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0031-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0031-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0031-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.3.2 FSS_0032 - Write file by omitting aOffset

Test identification	FSS_0032	
Test objectives	The File System Application shall be able to write a file in the SSP file system using FS-OP-FILE-WRITE-Service-Command by omitting aOffset.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0032-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0032-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-file6`
	`  },`
	`  aAccessMode {eWriteAccessMode, eReadAccessMode}`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0032-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0032-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_028
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0032-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0032-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aDataInfo aData : '6666666666'H`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0032-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0032-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_072
		RQ0606_074

5	FSCA gate sends aFSS-0032-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0032-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0032-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0032-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1,`
	`  aData '6666666666'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_068

7	FSCA gate sends aFSS-0032-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0032-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0032-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0032-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.3.3 FSS_0033 - Error when writing file without Write access right

Test identification	FSS_0033	
Test objectives	The File System Application shall not be able to open a file for reading if no Write access right is granted to the accessor.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0094 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0033-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0033-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-file4`
	`  },`
	`  aAccessMode eWriteAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0033-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0033-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-ACL-RULES-VIOLATIONS`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_029
		RQ0606_054
		RQ0606_055
		RQ0606_079

6.6.3.3.4 FSS_0034 - Error when trying to write a file while a previous command is ongoing in the same file session

Test identification	FSS_0034
Test objectives	If the SSP file system application sends a FS-OP-FILE-WRITE-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-NODE-BUSY.
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0096 is successfully executed.

Test sequence

Step	Description	Requirements

1	FSCA gate sends aFSS-0034-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0034-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-file6`
	`  },`
	`  aAccessMode {eWriteAccessMode, eReadAccessMode}`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0034-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0034-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_029
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0034-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0034-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '6666666666'H`
	`}`
	`-- ASN1STOP`	

4	Immediately after aFSS-0034-command-02 FSCA gate sends aFSS-0034-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0034-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '7777777777'H`
	`}`
	`-- ASN1STOP`	

5	FSCS gate sends aFSS-034-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0034-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-BUSY`
	`}`
	`-- ASN1STOP`
		RQ0606_073

6.6.3.4 Delete node

6.6.3.4.1 FSS_0041 - Delete file

Test identification	FSS_0041
Test objectives	The File System Application shall be able to delete a file using FS-ADMIN-DELETE-NODE-Service-Command. After the deletion of a node, all SSP links pointing to that node shall also be deleted by the SSP file system service.
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0091 is successfully executed.
The procedure PFSS_0097 is successfully executed.

Test sequence

Step	Description	Requirements

1	FSCA gate sends aFSS-0041-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0041-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0041-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0041-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_023
		RQ0606_031
		RQ0606_032
		RQ0606_041
		RQ0606_042
		RQ0606_048

3	FSCA gate sends aFSS-0041-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0041-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0041-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0041-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-NOT-FOUND`
	`}`
	`-- ASN1STOP`
		RQ0606_044
		RQ0606_079

5	FSCA gate sends aFSS-0041-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0041-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-link1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0041-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0041-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-NOT-FOUND`
	`}`
	`-- ASN1STOP`
		RQ0606_046

6.6.3.4.2 FSS_0042 - Delete directory

Test identification	FSS_0042	
Test objectives	The File System Application shall be able to delete a directory and a child node in the directory using FS-ADMIN-DELETE-NODE-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0092 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0042-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0042-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName  eFS-ID-directory1`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0042-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0042-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_023
		RQ0606_031
		RQ0606_032
		RQ0606_041
		RQ0606_042
		RQ0606_048

3	FSCA gate sends aFSS-0042-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0042-command-02 FS-CONTROL-SERVICE-GATE-Commands::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0042-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0042-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-NOT-FOUND,`
	`  aParameter {`
	`  aSessionID aSessionID_1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_044
		RQ0606_079

6.6.3.4.3 FSS_0043 - Delete directory content without delete access right

Test identification	FSS_0043	
Test objectives	The File System Application shall be able to delete a directory and a child node in the directory using FS-ADMIN-DELETE-NODE-Service-Command regardless of the value of eFSAccessRight-Delete of each contained node.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0043-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0043-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName  eFS-ID-directory4`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0043-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0043-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_023
		RQ0606_031
		RQ0606_032
		RQ0606_041
		RQ0606_042
		RQ0606_048

3	FSCA gate sends aFSS-0043-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0043-command-02 FS-CONTROL-SERVICE-GATE-Commands::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-directory3,`
	`  eFS-Name-directory4,`
	`  eFS-Name-file6`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0043-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0043-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-NOT-FOUND,`
	`  aParameter {`
	`  aSessionID aSessionID_1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_044
		RQ0606_079

6.6.3.4.4 FSS_0044 - Delete link

Test identification	FSS_0044	
Test objectives	The File System Application shall be able to delete a link node using FS-ADMIN-DELETE-NODE-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0097 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0044-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0044-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName : eFS-ID-link1`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0044-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0044-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_023
		RQ0606_031
		RQ0606_041
		RQ0606_042
		RQ0606_048

3	FSCA gate sends aFSS-0044-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0044-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	` aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-link1`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0044-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0044-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-NOT-FOUND`
	` }`
	`-- ASN1STOP`
		RQ0606_047
		RQ0606_079

5	FSCA gate sends aFSS-0044-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0044-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`  },`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0044-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0044-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_045

7	FSCA gate sends aFSS-0044-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0044-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0044-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0044-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.4.5 FSS_0045 - Error when deleting file without delete access right

Test identification	FSS_0045	
Test objectives	The File System Application shall not be able to delete a node in the SSP file system if no Delete access right is granted to the accessor.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0045-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0045-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file6`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0045-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0045-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-ACL-RULES-VIOLATIONS`
	`}`
	`-- ASN1STOP`
		RQ0606_031

6.6.3.4.6 FSS_0046 - Error when deleting file while a file session is open with the same file

Test identification	FSS_0046	
Test objectives	The SSP file system shall reject the deletion of a node if a session is ongoing on the node.	
Configuration reference	CFSS_006	
Initial conditions		
The procedure PFSS_0074 is successfully executed. The procedure PFSS_0091 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA_1 gate sends aFSS-0046-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0046-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0046-response-01 to FSCA_1 gate with:
	`-- ASN1START`
	`aFSS-0046-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA_2 gate sends aFSS-0046-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0045-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-DELETE-NODE-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0046-response-02 to FSCA_2 gate with:
	`-- ASN1START`
	`aFSS-0045-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-DELETE-NODE-Service-Response : {`
	`  aFS-Service-Response eFS-NODE-BUSY`
	`}`
	`-- ASN1STOP`
		RQ0606_043

6.6.3.5 Get Info

6.6.3.5.1 FSS_0051 - Get Info file

Test identification	FSS_0051
Test objectives	The File System Application shall be able to retrieve the information about a file in the SSP file system.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0092 shall be successfully executed.	
Test sequence	

Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0051-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0051-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`},`
	`aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0051-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0051-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`aFS-Service-Response eFS-OK,`
	`aParameter {`
	`aNodeDescriptorList {`
	`{`
	`aNodeName eFS-Name-file2,`
	`aShortName eFS-ID-file2,`
	`aNode aFile : {`
	`aFileSize aSizeFile2`
	`},`
	`aACL {`
	`{`
	`aAccessorIdentity eFS-ACC-FSA1,`
	`aAccessorRights eFS-ACL-file2`
	`}`
	`}`
	`}`
	`}`
	`}`
	`}`
	`-- ASN1STOP`
		RQ0606_007
		RQ0606_063
		RQ0606_064
		RQ0606_067

6.6.3.5.2 FSS_0052 - Get Info parent of a file

Test identification	FSS_0052	
Test objectives	The File System Application shall be able to retrieve the information about the parent of a file in the SSP file system.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0092 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0052-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0052-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`  },`
	`  aRequestType aParent`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0052-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0052-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-directory1,`
	`  aShortName eFS-ID-directory1,`
	`  aNode aDirectory : {`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-directory1`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`}`
	`-- ASN1STOP`
		RQ0606_007
		RQ0606_063
		RQ0606_064
		RQ0606_067

6.6.3.5.3 FSS_0053 - Get Info siblings

Test identification	FSS_0053	
Test objectives	The File System Application shall be able to retrieve the information about the siblings of a file in the SSP file system.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0093 shall be successfully executed. The procedure PFSS_0094 is successfully executed. The procedure PFSS_0097 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0053-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0053-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`  aNodeIdentity aShortName : eFS-ID-file3,`
	`  aRequestType aContain`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0053-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0053-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-file3,`
	`  aShortName eFS-ID-file3,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile3`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file3`
	`  }`
	`  }`
	`  },`
	`  {`
	`  aNodeName eFS-Name-file4,`
	`  aShortName eFS-ID-file4,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile4`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file4`
	`  }`
	`  }`
	`  },`
	`  {`
	`  aNodeName eFS-Name-link1,`
	`  aShortName eFS-ID-link1,`
	`  aNode aLink : {`
	`  aLinkedFileIdentity aShortName eFS-ID-file1,`
	`  aLinkedFileSize aSizeFile1`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-link1`
	`  }`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_007
		RQ0606_063
		RQ0606_064
		RQ0606_066
		RQ0606_067

6.6.3.5.4 FSS_0054 - Get Info link

Test identification	FSS_0054	
Test objectives	The File System Application shall be able to retrieve the information about a link in the SSP file system.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0097 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0054-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0054-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-link1`
	`  },`
	`  aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0054-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0054-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-link1,`
	`  aShortName eFS-ID-link1,`
	`  aNode aLink : {`
	`  aLinkedFileIdentity aShortName eFS-ID-file1,`
	`  aLinkedFileSize aSizeFile1`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file1`
	`  }`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_007
		RQ0606_063
		RQ0606_064
		RQ0606_065
		RQ0606_067

6.6.3.5.5 FSS_0055 - Error when getting info about file 6 without GetInfo access right

Test identification	FSS_0055
Test objectives	The File System Application shall not be able to retrieve the information about a file in the SSP file system if no GetInfo right is granted to the accessor.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0096 shall be successfully executed.	
Test sequence	

Step	Description	Requirements

1	FSCA gate sends aFSS-0055-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0055-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`aNodeIdentity aNodeReference : {`
	`eFS-Name-SSPFS,`
	`eFS-Name-directory1,`
	`eFS-Name-directory3,`
	`eFS-Name-directory4,`
	`eFS-Name-file6`
	`},`
	`aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0055-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0055-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`aFS-Service-Response eFS-ACL-RULES-VIOLATIONS`
	`}`
	`-- ASN1STOP`
		RQ0606_027

6.6.3.6 Update node

6.6.3.6.1 FSS_0061 - Update access control of a file

Test identification	FSS_0061
Test objectives	The File System Application shall be able to update the access control of a file node in the SSP file system using the FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0092 shall be successfully executed.	
Test sequence	

Step	Description	Requirements

1	FSCA gate sends aFSS-0061-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0061-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command : {`
	`aNodeIdentity aNodeReference : {`
	`eFS-Name-SSPFS,`
	`eFS-Name-directory1,`
	`eFS-Name-file2`
	`},`
	`aACL {`
	`{`
	`aAccessorIdentity eFS-ACC-FSA1,`
	`aAccessorRights eFS-ACL-file_upd`
	`}`
	`}`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0061-response-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0061-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_024
		RQ0606_030
		RQ0606_049
		RQ0606_052
		RQ0606_053

3	FSCA gate sends aFSS-0061-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0061-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-file2`
	`  },`
	`  aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0061-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0061-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-file2,`
	`  aShortName eFS-ID-file2,`
	`  aNode aFile : {`
	`  aFileSize aSizeFile2`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file_upd`
	`  }`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`	

6.6.3.6.2 FSS_0062 - Update access control of a link

Test identification	FSS_0062	
Test objectives	The File System Application shall be able to update the access control of a link node in the SSP file system using the FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0097 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0062-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0062-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command : {`
	`  aNodeIdentity aShortName : eFS-ID-link1,`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file_upd`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0062-response-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0062-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_024
		RQ0606_030
		RQ0606_033
		RQ0606_049
		RQ0606_050
		RQ0606_051
		RQ0606_052
		RQ0606_053

3	FSCA gate sends aFSS-0062-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0062-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory2,`
	`  eFS-Name-link1`
	`  },`
	`  aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0062-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0062-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-link1,`
	`  aShortName eFS-ID-link1,`
	`  aNode aLink : {`
	`  aLinkedFileIdentity aShortName eFS-ID-file1,`
	`  aLinkedFileSize 5`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-link1`
	`  }`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`	

6.6.3.6.3 FSS_0063 - Update metadata

Test identification	FSS_0063	
Test objectives	The File System Application shall be able to update the metadata of a file node in the SSP file system using the FS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0099 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0063-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0063-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-directory3,`
	`  eFS-Name-directory4,`
	`  eFS-Name-file8`
	`  },`
	`  aMetaData {`
	`  {`
	`  aTypeDatum eOID,`
	`  aData OCTET STRING : '11'H`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0063-response-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0063-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_024
		RQ0606_029
		RQ0606_049
		RQ0606_052
		RQ0606_053

3	FSCA gate sends aFSS-0063-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0063-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-NODE-GET-INFO-Service-Command : {`
	`  aNodeIdentity aNodeReference : {`
	`  eFS-Name-SSPFS,`
	`  eFS-Name-directory1,`
	`  eFS-Name-directory3,`
	`  eFS-Name-directory4,`
	`  eFS-Name-file8`
	`  },`
	`  aRequestType eRequestTypeDEF`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0063-response-02 FS-Response to FSCA gate with:
	`-- ASN1START`
	`aFSS-0063-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-NODE-GET-INFO-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aNodeDescriptorList {`
	`  {`
	`  aNodeName eFS-Name-file8,`
	`  aShortName eFS-ID-file8,`
	`  aNode aFile : {`
	`  aFileSize 5`
	`  },`
	`  aMetaData {`
	`  {`
	`  aTypeDatum  eOID,`
	`  aData OCTET STRING : '11'H`
	`  }`
	`  },`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file8`
	`  }`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`
	`-- ASN1STOP`	

6.6.3.6.4 FSS_0064 - Error when updating access control file without UpdateACL access right

Test identification	FSS_0064	
Test objectives	The File System Application shall not be able to update the access control of a node in the SSP file system if no UpdateACL access right is granted to the accessor..	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0096 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0064-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0064-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file6,`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eFS-ACC-FSA1,`
	`  aAccessorRights eFS-ACL-file6`
	`  }`
	`  }`
	`}`
	
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0064-response-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0064-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-ADMIN-UPDATE-NODE-ATTRIBUTES-Service-Response : {`
	
	`  aFS-Service-Response eFS-ACL-RULES-VIOLATIONS`
	`}`
	`-- ASN1STOP`
		RQ0606_049
		RQ0606_052
		RQ0606_079

6.6.3.7 Get position

6.6.3.7.1 FSS_0071 - Get Position

Test identification	FSS_0071
Test objectives	The File System Application shall be able to retrieve the current offset position in an SSP file using FS-OP-FILE-GET-POSITION-Service-Command.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0091 is successfully executed.	
Test sequence	

Step	Description	Requirements

1	FSCA gate sends aFSS-0071-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0071-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0071-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0071-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0071-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0071-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 2`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0071-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0071-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '0102'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006

5	FSCA gate sends aFSS-0071-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0071-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-GET-POSITION-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0071-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0071-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-GET-POSITION-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aCurrentOffset 3`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_076
		RQ0606_078

7	FSCA gate sends aFSS-0071-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0071-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0071-response-04 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0071-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.7.2 FSS_0072 - Error when trying to get the position while a previous command is ongoing in the same file session

Test identification	FSS_0072	
Test objectives	If the SSP file system application sends a FS-OP-FILE-GET-POSITION-Service-Command command while a previous command is ongoing in the same file session, the SSP file system shall reject the command with the error eFS-E-NODE-BUSY.	
Configuration reference	CFFS_001	
Initial conditions		
The procedure PFSS_0091 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0072-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0072-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0072-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0072-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0072-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0072-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 2`
	`}`
	`-- ASN1STOP`	

4	Immediately after aFSS-0072-command-02 FSCA gate sends aFSS-0072-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0072-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-GET-POSITION-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

5	FSCS gate sends aFSS-0072-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0071-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-GET-POSITION-Service-Response : {`
	`  aFS-Service-Response eFS-E-NODE-BUSY`
	`}`
	`-- ASN1STOP`
		RQ0606_077

6.6.3.8 Get capabilities

6.6.3.8.1 FSS_0081 - Get Capabilities

Test identification	FSS_0081
Test objectives	The File System Application shall be able to retrieve the capabilities of the SSP file system using FS-ADMIN-GET-CAPABILITIES-Service-Command.
Configuration reference	CFFS_001

Initial conditions

The procedure PFSS_0072 is successfully executed.

```
-- ASN1START
aFSSVERSION INTEGER ::= 15 /*<STORE(FSSVERSION)> it contains the value of the major and
minor release version supported by the file system control service gate */
aMAXFILESESSIONS INTEGER::= 2 /*<STORE(MAXFILESESSIONS)> it contains the value of the
maximum number of simultaneous file sessions supported*/
aMAXFILESESSIONS_PER_FILE INTEGER::= 15 /*<STORE(MAXFILESESSIONS_PER_FILE)>  it contains the
value of the maximum number of simultaneous file sessions supported on the same file*/
aCAPACITY INTEGER::= 0 /* STORE(CAPACITY)>
it contains the value of the total capacity of the SSP file system in bytes*/
aFREE_CAPACITY INTEGER::= 0 /*<STORE(FREE_CAPACITY)>
it contains the value of the remaining free capacity in the SSP file system in bytes*/
aMAXMETADATA_PER_NODE INTEGER ::=0 /*<STORE(MAXMETADATA_PER_NODE)
it contains the value of the maximum metadata size allowed per node in bytes*/
--ASN1STOP
```

Test sequence

Step	Description	Requirements

1	FSCA gate sends aFSS-0081-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0081-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-ADMIN-GET-CAPABILITIES-Service-Command : {`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0081-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0081-response-01 FS-CONTROL-SERVICE-GATE-Responses ::=`
	`aFS-ADMIN-GET-CAPABILITIES-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aVersion '0000'H /*<COMPARE(FSSVERSION,GT,EQ)>*/,`
	`  aSimultaneousFileSessions 1/* <COMPARE (MAXFILESESSIONS,GT,EQ)> */,`
	`  aSimultaneousFileSessionsPerFile 1/*<COMPARE (MAXFILESESSIONS_PER_FILE,GT,EQ)> */,`
	`  aTotalCapacity 0/*<COMPARE(CAPACITY,GT,EQ)>*/,`
	`  aFreeCapacity 0/*<COMPARE(FREE_CAPACITY,GT,EQ)>*/,`
	`  aMaxMetaDataSizePerNode 0 /*<COMPARE(MAXMETADATA_PER_NODE,GT,EQ)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_019
		RQ0606_020
		RQ0606_022
		RQ0606_034
		RQ0606_035

6.6.3.9 Other

6.6.3.9.1 FSS_0091 - Simultaneous file sessions on the same file

Test identification	FSS_0091
Test objectives	The File System Application shall be able to run two simultaneous file sessions on the same file.
Configuration reference	CFSS_005

Initial conditions

The procedure PFSS_0074 is successfully executed.
The procedure PFSS_0091 is successfully executed.

Test sequence

Step	Description	Requirements

1	FSCA_1 gate sends aFSS-0091-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0091-response-01 to FSCA_1 gate with:
	`-- ASN1START`
	`aFSS-0091-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA_2 gate sends aFSS-0091-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode {eWriteAccessMode, eReadAccessMode}`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0091-response-02 to FSCA_2 gate with:
	`-- ASN1START`
	`aFSS-0091-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_2 /*<STORE(aSessionID_2)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_028
		RQ0606_054
		RQ0606_055
		RQ0606_058
		RQ0606_019
		RQ1003_004
		RQ1003_005

5	FSCA_2 gate sends aFSS-0091-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-WRITE-Service-Command : {`
	`  aSessionID aSessionID_2, /*<REPLACE(aSessionID_2)>*/`
	`  aOffset 0,`
	`  aDataInfo aData : '1111111111'H`
	`}`
	`-- ASN1STOP`	

6	FSCS gate sends aFSS-0091-response-03 to FSCA_2 gate with:
	`-- ASN1START`
	`aFSS-0091-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-WRITE-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_2 /*<REPLACE(aSessionID_2)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_072
		RQ0606_074

7	FSCA_1 gate sends aFSS-0091-command-04 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-04 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

8	FSCS gate sends aFSS-0091-response-04 to FSCA_1 gate with:
	`-- ASN1START`
	`aFSS-0091-response-04 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '1111111111'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006
		RQ0606_020

9	FSCA_1 gate sends aFSS-0091-command-05 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-05 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_1 /*<REPLACE(aSessionID_1)>*/`
	`}`
	`-- ASN1STOP`	

10	FSCS gate sends aFSS-0091-response-05 to FSCA_1 gate with:
	`-- ASN1START`
	`aFSS-0091-response-05 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

11	FSCA_2 gate sends aFSS-0091-command-06 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0091-command-06 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-CLOSE-Service-Command : {`
	`  aSessionID aSessionID_2 /*<REPLACE(aSessionID_2)>*/`
	`}`
	`-- ASN1STOP`	

12	FSCS gate sends aFSS-0091-response-06 to FSCA_2 gate with:
	`-- ASN1START`
	`aFSS-0091-response-06 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-CLOSE-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_059
		RQ0606_061

6.6.3.9.2 FSS_0092 - Check if file session is closed

Test identification	FSS_0092
Test objectives	Test that all file sessions are closed upon closure of the pipe session between the file system control application gate and the file system control service gate.
Configuration reference	CFFS_001
Initial conditions	
The procedure PFSS_0091 is successfully executed.	
Test sequence	

Step	Description	Requirements

1	FSCA gate sends aFSS-0092-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0092-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file1,`
	`  aAccessMode  eReadAccessMode`
	`}`
	`-- ASN1STOP`	

2	FSCS gate sends aFSS-0092-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0092-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058

3	FSCA gate sends aFSS-0092-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0092-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

4	FSCS gate sends aFSS-0092-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0092-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aData '0102030405'H`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0606_057
		RQ0606_068
		RQ0606_070
		RQ1003_006

5	The administration gate in the Other host sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the File system service gate.	

6	FSCA gate sends aFSS-0092-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0092-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 5`
	`}`
	`-- ASN1STOP`	

7	FSCS gate sends aFSS-0092-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0092-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-BAD-SESSSION-ID`
	`}`
	`-- ASN1STOP`
		RQ1003_014

6.6.3.9.3 FSS_0093 - Check if data pipe session is closed

Test identification	FSS_0093	
Test objectives	Test that the file system data pipe sessions are closed upon closure of the pipe session between the file system control application gate and the file system control service gate.	
Configuration reference	CFFS_002	
Initial conditions		
The procedure PFSS_0095 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	FSCA gate sends aFSS-0093-command-01 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0093-command-01 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-OPEN-Service-Command : {`
	`  aNodeIdentity aShortName eFS-ID-file5,`
	`  aAccessMode  eReadAccessMode,`
	`  aGateURI '863391838cf658c28142d53611d52f12'H`
	`}`
	`-- ASN1START`	

2	FSCS gate sends aFSS-0093-response-01 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0093-response-01 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-OPEN-Service-Response : {`
	`  aFS-Service-Response eFS-OK,`
	`  aParameter {`
	`  aSessionID aSessionID_1 /*<STORE(aSessionID_1)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
	Pipe session is opened between the FSDS gate and the FSDA gate by using the 86339183-8cf6-58c2-8142-d53611d52f12 gate identifier.
		RQ0606_017
		RQ0606_018
		RQ0606_021
		RQ0606_026
		RQ0606_054
		RQ0606_055
		RQ0606_058
		RQ1003_015
		RQ1003_016

3	FSCA gate sends aFSS-0093-command-02 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0093-command-02 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 10`
	`}`
	`-- ASN1STOP`	

4	FSDS gate sends a stream with the content of file5 to FSDA gate	
5	Administration gate send an acknowledgement about receiving the content of file5 to administration gate in SCL host in the SSP host domain.	

6	FSCS gate sends aFSS-0093-response-02 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0093-response-02 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-OK`
	`}`
	`-- ASN1STOP`
		RQ0606_068
		RQ0606_071

7	The administration gate in the Other host sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate.	

8	FSCA gate sends aFSS-0093-command-03 to FSCS gate with:
	`-- ASN1START`
	`aFSS-0093-command-03 FS-CONTROL-SERVICE-GATE-Commands ::= aFS-OP-FILE-READ-Service-Command : {`
	`  aSessionID aSessionID_1, /*<REPLACE(aSessionID_1)>*/`
	`  aOffset 0,`
	`  aNumberOfBytes 10`
	`}`
	`-- ASN1STOP`	

9	FSCS gate sends aFSS-0093-response-03 to FSCA gate with:
	`-- ASN1START`
	`aFSS-0093-response-03 FS-CONTROL-SERVICE-GATE-Responses ::= aFS-OP-FILE-READ-Service-Response : {`
	`  aFS-Service-Response eFS-BAD-SESSSION-ID`
	`}`
	`-- ASN1STOP`
		RQ1003_014

6.6.3.9.4 FSS_0094 - Check the URN of SSP FS control service gate

Test identification	FSS_0094	
Test objectives	Check the URN of the SSP FS control service gate is listed in the GATE_LIST registry.	
Configuration reference	CFFS_003	
Initial conditions		
The SSP FS control service is available in the SSP.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).	
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.	
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The service identifier '366BD642-D7DE-584A-BD3B-A3DCE29FC075' shall be present.	
5	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed.	RQ1003_007

6.6.3.10 General Post Conditions

The General Post Conditions shall be executed after every test case in clause 6.6 (SSP File System).

General Post Conditions
The pipe session between the file system control application gate and the file system control service gate is closed. The SSP file system is deleted except the root directory eFS-Name-SSPFS.

6.6.3.11 Annex - End of ASN.1 structure

The annex shall be appended at the end of the SSP File System test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

6.6.3.12 Implicitly tested requirements

The following requirements identified in clause 5.2.6 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified.

RQ0606_001	RQ0606_002	RQ0606_003	RQ0606_007
RQ0606_010	RQ0606_012	RQ0606_013	RQ0606_014
RQ0606_015	RQ0606_016	RQ1003_001	RQ1003_008
RQ1003_009	RQ1003_010	RQ1003_011	RQ1003_012
RQ1003_013			

6.6.3.13 Non tested requirements

The following requirements identified in clause 5.2.6 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible.

RQ0606_006	RQ0606_059
RQ1003_002	RQ1003_003

The following requirements are out of scope of the present document:

RQ1003_017, RQ1003_018, RQ1003_019, RQ1003_020

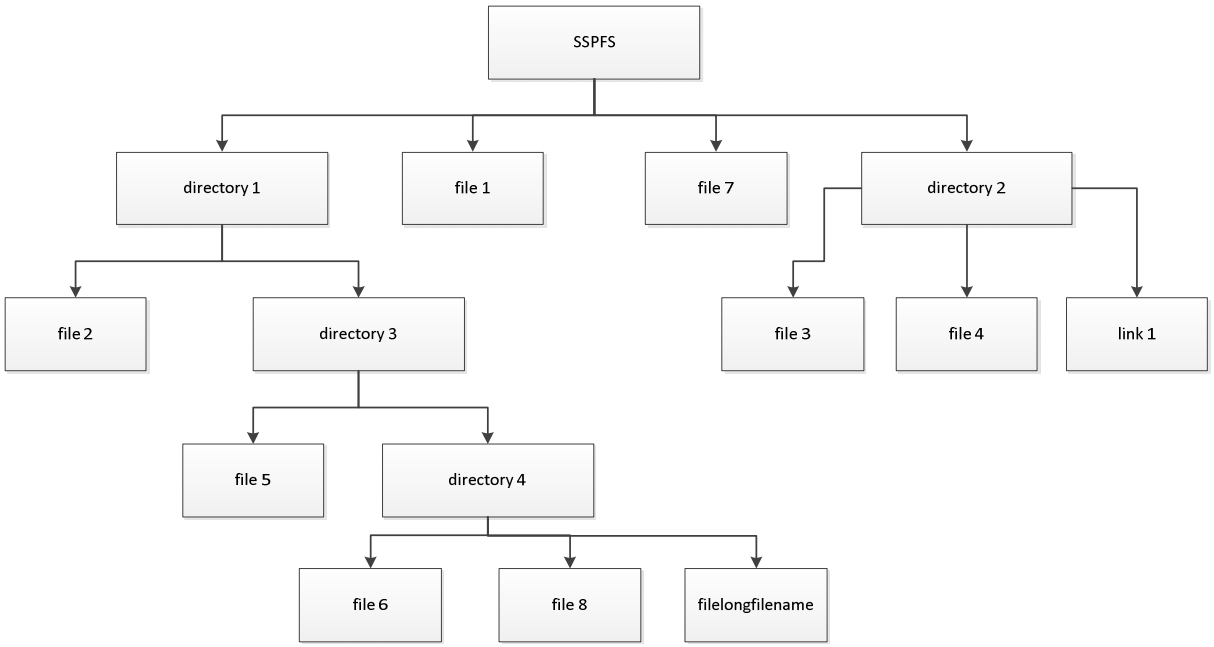
The following requirements are for further study:

RQ0606_009

The following requirements are not testable:

RQ0606_060

6.6.4 SSP File System configuration



aNodeName	aShortName	URN (NID:NSS)	aNode	aFileSize	aLinkedFileIdentity ->aShortName
SSPFS	eFS-ID-SSPFS	urn:etsi.org:SSPFS	aDirectory	-	-
directory1	eFS-ID-directory1	urn:etsi.org:SSPFS:directory1	aDirectory	-	-
directory2	eFS-ID-directory2	urn:etsi.org:SSPFS:directory2	aDirectory	-	-
directory3	eFS-ID-directory3	urn:etsi.org:SSPFS:directory1 :directory3	aDirectory	-	-
directory4	eFS-ID-directory4	urn:etsi.org:SSPFS:directory1 :directory3:directory4	aDirectory	-	-
file1	eFS-ID-file1	urn:etsi.org:SSPFS:file1	aFile	5	-
file2	eFS-ID-file2	urn:etsi.org:SSPFS:directory1 :file2	aFile	5	-
file3	eFS-ID-file3	urn:etsi.org:SSPFS:directory2 :file3	aFile	5	-
file4	eFS-ID-file4	urn:etsi.org:SSPFS:directory2 :file4	aFile	5	-
file5	eFS-ID-file5	urn:etsi.org:SSPFS:directory1 :directory3:file5	aFile	256	-
file6	eFS-ID-file6	urn:etsi.org:SSPFS:directory1 :directory3:directory4:file6	aFile	5	-
file7	eFS-ID-file7	urn:etsi.org:SSPFS:file7	aFile	5	-
file8	eFS-ID-file8	urn:etsi.org:SSPFS:directory1 :directory3:directory4:file8	aFile	5	-
filelongfilename	eFS-ID-filelongfilename	urn:etsi.org:SSPFS:directory1 :directory3:directory4:filelongfilename	aFile	5	-
link1	eFS-ID-link1	urn:etsi.org:SSPFS:directory2 :link1	aLink	5	'D44BD2F74D0B597BB70F2C66F2BE5F9B'H

aNodeName	aAccessorIdentity	aAccessorRights	aData	aNodeDirectoryIdentity ->aNodeReference
SSPFS	eFS-ACC-FSA1	eFS-ACL-SSPFS		
directory1	eFS-ACC-FSA1	eFS-ACL-directory1	-	eFS-Name-SSPFS
directory2	eFS-ACC-FSA1	eFS-ACL-directory2	-	eFS-Name-SSPFS
directory3	eFS-ACC-FSA1	eFS-ACL-directory3	-	"SSPFS:directory1"
directory4	eFS-ACC-FSA1	eFS-ACL-directory4	-	"SSPFS:directory1:directory3"
file1	eFS-ACC-FSA1	eFS-ACL-file1	'0102030405'	eFS-Name-SSPFS
file2	eFS-ACC-FSA1	eFS-ACL-file2	'1122334455'	"SSPFS:directory1"
file3	eFS-ACC-FSA1	eFS-ACL-file3	'3333333333'	"SSPFS:directory2"
file4	eFS-ACC-FSA1	eFS-ACL-file4	'4444444444'	"SSPFS:directory2"
file5	eFS-ACC-FSA1	eFS-ACL-file5	'010203040506070809 0A0B0C0D0E0F 101112131415161718 191A1B1C1D1E1F 202122232425262728 292A2B2C2D2E2F … E0E1E2E3E4E5E6E7 E8E9EAEBECEDEEE F F0F1F2F3F4F5F6F7F 8F9FAFBFCFDFEFF' (256 Bytes)	"SSPFS:directory1:directory3"
file6	eFS-ACC-FSA1	eFS-ACL-file6	'6666666666'	"SSPFS:directory1:directory3: directory4"
file7	eFS-ACC-FSA1	eFS-ACL-file7	'7777777777'	eFS-Name-SSPFS
file8	eFS-ACC-FSA1	eFS-ACL-file8	'8888888888'	"SSPFS:directory1:directory3: directory4"
filelongfilename	eFS-ACC-FSA1	eFS-ACL-filelongfilename	'1234567890'	"SSPFS:directory1:directory3: directory4"
link1	eFS-ACC-FSA1	eFS-ACL-link1		"SSPFS:directory2"

6.7 SSP identification

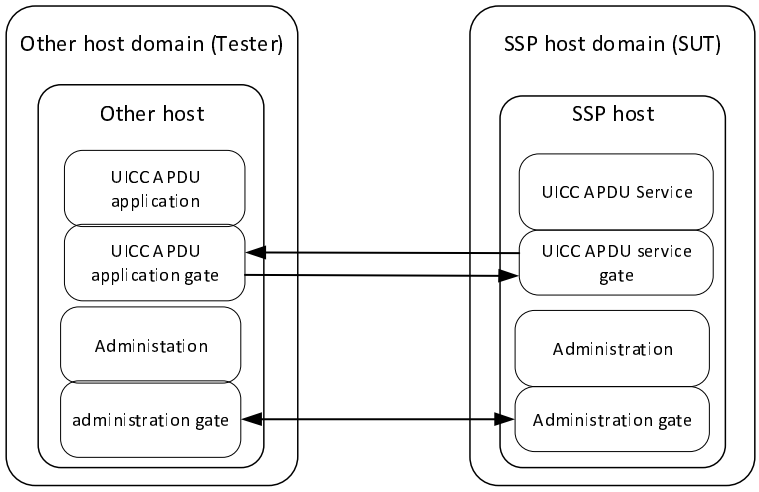
6.7.1 Requirements not tested

The following requirements identified in clause 5.2.7 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible: RQ0607_001.

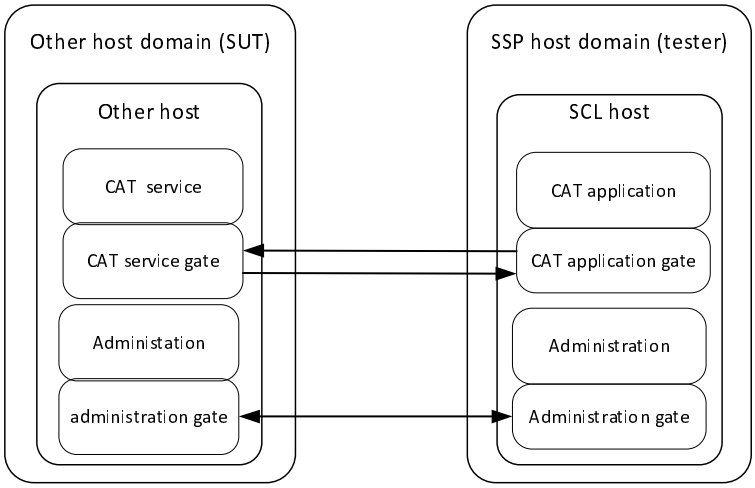
6.8 CAT-Runtime Environment

6.8.1 Configurations

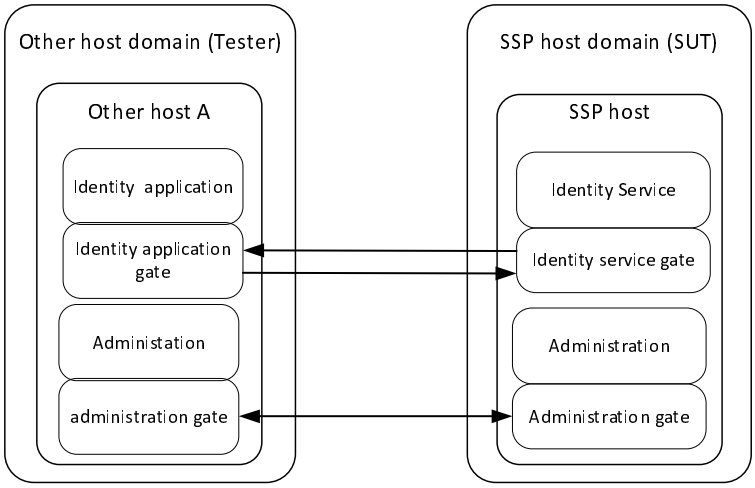
6.8.1.1 CCAT-RE_001

Configuration identification	CCAT-RE_001
Configuration description	

6.8.1.2 CCAT-RE_002

Configuration identification	CCAT-RE_002
Configuration description	

6.8.1.3 CCAT-RE_003

Configuration identification	CAT-RE_003
Configuration description	

6.8.2 Procedures

There are no procedures in this clause.

6.8.3 Test Descriptions

6.8.3.1 CAT-RE_001 - Open a pipe session with the identity gates

Test identification	CAT-RE_001	
Test objectives	The other host shall be able to open a pipe session to the identity gate of the SSP host. From the CAPABILITY_EXCHANGE registry, the capability of the SSP is extracted. The test is successful if: • A pipe session is open between the identity application in the other host and the identity service in the SSP host. • The CAPABILITY_EXCHANGE registry is present. • The SspUiccCapability record is readable from the CAPABILITY_EXCHANGE registry. • aSupportOfCardApplicationToolkit shall be TRUE	
Configuration reference	CCAT-RE_003	
Initial conditions		
The SCL host in the SSP host domain is present.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).	
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.	
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The service identifier 'dd61116f-f0dd-57f4-8a4f-52ee70276f24' shall be present. The test is successful if the previous requirement is satisfied.	
5	Identity application gate sends any_get_parameter (0x80) to the identity service gate.	

6	Identity service gate sends any_OK with **SspUiccCapability** to the Identity application gate. SSP capabilities shall have "SspUiccCapability" record.
	value1 SSPCapability ::= {
	aSspRelease '0000'H,
	aSspClass eSSPClass-Integrated,
	aSspUicc {
	aNumberOfLogicalChannels 4,
	aProactivePollingRequirement TRUE,
	aSupportOfUiccFileSystem TRUE,
	aSupportOfCardApplicationToolkit TRUE,
	aCardApplicationToolkitCapabilities '00'H
	}
	}
	and aSupportOfCardApplicationToolkit shall be TRUE.
	aCardApplicationToolkitCapabilities shall be present.
		RQ0608_002
		RQ1008_001

7	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed. This step is required to clean up the context of the tests but it is not essential for the test objective.	

6.8.3.2 CAT-RE_002 - Open a pipe session with the CAT gates

Test identification	CAT-RE_002	
Test objectives	The Other host shall be able to open a pipe session to the CAT application gate of the SSP host. From the gate_list registry, the uuid of the root accessor shall be listed. If the test is successful, then a pipe session is open between the CAT application in the SSP host and the CAT service in the Other host. The gate identifier of the CAT service is FF00453F-B0D5-59CE-B0D4-3AE178432F73 and is related to the RE only in order to be independent of the configuration.	
Configuration reference	CCAT-RE_001	
Initial conditions		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the CAT service gate. • $GATE_{CAT}$: The UUID gate identifier of the CAT gate (FF00453F-B0D5-59CE-B0D4-3AE178432F73).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the CAT application gate. • $GATE_{CAT}$: The UUID gate identifier of the CAT gate (FF00453F-B0D5-59CE-B0D4-3AE178432F73).	RQ0608_002

6.8.3.3 CAT-RE_003 - Open a pipe session with the APDU UICC gates

Test identification	CAT-RE_003	
Test objectives	The Other host shall be able to open a pipe session to the UICC APDU gate of the SSP host. If the test is successful, then a pipe session is open between the UICC APDU application in the Other host and the UICC APDU service in the SSP host. The gate identifier of the UICC APDU service is B9A3405D-1017-59AD-B959-2689DBEFF652.	
Configuration reference	CCAT-RE_001	
Initial conditions		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the UICC APDU application gate. • $GATE_{APDU}$: The UUID gate identifier of the UICC APDU gate (B9A3405D-1017-59AD-B959-2689DBEFF652).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the UICC APDU application gate. • $GATE_{APDU}$: The UUID gate identifier of the UICC APDU gate (B9A3405D-1017-59AD-B959-2689DBEFF652).	RQ0608_002

6.8.3.4 CAT-RE_004 - UICC capability

Test identification	CAT-RE_004	
Test objectives	CAT-RE specific field "SspUiccCapability" shall be in Capabilities of the SSP and aToolkitTerminalProfile shall be in TerminalCapability.	
Configuration reference	CCAT-RE_003	
Initial conditions		
The test CAT-RE_001 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate sends any_get_parameter (0x80) to the identity service gate.	

2	Identity service gate sends any_OK with **SspUiccCapability** to the Identity application gate.
	SSP capabilities shall have "SspUiccCapability" field.
	value1 SSPCapability ::= {
	aSspRelease '0000'H,
	aSspClass eSSPClass-Integrated,
	aSspUicc {
	aNumberOfLogicalChannels 4,
	aProactivePollingRequirement TRUE,
	aSupportOfUiccFileSystem TRUE,
	aSupportOfCardApplicationToolkit TRUE,
	aCardApplicationToolkitCapabilities '00'H
	}
	}
	and aSupportOfCardApplicationToolkit shall be TRUE.
	aCardApplicationToolkitCapabilities shall be present.
		RQ0608_002

6.8.3.5 CAT-RE_005 - Exchange Capabilities

Test identification	CAT-RE_005	
Test objectives	The other host shall retrieve from a SCL host in the SSP host domain supporting the UICC APDU service the SSPCapability record as defined in ETSI TS 103 666-1 [1], clause 6.4.2.5.	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_003 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	The UICC APDU application gate sends EVT_C-APDU event to UICC APDU service gate. The EVT-C-APDU contains the EXCHANGE CAPABILITIES as defined in ETSI TS 103 666-1 [1], clause 10.2.3.2. The terminal capability is defined in ETSI TS 103 666-1 [1], clause 6.4.2.4.	RQ0604_002 RQ0608_003
2	The UICC APDU service gate sends EVT_R-APDU event to UICC APDU application gate. The EVT_C-APDU event contains SSPCapability as defined in ETSI TS 103 666-1 [1], clause 6.4.2.5.	RQ0608_004

6.8.3.6 CAT-RE_006 - Event toolkit event

Test identification	CAT-RE_006	
Test objectives	Before sending the CAT command, SSP Host issues EVT_TOOLKIT_REQUEST to the other host.	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	The UICC APDU application gate sends EVT_C-APDU event to UICC APDU service gate. The EVT-C-APDU contains an APDU command, targeted to installed SSP application.	
2	The UICC APDU service gate sends EVT_R-APDU event to UICC APDU application gate. The EVT_C-APDU event contains an event EVT_TOOLKIT_REQUEST to the other host.	RQ0608_005

6.8.3.7 CAT-RE_007 - EXCHANGE CAPABILITIES Events

Test identification	CAT-RE_007	
Test objectives	During the EXCHANGE CAPABILITIES exchange, SSP Host shall trigger two events EVENT_PROFILE_DOWNLOAD and EVENT_FIRST_COMMAND_AFTER_ATR to all application installed in SSP host.	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
5	During the EXCHANGE CAPABILITIES exchange, SSP Host shall trigger two events EVENT_PROFILE_DOWNLOAD and EVENT_FIRST_COMMAND_AFTER_ATR as defined in ETSI TS 102 241 [14] and pass them to all the installed applications in the SSP Host.	RQ0608_006

6.8.3.8 CAT-RE_008 - CAT command exchanges

Test identification	CAT-RE_008	
Test objectives	SCL host in the SSP host domain shall be able to send and receive the CAT command via CAT application gate.	
Configuration reference	CCAT-RE_002	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	The CAT application gate sends CAT command to CAT service gate.	RQ0608_007
2	The CAT service gate sends CAT response to CAT application gate.	

6.8.3.9 CAT-RE_009 - CAT event triggers

Test identification	CAT-RE_009	
Test objectives	CAT-RE shall Trigger the applets based on events received by the CAT application gate.	
Configuration reference	CCAT-RE_002	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed. The test CAT-RE_008 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	Received events on CAT Application gate of SSP Host, CAT-RE shall forward these events to the installed application in the SSP Host.	RQ0608_008

6.8.3.10 CAT-RE_010 - CAT events

Test identification	CAT-RE_010	
Test objectives	CAT-RE shall Trigger the applets based on events received by the CAT application gate.	
Configuration reference	CCAT-RE_003	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	During the EXCHANGE CAPABILITIES exchange, SSP Host shall trigger two events EVENT_PROFILE_DOWNLOAD and EVENT_FIRST_COMMAND_AFTER_ATR as defined in ETSI TS 102 241 [14] and forward them to all installed applications in SSP Host.	RQ0608_009

6.8.3.11 CAT-RE_011 - External and file update events

Test identification	CAT-RE_011	
Test objectives	At file update of UICC file system, CAT-RE of SSP Host generate the EVENT_EXTERNAL_FILE_UPDATE and EVENT_REMOTE_FILE_UPDATE event.	
Configuration reference	CCAT-RE_003	
Initial conditions		
The test CAT-RE_001 shall be successfully passed. The test CAT-RE_005 shall be successfully passed. The test CAT-RE_008 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	AT files update, CAT-RE raise two events EVENT_EXTERNAL_FILE_UPDATE and EVENT_REMOTE_FILE_UPDATE according to ETSI TS 102 241 [14] and pass to install applet in SSP host.	RQ0608_010

6.8.3.12 Implicitly tested requirements

The following requirements identified in clause 5.2.8 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0608_001, RQ0608_003, RQ0608_011.

6.9 SSP Suspension

6.9.1 Configurations

There are no specific configurations for this topic. Configurations from the CAT-Runtime Environment clause of the present document are used in the following test descriptions.

6.9.2 Procedures

There are no specific procedures for this topic.

6.9.3 Test Descriptions

6.9.3.1 CAT-SUSPENSION_001 - Saving current state

Test identification	CAT-SUSPENSION_001	
Test objectives	SSP suspends and save its current state.	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	Other host shall send UICC suspension Command (defined in ETSI TS 102 221 [7]) to SSP host and SSP host shall suspend. SSP Host shall save its current state. • status of selected applications on each logical channel; • security context related to PIN verification status for each application; • selected EF, record pointer and tag pointer for each logical channel; • status of toolkit applications.	RQ0609_003 RQ0609_007
2	Tester shall deactivate the physical interface of SSP so that there is no power supply to the SSP.	RQ0609_002

6.9.3.2 CAT-SUSPENSION_002 - Resume last suspended state

Test identification	CAT-SUSPENSION_002	
Test objectives	SSP resumes from its last suspended state	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	Other host shall send UICC suspension Command (defined in ETSI TS 102 221 [7]) to SSP host and SSP host shall suspend.	RQ0609_003 RQ0609_007
2	Tester shall deactivate the physical interface of SSP and there is no Power supply to the SSP.	RQ0609_002
3	SSP host initialization process shall be the same as the previous "EXCHANGE CAPABILITIES" process.	RQ0609_005
4	Mandatory static gates shall open and the UICC APDU pipe session shall be open between SSP Host and Other host.	
5	Outside SSP host shall send UICC resume Command (defined in ETSI TS 102 221 [7]) to SSP host.	
6	SSP host shall load all the saved states: • status of selected applications on each logical channel; • security context related to PIN verification status for each application; • selected EF, record pointer and tag pointer for each logical channel; • status of toolkit applications.	RQ0609_004
7	Other host shall be able to access the UICC application and need not require PIN verification (if it is verified before suspend).	RQ0609_004

6.9.3.3 CAT-SUSPENSION_003 - Suspension rejection

Test identification	CAT-SUSPENSION_003	
Test objectives	In case SCL is used, SSP suspension shall be rejected when there is more than 1 pipe (only pipe available is for transporting APDUs as defined in ETSI TS 103 666-1 [1], clause 10.2.8) to the SSP.	
Configuration reference	CCAT-RE_001	
Initial conditions		
The test CAT-RE_001 shall be successfully passed.		
Test sequence		
Step	**Description**	**Requirements**
1	Tester creates more than one UICC APDU pipe session.	
2	Other host shall send UICC suspension Command (defined in ETSI TS 102 221 [7]) to SSP host and SSP host shall reject the UICC suspension Command.	RQ0609_008

6.9.3.4 Implicitly tested requirements

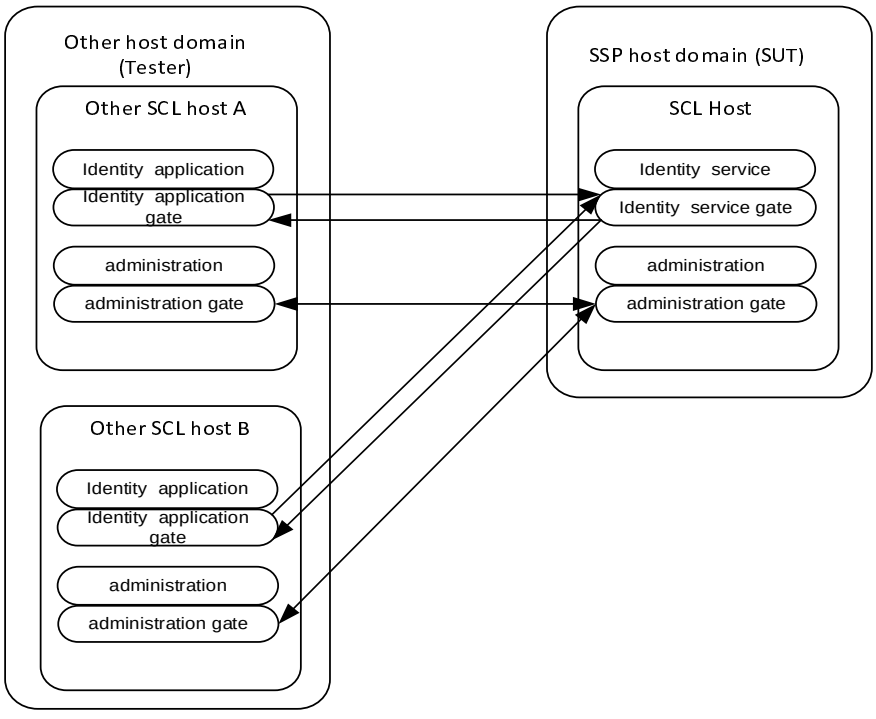
The following requirements identified in clause 5.2.9 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0609_001, RQ0609_002, RQ0609_003, RQ0609_006, 0609_007.

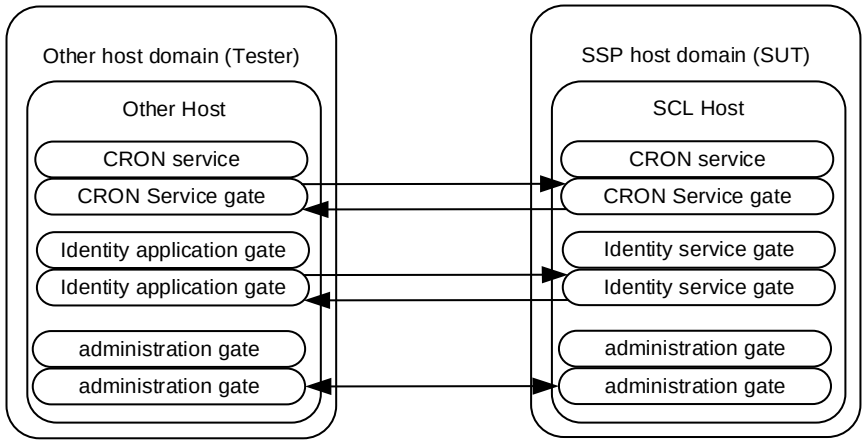
6.10 SSP Applications

6.10.1 Configurations

6.10.1.1 CAPP_001

Configuration identification	CAPP_001
Configuration description	

6.10.1.2 CAPP_002

Configuration identification	CAPP_002
Configuration description	

6.10.2 Procedures

There are no specific procedures for this topic.

6.10.3 Test Descriptions

6.10.3.1 APP_001

Test identification	APP_001	
Test objectives	If there are no restrictions of the execution environment and/or of the application protocol, one SSP Application shall not block another SSP Application from exchanging data with the terminal on a different SSP interface session.	
Configuration reference	CAPP_001	
Initial conditions		
• SSP shall support SCL layer. • SSP Host shall open two SSP interface sessions one with "other host A" and second with "other host B". • The Identity pipe session shall open between "SSP Host and other host A" and "SSP host and other host B".		
Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate of other host A sends any_get_parameter to the identity service gate.	RQ0610_003
2	Identity service gate of SSP host sends any_OK to the Identity application gate in other host A on same SSP interface session.	
3	Identity application gate of other host B sends any_get_parameter to the identity service gate.	
4	Identity service gate of SSP host sends any_OK to the Identity application gate of other host B on same SSP interface session. One Identity application shall not block another Identity application	

6.10.3.2 APP_002

Test identification	APP_002	
Test objectives	If there are no restrictions of the execution environment and/or of the application protocol, one SSP Application shall not block another SSP Application from exchanging data with the terminal on the same SSP interface session.	
Configuration reference	CAPP_002	
Initial conditions		
• SSP shall support SCL layer and CRON application gate as described in ETSI TS 103 666-1 [1]. • The CRON pipe session shall be open between SSP Host and Other host. • The Identity pipe session shall open between SSP Host and Other host.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate of the other host sends any_get_parameter to the identity service gate.	RQ0610_004
2	Identity service gate of SSP host sends any_OK to the Identity application gate in the other host.	
3	CRON application registers a timer in a CRON service in order to receive a notification (i.e. event) at a given time and date in the future.	
4	After time and date expiration, CRON service gate of the other host notifies the CRON application gate.	

6.10.3.3 Requirements not testable, implicitly verified or verified elsewhere

6.10.3.3.1 Requirements implicitly verified

The following requirements identified in clause 5.2.10 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0610_002.

6.10.3.3.2 Requirements verified elsewhere

The following requirements identified in clause 5.2.10 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body: RQ0610_005, RQ0610_006.

6.11 SSP security

6.11.1 Requirements not testable, implicitly verified or verified elsewhere

6.11.1.1 Requirements verified elsewhere

The following requirements identified in clause 5.2.11 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0611_004	RQ0611_005	RQ0611_006
RQ0611_007	RQ0611_008	RQ0611_009
RQ0611_010	RQ0611_014	RQ0611_015
RQ0611_016	RQ0611_017	

6.11.1.2 Requirements implicitly verified

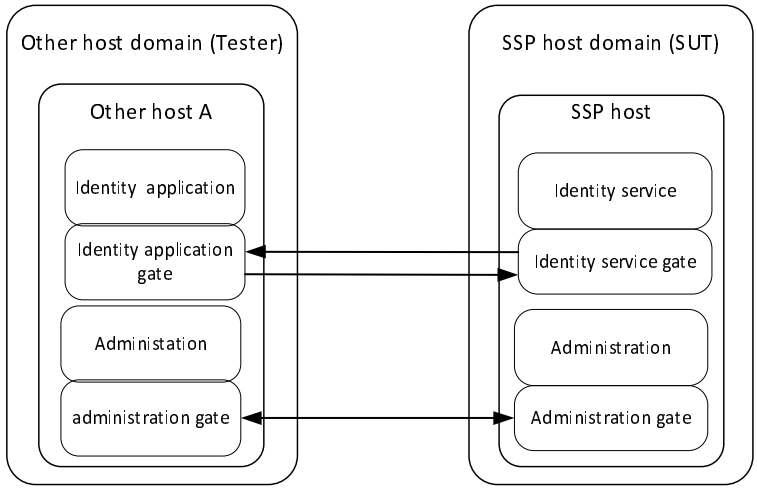
The following requirements identified in clause 5.2.11 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0611_001	RQ0611_002	RQ0611_003
RQ0611_011	RQ0611_012	RQ0611_013
RQ0611_018		

6.12 User interface

6.12.1 Configurations

6.12.1.1 CSSPUI_001

Configuration identification	CSSPUI_001
Configuration description	

6.12.1.2 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPINIconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) initialization (1)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   SSPClass ,
   SSPCapability,
   TerminalCapability,
VersionType
FROM SSPDefinitions;
-- ASN1STOP
```

6.12.2 Procedures

6.12.2.1 PSSPUI_001 - Open a pipe session with the Identity gate of the SSP host

Procedure identification	PSSPUI_001
Procedure objectives	The other host shall be able to open a pipe session to the identity gate of the SSP host.
Configuration reference	CSSPUI_001
Initial conditions	
The SSP host is registered to the SCL network controller host.	
Procedure sequence	
Step	**Description**
1	Administration gate in the other host sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate in the SSP host sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).

6.12.3 Test descriptions

6.12.3.1 SSPUI_001 - SSPCapabilities of SSPUI

Test identification	SSPUI_001
Test objectives	To test that the SSP indicates a URL in the capability exchange.
Configuration reference	CSSPUI_001
Initial conditions	

The procedure PSSPUI_001 is successfully executed.

```
-- ASN1START
aTrue BOOLEAN ::= TRUE    /*<STORE(aTrue)>*/
aFalse BOOLEAN ::= FALSE    /*<STORE(aFalse)>*/

aEMPTY_1 UTF8String ::= "" /*<STORE(aEMPTY_1)>*/
aEMPTY_2 OCTET STRING ::= ''H /*<STORE(aEMPTY_2)>*/
aSSPRELEASE VersionType::= '0F00'H /* <STORE(aSSPRELEASE)> *//* it indicates the release of the
present document that is implemented by the SSP*/
aSSPCLASS_1 SSPClass ::= eSSPClass-Integrated /* <STORE(aSSPCLASS_1)> */
aSSPCLASS_2 SSPClass ::= eSSPClass-Embedded-Type1 /* <STORE(aSSPCLASS_2)> */
aSSPCLASS_3 SSPClass ::= eSSPClass-Embedded-Type2 /* <STORE(aSSPCLASS_3)> */
aSSPCLASS_4 SSPClass ::= eSSPClass-Removable /* <STORE(aSSPCLASS_4)> */

aNBLOGICALCHANNELS_MIN INTEGER ::= 1 /* <STORE(aNBLOGICALCHANNELS_MIN)> *//* it indicates the
minimum nb of logical channels, including the default channel, that can be supported by an SSP*/
aNBLOGICALCHANNELS_MAX INTEGER ::= 14 /* <STORE(aNBLOGICALCHANNELS_MAX)> *//* it indicates the
maximum nb of logical channels, including the default channel, that can be supported by an SSP*/
-- ASN1STOP
```

Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate sends ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	

2	The Identity service gate sends aResponse to the Identity application gate.
	`-- ASN1START`
	`aResponse SSPCapability ::= {`
	`  aSspRelease '0000'H,`
	`/*<COMPARE(aSSPRELEASE,GT,EQ)>*/`
	`  aSspVendorName "0",`
	`/*<ISFIELDNOTEXIST()> OR <COMPARE(aEMPTY_1,DIF)>*/`
	`  aSspClass eSSPClass-Integrated /*<COMPARE(aSSPCLASS_1,EQ)> OR`
	`<COMPARE(aSSPCLASS_2,EQ)> OR <COMPARE(aSSPCLASS_3,EQ) OR`
	`<COMPARE(aSSPCLASS_4,EQ)>*/,`
	`  aClassSpecificCapabilities OCTET STRING : '00'H /*<ISFIELDNOTEXIST()> OR`
	`<COMPARE(aEMPTY_2,DIF)>*/,`
	`  aSspUicc {`
	`    aNumberOfLogicalChannels 1,`
	`/*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aNBLOGICALCHANNELS_MIN,EQ,GT)> AND`
	`<COMPARE(aNBLOGICALCHANNELS_MAX,EQ,LS)>`
	`*/`
	`    aProactivePollingRequirement FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse,EQ)> */`
	`    aSupportOfUiccFileSystem FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse,EQ)> */`
	`    aSupportOfCardApplicationToolkit FALSE, /*<ISFIELDNOTEXIST> OR`
	`<COMPARE(aTrue,EQ)> OR <COMPARE(aFalse, EQ)> */`
	`    aCardApplicationToolkitCapabilities '00'H`
	`/*<ISFIELDNOTEXIST> OR <COMPARE(aEMPTY_2,DIF)>*/`
	`  },`
	`  aSspUserInterface {`
	`    aUrl '00'H`
	`/*COMPARE(aEMPTY_1,DIF)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ0612_01

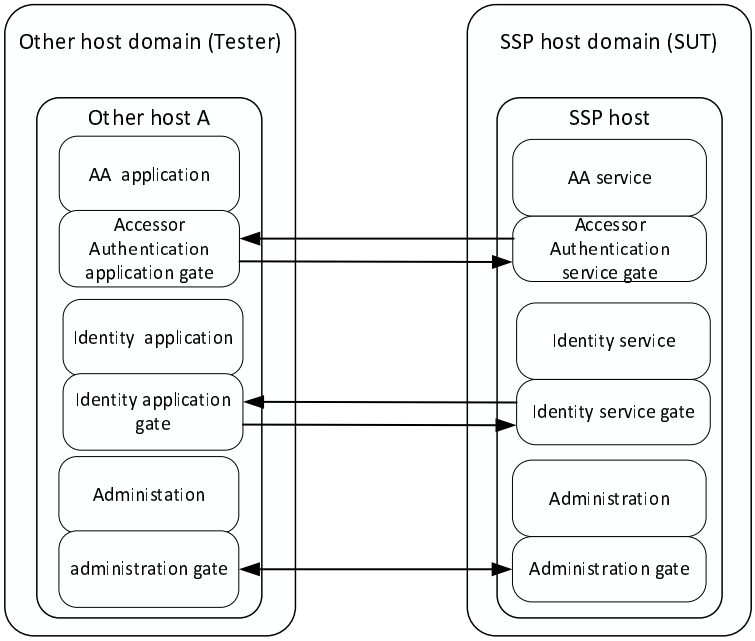
6.12.3.2 Non tested Requirements

The following requirements are not tested in the current version of the present document:
RQ0612_02, RQ0612_03, RQ0612_04

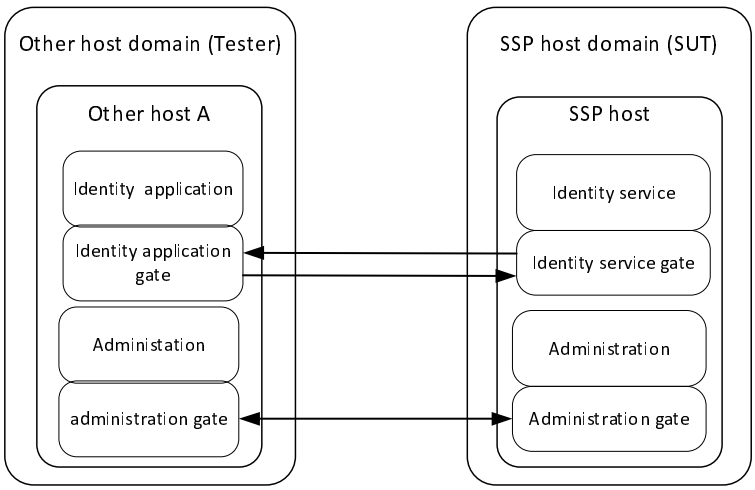
6.13 Accessor authentication service

6.13.1 Configurations

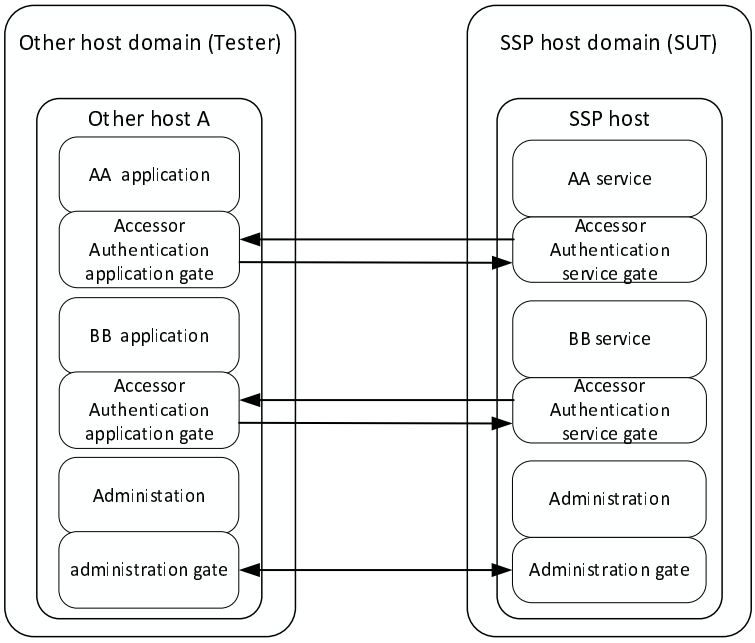
6.13.1.1 CAAS_001 - Accessor and Identity services

Configuration identification	CAAS_001
Configuration description	

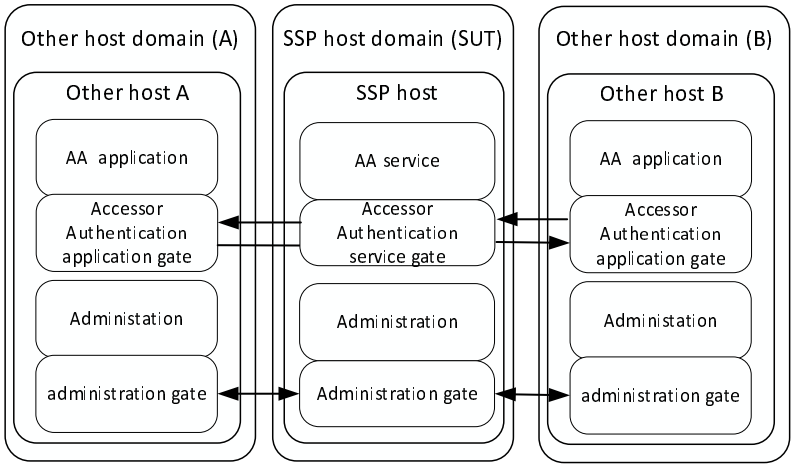
6.13.1.2 CAAS_002 - Identity service

Configuration identification	CAAS_002
Configuration description	

6.13.1.3 CAAS_003 - Generic Accessor

Configuration identification	CAAS_003
Configuration description	

6.13.1.4 CAAS_004 - Multiple host domains

Configuration identification	CAAS_004
Configuration description	

6.13.1.5 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPAASconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) aas (3)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   NodeName,  -- RFC5280 Certificate X.509v3
   AccessMode,
   UUID,
   SessionID,
   AccessorRights,
   AAS-CONTROL-SERVICE-GATE-Commands,
   AAS-CONTROL-SERVICE-GATE-Responses,
   Certificate,
   AuthenticationToken,
   AccessorConditionsPIN,
   AccessorConditions,
   Version,
   VersionType
FROM SSPDefinitions
   SubjectPublicKeyInfo
FROM PKIX1Explicit88
   ECDSA-Sig-Value,
id-ecPublicKey
FROM PKIX1Algorithms88;

eAASVersion VersionType ::='0100' --Version 01.00

-- urn:etsi.org:asn.1:accessor:test:1
eAS-ID-ACC-TEST-1    UUID::='7DFF3B1C6C345A49BC36F1380CEAA0C2'H
-- urn:etsi.org:asn.1:accessor:test:2
eAS-ID-ACC-TEST-2    UUID::='E23D733361D158A995EAF795649548F6'H
--urn:etsi.org:asn.1:accessor:test:group:1
eAS-ID-ACC-TEST-GROUP-1 UUID::='cb807bb95f6452fbade0fbbb3bfb3562'H
-- urn:etsi.org:asn.1:accessor:root
eAS-ID-ACC-ANONYMOUS    UUID::= '4E46645FE6005A70AD7A60D6E5345E0B'H
--  urn:etsi.org SSP:ASN.1:Anonymous
eAS-ID-ACC-ROOT      UUID::='DD61116FF0DD57F48A4F52EE70276F24'H
eAS-ID-AAS-Service      UUID::='DD61116FF0DD57F48A4F52EE70276F24'H
eAS-ID-AAS-GateID    UUID::='AAAAAAAABBBBCCCCDDDDEEEEEEEEEEE'H
eAS-Challenge        UUID::='BA64E9EE888952F4891DA79401758FF4'H

--eAASAccessRight-RequiresSecurePipe AccessorRights ::= {eRight-Bit1 }
--eAASAccessRight-Create AccessorRights ::= { eRight-Bit2 }
--eAASAccessRight-Delete AccessorRights ::= { eRight-Bit3 }
--eAASAccessRight-Update AccessorRights ::= { eRight-Bit4 }
--eAASAccessRight-UpdateACL AccessorRights ::= { eRight-Bit5 }
--eAASAccessRight-UpdateGroup AccessorRights ::= { eRight-Bit6 }
--eAASAccessRight-UpdateCredentialPolicy AccessorRights ::= { eRight-Bit7 }
--eAASAccessRight-UpdateCredentialStatus AccessorRights ::= { eRight-Bit8 }

-- The root accessor rights

eAS-ACL-ROOT            AccessorRights ::= {
--eAASAccessRight-RequiresSecurePipe--   eRight-Bit1,
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--       eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateGroup--     eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }
-- The TEST 1 accessor may update its ACL
eAS-ACL-TEST-1          AccessorRights ::= {

--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }
```

```
-- The TEST 1 accessor cannot update its ACL
eAS-ACL-TEST-1-F          AccessorRights ::= {

                    }

-- The TEST 2 accessor rights

eAS-ACL-TEST-2    AccessorRights ::= {
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--       eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateGroup--     eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
}

-- The TEST-GROUP-2 accessor ACL
eAS-ACL-TEST-GROUP-2          AccessorRights ::= {

--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }
-- The TEST 2 accessor rights

eAS-ACL-TEST-GROUP-1    AccessorRights ::= {
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--       eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateGroup--     eRight-Bit6
}

eAS-ACL-ROOT-GROUP      AccessorRights ::= {
--eAASAccessRight-UpdateGroup--     eRight-Bit6,

--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
}

-- Host domains identifiers

-- urn:etsi.org:asn.1:REE:hostdomain:A
eAS-ID-HOST-DOMAIN-A UUID::= '4af8347ad30358e29efbcebed01981d7'H
-- urn:etsi.org:asn.1:REE:hostdomain:B
eAS-ID-HOST-DOMAIN-B UUID::= 'EE9294D5B21558ECB0338A1F69386CA7'H

-- ASN1STOP
```

6.13.2 Procedures

6.13.2.1 PAAS_021 - Open a pipe session with the Identity gate

Procedure identification	PAAS_021
Procedure objectives	The other host shall be able to open a pipe session to the identity gate of the SSP host. From the GATE_LIST registry, the UUID of the root accessor shall be listed. If the procedure is successful then a pipe session is open between the identity application in the other host and the identity service in the SSP host.
Configuration reference	CAAS_002
Initial conditions	
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. The root accessor is available in SSP prepared for tests purpose. The tester acting as an accessor shall be able to be authenticated by using an authentication token authenticated by a certification path.	

Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The gate identifier 'DD61116F-F0DD-57F4-8A4F-52EE70276F24' shall be present. The procedure is successful if the previous requirement is satisfied.

6.13.2.2 PAAS_022 - Open a pipe session with the ROOT Accessor Authentication service

Procedure identification	PAAS_022
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CAAS_001
Initial conditions	
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. This UUID is also the identity of the Root accessor. This root accessor is dedicated for the tester and assigned to the test providers using the ETSI SSP tests. The procedure AAS_021 shall be successfully executed: • The ROOT accessor is present in the GATE_LIST registry of the Identity gate	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{ROOT}$: The UUID gate identifier of the root Accessor Authentication service gate (DD61116F-F0DD-57F4-8A4F-52EE70276F24).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{ROOT}$: The UUID gate identifier of the root Accessor Authentication application gate (DD61116F-F0DD-57F4-8A4F-52EE70276F24). $GATE_{ROOT}$ shall be present in one of the binding parameters (see VNP in GlobalPlatform: "Technology Virtual Primary Platform" [10]. If present then the procedure is successful.

6.13.2.3 PAAS_023 - Open a pipe session with the Anonymous Accessor Authentication service of the Anonymous Accessor

Procedure identification	PAAS_023
Procedure objectives	The other host shall be able to open a pipe session to the authentication service gate of the Anonymous Accessor in the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CAAS_001
Initial conditions	
Anonymous accessor (UUID: 4E46645F-E600-5A70-AD7A-60D6E5345E0B) is existing. This UUID is also the identity of the anonymous accessor. This anonymous accessor shall be available in the SSP host.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. $GATE_{ANONYMOUS\_ACCESSOR}$: The UUID gate identifier of the anonymous Accessor Authentication service gate (4E46645F-E600-5A70-AD7A-60D6E5345E0B).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{ANONYMOUS\_ACCESSOR}$: The UUID gate identifier of the anonymous Accessor Authentication application gate (4E46645F-E600-5A70-AD7A-60D6E5345E0B). $GATE_{ANONYMOUS\_ACCESSOR}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the procedure is successful.

6.13.2.4 PAAS_024 - Open a pipe session with the TEST-1 Accessor Authentication service

Procedure identification	PAAS_024
Procedure objectives	The other host shall be able to open a pipe session to the (TEST-1 accessor) authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CAAS_003
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. $GATE_{TEST\text{-}1}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST\text{-}1}$: The UUID gate identifier of the test accessor AA application gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2). $GATE_{TEST\text{-}1}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.

6.13.2.5 PAAS_025 - Open a pipe session with the TEST-2 Accessor Authentication service

Procedure identification	PAAS_025
Procedure objectives	The other host shall be able to open a pipe session to the (TEST-2 accessor) authentication service gate of the SSP host. If the procedure is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.
Configuration reference	CAAS_003
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. $GATE_{TEST-2}$: The UUID gate identifier of the test accessor AA service gate (E23D733361D158A995EAF795649548F6).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST-2}$: The UUID gate identifier of the test accessor AA application gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA E23D733361D158A995EAF795649548F60C2). $GATE_{TEST-2}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the procedure is successful.

6.13.2.6 PAAS_026 - Close a pipe session with an Accessor Authentication service

Procedure identification	PAAS_026
Procedure objectives	The other host shall close a pipe session on the SSP host. This procedure is generic and the pipe identifier assigned by the other host shall be stored by the test tool. This procedure shall be used each time a test description shall restart from a procedure where a pipe session is already open for a given gate.
Configuration reference	CAAS_003
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the accessor AA service gate. The pipe session between the AA application gate and the AA service gate is closed. This step is required to clean up the context of the tests but it is not essential for the test objective.

6.13.3 Test descriptions

6.13.3.1 Root accessor

6.13.3.1.1 AAS_311 - Authentication of the ROOT accessor

Test identification	AAS_311	
Test objectives	The root accessor shall be able to be authenticated with the Accessor Authentication service by using: • The aAAS-OP-GET-CHALLENGE-Service-Command command. • The aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command. The authentication mean is based on the authentication tokens.	
Configuration reference	CAAS_001	
Initial conditions		
The following procedure shall be executed in order: • PAAS_021: The ROOT accessor is present in the GATE_LIST registry of the identity gate. • PAAS_026: The pipe session with the identity gate is closed. • PAAS_022: The pipe session with the ROOT accessor authentication service is opened.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-311-command-01 to AAS gate with:
	`-- ASN1START`
	`aAAS-311-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CHALLENGE-Service-Command : {}`
	`-- ASN1STOP`	

2	AAS gate sends aAAS-311-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-311-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CHALLENGE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aChallenge eAS-Challenge,`
	`  aCertificates {eAS-CERT-01}}}`
	`-- ASN1STOP`
	aCertificate is a set of certificates.
	aChallenge is a random number (128 bit) generated by the AAS. The value expressed in the test is given as example.
		RQ0613_145
		RQ0613_061
		RQ0613_146
		RQ0613_147
		RQ0613_061

3	AAA gate sends an aAAS-311-command-02 command to AAS gate with:
	`-- ASN1START`
	`aAAS-311-command-02  AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {`
	`  aToken eAS-ATK-01, aTokenCertificationPath {eAS-CERT-01}}}`
	`-- ASN1STOP`
	The authentication token shall contain the challenge as recovered at the step 2.
	The authentication token shall be verified by using the certification path.	

4	AAS gate sends an aAAS-311-response-02 response to AAA gate with:
	`-- ASN1START`
	`aAAS-311-response-02 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {  aToken eAS-ATK-01, aTokenCertificationPath  {eAS-CERT-01} }`
	`}`
	`-- ASN1STOP`
	The authentication token shall contain the challenge as recovered at the step 2.
	The authentication token shall be verified by using the certification path.
	The test is successful if the same challenge is in all authentication tokens and all of them have been verified by their certification path.
		RQ0613_027
		RQ0613_034
		RQ0613_081
		RQ0613_134
		RQ0613_001
		RQ0613_136
		RQ0613_138

6.13.3.1.2 AAS_312 - Access to the Authentication Service from the ROOT accessor

Test identification	AAS_312	
Test objectives	The root accessor shall be able to retrieve the gate identifier for opening a secure pipe session with the Accessor Authentication service by using: • The aAAS-OP-ACCESS-SERVICE-Service-Command command. The test description allows to open a secure pipe session with the ROOT accessor authentication service.	
Configuration reference	CAAS_001	
Initial conditions		
The following test shall be successfully executed: • AAS_311: the ROOT accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate an aAAS-312-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-312-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {`
	`  aServiceIdentifier eAS-ID-AAS-Service,  aUseSecurePipe TRUE}`
	`-- ASN1STOP`	

2	AAS gate an aAAS-312-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-312-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {  aGateIdentifier eAS-ID-AAS-GateID  }}`
	`-- ASN1STOP`
	The AAS returns the gate identifier on which the authenticated root accessor can access the accessor authentication service by using a secure pipe.
	The test is successful if the AAS returns eAAS-OK.
		RQ0613_006
		RQ0613_008
		RQ0613_035
		RQ0613_139
		RQ0613_143
		RQ0613_036
		RQ0613_142
		RQ0613_144
		RQ0613_004

6.13.3.1.3 AAS_313 - Open a pipe session with the ROOT Accessor Authentication service

Test identification	AAS_313	
Test objectives	The other host shall be able to open a pipe session to the ROOT accessor authentication service gate of the SSP host. If the test is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.	
Configuration reference	CAAS_001	
Initial conditions		
Root accessor (UUID: DD61116F-F0DD-57F4-8A4F-52EE70276F24) is existing. This UUID is also the identity of the Root accessor. This root accessor is dedicated for the tester and assigned to the test providers using the ETSI SSP tests. The accessor has obtained the gate identifier on the accessor authentication service for the root accessor by using a secure pipe session. The test AAS_312 shall be successfully executed. The ROOT accessor is authenticated		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{CD}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. $GATE_{ROOTBIS}$: The dynamically assigned UUID gate identifier returned by AAS in AAS_312(AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{DC}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{ROOTBIS}$: The dynamically assigned UUID gate identifier returned by AAS in AAS_312(AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE). $GATE_{ROOTBIS}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful. A secure pipe session is opened between the AAA acting for the root accessor and AAS as the authentication service.	

6.13.3.1.4 AAS_314 - Access to the Authentication Service from the ROOT accessor (w/o secure pipe)

<table>
<tr><td colspan="2">Test identification</td><td>AAS_314</td></tr>
<tr><td colspan="2">Test objectives</td><td>The root accessor shall be able to be authenticated with the Accessor Authentication service by using:
• The aAAS-OP-ACCESS-SERVICE-Service-Command command.
The authentication mean is based on the authentication tokens. The ACL of the ROOT accessor shall mandate a secure pipe.</td></tr>
<tr><td colspan="2">Configuration reference</td><td>CAAS_001</td></tr>
<tr><td colspan="3">Initial conditions</td></tr>
<tr><td colspan="3">The test AAS_311 shall be successfully executed.</td></tr>
<tr><td colspan="3">Test sequence</td></tr>
<tr><td>Step</td><td>Description</td><td>Requirements</td></tr>
<tr><td>1</td><td>AAA gate an aAAS-314-command-01 command to AAS gate with:
<pre>-- ASN1START
aAAS-314-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {
  aServiceIdentifier eAS-ID-AAS-Service, aUseSecurePipe FALSE}
-- ASN1STOP</pre></td><td></td></tr>
<tr><td>2</td><td>AAS gate an aAAS-314-response-01 response to AAA gate with:
<pre>-- ASN1START
aAAS-314-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {
  aAAS-Service-Response eAAS-E-NOK,
  aParameter { aGateIdentifier eAS-ID-AAS-GateID }}
-- ASN1STOP</pre>The AAS returns eAAS-E-NOK because accessor shall access the accessor authentication service by using a secure pipe.
The test is successful if the AAS returns eAAS-E-NOK.</td><td>RQ0613_140
RQ0613_149</td></tr>
</table>

6.13.3.2 Creation of the TEST-1 accessor (pincode based)

6.13.3.2.1 AAS_321 - Creation of the TEST-1 accessor (without violations)

Test identification	AAS_321	
Test objectives	The Accessor Authentication application shall be able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The authentication mean shall be based on the pincode.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_313 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-321-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-321-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL`
	`{{`
	`aAccessorIdentity eAS-ID-ACC-ROOT,aAccessorRights eAS-ACL-ROOT`
	`},{`
	`aAccessorIdentity eAS-ID-ACC-TEST-1, aAccessorRights eAS-ACL-TEST-1`
	`}}`
	`},`
	`aCredential {aPinNumericCredential "1234"},`
	`aCredentialsPolicy {aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,`
	`aMinSize 4,`
	`aMaxSize 255,`
	`aMaxAttempts 2`
	`}},`
	`aCredentialsStatus {aPinNumericStatus`
	`  {aCommonStatus {aIsDisabled FALSE}}`
	`}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-321-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-321-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_005
		RQ0613_009
		RQ0613_010
		RQ0613_012
		RQ0613_013
		RQ0613_024
		RQ0613_025
		RQ0613_028
		RQ0613_029
		RQ0613_031
		RQ0613_054
		RQ0613_055
		RQ0613_064
		RQ0613_065
		RQ0613_066
		RQ0613_067
		RQ0613_082
		RQ0613_100
		RQ0613_101
		RQ0613_102
		RQ0613_103
		RQ0613_106

6.13.3.2.2 AAS_322 - Open a pipe session with the TEST-1 Accessor Authentication service

Test identification	AAS_322	
Test objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the test is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_041 shall be successfully executed. • The TEST-1 accessor has been created.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	RQ0613_020

6.13.3.2.3 AAS_323 - Authentication of the TEST-1 accessor

Test identification	AAS_323	
Test objectives	The Accessor Authentication application shall be able to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_322 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends aAAS-323-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-323-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1234"}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-323-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-323-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_054
		RQ0613_131
		RQ0613_026
		RQ0613_136

6.13.3.2.4 AAS_324 - Authentication of the TEST-1 accessor (failed)

Test identification	AAS_324	
Test objectives	The Accessor Authentication application shall not be able to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command and wrong credentials. Wrong value is sent to authenticate the accessor. The test is successful if the authentication is failed.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be successfully executed: • AAS_321. The TEST-1 accessor is created. • AAS_322. A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends aAAS-324-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-324-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1235"`
	`}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-324-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-324-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED,`
	`  aParameter aCredentialsStatus : {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE,`
	`  aRemainingAttempts 1`
	`  }}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_137
		RQ0613_141

6.13.3.2.5 AAS_325 - Authentication of the TEST-1 accessor (failed)

Test identification	AAS_325	
Test objectives	The Accessor Authentication application shall fail to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command if the credentials are wrong.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be successfully executed: • AAS_321. The TEST-1 accessor is created. • AAS_322. A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate an aAAS-325-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-325-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPasswordCredential : "HelloWorld2020"`
	`}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-325-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-325-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-E-CMD-PAR-UNKNOWN,`
	`  aParameter aCredentialsStatus : {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE,`
	`  aRemainingAttempts 3`
	`  }}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-E-CMD-PAR-UNKNOWN.
		RQ0613_056
		RQ0613_132
		RQ0613_148

6.13.3.2.6 AAS_326 - Authentication of the TEST-1 accessor (failed)

Test identification	AAS_326
Test objectives	The Accessor Authentication application shall fail to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command if the credentials are wrong.
Configuration reference	CAAS_003

Initial conditions

The test AAS_044 shall be successfully executed twice and the remaining attempts shall not be tested for the second test. The aRemainingAttempts is set to 1.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-326-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-326-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1235"`
	`}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-326-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-326-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED,`
	`  aParameter aCredentialsStatus : {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled TRUE,`
	`  aRemainingAttempts 0`
	`  }}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_130
		RQ0613_136

6.13.3.2.7 AAS_327 - Deletion of the TEST-1 accessor

Test identification	AAS_327
Test objectives	The Accessor Authentication application shall be able to delete an accessor from the Accessor Authentication service using an aAAS-ADMIN-DELETE-ACCESSOR-Service-Command.
Configuration reference	CAAS_003

Initial conditions

The test AAS_321 shall be successfully executed:

- The ROOT accessor is duly authenticated.
- The TEST-1 accessor has been created by the ROOT accessor.

Test sequence

Step	Description	Requirements

1	AAA gate sends aAAS-327-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-327-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-327-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-327-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_017
		RQ0613_127
		RQ0613_128
		RQ0613_045
		RQ0613_129

6.13.3.2.8 AAS_328 - Creation of the TEST-2 accessor (with violations)

Test identification	AAS_328
Test objectives	The Accessor Authentication application shall be not able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command due to a violation of the ACL. The authentication mean shall be based on the pincode.
Configuration reference	CAAS_003

Initial conditions

The test AAS_043 shall be successfully executed:
- The TEST-1 accessor is authenticated.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-328-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-328-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  },`
	`  {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorRights eAS-ACL-TEST-2`
	`  }`
	
	`}},`
	`  aCredential {aPinNumericCredential "1234"},`
	`  aCredentialsPolicy {aPinNumericPolicy`
	`  {aIsDisableForbidden FALSE,  aMinSize 4,  aMaxSize 255, aMaxAttempts 3  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinNumericStatus { aCommonStatus { aIsDisabled FALSE}}}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-328-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-328-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.
		RQ0613_029
		RQ0613_037
		RQ0613_044
		RQ0613_040
		RQ0613_100
		RQ0613_101
		RQ0613_102
		RQ0613_103
		RQ0613_003
		RQ0613_150

6.13.3.2.9 AAS_329 - Creation of the TEST-2 accessor (without violations)

Test identification	AAS_329
Test objectives	The Accessor Authentication application shall be able to create the TEST-2 accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The authentication mean shall be based on the pincode.
Configuration reference	CAAS_003

Initial conditions

The test AAS_313 shall be successfully executed:
- The ROOT accessor is authenticated.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-329-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-329-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL`
	`{{`
	`aAccessorIdentity eAS-ID-ACC-ROOT,aAccessorRights eAS-ACL-ROOT`
	`},{`
	`aAccessorIdentity eAS-ID-ACC-TEST-2, aAccessorRights eAS-ACL-TEST-2`
	`}}`
	`},`
	`aCredential {aPinNumericCredential "1234"},`
	`aCredentialsPolicy {aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,`
	`aMinSize 4,`
	`aMaxSize 255,`
	`aMaxAttempts 3`
	`}},`
	`aCredentialsStatus {aPinNumericStatus`
	`  {aCommonStatus {aIsDisabled FALSE}}`
	`}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-329-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-329-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.	

6.13.3.2.10 AAS_3210 - Authentication of the TEST-2 accessor

Test identification	AAS_3210	
Test objectives	The Accessor Authentication application shall be able to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_329 shall be successfully executed: • Creation of the TEST-1 Accessor Authentication. The procedure PAAS_025 shall be successfully executed: opening of a pipe session on the TEST-2 accessor authentication service.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends aAAS-3210-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-3210-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinNumericCredential : "1234"}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-3210-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-3210-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_054
		RQ0613_131
		RQ0613_026
		RQ0613_136

6.13.3.3 Creation of the TEST-1 accessor (password based)

6.13.3.3.1 AAS_331 - Creation of the TEST-1 accessor

Test identification	AAS_331	
Test objectives	The Accessor Authentication application shall be able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The accessor authentication means shall be based on a password.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_313 shall be successfully executed: • A pipe session is opened with the ROOT Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends aAAS-331-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-331-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  }}},`
	`  aCredential {aPinPasswordCredential "HelloWorld2020"},`
	`  aCredentialsPolicy {aPinPasswordPolicy {`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aRequiresLowerCaseLetter TRUE,`
	`  aRequiresUpperCaseLetter TRUE,`
	`  aRequiresNumber TRUE,`
	`  aRequiresSymbol TRUE,`
	`  aMaxAttempts 6`
	`  }},`
	`  aCredentialsStatus {`
	`  aPinPasswordStatus {`
	`  aCommonStatus {aIsDisabled FALSE}}}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using password code.	

2	AAS gate sends an aAAS-331-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-331-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_020
		RQ0613_029
		RQ0613_032
		RQ0613_056
		RQ0613_068
		RQ0613_069
		RQ0613_074
		RQ0613_070
		RQ0613_071
		RQ0613_100
		RQ0613_072
		RQ0613_073
		RQ0613_101
		RQ0613_102
		RQ0613_103

6.13.3.3.2 AAS_332 - Open a pipe session with the Accessor Authentication service for the TEST-1 accessor

Test identification	AAS_332	
Test objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the test is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_051 shall be successfully executed. • The TEST-1 accessor has been created.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	RQ0613_020

6.13.3.3.3 AAS_333 - Authentication of the TEST-1 accessor

Test identification	AAS_333	
Test objectives	The Accessor Authentication application shall be able to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_332 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-333-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-333-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPasswordCredential : "HelloWorld2020"}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-333-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-333-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_135
		RQ0613_016
		RQ0613_136

6.13.3.3.4 AAS_334 - Authentication of the TEST-1 accessor (failure)

Test identification	AAS_334	
Test objectives	The Accessor Authentication application shall fail to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command if the credentials are wrong.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_332 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-334-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-334-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPasswordCredential : "NoHelloWorld2019"}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-334-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-334-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED,`
	`  aParameter aCredentialsStatus : {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE,`
	`  aRemainingAttempts 2}}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_135
		RQ0613_030
		RQ0613_151
		RQ0613_141

6.13.3.3.5 AAS_335 - Deletion of an accessor

Test identification	AAS_335	
Test objectives	The Accessor Authentication application shall be able to delete an accessor from the Accessor Authentication service using an aAAS-ADMIN-DELETE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_331 shall be successfully executed: • The TEST-1 accessor has been created.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-335-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-335-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-335-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-335-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_017
		RQ0613_038
		RQ0613_126

6.13.3.3.6 AAS_336 - Authentication of the TEST-1 accessor (POLICY RULES VIOLATION)

Test identification	AAS_336	
Test objectives	The Accessor Authentication application shall fail to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command if the credentials are wrong.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_332 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-336-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-336-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPatternCredential : {`
	`  { x 1,  y 2 },  { x 2, y 4 }, { x 5, y 1 }, { x 7, y 1  }}}`
	
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-336-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-336-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-POLICYRULES-VIOLATION}`
	`-- ASN1STOP`
	`The test is successful if the aAAS-Service-Response is eAAS-POLICYRULES-VIOLATION`
		RQ0613_152
		RQ0613_153

6.13.3.4 Creation of the TEST-1 accessor (pattern based)

6.13.3.4.1 AAS_341 - Creation of the TEST-1 accessor

Test identification	AAS_0061
Test objectives	The Accessor Authentication application shall be able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The accessor authentication mean shall be based on a pattern.
Configuration reference	CAAS_003

Initial conditions

The test AAS_0033 shall be successfully executed:

- A pipe session is opened with the ROOT Accessor Authentication Service gate.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-341-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-341-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  }}},`
	`  aCredential {`
	`  aPinPatternCredential {  { x 1,  y 2 },  { x 2, y 4 }, { x 5, y 1 }, { x 7, y 1  }  }`
	`  },`
	`  aCredentialsPolicy {`
	`  aPinPatternPolicy {`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aEntryPanelMinSize 3,`
	`  aSamePointMultipleTimes FALSE,`
	`  aMaxAttempts 0`
	`  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinPatternStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE}}}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using a pattern.	

2	AAS gate sends an aAAS-341-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-341-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_029
		RQ0613_033
		RQ0613_057
		RQ0613_059
		RQ0613_060
		RQ0613_075
		RQ0613_076
		RQ0613_077
		RQ0613_078
		RQ0613_079
		RQ0613_098
		RQ0613_099
		RQ0613_100
		RQ0613_101
		RQ0613_102
		RQ0613_103
		RQ0613_104
		RQ0613_105
		RQ0613_096
		RQ0613_118
		RQ0613_106
		RQ0613_058

6.13.3.4.2 AAS_342 - Open a pipe session with the TEST-1 Accessor Authentication service

Test identification	AAS_342	
Test objectives	The other host shall be able to open a pipe session to the authentication service gate of the SSP host. If the test is successful then a pipe session is open between the accessor authentication application in the other host and the accessor authentication service in the SSP host.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_061 shall be successfully executed. • The TEST-1 accessor has been created.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST}$: The UUID gate identifier of the test accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2). $GATE_{TEST}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	RQ0613_020

6.13.3.4.3 AAS_343 - Authentication of the TEST-1 accessor

Test identification	AAS_343	
Test objectives	The Accessor Authentication application shall be able to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_342 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-343-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-343-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPatternCredential : {`
	`  { x 1,  y 2 },  { x 2, y 4 }, { x 5, y 1 }, { x 7, y 1  }}}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-343-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-343-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_136
		RQ0613_057
		RQ0613_133

6.13.3.4.4 AAS_344 - Authentication of the TEST-1 accessor (failure)

Test identification	AAS_344	
Test objectives	The Accessor Authentication application shall fail to authenticate an accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command if the credentials are wrong.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_342 shall be successfully executed: • A pipe session is opened with the TEST-1 Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-344-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-344-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aPinPatternCredential : {`
	`  { x 1,  y 2 },  { x 2, y 4 }, { x 5, y 1 }, { x 5, y 1  }}}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-344-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-344-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED,`
	`  aParameter aCredentialsStatus : {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE,`
	`  aRemainingAttempts 2}}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_030
		RQ0613_151
		RQ0613_141

6.13.3.4.5 AAS_345 - Deletion of an accessor

Test identification	AAS_345	
Test objectives	The Accessor Authentication application shall be able to delete an accessor from the Accessor Authentication service using an aAAS-ADMIN-DELETE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_0061 shall be successfully executed: • The TEST-1 accessor has been created.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-345-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-345-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-345-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-345-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.	

6.13.3.4.6 AAS_346 - Creation of the TEST-1 accessor with no update rights

Test identification	AAS_346
Test objectives	The Accessor Authentication application shall be able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The accessor authentication mean shall be based on a pattern.
Configuration reference	CAAS_003

Initial conditions

The test AAS_0033 shall be successfully executed:

- A pipe session is opened with the ROOT Accessor Authentication Service gate.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-346-command-01 command to AAS gate with:
	`-- ASN1START`
	
	`aAAS-346-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1-F`
	`  }}},`
	`  aCredential {`
	`  aPinPatternCredential {  { x 1,  y 2 },  { x 2, y 4 }, { x 5, y 1 }, { x 7, y 1  }  }`
	`  },`
	`  aCredentialsPolicy {`
	`  aPinPatternPolicy {`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aEntryPanelMinSize 3,`
	`  aSamePointMultipleTimes FALSE,`
	`  aMaxAttempts 0`
	`  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinPatternStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE}}}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-346-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-346-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_049
		RQ0613_051
		RQ0613_052

6.13.3.4.7 AAS_347 - Creation of the TEST-1 accessor

Test identification	AAS_347	
Test objectives	The Accessor Authentication application shall be able to create an accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The accessor authentication mean shall be based on a pattern with multiple time the same point.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_0033 shall be successfully executed: • A pipe session is opened with the ROOT Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-347-command-01 command to AAS gate with:
	`-- ASN1START`
	
	`aAAS-347-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  }}},`
	`  aCredential {`
	`  aPinPatternCredential {  { x 1,  y 2 },  { x 2, y 4 }, { x 1, y 2 }, { x 7, y 1  }  }`
	`  },`
	`  aCredentialsPolicy {`
	`  aPinPatternPolicy {`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aEntryPanelMinSize 3,`
	`  aSamePointMultipleTimes TRUE,`
	`  aMaxAttempts 0`
	`  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinPatternStatus {`
	`  aCommonStatus {`
	`  aIsDisabled FALSE}}}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-347-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-347-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_060

6.13.3.4.8 AAS_348 - Self-authentication of the TEST-1 accessor

Test identification	AAS_348
Test objectives	The Accessor Authentication application shall be able to be authenticated without using the authentication procedure if its credentials are disabled.
Configuration reference	CAAS_003

Initial conditions

The following tests shall be successfully executed:
- AAS_0082: the TEST-1 accessor has disabled its credentials.

Test sequence

Step	Description	Requirements
1	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the TEST-1 Accessor Authentication service gate. The pipe session is closed.	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the Accessor Authentication service gate. $GATE_{TEST-1}$: The UUID gate identifier of the TEST-1 accessor AA service gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2).	
3	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the Accessor Authentication application gate. • $GATE_{TEST-1}$: The UUID gate identifier of the TEST-1 accessor AA application gate (7DFF3B1C-6C34-5A49-BC36-F1380CEAA0C2). $GATE_{TEST-1}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	

4	AAA gate sends an aAAS-348-command-02 command to AAS gate with:
	`-- ASN1START`
	`aAAS-348-command-02 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aCredentialsPolicy {`
	`  aPinNumericPolicy {`
	`  aIsDisableForbidden FALSE,`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aMaxAttempts 1`
	`  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled TRUE,`
	`  aRemainingAttempts 1}}}}`
	`-- ASN1STOP`
	The AAA updates the pincode credential.	

5	AAS gate sends an aAAS-348-response-02 response to AAA gate with:
	`-- ASN1START`
	`aAAS-348-response-02 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_083

6.13.3.5 Capability of the TEST-1 accessor

6.13.3.5.1 AAS_351 - Capability of an accessor (eGlobalAuthenticationService)

Test identification	AAS_351
Test objectives	The Accessor Authentication application shall be able to get the capability of the Accessor Authentication service using an aAAS-OP-GET-CAPABILITIES-Service-Command. eGlobalAuthenticationService is requested.
Configuration reference	CAAS_003

Initial conditions

The test AAS_343 shall be successfully executed:

- The TEST-1 accessor is authenticated.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-351-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-351-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CAPABILITIES-Service-Command : {`
	`  aRequestType eGlobalAuthenticationService}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-351-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-351-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CAPABILITIES-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter aGlobalAuthenticationService : {`
	`  aAASVersion eAASVersion,`
	`  aAccessorList {`
	`  aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorConditions {aAccessConditionsTokens eTokenCertificate},`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  }}},`
	`  aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorConditions {aAccessConditionsPIN ePinPattern},`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  }}}}}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_088
		RQ0613_087
		RQ0613_090
		RQ0613_091
		RQ0613_092
		RQ0613_093

6.13.3.5.2 AAS_352 - Capability of an accessor (eAccessorStatus)

Test identification	AAS_352	
Test objectives	The Accessor Authentication application shall be able to get the capability of the Accessor Authentication service using an aAAS-OP-GET-CAPABILITIES-Service-Command. eAccessorStatus is requested.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_343 shall be successfully executed: • The TEST-1 accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-352-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-352-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CAPABILITIES-Service-Command : {`
	`  aRequestType eAccessorStatus}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-352-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-352-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CAPABILITIES-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter aAccessorStatus : {`
	`  aIsAuthenticated TRUE,`
	`  aAccessorConditions { aAccessConditionsPIN ePinPattern }}}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_089
		RQ0613_090
		RQ0613_094
		RQ0613_095
		RQ0613_085
		RQ0613_097

6.13.3.5.3 AAS_353 - Capability of an accessor (eAccessorStatus)

Test identification	AAS_353	
Test objectives	The Accessor Authentication application shall be able to get the capability of the Accessor Authentication service using an aAAS-OP-GET-CAPABILITIES-Service-Command. eAccessorStatus is requested.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_311 shall be successfully executed: • The ROOT accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-353-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-353-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CAPABILITIES-Service-Command : {`
	`  aRequestType eAccessorStatus}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-353-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-353-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CAPABILITIES-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK`
	`}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_086

6.13.3.6 Update of the TEST-1 accessor

6.13.3.6.1 AAS_361 - Update of an accessor

Test identification	AAS_361	
Test objectives	The Accessor Authentication application shall be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_323 shall be successfully executed: • The TEST-1 accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-361-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-361-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aSetCredential {aPinNumericCredential "0000"}}`
	`-- ASN1STOP`
	The AAA updates the pincode credential.	

2	AAS gate sends an aAAS-361-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-361-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_107
		RQ0613_039
		RQ0613_041
		RQ0613_047
		RQ0613_125

6.13.3.6.2 AAS_362 - Update of an accessor

Test identification	AAS_362	
Test objectives	The Accessor Authentication application shall be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_323 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-362-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-362-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aCredentialsPolicy {`
	`  aPinNumericPolicy {`
	`  aIsDisableForbidden FALSE,`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aMaxAttempts 1`
	`  }`
	`  },`
	`  aCredentialsStatus {`
	`  aPinNumericStatus {`
	`  aCommonStatus {`
	`  aIsDisabled TRUE,`
	`  aRemainingAttempts 1}}}}`
	`-- ASN1STOP`
	The AAA updates the pincode credential.	

2	AAS gate sends an aAAS-362-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-362-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_048
		RQ0613_051
		RQ0613_052
		RQ0613_053
		RQ0613_084
		RQ0613_109
		RQ0613_111
		RQ0613_113
		RQ0613_114
		RQ0613_042
		RQ0613_043
		RQ0613_049
		RQ0613_116
		RQ0613_123
		RQ0613_124
		RQ0613_123
		RQ0613_125

6.13.3.6.3 AAS_363 - Update of an accessor (ACL violation)

<table>
<tr><td>Test identification</td><td colspan="2">AAS_363</td></tr>
<tr><td>Test objectives</td><td colspan="2">The Accessor Authentication application shall not be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command with out-of-range credentials.</td></tr>
<tr><td>Configuration reference</td><td colspan="2">CAAS_003</td></tr>
<tr><td colspan="3">Initial conditions</td></tr>
<tr><td colspan="3">The following tests shall be executed in order:
• AAS_346; creation of the TEST-1 accessor without update rights.
• AAS_343; authentication of the TEST-1 accessor.</td></tr>
<tr><td colspan="3">Test sequence</td></tr>
<tr><td>Step</td><td>Description</td><td>Requirements</td></tr>
<tr><td>1</td><td>AAA gate sends an aAAS-363-command-01 command to AAS gate with:
<pre>-- ASN1START
aAAS-363-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {
  aAccessorIdentity eAS-ID-ACC-TEST-1,
  aSetCredential {
    aPinPatternCredential {
      {x 1,y 1 },{x 2, y 2},{x 3,y 3},{x 4,y 4}
    }
  }
}
-- ASN1STOP</pre>The AAA updates the pincode credential.</td><td></td></tr>
<tr><td>2</td><td>AAS gate sends an aAAS-363-response-01 response to AAA gate with:
<pre>-- ASN1START
aAAS-363-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {
  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS}
-- ASN1STOP</pre>The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.</td><td>RQ0613_049</td></tr>
</table>

6.13.3.6.4 AAS_364 - Update of an accessor (ACL violation)

Test identification	AAS_364	
Test objectives	The Accessor Authentication application shall not be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command with out-of-range credentials.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be executed in order: • AAS_346; creation of the TEST-1 accessor without update rights. • AAS_343; authentication of the TEST-1 accessor.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-364-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-364-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aCredentialsStatus {`
	`  aPinPatternStatus {`
	`  aCommonStatus {`
	`  aIsDisabled TRUE,`
	`  aRemainingAttempts 5`
	`  }`
	`  }`
	`  }}`
	`-- ASN1STOP`
	The AAA updates the pincode credential.	

2	AAS gate sends an aAAS-364-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-364-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.
		RQ0613_051

6.13.3.6.5 AAS_365 - Update of an accessor (ACL violation)

Test identification	AAS_365	
Test objectives	The Accessor Authentication application shall not be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command with out-of-range credentials.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be executed in order: • AAS_346; creation of the TEST-1 accessor without update rights. • AAS_343; authentication of the TEST-1 accessor.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-365-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-365-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aCredentialsPolicy {`
	`  aPinPatternPolicy {`
	`  aMinSize 4,`
	`  aMaxSize 255,`
	`  aEntryPanelMinSize 3,`
	`  aSamePointMultipleTimes FALSE,`
	`  aMaxAttempts 0`
	`  }`
	`}`
	`  }`
	`-- ASN1STOP`
	The AAA updates the pincode credential.	

2	AAS gate sends an aAAS-365-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-365-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.
		RQ0613_049
		RQ0613_115

6.13.3.6.6 AAS_366 - Update of an accessor (remove accessor condition)

Test identification	AAS_366	
Test objectives	The Accessor Authentication application shall be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_361 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-366-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-366-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aRemoveAccessorConditions {`
	`  aAccessConditionsPIN ePinNumeric`
	`  }`
	`}`
	`-- ASN1STOP`
	The AAA updates the aAccessConditionsPIN condition.	

2	AAS gate sends an aAAS-366-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-366-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_120

6.13.3.6.7 AAS_367 - Update of an accessor (set credential)

Test identification	AAS_367	
Test objectives	The Accessor Authentication application shall be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_361 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-367-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-367-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aSetCredential {`
	`  aPinNumericCredential "0000"`
	`  }`
	`}`
	`-- ASN1STOP`
	The AAA updates the aAccessConditionsPIN condition.	

2	AAS gate sends an aAAS-367-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-367-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_121

6.13.3.6.8 AAS_368 - Update of an accessor (remove credential)

Test identification	AAS_368	
Test objectives	The Accessor Authentication application shall be able to remove a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_361 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-368-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-368-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aRemoveCredential {`
	`  aAccessConditionsPIN ePinNumeric`
	`  }`
	`}`
	`-- ASN1STOP`
	The AAA removes a credential.	

2	AAS gate sends an aAAS-367-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-368-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_122

6.13.3.6.9 AAS_369 - Update of an accessor (policy rule violation)

Test identification	AAS_369	
Test objectives	The Accessor Authentication application shall not be able to update a credential of an accessor of the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command when violating the policy rules.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_361 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-369-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-369-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aSetCredential {`
	`  aPinNumericCredential "000"`
	`  }`
	`}`
	`-- ASN1STOP`
	The AAA updates the aAccessConditionsPIN condition.	

2	AAS gate sends an aAAS-367-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-369-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-POLICY-RULES-VIOLATIONS }`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS- POLICY-RULES-VIOLATIONS.
		RQ0613_110

6.13.3.7 Deletion of a ROOT accessor (violation)

6.13.3.7.1 AAS_371 - Deletion of an accessor (violation)

Test identification	AAS_371	
Test objectives	The Accessor Authentication application shall not be able to delete an accessor from the Accessor Authentication service using an aAAS-ADMIN-DELETE-ACCESSOR-Service-Command if its rights are non sufficient.	
Configuration reference	CAAS_003	
Initial conditions		
The test AAS_333 shall be successfully executed: • The TEST-1 accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends aAAS-371-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-371-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-ROOT}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-371-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-371-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-DELETE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS }`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.
		RQ0613_003
		RQ0613_150

6.13.3.8 Authentication of the Anonymous accessor

6.13.3.8.1 AAS_381 - Authentication of the anonymous accessor

Test identification	AAS_381	
Test objectives	The Accessor Authentication application shall be able to authenticate an anonymous accessor from the Accessor Authentication service using an aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The test PAAS_023 shall be successfully executed: • A pipe session is opened with the Anonymous Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-343-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-381-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-381-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-381-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_019
		RQ0613_018

6.13.3.9 Creation of the TEST-GROUP-1 accessor

6.13.3.9.1 AAS_391 - Creation of the TEST-GROUP-1 accessor

Test identification	AAS_391
Test objectives	The Accessor Authentication application shall be able to create a group accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. TEST-GROUP-1 accessor has no credentials. TEST-GROUP-2 accessor has no credentials. TEST-1 accessor has all rights (eAS-ACL-TEST-GROUP-1). TEST-2 accessor may only update the policy and status of the credentials (eAS-ACL-TEST-GROUP-2).
Configuration reference	CAAS_004

Initial conditions

These tests shall be successfully executed according to this order:

- AAS_323:
  - The TEST-1 accessor has been authenticated.
- AAS_329:
  - The TEST-2 accessor has been created but is not authenticated.
  - The ROOT accessor is authenticated.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-391-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-391-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorGroup : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-GROUP-1,`
	`  aMembersOfGroup {`
	`  eAS-ID-ACC-TEST-1,`
	`  eAS-ID-ACC-TEST-2`
	`  },`
	`  aACL {`
	`  {aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-GROUP-1},`
	`  {aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorRights eAS-ACL-TEST-GROUP-2}`
	`  }}`
	`  }`
	
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-391-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-391-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_010
		RQ0613_011

6.13.3.9.2 AAS_392 - Update of the TEST-GROUP-1 accessor

Test identification	AAS_392	
Test objectives	The TEST-1 accessor via the Accessor Authentication application shall be able to update the members list of a group accessor from the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command. TEST-1 accessor has all rights (eAS-ACL-TEST-GROUP-1). TEST-2 accessor may only update the policy and status of the credentials (eAS-ACL-TEST-GROUP-2). The authentication mean shall be based on the host domain list.	
Configuration reference	CAAS_004	
Initial conditions		
These tests shall be successfully executed according to this order: • AAS_391: – The TEST-GROUP-1 has been created. • AAS_323: – The TEST-1 accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-392-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-392-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-GROUP-1,`
	`  aMembersOfGroup {`
	`  eAS-ID-ACC-ROOT,`
	`  eAS-ID-ACC-TEST-1,`
	`  eAS-ID-ACC-TEST-2`
	`  }`
	`}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-392-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-392-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK. The ROOT accessor has been added in the group of members.
		RQ0613_010
		RQ0613_011
		RQ0613_014
		RQ0613_050
		RQ0613_117
		RQ0613_112

6.13.3.9.3 AAS_393 - Update of the TEST-GROUP-1 accessor (violation of the ACL)

Test identification	AAS_393	
Test objectives	The TEST-2 accessor via the Accessor Authentication application shall fail to update the members list of a group accessor from the Accessor Authentication service using an aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command. TEST-1 accessor has the eAS-ACL-TEST-GROUP-1 rights.	
Configuration reference	CAAS_004	
Initial conditions		
These tests shall be successfully executed according to this order: • AAS_3210 • AAS_344		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-393-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-393-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Command : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-GROUP-1,`
	`  aMembersOfGroup {`
	`  eAS-ID-ACC-ROOT,`
	`  eAS-ID-ACC-TEST-1,`
	`  eAS-ID-ACC-TEST-2`
	`  }`
	`}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-393-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-393-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-UPDATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-ACL-RULES-VIOLATIONS}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-ACL-RULES-VIOLATIONS.
		RQ0613_010
		RQ0613_011

6.13.3.10 Creation of the TEST-1 accessor with grantor

6.13.3.10.1 AAS_3101 - Creation of the TEST-1 accessor (with grantor)

Test identification	AAS_3101
Test objectives	The Accessor Authentication application shall be able to create the TEST-1 accessor with a grantor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The authentication mean shall be based on the pincode. The grantor is the ROOT accessor
Configuration reference	CAAS_003

Initial conditions

The test AAS_313 shall be successfully executed.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aAAS-3101-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-3101-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL`
	`{{`
	`aAccessorIdentity eAS-ID-ACC-ROOT,aAccessorRights eAS-ACL-ROOT`
	`},{`
	`aAccessorIdentity eAS-ID-ACC-TEST-1, aAccessorRights eAS-ACL-TEST-1,`
	`aGrantorIdentity  eAS-ID-ACC-ROOT}}`
	`},`
	`aCredential {aPinNumericCredential "1234"},`
	`aCredentialsPolicy {aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,`
	`aMinSize 4,`
	`aMaxSize 255,`
	`aMaxAttempts 3`
	`}},`
	`aCredentialsStatus {aPinNumericStatus`
	`  {aCommonStatus {aIsDisabled FALSE}}`
	`}}`
	`-- ASN1STOP`
	The root accessor has all rights on the test accessor.
	The test accessor shall be authenticated by using the pin code.	

2	AAS gate sends an aAAS-3101-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-3101-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-OK.
		RQ0613_002
		RQ0613_007

6.13.3.10.2 AAS_3102 - Creation of the TEST-2 accessor (without authentication)

Test identification	AAS_3102	
Test objectives	The Accessor Authentication application shall fail to create the TEST-2 accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command. The authentication is not required because the grantor is not authenticated.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be executed in order: • AAS_3101. The TEST-1 accessor is created with the ROOT accessor as grantor. • PAAS_024. A pipe session is opened with the TEST-1 Accessor Authentication service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-3102-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-3102-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL`
	`{{`
	`aAccessorIdentity eAS-ID-ACC-ROOT,aAccessorRights eAS-ACL-ROOT`
	`},{`
	`aAccessorIdentity eAS-ID-ACC-TEST-1, aAccessorRights eAS-ACL-TEST-1`
	`}}`
	`},`
	`aCredential {aPinNumericCredential "1234"},`
	`aCredentialsPolicy {aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,`
	`aMinSize 4,`
	`aMaxSize 255,`
	`aMaxAttempts 3`
	`}},`
	`aCredentialsStatus {aPinNumericStatus`
	`  {aCommonStatus {aIsDisabled FALSE}}`
	`}}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-3102-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-3102-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_015

6.13.3.10.3 AAS_3103 - Creation of the TEST-2 accessor (authentication)

Test identification	AAS_3103	
Test objectives	The Accessor Authentication application shall be to create the TEST-2 accessor from the Accessor Authentication service using an aAAS-ADMIN-CREATE-ACCESSOR-Service-Command.	
Configuration reference	CAAS_003	
Initial conditions		
The following tests shall be executed in order: • AAS_321. The TEST-1 accessor has been created. • AAS_363. The TEST-1 accessor is authenticated as grantor. • AAS_332. The ROOT accessor is authenticated.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aAAS-3103-command-01 command to AAS gate with:
	`-- ASN1START`
	`aAAS-3103-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Command : {`
	`  aAccessor aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-2,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL`
	`{{`
	`aAccessorIdentity eAS-ID-ACC-ROOT,aAccessorRights eAS-ACL-ROOT`
	`},{`
	`aAccessorIdentity eAS-ID-ACC-TEST-1, aAccessorRights eAS-ACL-TEST-1`
	`}}`
	`},`
	`aCredential {aPinNumericCredential "1234"},`
	`aCredentialsPolicy {aPinNumericPolicy {`
	`aIsDisableForbidden FALSE,`
	`aMinSize 4,`
	`aMaxSize 255,`
	`aMaxAttempts 3`
	`}},`
	`aCredentialsStatus {aPinNumericStatus`
	`  {aCommonStatus {aIsDisabled FALSE}}`
	`}}`
	`-- ASN1STOP`	

2	AAS gate sends an aAAS-3103-response-01 response to AAA gate with:
	`-- ASN1START`
	`aAAS-3103-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-ADMIN-CREATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-NOT-AUTHENTICATED}`
	`-- ASN1STOP`
	The test is successful if the aAAS-Service-Response is eAAS-NOT-AUTHENTICATED.
		RQ0613_015

6.13.3.11 Annexes - Accessor Authentication ASN.1 descriptions

6.13.3.11.1 Annex - Certificates and Tokens

The Authentication token and certificates are given as example.

```
-- ASN1START
eAS-ATK-01 AuthenticationToken::={
      tbsToken {
        version v1,
        subjectPublicKeyInfo {
          algorithm {
            algorithm { 0 0 }
          },
          subjectPublicKey '0'B
        },
        aATK-Content {
          aChallenge '00000000000000000000000000000000'H,
          aKey-Size e128,
          aStreamCipherIdentifier aAES-CGM-StreamCipherIdentifier
        }
      },
```

```
      signatureAlgorithm {
        algorithm { 0 0 }
      },
      signature {
        r 0,
        s 0
      }
    }
eAS-CERT-01 Certificate ::= {
   tbsCertificate {
     version v3,
     serialNumber 1,
     signature {
      algorithm { 0 0 },
      parameters OCTET STRING : '00'H
     },
     issuer rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     validity {
      notBefore utcTime : "000101000000Z",
      notAfter utcTime : "000101000000Z"
     },
     subject rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     subjectPublicKeyInfo {
      algorithm {
        algorithm id-ecPublicKey
      },
      subjectPublicKey '0'B
     },
     issuerUniqueID '0'B,
     subjectUniqueID '0'B,
     extensions {
      {
        extnID { 0 0 },
        critical FALSE,
        extnValue '00'H
      }
     }
   },
   signatureAlgorithm {
     algorithm { 0 0 },
     parameters OCTET STRING : '00'H
   },
   signature '0'B
  }

-- ASN1STOP
```

6.13.3.11.2 Annex - ASN.1 stop

The annex shall be appended at the end of the accessor authentication test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

6.13.3.12 Requirements not testable, implicitly verified or verified elsewhere

6.13.3.12.1 Requirements not tested

The following requirements identified in clause 5.2.13 are not covered by the present document:

RQ0613_046, RQ0613_152, RQ0613_153.

6.13.3.12.2 Implicitly tested requirements

The following requirements identified in clause 5.2.13 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0613_021	RQ0613_022	RQ0613_025	RQ0613_023
RQ0613_028	RQ0613_034	RQ0613_035	RQ0613_062
RQ0613_063	RQ0613_080	RQ0613_108	RQ0613_053
RQ0613_119	RQ0613_046		

7 Test Descriptions: Physical interfaces

7.1 Overview

Tests for the physical interfaces are to be executed in accordance with their respective test specifications. If more than one interface is used parallel access shall be performed whenever possible.

7.2 Reset

All Test Descriptions to verify RQ0702_01 and RQ0702_02 found in ETSI TS 102 230-1 [2], clause 5 shall apply to the terminal. Reset tests for the other physical interfaces are to be executed in accordance with their respective test specifications.

7.3 ISO/IEC 7816 interface

7.3.0 General information

The electrical specification for the ISO/IEC 7816 interface can be found in ETSI TS 102 221 [7]. Test Descriptions related to this interface are defined in ETSI TS 102 230-1 [2] and ETSI TS 102 230-2 [6]. Test requirements for the ISO/IEC 7816 interface reference the related test in the named test specification and shall be executed as defined in there.

7.3.1 Configurations

There are no specific configurations for this topic.

7.3.2 Procedures

There are no specific procedures for this topic.

7.3.3 Test descriptions

7.3.3.1 Electrical specifications of the interface

All Test Descriptions to verify RQ0703_001 found in ETSI TS 102 230-1 [2], clause 5.2 shall apply to the terminal.

7.3.3.2 Contacts

All Test Descriptions to verify RQ0703_002 found in ETSI TS 102 230-1 [2], clause 4 shall apply to the terminal.

7.3.3.3 Initial communication establishment procedures

7.3.3.3.1 SSP interface activation and deactivation

All Test Descriptions to verify RQ0703_003 found in ETSI TS 102 230-1 [2], clause 5.1.2 shall apply to the terminal.

7.3.3.3.2 Supply voltage switching

All Test Descriptions to verify RQ0703_004 found in ETSI TS 102 230-1 [2], clause 5.1.5 shall apply to the terminal.

RQ0703_005 is not testable.

7.3.3.4 Answer to Reset content

All Test Descriptions to verify RQ0703_006 and RQ0703_007 found in ETSI TS 102 230-1 [2], clause 6.1 shall apply to the terminal.

All Test Descriptions to verify RQ0703_008 found in ETSI TS 102 230-1 [2], clauses 6.2, 6.3 and 6.5 shall apply to the terminal.

NOTE: The verification of RQ0703_009 requires a specification for a valid extension of the ATR.

7.3.3.5 PPS procedure

All Test Descriptions to verify RQ0703_010 found in ETSI TS 102 230-1 [2], clause 5.1.5 shall apply to the terminal.

7.3.3.6 Reset procedure

All Test Descriptions for RQ0703_011 found in ETSI TS 102 230-1 [2], clause 5.1.5 shall apply to the terminal.

7.3.3.7 Clock stop mode

All voltage class specific Test Descriptions for RQ0703_012 found in ETSI TS 102 230-1 [2], clauses 6.2 and 6.3 shall apply to the terminal.

7.3.3.8 Bit/Character duration and sampling time

All Test Descriptions for RQ0703_013 found in ETSI TS 102 230-1 [2], clauses 7.1.1 and 7.1.2 shall apply to the terminal.

7.3.3.9 Error handling

No error handling specific Test Descriptions to verify RQ0703_014 are defined in ETSI TS 102 230-1 [2]. To validate correct handling, the power transition tests from clause 5.1 and the 'no ATR' test from clause 5.1.5.6 shall be executed.

7.3.3.10 Data link protocols

All Test Descriptions to verify the protocol timing and handling requirements from RQ0703_015, RQ0703_016 and RQ0703_017 found in ETSI TS 102 230-1 [2], clause 7.3 shall be applied to the terminal.

7.4 SPI Interface

Test Descriptions for the SPI Interface can be found in ETSI TS 103 813 [i.2].

7.5 I2C interface

FFS

7.6 SWP interface

FFS

7.7 USB interface

FFS

7.8 Proprietary interface

OOS

8 Test Descriptions: SSP Common Layer

8.1 Introduction

8.1.1 Requirements implicitly verified

The following requirements identified in clause 5.4.1 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0801_001, RQ0801_002

8.2 SCL network

8.2.1 Requirements implicitly verified

The following requirements identified in clause 5.4.2 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0802_002

8.2.2 Requirements verified elsewhere

The following requirements identified in clause 5.4.2 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0802_001, RQ0802_001a, RQ0802_003

8.3 Protocol layers

8.3.1 Requirements implicitly verified

The following requirements identified in clause 5.4.3 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0803_003

8.3.2 Requirements verified elsewhere

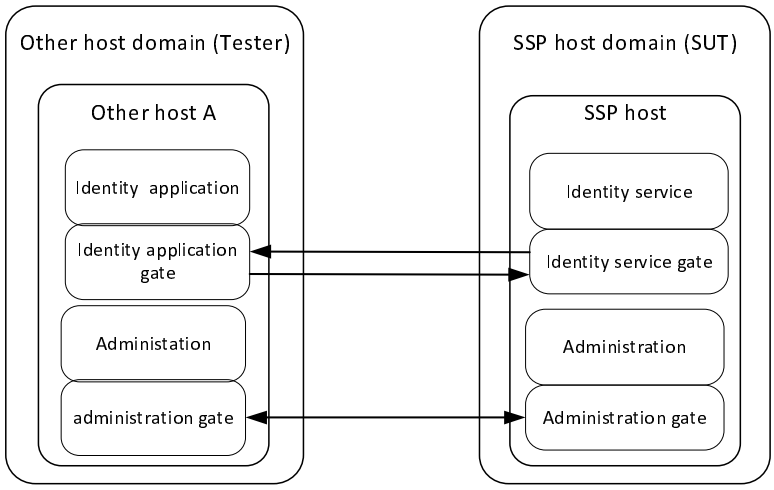
The following requirements identified in clause 5.4.3 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0803_002, RQ0803_004, RQ0803_005, RQ0803_006

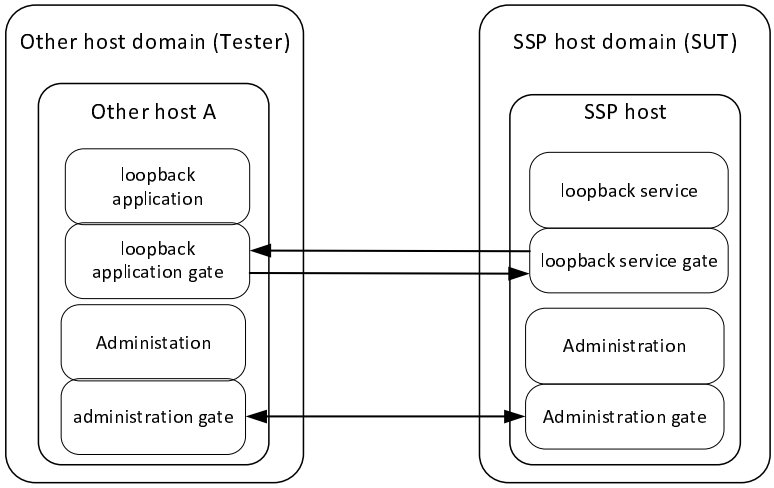
8.4 SCL core services

8.4.1 Configurations

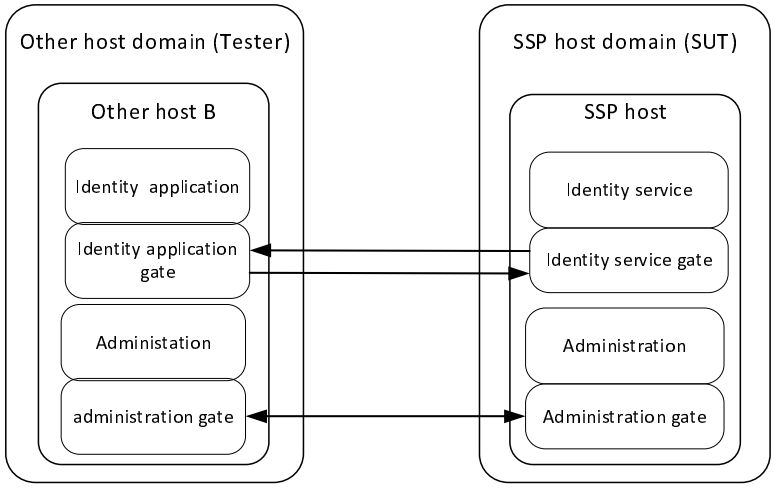
8.4.1.1 CSCL_001 - Identity service -host A

Configuration identification	CSCL_001
Configuration description	

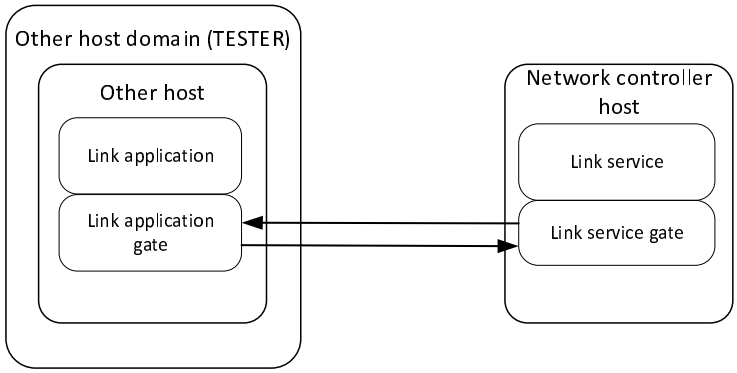
8.4.1.2 CSCL_002 - Loopback service

Configuration identification	CSCL_002
Configuration description	

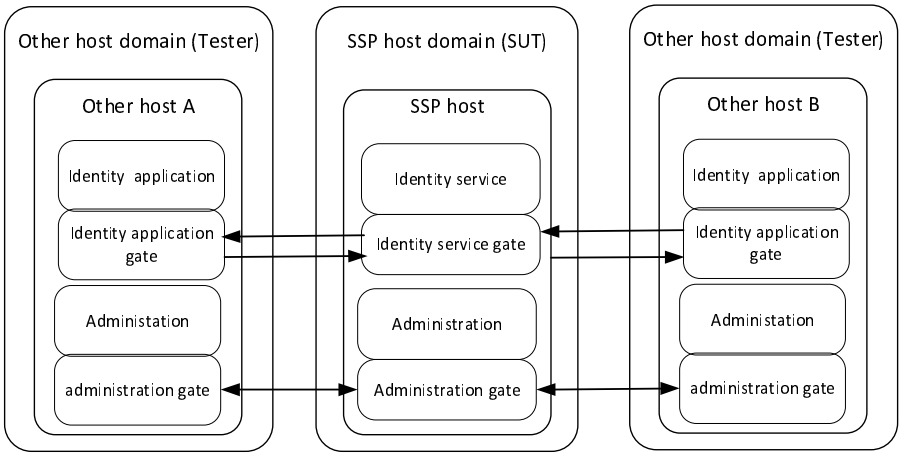
8.4.1.3 CSCL_003 - Identity service-host B

Configuration identification	CSCL_003
Configuration description	

8.4.1.4 CSCL_004 - Network host controller link

Configuration identification	CSCL_004
Configuration description	

8.4.1.5 CSCL_005 - Identity service-with multiple other host

Configuration identification	CSCL_005
Configuration description	

8.4.1.6 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPSCLconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) scl (4)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   URN-Description,
   URN-Description-List
FROM SSPDefinitions;
-- ASN1STOP
```

8.4.2 Procedures

8.4.2.1 PSCL_021 - Pipe session opening on the identity service/application gates

Procedure identification	PSCL_021
Procedure objectives	The other host A or B shall be able to open a pipe session to the identity gate of the SSP host.
Configuration reference	CSCL_001, CSCL_003, CSL_005
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate in other host 'X' sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate in SSP sends EVT_ADM_BIND to Administration gate in the other host 'X' with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).

8.4.2.2 PSCL_022 - Pipe session opening on the loopback service/application gates

Procedure identification	PSCL_022
Procedure objectives	The other host shall be able to open a pipe session to the loopback gate of the SSP host.
Configuration reference	CSCL_002
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate in other host sends EVT_ADM_BIND to Administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the loopback service gate. $GATE_{LOOKBACK}$: The UUID gate identifier of the loopback gate (1CE3D0F5-3B55-5470-B6F1-168352F27440).
2	Administration gate in SSP host sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the loopback application gate. $GATE_{LOOPBACK}$: The UUID gate identifier of the loopback gate (1CE3D0F5-3B55-5470-B6F1-168352F27440).

8.4.2.3 PSCL_023 - Retrieve the content of identity service registry by host A

Procedure identification	PSCL_023
Procedure objectives	The host A shall be able to retrieve the content of a registry.
Configuration reference	CSCL_001
Initial conditions	
The procedure PSCL_021 is successfully executed.	
Procedure sequence	
Step	**Description**
1	Identity application gate in host A sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.
2	Identity service gate in SSP host sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other hostA. The identity service identifier shall be present.

8.4.2.4 PSCL_024 - Retrieve the content of identity service registry by host B

Procedure identification	PSCL_024
Procedure objectives	The host B shall be able to retrieve the content of a registry.
Configuration reference	CSCL_003
Initial conditions	
The procedure PSCL_021 is successfully executed.	
Procedure sequence	
Step	**Description**
1	Identity application gate in host B sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.
2	Identity service gate in SSP host sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host B. The identity service identifier shall be present.

8.4.3 Test descriptions - SCL

8.4.3.1 SCL_031 - Data-flow control in multiple hosts environment

Test identification	SCL_031
Test objectives	The host A and B shall be able to retrieve the content of a registry simultaneously and without impacts on the reliability of the communication. Both hosts shall be able to query asynchronously and repeatedly the SSP identity service. The SSP identity service receives commands from both A and B and the SSP host shall manage globally the flow control for both hosts without using the credit-based data flow control principle available in the administration service.
Configuration reference	CSCL_005
Initial conditions	
The procedure PSCL_021 is successfully executed with other host A. The procedure PSCL_021 is successfully executed with other host B.	

Test sequence		
Step	**Description**	**Requirements**
1	The test sequence (Step 1 and Step 2) in PSCL_023 is executed successfully 100 times. There is no gap of time between receiving the response and sending the next command. Simultaneously the test sequence (Step 1 and Step 2) in PSCL_024 is executed successfully 100 times. There is no gap of time between receiving the response and sending the next command. Credit-based data flow control mechanism and the data acknowledgement mechanism shall not be used in the identity application gate. The test is successful, if no loss of message is detected. I.e.: the Identity service gate in SSP host sends ANY_GET_PARAMETER response to the identity application gates for every ANY_GET_PARAMETER command.	RQ0804_023 RQ0802_003 RQ0802_004

8.4.3.2 SCL_032 - loopback Data-flow control

Test identification	SCL_032	
Test objectives	The other host shall be able to send a continuous flow of EVT_LOOP_POST_DATA without loss of EVT_LOOP_ECHO_DATA events.	
Configuration reference	CSCL_002	
Initial conditions		
The procedure PSCL_022 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The loopback application shall set the counter C to 0.	
2	The loopback application gate sends 100 EVT_LOOP_POST_DATA events to the loopback service gate. Each EVT_LOOP_POST_DATA event contains a counter C as a data (1 byte). The counter C is incremented after the sending of the event.	RQ0803_001
3	The loopback application gate receives 100 EVT_LOOP_ECHO_DATA events. The test is successful if 100 EVT_LOOP_ECHO_DATA events are received and their data contains the counter C sequentially in order.	RQ0803_001

8.4.3.3 SCL_033 - Identity Service Gate parameter GATE_URN_LIST

Test identification	SCL_033	
Test objectives	To test that all UUID-s provided in the GATE_URN_LIST shall be present in the GATE_LIST. The additional registries may be present. The dynamic gate identifier related to the accessor authentication service shall not be present in the GATE_LIST.	
Configuration reference	CSCL_001	
Initial conditions		
The procedure PSCL_021 is successfully executed. The test description AAS_0032 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate sends ANY_GET_PARAMETER command with the register identifier '04' (GATE_LIST) to the Identity service gate.	
2	The Identity service gate returns the GATE_LIST entry as an array of UUID gate identifiers to the Identity application gate. The dynamic gate identifier related to the accessor authentication service shall not be present in the GATE_LIST	RQ0804_022 RQ0804_013
3	Identity application gate sends ANY_GET_PARAMETER command with the register identifier '81' (GATE_URN_LIST) to the Identity service gate.	RQ0804_022
4	The Identity service gate returns GATE_URN_LIST entry with the list of URN-s and UUID-s of the gates to the Identity application gate.	RQ0804_018
5	All UUID-s provided in the GATE_URN_LIST shall be present in the GATE_LIST	RQ0804_020

8.4.3.4 SCL_034 - Link Service Gate additional registry entry

Test identification	SCL_034	
Test objectives	To retrieve the additional registry entry of the Link Service Gate. Checking the minimal value of the SSP_MTU	
Configuration reference	CSCL_004	
Initial conditions		
None		
Test sequence		
Step	**Description**	**Requirements**
1	Link application gate sends ANY_GET_PARAMETER command with the register identifier '05' (SSP_MTU) to the Link service gate.	
2	The Link service gate sends ANY_OK to the Link application gate including the value of SSP_MTU which shall be equal to or greater than 20.	RQ0803_001 RQ0804_004 RQ0804_005 RQ0804_006

8.4.3.5 SCL_035 - Credit based data flow control on administration gate

Test identification	SCL_035	
Test objectives	The administration gate shall not generate EVT_ADM_RECEIVED or EVT_ADM_CREDIT with pipe identifier related to static pipes.	
Configuration reference	CSCL_001	
Initial conditions		
The procedure PSCL_021 is successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity application gate in host A sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.	
2	Identity service gate in SSP host sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other hostA. The identity service identifier shall be present.	
3	Step 1 and 2 is repeated 10 times The administration gate of the SSP host shall not generate EVT_ADM_RECEIVED or EVT_ADM_CREDIT with pipe identifiers related to identity gate.	RQ0804_010 RQ0804_011 RQ0804_012 RQ0804_014

8.4.3.6 End of ASN.1 structure

The annex shall be appended at the end of the SCL test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

8.4.3.7 Requirements not testable, implicitly verified or verified elsewhere

8.4.3.7.1 Requirements implicitly tested

The following requirements identified in 5.4.4 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0804_019

8.4.3.7.2 Requirements verified elsewhere

The following requirements identified in clause 5.4.3 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0804_001, RQ0804_002, RQ0804_003, RQ0804_024, RQ0804_025

RQ0804_007, RQ0804_008, RQ0804_009, RQ0804_021

8.6.3.7.3 Requirements tested in a different clause

The following requirements are tested in a different clause of the present document:

RQ0804_015, RQ0804_016, RQ0804_017, RQ0804_022

8.5 SCL procedures

8.5.1 Requirements verified elsewhere

The following requirements identified in clause 5.4.5 are not tested in accordance with the present document, as they are referencing requirements from a different standardization body:

RQ0805_001, RQ0805_002, RQ0805_005, RQ0805_006, RQ0805_007

8.5.2 Requirements not tested

The following requirements identified in clause 5.4.2 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ0805_003, RQ0805_004

9 Test Descriptions: Secure SCL

9.1 Protocol stack

There are no test descriptions related to clause 9.1 of ETSI TS 103 666-1 [1].

9.2 Secure datagram

There are no test descriptions related to clause 9.2 of ETSI TS 103 666-1 [1]. The requirements related to this clause are tested in SSL_034 below.

9.3 Security protocol

9.3.1 Configurations

9.3.1.1 Referred configurations

The test descriptions refer to the following configurations:

- CAAS_001-Accessor and Identity services.

9.3.1.2 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPSSLconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) secure_scl (5)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   UUID,
   SessionID,
   AccessorRights,
   AAS-CONTROL-SERVICE-GATE-Commands,
   AAS-CONTROL-SERVICE-GATE-Responses,
   Certificate,
   AuthenticationToken,
       AccessorTokenCredential,
   AccessorConditionsPIN,
   AccessorConditions,
   Version,
   VersionType
FROM SSPDefinitions
   SubjectPublicKeyInfo
FROM PKIX1Explicit88
   ECDSA-Sig-Value,
id-ecPublicKey
FROM PKIX1Algorithms88;
eAASVersion VersionType ::='0100' --Version 01.00

brainpool384r1 OBJECT IDENTIFIER ::={ 1 3 36 3 3 2 8 1 1 11}
eEADSASHA256 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 3 2}
-- urn:etsi.org:asn.1:accessor:test:1
eAS-ID-ACC-TEST-1    UUID::='7DFF3B1C6C345A49BC36F1380CEAA0C2'H
-- urn:etsi.org:asn.1:accessor:root
eAS-ID-ACC-ROOT      UUID::='DD61116FF0DD57F48A4F52EE70276F24'H
eAS-ID-AAS-Service      UUID::='DD61116FF0DD57F48A4F52EE70276F24'H
eAS-ID-AAS-GateID    UUID::='AAAAAAAABBBBCCCCDDDDEEEEEEEEEEE'H
eAS-Challenge        UUID::='BA64E9EE888952F4891DA79401758FF4'H

-- The root accessor has all accessor rights

eAS-ACL-ROOT            AccessorRights ::= {
--eAASAccessRight-RequiresSecurePipe--   eRight-Bit1,
--eAASAccessRight-Create AccessorRights--eRight-Bit2,
--eAASAccessRight-Delete--       eRight-Bit3,
--eAASAccessRight-Update AccessorRights--eRight-Bit4,
--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateGroup--     eRight-Bit6,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }
-- The TEST 1 accessor may update its ACL
eAS-ACL-TEST-1          AccessorRights ::= {

--eAASAccessRight-UpdateACL--       eRight-Bit5,
--eAASAccessRight-UpdateCredentialPolicy--  eRight-Bit7,
--eAASAccessRight-UpdateCredentialStatus--  eRight-Bit8
                    }

-- ASN1STOP
```

9.3.1.3 Implicit requirements

There are no implicit requirements.

9.3.1.4 Software tools

Software tools associated with these test descriptions are available in the ETSI forge repository as defined in [i.1]. These tools are provided as examples of how to generate the required data for tests.

The tools enable a tester to generate:

- The AAS certification path (authentic) leading to a correct certification path.
- The AAS certification path (fake) leading to a wrong certification path.
- The AAA certification path (authentic) leading to a correct certification path.
- The AAA certification path (fake) leading to a wrong certification path.
- A valid ATK.AAA.ECKA authentication token duly signed with a verifiable AAA certification path.
- An invalid ATK.AAA.ECKA authentication token duly signed with a verifiable AAA certification path.
- A valid ATK.AAS.ECKA authentication token duly signed with a verifiable AAS certification path.
- An invalid ATK.AAS.ECKA authentication token duly signed with a verifiable AAS certification path.

The generated authentication tokens and certification paths can be combined.

9.3.2 Procedures

9.3.2.1 Referred procedures

The test descriptions refer to the following procedures in clause 6.13.2 above:

- PAAS_021: Open a pipe session with the Identity gate
- PAAS_022: Open a pipe session with the ROOT Accessor Authentication service
- PAAS_023: Open a pipe session with the Anonymous Accessor Authentication service of the Anonymous Accessor
- PAAS_024: Open a pipe session with the TEST-1 Accessor Authentication service

9.3.3 Test descriptions- Security protocol

9.3.3.1 SSL_031 - Shared secret initialization

Test identification	SSL_031	
Test objectives	The root accessor shall be able to be authenticated with the Accessor Authentication service by using: • The aAAS-OP-GET-CHALLENGE-Service-Command command. • The aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command. This authentication is based on authentication tokens.	
Configuration reference	CAAS_001	
Initial conditions		
The procedure PAAS_021 shall be successfully executed. The ROOT accessor is present in the GATE_LIST registry of the identity gate. The procedure PAAS_022 shall be successfully executed. A pipe session is opened with the ROOT Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aSSL-0031-command-01 command to AAS gate with:
	`-- ASN1START`
	`aSSL-0031-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CHALLENGE-Service-Command : {}`
	`-- ASN1STOP`
		RQ0904_001

2	AAS gate sends an aSSL-0031-response-01 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0031-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CHALLENGE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aChallenge eAS-Challenge,`
	`  aCertificates {eAS-CERT-01}}}`
	`-- ASN1STOP`
		RQ0903_001
		RQ0903_002
		RQ0903_003
		RQ0904_002

3	AAA gate sends an aSSL-0031-command-02 command to AAS gate with:
	`-- ASN1START`
	`aSSL-0031-command-02 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {`
	`  aToken eAS-ATK-01, aTokenCertificationPath {eAS-CERT-02}}}`
	`-- ASN1STOP`
		RQ0903_004
		RQ0903_005
		RQ0903_006
		RQ0903_007
		RQ0903_012
		RQ0904_004

4	AAS gate sends an aSSL-0031-response-02 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0031-response-02 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter aServiceToken eAS-ATK-01`
	`}`
	`-- ASN1STOP`
		RQ0903_008
		RQ0903_009
		RQ0903_010
		RQ0903_011
		RQ0903_013

9.3.3.2 SSL_032 - Access to the Authentication Service from the ROOT accessor

Test identification	SSL_032	
Test objectives	The root accessor shall be able to be authenticated with the Accessor Authentication service by using: • The aAAS-OP-ACCESS-SERVICE-Service-Command command. The authentication mean is based on the authentication tokens.	
Configuration reference	CAAS_001	
Initial conditions		
The test SSL_031 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate an aSSL-032-command-01 command to AAS gate with:
	`-- ASN1START`
	`aSSL-0032-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-ACCESS-SERVICE-Service-Command : {`
	`  aServiceIdentifier eAS-ID-AAS-Service,  aUseSecurePipe TRUE}`
	`-- ASN1STOP`	

2	AAS gate an aSSL-0032-response-01 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0032-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-ACCESS-SERVICE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {  aGateIdentifier eAS-ID-AAS-GateID  }}`
	`-- ASN1STOP`
		RQ0903_014
		RQ0903_015
		RQ0903_016

9.3.3.3 SSL_033 - Shared secret initialization (failure)

Test identification	SSL_033	
Test objectives	The root accessor shall be able to be authenticated with the Accessor Authentication service by using: • The aAAS-OP-GET-CHALLENGE-Service-Command command. • The aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command. The authentication mean is based on the authentication tokens. The test is successful if the authentication failed , because…[explain what is used as wrong credential].	
Configuration reference	CAAS_001	
Initial conditions		
The procedure PAAS_021 shall be successfully executed. The ROOT accessor is present in the GATE_LIST registry of the identity gate. The procedure PAAS_022 shall be successfully executed. A pipe session is opened with the ROOT Accessor Authentication Service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aSSL-033-command-01 to AAS gate with:
	`-- ASN1START`
	`aSSL-0033-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CHALLENGE-Service-Command : {}`
	`-- ASN1STOP`
		RQ0903_001

2	AAS gate sends aSSL-033-response-01 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0031-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CHALLENGE-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter {`
	`  aChallenge eAS-Challenge,`
	`  aCertificates {eAS-CERT-01}}}`
	`-- ASN1STOP`
		RQ0903_002
		RQ0903_003

3	AAA gate sends an aSSL-033-command-02 command to AAS gate with:
	`-- ASN1START`
	`aSSL-0033-command-02  AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Command : {`
	`  aCredential aAccessorTokenCredential : {`
	`  aToken eAS-ATK-01, aTokenCertificationPath {eAS-CERT-01}}}`
	`-- ASN1STOP`
		RQ0903_004
		RQ0903_005
		RQ0903_006
		RQ0903_007
		RQ0903_012

4	AAS gate sends an aSSL-033-response-02 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0033-response-02 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-AUTHENTICATE-ACCESSOR-Service-Response : {`
	`  aAAS-Service-Response eAAS-E-NOK }`
	`-- ASN1STOP`
		RQ0903_008
		RQ0903_009
		RQ0903_010
		RQ0903_011
		RQ0903_013

9.3.3.4 SSL_034 - Capability of an accessor (secure SCL usage)

Test identification	SSL_034
Test objectives	The Accessor Authentication application shall send an aAAS-OP-GET-CAPABILITIES-Service-Command by using the secure SCL. The purpose of this test is to initiate a transmission using the secure SCL. AAS and AAS gate shall verify the following requirements related to the secure SCL: • The diversification of the IV and K according to the DIVERSIFIER as defined in clauses 9.2 and C.4 of ETSI TS 103 666-1 [1]. • The secure message fragment length is a multiple of 16 bytes. • The ICHECK is compliant with the ANSI X9.63 [30]. • The CGM counter is initiated at the opening of the pipe session.
Configuration reference	CSSL_003

Initial conditions

The following tests shall be successfully executed in order:
- The AAS_312 requesting to access the Accessor Authentication Service by using a secure SCL.
- The AAS_313 opening a pipe session to an Accessor Authentication Service gate by using a secure SCL.

Test sequence

Step	Description	Requirements

1	AAA gate sends an aSSL-034-command-01 command to AAS gate with:
	`-- ASN1START`
	`aSSL-0034-command-01 AAS-CONTROL-SERVICE-GATE-Commands ::= aAAS-OP-GET-CAPABILITIES-Service-Command : {`
	`  aRequestType eGlobalAuthenticationService}`
	`-- ASN1STOP`
		RQ0902_001
		RQ0902_002
		RQ0902_003
		RQ0902_004
		RQ0902_005
		RQ0902_006
		RQ0902_007
		RQ0902_008
		RQ0902_009
		RQ0902_010

2	AAS gate sends an aSSL-034-response-01 response to AAA gate with:
	`-- ASN1START`
	`aSSL-0034-response-01 AAS-CONTROL-SERVICE-GATE-Responses ::= aAAS-OP-GET-CAPABILITIES-Service-Response : {`
	`  aAAS-Service-Response eAAS-OK,`
	`  aParameter aGlobalAuthenticationService : {`
	`  aAASVersion eAASVersion,`
	`  aAccessorList {`
	`  aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorConditions {aAccessConditionsTokens eTokenCertificate},`
	`  aACL {`
	`  {`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  }}},`
	`  aAccessorUser : {`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorConditions {aAccessConditionsPIN ePinNumeric},`
	`  aACL {{`
	`  aAccessorIdentity eAS-ID-ACC-TEST-1,`
	`  aAccessorRights eAS-ACL-TEST-1`
	`  },{`
	`  aAccessorIdentity eAS-ID-ACC-ROOT,`
	`  aAccessorRights eAS-ACL-ROOT`
	`  }}}}}}`
	`-- ASN1STOP`
		RQ0902_001
		RQ0902_002
		RQ0902_003
		RQ0902_004
		RQ0902_005
		RQ0902_006
		RQ0902_007
		RQ0902_008
		RQ0902_009
		RQ0902_010

9.3.4 Annexes - Accessor Authentication ASN.1 description

9.3.4.1 Annex - Certificates and Tokens

9.3.4.1.0 Certificates and Tokens

The Authentication token and certificates.

```
-- ASN1START
eAS-ATK-01 AuthenticationToken::={
      tbsToken {
        version v1,
        subjectPublicKeyInfo {
          algorithm {
            algorithm { 0 0 }
          },
          subjectPublicKey '0'B
        },
        aATK-Content {
          aChallenge '00000000000000000000000000000000'H,
          aKey-Size e128,
          aStreamCipherIdentifier aAES-CGM-StreamCipherIdentifier
        }
      },
      signatureAlgorithm {
        algorithm eEADSASHA256
      },
      signature {
        r 0,
        s 0
      }
    }
eAS-CERT-01 Certificate ::= {
   tbsCertificate {
     version v3,
     serialNumber 1,
     signature {
      algorithm { 0 0 },
      parameters OCTET STRING : '00'H
     },
     issuer rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     validity {
      notBefore utcTime : "000101000000Z",
      notAfter utcTime : "000101000000Z"
     },
     subject rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     subjectPublicKeyInfo {
      algorithm {
        algorithm brainpool384r1
      },
      subjectPublicKey '0'B
     },
     issuerUniqueID '0'B,
     subjectUniqueID '0'B,
     extensions {
      {
        extnID { 0 0 },
        critical FALSE,
        extnValue '00'H
      }
     }
```

```
   },
   signatureAlgorithm {
     algorithm eEADSASHA256

   },
   signature '0'B
  }
eAS-CERT-02 Certificate ::= {
   tbsCertificate {
     version v3,
     serialNumber 1,
     signature {
      algorithm { 0 0 },
      parameters OCTET STRING : '00'H
     },
     issuer rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     validity {
      notBefore utcTime : "000101000000Z",
      notAfter utcTime : "000101000000Z"
     },
     subject rdnSequence : {
      {
        {
         type { 0 0 },
         value OCTET STRING : '00'H
        }
      }
     },
     subjectPublicKeyInfo {
      algorithm {
        algorithm brainpool384r1
      },
      subjectPublicKey '0'B
     },
     issuerUniqueID '0'B,
     subjectUniqueID '0'B,
     extensions {
      {
        extnID { 0 0 },
        critical FALSE,
        extnValue '00'H
      }
     }
   },
   signatureAlgorithm {
     algorithm { 0 0 },
     parameters OCTET STRING : '00'H
   },
   signature '0'B
  }

-- ASN1STOP
```

9.3.4.1.1 Annex - Certificates with valid certification path

The certificates and the private keys are published for test only and shall not be used for an operational authentication. Only the private keys are published. The public keys are deduced from the private keys and the certificate can be generated by using the tooling available on the ETSI repository.

ETSI-SSP-CI-private-key

```
Private-Key: (384 bit)
priv:
    8a:98:d5:15:cc:00:c7:0a:85:50:29:6c:86:8d:52:
    da:88:fc:8e:b8:5b:56:36:17:2b:b0:65:1c:ca:de:
    f9:b0:88:92:75:73:ff:81:62:5e:f7:1c:2b:12:f9:
    48:59:97
```

```
pub:
    04:45:df:a0:b2:68:bb:0c:0b:68:b9:10:d9:18:f8:
    fa:55:3a:6b:e6:d1:d2:f4:cd:02:a9:2f:3e:43:e9:
    7d:ae:26:b7:ab:ef:e9:60:36:c5:4d:ad:7f:0a:e4:
    70:13:87:bd:07:84:65:8c:3c:0d:cb:e5:aa:b6:cf:
    21:ca:a2:3d:72:0f:ec:4d:ba:bb:9b:71:4d:e4:f0:
    7c:90:ec:84:51:e1:50:28:6a:c6:d5:81:ad:1e:e1:
    8b:04:51:2f:29:b9:74
ASN1 OID: brainpoolP384r1
-----BEGIN EC PRIVATE KEY-----
MIGoAgEBBDCKmNUVzADHCoVQKWyGjVLaiPyOuFtWNhcrsGUcyt75sIiSdXP/gWJe
9xwrEvlIWZegCwYJKyQDAwIIAQELoWQDYgAERd+gsmi7DAtouRDZGPj6VTpr5tHS
9M0CqS8+Q+l9ria3q+/pYDbFTa1/CuRwE4e9B4RljDwNy+Wqts8hyqI9cg/sTbq7
m3FN5PB8kOyEUeFQKGrG1YGtHuGLBFEvKbl0
-----END EC PRIVATE KEY-----
```

ETSI-SSP-AAA-CA-private-key

```
Private-Key: (384 bit)
priv:
    3c:c0:ae:01:5d:39:99:4c:2c:a9:42:b0:b7:4f:64:
    29:d5:ef:1f:87:39:f1:c5:98:f4:80:b0:a5:ec:58:
    9b:dc:eb:36:0d:c8:a7:f6:de:e2:8f:d0:79:9d:47:
    35:6d:79
pub:
    04:19:ab:77:c8:78:2e:f4:9f:98:af:3c:23:42:88:
    73:00:51:cc:b6:3a:49:da:e2:90:2b:e8:9c:44:83:
    49:bb:67:96:48:4d:61:04:86:57:c6:c0:52:c3:38:
    bf:c3:d4:1e:5f:a8:80:52:a7:60:25:cd:63:4d:79:
    37:a6:bd:6c:1d:ca:dc:bb:33:0d:85:6f:3c:18:8c:
    27:2a:23:1a:eb:e0:12:f3:14:ff:ac:d7:22:96:41:
    7e:9d:bc:ed:fb:6a:0a
ASN1 OID: brainpoolP384r1
-----BEGIN EC PRIVATE KEY-----
MIGoAgEBBDA8wK4BXTmZTCypQrC3T2Qp1e8fhznxxZj0gLCl7Fib3Os2Dcin9t7i
j9B5nUc1bXmgCwYJKyQDAwIIAQELoWQDYgAEGat3yHgu9J+YrzwjQohzAFHMtjpJ
2uKQK+icRINJu2eWSE1hBIZXxsBSwzi/w9QeX6iAUqdgJc1jTXk3pr1sHcrcuzMN
hW88GIwnKiMa6+AS8xT/rNcilkF+nbzt+2oK
-----END EC PRIVATE KEY-----
```

ETSI-SSP-AAA-EE-private-key

```
Private-Key: (384 bit)
priv:
    66:21:47:d6:4a:6e:75:8d:5c:e4:03:57:7f:6a:cc:
    ea:12:9b:0c:c8:33:fd:d6:df:68:af:95:97:47:96:
    f2:d4:ef:c9:f1:df:cc:1a:f9:87:2d:c3:9b:80:14:
    89:86:97
pub:
    04:59:22:a0:b6:17:04:d2:1a:8c:5a:27:58:23:00:
    bb:77:26:eb:49:ad:2b:dd:85:5f:cb:34:92:18:8c:
    df:27:29:aa:34:ed:ae:58:db:6d:93:8c:6d:27:83:
    7d:cb:d7:87:84:f2:0a:2d:47:9b:17:6d:dc:ed:68:
    a3:db:76:47:f2:9c:5a:ae:20:97:24:27:fc:00:2c:
    b8:a7:a6:f2:52:82:60:f8:fb:3d:a2:75:5c:03:22:
    8d:ee:05:fd:66:6f:8a
ASN1 OID: brainpoolP384r1
-----BEGIN EC PRIVATE KEY-----
MIGoAgEBBDBmIUfWSm51jVzkA1d/aszqEpsMyDP91t9or5WXR5by1O/J8d/MGvmH
LcObgBSJhpegCwYJKyQDAwIIAQELoWQDYgAEWSKgthcE0hqMWidYIwC7dybrSa0r
3YVfyzSSGIzfJymqNO2uWNttk4xtJ4N9y9eHhPIKLUebF23c7Wij23ZH8pxariCX
JCf8ACy4p6byUoJg+Ps9onVcAyKN7gX9Zm+K
-----END EC PRIVATE KEY-----
```

ETSI-SSP-AAS-CA-private-key

```
Private-Key: (384 bit)
priv:
    31:1b:e2:b7:0f:d0:fe:81:bc:1f:8c:c9:5e:0a:ff:
    78:fc:88:26:b4:07:ae:c9:d1:94:51:df:32:2d:24:
    16:66:d5:a8:ad:2a:4a:49:29:95:48:2f:f2:e5:d7:
    76:ab:db
```

```
pub:
    04:8b:f7:79:02:f4:5f:a0:9a:9f:79:a7:5c:2f:ef:
    db:e7:e9:0c:f0:03:01:d0:9f:d8:5b:b3:06:be:3b:
    96:62:38:94:95:71:58:95:1c:27:74:c0:92:1c:9e:
    91:62:56:78:52:0b:5f:50:34:65:36:24:1f:03:b9:
    27:76:b8:52:22:fc:c9:97:e2:96:f9:e5:5a:58:05:
    f4:79:0c:33:48:b7:71:80:db:29:38:4c:72:42:44:
    a5:14:09:86:b3:04:4f
ASN1 OID: brainpoolP384r1
-----BEGIN EC PRIVATE KEY-----
MIGoAgEBBDAxG+K3D9D+gbwfjMleCv94/IgmtAeuydGUUd8yLSQWZtWorSpKSSmV
SC/y5dd2q9ugCwYJKyQDAwIIAQELoWQDYgAEi/d5AvRfoJqfeadcL+/b5+kM8AMB
0J/YW7MGvjuWYjiUlXFYlRwndMCSHJ6RYlZ4UgtfUDRlNiQfA7kndrhSIvzJl+KW
+eVaWAX0eQwzSLdxgNspOExyQkSlFAmGswRP
-----END EC PRIVATE KEY-----
ETSI-SSP-AAS-EE-private-key
Private-Key: (384 bit)
priv:
    39:06:12:a8:b3:a4:78:ac:29:15:d3:3e:30:6b:46:
    da:fe:c3:0b:ff:e1:bd:75:72:39:c3:6c:2a:6f:dd:
    01:87:76:7d:c3:37:54:7b:83:13:f9:13:b0:43:7d:
    3f:cc:cb
pub:
    04:60:90:64:71:ca:09:0f:a7:3d:ec:60:fa:8f:d8:
    9d:6b:c6:72:f9:93:33:a5:e4:02:d9:e5:19:d3:ee:
    02:4e:c5:b4:da:a4:97:c0:66:02:31:01:54:13:75:
    8c:14:3e:12:1c:c1:92:e7:8f:f8:c5:51:71:6d:30:
    9c:c9:52:0d:26:9f:02:c0:bb:12:87:47:40:6c:b4:
    54:33:7b:a7:27:3b:87:41:91:67:cd:60:bd:37:b6:
    ac:41:97:4a:8a:4b:54
ASN1 OID: brainpoolP384r1
-----BEGIN EC PRIVATE KEY-----
MIGoAgEBBDA5BhKos6R4rCkV0z4wa0ba/sML/+G9dXI5w2wqb90Bh3Z9wzdUe4MT
+ROwQ30/zMugCwYJKyQDAwIIAQELoWQDYgAEYJBkccoJD6c97GD6j9ida8Zy+ZMz
peQC2eUZ0+4CTsW02qSXwGYCMQFUE3WMFD4SHMGS54/4xVFxbTCcyVINJp8CwLsS
h0dAbLRUM3unJzuHQZFnzWC9N7asQZdKiktU
-----END EC PRIVATE KEY-----
```

9.3.4.2 Annex - End of ASN.1 structure

The annex shall be appended at the end of the accessor authentication test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

9.4 Accessor authentication service procedure

9.4.1 Requirements implicitly verified

The following requirements identified in clause 5.5.4 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ0904_001, RQ0904_002, RQ0904_003, RQ0904_004.

10 Test Descriptions: Communication layers above SCL

10.1 Overview

There are no requirements for test descriptions related to clause 10.1 of ETSI TS 103 666-1 [1].

10.2 APDU protocol

10.2.1 Introduction

There are no requirements for test descriptions related to clause 10.2.1 of ETSI TS 103 666-1 [1].

10.2.2 Command-response pairs

10.2.2.1 General definition

There are no test descriptions related to clause 10.2.2.1 of ETSI TS 103 666-1 [1].

Requirement RQ1002_001 is unspecific and will be tested implicitly with SSP command related tests if the UICC File System Service is supported.

10.2.2.2 CLA byte

There are no test descriptions related to clause 10.2.2.2 of ETSI TS 103 666-1 [1].

Requirement RQ1002_002 is unspecific. Class byte handling will be tested implicitly with SSP command related tests if the UICC File System Service is supported.

10.2.2.3 INS byte

There are no test descriptions related to clause 10.2.2.3 of ETSI TS 103 666-1 [1].

Requirement RQ1002_003 is implicitly tested with SSP command related tests if the UICC File System Service is supported.

Requirement RQ1002_004 is unspecific. As no specific error handling is defined in ETSI TS 102 221 [7], the requirement cannot be tested.

10.2.2.4 Status Word SW1 SW2

There are no test descriptions related to clause 10.2.2.4 of ETSI TS 103 666-1 [1].

The requirements RQ1002_005and RQ1002_006 are implicitly tested with SSP commands if the UICC File System Service is supported.

10.2.3 SSP commands

10.2.3.0 Applicability of SSP commands

The tester shall successfully execute test descriptions as defined in ETSI TS 102 230-2 [6] related to the defined SSP commands if the UICC File System Service is supported.

NOTE: Tests from ETSI TS 102 230-2 [6] identified as applicable for this clause are referenced as SSP_REF.

10.2.3.1 Overview

There are no requirements for test descriptions related to clause 10.2.3.1 of ETSI TS 103 666-1 [1].

10.2.3.2 EXCHANGE CAPABILITIES

10.2.3.2.0 Applicability of the EXCHANGE CAPABILITIES command

As the handling of the EXCHANGE CAPABILITIES command defined in clause 10.2.3.2 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 230-1 [2] or ETSI TS 102 230-2 [6]. EXCHANGE CAPABILITIES command related tests are FFS.

RQ1002_007, RQ1002_008, RQ1002_009, RQ1002_010 and RQ1002_011 cannot be tested currently.

10.2.3.3 SELECT

10.2.3.3.0 Applicability of the SELECT command

The provisions of ETSI TS 102 221 [7] shall apply. As there currently are no specific test descriptions for a SELECT command with P1 = "04' as defined in clause 10.2.3.3 of ETSI TS 103 666-1 [1] and its sub-clauses, the "select by DF name" command related test in accordance with ETSI TS 102 230-2 [6], clause 6.6.6.1.1 is to be used.

RQ1002_012 is implicitly tested when executing tests from ETSI TS 102 230-2 [6], clause 6.6.6.1.1.

10.2.4 Logical channels

10.2.4.0 Applicability of logical channel related commands

The provisions of ETSI TS 102 221 [7], clause 11.1.17 shall apply. The tester shall successfully execute test descriptions as defined in ETSI TS 102 230-2 [6] related to logical channels if the UICC File System Service is supported.

NOTE: Tests from ETSI TS 102 230-2 [6] identified as applicable for this clause are referenced as LCH_REF.

10.2.4.1 Overview

There are no test descriptions explicitly related to clause 10.2.4.1 of ETSI TS 103 666-1 [1].

RQ1002_013, RQ1002_014 and RQ1002_015 are implicitly tested with the Logical Channel tests defined in ETSI TS 102 230-2 [6], clause 6.6.8.

10.2.4.2 MANAGE CHANNEL

The provisions of ETSI TS 102 221 [7], clause 11.1.17 shall apply. Logical channel tests using the MANAGE CHANNEL command as defined in ETSI TS 102 230-2 [6], clause 6.6.8 have to be used.

RQ1002_016 and RQ1002_17 are implicitly tested with the logical channel tests defined in ETSI TS 102 230-2 [6], clause 6.6.8.

10.2.5 UICC file system commands

10.2.5.0 Applicability of UICC file system commands

The provisions of ETSI TS 102 221 [7], clause 8.4 shall apply. The tester shall successfully execute test descriptions as defined in ETSI TS 102 230-2 [6] related to UICC file system commands if the UICC File System Service is supported.

NOTE: Test from ETSI TS 102 230-2 [6] identified as applicable for this clause are referenced as UFS_REF.

10.2.5.1 Overview

There are no test descriptions explicitly related to clause 10.2.5.1 of ETSI TS 103 666-1 [1].

10.2.5.2 Methods for selecting a file

The provisions of ETSI TS 102 221 [7], clause 8.4 shall apply. Methods for selecting a file tests as defined in ETSI TS 102 230-2 [6], clause 6.6.5 have to be used.

Requirement RQ1002_018 is unspecific, but implicitly tested when executing tests from ETSI TS 102 230-2 [6], clause 6.6.5.

10.2.5.3 Reservation of file IDs

The provisions of ETSI TS 102 221 [7], clause 8.6 shall apply. Methods for reservation of file IDs tests as defined in ETSI TS 102 230-2 [6], clause 6.6.7 have to be used.

Requirement RQ1002_019 is unspecific, but implicitly tested when executing tests from ETSI TS 102 230-2 [6], clause 6.6.7.

10.2.5.4 Security features

The provisions of ETSI TS 102 221 [7], clause 9 shall apply. Methods for security feature tests as defined in ETSI TS 102 230-2 [6], clause 6.7 have to be used.

Requirement RQ1002_020 is unspecific, but implicitly tested when executing tests from ETSI TS 102 230-2 [6], clause 6.7.

10.2.5.5 Additional commands

The specific command related provisions of ETSI TS 102 221 [7], clause 11.1 and clause 11.3 shall apply. Methods for testing the identified additional commands as defined in ETSI TS 102 230-2 [6], clause 6.9.1 and clause 6.9.2 have to be used.

Requirement RQ1002_021 is unspecific, but is implicitly tested when executing tests from clauses 6.9.1.1, 6.9.1.3, 6.9.1.4, 6.9.1.5, 6.9.1.6, 6.9.1.7, 6.9.1.8, 6.9.1.9, 6.9.1.10, 6.9.1.11, 6.9.1.12, 6.9.1.13, 6.9.1.14, 6.9.1.15, 6.9.2.1 and 6.9.2.2 of ETSI TS 102 230-2 [6].

10.2.6 Card Application Toolkit

10.2.6.0 Applicability of Card Application Toolkit services

The provisions of ETSI TS 102 221 [7], clause 7.4.2 shall apply if the SSP indicates the support of Card Application Toolkit according to ETSI TS 102 223 [9], The tester shall successfully execute test descriptions as defined in ETSI TS 102 230-2 [6] related to UICC file system commands if the UICC File System Service is supported.

10.2.6.1 Overview

There are no test descriptions explicitly related to clause 10.2.6.1 of ETSI TS 103 666-1 [1].

As the handling of Card Application Toolkit commands defined in clause 10.2.6.1 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 230-1 [2] or ETSI TS 102 230-2 [6]. Card Application Toolkit command related tests are FFS.

RQ1002_022, RQ1002_023, RQ1002_024 and RQ1002_025 cannot be tested currently.

10.2.6.2 Terminal profile

There are no test descriptions explicitly related to clause 10.2.6.2 of ETSI TS 103 666-1 [1].

As the handling of the Terminal profile as defined in clause 10.2.6.2 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 230-1 [2] or ETSI TS 102 230-2 [6] Terminal profile handling related tests are FFS.

RQ1002_026, RQ1002_027, and RQ1002_028 cannot be tested currently.

10.2.6.3 Proactive polling

There are no test descriptions explicitly related to clause 10.2.6.3 of ETSI TS 103 666-1 [1].

As the handling of the Proactive polling as defined in clause 10.2.6.3 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 230-1 [2] or ETSI TS 102 230-2 [6] Proactive polling related tests are FFS.

RQ1002_029, RQ1002_030, and RQ1002_031 cannot be tested currently.

10.2.6.4 Additional commands

There are no test descriptions explicitly related to clause 10.2.6.4 of ETSI TS 103 666-1 [1].

The specific command related provisions of ETSI TS 102 221 [7], clause 11.1 and clause 11.2 shall apply. Methods for testing the identified additional commands as defined in ETSI TS 102 230-2 [6], clause 6.9.1 and clause 6.9.2 have to be used.

NOTE: Tests from ETSI TS 102 230-2 [6] identified as applicable for this clause are referenced as ADD_REF.

Requirement RQ1002_032 can be implicitly tested for the STATUS command when executing tests from clause 6.9.1.2. For the commands ENVELOPE, FETCH and TERMINAL RESPONSE no equivalent test description is available in ETSI TS 102 230-1 [2] or ETSI TS 102 230-2 [6]. Related tests are FFS.

10.2.7 SSP suspension

10.2.7.0 Applicability of SSP suspension

If suspension is supported, the additional commands defined in Table 10.6 of ETSI TS 103 666-1 [1] shall be supported by the SSP.

As the SSP suspension as defined in clause 10.2.7 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 230-2 [6] SSP suspension related tests are FFS.

RQ1002_033 can currently not be tested.

10.2.8 APDU transfer over SCL

10.2.8.0 Applicability of APDU transfer over SCL

The provisions of ETSI TS 102 622 [5] shall apply if the SSP indicates the support of APDU transfer over SCL.

10.2.8.1 Overview

There are no test descriptions explicitly related to clause 10.2.8.1 of ETSI TS 103 666-1 [1].

10.2.8.2 UICC APDU gate

10.2.8.2.0 Test Descriptions for the UICC APDU gate

There are no test descriptions explicitly related to clause 10.2.8.2 of ETSI TS 103 666-1 [1].

10.2.8.2.1 UICC APDU overview

The APDU transport over SCL shall use the HCP message structure as defined in ETSI TS 102 622 [5].

APDU gate related tests to verify RQ1002_034 and RQ1002_035 are FFS.

Requirements RQ1002_036 (partly) and RQ1002_037 032 can be implicitly tested when executing tests from clause 5.9 of ETSI TS 102 695-1 [3].

10.2.8.2.2 UICC APDU service gate

There are no test descriptions related to the UICC APDU service gate URN. Tests to verify RQ1002_038 are FFS.

Requirement RQ1002_039 related to the support of events and registry can be implicitly tested when executing tests from clauses 5.9.1.3 and 5.9.1.2 of ETSI TS 102 695-1 [3].

NOTE: Tests from ETSI TS 102 695-1 [3] identified as applicable for this clause are referenced as APDU_REF1.

10.2.8.2.3 UICC APDU application gate

10.2.8.2.3.0 Test Descriptions for the UICC APDU application gate

There are no test descriptions explicitly related to clause 10.2.8.2.3 of ETSI TS 103 666-1 [1].

10.2.8.2.3.1 Commands

There are no test descriptions related to the UICC APDU application gate supported commands. Tests to verify RQ1002_040 are FFS.

10.2.8.2.3.2 Events

Requirement RQ1002_041 related to the support of events defined in ETSI TS 102 622 [5] can be implicitly tested when executing tests from clause 5.9.2.3 of ETSI TS 102 695-1 [3].

NOTE: Test from ETSI TS 102 695-1 [3] identified as applicable for this clause are referenced as APDU_REF2.

10.2.8.2.3.3 EVT_TOOLKIT_REQUEST

As the EVT_TOOLKIT_REQUEST as defined in clause 10.2.8.2.3.3 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 622 [5] EVT_TOOLKIT_REQUEST related tests are FFS.

Tests to verify RQ1002_042 and RQ1002_043 are FFS.

10.2.8.2.4 State diagram for the UICC APDU gate

As the state diagram for the UICC APDU gate as defined in clause 10.2.8.2.4 of ETSI TS 103 666-1 [1] has no equivalent in ETSI TS 102 622 [5] the related tests are FFS.

Tests to verify RQ1002_044 are FFS.

10.3 File system protocol

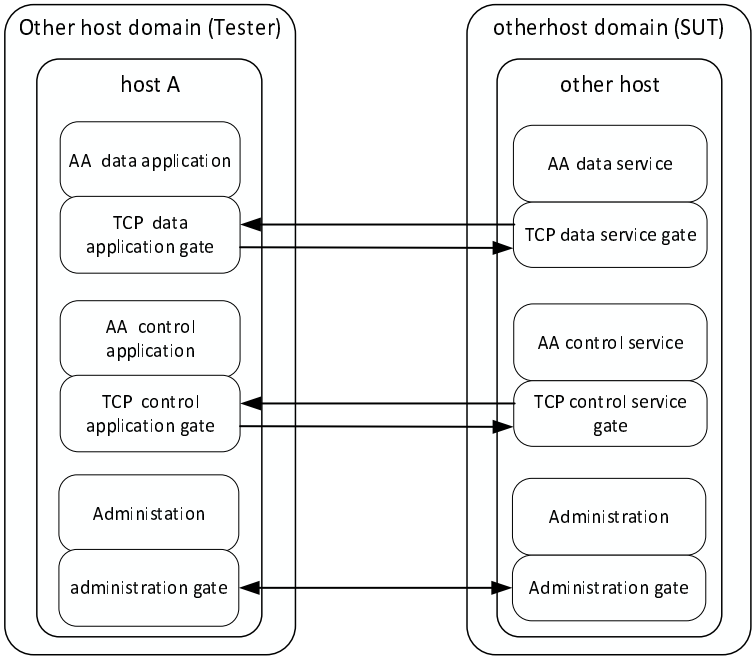
10.3.1 Tests referred to elsewhere

The test descriptions related to clause 10.3 of ETSI TS 103 666-1 [1] can be found in clause 6.6 above.

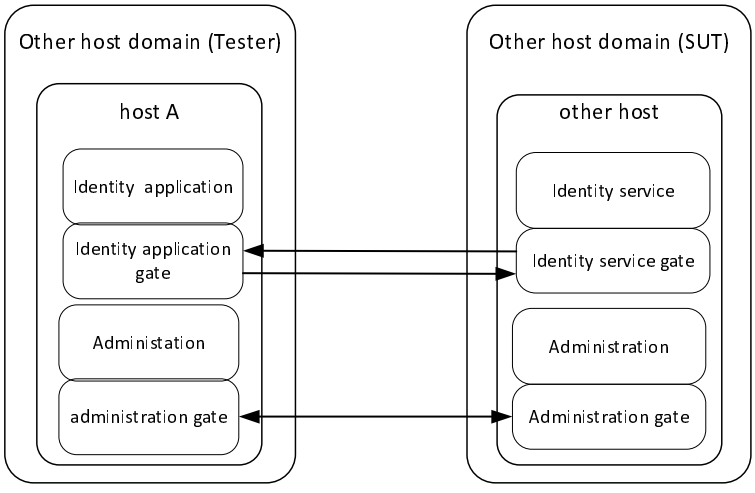
10.4 Transmission Control Protocol support

10.4.1 Configurations

10.4.1.1 CTCP_001-Generic TCP control service

Configuration identification	CTCP_001
Configuration description	

10.4.1.2 CTCP_002-Identity service

Configuration identification	CTCP_002
Configuration description	

10.4.1.3 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
TCPconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) tcp (6)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   UUID,
   Version,
   IPV4Addr,
   IPV6Addr,
FQDN,
TCP-Control-Application-Response,
TCP-CONTROL-SERVICE-GATE-Commands,
TCP-CONTROL-SERVICE-GATE-Responses,
TCP-CONTROL-APPLICATION-GATE-Commands,
TCP-CONTROL-APPLICATION-GATE-Responses,
TCP-CONTROL-APPLICATION-GATE-Events,
   VersionType
FROM SSPDefinitions;

eTCPVersion VersionType ::='0100' --Version 01.00
eGateID-test UUID   ::='00000000000000000000000000000000'H
eTimeout-passive  INTEGER ::= 0
eNetworkAccessName-test OCTET STRING ::='00'H
eUserLogin-test OCTET STRING ::='00'H
eUserPassword-test OCTET STRING ::='00'H
ePortNumber-test INTEGER ::= 17430 --'4416'H smart TLS for smart card
ePortNumber-test-1 INTEGER ::= 65535 --'FFFF'H The host is reachable but the port is not accessible
eIP-test    IPV4Addr::='00000000'H -- IPV4 address of the reachable remote server for test
eIP-test-1     FQDN::="etsi.eu" -- IPV4 FQDN of unreachable remote server for test
eIP-test-2     FQDN::= "etsi.org" -- FQDN address of reachable remote server for test
eIP-test-3     IPV4Addr::= '00000000'H -- IPV4 address of reachable remote server for test
eIP-test-4     IPV6Addr::='00000000000000000000000000000000'H -- IPV6 address of the reachable
remote server for test



-- ASN1STOP
```

10.4.2 Procedures

10.4.2.1 PTCP_021 - Open a pipe session with the Identity gate

Procedure identification	PTCP_021
Procedure objectives	The host A shall be able to open a pipe session to the identity service gate of the other host in the SUT. From the GATE_LIST registry, the UUID of the TCP control service shall be listed. The SUT may be one of the following hosts: • REE host • TEE host • MBM host If the test is successful then a pipe session is open between the identity application and the identity service in the other hosts (SUT).
Configuration reference	CTCP_002

Initial conditions
The tester runs the identity application gate and the SUT runs the identity service gate.

Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the other with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the other host with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The TCP control service identifier shall be present. The test is successful if the previous requirement is satisfied.
5	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the other host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed. This step is required to clean up the context of the tests but it is not essential for the test objective.

10.4.2.2 PTCP_022 - Open a pipe session with the TCP control service in the REE Host domain

Procedure identification	PTCP_022
Procedure objectives	The TCP control application gate shall be able to open a pipe session to the TCP control service gate of the REE host. If the test is successful, then a pipe session is open between the TCP control application in the host A and the TCP control service in the REE host.
Configuration reference	CTCP_001
Initial conditions	
The procedure PTCP_021 shall be successfully executed: • The TCP control service identifier is present in the GATE_LIST registry of the identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the host A with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the TCP control service gate. • $GATE_{TCP}$: The UUID gate identifier of the TCP control service gate (F3DBA7CC-3551-5170-BC79-8BED75TCP control application37TCP control application).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the TCP control application gate. • $GATE_{URN}$: The UUID gate identifier of the TCP control service gate (F3DBA7CC-3551-5170-BC79-8BED75TCP control application37TCP control application). $GATE_{TCP}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.

10.4.2.3 PTCP_023 - Open a pipe session with the TCP control service in the TEE Host domain

Procedure identification	PTCP_023
Procedure objectives	The TCP control application gate shall be able to open a pipe session to the TCP control service gate of the TEE host. If the test is successful then a pipe session is open between the TCP control application in the other host and the TCP control service in the TEE host.
Configuration reference	CTCP_001
Initial conditions	
The procedure PTCP_021 shall be successfully executed: • The TCP control service identifier is present in the GATE_LIST registry of the Identity service gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the other with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the TCP control service gate. • $GATE_{TCP}$: The UUID gate identifier of the TCP control service gate (727A3D1D-B52D-50CB-B20B-BCA7E9EE25CF).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the TCP control application gate. • $GATE_{TCP}$: The UUID gate identifier of the TCP control service gate (727A3D1D-B52D-50CB-B20B-BCA7E9EE25CF). $GATE_{TCP}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.

10.4.2.4 PTCP_024 - Open a pipe session with the TCP control service in the MBM Host domain+

Procedure identification	PTCP_024	
Procedure objectives	The host A shall be able to open a pipe session to the TCP control service gate of the MBM host. The TCP control service identifier is 'ADCE4843-A058-50F2-A98D-5D3C334504B0'H. If the test is successful, then a pipe session is open between the TCP control application in the host A and the TCP control service in the MBM host.	
Configuration reference	CTCP_001	
Initial conditions		
The procedure PTCP_021 shall be successfully executed: • The TCP control service identifier is present in the GATE_LIST registry of the Identity service gate of the MBM host.		
Procedure sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the other with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the TCP control service gate. • $GATE_{TCP}$: The UUID gate identifier of the TCP control service gate (8EC8017B-B734-533D-TCP control applicationA0-FF6D693EA85C).	
2	Administration gate sends EVT_ADM_BIND to Administration gate in the host A with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the TCP control application gate. • $GATE_{TCP}$: The UUID gate identifier of the TCP control service gate (8EC8017B-B734-533D-TCP control applicationA0-FF6D693EA85C). $GATE_{TCP}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	

10.4.3 Test descriptions

10.4.3.1 TCP Passive Connection opening

10.4.3.1.1 TCP_311 - Request to OPEN TCP Connection

Test identification	TCP_311	
Test objectives	TCP control service gate shall open TCP connection and return response successful.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-311-command-01 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-311-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aIP  eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName eNetworkAccessName-test,`
	`  aUserLogin eUserLogin-test,`
	`  aUserPassword eUserPassword-test`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-311-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-311-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_014
		RQ1004_031
		RQ1004_032
		RQ1004_036
		RQ1004_037
		RQ1004_039
		RQ1004_040
		RQ1004_041
		RQ1004_042
		RQ1004_052

10.4.3.1.2 TCP_312 - Request to OPEN TCP Connection without network parameters

Test identification	TCP_312	
Test objectives	TCP control service gate shall be able to open a TCP connection without network parameters (optional) and return response successful.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-312-command-02 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-312-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends aTCP-312-response-02 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-312-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_036
		RQ1004_037

10.4.3.1.3 TCP_313 - Request to OPEN TCP Connection for WAN

Test identification	TCP_313
Test objectives	TCP control service gate shall be able to open a TCP connection for WAN access.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends aTCP-313-command-01 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-313-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWWAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends aTCP-313-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-313-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_034
		RQ1004_038

10.4.3.1.4 TCP_314 - Request to OPEN TCP Connection for LAN

Test identification	TCP_314
Test objectives	TCP control service gate shall be open a TCP connection with on a LAN.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-314-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-314-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aIP: aIPV4: eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWLAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends an aTCP-314-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-314-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_042
		RQ1004_055

10.4.3.1.5 TCP_315 - Request to OPEN TCP Connection for LAN with a non-reachable endpoint

Test identification	TCP_315
Test objectives	TCP control service gate shall not open a TCP connection with on a LAN with an invalid address.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-315-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-315-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aFQDN : eIP-test-1,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWLAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends an aTCP-314-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-315-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-E-NOK`
	`}`
	`-- ASN1STOP`
		RQ1004_012
		RQ1004_042
		RQ1004_053
		RQ1004_054
		RQ1004_055

10.4.3.1.6 TCP_316 - Request to OPEN TCP Connection for LAN with a non-accessible port

Test identification	TCP_316	
Test objectives	TCP control service gate shall not open a TCP connection with on a LAN with an invalid port.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-316-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-316-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aIP eIP-test,`
	`  aPortNumber ePortNumber-test-1,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWLAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends an aTCP-314-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-316-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-E-NOK`
	`}`
	`-- ASN1STOP`
		RQ1004_012
		RQ1004_042
		RQ1004_053
		RQ1004_054
		RQ1004_055

10.4.3.1.7 TCP_317 - Request to OPEN TCP Connection for LAN with multiple TCP connections

Test identification	TCP_316	
Test objectives	TCP control service gate shall be able to open multiple TCP connections with on a LAN	
Configuration reference	CTCP_001	
Initial conditions		
Test sequence		
Step	**Description**	**Requirements**
1	The test TCP_311 is executed more than once.	RQ1004_020 RQ1004_007

10.4.3.1.8 TCP_318 - Request to OPEN TCP Connection with FQDN

Test identification	TCP_318	
Test objectives	If supported then TCP control service gate shall open TCP connection and return response successful.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-318-command-01 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-318-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aFQDN : eIP-test-2,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName eNetworkAccessName-test,`
	`  aUserLogin eUserLogin-test,`
	`  aUserPassword eUserPassword-test`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-318-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-318-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_011

10.4.3.1.9 TCP_319 - Request to OPEN TCP Connection with IPV4Adr address type

Test identification	TCP_319
Test objectives	TCP control service gate shall open TCP connection and return response successful.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-319-command-01 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-319-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aIP  eIP-test-3,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName eNetworkAccessName-test,`
	`  aUserLogin eUserLogin-test,`
	`  aUserPassword eUserPassword-test`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-319-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-319-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_014
		RQ1004_031
		RQ1004_032
		RQ1004_036
		RQ1004_037
		RQ1004_039
		RQ1004_040
		RQ1004_041
		RQ1004_042
		RQ1004_052

10.4.3.1.10 TCP_3110 - Request to OPEN TCP Connection with IPV6 address type

Test identification	TCP_3110
Test objectives	TCP control service gate shall open TCP connection with IPV6 address and return response successful.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-3110-command-01 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-3110-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode ePassiveLocal,`
	`  aDestinationAddress aIP  eIP-test-4,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName eNetworkAccessName-test,`
	`  aUserLogin eUserLogin-test,`
	`  aUserPassword eUserPassword-test`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-319-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-3110-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_014
		RQ1004_031
		RQ1004_032
		RQ1004_036
		RQ1004_037
		RQ1004_039
		RQ1004_040
		RQ1004_041
		RQ1004_042
		RQ1004_052

10.4.3.2 TCP Active Connection opening

10.4.3.2.1 TCP_321 - Request to OPEN TCP Connection

Test identification	TCP_321	
Test objectives	TCP control service gate shall open TCP connection in active mode and return response successful.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate an aTCP-321-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-321-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode eActive,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName eNetworkAccessName-test,`
	`  aUserLogin eUserLogin-test,`
	`  aUserPassword eUserPassword-test`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends aTCP-321-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-321-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_010
		RQ1004_029
		RQ1004_033

10.4.3.2.2 TCP_322 - Request to OPEN TCP Connection without network parameters

Test identification	TCP_322	
Test objectives	TCP control service gate shall open TCP connection without network parameters (optional) and return response successful.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-322-command-02 to other host TCP control service gate with:
	`-- ASN1START`
	`aTCP-322-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode eActive,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive`
	`}`
	`-- ASN1STOP`	

2	Other host TCP control service gate sends aTCP-322-response-02 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-322-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`	

10.4.3.2.3 TCP_323 - Request to OPEN TCP Connection for WAN

Test identification	TCP_323
Test objectives	TCP control service gate shall open TCP connection with a WAN access in active mode.
Configuration reference	CTCP_001

Initial conditions

The PTCP_021 shall be successfully executed.
One of following tests shall be executed:

- PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host.
- PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host.
- PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-323-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-323-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode eActive,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWWAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends an aTCP-323-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-323-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_017

10.4.3.2.4 TCP_324 - Request to OPEN TCP Connection for LAN

Test identification	TCP_324	
Test objectives	TCP control service gate shall open TCP connection with only with LAN access in active mode.	
Configuration reference	CTCP_001	
Initial conditions		
The PTCP_021 shall be successfully executed. One of following tests shall be executed: • PTCP_022. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the REE host. • PTCP_023. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the TEE host. • PTCP_024. The pipe session shall be open between the TCP control application gate and the TCP control service gate in the MBM host.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-324-command-01 to TCP control service with:
	`-- ASN1START`
	`aTCP-324-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-REQUEST-CONNECTION-Service-Command : {`
	`  aConnectionMode eActive,`
	`  aIP eIP-test,`
	`  aPortNumber ePortNumber-test,`
	`  aGateID eGateID-test,`
	`  aTimeout eTimeout-passive,`
	`  aNetworkParameters {`
	`  aBearerType eWLAN`
	`  }`
	`}`
	`-- ASN1STOP`	

2	TCP control service sends an aTCP-324-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-324-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-REQUEST-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<STORE(aConnectionID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_017

10.4.3.3 TCP Connection closing

10.4.3.3.1 TCP_331 - TCP control application requests to close the connection

Test identification	TCP_331	
Test objectives	The TCP control service shall be able to close connections, which are successfully created on the request of TCP control application. The test is valid whatever the host domain.	
Configuration reference	CTCP_001	
Initial conditions		
The following test shall be executed: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-321-command-02 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-331-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::=`
	`aTCP-CLOSE-CONNECTION-Service-Command : {`
	`  aConnectionID 0 /*<REPLACE(aConnectionID)>*/`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-331-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-331-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-CLOSE-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0/*<COMPARE(aConnectionID,EQ)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
	The tester shall verify that the TCP connection is closed on the remote TCP endpoint.
		RQ1004_044
		RQ1004_045
		RQ1004_067

10.4.3.3.2 TCP_332 - TCP control application requests to close the connection from the remote endpoint

Test identification	TCP_332
Test objectives	The TCP control service shall close the pipe session on the TCP data application if the connection is closed. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

One of the following tests shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Or the TCP_331 is successfully executed.

Test sequence

Step	Description	Requirements
1	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the host A with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the TCP data application gate.	RQ1004_023

10.4.3.3.3 TCP_333 - TCP control application requests to close pipe session on TCP data service gate

Test identification	TCP_333
Test objectives	The TCP control service shall close the pipe session on the TCP data application if the pipe session on the TCP control service gate is closed. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

One of the following tests shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Or the TCP_331 is successfully executed.

Test sequence

Step	Description	Requirements
1	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the other host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the TCP control service gate.	RQ1004_025
2	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the host A with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the TCP data application gate.	RQ1004_023 RQ1004_066

10.4.3.4 TCP Status connection

10.4.3.4.1 TCP_341 - TCP control application requests the status of a connection

Test identification	TCP_341	
Test objectives	The TCP control service able to request the status of a connection, which has been, successfully opened on the request of TCP control application.	
Configuration reference	CTCP_001	
Initial conditions		
One of the following test shall be executed: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-341-command-01 to the TCP control service gate with:
	`-- ASN1START`
	`aTCP-341-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-GET-STATUS-CONNECTION-Service-Command: {`
	`  aConnectionID 0 /*<REPLACE(aConnectionID)>*/`
	`}`
	`-- ASN1STOP`	

2	The TCP control service gate sends an aTCP-341-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-341-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-GET-STATUS-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0, /*<COMPARE(aConnectionID,EQ)>*/`
	
	`  aStateOfConnection eLISTEN`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_013
		RQ1004_047

10.4.3.4.2 TCP_342 - TCP control application requests the status of a connection

Test identification	TCP_342
Test objectives	The TCP control service able to response the status of a connection, which has been, successfully opened on the request of TCP control application. If the connection drops the status shall indicate this new status.
Configuration reference	CTCP_001

Initial conditions

One of the following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

The communication is closed with the remote TCP endpoint.

Test sequence

Step	Description	Requirements

1	TCP control application gate sends an aTCP-342-command-01 to the TCP control service gate with:
	`-- ASN1START`
	`aTCP-342-command-01 TCP-CONTROL-SERVICE-GATE-Commands ::= aTCP-GET-STATUS-CONNECTION-Service-Command: {`
	`  aConnectionID 0 /*<REPLACE(aConnectionID)>*/`
	`}`
	`-- ASN1STOP`	

2	Other host TCP control service gate sends an aTCP-342-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-342-response-01 TCP-CONTROL-SERVICE-GATE-Responses ::= aTCP-GET-STATUS-CONNECTION-Service-Response : {`
	`  aTCP-Control-Service-Response eTCP-OK,`
	`  aParameter {`
	`  aConnectionID 0, /*<COMPARE(aConnectionID,EQ)>*/`
	
	`  aStateOfConnection eCLOSED`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1004_015
		RQ1004_019
		RQ1004_024

10.4.3.5 TCP data exchange

10.4.3.5.1 TCP_351 - data stream exchange

Test identification	TCP_351	
Test objectives	To data transfer by TCP control application, TCP control application shall notify to other host TCP control service gate by an event.	
Configuration reference	CTCP_001	
Initial conditions		
The following test shall be executed at least twice: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**
1	Administration gate sends EVT_ADM_BIND to Administration gate with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the TCP data service gate. • $GATE_{TCP\_DATA}$: The UUID gate identifier of the TCP data service gate as retrieve by the TCP-REQUEST-CONNECTION-Service-Command.	RQ1004_018 RQ1004_066
2	Administration gate sends EVT_ADM_BIND to Administration gate with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the TCP data application gate. • $GATE_{TCP\_DATA}$: The UUID gate identifier of the TCP data service gate as retrieve by the TCP-REQUEST-CONNECTION-Service-Command. $GATE_{TCP}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.	
3	At data transfer, Other host TCP control application gate sends data. An event flow related to the credit-based data flow control and the data acknowledgement related to $PIPE_{AB}$ shall be observed as defined in clause D.1 in ETSI TS 103 666-1 [1].	RQ1004_016 RQ1004_021 RQ1004_022 RQ1004_070

10.4.3.6 TCP connection accept connection

10.4.3.6.1 TCP_361 - TCP control application accepts incoming connection

Test identification	TCP_361
Test objectives	The TCP control service (active mode) shall be able to accept an incoming connection. The test is valid whatever the host domain.
Configuration reference	CTCP_001
Initial conditions	
The following test shall be executed: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.	

Test sequence		
Step	**Description**	**Requirements**

1	TCP control application gate sends an aTCP-361-command-02 to TCP control service gate with:
	`-- ASN1START`
	`aTCP-361-command-01 TCP-CONTROL-APPLICATION-GATE-Commands ::= aTCP-ACCEPT-CONNECTION-Application-Command : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aSourcePortNumber 1`
	`}`
	`-- ASN1STOP`	

2	TCP control service gate sends an aTCP-361-response-01 response to TCP control application gate with:
	`-- ASN1START`
	`aTCP-361-response-01 TCP-CONTROL-APPLICATION-GATE-Responses ::= aTCP-ACCEPT-CONNECTION-Application-Response: {`
	`  aTCP-Control-Application-Response eTCP-OK`
	`}`
	`-- ASN1STOP`
		RQ1004_050
		RQ1004_051
		RQ1004_052
		RQ1004_054
		RQ1004_055

10.4.3.7 TCP connection event

10.4.3.7.1 TCP_371 - TCP control application events - eREDIRECTION

Test identification	TCP_371
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is a redirection. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence		
Step	**Description**	**Requirements**

1	TCP control service gate sends an aTCP-361-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-371-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eREDIRECTION,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_057
		RQ1004_059

10.4.3.7.2 TCP_372 - TCP control application events - eUNREACHABLE

Test identification	TCP_372
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the remote TCP endpoint is not reachable. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-361-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-372-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eUNREACHABLE,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_056
		RQ1004_058

10.4.3.7.3 TCP_373 - TCP control application events - eIP-HEADER-WRONG

Test identification	TCP_373
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the TCP adapter has detected a wrong IP header. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-373-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-373-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eIP-HEADER-WRONG,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_060

10.4.3.7.4 TCP_374 - TCP control application events - eTIMEOUT

Test identification	TCP_374	
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP endpoint is not accessible. The test is valid whatever the host domain.	
Configuration reference	CTCP_001	
Initial conditions		
The following test shall be executed: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control service gate sends an aTCP-374-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-374-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eTIMEOUT,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_073
		RQ1004_057

10.4.3.7.5 TCP_375 - TCP control application events - eLINK-DROPPED

Test identification	TCP_375
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-375-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-375-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eLINK-DROPPED,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_065

10.4.3.7.6 TCP_376 - TCP control application events - eACCESS-TECHNOLOGY-ERROR

Test identification	TCP_376	
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.	
Configuration reference	CTCP_001	
Initial conditions		
The following test shall be executed: • TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate. • TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	TCP control service gate sends an aTCP-376-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-376-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eACCESS-TECHNOLOGY-ERROR,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_074

10.4.3.7.7 TCP_377 - TCP control application events - eTERMINAL-BUSY

Test identification	TCP_377
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-377-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-377-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eTERMINAL-BUSY,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_061

10.4.3.7.8 TCP_378 - TCP control application events - eNETWORK-BUSY

Test identification	TCP_378
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-375-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-378-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eNETWORK-BUSY,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_062

10.4.3.7.9 TCP_379 - TCP control application events - eCALL-CONTROL-INTERACTION-ERROR

Test identification	TCP_379
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_311. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_312. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_313. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_314. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_321. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_322. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_323. The pipe session shall be open between the TCP control application gate and the TCP control service gate.
- TCP_324. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-375-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-379-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eCALL-CONTROL-INTERACTION-ERROR,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_063

10.4.3.7.10 TCP_3710 - TCP control application events - eDNS-RESOLUTION-ERROR

Test identification	TCP_3710
Test objectives	The TCP control service shall notify the TCP control application gate when the TCP adapter has detected that an error occurred. The event occurs when there is the port of the remote TCP communication has dropped. The test is valid whatever the host domain.
Configuration reference	CTCP_001

Initial conditions

The following test shall be executed:

- TCP_315. The pipe session shall be open between the TCP control application gate and the TCP control service gate.

Test sequence

Step	Description	Requirements

1	TCP control service gate sends an aTCP-375-event-01 to TCP control application gate with:
	`-- ASN1START`
	`aTCP-3710-event-01 TCP-CONTROL-APPLICATION-GATE-Events ::= aEVT-TCP-ERROR-Application-Event : {`
	`  aConnectionID 0, /*<REPLACE(aConnectionID)>*/`
	`  aErrorCode eDNS-RESOLUTION-ERROR,`
	`  aErrorInfo '0000'H`
	`}`
	`-- ASN1STOP`
		RQ1004_064

10.4.3.8 ASN.1 stop

```
-- ASN1START
END
-- ASN1STOP
```

10.4.3.9 Requirements not testable, implicitly verified or verified elsewhere

10.4.3.9.1 Requirements not tested

The following requirements identified in clause 5.6.4 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ1004_043, RQ1004_008

10.4.3.9.2 Implicit requirements

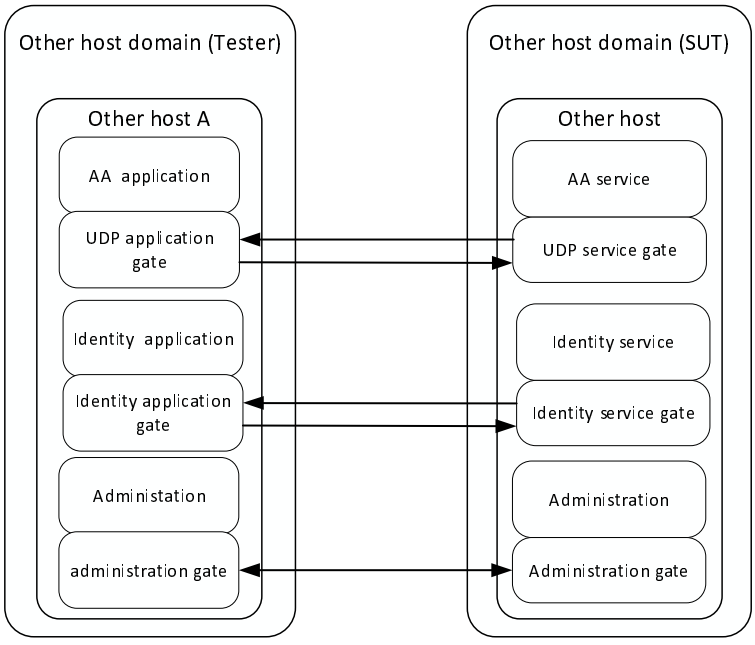
The following requirements identified in clause 5.6.4 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be handled as implicitly verified:

RQ1004_001	RQ1004_002	RQ1004_003	RQ1004_004	RQ1004_005
RQ1004_006	RQ1004_009	RQ1004_024	RQ1004_026	RQ1004_027
RQ1004_030	RQ1004_035	RQ1004_036	RQ1004_048	RQ1004_049
RQ1004_050	RQ1004_068	RQ1004_069	RQ1004_070	RQ1004_071
RQ1004_072	RQ1004_028	RQ1004_046	RQ1004_071	RQ1004_072

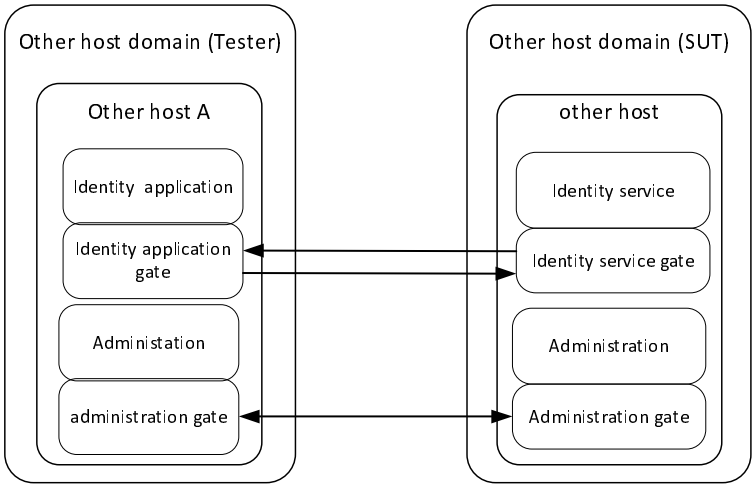
10.5 User Datagram Protocol support

10.5.1 Configurations

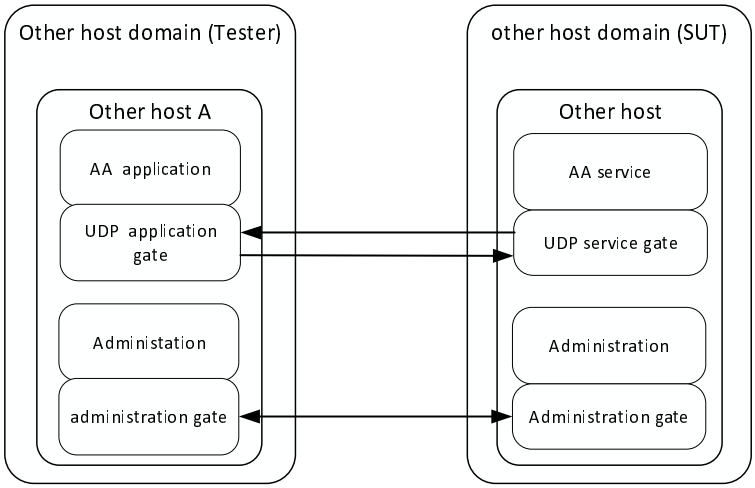
10.5.1.1 CUDP_001 - UDP and Identity services

Configuration identification	CUDP_001
Configuration description	

10.5.1.2 CUDP_002 - Identity service

Configuration identification	CUDP_002
Configuration description	

10.5.1.3 CUDP_003 - Generic UDP service

Configuration identification	CUDP_003
Configuration description	

10.5.1.4 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPUDPconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) udp (7)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
```

```
IMPORTS
   UUID,
   Version,
   VersionType,
UDP-SERVICE-GATE-Commands,
UDP-SERVICE-GATE-Responses,
UDP-SERVICE-GATE-Events,
UDP-APPLICATION-GATE-Events

FROM SSPDefinitions;

eUDPVersion VersionType ::='0100' --Version 01.00

-- ASN1STOP
```

10.5.2 Procedures

10.5.2.1 PUDP_0021 - Open a pipe session with the Identity gate

Procedure Identification	PUDP_0021
Procedure objectives	The tester host shall be able to open a pipe session to the identity service gate of the other host in the SUT. From the GATE_LIST registry, the UUID of the UDP service shall be listed. If the procedure is successful then a pipe session is open between the identity application and the identity service in the other hosts.
Configuration reference	CUDP_002
Initial conditions	
The tester runs the UDP application gate and the SUT runs the UDP service gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the SSP host with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The UDP service identifier shall be present. The test is successful if the previous requirement is satisfied.
5	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the SSP host with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed. This step is required to clean up the context of the tests but it is not essential for the test objective.

10.5.2.2 PUDP_0022 - Open a pipe session with the UDP service in the REE Host domain

Procedure identification	PUDP_0022
Procedure objectives	The other host shall be able to open a pipe session to the UDP service gate of the REE host. If the procedure is successful, then a pipe session is open between the UDP application in the other host and the UDP service in the SSP host.
Configuration reference	CUDP_001
Initial conditions	
The procedure PUDP_021 shall be successfully executed: • The UDP service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the UDP service gate. • $GATE_{UDP}$: The UUID gate identifier of the identity gate (34E27B41-3B9A-59A9-9BA4-2B91292DAFEA).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{URN}$: The UUID gate identifier of the identity gate (34E27B41-3B9A-59A9-9BA4-2B91292DAFEA). $GATE_{UDP}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the test is successful.

10.5.2.3 PUDP_0023 - Open a pipe session with the UDP service in the TEE Host domain

Procedure identification	PUDP_0023
Procedure objectives	The other host shall be able to open a pipe session to the UDP service gate of the TEE host. The UDP service identifier is '0091E79A-9A10-53D9-88AF-187DF566713B'H If the procedure is successful then a pipe session is open between the UDP application in the other host and the UDP service in the SSP host.
Configuration reference	CUDP_001
Initial conditions	
The procedure PUDP_021 shall be successfully executed: • The UDP service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the udp service gate. • $GATE_{UDP}$: The UUID gate identifier of the identity gate (0091E79A-9A10-53D9-88AF-187DF566713B).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{UDP}$: The UUID gate identifier of the identity gate (0091E79A-9A10-53D9-88AF-187DF566713B). $GATE_{UDP}$ shall be present in one of the binding parameters (see VNP[XX]). If present, then the procedure is successful.

10.5.2.4 PUDP_0024 - Open a pipe session with the UDP service in the MBM Host domain

Procedure identification	PUDP_0024
Procedure objectives	The other host shall be able to open a pipe session to the UDP service gate of the MBM host. The UDP service identifier is 'ADCE4843-A058-50F2-A98D-5D3C334504B0'H If the procedure is successful then a pipe session is open between the UDP application in the other host and the UDP service in the SSP host.
Configuration reference	CUDP_001
Initial conditions	
The procedure PUDP_021 shall be successfully executed: • The UDP service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the SSP with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the UDP service gate. • $GATE_{UDP}$: The UUID gate identifier of the identity gate (ADCE4843-A058-50F2-A98D-5D3C334504B0).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{UDP}$: The UUID gate identifier of the identity gate (ADCE4843-A058-50F2-A98D-5D3C334504B0). $GATE_{UDP}$ shall be present in one of the binding parameters (see VNP[XX]). If present then the procedure is successful.

10.5.3 Test descriptions

10.5.3.1 UDP-REQUEST-SOCKET-Command

10.5.3.1.1 UDP_0031 - Request to OPEN UDP Socket

Test identification	UDP_0031
Test objectives	UDP Service gate shall open UDP socket and return response successful.
Configuration reference	CUDP_001

Initial conditions

PUDP_0021 shall be successfully executed.
One of following procedures shall be executed:

- PUDP_0022. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0023. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0024. The pipe session shall be open between the UDP application gate and the UDP service gate.

Test sequence

Step	**Description**	**Requirements**

1	AA gate sends an aUDP -0031-command-01 to UDP gate with:
	`-- ASN1START`
	`aUDP-0031-command-01 UDP-SERVICE-GATE-Commands::= aUDP-REQUEST-SOCKET-Command : {`
	
	`  aPortNumber 1,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName '00'H,`
	`  aUserLogin '00'H,`
	`  aUserPassword '00'H`
	`  },`
	`  aLocalOnly FALSE`
	`}`
	`-- ASN1STOP`
	
	Set aLocalOnly FALSE in the command.	

2	UDP gate sends aUDP-0031-response-01 response to AA gate with:
	`-- ASN1START`
	`aUDP-0031-response-01 UDP-SERVICE-GATE-Responses ::= aUDP-REQUEST-SOCKET-Response : {`
	`  aUDP-Service-Response eUDP-OK,`
	`  aParameter {`
	`  aSocketID 100 /*<STORE(aSocketID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1005_016
		RQ1005_010
		RQ1005_012

10.5.3.1.2 UDP_0032 - Request to OPEN UDP Socket while port no is missing

Test identification	UDP_0032
Test objectives	UDP Service gate shall open UDP socket while port number is missing in command and return response successful.
Configuration reference	CUDP_001

Initial conditions

PUDP_0021 shall be successfully executed.
One of following procedures shall be executed:

- PUDP_0022. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0023. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0024. The pipe session shall be open between the UDP application gate and the UDP service gate.

Test sequence

Step	**Description**	**Requirements**

1	AA Application UDP gate sends an aUDP-0032-command-02 to other host UDP service gate with:
	`-- ASN1START`
	`aUDP-0032-command-01 UDP-SERVICE-GATE-Commands::= aUDP-REQUEST-SOCKET-Command : {`
	
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName '00'H,`
	`  aUserLogin '00'H,`
	`  aUserPassword '00'H`
	`  },`
	`  aLocalOnly FALSE`
	`}`
	`-- ASN1STOP`
	
	Set aLocalOnly FALSE in the command.	

2	Other host UDP gate sends aUDP-0032-response-02 response to AA application UDP gate with:
	`-- ASN1START`
	`aUDP-0032-response-01 UDP-SERVICE-GATE-Responses ::= aUDP-REQUEST-SOCKET-Response : {`
	`  aUDP-Service-Response eUDP-OK,`
	`  aParameter {`
	`  aSocketID 100 /*<STORE(aSocketID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1005_016
		RQ1005_010
		RQ1005_011

10.5.3.1.3 UDP_0033 - Request to OPEN UDP Socket with entities present in terminal

Test identification	UDP_0033
Test objectives	UDP Service gate shall open UDP socket with only with entities present in terminal.
Configuration reference	CUDP_001

Initial conditions

PUDP_0021 shall be successfully executed.
One of following tests shall be executed:

- PUDP_0022. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0023. The pipe session shall be open between the UDP application gate and the UDP service gate.
- PUDP _0024. The pipe session shall be open between the UDP application gate and the UDP service gate.

Test sequence

Step	**Description**	**Requirements**

1	AA gate sends an aUDP -0033-command-01 to UDP gate with:
	`-- ASN1START`
	`aUDP-0031-command-01 UDP-SERVICE-GATE-Commands::= aUDP-REQUEST-SOCKET-Command : {`
	
	`  aPortNumber 1,`
	`  aNetworkParameters {`
	`  aBearerType eDefaultBearer,`
	`  aNetworkAccessName '00'H,`
	`  aUserLogin '00'H,`
	`  aUserPassword '00'H`
	`  },`
	`  aLocalOnly TRUE`
	`}`
	`-- ASN1STOP`
	
	Set aLocalOnly TRUE in the command.	

2	UDP gate sends aUDP-0033-response-01 response to AA gate with:
	`-- ASN1START`
	`aUDP-0031-response-01 UDP-SERVICE-GATE-Responses ::= aUDP-REQUEST-SOCKET-Response : {`
	`  aUDP-Service-Response eUDP-OK,`
	`  aParameter {`
	`  aSocketID 100 /*<STORE(aSocketID)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1005_016
		RQ1005_015

3	At data transfer, Other host UDP application gate sends aUDP-0033-Event-01 to other host UDP service gate with:
	`-- ASN1START`
	
	`aUDP-0033-Event-01 UDP-SERVICE-GATE-Events ::= aEVT-UDP-DATAGRAM-OUT-Service-Event : {`
	`  aSocketID 100, /*<REPLACE(aSocketID)> */`
	`  aDestinationAddress aIP : aIPV4 : 'C0A80000'H,`
	`  aDestinationPortNumber 1,`
	`  aData '11223444556677889900'H`
	`}`
	`-- ASN1STOP`	

10.5.3.2 UDP-CLOSE-SOCKET-Command

10.5.3.2.1 UDP_0041 - UDP application requests to close the socket

Test identification	UDP_0041	
Test objectives	The UDP service able to close sockets, which are successfully created on the request of UDP application.	
Configuration reference	CUDP_001	
Initial conditions		
The following test shall be executed: • UDP_0031. The pipe session shall be open between the UDP application gate and the UDP service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AA Application UDP gate sends an aUDP-0041-command-02 to other host UDP service gate with:
	`-- ASN1START`
	`aUDP-0041-command-01 UDP-SERVICE-GATE-Commands ::= aUDP-CLOSE-SOCKET-Command : {`
	`  aSocketID 100 /*<REPLACE(aSocketID)>*/`
	`}`
	`-- ASN1STOP`	

2	Other host UDP service gate sends a udp-0041-response-02 response to SSP application UDP gate with:
	`-- ASN1START`
	`aUDP-0041-response-01 UDP-SERVICE-GATE-Responses ::= aUDP-CLOSE-SOCKET-Response : {`
	`  aUDP-Service-Response eUDP-OK,`
	`  aParameter {`
	`  aSocketID 100 /*<COMPARE(aSocketID,EQ)>*/`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1005_010
		RQ1005_018
		RQ1005_016

10.5.3.3 UDP-EVT-UDP-DATAGRAM-OUT-Service-Event

10.5.3.3.1 UDP_0051 - EVT-UDP-DATAGRAM-OUT-Service-Event trigger

Test identification	UDP_0051	
Test objectives	To data transfer by UDP application, UDP Application shall notify to other host UDP service gate by an event.	
Configuration reference	CUDP_001	
Initial conditions		
The following test shall be executed: • UDP_0031. The pipe session shall be open between the UDP application gate and the UDP service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	At data transfer, Other host UDP application gate sends aUDP-0051-Event-01 to other host UDP service gate with:
	`-- ASN1START`
	`aUDP-0051-Event-01 UDP-SERVICE-GATE-Events ::= aEVT-UDP-DATAGRAM-OUT-Service-Event : {`
	`  aSocketID 100, /*<REPLACE(aSocketID)> */`
	`  aDestinationAddress aIP : aIPV4 : 'A8C00000'H,`
	`  aDestinationPortNumber 1,`
	`  aData '11223444556677889900'H`
	`}`
	`-- ASN1STOP`
		RQ1005_022
		RQ1005_023

10.5.3.3.2 UDP_0052 - EVT-UDP-DATAGRAM-OUT-Service-Event trigger with FQDN values

Test identification	UDP_0052	
Test objectives	To data transfer by UDP application, UDP Application shall notify to other host UDP service gate by an event with FQDN values and UDP adapter shall perform DNS resolution if supported.	
Configuration reference	CUDP_001	
Initial conditions		
The following test shall be executed: • UDP_0031. The pipe session shall be open between the UDP application gate and the UDP service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	At data transfer, Other host UDP application gate sends aUDP-0052-Event-01 to other host UDP service gate with:
	`-- ASN1START`
	
	`aUDP-0052-Event-01 UDP-SERVICE-GATE-Events ::= aEVT-UDP-DATAGRAM-OUT-Service-Event : {`
	`  aSocketID 100, /*<REPLACE(aSocketID)> */`
	`  aDestinationAddress aFQDN :`
	`{"6D796D61696C2E736F6D65636F6C6C65",{0,0,0,10},"67652E656475"},`
	`  aDestinationPortNumber 1,`
	`  aData '11223444556677889900'H`
	`}-- ASN1STOP`
		RQ1005_022
		RQ1005_024

10.5.3.3.3 UDP_0063 - EVT-UDP-DATAGRAM-OUT-Service-Event trigger on pipe close

Test identification	UDP_0053	
Test objectives	All sockets are closed and UDP Application shall notify to other host UDP service gate by an event for data transfer.	
Configuration reference	CUDP_001	
Initial conditions		
The following test shall be executed: • UDP_0041. All open pipe session between UDP application and UDP service gate shall close.		
Test sequence		
Step	**Description**	**Requirements**

1	At data transfer, Other host UDP application gate sends aUDP-0063-Event-01 to other host UDP service gate with:
	`-- ASN1START`
	
	`aUDP-0063-Event-01 UDP-SERVICE-GATE-Events ::= aEVT-UDP-DATAGRAM-OUT-Service-Event : {`
	`  aSocketID 100, /*<REPLACE(aSocketID)> */`
	`  aDestinationAddress aIP : aIPV4 : 'A8C00000'H,`
	`  aDestinationPortNumber 1,`
	`  aData '11223444556677889900'H`
	`}`
	`-- ASN1STOP`
		RQ1005_008

2	Other host UDP service gate sends aUDP-0063-Event-01 to other host UDP application gate with:
	`-- ASN1START`
	
	`aUDP-0063-Event-01 UDP-APPLICATION-GATE-Events ::= aEVT-UDP-ERROR-Application-Event : {`
	`  aSocketID 100, /*<REPLACE(aSocketID)> */`
	`  aErrorCode eUNREACHABLE,`
	`  aErrorInfo '0007'H`
	`}`
	`-- ASN1STOP`	

10.5.3.4 UDP-EVT-UDP-DATAGRAM-IN-Application-Event

10.5.3.4.1 UDP_0061 - EVT-UDP-DATAGRAM-IN-Application-Event trigger

Test identification	UDP_0061	
Test objectives	To data transfer by UDP adapter, UDP service gate shall notify to other host UDP application gate by an event.	
Configuration reference	CUDP_001	
Initial conditions		
The following test shall be executed: • UDP_0031. The pipe session shall be open between the UDP application gate and the UDP service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	At data transfer, Other host UDP service gate sends aUDP-0061-Event-01 to other host UDP application gate with:
	`-- ASN1START`
	
	`aUDP-0061-Event-01 UDP-APPLICATION-GATE-Events ::= aEVT-UDP-DATAGRAM-IN-Application-Event : {`
	`  asocketID 100, /*<COMPARE(aSocketID,EQ)>*/`
	`  aSourceIP aIPV4 : 'A8C00000'H,`
	`  aSourcePortNumber 1,`
	`  aData '00112233445566778899'H`
	`}`
	`-- ASN1STOP`
		RQ1005_025
		RQ1005_026

10.5.3.4.2 UDP_0062 - EVT-UDP-ERROR-Application-Event trigger

Test identification	UDP_0062	
Test objectives	When an error occurred, UDP service gate notifies the UDP consumer via UDP application gate by an event.	
Configuration reference	UDP_001	
Initial conditions		
The following test shall be executed: • UDP_0031. The pipe session shall be open between the UDP application gate and the UDP service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	At an error, Other host UDP service gate sends aUDP-0062-Event-01 to other host UDP application gate with:
	`-- ASN1START`
	
	`aUDP-0062-Event-01 UDP-APPLICATION-GATE-Events ::= aEVT-UDP-ERROR-Application-Event : {`
	`  aSocketID 100, /*<COMPARE(aSocketID,EQ)>*/`
	`  aErrorCode eNETWORK-BUSY,`
	`  aErrorInfo '0007'H`
	`}`
	`-- ASN1STOP`
		RQ1005_025
		RQ1005_027

10.5.3.5 UDP ASN.1 descriptions

10.5.3.5.1 End of ASN.1 structure

The annex shall be appended at the end of the UDP test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

10.5.3.6 Requirements not testable, implicitly verified or verified elsewhere

10.5.3.6.1 Implicit requirements

The following requirements identified in clause 5.6.4 are generated from descriptive text. An explicit verification is not possible but with correct execution of the related function the requirements can be considered implicitly verified.

RQ1005_001	RQ1005_002	RQ1005_003
RQ1005_004	RQ1005_005	RQ1005_007
RQ1005_009	RQ1005_013	RQ1005_014
RQ1005_017	RQ1005_019	
RQ1005_020	RQ1005_021	

10.5.3.6.2 Not Testable Requirements

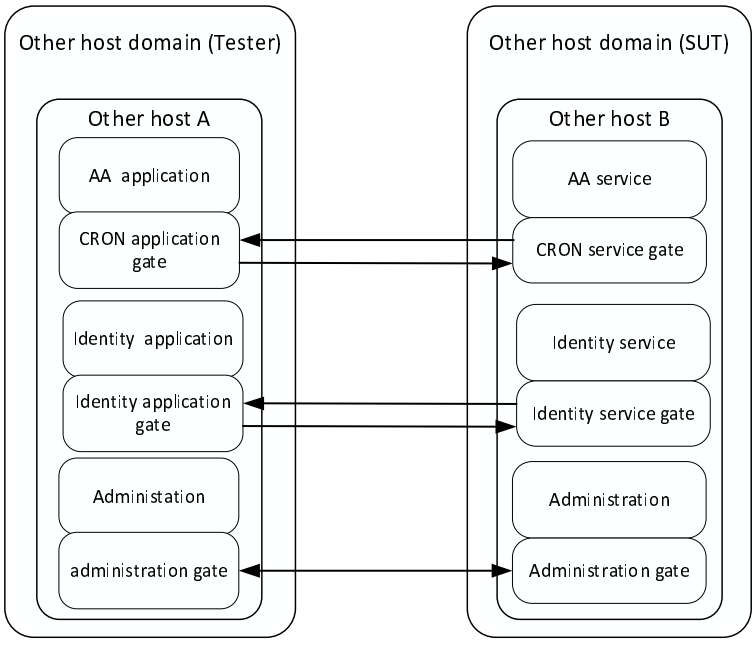
The following requirements identified in clause 5.6.4 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ1005_006	RQ1005_009	
RQ1005_013	RQ1005_014	RQ1005_017
RQ1005_019	RQ1005_020	RQ1005_021

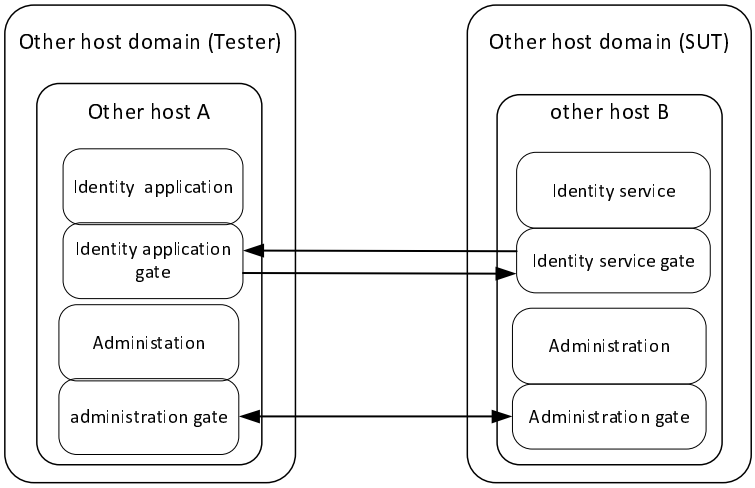
10.6 CRON service support

10.6.1 Configurations

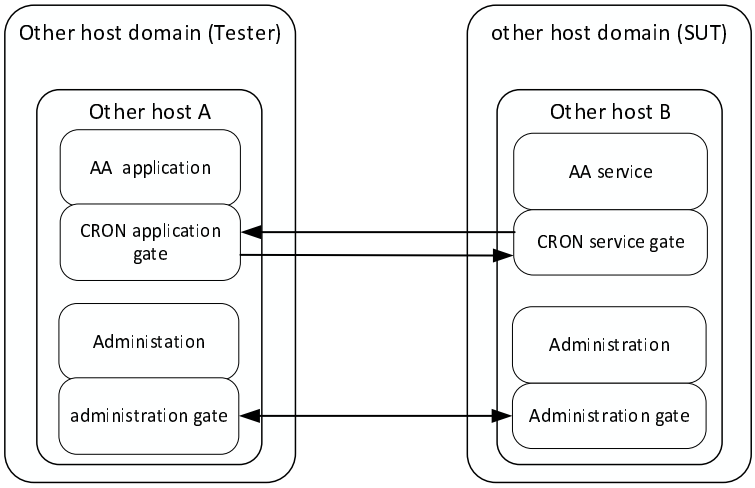
10.6.1.1 CCRO_001 - CRON and Identity services

Configuration identification	CCRO_001
Configuration description	

10.6.1.2 CCRO_002 - Identity service

Configuration identification	CCRO_002
Configuration description	

10.6.1.3 CCRO_003 - Generic CRON service

Configuration identification	CCRO_003
Configuration description	

10.6.1.4 ASN.1 definitions

The following definitions are used for the procedures and the test descriptions.

```
-- ASN1START
SSPCRONconfigurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part1 (1) test (2) cron (8)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
```

```
IMPORTS
   CRON-SERVICE-GATE-Commands,
   CRON-SERVICE-GATE-Responses,
   CRON-APPLICATION-GATE-Events,
   UUID,
   Version,
   VersionType
FROM SSPDefinitions;

eCRONVersion VersionType ::='0100' --Version 01.00
eDateFuture    GeneralizedTime::= "20410629114501.000" /* Date in the future June 29, 2041*/
eDatePast   GeneralizedTime::= "20110629114501.000" /* Date in the past June 29, 2011*/

-- ASN1STOP
```

10.6.2 Procedures

10.6.2.1 PCRO_021 - Open a pipe session with the Identity gate

Procedure identification	PCRO_021
Procedure objectives	The host A shall be able to open a pipe session to the identity service gate of the host B (SUT). From the GATE_LIST registry, the UUID of the CRON service shall be listed. If the procedure is successful then a pipe session is open between the identity application and the identity service in the other hosts.
Configuration reference	CCRO_002
Initial conditions	
The tester runs the CRON application gate and the SUT runs the CRON service gate.	
Procedure sequence	
Step	**Description**
1	Administration gate of the host A sends EVT_ADM_BIND to Administration gate in the host B with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate in the host B sends EVT_ADM_BIND to Administration gate in the host A with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the host B with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate in the other host. The CRON service identifier shall be present. The procedure is successful if the previous requirement is satisfied.
5	Administration gate in the host A sends EVT_ADM_UNBIND event to the administration gate in the host B with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed. This step is required to clean up the context of the procedure, but it is not essential for the procedure objective.

10.6.2.2 PCRO_022 - Open a pipe session with the CRON service in a host of the REE Host domain

Procedure identification	PCRO_022
Procedure objectives	The host A shall be able to open a pipe session to the CRON service gate of the host B. If the procedure is successful, then a pipe session is open between the CRON application in the other host and the CRON service in the host B.
Configuration reference	CCRO_001
Initial conditions	
The procedure PCRO_021 shall be successfully executed: • The CRON service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate in the host A sends EVT_ADM_BIND to Administration gate in the host B with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the CRON service gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (D67ABDB2-91AC-5B2E-8DF9-A53591E987C0).
2	Administration gate in the host B sends EVT_ADM_BIND to Administration gate in the host A with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (D67ABDB2-91AC-5B2E-8DF9-A53591E987C0). $GATE_{CRON}$ shall be present in one of the binding parameters (see VNP[XX]). If present, then the procedure is successful.

10.6.2.3 PCRO_023 - Open a pipe session with the CRON service in the TEE Host domain

Procedure identification	PCRO_023
Procedure objectives	The host A shall be able to open a pipe session to the CRON service gate of the host B in the TEE host domain. The CRON service identifier is 'E5C6D5E1-6376-5B2D-A158-F11B5E7BA7AE'H If the procedure is successful, then a pipe session is open between the CRON application gate in the host B and the CRON service gate in the host A.
Configuration reference	CCRO_001
Initial conditions	
The procedure PCRO_021 shall be successfully executed: • The CRON service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate in the host A sends EVT_ADM_BIND to Administration gate in the host B with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the CRON service gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (E5C6D5E1-6376-5B2D-A158-F11B5E7BA7AE).
2	Administration gate in the host B sends EVT_ADM_BIND to Administration gate in the host A with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (E5C6D5E1-6376-5B2D-A158-F11B5E7BA7AE). $GATE_{CRON}$ shall be present in one of the binding parameters (see VNP[XX]). If present, then the procedure is successful.

10.6.2.4 PCRO_024 - Open a pipe session with the CRON service in the MBM Host domain

Procedure identification	PCRO_024
Procedure objectives	The host A shall be able to open a pipe session to the CRON service gate of the MBM host. The CRON service identifier is '51FE5F0F-3BAA-506B-8CB5-AFD7562268E8'H If the test is successful then a pipe session is open between the CRON application in the other host and the CRON service in the MBM host.
Configuration reference	CCRO_001
Initial conditions	
The procedure PCRO_021 shall be successfully executed: • The CRON service identifier is present in the GATE_LIST registry of the Identity gate.	
Procedure sequence	
Step	**Description**
1	Administration gate in the host A sends EVT_ADM_BIND to Administration gate in the MBM host with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the CRON service gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (51FE5F0F-3BAA-506B-8CB5-AFD7562268E8).
2	Administration gate in the host B sends EVT_ADM_BIND to Administration gate in the host A with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{CRON}$: The UUID gate identifier of the identity gate (51FE5F0F-3BAA-506B-8CB5-AFD7562268E8). $GATE_{CRON}$ shall be present in one of the binding parameters (see VNP[XX]. If present then the procedure is successful.

10.6.3 Test Descriptions

10.6.3.1 CRON-REQUEST-TIMER-Command

10.6.3.1.1 CRO_031 - Request a CRON timer

Test identification	CRO_031	
Test objectives	CRON Service gate shall open a timer of absolute date and time specified in command.	
Configuration reference	CCRO_001	
Initial conditions		
PCRO_021 shall be successfully executed. One of following procedures shall be executed: • PCRO_022. The pipe session is opened between the CRON application gate and the CRON service gate. • PCRO_023. The pipe session is opened between the CRON application gate and the CRON service gate. • PCRO_024. The pipe session is opened between the CRON application gate and the CRON service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aCRO-031-command-01 to CRON gate with:
	`-- ASN1START`
	`aCRO-031-command-01 CRON-SERVICE-GATE-Commands ::= aCRON-REQUEST-TIMER-Command : {`
	`  aInitialNotificationDateTime aDateTimeAbsolute eDateFuture,`
	`aPeriod 36000`
	`}`
	`-- ASN1STOP`
	
	Set an alarm the June,29 2021 at 11H45 and then every hour.
		RQ1006_006
		RQ1006_007
		RQ1006_008

2	CRON gate sends aCRO-031-response-01 response to AAA gate with:
	`-- ASN1START`
	`aCRO-031-response-01 CRON-SERVICE-GATE-Responses ::= aCRON-REQUEST-TIMER-Response : {`
	`  aCRON-Service-Response eCRON-OK,`
	`  aParameter {`
	`  aCRON-ID 0, /*<STORE(eCRONSession)>*/`
	`  aPersistantOverPowerCycle FALSE`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1006_011
		RQ1006_012

10.6.3.1.2 CRO_032 - CRON service does not support absolute time

Test identification	CRO_032
Test objectives	CRON Service gate does not support absolute date and time. At absolute date and time CRON request of application gate, service gate reject it.
Configuration reference	CCRO_001

Initial conditions

PCRO_021 shall be successfully executed.
One of following tests shall be executed:

- PCRO_022. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_023. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_024. The pipe session is opened between the CRON application gate and the CRON service gate.

Test sequence

Step	**Description**	**Requirements**

1	AA Application CRON gate sends an aCRO-032-command-02 to other host CRON service gate with:
	`-- ASN1START`
	`aCRO-032-command-02 CRON-SERVICE-GATE-Commands ::= aCRON-REQUEST-TIMER-Command : {`
	`  aInitialNotificationDateTime aDateTimeAbsolute eDateFuture,`
	`  aPeriod 36000`
	`}`
	`-- ASN1STOP`
	
	Set an alarm the June,29 2021 at 11H45 and then every hour.
		RQ1006_006
		RQ1006_007
		RQ1006_008

2	Other host CRON gate sends aCRO-032-response-02 response to AA application CRON gate with:
	`-- ASN1START`
	`aCRO-032-response-02 CRON-SERVICE-GATE-Responses ::= aCRON-REQUEST-TIMER-Response : {`
	`  aCRON-Service-Response eCRON-E-NO-ABSOLUTE-TIME,`
	`  aParameter {`
	`  aCRON-ID 0, /*<STORE(eCRONSession)>*/`
	`  aPersistantOverPowerCycle FALSE`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1006_009

10.6.3.1.3 CRO_033 - CRON Application request absolute timer in the past

Test identification	CRO_033
Test objectives	CRON service gate executing one CRON timer of absolute time. There is one more request of absolute time. CRON Service gate shall reject this request.
Configuration reference	CCRO_001

Initial conditions

The PCRO_021 shall successfully executed.
One of following tests shall be executed:

- PCRO_022. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_023. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_024. The pipe session is opened between the CRON application gate and the CRON service gate.

CRO_031 shall be successfully executed.

Test sequence

Step	**Description**	**Requirements**

1	The CRON application gate sends an aCRO-033-command-03 to CRON service gate with:
	`-- ASN1START`
	`aCRO-033-command-03 CRON-SERVICE-GATE-Commands ::= aCRON-REQUEST-TIMER-Command : {`
	`  aInitialNotificationDateTime aDateTimeAbsolute eDatePast,`
	`  aPeriod 36000`
	`}`
	`-- ASN1STOP`
	
	Set an alarm the June,29 2019 at 11H45 and then every hour.
		RQ1006_006
		RQ1006_007
		RQ1006_008

2	The CRON service gate sends aCRO-033-response-03 response to CRON application gate with:
	`-- ASN1START`
	`aCRO-031-response-03 CRON-SERVICE-GATE-Responses ::= aCRON-REQUEST-TIMER-Response : {`
	`  aCRON-Service-Response eCRON-E-NOK,`
	`  aParameter {`
	`  aCRON-ID 0, /*<STORE(eCRONSession)>*/`
	`  aPersistantOverPowerCycle FALSE`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1006_010

10.6.3.2 CRON-READ-DATE-TIME-Command

10.6.3.2.1 CRO_041 - Read the time and date

Test identification	CRO_041
Test objectives	The CRON service gate able to return date and time successfully.
Configuration reference	CCRO_001

Initial conditions

The PCRO_021 shall be successfully executed.
One of following tests shall be executed:

- PCRO_022. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_023. The pipe session is opened between the CRON application gate and the CRON service gate.
- PCRO_024. The pipe session is opened between the CRON application gate and the CRON service gate.

Test sequence

Step	Description	Requirements

1	The CRON application gate sends an aCRO-041-command-02 to the CRON service gate with:
	`-- ASN1START`
	`aCRO-041-command-01 CRON-SERVICE-GATE-Commands ::= aCRON-READ-DATE-Command : {`
	`}`
	`-- ASN1STOP`
		RQ1006_013

2	The CRON service gate sends aCRO-041-response-02 response to CRON application gate with:
	`-- ASN1START`
	`aCRO-041-response-01 CRON-SERVICE-GATE-Responses ::= aCRON-READ-DATE-TIME-Response : {`
	`  aCRON-Service-Response eCRON-OK,`
	`  aParameter {`
	`  aDateTime "20210629114501.000"`
	`  }`
	`}-- ASN1STOP`
		RQ1006_014
		RQ1006_015

10.6.3.3 CRON-KILL-TIMER-Command

10.6.3.3.1 CRO_051 - CRON application requests to kill a timer

Test identification	CRO_051	
Test objectives	The CRON service able to kill a timer which are successfully created on the request of CRON application before expire.	
Configuration reference	CCRO_001	
Initial conditions		
The following test shall be executed: • CRO_031. A timer has been created.		
Test sequence		
Step	**Description**	**Requirements**

1	The CRON application gate sends an aCRO-051-command-02 to the CRON service gate with:
	`-- ASN1START`
	`aCRO-051-command-02 CRON-SERVICE-GATE-Commands ::= aCRON-KILL-TIMER-Command : {`
	`  aCRON-ID 0 /*<REPLACE(eCRONSession)>*/`
	`}`
	
	`-- ASN1STOP`
		RQ1006_016
		RQ1006_017

2	CRON service gate sends aCRO-051-response-01 response to the application CRON gate with:
	`-- ASN1START`
	
	`aCRO-051-response-02 CRON-SERVICE-GATE-Responses ::= aCRON-KILL-TIMER-Response : {`
	`  aCRON-Service-Response eCRON-OK`
	`}-- ASN1STOP`
		RQ1006_018

10.6.3.4 CRON-KILL-ALL-TIMERS-Command

10.6.3.4.1 CRO_061 - CRON application requests to kill all timers

Test identification	CRO_061	
Test objectives	The CRON service shall be able to kill all timers which are successfully created on the request of the CRON application before they expire.	
Configuration reference	CCRO_001	
Initial conditions		
The following test shall be executed: • CRO_031. A timer has been successfully created.		
Test sequence		
Step	**Description**	**Requirements**

1	The CRON application gate sends an aCRO-061-command-01 to the CRON service gate with:
	`-- ASN1START`
	`aCRO-061-command-01 CRON-SERVICE-GATE-Commands ::= aCRON-KILL-ALL-TIMERS-Command : {`
	`}`
	`-- ASN1STOP`
		RQ1006_019

2	The CRON service gate sends an aCRO-061-response-01 response to CRON application gate with:
	`-- ASN1START`
	
	`aCRO-061-response-01 CRON-SERVICE-GATE-Responses ::= aCRON-KILL-ALL-TIMERS-Response : {`
	`  aCRON-Service-Response eCRON-OK`
	`}`
	`-- ASN1STOP`
		RQ1006_020

10.6.3.4.2 CRO_062 - CRON application requests to kill all timers twice

Test identification	CRO_062	
Test objectives	The CRON service shall be able to kill all timers which have been already successfully killed. The command shall have no effect and shall always return eCRON-OK.	
Configuration reference	CCRO_001	
Initial conditions		
The following test shall be executed: • CRO_061. All timers have been already killed.		
Test sequence		
Step	**Description**	**Requirements**

1	The CRON application gate sends an aCRO-062-command-01 to the CRON service gate with:
	`-- ASN1START`
	`aCRO-062-command-01 CRON-SERVICE-GATE-Commands ::= aCRON-KILL-ALL-TIMERS-Command : {`
	`}`
	`-- ASN1STOP`
		RQ1006_019

2	The CRON service gate sends an aCRO-062-response-01 response to CRON application gate with:
	`-- ASN1START`
	
	`aCRO-062-response-01 CRON-SERVICE-GATE-Responses ::= aCRON-KILL-ALL-TIMERS-Response : {`
	`  aCRON-Service-Response eCRON-OK`
	`}`
	`-- ASN1STOP`
		RQ1006_020

10.6.3.5 CRON-ELAPSED-TIMER-Event

10.6.3.5.1 CRO_071 - Request a CRON timer

Test identification	CRO_071	
Test objectives	CRON Service gate shall open a timer of absolute date and time specified in command.	
Configuration reference	CCRO_001	
Initial conditions		
The PCRO_021 shall be successfully executed. One of following procedures shall be executed: • PCRO_022. The pipe session is opened between the CRON application gate and the CRON service gate. • PCRO_023. The pipe session is opened between the CRON application gate and the CRON service gate. • PCRO_024. The pipe session is opened between the CRON application gate and the CRON service gate.		
Test sequence		
Step	**Description**	**Requirements**

1	AAA gate sends an aCRO-071-command-01 to CRON service gate with:
	`-- ASN1START`
	`aCRO-071-command-01 CRON-SERVICE-GATE-Commands ::= aCRON-REQUEST-TIMER-Command : {`
	`  aInitialNotificationDateTime aTimeRelative 1,  aPeriod 360`
	`}`
	`-- ASN1STOP`
	
	Set a relative alarm.
		RQ1006_006
		RQ1006_007
		RQ1006_008

2	CRON service gate sends aCRO-071-response-01 response to AAA gate with:
	`-- ASN1START`
	`aCRO-071-response-01 CRON-SERVICE-GATE-Responses ::= aCRON-REQUEST-TIMER-Response : {`
	`  aCRON-Service-Response eCRON-OK,`
	`  aParameter {`
	`  aCRON-ID 0, /*<STORE(eCRONSession)>*/`
	`  aPersistantOverPowerCycle FALSE`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1006_011
		RQ1006_012

10.6.3.5.2 CRO_072 - ELAPSED-TIMER-Event trigger

Test identification	CRO_072	
Test objectives	At timer elapsed, CRON Service shall notify to the CRON application gate by an event.	
Configuration reference	CCRO_001	
Initial conditions		
The following test shall be executed: • CRO_071. A timer with relative time is created.		
Test sequence		
Step	**Description**	**Requirements**

1	At timer elapsed, the CRON service gate sends an aCRO-072-Event-01 response to the CRON application gate with:
	`-- ASN1START`
	
	`aCRO-072-event-01 CRON-APPLICATION-GATE-Events ::= aCRON-ELAPSED-TIMER-Event : {`
	`  aCRON-ID 0 /*<COMPARE(eCRONSession,EQ)>*/`
	`}`
	`-- ASN1STOP`
		RQ1006_022
		RQ1006_023

10.6.3.6 End of test descriptions - CRON ASN.1 descriptions

10.6.3.6.1 Annex - End of ASN.1 structure

The annex shall be appended at the end of the CRON test descriptions.

```
-- ASN1START
END
-- ASN1STOP
```

10.6.3.7 Requirements not testable

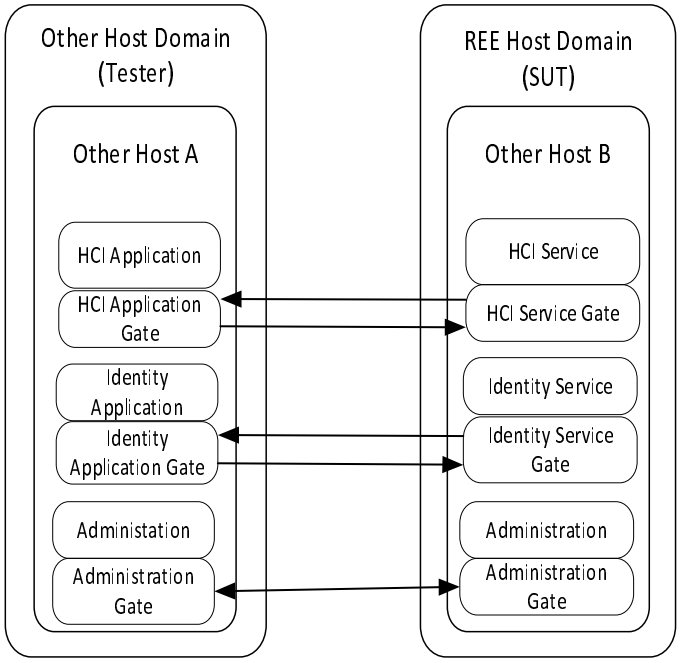
The following requirements identified in clause 5.6.6 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible.

RQ1006_001
RQ1006_002
RQ1006_003
RQ1006_004
RQ1006_005
RQ1006_021

10.7 Contactless related applications support

10.7.1 Configurations

10.7.1.1 CHCP_001 - HCP tunnelling over SCL

Configuration identification	CHCP_001
Configuration description	

10.7.2 Procedures

10.7.2.1 PHCP_021 - Open a pipe session with the Identity gate

Procedure identification	PHCP_021
Procedure objectives	The host A(tester) shall be able to open a pipe session to the identity service gate of the other host in the SUT. From the GATE_LIST registry, the UUID of the HCI service shall be listed.
Configuration reference	CHCP_001
Initial conditions	
None.	
Test sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the host A with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the host B with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
3	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate in the other host with the register '04'H.
4	Identity service gate sends ANY_GET_PARAMETER response (pipe $PIPE_{YX}$) to the identity application gate. The HCI service identifier shall be present. The procedure is successful if the previous requirement is satisfied.
5	Administration gate sends EVT_ADM_UNBIND event to the administration gate in the host A with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. The pipe session between the Identity application gate and the Identity service gate is closed.

10.7.2.2 PHCP_022 - Open a pipe session with the HCI service

Procedure identification	PHCP_022
Procedure objectives	The HCI application gate shall be able to open a pipe session to the HCI service gate. If the test is successful, then a pipe session is open between the HCI application in the host A and the HCI service in the host B.
Configuration reference	CHCP_001
Initial conditions	
The test HCP_021 shall be successfully executed.	
Test sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the other with: • $PIPE_{BA}$: a dynamically assigned pipe identifier for the HCI control service gate. • $GATE_{HCI}$: The UUID gate identifier of the HCI service gate (213CA645-9A22-5C5D-B340-60212840015B).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with a binding parameter equal to: • $PIPE_{AB}$: a dynamically assigned pipe identifier for the HCI control application gate. • $GATE_{URN}$: The UUID gate identifier of the HCI control service gate (213CA645-9A22-5C5D-B340-60212840015B). The procedure is successful if the pipe session is open.

10.7.3 Test descriptions

10.7.3.1 HCP_001 - HCP tunnelling over SCL-1

Test identification	HCP_ 001	
Test objectives	To verify HCI Host Controller's HCP tunnelling procedure over SCL. The pipe session between the HCI application and the HCI service acts as a link layer as defined in ETSI TS 102 622 [5], clause 4.1. To verify that the HCP tunneling can be operated as defined in ETSI TS 102 622 [5], the tests defined in ETSI TS 102 695-1 [3] shall be executed using configuration CHCP_001. Test case defined in ETSI TS 102 695-1 [3] are to be handled as test descriptions defined in the present document, where the term 'HS' shall be replaced by 'Tester' is a HCI host as defined in ETSI TS 102 622 [5] embedded in the host A and the term 'HCUT' shall be replaced by 'SUT' is a HCI host embedded with host B.	
Configuration reference	CHCP_001	
Initial conditions		
The procedure PHCP_022 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The tester shall successfully execute all test descriptions as defined in ETSI TS 102 695-1 [3].	RQ1007_003 RQ1007_004 RQ1007_005 RQ1007_006 RQ1007_007 RQ1007_008

10.7.3.2 HCP_002 - HCP tunnelling over SCL-2

Test identification	HCP_002	
Test objectives	To verify HCI Host Controller's HCP tunnelling procedure over SCL. The pipe session between the HCI application and the HCI service acts as a link layer as defined in the ETSI TS 102 622 [5], clause 4.1. To verify that the HCP tunneling can be operated as defined in ETSI TS 102 622 [5], the tests defined in ETSI TS 102 695-2 [4] shall be executed using configuration CHCP_001. Test cases defined in ETSI TS 102 695-2 [4] are to be handled as test descriptions defined in the present document, where the term 'HCS' shall be replaced by 'Tester' is a HCI host as defined in ETSI TS 102 622 [5] embedded in the host A and the term 'HUT' shall be replaced by 'SUT' is a HCI host embedded with host B.	
Configuration reference	CHCP_001	
Initial conditions		
The procedure PHCP_022 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The tester shall successfully execute all test descriptions as defined in ETSI TS 102 695-2 [4].	RQ1007_003 RQ1007_004 RQ1007_005 RQ1007_006 RQ1007_007 RQ1007_008

10.7.3.3 HCP_003 - limited pipe session

Test identification	HCP_003	
Test objectives	The objective of the test is to verify than at most one pipe session on the HCP service gate shall be open.	
Configuration reference	CHCP_001	
Initial conditions		
The procedure PHCP_022 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The procedure PHCP_022 shall fail	RQ1007_002

10.7.3.4 Requirements not testable, implicitly verified or verified elsewhere

10.7.3.4.1 Requirements not tested

The following requirements identified in clause 5.6.7 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ1004_043, RQ1004_008

10.7.3.4.2 Implicit requirements

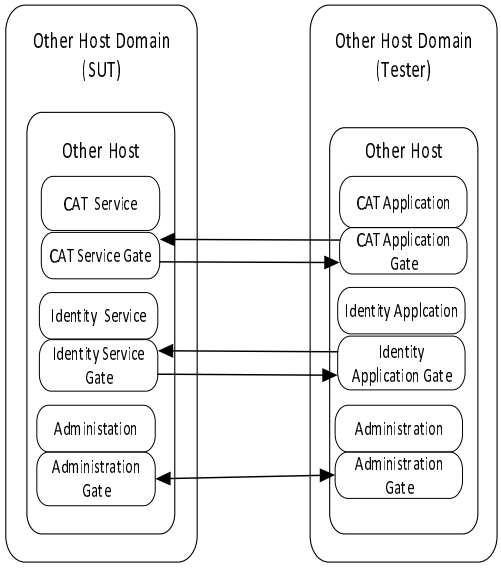
By executing the test descriptions defined in ETSI TS 102 695-1 [3], clause 5.2 the following requirements are implicitly tested:

RQ1007_001

10.8 Card Application Toolkit (CAT) over SCL

10.8.1 Configurations

In addition the following configurations are used in this clause.

Configuration identification	CSCL_001
Configuration description	

10.8.2 Procedures

10.8.2.1 PSCL_001 - Open a pipe session with the Identity gate of the Other host (SUT)

Procedure identification	PSCL_001
Procedure objectives	The Other host (Tester) shall be able to open a pipe session to the identity gate of the Other host (SUT).
Configuration reference	CSCL_001
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the Other host (SUT) with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the identity service gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the Other host (Tester) with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the identity application gate. • $GATE_{IDENTITY}$: The UUID gate identifier of the identity gate (416B66AC-A134-5082-8160-FA1BA497F917).

10.8.2.2 PSCL_002 - Open a pipe session with the CAT gate

Procedure identification	PSCL_002
Procedure objectives	The gate identifier of the CAT service is FF00453F-B0D5-59CE-B0D4-3AE178432F73 and is related to the REE only in order to be independent of the configuration.
Configuration reference	CSCL_001
Initial conditions	
Procedure sequence	
Step	**Description**
1	Administration gate sends EVT_ADM_BIND to Administration gate in the Other host (SUT) with: • $PIPE_{XY}$: a dynamically assigned pipe identifier for the CAT service gate. • $GATE_{CAT}$: The UUID gate identifier of the CAT gate (FF00453F-B0D5-59CE-B0D4-3AE178432F73).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the Other host (Tester) with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the CAT application gate. • $GATE_{CAT}$: The UUID gate identifier of the CAT gate (FF00453F-B0D5-59CE-B0D4-3AE178432F73).

10.8.3 Test descriptions

10.8.3.1 SCL_001 - CAT Service Gate URN in REE

Test identification	SCL_001	
Test objectives	Verify GATE Identifier of CAT Service gate to ensure CAT Service Gate supports URN syntax as defined in ETSI TS 103 666-1 [1], clause 8.2, with the values specified in ETSI TS 103 666-1 [1], Table 8.1.	
Configuration reference	CSCL_001 with REE host domain as Other host domain (SUT)	
Initial conditions		
PSCL_001 is successfully run.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity Application Gate in Other host (Tester) sends command "ANY_GET_PARAMETER" with parameter "GATE_LIST" ('04' H) to the Identity Service Gate in REE host domain: • Verify that this list contains GATE Identifier"FF00453F-B0D5-59CE-B0D4-3AE178432F73".	RQ1008_020

10.8.3.2 SCL_002 - CAT Service Gate URN in MBM

Test identification	SCL_002	
Test objectives	Verify GATE Identifier of CAT Service gate to ensure CAT Service Gate supports URN syntax as defined in ETSI TS 103 666-1 [1], clause 8.2, with the values specified in ETSI TS 103 666-1 [1], Table 8.1.	
Configuration reference	CSCL_001 with MBM host domain as Other host domain	
Initial conditions		
PSCL_001 is successfully run.		
Test sequence		
Step	**Description**	**Requirements**
1	Identity Application Gate in Other host (Tester) sends command "ANY_GET_PARAMETER" with parameter "GATE_LIST" ('04' H) to the Identity Service Gate in MBM host domain: • Verify that this list contains GATE Identifier "3D16542C-691F-53DB-A62A-B5AEF296159B".	RQ1008_020

10.8.3.3 SCL_003 - CAT Service Gate testing procedure

Test identification	SCL_003	
Test objectives	Verify CAT Service Gate Procedures	
Configuration reference	CSCL_001	
Initial conditions		
PSCL_001 is successfully run. PSCL_002 is successfully run.		
Test sequence		
Step	**Description**	**Requirements**
1	CAT Application Gate (Tester) sends event "EVT_PROACTIVE_CMD" ('10' H) with some new (undefined) COMPREHENSION-TLV data objects at end of command to CAT service gate (SUT): • Verify that there is no error from CAT Service Gate (SUT).	RQ1008_012 RQ1008_021 RQ1008_023
2	CAT Service Gate (SUT) shall issue back "EVT_TERMINAL_RESPONSE" ('11' H): • Verify that this is received to CAT Application Gate (Tester). • Verify that this message follows structure as defined in ETSI TS 102 622 [5], clause 5.2 HCP message structure. • Verify that it follows BER-TLV encoding. • Length encoding of this shall be verified against "Table 10.15: Length encoding" of ETSI TS 103 666-1 [1]. • Verify that COMPREHENSION-TLV data objects are provided in order as defined for this in ETSI TS 102 223 [9].	RQ1008_005 RQ1008_007 RQ1008_009 RQ1008_011 RQ1008_013 RQ1008_017 RQ1008_018
3	Force CAT Service Gate (SUT) to send some info which should be encapsulated in envelope: • Verify that "EVT_ENVELOPE_CMD" ('10' H) is received to CAT Application Gate (Tester). • Verify that this message follows structure as defined in ETSI TS 102 622 [5], clause 5.2 HCP message structure. • Verify that it follows BER-TLV encoding. • Length encoding of this shall be verified against "Table 10.15: Length encoding" of ETSI TS 103 666-1 [1]. • Verify that COMPREHENSION-TLV data objects are provided in order as defined for this in ETSI TS 102 223 [9]. • First byte of BER-TLV is constant, verify that it indicates a CAT command as defined in ETSI TS 101 220 [11] "Card application toolkit templates".	RQ1008_005 RQ1008_007 RQ1008_008 RQ1008_009 RQ1008_011 RQ1008_013 RQ1008_015 RQ1008_016
4	CAT Application Gate (Tester) sends back "EVT_ENVELOPE_RSP" ('11' H) with wrongly constructed length byte: • Verify that it should be treated as error and message shall be rejected. • Further action to be performed on with this response is not taken by CAT Service (SUT).	RQ1008_010 RQ1008_021
5	CAT Application Gate (Tester) sends back event "EVT_ENVELOPE_RSP" to CAT Service Gate (SUT), including optional payload and followed by SW1/SW2 Verify that there is no error from CAT Service Gate (SUT).	RQ1008_021 RQ1008_025

10.8.3.4 Requirements not tested

The following requirements identified in clause 5.6.8 are either generated from descriptive text or not testable in the defined test environment. A verification of the listed requirements is not possible:

RQ1008_002	RQ1008_014	RQ1008_024
RQ1008_003	RQ1008_019	RQ1008_026
RQ1008_004	RQ1008_022	RQ1008_027
RQ1008_006		

10.9 Access control protocol

There are no requirements for test descriptions related to clause 10.9 of ETSI TS 103 666-1 [1].

Annex A (normative): SSP Initial State

If the SSP under test supports the SSP File System it shall be configured for testing with a root directory "SSPFS" as defined in clause 6.6.4.

Annex B (informative): Change History

The table below indicates all changes that have been incorporated into the present document since it was published.

Change history								
Date	**Meeting**	**Plenary Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
08/07/2021	SCP#100	SCP(21)000098	-	-	-	Version 15.0.0 first publication	-	15.0.0

History

Document history		
V15.0.0	September 2021	Publication